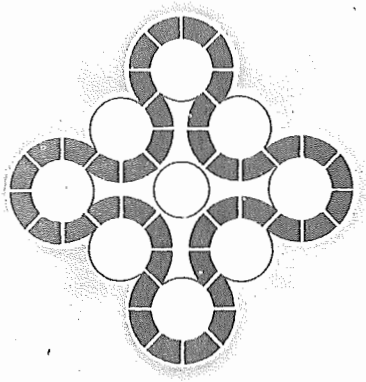
PARTICULARITIES OF FAULT TREE ANALYSIS

J.B. Fussell*



Aerojet Nuclear Company

IDAHO NATIONAL ENGINEERING LABORATORY

Idaho Falls, Idaho — 83401

DATE PUBLISHED—SEPTEMBER 1974

This document was prepared by Aerojet Nuclear Company in partial fulfillment of Contract N00030-73-C-004 with Automation Industries, Inc./Vitro Laboratories Division.

PARTICULARITIES OF FAULT TREE ANALYSIS

J. B. Fussell

September 1974

This document was prepared by Aerojet Nuclear Company in partial fulfillment of Contract N00030-74-C-0007 with Automation Industries, Inc./Vitro Laboratories Division.

ACKNOWLEDGMENTS

Acknowledgment is given to E. V. Waite for work done concerning house events. Also acknowledgment and gratitude are extended to N. D. Cox for his suggestions.

PREFACE

Fault tree analysis is a method of system reliability analysis that is generally applicable to complex dynamic systems. The purpose of this document is to present methods for treating several specialized situations encountered in practice.

OR logic gates are of three types. Although the inclusive OR gate is most commonly encountered in practice, occasionally the exclusive OR gate and the mutually exclusive OR gates are needed. These logic gates are discussed in Section I.

Transfers within fault trees often result in erroneous fault trees. The correct procedure for establishing transfers in a fault tree is given in Section II.

Circular failure logic is occasionally encountered during fault tree construction. If not properly treated, circular failure logic results in a fault tree from which the minimal cut sets cannot be determined by a computer. Proper treatment of circular failure logic is put forth in Section III.

Quantitative results from reliability analyses are becoming of increasing importance. Proper interpretation of several types of quantitative results are presented in Section IV.

In practice many plant systems are periodically shut down for maintenance and repair. Quantitative evaluations of such systems are discussed in Section V.

Component mean times to repair are required as input to quantitative evaluations of repairable systems. The results of such analyses are sensitive to these mean times to repair. Methods for determining these mean times to repair are given in Section VI.

Mutually exclusive events result in a major pitfall of fault tree analysis. Proper treatment of these events is presented in Section VII.

Section VIII contains a discussion of the use of house events. House events allow analysis of a specific situation from a generalized fault tree as well as provide additional visibility from the fault tree.

Finally, in Section IX, situations that require the priority AND logic gate are discussed.

CONTENTS

FIGURES

TABLES

I. THE VARIOUS TYPES OF OR LOGIC GATES

OR gates are of three types, the inclusive OR, the exclusive OR, and the mutually exclusiye OR. All these OR logic gates have the property of passing an output if one input event occurs.

1. THE INCLUSIVE OR GATE

The inclusive OR situation is the one commonly encountered in practice. This gate passes an output if one or more inputs occur. The truth table for an inclusive OR gate with three input events is given in Table I. The Venn diagram for the inclusive OR gate with three input events, A, B, and C, is shown in Figure 1. $A \cap B$ is read "A and B" and denotes the simultaneous occurrence of both A and B.

The probability equation for an inclusive OR gate with three inputs is

$$P_{(output)} = P(A) + P(B) + P(C) - P(A \cap B) - P(B \cap C) - P(A \cap C) + P(A \cap B \cap C). \quad (1)$$

TABLE I

TRUTH TABLE FOR AN INCLUSIVE OR GATE WITH THREE INPUT EVENTS

Input Events			Output Event
A	B	C	O
true	false	false	true
false	true	false	true
false	false	true	true
true	true	false	true
true	false	true	true
false	true	true	true
true	true	true	true
false	false	false	false

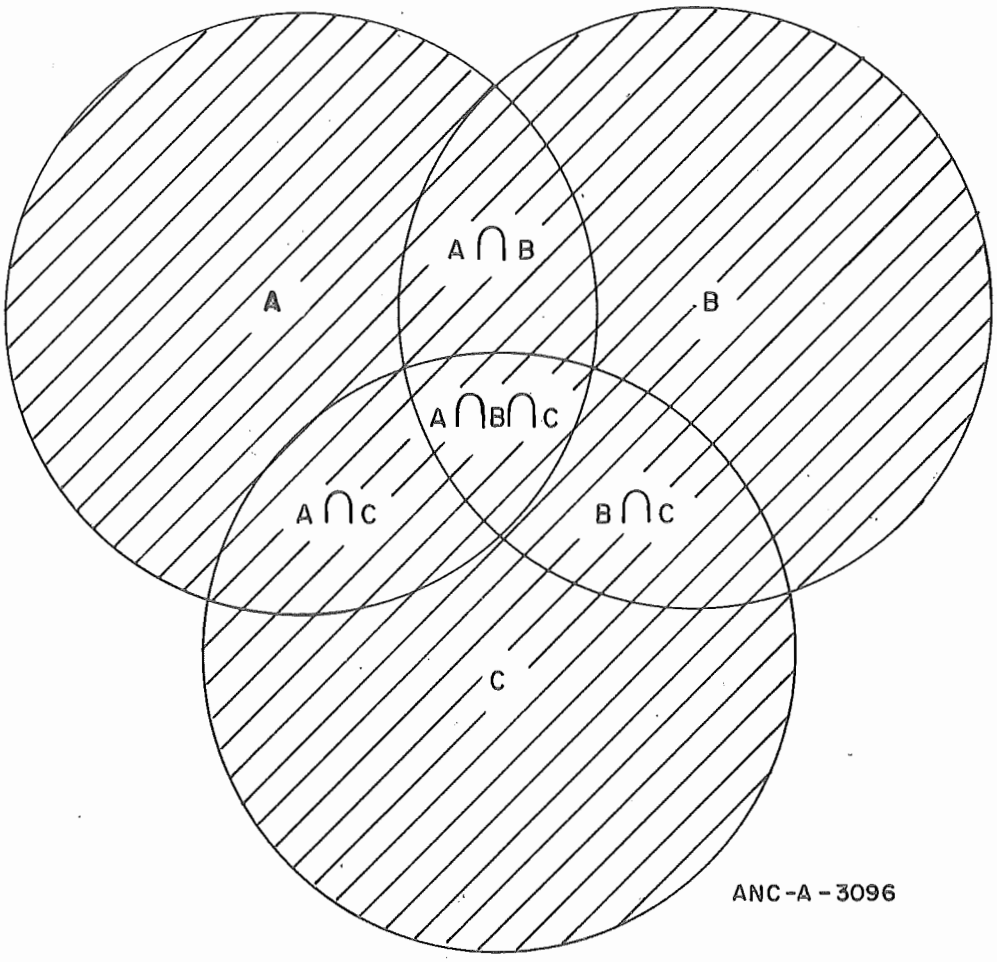


Fig. 1 Venn diagram for an inclusive OR gate with three input events.

The reason for the negative and positive terms in Equation (1) can be seen by studying Figure 1 while considering the area enclosed by each event to represent the probability of occurrence of the event. The shaded area then represents the probability of occurrence of the output event. By adding and subtracting areas as described by Equation (1) the shaded area is obtained. The general equation for an inclusive OR gate with N input events is given in Reference 1.

Inclusive OR gates are generally the most frequently used gate when modeling Boolean failure logic. The input events are a restatement of the output event and the occurrence of any input event immediately results in the output event. If more than one input event can occur (even if such an occurrence "probably" will not happen) and, given such an occurrence, the output event will exist, then the inclusive OR gate should be used.

An example of an inclusive OR situation is encountered during the development of a relay failing open. The relay fails open if the contacts corrode or the coil jams open. Conceivably both the coil and the contacts could be failed causing the relay to be failed open.

2. MUTUALLY EXCLUSIVE OR GATES

The mutually exclusive OR gate passes an output when any single input occurs. The occurrence of more than one input must be physically impossible for the mutually exclusive OR gate to be applicable. The truth table for a mutually exclusive OR gate with three input events is given in Table II. The Venn diagram for a mutually exclusive OR gate with three input events, A, B, and C, is given in Figure 2.

The probability equation for this OR gate with three input events is

$$P_{(output)} = P(A) + P(B) + P(C). \tag{2}$$

An example of mutually exclusive OR situations is a system that requires that a light be manually flashed on and off by using a switch. The operator will be unable to flash the light if the switch fails open (contact cannot be made) or if the switch fails closed (contact cannot be broken). It is not, however, possible for the switch to be failed open and closed.

TABLE II

TRUTH TABLE FOR A MUTUALLY EXCLUSIVE OR GATE WITH THREE INPUT EVENTS

Input Events			Output Event
A	B	C	O
true	false	false	true
false	true	false	true
false	false	true	true
false	false	false	false

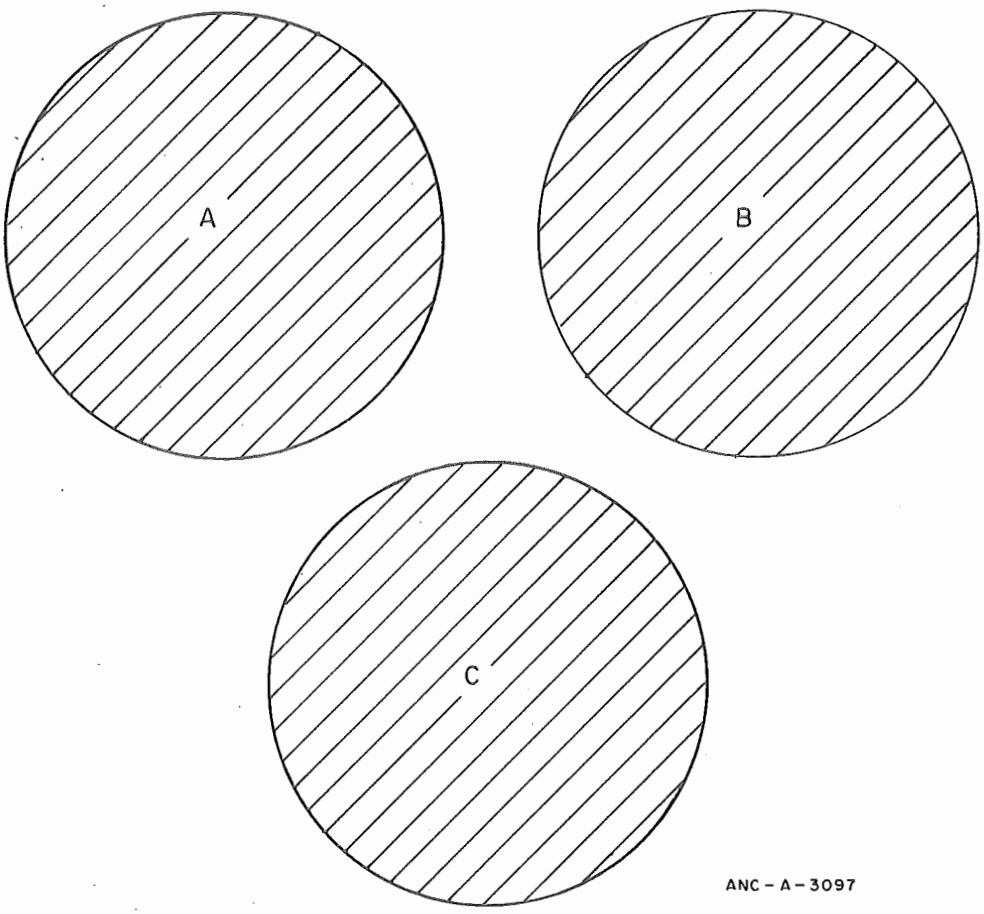


Fig. 2 Venn diagram for a mutually exclusive OR gate with three input events.

