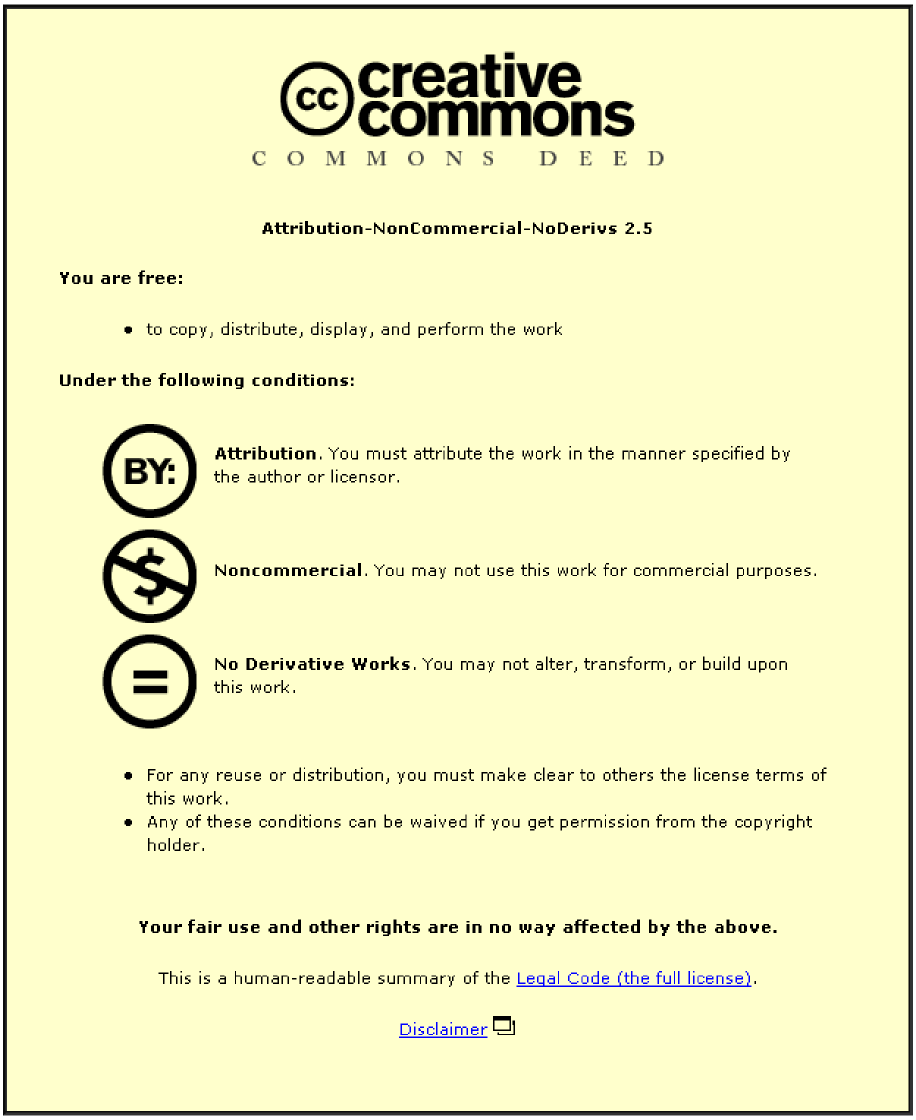


THE MODELLING OF FAULT CONDITIONS
IN CHEMICAL PROCESS PLANT

BY

BRIAN EDWARD KELLY

A DOCTORAL THESIS

Submitted in partial fulfilment of the requirements of Doctor of Philosophy of the Loughborough University of Technology (1987)

Acknowledgements

I wish to express my gratitude to the two companies for whom I have worked since leaving Loughborough University of Technology, Drew Scientific Limited, and Digital Applications International. Without the use of their facilities, producing this thesis would have been even harder.

To my family, for their incessant encouragement, which saw me through the hard times.

And last, but certainly not least, to my supervisor Professor F.P. Lees, for advice and encouragement. Without him, there would be no thesis.

Contents

1) Introduction 1-1

2) Literature Survey 2-1
2.1) Fault Tree Analysis 2-1
2.2) Fault Tree Synthesis 2-4

3) Basic Principles 3-1
3.1) Decomposition 3-1
3.1.1) Full Decomposition 3-4
3.1.2) Minimal Decomposition 3-7
3.2) Modelling 3-8
3.2.1) Modelling of Process Units 3-9
3.2.2) Fault Types 3-11
3.2.3) Definition of Models 3-12
3.2.3.1) Propagation Equations 3-13
3.2.3.2) Event Statements 3-14
3.2.3.3) Modified Decision Tables 3-17
3.2.4) The Modelling of Flow 3-18
3.2.5) Top Event Modelling 3-22
3.3) Fault Tree Synthesis 3-24
3.4) Fault Tree Synthesis - A Simple Example 3-26
3.4.1) Decomposition 3-26
3.4.2) Unit Modelling 3-27
3.4.2.1) Pipe 3-27
3.4.2.2) Counter-Current Heat Exchanger 3-28
3.4.2.3) Centrifugal Pump 3-34
3.4.3) Top Event 3-36
3.4.4) Fault Tree Synthesis 3-36

4) More Advanced Principles 4-1
4.1) More Detailed Modelling 4-1
4.1.1) Flow and Pressure Gradient 4-1
4.1.2) Temperature and Composition 4-6
4.1.3) Pressure 4-7
4.1.4) Temperature and Composition in Reverse Flow 4-9
4.1.5) Flow Ratio 4-11
4.1.6) Total Component Flow 4-14
4.2) A Synthesis Example 4-16
4.2.1) Decomposition 4-16
4.2.2) Unit Modelling 4-17
4.2.2.1) Pipe 4-17
4.2.2.2) Binary Mixer 4-18
4.2.3) Top Event Modelling 4-22
4.2.4) Synthesis 4-23

5) The Modelling of Vessels 5-1
5.1) Closed Tank containing Supercooled Liquid 5-2
5.1.1) Events at Vessel Ports 5-2
5.1.2) Events at Inlet and Outlet Ports 5-4
5.1.3) Summary 5-10
5.2) Binary Distillation Column 5-12
5.2.1) Events at Vessel Ports 5-13
5.2.2) Events at Inlet and Outlet Ports 5-15
5.2.3) Summary 5-17
5.3) Partial Reboiler 5-19
5.3.1) Events at Vessel Ports 5-19
5.3.2) Events at Inlet and Outlet Ports 5-22
5.3.3) Summary 5-23
5.4) Fault Tree Synthesis - An Example 5-25
5.4.1) Decomposition 5-25
5.4.2) Unit Modelling 5-25
5.4.3) Top Event Modelling 5-26
5.4.4) Fault Tree Synthesis 5-26

6) Control Loops 6-1
6.1) The Problem 6-1
6.2) The Approach of Others 6-4
6.3) A Solution 6-6
6.3.1) Deviation of a Regulated Variable 6-6
6.3.1.1) Feedback Loop Example 6-8
6.3.1.2) Feedforward Loop Example 6-9
6.3.2) Deviation of a Manipulated Variable 6-10
6.3.2.1) Feedback Loop Example 6-11
6.3.2.2) Feedforward Loop Example 6-12
6.3.3) Complex Control Loops 6-13
6.3.3.1) Complex Control Loop - Regulalated Variable 6-13
6.3.4) Modelling Control System Components 6-15
6.3.4.1) Sensors 6-15
6.3.4.2) Controllers 6-17
6.3.4.3) Control Valves 6-20

7) Trip Systems 7-1
7.1) The Problem 7-1
7.2) The Approach of Others 7-3
7.3) A Solution 7-5
7.3.1) Trip Valve Normally Open 7-7
7.3.2) Trip Valve Normally Closed 7-9
7.3.3) Trip Systems with Multiple Sensors 7-11
7.3.4) Complex Trip Systems 7-12
7.3.5) Feedforward Trip Systems 7-13
7.3.6) A Problem with Flow 7-14
7.3.7) Trip System Operational Failure 7-16
7.3.8) Trip System Component Modelling 7-17
7.3.8.1) Sensors 7-17
7.3.8.2) Trip Switches 7-17
7.3.8.3) Trip Valves 7-19

8) Secondary Failures 8-1
8.1) The Problem 8-1
8.2) The Approach of Others 8-1
8.3) A Solution 8-2
8.3.1) The Causes of Secondary Failures 8-2
8.3.2) The Effects of Secondary Failures 8-4
8.3.3) Sample Applications 8-5
8.3.3.1) Variable Deviation Effects 8-5
8.3.3.2) Basic Fault Effects 8-6
8.3.4) Multiple Secondary Failures 8-7
8.3.4.1) Same Secondary Failure in Different Locations 8-7
8.3.4.2) Different Secondary Failures at Same Location 8-8

9) Divider-Header Combinations 9-1
9.1) The Problem 9-2
9.2) The Approach of Others 9-3
9.3) A Solution 9-4
9.3.1) The Normal Modelling Approach 9-5
9.3.2) Low Flow through Combination Types II and III 9-5
9.3.3) Flow Faults Originating within the Combination 9-8
9.3.4) Enabling Faults 9-10
9.3.5) Nesting of Divider-Header Combinations 9-12

10) Sequencing 10-1
10.1) The Problem 10-1
10.2) The Approach of Others 10-2
10.3) A Solution 10-3
10.3.1) A Simple Pump Changeover 10-4

11) Worked Examples 11-1
11.1) A Temperature Control System 11-1
11.1.1) Decomposition 11-2
11.1.2) Fault Tree Synthesis 11-2
11.2) A Composition Control System 11-5
11.2.1) Decomposition 11-6
11.2.2) Fault Tree Synthesis 11-8
11.3) A Distillation Column 11-9
11.3.1) Decomposition 11-10
11.3.2) Fault Tree Synthesis 11-11
11.4) A Propane Pipeline Problem 11-13
11.4.1) Decomposition 11-14
11.4.2) Fault Tree Synthesis 11-16
11.5) Pump Changeover 11-22
11.5.1) Decomposition 11-23
11.5.2) Sequence Definition 11-23
11.5.3) Fault Tree Synthesis 11-25

12) Discussion and Conclusions 12-1

References R-1

Appendix A A-1

1) Introduction

As the complexity of chemical plants has increased over the years, so concern about their safe and reliable operation has increased. A number of special techniques have been developed to assist engineers to assess the safety and reliability of process plant design, and these are gaining increasing acceptance in industry.

Among the techniques available is fault tree analysis. A fault tree is a failure logic diagram, which splits a single, complex fault into several less complex faults. These less complex faults can themselves be further split up, this process continuing until faults that are well understood are obtained. Thus a complex event such as the release of a toxic gas, or the manufacture of a product with too many impurities can be broken down into combinations of events such as failures of control systems and failures of protective systems.

The reason for creating a fault tree for a process plant is that there is much more information, and much more reliable information, available about the frequency of failures in control and protective systems simply because many such systems have been installed in industry for several years. By contrast, only a few complete plants, if any at all, will be able to provide reliability information. Statistically, the information available from complete plants will be less accurate, since it derives from a much smaller sample.

There are several methods of analysing fault trees to produce both qualitative and quantitative results. One of the most common assessment methods is to calculate the minimum cutsets of a fault tree. A cutset of a fault tree is defined as a set of events that will cause the top event. A minimum cutset is a such set, with the extra condition that none of the subsets of the set are themselves cutsets. For example, suppose that if two protective systems fail, then the top event of the fault tree occurs. The two failures are a cutset. However, they are NOT a minimum cutset if either of the failures on its own will cause the top event. Qualitatively, the minimum cutsets can be used to assess the safety and reliability of the plant. A minimum cutset comprising only one event (a one event minimum cutset) is more likely to occur than a minimum cutset comprising two events (a two event minimum cutset). The more one event cutsets that a fault tree has, then the less safe/reliable the plant.

More accurate information can be derived from the minimum cutsets by finding out the frequency at which the various events occur. From this, the frequencies at which the minimum cutsets and top event occur can be calculated. Another useful quantitative result is the comparative frequency of the minimum cutsets. Decreasing the frequency at which the most frequently occuring minimum cutset occurs will have the most beneficial effect on the frequency of the top event.

Fault tree analysis is now normally done by computer, because of the amount of work involved. Fault trees even for comparatively simply systems can have in excess of a thousand minimum cutsets; large systems can have more than a million minimum cutsets.

Obviously, synthesising a fault tree for a system even with only a few thousand minimum cutsets requires considerable time and effort. However, it also requires detailed knowledge about design of the plant, the way in which the plant will be operated and the failures that can occur. Only recently, as computers have become faster, more powerful, and more adept at manipulating concepts as well as numbers have computers started to be used to assist in the synthesis of fault trees.

This thesis describes research undertaken at Loughborough University of Technology, and continues from earlier research done there. The earliest work was done by Andow et al [33, 34], who investigated both fault tree synthesis and alarm analysis. The basic approach used to synthesise fault trees was to model the failures of small components of process plant, such as pipes and valves, and to construct the failure model for a plant by combining these models together. This work has been extended by Martin-Solis et al [35, 36, 37], to investigate a small number of examples. The work described in this thesis extends the work of Martin-Solis to study larger and more complex examples.

The research outlined in this thesis concentrated on two topics. Firstly, there were several areas of modelling that Martin-Solis et al did not study in detail, such as reverse flow and multiple component systems. These topics were examined, and models that could cope with such situations were created and tested.

However, the majority of the research concentrated on overcoming the problems of a purely component based technique, such as that developed by Andow and Martin-Solis. As Shafaghi [39, 40, 41] has shown, structures such as control loops and trip systems have a profound effect on the modes of failure of a process plant. These effects tend to be lost in a purely component based approach. Consider, it is clear that a temperature control loop on a reactor is supposed to maintain the reactor temperature at some predefined value, or setpoint. Such a control loop typically comprises three component models, a temperature sensor, a controller and a control valve. However, simply connecting these three component models together as indicated by the plant design does not contain all the information about the control loop. For example, it cannot be determined from the component models alone what is being controlled. Temperature sensors certainly do normally control temperature. However, flow sensors are sometimes used to control composition, via a flow ratio control loop. However, even if a sensor knew what it was controlling, it does not know the location of what it is controlling. In the reactor example above, either the temperature in the reactor, or the temperature of the feed into the reactor could be controlled. The failure modes of the plant will differ depending on which temperature is being controlled.

Neither a controller nor a control valve can determine what they are controlling. A controller simply takes an input signal and outputs a signal dependent on the difference between the input signal and some predetermined value. A control valve knows what is being manipulated, but this may be significantly different from what is being controlled. For example, a control valve could manipulate the flow of cooling water to a reactor, thus controlling the temperature in the reactor.

This thesis overcomes these problems by introducing a two-tiered approach to fault tree synthesis. The upper tier involves modelling the important structures that can occur in a process plant, such as control loops and trip systems. About half a dozen structures of this type have been identified, including control loops. The failure behaviour of these structures is, in general terms, well defined, as Shafaghi has pointed out. In detailed terms, however, the failures in such structures depend on the failures that can occur in the components that comprise the structure. The lower tier involves modelling of the process plant components, along the lines presented by Andow and Martin-Solis. Synthesising a complete fault tree involves combining the general failure structures with the detailed failure models of the components of the structure.

This two-tier approach apparently complicates the modelling process, but this is not, in fact, the case. Modelling of the upper tier is handled automatically by the synthesis package. All that an analyst has to do is to provide the plant configuration and to create failure models for the components of the plant, such as pipes, valves and reactors.

Some of the work presented in this thesis has already been published [54-57].

The contents of this thesis are as follows. Chapter 2 is a literature survey, and examines fault tree analysis briefly, before considering the work of others in automating fault tree synthesis.

Modelling of the components in a process plant forms the basis of the lower level of the two tier approach, and is described in Chapters 3, 4 and 5. Chapter 3 is an introduction to the component modelling approach, and is very closely related to the work of Andow and Martin-Solis. Chapters 4 and 5 extend this work to cover situations not considered by Andow and Martin-

Solis. Chapter 4 considers modelling of pipe-type units to cover areas like reverse flow, flow ratio and pressure. Chapter 5 considers the modelling of units such as tanks and distillation columns where the events that occur within the unit are at least as important as the events at the boundaries of the unit. Such units require special consideration, as illustrated by Chapter 5.

Chapters 6 to 10 describe the items that comprise the upper tier of modelling. Chapter 6 describes control loops and Chapter 7 trip systems. Chapter 8 introduces a method for handling what Haasl [9] has termed secondary failures, namely failures that exist because of the properties of materials in process plant. Chapter 9 considers the splitting and subsequent rejoining of process streams, such as occurs around banks of pumps or on control valve bypasses. Chapter 10 describes a technique for plant with an element of sequencing.

Chapters 6 to 10 described the items that form the upper tier, and considered them in isolation. However, there is frequently an interaction between the elements of the upper tier. There may, for example, be both a control loop and a trip system on the temperature of a reactor. Chapter 11 considers several examples, most of which have previously been considered in the literature before. In these, there are several examples of the interactions of the various elements of the upper tier.

Chapter 12 contains the discussion and conclusions.

2) Literature Survey

The concept of fault trees, that is, logic diagrams based on failure logic, was introduced by H. Watson and coworkers developing the Minuteman Missile at the Bell Telephone Laboratories in 1961. The first published work on the subject appears at a systems safety symposium in 1965 [1].

Since then, the fault tree methodology has been growing in popularity, and there is an extensive literature on the subject. This literaure recognises two distinct aspects of the technique. Fault tree synthesis (or construction) is concerned with producing the fault tree. Fault tree analysis relates to the calculation of results, both qualitative and quantitative, from fault trees.

The research outlined in this thesis is concerned mostly with fault tree synthesis, and this is reflected in this survey. However, since synthesis is pointless without analysis (and vice versa), a few words will first be written concerning fault tree analysis.

2.1) Fault Tree Analysis

The purpose of fault tree analysis is to qualify and quantify the fault tree for a particular system and top event. A number of results can be computed, the most common of which are

a) minimum cutsets (qualitative)
b) importance (qualitative or quantitative)
c) top event unreliability and unavailability (quantitative)

The minimum cutsets of a fault tree are the complete list of all the combinations of events that are both necessary and sufficient to cause the top event. Minimum cutsets can be used to look for common cause failures. If, for example, three pumps are required to fail simultaneously for the top event to occur, then a three event cutset normally results. However, certain faults may cause all three pumps to fail simultaneously. Such an event is a common cause event, and will appear as a one event cutset. In this case loss of common power supply is a common cause failure.

Importance is a measure of the effect an event or a minimum cutset has on the top event of the tree. The more important a particular term is, the more likely it is to cause the top event. Qualitative importance is deduced from the minimum cutsets. An event that is a one event cutset is more important than an event that is part of a two event cutset. An event in several two event cutsets is more important than an event that appears in only one two event cutset, and so on. There are several different methods of calculating quantitative importance, as Lambert [3] points out. However, the end result is to attempt to pinpoint where the weakest link occurs. Effort can then be directed to where it will have most effect on reducing top event occurrence.

The likelihood of the top event occuring can also be computed using fault tree analysis. A variety of parameters relating to the top event may be calculated, as Vesely [4] demonstrates. Such parameters include

a) top event reliability and unreliability
b) top event availability and unavailability
c) mean time between failures
d) mean time to repair

Fault tree analysis, particularly of large fault trees is a time-consuming occupation. Therefore, considerable effort has been spent by numerous people to formalise and computerise analysis. A considerable number of packages are available, including

a) PREP - calculate minimum cutsets
b) COMCAN - identify potential common cause failures
c) IMPORTANCE - compute various measures of importance
d) KITT - compute various top event parameters

Arendt and Fussell [5] give an excellent review of the current techniques and the packages available. The PRA Procedures guide on risk assessment for nuclear power plant [6] also reviews the analysis packages available.

2.2) Fault Tree Synthesis

In contrast to fault tree analysis, little work has been done on automating fault tree synthesis in general, and on fault tree synthesis for chemical plants in particular. There are a variety of reasons for this. Until fairly recently, computing power just could not cope with the size and nature of the problems involved. Synthesis involves a large amount of data processing, in contrast to analysis, which is more concerned with mathematical computation. Fault tree synthesis for complex systems was the task of experts knowledgable about the system under study, and skilled in the synthesis technique. Furthermore, one of the principal attractions of fault tree synthesis is that the analyst gains a deep and detailed knowledge of the system being studied. Fussell [7] and Evans [8] note that automated fault tree synthesis techniques have been resisted because it was felt that this advantage would be lost.

The reasons why little work has been done on fault tree synthesis for chemical plants are covered by Powers and Tompkins [19], and are

a) until recently, chemical plants have seldom been complex enough to warrant analysis by fault trees

b) the consequences of failure are limited. In many cases, the plant can be restarted. Contrast this with failures in the aerospace industry

c) chemical plants tend to be very robust. Even when failures occur, they can be prevented from causing catastrophic events by protective systems

d) chemical plants are very complex to model, particularly in situations outside the normal working states. Models must cover not only unit performance, but the behaviour of the chemicals in the process and their interaction

e) detailed fault trees are very time consuming to construct manually. Powers and Lapp [22] give a figure of several days per fault tree and quote the experience of one detailed study of a nuclear plant, where a detailed fault tree analysis took 25 man-years.

In recent years, several formal methodologies have been proposed for synthesis of fault trees for particular types of system. Formalisation is a necessary precursor to computerisation, since a computer must follow a set of fixed rules. Many of these methodologies have been computerised.

Even manual fault tree synthesis is, to some extent, formal, since the amount of information that must be processed is too large to handle in a haphazard way. However, such techniques are rarely formal enough for computerisation directly.

Haasl [9] is the first author on the subject of synthesis. His paper presented a manual technique, suitable for electrical systems. However, a number of important points emerged. Haasl points out that there are two types of component failures, which he terms primary failures and secondary failures. Primary failures are direct failures in the component. Secondary failures are failures induced in the component by faults in other components. The example given by Haasl concerns the failure of relay contacts to open. Either the contacts could fail to open due to a fault in the contacts (primary failure), or the contacts could be welded together by the prolonged passage of a large current (secondary failure) caused by a faulty fuse and a short-circuited motor. In chemical plant terms, the nearest equivalent is the property of the materials in the process. For example, blockage could be caused by a fault in the unit concerned (primary failure), or by low temperature, if the fluid is liable to freezing.

Fussell [10,11] in 1973 was the first to introduce an automated fault tree synthesis technique for electrical systems. Such systems are more amenable to computerisation, since only binary logic is involved. The system can either work, or it can fail. Chemical systems, on the other hand, have a variety of failure modes. For example, flow can be too low, too high, non-existent or even reversed. Different degrees of high and low may also exist. For example, a flammable mixture might exist if there is too high a concentration, but will not exist if the concentration is far too high.

The basis of Fussell's technique is that a system is built up of a number of components, or "devices", that can be modelled independently. The modelling is done with transfer functions, which are essentially small fault trees. There is one failure transfer function for each output event of a particular device. For example, a fuse has two transfer functions, one for overload (too large a current through the fuse) and one for no current. The transfer functions indicate the various ways in which the output event can occur. Current overload is caused by a large current input to the fuse and failure of the fuse to open. No current is caused by no current input to the fuse or the fuse fails open. Note the different logic that links the causes. Overload involves AND logic, that is, all the causes must exist before the output event exists. No current involves OR logic, in other words, if any cause exists, then the output event also exists.

Fault trees are built using transfer functions by linking the transfer functions together in an appropriate way. Some of the cause events of one transfer function are the output events of other transfer functions. Suppose that the fuse in the example above is linked to a relay. Then, the input current to the fuse is the output current from the relay. The model for the relay will include transfer functions for the events that are causes in the transfer functions for the fuse. For example, no current input to the fuse, or no current output from the relay, has causes no current input to relay or relay contacts open. Some cause events are not the output events of other transfer functions. Fuse fails open is an example of this type of event. Such events are called basic faults because they cannot be expanded further. As Haasl [9] points out, certain basic events may be caused by secondary failures. The analyst must decide how much detail is required in the fault tree and which events can be regarded as basic events. Transfer functions must be provided for all causes that are not basic events.

Fussell's methodolgy is complicated by the need to divide a flowsheet into "panels", and to subdivide the panels into "coalitions". A panel is a complete electrical circuit, linked to other panels only by mechanical couplings. A coalition is a complete series circuit path. Typically, there are several coalitions per panel, and several panels per flowsheet. As a result of this subdivision, there are four types of event, all developed in different ways.

A first order event is developed manually, and is used solely to resolve a complex top event into a number of less complicated events that the methodology can handle. An example of this [10] is the top event "vessel ruptures due to overpressure". The sole cause of this might be "pump operates too long", which is an event the methodology can handle. The causes of first order events are third order events.

A third order event is developed with relation to the coalitions using second order events. An example of this [11] is that "no current" through a particular device requires no current through every circuit, and hence every coalition of which the component is a member. The causes of third order events are second order events, or fourth order events.

A second order event is developed using the transfer functions for the devices. An example of this [11] is "no current through fuse", which has causes "fuse fails open" or "no current to fuse". A second order event is always within a specific coalition. The causes of second order events can be basic events, additional second order events, or fourth order events.

A fourth order event is a fault that occurs in a device because of other components that are mechanically linked to the device. An example of this [11] is "relay held open". Such faults, like second order faults, are developed using transfer functions. The causes of fourth order events may be basic events, second order events or fourth order events.

This technique can handle some rather complicated systems, as an example in one paper [10] illustrates.

The prime attraction of modelling on a component level is that the models for each individual component can be created individually, and used over and over again, in a wide variety of different situations. Models that have this property are known as context-independent models. Brown [12] examines some of the problems inherent in creating such models.

A methodology similar in some respects to that of Fussell was developed by Salem, Apostolakis, Wu and Okrent [15-18] The technique is again based on components, but the modelling is done using decision tables. These are simply enhanced truth tables with the potential for multivalued logic. For example [15], a fuse model has three output states, normal current output, no current output, and overload current output. Each of these states is modelled by one or more decision tables. One decision table links all the causes in that table with an AND gate. If more than one decision table affects a specific event, then the decision tables are connected with an OR gate.

The Salem et al methodology improves on Fussell's technique in several ways. First, by introducing a model to link wires together, Salem bypasses the necessity for coalitions and third order events. Second, no distinction is made between mechanical connections and electrical connections, eliminating panels and fourth order events. Some manual synthesis of a complex top event, to reduce it to events recognised by the technique may still be required, but all other events are modelled using decision tables.

Salem et al [15] and Wu et al [16] spend a considerable time analysing the consistency and redundancy of events in automatically constructed fault trees. This is a problem general to all automated methodologies, and was noted by Fussell [10,11]. Salem and Wu, however, consider the problem in more detail.

There are two ways in which entries in the fault tree can violate the system state. The first is that a fault cannot be a cause of a converse of the fault. This type of entry corresponds to an impossible event. The second reason is that a fault cannot be cause of itself. This type of entry corresponds to a certain event. This is much less common than the first type.

There are two ways that events in the fault tree affect the system state. First, and more obviously, all events that occur above a particular position in the fault tree, in a direct line between the position and the top event, that is, higher in the same branch of the fault tree, affect the system state at this position. The effect of this is to ensure that an event cannot be caused either by itself, or by its converse. An example of this is that "no current through fuse" cannot be a cause of either "fuse overload" or of itself. Additionally, the events under an AND gate may affect the system state in the other branches of the AND gate. This has the effect of preventing cutsets in which some events are inconsistent, such as the cutset "fuse fails to open" AND "fuse fails open".

Salem and Wu give several examples illustrating the various possibilities [15,16].

The examples presented using this technique [15-18] cover a wide variety of situations from a simple sensor system [18] to two studies on nuclear power plant [15,16].

There are several problems associated specifically with chemical plant, which mean that the methodologies of Fussell and of Salem et al cannot generally be used. These problems include

a) two way fault propagation
b) protective systems such as control and trip loops

Two way fault propagation is essential to any modelling of chemical plants. An example of two way propagation is shutting an isolation valve in a pipeline. This has effects both upstream and downstream of the valve. A more complex example is that high temperature at a particular point may result from a high temperature upstream combined with flow in the normal direction, or by a high temperature downstream combined with reverse flow. Two way propagation also appears necessary in electrical systems. After all, no current in a particular component can result from faults on either side of the component. However, the fact that electrical circuits invariably form loops mean that locations on both sides can be found simply by going far enough in one direction. Since chemical systems do not always form loops, a two way propagation facility is essential.

The occurence of protective systems is a complication as far as fault tree synthesis is concerned. The straightforward modelling techniques of Fussell and Salem, although applicable (with the two way propagation limitation) to process items like pipes and valves, cannot be used to model protection system components successfully. The reason for this is that protective system behaviour is more complex than the behaviour of simple units, since the action taken by the protective system is dependent on the state of the process.

The problems these create for the straightforward modelling methods of Fussell and of Salem et al were found by Shafaghi [39]. He spent some time applying the existing techniques to chemical processing systems. The Salem et al technique led to problems in the areas outlined above. By clever modelling, Shafaghi was able to obtain two way fault propagation, but the problems associated with protective systems remained.

Three groups of workers have studied these problems over an extended period.

The first in the field were Powers and Tompkins [20] Their work identifies some of the problems specific to chemical plant fault tree synthesis, but cannot, apparently, cope with either two way propagation or protective systems. The methodology differs from all the other automated methodologies. The usual technique employed, for example by Fussell [10,11], is to use some expression for the causes of a particular event, and create a fault tree by linking several of these expressions together. The Powers and Tompkins technique involves creating an information flow structure for the plant. Potential sources where faults may initiate are identified from the models used. For example, a pump is a potential source of high pressure [20]. The information flow structure is used to trace through the plant from the source to the location of interest, noting which failures must also occur to permit propagation from the source to the point of interest. The causes of the enabling events just found are identified in the same way. A number of fault trees that were synthesised with this technique are presented [19-21].

The Powers and Tompkins method has been extended by Powers and Lapp [22] and, in more detail, by Lapp and Powers [23]. The behaviour of the models, including the behaviour when the unit has failed, was incorporated in the information flow structure. The result was a directed graph, or digraph. A computer program was developed to synthesis fault trees directly from digraphs.

Digraphs are an excellent medium for indicating how several items are related. They can be used to model situations where there is no relation, where the relation is conditional on another event or failure, and even when the nature of the relation changes depending on external circumstances. An example of this is the link between the air pressure signal to an air-to-close valve and the flow through the valve [23]. Normally, an increase in the air pressure results in an increase in the flow. If, however, the valve is reversed, then an increased air pressure will result in a decreased flow. Another possibility is that, if the valve is stuck, the flow is unchanged by an increase in the air pressure.

A digraph modelling technique cannot adequately handle the problems presented by protective systems. It will encounter the same problems identified by Shafaghi [39] in his study of the Salem et al methodology. However, Lapp and Powers overcome this problem by according a special treatment to protective systems. A protective system can be deduced from the digraph because such systems form loops in the information flow structure. Using the specific information that protective systems are designed to correct failures elsewhere, Lapp and Powers have grafted onto their technique a special method of synthesising the fault tree when an event is identified as being on a protective system. Two basic types of protective system are noted, feed-forward systems, and feed-back systems. Separate treatments are accorded, as detailed in the paper [23]. A heat exchanger system, incorporating both a feed-back loop and a feed-forward loop, is used as a test example. This example has aroused considerable interest in the technique, and much criticism has been directed at the feed-back loop model used, particularly at an exclusive OR gate [25-30]. Nevertheless, a couple of detailed fault trees utilising the technique have been published. Shaeiwitz, Lapp and Powers [31] extend the digraph models to handle plants that involve sequencing. Cummings, Lapp and Powers [32] present a complex power supply system with several feed-forward loops.

It is unclear from these examples whether the technique can handle two way fault propagation satisfactorily. The digraphs Powers and Lapp [22] and Cummings, Lapp and Powers [32] are in such a form that faults can be propagated in either direction. To obtain digraphs with this feature, all that is required is a link in both directions between two elements in the digraph. For example, inlet flow both affects and is affected by outlet flow. However, this is not the same as two way propagation. There are several problems associated with two way propagation that require careful handling, as described below (see page 2-16).

The second group of workers, based at Loughborough, were the first to appreciate the desirability of two way fault propagation. Andow [33], and Andow and Lees [34] modelled the two way propagation of flow faults by modelling pressure and flow in terms of each other. A high downstream pressure will cause low flow, while a high upstream pressure will cause high flow, and so on. By adopting the convention that outlet flows are modelled using the inlet and outlet pressures, and that inlet pressures are modelled using the inlet and outlet flows, Andow created a two way propagation structure.

The expressions used to relate flow to pressure, and vice versa were derived from full performance equations for flow and pressure, and reduced to a functional form. Functional equations are a concise method of displaying a lot of information. They take the form

$$a=f(b,-c)$$

This equation states that "a" will increase if either "b" increases, or "c" decreases, and that "a" will decrease if "b" decreases, or "c" increases. Four separate relationships are thus contained in a single statement. Functional equations can be derived either from full equations, or from a simple understanding of the unit behaviour. They are therefore easier to understand and to use than digraphs, decision tables or transfer functions. However, they do have the disadvantage that they cannot contain information on failures in the units modelled. Such information must be added in another way.

Andow was more concerned with alarm analysis, but the modelling technique he uses, being a general technique for modelling fault propagation, is applicable to fault tree synthesis.

Martin-Solis [35], and Martin-Solis, Andow and Lees [36] have extended the work of Andow on the alarm analysis side, and applied it to fault tree synthesis. In a paper presented at Heidelberg, Martin-Solis, Andow and Lees [37] discuss one of the problems inherent in the two way propagation method used, namely that modelling outlet flow in terms of inlet and outlet pressure, and inlet pressure in terms of inlet and outlet flow results in a structure that could loop indefinitely. High outlet flow is caused by high inlet pressure, caused by low outlet flow, caused by low inlet pressure, caused by high outlet flow, and so on. Boundary conditions, however, prevent this. Low outlet flow is not a possible cause of high outlet flow, and so can be deleted by the boundary conditions checks.

Further problems arise when basic events are included in the models. In some cases, basic events may violate the boundary conditions. An example of this is high outlet flow from a valve. One possible cause of this is high inlet pressure to the valve. But high inlet pressure will be caused if the valve is shut. Clearly, valve shut is not a realistic cause of high outlet flow. Lees, Andow and Murphy [38] identify this problem and suggest a solution based on not allowed faults. Every event has a list of basic events that are not allowed faults, in other words, faults that cannot be a cause of that event. Valve closed is a not allowed fault of high outlet flow from that valve. Martin-Solis [35] considers these problems in detail.

The work of Shafaghi [39] and Shafaghi, Andow and Lees [40] is a departure from the previous, component based, techniques. Shafaghi notes [39] that protective systems are generally the most important items with regard to fault tree synthesis. He has therefore developed a synthesis methodology based on the control and trip loops in a chemical plant. Each loop is modelled individually, based on a generalised fault tree for protective systems, which takes into account the various modes of failure of protective systems, and how they are related. Some faults, for instance, require failures in the components of the protective system before the failure can propagate through the protective system. Other faults are sufficient to cause the failure to propagate through the protective system.

Shafaghi uses a digraph to link together the various protective systems together to form a representation of the plant under study. Each protective system has one output, which is the variable monitored by that system. These are inputs to the other protective systems on the plant.

Shafaghi, Lees and Andow [41] present the fault tree for a complex plant synthesised using this methodology.

The third group of workers, based at Riso, Denmark have investigated various aspects of risk analysis. Taylor [42] has developed a fault tree synthesis package as part of this study. The modelling technique used is more complex than the modelling techniques used by other researchers in this field. The modelling expressions take the form

INPUT FAULT and NO COMPENSATION gives OUTPUT FAULT

The input and output faults are fairly standard fault expressions. The no compensation fault is more complicated and involves negative logic and tracing of the conditions that will cause compensation. The causes of no compensation are usually latent failures in control loops and trip systems, such as control valve stuck. As a result of this approach, very complicated models are required, and the fault trees produced are very large. For example, there are four variables that can affect the outlet temperature of a heat exchanger, namely the two inlet temperatures, and the two inlet flows. Each deviation of these variables may be compensated by changes in any of the other three variables. The fault tree for a temperature deviation of the outlet stream of a heat exchanger therefore involves sixteen branches.

The benefit of this complex approach is that protective systems require no special treatment. The models contain all the information necessary to synthesise fault trees involving such systems.

Taylor creates a two way propagation structure by using a pair of complex variable names, BACKPR and SUPPR, representing back pressure from some source downstream, and supply pressure from some upstream source. These are analogous to the two variables used by Andow [33]. Taylor uses suitable combinations of deviations of these variables to create a two way propagation structure. For example for a small increase in the flow, the following failure expressions are used

IN becomes DISTHISUPPR and OUT remains
NOCOMPHIBACKPR results in F becoming DISTHI

OUT becomes DISTLOBACKPR and IN remains
NOCOMPLOSUPPR results in F becoming DISTHI

Taylor uses the deviation qualifier DIST (short for disturbed) to represent a small deviation. The above expressions state that there are two causes of a slight increase in flow (F becomes DISTHI). The first is that the upstream pressure rises (IN becomes DISTHISUPPR), but only if the downstream pressure does not become compensatingly high (OUT remains NOCOMPHIBACKPR). The second cause is that the downstream pressure could drop. This can be compensated by decreasing the upstream pressure.

A compendium of examples using this technique has been published [43,44].

A number of other techniques have been proposed to construct logic diagrams for chemical processes. These will be outlined only briefly.

Reina and Squellati [45] propose a method that goes directly from models to minimum cutsets, bypassing the fault tree. The technique is component based, but requires detailed performance equations for the components.

Caceres and Henley [46] and Camarda, Corsi and Trentadue [47] suggest a method for synthesising fault trees based on the reliability graph for the process. This technique is limited to situations where there is a complex series/parallel arrangement of components. The example of Camarda, also investigated by Cummings, Lapp and Powers [32] is a power supply network for a nuclear plant. The power may be supplied by mains, desiel or battery. There is a complex protection system that is designed to activate the various backup systems on demand.

Lihou [48,49] proposes a method for fault tree synthesis based on hazard and operability studies.

The GO methodology, described by Williams et al [50] relies on the creation of a GO chart, similar in many respects to a reliability diagram.

3) Basic Principles

This chapter considers the elementary principles of computer-aided fault tree synthesis. There are three separate steps in modelling a plant, or plant section, as follows

a) Split the plant up into its component models (DECOMPOSITION)

b) Create models for all the units in the plant (MODELLING)

c) Construct the fault tree for a particular top event (SYNTHESIS)

These three steps are examined in detail below.

3.1) Decomposition

Before a plant or plant section can be analysed, it must be turned into a form that is suitable for study. The methodology described in this thesis requires that a 'configuration diagram' be produced. This diagram contains the information on which component models are to be used in the study, and what connections exist between these models. Some extra information, such as data defining the control loops in the plant, may also be required. The configuration diagram is derived from the block flow diagram, the piping and instrumentation diagram, or whatever representation of the plant is to be analysed. This procedure is called decomposition.

Figs 3.1 and 3.2 show the form that the configuration diagram may take. Fig 3.1 is derived from a block flow diagram, whereas Fig 3.2 corresponds to the greater detail found in a piping and instrumentation diagram.

There are two types of item in configuration diagrams. Units, such as reactors and valves, are the physical entities that make up the plant. Connections are the logical links that exist between the units. Note that connections are not pipes. Pipes are units, and are included in the configuration diagram as such (see Fig 3.2).

The units and connections in a configuration diagram are each numbered sequentially. Unit numbers are circled to distinguish them from connection numbers.

Each unit in the configuration diagram must be modelled. Frequently the same model is applicable to more than one unit. There are, for instance, five pipe units in Fig 3.2, but the same model is applicable to each unit. There is therefore only one pipe model. Modelling of units is addressed in Section 3.2.

Dummy Heads are units used to represent sources of process materials. Dummy Tails represent sinks for process flow. They are used solely to provide strictly defined limits on the section under study. The only alternative to starting a section of pipeline with a dummy head, or to terminating it with a dummy tail, is to use a storage tank.

Each unit has a number of interfaces, called 'ports', that may be connected to other units. For example, a pipe unit has two ports, one for flow into the pipe, and one for flow out of the pipe. A heat exchanger has four ports, an inlet and an outlet on both the hot stream and the cold stream. The ports on a unit are also numbered sequentially, and these numbers should appear on the configuration diagram, if confusion between the ports is possible.

The plant diagram of Fig 3.3 will be used to illustrate the decomposition stage. The description of the process is as follows [51].

Two pumps deliver propane at up to 50 te/hr to a 6" pipeline. A control loop operates at very low fluid flows, opening up a kickback line to the tank, to prevent the pumps pumping against no flow. There are three relief valves feeding a common header, which provide for pressure relief back to the storage tank.

There are a number of different ways in which this section could be decomposed. The choice will depend on the detail required in the fault tree, and whether this section is to be studied in isolation, or in conjunction with other plant sections.

Two examples will be discussed, one a full decomposition and the other a minimal decomposition. There will, of course, be intermediate choices as well.

3.1.1) Full Decomposition

A complete decomposition should be used when a detailed study of the plant section is required. A suitable top event for this example might be the pumps overheating and/or overpressuring.

In a full decomposition, each process unit should appear in some way in the fully decomposed configuration diagram. There is unlikely to be a one-to-one correspondence of process items to units for two reasons

a) several models may be required to correspond to a particular process unit.

b) a group of process units may conveniently be represented using a single model.

Several models may be required to correspond to a particular process unit, because is easier to construct models that do not have too much detail. For example, a process stream splitter (divider) may develop faults such as leaks and blockages. However, modelling the performance of such a unit is complex, even if such faults are omitted. An additional problem is that the location of such faults is important. For example, a blockage in the inlet leg of a divider restricts the flow through both outlet legs. However, a blockage in an outlet leg restricts only flow through that outlet leg. The divider models that are in the model library do not model the effects of leaks and blockages. The effects of such faults can, however, be included by specifying that the inlet and both outlets of the divider are connected to pipes, as shown in Fig 3.4. Since the model for a pipe contains these faults, the overall effect is the same as having a single large divider model containing these faults.

On the other hand, several small process units may conveniently be modelled as a single, larger unit. For example, Fig 3.5 shows the simplest representation of a flow sensor - a single unit. Fig 3.6 shows a much more detailed representation, and is built up using an orifice plate, a differential pressure transducer and several valves and pipes. Fig 3.5 is usually sufficiently accurate, but the representation of Fig 3.6 should be used when very fine detail is required.

Fig 3.7 is the configuration diagram of the fully decomposed plant section. Note the use of dummy tails to represent a sink for process flow.

Fig 3.7 includes all of the process equipment items, with the exception of the two level indicators on the tank. The indicators have been omitted because they have no failure modes that affect the plant behaviour, as described above. Presumably, in the actual plant, the indicators are used by the operators, and appropriate action can be taken if the indicators display an incorrect value. This behaviour could be modelled using an indicator model (which may give an incorrect reading), and an operator model, designed to react in different ways to different readings. The operator could be modelled to take the wrong action, including no action when action is required. However, since the action to be taken is undefined, the indicators and operators have been ignored.

Even with this simple example, there are one or two problems in decomposing the plant which cannot be resolved by studying the plant diagram (Fig 3.3). These relate to the setpoints for the three relief valves. Faults in the setpoints of the relief valves may be independent, but it is possible that there is some common cause element. If, for example, the setpoints were set by one maintenance engineer using the same equipment, then a fault in the equipment may cause faults in several setpoints. This problem cannot be answered until information on the plant maintenance procedure is known. Fig 3.7 shows the treatment that would be used on the assumption that the two relief valves on the pumps are set together, but that the third relief valve is done independently. A single setpoint unit is therefore linked to both pump relief valves, and a separate setpoint unit is linked to the third relief valve. The setpoint model, applicable to both setpoint units, is a representation of one form of interaction with the plant. In the context of this model, all that can be done is to fix the setpoint too high, too low or correctly. Note that the same model, but a different unit, is used to model the trip system setpoint.

3.1.2) Minimal Decomposition

The purpose of minimal decomposition is to decompose the plant simply, but without losing the functionality of the plant section. The function of this plant section is to pump propane from a storage tank into the next plant section. The simplest method of representing this is shown in Fig 3.8.

This three model representation does not consider protective equipment (the relief valves, non-return valves and kickback line), redundant equipment (the blocked-off pipe tapping and the open valves), and reduces to a minimum all duplicated equipment (the twin pumps). The functionality - the ability to pump from a tank - is, however, maintained.

This representation is clearly unsuitable for a detailed study, but is ideal for a first study of a large plant area which includes this section. The advantages of a minimal decomposition are that it is far easier to model, meaning that less time need be spent on modelling; and that larger plant sections can more easily be studied.

There are other decompositions that lie between these two extremes. These would be used at different points in the fault tree analysis. The actual decompostion selected depends on the depth and purpose of the analysis.

