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)

7) Trip Systems

Trip systems, like control loops, cannot satisfactorily be modelled using only component models. The reasons for this are similar to the reasons why control loops require a special treatment, viz. that the combination of models that together form a trip system does not contain all the information about the function of the trip system.

7.1) The Problem

Trip systems present problems of a similar nature to the problems encountered with control loops (see Section 6.1). In general terms, these problems are the same, namely deciding which events a particular trip system can protect against by operating correctly. However, the differences between trip systems and control loops mean that the treatment accorded to control loops (see Section 6.3) is not appropriate for trip systems. The principal reason for this is that the events that trip systems are designed to protect against are different from the events that control loops are designed to protect against.

An example will illustrate this point. Consider the configuration diagram displayed in Fig 7.1. The trip system is designed to prevent high composition of oxygen reaching some point further downstream by closing the trip valve. The trip system takes no action if the composition of oxygen is below some threshold value. This is significantly different from control loop action, which involves continuous regulation of the value of some variable.

Another difference between trip systems and control loops is that simply because a trip system can detect a particular event does not mean that it can correct it. In the approach used to handle control loops, it was assumed that all events, with the exceptions of no flow and reverse flow of the manipulated stream, could be corrected by a control loop, assuming that the control loop could detect the event. This assumption has been valid in all the examples studied with the methodology. However, consider again the system illustrated in Fig 7.1, and assume that the system is under vacuum pressure. A leak in pipe 4 will cause an intake of oxygen into the system. The trip sensor can detect this, but shutting the trip valve will not prevent oxygen going downstream.

These differences are addressed in the proposed solution to the problems presented by trip systems (Section 7.3).

7.2) The Approach of Others

The other researchers in this field have generally accorded trip systems a similar treatment to control loops (see Section 6.2).

The RIKKE code of J. R. Taylor [42-44] uses a complex modelling approach to include in the models information on how events may be compensated, as described in Section 6.2. To cover the different levels at which control loops and trip systems are normally set, Taylor uses three deviations of variable. Disturbed low and disturbed high deviations are correctable by control loops. Low and high deviations will overload control loops, but are correctable by trip systems. Very high and none deviations overload even trip systems. This approach complicates the modelling process, since it is necessary to model three times as many deviations.

Lapp and Powers [23] regard trip systems and control loops as information loops in the digraph that is a representation of the plant under study. The only difference noted between loops is whether they are feedforward loops or feedback loops. Different individual models for the protective system components are used to obtain the different fault trees associated with trip systems, as opposed to control loops.

Shafaghi [39-41], in modelling plants based on the protective systems in the plant, uses a general structure model approach for both trip systems and control loops. The general models are different, reflecting the different behaviour of the two types of protective system.

The models used by Lapp and Powers, and by Shafaghi are less concerned with the detail of leaks and blockages, and more concerned with the protective systems and information loops in the plant. The problems noted above (Section 7.1) will not, presumably, have been encountered by these researchers, since such problems centre around faults in the process units, and not in the protective systems.

Lawley [51] has studied manual fault tree synthesis, and considers the problems that trip systems present. His approach is to synthesise a fault tree for the top event of interest, initially ignoring any trip systems involved in the process. When the fault tree has been completely synthesised assuming that there are no trip systems present, Lawley reviews the fault tree and decides which faults can be detected by the trip system, and if these faults can then be prevented from causing the top event by the correct operation of the trip system. If such faults are both detectable and correctable, then the fault tree is modified to AND such faults with the failure of the trip system to act on demand.

7.3) A Solution

As was noted in Section 7.1, there is a need to treat control loops and trip systems using different techniques. The solution presented in this thesis is based on the approach of Lawley [51]. This involves synthesising the fault tree, initially ignoring the trip system, and then deciding which events can be protected against by the correct operation of the trip system. Implementing this approach in an automated methodology involves the synthesis of a number of separate fault trees, which will be called "sub-trees" to avoid confusion with the fault tree for the user-supplied top event.

This solution, like the control loop solution, requires that some additional information be provided as part of the configuration definition. This information is used to model the trip system behaviour.

The solution involves the following stages

a) synthesis of a sub-tree for the top event, but ignoring the possible effects of trip functional failures (ie failure of the trip system to act on demand). This sub-tree is known as the main sub-tree. A note is kept of all the trip systems that could possibly protect against each event in this sub-tree.

b) synthesis of a sub-tree to find the causes of the trip system failing to act for each trip system noted in a). This sub-tree is called the functional failure sub-tree.

c) synthesis of a sub-tree for the event "trip should activate" (the demand sub-tree), for each trip system noted in a).

d) comparison of the events in the main sub-tree and the demand sub-tree.

e) any events that occur in both the main and the demand sub-trees that the trip system can prevent from propagating further can be protected against by the correct operation of the trip system. The event in the main sub-tree should be ANDed with the functional failure sub-tree. The resulting tree is the final fault tree for the top event.

The problem with this approach is identifying which events can be prevented from causing the top event by the operation of the trip system. These are not simply the events that can be detected by the sensor (as is the case with control loops - see Section 6.3), as illustrated by the composition protection system considered in Section 7.1. The reason for this relates to the action taken by trip systems. Trip valves that are normally open shut in an attempt to prevent the event from propagating further. The only events, therefore, that such a trip system can prevent are those events beyond the trip valve on the propagation path. So, considering again the composition protection example of Section 7.1 (see Fig 7.1), the propagation is proceeding upstream (X5 HI - X4 HI etc). Therefore, only events upstream of the trip valve can be prevented from causing high composition downstream by shutting the trip valve. A leak in the pipe (Unit 4) is not upstream of the trip valve, and so cannot be prevented from causing the top event by the operation of the trip system.

Similar logic applies to the situation where the valve is normally shut, and opens in response to a demand. This situation is considered later (Section 7.3.2).

7.3.1) Trip Valve Normally Open

Consider again the configuration diagram of Fig 7.1. The trip system is designed to close the trip valve when the composition becomes high. The trip valve is assumed to be of the air-to-close type.

Two top events will be considered - HI COMP Unit 5 and LO COMP Unit 5, representing, respectively, high and low compositions downstream of the trip valve. Figs 7.2 to 7.5 show the four sub-trees relevant to these two top events. Fig 7.2 is the main sub-tree for the top event HI COMP Unit 5, and Fig 7.3 is the main sub-tree for the top event LO COMP Unit 5. These sub-trees would be the fault trees for the two top events if the trip system did not exist. Both these sub-trees involve faults that are potentially related to the trip system, since propagation through the trip valve occurs in both cases. However, whether there are any events that should cause the trip system to activate cannot be decided until the demand sub-tree has been synthesised. Therefore, for both systems, a functional failure sub-tree and a demand sub-tree are synthesised. These sub-trees are the same for both top events, and are given in Figs 7.4 and 7.5 respectively.

The functional failure sub-tree uses the deviation NCHA (no change) to represent the trip failing to act. Note that loss of instrument air (IAR-LOSS Unit 9) is one cause of trip functional failure. SHAC in the

demand sub-tree means "should activate"; the only causes of this are a genuine trip demand, which in this case is high composition at the sensor. The difference between the event "trip should activate" and the event "trip does activate" is important. Trip should activate is essentially a logical state, rather than an actual plant state, and represents the conditions that should exist in the plant for the trip to activate. Trip does activate is a completely different state, caused not only by the causes of trip should activate, but also the trip activating when there is no demand. This latter is known as operational failure, and is considered in Section 7.3.7.

The next step is the comparison of the main sub-trees with the demand sub-tree. In the current example, the only events that appear in both the main sub-tree and in the demand sub-tree, beyond the trip valve in the direction of propagation, are LK-HP-EN Unit 2 and X1 HI. They must therefore be ANDed with the trip functional failure sub-tree to produce the final fault tree for the top event HI COMP Unit 5. The event LK-HP-EN Unit 4, although it appears in both sub-trees, is not beyond the trip valve in the direction of propagation, and so is not ANDed with trip functional failure. The final fault tree for HI COMP Unit 5 is given in Fig 7.6.

The final fault tree for the top event LO COMP Unit 5 is identical to the main sub-tree (see Fig 7.3), because there are no events in the main sub-tree that are also in the demand sub-tree.

7.3.2) Trip Valve Normally Closed

Trip systems in which the valve is normally closed are treated in an almost identical manner to that outlined above for systems with normally open valves. The only differences between the two types are the possible causes of trip functional failure, and the identification of events that the trip system can prevent from causing the top event.

Consider the configuration diagram of Fig 7.7, derived from an example originally presented by Lawley [51]. A pump is designed to deliver liquid from a tank to some downstream user. To prevent the pump from overheating should the demand fall dramatically, a protective system has been installed. A flow sensor is designed to open a trip valve which provides a path for flow back to the tank. Since the trip activates only at very low flows, the NONE deviation of flow will be used to model the situation when the trip should activate.

A study of the configuration diagram will reveal that the faults that may result in trip functional failure include not only faults in the trip system components, but also the causes of no flow in the return line to the tank. The functional failure sub-tree must include these causes.

There are three sub-trees relevant to the top event NO FLOW Unit 4. The main sub-tree, ignoring possible functional failures is given in Fig 7.8; the functional failure sub-tree is given in Fig 7.9; Fig 7.10 displays the demand sub-tree. Note that the functional failure sub-tree (Fig 7.9) does not contain the event IAR-LOSS Unit 13 (complete loss of instrument air), since the trip valve is assumed to be of the air-to-close type.

Instrument air loss is a cause of the trip system activating when there is no demand, but is not a cause of it failing to act when there is a demand.

A trip valve that is normally shut opens to provide an alternative flow path. The only events such a trip system can prevent from causing the top event are therefore those events in parallel flow paths. In the current example, these are the causes of Q6 NONE, namely COMP-BLK Unit 7, LK-HP-EN Unit 7 and G7 NONE. The trip system cannot protect against events not in parallel flow paths. In this example, these include blockages in the pump and neighbouring pipes (Units 2 and 4).

In the current example, all the causes of Q6 NONE appear in both the main and demand sub-trees, and so should be ANDed with trip functional failure, as shown in the final fault tree of Fig 7.11.

Thus there are two differences between trip systems with valves that are normally open, and those with valves that are normally shut

a) causes of no flow in the line that is opened for flow when a trip system with a normally closed valve activates are causes of trip functional failure, and as such must appear in the functional failure sub-tree

b) trip systems with normally open valves can only protect against events beyond the trip valve on the propagation path; trip systems with normally closed valves can only protect against events in flow paths parallel to the flow path that is opened when the trip valve opens

7.3.3) Trip Systems with Multiple Sensors

Trip systems frequently have a bank of sensors all measuring the same variable, and incorporating voting logic in trip switches. Common voting patterns are two out of three, and one out of two.

Such systems are treated as if there were only one sensor involved, but with suitable configuration changes. Fig 7.12 is the configuration diagram for a trip system involving three sensors, with the switch taking action based on the readings of two of the sensors. The system is designed to prevent high composition propagating further. The only difference between this example and Fig 7.1 is the multiple sensors. The configuration input must include the information that all the sensors are part of the trip system.

Figs 7.13 to 7.15 are the three sub-trees relevant to the top event HI COMP Unit 7. The only difference between these sub-trees and the sub-trees if there was only a single sensor (Figs 7.2, 7.4 and 7.5) is in the trip functional failure sub-trees (Figs 7.14 and 7.4). The demand sub-trees are identical, since the presence of additional sensors does not affect the conditions which should cause the trip system to activate.

7.3.4) Complex Trip Systems

Complex trip systems are systems where a number of different sensors are connected to the same trip valve (but not including voting sensors), or where the same sensor (or group of voting sensors) is connected to a number of trip valves. Even more complex systems may exist where several sensors are interlinked and connected to several trip valves.

The handling of such complex systems is similar to the treatment used to handle complex control loops. A single complex is resolved into a number of simple systems where a single sensor is (or group of voting sensors are) connected to a single trip valve.

For example, the complex trip system shown in Fig 7.16 can be resolved into four simple systems

a) three composition sensors to the first trip valve
b) three composition sensors to the second trip valve
c) two temperature sensors to the first trip valve
d) two temperature sensors to the second trip valve

The complete set of sub-trees for this plant involves nine sub-trees, one main sub-tree and four pairs of trip sub-trees, (the demand sub-tree and the functional failure sub-tree, one pair for each trip system). Figs 7.17 to 7.20 display the functional failure sub-trees for each trip system. As with complex control systems, note that there are some events that are common to several trip functional failure sub-trees.

7.3.5) Feedforward Trip Systems

The trip systems studied to date have all been feedback trip systems, that is, systems in which the activation directly affects the variable detected by the sensor. However, as with control loops (see Section 6.3.1.2), it is possible to have feedforward trip systems, where the variable detected by the sensor is unaffected by the activation of the trip valve.

An example of a feedforward trip system is given in Fig 7.21. This is based on the Lapp-Powers heat exchanger example [23], but ignoring the control loop. The trip system is designed to protect against one of the potential causes of high nitric acid in connections 3 and 4. The system will shut off the flow of nitric acid through the heat exchanger if complete coolant loss occurs. This system is feedforward because shutting the trip valve has no effect on the coolant flow.

The synthesis of the various sub-trees presents no problems, and the three sub-trees are given in Figs 7.22 to 7.24. The problems arise in deciding which events the trip system can protect against. The solution is to investigate not only events beyond the trip valve (of which there are none that appear in both the main and demand sub-trees in this example), but also events that are in other branches of AND gates which are above the trip valve. The necessary AND gates are not apparent in the main sub-tree of Fig 7.22, but they are in fact present. The AND gate could be included in one of two ways. Firstly, the heat exchanger model could have an expression that high temperature will result if there is no flow of coolant AND some flow of nitric acid. However, a more general

way is to use the top event for high temperature given in Section 4.1.4, where the temperature deviation was ANDed with some flow.

The final fault tree when this procedure is followed is displayed in Fig 7.25.

7.3.6) A Problem with Flow

There is one situation in which the procedure to treat trip systems is inadequate. This relates to certain trip systems involving flow faults. Consider the level protection system illustrated in Fig 7.26. The trip system is designed to prevent high level in the tank by completely shutting off the inlet flow. Fig 7.27 is the fault tree for the top event HI FLOW Unit 4, ignoring the trip system. There are two types of cause, namely faults that will result in an increase in tank level (such as RACING Unit 3), and faults that will not (such as HV-F-OP Unit 11). The causes of the top event that also cause high level will not result in the top event unless the trip has functionally failed, since the trip will act to prevent any flow at all. However, careful consideration of the events that are beyond the trip valve indicate that no events will be ANDed with trip functional failure. The only events beyond the trip valve are high flow downstream and tank leaking, neither of which will result in high tank level.

The problem does not relate to the procedure used to analyse trip functional failures, but to a deficiency in the modelling of flow. The flow models given

indicate that some cause upstream, such as pump racing, is sufficient to cause a high flow. There is one assumption in this, and that is that there is a flow path downstream. If a trip valve shuts, then this assumption is invalid, and the modelling is inadequate. A suitable model incorporates a number of AND gates, which essentially state that not only must there be a cause of the flow deviation, there must also be a flow path. A suitable model is displayed in Fig 7.28. This model is similar to the models (see Section 4.1.1) required to correctly model the causes of some flow and reverse flow in situations where the assumption of a flow path is not valid.

This type of model includes an AND gate in the fault tree. The trip system analysis procedure will use this AND gate in the manner described in the section on feedforward trips (Section 7.3.5), and will arrive at the correct fault tree structure. The top event model is suitable for use whenever a top event of high flow is considered. The fault tree synthesised from the top event will, however, differ from the fault tree synthesised from the normal high flow top event model (a model with Q2out HI as its only cause) only when the assumption of a flow path is invalid.

However, using this solution is cumbersome, since it complicates the modelling process. It also presents additional problems to the synthesis package, and so the solution is not supported.

7.3.7) Trip System Operational Failure

Trip system operational failure, that is activation of the trip system in the absence of a genuine demand, requires no special treatment. Careful modelling of the trip valve and trip system component models is sufficient to model operational failure.

Fig 7.1 will be used as a simple example to illustrate operational failure. Fig 7.29 is the fault tree for the top event NO FLOW Unit 5. There are two basic causes of this, namely, faults in the pipework (e.g. COMP-BLK Unit 2), and the trip valve closes. The valve can close either because of a genuine demand (high composition detected by the sensor), or because the trip failed operationally. No distinction between these two types of cause of the trip valve closing is made in the modelling of the trip system units.

Because the trip system requires an air signal to close the valve, the deviation S SOME was used to model the state of the signal to close the trip valve. Because trip functional failure uses the deviations NCHA and SHAC only, both SOME and NONE are available for use to model the state when trip activation occurs. In this case SOME was used. However, had trip activation resulted from an interruption in the signal, S NONE would have been more appropriate to model the causes of the trip valve closing.

7.3.8) Trip System Component Modelling

As with control loop components, it is necessary to have special conventions when modelling trip system components.

7.3.8.1) Sensors

There is no difference in the modelling of sensors for trip systems and control loops, and the same models are used. This is an essential requirement, since some sensors are common to both trip and control loops. The requirements for modelling sensors are detailed in Section 6.3.4.1.

7.3.8.2) Trip Switches

Different models are required for

a) switches which activate on different deviations
b) switches which activate by interrupting a signal, or emitting a signal
c) switches with different voting patterns
d) electrical or pneumatic output signal

A number of examples will serve to illustrate this point.

Consider a trip switch with a single input that is designed to activate when this signal is high. Activation involves emitting a signal. Fig 7.30 is the representation of this switch.

The deviation SOME will be used to represent the trip switch activating (S2sig SOME), that is emitting a signal. There are various causes of this

a) the trip switch fails on (TSW-F-ON)
b) the trip switch receives a high input signal (S1sig HI)
c) the setpoint value is too low (W3in LO)

Trip functional failure, represented by S2sig NCHA may be caused by any of the following

a) the trip switch fails invariant (TSW-STK)
b) the input signal is invariant (S1sig NCHA)
c) the setpoint value is too high (W3in HI)
d) complete loss of utility has occurred (S4utl NONE)
e) the output port is blocked (pneumatic switch only - SIG-CB)

S2sig SHAC, which represents the situation where the trip switch should activate, has only a single cause

a) the trip switch receives a high input signal (S1sig HI)

If the switch incorporated 2/3 voting logic on the basis of signal input from three sensors, then the model would be slightly different. Fig 7.31 is the representation of this trip switch. The cause of the should activate deviation (S4sig SHAC) is unchanged (S1sig HI), but functional failure and trip activation now require that two of the input signals are of the correct deviation. For functional failure (S4sig NCHA), b) above should be replaced by

b) two of the three inputs signals are invariant (S1sig NCHA AND S2SIG NCHA, or S1sig NCHA AND S3sig NCHA, or S2sig NCHA AND S3sig NCHA). Decision tables are the best way of expressing this information

The causes of trip activation (S4sig SOME) should be modified along similar lines.

Switches that activate on different input signal deviations will require correspondingly different changes in input signal and setpoint.

Switches that activate by stopping an output signal are, however, significantly different. Trip activation in such switches is best modelled by use of the NONE deviation. The principal difference in the models for this category and the air-to-activate category is the effect of complete loss of utility. In no-air-to-activate switches, utility loss will result in trip activation. In air-to-activate switches, utility loss is a cause of functional failure.

7.3.8.3) Trip Valves

Models for trip valves should make use of the intermediate event TL-FN-F (trip loop functional failure) to represent the faults that may cause trip functional failure ANDed with the event "trip should activate". Additionally, the intermediate event TL-OR-F (operational failure) is available for use if desired. All variable deviations that may require the trip to be functionally failed must have as a possible cause TL-FN-F.

Fig 7.32 is the representation of a normally open trip valve. It is assumed that the valve requires a signal to shut it. The events that may require functional failure are all the events that do not involve no flow, no pressure and no relief. These latter events may result from trip activation. There are thus a large number of events that may require trip functional failure (high and low temperature, composition and flow etc). In a particular plant section, typically only one of these will actually require functional failure to occur, depending on the sensor associated with the trip valve.

Since the valve is air-to-close, the following may result in functional failure

a) trip valve fails to close
b) no change in the signal input

Conversely, trip activation will occur if

a) trip valve fails shut
b) some signal input is received

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

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

I TL-FN-F:G1in HI,G1in LO,G1in SOME,G1in REV
I TL-FN-F:Q2out HI,Q2out LO,Q2out SOME,Q2out REV
I TL-FN-F:T2out HI,T2out LO
I TL-FN-F:X2out HI,X2out LO
```

```
I TL-OR-F:G1in NONE,Q2out NONE

I C(DUMMY) AND V S3sig SHAC:TL-FN-F
F TV-FT-SH:C(DUMMY)
V S3sig NCHA:C(DUMMY)

F TV-F-SH:TL-OR-F
V S3sig SOME:TL-OR-F
```

C(DUMMY) is an intermediate event used to represent the causes of trip functional failure. Note that it is not necessary to associate any AND gates with the effects of TL-FN-F. Although in fault trees, events will be ANDed with functional failure, the inclusion of the AND gate is handled automatically.

The model is similar in many respects to the pipe model, but the effects of leaks and blockages in the valve have been ignored for simplicity. The model above ignores pressure, relief and temperature and composition under reverse flow conditions. However, these can easily be included, following the principles used above.

Fig 7.33 is the representation of a normally closed trip valve. Assuming that the valve is air-to-close, the possible causes of functional failure are

a) trip valve fails to open
b) there is no change in the input signal

Trip activation will result if

a) trip valve fails open
b) there is loss of input signal

Since the valve is air-to-close, the deviation NONE is better suited to modelling trip activation than the deviation SOME. The events that may require trip functional failure are the opposite of the events for a normally open trip valve, that is the events involving no flow, pressure and relief. Events such as some flow require that trip activation occurs.

The model for the normally closed trip valve is based on the model for a normally shut isolation valve, with extra information provided for the trip system. Most variable deviations must be ANDed with the trip valve opening, represented in the model below by the intermediate event TL-OR-F. The exception is no flow (G1in NONE and Q2out NONE), which may require functional failure of the trip (TL-FN-F) as well as restrictions on the other side of the trip valve.

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

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

I TL-OR-F AND V T1in LO:T2out LO
I TL-OR-F AND V T1in HI:T2out HI
I TL-OR-F AND V X1in LO:X2out LO
I TL-OR-F AND V X1in HI:X2out HI
I TL-FN-F:G1in NONE,Q2out NONE
V S3sig NONE:TL-OR-F
F TV-F-OP:TL-OR-F
I C(DUMMY) AND V S3sig SHAC:TL-FN-F
V S3sig NCHA:C(DUMMY)
F TV-FT-OP:C(DUMMY)
```

```
I TL-OR-F V G1in SOME T Q2out SOME
I TL-OR-F V Q2out SOME T G1in SOME
```

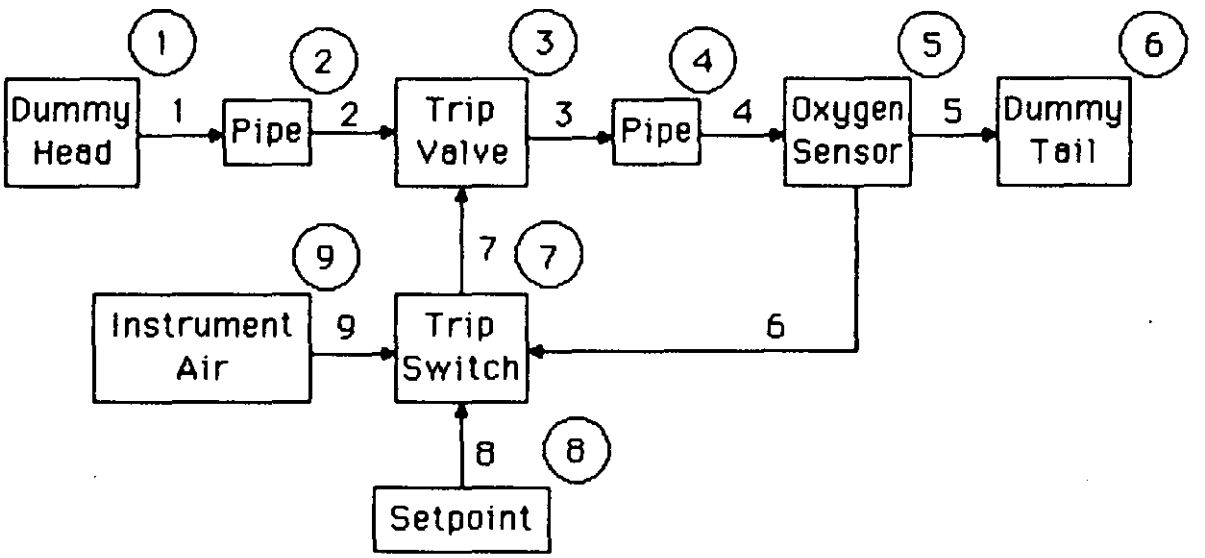

Figure 7.1 - configuration diagram of a trip system designed to prevent oxygen flowing downstream

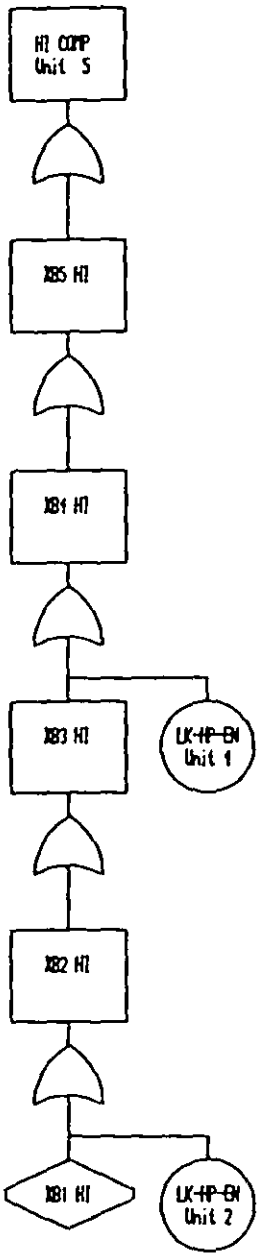

Figure 7.2 - the main sub-tree for the top event oxygen reaches downstream, for the system shown in Figure 7.1

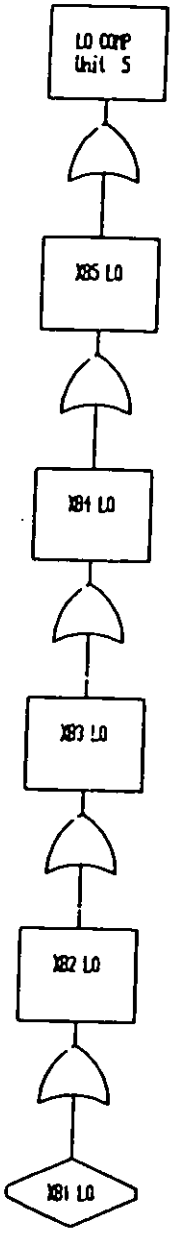

Figure 7.3 - the main sub-tree for the top event oxygen does not reach downstream, for the system shown in Figure 7.1

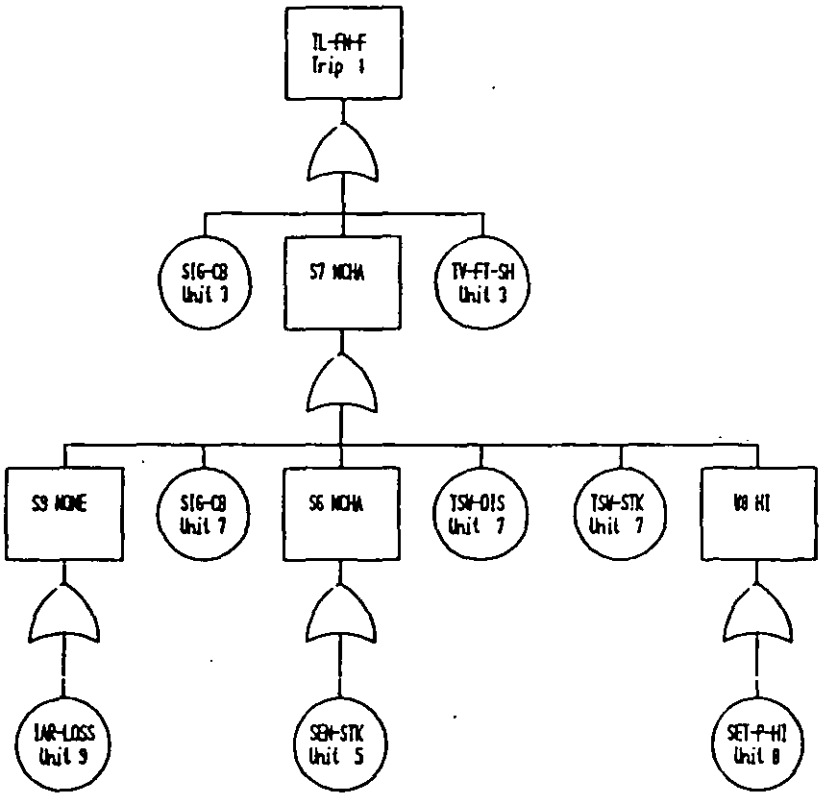

Figure 7.4 - the functional failure sub-tree for the system shown in Figure 7.1

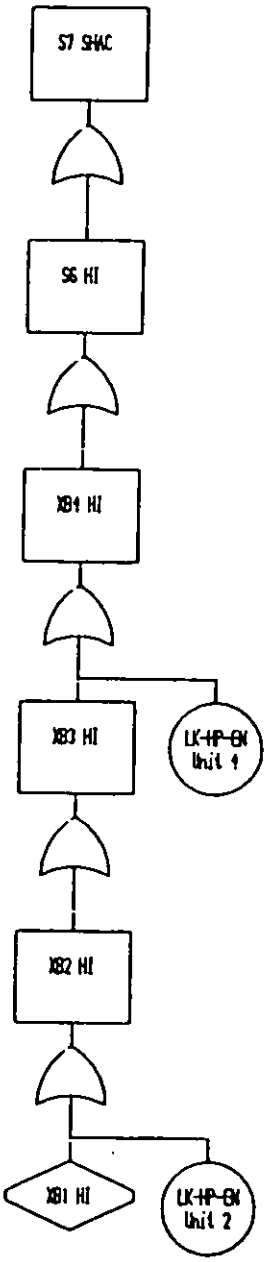

Figure 7.5 - the demand sub-tree for the system shown in Figure 7.1

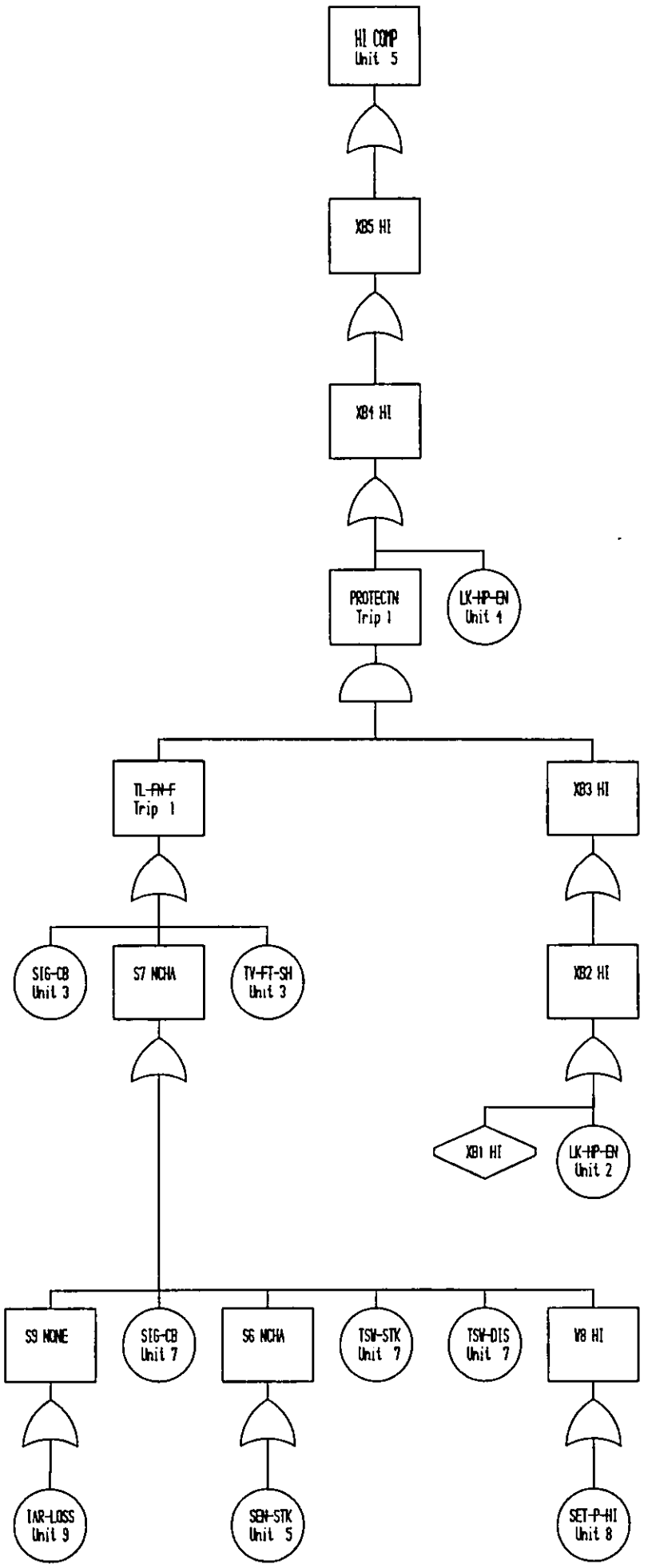

Figure 7.6 - the complete fault tree for
the system shown in Figure 7.1

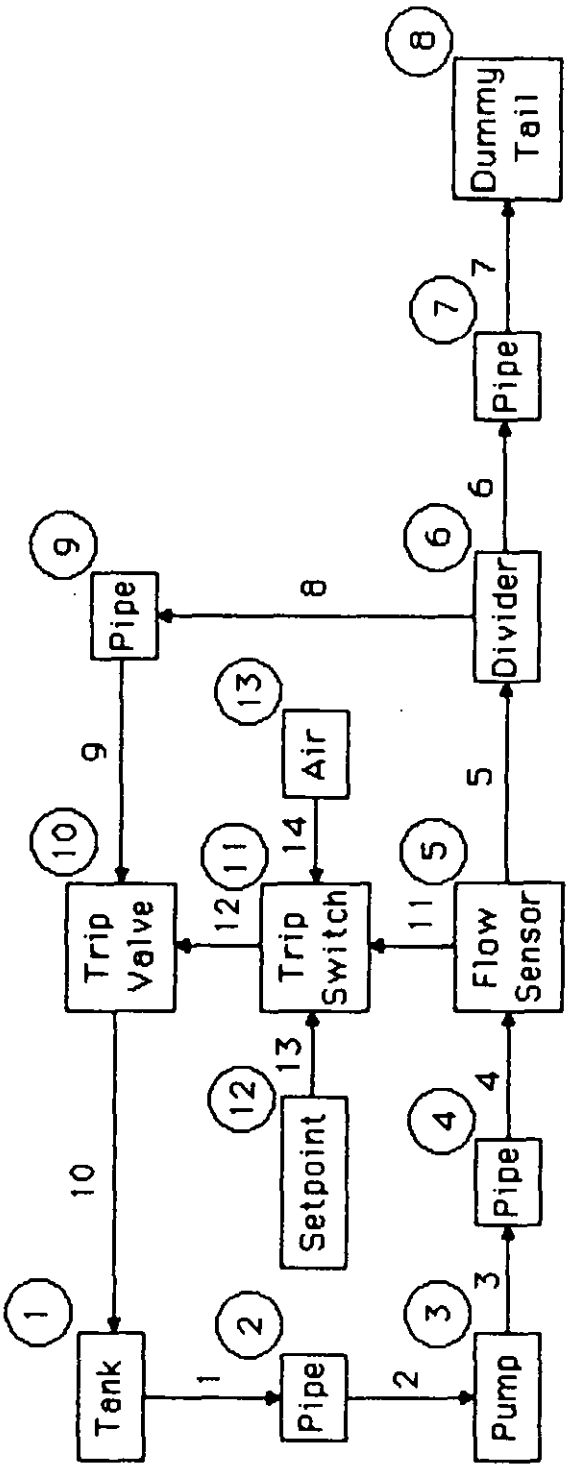

Figure 7.7 - configuration diagram for a pump protection system, after Lawley [51]

