Loughborough University

This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution**. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

= **No Derivative Works**. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

LOUGHBOROUGH
UNIVERSITY OF TECHNOLOGY
LIBRARY

AUTHOR
MARTIN-SOLIS, G

COPY NO. 116981/02

VOL NO.
CLASS MARK

~~date due~~
LOAN COPY
~~Date Due~~:

date due:-
-4 FEB 1991
LOAN 3 WKS. + 3
UNLESS RECALLED
[illegible]

date due:-
12 JUN 1991
LOAN 3 WKS. + 3
UNLESS RECALLED

date due:-
-9 SEP 1991
LOAN 3 WKS. + 3
UNLESS RECALLED
SOUTH BANK
POLY.

-1 JUL 1994

~~-1 JUL 1994~~

13 JUN 1997

011 6981 02

Bound by BADMINTON PRESS
Tel :Leicester 606534
England

date due:-
- 4 FEB 1991
LOAN 3 WKS. + 3
UNLESS RECALLED
GALLERY

date due:-
12 JUN 1991
LOAN 3 WKS. + 3
UNLESS RECALLED

date due:-
- 9 SEP 1991
LOAN 3 WKS. + 3
UNLESS RECALLED
SOUTH BANK
POLY.

- 1 JUL 1994

- 1 JUL 1994

13 JUN 199

date due:-
- 4 FEB 1991
LOAN 3 WKS. + 3
UNLESS RECALLED
GALLERY

date due:-
12 JUN 1991
LOAN 3 WKS. + 3
UNLESS RECALLED

date due:-
- 9 SEP 1991
LOAN 3 WKS. + 3
UNLESS RECALLED
SOUTH BANK
POLY.

- 1 JUL 1994

- 1 JUL 1994

13 JUN 199

SUMMARY

A method of fault tree synthesis relevant to the generation of fault trees for use both in hazard assessment in the design of plant and in the analysis of fault conditions using a process control computer is described. List processing techniques and bit manipulation have been used to reduce computation time and computer storage requirements. The programming language used in the development of the algorithm was RTL/2. Several examples are presented to illustrate the methodology.

FAULT TREE SYNTHESIS FOR REAL TIME AND DESIGN APPLICATIONS ON PROCESS PLANT

by

Gregorio Alberto Martin-Solis

A Doctoral Thesis submitted in partial fulfilment
of the requirements for the award of Doctor of Philosophy
of the Loughborough University of Technology

November 1978

Supervisors: Dr. P.K. Andow and
Prof. F.P. Lees

Department of Chemical Engineering

© by Gregorio Alberto Martin-Solis 1978

Loughborough University
of Technology Library
Date Mar. 79
Class
Acc. No. 116981/02

TO MY WIFE AND FAMILY

TO MY PARENTS

TABLE OF CONTENTS

Tables

Figures

Figures

Page

Figures Page

ACKNOWLEDGEMENTS

The author wishes to thank Dr. P.K. Andow and Professor F.P. Lees for their advice and encouragement during the course of the work herein reported.

The author is also indebted to the British Council for its financial support and to all those others who, although not mentioned here, influenced the process of completing this work.

Chapter 1

INTRODUCTION

When chemical processing plants started to become more complex, new techniques to help in the control of these plants were developed. This made them, in some respects more reliable.

Initially automation helped the operator to achieve acceptable control of complex systems. Later on the widespread use of analogue controllers and the arrival of the computer changed the functions of the operators. The number of operating personnel in the plants was reduced and a tendency to locate the control of a plant in central control rooms arose. With this trend the operator became more and more like a supervisor whose functions were to diagnose the faults that occurred in the plant and to take remedial action. This was not an easy task and sometimes costly mistakes were made.

Nowadays the new complex processing plants have a system of visual and audible alarms to help the operator in the diagnosis of a fault, but even with these aids his problems have not been solved. Sometimes he finds himself monitoring several hundred variables at a time, if a fault occurs in a part of the plant, several alarms may become active within a few seconds and he has to find out which of the variables that triggered the alarms is in fact the original cause of the trouble. When a difficult situation suddenly arises, the operator is under enormous stress, he has by means of his knowledge of the system to be able to reach a decision, to correct the fault and bring the plant back to normal. He has to do this without delay because, if a certain combination of conditions occurr or no remedial action is taken after a fixed time delay, a "trip" system might be activated to shutdown the plant. Bearing in mind that the more complex a plant is the higher are the costs of shutting it down, usually the operator will try to avoid the use of the "trip" system. It is in these difficult situations that his diagnosis is prone to error.

It was felt that something should be done to help the operator in those difficult situations. The first plants that started to incorporate aids concerned with the problem were nuclear power plants; they introduced alarm analysis schemes.(K1) The alarm analysis technique attempts to help in the solution of the problem by determining the relationships between the different alarms and providing the operator with this information. This technique is now well established in nuclear power stations, but in other industries like the chemical process industry little has been done to solve the problem.

This work originated from the studies carried out by Dr. P.K. Andow and Prof. F.P. Lees on the use of a process control computer to analyse process alarms as they occur in the chemical plant. In their studies they found some deficiencies in the alarm systems and it was suggested that one facility which would greatly improve alarm systems could be the ability to analyse alarms to determine the basic cause of trouble.

Dr. Andow developed a method(A2) based on the concept of information flow and in which the determination of the alarm relations is carried out automatically. The technique used in his method, is first to generate the network relating to all the process variables and then reduce the network to a network relating only to those process variables on which there are alarms. A unit model approach is used and functional models with direction are the basis of the approach. With the experience obtained from the development of the method for alarm analysis, it was considered that an investigation of other alternatives such as fault trees could provide some answers to the problem of alarm analysis in Chemical Process plants.

Fault tree analysis (FTA) and Fault Tree Synthesis (FTS) can have several uses in the Chemical Process industry, some of the aspects in which it could be used are:

1) To find the main cause of the problem.
2) To evaluate reliability in a plant.
3) To find the most effective place for the process alarms.
4) To help the operator in the "control room" to make the correct decision.

FTA could be used in a wide range of the chemical engineer's activities, from the design stage until the full commissioning and operation of a plant; still fault tree analysis is not widespread in the Chemical Process industries, as it is in other industries (aerospace, nuclear). The literature survey revealed that the causes of the slow spread have been:

1) The nature of chemical processing systems. They have an arbitrary configuration, their performance has to be described taking into account their material, momentum and energy balance relationships.

2) The lack of skilled fault tree analysts in the chemical process industry.

3) The lack of automated systematic methodologies for fault tree generation.

Nowadays some new methodologies have emerged and the state of the art has been advanced. Fault trees have been used at the design stage of a plant for hazard assessment and the automated generated trees, have proved useful for this purpose.

In spite of the new methodologies some areas in which the fault tree techniques could be used still remain "untouched". Some authors(A2,L1,F3) have mentioned the possible use of fault trees for alarm analysis and as a way to help the operator in the control room to make the correct decisions, but nothing has gone beyond that point.

This work was intended to explore new areas of the Chemical Process industry in which fault tree analysis could be applied. By looking at some of the problems of Alarm Analysis it was found that fault trees could be used to help the operator in the difficult task of finding the original cause of trouble in a plant, when several alarms were active.

A feasible approach was to consider every alarm as a top event and then by developing the top event, the causes of the alarm could be found. Measurements could be checked for each variable with an alarm, at the moment of developing the tree, in this way some branches of the tree would not have to be developed if the variables were inside their limits. Other basic faults such as blockage or mechanical failure could be considered by giving them a probability weight in the tree. Once the tree was developed the operator could use the information provided by the tree to reach a better decision about any further action in the plant. The approach described, required a methodology capable of overcoming the real time problems of fault tree synthesis. The aim of this work is to present a methodology capable of dealing with these problems.

To develop the methodology described in this work the use of a process computer was ideally suited, but it was necessary to have a computer language capable of handling the data required to build the fault trees in an efficient way. The main reason for this was that computational problems involved were quite different from those found in numerical programming problems. RTL/2 was the language chosen for this.

The content of this thesis is arranged in such a way that the central theme of "fault trees for design and real time purposes" is preserved. Other aspects of the work which are peripheral to this theme will be confined to the Appendices. The literature survey presented in Chapter 2 is mainly a review of the most important papers concerned with Fault Tree Analysis and Synthesis since the origin of these techniques.

The methodology used in this work has several input requirements which are summarised in Fig. 1.1 Each one of these requirements will be discussed in the work. Chapter 3 is devoted to discussing the basis of the modelling method used in the methodology and the role played by the models in the construction of the minitrees. These minitrees were manually built in this work. Chapter 4 describes the use of the minitrees to construct fault trees. Chapter 5 presents the synthesis methodology and the different algorithms required. The use of the methodology for design and real time purpose is illustrated by means of several examples in Chapter 6. Finally Chapter 7 is devoted to a discussion of the work and its further applications.

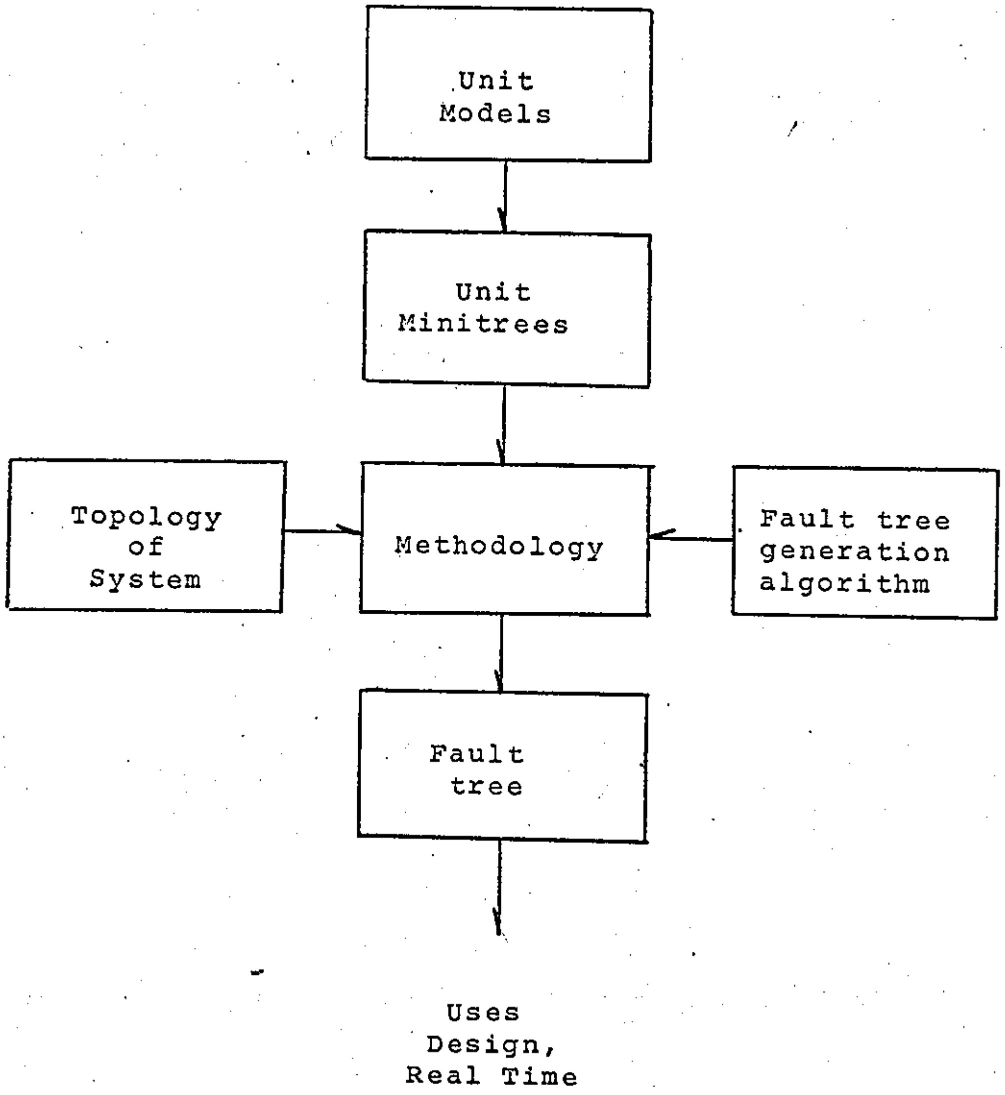


Fig. 1.1 Input requirements for the methodology described in this work

Chapter 2

Literature Survey

The distinction between analysis and synthesis can be stated as follows: Analysis deals with the understanding of how things are and how they work, once the analysis is done, the result can be used in the creation of artificial things with certain desired properties, this creation process is called synthesis.(R1).

Bearing in mind this distinction the aim of this chapter is to cover the area of Fault Tree Analysis (FTA). It is mainly a review of the most important papers concerned with FTA since its origin. Some of the basic concepts and techniques are described in order to make it easier, for those non-familiar with the subject, to have a better understanding of the chapters to follow. Any other references with other aspects of this topic will be inserted where relevant to the text.

The concept of fault tree analysis was originated by H.A. Watson of Bell Telephone Laboratories in 1961 and refined by a study team of the same firm 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 analysing the "false" logic which results from component failures. Further, this was ideally suited to the application of probability theory in order to numerically define the critical events. The Minuteman was successfully completed using the new technique, and provided convincing arguments for the incorporation of a number of equipment and procedure modifications.(H1)

Since the concept of Fault Tree Analysis (FTA) was introduced, significant refinement of the analytical and mathematical techniques used has taken place. Fault Tree Analysis evolved in the aerospace industry in the early 1960's and later on in the nuclear industry.

In 1965 at a Safety Symposium in Seattle, Washington; D.F. Haasl and other authors presented their advances achieved in using the fault tree technique. The two major steps of FTA were identified as:

1) The construction of the fault tree.
2) Its evaluation.

In his presentation Haasl(H1) described two techniques, that he and a team of analysts of the Boeing Company, used for fault tree construction. He called them "Primary Failure Technique" and "Secondary Failure Technique". For Haasl the failure of a component was "primary" if it occurred while the part was functioning within the operating parameters for which it had been designed. It was termed "secondary" if the failure occurred when the component was subjected to abnormal environmental stresses such as failures in related equipment. With the Primary Failure Technique, the failure of one component is presumed to be unrelated to the failure of any other component. The tree must be developed only to the point where identifiable primary component failures will directly produce the required fault events. With the Secondary Failure Technique, all significant fault interrelationships must be developed. The analysis does not stop when it reaches the component level. It continues until the effect of each component, and the possible failures of all related components, has been portrayed.

Haasl considered that the fault tree provides a concise and orderly description of the various combinations of possible occurrences within a system which can result in a predefined "undesired event". Before construction of the fault tree can be carried out the analyst must acquire a thorough understanding of the system.

Figs. 2.1 and 2.2 show the nomenclature that Haasl and other participants in the symposium used in their presentation to represent the trees in fault tree analysis.

The nomenclature can be classified in two types:

1) Logic gates.
2) Event symbols.

The fundamental logic gates for fault tree construction are the AND and the OR gates. The AND gate describes the logical operation whereby the coexistence of all input events is required to produce 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.

The Event symbols used to represent the components of the tree were not always the same but the most "popular" are shown in Fig. 2.2.

A rectangle defines an event that is the output of a logic gate and is dependent on the type of logic gate and the inputs to the logic gate. A circle defines a basic inherent failure of a system element when operated within its design specifications. A diamond defines fault inputs that were considered basic in a given fault tree, but the possible causes of the event were not developed either because the event was of insufficient consequence or the necessary information was unavailable. Triangles are used as transfer symbols. A line from the apex of the triangle indicates a transfer in, and a line from the side denotes a transfer out.

There are other logic gates, but they are based upon special combinations or modifications of the fundamental gates. One of these "special" gates is the "INHIBIT" gate, it was used by Haasl in his secondary fault technique. The "INHIBIT" gate defines the situation where the co-existence of an input event and a conditional event is necessary in order to produce the output event. In this situation a dependency relationship exists between the input event and the conditional event. The input event directly produces the output event if the indicated condition is satisfied, (see Fig. 2.3).

This nomenclature is now familiar to anyone initiated in fault tree analysis and has become the common nomenclature to represent fault trees.

The example used by Haasl to illustrate the difference between the two techniques for fault tree construction has become a classical example in the fault trees literature.(F2,N2) This example is shown in Fig. 2.4. It is a system designed to make available mechanical energy from the motor whenever the switch is closed by the action of an external 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 the 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-energise the circuit. Haasl assumed that the "undesired event" was the destruction of the wire between A and B due to overheating. From this point Haasl started the construction of the tree, by looking at the causes that could produce the "undesired event" and then the combination of events which could produce those causes and so on until he reached a point where he had to choose between the two techniques of analysis; either to consider only primary failures or to include secondary failures as well. Fig. 2.5 and 2.6 show the resultant fault trees for each technique. It can be seen from Fig. 2.6 that for large systems the magnitude of the task assumes phenomenal proportions, but Haasl considered that the resultant definition of system failure nodes invariably justified the effort.

The next step after the construction of the fault tree is the evaluation of the tree. Given that the fault tree is the representation of events in a symbolic logic format, Boolean algebra may be used to symbolically express the fault tree. On this basis Haasl proposed two methods for probability evaluation:

1) Analytic, which implies the probability evaluation of each input event in order to compute by means of an algebraic expression the probability of the output event. The drawback in the use of this method was that all available computational techniques required approximations which could not be avoided.

2) Simulation, this was a way to avoid approximations and it was done by actually simulating the various logic gates and input events within a computer. The drawback of simulations is the amount of computer time involved.

In his conclusion he pointed out that it was doubtful that significant improvements in fault tree construction techniques could occur soon, however he expected improvements in probability evaluation methods.

R.J. Feutz(F4) was another of the participants in this symposium. In his presentation he refers to the term "fault tree" as a technique that graphically depicts the relationship between certain events and the ultimate undesired events, where the probability of occurrence of this undesired event, represents a measurement of system safety. He considers that due to the myriad details that the analyst needs, to determine all the probable ways in which a system can fail, the first step in the construction of the fault tree is to understand the system down to its basic components and develop a mental picture of their relationship. He stresses that the ability to construct a meaningful fault tree comes only from experience. Feutz's examples are based in the aerospace industry but nevertheless are very illustrative.

A.B. Mearns (M1) from Bell Telephone Laboratories also contributed to this symposium with a paper. For him Fault Tree Analysis was a technique which provided a method of analysing a system, recording a vast variety of combinations of events in an easy-to-read diagram, and straightforwardly highlighting the most important elements.

Mearns' presentation summarises the fault tree analysis in the following six steps:

1) <u>Obtain source materials</u>. By this he means all the physical layout plans, detailed logic and schematic diagrams which are required for a preliminary failure analysis.

2) <u>Conduct preliminary analysis</u>. The aim of the preliminary analysis is that it begins to isolate the general portions of the overall system that are important from the standpoint of safety and which will lead after the construction of the tree. This is the equivalent to the step of defining the "undesired event" which Haasl and Feutz mentioned in their presentations.

3) <u>Construct the fault tree</u>. Based on the two former steps Mearns discussed at this point how to build the tree and the symbols to use. He used the same logic gates already depicted in Fig. 2.1, although he used different symbols for the events of the tree.

4) <u>Simplify fault tree</u>. Mearns used Boolean Algebra to simplify the tree; this notation is shown in Fig. 2.7 and an example for a small tree is depicted in Fig. 2.8. The product of this step is a simplified Boolean expression containing only actual contributors to the investigated event.

5) <u>Estimate probabilities of failure</u>. At this step Mearns noted that the basic events in the fault tree were chosen to be statistically independent; that is, the existence of a failure of a given component did not affect the probability of failure of any other component and this fact, permits probabilities of failures to be substituted almost directly into the reduced Boolean expression for the fault tree. Unfortunately, this assumption is not always true. In some cases (chemical plants) modifications are necessary. The analyst must be careful at this stage

in order to obtain a conservative answer.

6) Identify sensitive areas. He noted that once the former steps have been completed, the probability of the undesired event can be evaluated and the result might typically be as below:

$$P(\text{undesired event}) = 2.5*10^{-12}+2.3*10^{-4}+1.6*10^{-10}$$

It can be seen clearly that the combination of events which make up the second term is the most probable cause of the undesired event. This step has the purpose to point up the most likely cause of an event and compare it to other possible causes.

Even though the cited authors did not use exactly the same terminology in their presentations, all of them were very alike and gave a great momentum to the development of the fault tree technique. It can be said that this symposium marked the beginning of a widespread interest in using the fault tree technique as a reliability and safety tool in aerospace, nuclear and some other industries.(F5)

The nuclear industry soon realised that FTA could be very useful. In March 1966 Kay and Heywood(K1) presented a paper where they mention the use of tree analysis at Oldbury Nuclear Power Station. In November 1968 Welbourne(W3) published a paper concerning the data processing and computer control system at Wylfa Nuclear Power Station. The various possible analysis methods described by Kay and Heywood were discussed and the tree-analysis was chosen as the most suitable method to use in alarm analysis.

In February 1970 Ericson(E1) published a paper where he described the fault tree methodology. It is a good review of all the important concepts used in FTA. In this paper he describes the different "special" gates that can be used in FTA. He noted that all these special gates such as the EX-OR gate, the INHIBIT gate and some others are based upon special combinations or modifications of the fundamental gates AND and OR. Ericson considered the two types of evaluation are possible, qualitative and quantitative. Qualitative evaluation could be regarded as an inspection or an engineering judgement assessment of the fault tree. Quantitative evaluation was defined as the traditional numerical evaluation, where failure rates of the system elements are inserted into the fault tree structure and mathematically combined to yield probabilities. Ericson considered that the purpose of the evaluation was to determine from the fault tree the risk that is associated with the undesired event and to identify which event, or events are unacceptable and must be eliminated or controlled in order to eliminate or control the occurrence of the undesired event.

Crossetti and Bruce(C1) published a paper in April 1970 where they mention that Douglas United Nuclear adapted and applied the fault tree technique, to the nuclear reactor plants they operate for the Atomic Energy Commission, with good results.

In August 1971 Crossetti(C2) published another paper on FTA. He said "The Technique has proven to be a cost-effective systematic and descriptive method that can be applied to safety and operational analysis of a system from its conception and design through the manufacturing testing and operation phases. Fault tree analysis is also flexible enough for application to individual pieces of equipment or to the overall plant". Crossetti's paper mentions that an important feature of the fault tree approach is that it can include a sensitivity analysis to evaluate the significance of events for which information does not exist or is of poor quality, in this way, insignificant events can then be eliminated while the significant events are reviewed in terms of data quality needed or degree of control required.

With the development of FTA the need for automating fault tree construction and fault tree evaluation for computer implementation soon arose. To satisfy this need several methodologies were developed(V1,V2, V3,S3), most of them were concerned wtih the evaluation of the trees and very few with the automating construction of the trees. One of the most useful methodologies, was the one developed by Fussell and Vesely(F1) to help in the qualitative evaluation of the fault trees, to obtain all the unique modes of system failure, called minimal cut sets. A cut set can be defined as a set of basic events whose occurrence cause the "undesired" or top event to occur.(B1) A cut set is considered minimal if it cannot be reduced and still ensures the occurrence of the top event. Fussell(F2) defines a minimal cut set as the smallest set of primary events which must all occur in order for the undesired event to occur. A complete set of minimal cut sets are, all the unique failure modes for a given system and its "undesired" top event. Automated analysis was necessary for the large trees and the algorithm developed by Fussell and Vesely did not have the drawback of other programs available, which was the excessive execution time, to obtain minimal cut sets that contained a large number of primary events. The algorithm is based on the fact that an AND gate always increases the size of a cut set, while an OR gate always increases the number of cut sets. The execution time is then approximately a linear function of the length of the cut sets. The algorithm obtains cut sets such that, if all the primary events were different, the cut sets so generated would be precisely the minimal cut sets. When this is not the case, the cut sets generated by the algorithm are then reduced to minimal cut sets. As a matter of illustration on minimal cut sets an example on the use of the algorithm is presented here for the simple fault tree shown in Fig. 2.9.

The gates AND and OR are labelled in this case from 1 to 3. The algorithm begins with the gate immediately below the top event, if the gate is an OR gate each input is an entry in separate rows of entries matrix. If the gate is an AND gate each input is used as an entry in the first row of the matrix. In this example gate 1 is an OR gate so the matrix will have A and 2 in separate rows. The idea of the algorithm is to replace each gate by its input gates and basic events. This procedure is repeated until all the entries of the matrix are basic events. Fig. 2.10 shows how the replacement takes place for the example considered. The cut sets obtained by the algorithm are called Boolean Indicated Cut Sets or (BICS). In this case there is no replication of basic events in the fault tree and the BICS are the minimal cut sets. Once the BICS are all determined and there are some replications of the basic events a simple search is used to determine the minimal cut sets.

In 1972 Haasl(L1) presented a "structuring process" that established rules to determine the type of gate to use and inputs to the gate. The structuring process is used to develop fault flows in a fault tree when a system is examined on a functional basis. Fig. 2.11 shows the different levels of fault tree development that Haasl used in his "structuring process". Haasl's structuring process identifies three failure mechanisms or causes that can contribute to a component being in a fault state:

1) A primary failure is a failure due to the internal characteristics of the system under consideration.
2) A secondary failure is a failure due to excessive environmental or operational stress placed on the system element.
3) A command fault is the inadvertent operation or non-operation of a system element due to failure(s) of initiating element(s) to respond as intended to system conditions.

In March 1973 Fussell(F2) presented a formal methodology for fault tree construction. He called it "Synthetic Tree Model" (STM). It was presented as a model for formulating the Boolean failure logic, or fault tree, for electrical systems from associated schematic diagrams and system-independent component information.

The Synthetic Tree Model could be summarised as a synthesis technique for piecing together with proper editing, a fault tree from small segments called "component failure transfer functions". The component failure transfer functions, can be considered as minifault trees for components in a fault state and are obtained from a system independent failure mode analysis of individual components. The failure mode analysis identifies all possible means by which a component can fail to perform its required functions. Once the "component transfer functions" are obtained, they may be used repeatedly without modification for any system in which the component appears.

In his STM Fussell uses a different structuring process, he considers four basic types of fault events, in contrast to Haasl's$^{(H1)}$ two basic fault events. Fussell considered that fault events that are used only as top events are First Order Fault Events and must be developed manually to the level of higher order fault events before STM can automatically construct the fault tree. Fault events that state a fault condition of the system that extends beyond any single component are Second Order Fault Events. Fault events that cause a component to "behave failed" because part of the system itself, not simply another individual component, is causing that component to behave failed are Third Order Fault Events. Fussell classified Third Order Fault Events in two classes; Class I indicated a Third Order Fault Event that required an AND gate while Class II required an OR gate. Fault events that resulted in component A behaving failed because another component had direct input to component A were Fourth Order Fault Events. The symbols used in STM are the same as those used by Haasl. The methodology for the fault tree construction is programmed in a computer code called DRAFT. The information required as input to the code is:

1) A schematic of the system.
2) The initial operating state of each component if applicable.
3) The boundary conditions that can impose restrictions on the top event to be developed. These boundary conditions will affect all the events that result from the development of the top event.

With this information the computer proceeds to find the series circuit path for each component in which the components share an alliance with respect to flow, Fussell calls them component coalitions, and to identify the order of each event that has to be developed. A flow diagram of the methodology is shown in Fig. 2.12.

Fussel claims that any number of analysts constructing fault trees independently for a given system and main failure event using this model will obtain identical fault trees. The model has some limitations, it does not account for secondary failures, that is, a failure that occurs when the component is subject to abnormal environmental stresses as failures in related equipment, is ignored by the model. The fault trees are constructed to the point where identifiable primary component failures will directly produce the fault event in question. The automated analysis is a hardware-orientated approach and does not include environmental and human effects that can cause failures. Fussell considers that automated analysis should be thought of as a distinct type of analysis that could never replace conventional fault tree analysis.

In spite of these limitations Fussell's paper is the first to develop formal concepts and techniques of fault tree synthesis for electrical systems; without doubt this paper marked the beginning of a new stage in fault tree analysis.

In April 1974 as a result of the NATO Advanced Study on Generic Techniques of System Reliability Assessment (Liverpool, July 1973) an article was published by Fussell, Powers and Bennetts.(F3) In this paper each author expressed his point of view about the state of the art of fault trees; the authors do not always agree especially Fussell and Powers concerning the limitations of automated fault tree synthesis.

Fussell recalls some of the points that he has already mentioned in his previous papers;(F2,F5) he makes reference to the published information dealing with fault trees as quite a limited one. He quotes Haasl as - one who has described some general concepts, himself as that one who has presented a formal, deductive construction methodology for electrical systems and Powers as one who has formulated a formal, deductive technique for chemical processing systems. He emphasises again that in order to construct a meaningful fault tree the analyst must have a complete understanding of the system; that the automated approach is a distinct approach that does not replace conventional fault tree construction; it is a hardware-oriented approach because it does not include environmental and human effects that can cause failures and therefore is apart from a true indepth fault tree analysis.

On the other hand, Powers describes how it could be possible to apply FTA for chemical processing systems with the help of computers, performance and failure models, property data and a way to define the logic for the propagation of mass, momentum, energy failures through a complex system. He mentions that they are working on developing a system of computer programs and failure models for common chemical processing units to help the chemical process safety analysts in the solution of the fault trees problems.

With reference to the computer system he says, that although they are just beginning to gain experience with it, the approach appears feasible and they are currently expanding their performance and failure models to include environmental and human effects; one of the advantages that he sees in those models is that they can be used for operators training because they constitute a Boolean simulation of the process and the effects of failures are easily generated from the Boolean model.

Bennetts approaches fault trees from electronics engineering logic-networks point of view, the terminology he uses is slightly different; primary failures become primary inputs; top event becomes primary output; minimal cut-set becomes prime implicant. He mentions a technique for analysing combinational networks that had not been mentioned in fault trees literature and that he describes in his Ph.D dissertation.(B4) It is the same in concept to that one used by Fussell for obtaining minimal cut-sets but the difference arises in the implementation. Each gate output is defined by its inputs and logical function using a reverse Polish notation and these are used to develop a reverse Polish expression for the primary output in terms of the primary inputs. (Reverse Polish Notation (RPN) is a notation originated by the Polish mathematician Lukasiewicz. It is used to translate arithmetic expressions so that the operators are written at the end of the expression instead of in the middle. Thus A + B would be written in RPN as AB+).(F6) This expression is then unpacked into an equivalent reduced sum-of-product expression and, by repeated application of the absorption rule (A+AB=A) and other rules, the final expression consists only of prime implicants (minimal-cut sets). A feature of this technique is that it is capable of analysing networks containing logic functions other than the simple "AND" and "OR" functions, e.g. "EX-OR" and "NOT" . (EX-OR operator has two inputs and will produce a true output if one, but not both, of them is true).

Bennetts called the attention of analysts to the fact that the analysis of logic networks had not been studied too closely, except through simulation. It was very likely that the algorithms and software developed to solve the problem of adequate testing sequences for networks, could be used to evaluate fault trees.

In March 1974 Powers and Tompkins(P1,P2) presented a procedure for automatically generating fault trees for chemical process and from this they conclude that an approach to chemical plant safety analysis appears to be feasible. They mention that the trends to increase plant complexity and larger process size is a challenge that makes it desirable to develop these techniques. They point out that the use of fault trees in the chemical process industry has not been widespread because there existed no automated systematic methodology for fault tree generation for this industry. The approach has been advocated by Recht(P1) and a simplified technique based upon fault tree analysis has been implemented by Browning,(P1) but the limitations imposed by hand generation offset the inherent utility of the technique. Kletz(K2) used fault tree analysis, though he called the trees logic diagrams, in specifying and designing protective systems for chemical plants, he mentions that at least one week was spent in getting the logic diagram correct. Lawley(L2) used fault tree analysis in hazard analysis, he called the trees logic trees and reported a total of 1 man day to construct the logic tree for a simple process.

The information requirements for fault tree generation that Powers and Tompkins believe are important to perform a meaningful fault tree analysis is shown in Fig. 2.13. In the methodology presented by Powers and Tompkins a modular approach is considered and the concepts of information flow(R1) are utilised, to formulate each unit model. Two types of models are used in the methodology, the performance models and the failure models; a failure model is associated with each unit performance model. The models are based on mass energy and momentum balances for each unit. The topology of the system is considered as a network consisting of units interconnected in a specific fashion. Fault tree generation starts with the definition of final hazard states. The final hazard states are defined by mini-fault trees, as the one shown in Fig. 2.14, which corresponds to species or process properties. The specific form and location for the final hazard event is dependent on the properties of the materials and the characteristics of the equipment in the system under study. After the top event is defined, the unit models and the topology are used to identify all the possible ways in which each of the events that are inputs to final hazard event minitree can be caused. The performance models are used to determine, if a specific failure event under study, will propagate through the units; once this is done the failure models are used to develop the major failure events to their primal events in those units that will propagate the specific failure under study. The overall strategy for generating event trees is shown in Fig. 2.15.

A more detailed description of the methodology is presented by Tompkins(T1) in his Ph.D dissertation. In it he compares his work with

Fussell's. He mentions that "the framework proposed by Fussell is not adequate to handle the case of a chemical processing system of arbitrary configuration". The basic criticism is that his unit models (component failure transfer functions) bear no relationship to and take no account of the material, momentum and energy relationships which describe the performance of chemical processing systems.

The importance of Fussell's contribution is that it demonstrates the need for formal standardised fault tree generation procedures, and shows that it is possible to automate this procedure for electrical circuits.

Tompkins' orientation is towards consideration of internal failure propagation as opposed to external propagation. Only those failure events are explicitly treated which propagate from unit to unit based upon physical connections.

Tompkins' methodology can be summarised in five steps:

1) Determination of the top event.
2) Location of potential information flow sources.
3) Generation of source to destination information flow pathway.
4) Development of the required failure events for each pathway.
5) Formation of the fault tree as the union of successful pathways.

One of the conclusions of his work is that although FTA provides a powerful means for the _a priori_ enumeration and evaluation of failure events, it is necessary to have major contributions to the area of fault tree generation before it will be adopted by the chemical process industry. Although the methodology was not implemented on a digital computer, the principles developed were quite useful to advance the state of the art and was the first approach to solve the problem of fault tree synthesis in the chemical process industry.

In July 1974 a paper based on testimony given by Rasmussen (R3) before the United States Congress Joint Committee on Atomic Energy Hearings on Nuclear Reactor Safety (Sept. 1973) was published. This article was effectively an advance copy of the report of the Reactor Safety Study made by the U.S. Atomic Energy Commission (R2) which was published a few months later. Rasmussen describes the way his team conducted the study and how it uses the fault tree technique as a tool to obtain the probability of the branches of event trees. He recognises that there are very few people skilled in the techniques on FTA and because of that they had to borrow people from other companies to carry out part of the study. He notes that one of the problems that would have to be overcome before the technique could have widespread use is the shortage of trained analyists.

Another approach to fault trees was presented by Nielsen (N2) in 1974. He calls the method Cause Consequence Chart (CCC) and it is a unique blend of fault tree analysis and failure mode and effect analysis (FMEA). The difference (L1) between FMEA and FTA is that FMEA involves generation of event sequences from initiating events to final events, while FTA begins with the final event and works towards the initiating events. FTA has a distinct advantage in that only event sequences leading to failures of interest are considered.

Taylor(T4) refers to CCC as a diagrammatic technique for presenting the sequence of events leading to a failure and conditions under which these events can take place. Nielsen's method applies to dynamic models as well and is a more detailed approach to find the "Failure Transfer Functions" that Fussell uses in his Synthetic Tree Model, the examples presented by Nielsen are electrical systems but he claims that they can be used in other kinds of systems.

In October of 1974, Stewart(S4) published a paper where he describes the design and operation of a protective system used by ICI on its plants to reduce the frequency of spurious shutdowns and to increase the system reliability. He calls these protective systems "High Integrity Protective Systems" (HIPS). The logic diagram for the design of the system is a fault tree even though he calls it a "family tree". It is a good example of the application of FTA in a plant already in use, FTA had been used before, as a design tool rather than as an operational one in systems already designed and in use.

In January 1975 Neogy(H1) presented a paper in which he describes the use of fault trees in Ocean systems and demonstrates the applicability of FTA to this field. He concludes that the state of the art in fault tree techniques has yet to be advanced and will do so more rapidly by more people attempting to use it in the real world; the only disadvantage he sees in the fault tree is the considerable time involved in its construction.

At the same Symposium as Neogy, Bass[(B3)] et al., presented a paper on Fault Tree graphics. This paper describes a system concerned with helping the analysts when using FTA by means of a digital computer. The system described allows the analyst to draw, modify and evaluate trees, this is done by means of an interactive computer graphics terminal that is composed of a light pen, a typewriter keyboard, a function keyboard and a Cathode Ray Tube (CRT). The fault tree evaluation is carried out with the help of the MOCUS computer algorithm and the KITT program. The authors claim that with this system, complex engineering designs can be analysed. The system seems to be quite useful and could save some time to the analyst, mainly at the design stage of a project when major changes are involved.

In April 1975 Pandl[(P3)] et al., presented a computer program to carry out fault tree analysis. The two codes TREEL and MICSUP are described and an example is used to illustrate the codes. They claim that the codes perform functions similar to MOCUS and KITT but with a better methodology and efficiency. They mention that better methods to evaluate fault trees are needed to cope with the wider class of fault trees.

In October 1975[(L1)] Lambert presented the description of a general simulation model of system failure in terms of fault tree logic. He claims that this model can be used to assist an operator in making decisions under a time constraint regarding the future course of operations. The model that he suggests, assumes that a fault tree for the system under study has been constructed and that its cut sets and failure data have been stored in the memory of a digital computer. When a fault occurs the operator feeds the computer with data of the state of the system. The answer that he gets from the computer depends on the time available for checking any known component failures. The computer can give the operator a list of the events that he should check first and the operator will feed the information required back to the computer through a teletype. The iteration process can be carried on until the occurrence of a min cut set is observed or a false alarm diagnosed. In his conclusions Lambert mentions that a major disadvantage of FTA is the possibility of oversight and omission. He considers that a solution to this could be automated fault tree construction. The automated approach can be used to standardise fault tree analysis and eliminate the confusion created by the different ways in which analysts can construct fault trees.

A problem that he sees in fault tree modelling is the difficulty to apply Boolean logic to describe failures of system components that can be partially successful in operation and thereby having effects on the performance of the system; to illustrate this he mentions the leakage through a heat exchanger. To solve this problem the analyst may have to describe the process analysed in terms of the basic laws of mass, energy and momentum balances. He considers that it would be a difficult task to program his fault tree simulation model.

Andow and Lees[(A4)] published a paper at the same time as Lambert's. In the paper they discuss the use of fault trees in alarm analysis together with some other techniques which could be useful in alarm analysis for chemical processes.

In November 1975 Salem(S1) et al., published a paper where they described a method for the automatic construction of fault trees. A computer code CAT is used to construct the fault trees. The method employed models various components in terms of "Decision Tables", which are extensions of Truth Tables using multistate variables. They claim that one of the advantages in using decision tables is the complete generality of the approach, which allows the representation of any number of signal or flow states, and is not restricted to modeling hardware only. The paper presents little detail about the methodology. Although the authors mention that much work remains to be done they claim that techniques for compacting the decision tables have been developed to reduce program storage and running time, and that the program has produced good results for simple systems.

In 1975 a book on Reliability and Fault Tree Analysis was published by Barlow(B5) et al., it presented several papers on FTA. Most of the content of those papers had been published before by their authors and have already been mentioned in this survey. The book can be considered as a good review of the theory and applications of FTA. The paper by Powers(B6) et al., in this book presents their work on the development of a simulation language for safety analysis of chemical processing systems. They call the language SESIL and it is based on a set of Boolean models which they claim describes the safety performance of equipment commonly used in chemical processes. They consider five phases in the use of the language:

1) Preparation of input data.
2) Definition of potential hazards.

3) Evaluation of hazards.

4) Fault tree synthesis.

5) Fault tree evaluation.

In the synthesis of the tree they used the method presented by Tompkins.(T1) The paper is rather general about the language. The authors mention that the lack of persons skilled in both chemical process design and fault tree analysis methods as the major reason that has limited the wide use of fault trees in the chemical process industry and claim that the approach presented will help to solve this problem. Powers et al., expected their system to be operational by the end of 1975.

In April 1976 Powers and Lapp(P4) published a paper which described briefly a Fault Tree Synthesis (FTS) program. The major steps they suggest to carry out a quantitative safety analysis of a chemical process is shown in Fig.2.16. They describe the FTS program as a symbolic process simulation. Following identification and evaluation of process hazards, a symbolic model of the complete process is assembled from models of individual pieces of equipment within the process. The models used are signed diagraphs and they claim that they have developed an algorithm that deduces the fault tree directly from the properties of the diagraph. Once the fault tree has been generated, it is placed in minimal cut set form and the probability of the top event is computed. Most of the paper refers to points already mentioned by the authors in previous papers.(P1,P2,P5) The key features of the algorithm that the authors mention in the paper are:

1) The topology of the diagraph is extremely important. Negative feedback and feedforward loops are detected and their elements determined. Cases of nested loops are also considered.

2) Conditional expansion of events is performed.
3) The changes in relationships between variables due to failures are included.
4) Common cause failures are detected directly from the diagraph.
5) Human operator actions are included.
6) Large deviations from normal conditions that alter relationships between variables are considered.
7) Events which have been previously developed are detected and copied.

It can be seen from the article that there are similarities with Fussell's (F2) model for FTS. Fussell deals with the conditional expansion of the events by means of the discriminator used in his failure transfer functions.

The concept of diagraphs used by Powers and Lapp is very similar to that one of networks described by Andow and Lees (A4) to solve the problem of information flow in their Alarm Analysis method. Andow and Lees generate the network relating all the process variables and then reduce this to a network relating only all the process variables in which there are alarms. Powers and Lapp use diagraphs to obtain the information flow of the process including failure modes of the units, then they convert the diagraph to a signed diagraph from which they deduce the correct system fault tree for the process.

In 1976 Caceres and Henley (C4) presented a method to generate fault trees based on a block diagram of the system under study. The technique is based on a computer-oriented algorithm developed to detect all the minimal paths leading to the success of a system represented by a block diagram. The tree generated by the method only depicts how the failure of the system will occur if the elements fail but it does not give the causes of failure for each element.

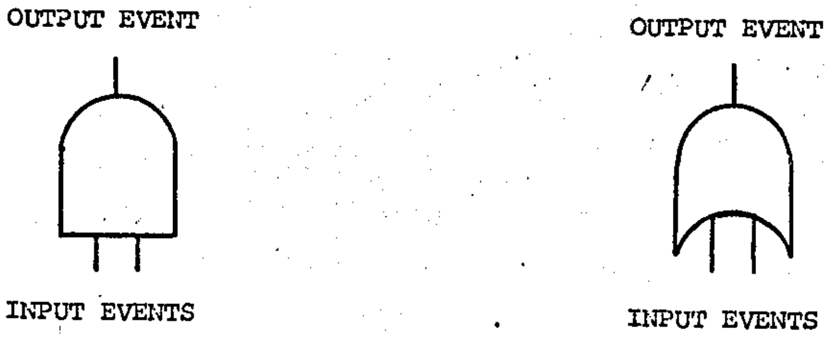
In November 1976 Salem (S2) et al., published a more detailed report on the computer code CAT (Computer Automatic Tree) used to construct fault trees. In the report the authors analyse several of the computer codes used to evaluate or to construct fault trees. Some examples are presented on the construction and reduction of the decision tables used to construct the fault trees. The approach is another step further on fault tree synthesis. Although the fault trees do not always look the same as those constructed manually, the authors claim the min cut sets obtained from the synthetic tree gives the same results.

In December 1976 Hollo and Taylor (H2) presented a methodology for consequence diagram and fault tree construction. Their algorithm for fault tree construction is based on Fussell's method and uses the list processing technique (F7) their algorithm has been implemented in a small computer and due to the limited storage capabilities has only been tested for smaller examples.

In April 1977 Lapp and Powers (L4) published another paper on their algorithm for the synthesis of fault trees. In this paper they compare their work with the one of Fussell, Taylor and Tompkins-Powers. They claim that their method has better features than any of the former algorithms because it can deal with complex systems and more gates.

This paper presents a more detailed explanation of the method than their other papers.(P4,P5) They consider that any system of constructing fault trees should have the following four characteristics:

1) Handle complex systems efficiently.
2) Consider the system topology as well as actual components in constructing the tree.
3) Handle multivalued logic.
4) During fault-tree construction, make checks to ensure consistency among events.



AND Gate

OR Gate

Fig. 2.1 Symbols for Logic gates

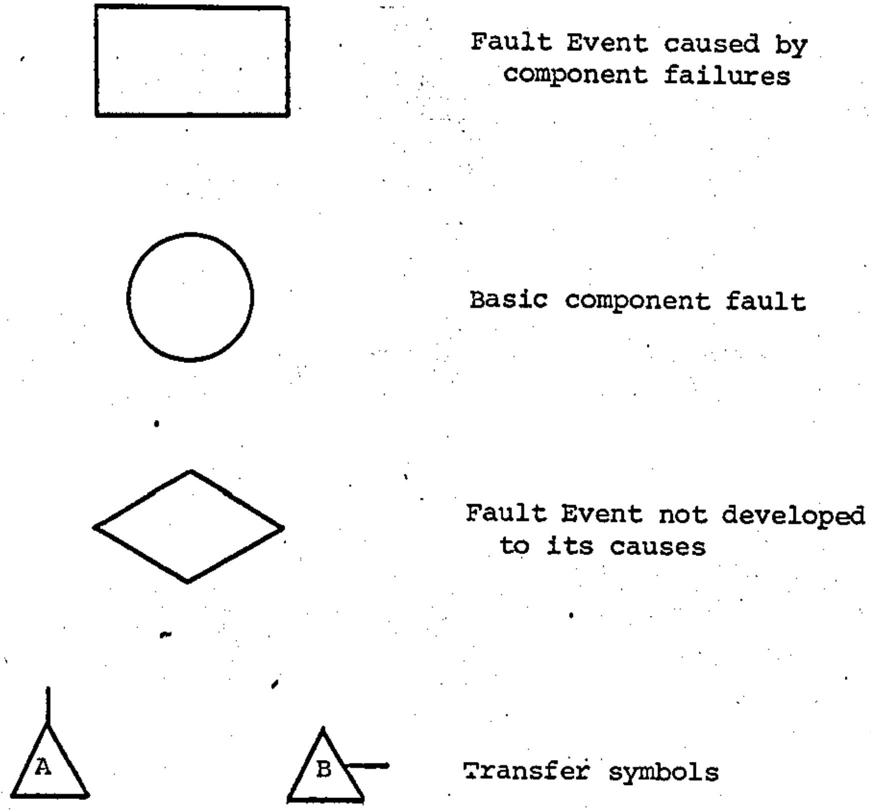

Fault Event caused by component failures
Basic component fault
Fault Event not developed to its causes
A
B
Transfer symbols

Fig. 2.2 Event symbols

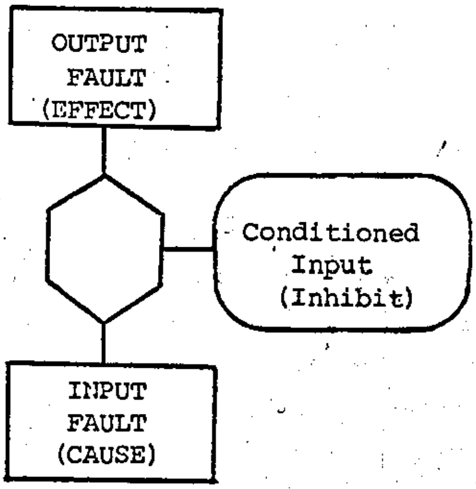


Fig. 2.3 INHIBIT Gate

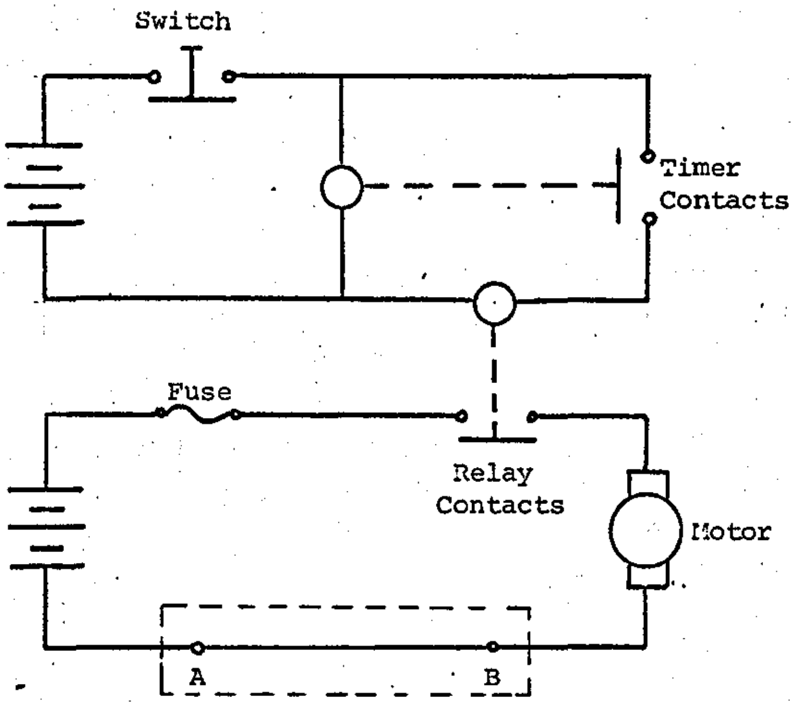


Fig. 2.4 A system designed to make available mechanical energy from the motor whenever the switch is closed by the action of an external control system

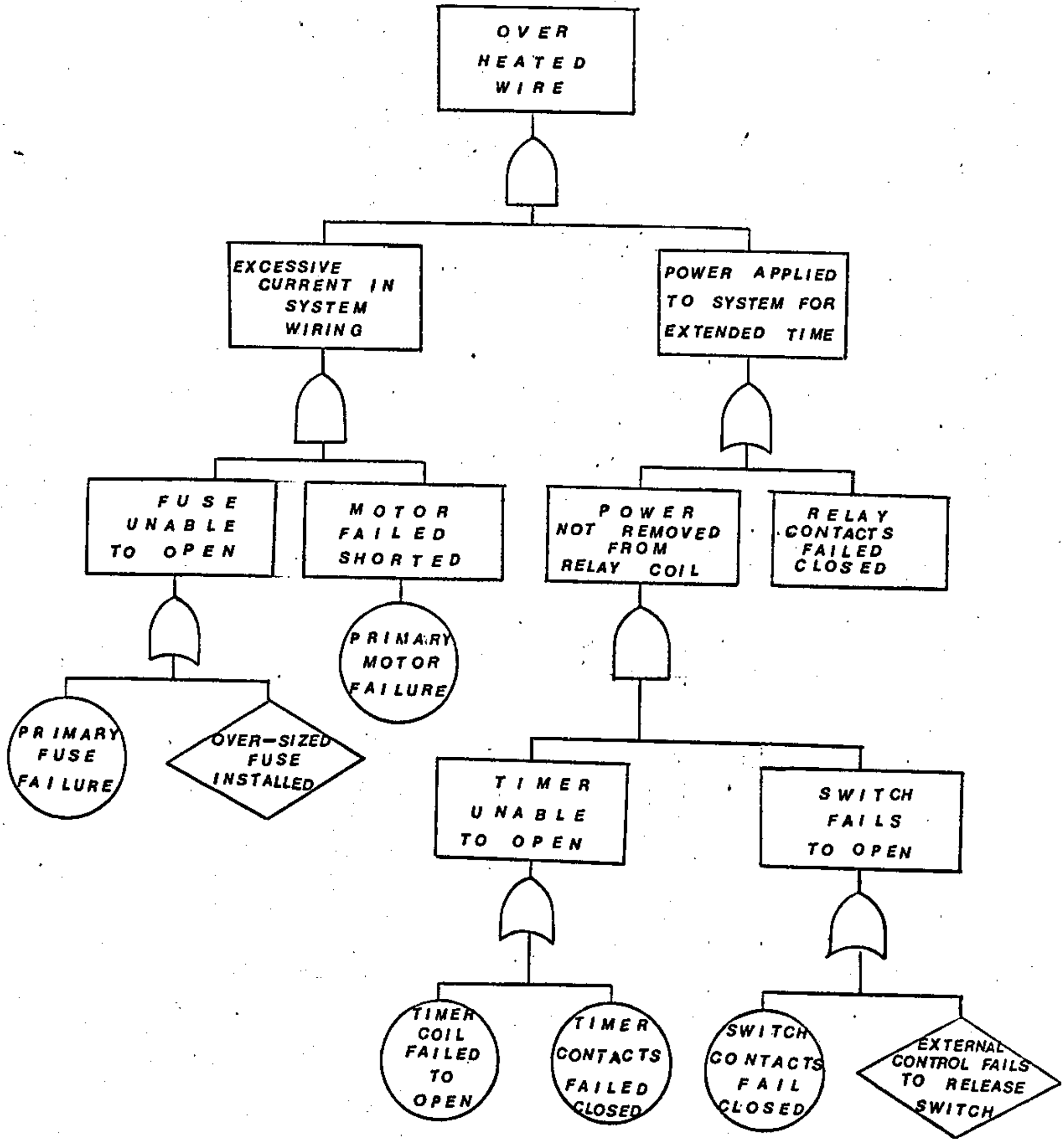

OVER HEATED WIRE
EXCESSIVE CURRENT IN SYSTEM WIRING
POWER APPLIED TO SYSTEM FOR EXTENDED TIME
FUSE UNABLE TO OPEN
MOTOR FAILED SHORTED
POWER NOT REMOVED FROM RELAY COIL
RELAY CONTACTS FAILED CLOSED
PRIMARY FUSE FAILURE
OVER-SIZED FUSE INSTALLED
PRIMARY MOTOR FAILURE
TIMER UNABLE TO OPEN
SWITCH FAILS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

Fig. 2.5 Fault tree with Primary Failure Technique

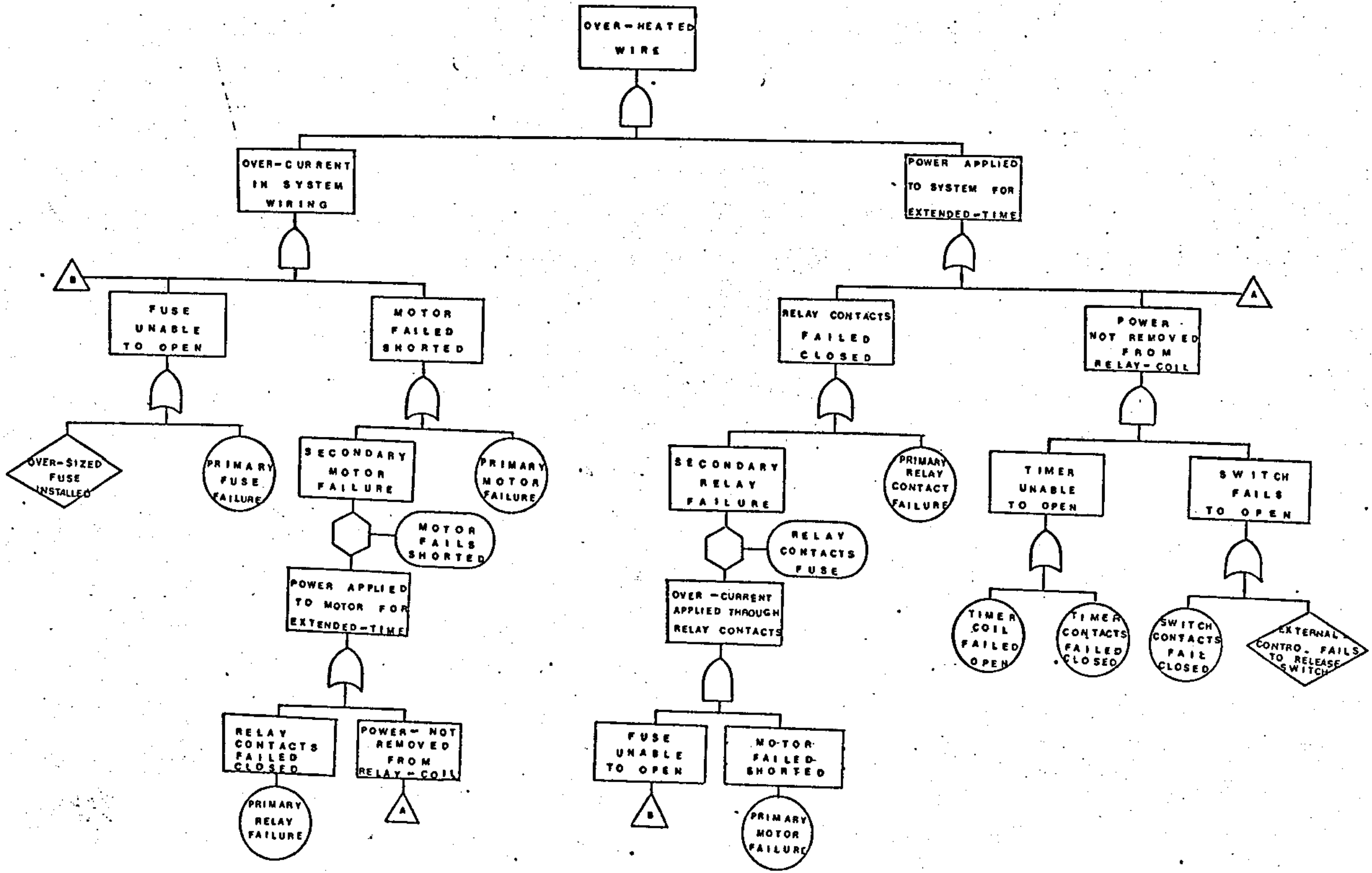
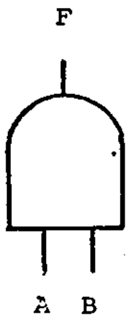
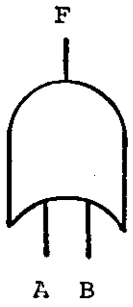

OVER-HEATED WIRE
OVER-CURRENT IN SYSTEM WIRING
POWER APPLIED TO SYSTEM FOR EXTENDED-TIME
B
FUSE UNABLE TO OPEN
MOTOR FAILED SHORTED
RELAY CONTACTS FAILED CLOSED
POWER NOT REMOVED FROM RELAY-COIL
A
OVER-SIZED FUSE INSTALLED
PRIMARY FUSE FAILURE
SECONDARY MOTOR FAILURE
PRIMARY MOTOR FAILURE
SECONDARY RELAY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

Fig 2.6 Fault tree with secondary failure technique



"F" exists when 'A' and 'B' exist

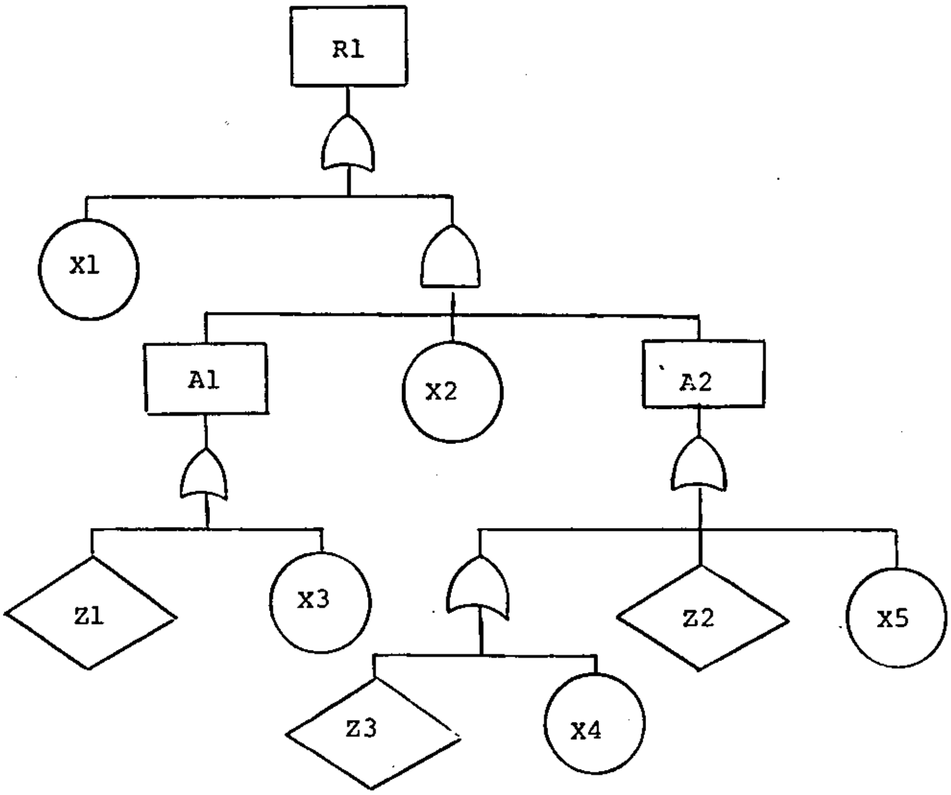
Boolean expression: $F=A*B$



"F" exists when 'A' and 'B' exist

Boolean expression: $F = A + B$

Fig. 2.7 "AND" and "OR" Boolean expressions



$R1 = X1+X2*(X3+Z1)*(X4+X5+Z2+Z3)$

Fig. 2.8 Boolean Expression

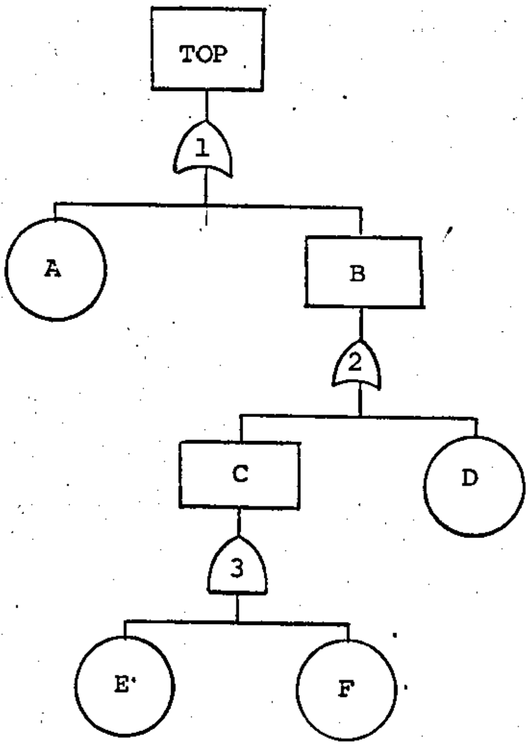


Fig. 2.9 Example Fault tree used to illustrate Fussell-Vesely min cut algorithm

			BICS	MIN CUT SET
A	A	A	A	A
(2) $\overrightarrow{\text{OR}}$ {	D	D	D	D
	(3) $\overrightarrow{\text{AND}}$	{EF	EF	EF

( )*gate

Fig. 2.10 Replacement of gates by basic events to obtain cut sets

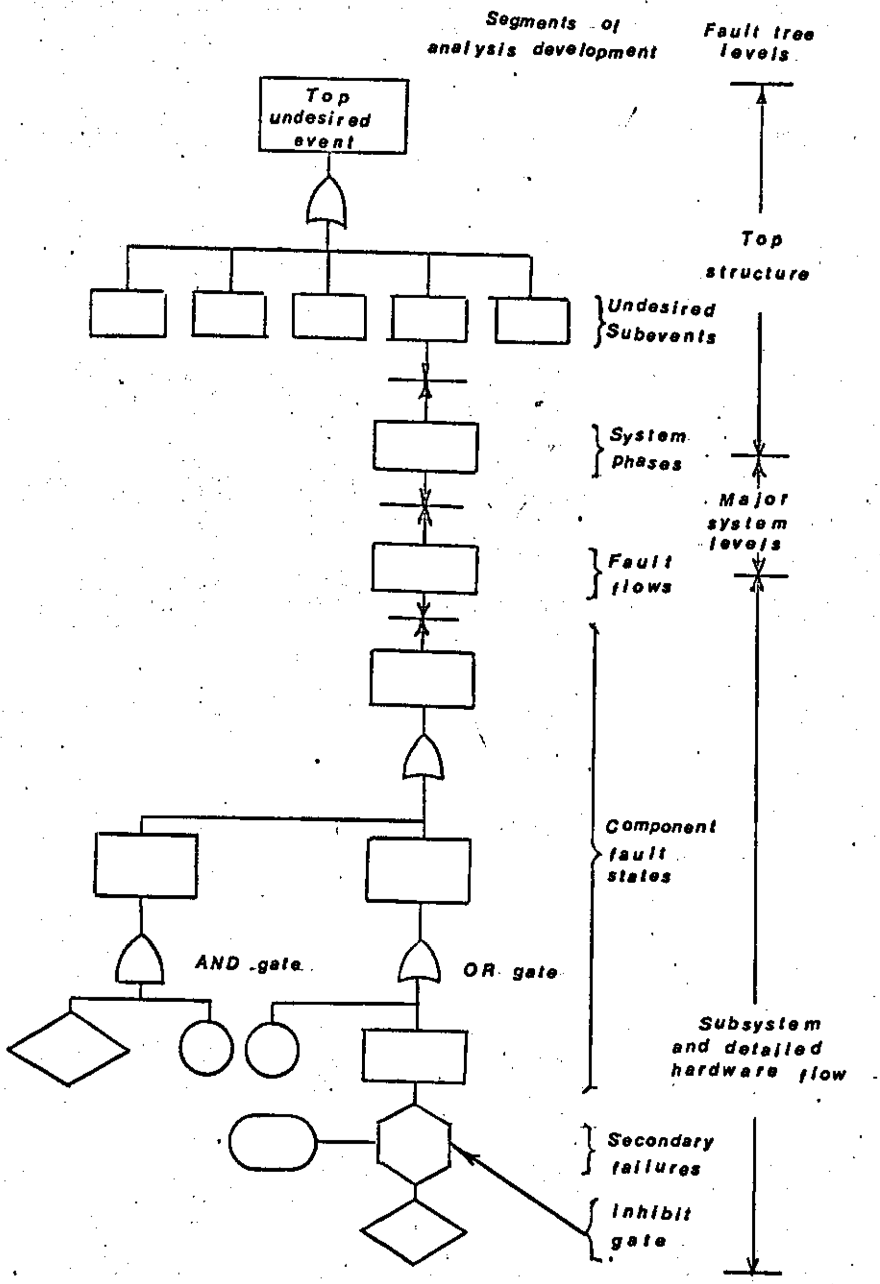


Fig. 2.11 <u>Haasl's levels of fault tree development</u>

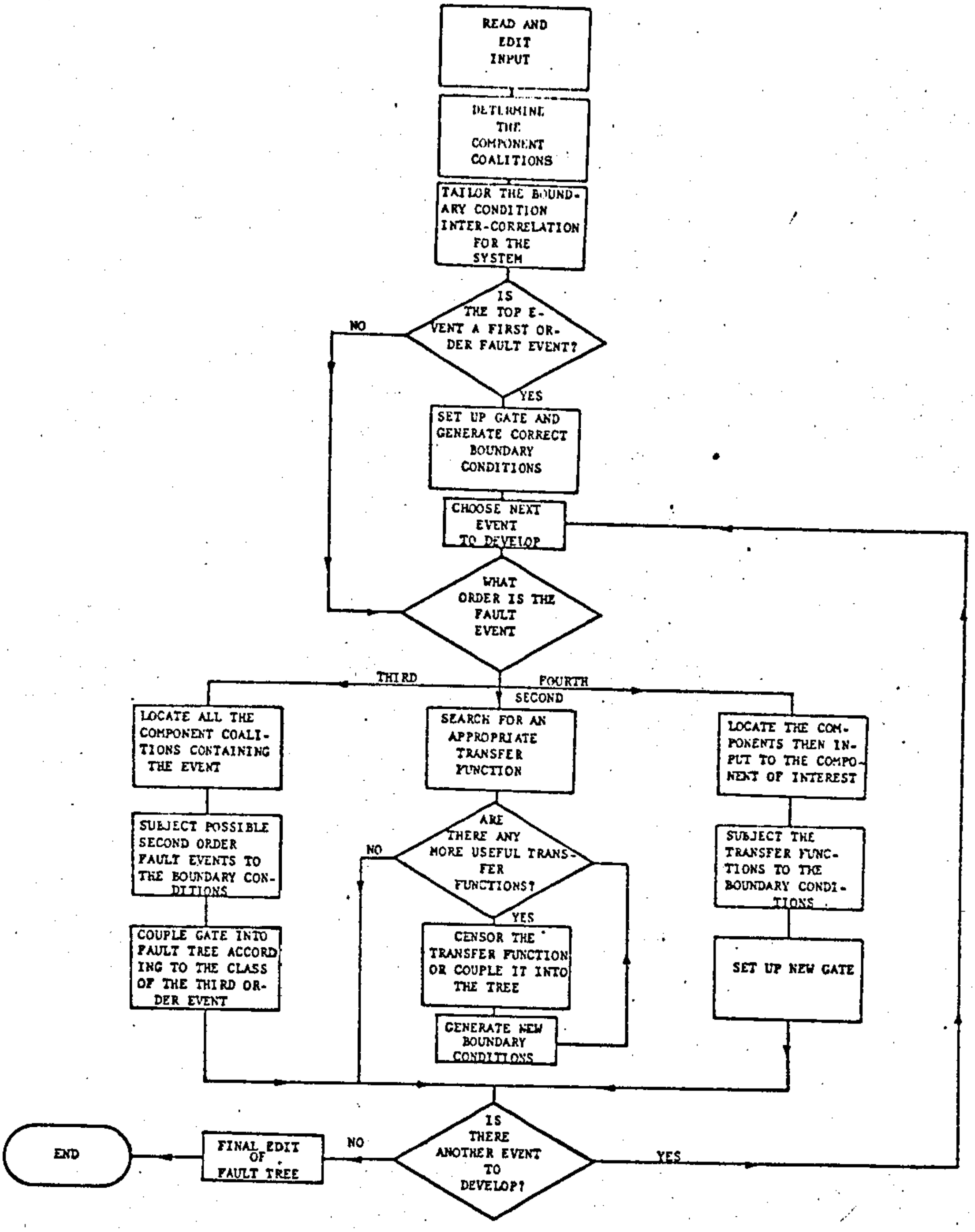

READ AND EDIT INPUT
DETERMINE THE COMPONENT COALITIONS
TAILOR THE BOUNDARY CONDITION INTER-CORRELATION FOR THE SYSTEM
IS THE TOP EVENT A FIRST ORDER FAULT EVENT?
NO
YES
SET UP GATE AND GENERATE CORRECT BOUNDARY CONDITIONS
CHOOSE NEXT EVENT TO DEVELOP
WHAT ORDER IS THE FAULT EVENT
THIRD
FOURTH
SECOND
LOCATE ALL THE COMPONENT COALITIONS CONTAINING THE EVENT
SUBJECT POSSIBLE SECOND ORDER FAULT EVENTS TO THE BOUNDARY CONDITIONS
COUPLE GATE INTO FAULT TREE ACCORDING TO THE CLASS OF THE THIRD ORDER EVENT
SEARCH FOR AN APPROPRIATE TRANSFER FUNCTION
ARE THERE ANY MORE USEFUL TRANSFER FUNCTIONS?
NO
YES
CENSOR THE TRANSFER FUNCTION OR COUPLE IT INTO THE TREE
GENERATE NEW BOUNDARY CONDITIONS
LOCATE THE COMPONENTS THEN INPUT TO THE COMPONENT OF INTEREST
SUBJECT THE TRANSFER FUNCTIONS TO THE BOUNDARY CONDITIONS
SET UP NEW GATE
IS THERE ANOTHER EVENT TO DEVELOP?
NO
YES
FINAL EDIT OF FAULT TREE
END

