ADVANCED CONCEPTS IN FAULT TREE ANALYSIS

BY

DAVID F, HAASL SYSTEM SAFETY ENGINEER MISSILE BRANCH, AERO-SPACE
DIVISION THE BOEING COMPANY, SEATTLE, WASHINGTON

PRESENTED AT SYSTEM SAFETY SYMPOSIUM SPONSORED BY UNIVERSITY OF
WASHINGTON AND THE BOEING COMPANY JUNE 8-9, 1965

OLYMPIC HOTEL SEATTLE, WASHINGTON

BIOGRAPHY

David F. Haasl graduated cum laude from Gonzaga University in 1961, with a B.S. in chemical engineering. His program of study included a major in applied mathematics with emphasis on computer applications. Following graduation he was employed by The Boeing Company and is currently assigned to the System Safety organization of the Minuteman project. Mr. Haasl was responsible for developing the advanced techniques now used in Boeing’s fault tree analysis programs.

ADVANCED CONCEPTS
IN FAULT TREE
ANALYSIS

The singular success of the Minuteman Safety Program is attributable to two separate, but complementary, innovations. The first consisted of the establishment of Safety as an autonomous discipline in System Engineering. This management action augmented the traditional role of Safety with an engineering responsibility in all phases of system design, development, and operation. The second factor was the introduction and evolution of Fault Tree Analysis as the principal analytical tool of the System Safety Engineer. A Fault Tree provides a concise and orderly description of the various combinations of possible occurrences within a system which can result in a pre-defined "undesired event". Equally as important, it makes available a means with which to measure the level of Safety inherent in any particular configuration. The result is an engineering capability to not only identify potential problem areas but also evaluate their over-all system impact. Experience has demonstrated that appropriate utilization of Fault Tree Analysis, within a technically qualified safety engineering organization, is an effective approach to the problem of achieving and maintaining system safety design criteria.

The concept of Fault Tree Analysis was originated by Bell Telephone Laboratories as a technique with which to perform a safety evaluation of the Minuteman Launch Control System. Bell engineers discovered that the method used to describe the flow of "correct" logic in data processing equipment could also be used for analyzing the "false" logic which results from component failures. Further, such a format was ideally suited to the application of probability theory in order to numerically define the critical fault modes. The Minuteman Safety Study was successfully completed using the new technique, and provided convincing arguments for the incorporation of a number of equipment and procedure modifications.

Significant refinement of the analytical and mathematical techniques used in Fault Tree Analysis has taken place since the concept was first introduced in 1961. The evolution has been natural; resulting from the application of Fault Tree Analysis to many different kinds of systems. Two techniques for fault tree construction and two methods for probability evaluation are described on the following pages. In each case, the second process described represents a later stage of development and permits the most realistic analysis. Due to their simplicity, however, the earlier techniques are still useful. It is unrealistic to apply a rigorous technique to an analytical situation requiring only an approximate solution. Similarly, it is imprudent to be satisfied with approximations when only an exact answer will suffice. The selection of methods for a particular system analysis is dependent on such things as; the complexity of the system, the likelihood of the "undesired event", the degree of system design completion, the availability of time and funding, and the capabilities of the analysts. Systems and circumstances are too variable to permit the development of a stereotype fault tree analysis program.

FAULT TREE CONSTRUCTION: GENERAL

The two techniques for fault tree construction are best described through the use of an illustrative example. Accordingly, the "system" depicted in Figure One prepared to serve as the basis for a comparison of the two techniques.

Suppose that prior analyses have indicated that destruction of the wire between A and B from over-heating is a critical event. Perhaps this particular wire is proximate to ordnance wiring to which it may short circuit; or maybe it passes through an area that is frequently saturated with combustible vapors. In any event, a fault tree must be constructed to define the failure modes by which the wire between A and B can be over-heated.