Pump Protection System Example

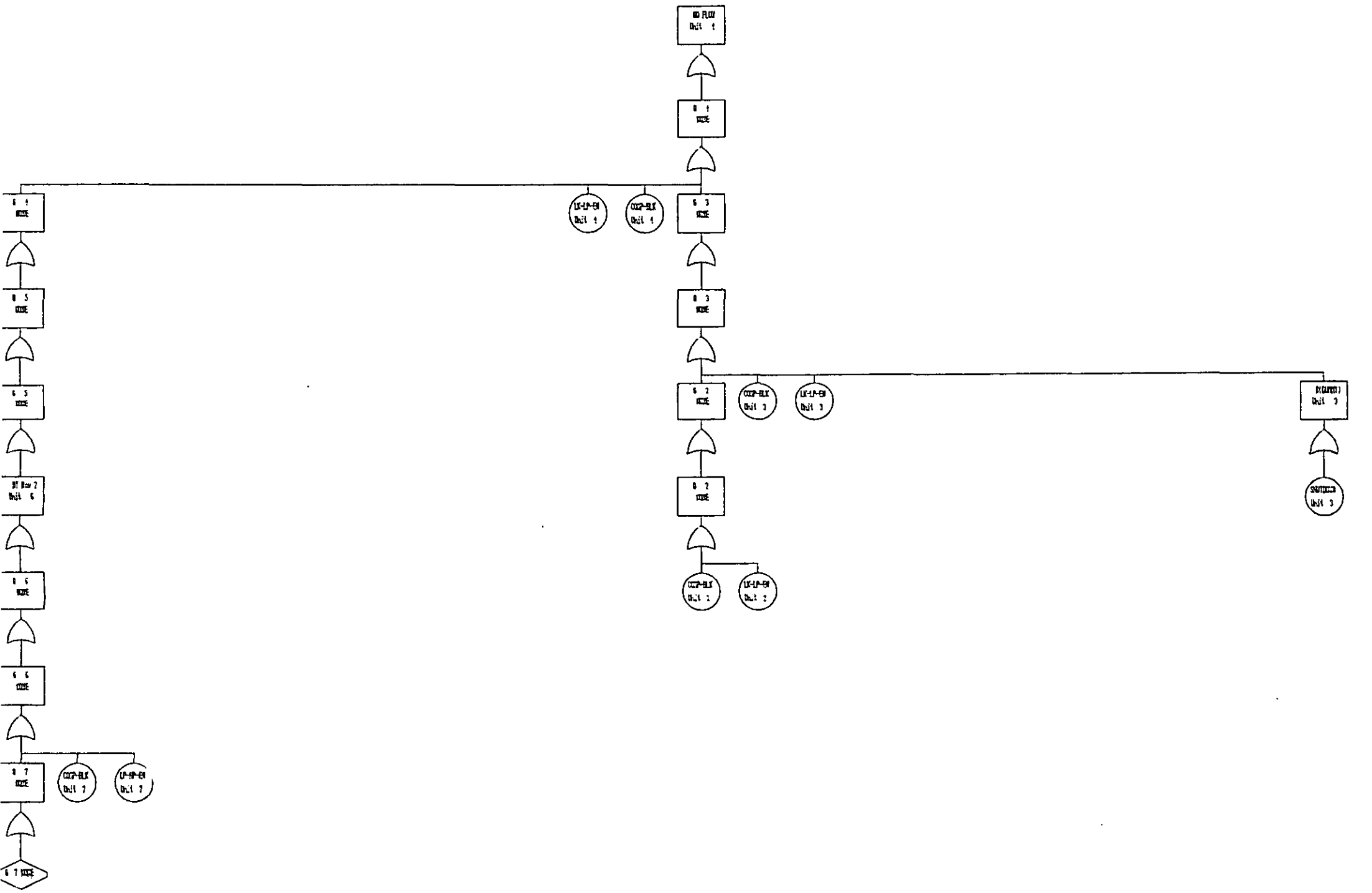

Figure 7.8 - main sub-tree for the
system shown in Figure 7.7

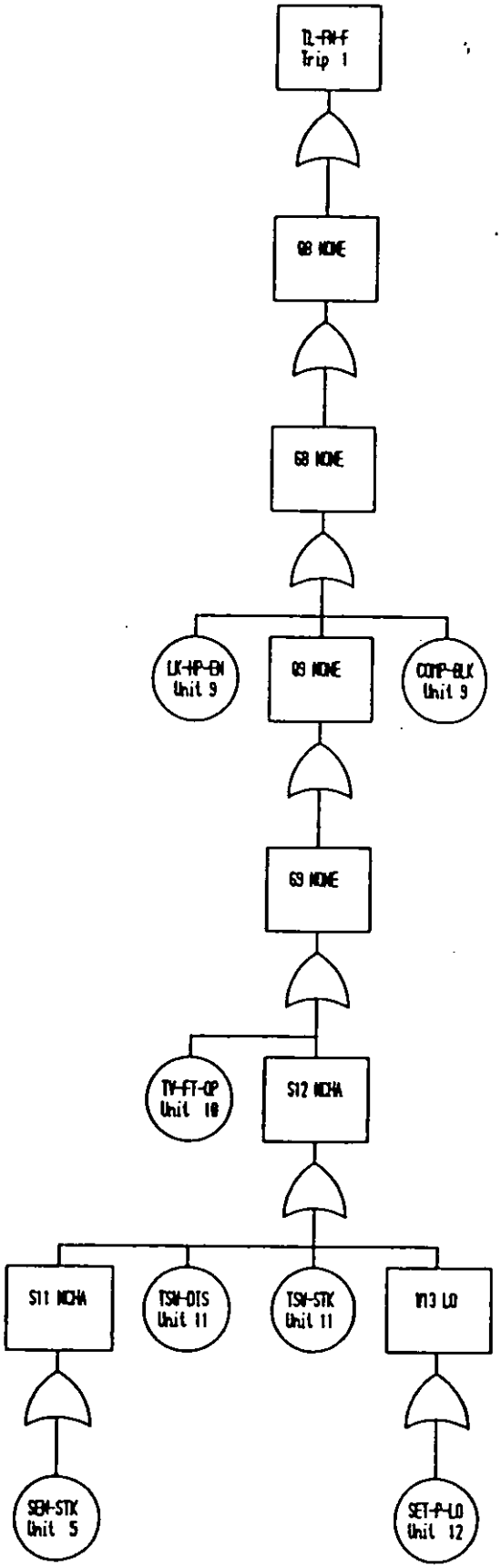

Figure 7.9 - the functional failure sub-tree for the system shown in Figure 7.7

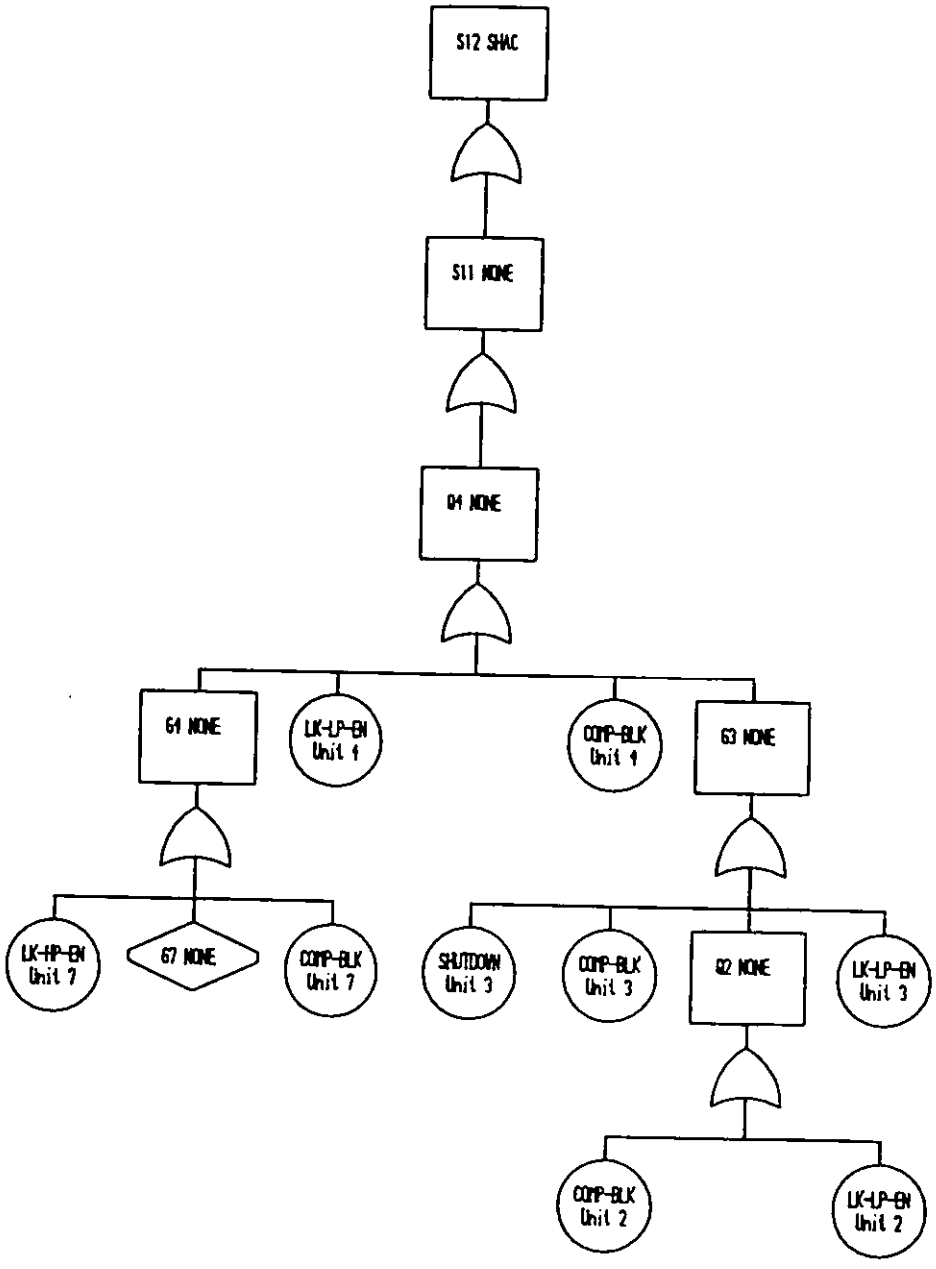

Figure 7.10 - the demand sub-tree for the system shown in Figure 7.7

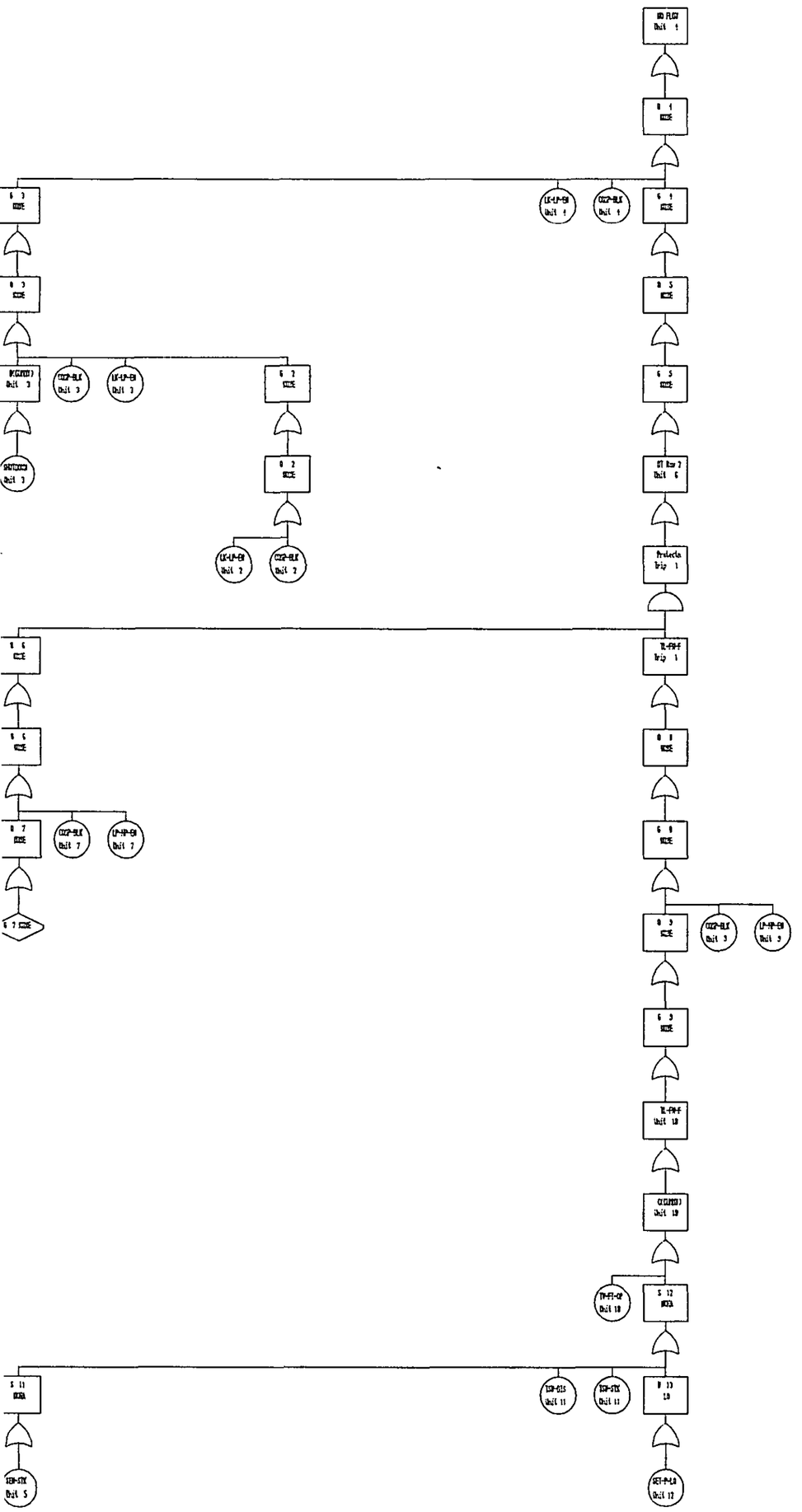

Figure 7.11 - complete fault tree for the system shown in Figure 7.7

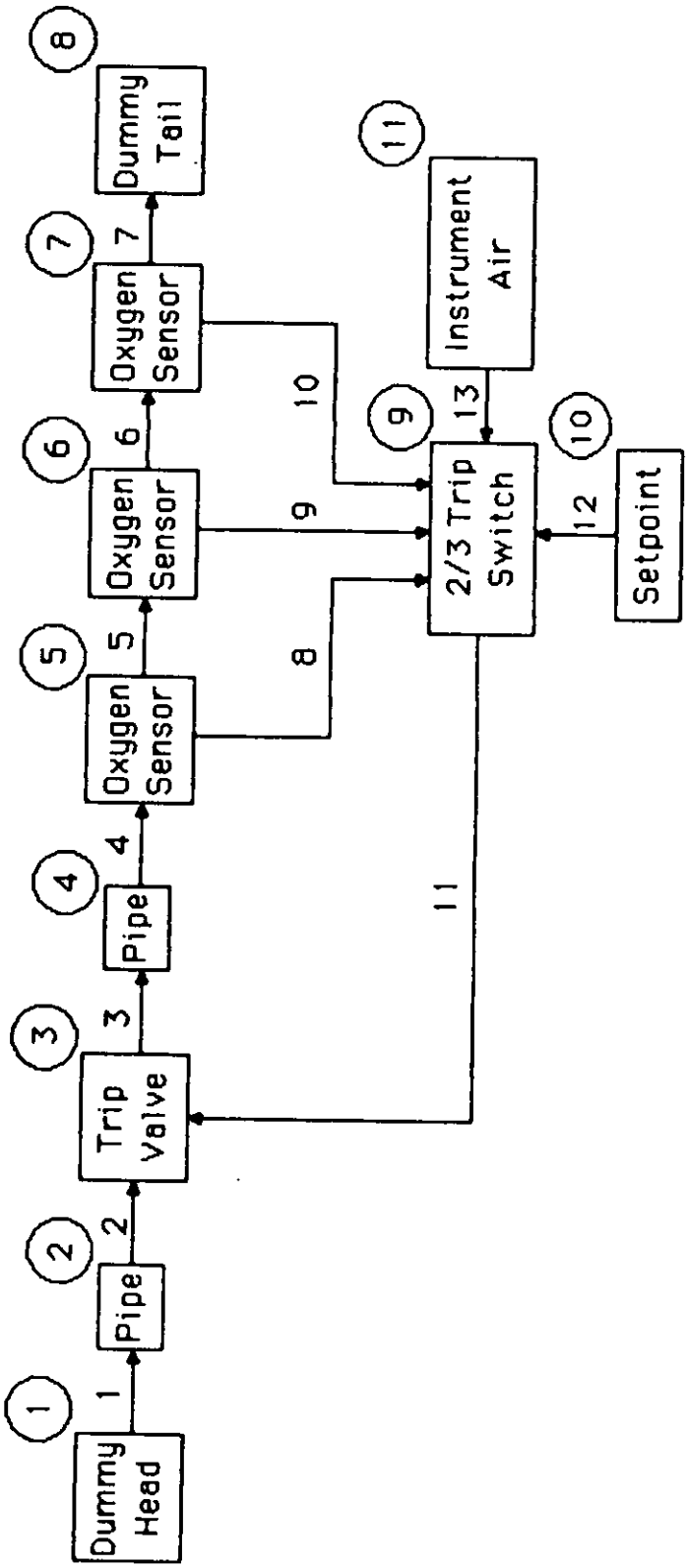

Figure 7.12 - configuration diagram for a composition protection system incorporating voting sensors, after Lihou [48]

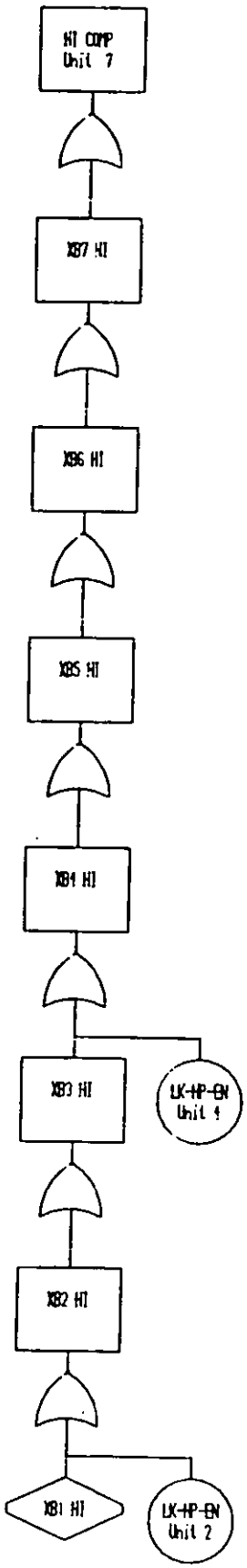

Figure 7.13 - the main sub-tree for the top event oxygen reaches downstream, for the system shown in Figure 7.12

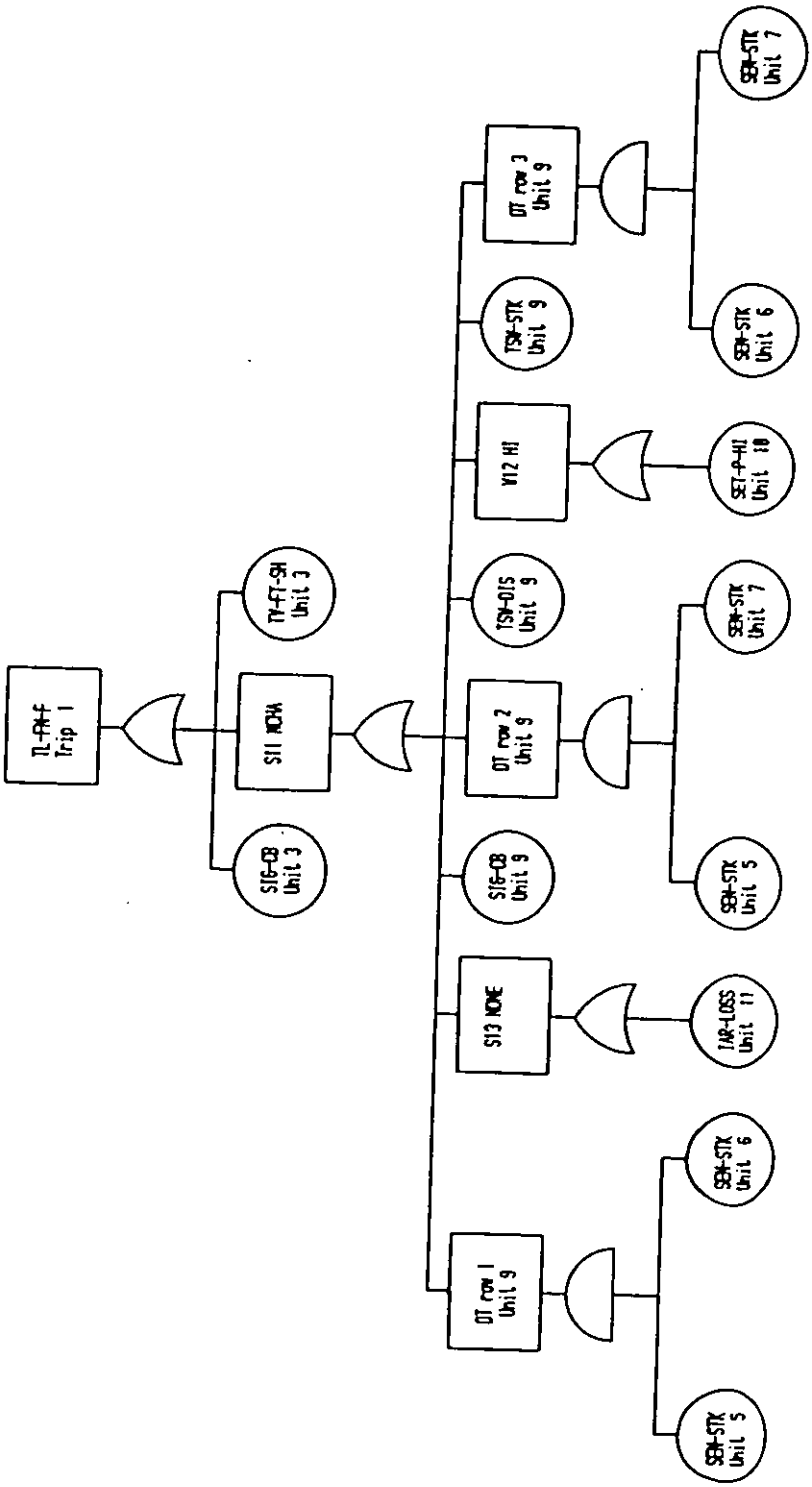

Figure 7.14 - the functional failure sub-tree for the system shown in Figure 7.12

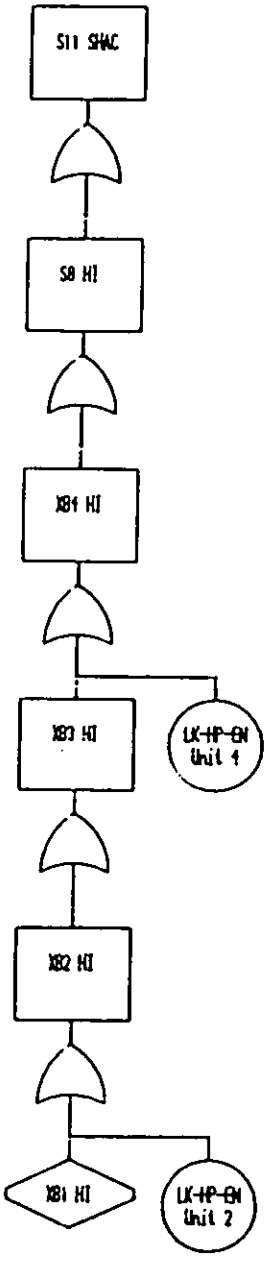

Figure 7.15 - the demand sub-tree for the system shown in Figure 7.12

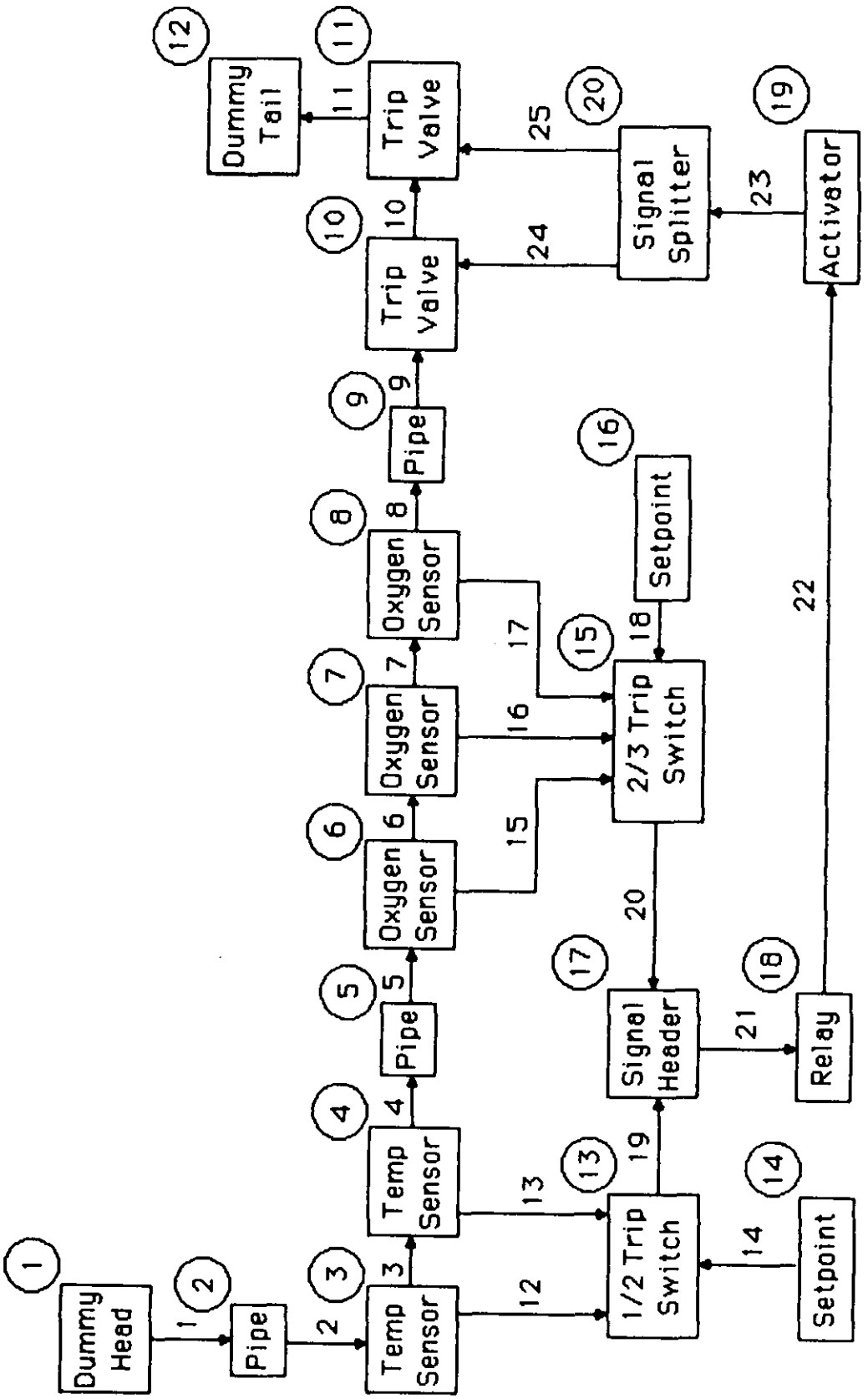

Figure 7.16 - configuration diagram for a system incorporating a complex trip system, after Lihou [48]

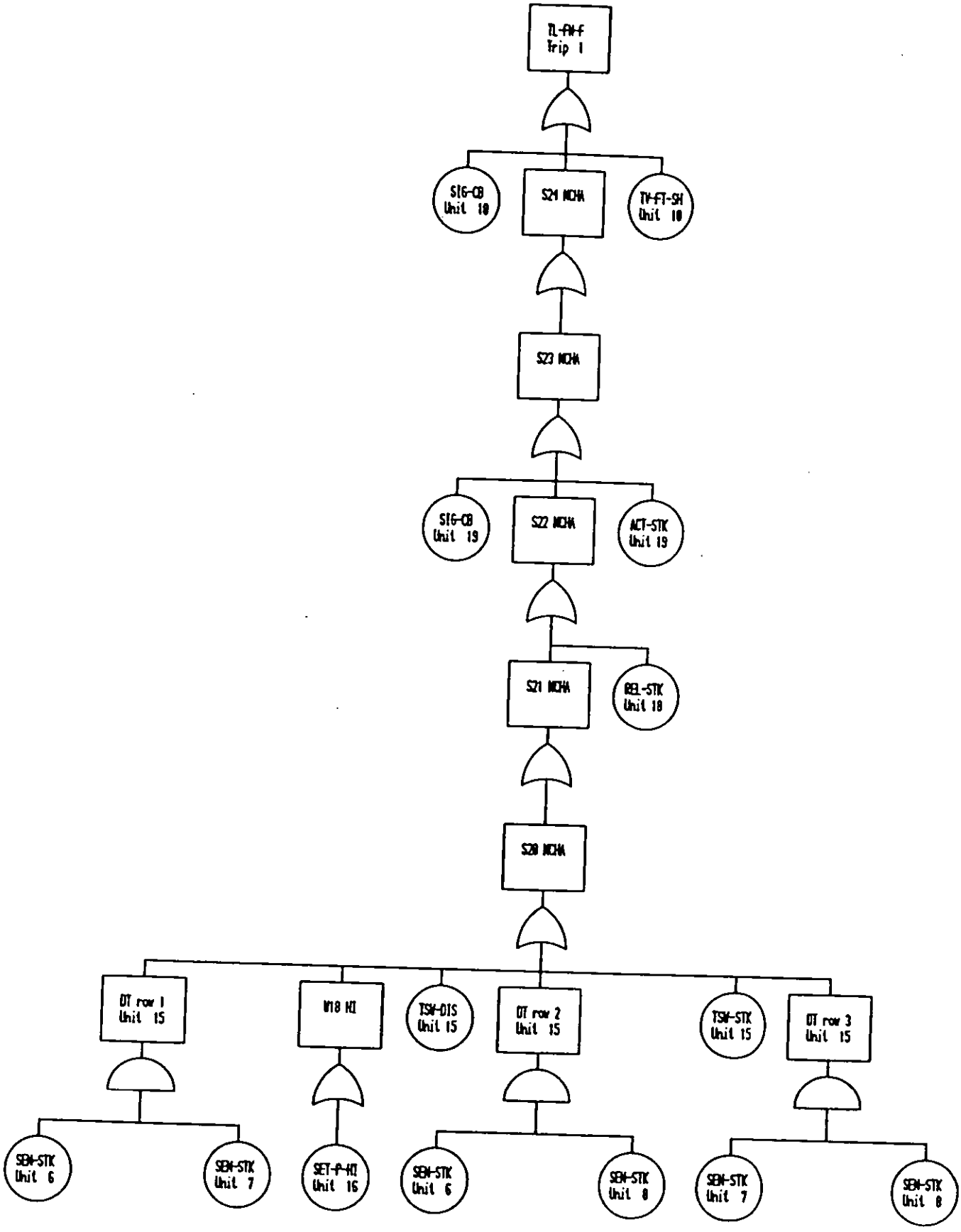

Figure 7.17 - the functional failure sub-tree for the first trip system of Figure 7.16

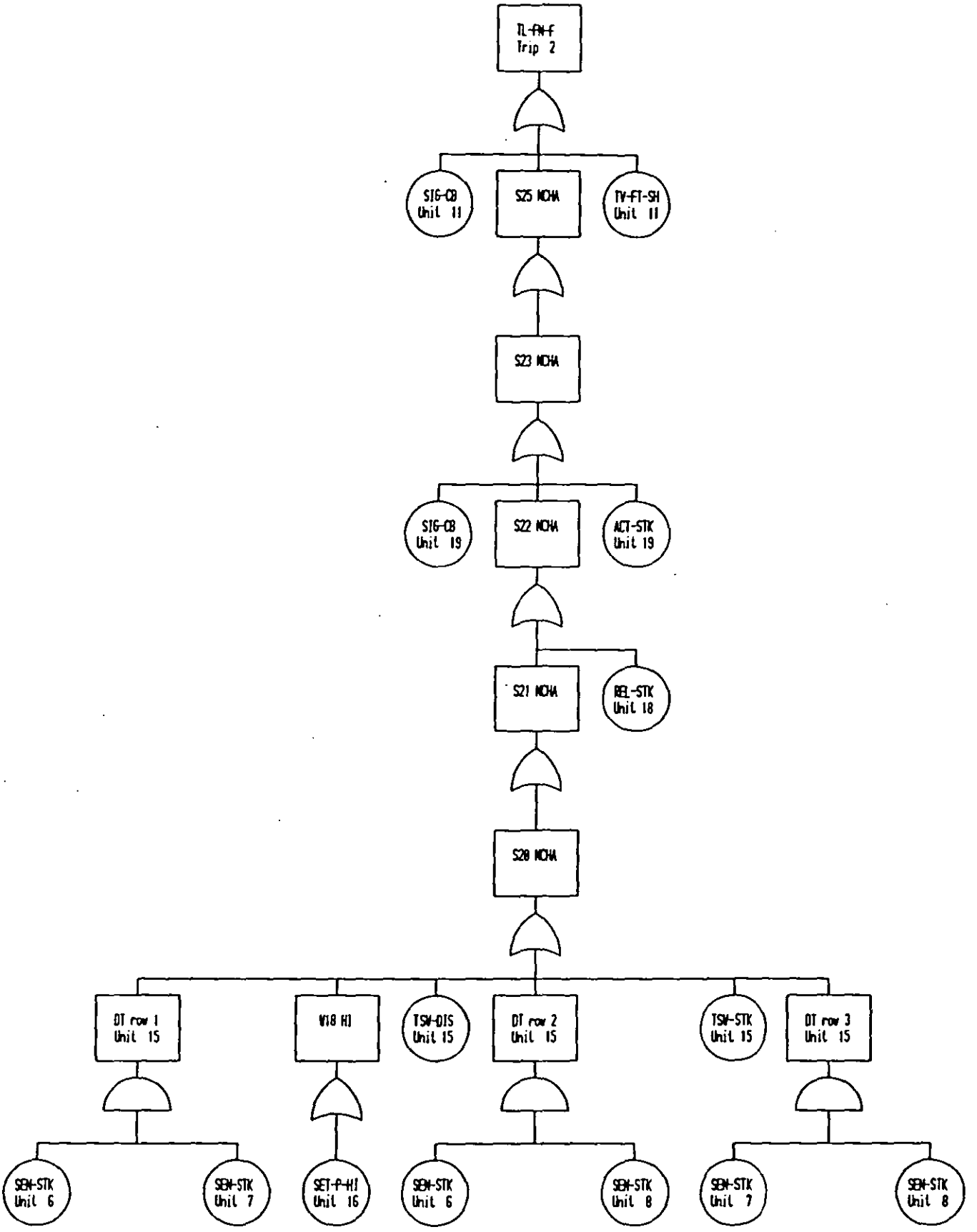

Figure 7.18 - the functional failure sub-tree for the second trip system of Figure 7.16

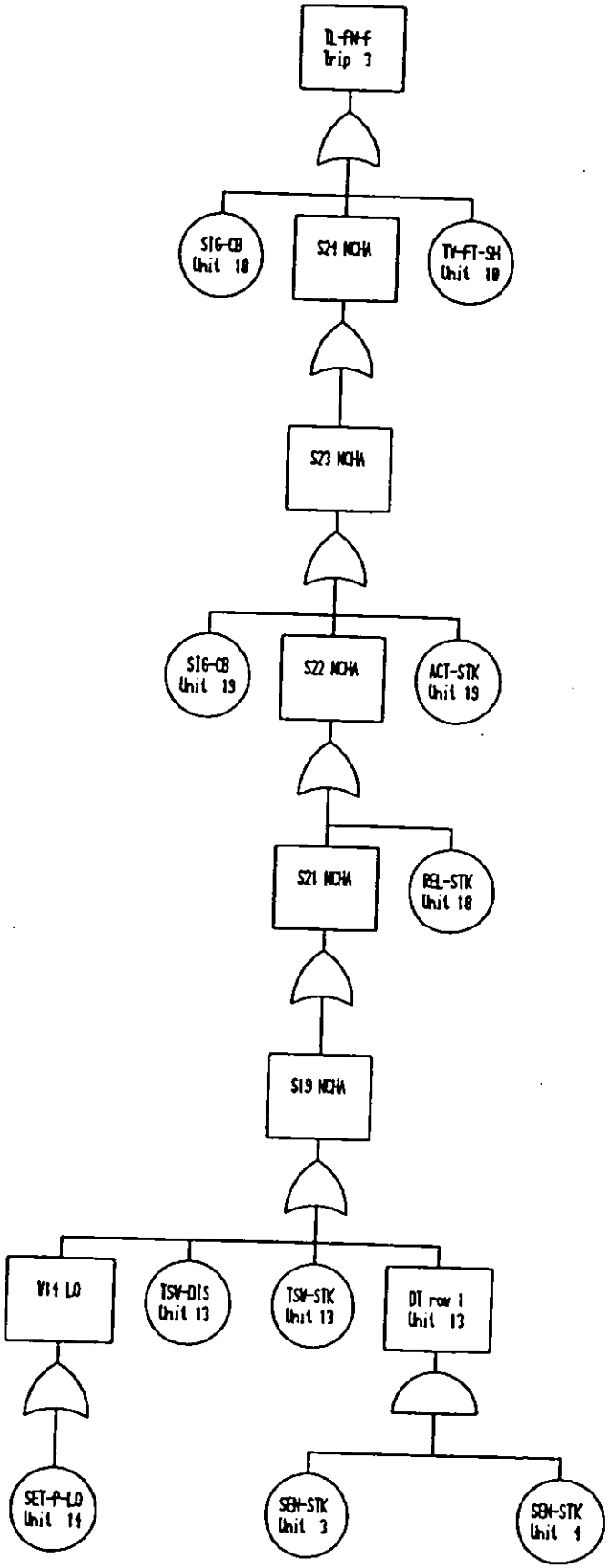

Figure 7.19 - functional failure sub-tree for the third trip system of Figure 7.16

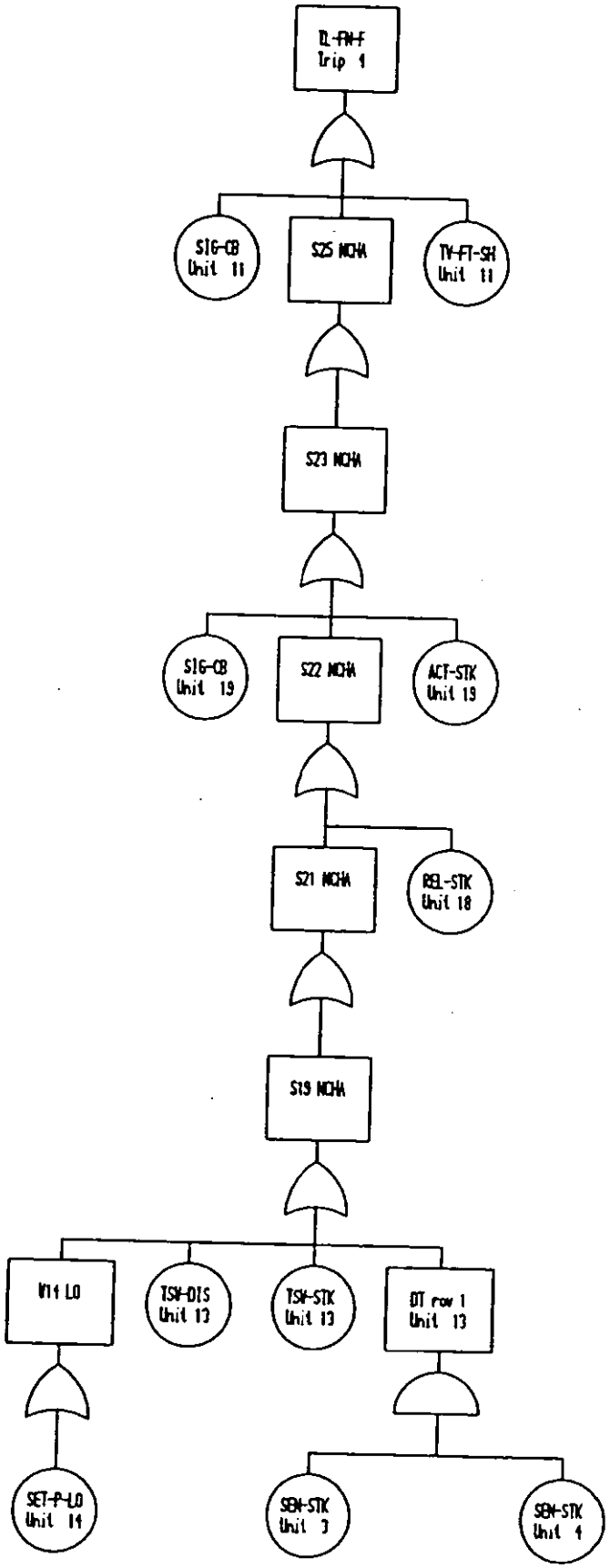

Figure 7.20 - functional failure sub-tree for the fourth trip system of Figure 7.16

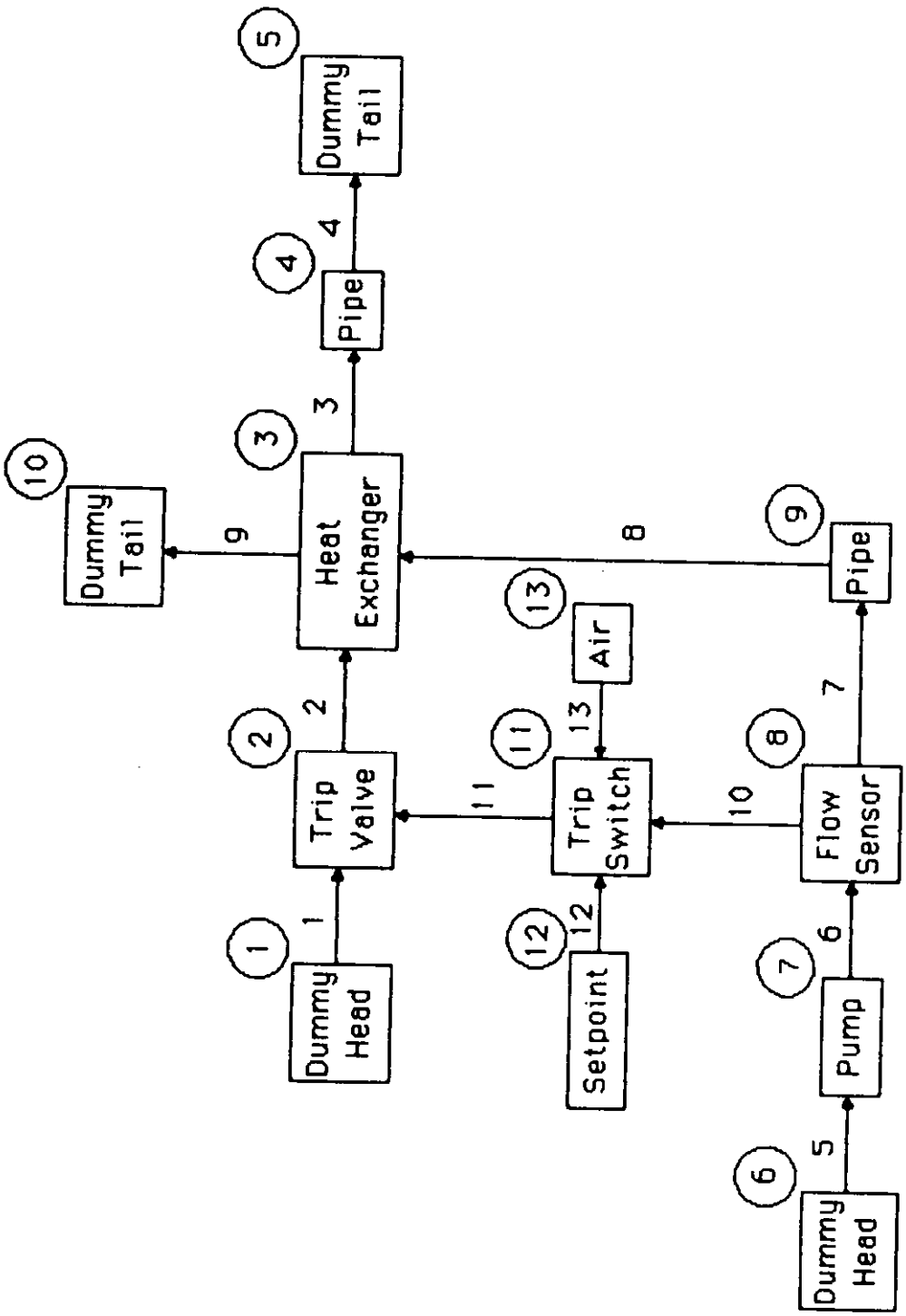

Figure 7.21 - configuration diagram for a temperature protection system, after Lapp and Powers [23]

Feedforward Trip System Example

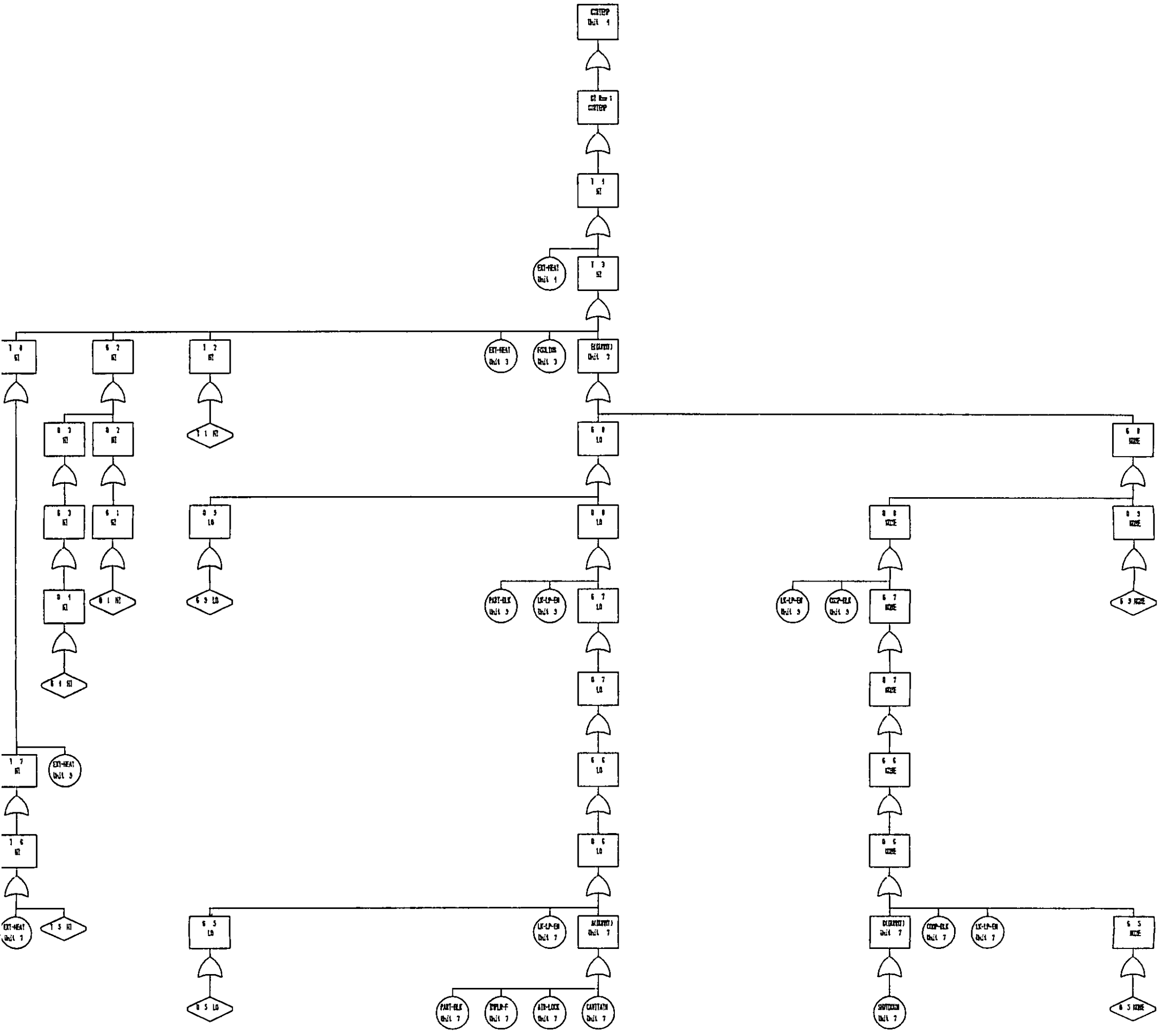

Figure 7.22 - main sub-tree for the
system shown in Figure 7.21

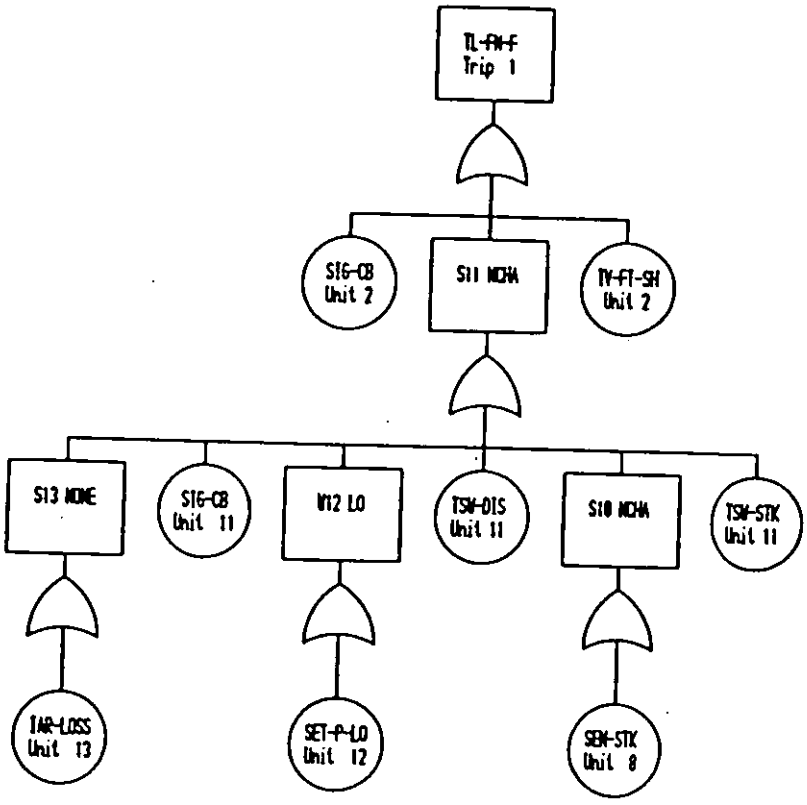

Figure 7.23 - the functional failure sub-tree for the system shown in Figure 7.21