Fig. 2.12 DRAFT flow diagram

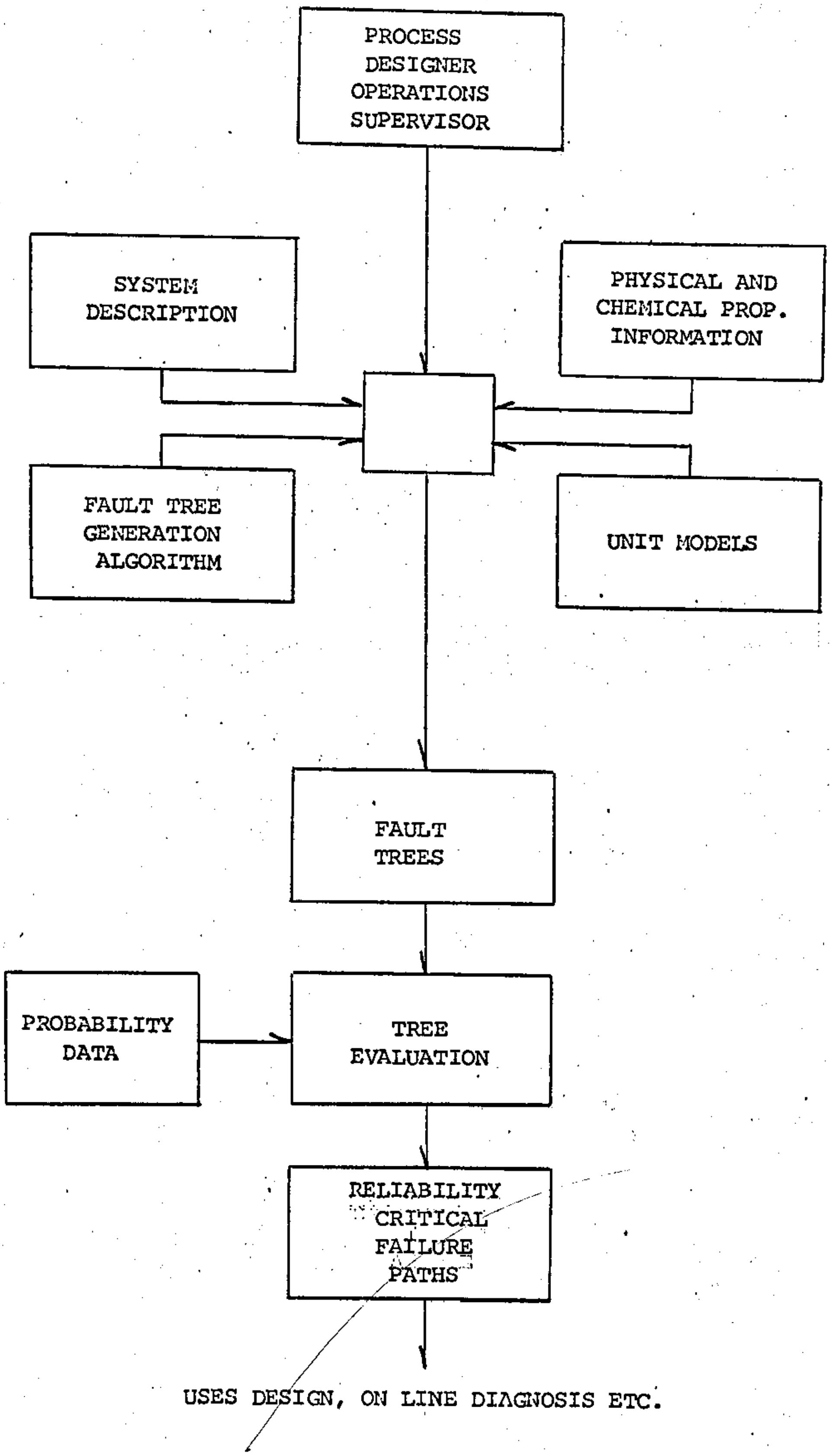


Fig. 2.13 Fault trees information requirements of Powers and Tompkins' methodology

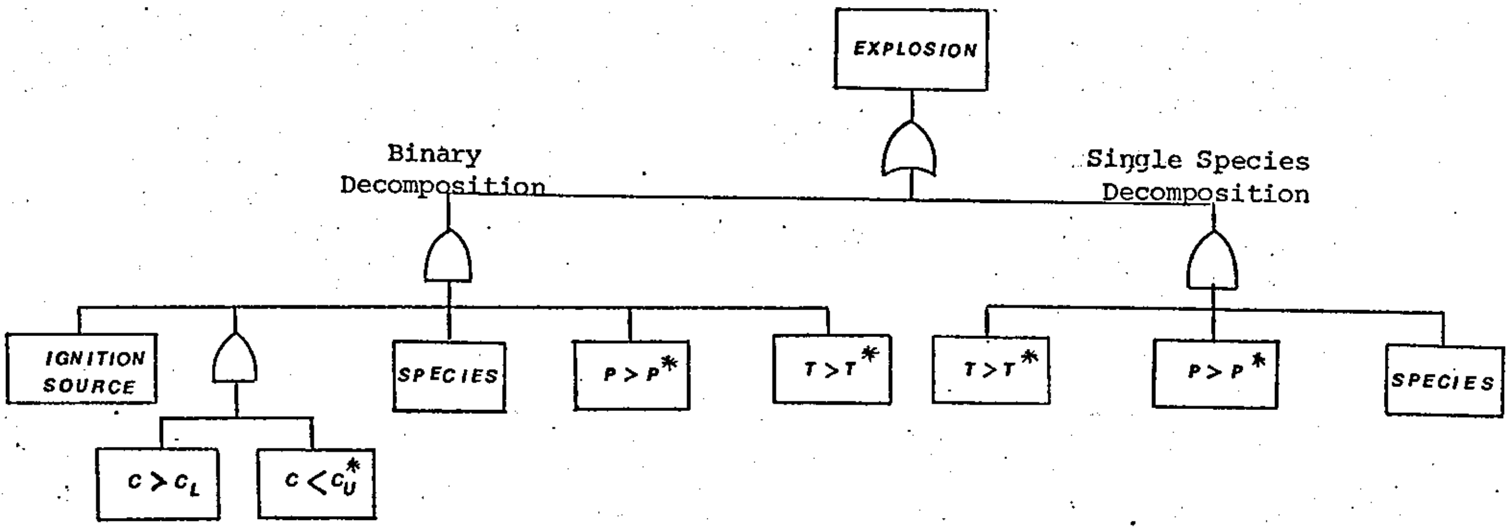


Fig. 2.14 Minifault tree for hazard state (explosion)

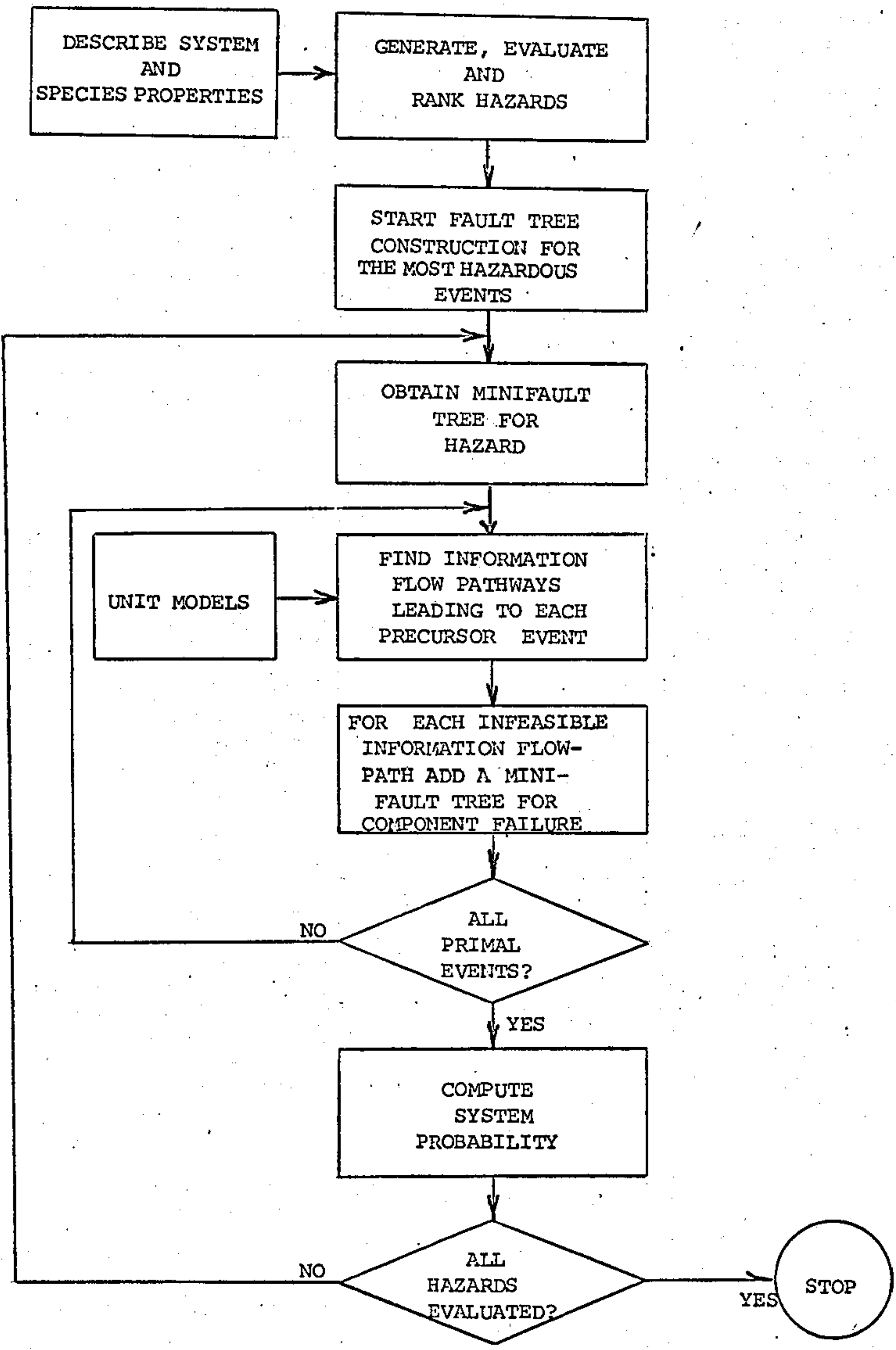

DESCRIBE SYSTEM AND SPECIES PROPERTIES
GENERATE, EVALUATE AND RANK HAZARDS
START FAULT TREE CONSTRUCTION FOR THE MOST HAZARDOUS EVENTS
OBTAIN MINIFAULT TREE FOR HAZARD
UNIT MODELS
FIND INFORMATION FLOW PATHWAYS LEADING TO EACH PRECURSOR EVENT
FOR EACH INFEASIBLE INFORMATION FLOW-PATH ADD A MINI-FAULT TREE FOR COMPONENT FAILURE
ALL PRIMAL EVENTS?
NO
YES
COMPUTE SYSTEM PROBABILITY
ALL HAZARDS EVALUATED?
NO
YES
STOP

Fig. 2.15 Strategy for Generating Event Trees

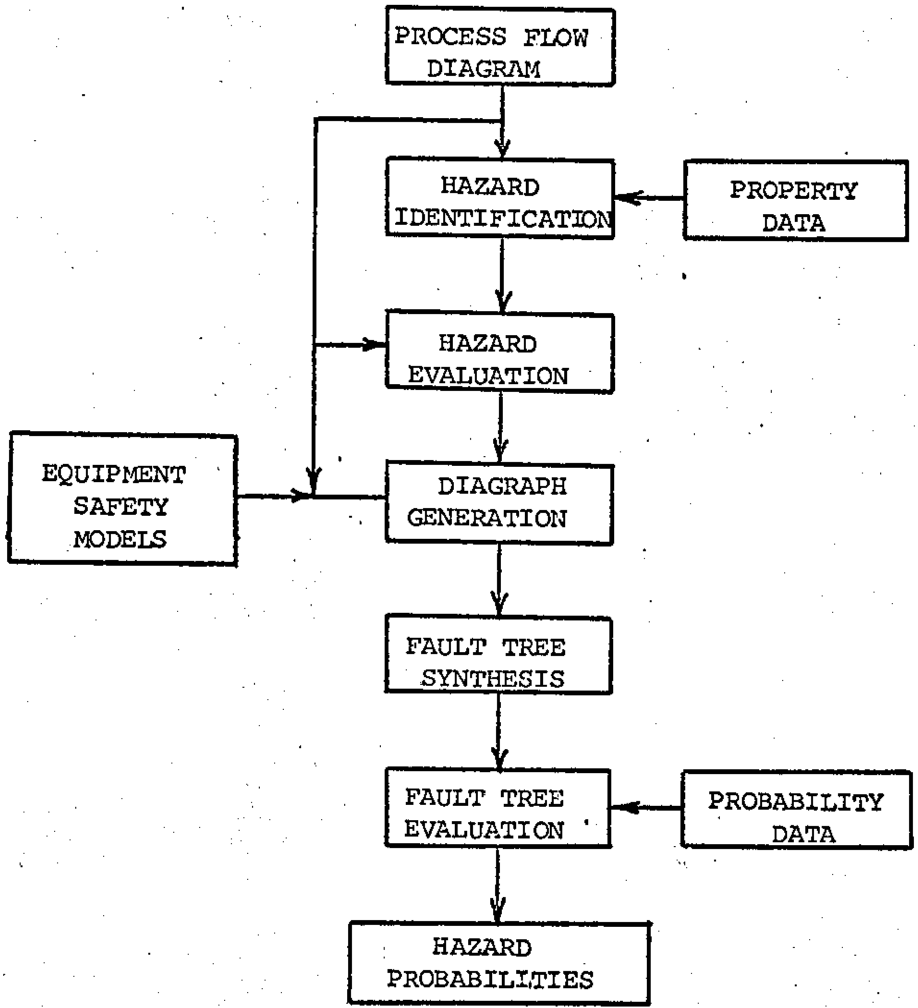


Fig. 2.16 The major steps to carry out a quantitative safety analysis of a chemical process proposed by Powers and Lapp

Chapter 3

UNIT MODELS AND UNIT MINITREES

This chapter is devoted to discussing the basis of the modelling method used in the methodology described in this work and the role played by the models in the construction of the minitrees.

3.1 Types of Plant Models

The first step before attempting the synthesis of a fault tree is to acquire a thorough understanding of the elements which form the system under study. For the case of a chemical process system the best way to acquire this knowledge is by means of a model of the plant under study. Plant models can be classified in several forms but for the purposes of this discussion they will be considered as follows:

1) Mental models. This is the model of the plant that an engineer has made in his mind. It requires a lot of consultations between the personnel involved in the plant and sometimes may be inaccurate. This sort of model is not suitable for the purposes of this work.

2) Functional models. These models are based on the knowledge that a particular variable is the function of several others. The function is undefined for this type of model. The functional model is the loosest form of equation that may be conveniently expressed on paper. No quantitative information is associated with this model.

3) Enhanced functional models. These models are similar in form to functional models but also contain directional information.

4) Statistical Models. These models are based on regression and correlation of data accumulated by logging plant variables either during commissioning or on a similar plant elsewhere.

5) Full equation models. This type of model is more difficult to derive since it requires an accurate conception of the dynamics of the system. The full model will consist of a set of simultaneous differential and algebraic equations. The number of equations is likely to be large and hence the time required for model production is likely to be excessive.(A2)

Each type of model mentioned above has advantages and disadvantages. The functional or the enhanced functional models require considerably less effort to assemble than the full equation or statistical models. The disadvantage of the functional or the enhanced functional models is that, by their nature, they are not suitable for simulation of plant behaviour.

3.1.2 Unit model approach

A way to facilitate the understanding of the plant can be by looking at it as a set of "units" linked together. This approach has the advantage that it focuses the analyst's attention on a manageable portion of the problem and allows him to acquire a better understanding of the unit. This approach also enables the analyst to produce a library of models for use as required rather than models that only reflect particular plants. Some units might have several models with different degrees of complexity. This implies that the model should be general in nature and respond in the correct manner to a variety of external stimuli.

3.2 Formulation of Models

The unit model approach was used to formulate the plant models. This was decided because of the following reasons:

1) The model for each unit may be based on the equations of the familiar "unit operations" commonly found in text books. Each model is defined by a name and a series of input and output streams linking it to other units.

2) Each individual model may be tested by supplying "dummy units" for each input and output stream. In this way the response of the unit to various process disturbances may be found.

3) Simple models of units may be used initially and may be replaced by more complex ones where necessary.

4) A library of the most common units may be built up. This is ideally suited for saving memory when the models are used in computer programs.

5) It allows a great flexibility when modifications have to be made to the plant; units can be deleted or added without problem to the model.

3.2.1 Conventions used with the models

The models used in this work are general purpose models. These models represent the individual process units and give the relations between the process variables. The process units are linked together by means of streams, which carry the information flow of the process variables to other units.

In order to preserve continuity in process streams passing through several units, use is made of high-gain differential equations to set intermediate stream pressures. The requirements for this type of equation arises because conventional equations for liquid flow generally assume an incompressible fluid, and in this work compressibility is needed to assure the flow of information of the process variables upstream and downstream of the process units. The need for high-gain equations is discussed by Franks.(F9)

Since information must flow in both directions it follows that some unit model variables must be set in the input streams and some in the output streams. The convention adopted in this work is:

1) The pressure variable is set in model input streams.
2) All other variables are set in model output streams.

The example shown in Fig. 3.1 helps to illustrate the convention. It shows a pipe section (the simplest plant model) with liquid flow. Equation 3.1(a) is sufficient to transmit information concerning changes in P_{IN} and P_{OUT} to Q_{OUT}. Equation 3.1(b) is the other relation needed to bring in the affect of changing Q_{IN}, this high-gain differential equation reflects the lag induced by liquid compressibility.

These two equations transmit information in opposite directions through the unit, reflecting the way in which pressure transients propagate. This concept of information flow is important for general models and for the purpose of this work. For the example shown in Fig. 3.1 only two properties were considered in the model, but temperature, concentration etc., can also be included without problem by applying the convention adopted.

In certain cases, the strict use of the convention may lead to unnecessarily complicated models. For those cases the convention may be relaxed but, if the model is intended to be used as part of a library it should be marked as unsuited for general use. The use of high-gain equations with this sort of convention and cases when the convention is relaxed have been discussed by Andow.(A2)

3.2.2 Models in enhanced functional form

Another way of writing the unit models may be in the enhanced functional form; this form reflects only the way in which the variables affect each other in qualitative, but directional form only. The proportionality constants of the more familiar algebraic form are not needed for this form. The enhanced functional model may be thought of as an engineer's word model and hence is easily obtained for common items. Equations 3.1(c) and 3.1(d) show the functional form for the pipe model in the form of a network of information paths, (see Fig. 3.2).

3.3 The Use of Unit Models to Generate Minitrees

Once the information of each of the elements of the system has been collected it is possible to start thinking about the construction of fault trees for the system. The models can be considered as performance models for each unit, these models are now used to find out how each of the units can fail. This is done by means of a failure analysis for each model. In this work it is carried out manually.

3.3.1 Failure analysis based on the unit models

The aim of the failure analysis is to obtain a set of minitrees for each unit model; these minitrees will reflect all the different ways by means of which the unit variables may fail. The minitree can be considered then, as a failure model of the unit under analysis.

The failure mode analysis for each unit model is carried out, by picking out each of the Left Hand Side (LHS) variables of each equation of the unit model under analysis. The top event of each minitree is defined on the basis of the different failure states that a specific LHS variable can have, (e.g. P_{HIGH}, P_{LOW} for Pressure, etc.). Once the top event is defined the tree for that particular event is developed and this is done by considering the different ways by means of which the fault event can be produced. At this point the analyst needs to use the best of his knowledge about the unit under study, so he can define the type of gate needed by the top event and the inputs to the gate. To help in the decision, use is made of the right hand side variables of the equation being considered. These variables transmit the stimuli due to external faults. Internal faults which from experience are known to be possible causes of the top event are also considered. The failure analysis for a unit model ends when all the failure states required for each LHS variable of the model have been considered.

For the purpose of this work, it will be assumed in all the analyses, that the system was constructed with no components installed that do not meet the specifications for which they were designed.

3.3.2 Nomenclature used in the minitrees

Most of the fault trees nomenclature that will be used in this work has already been mentioned in Chapter 2; however, due to the nature of the methodology it was necessary to introduce a particular nomenclature to be used when generating the minitrees. This section is devoted to the definition of the different types of events and nomenclature used to develop the minitrees.

3.3.2.1 Types of events

The types of events are classified as follows:

1) Main Event (M)

This is the top event of each minitree, it was decided to call it "main" instead of top event in order to make a distinction between the top event in a tree and the top event of a minitree. It will be represented as a rectangle. This type of event has always a gate associated with it.

2) Transmissive Event (T)

The failure states of variables that transmit the stimulus due to external faults are called transmissive events. This type of event always requires further development. It will be represented as a circle within a diamond.

3) Basic Event (B)

An event that does not require further development is called a basic event. It will be represented as a circle.

4) Replaced Event (R)

During the development of the minitrees idea for this work, it was noted that some of the minitrees had several gates which made them too long. The cause of this long minitree was the presence of some non-basic events and non-transmissive events.

Bearing in mind that the aim of the minitrees was to use them as a source of information to construct fault trees, it was thought that this information should be easily stored and retrieved. Long minitrees were not very suitable for this purpose. It was necessary to have small minitrees and avoid the repetition of the same information in a particular set of minitrees, and at the same time keep the completeness of information provided by the long minitrees. To solve the problem it was decided that all the minitrees should have only one gate and that the flow of information should be maintained by means of Replaced events.

A Replaced event is defined as an event that requires further development, but the information required to develop the event is found among the set of minitrees of the same unit. It will be represented in this work as a rectangle within a diamond. The use of Replaced events will be described later in this chapter.

3.3.2.2 Types of gates

The gates mainly used in the construction of the minitrees are the fundamental AND and OR gates. A special gate could be useful sometimes and is included in this section for completeness. This is the Exclusive OR gate or EX-OR gate. This gate performs the same function as the OR gate, with the restriction tht certain specified inputs cannot coexist.(E1) If one input event occurred, thus causing the output event to occur, and then the other event occurred, the output event would then cease to exist.

The symbology used in building the minitrees is shown in Fig.3.3.

3.3.2.3 Boundary Conditions and Not-allowed faults

Once the failure analysis is carried out and before considering the minitrees ready, there is a need to assure its consistency. This means to make sure that when the minitree is used in the construction of the tree, nothing could arise below the top event of the minitree in such a way that a contradiction existed.

To do this use is made of Boundary Conditions and Not-allowed faults. For example, consider Q_{OUT} HI to be the top event for one of the minitrees of the pipe model mentioned before. The Boundary Conditions and Not-allowed faults for this minitree are Q_{OUT} LO and Blockage. They will affect all the branches under the top event so that no contradiction could arise when developing the top event Q_{OUT} HI. The Boundary Conditions and Not-allowed faults will be represented attached to the gates of the minitrees. The importance of the Boundary Conditions and Not-allowed faults will be demonstrated in the fault tree generation algorithm.

3.3.3 Construction of the minitrees and use of Replaced events

The best way to illustrate this point is by means of an example. Bearing in mind the points mentioned before, consider again the pipe model shown in Fig. 3.1. From equation 3.1(a) the following failure states can be obtained:

LHS varaiable: Q_{OUT}

Failure states: Q_{OUT}(HI), Q_{OUT}(LO)

Writing the failure states as top events two minitrees can be developed. The trees for these failure events are shown in Figs. 3.4 and 3.5. Note that use has been made of the right hand side variables involved in the equation and of basic events such as blockage and leakage events that from experience are known to be possible causes of the top event.

Before considering the minitree ready the Boundary Conditions and Not-allowed faults have to be stated. They are part of the minitree as well and are represented attached to the gate of the minitree. Boundary Conditions and Not-allowed faults affect all the branches of the minitree. For the top event Q_{OUT}(HI) the Boundary Condition is Q_{OUT}(LO) and the Not-allowed fault is Blockage. For the top event Q_{OUT}(LO) the Boundary Condition is Q_{OUT}(HI). Once the Boundary Conditions and Not-allowed faults are stated the minitrees are ready to be used in the construction of fault trees.

The failure analysis for the unit is finished when all the equations have been considered. In this example there is still another equation to be considered for the pipe model. Looking now to equation 3.1(b) and following the same procedure another two minitrees are generated. They are shown including its Boundary Conditions and Not-allowed faults in Figs. 3.6 and 3.7.

Consider now the example of a control valve. The unit model is shown in Fig. 3.8. This model is a good example to illustrate the failure analysis and the use of the Replaced events. In this model the conventions stated for the models are relaxed, so P_C is set at the controller output. By relaxing the conventions the model has been simplified so the mass flow of air to the control valve and its effects on pressure are ignored.

To start the failure analysis consider first equation 3.2(a). The failure states Q_{OUT}(HI) and Q_{OUT}(LO). The minitree for Q_{OUT}(HI) without Replaced events is shown in Fig. 3.9. The minitree for Q_{OUT}(HI) shown in Fig. 3.9 has three gates, the minitree in itself is correct, but for the purposes of the methodology is too long. It is for these kind of minitrees that the Replaced events were defined. It will be necessary to split the minitree so only one gate remains in it. In this case there will be two new minitrees and this will imply the creation of two dummy faults, one for each Replaced event. It can be said that there are three domains in the minitree of Fig. 3.9 and for our purposes only the first domain can remain in the Minitree. Fig. 3.10 shows the new minitrees with the Replaced events. Note that the Boundary Conditions and the Not-allowed conditions of the original minitree affect the new minitrees as well. This will be more evident when the fault tree generation algorithm is explained in the following chapters.

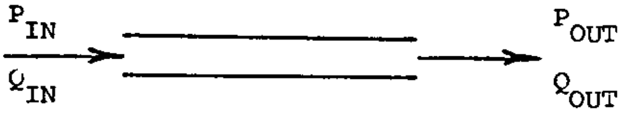
Following the same procedure the minitree for Q_{OUT}(LO) is obtained and it is shown in Figs. 3.11 and 3.12. Note that the use of Replaced events helps to avoid repetition of some branches of the minitrees, in this example the minitree for the dummy fault C is the same in Figs. 3.10 and 3.12. Therefore repetition of information may be avoided in the set of minitrees for the control valve. The full set of minitrees for the control valve obtained by following the procedure established, together with the set of minitrees and models of other units used in this work are presented in Appendix I. Note that for the examples used in this chapter only two states of the variables have been considered, but the same procedure can be used for any other state of the variables that needed to be considered.

The different steps to obtain the set of minitrees for any unit model are summarised in Fig. 3.13.

General purpose model

Flow rate: $$Q_{OUT} = k(P_{IN}-P_{OUT})^{\frac{1}{2}} \qquad 3.1(a)$$

Continuity: $$\frac{dP_{IN}}{dt} = G(Q_{IN}-Q_{OUT}) \qquad 3.1(b)$$



Functional form model

$$Q_{OUT} = f(P_{IN}^{\;+}, P_{OUT}^{\;-}) \qquad 3.1(c)$$

$$\frac{dP_{IN}}{dt} = f(Q_{IN}^{\;+}, Q_{OUT}^{\;-}) \qquad 3.1(d)$$

Fig. 3.1 Pipe model

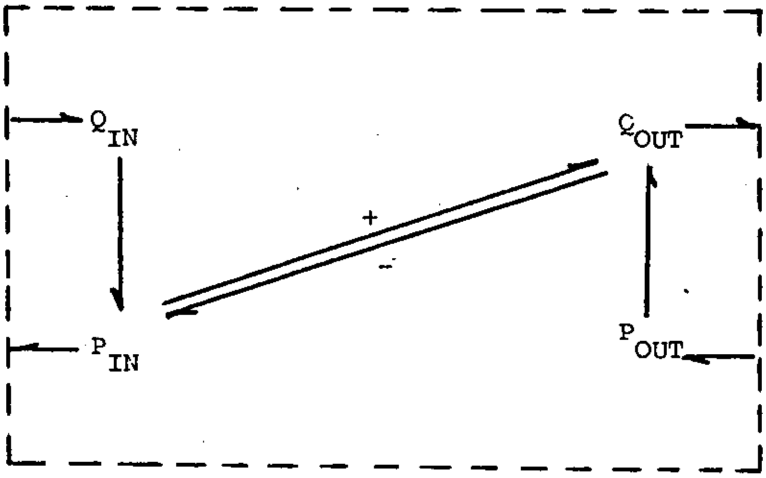


Fig. 3.2 Information flow diagram for pipe model in functional form

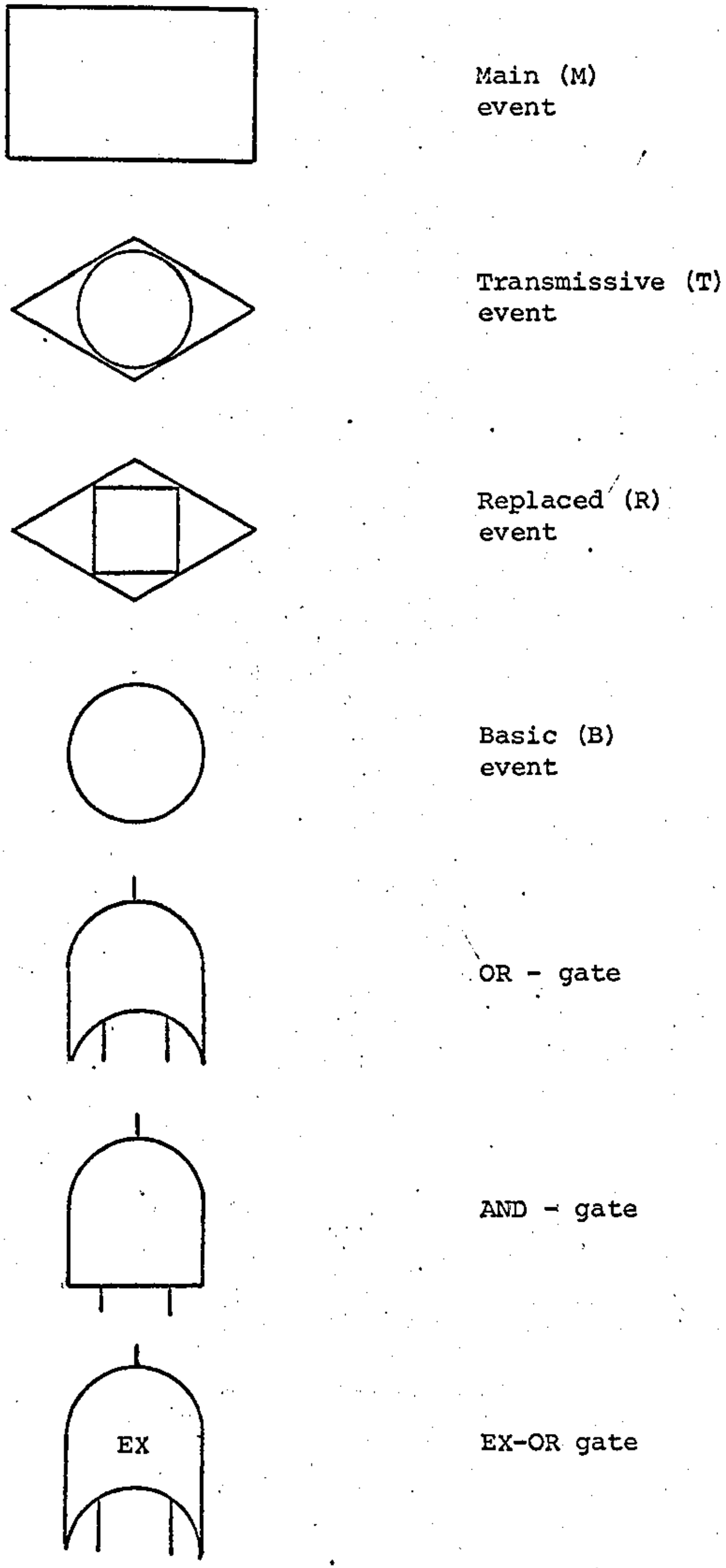


Fig 3.3 Symbols used to represent minitrees

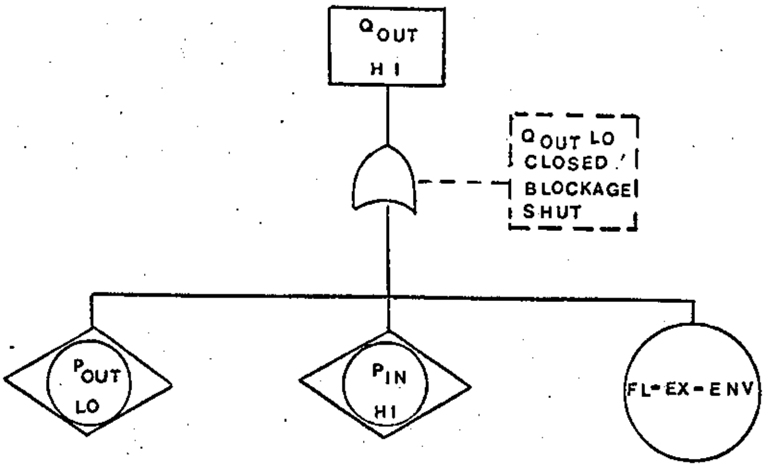


Fig. 3.4 Minitree of Q_{OUT} HI for a Pipe

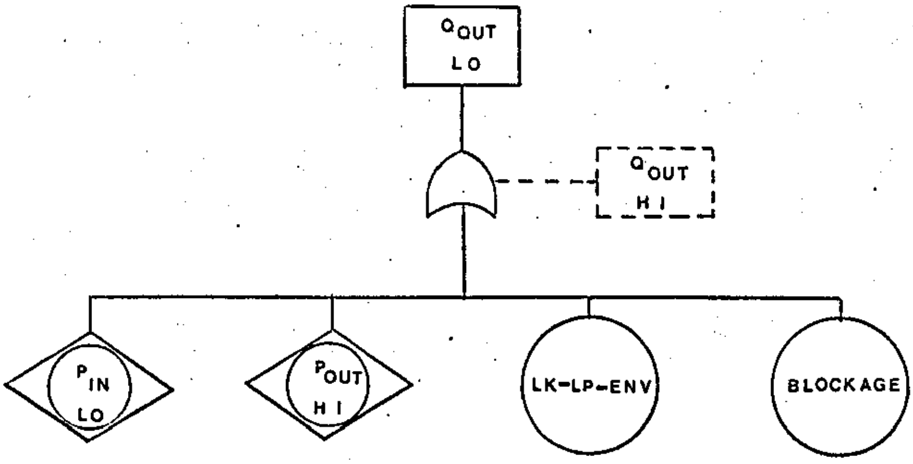


Fig. 3.5 Minitree of Q_{OUT} LOW for a Pipe

<u>LHS variable:</u> P_{IN}

<u>Failure states:</u> P_{IN}(HIGH), P_{IN}(LOW)

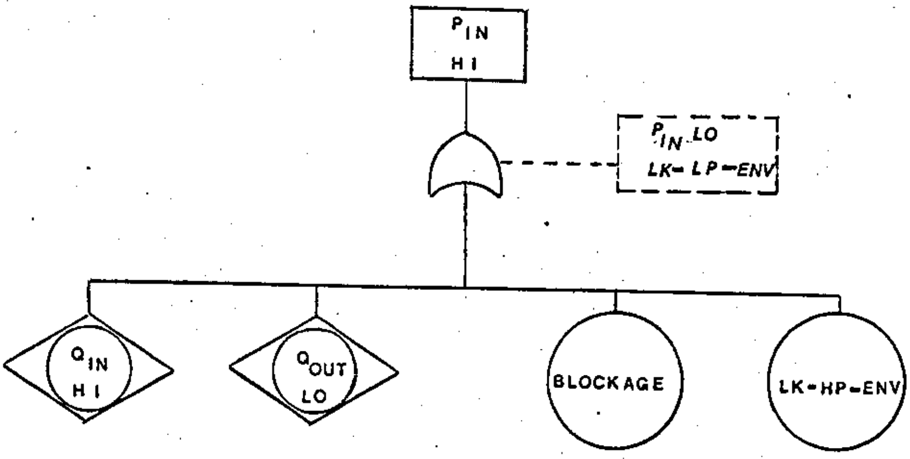


Fig. 3.6 <u>Minitree of P_{IN} HI for a Pipe</u>

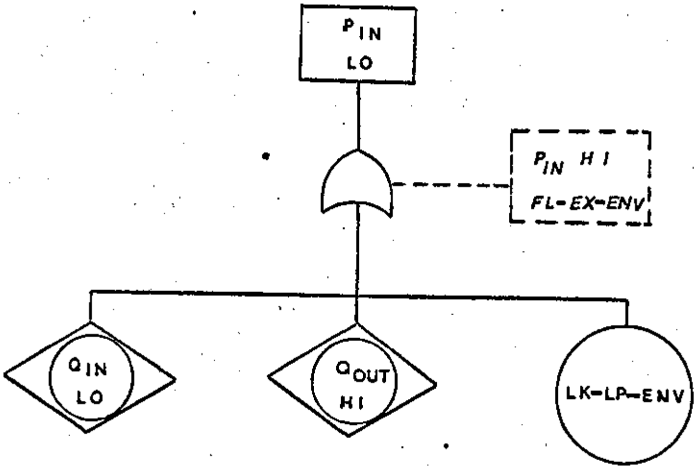
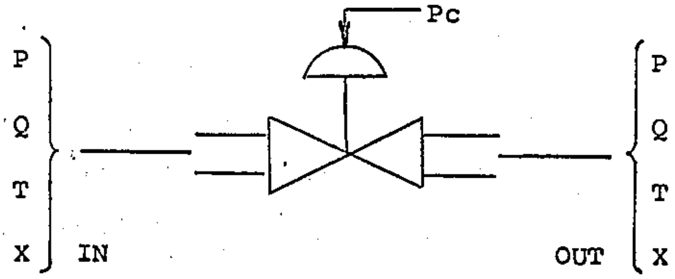


Fig. 3.7 <u>Minitree of P_{IN} LOW for a Pipe</u>



Equations:

Flow rate: $Q_{OUT} = k_v f(B_{IN})(P_{IN}-P_{OUT})^{\frac{1}{2}}$ 3.2(a)

Continuity: $\frac{dP_{IN}}{dt} = G(Q_{IN}-Q_{OUT})$ 3.2(b)

Temperature: $T_{IN} = T_{OUT}$ 3.2(c)

Concentration: $X_{IN} = X_{OUT}$ 3.2(d)

Fig. 3.8 Control valve model

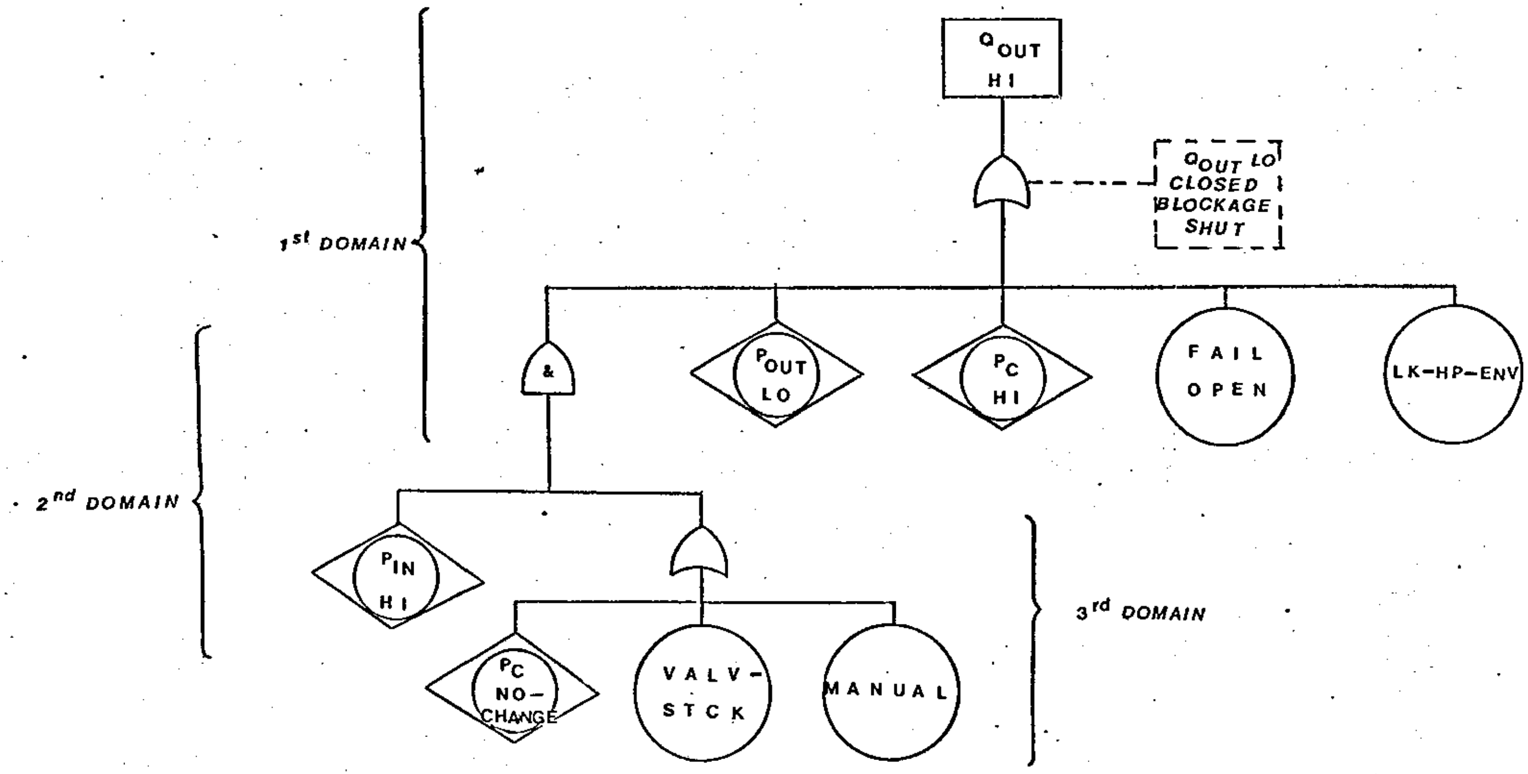


Fig. 3.9 Minitree of Q_{OUT} HI for control valve (without "R" events)

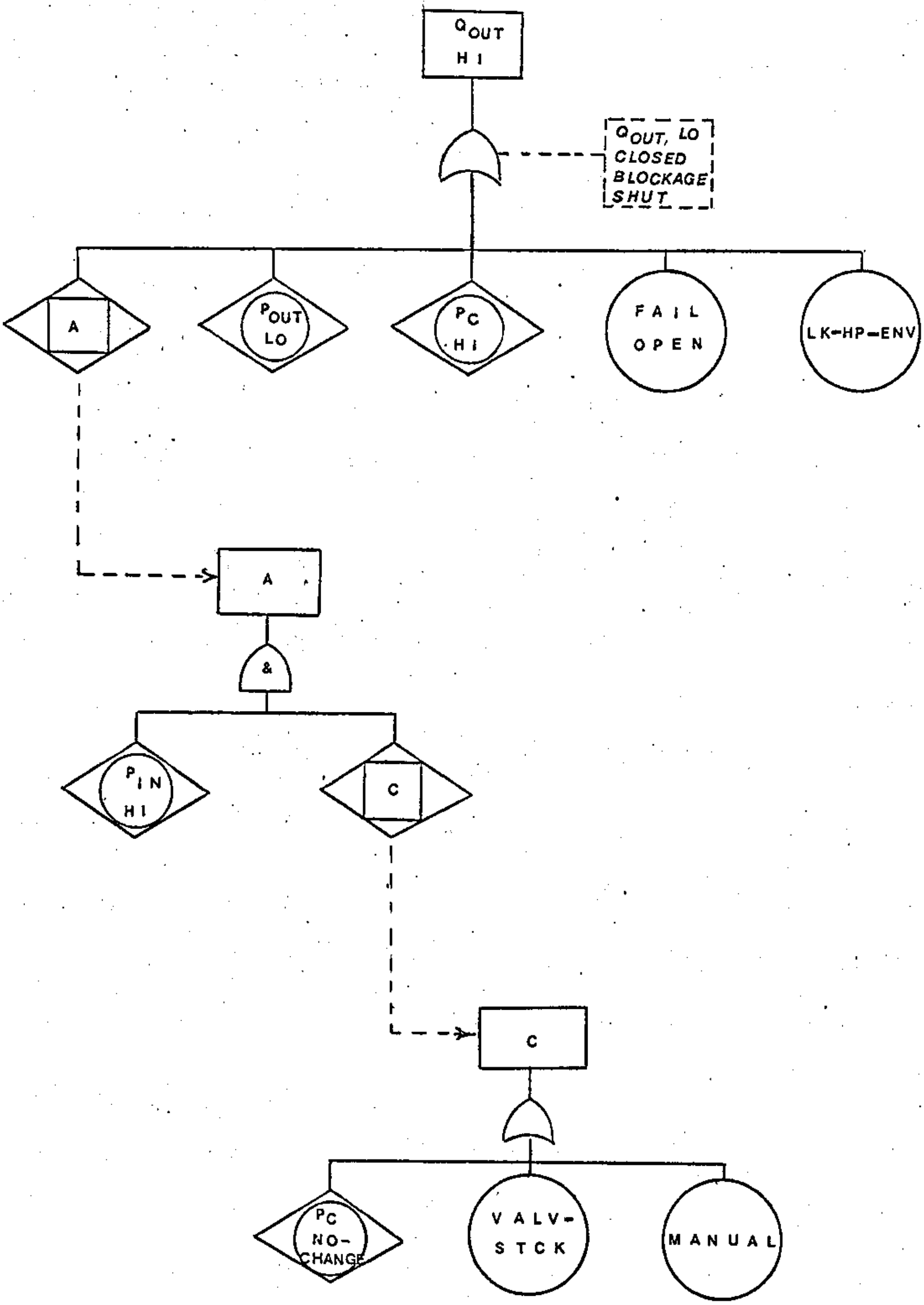


Fig. 3.10 Minitree of Q_{OUT} HI for control valves (with "R" events)

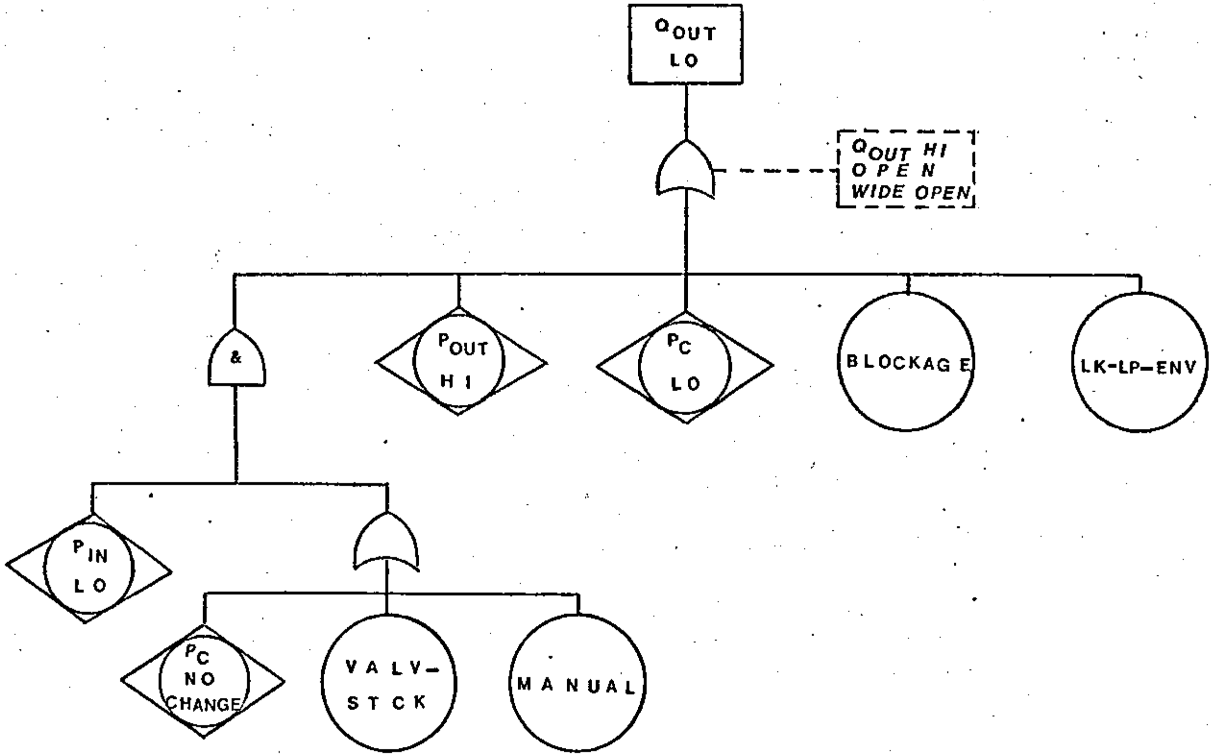


Fig. 3.11 Minitree of Q_{OUT} LO for control valve (without "R" events)

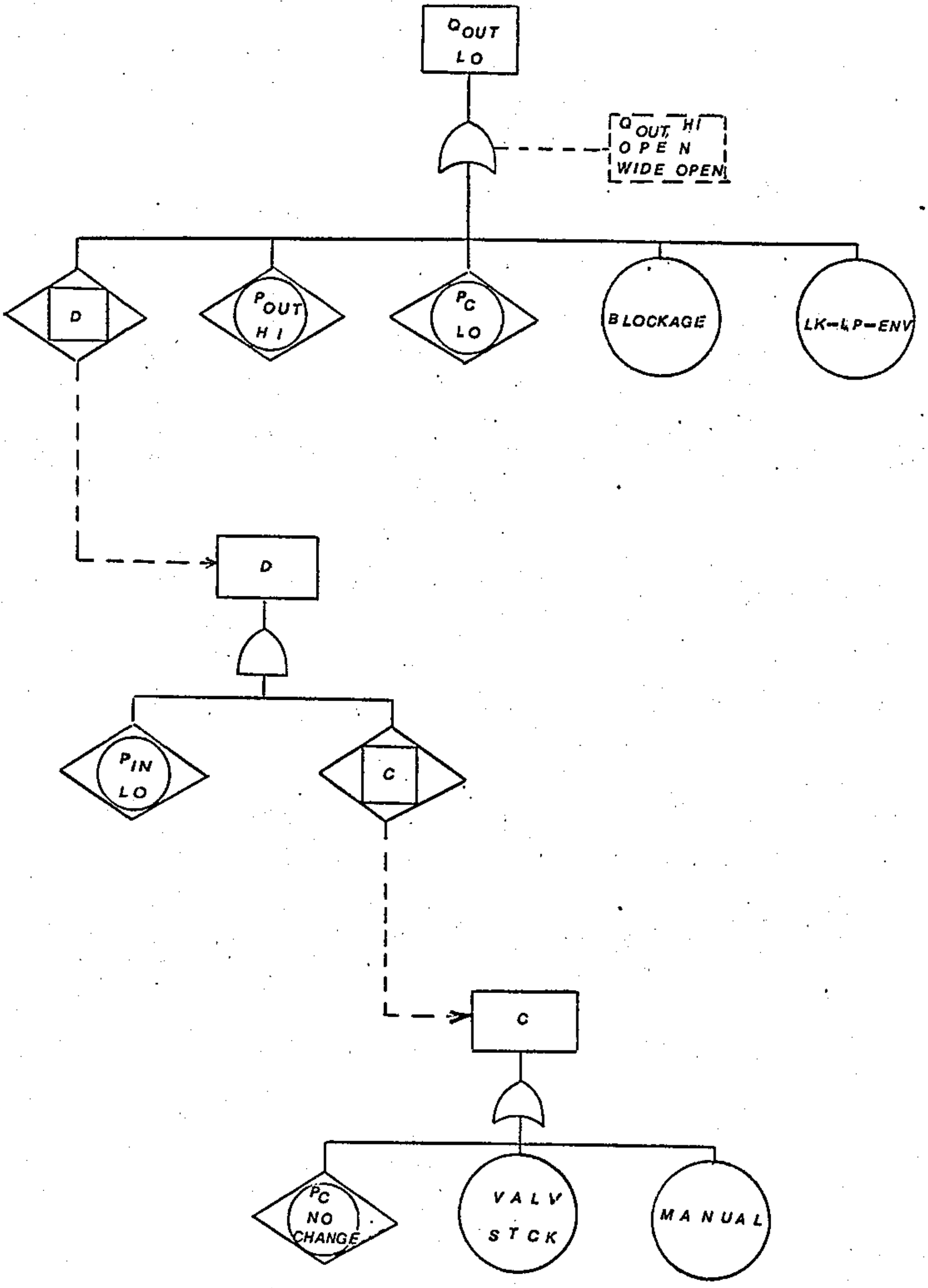


Fig. 3.12 Minitrees of Q_{OUT} LO for control valve (with "R" events)

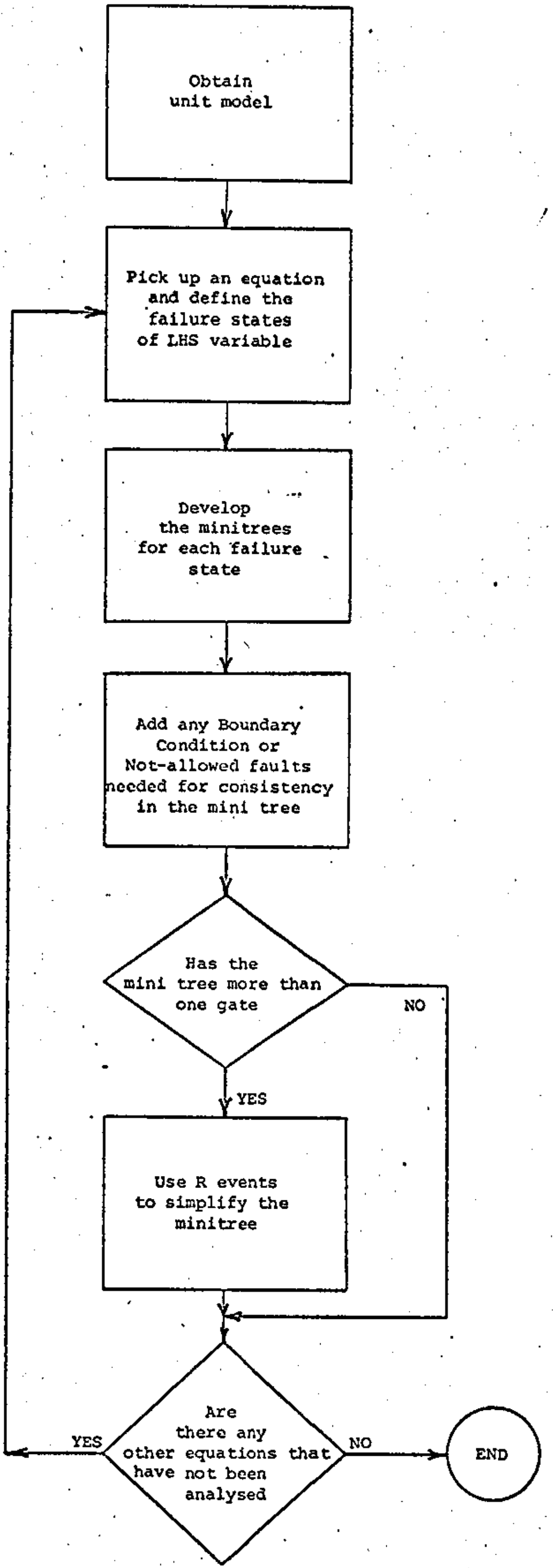

Obtain unit model
Pick up an equation and define the failure states of LHS variable
Develop the minitrees for each failure state
Add any Boundary Condition or Not-allowed faults needed for consistency in the mini tree
Has the mini tree more than one gate
NO
YES
Use R events to simplify the minitree
Are there any other equations that have not been analysed
YES
NO
END

Fig. 3.13 Steps to obtain the set of minitrees for a unit model

Chapter 4

USE OF THE UNIT MINITREES TO CONSTRUCT FAULT TREES FOR A SYSTEM

In the previous chapter the unit models and the method of derivation of the minitrees were discussed. Once the unit minitrees are obtained, they can be used anytime. The specific unit appears in a system, with the confidence that it will transmit what is happening in the unit to the other units of the system. In the case of a fault that had occurred in another unit, the unit would react to it and would transmit it to the units linked to it if appropriate.

The units and their minitrees will be used in this chapter as the basis of the fault tree synthesis methodology. Since the unit model equations have been produced in a consistent manner the resulting minitrees will also be "plug compatible" with each other. Once the top event in a system is defined, it is now possible, by means of interlinking the minitrees of each unit model of the system, to trace the possible causes in each unit which could lead to the top event.

This chapter is devoted to describing how the unit minitrees are linked together, to produce a fault tree for a specified top event in a system. This systematic linking process is the basis of the fault tree generation algorithm that will be described in the following chapters.

4.1 Topology of the System Under Study

The topology of the system under study is of major importance when the flow of information is required in generating the fault trees. It is by means of the topology that is possible to know how the units are linked, which are the input and output streams of each unit and which are the variables involved in each stream. The flow of information plays a vital role in the transmission of faults through the system.

The topology of the system under study can be depicted with the help of a flow sheet diagram. The aim is to assemble the complete plant model, using the individual unit models obtained by means of the process described in Chapter 3.

The first step will be to identify each individual unit on the flow sheet diagram and the process streams linking the units together. For this purpose each unit is given a unique name consisting of the name of the unit and a number, e.g. Pipe 1, Heat exchanger 3, Valve 9 etc. Each input and output stream of the unit being considered is identified by means of the names of the variables that according to the unit model are associated with each stream. By classifying the process streams linking a unit with its neighbours as "inputs" or "outputs" rather than as streams, simple checks can be made because for every input stream an output stream must exist on other units. It is also consistent with the conventions adopted in Chapter 3 for setting the unit model variables in input or output streams, depending on the property concerned. These same sort of conventions have been used with success by Andow(A2) in his method for process computer alarms analysis.

As an example consider the system shown in Fig. 4.1, it is a simple system but adequate to illustrate the point. It consists of a valve and two pieces of pipe, one at each side of the valve. For simplicity only two properties Pressure (P) and Flow (Q) are considered in this example, but that does not mean that other properties cannot be included in the system. Once the units and streams have been identified the assembly of the complete system topology can be carried out, so that it can be used in the generation of fault trees. Since the variables in process streams have one name in each unit and these names do not always have the same letters for the same properties in both units, it was decided for the purpose of this work to use a complete variable described by two letters and a number. The first letter will always be the name given to the variable in the output stream of the other unit, the second letter will be the name given to the same variable in the input stream of the unit concerned. The number will be a unique number given to the stream linking both units. Fig. 4.2 shows the two pipes and valve system but now with the complete variables and the description of the units. Note that in this example the names given to the properties considered in the input and output streams of the units are the same, e.g. PP2, QQ2. Also note that the complete variables in streams 1 and 4 have a blank space to show that no name has been allocated to them due to the boundaries of the system. To solve this problem dummy units are placed before or after any unit whose stream comes from or goes to the "environment" of the system under study. Fig. 4.3 shows the complete system.

The only case in which a blank space may appear in a complete variable is when it represents an internal variable of a unit. As an example consider the system shown in Fig. 4.4 In this example the variable representing the level of the tank is not an input nor an output variable. It is represented only with the name given to it in the unit model of the tank (see Appendix I) and a number. To be consistent with the nomenclature used to describe the topology of a system, it was decided that the number allocated to the first input stream of the unit, where the internal variable is found, would be the one used to identify the internal variable. Therefore the complete variable for the internal variable of the tank in this example is L_2.

4.2 Construction of a Fault Tree

Once the system is defined in the way described, the next step taken to construct the fault tree is to define the undesired event or top event. As a first case assume that the top event for the two pipe and valve system shown in Fig. 4.3 is QQ4.LO.

After the definition of the top event the next step to develop it is to answer the question: In which unit does the undesired event occur? To answer this question reference is made to Fig. 4.3. It can be seen that QQ4 is at the output stream of pipe 4 and is at the same time an input to the dummy unit placed at the end of the system. According to the conventions stated in Chapter 3, QQ4 is the equivalent to Q_{OUT} in the pipe model; therefore, the unit in which the top event occurs is pipe 4. Once the unit has been located, the point at which the minitrees are needed has been reached. The next step is then, to look at the set of minitrees for the pipe unit model obtained from the failure analysis, and find which of the Main events in the set of minitrees corresponds to the top event of the tree that is being developed.

To find the correct minitree, reference is made to Appendix I where all the models and minitrees used for this work are shown. The name of the variable used to look for the minitree is the first letter of the complete variable $\underline{Q}Q4$. The one needed in this case is the pipe's minitree which has as Main event Q_{OUT} LO.

Once the correct minitree is found the next step is to write the appropriate variables in that minitree. Fig. 4.5 shows the minitree in the form as it was obtained by means of the failure analysis. Fig. 4.6 shows the same minitree but now with the appropriate variables according to the topology of the system under study. Note that the minitree has now become the tree for the top event and the construction will be completed only when the tree has been developed up to the point of basic fault events. In this case it has not yet been completed. There are two events in the tree that are Transmissive events PP3 LO and PP4 HI. According to the definitions given in Chapter 3, these events require further development.

To develop these events use is made again of the minitrees shown in Appendix I. Consider first the event PP3 LO, according to the conventions adopted for stream properties, pressure is set at the unit input stream. In this case PP3 is at the input stream of Pipe 4 and the correct minitree should be found among the pipe's minitrees. The minitree which has P_{IN} LO as Main event, is the one needed in this case and is shown in Fig. 4.7.

By writing the proper variables in the minitree shown in Fig. 4.7 and by adding it to the appropriate branch in Fig. 4.6 a new stage in the construction of the tree is reached and is shown in Fig. 4.8. Note that the Boundary Conditions and Not-allowed faults of the first gate are added to those of the new gate. This is done to ease the consistency checks when developing the tree further on.

At this stage there are three events in the tree that need to be developed. Consider now PP4 HI; according to the conventions pressure is set at the input stream and in this case PP4 is being considered at the output of pipe 4. To develop this event use has to be made of the next unit, for which PP4 is considered an input. In this case the next unit is a "Dummy Fail" unit. Its minitrees are shown in Appendix I. The name of the variable used to look for the correct minitree is the second letter of the complete variable. The minitree with P_{IN} HI as Main event is the one needed in this case. By following the same procedure of writing the proper variables and by adding it to the branch that is being developed, a new stage in the construction is reached, it is shown in Fig. 4.9.

By considering the top event of the tree as level zero and by looking at Fig. 4.9 it can be seen that all the events at level one have been developed. Before developing the events at any further level, it is necessary to have a look at the new events in order to check that they may coexist in the fault tree. In this case it can be seen that the event QQ4 HI cannot coexist in the tree because it is under the domain of QQ4 LO and therefore contradicts the top event. This is stated by the Boundary Condition QQ4 HI that affects all the new gates. Therefore QQ4 HI has to be deleted from the tree. With the deletion of QQ4 HI some changes are introduced to the tree. Fig. 4.10 shows the new tree after the modifications. Note that PP4 HI is now represented as a diamond event. Due to the boundaries of the system it is not possible to develop the event any further because there is not enough information to do so.

QQ3 LO is now the only event that has to be developed further. Once again use is made of the conventions adopted. QQ3 is being considered in the tree as an input to pipe 4. To develop the event use has to be made of valve 3 in which QQ3 has been set as an output. The name of the variable used to look for the correct minitree is the first letter of the complete variable. The minitree is found in the set of minitrees for the valve unit, the Main event is Q_{OUT} LOW. By adding this new branch with the appropriate variables to the tree in Fig. 4.10 a new stage in the development of the tree is obtained, the new stage is shown in Fig. 4.11.

It can be seen from Fig. 4.11 that all the events at levels 1 and 2 are now developed, before developing the events at the next level a check for consistency has to be carried out. There is only one contradiction at this level, PP3 HI cannot coexist in the tree according to the Boundary Conditions of the gate for which this event is an input. PP3 HI must be deleted. Any time that a contradiction could exist in the tree, the event that according to the Boundary Conditions and Not-allowed faults is the cause of contradiction and should be deleted from the tree.

Once the Boundary Conditions and Not-allowed conditions are checked, the development of the tree can be continued. There are now two events that have to be developed further PP2 LO and "Valve Closed". Consider first PP2 LO, in order to develop this event the same procedure as before is followed. The minitree needed is found in the set of minitrees for the valve unit, its Main event is P_{IN} LO. A new stage of the development of the tree is shown in Fig. 4.12. Note that the event PP3 HI has been deleted and that the event which has not been developed remains as it was before with the same symbol. Only when the event is developed, is its representation changed to the common one used for fault trees.

The next event to be developed is "Valve Closed". It is at the same level as PP2 LO. In this case the event is a Replaced one. According to the definitions given in Chapter 3, the minitree needed to develop this event can be found in the valve unit. Therefore the minitree with "Valve Closed" as Main event is the one needed. By replacing it in the appropriate branch the tree shown in Fig. 4.13 is obtained.

Before developing the events at the next level the Boundary Conditions and Not-allowed faults have to be checked for the new events added to the tree. From Fig. 4.13 it can be seen that the events QQ3 HI and the fault "Valve Wide Open" are against the Boundary Conditions and the Not-allowed faults therefore have to be deleted from the tree. Once again the development of the tree has to be continued, with only one event needing further development; this event is QQ2 LO. At this stage a similar problem to the one presented with QQ3 in the early stages is faced here for QQ2. To develop the event use has to be made of the unit to the left of the valve; in this case pipe 2 is the unit that according to the conventions has QQ2 set as an output. The set of minitrees for the pipe has already been used in the other pipe of the system. By means of the minitree with the Main event Q_{OUT} LO, the event QQ2 LO can be developed as is shown in Fig. 4.14. Note that the event QQ3 HI and the fault "Wide Open" do not appear in the new tree because they were a contradiction in the tree. Note also how the number of Boundary Conditions and Not-allowed faults has been incremented as the tree has been developed.

At this stage the checking of the Boundary Conditions and Not-allowed faults is carried out again before developing the new added events. The event PP2 HI has to be deleted on this occasion. The only event that needs development now is PP1 LO. With the help of the pipe's minitrees this is done and the minitree with Main event P_{IN} LO is used again but the variables are now related to pipe 2. The resultant tree is shown in Fig. 4.15. Note that one of the Boundary Conditions is QQ2 HI, therefore, the event QQ2 HI has to be deleted from the tree. The new event QQ1 LO is the only one left to be developed. In this case as at the beginning of the development of the tree use has to be made of another "dummy" unit. This is a "Dummy Head" unit and the minitree needed to develop QQ1 LO is the one with Q_{OUT} LO as Main event. The new stage of the tree is shown in Fig. 4.16. Note that the event PP1 HI cannot coexist in the tree with PP1 LO, the Boundary Condition PP1 HI is present and PP1 HI has to be deleted from the tree. The event QQ1 LO cannot be developed any further because of the lack of information due to the boundaries of the system. It will then be represented as a diamond event. The final tree is shown in Fig. 4.17. Note that the final tree does not present any trace of the minitrees used, all the transmissive and Replaced events have been developed and the tree is developed up to the point of basic fault events and diamond events.

The example described illustrates the systematic linking procedure which is the basis of the fault tree generation algorithm and also shows some of the problems that can be faced during this procedure. Fig. 4.18 shows another tree which was developed for a top event in the two pipe and valve system. Note that the top event in this case PP2 HI is located in the middle of the system and in spite of this the flow of information travels in both directions and not only downstream. This feature is very important for the purpose of this work, because it allows the engineer to select any variable of the system and obtain the fault tree for the variable, with the confidence that the tree will show how the fault propagates throughout the system.

4.3 Units with Two or More Input/Output Streams

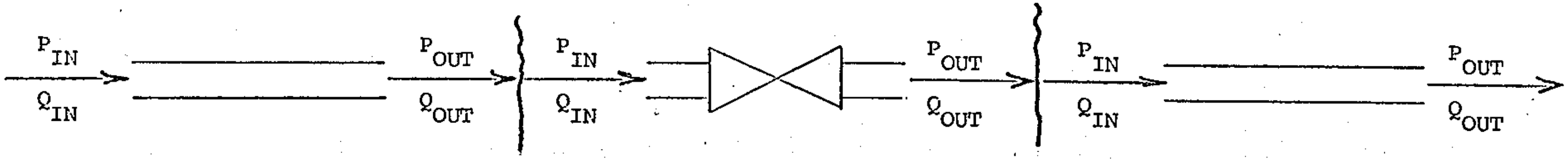
When units with more than two streams are used some problems may arise, but the use of the complete variables and of the conventions stated for the naming of these variables helps to avoid problems. It also makes sure that the flow of information is correct. To illustrate this point consider the example shown in Fig. 4.19. It shows a system formed by a heat exchanger, a pipe and a valve. Note that in this case the heat exchanger model has two input streams and two output streams. The properties for the cold streams considered for this example are Pressure (A) and Flow (B) and the properties for the hot stream are the same, but their names are the common P and Q used in the models with only one input and output stream considered before.

