2/
6-27-78

ANL/EES-CP-3

CONF-780656--1

INTRODUCTION TO FAULT TREE SYNTHESIS USING THE LAPP-POWERS METHODOLOGY

Edward P. Lynch

U of C-AUA-USDOE

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

Operated under Contract W-31-109-Eng-38 for the
U. S. DEPARTMENT OF ENERGY

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Department of Energy, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights. Mention of commercial products, their manufacturers, or their suppliers in this publication does not imply or connote approval or disapproval of the product by Argonne National Laboratory or the U. S. Department of Energy.

ANL/EES-CP-3

INTRODUCTION TO FAULT TREE SYNTHESIS USING THE LAPP-POWERS METHODOLOGY

by

Edward P. Lynch

Energy and Environmental Systems Division
Argonne National Laboratory
Argonne, Illinois 60439

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Paper Presented at

1978 Symposium on Instrumentation and Control
for Fossil Demonstration

Sponsored by

Argonne National Laboratory, U.S. Dept. of Energy, and
Calif. Chapter Instrument Society of America

Newport Beach, California
June 19-21, 1978

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

INTRODUCTION TO FAULT TREE SYNTHESIS USING THE LAPP-POWERS METHODOLOGY

By

Edward P. Lynch, Argonne National Laboratory

In the design of any complex system the question of reliability of equipment and instrumentation arises. A large, single train chemical plant--such as a high tonnage ammonia plant--is more economical to build and operate than a plant of the same capacity using smaller multiple trains. However, equipment or instrument failure which causes a shutdown will result in a much greater economic loss than the failure of one train of a multiple train plant. The trend is toward the high capacity single train plant but the question 'what if---?' keeps coming up. There are two approaches to answering this question, both of which have been used successfully in the aerospace and electronics industries. These are Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA). It is only quite recently that these methods have been applied to the chemical industry.

Failure Modes and Effects Analysis (FMEA) is a formalized method for asking the question 'what if---?'. All of the possible component failures such as valves leaking, pump couplings or shafts breaking, line blockages, instrument failure, operator error, etc., are hypothesized and possible effects on the system are determined by investigating the system response to each failure or combination of failures. This can be done quite readily with a digital computer providing an adequate model of the system is available. The key word is 'adequate'. Rarely will a model adequate enough for FMEA exist for a chemical plant, particularly in the design stage where FMEA is most useful. Changes in process conditions, process constraints, physical constraints, etc., make the process flow diagrams ever-changing documents--sometimes to the point where one thinks that order will never be brought out of the chaos.

Fault Tree Analysis is a method of determining the possibility and/or probability of a specific designated failure occurring. A complete logic diagram is constructed which identifies the immediate precursor events leading to the failure, the precursors of these events, and so on until a pyramid structure or 'Tree' is generated. A probability is assigned to each event in the tree and the overall probability of the designated failure occurring is calculated.

The primary difference between FMEA and FTA is that the former starts with the primal precursor events and works forward (or upward) to detect possible failures while the latter identifies a specific failure and works backward (or downward) to identify the precursor events which could cause the failure to occur.

Both FMEA and FTA are systems approaches. To use either one it is necessary to have complete engineering flow diagrams (usually referred to as P&ID's) and complete logic diagrams. It is also necessary to define the system adequately. Although this may seem elementary it is sometimes the most difficult part of the analysis.

Not too many years ago failure analysis was just that--the analysis of failures which had already occurred. This led to systems of preventive maintenance which became more and more sophisticated over the years. No one argues that preventive maintenance is not required. However, preventive maintenance systems are based to a great extent on an analysis of previous failures and do not take into account the unforseen failure. There is no doubt that the more highly sophisticated systems of preventive maintenance could benefit from a FMEA or FTA of the plant being maintained.

Most process design engineers attempt to predict failure pathways through the process they are designing. These predictions are usually based on experience or a 'gut feeling' for a particular pathway. While this approach is not all bad it is definitely inadequate. The preparation of detailed process logic diagrams would in itself be of considerable assistance in predicting failure pathways. Unfortunately it appears that the great majority of engineers engaged in the design of chemical and/or mechanical process plants are blissfully unaware that such a powerful tool as the logic diagram exists. The intuitive approach now used will invariably not consider all failure modes except for the simplest processes. The Fault Tree Analysis system is a far superior, but no means infallible, approach.

The material presented here is not designed to give the reader an in-depth knowledge of the synthesis of fault trees. Many approaches have been made to this subject and the body of literature associated with fault trees is growing rapidly. The intent here is to show how logic diagrams may be used in fault tree work. In the author's opinion, this can be demonstrated most adequately using the techniques developed by Lapp and Powers[1,2]. Only the simplest systems will be considered. Fault tree analysis is a vast field and it would require a book to give an adequate introduction to the subject. The first systems we will consider are those which involve combinational logic only. Although various time lags may be encountered, everything is assumed to happen in the logical 'now'. Later we will consider a sequential logic system where one or more events cannot occur until one or more previous events have been completed.

Any given chemical or mechanical process plant is built from basic components which may interact in many ways. Some type of control instrumentation is almost invariably required for proper operation of the system. This instrumentation may involve open-loop control, closed-loop feed back control, closed-loop feed forward control, or various combinations of any or all of these loops. All of this must be considered in synthesizing a fault tree.

We will consider only simple systems containing few components and develop the trees for these. One of the simplest which could be selected is shown in Figure 1. Here we have a shell and tube cooler with no control instrumentation. Assume that the top event in our tree is a high temperature in stream 4, designated as T4(+1). (A low temperature would be designated as T4(-1)). What could cause this? The possible causes are:

M1(-1)	Mass flow in stream 1 decreases
M3(+1)	Mass flow in stream 3 increases
T3(+1)	Inlet temperature of hot fluid increases
T1(+1)	Inlet temperature of cooling water increases
	Heat exchanger fouled
	External fire at heat exchanger

These conditions are shown graphically in Figure 2. This is a vectored diagram, usually referred to as a digraph (which is a coined word for directed graph). The fault tree for this example is shown in Figure 3. It is evident that occurrence of any of these events would result in the top event.

We will now complicate this system by adding a simple feed back control loop as shown in Figure 4. Now the number of events which can cause the top event is considerably increased. These could be listed but it is much easier to indicate them on the digraph, Figure 5.

The symbols used on a digraph are completely arbitrary. Because they are easily drawn, circles are usually used to designate discreet conditions such as flow rate, temperature, pressure, concentration, etc. Lapp and Powers[1,2] refer to these circles as "nodes" which is as good terminology as any and one which we will use here. Inputs to a node are indicated by directed lines. Lapp and Powers[1,2] use the term "edge" to describe these lines. We will adopt this terminology also. If a relationship between two nodes, shown by an edge, depends upon another relationship, the edge is known as a conditional edge. A node which has no input is called a primal node or prime event. (Note that in Figure 3 'HX Fouled' is a prime event while in Figure 5 it is an event causing a conditional edge. In the second case fouling may be compensated for up to the point where the control valve is wide open).

A gain is always connected with an edge. Gain is Δ output/Δ input. If the gain is greater than 1 it is defined as positive and if it is less than 1 it is defined as negative. If there is no change in output with change in input, the gain is zero. In a hypothetical system we have no way of knowing what the absolute value of the gain is. We can refer to gains only as zero, small or large. For comparison purposes we can arbitrarily assign values to these, e.g., large = ± 1000, small = ± 100. Lapp and Powers[1,2] use ± 10, ± 1, and 0. These are convenient numbers to use so we will adopt them. It must be emphasized that these values are arbitrary. For example, a small external fire at a heat exchanger would be assigned a +1, and a large external fire a +10. How about a medium fire? We are only allowing values of 10, 1 and 0. When in doubt call it a large fire.

All of the information needed to analyze the system is given on the digraph. It is not, however, in a readily usable form. The information is put into usable form through the use of a logic diagram. Before we can do this we must learn more about loops, gains, and deviations.

In Figure 5 we have a control loop which must be classified. The elements of the loop are:

$$T2 \xleftarrow{-1} M4 \xleftarrow{+1} P6 \xleftarrow{+1} P5 \xleftarrow{+1} T2$$

The gains for normal operation are shown above the arrows. The net gain is (-1) (+1) (+1) (+1) = -1 so this is a negative feedback loop (NFBL). We must now consider the magnitude of the disturbances which could occur. We have used (+1) and (-1) to indicate normal disturbances. For big and/or fast disturbances we will use (+10) and (-10). Could our loop handle such disturbances? To answer this we must analyze the interior elements of the loop.

M4 (-10)	Severe decrease in flow rate or loss of flow from supply. Neither big nor fast disturbances could be handled. Opening the control valve would not increase the flow.
M4(+10)	Large increase in flow rate from supply. Both large and fast disturbances could be handled by throttling the control valve.
P6(-10)	Loss of instrument air pressure to the temperature recording controller. Neither large nor fast disturbances can be handled. The TRC will cause the valve to go fully open or fully closed depending upon its design.

Large and/or fast disturbances external to the loop which will cause the top event are:

M1(+10)
T1(+10)
T4(+10) or T7(+10)
Large external fire at heat exchanger

It is obvious that the fault tree for this system is much more complex than that shown in Figure 3. An excellent methodology for synthesizing such a tree has been developed by Lapp and Powers. This is best described by the Lapp-Powers Fault Tree Synthesis Algorithm. This algorithm is shown in Figures 6a through 6d. For simple cases such as we are considering, a fault tree may be synthesized manually by using this method. For more complex systems the computer approach developed by Lapp and Powers is recommended.

We will now construct the fault tree step by step for the system shown in Figure 4. We will select T8(+1), i.e., T8 is high, as the top event. We have already complied with the first four steps of the algorithm. The only variable directly affecting T8 is T2 so this is chosen as our undeveloped variable. We now ask, and answer, these questions:

Is T2 on a NFBL?	Yes.
Does the output have value = 0 ?	No.

