FAULT-TREE ANALYSIS OF DIFFERENT SYSTEMS

Thesis submitted in partial fulfillment of the requirement for

The award of the degree of

Masters of Science

In

Mathematics and Computing

Submitted by

Ankush Kumar

Roll no. 30703003

Under the guidance of

Dr. Rajesh Kumar Gupta

and

Dr. Amit Kumar

JULY 2009

School of Mathematics and Computer Applications

Thapar University

Patiala-147004 (PUNJAB)

INDIA

DEDICATED

TO

GOD, MY PARENTS AND MY SUPERVISIOR'S

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled "**Fault tree Analysis of Different Systems**" in partial fulfillment of the requirements for the award of degree of **Master of Science**, School of Mathematics and Computer Applications, Thapar University, Patiala is an authentic record of my own work carried out under the supervision of **Dr. Rajesh Kumar Gupta** and **Dr. Amit Kumar**.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Ankush Kumar
(**Ankush Kumar**)

This is to certify that the above statement made by the candidate is correct and true to the best of our knowledge.

14/07/09

(Dr. Rajesh Kumar Gupta)
Supervisor
Lecturer, SMCA
Thapar University, Patiala.

(Dr. Amit Kumar)
Supervisor
Lecturer, SMCA
Thapar University, Patiala.

Countersigned by:

14.7.09

Dr. S.S. Bhatia
(Professor & Head)
School of Mathematics & Computer Applications
Thapar University, Patiala.

22/7/09

Dr. R.K. Sharma
Dean of Academic Affairs
Thapar University
Patiala.

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled "**Fault tree Analysis of Different Systems**" in partial fulfillment of the requirements for the award of degree of **Master of Science**, School of Mathematics and Computer Applications, Thapar University, Patiala is an authentic record of my own work carried out under the supervision of **Dr. Rajesh Kumar Gupta** and **Dr. Amit Kumar**.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(**Ankush Kumar**)

This is to certify that the above statement made by the candidate is correct and true to the best of our knowledge.

14/07/09

(Dr. Rajesh Kumar Gupta)
Supervisor
Lecturer, SMCA
Thapar University, Patiala.

(Dr. Amit Kumar)
Supervisor
Lecturer, SMCA
Thapar University, Patiala.

Countersigned by:

14.7.09

Dr. S.S. Bhatia
(Professor & Head)
School of Mathematics & Computer Applications
Thapar University, Patiala.

22/7/09

Dr. R.K. Sharma
Dean of Academic Affairs
Thapar University
Patiala.

ACKNOWLEDGMENTS

First of all, I would like to thank the Almighty for granting perseverance. I would like to express my gratitude to **Dr. Rajesh Kumar Gupta and Dr**. **Amit Kumar**, **Lecturer**, **SMCA**, **Thapar University**, **Patiala**, for their patient guidance and support throughout this work. I was truly very fortunate to have the opportunity to work under them as a student. It was both an honor and a privilege to work with them. They also provide help in technical writing and presentation style and I found this guidance to be extremely valuable. I take this opportunity to express my sincere thanks to **Prof**. **S**.**S**.**Bhatia**, **Head SMCA**, **Thapar University**, **Patiala**, for their valuable support and help without which it would not have been possible for me to complete this work.

I am also thankful to all my friends who devoted their valuable time and helped me in all possible ways towards successful completion of this work. I do not find enough words with which I can express my feeling of thanks to the entire faculty and staff of SMCA, Thapar University, Patiala, for their help, inspiration and moral support which went a long way in successful completion of my work. I thank all those who have contributed directly or indirectly to this work.

Lastly, and more importantly, I would like to thank my parents for their years of unyielding love and encouragement. They have always wanted the best for me and I admire my parent's determination and sacrifice to put me through college.

Patiala

July 14, 2009 (Ankush Kumar)

ABSTRACT

This thesis is devoted to fault tree analysis of different systems, the qualitative and quantitative analysis of their fault trees. The fault tree analysis is a deductive failure based approach, where an undesired state of the system is specified and analyzed to find the ways in which the system can reach that state. The main topics are the introduction to fault tree analysis, qualitative and quantitative analysis of Computer System, Lathe Machine & Printing Circuit Board Assembly.

The chapter-wise summary of the thesis is as follows:

Chapter 1 is introductory in nature. This chapter includes basic definitions, operations and concepts used throughout the work.

Chapter 2 presents brief review of the work done in the area of fault tree analysis, qualitative and quantitative analysis of fault tree.

In **Chapter 3**, the basic reasons for the failure of computer system have been studied. Fault tree has been constructed on the basis of this study and qualitative and quantitative analysis has been done.

In **Chapter 4**, the basic reasons for the failure of tool broken of the lathe machine has been studied. Fault tree has been constructed on the basis of this study and qualitative and quantitative analysis has been done.

In the last chapter, the basic reasons for the failure of PCBA system have been studied. Fault tree has been constructed on the basis of this study and qualitative and quantitative analysis has been done.

TABLE OF CONTENTS

Chapter	*Page No.*
1. INTRODUCTION TO FAULT TREE ANALYSIS	1 - 15
2. LITERATURE REVIEW	16 - 20
3. FAULT TREE ANALYSIS OF COMPUTER SYSTEM	21 - 31
4. FAULT TREE ANALYSIS OF LATHE MACHINE	32 - 45
5. FAULT TREE ANALYSIS OF PRINTING CIRCUIT BOARD ASSEMBLY	46 - 65
REFERENCES	66-69

Chapter 1

INTRODUCTION TO FAULT TREE ANALYSIS

1.1 Introduction

The fault tree analysis is a deductive failure based approach, where an undesired state of the system is specified and analyzed to find the ways in which the system can reach that state. In the manufacturing industry, fault tree analysis (FTA) can be used as a diagnostic aid to detect when a failure occurs in a system. A fault tree may be used to identify the source of the failure by evaluating the combinations of basic events (fault symptoms) which may be a root event (a particular fault). The fault tree shows the relationship between the events and logical symbols such as the AND, OR, XOR, etc. are used to depict the relationship between the events. The two basic gate categories used in fault tree are the OR gate and the AND gate. Due to the fact that the gates relate the events in the same way as Boolean expressions, all the Boolean laws can be applied in the fault tree.

The most difficult part of creating a fault tree is the determination of the top level event. The selection of the top event is crucial since hazards in the systems will not be comprehensive unless the fault trees are drawn for all significant top level events. Once the top event has been defined, the next step is to determine the events related to the top event and the logical relations between them, using logical symbols to define the relations. The output of an AND gate only exists if all the inputs events exist. The output of an OR gate exists provided at least one on the input events exist. The relationships between the events are standard logical relations and can therefore be expressed using any form of Boolean algebra or truth table.

A successful FTA requires the following steps be carried out:

1. Identify the *objective* for the FTA.

2. Define the *top event* of the fault tree.

3. Define the *scope* of the FTA.

4. Define the *resolution* of the FTA.

5. Define *ground rules* for the FTA.

6. *Construct* the fault tree.

7. *Evaluate* the fault tree.

8. *Interpret* and present the results.

The first five steps involve the problem formulation for FTA. The remaining steps involve the actual construction of the fault tree, the evaluation of the fault tree, and the interpretation of the fault tree results. While most of the steps are performed sequentially, steps 3-5 can proceed concurrently.

The first five steps involve concurrently. The inter relationship of the eight steps are shown in Figure 1.1.

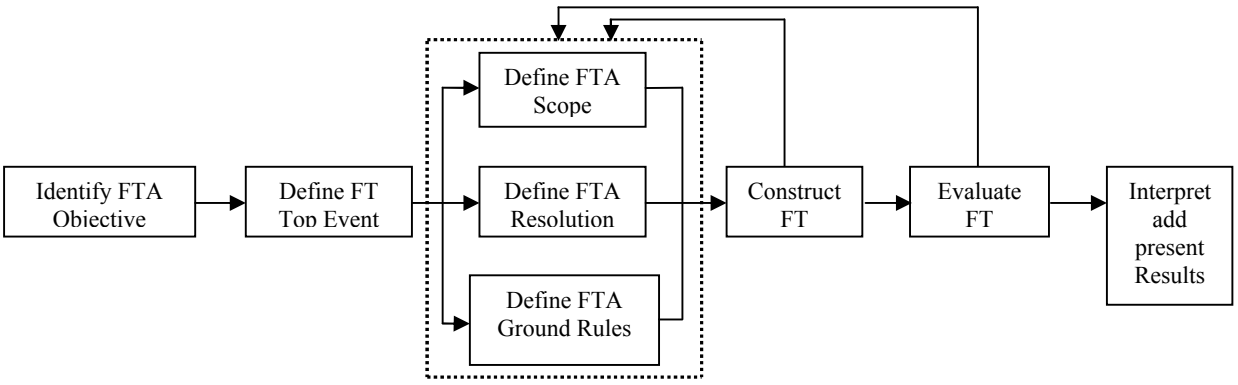


Figure 1.1: Inter relationship of the eight steps

The first step for a successful FTA is to define the objective of the FTA. The objective should be phrased in terms of a failure of the system to be analyzed.

Once the objective is defined, the top event of the fault tree is also defined in Step 2. The top event of the fault tree is the event for which the failure causes will be

resolved and the failure probability determined. The top event defines the failure mode of the system that will be analyzed.

In Step 3, the scope of the analysis is defined. The scope of the FTA indicates which of the failures and contributors will be included and which will not be included.

In Step 4 of the process, the resolution of the FTA is defined. The resolution is the level of detail The development of a quantitative model is based on the need to get the best possible estimate for the top event probability, considering the data and other information that are available. Fault trees are developed to a level of detail where the best failure probability data are available to which the failure causes for the top event will be developed.

In Step 5 any ground rules for the FTA are defined. These ground rules include the procedure and nomenclature by which events and gates are named in the fault tree. The naming scheme used is very important in creating an understandable fault tree.

In Step 6 involves the actual construction of the fault tree. The subsequent sections describe in detail the thinking and logic that is involved in constructing the fault tree from the system schematics and descriptions. The symbols that are used in the fault tree to represent the relationships between events are also described.

In Step 7 involves the evaluation of the fault tree. The evaluation includes both a qualitative and Cut sets are usually sorted by cut set order (the number of events in a cut set) to provide information on the combinations of basic events that can result in the top event. Finally, Eighth step involves the interpretation and presentation of the results. Emphasis is placed The top event of the fault tree directs all of the rest of the analysis. If the top event is incorrectly defined (and this happens a surprising number of times) then the FTA will be incorrect, which can result in wrong decisions being

made upon the interpretation and not simply on the presentation.

1.2 Basic paradigm in constructing a fault tree

The basic paradigm in constructing a fault tree is "think small", or more accurately "think myopically". For each event that is analyzed, the necessary and sufficient immediate events (i.e., the most closely related events) that result in the event are identified. Instead, a small step is taken and the immediate events that result in the event are identified. This smallest stepping provides the analyst with the insight into the relationships that are necessary and sufficient for the occurrence of the top event of the fault tree. This backward stepping ends with the basic causes identified that constitute the resolution of the analysis.

1.2.1 Top event: The top event of the fault tree directs all of the rest of the analysis. If the top event is incorrectly defined (and this happens a surprising number of times) then the FTA will be incorrect, which can result in wrong decisions being made. It is extremely important to define and understand the objectives of the analysis and the problem to be solved. It is often fruitful to define several potential top events and then decide the appropriate one or ones based on consultation with the decision maker and others involved.

In defining the top event, it is important to define the event in terms of the specific criteria that define the occurrence of the event. Generally to do this for a system failure, the system success criteria are first defined. Then failure of the system is defined as the failure to satisfy the given success criteria.

These points can be summarized as follows:

To define the top event, define the criteria for the occurrence of the event. For a system failure, first define the system success criteria.

1. Assure that the top event is consistent with the problem to be solved and the

objectives of the analysis.

2. If unsure of the top event, define alternative definitions that cover the top event and assess the applicability of each one.

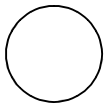
1.3 Faults vs. Failures

A distinction is made here between the rather specific word "failure" and the more general word "fault." Failures are basic abnormal occurrences, whereas faults are "higher order" or more general events. All failures are faults but not all faults are failures. The proper definition of a fault requires the specification of not only what the undesirable component state is but also when it occurs. These "what" and "when" specifications should be part of the event descriptions which are entered into the fault tree.

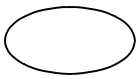
1.4 Symbology -The building blocks of the Fault Tree

1.4.1 Primary events

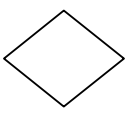
The primary events of a fault tree are those events, which, for one reason or another, have not been further developed. These are the events for which probabilities will have to be provided if the fault tree is to be used for computing the probability of the top event. These are:



Basic event: A basic initiating fault requiring no further development. These are found on the bottom tiers of the tree and require no further development or breakdown. There are no gates or events below the basic event.



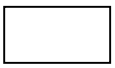
Conditioning event: Specific conditions or restrictions that apply to any logic gate.



Undeveloped event: An event which is not further developed either because it is of sufficient consequence or because information is unavailable.

External event: An event which is normally expected to occur.

1.4.2 Intermediate event symbols



Intermediate event: A fault event that occurs because of one or more antecedent causes acting through logic gates.

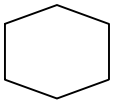
1.4.3 Gate symbols

And: represents a condition in which all the events shown below the gate (input gate) must be present for the event shown above the gate (output event). This event will occur only if all of the input events exist.

Or: represents a situation in which any of the events shown below the gate (input gate) will lead to the event shown above the gate (output event). The event will occur if only one or any combination of the input.

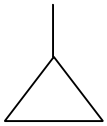
Exclusive or: Output fault occurs if exactly one of the input faults occurs.

Priority and: Output fault occurs if all of the input faults occur in a specific sequence (the sequence is represented by a CONDITIONING EVENT drawn to the right of the gate).

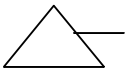


Inhibit: Output fault occurs if the (single) input fault occurs in the presence of an enabling condition.

1.4.4 Transfer symbols



Transfer in: Indicates that the tree is developed further at the occurrence of the corresponding TRANSFER OUT (e.g., on another page).



Transfer out– Indicates that this portion of the tree must be attached at the corresponding TRANSFER IN.

1.5 Basic rules for fault tree construction

The construction of the fault trees is a process that has evolved gradually over

a period of about 15 years. In the beginning it was thought of as an art, but it was soon realized that successful trees were all drawn in accordance with a set of basic rules. Observance of these rules helps to ensure successful fault trees so that the process is now less of an art and more of a science. We shall now examine the basic rules for successful fault tree analysis.

Consider Figure 1.2 it is a simple fault tree or perhaps a part of a larger fault tree. Note that none of the failure events have been "written in"; they have been designated just Q, A, B, C, D.

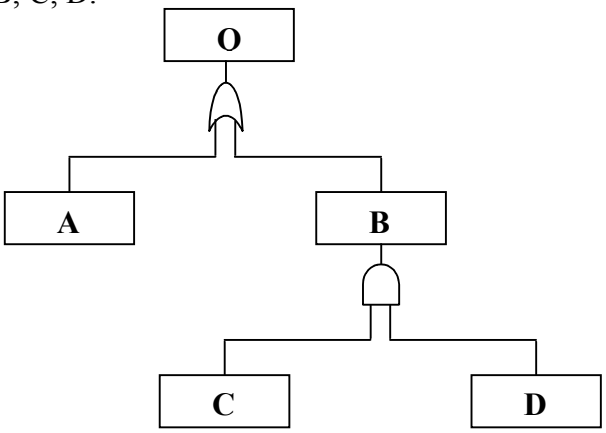


Figure 1.2: A simple fault tree

Now, when we have a specific problem in hand, it becomes necessary to describe exactly what such events such as Q, A, B, C, D actually are, and the proper procedure for doing this constitutes Ground Rule I:

Write the statements that are entered in the event boxes as faults; state precisely what the fault is and when it occurs.

The "what condition" describes the relevant railed (or operating) state of the component. The "when condition" describes the condition of the system-with respect to the component of the interest-which makes that particular state of existence of the component a fault.

The next step in the procedure is to examine each boxed statement and ask the question: "Can this fault consist of a component failure?" This question and its

answer leads us to Ground Rule II:

If the answer to the question, “Can this fault consist of a component failure? ” is “Yes” classify the event as a “ state-of-component fault.” If the answer is “ No,” classify the event as a” state-of-system fault.”

If the fault event is classified as “state-of-component,” add an OR-gate below classified as “state-of-system,” fault event may require an AND-gate, an OR-gate, an INHIBIT-gate, or possibly no gate at all. As a general rule, when energy originates from a point outside the component, the event may be classified as “state-of-system.”

In addition to the above ground rules, there are a number of other procedural statements that have been developed over the years. The first of these is the No Miracles Rule:

If the normal functioning of a component propagates a fault sequence, then it is assumed that the component functions normally.

We might find, in the course of a system analysis, that the propagation of a particular fault sequence could be blocked by the miraculous and totally unexpected failure of the some component. The correct assumption to make is that the component functions normally, thus allowing the passage of the fault sequence in question. However, if the normal functioning of a component acts to block the propagation of a fault sequence, then that normal functioning must be defeated by faults if the fault sequence is to continue up the tree. Another way of starting this is to say that, if an AND situation exists in the system, the model must take it into account.

Two other procedural statements address the dangers of not being methodical and attempting to shortcut the analysis process. The first is the Complete-the-Gate Rule: All inputs to a particular gate should be completely defined before further analysis of any one of them is undertaken.

The second is the No Gate-to-Gate Rule: Gate inputs should be properly defined fault events, and gates should not be directly connected to other gates. The Complete-the-Gate Rule states that the fault tree should be developed in levels, and each level should be completed before any consideration is given to a lower level. With regard to the No-Gate-to-Gate Rule, a "shortcut" fault tree is shown in Figure 1.3.

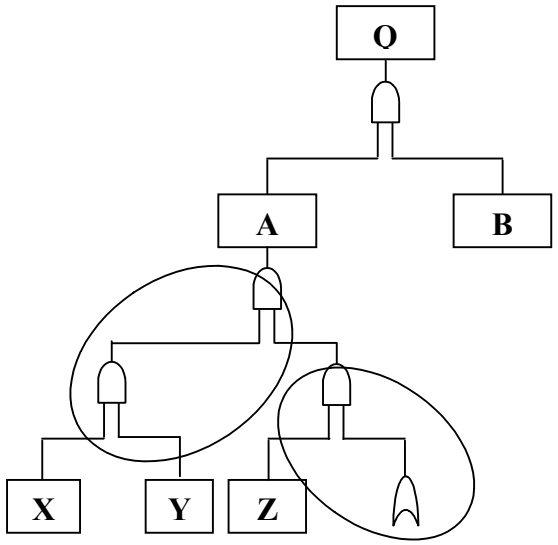


Figure 1.3: A short-cut fault tree

The "gate-to-gate" connection is indicative of sloppy analysis. The "gate-to-gate" shortcutting may be all right if a quantitative evaluation is being performed and the fault tree is being summarized. However, when the tree is actually being constructed, the gate-to-gate shortcuts may lead to confusion and may demonstrate that the analyst has an incomplete understanding of the system. A fault tree can be successful only if the analyst has a clear and complete understanding of the system to be modelled.