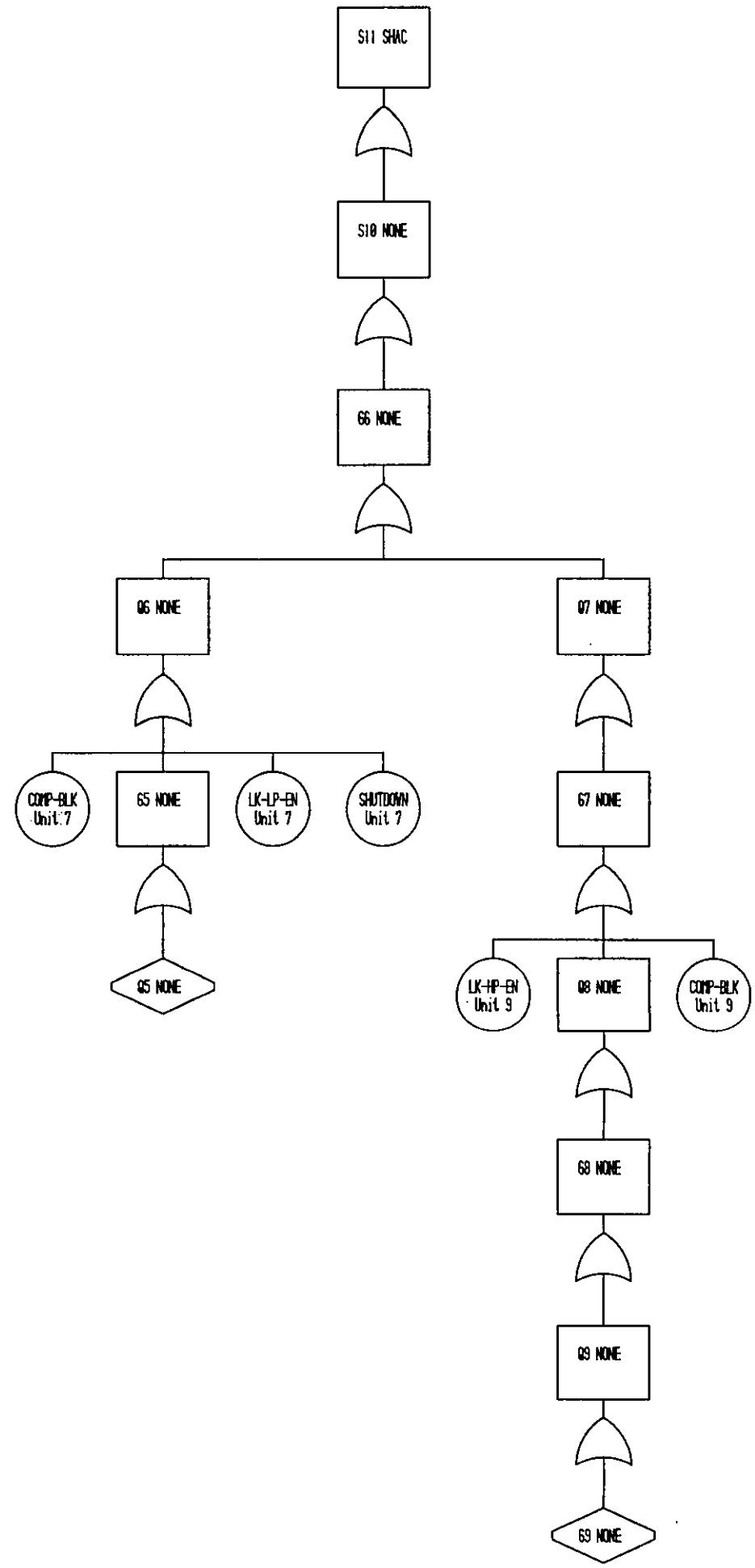

Figure 7.24 - demand sub-tree for the system shown in Figure 7.21

Feedforward Trip System Example

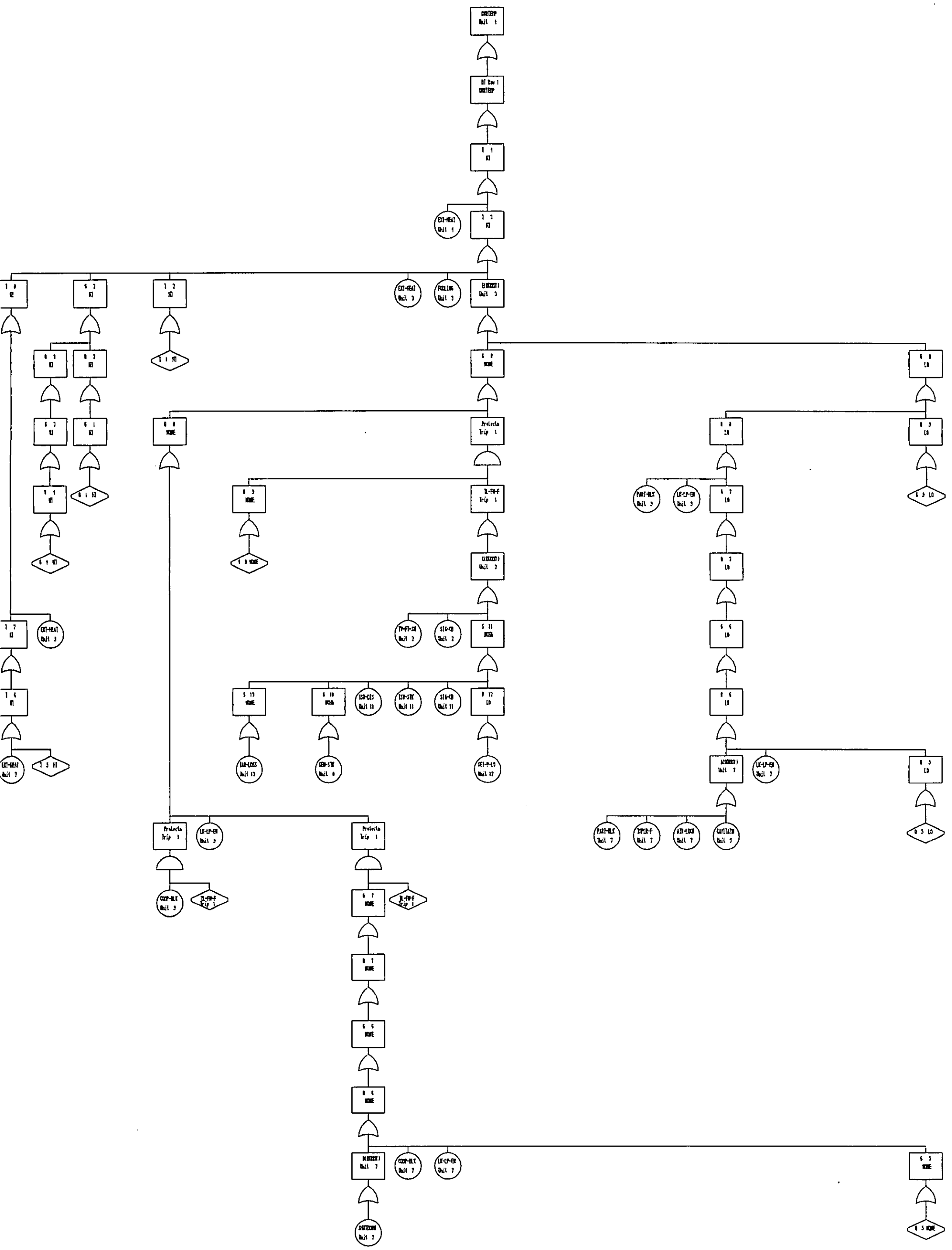

Figure 7.25 - complete fault tree for the system shown in Figure 7.21

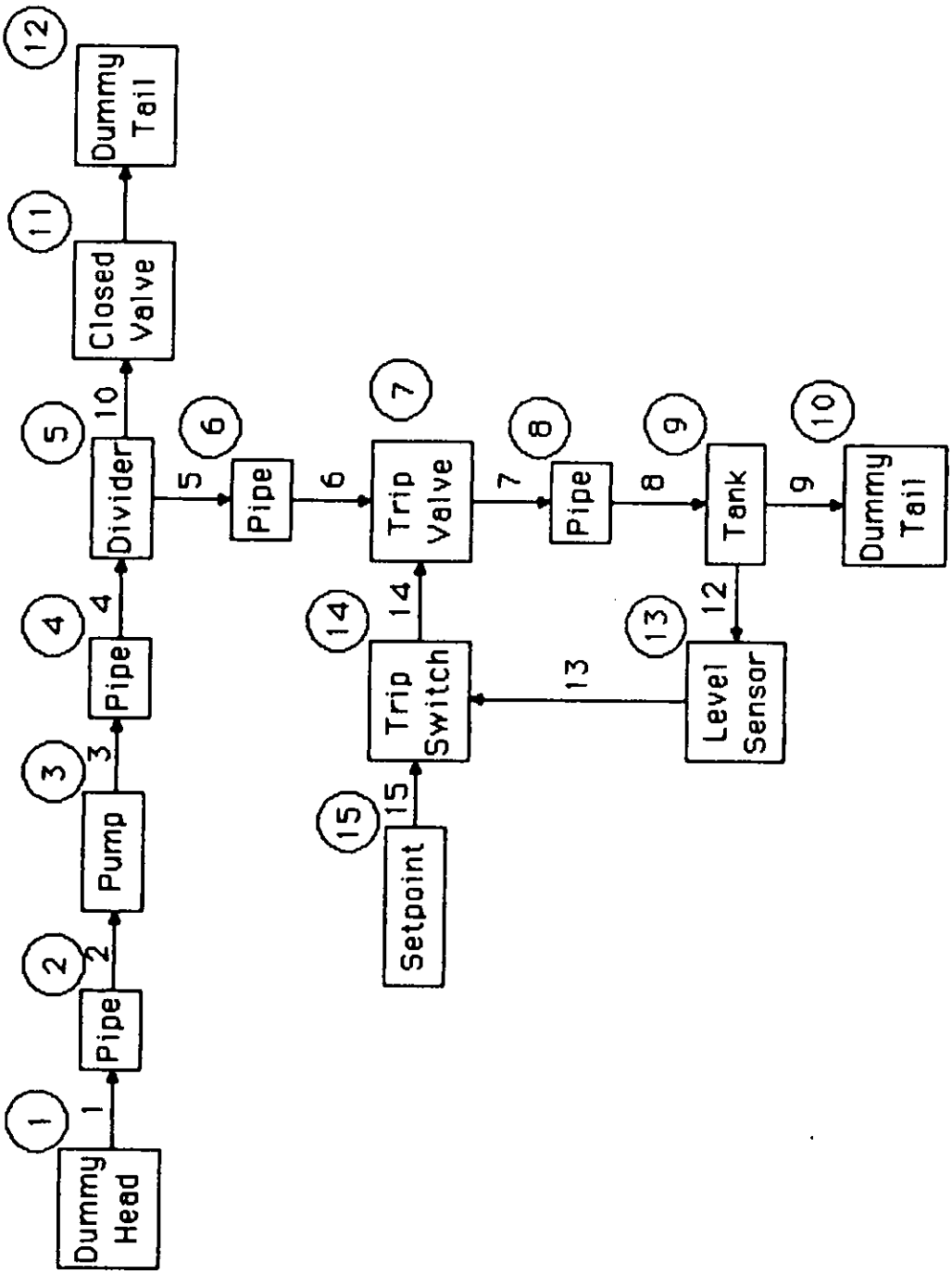

Figure 7.26 - configuration diagram for a level protection system, after Lawley[51]

Level Protection System

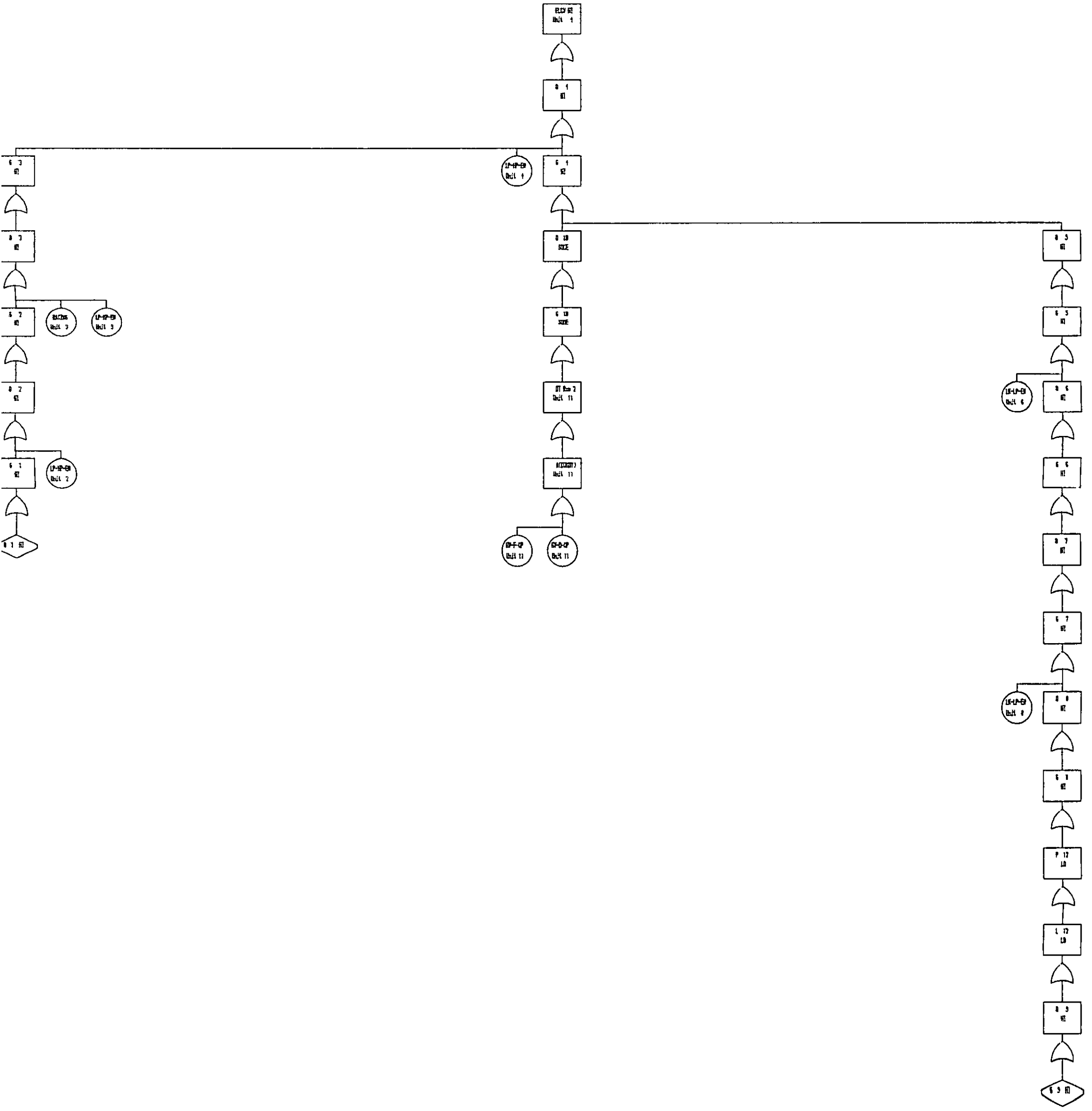

Figure 7.27 - main sub-tree for the
system shown in Figure 7.26

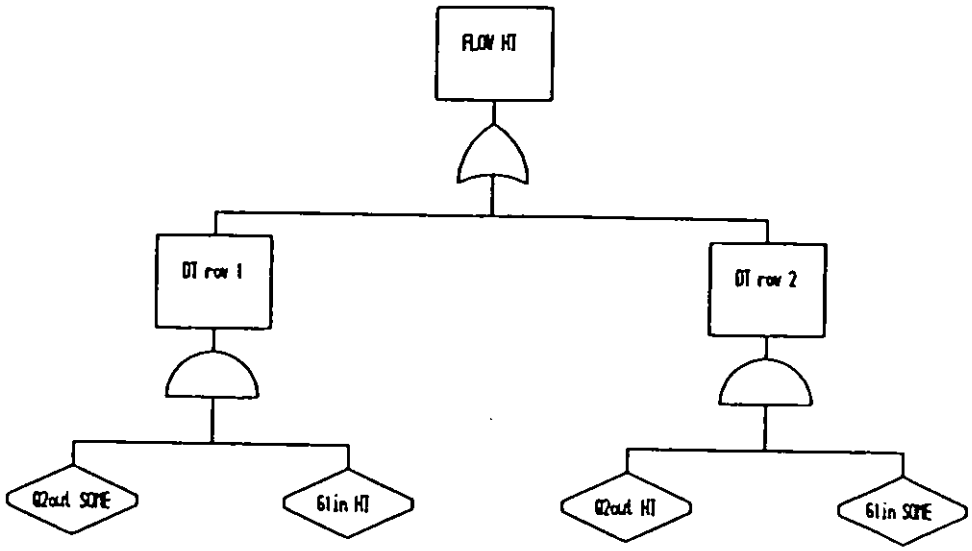

Figure 7.28 - a top event model for high flow, designed to overcome a problem with trip systems

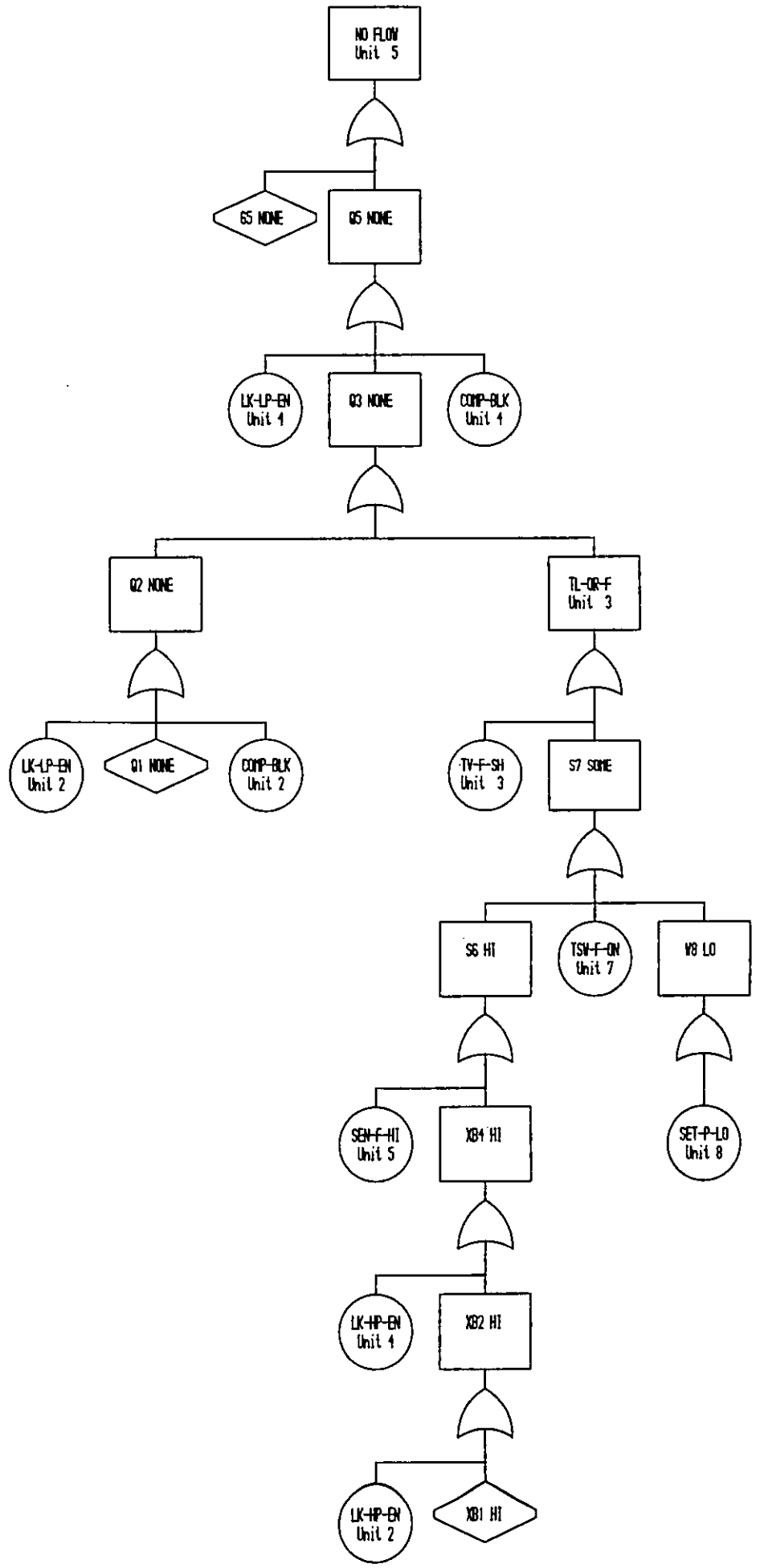

Figure 7.29 - complete fault tree for the system shown in Figure 7.1

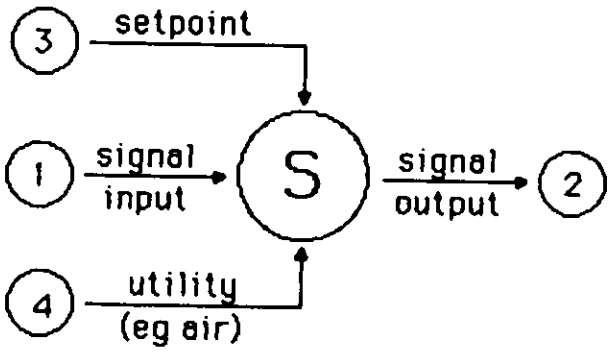

Figure 7.30 - the representation of a trip switch, designed to emit a signal when the input is high

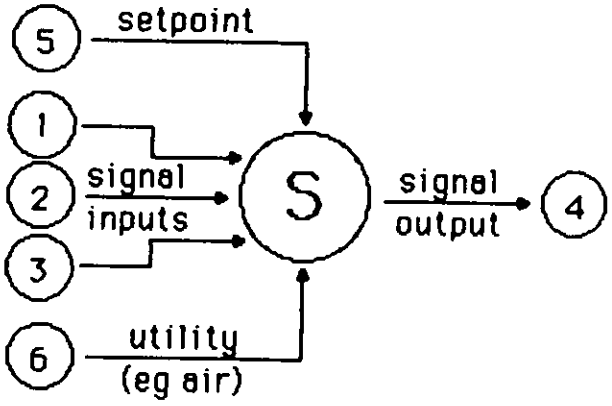

Figure 7.31 - the representation of a trip switch, designed to emit a signal when two of the three inputs are high

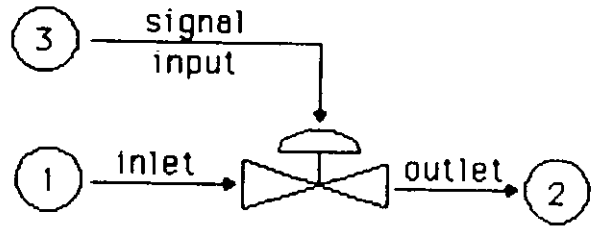

Figure 7.32 - the representation of
a normally-open trip valve

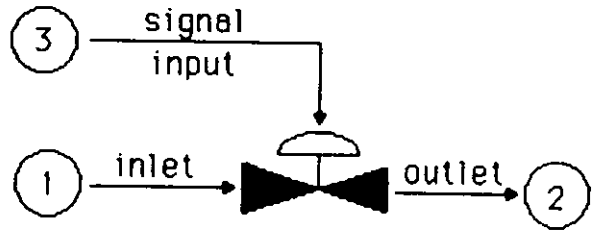

Figure 7.33 - the representation of a
normally-closed trip valve

8) Secondary Failures

Secondary failures are a concept described by Haasl [9]. They are failure mechanisms that induce a particular component to fail. An example of a secondary failure in a chemical plant is low temperature in a pipe, causing blockage by freezing.

8.1) The Problem

The modelling of secondary failures is not a problem in itself. The model for a pipe can be extended to indicate that low temperature has the same effects as blockage. Such a technique is not, however, desirable, because the models then become less general. Obviously, not all pipes are liable to freezing. Furthermore, some secondary failures may have different effects, depending on the specific situation. For example, high concentration of impurity may cause a reaction that could be exothermic or endothermic, explosive, result in solids formation, or corrode the pipe.

8.2) The Approach of Others

None of the other published methodologies take explicit note of secondary failures. The only solution that can be used is to model the units to incorporate any secondary failures considered to be important.

8.3) A Solution

There are two aspects to the modelling of secondary failures. One is the effects of a secondary failure, and the other is the causes of the secondary failure. Considering again the example of freezing, the effect of freezing is blockage; the cause is low temperature. Generally, the cause of a specific secondary failure is fixed. However, the effects may change depending on the particuar plant under study. The solution introduced reflects this behaviour.

8.3.1) The Causes of Secondary Failures

The causes of secondary failures are modelled in a way similar to the method used to model units and top events (see Section 3.2). The important point to note is that this modelling is done independently of the configuration. The causes of a secondary failure are modelled using event statements and/or decision tables. This information is processed to produce minitrees for the causes of the secondary failure. The processing employed is identical to the processing accorded when modelling units, and is described in Section 3.2.3.2 for event statements, and Section 3.2.3.3 for decision tables.

The formats of the event statements and decision tables are slightly different to the formats used for unit modelling. The input information comprises variable deviations and/or basic faults. Intermediate events can be used if desired. The output events can only be intermediate events, or the secondary failure itself. When using variable deviations, the variable used must be a variable that propagates out of a model.

Therefore, the variable T must occur at either an outlet port (e.g. T2out) or a vessel port (e.g. T6ves). For flow variables, either G1in or Q2out may be used. G2out and Q1in, being variables that propagate into a model, are not suitable. This restriction on variables also applies to the modelling of top events (see Section 3.2.5), and is made for the same reason, namely to ensure that a minitree from the correct model is used to model the causes of the secondary event.

A suitable model for freezing, incorporating only decision tables is

```
V T2out LO V G1in SOME V Q2out SOME T FREEZING
V U1in LO V Q2out REV T FREEZING
```

This information states that there are two causes of freezing in a particular unit. One cause is a low temperature upstream (T2out LO), combined with flow in the normal direction. The second cause is low temperature downstream (U1in LO), combined with reverse flow. Note the two term expression for some flow in the first decision table. This is to start the fault tree branch correctly, as described in Section 4.1.1.

Care must be taken when using secondary failure models to ensure that the port numbers in the secondary failure model correspond to the port numbers in the unit models where the secondary failure may occur. Units with multiple ports, such as heat exchangers and vessels may well require special secondary failures with different port numbers.

8.3.2) The Effects of Secondary Failures

The effects of secondary failures can be expressed in one of two forms. One possibility is to relate the effects of a secondary failure in terms of basic faults. An example of this is that freezing causes a partial blockage. The alternative is to give the effects in terms of variable deviations. So, for example, freezing causes low inlet flow, low outlet flow, low outlet pressure and low inlet relief, which are the effects of partial blockage. Some secondary failures are more economically expressed in one form, and some in the other.

It is not possible to have the effects of a single secondary failure expressed in both these forms. A secondary failure expressed in terms of variable deviations is called a 'Physical and Phase Change'. If a secondary failure is expressed in terms of basic faults, then it is called a 'Materials Failure'. This distinction is purely cosmetic. Simply by giving the effects of a secondary failure in one form or another categorises it as either a physical and phase change, or as a materials failure.

Care must be taken to ensure that all the possible effects are specified. If the effect is, for example, high temperature, then there are two variable deviation effects, namely T HI and U HI. Note that neither port numbers nor port types appear in these variable deviations. This information is not required, since the information is used in a plant context rather than a unit modelling context.

The other information required to indicate the effects of secondary failures is the locations which

are susceptible to the failures. In a large plant, typically only a few locations carry liquid that is liable to freezing.

8.3.3) Sample Application

Fig 8.1 is the configuration diagram for a system comprising two pipes and a pump. This plant section will be used in two simple examples.

8.3.3.1) Variable Deviation Effects

This is an example of a physical and phase change.

The secondary failure that affects this system is that high concentration of impurity will lead to an exothermic reaction, resulting in a high pipeline temperature. The cause of high impurity is X2out HI. The effect of the secondary failure is T HI in connections 1 to 4 inclusive. In this example reverse flow effects will be ignored, and so the Y and U variables that could appear in the causes and effects of the secondary failure are omitted. The fault tree for high pipeline temperature is given in Fig 8.2. There are two types of cause in this tree. The first is the primary failures, namely high inlet temperature (T1 HI) and external heat sources (EXT-HEAT Units 1, 2 and 3). The second is the secondary failures, that is high inlet composition of impurity (X1 HI).

A secondary failure called DILUTION is used to represent this secondary failure. The name arises

because the first impurity examined in the development of the method was water, and the name DILUTION was used to reflect this. Nevertheless, the model is suitable for impurities other than water, and so it is used in this example.

Note that the secondary failure appears only once in the fault tree, as a cause of the event T3 HI. Although the secondary failure is also a cause of T2 HI and T1 HI, the secondary failure is not examined at this point because it has already been considered. The effects of a secondary failure are not generally strongly location dependent, and it is assumed that investigating the effects of the secondary failure once per fault tree branch will give the correct results.

8.3.3.2) Basic Fault Effects

This is an example of a materials failure.

The configuration diagram of Fig 8.1 will be used again, this time to illustrate a situation where a secondary failure has effects defined in terms of basic faults. The secondary failure to which this section is subject is blockage by freezing. The cause of the secondary failure is T2out LO. The effect of the secondary failure is PART-BLK in Units 2, 3 and 4. As in the above example, reverse flow effects will be ignored. The fault tree for the top event low outlet flow is given in Fig 8.3. Note the similarity between this fault tree and the one given earlier. The only difference is that, in this case, the secondary failure is associated with a basic fault, rather than a variable deviation.

8.3.4) Multiple Secondary Failures

The synthesis package can model adequately systems which involve several secondary failures, including situations where there are some independent secondary failures of the same type. As with complex control loops (see Section 6.3.3), the solution is to define each different secondary failure individually.

8.3.4.1) Same Secondary Failure at Different Locations

Consider the heat exchanger system based on the Lapp-Powers Nitric Acid Cooler [23], shown in Fig 8.4. Streams 1 to 4 carry nitric acid, and streams 5 to 8 carry water. Mixing these two components will result in an exothermic reaction, and hence in a high temperature. There are, however, two distinct secondary failures in this example. One is nitric acid in the water line, and the other is water in the nitric acid line. Both these failures can be modelled using the variable X without subscripts, if it is noted that the component represented by X changes to take account of the location. X HI in the nitric acid line represents some water in the line. X HI in the water line represents some nitric acid in the water line.

Although these two secondary failures have the same cause (high composition of impurity) and the same effect (high temperature), they are different because they occur in completely independent locations.

Two separate secondary failure effects must be included in the configuration. The first states the high impurity in connections 1 to 4 will cause high temperature. The second states that high impurity in

connections 5 to 9 will also cause high temperature.

The fault tree for this system, again with reverse flow effects ignored, for the top event HI TEMP Unit 4 is given in Fig 8.5. Note that each secondary failure appears once in the fault tree. If only one secondary failure was defined to the package, with effects in all the flow streams, then only one examination of the causes of the secondary failure would appear in the fault tree. However, since the secondary failure in the water line is defined as a different failure from the failure in the nitric acid line, the fault tree examines the causes of high impurity in each stream.

8.3.4.2) Different Secondary Failures at Same Location

The other possibility for multiple secondary failures is that two different secondary failures could have the same effects at the same locations. An example of this will be illustrated using the configuration diagram of Fig 8.1. Assume that the material in the pipeline will freeze if the temperature falls, but will polymerise if the temperature rises. The effect of each of these failures is a blockage in the pipeline.

These two secondary failures are each defined as having effects PART-BLK in units 2, 3 and 4, and are called FREEZING and POLYMERI (names are restricted to eight characters). The fault tree for this system for the top event FLOW LO Unit 4 is displayed in Fig 8.6, and illustrates that both low and high temperature may cause the top event.

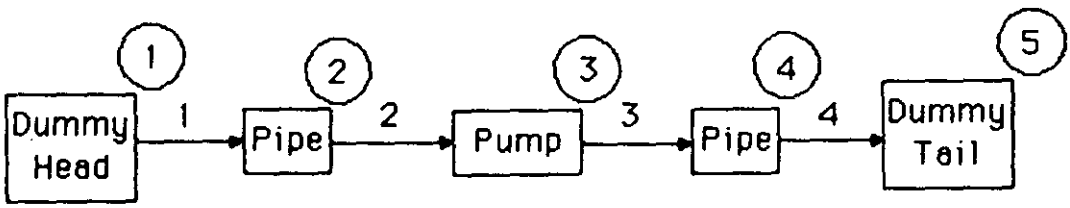

Figure 8.1 - configuration diagram for a simple pipeline system

Secondary Failures Example

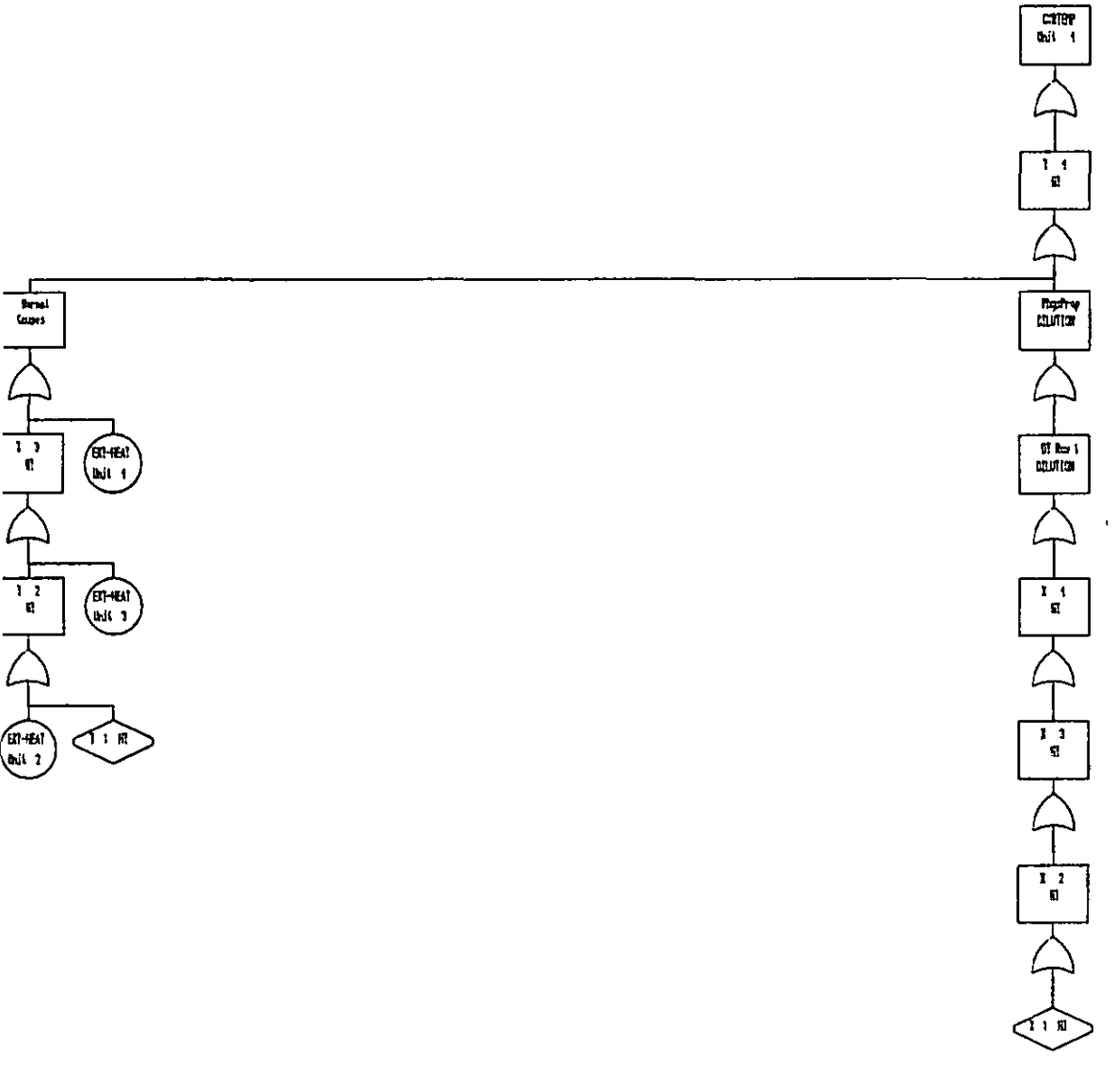

Figure 8.2 - complete fault tree for the system shown in Figure 8.1

Secondary Failures Example

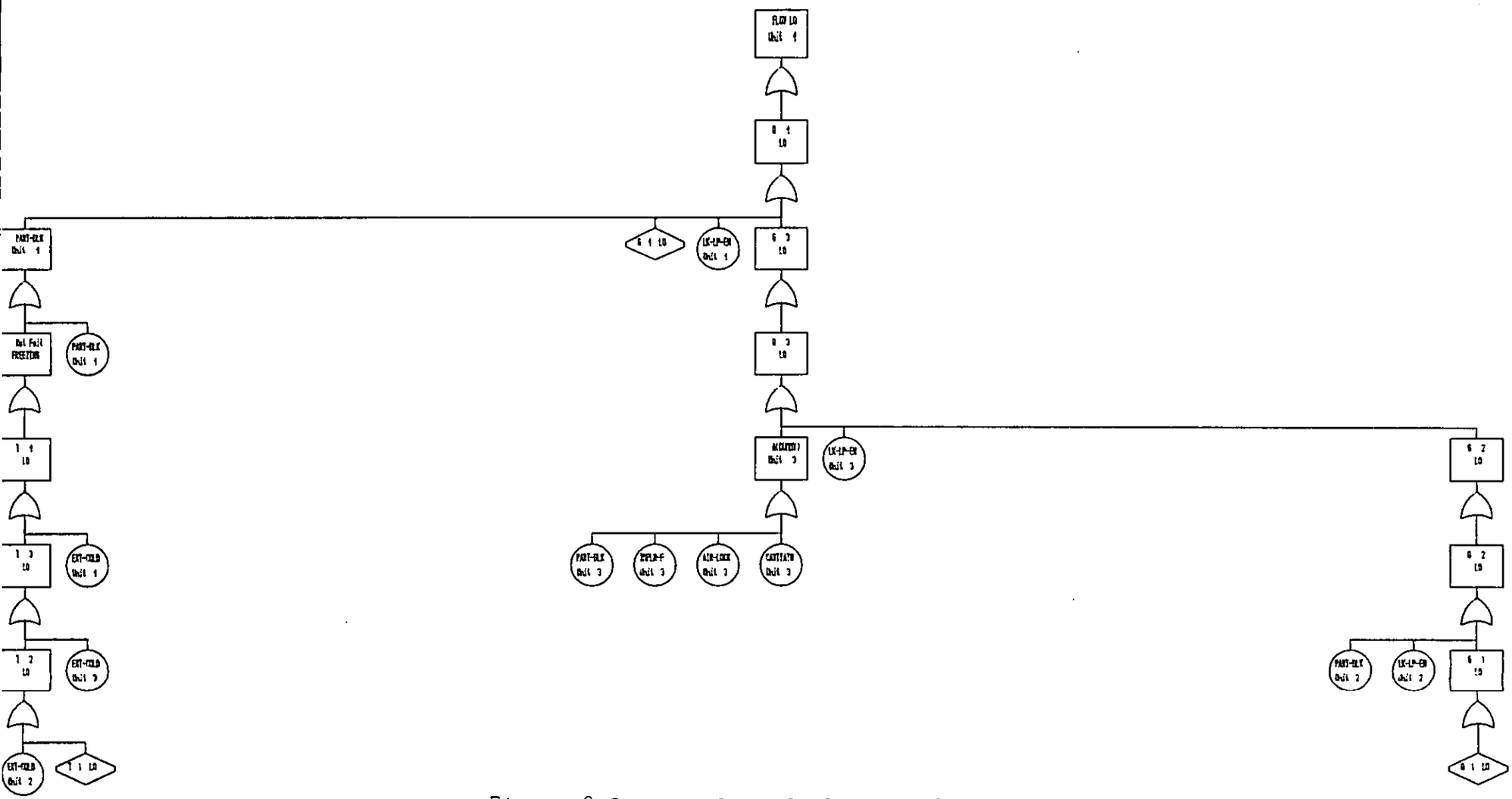

Figure 8.3 - complete fault tree for
the system shown in Figure 8.1

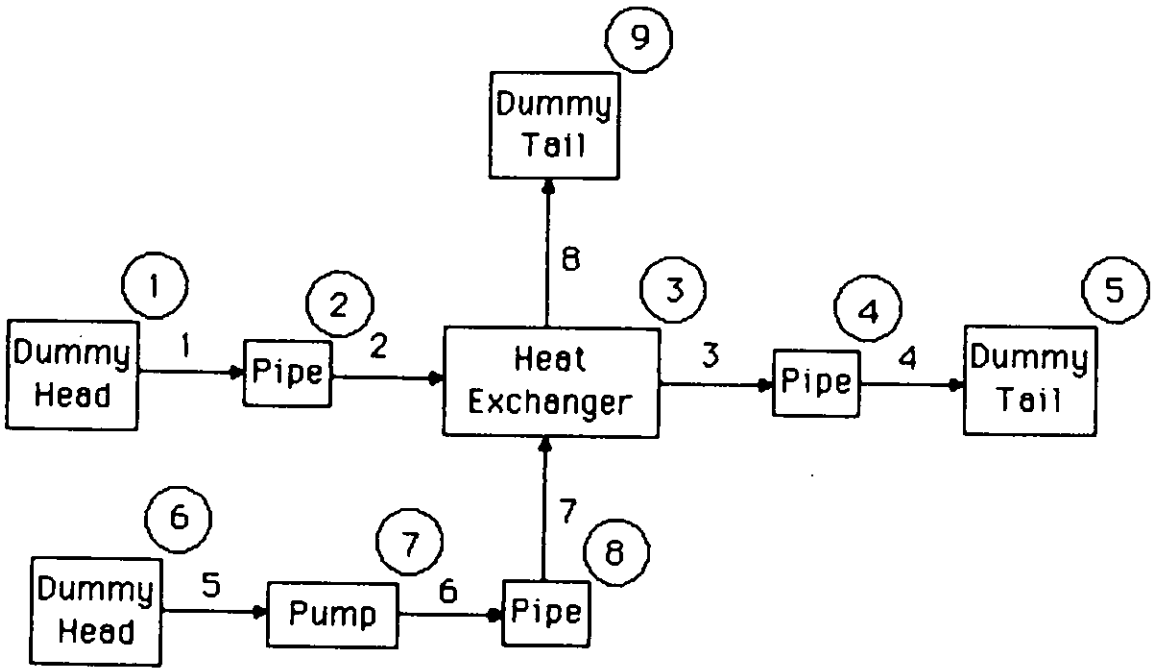

Figure 8.4 - configuration diagram for a heat exchange system, after Lapp & Powers [23]