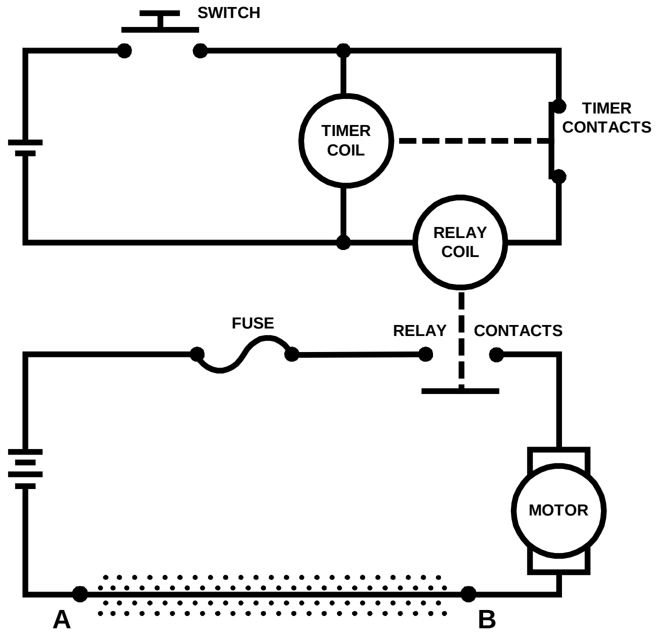

FIGURE 1: SAMPLE SYSTEM

Before attempting any analysis, it is necessary to learn how the system functions. The sample system is designed to make available mechanical energy from the MOTOR whenever the SWITCH is closed by the action of an external control system. When the SWITCH is closed, power is applied to the RELAY COIL through the TIMER CONTACTS. With power on the RELAY COIL, the RELAY CONTACTS close and cause power to be applied through the FUSE to the MOTOR. When the SWITCH is later opened, power is removed from the RELAY COIL, thereby opening the RELAY CONTACTS and removing power from the MOTOR. The TIMER and FUSE are safety devices. If the SWITCH fails to open after some pre-set time interval, the TIMER CONTACTS should open and remove power from the RELAY COIL. If the

MOTOR fails shorted while the RELAY CONTACTS are closed, the FUSE should open and de-energize the circuit.

Before proceeding to the definition of failure modes, the appropriate logic symbols must be defined. Since time is limited, only those symbols required for the illustrative example are explained. The "AND" GATE describes the logical operation whereby the coexistence of all input events are required to produce the output event. The "OR" GATE defines the situation whereby the output event will exist if at least one of the input events is present. There are no restrictions on the number of inputs to either gate. It is also necessary to define symbols with which to represent the fault events: The rectangle is used to describe fault events that result from the combination of more basic faults acting through the logic gates.

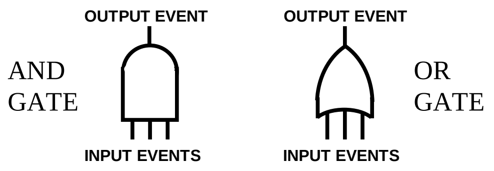

Figure 2: LOGIC GATES

The circle describes basic fault inputs that require no further development. (This category includes component. failures whose frequency and mode of failure are derived through laboratory testing.) The diamond describes fault inputs that are considered basic in a given fault tree. However, the event described is not basic in the sense that laboratory data is applicable. Rather, the fault tree is simply not developed further, either because the event is of insufficient consequence or the necessary information is unavailable.

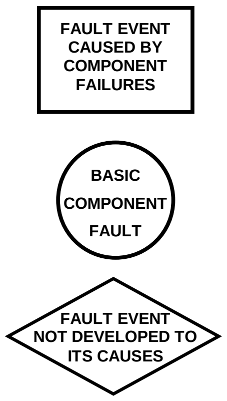
FAULT EVENT CAUSED BY COMPONENT FAILURES
BASIC COMPONENT FAULT
FAULT EVENT NOT DEVELOPED TO ITS CAUSES

Figure 3: FAULT EVENTS

The actual preparation of a fault tree begins with a definition of the final "undesired event" and proceeds with a series of "a posteriors" judgments until basic input events are defined. In the sample problem, overheating of the wire between A and B can result only from the application of current beyond the rated capacity of the wire for an extended period of time. The coexistence of both excessive current and an "over-run" condition are essential to produce the "undesired event". Using the AND gate this is represented in Figure Four.

Next, the combination of events that can produce EXCESSIVE CURRENT IN SYSTEM WIRING and POWER APPLIED TO SYSTEM FOR EXTENDED TIME are developed. (For simplicity of illustration, the two power supplies are assumed to contribute nothing to the critical system failure modes.) Considering each in turn: EXCESSIVE

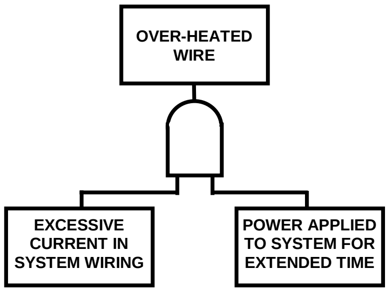
OVER-HEATED WIRE
EXCESSIVE CURRENT IN SYSTEM WIRING
POWER APPLIED TO SYSTEM FOR EXTENDED TIME

