Fault Tree Analysis

Clifton A. Ericson II

cericson@aot.com

cliftonericson@cs.com

Fault Tree Analysis

Clifton A. Ericson II

Sept. 2000

cliftonericson@cs.com or FaultTree1@cs.com

Outline

- Overview
- History
- Basic Process
- Definitions
- Construction
- Mathematics
- Evaluation
- Pitfalls
- Rules
- Examples

FTA Overview

Introduction

"To design systems that work correctly we often need to understand and correct how they can go wrong."

Dan Goldin, NASA Administrator, 2000

FTA identifies, models and evaluates the unique interrelationship of events leading to :

- **Failure**
- **Undesired Events / States**
- **Unintended Events / States**

FTA - Description

- Tool
  - evaluate complex systems
  - identify events that can cause an Undesired Event
  - safety, reliability, unavailability, accident investigation

- Analysis
  - identifies root causes
  - deductive (general to the specific)
  - provides risk assessment
    - ☞ cut sets (qualitative)
    - ☞ probability (quantitative)

FTA - Description

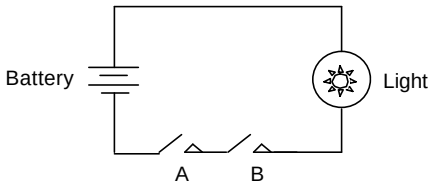
- Model
  - visual
  - displays cause-consequence relationships
  - fault events, normal events, paths
  - probability

A picture is worth a 1,000 words!

- Methodology
  - defined, structured and rigorous
  - easy to learn, perform and follow
  - utilizes Boolean Algebra, probability theory, reliability theory, logic
  - follows the laws of physics, chemistry and engineering

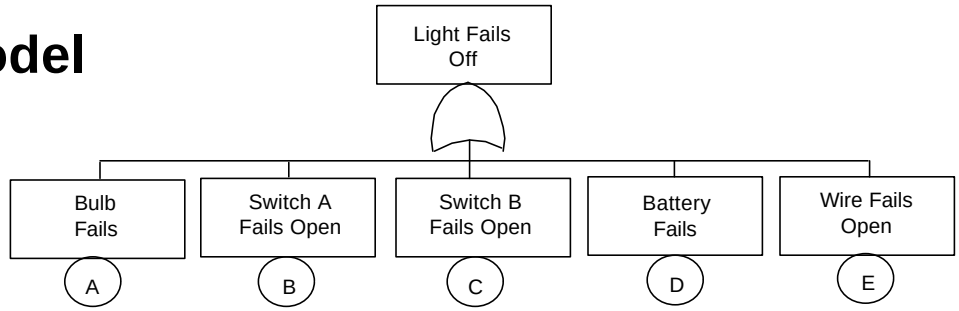
Example FT

System



System Undesired Event: *Light Fails Off*

FT Model



Cut Sets

Event combinations that can cause Top Undesired Event to occur

CS	Probability
A	$P_A=1.0x10^{-6}$
B	$P_B=1.0x10^{-7}$
C	$P_C=1.0x10^{-7}$
D	$P_D=1.0x10^{-6}$
E	$P_E=1.0x10^{-9}$

FTA Application – Why

- Root Cause Analysis
  - Identify all relevant events and conditions leading to Undesired Event
  - Determine parallel and sequential event combinations
  - Model diverse/complex event interrelationships involved
- Risk Assessment
  - Calculate the probability of an Undesired Event (level of risk)
  - Identify safety critical components/functions/phases
  - Measure effect of design changes
- Design Safety Assessment
  - Demonstrate compliance with requirements
  - Shows where safety requirements are needed
  - Identify and evaluate potential design defects/weak links
  - Determine Common Mode failures

FTA -- Coverage

- Failures
- Fault Events
- Normal Events
- Environmental Effects
- Systems, subsystems, and components
- System Elements
  - hardware, software, human, instructions
- Time
  - mission time, single phase, multi phase
- Repair

FT Strengths

- Visual model -- cause/effect relationships
- Easy to learn, do and follow
- Models complex system relationships in an understandable manner
  - Follows paths across system boundaries
  - Combines hardware, software, environment and human interaction
- Probability model
- Scientifically sound
  - Boolean Algebra, Logic, Probability, Reliability
  - Physics, Chemistry and Engineering
- Commercial software is available
- FT's can provide value despite incomplete information
- Proven Technique

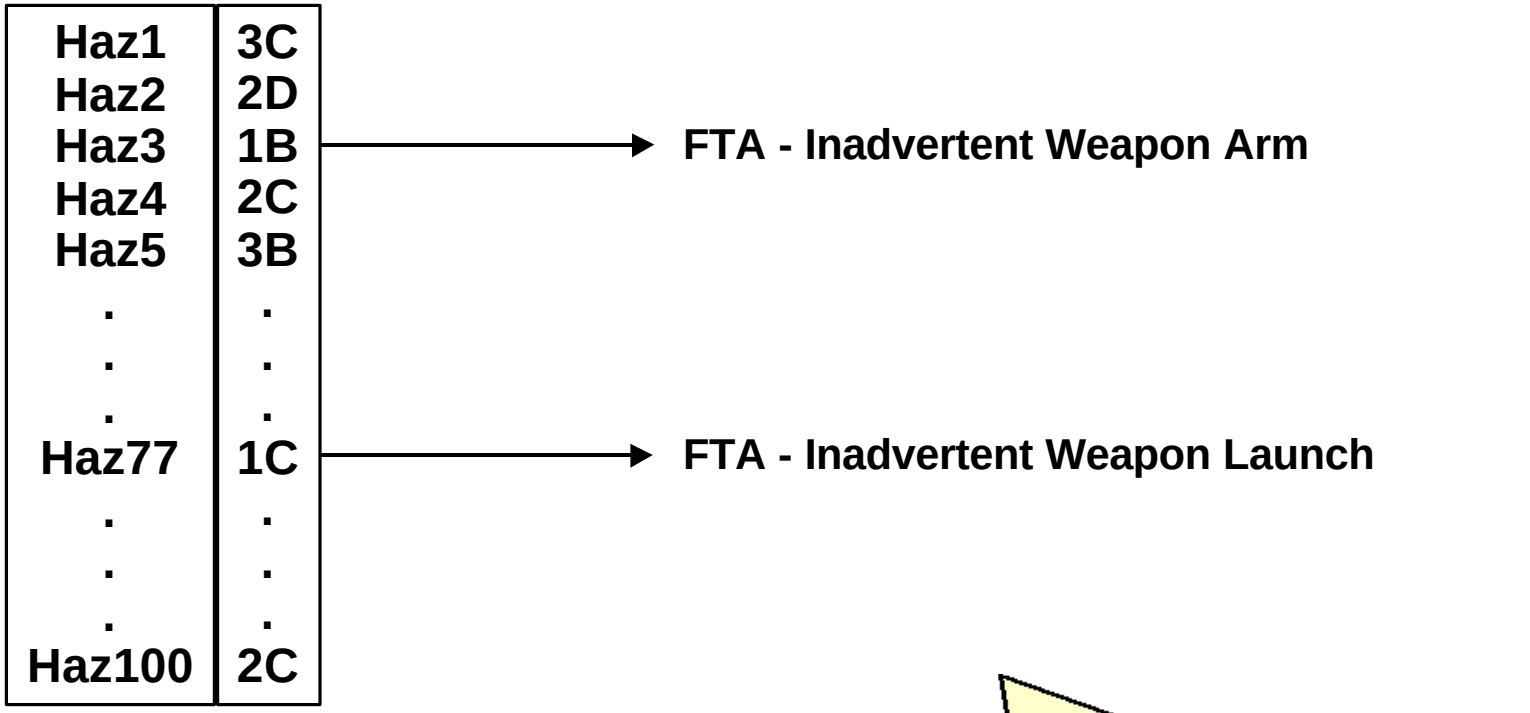
FTA Misconceptions

- Not a Hazard Analysis
  - root cause analysis vs. hazard analysis
  - deductive vs. inductive
- Not an FMEA
  - FMEA is bottom up single thread analysis
- Not an Un-Reliability Analysis
  - System Integrity vs. Availability
  - not an inverse Success Tree
- Not a model of all system failures
  - only includes those failures pertinent to the top Undesired Event
- Not 100% fidelity – model of reality only
  - estimate, not an exact duplicate
  - perception of reality

FTA Application -- When

- Required by customer
- Required for certification
- Necessitated by the risk involved with the product (risk is high)
- Accident/incident/anomaly investigation
- To make a detailed safety case for safety critical system
- To evaluate corrective action or design options
- Need to evaluate criticality, importance, probability and risk
- Need to know root cause chain of events
- To evaluate the effect of safety barriers
- Determine best location for safety devices (weak links)

FTA Is Not For Every Hazard



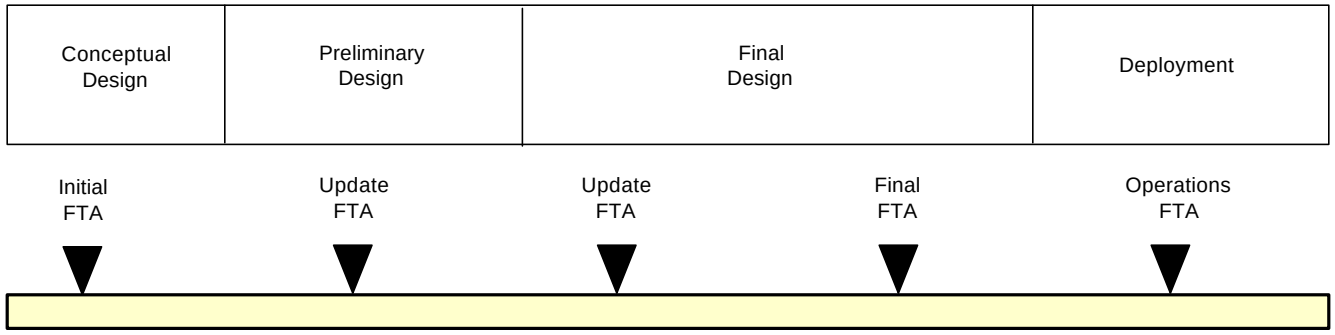
Only do FTA on Safety Critical hazards.

Example Applications

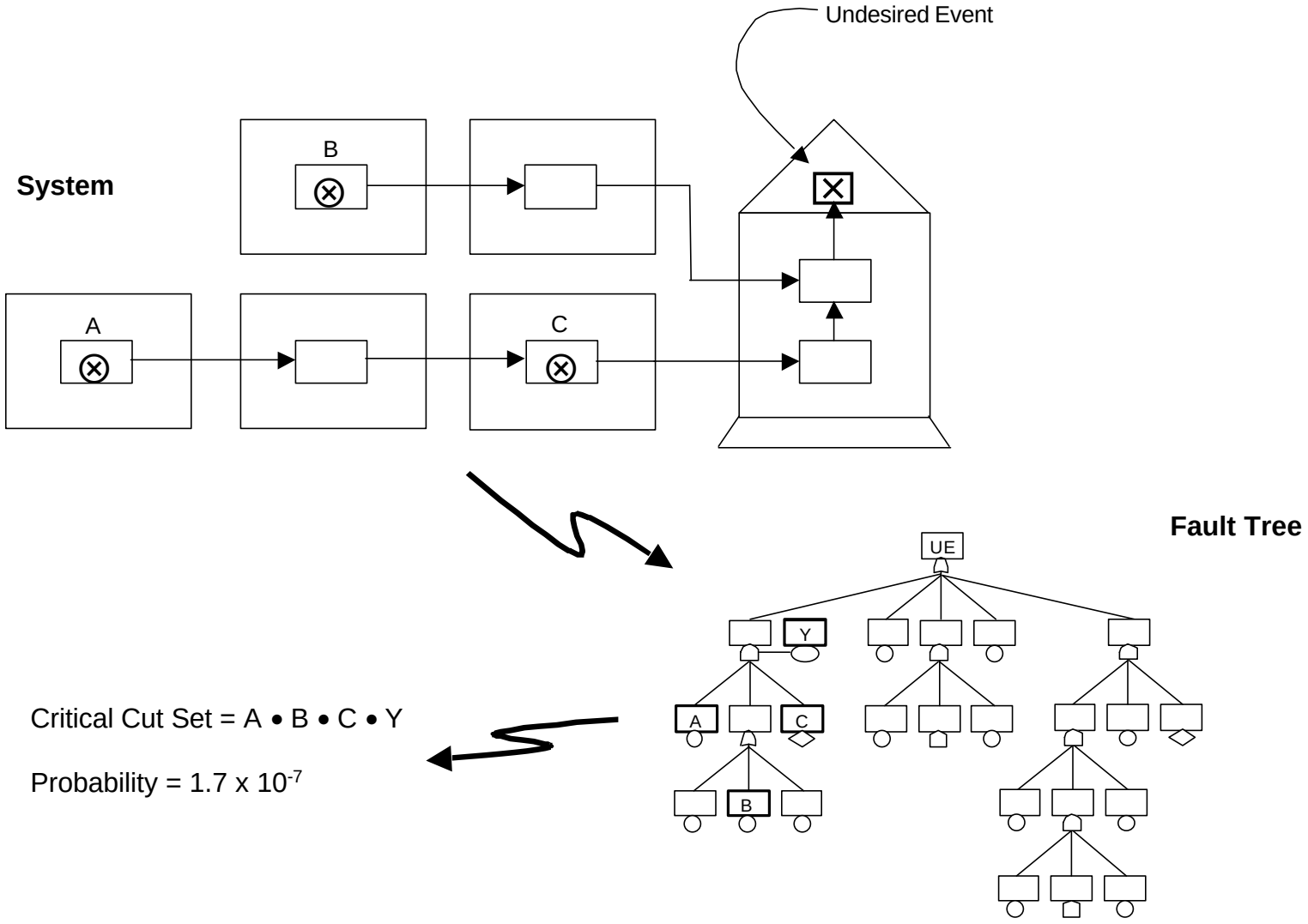
- Evaluate inadvertent arming and release of a weapon
- Calculate the probability of a nuclear power plant accident
- Evaluate an industrial robot going astray
- Calculate the probability of a nuclear power plant safety device being unavailable when needed
- Evaluate inadvertent deployment of jet engine thrust reverser
- Evaluate the accidental operation and crash of a railroad car
- Evaluate spacecraft failure
- Calculate the probability of a torpedo striking target vessel
- Evaluate a chemical process and determine where to monitor the process and establish safety controls