This indicates that for Step 1 we should go to Case D which is shown on Figure 6d. The result of this step is shown on Figure 7. In this step we have developed the variables T2, M1, T1, T4, external fire, and heat exchanger (HX) fouled. All of these are primal events because they are not subject to control within the system as defined. The variables remaining to be developed are M4, P6, and P5. We will develop them in that order, which is the sequence in which they appear in the loop. In respect to M4 we again ask, and answer, these questions:

Is M4 on a NFBL?	Yes.
Does the output have value = 0 ?	Note that M4 appears twice, once with value =0 and once with value = -1. We will consider the value = 0 condition first.

When M4 has the output value = 0 we go to Case C. Step 2 is shown on Figure 8. When M4 has the output value = -1 we go to Case D. Step 3 is shown on Figure 9. Note the area on this figure enclosed by the dashed line. This area was developed in Figure 8 and it is unnecessary to show it twice. In our final diagram we directly connect this area to its multiple destinations if it is convenient to do so. Otherwise, we will indicate duplicate areas by match marking as 1⃤ 1⃤ , 2⃤ 2⃤ , etc.

We must now develop P6. Again we have two output values, P6(0) and P6(-1). We will consider P6(0) first. P6 is on the NFBL so the algorithm directs us to use Case C. Step 4 is shown on Figure 10. For the condition where the output value = -1 we are referred to Case D. Step 5 is shown on Figure 11. This leaves only P5 to be developed. This also has two values. Step 6, for P5(0) is shown on Figure 12. In this figure, the x across the input T2(0) means that this is not an allowable input. We have already established that T2 has the value = +1. Step 7 for P5(-1) is shown on Figure 13.

We now combine Figures 7 through 13 in Figure 14 with duplications omitted. At this point there will be a natural desire to 'collapse' the tree somewhat. It is obvious that several OR functions could be combined and single input functions could be eliminated.

It is also possible that certain Boolean manipulations may be made to simplify the diagram. Resist the temptation to do either. A relatively minor change in the process or in the instrumentation could cost many man-hours to find how it affected the tree if such 'simplifications' were made.

The concept of 'cut sets' is useful in analysing a fault tree. A cut set is the set of events along a pathway up the tree which will cause the top event to occur. Minimal cut sets are those which contain no other cut sets within them. There may be many cut sets and minimal cut sets in a large tree and they will contain many elements. These cut sets will tend to proceed through OR and EOR gates but may also encounter AND gates. In the simple fault tree shown in Figure 14 there are several pathways and several minimal cut sets. Because of the simplicity of this system most of these sets contain one element only (ignoring ∅ which is an element of every set but which here indicates the absence of an event). These sets are:

Set No.	Elements
1	{M1(+10)}
2	{T1(+10)}
3	{T3(+10)}
4	{Large fire at heat exchanger}
5	{Control valve reversed}
6	{M7(-10)}
7	{TAP(-10)}
8	{Set point(+1)}
9	{Temperature sensor failed low}
10	{TRC reversed}
11	{M1(+1), control valve stuck}
12	{T1(+1), control valve stuck}
13	{T4(+1), control valve stuck}

The first ten of these are more critical than the last three because they depend on one event only. Using these ten only and using the set numbers as event numbers we may reduce the tree in Figure 14 to the tree in Figure 15.

The more AND gates we can get into the tree, particularly near the top, the more reliable our system will be--at least theoretically. The control system for the heat exchanger shown in Figure 4 could be modified as shown in Figure 16. Here we have added a backup system consisting of a second temperature sensor, a pneumatic/electric transducer (I/P), and two solenoid valves. The construction of the digraph and fault tree for the modified system is not included here because of space limitations. Note that there are three feedback loops in this system and that the backup valves fail safe on loss of either air pressure or electric power. If, in a real-life situation, T8(+1) was a critical condition which could cause a hazardous event, something such as this backup system would be justified.

So far we have considered systems involving combinational logic only. Systems involving sequential logic, or a mixture of the two, are of equal or greater importance. Here we must distinguish between sequential systems and sequential logic. It is a very common error to assume that the logic encountered in a sequential system is sequential logic. It may or may not be. An example frequently used to illustrate a sequential system is the dual adsorption tower air drying unit with hot air regeneration of the adsorbent. This unit has four distinct modes or operation which are:

MODE	TOWER A	TOWER B
1	regenerating	in service
2	cooling	in service
3	in service	regenerating
4	in service	cooling

The system operation is sequential in that the modes are established in sequence by a timer. The logic within each mode is, however, combinational and the synthesis of the fault tree is the combination of the subtrees for each mode. Shaewitz, Lapp, and Powers[3] have presented a very thorough analysis of this system. The digraph they show covers all of the modes so some care must be taken in following the loops for each mode.

An example of a simple system involving sequential logic is that of two pumps starting successively. Pump A must run before Pump B can run. Pump B must, however, pump within a given flow range or it will shut down and automatically shut down Pump A. A logic diagram for this system is shown in Figure 17. Because this system is completely electrical, a partial* elementary wiring diagram (ladder diagram), constructed from the logic diagram, is shown in Figure 18. This diagram indicates relay circuitry although in actual practice solid-state circuitry would probably be used.

*Partial in that thermal overload relays, fuses, etc. are omitted.

For readers who are not familiar with ladder diagrams an explanation of the symbols is given at the end of the paper. An excellent explanation of these diagrams may also be found in the November 15, 1971 issue of CHEMICAL ENGINEERING. Figure 19 is the digraph for this system. In this digraph we use the relay and contact symbols used in the ladder diagram.

The concept of gain in a relay type electrical system is somewhat restricted. If a circuit is energized and the corresponding relay, or relays, are energized, the gain is +1. If the circuit is deenergized and the corresponding relays are deenergized, the gain is -1. If energizing or deenergizing the circuit has no effect on the relays (relay stuck, burned out, etc.), the gain is either +1 or -1 depending on whether the contacts are open or closed at the time. Because current is either flowing or not flowing there is no zero again. A gain other than these has no meaning in relay circuitry. This is not true of solid state circuitry where voltage surges, etc. become important. On the Fault Tree itself, only the circuit conditions 1 and 0 may appear.

There are four failure modes for this system. They are:

1. Pump A will not start.
2. Pump A starts, Pump B will not start, Pump A shuts down.
3. Pump A starts. Pump B will not start, Pump A continues to run.
4. Pump A starts, Pump B starts, system shuts down after x seconds.

The fault tree for this system is found by combining all of the fault trees for the four failure modes. This is shown in Figure 20. Some of the features of the Lapp-Powers algorithm will be found in this fault tree but at present (1978) there is no algorithm available for a system involving truly sequential logic.

So far we have considered fault tree synthesis only. A fault tree analysis requires assignment of the probability of occurrence to each of the events in the tree and combining these probabilities to obtain the overall probability of the top event occurring. The method of combining the individual probabilities is covered in any standard text on probability. Obtaining the individual probabilities is the problem. Although the body of literature on the subject is growing it is still quite scant and narrow in scope. There is a vast amount of data buried in the maintenance files of all medium to large size companies. Unearthing these data and putting them into useful form would be an extremely expensive and time consuming task. If any of the large companies were willing to undertake such a project, they would understandably be unwilling to release the results for general use. If, however, the Federal Government were to fund such a project, the information would be in the public domain and available to everyone.

Note: This article is excerpted from the manuscript of an as yet unpublished book, tentatively titled "Applied Symbolic Logic," by Edward P. Lynch. The author thanks Dr. Gary J. Powers and Dr. Stephen A. Lapp, both of Carnegie Mellon University, for permission to use certain material included in their 'Short Course on Fault Tree Analysis' in the manuscript for this book. Figures in the foregoing article which are part of this material are identified as being the work of Drs. Powers and Lapp.

REFERENCES

1. Lapp, S.A. and Powers, G.J., unpublished material from the *Short Course on Fault Tree Analysis* offered by Carnegie Mellon University, (1977).

2. Lapp, S.A. and Powers, G.J., *Computer Aided Synthesis of Fault Trees*, IEEE Transactions on Reliability, April 1977.

3. Shaeiwitz, J.A., Lapp, S.A., and Powers, G.J., *Fault Tree Analysis of Sequential Systems*, Ind. Eng. Chem., Process Des. Dev., Vol. 16, No. 4, 1977.