3.2) Modelling

Decomposition, studied in the previous section, has investigated the breakdown of the plant into smaller elements, typically process units such as pipes and valves. These smaller elements are, as a rule, easier to model than a single, large entity. This section considers the modelling of these smaller elements. It should be borne in mind that the purpose of analysis is the synthesis of fault trees for the complete plant, therefore modelling is orientated towards this purpose. There are three aspects of modelling to consider :-

a) top event modelling
b) propagation modelling
c) spontaneous failure modelling

The top event of a fault tree may be a complex event, such as explosion, requiring a suitable combination of, for example, temperature, pressure and composition. Alternatively, a top event may be a simple event, such as a deviation only of temperature. There is therefore a requirement to model the causes of a particular top event.

Generally, the causes of top events do not occur at the location of the top event. For example, a high temperature in a tank may be caused by faulty operation of a heat exchanger upstream of the tank. An important part of fault tree synthesis is therefore the propagation of faults from where they occur to where the top event occurs. Fault propagation is complicated by the presence of protective systems such as control loops and trip systems. Frequently, a fault will not propagate through a protective system unless there is a fault within the protective system. Control loops are examined in Section 6, and trip systems are considered in Section 7.

Spontaneous failures are the causes of events that propagate and cause the top event. For example, a blockage in a pipe will cause low flow to propagate out of the pipe. Spontaneous failures are defined as faults without cause, or, more accurately, as faults without a cause which requires further clarification. Mechanical failures such as leaks and blockages, and failures of protective equipment are, by convention, regarded as spontaneous failures. To a certain extent, however, there is no such thing as a completely spontaneous failure. Everything, ultimately, has a cause. In chemical plant terms, for example, blockage may result because freezing has occurred. Freezing is caused by a low temperature, the causes of which may require further clarification. Section 8 presents a method to permit the analyst, if he so desires, to model the causes of events that would normally be considered as spontaneous failures.

3.2.1) Modelling of Process Units

The modelling of process units covers the propagation of faults, and the occurrence of spontaneous failures. Top event modelling is done independently of process units, since the causes of a particular top event may be dependent on factors other than the process unit. As was pointed out in the Introduction (Section 1), the process unit models should be made as context-independent as possible, so that the models can be used in as many different situations as possible. Top event modelling is considered in Section 3.2.5.

Faults propagate through units by entering and leaving at the ports of units. For example, in a pipe (see Fig 3.9), a high temperature entering the pipe at its inlet port (port 1) leaves at the outlet port (port 2).

Spontaneous failures are causes of faults that propagate out of models. A spontaneous failure that will cause high temperature to propagate out of a pipe model is an external hot source near the pipe.

A concise way to express this information is that a fault that propagates out of a model has two types of cause, namely faults that propagate into the model; and spontaneous failures that occur within the model. So, high temperature propagates out of a pipe if either high temperature propagates into the pipe, or an external hot source occurs near the pipe. Mini fault trees (or minitrees for short), first introduced by Fussell [10, 11], can be used to express this information graphically. Fig 3.10 is the minitree for high temperature propagating out of a pipe, and conforms to the standards for drawing fault trees [9] (see Fig 3.11). The event 'high inlet temperature' is drawn as a diamond event because it requires further development (what are its causes?), but it cannot be further clarified here. The causes of the event depend on what is upstream of the pipe and could, for example, be a heat exchanger or a reactor.

The model for a process unit comprises one minitree for each fault that can propagate out of that unit. A simple model for a pipe will therefore have minitrees for high and low temperature, composition and flow. A more complex model will consider other flow deviations (such as no flow and reverse flow), the temperature and composition effects of reverse flow (temperature and composition deviations propagate as a result of flow; therefore, reverse flow will change the direction in which such events propagate), and the modelling of pressure.

3.2.2) Fault Types

Faults are described to the synthesis package in one of four ways

a) a variable deviation from a normal value
b) an incorrect (faulty) state of operation
c) an intermediate event
d) a decision table

High temerature, no flow and low pressure are all variable deviations, that is, they represent values that deviate from the normal, expected state. Variable deviations have names of the form VNtyp D. V is the variable mnemonic, N the port number, typ is the port type indicator, and D is the deviation mnemonic. Tables 3.1 to 3.3 are the lists of the recognised variable, port type and deviation mnemonics respectively. No distinction is made between different degrees of low or high - the deviations LO and HI are used for all degrees. Other deviations, such as drifting, have not been included because thay were not found to be necessary in the examples studied. However, since the methodology is model driven, if models containing additional deviations were created, fault trees containing these deviations could be synthesised. Similarly, additional variables could be modelled. An example of a variable deviation is T1in HI, which represents high temperature at port 1, an inlet port. Variable deviaitons are used to model the propagation of faults through process units.

Pump shutdown, valve fails closed, and blockage are all faulty unit states. Such events are called basic events, after Andow et al [33, 34]. There is a standard list of fault names (Table 3.4). An example of a basic event is PART-BLK, representing a partial blockage. Basic events are used to represent events that are normally regarded as spontaneous failures.

Intermediate events are generic fault types that are defined in a particular model. Valve open is an intermediate event, and may be cause by the valve failing open, being directed open, or failing to close. Intermediate events are used solely to structure a model. The names of intermediate events must be in the standard fault list (Table 3.4). Dummy faults, such as A(DUMMY), are frequently used as intermediate events names.

Decision tables are a special type of intermediate event ideally suited to model units that have mixed OR and AND logic minitrees. Decision table names are automatically calculated.

3.2.3) Definition of Models

Three types of information can be used to define a model

a) propagation equations
b) event statements
c) modified decision tables

3.2.3.1) Propagation Equations

Propagation equations are a clear and concise method of representing the relationships between variables. They can be derived either from full performance equations (differential or algebraic), or they can be arrived at heuristically. Propagation equations were first used by Andow et al [33, 34], and take the form

$$a=f(b,-c...)$$

This equation states that if 'b' becomes high, then 'a' also becomes high, but that if 'c' becomes high, then 'a' becomes low, and so on. A minus sign is used when a deviation of a variable in one direction causes a deviation of the output in the other direction; no minus sign indicates that the deviations occur in the same direction.

The variable on the left hand side of an equation must be a variable that occurs within or propagates out of the model. Except in two special circumstances, the variables on the right hand side must be variables that propagate into the model. One exception is the modelling of flow, considered in Section 3.2.4. The other exception is that the right hand side may contain a single variable that propagates out of the model, provided that this is the only item on the right hand side of the equation. The reason for the restrictions is to prevent confusion about the direction of the deviations that propagate into the model necessary to cause the output deviation. This point will be returned to later.

Continuing the example of Section 3.2.1 (pipe), a suitable propagation equation for the outlet temperature is

T2out=f(T1in)

The propagation equation says that the outlet temperature is directly (as opposed to inversely) related to the inlet temperature.

From this propagation equation, two minitrees may be drawn (see Fig 3.12), one for T2out LO and one for T2out HI.

3.2.3.2) Event Statements

Martin-Solis et al [35,36,38] point out that propagation equations are unable to include all the failure information required. Event statements, introduced by Martin-Solis [35], are used to model how basic events in units affect the variables that propagate out of the units, and to include the effects of variable deviations that cannot be included using propagation equations. Event statements can also be used in certain circumstances to include AND gates and r/n gates in models. Event statements take the forms

t cause:effect1,effect2,...
t cause1 AND cause2...:effect1,effect2,...
t cause1 ANDr cause2...:effect1,effect2,...

t is used to identify the cause type of each cause in the event statement. For example, 'V' is used to denote a variable deviation and 'I' an intermediate event. Table 3.5 is a list of the cause types and the faults they represent. The second event statement shows how an AND gate is included. The ANDr in the third event statement is an r/n gate, with the r being the number of causes that must exist for the event to be caused. n is implicitly derived from the number of causes in the event statement.

Event statements are used in a pipe to include the effects of external heat sources

```
F EXT-HEAT:T2out HI
F EXT-COLD:T2out LO
```

EXT-HEAT and EXT-COLD are in the standard fault list (Table 3.4), and represent, respectively, an external hot source, and an external cold source.

Resolving event statements into minitrees is, in most cases, very straightforward. When the event statements above are combined with the propagation equation for the pipe given in the previous section, the two minitrees of Figs 3.13 are generated.

Event statements can be used in situations where the model requires AND or r/n logic in minitrees. They are not suitable for use where combined AND and OR logic is required - decision tables (see the next section) should be used in such cases. An example of an event statement involving an AND gate is high temperature propagating out of a valve that is normally shut. Fig. 3.14 is a diagram of this unit. Not only does high temperature need to propagate into the valve, the valve must also be open. An appropriate event statement is

```
V T1in HI AND I OPEN:T2out HI
```

OPEN is an intermediate event, and therefore has a cause type identifier of I. Its causes are all the ways in which the valve could be open. It could, for example, fail open, or an operator could have opened it. The causes of OPEN can be expressed using event statements

F HV-F-OP:OPEN
O HV-D-OP:OPEN

HV-F-OP represents the valve failing open, and HV-D-OP represents the operator directing the valve open. The former event is a basic event, and its cause type identifier is therefore F. The latter is an operator action, and has a cause type indicator of O. The minitrees that result from these event statements are shown in Fig 3.15.

It would not be correct to model this behaviour using the two event statements below

V T1in HI AND F HV-F-OP:T2out HI
V T1in HI AND F HV-D-OP:T2out HI

The way in which event statements are resolved, involves simply replacing the default OR logic of a minitree which the specified logic (AND above). The above two event statements are equivalent to

V T1in HI AND F HV-F-OP AND F HV-D-OP:T2out HI

The minitree that corresponds to this event statement is shown in Fig 3.16, and is clearly an inappropriate model. What is required is a combination of AND and OR logic. This can be provided either using an intermediate event (OPEN above), or by using decision tables, as described in the next section.

3.2.3.3) Modified Decision Tables

Mixed OR and AND logic is most efficiently included in models using a decision table format, modified from the traditional form, as used, for example, by Salem et al [15-18]. The modified format used is

t cause1 t cause2... T effect1,effect2,...

The cause types are the same as used in event statements (see Table 3.5); T is used to delimit the causes and effects. All the events given on the left hand side of the decision table are ANDed together. Events not specified on the left hand side are regarded as "don't care" events.

Decision tables can be used to model the propagation of high temperature out of the normally shut valve. Appropriate tables are

V T1in HI F HV-F-OP T T2out HI
V T1in HI F HV-D-OP T T2out HI

The minitrees generated from these decision tables are given in Fig 3.17. Compare these minitrees with those generated using event statements and intermediate events (Fig 3.15). The minitrees are logically the same, illustrating that decision tables are simply a special type of intermediate event.

3.2.4) The Modelling of Flow

Flow is by far the most important variable in modelling. Fault trees of process plants are almost certain to involve flow because the presence of control loops ensures that deviations in flow will have effects on other variables.

The effects of flow faults propagate both upstream and downstream from the location of the fault. For example, a leak in a pipe will result in an increased flow into the pipe, but in a reduced flow out of the pipe. As can be seen by studying the temperature example presented earlier, one variable can propagate failures in only one direction.

Andow et al [33, 34] overcame this problem by relating flow and absolute pressure. Flow was defined as a variable that propagated out of outlet ports, and pressure as a variable that propagated out of inlet ports. The following propagation equations were used to model this behaviour

$$Q2out=f(P1in,-P2out)$$
$$P1in=f(Q1in,-Q2out)$$

This relationship is adequate until the causes of other flow deviations (none, some and reverse flow) are considered. For example, one cause of low flow is high downstream pressure; slightly higher downstream pressure will cause no flow, and even higher downstream pressure will cause reverse flow. Some flow will exist if the upstream pressure is higher than the downstream pressure, regardless of the absolute levels. Relating flow to pressure therefore requires the ability to distinguish between these various pressure levels.

While this would certainly be possible, it could lead to confusion. An alternative approach has therefore been adopted in this thesis.

Flow and pressure gradient are more closely related than flow and absolute pressure. Consider

a) high flow is accompanied by a high pressure gradient
b) low flow is accompanied by a low pressure gradient
c) no flow is accompanied by no pressure gradient
d) reverse flow is accompanied by a reversed pressure gradient
e) some flow is accompanied by some pressure gradient

An equation relating flow and pressure gradient [52] is

$$-P_f/(\rho g) = 4f\ (l/d)\ (u^2/2g)$$

where P_f is the pressure drop, f the Fanning friction factor, l the pipe length and d its diameter, ρ the density of the fluid, u its velocity and g the acceleration due to gravity. The pressure gradient is simply $-P_f/l$; for a particular pipe, flow is proportional to the velocity. Therefore, two propagation equations can be derived from the above equation

$$G = f(Q)$$
$$Q = f(G)$$

G being the pressure gradient and Q the flow.

Normally, it is incorrect to derive two propagation equations from one algebraic equation. However, since both the propagation equations are used solely to model flow, the algebraic equation is effectively being used only to calculate flow.

Neither the propagation equations nor the algebraic equation identifies the ports from which flow and pressure gradient values must be taken. At steady state, inlet flow equals outlet flow, and the pressure gradient is uniform. Therefore, the above propagation equations are appropriate for any combination of inlet and outlet ports.

Adapting the convention introduced by Andow and defining pressure gradient as a variable that propagates out of inlet ports, and flow as a variable that propagates out of outlet ports, the following propagation equations can be derived from the above equation

Q2out=f(G1in,G2out)
G1in=f(Q1in,Q2out)

To trace the causes of flow faults in one particular direction only half of the information contained in these equations is necessary, as illustrated by Figs 3.18 and 3.19.

Note that the right hand side of these equations involve variables that propagate out of the model. The reason for this exception to the restrictions on the terms that may appear on the right hand sides of propagation equations noted in Section 3.2.3.1 is the requirement to have flow faults propagating both upstream and downstream.

Pipes may also develop mechanical faults, such as leaks and blockages, and the effects of these must be included using event statements. Calculating the effect of, for example, a partial blockage on the inlet pressure gradient may at first sight appear difficult, but in fact inlet pressure gradient is synonomous with inlet flow. All that need be done, therefore, is to calculate the effect on the inlet flow, and the effect on inlet pressure gradient will be identical.

The event statements are

```
F LK-LP-EN:G1in HI,Q2out LO
F LK-HP-EN:G1in LO,Q2out HI
F PART-BLK:G1in LO,Q2out LO
```

In plain language, these state that leak to a low pressure environment will result in low outlet flow, but high inlet flow, whereas leak from a high pressure environment will have the opposite effects. A partial blockage results in low inlet and in low outlet flows.

3.2.5) Top Event Modelling

The purpose of top event modelling is to relate how one or more variable deviations may combine to cause some undesired event to occur. Top event modelling is done independently of the modelling of process units for two reasons

a) there is a large number of top events, and including each of these in every model would be time consuming and could obscure the other aspects of modelling

b) the causes of some top events, particularly the more complex events such as explosion, vary from plant to plant. Since one of the advantages of a component-based methodology is to provide a library of standard models which can be used in many different situations, the inclusion of situation-dependent information in models is not appropriate

A top event model is created in terms of event statements and decision tables, as described in Sections 3.2.3.2 and 3.2.3.3 for creating unit models. Propagation equations are not of use in modelling top events.

A simple top event is high temperature in a pipe. Represented by OVRTEMP (over temperature), its sole cause can be modelled using an event statement

```
V T2out HI:OVRTEMP
```

The minitree that is derived from this statement is shown in Fig 3.20.

Note that a variable that propagates out of a model is used as a cause of the top event. All top events should be expressed in terms of such events, to ensure that a minitree in the model where the top event occurs appears at the top of the fault tree. If this restriction is not adhered to, then the fault tree may more accurately be for, for example, high temperature inlet to a heat exchanger rather than high temperature in the exchanger itself. The difference between these events is significant. High temperature in a heat exchanger may be caused not only by a high inlet temperature but also, for example, by a low flow of cooling medium.

The port numbers in the top event model must correspond to the port numbers in the model where the top event exists. This top event above is applicable to all models where the event T2out HI is of interest. In the pump model presented in Section 3.4.2.3, the top event represents high outlet temperature, as in the pipe. In the heat exchanger model presented in Section 3.4.2.2, the top event represents high outlet temperature of the hot fluid. A different top event model must be used for high temperature of the cold fluid, with cause T4out HI.

3.3) Fault Tree Synthesis

The information contained in minitrees is ideally suited to synthesising fault trees. In principle, a fault tree can be synthesised by linking together the appropriate minitrees, since the events that propagate into one model are simply the events that propagate out of some other model. In practice, however, there are complications, as an example will show. Consider three pipes linked together as shown in the configuration diagram of Fig 3.21. Note that each connection and each unit has been numbered. These numbers will be used during the synthesis process.

Consider the causes of the event Q3 LO, or low flow out of pipe number 3. The causes of this are defined by the minitree for Q2out LO for the pipe model. The causes of this, expressed in relation to the example as a whole, rather than to the pipe model are

a) G2 LO
b) G3 LO
c) PART-BLK Unit 3 (partial blockage)
d) LK-LP-EN Unit 3 (leak to a low pressure environment)

Causes c) and d) above are regarded as spontaneous events, and need not be considered further. However, minitrees exist for G2 LO and G3 LO, which define the causes of these two events. G2 LO is G1in LO to pipe number 3, and G3 LO is G1in LO to pipe number 4. The causes of these are similar, differing only in the connection and unit numbers. Adding the minitrees for these events to the minitree for Q3 LO results in the fault tree of Fig 3.22.

Careful examination of this figure indicates several duplicated events and one event that is inconsistent with the other events in the fault tree. The duplicated events, all at the lowest level of the fault tree are Q3 LO (twice) and PART-BLK Unit 3 (only once). Exploring the causes of these events is pointless, since they can never add definition to the fault tree, and it would therefore be sensible to remove such events from the fault tree.

In addition to these duplicated events, there is also an inconsistent event in the fault tree - LK-HP-EN Unit 3 (leak from a high pressure environment). This fault will tend to increase the flow out of the pipe, but the top event of the fault tree is low flow out of the pipe. LK-HP-EN Unit 4 is not, therefore, a realistic cause of G2 LO, and should therefore be removed. Fig 3.23 is the fault tree with the inconsistent and duplicated events removed. Note that, although LK-HP-EN Unit 3 is inconsistent with the top event, LK-HP-EN Unit 4 is not inconsistent, and remains in the fault tree.

These unnecessary events occur in the fault tree because of the two way fault propagation facility - at each point during synthesis, every flow fault is propagated in both directions. Except for flow faults at the top of fault tree branches, two way propagation will inevitably lead to consideration of an area that has already been analysed.

To overcome these problems, there are a series of internal consistency checks that were developed to ensure that the fault trees generated by the methodology contain events that are neither inconsistent with the fault tree, nor that add nothing to the clarification of the causes of the top event.

3.4) Fault Tree Synthesis - A Simple Example

This section considers a simple plant section and demonstrates the three main features of the methodology. The plant section is shown in Fig 3.24, is based on a heat exchanger system presented by Lapp and Powers [23].

3.4.1) Decomposition

Decomposition is not a problem in this case. The plant section is so simple that full decompostion is the only choice. The only point is that both water and nitric acid streams must originate from a dummy head and terminate in a dummy tail. The purpose of these dummy units is simply to enable a connection number to be associated with all process unit ports. Fig 3.25 is the configuration diagram for this system.

3.4.2) Unit Modelling

Models are required for three units in this example

a) a pipe unit
b) a heat exchanger unit
c) a pump unit

Simple models for these units are generated below.

3.4.2.1) Pipe

The pipe model has been used as an example extensively throughout this chapter. With the exception of composition, all the necessary propagation equations and event statements have already been considered. Composition itself is straightforward to model - it has propagation equations similar to those for temperature, and no event statements. A simple model for a pipe is, therefore

```
G1in=f(Q1in,Q2out)
Q2out=f(G1in,G2out)
T2out=f(T1in)
X2out=f(X1in)

F LK-LP-EN:G1in HI,Q2out LO
F LK-HP-EN:G1in LO,Q2out HI
F PART-BLK:G1in LO,Q2out LO
F EXT-HEAT:T2out HI
F EXT-COLD:T2out LO
```

3.4.2.2) Counter-Current Heat Exchanger

See Fig 3.26. Port 1 is the inlet port for the hot fluid, port 2 the outlet port for hot fluid, port 3 the inlet port for the coolant, and port 4 the outlet port for the coolant.

The heat exchanger can be regarded as two separate pipes, each carrying fluid. The propagation equations for flow and pressure gradient are therefore based on the propagation equations for the pipe model, and are

G1in=f(Q1in,Q2out)
Q2out=f(G1in,G2out)
G3in=f(Q3in,Q4out)
Q4out=f(G3in,G4out)

The leaks and blockages included in the pipe model will not be included in this model, because this could lead to confusion about which flow stream was affected by the faults. One flow fault that could be included is internal leak (leak between the two flow streams). In this simple model, this will be ignored.

The propagation equations for composition are similarly based on the propagation equation for the pipe model

X2out=f(X1in)
X4out=f(X3in)

Internal leak may affect the compositions of the streams. For simplicity, as was mentioned above, this will be ignored.

The modelling of temperature is, however, more complicated. The temperature in the exchanger may be calculated from the heat balance.

The heat balance equation for the hot stream is

$$E = Q1in*Cp1*(T1in - T2out)$$

E is the heat transfer rate and Cp1 the heat capacity of the hot stream. The other variables are the model variables.

The heat balance equation for the coolant is

$$E = Q3in*Cp3*(T4out - T3in)$$

Cp3 is the heat capacity of the coolant.

So, equating these two equations, and rearranging

$$T4out = T3in + \frac{Q1in*Cp1*(T1in - T2out)}{Q3in*Cp3}$$

It is not possible to resolve this equation into a propagation equation as it stands. The reason for this is the occurrence of T2out in the equation. Changes to, for instance, T3in will affect T2out as well as T4out. Until the effect of changing T3in on the value of T2out is known, it will be impossible to determine its effect on T4out.

This illustrates the reason for the restriction (see Section 3.2.3.1) that events on the right hand side of a propagation equation should (with two exceptions) be variables that propagate into the model. One exception to this rule is flow, noted in Section 3.2.4. The other exception is that the right hand side may contain an event that propagates out of a model, provided that there is only one event on the right hand side. In this circumstance, there is no possible confusion in determining how variables propagating into the model affect the events that propagate out.

To replace T2out with T1in would not be correct, since the two variables are different. An equation for T2out must be derived, and this can be used to replace T2out in the equation. Furthermore, the heat balance equation used earlier cannot be used to generate an expression for T2out, since the same equation would then be used to provide two different variables. The effect of this, mathematically, is to have a series of unsolvable equations.

The heat transfer equation provides another way of calculating T2out. This equation is

$$E = U*A*\Delta T_{LM}$$

In this equation, U is the overall heat transfer coefficient, A is the heat transfer area, and ΔT_{LM} the log mean temerature difference, approximately given by the average temperature difference in the exchanger

$$\Delta T_{LM} \simeq 0.5*((T1in-T4out) + (T2out-T3in))$$

Equating this with the heat balance on the hot stream gives

$$U*A*((T1in-T4out) + (T2out-T3in)) = 2*Q1in*Cp1*(T1in-T2out)$$

Rearranging this equation

$$T2out = \frac{(2*Q1in*Cp1 - U*A)*T1in + (T4out+T3in)*U*A}{(2*Q1in*Cp1 + U*A)}$$

Substituting for T2out in the equation for T4out and rearranging gives

$$T4out = \frac{2*U*A*Q1in*Cp1*T1in + 2*Q1in*Cp1*Q3in*Cp3*T3in + U*A*Q3in*Cp3*T3in - U*A*Q1in*Cp1*T3in}{2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 + Q1in*Cp1*U*A}$$

Calculating the effects that the events that propagate into the exchanger have on T4out requires this equation to be differentiated with respect to each of Q1in, Q3in, T1in and T3in in turn, and seeing if the differential is positive or negative. If the differential is positive, then an increase in that variable will result in an increase T4out; if the differential is negative, increasing that variable will decrease T4out. From this information the propagation equation for T4out can be written. Because of the complexity of the differentiations, the calculations are not presented here, but are given in Appendix A. The propagation equation that results is

$$T4out=f(T1in,Q1in,T3in,-Q3in)$$

This equation requires one further change before it can be used to model the behaviour of a heat exchanger. The flow terms (Q1in and Q3in) should be variables that propagate out of the exchanger. The reason for this is similar to the reason why events that propagate out of a model are used to model top events (see Section

3.2.5), namely to ensure that the fault tree synthesised is for flow in the heat exchanger, as oppose to flow into the exchanger. In this example, these two events are identical, and information would not be lost by using Q1in. Nevertheless, there are situations in which the events are different. Q1in could be replaced by either G1in or by Q2out, these two events being the same since there are no flow faults that can occur within the exchanger. In this example, G1in will be selected. Similarly, Q3in could be replaced by either G3in or by Q4out. G3in has been used here.

The final propagation equation for T4out is, therefore

T4out=f(T1in,G1in,T3in,-G3in)

A similar process is required to find the propagation equation for T2out. It transpires that the propagation equation for T2out is identical to the equation for T4out, namely

T2out=f(T1in,G1in,T3in,-G3in)

These equations also correspond to a heuristic model of a heat exchanger. Either outlet temperature will increase if

a) either inlet temperature increases
b) the coolant flow (G3in) decreases
c) the heating fluid flow (G1in) increases

A number of faults can be added to these equations, depending on the type of heat exchanger. Vapour blanketing in a condenser and frothing in a reboiler will both have effects on the rate of heat transfer. In this simple example, fouling of the exchanger is assumed to be the only applicable fault. Its effects are high outlet temperature of the hot stream, and low outlet temperature of the coolant. The event statement to model this behaviour is

F FOULING:T2out HI,T4out LO

The heat effects of a pipe can also be included. Unlike the flow faults, there is no confusion about which ports are affected - the heat effects affect both streams.

F EXT-HEAT:T2out HI,T4out HI
F EXT-COLD:T2out LO,T4out LO

From these propagation equations and event statements, four minitrees can be drawn, one each for T2out LO, T2out HI, T4out LO, and T4out HI. These are shown in Fig 3.27. As this figure shows, propagation equations are a very compact method of storing information.

3.4.2.3) Centrifugal Pump

See Fig 3.28. Port 1 is the inlet, and port 2 the outlet.

Heuristically, the pressure change across a pump increases as the flow through the pump decreases. Since the pressure at the outlet of a pump is higher than at the inlet, this pressure increase corresponds to a pressure gradient drop (remember that pressure gradient is defined as inlet pressure minus outlet pressure - see Section 3.2.4). Again heuristically, the flow through a centrifugal pump increases as the pressure difference across the pump decreases (i.e. the pressure gradient increases). Therefore, although the absolute pressures differ significantly from those in a pipe, this is the same functional behaviour as exhibited by a pipe. The propagation equations that model the flow and pressure gradient through a pump are therefore the same as the propagation equations that model the behaviour of a pump, namely

G1in=f(Q1in,Q2out)
Q2out=f(G1in,G2out)

A pump has many faults that affect the flow. A group of these faults all have the same effects as a partial blockage, and will be grouped under the intermediate event A(DUMMY). The event statements needed are

F LK-LP-EN:G1in HI,Q2out LO
F LK-HP-EN:G1in LO,Q2out HI
I A(DUMMY):G1in LO,Q2out LO
F PART-BLK:A(DUMMY)
F IMPLR-F:A(DUMMY)
F AIR-LOCK:A(DUMMY)
F CAVITATN:A(DUMMY)

IMPLR-F represents an impeller failure, AIR-LOCK an air pocket in the pump, and CAVITATN the pump cavitating. Fig 3.29 displays the minitrees associated with flow through a centrifugal pump.

The modelling of temperature and composition is identical to the modelling of a pipe.

T2out=f(T1in)
X2out=f(X1in)

F EXT-HEAT:T2out HI
F EXT-COLD:T2out LO

More complex models could include the possibility that high temperature occurs because the pump is switched on, but there is no flow through the pump, and that cavitation occurs because of high temperature and/or low pressure.

3.4.3) Top Event

The top event of this example is OVRTEMP Unit 4, representing a high outlet temperature from the plant section. The model for this top event has been given in Section 3.2.5, but is repeated here for convenience

V T2out HI:OVRTEMP

3.4.4) Fault Tree Synthesis

The top event is OVRTEMP Unit 4. The sole cause of this, derived from the top event model, is T2out HI of unit 4, a pipe. This corresponds to T4 HI in the fault tree. The causes of T4 HI, or T2out HI of a pipe, are given by the appropriate minitree in the pipe model, which has causes T1in HI or EXT-HEAT. In fault tree terms, these are T3 HI and EXT-HEAT Unit 4.

T3 HI is the outlet temperature of the hot stream of the heat exchanger, and so the minitree for this event is located in the heat exchanger model. Since the hot stream outlet is port 2 of this model, the minitree for T2out HI in the heat exchanger model is the correct minitree. The causes of this are T1in HI (T2 HI in the fault tree), T3in HI (T7 HI), G1in HI (G2 HI), G3in LO (G7 LO), FOULING (FOULING Unit 3), or EXT-HEAT (EXT-HEAT Unit 3).

T2 HI is the outlet temperature of a pipe. Its causes are high inlet temperature (in this case T1 HI) and EXT-HEAT (Unit 2). T1 HI is associated with a dummy head (Unit 1), and so is drawn as a diamond event. This is used to indicate an event that requires further clarification, but is not expanded here, because it crosses the plant section boundary.

T7 HI can be synthesised similarly, until the fault propagates to another dummy head unit (Unit 5).

The fault tree at this stage in its development is displayed in Fig 3.30.

G2 HI is the inlet pressure gradient to the heat exchanger, and is defined by the minitree for G1in HI in this model. There are two causes of this, Q1in HI (Q2 HI) and Q2out HI (Q3 HI).

Q2 HI is the outlet flow of Unit 2, a pipe. It has causes G1in HI (G1 HI), G2out HI (G2 HI), or LK-HP-EN (Unit 2). Now G2 HI has already occured in the fault tree directly above this event, and including the event in the fault tree again will not add any new failure paths. The current cause G2 HI (not the earlier G2 HI) is identified as violating the internal consistency checks, and so is removed from the tree.

With G2 HI removed from the fault tree, the only cause that requires development is G1 HI, or, more pertinently, G1in HI of the pipe model. The causes of this are Q1in HI (or Q1 HI), Q2out HI (Q2 HI), or LK-LP-EN (Unit 2). Q1 HI is associated with a dummy unit, and so is drawn as a diamond event. Q2 HI has already appeared earlier in the current branch of the tree, and so should not be drawn here. LK-LP-EN is clearly not consistent with the current branch of the tree, since it is not a potential cause of Q2 HI, namely high outlet flow from the pipe.

The branch under G2 HI has traced the upstream causes of high flow of nitric acid. The downstream causes of high flow are found by considering the causes of Q3 HI. The synthesis of this event is similar to the synthesis of Q2 HI. Note that the internal consistency checks force the causes always to be downstream.

The only remaining branch to be synthesised is for the event G7 LO, that is, low flow of the coolant. The synthesis of the causes of this event presents no new problems. The fault tree generated by the methodology is displayed in Fig 3.31.

Variable Mnemonic	Description
G	Pressure Gradient
Q	Flow
T	Temperature
X	Composition
P	Absolute Pressure
R	Relief
L	Level
S	Signal
W	Setpoint
U	Temperature under Reverse Flow Conditions
Y	Composition under Reverse Flow Conditions

Table 3.1 - the list of recognised variables

Port Type	Description
in	where flow into the unit exists
out	where flow out of the unit exists
sig	where a signal is received or output
ves	an internal port, frequently associated with the measurement of level in vessel units
utl	a utility, such as instrument air or electrical power enters or leaves the unit at this point

Table 3.2 - the list of recognised port types

Deviation Mnemonic	Description
HI	the value of the variable is higher than expected
LO	the value of the variable is lower than expected
SOME	the variable has a finite positive value
NONE	the variable has a negligible value
REV	the variable has a negative, or reversed value
NCHA	the variable has not changed, when a change is expected
NOP	a special variable (see Chapter 4.1.3)
NOR	a special variable (see Chapter 4.1.3)

Table 3.3 - the list of recognised deviations

Fault Mnemonic	Description
LK-LP-EN	leak to low pressure environment
LK-HP-EN	leak from high pressure environment
PART-BLK	partial blockage
COMP-BLK	complete blockage
EXT-COLD	external cold source
EXT-HEAT	external hot source
HV-D-SH	hand valve directed shut
HV-D-OP	hand valve directed open
HV-F-SH	hand valve fails shut
HV-F-OP	hand valve fails open
CV-F-HA	control valve fails giving high aperture
CV-F-LA	control valve fails giving low aperture
CV-STK	control valve stuck
CV-F-SH	control valve fails shut
TV-F-SH	trip valve fails shut
TV-F-OP	trip valve fails open
TV-FT-SH	trip valve fails to shut
TV-FT-OP	trip valve fails to open
RV-F-OP	relief valve fails open
RV-FT-OP	relief valve fails to open
RV-UNDSZ	relief valve undersized
SEN-F-HI	sensor fails high
SEN-F-LO	sensor fails low
SEN-STK	sensor stuck

Table 3.4 - list of recognised faults

Fault Mnemonic	Description
CNT-F-HI	controller fails high
CNT-F-LO	controller fails low
CNT-STK	controller stuck
CNT-MAN	controller on manual
SET-P-HI	setpoint too high
SET-P-LO	setpoint too low
TSW-F-ON	trip switch fails on
TSW-F-OF	trip switch fails off
TSW-STK	trip switch stuck
TSW-DIS	trip switch disarmed
SIG-PB	signal line partially blocked
SIG-CB	signal line completely blocked
STARTUP	pump started up
SHUTDOWN	pump shutdown
FT-ST-UP	pump fails to start up
RACING	pump racing
IMPLR-F	impeller failure in pump
AIR-LOCK	pump has an air lock
CAVITATN	pump is cavitating
FLOODING	condenser is flooded
VAP-BLKT	reboiler is blanketed with vapour
FROTHING	reboiler is frothing
INT-LK	heat exchanger has an internal leak
FOULING	heat exchanger has fouling
CAT-DETN	catalyst deactivation
POOR-MIX	poor mixing

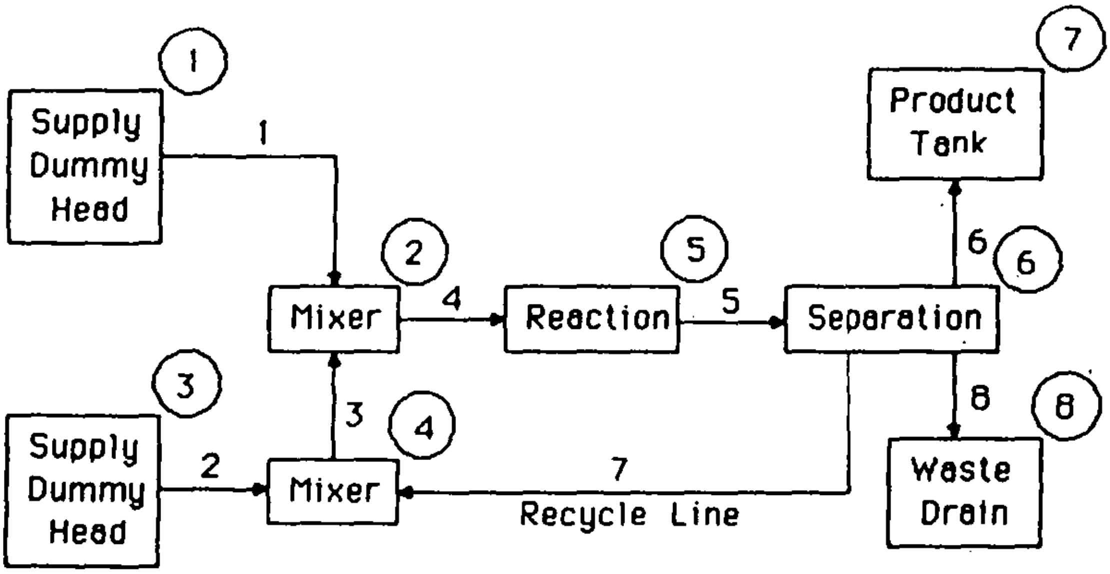
Table 3.4 - list of recognised faults (cont)

Fault Mnemonic	Description
IAR-LOSS	instrument air loss
POW-LOSS	electrical power loss
NORMAL	normal state
IMPOSS	impossible state
A(DUMMY)	unnamed intermediate event
B(DUMMY)	unnamed intermediate event
C(DUMMY)	unnamed intermediate event
D(DUMMY)	unnamed intermediate event
E(DUMMY)	unnamed intermediate event
F(DUMMY)	unnamed intermediate event
CL-F-HA	control loop fails giving low aperture
CL-F-LA	control loop fails giving high aperture
CL-F-NA	control loop fails giving no aperture
CL-STK	control loop stuck
TL-FN-F	trip loop functional failure
TL-OR-F	trip loop operational failure
SEQ-F-AT	sequence fails at
SEQ-F-AF	sequence fails after

Table 3.4 - list of recognised faults (cont)

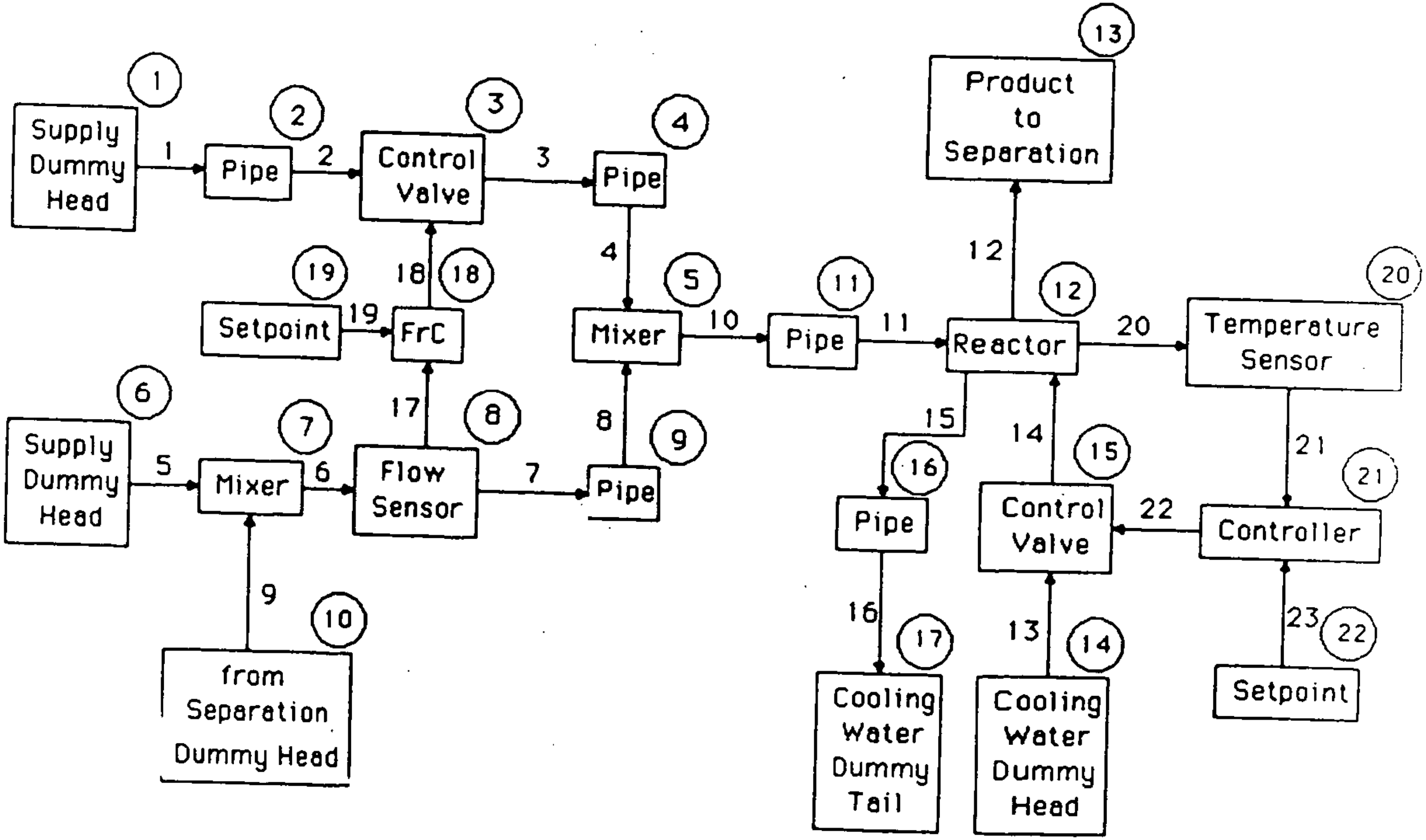
Variable Type	Description
V	variable deviation
F	spontaneous failure, or basic event
I	intermediate event
O	operator action/inaction
S	state (normal or impossible)

Table 3.5 - the list of recognised variable types



① unit number
1 connection number

Figure 3.1 - configuration diagram based on a Block Flow Diagram



① unit number
1 connection number

Figure 3.2 - configuration diagram based on a Piping and Instrumentation Diagram

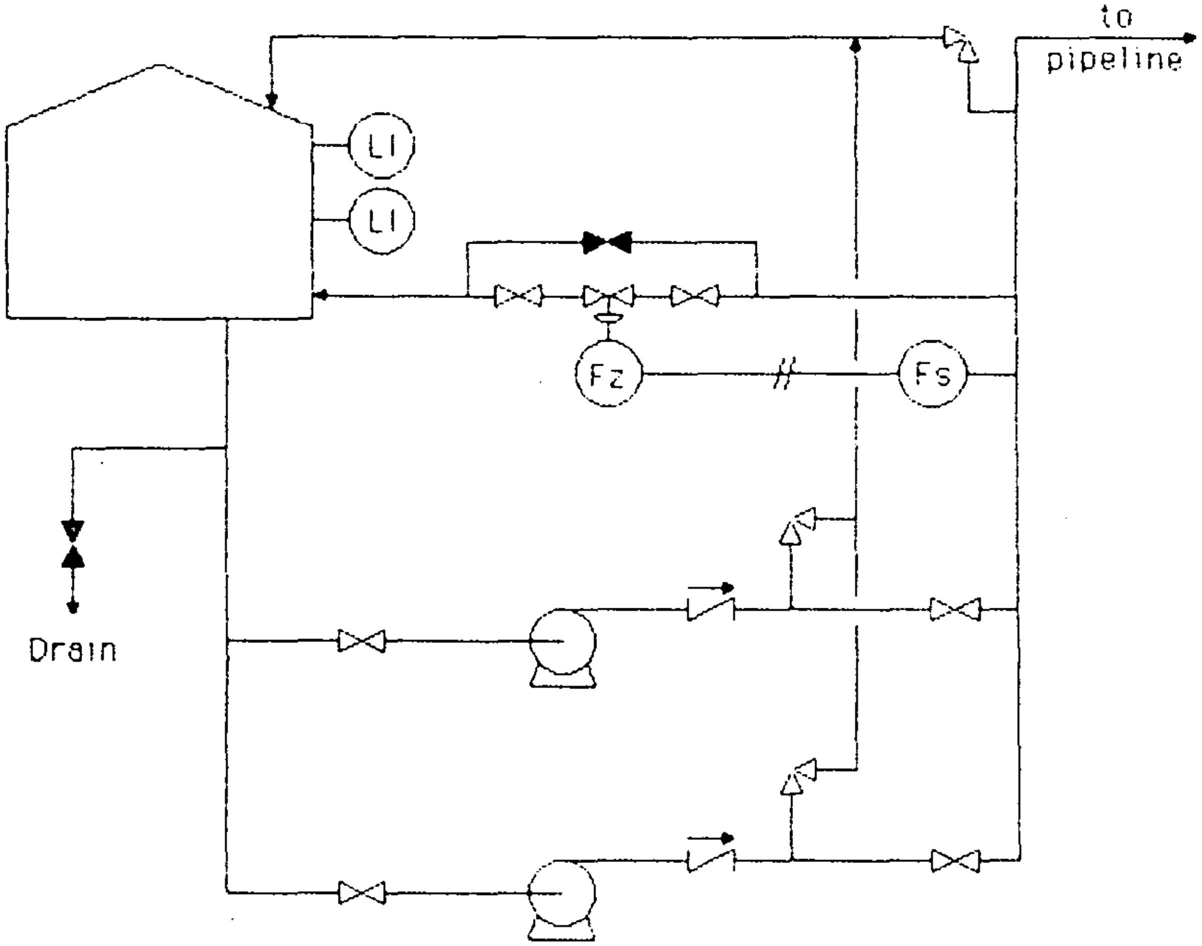


Figure 3.3 - a propane storage tank,
and pump protection system,
after Lawley [51]