FTA Timeline

- Design Phase
  - FTA should start early in the program
  - The goal is to influence design early, before changes are too costly
  - Update the analysis as the design progresses
  - Each FT update adds more detail to match design detail
  - Even an early, high level FT provides useful information
- Operations Phase
  - FTA during operations for root cause analysis
  - Find and solve problems (anomalies) in real time



FTA – Summary



Critical Cut Set = A • B • C • Y

Probability = 1.7×10^{-7}

FTA – Summary

- FTA is an **analysis tool**
  - Strengths – methodical, structured, graphical, quantitative, easy to model complex systems
  - Coverage – hardware, software, humans, procedures, timing
  - Like any tool, the user must know when, why and how to use it correctly
- FTA is for **system evaluation**
  - Safety – hazardous and catastrophic events
  - Reliability – system unavailability
  - Performance – unintended functions
- FTA is for **decision making**
  - Root cause analysis
  - Risk assessment
  - Design assessment

FTA History

FTA Historical Stages

The Beginning Years (1961 – 1970)

- H. Watson of Bell Labs, along with A. Mearns, developed the technique for the Air Force for evaluation of the Minuteman Launch Control System, circa 1961
- Recognized by Dave Haasl of Boeing as a significant system safety analysis tool (1963)
- First major use when applied by Boeing on the entire Minuteman system for safety evaluation (1964 – 1967, 1968-1999)
- The first technical papers on FTA were presented at the first System Safety Conference, held in Seattle, June 1965
- Boeing began using FTA on the design and evaluation of commercial aircraft, circa 1966
- Boeing developed a 12-phase fault tree simulation program, and a fault tree plotting program on a Calcomp roll plotter
- Adopted by the Aerospace industry (aircraft and weapons)

FTA Historical Stages

The Early Years (1971 – 1980)

- Adopted by the Nuclear Power industry
- Power industry enhanced codes and algorithms
- Some of the more recognized software codes include:
  - Prepp/Kitt, SETS, FTAP, Importance and COMCAN

FTA Historical Stages

The Mid Years (1981 – 1990)

- Usage started becoming international, primarily via the Nuclear Power industry
- More evaluation algorithms and codes were developed
- A large number of technical papers were written on the subject (codes & algorithms)
- Usage of FTA in the software (safety) community
- Adopted by the Chemical industry

FTA Historical Stages

The Present (1991 – 1999)

- Continued use on many systems in many countries
- High quality fault tree Commercial codes developed that operates on PC's
- Adopted by the Robotics and Software industry

FTA Definitions

FT Building Blocks

Node Types:

- **Basic Events (BE)**
- **Gate Events (GE)**
- **Condition Events (CE)**
- **Transfer Events (TE)**

FT Node Types

Basic Event (BE)

- **Failure Event**
  - Primary Failure - basic component failure (circle symbol)
  - Secondary Failure - failure caused by external force (diamond symbol)
- **Normal Event**
  - An event that describes a normally expected system state
  - An operation or function that occurs as intended or designed, such as “Power Applied At Time T1”
  - The Normal event is usually either On or Off, having a probability of either 1 or 0
  - House symbol
- The BE’s are where the failure rates and probabilities enter the FT

FT Node Types

Gate Event (GE)

- A logic operator combining input nodes
- A gate that permits or inhibits fault logic up the tree
- Five basic types
  - AND, OR, Inhibit, Priority AND and Exclusive OR
- Represents a fault state that has further causes to be developed

FT Node Types

Condition Event (CE)

- A condition attached to a gate event
- It establishes a condition that is required in order for the gate event to occur
- Three basic types
  - Inhibit, Priority AND and Exclusive OR

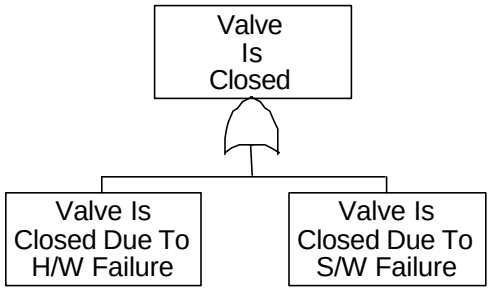
FT Node Types

Transfer Event (TE)

- A pointer to a tree branch
- Indicates a subtree branch that is used elsewhere in the tree (transfer in/out)
- A Transfer always involves a Gate Event node on the tree, and is symbolically represented by a Triangle
- The Transfer is for several different purposes:
  - Starts a new page (for plots)
  - It indicates where a branch is used numerous places in the same tree, but is not repeatedly drawn (Internal Transfer) (MOB)
  - It indicates an input module from a separate analysis (External Transfer)

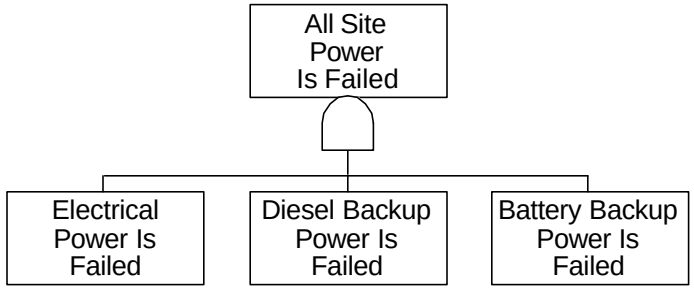
OR Gate

- Causality never passes through an OR gate
  - The input faults are never the cause of the output fault
  - Inputs are identical to the output, only more specifically defined (refined) as to cause

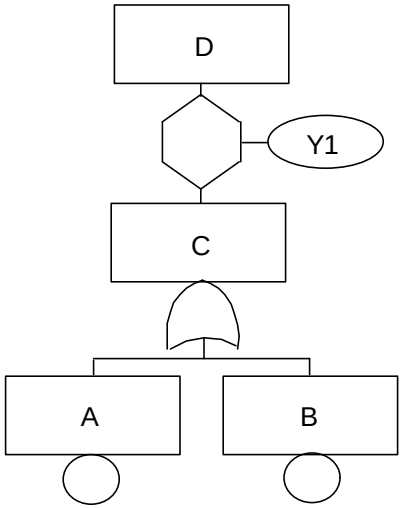


AND Gate

- Specifies a causal relationship between the inputs and the output
  - The input faults collectively represent the cause of the output fault
  - Implies nothing about the antecedents of the input faults

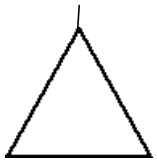


Inhibit Gate

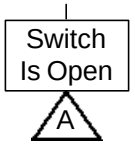


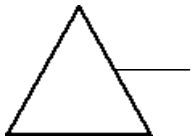
- Both C and Y1 are necessary to cause D
- Y1 is a condition or probability
- Pass through if condition is satisfied
- Essentially an AND gate

Transfer Symbols

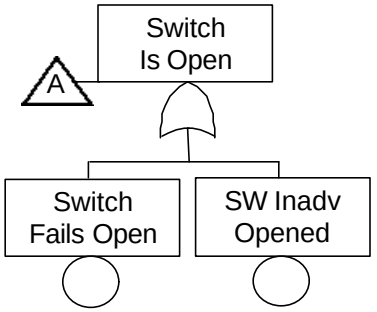


Transfer In





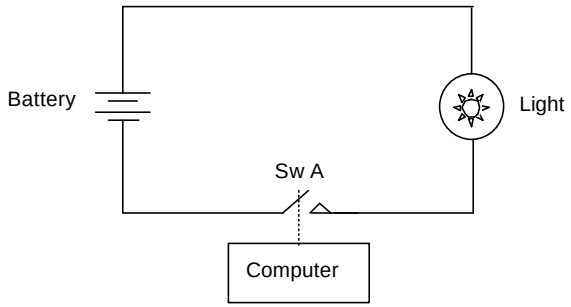
Transfer Out

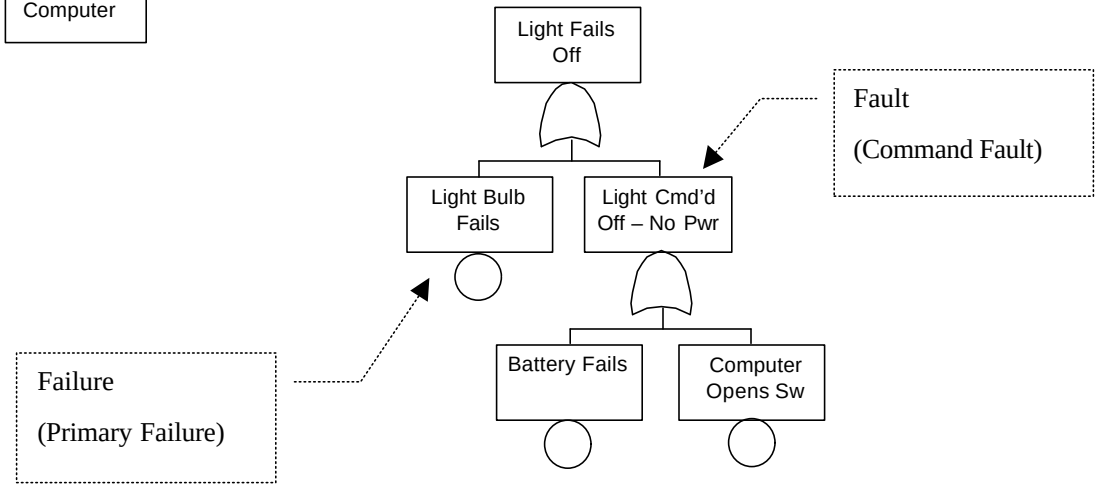


Failure / Fault

- **Failure**
  - The occurrence of a ***basic component failure***.
  - The result of an internal inherent failure mechanism, thereby requiring no further breakdown.
  - Example - *Resistor R77 Fails in the Open Circuit Mode.*
- **Fault**
  - The occurrence or existence of an ***undesired state*** for a component, subsystem or system.
  - The result of a failure or chain of faults/failures; can be further broken down.
  - The component operates correctly, except at the wrong time, because it was commanded to do so.
  - Example – The light is failed off because the switch failed open, thereby removing power.

Failure / Fault Example





All failures are faults, but not all faults are failures.

Primary, Secondary, Command Fault

- **Primary Fault / Failure**
  - A *component failure* that cannot be further defined at a lower level.
  - Example – diode inside a computer fails due to materiel flaw.
- **Secondary Fault / Failure**
  - A component failure that can be further defined at a lower level, but is *not defined in detail* (ground rules).
  - Example – computer fails (don't care about detail of why).
  - A component failure that is *caused by an external force* to the system, can be further defined.
  - Example – Fuel tank ruptures due to little boy shooting it with an armor piercing bow and arrow.
  - They are also important when performing a Common Cause Analysis.

Primary, Secondary, Command Fault

- **Command Fault / Failure**
  - A fault state that is commanded by an upstream fault / failure.
  - Normal operation of a component, except in an inadvertent or untimely manner. The normal, but, undesired state of a component at a particular point in time.
  - The component operates correctly, except at the wrong time, because it was commanded to do so by upstream faults.
  - Example – a bridge opens (at an undesired time) because someone accidentally pushed the Bridge Open button.

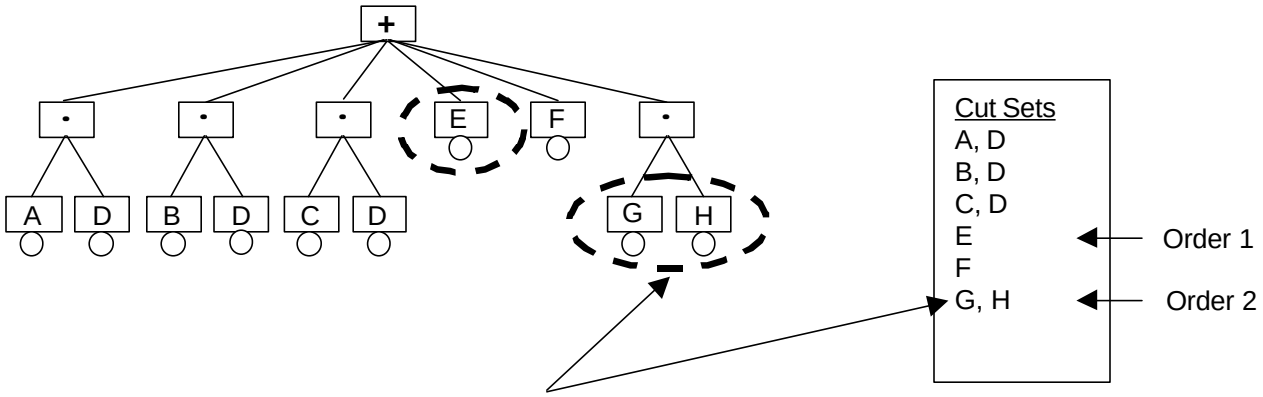
System Complexities Terms

- **MOE**
  - A Multiple Occurring Event or failure mode that occurs more than one place in the FT
  - Also known as a redundant or repeated event
- **MOB**
  - A multiple occurring branch
  - A tree branch that is used in more than one place in the FT
  - All of the Basic Events within the branch would actually be MOE's
- **Branch**
  - A subsection of the tree (subtree), similar to a limb on a real tree
- **Module**
  - A subtree or branch
  - An independent subtree that contains no outside MOE's or MOB's, and is not a MOB

Cut Set Terms

- **Cut Set**
  - A set of events that together cause the tree Top UE event to occur
- **Min CS (MCS)**
  - A CS with the minimum number of events that can still cause the top event
- **Super Set**
  - A CS that contains a MCS plus additional events to cause the top UE
- **Critical Path**
  - The highest probability CS that drives the top UE probability
- **Cut Set Order**
  - The number of elements in a cut set
- **Cut Set Truncation**
  - Removing cut sets from consideration during the FT evaluation process
  - CS's are truncated when they exceed a specified order and/or probability