Consider the top event to be PA5 HI. According to the convention stated in Chapter 3 pressure is defined at the input stream of the units, therefore the minitree needed to develop the top event, should be found among the set of minitrees for the heat exchanger. From the complete variable it can be seen that A is the letter used to represent pressure at the input of the cold stream. Therefore the minitree with the Main event A_{IN} HI is the one needed in this case. Note the importance of defining the topology of the system according to the names used for the different properties in the unit models. Although the property is pressure, the letters used to represent it at the hot and cold streams are different in order to avoid any possible mistake when the search for the correct minitree is made. Had the complete variable been wrong, say AP5 instead of PA5, then the minitree chosen would have been P_{IN} HI. This is the wrong one because this minitree refers to the pressure at the input of the hot stream as it can be seen in the unit model for the heat exchanger described in Appendix I.

4.4 Deletion of Events Under the Domain of AND Gates

When an event that is under the domain of an AND gate has to be deleted from the fault tree being developed, all the events that are under the domain of the AND gate will also have to be deleted. This is because when the minitree used to develop the output event of the

AND gate was obtained, all the input events to the gate were required to occur, to cause the output event. If a minitree with an AND gate is used to construct a fault tree and one of the input events has to be deleted, then the output event cannot occur any more. Therefore, there is no need to keep it in the fault tree, nor any of the input events to that particular AND gate. Fig. 4.20 illustrates this case. Event B has to be deleted because of event G, which was an input event to the AND gate, and cannot exist in the tree due to the Boundary Condition NO G. Further checks should be made when an event such as B is deleted from a tree. It may well be that B was the input event of another AND gate and the same procedure of deletion would need to be repeated until no AND gates affected by those deletions were found. Fig. 4.21 shows the new tree for the top event once the necessary deletions were carried out.



Unit:	Pipe 1	Valve 2	Pipe 3
Model:	$Q_{OUT} = k_1(P_{IN}-P_{OUT})^n$	$Q_{OUT} = k_2(P_{IN}-P_{OUT})^{\frac{1}{2}}$	$Q_{OUT} = k_3(P_{IN}-P_{OUT})^n$
	$\frac{dP_{IN}}{dt} = G_1(Q_{IN}-Q_{OUT})$	$\frac{dP_{IN}}{dt} = G_2(Q_{IN}-Q_{OUT})$	$\frac{dP_{IN}}{dt} = G_3(Q_{IN}-Q_{OUT})$

Fig. 4.1 Identifying units and process streams in a two pipe and valve system

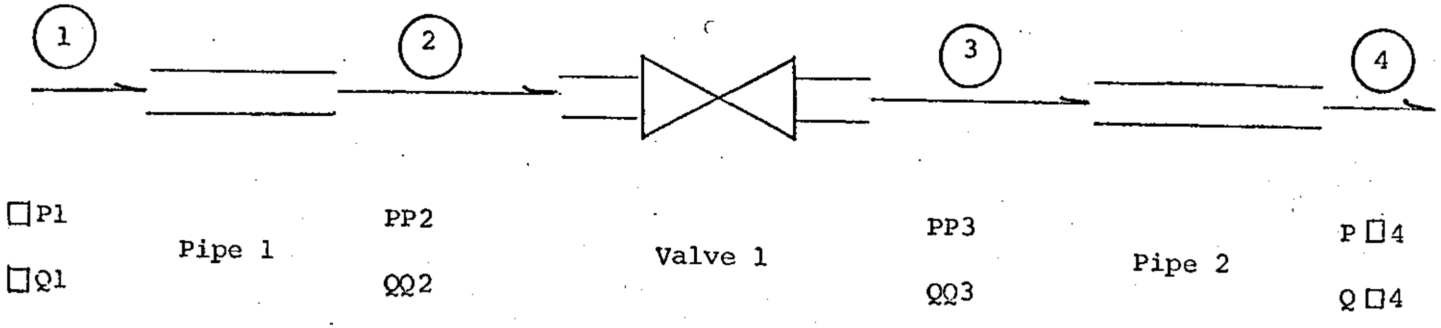


Fig. 4.2 Two pipes and valve system with complete variables and the units description

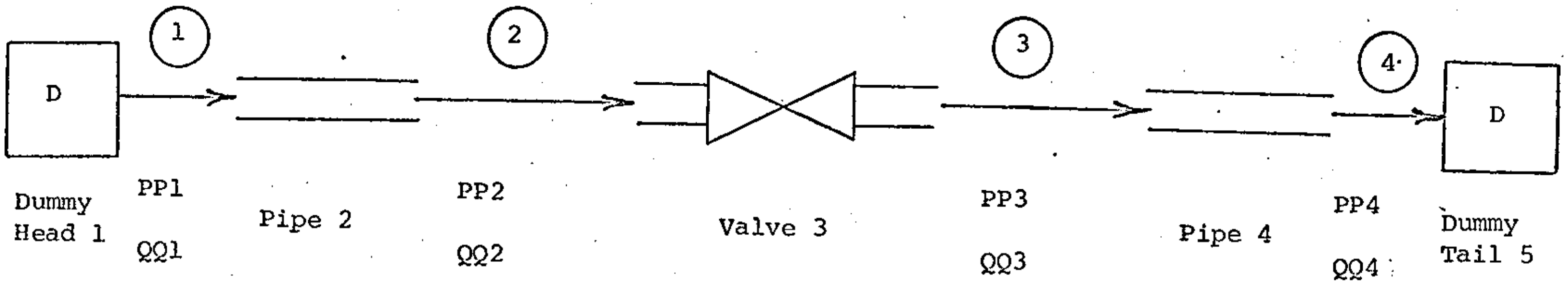


Fig. 4.3 Two pipe and valve system with "dummy units"

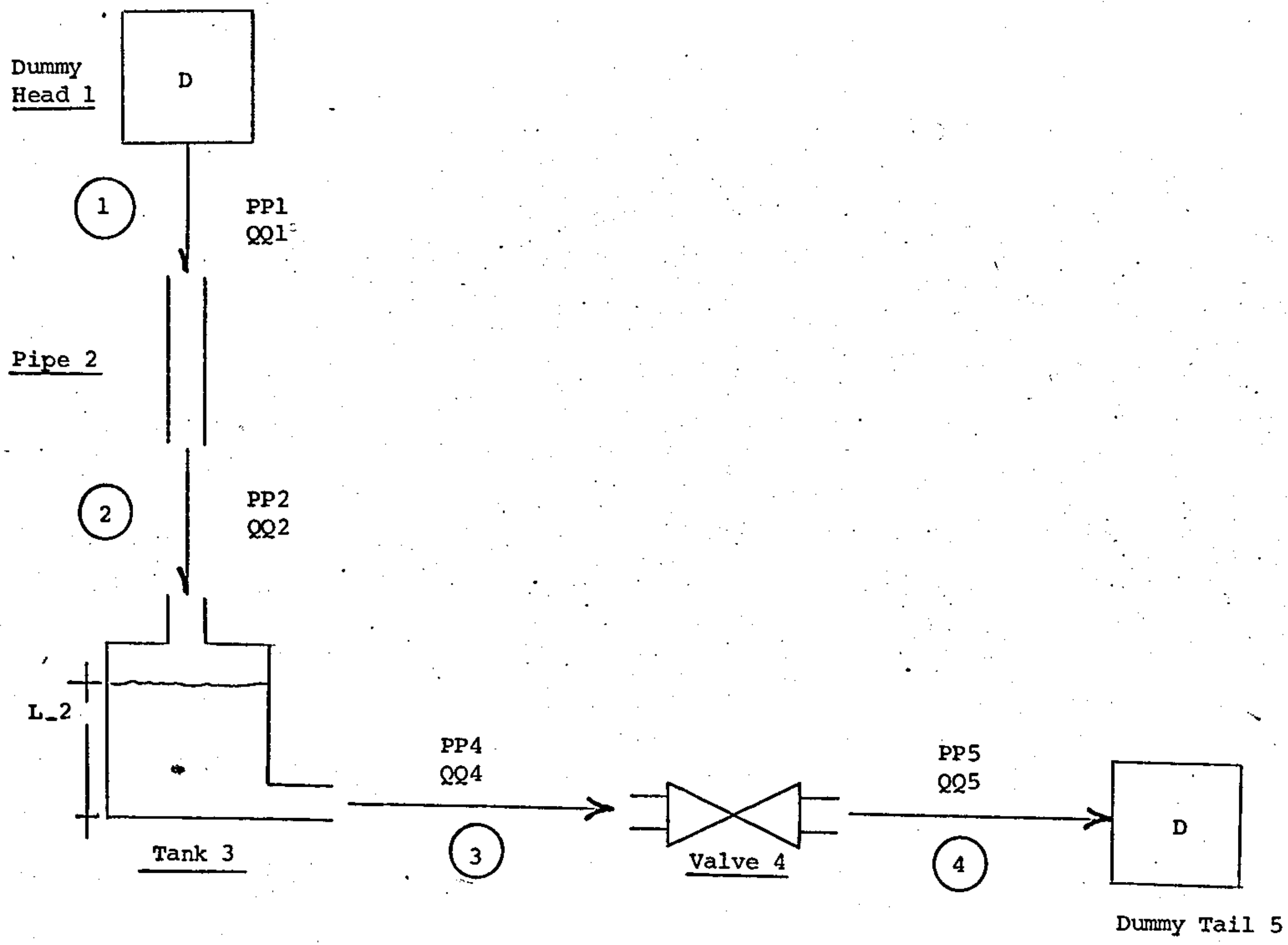


Fig. 4.4 Pipe, tank and valve system

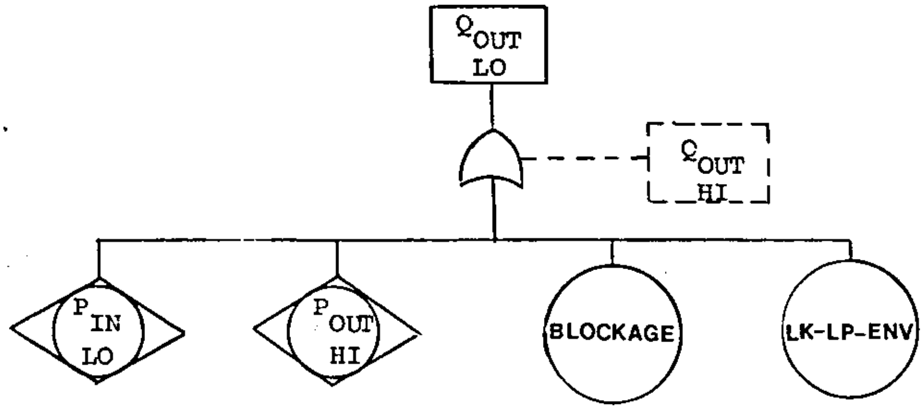


Fig. 4.5 Minitree used to develop the top event QQ4 LO.

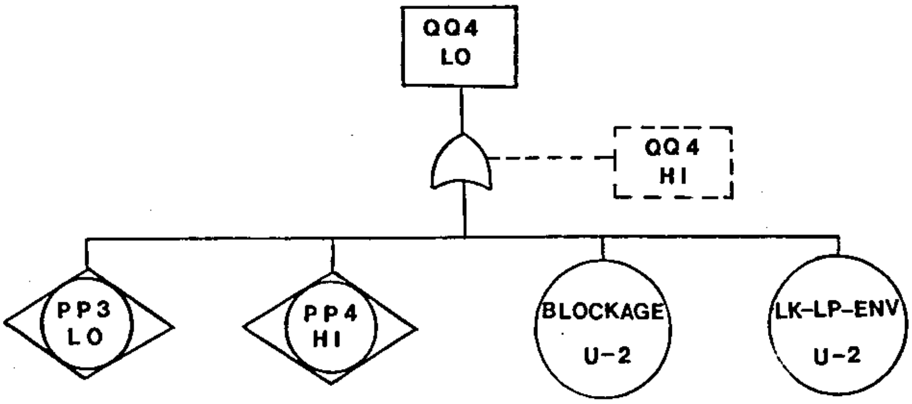


Fig. 4.6 First stage in the development of the top event QQ4 LO

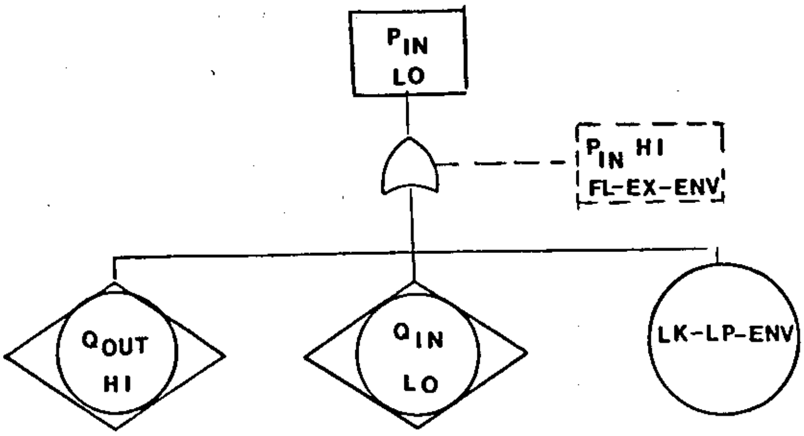


Fig. 4.7 Minitree used to develop the event PP4 LO

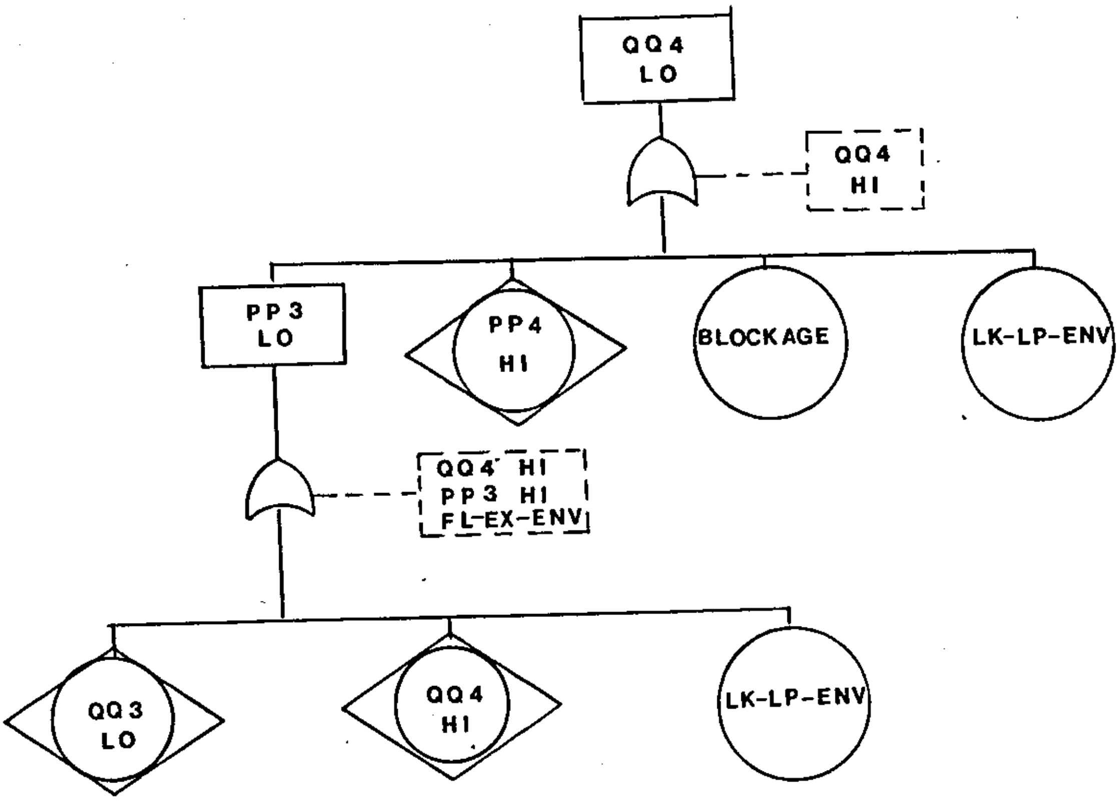


Fig. 4.8 Another stage in the development of the top event QQ4 LO

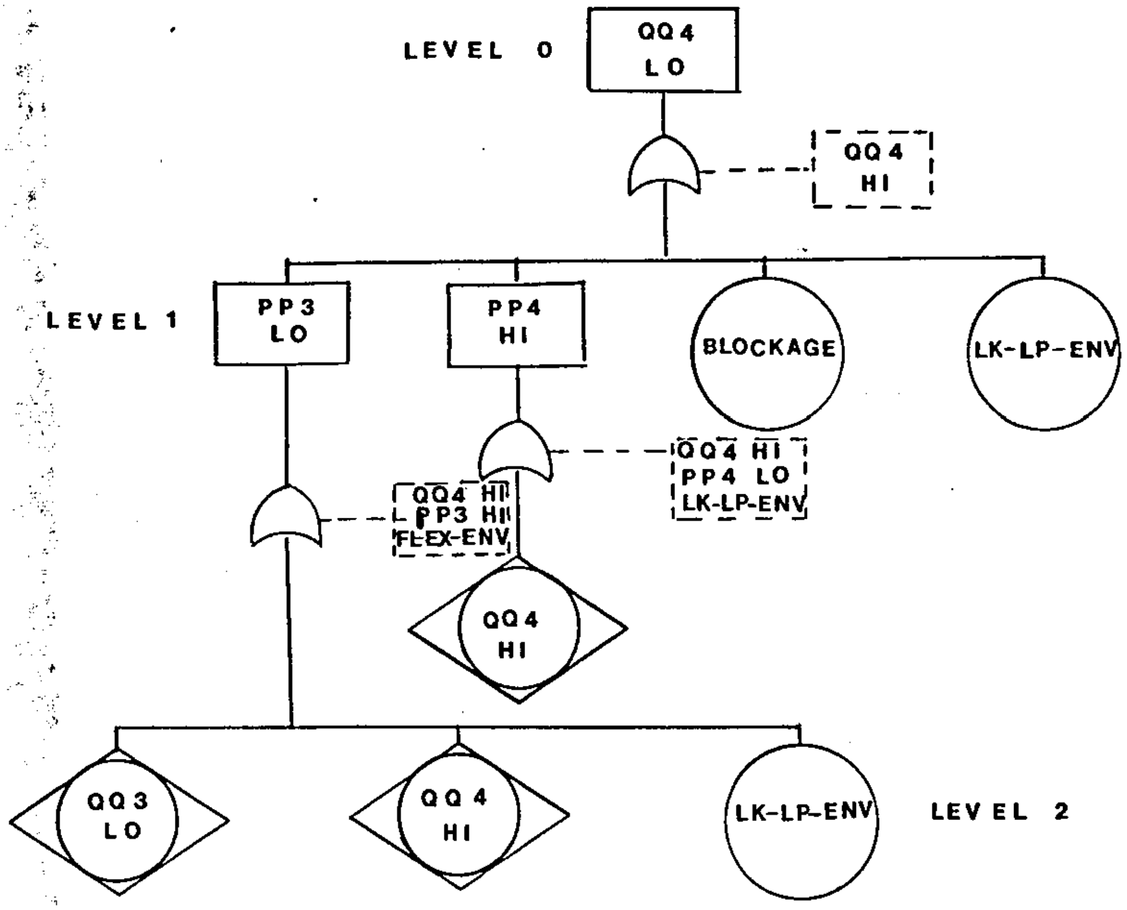


Fig. 4.9 Another stage in the development of the top event QQ4 LO

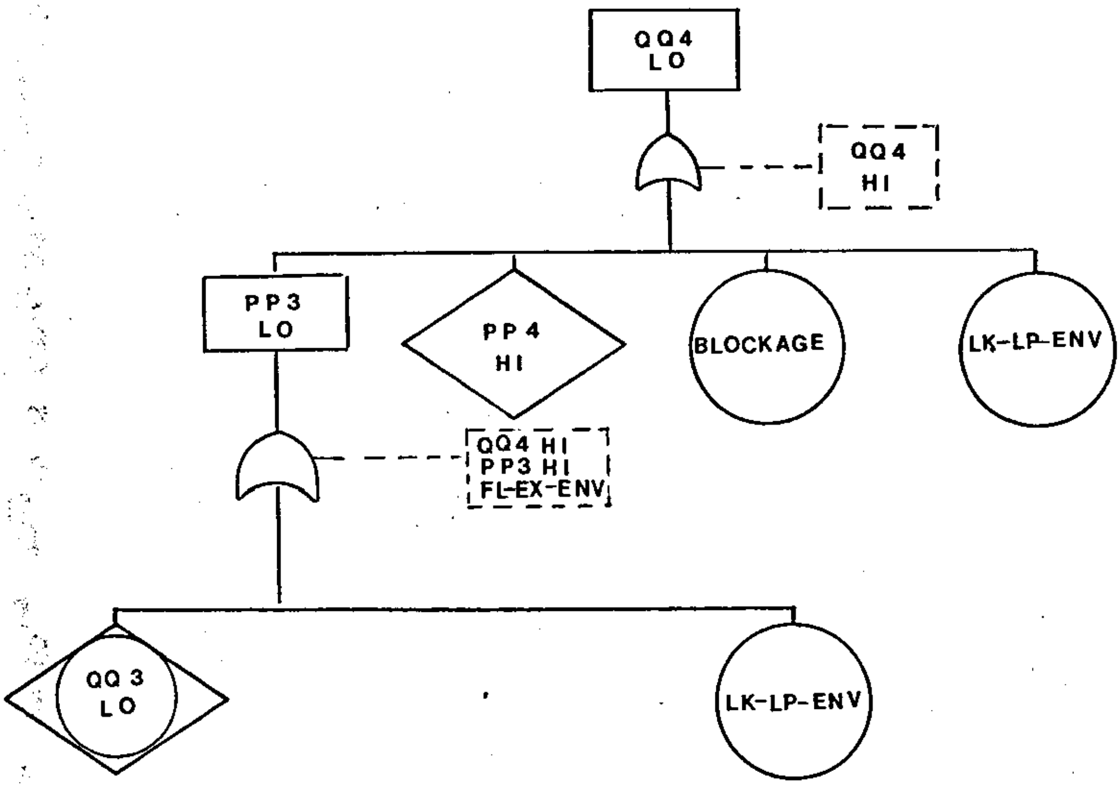


Fig. 4.10 Another state of the Fault Tree for QQ4 LO after Boundary Conditions and Not-allowed faults have been checked

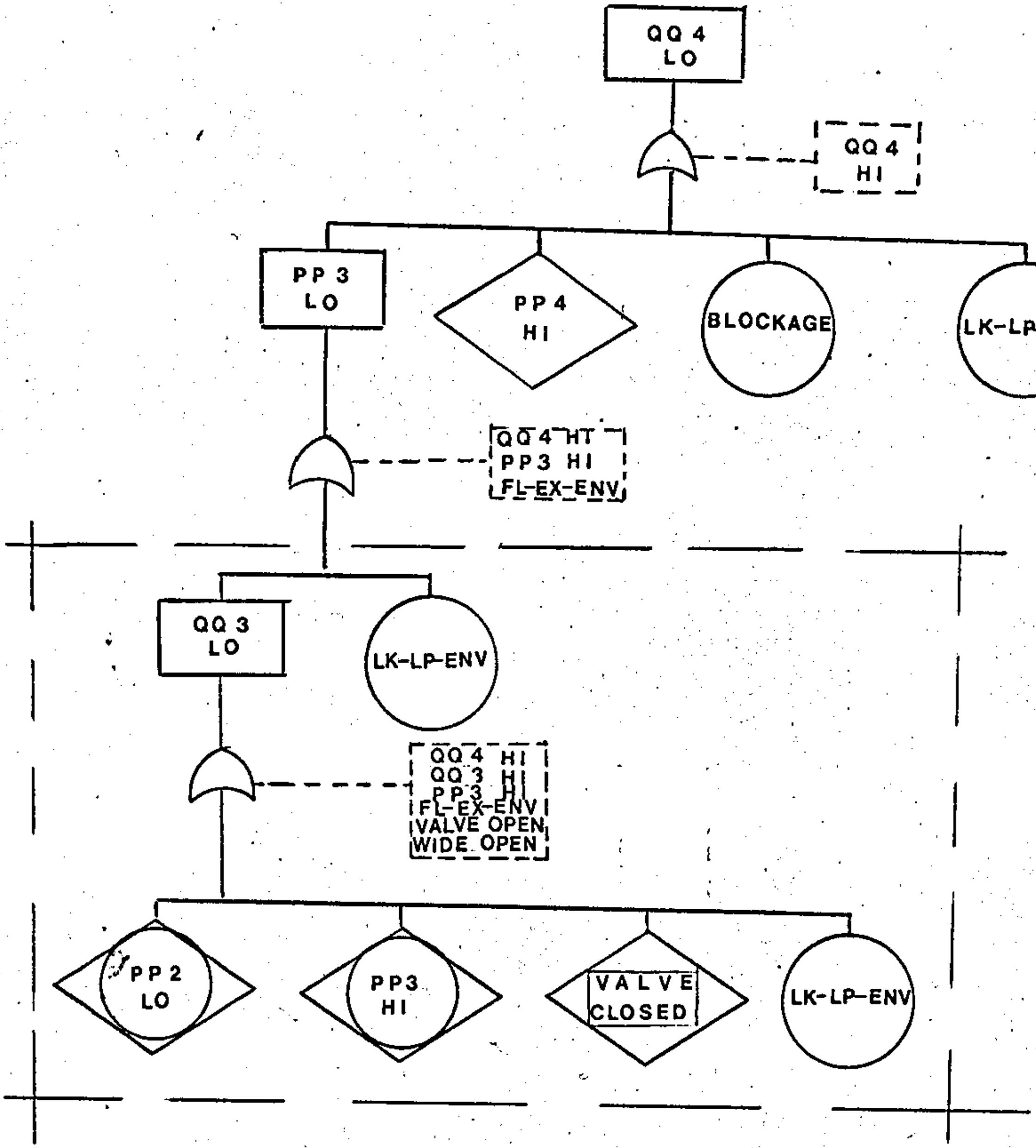


Fig. 4.11 Development of QQ3 LO in the tree

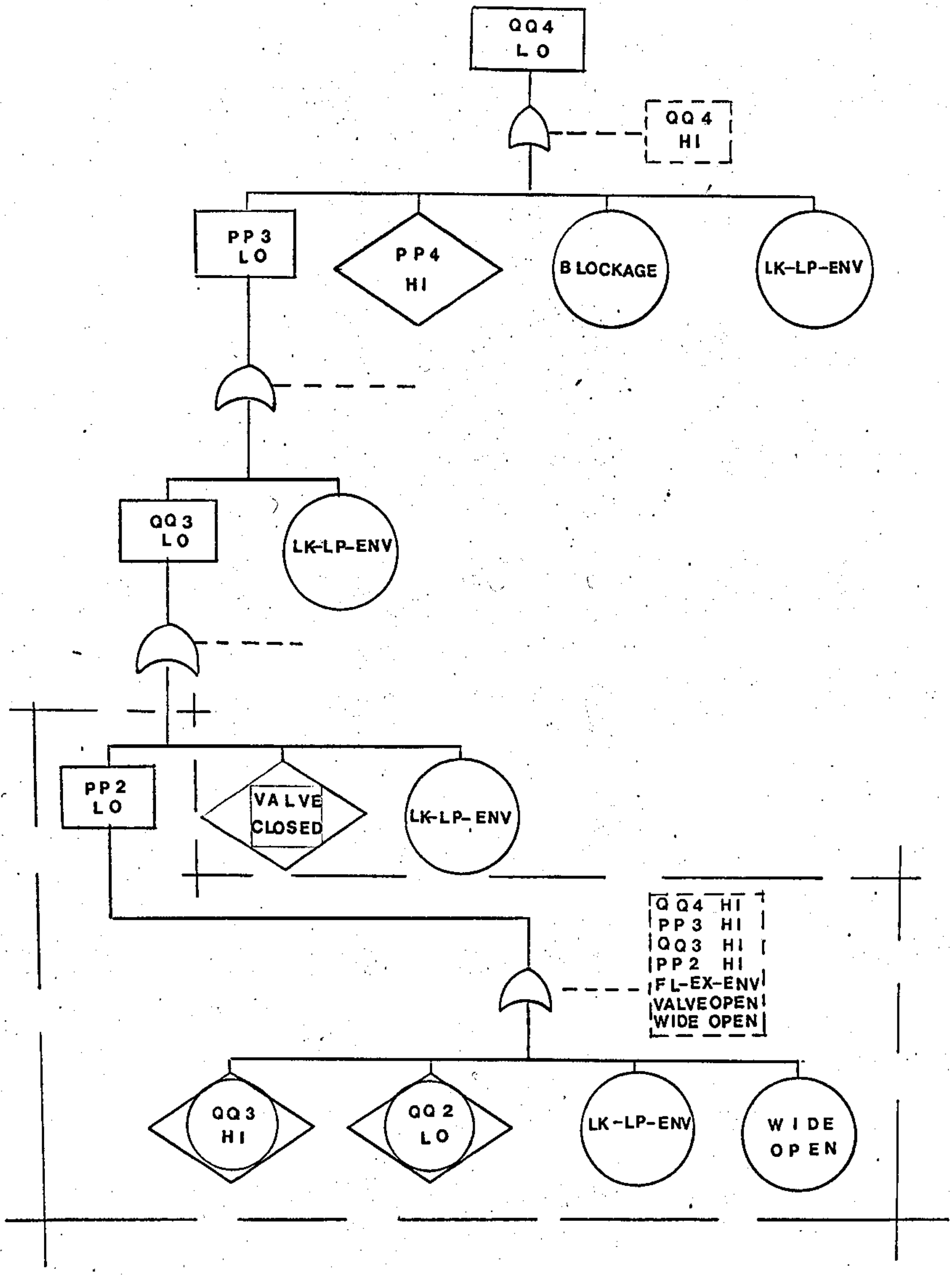


Fig. 4.12 Developing PP2 LO in the tree

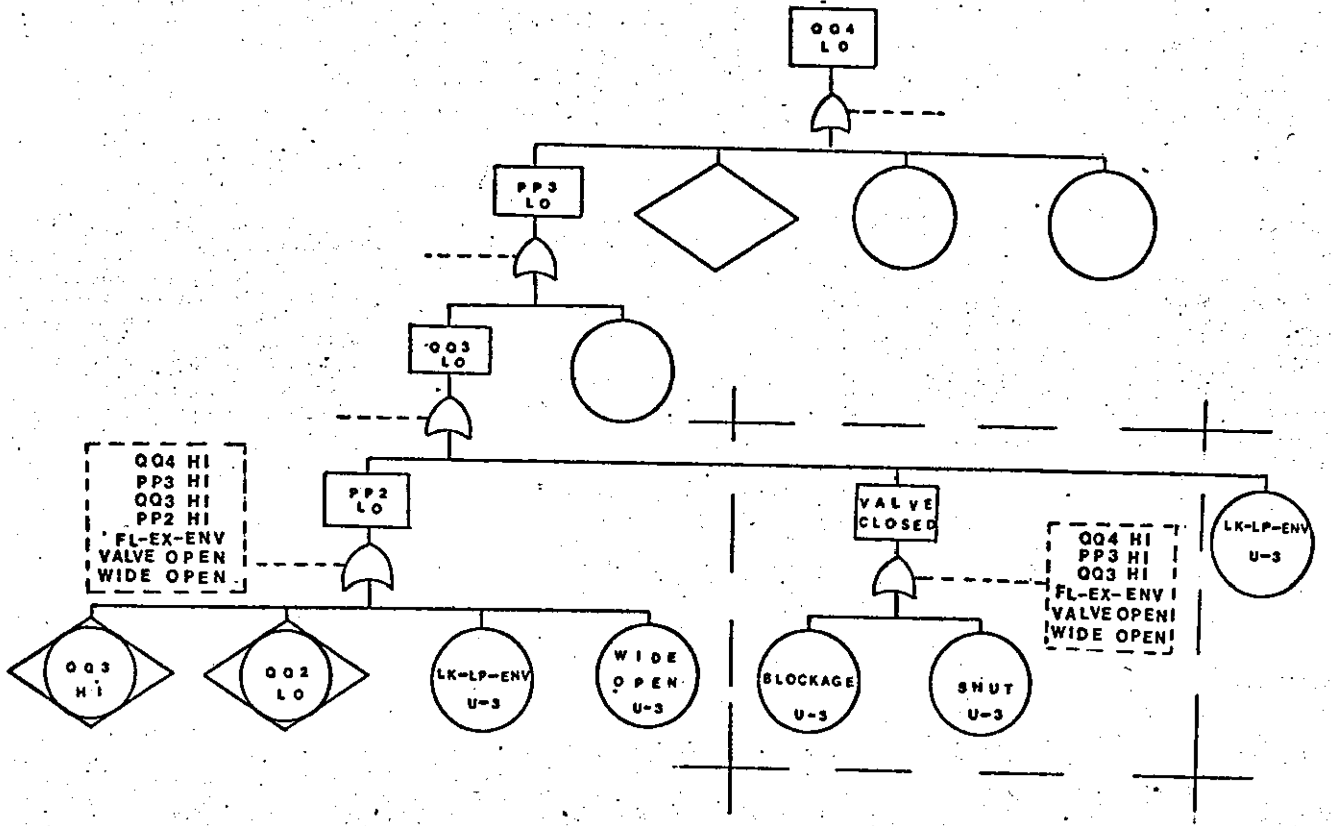


Fig. 4.13 Development of the Replaced event "valve closed"

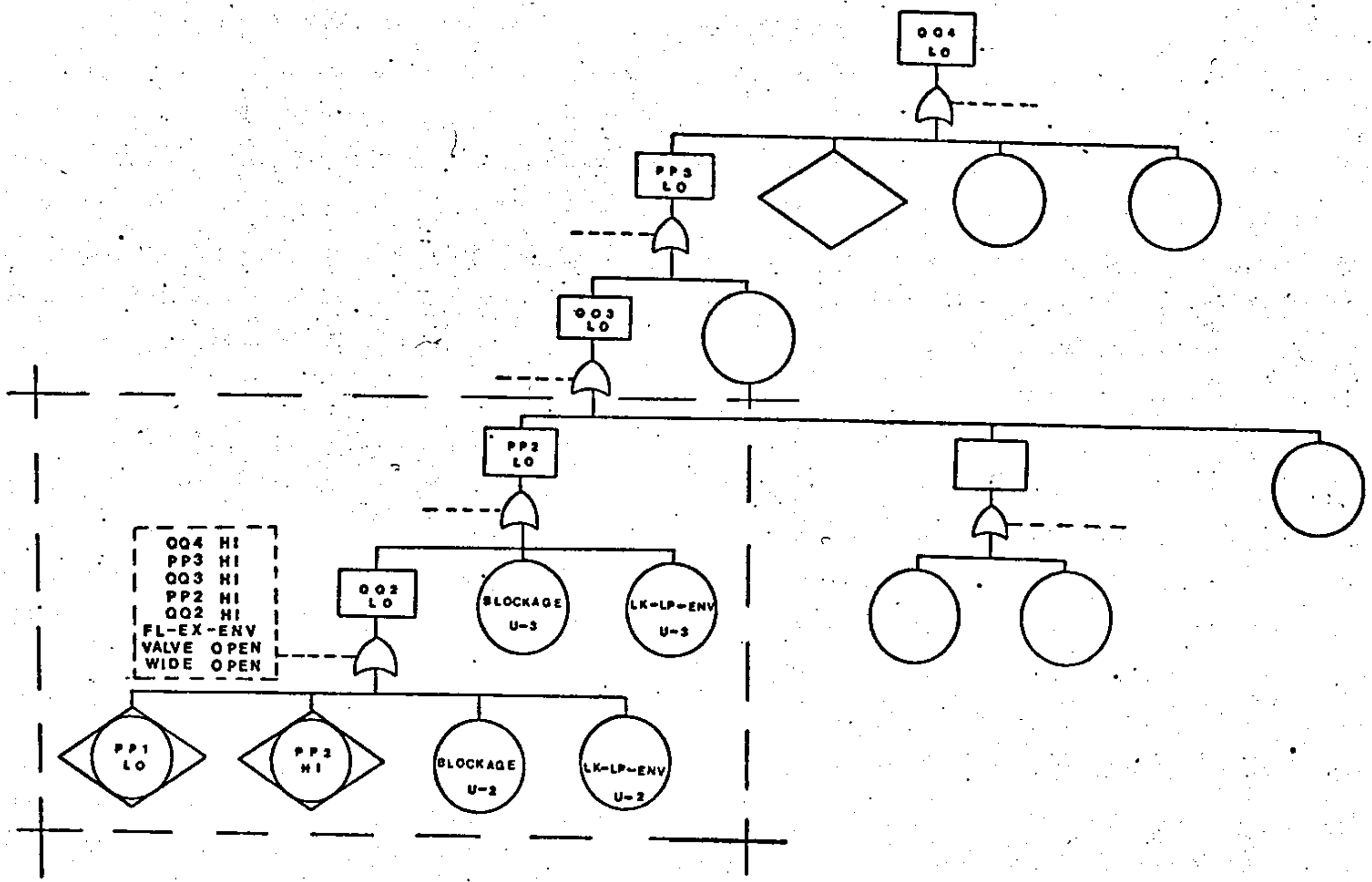


Fig. 4.14 Development of QQ2 LO

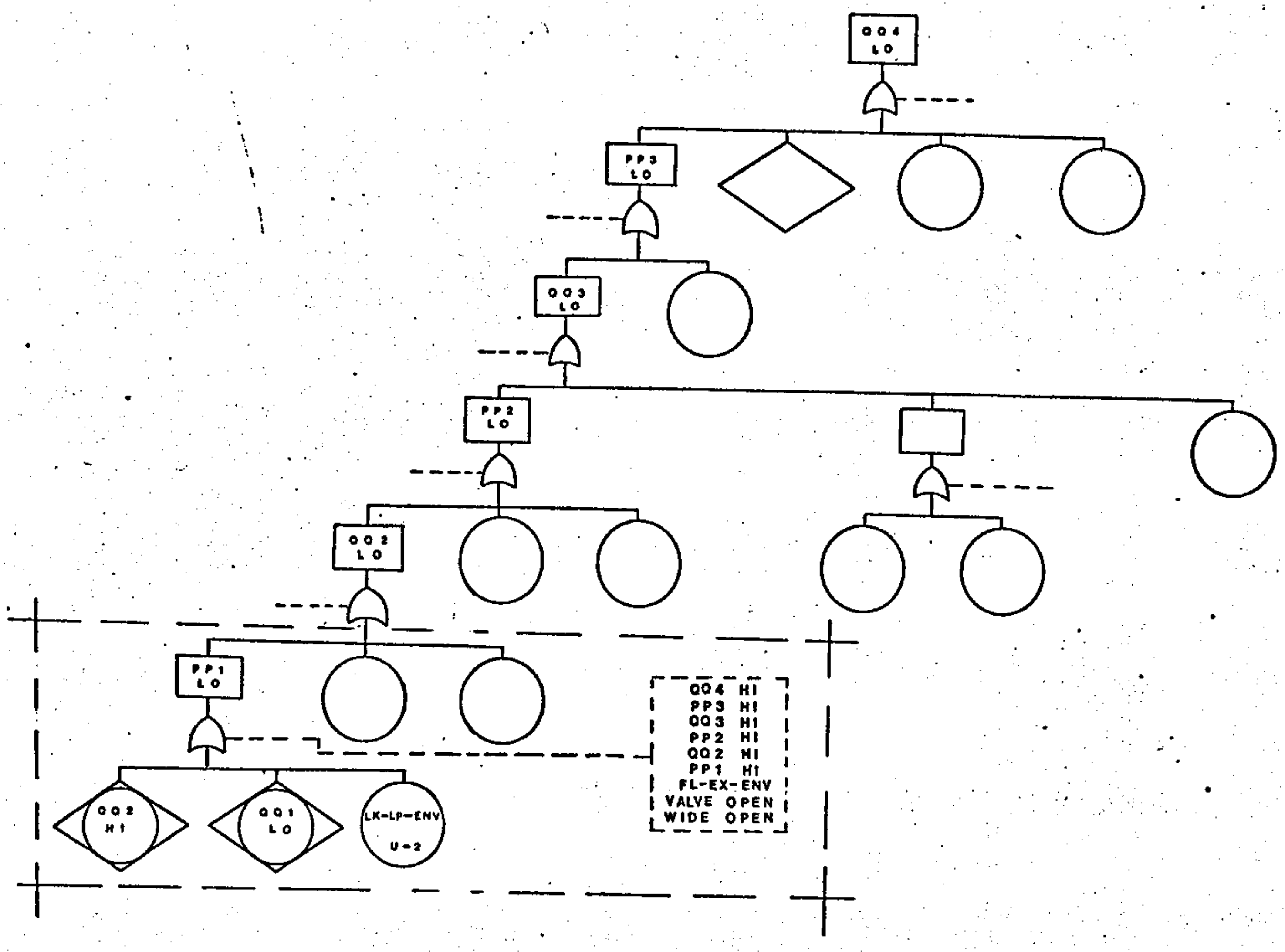


Fig. 4.15 Development of PP1 LO

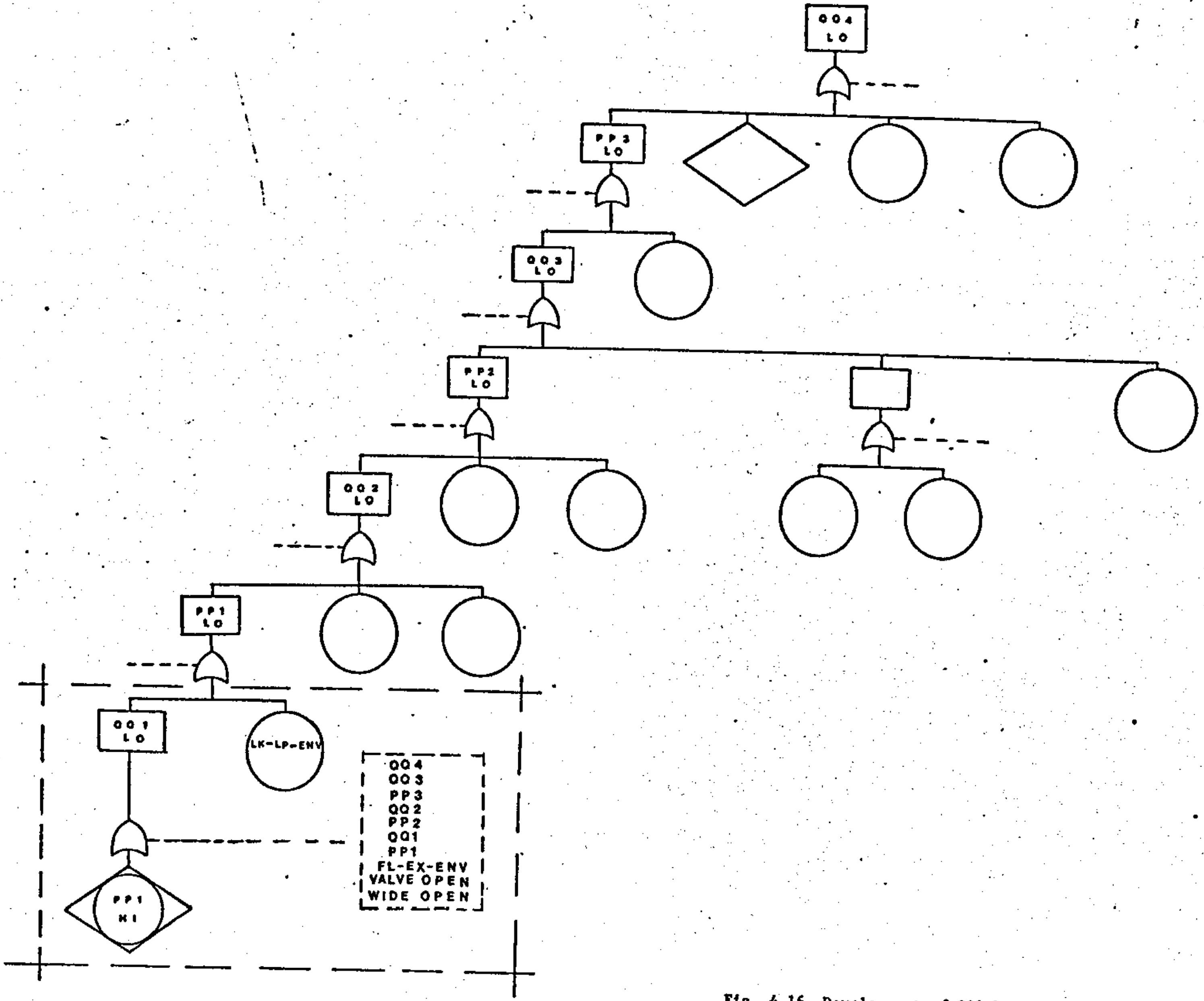


Fig. 4.16 Development of QQ1 LO

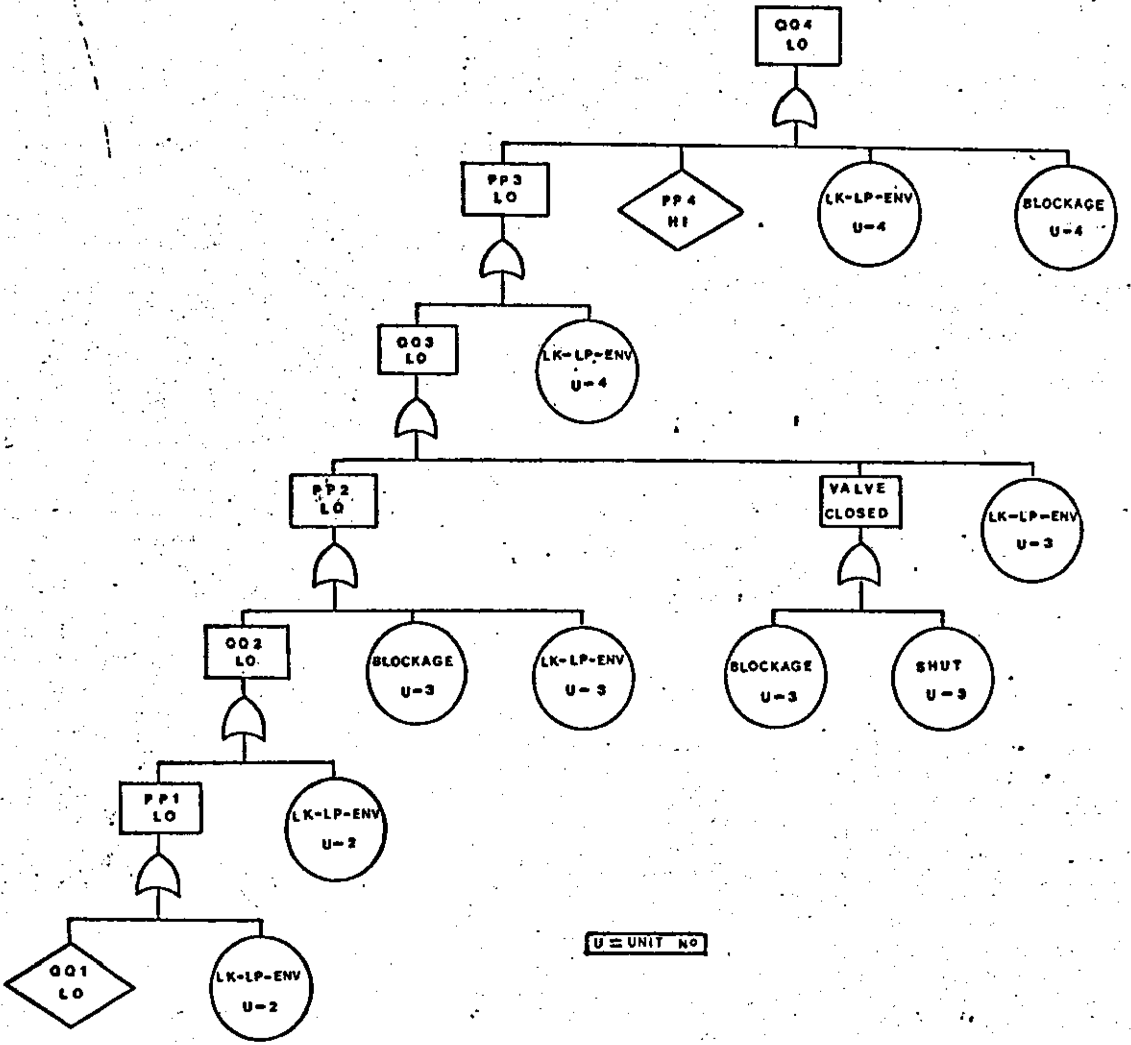


Fig. 4.17 Final fault tree for QQ4 LO in a two pipe and valve system

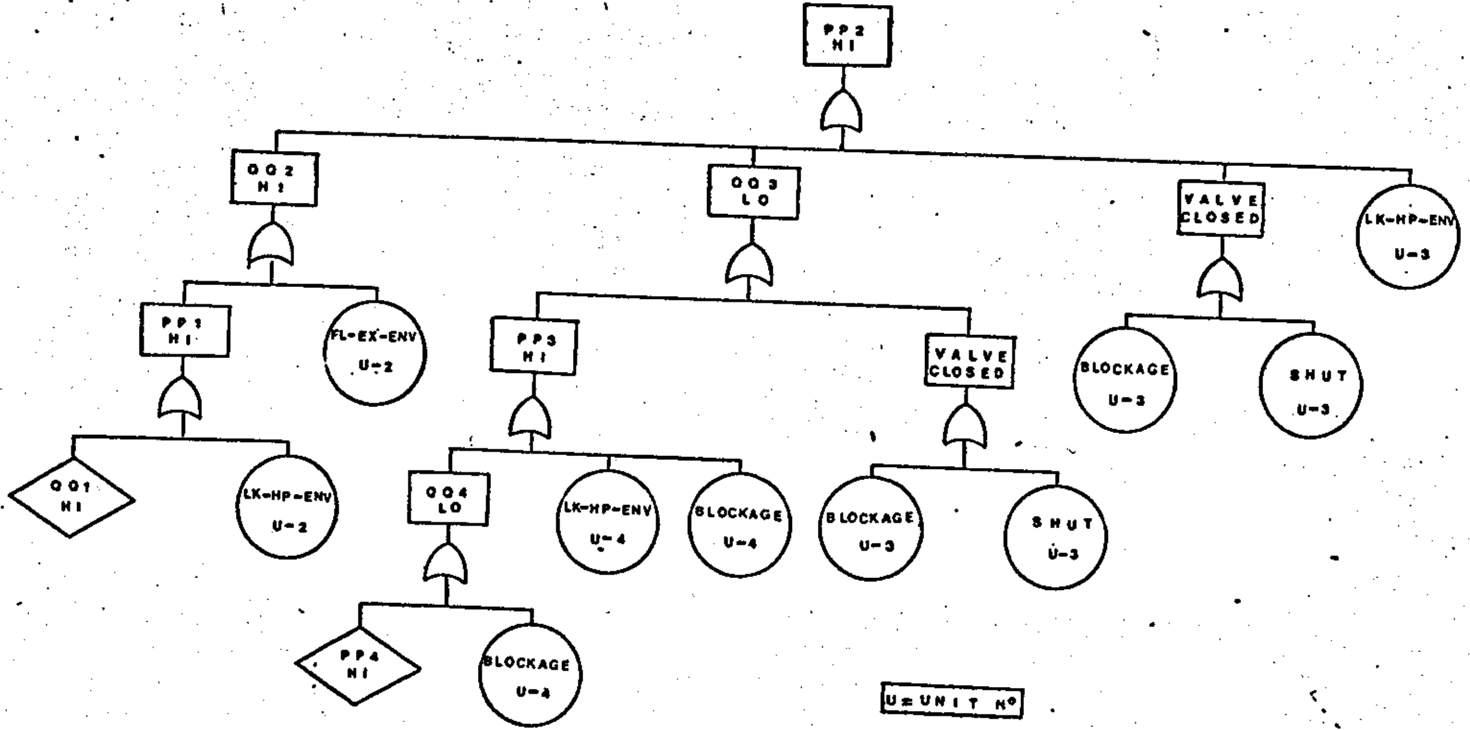


Fig. 4.18 Final fault tree for top event PP2 HI in a two pipe and valve system

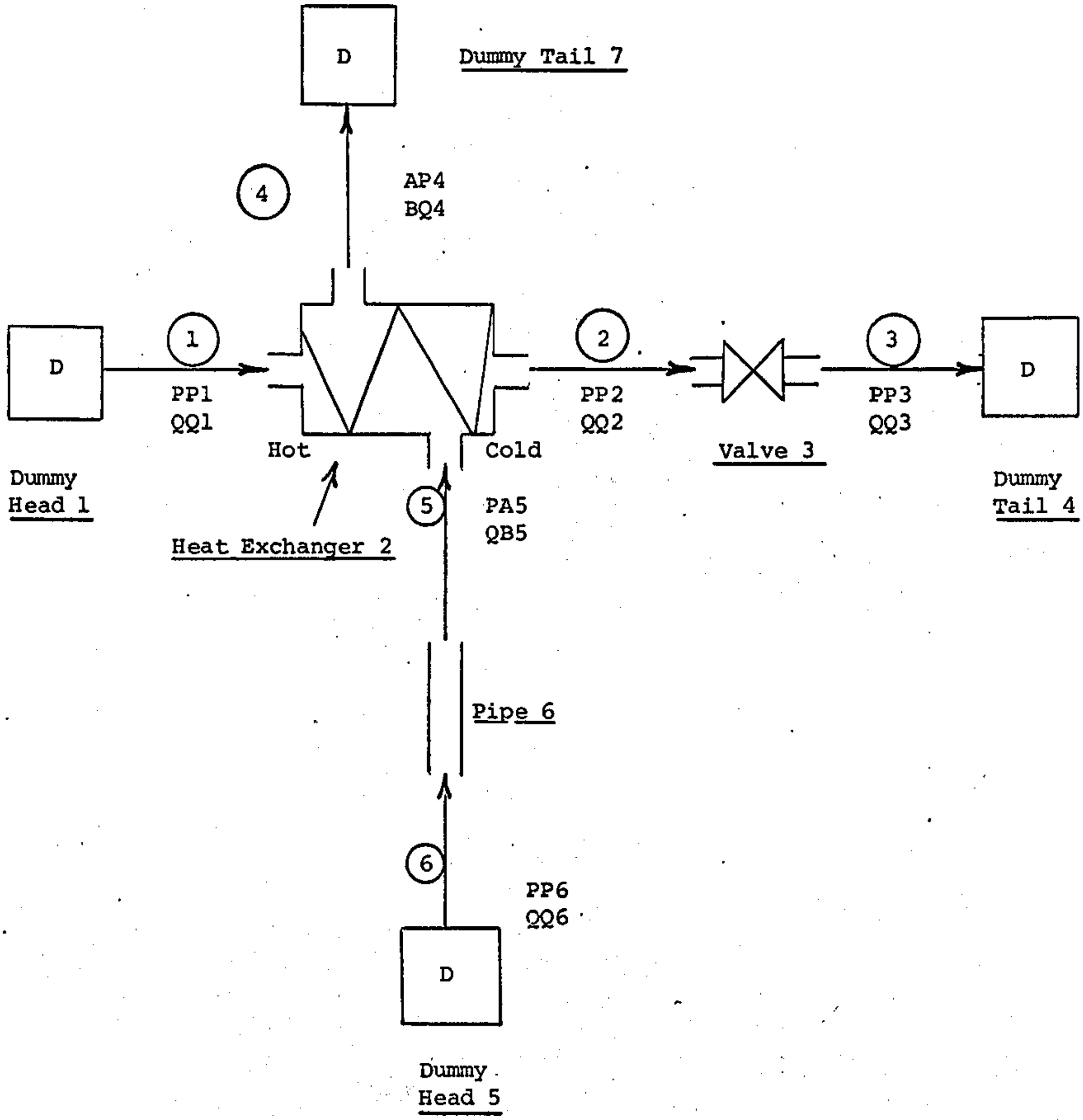


Fig. 4.19 Heat exchanger, pipe and valve system

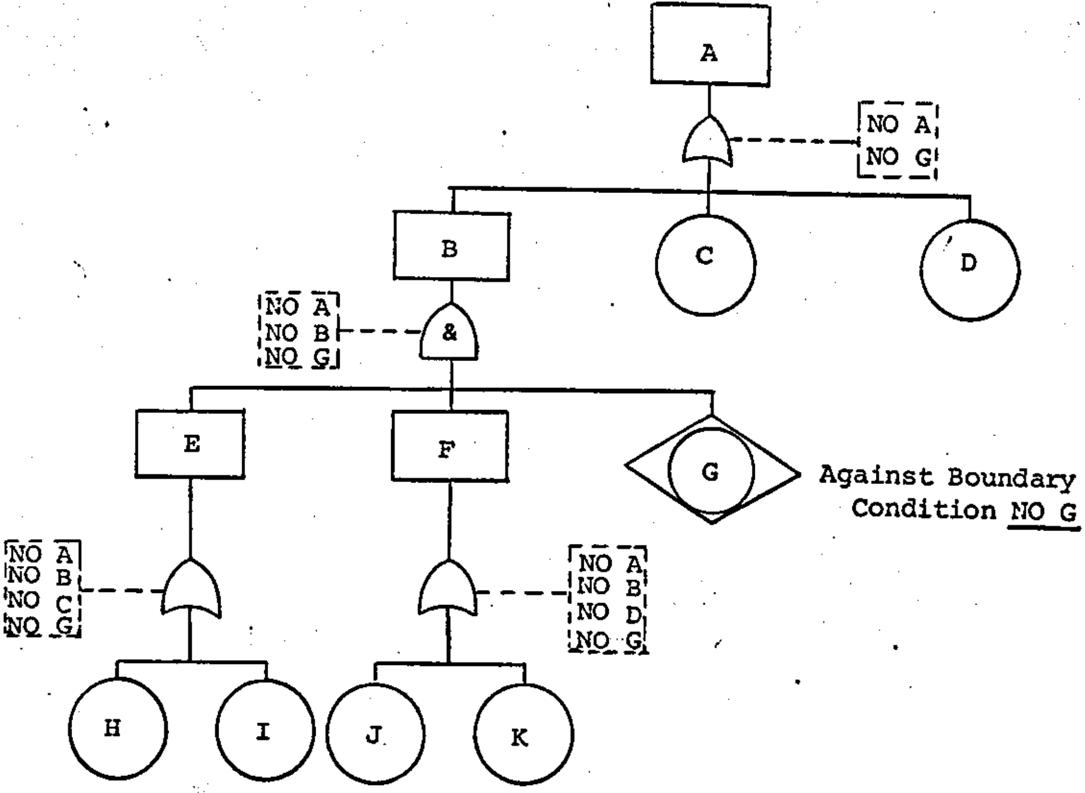


Fig. 4.20 Event G is under the domain of an AND gate

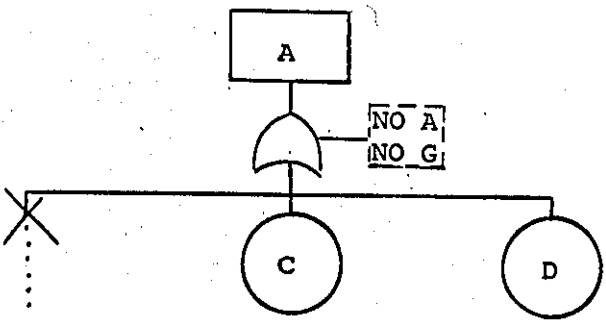


Fig. 4.21 Tree after deletion of event B

Chapter 5

IMPLEMENTATION OF THE FAULT TREE SYNTHESIS METHODOLOGY ON A DIGITAL COMPUTER

The input requirements for the Fault Tree Synthesis methodology were described in the former chapters. This chapter is devoted to describing the implementation of the methodology on a digital computer.

5.1 Alternative Approaches to Implementation of the Methodology

To solve the problem of implementing the methodology on a digital computer two basic approaches were considered:

1) Special Program Approach

   The program is specific to the plant and can only be used for that specific system.

2) Standard Program and Specific Data Approach

   This approach requires one general program for all systems and a specific data base for each plant.

Due to the nature of this work, it was decided that the second approach was the most appropriate if a flexible program was desired. This approach would allow a relatively inexperienced user to set up his plant model and use the methodology to carry out a fault tree synthesis for a specific top event of his plant model.

Fig. 5.1 illustrates the approach used in this work. Note how the methodology can be used either for real time or design purposes.

5.2 Implementation of the Methodology

To achieve the implementation of the methodology in a digital computer several problems had to be solved. One of the main problems was to find a suitable computer language, capable of handling the data required to construct the fault trees. A language with list processing as one of its features was thought to be desirable, because list processing had proved to be very useful when networks and fault trees were handled by computers.(A2,K3)

At the early stages of this work Algol 68(A5) was used but, later on, due to its features and the facilities available in the Department, RTL/2 was the language chosen to develop the computer programs. RTL/2 is a high level programming language developed at the Corporate Laboratory of Imperial Chemical Industries Ltd. It is intended primarily for use in multitask systems on smaller computers and it clearly incorporates many features of other languages such as Algol 60, Algol 68, Algol W, 3CP2, Coral 66 FORTRAN, PL/1 and POP-2.

The computer used for this work was a PDP 11/20 and the operating system was RSX-11M. The operating system is a multiprogramming, real time system and its fundamental function is to provide the control for sharing system resources, among any number of user prepared tasks. The tasks stored on a file-structured volume may be installed into the system and subsequently run by issuing a command to the Monitor Console Routine (MCR). MCR provides the language interface between the operator and the system. MCR has an indirect file processor task (.AT) which is capable of reading a command input file and interpreting each line as either a command to be passed directly to MCR or a request for action by the task itself.

For this work all the interaction with the computer was carried out through a teletype terminal and the indirect command files were widely used. A more detailed description of the system and of the language used is presented in Appendix II.

Once the problem of an adequate language was solved, a data-base restricted to a maximum of 8K (due to the space limitations imposed by the hardware) was created. A detailed description of the data-base used can be found in Appendix III.

Three main algorithms were developed to achieve the purpose of synthesising the fault trees. According to the methodology used they can be named as:

1) Algorithm to set up the minitrees.
2) Algorithm to define the topology.
3) Algorithm to build fault trees.

Each algorithm will be discussed in the following sections of this chapter.

5.2.1 Algorithm to set up the minitrees

The use of the minitrees to construct fault trees was described in Chapter 4. If the minitrees are employed to construct the fault trees by means of a digital computer it is necessary to have this information stored in such a way, that it can be handled by the computer. To implement the minitrees information in a form which can be easily stored and retrieved, an algorithm was developed. Its flowchart is shown in Fig. 5.2. The algorithm uses as input the set of minitrees obtained from each unit model by means of the failure analysis method described in Chapter 3. The algorithm was developed bearing in mind that it should be as simple as possible so that a relatively inexperienced user would be able to set up the minitrees without problem. Resemblance to the events described in Chapter 3 was maintained so the user can feed the data into the computer by interacting with it through a teletype. Once the information is saved, it can easily be retrieved at any time - a specific minitree is needed to help in the construction of a fault tree.

Several programs are used by this algorithm, all of them have been gathered together and built into a task as a module. The name given to each task in this work was restricted by the operating system, to only three letters. Each name was given, bearing in mind the use of each task.* In the case of the algorithm to build the minitrees the task name given was BMT (Build Mini Trees). The same name is used as keyword in the indirect command utilised to set up the minitrees for a specific unit model. The indirect command can be formed in two parts:

1) The name of the task required.
2) The input/output files and the devices required for the specific task. If the task has several options, the option desired is included too.

For the case of the pipe minitrees the command would be as follows:

---

*For a description of each task see page 560.

$$\underbrace{\text{BMT}}_{1}\ \ \underbrace{\text{TI}}_{2}\text{:=}\underbrace{\text{DK0}}_{3}\text{:}\underbrace{\text{PIPE.DAT}}_{4}$$

1—Is the name of the task.

2—Is the output device (in this case the teletype).

3—The source device (in this case the disk).

4—The source file for the unit model pipe.

The use of keywords/data driven programs proved to be very useful during this work. The input files needed for the different tasks can be produced using the editor provided by the operating system. In this way any typing mistake in the input data is easily corrected and there is no need for retyping all the input data. Each data file is given a name according to the unit or system described. An example of these files is shown in Fig. 5.3 It corresponds to the input data for the pipe's minitrees.* All the files of this type for each unit model used in this work can be found in Appendix I together with the unit models and minitrees. Note the order in which the events are described. "M" events first together with Boundary Conditions and Not-allowed faults. "T" and "R" events after the "M" events. "B" events at the end together with its probability. This order of description makes easier any correction or modification that could be needed. The same order should be used if any other minitree is added to these files or new files are created. The use of BMT is shown in Appendix III listings 1 and 2. Listing III.1 is the input data for the pipe's minitree. Listing III.2 is the input data for the heat exchanger's minitrees. Note that in both listings a probability for the basic events is required. For this work they are dummy probabilities, they are included in the data as matter of completeness for future work when the trees may

---

*See also page 583.

need to be evaluated. Also note that both listings were produced by using the data files of each unit model. If the operator wishes to interact with the computer through the teletype, he has the option to do so. There will be tasks in which he would like to do so. In the case of BMT the command would be as follows:

```
BMT TI:=TI:
```

In this case the input will be coming from the teletype terminal and not from an input file. The big disadvantage of doing this, is that any typing mistake will mean retyping all the data with the consequent delays.

The result of the data used to set up the minitrees can be checked by using another task. PMT (Print Mini Trees) is the keyword used in this case. All the minitree printouts shown in Appendix I were obtained with the help of PMT.

Details of all the programs, the tasks and their names can be found in Appendix III.

5.2.2 Algorithm to define the topology

To input the topology information for a specific system into the computer an algorithm was developed. The algorithm assembles the complete plant model from the data provided for each unit model of the system. The data required by the algorithm should be in the form used to describe the topology presented in Chapter 4.

The programs used by the algorithm have been split into two tasks. The first task used is REU (REad Units) and requires as input data the total number of units in the system, their names and the number allocated to them for identification, dummy units should be included too. The second task, DES, (DEScription) sets up the topology of the system using the data already provided through task REU and requires as input data the streams and variables of each unit.

As a convention for the algorithm, to be consistent with the convention presented in Chapter 4, any internal variable of the unit being described, should be set as a variable of the first input stream. Recall that internal variables are the only complete variables that may have a blank space in their name. The type of data supplied to the programs of task REU are shown in Appendix III listng 3. The type of input data required by task DES is shown in Appendix III listing 4. Both listings refer to the topology of the two pipe and valve system described in Fig. 4.3.

The output of the data used to set up the topology can be checked by using the task PRI (PRInt). Fig. 5.4 shows the output provided by the task PRI for the topology of the two pipe and valve system. If a variable is measured in the system it will be marked with an (M) in the output. Fig. 5.5 shows the flow chart of the algorithm.

5.2.3 Algorithm to build fault trees

Before the algorithm was developed two approaches to construct the tree were considered:

1) Vertical development

   Using this approach each branch of the tree is developed completely up to the point of basic events. Fig. 5.6 illustrates this approach. The numbers are used to show the order in which each event is developed.

2) Horizontal development

   Using this approach the tree is developed according to the different levels of the tree. Fig. 5.7 illustrates this approach. Again the numbers are used to show the order in which each event is developed.

The approach used only helps, to decide which event is to be developed next, but it does not affect the linking process of the mini-trees. At the end the fault tree obtained by either approach should be the same. Although the systematic linking process described in Chapter 4 uses the horizontal approach, it was decided that for the purposes of this work the vertical approach should be used in the algorithm to generate the fault trees. The main reason for this was that if the fault tree is generated in real time, the vertical approach, due to its nature, can trace the basic causes of the top event faster than the horizontal approach and show how far the fault has been propagated through other units in the system.

5.2.3.1 Fault trees for design purposes

The algorithm was developed in such a way that it can be used to generate the fault trees either for design or real time purposes. Fig. 5.8 shows the flow chart of the algorithm.

The task used to generate the fault trees is BTR (Build TRees). The indirect command and the type data supplied to generate the fault trees for design purposes is shown in Fig. 5.9. Note that in this case there is an extra parameter in the command line; there is a number after a star character. This number indicates which option is required to build the fault trees. 0 is used to indicate design purposes and 1 to indicate for design purposes. The "%" character after the name of the variable, is a terminator character required by the program that reads the name, in order to know, when the input data has been concluded. The same explanation applies for the "%" character after the fault of the variable and in general to all the "%" characters that appear in this work, it is merely a terminator character required because of the way the reading program was developed.

The data supplied to BTR, must be related to the system under study and for which its topology has been previously assembled by means of the tasks described in section 5.2.2. Note that the algorithm will not be able to generate any fault tree until the topology of the system or the minitrees related to the units of the system have been defined. If a minitree or unit variable is not found by the algorithm a self-explanatory message is produced. For debugging purposes task DEB (DEBugging) can be used, it can print any of the arrays used in the data base. The type of output produced by DEB is shown in Appendix III listing 5.

To print the fault tree generated, task PTR (Print TRees) is used. Due to the restrictions of the printer available, it was not possible to print the tree in the usual tree-like format. Fig. 5.10 shows the listing produced by PTR for the top event QQ4 LO in the two pipe and valve system depicted in Fig. 4.3. It shows the tree as it was developed using the vertical approach. The tree shown in Fig. 5.11 was drawn using the data provided by the listing of Fig. 5.10.

Note that the tree shown in Fig. 5.11 was developed using the vertical approach of the algorithm and it is the same tree as the one shown in Fig. 4.17, developed manually and using the horizontal approach.

Further examples of fault trees generated by the algorithm will be presented in the next chapter.

5.2.3.2 Fault trees for real time purposes

The use of the algorithm to generate fault trees for real time purposes is a major feature of the methodology described in this work. When the algorithm was used for real time purposes the following assumptions were made in this work:

1) The algorithm has access to all the measured variables of the system under study and checks on the state of the variables are carried out at each stage.
2) Every time a scanning of the variables is carried out a snap shot is taken "to freeze" the moment. The algorithm uses the values of the variables obtained in this way to carry out the analysis of the system.
3) No malfunction of process instruments is considered at this stage.

On this basis if one or more of the variables is out of range, the algorithm will take as top event the variable with highest priority.

The priority of a variable can be fixed according to the importance of each variable as defined by the hazard analysis. If all the variables happen to have the same priority the first variable at the top of the list will be the one defined as the top event.

Once the top event is defined the algorithm starts the building of the fault tree in the same way as it does when used for design purposes, the only difference now, is that not all the branches will have to be developed up to the point of basic faults. Every time the algorithm finds a branch in which a measured variable is involved, it checks, before going into any further development, whether the variable is in fact out of range as specified by the fault which is going to be developed or not. If the state of the variable corresponds to the fault being considered the algorithm continues with the development of the branch, otherwise the algorithm does not develop that branch. The gate to which the event not developed was an input, is checked so that no contradiction can exist in the tree (AND, EX-OR gates). If any contradiction exists a "prune" is made by a "garbage collector" (see Appendix III) and the analysis is continued with any other branch of the tree pending development. The final tree obtained will show how far the fault has been traced in the system and also present to the operator the more likely basic causes of the top event.

