



Quality Management in the Bosch Group

# 15. Fault Tree Analysis **FTA**



**BOSCH**

Invented for life





## Table of Contents

Register of figures .....	3
1. Preface.....	4
2. Introduction.....	5
2.1. Objectives of the FTA .....	5
2.2. History of the FTA.....	5
2.3. Benefits and drawbacks of the FTA.....	5
2.3.1. Benefits of the method .....	5
2.3.2. Drawbacks of the method .....	6
2.4. FTA application areas .....	6
3. Fundamentals of the FTA .....	7
3.1. Roles .....	7
3.2. “The 8 steps of the FTA” – an overview .....	8
3.3. FTA software at BOSCH .....	8
4. The Bosch approach to prepare a FTA .....	9
4.1. Step 0: Preparation including system analysis .....	9
4.1.1. General .....	9
4.1.2. Preventive / corrective FTA.....	10
4.2. Step 1: Definition of the undesirable event (Top Event).....	10
4.3. Step 2: Establish the criteria for the objective of the analysis .....	10
4.3.1. General .....	10
4.3.2. Preventive / corrective .....	11
4.4. Step 3: Construct the fault tree (qualitative description) .....	11
4.4.1. General .....	11
4.4.2. Symbols and modeling recommendations.....	11
4.4.3. Breakdown principles.....	12
4.5. Step 4: Qualitative interpretation .....	17
4.5.1. General .....	17
4.5.2. Fault combinations.....	18
4.6. Step 5: Determine the probability of occurrence of basic events (quantitative description) 29	
4.6.1. General .....	29
4.6.2. Preventive / corrective .....	30
4.7. Step 6: Quantitative interpretation.....	30
4.7.1. General .....	30
4.7.2. Definition of the computing parameters in the FTA tool.....	30
4.7.3. Numerical value of the Top Gates.....	31



## Fault Tree Analysis

4.7.4.	Identify optimization potential .....	32
4.8.	Step 7: Establish the need for action and success monitoring.....	36
4.9.	Step 8: Release and documentation of the FTA.....	37
5.	Literature on FTA.....	39
5.1.	Norms.....	39
5.2.	Standards.....	39
5.3.	Handbooks.....	39
5.4.	Reference books.....	40
6.	Glossary .....	41
7.	Attachment 1 Symbols and modeling recommendations.....	44
7.1.	Handling variants.....	44
7.2.	Modeling application boundary conditions: .....	45
7.3.	Special hints on fault tree construction for evidence per ISO 26262.....	46
7.4.	Modeling monitoring (monitors).....	47
7.5.	Overview of the event and gate types in the Tool FaultTree+.....	49
7.5.1.	Gate types available .....	49
7.5.2.	Available event type / event symbols .....	55
7.5.3.	Available fault models .....	56
7.5.4.	ISO 26262: Relationship of failure tolerance time – fault model consideration time (mission time) for continuous or initial monitoring .....	58
7.6.	Recommendations on the naming convention .....	60
7.6.1.	Naming Events / Gates .....	60
7.6.2.	Use event groups.....	62
7.6.3.	Special feature when naming gates .....	62
7.7.	Hints and tricks in the preparation, computation and handling of fault trees .....	64
7.7.1.	Multiple definition of a single basic event .....	64
7.8.	Application of NOT or XOR gates for activated function “Full Not Logic” .....	65
7.9.	Unintentional / intentional absorption of multiple point faults .....	68
7.10.	Use of cut-off rules for the computation .....	69
7.11.	Taking inputs into consideration that have no influence on a gate.....	72
7.12.	Open points in the FTA (=> Transfer Gates, Labels etc.) .....	73
7.13.	Modeling Common-Cause Failures .....	73
7.13.1.	Modeling with the $\beta$ -factor model .....	73
7.13.2.	Modeling by using the Root-Cause event .....	73
8.	Attachment 2 – example of a report.....	74



## Register of figures

Figure 4.1: Sensor substitute model for the sensor XXX.....	13
Figure 4.2: Fault types .....	15
Figure 4.3: Generate FTA / FMEDA interfaces .....	16
Figure 4.4: FTA / FMEDA interfaces - example.....	17
Figure 4.5: Example of a fault tree .....	18
Figure 4.6: Retain results activated exclusively for the Top Gate .....	19
Figure 4.7: Cut-Set list in FaultTree+ for the Top Event .....	20
Figure 4.8: Importance list in FaultTree+ .....	21
Figure 4.9: Monitored faults in FaultTree+ in the Cut-Set list.....	22
Figure 4.10: Unambiguous classification using BI: dotted - single point faults with a rare operating condition – interrupted line – monitored faults .....	22
Figure 4.11: Step 1) – determining potentially latent / non-latent paths.....	24
Figure 4.12: Step 2) – inheriting the initial partitioning on the directly associate FT elements (OR gates or events) .....	25
Figure 4.13: Step 3ff) – analysis of the lower-level AND gates .....	26
Figure 4.14 : Fault tree example.....	34
Figure 7.1: Handling variants for a supplementary branch.....	44
Figure 7.2: Handling variants of mutually excluding options.....	45
Figure 7.3: Modeling application boundary conditions .....	46
Figure 7.4: Modeling a monitoring system implemented in hardware .....	47
Figure 7.5: Modeling a monitoring system implemented in software.....	48
Figure 7.6: Naming convention in the event table.....	61
Figure 7.7: Event groups.....	62
Figure 7.8: Naming convention gate table .....	63
Figure 7.9: Absorption of wheel-speed faults .....	69
Figure 7.10: Project options – Cut-Offs .....	70
Figure 7.11: Example fault tree for demonstrating Cut-Offs .....	71



## 1. Preface

This document describes the Fault Tree Analysis method as can be applied in all company divisions. As a deductive method it is one way to meet the requirements of ISO 26262 on functional safety in the automobile industry and it is considered as a recognized standard.

The FTA can be used in a preventive or in a corrective manner. Systematic consideration of potential failures and the documentation of these with the means of the FTA help to describe failure mechanisms and derive relevant actions as well as documenting their effects. This contributes to the development of robust products and processes and in this way safeguards the company's successes as well.

The effectiveness of FTA depends on it being carried out at a good time, on the participation of skilled associates and on concentrating on the aspects that are relevant.

The FTA documentation and contents of this constitute together with other documents - like for example FMEA, drawings, manufacturing and test instructions – sensitive know-how and may only be forwarded under defined boundary conditions.

As a method of qualitative and quantitative risk analysis the FTA is included in engineering processes and manufacturing processes.



## 2. Introduction

### 2.1. Objectives of the FTA

The FTA serves in the first instance to detect and eliminate weaknesses as well as to make comparative studies.

With the help of this method the probability for the occurrence of a previously defined event (Top Event) as well as the corresponding causes shall be determined. The FTA approach here is deductive – from the effect to the cause (Top-Down approach).

The data acquired by the FTA makes possible, amongst others:

- Identification of causes and combinations of causes that lead to an undesirable event (Top Event).
- Computation of the probability of occurrence of the undesirable event or of the system availability (Boolean algebra).
- Identification of particularly critical events and combinations of events (fault paths).
- Identification of particularly effective improvement possibilities.
- Showing and documentation of the failure mechanisms and the functional relationships.
- Determining the characteristic values for demonstrating safety per ISO 26262.

### 2.2. History of the FTA

The first safety / risk analyses (USA 1950) were limited to investigating the different types of failures (Failure Modes) of components / assemblies of a system and the effects (Failure Effects) of the particular failure mode.

It soon became apparent however that an analysis of only the failure modes and follow-on failures was difficult to carry out because of the increasing complexity of the devices and systems. And also it turned out, that this method was not suitable for a quantitative reliability analysis.

Based on the knowledge of theory of reliability and the Boolean algebra engineers at Bell Telephone Laboratories (H. Watson, 1961) were able to show the abnormal behavior of control systems in a Boolean model with logic symbols. The FTA was born!

The FTA passed its first practical test at Boeing in the 1960s. In the years that followed it was adapted for use in aerospace and nuclear engineering. Later the chemicals, robotics and software industry started using the FTA for their safety analysis.

FTA has been refined further in recent years and has since become a widely used analysis method for assessing the safety and reliability of large and complex systems from the technical viewpoint.

The ISO 26262 introduced in automotive engineering in November 2011 prescribes the use of deductive methods like e.g. FTA.

### 2.3. Benefits and drawbacks of the FTA

The success of the FTA or the value of the analytical results determined in this way depends to a large degree on the external boundary conditions.

The analysis by means of a fault tree Fault status...

- Needs a qualified moderator that methodically guides the team.
- Requires a high level of discipline in preparing the fault tree to prevent errors.
- Requires a separate subtree / branch for each undesirable event.

#### 2.3.1. Benefits of the method

The analysis by means of a fault tree...



## Fault Tree Analysis

- Systematically gives the logic path beginning with the undesirable event, that is to say from a certain effect, back to the actual cause and documents this path in a graphical and easily comprehensible form.
- Computes the probabilities on the basis of Boolean algebra.
- Allows the identification of cause-effect relationships not detected up to now.

The result of the analysis makes it possible...

- To make quantitative and qualitative statements on system parameters like e.g. availability, reliability, failure probability etc. This is of particular value for large and complex systems.
- To consider multiple events as causes as well.

The analysis can...

- Handle parallel, redundant and alternative failure or event paths.
- Handle any kind of technical and non-technical system.
- Show the significance of fault causes for the undesirable event.

### 2.3.2. Drawbacks of the method

The analysis by means of a fault tree...

- Only establishes the relationship between the causes found and the analyzed main event (Top Event). These causes can however lead to other effects not yet shown.
- Can only model the time behavior of systems and dynamic processes with difficulty.
- Is not always exactly quantifiable since the complete data basis for the basic causes is not always given.
- Is an elaborate method when a complex system shall be quantitatively analyzed. It is therefore usually used only for a selection of few relevant undesirable events.

## 2.4. FTA application areas

The application areas of the FTA can be summarized briefly by the term “RAMS”.

To be understood here:

- **R**eliability
- **A**vailability
- **M**aintainability
- **S**afety

Principally there are two areas of focus when applying the FTA:

- Preventive approach
- Corrective approach

With the preventive application of the FTA the focus is on the hazard or risk analysis.

Objectives of preventive use of the FTA:

- Minimize design errors (malfunction, non-function),
- Verify and demonstrate the system safety,
- Increase the system reliability and
- Assess potential risks within the scope of risk management.

The corrective approach of the FTA is used in the damage analysis or risk analysis.

In case of analyzing damages the FTA serves as decision-making assistance for legal issues (e.g. product liability) and also in determining the risks for the risk-management process.

The results from the analysis are used to assess a damage or accident sequence.



### 3. Fundamentals of the FTA

#### 3.1. Roles

##### Active roles in preparing the FTA

###### FTA coordinator

The *FTA coordinator* is the representative of the method for a unit (e.g. company division, product unit, ...). This associate designs the application process of the method in that unit and is responsible for empowering the *FTA experts* and members of the *FTA team*. The associate assumes the following tasks here: knowledge multiplier, coach, specialist contact person, methodic decision-maker. The *FTA coordinator* should always be a FTA expert as well.

###### FTA expert

The *FTA expert* is responsible for the moderation on the *FTA team*. This associate prepares fault trees in collaboration with the team members and is responsible for ensuring that the method and the relevant presentation media are used correctly. The FTA expert ideally has product knowledge as well so as to be able to take an active part in the discussions.

Besides the tasks associated with preparing the FTA, the *FTA expert* also has other tasks: e.g. coaching members of the *FTA team* on the method and the tool, preparing interpretations and reports, providing support in the evaluation the results, presentation of the results.

###### FTA team

The *FTA team* is made up of the *FTA experts* and the relevant specialists. When a FTA is performed within the scope of a project, the participation of a technical project associate (e.g. Project Safety Manager) is necessary. The *FTA team* is responsible for establishing the contents of the FTA, that is to say to depict the design. In the case of a quantitative FTA additionally failure rates have to be determined here.

##### Passive roles in the preparation of the FTA

###### FTA contractor

The *FTA contractor* defines the goals and the scope of the considerations for the FTA to be conducted. The contractor is the "Sponsor" of the FTA and ensures the budget and the resources for the work. Based on the results of the analysis the contractor decides on the further course of action, e.g. introduction of technical adjustments, negotiations with the customer (accepting risks).

###### FTA reviewer

The *FTA reviewer* is responsible for reviewing the contents of a FTA before it is released.

###### FTA assessor

The *FTA assessor* is responsible for the staging assessments to establish the maturity of the implementation of the method and to support the further-development of the organization.

###### Other roles

- Customer (OEM / Line/ ...)
- Quality management
- Supplier(s)
- etc.



### 3.2. “The 8 steps of the FTA” – an overview

A fault tree analysis can be broken down into following work steps:



- 0 Preparation including system analysis
- 1 Definition of the undesirable event (Top Event)
- 2 Establish the criteria for the objective of the analysis
- 3 Prepare the fault tree (qualitative description)
- 4 Qualitative interpretation
- 5 Determine the probability of occurrence of the basic events (quantitative description)
- 6 Quantitative interpretation
- 7 Establish the need for action, the actions, and success monitoring
- 8 Documentation

### 3.3. FTA software at BOSCH

At Bosch the preferred solution for the processing and documentation of FTAs is by using the program FT+ from Isograph.

At the present time: FT+, floating license with central administration.



## 4. The Bosch approach to prepare a FTA

Hints and examples on implementing the method of FTA

### 4.1. Step 0: Preparation including system analysis

#### 4.1.1. General

In the preparation of an FTA both the contents as well as formal and organizational aspects have to be established.

The contents to be established include:

- Description of the assignment (if possible: Formulation of the Top Events)
- Scope of considerations (system with its limits, interfaces and boundary conditions)  
Included in the boundary conditions are e.g.
  - Probabilities of occurrence (e.g. from other FTAs / FMEDAs) from input signals (important for functions that are distributed throughout several systems)
  - Consideration of disturbances from the environment
  - Agreement on the allocation of safety mechanisms at the interfaces (no multiple use that might lead to falsification)
  - Information about the higher-level system (e.g. vehicle control unit) needed for preparation of the FTA
- Specification of failure modes to be considered: Systematic failures and / or random (hardware) failures
- Type and extent of the analysis:
  - Qualitative: Analysis of the cause-effects relationships
  - Quantitative: Computation of the probability of occurrence of events and additional quantitative parameters. For a quantitative analysis it is necessary to specify the data basis and the service / operating conditions.
- When information from the customer is needed for preparation of the FTA, the availability of this information must be ensured. The consequences of such information not being provided shall be indicated.
- Requirements on the FTA method:  
The DIN EN 61025 serves as the standards base for the FTAs at Bosch. If the customer (e.g. OEM) has called for a FTA per a different norm or guidelines for the method, then this has to be agreed on a project-specific basis. It shall be recorded in writing that the customer assumes the responsibility for the consequences from this demand.

The above-referenced determinations and other assumptions / boundary conditions shall be documented and acknowledged by the customer in writing.

Included in the formal determinations are e.g.:

- Customer, FTA team, payment, acceptance criteria, time frame
- Work products: Report, fault tree file, special analyses
- Documentation, release / handover to the customer, archiving
- For FTAs that are for customer projects:
  - Clarification of the type of presentation and the time plan for this
  - Clarification of which documents can / shall be handed over to external customers (observe:[https://rb-wam.bosch.com/socos-c/SOCOS/finder.cgi?CD-03743-000\\_XXX\\_X\\_EN](https://rb-wam.bosch.com/socos-c/SOCOS/finder.cgi?CD-03743-000_XXX_X_EN))



#### 4.1.2. Preventive / corrective FTA

##### Preventive FTA

Prerequisite for beginning an FTA is the at least preliminary description of the item under consideration ("System"). Of this preliminary description, an error-free function has to be available in a documented form ("System analysis"). This description can be very detailed or more abstract depending on the analysis goal and the scope of considerations.

Typical sources of data are: Requirements specifications, function diagrams, effect chains, system architecture models, behavior / data models, block diagrams, design drawings, flow diagrams, process descriptions, ...

Ideally at the start of an FTA there are already descriptions available from other risk analyses and failure descriptions, e.g.: PHA, FHA, FMEA, FMEDA, DRBFM, QFD, 8D reports.

For analyses of the functional safety, other sources of data should be available at the beginning of the FTA: Hazard and risk analysis, safety goals, safety concept including an overview of the safety mechanisms, system design and definition of the safe states.

##### Corrective FTA

The corrective FTA analyzes the events that have actually occurred. At the beginning all relevant information about the event has to be compiled, e.g.:

- Failure description
- 8D report
- Interpretation of fault-code memories
- Design descriptions of the object, e.g. TCD
- Protocols of the statements from the user / discoverer of the event
- Objects with discrepancies

#### 4.2. Step 1: Definition of the undesirable event (Top Event)

The undesirable event (often also referred to as "Top Event") is the failure or the malfunction to be considered in the system to be investigated. The abstraction level where the event is defined is of no significance here. A malfunction at the vehicle level, discrepant control unit function, a wrong sensor signal or the failure of a circuit group can be considered.

Attention shall be paid in the description of the undesirable event that both the event itself as well as the boundary conditions valid for this are unambiguously defined. The significance of the entire FTA depends crucially on this description.

In the *preventive* application case of the FTA the definition of the undesirable event is based on either

- the non-fulfillment of functions or requirements (e.g. requirements specifications)
- or
- undesirable events from earlier investigations (e.g. FMEA, FHA, PHA, G&R or H&R).

In the *corrective* approach a failure that has actually occurred or a malfunction of the system is defined as the undesirable event.

#### 4.3. Step 2: Establish the criteria for the objective of the analysis

##### 4.3.1. General

Once the undesirable event (Top Event) has been defined, it is established in this step which goals shall be pursued by conducting the FTA. The FTA provides information whether the defined goals have already been reached. If this is not the case then it can be shown which possibilities to modify the product are there to fulfill the defined requirements.



## Fault Tree Analysis

In defining the goal the differentiation is made as a rule between qualitative and quantitative goals.

Examples for *qualitative* goals are:

- No Single Point failure acceptable, i.e. no single event may lead to the Top Event
- Show all cut-Sets
- Show all “Critical path” (with diagnostics)

Examples for *quantitative* objective criteria are:

- Probability of occurrence, Q, for the Top Event
- Probabilities of occurrence and sequence of the Minimal Cut-Sets
- Show the “Critical path” (isolated event or probability of occurrence)
- Compliance with the safety goals (e.g. failure probability  $< 10^{-8}$ )
- Identify the several importances (e.g. Birnbaum importance)

Valid norms, guidelines from industry associations (e.g. VDA), customer requirements and company-internal requirements are decisive for the purpose of the FTA.

Examples for the objectives having their origin in the respective documents are:

- Failure rate for “Random HW failures” according to ISO 26262 (“Functional safety of road vehicles”)
- Meeting “State-of-the-art” and “Highest priority for preventing failures” per CDQ0214 (“Requirements on product safety”)
- Safety against single failures according to the “E-Gas monitoring concept” or “3-level concept” of the ETC work group

### 4.3.2. Preventive / corrective

Principally both application forms of the FTA aim for similar objectives. It applies to show the structural build of a design and thereby detect the possible critical causes that can lead to or have led to the undesirable event.

It is objective in both applications to highlight concrete possibilities to improve the design and show, where appropriate, the influence on the probability of occurrence.

Unlike the *preventive* FTA with the *corrective* FTA there is a true discrepancy (undesirable event) already available.

## 4.4. Step 3: Construct the fault tree (qualitative description)

### 4.4.1. General

The structure of a FTA and its build depend greatly on the undesirable event (Top Event) to be analyzed and also on the architecture of the object to be analyzed.

Generally valid is the deductive approach (top-down): The start is at the undesirable event. The tree is then constructed over several levels down to the smallest unit of consideration (Basic Event).

The structure of the tree is oriented on the failure structure or the failure net of the object to be considered. A functions analysis already conducted during the preparation phase is very helpful for the preparation of the fault tree. The starting information for the construction of a fault tree can be e.g. the FMEA and the DRBFM.

### 4.4.2. Symbols and modeling recommendations

The types of symbols used for preparing an FTA are defined in “Attachment 1 Symbols and modeling recommendations”:

- Event and gate symbols
- Recommendations for naming conventions
- Fault models



- Handling variants
- Modeling application boundary conditions
- Modeling monitoring
- Modeling “Common Causes”

### 4.4.3. Breakdown principles

The partitioning is made e.g. according to

1. Effect chains (e.g. actuator – control unit – sensor)
2. Breakdown according to physical domains (e.g. subtrees for the mechanical subsystem, electronic subsystem, hydraulic subsystem,...), that are then linked under one Top Event with one gate
3. Breakdown according to integration levels (e.g. whole system – function level – component level)

A modularization of the FTA is recommended for very large objects and for objects where the analysis is very complex. The advantage here is that the total number of gates and events in the system the FTA is kept as small as possible and hence the computing times can be reduced as well.

Then the Top Events of the lower-level FTAs (e.g. the FTA of a sensor) are the basic events of the higher-level FTA (e.g. FTA of a system that includes this sensor). These events are referred to as interface elements in the following. Together the elements constitute a kind of FTA substitute model.

If the interface element is in an AND relationship with other components, then possible common fault causes (Common Cause faults) shall be considered in particular. For that purpose the evidence of possible common causes with other components can be required from the components authority, e.g. for the definition of the interface elements.

#### **Example 1: Sensor substitute model**

In the following an example of the determination of interface elements is shown. As an example a sensor that shall be in an AND relationship with another sensor (plausibility sensor) is taken.

The basic idea of this modeling is to make a differentiation according to single point faults (monitored and dormant), multiple faults and Common Cause influences possible at the level of the system FTA for the purpose of analysis. It is made possible in this way to model the **exact** contributions from all sensors to the violation of the safety goal at system level, including the contribution from Common Causes from the system level on the sensors.

The methodical limitation is that all possible Common Causes have to be determined beforehand in a qualitative analysis of the Common Cause faults. In order to have later all basic events of the FTA as disjunctive events. In the example it is assumed that there are two types of Common Cause faults: Firstly the supply voltage (SUPP) from the higher-level system and secondly electromagnetic interference from external sources of disturbance within or without the specified range (EXT EMI IN or EXT EMI OUT).

The fault model of the sensor is thereby broken down as follows:

- Not-monitored isolated failures (Single Point Faults (“SENS SPF”) that have no monitoring whatsoever)
- Single point faults with a monitoring outside of the failure tolerance time (“SENS\_SPF DORM”)
- Multiple point faults (“SENS CIRC THR”) whereby both types of double point faults are included here (faults of the function & failure of the associate monitoring as well as faults of function 1 & faults of function 2), as well as faults of a higher order.

(Symbols on the basic events and the logical gates of fault tree are explained in Attachment 1.)