Figure 4

CURRENT IN SYSTEM WIRING can occur only if the MOTOR f and the FUSE is unable to open. Therefore, another AND gate is required to combine the two inputs, MOTOR FAILED SHORTED and FUSE UNABLE TO OPEN, and relate them to the O HEATED WIRE. POWER APPLIED TO SYSTEM FOR EXTENDED TIME can occur only if the RELAY CONTACTS remain closed after system operation. There are two possible causes of this event: Either the RELAY CONTACTS themselves can fail closed, or the SWITCH and TIMER can both fail in such a manner that power is not removed from the RELAY COIL. Both an OR gate and an AND gate are required to relate these possibilities. Figure Five portrays the status of the fault tree at

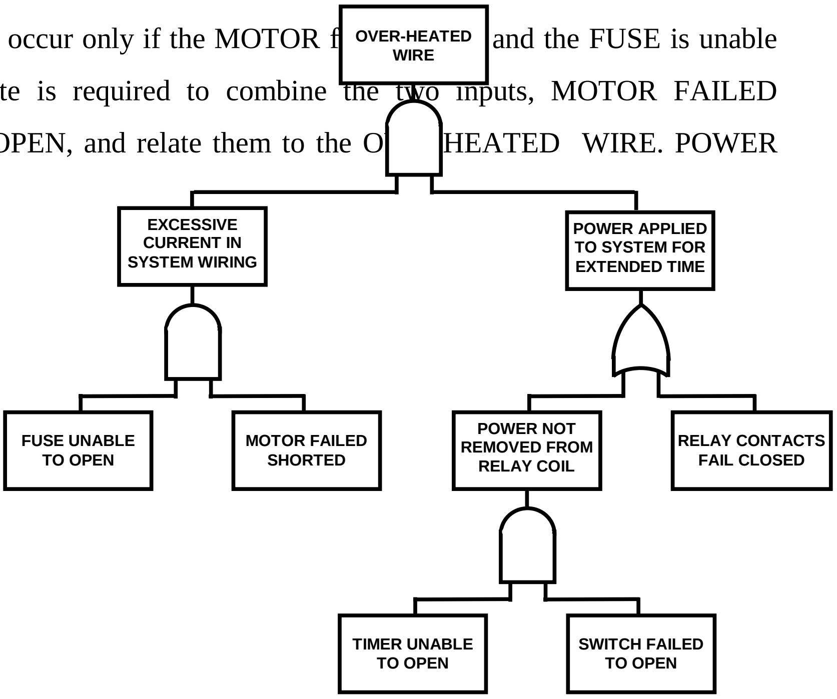
OVER-HEATED WIRE
EXCESSIVE CURRENT IN SYSTEM WIRING
POWER APPLIED TO SYSTEM FOR EXTENDED TIME
FUSE UNABLE TO OPEN
MOTOR FAILED SHORTED
POWER NOT REMOVED FROM RELAY COIL
RELAY CONTACTS FAIL CLOSED
TIMER UNABLE TO OPEN
SWITCH FAILED TO OPEN

Figure 5

It should be noted that both of the AND gates, which have been added, require a sequential occurrence of input events. The FUSE must fail prior to the MOTOR; likewise the TIMER failure must precede that of the SWITCH. If the converse happens in either case, the system will fail in a mode unrelated to the fault tree output.

The fault tree has now been developed to a point where the analyst must choose between the two techniques of analysis. He may decide to consider only primary component failures or he may wish to include secondary failures as well. The failure of a component is called “primary” if it occurs while the part is functioning within the operating parameters for which it was designed. It is termed “secondary” if the failure occurs when the component is subjected to abnormal environmental stresses such as failures in related equipment. Since both analytical methods are to be described, they will be differentiated by the terms, “Primary Failure Technique” and “Secondary Failure Technique”.

FAULT TREE CONSTRUCTION: PRIMARY FAILURE TECHNIQUE

Construction of a fault tree, using the Primary Failure Technique, is a relatively straightforward process. The tree must be developed only to the point where identifiable primary component failures will directly produce the required fault events. The technique is used principally in fault tree analyses dealing with communication and data processing systems.