3. EXCLUSIVE OR GATES

Exclusive OR gates also pass an output when any single input occurs. If, however, more than one input event occurs, the exclusive OR gate does not pass an output. By using this definition for an exclusive OR gate, any truth table logic funtion can be constructed from inclusive OR gates, AND gates, and exclusive OR gates. The truth table for an exclusive OR gate with three input events is given in Table III. The Venn diagram for this gate with three inputs, A, B, and C, is shown in Figure 3.

The probability equation for an exclusive OR gate with two inputs, A and B, is

$$P_{(output)} = P(A) + P(B) - 2P(A \cap B). \quad (3)$$

TABLE III

TRUTH TABLE FOR AN EXCLUSIVE OR GATE WITH THREE INPUT EVENTS

Input Events			Output Event
A	B	C	O
true	false	false	true
false	true	false	true
false	false	true	true
true	true	false	false
true	false	true	false
false	true	true	false
true	true	true	false
false	false	false	false

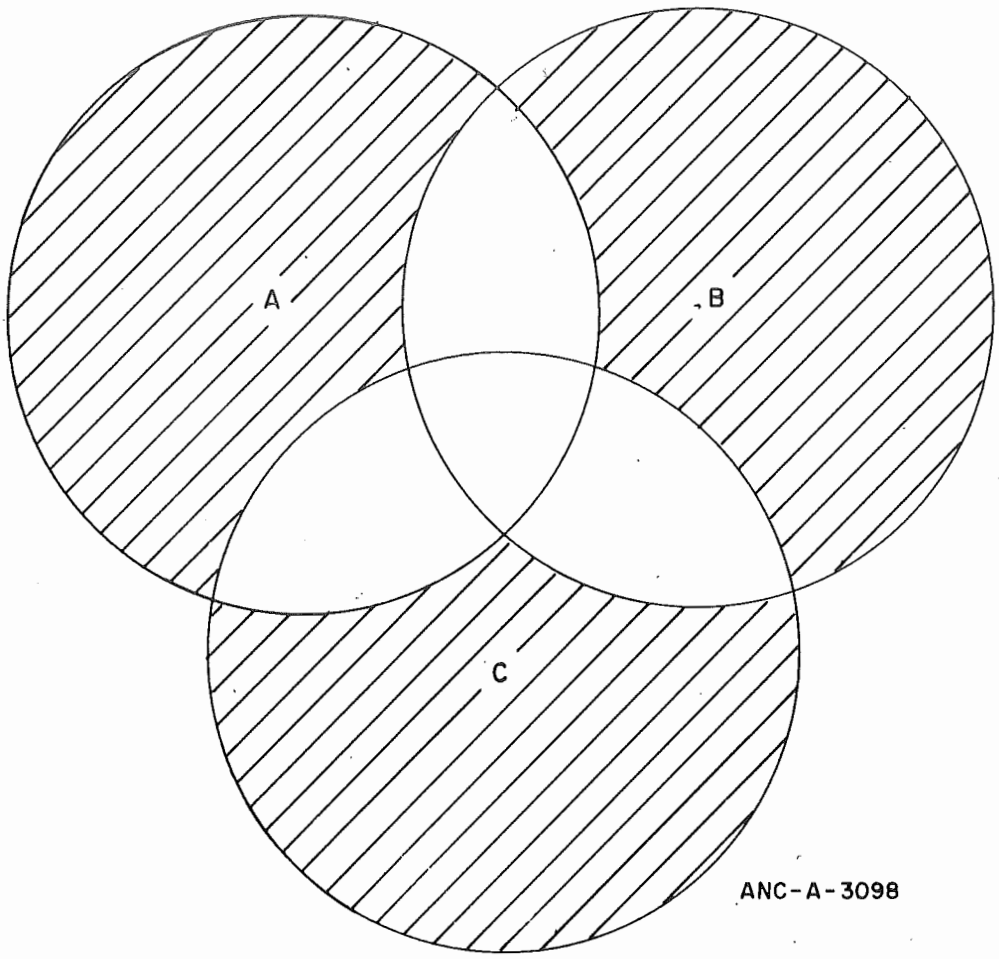


Fig. 3 Venn diagram for an exclusive OR gate with three input events.

The probability equation for an exclusive OR gate with three input, A, B, and C, is

$$P_{(output)} = P(A) + P(B) + P(C) - 2P(A \cap B) - 2P(A \cap C) - 2P(B \cap C) + 3P(A \cap B \cap C). \quad (4)$$

A classical example of the exclusive OR situation involves a twin engine aircraft, one engine on each wing. The engines occasionally burn a valve and consequently experience a 25% power loss. The aircraft being stressed because of asymetric thrust is the fault of interest. This fault occurs if one engine or the other experiences the power loss but not if both experience the power loss. Both can experience the power loss but asymmetric thrust does not result.

II. TRANSFERS WITHIN A FAULT TREE

Transfers within a fault tree are useful to abbreviate the fault tree without loss of detail. Figure 4 indicates how transfers are made within a fault tree. When a triangle symbol is reached under an event description in a rectangle, the reader of the fault tree then transfers to the triangle with the same name that is alongside of another rectangle with the same event description. Many transfers can be made to the same event description. If the fault tree requires many pages, good practice is to include in the triangle symbol the number of the page that contains the event description to which the transfer is made.

A pitfall of fault tree analysis is to use transfers to reduce fault tree construction effort. A transfer cannot necessarily be made simply because two event descriptions are identical. Events with the same event descriptions can require different logical development because the "effective boundary conditions"[2] of the events can be different. Only if all the events are developed and the development is identical can a rigorous transfer be guaranteed. The transfer then abbreviates the fault tree itself but not its construction time.

Also circular failure logic can result from transfers in a tree. This subject is discussed in the next section.

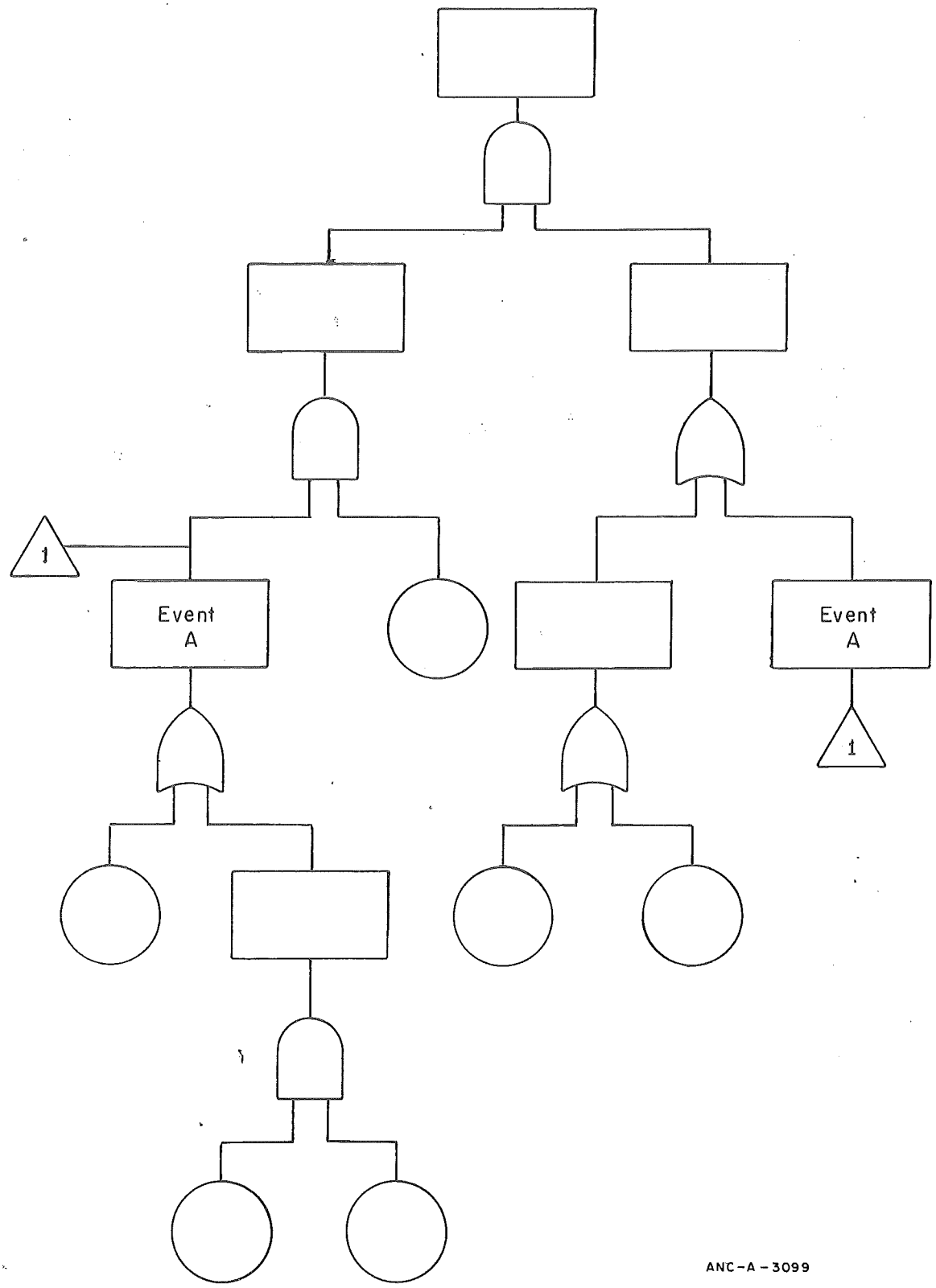


Fig. 4 Illustration of a transfer within a fault tree.

III. CIRCULAR FAILURE LOGIC

Circular failure logic is occasionally encountered in practice during the construction of a fault tree. Circular failure logic is the situation encountered when, during the development of an event, A, the identical event A occurs again in a lower tier of the fault tree[a]. Presently used methods cannot locate the minimal cut sets for a fault tree containing circular failure logic.

An elementary example is given here to illustrate a circular failure logic situation. The reset switch shown in the schematic given in Figure 5 is closed to latch the circuit and provide current to the light bulb. The system boundary conditions for fault tree construction are as follows:

TOP Event	No current in Circuit 1
Initial Conditions	Switch closed
	Relay A contacts closed
	Relay B contacts closed
	Reset switch open
Not-Allowed Events	Wiring failures
	Operator failures
	Switch failure
	Common mode failures
Existing Events	Reset switch open

The fault tree for this system showing circular failure logic is given in Figure 6.

---

[a] For two events to be identical the event descriptions and the "effective boundary conditions"[2] of the events must be identical. That is, the events must be developed exactly the same. Two events are not identical simply because their event descriptions are identical.