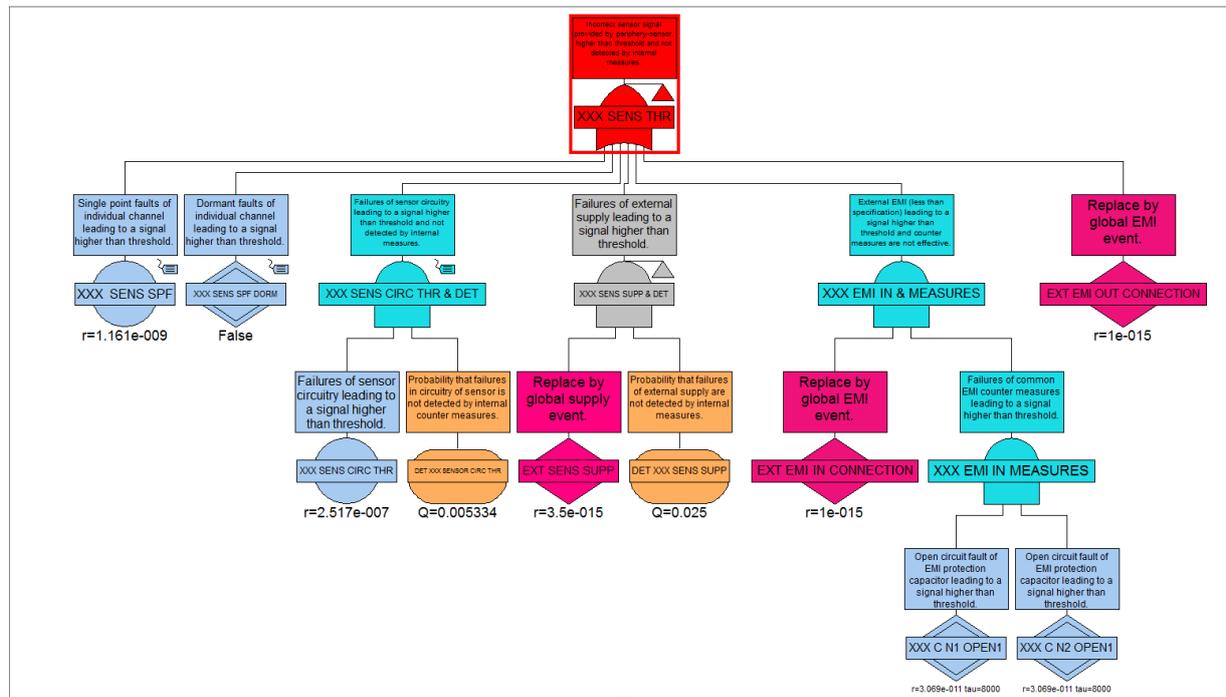


Figure 4.1: Sensor substitute model for the sensor XXX

Determine the interface elements (the inputs of the OR gate are referred to as elements in the following):

**Element 1 = Basic event XXX SENS SPF**

Determine the single point faults (XXX SENSOR SPF)

In the underlying sensor FTA a Cut-Set analysis is performed for the Top Event to be investigated. The failure rate for the single point faults is given from the sum of all Cut-Sets of the 1st order (true single point faults). The computations are typically executed after exporting the Cut-Sets into an Excel file.

**Element 2 = Basic event XXX SENS SPF DORM**

The failure rate is given here from the sum of all failures with a monitoring outside of the failure tolerance time. In this example there was none and therefore the Basic event appears in the fault tree without any numerical value.

**Element 3 = gate XXX SENSOR CIRC THR & DET**

The Gate SENSOR CIRC THR & DET stands for all multiple point faults and those failures with monitoring within the fault tolerance time. To determine the failure rates a Cut-Set analysis is again executed in Excel. Prerequisite is the compliance with a naming convention that allows effective filtering in Excel.

First of all the isolated-failure and the Common Cause relevant Cut-Sets with EMI and with Power Supply are filtered out. The failure rate for the “XXX SENSOR CIRC THR & DET” is given from the sum of the remaining Cut-Sets. The failure rate for the “XXX SENSOR CIRC THRES” is given from the sum of the failure rates of the associated basic events (function failures without monitoring). The gap in the monitoring “DET SENSOR CIRC THRES” can then be calculated as the quotient of the failure rates XXX SENSOR CIRC THR & DET / XXX SENSOR CIRC TRHES. Detailed explanatory notes on the quantitative interpretation of fault trees can be found in Section 4.5 on Step 4.

**Element 4 = gate XXX SENSOR SUPP & DET**

The event EXT SENSOR SUPP is identified here with an event of the system FTA. It is an example for modeling a circuit external to the sensor (hence the naming convention “EXT”) yet internal power supply for the system circuit. This power supply circuit can show a *Common Cause* for the other connected sensors and ICs.



The gap in the monitoring DET XXX SENS SUPP is determined in a manner similar to element 3.

#### **Element 5 = gate XXX EMI IN & MEASURES**

The modeling of an EMI fault constituting a Common Cause for the sensors is shown in a similar manner by the AND relationship of a basic event EXT EMI IN CONNECTION with the failure rate of the EMI protective circuit (AND relationship of the failures of two protective capacitances).

#### **Element 6 = gate XXX EMI OUT CONNECTION**

For this external fault it is assumed that it is outside the limit of the specification (naming convention EMI OUT) and therefore that the implemented protective measures do not act (any longer). The modeling is a rate action for the **Fehler! Verweisquelle konnte nicht gefunden werden.** or the failure rate of the external event.

This example is from CC-PS/EPH. More detailed information on modeling methods can be provided.

#### **Example 2: Transition from FTA to FMEDA**

When components of a system are analyzed by FMEDA (Failure Mode Effect and Diagnosis Analysis) a clearly defined transition between the two methods is necessary. The FTA defines the safety requirement, the fulfillment of which shall be confirmed by the results of the FMEDA. To incorporate the FMEDA in a FTA the following failure modes shall be determined by the FMEDA.

##### *Single point fault:*

A single point fault is a fault that is not covered by any safety mechanism and leads directly to violation of the safety goal.

FMEDA result: Sum of the failure rates of all single point faults:  $\lambda_{SPF}$  = Lambda Single Point Faults

##### *Residual fault:*

The residual fault is that proportion of a fault that is not recognized by monitoring and that leads to violation of the safety goal.

FMEDA result: Sum of the failure rates of all residual faults of monitored faults:  $\lambda_{RF}$  = Lambda Residual Faults (RF)

##### *Latent fault:*

A latent fault is one of the causes of a multiple point fault that alone however does not lead to violation of the safety goal and will not be detected within its latency period.

FMEDA Result: Sum of the failure rates of all latent (multiple-point) faults:  $\lambda_{MPF,L}$  = Lambda Multiple Point Faults Latent

This is the sum of the failure rates of all faults that only violate the safety goal in the multiple point faults case and for which no action against the latency has been implemented (or the gap of such actions against latency).

##### *Total failure rate:*

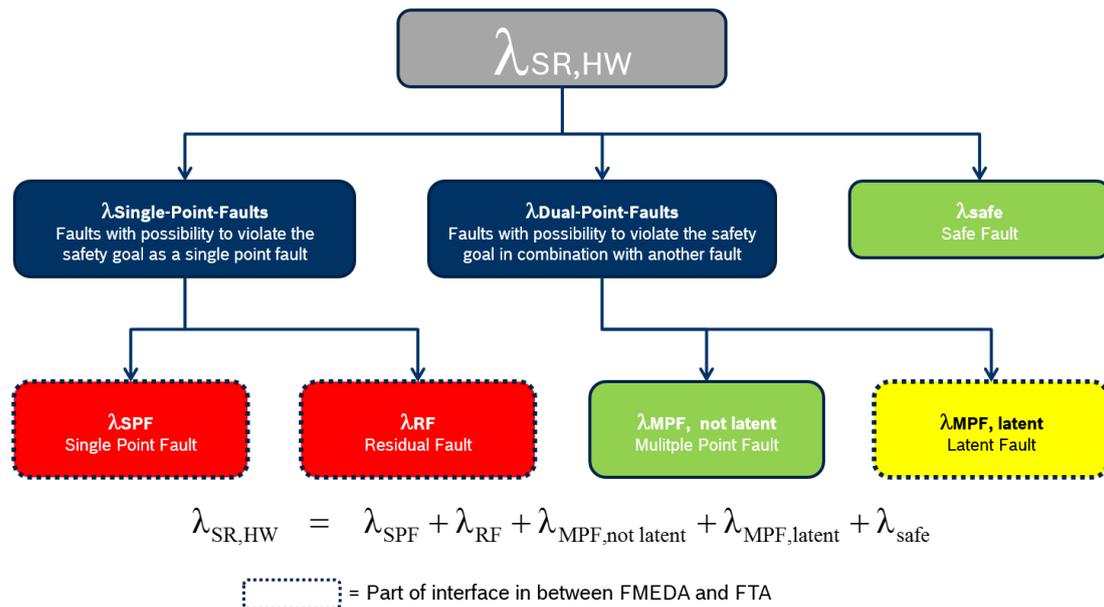
For correct consideration of the multiple point fault probability the sum of the total failure rate of all elements of relevance for the safety goal is necessary:

FMEDA-Result:  $\lambda_{SR,HW}$  = Lambda Safety Related Hardware Elements or  $\lambda_{SRHE}$  = Lambda Related Hardware Elements

This is the sum of all failure rates of components that can have an influence on the safety goal in one of the ways mentioned above. The failure rates before taking Redundancy into consideration (=> leads to multiple point faults) or monitoring (=> leads to residual faults) are summed.



The following diagram describes the relationships between the mentioned fault classes.



**Figure 4.2: Fault types**

The safety requirement to be considered is handled as a gate in the FTA. The corresponding failure rate ( $\lambda_{SR,HW}$ ) is calculated from the failure rates determined from the FMEDA according to its significance for the system.

Determined single point and residual fault failure rates influence the fault tree directly. A single point fault of the FMEDA does not necessarily have to remain a single point fault in the FTA as such, but rather can also become part of a multiple point fault by the system architecture depicted in the FTA.

A multiple point fault determined in the FMEDA has to remain a multiple point fault in the FTA because otherwise in the FTA it is visible that this fault already concerns a multiple point fault at the components level. The FTA / FMEDA interface must take this into account. Since the FMEDA cannot show in detail by methodic limits the fault associated with each fault declared as a multiple point fault, a conservative modeling on interface level has to be realized for the resulting probability of multiple point faults. This is realized by an AND gate in the interface that combines the failure rate of the undetected multiple point faults with the failure rate of all safety-relevant elements. It can be assumed here that the probability of the "real" second fault of the multiple point faults is lower than the sum of the failure rate of all safety-relevant elements; i.e. is the product probability as a conservative calculation. Attention shall thereby be paid that the undetected multiple point fault must be modeled as a "latent" fault with the appropriate latency period.

Figure 3 and 4 show how the FTA / FMEDA interfaces have been realized as fault tree logic.



# Fault Tree Analysis

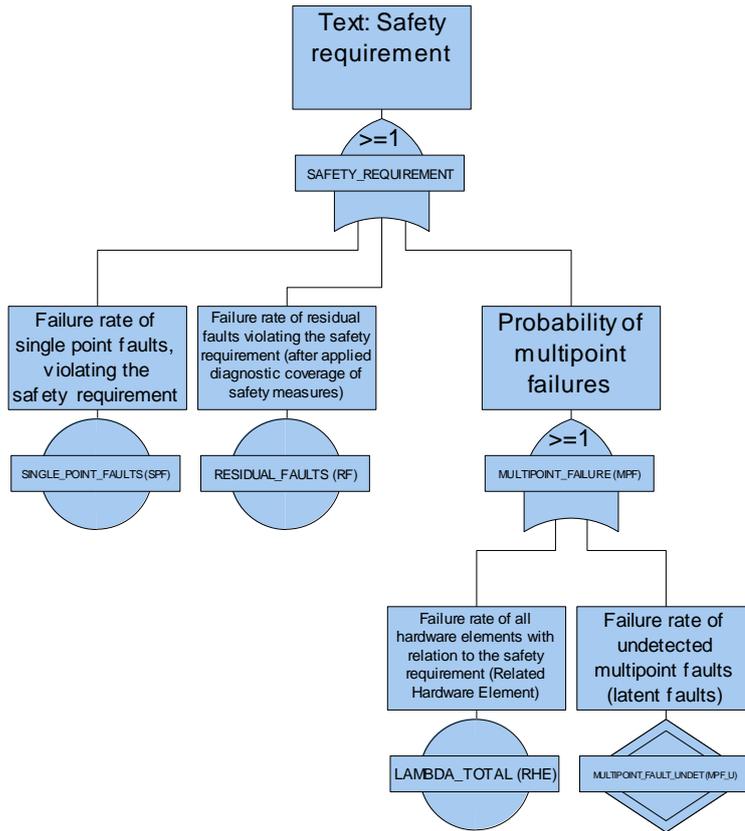


Figure 4.3: Generate FTA / FMEDA interfaces



## Fault Tree Analysis

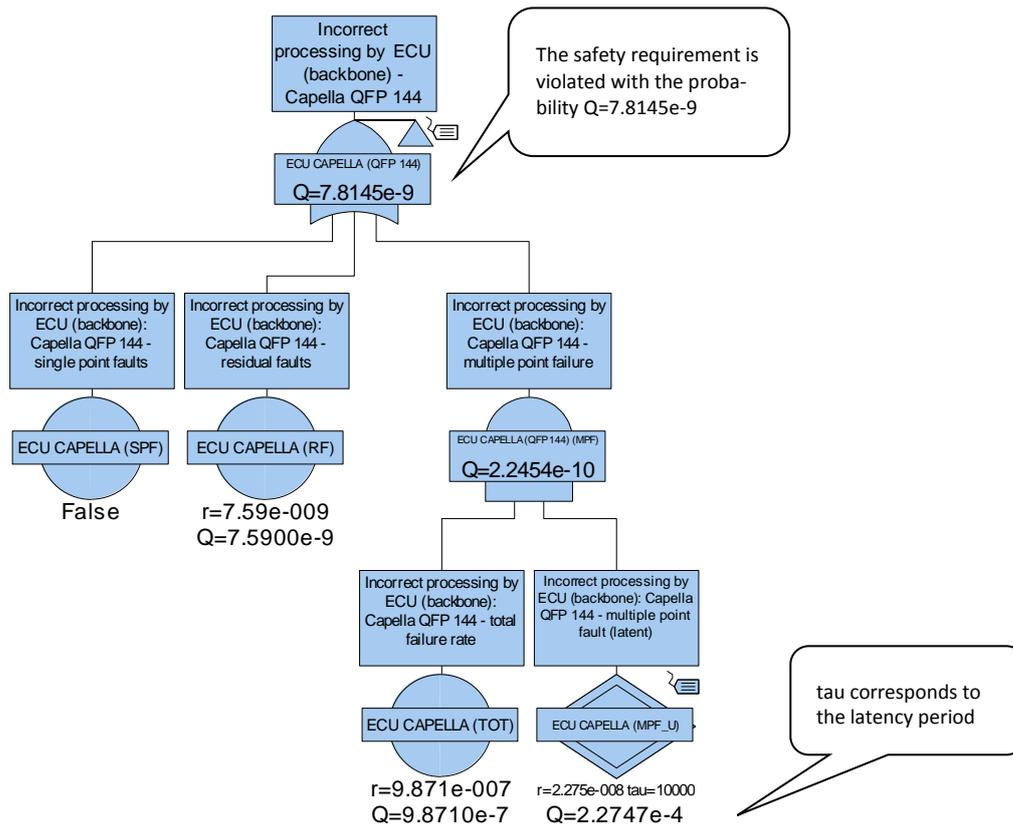


Figure 4.4: FTA / FMEDA interfaces – example

## 4.5. Step 4: Qualitative interpretation

### 4.5.1. General

The qualitative results of an FTA comprise

- Fault tree depiction
- Fault combinations (minimal cuts or cut-sets)
- Importance (Birnbbaum) for the events linked by logic in the FTA

The interpretations regarding fault combinations and importance are always made with respect to a selected undesirable event (Top Event). This Top Event is represented by a logic gate below where the actual fault tree begins.

The evaluation examples shown in the following sections have been created with the following fault tree.



# Fault Tree Analysis

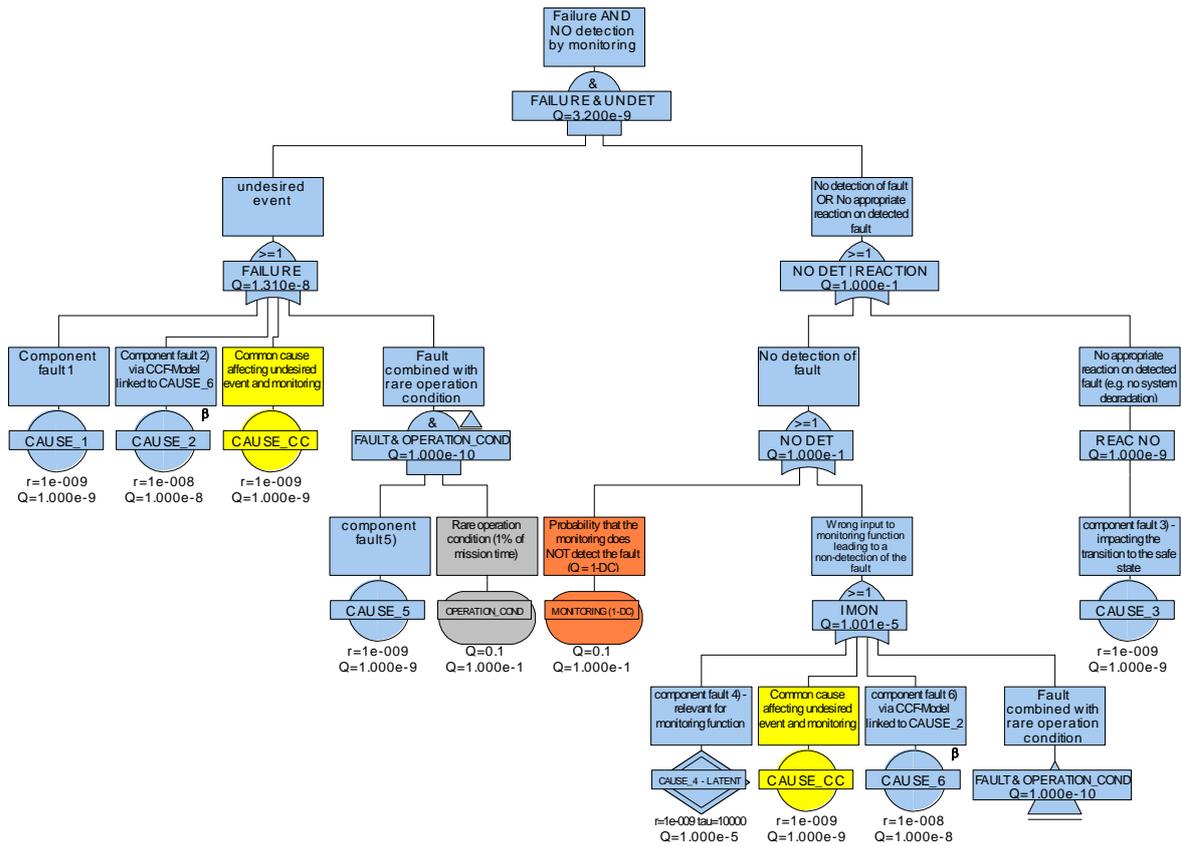


Figure 4.5: Example of a fault tree

The example shows monitored fault causes.

CAUSE\_1, CAUSE\_2, CAUSE\_5 as well as CAUSE\_CC constitute the faults to be monitored whereby CAUSE\_5 is only effective in combination with rare operating conditions. Both the Gate “FAULT & OPERATING CONDITION” as well as the CAUSE\_CC are also linked here at the input of the monitoring (Gate “I MON”) – it is therefore to be expected that the AND gate “FAULT & NOT DETECTED” is lifted for these faults.

CAUSE\_4\_LATENT and CAUSE\_6 constitute faults where the monitoring mechanism fails when these occur. If CAUSE\_3 occurs then there is no appropriate reaction (e.g. system degradation) triggered for a known fault (shown here in the gate “NO SYS-DEG”).

Furthermore, as a special feature, CAUSE\_2 is linked with CAUSE\_6 by means of a Common Cause fault model (CCF model) (character  $\beta$ ). The CCF model describes the proportion of the total probability with the two faults (CAUSE\_2 and CAUSE\_6) occurring *at the same time*. Remark: The Computation results of the beta model can be influenced in the project options “Sets Generation” of FaultTree+ (CCF analysis area).

Note: If a FTA shall be prepared for purely qualitative considerations then it is still meaningful to assign failure rates to the basic events. To avoid possible misunderstandings in any presentations it is recommended to use a generically created fault model that is then assigned to all events. Effect: The computed fault combinations are outputted by the tool sorted according to their probability. The displaying of the resulting probabilities of the gates can be suppressed in Tool options “View”.

## 4.5.2. Fault combinations

Different types of fault combinations can be identified. These are single point faults, residual faults (monitored faults), multiple point faults and as a sub- quantity of the multiple point faults, the group of latent multiple point faults.

2020-04-06 - SOCCOS



**a) Single point faults**

One objective of the FTA is to identify those single point faults that lead to an undesirable event.

A single point fault is characterized as follows:

- There are no safety mechanisms (e.g. monitoring or redundancy) that prevent the associated undesirable event (Top Event) from occurring when the fault occurs.

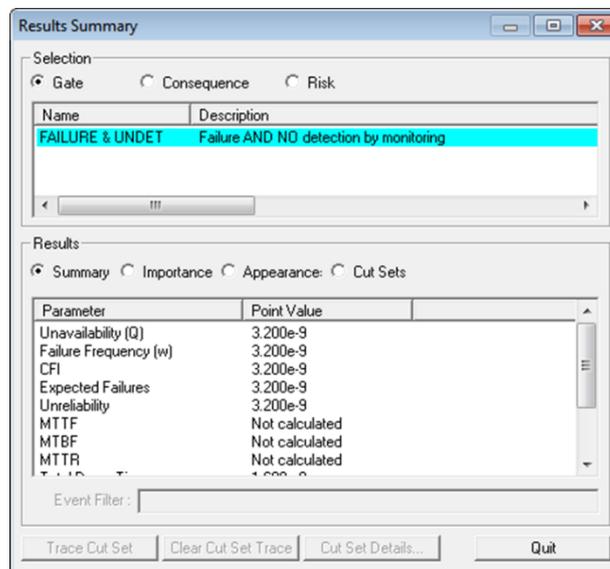
**Interpretation in the FT tool:**

The FT tool essentially offers two possibilities for the interpretation of fault combinations. These are:

- The Cut-Set list which arranges the fault combinations (minimal cuts) computed by the tool according to the probability of their occurrence, and
- The importance list that shows the significance (importance) for each basic event included in the fault tree. The Birnbaum importance can be taken for a qualitative interpretation.

Compliance with a naming convention for the designations of the basic events (Event Names) is indispensable to be able to efficiently evaluate the lists offered by the FT tool. This enables a quick identification of the basic events involved on the basis of their name. Even in the example above (see Figure 4.5) there is a naming convention in a rudimentary form (e.g. all faults are termed "CAUSE\_#"). For notes and recommendations concerning this topic refer also to "Attachment 1 Symbols and modeling recommendations".

Remark: Before beginning with the interpretation it must be ensured that for the gate of interest the saving of the fault combinations has been activated. This is realized in the FT tools used at Bosch by activation of the Gate option "Retain Results". Otherwise the gate of interest will not be included in the list of gates available. Figure 4.6 shows the evaluation dialog when in the fault tree example the option "Retain Results" is only activated for the Top Gate. In the upper area only 1 gate is offered (unlike in Figure 4.7 where several gates can be selected for analysis).