Continuing the development of the fault tree shown in Figure Five using the Primary Failure Technique; it is found that the input, FUSE UNABLE TO OPEN, can be caused by one of two things either a primary failure of the FUSE or the installation of an over-sized device. (Note - The failure of the FUSE is a basic fault input and is represented by a circle. The possibilities of installing an over-sized fuse are not easily defined, so the input is represented by a diamond.) The faults, MOTOR FAILED SHORTED and RELAY CONTACTS FAILED CLOSED, describe the primary failures of single components, and are shown as circles. There are two possible causes of TIMER UNABLE TO OPEN, both of them primary failure modes of the timer. The TIMER COIL can fail open so that it is never energized or the TIMER CONTACTS can fail closed. There are also two faults which can cause SWITCH FAILS TO OPEN. The first is the primary failure of the SWITCH CONTACTS. The second is a failure of the external control system to release the switch. (Since no information is available concerning the external control system its failure must be pictured as a diamond.) The final version of the fault tree developed according to the Primary Failure Technique is shown in Figure Six.

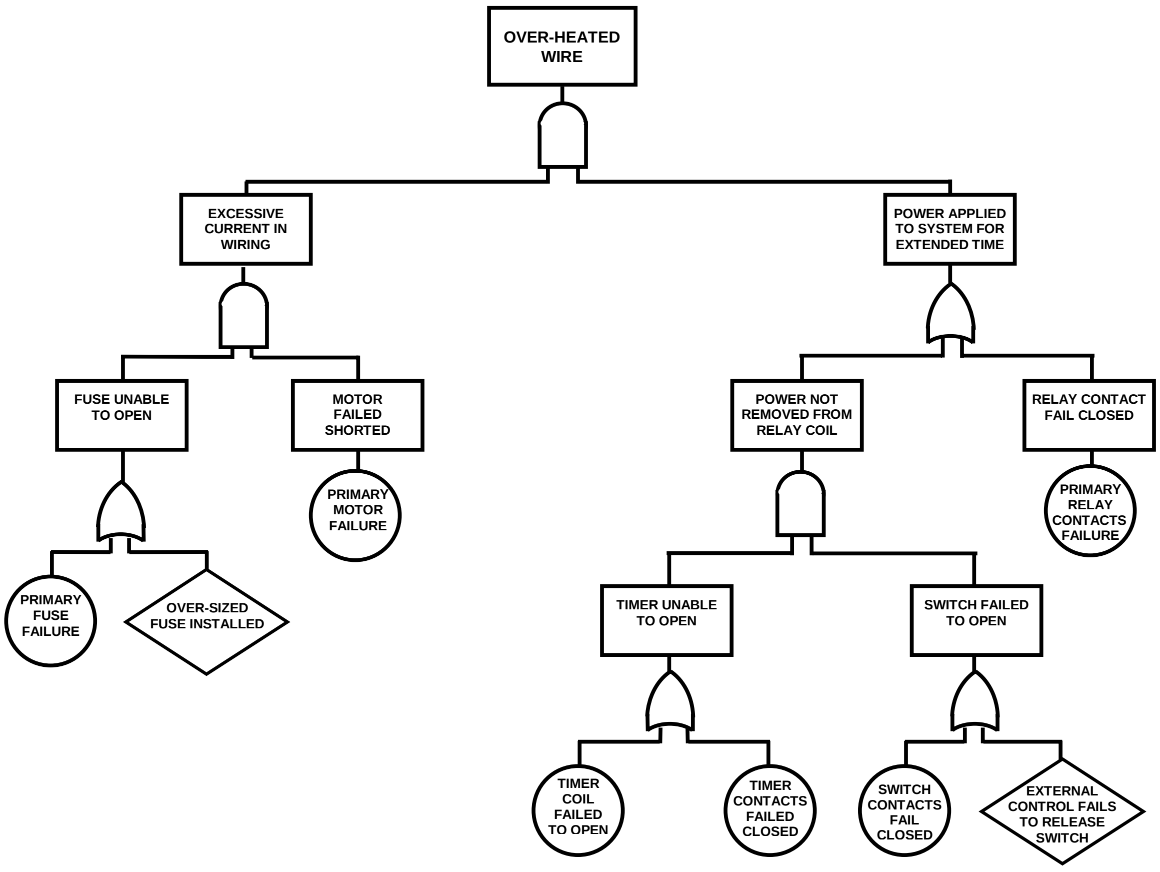
OVER-HEATED WIRE
EXCESSIVE CURRENT IN WIRING
POWER APPLIED TO SYSTEM FOR EXTENDED TIME
FUSE UNABLE TO OPEN
MOTOR FAILED SHORTED
POWER NOT REMOVED FROM RELAY COIL
RELAY CONTACT FAIL CLOSED
PRIMARY MOTOR FAILURE
PRIMARY RELAY CONTACTS FAILURE
PRIMARY FUSE FAILURE
OVER-SIZED FUSE INSTALLED
TIMER UNABLE TO OPEN
SWITCH FAILED TO OPEN
TIMER COIL FAILED TO OPEN
TIMER CONTACTS FAILED CLOSED
SWITCH CONTACTS FAIL CLOSED
EXTERNAL CONTROL FAILS TO RELEASE SWITCH