The approach used in this work to test the algorithm for real time purposes was as follows:

1) All the measured variables were given as "OK status" value and a priority when the topology was set up by means of the tasks already mentioned in Section 5.2.2.

2) To simulate the input values from the plant a task was used. This task was RVA (Real VAlues). It allows the user to change the values of the measured variables and its priorities, so every time the values are changed a new analysis is carried out by the algorithm. A listing of the measured variables with their values, priorities and status is also produced if required by the user.

3) Every time a change in the value of the variables is made task BTR is used, with the real time option, to carry out the scanning of the variables and to generate the fault trees according to the priorities of each variable.

To illustrate the use of the algorithm for real time purposes consider the topology shown in Fig. 5.4 There are only two measured variables in this simple system, PP3 and QQ4. Fig. 5.12 shows these variables obtained with the help of RVA before their values are changed. Fig. 5.13 shows the use of RVA to change the values of PP3 and QQ4. Note that QQ4 has been given a higher priority than PP3. Also note that when the name of the variable is typed a "%" character is added, this is a terminator character to tell the program that the name is complete. (Recall internal variables). Fig. 5.14 shows the variables for the system considered after their values have been changed. Fig. 5.12 can be considered the state of the system at a time (t_1) and Fig. 5.14 at time (t_2).

When the scanning of the variables takes place at a time (t_3) the algorithm finds that there are two variables out of range but QQ4 is the one with the highest priority therefore, it is considered the top event and a fault tree is generated for QQ4 LO. Note that when the algorithm is used for real time purposes only measured variables can be subjects for top events. Fig. 5.15 shows the listing of the fault tree produced in this case. Fig. 5.16 shows the fault tree drawn using the data provided by the listing of Fig. 5.15. Note that the other measured variable that was also out of range in this example is presented in the tree as a cause of the top event.

Consider now that at time t_4 PP3 changes value again to its "steady state" value to simulate this case RVA is used again as is shown in Fig. 5.17. When the new scanning of the variables takes place at time t_5 (Fig. 5.18) the algorithm finds that there is only one variable out of range QQ4. The fault tree produced in this case is shown in the listing of Fig. 5.19 and the tree drawn by using this data is shown in Fig. 5.20. By comparing Fig. 5.16 and 5.20 it can be seen that in Fig. 5.20 the branch referring to PP3 LO has not been developed. This is because at time t_5, when the scanning of the variables was carried out, the value of PP3 was inside its limits (Status "OK").

If QQ4 returns to its "steady state" value and PP3 does not change again the algorithm will not produce any more "real time" trees until one of the measured variables changes its value. Note that the values used are fictitious ones. This simple example has been used to illustrate in some detail the use of the algorithm. Further examples for other systems will be considered in the next chapter.

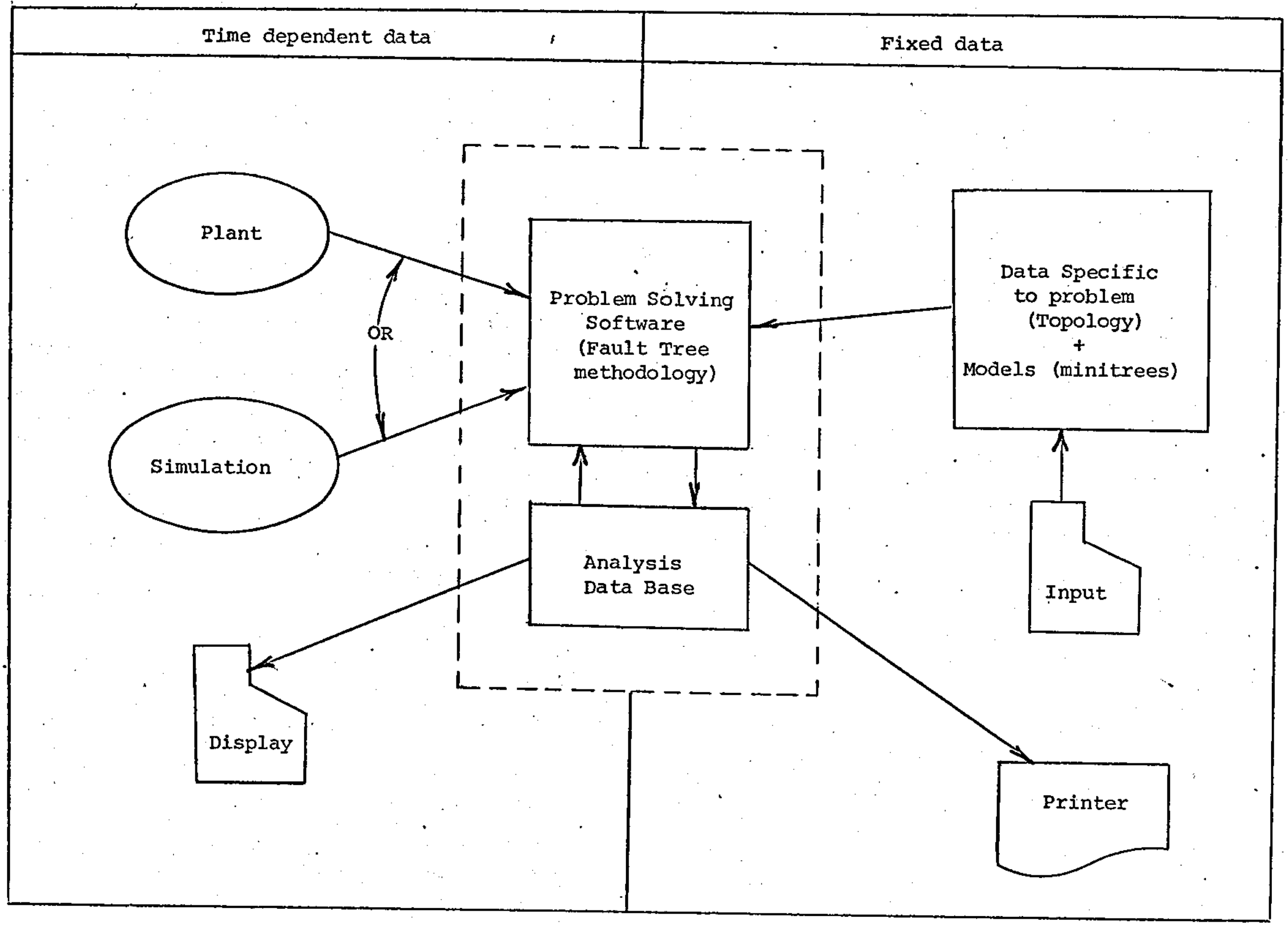

Time dependent data
Fixed data
Plant
Simulation
OR
Problem Solving Software (Fault Tree methodology)
Analysis Data Base
Data Specific to problem (Topology) + Models (minitrees)
Input
Display
Printer

Fig. 5.1 Illustration of the approach used in this work to implement the fault tree generation methodology

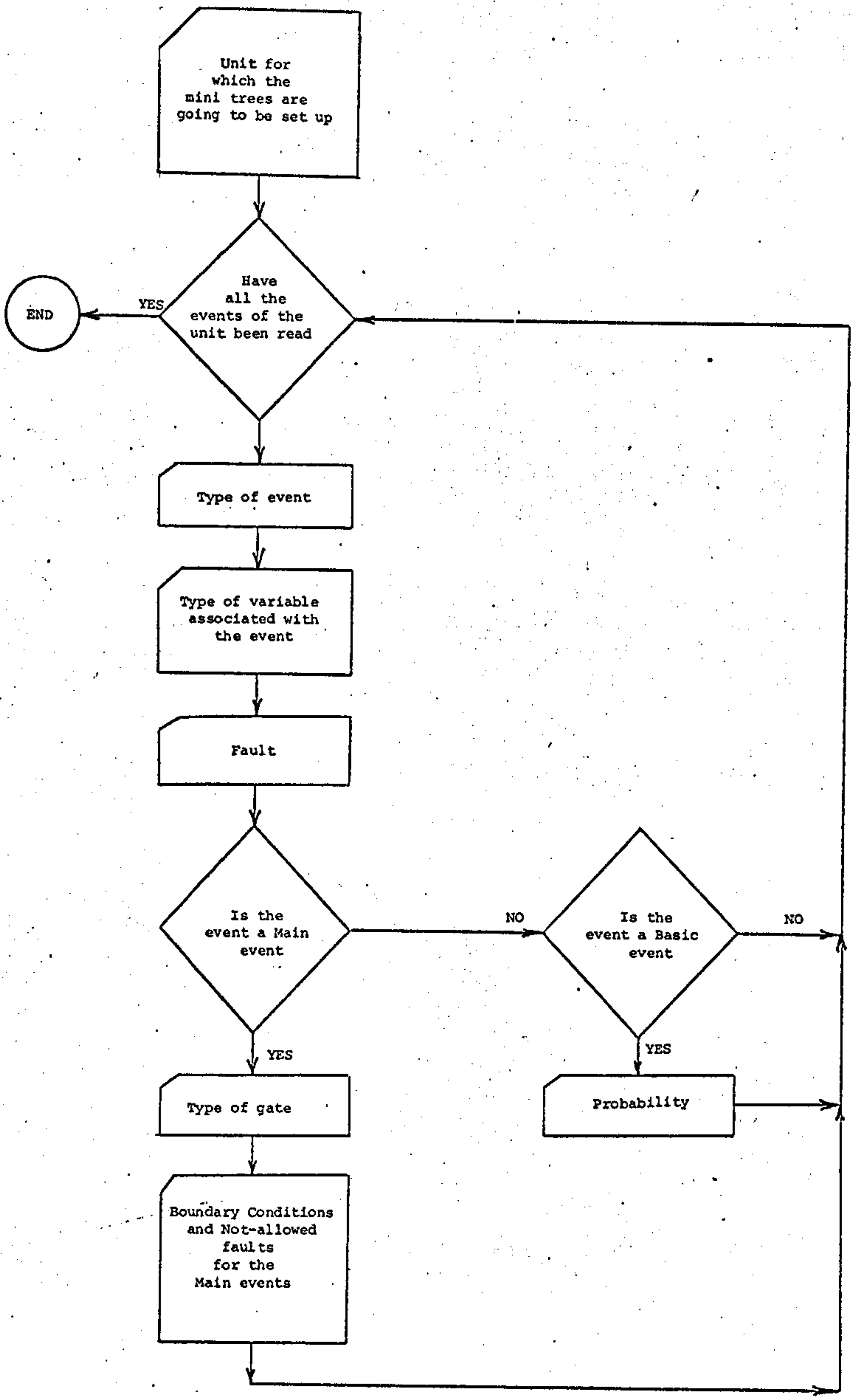

Unit for which the mini trees are going to be set up
Have all the events of the unit been read
YES
END
Type of event
Type of variable associated with the event
Fault
Is the event a Main event
NO
Is the event a Basic event
NO
YES
YES
Type of gate
Probability
Boundary Conditions and Not-allowed faults for the Main events

Fig. 5.2 Flow chart of algorithm to set up mini trees

```
PIPE%
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*@*%
T
P
1
HI%
T
P
0
LO%
B

2
FL-EX-ENV%
0.0001
M
Q
0
LO%
0
*@*%
T
P
1
LO%
T
P
0
HI%
B

2
BLOCKAGE%
0.0002
B

2
LK-LP-ENV%
0.0025
M
P
1
HI%
0
LK-LP-ENV%
*@*%
T
Q
1
HI%
T
Q
0
```

Fig. 5.3 Input data file for pipe's minitrees

/continued

```
PAGE
--------
```

```
LO%
B

2
BLOCKAGE%
0.0003
B

2
LK-HP-ENV%
0.0004
M
P
1
LO%
0
FL-EX-ENV%
*@*%
T
Q
0
HI%
T
Q
1
LO%
B

2
LK-LP-ENV%
0.0005
M
T
0
HI%
0
*@*%
T
T
1
HI%
B

2
EXT-FIRE%
0.0001
M
T
0
LO%
0
*@*%
T
T
1
LO%
M
X
0
HI%
0
```

Fig. 5.3 /continued

```
PAGE
--------
```

```
*@*%
T
X
1
HI%
M
X
0
LO%
0
*@*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*@*%
T
Q
1
NO-FLOW%
B

2
COMP-BLOC%
0.0001
M
Q
0
GT0%
0
*@*%
T
Q
1
GT0%
*
*+*+*%
```

Fig. 5.3

```
PAGE
---------
```

```
TOPOLOGY OF THE SYSTEM
*********************


******************************************************************
UNIT NO. : 1
------------
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 1
T0 : 2
VARIABLES : PP1,QQ1
******************************************************************


******************************************************************
UNIT NO. : 2
------------
TYPE OF UNIT : PIPE
** INPUT STREAMS **
FROM : 1
T0 : 2
VARIABLES : PP1,QQ1
** OUTPUT STREAMS **
FROM : 2
T0 : 3
VARIABLES : PP2,QQ2
******************************************************************


******************************************************************
UNIT NO. : 3
------------
TYPE OF UNIT : VALVE
** INPUT STREAMS **
FROM : 2
T0 : 3
VARIABLES : PP2,QQ2
** OUTPUT STREAMS **
FROM : 3
T0 : 4
VARIABLES : PP3 (M) ,QQ3
******************************************************************


******************************************************************
UNIT NO. : 4
------------
TYPE OF UNIT : PIPE
** INPUT STREAMS **
FROM : 3
T0 : 4
VARIABLES : PP3 (M) ,QQ3
** OUTPUT STREAMS **
FROM : 4
T0 : 5
VARIABLES : PP4,QQ4 (M)
******************************************************************
```

Fig. 5.4 Topology of the two pipe and valve system /continued

```
PAGE
---------
```

```
*****************************************************************
UNIT NO. : 5
-----------
TYPE OF UNIT : DUMMY-T
** INPUT STREAMS **
FROM : 4
T0 : 5
VARIABLES : PP4,QQ4 (M)
** OUTPUT STREAMS **
NONE
*****************************************************************
```

Fig. 5.4 /continued

```
PAGE
--------
```

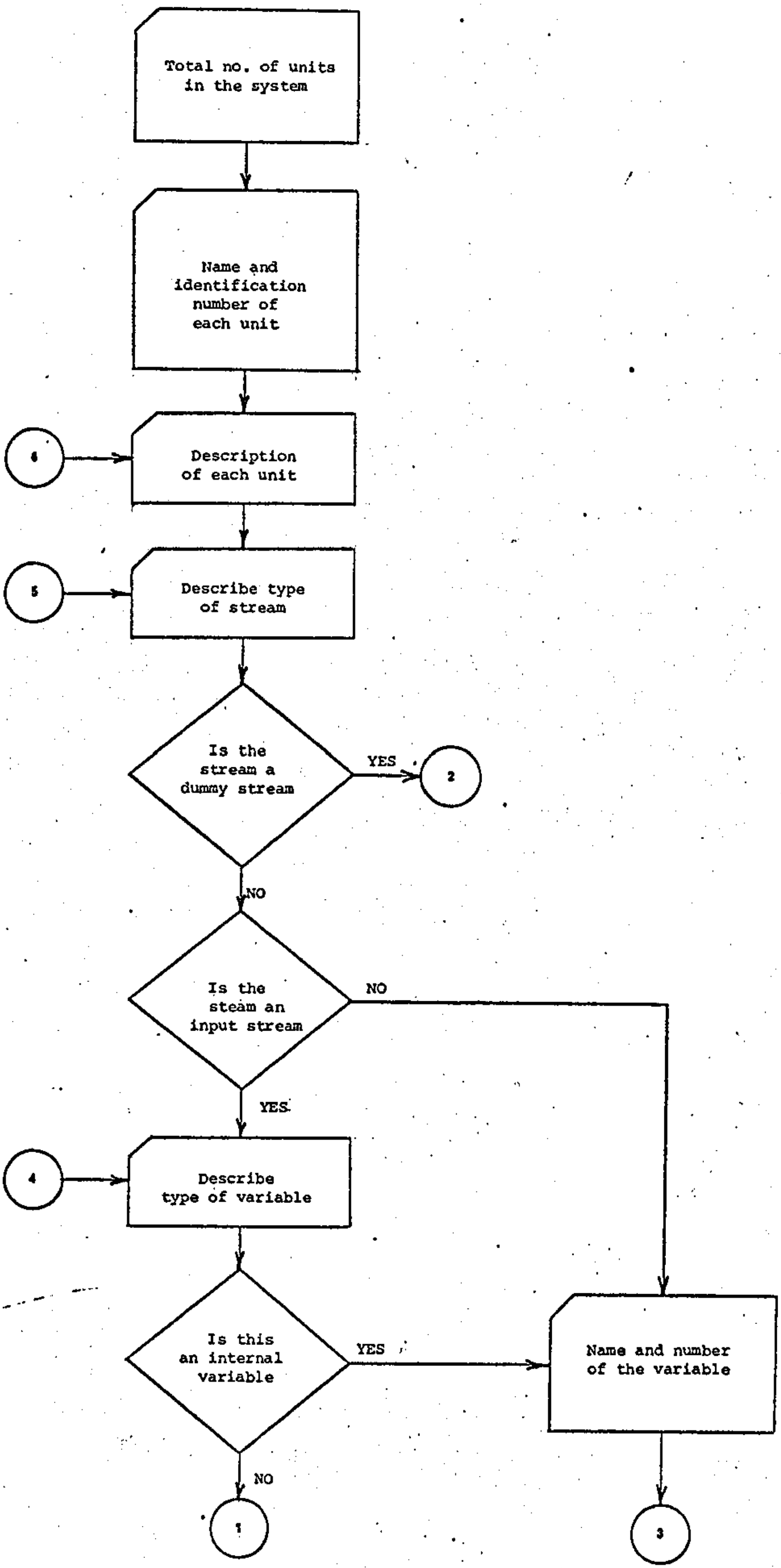

Total no. of units in the system
Name and identification number of each unit
6
Description of each unit
5
Describe type of stream
Is the stream a dummy stream
YES
2
NO
Is the steam an input stream
NO
YES
4
Describe type of variable
Is this an internal variable
YES
Name and number of the variable
NO
1
3

Fig. 5.5 Flow chart of algorithm to set up topology.

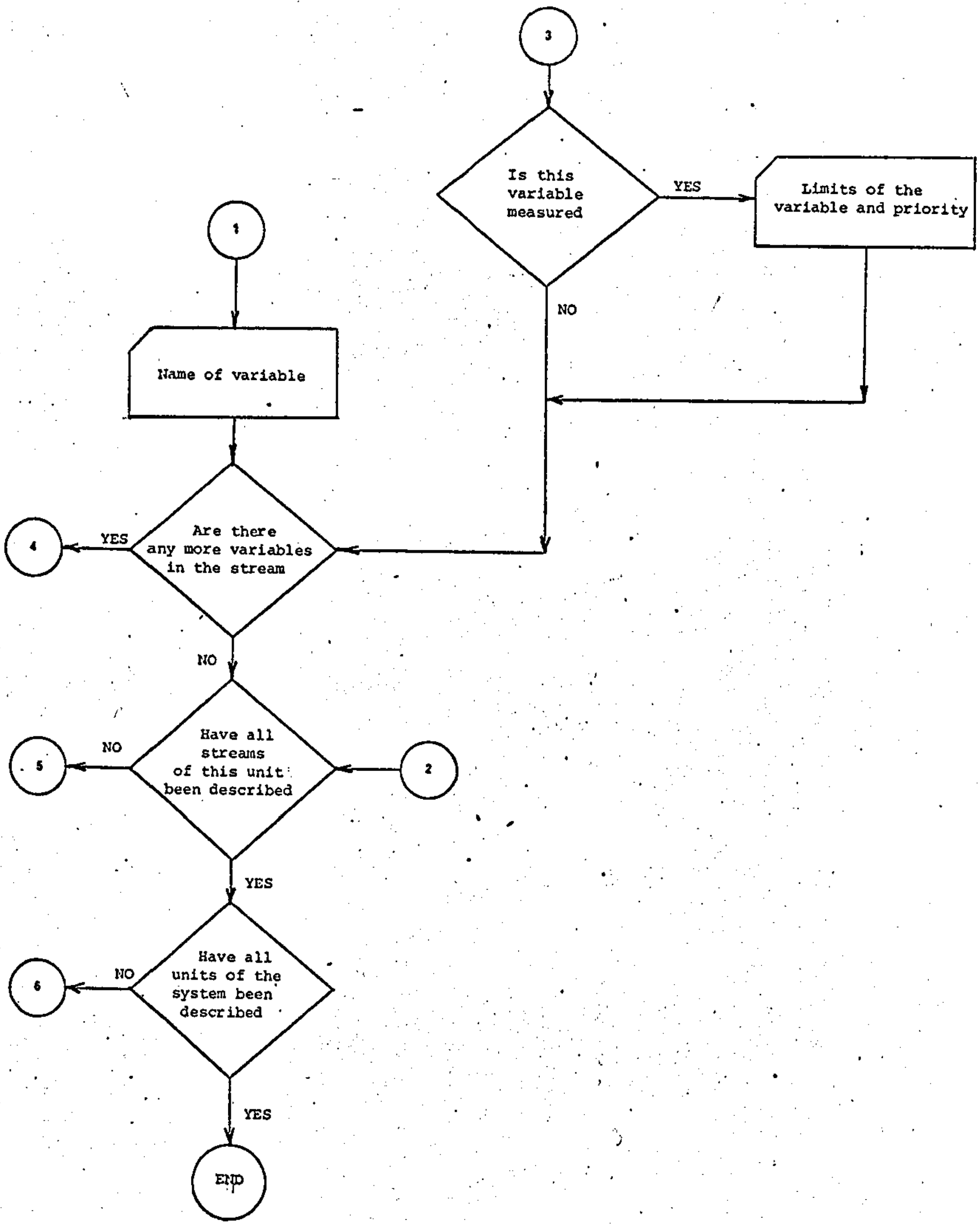

3
Is this variable measured
YES
Limits of the variable and priority
NO
1
Name of variable
Are there any more variables in the stream
YES
4
NO
Have all streams of this unit been described
NO
5
2
YES
Have all units of the system been described
NO
6
YES
END

Fig. 5.5

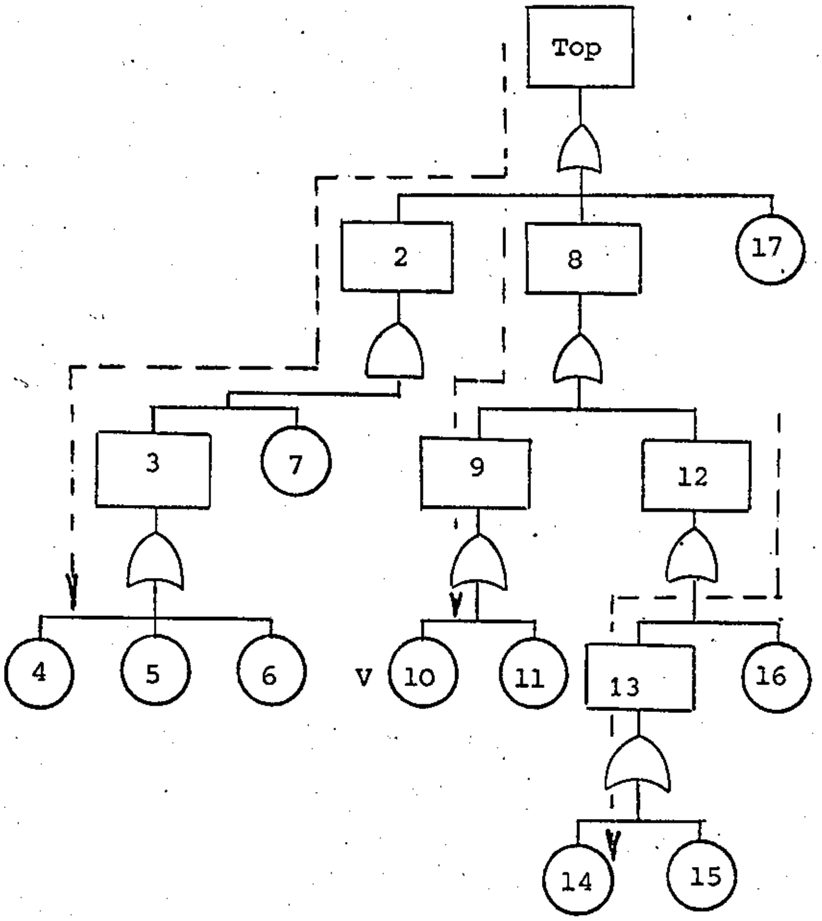


Fig. 5.6 Vertical development of a tree

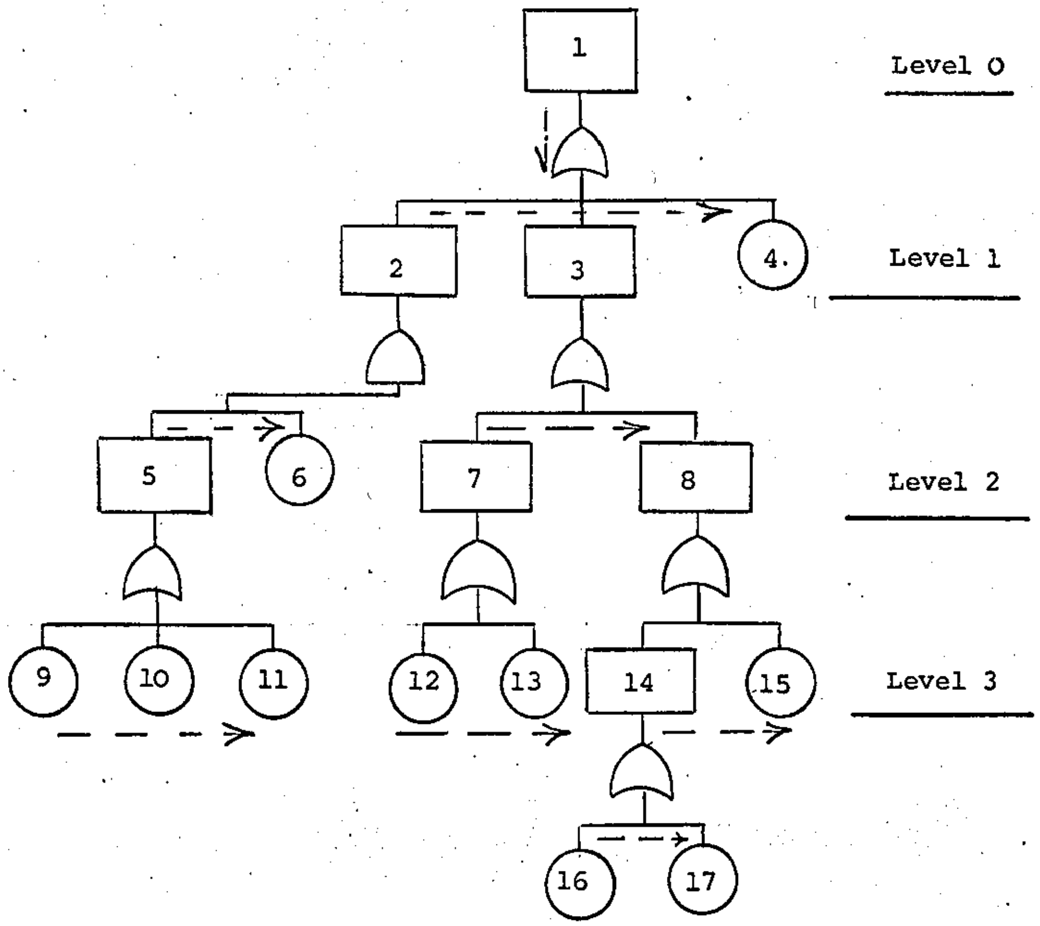


Fig. 5.7 Horizontal development of a tree

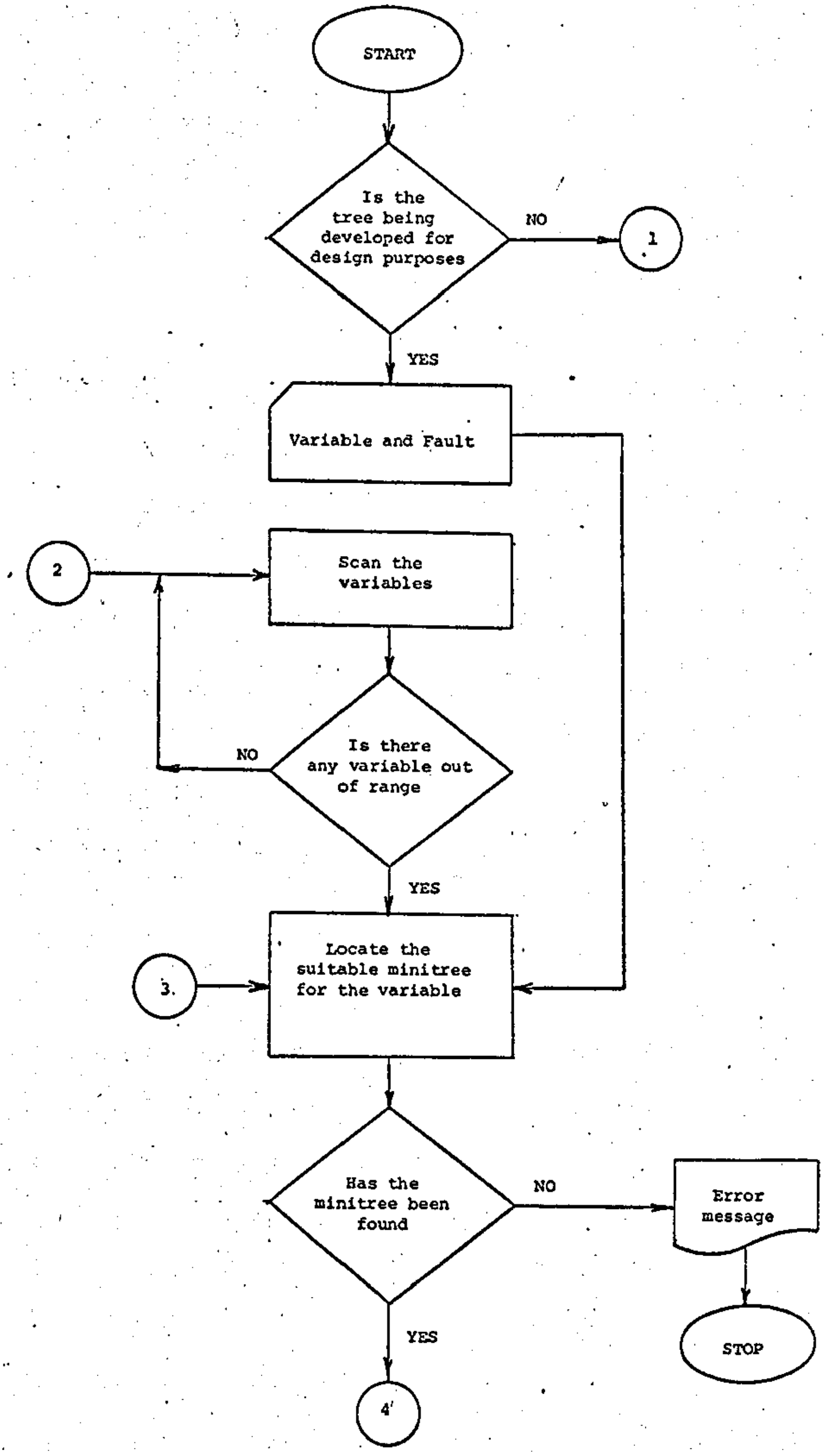

START
Is the tree being developed for design purposes
NO
1
YES
Variable and Fault
2
Scan the variables
Is there any variable out of range
NO
YES
3.
Locate the suitable minitree for the variable
Has the minitree been found
NO
Error message
STOP
YES
4

Fig. 5.8 Fault tree generation algorithm

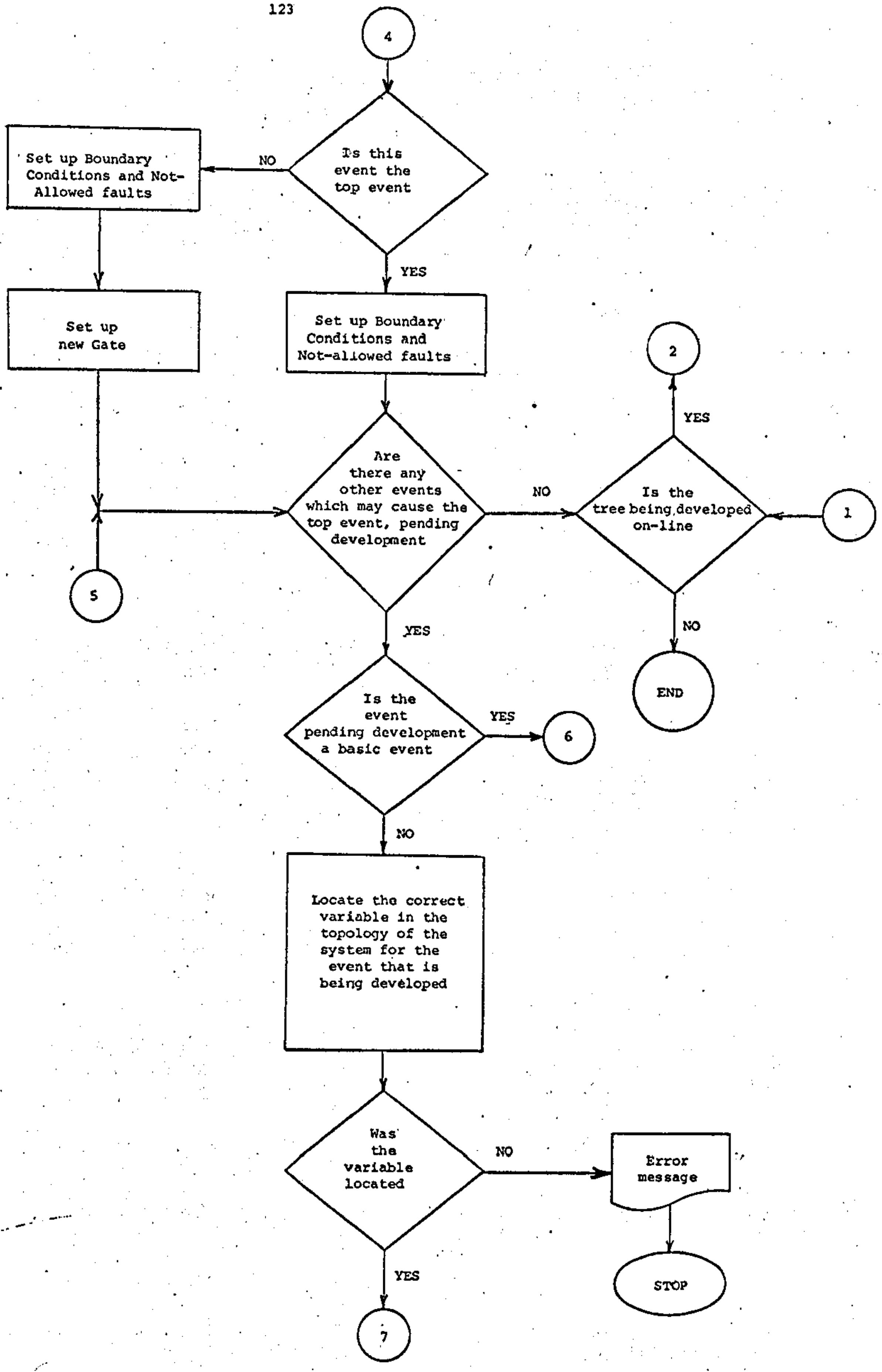

4
Is this event the top event
NO
YES
Set up Boundary Conditions and Not-Allowed faults
Set up new Gate
Set up Boundary Conditions and Not-allowed faults
Are there any other events which may cause the top event, pending development
NO
YES
5
Is the tree being developed on-line
YES
NO
2
1
END
Is the event pending development a basic event
YES
NO
6
Locate the correct variable in the topology of the system for the event that is being developed
Was the variable located
NO
YES
Error message
STOP
7

Fig. 5.8 /continued

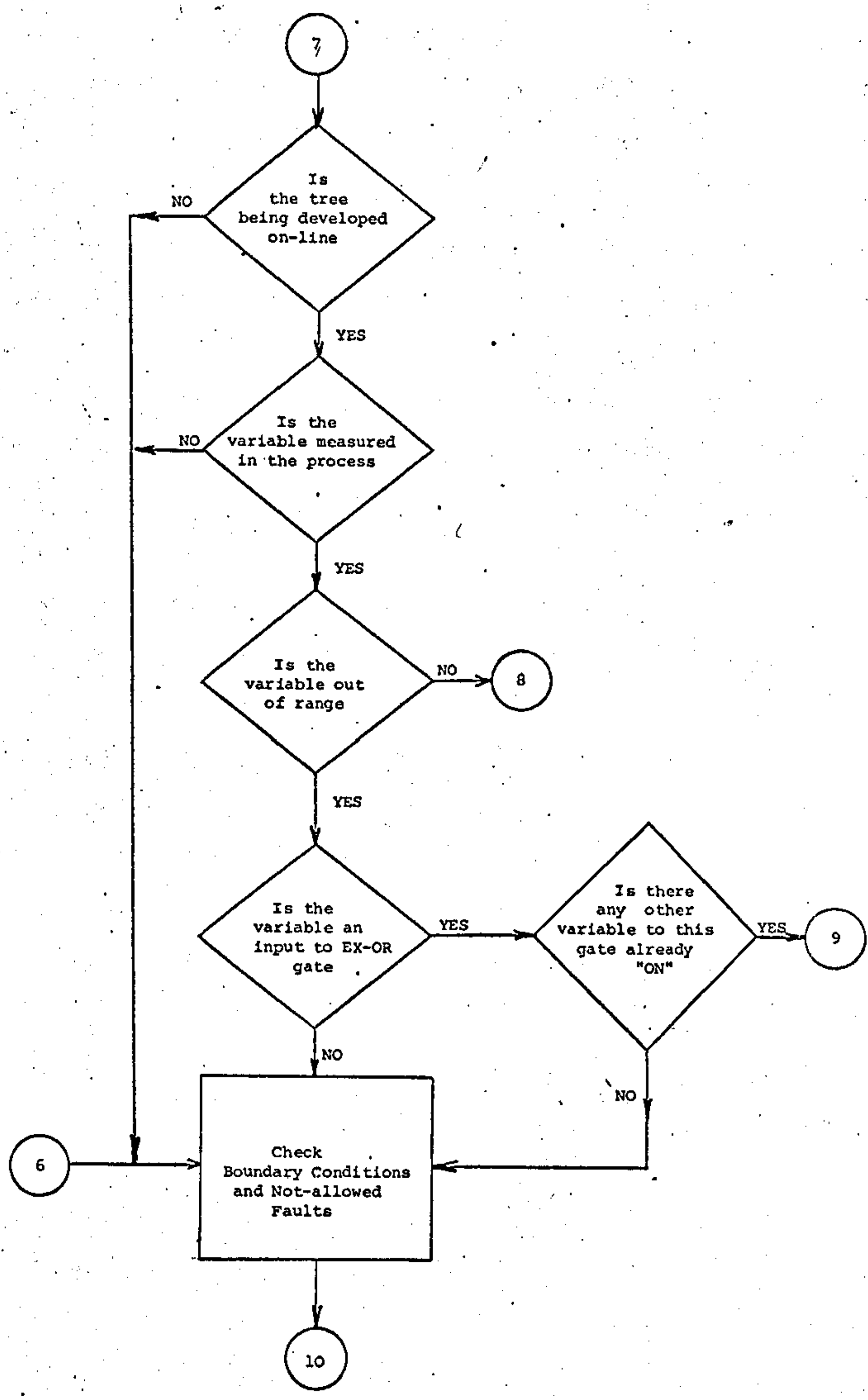

7
Is the tree being developed on-line
NO
YES
Is the variable measured in the process
NO
YES
Is the variable out of range
NO
8
YES
Is the variable an input to EX-OR gate
YES
Is there any other variable to this gate already "ON"
YES
9
NO
NO
6
Check Boundary Conditions and Not-allowed Faults
10

Fig. 5.8/continued

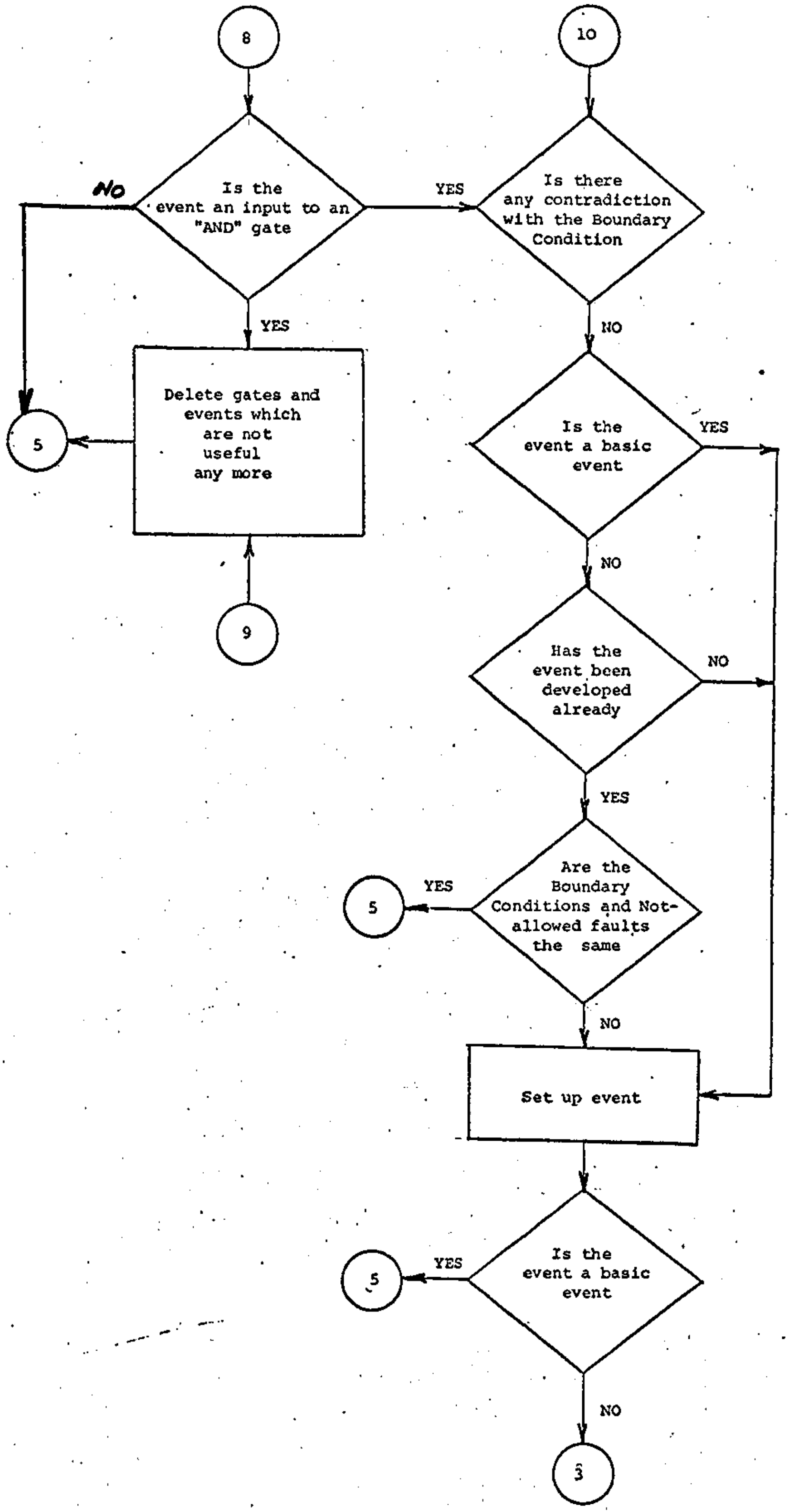

8
Is the event an input to an "AND" gate
No
YES
Delete gates and events which are not useful any more
5
9
YES
10
Is there any contradiction with the Boundary Condition
NO
Is the event a basic event
YES
NO
Has the event been developed already
NO
YES
Are the Boundary Conditions and Not-allowed faults the same
YES
5
NO
Set up event
Is the event a basic event
YES
5
NO
3

Fig. 5.8

```
>BTR TI: = TI:*0

TREE FOR DESIGN

NAME OF VARIABLE (ADD % AT THE END) ? QQ%

NO. OF VARIABLE ? 4

FAULT ? LO%
```

Fig. 5.9 Indirect command and type of input data supplied to task BTR to generate fault trees for design purposes (the underlined characters are typed by the user)

```
NOMENCLATURE :
-------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

*********************************************************************
*************************      DESIGN     *************************
*********************************************************************

                         *** FAULT TREE ***
                         ------------------


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
TOP EVENT :   QQ    4  LO         UNIT NO.   4    PIPE         OR

  T-EVENT :   PP    3  LO         UNIT NO.   4    PIPE
  T-EVENT :   PP    4  HI         UNIT NO.   4    PIPE
  B-EVENT :            BLOCKAGE   UNIT NO.   4    PIPE
  B-EVENT :            LK-LP-ENV  UNIT NO.   4    PIPE


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
    EVENT :   PP    3  LO         UNIT NO.   4    PIPE         OR

  T-EVENT :   QQ    3  LO         UNIT NO.   4    PIPE
  B-EVENT :            LK-LP-ENV  UNIT NO.   4    PIPE


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
    EVENT :   QQ    3  LO         UNIT NO.   4    PIPE         OR

  T-EVENT :   PP    2  LO         UNIT NO.   3    VALVE
  R-EVENT :            CLOSED     UNIT NO.   3    VALVE
  B-EVENT :            LK-LP-ENV  UNIT NO.   3    VALVE


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
    EVENT :   PP    2  LO         UNIT NO.   3    VALVE        OR

  T-EVENT :   QQ    2  LO         UNIT NO.   3    VALVE
  B-EVENT :            LK-LP-ENV  UNIT NO.   3    VALVE
```

Fig. 5.10 Fault tree for top event QQ4 LO in a two pipe and valve system

```
PAGE
---------
```

/continued

```
                NAME NO.
                VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
                ---- ---- -----      ------------------       ----
    EVENT :      QQ    2  LO        UNIT NO.  3    VALVE       OR

 T-EVENT :      PP    1  LO        UNIT NO.  2    PIPE
 B-EVENT :                BLOCKAGE  UNIT NO.  2    PIPE
 B-EVENT :                LK-LP-ENV UNIT NO.  2    PIPE


                NAME NO.
                VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
                ---- ---- -----      ------------------       ----
    EVENT :      PP    1  LO        UNIT NO.  2    PIPE        OR

 T-EVENT :      QQ    1  LO        UNIT NO.  2    PIPE
 B-EVENT :                LK-LP-ENV UNIT NO.  2    PIPE


                NAME NO.
                VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
                ---- ---- -----      ------------------       ----
    EVENT :      QQ    1  LO        UNIT NO.  2    PIPE
                                      *DIAMOND EVENT*


                NAME NO.
                VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
                ---- ---- -----      ------------------       ----
    EVENT :                CLOSED    UNIT NO.  3    VALVE       OR

 B-EVENT :                BLOCKAGE  UNIT NO.  3    VALVE
 B-EVENT :                SHUT      UNIT NO.  3    VALVE


                NAME NO.
                VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
                ---- ---- -----      ------------------       ----
    EVENT :      PP    4  HI        UNIT NO.  4    PIPE
                                      *DIAMOND EVENT*
```

Fig. 5.10

```
PAGE
--------
```

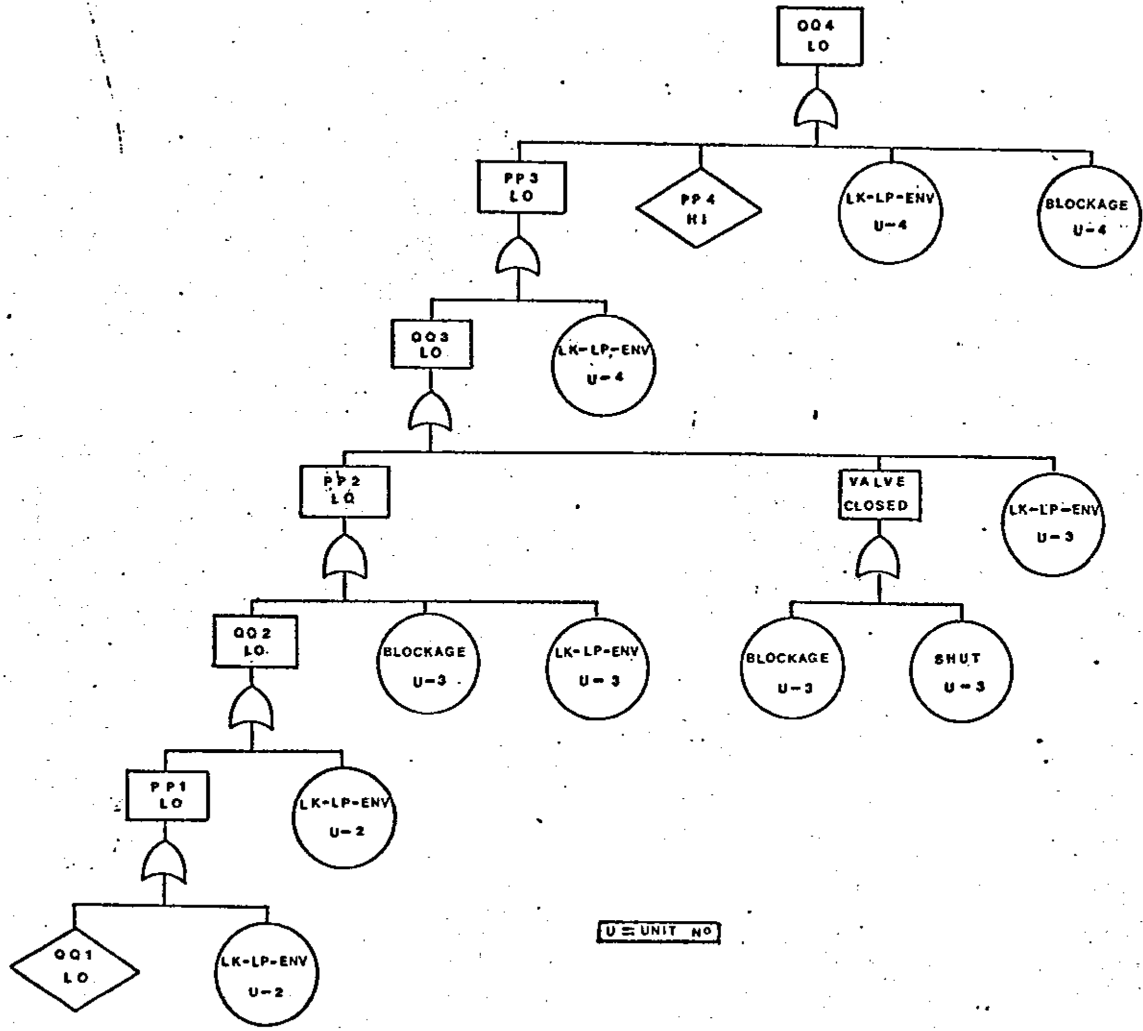


Fig. 5.11 Fault tree for QQ4 LO drawn using the data provided by the listing of Fig. 5.10

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ? Y


                        MEASURED VARIABLES
                        ******************

      VARIABLES    HI LIMIT    LO LIMIT    STATUS    PRIORITY
      ---------    --------    --------    ------    --------

         PP  3        100          0         OK           1
         QQ  4        100          0         OK           2

READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 5.12 Measured variables obtained with the help of RVA before their values are changed

```
PAGE
--------
```

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ? N


READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? PP%

NO. OF VARIABLE ? 3

DO YOU WANT TO CHANGE THE LIMITS(Y/N) ? N

VALUE OF VAR. ? 5

NAME OF VAR.(ADD %,* TO TERM.) ? QQ%

NO. OF VARIABLE ? 4

DO YOU WANT TO CHANGE THE LIMITS(Y/N) ? N

VALUE OF VAR. ? 2

NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 5.13 Use of task RVA to change the values of PP3 and QQ4

```
PAGE
--------
```

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ?


                              MEASURED VARIABLES
                              ******************

        VARIABLES    HI LIMIT    LO LIMIT    STATUS    PRIORITY
        ---------    --------    --------    ------    --------
        .........    ........    ........    ......    ........

          PP  3         100          10        LO          1
          QQ  4           5           1        LO          2

READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ?
```

Fig. 5.14 The measured variables after their values have been changed

```
PAGE
---------
.........
```

```
NOMENCLATURE :
--------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

*************************************************************
************************   REAL   TIME   ********************
*************************************************************

                       *** FAULT TREE ***
                       ------------------


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
             ---- ---- -----      -------------------      ----
TOP EVENT :   QQ    4  LO        UNIT NO.  4   PIPE        OR

  T-EVENT :   PP    3  LO        UNIT NO.  4   PIPE
  T-EVENT :   PP    4  HI        UNIT NO.  4   PIPE
  B-EVENT :            BLOCKAGE  UNIT NO.  4   PIPE
  B-EVENT :            LK-LP-ENV UNIT NO.  4   PIPE


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
             ---- ---- -----      -------------------      ----
    EVENT :   PP    3  LO        UNIT NO.  4   PIPE        OR

  T-EVENT :   QQ    3  LO        UNIT NO.  4   PIPE
  B-EVENT :            LK-LP-ENV UNIT NO.  4   PIPE


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
             ---- ---- -----      -------------------      ----
    EVENT :   QQ    3  LO        UNIT NO.  4   PIPE        OR

  T-EVENT :   PP    2  LO        UNIT NO.  3   VALVE
  R-EVENT :            CLOSED    UNIT NO.  3   VALVE
  B-EVENT :            LK-LP-ENV UNIT NO.  3   VALVE


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
             ---- ---- -----      -------------------      ----
    EVENT :   PP    2  LO        UNIT NO.  3   VALVE       OR

  T-EVENT :   QQ    2  LO        UNIT NO.  3   VALVE
  B-EVENT :            LK-LP-ENV UNIT NO.  3   VALVE
```

Fig. 5.15 Listing of the fault tree QQ4 LO for Real time

```
PAGE
--------
```

/continued

```
               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT      GATE
               ---- ---- -----       -------------------      ----
     EVENT :    QQ    2  LO        UNIT NO.   3    VALVE      OR

  T-EVENT :     PP    1  LO        UNIT NO.   2    PIPE
  B-EVENT :              BLOCKAGE  UNIT NO.   2    PIPE
  B-EVENT :              LK-LP-ENV UNIT NO.   2    PIPE


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT      GATE
               ---- ---- -----       -------------------      ----
     EVENT :    PP    1  LO        UNIT NO.   2    PIPE       OR

  T-EVENT :     QQ    1  LO        UNIT NO.   2    PIPE
  B-EVENT :              LK-LP-ENV UNIT NO.   2    PIPE


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT      GATE
               ---- ---- -----       -------------------      ----
     EVENT :    QQ    1  LO        UNIT NO.   2    PIPE
                                    *DIAMOND EVENT*


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT      GATE
               ---- ---- -----       -------------------      ----
     EVENT :             CLOSED    UNIT NO.   3    VALVE      OR

  B-EVENT :              BLOCKAGE  UNIT NO.   3    VALVE
  B-EVENT :              SHUT      UNIT NO.   3    VALVE


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT      GATE
               ---- ---- -----       -------------------      ----
     EVENT :    PP    4  HI        UNIT NO.   4    PIPE
                                    *DIAMOND EVENT*
```

Fig. 5.15 Listing of the fault tree QQ4 LO for Real time

```
PAGE
--------
```

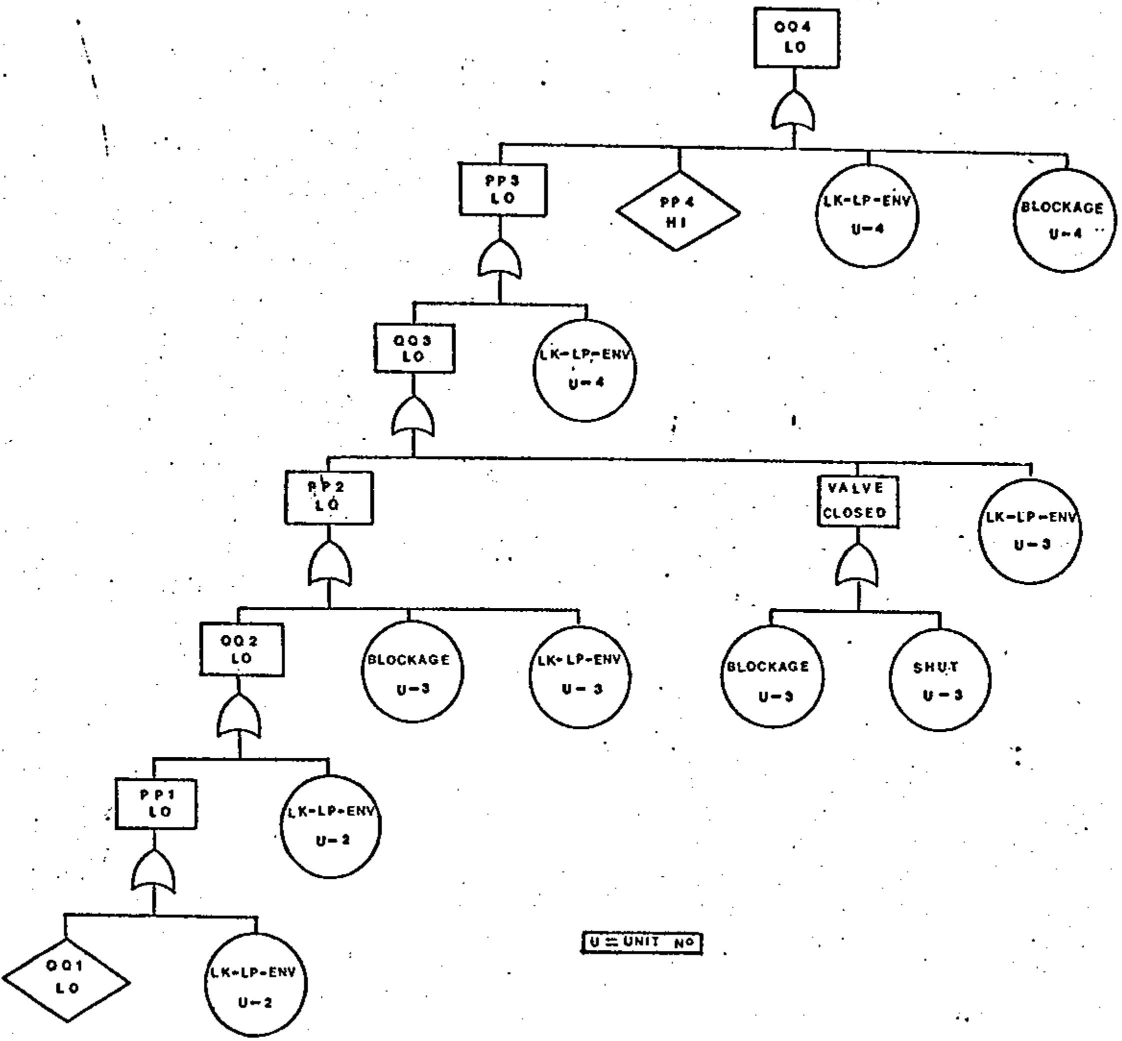


Fig. 5.16 Fault tree for QQ4 LO drawn using the data provided by the listing of Fig. 5.15

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ? Y


                               MEASURED VARIABLES
                               ******************

        VARIABLES    HI LIMIT    LO LIMIT    STATUS    PRIORITY
        ---------    --------    --------    ------    --------
        .........    ........    ........    ......    ........

          PP  3         122         12         LO          1
          QQ  4           5          1         LO          2

READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? PP%

NO. OF VARIABLE ? 3

DO YOU WANT TO CHANGE THE LIMITS(Y/N) ? N

VALUE OF VAR. ? 22


NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 5.17 Use of task RVA to change the value of PP3 to a "steady state" value

```
PAGE
--------
........
```

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ? Y


                        MEASURED VARIABLES
                        ******************


        VARIABLES    HI LIMIT    LO LIMIT    STATUS    PRIORITY
        ---------    --------    --------    ------    --------

          PP  3         122         12         OK          1
          QQ  4           5          1         LO          2


READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 5.18 State of measured variables at time (t5)

```
PAGE
--------
```

```
NOMENCLATURE :
-------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

****************************************************************
************************   REAL   TIME   ************************
****************************************************************

                          *** FAULT TREE ***
                          ------------------


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT      GATE
               ---- ---- -----       ------------------       ----
TOP EVENT :     QQ    4  LO          UNIT NO.  4    PIPE      OR

  T-EVENT :     PP    4  HI          UNIT NO.  4    PIPE
  B-EVENT :              BLOCKAGE    UNIT NO.  4    PIPE
  B-EVENT :              LK-LP-ENV   UNIT NO.  4    PIPE


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT      GATE
               ---- ---- -----       ------------------       ----
    EVENT :     PP    4  HI          UNIT NO.  4    PIPE
                                      *DIAMOND EVENT*
```

Fig. 5.19 Listing of the fault tree produced by the methodology when only QQ4 is out of range