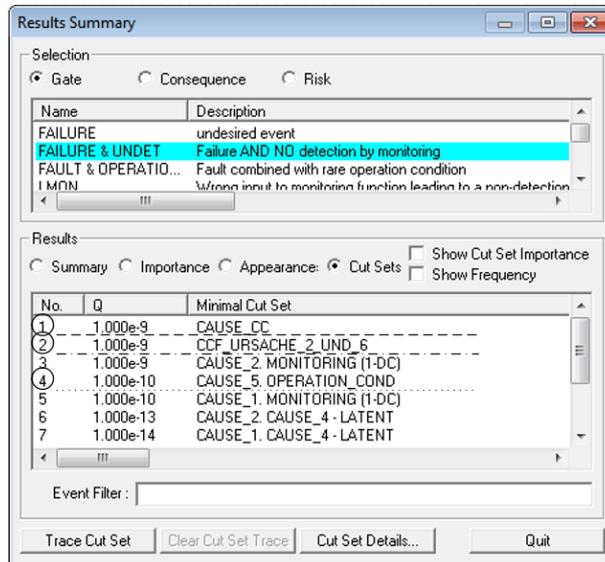


**Figure 4.6: Retain results activated exclusively for the Top Gate**



Single point fault in the Cut-Set list:

Single point faults can now be identified in the Cut-Set list of the respective Top Events:



**Figure 4.7: Cut-Set list in FaultTree+ for the Top Event**

- Cut-Sets of the 1st order  
 → the number of events involved is 1 (Cut-Set numbers 1 and 2 in Figure 4.7)  
 CAUSE\_CC is the modeled Common Cause fault that is attached both to faults as well as to the monitoring side.  
 In the example: “CCF\_CAUSE\_2\_UND\_6” is not represented in the fault tree by a basic event. The tool moreover computes on the basis of the CCF model used the probability of the faults from CAUSE\_2 and CAUSE\_6 at the same time and shows this value as a single point fault entry (with the name of the CCF model) in the Cut-Set list. By tracing this Cut-Set using the FT tool back to its origins (using the option Trace Cut-Set) the “Trace” ends at CAUSE\_2 respectively CAUSE\_6.
- Cut-Sets order > 1  
 → Max. 1 event constitutes the fault - the remaining events involved represent probability-reducing conditions (e.g. rare operating conditions – that can be modeled as “Conditional Events” with a fixed probability of occurrence) (Cut-Set number 4 in Figure 4.7)  
 In the example: “CAUSE\_5” combined with the “OPERATING CONDITION” (this event describes neither a fault nor a monitoring mechanism)

Single point fault in the importance list:

In the importance list of the considered Top Event the Birnbaum importance (BI) can be used to identify single point faults.

For computation of the BI it is irrelevant *which* data assignment is allocated to the particular single point fault event; important is that failure rates have been defined at all. In the case of a missing or incomplete data assignment ( $r = 0$ ) then the importance list cannot be meaningfully used because in this case, the BI the computation would have to be divided by zero. The BI is defined as the ratio of the probabilities of occurrence and describes the conditional probability of the Top Event when the event considered has already occurred. In the case of a single point fault – i.e. without any safety mechanism – the Top Event occurs definitely (probability is 1), the BI for the single point fault must therefore equal 1. For details refer to “Step 6: Quantitative interpretation”.



Consequentially, all events with a BI = 1 invariably constitute single point faults. (CAUSE\_CC and CCF\_CAUSE\_2\_AND\_6 – boxed with a dashed line in Figure 4.8)

*N.b.: The BI for those events that occur in combination with rare operating conditions do not permit any clear statement about whether a single point fault is concerned here. This is because a BI < 1 can also be caused by the rare operating condition and therefore does not necessarily mean that this concerns a combination multiple point faults for the event being considered. (CAUSE\_5 – boxed in dots in Figure 4.8)*

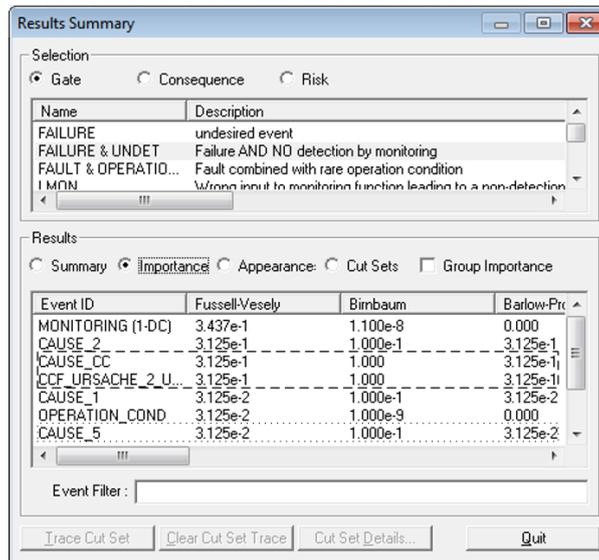


Figure 4.8: Importance list in FaultTree+

**b) Residual faults (monitored faults)**

The FTA is able to identify monitored (single point) faults (Residual Faults in the context of the ISO 26262).

A monitored fault constitutes a combination of a fault and one or several basic elements that show the gap in the diagnostic coverage of the monitoring functions.

**Interpretation in the FT tool:**

The FT tool offers two possibilities for the interpretation of fault combinations. These are:

- The Cut-Set list containing the fault combinations (minimal cuts) computed by the tool arranged according to the probability of their occurrence,
- The importance list that shows the significance (importance) for each basic event included in the fault tree. The *Birnbaum* importance can be taken for a qualitative interpretation.

Monitored faults in the Cut-Set list:

Monitored faults can be identified in the Cut-Set list of the respective Top Event:

- Cut-Sets of the 2nd order are monitored faults, if
  - (1) an event constitutes the monitored fault

AND

  - (2) the second event represents the gap in the diagnostic coverage of the monitoring
- Cut-Sets Order > 2 are to be evaluated as monitored faults if
  - (1) *Not more than one single* event constitutes a fault

AND



## Fault Tree Analysis

- (2) Of the remaining events involved *at least* one represents the diagnostic coverage of the monitoring

AND

- (3) The remaining events represent probability-reducing conditions (e.g. rare operating conditions) or further diagnostic coverage, *NOT* however faults.

No.	Q	Minimal Cut Set
1	1.000e-9	CAUSE_CC
2	1.000e-9	CCF_URSACHE_2_UND_6
3	1.000e-9	CAUSE_2_MONITORING (1-DC)
4	1.000e-10	CAUSE_5_OPERATION_COND
5	1.000e-10	CAUSE_1_MONITORING (1-DC)
6	1.000e-13	CAUSE_2_CAUSE_4_LATENT
7	1.000e-14	CAUSE_1_CAUSE_4_LATENT

Figure 4.9: Monitored faults in FaultTree+ in the Cut-Set list

### Monitored faults in the importance list:

On the basis of the importance list, no unambiguous conclusions can be drawn with regard to monitored faults. Birnbaum Importance's listed in the example considered with  $0.01 < BI < 1$  are not necessarily justified by monitoring. Events with a  $BI = 0.1$  can be combined either with rare operating conditions (CAUSE\_5 – boxed in dots in Figure 4.10:) or with monitoring (with a diagnostic coverage of the monitoring of 90 %: Cause\_1 and CAUSE\_2 boxed by a dashed line in Figure 4.10 (refer also to: C) Multiple point combination of faults, p. 22).

Event ID	Fussell-Vesely	Birnbaum	Barlow-Pratt
MONITORING (1-DC)	3.437e-1	1.100e-8	0.000
CAUSE_2	3.125e-1	1.000e-1	3.125e-1
CAUSE_CC	3.125e-1	1.000	3.125e-1
CCF_URSACHE_2_U...	3.125e-1	1.000	3.125e-1
CAUSE_1	3.125e-2	1.000e-1	3.125e-2
OPERATION_COND	3.125e-2	1.000e-9	0.000
CAUSE_5	3.125e-2	1.000e-1	3.125e-2

Figure 4.10: Unambiguous classification using BI: dotted - single point faults with a rare operating condition – interrupted line – monitored faults

### c) Combinations of multiple point faults

A combination of multiple point faults (Multiple Point Failure) is made up of several faults.



### Interpretation in the FT tool:

#### Combinations of multiple point faults in the Cut-Set list:

Combinations of multiple point faults can be identified in the Cut-Set list of the respective Top Event:

- Cut-Sets of the 2nd order are combinations of multiple point faults if BOTH events constitute faults but neither a diagnostics gap nor operating conditions.
- Cut-Sets of the higher order are combinations of multiple point faults if AT LEAST TWO events depict faults – the remaining events may then represent other contents (e.g. operating conditions, monitoring and similar).

*N.b.: In the context of the ISO 26262 it is necessary to distinguish between monitored combinations of multiple point faults and monitored single point faults (Residual Faults) because the norm places different requirements on dealing with such fault combinations.*

#### **d) Latent faults**

Latent faults are faults that require a second fault to be able to trigger the undesirable event (Top Event). Latent faults should be differentiated from “sleeping” faults that require a rare *operating condition* to be active for the undesirable event.

At the point in time of their occurrence latent faults therefore

- Are without any direct effect on the undesirable event (a second fault has to occur),
- Cannot be noticed by secondary effects (e.g. noise or loss of comfort),
- Cannot be detected by monitoring during their latency period.

Single point faults and monitored faults can in the context of the definition given above therefore never be latent. Unlike “sleeping” faults that are combined with a rare operating condition, a fault identified as latent must therefore be searched in combinations of multiple point faults – monitored or not monitored.

The identification of latent faults is possible in several ways:

- a) The fault tree is examined at the AND gates during the preparation phase.
- b) The composition of the computed Cut-Set is examined for latent faults.

#### **Investigate the fault tree to determine latent faults (Option a)**

Latent faults can be identified by considerations of the fault tree structure. Since latent faults are always part of failure combinations of multiple point faults, only gates that can lead to failure combinations because of their logic (AND, XOR, VOTE) are of interest.

Example:

The fault tree example given in Section **Fehler! Verweisquelle konnte nicht gefunden werden.** (refer also to Figure 4.5) is examined for the presence of potentially latent faults.

The considerations begin at the gate “FAULT & NOT DETECTED”. This is an AND gate that combines a fault with a monitoring:

Step 1) The inputs of “FAULT & NOT DETECTED” are first of all investigated for *potential* latency.

In the example the gate constitutes a combination of faults and monitoring. Since the fault without monitoring would most certainly lead to the undesirable event, the fault path (Gate “FAULT”) is assessed as being non-latent. A failure of the monitoring is without any consequences as long as the fault does not occur (=> gate “NO DETECTION | REACTION”)



# Fault Tree Analysis

N.b.: Color-coding the gates already analyzed when working through this task can be useful.

In the example (see Figure 11. Figure 12, Figure 13): **GREEN**: No potential latency; **PINK**: Potentially latent; **YELLOW / CREAM**: Faults; **GRAY**: Operating condition; **ORANGE**: Gap in diagnostic coverage; **LIGHT BLUE**: Still to be processed

2020-04-06 - SOCOS

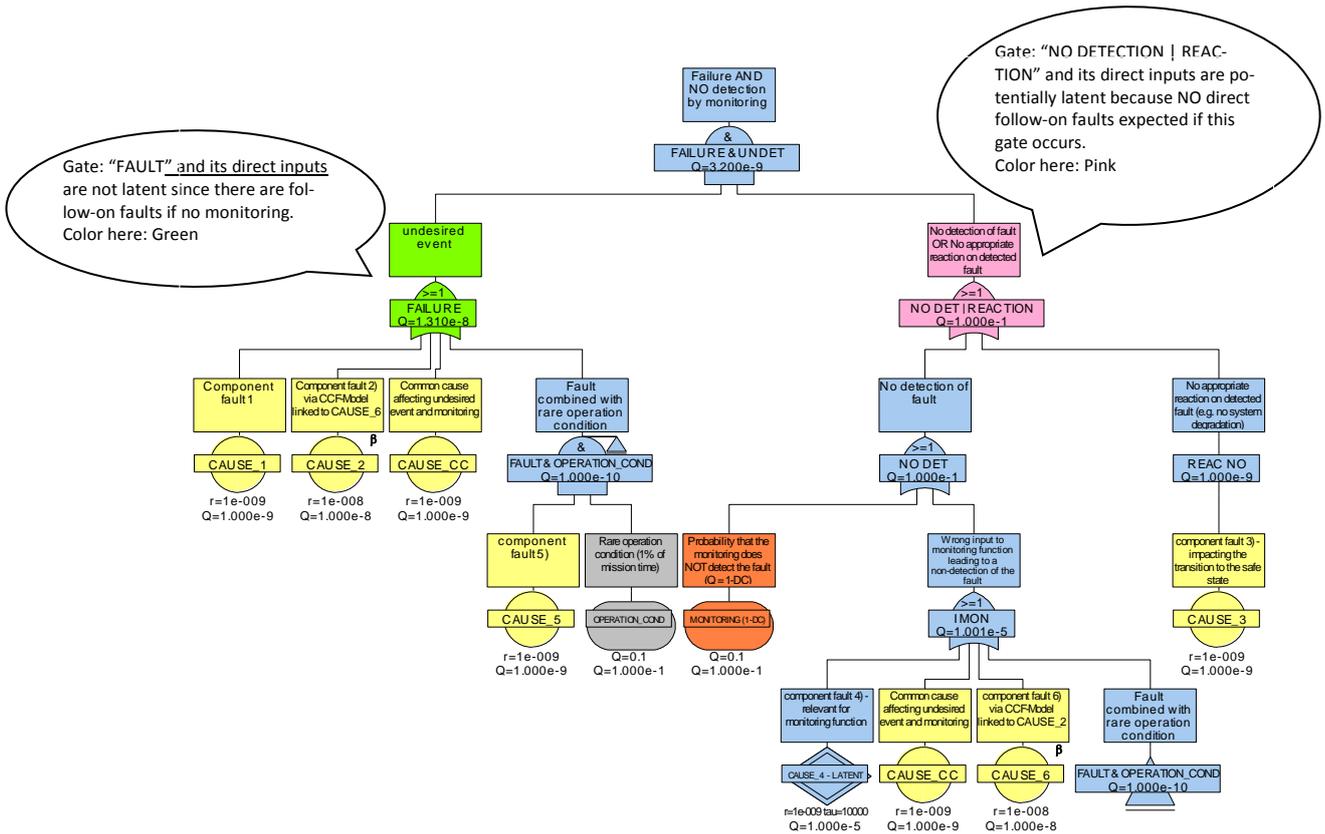


Figure 4.11: Step 1) – determining potentially latent / non-latent paths

In the next step all fault trees and events that have a direct connection with the input considered (i.e. not via an AND gate) are marked according to the classification of the gate.



# Fault Tree Analysis

2020-04-06 - SOCOS

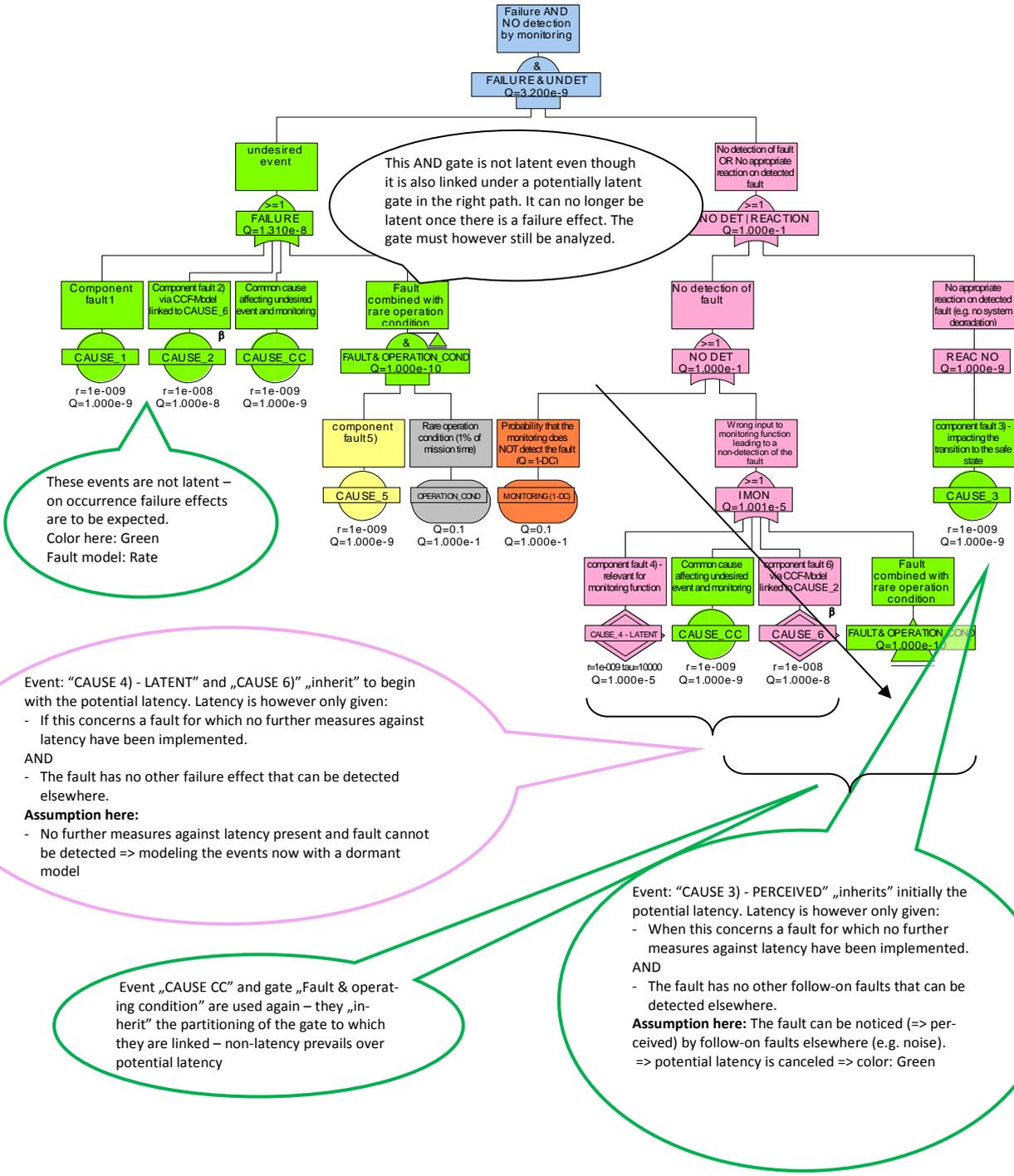


Figure 4.12: Step 2) – Inheritance of the initial classification on the directly associate FT elements (OR gates or events)



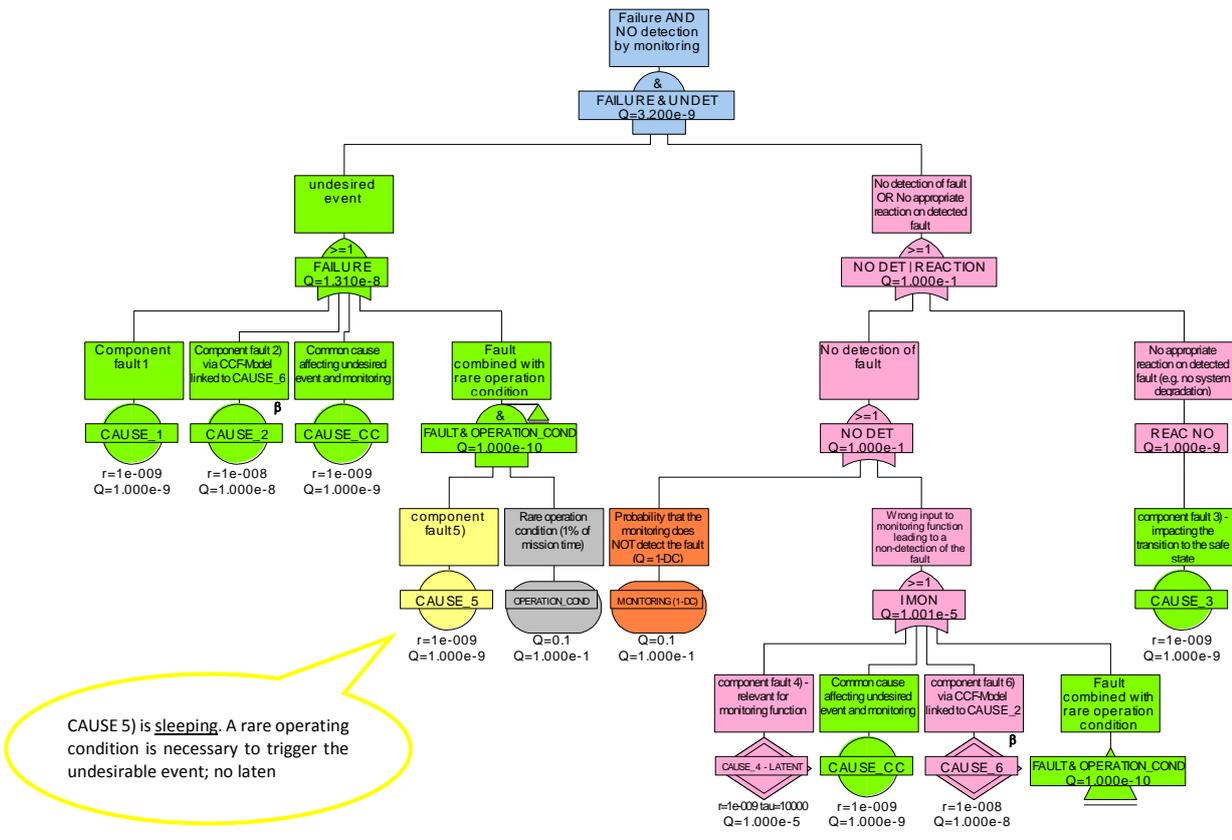


Figure 4.13: Step 3ff) – analysis of the AND gates on lower levels

**Result:**

The following events have been identified as potentially latent in the fault tree:

CAUSE\_4\_LATENT; CAUSE\_6

If faults have been classified as latent this should be accordingly taken into account by the respective fault models (e.g. dormant-fault model assigned). The latency period of individual faults (parameter  $\tau$  in the dormant model) must include the available actions against latency (e.g. power-on self-tests not considered in the fault tree either).

*Note:* Option a) is suitable for the smaller fault trees with a limited number of AND gates. It can also be carried out to accompany construction of the fault tree. To limit the efforts alternatively Option b) is available for extensive fault trees.

**Examination of the established fault combinations to determine latent faults (Option b)**

An examination for latent faults can also be carried out on the basis of minimal cuts (Cut-Sets). Naturally only Cut-Sets of at least 2nd order are in focus here since a latent fault neither can have any direct failure effect nor can be detected.

*N.b.:* The identification of latent faults on the basis of the Cut-Sets cannot be made directly in the FT tool but requires a data export into a suitable spreadsheet application. The sorting of the Cut-Sets according to the probability of occurrence may not be changed in the spreadsheet.



## Fault Tree Analysis

During the examination of the fault combinations, each event of the Cut-Set shall be considered separately – if one of the following statements applies, then a latent fault is not concerned here.

- There are monitoring mechanisms that will detect the fault (e.g. initial tests or online monitoring)
- The fault has consequences that although do not trigger the undesirable event directly (it only occurs in a multiple point fault), can however be noticeable in a different relationship (e.g. comfort limitation, noise or however that the fault is a single point fault with regard to another undesirable event in the same system) → the fault is “perceived”.
- A fault identified as being potentially latent does not have under any circumstances an effect on the undesirable event. This is the case if there is more than one single point non-latent fault in the combination of the faults of the order  $n$  (where  $n > 2$ ). This is because in a combination of faults with more than 1 fault the second non-latent fault prevents the latent fault from causing the undesirable event.

Timescale	Fault 1 not latent	Fault 2 latent	Fault 3 latent	Effect on the system
1	0	0	1	No effect
2	0	1	1	No effect
3	1	1	1	Undesirable effect occurs without any advance warning

Timescale	Fault 1 not latent	Fault 2 not latent	Fault 3 latent	Effect on the system
1	0	0	1	No effect
2	0	1	1	Fault 2 detected or perceived

Timescale	Fault 1 not latent	Fault 2 not latent	Fault 3 latent	Effect for the system
1	0	0	1	No effect
2	1	0	1	Fault 1 detected or perceived

- The considered event describes one of the following conditions (operating condition, monitoring gap (monitor), other conditions); these events are not seen as being latent.

Once made the categorization then applies for the event and its role in the fault tree being considered here.

Example of a table for the exemplary fault tree from Figure 4.5:

The search for latent faults using the second method identifies the same events as the first method.