Heat Exchanger Example

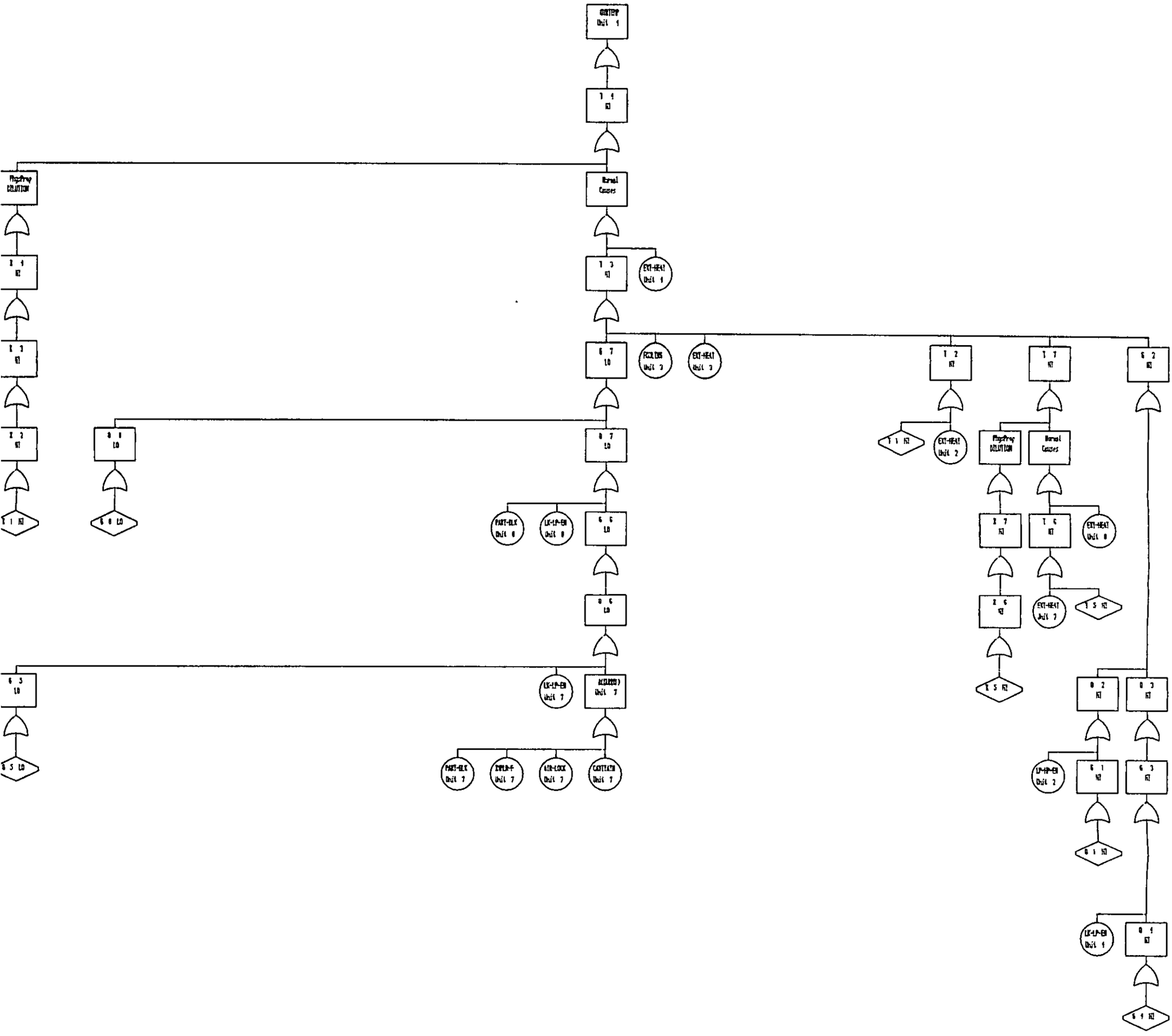

Figure 8.5 - complete fault tree for the . . . .
system shown in Figure 8.4

Secondary Failures Example

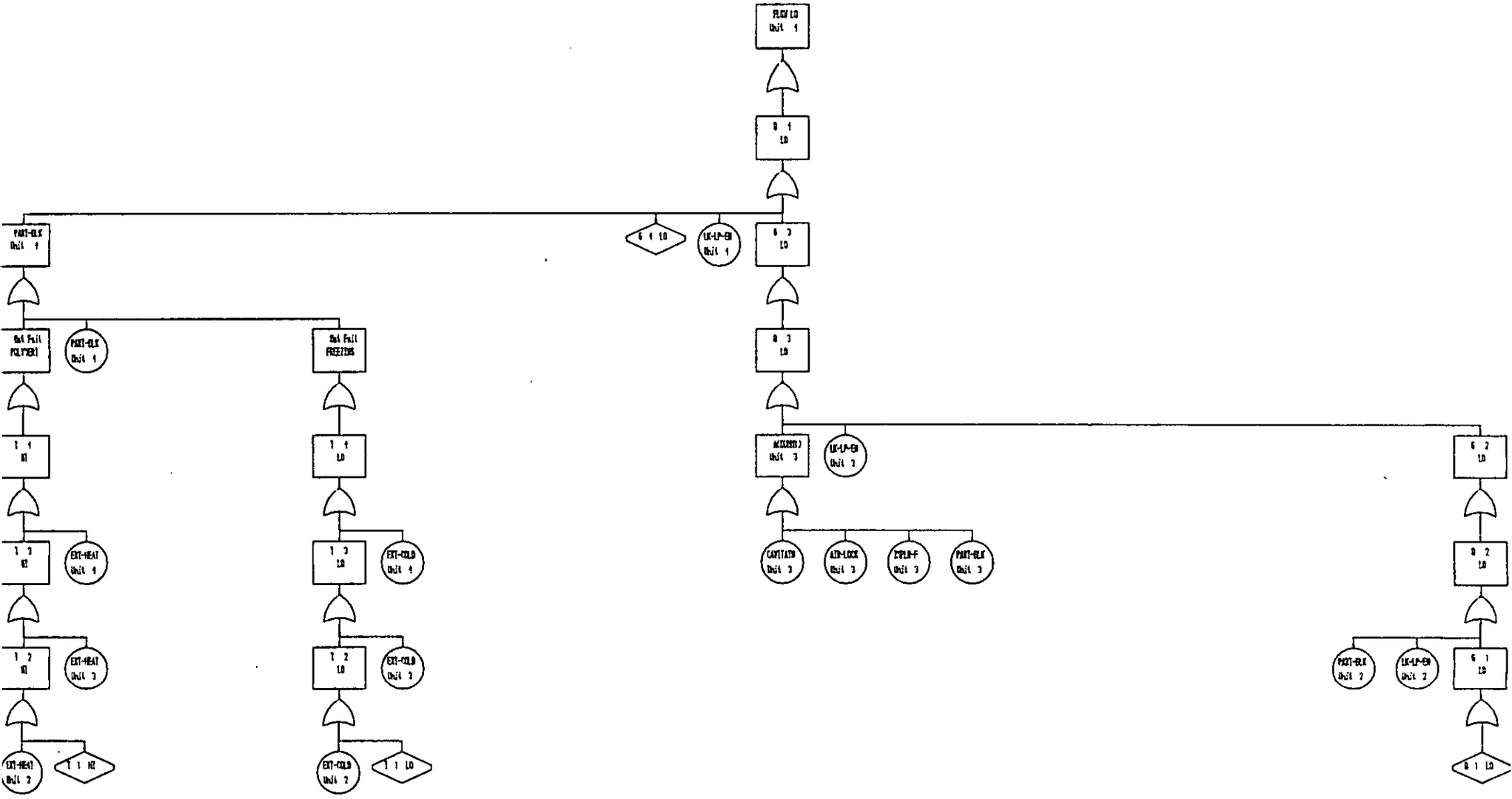

Figure 8.6 - complete fault tree for the system shown in Figure 8.1

9) Divider-Header Combinations

Divider-header combinations are groups of units that provide parallel flow paths from a divider, or process stream splitter, to a header, or process stream mixer. There are three types of divider-header combination, each of which is treated slightly differently. The types are

I Bypass normally has no flow. Bypasses on control valves are the most common systems of this type.

II Bypass normally has some flow. A bypass around a heat exchanger, where the flows are manipulated to obtain a desired temperature, is an example of this type.

III Parallel system, where numerous legs must be operational for the desired throughput to exist. Pump banks, possibly with one or more pumps on standby, are the standard system of this type.

Plant sections that incorporate dividers and headers in combination, like control loops and trip systems, cannot satifactorily be modelled using component models alone.

9.1) The Problem

There are three reasons why divider-header combinations require special treatment. Firstly, for combinations of type I and II, the continuity of flow imposes extra restrictions on the events that can occur in a fault tree over and above the restrictions that normally apply. Special treatment is required to ensure that these extra restrictions are not violated. For example, low flow at the combination outlet cannot be caused by reverse flow at the combination inlet. This condition, of course, applies to single units as much as to divider-header combinations. However, in modelling single units it is easy to create models that avoid such inconsistencies. Unfortunately, it is not always possible to avoid such problems when dealing with divider-header combinations.

Consider the plant section shown in Fig 9.1. The bypass is provided so that online maintenance of the control valve can be carried out. Normally there is no flow in the bypass. Consider the top event "low flow out of combination". One possible cause of this is reverse flow down the bypass line. As the partial fault tree of Fig 9.2 shows, this fault is propagated to reverse flow at the combination inlet. One cause of low flow out of the combination is, therefore, bypass valve open AND reverse flow at the combination inlet. However, reverse flow at the combination inlet is not a valid cause of low flow at the combination outlet.

The second reason why divider-header combinations require special treatment, applying to combination types II and III, is that simple unit modelling cannot cope adequately with faults that relate to low flow through the combination. Suppose that a pump bank

comprises three pumps, all running, and each capable of supplying 50% of the desired throughput. Then, for low flow through the combination to result, at least two of these pumps must have restricted flow (low or none) through them. However, a leak in any one of these pumps may be sufficient to cause low flow through the combination, since the other pumps will be unable to compensate for the loss of fluid.

The third reason for a special treatment of divider-header combinations is the wide range of possible parallel systems, such as one out of two, two out of three, three out of five, and so on. Each of these possibilites, unless a special treatment is developed, requires different models for the divider and header.

9.2) The Approach of Others

No other researchers in the field have examined these problems.

9.3) A Solution

The solution adopted in the package described in this thesis is to use the approach that was used to solve the problems associated with control loops. A general model, which describes the behaviour of divider-header combinations, is used in the fault tree to set the synthesis process on the right track. This general model is filled in during the synthesis process using configuration information about the combinations in the process, so that the correct faults appear in the correct locations in the tree, and that no inconsistencies exist.

Two general models for divider-header combinations exist. These are shown in Figs 9.3 and 9.4. The model displayed in Fig 9.4 is used only when the event under study is low flow through a combination of type II or III. In all other cases, the general model shown in Fig 9.3 is used. These models apply only to situations where the event under study originates outside the combination. In cases where the event under study originates inside the combination, a different approach is used. This is covered later (Section 9.3.3).

Both models classify failures into two groups, failures that occur inside the combination, or internal faults, and failures that occur outside the combination, or external faults. Each branch in the general models comprises faults that belong to only one of these groups. During synthesis of these branches, any cause that is found to belong to the other group is removed from the branch currently being synthesised.

9.3.1) The Normal Modelling Approach

The synthesis process can now be illustrated, using the control valve bypass example considered earlier (see Figs 9.1 and 9.2). Using the general model of Fig 9.3 for this combination, the fault tree synthesised is given in Fig 9.5. Note that there are entries in the INTERNAL and EXTERNAL branches of the general model, but that the ENABLING branch is missing. The INTERNAL and EXTERNAL branches are self-explanatory. The ENABLING branch contains faults that must occur within the combination for faults to propagate through the combination. An example is that a non-return valve inside a combination must fail to prevent reverse flow, before reverse flow can propagate through the leg that contains the non-return valve. In the example considered here, there are no such faults, and so the ENABLING branch is missing. An example that involves enabling faults is given later (Section 9.3.4).

9.3.2) Low Flow in Combination Types II and III

The procedure outlined in the previous section is applicable to most situations involving divider-header combinations. However, when there is more than one combination leg that normally has some flow, and the event of interest is low flow through the combination, then the situation is more complicated. As was pointed out in Section 9.1, there are two distinct sub-classes of internal faults for this type of combination. These are failures that are sufficient to cause low flow, and failures that require additional failures in other legs. This distinction is the reason why a different general model is used in such situations.

Consider the configuration shown in Fig 9.6, depicting three pumps in parallel. This is a type III combination. Suppose that each pump is capable of delivering 50% of the desired capacity, and that all are normally working. Therefore, two of the legs must deliver low flow, or there must be a leak of fluid in one of the legs, for low flow through the combination to occur. Generally, it is faults that have similar effects to blockage faults, such as pump shutdown or valve closed, that require additional faults in other legs. For this reason, in the general model, this branch is known as the BLOCKAGE branch. The other branch that contains faults internal to the combination is called the LEAKAGE branch, because the events in this branch have effects similar to the leak faults.

When synthesing a fault tree that includes this general model, it is essential to be able to distinguish between faults that should appear in the LEAKAGE and BLOCKAGE branches. The distinguishing characteristic used is that BLOCKAGE-type faults result in low flow both upstream and downstream, whereas LEAKAGE-type faults result in low flow on one side, but high flow on the other side. Figs 9.7 to 9.9 illustrate how the methodology differentiates between these two types of fault. Fig 9.7 comprises the pipework, appearing in one leg of the combination displayed in Fig 9.6. Fig 9.8 shows two fault trees for low flow. One tree is for low flow into the system, and the other is for low flow out of the system. BLOCKAGE-type faults are those faults that occur in both branches, namely

a) AIR-LOCK (air lock in pump)
b) IMPLR-F (impeller failure)
c) CAVITATN (cavitation in pump)
d) PART-BLK (partial blockage)

The LEAKAGE-type faults are

a) LK-LP-EN (leak to low pressure environment)
b) LK-HP-EN (leak from high pressure environment)

In any particular fault tree, either one of the branches of Fig 9.8 may be the LEAKAGE branch, with the other being the BLOCKAGE branch. Which branch is which depends on whether fault propagation is tracing faults to upstream causes, or to downstream causes. The LEAKAGE branch is set up so that fault propagation in this branch is always in the same direction as fault propagation in the main fault tree. The propagation direction in the BLOCKAGE branch is always in the opposite direction. So, in the example of Fig 9.7, if the fault tree is finding the upstream causes of low flow downstream of the combination, then the LEAKAGE branch is the branch starting at the header, and the BLOCKAGE branch is the branch starting at the divider. After events of one type have been removed from the branch of the other type, the branches are as shown in Fig 9.9.

The example of which events should be included in the LEAKAGE branch, and which in the BLOCKAGE branch, considered above has not considered the effects of no flow. No flow in any of the legs of the pump bank system illustrated in Fig 9.6 will contribute towards a low total flow through the combination. Therefore, to examine completely the possible causes of low flow through such a system, no flow effects must be

considered. As with low flow, the causes of no flow can be split into LEAKAGE and BLOCKAGE type faults, following the procedure outlined above for low flow.

So, returning again to the pump bank depicted in Fig 9.6, the fault tree for the top event low flow out of the combination is illustrated in Fig 9.10. At first sight this tree appears large. The reason for this is that there are three legs to the pump bank, each of which has LEAKAGE and BLOCKAGE type faults for both no flow and low flow through that leg. LEAKAGE faults are grouped under the intermediate events INTERNAL and NO FLOW; the BLOCKAGE faults are grouped under the intermediate events E(DUMMY) and three D(DUMMY) faults. Note that the BLOCKAGE faults in the three legs are grouped together under a 2/3 gate, since two of the legs must have BLOCKAGE-type faults before low flow out of the combination occurs.

9.3.3) Flow Faults Originating within Combinations

This procedure of separating the LEAKAGE-type faults from the BLOCKAGE-type faults applies also to type II combinations. The typical combination of this type will require BLOCKAGE-type faults in both legs but LEAKAGE-type faults in only one leg to cause low flow through the combination. The reason why types II and III combinations differ is that, for type II combinations, faults frequently originate from inside the combination. For example, temperature deviations in the heat exchanger may be caused by flow deviations through the leg of the combination containing the exchanger. The flow deviation therefore originates from inside the combination. It is necessary to have more detailed models for the divider and header units to cover this

situation. Or, more accurately, it is possible to have less complex models when the fault originates outside the combination. The fact that the models are less complex means that more legs can be modelled within the limitation imposed by the amount of computer memory available. The less detailed models of combination type III can handle up to five legs. The more detailed models required by type II mean that only two legs can be handled.

Essentially, the extra information required in the type II models is an indication of how the flow in one leg affects the flow in the other legs. This can be illustrated by a simple example. Consider the configuration shown in Fig 9.11. The function of the heat exchanger is to cool the fluid flowing through the combination. The fault tree for the top event low temperature of this fluid is shown in Fig 9.12. One cause of the top event is low flow of fluid through the heat exchanger leg of the combination.

The causes of low flow through the heat exchanger leg of the combination are unaffected by the general models that were required when the fault originated outside the combination. Nevertheless, there is still a restriction on the events that can occur in the fault tree. This restriction takes the form that a propagation path can pass through either the divider or the header, but not both. The practical effects of this can be seen by considering one cause of low flow through the exchanger, namely high flow through the other leg. Obviously, one cause of this is high flow of fluid through the combination, but this cause is not consistent with low flow through the heat exchanger. The restriction is necessary to prevent the propagation path Q5 LO - Q9 HI - Q8 HI - Q2 HI, which violates none

of the standard restrictions.

Note that in this example, the standard divider-header combination treatment is applied to the temperature deviation that propagates into the combination.

9.3.4) Enabling Faults

Enabling faults are those faults that must occur within the combination to permit the propagation of external faults through the combination. The combination enabling event is built up using the enabling faults of each leg in the combination. Enabling faults normally occur only in fault trees that involve no flow, some flow, or reverse flow. For the deviations given above, if only one leg permits propagation of the flow deviation through the combination, then the fault will propagate through the combination. Hence the leg enabling faults are ORed together to give the combination enabling event. This combination enabling event is the ENABLING branch that appears in the combination general models. Generally, there may be several enabling faults per branch. If there are no enabling faults in a particular branch, then the external fault is certain to propagate through the combination, and the ENABLING branch of the general model is certain to occur. The main leg of the combination contained in Fig 9.1 (control valve bypass example - see Section 9.3.1) has no enabling faults, and so the ENABLING branch is certain to occur. This is reflected in the fault tree for this system (see Fig 9.5).

An enabling fault is identified as a fault that appears in an AND gate that is a cause of the same variable deviation as the event at the top of the combination general model.

Consider the configuration diagram shown in Fig 9.13, comprising a pump bank of two pumps. The top event of interest is reverse flow through the combination. The fault tree for reverse flow through one of the legs is shown in Fig 9.14. For the purposes of modelling, it has been assumed that reverse flow cannot occur through the pump, while the pump is still running. This fault tree contains two AND gates that involve reverse flow. The non-reverse flow causes of these gates are NRV-F-OP Unit 10 (the non-return valve) and SHUTDOWN Unit 9 (the pump). These faults are the leg enabling faults of this leg. Before reverse flow through this leg can occur, both these faults must occur. The two faults are therefore ANDed together to produce the leg enabling fault.

The other combination leg has a similar leg enabling fault. The combination enabling event is simply the result of ORing together the two combination leg enabling faults.

The complete fault tree for the top event reverse flow through the combination, incorporating this combination enabling structure, is given in Fig 9.15. Note that the event SHUTDOWN does not appear in the INTERNAL branch of the fault tree, since there are no faults internal to the combination with which SHUTDOWN should be ANDed. Nevertheless, the synthesis procedure identifies SHUTDOWN as an event that must occur before reverse flow can propagate through the combination. Hence, SHUTDOWN appears in the ENABLING branch.

9.3.5) Nesting of Divider-Header Combinations

Divider-header combinations occasionally occur in tandem with other divider-header combinations. One combination may occur within another combination, or, more unusually, the combinations may overlap. This latter case is shown in Fig 9.16. As with all such similar situations that may arise, for example complex control loops, the basic procedure is to treat each combination as a separate entity. The general model for each combination, if appropriate, is applied individually to each combination.

The example presented in Section 11.4 involves an overlapping divider-header combination.

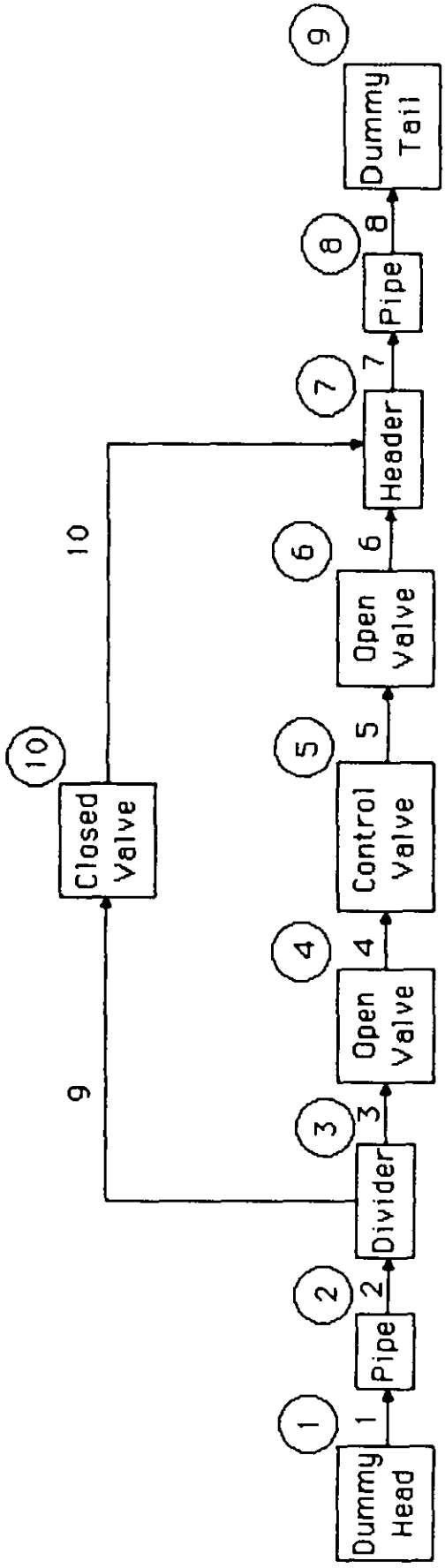

Figure 9.1 - configuration diagram for
a control valve bypass system

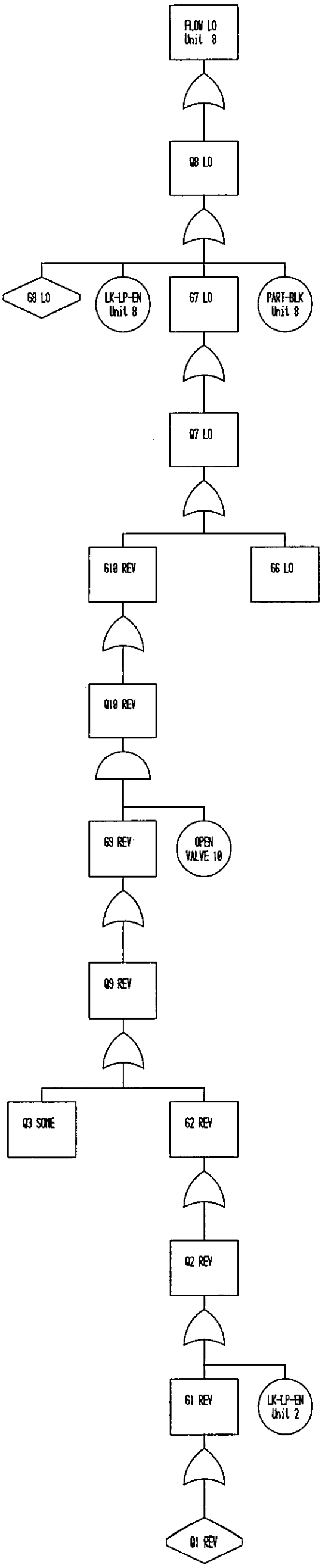

Figure 9.2 - partial fault tree for the system shown in Figure 9.1, without special divider-header combination treatment

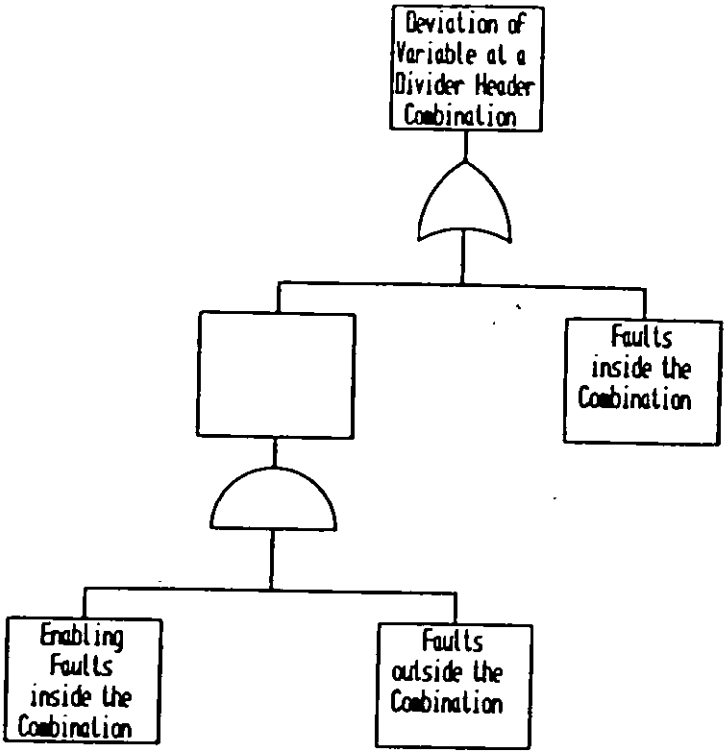

Figure 9.3 - the general model for a divider-header combination

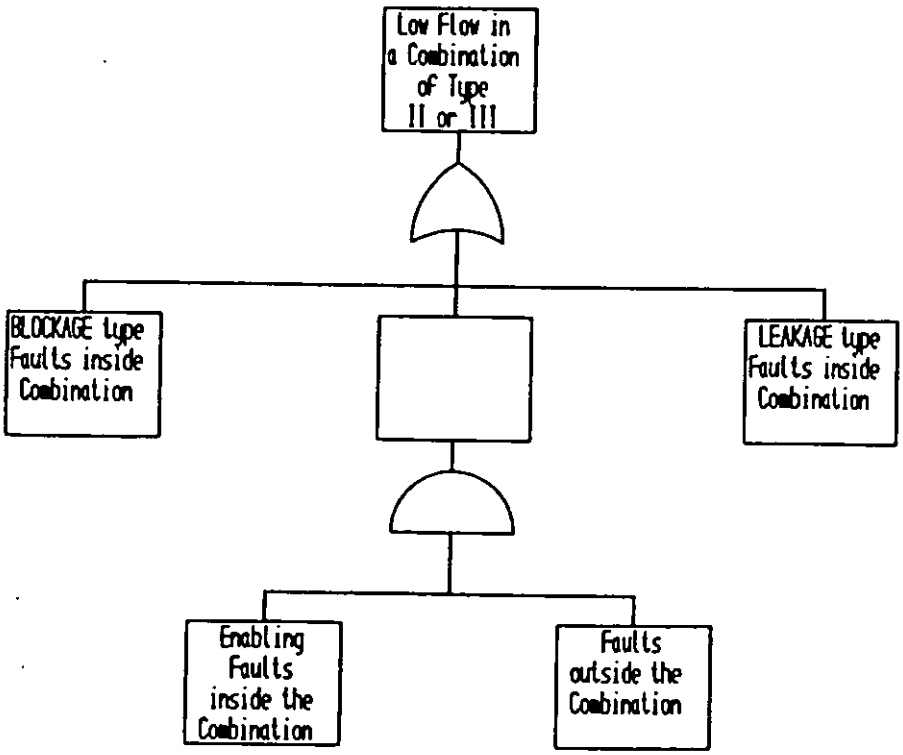

Figure 9.4 - the general model for low flow in a Type II or Type III combination

Control Valve Bypass System

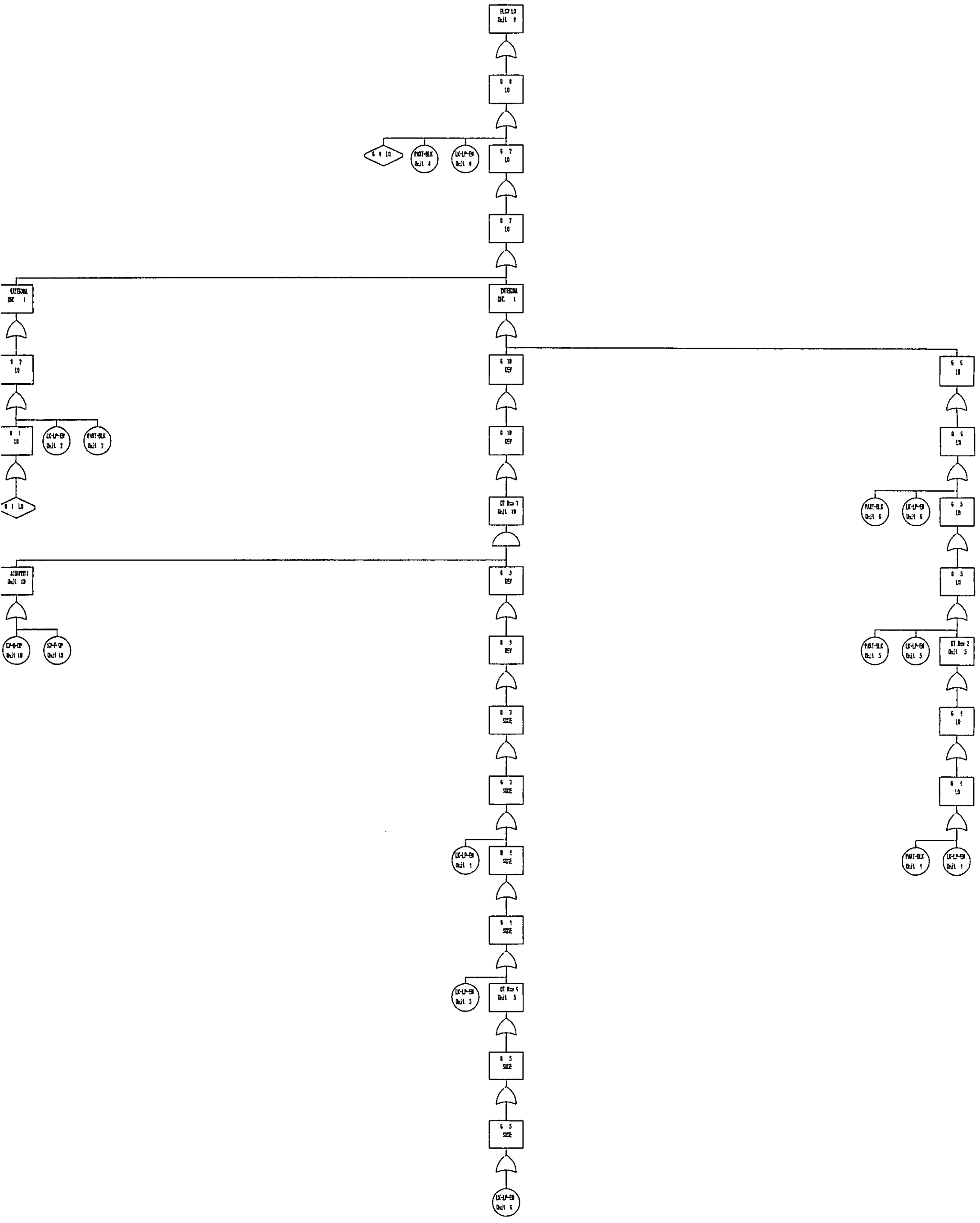

Figure 9.5 - complete fault tree for the system shown in Figure 9.1, with special divider-header combination treatment

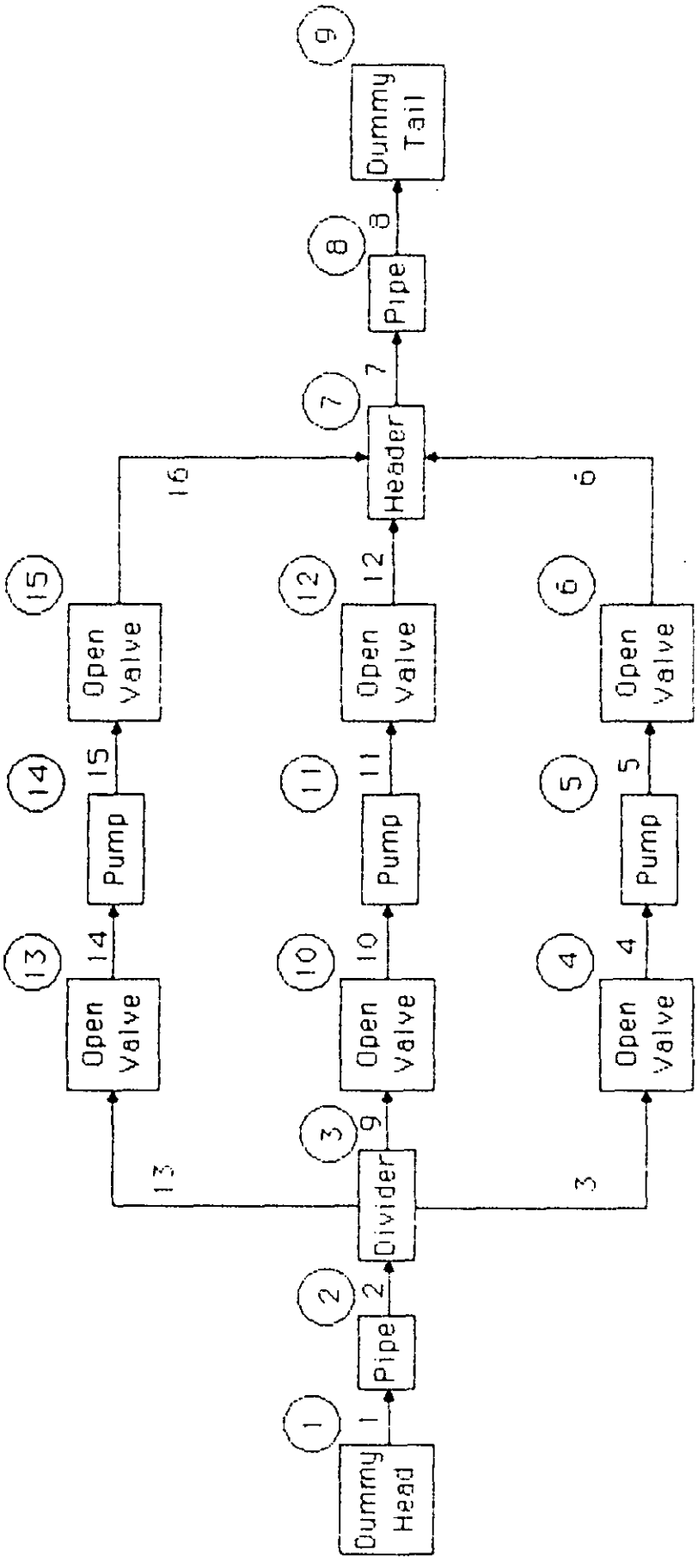

Figure 9.6 - configuration diagram for a two-out-of-three parallel pump system

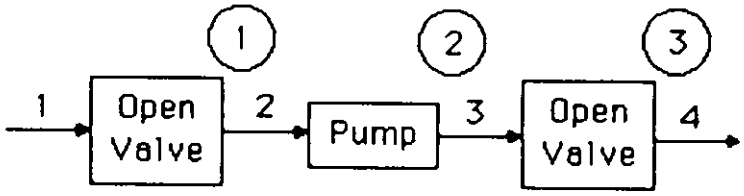

Figure 9.7 - configuration diagram for the pipework in one of the legs of Figure 9.6