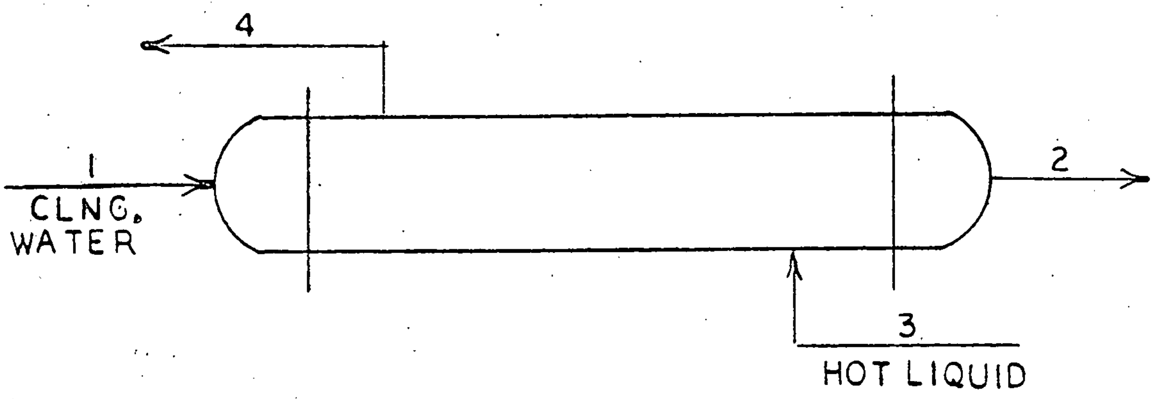


FIGURE 1

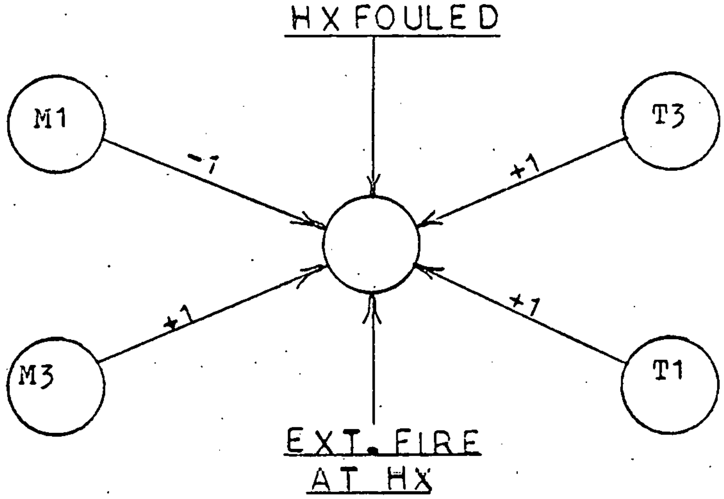


FIGURE 2

Courtesy of Lapp and Powers

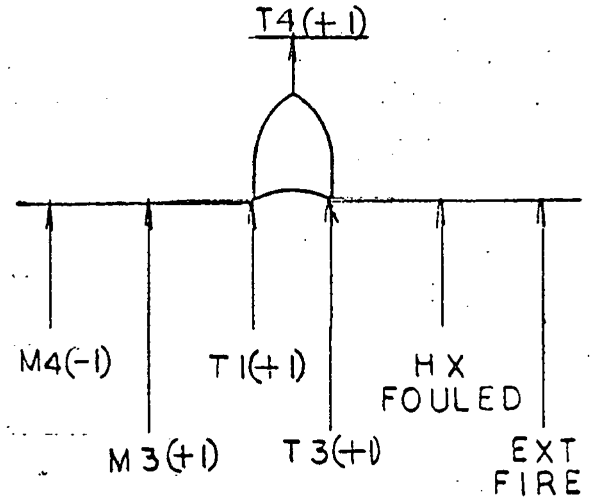


FIGURE 3

Courtesy of Lapp and Powers

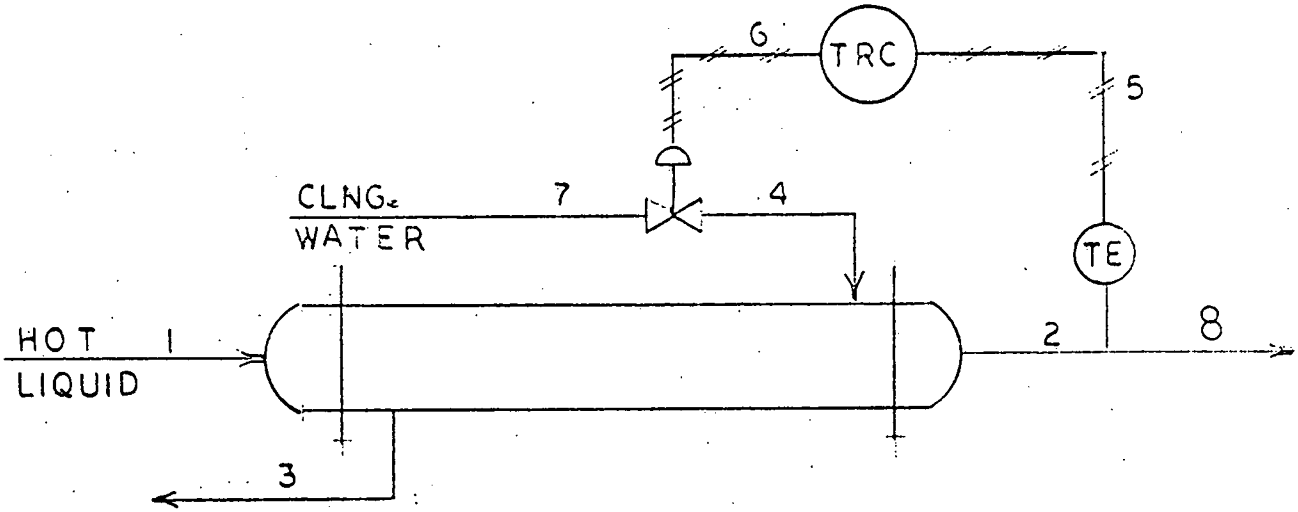


FIGURE 4

Courtesy of Lapp and Powers

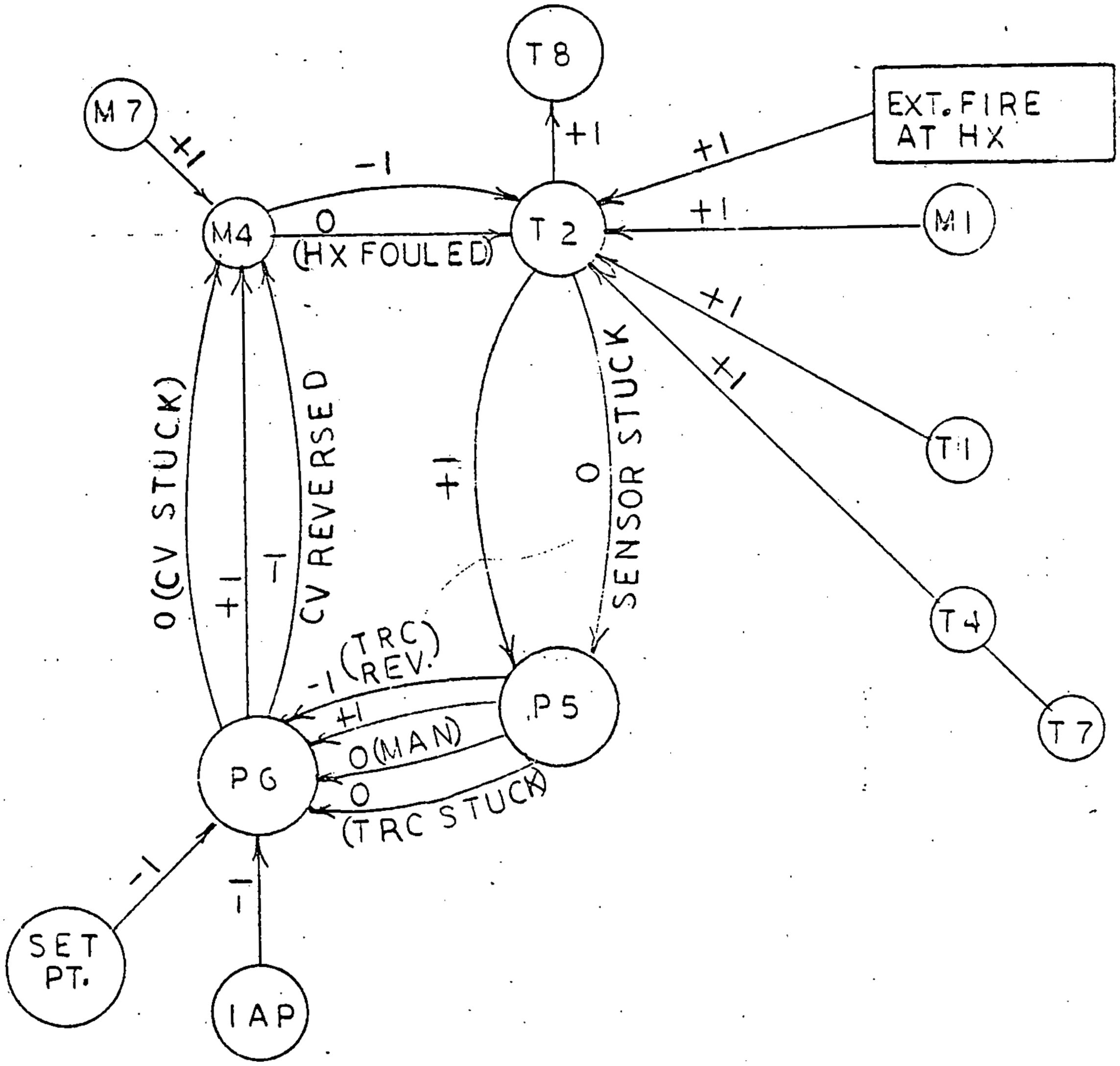


FIGURE 5

Courtesy of Lapp and Powers