Each Cut-Set is now considered separately on a line-by-line basis. The 1st involved fault (1st fault) is classified and this classification is then applied to all rows where the fault being considered occurs. A fault is then to be evaluated as being latent when the potential latency is confirmed such, that there are no actions against the latency. In the table this is indicated by both an entry “pL” and a further entry “L”. If unlike this the potential latency is not confirmed later then a “P” can be entered (in the example given this applies for CAUSE\_3)

*N.b.: Features of the spreadsheet application like e.g. filter options or a meaningfully programmed macro can provide efficient support when completing the table and speed up the analysis considerably!*

Classifications used in the table:

- n.a. = not relevant;
- M = Monitor gap (→ not latent);
- B = operating condition (→ not latent);
- P = Perceived (noted → not latent);
- L = Latent;
- SPF = Single Point Fault (=> not latent);
- RF = Residual Fault (monitored faults => not latent);



## Fault Tree Analysis

S = dormant fault (not latent);  
pL = potentially latent

Number	Cut-Set	Event descriptions	Unavailability	Order	1st fault	latent fault perceived fault	2nd fault	latent fault perceived fault
1	CAUSE_CC	Fault cause - Common Cause of signal error and monitoring	1E-09	1	SPF	n.a.		
2	CCF_CAUSE_2_AND_6	Common Cause faults model	1E-09	1	RF	n.a.		
3	CAUSE_2. MONITOR GAP	Fault cause 3) via the CCF model connected with fault cause 6) Gap of the monitoring quality (= 1-DC)	1E-09	2	RF	n.a.	M	n.a.
4	CAUSE_1. MONITOR GAP	Fault cause 1) Gap of the monitoring quality (= 1-DC)	1E-10	2	RF	n.a.	M	n.a.
5	CAUSE_5. OPERATING CONDITION	Fault cause 5) Rare operating condition (1% of the operating time)	1E-11	2	S	n.a.	B	n.a.
6	CAUSE_2. CAUSE_6	Fault cause 3) via the CCF model connected with fault cause 6) Fault cause 6) via the CCF model connected with fault cause 2)	5E-13	2	RF	n.a.	pL	L
7	CAUSE_2. CAUSE_4 - LATENT	Fault cause 3) via the CCF model connected with fault cause 6) Fault cause 4) - relevant for the monitoring function	5E-14	2	RF	n.a.	pL	L
8	CAUSE_1. CAUSE_6	Fault cause 1) Fault cause 6) via the CCF model connected with fault cause 2)	5E-14	2	RF	n.a.	pL	L
9	CAUSE_1. CAUSE_4 - LATENT	Fault cause 1) Fault cause 4) - relevant for the monitoring function	5E-15	2	RF	n.a.	pL	L
10	CAUSE_2. CAUSE_3	Fault cause 3) via the CCF model connected with fault cause 6) Fault cause 3) - relevant for the backup level - noticeable by the driver	1E-17	2	RF	n.a.	pL	P
11	CAUSE_1. CAUSE_3	Fault cause 1) Fault cause 3) - relevant for the backup level - noticeable by the driver	1E-18	2	RF	n.a.	pL	P

### Result:

The result agrees with the graphical analysis method. In the Cut-Set list the following events were identified as potentially latent:

CAUSE\_4 – LATENT; CAUSE\_6

The latency period of the latent faults (parameter  $\tau$  in the dormant model) must include available actions against latency (e.g. modeled power-on self-tests not considered in the fault tree).

### Identified latent faults – consequences for fault tree modeling

Latent faults must have a fault model that takes their latency period into account. This can be realized by assigning a dormant model in the FTA tool.

The latency is determined by the time interval that can elapse between the occurrence of the fault and its possible detection.



Example: If a fault is detected by a test executed on a regular basis (e.g. every 10 operating hours) then the longest possible time interval of 10 h corresponds to its latency (fault occurs immediately after the last test; it takes another 10 h until the next test).

By applying the dormant fault model, the latency period leads to a considerably higher failure probability of latent faults compared to non-latent faults when determining the failure probability of the fault – assuming a meaningful selection of the fault tree computation parameters. It is therefore meaningful to take actions against the latency of faults in systems.

For details about fault models see *Attachment 1: Symbols and modeling recommendations*

### 4.6. Step 5: Determine the probability of occurrence of basic events (quantitative description)

#### 4.6.1. General

To be able to perform a quantitative analysis of the fault tree the basic events must be assigned the respective probability of occurrence. These probabilities of occurrence can be determined on the basis of suitable reliability parameters.

Reliability parameters are for example

- the failure rate  $\lambda$  (time-dependent variable, e.g.  $r = 1 \text{ E-}09/\text{h} = 1 \text{ FIT}$ )
- the failure probability  $Q_A$  (time-independent probability, e.g.  $Q = 0.1$ )
- the probability of occurrence  $Q_E$  (time-independent probability, e.g. ambient conditions like rain, snow, ...)

There are different approaches to determine the reliability parameters for the quantitative analysis of the fault tree. Often a certain approach is specified by the customer. If this is not the case then the approach has to be agreed in the team.

As a rule the reliability parameters are determined from one of the following five sources of information:

- In-house experience from the field or from operation  
Advantage: Real service conditions  
Disadvantage: The product has to be in the field, data are then available later.  
Alternative: Use data for a comparable predecessor product (limitation: The applicability has to be checked)
- In-house tests and / or test experience  
Advantage: Information is available before the product is in the field (random sample / time compression)  
Disadvantage: Only test conditions; additional efforts and costs
- Literature data / generic data from data collections (e.g. handbooks; the use of “Siemens SN29500” or “IEC62380” is recommended)  
Advantage: Wide data basis, recognized data basis  
Disadvantages: Applicability to the application in question is not always given;  
New types of components / new technologies are not included;  
historic data – more conservative

Note: If the service or boundary conditions (e.g. temperature profile) on which the handbook data are based should be different for the undesirable event or product to be investigated then the data have to be adjusted before using in the fault tree.

For the FTA in the context of automotive applications, the Siemens norm (SN29500) has asserted itself in determining the reliability parameters for the verification of products.

- Experience of Bosch-external sources e.g. suppliers



## Fault Tree Analysis

- Expert estimates (with justification) if none of the options given above provides a suitable reliability parameter for the application case

The data are normally given as failure rates, in “ppm per time unit” or “FIT”.

As a rule, the values refer to the whole component. In the FTA however, the failure rate with respect to the fault mode of a part of the component is needed.

For simple components (e.g. diode, resistor,...) the overall rate can be broken down according to fault catalogs (e.g. short, open, drift). Used most often is the standard work from A. Birolini (Reliability Engineering: Theory and Practice, Springer Verlag).

For complex components (e.g. microcontrollers) the total failure rate is typically broken down according to the surface proportions of the circuit blocks. The fineness of the breakdown is determined by the requirements of the FTA.

It can be necessary in the course of a development of a project to update the reliability parameters.

### 4.6.2. Preventive / corrective

In the preventive use of the FTA all information sources for determining reliability parameters can principally be used (see 4.6.1).

In the corrective application case of the FTA reliability parameters preference is given to using in-house or external experience from the field and operation because in this case any discrepancy occurring in the application shall be investigated.

## 4.7. Step 6: Quantitative interpretation

### 4.7.1. General

In order that the quantitative interpretation can lead to informative results, the qualitative analysis – i.e. of the fault tree – must be concrete. Furthermore both the data assignment of the basic events must be explained and the parameters for the computation of the fault tree results have to be set.

### 4.7.2. Definition of the computing parameters in the FTA tool

#### Mission time

It shall established as early as possible during the preparation of the FTA, which mission time (computation time) – in the automotive field this corresponds to the drive cycle – will be selected for the quantitative interpretation of the FTA. The results, that the FTA will provide, depend crucially on this.

Remark: The corresponding option in the FT tool FT+ is “System life time” (Options/Calculation).

Depending on the application case, different times can be meaningful here. In the automobile industry a mission time of 1 h has asserted itself e.g. in the use of the FTA for supporting the safety evidence per ISO 26262. This has a certain reference to the average utilization time of vehicles and enables a meaningful computation of the failure rate of non-latent faults (rates model) in the comparison with latent faults (dormant model).



### Dormant fault model

Computation of the dormant fault models is possible in different ways. The most conservative result (highest failure probability) is given by the setting “Max”. It is assumed here that the latent fault occurs at the beginning of the vehicle lifetime and then stays over the whole latency period. Here the fault occurs by chance always at the beginning of the vehicle lifetime – in reality this not the case.

The option “Mean” assumes an equal distribution of the fault occurrence over the vehicle lifetime. The calculations therefore give the mean value for the probability ( $Q = \lambda * \tau / 2$ ) of a latent fault.

The other computing method (ISO61508) can be meaningful as well depending on the project. Further information here is provided by the Help available in FaultTree+ (or RWB)

### **4.7.3. Numerical value of the Top Gates**

The FTA tool automatically calculates for each gate

- the non-availability Q
- the occurrence frequency  $\omega$
- the Mean Time To Failure (MTTF)
- the Mean Time to Repair (MTTR)
- the Mean Time between Failure (MTBF)

The tool supports displaying in each case the type of results desired in the fault tree.

Depending on the boundary conditions under which an FTA is prepared, the respective parameters have to be in line with the previously defined target values.

#### Non-availability Q:

Q is dimensionless and describes the probability of occurrence of the undesirable event at the end of the computed time period. If statements have to be made for Q with regard a certain time period then the computation time has to be taken into account. If a computation time (mission time, “System life time” see above) other than 1 hour is selected then this will lead to errors in the end result because the division of Q by a number limits the probability per hour.

Example: An event occurring with certainty after 10.000 hours with the probability  $Q = 1$  is given by division as:

$$Q^*/1 \text{ h} = Q / 10.000 \text{ h} = 1\text{E-}04 \text{ 1/h}$$

I.e. the probability of this event can never be higher than  $Q^*/1 \text{ h} = 1\text{E-}04 \text{ 1/h}$ ! This is not plausible.

#### Occurrence frequency $\omega$

$\omega$  has the dimension 1/h and corresponds to Q differentiated according to the time. I.e.  $\omega$  describes the course over time (mathematically the gradient) of the probability curve Q with t. As long as the products of the failure rates and the computation time are very much smaller than 1 ( $\lambda * t \ll 1$ ) the probability curve is in the linear range, i.e. the derivation of the exponential curve provides the same value as the curve itself  $\rightarrow \omega \approx Q$ .

Fault trees that include numerous basic events with fixed (constants) probabilities ( $\omega = 0$  here) can lead to differences in the numerical values between  $\omega$  and Q.



Mean Time To Failure (MTTF), Mean Time to Repair (MTTR), Mean Time between Failure (MTBF)

These terms play a role as a rule in systems that can be repaired whereby the MTTF is related to the failure rate  $\lambda$  (under certain prerequisites it applies:  $MTTF \approx 1/\lambda$ ). These terms are not gone into in any further detail in this guideline.

**4.7.4. Identify optimization potential**

If a need for action is determined based on the considerations of the numerical values in the Top Event because e.g. the set goals have not yet been reached, then optimization potential can be identified. This is done by consideration of the importance of basic events. Importance considerations allow statements to be made on the significance of basic events with regard to particular issues.

**Example 1: Reduction of the total failure probability**

One possible task after determining the probability of occurrence of the Top Event might be: *The total probability of occurrence must be lowered to below a certain value.* To optimize efficiently the parts of the FTA have to be identified where changes with respect to the assigned task are particularly active. The issue here is:

*By which amounts do the basic events contribute to the total failure rate of the Top Event?*

This question can be answered using the Fussel-Vesely Importance.

**Fussel-Vesely Importance (FVI):**

The FVI is a quotient of:

- the sum of the probability of occurrence of all minimal cuts where the considered Basic event A is involved with the probability of occurrence  $q_A$ , and
- the probability of occurrence of the undesirable event  $Q_{SYS}$ .

$$I_A^{FV} = \frac{q_A \cdot q_B + q_A \cdot q_C + \dots}{Q_{SYS}}$$

Since  $q_A$  has to be greater than zero and can never be larger than  $Q_{SYS}$ , it applies for the FVI that

$$0 < FVI \leq 1.$$

If the FVI were = 0 then the event would have no contribution to the overall result – and then it may not occur in any minimal cut that can trigger the Top Event, and thus it must apply that  $FVI > 0$ .

If unlike this  $FVI = 1$ , then the Basic event is the *only* event than can trigger the undesirable event.

*Remark: The selection of the approximation method “Rare” in the Tool FT+ can lead to  $FVI = 1$  and despite this that other causes can trigger the Top Event. Background here is a simplified computation of the total probability of occurrence as a straightforward sum of the part-probabilities of the available Cut-Sets.*

Basic events that have a high FVI (e.g.  $FVI = 0.7$ ) also contribute significantly to the total probability of occurrence – 70% in the example here. Apart from this, it is likely that basic events with a high FVI are single point faults, or are monitored faults with a high failure rate. These events and their position in the fault tree are therefore of interest when it applies to reduce the probability of occurrence of the undesired event.

See Step 7 for actions to reduce the influence.



**Example 2: Effectiveness of monitoring and redundancy for safeguarding against the Top Event**

When the results of the quantitative analysis are available it can be of interest to know how likely the occurrence of the undesirable event is when a fault has occurred.

The associated question here might then be:

*What is the probability of the undesirable event occurring if the basic event to be considered has occurred?*

This question can be answered using the Birnbaum Importance.

**Birnbaum Importance (BI)**

The Birnbaum Importance (BI) is defined as the quotient of:

- the sum of the probability of occurrence of *all* minimal cuts where the considered basic event A is involved with the probability of occurrence  $q_A$ , and
- the probability of occurrence of the basic event  $q_A$  itself

$$I_A^{BI} = \frac{q_A \cdot q_B + q_A \cdot q_C + \dots}{q_A}$$

Since the denominator  $q_A$  has to be greater than zero and can never be larger than  $q_A$  (when  $q_A$  is a single point fault in the numerator), the BI is defined as

$$0 < BI \leq 1.$$

If the BI were = 0 then the event would have no contribution to the overall result – and then it may not occur in any minimal cut that can trigger the Top Event, and thus it must apply that  $BI > 0$ .

Faults where  $BI = 0$  are not included in the importance list.

If unlike this  $BI = 1$ , then the basic event is a single point fault that can trigger the undesirable event. Then  $q_A$  must be in the numerator as the minimal cut (without further faults that would cause a smaller product probability) – the denominator is still  $q_A$ .

*Remark: In some cases  $BI > 1$  is outputted despite the definition given above. This indicates logic problems in the fault tree and occurs amongst others, when one and the same event Cut-Sets forms with several further events within a fault tree. The sum of the probabilities of these basic events must exceed 1. Such events with high probabilities often concern the representation of diagnostic slip. A check of the fault tree logic is then appropriate. Example:  $q_A = 1E-07$ ;  $q_B = 0.6$ ;  $q_C = 0.7$  with the fault combinations  $q_A \cdot q_B$  OR  $q_A \cdot q_C$  would give  $BI(q_A) = 1.3$ . since  $q_B + q_C = 1.3$*

Basic events that have a high BI lead with high probability or even certainty (in the case of  $BI = 1$ ) to the undesirable event. Unlike like this, basic events with a lower BI need at least a second or third event for the undesirable event to be triggered.

The following additional issue can be examined using the Birnbaum Importance:

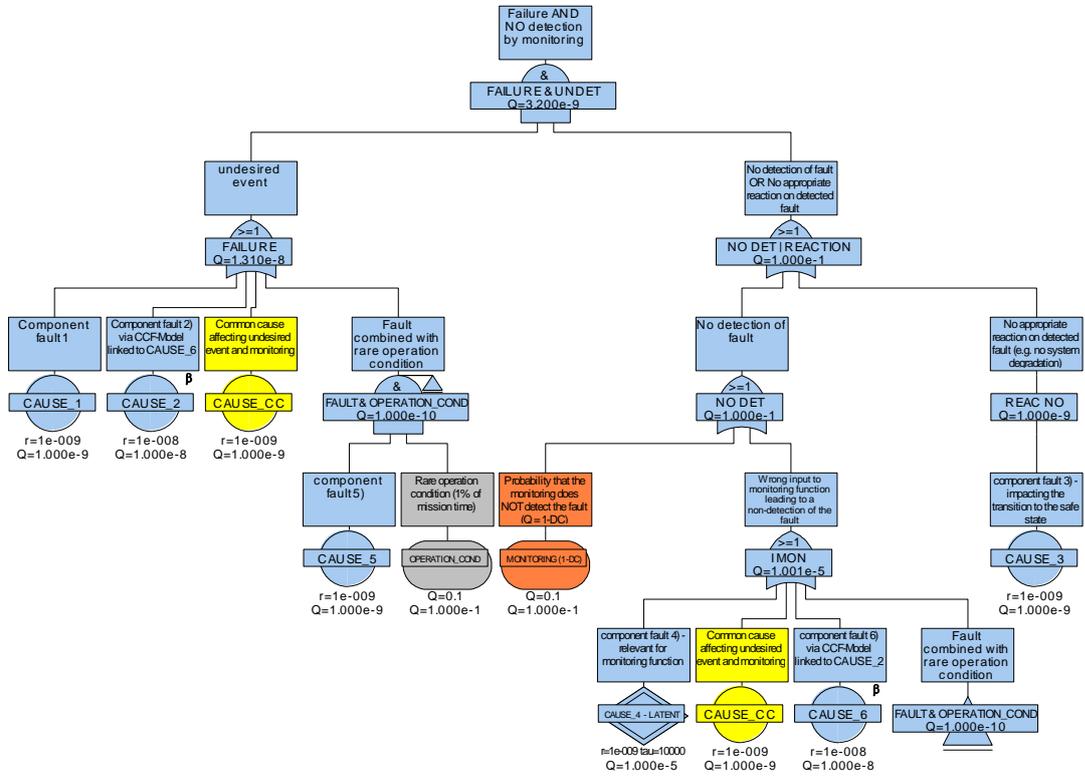
- What is the significance of the monitoring in the system?
  - ⇒ The BI for the basic event that defined the failing of the monitoring is provided by the sum of the probabilities of occurrence of all faults that can be detected with the monitoring – the higher the BI the greater the significance of the monitoring.
- Which basic events are safeguarded by monitoring?
  - ⇒ Monitored faults have a BI between 0.001 and 1 depending on the monitoring gap of the associated monitoring (i.e. a 99% effective monitoring has a gap of 1% = 0.01 and this number then appears as the BI of the monitored fault).

*Remark: "Sleeping faults" (faults that are only active in combination with rare operating conditions) have the BI corresponding to the probability of the rare operating condition. In certain circumstances there is a danger of mistaking this for monitored faults.*



**Example 3: Interpretation of a fault tree example:**

The fault tree example shown in Figure 4.14 has been used already in Step 4: Qualitative interpretation. Explanatory notes on the technical background of the fault tree can also be found here.



**Figure 4.14 : Fault tree example**

The fault tree above provides the following minimum steps for the Top Gate “FAULT & NOT DETECTED”:

No.	Cut-Set	Event Descriptions	Unavailability	Order
1	CAUSE_CC	Fault cause - Common Cause of signal error and monitoring	1E-09	1
2	CCF_CAUSE_2_AND_6	Common Cause faults model	1E-09	1
3	CAUSE_2. MONITOR GAP	Fault cause 2) via the CCF model connected with fault cause 6) Gap of the monitoring quality (= 1-DC)	1E-09	2
4	CAUSE_5. BOPERATING CON- DITION	Fault cause 5) Rare operating condition (10% of the operating time)	1E-10	2
5	CAUSE_1. MONITOR GAP	Fault cause 1) Gap of the monitoring quality (= 1-DC)	1E-10	2
6	CAUSE_2. CAUSE_4 - LATENT	Fault cause 2) via the CCF model connected with fault cause 6) Fault cause 4) - relevant for the monitoring function	5E-14	2



## Fault Tree Analysis

No.	Cut-Set	Event Descriptions	Unavailability	Order
7	CAUSE_1. CAUSE_4 - LATENT	Fault cause 1) Fault cause 4) - relevant for the monitoring function	5E-15	2
8	CAUSE_2. CAUSE_6	Fault cause 2) via the CCF model connected with fault cause 6) Fault cause 6) via the CCF model connected with fault cause 2)	1E-16	2
9	CAUSE_1. CAUSE_6	Fault cause 1) Fault cause 6) via the CCF model connected with fault cause 2)	1E-17	2
10	CAUSE_2. CAUSE_3	Fault cause 2) via the CCF model connected with fault cause 6) Fault cause 3) - relevant for the backup level	1E-17	2
11	CAUSE_1. CAUSE_3	Fault cause 1) Fault cause 3) - relevant for the backup level	1E-18	2

The thereby associated importance list for the Top Gate "FAULT & NOT DETECTED" is:

Name	Fussell-Vesely Importance	Birnbaum Importance
MONITOR GAP	0.343744	1.1E-08
CAUSE_2	0.31251	0.100005
CAUSE_CC	0.312495	1
CCF_CAUSE_2_AND_6	0.312495	1
CAUSE_1	0.031251	0.100005
OPERATING CONDITION	0.031249	1E-09
CAUSE_5	0.031249	0.1
CAUSE_4 - LATENT	1.72E-05	1.1E-08
CAUSE_6	3.44E-08	1.1E-08
CAUSE_3	3.44E-09	1.1E-08

### Evaluation of the basic events on the basis of the Fussell-Vesely Importance (FVI)

It can be seen on the basis of the importance list (sorted according to the FVI) that there are four basic events (MONITORING\_GAP, CAUSE\_2, CAUSE\_CC CCF\_CAUSE\_2\_AND\_6), of which three fault combinations (see Cut-Set list), each more than 30 %, contribute to the result (FVI > 0.3). If the goal is to reduce the total result, these fault causes and their fault combinations should be focussed. Two cases concern Common Cause faults for both sub-fault trees (CCF\_CAUSE\_2\_AND\_6 as well as CAUSE\_CC). These are apparent as single point faults – and can be seen by amongst others, from their values for the Birnbaum Importance (BI = 1).

Three further basic events contribute by their fault combinations in each case with about 3%, to the total result (CAUSE\_1. OPERATING CONDITION, CAUSE\_5 WITH FVI = 0.031251).

Finally there are also three basic events where the fault combinations of these events have only a small influence on the total result. Their FVI is therefore very small (FVI << 1). The optimization of these events or of the fault tree where these events have an influence is therefore not a meaningful option when the total result shall be lowered.

### Evaluation of the basic events on the basis of the Birnbaum Importance (BI)

The monitoring MONITOR GAP detects faults, the probability of which is 1.1E-08. When it is considered that MONITOR GAP has a fixed probability of 0.1 (see

Figure 4.14), then the BI of CAUSE 1 and CAUSE 2 with BI = 0.1 can be understood – since these faults are monitored by the MONITOR GAP from which the CAUSE 1 provides a probability 1E-09 and CAUSE



2 a probability of  $1E-08$  (to be seen in the fault tree shown). Reason for the fact, that the BI for these events is not exactly 0.1 ( $BI = 0.100005$ ) is that there are combinations of these faults with other basic events.

On the basis of the BI statements about the “importance” of a monitoring can thus a comparison between the different monitoring types be made. If the BI of a monitoring lies e.g. in the region of the probability of a double point faults (typical for ISO 26262 applications  $Q < 1E-12$ ), then it can be questioned whether this monitoring is needed for safeguarding against a single point faults – though it should not be overlooked that there might already be some effect here against latent faults occurring.

The sleeping fault is CAUSE 5, its BI of 0.1 is exactly the same as the fixed probability of the OPERATING CONDITION.

If the latent faults CAUSE 4) – LATENT occurs, then the undesirable event FAULT & NOT DETECTED follows with a probability of  $1.1E-08$  (CAUSE 1 WITH  $Q = 1.0E-09$  or still occur).

### Further importances:

Further importance's offered by the fault tree Tool FT+ (Barlow-Proschan, Sequential Importance) only play a role in combination with sequentially occurring faults and are not considered any further here.

## **4.8. Step 7: Establish the need for action and success monitoring**

Following the qualitative or quantitative interpretation of the fault tree, a decision can be made on the basis of the results whether there is a need for action and if so, then the extent of the action.