Cut Sets



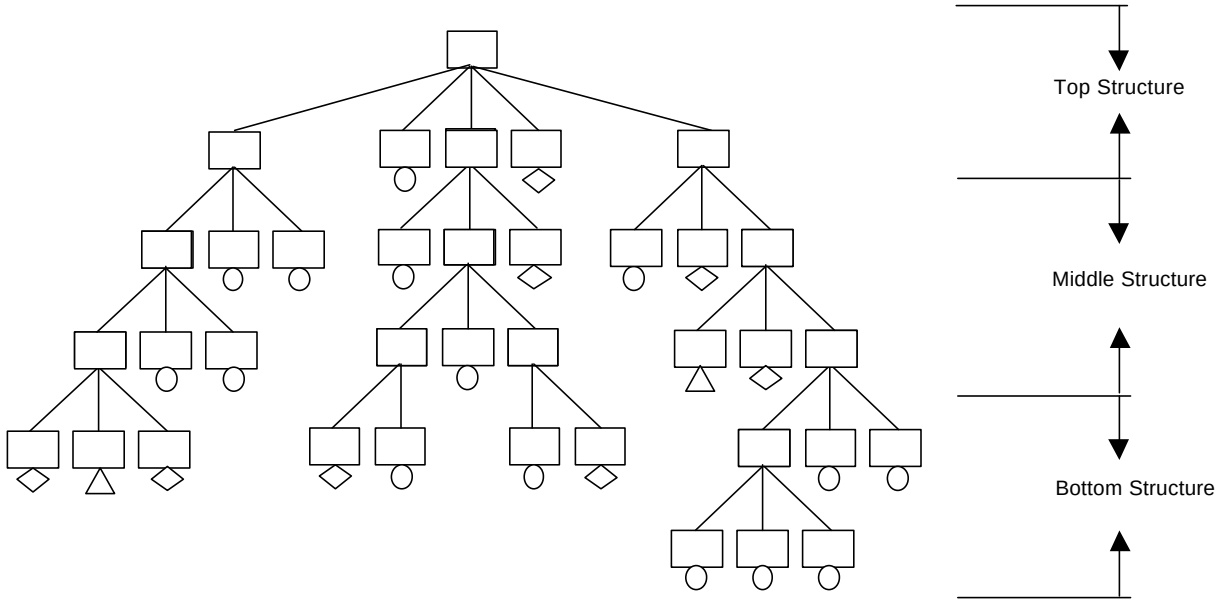
AND gate means that both G & H must occur. Since they go directly to top, they comprise a CS, denoted by {G, H}.

Cut Set (CS)

A unique set of events that cause the Top UE to occur.

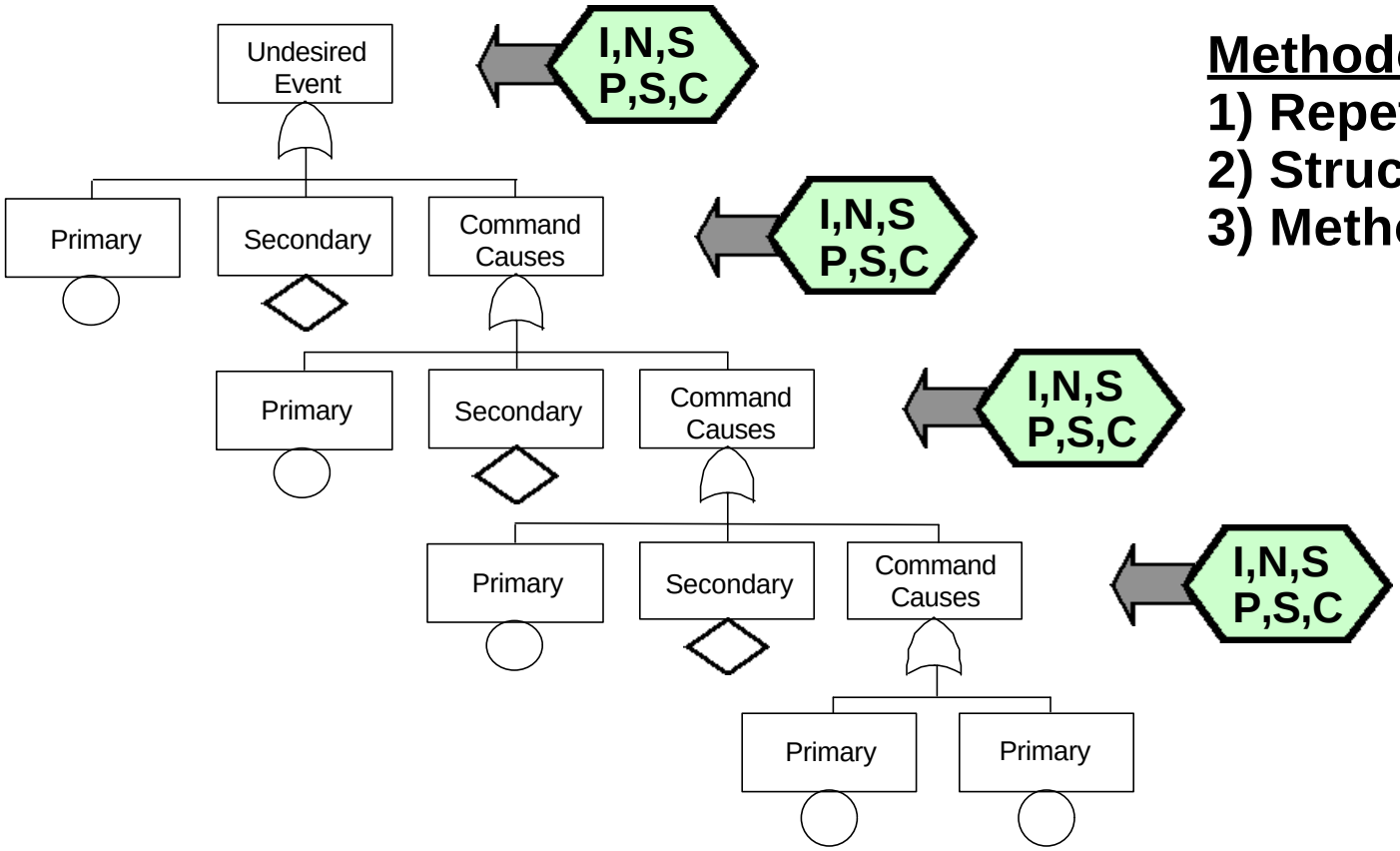
FTA Construction

Construction Process - Overview



- Tree is developed in Layers, Levels, and Branches
- Levels represent various stages of detail
  - Top - shapes tree, combines systems
  - Middle - subsystems, functions, phases, fault states
  - Bottom - basic events, component failures

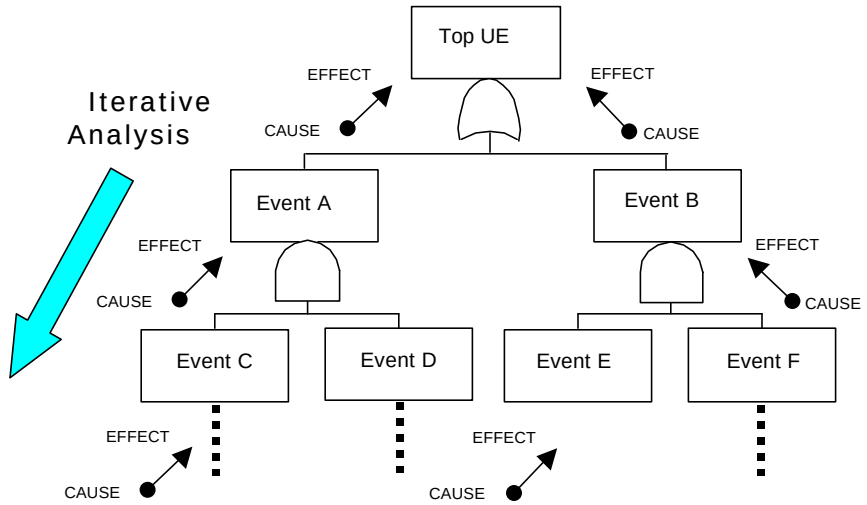
FT Construction



Methodology
1) Repetitive
2) Structured
3) Methodical

I,N,S=Immediate, Necessary, Sufficient
P,S,C=Primary, Secondary, Command

FT Construction -- Iterative Process



1) Review the Gate Event under investigation
2) Identify all the possible causes of this event
3) Ensure you do not jump ahead of a possible cause event
4) Identify the relationship or logic of the Cause-Effect events
5) Structure the tree with these events and logic gate
6) Keep looking back to ensure identified events are not repeated
7) Repeat the process for the next gate.

Node Construction -- Three Step Process

- Construction at each gate node involves a 3 step process:
  - Step 1 – Immediate, Necessary and Sufficient (INS)
  - Step 2 – Primary, Secondary and Command (PSC)
  - Step 3 – State of the System or Component

Step 1

Step 1 - Immediate, Necessary and Sufficient (INS)?

- Read the IG event wording
- Identify all ***Immediate***, ***Necessary*** and ***Sufficient*** events to cause the IG event
- Structure the INS casual events with appropriate logic:
  - Immediate – do not skip past events
  - Necessary – include only what is actually necessary
  - Sufficient – do not include more than the minimum necessary
- Mentally test the events and logic until satisfied

Step 2

Step 2 - Primary, Secondary and Command (PSC)?

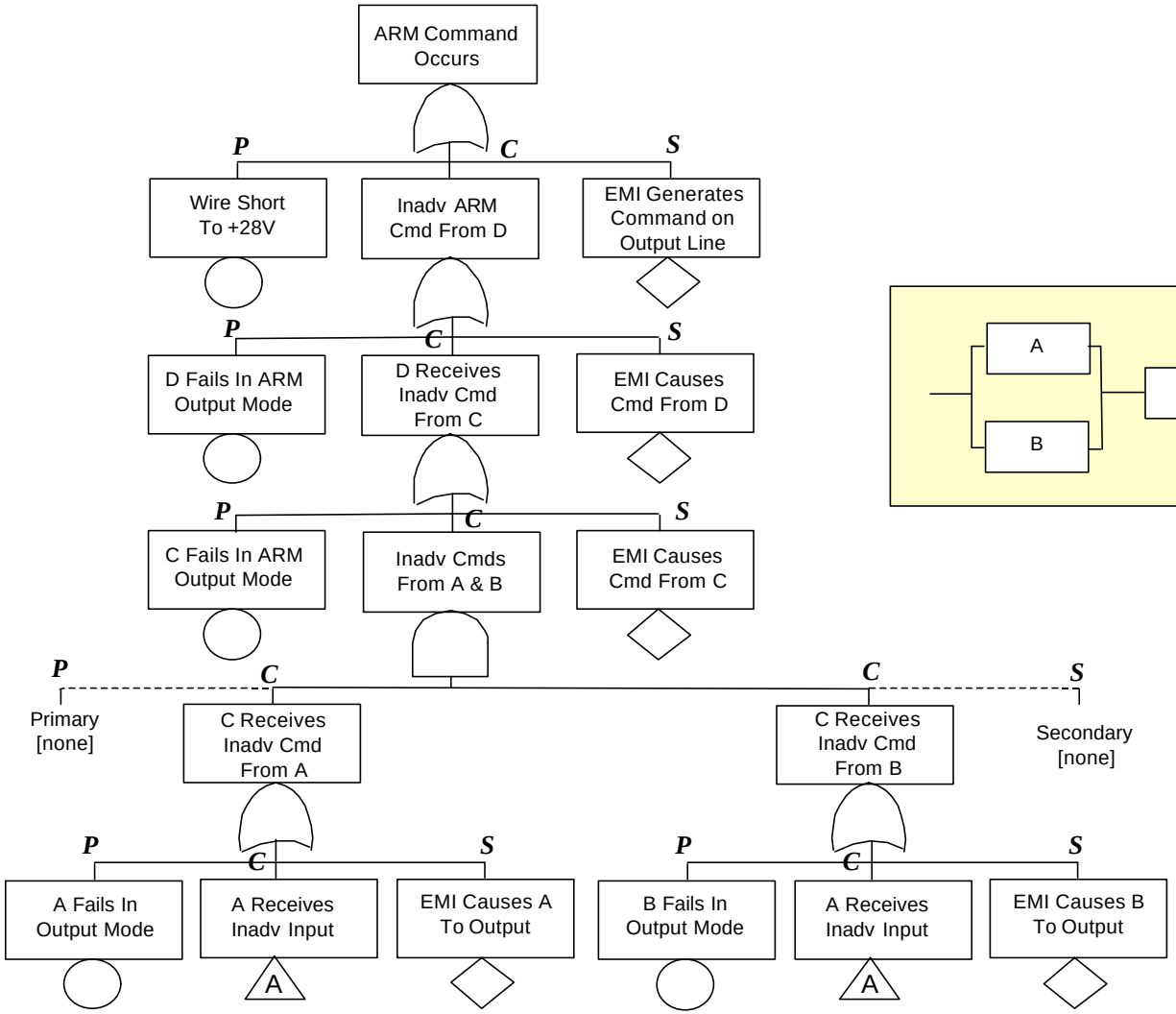
- Read the IG event wording
- Ask “what is *Immediate*, *Necessary* and *Sufficient*” to cause event (Step 1)
- Word Gate events in terms of Input or Output
- Consider the type of fault path for each Enabling Event
  - identify each causing event as one of the following path types
    - ☞Primary Fault
    - ☞Secondary Fault
    - ☞Command Fault (Induced Fault, Sequential Fault)
  - structure the sub events and gate logic from the path type
  - any event that is not a BE (component) event is another Enabling Event (Command Path)

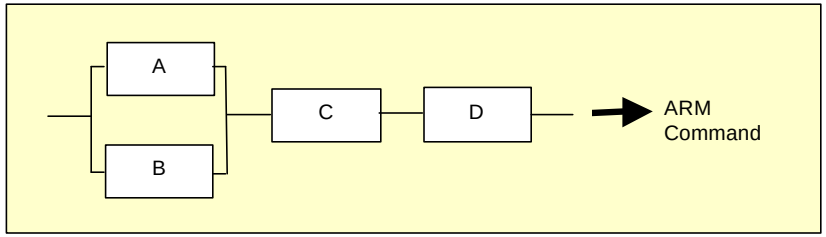
Step 3

Step 3 - State of the System or Component?