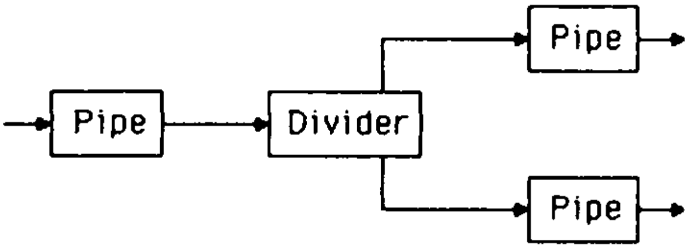


Figure 3.4 - one method of incorporating pipe-type faults in a divider model

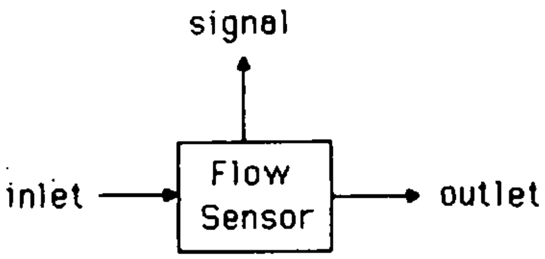


Figure 3.5 - a minimal decomposition of a flow sensor

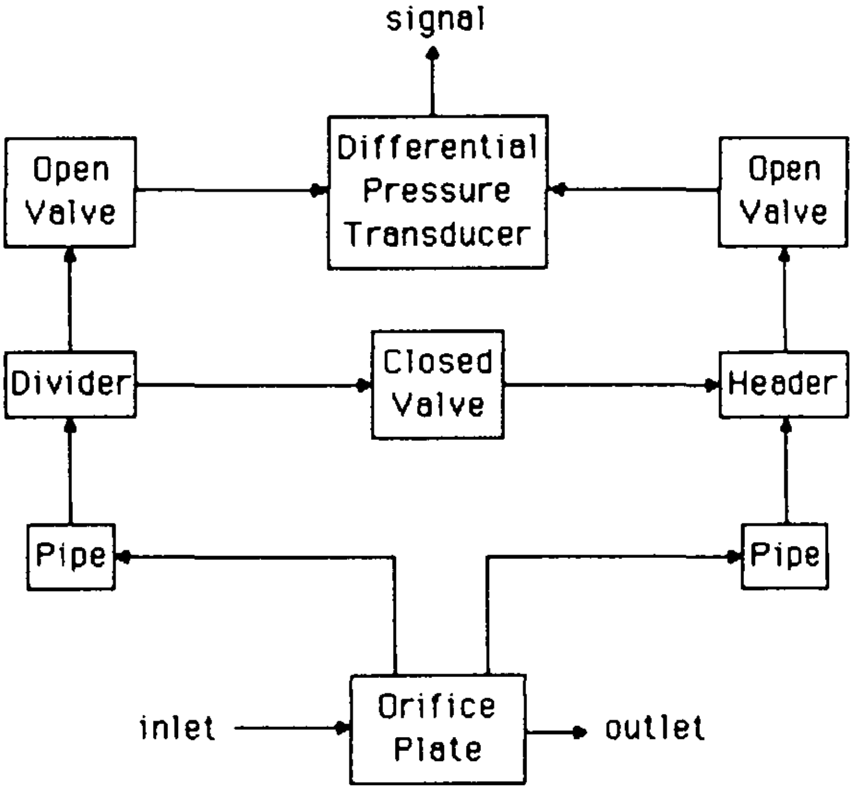


Figure 3.6 - a full decomposition of a flow sensor

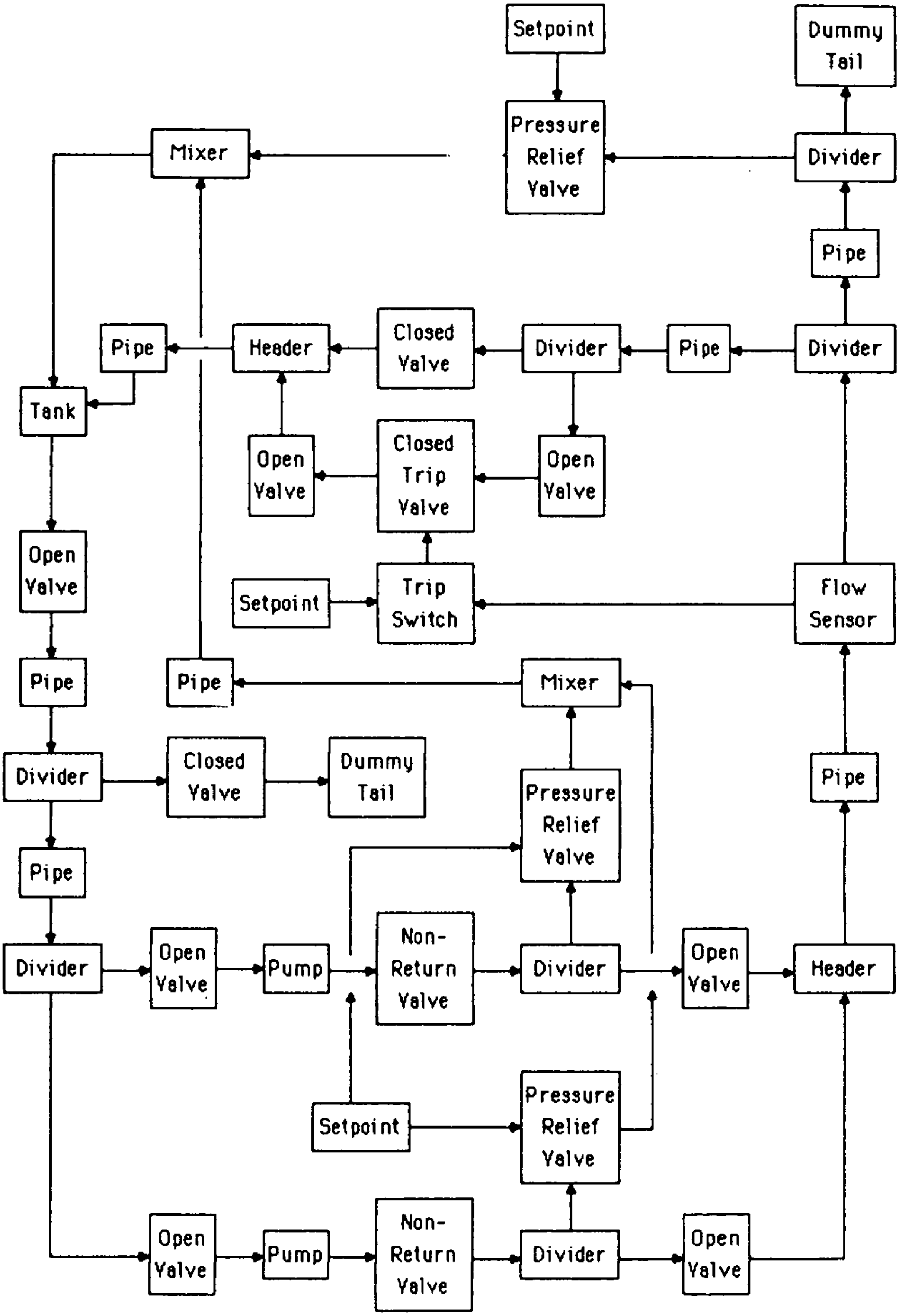


Figure 3.7 - a full decomposition of the system shown in Figure 3.3

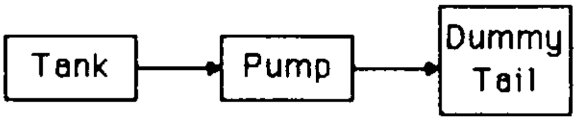


Figure 3.8 - a minimal decomposition of the system shown in Figure 3.3

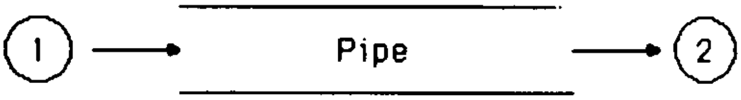


Figure 3.9 - the representation of a pipe

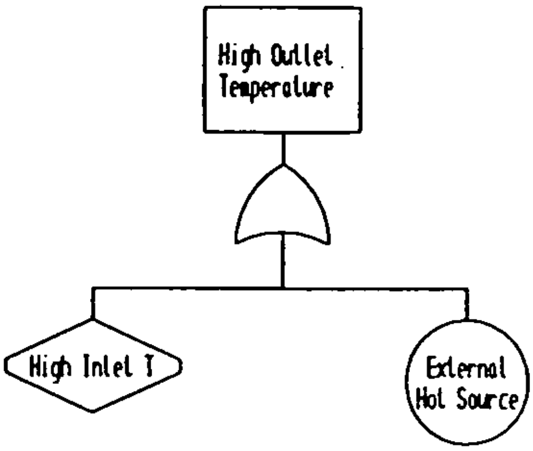


Figure 3.10 - the minitree for high temperature at the outlet of a pipe

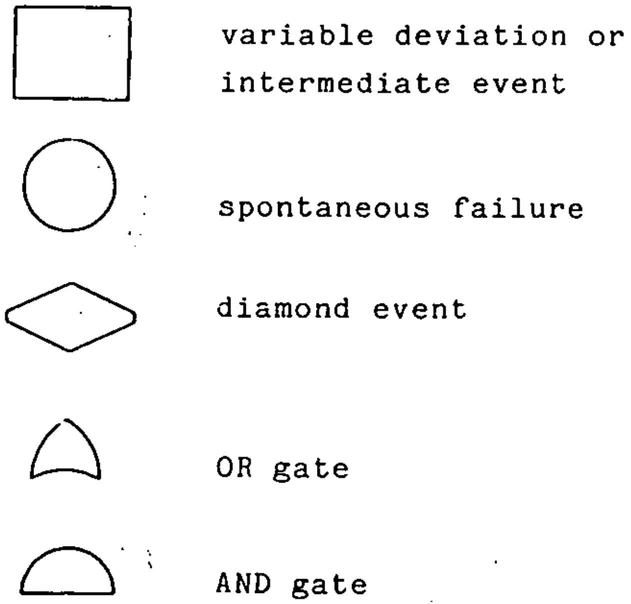


Figure 3.11 - standard fault tree symbols, after Haasl [9]

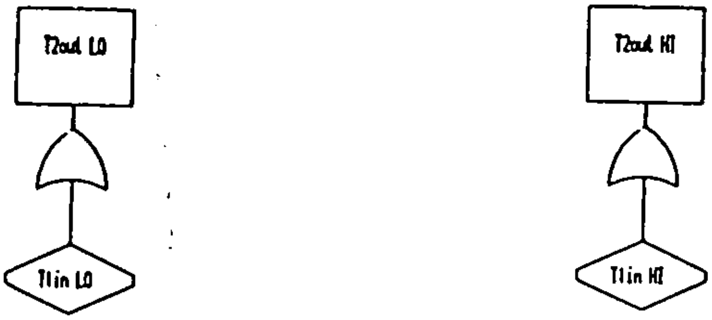


Figure 3.12 - the two minitrees for the deviation of temperature at the outlet of a pipe, derived from the propagation equation for temperature

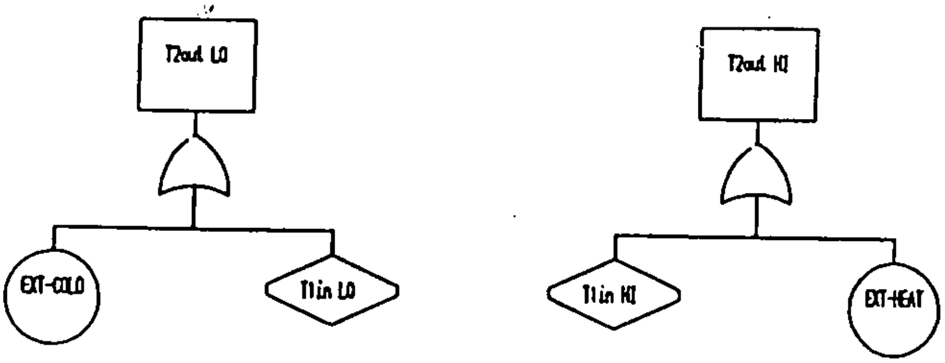


Figure 3.13 - the two minitrees for the deviation of temperature at the outlet of a pipe, derived from a propagation equation and event statements

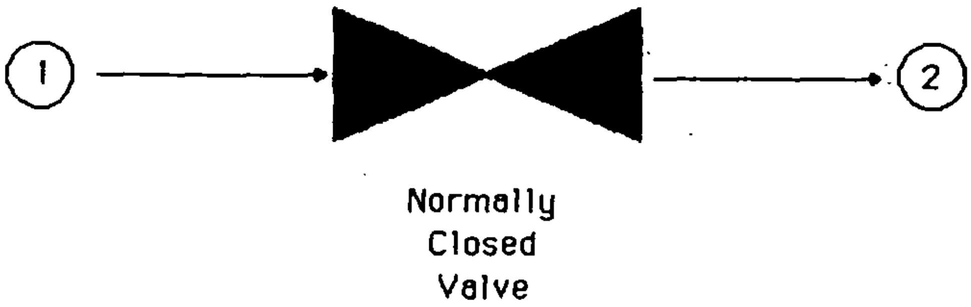


Figure 3.14 - the representation of a closed valve

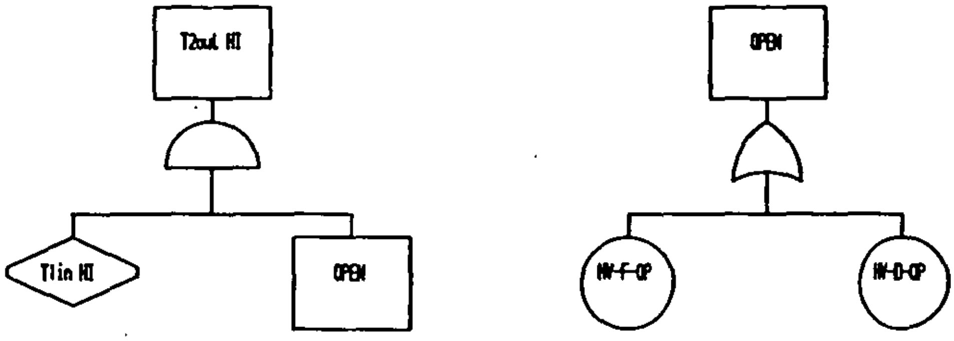


Figure 3.15 - the minitrees for high temperature leaving a closed valve, derived from two event statements

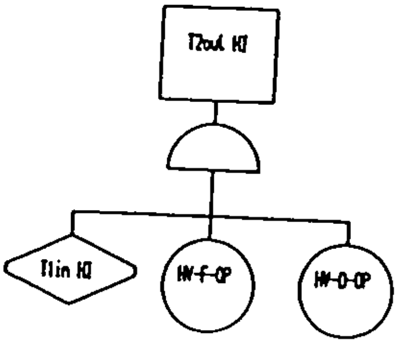


Figure 3.16 - the incorrect minitree for high temperature leaving a closed valve, derived from a single event statement

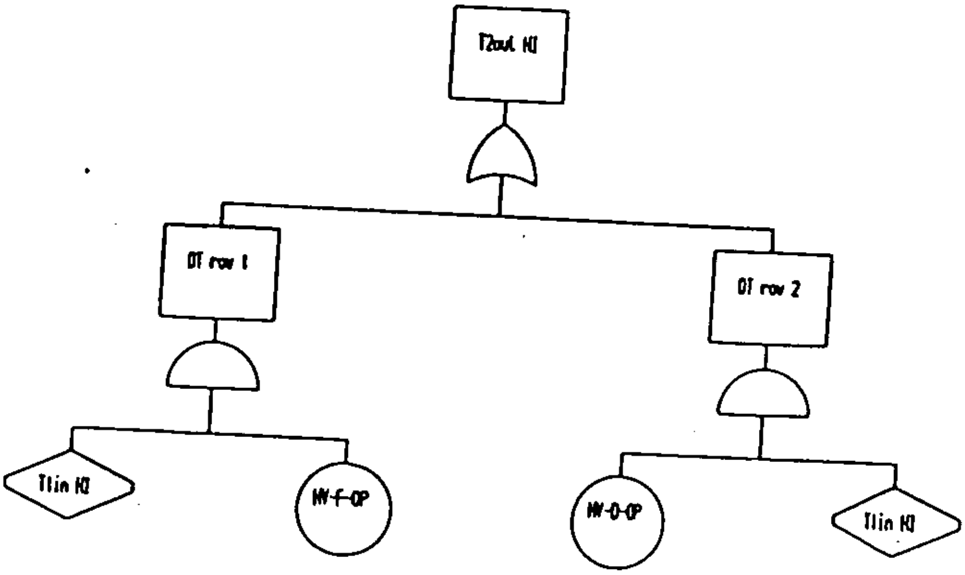


Figure 3.17 - the minitree for high temperature leaving a closed valve, derived from two decision tables

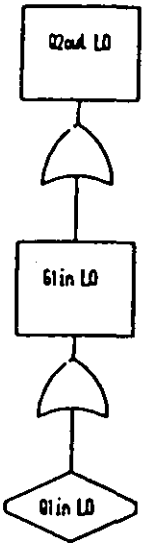


Figure 3.18 - flow propagation from a downstream effect to an upstream cause

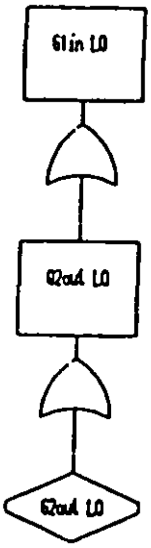


Figure 3.19 - flow propagation from an upstream effect to a downstream cause

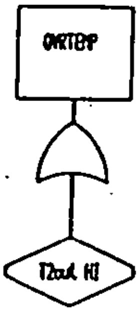


Figure 3.20 - the minitree for the top event OVRTEMP

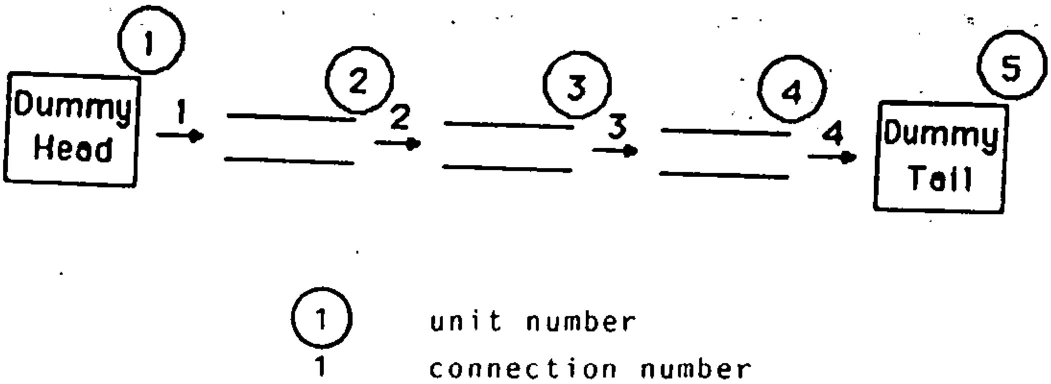


Figure 3.21 - the configuration diagram for a simple pipework system

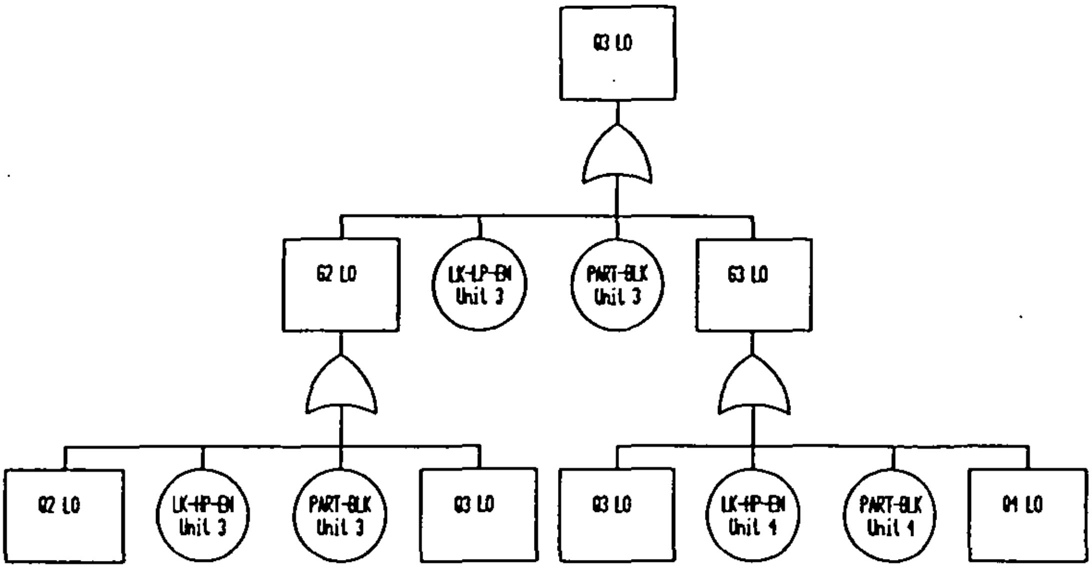


Figure 3.22 - the fault tree for the system shown in Figure 3.21, including inconsistent events

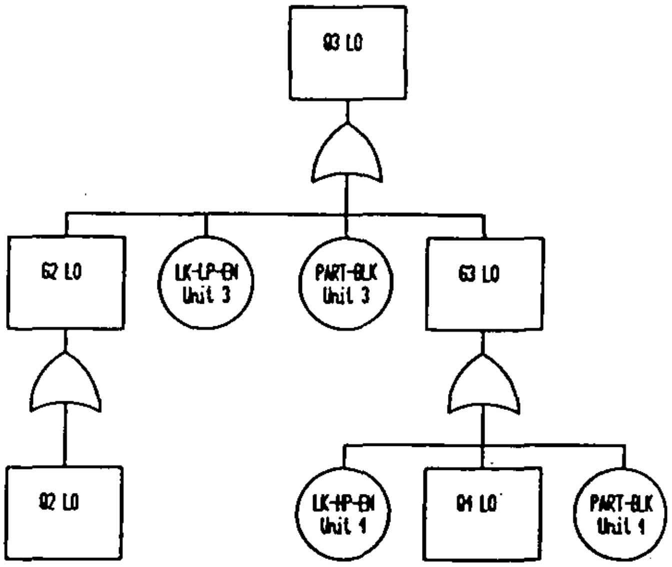


Figure 3.23 - the fault tree for the system shown in Figure 3.21, with inconsistent events removed

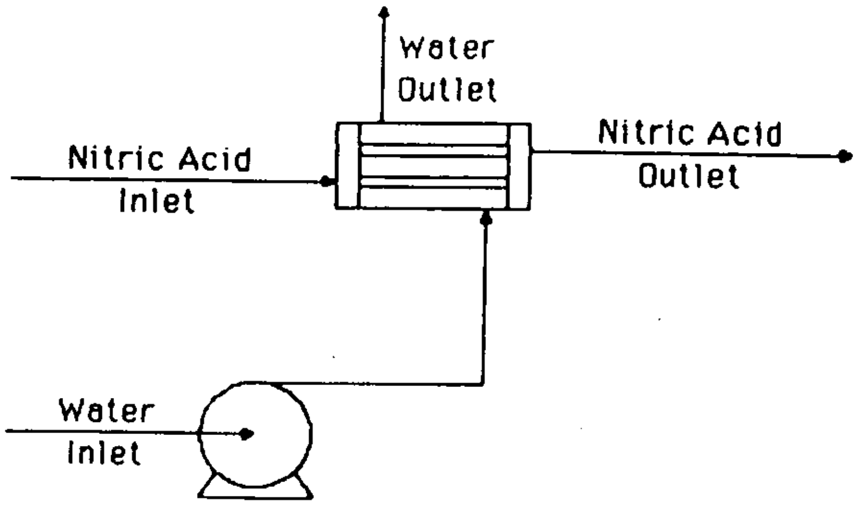
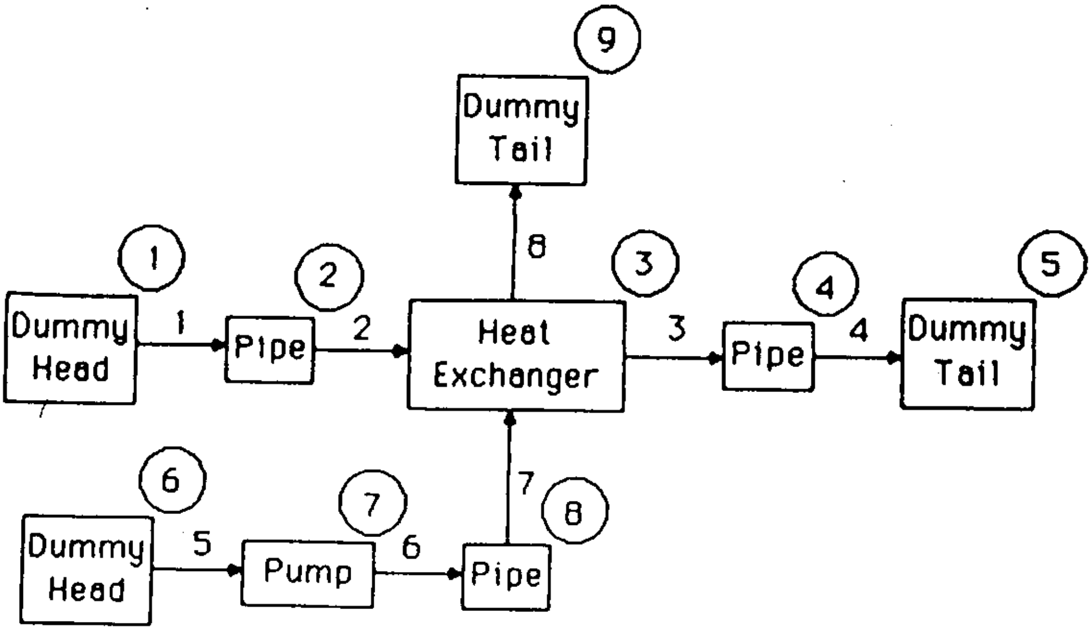


Figure 3.24 - a simple nitric acid cooling system, after Lapp & Powers [23]



① unit number
1 connection number

Figure 3.25 - configuration diagram for the system shown in Figure 3.24

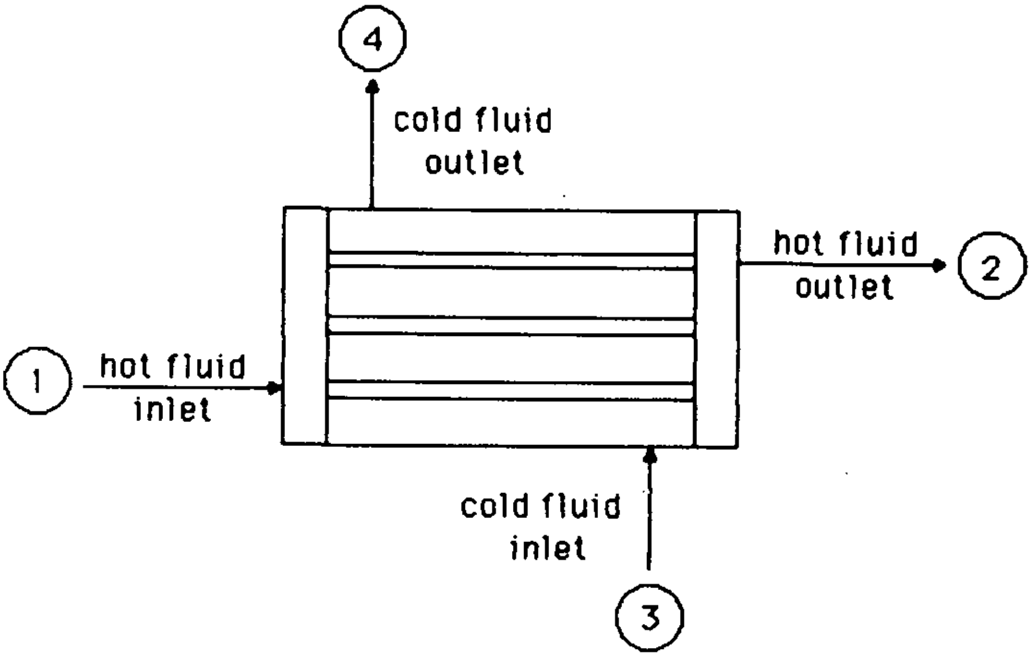


Figure 3.26 - the representation of a heat exchanger

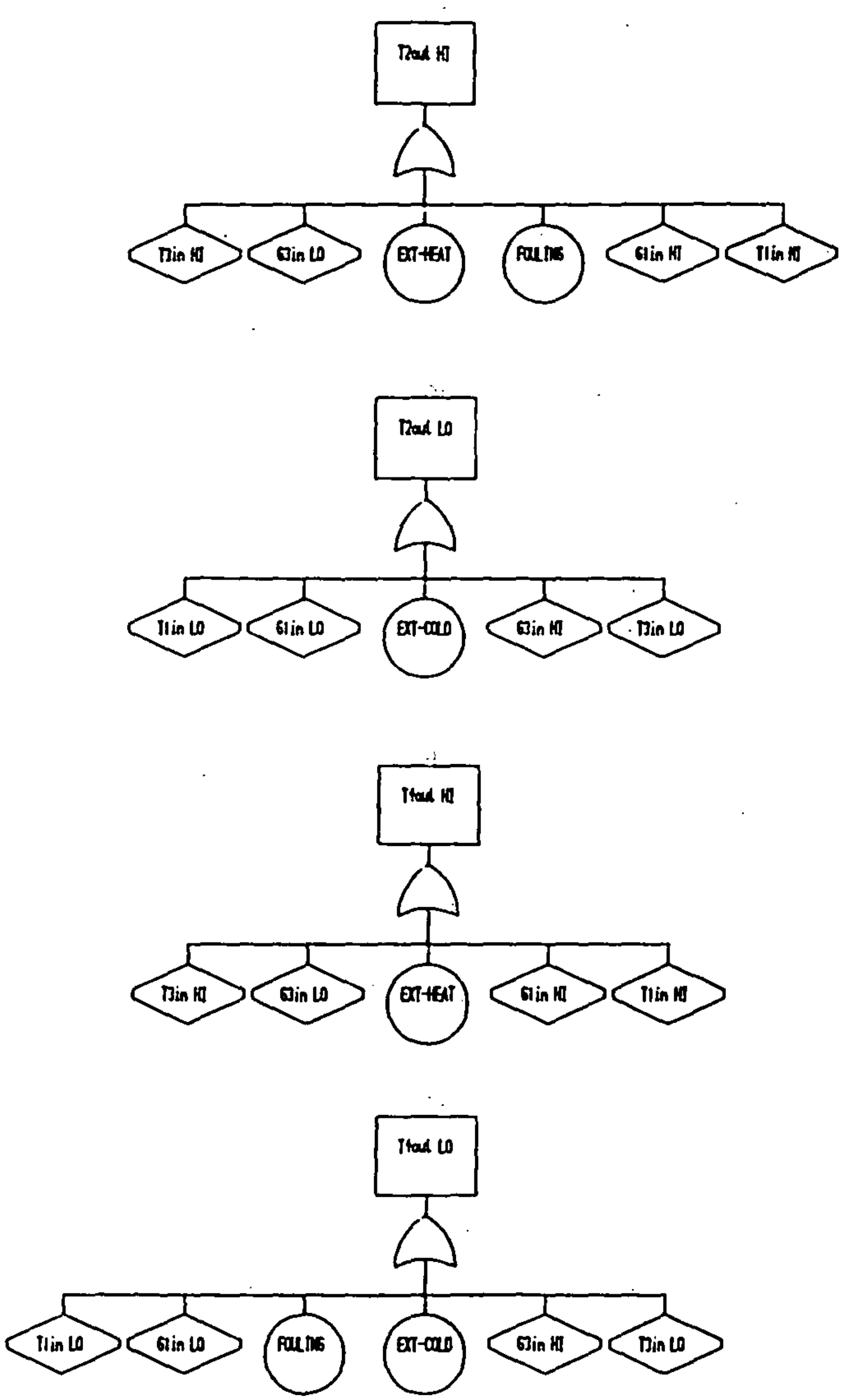


Figure 3.27 - the minitrees for the deviations of temperature in a heat exchanger

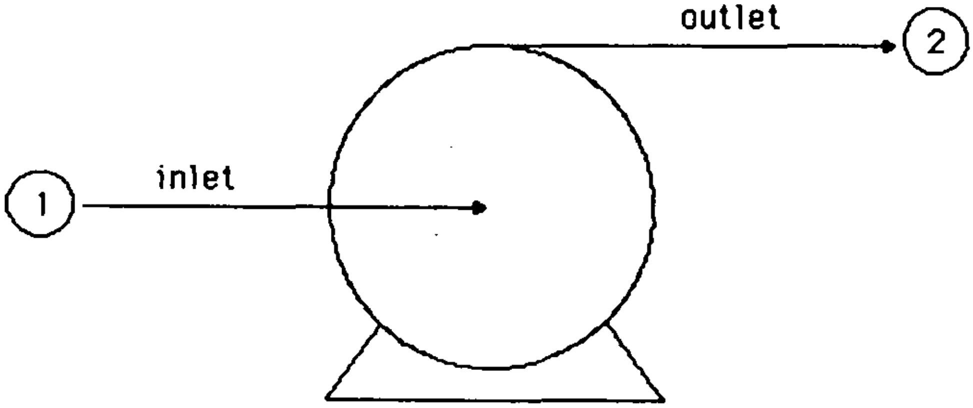


Figure 3.28 - the representation of a centrifugal pump

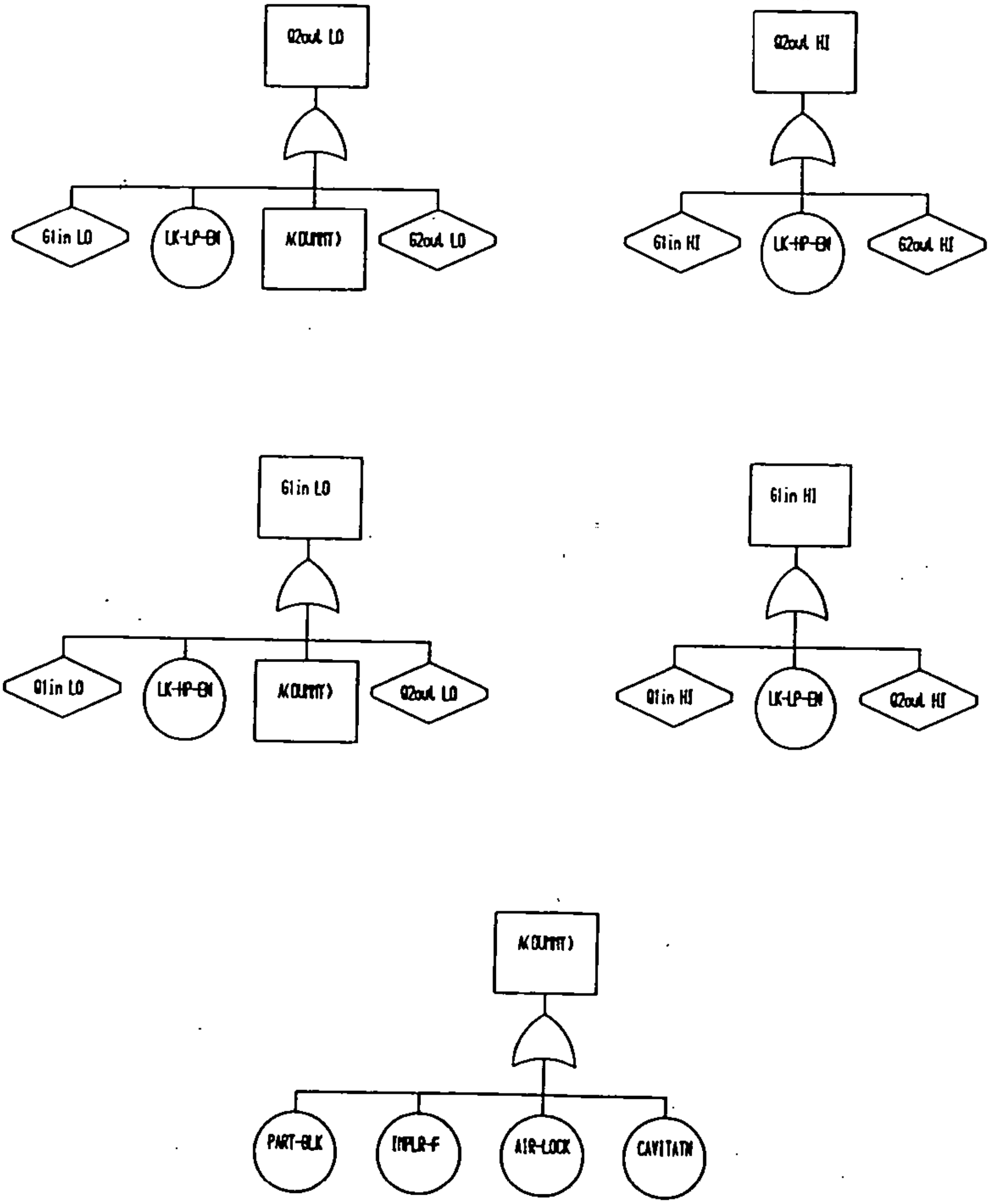


Figure 3.29 - the minitrees for the deviations of flow in a centrifugal pump

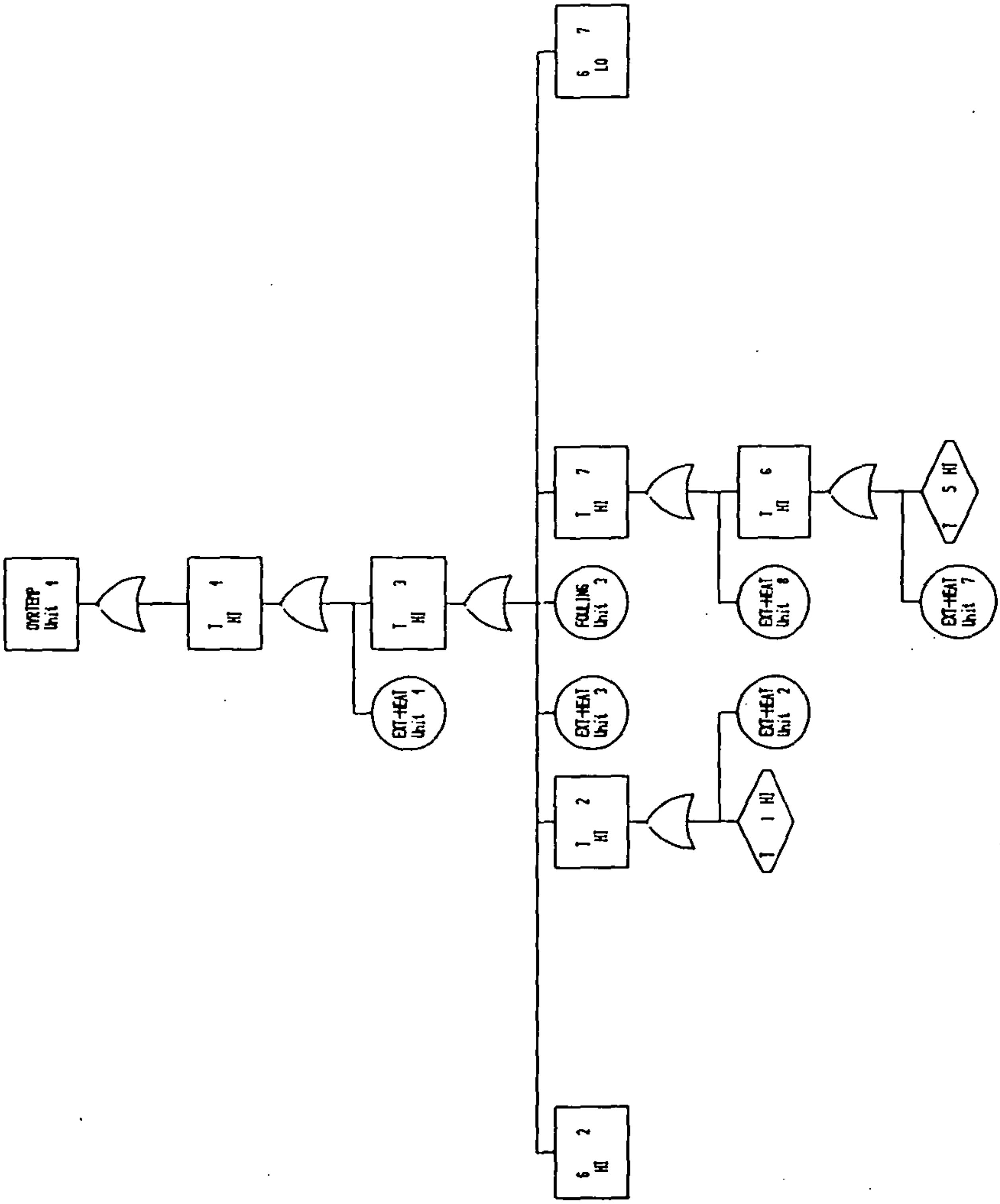


Figure 3.30 - partial fault tree for the system shown in Figure 3.25

Heat Exchanger Example

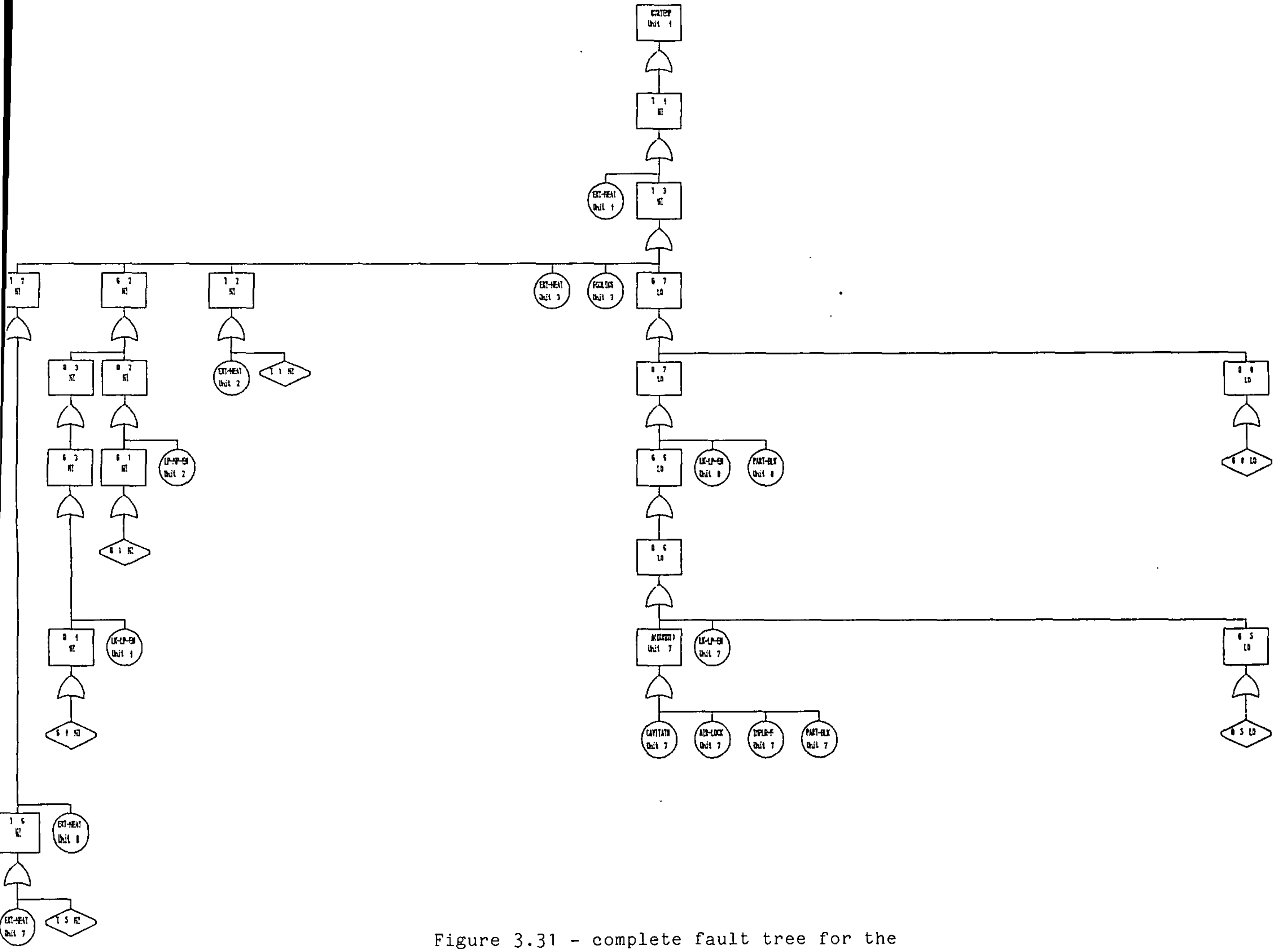


Figure 3.31 - complete fault tree for the system shown in Figure 3.25

4) More Advanced Principles

The previous chapter introduced the basic principles of fault tree synthesis, namely decomposition, modelling, and synthesis itself. The modelling aspect was limited almost entirely to the deviations LO and HI. This chapter examines modelling in more detail, and presents a complete example utilising the more detailed models.

4.1) More Detailed Modelling

The previous chapter considered only simple modelling, namely the high and low deviations of flow, temperature and composition in pipes and similar units. Generally, modelling is required to cover a wider range of situations, such as the causes and effects of reverse flow, and the modelling of absolute pressure. The sub-sections below discuss the modelling of such events in pipe type units. Chapter 5 discusses the modelling of vessel units, such as tanks.