The objective criteria established before the analysis should be used again in this step.

For example the interpretation of the fault tree can show that there are single point faults that lead directly to the Top Event or that the probability of occurrence of the Top Event is higher than is required for compliance with the safety goal.

In such cases it can be necessary to define suitable actions so to reach the exact goals of the FTA.

The decision here is thereby made by the customer.

The FTA moderator can provide support in the interpretation of the results, and the FTA team acts in an advisory capacity and can assist in the definition and implementation of actions.

For interpretation of the Cut-Set lists and the importance lists, refer also to Step 6 (quantitative interpretation).

Examples for possible actions:

- Use of *components with a reduced failure rate* ( $\Rightarrow$  leads to a different data assignment of the basic event)
- *Introduce redundancy* that can make the basic event concerned into a multiple point fault ( $\Rightarrow$  has an influence on the architecture and moves the basic event into a multiple point fault  $\Rightarrow$  effectiveness better than for introducing monitoring)
- *Introduction of monitoring* that reduces the failure rate with which a basic event has an effect on the undesirable event (turns a single point fault into a monitored fault  $\Rightarrow$  effectiveness depending on the quality of the monitoring)
- *Check of the specified service conditions (e.g. temperature,...)* ( $\Rightarrow$  influence on the failure rates of components already in use and on the monitoring is possible)
- *Detailing of the analysis for checking the status of the single point fault* (an analysis in detail can highlight safety measures not considered up to now due to simplification, e.g. monitoring or redundancy in a sub-component)
- *Measures in the field*



### Success monitoring

If the probability of the Top Event shall be reduced, then success monitoring of the applied actions for the preventive FTA can be realized by repeating the computation of the probability of occurrence of the Top Event.

If the number of single point faults shall be reduced then the Cut-Set list can be used for checking the actions.

With the corrective use of the FTA (implementation of the thereby derived actions) success monitoring is given by the absence of the fault status, e.g. no further field failures.

## 4.9. Step 8: Release and documentation of the FTA

A detailed documentation of the results of an FTA is indispensable since the FTA diagram alone as a rule does not contain (or show) all the information that is needed for an understanding or evaluation. Merely saving the FTA file does not fulfill the purpose as a rule. An added factor is that a changing composition of the team causes the knowledge carriers of the project to no longer be available at a later point in time. Also, the FTA report forms the basis for a possible routing for signatures that – comparable with routing for signatures for an FMEA – can be realized by means of the “eSignature”.

Because of the diversity of the individual fault tree status analyses in the scope, depth of the considerations made, etc. no fixed requirements on the design of the FTA report can be given. It is for this reason that only a recommendation can be given in this document.

It shall principally be observed that an FTA and the associated report can contain sensitive information. When handing over to external recipients (e.g. customer) the [“Handling of Results from Qualitymanagement-Methods towards Customers”](#) in its currently valid edition are therefore to be observed.

Best practice structure of an FTA report:

- Summary of the results (similar to the cover sheet of an FMEA):
  - Overview of the task (e.g. Top Events)
  - Participating persons
  - Detailed listing of the results (target / actual comparison) for each individual Top Event (including the assessment by experts):
    - Name / description
    - Target values / determined values
    - Top-Faults (e.g. on the basis of a Cut-Sets analysis)
    - Listing the assumptions made (e.g. limiting the scope of the considerations, definition of the “safe state”)
    - Information about the data basis used (e.g. boundary conditions, sources for the failure rates, assumptions for the monitoring quality)
    - Listing of the attachments included
- List of recommended attachments:
  - List of defined gates in the FTA (including remarks; can be exported from FT+ )
  - List of the events defined in the FTAs (including remarks on deriving the failure rates or the data source; can be exported from FT+)
  - Printout of the fault tree (FTA diagram)
  - Printout of a “list of open items” if maintained



## Fault Tree Analysis

- Printout of a schedule / attendance / participation list
- Block diagrams / diagrams / graphics / view cells that contribute to understanding the system

An example for such a report on the basis of the Bosch eForms template “Report” can be found in “Attachment 2 – example of a report”



## 5. Literature on FTA

Norms, Guidelines, Handbooks

### 5.1. Norms

**[5.1.1] DIN 25424-1, Fehlerbaumanalyse – Methode und Bildzeichen (Deutsche Norm)**

Deutsches Institut für Normung, Sep. 1981, Beuth Verlag GmbH, Berlin

**[5.1.2] DIN 25424-2, Fehlerbaumanalyse – Handrechenverfahren zur Auswertung eines Fehlerbaumes (Deutsche Norm)**

Deutsches Institut für Normung, April 1990, Beuth Verlag GmbH, Berlin

**[5.1.3] DIN EN 61025 Fehlzustandsbaumanalyse (Deutsche Version der europ. Norm EN 61025)**

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE, August 2007, Beuth Verlag GmbH, 10772 Berlin

**[5.1.4] ISO 26262 Straßenfahrzeuge – Funktionale Sicherheit (Teil 1 - 10)**

Internationale Organisation für Normung, November 2011, Genf (Schweiz)

**[5.1.5] SN 29500, Ausfallraten Bauelemente (Teil 1-16) (Siemens Norm)**

Siemens AG, 2004-2014, München und Erlangen

### 5.2. Standards

**[5.2.1] Qualitätsmanagement in der Automobilindustrie, Band 4, Kapitel 4, Fehlerbaumanalyse**

VDA, Verband der Automobilindustrie e.V., Oberursel, Deutschland (2003)

### 5.3. Handbooks

**[5.3.1] Fault Tree Handbook, NUREG-0492 (US-Standard)**

D. F. Haasl et al., U.S. Nuclear Regulatory Commission, Washington, USA (1981)

**[5.3.2] Fault Tree Analysis Application Guide (international standard)**

D. J. Mahar et al., Reliability Analysis Center, Rome, USA (1990)

**[5.3.3] Fault Tree Handbook with Aerospace Applications (Aerospace industry)**

W. Vesely et al., NASA Office of Safety and Mission Assurance, Washington, USA (2002)

**[5.3.4] VDI 4008 Blatt 7, Strukturfunktion und ihre Anwendung (VDI-Handbuch Technische Zuverlässigkeit)**

Verein Deutscher Ingenieure, VDI-Verlag GmbH, Düsseldorf (1986), Bezug: Beuth Verlag GmbH, Berlin

**[5.3.5] IEC TR 62380, Reliability Data Handbook (Internationaler Standard)**

(Ermittlung von Zuverlässigkeitsdaten, die für die Berechnung der FTA benötigt werden)

Internat. Electrotechnical Commission, Geneva, Switzerland (2004)

**[5.3.6] Reliability Engineering: Theory and Practice**

Alessandro Birolini, ISBN: 3-642-39534-1, Springer Verlag, 2014 (7th edition)



## 5.4. Reference books

### [5.4.1] Die Fehlerbaum-Methode

W. Schneeweiss, ISBN 3-934447-02-3, LiLoLe Verlag, Hagen, Germany (1999)

### [5.4.2] Taschenbuch der Zuverlässigkeits- und Sicherheitstechnik

(Darstellung unterschiedlicher Methoden und Techniken der Zuverlässigkeits- und Sicherheitstechnik)  
A. Meyna, B. Pauli et al., ISBN 3-446-21594-8, Carl Hanser Verlag, München – Wien (2003)

### [5.4.3] Fault Tree Analysis Primer

Clifton A. Ericson II, ISBN 9781466446106, CreateSpace Inc., Charleston, NC (2011)

### [5.4.4] Fehlerbaumanalyse in Theorie und Praxis

(Grundlagen und Anwendung der Methode), F. Edler, M. Soden, R. Hankammer  
ISBN 9783662481653, Springer-Verlag, Berlin Heidelberg (2015)

### [5.4.5] Erstellung von Fehlerbäumen

(Eine strukturierte und systematische Methode), W. Freese, ISBN 9783446445161,  
Carl Hanser Verlag, München (2015)



## 6. Glossary

Failure	Termination of the capability of a unit under consideration to fulfill the required function.  Remark: With complex systems "Failure" is used synonymous with "breakdown". (DIN 40 041-3)
Operating condition	The working point, also known as the operating point or condition is a certain point in the map, the characteristics of a technical device or system that is assumed because of the system properties and acting external influences and parameter.
CCF	Common Cause Failure
Common Cause	Common fault cause within components that within a system are actually seen as being independent and that form a redundancy. The Common Cause overcomes this redundancy. Example: Faults of a common voltage supply of two independent sensors.
Cut-Set-Analysis	Is an interpretation of the fault tree and provides a list of the minimal cuts of the fault tree.
Diagnostic coverage / degree of diagnostic coverage	Proportion of the total failure rate of a fault cause that is detected or controlled.
DRBFM	Design Review Based on Failure Modes
ECU	Electronic Control Unit, control unit
E-Gas	Electronic accelerator pedal
et al.	<b>et al.</b> means literally "and others".
Fault status	Non-fulfillment of at least one requirement on a required characteristic of a unit under considerations. (VDI / VDE 3542)
FHA	Functional Hazard Analysis
FIT	<b>Failure In Time</b> For technical systems the failure rate is usually given in FIT. 1 FIT = 1·10 <sup>-9</sup> failures per second
FMEA	Failure Modes and Effects Analysis
FMEDA	Failure Modes Effects and Diagnostic Analysis
G&R	Hazard & Risk analysis
H&R	Hazard & Risk Analysis



## Fault Tree Analysis

Importance	Influence of certain causes = basic events on the safety, reliability, probability of occurrence or availability of the considered Top Event
Critical event	Critical events can be minimal cuts or other events defined within the scope of the system analysis that is subject to a special consideration focus. The prioritization can be made e.g. on the basis of the probability of occurrences or by establishing singular events.
Critical path	Critical paths serve the visualization / designation of the causal chain from the Top Event through to an event / combination of events that is classified as critical.
Minimal cut	<p><b>Minimal cut per EN 61025</b> Smallest event quantity needed for the main event to occur.</p> <p><b>Minimal Cut-Set</b> The fault tree structure constitutes the functional relationship of the Top Event by the logical Left through to the basic events. The logical Left lead to the formation of sub-quantities of events by the occurrence of which the Top Event is causes or the unit under consideration (e.g. system) fails. These are referred to as cuts. The smallest common quantity of basic events forms a <b>minimal cut</b> when it does not include any other cut as a true sub-quantity. The sum of all minimal cuts describes the failure behavior of the fault tree structure completely. The smallest quantity of elements of the minimal cuts can constitute an event. The maximum quantity of elements of minimal cuts and number of minimal cuts is determined by the logical Left of the fault tree structure.</p>
MPF	Multiple-Point Fault (designation from the context ISO 26262): Fault that as long as this occurs alone does not lead directly to the violation of a safety goal or to the occurrence of the undesirable event. A second fault is needed for this.
OEM	Original Equipment Manufacturer; in this publication it is thus the vehicle manufacturer that is meant here
PHA	Preliminary Hazard Analysis
ppm	<b>parts per million</b> Relationship of the faults with respect to 1 million parts. 100 ppm means 100 faults / 1 million parts. This corresponds to 0.01 % faults.
QFD	Quality-Function-Deployment, or depicted quality functions
RAMS	<b>R:</b> Reliability <b>A:</b> Availability



## Fault Tree Analysis

	<p><b>M:</b> Maintenance (maintainability)  <b>S:</b> Safety (Safety)</p>
Redundancy	<p>Presence of more operational means in a unit that are needed for fulfillment of the required function.  <u>Remark 1:</u> How many means without redundancy are needed depends on the particular case.  <u>Remark 2:</u> Maintaining the redundancy requires maintenance, i.e. the monitoring, the maintenance for in the case of failure the restoration of the functional capability of all means. (DIN 40 041)</p> <p><b>In the case of this publication:</b>          Redundancy is the additional presence of functionally equal or comparable resources of a <a href="#">technical system</a>, when these are not needed in the normal case for fault-free operation. (source: Wikipedia)</p>
RF	Residual Fault (designation from the context ISO 26262) = residual fault of a Single Point Fault that after considering the safety mechanism (monitoring) leads to the violation of safety goal or the occurrence of the undesirable event.
RWB	Reliability Work Bench (ISOGRAPH SW); FT+ successor
Sensitivity Analysis	Methodology used to assess how sensitive indicators react to small changes in input parameters.
SPF	Single Point Fault (designation from the context ISO 26262) = faults that lead directly and without any safety mechanism to the violation of safety goals or the occurrence of the undesirable event.
TCD	Technical Customer Document(s)
VDA	Association of the automobile industry
ZKG	Reliability parameters
8D report	<p>The <b>8D report</b> is a document that within the scope of <a href="#">Quality Management</a> is exchanged in the event of a complaint between supplier and customer. <b>8D</b> thereby stands for the <b>eight</b> obligatory <b>disciplines</b> (process steps) that are necessary for working through a <a href="#">complaint</a> to completion in order to overcome the fundamental problems. In the 8D report the type of complaint, the responsibilities and the actions for rectifying the deficiency are formalized. The 8D report is standardized, amongst others by the <a href="#">Association of the automobile industry</a>.</p>

2020-04-06 - SOCCOS



## 7. Attachment 1 Symbols and modeling recommendations

### 7.1. Handling variants

Subtrees can be activated or deactivated using logical switches for showing variants. The tool Fault-Tree+ has the event symbol “House” to this end.

*Handling variants for a supplementary branch*

Figure 16 shows a branch that can be activated by the switch “SW ADDITIONAL FUNCTION”.

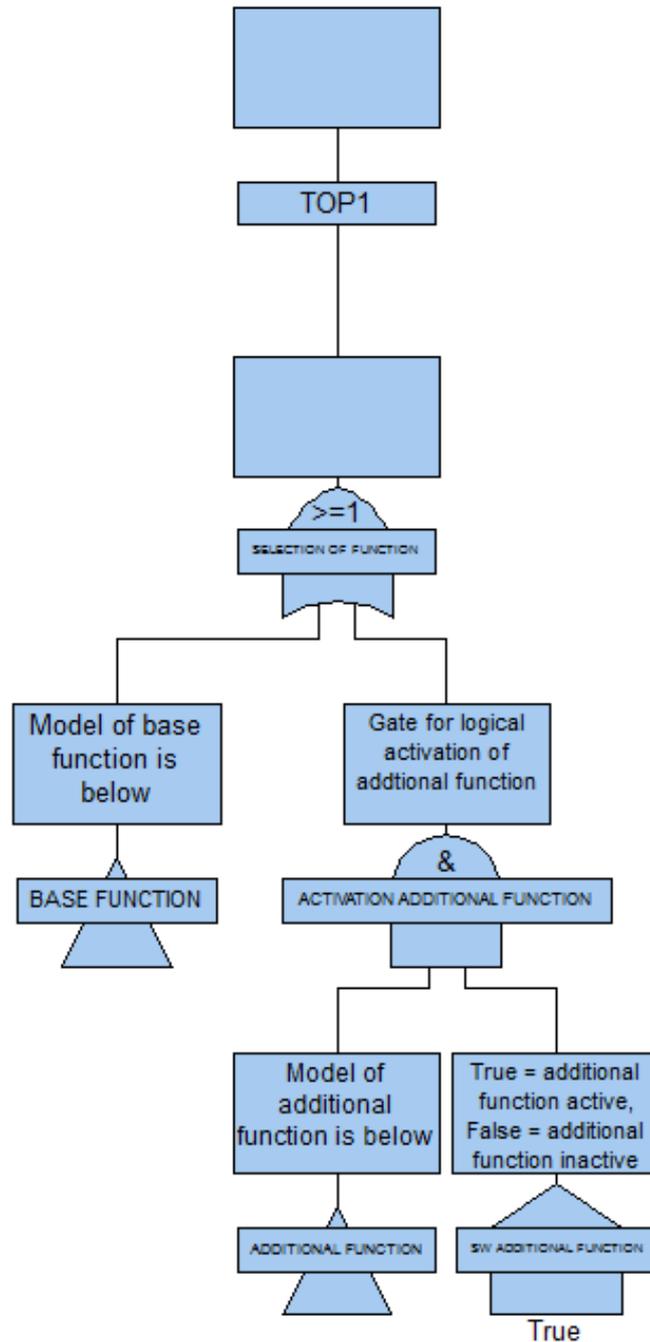


Figure 7.1: Handling variants for a supplementary branch



Figure 7.2 shows a logic “either-or switch” for the functions 1 and 2 as an example

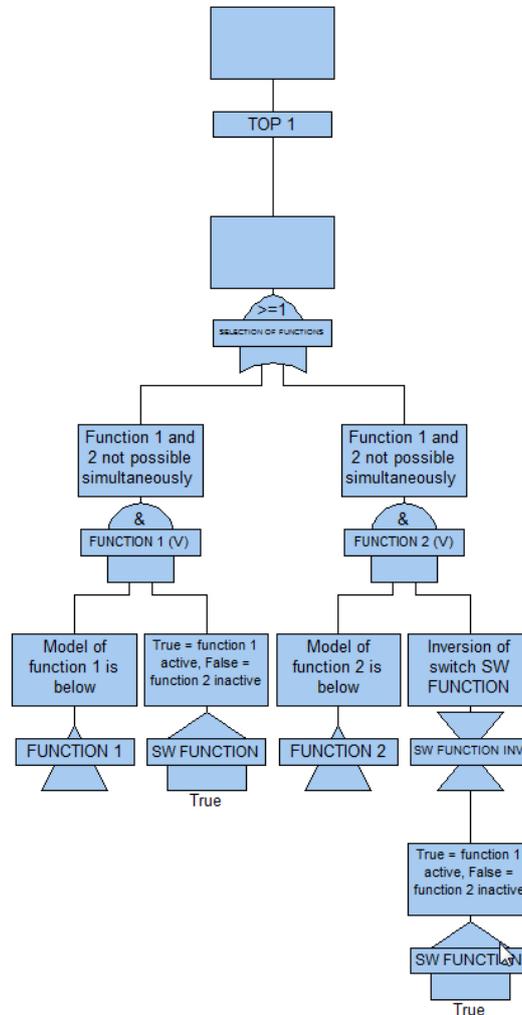


Figure 7.2: Handling variants of mutually excluding options

## 7.2. Modeling application boundary conditions:

Often functions are only active under very specific boundary conditions (e.g. warnings to the driver only when the vehicle is in motion) or faults occur only in special driving conditions (e.g. rough roads). These special cases are modeled by an AND link with one condition (= “conditional event”). If the probability of occurrence of the special case is constant with time, then the conditional event is modeled with a constant probability (Q).



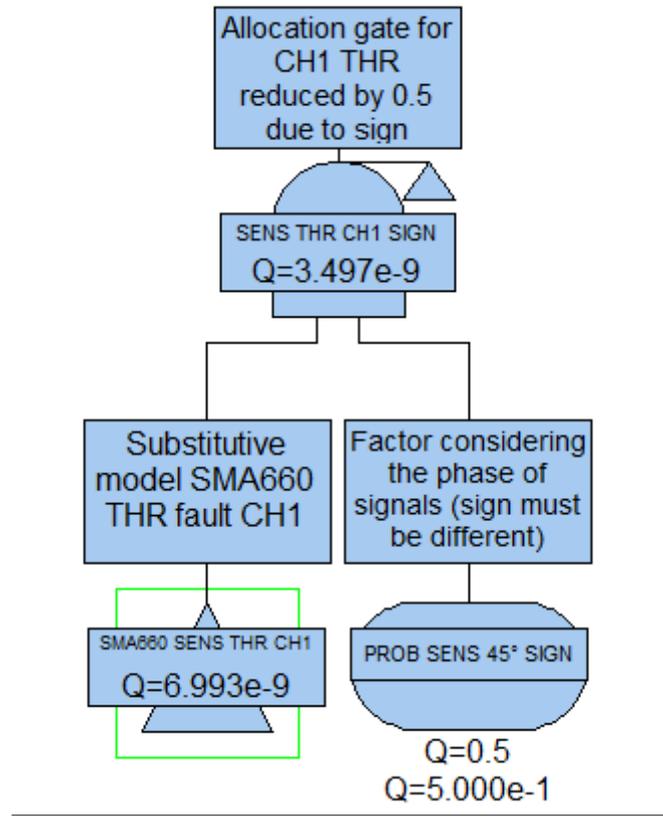


Figure 7.3: Modeling application boundary conditions

In the example given above the probability for the same or different sign of a signal (each 50%) is modeled by the oval depicted conditional event with  $Q = 0.5$ .

### 7.3. Special hints on fault tree construction for evidence per ISO 26262

The ISO 26262 suggests the use of the FTA.

- A qualitative FTA for identification of systematic faults (ISO 26262-4) – whereby all fault types (mechanical, electrical, software) should be considered (in Vol. 4 the ISO 26262 does not state that the analysis is limited to purely electrical faults).
- A quantitative FTA for evidence of the hardware metrics (ISO 26262-5) – whereby only the randomly occurring *electrical* hardware faults shall be considered.

Both requirements - contradictory in the justification - can be encountered by an appropriate data assignment of the respective basic events. By a failure rate selected appropriately low for the mechanical faults or software implementation faults, the influence can be controlled such on the one hand it can be shown in the results lists (Cut-Sets) whereas on the other hand however, has no *significant* influence on the extent of the hardware metrics. To be able to distinguish better in the composition of the computation results, different failure rates can be entered for the different fault types.

The following data assignment has asserted itself:

- Mechanical faults: Data model “Rate” where  $\lambda = 1E-15$  1/h  
(The resulting probability of the mechanical single point faults is higher than that of a double point faults → the mechanical single point fault is noticeable in a Cut-Set list)
- Software implementation fault: Data model “Fixed” where  $r = 1E-12$  1/h  
(The data model “Fixed” is selected because the probability of a software error remains constant over time.)



This data assignment is only meaningful in the context of the ISO 26262. It has been chosen intentionally and has no reference whatsoever to any field data!

### 7.4. Modeling monitoring (monitors)

The following problem often arises in the quantitative modeling of typical electrical-electronic systems: One hardware element is monitored from another element. The monitoring (the monitor) is not however perfect (degree of diagnostic coverage DC < 100%), and also because the monitoring circuit itself can fail, or the foreseen system reaction triggered from the monitor can fail.

Any possible modeling in the fault tree must therefore take into account:

- the fault to be detected
- the degree of diagnostic coverage of the monitoring (modeled using a fixed value)
- the possible comparison signals used by the monitoring
- the system reaction triggered from the monitoring that leads to the safe state

Modeling the monitoring implemented in a hardware

→ Modeling principle

Monitor function is only evaluated for the HW metrics when these are active within the fault tolerance time.

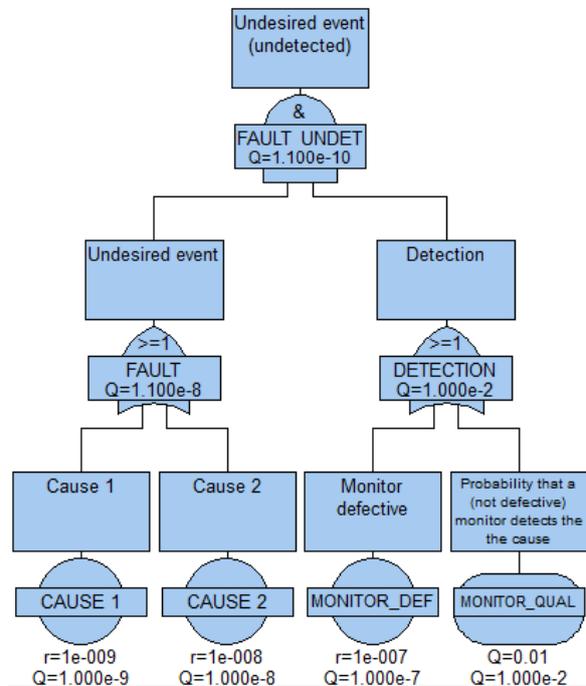


Figure 7.4: Modeling a monitoring system implemented in hardware



## Fault Tree Analysis

Modeling a “model-based” monitoring for which software could be necessary. The task of the monitoring is the comparison of a signal to be monitored (CAUSE\_1 and CAUSE\_2) with a reference signal (CAUSE\_4).