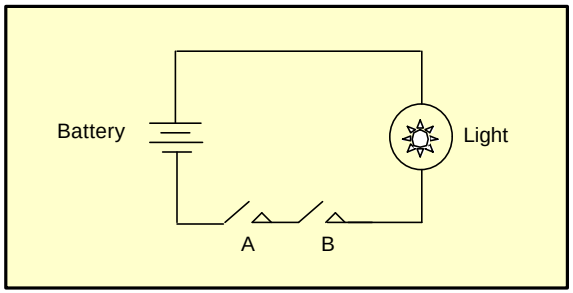
- Read the IG event wording
- Ask " is the IG a *State of the System* or *State of the Component* event"
  - State of the Component is identified by being at the component level
  - State of the System is identified by being composed of more IG events
  - If its not State of the Component then it must be a State of the System

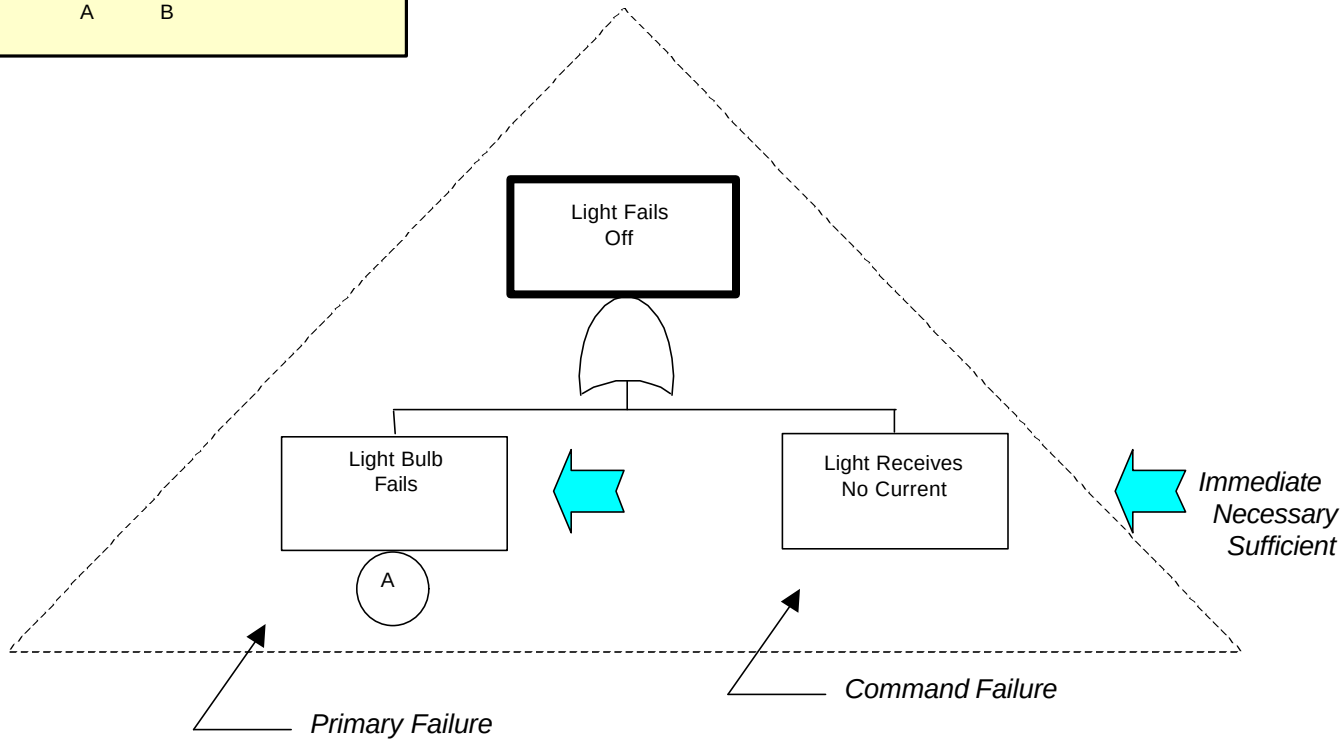
Example



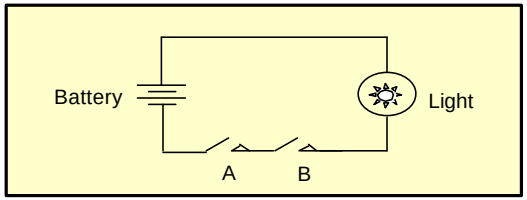


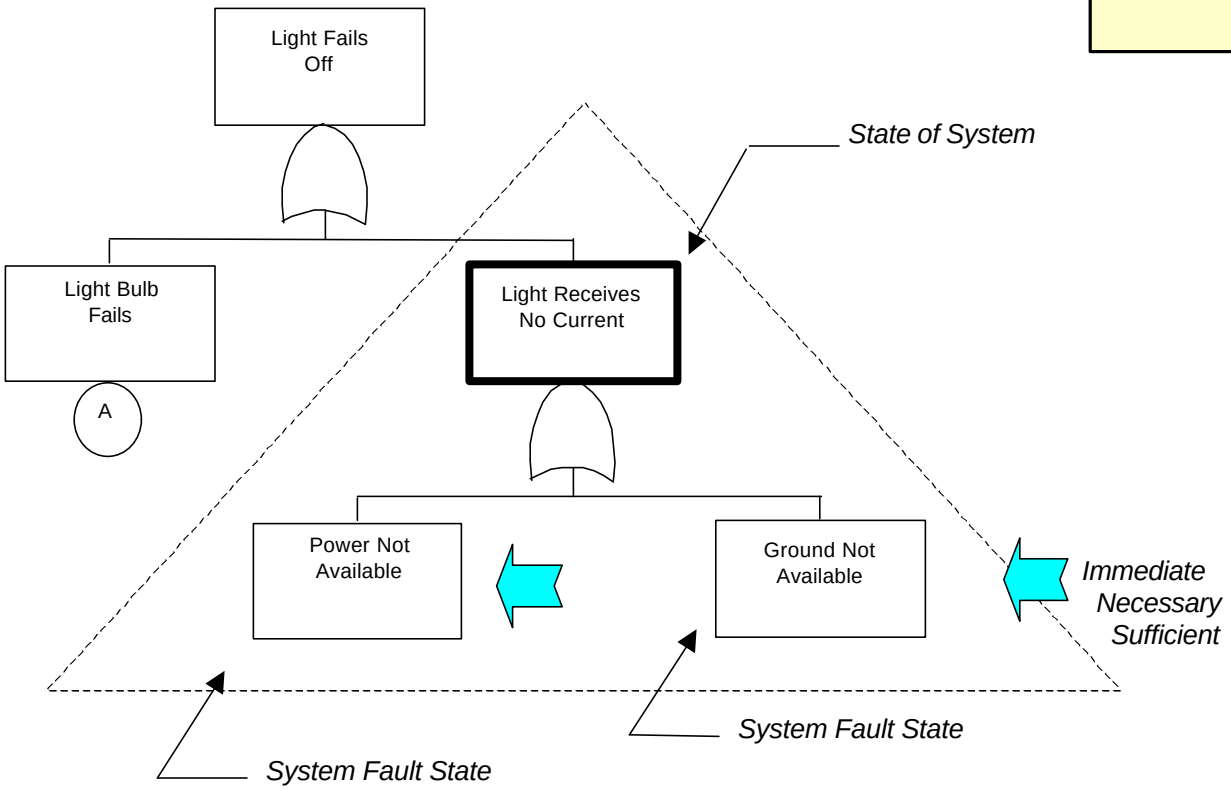
Construction Example





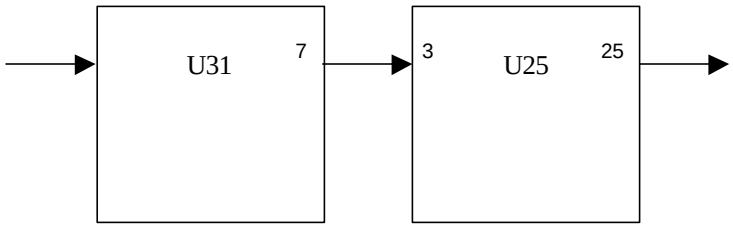
Construction Example (continued)

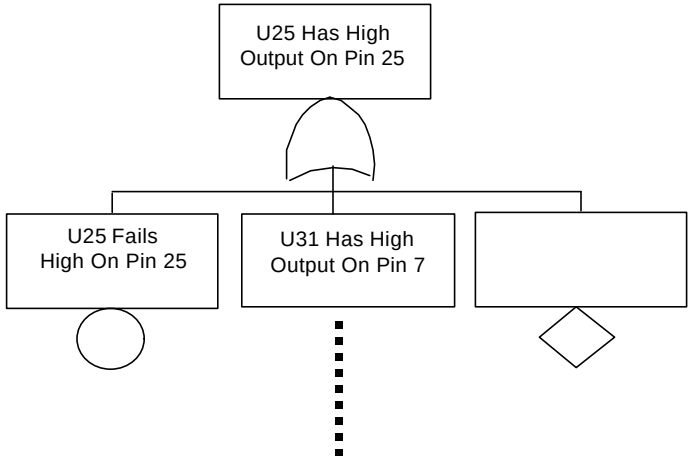




Node Wording Is Important

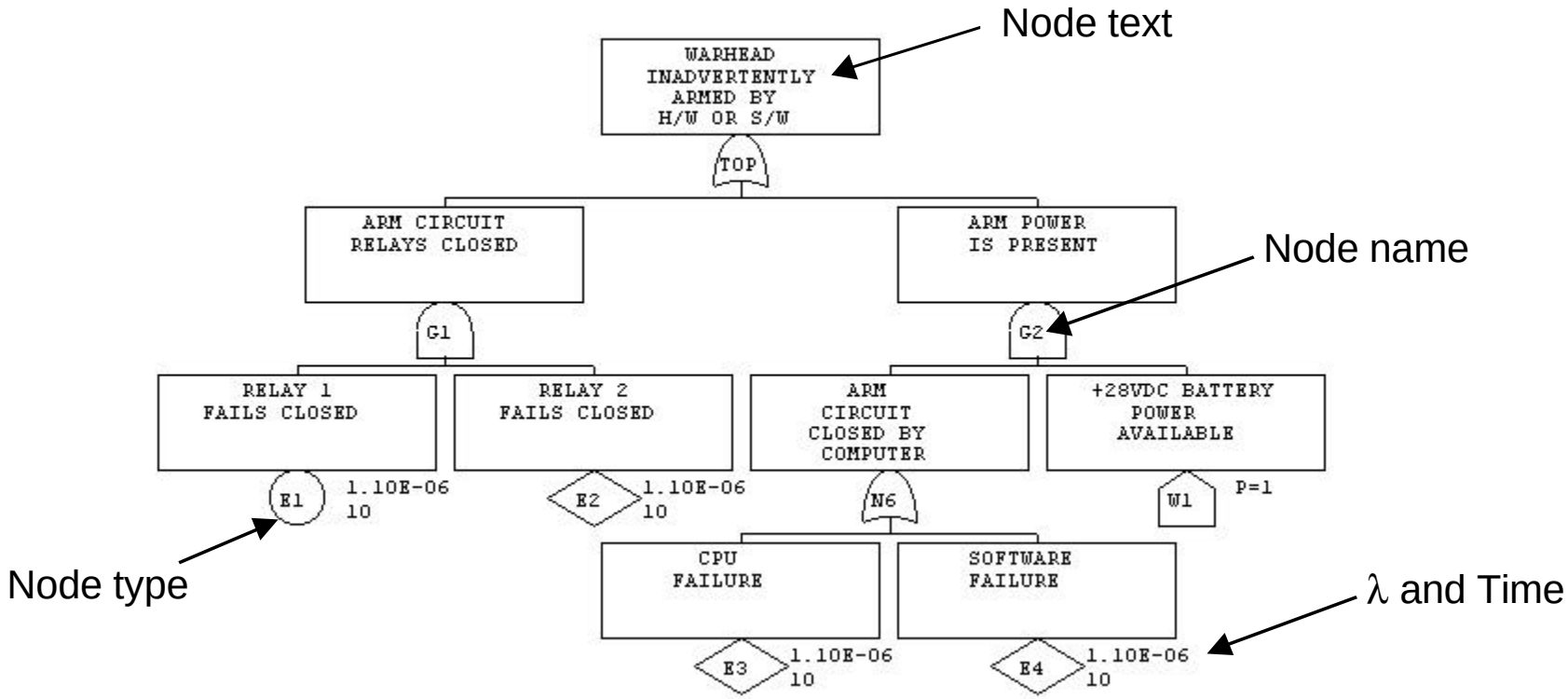
- Node wording is important and helps the analysis process
- Be clear and precise
- Always express device transitions in terms of the output device that causes the transition
- Do not use *failure* and *fault* terms for state transitions if not necessary





FT Data Requirements

- Node name √
- Node text √
- Node type √
- Node failure rate & exposure time √



FTA Mathematics

Basic Reliability Equations

- $R = e^{-\lambda T}$
- $R + Q = 1$
- $Q = 1 – R = 1 - e^{-\lambda T}$
- Approximation
  - When $\lambda T < 0.001$ then $Q \approx \lambda T$

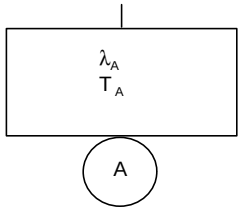
- Where:
  - R = Reliability or probability of success
  - Q = Unreliability or probability of failure
  - λ = component failure rate = 1 / MTBF
  - T = time interval (mission time or exposure time)