Rules for Boolean Algebra

(1a) $X \cap Y = Y \cap X$	$X \cdot Y = Y \cdot X$	Commutative
(1b) $X \cup Y = Y \cup X$	$X + Y = Y + X$	Law
(2a) $X \cap (Y \cap Z) = (X \cap Y) \cap Z$	$X \cdot (Y.Z) = (X.Y).Z$	Associative
(2b) $X \cup (Y \cup Z) = (X \cup Y) \cup Z$	$X + (Y + Z) = (X + Y) + Z$	Law

(3a) $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$	$X.(Y+Z) = (X.Y) + (X.Z)$	Distributive Law
(3a) $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$	$X + (Y.Z) = (X+Y).(X+Z)$	
(4a) $X \cap X = X$	$X.X = X$	Idempotent Law
(4b) $X \cup X = X$	$X + X = X$	
(5a) $X \cap (X \cup Y) = X$	$X.(X+Y) = X$	Law of Absorption
(5b) $X \cup (X \cap Y) = X$	$X + (X.Y) = X$	
(6a) $X \cap X' = \phi$	$X.X' = \phi$	Complement ation
(6b) $X \cup X' = 1$	$X + X' = 1$	
(6c) $(X')' = X$	$(X')' = X$	
(7a) $(X \cap Y)' = X' \cup Y'$	$(X.Y)' = X' + Y'$	De-Morgan's Law
(7b) $(X \cup Y)' = X' \cap Y'$	$(X+Y)' = X'.Y'$	
(8a) $\phi \cap X = \phi$	$\phi.X = \phi$	Operations with ϕ
(8b) $\phi \cup X = X$	$\phi + X = X$	
(9a) $X \cup (X' \cap Y) = X \cup Y$	$X + (X'.Y) = X + Y$	
(9b) $X' \cap (X \cup Y') = X' \cap Y' = (X \cup Y)'$	$X'.(X+Y') = X'.Y' = (X+Y)'$	

1.6 Minimal cut set

A minimal cut set is a smallest combination of component failures which, if they all occur, will cause the top event to occur. By the definition, a minimal cut set is thus a combination of primary events sufficient for the top event. The combination is a "smallest" combination in that all the failures all the failures are needed for the top event to occur; if one of the failures in the cut set does not occur, then the top event will not occur.

Any fault tree will consist of a finite number of minimal cut sets, which are unique for the top event. The one-component minimal cut sets, if there are any, represent those single failures which will cause the top event to occur. The two-component minimal cut sets represent the double failures which together will cause the top event to occur. For an n-component minimal cut set, all n components in the

cut set must fail in order for the top event to occur. The minimal cut set expression for the top event can be written in the general form,

$$T = M_1 + M_2 + \cdots + M_k$$

Where T is the top event and M_i are the minimal cut sets. Each minimal cut set consists of a combination of specific component failures, and hence the general n-component minimal cut can be expressed as

$$M_n = X_1 \cdot X_2 \cdots X_n$$

Where $X_1, X_2, \ldots, X_n$ are component failures on the tree.

To determine the minimal cut sets of a fault tree, the tree is first translated to its equivalent Boolean equations and then "top-down" or "bottom-up" substitution method is used. The methods are straightforward and they involve substituting and expanding Boolean expressions. Two Boolean laws, the distributive law and the law of absorption, are used to remove the redundancies.

Consider the simple fault tree shown in Figure 1.4 the equivalent Boolean equations are shown below the tree.

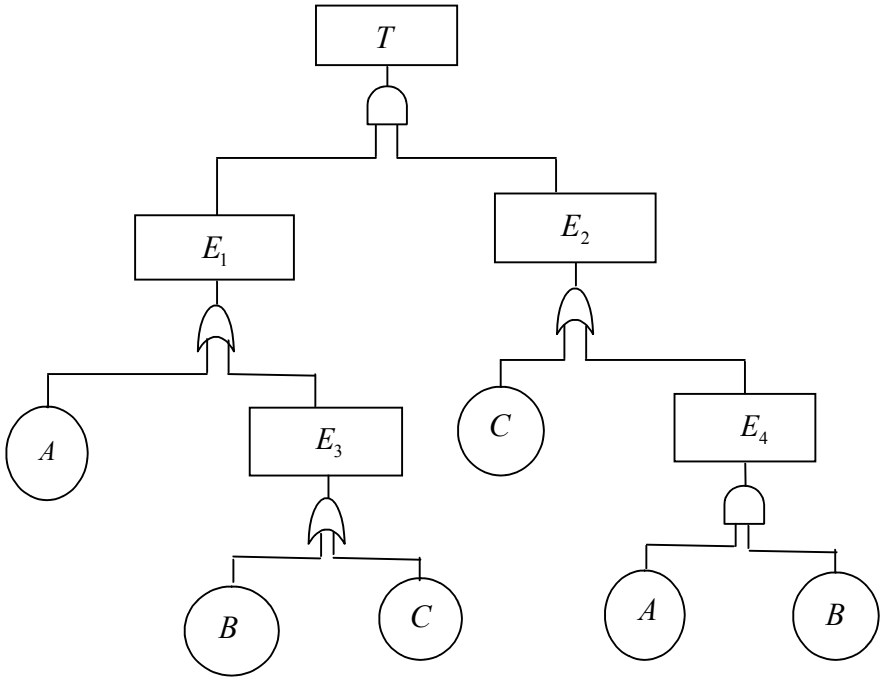


Figure 1.4: Simple fault tree

$$T = E_1 \cdot E_2,$$
$$E_1 = A + E_3,\ E_3 = B + C,\ E_2 = C + E_4,\ E_4 = A \cdot B$$

Because E_4 has only basic failures, substitute into E_2 to obtain

$$E_2 = C + A.B$$

The minimal cut sets of E_2 are thus C and E_3 is already in reduced from having minimal cut sets B and C. Substituting into E_1, we obtain $E_1 = A + B + C$ so E_1 has three minimal cut sets A, B and C. Finally, substituting the expressions for E_1 and E_2 into the equation for T, expanding and applying the absorption law, we have

$$T = (A + B + C).(C + A.B)$$
$$T = A.C + A.A.B + B.C + B.A.B + C.C + C.A.B$$
$$T = A.C + A.B + B.C + A.B + C + A.B.C \qquad \text{(Idemponent Law)}$$
$$T = C + A.B \qquad \text{(Absorption \& Idemponent Law)}$$

The minimal cut sets of the top event are thus again C and $A.B$.

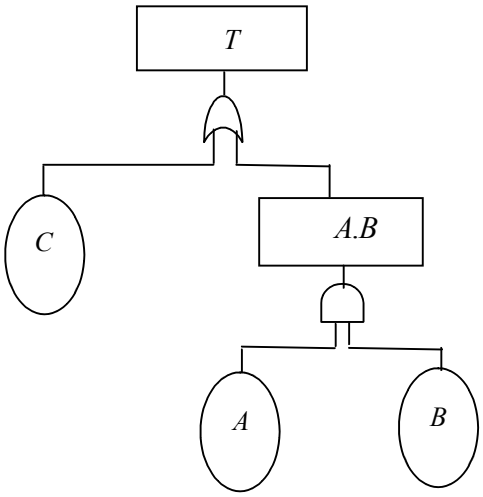


Figure 1.5: Minimal cut set

The Boolean techniques discussed in this chapter have immediate practical importance in relation to fault trees. A fault tree can be thought of as a pictorial representation of those Boolean relationships among fault events that cause the top

event to occur. In fact, a fault tree can always be translated into an entirely equivalent set of Boolean equations. Thus an understanding of the rules of Boolean algebra contributes materially toward the construction and simplification of fault trees. Once a fault tree has been drawn, it can be evaluated to yield its qualitative and quantitative characteristics. These characteristics cannot be obtained from the fault tree, but they can be obtained from the equivalent Boolean equations. In this evaluation process the algebraic reduction techniques can be used.

1.7 Qualitative and quantitative evaluations of a fault tree

Both qualitative and quantitative evaluations can be performed on a fault tree. The fault tree itself is a qualitative assessment of the events and relationships that lead to the top event. In constructing the fault tree, significant insights and understanding are gained concerning the causes of the top event. Additional evaluations serve to further refine the information that the fault tree provides.

The qualitative evaluations basically transform the fault tree logic into logically equivalent forms that provide more focused information. The principal qualitative results that are obtained are the minimal cut sets (MCSs) of the top event. A cut set is a combination of basic events that can cause the top event. An MCS is the smallest combination of basic events that result in the top event. The basic events are the bottom events of the fault tree. Hence, the minimal cut sets relate the top event directly to the basic event causes. The set of MCSs for the top event represent all the ways that the basic events can cause the top event. A more descriptive name for a minimal cut set may be “minimal failure set.” The set of MCSs can not only be obtained for the top event, but for any of the intermediate events (e.g., gate events) in the fault tree.

A significant amount of information can be obtained from the structure of

MCSs. Any MCS with one basic event identifies a single failure or single event that alone can cause the top event to occur. These single failures are often weak links and are the focus of upgrade and prevention actions. Examples of such single failures are a single human error or single component failure that can cause a system failure. An MCS having events with identical characteristics indicates a susceptibility to implicit dependent failure, or common cause, that can negate a redundancy. An example is an MCS of failures of identical valves. A single manufacturing defect or single environmental sensitivity can cause all the valves to simultaneously fail.

The quantitative evaluations of a fault tree consist of the determination of top event probabilities and basic event importance. Uncertainties in any quantified result can also be determined. Fault trees are typically quantified by calculating the probability of each minimal cut set and by summing all the cut set probabilities. The cut sets are then sorted by probability. The cut sets that contribute significantly to the top event probability are called the dominant cut sets. While the probability of the top event is a primary focus in the analysis, the probability of any intermediate event in the fault tree can also be determined. Different types of probabilities can be calculated for different applications. Top event frequencies, failure or occurrence rates, and availabilities can also be calculated. These characteristics are particularly applicable if the top event is a system failure.

In addition to the identification of dominant cut sets, importance of the events in the fault tree are some of the most useful information that can be obtained from fault tree quantification. Quantified importance allows actions and resources to be prioritized according to the importance of the events causing the top event. The importance of the basic events, the intermediate events, and the minimal cut sets can be determined. Different importance measures can be calculated for different

applications. One measure is the contribution of each event to the top event probability. Another is the decrease in the top event probability if the event were prevented from occurring. A third measure is the increase in the top event probability if the event were assured to occur. These importance measures are used in prioritization, prevention activities, upgrade activities, and in maintenance and repair activities.

Chapter 2

LITERATURE REVIEW

The Fault Tree Analysis technique was developed by HR Watson of Bell Telephone Laboratories in 1961 [33]**.** FTA was first applied to anti-ballistic systems. Boeing further developed and refined the process becoming the foremost proponents of the method. In June 1965, the Boeing Company and the University of Washington organized a symposium on safety analysis and system safety in Seattle, where a large number of papers were presented by Boeing employees including Haasl, Michels [11, 25]. Since then, the concepts and techniques used in fault tree analysis continued to be developed. FTA is regarded as an established methodology for the safety and reliability study of large and complex systems. 1966, Boeing developed a computer fault tree simulation program called BACSIM (Boeing Aerospace Corporation Simulation) for the evaluation of multi-phase fault trees. In 1972 to 1974, Fussell and Vesely developed the top down cut set generation algorithm called MOCUS (Method of obtaining cut sets) [7, 8, 9, 30]. In 1973, a fault tree construction methodology for electrical systems from the failure transfer function of system components has been introduced in [7]. In 1975, nuclear engineers took the next step by augmenting a fault tree for nuclear power plants with the probabilities of those faults occurring and a measure of the severity of the consequences of each failure. By adding and multiplying the probabilities, they came up with an overall probability of failure with a given level of severity. In 1976, event tree/fault tree analysis was used to determine the chances of plant accidents. An "event tree" defines an initial failure somewhere within a system and then traces, like a

genealogy, the consequences of that failure [21]. The fault tree is a graphical method of all the possible combination of the faults that would result into the undesired event [30].

Fault tree analysis [30] (1981) is a deductive safety analysis technique which is applied during the design phase. It is a top-down approach; input consists of knowledge of the system's functions as well as its failure modes and their effects. The result of the analysis is a set of combinations of component failures that can result in a specific mal function. There are several variations and extensions. The goal of the analysis is to find all the minimal cut sets, where a *minimal cut set* is a smallest combination (intersection) of component failures (basic fault events) which, if they all occur, will cause the top event to occur. The combination is a "smallest" combination in that all the failures are needed for the top event to occur; if one of the failures in the cut set does not occur, then the top event will not occur (by this combination) [30](1981).

The faults that are depicted can be due to hardware failures, human errors, software errors or any other event which might lead to the failure of the system. In 1988, nine FTA publications related to FTA appeared. The fault tree synthesis algorithm of [16](1986) was based on a mini fault tree. Bossche [2] used the 'relations' between process variables and component states to model the fault tree of a system. De Vries [4] developed a quantitative procedure to fault tree from a schematic diagram of electrical systems. In [31] another methodology based on extended decision tables of system components was presented.

Essentially, fault tree analysis is a qualitative model which reveals the possible combinations of identified basic events sufficient to cause the undesired top event. But it is also often used in probabilistic analyses, such as failure rate

calculation of the top event given the failure rates of basic fault events [8] two phases are involved in the technique: Fault Tree Synthesis (FTS) and Fault Tree Analysis (FTA). Fault tree synthesis involves the construction of fault trees, and typically 36 details analysis of the target system. The second phase, fault tree analysis, involves analysing and manipulating fault trees. This can be useful as a design tool, helping to identify weak points in the target system where there might be potential sources of accident or breakdown.

There are a lot of publications concerning various aspects of description, construction, and qualitative and quantitative analysis of fault trees. Therefore, several papers are mentioned that represent state-of-the-art of the methods of fault tree analysis. Methods of estimation of probability of setting in of a top event not using minimum cut sets, are proposed in [1, 26]. Algorithms of analysis of fault trees with so-called replicated gates are developed in [3, 28]. Methods of trees modularization are considered in [18]. A method of multilevel representation of fault trees that significantly accelerates calculations is proposed in [12] for analysis of large fault trees containing a significant number of replicated and negated gates. The state-of-the-art of the studies is presented in [19, 20, 27]. In all the above-mentioned publications, fault trees can be used to describe structure and operation of dichotomous systems only, i.e., systems whose states can be divided into "serviceable" and "failed." If the main criterion of system operation is its efficiency, and a failure in a part of equipment does not result in a failure of the system but only reduces the efficiency of operation of the system and its subsystems, then such a model cannot be used directly.

In the last decade, some useful methods have been introduced to the constru-

ction of fault trees from a system topology diagram. The works explained in [32] and [29] are examples of this type of methods. This paper introduces a novel algorithm for the construction and analysis of fault trees. Once the fault tree is constructed, the Min Cut Set (MCS) approach is used in the quantitative analysis of the fault tree. The Minimal Cut Set approach is used for solving static fault trees.

A static fault tree consists only of static gates like AND gates and OR gates. The minimal cut set is a unique combination of component failures that can lead to system failure. A cut set is said to be a minimal cut set, if when one of the components is removed from the set, then the remaining is no longer a cut set.

According to Kales [14], Ejlali and Miremadi [5], Zemva and Zajc [34] fault tree analysis is a powerful diagnosis technique and is used widely for demonstrating the root causes of undesired event in system failure, and logical functional relationship among components, manufacturing processes, and subsystems of a system. As Li and Zuo [22], Khan and Abbasi [17] and Hu et al. [13] pointed out that fault tree analysis is suitable for the manufacturing system/process fault diagnosis because it has the following characteristics:

(a) FTA can be used deeply to analyse a specific fault level by level. It uses clear graphics to vividly give visibility to fault dependencies between the part/component faults and the manufacturing system/process fault.

(b) The fault tree can clearly point out aspects of the system/process relevant to significant failures. It can also be seen whether a part/component fault will cause a system/process fault, what the effect is, how great the effect is, and its mechanism.

(c) A fault tree is a clear illustration for those management and maintenance personnel who have never participated in the system/process design and trial-

manufacture, which will greatly shorten the training time of the maintenance personnel, and therefore cut down the expense for personnel training.

(d) The qualitative analysis of fault trees a system/process may make the designers with an insight into the system/process behaviour, so as to be able to find out the weak links in the design scheme, and take corrective measures. FTA is now widely used in many fields, such as in the semi-conductor industry [22], flexible manufacturing system [13], and chemical and aviation industries [17].

Chapter 3

FAULT TREE ANALYSIS OF COMPUTER SYSTEM

3.1 Introduction

Figure 3.1 shows a computer system. The power required for the total operation is provided either from the mains or, if the mains fail, from an auxiliary genset (or from an uninterrupted power supply unit). The cooling system consists of a compressor unit, a heat exchanger, and an expander with coolant pipes. The compressor unit receives its power either from the mains or from the genset. The heat exchanger receives its water supply from one of the three alternative sources, namely, metro, overhead tank, or sump and pump system. The pump is supplied power either from the mains or from the genset. The computer is not allowed to function if there is no supply of cold air.

In this chapter, the basic reasons for the failure of computer system have been studied. Fault tree has been constructed on the basis of this study and qualitative and quantitative analysis has been done.

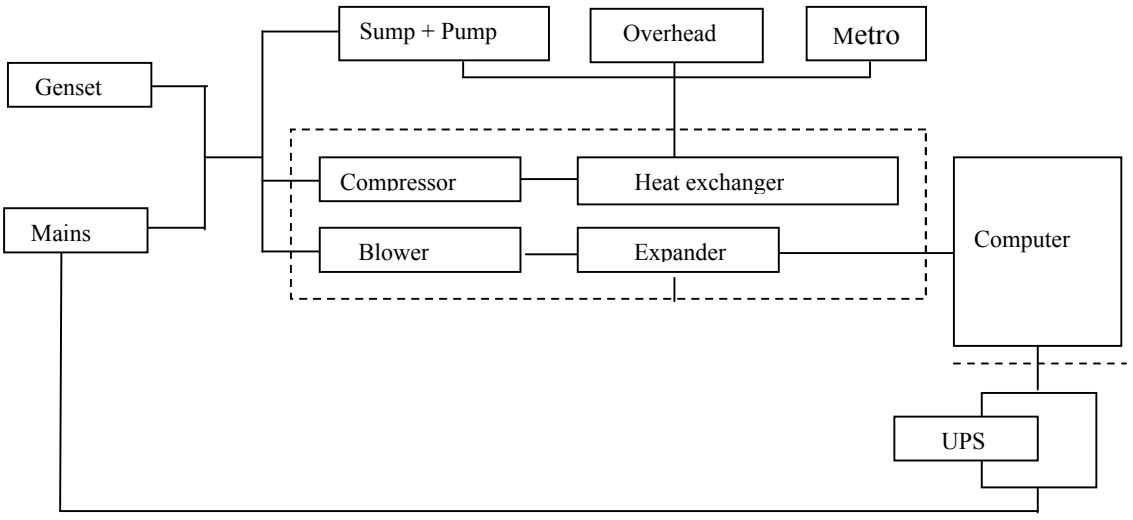


Figure 3.1: computer system

3.2 Construction of a fault tree

Figures 3.2 to 3.12 presents the fault tree constructed for the problem under study. The top event for this problem is the computer not functioning. The computer functioning failure can be broadly classified into two parts: no power and no cold air failure. The logical relation between these two intermediate events and the top event is represented using or gate. The power failure and cold air failure are the first level events which cause the system to fail. The causes for a failure observed during computer functioning failure shown in Figure 3.2.