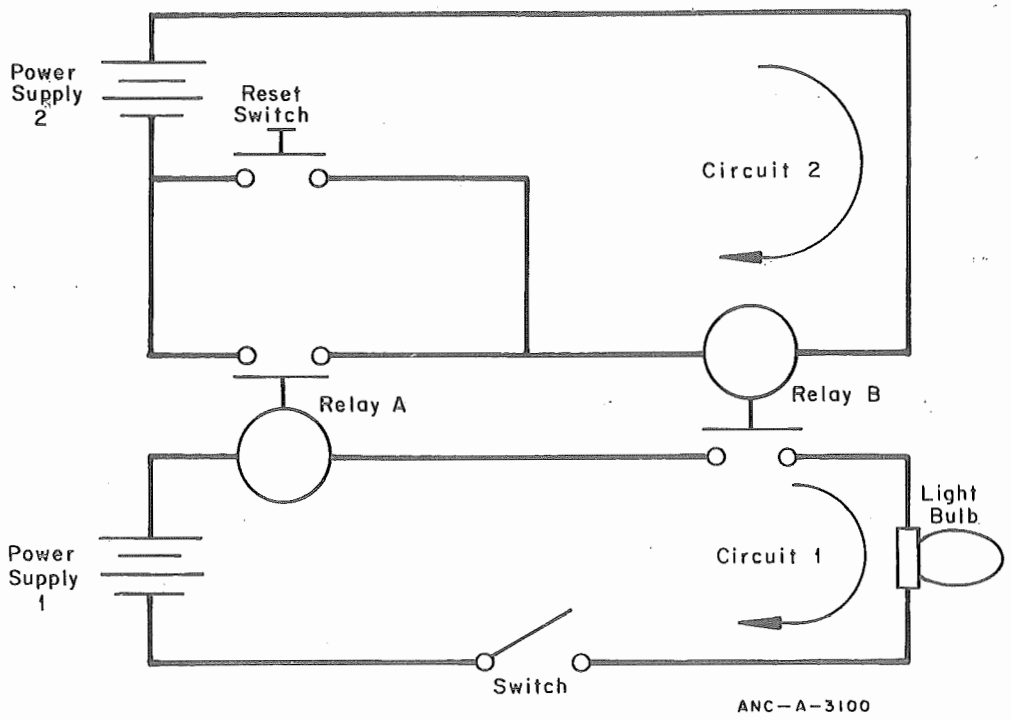


Fig. 5 Schematic for circular failure logic example.

To remove any circular failure logic situation, the second (lower tier) occurrence of the event is treated as an event with zero probability of occurrence. If the second occurrence of the event is attached to an OR gate, the event is removed from the fault tree. If the event is attached to an AND gate, the entire AND gate and all immediately preceding AND gates up to the next OR gate are removed from the fault tree. The justification for this approach is given in Appendix A.

The fault tree given in Figure 6 is shown in Figure 7 with the circular failure logic removed.

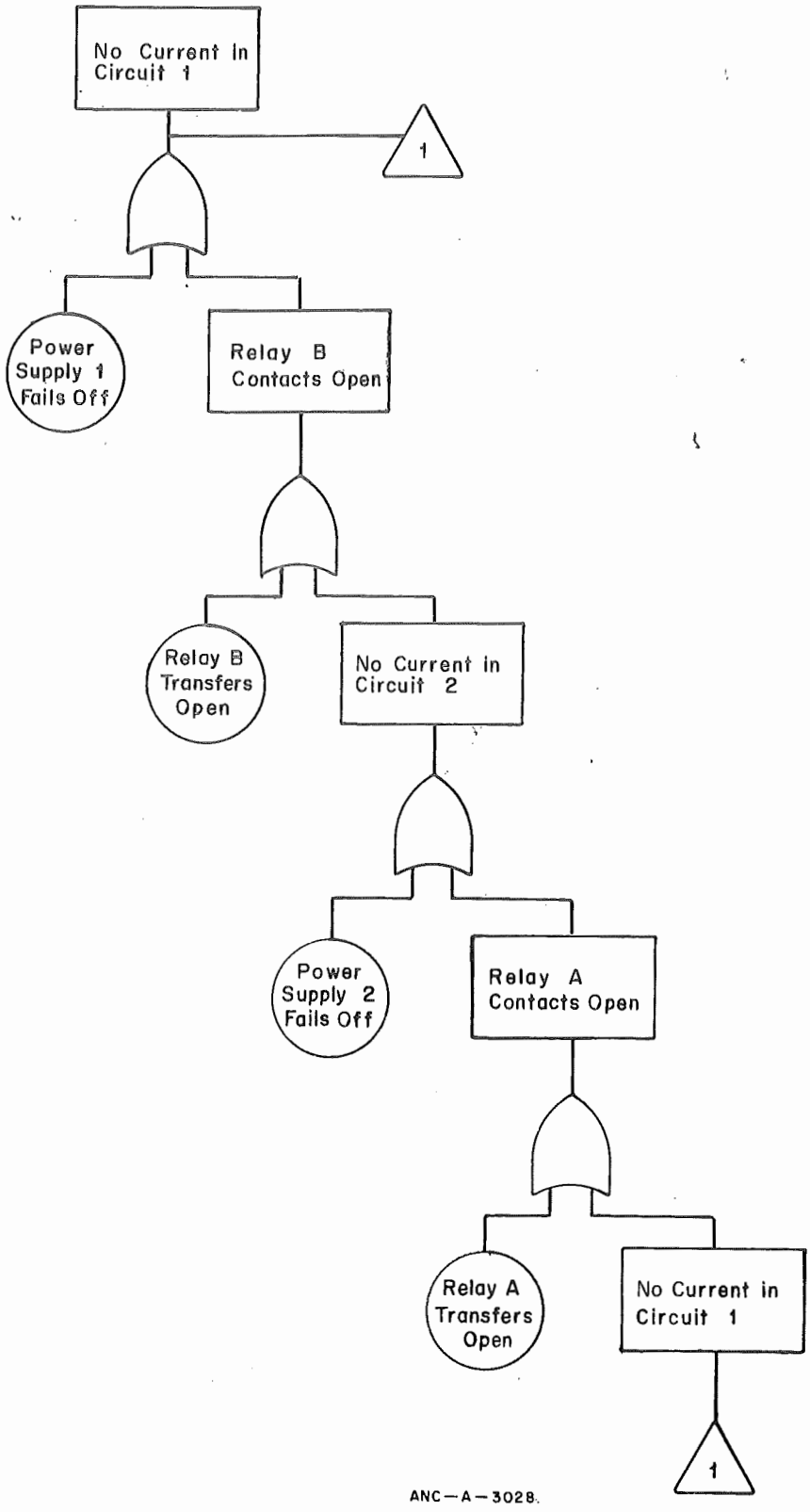


Fig. 6 Fault tree showing circular failure logic.

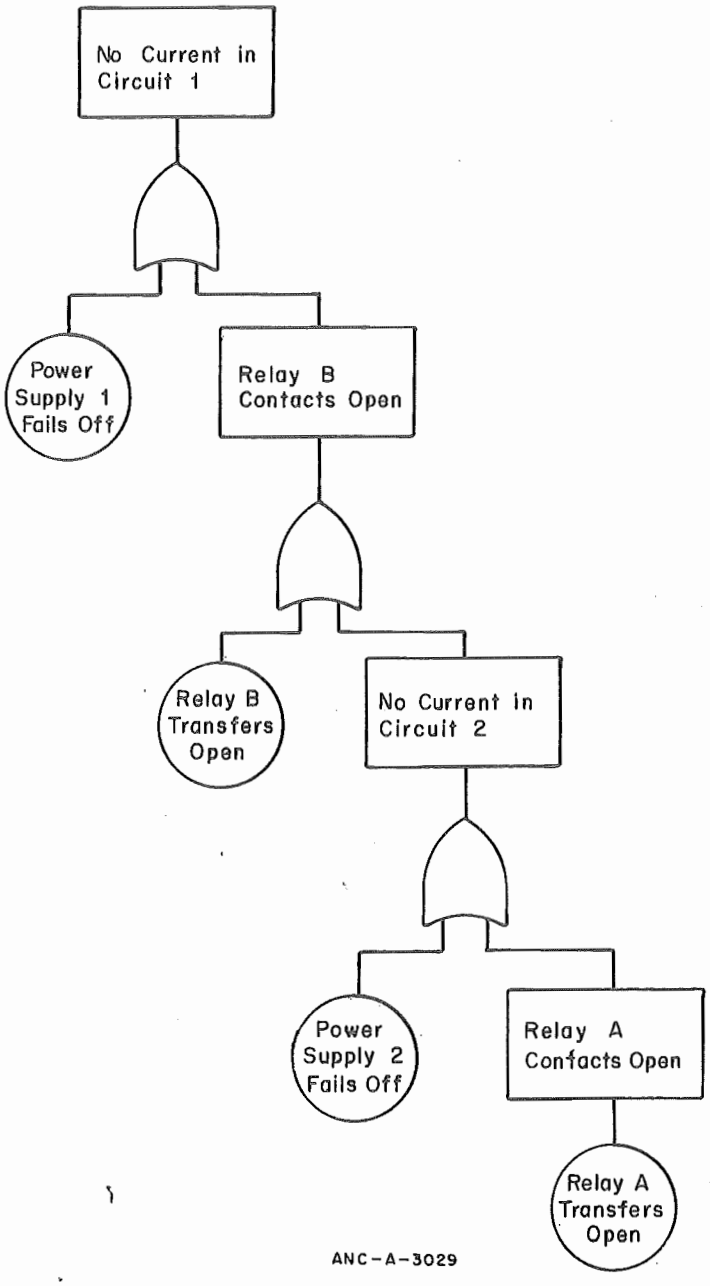


Fig. 7 Fault tree shown in Figure 6 with circular failure logic removed.

IV. QUANTITATIVE SYSTEM RELIABILITY ANALYSIS RESULTS

Quantitative reliability analysis is of value for analysis of system design, demonstration of compliance with safety requirements, and justification of system changes or additions. The steps in a system reliability analysis that usually precede a quantitative analysis are:

(1) Definition of the system to be analyzed

(2) Construction of the reliability logic model

(3) Qualitative analysis.

These steps are discussed in References 3, 4, 5, and 6. On completion of the qualitative analysis, the system modes of failure are determined. These modes of failure, often called minimal cut sets, are collections of independent component failures, called primary failures, that collectively cause the undesired system failure. If the system failure is to occur by a given mode, then the occurrence of all the primary failures in that minimal cut set is necessary and sufficient to cause the system failure. A collection of all the minimal cut sets represents all the unique modes of system failure. At present, generally applicable quantitative reliability analysis methods, with the exception of the Monte Carlo technique, require the system minimal cut set as input.

From these minimal cut sets and the component reliability characteristics, the reliability characteristics for a particular system failure can be calculated. Component reliability characteristics are completely described by their time-dependent failure rate (hazard rate) and their time-dependent repair rate. In practice, both failure rates and repair rates are often assumed to be constant. Failure rates for various types of equipment are given in References 7 and 8. Repair rates are generally facility dependent and must be determined by the analyst as discussed in Section VI.

From these failure rates, repair rates, and minimal cut sets, the time-dependent characteristics for a particular system failure that can be calculated include:

(1) Availability and unavailability

(2) Reliability and unreliability

(3) Rate of failure

(4) Expected number of failures

(5) Failure rate

(6) Repair rate.

1. AVAILABILITY AND UNAVAILABILITY

The availability is the probability the system failure does not exist at some specified time in the future. The unavailability is the probability the system failure does exist and numerically is equal to unity minus the availability. The general assumption used in the availability calculation is that no system component failures exist at time t = 0.

Availability is usually associated with repairable systems. In which case, the availability at time t contains no information about system failures before time t. Therefore, the availability is a useful characteristic if the occurrence of the specified system failure is tolerable at least some expected fraction of the time. The average availability during a time interval is the expected fraction of the time the system failure exists. Therefore, the availability is a useful characteristic when system downtime has economic implications.

As an example of a system for which unavailability is a factor of merit, a building fire extinguishing system that will extinguish any building fire, given it functions, is considered. The unavailability is the probability the system will not be available to extinguish a fire because a sufficient number of system components are failed. Although for the extinguishing system to fail to operate during a fire would probably be considered intolerable, the unavailability still has merit because a building fire requiring the system is expected to exist only a small fraction of the time. The unavailability at time t can be interpreted here as the probability the fire is not extinguished, given a fire exists.

In summary, availability and unavailability are

(1) Generally reported only for repairable systems

(2) Useful characteristics if the occurrence of the system failure is tolerable at least some expected fraction of the time.

Techniques for calculating availability and unavailability are well documented and appear in References 9 and 10.

2. RELIABILITY AND UNRELIABILITY

The reliability is the probability the system has experienced no failures to a given time t. The unreliability is the probability the system has experienced one or more failures to time t and numerically is unity minus the reliability.