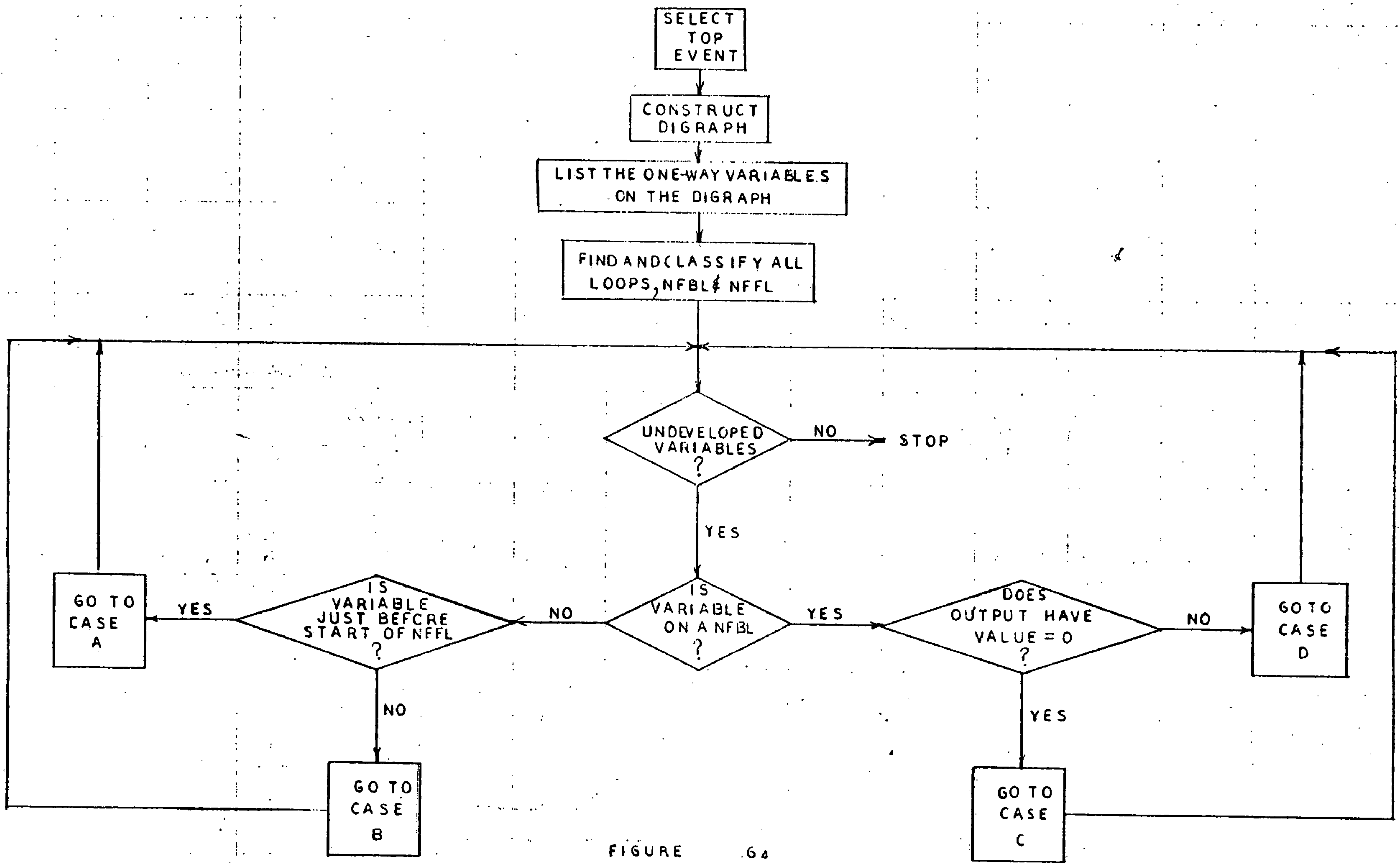
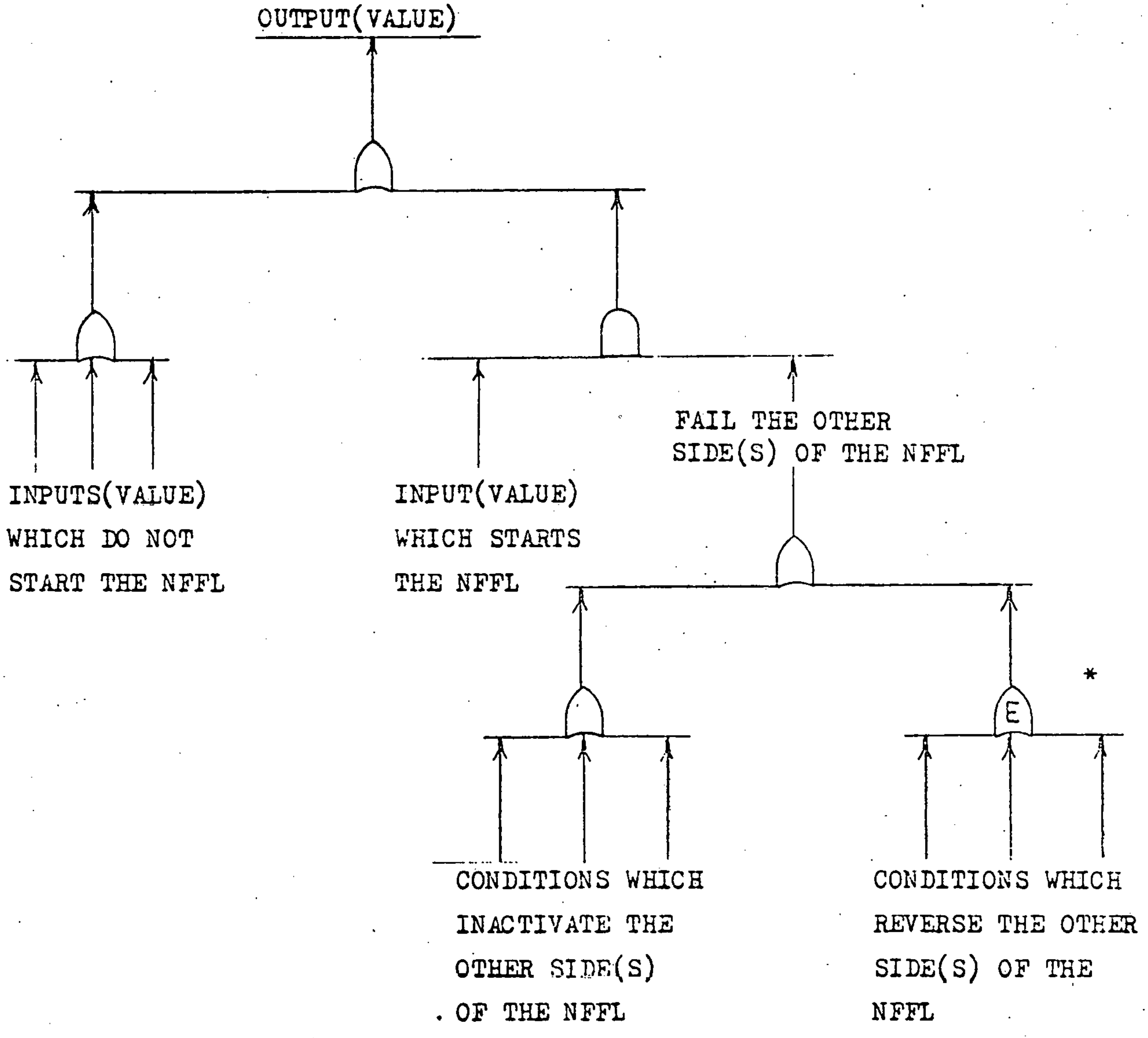


FIGURE 6a

LAPP POWERS ALGORITHM

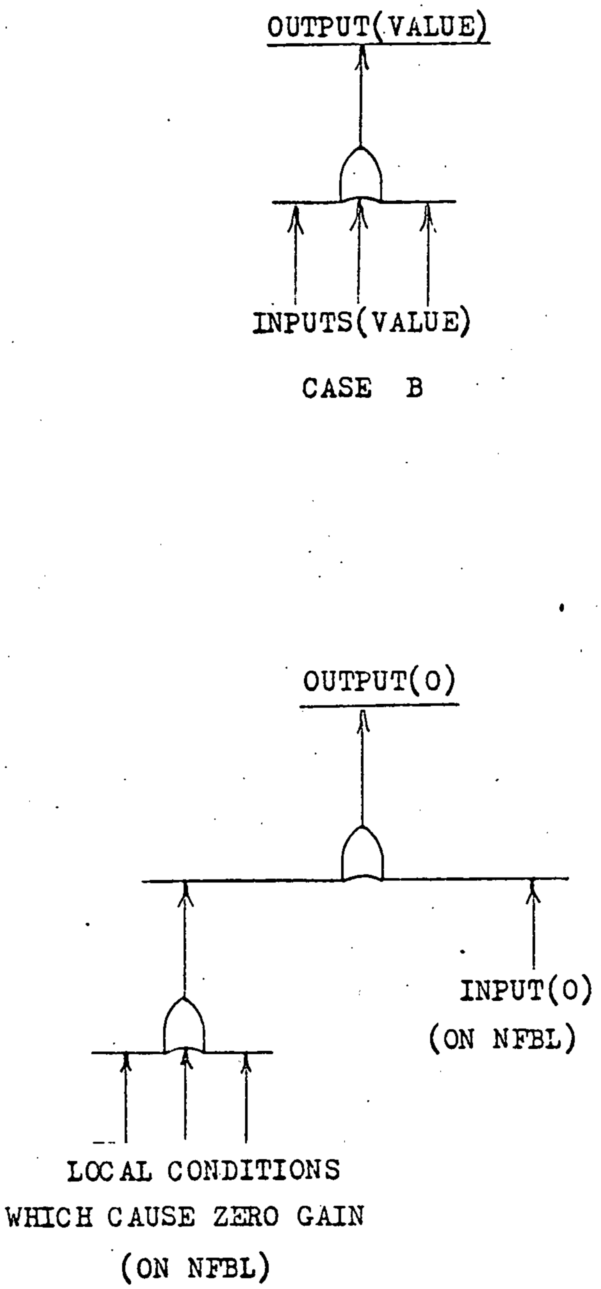

OUTPUT(VALUE)
INPUTS(VALUE) WHICH DO NOT START THE NFFL
INPUT(VALUE) WHICH STARTS THE NFFL
FAIL THE OTHER SIDE(S) OF THE NFFL
E
*
CONDITIONS WHICH INACTIVATE THE OTHER SIDE(S) OF THE NFFL
CONDITIONS WHICH REVERSE THE OTHER SIDE(S) OF THE NFFL