4.1.1) Flow and Pressure Gradient

Flow and pressure gradient variables have already been modelled for the deviations HI and LO (see Section 3.2.4.1). Two propagation equations were written to model the connection between flow and pressure gradient :-

G1in=f(Q1in,Q2out)
Q2out=f(G1in,G2out)

Altogether, there are five flow deviations of interest - high flow, low flow, no flow, some flow and reverse flow. These deviations can be modelled by defining five deviations (HI, LO, NONE, SOME and REV) for both the 'Q' and 'G' variables. The two propagation equations above contain all the information necessary to develop the minitrees for all the deviations of each variable. These give rise to the ten minitrees of Fig 4.1.

Note the exact correspondence between the meaning of the deviations for the flow variable and the pressure gradient variable. For example, Q REV means reverse flow. G REV means that the pressure gradient is reversed (pressure at the outlet is larger than pressure at the inlet). This, of course, results in reverse flow.

The pipe model created using these minitrees is not sufficient to correctly model all possible situations involving flow deviations, as a simple example will illustrate.

Consider the section of pipework shown in Fig 4.2, containing two valves that are normally closed. Clearly, for flow to exist in the pipe between the valves, both valves must be open. A fault tree of the form shown in Fig 4.3 is required. Note that the top event is associated with an AND gate. However, the model for some flow in a pipe involves only OR gates. It therefore appears necessary to modify the pipe model by giving additional information in decision table form :-

V G1in SOME V G2out SOME T Q2out SOME
V Q1in SOME V Q2out SOME T G1in SOME

This certainly provides an AND gate, and is a possible solution. It is, however, very cumbersome, especially when mechanical failures are included in the model. Decision tables would also be necessary to incorporate the effects of a leak to a low pressure environment (LK-LP-EN) and a leak from a high pressure environment (LK-HP-EN). Suitable expressions are

```
F LK-LP-EN V Q1in SOME T G1in SOME
F LK-HP-EN V G2out SOME T Q2out SOME
```

The first statement says that an upstream source (Q1in SOME) AND a leak to low pressure environment in the unit is a cause of some flow into the unit (G1in SOME). For some outlet flow to exist (Q2out SOME), a leak from a high pressure environment and a downstream sink (G2out SOME) is necessary.

There is another solution, involving the use of top event models. The AND gate is supplied by the top event model, and the pipe, and similar models, contain only the propagation equations and suitable event statements. The top event SOMEFLOW is used to represent some flow. The minitree for this event is shown in Fig 4.4. When this is used to synthesise a fault tree for some flow through the pipe, the resulting tree contains an AND gate in the correct position, as shown in Fig 4.5. Note that each branch of the top event is propagated in only one direction; the initial removals of Q4 SOME under G3 SOME of the top event, and of G3 SOME under Q4 SOME of the top event, are made using the internal consistency checks.

Another point to note in the fault tree of Fig 4.5 is that the events Q1 SOME and G6 SOME, the events at the dummy head and the dummy tail do not appear. The reason for this is that it is assumed that some flow from dummy heads and to dummy tails is a normal state, therefore such events should not appear in a FAULT tree. The result of removing these events from the fault tree as normal states is to remove a complete fault tree branch, up to the AND gate above the normal state, since (logically) a normal state ORed with some fault results in a normal state, but a normal state ANDed with a fault results in the fault. In this case, the AND gate appears in the model for the closed valve - for flow to exist through the valve there must be flow into the valve AND the valve must be open. The event removals up to the AND gate are handled by the internal consistency checks.

Reverse flow is more complicated. There are some situations in which an AND gate is required in the final fault tree, and there are some situations in which an OR gate is correct. In the example considered above, an AND gate is clearly required, since faults both upstream and downstream must exist before reverse flow can occur. On the other hand, reverse flow can sometimes occur because of a large decrease in pressure upstream, or a large increase in pressure downstream. The analyst must decide on whether an AND gate or an OR gate is required depending on the situation, and use an appropriate top event model.

In certain situations, this approach should also be applied to the modelling of high flow and low flow. The minitrees for high flow state that a high flow will result if there is an increase in pressure upstream. This is not strictly true. High flow will only result if there is also a route for flow downstream. In virtually all situations however, this proviso can be ignored. The analyst should use the deviation SOME when modelling flow in locations where flow does not normally exist. High and low flow deviations are therefore used only in situations where flow does normally occur. The probability that there will be a route for flow downstream is close to 1, and so, in failure terms, the event can be safely ignored. Section 7 illustrates one circumstance in which this assumption proves invalid.

Some models require the AND gate that is normally provided by the appropriate top event model. This occurs when a new fault tree branch is started. In a heat exchanger model, one cause of high outlet temperature is no flow of coolant, provided that some flow of the hot medium exists. It would not be sufficient to use a decision table like

V G3in NONE V G1in SOME T T2out HI

since there would be no AND gate associated with the causes of G1in SOME. Instead, the AND gate has to be supplied by the decision table, as in

V G3in NONE V G1in SOME V Q2out SOME T T2out HI

The principle that the component models should be kept as simple as possible appears several times during modelling. More complex modelling is sometimes required, but the need is restricted to a few models under the conditions noted above.

4.1.2) Temperature and Composition

This section is restricted to consideration of the 'X' and 'T' variables, that is situations where reverse flow is not involved. Section 4.1.4 describes the modelling of situations where temperature and composition may be transported by reverse flow.

Only the deviations HI and LO are used with the X and T variables. For temperature they have the obvious meanings of high and low temperature, respectively. When used with composition, however, HI may mean either that there is high concentration of the component, or that some of the component is present when none is expected. The event X SOME has not been used to represent the latter of these two to avoid the introduction of an unnecessary event that would add to the model complexity. X LO has the obvious meaning of low (or even no) concentration of the impurity. There is no event X NONE, for reasons similar to why there is no X SOME event. In any given study, only one of the two possible meanings of the HI and LO deviations is sensible. Confusion should not, therefore, occur.

The propagation equations that describe this performance are

T2out=f(T1in)
X2out=f(X1in)

If necessary, multiple components can be modelled by using variable subscripts. XA and XB are recognised as different variables. Only models that involve multiple components need have propagation equations for each of the different components. In units such as pipes where the functional behaviour of each component is identical the standard propagation equation given above is sufficient. An example of a model involving multiple components is the binary mixer (see Section 4.2.2.2).

Situations where the same component exists in both vapour and liquid phases, when the distinction between the phases is important, can be modelled using component subscripts. One subscript can be used for the liquid phase, and a different subscript used for the vapour phase. An alternative solution, applicable only to vessel models, is illustrated in the binary distillation column model (Section 5.2).

4.1.3) Pressure

Deviations of pressure, like deviations of flow, propagate both upstream and downstream of the initiating fault. As explained in the section on flow (Section 3.2.4) this requires that two variables are used, one propagating out of outlet ports, the other propagating out of inlet ports. A suitable variable to associate with pressure is relief. By convention, pressure propagates out of outlet ports, and relief propagates out of inlet ports.

It has not proved possible to write propagation equations linking pressure and relief similar to the flow and pressure gradient equations. The reason for this is that deviations of pressure and of relief are not identical in the same way that flow deviations are synonomous with deviations of pressure gradient (see Section 4.1.1). A different approach to obtain two way propagation of pressure and relief faults has therefore been adopted.

The pressure variable, 'P', since it is defined at outlets, can trace faults to causes in units upstream of the deviation (like the T and X variables do). The relief variable, 'R', on the other hand, traces faults to causes downstream of the deviation. It is therefore possible to combine deviations of the two variables to build up a two way propagation structure for pressure and relief faults. Top event models are used to create complete pressure and relief events by combining variable deviations of the 'P' and 'R' variables. For example, the top event OVRPRES, representing high pressure in a pipe involves the following elements

a) events upstream - connection to a high pressure source, represented by P2out HI

b) events downstream - there is insufficient relief, no relief at all, or a back pressure from downstream (R1in LO or R1in NONE, or R1in REV)

The minitree for this top event is given in Fig 4.6.

Table 4.1 contains a list of the deviations of the P and R variables and their meaning. Note the two special deviations NOR (no relief), which is a deviation of the P variable, although it is actually a relief state, and NOP (no pressure), which is a deviation of R, although it represents a pressure state. The remainder of the deviations are more logical, if it is noted that P REV (some relief) can be regarded as reverse pressure, and that R REV (some pressure) can be considered as reverse relief.

Similar minitrees exist for the other top events that represent pressure and relief faults in pipe type units. These minitrees need only appear in top events, and in models where pressure appears at the start of a new fault tree branch, as described earlier (Section 4.1.1). For other pipe type models the standard propagation equations are

R1in=f(R2out)
P2out=f(P1in)

4.1.4) Temperature and Composition in Reverse Flow

Section 4.1.2 has given an outline of the use of the 'T' and 'X' variables for the modelling of temperature and composition. The limitation with these variables is that they can only trace faults to causes that are upstream. Obviously it is possible for reverse flow to carry temperature and composition faults from downstream.

To model this, it is necessary to introduce two new variables which can look for temperature and composition variations downstream. 'U' has been used for temperature, and 'Y' for composition. The standard propagation equations for these variables are

U1in=f(U2out)
Y1in=f(Y2out)

Note that these variables propagate out of models at inlet ports. This ensures that the variables can trace faults to causes downstream. In this sense, they are similar to the relief variable 'R'.

In virtually all circumstances, deviations of these variables must be ANDed with reverse flow to cause a particular variable deviation at a specified point. However, the AND gate need only appear in the top event, and in situations where these events are at the start of a new fault tree branch, as described in Section 4.1.1.

If the model is a multiple component model, then variable subscripts must be used with the 'Y' variable in the same way that they are used with the 'X' variable (Section 4.1.2).

To model completely temperature and composition deviations, it is necessary to combine these reverse flow variables with the normal flow variables, T and X. The top event model for OVRTEMP (over temperature, or too high a temperature at a given point), is defined using two decision tables

V T2out HI V G1in SOME V Q2out SOME T OVRTEMP
V U1in HI V Q2out REV T OVRTEMP

The minitrees for this model are shown in Fig 4.7. Similar top event models for low temperature and composition deviations exist. Note that the T variable is ANDed with some flow through the unit. This expression involves two deviations. This is necessary to start the synthesis of the some flow branch on the correct lines, as described in Section 4.1.1.

4.1.5) Flow Ratio

The modelling of systems which involve flow ratio presents some problems to a modelling approach based on component models. Consider the mixer shown in Fig 4.8. Stream 1 is mixed with stream 3 to produce stream 2. The composition of stream 2 depends on the flow ratio between the two inlet streams. A suitable propagation equation for composition in the mixer appears to be

$$X2out=f(G1in,-G3in)$$

However, as the fault tree shown in Fig 4.9 shows, this propagation equation results in a fault tree in which flow faults of both deviations downstream of the mixer result in a composition deviation. This is clearly incorrect. The problem arises because the propagation package is simple minded in its search for possible causes of an event. Low flow downstream will cause low flow of one inlet stream, and so is included in the fault tree. High flow downstream is similarly identified as a potential cause of high inlet flow of the other stream. The modelling package does not normally take into account the possible ameliorating effects that these faults may induce. In this example, such effects exist. For example, a pressure drop downstream will increase the flows of both inlet streams, and the composition will be unchanged.

Specifying that ameliorating effects should be taken into account when studying the causes of a particular event involves modelling the mixer slightly differently. The information that the causes of one event may be compensated for by other effects of that cause is included by using a special term in the propagation equation for the mixer. A suitable propagation equation for composition in the mixer is

X2out=f(G1in/G3in)

The '/' denotes that the causes of the two events on the right hand side of the propagation equation may have effects that are contradictory, and that any such contradicatory events should be removed from the fault tree. This operator is known as the flow ratio operator, because it is most commonly used in situations involving flow ratio.

This propagation equation is resolved into two causes, which appear identical to the causes of the earlier mixer, namely low flow of one inlet stream and high flow of the other inlet stream. However, the flow ratio operator alerts the synthesis package to take special note of the causes of the flow events, and remove any causes that are contradictory.

Flow ratio expressions can appear in event statements, in which case they take the form

V G1in/G3in LO:effects

The complex flow ratio variable is entered as if it were a simple variable.

There are a number of restrictions that should be adhered to when using modelling flow ratio. Only variable deviations can appear in a flow ratio expression, and only two entries are permitted in the ratio term. AND gates and r/n gates are not allowed in the same event statement, nor can flow ratios appear in decision tables, to avoid confusion over the logic of the resulting minitree. However, there is no restriction on including a simple (i.e. non flow ratio) event that has flow ratio causes appearing in an event statement with an AND gate, or in a decision table.

Furthermore, it is recommended that flow ratio causes are specified as the sole cause of a particular event. The reason for these restrictions is to prevent other causes in the fault tree being subjected to the same treatment as the flow ratio causes.

Dummy variables can be used to conform to these restrictions. For example, if components A, B and C are being used in a particular study, then component D can be used as a dummy to represent the flow ratio, viz

XD2out=f(G1in/G3in)

XD2out can be used in place of the flow ratio term in decision tables, or in event statements involving AND or r/n gates. Alternatively, intermediate events can be used instead of a dummy variable.

The fault tree for the mixing system with the special flow ratio treatment is given in Fig 4.10. Note that no deviation of flow downstream of the mixer appears in the fault tree.

The detailed example at the end of this chapter (Section 4.2) examines a mixing system in more detail.

The use of the flow ratio operator is not restricted to modelling composition. It can equally be used to model the deviations of other variables, should this prove necessary.

4.1.6) Total Component Flow

Total component flow is almost identical to flow ratio. The only difference in modelling terms is how propagation equations are resolved into causes.

The reason why it is necessary to have a special treatment for total component flow is analogous to the reason why flow ratio requires a special treatment. Consider the reactor shown in Fig 4.11. Stream 1 carries a dilute mixture of oxygen into the reactor. If there is too much oxygen in the reactor, then reaction runaway may occur. Too much oxygen may be caused by too high an oxygen composition at the inlet, or by the feed rate of the oxygen stream being too high. But the causes of these may well be contradictory. Consider the diagram of Fig 4.12. The oxygen is diluted by mixing with an inert gas. Too high a composition of oxygen will be caused by too low a flow of the dilution stream. On the other hand, too high a flow of the dilution stream will result in a high flow of feed to the reactor. However, neither of these causes will actually result in too much oxygen actually in the reactor.

Total component flow is designed to overcome this problem. As with flow ratio, the causes of total component flow events must not be contradictory. Modelling total component flow takes the form

T3ves=f(G1in*X1in)

The '*' is used as the total component flow operator.

The propagation equation is resolved into minitrees where G1in and X1in have the same deviations. It is permissible to have minus signs in front of the cause. These are the only modelling differences between flow ratio and total component flow (flow ratio was resolved into causes with opposite deviations). As with flow ratio, it is recommended that a total component flow cause should be the only cause of an event. Dummy variables or intermediate events can be used to achieve this.

4.2) A Synthesis Example

This section illustrates the use of some of the more advanced modelling concepts introduced in this chapter. The system considered is a mixing system. This system, illustrated in Fig 4.13, is based on an example presented by Taylor [43].

The plant is designed to mix two components, A and B, to produce an outlet stream of particular composition.

4.2.1) Decomposition

Decomposition in the example is straightforward. The configuration diagram for this system is shown in Fig 4.14. Note that connections 1 and 2 carry component A, and connections 3 and 4 carry component B. Connections 5 and 6 are the mixture.

4.2.2) Unit Modelling

This section details the models required for this example.

4.2.2.1) Pipe

The pipe model used in this example is a more detailed model than used in Section 3.4. The propagation equations for this model have been derived already (see Section 4.1.1 to 4.1.4). Event statements have been added to the model below. The only new event is COMP-BLK, used to represent a complete blockage of the pipe.

```
G1in=f(Q1in,Q2out)
R1in=f(R2out)
U1in=f(U2out)
Y1in=f(Y2out)
Q2out=f(G1in,G2in)
T2out=f(T1in)
X2out=f(X1in)
P2out=f(P1in)

F PART-BLK:G1in LO,Q2out LO,R1in LO,P2out LO
F COMP-BLK:G1in NONE,Q2out NONE,R1in NONE,
                    R1in NOP, P2out NONE, P2out NOR

F LK-LP-EN:G1in HI,G1in SOME,R1in HI,R1in SOME
F LK-LP-EN:Q2out LO,Q2out NONE,Q2out REV
F LK-LP-EN:P2out LO,P2out NONE,P2out REV

F LK-HP-EN:G1in LO,G1in NONE,G1in REV
F LK-HP-EN:R1in LO,R1in NONE,R1in REV
F LK-HP-EN:Q2out HI,Q2out SOME,P2out HI,P2out SOME
```

4.2.2.2) Binary Mixer

This unit is designed to mix component A and component B to produce a single stream. Port 1 carries component A into the model and port 3 carries component B into the model. Port 2 is the outlet port.

The propagation equations for flow in this unit are

G1in=f(Q1in,Q2out,-G3in)
Q2out=f(G1in,G3in,G2out)
G3in=f(Q3in,Q2out,-G1in)

The equation for the flow variable, Q, is similar to the corresponding equation in the pipe model. However, whereas in the pipe model, there was only one upstream port, in this model there are two upstream ports, which must both appear in the propagation equation. The equations for G are also based on the pipe model. Both equations involve a term upstream (Q1in in the first equation, and Q3in in the second), and a term downstream (Q2out). However, the pressure in one inlet affects the flow in the other inlet, and the third term in each equation models this. For example, as the pressure, and so the pressure gradient, at port 1 increases, so the flow at port 3 will decrease.

These propagation equations do not contain all the necessary information about how the flow variables interact. Supplementary information in both event statement and decision table forms is required.

```
V G1in NONE:G3in HI,C(DUMMY)
V G3in NONE:G1in HI,C(DUMMY)
V G1in SOME:G3in REV
V G3in SOME:G1in REV
V G1in REV:G3in SOME,D(DUMMY),Q2out NONE,G3in HI
V G3in REV:G1in SOME,D(DUMMY),Q2out NONE,G1in HI
I C(DUMMY):Q2out LO
I D(DUMMY):Q2out LO

V G1in NONE V G3in NONE T Q2out NONE
```

This information is more easily understood if the minitrees for the flow variables are considered. Fig 4.15 gives the minitrees for the deviations of G1in, and Fig 4.16 the minitree for the deviations of Q2out. The minitrees for G3in are similar to those of G1in. One point of interest is the interaction of the deviations SOME and REV. If reverse flow occurs through one inlet port, then there is a sink for some flow through the other inlet port. Similarly, if some flow occurs through an inlet port, then there will be source for reverse flow to occur through the other inlet port. Therefore, some and reverse flows at the two inlet ports are closely related.

Pressure and relief deviations will be ignored in this model, since they are not required in the current study.

In the absence of any information about the normal temperatures of the two component streams, it will be assumed that the temperatures are similar. The outlet temperature is therefore more dependent on the inlet temperatures rather than the ratio of the inlet flows. The following equation models this

T2out=f(T1in,T3in)

The reverse temperature variable, U, is also considered. The equations for this are

U1in=f(U2out,T3in)
U3in=f(U2out,T1in)

These equations are similar to the pipe equations. However, if reverse flow through either inlet port exists, then there are effectively two temperature sources, one from the normal outlet port, and the other inlet port. Therefore, the inlet temperature at port 1 affects the reverse inlet temperature at port 3, and vice versa.

Composition in this model should be modelled using multiple component subscripts, since the functional behaviour of composition depends on which component is being modelled. The outlet compositions depend not only on the inlet compositions, but also on the ratio of the inlet flows. The propagation equation for component A is

XA2out=f(XA1in,XA3in,G1in/G3in)

The G1in/G3in expression is the flow ratio of port 1 to port 3. Flow ratio is described in Section 4.1.5. This equation is not ideal, nor does it completely model the behaviour of composition. One problem with this equation is that the flow ratio is not the sole cause of an event. This can, however, be achieved by introducing two intermediate events. A(DUMMY) is used to represent a low flow ratio, and B(DUMMY) a high flow ratio, as follows

```
V G1in/G3in LO:A(DUMMY)
V G1in/G3in HI:B(DUMMY)
```

These intermediate events can be specified as causes of the appropriate composition deviations.

The other problem with the propagation equation for composition is that it does not include the effects of complete loss of flow in the inlet streams. Complete loss of flow at port 1 will result in a high outlet composition of B, but only if some flow through port 3 remains. Similarly for no flow through port 3.

The complete model for composition now involves the following statements

```
XA2out=f(XA1in,XA3in)
XB2out=f(XB1in,XB3in)

V G1in/G3in LO:A(DUMMY)
V G1in/G3in HI:B(DUMMY)
I A(DUMMY):XA2out LO,XB2out HI
I B(DUMMY):XA2out HI,XB2out LO

V G1in NONE V G3in SOME V Q2out SOME T XA2out LO,
                                       XB2out HI
V G3in NONE V G1in SOME V Q2out SOME T XA2out HI,
                                       XB2out LO
```

The decision tables involve two variables that represent some flow. This is done to ensure that the AND gate implicit in some flow deviations appears in the fault tree, as described in Section 4.1.1. The minitrees for the deviation of the variable XB2out are given in Fig 4.17. The minitrees for deviations of XA2out are similar. The decision tables have been named

DTrow 2 and DTrow 3, since DTrow1 has been used to model the causes of Q2out NONE.

The effects of reverse flow on the outlet composition have been ignored.

The reverse flow composition variables are similar to the reverse flow temperature variables. The only point to note is that, since the model is a multiple component model, the reverse flow composition variables must also have multiple component subscripts.

YA1in=f(YA2out,XA3in)
YB1in=f(YB2out,XB3in)
YA3in=f(YA2out,XA1in)
YB3in=f(YB2out,XB1in)

4.2.3) Top Event Modelling

The top event for which a fault tree will be synthesised is IMP B HI Unit 6. IMP B HI represents high composition of component B. The location of the top event is the outlet of the system. The model for the top event, in decision table form is

V XB2out HI V G1in SOME V Q2out SOME T IMP B HI
V YB1in HI V Q2out REV T IMP B HI

Deviation of the composition at the outlet of the system can therefore occur either as a result of a compostion deviation upstream, carried by flow in the normal direction, or by a composition deviation downstream, carried by reverse flow. Note that the expression for some flow involves two deviations ANDed together. This is to start the synthesis of the event some flow correctly, as described in Section 4.1.1.

4.2.4) Synthesis

Much of the fault tree synthesis in this example is straightforward, and bears a close resemblance to the synthesis described in Section 3.4.4. This section will concentrate on the synthesis aspects that differ from Section 3.4.4.

The top event is IMP B HI in Unit 6, which is a pipe model. The two decision table causes of this relate to composition deviations upstream, accompanied by flow in the normal direction, and composition deviations downstream, accompanied by reverse flow. Synthesising the causes of the decision table representing downstream effects is straightforward, and requires no description. Synthesising the causes of the decision table representing upstream effects is straightforward, until the causes of composition deviations at the mixer are considered. The fault tree, with all the other events completely synthesised is shown in Fig 4.18. One point to note in this tree is that it is assumed that SOME flow deviations at dummy units are certain to occur, and therefore will not appear in the fault tree, as discussed in Section 4.1.1. However, in Fig 4.18 they are included for clarity.

The event that remains to be studied is XB5 HI, or high composition of component B at the outlet of the mixer. As the model for the mixer indicates, there are several causes of this event. Two relate to high composition of B in each of the inlets to the mixer, and are XB2 HI and XB4 HI. A third cause is A(DUMMY) in the mixer, the intermediate event used to represent a low flow ratio of flow through inlet port 1 to inlet port 3. The fourth cause is the second decision table, which is no flow through port 1, but some flow through port 3. The minitree for XB2out HI of the mixer, involving these causes is detailed in Section 4.2.2.2, and illustrated in Fig 4.17.

The two composition deviations, XB2 HI and XB4 HI can be traced to composition deviations in the supplies. These are repesented by the diamond events XB1 HI and XB3 HI.

The causes of A(DUMMY) in the mixer model are G1in LO (G2 LO) or G3in HI (G4 HI), with the causes being flow ratios, rather than normal flow deviations. There are potential causes of each flow event upstream, downstream and in the other flow leg.

Consider the event G1in LO, or G2 LO. The causes of this are Q1in LO (Q2 LO), Q2out LO (Q5 LO) or G3in HI (G4 HI). Ignoring Q2 LO for the moment, the causes of Q5 LO (Q2out LO) are G2 LO (G1in LO), G4 LO (G3in LO) or G5 LO (G2out LO). The causes of G4 HI (G3in HI) are Q4 HI (Q1in HI), Q5 HI (Q2out HI), G2 LO (G1in LO), G2 NONE or G2 REV. These causes are derived from the flow minitrees of the mixer model, given in Figs 4.15 and 4.16. The fault tree for G2 LO displaying these causes is given in Fig 4.19. There are several inconsistencies in this tree.

One set of inconsistencies in the tree of Fig 4.19 relate to deviations of G2, other than the top event. These are the events G2 LO, which is a cause of both Q5 LO and G4 HI, and G2 NONE and G2 REV, which are causes of G4 HI. These events should be removed from the fault tree since they are either duplicates of the top event, or inconsistent with the top event.

The second set of inconsistencies are the events G4 LO, a cause of Q5 LO, and Q5 HI, a cause of G4 HI. Neither of these events is a possible cause of the top event, G2 LO. The events will tend to cause an increase in the flow through port 1, not a decrease.

The final fault tree for G2 LO, with these two sets of inconsistencies removed, is displayed in Fig 4.20. A similar tree exists for G4 HI, and is given in Fig 4.21. However, the fact that the events G2 LO and G4 HI are a flow ratio affects the causes that should appear in the fault tree for the top event IMP B HI Unit 6. As was noted when flow ratios were introduced (see Section 4.1.5), the causes of the events in a flow ratio cannot involve faults that are contradictory. In this example, this has the effect of removing the potential causes of each of these events that are downstream and in the other inlet leg. The two sets of causes downstream contradict each other. One set states that high flow downstream will cause the required composition deviation; the other set states at low flow downstream will cause the same deviation. Both sets of causes are therefore removed from the fault tree.

There is yet another set of inconsistent events in the fault tree that should be removed before the fault tree is complete. These are the duplicate and inconsistent events of one of the flow ratio events in the causes of the other flow ratio event, namely G4 HI as a cause of G2 LO (see Fig 4.20), and G2 LO, G2 NONE and G2 REV, all causes of G4 HI (see Fig 4.21)

The net effect of the removal of these inconsistencies is to limit the causes of the two flow events to causes that are upstream. The fault tree for the top event IMP B HI Unit 6 which exists at this point, and is given in Fig 4.22, illustrates this.

There is only one event remaining to be considered. This is the decision table, representing no flow of component A and some flow of component B. The causes of this decision table, linked by an AND gate, are G2 NONE, G4 SOME and Q5 SOME. The causes of these causes involve some removals. First, consider the causes of G4 SOME. These are Q4 SOME, Q5 SOME or G2 REV. Only the first of these causes is a valid cause. Q5 SOME is removed since the event has already appeared in this fault tree branch. G2 REV is removed because it is inconsistent with an event in another branch of the AND gate, namely the event G2 NONE. For the AND gate to be valid, G2 NONE must exist. G2 REV violates this condition, and so is removed. Note that it is possible for the AND gate to occur even if G2 REV does not occur. There is therefore no reason to remove the event G2 NONE.

Similar inconsistencies occur when considering the causes of Q5 SOME and G2 NONE. Fig 4.23 shows all the potential causes of the events under the AND gate, and identifies those that are inconsistent with the fault tree.

The remaining events in the fault tree can be synthesised with no further problems.

The complete fault tree for this system, with all the inconsistencies removed, is shown in Fig 4.24.

Variable Deviation	Description
P HI	high pressure upstream
P LO	low pressure upstream
P NONE	no pressure source upstream
P SOME	pressure source upstream exists
P REV	a sink for pressure relief upstream exists
P NOR	there is no sink for pressure relief upstream
R HI	there is a larger-than-normal sink of pressure relief downstream
R LO	there is a smaller-than-normal pressure relief sink downstream
R NONE	there is no pressure relief sink downstream
R SOME	there is a sink for pressure relief downstream
R REV	there is a pressure source downstream
R NOP	there is no pressure source downstream

Table 4.1 - the deviations for the P (pressure) and R (relief) variables

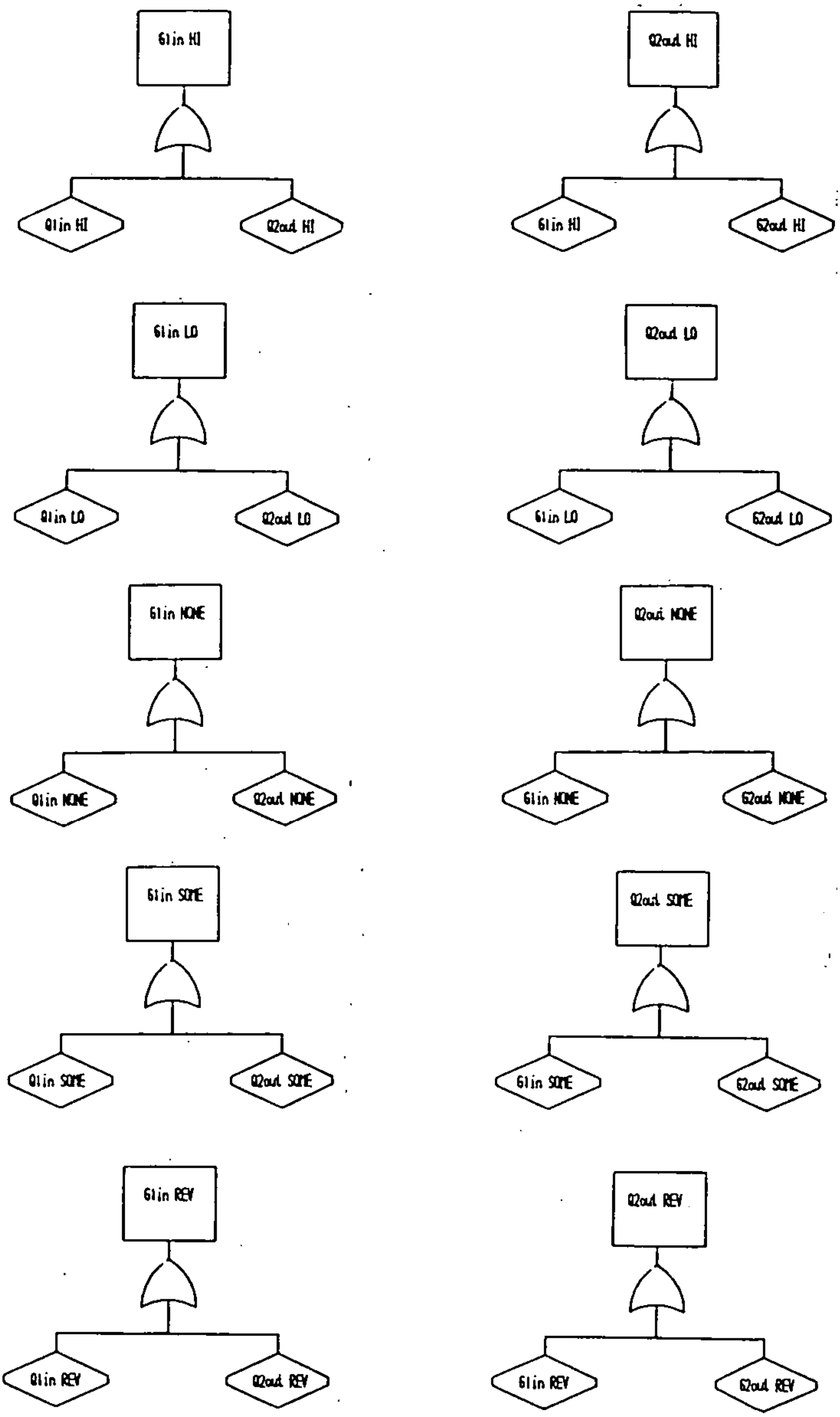


Figure 4.1 - minitrees for Q2out and G1in for a pipe

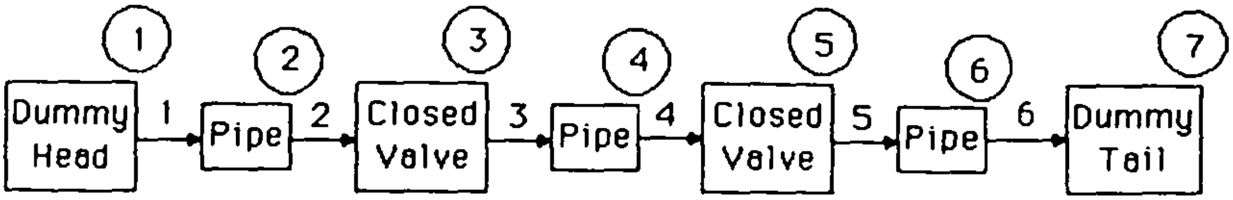


Figure 4.2 - configuration diagram for a pipework system incorporating two valves that are normally closed

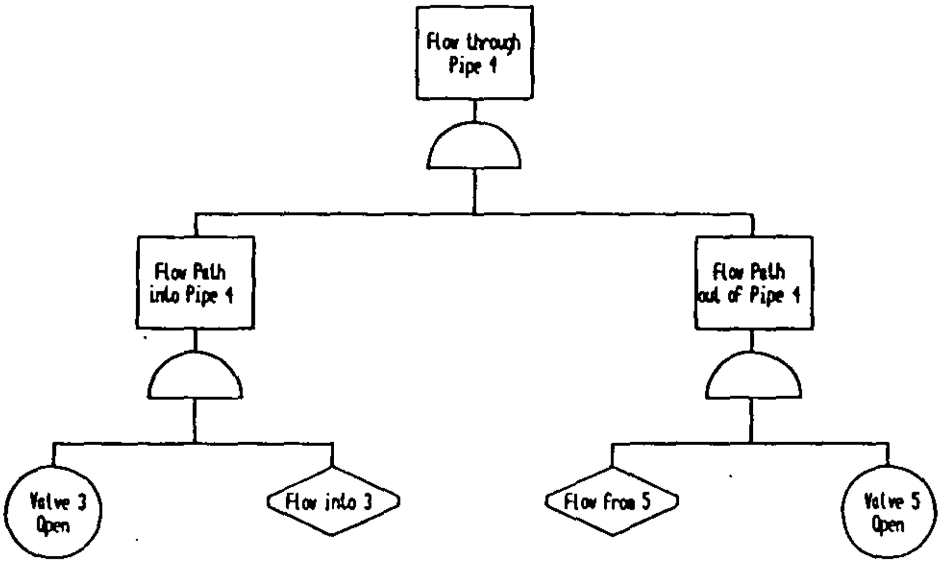


Figure 4.3 - the fault tree required for the system shown in Figure 4.2

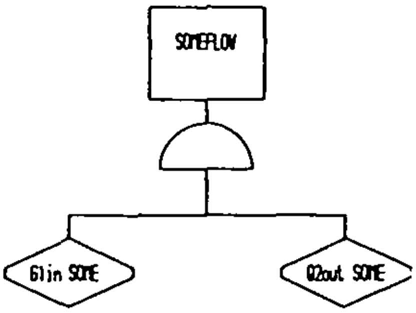


Figure 4.4 - the top event model
for SOMEFLOW

Closed Valve System

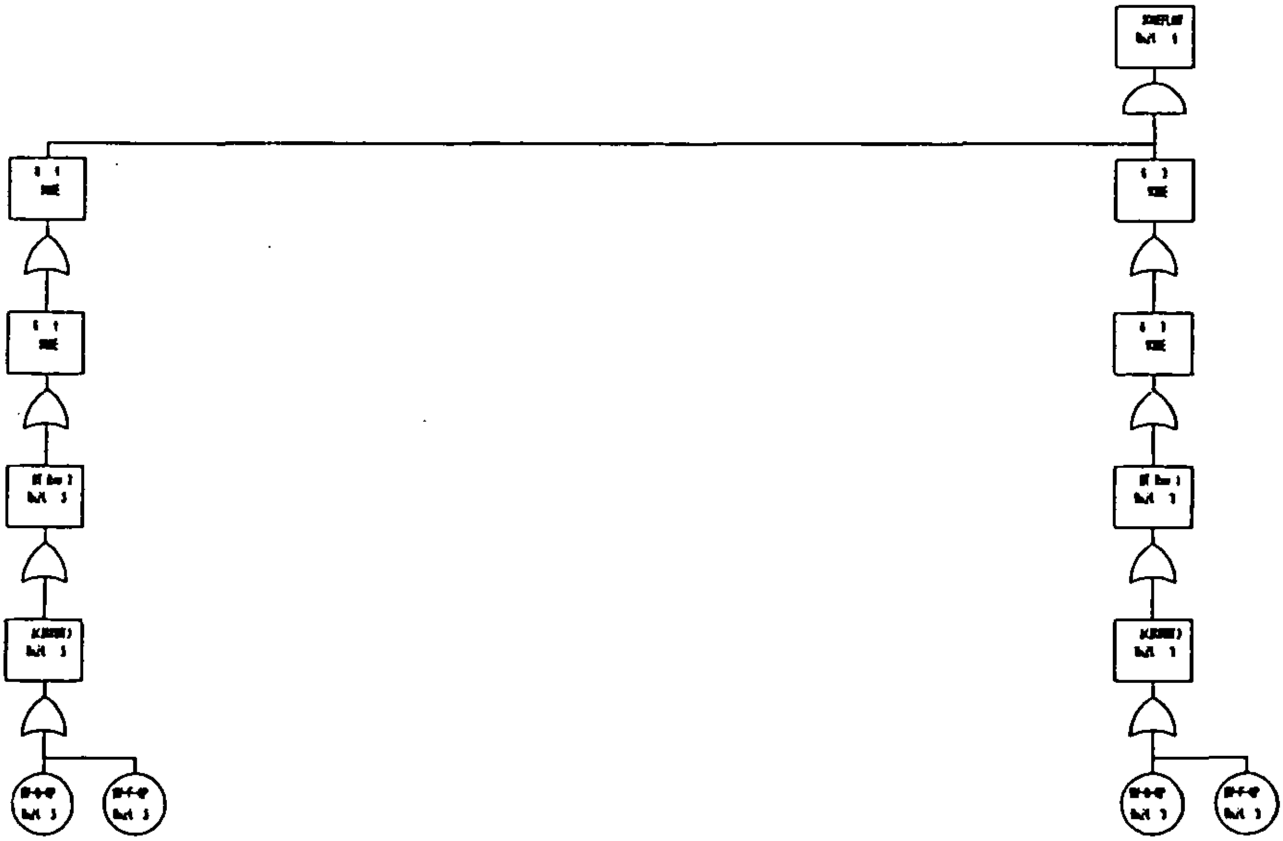


Figure 4.5 - fault tree for the system shown in Figure 4.2, using the top event model of Figure 4.4

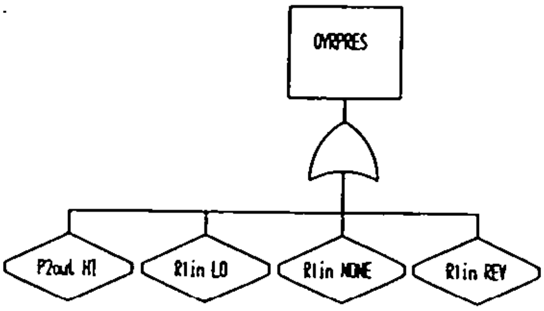


Figure 4.6 - the top event model for OVRPRES (high pressure)

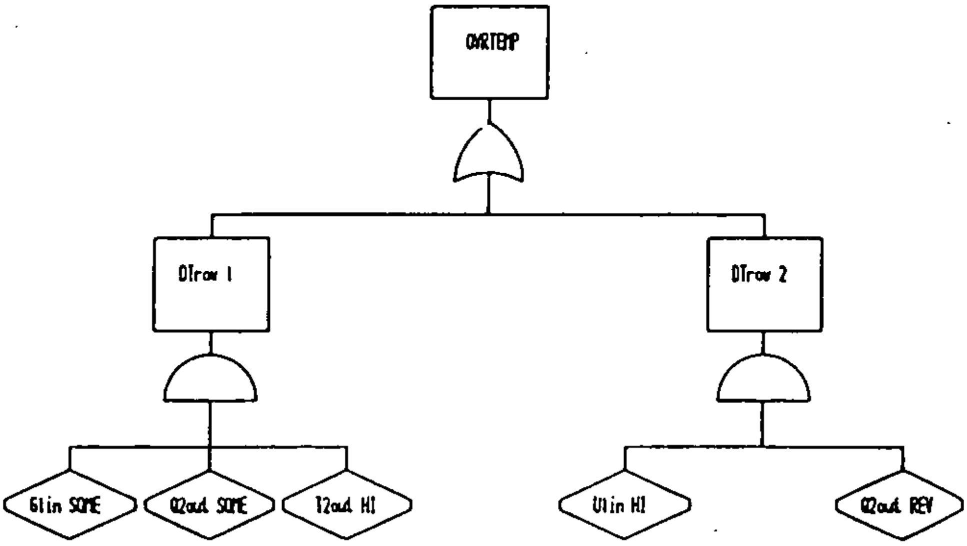


Figure 4.7 - the top event model for OVRTEMP (high temperature)

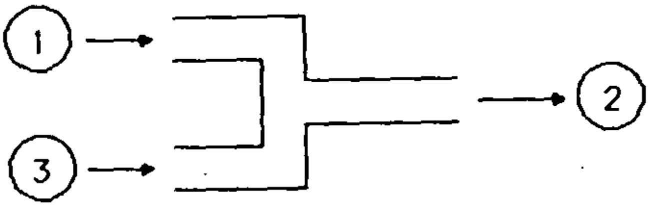


Figure 4.8 - a mixer

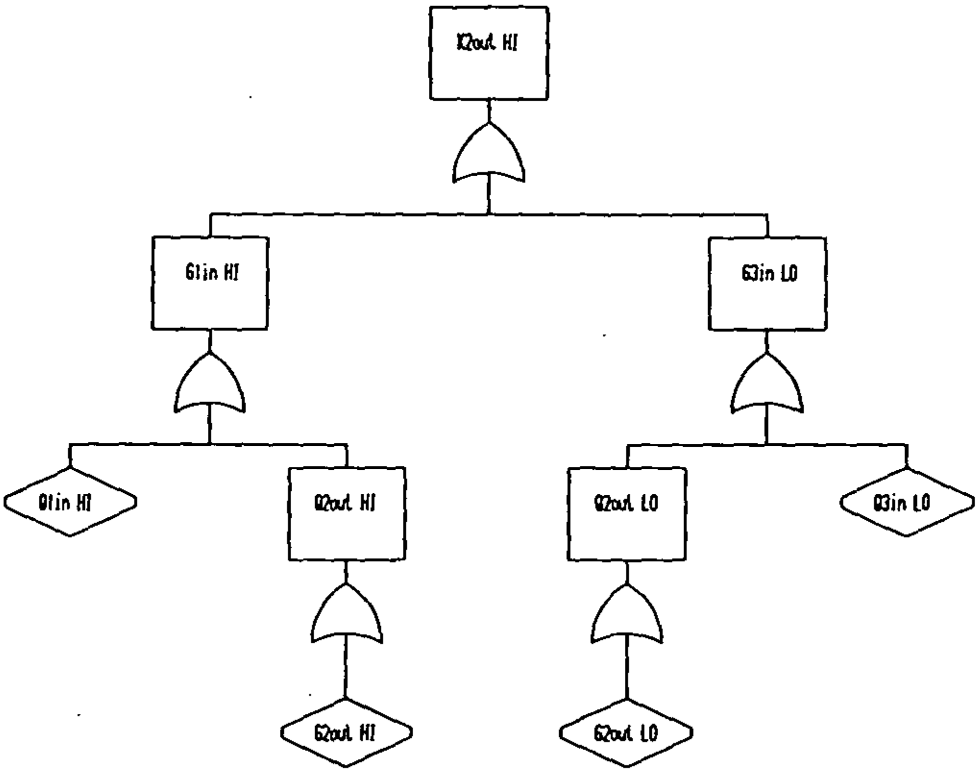


Figure 4.9 - the fault tree for high composition of component 1 at the outlet of the mixer (ignoring flow ratio)

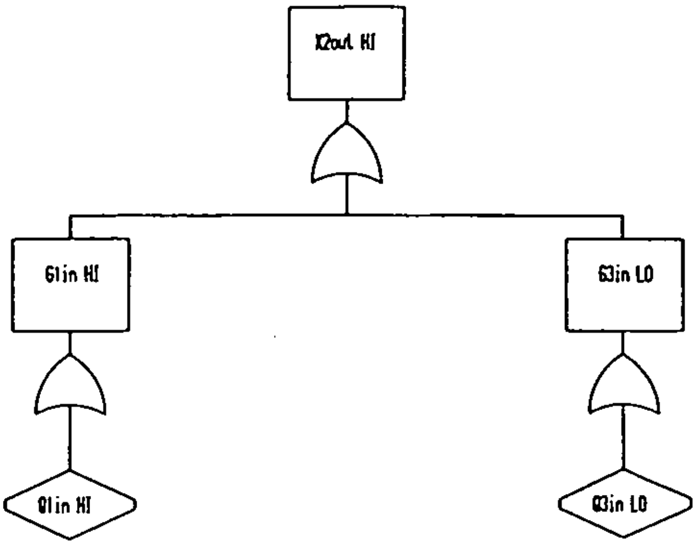


Figure 4.10 - the fault tree for high composition of component 1 at the outlet of the mixer (including flow ratio)