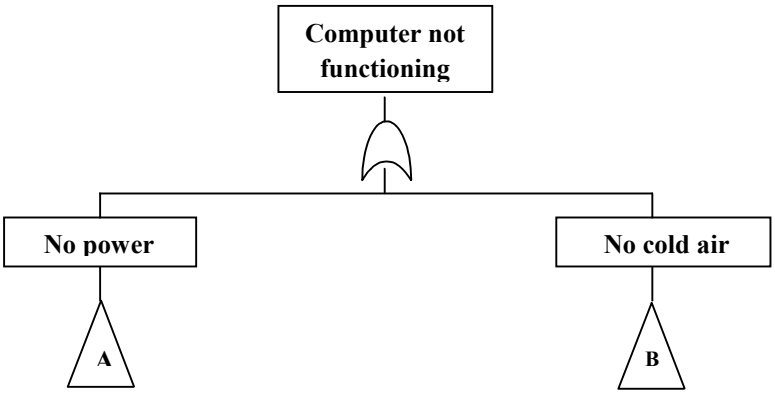


Figure 3.2: Fault tree for top event computer not Functioning

The no power can be attributed to two causes: mains out and UPS out. These causes are shown in Figure 3.3. The logical relationship between these two intermediate events and no power event is represented using an AND gate.

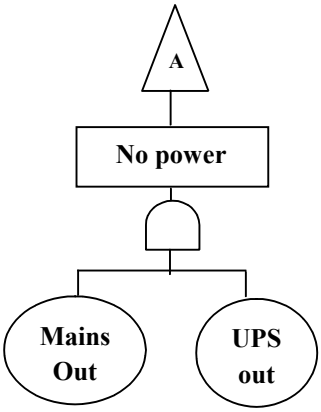


Figure 3.3: Fault tree for no power

The no cold air can be attributed to three causes: blower not working, no cooling water supply and compressor not functioning. These causes are shown in Figure 3.4. The logical relationship between these three intermediate events and no cold air event is represented using an OR gate.

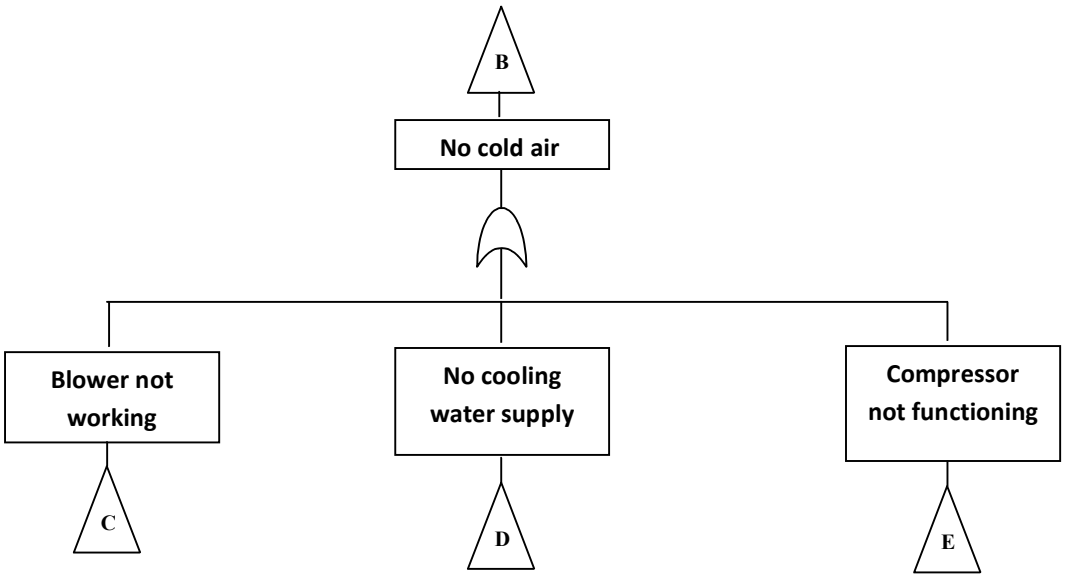


Figure 3.4: Fault tree for no cold air

The blower not working can be attributed to two causes: blower defective and no power to blower. These causes are shown in Figure 3.5. The logical relationship between these two intermediate events and blower not working event is represented using an OR gate.

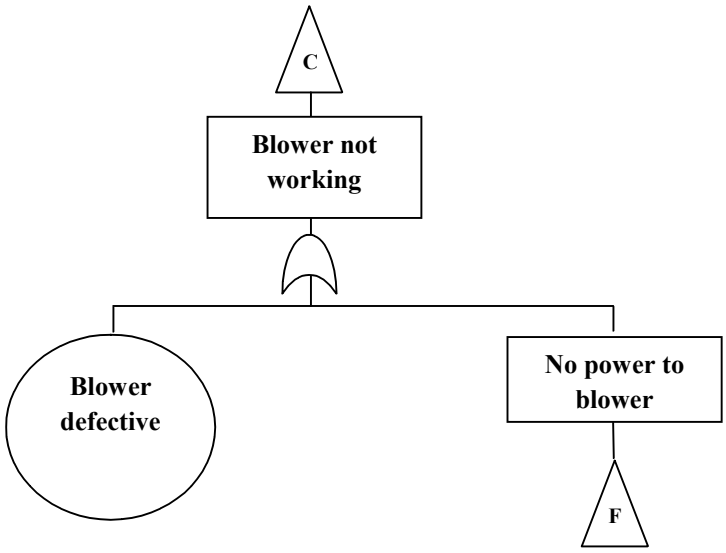


Figure 3.5: Fault tree for blower not working

The no power to blower can be attributed to two causes: mains out and genset out. These causes are shown in Figure 3.6. The logical relationship between these two intermediate events and no power to blower event is represented using an AND gate.

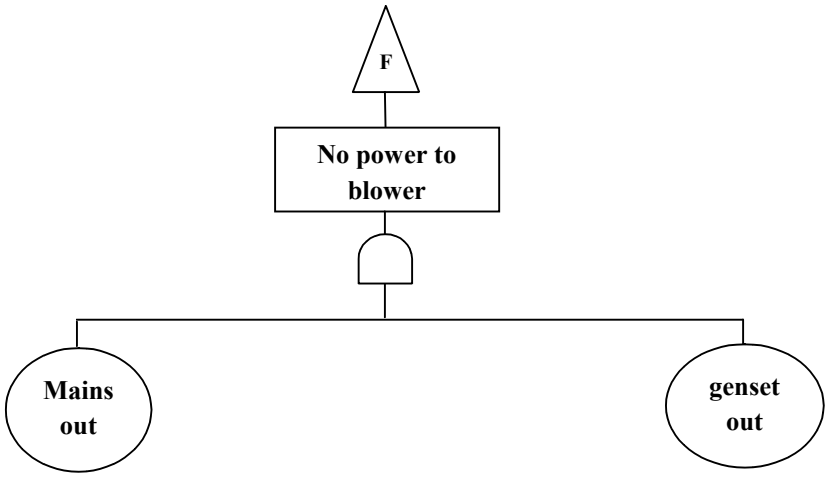


Figure 3.6: Fault tree for no power to blower

The no cooling water supply can be attributed to three causes: no metro water, over head tank empty and no supply from sump. These causes are shown in Figure 3.7. The logical relationship between these three intermediate events and no cooling water supply is represented using an AND gate.

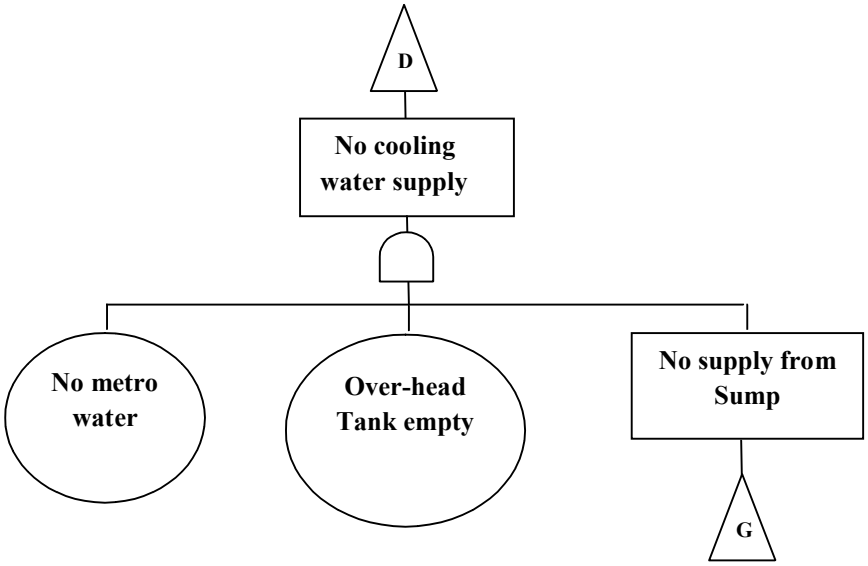


Figure 3.7: Fault tree for no cooling water supply

The no supply from sump can be attributed to two causes: sump empty and pump not working. These causes are shown in Figure 3.8. The logical relationship

between these two intermediate events and no supply to sump event is represented using an OR gate.

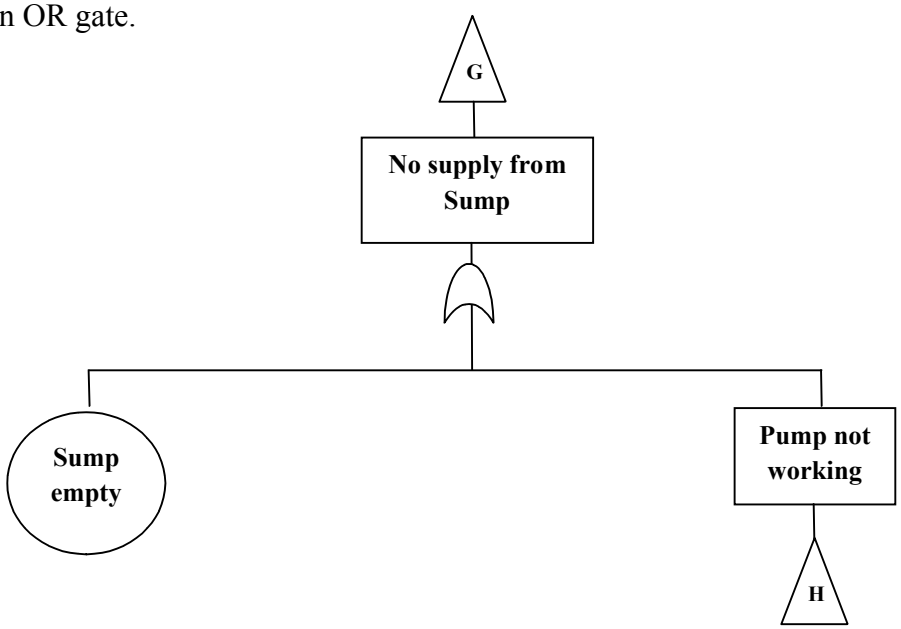


Figure 3.8: Fault tree for no supply from sump

The pump not working can be attributed to two causes: pump is out and no power to pump. These causes are shown in Figure 3.9. The logical relationship between these two intermediate events and pump not working event is represented using an AND gate.

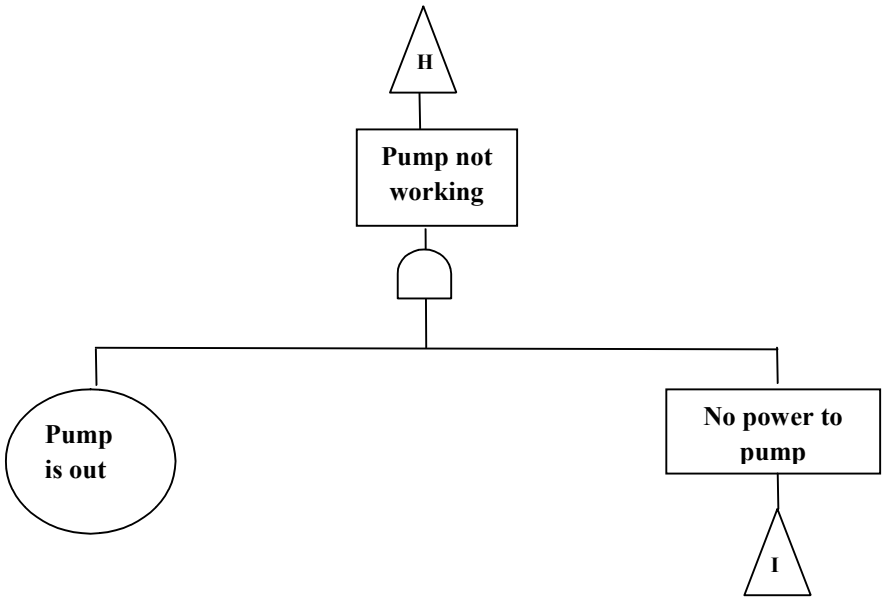


Figure 3.9: Fault tree for pump not working

The no power to pump can be attributed to two causes: mains out and genset out. These causes are shown in Figure 3.10. The logical relationship between these two intermediate events and no power to pump event is represented using an AND gate.

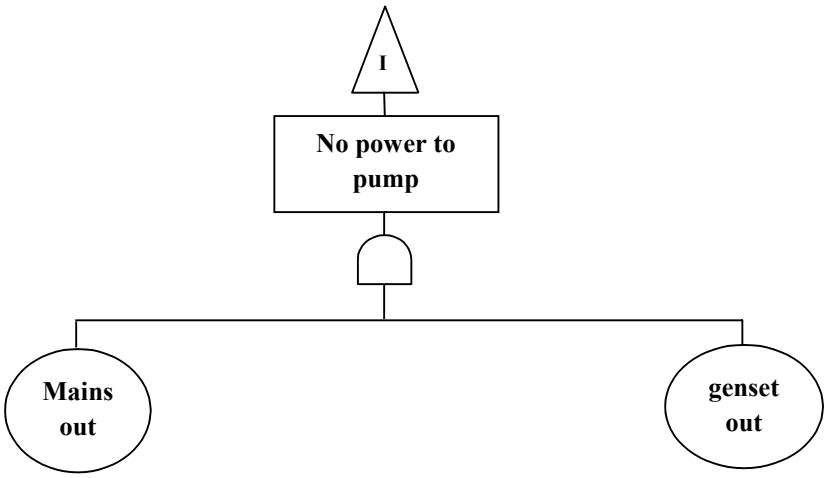


Figure 3.10: Fault tree for no power to pump

The compressor not functioning can be attributed to two causes: compressor out and no power. These causes are shown in Figure 3.11. The logical relationship between these two intermediate events and compressor not functioning event is represented using an OR gate.

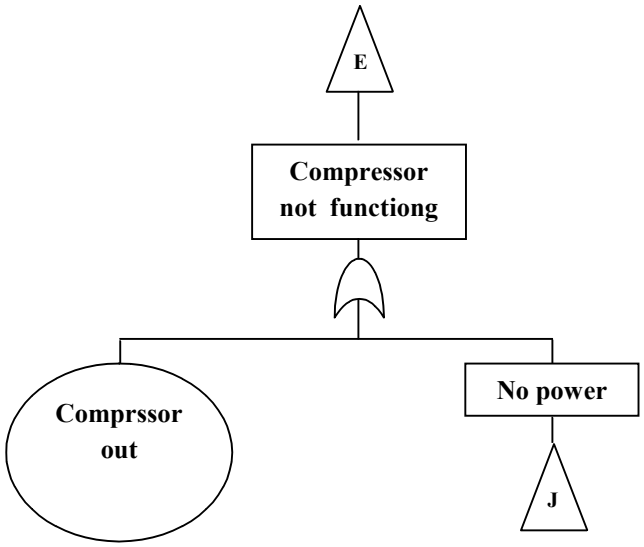


Figure 3.11: Fault tree for compressor not functioning

The no power can be attributed to two causes: mains out and genset out. These causes are shown in Figure 3.12. The logical relationship between these two

Intermediate events and no power event is represented using an AND gate.

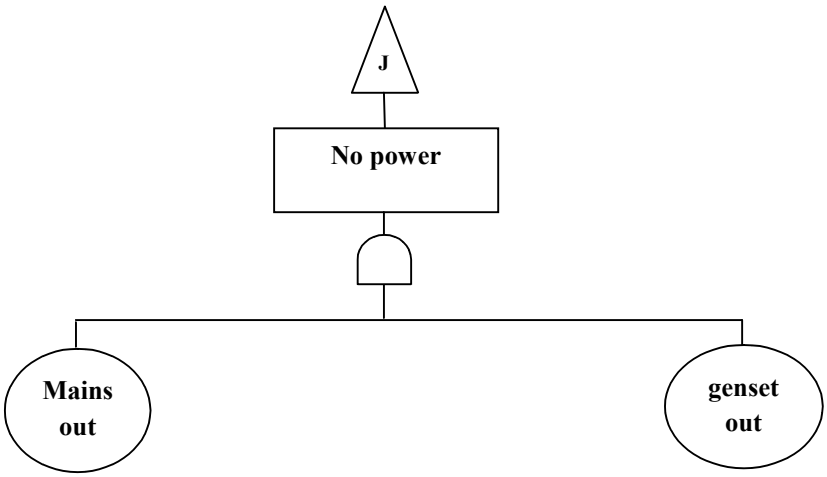


Figure 3.12: Fault tree for no power

3.3 Qualitative evaluation of the fault tree

In this section qualitative analysis of computer system has been presented.

3.3.1 Notations

For the qualitative analysis of computer system, the different basic events, due to which computer system may fail, are represented by $V_1,\ldots,V_9$ which are as follows:

V_1: represents the event that the mains out.

V_2: represents the event that UPS out.

V_3: represents the event that the blower is defective.

V_4: represents the event that the genset out.

V_5: represents the event that there is no metro water.

V_6: represents the event that the overhead tank is empty.

V_7: represents the event that the sump is empty.

V_8: represents the event that pump is out.

V_9 : represents event that compressor is out.

F_1 : represents the event that there is no power supply to the computer system.

F_2 : represents the event that there is no power supply to the blower.

F_3 : represents the event that blower is not working.

F_4 : represents the event that pump is not working.

F_5 : represents the event that there is no supply from sump.

F_6 : represents the event that there is no supply of cooling water.

F_7 : represents the event that compressor is not functioning.

F_8 : represents the event that no supply of cold air.

F : represents the event that computer not functioning.