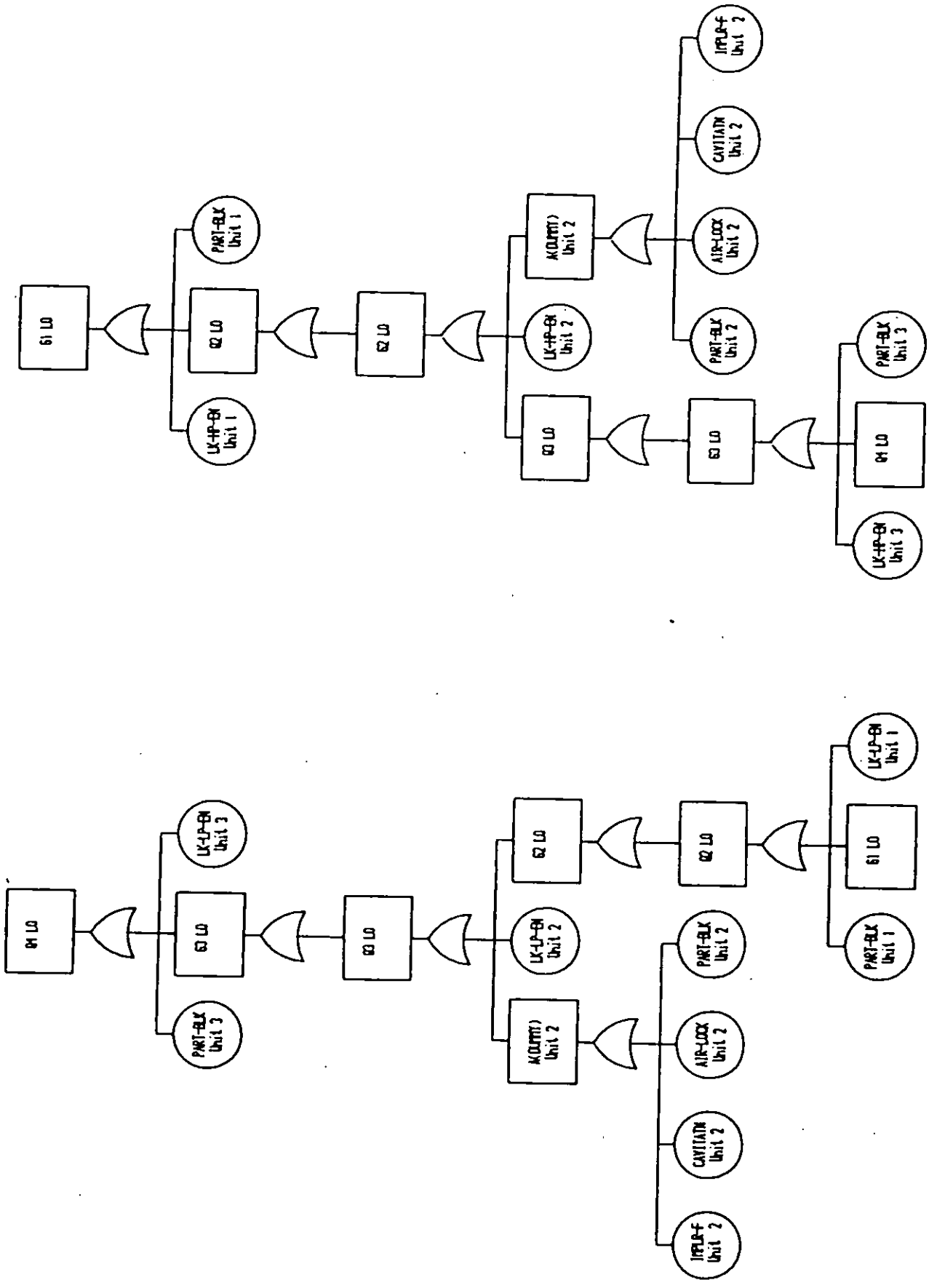

Figure 9.8 - two fault trees for low flow through the system shown in Figure 9.7

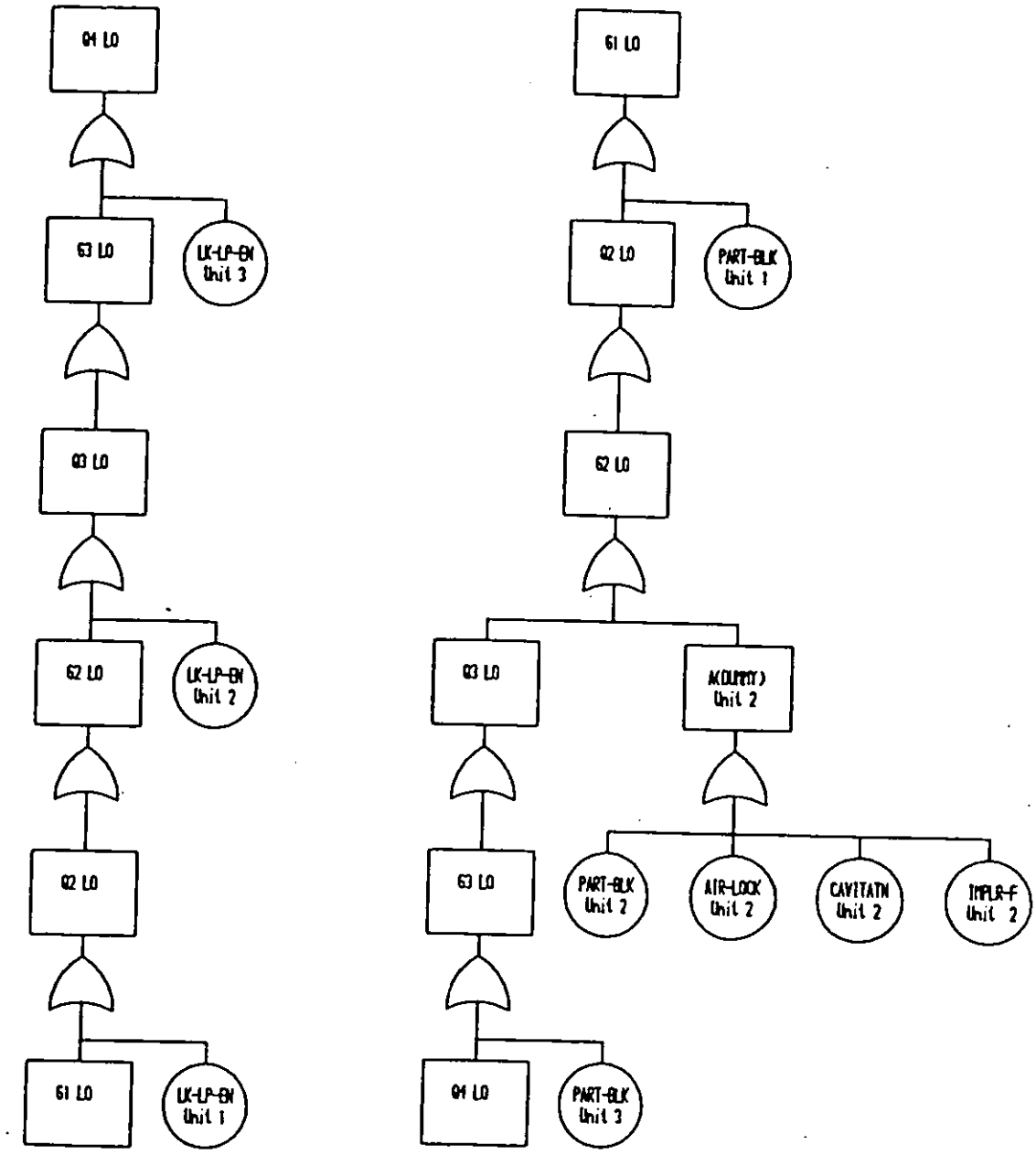

Figure 9.9 - the two fault trees for the LEAKAGE (top event Q4 LO) and BLOCKAGE (top event G1 LO) branches of the system shown in Figure 9.7

2-out-of 3 Pump Bank System

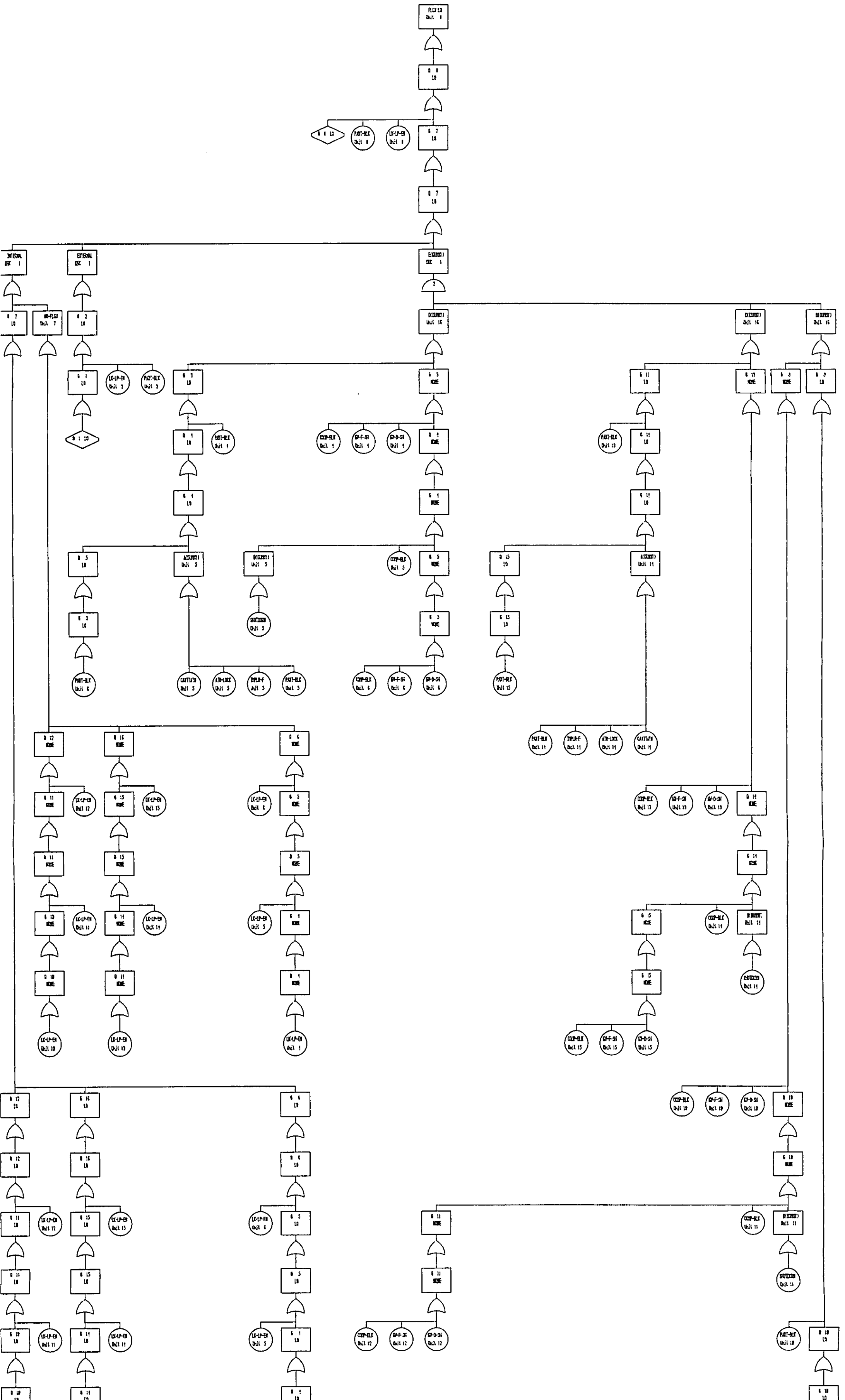

Figure 9.10 - complete fault tree for the system shown in Figure 9.6

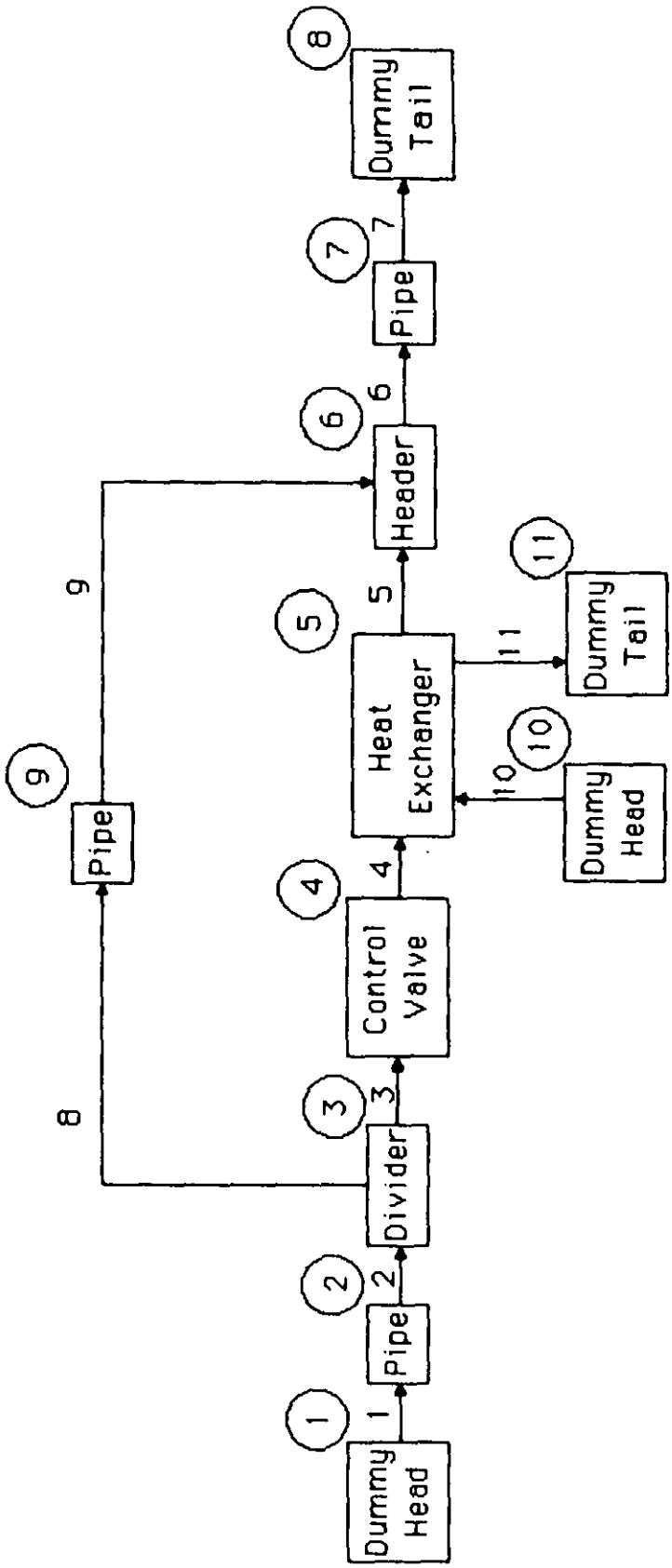

Figure 9.11 - configuration diagram for a heat exchanger bypass system

Heat Exchanger Bypass System

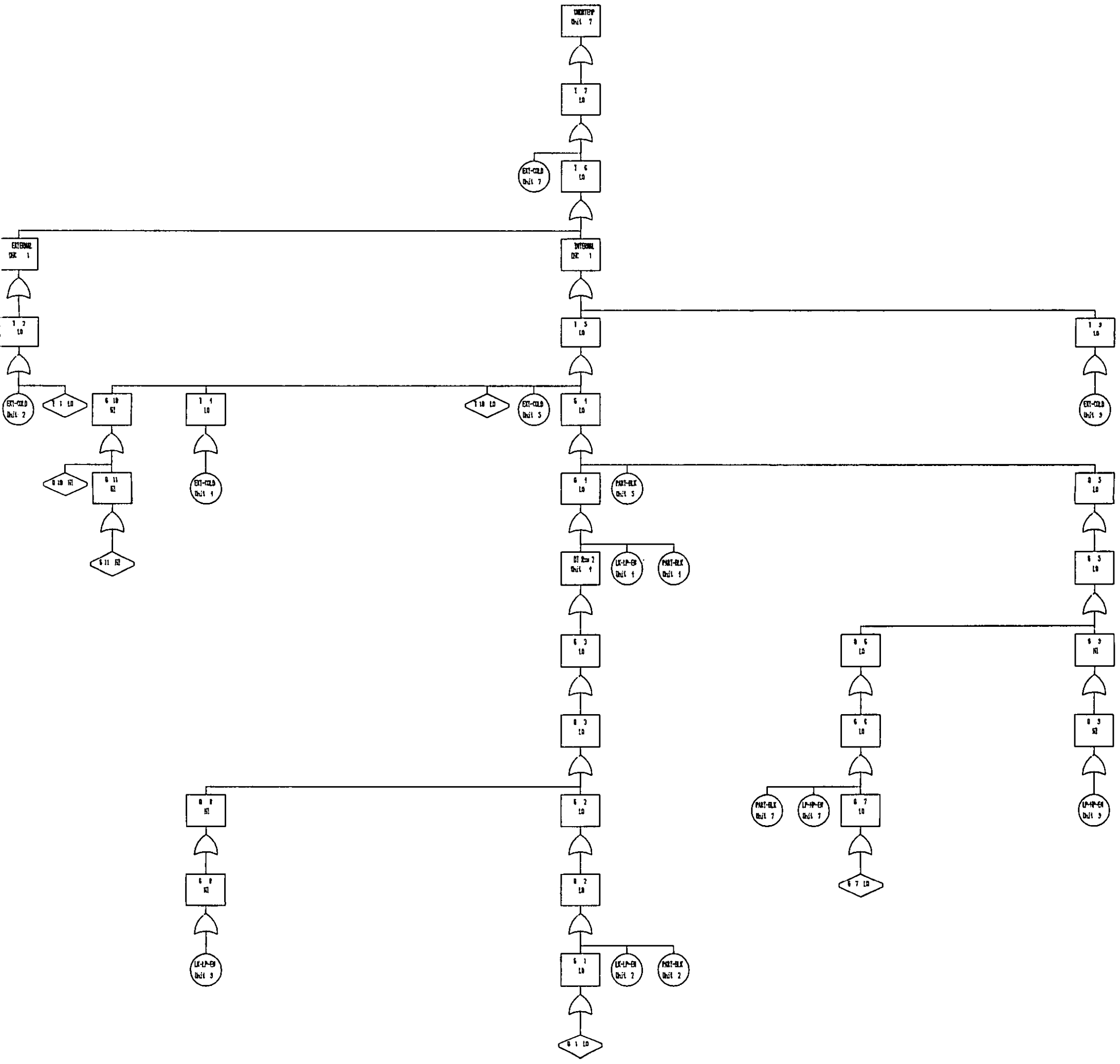

Figure 9.12 - complete fault tree for the system shown in Figure 9.11

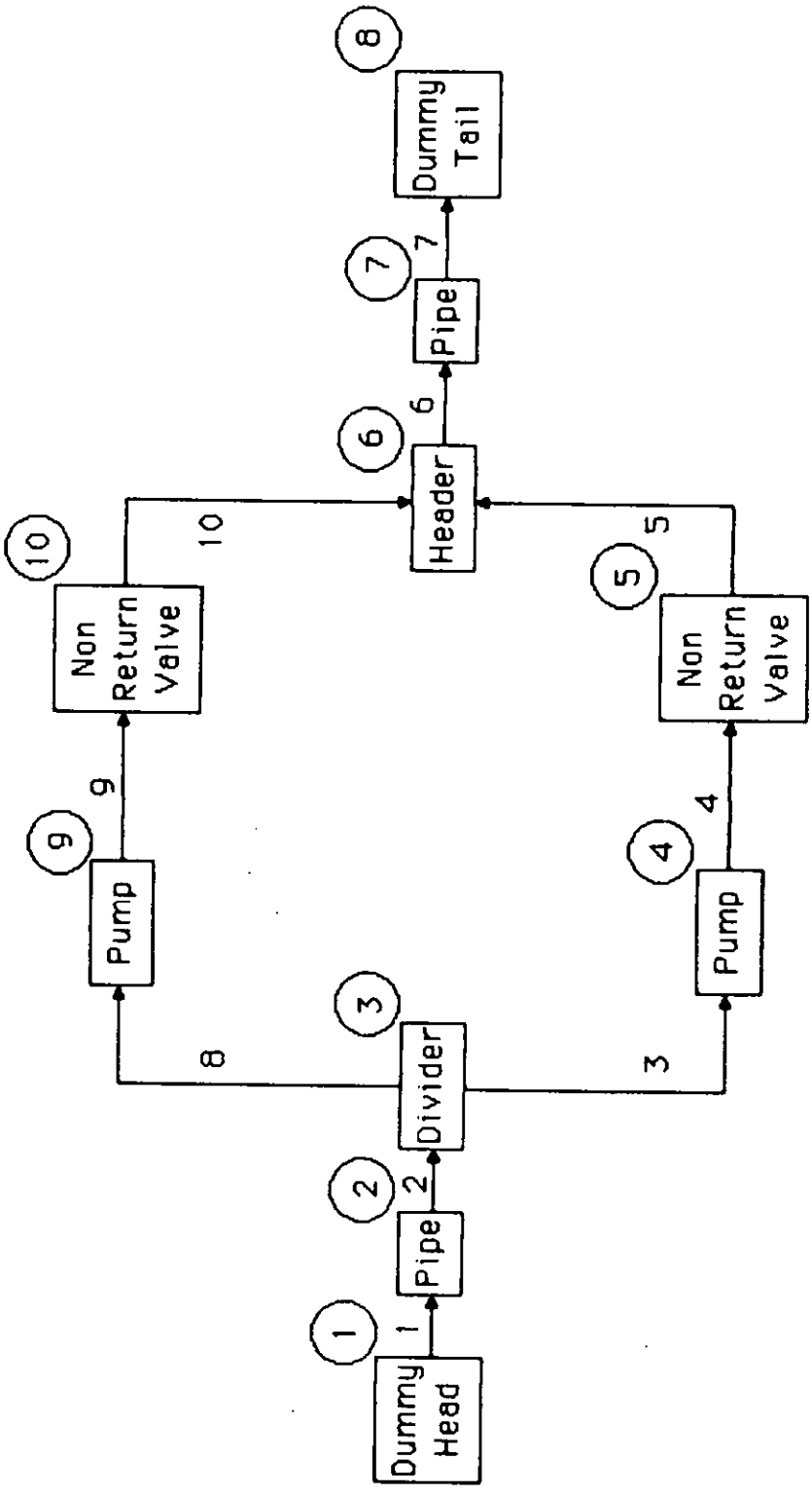

Figure 9.13 - configuration diagram for a pump bank comprising two pumps

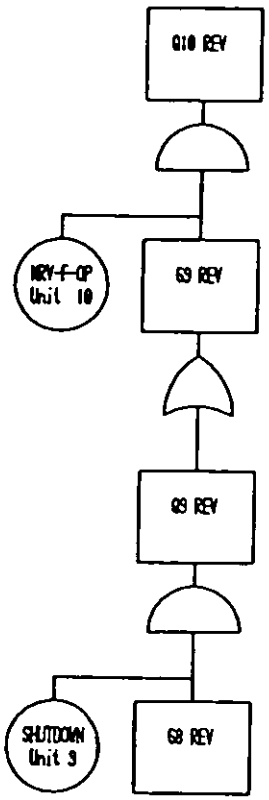

Figure 9.14 - fault tree for reverse flow through one of the legs of the system shown in Figure 9.13

Enabling Faults Example

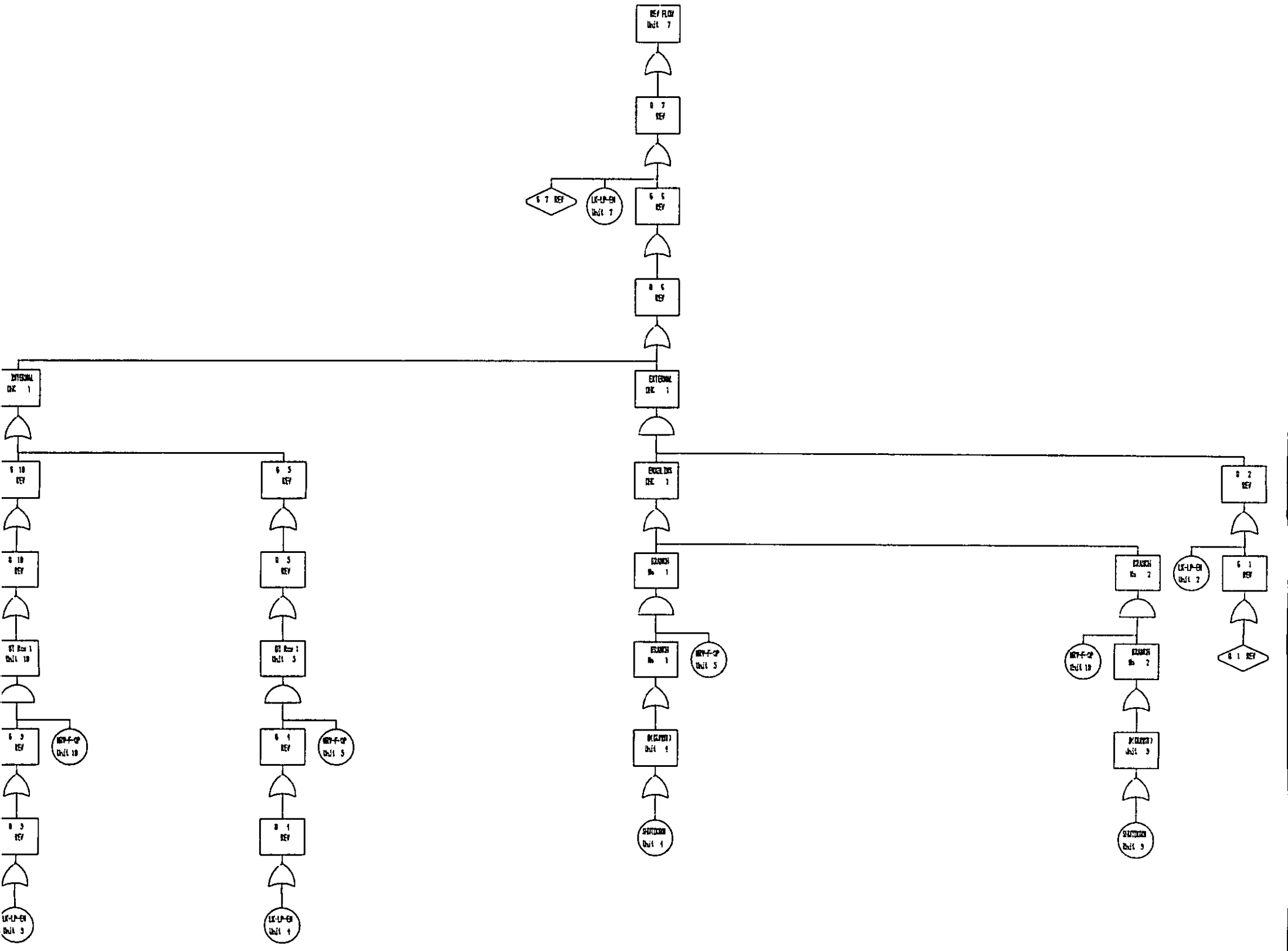

Figure 9.15 - complete fault tree for the system shown in Figure 9.13

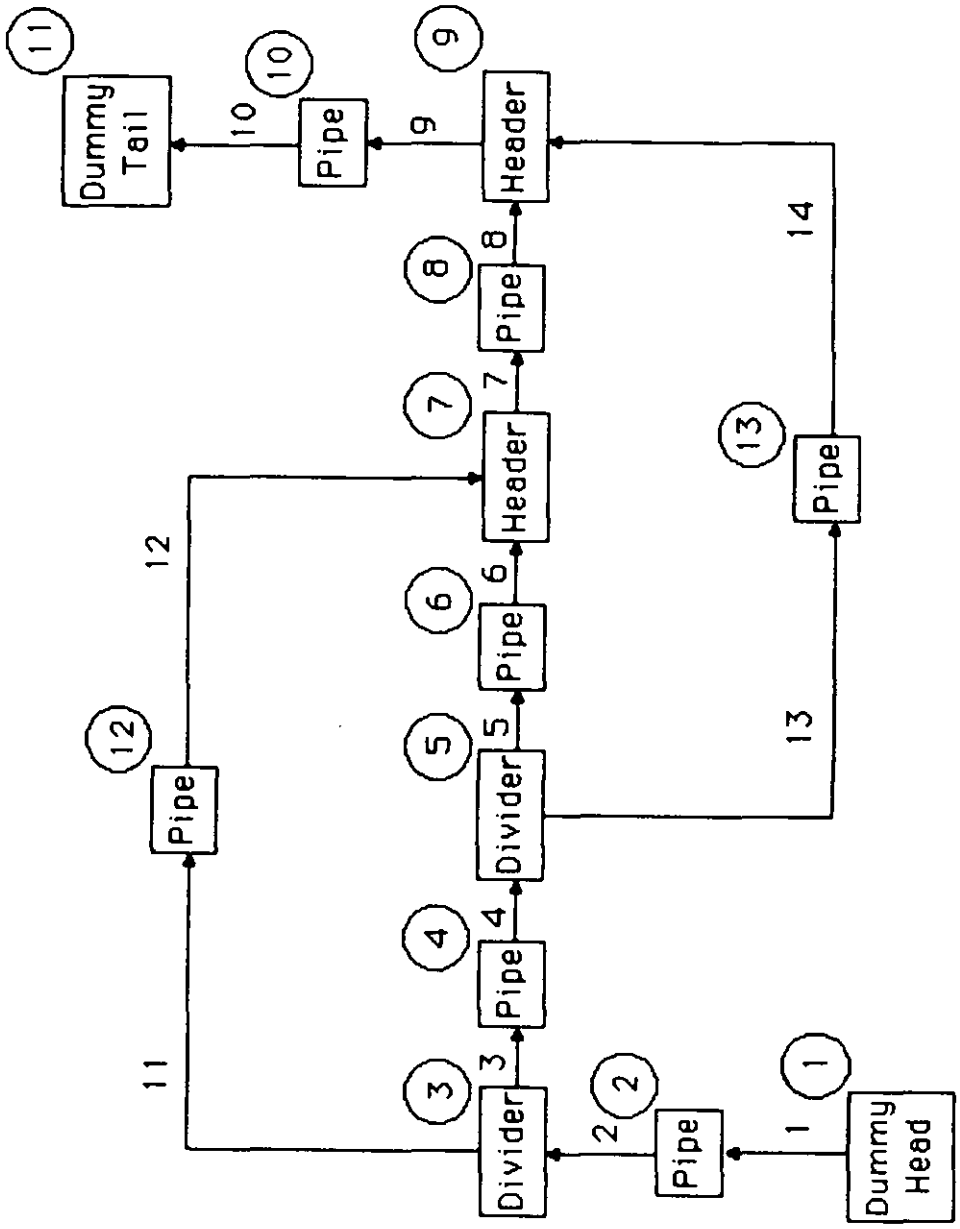

Figure 9.16 - configuration diagram for a system incorporating overlapping divider-header combinations

10) Sequencing

In many chemical plants, especially in plants that involve batch processing, there is some element of sequencing involved. Essentially, sequencing means that the plant state will change during the study. Typically, valves may open and close, and pumps may be started up or shutdown.

A fault tree normally takes no account of time effects. For example, a fault tree for low flow out of a tank might be caused by a leak in the tank, which causes the tank to empty. If the tank is large, then it might take several hours, or even days for the top event to occur. Nevertheless, the fault tree identifies the leak as a cause of the top event. Similarly, a fault tree for a plant involving sequencing ignores time effects, except by virtue of modelling the different states of the plant as the sequence proceeds. The time taken to progress through the sequence is not reflected in the fault tree.

10.1) The Problem

Sequencing presents complications for computer-aided fault tree synthesis. If the state of a unit changes, then the failure modes of that unit also change. For example, no flow through a valve is a normal state if the valve is supplosed to be closed, but is a failure state if the valve is supposed to be open.

The top event of a fault tree might also undergo some change during the sequencing procedure. If the failure states of the plant are variable, then the top event might have to change to reflect the changing plant state. An example of this is a computer controlled sequencing operation. One of the items of

interest may be the reliability of the sequence. Typically, the computer will take in measurements from the plant to decide if and when to proceed with the sequence. The events that will cause the computer to halt the sequence will typically be different at each sequence step.

10.2) The Approach of Others

Both the RIKKE code and the Lapp and Powers methodology take some explicit account of sequencing. None of the other published literature on the other codes makes any explicit mention of the problem, but the approach used in the Lapp and Powers code, based as it is on a purely modelling approach may be adapted to fit these other codes.

The approach used in the Lapp and Powers code, as described by Shaelwitz et al [31], involves the creation of complex models that cover all the possible states of a component. For example, a valve model incorporates the failure expressions for both a normally open valve and a normally closed valve. These failure expressions are made conditional on the value of a special input signal to the valve. This input comes from a timer model, and can take one of two values, which indicates whether the valve is supposed to be open or closed. The timer model contains the sequnce logic, in the form of a changing output signal based on the actual time.

This approach is satisfactory only if the top event of interest is constant throughout the sequence.

The approach of RIKKE, as described by Taylor [43] is similar, but involves a better protocol to define the sequencing operations involved.

10.3) A Solution

An alternative way of viewing a plant that involves sequencing is to regard the plant as a number of different plants that differ from each other only in terms of the states of some of the units. Each of these plants has its own fault tree. A fault tree for a complete sequence can be obtained by combining together all the fault trees for each of the plants. By specifying the top event for each plant independently of the other top events, a complex, changing, top event can be successfully modelled.

Such an approach also has the advantage that it is unnecessary to create models that cover all the possible states of a particular unit, and so modelling is easier. In fact, the models used during sequencing differ in no respect from the standard form. Some sequencing information is required, and this is provided as an extension to the configuration input. The configuration input is used to indicate the initial state of the plant. The sequencing input is the list of the units which change state at each step during the sequence operation, and the new models that correspond to the new states. Also required is the sequence step top event, which can be different for each step in the sequence. It is even possible for the top event for a particular step to be non-existent. This corresponds to a sequence step that cannot cause the top event.

The general model approach can be used to structure a fault tree for a system that involves sequencing in the required way. The top event for a system has two causes - the top event occurs either because it occurs at step 1 in the sequence, or because it occurs after step 1. The event 'top event occurs after step 1'

occurs because either the top event occurs at step 2, or because it occurs after step 2. This procedure can be followed for all the steps in a sequence, as illustrated in Fig 10.1. The definitions of the event 'top event occurs at step n' (n being any step) can vary from step to step, by supplying the appropriate sequencing input, as described above.

10.3.1) A Simple Pump Changeover

This example is a simplified version of the British Gas Pump Changeover system, described in detail in Section 11.5. The configuration diagram shown in Fig 11.2 covers only one of the pumps in the complete example, and omits much of the detail involved. The top event of interest is the sequence does not complete, because one of the checks carried out during the sequence is not validated.

The sequencing operation involves bringing the pump, which is initially off, into operation. This involves the following sequencing operations

a) priming the pump, by opening valves 2 and 9

b) switching on the pump, by closing valve 9 and switching on pump 4

c) bringing the pump online, by opening valve 7

At each of these steps, checks may be made to confirm that the operations carried out during the step were completed successfully. In the present study, the following checks are carried out, corresponding to the operations above

a) confirm that the pressure sensor on the pump inlet detects a pressure

b) confirm that the pressure sensor on the pump outlet detects a pressure

c) no checks are made

The decision on how many sequence steps this represents is at the discretion of the analyst. He can define a sequence of many steps, with only one unit changing its state per step. Alternatively, several units can change their state per step. The minimum requirement is that at least one unit must change its state per sequence step. The sequence checks also have some effect on the sequence steps. The checks carried out at each step may be very complicated - there is no requirement for the sequence check to be a single event. Alternatively, a step may have no checks. However, sequence checks that are carried out when the plant is in different states must be part of separate sequence steps - the step is required to change the plant state. In the present example, a sequence of three steps, corresponding to the three sequence operations noted above will be used.

The sequencing information, as it stands, is not suitable for input to the synthesis package. The information must be specified in a more concise manner. Fig 10.3 expresses the information in the required format, which comprises

a) a list of the units which change state at each point in the sequence

b) the model that models the new state of each of these units

c) a list of the events that will cause the sequence to abort at each step of the sequence

Most of the units that change state can be easily derived from the sequencing operations prepared earlier. Similarly, the new models required for these units follows from this information. The exceptions are the changes required to divider units and header units. As flow paths are enabled and disabled by opening and closing valves, the divider and header models must be changed to reflect the normal flow state through the unit at each point in the sequence. There are three models each for divider and header units, reflecting the following normal flow states

a) there is normally no flow through the unit

b) there is normally flow through only one leg of the unit

c) there is normally flow through both legs of the unit

There is, however, a further complication with divider and header models where flow normally exists through only one leg (type b above). At different times during the sequence, the leg where flow normally exists may change (for a divider) from one outlet leg to the other. For example, the model for this type of divider in the library has port 2 as the leg where flow normally exists, and port 3 as the leg where no flow is expected. If the sequence reaches a point where flow is expected to occur out of port 3, but not out of port 2, then additional changes are required. A special type of unit change, called a 'port swap' must be included in the list of changes for this step of the sequence. The effect of this change is to modify the plant connections such that port 2 is linked to what port 3 was connected to, and vice versa.

As an example of the process involved in defining the changes required at one step of a sequence, consider the third step of the pump changeover sequence, when the pump is brought online. Valve 7 is opened, permitting flow through the pump. No flow existed through the pump at step 2 of the sequence, since both valves 7 and 9 were shut. The model of the divider must therefore be changed to a model where flow normally exists through only one leg (type b above). However, the configuration was initially set up such that port 3 of this divider was linked to unit 7. The reason for this was so that, when the pump was primed at step 1, no port swap changes were required to send the flow to drain, via valve 9. At step 3, therefore, the divider must undergo two changes, one to define a new model to reflect the new flow state through the unit, the other to swap the ports so that flow goes in the required direction.

The list of events that will cause the sequence to abort can be derived from the sequence checks that must be fulfilled. The events must be specified as either variable deviations or basic events. Intermediate events can be used to structure the events.

The complete fault tree for the sequence, given in Fig 10.4 consists of the three fault trees that cause the sequence to abort at each of the three sequence steps. The top event of the complete fault tree is 'Sequence Aborts'. The intermediate events SEQ-F-AT (sequence fails at) and SEQ-F-AF (sequence fails after) are, like Sequence Aborts, part of the sequencing general model of Fig 10.1. The causes of the SEQ-F-AT events are the top events of each step in the sequence - S12 NONE for step 1, and S15 NONE for step 2. There is no top event for step 3, and so the event SEQ-F-AT Step 3 does not appear in Fig 10.4. There are only three steps in the sequence, so there is no event SEQ-F-AF Step 3.

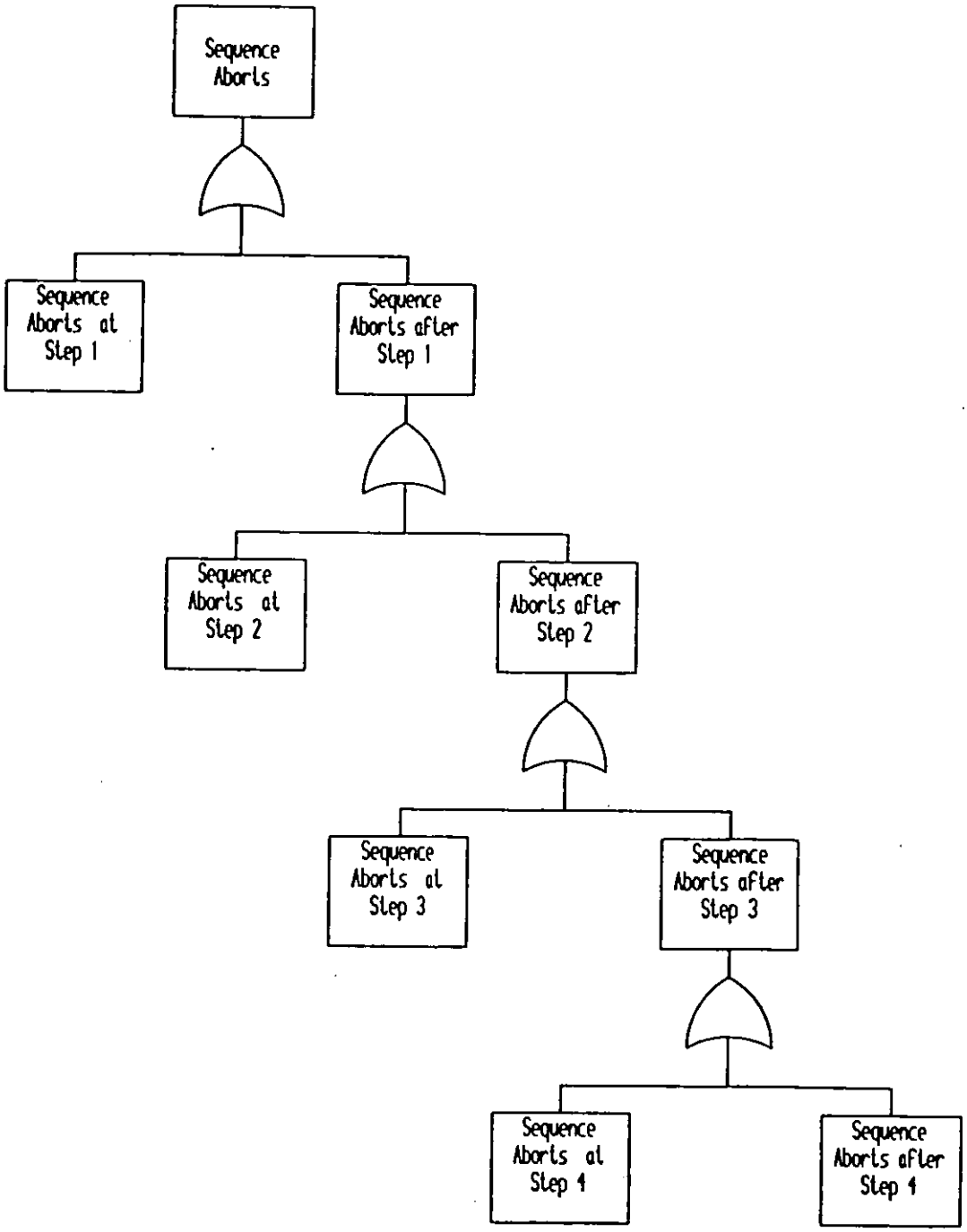

Figure 10.1 - the general model for fault trees involving sequencing

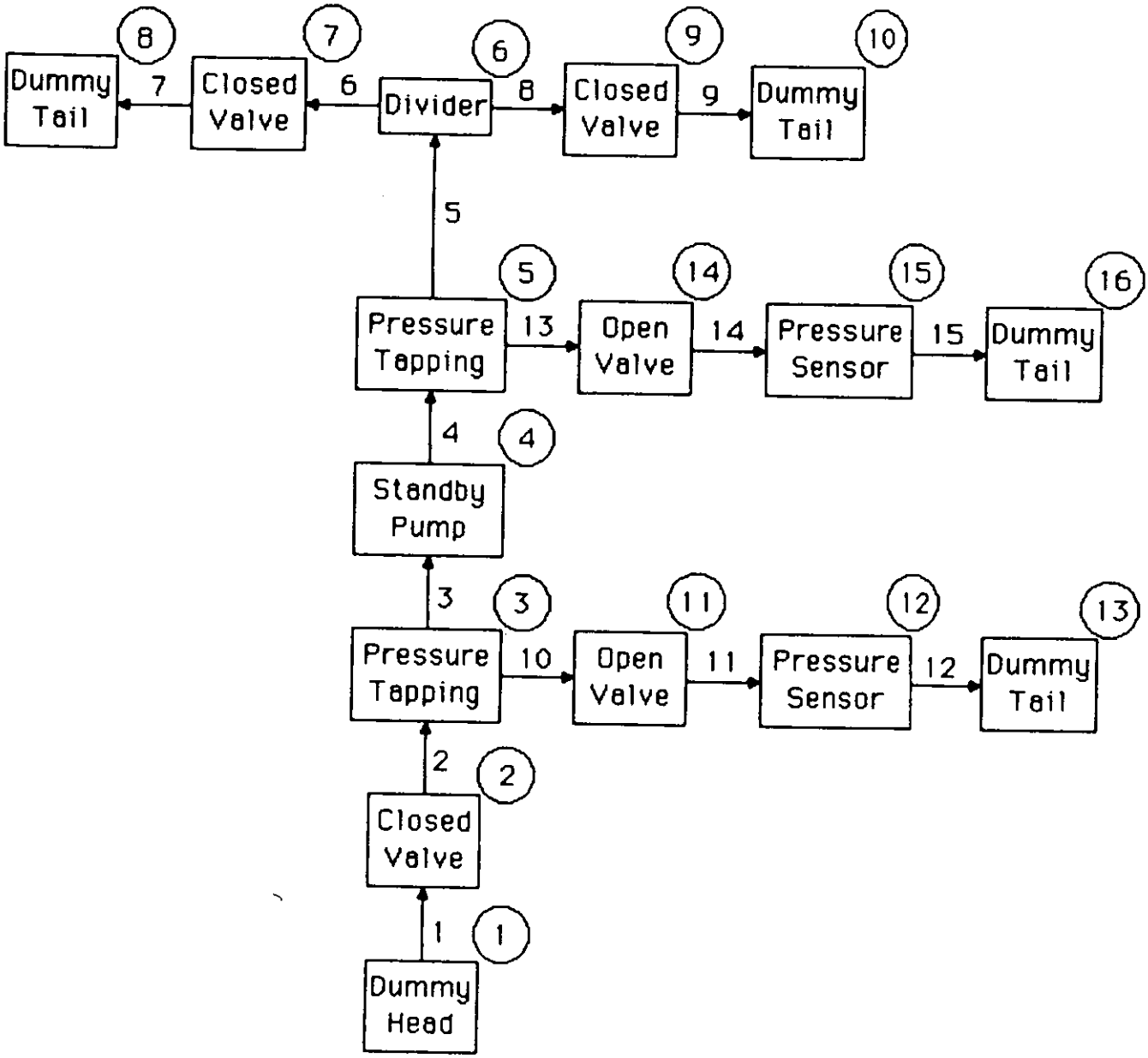

Figure 10.2 - configuration diagram for a simplified pump changeover system

Step 1

Unit 2 becomes Model 9 (Open Valve)
Unit 9 becomes Model 9 (Open Valve)
Unit 6 becomes Model 21 (Unsymmetrical Divider)

Sequence Aborts if S12 NONE exists

Step 2

Unit 9 becomes Model 3 (Closed Valve)
Unit 6 becomes Model 15 (Symmetrical Divider)
Unit 4 becomes Model 61 (Running Pump)

Sequence Aborts if S15 NONE exists

Step 3

Unit 7 becomes Model 9 (Open Valve)
Unit 6 becomes Model 21 (Nonsymmetrical Divider)
Unit 6 becomes Port Swap

Figure 10.3 - sequence steps and abort conditions for a simplified pump changeover system

Simple Pump Changeover

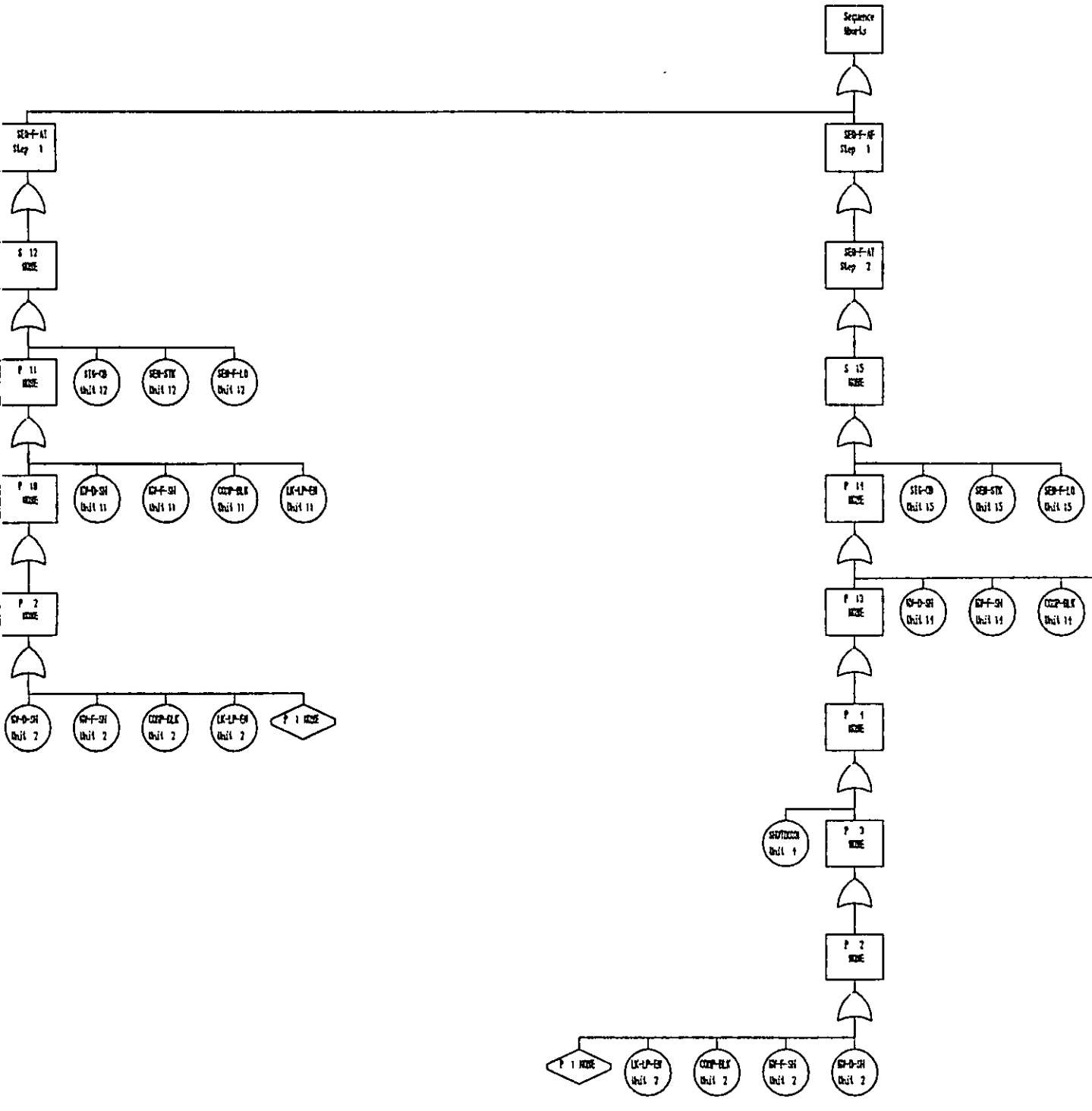

Figure 10.4 - complete fault tree for the system shown in Figure 10.2

11) Worked Examples

This chapter examines five complete plant sections in some detail. All the examples have been considered earlier in the thesis, when they were used to illustrate particular points about one particular type of special system, such as contol loops or sequencing. This chapter considers these plants as a single unit, and shows how the various special systems interact.