Effects of Failure Rate & Time

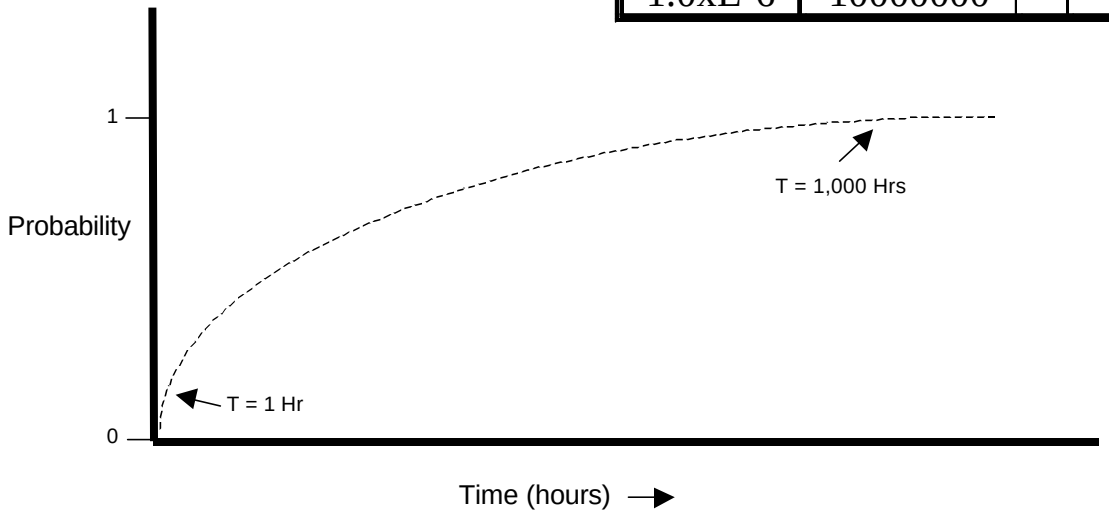
- The longer the mission (or exposure time) the higher the probability of failure
- The smaller the failure rate the lower the probability of failure

Example

The Effect of Exposure Time On Probability



λ_A (FPH)	T_A (HRS)		$P_A=1 - e^{-\lambda T}$
1.0xE-6	1		9.99xE-7
1.0xE-6	10		9.99xE-6
1.0xE-6	100		9.99xE-5
1.0xE-6	1000		9.99xE-4
1.0xE-6	10000		9.95xE-3
1.0xE-6	100000		0.095
1.0xE-6	1000000		0.6321
1.0xE-6	10000000		0.99995



Axioms of Boolean Algebra

[A1]	ab = ba	Commutative Law
[A2]	a + b = b + a	
[A3]	(a + b) + c = a + (b + c) = a + b + c	Associative Law
[A4]	(ab)c = a(bc) = abc	
[A5]	a(b+c) = ab + ac	Distributive Law

Theorems of Boolean Algebra

[T1] $a + 0 = a$

[T2] $a + 1 = 1$

[T3] $a \bullet 0 = 0$

[T4] $a \bullet 1 = a$

[T5] $a \bullet a = a$ ✓

[T6] $a + a = a$ ✓

} Idempotent Law

[T7] $a \bullet \overline{a} = 0$

[T8] $a + \overline{a} = 1$

[T9] $a + ab = a$ ✓

[T10] $a(a + b) = a$ ✓

} Law of Absorption

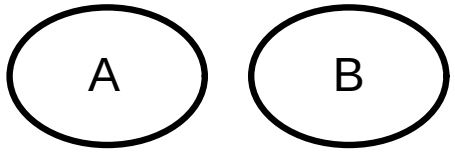
[T11] $a + \overline{a}b = a + b$

where $\overline{a}$ = not a

Probability

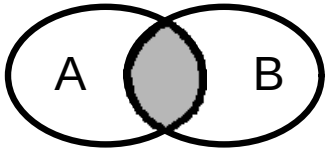
Union

For two events A and B, the union is the event {A or B} that contains all the outcomes in A, in B, or in both A and B.



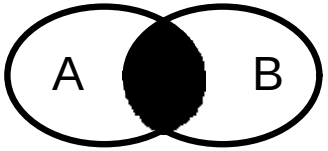
Case 1 - Disjoint Events

$P=P(A) + P(B)$



Case 2 - Non Disjoint Events

$P=P(A) + P(B) - P(A)P(B)$



Case 3 - Mutually Exclusive Events

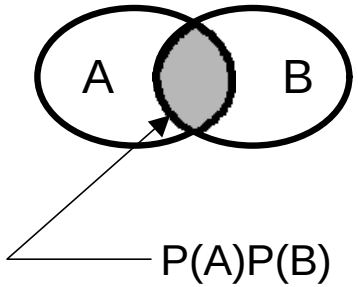
$P=P(A) + P(B) - 2P(A)P(B)$

Note - Exclusive OR is not the same as Disjoint.

Probability

Intersection

For two events A and B, the intersection is the event {A and B} that contains the occurrence of both A and B.



Case 1 - Independent Events

$P=P(A)P(B)$

Case 2 - Dependent Events

$P=P(A)P(B/A)$

CS Expansion Formula

$P=\Sigma(\text{singles}) - \Sigma(\text{pairs}) + \Sigma(\text{triples}) - \Sigma(\text{fours}) + \Sigma(\text{fives}) - \Sigma(\text{sixes}) + \bullet\bullet\bullet$

CS {A; B; C; D}

$$P = (P_A + P_B + P_C + P_D) - (P_{AB} + P_{AC} + P_{AD} + P_{BC} + P_{BD} + P_{CD}) + (P_{ABC} + P_{ABD} + P_{ACD} + P_{BCD}) - (P_{ABCD})$$

$$P=P_A + P_B + P_C + P_D - (P_{AB} + P_{AC} + P_{AD} + P_{BC} + P_{BD} + P_{CD}) + (P_{ABC} + P_{ABD} + P_{ACD} + P_{BCD}) - (P_{ABCD})$$

Size and complexity of the formula depends on the total number of cut sets and MOE's.

FTA Evaluation

FT Evaluation - Purpose

- Obtaining the results and conclusions from the FT
- Using the FT for its intended purpose
  - evaluate risk / decision making
  - determining if the UE is safe
  - identify root causes
  - identify critical components and paths
- Using the FT to impact design
  - identify weak links
  - evaluate impact of changes

Evaluation Process

- Process
  - generate Cut Sets √
  - apply failure data
  - compute probabilities
  - compute criticality measures

Types

Two type of Evaluation:

- Qualitative
  - Cut Sets only
- Quantitative
  - Cut Sets and Probability
  - Importance Measures

Requirements

- Requires knowledge and use of:
  - FT mathematics (probability and Boolean algebra)
  - FT algorithms
  - FT approximation methods
  - FT computer programs

Cut Set

- A unique set of events that together cause the Top UE event to occur
- One (of possibly many) root causes of the Top UE
- A CS can consist of one event or 10 simultaneous events

The Value of Cut Sets

- Cut Sets identify which component failures and/or events can cause an accident or undesired event (UE) to occur
- CS's show which unique event combinations can cause the UE
- CS's provide the mechanism for probability calculations
- Cut Sets reveal the critical and weak links in a system design
  - high probability
  - bypass of intended safety or redundancy features

Note:
Always check all CS's against the system design to make sure they are valid and correct.

Qualitative Evaluation

- Non-numerical
- Process
  - Obtain the entire list of Min Cut Sets from the FT
- Qualitatively evaluate and analyze the Cut Sets for design problems/concerns

Note:
Slightly subjective than quantitative evaluation.

Value of Qualitative Evaluation

- CS importance by order number
  - Lower order CS are generally more important
- Component importance by number of times it appears in different CS's
- Analyze the CS for:
  - identifying (unexpected) root cause combinations
  - design weak points
  - bypass of intended safety features
  - common cause problems

Quantitative Evaluation

- Numerical – Probability of Event occurrence
- Process
  - Obtain the entire list of Min Cut Sets from the FT
  - Compute FT probabilities from the Min CS's
  - Compute FT Importance Measures from the CS's
  - Requires component failure rates & exposure times
- Quantitatively evaluate and analyze the Cut Sets and Probabilities for design problems/concerns

Value of Quantitative Evaluation

- Tree & gate probability estimates
- Probabilistic Risk Assessment (PRA)
- More precise evaluation of FT, not as subjective
- Quantitative measures for:
  - FT (UE) probability
  - component criticality & importance
  - CS criticality & importance
  - critical path ranking

Basic Evaluation Methods

- Manual
  - possible for small/medium noncomplex trees
- Computer
  - required for large complex trees
  - two approaches
    - analytical
    - simulation

Methods For Finding Min CS

- Boolean reduction
- Bottom up reduction algorithms
  - MICSUP algorithm
- Top down reduction algorithms
  - MOCUS algorithm
- BDD (Binary Decision Diagram)
- Min Terms method (Shannon decomposition)
- Modularization methods
- Genetic algorithms

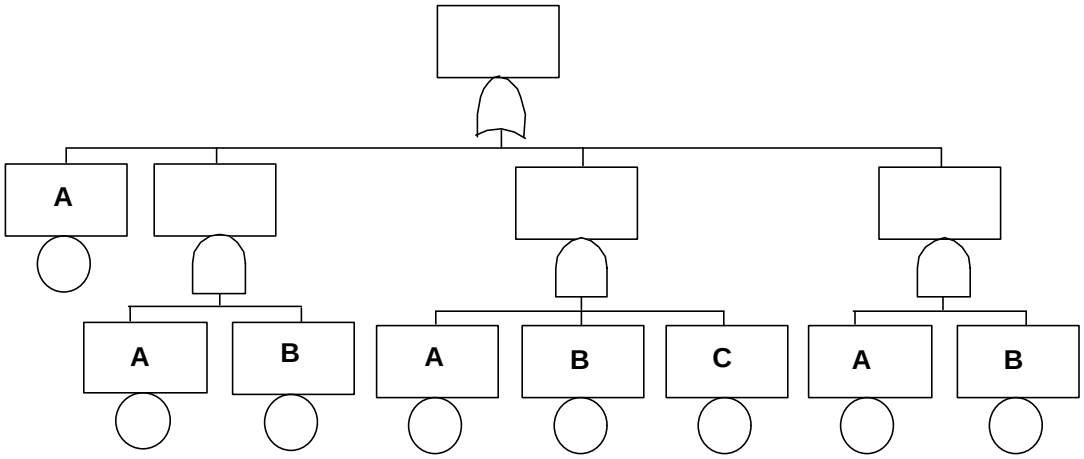
Evaluation Trouble Makers

- Tree size
- Tree Complexity
  - from redundancy (MOE's)
  - from large AND/OR combinations
- Exotic gates
- Computer limitations
  - speed
  - memory size
  - software language

Min CS

- A CS with the minimum number of events that can still cause the top event
- The true list of CS's contributing to the Top
- The final CS list after removing all SCS and DupCS
- Additional CS's are often generated, beyond the MinCS's
  - Super Cut Sets (SCS) – result from MOE's
  - Duplicate Cut Sets (DupCS) - result from MOE's or AND/OR combinations
- Why eliminate SCS and DupCS?
  - laws of Boolean algebra
  - would make the overall tree probability slightly larger (erroneous but conservative)

Min CS



Cut Sets:

A

A,B ← **SCS**

A,B,C ← **SCS**

A,B ← **DupCS, SCS**

Min Cut Sets:

A

FTA Pitfalls

Pitfall #1 – FT Design

- Lack of proper FT planning and design can result in problems
  - Might necessitate restructure of entire tree
  - Might necessitate renaming all events in tree
  - Rework will cost time and money
- Must plan ahead
  - Leave room for future tree expansion
  - Allow for possible future changes in tree without repercussions
  - Structure tree carefully, later changes can impact entire tree
    - ☞ Carefully develop a name scheme - events, MOE's, transfers
- Large FT's require more design foresight
  - Develop organized plan when several analysts work on same FT