3.3.2: Fault tree

The fault tree of computer system has been shown in fig. 3.13

3.3.3: Qualitative analysis

In this sub section using the sub fault tree's and Boolean algebra, the expressions for evaluating the probability of failure of computer systems has been obtained which are described below:

$$F_1 = V_1 V_2 \qquad (\text{using Figure } 3.3)$$

$$F_2 = V_1 V_4 \qquad (\text{using Figure } 3.6)$$

$$F_3 = V_3 + V_1 V_4 \qquad (\text{using Figure } 3.5)$$

$$F_4 = V_8 (V_1 V_4) \qquad (\text{using Figure } 3.9)$$

$$F_5 = V_7 + (V_8 (V_1 V_4)) \qquad (\text{using Figure } 3.8)$$

$$F_6 = V_5 V_6 (V_7 + V_8 V_1 V_4) \qquad (\text{using Figure } 3.7)$$

$$F_7 = V_9 + V_1 V_2 \qquad (\text{using Figure } 3.11)$$

$$F_8 = (V_3 + V_1 V_4) + (V_5 V_6 (V_7 + V_8 V_1 V_4)) + V_9 + V_1 V_2 \qquad (\text{using Figure } 3.4)$$

$$= V_3 + V_1 V_4 + V_5 V_6 V_7 + V_5 V_6 V_8 V_1 V_4 + V_9 + V_1 V_2$$

$$F = V_1 V_2 + V_3 + V_1 V_4 + V_5 V_6 V_7 + V_5 V_6 V_8 V_1 V_4 + V_9 + V_1 V_2 \qquad (\text{using Figure } 3.13)$$

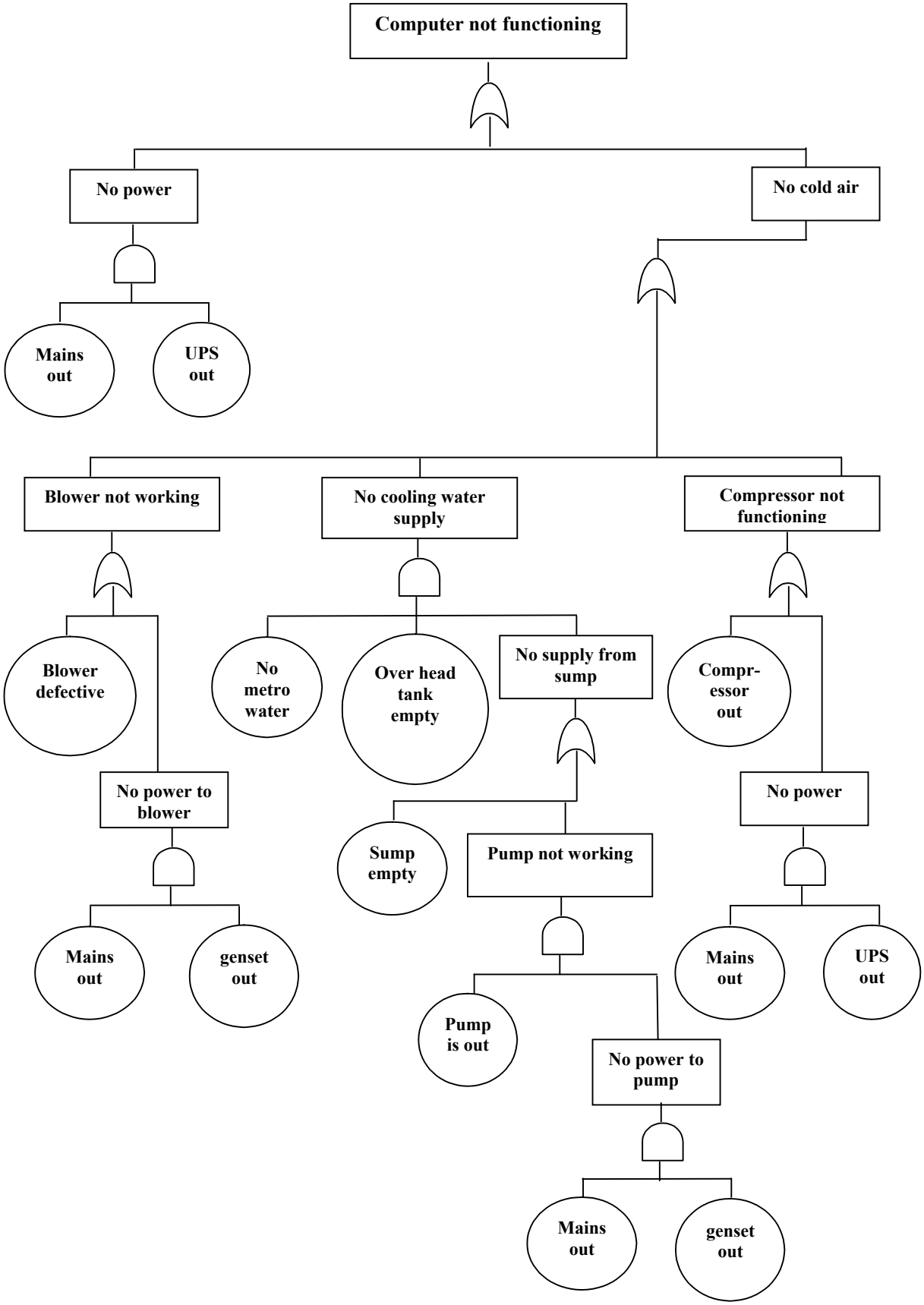


Figure: 3.13 Computer system

Using absortion law i.e., $V_5V_6V_8V_1V_4 + V_1V_4 = V_1V_4$

Finally, the failure is given by:

$$F=V_1V_2+V_3+V_1V_4+V_5V_6V_7+V_9$$

3.4 Quantitative analysis of the fault tree

For the quantitative evaluation of fault tree of computer system the probability of failures of different components are collected and shown in Table 3.1.

Table 3.1 The descriptions of the sub-events of a computer system:

S.NO	Code	Fault	Probability of failure
1	V_1	Mains out	0.120
2	V_2	UPS out	0.070
3	V_3	Blower defective	0.080
4	V_4	Genset out	0.050
5	V_5	No metro water	0.090
6	V_6	Over-head tank empty	0.040
7	V_7	Sump empty	0.130
8	V_8	Pump is out	0.070
9	V_9	Compressor out	0.860

Putting all the values of $V_1,\ldots,V_9$ in equation (3.1).

The probability of failure P(F) of computer not functioning is:

$$P(F)=1-(1-V_1V_2)(1-V_1V_4)(1-V_3)(1-V_5V_6V_7)(1-V_9) \quad (3.1)$$
$$= 0.120817233$$

3.5 Result and Discussion

The probability of failures of different components in increasing order is shown in Figure 3.14. It will help the user in case of failure of computer system i.e. if a computer system fails then a user should check first the component with maximum probability of failure. If there is no fault in that component then user should check next component and so on.

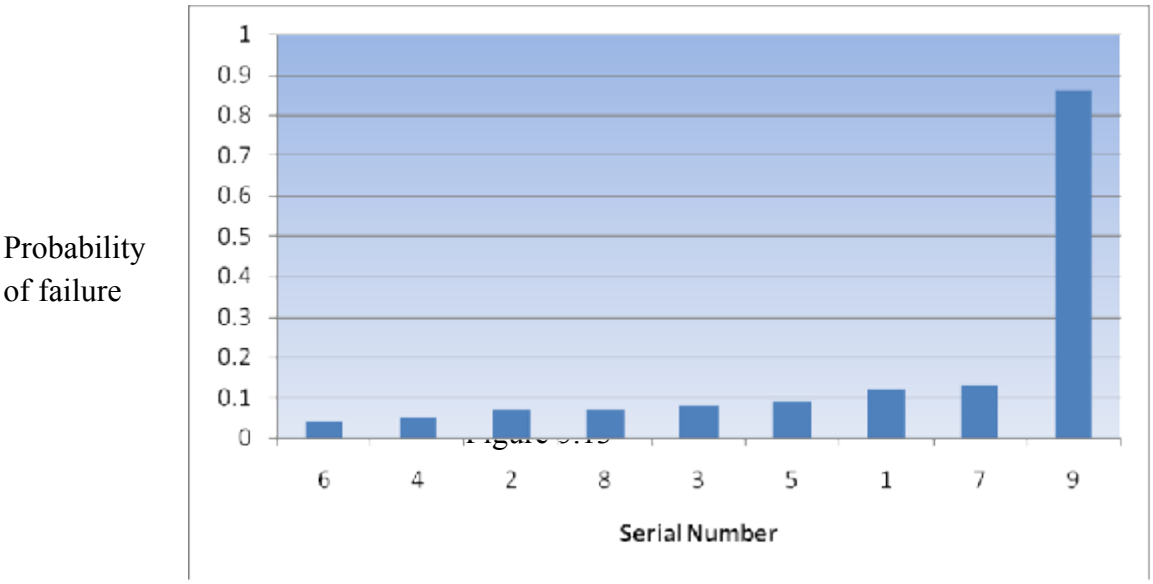


Figure 3.14

3.6 Conclusion

In the first part of this chapter, the basic reasons for the failure of computer system have been studied and fault tree has been constructed on the basis of this study which results in the qualitative analysis of the computer system. Further, data has been collected to perform the quantitative analysis. After the collection of the data, quantitative analysis has been done.

Chapter 4

FAULT TREE ANALYSIS OF LATHE MACHINE

4.1 Introduction

A machine tools is a power driven machine for making articles of a given shape, size and accuracy by removing metal from work pieces in the form of chips. Machine tools are factory equipment for producing machines, instruments and tools of all kinds. So, it can be said that the machine tool is the mother of all machines. Hence, the size of a country's stock of machine tools and their technical quality and condition largely characterize its industrial potential.

Most machine tools perform the following four functions;

1. Hold the job.
2. Hold the cutting tools.
3. Move one or both of these.
4. Provide a feeding motion for one of these.

A lathe is one of the oldest and perhaps most important machine tools ever developed. The job to be machined is rotated (turned) and the cutting tool moved relative to the job. That is why, the lathes are also called as " Turning machines" if the tools moves parallel to the axis of the rotation of the work piece, cylindrical surface is produced, while, if it moves perpendicular to this axis, it produce the flat surface.

A lathe was basically developed to machine cylindrical surfaces. A lathe is called "the father of the entire machine tool family".

`In this chapter the basic reasons for the failure of tool broken of the lathe machine have been studied. Fault tree has been constructed on the basis of this study and qualitative and quantitative analysis has been done.

4.2 Lathe machines

It is so called because the first of this type of lathe was driven by a steam engine. It is also called “centre lathe”, because, it has two centres between which the job can be held and rotated. A high percentage of all lathe work is turned between centres. The main parts of a centre lathe are: Bed, Head stock, Tail stock, Carriage, Tool post.

1. Bed. The bed is the base of foundation of the lathe. It is a heavy and rigid casting made in one piece to resist deflection and vibrations. It holds or supports all other parts, that is, head stock, tail stock and carriage etc. The top of the bed is planned to form “guides” or “ways”. Ways are accurate rails which support carriage and the tailstock. More expensive lathes have combination of V ways and flat ways. Less expensive lathes have flat ways. Directly under the front way on the bed is a rack for moving the carriage when the hand wheel is turned. The bed is usually fastened to steel legs so that the lathe can be bolted to the shop floor.

2. Head stock. The headstock assembly is permanently fastened to the left hand end of the lathe. It serves to support the first operative unit of the lathe, that is, the spindle. The spindle revolves in bearing, one at each end of the headstock. The spindle is rotated by a combination of gears and cone pulleys or by gears alone. Present day lathes have individual motor drives and most of them have geared headstocks. The steel spindle is hollow to take long bar stock. The spindle has definite taper at the front end for holding centres and other tools having a tapered shank. The hole through the spindle makes it possible to use a knockout bar to remove such tools. Work–

holding attachments such as driving plate, face plate or various types of chucks may be mounted on the threaded spindle nose. Some type of work may also be held in a collect which is inserted into hollow headstock spindle. A taper sleeve fits into the taper spindle hole. The headstock or live centre fits into the sleeve. This centre is called live centre because it turns with the work. The centre is tapered metal part with a pointed end. This supports the end of the work. piece as it turns between the centres. All centres points have a 60-degree included angle.

3. Tail stock. Tail stock is on the other end of the bed from the head stock. Its chief function is to hold the dead centre so that long work pieces can be supported between centres. It can be moved along the bed and clamped to the bed at various desired locations to suit the length of the work piece.

Tailstock consists of two main parts. The lower part rests directly on the bed ways, and the upper part rests on the lower part. Adjusting screws hold the two parts together. The upper casting can be moved toward or away from the operator to offset the tailstock for taper turning and to realign the tailstock centre for straight turning. The body of the tailstock has a bore for the hollow cylindrical sliding member, known as a “quill”. This quill sometime called as “tailstock spindle” even though it cannot rotate. The quill moves in and out of the tailstock bore when the tailstock hand wheel is turned. Once set, the quill may be clamped to remain in a desired position. The quill has a taper hole into which the dead centre is fitted. Drills, reamers, taps and other end cutting tools are held end fed to the work piece by the quill, the shanks of the tools being held in the tapered hole of the quill.

4. Tool post. The tool post is mounted on the compound rest and slides in a T-slot. Cutting tool/tool holder is firmly held in it. The tool can be swivelled as well as tilted by means of a rocker and a concave ring collar,

4.3 Construction of a fault tree

Figures 4.1 to 4.10 presents the fault tree constructed for the problem under study. The top event for this problem is the tool broken. The tool broken can be broadly classified into seven parts: blunt tool, feed not proper, improper speed, improper mounting, improper coolant supply, job material fault and selection of tool. The logical relation between these seven intermediate events and the top event is represented using OR gate. The blunt tool, feed not proper, improper speed, improper mounting, improper coolant supply, job material fault and selection of tool are the first level events which cause the system to fail. The causes for a failure observed during computer functioning failure shown in Figure 4.1

.

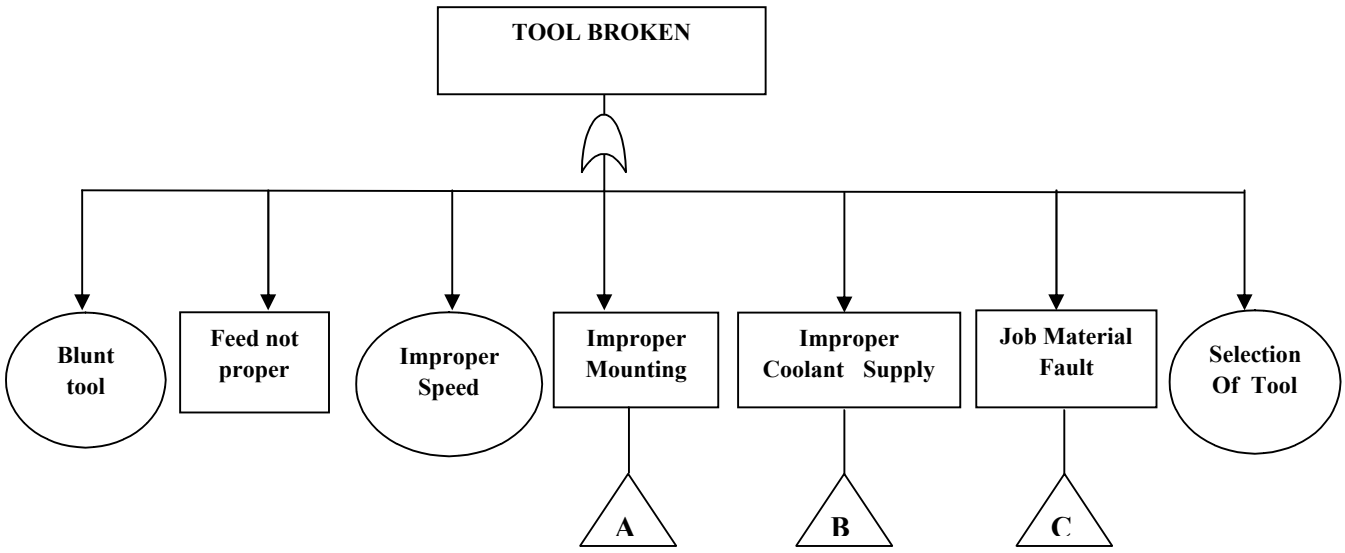


Figure 4.1: Fault tree for tool broken

The improper coolant supply can be attributed to two causes: leakage and dislocation of pipe. These causes are shown in Figure 4.2. The logical relationship between these two intermediate events and improper coolant supply is represented using an OR gate.

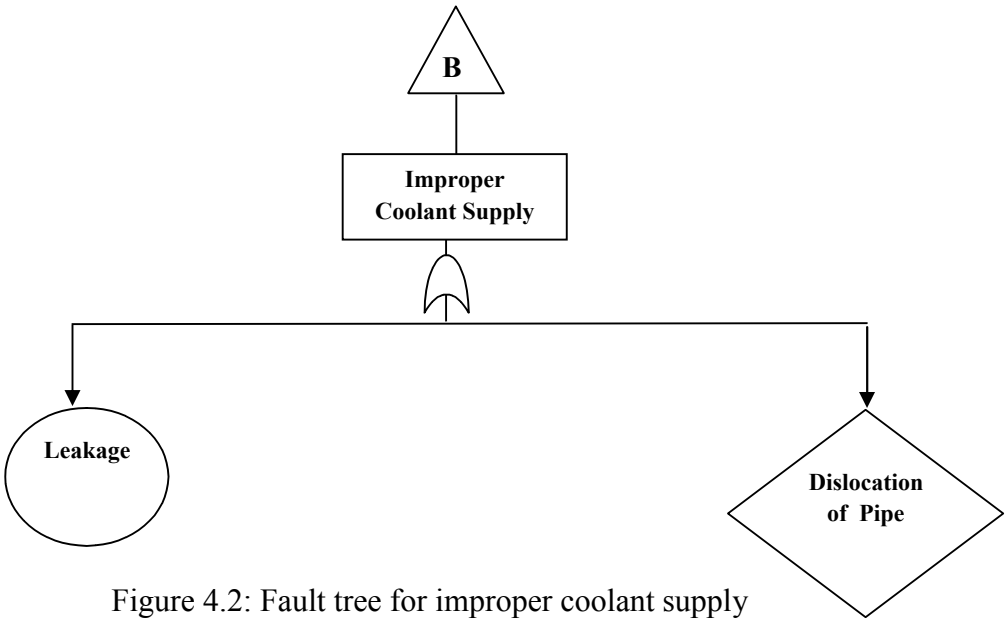


Figure 4.2: Fault tree for improper coolant supply

The improper mounting can be attributed to three causes: centre height, over hanging and tool post fault. These causes are shown in Figure 4.3. The logical relationship between these three intermediate events and improper mounting is represented using an OR gate.

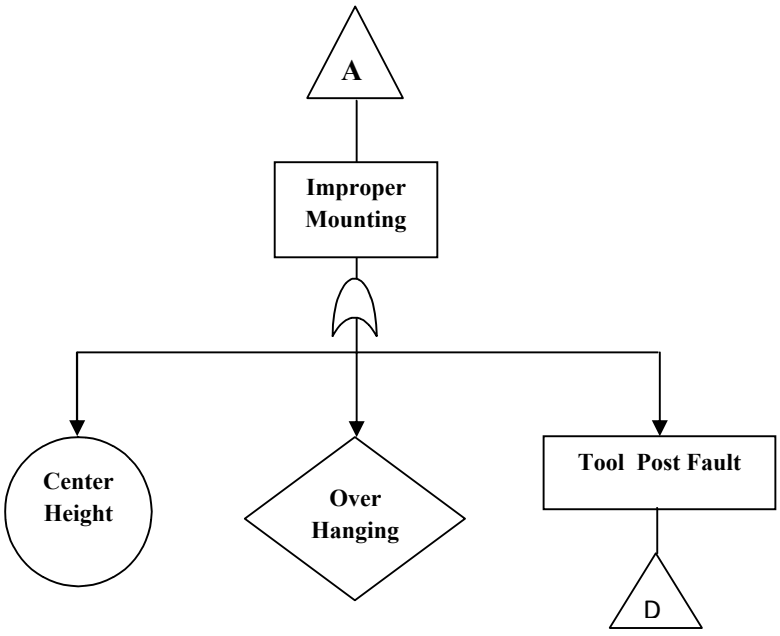


Figure 4.3: Fault tree for improper mounting

The tool post can be attributed to two causes: backlash. These causes are shown in Figure 4.4. The logical relationship between these two intermediate events and tool post is represented using an OR gate.