```
PAGE
---------
```

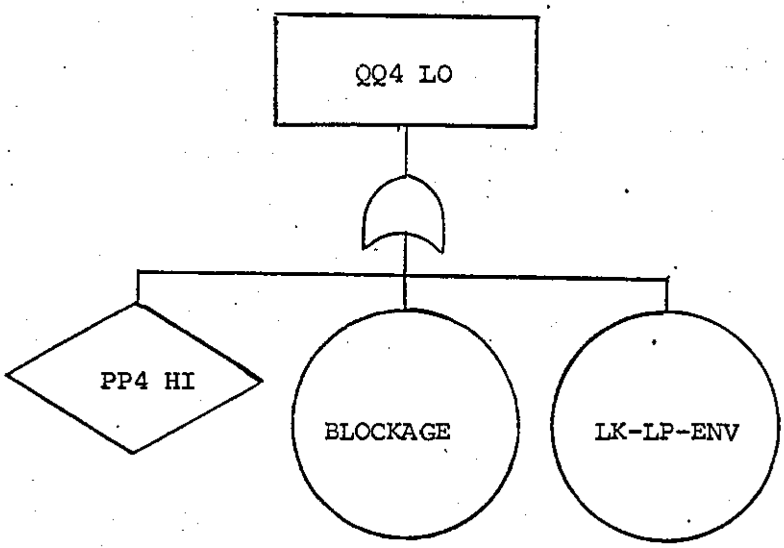


Fig. 5.20 Fault tree for QQ4 LO drawn using the data provided by the listing of Fig. 5.19

Chapter 6

APPLICATIONS OF THE METHODOLOGY

The previous chapters have presented the development of the conceptual basis of a methodology for fault tree generation and its implementation on a digital computer. In this chapter three different systems are used to illustrate the applications of the methodology.

6.1 Flow Control System

One of the most common examples employed to illustrate the use of control loops in a Chemical Process is a flow control system like the one shown in Fig. 6.1. The system is a simple one, but it was considered appropriate to be used as the first example, to illustrate how the methodology handles a control loop.

The first step in the generation of the fault trees for this system is to follow the procedure already mentioned in the previous chapters. Fig. 6.2 shows the same flow control system but now described according to the requirements of the methodology. The limits of the system are determined by the dummy units and the name of the variables have been described according to the names used in the unit models described in Appendix I. The next step is to set up the minitrees and the topology of the system, before the actual construction of the fault trees can be carried out. To do this two indirect command files are used the first one is BTREES. This indirect command file is the one that carries out the installation of all the tasks used by the methodology. Once the tasks have been installed, BTREES uses some of them, to link the arrays of the data base and provide the basic information required by the second indirect command file. BTREES is always required as the first step in setting up any system. Fig. 6.3 shows this file.

The second indirect command file, (in this case TLAP) is the one that carries out the setting up of the unit minitrees that form the particular system under study. It also sets up the topology of the system. Fig. 6.4 shows the file TLAP used for this first example. Note that it makes use of the tasks BMT (to set up the minitrees), DES and REU (to set up the topology) already described in Chapter 5. All the output files are written on the disk and then deleted, leaving in core only the minitrees of the units forming the specific system and its topology. This information is later required to generate the fault trees. Task PRI can be used now to check if the topology kept in core is correct. Fig. 6.5 shows the output provided by PRI for the topology of the flow control system. The minitrees can also be checked by means of task PMT, examples of the results provided by PMT are shown in Appendix I.

From the topology shown in Fig. 6.5 it can be seen that this particular example does not have measured variables defined. Due to this, it will only be used in the construction of fault trees for design purposes.

Consider as top event QQ3 HI. Once the top event has been chosen task BTR is used to produce the fault tree for this particular event. Fig. 6.6 shows the indirect command used and the data given to BTR in order to produce the fault tree desired. Note that in this case, the option (0) for design purposes is the one used. The resultant tree can be printed by using task PTR. Fig. 6.7 shows the listing obtained and Fig. 6.8 shows the fault tree for QQ3 HI in a tree-like format drawn using the data of Fig. 6.7.

This example was also used by Lapp and Powers$^{(L5)}$ to illustrate their methodology. It is interesting to compare their results with the ones obtained here. The best way to do this is by means of comparison of the cut sets. Tables 6.1 and 6.2 show the results of the two methodologies. It can be seen that there are some differences in these tables. The reasons for these differences is that the methodologies are based on different assumptions and models. The methodology used in this work does not consider in the development of the minitrees any component installed in a improper way such as reversed valve; only the failure modes of the units are considered important from the point of view of design. If the assumptions made in Chapter 3, for the development of the minitrees are relaxed, the reversed faults can easily be included in the unit minitrees and therefore the cut set "reversed valve", may be obtained.

When the use of the unit models to develop the minitrees was presented in Chapter 3 only two states of the variables were considered (HI and LO). This does not mean that they are the only possible states of the variables. If any other state of a variable requires to be considered it can be done without any problem. Each new state of the variables will require a new minitree. The way to introduce new minitrees is by carrying out a failure analysis for the new state of the variable in a similar fashion as for the other minitrees presented in Chapter 3 and Appendix I. For the example being considered, the (+10) state used by Lapp & Powers can easily be defined as a Very High (VHI) fault and a new minitree for each unit could be developed, so that when the fault tree is constructed the cut set QQ3VHI may be obtained. In this example this fault was not considered because it implies that the sudden change in QQ1 should be very fast indeed to avoid any action of the control loop. Further examples where other states of the variables besides HI and LO are considered will be presented in the following examples. Note that the methodology used here uses two-way models to ensure the flow of information. It allows the construction of fault trees for other top events that are not at the downstream end of the system. This will be illustrated with several examples in the next sections of this chapter.

Note in this example the presence of the AND gate. It comes from the control valves minitrees. Its input events were developed using the flow of information through the system. Recall that all the different gates of the minitrees are developed when the failure analysis of the unit model is carried out. The algorithm used to produce the fault tree only links the minitrees, it does not create any events or gates. The quality of the fault tree is closely related to the quality of the minitrees.

6.2 Two tank and control valve system

Consider now the example shown in Fig. 6.9. This is a bigger system intended to illustrate the way in which the methodology can handle more complicated systems. In this system all the variables are assumed to be measured, therefore, it can be used to produce fault trees, either for design or real time purposes.

The indirect command files used to define the system of Fig. 6.9 are BTREES and TOCON. The second file is the one specific to the system being considered and is shown in Fig. 6.10. The output provided by the task PRI for the topology is shown in Fig. 6.11. Note that two internal variables are involved in this system.

Consider first the system for design purposes. Figs. 6.12, 6.14 and 6.16 show the listing produced by the task PTR for the fault trees of three top events. Fig. 6.12 refers to the top event QQ7 HI, the fault tree shown in Fig. 6.13 was drawn using the data of Fig. 6.12. The tree for top event QQ5 LO is shown in Fig. 6.15, it was drawn using the data provided by the listing of Fig. 6.14. Finally the listing of Fig. 6.16 refers to the top event PP3 HI, the fault tree shown in Fig. 6.17 is based on the data shown in Fig. 6.16.

Note in Figs. 6.15 and 6.17(a) how the flow of information travels downstream and upstream providing a complete picture of how a fault propagates throughout the system. Without a two-way flow of information this would not be possible. Fig. 6.17(b) shows how the tree for P3 HI would be if only one-way flow of information were used in the development of the tree. It is in this aspect that the methodology used in this work has advantage over the methodology used by Salem(S1) et al. Their methodology only allows the development of the trees in one direction. The capacity of constructing fault trees for top events occurring at any place of the system under study is without doubt a major feature of the methodology used in this work. No other methodology reported in the literature has claimed this feature.

Consider now the tank and control valve system for real time purposes. The state of the variables before any change is carried out is shown in Fig. 6.18, this listing was obtained with the help of task RVA. Fig. 6.19 shows the same measured variables but after RVA has been used to change some of their values and priorities. This is the picture that the task BTR finds when it starts the scanning of the variables. There are several variables out of range but the one with highest priority in this example is QQ7 and therefore it is considered the top event to be developed. The tree shown in Fig. 6.21 was obtained from the listing of Fig. 6.20. Note that the tree shows how far the fault of the top event has propagated through other units, only those measured variables that were out of range when the "snap shot" (shown in Fig. 6.19) was taken, are portrayed in the tree. The tree also shows, according to the information available at the moment of building the tree, which are the more likely causes of the top event. This information can be of great help to the operator in the control room when he has to make a decision about the problem that he is facing at that very moment. New trees can be produced every time the scanning is carried out and measured variables found out of range.

6.3 Heat Exchanger and Control Loops System

The last example to be considered in this chapter is a little more complicated than the former two already discussed. Consider now the system shown in Fig. 6.22. It represents a system that can easily be found in many Chemical Process plants. The function of the process considered in this example is to cool the hot stream to a specific temperature, before it can be used in other parts of the plant. Water is pumped to cool the hot stream. A trip valve has been placed at the inlet of the heat exchanger's hot stream. The valve will be activated by a signal, from the flow sensor installed at the output stream of the pump when the flow is stopped, due either to the shutdown of the pump or to any other cause. Note that there are two control loops in this system and that other variables, besides pressure and flow, are considered.

This system has units with more than one input/output stream and will be quite useful to illustrate the importance of the complete variables to assure the correct development of the fault trees. Other states of the variables such as NO FLOW and FLOW GREATER THAN 0 (GT0) are also considered in this example.

The initial steps before the actual constructions of the trees, have to be carried out in a similar fashion to the former examples discussed in this chapter. The indirect commands are again, a general one BTREES, and a specific one for the system under study. Fig. 6.23 shows the specific command file TOT4 used for the Heat Exchanger and control loops system. Fig. 6.24 shows the topology of the system obtained by means of task PRI. From Fig. 6.24 it can be seen that some of the variables have been considered as measured ones; therefore both options of the methodology can be illustrated using the example described in Fig. 6.22.

Consider first the example for design purposes. TT4 HI will be the top event in this case. Task BTR is used with the design option (0). Fig. 6.25 shows the listing obtained by means of task PTR and Fig. 6.26 the fault tree for TT4 HI in a tree-like format. As in the previous examples the tree-like format uses the data produced by PTR. Note that in the listing of Fig. 6.25 there are two R events with the same dummy fault C but by means of the description of the unit, provided in the same listing, any possibility of confusion is excluded. As a rule when the minitrees are developed for a specific unit, no two R events can have the same dummy fault unless they are the same events. One thing that is useful when the trees are drawn using the data provided by PTR, is to remember that the listing was produced in the same way as the tree was developed, following the vertical approach.

The tree shown in Fig. 6.26 shows clearly how the methodology handles those units, like the heat exchanger, with more than one input/output stream. Note the presence of the complete variables such as CT5, AP5, QB7 in the tree and how the flow of information travels through the system without any problem in spite of the presence of the control loops. Also note the presence in the tree of two states of the variable flow that had not appeared before in any of the other examples considered. (These states are highlighted in the tree of

Fig. 6.26). A failure analysis as discussed in Chapter 3, was carried out in each unit to obtain the minitrees for NO-FLOW and FLOW GREATER THAN 0. (In this case simplified to consider only the flow variable). These minitrees are shown in Appendix I for each unit. Note that the same set of minitrees has been used in all the examples discussed in this chapter. The methodology only uses those minitrees that are required according to the type of event that is being developed. It can be seen from this that the methodology is flexible. This feature allows the use of more general models capable of coping with the multi-state nature of the variables involved in the Chemical Process.

The difference between Fussell's methodology and the one used in this work is without doubt the capability of dealing with multistate variables. Fussell usually only considers two states because his methodology is aimed at electrical systems where the only states considered are ON and OFF. This example shows that Fussell's methodology would not be applicable here, because more than two states were involved. The states of the flow variable considered in Fig. 6.26 are:

1) Flow HI.
2) Flow LO.
3) NO-FLOW.
4) Flow present in the system (GT0).

An example quite similar to the one discussed here has been presented in the literature by Lapp & Powers.(L4) Most of the results shown in their paper are obtained here, but the differences in this case are due to the introduction of a sensor (unit 9) in this example (that theirs does not have) and the use of models that are more general than the ones used by them in their methodology. The tree obtained in this case was for an event that was at the downstream end of the system. None of Lapp and Powers' papers have shown trees for top events in the middle of the system as the ones shown in the last section for the two tank and valve system.

The methodology described in this work gives the safety analyst a greater capability in the synthesis of fault trees than the one presented by Lapp & Powers. The flexibility to introduce new states of the variables together with the choice of top events throughout the system under study, provides a useful tool at the design stage of a process.

Note in Fig. 6.26 that the AND gates that appear there, come from the unit minitrees used. No special consideration is required for the loops. Using the flow of information the methodology plugs the correct minitree together to produce the final tree.

Consider now the real time option for the same system. Fig. 6.27 shows the state of the measured variables before RVA is used to change their values. Fig. 6.28 shows the same variables after their values have been changed. Note that the variable TT4 has now the highest priority to ensure that the tree developed in this case has TT4 as top event. The listing obtained after BTR and PTR have been used is shown in Fig. 6.29. The fault tree obtained from the data of Fig. 6.29 is shown in Fig. 6.30.

Note again that when the real time option is used, the methodology does not develop all the possible branches of the tree, only those variables that were out of range when the tree was developed are included. By comparing the trees of Figs. 6.26 and 6.30 the difference between design trees and real time trees can be appreciated. Note that in the real time tree only one state of the variable is present and this is because in real time the variable has a defined value. Therefore all the other options that appear in a design tree cannot appear in a real time one, and are ruled out by the Boundary Conditions.

The number of units that the methodology can handle is only limited by hardware restrictions of the computer used. In this work the data base was restricted to 8K by the hardware and as a consequence the maximum number of units in a system is restricted to 15. The example discussed in this section consists of 12 units including the dummy ones and with the facilities available for this work it was possible to produce fault trees for design purposes with 60 gates and more than 100 events. The space available becomes a problem when the methodology is used for design purposes for larger systems, but Real time trees do not require so much space, because they will never be as large as the equivalent design ones.

Table 6.1

Minimal cut sets of the tree shown in Fig. 6.7

1) Valve fails open
2) Controller fails High
3) Set point High (WW1 HI)
4) Sensor fails Low
5) QQ1 High AND valve stuck
6) QQ1 High AND sensor stuck
7) QQ1 High AND controller stuck
8) QQ1 High AND valve on manual
9) Leakage from High Pressure Environment in valve

Table 6.2

Minimal cut sets from Lapp & Powers Example$^{(L5)}$

1) QQ1 (+10)
2) Valve fails open
3) Valve reversed
4) Controller fails High
5) Set point High
6) Controller Reversed
7) Line 4 ruptured
8) Sensor fails low
9) Sensor Reversed
10) QQ1 (+1) AND valve stuck
11) QQ1 (+1) AND sensor stuck
12) QQ1 (+1) AND controller stuck
13) QQ1 (+1) and ON Manual

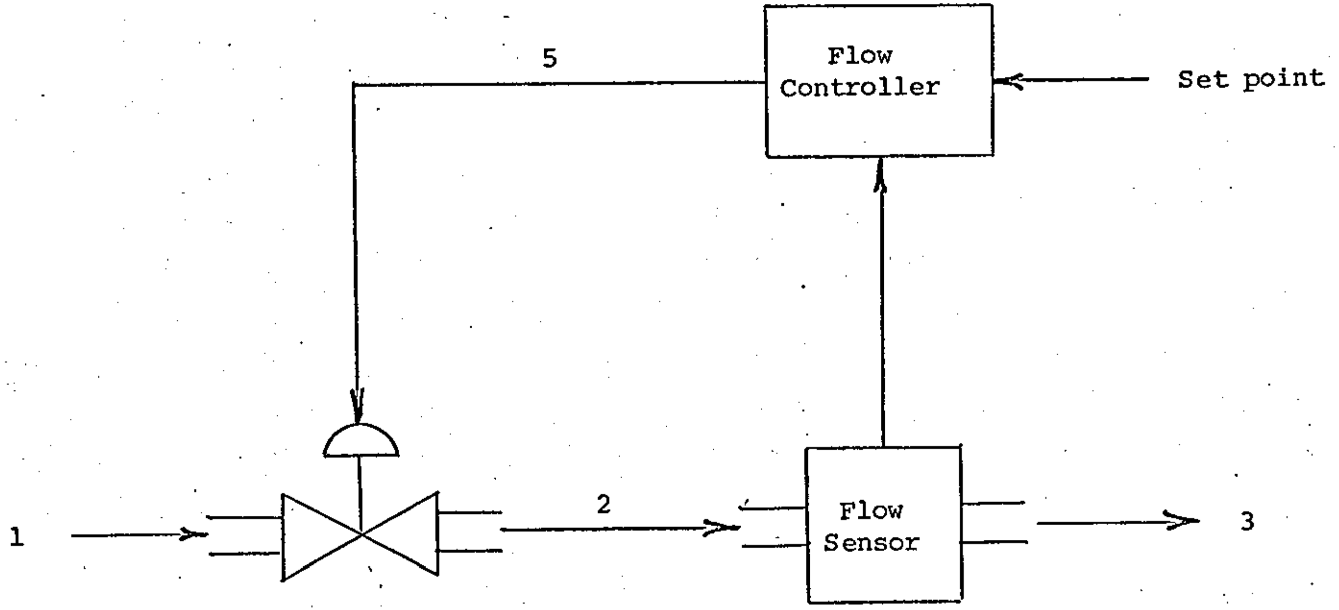
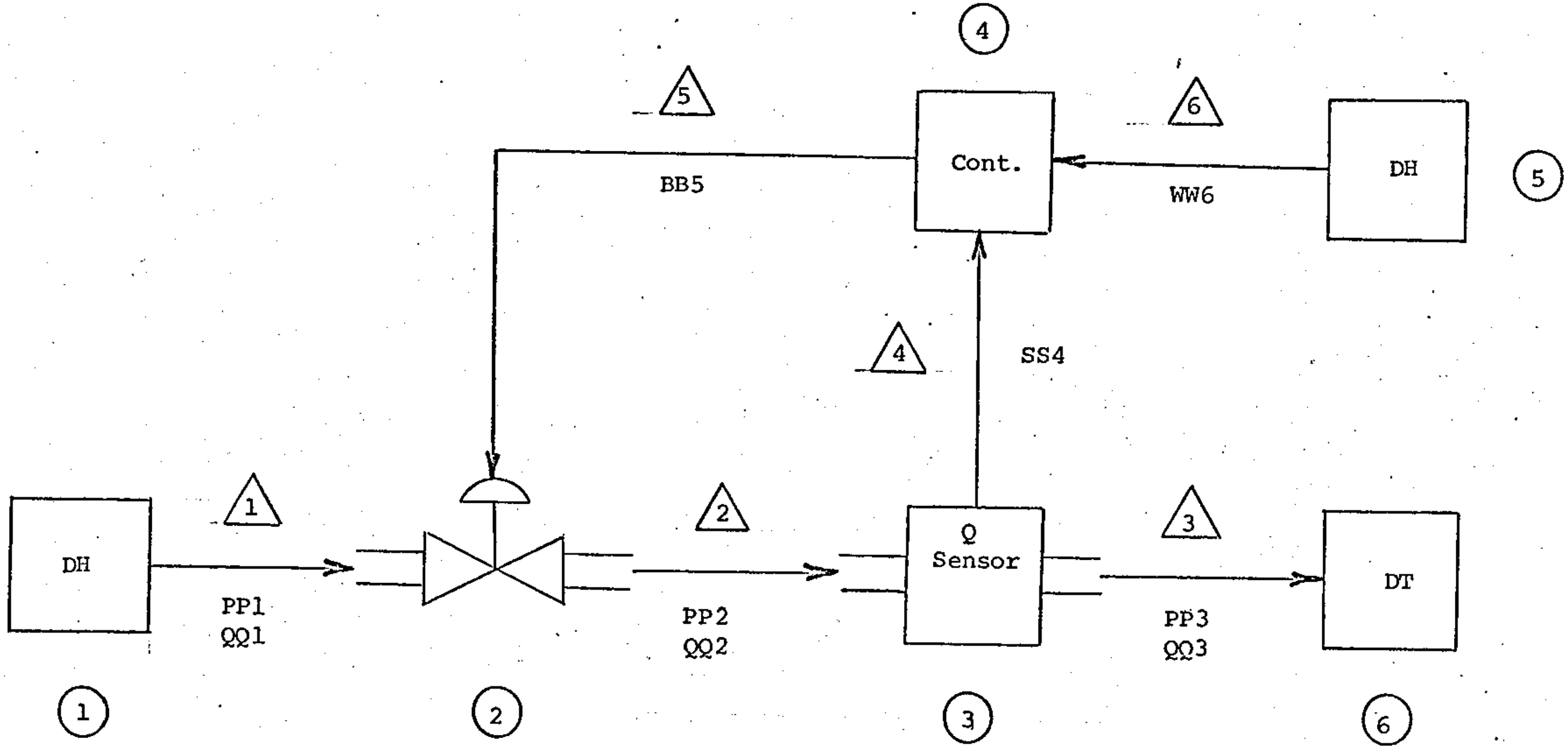


Fig. 6.1 Flow control system



△ Number of stream and variable

○ Number of unit

Fig. 6.2 Flow control system described according to the methodology used in this work

```
INS [1,1]DATBAS
INS [1,1]FTREES
INS BMT
INS BTR
INS CAL
INS DEB
INS DES
INS LIT
INS PMT
INS PRI
INS PTR
INS REU
INS RFA
INS RTU
INS RVA
CAL TI:=TI:*2
LIT TI:=TI:
RFA DK2:R.TNT;1=DK2:FAULTS.DAT
RTU DK2:T.TNT;1=DK0:TYPE.DAT
PIP *.TNT;*/DE
```

Fig. 6.3 BTREES file to install all the tasks used by the methodology

```
PAGE                          BTREES.CMD
```

```
BMT DK2:A1.TNT;1=DK2:DUMMYH.DAT
BMT DK2:B1.TNT;1=DK2:DUMMYT.DAT
BMT DK2:C.TNT;1=DK2:CONTVA.DAT
BMT DK2:D.TNT;1=DK2:CONTLL.DAT
BMT DK2:E.TNT;1=DK2:SENSW.DAT
REU DK2:A.TNT;1=DK2:LAF.DAT
DES DK2:B.TNT;1=DK2:LAPC.DAT
PIP *.TNT;*/DE
```

Fig. 6.4 <u>TLAP file used to define the minitrees and topology of the system</u>

```
PAGE                    TLAP  .CMD
```

```
TOPOLOGY OF THE SYSTEM
**********************


******************************************************************
UNIT NO. : 1
------------
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 1
T0 : 2
VARIABLES : PP1,QQ1
******************************************************************


******************************************************************
UNIT NO. : 2
------------
TYPE OF UNIT : CNTRL-VAL
** INPUT STREAMS **
FROM : 1
T0 : 2
VARIABLES : PP1,QQ1
FROM : 4
T0 : 2
VARIABLES : BB5
** OUTPUT STREAMS **
FROM : 2
T0 : 3
VARIABLES : PP2,QQ2
******************************************************************


******************************************************************
UNIT NO. : 3
------------
TYPE OF UNIT : SENSOR-Q
** INPUT STREAMS **
FROM : 2
T0 : 3
VARIABLES : PP2,QQ2
** OUTPUT STREAMS **
FROM : 3
T0 : 4
VARIABLES : SS4
FROM : 3
T0 : 6
VARIABLES : PP3,QQ3
******************************************************************


******************************************************************
UNIT NO. : 4
------------
TYPE OF UNIT : CNTROLLER
** INPUT STREAMS **
FROM : 3
T0 : 4
```

Fig. 6.5 <u>Topology of the flow control system</u>

/continued

```
PAGE .
---------
```

```
VARIABLES : SS4
FROM : 5
T0.: 4
VARIABLES : WW6
** OUTPUT STREAMS **
FROM : 4
T0 : 2
VARIABLES : BB5
******************************************************************


******************************************************************
UNIT NO. : 5
------------
TYPE"OF"UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 5
T0 : 4
VARIABLES : WW6
******************************************************************


******************************************************************
UNIT NO. : 6
------------
TYPE"OF"UNIT : DUMMY-T
** INPUT STREAMS **
FROM : 3
T0 : 6
VARIABLES : PP3,QQ3
** OUTPUT STREAMS **
NONE
******************************************************************
```

Fig. 6.5

```
PAGE
---------
```

```
>BTR.TI:=TI:*0

TREE FOR DESIGN

NAME OF VARIABLE (ADD % AT THE END) ? QQ%

NO. OF VARIABLE  ? 3

FAULT ? HI%
```

Fig. 6.6 Indirect command and data required to produce the fault tree for top event QQ3 HI for a flow control system (the underlined characters are typed by the user)

```
NOMENCLATURE :
--------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

*******************************************************************
**************************     DESIGN     *************************
*******************************************************************

                          *** FAULT TREE ***
                          ------------------


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
TOP EVENT :  QQ   3   HI         UNIT NO.  3   SENSOR-Q    OR

  T-EVENT :  PP   2   HI         UNIT NO.  3   SENSOR-Q
  T-EVENT :  PP   3   LO         UNIT NO.  3   SENSOR-Q


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
    EVENT :  PP   2   HI         UNIT NO.  3   SENSOR-Q    OR

  T-EVENT :  QQ   2   HI         UNIT NO.  3   SENSOR-Q


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
    EVENT :  QQ   2   HI         UNIT NO.  3   SENSOR-Q    OR

  R-EVENT :           A          UNIT NO.  2   CNTRL-VAL
  T-EVENT :  BB   5   HI         UNIT NO.  2   CNTRL-VAL
  B-EVENT :           FAIL-OPEN  UNIT NO.  2   CNTRL-VAL
  B-EVENT :           LK-HP-ENV  UNIT NO.  2   CNTRL-VAL


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
    EVENT :           A          UNIT NO.  2   CNTRL-VAL   AND

  T-EVENT :  PP   1   HI         UNIT NO.  2   CNTRL-VAL
  R-EVENT :           C          UNIT NO.  2   CNTRL-VAL
```

Fig. 6.7 Fault tree for QQ3 HI in a flow control system

/continued

```
PAGE
--------
```

```
              NAME NO.
              VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
              ---- ---- -----       ------------------         ----
   EVENT :     PP    1  HI        UNIT NO.   2    CNTRL-VAL    OR

 T-EVENT :     QQ    1  HI        UNIT NO.   2    CNTRL-VAL


              NAME NO.
              VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
              ---- ---- -----       ------------------         ----
   EVENT :     QQ    1  HI        UNIT NO.   2    CNTRL-VAL
                                    *DIAMOND EVENT*


              NAME NO.
              VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
              ---- ---- -----       ------------------         ----
   EVENT :              C         UNIT NO.   2    CNTRL-VAL    OR

 T-EVENT :     BB    5  NO-CHANGE UNIT NO.   2    CNTRL-VAL
 B-EVENT :              VALV-STCK UNIT NO.   2    CNTRL-VAL
 B-EVENT :              MANUAL    UNIT NO.   2    CNTRL-VAL


              NAME NO.
              VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
              ---- ---- -----       ------------------         ----
   EVENT :     BB    5  NO-CHANGE UNIT NO.   2    CNTRL-VAL    OR

 T-EVENT :     SS    4  NO-CHANGE UNIT NO.   4    CNTROLLER
 B-EVENT :              CONT-STCK UNIT NO.   4    CNTROLLER


              NAME NO.
              VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
              ---- ---- -----       ------------------         ----
   EVENT :     SS    4  NO-CHANGE UNIT NO.   4    CNTROLLER    OR

 B-EVENT :              SENS-STCK UNIT NO.   3    SENSOR-Q


              NAME NO.
              VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
              ---- ---- -----       ------------------         ----
   EVENT :     BB    5  HI        UNIT NO.   2    CNTRL-VAL    OR

 T-EVENT :     SS    4  LO        UNIT NO.   4    CNTROLLER
 T-EVENT :     WW    6  HI        UNIT NO.   4    CNTROLLER
 B-EVENT :              CONT-F-HI UNIT NO.   4    CNTROLLER


              NAME NO.
              VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
```

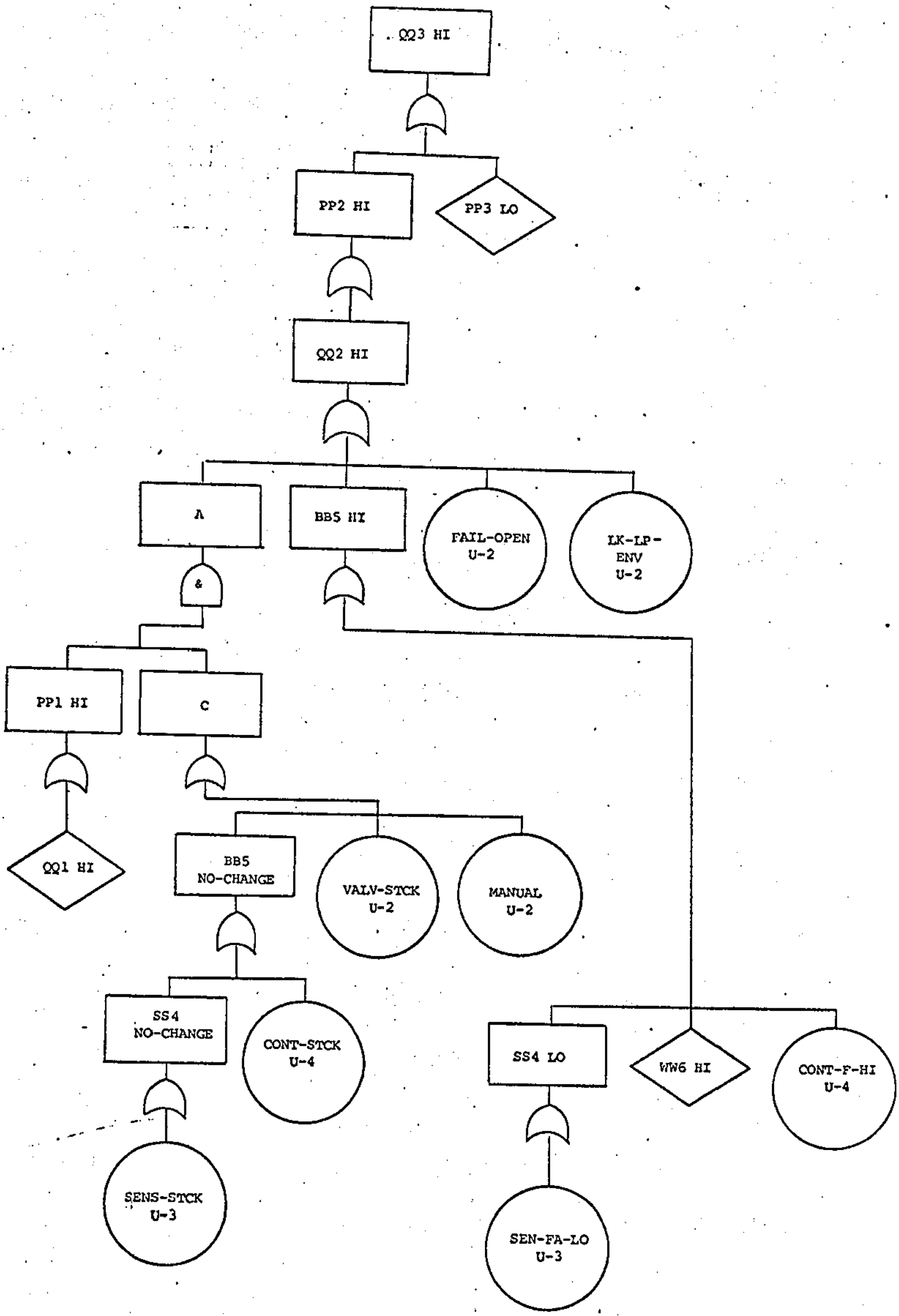
Fig. 6.7 /continued

```
PAGE
--------
```

```
                ----  ----  -----        -----------------     ----
  EVENT   :     SS     4    LO           UNIT NO.  4   CNTROLLER  OR

B-EVENT   :                 SEN-FA-LO UNIT NO.  3   SENSOR-Q


                NAME  NO.
                VAR.  VAR.  FAULT         DESCRIPTION OF UNIT     GATE
                ----  ----  -----         -----------------       ----
  EVENT   :     WW     6    HI           UNIT NO.  4   CNTROLLER
                                          *DIAMOND EVENT*


                NAME  NO.
                VAR.  VAR.  FAULT         DESCRIPTION OF UNIT     GATE
                ----  ----  -----         -----------------       ----
  EVENT   :     PP     3    LO           UNIT NO.  3   SENSOR-Q
                                          *DIAMOND EVENT*
```

Fig. 6.7

```
PAGE
---------
```



U = UNIT Number

Fig. 6.8 Fault tree for QQ3 HI in a flow control system drawn using the data of Fig. 6.7

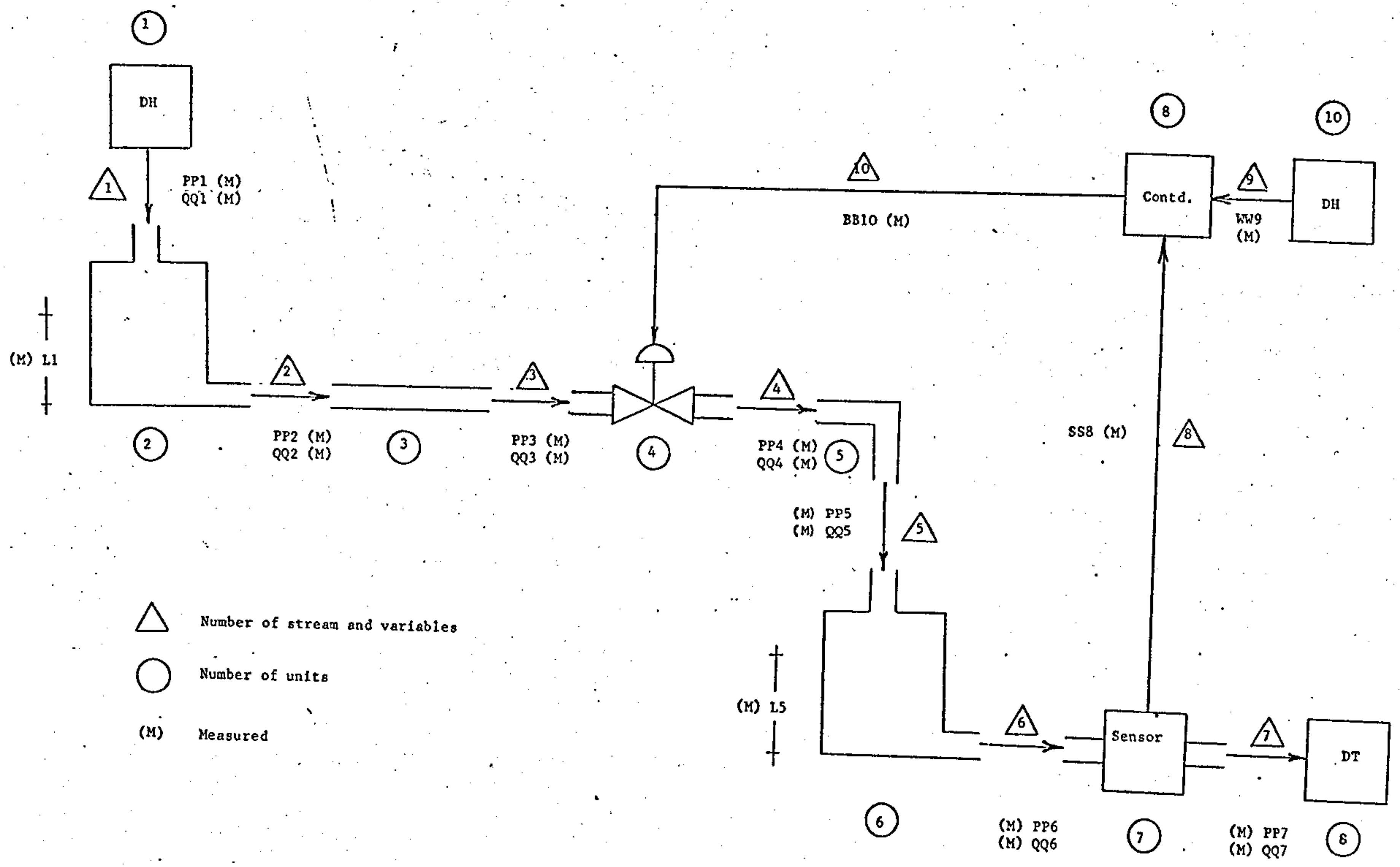


Fig. 6.9 Two tank and control valve system

```
BMT DK2:A1.TNT;1=DK2:CLTK.DAT
BMT DK2:A2.TNT;1=DK2:PIPE.DAT
BMT DK2:A3.TNT;1=DK2:DUMMYH.DAT
BMT DK2:B1.TNT;1=DK2:DUMMYT.DAT
BMT DK2:C.TNT;1=DK2:CONTVA.DAT
BMT DK2:D.TNT;1=DK2:CONTLL.DAT
BMT DK2:E.TNT;1=DK2:SENSQ.DAT
REU DK2:A.TNT;1=DK2:TOCV.DAT
DES DK2:B.TNT;1=DK2:TOCVC.DAT
PIP *.TNT;*/DE
```

Fig. 6.10 TOCON file used to set up the minitrees and topology of a two tank and control valve system

```
PAGE                     TOCON .CMD
--------
```

```
TOPOLOGY OF THE SYSTEM
**********************


***********************************************************
UNIT NO. : 1
------------
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 1
TO : 2
VARIABLES : PP1 (M) ,QQ1 (M)
***********************************************************


***********************************************************
UNIT NO. : 2
------------
TYPE OF UNIT : CLOSED-TK
** INPUT STREAMS **
FROM : 1
TO : 2
VARIABLES : PP1 (M) ,QQ1 (M)
INTERNAL VARIABLES : L 1 (M)
** OUTPUT STREAMS **
FROM : 2
TO : 3
VARIABLES : PP2 (M) ,QQ2 (M)
***********************************************************


***********************************************************
UNIT NO. : 3
------------
TYPE OF UNIT : PIPE
** INPUT STREAMS **
FROM : 2
TO : 3
VARIABLES : PP2 (M) ,QQ2 (M)
** OUTPUT STREAMS **
FROM : 3
TO : 4
VARIABLES : PP3 (M) ,QQ3 (M)
***********************************************************


***********************************************************
UNIT NO. : 4
------------
TYPE OF UNIT : CNTRL-VAL
** INPUT STREAMS **
FROM : 3
TO : 4
VARIABLES : PP3 (M) ,QQ3 (M)
FROM : 8
TO : 4
VARIABLES : BB12 (M)
** OUTPUT STREAMS **
```

Fig. 6.11 Topology of a two tank and control valve system

/continued

```
PAGE
--------
```

```
FROM : 4
T0 : 5
VARIABLES : PP4 (M) ,QQ4 (M)
*****************************************************************


*****************************************************************
UNIT NO. : 5
------------
TYPE OF UNIT : PIPE
** INPUT STREAMS **
FROM : 4
T0 : 5
VARIABLES : PP4 (M) ,QQ4 (M)
** OUTPUT STREAMS **
FROM : 5
T0 : 6
VARIABLES : PP5 (M) ,QQ5 (M)
*****************************************************************


*****************************************************************
UNIT NO. : 6
------------
TYPE OF UNIT : CLOSED-TK
** INPUT STREAMS **
FROM : 5
T0 : 6
VARIABLES : PP5 (M) ,QQ5 (M)
INTERNAL VARIABLES : L 5 (M)
** OUTPUT STREAMS **
FROM : 6
T0 : 7
VARIABLES : PP6 (M) ,QQ6 (M)
*****************************************************************


*****************************************************************
UNIT NO. : 7
------------
TYPE OF UNIT : SENSOR-Q
** INPUT STREAMS **
FROM : 6
T0 : 7
VARIABLES : PP6 (M) ,QQ6 (M)
** OUTPUT STREAMS **
FROM : 7
T0 : 8
VARIABLES : SS8 (M)
FROM : 7
T0 : 9
VARIABLES : PP7 (M) ,QQ7 (M)
*****************************************************************


*****************************************************************
UNIT NO. : 8
------------
TYPE OF UNIT : CNTROLLER
** INPUT STREAMS **
FROM : 7
```

Fig. 6.11 /continued

```
PAGE
---------
```

```
T0 : 8
VARIABLES : SS8 (M)
FROM : 10
T0 : 8
VARIABLES : WW9 (M)
** OUTPUT STREAMS **
FROM : 8
T0 : 4
VARIABLES : BB10 (M)
***********************************************************************

***********************************************************************
UNIT NO. : 9
------------
TYPE OF UNIT : DUMMY-T
** INPUT STREAMS **
FROM : 7
T0 : 9
VARIABLES : PP7 (M) , QQ7 (M)
** OUTPUT STREAMS **
NONE
***********************************************************************

***********************************************************************
UNIT NO. : 10
------------
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 10
T0 : 8
VARIABLES : WW9 (M)
***********************************************************************
```

Fig. 6.11

```
PAGE
---------
```

```
NOMENCLATURE :
-------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

*********************************************************************
************************   DESIGN   ************************
*********************************************************************

                         *** FAULT TREE ***
                         ------------------


              NAME NO.
              VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
              ---- ---- -----      -------------------      ----
TOP EVENT :    QQ    7  HI        UNIT NO.  7   SENSOR-Q     OR

  T-EVENT :    PP    6  HI        UNIT NO.  7   SENSOR-Q
  T-EVENT :    PP    7  LO        UNIT NO.  7   SENSOR-Q


              NAME NO.
              VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
              ---- ---- -----      -------------------      ----
    EVENT :    PP    6  HI        UNIT NO.  7   SENSOR-Q     OR

  T-EVENT :    QQ    6  HI        UNIT NO.  7   SENSOR-Q


              NAME NO.
              VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
              ---- ---- -----      -------------------      ----
    EVENT :    QQ    6  HI        UNIT NO.  7   SENSOR-Q     OR

  T-EVENT :    PP    5  HI        UNIT NO.  6   CLOSED-TK
  B-EVENT :             FL-EX-ENV UNIT NO.  6   CLOSED-TK


              NAME NO.
              VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
              ---- ---- -----      -------------------      ----
    EVENT :    PP    5  HI        UNIT NO.  6   CLOSED-TK    OR

  T-EVENT :    L     5  HI        UNIT NO.  6   CLOSED-TK
  B-EVENT :             LK-HP-ENV UNIT NO.  6   CLOSED-TK


              NAME NO.
```

Fig. 6.12 Fault tree for top event QQ7 HI in a two tank and control valve system

```
PAGE
----
```

/continued

```
         VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
         ---- ---- -----      -------------------       ----
  EVENT : L      5 HI        UNIT NO.  6   CLOSED-TK    OR

T-EVENT : QQ     5 HI        UNIT NO.  6   CLOSED-TK
B-EVENT :          LK-HP-ENV UNIT NO.  6   CLOSED-TK


         NAME NO.
         VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
         ---- ---- -----      -------------------       ----
  EVENT : QQ     5 HI        UNIT NO.  6   CLOSED-TK    OR

T-EVENT : PP     4 HI        UNIT NO.  5   PIPE
B-EVENT :          FL-EX-ENV UNIT NO.  5   PIPE


         NAME NO.
         VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
         ---- ---- -----      -------------------       ----
  EVENT : PP     4 HI        UNIT NO.  5   PIPE         OR

T-EVENT : QQ     4 HI        UNIT NO.  5   PIPE
B-EVENT :          LK-HP-ENV UNIT NO.  5   PIPE


         NAME NO.
         VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
         ---- ---- -----      -------------------       ----
  EVENT : QQ     4 HI        UNIT NO.  5   PIPE         OR

R-EVENT :          A         UNIT NO.  4   CNTRL-VAL
T-EVENT : BB    13 HI        UNIT NO.  4   CNTRL-VAL
B-EVENT :          FAIL-OPEN UNIT NO.  4   CNTRL-VAL
B-EVENT :          LK-HP-ENV UNIT NO.  4   CNTRL-VAL


         NAME NO.
         VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
         ---- ---- -----      -------------------       ----
  EVENT :          A         UNIT NO.  4   CNTRL-VAL    AND

T-EVENT : PP     3 HI        UNIT NO.  4   CNTRL-VAL
R-EVENT :          C         UNIT NO.  4   CNTRL-VAL


         NAME NO.
         VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
         ---- ---- -----      -------------------       ----
  EVENT : PP     3 HI        UNIT NO.  4   CNTRL-VAL    OR

T-EVENT : QQ     3 HI        UNIT NO.  4   CNTRL-VAL


         NAME NO.
```

Fig. 6.12 /continued

```
PAGE
---------
```

```
            VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
            ---- ---- -----        -------------------       ----
  EVENT :   UU    3   HI          UNIT NO.  4   CNTRL-VAL    OR

T-EVENT :   PP    2   HI          UNIT NO.  3   PIPE
B-EVENT :             FL-EX-ENV   UNIT NO.  3   PIPE


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
            ---- ---- -----        -------------------       ----
  EVENT :   PP    2   HI          UNIT NO.  3   PIPE         OR

T-EVENT :   QQ    2   HI          UNIT NO.  3   PIPE
B-EVENT :             LK-HP-ENV   UNIT NO.  3   PIPE


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
            ---- ---- -----        -------------------       ----
  EVENT :   QQ    2   HI          UNIT NO.  3   PIPE         OR

T-EVENT :   PP    1   HI          UNIT NO.  2   CLOSED-TK
B-EVENT :             FL-EX-ENV   UNIT NO.  2   CLOSED-TK


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
            ---- ---- -----        -------------------       ----
  EVENT :   PP    1   HI          UNIT NO.  2   CLOSED-TK    OR

T-EVENT :   L     1   HI          UNIT NO.  2   CLOSED-TK
B-EVENT :             LK-HP-ENV   UNIT NO.  2   CLOSED-TK


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
            ---- ---- -----        -------------------       ----
  EVENT :   L     1   HI          UNIT NO.  2   CLOSED-TK    OR

T-EVENT :   QQ    1   HI          UNIT NO.  2   CLOSED-TK
B-EVENT :             LK-HP-ENV   UNIT NO.  2   CLOSED-TK


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
            ---- ---- -----        -------------------       ----
  EVENT :   UU    1   HI          UNIT NO.  2   CLOSED-TK
                                  *DIAMOND EVENT*


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
            ---- ---- -----        -------------------       ----
  EVENT :             C           UNIT NO.  4   CNTRL-VAL    OR
```

Fig. 6.12 /continued

```
PAGE
--------
```

```
T-EVENT :   BB    12   NO-CHANGE UNIT NO.  4    CNTRL-VAL
B-EVENT :              VALV-STCK UNIT NO.  4    CNTRL-VAL
B-EVENT :              MANUAL    UNIT NO.  4    CNTRL-VAL


           NAME NO.
           VAR. VAR.  FAULT        DESCRIPTION OF UNIT      GATE
           ---- ----  -----        -------------------      ----
  EVENT :   BB   12   NO-CHANGE UNIT NO.  4    CNTRL-VAL    OR

T-EVENT :   SS    8   NO-CHANGE UNIT NO.  8    CNTROLLER
B-EVENT :             CONT-STCK UNIT NO.  8    CNTROLLER


           NAME NO.
           VAR. VAR.  FAULT        DESCRIPTION OF UNIT      GATE
           ---- ----  -----        -------------------      ----
  EVENT :   SS    8   NO-CHANGE UNIT NO.  8    CNTROLLER    OR

B-EVENT :             SENS-STCK UNIT NO.  7    SENSOR-Q


           NAME NO.
           VAR. VAR.  FAULT        DESCRIPTION OF UNIT      GATE
           ---- ----  -----        -------------------      ----
  EVENT :   BB   12   HI        UNIT NO.  4    CNTRL-VAL    OR

T-EVENT :   SS    8   LO        UNIT NO.  8    CNTROLLER
T-EVENT :   WW    9   HI        UNIT NO.  8    CNTROLLER
B-EVENT :             CONT-F-HI UNIT NO.  8    CNTROLLER


           NAME NO.
           VAR. VAR.  FAULT        DESCRIPTION OF UNIT      GATE
           ---- ----  -----        -------------------      ----
  EVENT :   SS    8   LO        UNIT NO.  8    CNTROLLER    OR

B-EVENT :             SEN-FA-LO UNIT NO.  7    SENSOR-Q


           NAME NO.
           VAR. VAR.  FAULT        DESCRIPTION OF UNIT      GATE
           ---- ----  -----        -------------------      ----
  EVENT :   WW    9   HI        UNIT NO.  8    CNTROLLER
                                 *DIAMOND EVENT*


           NAME NO.
           VAR. VAR.  FAULT        DESCRIPTION OF UNIT      GATE
           ---- ----  -----        -------------------      ----
  EVENT :   PP    7   LO        UNIT NO.  7    SENSOR-Q
                                 *DIAMOND EVENT*
```

Fig. 6.12

```
PAGE
--------
```

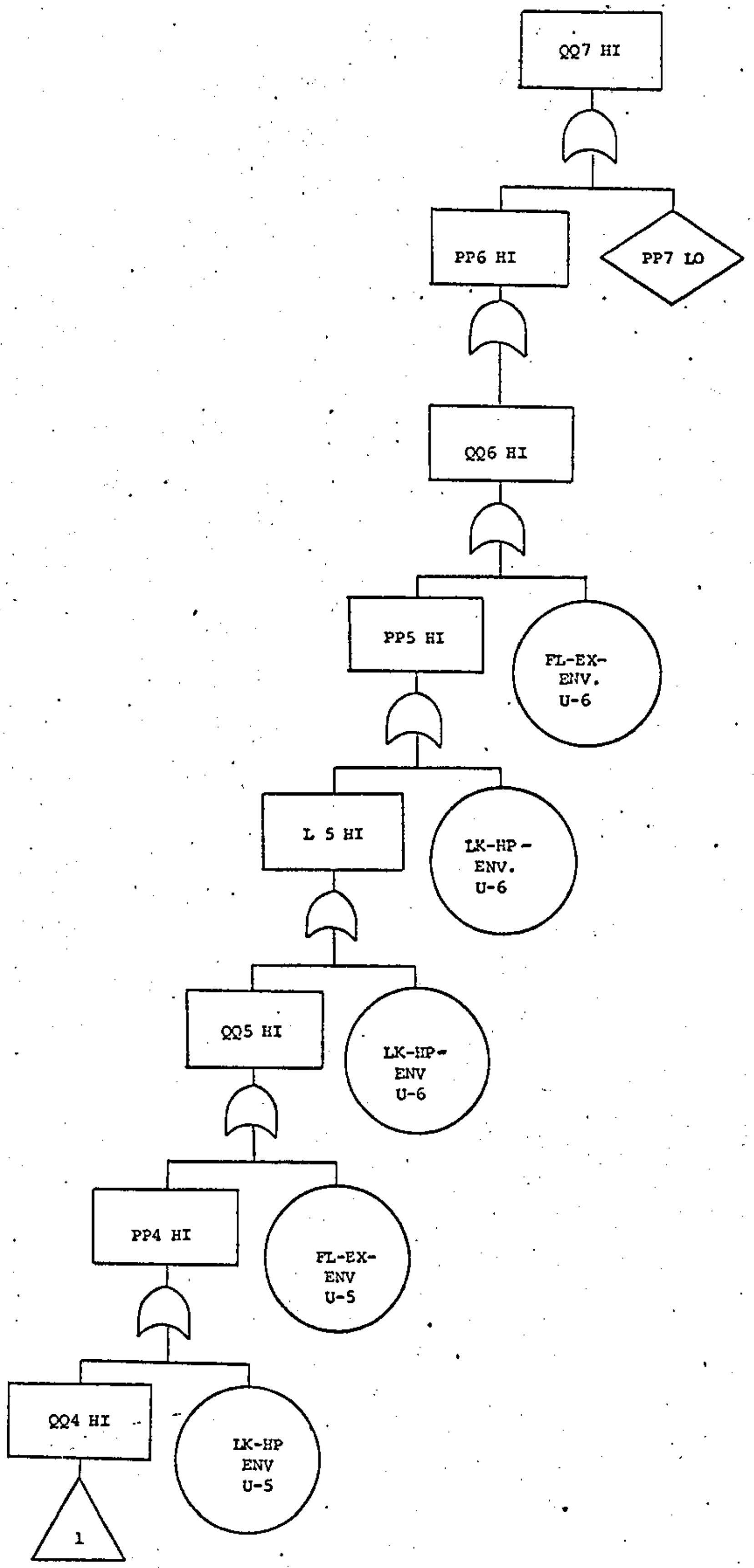
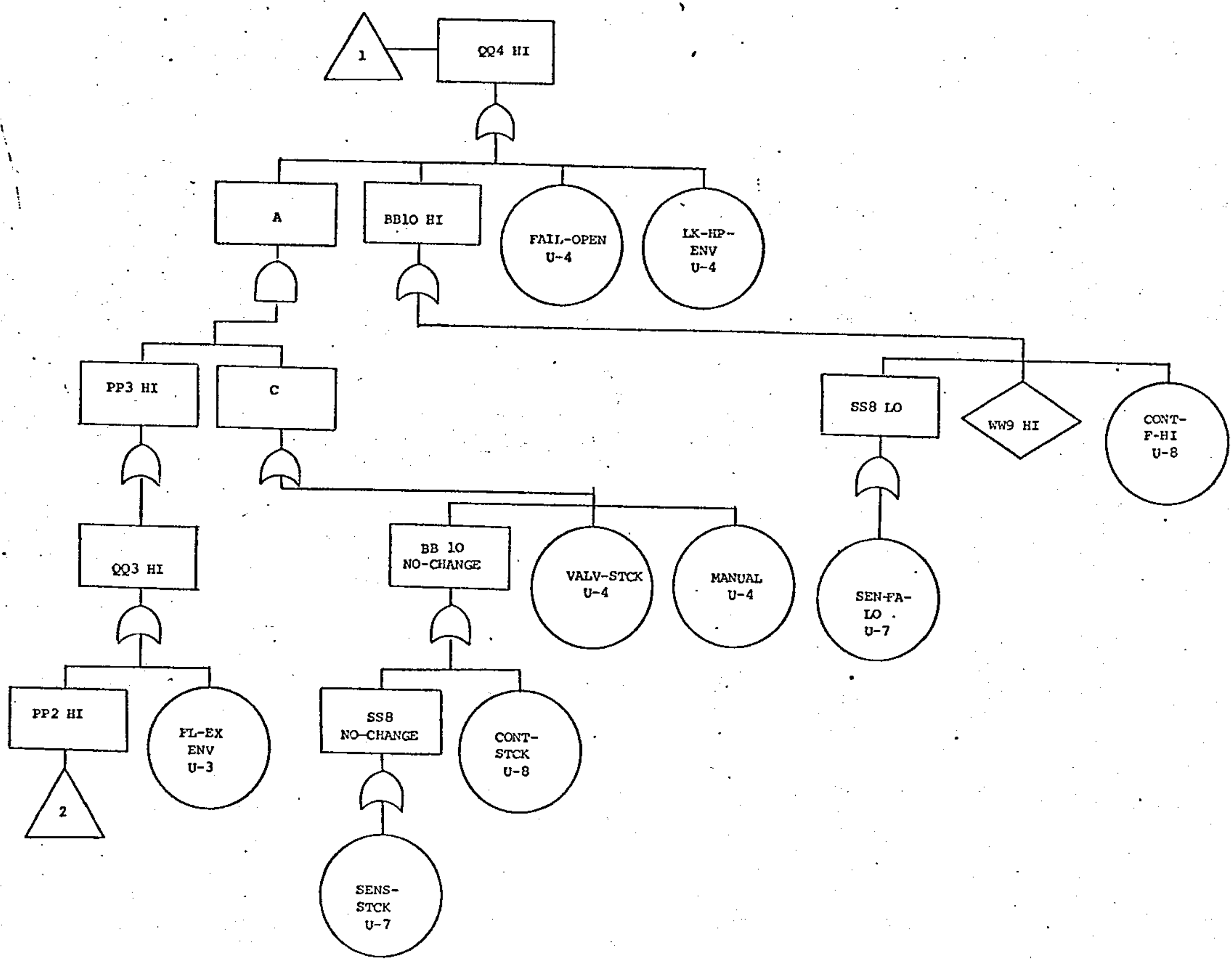
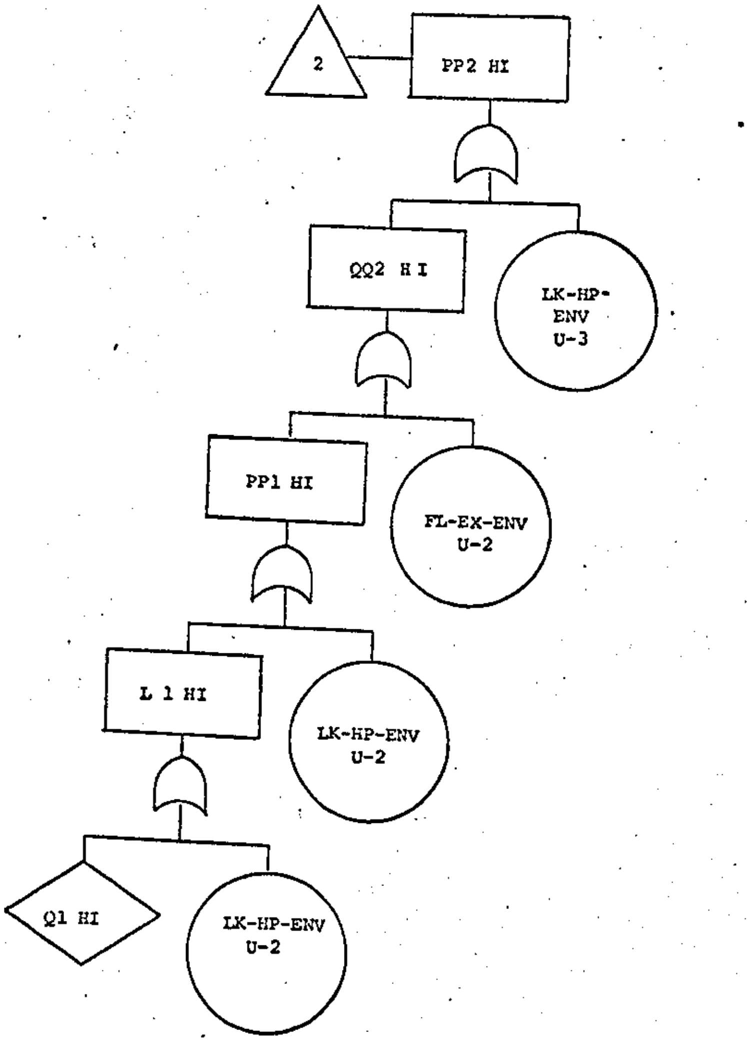


Fig. 6.13 Fault tree for top event QQ7 HI drawn using the data of Fig. 6.12

/continued



6.13 /continued



U = Unit number

Fig. 6.13

```
NOMENCLATURE :
--------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT .

*************************************************************
***********************    DESIGN    ************************
*************************************************************

                      *** FAULT TREE ***
                      ------------------


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT        GATE
             ---- ---- -----      -------------------        ----
TOP EVENT :   QQ    5  LO        UNIT NO.  5   PIPE          OR

  T-EVENT :   PP    4  LO        UNIT NO.  5   PIPE
  T-EVENT :   PP    5  HI        UNIT NO.  5   PIPE
  B-EVENT :            BLOCKAGE  UNIT NO.  5   PIPE
  B-EVENT :            LK-LP-ENV UNIT NO.  5   PIPE


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT        GATE
             ---- ---- -----      -------------------        ----
    EVENT :   PP    4  LO        UNIT NO.  5   PIPE          OR

  T-EVENT :   QQ    4  LO        UNIT NO.  5   PIPE
  B-EVENT :            LK-LP-ENV UNIT NO.  5   PIPE


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT        GATE
             ---- ---- -----      -------------------        ----
    EVENT :   QQ    4  LO        UNIT NO.  5   PIPE          OR

  R-EVENT :            D         UNIT NO.  4   CNTRL-VAL
  T-EVENT :   BB   10  LO        UNIT NO.  4   CNTRL-VAL
  B-EVENT :            BLOCKAGE  UNIT NO.  4   CNTRL-VAL
  B-EVENT :            LK-LP-ENV UNIT NO.  4   CNTRL-VAL
  B-EVENT :            FAI-CLOSE UNIT NO.  4   CNTRL-VAL


             NAME NO.
             VAR. VAR. FAULT      DESCRIPTION OF UNIT        GATE
             ---- ---- -----      -------------------        ----
    EVENT :            D         UNIT NO.  4   CNTRL-VAL     AND
```

Fig. 6.14 Fault tree for event QQ5 LO in a two tank and control valve system

```
PAGE
----
```

/continued

```
T-EVENT :  PP     3  LO          UNIT NO.  4    CNTRL-VAL
B-EVENT :            C           UNIT NO.  4    CNTRL-VAL


          NAME NO.
          VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
          ---- ---- -----        -------------------        ----
  EVENT :  PP     3  LO          UNIT NO.  4    CNTRL-VAL   OR

T-EVENT :  QQ     3  LO          UNIT NO.  4    CNTRL-VAL
B-EVENT :            LK-LP-ENV   UNIT NO.  4    CNTRL-VAL


          NAME NO.
          VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
          ---- ---- -----        -------------------        ----
  EVENT :  QQ     3  LO          UNIT NO.  4    CNTRL-VAL   OR

T-EVENT :  PP     2  LO          UNIT NO.  3    PIPE
B-EVENT :            BLOCKAGE    UNIT NO.  3    PIPE
B-EVENT :            LK-LP-ENV   UNIT NO.  3    PIPE


          NAME NO.
          VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
          ---- ---- -----        -------------------        ----
  EVENT :  PP     2  LO          UNIT NO.  3    PIPE        OR

T-EVENT :  QQ     2  LO          UNIT NO.  3    PIPE
B-EVENT :            LK-LP-ENV   UNIT NO.  3    PIPE


          NAME NO.
          VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
          ---- ---- -----        -------------------        ----
  EVENT :  QQ     2  LO          UNIT NO.  3    PIPE        OR

T-EVENT :  PP     1  LO          UNIT NO.  2    CLOSED-TK
B-EVENT :            LK-LP-ENV   UNIT NO.  2    CLOSED-TK
B-EVENT :            BLOCKAGE    UNIT NO.  2    CLOSED-TK


          NAME NO.
          VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
          ---- ---- -----        -------------------        ----
  EVENT :  PP     1  LO          UNIT NO.  2    CLOSED-TK   OR

T-EVENT :  L      1  LO          UNIT NO.  2    CLOSED-TK
B-EVENT :            LK-LP-ENV   UNIT NO.  2    CLOSED-TK


          NAME NO.
          VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
          ---- ---- -----        -------------------        ----
  EVENT :  L      1  LO          UNIT NO.  2    CLOSED-TK   OR

PAGE
---------
```

Fig. 6.14 /continued

```
T-EVENT :   QQ     1   LO          UNIT NO.   2    CLOSED-TK
B-EVENT :              LK-LP-ENV   UNIT NO.   2    CLOSED-TK
B-EVENT :              BLOCKAGE    UNIT NO.   2    CLOSED-TK


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  QQ    1  LO        UNIT NO.   2    CLOSED-TK
                                 *DIAMOND EVENT*


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :           C         UNIT NO.   4    CNTRL-VAL    OR

T-EVENT :   BB    12   NO-CHANGE UNIT NO.   4    CNTRL-VAL
B-EVENT :              VALV-STCK UNIT NO.   4    CNTRL-VAL
B-EVENT :              MANUAL    UNIT NO.   4    CNTRL-VAL


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  BB   12  NO-CHANGE UNIT NO.   4    CNTRL-VAL    OR

T-EVENT :   SS     8   NO-CHANGE UNIT NO.   8    CNTROLLER
B-EVENT :              CONT-STCK UNIT NO.   8    CNTROLLER


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  SS    8  NO-CHANGE UNIT NO.   8    CNTROLLER    OR

B-EVENT :              SENS-STCK UNIT NO.   7    SENSOR-Q


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  BB   12  LO        UNIT NO.   4    CNTRL-VAL    OR

T-EVENT :   SS     8   HI        UNIT NO.   8    CNTROLLER
T-EVENT :   WW     9   LO        UNIT NO.   8    CNTROLLER
B-EVENT :              CONT-F-LO UNIT NO.   8    CNTROLLER


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  SS    8  HI        UNIT NO.   8    CNTROLLER    OR
```

Fig. 6.14 /continued

```
PAGE
--------
```

```
T-EVENT :   QQ     6  HI          UNIT NO.  7    SENSOR-Q
B-EVENT :             SEN-FA-HI   UNIT NO.  7    SENSOR-Q



             NAME NO.
             VAR. VAR. FAULT         DESCRIPTION OF UNIT        GATE
             ---- ---- -----         -------------------        ----
   EVENT :   QQ    6   HI           UNIT NO.  7    SENSOR-Q     OR

T-EVENT :   PP     5  HI          UNIT NO.  6    CLOSED-TK
T-EVENT :   PP     6  LO          UNIT NO.  6    CLOSED-TK



             NAME NO.
             VAR. VAR. FAULT         DESCRIPTION OF UNIT        GATE
             ---- ---- -----         -------------------        ----
   EVENT :   PP    5   HI           UNIT NO.  6    CLOSED-TK    OR

T-EVENT :   L      5  HI          UNIT NO.  6    CLOSED-TK
B-EVENT :             LK-HP-ENV   UNIT NO.  6    CLOSED-TK



             NAME NO.
             VAR. VAR. FAULT         DESCRIPTION OF UNIT        GATE
             ---- ---- -----         -------------------        ----
   EVENT :   L     5   HI           UNIT NO.  6    CLOSED-TK    OR

B-EVENT :             LK-HP-ENV   UNIT NO.  6    CLOSED-TK



             NAME NO.
             VAR. VAR. FAULT         DESCRIPTION OF UNIT        GATE
             ---- ---- -----         -------------------        ----
   EVENT :   PP    6   LO           UNIT NO.  6    CLOSED-TK    OR

T-EVENT :   QQ     7  HI          UNIT NO.  7    SENSOR-Q



             NAME NO.
             VAR. VAR. FAULT         DESCRIPTION OF UNIT        GATE
             ---- ---- -----         -------------------        ----
   EVENT :   QQ    7   HI           UNIT NO.  7    SENSOR-Q     OR

T-EVENT :   PP     7  LO          UNIT NO.  7    SENSOR-Q



             NAME NO.
             VAR. VAR. FAULT         DESCRIPTION OF UNIT        GATE
             ---- ---- -----         -------------------        ----
   EVENT :   PP    7   LO           UNIT NO.  7    SENSOR-Q
                                     *DIAMOND EVENT*



             NAME NO.
```

Fig. 6.14 /continued

```
PAGE
---------
```

```
               VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------        ----
     EVENT :   WW     9  LO         UNIT NO.  8   CNTROLLER
                                     *DIAMOND EVENT*


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------        ----
     EVENT :   PP     5  HI         UNIT NO.  5   PIPE          OR

   T-EVENT :   L      5  HI         UNIT NO.  6   CLOSED-TK
   B-EVENT :             LK-HP-ENV  UNIT NO.  6   CLOSED-TK


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------        ----
     EVENT :   L      5  HI         UNIT NO.  6   CLOSED-TK     OR

   T-EVENT :   QQ     6  LO         UNIT NO.  6   CLOSED-TK
   B-EVENT :             BLOCKAGE   UNIT NO.  6   CLOSED-TK
   B-EVENT :             LK-HP-ENV  UNIT NO.  6   CLOSED-TK


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------        ----
     EVENT :   QQ     6  LO         UNIT NO.  6   CLOSED-TK     OR

   T-EVENT :   PP     6  HI         UNIT NO.  6   CLOSED-TK
   B-EVENT :             BLOCKAGE   UNIT NO.  6   CLOSED-TK


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------        ----
     EVENT :   PP     6  HI         UNIT NO.  6   CLOSED-TK     OR

   T-EVENT :   QQ     7  LO         UNIT NO.  7   SENSOR-Q


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------        ----
     EVENT :   QQ     7  LO         UNIT NO.  7   SENSOR-Q      OR

   T-EVENT :   PP     7  HI         UNIT NO.  7   SENSOR-Q


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------        ----
     EVENT :   PP     7  HI         UNIT NO.  7   SENSOR-Q
                                     *DIAMOND EVENT*
```

Fig. 6.14

```
PAGE
----------
```

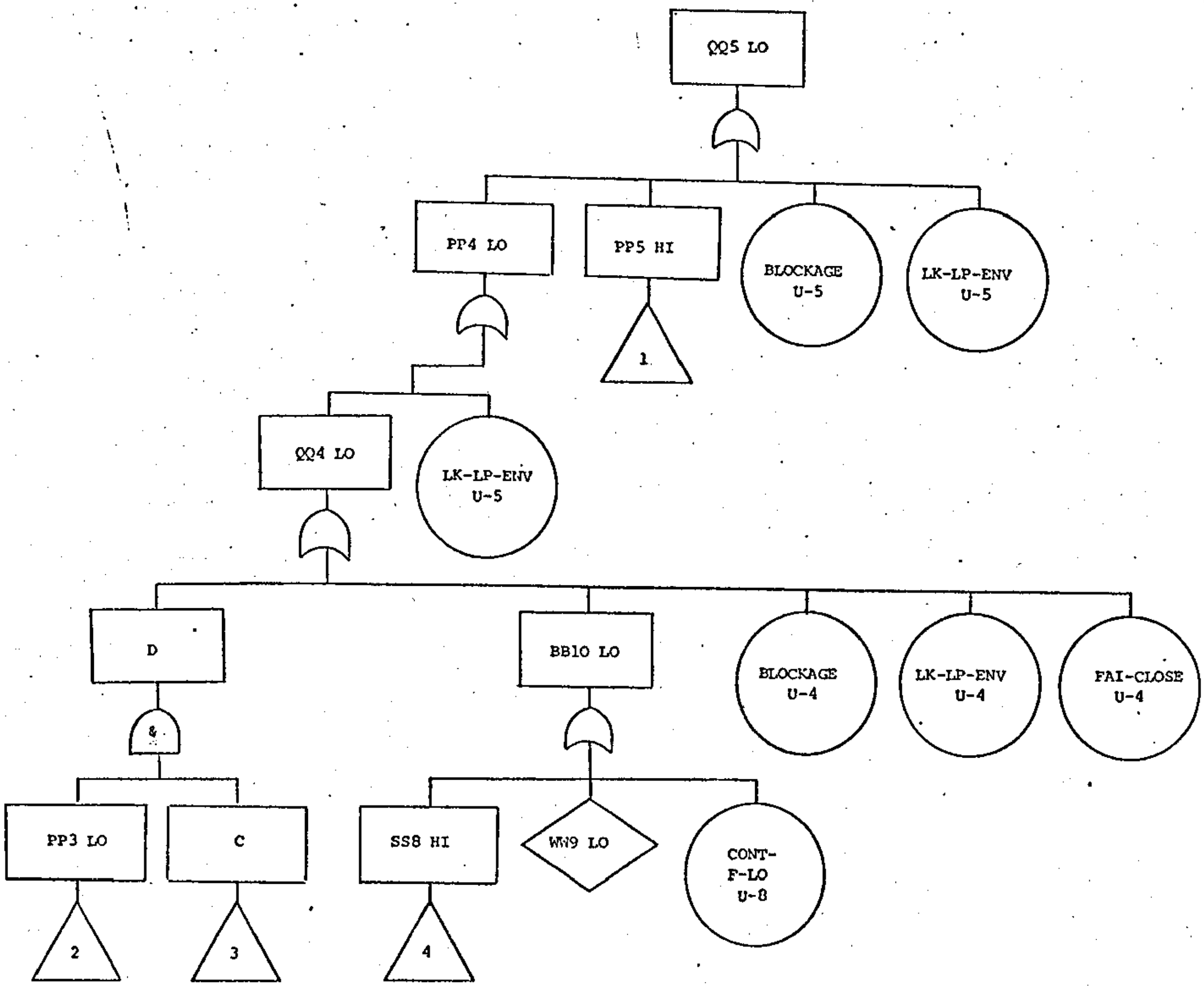


Fig. 6.15 Fault tree for top event QQ5 LO drawn using the data of Fig. 6.14

/continued

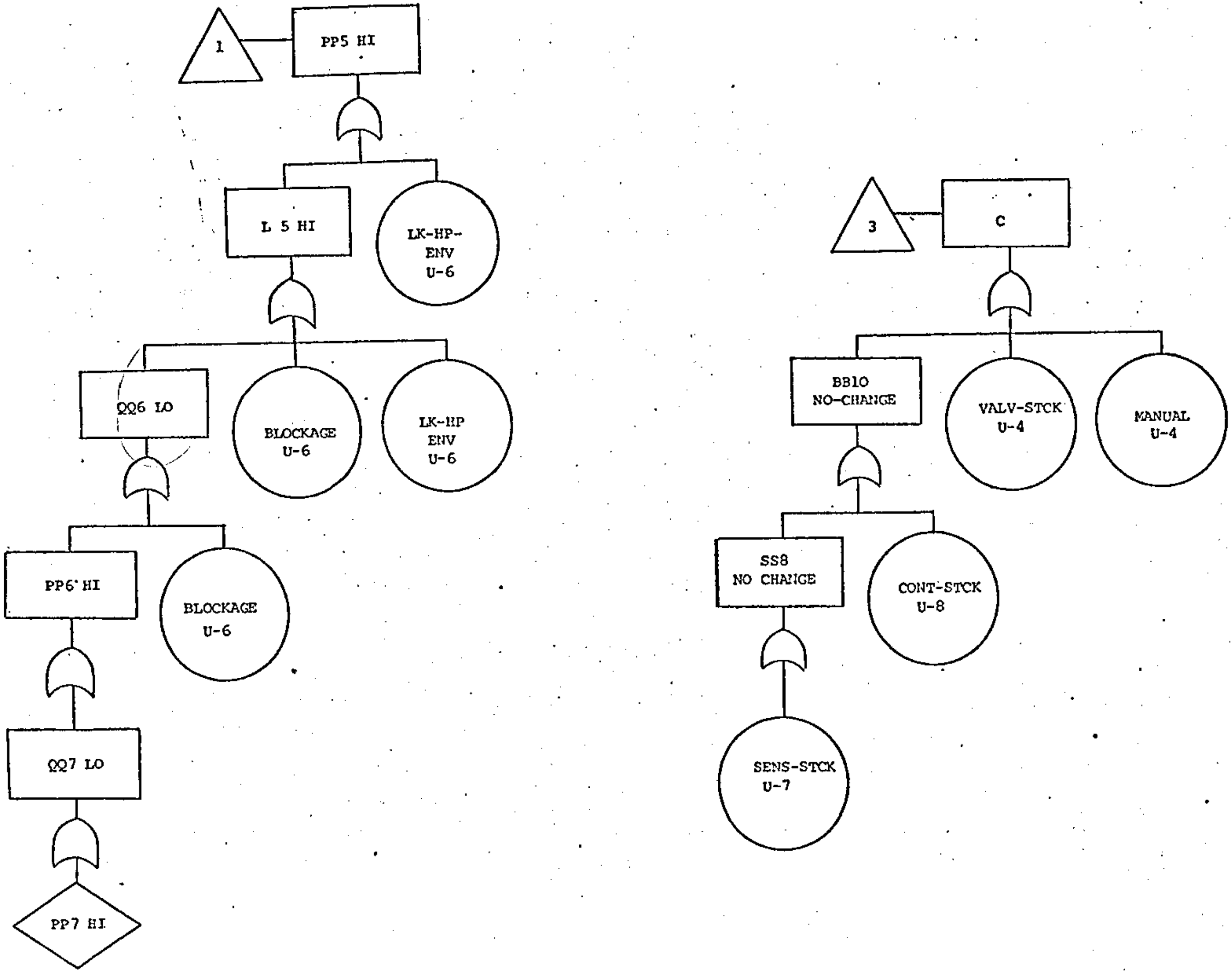


Fig. 6.15 /continued

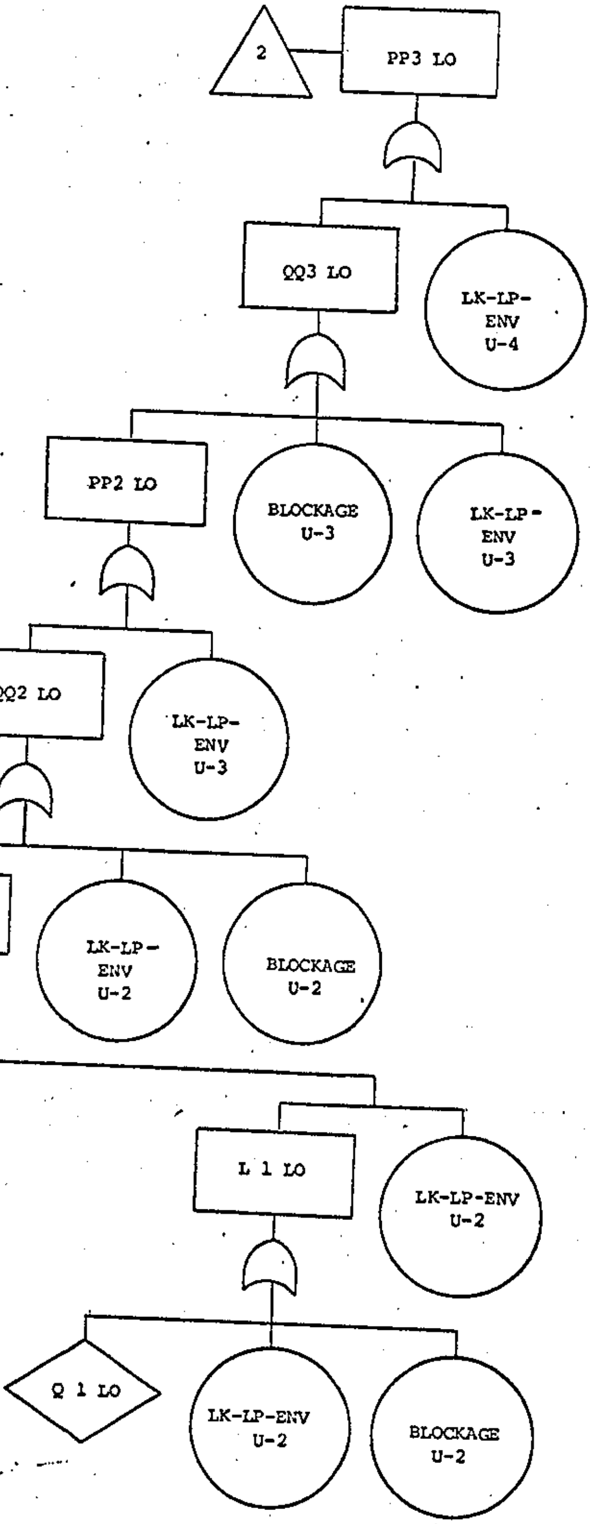


Fig. 6.15 /continued

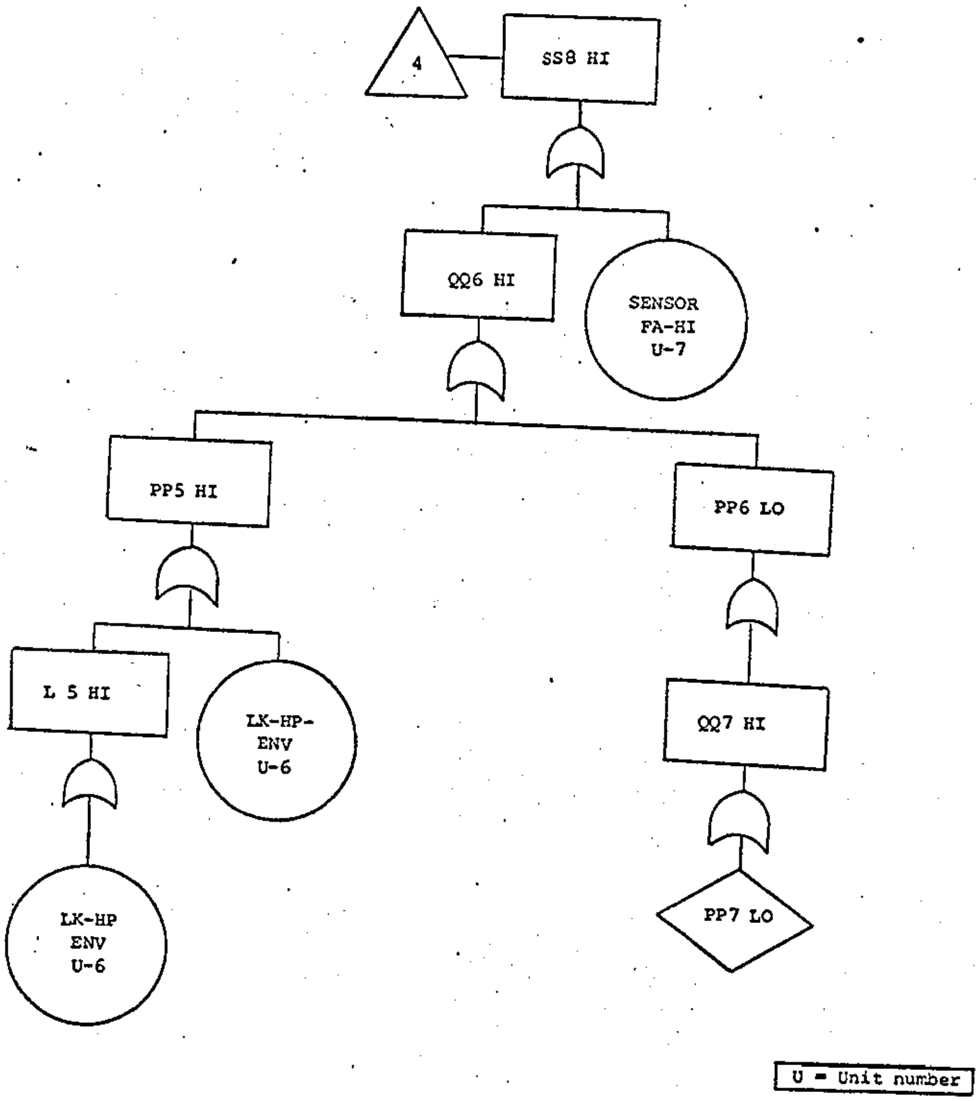

4
SS8 HI
QQ6 HI
SENSOR
FA-HI
U-7
PP5 HI
PP6 LO
L 5 HI
LK-HP-
ENV
U-6
QQ7 HI
LK-HP
ENV
U-6
PP7 LO
U = Unit number

Fig. 6.15