Poorly Planned FT

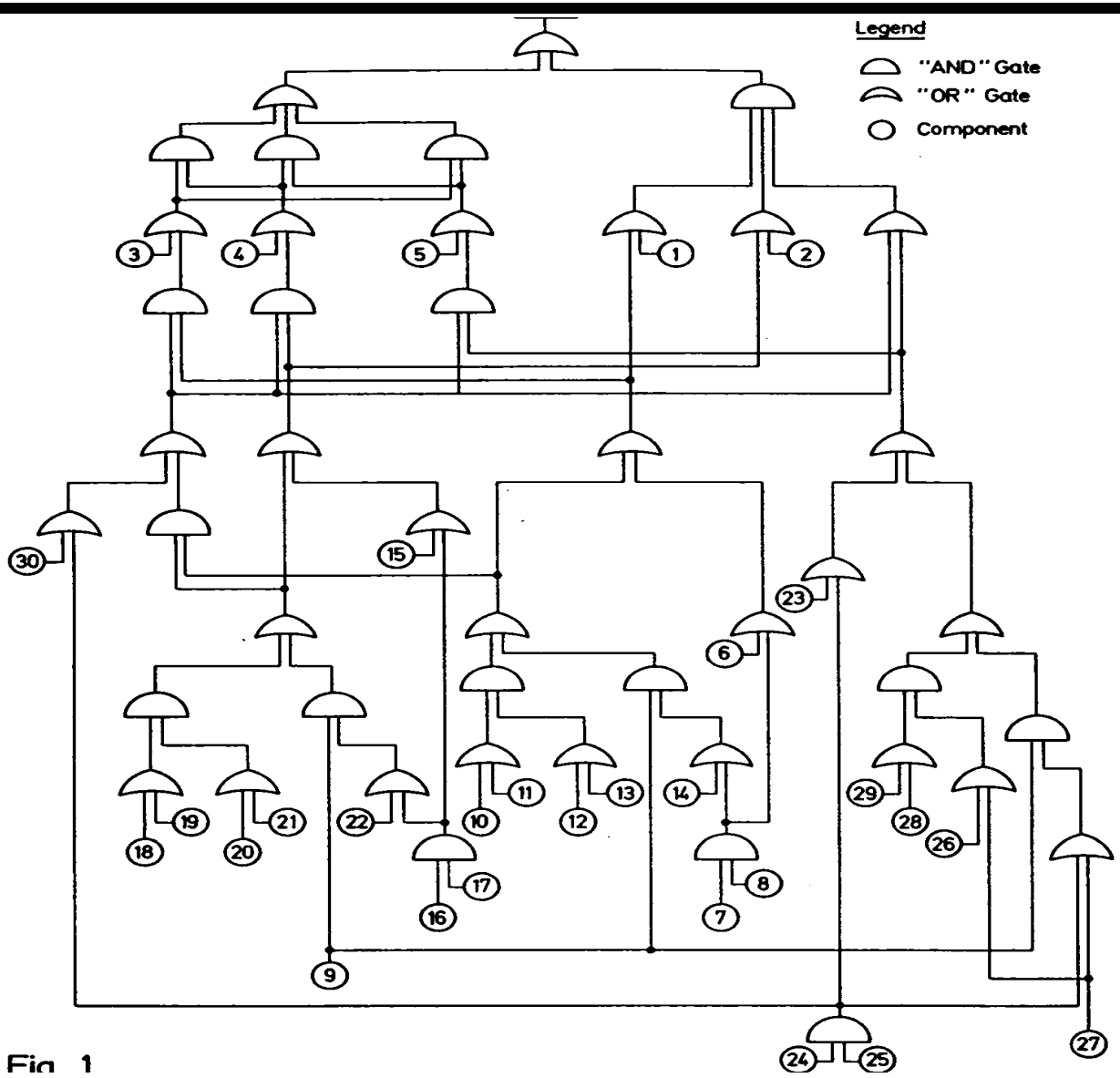


Fig. 1

Pitfall #3 – AND Gate Overconfidence

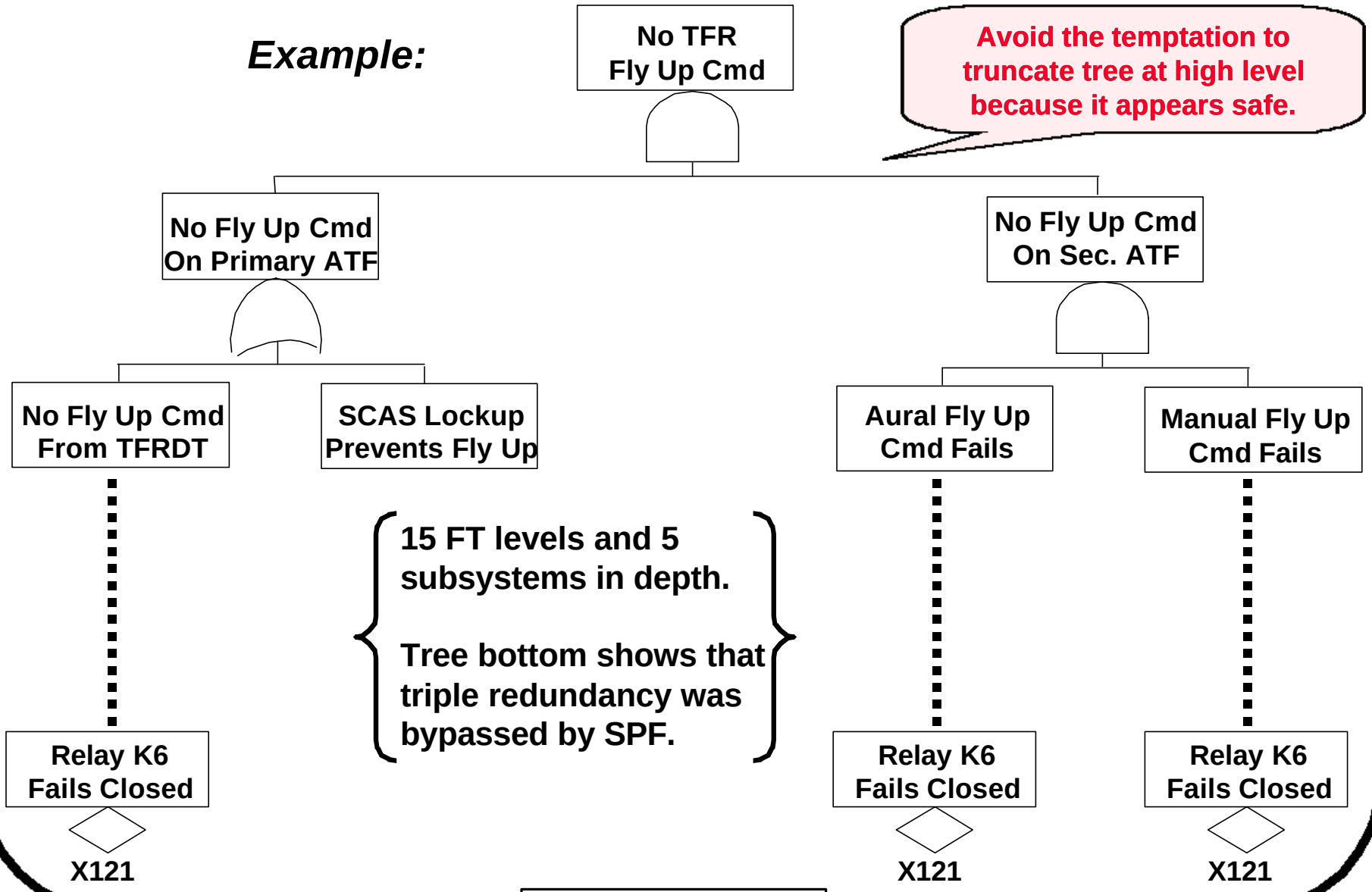
- An over confident assumption is often made that a system is safe because it requires at least 3 or 4 inputs to an AND gate
- The probability for a 3 input AND gate is usually very small ($10^{-3} \cdot 10^{-3} \cdot 10^{-3} = 10^{-9}$)
- However, an MOE in each branch of the AND gate can reduce the probability to a SPF (making the probability 10^{-3})

The effect of a Repeated event.

Or, common mode failure.

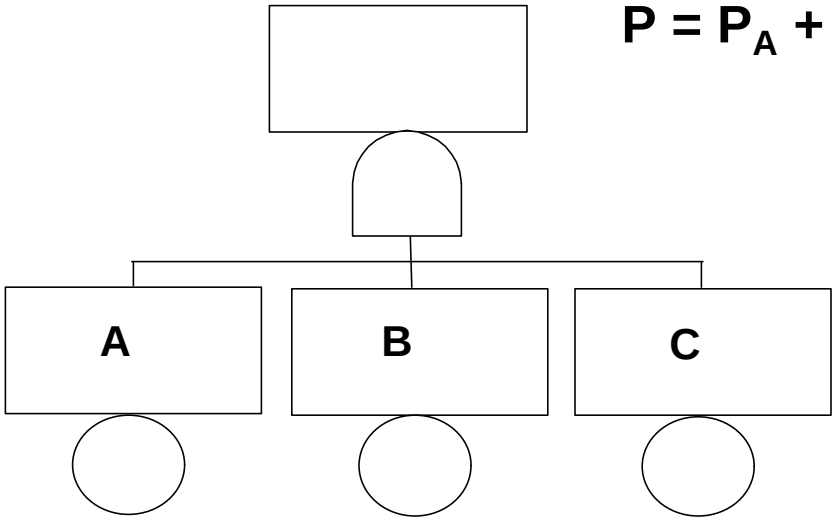
Pitfall #3 – AND Gate Overconfidence

Example:



Pitfall #4 – Incorrect Exposure Time

- If the Time in P=1.0 – $e^{-\lambda T}$ is not correct, errors are injected into the FT probability calculations
- Note the quantitative impact for different exposure times

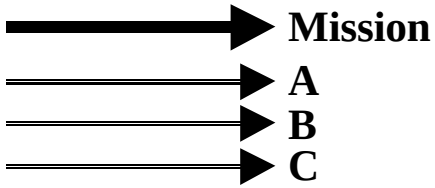
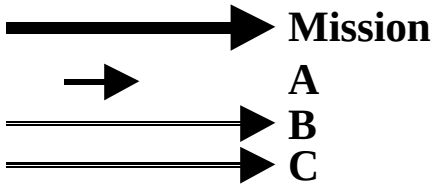
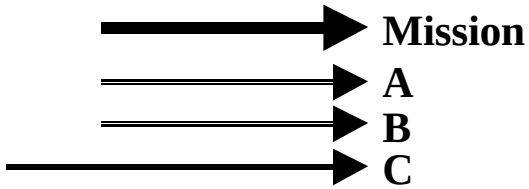


$$P = P_A + P_B + P_C$$

$\lambda_A = 1.0 \times 10^{-6}$
$\lambda_B = 1.0 \times 10^{-6}$
$\lambda_C = 1.0 \times 10^{-6}$

The impact of exposure time.

Exposure Time

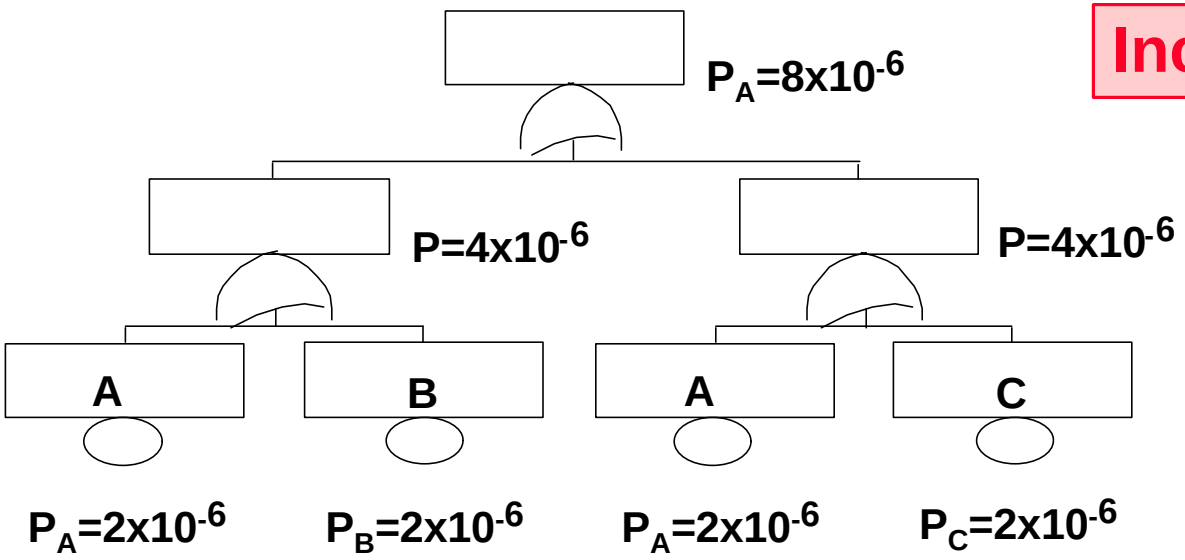
Scenario	Calculation	Time
Standard	$P_A = (1x10^{-6})\ (10) = 10x10^{-6}$ $P_B = (1x10^{-6})\ (10) = 10x10^{-6}$ $P_C = (1x10^{-6})\ (10) = 10x10^{-6}$ $P = (10x10^{-6}) \times (10x10^{-6}) \times (10x10^{-6})$ $= 1{,}000x10^{-18}$ $= 1.0x10^{-15}$	
Component used for short duration	$P_A = (1x10^{-6})\ (0.1) = 0.1x10^{-6}$ $P_B = (1x10^{-6})\ (10) = 10x10^{-6}$ $P_C = (1x10^{-6})\ (10) = 10x10^{-6}$ $P = (0.1x10^{-6}) \times (10x10^{-6}) \times (10x10^{-6})$ $= 10x10^{-18}$ $= 1.0x10^{-17}$	
Standby component, unchecked (latent fault)	$P_A = (1x10^{-6})\ (10) = 10x10^{-6}$ $P_B = (1x10^{-6})\ (10) = 10x10^{-6}$ $P_C = (1x10^{-6})\ (1000) = 1x10^{-3}$ $P = (10x10^{-6}) \times (10x10^{-6}) \times (1x10^{-3})$ $= 100x10^{-15}$ $= 1.0x10^{-13}$	

Pitfall #10 – Gate Calculations Errors

Gate Calculations Errors due to MOEs

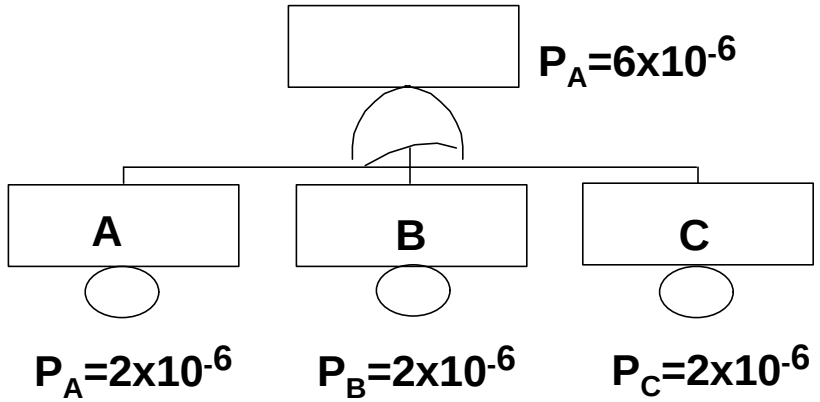
- An often used method of tree calculation is the bottom-up gate to gate calculation
- This method is valid as long as the tree has no MOEs in it
- If MOEs exist, this method generally produces very erroneous results
  - as shown below, error can range from very large to no error, depending on tree structure
- Some computer programs print the probability calculations for each gate on the tree, which is very dangerous if MOEs exist
- Must resolve MOE's for correct tree probability calculation