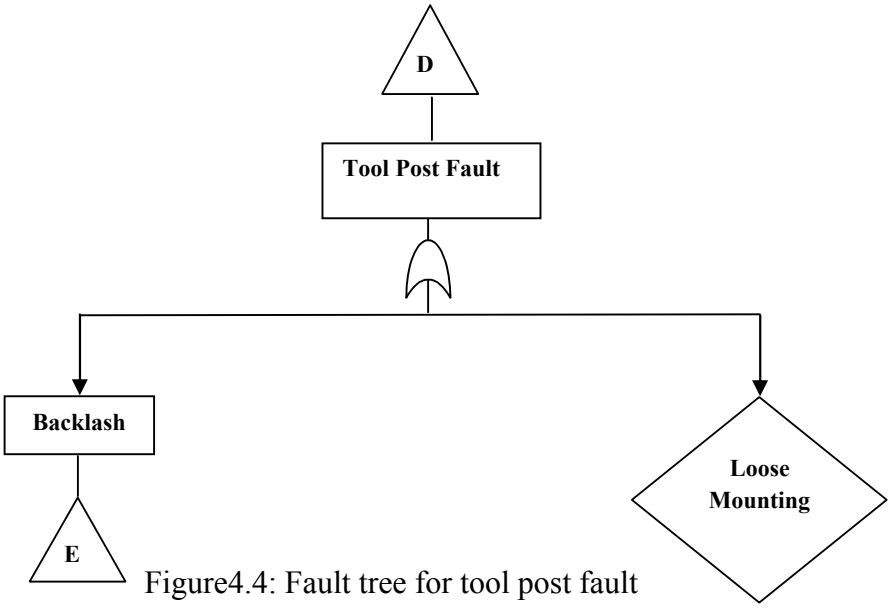


Figure4.4: Fault tree for tool post fault

The backlash can be attributed to two causes: lead screw fault, wear and tear. These causes are shown in Figure 4.5. The logical relationship between these two intermediate events and backlash is represented using an OR gate.

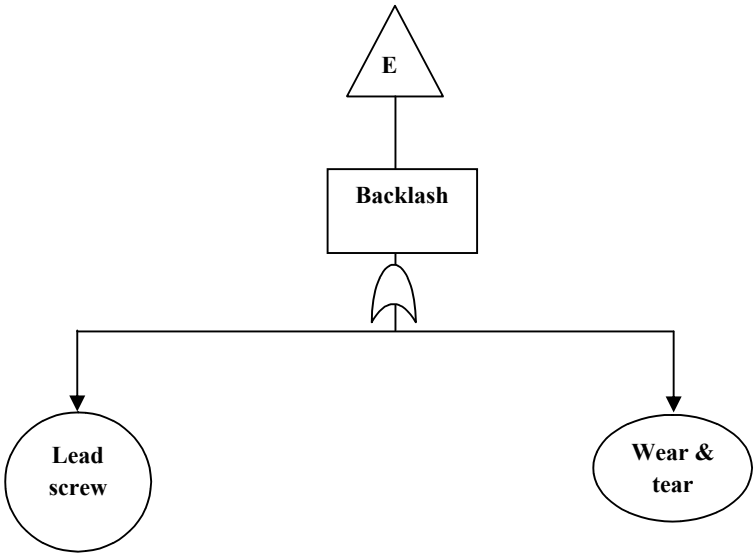


Figure 4.5: Fault Tree for Backlash

The job material fault can be attributed to two causes: material hardness and blow holes. These causes are shown in Figure 4.6. The logical relationship between these two intermediate events and job material fault is represented using an OR gate.

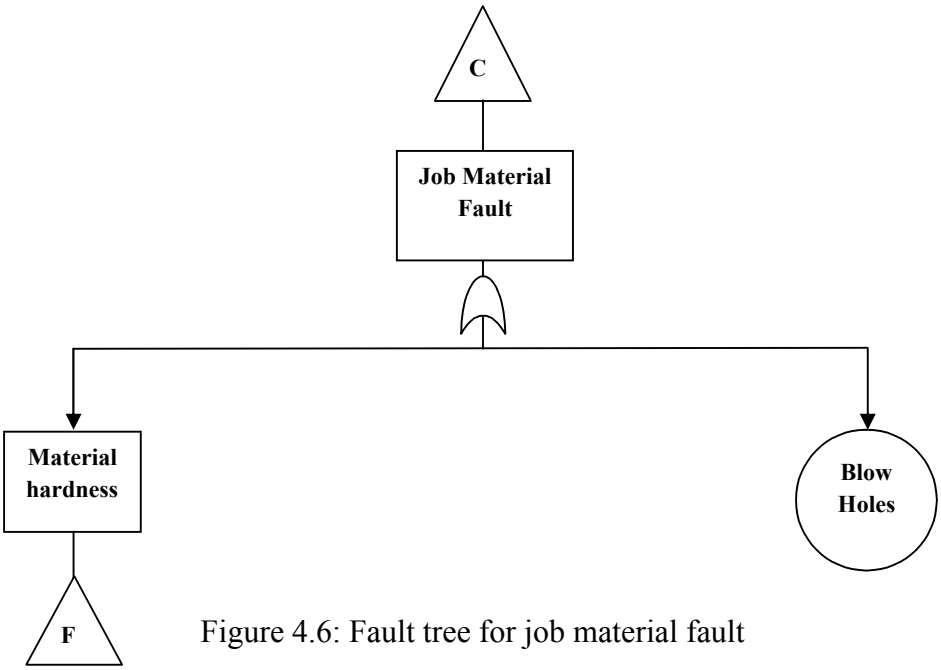


Figure 4.6: Fault tree for job material fault

The material hardness can be attributed to two causes: improper heat treatment and improper alloying. These causes are shown in Figure 4.7. The logical relationship between these two intermediate events and material hardness is represented using an OR gate.

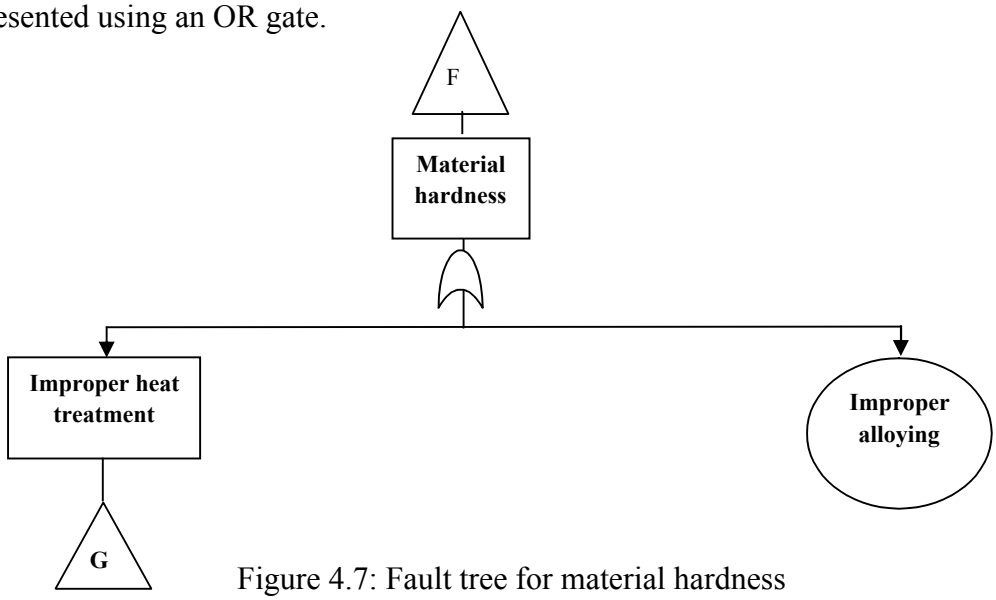


Figure 4.7: Fault tree for material hardness

The improper heat treatment can be attributed to two causes: improper tempering and non uniform hardness. These causes are shown in Figure 4.8. The logical relationship between these two intermediate events and improper coolant supply is represented using an OR gate.

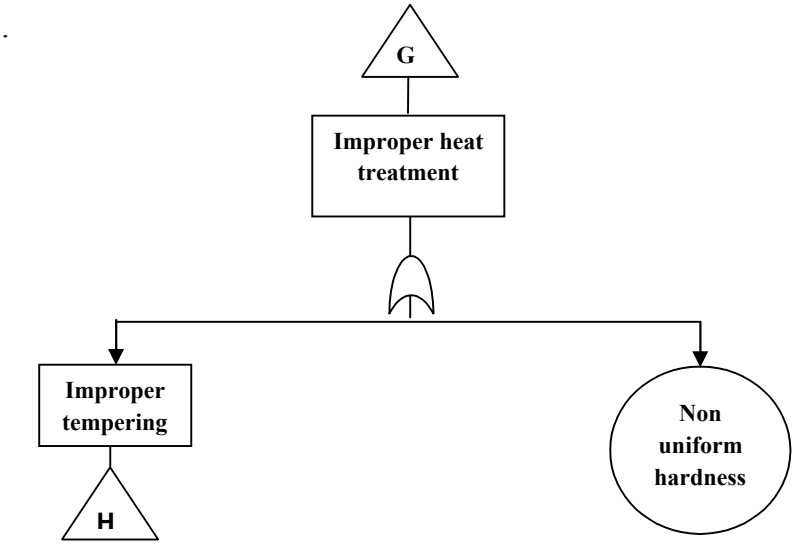


Figure 4.8: Fault tree for improper heat treatment

The improper tempering can be attributed to two causes: improper quenching media and improper coolant timing. These causes are shown in Figure 4.9. The logical relationship between these two intermediate events and tempering is represented using an OR gate.

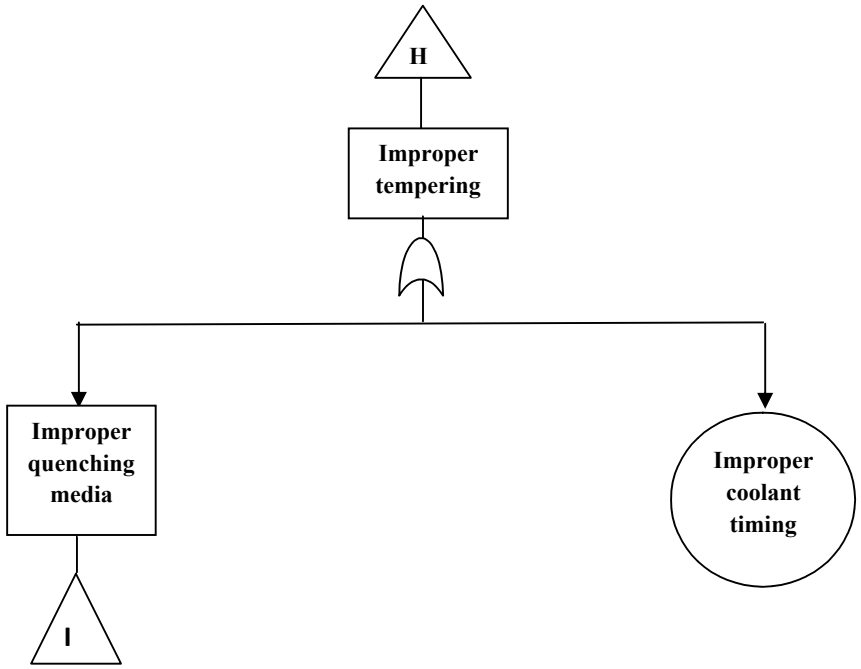


Figure 4.9: Fault tree for improper tempering

The improper quenching media can be attributed to three causes: wrong choice, using beyond shelf life and unclean media. These causes are shown in Figure 4.10. The logical relationship between these three intermediate events and improper

quenching media is represented using an OR gate.

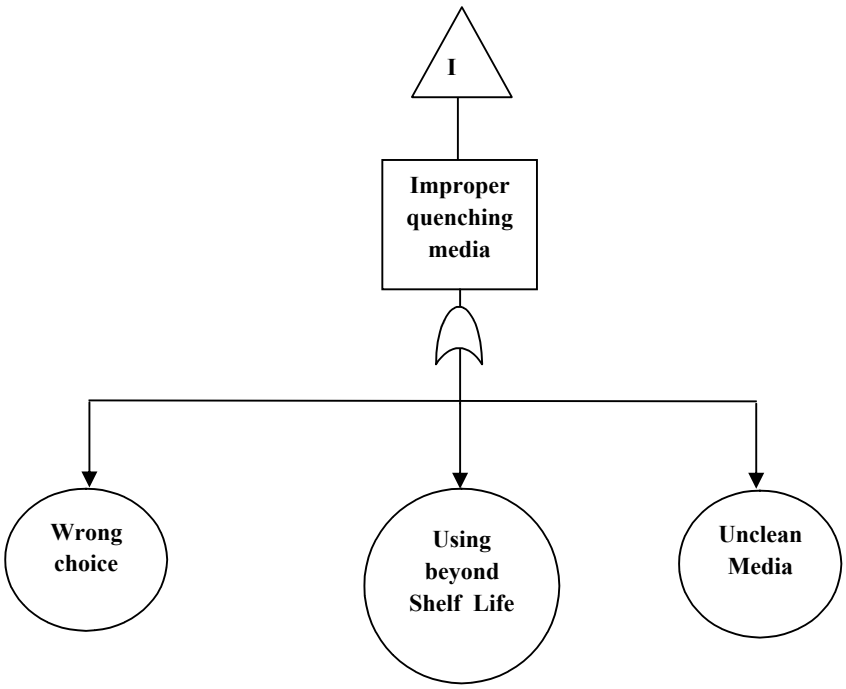


Figure 4.10: Fault tree for improper quenching media

4.4 Qualitative evaluation of the fault tree

In this section qualitative analysis of lathe machine has been presented.

4.4.1 Notations

For the qualitative analysis of lathe machine, the different basic events, due to which lathe machine may fail, are represented by $V_1,\ldots,V_{18}$ which are as follows:

V_1: represents the event that blunt tool.

V_2: represents the event that improper speed.

V_3: represents the event that lead screw fault.

V_4: represents the event that the wear and tear.

V_5: represents the event that center height.

V_6: represents the event that over hanging.

V_7 : represents the event that the loose mounting

V_8 : represents the event that leakage.

V_9 : represents the event that dislocation of pipe.

V_{10} : represents the event that selection of tool.

V_{11} : represents the event that wrong choice.

V_{12} : represents the event that using beyond shelf life.

V_{13} : represents the event that unclean media.

V_{14} : represents the event that improper coolant timing.

V_{15} : represents the event that non uniform hardness.

V_{16} : represents the event that improper alloying.

V_{17} : represents the event that blow holes.

V_{18} : represents the event that feed not proper.

F_1 : represents the event that backlash.

F_2 : represents the event that tool post fault.

F_3 : represents the event that improper mounting.

F_4 : represents the event that improper coolant supply.

F_5 : represents the event that improper quenching media.

F_6 : represents the event that improper tempering.

F_7 : represents the event that improper heat treatment.

F_8 : represents the event that material hardness.

F_9 : represents the event that job material fault.

F : represents the top event tool broken.

4.4.2: Fault tree

The fault tree of tool broken has been shown in fig. 4.11

4.4.3: Qualitative analysis

In this sub section using the sub fault tree's and Boolean algebra, the expressions for evaluating the probability of failure of tool broken has been obtained which are described below:

$F_1 = V_3 + V_4$ (using Figure 4.5)

$F_2 = F_1 + V_7$ (using Figure 4.4)

$F_2 = V_3 + V_4 + V_7$

$F_3 = F_2 + V_5 + V_6$ (using Figure 4.3)

$F_3 = V_3 + V_4 + V_7 + V_5 + V_6$

$F_4 = V_8 + V_9$ (using Figure 4.2)

$F_5 = V_{11} + V_{12} + V_{13}$ (using Figure 4.10)

$F_6 = F_5 + V_{14}$

$F_6 = V_{11} + V_{12} + V_{13} + V_{14}$ (using Figure 4.9)

$F_7 = F_6 + V_{15}$

$F_7 = V_{11} + V_{12} + V_{13} + V_{14} + V_{15}$ (using Figure 4.8)

$F_8 = F_7 + V_{16}$

$F_8 = V_{11} + V_{12} + V_{13} + V_{14} + V_{15} + V_{16}$ (using Figure 4.7)

$F_9 = F_8 + V_{17}$

$F_9 = V_{11} + V_{12} + V_{13} + V_{14} + V_{15} + V_{16} + V_{17}$ (using Figure 4.6)

$F = V_1 + V_2 + V_{10} + F_3 + F_4 + F_9$ (using Figure 4.11)

$$F = V_1 + V_2 + V_3 + V_4 + V_5 + V_6 + V_7 + V_8 + V_9 + V_{10} + V_{11} + V_{12} + V_{13} + V_{14} + V_{15} + V_{16} + V_{17} + V_{18}$$

4.5 Quantitative evaluation of a fault tree

For the quantitative evaluation of fault tree of lathe machine the probability of failures of different components are collected and shown in Table 4.1.