Reliability is generally a pertinent characteristic of both repairable and nonrepairable systems. For nonrepairable systems, the reliability is numerically equivalent to the availability since if the system fails, it is thereafter unavailable. For nonrepairable systems, reliability is conventionally the reported quantity rather than availability.

Repairable system reliability calculations account for the possibility of a redundant component failing and being repaired without system failure necessarily occurring. Repairable system time-dependent reliability, called the distribution of time to first failure by mathematicians, cannot be calculated exactly for the general case. Excellent, conservative, tightly bounding approximations for repairable systems do exist, however, for the distribution of time to first failure[11,12].

System reliability is in general an interesting system characteristic and is the pertinent factor of merit when the system failure is considered intolerable. For catastrophic system failures, such as explosions, system reliability is reported; system availability has no engineering meaning for such failures.

3. RATE OF FAILURE AND EXPECTED NUMBER OF FAILURES

The rate of failure as a function of time is the expected number of failures per unit time at time t. The rate of failure should not be confused with the failure rate (a misnomer) to be discussed later. The changes in the rate of failure as a function of time are of interest from a qualitative point of view as well as from a quantitative point of view. Also the rate of failure is used to calculate the expected number of failures.

The expected number of failures during a specified time interval is given by the integral of the rate of failure over that time interval. If the cost of each system failure is known, the expected number of failures multiplied by the cost per failure gives the expected cost (nonprofit insurance cost) during the time interval due to the specified system failure. This cost need not be measured in terms of money.

The expected number of failures is in a sense a more global system characteristic than either availability or reliability because it contains information about the system with respect to an entire time interval. Unlike probabilistic quantities, the expected number of failures for repairable systems can be greater than unity.

4. FAILURE RATE AND REPAIR RATE

The failure rate as a function of time t is the probability of failure during t to t + dt given no failures before t. The term failure rate is a misnomer; hazard rate is a more fitting and sometimes used name. The term failure rate is so intrenched in the literature, however, that little chance exists that it will be discarded.

The time-dependent system failure rate completely describes the density of time to first system failure and, consequently, the distribution of time to first system failure (unreliability). If the repairable system failure rate is constant, the mean time between system failures is given by the inverse of the system failure rate.

The repair rate as a function of time is the probability of repair during t to t + dt given no repair before t. The repair rate is a misnomer also for the same reason as is the failure rate. The time-dependent system repair rate completely describes the probabilistic system repair characteristics. If the repair rate is constant, the mean time to repair is the inverse of the system repair rate.

From the system failure rate and repair rate the time-dependent system reliability, availability, and expected number of failures can immediately be determined. Also, the failure rate and repair rate supply the quantities needed to treat the present system as a component in a subsequent analysis.

V. PERIODIC SYSTEM SHUTDOWN FOR MAINTENANCE AND REPAIR

Many systems undergo periodic preventive maintenance. For example, automobiles are generally serviced and necessary repairs are made at regular intervals. The time between maintenances is called the operating interval. In general, two situations can exist during the operating interval; the system can be repairable or nonrepairable.

As an example, a mechanic plans to take a long car trip from Town A to Town B, and then return to Town A. The system failure is defined as the car being unable to travel under its own power at any time during the trip. One or more cylinders misfiring is then not a system failure so long as the car can still be driven.

The mechanic will carry enough spare parts and tools to make any component repairs. All such component repairs will be made when discovered during the entire trip. Before leaving Town A and upon arrival at Town B, the car will be serviced and all defective components will be repaired. The first operating interval is the time to drive from Town A to Town B. The second and final operating interval is the time to drive back from Town B to Town A. This example is one of a system that is repairable during the operating intervals.

On another occasion the mechanic's wife plans to make the same trip, but will not make any repairs during the trip. However, before leaving Town A and upon arrival in Town B, the car will be serviced and all defective components will be repaired. This example is then one of a system that is nonrepairable during the operating intervals.

In either case, the system reliability is the characteristic of interest if the system failure is considered intolerable. If, however, the analyst is interested in the probability the system failure will exist at some specified time during the trip, the unavailability is of interest.

In practice, during some operating intervals the system can be repairable and during others nonrepairable; or repairable and nonrepairable phases may exist in the same operating interval. All of these situations can be treated with presently available reliability methods.

In any case, on maintenance and repair, the unavailability and unreliability return to zero. The calculation is then carried out for the next operating interval with applicable component characteristics. A typical example plot of reliability as a function of time over several operating intervals is given in Figure 8. Availability plots are similar to reliability plots with a different curvature for repairable systems.

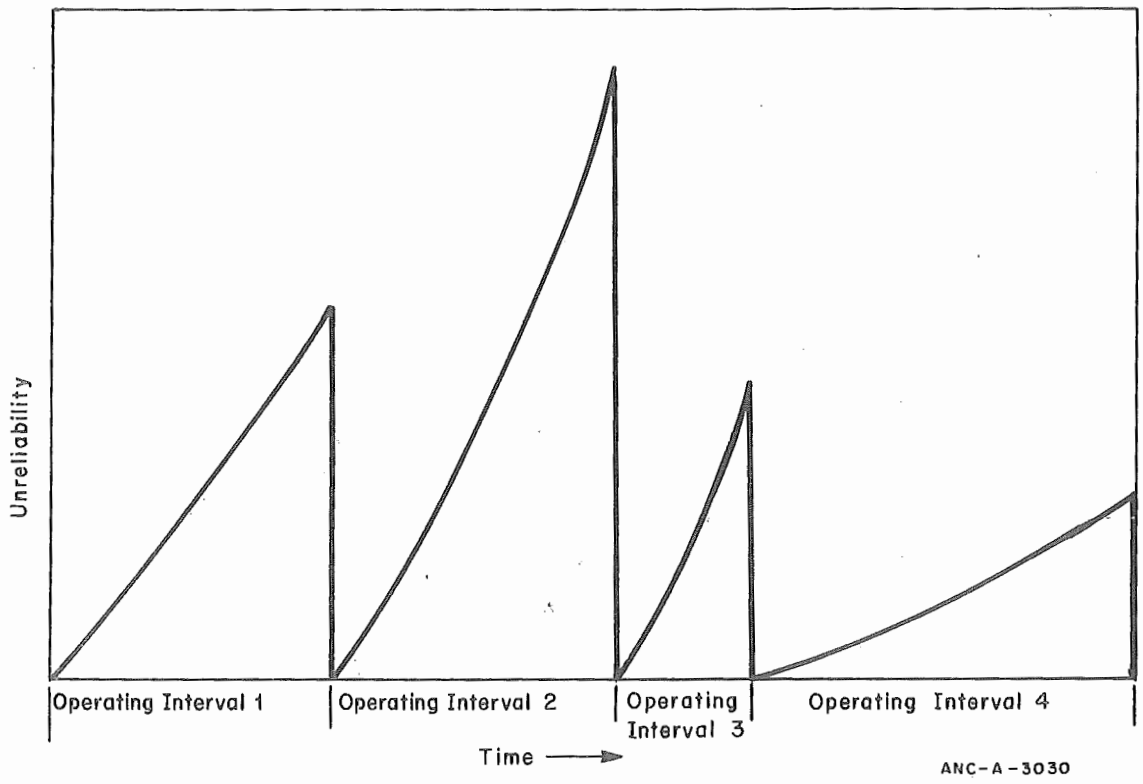


Fig. 8 Sample plot of unreliability for periodically maintained systems.

VI. DETERMINING COMPONENT MEAN TIME TO REPAIR

Values obtained from a quantitative reliability analyses are generally highly sensitive to the component mean times to repair. The component mean time to repair, τ, more appropriately called component mean dead time, is the mean time from the instant the component fails to the instant the component is restored to service.

The values are generally unique to the system being analyzed and must be determined by the analyst. For an example of determining an expression for calculating the value of τ for unannounced failures, the following terms are defined.

(1) $\overline{T}$ $\equiv$ mean time the component is failed before it is tested.

(2) $\overline{C}$ $\equiv$ mean time to request repair after the component has been tested and found to be failed.

(3) $\overline{D}$ $\equiv$ mean time for arrival of repair facility after the repair has been requested.

(4) $\overline{R}$ $\equiv$ mean time to final component restoration after arrival of the repair facility.

(5) p_1 $\equiv$ probability of failing to detect the component failure during the first test after failure.

(6) p_i $\equiv$ probability of failing to detect the component failure during the i^{th} subsequent testing given the component was failed and not detected on all previous tests.

To determine a value for $\overline{T}$ when the component fails exponentially, the testing interval is T, and T is less than one tenth the mean time between failures ($1/\lambda$), A and B are defined as

A $\equiv$ the event the component fails at time t

B $\equiv$ the event the component is failed at time T.

Then,

$$\overline{T} = \int_0^T t \, \frac{dP(A|B)}{dt} \, dt$$

and

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B|A)}{P(B)}$$

but P(B|A) equals unity since the component is not repaired before it is tested. Therefore,

$$P(A|B) = \frac{P(A)}{P(B)}$$

$$= \frac{1-e^{-\lambda t}}{1-e^{-\lambda T}} .$$

However, since

$$1 - e^{-\lambda t'} \approx \lambda t'$$

when

$$\lambda t < 0.1, \text{ then}$$

$$P(A|B) \approx \frac{\lambda t}{\lambda T} = \frac{t}{T} .$$

Therefore,

$$\overline{T} \approx \int_0^T t \frac{d(\frac{t}{T})}{dt} \, dt$$

$$\approx \int_0^T \frac{t}{T} \, dt$$

$$\overline{T} \approx \frac{T}{2} .$$

If the failure is detected on the first test, for which the probability equals $(1 - p_1)$, then $\tau = X$ where $X = (\overline{T} + \overline{C} + \overline{D} + \overline{R})$. If the failure is not detected until the next testing time, for which the probability is $[p_1(1-p_2)]$, then τ equals $X + T, \ldots$. Consequently,

$$\tau = X(1-p_1) + (X+T)\, p_1(1-p_2) + (X+2T p_1\, p_2\, (1-p_3) + \ldots$$

$$\tau = X + T \sum_{i=1}^{\infty} \left\{ \prod_{j=1}^{i} p_j \right\} . \qquad (1)$$

Cases of Equation (1) that are often of practical interest are:

Case A: $p_i = p$ all i

Case B: $p_i = p_2$ all $i \geq 2$.

For Case A, Equation (1) becomes

$$\tau = X + Tp \sum_{i=0}^{\infty} p^i \qquad (2)$$

$$\tau = X + \frac{pT}{1-p} .$$

For Case B, Equation (1) becomes

$$\tau = X + p_1 T \sum_{i=0}^{\infty} p_2^{\,i} \qquad (3)$$

$$= X + \frac{p_1 T}{1-p_2} .$$

Equations (2) and (3) are valid in many situations encountered in practice. Other expressions may be required for more unusual situations. In any case, the mean time to repair is generally extended beyond the mean time to component restoration after arrival of the repair facility.

VII. MUTUALLY EXCLUSIVE FAULT EVENTS

The anomaly of mutually exclusive primary events appearing together in one minimal cut set, when this minimal cut set is determined by conventional fault tree analysis techniques, has been long observed. Also conventional analysis techniques can declare a group of primary events, that contain no mutually exclusive primary events, as a minimal cut set when indeed this collection of events being failed will not cause system failure even if an "error free" fault tree is used as input to the analysis. The purpose of this section is to examine the causes of these erroneous minimal cut sets and to indicate how present methods of determining minimal cut sets must be modified to remove this deficiency.

Methodologies to determine the minimal cut sets from a fault tree have been described by Vesely and Narum[14] and Fussell, Henry, and Marshall[15]. These methodologies are limited in that they cannot handle fault trees with mutually exclusive fault events if these fault events occur in the domain of the same AND logic gate.