```
NOMENCLATURE :
--------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

*****************************************************************
************************      DESIGN     ************************
*****************************************************************

                        *** FAULT TREE ***
                        ------------------



            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
            ---- ---- -----      -------------------      ----
TOP EVENT :  PP    3  HI        UNIT NO.  4   CNTRL-VAL   OR

  T-EVENT :  QQ    3  HI        UNIT NO.  4   CNTRL-VAL
  T-EVENT :  QQ    4  LO        UNIT NO.  4   CNTRL-VAL
  B-EVENT :           BLOCKAGE  UNIT NO.  4   CNTRL-VAL
  B-EVENT :           SHUT      UNIT NO.  4   CNTRL-VAL


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
            ---- ---- -----      -------------------      ----
    EVENT :  QQ    3  HI        UNIT NO.  4   CNTRL-VAL   OR

  T-EVENT :  PP    2  HI        UNIT NO.  3   PIPE
  B-EVENT :           FL-EX-ENV UNIT NO.  3   PIPE


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
            ---- ---- -----      -------------------      ----
    EVENT :  PP    2  HI        UNIT NO.  3   PIPE        OR

  T-EVENT :  QQ    2  HI        UNIT NO.  3   PIPE
  B-EVENT :           LK-HP-ENV UNIT NO.  3   PIPE


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT      GATE
            ---- ---- -----      -------------------      ----
    EVENT :  QQ    2  HI        UNIT NO.  3   PIPE        OR

  T-EVENT :  PP    1  HI        UNIT NO.  2   CLOSED-TK
  B-EVENT :           FL-EX-ENV UNIT NO.  2   CLOSED-TK
```

Fig. 6.16 Fault tree for event PP3 HI in a two tank and control valve system

```
PAGE
----
```

/continued

```
                NAME NO.
                VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
                ---- ---- -----        -------------------        ----
      EVENT :   PP    1   HI          UNIT NO.  2   CLOSED-TK     OR

   T-EVENT :    L     1   HI          UNIT NO.  2   CLOSED-TK
   B-EVENT :              LK-HP-ENV   UNIT NO.  2   CLOSED-TK


                NAME NO.
                VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
                ---- ---- -----        -------------------        ----
      EVENT :   L     1   HI          UNIT NO.  2   CLOSED-TK     OR

   T-EVENT :    QQ    1   HI          UNIT NO.  2   CLOSED-TK
   B-EVENT :              LK-HP-ENV   UNIT NO.  2   CLOSED-TK


                NAME NO.
                VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
                ---- ---- -----        -------------------        ----
      EVENT :   QQ    1   HI          UNIT NO.  2   CLOSED-TK
                                       *DIAMOND EVENT*


                NAME NO.
                VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
                ---- ---- -----        -------------------        ----
      EVENT :   QQ    4   LO          UNIT NO.  4   CNTRL-VAL     OR

   T-EVENT :    PP    4   HI          UNIT NO.  4   CNTRL-VAL
   T-EVENT :    BB   12   LO          UNIT NO.  4   CNTRL-VAL
   B-EVENT :              BLOCKAGE    UNIT NO.  4   CNTRL-VAL
   B-EVENT :              FAI-CLOSE   UNIT NO.  4   CNTRL-VAL


                NAME NO.
                VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
                ---- ---- -----        -------------------        ----
      EVENT :   PP    4   HI          UNIT NO.  4   CNTRL-VAL     OR

   T-EVENT :    QQ    5   LO          UNIT NO.  5   PIPE
   B-EVENT :              BLOCKAGE    UNIT NO.  5   PIPE
   B-EVENT :              LK-HP-ENV   UNIT NO.  5   PIPE


                NAME NO.
                VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
                ---- ---- -----        -------------------        ----
      EVENT :   QQ    5   LO          UNIT NO.  5   PIPE          OR

   T-EVENT :    PP    5   HI          UNIT NO.  5   PIPE
   B-EVENT :              BLOCKAGE    UNIT NO.  5   PIPE
```

Fig. 6.16 /continued

```
PAGE
----------
```

```
              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
              ---- ---- -----        ------------------         ----
    EVENT :   PP    5   HI         UNIT NO.  5    PIPE          OR

  T-EVENT :   L     5   HI         UNIT NO.  6    CLOSED-TK
  B-EVENT :             LK-HP-ENV  UNIT NO.  6    CLOSED-TK


              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
              ---- ---- -----        ------------------         ----
    EVENT :   L     5   HI         UNIT NO.  6    CLOSED-TK     OR

  T-EVENT :   QQ    6   LO         UNIT NO.  6    CLOSED-TK
  B-EVENT :             BLOCKAGE   UNIT NO.  6    CLOSED-TK
  B-EVENT :             LK-HP-ENV  UNIT NO.  6    CLOSED-TK


              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
              ---- ---- -----        ------------------         ----
    EVENT :   QQ    6   LO         UNIT NO.  6    CLOSED-TK     OR

  T-EVENT :   PP    6   HI         UNIT NO.  6    CLOSED-TK
  B-EVENT :             BLOCKAGE   UNIT NO.  6    CLOSED-TK


              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
              ---- ---- -----        ------------------         ----
    EVENT :   PP    6   HI         UNIT NO.  6    CLOSED-TK     OR

  T-EVENT :   QQ    7   LO         UNIT NO.  7    SENSOR-Q


              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
              ---- ---- -----        ------------------         ----
    EVENT :   QQ    7   LO         UNIT NO.  7    SENSOR-Q      OR

  T-EVENT :   PP    7   HI         UNIT NO.  7    SENSOR-Q


              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
              ---- ---- -----        ------------------         ----
    EVENT :   PP    7   HI         UNIT NO.  7    SENSOR-Q
                                    *DIAMOND EVENT*


              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
```

Fig. 6.16 /continued

```
PAGE
--------
```

```
                ---- ---- -----       ------------------     ----
   EVENT :  "BB" "10" LO               UNIT NO.  4   CNTRL-VAL   OR

T-EVENT :    SS     8  HI              UNIT NO.  8   CNTROLLER
T-EVENT :    WW     9  LO              UNIT NO.  8   CNTROLLER
B-EVENT :              CONT-F-LO       UNIT NO.  8   CNTROLLER


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
               ---- ---- -----        ------------------        ----
   EVENT :    "SS"  "8"  HI           UNIT NO.  8   CNTROLLER   OR

T-EVENT :      QQ     6  HI           UNIT NO.  7   SENSOR-Q
B-EVENT :                SEN-FA-HI    UNIT NO.  7   SENSOR-Q


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
               ---- ---- -----        ------------------        ----
   EVENT :    "QQ"  "6"  HI           UNIT NO.  7   SENSOR-Q    OR

T-EVENT :      PP     5  HI           UNIT NO.  6   CLOSED-TK
T-EVENT :      PP     6  LO           UNIT NO.  6   CLOSED-TK
B-EVENT :                FL-EX-ENV    UNIT NO.  6   CLOSED-TK


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
               ---- ---- -----        ------------------        ----
   EVENT :    "PP"  "5"  HI           UNIT NO.  6   CLOSED-TK   OR

T-EVENT :      L      5  HI           UNIT NO.  6   CLOSED-TK
B-EVENT :                LK-HP-ENV    UNIT NO.  6   CLOSED-TK


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
               ---- ---- -----        ------------------        ----
   EVENT :    "L"   "5"  HI           UNIT NO.  6   CLOSED-TK   OR

T-EVENT :      QQ     5  HI           UNIT NO.  6   CLOSED-TK
B-EVENT :                LK-HP-ENV    UNIT NO.  6   CLOSED-TK


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT       GATE
               ---- ---- -----        ------------------        ----
   EVENT :    "QQ"  "5"  HI           UNIT NO.  6   CLOSED-TK   OR

T-EVENT :      PP     4  HI           UNIT NO.  5   PIPE
B-EVENT :                FL-EX-ENV    UNIT NO.  5   PIPE


               NAME NO.
```

Fig. 6.16 /continued

```
PAGE
--------
```

```
           VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
           ---- ---- -----      -----------------         ----
  EVENT :  PP      4 HI        UNIT NO.   5    PIPE        OR

B-EVENT :            LK-HP-ENV UNIT NO.   5    PIPE


           NAME NO.
           VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
           ---- ---- -----      -----------------         ----
  EVENT :  PP      6 LO        UNIT NO.   6    CLOSED-TK   OR

T-EVENT :  QQ      7 HI        UNIT NO.   7    SENSOR-Q


           NAME NO.
           VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
           ---- ---- -----      -----------------         ----
  EVENT :  QQ      7 HI        UNIT NO.   7    SENSOR-Q    OR

T-EVENT :  PP      7 LO        UNIT NO.   7    SENSOR-Q


           NAME NO.
           VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
           ---- ---- -----      -----------------         ----
  EVENT :  PP      7 LO        UNIT NO.   7    SENSOR-Q
                                *DIAMOND EVENT*


           NAME NO.
           VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
           ---- ---- -----      -----------------         ----
  EVENT :  WW      9 LO        UNIT NO.   8    CNTROLLER
                                *DIAMOND EVENT*
```

Fig. 6.16

```
PAGE
---------
```

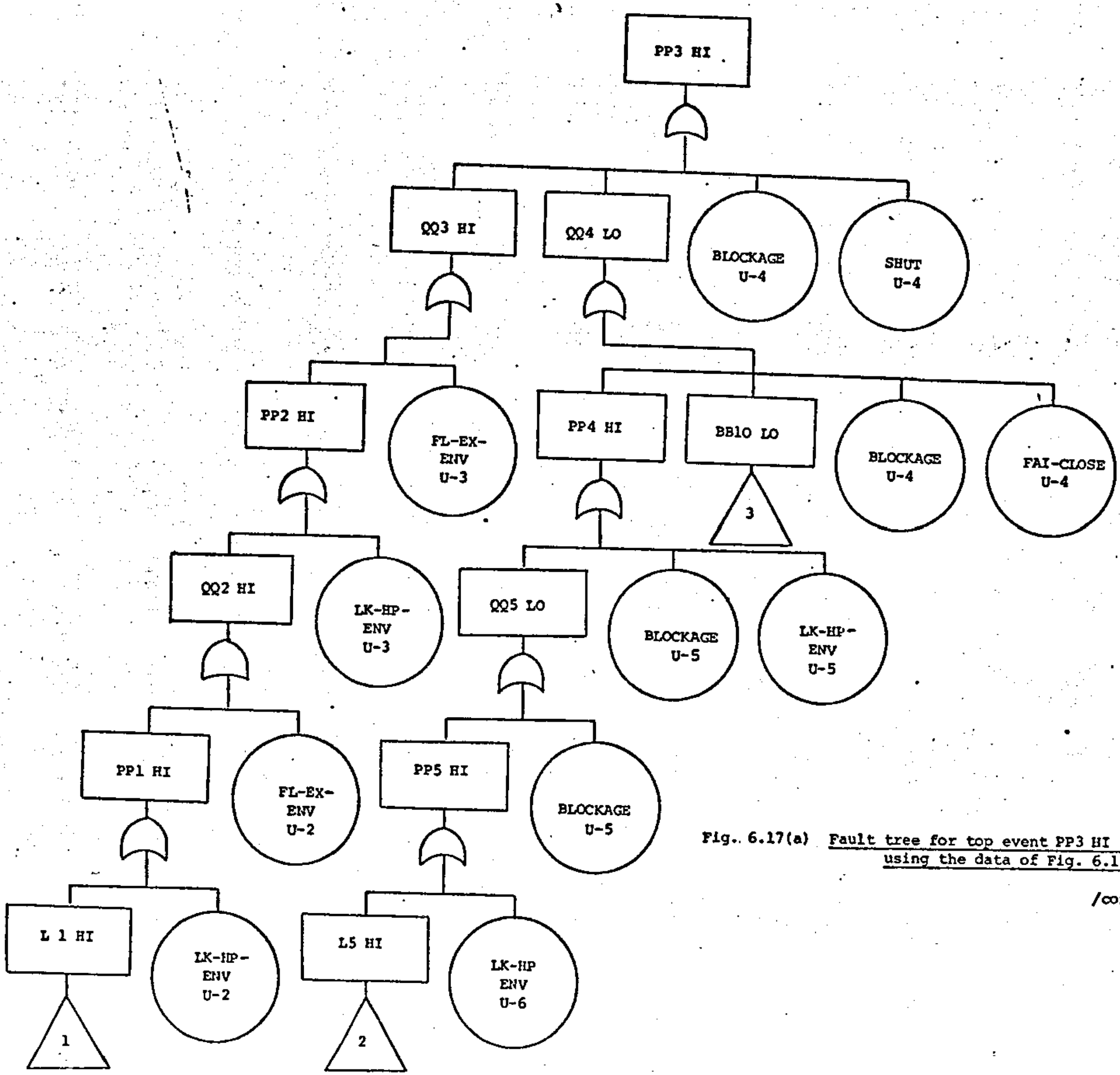


Fig. 6.17(a) Fault tree for top event PP3 HI drawn using the data of Fig. 6.16

/continued

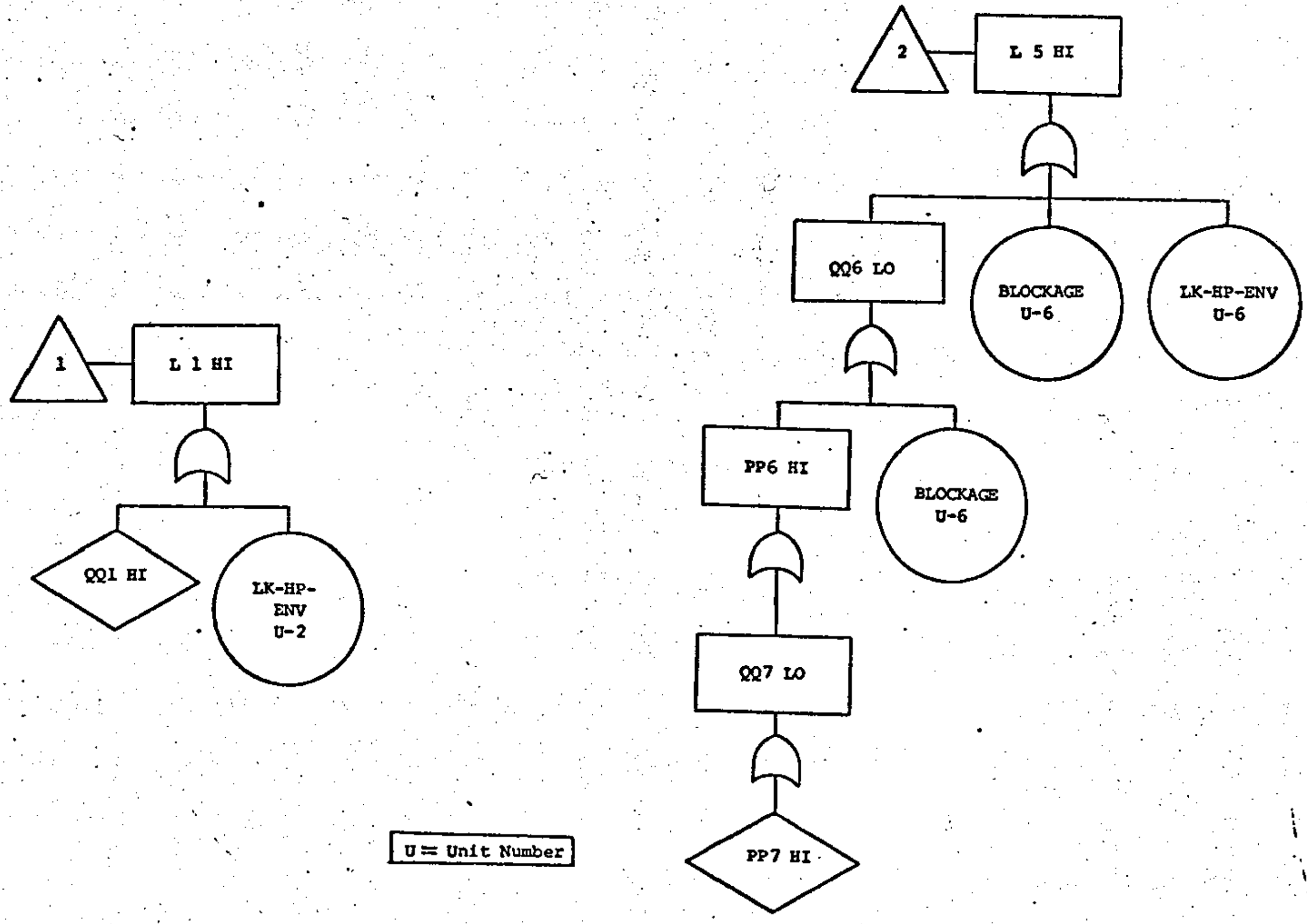


Fig. 6.17(a) /continued

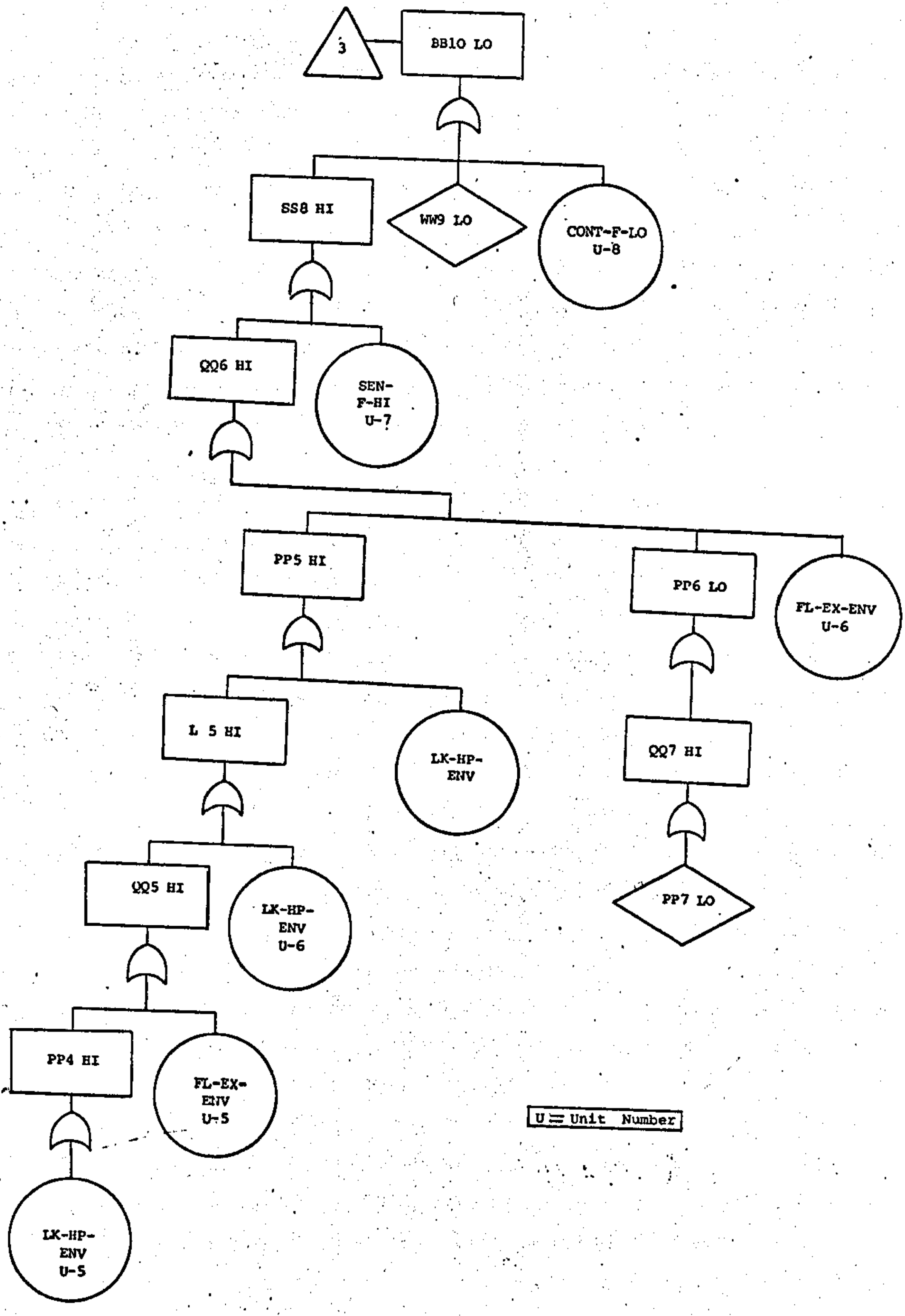

3
BB10 LO
SS8 HI
WW9 LO
CONT-F-LO
U-8
QQ6 HI
SEN-
F-HI
U-7
PP5 HI
PP6 LO
FL-EX-ENV
U-6
L 5 HI
LK-HP-
ENV
QQ7 HI
QQ5 HI
LK-HP-
ENV
U-6
PP7 LO
PP4 HI
FL-EX-
ENV
U-5
LK-HP-
ENV
U-5
U = Unit Number

Fig. 6.17(a)

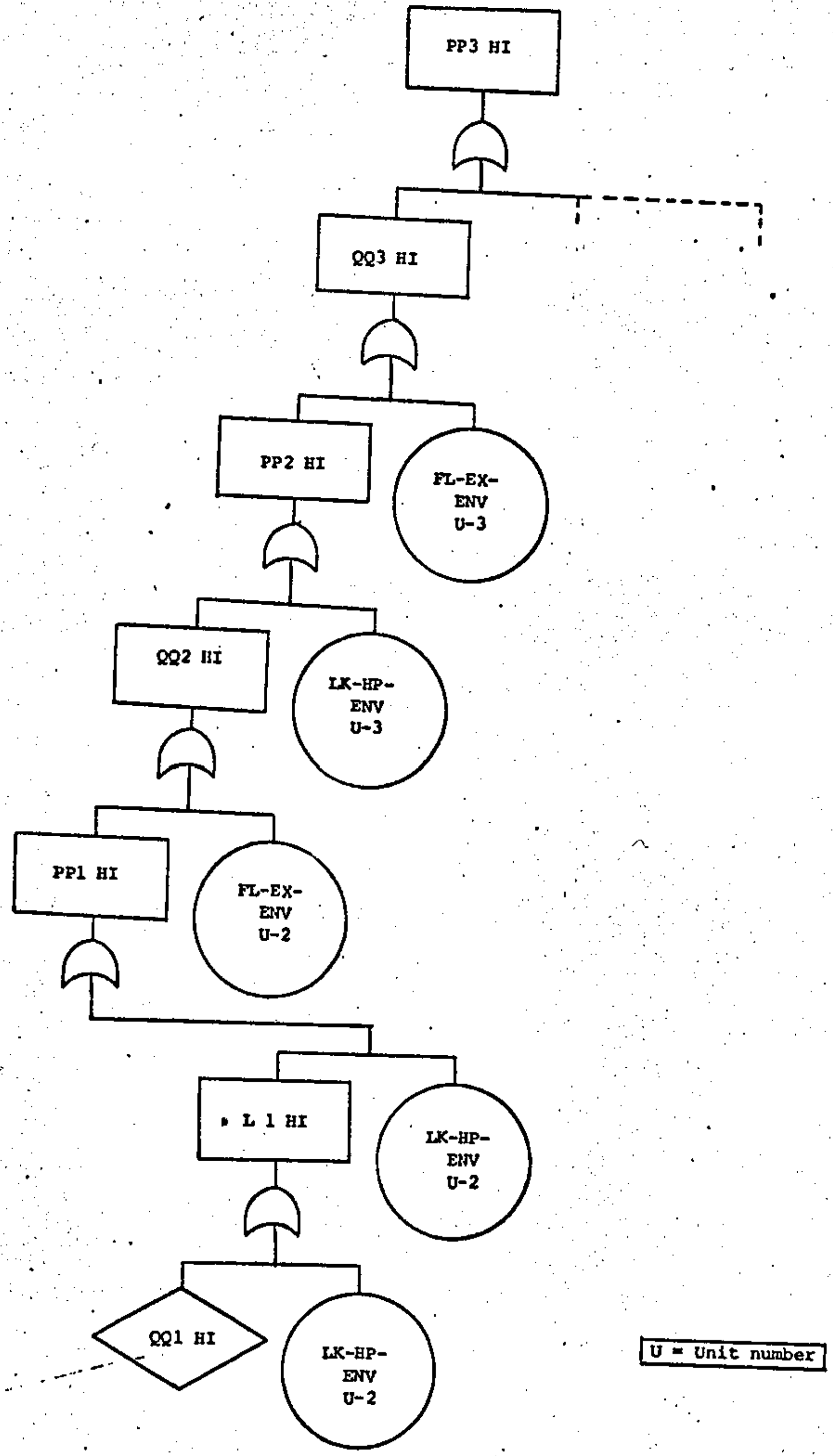


Fig. 6.17(b) Fault tree (with only one way flow of information) for top event PP3 HI in a two tank and control valve system

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ?  Y


                       MEASURED VARIABLES
                       ******************

     VARIABLES   HI LIMIT    LO LIMIT    STATUS    PRIORITY
     ---------   --------    --------    ------    --------

        PP  1       100          0         OK          1
        QQ  1       100          0         OK          1
        L   1       100          0         OK         10
        PP  2       100          0         OK          1
        QQ  2       100          0         OK          1
        PP  3       100          0         OK          9
        QQ  3       100          0         OK          1
        BB 10       100          0         OK          1
        PP  4       100          0         OK          1
        QQ  4       100          0         OK          1
        PP  5       100          0         OK          1
        QQ  5       100          0         OK          1
        L   5       100          0         OK         10
        PP  6       100          0         OK          1
        QQ  6       100          0         OK          1
        SS  8       100          0         OK          1
        PP  7       100          0         OK          1
        QQ  7       100          0         OK          1
        WW  9       100          0         OK          1

READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 6.18 The measured variables of the two tank and control valve system before any change has occurred to them

```
PAGE
--------
```

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ? Y


                                   MEASURED VARIABLES
                                   ******************

           VARIABLES      HI LIMIT      LO LIMIT      STATUS      PRIORITY
           ---------      --------      --------      ------      --------
           .........      ........      ........      ......      ........

               PP  1         100             0          OK              1
               QQ  1         100             0          OK              1
               L   1         100             0          OK             10
               PP  2         100             0          OK              1
               QQ  2         100             0          OK              1
               PP  3         100             0          OK              9
               QQ  3         100             0          OK              1
               BB 10         100             0          OK              1
               PP  4         100             0          OK              1
               QQ  4         100             0          OK              1
               PP  5         100             0          HI              1
               QQ  5         100             0          OK.             1
               L   5         100             0          HI             10
               PP  6         100             0          HI              1
               QQ  6         100             0          HI              1
               SS  8         100             0          OK              1
               PP  7         100             0          OK              1
               QQ  7         100             0          HI            250
               WW  9         100             0          OK              1

READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 6.19 The measured variables of the two tank and valve system after some values and priorities have been changed

```
PAGE
--------
........
```

```
NOMENCLATURE :
-------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

*******************************************************************
*************************   REAL   TIME   *************************
*******************************************************************

                        *** FAULT TREE ***
                        ------------------


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
TOP EVENT :  QQ    7   HI        UNIT NO.   7    SENSOR-Q     OR

  T-EVENT :  PP    6   HI        UNIT NO.   7    SENSOR-Q


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
    EVENT :  PP    6   HI        UNIT NO.   7    SENSOR-Q     OR

  T-EVENT :  QQ    6   HI        UNIT NO.   7    SENSOR-Q


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
    EVENT :  QQ    6   HI        UNIT NO.   7    SENSOR-Q     OR

  T-EVENT :  PP    5   HI        UNIT NO.   6    CLOSED-TK
  B-EVENT :            FL-EX-ENV UNIT NO.   6    CLOSED-TK


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
             ---- ---- -----       -------------------        ----
    EVENT :  PP    5   HI        UNIT NO.   6    CLOSED-TK    OR

  T-EVENT :  L     5   HI        UNIT NO.   6    CLOSED-TK
  B-EVENT :            LK-HP-ENV UNIT NO.   6    CLOSED-TK


             NAME NO.
             VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
```

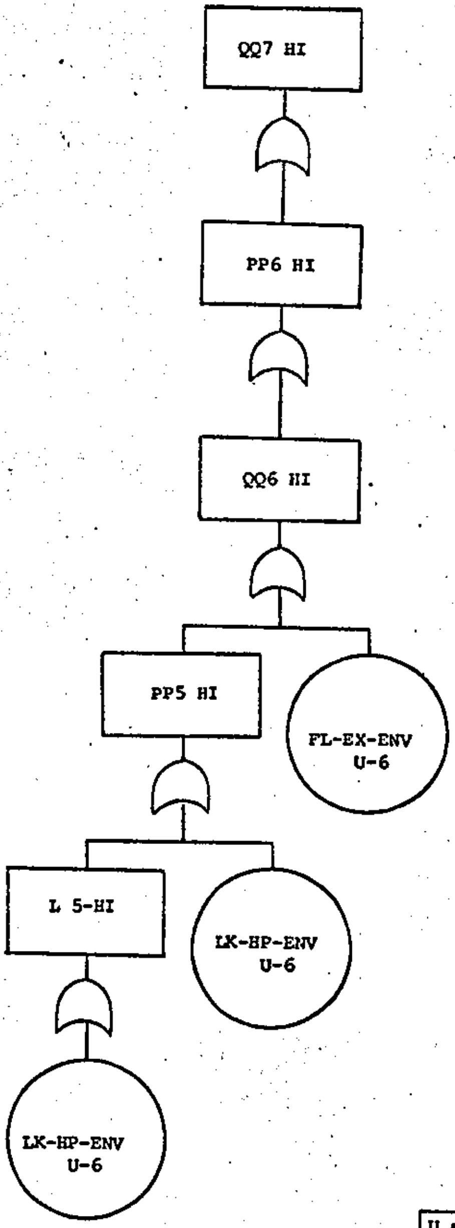
Fig. 6.20 Fault tree for QQ7 HI of the two tank and valve system when used for real time purposes

```
PAGE
--------
```

/continued

```
          ---- ---- -----     -------------------      ----
  EVENT : L       5 HI        UNIT NO.  6   CLOSED-TK  OR
B-EVENT :           LK-HP-ENV UNIT NO.  6   CLOSED-TK
```

Fig. 6.20



U = Unit Number

Fig. 6.21 Fault tree drawn by using the data provided by the listing of Fig. 6.20

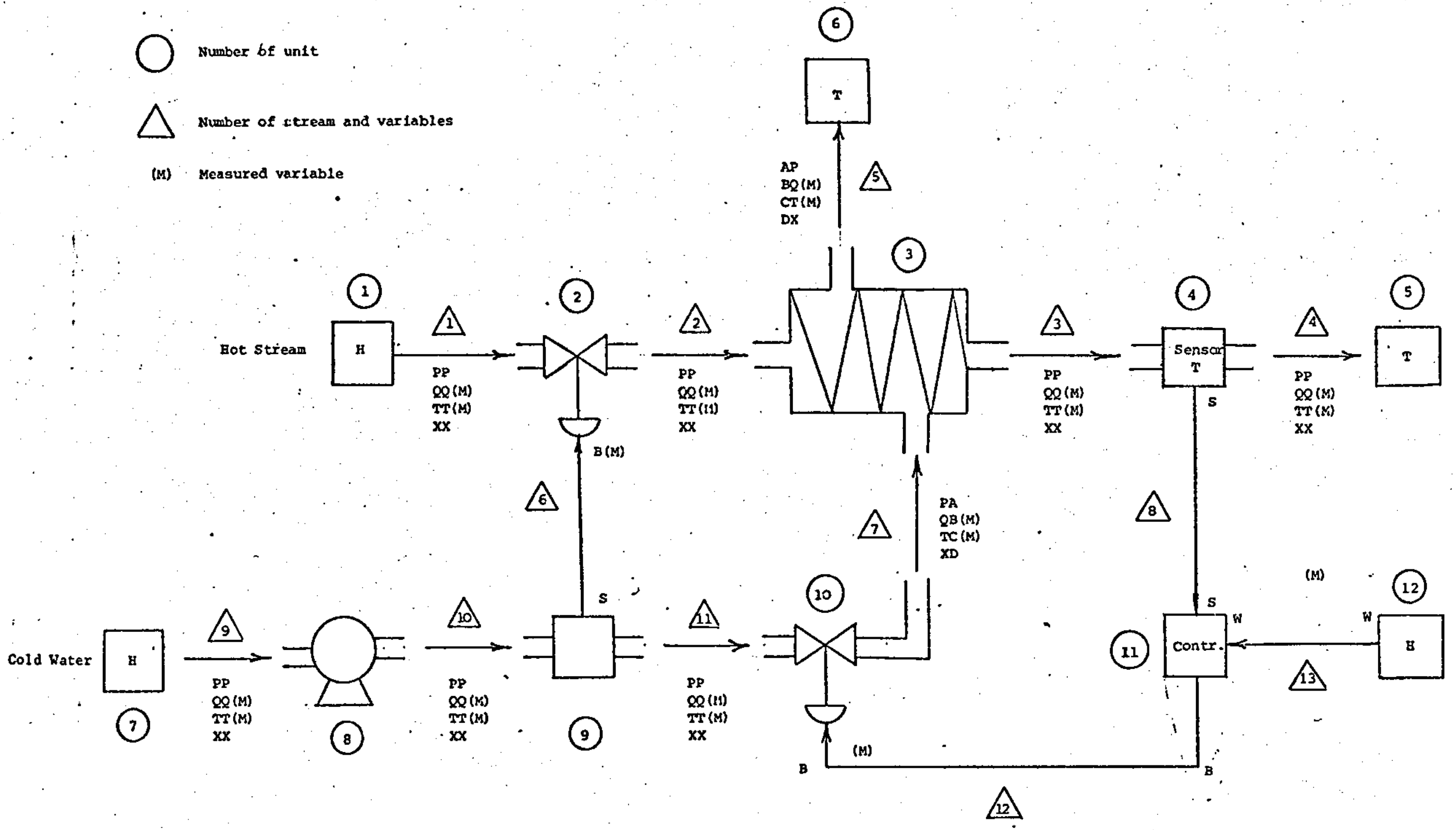


Fig. 6.22 Heat Exchanger and control loop system

```
EMT DK2:A.TNT;1=DK2:CENFUM.DAT
EMT DK0:C.TNT;1=DK2:CONTVA.DAT
EMT DK2:D.TNT;1=DK0:CONTLL.DAT
EMT DK2:E.TNT;1=DK2:DUMMYH.DAT
EMT DK2:F.TNT;1=DK2:DUMMYT.DAT
EMT DK2:G.TNT;1=DK0:HEATEX.DAT
EMT DK2:I.TNT;1=DK2:SENSQ.DAT
EMT DK0:J.TNT;1=DK2:SENST.DAT
REU DK2:A1.TNT;1=DK2:TOT4.DAT
DES DK2:A2.TNT;1=DK2:TOT4C.DAT
PIP *.TNT;*/DE
```

Fig. 6.23 TOT4 file used to set up the minitrees and define the topology of the Heat Exchanger and control loop system

```
PAGE                    TOT4   .CMD
--------
...............
```

```
TOPOLOGY OF THE SYSTEM
**********************


**************************************************************
UNIT NO. : 1
-----------
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : -1
TO : 2
VARIABLES : PP1,QQ1 (M) ,TT1 (M) ,XX1
**************************************************************


**************************************************************
UNIT NO. : 2
-----------
TYPE OF UNIT : CNTRL-VAL
** INPUT STREAMS **
FROM : 1
TO : 2
VARIABLES : PP1,QQ1 (M) ,TT1 (M) ,XX1
FROM : 7
TO : 2
VARIABLES : BB11 (M)
** OUTPUT STREAMS **
FROM : 2
TO : 3
VARIABLES : PP2,QQ2 (M) ,TT2 (M) ,XX2
**************************************************************


**************************************************************
UNIT NO. : 3
-----------
TYPE OF UNIT : HEAT-EX
** INPUT STREAMS **
FROM : 2
TO : 3
VARIABLES : PP2,QQ2 (M) ,TT2 (M) ,XX2
FROM : 12
TO : 3
VARIABLES : PA6,QB6 (M) ,TC6 (M) ,XD6
** OUTPUT STREAMS **
FROM : 3
TO : 4
VARIABLES : PP3,QQ3.(M) ,TT3 (M) ,XX3
FROM : 3
TO : 5
VARIABLES : AP5,BQ5 (M) ,CT5 (M) ,DX5 .
**************************************************************


**************************************************************
UNIT NO. : 4
-----------
TYPE OF UNIT : SENSOR-T
```

Fig. 6.24 Topology of the Heat-Exchanger and control loop system

```
PAGE
---------
```

/continued

```
** INPUT STREAMS **
FROM : 3
TO : 4
VARIABLES : PP3,QQ3 (M) ,TT3 (M) ,XX3
** OUTPUT STREAMS **
FROM : 4
TO : 6
VARIABLES : PP4,QQ4 (M) ,TT4 (M) ,XX4
FROM : 4
TO : 13
VARIABLES : SS14 (M)
**********************************************************


**********************************************************
UNIT NO. : 5
------------
TYPE OF UNIT : DUMMY-T
** INPUT STREAMS **
FROM : 3
TO : 5
VARIABLES : AP5,BQ5 (M) ,CT5 (M) ,DX5
** OUTPUT STREAMS **
NONE
**********************************************************


**********************************************************
UNIT NO. : 6
------------
TYPE OF UNIT : DUMMY-T
** INPUT STREAMS **
FROM : 4
TO : 6
VARIABLES : PP4,QQ4 (M) ,TT4 (M) ,XX4
** OUTPUT STREAMS **
NONE
**********************************************************


**********************************************************
UNIT NO. : 7
------------
TYPE OF UNIT : CNTROLLER
** INPUT STREAMS **
FROM : 8
TO : 7
VARIABLES : WW12 (M)
FROM : 11
TO : 7
VARIABLES : SS12 (M)
** OUTPUT STREAMS **
FROM : 7
TO : 2
VARIABLES : BB11 (M)
**********************************************************


**********************************************************
UNIT NO. : 8
------------
....................
```

Fig. 6.24 /continued

```
PAGE
---------
.................
```

```
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 8
TO : 7
VARIABLES : WW10 (M)
************************************************************


************************************************************
UNIT NO. : 9
-----------
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 9
TO : 10
VARIABLES : PP9,QQ9 (M) ,TT9 (M) ,XX9
************************************************************


************************************************************
UNIT NO. : 10
-----------
TYPE OF UNIT : CENT-PUMP
** INPUT STREAMS **
FROM : 9
TO : 10
VARIABLES : PP9,QQ9 (M) ,TT9 (M) ,XX9
** OUTPUT STREAMS **
FROM : 10
TO : 11
VARIABLES : PP8,QQ8 (M) ,TT8 (M) ,XX8
************************************************************


************************************************************
UNIT NO. : 11
-----------
TYPE OF UNIT : SENSOR-Q
** INPUT STREAMS **
FROM : 10
TO : 11
VARIABLES : PP8,QQ8 (M) ,TT8 (M) ,XX8
** OUTPUT STREAMS **
FROM : 11
TO : 7
VARIABLES : SS12 (M)
FROM : 11
TO : 12
VARIABLES : PP7,QQ7 (M) ,TT7 (M) ,XX7
************************************************************


************************************************************
UNIT NO. : 12
-----------
TYPE OF UNIT : CNTRL-VAL
** INPUT STREAMS **
```

Fig. 6.24 /continued

```
PAGE
---------
................
```

```
FROM : 11
TO : 12
VARIABLES : PP7,QQ7 (M) ,TT7 (M) ,XX7
FROM : 13
TO : 12
VARIABLES : BB13 (M)
** OUTPUT STREAMS **
FROM : 12
TO : 3
VARIABLES : PA6,QB6 (M) ,TC6 (M) ,XD6
*****************************************************************

*****************************************************************
UNIT NO. : 13
-----------
TYPE OF UNIT : CNTROLLER
** INPUT STREAMS **
FROM : 14
TO : 13
VARIABLES : WW15 (M)
FROM : 4
TO : 13
VARIABLES : SS14 (M)
** OUTPUT STREAMS **
FROM : 13
TO : 12
VARIABLES : BB13 (M)
*****************************************************************

*****************************************************************
UNIT NO. : 14
-----------
TYPE OF UNIT : DUMMY-H
** INPUT STREAMS **
NONE
** OUTPUT STREAMS **
FROM : 14
TO : 13
VARIABLES : WW15 (M)
*****************************************************************
```

Fig. 6.24

```
PAGE
--------
```

```
NOMENCLATURE :
--------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

*****************************************************************
*************************     DESIGN     ************************
*****************************************************************

                      *** FAULT TREE ***
                      ------------------


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT       GATE
           ---- ---- -----       -------------------       ----
TOP EVENT : TT    4  HI         UNIT NO.  4   SENSOR-T     OR

  T-EVENT : TT    3  HI         UNIT NO.  4   SENSOR-T
  B-EVENT :          EXT-FIRE   UNIT NO.  4   SENSOR-T


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT       GATE
           ---- ---- -----       -------------------       ----
    EVENT : TT    3  HI         UNIT NO.  4   SENSOR-T     OR

  T-EVENT : TT    2  HI         UNIT NO.  3   HEAT-EX
  T-EVENT : QQ    3  HI         UNIT NO.  3   HEAT-EX
  R-EVENT :          Z-LO       UNIT NO.  3   HEAT-EX
  B-EVENT :          EXT-FIRE   UNIT NO.  3   HEAT-EX


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT       GATE
           ---- ---- -----       -------------------       ----
    EVENT : TT    2  HI         UNIT NO.  3   HEAT-EX      OR

  T-EVENT : TT    1  HI         UNIT NO.  2   CNTRLV-SC
  B-EVENT :          EXT-FIRE   UNIT NO.  2   CNTRLV-SC


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT       GATE
           ---- ---- -----       -------------------       ----
    EVENT : TT    1  HI         UNIT NO.  2   CNTRLV-SC
                                 *DIAMOND EVENT*
```

Fig. 6.25 Fault tree for top event TT4 HI in a Heat Exchanger and control loop system

/continued

```
PAGE
---------
```

```
                NAME NO.
                VAR. VAR. FAULT         DESCRIPTION OF UNIT          GATE
                ---- ---- -----        -------------------           ----
   EVENT :       QQ    3  HI           UNIT NO.   3    HEAT-EX       OR

 T-EVENT :       PP    2  HI           UNIT NO.   3    HEAT-EX
 T-EVENT :       PP    3  LO           UNIT NO.   3    HEAT-EX
 B-EVENT :                FL-EX-ENV    UNIT NO.   3    HEAT-EX


                NAME NO.
                VAR. VAR. FAULT         DESCRIPTION OF UNIT          GATE
                ---- ---- -----        -------------------           ----
   EVENT :       PP    2  HI           UNIT NO.   3    HEAT-EX       OR

 T-EVENT :       QQ    2  HI           UNIT NO.   3    HEAT-EX
 B-EVENT :                LK-HP-ENV    UNIT NO.   3    HEAT-EX


                NAME NO.
                VAR. VAR. FAULT         DESCRIPTION OF UNIT          GATE
                ---- ---- -----        -------------------           ----
   EVENT :       QQ    2  HI           UNIT NO.   3    HEAT-EX       OR

 T-EVENT :       PP    1  HI           UNIT NO.   2    CNTRLV-SC
 B-EVENT :                LK-HP-ENV    UNIT NO.   2    CNTRLV-SC


                NAME NO.
                VAR. VAR. FAULT         DESCRIPTION OF UNIT          GATE
                ---- ---- -----        -------------------           ----
   EVENT :       PP    1  HI           UNIT NO.   2    CNTRLV-SC     OR

 T-EVENT :       QQ    1  HI           UNIT NO.   2    CNTRLV-SC


                NAME NO.
                VAR. VAR. FAULT         DESCRIPTION OF UNIT          GATE
                ---- ---- -----        -------------------           ----
   EVENT :       QQ    1  HI           UNIT NO.   2    CNTRLV-SC
                                        *DIAMOND EVENT*


                NAME NO.
                VAR. VAR. FAULT         DESCRIPTION OF UNIT          GATE
                ---- ---- -----        -------------------           ----
   EVENT :       PP    3  LO           UNIT NO.   3    HEAT-EX       OR

 T-EVENT :       QQ    4  HI           UNIT NO.   4    SENSOR-T


                NAME NO.
                VAR. VAR. FAULT         DESCRIPTION OF UNIT          GATE
                ---- ---- -----        -------------------           ----
   EVENT :       QQ    4  HI           UNIT NO.   4    SENSOR-T      OR
```

Fig. 6.25 /continued

```
PAGE
--------
```

```
T-EVENT :  PP     4  LO          UNIT NO.  4   SENSOR-T


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT     GATE
            ---- ---- -----        -------------------     ----
  EVENT :   PP    4   LO          UNIT NO.  4   SENSOR-T
                                   *DIAMOND EVENT*


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT     GATE
            ---- ---- -----        -------------------     ----
  EVENT :             Z-LO        UNIT NO.  3   HEAT-EX    OR

T-EVENT :   CT    5   HI          UNIT NO.  3   HEAT-EX


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT     GATE
            ---- ---- -----        -------------------     ----
  EVENT :   CT    5   HI          UNIT NO.  3   HEAT-EX    OR

T-EVENT :   TC    7   HI          UNIT NO.  3   HEAT-EX
R-EVENT :             B           UNIT NO.  3   HEAT-EX
B-EVENT :             EXT-FIRE    UNIT NO.  3   HEAT-EX

                                   :

            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT     GATE
            ---- ---- -----        -------------------     ----
  EVENT :   TC    7   HI          UNIT NO.  3   HEAT-EX    OR

T-EVENT :   TT   11   HI          UNIT NO. 10   CNTRL-VAL
B-EVENT :             EXT-FIRE    UNIT NO. 10   CNTRL-VAL


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT     GATE
            ---- ---- -----        -------------------     ----
  EVENT :   TT   11   HI          UNIT NO. 10   CNTRL-VAL  OR

T-EVENT :   TT   10   HI          UNIT NO.  9   SENSOR-Q
B-EVENT :             EXT-FIRE    UNIT NO.  9   SENSOR-Q


            NAME NO.
            VAR. VAR. FAULT        DESCRIPTION OF UNIT     GATE
            ---- ---- -----        -------------------     ----
  EVENT :   TT   10   HI          UNIT NO.  9   SENSOR-Q   OR

T-EVENT :   TT    9   HI          UNIT NO.  8   CENT-PUMP
B-EVENT :             EXT-FIRE    UNIT NO.  8   CENT-PUMP
```

Fig. 6.25 /continued

```
PAGE
---------
```

```
             NAME NO.
             VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
             ---- ---- -----        -------------------      ----
   EVENT :    TT    9  HI          UNIT NO.   8   CENT-PUMP
                                    *DIAMOND EVENT*


             NAME NO.
             VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
             ---- ---- -----        -------------------      ----
   EVENT :             B           UNIT NO.   3   HEAT-EX    EX-OR

 T-EVENT :    BQ    5  LO          UNIT NO.   3   HEAT-EX
 R-EVENT :             C           UNIT NO.   3   HEAT-EX


             NAME NO.
             VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
             ---- ---- -----        -------------------      ----
   EVENT :    BQ    5  LO          UNIT NO.   3   HEAT-EX    OR

 T-EVENT :    PA    7  LO          UNIT NO.   3   HEAT-EX
 T-EVENT :    AP    5  HI          UNIT NO.   3   HEAT-EX
 B-EVENT :             BLOCKAGE    UNIT NO.   3   HEAT-EX
 B-EVENT :             LK-LP-ENV   UNIT NO.   3   HEAT-EX


             NAME NO.
             VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
             ---- ---- -----        -------------------      ----
   EVENT :    PA    7  LO          UNIT NO.   3   HEAT-EX    OR

 T-EVENT :    QB    7  LO          UNIT NO.   3   HEAT-EX
 B-EVENT :             LK-LP-ENV   UNIT NO.   3   HEAT-EX


             NAME NO.
             VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
             ---- ---- -----        -------------------      ----
   EVENT :    QB    7  LO          UNIT NO.   3   HEAT-EX    OR

 R-EVENT :             D           UNIT NO.  12   CNTRL-VAL
 T-EVENT :    BB   12  LO          UNIT NO.  12   CNTRL-VAL
 B-EVENT :             BLOCKAGE    UNIT NO.  12   CNTRL-VAL
 B-EVENT :             LK-LP-ENV   UNIT NO.  12   CNTRL-VAL
 B-EVENT :             FAI-CLOSE   UNIT NO.  12   CNTRL-VAL


             NAME NO.
             VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
             ---- ---- -----        -------------------      ----
   EVENT :             D           UNIT NO.  12   CNTRL-VAL  AND

 T-EVENT :    PP   11  LO          UNIT NO.  12   CNTRL-VAL
```

Fig. 6.25 /continued

```
PAGE
---------
```

```
R-EVENT :              C            UNIT NO. 10   CNTRL-VAL


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT      GATE
           ---- ---- -----          ------------------       ----
   EVENT :  PP   11  LO             UNIT NO. 10   CNTRL-VAL  OR

T-EVENT :   QQ   11  LO             UNIT NO. 10   CNTRL-VAL
B-EVENT :            LK-LP-ENV UNIT NO. 10   CNTRL-VAL


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT      GATE
           ---- ---- -----          ------------------       ----
   EVENT :  QQ   11  LO             UNIT NO. 10   CNTRL-VAL  OR

T-EVENT :   PP   10  LO             UNIT NO.  9   SENSOR-Q


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT      GATE
           ---- ---- -----          ------------------       ----
   EVENT :  PP   10  LO             UNIT NO.  9   SENSOR-Q   OR

T-EVENT :   QQ   10  LO             UNIT NO.  9   SENSOR-Q


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT      GATE
           ---- ---- -----          ------------------       ----
   EVENT :  QQ   10  LO             UNIT NO.  9   SENSOR-Q   OR

T-EVENT :   PP    9  LO             UNIT NO.  8   CENT-PUMP
T-EVENT :   QQ    9  LO             UNIT NO.  8   CENT-PUMP
B-EVENT :            BLOCKAGE       UNIT NO.  8   CENT-PUMP
B-EVENT :            LK-LP-ENV UNIT NO.  8   CENT-PUMP


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT      GATE
           ---- ---- -----          ------------------       ----
   EVENT :  PP    9  LO             UNIT NO.  8   CENT-PUMP  OR

T-EVENT :   QQ    9  LO             UNIT NO.  8   CENT-PUMP
B-EVENT :            LK-LP-ENV UNIT NO.  8   CENT-PUMP


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT      GATE
           ---- ---- -----          ------------------       ----
   EVENT :  QQ    9  LO             UNIT NO.  8   CENT-PUMP
                                    *DIAMOND EVENT*
```

Fig. 6.25 /continued

```
PAGE
--------
```

```
               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT     GATE
               ---- ---- -----       ------------------      ----
    EVENT :    UU    9   LO         UNIT NO.  8   CENT-PUMP  OR

  T-EVENT :    PP    9   HI         UNIT NO.  7   DUMMY-H


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT     GATE
               ---- ---- -----       ------------------      ----
    EVENT :    PP    9   HI         UNIT NO.  7   DUMMY-H    OR

  B-EVENT :              BLOCKAGE   UNIT NO.  8   CENT-PUMP


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT     GATE
               ---- ---- -----       ------------------      ----
    EVENT :              C          UNIT NO. 12   CNTRL-VAL  OR

  T-EVENT :    BB   12   NO-CHANGE  UNIT NO. 12   CNTRL-VAL
  B-EVENT :              VALV-STCK  UNIT NO. 12   CNTRL-VAL
  B-EVENT :              MANUAL     UNIT NO. 12   CNTRL-VAL


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT     GATE
               ---- ---- -----       ------------------      ----
    EVENT :    BB   12   NO-CHANGE  UNIT NO. 12   CNTRL-VAL  OR

  T-EVENT :    SS    8   NO-CHANGE  UNIT NO. 11   CNTROLLER
  B-EVENT :              CONT-STCK  UNIT NO. 11   CNTROLLER


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT     GATE
               ---- ---- -----       ------------------      ----
    EVENT :    SS    8   NO-CHANGE  UNIT NO. 11   CNTROLLER  OR

  B-EVENT :              SENS-STCK  UNIT NO.  4   SENSOR-T


               NAME NO.
               VAR. VAR. FAULT       DESCRIPTION OF UNIT     GATE
               ---- ---- -----       ------------------      ----
    EVENT :    BB   12   LO         UNIT NO. 12   CNTRL-VAL  OR

  T-EVENT :    SS    8   HI         UNIT NO. 11   CNTROLLER
  T-EVENT :    WW   13   LO         UNIT NO. 11   CNTROLLER
  B-EVENT :              CONT-F-LO  UNIT NO. 11   CNTROLLER


               NAME NO.
```

Fig. 6.25 /continued

```
PAGE
--------
```

```
              VAR.  VAR.  FAULT          DESCRIPTION OF UNIT        GATE
              ----  ----  -----          -------------------        ----
    EVENT :   "SS"  "8"   HI             UNIT NO. 11   CNTROLLER    OR
 B-EVENT :                SEN-FA-HI  UNIT NO.  4    SENSOR-T


              NAME  NO.
              VAR.  VAR.  FAULT          DESCRIPTION OF UNIT        GATE
              ----  ----  -----          -------------------        ----
    EVENT :   "WW"  "13"  LO             UNIT NO. 11   CNTROLLER    ........
                                          *DIAMOND EVENT*


              NAME  NO.
              VAR.  VAR.  FAULT          DESCRIPTION OF UNIT        GATE
              ----  ----  -----          -------------------        ----
    EVENT :   "AP"  "5"   HI             UNIT NO.  3   HEAT-EX      ........
                                          *DIAMOND EVENT*


              NAME  NO.
              VAR.  VAR.  FAULT          DESCRIPTION OF UNIT        GATE
              ----  ----  -----          -------------------        ----
    EVENT :   ....  ....  C              UNIT NO.  3   HEAT-EX      AND
 T-EVENT :    BQ     5    NO-FLOW    UNIT NO.  3    HEAT-EX
 T-EVENT :    QQ     3    GT0        UNIT NO.  3    HEAT-EX


              NAME  NO.
              VAR.  VAR.  FAULT          DESCRIPTION OF UNIT        GATE
              ----  ----  -----          -------------------        ----
    EVENT :   "BQ"  "5"   NO-FLOW    UNIT NO.  3   HEAT-EX          OR
 T-EVENT :    QB     7    NO-FLOW    UNIT NO.  3    HEAT-EX
 B-EVENT :                COMP-BLOC  UNIT NO.  3    HEAT-EX


              NAME  NO.
              VAR.  VAR.  FAULT          DESCRIPTION OF UNIT        GATE
              ----  ----  -----          -------------------        ----
    EVENT :   "QB"  "7"   NO-FLOW    UNIT NO.  3   HEAT-EX          OR
 T-EVENT :    QQ    11    NO-FLOW    UNIT NO. 12    CNTRL-VAL
 B-EVENT :                COMP-BLOC  UNIT NO. 12    CNTRL-VAL
 B-EVENT :                SHUT       UNIT NO. 12    CNTRL-VAL


              NAME  NO.
              VAR.  VAR.  FAULT          DESCRIPTION OF UNIT        GATE
              ----  ----  -----          -------------------        ----
    EVENT :   "QQ"  "11"  NO-FLOW    UNIT NO. 12   CNTRL-VAL        OR
 T-EVENT :    QQ    12    NO-FLOW    UNIT NO.  9    SENSOR-Q
```

Fig. 6.25 /continued

```
AGE
--------
..............
```

```
B-EVENT :                    COMP-BLOC UNIT NO.  9   SENSOR-Q


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------         ----
    EVENT :    QQ    10  NO-FLOW    UNIT NO.   9    SENSOR-Q     OR

  T-EVENT :    QQ     9  NO-FLOW    UNIT NO.   8    CENT-PUMP
  B-EVENT :              SHUTDOWN   UNIT NO.   8    CENT-PUMP
  B-EVENT :              COMP-BLOC  UNIT NO.   8    CENT-PUMP


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------         ----
    EVENT :    QQ     9  NO-FLOW    UNIT NO.   8    CENT-PUMP
                                     *DIAMOND EVENT*


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------         ----
    EVENT :    QQ     3  GT0        UNIT NO.   3    HEAT-EX      OR

  T-EVENT :    QQ     2  GT0        UNIT NO.   3    HEAT-EX


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------         ----
    EVENT :    QQ     2  GT0        UNIT NO.   3    HEAT-EX      AND

  T-EVENT :    QQ     1  GT0        UNIT NO.   2    CNTRLV-SC
  T-EVENT :    SB     6  NO-SIGNAL  UNIT NO.   2    CNTRLV-SC


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------         ----
    EVENT :    QQ     1  GT0        UNIT NO.   2    CNTRLV-SC
                                     *DIAMOND EVENT*


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----       -------------------         ----
    EVENT :    SB     6  NO-SIGNAL  UNIT NO.   2    CNTRLV-SC    EX-OR

  T-EVENT :    QQ    10  GT0        UNIT NO.   9    SENSOR-Q
  R-EVENT :              A          UNIT NO.   9    SENSOR-Q


               NAME NO.
```

Fig. 6.25 /continued

```
PAGE
--------
```

```
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  QQ   10  GT0       UNIT NO.   9     SENSOR-Q    OR

 T-EVENT :  QQ    9  GT0       UNIT NO.   8     CENT-PUMP


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  QQ    9  GT0       UNIT NO.   8     CENT-PUMP
                                 *DIAMOND EVENT*


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :           A         UNIT NO.   9     SENSOR-Q    AND

 T-EVENT :  QQ   10  NO-FLOW   UNIT NO.   9     SENSOR-Q
 B-EVENT :           SI-STR-PL UNIT NO.   9     SENSOR-Q


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  QQ   10  NO-FLOW   UNIT NO.   9     SENSOR-Q    OR

 T-EVENT :  QQ    9  NO-FLOW   UNIT NO.   8     CENT-PUMP
 B-EVENT :           SHUTDOWN  UNIT NO.   8     CENT-PUMP
 B-EVENT :           COMP-BLOC UNIT NO.   8     CENT-PUMP


           NAME NO.
           VAR. VAR. FAULT       DESCRIPTION OF UNIT        GATE
           ---- ---- -----       -------------------        ----
   EVENT :  QQ    9  NO-FLOW   UNIT NO.   8     CENT-PUMP
                                 *DIAMOND EVENT*
```

Fig. 6.25