CASE A

FIGURE 6b

Lapp-Powers Algorithm

* Ⓔ is the symbol for the EXCLUSIVE-OR

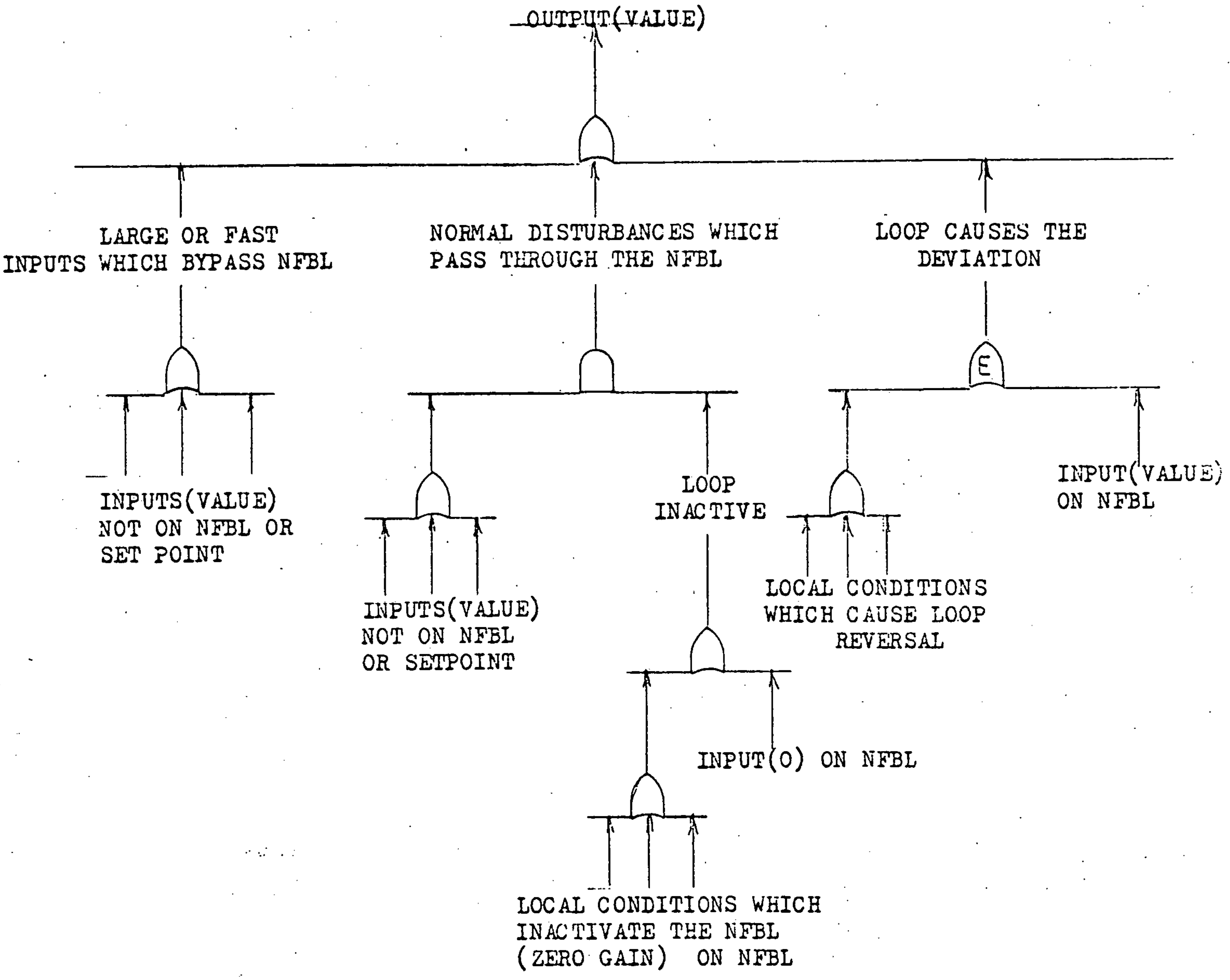


CASE B

CASE C

FIGURE 6c

Lapp-Powers Algorithm


OUTPUT(VALUE)
LARGE OR FAST INPUTS WHICH BYPASS NFBL
NORMAL DISTURBANCES WHICH PASS THROUGH THE NFBL
LOOP CAUSES THE DEVIATION
E
INPUTS(VALUE) NOT ON NFBL OR SET POINT
INPUTS(VALUE) NOT ON NFBL OR SETPOINT
LOOP INACTIVE
LOCAL CONDITIONS WHICH CAUSE LOOP REVERSAL
INPUT(VALUE) ON NFBL
INPUT(O) ON NFBL
LOCAL CONDITIONS WHICH INACTIVATE THE NFBL (ZERO GAIN) ON NFBL