1. AN EXAMPLE TO ILLUSTRATE THE PROBLEM

Obtaining erroneous minimal cut sets is best illustrated by an example. The system schematic for the example is shown in Figure 9. The purpose of the system is to provide light from the bulb. When the switch is closed, the relay contacts close and the contacts of the circuit breaker, defined here as a normally closed relay, open. Should the relay contacts transfer open the light will go out and the operator will immediately open the switch which in turn causes the circuit breaker contacts to close and restore the light. The system boundary conditions are then:

TOP event	No light
Initial Conditions	Switch closed
	Relay contacts closed
	Circuit breaker contacts open
Not-Allowed Events	Operator failures
	Wiring failures
	Secondary failures.

Operator failures, wiring failures, and secondary failures are neglected to simplify the resulting fault tree. This fault tree is shown in Figure 10.

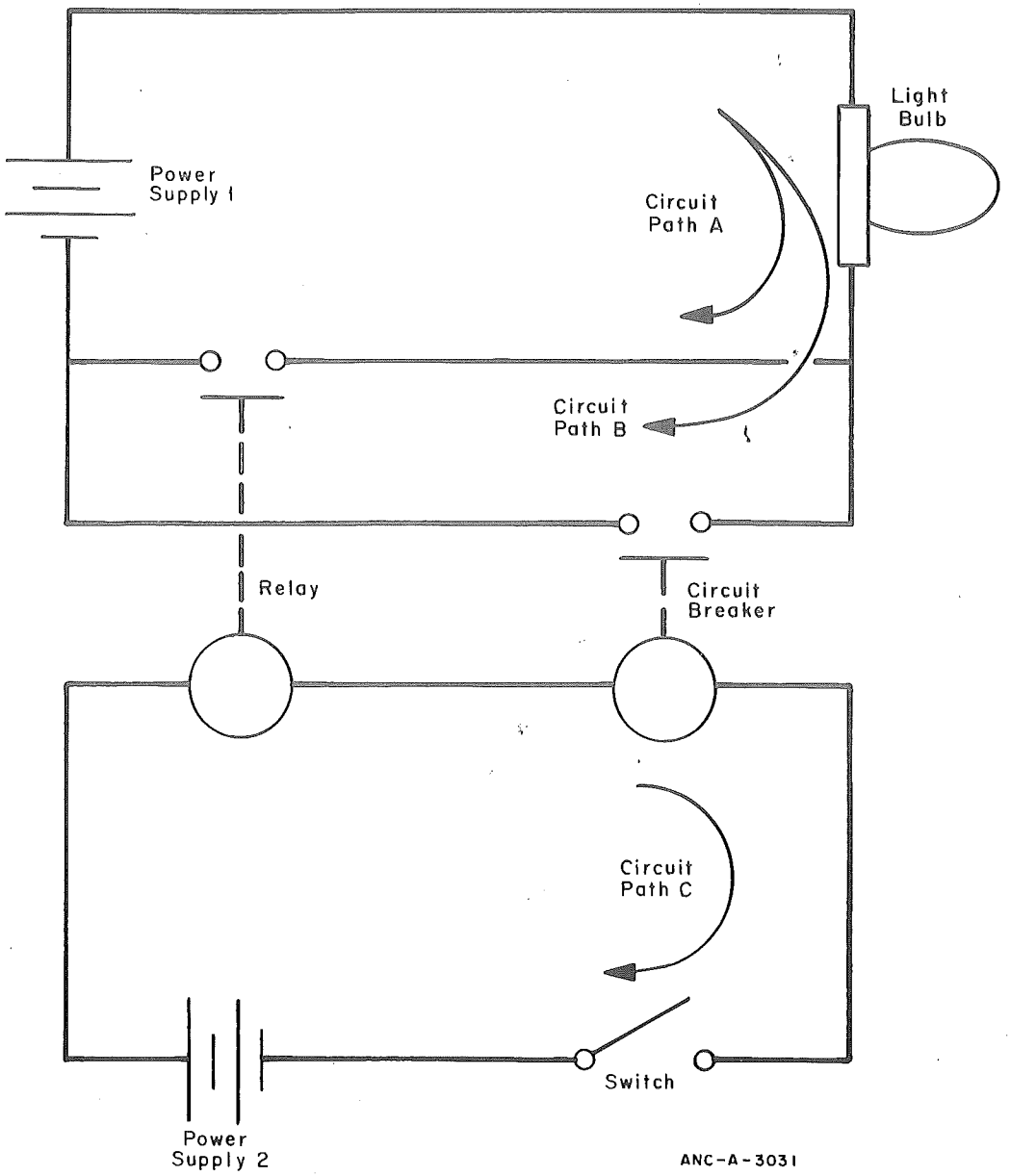


Fig. 9 Sample system for mutually exclusive events.

Table IV gives collections of primary events that are declared to be minimal cut sets by conventional methods of determining minimal cut sets.

As can be reasoned from Figure 10, Sets (6), (8), (10), and (12) of Table IV will not cause the TOP event. Only Set (12) being erroneous could have been detected from the minimal cut sets themselves.

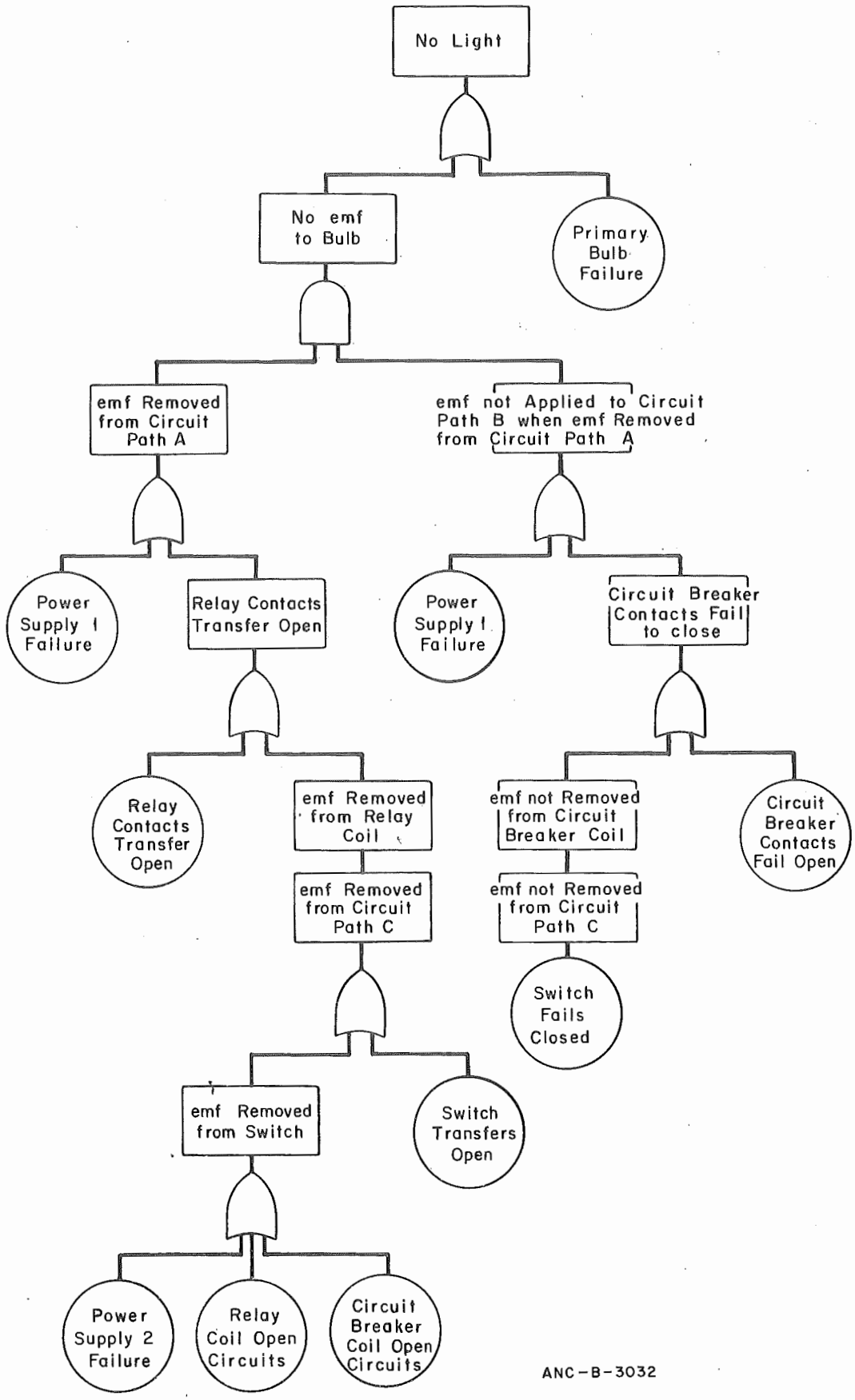


Fig. 10 Fault tree for sample system in Figure 9.

TABLE IV

MINIMAL CUT SETS FOR SAMPLE SYSTEM

(1)	Primary bulb failure
(2)	Primary power Supply 1 failure
(3)	Relay contacts transfer open Circuit breaker contacts fail open
(4)	Relay contacts transfer open Switch fails closed
(5)	Power Supply 2 failure Circuit breaker contacts fail open
(6)	Power Supply 2 failure Switch fails closed
(7)	Relay coil open circuits Circuit breaker contacts fail
(8)	Relay coil open circuits Switch fails closed
(9)	Circuit breaker coil opens circuit Circuit breaker contacts fail open
(10)	Circuit breaker coil opens circuit Switch fails closed
(11)	Switch transfers open Circuit breaker contacts fail open
(12)	Switch transfers open Switch fails closed

2. THE CAUSE OF THE PROBLEM

The reason for these erroneous minimal cut sets is that the fault events "power removed from circuit Path C", hereafter called $\overline{X}$, and the fault event "power not removed from circuit Path C", hereafter called $\overline{Y}$, are mutually exclusive fault events. Consequently, collections of component failures that reflect certain combinations of the primary events used to develop these events will not cause TOP failure. Since these events, $\overline{X}$ and $\overline{Y}$, are

both in the domain of an AND logic gate they are, indeed, combined by existing methodologies of determining the minimal cut sets. This explanation using $\overline{X}$ and $\overline{Y}$ is extended to include any mutually exclusive events X and Y.

3. SOLUTION TO THE PROBLEM

The mutually exclusive events X and Y, that is $X \cap Y = \phi$, both in the domain of an AND logic gate are considered. If x_i and y_i represent the i^{th} set in the complete collection of minimal cut sets for X and Y respectively, where a minimal cut set for a fault event is defined analogously to those of the TOP event, then $x_i \cap y_j = \phi$ for all i and all j since each x_i is simply a reexpression of X and each y_j is simply a reexpression of Y. Next, if Z_k represents the k^{th} set of primary events that results from lumping together all the events in x_i with all the events in y_j for each i with each j over all i and j, then i x j such sets will exist. If M_ℓ is the ℓ^{th} collection of primary events suspected to be a minimal cut set; that is, the output from conventional methods of determining minimal cut sets, then all $M_\ell = 0$ if $M_\ell \supset Z_k$ for all ℓ and k, where $M_\ell \supset Z_k$ means that Z_k is a subset of M_ℓ. Then any collection of primary events that contains any Z_k is not a minimal cut set and should be discarded.

As an illustration Table IV is again considered. This table forms a list M_ℓ sets, M_1 through M_{12}, for the fault tree shown in Figure 10. The x_i and y_i sets for events X and Y are:

x_1 = Power Supply 2 failure

x_2 = Relay coil open circuits

x_3 = Circuit breaker coil open circuits

x_4 = Switch transfers open

y_1 = Switch fails closed.

In general, the x_i's and y_i's are collections of primary events; that is, minimal cut sets for the events X and Y.