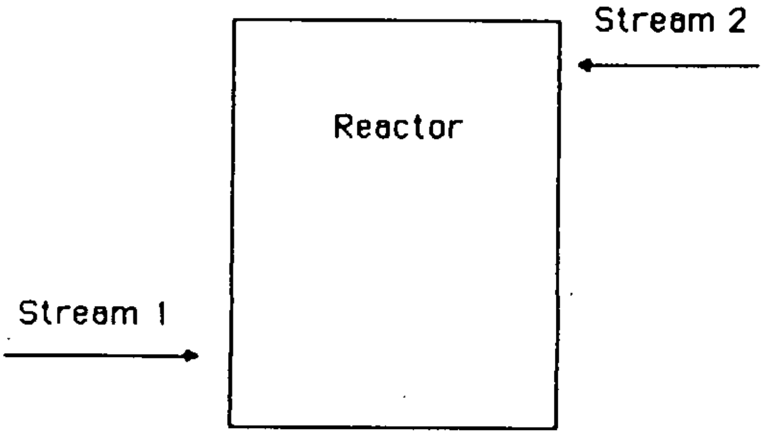


Figure 4.11 - a reactor

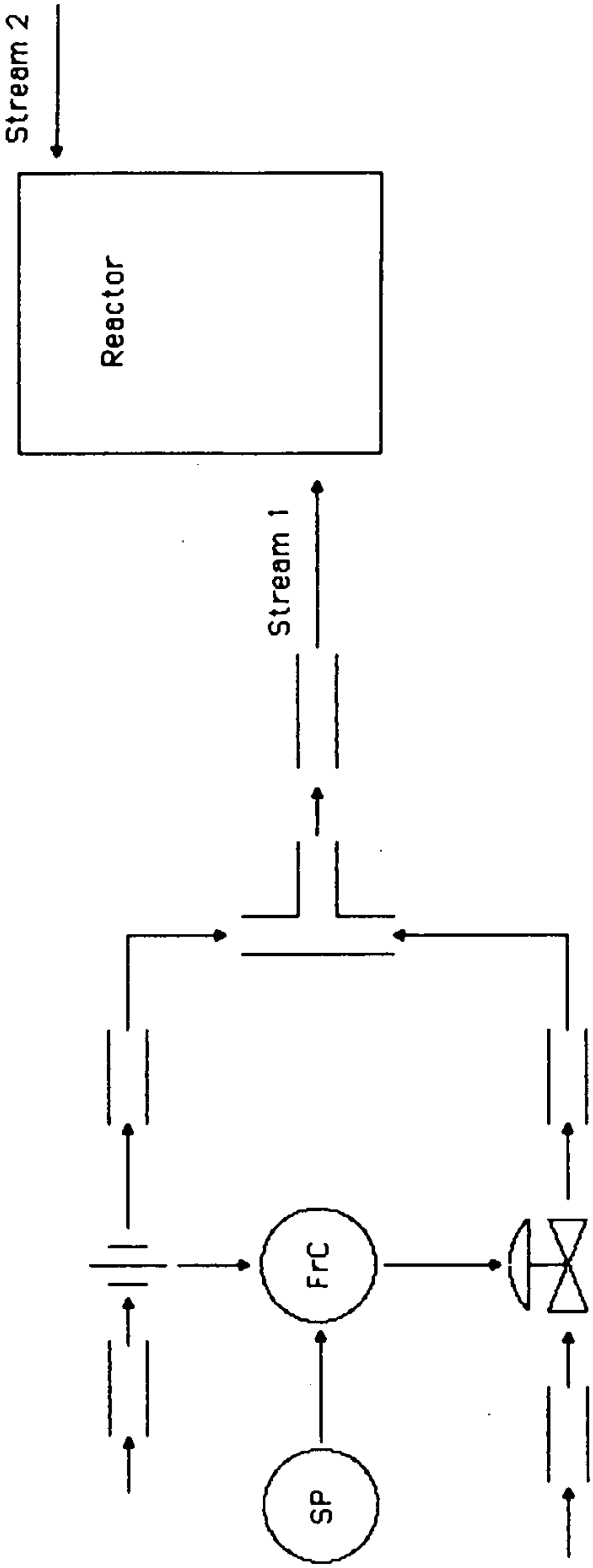


Figure 4.12 - the reactor, showing more detail of the system upstream

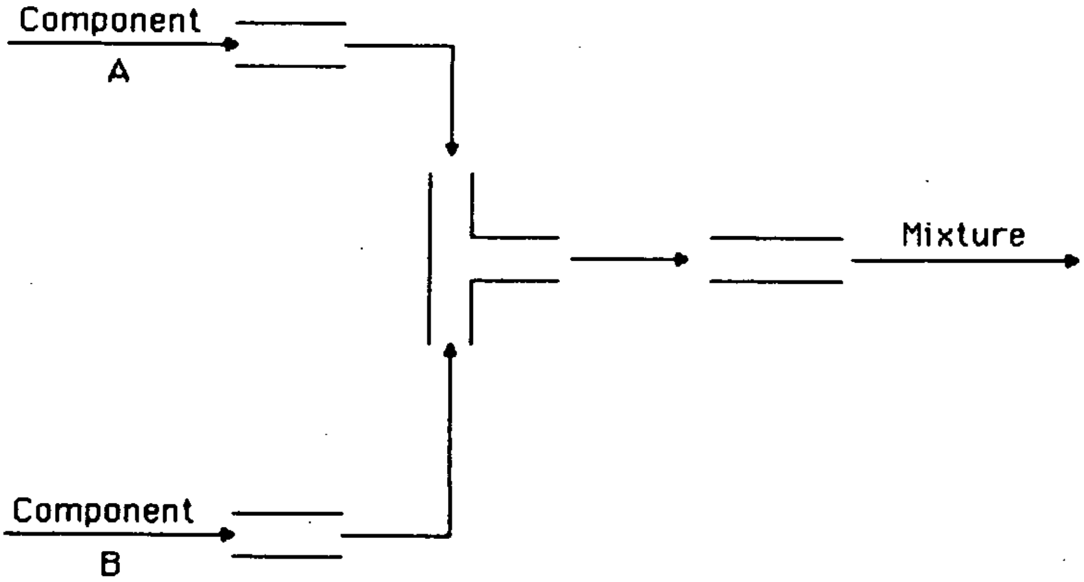


Figure 4.13 - a mixing system, after Taylor [43]

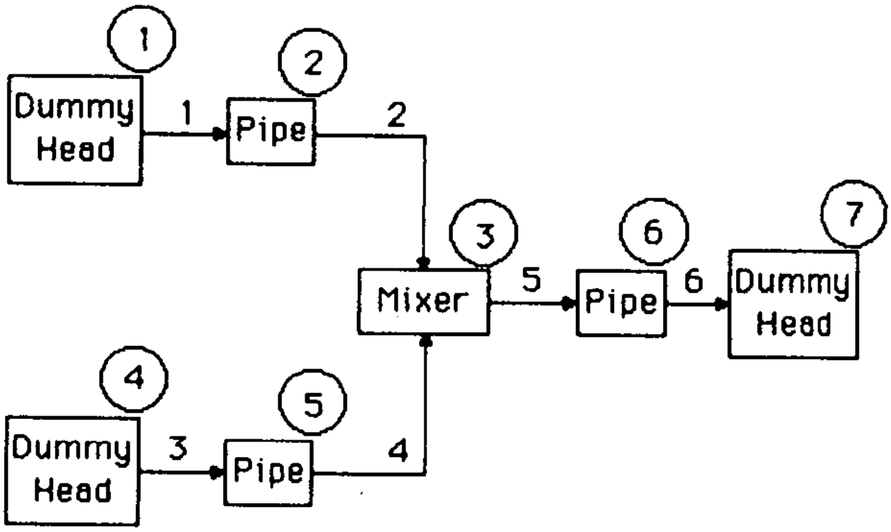


Figure 4.14 - the configuration diagram for the mixing system (Figure 4.13)

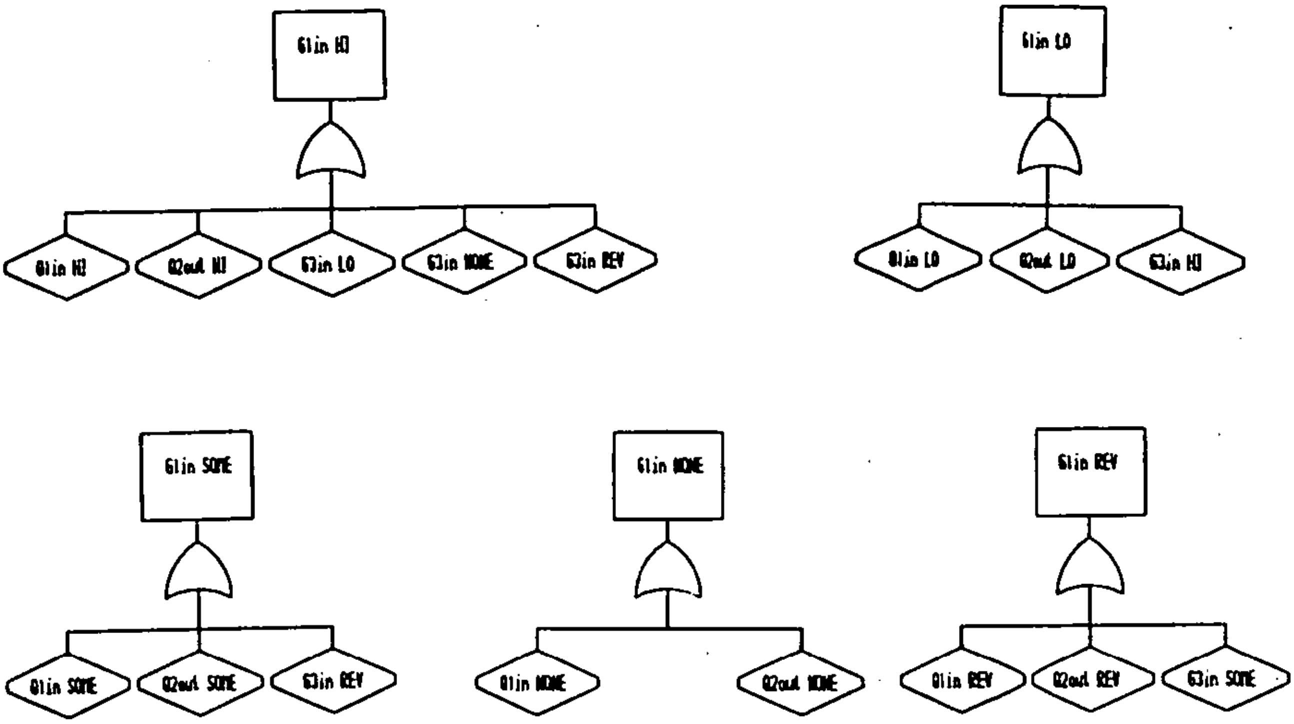


Figure 4.15 - minitrees for the deviations of G1in in a binary mixer

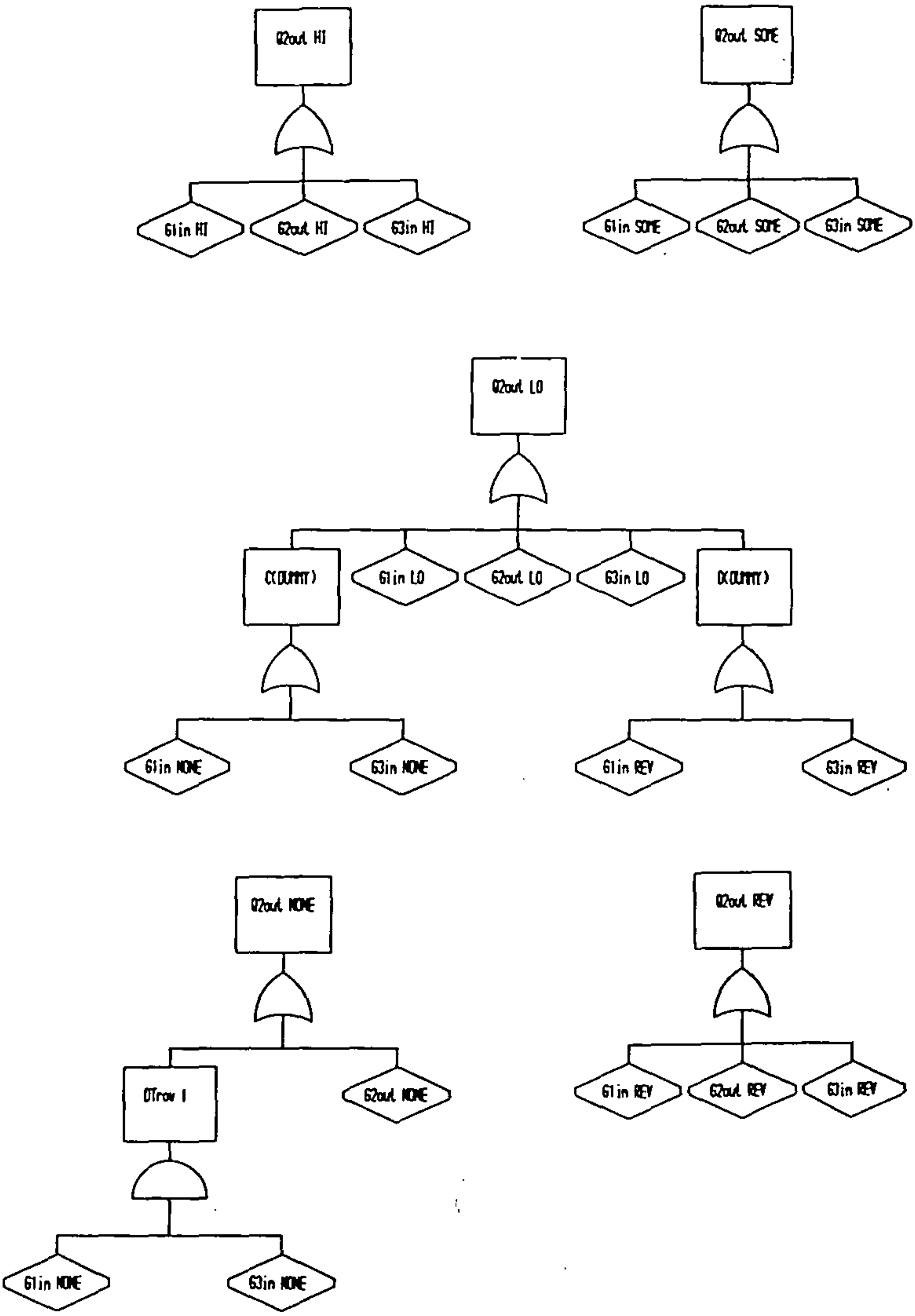


Figure 4.16 - minitrees for the deviations of Q2out in a binary mixer

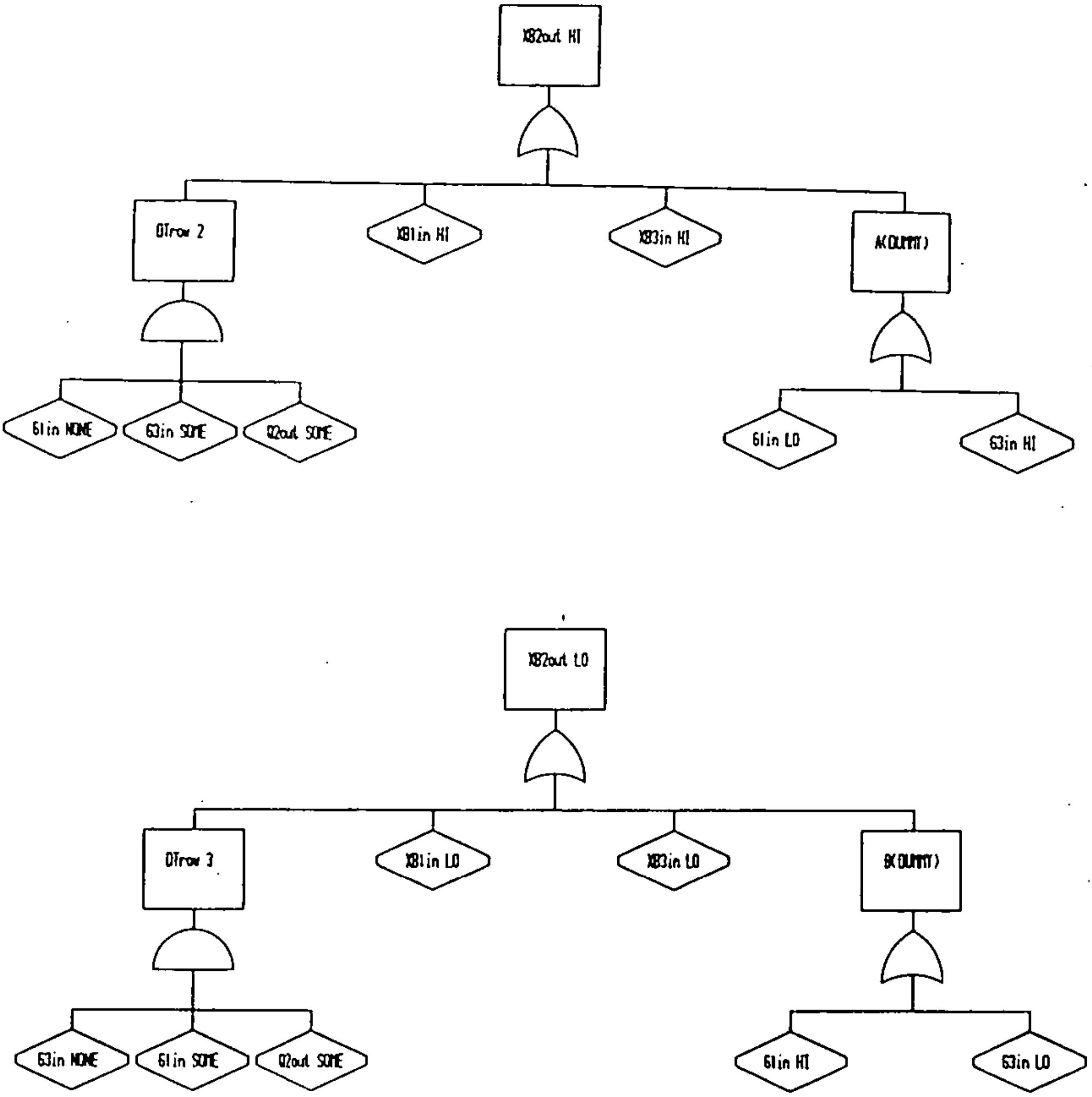


Figure 4.17 - minitrees for the deviations of XB2out in a binary mixer

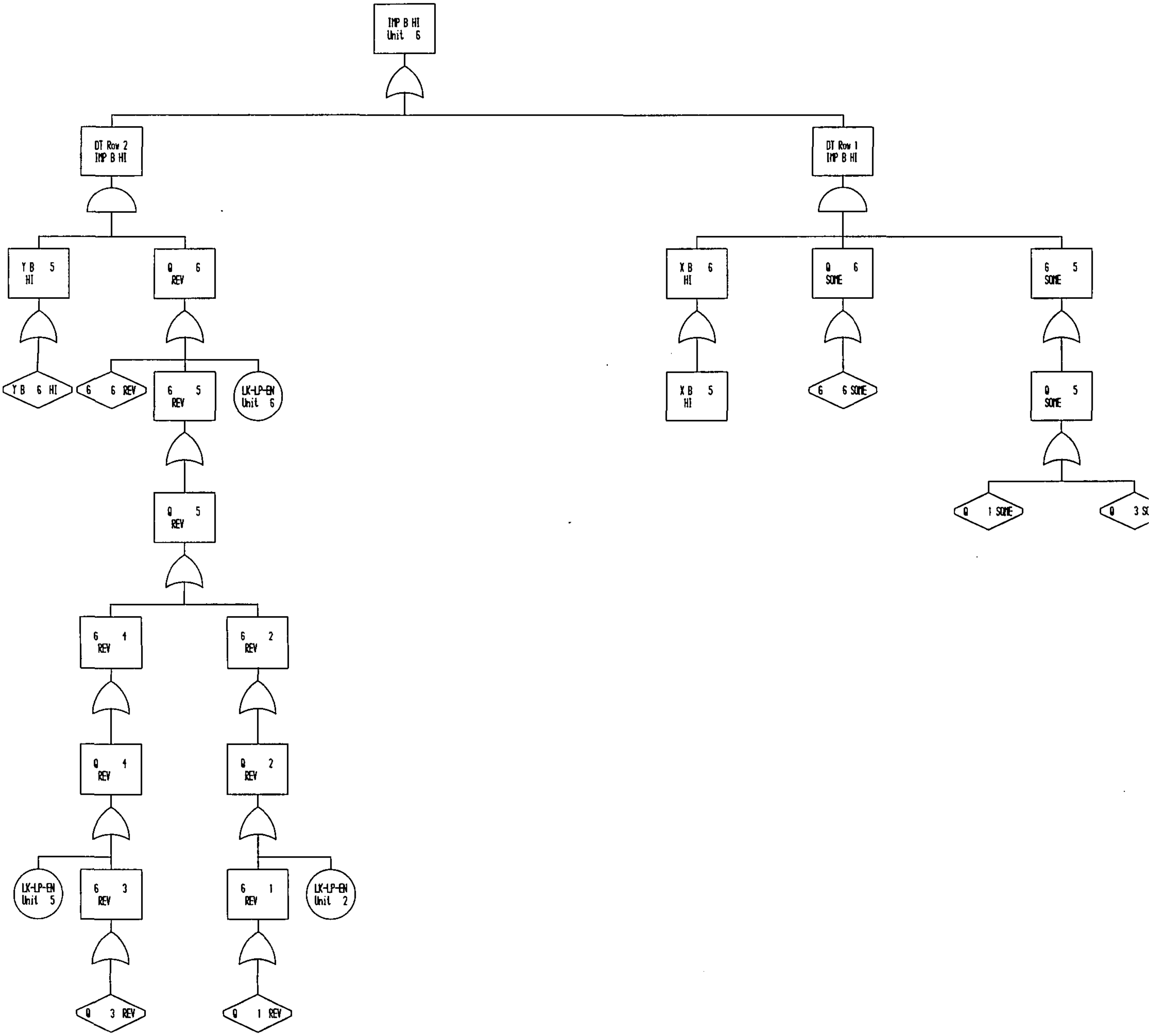


Figure 4.18 - partial fault tree for the [illegible] system shown in Figure 4.14

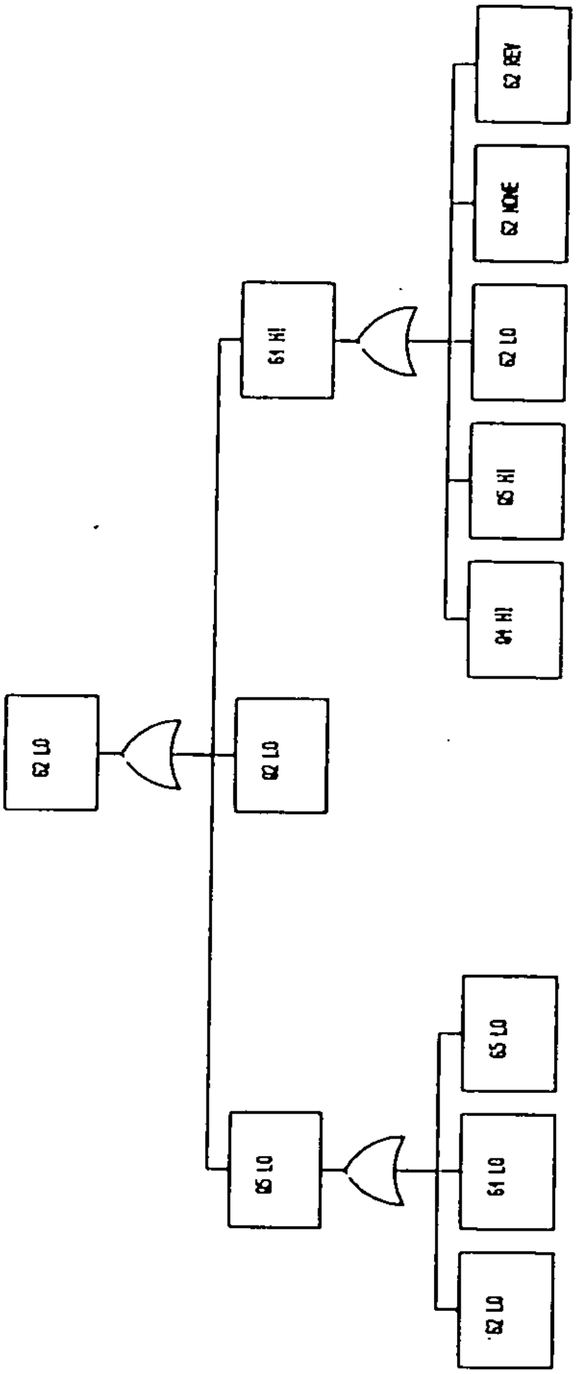


Figure 4.19 - causes of the event G2 LO for the system shown in Figure 4.14, including inconsistent causes

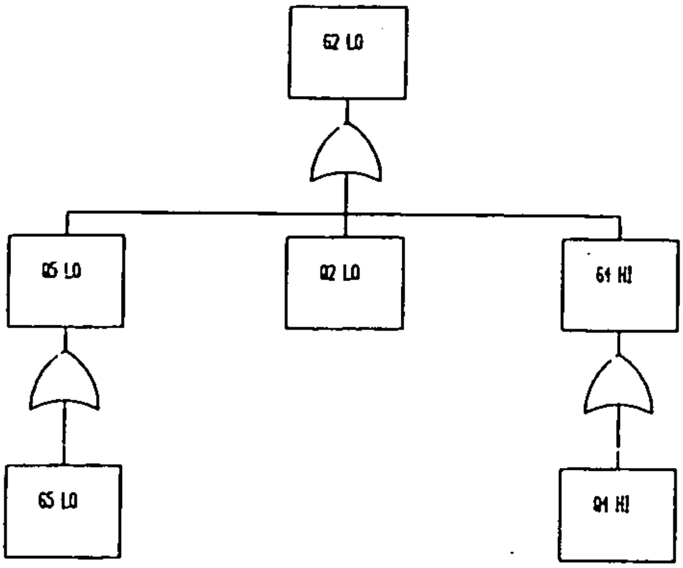


Figure 4.20 - causes of the event G2 LO for the system shown in Figure Figure 4.14, with inconsistent causes removed

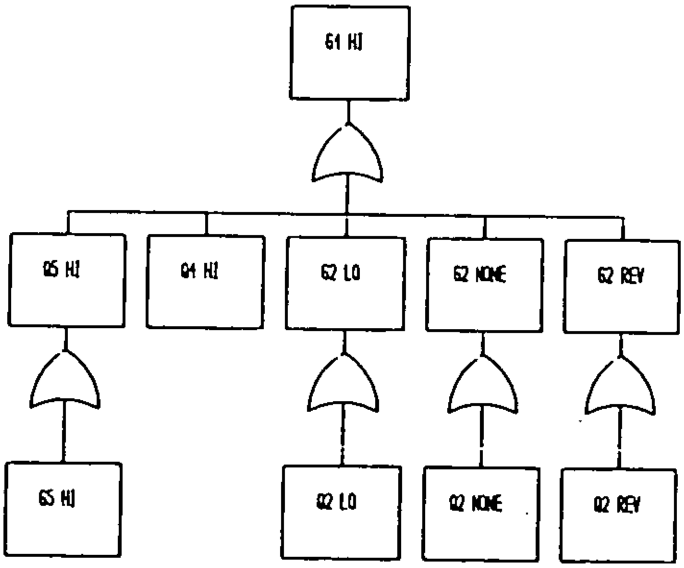


Figure 4.21 - causes of the event G4 HI for the system shown in Figure 4.14, with inconsistent causes removed

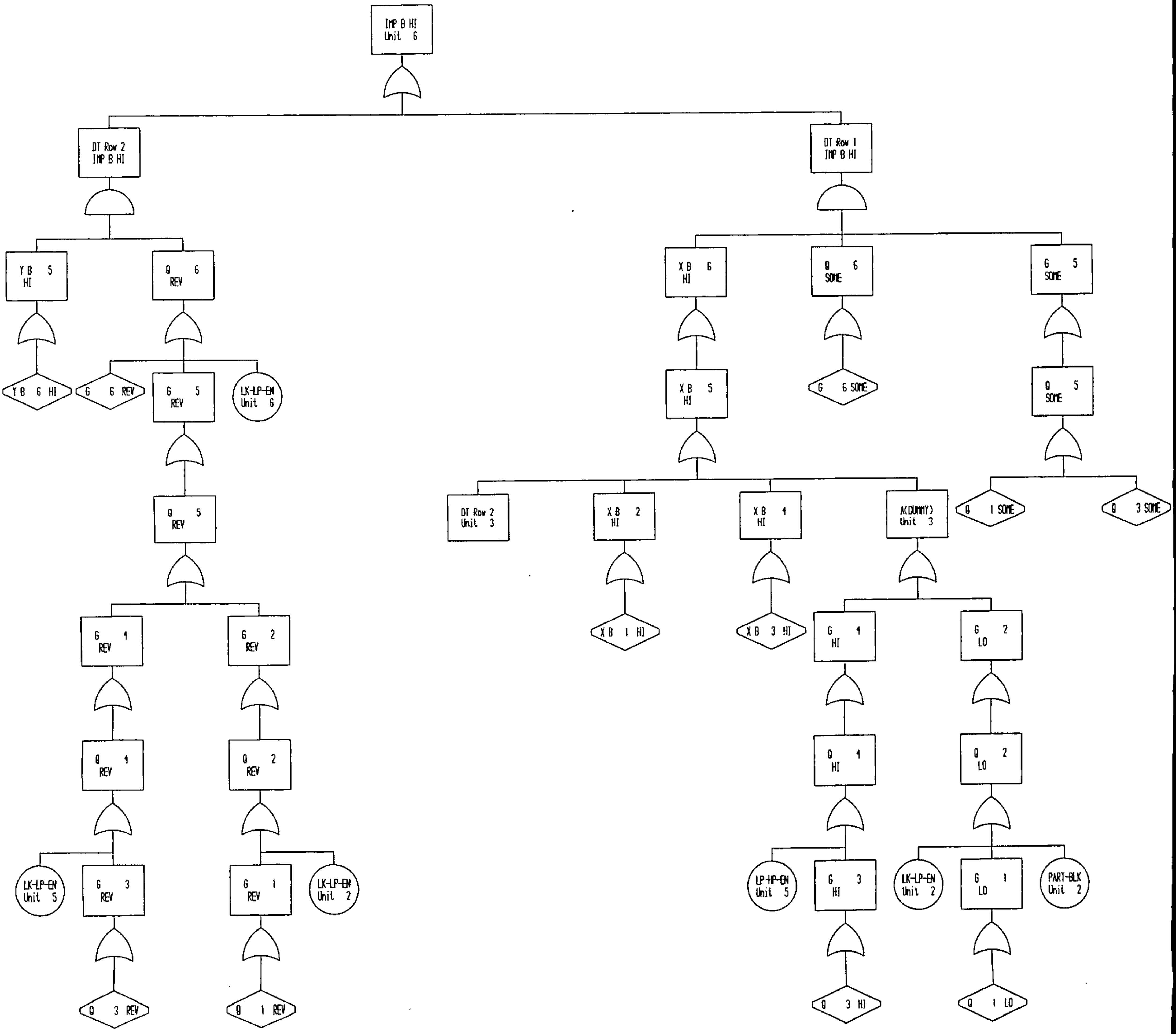


Figure 4.22 - partial fault tree for the system shown in Figure 4.14

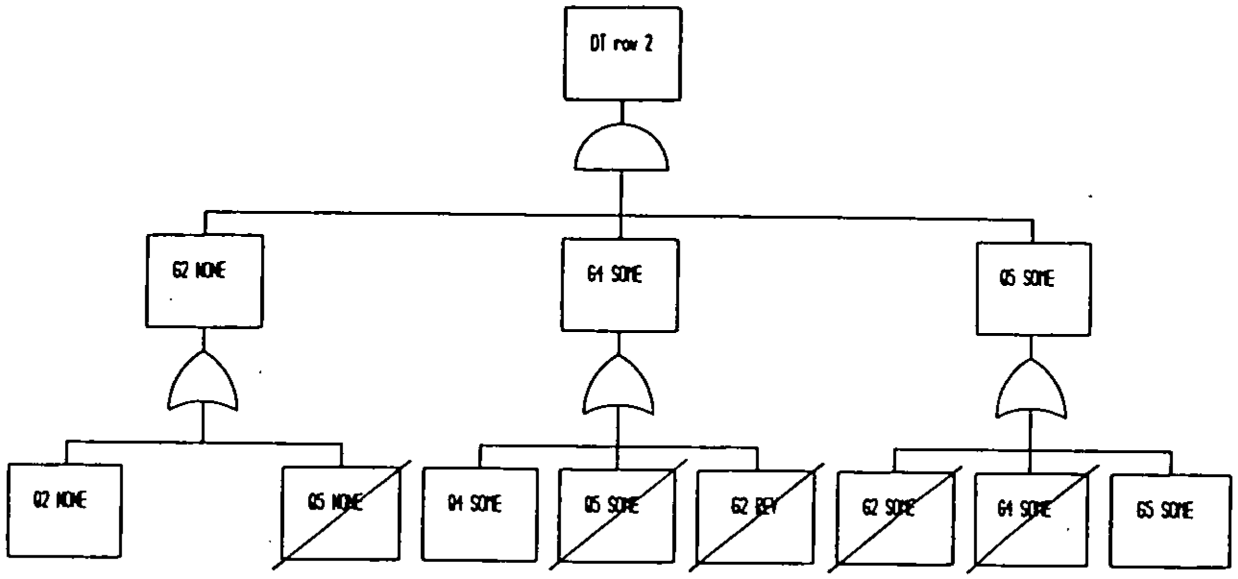


Figure 4.23 - causes of the event DT row 2 for the system shown in Figure 4.14, with inconsistent causes identified

Mixing System Example

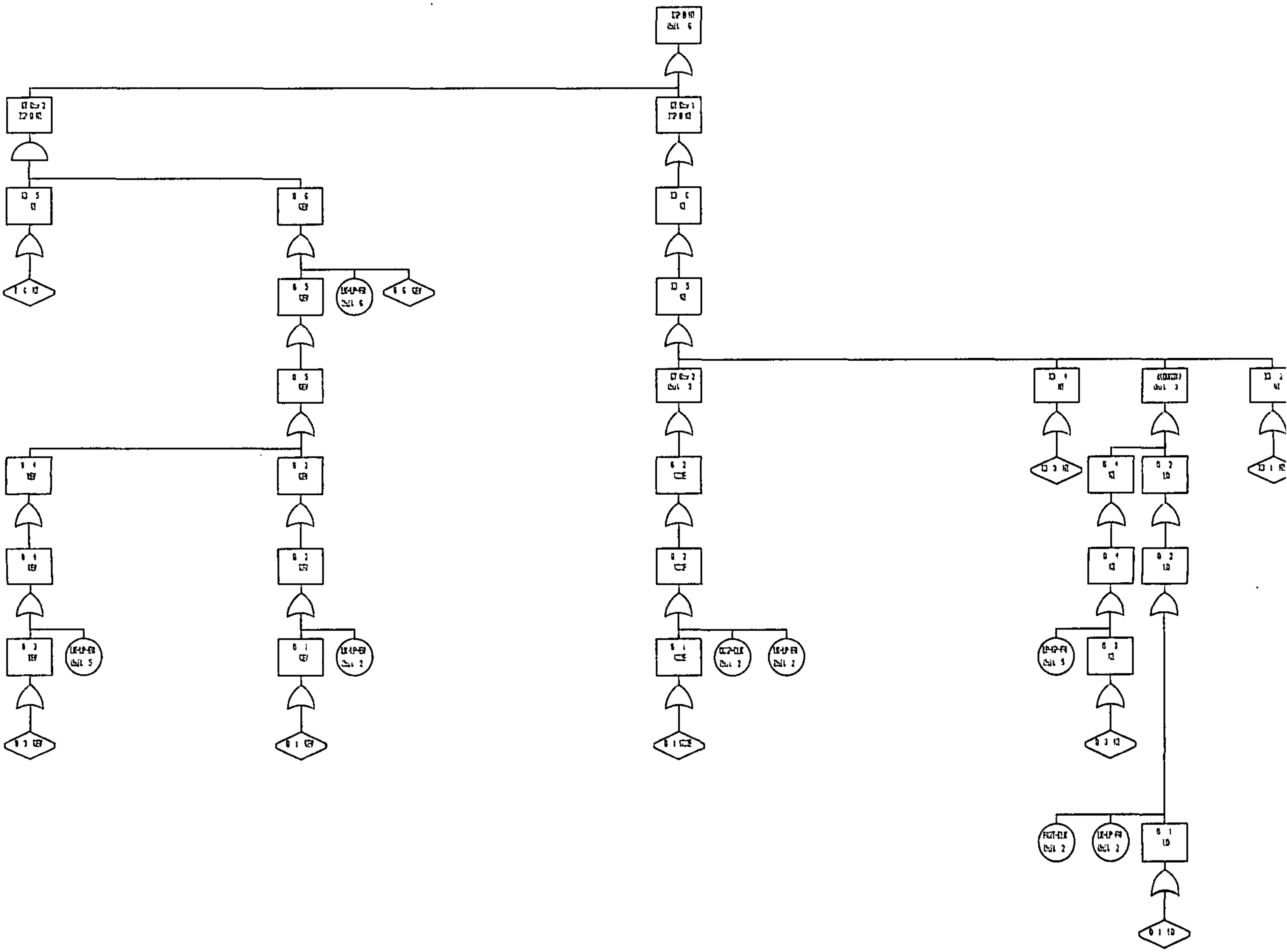


Figure 4.24 - complete fault tree for the system shown in Figure 4.14

5) The Modelling of Vessels

Vessels are defined as units in which there is an accumulation of either mass or energy (or both) which affects the deviations of variables at the inlet and/or outlet ports of that unit. The typical vessel is a tank, and it is recognisable as a vessel because the level within the tank affects the flow in and out of the tank.

To model vessels, it is necessary to be able to model the internal variables of importance. Since variables, by definition, exists only at the ports of a model, a new port type is required. This is the internal or vessel port, with identifier "ves". Like all other ports, this must be linked to some other unit, so that a connection number can be associated with the port. In many cases, the port will be linked to a sensor or an indicator, but, if this does not occur, then a dummy tail should be linked to the port.

There are only a limited number of variables that can be used to model internal events. These are the variables that are modelled at outlet ports, namely Q, T, X and P, plus the level variable, L. Q cannot be used at the same port as level. The use of flow as an internal variable is illustrated in the partial reboiler model (Section 5.3).

Vessel models are less general than pipe type models, because of the variety of duties and types of vessel. Even within a particular class of vessel, such as tanks, there is a variety of possible models. This chapter will illustrate the modelling of vessels by considering individual examples.

5.1) Closed Tank containing Supercooled Liquid

This is one of the simpler vessel models. A diagram is given in Fig 5.1. Port 1 is the inlet port, port 2 the outlet port, and port 3 the vessel port.

5.1.1) Events at Vessel Ports

There are four internal events that are generally of interest, namely level, pressure, temperature and composition. These will be considered in turn.

Three deviations of level are recognised - NONE, LO and HI, which correspond, respectively to no significant level of fluid in the tank, a low level in the tank, and a high level in the tank. Tank overflow is only of interest as the top event of a fault tree, and has the same causes has high level. The only difference is that these causes must exist for longer before overflow occurs. No level, however, is an important event, since the effect on the flow out of the tank differs from the effect of low tank level.

The propagation equation that models level is

```
L3ves=f(G1in,-Q2out)
```

Note the use of G1in, rather than Q1in, as the inlet flow variable. This is done to obey the restriction noted in Section 3.2.4.

This equation does not model the behaviour of the tank completely. Additional event statements must be given

```
V G1in NONE:L3ves LO
V G1in REV:L3ves LO,L3ves NONE
V Q2out NONE:L3ves HI
V Q2out REV:L3ves HI
V Q2out HI:L3ves NONE
```

These give rise to the minitrees that are given in Fig 5.2.

Vessel pressure is dependent only on the vessel level. As the level rises, so the pressure above the liquid also rises. The following propagation equation models this behaviour.

```
P3ves=f(L3ves)
```

Only the pressure states HI and LO are meaningful when applied to the pressure in a vessel, as opposed to the pressure at the inlet and outlet ports of the vessel.

Temperature and composition are basically straightforward, the only complications centring around reverse flow carrying temperature and composition deviations into the tank. A combination of propagation equations and decision tables are used

```
T3ves=f(T1in)
X3ves=f(X1in)
```

```
V U2out LO V Q2out REV T T3ves LO
V U2out HI V Q2out REV T T3ves HI
V Y2out LO V Q2out REV T X3ves LO
V Y2out HI V Q2out REV T X3ves HI
```

Note that the deviations of the reverse temperature and composition variables are ANDed with reverse flow. This is one of the models in which this AND gate must be explicitly specified (see Section 4.1.4). The reason for this is that, if the top event is, for example, high tank temperature, the AND gate can be included in no other way.

5.1.2) Events at Inlet and Outlet Ports

The events of interest at the inlet and outlet ports are the inlet and outlet flows, temperatures, compositions, pressures and reliefs. These will be considered in turn.

Inlet flow is dependent, at least to some extent, on the pressure in the tank. Whether or not the tank pressure has a significant effect on flow, or, to be more precise, whether or not changes in the tank pressure result in significant changes in the inlet flow, depends on the particular situation. If the tank pressure is very low compared to the fluid pressure in the inlet pipeline, then a change in tank pressure cannot affect the inlet flow by very much. On the other hand, if the tank pressure is almost as large as the fluid pressure, then pressure changes in the tank will significantly affect the inlet flow.

There are therefore two propagation equations that may be used for inlet flow, reflecting these two situations

G1in=f(Q1in)

or G1in=f(Q1in,-P3ves)

Note that both equations involve the term Q1in. This is because the start of the inlet is like a pipe, and must therefore involve the pipe type term. The first equation covers the situation where tank pressure does not affect the inlet flow; the second equation covers the situation where the tank pressure is important.

These equations are suitable for high and low flow deviations, and also contain all the information about how the upstream failures affects the inlet flow for the other flow deviations. However, the relationship between the tank pressure and no (G1in NONE) and reverse (G1in REV) inlet flows, if any, is not in these equations. If the tank pressure does not affect the inlet flow, then deviations in tank pressure can never cause no or reverse inlet flow. The causes of these events will always be upstream. However, if tank pressure is important, then sufficiently high tank pressure may result in no and even reverse inlet flows. In addition to a high pressure, if the inlet pipe is not normally submerged, reverse flow will also require a high tank level to submerge the inlet port. Since pressure and level are synonomous in this example, this last proviso can be ignored, since if high pressure exists, a high level will also exist.

G1in SOME should always be specified as a normal state, since it does not require a fault condition to cause some flow into the tank.

To include all these deviations correctly, the following event statements must be combined with the appropriate propagation equation

```
V P3ves HI:G1in NONE
V P3ves HI:G1in REV
S NORMAL:G1in SOME
```

The first two statements are used only if tank pressure can cause no or reverse inlet flow.

The outlet flow relationships are similarly dependent on the detail of the tank. There are two driving forces for flow out of the tank, the tank pressure, and the tank level. In the current example, tank pressure is related solely to the tank level, and so it is not an independent factor. Effectively, therefore, tank level is the only driving force for flow out of the tank. As with inlet flow, tank level may or may not be an important factor in changes in the outlet flow. Two propagation equations are required to cover these two situations

```
    Q2out=f(L3ves,G2out)
or  Q2out=f(G2out)
```

The first equation, where level is an important influence, will be appropriate for gravity feed tanks. The second equation will be appropriate where there is a pump connected to the tank outlet. The pump will have a far greater influence on the outlet flow than the tank level, over a wide range of levels.

As with inlet flow, these equations do not model the complete behaviour of outlet flow. No outlet flow (Q2out NONE) will result if the tank is empty, and there is no inlet flow, whether or not the tank is modelled with outlet flow independent of tank level. Some outlet flow (Q2out SOME), like some inlet flow, is a normal state, since it exists in the absence of any faults in the tank. Reverse flow at the outlet (Q2out REV) is more difficult to model, and may depend on the tank level. A more conservative approach is to model the tank such that, provided a reverse flow driving force exists downstream, then reverse flow into the tank is possible.