With the exception of the pump changeover system, all the examples have been studied by others in the literature.

11.1) A Temperature Control System

This example was presented by Lapp and Powers [23], and has aroused considerable discussion in the literature. The system, pictured in Fig 11.1, is designed to cool hot nitric acid using cooling water. There is a feedback temperature control system, and a feedforward trip system, designed to stop the flow of nitric acid should the cooling water flow fail completely.

The study in this section involves more detail than appears in the Lapp and Powers paper, and involves three secondary failures. Firstly, water in the nitric acid line will result in an exothermic reaction, and a high temperature. Secondly, nitric acid in the water line will have the same effects. The third secondary failure is that nitric acid in the water line will cause corrosion, and lead to a leak to low pressure environment.

11.1.1) Decomposition

The configuration diagram is given in Fig 11.2. There were no special problems in creating the configuration diagram.

The control loop and the trip system are straightforward. The control loop controls the temperature of the nitric acid downstream of the heat exchanger by manipulating the flow of cooling water, and the trip system stops the flow of nitric acid should the cooling water flow drop to zero.

The effects of the three secondary failures have been noted above. Their causes are all modelled as X HI. Note that the component represented by X is different in the two streams. In the nitric acid stream, X HI represents a high concentration of water, while in the water stream, X HI represents high concentration of nitric acid. An alternative approach would have been to have used component subscripts, for example XA and XB to model the two impurities.

11.1.2) Fault Tree Synthesis

The fault tree for this figure is displayed in Fig 11.3. The top event, high temperature of nitric acid has two causes, either high temperature upstream accompanied by flow in the normal direction, or high temperature downstream, accompanied by reverse flow. High temperature downstream may be caused, directly, by a hot source downstream (represented by the diamond event U5 HI), or, indirectly, by the presence of water

downstream (represented by the diamond event Y5 HI). Reverse flow may be caused in several ways, including an internal leak in the heat exchanger (INT-LK Unit 3), the assumption being that the nitric acid is at a higher pressure than the cooling water.

High temperature upstream can result only if the control loop is faulty, or is overloaded. As noted in Section 6, a control loop is overloaded by either no flow or reverse flow in its manipulated stream, which is, in this case, the cooling water. Three overloading paths have been identified. The first is no flow of cooling water into the heat exchanger, leading directly to loss of cooling. The trip system comes into consideration since it acts when no flow of cooling water is detected and, furthermore, can prevent the top event by preventing flow of nitric acid in the normal direction. With one exception, all the causes of no flow of cooling water into the nitric acid cooler are detectable by the trip sensor. The exception is a large leak in the control valve (LK-LP-EN Unit 10), which causes an increased flow at the trip sensor. All the other causes of no flow of cooling water into the heat exchanger are, therefore, ANDed with functional failure of the trip system. It should be noted that corrosion is not identified as a potential cause of the leaks causing no flow of cooling water. This omission is due to the fact that the fault tree methodolgy described in this thesis makes no allowance for time effects. Corrosion requires that impurity be transported to the site where the corrosion occurs, in other words, some flow - in either direction - is required. However, the methodology rejects some flow as inconsistent with no flow, and so corrosion is not identified as a potential cause of no flow of cooling water.

The other two overloading branches occur as a result of reverse flow of cooling water. One branch is a direct cause of loss of cooling, it being assumed that reverse flow can never cool the nitric acid sufficiently. The other arises because reverse flow in conjunction with impurity downstream increases the temperature of the cooling water, and so is a cause of loss of cooling. This latter branch is redundant in minimum cutset terms because of the assumption that reverse flow itself causes loss of cooling - there is no need for impurity downstream to occur as well. The occurrence of redundant branches is a common feature of rigorous fault tree synthesis techniques. In the effort to trace every possible cause of an event, some of the more obscure causes are frequently redundant. In some cases it is possible for the methodology to identify such redundancy, and to remove it. In other cases, as above, it is not possible. The only effect of not removing redundant branches is that the fault tree is larger than it needs to be. One situation where the methodology has removed redundant branches in this fault tree relates to the reasons why corrosion is not identified as a potential cause of the leaks causing reverse flow. As for no flow, the methodology rejects flow in the normal direction carrying impurity to the site of corrosion as inconsistent with reverse flow. However, reverse flow carrying impurity from downstream is a redundant branch. Since reverse flow already occurs in the current branch, reverse flow and impurity downstream can add no additional mimumum cutsets to the fault tree, and so is redundant. Because of the method by which this branch synthesised, the methodology can reject it as redundant, and so removes it from the fault tree.

The remaining causes of the top event are caused by failures in the control loop (e.g. CV-F-LA Unit 10), or by events causing a temperature deviation of the nitric acid at the outlet of the heat exchanger, accompanied by events causing the control loop to fail to respond (e.g. CV-STK Unit 10). Events causing a temperature deviation include impurities in both the nitric acid and the cooling water streams, and increases in the supply temperature of either stream.

11.2) Composition Control System

This example has been studied by Lihou [48]. Hydrocarbon and oxygen are mixed in a packed bed catalytic reactor, to produce an inert product. It is important that the product contain no oxygen. There is therefore a complex trip system that will activate if oxygen is detected in the product, or if the temperature in the reator becomes too low, indicating an incomplete reaction. The control system is also complex, and is designed to ensure that the reactants are mixed in the correct proportions. Both inlet streams are under flow control, with the setpoint for the hydrocarbon flow control loop being determined by the oxygen flow rate.

A flow diagram for this system is given in Fig 11.4.

11.2.1) Decomposition

The configuration diagram for this system is displayed in Fig 11.5.

The main problem with decomposition is defining the control and trip systems in terms of several simple control and trip loops. There are in fact three control loops (identified below by CL and an index number) and four trip systems (TS), as follows :-

CL1: control the composition of the product downstream of the reactor by measuring the flow of oxygen and manipulating the flow of hydrocarbon

CL2: control the flow of oxygen using a standard flow control loop

CL3: control the flow of hydrocarbon using a standard flow control loop, but with the setpoint determined by Control Loop 1

TS1: use the three composition sensors to control Trip Valve Unit 18

TS2: use the two temperature sensors to control Trip Valve Unit 18

TS3: use the three composition sensors to control Trip Valve Unit 19

TS4: use the two temperature sensors to control Trip Valve Unit 19

Note the treatment used to model the trip relay. In the flow diagram, it was linked to the trip switches for both the composition and temperature trip systems. This approach was rejected in decomposition, since it would lead to an relay model specific to this particular plant. In the configuration diagram, therefore, a signal header unit has been used to combine the two trip signals into one. The relay can thus be modelled as a unit with a single input, and a single output, and is therefore a much more general model.

A further point to note in the decomposition is the vessel port splitter used to attach two different temperature sensors to the reactor. This was done to avoid the need to model the reactor with two vessel ports to which the two temperature sensors could be linked.

The plant is provided with both electric power and instrument air utilities, which feed several different units. This is shown in the configuration diagram, where the air supply is unit number 38 and the power supply is unit number 39. For convenience, the connections to the utilities are shown separately (Fig 11.5(b)).

11.2.2) Fault Tree Synthesis

The fault tree for this system is shown in Fig 11.6. The model for the reactor has exactly the same causes for low temperature and the presence of oxygen, and so, although the top event is a deviation in composition, all four trip systems can protect against all the potential causes of oxygen in the product. The composition trips can detect the presence of oxygen directly. The temperature trips can detect its presence indirectly, via low temperature in the reactor.

The first control loop to be analysed is the composition control loop. There is one overload branch for this loop, the causes of complete loss of hydrocarbon flow.

Remaining events can be split into two groups, those that are detectable by the composition control loop, and those that are not. Events detectable by the composition control loop are ANDed with that control loop, and are the events causing high flow of the oxygen stream into the reactor detectable by the oxygen sensor. Such events are in the domain of the oxygen flow control loop, and so the special treatment accorded to control loops is applied to the events detectable to the composition control loop.

Events not detectable by the composition control loop include catalyst deactivation, poor mixing in the reactor and low flow of hydrocarbon. The last of these is in the domain of the hydrocarbon flow control loop, and so the special control loop treatment is applied once again.

Minimum cutsets are important in the analysis of a fault tree of this type, because failures in some components affect the performance of several control loops or trip systems. The event Q7 HI, for example, requires latent failures in two control loops (the oxygen flow loop, and the composition loop), and functional failure in all four trip systems. However, all these conditions are fulfilled by the two events SEN-STK Unit 11 and ACT-STK Unit 36.

11.3) A Distillation Column

This system is based on one of the distillation columns in a plant used by Shepherd et al [53] in the training of process operators, and is displayed in Fig 11.7. The column is a binary distillation column, with a saturated vapour feed, and is designed to separate components A, the more volatile component, and B. There are four control loops on the column. One control loop is designed to regulate the composition of the distillate, and it does this by sensing the tops temperature and adjusting the reflux flow to the column. A second control loop manipulates the distillate product rate to maintain a constant level in the reflux tank. A third control loop manipulates the boilup rate to control the bottoms product composition. Finally, the level in the partial reboiler is controlled by manipulating the bottoms takeoff flow rate.

11.3.1) Decomposition

The configuration diagram for this system is displayed in Fig 11.8. The control loops are as follows :-

CL1: tops composition, manipulating the reflux flow and sensing the tops temperature

CL2: reflux tank level by manipulating the distillate takeoff flow

CL3: bottoms composition, adjusting the flow of steam to the reboiler on the basis of the temperature at the base of the distillation column

CL4: reboiler tank level by manipulating the bottoms product takeoff flow

The only point to note is that vessel ports not linked to sensors must be linked to dummy tails, so that all the vessel ports can be associated with a connection number, for example the vapour vessel port of the reboiler.

11.3.2) Fault Tree Synthesis

The fault tree for this system is displayed in Fig 11.9, and was synthesised with all pipe type faults (leaks, blockages and external hot and cold sources) suppressed, reverse flow effects ignored and drawn with variable deviations and intermediate events with only one cause concertina-ed together. Even so, the fault tree is still rather large.

Comparing this fault tree with the fault tree synthesised for this system in Section 5.4.4 (see Fig 5.7), when control loops were ignored, indicates the importance control loops have on fault tree synthesis. The fault tree of Fig 11.9, incorporating control loops, splits approximately into two halves. The top half of the tree contains the events that overload the tops composition control loop, i.e., faults that cause no flow of reflux. Four such faults were diagnosed at various stages of the synthesis, and so there are four overload branches. One branch occurs as a result of the decision table in the distillation column which notes that a low reflux ratio exists if there is no reflux and some distillate takeoff. The causes of this are surprisingly large, and cover not just blockages in the reflux line, but an increased takeoff of distillate. One of the causes of this is identified as a high level in the reflux tank, resulting in an increased aperture in the distillate takeoff valve. There is a fault here in the methodology, in that this should be identified as a potential cause of increased reflux flow, not of complete loss of reflux flow. It is this type of problem that flow ratio was introduced to overcome, and it appears that there is a need to extend a similar treatment to a wider variety of situations.

The other three overload branches are the result of more complex propagation paths, and are in fact redundant in minimum cutset terms. Two are identical, and arise because high column pressure can be caused by low tops takeoff, and hence no flow of reflux. High column pressure increases the temperature at the base of the column. The bottoms composition control loop should respond to the temperature increase by decreasing the steam flow to the reboiler, thus reducing the pressure. However, if it does not, an increasing amount of component B will be "boiled-up", increasing the composition of component B in the distillate. These overload branches therefore require latent failure of the bottoms composition control loop to cause the top event.

The final overload branch occurs because complete loss of reflux will cause a high boilup ratio, since the liquid takeoff will be reduced. High boilup ratio will increase the composition of component B in the distillate. It is unclear why this branch, like the two branches above, does not require the bottoms composition loop to have failed latently. After all, reducing the steam flow to the reboiler will reduce the amount of boilup, and hence the boilup ratio. It is possible that the bottoms composition loop is identified as being overloaded, but this should not occur - reducing the boilup to zero will prevent the top event from occuring.

The bottom half of the fault tree is much more closely related to the tree synthesised in Section 5.4. The only additional events that appear in the current tree relate to control loops component faults.

11.4) A Propane Pipeline Problem

This example was first introduced by Lawley [51]. The flow diagram for the system is shown in Fig 11.10. The process description is as follows.

The plant as shown is a proposal to utilise an existing 10 mile long mild steel pipeline to transport propane from a storage tank to a consumer buffer tank. The problem is that, in the storage tank, the propane is at -45 C, a temperature that the mild steel pipeline is not designed to withstand. Therefore, before passing through this pipeline, the propane is to be heated using glycol as a heating medium. A glycol supply already exists on site, and supplies various other users. The glycol itself is heated using low pressure steam.

There is a temperature control loop and an independent temperature trip system designed to prevent propane at low temperature entering the mild steel pipeline. There are additional control loops which regulate the level of the consumer buffer tank, the glycol temperature and the steam condensate tank level. There are also trip systems on the consumer buffer tank, to prevent the tank overflow, and on the storage tank pumps, to prevent the pumps pumping against no flow. This trip system works by opening up a kickback line to the storage tank, to permit a small circulation rate through the pumps.

The top event of interest is too low a temperature in the mild steel pipeline. In addition to the normal causes of this, low temperature may also result if the propane is subject to a sudden depressurisation. This will lead to flashing of the propane, which is in

liquid form, and a drop in temperature.

11.4.1) Decomposition

The configuration diagram for this system is shown in Fig 11.11. The control loops and trip systems are defined to be :-

CL1: control the temperature of the propane by manipulating the flow of glycol

CL2: control the level in the consumer buffer tank by manipulating the flow of propane into the tank

CL3: control the temperature of the glycol by manipulating the flow of steam to the glycol heater

CL4: control the level of condensate in the condensate drum, by manipulating the takeoff flow of condensate

TS1: prevent low temperature propane reaching the mild steel pipeline by shutting Trip Valve Unit 31

TS2: prevent overflow of the consumer buffer tank by shutting the combined Control/Trip Valve Unit 41

TS3: prevent the overflow of the consumer buffer tank by shutting the Trip Valve Unit 45

TS4: prevent the pumps pumping against no head by opening a kickback line to the supply tank through Trip Valve Unit 59

There are several points that should be noted in the decomposition of this system.

Firstly, the bank of glycol pumps is a parallel system, with four pumps normally working, and the fifth on standby. To produce the desired throughput, at least three of these pumps must be working. Note that a standby pump is used to represent the pump that is not normally working. The model for this is slightly different from a pump that is normally on - one difference is the state of the pump to cause no flow through the pump. For a pump that is normally on, shutdown is a cause. For a standby pump, failure to start up on demand is a cause.

Secondly, in the definition of the propane temperature control loop, the stream that bypasses the heat exchanger is not specified as a manipulated stream of this loop. Although the flow through the bypass is dependent on the position of the three way control valve, the definition of what is the normal flow through the bypass depends on the control valve position, and not on the value of the sensed variable. For instance, low flow through the bypass is caused only by process unit faults, such as blockages, and not by the control valve position.

Thirdly, the pump protection system is specified such that it will activate on no flow through the pumps, rather than low flow. The reason for this is that the flow at which the trip is designed to activate is very small compared to the normal flow.

11.4.2) Fault Tree Synthesis

The fault tree for this system is displayed in Fig 11.12, and is very large. It can, however, be split into a number of smaller sections for the purpose of considering it in some detail. The first distinction that can be made is between primary and secondary failures.

The temperature trip system can protect against all the primary failures that cause low temperature, including failure of the temperature control system.

The temperature control loop can detect all the primary faults that can cause low propane temperature at the pipeline, and can correct for all of these except for the faults that result in complete loss of glycol flow through the exchanger. These faults form the overload branch, and include events such as all the glycol pumps fail together, and too much demand on the glycol supply by other users (Q103 HI).

The events that can be corrected for by the temperature control system are the four events that cause the heat exchanger to heat the propane insufficiently. These are corollaries of the events that caused the nitric acid to be of too high a temperature in the example presented in Section 11.1, namely deviations of the two inlet temperatures and the two inlet flows. The fact that the Propane Pipeline System is more complicated than example considered in Section 11.1 means that the causes of these events are more complex. Nevertheless, the causes are still basically the same.

Low temperature of the propane to the exchanger has only one cause, which is low temperature in the storage tank. Low temperature of the glycol has more complex causes, since the glycol temperature is determined by a contol loop and another heat exchanger. However, it is essentially the same as low temperature of the propane. The glycol temperature control loop is overloaded by complete loss of low pressure steam, but can otherwise correct for all the other potential causes of glycol temperature deviations, which are deviations of the inlet temperatures and the inlet flows to the glycol heat exchanger. So the process continues, propagating through the steam condensate control loop, which affects the steam flow through the glycol exchanger.

Low flow of glycol to the propane heater is another potential cause of low propane temperature. However, high flow of glycol is a potential cause of low glycol temperature, which itself causes low propane temperature. These opposite deviations both appear in the fault tree, since both are found to be potential causes of the top event.

The causes of high flow of propane are influenced by the consumer buffer tank control and trip systems. High flow of propane into the tank should be resisted by the control loop, and should eventually be prevented by the activation of the level trip systems. However, there are causes of high propane flow that do not also result in high tank level, such as the drain valves Units 52 and 54 being open. Such faults are therefore ANDed neither with level control loop stuck nor with functional failure of either trip system. There is, however, a slight problem in correctly ANDing the flow faults with trip functional failure, as described in Section 7.3.6. As a result, there are some events, such

as valve 51 opening, that should be ANDed with trip functional failure of the level protection systems, but are not so ANDed.

This covers the description of the primary failures. Although there are a large number of such failures, the fault tree approach is ideally suited to considering these in a logical fashion.

The causes of the secondary failure, depressurisation of the propane pipeline causing low temperature by flashing present a number of interesting propagation paths.

There are two basic causes of depressurising the propane pipeline, events that cause depressurisation upstream of the mild steel pipeline, and events that are downstream. Since the propane temperature control loop can detect the low temperature that will result from depressurisation, all the depressurisation causes are ANDed with the control loop being stuck. It is assumed that the control loop can correct for the low temperature by increasing the flow of glycol to the propane heater. However, the propane temperature trip system, although it can detect the low temperature, cannot protect against the downstream causes of depressurisation. This is because closing the trip valve, which is upstream of the pipeline, can never prevent depressurisation of the pipeline through units that are downstream of the pipeline. However, the trip can protect against upstream causes, since closing the trip valve will effectively isolate the pipeline from the source of depressurisation.

The upstream and downstream causes of depressurisation are modelled, respectively, by P REV

and R HI. The causes that are downstream are fairly straightforward, and consist entirely of valves that are supposed to be closed opening. One point to note is that failures in the relief valve on the consumer buffer tank can be protected against by the level trip system. This is because, if the relief valve is open, high flow into the tank, and hence high level will result. Closing the trip valves of the level trip will isolate the source of depressurisation from the pipeline.

It is the upstream causes of depressurisation that are much more interesting. As with the downstream causes, the causes of depressurisation are generally the opening of valves that are supposed to be closed. There are however, some points to note. The non-return valves are assumed to be able to prevent the reverse flow implicit in depressurisation upstream, and so the non-return valves must fail open to propagate the causes further. Additionally, it is assumed that depressurisation will not occur through the pumps, unless the pumps are stopped. However, it is assumed that pressure relief into the propane storage tank is certain to occur, since the tank is at a lower pressure than the pipeline. The direct causes of upstream depressurisation are therefore back through the various flow paths into the storage tank, which are

a) through either of the propane pumps
b) through the pressure relief valves
c) through the kickback line

Pressure relief will occur through the kickback line if either the pump protection trip valve, or the trip valve bypass valve open. The trip valve can open through operational failure of the trip system, but can

also open because the trip system detects no flow through the pumps. There are numerous causes of this, such as both pumps failing off. However, the downstream causes are much more interesting. The pump protection trip will detect no flow if, for example, the level trip on the consumer buffer tank activates, either in error, or through a genuine demand. However, it should be noted that, although activation of the propane temperature trip will also cause the pump protection trip to detect no flow, this fault does not appear in the fault tree. The reason for this is that activation of the temperature trip system precludes the propagation of depressurisation to the pipeline. Temperature trip activation, therefore, is not a valid cause of the top event.

There are three slight problems with the fault tree displayed in Fig 11.12. Firstly, when examining the causes of no steam flow to the glycol heater (which causes low glycol temperature, and hence low propane temperature), one cause is low flow out of the steam condensate tank. Clearly, no flow into the tank will not occur while flow out of the tank exists. The problem here is the scale. No flow into the tank will exists if the tank level becomes too high. High tank level will occur if the outlet flow becomes low. Ergo, low outlet flow causes no inlet flow.

The second problem occurs when examining the causes of high flow of glycol (which causes low glycol temperature, and hence low propane temperature), and occurs because of the divider-header combinations that occur around the three-way control valve. The fault tree synthesis package assumes that divider-header combinations, if nested at all, are nested in such a way that a particular combination is completely

contained within another combination. This does not occur with the divider-header combinations that exist in this particular case, and the system is confused. As a result, the fault valve closed Unit 91 is identified as possible cause of high flow of glycol, when in fact it will tend to cause low flow of glycol.

The third problem, already noted in Section 11.1, concerns effects that occurred in the past. These occur when examining the causes of why the level trip system should activate (causing no flow through the pumps, the pump kickback trip valve to open, and depressurisation through the kickback line into the storage tank). Clearly, one cause of high tank level is that high flow into the tank occured some time previously. However, the causes of high flow are deleted, since they are inconsistent with no flow of propane.

The fourth problem, already noted in this section, and in Section 7 concerns the interaction of high flow into the consumer buffer tank and the level protection systems on the tank.

11.5) Pump Changeover

This example was undertaken as an example with an industrial company. The system, depicted in Fig 11.13, is a complex, computer controlled pump system, in which one pump is normally working, and the other is off. Each pump is supplied with high and low pressure seal water, and can be flushed with water when required. Each pump system has two drains directly on the pump and a third drain line further downstream. All six drain lines feed a common header and return to a tank, not shown on Fig 11.13.

Pump changeover, involving starting up the pump that is off, and shutting down the pump that is on, is controlled by a computer, and involves several steps. At each step, the computer changes the state of the plant, by opening and closing valves, or starting up or shutting down a pump. The computer is linked to the plant though a number of sensors, and the computer uses these to detect the success of its operations, and as a guide as to whether to proceed to the next step in the sequence. The valves have proving switches that are linked to the computer, which enables the computer to determine the position of each valve.

The top event of interest is sequence fails to complete, because one of the computer checks is not fulfilled, or is detected as unfulfilled, at some point in the sequence.

11.5.1) Decomposition

The configuration diagram for the initial configuration (pump PU2A operating and pump PU2B shutdown) is shown in Fig 11.14. There are four control loops in the configuration, controlling the low and high pressure seal water flows to each pump. There were no particular problems associated with decomposition.

11.5.2) Sequence Definition

The sequence operations and the conditions which must be fulfilled for the sequence to proceed further are given in Fig 11.15. These are not in a form that is suitable for input to the fault tree synthesis package. The form required can be derived from this information, however, contains three types of information. Firstly, information on the units which change state at each sequence step, and the library model reference numbers that model the new states. Note that the divider and header models must be updated to ensure that the model which reflects the normal flow state at each step in the sequence is used.

The second type of information is used to ensure that the dividers and headers can be changed to conform to the normal flow state at each step in the sequence. Not only is it necessary to ensure that the correct type of model is used, it is also essential that the ports be linked correctly. Sometimes it is necessary to interchange the connections of the two outputs of a divider, or the two inputs of a header. This type of information is called a port swap change, and is necessary when using dividers and headers which have

different flow states through their outlet and inlet ports.

The third type of information are the events that cause the sequence to abort at each stage, and can be derived from the conditions that must be fulfilled, listed in Fig 11.15. Intermediate events are used to structure the sequence abort conditions, and to reduce the number of causes in each minitree. Note that valve proving checks generate two failures, which are that the valve fails to move; and that the computer fails to detect that the valve has moved correctly. These faults are represented by the basic events TV-FT-SH (valve fails to shut) and TV-FT-OP (valve fails to open) for the first type, and TSW-F-ON (switch fails on, or valve proves open) and TSW-F-OF (switch fails off, or valve proves closed).

Fig 11.16 lists this information for Step 6 in the sequence. Although only one valve (S23B) is opened during the step, changes need to be made to four units. Firstly, the model for the valve must be changed to an open valve model. Changes in the models used for the divider Unit 3 and header Unit 12 are necessary since these now involve flow in both legs. Previously, flow existed in only one leg, through pump PU2A. Finally, the divider Unit 21 must also change state. In the previous step, it had no flow through it, now its normal state is for flow out of one leg, to the solution outlet. The model used must therefore be changed. However, it is also necessary to perform a port swap on the unit, since the ports of the unit are linked such that flow will go to the drain, not the solution outlet. This was the configuration required for Steps 2, 3 and 4, when the pump was primed.

Section 10.3.1 contains more details on sequence definition.

11.5.3) Fault Tree Synthesis

The fault tree for the complete sequence abort event is enormous, and is too large to be synthesised at one time. Instead, five fault trees were synthesised, which are the causes of the sequence aborting at different points in the sequence. Combining these fault trees will give the entire sequence aborts fault tree. The five fault trees are given in Figs 11.17 to 11.21.

Part of the reason for the size of the complete fault tree is that it is made up of thirteen individual fault trees, one for each step of the sequence. Some of the fault trees are themselves quite large, because of the large number of propagation paths. When tracing the causes of, for instance, some pressure, each divider and each header provides two propagation paths that require following. In this respect, the pumps perform as complex divider/headers with a total of six flow paths in and out of the unit.

The complete fault tree is too large to consider in detail, but one sequence step will be examined. Fig 11.20 is the fault tree for the sequence aborting at steps 7, 8, 9 or 10. Step 8 will be examined in detail. At this point in the sequence, the pump that was running has just been switched off, and the computer checks the pressure switch at the pump outlet to confirm that there is no significant pressure there. The sequence will abort, therefore, if significant pressure does exist.

There are three reasons why pressure could be detected at the pump outlet. Firstly, the pressure switch could have failed so that it indicates that it detects a significant pressure. In the fault tree, this

is represented by SEN-F-HI Unit 67 (sensor fails high) or SEN-STK Unit 67 (sensor stuck).

Secondly, pressure could actually exist. The most straightforward cause of this is that the pump has not been switched off. However, due to the pipework layout there are numerous ways in which switch could be pressurised even if the pump is switched off. These include the valve from the flushing water failing open, or some path to the outlet of the pump that has just been switched on existing. There are several such paths, and so the fault tree for the sequence aborting at Step 8 is larger than might at first have been thought.

Finally, pressure could be detected by the pressure switch if the pressure in the pipework that existed before the pump was switched off is "locked in". Since the valves downstream of the pump (Units 11, 49 and 51) are expected to be shut, there is no fault needed downstream to cause the pressure to be locked in. Pressure will be locked in, therefore, if valve 4 is shut, or if, for some reason, pressure cannot be relieved either upstream or through the other pump.

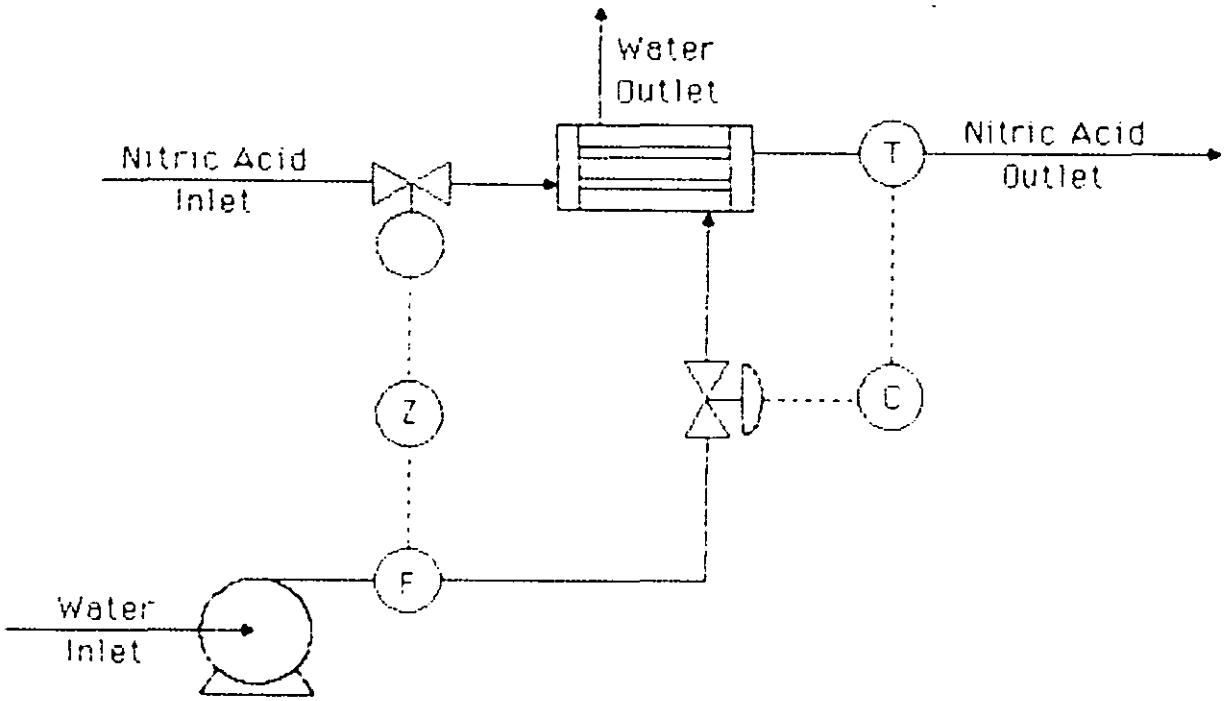

Figure 11.1 - a nitric acid cooling system, after Lapp & Powers [23]

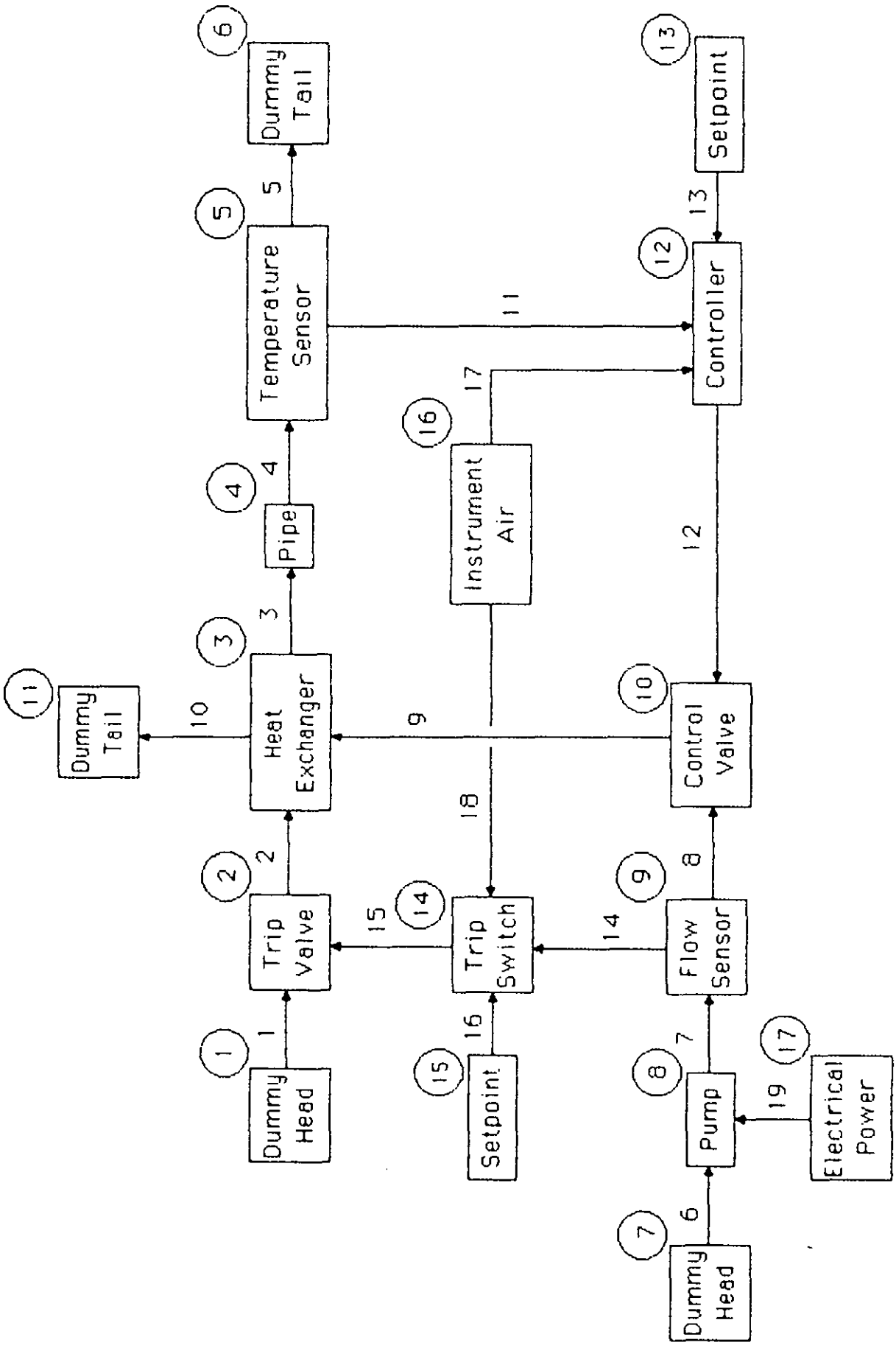

Figure 11.2 - configuration diagram for the system shown in Figure 11.1

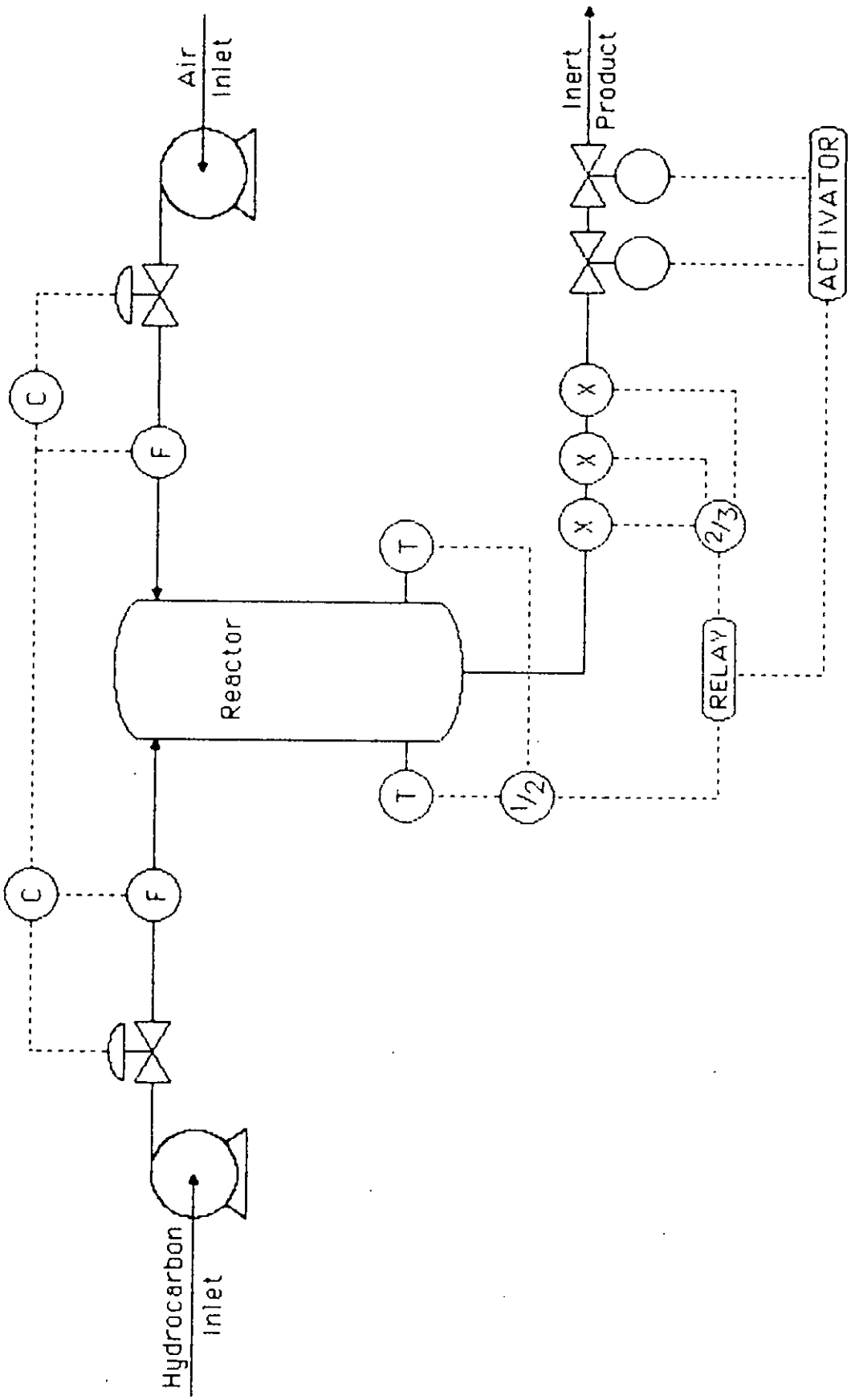

Figure 11.4 - an inert gas reactor, and associated control systems, after Lihou [48]