CASE D

FIGURE 6d

Lapp-Powers Algorithm

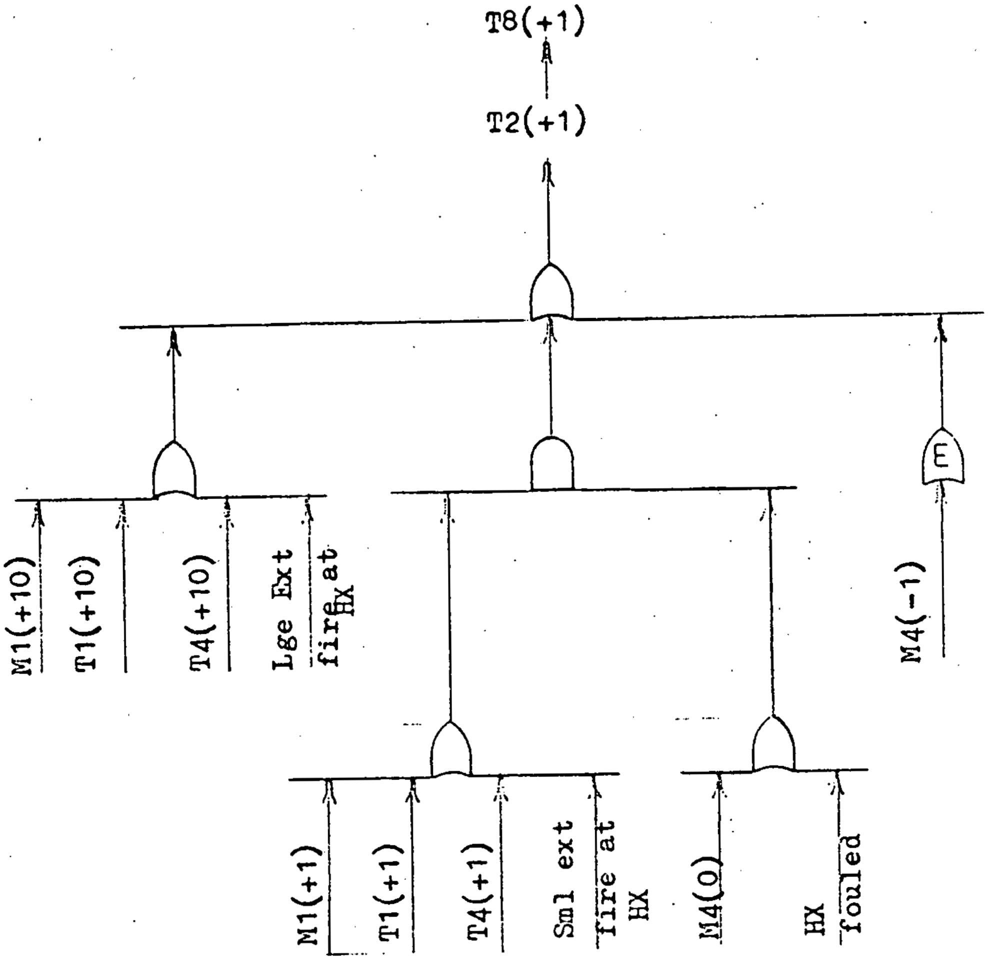


FIGURE 7

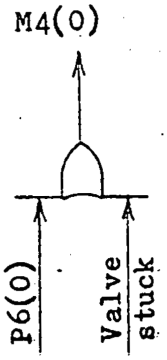


FIGURE 8

Courtesy of Lapp and Powers

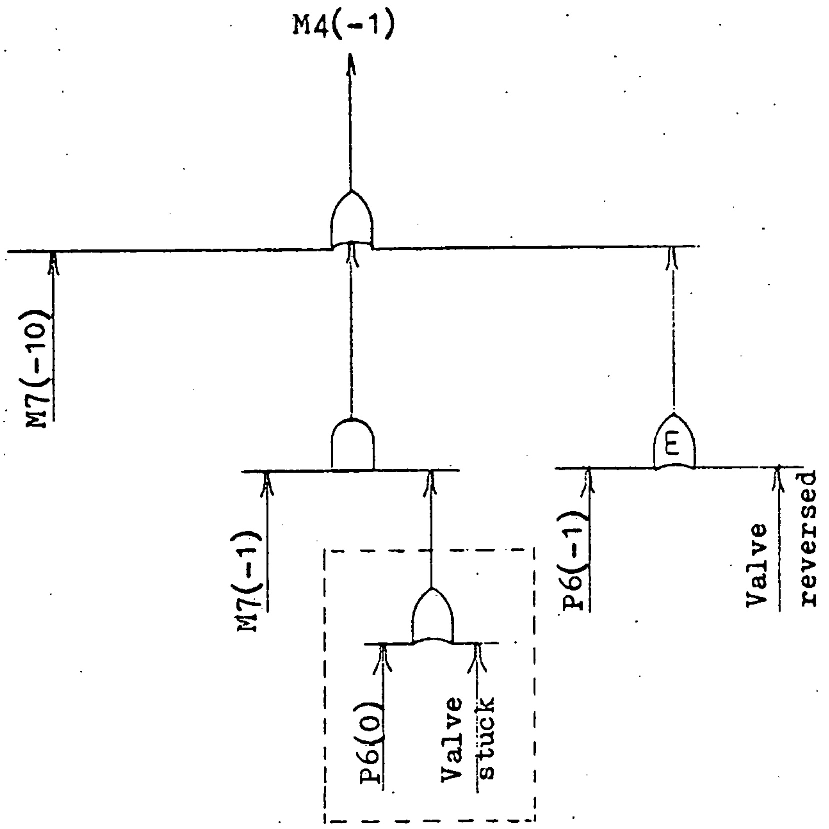


FIGURE 9

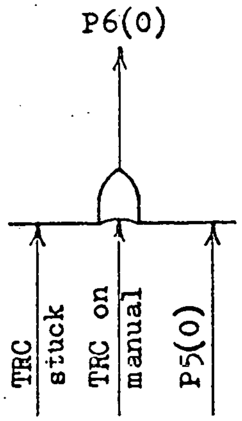


FIGURE 10

Courtesy of Lapp and Powers

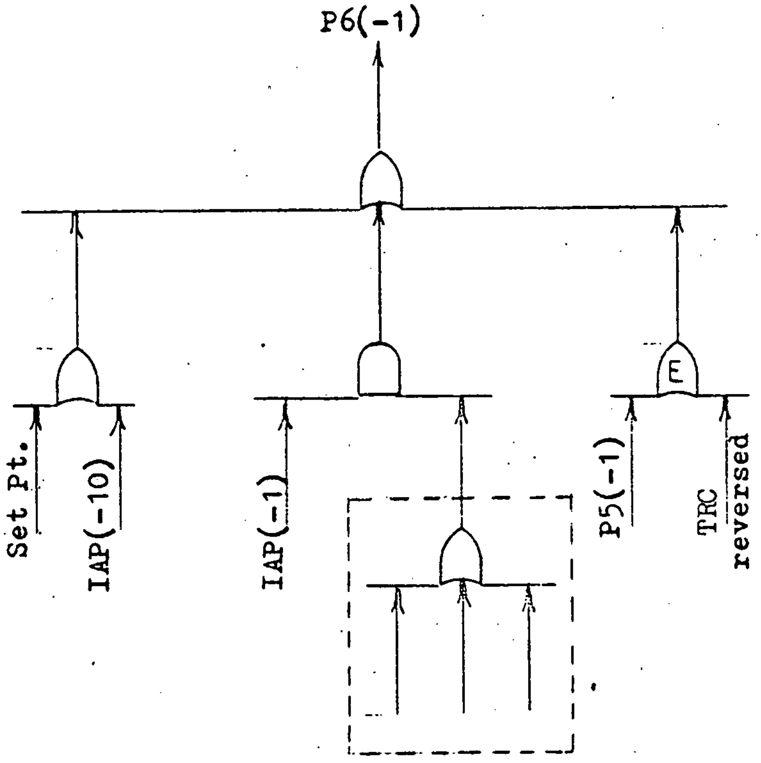


FIGURE 11

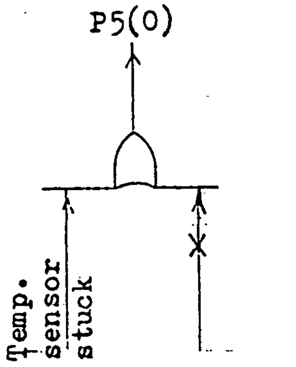


FIGURE 12

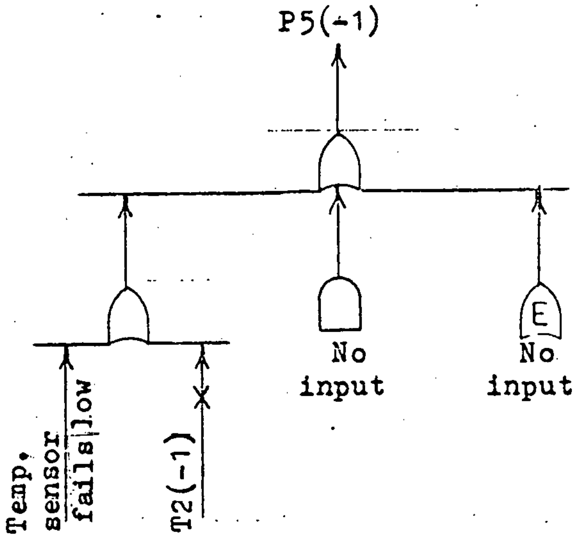


FIGURE 13

Courtesy of Lapp and Powers

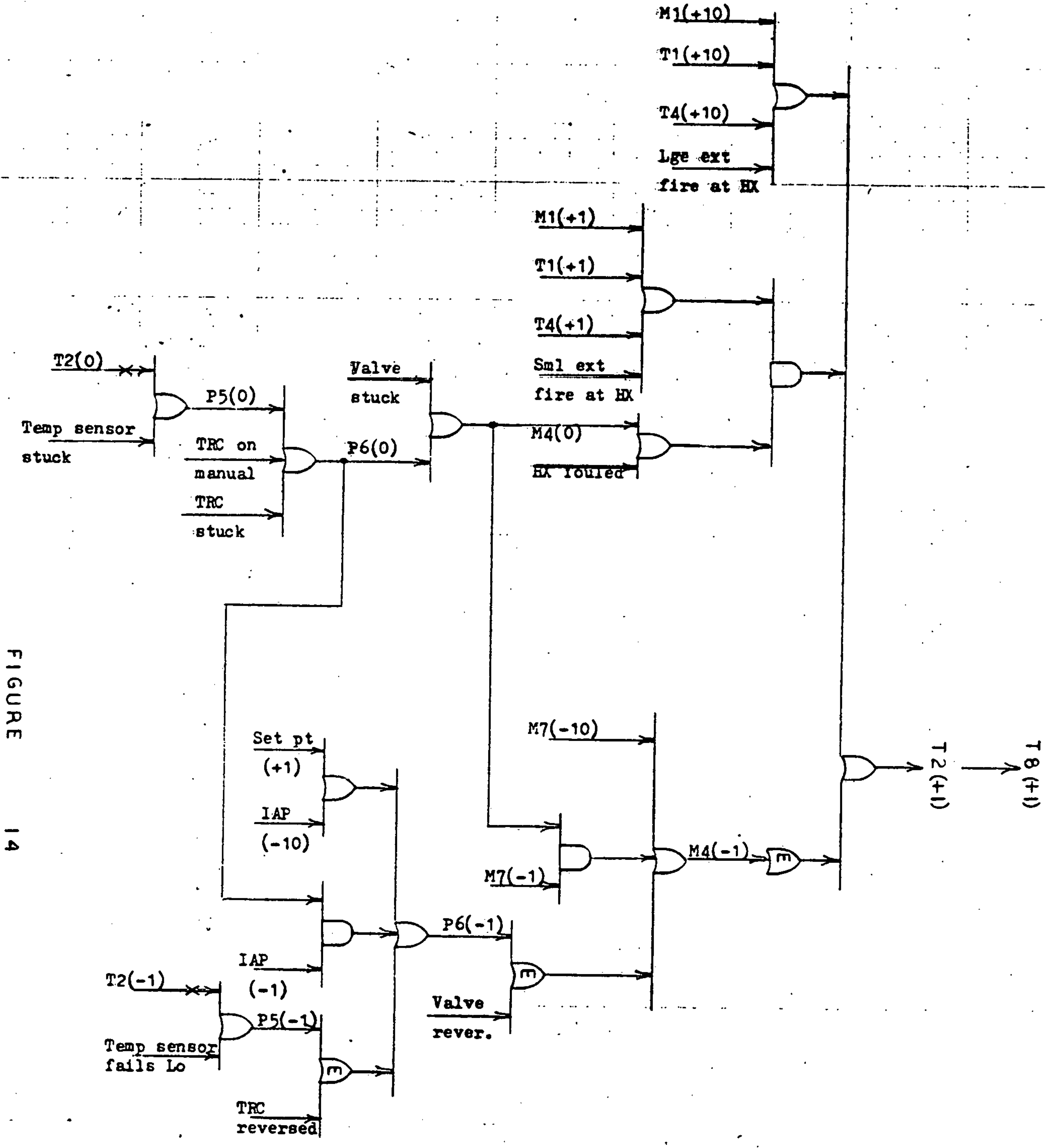

M1(+10)
T1(+10)
T4(+10)
Lge ext fire at HX
M1(+1)
T1(+1)
T4(+1)
Sml ext fire at HX
M4(0)
HX fouled
T2(0)
Temp sensor stuck
P5(0)
TRC on manual
TRC stuck
P6(0)
Valve stuck
T2(+1)
T8(+1)
M7(-10)
M7(-1)
M4(-1)
Set pt (+1)
IAP (-10)
IAP (-1)
T2(-1)
Temp sensor fails Lo
P5(-1)
TRC reversed
P6(-1)
Valve rever.