Then,

$$Z_1 = \begin{bmatrix} \text{Power Supply 2 failure} \\ \text{Switch fails closed} \end{bmatrix}$$

$$Z_2 = \begin{bmatrix} \text{Relay coil open circuits} \\ \text{Switch fails closed} \end{bmatrix}$$

$$Z_3 = \begin{bmatrix} \text{Circuit breaker coil open circuits} \\ \text{Switch fails closed} \end{bmatrix}$$

$$Z_4 = \begin{bmatrix} \text{Switch transfers open} \\ \text{Switch fails closed.} \end{bmatrix}$$

From Table IV and the Z_i's, sets M_6, M_8, M_{10}, and M_{12} are discarded because $M_6 \supset Z_1$, $M_8 \supset Z_2$, $M_{10} \supset Z_3$, and $M_{12} \supset Z_4$. In general, the M_ℓ sets will be larger than the Z_i sets.

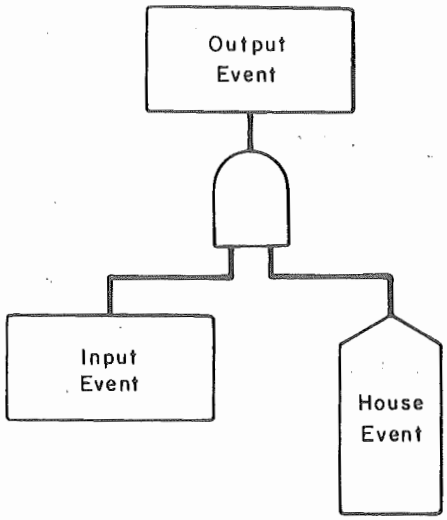
The implementation of this technique is considered to be the best approach to handling mutually exclusive events in the methodology described in Reference 13. For the methodology described in Reference 14 a more straight forward approach is possible. Before the minimal cut sets are determined mutually exclusive events are flagged. These events are then never combined so as to form erroneous minimal cut sets.

In any case, the mutually exclusive fault events must be recognized. Conventional techniques of fault tree construction make such recognition difficult. Again events $\overline{X}$ and $\overline{Y}$ in the previous sample are considered. Neither of these events are needed and would probably not have been included in the conventional fault tree analysis. If events $\overline{X}$ and $\overline{Y}$ were not included, recognizing the events "emf removed from the relay coil" and "emf not removed from the circuit breaker coil" as being mutually exclusive would be difficult. Such difficulty is even more pronounced for complex systems where such events often appear on different pages of the drawn fault tree.

Most often, however, mutually exclusive fault events that are in the domain of the same AND logic gate give rise to at least one minimal cut set that contains mutually exclusive primary events. M_{12} in the previous sample problem is an example. From this clue, the mutually exclusive fault events can be somewhat more easily located.

VIII. THE USE OF HOUSE EVENTS

A house event is an event in a fault tree which is assumed to exist or to not exist for the duration of the situation being analyzed. A house event then has an occurrence probability of zero or unity, at the option of the analyst. The house event can be input to an AND logic gate, as shown in Figure 11, or input to an OR logic gate.



ANC-A-3033

Fig. 11 House event input to an AND logic gate.

If a house event has an occurrence probability of zero and the logic gate the house event is input to is an OR gate, the house event is simply not used in the fault tree. A simple argument to demonstrate that the removal of the house event is valid is to consider two events, A and B, where B is a house event with zero probability of occurrence. Since

$$P(A \cap B) = P(A) + P(B) - P(A)\,P(B|A)$$

and P(B) equals zero,

$$P(A \cap B) = P(A),$$

that is, Event B is unnecessary and can be discarded.

If the house event has zero probability but is input to an AND logic gate the entire AND gate is removed from the fault tree, as are all the immediately preceding AND gates up to the next OR gate. Such removal is warranted because if one of the inputs to the AND gate cannot occur the output event of the AND gate cannot occur; hence no failure through that AND gate can occur. The AND gate is then unnecessary for the fault tree and is removed because it too has zero occurrence probability. The same argument can be extended to all immediately preceding AND gates. A probabilistic argument is demonstrated by again considering two events, A and B, where B is a house event with zero probability of occurrence. Since

$$P(A \cap B) = P(\dot{A})\,P(B|A)$$

and P(B) equals zero [hence, P(B|A)=0],

$$P(A \cap B) = 0.$$

If the house event has an occurrence probability of unity and the logic gate the house event is input to is an AND gate, the fault event is simply not used in the fault tree. This simple removal is possible because an input to an AND gate being "true" makes no contribution to the fault tree. Again if A and B are events and B is a house event with unity occurrence probability,

$$P(A \cap B) = P(A)\, P(B|A)$$
$$= P(A).$$

If, however, a house event with unity occurrence probability is input to an OR logic gate, the entire OR gate is removed from the fault tree, as are all the immediately preceding OR gates up to the next AND gate.

Here

$$P(A \cap B) = P(A) + P(B) - P(A)\, P(B|A)$$

but, if $P(B) = 1$ then $P(B|A) = 1$, then

$$P(A \cap B) = P(A) + 1 - P(A)$$
$$= 1.$$

When a house event is used in conjunction with an AND gate, the house event may be thought of as an enabling input. If the house event occurrence probability is unity, the AND gate is enabled; that is, the other inputs can cause the output event to occur. If the house event occurrence probability is zero, the AND gate is continuously disabled, and none of the events below that gate can cause a system failure. In other words, all of the inputs to the AND gate have no effect on the minimal cut sets of the tree.

The most common use of house events is to attach a house event, that describes an event that is certain to occur, to an AND gate and assign a unity occurrence probability. The house event is then used to show that the event or condition described in the house event symbol has not been neglected but rather is assumed to exist. An example of this situation is given in Figure 12.

If the house event shown in Figure 12 had an assigned occurrence probability of zero, the effect would be to mask out the entire branch of the fault tree. Deleting portions of a fault tree is frequently useful in the study of subsystem importance and design modifications. In these cases the house event may be "artificial" in the sense that no event description is supplied, but rather the house event is simply used as a switch. Also the use of generalized fault trees can be implemented by using house events. For example, if a general fault tree for the hazards of automobile accidents were attempted, during the analysis of low speed front-end collisions, using the

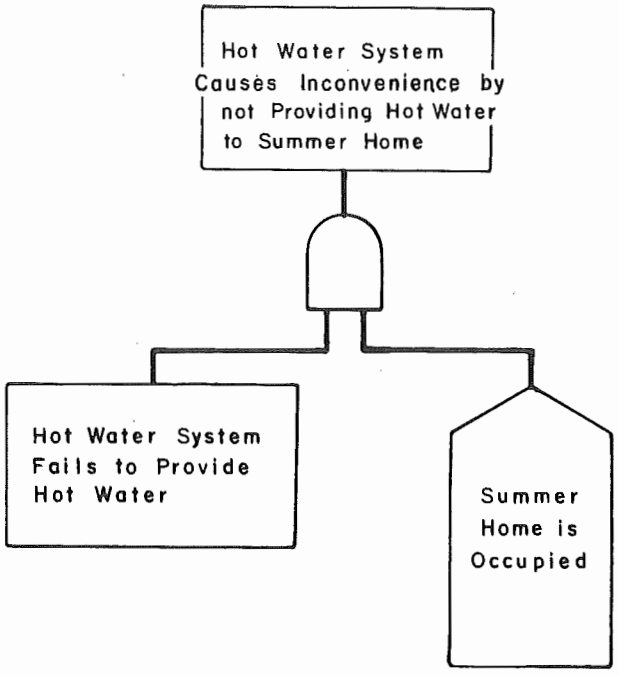


Fig. 12 First example of the use of a house event.

generalized fault tree, gasoline tank explosions might be assumed to be incredible. In which case the house event could be used as shown in Figure 13.

House events input to AND logic gates can also be used to analyze the effect of combinations of failures. In Figure 14 all combinations of three fault events, A, B, and C, are considered. By turning off all house events except one, any failure combination can be analyzed from one fault tree representation.

A more efficient, but unconventional, method for analyzing combinations of failures is to use house events input to OR logic gates. The situation depicted in Figure 14 can be analyzed using the fault tree shown in Figure 15. Here the events to be eliminated from study require the associated house event be assigned a unity occurrence probability. For example, if the combination of Events B and C is to be studied, House Event 1 is assigned an occurrence probability of unity and House Events 2 and 3 are assigned an occurrence probability of zero.

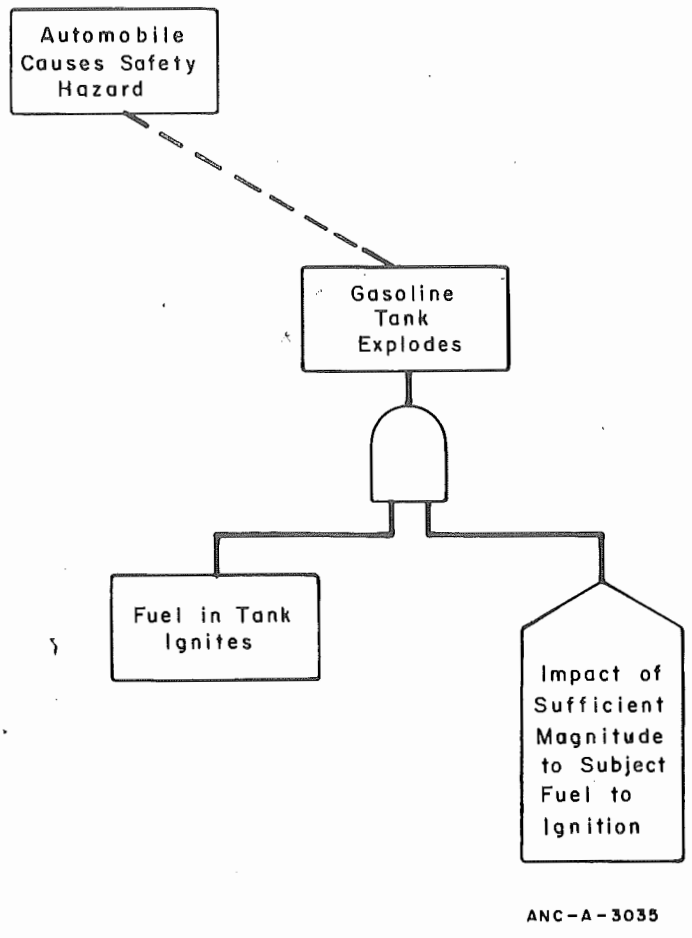


Fig. 13 Second example of the use of a house event.