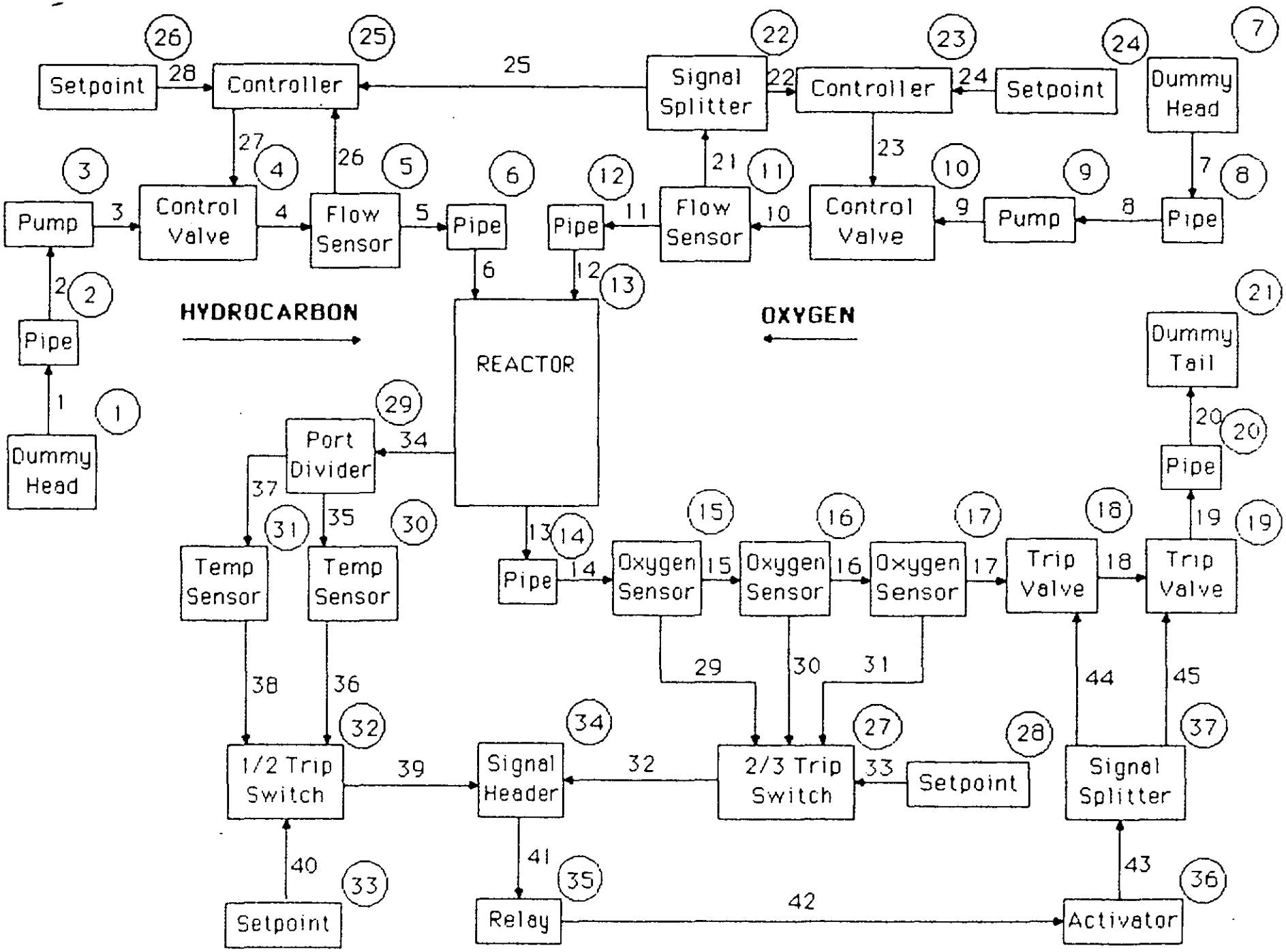

Figure 11.5 - configuration diagram for the system shown in Figure 11.4

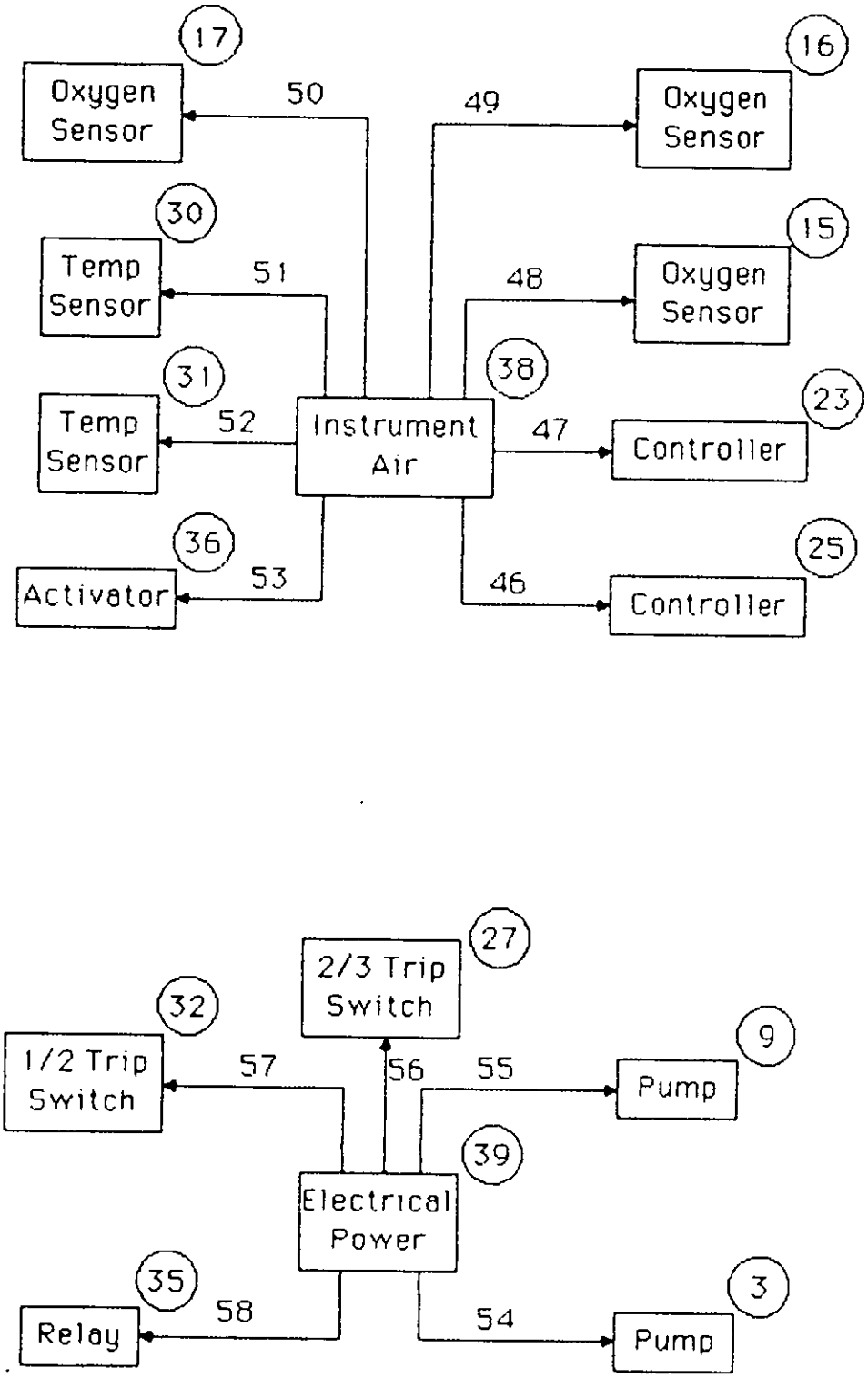

Figure 11.5 - configuration diagram for the system shown in Figure 11.4 (cont)

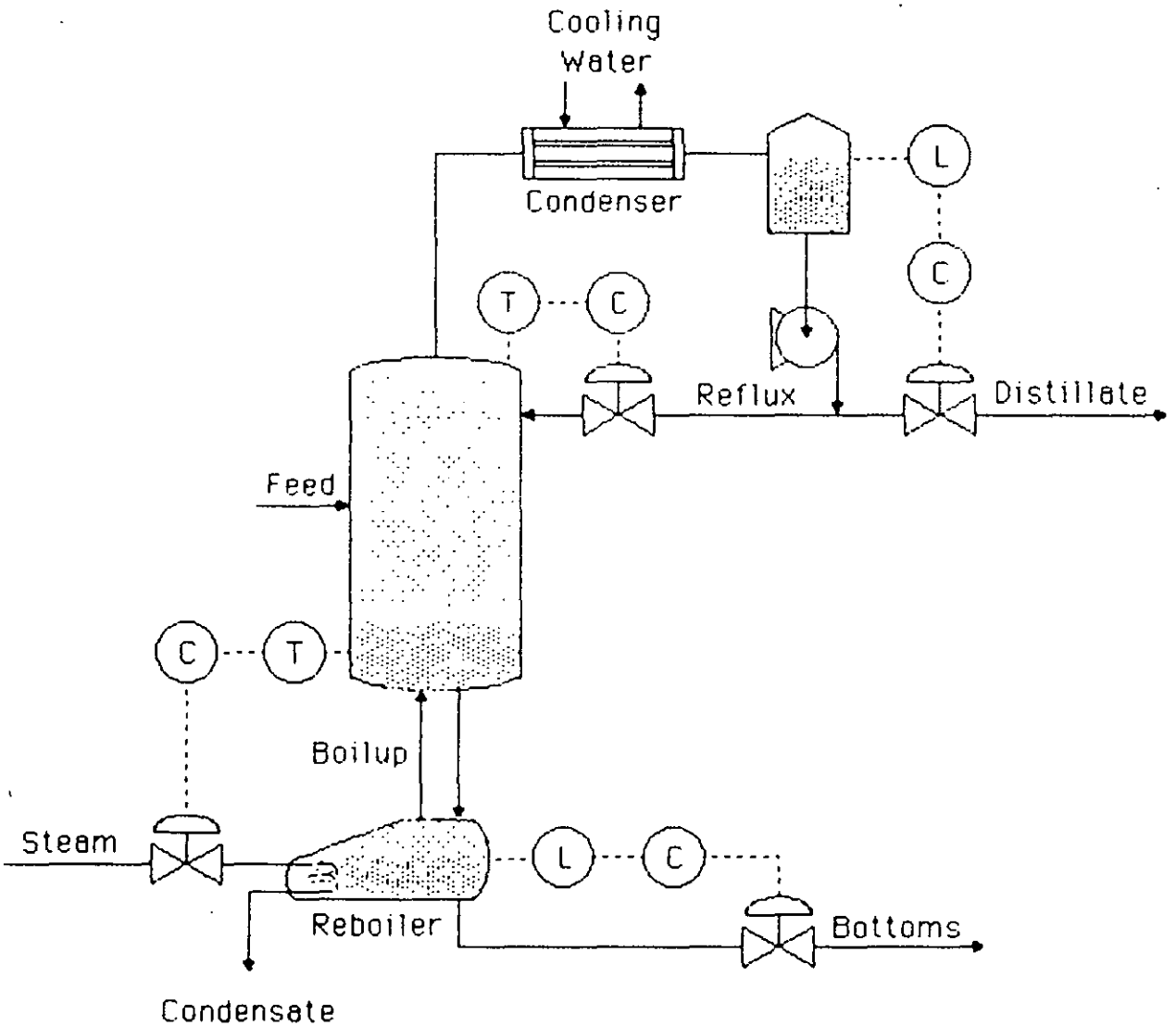

Figure 11.7 - a distillation column and associated control systems, after Shepherd et al [53]

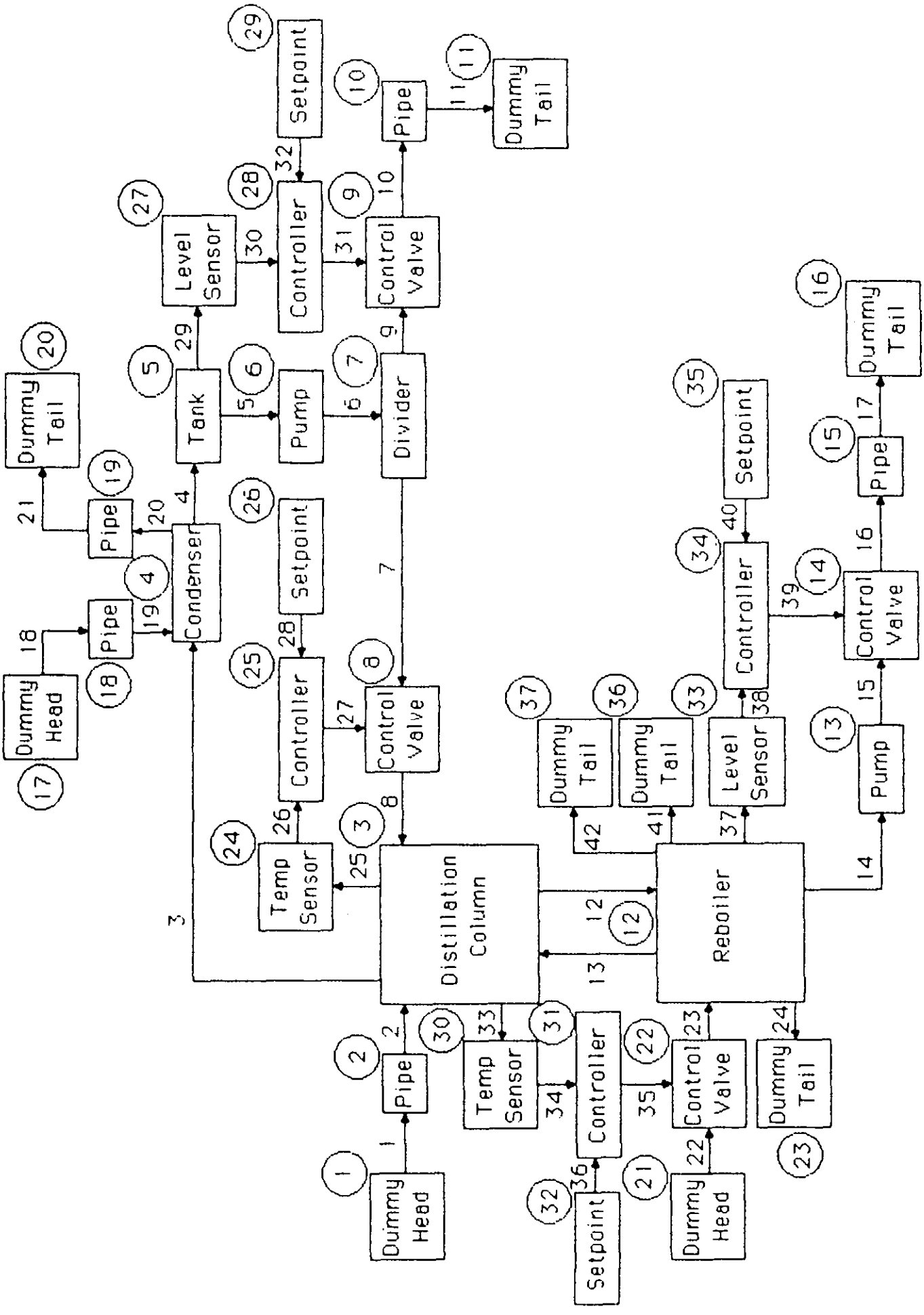

Figure 11.8 - configuration diagram for the system shown in Figure 11.7

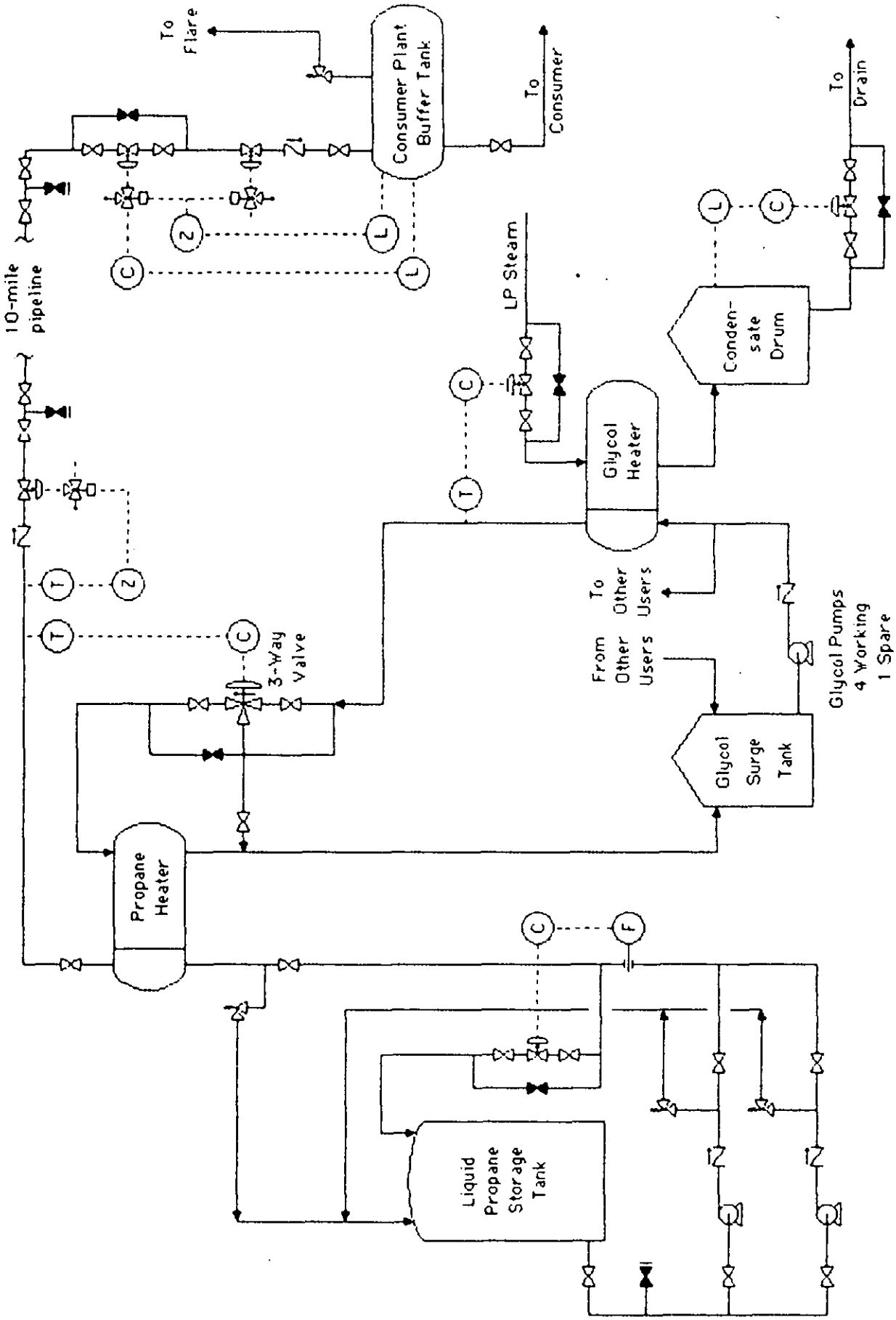

Figure 11.10 - a proposed propane transport system, after Lawley [51]

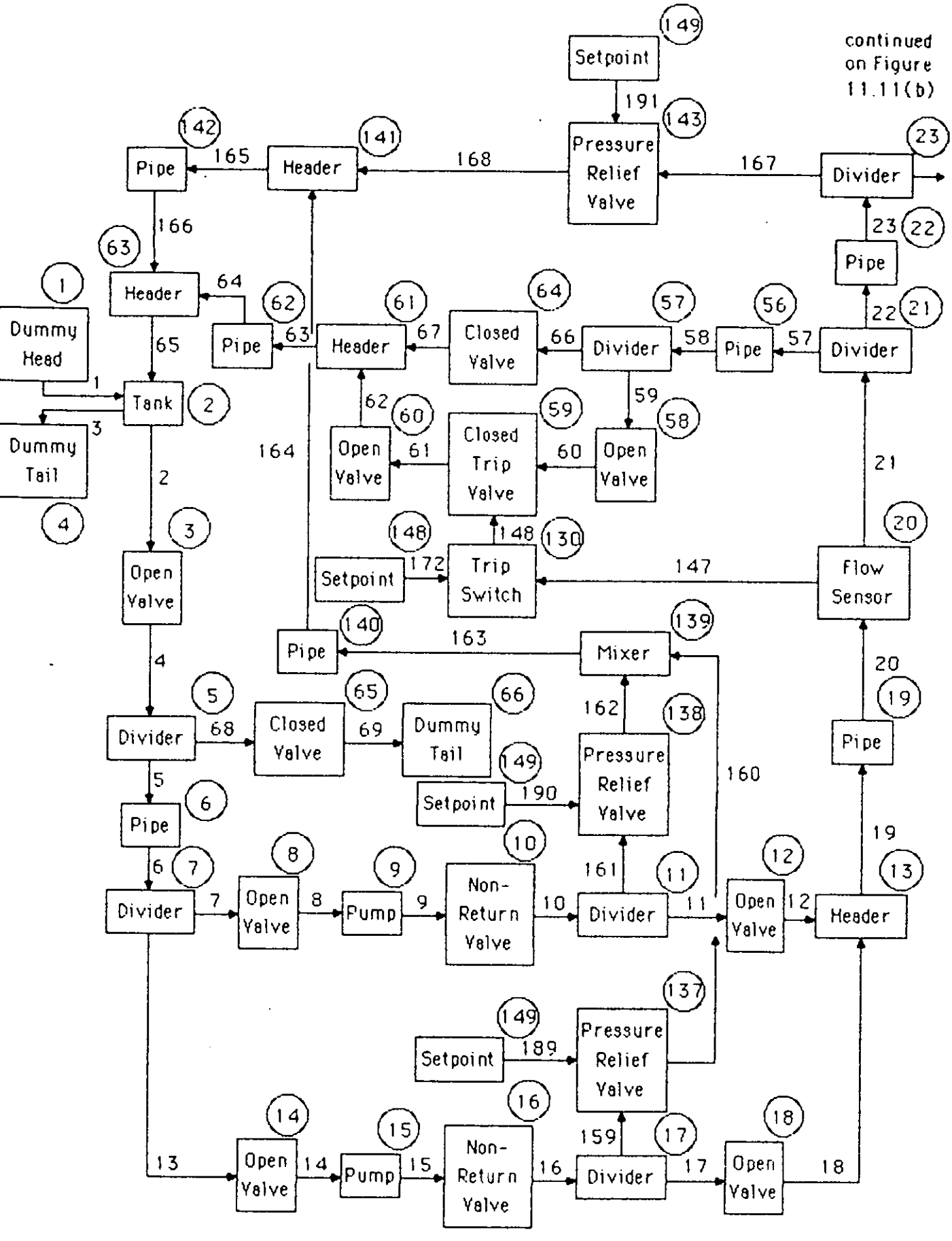

Figure 11.11(a) - configuration diagram for the system shown in Figure 11.10

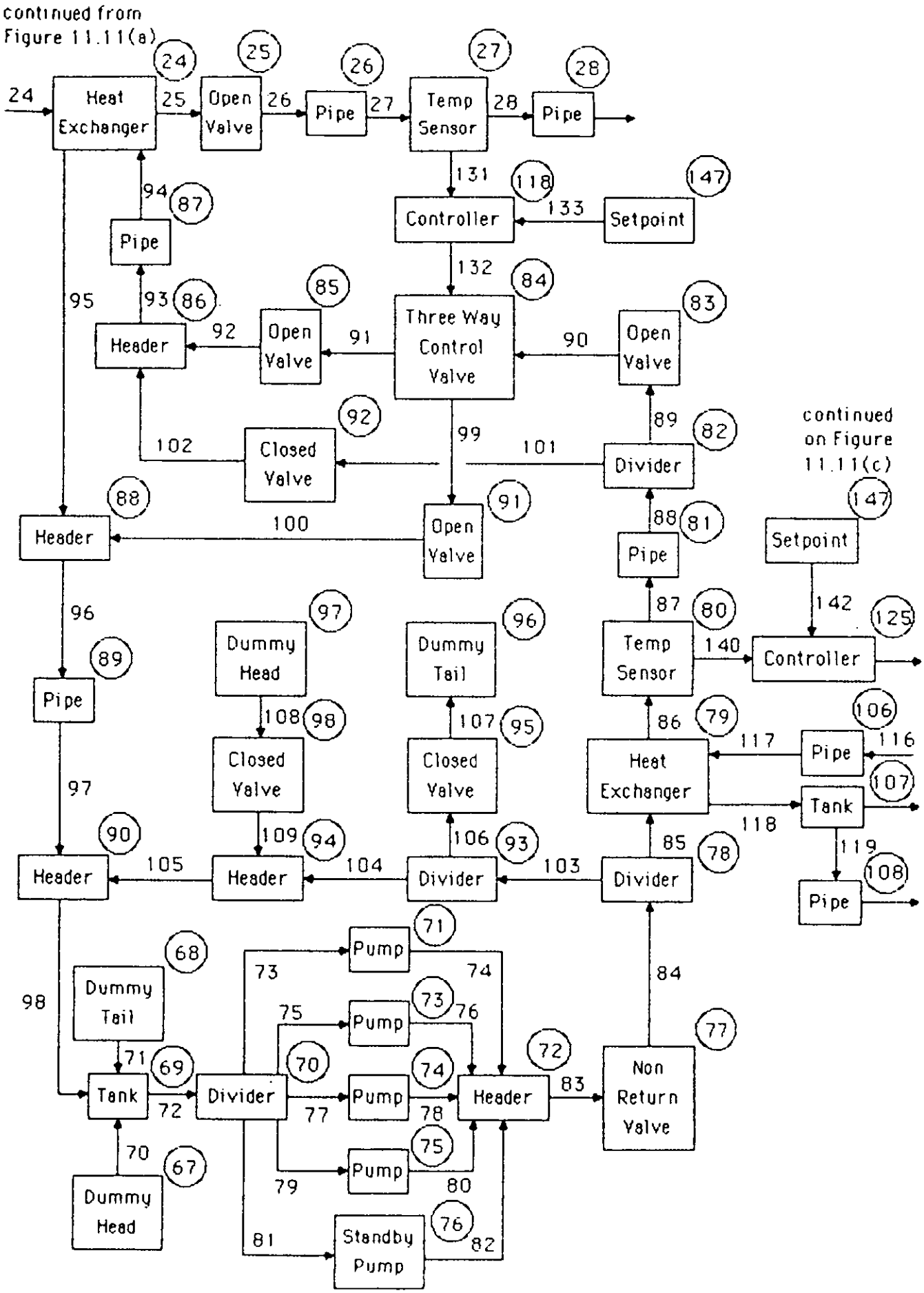

Figure 11.11(b) - configuration diagram for the system shown in Figure 11.10 (cont)

continued on Figure 11.11(b)

continued on Figure 11.11(d)

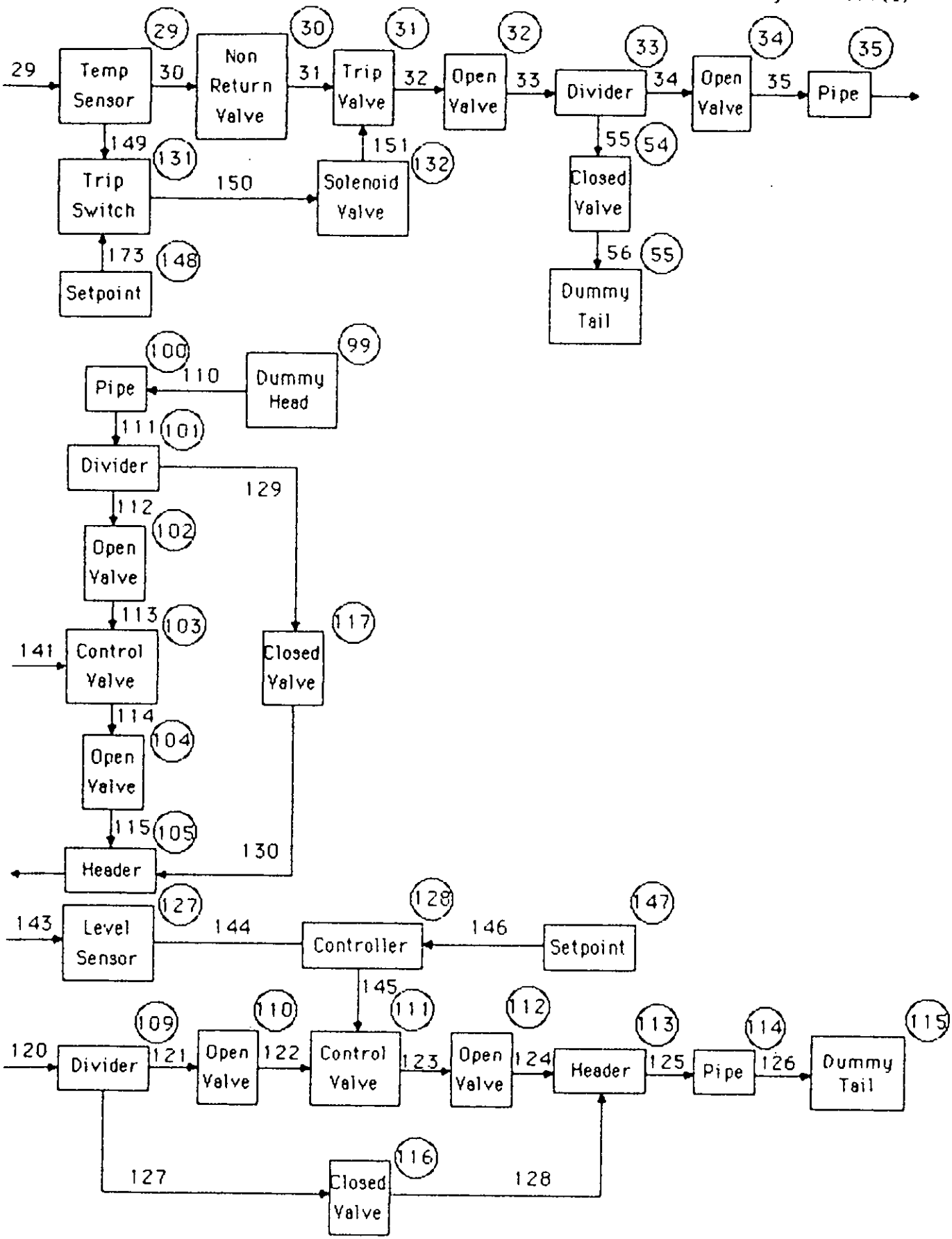

Figure 11.11(c) - configuration diagram for the system shown in Figure 11.10 (cont)

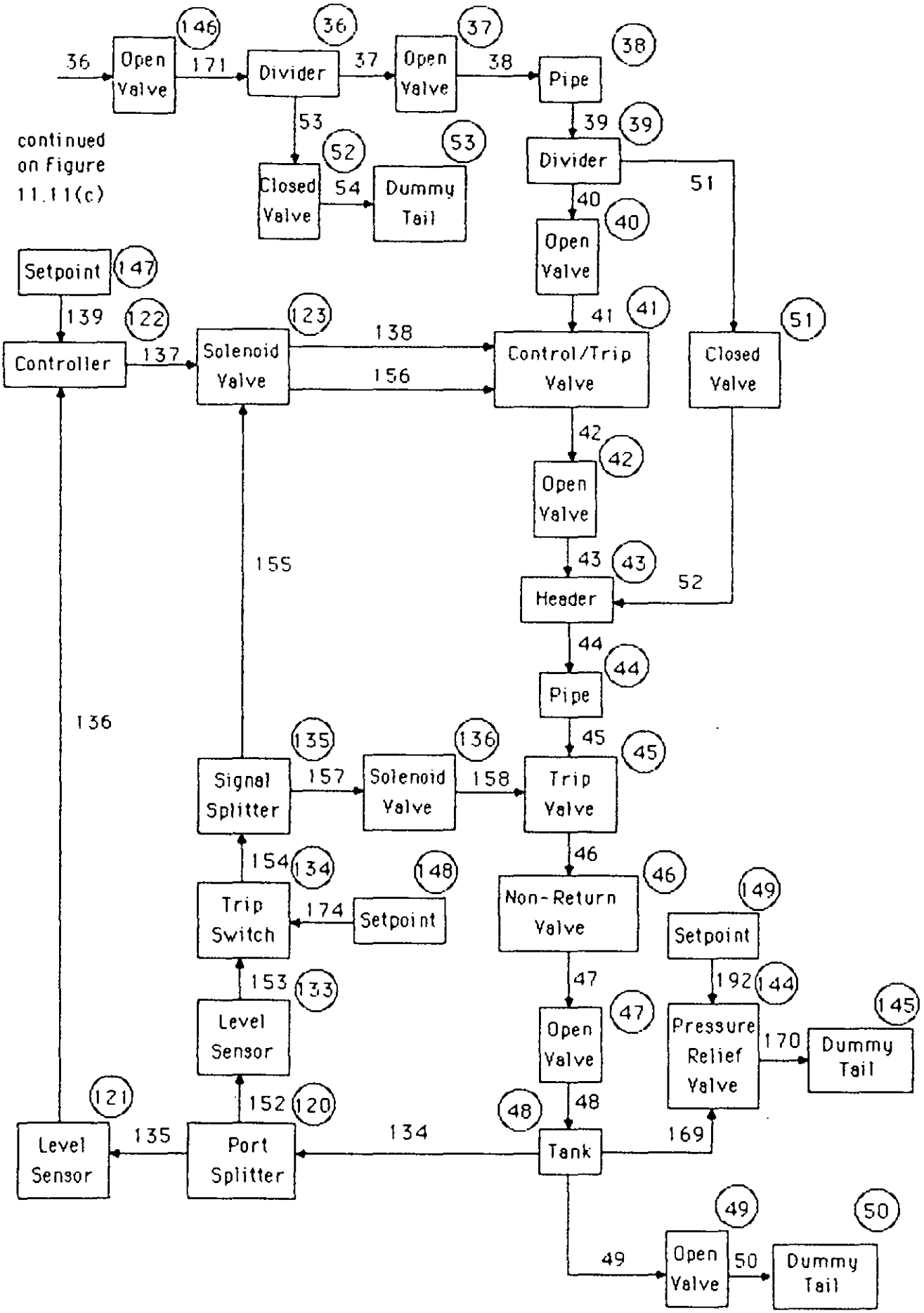

Figure 11.11(d) - configuration diagram for the system shown in Figure 11.10 (cont)

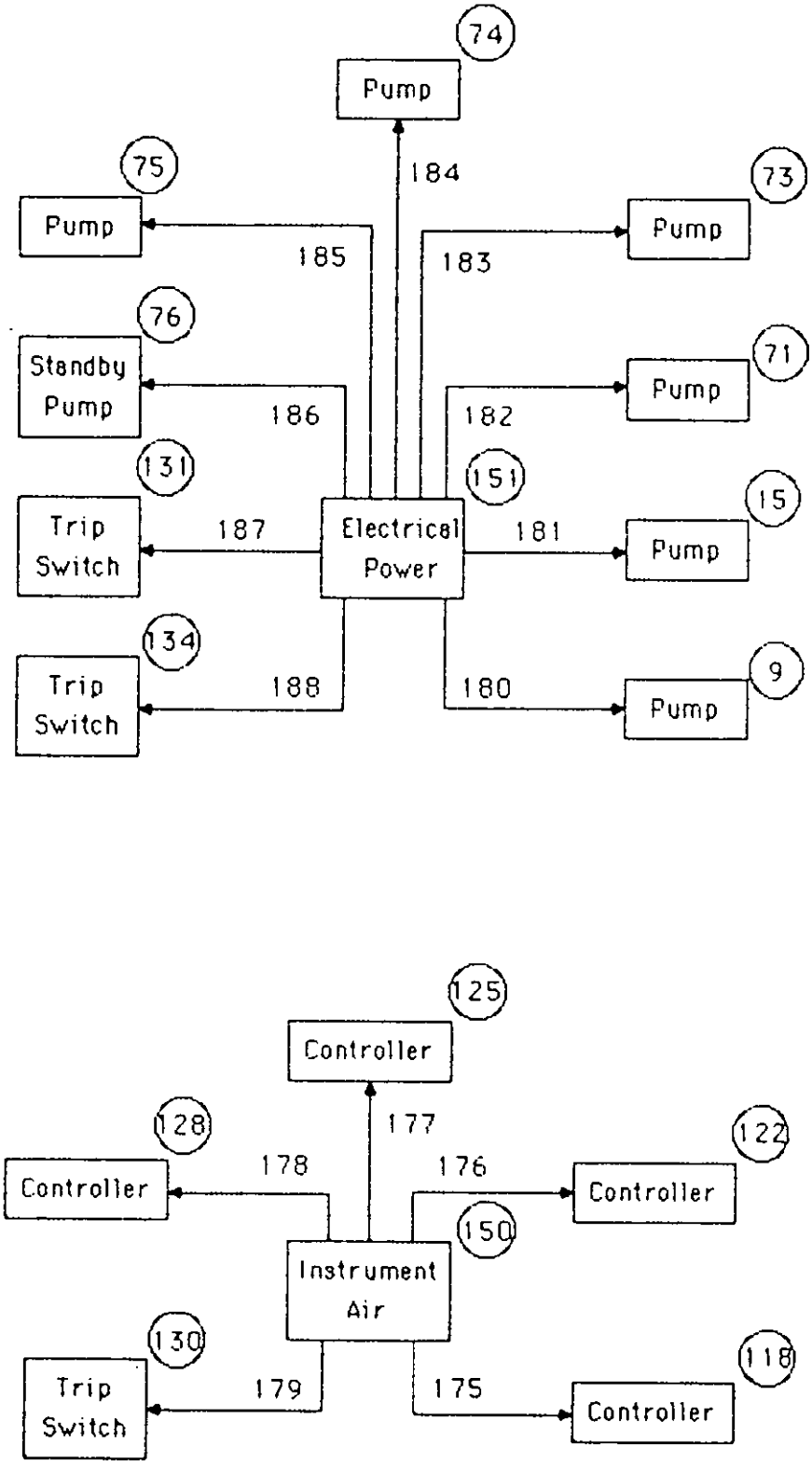

Figure 11.11(e) - configuration diagram for the system shown in Figure 11.10 (cont)