```
PAGE
--------
```

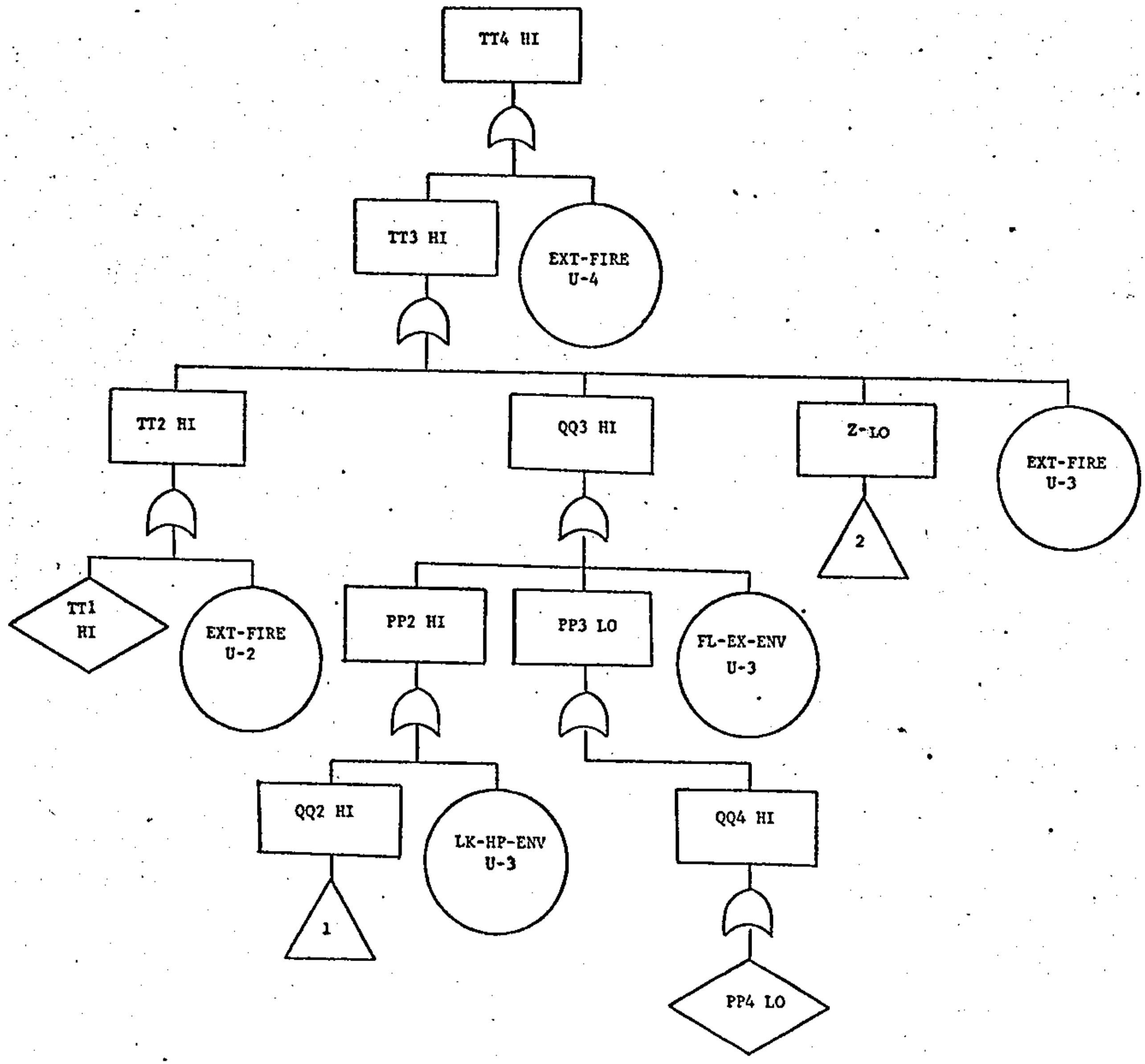


Fig. 6.26 Fault tree for TT4 HI drawn using the data of Fig. 6.25

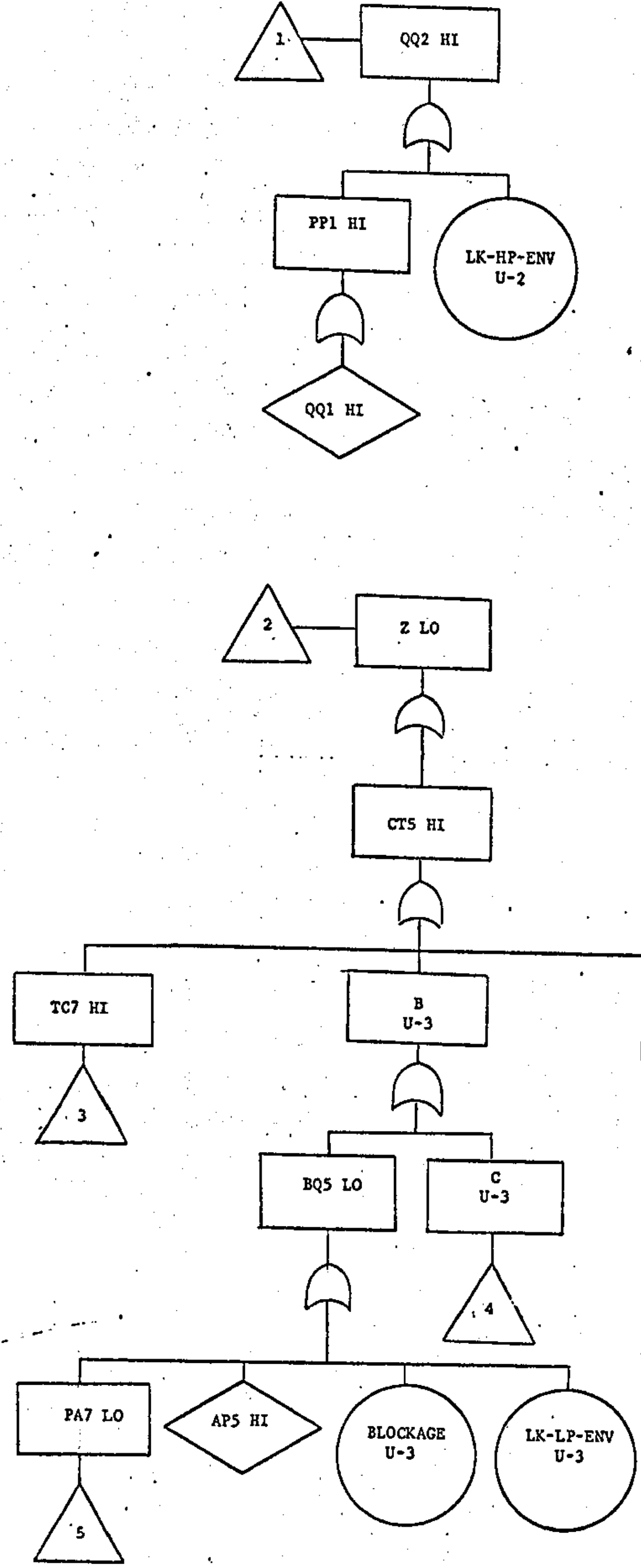


Fig. 6.26 /continued

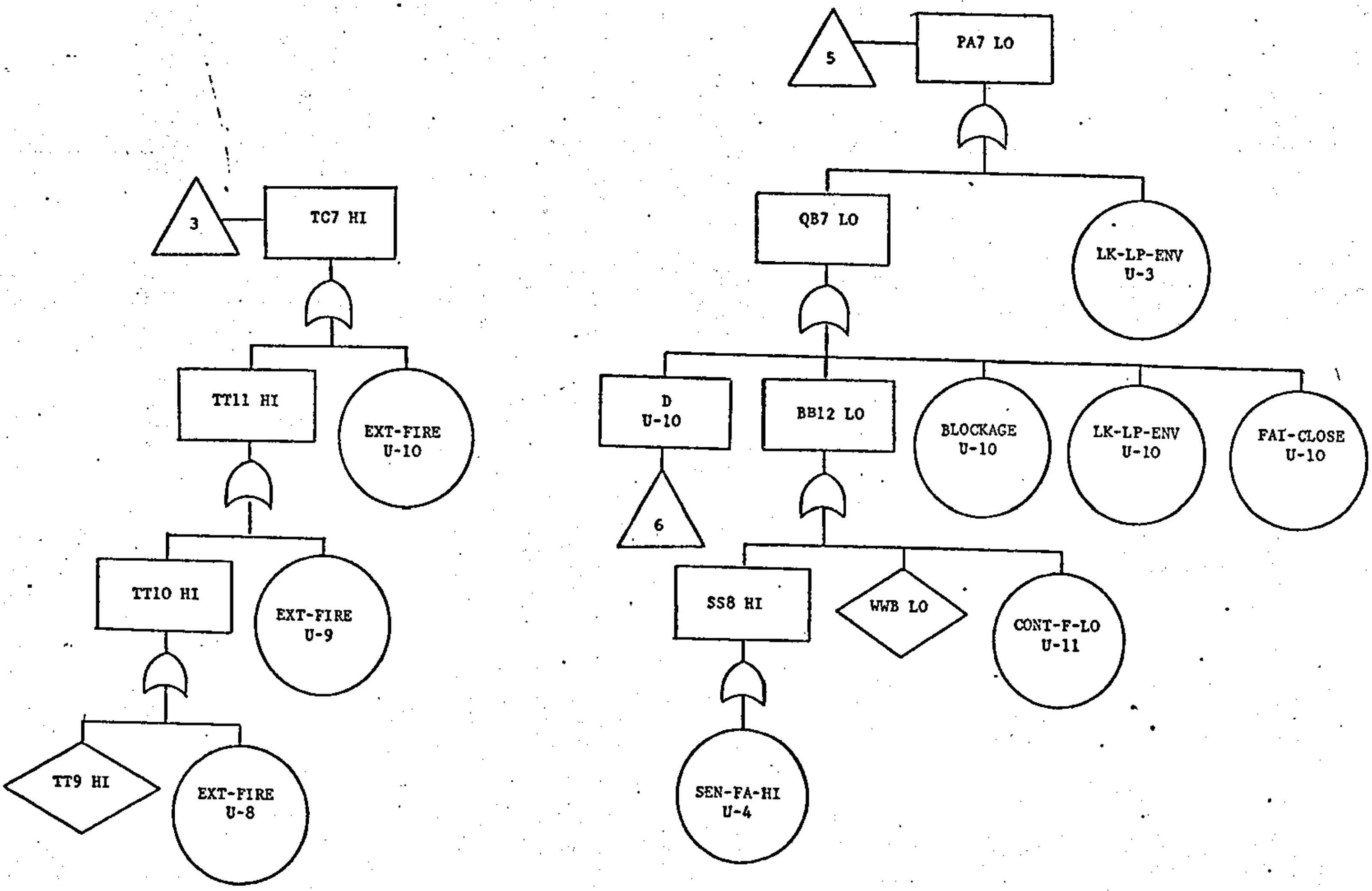


Fig. 6.26 /continued

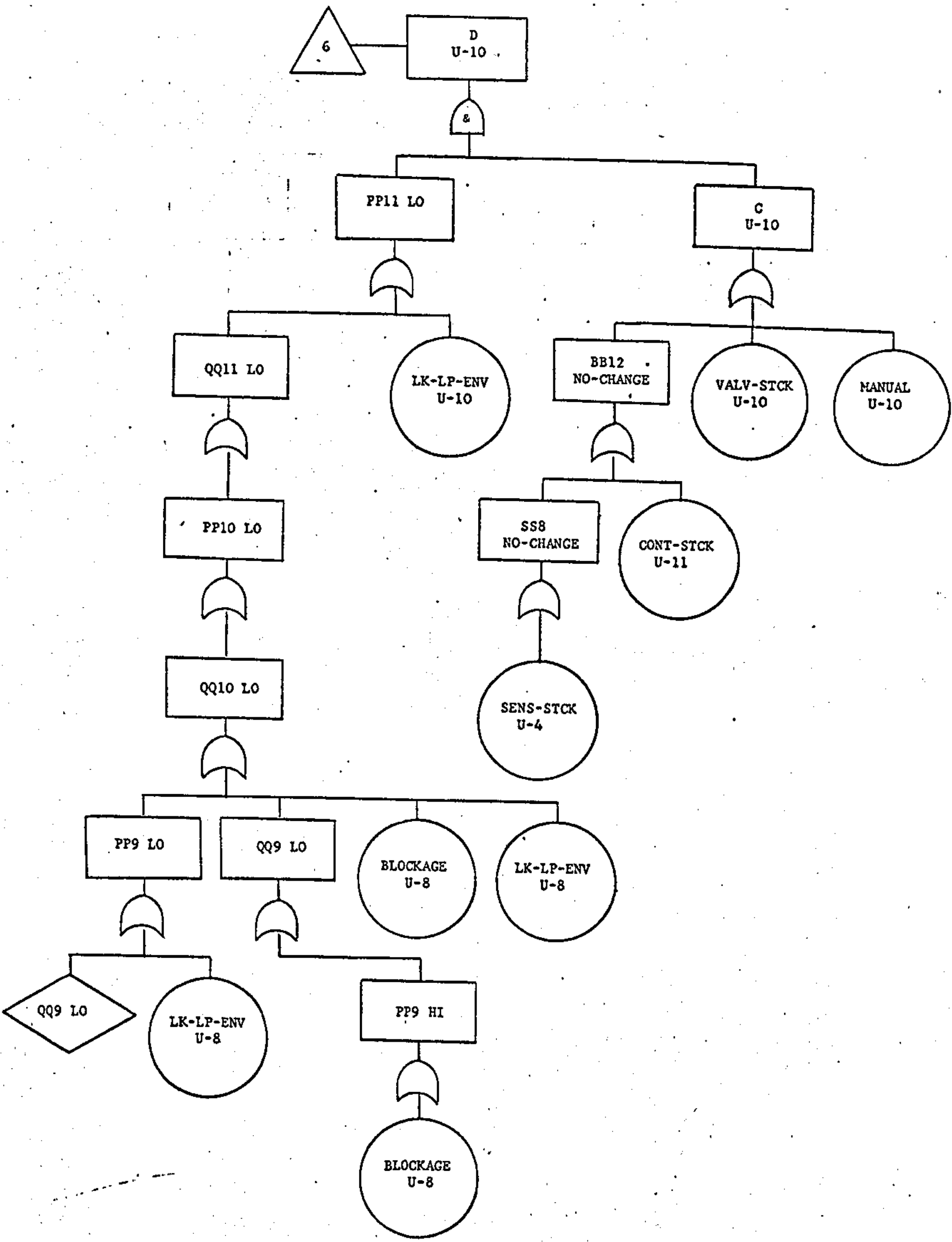


Fig. 6.26 /continued

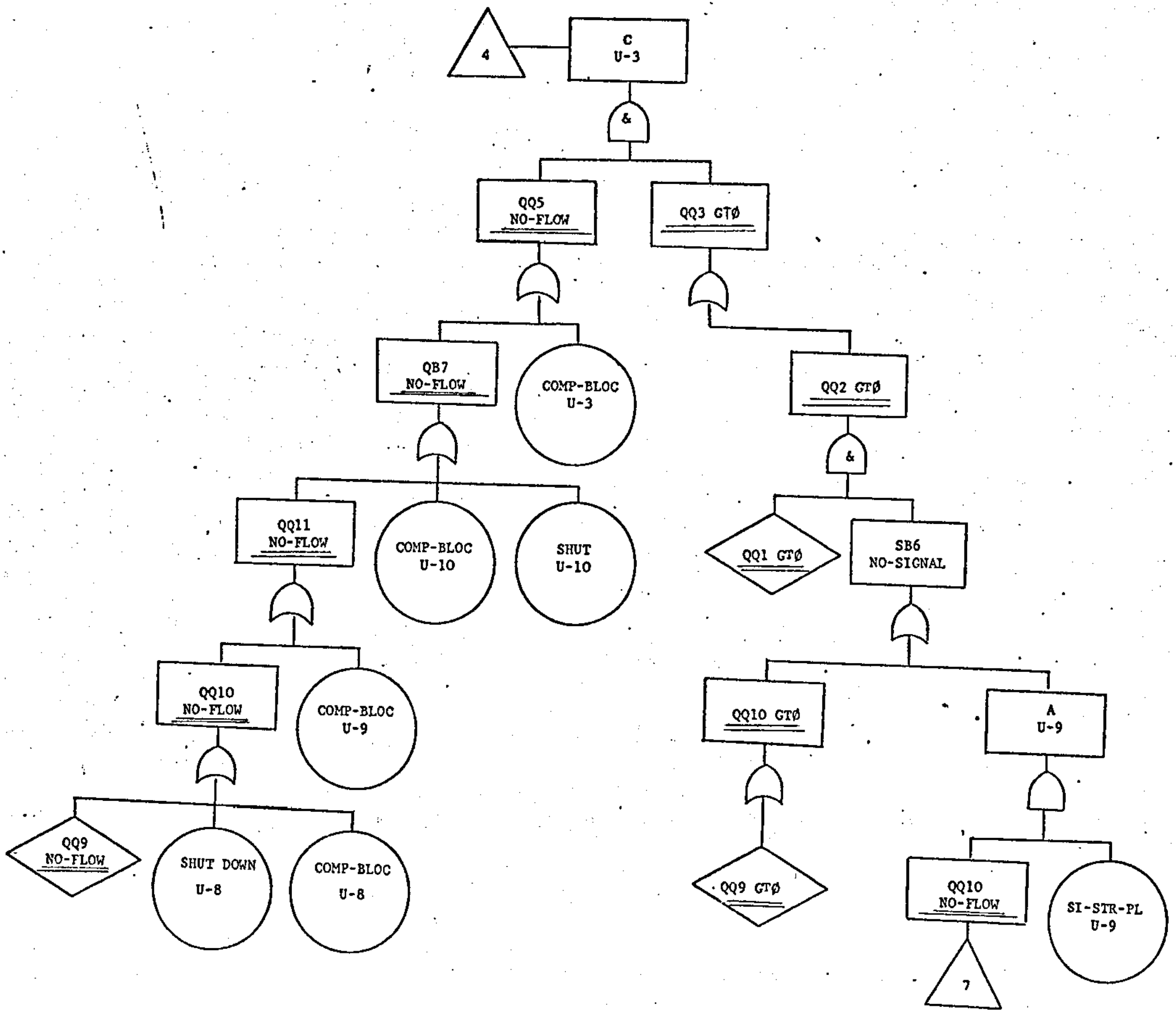


Fig. 6.26 /continued

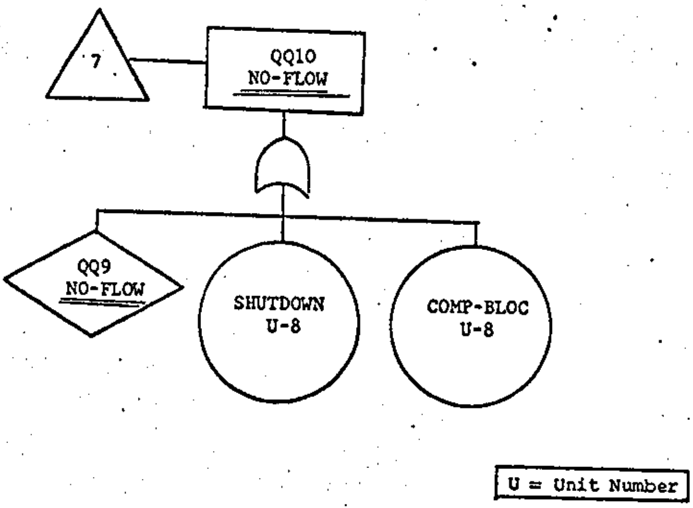

7
QQ10
NO-FLOW
QQ9
NO-FLOW
SHUTDOWN
U-8
COMP-BLOC
U-8
U = Unit Number

Fig. 6.26

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ? Y


                                  MEASURED VARIABLES
                                  ******************

          VARIABLES     HI LIMIT     LO LIMIT     STATUS      PRIORITY
          ---------     --------     --------     ------      --------
          .........     ........     ........     ......      ........

             QQ  1        100             0         OK             1
             TT  1        100             0         OK             2
             SB  6        100             0         OK            24
             QQ  2        100             0         OK             4
             TT  2        100             0         OK             5
             QB  7        100             0         OK            28
             TC  7        100             0         OK            29
             QQ  3        100             0         OK             7
             TT  3        100             0         OK             8
             BQ  5        100             0         OK            10
             CT  5        100             0         OK            11
             QQ  4        100             0         OK            13
             TT  4        100             0         OK            14
             SS  8        100             0         OK            16
             QQ  9        100             0         OK            18
             TT  9        100             0         OK            19
             QQ 10        100             0         OK            21
             TT 10        100             0         OK            22
             QQ 11        100             0         OK            25
             TT 11        100             0         OK            26
             BB 12        100             0         OK            33
             WW 13        100             0         OK            34

READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 6.27 Measured variables in the Heat Exchanger and control loop system before their values were modified

```
PAGE
---------
.........
```

```
DO YOU WANT TO LIST THE VARIABLES(Y/N) ? Y


                         MEASURED VARIABLES
                         ******************

     VARIABLES      HI LIMIT     LO LIMIT    STATUS    PRIORITY
     ---------      --------     --------    ------    --------
     .........      ........     ........    ......    ........

        QQ  1          100            0        OK           1
        TT  1          100            0        OK           2
        SB  6          100            0        OK          24
        QQ  2          100            0        OK           4
        TT  2          100            0        OK           5
        QB  7          100           20        LO          28
        TC  7          100            0        OK          29
        QQ  3          100            0        OK           7
        TT  3          100            0        HI           8
        BQ  5          100           20        LO          10
        CT  5          100            0        HI          11
        QQ  4          100            0        OK          13
        TT  4          100            0        HI         250
        SS  8          100            0        OK          16
        QQ  9          100           20        LO          18
        TT  9          100            0        OK          19
        QQ 10          100           20        LO          21
        TT 10          100            0        OK          22
        QQ 11          100           20        LO          25
        TT 11          100            0        OK          26
        BB 12          100            0        OK          33
        WW 13          100            0        OK          34


READING VALUES OF MEASURED VARIABLES
************************************


NAME OF VAR.(ADD %,* TO TERM.) ? *%
```

Fig. 6.28 Measured variables in the Heat-Exchanger and control loop system after the values have been modified

```
PAGE
--------
........
```

```
NOMENCLATURE :
-------------

B-EVENT : IS A BASIC EVENT AND DOES NOT REQUIRE FURTHER DEVELOPMENT

R-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT BUT IT
          IS RELATED TO A REPLACED EVENT IN THE UNIT

T-EVENT : IS AN EVENT THAT REQUIRES FURTHER DEVELOPMENT

********************************************************************
*************************   REAL  TIME   ***************************
********************************************************************

                         *** FAULT TREE ***
                         ------------------



            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
TOP EVENT :  TT    4  HI        UNIT NO.  4   SENSOR-T     OR

  T-EVENT :  TT    3  HI        UNIT NO.  4   SENSOR-T
  B-EVENT :           EXT-FIRE  UNIT NO.  4   SENSOR-T


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
    EVENT :  TT    3  HI        UNIT NO.  4   SENSOR-T     OR

  R-EVENT :           Z-LO      UNIT NO.  3   HEAT-EX
  B-EVENT :           EXT-FIRE  UNIT NO.  3   HEAT-EX


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
    EVENT :           Z-LO      UNIT NO.  3   HEAT-EX      OR

  T-EVENT :  CT    5  HI        UNIT NO.  3   HEAT-EX


            NAME NO.
            VAR. VAR. FAULT      DESCRIPTION OF UNIT       GATE
            ---- ---- -----      -------------------       ----
    EVENT :  CT    5  HI        UNIT NO.  3   HEAT-EX      OR

  R-EVENT :           B         UNIT NO.  3   HEAT-EX
  B-EVENT :           EXT-FIRE  UNIT NO.  3   HEAT-EX


            NAME NO.
```

Fig. 6.29 Fault tree for top event TT4 HI in a Heat-Exchanger and control loop system

```
PAGE
--------
```

/continued

```
              VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
              ---- ---- -----        -------------------      ----
   EVENT :              B           UNIT NO.  3   HEAT-EX     EX-OR

 T-EVENT :     BQ    5  LO          UNIT NO.  3   HEAT-EX



              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
              ---- ---- -----        -------------------      ----
   EVENT :    BQ     5  LO          UNIT NO.  3   HEAT-EX     OR

 T-EVENT :     PA    7  LO          UNIT NO.  3   HEAT-EX
 T-EVENT :     AP    5  HI          UNIT NO.  3   HEAT-EX
 B-EVENT :              BLOCKAGE    UNIT NO.  3   HEAT-EX
 B-EVENT :              LK-LP-ENV   UNIT NO.  3   HEAT-EX



              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
              ---- ---- -----        -------------------      ----
   EVENT :    PA     7  LO          UNIT NO.  3   HEAT-EX     OR

 T-EVENT :     QB    7  LO          UNIT NO.  3   HEAT-EX
 B-EVENT :              LK-LP-ENV   UNIT NO.  3   HEAT-EX



              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
              ---- ---- -----        -------------------      ----
   EVENT :    QB     7  LO          UNIT NO.  3   HEAT-EX     OR

 R-EVENT :              D           UNIT NO. 12   CNTRL-VAL
 B-EVENT :              BLOCKAGE    UNIT NO. 12   CNTRL-VAL
 B-EVENT :              LK-LP-ENV   UNIT NO. 12   CNTRL-VAL
 B-EVENT :              FAI-CLOSE   UNIT NO. 12   CNTRL-VAL



              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
              ---- ---- -----        -------------------      ----
   EVENT :              D           UNIT NO. 12   CNTRL-VAL   AND

 T-EVENT :     PP   11  LO          UNIT NO. 12   CNTRL-VAL
 R-EVENT :              C           UNIT NO. 12   CNTRL-VAL



              NAME NO.
              VAR. VAR. FAULT        DESCRIPTION OF UNIT      GATE
              ---- ---- -----        -------------------      ----
   EVENT :    PP    11  LO          UNIT NO. 12   CNTRL-VAL   OR

 T-EVENT :     QQ   11  LO          UNIT NO. 12   CNTRL-VAL
 B-EVENT :              LK-LP-ENV   UNIT NO. 12   CNTRL-VAL
```

Fig. 6.29 /continued

```
PAGE
---------
```

```
               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----        -----------------          ----
   EVENT :     "QQ" "11" LO           UNIT NO. 12   CNTRL-VAL    OR

 T-EVENT :      PP   12  LO           UNIT NO.  9   SENSOR-Q


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----        -----------------          ----
   EVENT :     "PP" "12" LO           UNIT NO.  9   SENSOR-Q     OR

 T-EVENT :      QQ   12  LO           UNIT NO.  9   SENSOR-Q


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----        -----------------          ----
   EVENT :     "QQ" "12" LO           UNIT NO.  9   SENSOR-Q     OR

 T-EVENT :      PP    9  LO           UNIT NO.  8   CENT-PUMP
 T-EVENT :      QQ    9  LO           UNIT NO.  8   CENT-PUMP
 B-EVENT :               BLOCKAGE     UNIT NO.  8   CENT-PUMP
 B-EVENT :               LK-LP-ENV    UNIT NO.  8   CENT-PUMP


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----        -----------------          ----
   EVENT :     "PP"  "9" LO           UNIT NO.  8   CENT-PUMP    OR

 T-EVENT :      QQ    9  LO           UNIT NO.  8   CENT-PUMP
 B-EVENT :               LK-LP-ENV    UNIT NO.  8   CENT-PUMP


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----        -----------------          ----
   EVENT :     "QQ"  "9" LO           UNIT NO.  8   CENT-PUMP
                                        *DIAMOND EVENT*


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----        -----------------          ----
   EVENT :     "QQ"  "9" LO           UNIT NO.  8   CENT-PUMP    OR

 T-EVENT :      PP    9  HI           UNIT NO.  7   DUMMY-H


               NAME NO.
               VAR. VAR. FAULT        DESCRIPTION OF UNIT        GATE
               ---- ---- -----        -----------------          ----
```

Fig. 6.29 /continued

```
PAGE
---------
```

```
   EVENT :   PP     9  HI          UNIT NO.  7   DUMMY-H      OR

 B-EVENT :             BLOCKAGE    UNIT NO.  8   CENT-PUMP


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT       GATE
           ---- ---- -----          -------------------       ----
   EVENT :           C             UNIT NO. 12   CNTRL-VAL    OR

 B-EVENT :             VALV-STCK  UNIT NO. 12   CNTRL-VAL
 B-EVENT :             MANUAL     UNIT NO. 12   CNTRL-VAL


           NAME NO.
           VAR. VAR. FAULT          DESCRIPTION OF UNIT       GATE
           ---- ---- -----          -------------------       ----
   EVENT :   AP     5  HI          UNIT NO.  3   HEAT-EX
                                     *DIAMOND EVENT*
```

Fig. 6.29

```
PAGE
---------
```

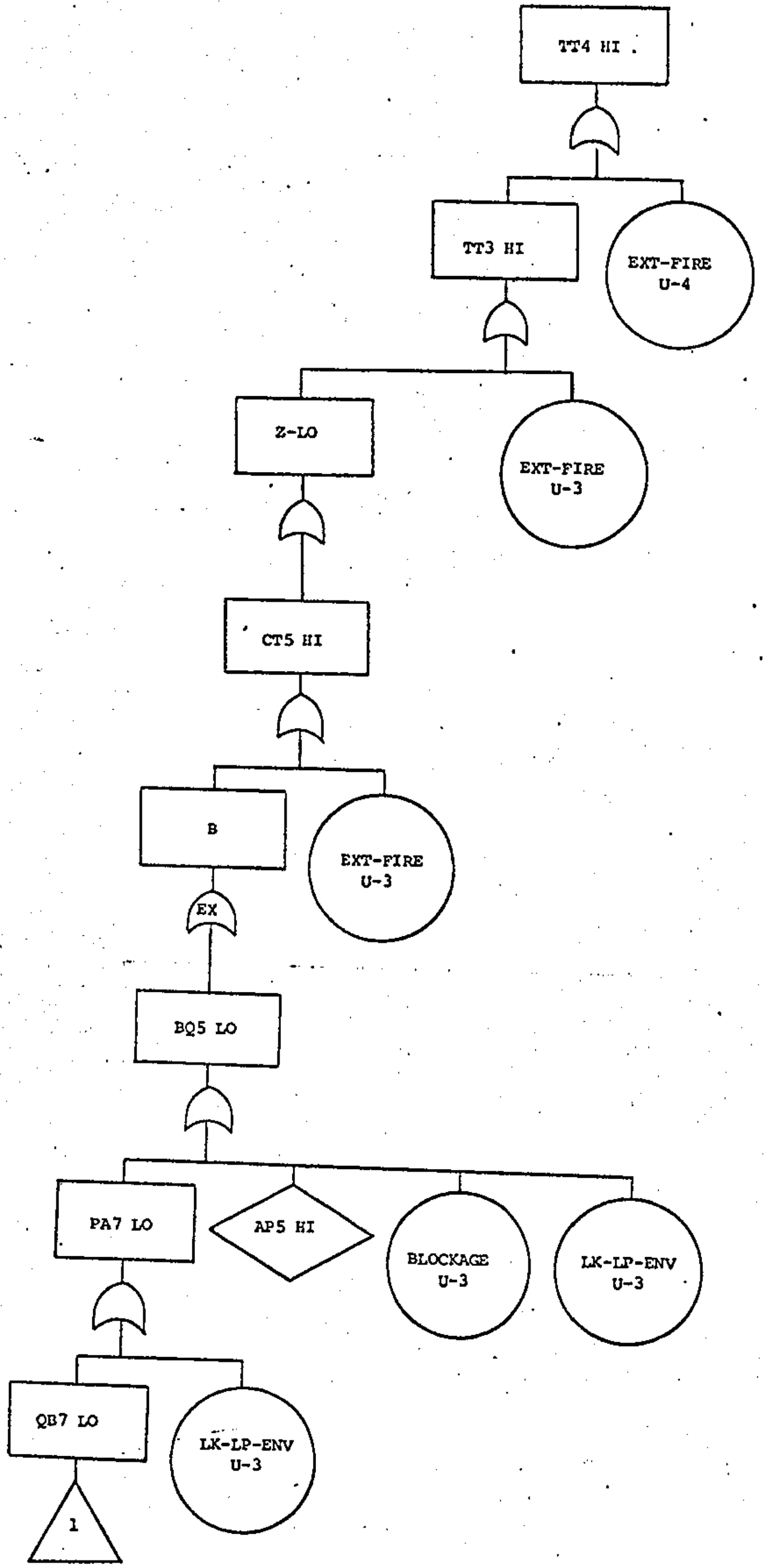


Fig. 6.30 Fault tree for TT4 HI drawn using the data of Fig. 6.29

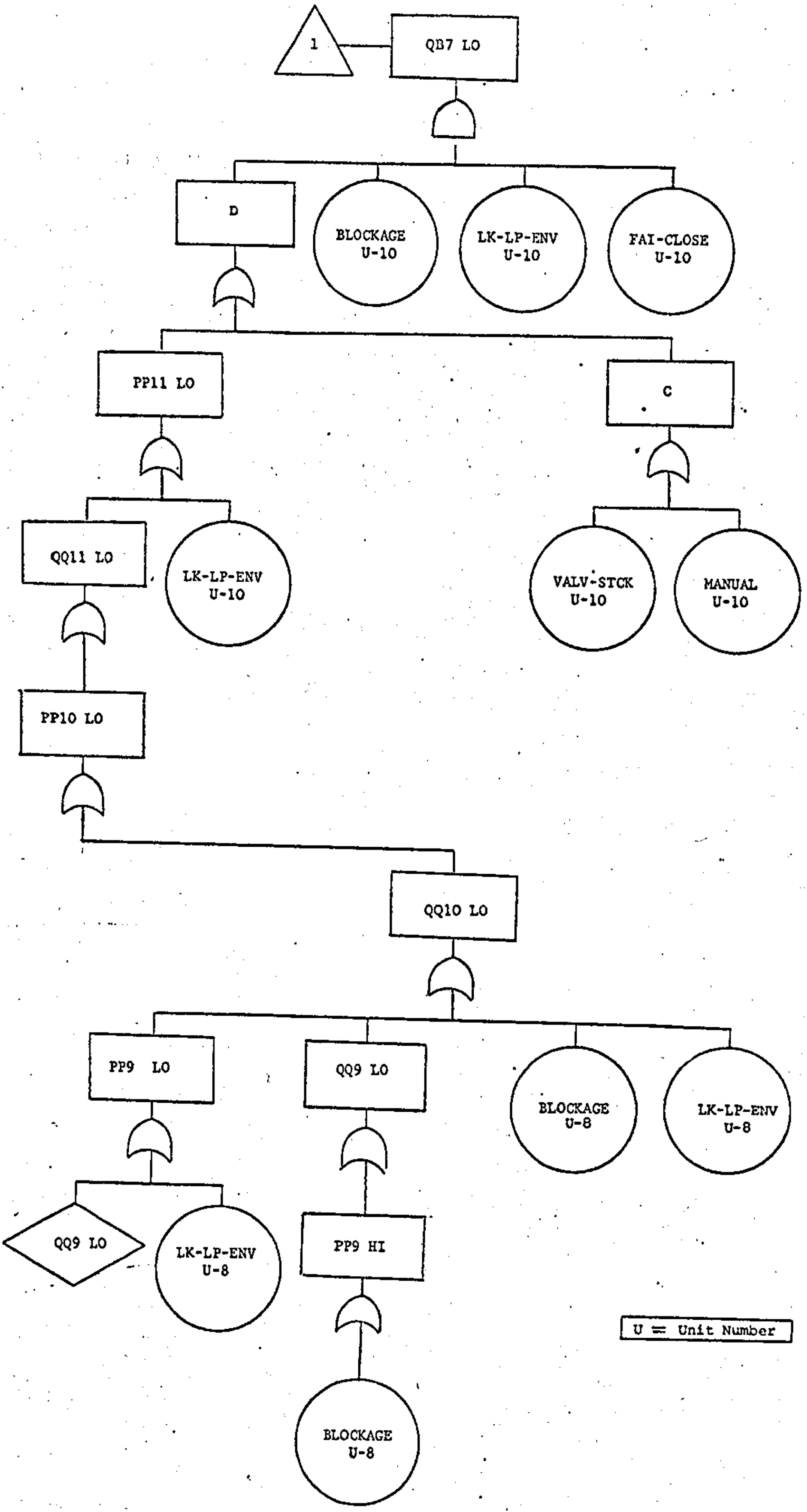

1
QB7 LO
D
BLOCKAGE U-10
LK-LP-ENV U-10
FAI-CLOSE U-10
PP11 LO
C
QQ11 LO
LK-LP-ENV U-10
VALV-STCK U-10
MANUAL U-10
PP10 LO
QQ10 LO
PP9 LO
QQ9 LO
BLOCKAGE U-8
LK-LP-ENV U-8
QQ9 LO
LK-LP-ENV U-8
PP9 HI
BLOCKAGE U-8
U = Unit Number

Fig. 6.30

Chapter 7

DISCUSSION AND CONCLUSIONS

The methodology and its applications were presented in the previous chapters. It is likely that as experience is gained on the applications of fault trees to Chemical Process Plants, the methodology and the programs developed may need some alterations. A review of the work, considering the problems that may arise, together with the conclusions and recommendations for further work is contained in this chapter.

7.1 Unit Models

The unit approach used in this work has proven quite useful in the development of the methodology for the synthesis of fault trees. The examples shown in Chapter 6 illustrate how the use of the conventions stated in Chapter 3 for the setting of the variables in the unit models, provides a two-way flow of information. This is a major feature of the methodology because it allows the construction of fault trees at any place in the system, not only at the downstream end.

The use of unit models has the advantage that it focuses the safety analyst's attention on a manageable portion of the problem. It also enables him to produce a library of models for use as required rather than models that only reflect particular plants. The unit models used in this work are general models that can be easily recognised by the engineer who is not a fault tree specialist. This is an advantage at the design stage of a plant, because the gap between a fault tree specialist and a non-specialist engineer is narrowed, greater interaction can be achieved between the design engineer and the safety analyst.

Unit models are flexible and when changes are required, it is only the specific unit involved that requires attention, and not the whole plant being considered.

7.2 The minitrees

The minitrees used in the methodology are the basis for the construction of the fault trees. Although in this work they were produced manually, the production of minitrees based on the equations is carried out without difficulty. It was decided to use this approach because it was a simple and straightforward method that can encourage a non-fault tree specialist to produce his own minitrees. The only requirements are the unit models and some knowledge of the internal faults that may affect the different states of the Left Hand Side (LHS) variables of the equations. When the failure analysis is carried out to produce the minitrees a better insight of the unit being considered is obtained. As experience is gained in the use of the minitrees to produce the fault trees, some modifications or other states of the variables are found, that need to be included in the minitrees, so that the set of minitrees for each unit can have a higher quality. This again is not a problem in the methodology because the unit approach allows the addition or modification of the minitrees without problem. This feature can be very useful at the design stage of a Plant. Any modification in the design can easily be carried out in the minitrees of the units involved and up-to-date fault trees for the design can be produced. The quality of the fault trees produced by the methodology is closely related to the quality of the minitrees used to build them. If the failure analysis is carried out thoroughly the unit minitrees will reflect this and the results obtained will be better.

The minitrees used in this work for each unit have some similarity with Fussell's(F2) failure transfer functions. The difference is that the minitrees used here are based on equations which describe the behaviour of the unit and allowed the flow of information in two directions. Fussell's were obtained on the basis of experience of the mechanisms that could cause the failure of the component. His transfer functions only allow the flow of information in one way. An explanation of these differences is that Fussell's work was applied to simple electrical systems, whose components are less complex than those components that are used in chemical plants and hence the failure modes analysis can be carried out directly.

The decision tables approach used by Salem(S1) et al., has some similarities to the minitrees used here. The main difference with this work is that their decision tables, although quite complete because they consider several states of the variables, do not have the two-way flow property that the minitrees have. Therefore, their trees are restricted to be developed only in one direction for top events that are located at the end of the system. Lapp and Powers'(L4) methodology has the same problem, only one way flow of information is considered.

The nomenclature used for the development of the minitrees during this work is particular to the methodology but not very different from that commonly used in the literature. The Replaced events (R events) help avoid the repetition of events that appear sometimes in the minitrees and make it easier to modify them. The nomenclature used could be considered equivalent to the one defined by Fussell for his Ordered fault events in his Methodology.

7.3 Control Loops and Multi-State Variables

One of the problems that has prevented the use of the fault tree techniques in Chemical Process plants is the presence of control loops and multi-state variables. The methodology described in this work does not need to give the loops and multistate variables a special treatment.

Due to the unit approach used, the control loop is not considered as a whole. Each component of the loop is considered a unit in itself and has its own minitrees obtained according to the rules mentioned in Chapter 3. Note how the minitrees shown in Appendix I for the controller, the sensors and the control valve (which are the components of the control loops considered in the examples of Chapter 6) are defined, based only on the equations of their unit models and the internal faults, that may affect the different states considered for the LHS variables. The two-way information flow is very useful in this case. The methodology only requires the topology of the system and the unit minitrees of the components of the system. The information travels from one unit to the other and the correct minitrees are linked together to produce the final tree. A good example of how the methodology can handle control loops are the fault trees shown in Chapter 6. Note that the general models used for the units provide general minitrees. An example of this is the flow sensor unit. The same set of minitrees was used in two different loops in Chapter 6. It was used first in the two tank and control loop system and later on in the Heat-Exchanger and Control loops system with satisfactory results.

A good example of how the methodology deals with multi-state variables is the Heat-Exchanger and Control Loops system shown in Chapter 6. For this example a trip valve model was developed. It was considered as a special case of the control valve where the normal state of the valve was considered to be wide-open. Some modifications were made to the control valve's minitrees, due to this consideration. A new state condition for the flow variable was considered NO-FLOW and the final result was the control-valve special case shown in Appendix I. Similar minitrees with the NO-FLOW condition were obtained (considering only the variable flow for simplicity) for the rest of the units so that flow of information due to this specific state of the Flow variable could be transmitted in the same fashion as the rest of the faults for the other variables. The top event TT4 HI (for design purposes) shown in Chapter 6 shows how the methodology linked up the correct minitrees and produced a fault tree that presents multi-state variables.

7.4 AND Gates and Boundary Conditions

In order to preserve the coherency of the fault trees produced by the methodology the Boundary Conditions and Not-allowed faults were used. Each time a new branch of the tree is developed by the methodology the new events are checked against the Boundary Conditions and Not-allowed faults already existing in the tree. Each gate of the tree carries these Boundary Conditions and Not-allowed faults for use further down the tree. Every time a new gate is inserted in the tree the existing Boundary Conditions and Not-allowed faults of the gate at the next level up, are passed onto the gate and the same process is repeated with the new gates that are created at the following levels. A detailed example of how the Boundary Conditions and Not-allowed Faults are used was presented in Chapter 4. The use of the Not-allowed faults in this fashion avoids the need for adding extra AND gates to the tree with the negation of the faults that could affect the coherency of the tree. Fig. 7.1 shows how the minitree for the fault Q_{OUT} HI would look if the Not-allowed fault "Blockage" in the Pipe model were added as an additional branch of the tree. This would require more space to store the minitrees and the process of building the tree would take more time because of the AND gates involved and the extra checking required.

When AND gates are involved in the minitrees sometimes the Boundary Conditions and Not-allowed faults detect an event that cannot be developed. This implies that the whole branch where the AND gate is, has to be deleted because it can no longer exist in the tree. The methodology takes care of this by means of a garbage collector program, that has been included in the algorithm that builds the trees. The garbage collector makes sure that all the AND gates that remain in the tree,after the deletion of the branch that was first found inconsistent, does not cause any contradiction in the remaining tree. This feature is a major advantage when the methodology is used for real time purposes. The AND gates that appear in the examples of Chapter 6 have all been checked by the algorithm before inclusion in the final tree. Note that the AND gates come from the minitrees used to generate the tree. They were generated during the failure analysis of the unit models. The algorithm only links the correct minitrees to produce the final tree. No gate or event is created by the algorithm.

7.5 The Implementation of the Methodology in a Digital Computer

The approach used in this work to implement the fault tree synthesis methodology, was the one of a standard program and specific data. This approach allows the flexibility required, so that, a relatively inexperienced user may use the methodology and produce fault trees for his plant model without difficulty. The language chosen to develop the programs was RTL/2. It is a language intended for real-time work, but incorporates many features of other languages such as Algol 68 and FORTRAN.

RTL/2 has some advantages over some more common programming languages such as FORTRAN. It incorporates some features that FORTRAN does not have and that are useful to handle (in an efficient way) the type of data employed to build the fault trees. FORTRAN was not suitable to deal with the problems involved in the generation of the trees because the computational problems involved in this work were quite different from those usually found in numerical programming problems. RTL/2 enables the programmer to declare new modes so that groups of related information (as was the case in this work) can be handled in a convenient way by means of records. List processing techniques were used in this work thanks to this powerful feature. The use of

FORTRAN to solve the same problems would have required many more arrays than those actually used in this work and it would have produced less efficient programs, because subscript evaluation is less efficient than component selection. Another reason why RTL/2 was used instead of FORTRAN or any other of the better known languages, was that RTL/2 is ideally suited for real time purposes and small computers which were some of the major features required for the use of the methodology described in this work.

7.5.1 The programs

Due to the space limitations imposed by the hardware of the system used in this work, the development of the programs was carried out in a manner intended to use the space available in the best possible way. 8 bit BYTES, rather than 16 bit words, were used in the definition of the MODES whenever it was possible. Bit manipulation was used in some of the programs to pack the most information in the minimum space. The faults and unit names were represented by integers related to look-up tables, so that strings were not needed in the handling of the trees. By inspecting the procedures it was found that some of them were similar in many respects to others. In order to save space, those procedures were, when possible, combined into one procedure.

In spite of the savings, the maximum number of units that was possible to handle with the facilities available was 15. The data base of 8K was able to accommodate 500 events and 150 gates. Usually half of the events and gates available were used for storing the different unit minitrees, leaving only the other half free to be used in the construction of the fault trees.

Fault trees with 60 gates and over 100 events have been produced by the algorithm used in this work. Considering the facilities available and the results reported in the literature,(L4) the results obtained during this study are encouraging. Most of the workers mentioned in the literature have not generated trees with more than 25 gates. The largest trees reported that have been synthesised by computer contain 143 gates. If a larger machine is used, it would be possible to have more space available and larger trees than those actually obtained during this study which can be produced without problem. Consider for example a data area of 24K. With this space available it would be possible to have more than 30 unit models and produce trees with more than 700 events and 300 gates without change to the programs.

The approach of a standard program and specific data used in this work, allows the use of a standard program to synthesise fault trees for almost any chemical process. The keystone to achieve this is the specific data of the system under study (minitrees and topology). The programs were developed bearing this in mind. Flexibility is always present, so that any change needed in the specific data can always be carried out without having to change the programs. With the help of the indirect command files and the editor provided with the operating system the time needed to carry out any modifications to the specific data is minimal. A good example of this, is the case of multi-state variables mentioned in Section 7.3. When it was necessary to include other states of the variable flow for the Heat-Exchanger and control loops system, use was made of the editor and the indirect command files. The trip-valve model was obtained from the control valve model by considering the normal state of the valve to be wide open. A copy of the minitrees for the control valve was made and modified by means of the editor to suit the needs of the trip valve. The minitrees for the NO-FLOW state and FLOW in the unit (GT0) were added to each of the unit files already available. This job was carried out in less than two hours and did not involve any modifications to the RTL/2 programs. Once the input data files for each unit minitree were modified the same tasks used in all the previous examples were employed to set up the minitrees for the units forming part of the system under study. The topology was defined and the trees shown in Chapter 6 were produced without problem. This flexibility would allow any change in the models, minitrees or topology that is required during the different stages of plant design. In this way, up-to-date trees could always be synthesised saving a lot of time for both the safety analyst and the design engineer. Fig. 7.2 helps to illustrate how the different tasks created during this study are used to set up the "environment" needed in order to build the minitrees. Task BTR is the standard program that constructs the fault trees; it interacts with the different parts of the data base to produce the fault trees.

7.6 Applications of the Methodology

The methodology described in this work can be used mainly for two purposes as has already been shown in the former chapters. However there are some other applications that may be considered:

1) The methodology could be used to help in the training of future control room operators. With the help of a computer, some situations could be simulated (via a task like RVA to change the value of the measured variables), so that the operator may learn from them and be exposed beforehand to cases that he might have to face in real time.

2) The use of fault trees in real time to produce a check list for the operator as was suggested by Lambert(L3) can now be carried out with the help of the methodology described here. It would require the implementation of a method to find the cut sets, similar to the ones described by Fussell and Vesely(F1) and a larger data base, but these improvements were not difficult to carry out, since the methodology is well defined.

3) If a method for the evaluation of the trees produced is implemented, the methodology has the capability (due to the way in which the MODES of the data base were defined) to produce the sort of Dual Tree described by Pandle(P3) et al., required to evaluate the minimum paths. The OR gates can easily be converted into AND gates and viceversa by means of setting and clearing bits in each of the gates of the tree under study.

4) If the methodology is used only for design purposes, it could be possible to translate the programs into ALGOL-68 and use them off-line in a bigger computer. This would solve the problem of memory limitations that could arise when larger sections of plant are analysed. The use of a bigger computer, would also allow more freedom in the formats to print the trees, and special programs to print the trees in a more tree-like format could be developed.

7.6.1 Real time application

The use of real time fault trees has been mentioned in the literature but only as a possibility of several possible applications of fault trees in the Chemical Process industry. The methodology presented in this work makes this possibility more likely. This work is the first to explore the use of synthesised fault trees for real time purposes, and therefore further considerations are still needed before a complete system with real time fault trees can be completely satisfactory for alarm analysis purposes.

Although it may seem that Real Time and design fault trees are treated in the same way by the methodology, there are some differences between them that have to be considered. These differences are:

1) Fault tree for design purposes only require the specific data of the system being considered (mini-trees and topology). No checking of the measured variables is required. More space is required for the trees because almost all their possible branches are developed.

2) Fault tree for real time purposes require, besides the specific data, the signals of the measured variables coming from the plant. Extra checking of the measured variables is involved but the trees are not likely to be as large as the ones for design purposes.

In spite of these differences the methodology has produced satisfactory results with consistent trees. In this work some of the real time problems, due to the incoming signals from the plant have been tackled with the help of the simulating task RVA. This task was used to change the values of the measured variables through a teletype. This approach was useful to test the methodology for the purposes of this study, but that does not mean that all the problems have been solved. The next step required is to use a pilot plant and obtain these signals from the plant. Some of the points that will require special attention are:

1) <u>The quality of the incoming signals from the measuring devices</u>

This is a very important point because if the quality of the signal received is not good erroneous fault trees could be produced.

2) The limitations imposed by the display devices

In this work the trees produced were printed in the same way as they were developed. This is certainly not an adequate format to display the trees to the operator. Some changes are needed in the formats to improve the way in which the information is presented. The experience of other industries such as the nuclear one, in the solution of display problems could be used.

3) The method of handling variables out of range

During this work the variables out of range were handled according to their priorities. The top event after scanning the measured variables was the variable with the highest priority. This is certainly not the only way of handling the problem and may require some modifications to some of the programs that deal with the definition of the top event.

7.7 Further work

It is surprising to find that so little work has been done on the use of fault trees in the Chemical Process indsutry. There is a great potential in this area that has been neglected. Generally, different aspects of this work need further development, some suggestions have been made in the previous sections of this chapter but the following may be highlighted as the most immediate ones:

1) The unit models and minitrees

The quality of the fault trees produced by the methodology is closely related to the quality of the unit models and minitrees. It is highly desirable to continue with the development of other unit models and minitrees. The failure analysis can be carried out beyond the component level. As an example consider Fig. 7.3 a minitree for the event "LEAK TO L.P. ENV. IN A PIPE". This event which was considered a basic event in this work, can be depicted as a fault event which could occur if the events "HOLE IN PIPE" and "LOW PRESSURE IN THE EXTERNAL SIDE OF THE PIPE" were present in the system under study. Other possible states of the units can also be considered i.e. for control valves some of the states could be Fully Open, Normal, Fully Closed, etc. Environmental and human effects could also be included in the minitrees. A closer look at the unit models and consideration of other possible modelling methods is also desirable.

2) Evaluation of the trees

The creation of a package which would include a method to evaluate the trees produced by the methodology could increase the use of the fault trees in the Chemical Process Industry. Several evaluation methods are already available and it is just a matter of developing the program that would put the methodology and the evaluation method together. The package could include a data bank containing the probability/failure rate data already available in the literature. Provision has already been made in the definition of the data base to include the probability data in the basic events. Reduction techniques to make more efficient evaluation of the trees could also be included in the package.

3) Use of a pilot plant

The practical use of the methodology in a pilot plant is the next stage required. The same data base and the facilities available can be used for a small system. Some simple programs to handle the analog signals coming from the pilot plant would be needed but no major changes to the programs or data base are required. If a better use of the space available is desired, some modifications could be done. This might be carried out by using another device such as a magnetic tape, to save the information related to the minitrees. A minor modification to the program that build the minitrees might also be required. The modification would imply that any time the minitrees of a specific unit were required by BTR they could be retrieved from the magnetic tape and remain in core only as long as the information is needed. Eventually if larger plant sections are required to be analysed the best solution would be to use a larger machine.

The use of a pilot plant could provide many answers to the points discussed in Section 7.6.1 and in the future could lead to the implementation of a pilot system for alarm analysis based on the synthesised fault trees.

7.8 Conclusions

A method for process fault tree synthesis has been presented in this work. The methodology developed is based on the use of unit models. These unit models are conventional dynamic models of the common process units found in chemical processes. These models are used to produce, by means of a failure analysis, unit minitrees. A new nomenclature to deal with the minitrees has been defined. The minitrees are used in the construction of fault trees for systems formed by different units. The methodology has been implemented on a PDP-11/20 digital computer. The operating system used was RSX-11M. The computer programs used to implement the methodology were developed in the RTL/2 programming language. It allowed the use of list processing techniques and bit manipulation. The synthesis of several fault trees have been performed automatically with good results. The results obtained lead to the following conclusions:

1) A systematic approach for synthesis of fault trees is possible.

2) The unit models and conventions used have proved to be useful in solving the problem of information flow through the different systems studied.

3) The two-way information flow feature, of the unit models used, allows the construction of fault trees at any part of the system under study and not only at the downstream end.

4) Multi-state variables typical of Chemical Processes can be handled by the methodology without difficulty.

5) The nomenclature used for the development of the minitrees helped to avoid the repetition of information and made easier any modification required.

6) There is a close relationship between the quality of the fault trees produced by the methodology and the quality embodied in the unit models and minitrees used.

7) The methodology described can be used in the automatic synthesis of fault trees for design and real time purposes.

8) The approach used to implement the methodology on a digital computer allows the flexibility needed so that a relatively inexperienced user may use the methodology with his own models.

9) The use of list processing techniques and bit manipulation proved to be very useful and allowed an efficient way of handling the construction of the fault trees.

Every effort has been made to find any possible "bugs" in the programs. All those programs that might present problems when the methodology is used in a pilot plant have been provided with self-explanatory messages.

It is hoped that this work will provide a better understanding of the many possible uses of fault trees in the Chemical Process industry, so that major use of this tool may be achieved with consequent improvements in plant design and operation.

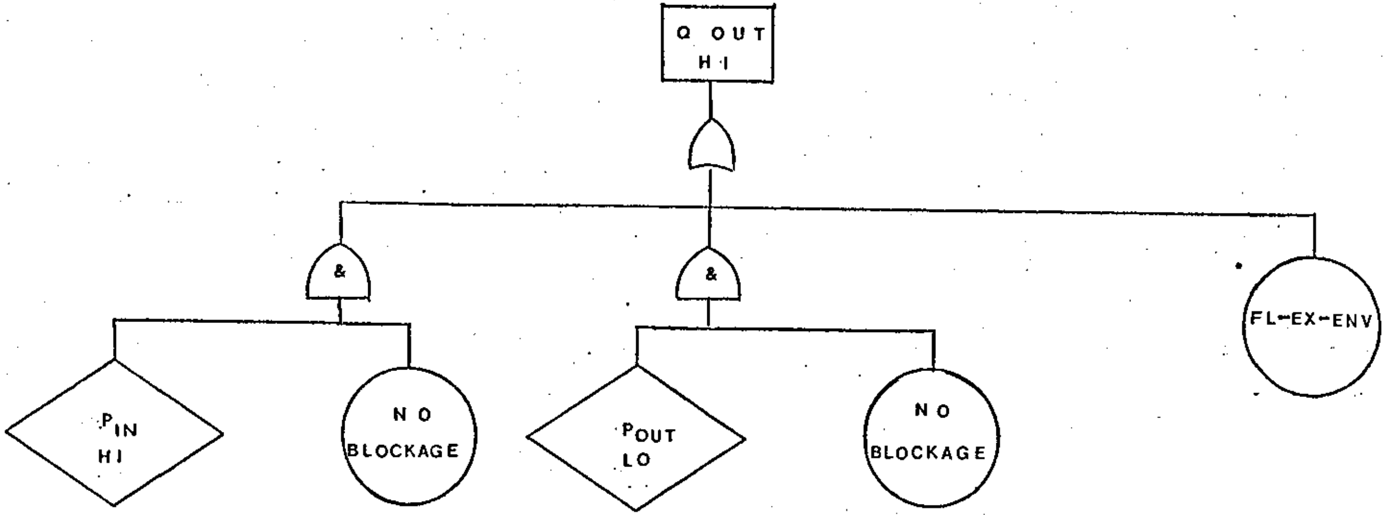


Fig. 7.1 Minitree without the Not-allowed fault "BLOCKAGE" attached to the gate

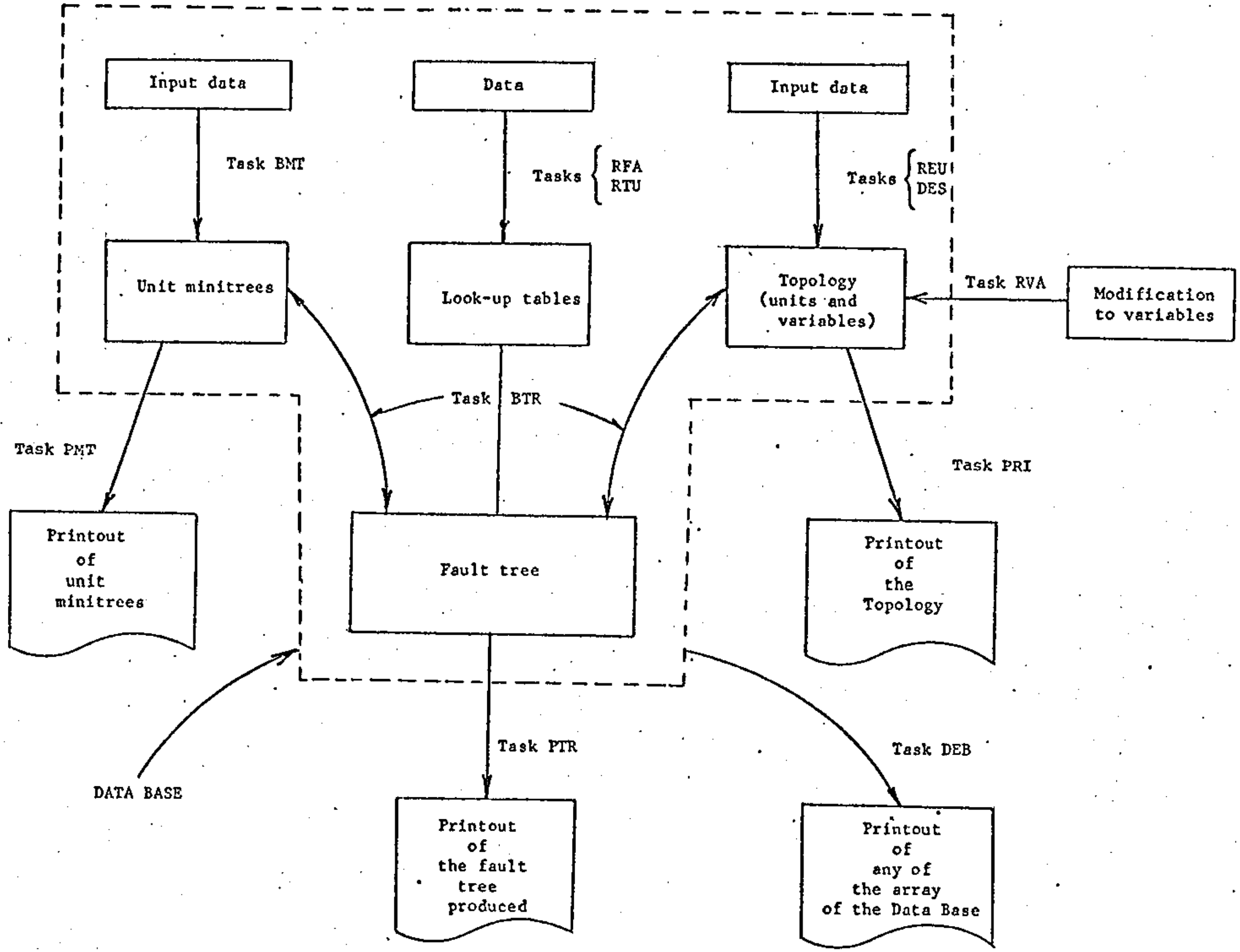


Fig. 7.2 The tasks used in this work and their relation to the Data Base

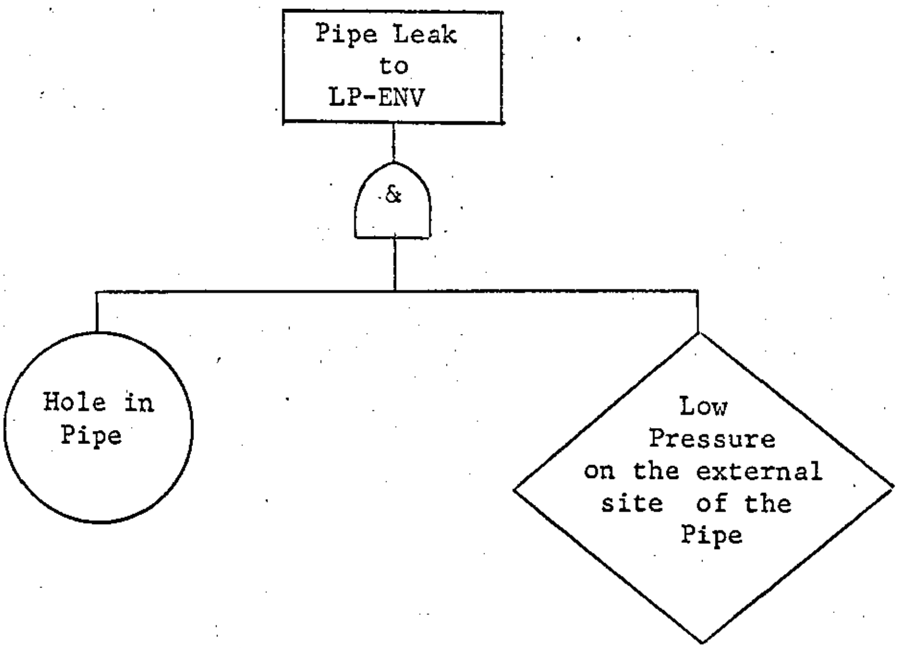


Fig. 7.3 Further development of the event leak to LP environment in pipe

REFERENCES

(A1) ANYAKORA, S.N. "Malfunction of Process Instruments and its detection using a process control computer"; Ph.D. thesis Loughborough Univ. of Technology; (1971).

(A2) ANDOW, P.K. "A Method for Process Computer Alarm Analysis"; Ph.D. thesis; Loughborough Univ. of Technology; (1973).

(A3) ANDOW, P.K. and LEES, F.P. "Process Plant Alarm Systems: General Considerations"; Proceedings of the "First International Symposium on Loss Prevention", Buschmann, C.H. (Ed.), Hague, May (1974).

(A4) ANDOW, P.K. and LEES, F.P. "Process Computer Alarm Analysis: Outline of a Method based on list Processing"; Trans. Inst. Chem. Engrs.; Vol. 53, No. 4, October (1975).

(B1) BARLOW, R.E. and PROSCHAN, F. "Importance of System Components and Fault Tree Events"; California Univ. Berkeley Operational Research Center; AD-777103, February (1974).

(B2) BARLOW, R.E. and CHATTERJEE, P. "Introduction to Fault Tree Analysis"; Operations Research Center, College of Engineering, University of California, Berkeley, ORC 73-30, December (1973).

(B3) BASS, L., WYNHOLDS, H.W. and PORTERFIELD, W.R. "Fault Tree Graphics"; Proceedings 1975, Annual Reliability and Maintainability Symposium; 1267, 75RM054; Washington D.C., 28-30 January (1975). pp.292-297.

(B4) BENNETTS, R.G. "Computer Aided Generation of Test Sequences for Logical Systems"; Ph.D. Dissertation; Univ. Southampton, Southampton, England (1972).

(B5) BARLOW, R.E., FUSSELL, J.B. and SINGPURWALLA, N.D. (Eds.). "Reliability and Fault Tree Analysis"; Society for Industrial and Applied Mathematics (SIAM); Philadelphia, Pennyslvania (1975).

(B6) BARNES, J.G.P. "RTL/2 Design and Philosophy"; Heyden and Son Ltd.; London (1976).

(C1) CROSSETTI, P.A. and BRUCE, R.A. "Commercial Application of Fault Tree Analysis"; (Douglas United Nuclear, Inc., Richland Wash.); DUN-SA-139; 7 April (1970).

(C2) CROSSETTI, P.A. "Fault Tree Analysis for System Reliability"; Instrumentation Technology; Vol. 18; August (1971) pp.52-56.

(C3) ✓CHATTERJEE, P. "Fault Tree Analysis, Min Cut Set Algorithms"; Operations Research Center, Univ. of California, Berkeley, ORC 74-2, January (1974).

(C4) ✓CACERES, S. and HENLEY, E.J. "Process Failure Analysis by Block Diagrams and Fault Trees"; Ind. Eng. Chem. Fundam, Vol. 15, No. 2, (1976).

(E1) ERICSON, C.A. "System Safety Analytical Technology Fault Tree Analysis"; Boeing Co. Seattle, Wash. Aerospace Systems Div.; February (1970); AD865618.

(F1) FUSSELL, J.B. and VESELY, W.E. "A New Methodology for Obtaining Cut Sets for Fault Trees"; Trans. Am. Nuclear Society, Vol. 15(1); (1972), pp.262-263.

(F2) FUSSELL, J.B. "Synthetic Tree Model - A Formal Methodology for Fault Tree Construction"; Aerojet Nuclear Company; Report ANCR-1098, March (1973).

(F3) FUSSELL, J.B., POWERS, G.J. and BENNETTS, R.G. "Fault Trees - A State of the Art Discussion"; IEEE Trans. on Reliability; R-23(1) (April 1974), pp.51-55.

(F4) FEUTZ, R.J. "System Safety Symposium"; June 8-9 (1965); Seattle Wash.; The Boeing Co.

(F5) FUSSELL, J.B. "A Formal Methodology for Fault Tree Construction"; Nuclear Science and Engineering; Vol. 52 (1973), pp.421-432.

(F6) FORSYTHE, A.I., KEENAN, T.A., ORGANICK, E.I. and STENBERG, W. "Computer Science"; John Wiley and Sons Inc., 2nd Edition (1975).

(F7) ✓FOSTER, J.M. "List Processing"; MacDonald; London and American Elsevier Inc; New York; 6th Impression; (1970).

(F8) ✓FUSSELL, J.B. "Fault Trees Analysis - Concepts and Techniques"; NATO Advanced Study Institute on Generic Techniques of System Reliability Assessment; Liverpool, England, (1973).

(F9) FRANKS, R.G.E. "Modelling and Simulation in Chemical Engineering"; Wiley interscience; (1972).

(H1) ✓HAASL, D.F. "Advanced Concepts in Fault-Tree Analysis"; System Safety Symposium, 1965, June 8-9; Seattle Wash., The Boeing Co.

(H2) HOLLO, E. and TAYLOR, J.R. "Algorithms and Programs for Consequence Diagram and Fault Tree Construction, RISÖ-M-1907; December (1976).

(K1) KAY, P.C.M. and HEYWOOD, P.W. "Alarm Analysis and Indication at Oldbury Nuclear Power Station"; IEE Conf. Publication 16 Part 1 (1966).

(K2) KLETZ, T.A. "Specifying and Designing Protective Systems", Loss Prevention No. 6, Chem. Eng. Prog., Technical Manual (1972), p.15.

(K3) KOEN, B.V. and CARNINO, A. "Reliability Calculations with a List Processing Technique"; IEEE Transactions on Reliability; Vol. R-23, No. 1, April (1974), pp.43-50.

(L1) LAMBERT, H.E. "Fault Trees for Decision Making in System Analysis"; California Univ; Livermore (USA). Lawrence Livermore Lab. 9 October (1975), UCRL-51829.

(L2) LAWLEY, H.G. "Operability Studies and Hazard Analysis"; Chem. Eng. Progress (Vol. 70, No. 4); April 1975, pp.45-56.

(L3) LAMBERT, H.E. "Fault Trees for Locating Sensors in Process Systems"; CEP; August (1977), pp.81-85.

(L4) LAPP, S.A. and POWERS, G.J. "Computer-Aided Synthesis of Fault Trees"; IEEE Transactions on Reliability; April (1977), pp.2-13.

(L5) LAPP, S.A. and POWERS, G.J. "Computer-assisted generation and analysis of fault trees"; 2nd International Symposium on Loss Prevention and Safety Promotion in the Process Industries; Heidelberg, Federal Republic of Germany; 6-9 September (1977), pp.377-384.

(M1) MEARNS, A.B. "Fault Tree Analysis: The Study of Unlikely Events in Complex Systems"; System Safety Symposium, June 8-9, (1975); Seattle Wash.; The Boeing Co.

(M2) MARTIN-SOLIS, G.A., ANDOW, P.K., LEES, F.P. "An approach to fault tree synthesis for process plants"; 2nd International Symposium on Loss Prevention and Safety Promotion in the Process Industries; Heidelberg, Federal Republic of Germany, 6-9 September (1977), pp.367-376.

(N1) NEOGY, R. "Fault Trees in Ocean Systems"; Proceedings 1975 Annual Reliability and Maintainability Symposium; 1265, Washington D.C. 28-30 January (1975), pp.280-285.

(N2) NIELSEN, D.S. "Use of Cause-Consequence Charts in Practical Systems Analysis"; Conference on Reliability and Fault Trees Analysis; Operations Research Center; Berkeley, California, September (1974).

(P1) POWERS, G.J. and TOMPKINS, F.C. Jr. "A Synthesis Strategy for Fault Trees in Chemical Processing Systems"; Loss Prevention, 8, CEP Technical Manual, AICHE, New York, (1974).

(P2) POWERS, G.J. and TOMPKINS, F.C. "Fault Tree Synthesis for Chemical Processes"; AICHE Journal, Vol. 20, March (1974), pp.376-387.

(P3) PANDE, P.K., SPECTOR, M.E., CHATTERJEE, P. "Computerized Fault Tree Analysis: Treel and Micsup"; California Univ. Berkeley Operations Research Center; April 1975, AD-A01046.

(P4) POWERS, G.J. and LAPP, S.A. "Computer-Aided Fault Tree Snythesis"; CEP, April 1976, pp.89-93.

(P5) POWERS, G.J., TOMPKINS, F.C. and LAPP, S.A. "A Safety Simulation Language for Chemical Processes: A procedure for Fault Tree Synthesis"; Reliability and Fault Tree Analysis; SIAM, Philadelphia (1975), pp.57-75.

(R1) RUDD, D.F., POWERS, G.J. and SIIROLA, J.J. "Process Synthesis"; Prentice Hall, Inc.; New Jersey (1973).

(R2) REACTOR SAFETY STUDY "An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants"; WASH-1400 (NUREG-75/014); U.S. Nuclear Regulatory Commission, Washington D.C.; October (1975).

(R3) RASMUSSEN, J.C. "The AEC Study on the Estimation of Risks to the public from Potential Accidents in Nuclear Power Plants"; Nuclear Safety, Vol. 15(14), August (1974), pp.375-383.

(R4) RUDD, D.F. "The Synthesis of System Designs: Elementary Decomposition Theory"; AICAE Journal; Vol. 14, No. 2, March 1968, pp.343-349.

(R5) RIVAS, J.R. "Computer-Aided Disaster Interception"; Ph.D. thesis, Univ. of Wisconsin (1973).

(S1) SALEM, S.L., APOSTOLAKIS, G.E. and OKRENT, D. "On the Automatic Construction of Fault Trees"; TRANS ANS. Vol. 22, November 1975, p.475.

(S2) SALEM, S.L., APOSTOLAKIS, G.E. and OKRENT, D. "A Computer-Oriented Approach to Fault Tree Construction"; Energy and Kinetics Department, School of Engineering and Applied Science. Univ. of California, Los Angeles, Calif. 90024; November 1976; EPRI NP-288.

(S3) SEMANDERES, S.N. "ELRAFT A Computer Program for the Efficient Logic Reduction Analysis of Fault Trees"; IEEE Trans. on Nuclear Science; Vol. 18; February (1971), pp.481-487.

(S4) STEWART, R.M. "The Design and Operation of High Integrity Protective Systems"; The Chemical Engineer; October (1974), pp.622-626.

(T1) TOMPKINS, F.C. "A Methodology for Failure Analysis in Chemical Processing"; Ph.D. thesis; Massachusetts Institute of Technology, Cambridge, Mass.; August (1974).

(T2) TAYLOR, J.R. "A Normalisation of Failure Mode Analysis of Control Systems"; Danish Atomic Energy Commission, RISÖ-M-1654.

(T3) THAKUR, R. and MISRA, K.B. "Development of Fault Tree for Reliability Studies of a Data Processing System"; International Journal of Systems Science; 8(7); (1977); pp.771-780.

(T4) TAYLOR, J.R. "A Semiautomatic Method for Qualitative Failure Mode Analysis"; CSNI Specialist Meeting on: The Development and Application of Reliability Techniques to Nuclear Plant; Liverpool; April (1974); pp.8-10.

(V1) VESELY, W.E. "Analysis of Fault Trees by Kinetic Tree Theory"; (Idaho Nuclear Corp. Idaho Falls); IN-1330; October (1969).

(V2) VESELY, W.E. and NARUM, R.E. "PREP and KITT Computer Codes for the Automatic Evaluation of a Fault Tree"; Idaho Nuclear Corporation, Idaho Falls, Idaho IN-1349; (1970).

(V3) VESELY, W.E. "A time-dependent Methodology for Fault Tree Evaluation"; Nuclear Eng. and Design; 13 (1970); North-Holland Publishing Company, pp.337-360.

(W1) WONG, P.Y. "Fautran A Fault Tree Analyser"; Atomic Energy of Canada Ltd.; Chalk River Nuclear Laboratories, Chalk River, Ontario; AECL-5182; August (1975).

(W2) WHEELER, D.B., HSUAN, J.S., DUERSCH, R.R. and ROE, G.M. "Fault Tree Analysis using bit manipulation"; IEEE Transactions on Reliability; Vol. R-26, No. 2; June (1977); pp.95-99.

(W3) ✓ WELBOURNE, D. "Data Processing and Control by a Computer at Wylfa Nuclear Power Station", in "Advances in Automatic Control", I.Mech.E., (1965).

(W4) ✓ WELBOURNE, D. "Alarm Analysis and Display at Wylfa Nuclear Power Station"; Proc. I.E.E. 115(11); (1968).

(W5) WOODWARD, P.M. and BOND, S.G. "ALGOL-68 Users guide"; Royal Radar Establishment, Ministry of Defence, Malvern, Worcs. (UK), 2nd Ed. (1974).

APPENDIX I

UNIT MODELS AND UNIT MINITREES

Table of Contents

Section Page

Tables

Figures

Appendix I

Unit Models and Unit Minitrees

I.1 Introduction

The unit models described in this appendix are intended to be representative of the class of models, that may be conveniently used in the failure analysis, required to produce the unit minitrees.

The nomenclature which follows defines the characters used to denote common quantities. Any exceptions to the nomenclature will be defined in the model concerned. All the unit model streams are based on mass or volume units, except where the opposite is stated in the unit model. In order to comply with the conventions stated in Chapter 3 of this work all stream properties but pressure are set in the unit outputs. Pressure is normally set in the unit input streams. The conventions may be relaxed for those cases in which the strict use of them may lead to unnecessarily complicated models. In those cases, where the conventions are relaxed it is noted in the model.

High gain differential equations are used in the unit models to set intermediate stream pressures. These type of equations help to preserve continuity in process streams passing through several units. The need for this type of equation is discussed by Franks.(F9) The use of high-gain equations with the conventions used in this work and cases when the convention is relaxed have been discussed by Andow.(A2)

The unit minitrees that are described after each unit model, are the ones used in the various examples considered in this work. They were obtained by means of the derivation method discussed in Chapter 3; the derivation of the minitrees was carried out manually. The auxiliary data files that were used to implement the minitrees in the digital computer are also shown for each unit. Task PMT was used to produce the printouts.

A key to the names of the unit models used and to the faults of the minitrees is shown in Tables I.2 and I.3. Note that the name of the faults is restricted to ten characters.

I.2 Unit Models and Minitrees

The unit models and the minitrees used in this work are the following:

I.2.1 Centrifugal Pump

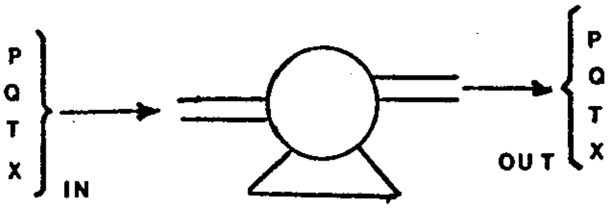


Fig. I.2.1 Centrifugal pump

Equations Used	Assumptions
1) $\frac{dP_{IN}}{dt} = G(Q_{IN}-Q_{OUT})$	1) Isothermal Operation
2) $Q_{OUT} = \left(k_1 Q_{IN}^{\frac{2}{3}} + k_2(P_{IN}-P_{OUT}) + k_3 Q_{IN}^2\right)^{\frac{1}{2}}$	2) Constant rotational speed of impeller
3) $X_{OUT} = X_{IN}$	
4) $T_{OUT} = T_{IN}$	
Name used to identify the unit: CENT-PUMP	

```
*********************************************************************

                         ************
                         * MINITREES *
                         *    FOR    *
                         * CENT-PUMP *
                         ************

              VARIABLE                                     B.C. & NOT
             DESCRIPTION        FAULT        GATE        ALLOWED FAULTS
             -----------        -----        ----        ------------

     M-EVENT :     Q     OUT      HI           OR          LO
                                                           CLOSED
                                                           BLOCKAGE
                                                           SHUT

     T-EVENT :     P     IN       HI
     T-EVENT :     P     OUT      LO
     T-EVENT :     Q     IN       HI
     B-EVENT :                    FL-EX-ENV

              VARIABLE                                     B.C. & NOT
             DESCRIPTION        FAULT        GATE        ALLOWED FAULTS
             -----------        -----        ----        ------------

     M-EVENT :     Q     OUT      LO           OR          HI

     T-EVENT :     P     IN       LO
     T-EVENT :     P     OUT      HI
     T-EVENT :     Q     IN       LO
     B-EVENT :                    BLOCKAGE
     B-EVENT :                    LK-LP-ENV

              VARIABLE                                     B.C. & NOT
             DESCRIPTION        FAULT        GATE        ALLOWED FAULTS
             -----------        -----        ----        ------------

     M-EVENT :     P     IN       HI           OR          LO
                                                           LK-LP-ENV

     T-EVENT :     Q     IN       HI
     T-EVENT :     Q     OUT      LO
     B-EVENT :                    BLOCKAGE
     B-EVENT :                    LK-HP-ENV

              VARIABLE                                     B.C. & NOT
             DESCRIPTION        FAULT        GATE        ALLOWED FAULTS
             -----------        -----        ----        ------------

     M-EVENT :     P     IN       LO           OR          HI
                                                           FL-EX-ENV

     T-EVENT :     Q     IN       LO
     T-EVENT :     Q     OUT      HI
     B-EVENT :                    LK-LP-ENV

              VARIABLE                                     B.C. & NOT
             DESCRIPTION        FAULT        GATE        ALLOWED FAULTS
             -----------        -----        ----        ------------
```