FIGURE 14

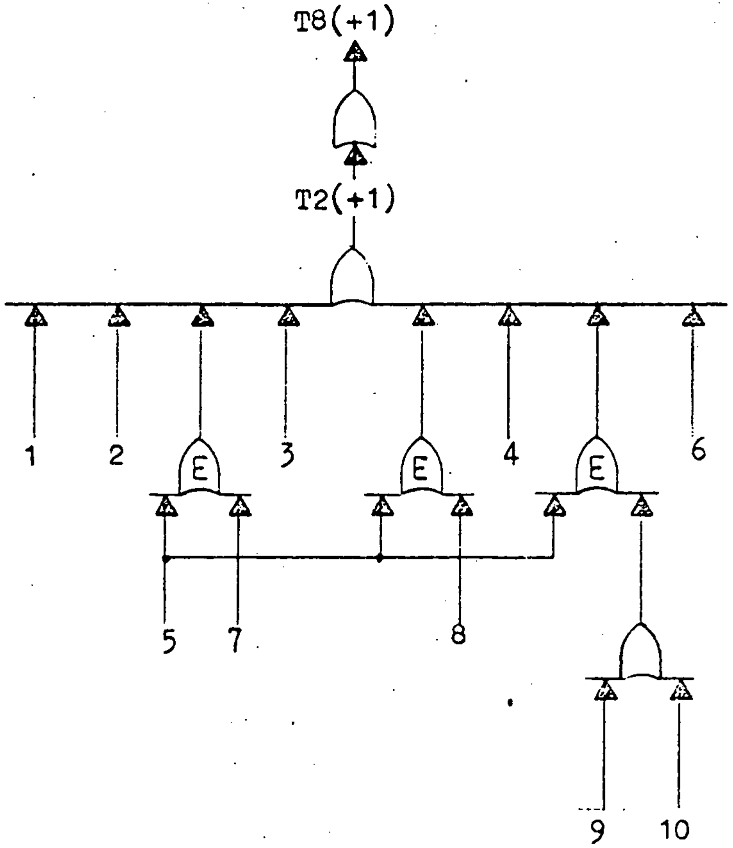


Figure 15

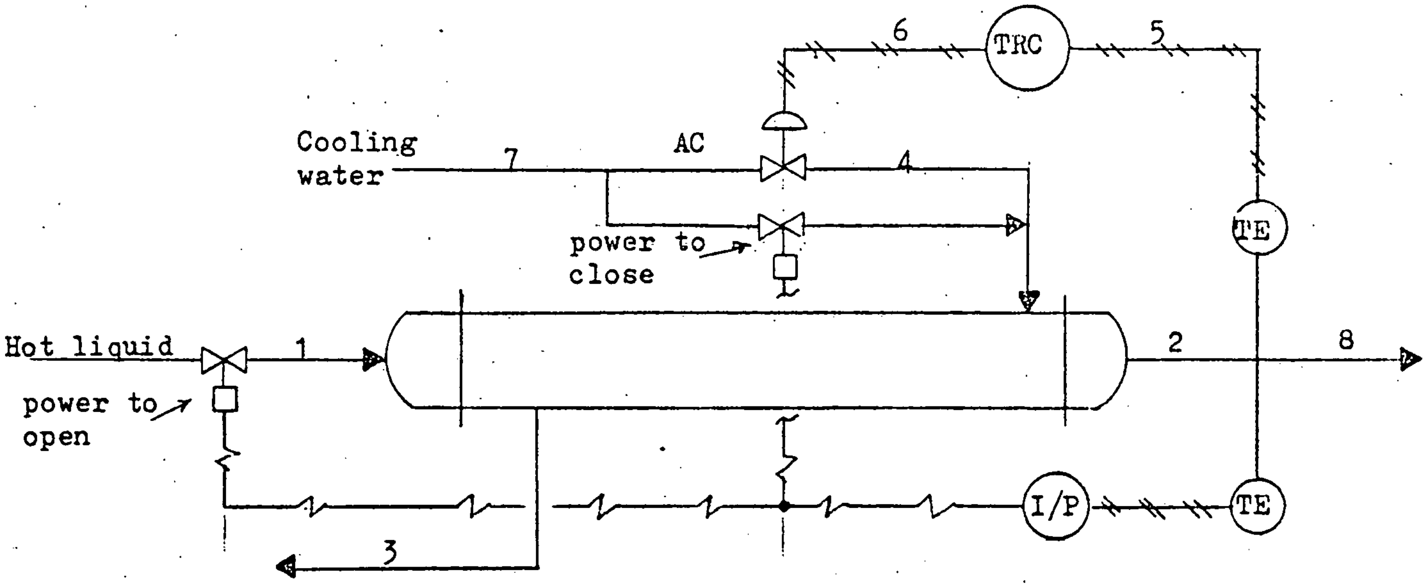
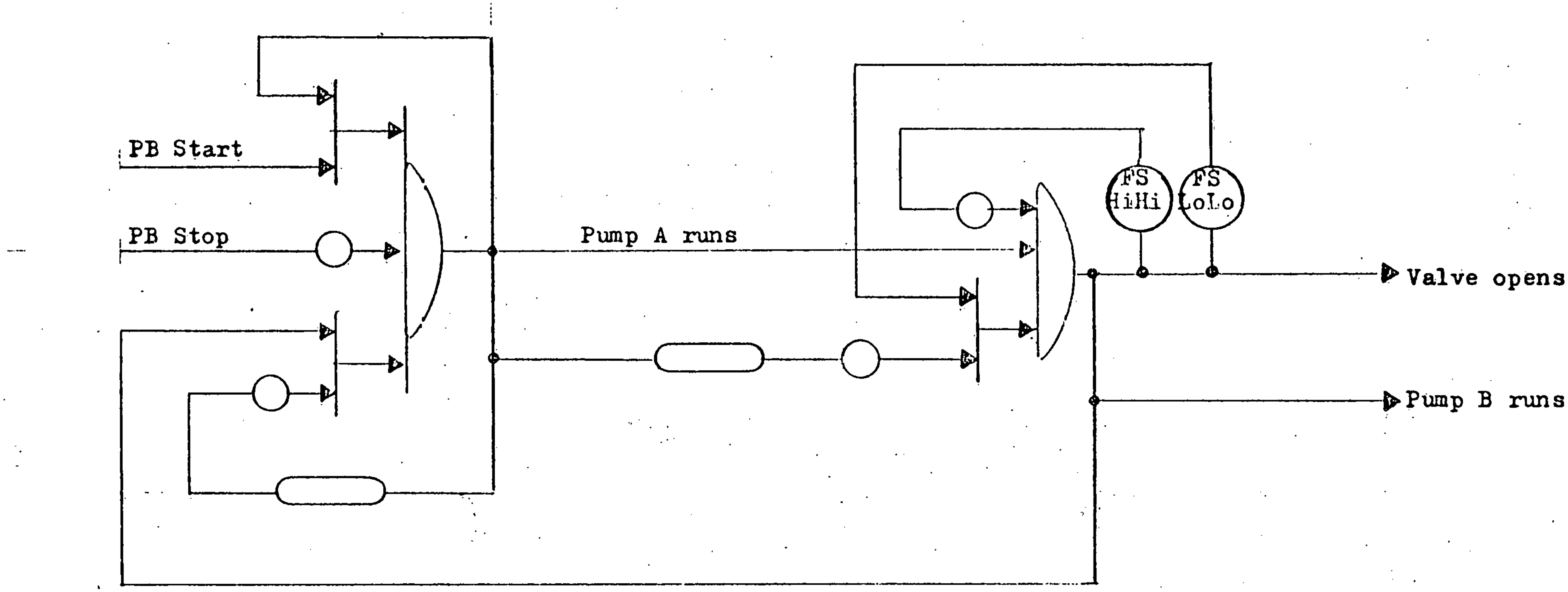


Figure 16


PB Start
PB Stop
Pump A runs
FS HiHi
FS LoLo
Valve opens
Pump B runs

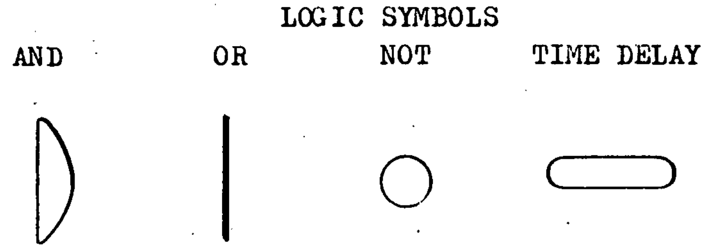

LOGIC SYMBOLS
AND
OR
NOT
TIME DELAY

Figure 17

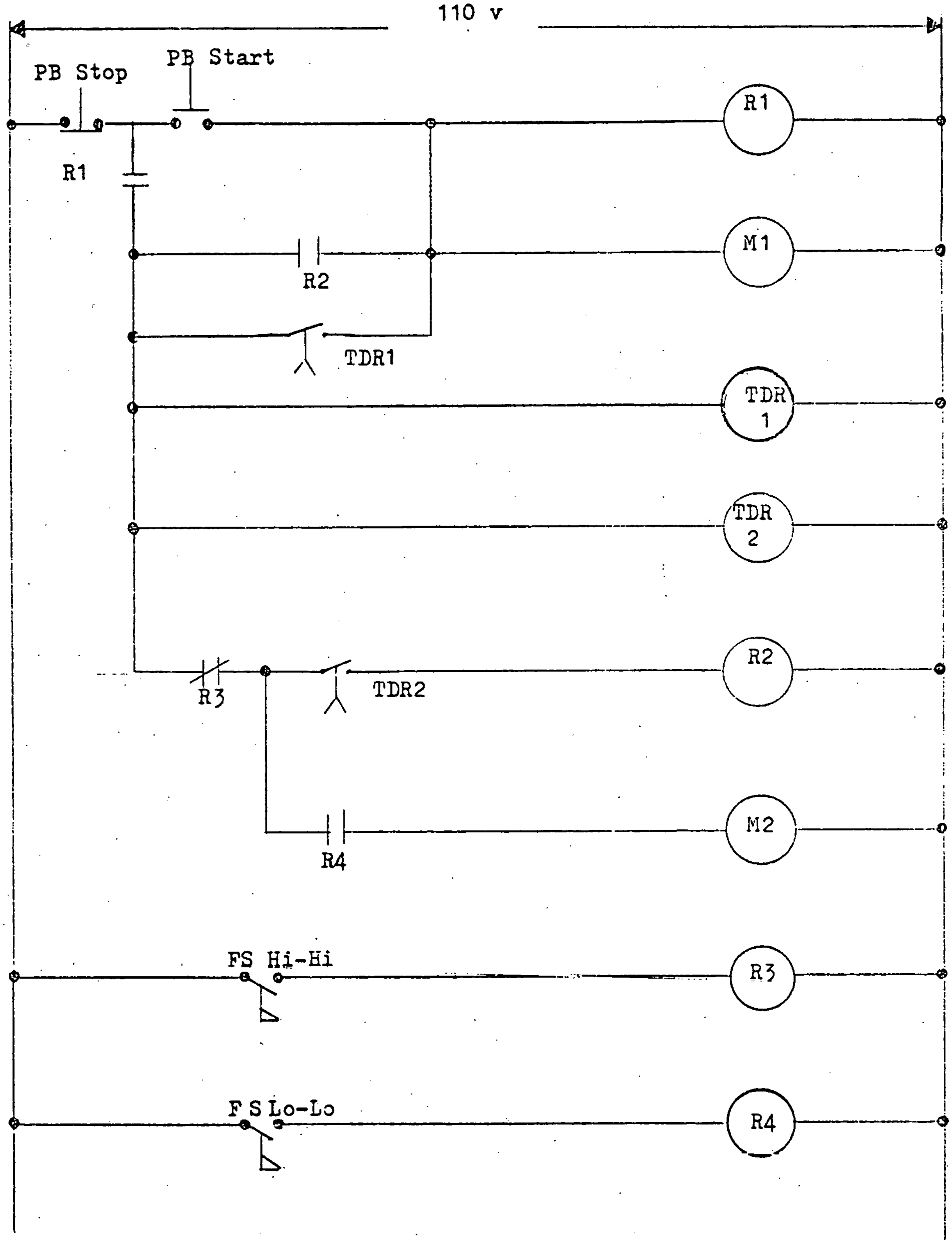

110 v
PB Stop
PB Start
R1
R1
M1
R2
TDR1
TDR 1
TDR 2
R3
TDR2
R2
M2
R4
FS Hi-Hi
R3
F S Lo-Lo
R4