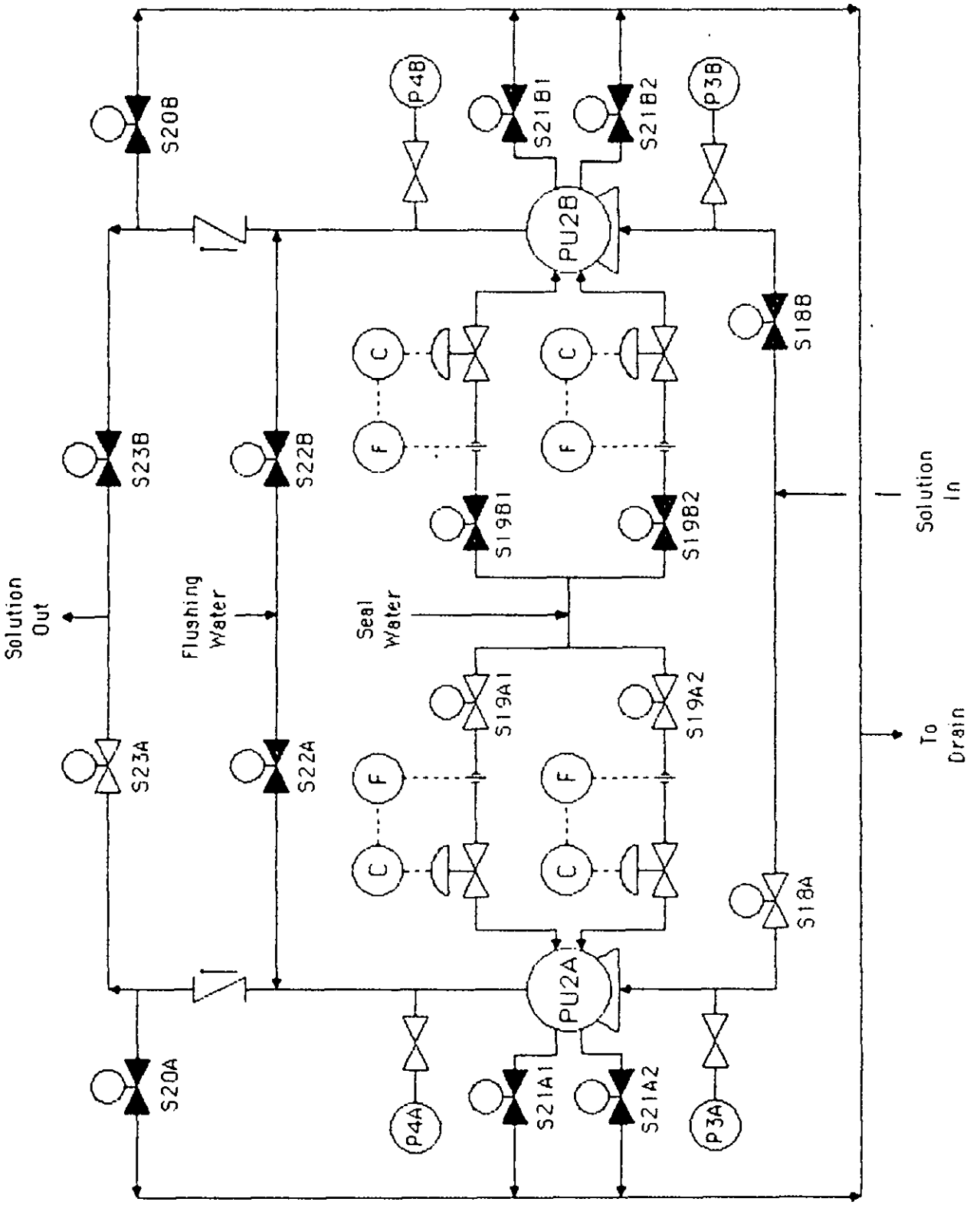

Figure 11.13 - a complex, computer-controlled pumping system

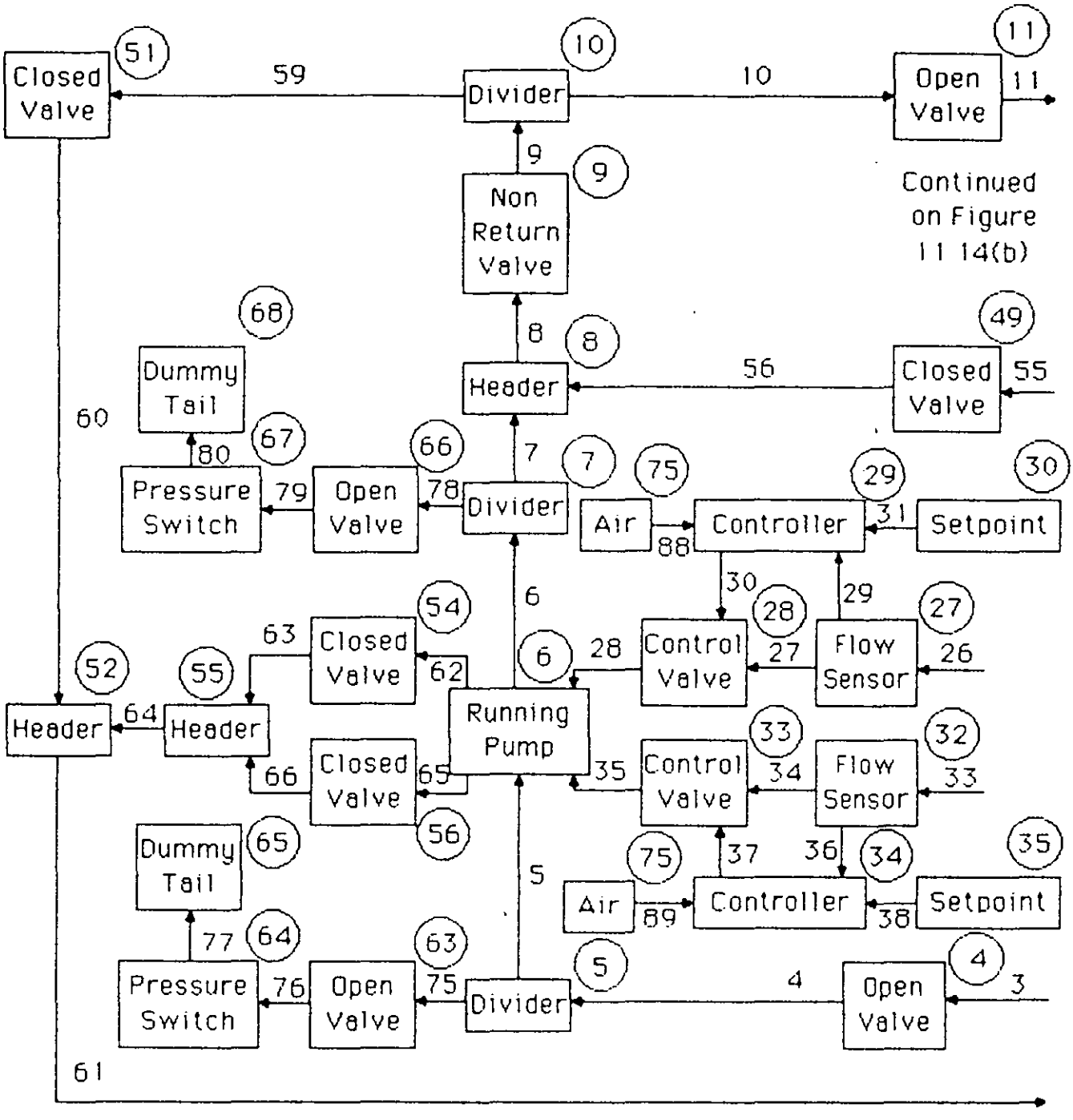

Figure 11.14(a) - configuration diagram
the system shown
in Figure 11.13

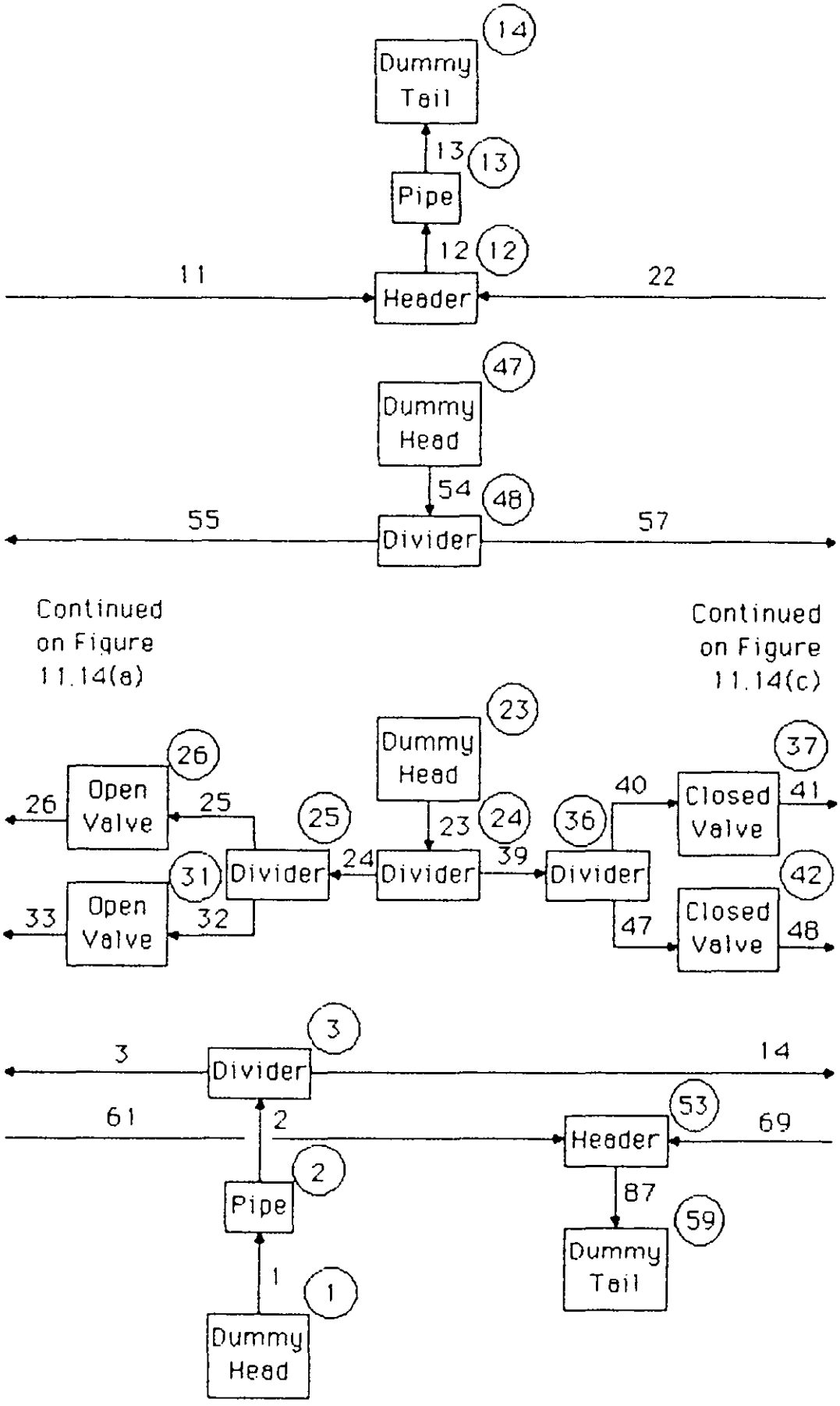

Figure 11.14(b) - configuration diagram for the system shown in Figure 11.13

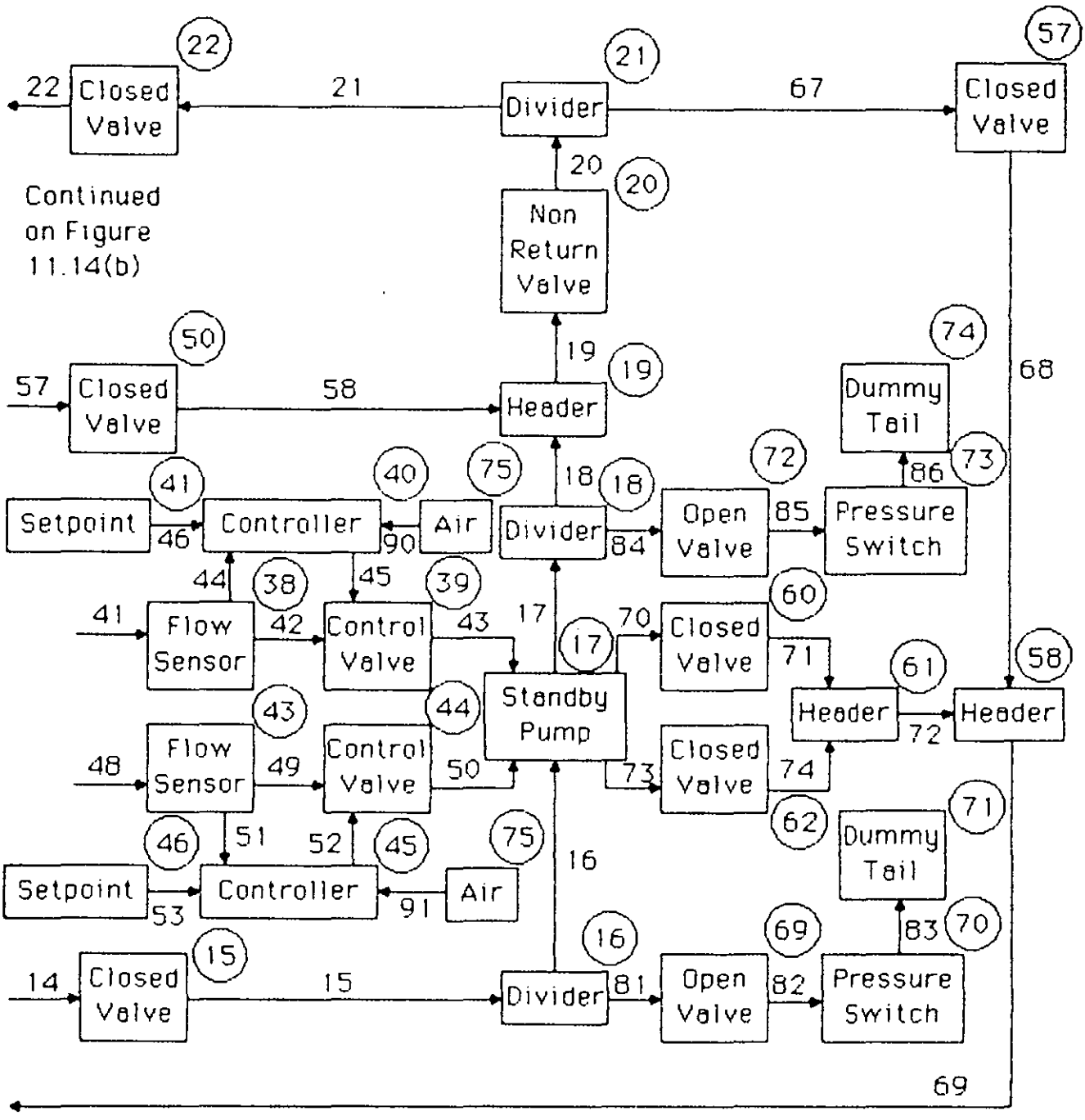

Figure 11.14(c) - configuration diagram for the system shown in Figure 11.13

1. Confirm that valves S18B, S19B1, S19B2, S20B and S23B are closed. Confirm that the supply and flushing water tanks contain sufficient level. Confirm that there is no flow of seal water to PU2B, and that there is no pressure at the inlet of PU2B.

2. Establish low pressure seal water flow to PU2B by opening valves S19B2, S21B1, S21B2 and S20B. Confirm that the valves open, and that seal water flow is established

3 Establish high pressure seal water flow to PU2B by opening valve S19B1 Confirm that the valve opens, and that seal water flow is established

4. Prime pump PU2B by opening valve S18B. Confirm that the valve opens, and that pressure exists at the inlet of PU2B.

5. Stop priming by closing valves S21B1, S21B2 and S20B. Prove the valves closed. Start pump PU2B and confirm that pressure exists at the outlet of PU2B.

6. Bring pump PU2B online by opening valve S23B Prove the valve open.

7. Take pump PU2A offline by closing valve S23A. Prove the valve closed.

Figure 11.15 - definition of sequence to perform a pump changeover for the system shown in Figure 11.13

8 Stop pump PU2A Confirm that no pressure exists at the outlet of PU2A.

9 Stop the inlet flow to PU2A by closing valve S18A Prove the valve closed.

10 Depressurise pump PU2A by opening valves S20A, S21A1 and S21A2. Prove the valves open, and confirm that the is no pressure at the inlet of PU2A

11. Start flushing pump PU2A by opening valve S22A Prove the valve open

12. Stop seal water flow to pump PU2A by closing valves S19A1 and S19A2. Prove the valves closed, and confirm that there is no seal water flow

13. Stop flushing pump PU2A by closing valve S22A. Prove the valve closed.

14. Isolate pump PU2A by closing valves S20A, S21A1 and S22A Prove the valves closed.

Figure 11.15 - definition of sequence to perform a pump changeover for the system shown in Figure 11.13 (cont)

Step 6

Unit 22 becomes Model 9 (Open Valve)
Unit 3 becomes Model 35 (Symmetrical Divider)
Unit 12 becomes Model 36 (Symmetrical Header)
Unit 21 becomes Model 21 (Unsymmetrical Divider)
Unit 21 becomes Port Swap

Sequence Aborts if

(1) TV-FT-OP Unit 22
(2) TSW-F-OF Unit 22

Figure 11.16 - definition of one step of the sequence defined in Figure 11.15, in the format required for input to the methodology

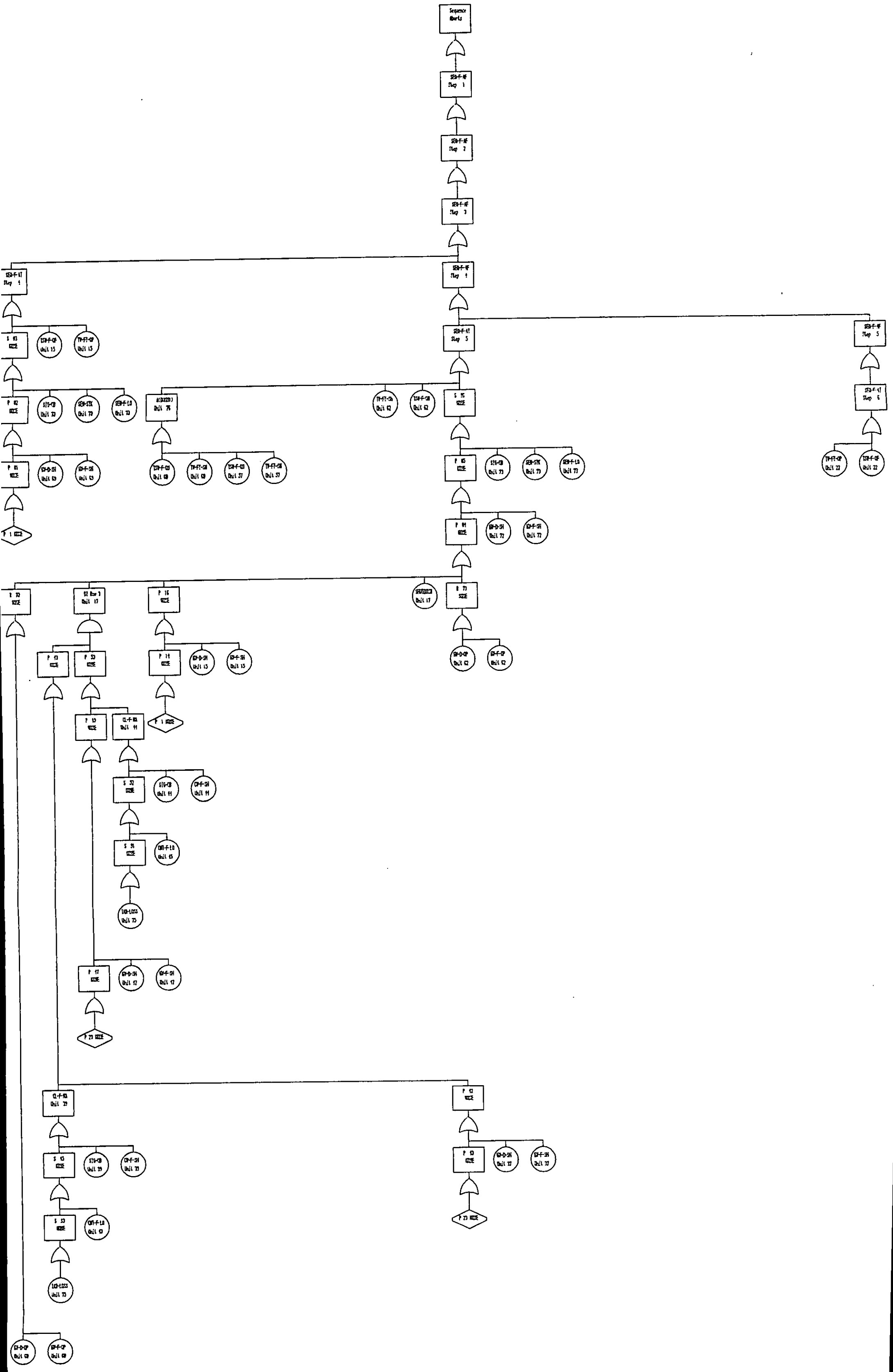

Figure 11.19 - fault tree for sequence steps 4 to 6
for the system shown in Figure 11.13

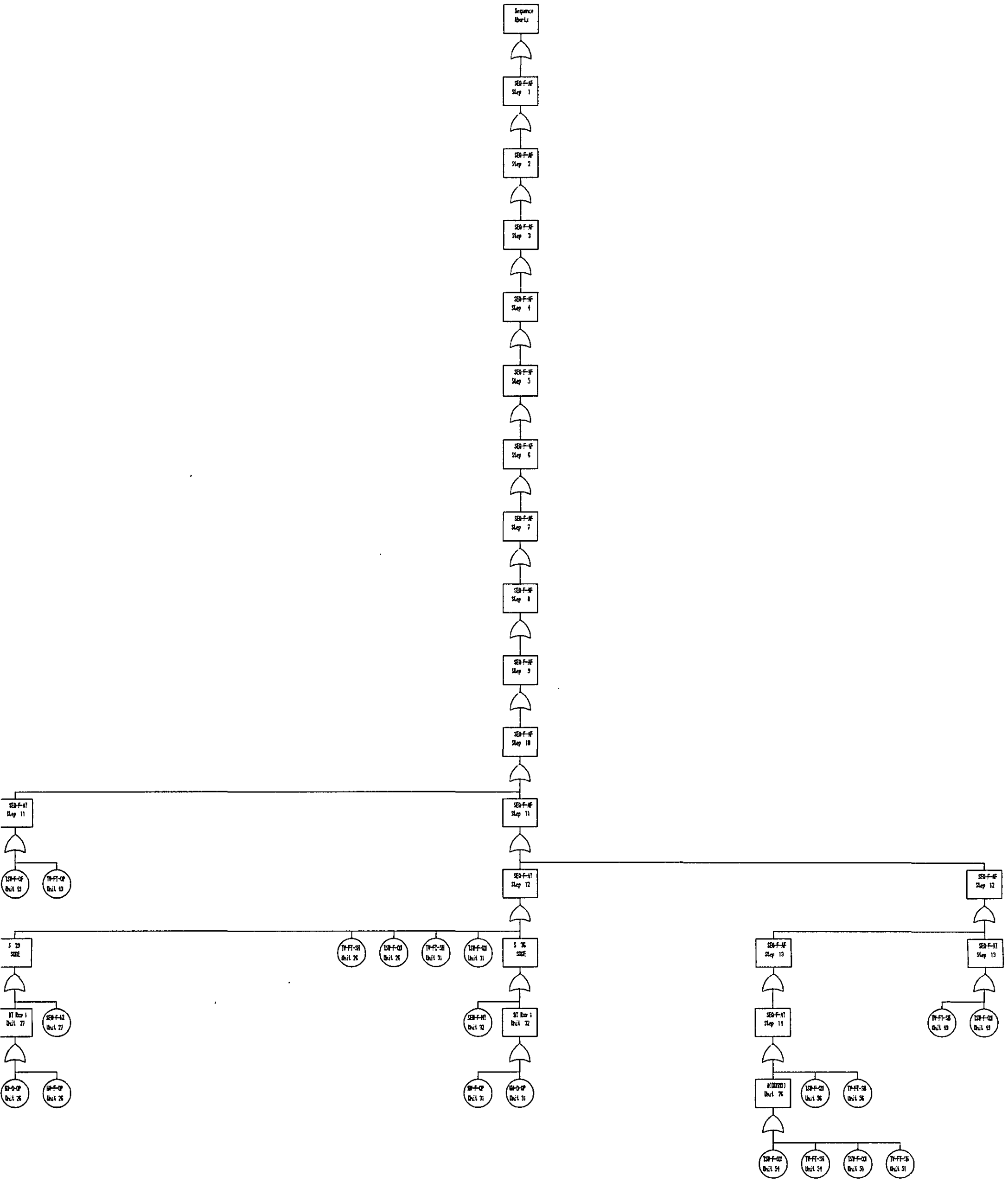

Figure 11.21 - fault tree for sequence steps 11 to 14
for the system shown in Figure 11.13

12) Discussion and Conclusions

This thesis has described research carried out at Loughborough University of Technology on the computer-aided modelling of fault conditions in process plant, in particular using the fault tree approach. The research continued from work previously done at Loughborough by Andow, Martin-Solis, Lees and Murphy [33-38]. The approach developed was to decompose a process plant section into its constituent units, and to model these units individually. The attraction of such an approach is that modelling several smaller items (the units) is easier than modelling a single larger item (the plant section). Furthermore, if modelling is done in a structured manner, the same models can be used in many different studies. Techniques were developed to use these models to synthesise fault trees and for alarm analysis.

The objective of the research described in this thesis was to expand the modelling technique to cater for a wider range of more complex, industrial scale, examples. Five complete examples were studied, as documented in Section 11, and many other smaller examples were used to test the methodology developed.

In the study of the examples, it became apparent that the approach previously developed was not adequate to cope in certain situations. Three distinct areas of work were undertaken to overcome these deficiencies

a) the modelling of additional units, to build up a library of models to cope with the examples studied

b) extension of the basic unit modelling to areas not previously covered, such as reverse flow and the modelling of vessels such as tanks and distillation columns

c) provision of a modelling level above the basic modelling level to cater for examples incorporating control loops and secondary failures

The research on these three topics is outlined briefly below.

Firstly, the library of models had to be considerably extended before the examples documented in this thesis could be considered. The library provided by the previous researchers had been created to test the methodology on a single example (the Lapp and Powers Heat Exchanger - see [23] and Section 11.1), although the units that should form the standard library had been identified. As part of the research, over fifty additional models were added to the library. Even so, not all the units that should be in the standard library have yet been modelled. The reason for this is that the approach used to model units was that a model would only be developed when it was required for use in an example. This approach has ensured that only tested models are present in the standard library.

Secondly, the basic modelling approach did not consider certain situations encountered in the examples studied as part of the research. The modelling of units was extended to permit such modelling, and the fault tree synthesis algorithm adapted to correctly synthesise fault trees for examples involving such situations. The following facilities have been added

a) reverse flow
b) the effects of reverse flow
c) pressure and relief
d) flow ratio and total component flow
e) decision table modelling
f) modelling of vessels

To model reverse flow required that the technique used to model flow be changed, as described in Sections 3.2.4 and 4.1.1. Briefly, this change involved modelling flow as a function of pressure gradient, rather than the earlier approach of modelling flow as a function of absolute pressure. However, the modelling of reverse flow is not restricted to flow alone. Reverse flow can transport material and energy from downstream, and this can have a significant impact on a fault tree. Modelling of these effects of reverse flow is described in Section 4.1.4. Pressure and relief, detailed in Section 4.1.3, was required in the study of two examples, the Lawley Propane Pipeline (see Section 11.4) and the Pump Changeover Sequence (see Section 11.5). Flow ratio and total component flow are very similar to each other, as described in Sections 4.1.5 and 4.1.6 respectively, but require special consideration during fault tree synthesis. Such treatment has proved to have a wider domain than simple flow ratio, as the Distillation Column example detailed in Section 11.3 has proved, and work is still required on this point.

The modelling of units was made more user-friendly by providing a facility to specify models using decision tables, in addition to propagation equations and event information. Decision table input is described in Section 3.2.3.3. The modelling of vessel units, as described in Section 5, has required the

careful study of the rules for modelling, to avoid models that are contradictory. More work is required on this topic.

However, most of the research concentrated on correctly synthesising fault trees for structures within process plant, such as control loops and trip systems. Five types of structure have been identified

a) control loops
b) trip systems
c) secondary failures
d) divider-header combinations
e) sequencing

One section was devoted to each of these topics. Section 6 describes control loops, Section 7 trip systems, Section 8 secondary failures, Section 9 divider-header combinations and Section 10 sequencing. The common theme running through all five is the presence of a level of modelling above the models specified for the component units of a plant section. This upper level of modelling is handled automatically during fault tree synthesis, so the analyst can concentrate solely on modelling the component units for use in a particular study.

Control loops and trip systems each require a special treatment because their intended performance cannot be deduced solely from the component models (sensors, controllers, trip switches etc.) that form the control loop or trip system. Divider-header combinations, locations where process streams are split and rejoined, such as a control valve bypass or pump bank require a special treatment for the same reason. The logic behind the special treatment is that each of

these systems behaves in a generic fashion, dependent to a limited extent on the system type. There are, for example, three behaviour modes for a control loop, depending on whether the control loop is regulating or manipulating the variable being considered, and, if manipulating it, whether the control loop is feedforward or feedback. Information defining the control loop, trip system or divider-header combination is entered as part of the information defining the plant to be studied. From then on, the special treatments are applied automatically.

Sequencing was developed to cope with examples where the state of the plant changed during a study as valves were opened and shut, and pumps turned on and off. The approach to sequencing is to synthesise a fault tree for each step in a sequence. Fault trees for a complete sequence therefore tend to be rather large, as the example of Section 11.5 illustrates.

Secondary failures are not essential to the methodology developed in the same way as the other four special treatments - it would be possible to synthesise fault trees correctly without the special techniques of secondary failures by specifying the information which secondary failures provide, i.e. where and how plant specific failure modes, such as freezing and corrosion, occur in the component models. However, this approach is inflexible, in that it would be difficult to change the secondary failures that could occur at particular points, and would lead to a large number of very similar models - different models would be required for a pipe susceptible to freezing, susceptible to corrosion, susceptible to neither and susceptible to both. The approach of secondary failures is to provide the plant specific failure modes independently of the

component models. the component models therefore remain much more general, and so of use in a much larger number of applications.

In summary, the research described in this thesis has extended a computer-aided fault tree synthesis technique to cater for studies of plants involving a wide range of components, control and protective systems, and plant specific failures. As has been highlighted during the thesis, there are a few situations that the current technique cannot yet cope with correctly, which further work should be able to cure. The thesis has concentrated on the logic behind the fault tree synthesis technique, rather than the computer programs developed to implement the technique. Although, inevitably, some of the logic is dependent on the modelling method used, some, such as the specification of the generic behaviour of control loops, is independent, and therefore of possible application in other computer-aided, or even manual, fault tree synthesis methodologies.

The end result is a methodology for modelling chemical process plant, and an application that can synthesise fault trees. As it stands, the application processes the input (comprising the component models, the plant layout and information such as the intended performance of control loops) and synthesises fault trees. The research could be extended in two directions. Firstly, other applications based on the modelling methodology could be developed, for instance event tree and cause consequence diagram synthesis. Secondly, the fault tree synthesis application itself could be extended, for instance by assisting the analyst to identify common mode failures. One extension

has already been developed - the output of the application has been extended to include information about the fault tree structure in a format that permits the PREP/KITT fault tree analysis package to calculate minimum cutsets and various quantitative results.

References

1. System Safety Symposium, Seattle, Wash.; The Boeing Company (1965).

2. R.E. Barlow, J.B. Fussell, N.D. Singpurwalla (eds); Reliability and Fault Tree Analysis; SIAM, Philadelphia (1975).

3. H.E. Lambert; Measures of Importance of Events and Cut Sets; in reference [2] pp77-100.

4. W.E. Vesely; A Time-Dependent Methodology for Fault Tree Evaluation; Nuclear Engineering and Design, 13 pp337-360 (1970).

5. J.S. Arendt, J.B. Fussell; System Reliability Engineering Methodology for Industrial Application; Loss Prevention, 14, pp18-28; AIChE 1981.

6. PRA Procedures Guide : A Guide to the Performance of Probabilistic Risk Assessment for Nuclear Power Plants; NUREG/CR-2300 Vol. 1 Rev. 1, pp6.32-6.68 (April 1982).

7. J.B. Fussell, G.J. Powers, R.G. Bennetts; Fault Trees - A State of the Art Discussion; IEEE Transactions on Reliability, R-23 pp51-55 (1974).

8. R.A. Evans; Editorial : Automatic Fault Tree Generation - What & Why; IEEE Transactions on Reliability, R-27 p241 (1978).

9. D.F. Haasl; Advanced Concepts in Fault Tree Analysis; in reference [1] pp1-12.

10. J.B. Fussell; A Formal Methodology for Fault Tree Construction; Nuclear Science and Engineering, 52 pp421-432 (1973).

11. J.B. Fussell; Computer Aided Fault Tree Construction for Electrical Systems; in reference [2] pp37-56.

12. J.S. Brown, J. De Kleer; Towards a Theory of Qualitative Reasoning about Mechanisms and its Role in Trouble Shooting; in J. Rasmussen, W.B. Rouse (eds); Human Detection and Diagnosis of System Failures; NATO Conference Series III: Human Factors (Plenum, 1980).

13. J.B. Fussell, G.R. Burdick (eds); Nuclear Systems Reliability Engineering and Risk Assessment; SIAM, Philadelphia (1977).

14. G. Apostolakis, S.Garriba, G. Volta (eds); Synthesis and Analysis Methods for Safety and Reliability Studies; Plenum (1980).

15. S.L. Salem, G.E. Apostolakis, D. Okrent; A New Methodology for the Computer-Aided Construction of Fault Trees; Annals of Nuclear Energy, 4 pp417-433 (1977).

16. J.S. Wu, S.L. Salem, G.E. Apostolakis; The Use of Decision Tables in the Systematic Construction of Fault Trees; in reference [13] pp800-824.

17. S.L. Salem, J.S. Wu, G.E. Apostolakis; Decision Table Development and Application to the Construction of Fault Trees; Nuclear Technology, 42 pp51-64 (1979).

18. S.L. Salem, G.E. Apostolakis; The CAT Methodology for Fault Tree Construction; in reference [14] pp109-128.

19. G.J. Powers, F.C. Tompkins; Computer-Aided Fault Tree Synthesis for Chemical Processing Systems; AIChE Journal; 20 pp376-387 (1974).

20. G.J. Powers, F.C. Tompkins; A Synthesis Strategy for Fault Trees in Chemical Processing Systems; Loss Prevention 8 pp91-99, CEP Technical Manual, AIChE, (1974).

21. G.J. Powers, F.C. Tompkins, S.A. Lapp; A Safety Simulation Language for Chemical Processes : A Procedure for Fault Tree Synthesis; in reference [2] pp57-75.

22. G.J Powers, S.A. Lapp; Computer Aided Fault Tree Synthesis; Chemical Engineering Progress, 72 pp89-93 (1976).

23. S.A. Lapp, G.J. Powers; Computer-Aided Synthesis of Fault Trees; IEEE Transactions on Reliability, R-26 pp2-13 (1977).

24. S.A. Lapp, G.J. Powers; The Synthesis of Fault Trees; in reference [14] pp778-799.

25. E.J. Henley, H. Kumamoto; Comment on : Computer-aided Synthesis of Fault Trees; IEEE Transactions on Reliability, R-26 pp316-317 (1977).

26. M.O. Locks; Synthesis of Fault Trees : An Example of Noncoherence; IEEE Transactions on Reliability, R-28 pp2-5 (1979).

27. H.E. Lambert; Comments on the Lapp-Powers 'Computer-Aided Synthesis of Fault Trees'; IEEE Transactions on Reliability, R-28 pp6-9 (1979).

28. T.W. Yellman; Comment on : "Comment on : Computer-aided Synthesis of Fault Trees"; IEEE Transactions on Reliability, R-28 pp10-11 (1979).

29. S.A. Lapp, G.J. Powers; Update of Lapp-Powers Fault Tree Synthesis Algorithm; IEEE Transactions on Reliability, R-29 p12-15 (1979).

30. M.O. Locks; The Fail-Safe Feature of the Lapp-Powers Fault Tree; IEEE Transactions on Reliability, R-29 pp10-11 (1979).

31. J.A. Shaeiwitz, S.A. Lapp, G.J. Powers; Fault Tree Analysis of Sequential Systems; Industrial and Engineering Chemistry Process Design and Development, 16 pp529-549 (1977).

32. D.L Cummings, S.A. Lapp, G.J. Powers; Fault Tree Synthesis from a Directed Graph Model for a Power Distribution Network; IEEE Transactions on Reliability, R-32 pp140-149 (1983).

33. P.K. Andow; A Method for Process Computer Alarm Analysis; Ph. D. Thesis, Loughborough University of Technology (1973).

34. P.K. Andow, F.P. Lees; Process Computer Alarm Analysis : Outline of a Method Based on List Processing; Transactions of the Institute of Chemical Engineers, 53 pp195-208 (1975).

35. G.A. Martin-Solis; Fault Tree Synthesis for Real Time and Design Applications on Process Plant; Ph. D. Thesis, Loughborough University of Technology (1978).

36. G.A. Martin-Solis, P.K. Andow, F.P. Lees; Fault Tree Synthesis for Design and Real Time Applications; Transactions of the Institute of Chemical Engineers, 60 pp14-25 (1982).

37. G.A. Martin-Solis, P.K. Andow, F.P. Lees; An Approach to Fault Tree Synthesis for Process Plant; in Second International Symposium on Loss Prevention and Safety Prevention in the Process Industries; Heidelberg (1977).

38. F.P. Lees, P.K. Andow, C.P. Murphy; The Propagation of Faults in Process Plants : A Review of the Basic Event/Fault Information; Reliability Engineering, 1 pp149-163 (1980).

39. A. Shafaghi; Plant Modelling for Systems Safety Analysis; Ph. D. Thesis, Loughborough University of Technology (1982).

40. A. Shafaghi, P.K. Andow, F.P. Lees; Fault Tree Synthesis based on Control Loop Structure; Transactions of the Institute of Chemical Engineers, 62 pp101 (1984).

41. A.Shafaghi, F.P. Lees, P.K. Andow; An Illustrative Example of Fault Tree Synthesis Based on Control Loop Structure; Reliability Engineering, 8 pp193-223 (1984).

42. J.R. Taylor; An Algorithm for Fault Tree Construction; IEEE Transactions on Reliability, R-31 pp137-146 (1982).

43. J.R. Taylor; Fault Tree and Cause Consequence Diagram Construction - A Compendium of Examples, Volume I. Riso-M-2307 (1981).

44. J.R. Taylor; Fault Tree and Cause Consequence Diagram Construction - A Compendium of Examples, Volume II.

45. G. Reina, G. Squellati; L.A.M. Techniques : Systematic Generation of Logical Structures in Systems Reliability Studies; in reference [14] pp129-181.

46. S. Caceres, E.J. Henley; Process Failure Analysis by Block Diagrams and Fault Trees, Industrial and Engineering Chemistry Fundamentals, 15 pp128-134 (1976).

47. P. Camarda, F. Corsi, A. Trentadue; An Efficient Simple Algorithm for Fault Tree Automatic Synthesis from the Reliability Graph, IEEE Transactions on Reliability, R-27 pp215-221 (1978).

48. D. Lihou; Efficient Use of Operability Studies; Safety Promotion and Loss Prevention in the Process Industries, Oyez, London (1980).

49. D. Lihou; Fault Trees from Operability Studies; Safety Promotion and Loss Prevention in the Process Industries, Oyez, London (1980).

50. R.L. Williams, W.Y. Gately; Use of the GO Methodology to Directly Generate Minimal Cutsets; in reference [13] pp825-849.

51. H.G. Lawley; Safety Technology in the Chemical Industry : A Problem in Hazard Analysis with Solution, Reliability Engineering, 1 pp89-113 (1980).

52. J.M. Coulson, J.F. Richardson; Chemical Engineering, Volume I (3rd Edition), p41, Pergammon (1978).

53. A. Shepherd, E.C. Marshall, A. Turner, K.D. Duncan; Diagnosis of Plant Failures from a Control Panel : A Comparison of Three Training Methods, Ergonomics, 20 pp347-361 (1977).

54. B.E. Kelly, F.P. Lees; The Propagation of Faults in Process Plants: 1. Modelling of Fault Propagation, Reliability Engineering 16 pp3-38 (1986).

55. B.E. Kelly, F.P. Lees; The Propagation of Faults in Process Plants: 2. Fault Tree Synthesis, Reliability Engineering 16 pp39-62 (1986).

56. B.E. Kelly, F.P. Lees; The Propagation of Faults in Process Plants: 3. An Interactive, Computer-Based Facility, Reliability Engineering 16 pp63-86 (1986).

57. B.E. Kelly, F.P. Lees; The Propagation of Faults in Process Plants: 4. Fault Tree Synthesis of a Pump System Changeover Sequence, Reliability Engineering 16 pp87-108 (1986).

Appendix A

This Appendix continues from Section 3.4.2.2, which considered the modelling of a counter current heat exchanger. A complex expression for T4out was obtained, which required differentation to calculate the propagation equation for T4out. This Appendix details these calculations.

The expression for T4out obtained was

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

Differentiating first with respect to T1in gives

$$\frac{dT4out}{dT1in} = \frac{2*U*A*Q1in*Cp1}{2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 + Q1in*Cp1*U*A}$$

This is always positive, so T4out increases as T1in increases.

Differentiating next with respect to T3in gives

$$\frac{dT4out}{dT3in} = \frac{2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 - U*A*Q1in*Cp1}{2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 + Q1in*Cp1*U*A}$$

The denominator is always positive, so the differential will have the same sign as the numerator. Remembering the heat transfer equations, Q1*Cp1 can be

replaced by E/(T1in-T2in), Q3in*T3in can be replaced by E/(T4out-T3in), and U*A by 2*E/((T1in-T4out)+(T2out-T3in)), E being the rate of heat transfer. This gives a numerator of

$$\frac{2*E^2}{(T1in-T2out)(T4out-T3in)}$$

$$+ \frac{2*E^2}{((T1in-T4out)+(T2out-T3in))(T4out-T3in)}$$

$$- \frac{2*E^2}{((T1in-T4out)+(T2out-T3in))(T1in-T2out)}$$

Multiplying this to form an expression with a single denominator gives

$$\frac{2*E^2*(((T1in-T4out)+(T2out-T3in)) + (T1in-T2out) - (T4out-T3in))}{(T1in+T2out-T3in-T4out) * (T1in-T2out) * (T4out-T3in)}$$

The numerator of this expression simplifies to

$$4*E*E*(T1in-T4out)$$

This is always positive, since the nitric acid enters the heat exchanger at a higher temperature than the cooling water leaves. Since the denominator of the above expression is always positive, the differential itself is also positive. Therefore, T4out will increase as T3in increases.

Differentiating next with respect to Q1in gives

$$\frac{dT4out}{dQ1in} = \frac{\begin{array}{l}(2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 + \\ \qquad Q1in*Cp1*U*A) * \\ (2*U*A*Cp1*T1in + 2*Cp1*Q3in*Cp3*T3in - \\ \qquad U*A*Cp1*T3in) - \\ (2*U*A*Q1in*Cp1*T1in + 2*Q1in*Cp1*Q3in*Cp3*T3in + \\ \qquad U*A*Q3in*Cp3*T3in - U*A*Q1in*Cp1*T3in) * \\ (2*Cp1*Q3in*Cp3 + Cp1*U*A)\end{array}}{\begin{array}{l}(2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 + \\ \qquad Q1in*Cp1*U*A)\end{array}}$$

Considering only the numerator, and grouping gives

$$\begin{array}{l}U*A*Q3in*Cp3*(2*U*A*Cp1*T1in + 2*Cp1*Q3in*Cp3*T3in - \\ \qquad U*A*Cp1*T3in) \\ \quad - U*A*Q3in*Cp3*T3in*(2*Cp1*Q3in*Cp3 + Cp1*U*A)\end{array}$$

or,

$$2*U^2*A^2*Q3in*Cp3*Cp1*(T1in - T3in)$$

This is always positive, since T1in (the hot nitric acid inlet temperature) is always larger than T3in (the cooling water inlet temperature). Since both the numerator and the denominator are positive, the differential is therefore positive, and so T4out increases as Q1in increases.

Finally, differentating next with respect to Q3in gives

$$\frac{dT4out}{dQ3in} = \frac{\begin{array}{l}(2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 + \\ \qquad Q1in*Cp1*U*A) * \\ \quad (2*Q1in*Cp1*Cp3*T3in + U*A*Cp3*T3in) - \\ (2*U*A*Q1in*Cp1*T1in + 2*Q1in*Cp1*Q3in*Cp3*T3in + \\ \quad U*A*Q3in*Cp3*T3in - U*A*Q1in*Cp1*T3in) * \\ \quad (2*Q1in*Cp1*Cp3 + Cp3*U*A)\end{array}}{\begin{array}{l}(2*Q1in*Cp1*Q3in*Cp3 + U*A*Q3in*Cp3 + \\ \qquad Q1in*Cp1*U*A)\end{array}}$$

Considering only the numerator, and grouping gives

$$U*A*Q1in*Cp1*(U*A*Cp3*T3in + 2*Q1in*Cp1*Cp3*T3in) - (2*U*A*Q1in*Cp1*T1in - U*A*Q1in*Cp1*T3in)* (2*Q1in*Cp1*Cp3 + Cp3*U*A)$$

rearranging,

$$2*U^2*A^2*Q1in*Cp1*Cp3*T3in + 4*U*A*Q1in2*Cp12*Cp3*T3in - 2*U*A*Q1in*Cp1*T1in*(2*Q1in*Cp1*Cp3 + U*A*Cp3)$$

or,

$$2*U*A*Q1in*Cp1*Cp3*(2*Q1in*Cp1 + U*A)*(T3in - T1in)$$

This is always negative, since T1in (the hot nitric acid inlet temperature) is always larger than T3in (the cooling water inlet temperature). Since both the numerator is negative and the denominator is positive, the differential is therefore negative, and so T4out decreases as Q3in increases.

In summary, differentiating the expression for T4out has shown that T4out will increase as T1in increases, T3in increases, Q1in increases and Q3in decreases.