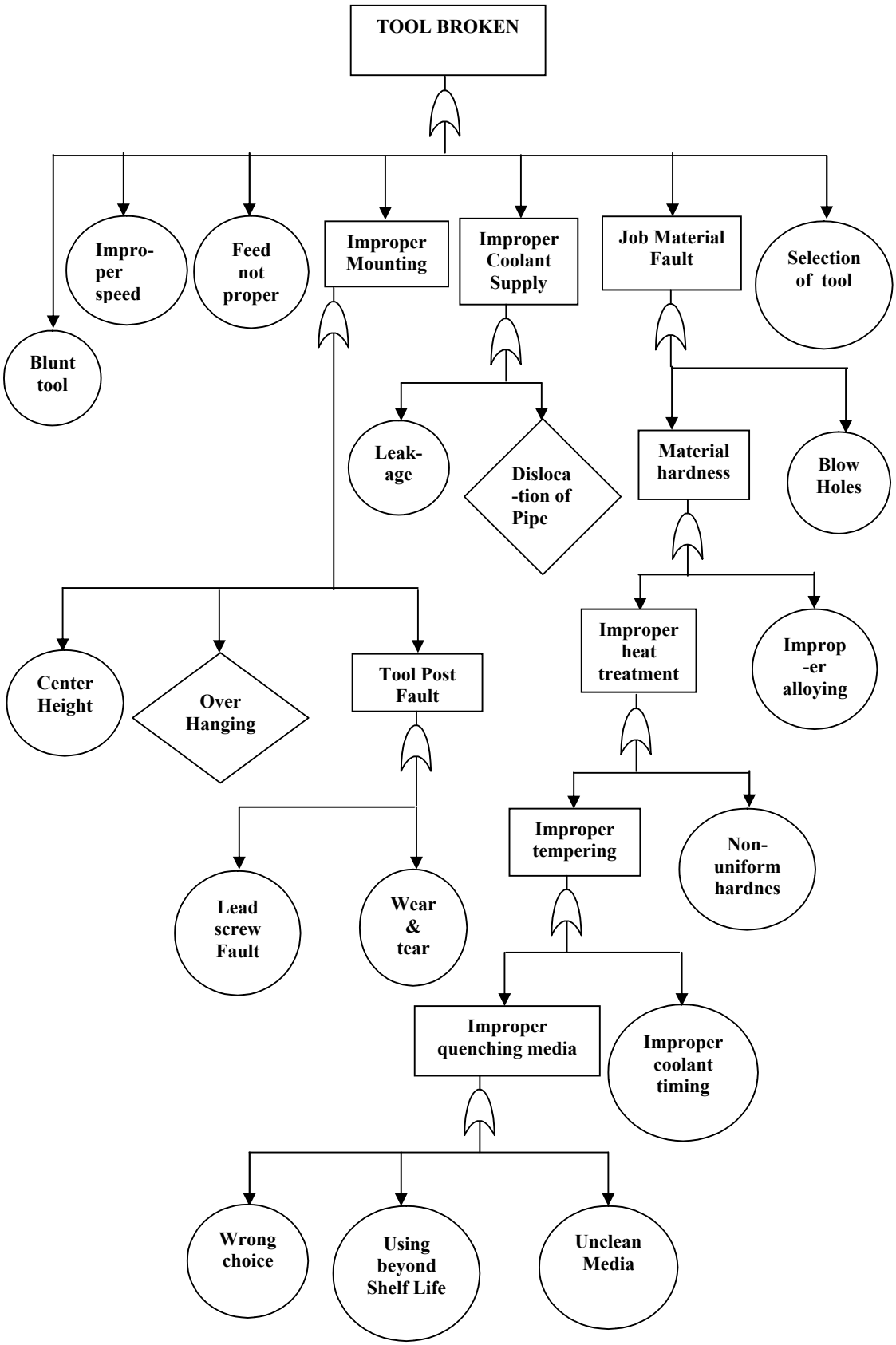


Figure 4.11: Tool broken

Table 4.1 The description of the bottom events of Lathe Machine:

Code	Fault	Probability of failure
V_1	Blunt tool	0.010
V_2	Improper speed	0.015
V_3	Lead screw fault	0.010
V_4	Wear and tear	0.010
V_5	Centre height	0.015
V_6	Over hanging	0.010
V_7	Loose mounting	0.015
V_8	Leakage	0.005
V_9	Dislocation of pipe	0.010
V_{10}	Selection of tool	0.015
V_{11}	Wrong choice	0.010
V_{12}	Using beyond shelf life	0.015
V_{13}	Unclean media	0.010
V_{14}	Improper coolant timing	0.015
V_{15}	No uniform hardness	0.010
V_{16}	Improper alloying	0.010
V_{17}	Blow holes	0.010
V_{18}	Feed not proper	0.010

Putting all the values of $V_1 \ldots V_{18}$ in equation (4.1).

The probability of failure P(F) tool broken is

$$P(F) = 1-(1-V_1)\times(1-V_2)\times(1-V_3)\times(1-V_4)\times(1-V_5)\times(1-V_6)\times(1-V_7)\times(1-V_8)\times(1-V_9)$$
$$\times(1-V_{10})\times(1-V_{11})\times(1-V_{12})\times(1-V_{13})\times(1-V_{14})\times(1-V_{15})\times(1-V_{16})\times(1-V_{17})$$
$$\times(1-V_{18}) \qquad (4.1)$$
$$= 0.18636878$$

4.6 Result and Discussion

The probability of failures of different components in increasing order is shown in Figure 4.12. It will help the user in case of failure of lathe machine i.e. if a lathe machine fails then a user should check first the component with maximum probability of failure. If there is no fault in that component then user should check next component and so on.

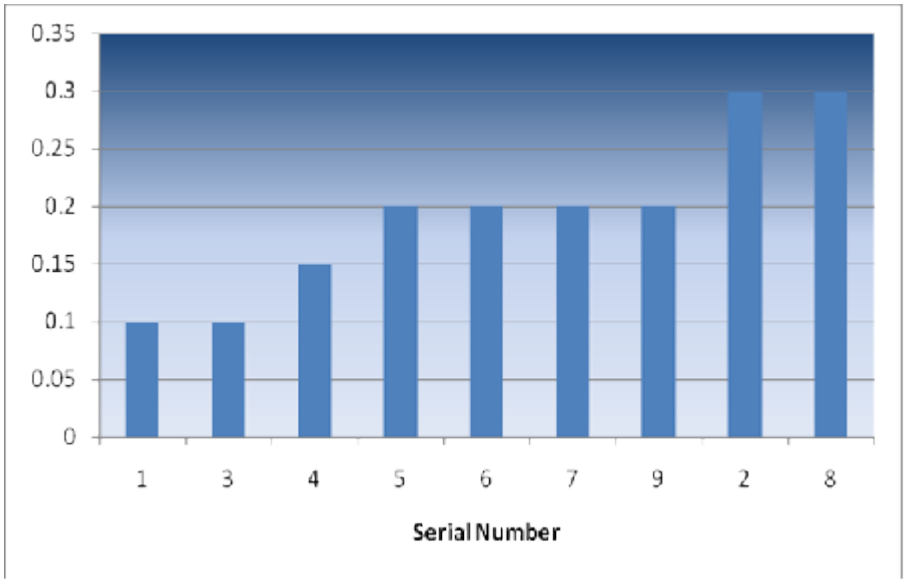


Figure 4.12

4.7 Conclusion

In the first part of this chapter, the basic reasons for the failure of tool broken of lathe machine have been studied and fault tree has been constructed on the basis of this study which results in the qualitative analysis of the lathe machine. Further, data has been collected to perform the quantitative analysis. After collective the data quantitative analysis has been done.

Chapter 5

FAULT TREE ANALYSIS OF PRINTING CIRCUIT BOARD ASSEMBLY

5.1 Introduction

TV's, computers, stereos, phones and many other electronic products that characterize life at the end of the 20th century contain PCBs. During the last two decades, some technological advances have given rise to PCBs of increasing complexity. PCBs that require hundreds or thousands of assembly operations are not uncommon in the manufacture of today's electronic products. Consumers of these products are also demanding higher quality and more reliable items. Under these circumstances, greater control is necessary over the manufacturing processes used to produce PCBs. For example, Surface Mount Technology (SMT) is more widely used because it is faster and more precise. With SMT assembly, the components are placed on a solder adhesive at pre-specified locations on the board. In this chapter, the proposed method is applied for the failure analysis problem of printing circuit board assembly (PCBA) to generate the PCBA fault tree, fault tree nodes, then failure probability of PCBA fault tree. PCBA line for automatic component printing usually compromises four successive work phases depicted as Figure. 5.1.

In this chapter, the basic reasons for the failure of PCBA system have been studied. Fault tree has been constructed on the basis of this study and qualitative and quantitative analysis has been done.

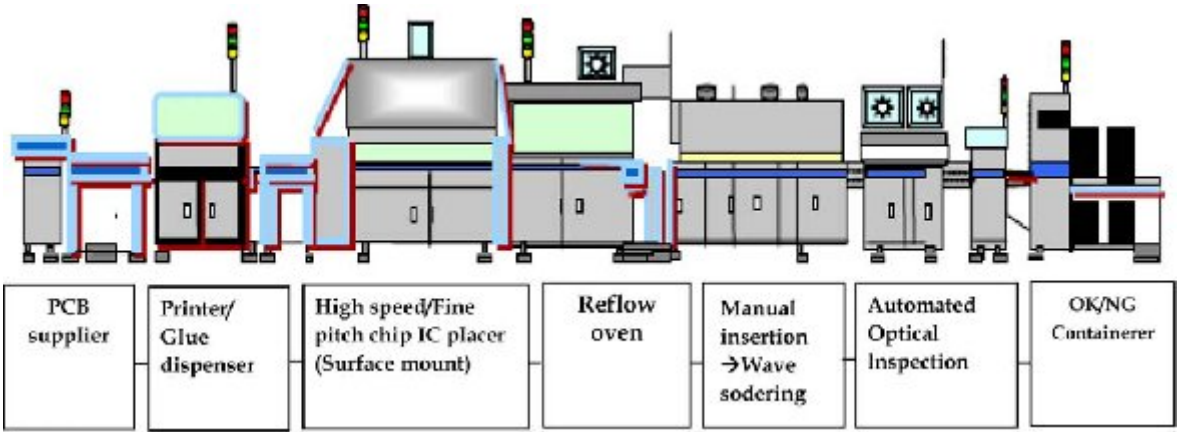


Figure 5.1: PCBA line for automatic component

In the first phase, an initially empty PCB passes a stencil printing and glue dispenser which inserts a glue dot at each onsertion locus or draws adhesive paste over the whole board in order to fixate the electric components. The key is to place the right amount of solder in the right place on the board. Too much paste, for example, could create a solder short that would cause the board to fail. In the second phase, the surface mount device (SMD) machine/onsertion machine is used for the component printing. The machine gets surface mount components from four carriage modules of the feeder unit comprising in total six carriage modules while the other four are used in printing. The set-ups and component printing of the SMD machine consume most of the production time and it is the bottleneck of the whole production line. In the third phase the PCB visits an oven, which heats it in order to harden the glue/paste.

Once a uniform temperature is achieved across the board the temperature is increased again so that reflow (the process of melting the solder paste on the printed circuit board to provide electrical connectivity and hold components) occurs uniformly. In the fourth phase the PCBs wait in the buffer storage and pass a manual insertion phase in which some large components are inserted, then finally the board is passing over

molten solder alloy to make the connections. After these phases, automated optical inspection (AOI) provides in PCB production environment automation to detect defectives and maintain quality. The large components are inserted, then finally the board is passing over molten solder alloy to make the connections. After these phases, automated optical inspection (AOI) provides in PCB production environment automation to detect defectives and maintain quality. The large numbers of operations that are required for producing PCBs make the process very complex and therefore high failure rates are expected.

5.2 Construction of the fault tree

Figures 5.2 to 5.16 presents the fault tree constructed for the problem under study. The top event for this problem is the PCBA fault. The PCBA fault can be broadly classified into five sections: material defective, manufacturing process failure, design mistake, equipment failure and personnel error. The logical relationship between these five intermediate events and the top event is represented using an OR gate. The material defective, manufacturing, design mistake, equipment failure and personnel error are the first level events which cause the system to fail. The cause for a failure observed during PCBA fault shown in Figure 5.2.

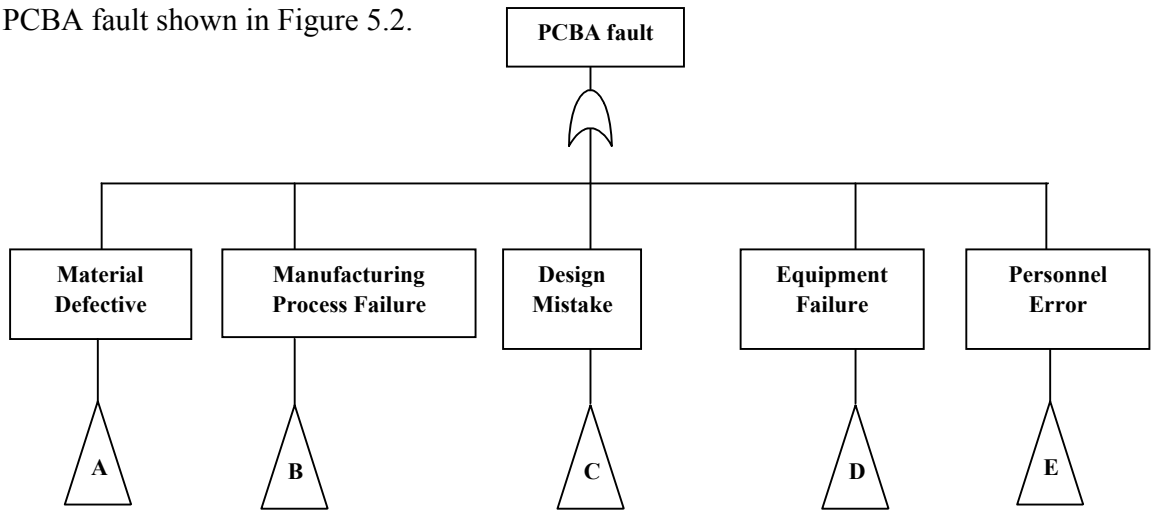


Figure 5.2: Fault tree for PCB fault

The material defective can be attributed to three causes: IC lead bend, improper electrode oxide and improper electroplate. These causes are shown in Figure 5.3. The logical relationship between these three intermediate events and the material defective is represented using an OR gate.

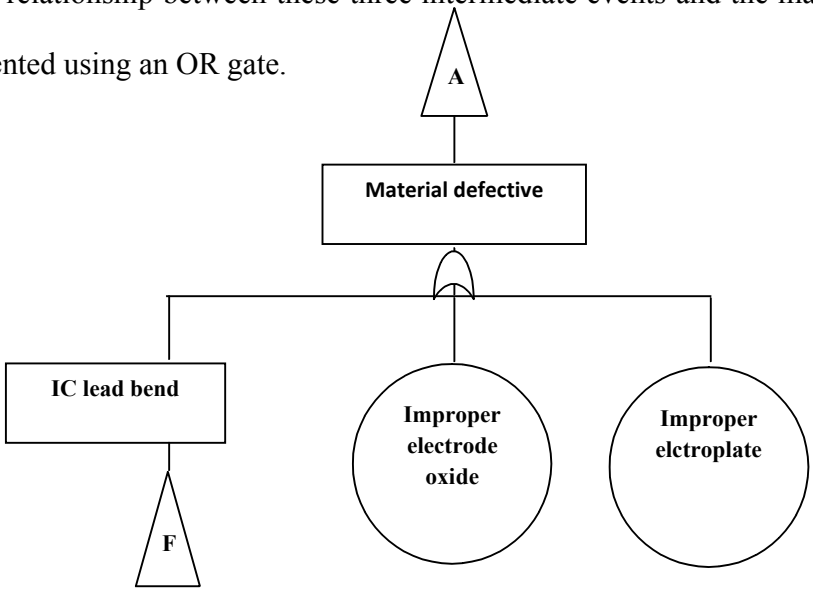


Figure 5.3: Fault tree for material defective

The IC lead bend can be attributed to three causes: insufficient IQC sampling, mixed up failure IC's from vendors and bad handling operation. These causes are shown in Figure 5.4. The logical relationship between these three intermediate events and the IC lead bend is represented using an OR gate.

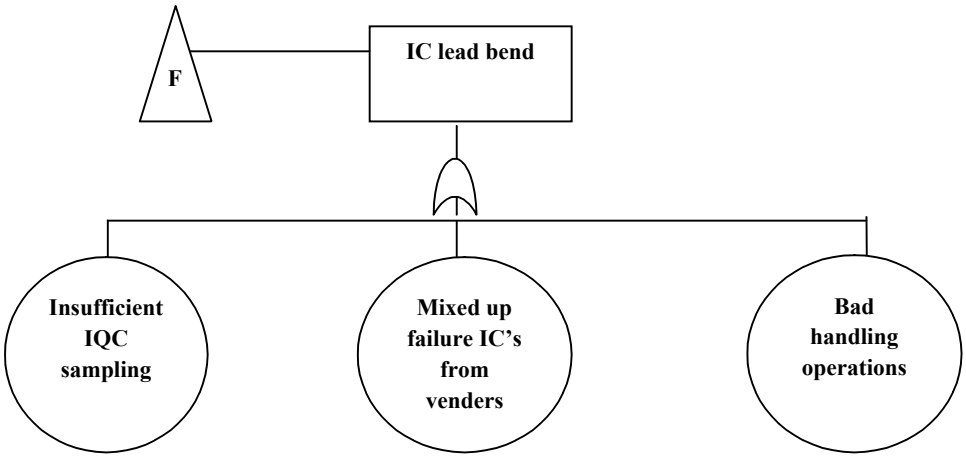


Figure 5.4: Fault tree for IC Lead bend

The manufacturing process failure can be attributed to six causes: excessive temperature profile, electrostatic discharge, improper fixture, inappropriate solder paster, lack of uniformity solder and printing halftone. These causes are shown in Figure 5.5. The logical relationship between these six intermediate events and the manufacturing process failure is represented using an OR gate.

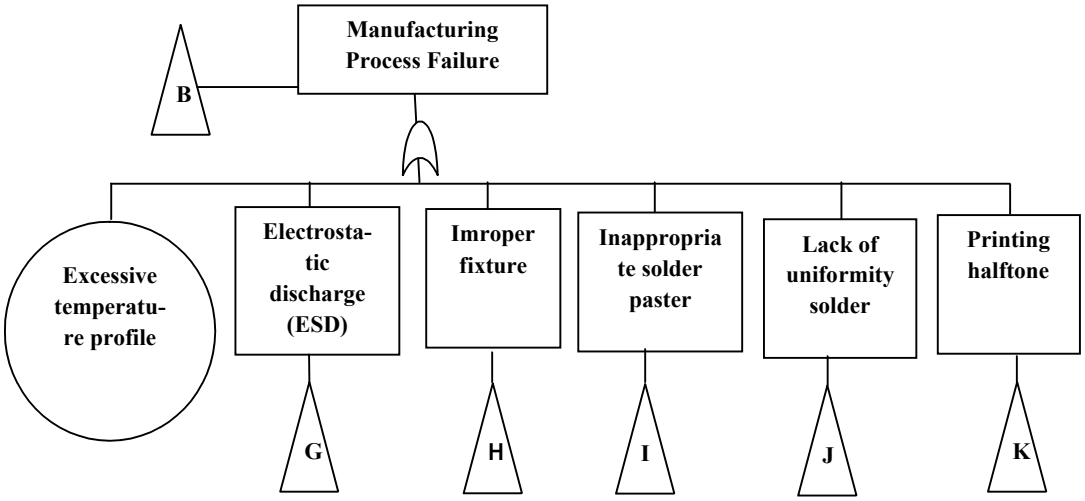


Figure 5.5: fault tree for manufacturing process failure

The electrostatic discharge can be attributed to three causes: insufficient skill, improper ground connection and inferior electrostatic discharge technology. These causes are shown in Figure 5.6. The logical relationship between these three intermediate events and the electrostatic discharge is represented using an OR gate.

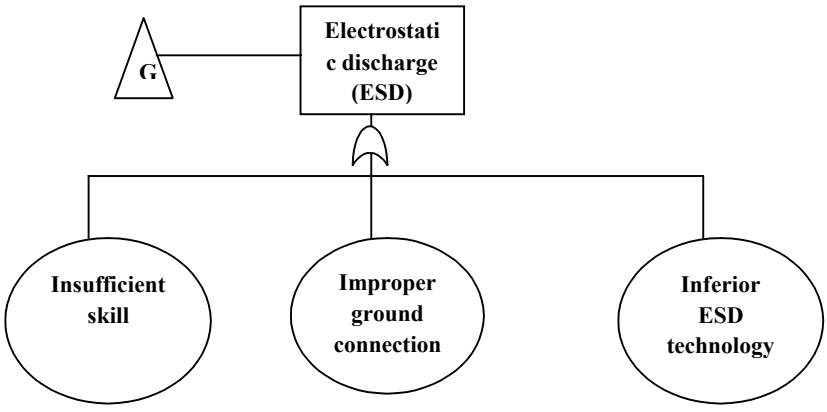


Figure 5.6: Fault tree for electrostatic discharge (ESD)

The improper fixture can be attributed to two causes: lower technology and tolerance mistake. These causes are shown in Figure 5.7. The logical relationship between these two intermediate events and the improper fixture is represented AND gate.