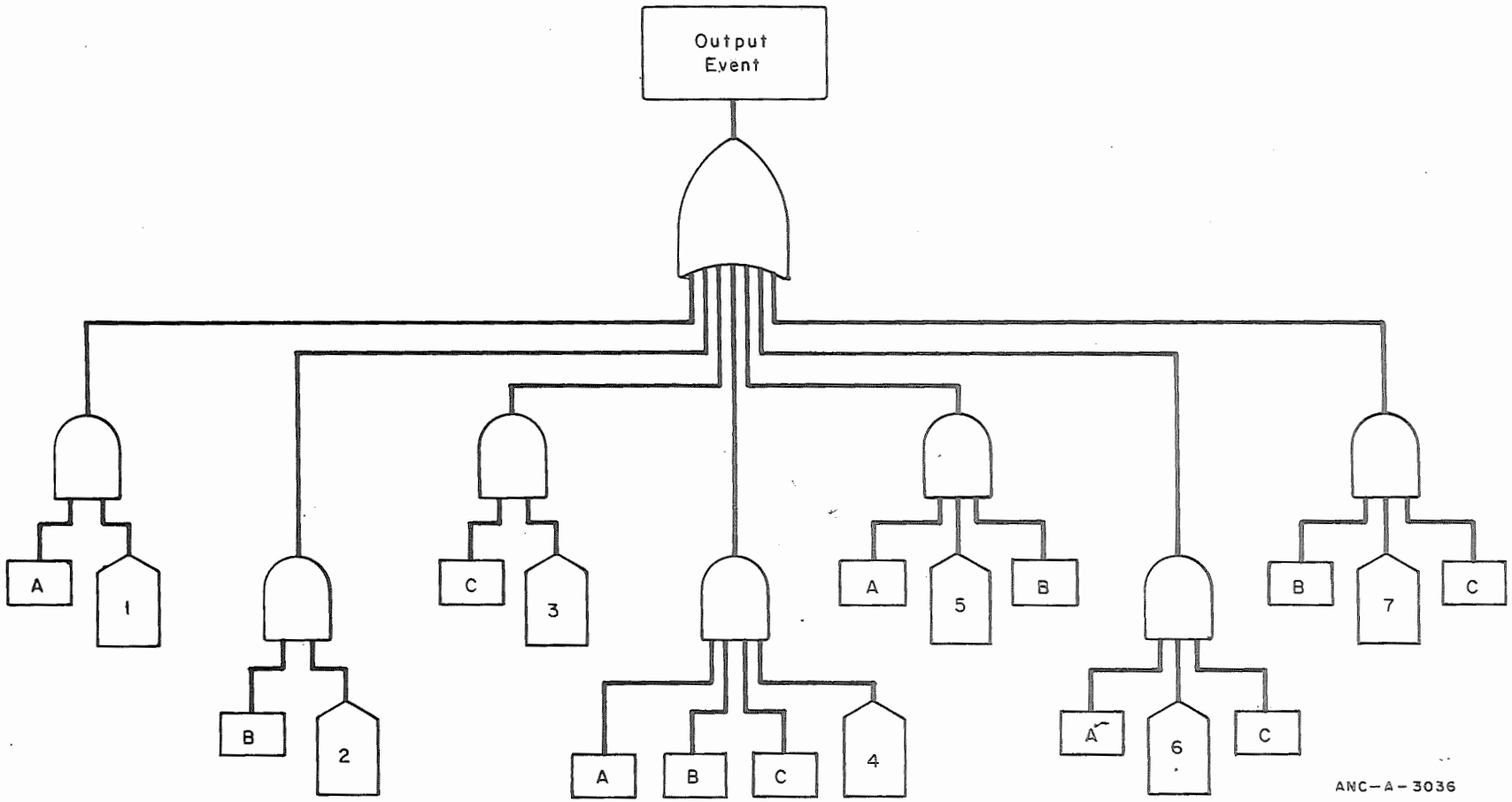


Fig. 14 Third example of the use of house events.

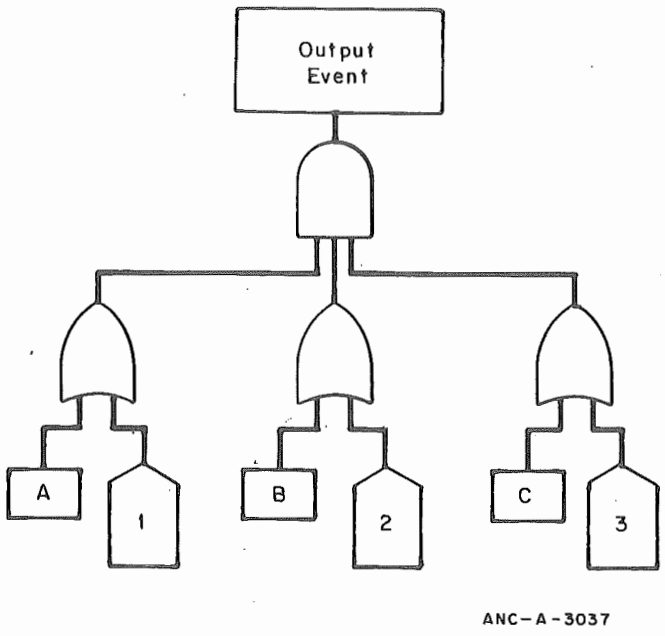


Fig. 15 Streamlined method of handling the situation shown in Figure 14.

IX. PRIORITY AND GATE LOGIC SITUATIONS

A priority AND logic gate is logically equivalent to an AND gate with the additional requirement that the input events must occur in a specific order. Reference 15 presents details on the symbology and quantitative aspects of this logic gate. This presentation is concerned with the situations requiring the priority AND logic gate.

The priority AND logic gate is concerned with sequence of failures. To pass an output, one specific sequence of failures must occur; therefore, the priority AND gate has limited use. Whereas many systems require a specific sequence of series component successes to obtain system success, these systems will generally fail if any component failures occur regardless of the order of failure. A priority AND gate is, therefore, not required. Also, the priority AND gate is not used simply if failures are expected to occur in a specific sequence, but rather is used only if failures must occur in a specific sequence in order to generate the output event.

Priority AND logic situations sometimes occur in switching circuits. As a simple example, the schematic shown in Figure 16 is considered. The power supply periodically, but very rarely, surges in voltage for a short period of time, but always with enough intensity to blow the fuse. The fault tree using the priority AND gate for this system is given in Figure 17 for the following boundary conditions.

TOP Event	Fuse fails open
Initial Conditions	Switch open
Not-Allowed Conditions	(1) Wiring failures
	(2) Failures external to system
	(3) Common mode failures
	(4) More than one power supply surge in one mission time.

The event that must occur first, A, is always listed as the leftmost input event in the fault tree, the event that must occur second is listed immediately to the right of A, ... In the example, as shown in Figure 17, if the secondary fuse failure is to occur, the switch must transfer closed before the power supply surges.

An example of a situation for which the priority AND logic gate is commonly misused is that of a standby system used with a principal system. The principal system is normally operating. Although the principal unit may be expected to fail before the standby unit, the system failure will occur so long as both units fail regardless of the order of the failures. A priority AND gate is not used to describe this situation.

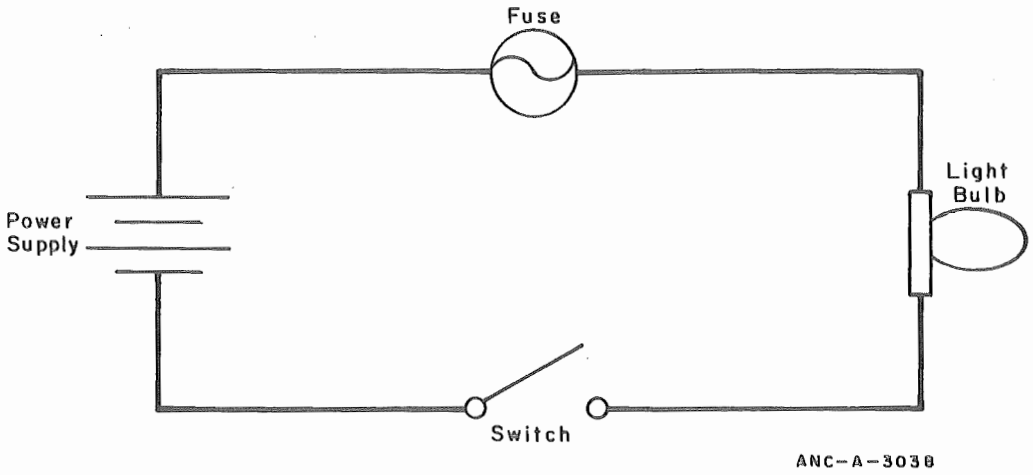


Fig. 16 System schematic for priority AND logic examples.

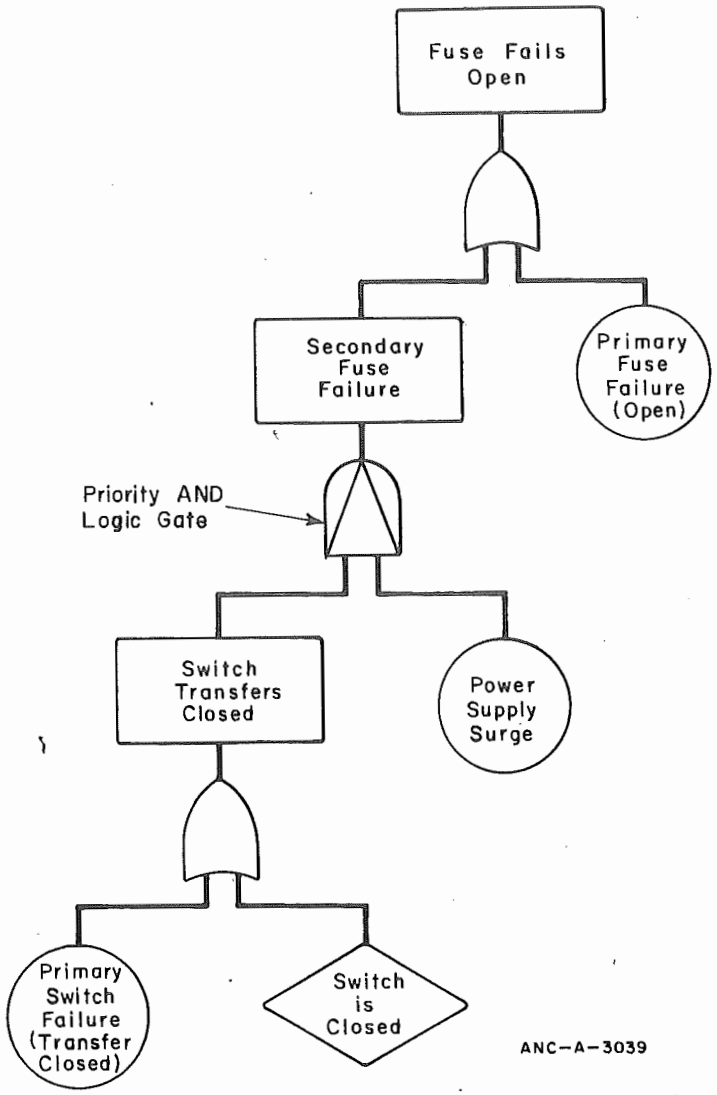


Fig. 17 Fault tree for priority AND logic example.

X. REFERENCES

1. W. E. Vesely, "A Time-Dependent Methodology for Fault Tree Evaluation", *Nuclear Engineering and Design,13(2)* (August 1970) pp 337-360.

2. J. B. Fussell, *Synthetic Tree Model - A Formal Methodology for Fault Tree Construction,* ANCR-1098 (March 1973).

3. B. J. Garrick, "Principles of Unified Systems Safety Analysis", *Nuclear Engineering and Design 13* (August 1970) pp 245-321.

4. D. F. Haasl, "Advanced Concepts in Fault Tree Analysis", *System Safety Symposium,* June 8-9, 1965, Seattle: The Boeing Company. Available from the University of Washington Library, Seattle, Washington.

5. H. E. Lambert, *System Safety Analysis and Fault Tree Analysis,* UCID-16238, May 9, 1973. Available from TID Library, Lawrence Livermore Laboratories, Livermore, California 94550.

6. J. B. Fussell, "Fault Tree Analysis - Concepts and Techniques", *NATO Advanced Study Institute on Generic Techniques of System Reliability Assessment,* to be published by Nordhoff Publishing Company.

7. "Reliability Stress and Failure Rate Data for Electronic Equipment", *Military Standardization Handbook,* MIL-HDBK-217A. Available from the Chief Bureau of Naval Weapons, Department of the Navy, Washington, D.C. 20360.

8. *Failure Rate Data Handbook* (FARADA), U. S. Naval Fleet Missile Systems Analysis and Evaluation Document No. SP 63-470. Available from Naval Weapons Station, Seal Beach, Corona, California 91720.

9. R. E. Barlow and P. Chatterjee, *Introduction to Fault Tree Analysis,* University of California, Berkeley, ORC 73-30, p 24, December 1973. Available from Operations Research Center, University of California, Berkeley, California 94720.

10. A. E. Green and A. J. Bourne, *Reliability Technology,* John Wiley and Sons Ltd., London, Ch. 11 and 12, 1972.

11. J. D. Esary and F. Proschan, "A Reliability Bound for Systems of Maintained, Interdependent Components", *Journal of the American Statistical Association, 65,* No. 329 (March 1970) pp 329-338.