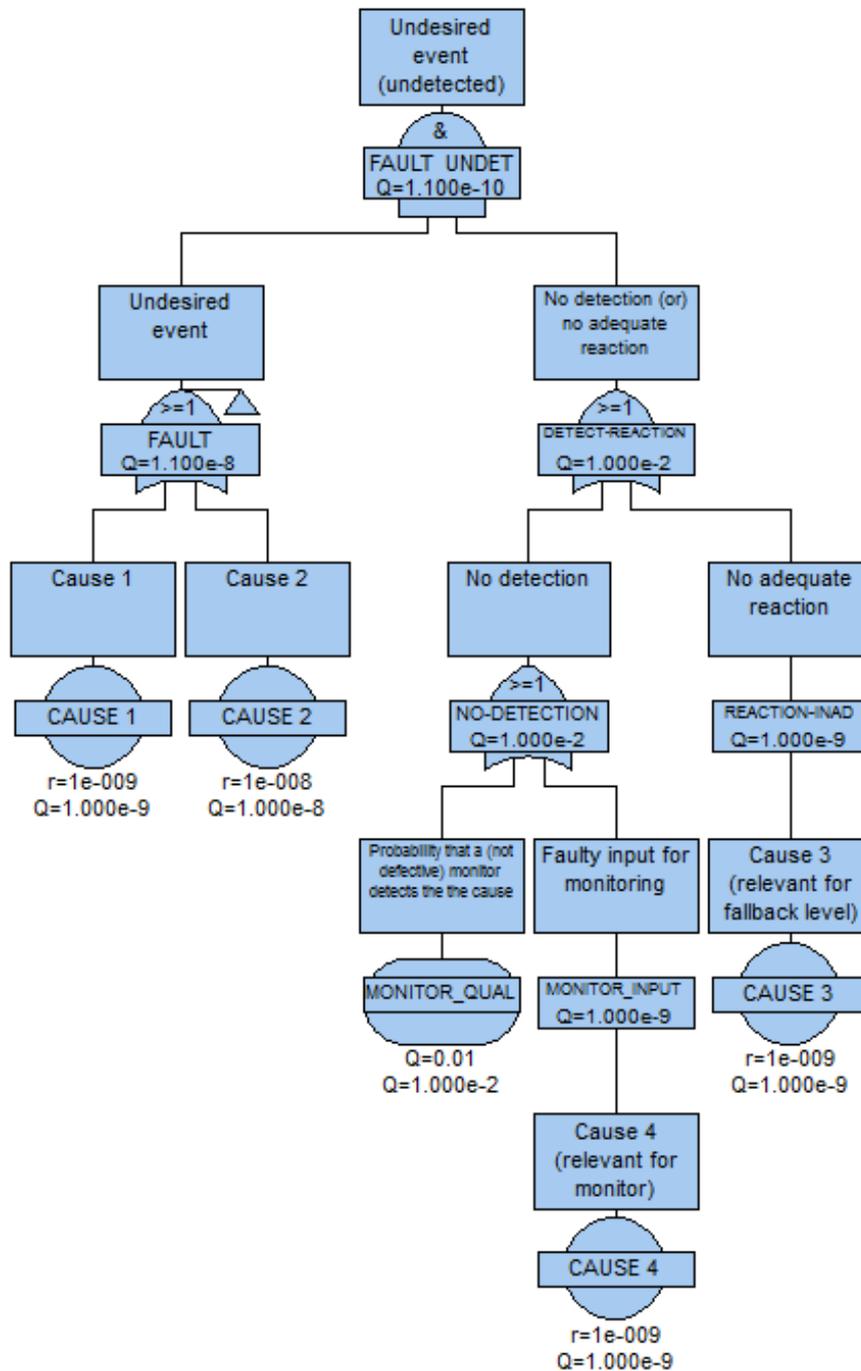
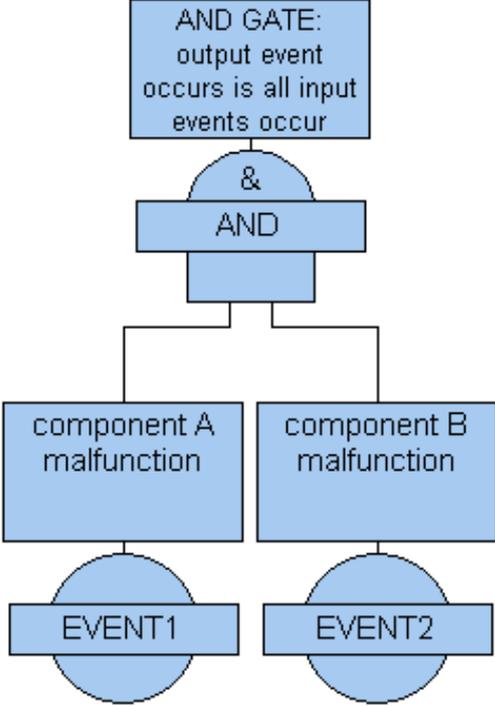
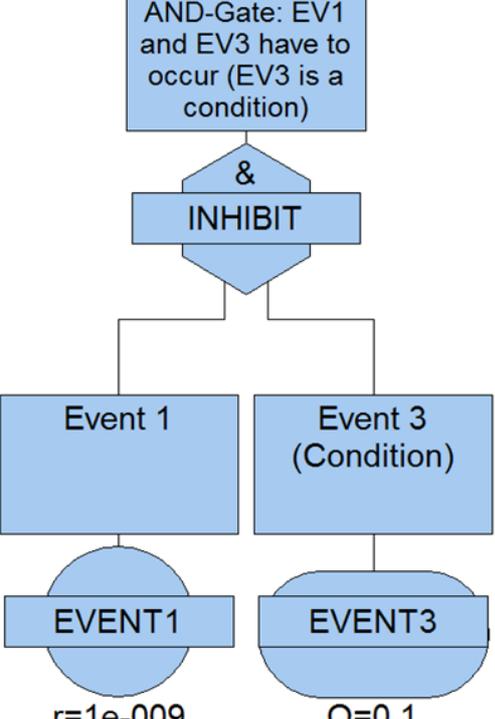


Figure 7.5: Modeling a monitoring system implemented in software

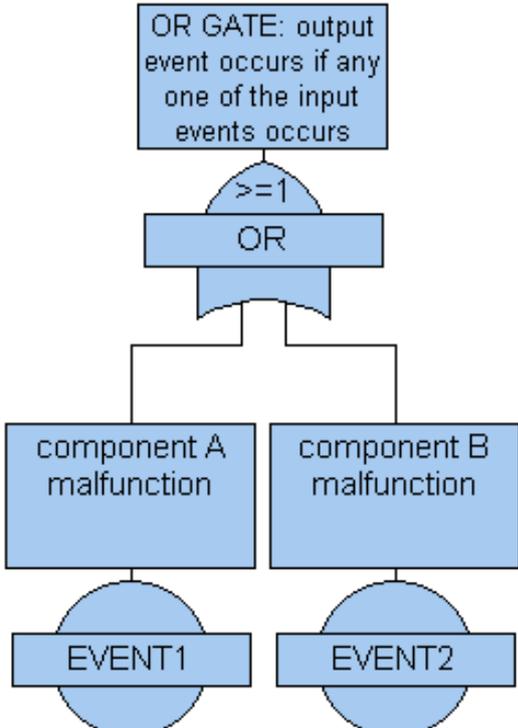
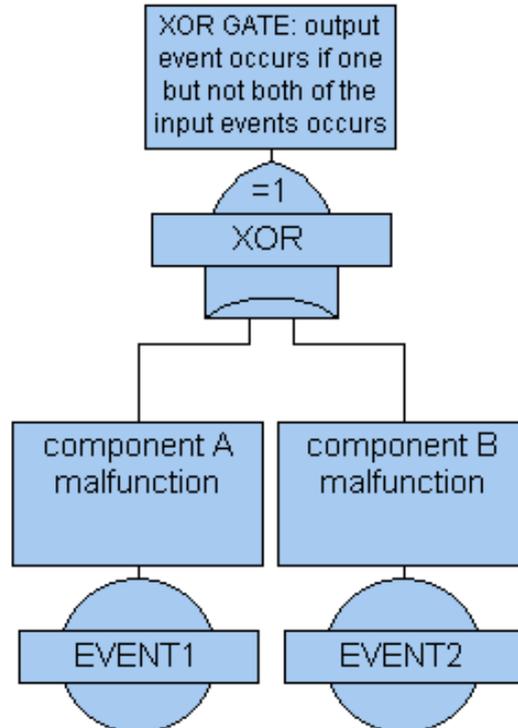


## 7.5. Overview of the event and gate types in the Tool FaultTree+

### 7.5.1. Gate types available

Type	Symbol	Remark
AND gate		<ul style="list-style-type: none"> <li>• All inputs must be TRUE.</li> <li>• <math>A \wedge B</math></li> <li>• Cut-Set list: EVENT1.EVENT2</li> <li>• Computation of Q <math>Q = q_A * q_B</math></li> <li>• In the example <math>Q = 0.1 * 0.1 = 0.01</math></li> <li>• Remark: FaultTree+ limits the maximum number of inputs to 18</li> </ul>
Inhibit		<ul style="list-style-type: none"> <li>• All inputs must be TRUE – one input represents a condition</li> <li>• <math>A \wedge B</math></li> <li>• Cut-Set list: EVENT1.EVENT3</li> <li>• Computation of Q <math>Q = q_A * q_B</math></li> <li>• In the example <math>Q = 0.1 * 0.1 = 0.01</math></li> <li>• Remark: FaultTree+ limits the maximum number of inputs to 18</li> </ul>



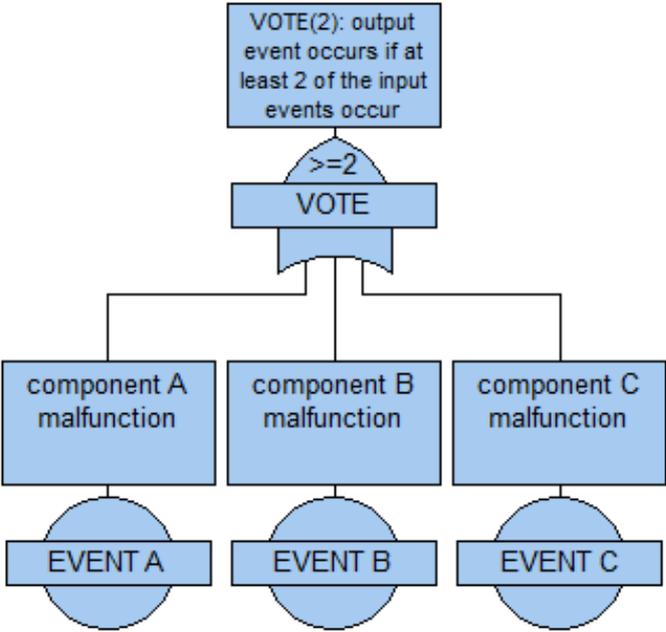
Type	Symbol	Remark
OR gate		<ul style="list-style-type: none"> <li>• One input must be TRUE</li> <li>• <math>A \vee B</math></li> <li>• Cut-Set list: EVENT1 EVENT2</li> <li>• Computation of Q (summation rule) <math>Q = q_A + q_B - q_A * q_B</math></li> <li>• In the example <math>Q = 0.1 + 0.1 - (0.1 * 0.1)</math> <math>= 0.2 - 0.01 = 0.19</math></li> <li>• Remark: FaultTree+ limits the maximum number of inputs to 18</li> </ul>
Exclusive OR gate		<ul style="list-style-type: none"> <li>• Only one input may be TRUE, the other must be FALSE.</li> <li>• <math>(-A \wedge B) \vee (A \wedge -B)</math></li> <li>• Cut-Set list: -EVENT1.EVENT2 EVENT1.-EVENT2</li> <li>• Computation of Q <math>Q = (1 - q_A) * q_B + (1 - q_B) * q_A - 0</math> <i>since (-EVENT1.EVENT2) * (EVENT1.-EVENT2) = 0</i></li> <li>• In the example: <math>Q = (1 - 0.1) * 0.1 + (1 - 0.1) * 0.1</math> <math>= 0.9 * 0.1 + 0.9 * 0.1</math> <math>= 0.09 + 0.09 = 0.18</math></li> </ul>

2020-04-06 - SOCOS



Type	Symbol	Remark
<p><b>Equivalent circuit diagram XOR</b></p>		
<p>NOT gate</p>		<ul style="list-style-type: none"> <li>• The input must be FALSE</li> <li>• -A</li> <li>• Cut-Set list: -EVENT1</li> <li>• Computation of Q <math>Q = (1 - q_A)</math></li> <li>• In the example: <math>Q = 1 - 0.1 = 0.9</math></li> </ul>



Type	Symbol	Remark
Vote gate (n from m)		<p><b>Remark:</b></p> <ul style="list-style-type: none"> <li>• n from m inputs must be TRUE. Here: 2 from 3 inputs...</li> <li>• <math>(A \wedge B) \vee (A \wedge C) \vee (B \wedge C)</math></li> <li>• Cut-Set list: EVENT1.EVENT2 EVENT1.EVENT6 EVENT2.EVENT6</li> <li>• Computation of Q (summation rule)  <math display="block">Q = q_A * q_B + q_A * q_C + q_B * q_C</math> <math display="block">- [(q_A * q_B) * (q_A * q_C) + (q_A * q_B) * (q_B * q_C) + (q_B * q_C) * (q_A * q_C)]</math> <math display="block">+ (q_A * q_B) * (q_A * q_C) * (q_B * q_C)</math> <math display="block">= q_A * q_B + q_A * q_C + q_B * q_C - [(q_A * q_B * q_C) + (q_A * q_B * q_C) + (q_B * q_C * q_A)]</math> <math display="block">+ (q_A * q_B * q_C)</math> </li> <li>• In the example the following applies...  <math>q_x * q_y = 0.01</math>  <math>q_x * q_y * q_z = 0.001</math> </li> <li>• Also:  <math display="block">Q = 0.01 + 0.01 + 0.01 - [0.001 + 0.001 + 0.001] + 0.001</math> <math display="block">= 0.03 - 0.003 + 0.001</math> <math display="block">= 0.028</math> </li> <li>• Remark: FaultTree+ limits the maximum number of inputs to 18</li> </ul>

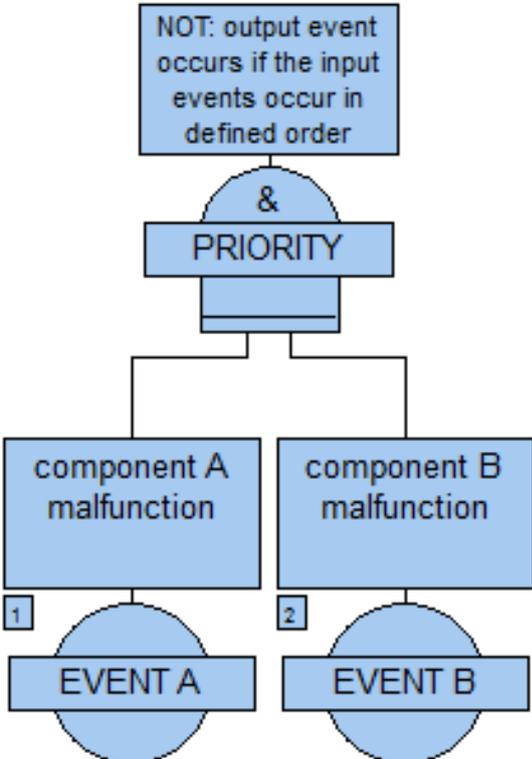
2020-04-06 - SOCOS



Type	Symbol	Remark
<p>Equivalent circuit diagram VOTE (2 from 3)</p>		
<p>Transfer gate</p>		<ul style="list-style-type: none"> <li>• Gate has no inputs or no visible inputs</li> <li>• Flagged gate where the inputs are not yet defined or those that are defined as the Top Gate of a new side in the Fault-Tree+</li> <li>• Transfer gate without a defined input can be shown in FaultTree+ using the command "Verify Data".</li> </ul>

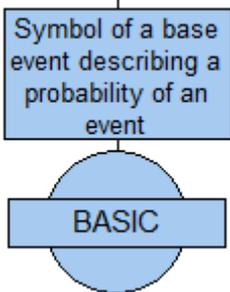
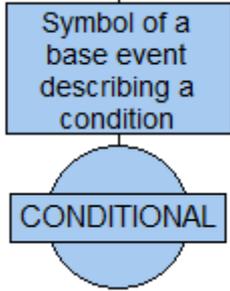
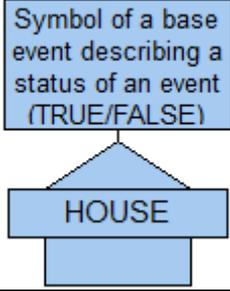
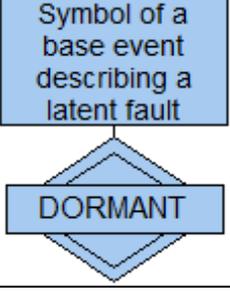
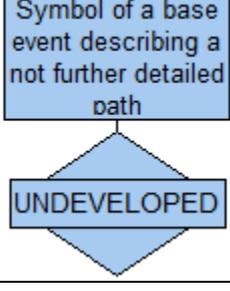
2020-04-06 - SOCOS



Type	Symbol	Remark
Zero gate		<ul style="list-style-type: none"> <li>• Gate has no logic function</li> <li>• <math>A = A</math></li> <li>• Cut-Set list EVENT1</li> <li>• Computation of Q:</li> <li>• In the example: <math>Q = q_A</math></li> <li>• Zero gates can be used for documentation purposes (e.g. documentation of the changed signal names in an active chain).</li> </ul>
Priority		<ul style="list-style-type: none"> <li>• All inputs must be TRUE in the defined sequence.</li> <li>• <math>A \wedge B</math></li> <li>• Cut-Set list: EVENT1.EVENT2</li> <li>• Computation of Q Compared to the result of the purely AND gate where the sequence plays no role, the result is reduced.</li> <li>• Refer also to the FaultTree+ Help for details of the Priority gate</li> <li>• Remark: FaultTree+ limits the maximum number of inputs to 18</li> </ul>



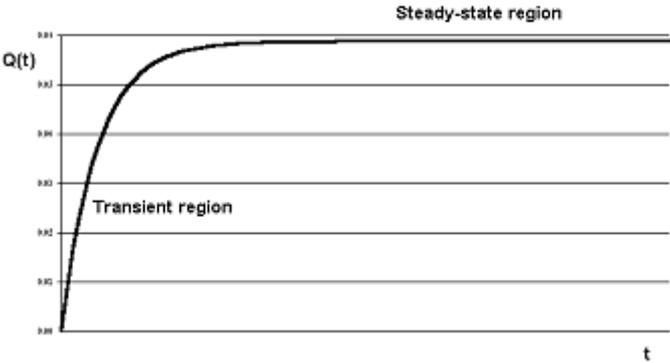
7.5.2. Available event type / event symbols

Typ	Symbol	Remark
<b>Basis</b>		<p><b>Application:</b></p> <ul style="list-style-type: none"> <li>All types of faults</li> </ul> <p><b>Typical fault models used:</b></p> <ul style="list-style-type: none"> <li>All except dormant</li> </ul>
<b>Conditional</b>		<p><b>Application:</b></p> <ul style="list-style-type: none"> <li>For taking the state in fault trees or the gap in monitoring into account</li> </ul> <p><b>Typical fault models used:</b></p> <ul style="list-style-type: none"> <li>Fixed or TRUE and FALSE</li> </ul> <p>(The rate model is only to be used when the probability of the state increases with the operating time.)</p>
<b>House</b>		<p><b>Application:</b></p> <ul style="list-style-type: none"> <li>For controlling the influence of sub-fault trees</li> </ul> <p><b>Typisch fault models used:</b> Compelling logic TRUE or FALSE</p>
<b>Dormant</b>		<p><b>Application:</b></p> <ul style="list-style-type: none"> <li>For taking latent faults into account</li> </ul> <p><b>Typical fault models used:</b> Dormant</p>
<b>Undeveloped</b>		<p><b>Application:</b></p> <ul style="list-style-type: none"> <li>For the documentation of paths not pursued further or for the documentation of inputs with and without influence</li> </ul> <p><b>Typical fault models used:</b> All except Dormant, including TRUE and FALSE</p>

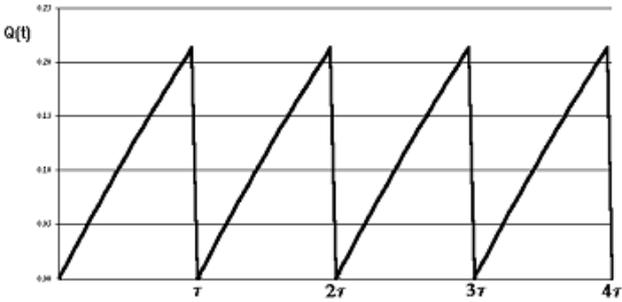
2020-04-06 - SOCCOS



7.5.3. Available fault models

Type	Application	Remark
<p>Rate model "Rate"</p>	<p>For the description of randomly occurring faults with a constant failure rate that are detected immediately or are rectified.</p> <p>Early failures and failures at the end of the lifetime are not considered.</p> <p><u>Application:</u> Electrical faults</p> <p>Can be used within in the scope of safety evidence per ISO 26262-5.</p>	<p><u>Major parameter:</u> r: failure rate lambda (1/h) t: considered time period (computing time, or also the mission time) <u>Computation (source: FaultTree+ Help):</u></p> $Q(t) = \frac{\lambda}{\lambda + \mu} (1 - e^{-(\lambda + \mu)t})$ $\omega(t) = \lambda(1 - Q(t))$ <p>where <math>Q(t)</math> = component unavailability  <math>\omega(t)</math> = component failure frequency  <math>\lambda</math> = component failure rate  <math>\mu</math> = component repair rate</p>  <p>Remark: <math>\mu</math>: Repair rate is zero in systems that cannot be repaired</p>
<p>Fixed failure rate: "Fix"</p>	<p>For the description of randomly occurring faults that occur with a constant probability.</p> <p><u>Application:</u> Operating conditions; diagnostic slip because of the principle</p> <p>Can be used within in the scope of safety evidence per ISO 26262-5.</p>	<p><u>Major parameter:</u> <math>Q(t)</math> = constant probability</p>



<p>Dormant</p>	<p>For computation of the probability of occurrence of latent occurring faults.</p> <p>Computation result depends on the project settings in the area "Sets Generation":</p> <p><u>Application:</u> Randomly occurring faults in the context of the Rate model that remain undetected over a defined time period, i.e. are present as latent faults</p> <p>Can be used within in the scope of safety evidence per ISO 26262-5.</p>	<p><u>Major parameters for systems that cannot be repaired:</u> r = failure rate lambda (1/h) Inspection interval = latency period in h</p> <p><u>Course of Q(t) for latency <math>\tau</math>:</u> The course of Q(t) as depicted below is also shown in the FaultTree+ Help. The course over time cannot however be shown in this way and only serves to explain the principle of how the computed final value shall be understood. This computed final value is used in the tool as constant Unavailability.</p>  <p><i>Q versus t plot for the dormant failure model with <math>t \ll MTTF</math></i></p> <p>Setting options for FaultTree+ in the area "Sets Generation":</p> <div data-bbox="746 1075 1165 1187" style="border: 1px solid gray; padding: 5px;"> <p>Dormant Failure Model</p> <p><input checked="" type="radio"/> Mean   <input type="radio"/> Max   <input type="radio"/> IEC 61508</p> </div> <p><u>Mean:</u> Can be used when it can be assumed that the occurrence of the latent faults is distributed equally throughout the system lifetime.</p> <p><u>Remark:</u> MTTR: "Mean Time To Repair" is zero for systems that cannot be repaired</p> <p><u>Calculation rule</u> (source: FaultTree+ Help):</p> <p>For <math>\lambda\tau \ll 1</math> and <math>\lambda \cdot MTTR \ll 1</math> it applies simplified</p> $Q_{mean} = \frac{\lambda \cdot \tau}{2} + \lambda \cdot MTTR$ <p><u>Max:</u> Can be used when it can be assumed that the occurrence of the latent fault always occurs in the first hours of the system lifetime (=&gt; conservative approach).</p> <p><u>Calculation rule:</u></p> $Q_{max} = 1 - e^{-\lambda t}$
----------------	--	--

2020-04-06 - SOCCOS



<p>Weibull</p>	<p>For the description randomly occurring faults with a variable failure rate.</p> <p>Takes early failures and failures at the end of the lifetime into account.</p> <p><u>Application:</u> E.g. mechanical faults for pronounced wear behavior</p> <p>! Do NOT use for the safety evidence per ISO 26262-5!</p>	<p><u>Failure rate:</u></p> $r(t) = \frac{\beta(t - \gamma)^{\beta-1}}{\eta^\beta}$ <p>where <math>r(t)</math> is the failure rate  <math>\eta</math> = characteristic life parameter  <math>\beta</math> = shape parameter  <math>\gamma</math> = location parameter</p> <p><u>Non-availability:</u></p> $Q(t) = F(t)$ <p>with the "Unreliability"</p> $F(t) = 1 - \exp\left[-\left(\frac{t - \gamma}{\eta}\right)^\beta\right]$
----------------	--	---

The other fault models available in FaultTree+ are not covered here in any further detail due to their minor significance of in practice.