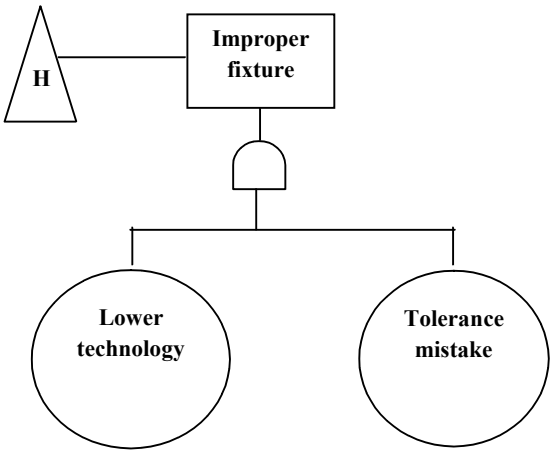


Figure 5.7: Fault tree for improper fixture

The inappropriate solder paster can be attributed to three causes: without building up standard, deficient execution and refrigerator capability problem. These causes are shown in Figure 5.8. The logical relationship between these three intermediate events and the inappropriate solder paster is represented using an OR gate.

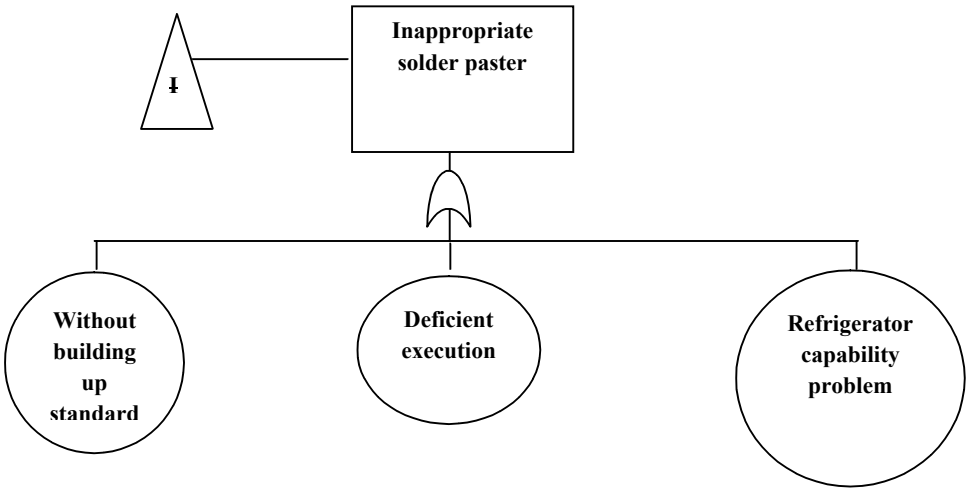


Figure 5.8: Fault tree for inappropriate solder paster

The lack of uniformity solder can be attributed to three causes: squeezing speed problem, squeezing pressure problem and printer machine horizontal problem. These causes are shown in Figure 5.9. The logical relationship between these three intermediate events and the lack of uniformity solder is represented using an OR gate.

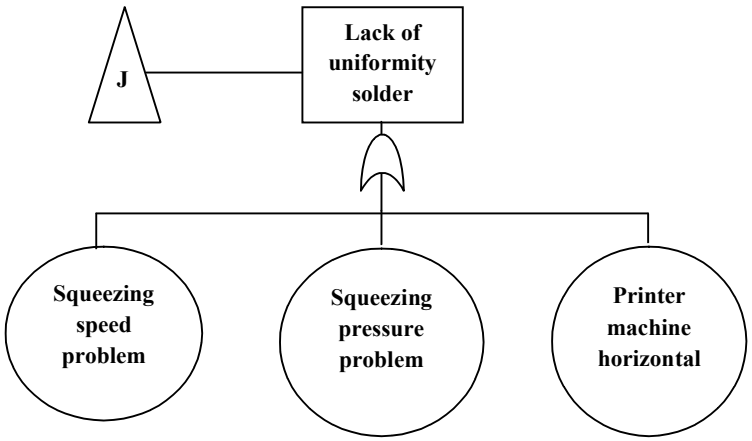


Figure 5.9: Fault tree for lack of uniformity solder

The printing halftone can be attributed to three causes: interior technology, halftone bad management and clean solvent problem. These causes are shown in Figure 5.10. The logical relationship between these three intermediate events and the printing halftone is represented using an OR gate.

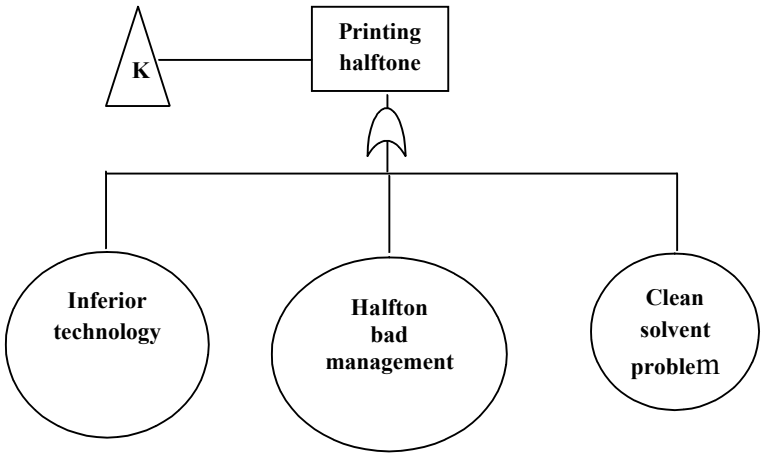


Figure 5.10: fault for printing halftone

The design mistake can be attributed to three causes: PCB pad disequilibrium, component interval problem and matching failure. These causes are shown in Figure 5.11. The logical relationship between these three intermediate events and the design mistake is represented using an OR gate.

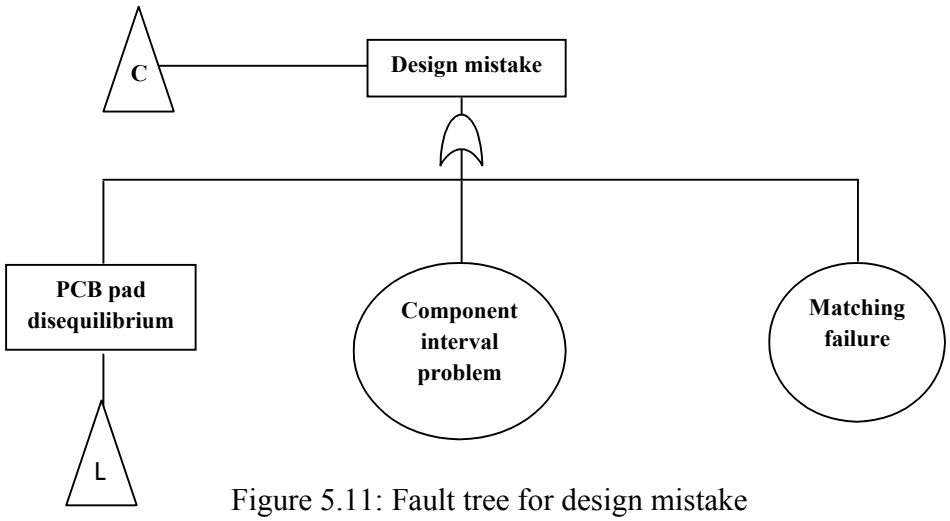


Figure 5.11: Fault tree for design mistake

The PCB pad disequilibrium can be attributed to two causes: lower technology and without building up standard. These causes are shown in Figure 5.12. The logical relationship between these three intermediate events and the PCB pad disequilibrium is represented using an AND gate.

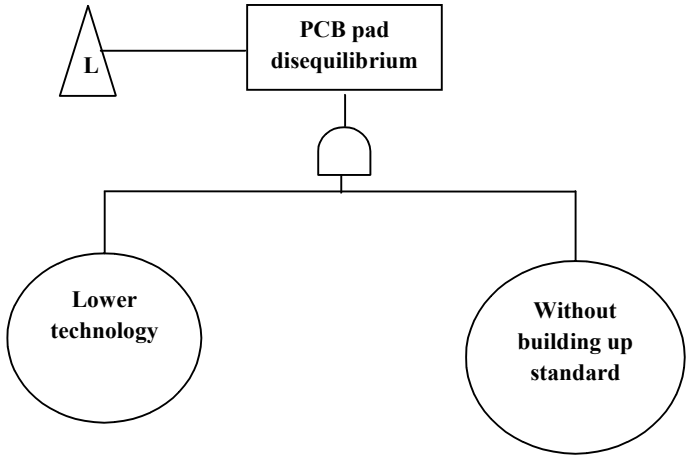


Figure 5.12: tree for PCB pad disequilibrium

The equipment failure can be attributed to four causes: Inaccuracy, optics deviation, oven Δt problem and feeder without pre-maintenance. These causes are shown in Figure 5.13. The logical relationship between these four intermediate events and the equipment failure is represented using an OR gate.

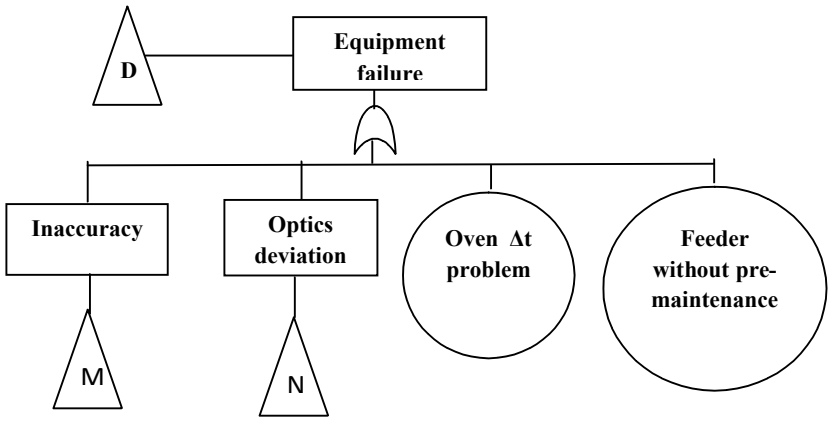


Figure 5.13: Fault tree for equipment

The inaccuracy can be attributed to three causes: short of preventive maintenance, inadequate machine capability and lower technology. These causes are shown in Figure 5.14. The logical relationship between these three intermediate events and the inaccuracy is represented using an OR gate.

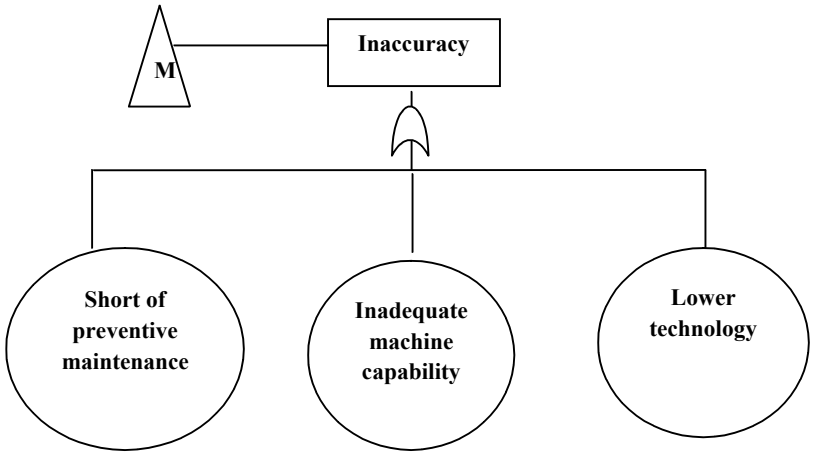


Figure 5.14: Fault tree for inaccuracy

The optics deviation can be attributed to three causes: tolerance of brightness problem, bad preventive and lower technology. These causes are shown in Figure 5.15. The logical relationship between these three intermediate events and the optics deviation is represented using an AND gate.

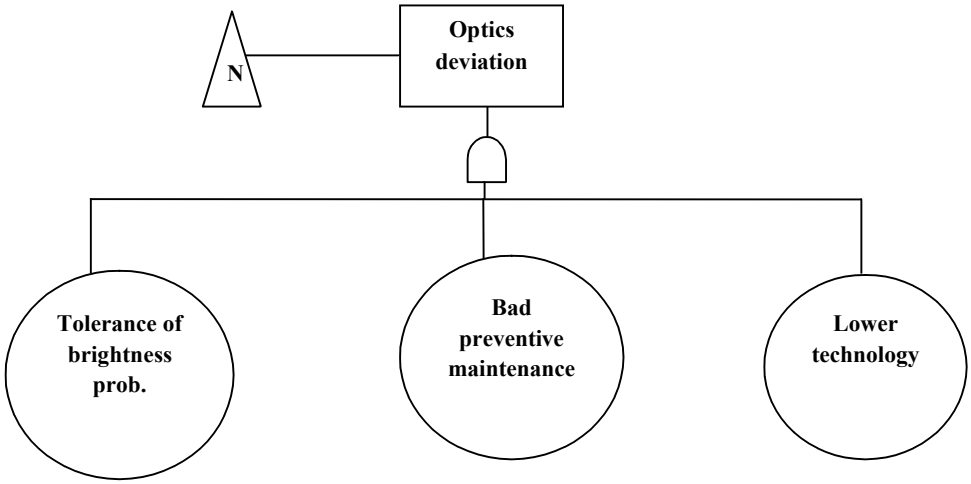


Figure 5.15: Fault tree for optics deviation

The personnel error can be attributed to three causes: insufficient training, incorrect set up equipment and improper handling. These causes are shown in Figure 5.16. The logical relationship between these three intermediate events and the personnel error is represented using an OR gate.

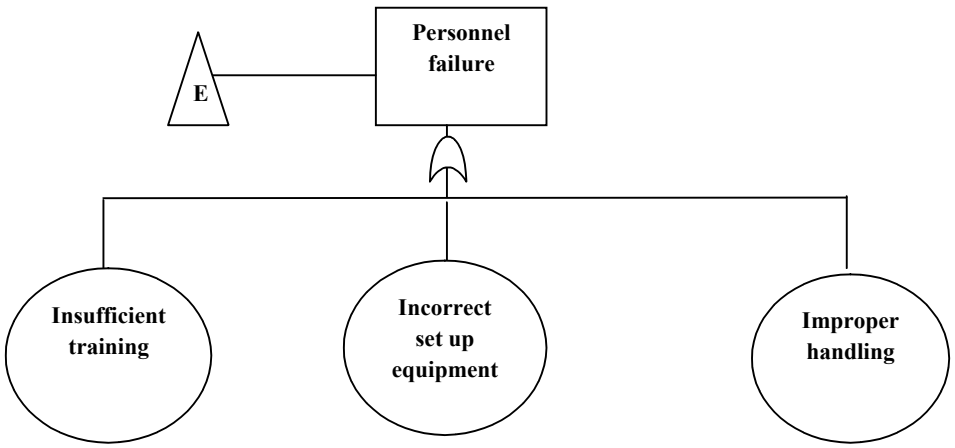


Figure 5.16: fault tree for personnel failure

5.3 Qualitative Evaluation of the fault tree

In this section qualitative analysis of PCBA systems has been presented.

5.3.1 Notations

For the qualitative analysis of PCBA system, the different basic events, due to which PCBA system may fail, are represented by $V_1,\ldots,V_{35}$ which are as follows:

V_1: represents the event that insufficient IQC sampling.

V_2: represents the event that mixed up failure IC's from vendors.

V_3: represents the event that bad handling operation.

V_4: represents the event that improper electrode.

V_5: represents the event that improper electroplate.

V_6: represents the event that excessive temperature profile.

V_7: represents the event that insufficient skill.

V_8: represents the event that improper ground connection.

V_9: represents the event that inferior ESD technology.

V_{10}: represents the event that lower technology.

V_{11}: represents the event that tolerance mistake.

V_{12}: represents the event that without building up standard.

V_{13}: represents the event that deficient execution.

V_{14}: represents the event that refrigerator capability problem.

V_{15} = represents the event that squeezing speed problem.

V_{16} : represents the event that squeezing pressure problem.

V_{17} : represents the event that printer machine horizontal problem.

V_{18} : represents the event that inferior technology.

V_{19} : represents the event that half ton bad management.

V_{20} : represents the event that clean solvent problem.

V_{21} : represents the event that lower technology.

V_{22} : represents the event that without building up standard.

V_{23} : represents the event that component interval problem.

V_{24} : represents the event that matching failure.

V_{25} : represents the event that short of preventive maintenance.

V_{26} : represents the event that inadequate machine capability.

V_{27} : represents the event that lower technology.

V_{28} : represents the event that tolerance of brightness problem.

V_{29} : represents the event that bad preventive maintenance.

V_{30} : represents the event that lower technology.

V_{31} : represents the event that oven t technology.

V_{32} : represents the event that feeder without pre-maintenance.

V_{33} : represents the event that insufficient training.

V_{34} : represents the event that incorrect set up equipment.

V_{35} : represents the event that improper handling.

F_1 : represents the event that IC lead bend.

F_2 : represents the event that material defective.

F_3 : represents the event that electrostatic discharge (ESD).

F_4 : represents the event that improper fixture.

F_5 : represents the event that inappropriate solder paster.

F_6 : represents the event that lack of uniformity solders.

F_7 : represents the event that printing halftone.

F_8 : represents the event that manufacturing process failure.

F_9 : represents the event that PCB pad disequilibrium.

F_{10} : represents the event that design mistake.

F_{11} : represents the event that inaccuracy.

F_{12} : represents the event that optics deviation.

F_{13} : represents the event that equipment failure.

F_{14} : represents the event that personnel error.

F : represents the event that PCBA fault.