The event statements and decision tables required in addition to the appropriate propagation equation are

S NORMAL:Q2out SOME
V Q2out REV AND S NORMAL:Q2out REV

V L3ves NONE V G1in NONE T Q2out NONE

The event statement for reverse flow can safely include the AND gate, since this is the only information that appears in the minitree for reverse outlet flow. A decision table is required for no outlet flow, since another cause of no outlet flow is derived from the propagation equation, which is no flow downstream (G2out NONE).

The event statement for reverse flow is slightly curious at first sight. It is similar in intent to the standard flow propagation equations, in which only half the information is used when propagating faults in one direction. When propagating reverse outlet flow from a downstream effect to causes in the tank, the NORMAL status event applies, since the G2out REV event will be removed by the internal consistency conditions checks. The tank therefore has the potential to accept reverse flow. When propagating reverse flow out of the tank, the G2out REV event is not deleted, and so the status event is effectively ignored. G2out REV traces the causes of reverse flow to faults downstream of the tank. Fault tracing in this direction will occur when investigating, for instance, how a high tank temperature could be caused by a combination of reverse flow and high downstream temperature.

Temperature and composition are straightforward. The values at the inlet and outlet ports are the same as the internal values. This applies to both the normal temperature and composition variables, and to the reverse flow variables. The following propagation equations model this behaviour

T2out=f(T3ves)
X2out=f(X3ves)
U1in=f(T3ves)
Y1in=f(X3ves)

The AND gate need not be explicitly specified with the reverse temperature and composition deviations at the inlet port, since these events will not appear at the start of a new fault tree branch, but will be part of a branch tracing the causes of a temperature or composition deviation propagating upstream under reverse flow conditions. The AND gate implicit in such a branch will already have been included in the fault tree.

Pressure and relief at the inlet and outlet ports must be modelled to the conventions noted when modelling pressure and relief in pipe type units (Section 4.3), and are not the same as the internal tank pressure. The modelling of internal tank pressure was covered in Section 5.1.1.

As with inlet flow, relief into the tank is, in principle, dependent on the pressure in the tank.

However, as was noted when considering inlet flow, the difference in pressure between the pipeline and the tank may be so large that changes in tank pressure will have negligible effects on the inlet flow. The same is true of relief into the tank. High relief and low relief may therefore be dependent on the tank pressure, or they may be independent of tank pressure. No relief into the tank (R1in NONE) is very similar to no inlet flow. It may or may not be possible to increase the tank pressure such that no relief into the tank is possible. Some relief into the tank (R1in SOME), like some inlet flow, is a normal state. R1in REV, corresponding to a back pressure out of the tank, may, like no inlet relief, be impossible to obtain, or may be caused by a high tank pressure. R1in NOP, corresponding to no back pressure out of the tank, is the opposite of R1in REV, and can usually be considered a normal state.

Outlet pressure bears a close similarity to outlet flow. The tank level and pressure may be unimportant in determining the pipeline pressure, if, for instance, there is a pump downstream, or they may be important factors. No outlet pressure (P2out NONE) corresponds to an empty tank and no inlet flow, and so has the same causes as no outlet flow. Some outlet pressure (P2out SOME) is, like some outlet flow, a normal state. P2out REV, corresponding to relief into the tank from downstream, can be regarded as a normal state. There are not, as there were with reverse flow, complications about the direction of fault propagation, because of the way the two way propagation structure of pressure and relief is modelled. P2out NOR, corresponding to no relief into the tank from downtream is the opposite of P2out REV, and is usually regarded as an impossible state, i.e. a state that can never exist in a unit, regardless of the faults that can occur. Used to model P2out NOR it indicates that, no matter what the level in the tank, relief into the tank from downstream is always possible.

The propagation equations, event statements and decision tables that model this behaviour are

R1in=f(-P3ves)

or no propagation equation for R1in

P2out=f(L3ves)

or no propagation equation for P2out

V P3ves HI:R1in NONE,R1in REV

or no causes of R1in NONE and R1in REV

S NORMAL:R1in SOME,R1in NOP,P2out SOME,P2out REV
S IMPOSS:P2out NOR

V L3ves NONE V G1in NONE T P2out NONE

5.1.3) Summary

The following list of propagation equations, event statements and decision tables is the complete model for a closed tank containing supercooled liquid. It has been assumed in this summary that the level has an effect on the inlet flow, but not on the outlet flow.

G1in=f(Q1in,-P3ves)
R1in=f(-P3ves)
U1in=f(T3ves)
Y1in=f(X3ves)

```
Q2out=f(G2out)
T2out=f(T3ves)
X2out=f(X3ves)
L3ves=f(G1in,-Q2out)
T3ves=f(T1in)
X3ves=f(X1in)
P3ves=f(L3ves)

V G1in NONE:L3ves LO
V G1in REV:L3ves LO,L3ves NONE
V Q2out NONE:L3ves HI
V Q2out REV:L3ves HI
V Q2out HI:L3ves NONE
V P3ves HI:G1in NONE,R1in NONE
V P3ves HI:G1in REV,R1in REV
V Q2out REV AND S NORMAL:Q2out REV
S NORMAL:G1in SOME,Q2out SOME
S NORMAL:R1in SOME,R1in NOP,P2out SOME,P2out REV
S IMPOSS:P2out NOR

V L3ves NONE V G1in NONE T Q2out NONE,P2out NONE
V U2out LO V Q2out REV T T3ves LO
V U2out HI V Q2out REV T T3ves HI
V Y2out LO V Q2out REV T X3ves LO
V Y2out HI V Q2out REV T X3ves HI
```

5.2) Binary Distillation Column

See Fig 5.3. Port 1 is the feed, port 2 is the distillate takeoff, port 3 the bottoms takeoff. Port 4 is the reflux, and port 5 the boilup. Port 6 is a vessel port relating to the distillate, and port 7 a vessel port relating to the bottoms.

This model is separate from the reboiler and condenser models that are invariably associated with distillation columns.

The distillation separates component 'A' (the more volatile component) from component 'B'. The feed is assumed to be a vapour at its boiling point.

Because the system involves vapour and liquid in equilibrium, the three variables temperature, pressure and compostion are not independent. If any two are known, then the third is fixed. In this model, the temperatures at the top and bottom of the column are expressed in terms of the compositions and pressures that exist. Compositions are modelled using distillation equations, and pressure is obtained by a vapour balance on the column.

Reverse flow effects, and pressure and relief at the inlet and outlet ports will not be included in this model, for simplicity. The modelling of these is very similar to the modelling in the tank model already considered (Section 5.1).

5.2.1) Events at Vessel Ports

The events of interest inside the column are the level (L7ves), the pressure (P6ves), the temperatures of both the liquid and vapour phases (T6ves and T7ves), and the compositions of each phase (XA6ves and XB7ves). Only one component is defined for each phase. The composition of the other component can be deduced from this, since there are only two components in the column. The component modelled in each phase is the desired component of that phase, that is the more volatile component (A) in the distillate, and the less volatile component in the bottoms.

Column level is simply obtained from the liquid accumulation in the bottom of the column

L7ves=f(G4in,-Q3out)

Vessel pressure is the vapour accumulation within the column

P6ves=f(G1in,G5in,-Q2out)

As with the tank models, this information needs to be supplemented with some event statements.

```
V G1in NONE:P6ves LO
V Q2out NONE:P6ves HI
V Q3out NONE:L7ves HI
V G4in NONE:L7ves LO
V G5in NONE:P6ves LO
V Q3out HI:L7ves NONE
```

Tops composition is related to the composition of the vapour input to the column, and the reflux ratio.

The streams flowing into the column are the inlet, and the boilup. The inlet composition is simply XA1in, but the boilup composition is best modelled using the composition at the bottom of the column, XB7ves. The reflux ratio is the ratio of the reflux to the takeoff, and can be modelled as G4in/Q2out. As noted in Section 4.5, it is recommended that flow ratio causes should be the sole cause of an event. To achieve this in the model, a dummy variable can be used. Components A and B are used already, and so component C is suitable. XC6ves is therefore used to model the reflux ratio. The two propagation equations needed are

```
XA6ves=f(XC6ves,XA1in,-XB7ves)
XC6ves=f(G4in/Q2out)
```

A cause of XA6ves LO not modelled by these propagation equations is complete loss of reflux (G4in NONE). It is necessary to AND this with some distillate takeoff, otherwise, no flow of distillate will be identified as a potential cause of low reflux ration. This information can be included using a decision table

```
V G4in NONE V Q2out SOME T XA6ves LO
```

Bottoms composition is similarly related to the composition of the liquid input to the column, and the boilup rate. The only liquid input to the column is the reflux, and the composition of this is best modelled by XA6ves. XC7ves can be used to model the boilup ratio

```
XB7ves=f(XC7ves,-XA6ves)
XC7ves=f(G5in/Q3out)
```

Complete loss off boilup may cause low product purity

V G5in NONE V Q3out SOME T XB7ves LO

The temperatures in the column are dependent on the compositions and the pressure.

T6ves=f(-XA6ves,P6ves)
T7ves=f(XB6ves,P6ves)

5.2.2) Events at Inlet and Outlet Ports

The flows in and out of the column are very similar to the flows in and out of the tank model. However, the problems associated with whether the column conditions will affect specific inlet and outlet flows are not so great, since distillation columns are more standardised than tanks. The following description will be used to select the appropriate flow equations. If, for a different column, the description is different, an alternative set of equations will have to be used.

The feed to the column is pumped. Column pressure will therefore have little effect on the inlet flow. The reflux to the column is also pumped. However, all the other flows are driven by vapour pressures, and the conditions in the column can be assumed to have a significant influence on the flows. The following equations model this behaviour

```
G1in=f(Q1in)
Q2out=f(P6ves,G2out)
Q3out=f(P6ves,L7ves,G3out)
G4in=f(Q4in)
G5in=f(Q5in,-P6ves)
```

As in the tanks, additional event statements are necessary to completely model the column

```
S NORMAL:G1in SOME,G4in SOME,G5in SOME
V G2out SOME AND S NORMAL:Q2out SOME
V G3out SOME AND S NORMAL:Q3out SOME
V P6ves HI:G5in NONE
```

The reason that the two outlet flow deviations SOME are ANDed with the normal state is the same as the reason why reverse outlet flow in the tank model (see Section 5.1.2), namely, that the SOME outlet flow deviation can start either from within the tank (through the decision tables for composition deviations), or from outside the column. If the deviation is from outside, then some flow out of the column is the normal state. If, however, the deviation is some flow out of the tank, then a flow path downstream is required. Event statements can be used to include the AND gate, since there are no other causes of the SOME flow deviations.

Temperatures and compositions at the outlet ports are directly related to the relevant internal state

T2out=f(T6ves)
XA2out=f(XA6ves)
XB2out=f(-XA6ves)
T3out=f(T7ves)
XA3out=f(-XB7ves)
XB3out=f(XB7ves)

5.2.3) Summary

G1in=f(Q1in)
Q2out=f(P6ves,G2out)
T2out=f(T6ves)
XA2out=f(XA6ves)
XB2out=f(-XA6ves)
Q3out=f(P6ves,L7ves,G3out)
T3out=f(T7ves)
XA3out=f(-XB7ves)
XB3out=f(XB7ves)
G4in=f(Q4in)
G5in=f(Q5in,-P6ves)

P6ves=f(G1in,G5in,-Q2out)
T6ves=f(P6ves,-XA6ves)
XA6ves=f(XC6ves,XA1in,-XB7ves)
XC6ves=f(G4in/Q2out)
L7ves=f(G4in,-Q3out)
T7ves=f(P6ves,XB7ves)
XB7ves=f(XC7ves,-XA6ves)
XC7ves=f(G5in/Q3out)

```
S NORMAL:G1in SOME,G4in SOME,G5in SOME
V G2out SOME AND S NORMAL:Q2out SOME
V G3out SOME AND S NORMAL:Q3out SOME
V P6ves HI:G5in NONE
V G1in NONE:P6ves LO
V Q2out NONE:P6ves HI
V Q3out NONE:L7ves HI
V G4in NONE:L7ves LO
V G5in NONE:P6ves LO
V Q3out HI:L7ves NONE

V G4in NONE V Q2out SOME T XA6ves LO
V G5in NONE V Q3out SOME T XB7ves LO
```

5.3) Partial Reboiler

A diagram of this model is given in Fig 5.4. Port 1 is the inlet stream. Port 2 is the liquid takeoff, and port 3 the vapour takeoff. Port 4 is the heating medium inlet, and port 5 its outlet. Port 6 is a vessel port relating to the liquid, and port 7 a vessel port relating to the vapour. Port 8 is used to model the internal flow that exists between the liquid and vapour phases. The variable Q8ves is used to model the flow from the liquid to the vapour. Q8ves HI therefore corresponds to a high boilup rate of liquid.

As with all vessel models, some assumptions must be made about how the vessel pressure and level affect the inlet and outlet flows. The assumptions made in this model are that the vessel pressure affects the inlet and the vapour takeoff flows, but that the liquid takeoff is unaffected by either the level or the pressure.

5.3.1) Events at Vessel Ports

Vessel level is simply a liquid balance on the reboiler.

L6ves=f(G1in,-Q2out,-Q8ves)

Vessel pressure is similarly a vapour balance on the unit

P7ves=f(Q8ves,-Q3out)

As with the previous models, these equations must be supplemented to indicate the effects of other deviations of flow.

```
V G1in NONE:L6ves LO
V Q2out NONE:L6ves HI
V Q3out NONE:P7ves HI
V Q8ves NONE:L6ves HI,P7ves LO
V Q2out HI:L6ves NONE
```

The vessel compositions are dependent on the boilup rate and the input compositions. The direction of flow can be regarded as from inlet port 1, into the vessel liquid port, and hence to the vapour vessel port. The input composition for the liquid is therefore the inlet composition, and the input composition for the vapour is the liquid vessel composition. The effect of the boilup rate on the compositions can be deduced from vapour liquid equilibrium. The vapour composition will contain the purest more volatile component at the lowest boilup rate. The liquid will contain more of the more volatile component as the boilup rate increases. These relationships are summed up in the following propagation equations

```
XB6ves=f(XB1in,Q8ves)
XA7ves=f(-XB6ves,-Q8ves)
```

Note that, as in the distillation column, the component modelled in each phase is the desired component of that phase.

The temperatures can be deduced from the compositions and the pressure

```
T6ves=f(P7ves,XB6ves)
T7ves=f(P7ves,-XA7ves)
```

The boilup rate can be calculated heuristically. The boilup rate will increase if the heating medium temperature or flow rate increases, or if the composition of the liquid becomes higher in terms of the more volatile component, or if the vessel pressure drops. The following equation models this behaviour

```
Q8ves=f(G4in,T4in,-XB6ves,-P7ves)
```

The boilup rate will be reduced if the rate of heat transfer is reduced. There are several causes of this. Firstly there is fouling of the exchanger coils. Another cause is frothing of the liquid phase. Both these causes are modelled as basic events. It is assumed that fouling can eventually result in a complete loss of boilup, but that frothing is less serious. A third cause is low liquid level, which will result in a low heat transfer surface area if the exchanger coil is not completely covered with liquid. The event statements to model this behaviour are

```
F FOULING:Q8ves LO,Q8ves NONE
F FROTHING:Q8ves LO
V L6ves LO:Q8ves LO
V L6ves NONE:Q8ves NONE
```

5.3.2) Events at Inlet and Outlet Ports

The events at the inlet and outlet ports 1, 2 and 3 present no problems that have not been considered under the two previous vessel models (see Sections 5.1.2 and 5.2.2). The relevant equations will simply be stated

```
G1in=f(Q1in,-P7ves)
Q2out=f(G2out)
T2out=f(T6ves)
XA2out=f(-XB6ves)
XB2out=f(XB6ves)
Q3out=f(L6ves,P7ves,G3out)
T3out=f(T7ves)
XA3out=f(XA7ves)
XB3out=f(-XA7ves)

V P7ves HI:G1in NONE
S NORMAL:G1in SOME,Q2out SOME,Q3out SOME
```

The events that relate to the heating medium are closely related to the events in a standard heat exchanger model. The flow faults are modelled using the standard pipe type flow equations, but the temperature requires a slightly different equation that more accurately models the reboiler. Appropriate equations are

```
G4in=f(Q4in,Q5out)
Q5out=f(G4in,G5out)
T5out=f(G4in,T4in,T6ves)
XC5out=f(XC4in)

F FOULING:T5out HI
F FROTHING:T5out HI
```

```
  L6ves LO:T5out HI
V L6ves NONE:T5out HI
```

Note that the composition variable used must have a subscript. This is because the model is a multiple component model, and so all composition variables must have subscripts. C is used to avoid confusion with the components in the reboiler.

The basic faults and level deviations that affect the boilup rate also affect the outlet temperature of the heating medium, as indicated by the event statements above.

5.3.3) Summary

```
G1in=f(Q1in,-P7ves)
Q2out=f(G2out)
T2out=f(T6ves)
XA2out=f(-XB6ves)
XB2out=f(XB6ves)
Q3out=f(L6ves,P7ves,G3out)
T3out=f(T7ves)
XA3out=f(XA7ves)
XB3out=f(-XA7ves)


G4in=f(Q4in,Q5out)
Q5out=f(G4in,G5out)
T5out=f(G4in,T4in,T6ves)
XC5out=f(XC4in)
L6ves=f(G1in,-Q2out,-Q8ves)
T6ves=f(P7ves,XB6ves)
XB6ves=f(XB1in,Q8ves)
```

```
P7ves=f(Q8ves,-Q3out)
T7ves=f(P7ves,-XA7ves)
XA7ves=f(-XB6ves,-Q8ves)
Q8ves=f(G4in,T4in,-XB6ves,-P7ves)

V P7ves HI:G1in NONE
F FOULING:Q8ves LO,Q8ves NONE,T5out HI
F FROTHING:Q8ves LO,T5out HI
V L6ves LO:Q8ves LO,T5out HI
V L6ves NONE:Q8ves NONE,T5out HI
V G1in NONE:L6ves LO
V Q2out NONE:L6ves HI
V Q3out NONE:P7ves HI
V Q8ves NONE:L6ves HI,P7ves LO
V Q2out HI:L6ves NONE
S NORMAL:G1in SOME,Q2out SOME,Q3out SOME
```

5.4) Fault Tree Synthesis - An Example

This section considers the fault synthesis for a distillation column, which involves several vessels. The flow diagram for the system based on a column in a paper presented by Shepherd et al [53], is given in Fig 5.5. The column is a binary distillation column, with saturated vapour feed, and is intended to separate component A, the more volatile component, from component B. The column has ancillary equipment, which are a partial reboiler, a total condenser and a condensate reflux tank.

5.4.1) Decomposition

Decomposition includes all the equipment shown in Fig 5.5. The configuration diagram is shown in Fig 5.6. The only point to note is that all the vessel ports are linked to dummy tails. This is to provide a connection number for each of the vessel ports.

5.4.2) Unit Modelling

The vessel models used in this study are the models described earlier in this chapter. The distillation column model is given in Section 5.2. The reboiler model is considered in Section 5.3. The reflux tank is considered in Section 5.1. The condenser model although different from the heat exchanger model presented in Section 3.4.2.2 will have the same failure modes, if it is assumed that the vapour phase is totally condensed and supercooled, and that changes in heat transfer rates affect only the outlet temperature of the condenser, and cannot cause vapour breakthrough. A more complex model is required to model the causes and effects of vapour breakthrough.

5.4.3) Top Event Modelling

The top event of interest is IMP B HI in Unit 9. IMP B HI represents a high concentration of component B, and in this case refers to the distillation failing to produce distillate of the desired quality. The top event model for this event is expressed in decision table form

V XB2out HI V G1in SOME V Q2out SOME T IMP B HI
V YB1in HI V Q2out REV T IMP B HI

This model has been used before, in Section 4.2.3.

5.4.4) Fault Tree Synthesis

There are no particular problems associated with fault tree synthesis in this example that have not already been considered in the examples of earlier chapters (Section 3.4.4 and Section 4.2.4). The fault tree for this system is displayed in Fig 5.7. Note that two options have been selected to simplify the fault tree. Firstly, reverse flow effects have been suppressed, both for simplicity and because the vessel models described earlier did not include reverse flow effects. Secondly, some events that were actually synthesised were not drawn. These events are variable deviations and intermediate events with only a single variable deviation or intermediate event cause. This prevents long chains of events with no additional causes appearing in the fault tree.

There are two instances where flow ratio occurs. The flow ratios are low reflux ratio at the top of the column, and high boilup ratio at the bottom of the column.

The internal consistency checks have not worked perfectly in this example. Firstly, the event PART-BLK Unit 8 (partial blockage of the reflux line) does not appear in the fault tree as a cause of low reflux flow. The reason for this is that low reflux flow is part of a ratio with high distillate takeoff rate. The synthesis package incorrectly identifies that PART-BLK Unit 8 will tend to decrease the distillate takeoff rate, and so removes the event as being inconsistent with high takeoff of distillate, the other cause in the flow ratio. PART-BLK Unit 8 would indeed cause a decrease in the distillate takeoff rate, were it not for the level control loop on the reflux tank. To cure this limitation, a more complex flow ratio treatment would have to be introduced, which could identify when an event like PART-BLK Unit 8 is not inconsistent with the flow ratio.

The other occasion where the internal consistency checks have not led to the correct removals relates to the LK-LP-EN (leak to low pressure environment) faults in the distillate product line. Although such events cause low reflux flow, the consistency checks remove these events as inconsistent with some flow of distillate. The reason for this is that, in pipe type models, LK-LP-EN is also specified as a potential cause of none and reverse outlet flow deviations, and so is identified as being inconsistent with some flow. This problem could be cured by having different leak faults to represent the different effects, in the same way that the blockage faults PART-BLK (partial blockage) and COMP-BLK (complete blockage) have different effects.

One interesting effect in the fault tree is that three inconsistent deviations of inlet flow appear. Q1 HI, Q1 LO and Q1 NONE are all identified as potential causes of the top event. There is no inconsistency involved in having inconsistent causes resulting in the same effect, if such causes occur under an OR gate. If they occur under an AND gate, then consistency checks could remove one or more of the causes. Q1 HI is identified as a possible cause of the top event because high flow of (vapour) feed into the column causes high pressure in the column. This will cause a low reflux ratio, thus decreasing the purity of the distillate. On the other hand, low flow of feed into the column causes low pressure in the column. At low pressure, the boilup ratio will be high, and the purity of the bottoms product will increase, but the bottoms product flow will decrease. If the net result is less of the less volatile component in the bottoms product, then there must be more in the distillate, thus causing the top event.

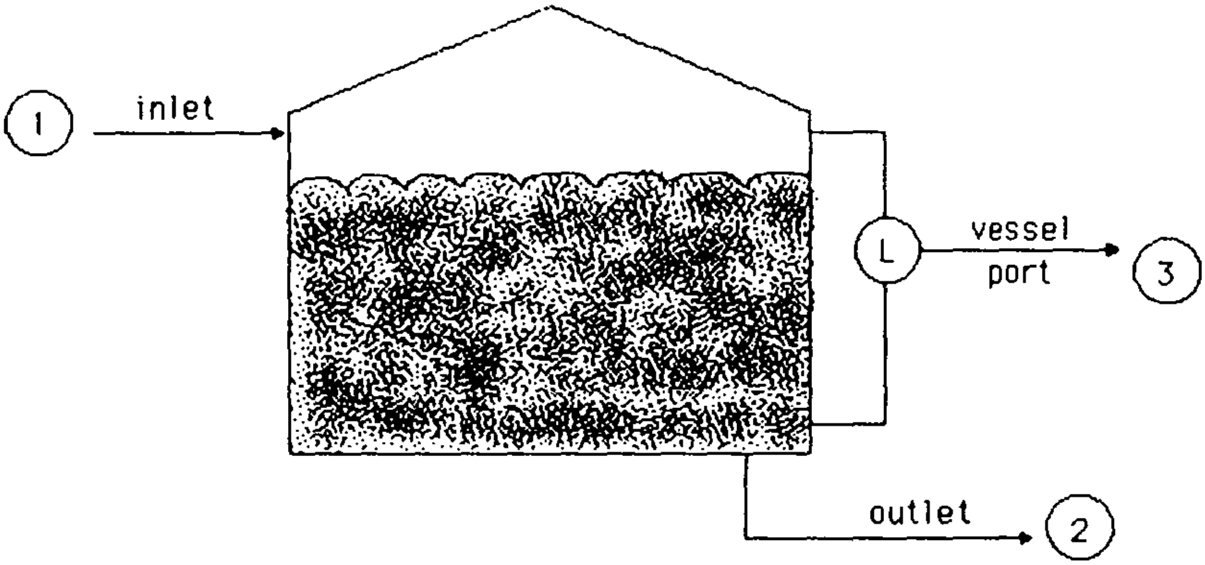


Figure 5.1 - a closed tank model

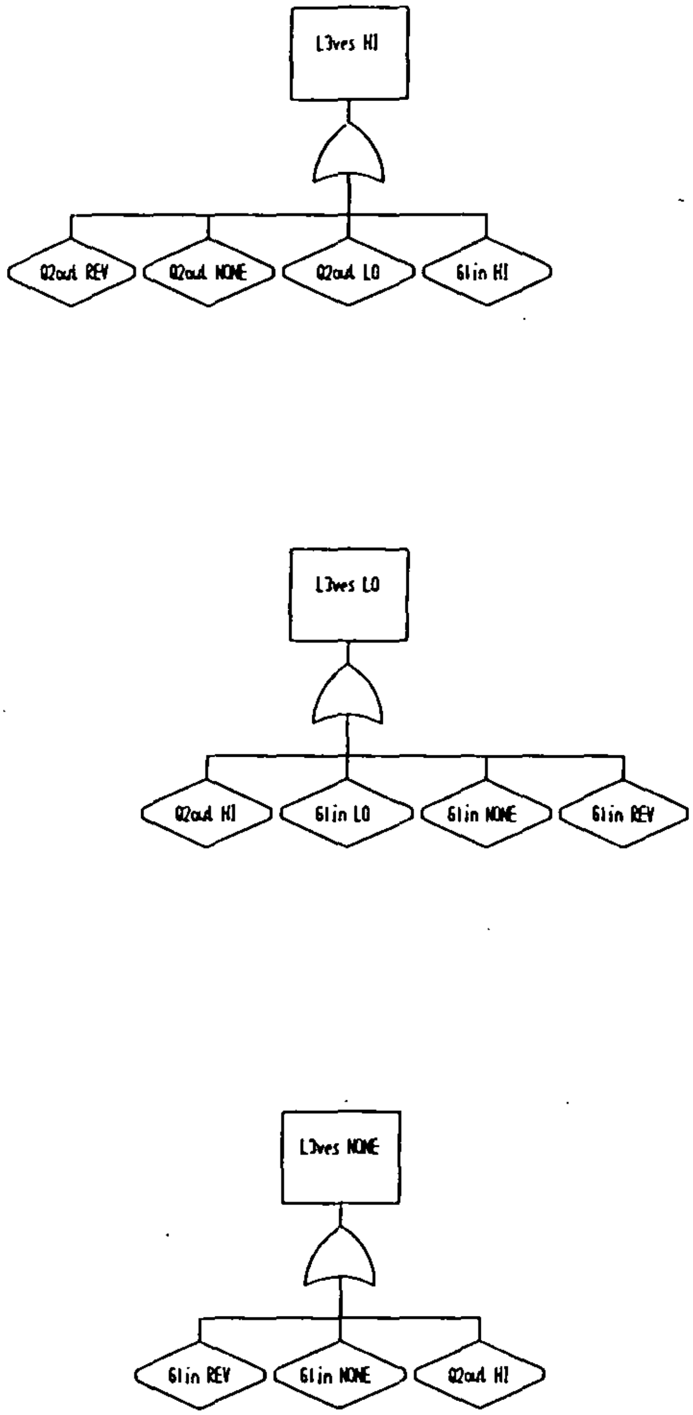


Figure 5.2 - the minitrees for the deviations of level in the closed tank model of Figure 5.1

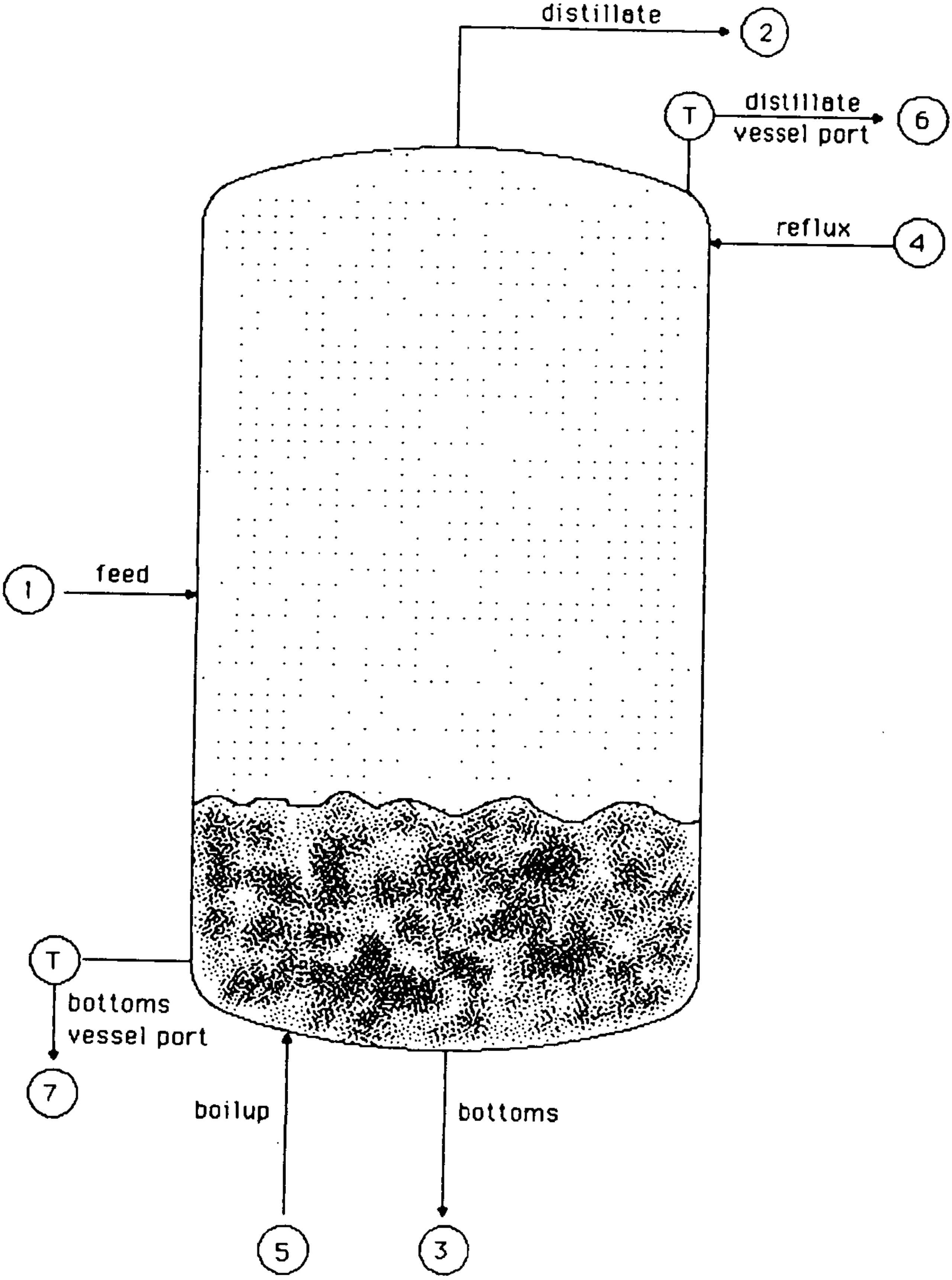


Figure 5.3 - a distillation column model

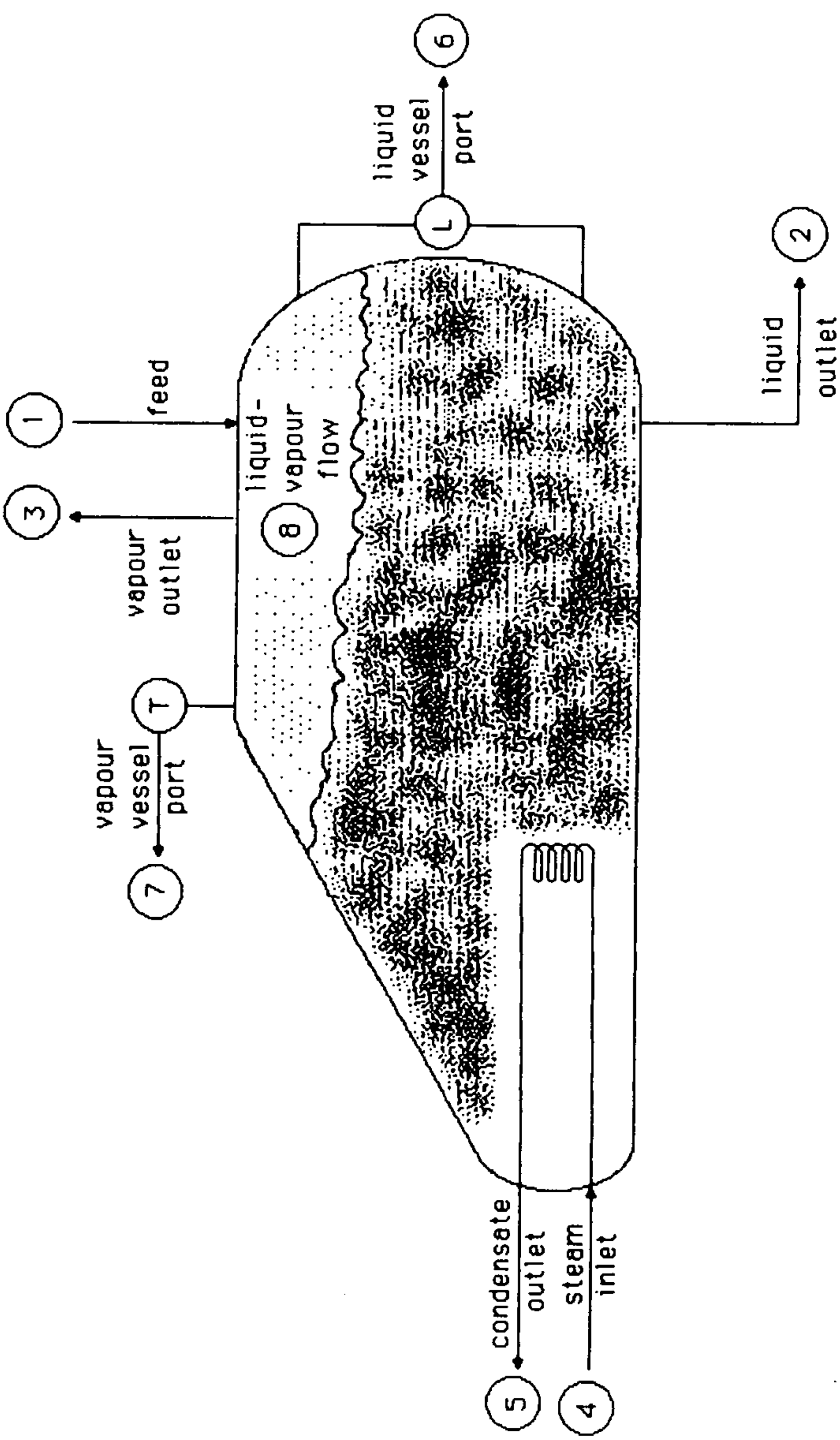


Figure 5.4 - a reboiler model

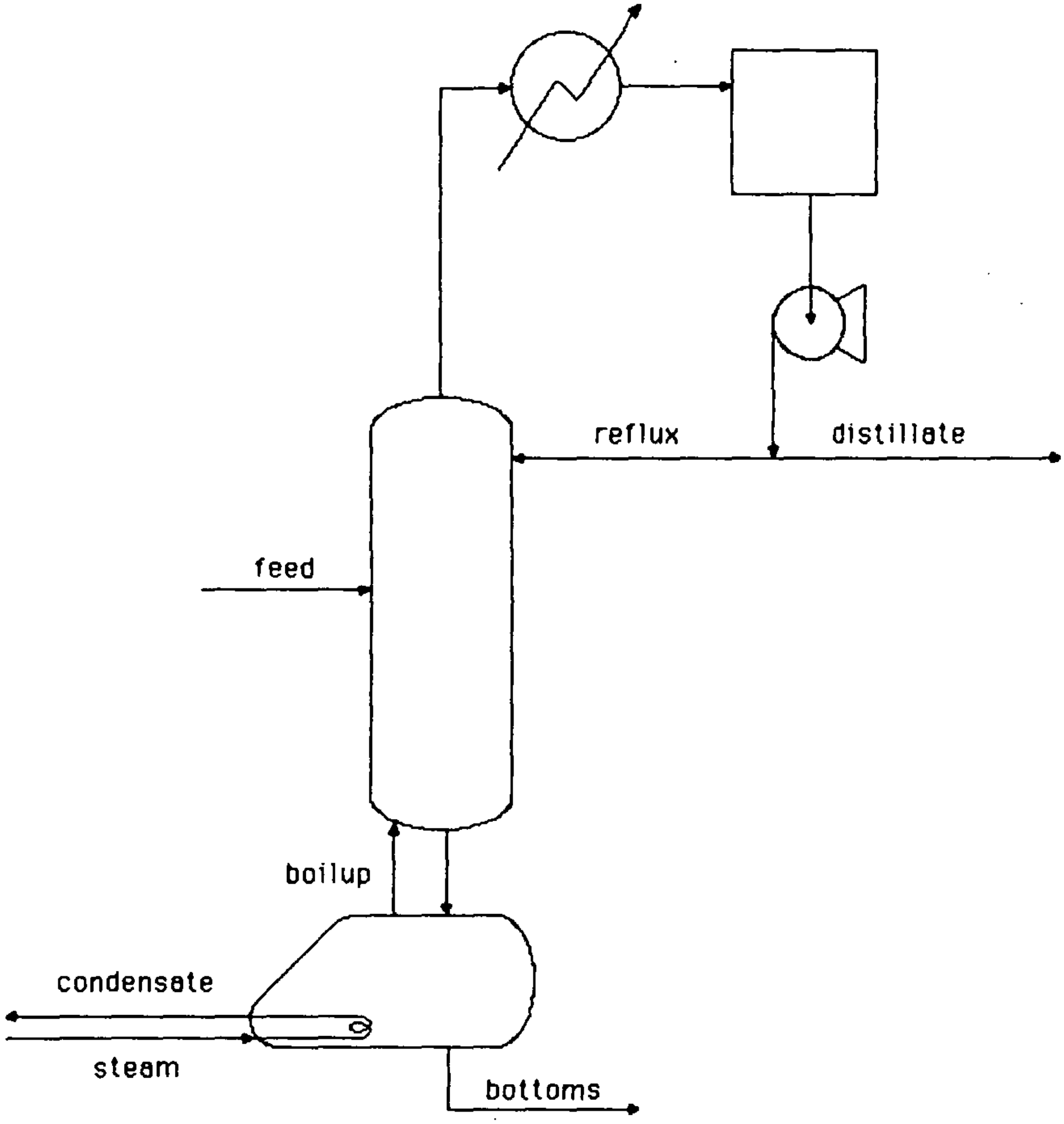


Figure 5.5 - a distillation column
system, based on
Shepherd et al [53]

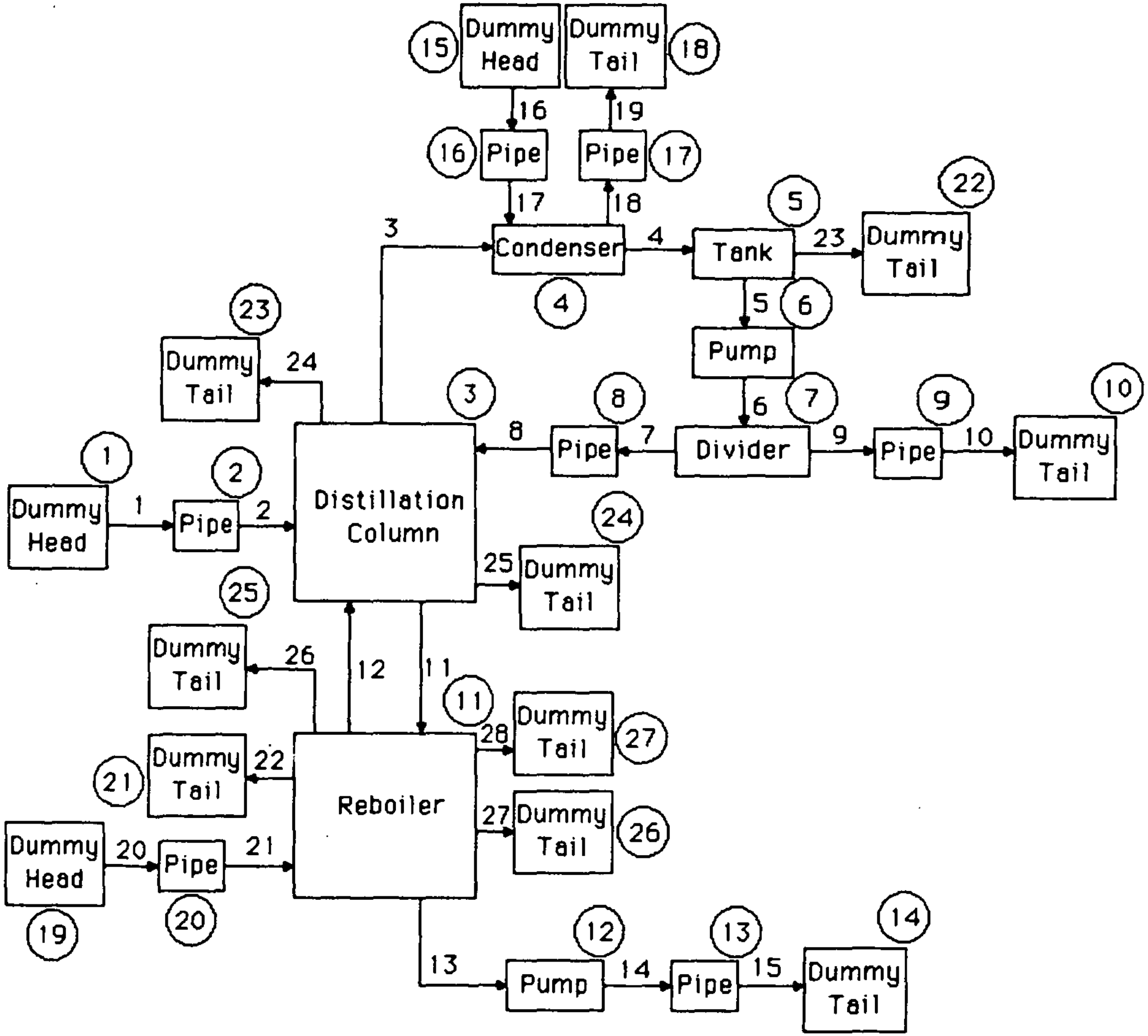


Figure 5.6 - the configuration diagram for the distillation column system shown in Figure 5.5

Distillation Column Example

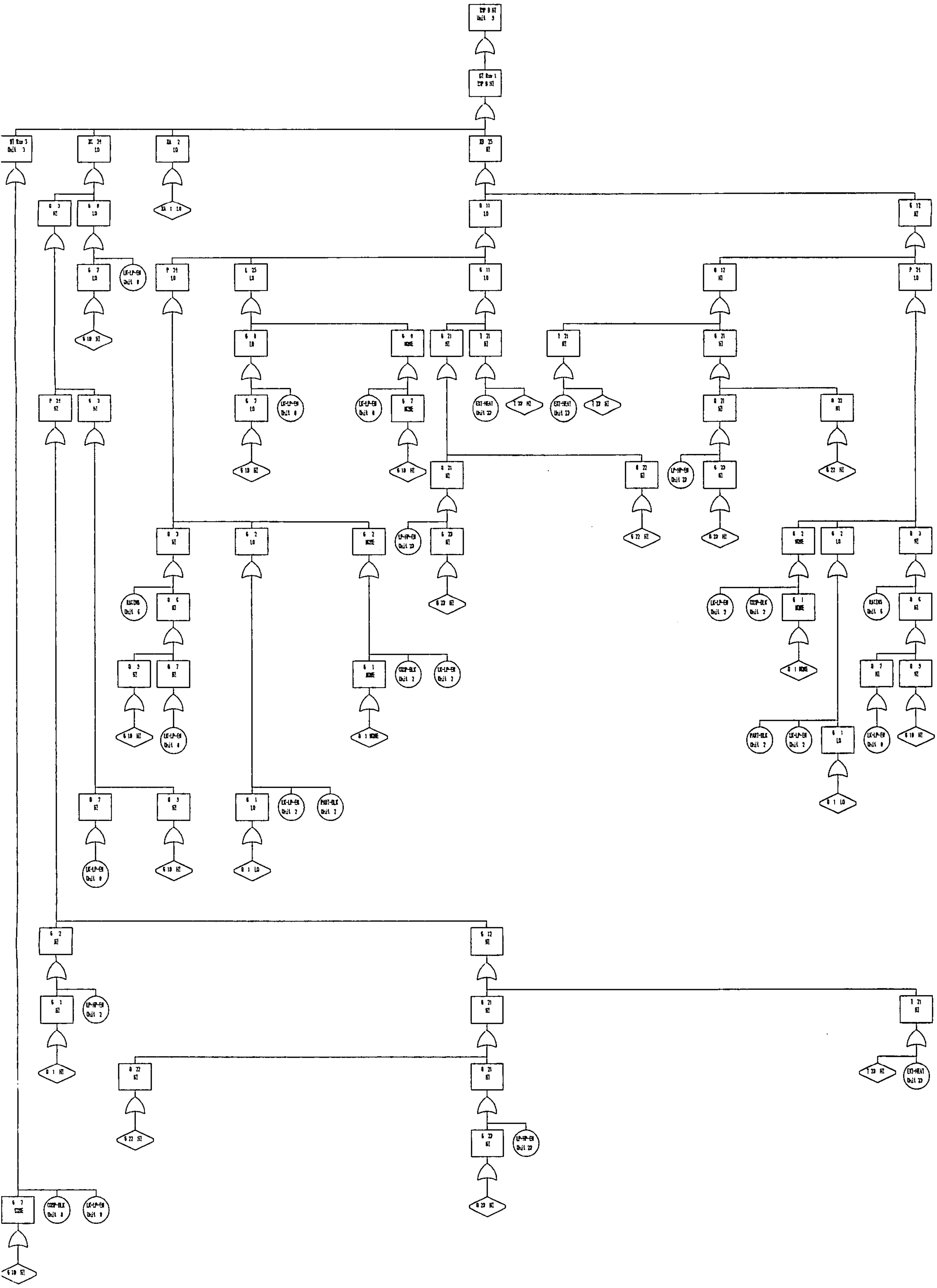


Figure 5.7 - complete fault tree for the system shown in Figure 5.6

6) Control Loops

The previous three chapters have illustrated the basic principles of the methodology, and have concentrated on modelling individual items of process equipment. Synthesising fault trees from groups of these models is a completely automatic process - given the models, the connections between the models and a top event, a fault tree can be generated.