FIGURE 6: FAULT TREE WITH PRIMARY FAILURE TECHNIQUE

FAULT TREE CONSTRUCTION: SECONDARY FAILURE TECHNIQUE

Development of a fault tree according to the Secondary Failure Technique requires greater insight to the system. The analysis does not stop when it reaches the component level. It continues until the affect on each component, of the possible failure of all related components, has been portrayed. In a large system the magnitude of task assumes phenomenal proportions; nonetheless, the resultant definition of system failure modes invariably justifies the effort.

Construction of the fault tree, to the extent shown in Figure Five, is the same regardless of which technique is used. The analysis has been conducted to a point where the critical component failures are identified. It is in the development of the causes of component failure that the two techniques differ. With the Primary Failure Technique, the failure of one component is presumed to be unrelated to the failure of any other component. With the Secondary Failure Technique, all significant fault inter-relationships must be developed. Referring to the five component failures described in Figure Five it is found that three are adequately defined as primary failures. The FUSE, TIMER, and SWITCH will each be unaffected by the failure of any or all of the other components. (For simplicity of description, the degradation in system reliability due to component drifting-out-of tolerance has been ignored.) The MOTOR and RELAY are, however, sensitive to the failure of one another. The most likely cause of RELAY CONTACTS FAILED CLOSED is excessive current, resulting from the MOTOR failing shorted, being conducted through the contacts. Similarly, MOTOR FAILED SHORTED is most apt to occur if the RELAY CONTACTS remain closed and allow the MOTOR to run for an extended time.

To describe secondary failures of this type, it is convenient to define another category of AND gate. Depicted in Figure Seven, the INHIBIT gate describes a causal relationship between one fault and another. The input fault directly produces the output fault if the indicated condition is satisfied. The conditional input defines a state of the system that permits the fault sequence to occur, and may be either normal to the system or be the result of equipment failures.

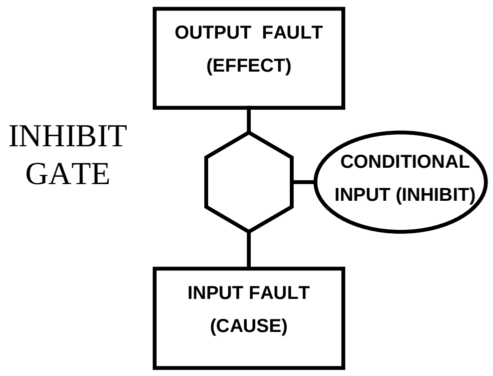

Figure 7

Considering the occurrence of MOTOR FAILED SHORTED and RELAY CONTACTS FAILED CLOSED as both primary and secondary failures the fault tree is completed in Figure Eight.

PROBABILITY EVALUATION: GENERAL

The second facet of Fault Tree Analysis involves the determination of probability values. The object is to establish the likelihood of occurrence of the “undesired event” and to evaluate the relative contribution of each indicated failure mode. With this information the safety analyst can identify the critical system failure modes and decide whether or not corrective action is warranted.