```
M-EVENT :     T     OUT       HI               OR        LO

T-EVENT :     T     IN        HI
B-EVENT :                     EXT-FIRE

           VARIABLE                                       B.C. & NOT
          DESCRIPTION         FAULT            GATE      ALLOWED FAULTS
          -----------         -----            ----      ------------

M-EVENT :     T     OUT       LO               OR        HI

T-EVENT :     T     IN        LO

           VARIABLE                                       B.C. & NOT
          DESCRIPTION         FAULT            GATE      ALLOWED FAULTS
          -----------         -----            ----      ------------

M-EVENT :     X     OUT       HI               OR        LO

T-EVENT :     X     IN        HI

           VARIABLE                                       B.C. & NOT
          DESCRIPTION         FAULT            GATE      ALLOWED FAULTS
          -----------         -----            ----      ------------

M-EVENT :     X     OUT       LO               OR        HI

T-EVENT :     X     IN        LO

           VARIABLE                                       B.C. & NOT
          DESCRIPTION         FAULT            GATE      ALLOWED FAULTS
          -----------         -----            ----      ------------

M-EVENT :     Q     OUT       NO-FLOW          OR

T-EVENT :     Q     IN        NO-FLOW
B-EVENT :                     SHUTDOWN
B-EVENT :                     COMP-BLOC

           VARIABLE                                       B.C. & NOT
          DESCRIPTION         FAULT            GATE      ALLOWED FAULTS
          -----------         -----            ----      ------------

M-EVENT :     Q     OUT       GT0              OR

T-EVENT :     Q     IN        GT2

*****************************************************************
```

```
CENT-PUMP%
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
F
1
HI%
T
F
0
LO%
T
Q
1
HI%
B

2
FL-EX-ENV%
0.0001
M
Q
0
LO%
0
*0*%
T
F
1
LO%
T
F
0
HI%
T
Q
1
LO%
B

2
BLOCKAGE%
0.0002
B

2
LK-LP-ENV%
0.0003
M
P
1
HI%
0
LK-LP-ENV%
```

```
*0*%
T
Q
1
HI%
T
Q
0
LO%
B

2
BLOCKAGE%
0.0224
B

2
LK-HP-ENV%
0.0225
M
F
1
LO%
0
FL-EX-ENV%
*0*%
T
Q
1
LO%
T
Q
0
HI%
B

2
LK-LF-ENV%
0.0226
M
T
0
HI%
0
*0*%
T
T
1
HI%
B

2
EXT-FIRE%
0.0221
M
T
0
LO%
0
*0*%
T
```

```
T
1
LO%
M
X
0
HI%
0
*0*%
T
X
1
HI%
M
X
0
LO%
0
*0*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*0*%
T
Q
1
NO-FLOW%
B

2
SHUTDOWN%
0.0023
B

2
COMP-BLOC%
0.0024
M
Q
0
GT0%
0
*0*%
T
Q
1
GT0%
*
*+*+*%
```

I.2.2 Closed Tank

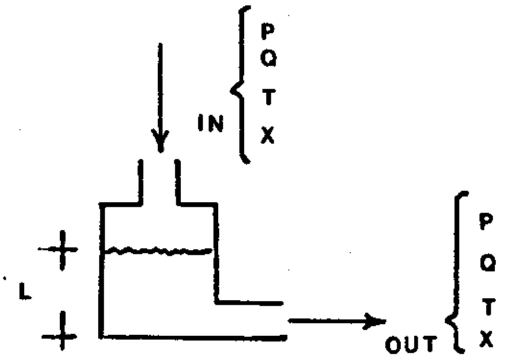


Fig. I.2.2 Closed Tank

Equations Used	Assumptions
1) $\frac{dL}{dt} = (Q_{IN}-Q_{OUT})/A_{Tank}$	1) Closed Vessel
2) $P_{IN} = \frac{kT}{V_{Tank} - \text{Area } L}$	2) Perfectly Mixed
3) $Q_{OUT} = k_2 \left(P_{IN}+k_L L-P_{OUT}\right)^{\frac{1}{2}}$	3) No heat loss
4) $T_{IN} = T_{OUT}$	4) No volume changes
5) $X_{IN} = X_{OUT}$	5) No phase changes
	6) Ideal gas behaviour
Name used to identify the unit: CLOSED-TK	

```
*******************************************************************************

                              *************
                              * MINITREES *
                              *    FOR    *
                              * CLOSED-TK *
                              *************

                 VARIABLE                                          B.C. & NOT
                DESCRIPTION          FAULT          GATE         ALLOWED FAULTS
                -----------          -----          ----         -------------

     M-EVENT :     L                 HI             OR           LO
                                                                 LK-LF-ENV

     T-EVENT :     Q      IN         HI
     T-EVENT :     Q      OUT        LO
     B-EVENT :                       BLOCKAGE
     B-EVENT :                       LK-HP-ENV

                 VARIABLE                                          B.C. & NOT
                DESCRIPTION          FAULT          GATE         ALLOWED FAULTS
                -----------          -----          ----         -------------

     M-EVENT :     L                 LO             OR           HI
                                                                 FL-EX-ENV

     T-EVENT :     Q      IN         LO
     T-EVENT :     Q      OUT        HI
     B-EVENT :                       LK-LF-ENV
     B-EVENT :                       BLOCKAGE

                 VARIABLE                                          B.C. & NOT
                DESCRIPTION          FAULT          GATE         ALLOWED FAULTS
                -----------          -----          ----         -------------

     M-EVENT :     F      IN         HI             OR           LO
                                                                 LK-LP-ENV

     T-EVENT :     L                 HI
     B-EVENT :                       LK-HP-ENV

                 VARIABLE                                          B.C. & NOT
                DESCRIPTION          FAULT          GATE         ALLOWED FAULTS
                -----------          -----          ----         -------------

     M-EVENT :     P      IN         LO             OR           HI
                                                                 FL-EX-ENV

     T-EVENT :     L                 LO
     B-EVENT :                       LK-LF-ENV

                 VARIABLE                                          B.C. & NOT
                DESCRIPTION          FAULT          GATE         ALLOWED FAULTS
                -----------          -----          ----         -------------

     M-EVENT :     Q      OUT        HI             OR           LO
                                                                 CLOSED
                                                                 BLOCKAGE
                                                                 SHUT
```

```
T-EVENT :    P     IN        HI
T-EVENT :    P     OUT       LO
B-EVENT :                    FL-EX-ENV

          VARIABLE                                  B.C. & NOT
         DESCRIPTION         FAULT       GATE     ALLOWED FAULTS
         -----------         -----       ----     -------------

M-EVENT :    Q     OUT       LO          OR       HI

T-EVENT :    P     IN        LO
T-EVENT :    P     OUT       HI
B-EVENT :                    LK-LP-ENV
B-EVENT :                    BLOCKAGE

          VARIABLE                                  B.C. & NOT
         DESCRIPTION         FAULT       GATE     ALLOWED FAULTS
         -----------         -----       ----     -------------

M-EVENT :    T     OUT       HI          OR       LO

T-EVENT :    T     IN        HI
B-EVENT :                    EXT-FIRE

          VARIABLE                                  B.C. & NOT
         DESCRIPTION         FAULT       GATE     ALLOWED FAULTS
         -----------         -----       ----     -------------

M-EVENT :    T     OUT       LO          OR       HI

T-EVENT :    T     IN        LO

          VARIABLE                                  B.C. & NOT
         DESCRIPTION         FAULT       GATE     ALLOWED FAULTS
         -----------         -----       ----     -------------

M-EVENT :    X     OUT       HI          OR       LO

T-EVENT :    X     IN        HI

          VARIABLE                                  B.C. & NOT
         DESCRIPTION         FAULT       GATE     ALLOWED FAULTS
         -----------         -----       ----     -------------

M-EVENT :    X     OUT       LO          OR       HI

T-EVENT :    X     IN        LO

          VARIABLE                                  B.C. & NOT
         DESCRIPTION         FAULT       GATE     ALLOWED FAULTS
         -----------         -----       ----     -------------

M-EVENT :    Q     OUT       NO-FLOW     OR

T-EVENT :    Q     IN        NO-FLOW
B-EVENT :                    COMP-BLOC

          VARIABLE                                  B.C. & NOT
         DESCRIPTION         FAULT       GATE     ALLOWED FAULTS
         -----------         -----       ----     -------------
```

```
     M-EVENT :     Q   OUT      GT0           OR

     T-EVENT :     Q   IN       GT0

**********************************************************************
```

```
CLOSED-TK%
M
L
3
HI%
0
LK-LF-ENV%
*0*%
T
Q
1
HI%
T
Q
0
LO%
B

2
BLOCKAGE%
0.0221
B

2
LK-HP-ENV%
0.0222
M
L
3
LO%
0
FL-EX-ENV%
*0*%
T
Q
1
LO%
T
Q
0
HI%
B

2
LK-LF-ENV%
0.0223
B

2
BLOCKAGE%
0.024
M
F
1
HI%
0
LK-LF-ENV%
*0*%
T
L
3
```

```
HI%
B

2
LK-HP-ENV%
0.0025
M
F
1
LO%
0
FL-EX-ENV%
*0*%
T
L
3
LO%
B

2
LK-LF-ENV%
0.0026
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
F
1
HI%
T
P
0
LO%
B

2
FL-EX-ENV%
0.0027
M
W
0
LO%
0
*0*%
T
F
1
LO%
T
F
0
HI%
B

2


PAGE                    CLTK   .DAT
-------
```

```
LK-LF-ENV%
0.0028
B

2
BLOCKAGE%
0.0029
M
T
0
HI%
0
*0*%
T
T
1
HI%
B

2
EXT-FIRE%
0.0021
M
T
0
LO%
0
*0*%
T
T
1
LO%
M
X
0
HI%
0
*0*%
T
X
1
HI%
M
X
0
LO%
0
*0*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*0*%
T
Q
1
```

```
NO-FLOW%
B

2
COMP-BLOC%
0.0001
M
W
0
GT0%
0
*0*%
T
W
1
GT0%
*
*+*+*%
```

I.2.3 Control Valves

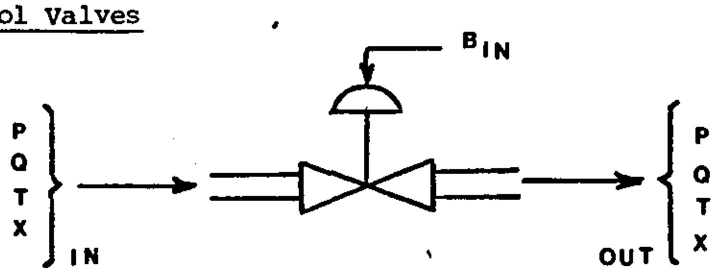


Fig. I.2.3 Control Valve

<table>
<tr><th>Equations Used</th><th>Assumptions</th></tr>
<tr><td rowspan="3">1) $\frac{dP_{IN}}{dt} = G(Q_{IN}-Q_{OUT})$
2) $Q_{OUT} = k_{Valve}f(B_{IN})(P_{IN}-P_{OUT})^{\frac{1}{2}}$
3) $T_{OUT} = T_{IN}$
4) $X_{OUT} = X_{IN}$</td><td>1) Air to open
2) Isothermal flow through sliding-steam valve
3) Contrary to the conventions of Chapter 3 B_{IN} (Pressure) is assumed to be fixed at the controller.</td></tr>
<tr><td>Special Nomenclature</td></tr>
<tr><td>B = Pressure</td></tr>
<tr><td colspan="2">Name used to identify unit: CNTRL-VAL</td></tr>
<tr><td colspan="2">Special Case: CNTRLV-SC</td></tr>
<tr><td colspan="2">Note: The special case considered in this work refers to a valve that will remain open all the time, unless a signal is sent to close the valve. Therefore it is not like the general use valve that can open and close according to the value of B_{IN}.</td></tr>
</table>

```
*******************************************************************************

                              ************
                              * MINITREES *
                              *    FOR    *
                              * CNTRL-VAL *
                              ************

                VARIABLE                                          B.C. & NOT
               DESCRIPTION         FAULT          GATE        ALLOWED FAULTS
               -----------         -----          ----        ------------

    M-EVENT :      P     IN         HI             OR          LO
                                                               LK-LP-ENV

    T-EVENT :      Q     IN         HI
    T-EVENT :      Q     OUT        LO
    B-EVENT :                       BLOCKAGE
    B-EVENT :                       SHUT

                VARIABLE                                          B.C. & NOT
               DESCRIPTION         FAULT          GATE        ALLOWED FAULTS
               -----------         -----          ----        ------------

    M-EVENT :      P     IN         LO             OR          HI
                                                               LK-HP-ENV

    T-EVENT :      Q     IN         LO
    T-EVENT :      Q     OUT        HI
    B-EVENT :                       LK-LP-ENV

                VARIABLE                                          B.C. & NOT
               DESCRIPTION         FAULT          GATE        ALLOWED FAULTS
               -----------         -----          ----        ------------

    M-EVENT :      Q     OUT        HI             OR          LO
                                                               CLOSED
                                                               BLOCKAGE
                                                               SHUT

    R-EVENT :                       A
    T-EVENT :      P     OUT        LO
    T-EVENT :      B     IN         HI
    B-EVENT :                       FAIL-OPEN
    B-EVENT :                       LK-HP-ENV

                VARIABLE                                          B.C. & NOT
               DESCRIPTION         FAULT          GATE        ALLOWED FAULTS
               -----------         -----          ----        ------------

    M-EVENT :      Q     OUT        LO             OR          HI
                                                               OPEN
                                                               WIDE-OPEN
                                                               FAIL-OPEN

    R-EVENT :                       D
    T-EVENT :      P     OUT        HI
    T-EVENT :      B     IN         LO
    B-EVENT :                       BLOCKAGE
    B-EVENT :                       LK-LP-ENV
```

```
B-EVENT :                           FAI-CLOSE

               VARIABLE                                           B.C. & NOT
              DESCRIPTION           FAULT         GATE        ALLOWED FAULTS
              -----------           -----         ----        -------------

M-EVENT :                           A             AND

T-EVENT :        P     IN           HI
R-EVENT :                           C

               VARIABLE                                           B.C. & NOT
              DESCRIPTION           FAULT         GATE        ALLOWED FAULTS
              -----------           -----         ----        -------------

M-EVENT :                           C             OR

T-EVENT :        B     IN           NO-CHANGE
B-EVENT :                           VALV-STCK
B-EVENT :                           MANUAL

               VARIABLE                                           B.C. & NOT
              DESCRIPTION           FAULT         GATE        ALLOWED FAULTS
              -----------           -----         ----        -------------

M-EVENT :                           D             AND

T-EVENT :        P     IN           LO
R-EVENT :                           C

               VARIABLE                                           B.C. & NOT
              DESCRIPTION           FAULT         GATE        ALLOWED FAULTS
              -----------           -----         ----        -------------

M-EVENT :        T     OUT          HI            OR          LO

T-EVENT :        T     IN           HI
B-EVENT :                           EXT-FIRE

               VARIABLE                                           B.C. & NOT
              DESCRIPTION           FAULT         GATE        ALLOWED FAULTS
              -----------           -----         ----        -------------

M-EVENT :        T     OUT          LO            OR          HI

T-EVENT :        T     IN           LO

               VARIABLE                                           B.C. & NOT
              DESCRIPTION           FAULT         GATE        ALLOWED FAULTS
              -----------           -----         ----        -------------

M-EVENT :        X     OUT          HI            OR          LO

T-EVENT :        X     IN           HI

               VARIABLE                                           B.C. & NOT
              DESCRIPTION           FAULT         GATE        ALLOWED FAULTS
              -----------           -----         ----        -------------

M-EVENT :        X     OUT          LO            OR          HI
```

```
T-EVENT :     X      IN          LO

                VARIABLE                                     B.C. & NOT
              DESCRIPTION        FAULT         GATE       ALLOWED FAULTS
              -----------        -----         ----       -------------

M-EVENT :     Q      OUT         NO-FLOW       OR

T-EVENT :     Q      IN          NO-FLOW
B-EVENT :                        COMP-BLOC
B-EVENT :                        SHUT

                VARIABLE                                     B.C. & NOT
              DESCRIPTION        FAULT         GATE       ALLOWED FAULTS
              -----------        -----         ----       -------------

M-EVENT :     Q      OUT         GT0           OR

T-EVENT :     Q      IN          GT0

*************************************************************************
```

```
CNTRL-VAL%
M
P
1
HI%
0
LK-LP-ENV%
*0*%
T
Q
1
HI%
T
Q
0
LO%
B

2
BLOCKAGE%
0.0001
B

2
SHUT%
0.0002
M
P
1
LO%
0
LK-HP-ENV%
*0*%
T
Q
1
LO%
T
Q
0
HI%
B

2
LK-LP-ENV%
0.0003
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
R

2
A%
T
P
```

```
0
LO%
T
B
1
HI%
B

2
FAIL-OPEN%
0.0024
B

2
LK-HF-ENV%
0.0205
M
Q
0
LO%
2
OPEN%
WIDE-OPEN%
FAIL-OPEN%
*B*%
R

2
D%
T
P
0
HI%
T
B
1
LO%
B

2
BLOCKAGE%
0.0006
B

2
LK-LF-ENV%
0.0007
B

2
FAI-CLOSE%
0.0005
M

2
A%
1
*B*%
T
F
1
```

```
HI%
R

2
C%
M

2
C%
0
*@*%
T
B
1
NO-CHANGE%
B

2
VALV-STCK%
0.0009
B

2
MANUAL%
0.0010
M

2
D%
1
*@*%
T
P
1
LO%
R

2
C%
M
T
0
HI%
0
*@*%
T
T
1
HI%
B

2
EXT-FIRE%
0.0001
M
T
0
LO%
0
*@*%
T
```

```
T
1
LO%
M
X
0
HI%
0
*@*%
T
X
1
AI%
M
X
0
LO%
0
*@*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*@*%
T
Q
1
NO-FLOW%
B

2
COMP-BLOC%
0.0221
B

2
SHUT%
0.0221
M
Q
0
GT2%
0
*@*%
T
Q
1
GT2%
*
*+*+*%
```

```
*******************************************************************************

                              *************
                              * MINITREES *
                              *    FOR    *
                              * CNTRLV-SC *
                              *************

               VARIABLE                                   B.C. & NOT
             DESCRIPTION         FAULT          GATE     ALLOWED FAULTS
             -----------         -----          ----     ------------

     M-EVENT :    P    IN        HI             OR       LO
                                                         LK-LP-ENV

     T-EVENT :    Q    IN        HI
     T-EVENT :    Q    OUT       LO
     B-EVENT :                   BLOCKAGE
     B-EVENT :                   SHUT

               VARIABLE                                   B.C. & NOT
             DESCRIPTION         FAULT          GATE     ALLOWED FAULTS
             -----------         -----          ----     ------------

     M-EVENT :    P    IN        LO             OR       HI
                                                         LK-HP-ENV

     T-EVENT :    Q    IN        LO
     T-EVENT :    Q    OUT       HI
     B-EVENT :                   LK-LP-ENV

               VARIABLE                                   B.C. & NOT
             DESCRIPTION         FAULT          GATE     ALLOWED FAULTS
             -----------         -----          ----     ------------

     M-EVENT :    Q    OUT       HI             OR       LO
                                                         CLOSED
                                                         BLOCKAGE
                                                         SHUT

     T-EVENT :    P    IN        HI
     T-EVENT :    P    OUT       LO
     B-EVENT :                   LK-HP-ENV

               VARIABLE                                   B.C. & NOT
             DESCRIPTION         FAULT          GATE     ALLOWED FAULTS
             -----------         -----          ----     ------------

     M-EVENT :    Q    OUT       LO             OR       HI
                                                         OPEN
                                                         WIDE-OPEN
                                                         FAIL-OPEN

     T-EVENT :    P    IN        LO
     T-EVENT :    P    OUT       HI
     T-EVENT :    B    IN        LO
     B-EVENT :                   BLOCKAGE
     B-EVENT :                   LK-LP-ENV
     B-EVENT :                   FAI-CLOSE
     B-EVENT :                   VALV-STCK
```

```
B-EVENT :                     MANUAL

           VARIABLE                                     B.C. & NOT
          DESCRIPTION         FAULT         GATE       ALLOWED FAULTS
          -----------         -----         ----       ------------

M-EVENT :     T     OUT       HI            OR         LO

T-EVENT :     T     IN        HI
B-EVENT :                     EXT-FIRE

           VARIABLE                                     B.C. & NOT
          DESCRIPTION         FAULT         GATE       ALLOWED FAULTS
          -----------         -----         ----       ------------

M-EVENT :     T     OUT       LO            OR         HI

T-EVENT :     T     IN        LO

           VARIABLE                                     B.C. & NOT
          DESCRIPTION         FAULT         GATE       ALLOWED FAULTS
          -----------         -----         ----       ------------

M-EVENT :     X     OUT       HI            OR         LO

T-EVENT :     X     IN        HI

           VARIABLE                                     B.C. & NOT
          DESCRIPTION         FAULT         GATE       ALLOWED FAULTS
          -----------         -----         ----       ------------

M-EVENT :     X     OUT       LO            OR         HI

T-EVENT :     X     IN        LO

           VARIABLE                                     B.C. & NOT
          DESCRIPTION         FAULT         GATE       ALLOWED FAULTS
          -----------         -----         ----       ------------

M-EVENT :     Q     OUT       NO-FLOW       OR

T-EVENT :     Q     IN        NO-FLOW
B-EVENT :                     SHUT
B-EVENT :                     COMP-BLOC

           VARIABLE                                     B.C. & NOT
          DESCRIPTION         FAULT         GATE       ALLOWED FAULTS
          -----------         -----         ----       ------------

M-EVENT :     Q     OUT       GT0           AND

T-EVENT :     Q     IN        GT0
T-EVENT :     B     IN        NO-SIGNAL

****************************************************************************
```

```
CNTRLV-SC%
M
P
1
HI%
0
LK-LP-ENV%
*0*%
T
Q
1
HI%
T
Q
0
LO%
B

2
BLOCKAGE%
0.0001
B

2
SHUT%
0.0002
M
P
1
LO%
0
LK-HP-ENV%
*0*%
T
Q
1
LO%
T
Q
0
HI%
B

2
LK-LP-ENV%
0.0003
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
P
1
HI%
T
P
```

```
0
LO%
B

2
LK-HP-ENV%
0.0025
M
Q
0
LO%
0
OPEN%
WIDE-OPEN%
FAIL-OPEN%
*0*%
T
F
1
LO%
T
F
0
HI%
T
B
1
LO%
B

2
BLOCKAGE%
0.0006
B

2
LK-LF-ENV%
0.0007
B

2
FAI-CLOSE%
0.0005
B

2
VALV-STCK%
0.0009
B

2
MANUAL%
0.0010
M
T
0
HI%
0
*0*%
T
T
```

```
1
HI%
B

2
EXT-FIRE%
0.0001
M
T
0
LO%
0
*@*%
T
T
1
LO%
M
X
0
HI%
0
*@*%
T
X
1
HI%
M
X
0
LO%
0
*@*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*@*%
T
Q
1
NO-FLOW%
B

2
SHUT%
0.0001
B

2
COMP-BLOC%
0.0001
M
Q
0
GT0%


PAGE                           CONTSP.DAT
--------
```

```
1
*@*%
T
W
1
GT0%
T
H
1
NO-SIGNAL%
*
*+*+*%
```

I.2.4 Controller

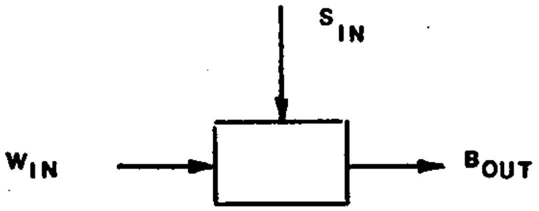


Fig. I.2.4 Controller

Equations Used	Assumptions
1) $B_{OUT} = k_c \varepsilon + P_o$ 2) $\varepsilon = W_{IN} - S_{IN}$	1) Proportional controller 2) Ideal action 3) Contrary to the conventions of Chapter 3 B_{OUT} (Pressure is specified here.
	Special Nomenclature
	B = Pressure ε = Error P_o = Steady-state output W = Setpoint S = Input signal from sensor
Name used to identify unit: CNTROLLER	

```
**********************************************************************

                             *************
                             * MINITREES *
                             *    FOR    *
                             * CNTROLLER *
                             *************

                VARIABLE                                    B.C. & NOT
               DESCRIPTION         FAULT        GATE     ALLOWED FAULTS
               -----------         -----        ----     -------------

     M-EVENT :    B     OUT        HI           OR       LO

     T-EVENT :    S     IN         LO
     T-EVENT :    W     IN         HI
     B-EVENT :                     CONT-F-HI

                VARIABLE                                    B.C. & NOT
               DESCRIPTION         FAULT        GATE     ALLOWED FAULTS
               -----------         -----        ----     -------------

     M-EVENT :    B     OUT        LO           OR       HI

     T-EVENT :    S     IN         HI
     T-EVENT :    W     IN         LO
     B-EVENT :                     CONT-F-LO

                VARIABLE                                    B.C. & NOT
               DESCRIPTION         FAULT        GATE     ALLOWED FAULTS
               -----------         -----        ----     -------------

     M-EVENT :    B     OUT        NO-CHANGE    OR

     T-EVENT :    S     IN         NO-CHANGE
     B-EVENT :                     CONT-STCK

**********************************************************************
```

```
CNTROLLER%
M
B
0
HI%
0
*@*%
T
S
1
LO%
T
W
1
HI%
B

2
CONT-F-HI%
0.0221
M
B
0
LO%
0
*@*%
T
S
1
HI%
T
W
1
LO%
B

2
CONT-F-LO%
0.0222
M
B
0
NO-CHANGE%
0
*@*%
T
S
1
NO-CHANGE%
B

2
CONT-STCK%
0.0223
*
*+*+*%
```

I.2.5 Dummy Head

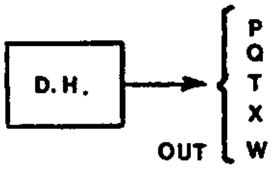


Fig. I.2.5 Dummy Head

Equations Used	Assumptions
1) $Q_{OUT} = f(P_{OUT})$ 2) $W_{OUT} = f$ (other causes) 3) $T_{OUT} = f$ (other causes) 4) $X_{OUT} = f$ (other causes)	1) No input streams 2) General purpose use **Special Nomenclature** W = set point
Name used to identify unit: DUMMY-H	

```
*********************************************************************

                         *************
                         * MINITREES *
                         *    FOR    *
                         *  DUMMY-H  *
                         *************

               VARIABLE                                     B.C. & NOT
              DESCRIPTION       FAULT        GATE       ALLOWED FAULTS
              -----------       -----        ----       -------------

     M-EVENT :    Q      OUT    HI           OR         LO
                                                        CLOSED
                                                        BLOCKAGE
                                                        SHUT

     T-EVENT :    P      OUT    LO
               VARIABLE                                     B.C. & NOT
              DESCRIPTION       FAULT        GATE       ALLOWED FAULTS
              -----------       -----        ----       -------------

     M-EVENT :    Q      OUT    LO           OR         HI

     T-EVENT :    P      OUT    HI
               VARIABLE                                     B.C. & NOT
              DESCRIPTION       FAULT        GATE       ALLOWED FAULTS
              -----------       -----        ----       -------------

     M-EVENT :    W      OUT    HI           OR         LO
                                                        OTHER-CAU

     B-EVENT :                  OTHER-CAU
               VARIABLE                                     B.C. & NOT
              DESCRIPTION       FAULT        GATE       ALLOWED FAULTS
              -----------       -----        ----       -------------

     M-EVENT :    W      OUT    LO           OR         HI
                                                        OTHER-CAU

     B-EVENT :                  OTHER-CAU
               VARIABLE                                     B.C. & NOT
              DESCRIPTION       FAULT        GATE       ALLOWED FAULTS
              -----------       -----        ----       -------------

     M-EVENT :    T      OUT    HI           OR         LO
                                                        OTHER-CAU

     B-EVENT :                  OTHER-CAU
               VARIABLE                                     B.C. & NOT
              DESCRIPTION       FAULT        GATE       ALLOWED FAULTS
              -----------       -----        ----       -------------

     M-EVENT :    T      OUT    LO           OR         HI
                                                        OTHER-CAU
```

```
B-EVENT :                     OTHER-CAU

         VARIABLE                                   B.C. & NOT
        DESCRIPTION           FAULT        GATE    ALLOWED FAULTS
        -----------           -----        ----    ------------

M-EVENT :   X      OUT        HI           OR      LO
                                                   OTHER-CAU

B-EVENT :                     OTHER-CAU

         VARIABLE                                   B.C. & NOT
        DESCRIPTION           FAULT        GATE    ALLOWED FAULTS
        -----------           -----        ----    ------------

M-EVENT :   X      OUT        LO           OR      HI
                                                   OTHER-CAU

B-EVENT :                     OTHER-CAU

         VARIABLE                                   B.C. & NOT
        DESCRIPTION           FAULT        GATE    ALLOWED FAULTS
        -----------           -----        ----    ------------

M-EVENT :   Q      OUT        NO-FLOW      OR      OTHER-CAU

B-EVENT :                     OTHER-CAU

         VARIABLE                                   B.C. & NOT
        DESCRIPTION           FAULT        GATE    ALLOWED FAULTS
        -----------           -----        ----    ------------

M-EVENT :   Q      OUT        GT0          OR      OTHER-CAU

B-EVENT :                     OTHER-CAU

***********************************************************************
```

```
DUMMY-H%
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
P
0
LO%
M
Q
0
LO%
2
*0*%
T
P
0
HI%
M
W
0
HI%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0002
M
W
0
LO%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0002
M
T
0
HI%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0221
M
```

```
T
0
LO%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0000
M
X
0
HI%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0000
M
X
0
LO%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0000
M
Q
0
NO-FLOW%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0000
M
Q
0
GT0%
0
OTHER-CAU%
*0*%
B

2
OTHER-CAU%
0.0000
*
*****%
```

I.2.6 Dummy Tail

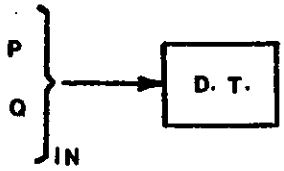


Fig. I.2.6 Dummy Tail

Equations Used	Assumptions
1) $P_{IN} = f\ (Q_{IN})$	1) No output streams 2) General purpose use
Name used to identify unit: DUMMY-T	

```
****************************************************************

                         *************
                         * MINITREES *
                         *    FOR    *
                         *  DUMMY-T  *
                         *************

              VARIABLE                                B.C. & NOT
              DESCRIPTION      FAULT        GATE     ALLOWED FAULTS
              -----------      -----        ----     ------------

    M-EVENT :    P    IN       HI           OR       LO
                                                     LK-LP-ENV

    T-EVENT :    Q    IN       HI
              VARIABLE                                B.C. & NOT
              DESCRIPTION      FAULT        GATE     ALLOWED FAULTS
              -----------      -----        ----     ------------

    M-EVENT :    P    IN       LO           OR       HI
                                                     FL-EX-ENV

    T-EVENT :    Q    IN       LO

****************************************************************
```

```
DUMMY-T%
M
F
1
HI%
2
LK-LP-ENV%
*@*%
T
Q
1
AI%
M
F
1
LO%
2
FL-EX-ENV%
*@*%
T
Q
1
LO%
*
*+*+*%
```

I.2.7 Heat Exchanger

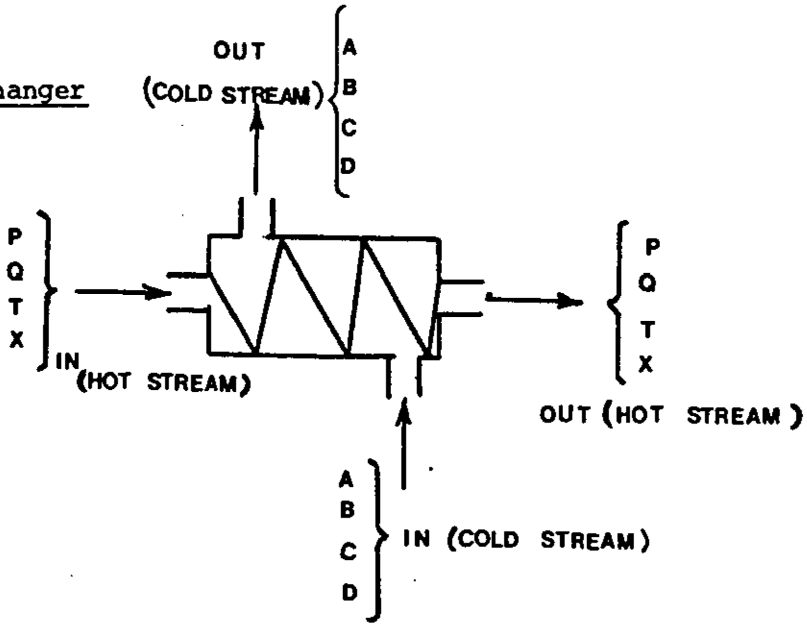


Fig. I.2.7 Heat Exchanger

Equations Used	Assumptions
1) $\frac{dP_{IN}}{dt} = G_1\ (Q_{IN}-Q_{OUT})$	1) Shell and Tube design
2) $\frac{dA_{IN}}{dt} = G_2\ (B_{IN}-B_{OUT})$	2) Shell perfectly mixed
	3) Plug flow in tube
3) $Q_{OUT} = k_1\ (P_{IN}-P_{OUT})^{\frac{1}{2}}$	4) No phase change
	5) No wall resistance
4) $B_{OUT} = k_2\ (A_{IN}-A_{OUT})^{\frac{1}{2}}$	6) No density changes
5) $Z = \text{UArea}\ \frac{\left((T_{IN}-C_{OUT})+(T_{OUT}-C_{IN})\right)}{2}$	7) $\Delta T = \frac{\Delta T_1 + \Delta T_2}{2}$
6) $T_{OUT} = \frac{-Z+Q_{IN}\rho_{HOT}C_{pHOT}T_{IN}}{Q_{OUT}\rho_{HOT}C_{pHOT}}$	8) Counter-current
7) $C_{OUT} = \frac{Z+B_{IN}\rho_{cold}C_{pcold}C_{IN}}{B_{OUT}\rho_{cold}C_{pcold}}$	**Special Nomenclature in the model**

8) $X_{OUT} = X_{IN}$	A = Pressure
	B = Flow
	C = Temperature
	D = Molar fraction
	For cold streams

9) $D_{OUT} = D_{IN}$	
Name used to identify unit: HEAT-EX	

```
******************************************************************************

                              *************
                              * MINITREES *
                              *    FOR    *
                              *  HEAT-EX  *
                              *************

                  VARIABLE                                        B.C. & NOT
                 DESCRIPTION        FAULT          GATE        ALLOWED FAULTS
                 -----------        -----          ----        -------------

     M-EVENT :     T    OUT         HI             OR          LO

     T-EVENT :     T    IN          HI
     T-EVENT :     Q    OUT         HI
     R-EVENT :                      Z-LO
     B-EVENT :                      EXT-FIRE

                  VARIABLE                                        B.C. & NOT
                 DESCRIPTION        FAULT          GATE        ALLOWED FAULTS
                 -----------        -----          ----        -------------

     M-EVENT :     T    OUT         LO             OR          HI

     T-EVENT :     T    IN          LO
     R-EVENT :                      A
     R-EVENT :                      Z-HI

                  VARIABLE                                        B.C. & NOT
                 DESCRIPTION        FAULT          GATE        ALLOWED FAULTS
                 -----------        -----          ----        -------------

     M-EVENT :                      A              EX-OR

     T-EVENT :     Q    OUT         LO
     T-EVENT :     Q    OUT         NO-FLOW

                  VARIABLE                                        B.C. & NOT
                 DESCRIPTION        FAULT          GATE        ALLOWED FAULTS
                 -----------        -----          ----        -------------

     M-EVENT :     C    OUT         HI             OR          LO

     T-EVENT :     C    IN          HI
     R-EVENT :                      B
     R-EVENT :                      Z-HI
     B-EVENT :                      EXT-FIRE

                  VARIABLE                                        B.C. & NOT
                 DESCRIPTION        FAULT          GATE        ALLOWED FAULTS
                 -----------        -----          ----        -------------

     M-EVENT :                      B              EX-OR

     T-EVENT :     B    OUT         LO
     R-EVENT :                      C

                  VARIABLE                                        B.C. & NOT
                 DESCRIPTION        FAULT          GATE        ALLOWED FAULTS
                 ----------         -----         ----      -----------

M-EVENT :                           C             AND

T-EVENT :        B     OUT          NO-FLOW
T-EVENT :        Q     OUT          GT0

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 ----------         -----         ----      -----------

M-EVENT :        Q     OUT          GT0           OR

T-EVENT :        Q     IN           GT0

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 ----------         -----         ----      -----------

M-EVENT :        C     OUT          LO            OR        HI

T-EVENT :        C     IN           LO
T-EVENT :        B     OUT          LO
R-EVENT :                           Z-LO

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 ----------         -----         ----      -----------

M-EVENT :                           Z-HI          OR        Z-LO

T-EVENT :        T     OUT          HI
T-EVENT :        C     OUT          LO

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 ----------         -----         ----      -----------

M-EVENT :                           Z-LO          OR        Z-HI

T-EVENT :        T     OUT          LO
T-EVENT :        C     OUT          HI

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 ----------         -----         ----      -----------

M-EVENT :        P     IN           HI            OR        LO
                                                            LK-LP-ENV

T-EVENT :        Q     IN           HI
T-EVENT :        Q     OUT          LO
B-EVENT :                           BLOCKAGE
B-EVENT :                           LK-HP-ENV

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 ----------         -----         ----      -----------

M-EVENT :        P     IN           LO            OR        HI
                                                            FL-EX-ENV
```

```
T-EVENT :     Q     IN       LO
T-EVENT :     Q     OUT      HI
B-EVENT :                    LK-LP-ENV

              VARIABLE                                B.C. & NOT
             DESCRIPTION     FAULT       GATE        ALLOWED FAULTS
             -----------     -----       ----        ------------

M-EVENT :     A     IN       HI          OR          LO
                                                     LK-LP-ENV

T-EVENT :     B     IN       HI
T-EVENT :     B     OUT      LO
B-EVENT :                    BLOCKAGE
B-EVENT :                    LK-HP-ENV

              VARIABLE                                B.C. & NOT
             DESCRIPTION     FAULT       GATE        ALLOWED FAULTS
             -----------     -----       ----        ------------

M-EVENT :     A     IN       LO          OR          HI
                                                     FL-EX-ENV

T-EVENT :     B     IN       LO
T-EVENT :     B     OUT      HI
B-EVENT :                    LK-LP-ENV

              VARIABLE                                B.C. & NOT
             DESCRIPTION     FAULT       GATE        ALLOWED FAULTS
             -----------     -----       ----        ------------

M-EVENT :     Q     OUT      HI          OR          LO
                                                     CLOSED
                                                     BLOCKAGE
                                                     SHUT

T-EVENT :     P     IN       HI
T-EVENT :     P     OUT      LO
B-EVENT :                    FL-EX-ENV

              VARIABLE                                B.C. & NOT
             DESCRIPTION     FAULT       GATE        ALLOWED FAULTS
             -----------     -----       ----        ------------

M-EVENT :     Q     OUT      LO          OR          HI

T-EVENT :     P     IN       LO
T-EVENT :     P     OUT      HI
B-EVENT :                    BLOCKAGE
B-EVENT :                    LK-LP-ENV

              VARIABLE                                B.C. & NOT
             DESCRIPTION     FAULT       GATE        ALLOWED FAULTS
             -----------     -----       ----        ------------

M-EVENT :     B     OUT      HI          OR          LO
                                                     CLOSED
                                                     BLOCKAGE
                                                     SHUT
```

```
T-EVENT :    A     IN        HI
T-EVENT :    A     OUT       LO
B-EVENT :                    FL-EX-ENV

           VARIABLE                                B.C. & NOT
          DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
          -----------        -----        ----     -------------

M-EVENT :    B     OUT       LO           OR       HI

T-EVENT :    A     IN        LO
T-EVENT :    A     OUT       HI
B-EVENT :                    BLOCKAGE
B-EVENT :                    LK-LP-ENV

           VARIABLE                                B.C. & NOT
          DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
          -----------        -----        ----     -------------

M-EVENT :    X     OUT       HI           OR       LO

T-EVENT :    X     IN        HI

           VARIABLE                                B.C. & NOT
          DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
          -----------        -----        ----     -------------

M-EVENT :    X     OUT       LO           OR       HI

T-EVENT :    X     IN        LO

           VARIABLE                                B.C. & NOT
          DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
          -----------        -----        ----     -------------

M-EVENT :    D     OUT       HI           OR       LO

T-EVENT :    D     IN        HI

           VARIABLE                                B.C. & NOT
          DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
          -----------        -----        ----     -------------

M-EVENT :    D     OUT       LO           OR       HI

T-EVENT :    D     IN        LO

           VARIABLE                                B.C. & NOT
          DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
          -----------        -----        ----     -------------

M-EVENT :    B     OUT       NO-FLOW      OR

T-EVENT :    B     IN        NO-FLOW
B-EVENT :                    COMP-BLOC

           VARIABLE                                B.C. & NOT
          DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
          -----------        -----        ----     -------------

M-EVENT :    Q     OUT       NO-FLOW      OR
```

```
T-EVENT :     Q     IN         NO-FLOW
B-EVENT :                      COMP-BLOC

               VARIABLE                                 B.C. & NOT
              DESCRIPTION      FAULT         GATE      ALLOWED FAULTS
              -----------      -----         ----      -------------

M-EVENT :     B     OUT        GT0           OR

T-EVENT :     B     IN         GT0

*********************************************************************
```

```
HEAT-EX%
M
T
0
HI%
0
*0*%
T
T
1
AI%
T
Q
0
HI%
R

2
Z-LO%
B

2
EXT-FIRE%
0.0021
M
T
0
LO%
0
*0*%
T
T
1
LO%
R

2
A%
R

2
Z-HI%
M

2
A%
2
*0*%
T
Q
0
LO%
T
Q
2
NO-FLOW%
M
C
0
HI%
0
```

```
*0*%
T
C
1
AI%
R

2
B%
R

2
Z-HI%
B

2
EXT-FIRE%
0.0022
M

2
B%
2
*0*%
T
B
0
LO%
R

2
C%
M

2
C%
1
*0*%
T
B
0
NO-FLOW%
T
Q
0
GT0%
M
Q
0
GT0%
0
*0*%
T
Q
1
GT0%
M
C
0
LO%
0
```

```
*0*%
T
C
1
LO%
T
B
0
LO%
R

2
Z-LO%
M

2
Z-HI%
0
Z-LO%
*0*%
T
T
2
HI%
T
C
0
LO%
M


2
Z-LO%
0
Z-HI%
*0*%
T
T
2
LO%
T
C
0
HI%
M
P
1
HI%
0
LK-LP-ENV%
*0*%
T
Q
1
HI%
T
Q
0
LO%
B



PAGE                    HEATEX.LAT
--------
```

```
2
BLOCKAGE%
0.0003
B

2
LK-HP-ENV%
0.0024
M
P
1
LO%
0
FL-EX-ENV%
*@*%
T
Q
1
LO%
T
Q
0
HI%
B

2
LK-LP-ENV%
0.0025
M
A
1
HI%
0
LK-LP-ENV%
*@*%
T
B
1
HI%
T
B
0
LO%
B

2
BLOCKAGE%
0.2206
B

2
LK-HP-ENV%
0.2017
M
A
1
LO%
0
FL-EX-ENV%
*@*%
T
```

```
B
1
LO%
T
B
0
HI%
B

2
LK-LP-ENV%
0.0027
M
Q
0
HI%
2
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
P
1
HI%
T
F
0
LO%
B

2
FL-EX-ENV%
0.0228
M
Q
0
LO%
0
*0*%
T
F
1
LO%
T
F
0
HI%
B

2
BLOCKAGE%
0.0009
B

2
LK-LP-ENV%
0.0010
M
B
0
```

```
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*@*%
T
A
1
HI%
T
A
0
LO%
B

2
FL-EX-ENV%
0.0011
M
B
0
LO%
0
*@*%
T
A
1
LO%
T
A
0
HI%
B

2
BLOCKAGE%
0.0012
B

2
LK-LP-ENV%
0.0013
M
X
0
HI%
0
*@*%
T
X
1
HI%
M
X
0
LO%
0
*@*%
T
X
```

```
1
LO%
M
D
0
HI%
0
*0*%
T
D
1
HI%
M
D
0
LO%
0
*0*%
T
D
1
LO%
M
B
0
NO-FLOW%
0
*0*%
T
B
1
NO-FLOW%
B

2
COMP-BLOC%
0.0001
M
Q
0
NO-FLOW%
0
*0*%
T
Q
1
NO-FLOW%
B

2
COMP-BLOC%
0.0001
M
B
0
GT0%
0
*0*%
T
B
1
```

```
GT0%
*
*+*+*%
```

I.2.8 Pipe

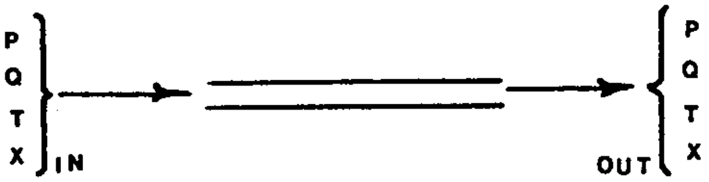


Fig. I.2.8 Pipe

Equations Used	Assumptions
1) $\frac{dP_{IN}}{dt} = G\ (Q_{IN}-Q_{OUT})$ 2) $Q_{OUT} = k(P_{IN}-P_{OUT})^{\eta}$ 3) $T_{IN} = T_{OUT}$ 4) $X_{IN} = X_{OUT}$	1) Isothermal flow 2) No density change
Name used to identify unit: PIPE	

```
*********************************************************************

                           ************
                           * MINITREES *
                           *    FOR    *
                           *   PIPE    *
                           ************

                VARIABLE                                    B.C. & NOT
               DESCRIPTION       FAULT         GATE        ALLOWED FAULTS
               -----------       -----         ----        ------------

     M-EVENT :    Q     OUT      HI            OR          LO
                                                           CLOSED
                                                           BLOCKAGE
                                                           SHUT

     T-EVENT :    P     IN       HI
     T-EVENT :    P     OUT      LO
     B-EVENT :                   FL-EX-ENV

                VARIABLE                                    B.C. & NOT
               DESCRIPTION       FAULT         GATE        ALLOWED FAULTS
               -----------       -----         ----        ------------

     M-EVENT :    Q     OUT      LO            OR          HI

     T-EVENT :    P     IN       LO
     T-EVENT :    P     OUT      HI
     B-EVENT :                   BLOCKAGE
     B-EVENT :                   LK-LP-ENV

                VARIABLE                                    B.C. & NOT
               DESCRIPTION       FAULT         GATE        ALLOWED FAULTS
               -----------       -----         ----        ------------

     M-EVENT :    P     IN       HI            OR          LO
                                                           LK-LP-ENV

     T-EVENT :    Q     IN       HI
     T-EVENT :    Q     OUT      LO
     B-EVENT :                   BLOCKAGE
     B-EVENT :                   LK-HP-ENV

                VARIABLE                                    B.C. & NOT
               DESCRIPTION       FAULT         GATE        ALLOWED FAULTS
               -----------       -----         ----        ------------

     M-EVENT :    P     IN       LO            OR          HI
                                                           FL-EX-ENV

     T-EVENT :    Q     OUT      HI
     T-EVENT :    Q     IN       LO
     B-EVENT :                   LK-LP-ENV

                VARIABLE                                    B.C. & NOT
               DESCRIPTION       FAULT         GATE        ALLOWED FAULTS
               -----------       -----         ----        ------------

     M-EVENT :    T     OUT      HI            OR          LO
```

```
T-EVENT :     T     IN         HI
B-EVENT :                      EXT-FIRE

            VARIABLE                                      B.C. & NOT
           DESCRIPTION         FAULT        GATE         ALLOWED FAULTS
           -----------         -----        ----         ------------

M-EVENT :     T     OUT        LO           OR           HI

T-EVENT :     T     IN         LO

            VARIABLE                                      B.C. & NOT
           DESCRIPTION         FAULT        GATE         ALLOWED FAULTS
           -----------         -----        ----         ------------

M-EVENT :     X     OUT        HI           OR           LO

T-EVENT :     X     IN         HI

            VARIABLE                                      B.C. & NOT
           DESCRIPTION         FAULT        GATE         ALLOWED FAULTS
           -----------         -----        ----         ------------

M-EVENT :     X     OUT        LO           OR           HI

T-EVENT :     X     IN         LO

            VARIABLE                                      B.C. & NOT
           DESCRIPTION         FAULT        GATE         ALLOWED FAULTS
           -----------         -----        ----         ------------

M-EVENT :     Q     OUT        NO-FLOW      OR

T-EVENT :     Q     IN         NO-FLOW
B-EVENT :                      COMP-BLOC

            VARIABLE                                      B.C. & NOT
           DESCRIPTION         FAULT        GATE         ALLOWED FAULTS
           -----------         -----        ----         ------------

M-EVENT :     Q     OUT        GT0          OR

T-EVENT :     Q     IN         GT0

*****************************************************************
```

```
PIPE%
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
P
1
AI%
T
P
0
LO%
B

2
FL-EX-ENV%
0.0021
M
Q
0
LO%
0
*0*%
T
P
1
LO%
T
P
0
HI%
B

2
BLOCKAGE%
0.0202
B

2
LK-LP-ENV%
0.0225
M
P
1
AI%
0
LK-LP-ENV%
*0*%
T
Q
1
AI%
T
Q
0
```

```
LO%
B

2
BLOCKAGE%
0.0003
B

2
LK-HP-ENV%
0.0004
M
P
1
LO%
0
FL-EX-ENV%
*0*%
T
Q
0
HI%
T
Q
1
LO%
B

2
LK-LP-ENV%
0.0205
M
T
0
HI%
0
*0*%
T
T
1
HI%
B

2
EXT-FIRE%
0.0221
M
T
0
LO%
0
*0*%
T
T
1
LO%
M
X
0
HI%
0
```

```
*0*%
T
X
1
HI%
M
X
0
LO%
0
*0*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*0*%
T
Q
1
NO-FLOW%
B

2
COMP-BLOC%
0.0001
M
Q
0
GT0%
0
*0*%
T
Q
1
GT0%
*
*+*+*%
```

I.2.9 Sensors

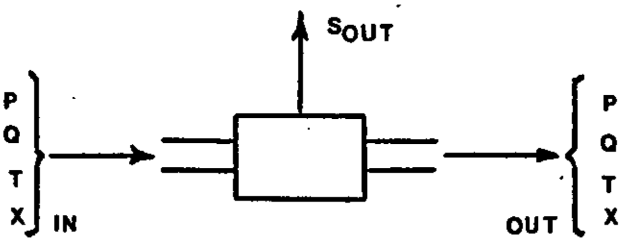


Fig. I.2.9 Sensor

<table>
<tr><th>Equations Used</th><th>Assumptions</th></tr>
<tr><td rowspan="3">1) $\frac{dP_{IN}}{dt} = G\ (Q_{IN} - Q_{OUT})$
2) $Q_{OUT} = k(P_{IN} - P_{OUT})^{\frac{1}{2}}$
3) $T_{OUT} = T_{IN}$
4) $X_{OUT} = X_{IN}$
5) $S_{OUT} = f(Y^*)$</td><td>1) Isothermal flow
2) No density change
3) Multipurpose use</td></tr>
<tr><th>Special Nomenclature</th></tr>
<tr><td>*Y = Variable under study, it may be Flow(Q), Pressure(P) or Temperature(T)
S = Output signal to controller</td></tr>
<tr><td colspan="2">Name used to identify unit: SENSOR-Y*</td></tr>
</table>

```
**********************************************************************

                            *************
                            * MINITREES *
                            *    FOR    *
                            * SENSOR-Q  *
                            *************

                 VARIABLE                                      B.C. & NOT
                DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                -----------        -----        ----     -------------

      M-EVENT :    Q     OUT       HI           OR       LO
                                                         CLOSED
                                                         BLOCKAGE
                                                         SHUT

      T-EVENT :    P     IN        HI
      T-EVENT :    P     OUT       LO

                 VARIABLE                                      B.C. & NOT
                DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                -----------        -----        ----     -------------

      M-EVENT :    Q     OUT       LO           OR       HI

      T-EVENT :    P     IN        LO
      T-EVENT :    P     OUT       HI

                 VARIABLE                                      B.C. & NOT
                DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                -----------        -----        ----     -------------

      M-EVENT :    P     IN        HI           OR       LO
                                                         LK-LP-ENV

      T-EVENT :    Q     IN        HI
      T-EVENT :    Q     OUT       LO

                 VARIABLE                                      B.C. & NOT
                DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                -----------        -----        ----     -------------

      M-EVENT :    P     IN        LO           OR       HI
                                                         LK-HP-ENV

      T-EVENT :    Q     IN        LO
      T-EVENT :    Q     OUT       HI

                 VARIABLE                                      B.C. & NOT
                DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                -----------        -----        ----     -------------

      M-EVENT :    S     OUT       HI           OR       LO

      T-EVENT :    Q     IN        HI
      B-EVENT :                    SEN-FA-HI

                 VARIABLE                                      B.C. & NOT
                DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
```

```
                 -----------        -----        ----     ------------

M-EVENT :         S      OUT        LO           OR       HI

T-EVENT :         Q      IN         LO
B-EVENT :                           SEN-FA-LO

                  VARIABLE                                B.C. & NOT
                 DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                 -----------        -----        ----     ------------

M-EVENT :         S      OUT        NO-CHANGE    OR

B-EVENT :                           SENS-STCK

                  VARIABLE                                B.C. & NOT
                 DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                 -----------        -----        ----     ------------

M-EVENT :         T      OUT        HI           OR       LO

T-EVENT :         T      IN         HI
B-EVENT :                           EXT-FIRE

                  VARIABLE                                B.C. & NOT
                 DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                 -----------        -----        ----     ------------

M-EVENT :         T      OUT        LO           OR       HI

T-EVENT :         T      IN         LO

                  VARIABLE                                B.C. & NOT
                 DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                 -----------        -----        ----     ------------

M-EVENT :         X      OUT        HI           OR       LO

T-EVENT :         X      IN         HI

                  VARIABLE                                B.C. & NOT
                 DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                 -----------        -----        ----     ------------

M-EVENT :         X      OUT        LO           OR       HI

T-EVENT :         X      IN         LO

                  VARIABLE                                B.C. & NOT
                 DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                 -----------        -----        ----     ------------

M-EVENT :         Q      OUT        NO-FLOW      OR

T-EVENT :         Q      IN         NO-FLOW
B-EVENT :                           COMP-BLOC

                  VARIABLE                                B.C. & NOT
                 DESCRIPTION        FAULT        GATE     ALLOWED FAULTS
                 -----------        -----        ----     ------------
```

```
M-EVENT :      S      OUT        NO-SIGNAL      EX-OR

T-EVENT :      Q      IN         GT0
R-EVENT :                        A

             VARIABLE                                     B.C. & NOT
            DESCRIPTION          FAULT          GATE      ALLOWED FAULTS
            -----------          -----          ----      ------------

M-EVENT :                        A              AND

T-EVENT :      Q      IN         NO-FLOW
B-EVENT :                        SI-STR-PL
*************************************************************
```

```
SENSOR-Q%
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
P
1
HI%
T
P
0
LO%
M
Q
0
LO%
0
*0*%
T
P
1
LO%
T
P
0
HI%
M
P
1
HI%
0
LK-LP-ENV%
*0*%
T
Q
1
HI%
T
Q
0
LO%
M
P
1
LO%
0
LK-HP-ENV%
*0*%
T
Q
1
LO%
T
Q
0
```

```
HI%
M
S
0
HI%
0
*@*%
T
Q
1
HI%
B

2
SEN-FA-HI%
0.0221
M
S
0
LO%
0
*@*%
T
Q
1
LO%
B

2
SEN-FA-LO%
0.0002
M
S
0
NO-CHANGE%
0
*@*%
B

2
SENS-STCK%
0.0023
M
T
0
HI%
0
*@*%
T
T
1
HI%
B

2
EXT-FIRE%
0.0031
M
T
0
LO%
```

```
0
*@*%
T
T
1
LO%
M
X
0
HI%
0
*@*%
T
X
1
HI%
M
X
0
LO%
0
*@*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*@*%
T
Q
1
NO-FLOW%
B

2
COMP-BLOC%
0.0001
M
S
0
NO-SIGNAL%
2
*@*%
T
Q
1
GT0%
R

2
A%
M

2
A%
1
*@*%


PAGE                         SENSW .DAT
--------
```

```
T
Q
1
NO-FLOW%
B

2
SI-STR-PL%
0.02001
*
*+*+*%
```

```
*********************************************************************

                              ************
                              * MINITREES *
                              *    FOR    *
                              * SENSOR-T  *
                              ************

                 VARIABLE                                       B.C. & NOT
                DESCRIPTION          FAULT          GATE        ALLOWED FAULTS
                -----------          -----          ----        ------------

       M-EVENT :     Q     OUT       HI             OR          LO
                                                                CLOSED
                                                                BLOCKAGE
                                                                SHUT

       T-EVENT :     P     IN        HI
       T-EVENT :     P     OUT       LO

                 VARIABLE                                       B.C. & NOT
                DESCRIPTION          FAULT          GATE        ALLOWED FAULTS
                -----------          -----          ----        ------------

       M-EVENT :     Q     OUT       LO             OR          HI

       T-EVENT :     P     IN        LO
       T-EVENT :     P     OUT       HI

                 VARIABLE                                       B.C. & NOT
                DESCRIPTION          FAULT          GATE        ALLOWED FAULTS
                -----------          -----          ----        ------------

       M-EVENT :     P     IN        HI             OR          LO
                                                                LK-LP-ENV

       T-EVENT :     Q     IN        HI
       T-EVENT :     Q     OUT       LO

                 VARIABLE                                       B.C. & NOT
                DESCRIPTION          FAULT          GATE        ALLOWED FAULTS
                -----------          -----          ----        ------------

       M-EVENT :     P     IN        LO             OR          HI
                                                                LK-HP-ENV

       T-EVENT :     Q     IN        LO
       T-EVENT :     Q     OUT       HI

                 VARIABLE                                       B.C. & NOT
                DESCRIPTION          FAULT          GATE        ALLOWED FAULTS
                -----------          -----          ----        ------------

       M-EVENT :     S     OUT       HI             OR          LO

       T-EVENT :     T     IN        HI
       B-EVENT :                     SEN-FA-HI

                 VARIABLE                                       B.C. & NOT
                DESCRIPTION          FAULT          GATE        ALLOWED FAULTS
```

```
                 -----------        -----         ----      ------------

M-EVENT :        S     OUT          LO            OR        HI

T-EVENT :        T     IN           LO
B-EVENT :                           SEN-FA-LO

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 -----------        -----         ----      ------------

M-EVENT :        S     OUT          NO-CHANGE     OR

B-EVENT :                           SENS-STCK

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 -----------        -----         ----      ------------

M-EVENT :        T     OUT          HI            OR        LO

T-EVENT :        T     IN           HI
B-EVENT :                           EXT-FIRE

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 -----------        -----         ----      ------------

M-EVENT :        T     OUT          LO            OR        HI

T-EVENT :        T     IN           LO

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 -----------        -----         ----      ------------

M-EVENT :        X     OUT          HI            OR        LO

T-EVENT :        X     IN           HI

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 -----------        -----         ----      ------------

M-EVENT :        X     OUT          LO            OR        HI

T-EVENT :        X     IN           LO

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 -----------        -----         ----      ------------

M-EVENT :        Q     OUT          NO-FLOW       OR

T-EVENT :        Q     IN           NO-FLOW
B-EVENT :                           COMP-BLOC

                  VARIABLE                                  B.C. & NOT
                 DESCRIPTION        FAULT         GATE      ALLOWED FAULTS
                 -----------        -----         ----      ------------


PAGE                       SENT   .PRO
--------
```

```
     M-EVENT :     Q    OUT       GT0           OR

     T-EVENT :     Q    IN        GT0

************************************************************************
```

```
SENSOR-T%
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
P
1
AI%
T
P
0
LO%
M
Q
0
LO%
0
*0*%
T
P
1
LO%
T
P
0
HI%
M
P
1
AI%
0
LK-LP-ENV%
*0*%
T
Q
1
AI%
T
Q
0
LO%
M
P
1
LO%
0
LK-HP-ENV%
*0*%
T
Q
1
LO%
T
Q
0
```

```
HI%
M
S
0
HI%
0
*0*%
T
T
1
HI%
B

2
SEN-FA-HI%
0.0021
M
S
0
LO%
0
*0*%
T
T
1
LO%
B

2
SEN-FA-LO%
0.0022
M
S
0
NO-CHANGE%
0
*0*%
B

2
SENS-STCK%
0.0023
M
T
0
HI%
0
*0*%
T
T
1
HI%
B

2
EXT-FIRE%
0.0021
M
T
0
LO%
```

```
0
*0*%
T
T
1
LO%
M
X
0
HI%
0
*0*%
T
X
1
HI%
M
X
0
LO%
0
*0*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*0*%
T
Q
1
NO-FLOW%
B

2
COMP-BLOC%
0.0201
M
Q
0
GT0%
0
*0*%
T
Q
1
GT0%
*
*+*+*%
```

I.2.10 Valves

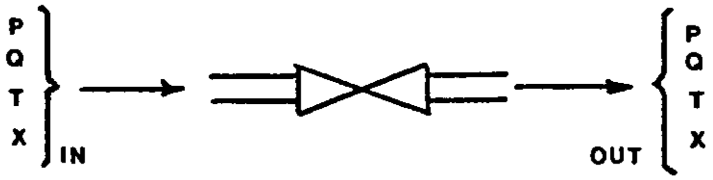


Fig. I.2.10 Valves

Equations Used	Assumptions
1) $\frac{dP_{IN}}{dt} = G\ (Q_{IN}-Q_{OUT})$	1) No heat loss
2) $Q_{OUT} = k\ (P_{IN}-P_{OUT})^{\eta}$	2) Subsonic flow
3) $T_{IN} = T_{OUT}$	
4) $X_{IN} = X_{OUT}$	
Name used to identify unit: VALVE	

```
**********************************************************************

                           ************
                           * MINITREES *
                           *    FOR    *
                           *   VALVE   *
                           ************

           VARIABLE                                      B.C. & NOT
          DESCRIPTION          FAULT          GATE      ALLOWED FAULTS
          -----------          -----          ----      ------------

  M-EVENT :     Q     OUT      HI             OR        LO
                                                        CLOSED
                                                        BLOCKAGE
                                                        SHUT

  T-EVENT :     P     IN       HI
  T-EVENT :     P     OUT      LO
  B-EVENT :                    WIDE-OPEN
  B-EVENT :                    FL-EX-ENV

           VARIABLE                                      B.C. & NOT
          DESCRIPTION          FAULT          GATE      ALLOWED FAULTS
          -----------          -----          ----      ------------

  M-EVENT :     Q     OUT      LO             OR        HI
                                                        OPEN
                                                        WIDE-OPEN
                                                        FAIL-OPEN

  T-EVENT :     P     OUT      HI
  T-EVENT :     P     IN       LO
  R-EVENT :                    CLOSED
  B-EVENT :                    LK-LP-ENV

           VARIABLE                                      B.C. & NOT
          DESCRIPTION          FAULT          GATE      ALLOWED FAULTS
          -----------          -----          ----      ------------

  M-EVENT :                    CLOSED         OR        OPEN

  B-EVENT :                    BLOCKAGE
  B-EVENT :                    SHUT

           VARIABLE                                      B.C. & NOT
          DESCRIPTION          FAULT          GATE      ALLOWED FAULTS
          -----------          -----          ----      ------------

  M-EVENT :     P     IN       HI             OR        LO
                                                        LK-LP-ENV

  T-EVENT :     Q     IN       HI
  T-EVENT :     Q     OUT      LO
  R-EVENT :                    CLOSED
  B-EVENT :                    LK-HP-ENV

           VARIABLE                                      B.C. & NOT
          DESCRIPTION          FAULT          GATE      ALLOWED FAULTS
          -----------          -----          ----      ------------
```

```
M-EVENT :     P     IN      LO              OR      HI
                                                    FL-EX-ENV

T-EVENT :     Q     OUT     HI
T-EVENT :     Q     IN      LO
B-EVENT :                   LK-LP-ENV
B-EVENT :                   WIDE-OPEN

            VARIABLE                                B.C. & NOT
           DESCRIPTION      FAULT           GATE    ALLOWED FAULTS
           -----------      -----           ----    ------------

M-EVENT :     T     OUT     HI              OR      LO

T-EVENT :     T     IN      HI
B-EVENT :                   EXT-FIRE

            VARIABLE                                B.C. & NOT
           DESCRIPTION      FAULT           GATE    ALLOWED FAULTS
           -----------      -----           ----    ------------

M-EVENT :     T     OUT     LO              OR      HI

T-EVENT :     T     IN      LO

            VARIABLE                                B.C. & NOT
           DESCRIPTION      FAULT           GATE    ALLOWED FAULTS
           -----------      -----           ----    ------------

M-EVENT :     X     OUT     HI              OR      LO

T-EVENT :     X     IN      HI

            VARIABLE                                B.C. & NOT
           DESCRIPTION      FAULT           GATE    ALLOWED FAULTS
           -----------      -----           ----    ------------

M-EVENT :     X     OUT     LO              OR      HI

T-EVENT :     X     IN      LO

            VARIABLE                                B.C. & NOT
           DESCRIPTION      FAULT           GATE    ALLOWED FAULTS
           -----------      -----           ----    ------------

M-EVENT :     Q     OUT     NO-FLOW         OR

T-EVENT :     Q     IN      NO-FLOW
B-EVENT :                   COMP-BLOC

            VARIABLE                                B.C. & NOT
           DESCRIPTION      FAULT           GATE    ALLOWED FAULTS
           -----------      -----           ----    ------------

M-EVENT :     Q     OUT     GT0             OR

T-EVENT :     Q     IN      GT0

*****************************************************************
```

```
VALVE%
M
Q
0
HI%
0
BLOCKAGE%
CLOSED%
SHUT%
*0*%
T
P
1
AI%
T
P
0
LO%
B

2
WIDE-OPEN%
0.0221
B

2
FL-EX-ENV%
0.0222
M
Q
0
LO%
0
OPEN%
WIDE-OPEN%
FAIL-OPEN%
*0*%
T
P
0
HI%
T
P
1
LO%
R

2
CLOSED%
B

2
LK-LP-ENV%
0.0223
M

2
CLOSED%
0
*0*%
B
```

```
2
BLOCKAGE%
0.0024
B

2
SHUT%
0.0025
M
P
1
AI%
0
LK-LP-ENV%
*0*%
T
Q
1
AI%
T
Q
0
LO%
R

2
CLOSED%
B

2
LK-HP-ENV%
0.0026
M
P
1
LO%
0
FL-EX-ENV%
*0*%
T
Q
0
HI%
T
Q
1
LO%
B

2
LK-LF-ENV%
0.0027
B

2
WIDE-OPEN%
0.0028
M
T
0
```

```
HI%
0
*@*%
T
T
1
AI%
B

2
EXT-FIRE%
0.0201
M
T
0
LO%
0
*@*%
T
T
1
LO%
M
X
0
HI%
0
*@*%
T
X
1
HI%
M
X
0
LO%
0
*@*%
T
X
1
LO%
M
Q
0
NO-FLOW%
0
*@*%
T
Q
1
NO-FLOW%
B

2
COMP-BLOC%
2.0201
M
Q
0
GT0%
```

```
0
*@*%
T
Q
I
GT0%
*
*+*+*%
```

Table I.1

Nomenclature of the Unit Minitrees

Symbol	Quantity	Typical Units
A	Area	m^2
G	Gain constant (for high-gain equations)	-
K	Constant	-
L	Level	m
M	Mass	kg
P	Pressure	N/m^2
Q	Volumetric flow rate	m^3/s
T	Temperature	^oK
t	time	sec
U	Overall Heat Transfer Coefficient	$W/_{m^2}\,^oK$
V	Volume	m^3
X	Mass fraction	-
Z	Heat flux	W/m^2
ρ	Density	kg/m^3

Table I.2

Key to the Names of the Unit Models

Name Used in the Minitrees and trees	Name of the Unit
CENT-PUMP	Centrifugal Pump
CLOSED-TK	Closed Tank
CNTRLV-SC	Control Valve (Special Case)
CNTRL-VAL	Control Valve
CNTROLLER	Controller
DUMMY-H	Dummy Head
DUMMY-T	Dummy Tail
HEAT-EX	Heat Exchanger
PIPE	Pipe
SENSOR-Q	Flow Sensor
SENSOR-P	Pressure-Sensor
SENSOR-T	Temperature Sensor
VALVE	Valve

Table I.3

Key to the Faults Used in the Trees and Unit Minitrees

Name used in the trees and unit minitrees	Fault
A	Dummy fault
B	Dummy fault
BLOCKAGE	Blockage
C	Dummy fault
CLOSED	Closed
COMP-BLOC	Completely Blocked
CONT-F-HI	Controller Fails High
CONT-F-LO	Controller Fails Low
CONT-STCK	Controller Stuck
BLOC-OUTL	Blockage at the outlet of tank
BLOC-INLE	Blockage at the inlet of tank
D	Dummy fault
DUMMY	Dummy fault (for general use)
EXT-FIRE	External fire
FAI-CLOSE	Fails Close
FAIL-HI	Fails High
FAIL-LO	Fails Low
FAIL-OPEN	Fails Open
FAIL-TO-CL	Fails to close on demand
FAIL-TO-OP	Fails to open on demand
FL-EX-ENV	Flow from External Environment
GT0	Flow in the unit
HI	High
LO	Low
LK-LP-ENV	Leak to Low Pressure Environment
LK-HP-ENV	Leak from High Pressure Environment
MANUAL	Manual
MECH-FAIL	Mechanic failure
NO-CHANGE	No change
NO-FLOW	No flow

/continued

Table I.3 (continued)

Name used in the trees and unit minitrees	Fault
NO-SIGNAL	No Signal
OPEN	Open
OTHER-CAU	Other Causes
SENS-STCK	Sensor Stuck
SEN-FA-HI	Sensor Fails High
SEN-FA-LO	Sensor Fails Low
SET-PO-HI	Set Point High
SET-PO-LO	Set Point Low
SHUT	Shut
SHUTDOWN	Shutdown
SI-STR-PL	Signal Stream Plugged
VALV-STCK	Valve Stuck
WIDE-OPEN	Valve Wide Open
Z-HI	Heat flux High
Z-LO	Heat flux Low

Note: The dummy faults are used only in Replaced Events