#### 7.5.4. ISO 26262: Relationship of failure tolerance time – fault model consideration time (mission time) for continuous or initial monitoring

Within the scope of the ISO 26262 standard for the automotive field it is important for safety evidence that only those monitoring events are considered that run "fast enough" to prevent the occurrence of the Top Event by fault detection and system reaction. I.e. the sum of the execution time and fault reaction time must be less than the failure tolerance time FTTI (Fault Tolerance Time Interval) of the Top Event to be prevented.

Regarding the modeling is shall be differentiated for the quantitative interpretation whether this shall always take place only for a certain consideration time (= computing time = mission time) of 1 h (1 h is seen by many OEMs as the power-on-cycle, Case A) or whether other, longer, mission times shall be considered (Case B), e.g. 8000 h the average vehicle utilization time or 15000 h as the maximum vehicle utilization time.

##### Case A (mission time = 1 h):

In this case a non-latent basic event that experiences only an initial monitoring (i.e. once after ramping up the system) can be modeled in 2 different ways. These are mathematically equal, i.e. the magnitude of Q after 1 h is exactly the same. The initial monitoring cannot however be used here for a reduction of Q, because the fault to be detected can be active in the time between the executions (repetition period of the monitoring of 1 h >> FTTI of the Top Event to be prevented). The following modeling options are available:

1. Symbol "Basic" and fault model "rate"
2. Symbol "Basic" and fault model "dormant" with "inspection interval" = mission time. Disadvantage is that for computing time > 1 h (in case A is not relevant) the value for the Q does not increase, i.e. an inspection without a gap is assumed here. Also, the global project setting of the "dormant failure model" = MAX has to be selected so that the dormant fault model does not output any probabilities that are too small for this application purpose. All other "true" latent faults are then computed according to this dormant fault model as well (with appropriate adjustment of the inspection interval to the latency period). The MAX model has the meaning that it is assumed that all latent faults always occur within the first hours of the system running.



Latent faults that have no monitoring *whatsoever* during the vehicle lifetime have to be modeled with the fault model “dormant” and “inspection interval” = assured lifetime of the ECU (h).

**Case B (mission time > 1 h):**

Principally for the considerations in combination with the ISO 26262, computations of mission times > 1 h are not to be advised since in the final determination of the unavailability after one hour, fundamental computing errors would then be made that lead to a limiting of the maximum attainable unavailability after one hour of  $1/(\text{mission time})$ . This is because the result has to be with respect to 1 h and not to mission times > 1 h. Thus e.g. for a mission time of 10,000 h, the maximum attainable unavailability after one hour for the event occurring with certainty (i.e.  $Q = 1$  for the 10,000 h) is limited to  $Q = 1E-04$  because the mission time has to be averaged when converting back to 1 h. That this result cannot be correct is obvious. The higher the unavailability at the end of the mission time, the greater is the impact of such computing errors.

If despite the factual situation outlined above such computations have to be made then there are several options for the modeling:

1. Option: The effectiveness of the initial tests shall be taken into account:

Non-latent faults: A non-latent basic event that is only initially monitored at the beginning of a trip (with a journey time of 1 h) is marked with the symbol “Basic” and modeled with the fault model “dormant” and “inspection interval” = 1 h (project option “dormant failure model” = Max has to be selected). Otherwise it would have to be assumed that no initial monitoring has been installed in the system whatsoever or that the system is never shut down during the whole lifetime.

Disadvantages: Modeling is given that is based on 100% effectiveness of the initial test. A possible gap in the monitoring is not shown.  $Q$  is reduced by the subsequent division for these faults to values that are then too small.

Latent faults: Faults that remain latent throughout the vehicle lifetime (i.e. remain undetected and without consequences for the Top Event) should be marked with the symbol “Dormant” and be modeled with the fault model “dormant” with “inspection interval” = system lifetime. The consequence of selecting the option “dormant failure model” = max selected because of the non-latent faults is that  $Q$  is calculated as if the occurrence of the latent faults always takes in the first hour of the system lifetime.

2. Option: Effectiveness of the initial test not taken into account:

Non-latent faults:

In this case a non-latent basic event that is only monitored initially at the beginning of a trip (with journey time 1 h) is modeled with the fault model “rate”.

Effect: A very conservative modeling is given that does not take into account the weakening effect of the initial test and the shutdown of the systems after one hour of operation.

True latent faults: Latent faults that have no monitoring whatsoever during the vehicle lifetime could be modeled with the fault model “dormant”, “inspection interval” = mission time. Here also the setting of the “dormant failure models” = MAX has to be selected so that the dormant fault model does not output any probabilities for this application purpose that are too small. Effect: The unavailability computed in this way corresponds exactly to the result of modeling with the “rate” model. Latent faults hence cannot be distinguished from non-latent faults on the basis of their unavailability. According to this it makes no difference which modeling is selected. A simple general modeling with “rate” for all faults gives the same result. This is a further indication that computations of mission times > 1 h are only meaningful for systems where the period of uninterrupted operation are in fact so long (e.g. power plants and similar). Unlike this for systems with shorter mission times problems occur when making (appropriate) considerations for the influence of latent faults.

2020-04-06 - SOCOS



## 7.6. Recommendations on the naming convention

Specifying the naming conventions in the FTA is useful, amongst other reasons, for combining FTAs, ensures better readability, simplifies the interpretation and the orientation within the FTA.

### 7.6.1. Naming Events / Gates

#### Naming from general to special

It shall be taken into consideration here that the tool environment in FaultTree+ preferentially supports certain implementations of a naming convention because some dialogs (Event Table, Gate Table) of the tool always sort the available lists in alphabetical order. If wanting to always obtain similar gates side-by-side in the selection of gates is desired, then it is recommended to use a naming from general to special (example: signal error\_deviation direction: PS\_HIGH, PS\_LOW, component\_fault mode: R19\_OPEN, R19\_SHORT).

The naming convention selected in the example fault tree from Step 4 follows one order. Thus all faults are named "CAUSE\_\*".

#### Prevent special characters

The use of the decimal point "." should be avoided in the gate name / event name:

Reason: The point separates the events involved within multiple Cut-Sets in the default setting for FaultTree+ for the interpretation of the Cut-Sets. A decimal point might be confusing here.

The blank in the gate name / event name can be substituted by an underline or by a dash (better readability in FaultTree+).

#### Last character of event and gate designations

The last character of an event / gate name should not be a number or a blank.

Reason: When re-using trees or events in the mode "paste special" Fault-Tree+ adds a number to the end of the name. If there is already a number here then this number is simply incremented by Fault-Tree+ by "1" when pasting using the "paste special" function.

→ Poor recognition of unintentional changes ("Fault pressure sensor 1" unintentionally becomes "Fault pressure sensor 2").

A blank at the end of an event or gate name can lead to problems with the unambiguous identification of gates.

Example from the fault tree in Step 4:

CAUSE\_1)1 can be seen as an event not yet edited that has been added in the "paste special" mode.



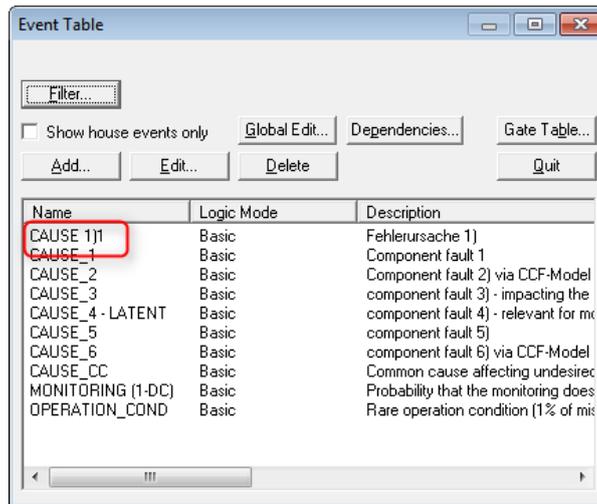


Figure 7.6: Naming convention in the event table

**Assign designations of events according to the fault type**

The event name can be made up of: Event class; name / designations; discrepancy class; signal deviation

Examples from an FTA for the product ESP (Electronic Stability Program):

Software errors:

SW\_HAL (Software, Hydraulic Actuation Layer)

SW\_VDC (Software, Vehicle, Dynamics Controller)

Mechanical faults:

M\_EV\_U\_CL (Mechanical fault, Intake Valve, Unintended, Closed)

M\_USV\_U\_O (Mechanical fault, Changeover Valve, Unintended, Open)

Control unit faults:

ECU\_EV\_U\_O (Control Unit fault, Intake Valve, Unintended, Open)

ECU\_HSV\_C2H (Control Unit fault, High-pressure Switching Valve, Current, 2(too), High)

ECU\_PHZ\_2H(>+30) (Control Unit fault, HZ Pressure Signal, Too High, deviation > 30 bar)

ECU\_PHZ\_2L(>20%) (Control Unit fault, HZ Pressure Signal, Too Low, deviation > 20%)

Signal faults:

P-WHEEL\_2L(<-15) (Wheel Pressure Signal, Too Low, deviation > 15 bar)

Monitoring gap

UNDET\_BLS\_PERM\_H (Monitoring Gap, BLS Permanently High)

UNDET\_MOM\_LOW (Monitoring Gap, Modulator Monitoring, MomLowPressure)



### 7.6.2. Use event groups

If the number of assigned events increases, then sorting the events into groups can help to retain the overview. Event groups are created in FaultTree+ in the project Explorer. Assigning an event to several event groups is also possible.

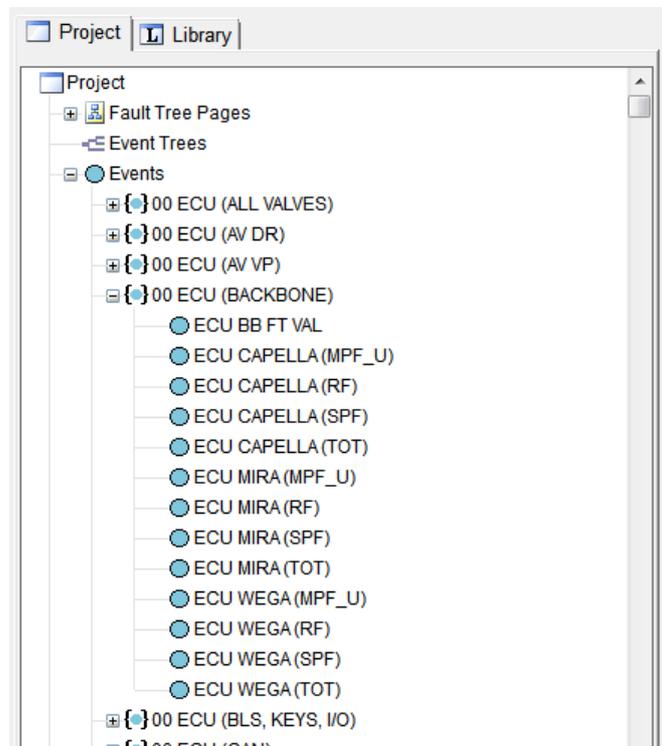


Figure 7.7: Event groups

### 7.6.3. Special feature when naming gates

If a file contains several sub-fault trees that analyze the different Top Events, then it can be meaningful when the designations of the gates include indications about the inclusion in the particular sub-fault tree. If the note is added at the end of the designation then this has the advantage that similar gates which are sorted alphabetically are listed side-by-side by FaultTree+ in different hazards.

Advantage: Especially the relationships between the gates are easier to understand. Contradictions (repeatedly used) signal-fault trees and the Top Event (the Hazard) of the fault tree can be (more) easily identified.

Example for the identification of gates within fault trees (last letters):

AV(FL) OP E|C \* → [Exhaust Valve] [(Wheel Front Left)] [Open] [Electrical or Control fault] [identification for the respective fault tree (L/ND, L/NP, R,...)]

AV(FL) OP M|E \* → [Exhaust Valve] [(Wheel Front Left)] [Open] [Mechanical or Electrical fault] [identification for the respective fault tree (L/ND, L/NP, R, ...)]



2020-04-06 - SOCOS

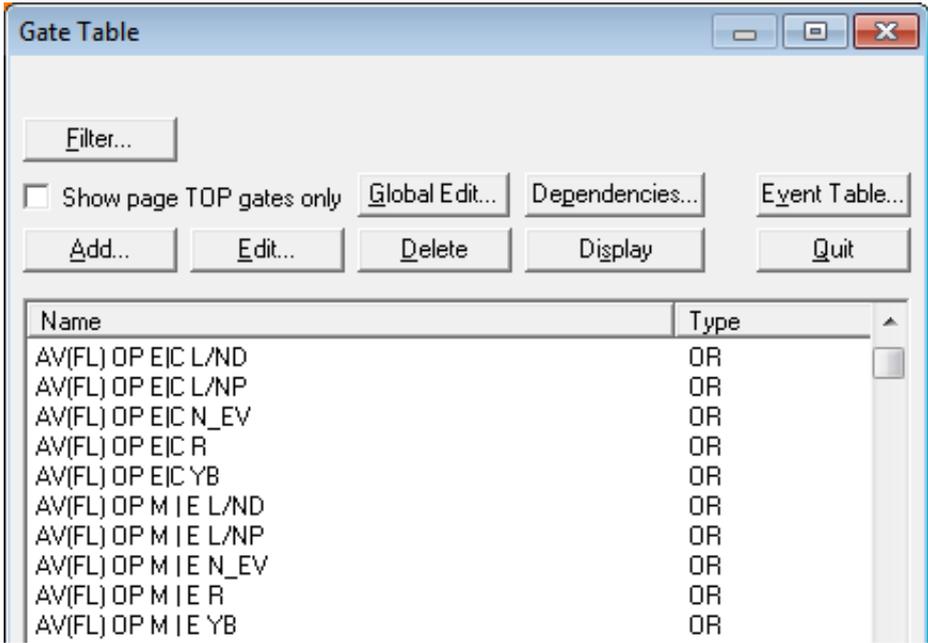


Figure 7.8: Naming convention gate table



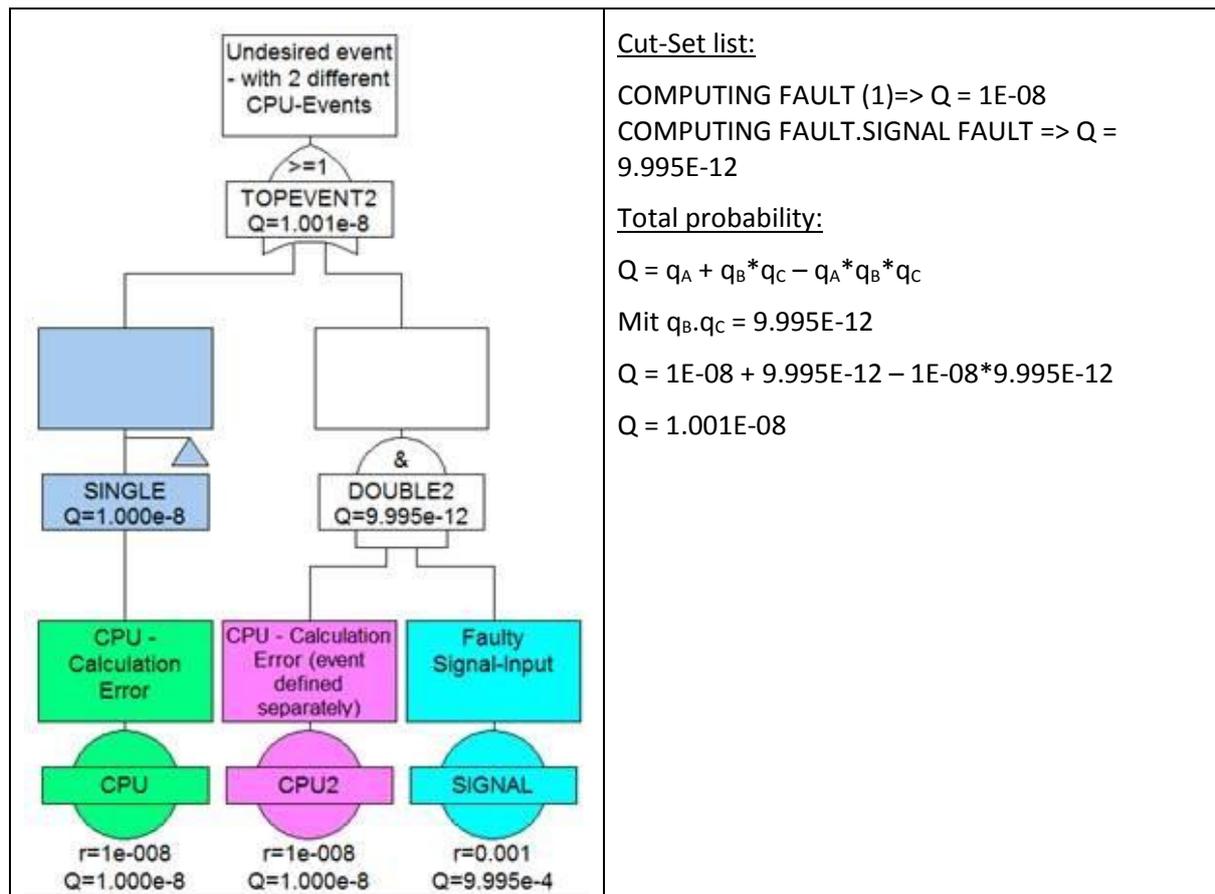
## 7.7. Hints and tricks in the preparation, computation and handling of fault trees

### 7.7.1. Multiple definition of a single basic event

If *one and the same* fault can be relevant at different points in a fault tree then attention has to be paid that the same event is always in the fault tree. This ensures that the fault tree logic computes the influence of the event concerned correctly.

Fault tree logic	Result
	<p><u>Cut-Set list:</u></p> <p>COMPUTING FAULT with <math>Q = 1.0E-08</math></p> <p>The double point fault from the COMPUTING FAULT and the SIGNAL FAULT is absorbed when using the identical event COMPUTING FAULT because:</p> $Q = q_A \vee (q_A \wedge q_B) = q_A$ <p><u>Total probability:</u></p> <p><math>Q = 1e-08</math></p>





### 7.8. Application of NOT or XOR gates for activated function “Full Not Logic”

The preparation of a fault tree when using NOT gates and / or XOR gates for an activated function “Full Not Logic” has to be well-considered. Possible application cases might be such that e.g. certain fault combinations shall be excluded by the safety logic. By an AND combination with a NOT gate or XOR invalid fault combinations shall then be expunged.

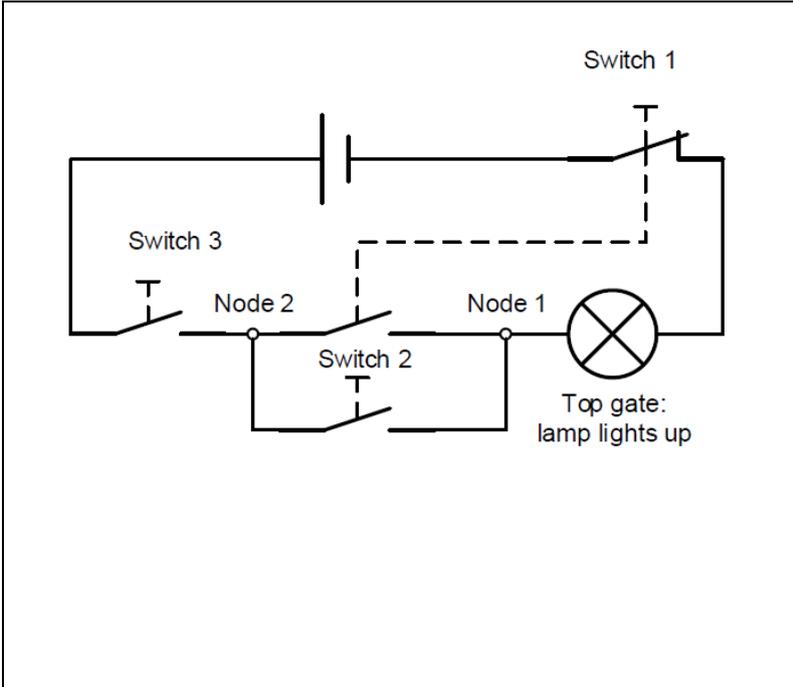
Problems arise from the type of the result that the NOT gate or the XOR gate provides. If faults are linked under these gate types the logic mode of which is neither TRUE nor FALSE, then NOT and XOR provide as the result the probability of the NOT fault (i.e. the probability that the fault does occur not just at this moment).

In the case of large fault trees this also leads to the condition - in addition to the actual relevant fault combinations – that NOT faults are combined as well. As a rule the Order of the thereby resulting Cut-Sets then increases dramatically – even though not much has changed in the magnitude of the overall result. For the corresponding logic, the Cut-Sets cannot then be displayed completely either in the tool FaultTree+ or by an export to MS EXCEL because the length of the Cut-Set string to be displayed exceeds the possibilities of both tools.

Alternatives to using NOT gates or XOR gates are:

- Avoiding the use of NOT or XOR gates underneath an AND gate and thereby knowingly accepting fault combinations where their occurrence is actually ruled out. When these are the most likely fault combinations in the overall result then this is good news because meaningful fault combinations are even more unlikely than those that are mutually excluding.
- Not to use NOT or XOR gates by cutting out the fault trees while taking into account that the relationship cannot be displayed correctly (the expunge effect from the NOT gate is explicitly taken into account in the fault tree).