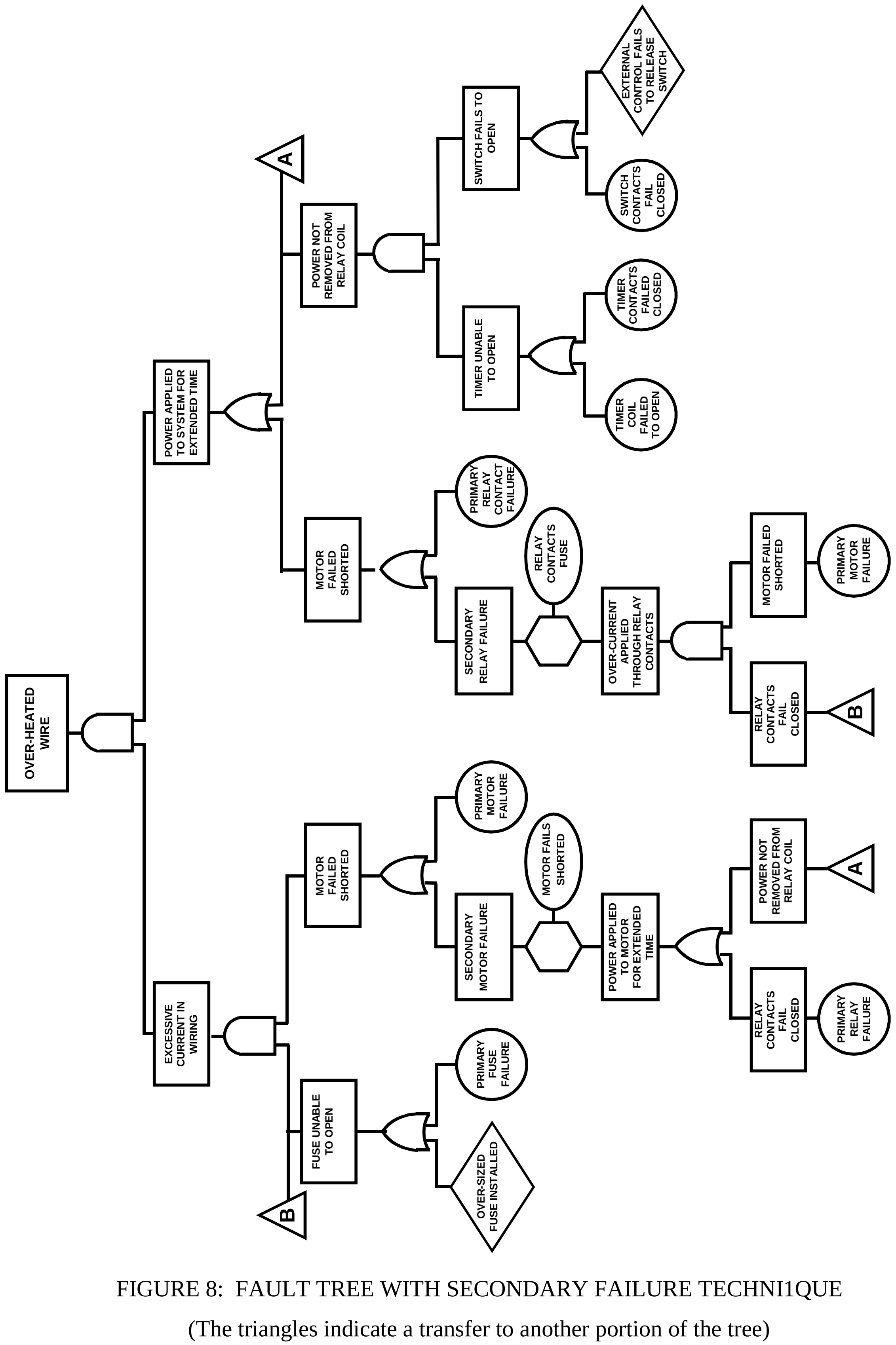
OVER-HEATED WIRE
POWER APPLIED TO SYSTEM FOR EXTENDED TIME
EXCESSIVE CURRENT IN WIRING
A
POWER NOT REMOVED FROM RELAY COIL
SWITCH FAILS TO OPEN
EXTERNAL CONTROL FAILS TO RELEASE SWITCH
SWITCH CONTACTS FAIL CLOSED
TIMER UNABLE TO OPEN
TIMER CONTACTS FAILED CLOSED
TIMER COIL FAILED TO OPEN
MOTOR FAILED SHORTED
PRIMARY RELAY CONTACT FAILURE
SECONDARY RELAY FAILURE
RELAY CONTACTS FUSE
OVER-CURRENT APPLIED THROUGH RELAY CONTACTS
MOTOR FAILED SHORTED
PRIMARY MOTOR FAILURE
RELAY CONTACTS FAIL CLOSED
B
MOTOR FAILED SHORTED
PRIMARY MOTOR FAILURE
SECONDARY MOTOR FAILURE
MOTOR FAILS SHORTED
POWER APPLIED TO MOTOR FOR EXTENDED TIME
POWER NOT REMOVED FROM RELAY COIL
A
RELAY CONTACTS FAIL CLOSED
PRIMARY RELAY FAILURE
FUSE UNABLE TO OPEN
PRIMARY FUSE FAILURE
OVER-SIZED FUSE INSTALLED
B