Certain configurations of process items, however, cannot be adequately modelled using the principles of the three previous chapters. The remaining chapters of this thesis will look at such situations. This chapter examines control loops.

6.1) The Problem

At first sight it might appear that the elements of control loops can be modelled in exactly the same way as the process equipment items that were modelled in Section 4. This is true only to a certain extent, as Shafaghi [39-41] has found. Consider the simple feedback flow control loop shown in Fig 6.1. It involves three pipes, a flow sensor, a controller, a control valve and a setpoint unit. The pipe model has already been developed (Section 3.4.2.1), but the relevant minitrees are repeated in Fig 6.2 for convenience. Figs 6.3 to 6.6 give the minitrees for the flow sensor, control valve, controller and setpoint units respectively. Section 6.3.4 discusses the modelling of control loop components.

The control valve is assumed to be of the air-to-open type. The model for this is very similar to the pipe model. The only differences are that the flow through the valve is affected by the control loop signal, either because the signal changes incorrectly, or because the signal does not change to take into account the changing flow in the pipeline. This is the logic behind the decision tables involving the intermediate event CL-STK (control loop stuck).

The fault tree synthesised from these models for the top event Q6 LO (low flow out of the control valve) is given in Fig 6.7. A close study of this reveals a number of errors

a) neither Q1 HI nor LK-HP-EN Unit 2 can cause Q6 LO

b) the control loop should be able to correct PART-BLK Unit 5, PART-BLK Unit 6 and G6 LO (equivalent to restrictions further downstream) by opening the valve. These faults should therefore be ANDed with 'control loop stuck' (CL-STK)

c) LK-LP-EN Unit 4 should not be ANDed with CL-STK. The fault tree has the fault ANDed in one branch, but not ANDed in another branch.

The basic cause of these problems is that the combination of sensor, controller and control valve behave in a way over and above what can be deduced directly from the individual models. In fault tracing, it is important that no possible causes are overlooked. The technique employed is to find faults that may cause a particular event to occur, without examining the other effects of the fault. So, for example, PART-BLK Unit 6 will certainly cause Q6 LO. The structure of the models suggests that the fault should not be ANDed with control loop stuck. However, PART-BLK Unit 6 will also cause low flow through the sensor. The control loop should respond by opening the valve, until the low flow fault is cured. PART-BLK Unit 6 should therefore be ANDed with control loop stuck. This reasoning involves inductive logic, as well as the deductive logic normally used in fault tree synthesis.

A similar argument can be used to show that the control loop can detect, and presumably correct, PART-BLK Unit 5 and G6 LO.

Q1 HI and LK-HP-EN Unit 2 both have two effects on the plant. More obviously, they result in an increase in the outlet flow. But, by acting through the control loop, they also decrease the control valve aperture. Decreased control valve aperture is recognised as a potential cause of low flow out of the control valve, and so Q1 HI and LK-HP-EN Unit 2 are identified as a potential cause of low outlet flow. However, since the first effect of these events is the more dominant, these faults should not appear in the fault tree.

LK-LP-EN Unit 4 also has two effects. However, unlike the opposite effects noted above, these effects result in Q6 LO occurring in two different ways. Directly, they result in low flow out of the units. As such, the models indicate that LK-LP-EN Unit 4 should be ANDed with the control loop being stuck. However, the event also results in a high flow through the sensor. To correct this, the control loop will act to shut the valve, thus causing Q6 LO. The control loop cannot, therefore, compensate for LK-LP-EN Unit 4, and the event should not, therefore, be ANDed with control loop stuck.

6.2) The Approach of Others

The RIKKE code, developed by J.R. Taylor at Riso [42-44], includes in each model information on which other events may be able to compensate for a particular fault. For example, a restriction to flow downstream may be compensated by increasing the upstream pressure. Therefore, for an event to occur, FAULT AND NO COMPENSATION must exist. The causes of NO COMPENSATION are found by introducing a NOT gate, and tracing the causes of COMPENSATION. Modelling is complex, since the models must contain information not only on the cause effect relationships, but also on how each potential cause can be compensated.

The Lapp & Powers' methodology [23], based on a representation of a process plant using a directed graph, or digraph, imposes a structure on fault trees when a control loop is identified. Control loops in the plant can be recognised because they form a negative feedback loop in the information flow in the digraph. The models for control loops used have come in for some criticism [25-30], particularly over the use of XOR (exclusive OR) gates. Two way propagation of flow faults also presents problems to the Lapp & Powers methodology. Because of the way that AND gates are included in the fault tree, the Lapp & Powers technique will not arrive at the correct fault tree for the example above. The AND gate associated with control loop latent failures is included in the fault tree only when the loop in the digraph is encountered. This, in the example above, will be after the faults "restriction downstream" (G6 LO) and "partial blockage in pipe 5" (PART-BLK Unit 5) have been discovered. These faults will not, therefore be ANDed with control loop stuck. Furthermore, since pipe 3 is on the information flow loop in the digraph, a leak in this pipe will be ANDed with control loop stuck. Nevertheless, the approach of Lapp & Powers does solve some of the problems that control loops present.

The work of Shafaghi [39-41] has involved a close study of control loops in chemical plants. He proposes a fault tree synthesis method which uses the control loops, as opposed to process units, as the basic building blocks of process plant. Part of the work involves imposing a structure based on the general behaviour of control loops upon the fault tree. The prime disadvantage of the method is that the structure of each control loop in the plant must be worked out manually. Because the process units in each control loop domain have a profound influence on the final structure, constructing a library of standard control loops is not feasible.

6.3) A Solution

The method of treating control loops used in the synthesis package described in this thesis uses a control loop general structure similar to the structure proposed by Shafaghi [39-41]. This general structure is implicit in the fault tree synthesis package, and is expanded into specific faults when a particular control loop is studied.

The decision on what faults should appear in which branch of the general structure is calculated automatically by the fault tree synthesis package. However, some information on the control loop and its intended performance must be supplied as part of the plant configuration information. The information required is the information about a control loop that cannot be deduced from the component models alone, such as what is being regulated, and how control is maintained. This information defines a control loop model, which is used in conjunction with the component models to synthesise fault trees for systems involving control loops.

6.3.1 Deviation of a Regulated Variable

Fig 6.8 shows the general structure for the deviation of a variable regulated by a control loop. There is a different general structure for deviations of a manipulated variable. This will be examined later.

Fig 6.8 involves five branches

a) control loop spontaneous failures
b) misleading or undetectable faults
c) detectable and correctable faults
d) control loop latent failures
e) overloading faults

The fault tree for the flow control loop considered above with this general structure imposed on it is given in Fig 6.9. Intermediate events are used in most cases to represent the branches in the control loop general model and are, with the letters corresponding to the descriptions of each branch given above

a) CL-F-LA or CL-F-HA. CL-F-LA is used for spontaneous failures leading to a low control valve aperture, CL-F-HA is used if a high control valve aperture results

b) C(DUMMY)

c) an intermediate event is not used; the variable deviation at the sensor that causes the top event is used, and in this example it is Q2 LO

d) CL-STK

e) F(DUMMY) (does not appear in Fig 6.9)

In addition to these events, there are additional events in the fault tree that occur because of the presence of the control loop. These events are designed to connect the events described above using the correct logic, and are A(DUMMY), D(DUMMY) and E(DUMMY).

Control loop spontaneous failures are failures in the control loop that cause a deviation in the variable regulated by the loop. In the fault tree for the flow control loop (Fig 6.9) this includes events such as CNT-F-LO Unit 8 (the controller outputs too low a signal) and CV-F-LA Unit 5 (control valve aperture fails too low).

The only misleading fault (there are no undetectable faults in this example) is LK-LP-EN Unit 4 (leak to a low pressure environment in the pipe.

There are no overloading faults in this example. Such faults are assumed to be either NONE or REV deviations of the manipulated stream. The control loop cannot induce a normal flow by altering the valve position if either of these conditions exists. On the other hand, it is assumed that all LO and HI deviations are correctable if the control loop can detect such faults and is working properly. Most of the process item (as oppose to control loop component) faults come into this detectable and correctable category. Examples are G6 LO (restrction to flow further downstream) and PART-BLK Unit 2 (partial blockage in the pipe).

Control loop latent failures cause the control valve to be invariant. Examples are CNT-STK Unit 8 (controller does not respond to a change in input signal) and SEN-STK Unit 3 (sensor does not respond to a change in flow).

6.3.1.1) Feedback Loop Example

This example is based on the Lapp & Powers nitric acid cooler [23], but the present study omits the trip system which stops the flow of nitric acid should the cooling water pump stop. The configuration diagram is shown in Fig 6.10.

The system involves only a simple feedback temperature control loop. This loop regulates the outlet temperature of the heat exchanger (connections 3, 4 and 5) by manipulating the flow of cooling water (connections 6 to 10 inclusive).

The fault tree for high outlet temperature is shown in Fig 6.11. Note that there are no misleading/undetectable faults, but that there is an overloading fault associated with no flow of cooling water.

This fault tree can be compared with the example presented in Section 3.4. The examples are similar, the only difference being the presence of the temperature control loop in the example presented in the current section. Note, however, that the heat exchanger model used here is slightly more detailed than that used in Section 3.4 - the effects of complete loss of coolant flow have been included.

6.3.1.2) Feedforward Loop Example

The same general structure used above to synthesise fault trees for systems involving feedback control loops is also suitable for studying systems involving feedforward control loops. Consider the mixing system shown in Fig 6.12. This involves mixing component 'A' (streams 1-4) and component 'B' (streams 5-8) to produce a single stream of regulated composition (streams 9-10). The fault tree for this system is complicated by the presence of flow ratio. Flow ratio requires a special treatment and this is discussed in Section 4.1.5. The fault tree for the plant, with the special treatment for flow ratio used, is given in Fig 6.13.

Note the large number of faults that are undetectable by the control loop. This is a characteristic of feedforward control loops.

6.3.2) Deviation of a Manipulated Variable

The examples studied so far have involved deviations of regulated variables, that is, variables that the control loop attempts to maintain at a preset value. This regulation is frequently achieved by the manipulation of the flow of some other stream. In the temperature control loop (Section 6.3.1.1, Fig 6.10), the temperature of streams 3, 4 and 5 is regulated by manipulating the flow of streams 6 to 10. The flow in the manipulated stream is therefore dependent on the control loop, but in a different way to the regulated variable. Another general structure is therefore required. In fact, the general structure is different for feedback and feedforward loops. The two structures are given in Fig 6.14 (feedback loop) and Fig 6.15 (feedforward loop).

These involve the following branches

a) faults which cause the control loop to induce the deviation as part of the normal operation of the control loop. This is represented in fault trees by the intermediate event F(DUMMY)

b) faults which cause the deviation directly. These are the causes that would exist if the control loop were absent. This is represented in fault trees by the intermediate event A(DUMMY)

c) control loop latent failures, represented in fault trees by the intermediate event CL-STK

d) control loop spontaneous failures, represented in fault trees by the intermediate events CL-F-HA or CL-F-LA, depending on the control valve aperture deviation (high or low)

In addition to the intermediate events described above, the intermediate events C(DUMMY), D(DUMMY) and E(DUMMY) are used to connect the various branches using the correct logic.

The only difference between the two structures is that the feedback loop can counteract deviations in the manipulated stream, whilst the feedforward loop cannot. This can be seen by examining the fault trees for deviations in the regulated variable for each loop type (Figs 6.11 and 6.13).

6.3.2.1) Feedback Loop Example

Fig 6.16 shows the configuration diagram for a simple level control loop on a tank, operating by manipulating the tank outlet flow. If the top event were a deviation in tank level, then the general structure used would be that for a regulated variable deviation. However, when the top event is a deviation in outlet flow, then the manipulated variable deviation structure must be used. The fault tree for this top event is shown in Fig 6.17.

Note that low inlet flow (Q1 LO) is in the branch that represents normal action of the control loop.

Although Q1 LO will cause a deviation in the outlet flow if the control loop is not present, the fault does not appear in the normal causes branch. This is because any events that appear in both branches have a more direct effect through the operation of the control loop. Such events, therefore, only appear in the control loop action branch. In fact, leaving the event in the normal causes would not affect the logic in the fault tree, since leaving it in the fault tree would add a redundant two event cutset of the form Q1 LO AND Control Loop Stuck. This cutset is redundant because Q1 LO appears as a one event cutset in the control loop action branch.

6.3.2.2) Feedforward Loop Example

This example uses the feedforward system studied in the section on regulated variable deviations (Section 6.3.1.2, Fig 6.12). In this case, the top event is HI FLOW Unit 4. The fault tree for this top event is displayed in Fig 6.18. Note that the normal causes branch is not ANDed with control loop latent failure. This is because feedforward loops, by definition, cannot detect changes in the flow of the manipulated stream.

6.3.3) Complex Control Loops

Complex control loops are control loops with more than one sensor and, occasionally, more than one controller. A cascade control loop is a typical example (see Fig 6.19). A flow control loop receives its setpoint from a slower acting temperature control loop.

The treatment of complex control loops always involves defining each individual loop of the complex loop individually to the package. So, the cascade control loop of Fig 6.19 is defined as two control loops, one a temperature control loop and the other a flow control loop. Both loops are of the feedback type.

When defining complex control loops to the package, part of the information required is which control loop is the master loop, and which is the slave loop. In the cascade system considered above, the temperature loop is the master loop, and the flow loop is the slave loop. All complex loops can be decomposed on this basis. This information is required so that the fault tree indicates that the master loop can correct for faults in the slave loop, but that the slave loop cannot correct for faults in the master loop. In common with the other information required to define a control loop (eg what is regulated), this information is not explicitly contained in a configuration diagram - it must be supplied by the analyst.

6.3.3.1) Complex Control Loop - Regulated Variable

This example is one used to test the RIKKE computer code [43]. It consists of a complex and unusual control system to regulate the composition of a stream produced by mixing two streams of different compositions. One control loop is a simple feedback composition control loop, and the other is a feedforward control loop similar to the loop examined in Section 6.3.2. These loops share a common controller and control valve.

The configuration diagram is displayed in Fig 6.20. The feedback composition loop is the master loop, and the feedforward loop is the slave loop. A new unit has been introduced for this example, an Instrument Air utility, connected to the common controller. The controller is modelled such that complete loss of the utility causes no output signal out of the controller. Since the control valve in this example is an air-to-open valve, utility loss is a cause of the valve failing shut.

The fault tree for the top event HI COMP Unit 12 (high composition of the manipulated stream component out of the system) is shown in Fig 6.21. The reason for defining the master loop first can be found by studying this tree. The composition loop can correct faults in the flow ratio loop, but the reverse is not true. To include this information in the fault tree, it is essential that the master composition loop be defined first.

Many of the events that occur in this fault tree are common to several of the control loop model branches. For example, CV-STK Unit 3 (control valve stuck) appears in the latent failures of both general models, since the control valve belongs to both control loops. There are therefore several redundant cutsets in this fault tree, but the minimum cutsets are correct. Generally, complex control loops will have redundant cutsets, since, by the definition of a complex control loop, some components are members of more than one control loop.

6.3.4) Modelling Control System Components

Because of the special treatment of control loops in the methodology, there are a number of conventions that must be used when creating models for control loop components. These are designed to ensure that the fault tree synthesis program can identify the various fault types in the control loop general models.

6.3.4.1) Sensors

There are two basic types of sensor. The first is a vessel mounted sensor. This model has two ports. One port is a vessel port, and will, in the configuration diagram be linked to some vessel. The second port is the signal output port. Fig 6.22 is the representation of a vessel mounted temperature sensor. The model for this unit is straightforward

```
S2sig=f(T1ves)

F SEN-F-HI:S2sig HI
F SEN-F-LO:S2sig LO
F SEN-STK:S2sig NCHA
S NORMAL:S2sig SOME
```

There will normally be some output signal from the sensor, and so S2sig SOME is expressed as a normal state.

The second type of sensor is a pipe-type unit, with a third port for the signal output. With only one exception, deviations in the signal output are related to variable deviations in the inlet stream. The only exception is Ssig SOME for a flow sensor (see below).

This corresponds to the modelling principles introduced in Section 3.2.3.1 and justified in Section 3.4.2.2.

Fig 6.23 is the representation of a pipe-mounted temperature sensor. The model for this unit is similar to the pipe model. For simplicity, this model does not include faults such as leak and blockage faults. However, as described in Section 3.1.1, linking the unit to pipe units will include the effect of such faults in fault trees. The model for this unit is

S3sig=f(T1in)

F SEN-F-HI:S3sig HI
F SEN-F-LO:S3sig LO
F SEN-STK:S3sig NCHA
S NORMAL:S3sig SOME

Propagation equations for variables at ports 1 and 2 are identical to those in the pipe model; there are no event statements or decision tables for variables at ports 1 and 2. Note that the output signal is expressed in terms of the inlet temperature (T1in), and not in terms of the outlet temperature (T2out).

Fig 6.24 is the representation of a flow sensor. The model for this unit is

S3sig=f(G1in)

F SEN-F-HI:S3sig HI,S3sig SOME
F SEN-F-LO:S3sig LO
F SEN-STK:S3sig NCHA

V G1in SOME V Q2out SOME T S3sig SOME

As with the model for the pipe-mounted temperature sensor, the model for the flow sensor does not consider the effects of leaks and blockages. The expressions for deviations of variables at ports 1 and 2 are therefore identical to the expressions contained in the pipe-mounted temperature sensor.

Note, firstly that the signal is expressed in terms of G1in, and not Q1in. As pointed out in Section 3.2.4, these two terms are identical. G1in is used to conform to the convention that output flow variables should (except in propagation equations for flow) appear in propagation equations (see Section 3.4.2.2). The second point of interest is the decision table expressing S3sig SOME in terms of flow. In contrast to the previous sensor models examined, S3sig SOME is not a normal state. It will occur only if there is some flow past the sensor (or the sensor has failed - SEN-F-HI is used in the model to represent this failure). However, as noted in Section 4.1.1, the SOME flow deviation must involve an AND gate. It is inconvenient to include this gate in the minitrees for flow, and it was noted that the gate need only appear explicitly at the start of new fault tree branches.

6.3.4.2) Controllers

Modelling controller units is, in most cases, straightforward. The only problems that can arise occur in complex controllers, that is controllers with more than one signal input. Complex controllers will be dealt with later in this section.

Controllers are of two basic types. The output signal may be directly proportional to the input signal (direct controller), or inversely proportional to the input signal (inverse controller). The type of controller selected for a particular loop must be based on the performance of the loop and the type of control valve (air-to-open or air-to-close). If the wrong type is inadvertently selected, then the control loop will be unable to correct for faults that it should detect, and will be able to correct for faults that should mislead it. This will be apparent in the fault tree - events in the detectable and correctable branch, and in the misleading/undetectable branch will be interchanged.

Fig 6.25 is the representation of a simple inverse controller, that is a controller where the signal output is inversely proportional to the sole signal input. This representation also includes a utility input to the controller. In pneumatic controllers, the utility will be instrument air; in electrical controllers it will be electrical power.

In simple controllers, in the propagation equation for the output signal, the setpoint and the input signal must have opposite signs. The reason for this is that a controller acts to counteract a deviation in input signal. Therefore, to cause that same deviation, in other words to change the setpoint, the controller must be modelled as described. The model for a simple inverse controller is :-

```
S2sig=f(-S1sig,W3in)

F CNT-F-HI:S2sig HI
F CNT-F-LO:S2sig LO
F CNT-STK:S2sig NCHA
V S4utl NONE:S2sig NONE
```

S NORMAL:S2sig SOME

The only utility failure that is considered is complete utility loss, as modelled by S4utl NONE. The effect of this is to cause no output signal to be output by the controller. Partial utility loss has not been modelled.

As with sensors other than the flow sensor, the normal state of controllers is that they output some signal. S2sig is therefore a NORMAL state.

Complex controllers are modelled as combinations of simple controllers. For example a controller with two inputs (ports 1 and 2) in which the output (port 3) is directly proportional to one input and inversely proportional to the other input will have a propagation equation of the form

S3sig=f(S1sig,-S2sig,±W4in)

The output signal is directly proportional to the signal from port 1, but inversely proportional to the signal from port 2. The effect of setpoint (port 4) changes depends on which of the inputs comes from the master loop. Since the setpoint of the control loop is based on the sensed variable of the master loop, the effect of setpoint on the output signal obviously dependent on which input comes from the master loop. If port 1 is from the master loop, then the output signal is inversely proportional to the setpoint. If port 2 is from the master loop, then the output signal is directly proportional to the setpoint. To avoid confusion, a convention will be introduced. Port 1 will always come from the master control loop. The above controller will be called a direct/inverse controller.

An inverse/direct controller would have the master term inversely proportional to the signal received from port 1. The model for the direct/inverse controller is

```
S3sig=f(S1sig,-S2sig,-W4in)

F CNT-F-HI:S3sig HI
F CNT-F-LO:S3sig LO
F CNT-STK:S3sig NCHA
V S5utl NONE:S3sig NONE
S NORMAL:S3sig SOME
```

The event statements in this model correspond to those in the model for a simple inverse controller.

6.3.4.3) Control Valves

Modelling of control valves involves the following conventions

a) the intermediate events CL-F-HA (control loop fails resulting in the control valve having too large an aperture), CL-F-LA (control loop fails resulting in low aperture) and CL-STK (control loop stuck) must be used to model these faults

b) high and low deviations in the inlet and outlet flow must be related to the appropriate faults under a)

c) there is no special treatment needed for no, some and reverse flow deviations

Fig 6.26 is the representation of a control valve .

Since the control valve is a unit with flow, much of the modelling relates to this. This aspect closely resembles the pipe model. The statements below are the additional information needed for the control valve aspect. The valve is assumed to be of the air-to-open type.

```
G1in=f(Q1in,Q2out,S3sig)
Q2out=f(G1in,G2out,S3sig)

V S3sig HI:CL-F-HA
V S3sig LO:CL-F-LA
V S3sig NCHA:CL-STK
F CV-F-HA:CL-F-HA
F CV-F-LA:CL-F-LA
F SIG-PB:CL-F-LA
F CV-STK:CL-STK
I CL-F-HA:G1in HI,Q2out HI
I CL-F-LA:G1in LO,Q2out LO

V G1in HI I CL-STK T Q2out HI
V G1in LO I CL-STK T Q2out LO
V Q2out HI I CL-STK T G1in HI
V Q2out LO I CL-STK T G1in LO
```

These statements are fairly straightforward and require little explanation. It should be noted, however, that the minitrees for control valve units that appear in fault trees differ from the minitrees in this model. This is because the minitrees that involve the special control loop intermediate events (CL-F-LA etc) are used by the synthesis package to generate the correct control loop structure, and do not appear in the fault tree like normal intermediate events.

To cover the additional flow deviations (none, some and reverse flow), the following information is appropriate

```
V S3sig NONE:CL-F-NA
F CV-F-SH:CL-F-NA
F SIG-CB:CL-F-NA
I CL-F-NA:G1in NONE,Q2out NONE

V G1in SOME V S3sig SOME T Q2out SOME
V Q2out SOME V S3sig SOME T G1in SOME
V G1in REV V S3sig SOME T Q2out REV
V Q2out REV V S3sig SOME T G1in REV
```

The intermediate event CL-F-NA, representing the control loop failing so that the valve is shut, is not necessary to the model in the same way that the other CL- faults are. However, it is convenient to group such faults under a single event.

Pressure and relief deviations, if so desired, can be similarly modelled.

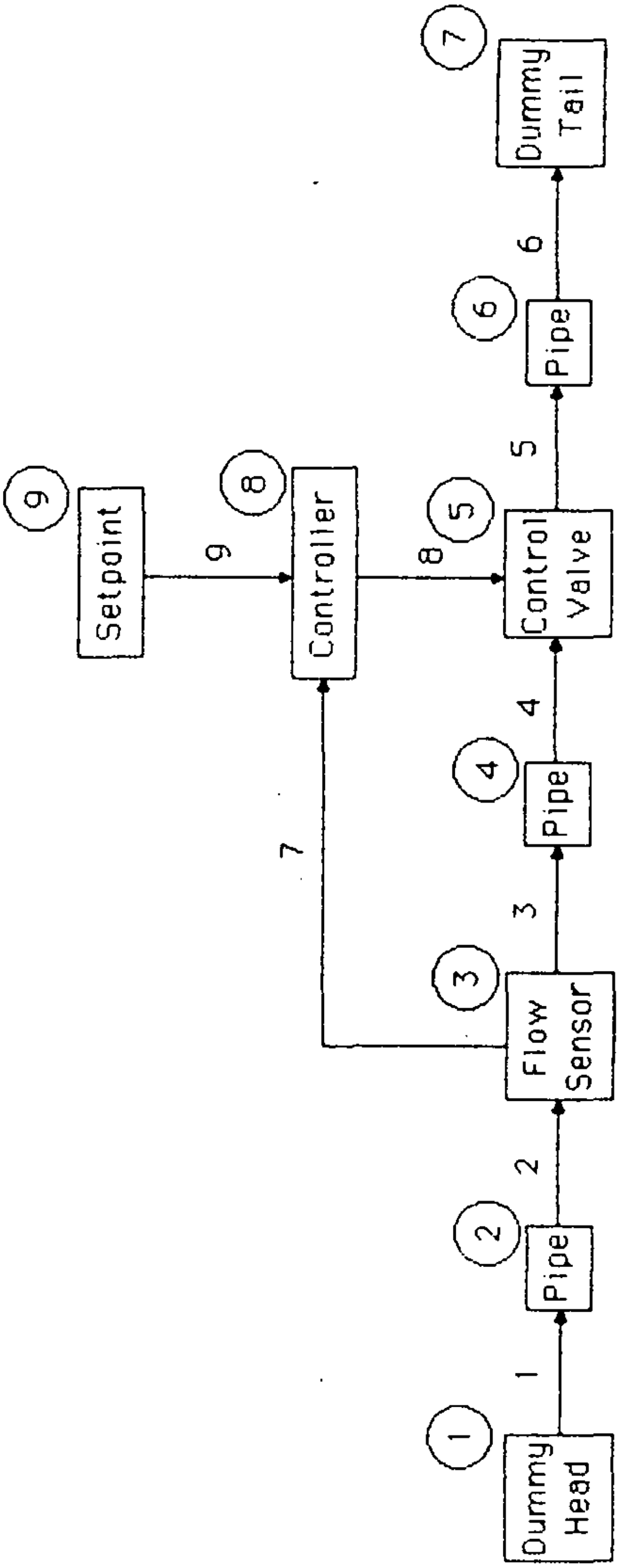


Figure 6.1 - configuration diagram for a simple flow control loop

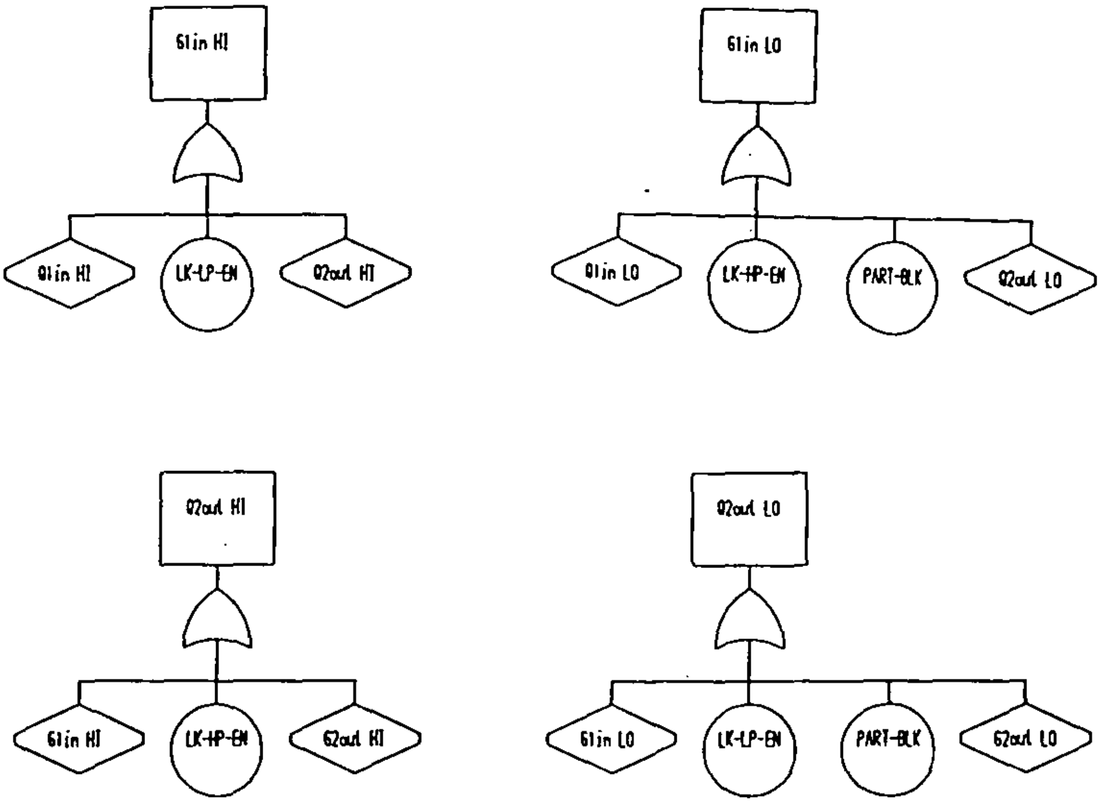


Figure 6.2 - the minitrees for the deviations of flow in a pipe used in the study of Figure 6.1

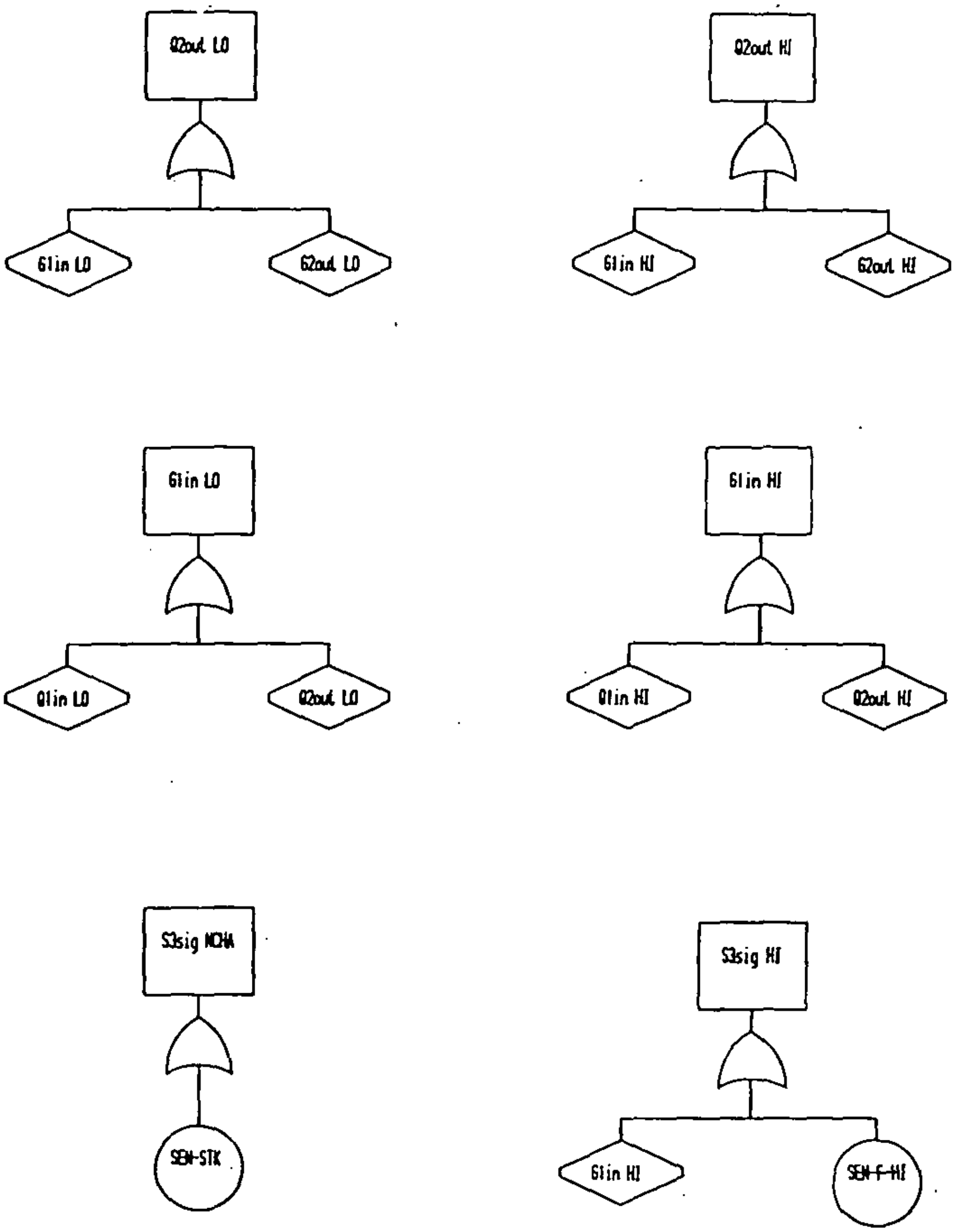


Figure 6.3 - the minitrees for a flow sensor used in the study of Figure 6.1

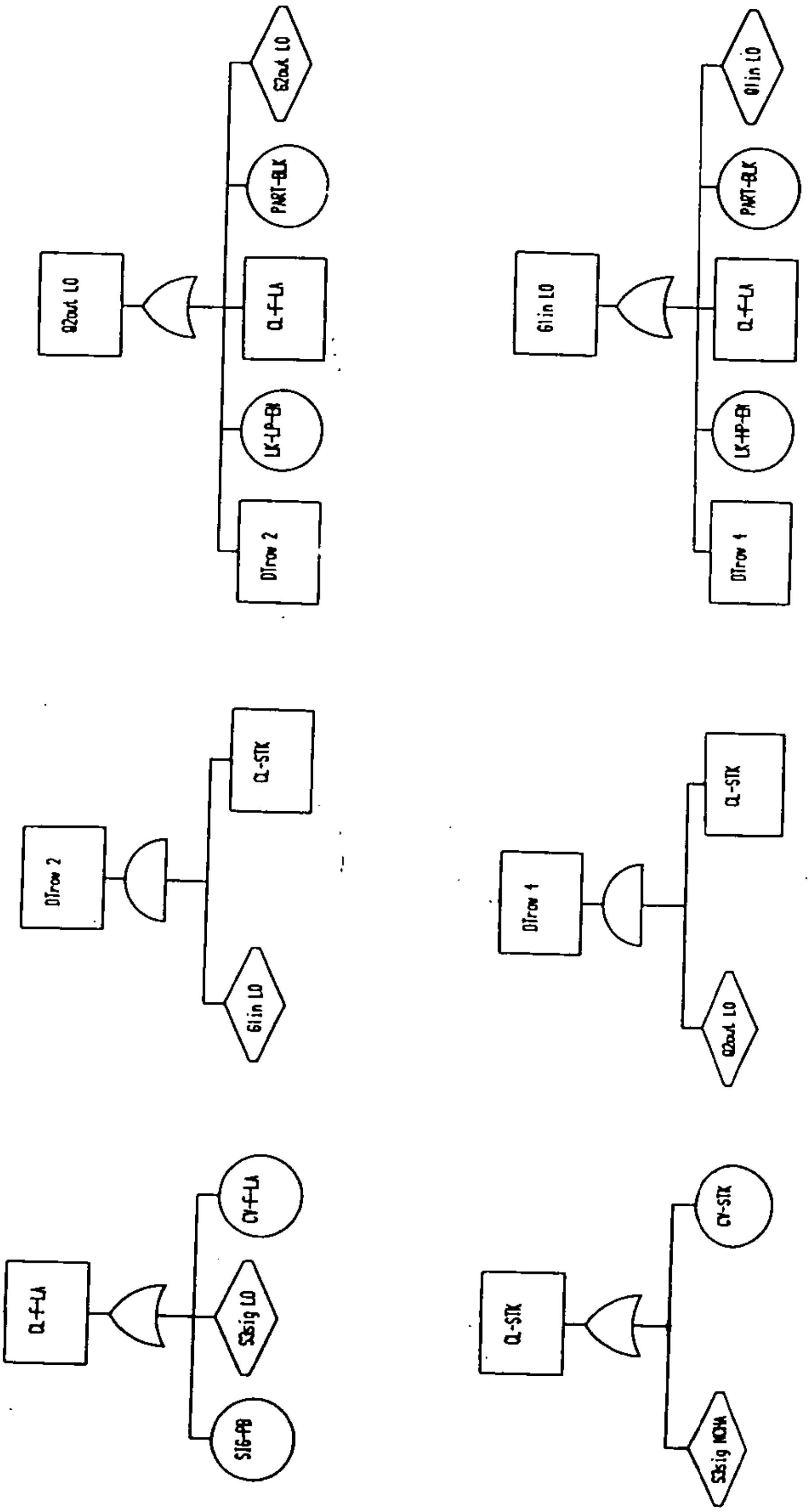


Figure 6.4 - the minitrees for a control valve used in the study of Figure 6.1

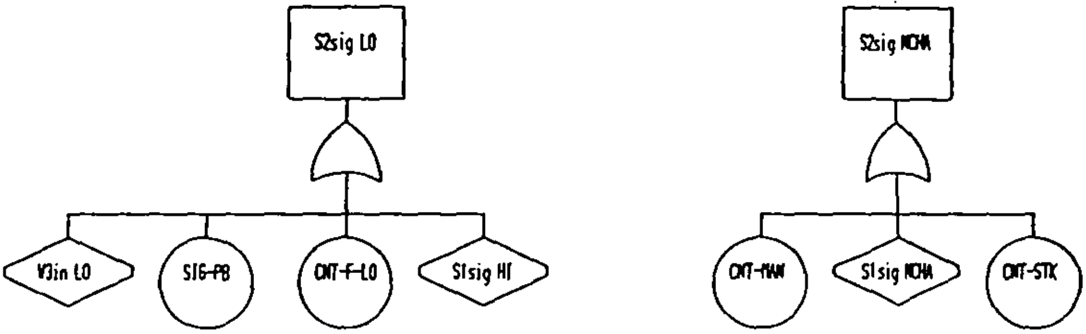


Figure 6.5 - the minitrees for a controller used in the study of Figure 6.1

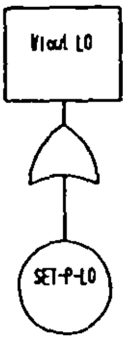


Figure 6.6 - the minitree for a setpoint unit used in the study of Figure 6.1

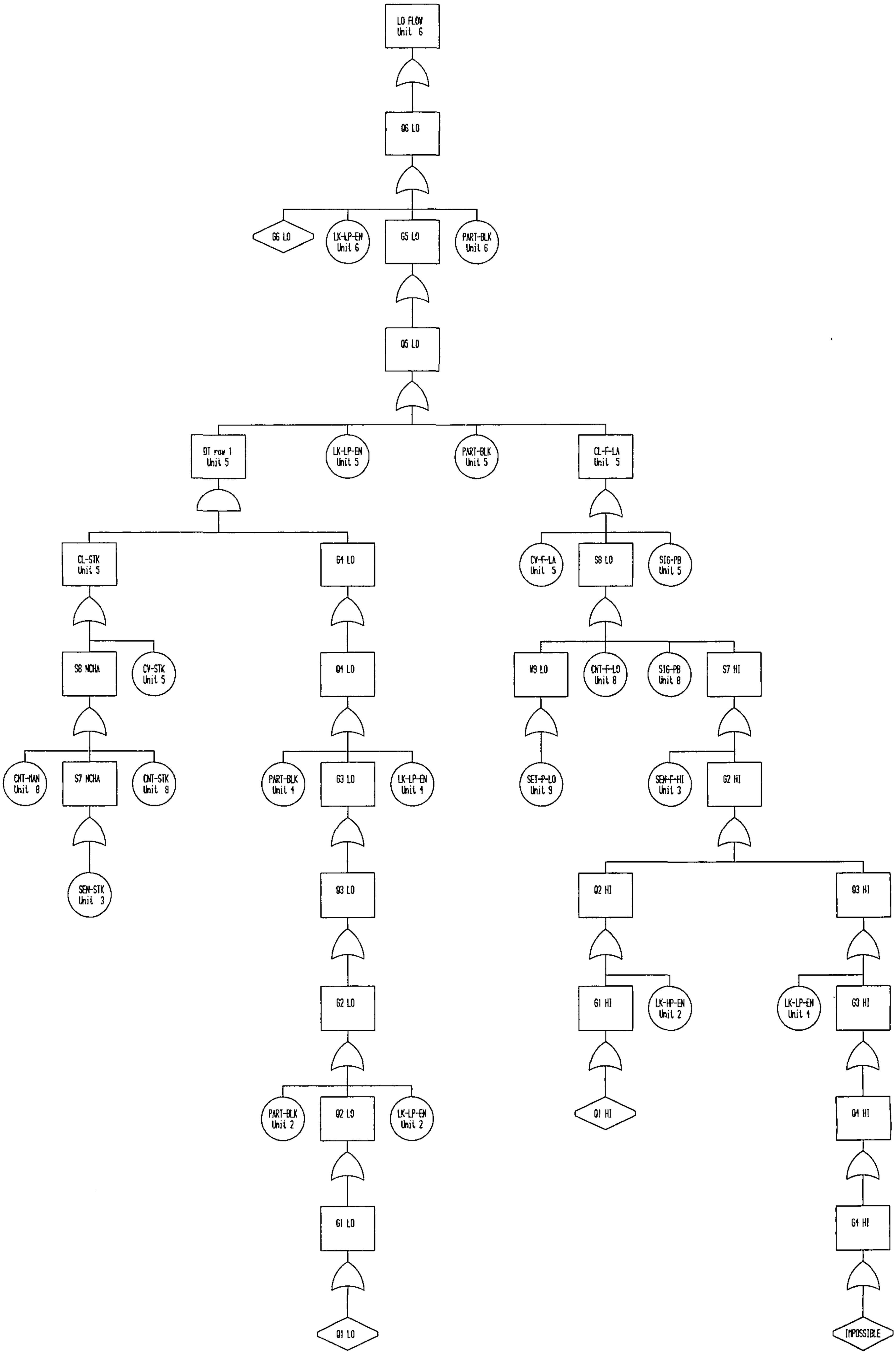


Figure 6.7 - complete fault tree for the system shown in Figure 6.1, with no special control loop treatment