12. S. M. Ross, *Multicomponent Reliability Systems,* University of California, Berkeley, ORC 74-4, February 1974. Available from Operations Research Center, University of California, Berkeley, California 94720.

13. W. E. Vesely and R. E. Narum, *PREP and KITT: Computer Codes for the Automatic Evaluation of a Fault Tree,* IN-1349 (August 1970).

14. J. B. Fussell and W. E. Vesely, "A New Methodology for Obtaining Cut Sets for Fault Trees", *Transactions of the American Nuclear Society, 15(1)* (1972).

15. J. B. Fussell, *Special Techniques for Fault Tree Analysis,* Aerojet Nuclear Report for Automation Industries (March 1974).

APPENDIX

ARGUMENT FOR METHOD OF TREATING

CIRCULAR FAILURE LOGIC

The justification for the method of treating circular failure logic that is presented is based on showing that the correct minimal cut sets result.

First the case of the second occurrence of an event being attached to an OR gate is considered. Figure A-1 describes the situation. The Boolean indicated cut sets as obtained by the MOCUS algorithm[A-1] are as follows:

(1) A B

(2) A C D

(3) A C A B

(4) A C A C D

(5) A C A C A B

(6) A C A C A C D

(7) A C A C A C A B

(8) A C A C A C A C D

.
.
.

(n-1) A C A C . . . . A B

(n) A C A C . . . . A C D

.
.
.

Boolean indicated cut set n contains n + 1 events where n is an even integer. When super sets are eliminated, the minimal cut sets remain and are as follows:

(1) A B

(2) A C D.

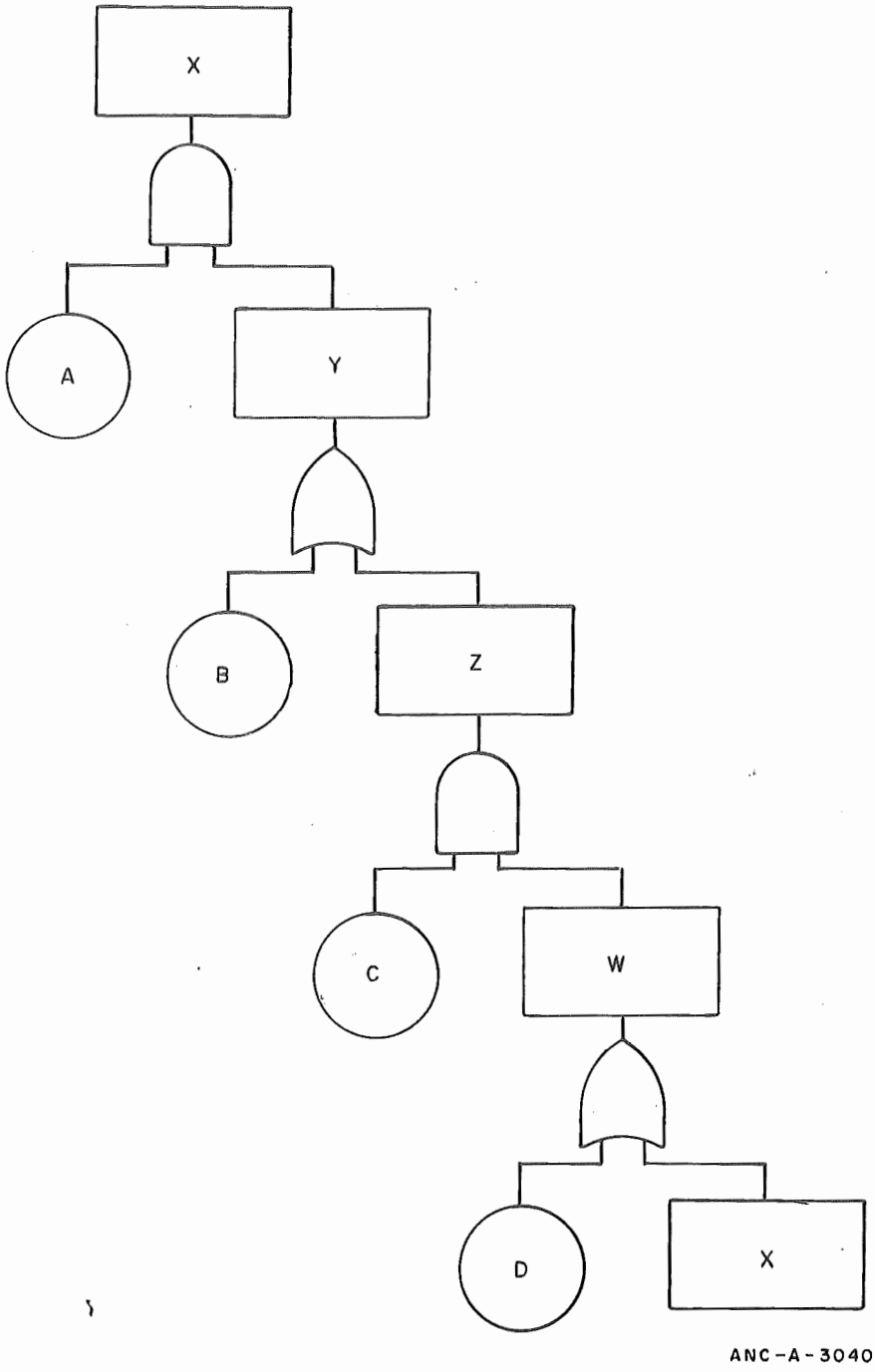


Fig. A-1 Fault tree in which second occurrence of an event is attached to an OR gate.

If the second occurrence of Event x is removed from the fault tree in Figure A-1 the Boolean indicated cut sets (and the minimal cut sets) are immediately:

(1) A B

(2) A C D.

Since the minimal cut sets are identical, a fault tree with circular failure logic is equivalent to the fault tree resulting when the second occurrence of an event is removed from the fault tree, if this second occurrence is attached to an OR gate.

Next the case of the second occurrence of an event being attached to an AND gate is considered. Figure A-2 describes the situation. The Boolean indicated cut sets as obtained by the MOCUS algorithm are as follows:

(1) A

(2) B C

(3) B D E A

(4) B D E B C

(5) B D E B D E A

(6) B D E B D E B C

(7) B D E B D E B D E A

(8) B D E B D E B D E B C

.
.
.

(n-1) B D E B D E . . . . . A

(n) B D E B D E . . . . . B C

.
.
.

Boolean indicated cut set n contains $\frac{3}{2}$n - 1 events where n is an even integer. When super sets are eliminated the minimal cut sets remain and are as follows:

(1) A

(2) B C.

If the AND gate, to which the second occurrence of Event x is attached, and all immediately preceding AND gates up to the next OR gate are removed from the fault tree, the Boolean indicated cut sets (and the minimal cut sets) are immediately:

Since the minimal cut sets are identical, a fault tree with circular failure logic is equivalent to the fault tree resulting when the second occurrence of an event is removed from the fault tree, if this second occurrence is attached to an OR gate.

Next the case of the second occurrence of an event being attached to an AND gate is considered. Figure A-2 describes the situation. The Boolean indicated cut sets as obtained by the MOCUS algorithm are as follows:

(1) A

(2) B C

(3) B D E A

(4) B D E B C

(5) B D E B D E A

(6) B D E B D E B C

(7) B D E B D E B D E A

(8) B D E B D E B D E B C

.
.
.

(n-1) B D E B D E . . . . . A

(n) B D E B D E . . . . . B C

.
.
.

Boolean indicated cut set n contains $\frac{3}{2}$n - 1 events where n is an even integer. When super sets are eliminated the minimal cut sets remain and are as follows:

(1) A

(2) B C.

If the AND gate, to which the second occurrence of Event x is attached, and all immediately preceding AND gates up to the next OR gate are removed from the fault tree, the Boolean indicated cut sets (and the minimal cut sets) are immediately:

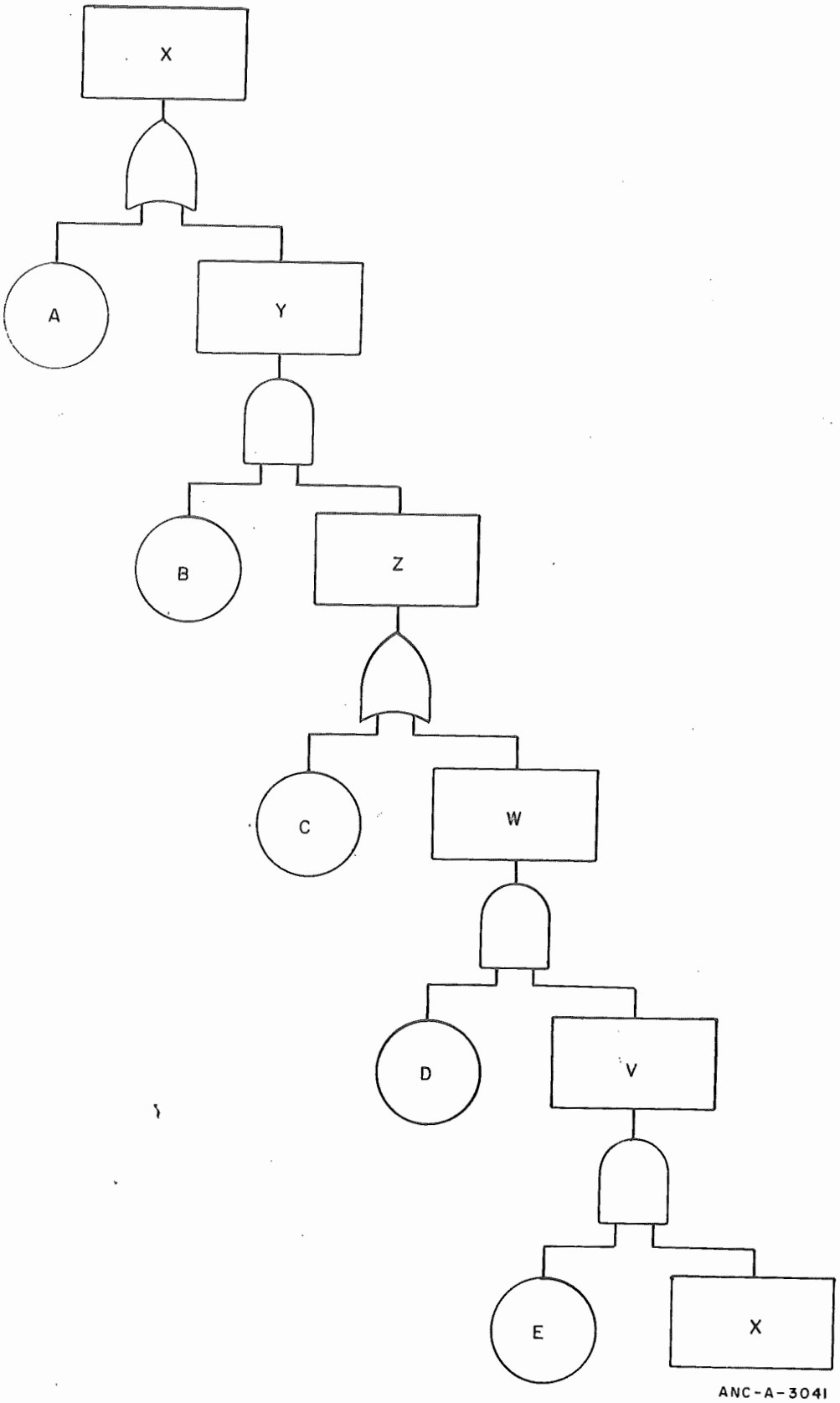


Fig. A-2 Fault tree in which second occurrence of an event is attached to an AND gate.

(1) A

(2) B C .

Since the minimal cut sets are identical, the procedure is validated.

Reference

A-1. J. B. Fussell, E. B. Henry, N. H. Marshall, *MOCUS -- A Computer Program to Obtain Minimal Sets from Fault Trees,* ANCR-1156 (August 1974).