MOE Error Example 1



Incorrect

Correct



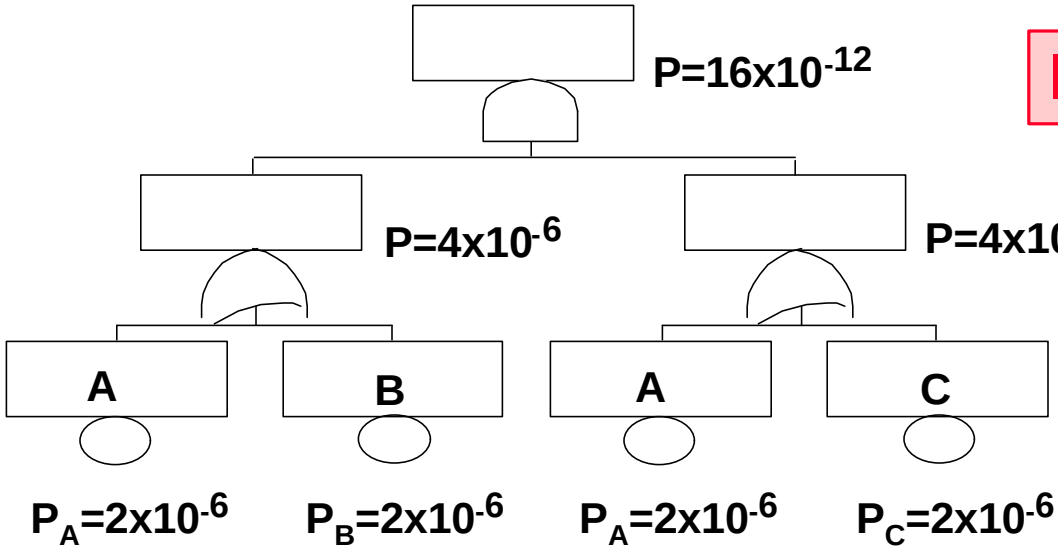
Cut Sets = A ; B ; C

$P = P_A + P_B + P_C$

$= (2x10^{-6}) + (2x10^{-6}) + (2x10^{-6})$

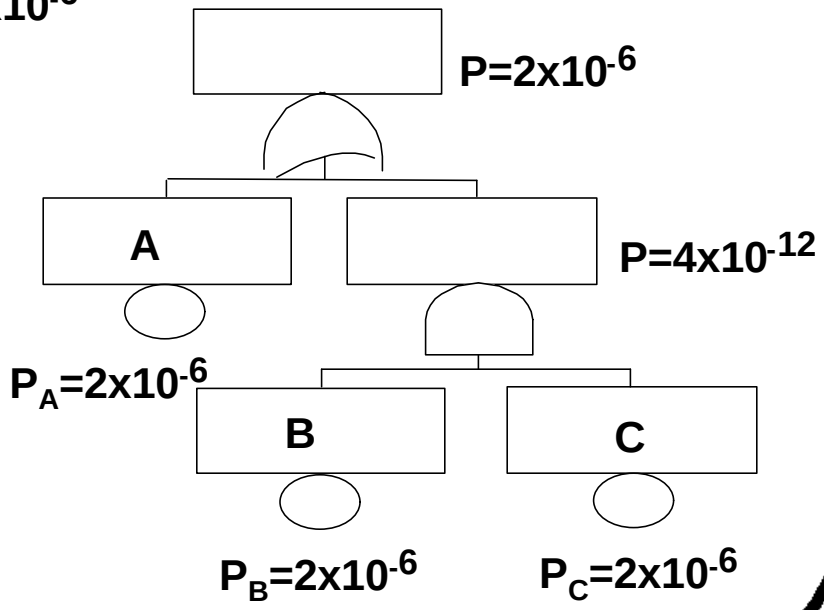
$= 6x10^{-6}$ [upper bound]

MOE Error Example 2



Incorrect

Correct



Cut Sets = A ; B,C

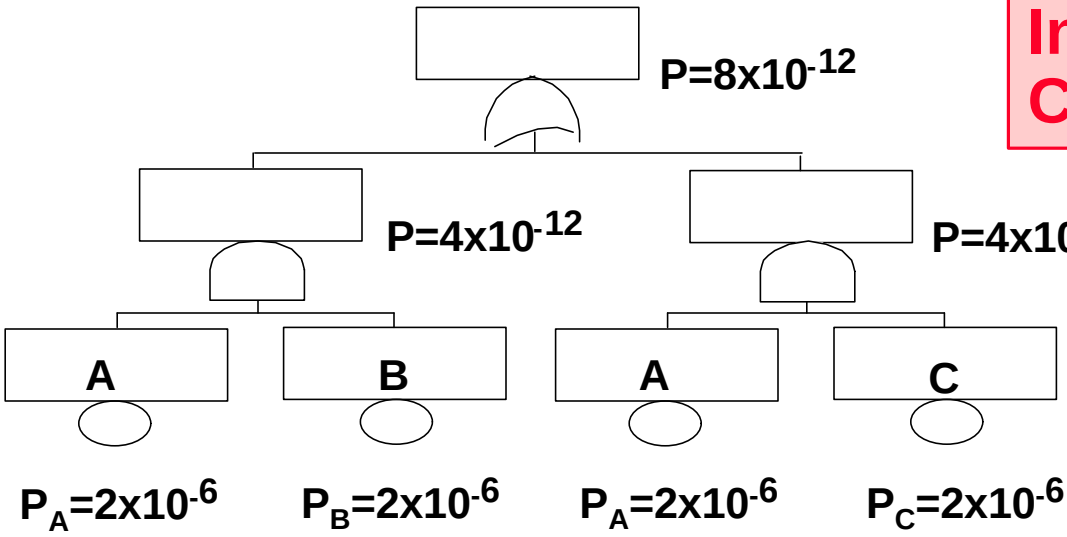
$P = P_A + P_B P_C$

$= (2x10^{-6}) + (2x10^{-6})(2x10^{-6})$

$= 2x10^{-6} + 4x10^{-12}$

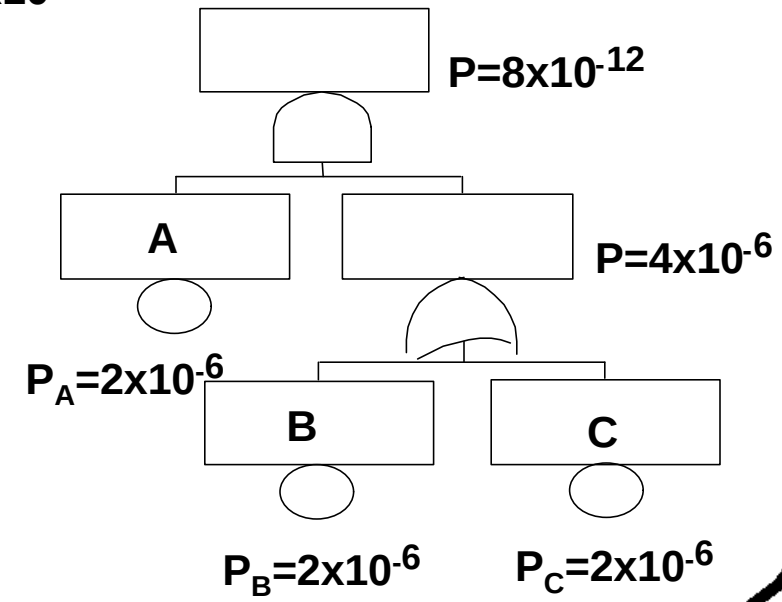
$= 2x10^{-6}$ [upper bound]

MOE Error Example 3



Incorrect but Correct

Correct



Cut Sets = A,B ; A,C

$$P = P_A P_B + P_A P_C$$
$$= (2\times10^{-6})(2\times10^{-6}) + (2\times10^{-6})(2\times10^{-6})$$
$$= 4\times10^{-12} + 4\times10^{-12}$$
$$= 8\times10^{-12}$$

[upper bound]

FTA Rules

Rule #1

Rule #1 – Know The Purpose And Strengths Of FTA

- Use the right tool
- Use the tool correctly
- Remember, FTA is a tool for:
  - root cause deductive analysis
  - identifies events contributing to an Undesired Event
  - computes the probability of an Undesired Event
  - measures the relative impact of a design fix
  - fault path diagrams for presentation
- Know when to use another tool

Rule #2

Rule #2 -- Know The Purpose And Objectives Of Your FTA

- Solve the right problem / do the right analysis
- Establish a problem/solution statement
  - what is the problem statement
  - what are the solution requirements
  - show how FTA results will satisfy or solve the problem
  - test potential FTA results against the problem
- Make sure top Undesired Event (UE) is correct and reasonable
  - correct/reasonable model
  - don't solve the wrong problem
  - don't try the impossible
  - make sure analysis will meet desired objectives/goals

Rule #3

Rule #3 -- Establish Your FTA Ground Rules

- Define and document assumptions
- Scope the problem
  - size, level of analysis, level of detail
- Set analysis scope and boundaries
- Establish analysis definitions
- Make sure top UE is correct and reasonable (do the right analysis)
- Publish FTA ground rules before starting (living document)
  - definitions, scope, boundaries, level of detail and analysis depth
  - construction rules, FT format
- Obtain agreement on ground rules
  - design team, customer

Rule #4

Rule #4 -- Intentionally Design Your Fault Tree

- Follow FTA ground rules and formats
  - Make checks against ground rules
- Establish name convention for Events, MOEs and Transfers
  - use a methodology
  - by hardware type, supplier, subsystem
  - short names are usually better (long names becomes burdensome, time consuming)
- Maintain event databases and cross references
  - basic failure events, gate events, condition events, MOE's, transfers
- Establish tree structure approach
  - functional or subsystem

Rule #4 (continued)

- Determine level of analysis detail
  - subsystem, LRU, component
- Use gate types cautiously
  - AND, OR and Inhibit gates do almost everything
  - if you think an exotic gate is necessary, that's the first clue to re-analyze your problem
- Be very descriptive in writing event text
  - avoid using word "fail" -- not enough information
  - "power supply fails" vs. "power supply does not provide +5 VDC"
  - do not use the terms primary failure or secondary failure (provide more description)
- Use FT programs and design around their capabilities

Rule #4 (continued)

- Maintain tree metrics
  - event counts – Basic Events, Gate Events
  - complexity
  - complexity
- Tree size (more effort for larger trees)
  - small (< 100 event)
  - medium (100 to 750 events)
  - large (750 to 2,000 events)
  - huge (>2,000 events)
- Conduct tree peer review
  - other FT experts
  - system designers

Rule #5

Rule #5 -- Know Your System

- Know the system design and operation
- Know the interfaces between subsystems
- Utilize all sources of design information
  - drawings, procedures, block diagrams, flow diagrams, FMEA's
  - stress analyses, failure reports, maintenance procedures
- Drawings and data must be current for current results
- Requires system engineering skills -- electronics, mechanics, software, etc.
- Make periodic checks to make sure the FT model is correct
  - reviews - peer , designers, customer
- The model and design data can be iterative
  - preliminary model progresses to detailed model

Rule #6

Rule #6 -- Understand Your Failure Data

- Failure data must be obtainable for quantitative evaluation
- Must understand failure modes, failure mechanisms and failure rates
- Data accuracy and trustworthiness must be known (confidence)
- Data estimates are useful and can be used, but results must be understood

Rule #7

Rule #7 -- Know Your Fault Tree Tools

- Know basic tool capabilities
  - construction, editing, plotting, reports, cut set evaluation
- Know tool user friendliness
  - intuitive operation
  - easy to use and remember
  - changes are easy
- Single vs. multi-phase tree
- Qualitative vs. quantitative evaluation
- Simulation vs. analytical evaluation (considerations include size, accuracy, phasing)

Rule #7 (continued)

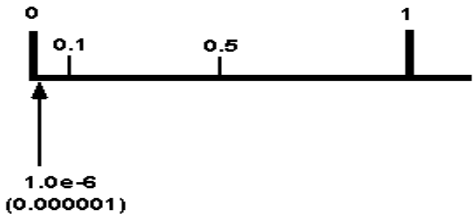
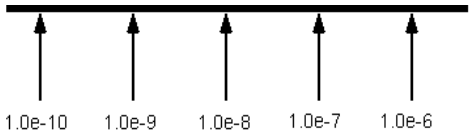
- Know tool limits
  - tree size
  - cut set size
  - plot size
- Understand cutoff methods, some can cause errors
- Gate probabilities could be incorrect when MOE's are involved

Rule #8

Rule #8 -- Understand (Appreciate) Small Numbers

- Failure rates and probabilities are between 0 and 1
- FT's generally deal with small numbers (< 1.0e-6)
- Small numbers are somewhat abstract
- The exponent size is of prime interest (e-6, e-15, e-35)
  - Decimal places are somewhat significant within the same range (1.11e-6 vs 1.97e-6)
  - Decimal places are not as significant for a wide range (1.1e-6 vs. 1.778e-9)
  - As numbers get very very small, decimal place are probably insignificant (ie, 1.0e-35 vs. 1.2e-35)

Rule #8 (continued)

	Probability range is between 0 and 1. FT events and cut sets also fall into this probability range. A number of 1.0e-6 looks very abstract on this chart. 1 failure per million hrs = 0.000001 = 1.0e-6
	Looking at a small number within a range of small numbers provides more valuable information.
What Are Small Numbers ?	