FIGURE 8: FAULT TREE WITH SECONDARY FAILURE TECHNI1QUE

(The triangles indicate a transfer to another portion of the tree)

There are, in effect, two ways in which the probability of an event may be determined. To illustrate, consider the methods that could be used to establish the probability of rolling a “seven” in one cast of a pair of dice: The first method requires an examination of each die to find the number of different possibilities. This is followed by the preparation of a list of combinations totalling seven If the dice are cubes and numbered in the usual manner the combinations, 1:6, *2:5,* 3:4,4:3, *5:2,* and 6:1, satisfy the requirement. The probability of each combination is then computed and the resultant values summed to determine the overall probability.* The second method is to record the results of a large number of casts and compare the number of ”sevens” with the total number of combinations recorded. If the number of casts is sufficiently large, the ratio of the two values will closely approach the correct probability value.

Both approaches are used in the probability evaluation of fault trees. The principles illustrated in the first method form the basis for a number of computational schemes. To date, however, none have been devised that can be applied without one or more simplifying assumptions. The simulation approach described as the second method will provide a true answer in all cases, but its use is contingent on the availability of extensive computer facilities.

PROBABILITY EVALUATION: COMPUTATION

A fault tree is essentially the representation of events in a symbolic logic format. This being the case, Boolean algebra may be used to symbolically express the fault tree in terms of its basic inputs. The algebraic expression can then be simplified to a non-redundant form suitable for the combination of probability values. For example, the logic tree shown in Figure Nine is algebraically defined as F =A. B + C where F is the output event and A, B, and C are the inputs. The dot represents the “AND” combination and the “plus-sign” signifies “OR”. Probabilities are then computed for the input events and can be combined to determine the probability of the output event.

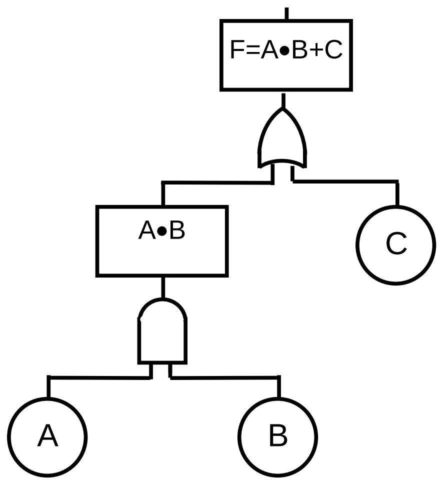

Figure 9

*$P_{1:6} = P_{2:5} = P_{3:4} = P_{4:3} + P_{5:2} = P_{6:1} = 1/6 \times 1/6 = 1/36$
PTOTAL= 1/36 + 1/36 +1/36 + 1/36 + 1/36 + 1/36 = 6/36 = 1/6

In the dice example, all that had to be computed was the probability of occurrence of each combination. Since the various combinations were mutually exclusive and there was only one trial, determination of the over-all probability was simply a matter of addition. The probability evaluation of a fault tree presents a more difficult situation. It is not sufficient to merely compute the probability of occurrence of a given event. If a failure does occur, it will remain in existence for some length of time. Further, a given component may fail and be repaired a number of times during the life of a system. A component with a very low failure rate and an extremely long “time-to-repair” can be more significant from a systems standpoint than one which fails frequently and is repaired immediately. Another analytical complication is introduced when basic fault events combine to produce a secondary failure which has some new “time-to-repair”. Several attempts have been made to devise general probability integration schemes to handle the “repair problem”. To date, no satisfactory solution has been found.

Lacking an exact solution, approximations must be made. The most straightforward approach is to assume that failures are not repaired. This assumption results in probability values that are extremely conservative. While a conservative approximation of the true value is satisfactory for very remote probabilities, it is decidedly unrealistic if the probability is such that the event may be expected to actually occur. A more useful approach was developed by Boeing in 1963, and later incorporated, with a Boolean reduction program, into an IBM 7094 computer routine. In effect, the computer considers each logic gate and computes a probability of occurrence and a “time-to-repair” for the output event as a function of the input events. It is limited, however to relatively small fault trees - two hundred inputs or less - and is based on the assertion of constant failure rate* and constant “time-to-repair”. If these conditions are acceptable a very close approximation of the correct probability value is achieved.