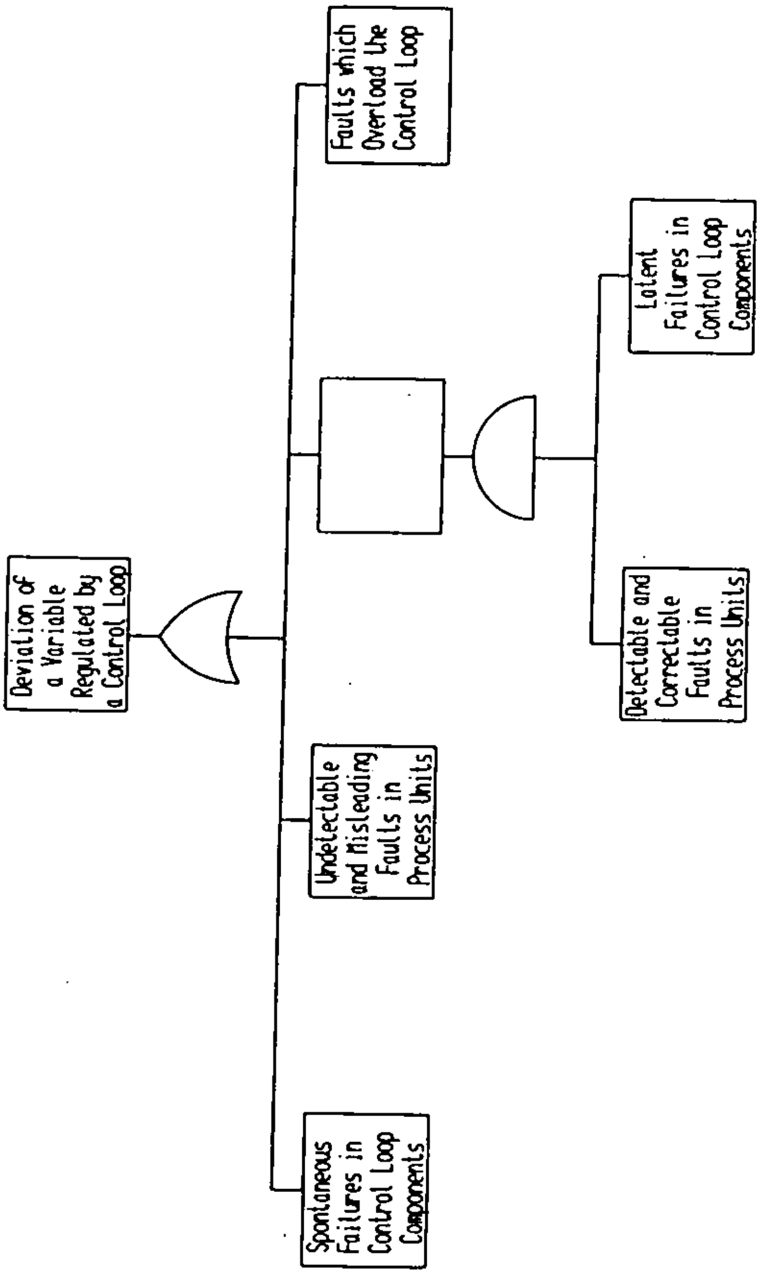

Deviation of a Variable Regulated by a Control Loop
Spontaneous Failures in Control Loop Components
Undetectable and Misleading Faults in Process Units
Faults which Overload the Control Loop
Detectable and Correctable Faults in Process Units
Latent Failures in Control Loop Components

Figure 6.8 - the general model for a regulated variable deviation

Flow Control Loop Example

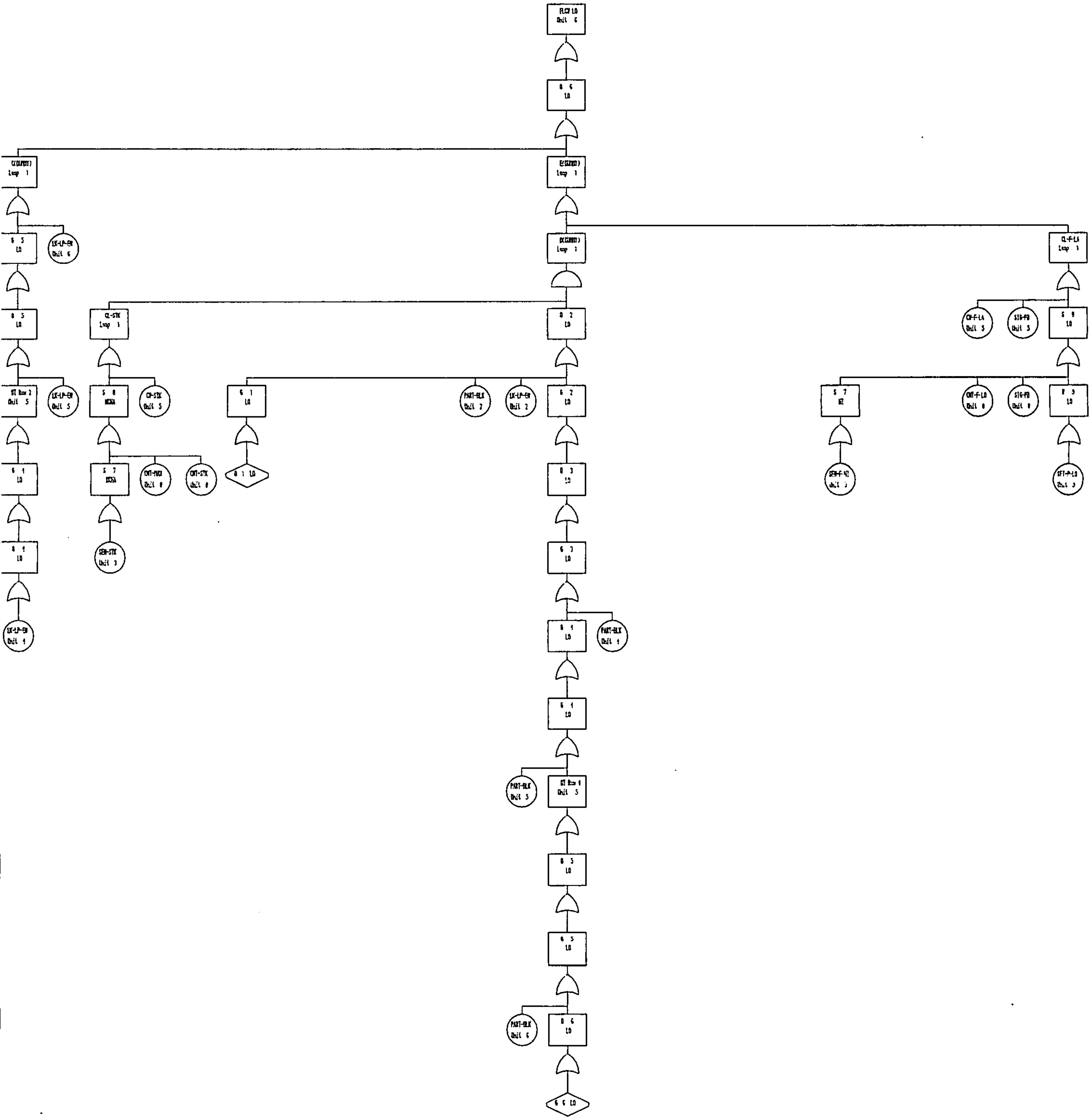


Figure 6.9 - complete fault tree for the
system shown in Figure 6.1,
with special control loop treatment

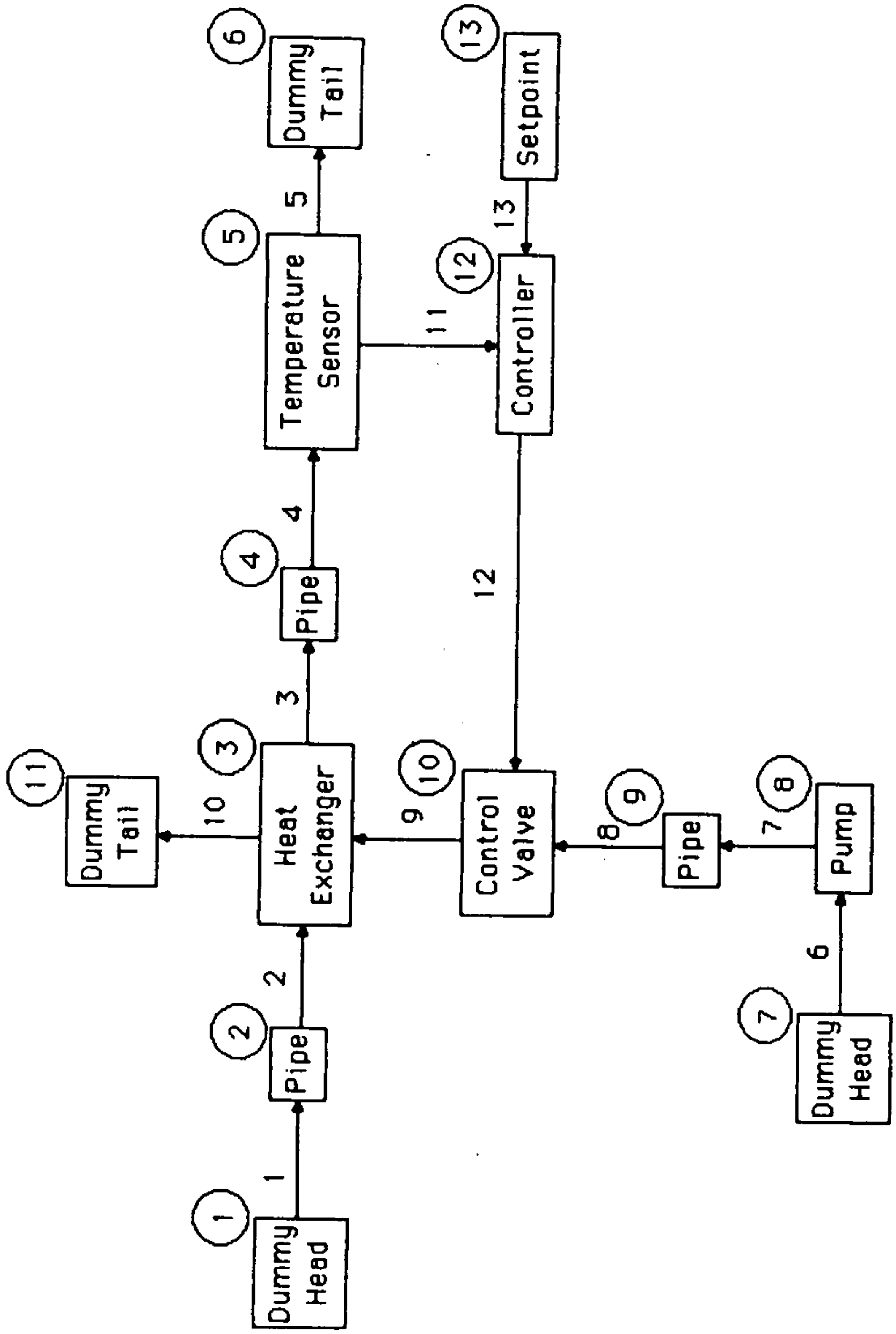


Figure 6.10 - configuration diagram for an example based on the Lapp & Powers Nitric Acid Cooler [23]

Temperature Control Loop Example

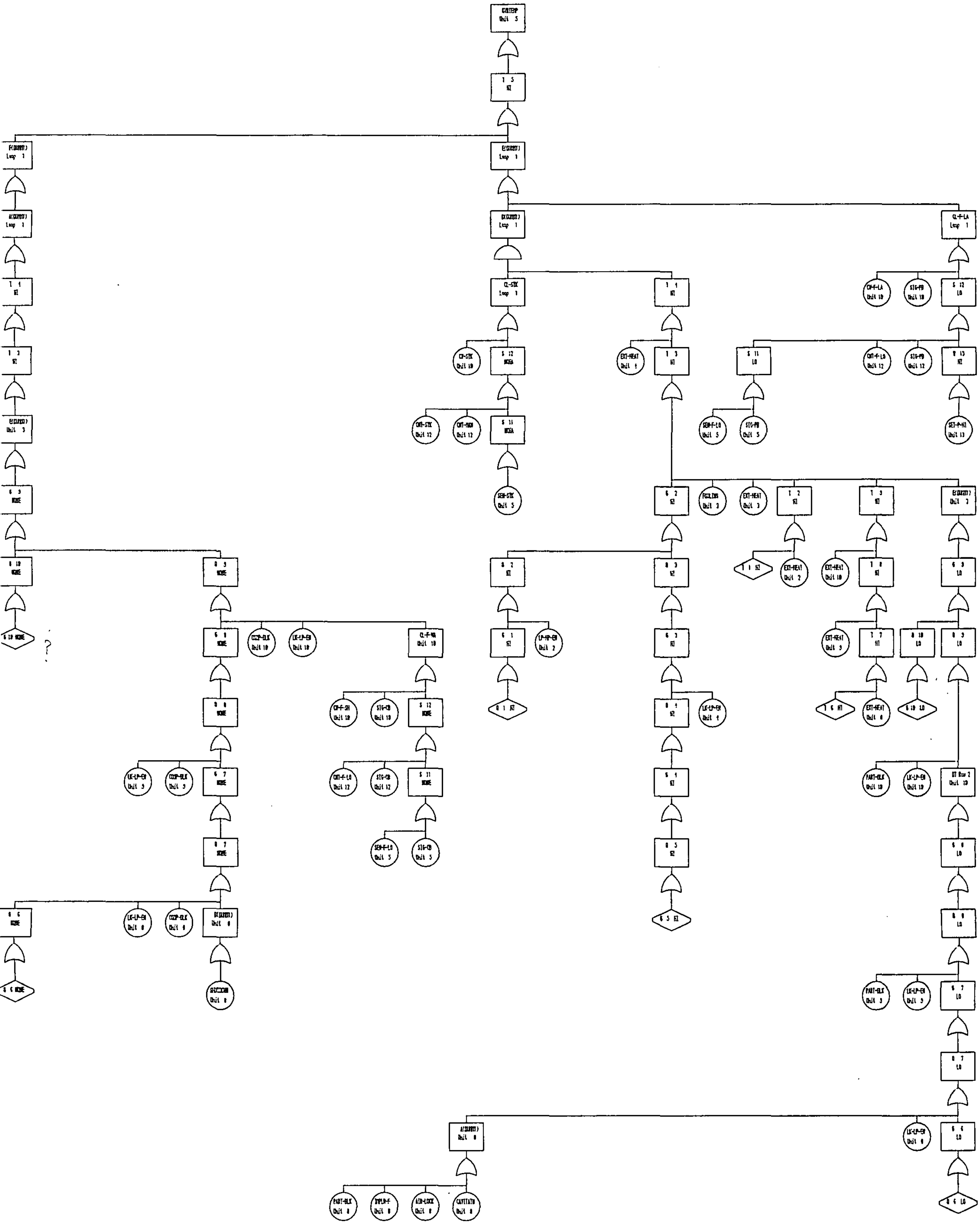


Figure 6.11 - complete fault tree for the
system shown in Figure 6.10

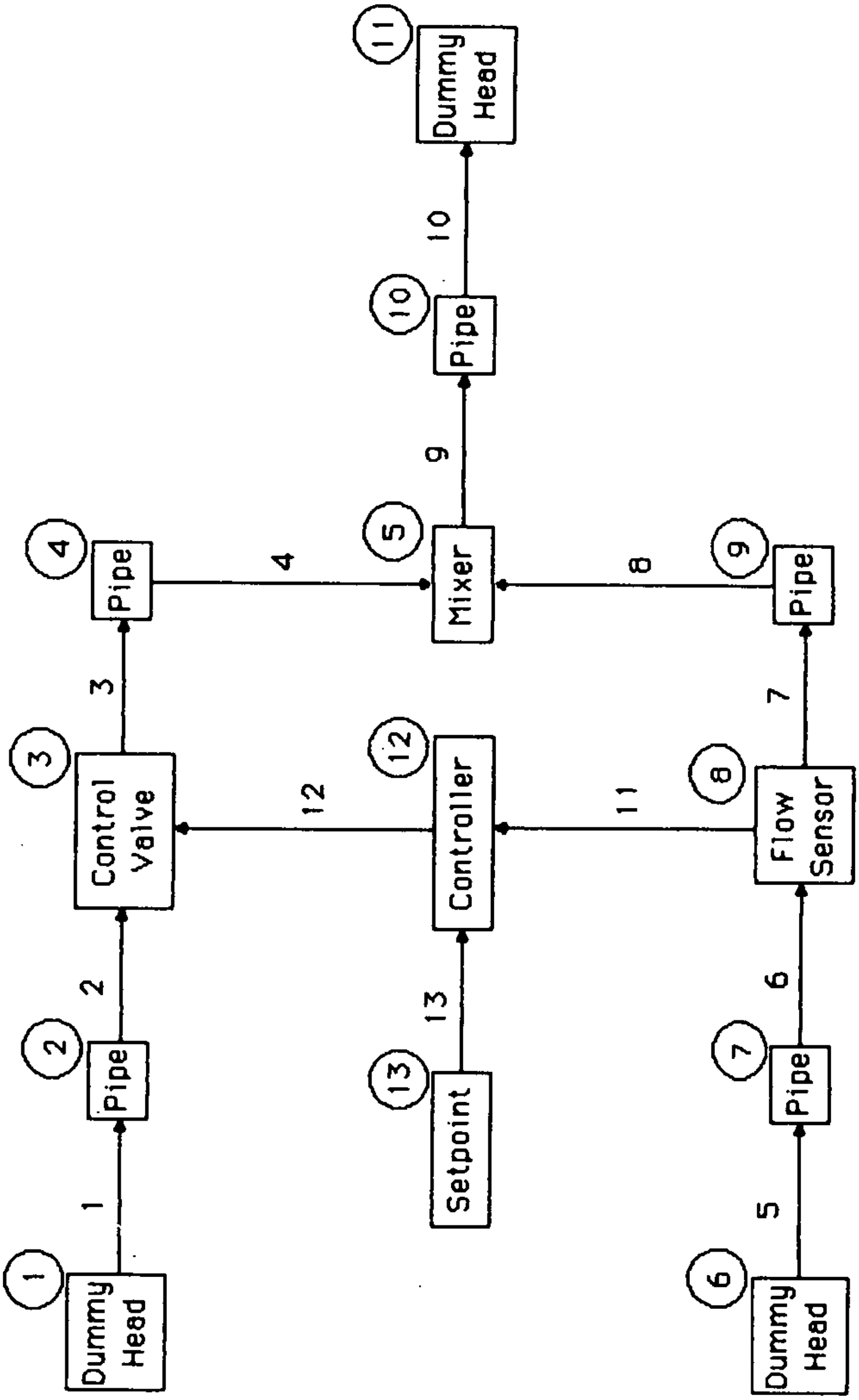


Figure 6.12 - configuration diagram for
a mixing system involving
a feedforward composition
control loop, after Taylor [43]

FeedForward Loop Example

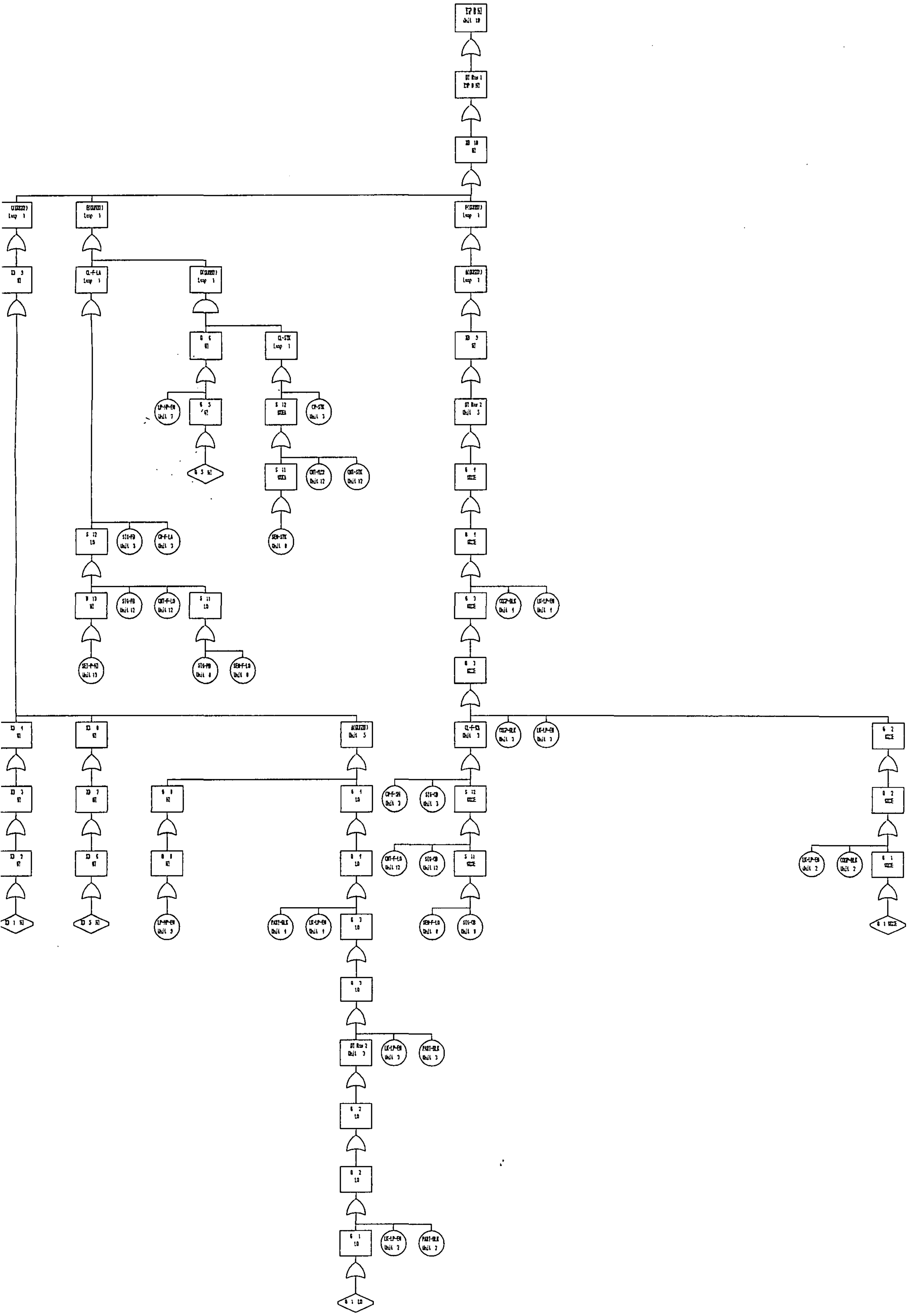


Figure 6.13 - complete fault tree for the system shown in Figure 6.12

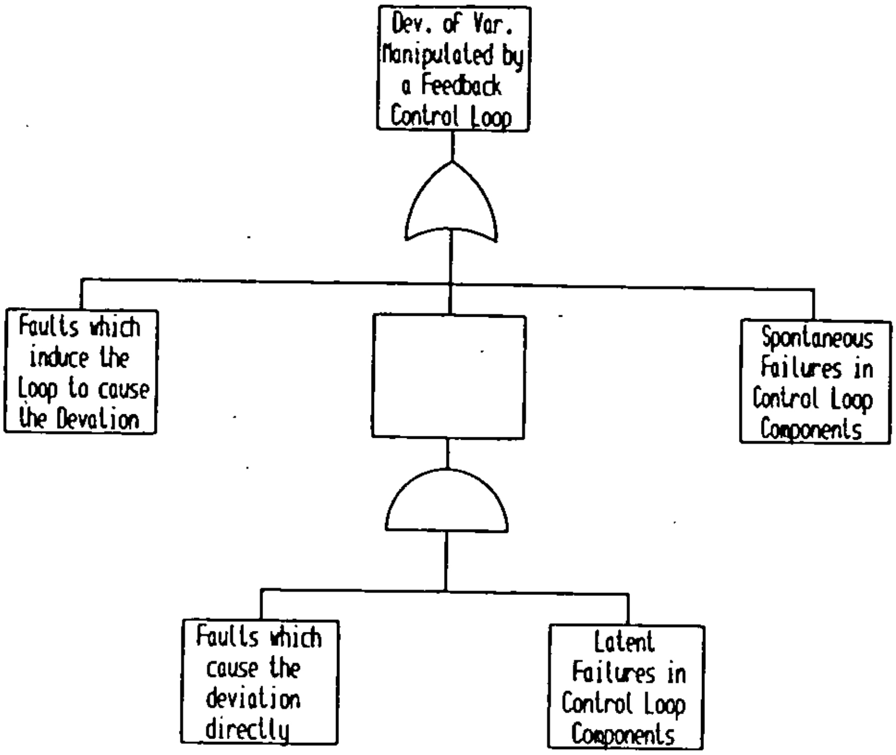

Dev. of Var. Manipulated by a Feedback Control Loop
Faults which induce the Loop to cause the Devation
Spontaneous Failures in Control Loop Components
Faults which cause the deviation directly
Latent Failures in Control Loop Components

Figure 6.14 - the general model for a manipulated variable deviation in a feedback control loop

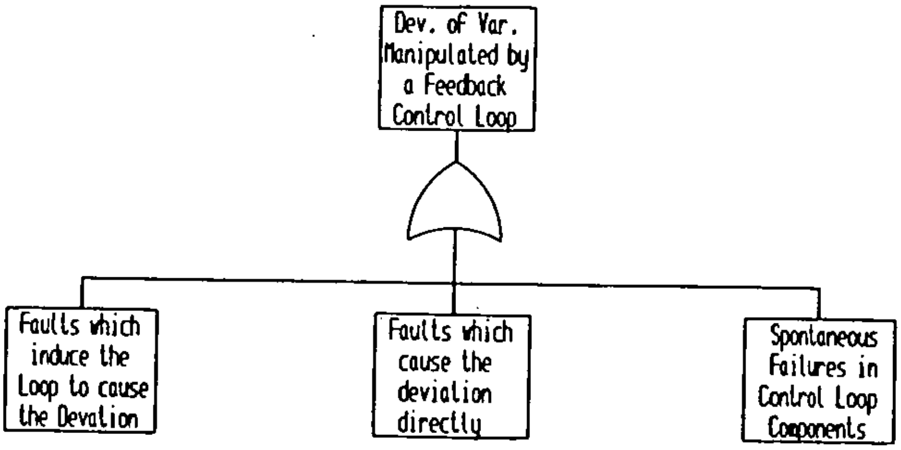

Dev. of Var. Manipulated by a Feedback Control Loop
Faults which induce the Loop to cause the Devation
Faults which cause the deviation directly
Spontaneous Failures in Control Loop Components

Figure 6.15 - the general model for a manipulated variable deviation in a feedforward control loop

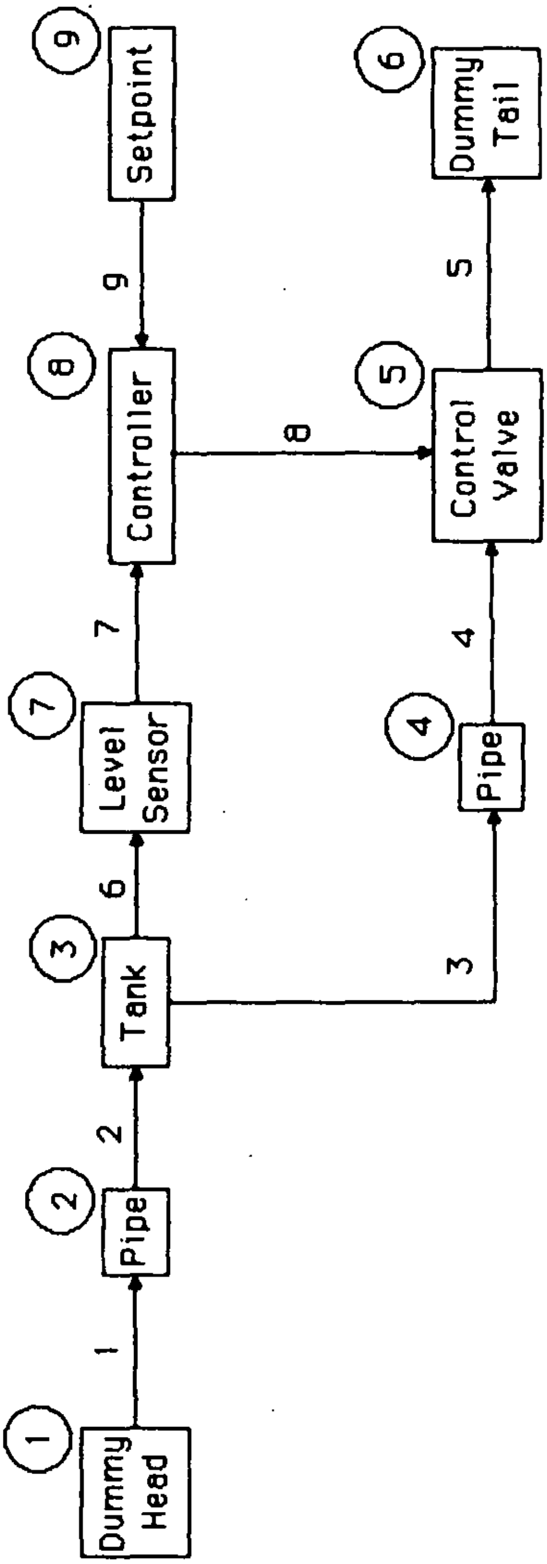


Figure 6.16 - configuration diagram for a level control system

Manipulated Variable Deviation Example

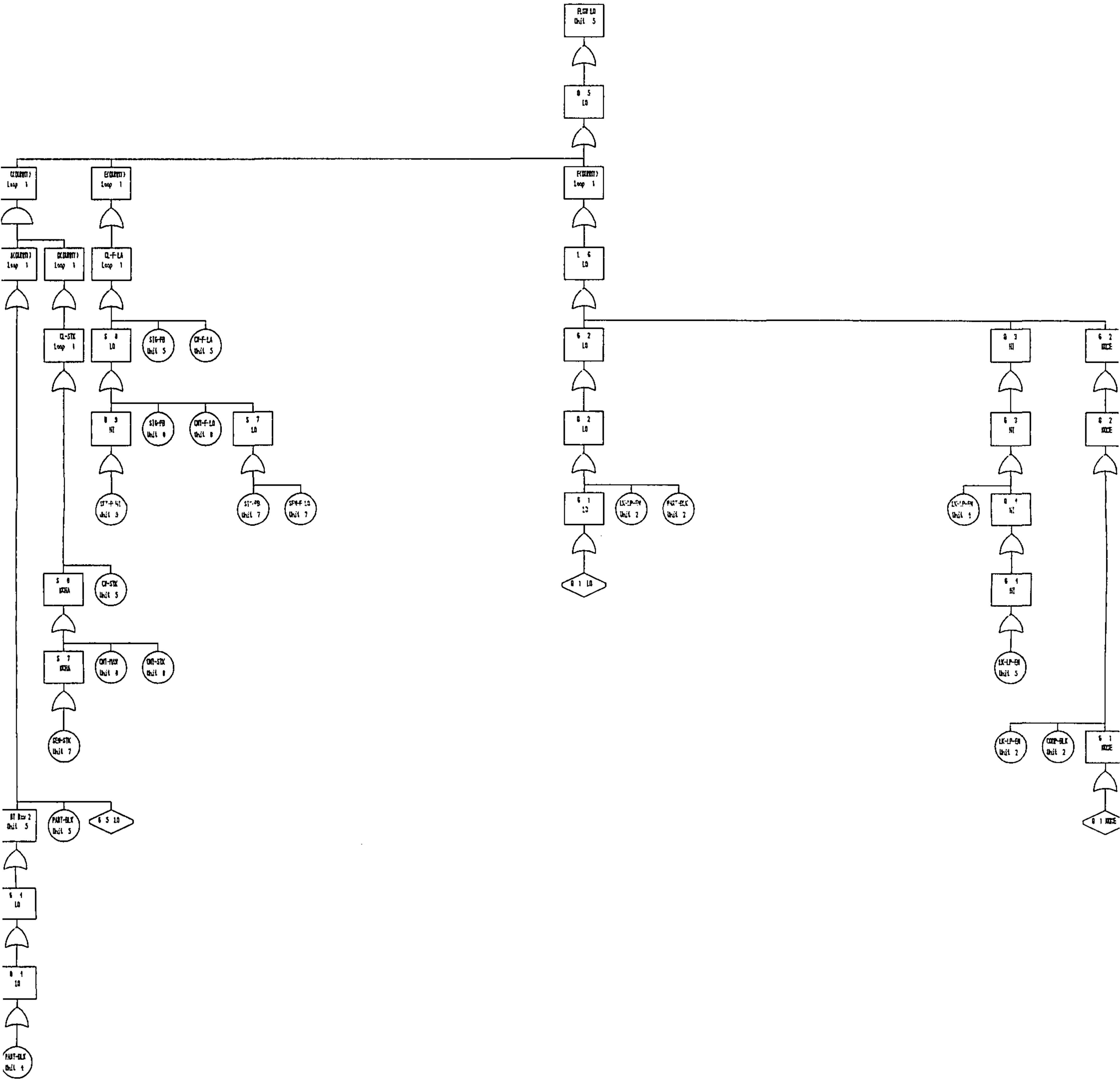


Figure 6.17 - complete fault tree for the system shown in Figure 6.16

FeedForward Loop Example

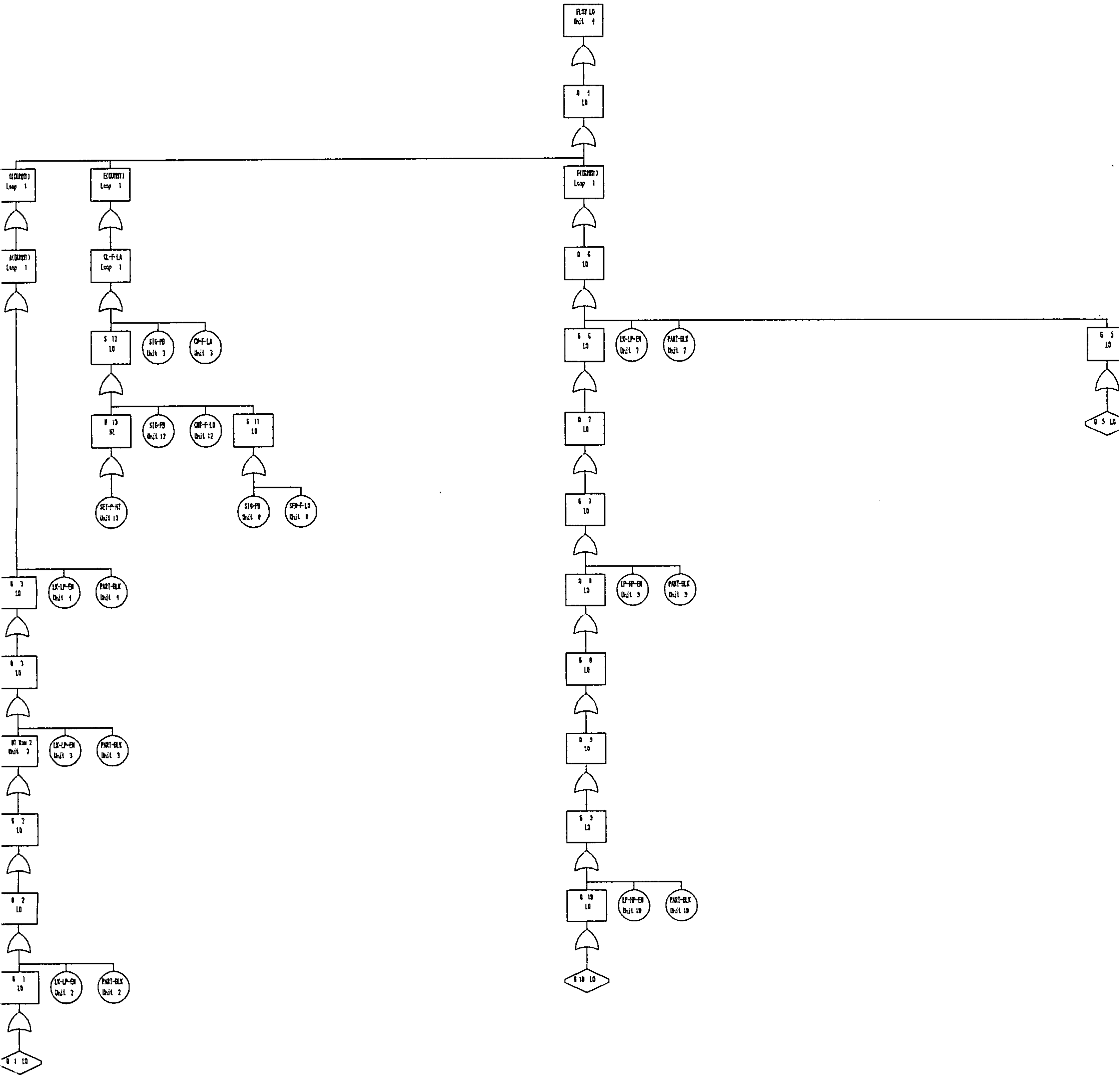


Figure 6.18 - complete fault tree for the system shown in Figure 6.12

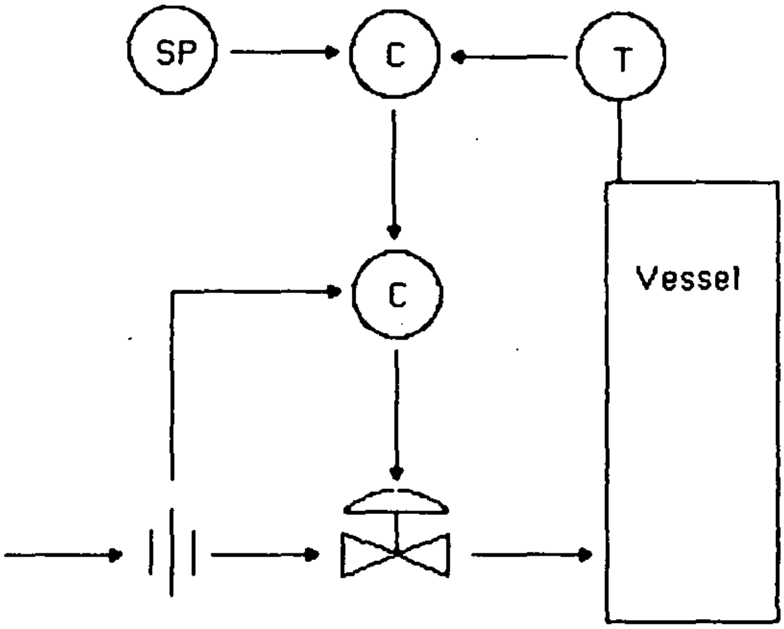


Figure 6.19 - a complex control loop :
a cascade control system

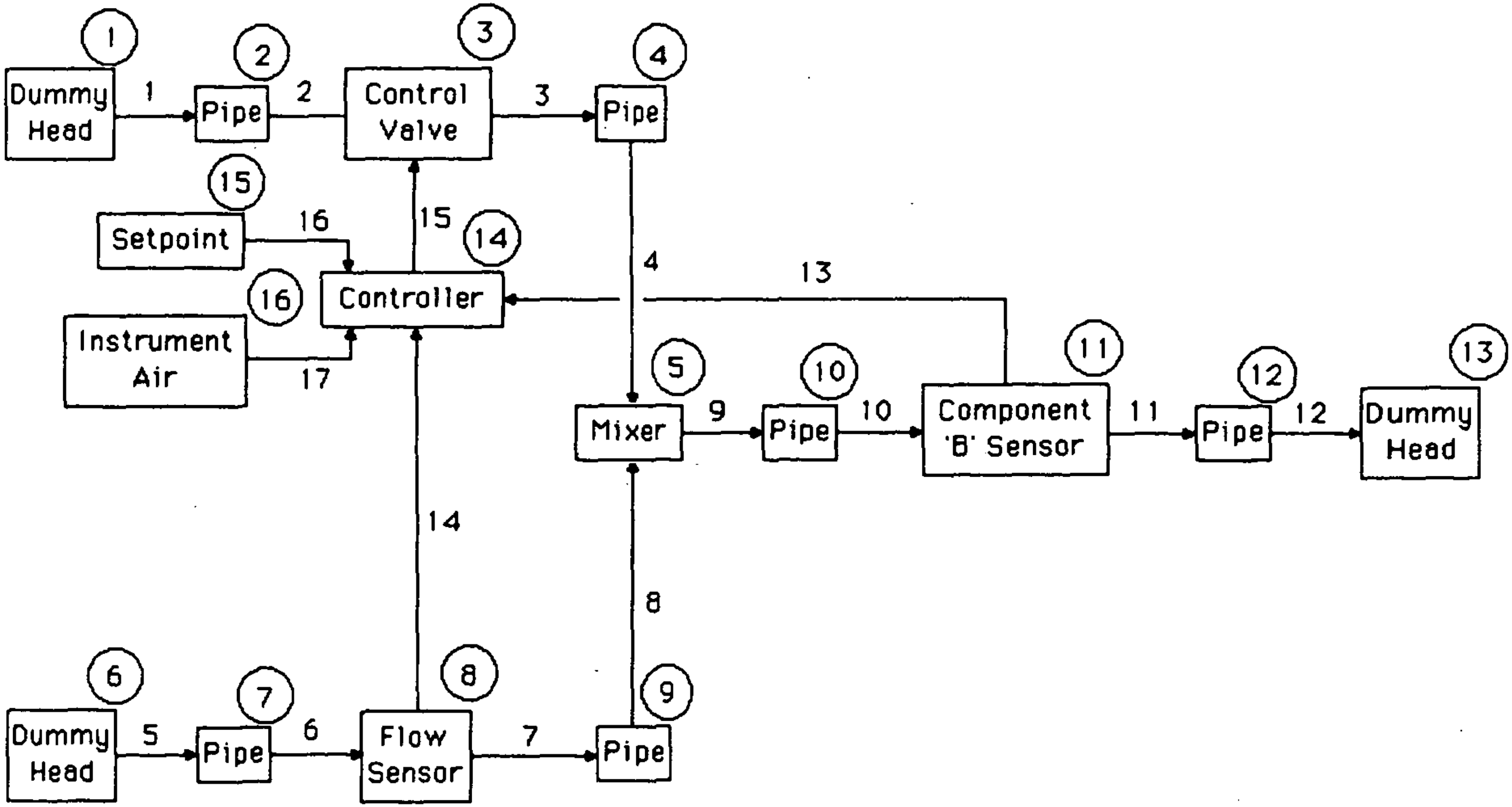


Figure 6.20 - a complex composition control system, after Taylor [43]

Pipeline Mixer System - 25th June 1984

Figure 6.21 - complete fault tree for the system shown in Figure 6.20

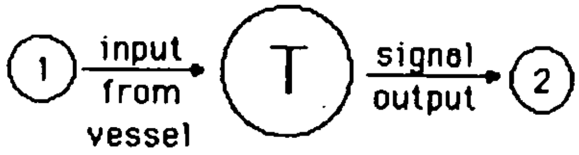


Figure 6.22 - the representation of a vessel-mounted temperature sensor

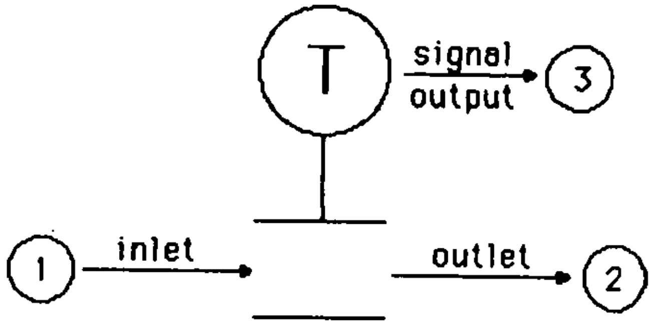


Figure 6.23 - the representation of a pipe-mounted temperature sensor

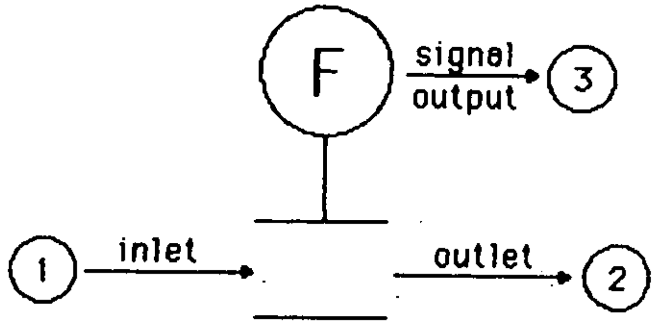


Figure 6.24 - the representation of a pipe-mounted flow sensor

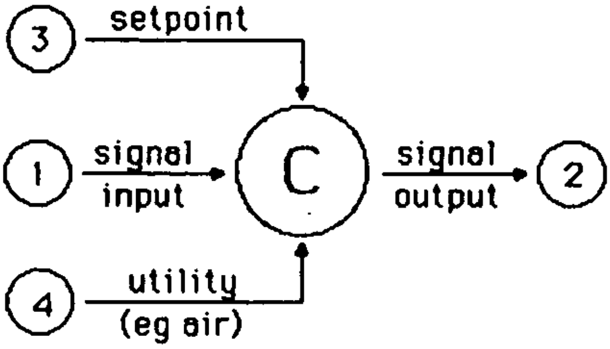


Figure 6.25 - the representation of
a simple inverse controller

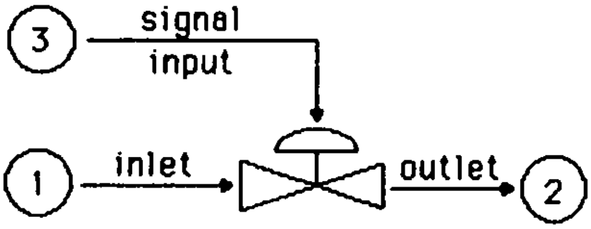


Figure 6.26 - the representation of
a control valve