Rule #8 (continued)

Rule #8 -- Understand (Appreciate) Small Numbers

- Don’t get carried away with numbers
  - All results are essentially estimates for relative comparisons
  - is system 1.0e-3 or 1.0e-7 is relevant
  - is system 1.1e-6 or 8.7e-6 is not as relevant
  - is system 1.1e-6 or 1.123767e-6 is not relevant
- Remember, the model is *only* a model and does not have 100% fidelity to the true system, therefore, everything is somewhat relative

Rule #9

Rule #9 -- Understand Your Results

- Make reasonableness tests on the results
  - are the results correct
  - look for analysis errors (data, model, computer results)
  - are CS's credible and relevant, if not revise tree
  - take nothing for granted from the computer
  - test your results via hand calculations
- Verify that the FTA goals were achieved
  - are the results meaningful
  - was the analysis objective achieved
  - was the right tool used

Rule #9 (continued)

- Probability calculations are important, but nothing more than a mathematical exercise
- CS's are very important -- shows where to fix system, importance of specific events
- If exotic gates are used, check results, check assumptions
- Effect of MOEs is very important
  - they can cause large numerical impact or none at all
  - review carefully

Rule #10

Rule #10 – Remember FT's Are Models

- Remember that FT's are models
  - perception or model of reality
  - not 100% fidelity to exact truth
- Remember that models are approximations (generally)
  - not necessarily 100% exact
  - still a valuable predictor
  - Newton's law of gravity is an approximation
- Do not represent FTA results as an exact answer
  - use engineering judgment
  - small number are relative (2.0×10^{-8} is as good as 1.742135×10^{-8})
  - anything overlooked by the FTA skews the answer
    - ☞minor things left out can make results conservative (understate results)
    - ☞major things left out can be significant (overstate results)

Rule #11

Rule #11 -- Publish/Document Your Analysis And Results Completely

- Formally document and publish the entire FTA
  - may need to provide to customer (product)
  - may need to defend at a later date
  - may need to modify at a later date
  - may perform a similar analysis at a later date
  - may need records for an accident/incident investigation
- Even a small analysis should be documented for posterity

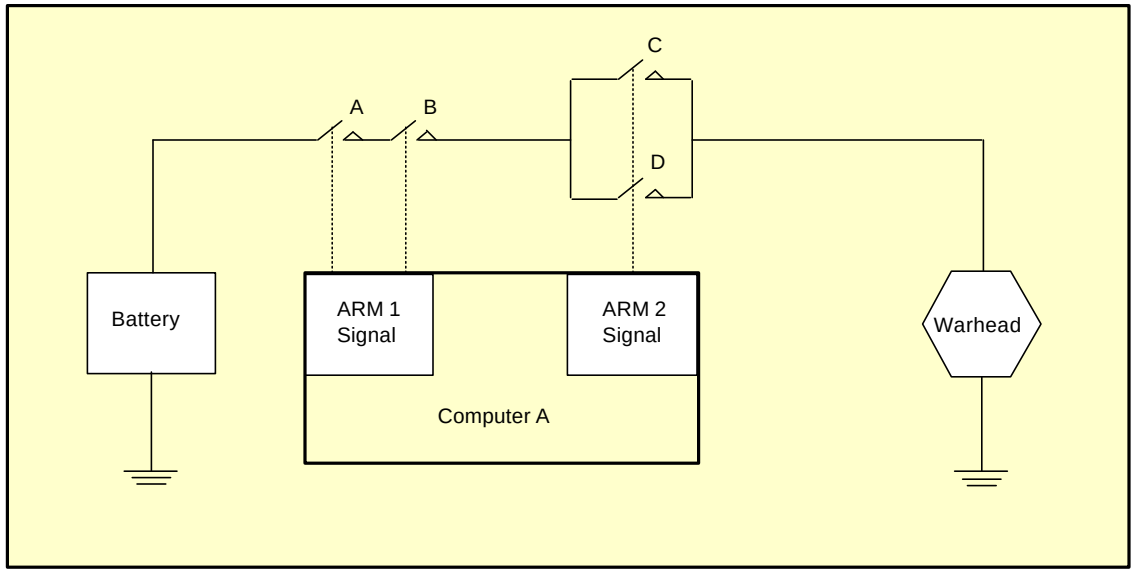
Rule #11 (continued)

- Provide complete documentation
  - problem statement
  - definitions
  - ground rules
  - references
  - comprehensive system description
  - data and sources (drawings, failure rates, etc.)
  - FT diagrams
  - tree metrics
  - FT computer tool description
  - results
  - conclusions

FTA Examples

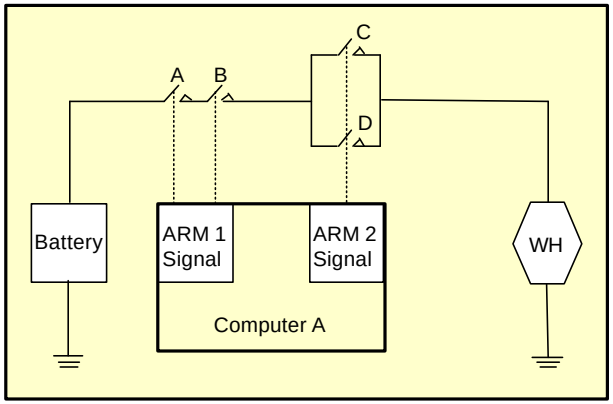
Problem #2

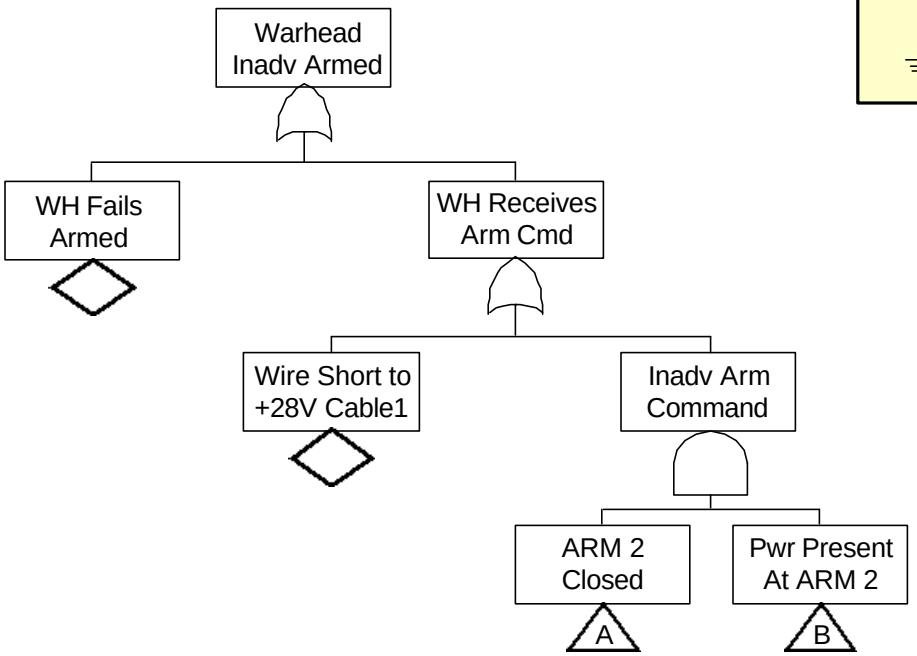
- Construct a FT for the following system
  - The Undesired Event is “Inadvertent Warhead Arming”
  - Construct the Fault Tree
  - Ground Rules:
    - ☞When all the switches are closed the Warhead receives the Arm command.



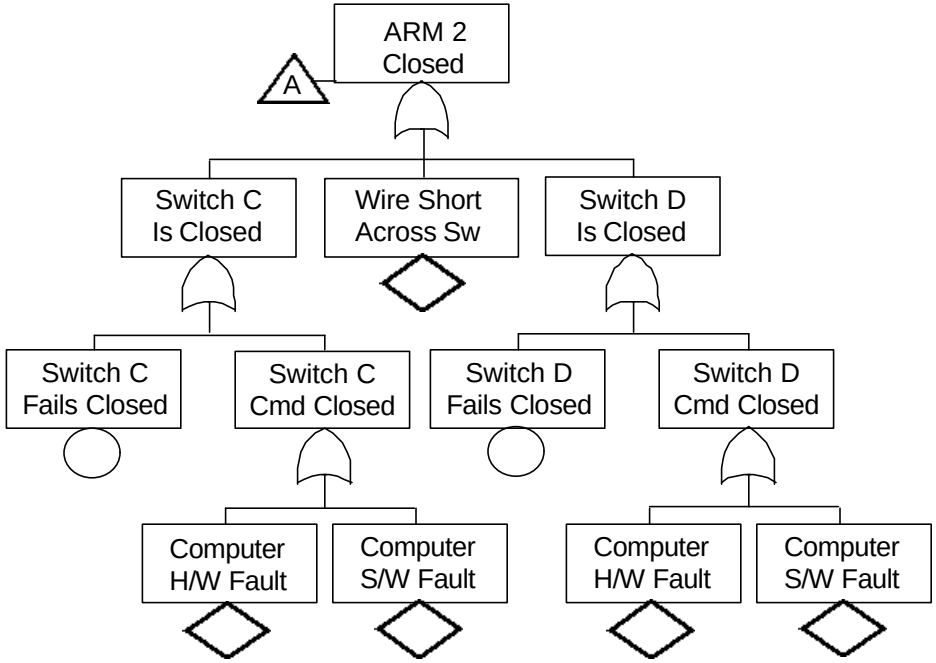
Problem 2 (cont'd)

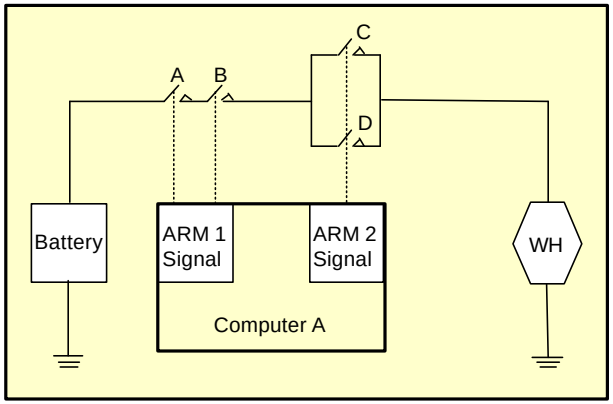
Method 1 – Structured
(Using Functional Approach)



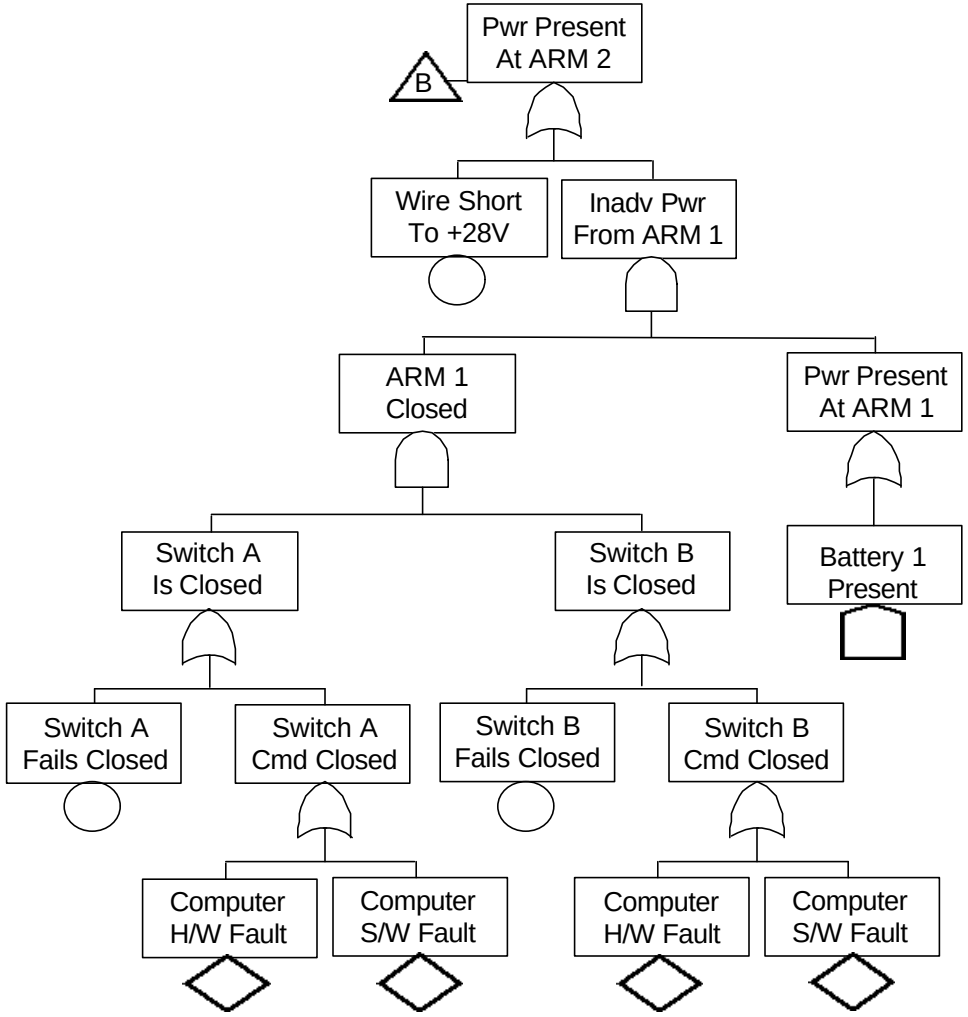


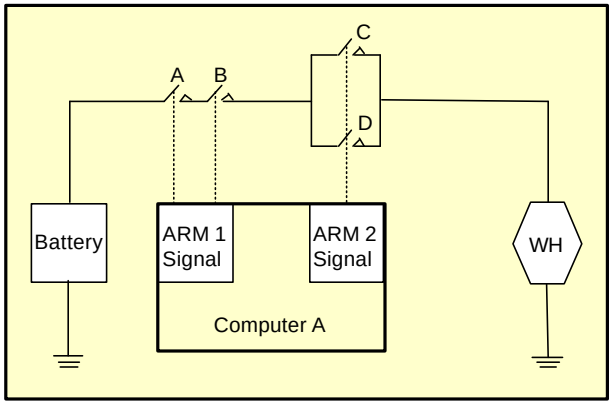
Problem 2 (cont'd)





Problem 2 (cont'd)





FTA References

Reference Books

- *Reliability And Fault Tree Analysis*, Conference On Reliability And Fault Tree Analysis; UC Berkeley; SIAM Pub, R. E. Barlow & J. B. Fussell & N. D. Singpurwalla, 1975.
- *Fault Tree Handbook*, NUREG-0492, 1981, N. H. Roberts, W. E. Vesely, D. F. Haasl & F. F. Goldberg, 1981.
- *Reliability and Risk Assessment*, Longman Scientific & Technical, 1993, J. D. Andrews & T. R. Moss, 1993.
- *Probabilistic Risk Assessment And Management For Engineers And Scientists*, IEEE Press (2nd edition), 1996, E. J. Henley & H. Kumamoto, 1996.

Web Sites

- www.system-safety.org
- www.FaultTree.net or www.fault-tree.net
- www.aot.com

3 Day Course

1.0 FTA Introduction
2.0 FTA History
3.0 FT Overview
4.0 FT Definitions
5.0 FT Construction – Methodology
6.0 FT Construction – FT Design
7.0 FT Mathematics
8.0 FT Evaluation
9.0 FT Validation
10.0 Multi-Phase FTs
11.0 Common FTA Pitfalls
12.0 FTA Rules
13.0 FT Software Codes
14.0 FTAB Operation
15.0 FT Repair
16.0 Other Tree Types (Success Tree, Event Tree, Casual Tree)
17.0 FT Dependent Events
18.0 Special Cases (standby, latency, spares)
19.0 Exposure Times
Appendix A – FT References
Appendix B – Exercises
Appendix C – Class Project
Appendix D – Example Fault Trees