PROBABILITY EVALUATION: SIMULATION

Since all available computational techniques require approximations, the only way to avoid approximations is to avoid computation. This can be accomplished through’ fault tree simulation. A simulation provides the means to “cast the dice” the necee number of times to permit the comparison of “sevens” with the total number of trials. This is done by actually simulating the various logic gates and input events within a

* A constant failure rate is characteristic of the exponential distribution, i.e., the failures occur in the “flat” portion of the reliability “bath tub” curve.

computer. The simulated functions physically perform the operations of their mathematical counterparts. Such a system may be shown as follows:

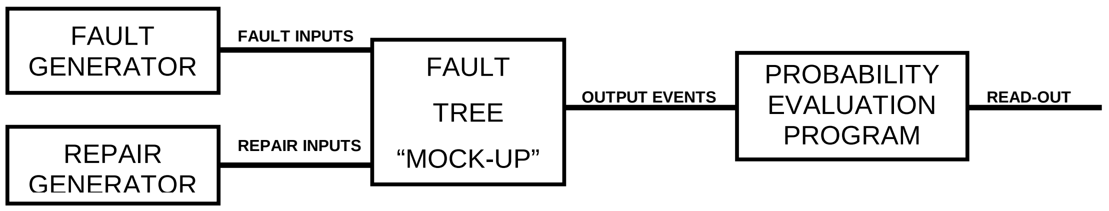

FIGURE 10: FAULT TREE SIMULATION

Fault events are initiated within a FAULT GENERATOR. This equipment provides outputs representing the occurrence of each independent input to the fault tree. These outputs are randomly generated in accordance with a probability and a distribution function that is characteristic of each input. Whenever a fault input occurs, a variable "time-to-repair" typical of that particular fault is developed within the REPAIR GENERATOR. It is then "attached" to the fault input in order to specify its duration.

Within the FAULT TREE "MOCK-UP" the fault input initiates appropriate fault sequences as described by the fault tree. As the sequence continues through the paths of the fault tree, new repair times designed to simulate the duration of secondary failures are extracted from the REPAIR GENERATOR. In this manner the chain of faults continues until it is either repaired or produces an output. The outputs of the FAULT TREE "MOCK-UP" are accumulated by the PROBABILITY EVALUATION PROGRAM, and evaluated to determine the probability values associated with the fault tree.

Fault tree simulation avoids the problems associated with computational techniques. If the "time-to-repair" of an input must be variable, a repair distribution is inserted in the REPAIR GENERATOR. If the probability of occurrence of fault input is definable only by some unusual distribution, the distribution is placed in the FAULT GENERATOR. If requirements regarding the sequencing of faults at a logic gate are needed, they are defined in the FAULT TREE "MOCK-UP". Theoretically at least, fault tree simulation is the ideal probability evaluation technique. The drawback in the use of simulation techniques has been the amount of computer time involved. Probability values to be demonstrated may vary from nearly unity to 1010 and beyond. As the probability number decreases, an ever-increasing number of trials must be conducted to obtain statistical significance. For example, the verification of a probability of l0~ per trial, at a ten-percent confidence level, requires some 105, 000 trials if no output faults are experienced and 530, 000 trials if one fault occurs. For a mathematical confidence level of ninety-percent, the required number of trials is increased to 2, 300, 000 and 3, 900, 000 respectively. Assume that a trial period must be divided into 10, 000 time increments for appropriate event definition. If a computer requires ten microseconds to examine each increment, the trial time is one-tenth of a second. If a million trials are required to demonstrate the output probability, nearly twenty-eight hours of computer time will be used. General purpose computer facilities are usually not available for such periods. Two possible solutions are evident: Either acquire a fast computer that's cheap to run, or find a mathematical method with which to reduce the necessary number of trials. At the present time both solutions appear feasible. The fast, "cheap" computer is within the "state-of-the-art"; and Monte Carlo reduction techniques are now being tested with excellent results.

SUMMARY AND CONCLUSIONS

Fault Tree Analysis provides the systems safety engineer with an effective tool for the identification and evaluation of potential system hazards. Fault tree construction techniques have been devised that identify secondary component failures as well as primary events. Computational scheme s are available to make rapid, but approximate fault tree probability evaluations. Simulation is recognized as the best means to accomplish these evaluations, but lengthy computer mn-times presently limit its application to high-priority problems.

It is doubtful that significant improvements in fault tree construction techniques shall occur in the near future. The availability of capable analysts is such that further refinements appear to be impractical if large systems are to be analyzed. Further progress is expected, however, in the improvement of probability evaluation methods, particularly in the area of simulation. At the present rate of development, it is expected that an economical schen~ will be available to the industry before the end of the year.