Figure 18

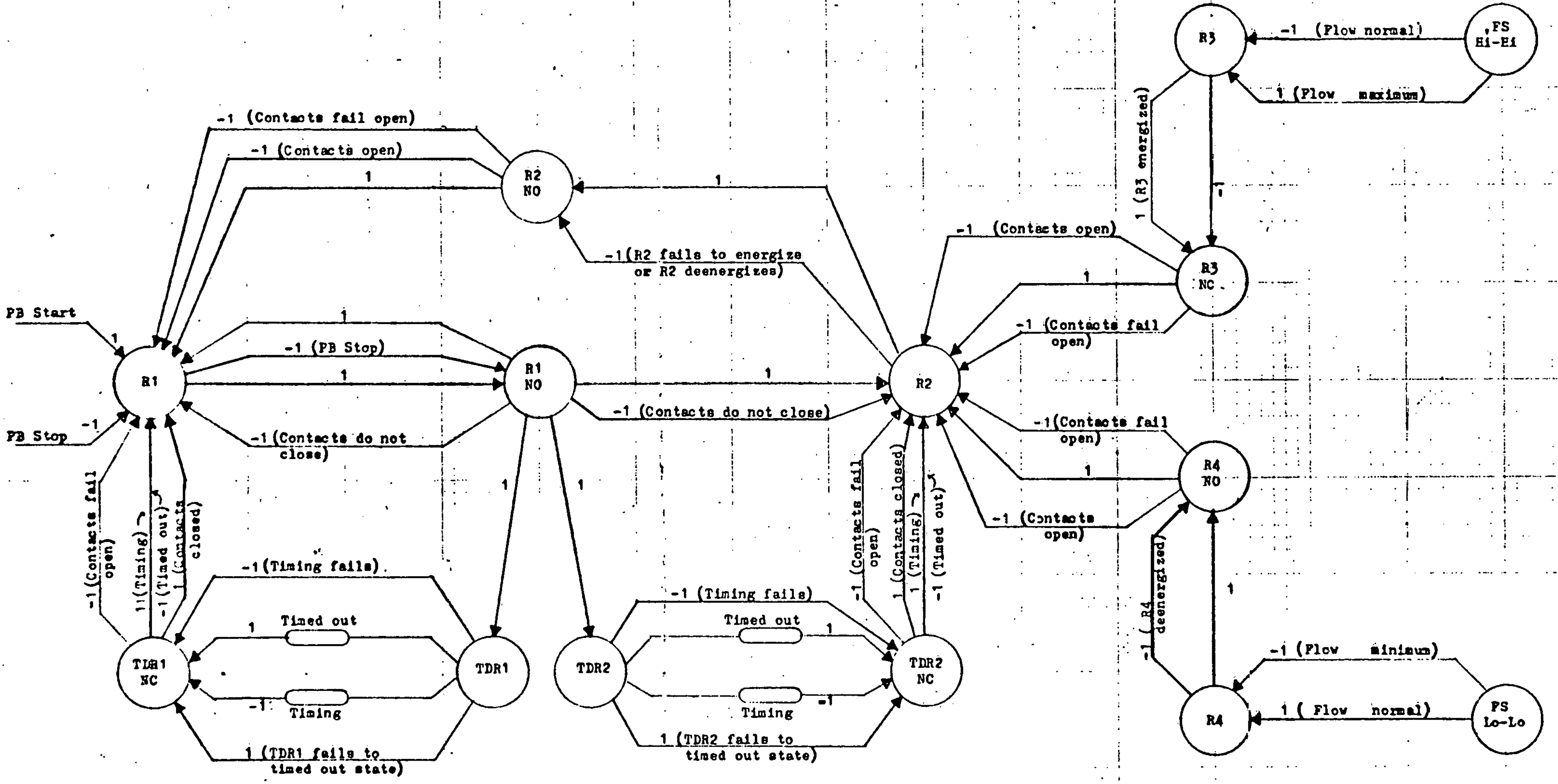

R1
R1 NO
R2
R2 NO
R3
R3 NC
R4
R4 NO
TDR1
TDR1 NC
TDR2
TDR2 NC
FS Hi-Hi
FS Lo-Lo
PB Start
PB Stop
1
-1
-1 (Contacts fail open)
-1 (Contacts open)
-1 (PB Stop)
-1 (Contacts do not close)
-1 (R2 fails to energize or R2 deenergizes)
-1 (Contacts open)
-1 (Contacts fail open)
-1 (Contacts fail open)
-1 (Contacts open)
-1 (Flow normal)
1 (Flow maximum)
1 (R3 energized)
-1 (R4 deenergized)
-1 (Flow minimum)
1 (Flow normal)
-1 (Contacts fail open)
1 (Timing)
-1 (Timed out)
1 (Contacts closed)
-1 (Timing fails)
Timed out
Timing
1 (TDR1 fails to timed out state)
-1 (Timing fails)
1 (TDR2 fails to timed out state)
-1 (Contacts fail open)
1 (Contacts closed)
1 (Timing)
-1 (Timed out)

Figure 19

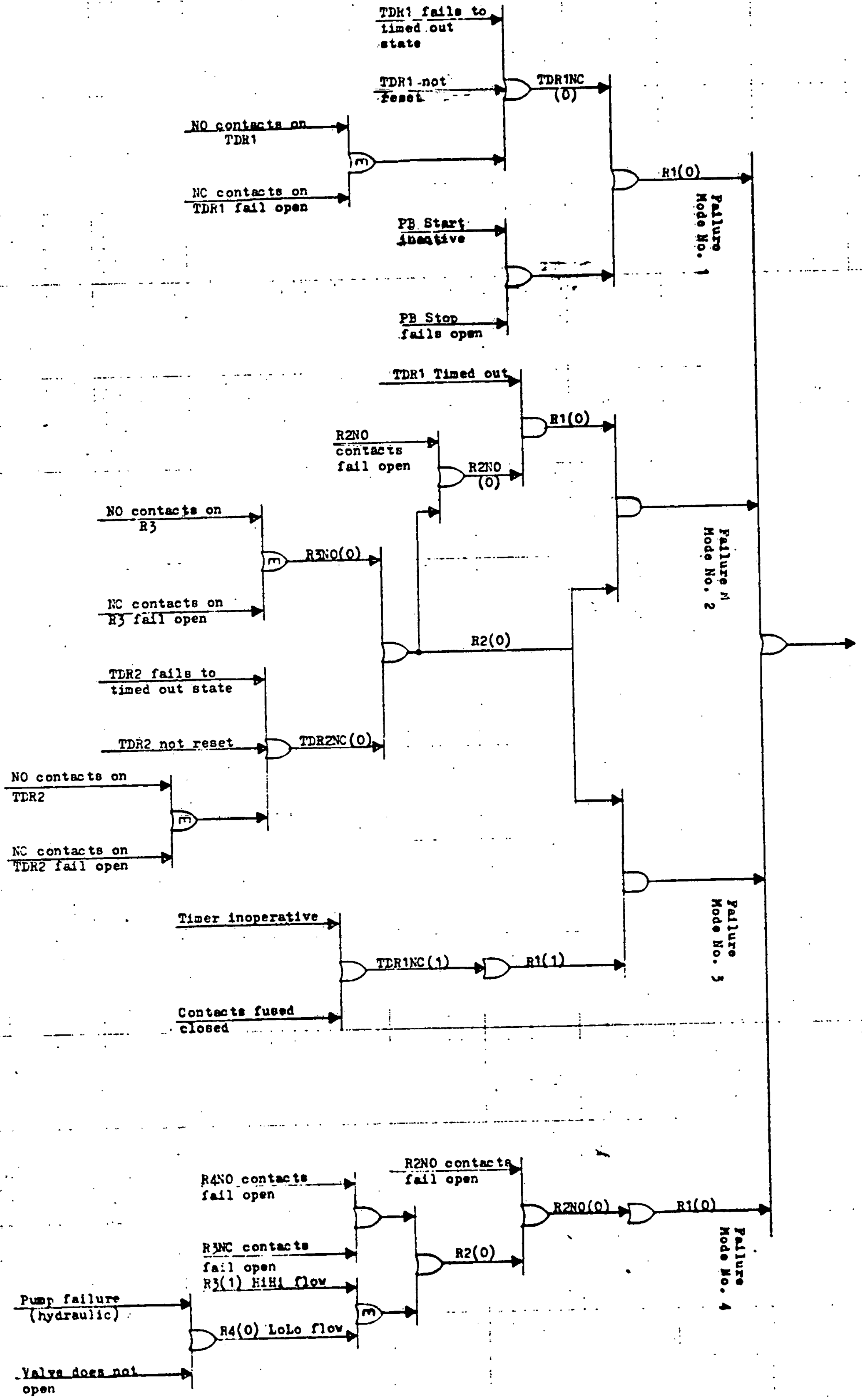

TDR1 fails to timed out state
TDR1 not reset
TDR1NC (0)
NO contacts on TDR1
NC contacts on TDR1 fail open
E
R1(0)
PB Start inactive
PB Stop fails open
Failure Mode No. 1
TDR1 Timed out
R1(0)
R2NO contacts fail open
R2NO (0)
NO contacts on R3
E
R3NO(0)
NC contacts on R3 fail open
Failure Mode No. 2
R2(0)
TDR2 fails to timed out state
TDR2 not reset
TDR2NC(0)
NO contacts on TDR2
E
NC contacts on TDR2 fail open
System Failure
Timer inoperative
TDR1NC(1)
R1(1)
Contacts fused closed
Failure Mode No. 3
R4NO contacts fail open
R2NO contacts fail open
R2NO(0)
R1(0)
R3NC contacts fail open
R2(0)
R3(1) HiHi flow
Pump failure (hydraulic)
E
R4(0) LoLo flow
Valve does not open
Failure Mode No. 4

Figure 20