5.3.2: Fault tree

The fault tree of PCBA system has been shown in fig. 5.17

5.3.3: Qualitative analysis

In this sub section using the sub fault tree's and Boolean algebra, the expressions for evaluating the probability of failure of PCBA systems has been obtained which are described below:

$$F_1 = V_1 + V_2 + V_3 \quad (\text{using Figure 5.4})$$

$$F_2 = F_1 + V_4 + V_5 \quad (\text{using Figure 5.3})$$
$$= V_1 + V_2 + V_3 + V_4 + V_5$$

$$F_3 = V_7 + V_8 + V_9 \quad (\text{using Figure 5.6})$$

$$F_4 = V_{10} V_{11} \quad (\text{using Figure 5.7})$$

$$F_5 = V_{12} + V_{13} + V_{14} \quad (\text{using Figure 5.8})$$

$$F_6 = V_{15} + V_{16} + V_{17} \quad (\text{using Figure 5.9})$$

$$F_7 = V_{18} + V_{19} + V_{20} \quad (\text{using Figure 5.10})$$

$$F_8 = V_6 + F_3 + F_4 + F_5 + F_6 + F_7 \quad (\text{using Figure 5.5})$$
$$= V_6 + V_7 + V_8 + V_9 + V_{10} V_{11} + V_{12} + V_{13} + V_{14} + V_{15} + V_{16} + V_{17} + V_{18} + V_{19} + V_{20}$$

$$F_9 = V_{21} V_{22} \quad (\text{using Figure 5.12})$$

$$F_{10} = V_{23} + V_{24} + F_9 \quad (\text{using Figure 5.11})$$
$$= V_{23} + V_{24} + V_{21} V_{22}$$

$$F_{11} = V_{25} + V_{26} + V_{27} \quad (\text{using Figure 5.14})$$

$$F_{12} = V_{28} V_{29} V_{30} \quad (\text{using Figure 5.15})$$

$$F_{13} = F_{11} + F_{12} + V_{31} + V_{32} \quad (\text{using Figure 5.13})$$
$$= V_{25} + V_{26} + V_{27} + V_{28} V_{29} V_{30} + V_{31} + V_{32}$$

$$F_{14} = V_{33} + V_{34} + V_{35} \quad (\text{using Figure 5.16})$$

$$F = F_2 + F_8 + F_{10} + F_{13} + F_{14} \quad (\text{using Figure 5.17})$$

$$\begin{aligned} F = {} & V_1 + V_2 + V_3 + V_4 + V_5 + V_6 + V_7 + V_8 + V_9 + V_{10} V_{11} + V_{12} \\ & + V_{13} + V_{14} + V_{15} + V_{16} + V_{17} + V_{18} + V_{19} + V_{20} + V_{23} + V_{24} \\ & + V_{21} V_{22} + V_{25} + V_{26} + V_{27} + V_{28} V_{29} V_{30} + V_{31} + V_{32} + V_{33} \\ & + V_{34} + V_{35} \end{aligned}$$

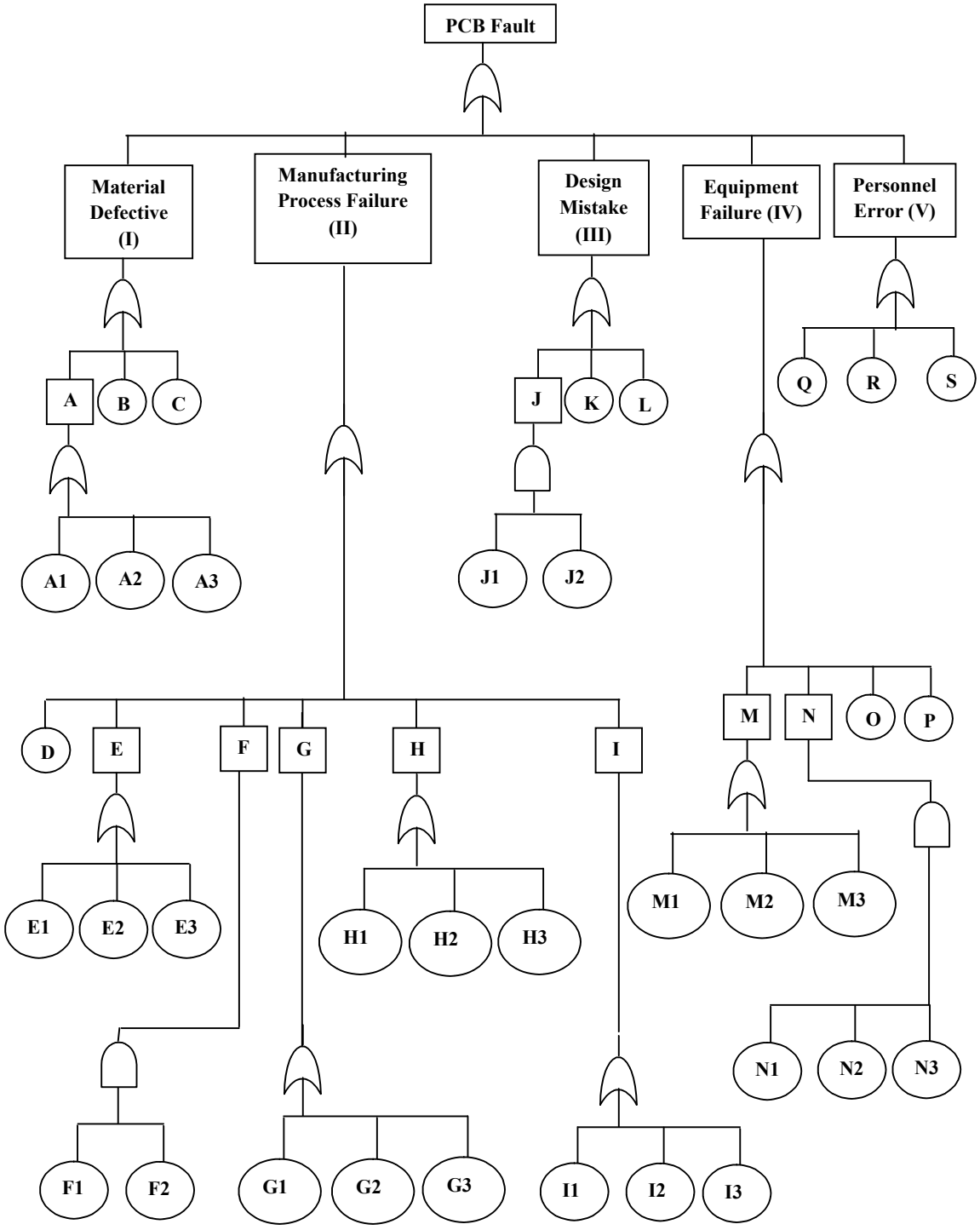


Figure 5.17: Fault tree for PCBA system

Table 5.1 The descriptions of the sub-events of PCBA fault:

S. no.	Code	Fault
1	A	IC lead bend
2	B	Improper electrode oxide
3	C	Improper electroplate
4	D	Excessive temperature profile
5	E	Electrostatic discharge(ESD)
6	F	Improper fixture
7	G	Inappropriate solder paster
8	H	Lack of uniformity solder
9	I	Printing halftone
10	J	PCB pad disequilibrium
11	K	Component interval problem
12	L	Matching failure
13	M	Inaccuracy
14	N	Optics deviation
15	O	Oven $\triangle t$ problem
16	P	Feeder without pre-maintenance
17	Q	Insufficient training
18	R	Incorrect set up equipment
19	S	Improper handling
20	A_1	Insufficient IQC sampling
21	A_2	Mixed up failure ICs from vendors
22	A_3	Bad handling operation
23	E_1	Insufficient skill

24	E_2	Improper ground connection
25	E_3	Inferior ESD technology
26	F_1	Lower technology
27	F_2	Tolerance mistake
28	G_1	Without building up standard
29	G_2	Deficient execution
30	G_3	Refrigerator capability problem
31	H_1	Squeezing speed problem
32	H_2	Squeezing pressure problem
33	H_3	Printer machine horizontal problem
34	I_1	Inferior technology
35	I_2	Half ton bad management
36	I_3	Clean solvent problem
37	J_1	Lower technology
38	J_2	Without building up standard
39	M_1	Short of preventive maintenance
40	M_2	Inadequate machine capability
41	M_3	Lower technology
42	N_1	Tolerances of brightness prob.
43	N_2	Bad preventive maintenance
44	N_3	Lower technology

5.4 Quantitative evaluation of fault tree

For the quantitative evaluation of fault tree of printed circuit board assembly the

probability of failures of different components are collected and shown in Table 5.2.

Table 5.2 The description of the bottom events of PCBA fault:

Code	Fault	Probability of failure
V_1	Insufficient IQC sampling	0.0011
V_2	Mixed up failure IC's from vendors	0.0011
V_3	Bad handling operations	0.00473
V_4	Improper electrode oxide	0.00121
V_5	Improper electroplate	0.00099
V_6	Excessive temperature profile	0.00165
V_7	Insufficient skill	0.0022
V_8	Improper ground connection	0.00077
V_9	Inferior ESD technology	0.00055
V_{10}	Lower technology	0.0011
V_{11}	Tolerance mistake	0.0022
V_{12}	Without building up standard	0.00055
V_{13}	Deficient execution	0.0033
V_{14}	Refrigerator capability problem	0.0011`
V_{15}	Squeezing speed problem	0.00055
V_{16}	Squeezing pressure problem	0.00297
V_{17}	Printer machine horizontal problem	0.0011
V_{18}	Inferior technology	0.0033
V_{19}	Halftone bad management	0.0033

V_{20}	Clean solvent problem	0.00242
V_{21}	Lower technology	0.00462
V_{22}	Without building up standard	0.00264
V_{23}	Component interval problem	0.00077
V_{24}	Matching failure	0.00121
V_{25}	Short of preventive maintenance	0.00583
V_{26}	Inadequate machine capability	0.00022
V_{27}	Lower technology	0.00143
V_{28}	Tolerance of brightness prob.	0.00231
V_{29}	Bad preventive maintenance	0.00132
V_{30}	Lower technology	0.00176
V_{31}	Oven $\triangle t$ problem	0.00121
V_{32}	Feeder without pre-maintenance	0.00286
V_{33}	Insufficient training	0.0044
V_{34}	Incorrect set up equipment	0.00187
V_{35}	Improper handling	0.00418

Putting all the values of $V_1,\ldots,V_{35}$ in equation (5.1).

The probability of failure P(F) of PCBA is

$$\begin{aligned} P(F) = {} & 1-(1-V_1)\times(1-V_2)\times(1-V_3)\times(1-V_4)\times(1-V_5)\times(1-V_6)\times(1-V_7)\times(1-V_8) \\ & \times(1-V_9)\times(1-V_{10}V_{11})\times(1-V_{12})\times(1-V_{13})\times(1-V_{14})\times(1-V_{15})\times(1-V_{16})\times(1-V_{17}) \\ & \times(1-V_{18})\times(1-V_{19})\times(1-V_{20})\times(1-V_{21}V_{22})\times(1-V_{23})\times(1-V_{24})\times(1-V_{25})\times(1-V_{26}) \\ & \times(1-V_{27})\times(1-V_{28}V_{29}V_{30})\times(1-V_{31})\times(1-V_{32})\times(1-V_{33})\times(1-V_{34})(1-V_{35}) \qquad (5.1) \\ = {} & 0.056221154 \end{aligned}$$

5.5 Result and Discussion

The probability of failures of different components in increasing order is shown in Figure 5.13. It will help the user in case of failure of PCBA system i.e. if a PCBA fails then a user should check first the component with maximum probability of failure. If there is no fault in that component then user should check next component and so on.

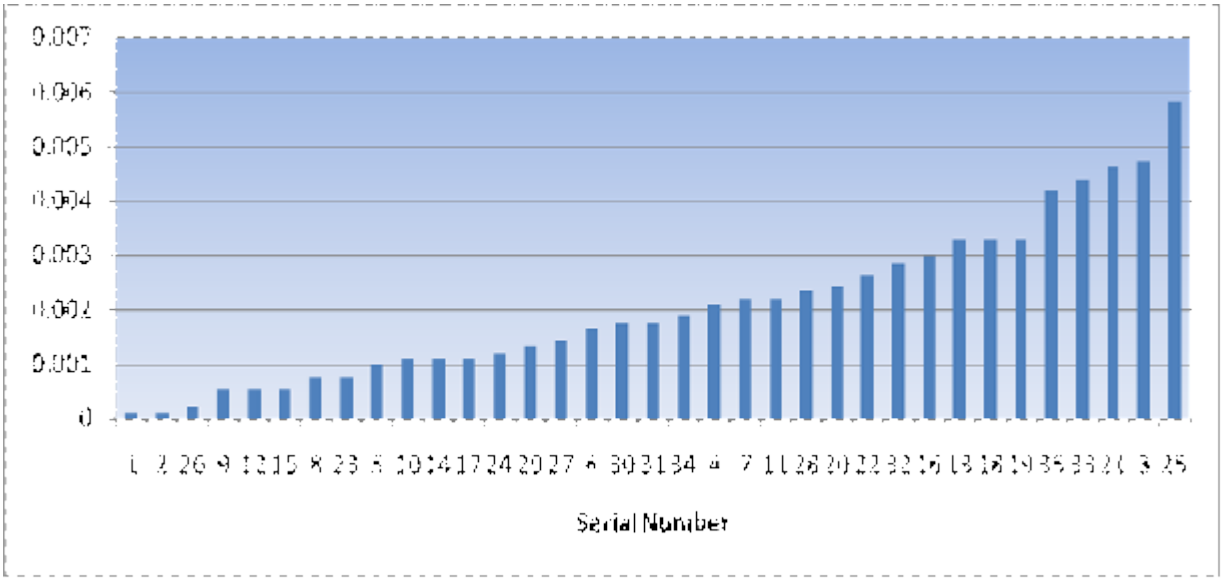


Figure 5.13

5.6 Conclusion

In the first part of this chapter, the basic reasons for the failure of PCBA system have been studied and fault tree has been constructed on the basis of this study which results in the qualitative analysis of the PCBA fault. Further, data has been collected to perform the quantitative analysis. After collective the data quantitative analysis has been done.

REFERENCES

[1]. Arndt, K. and Kirstein, B. M., *FAULT_TREE. A computer code for exact calculation of fault tree characteristics without cut sets*: Proc. 3rd Workshop on Living-PSA-Application, Hamburg 1992.

[2]. Bossche, A., *Computer-Aided Fault Tree Synthesis. System Modelling and Causal Trees*, Reliab. Eng., 1991, Vol. 32, pp. 217-241.

[3]. Brown, K. S., *Evaluating fault trees (AND & OR gates only) with repeated events*, IEEE Trans. Reliability, 1990, pp. 226-235.

[4]. De Vries, R. C., *An Automated Methodology for Generating A Fault Tree*, IEEE Trans. on Reliability, 1990, Vol. 39, pp.76-86.

[5]. Ejlali A. & Miremadi SG, *FPGA-Based Monte Carlo Simulation For Fault Tree Analysis,* Microelectronics Reliability, 2004.

[6]. Feutz, R. J. and Waldeck, T. A., *The Application of Fault Tree Analysis to Dynamic Systems*, Boeing/UW System Safety Symposium, 1965.

[7]. Fussell, J. B., *A Formal Methodology for Fault Tree Construction*, Nuclear Science and Engineering, Vol. 52, 1973, pp. 421-432.

[8]. Fussell, J. B., *Synthetic Tree Model- A Formal Methodology for Fault Tree Construction*, ANCR-1098, 1973.

[9]. Fussell, J. B., Henry, E. B. and Marshall, N. H., *MOCUS – A Computer to Obtain Minimal Sets from Fault Trees*, Aerojet Nuclear Corp, ANCR-1156, 1974.

[10]. Fussell, J. B. and Vesely, W. E., *A New Methodolgy for Obtaining Cut Sets for Fault Trees*, Transactions American Nuclear Society, Vol. 15, 1972, pp. 262-263.

[11].Haasl, D. F., *Advanced Concepts in Fault Tree Analysis*, Boeing/UW System Safety Symposium, 1965.

[12].Hennings, W. and. Kuznetsov, N. Yu, *FAMOCUTN and CUTQN — computer codes for fast analytical evaluation of large fault trees with replicated and negated gates*, IEEE Trans. Reliability, 1995, pp. 368–376.

[13].Hu W, Starr AG & Leung AYT, *Operational Fault Diagnosis of Manufacturing Systems*, J Mater Process Technol, 2003.

[14].Kales P., *Reliability for Technology, Engineering and Management*, Prentice-Hall, 1998.

[15].Kanda, K., *Concept of System Safety Mathematics*, Boeing/UW System Safety Symposium, 1965.

[16].Kelly, B. E. and Lee, F. P., *The Propagation of Faults in Process Plants. Modeling of Fault Propagation*, Reliability Engineering, vol. 16, 1986, pp. 3-38.

[17].Khan FI & Abbasi SA. *Analytical Simulation And PROFAT II: A New Methodology And a Computer Automated Tool For Fault Tree Analysis In Chemical Process Industries*, J Hazard Mater, 2000.

[18].Kohda, T., Henley, E.J. and Inoue, K., *Finding modules in fault trees*, IEEE Trans. Reliability, 1989, pp.165–176.

[19].Kovalenko, I. N., Kuznetsov, N. Yu. and Pegg, Ph. A., *Mathematical Theory of Reliability of Time Dependent Systems with Practical Applications*, Wiley, Chichester 1997.Kuznetsov, N. Yu., *Fault trees — problems and the modern state of investigation*, Kibern. Sist. Analiz, 1994, pp.128–150.

[20].Lee, W. S., Grosh, D. L., Tillman, F. A and Lie, C. H. , *Fault tree analysis, methods and applications — a review*, IEEE Trans. Reliability, 1985, pp.194-203.

[21].Levine, S. and Vesely, W. E., *Important Event Tree And Fault Tree Consideration In The Reactor Safety Study*, IEEE Transactions On Reliability, 1976.

[22].Li H-X, Zuo MJ, *A Hybrid approach For Identification Of Root Causes And Reliability Improvement Of a Die Bonding Process*- A Case Study Reliability Engineering System safety, 1999.

[23].Lyu, M. R., editor. *Handbook of Software Reliability Engineering*. McGraw-Hill, 1995.

[24].Mearns, A. B., *Fault Tree Analysis: The Study of Unlikely Events in Complex Systems*, Boeing/UW System Safety Symposium, 1965.

[25].Michels, J. M., *Computer Evaluation of the Safety Fault Tree Model*, Boeing/UW System Safety Symposium, 1965.

[26].Page, L. B. and Perry, J. E., An *algorithm for exact fault tree probabilities without cut sets*, IEEE Trans. Reliability, 1986, pp**.** 544–558.

[27].Schneeweiss W. G., *The Fault Tree Method (in the Fields of Reliability and Safety Technology)*, Lilole-Verlag, Berlin 2000.

[28].Stecher, K., *Evaluation of large fault trees with repeated events using an efficient bottom-up algorithm*, IEEE Trans. Reliability, 1986, pp. 51–58.

[29].Vemuri, K. K., Dugan, J. B., *Automatic Synthesis of Fault Trees for Computer-Based Systems*, IEEE Transactions on Reliability, 1999, Vol. 48, No. 4, pp. 394-402.

[30].Vesely, W. E., Goldberg, F. F. , Roberts, N. H. and Haasl D. F., *Fault tree handbook. Technical Report NUREG-0492*, U.S. Nuclear Regulatory Commission, Washington, D.C, Jan 1981.

[31].Wang, J. D., Liu, T. S., *A Component Behavioral Model for Automatic Fault Tree Construction*, Reliability Engineering and System Safety, 1993, Vol. 42, pp. 87-100.

[32].Wang, Y., Teague, T., West, H., Mannan, S., *A New Algorithm for Computer-Aided Fault Tree Synthesis*, Journal of Prevention in the Process Industries, 2002, Vol. 15, pp. 265-277.

[33].Watson, H. A., *Launch Control Safety Study*, Bell Labs; Murray Hill, NJ, Section VII Vol. 1, 1961.

[34].Zemva A. & Zajc B., *Test Generation for Technology-Specific Multi-Faults Based On Detectable Perturbations*, Microelectronics Reliability, 2005.