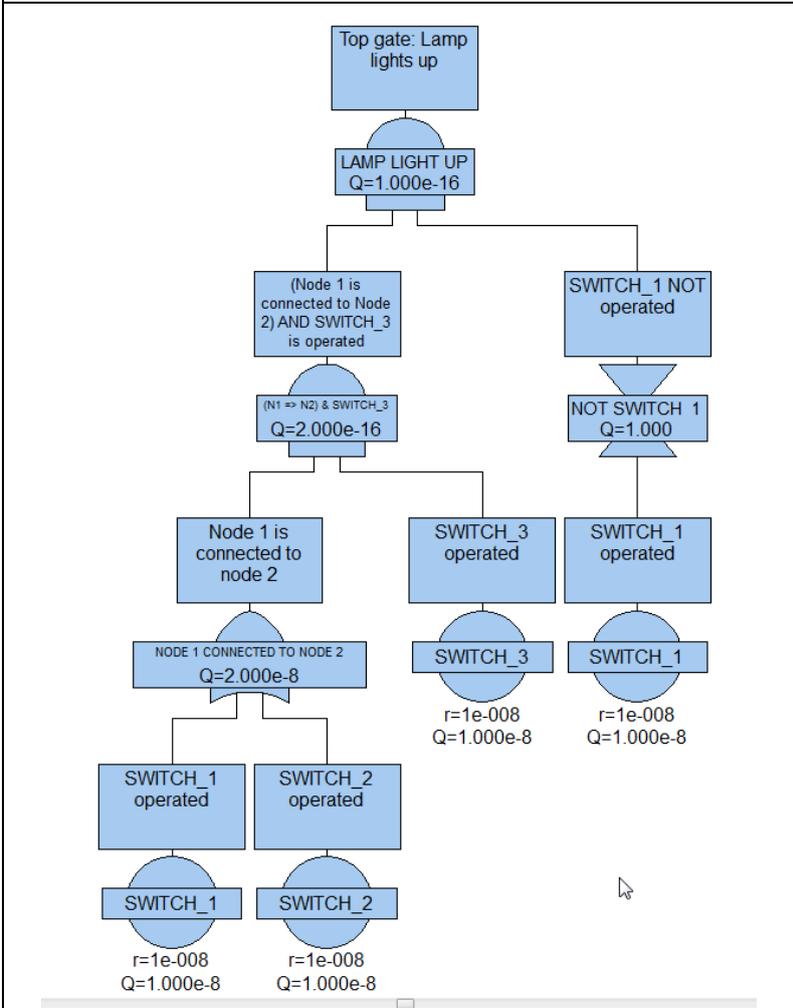
Example of a circuit diagram for lamp control.

Event to be examined: "LAMP LIGHTS UP"

Switch 1 operates two switching elements in synchronization. One of these is switched as the "Opener" in series with all other elements and as a "Closer" in parallel with the switching element that switch 2 operates.

Switch 3 operates a "Closer" and is connected in series with all switches.

Node 1 and node 2 are points where the flow of current can be analyzed.



Fault tree preparation thereby using "NOT gate":

The analysis showed:

The lamp goes on when switch\_1 is NOT operated ("Opener" not operated). The analysis of the other parts of the circuit also showed that node 1 has to be connected with node 2 connected (SWITCH\_1 or ). At the same time SWITCH\_3 must be operated.

Cut-Set:

SWITCH\_2.SWITCH\_3.-SWITCH\_1

Expunged Cut-Set:

SWITCH\_1.SWITCH\_2.-SWITCH\_1

Alternatives:

Deactivate: "Full Not Logic" in the Fault-Tree options (→ accept fault combinations)

Computing the fault tree again from the top for a deactivated "Full Fault Tree NOT logic" leads to the NOT gate or XOR gate being ignored.



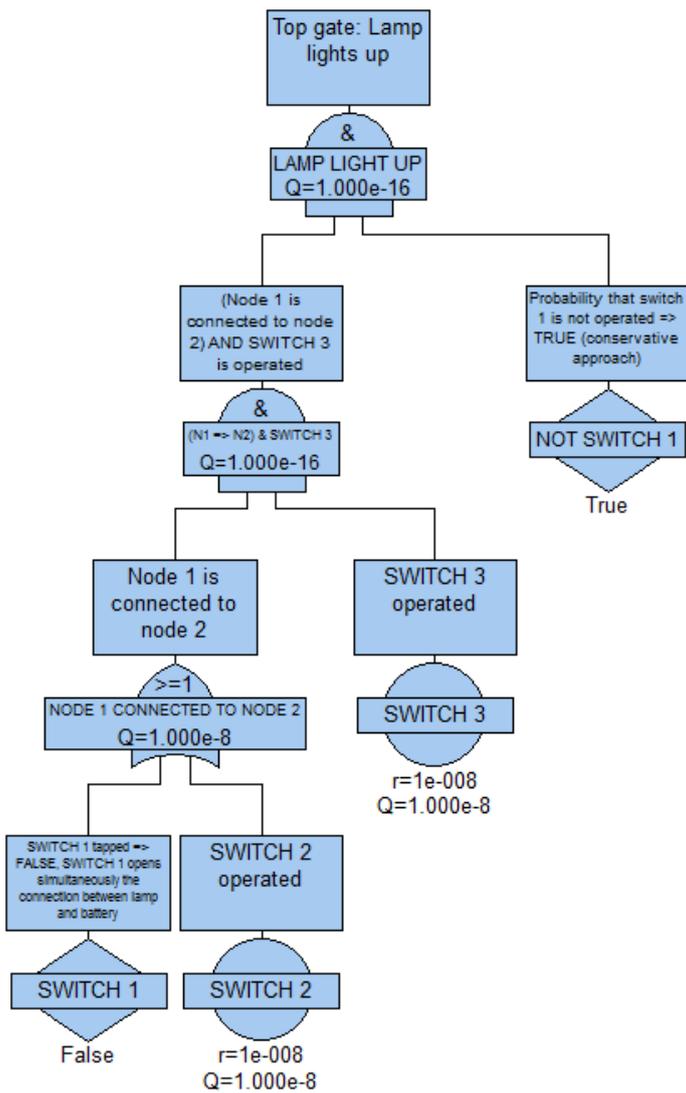
# Fault Tree Analysis

2020-04-06 - SOCCOS

The computation results of the fault tree change as follows: Cut-Set list:  
 SWITCH2.SWITCH3  
 SWITCH1.SWITCH3  
 The second Cut-Set has to be accepted here even though it is evidently invalid.



Alternative preparation of the fault tree by cutting free:



The alternative fault tree logic takes the technical relationships into account and provides a conservative (i.e. slightly too high) overall result.

The NOT gate has been converted into an Undeveloped Event “NOT SWITCH1” and set to logically TRUE. Compared to the probability of non-operation ( $Q = 1 - 1E-08$ ) this leads a slightly higher result in the Top Event ( $\rightarrow$  conservative)

The data assignment of SWITCH1 is set to logically FALSE while taking into account the contradictory fault effects (closing and at the same time opening the current circuit). This means that SWITCH1 cannot contribute to the lamp going on by a connection of node1 with node2.

Cut-Set list:

SWITCH2.SWITCH3

The probability of non-operation of SWITCH1 remains unconsidered because NOT SWITCH 1 = TRUE. This way of proceeding is meaningful for basic event where the probability is  $\ll 1$ .

If the probability is higher then the attempt can be made to take into account the reciprocal probability in a basic event that describes the non-occurrence of the fault (here “NOT SWITCH 1”).

### 7.9. Unintentional / intentional absorption of multiple point faults

With the combination of AND and OR gates, absorption effects can take place because due to the rules of Boolean algebra. These effects can be used beneficially to still be able to show for the full analysis depth the irrelevance of inputs.

Example from the system FTA of the ESP (Electronic Stability Program):

In the product ESP there is a wheel-slip controller than can only be activated by a request from the vehicle dynamics controller. In response to the request, the wheel-slip controller then compares the measured wheel velocities with a computed vehicle speed.

Assuming that the wheel-slip regulator should be unintentionally active (i.e. without a request) the following fault tree is given...



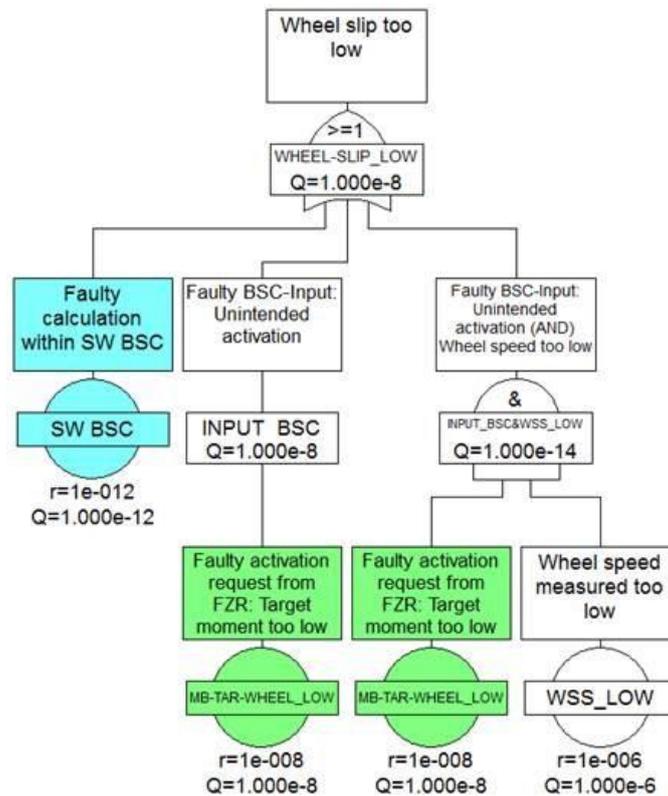


Figure 7.9: Absorption of wheel-speed faults

MB\_TAR\_WHEEL\_ZU\_NIEDRIG is negligible because of the probability larger by a factor of about 100 for WSS NIEDRIG. I.e. the wheel-velocity signal does not contribute to the total probability even though it was fully taken into account by the analysis (useful for discussions with customers with respect to complete analysis).

This absorption effect can however also take place unintentionally. Special attention shall therefore always be given to fault trees where OR gate single point faults and AND gate are combined.

### 7.10. Use of cut-off rules for the computation

Sometimes fault trees are so large that the computing times for the combinations of faults that are given are very long. Remedial help can be found by the activation of so-called Cut-Offs. In this way the computation in FaultTree+ is prematurely aborted – a corresponding loss in the accuracy of the results must however then be taken into account.

FaultTree+ offers two Cut-Off conditions (Order Cut-Off and Probability Cut-Off) that are available for the particular problem assignments. Both conditions can be combined with one another.

#### Order Cut-Off:

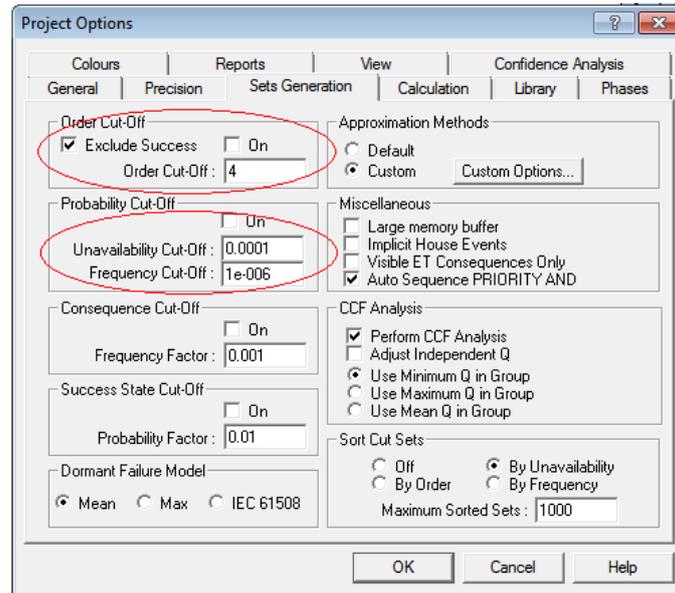
FaultTree+ only computes fault combinations up to the designated magnitude of the Order. The probability of the combination of faults does not play any role here.



## Fault Tree Analysis

### Probability Cut-Off:

FaultTree+ only computes fault combinations up to the designated Unavailability / Frequency. The Order of the fault combinations does not play any role here.

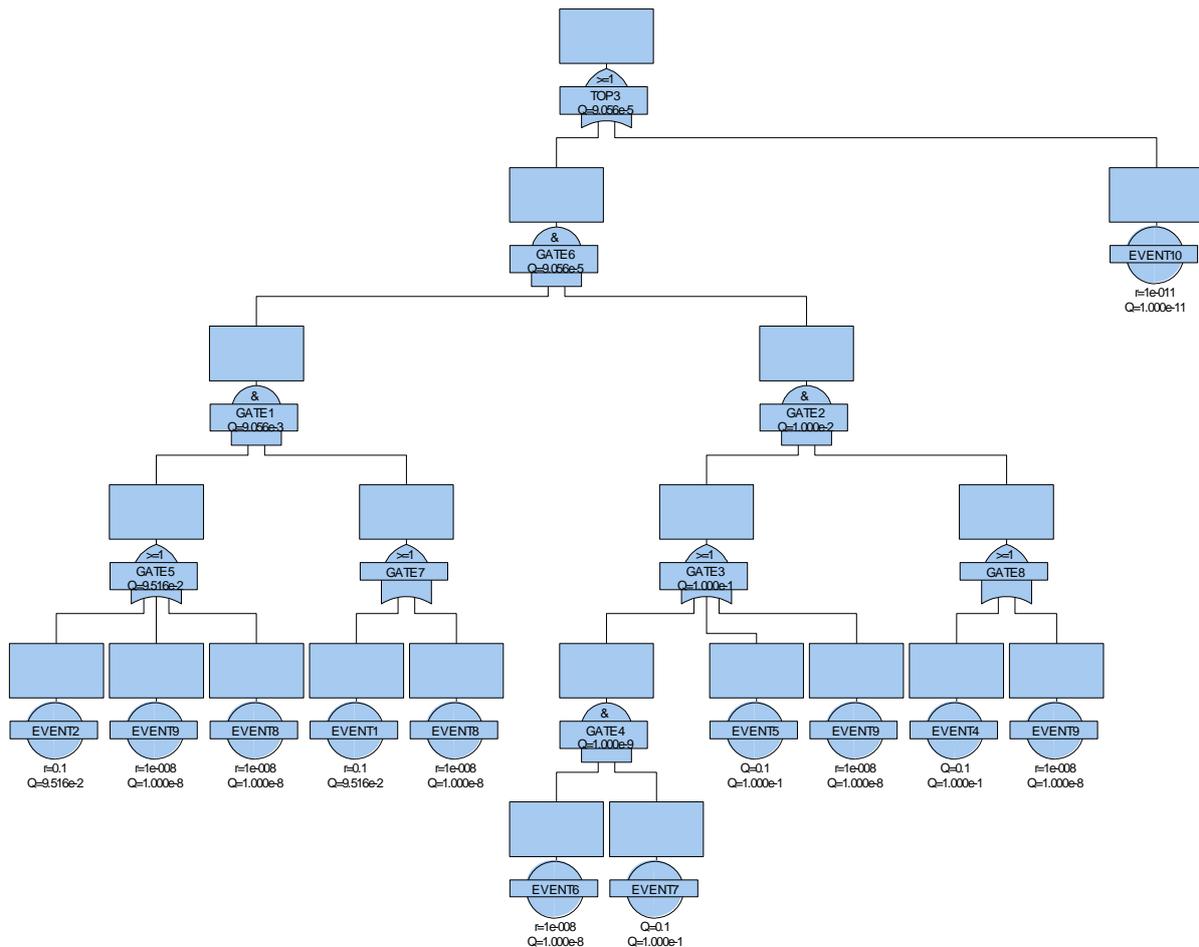


**Figure 7.10: Project options – Cut-Offs**

Inappropriate selection of the Cut-Offs can have unexpected effects on the computed result. Therefore the effectiveness of the Cut-Off conditions shall be demonstrated using an example fault tree. The logic in Figure 7.11 has no technical background, does however provide the corresponding fault combinations from single-point through to the four-point faults. A special feature here is that there are some events for which the failure probability lies in the region of 0.1.



## Fault Tree Analysis



**Figure 7.11: Example fault tree for demonstrating Cut-Offs**

The example fault tree gives the following results for a 1 h mission time ("System lifetime").

Without activated Cut-Off:

Unavailability TOP3:  $Q = 9.056 \text{ E-}05$

Cut-Set list:

Number	Cut-Set	Unavailability	Order
1	EVENT2. EVENT1. EVENT5. EVENT4	9.05592E-05	4
2	EVENT9. EVENT1	9,51626E-10	2
3	EVENT8. EVENT5. EVENT4	1E-10	3
4	EVENT10	1E-11	1
5	EVENT2. EVENT1. EVENT4. EVENT6. EVENT7	9.05592E-13	5
6	EVENT8. EVENT9	1E-16	2
7	EVENT8. EVENT4. EVENT6. EVENT7	1E-18	4

The most likely combination of faults has an Order of 4! In practice single point faults that are connected with several conditions (Q in the percentile range) can form such Cut-Sets.

The activation of a Order Cut-Off with the value 3 has the corresponding consequences:

The result is improved by a factor of 10,000 because the cut-off and presumably insignificant Cut-Set with the Order 4 provided the decisive contribution to the total result.



## Fault Tree Analysis

Unavailability TOP3:  $Q = 1.062 \text{ E-}09$

Cut-Set list:

Number	Cut-Set	Unavailability	Order
1	EVENT9. EVENT1	9.51626E-10	2
2	EVENT8. EVENT5. EVENT4	1E-10	3
3	EVENT10	1E-11	1
4	EVENT8. EVENT9	1E-16	2

Probability Cut-Off with value 1E-12:

For activation of a Probability Cut-Off with the Unavailability value 1E-12 the overall result remains almost unchanged because the difference between the Cut-Off and the expected overall result is accordingly high.

Unavailability TOP3:  $Q = 9.056 \text{ E-}05$

Cut-Set list:

Number	Cut-Set	Unavailability	Order
1	EVENT2. EVENT1. EVENT5. EVENT4	9.05592E-05	4
2	EVENT9. EVENT1	9.51626E-10	2
3	EVENT8. EVENT5. EVENT4	1E-10	3
4	EVENT10	1E-11	1

The meaningfully selected Probability Cut-Off here takes into account the magnitude where the expected result will lie – it is smaller by a factor 1E-07 (!) than overall result. (Example from practice (ESP-FTA): Target value lies in the range 1E-08 to 1E-07 => Probability Cut-Off at 1E-30).

Summary:

An *Order Cut-Off* should only be used when:

- It is certain that the result will not be falsified by the Order Cut-Off.
- The focus is in fact on the output of fault combinations with a limited Order.

A *Probability Cut-Off* can be selected when taking the expected result into account such that the overall result will not be significantly influenced.

### 7.11. Taking inputs into consideration that have no influence on a gate

It can be meaningful to model inputs in a fault tree as well when it is known that they have no influence. This can be realized by attaching basic events with the symbol “Undeveloped” and the logic mode = FALSE.

Advantages:

- The completeness of the analysis can be demonstrated and reasons for the non-relevance of inputs can be appropriately documented.
- When information is missing in the fault tree then this can be clearly interpreted.
- The comparability of fault trees increases.
- The completeness of the analysis can be easily checked.

Disadvantage: The number of events to be created increases.



## 7.12. Open points in the FTA (=> Transfer Gates, Labels etc.)

Open points in the FTA can be flagged by using labels on FaultTree pages as well by including gates without any input (Transfer Gates) in the fault tree.

Drawback of using labels: Labels are always assigned to one page in the fault tree. If the allocation of the pages in the fault tree changes (pages are defined by the option “Page” in gates defined), and hence the allocation of the label (of the open points) to the fault tree is lost.

Advantage of using transfer gates: The Option “Verify Data” in the menu “Analysis” allows straightforward identification of such transfer gates. If this query does not provide any more open gates then all open points are closed.

## 7.13. Modeling Common-Cause Failures

There are principally 2 options for the modeling:

1. Modeling with a coupling factor between basic events that can have a common Root Cause failure ( $\beta$ -factor model).
2. Modeling by using the Root-Cause event at all relevant points in the fault tree.

### 7.13.1. Modeling with the $\beta$ -factor model

The  $\beta$ -factor is the simplest model for the description of the dependency of two events A and B that are initially assumed to be independent.

With the help of a factor (the  $\beta$ -factor that can vary between 0% and 100%) an expression is given for the proportion of the failure rate of A that leads to the same type of fault for B. In the most straightforward case of homogeneous redundancy where  $Q_A = Q_B$  it the following applies:

$$Q_{CCF} = \beta \times Q_A$$

$$Q_{TOP} = Q_{CCF} + (1 - \beta)Q_A * (1 - \beta)Q_B$$

Example:  $Q_A = Q_B = 1E-03$ .  $\beta = 10\%$

$$Q_{TOP} = 0.1 * 1E-03 + 0.9 * 1E-03 * 0.9 * 1E-03 \approx 1.0081E-04$$

For other cases refer to the FT+ User Manual.

The  $\beta$ -factor model is not undisputed since there is no generally accepted method for determining the  $\beta$  factor. The norm IEC61508 includes a checklist for determining the  $\beta$  factor that is often used.

### 7.13.2. Modeling by using the Root-Cause event

When modeling Common Causes by using the Root-Cause attention has only to be given that the same event is being linked.

Example: In Figure 7.11 (see Section: 7.10 Use of cut-off rules for the computation) the EVENTS 1 and EVENT 9 are linked as Root-Cause events for the higher precedence AND gate.



## 8. Attachment 2 – example of a report

2020-04-06 - SOCOS


**BOSCH**

From	Our Reference	Tel	Location Date
------	---------------	-----	------------------

**Report**

Issue **X.X**  
 Topic **PROJECT**  
 Description **FTA-Report**

**1. Task:**

Creation of a Fault Tree Analysis (FTA) with the following FTA Top Events for project „X.X“:

TopEvent-Name	Description	Comment / Note
TopEvent1	XXX	XXX
TopEvent2	XXX	XXX
TopEvent3	XXX	XXX
TopEvent4	XXX	XXX

**2. Work group:**

Name	Departement	Task in FTA-Team	Comment / Note
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX

**3. Results:**

The analysis provided the following results for each TopEvent:

TopEvent-Name	(if applicable) ASIL	Target	Result value	Comment / Note / Rating
TopEvent1	XXX	XXX	XXX	XXX
TopEvent2	XXX	XXX	XXX	XXX
TopEvent3	XXX	XXX	XXX	XXX
TopEvent4	XXX	XXX	XXX	XXX

Page 1 of 3





From | Our Reference | Tel | Location  
Date

Report  
Issue X.X  
Topic PROJECT

The following Basic Events / Minimal Cuts have a relevant influence on each TopEvent:

<i>TopEvent1</i>	<i>Description</i>	<i>Result value</i>
<i>Basic Event / Minimal Cut1</i>	<i>Description</i>	<i>Result</i>
<i>Basic Event / Minimal Cut2</i>	<i>Description</i>	<i>Result</i>
<i>Basic Event / Minimal Cut3</i>	<i>Description</i>	<i>Result</i>

**4. Assumptions made:**

Following assumptions have been made during the FTA creation:

<i>Assumption</i>	<i>implementation</i>
<i>Scope</i>	<i>e.g. variant / placement / ...</i>
<i>Limitation of system</i>	<i>...</i>
<i>...</i>	<i>...</i>

**5. Data base**

The results are based on the following derived failure rates...

- *Example source: „Siemens-Norm“ SN29500*
- *...*
- *Include table of events and their failure rates here or give a note of table included in the attachment (Export of Events)*





From | Our Reference | Tel | Location  
Date

Report  
Issue X.X  
Topic PROJECT

6. Attachement (Examples)

- Block chart
- FTA chart
- List of defined FTA gates („GATE“)
- List of defined FTA Basic Events („Events“)
- Printout "open point list"
- Printout meeting / presence / participant lists

7. Release

*If a release of this report (and of associated FTA) is desired in each project, at this point a list of each person who have to sign can be inserted.*

*The workflow itself – comparable to FMEA signature workflow – can be done with „eSignature“.*



Blank page

2020-04-06 - SOCOS



Blank page

2020-04-06 - SOCOS



Blank page

2020-04-06 - SOCOS



Blank page

2020-04-06 - SOCOS



**Robert Bosch GmbH**

C/QMM

Postfach 30 02 20

D-70442 Stuttgart

Germany

Phone +49 711 811-71 39

Fax +49 711 811-4 51 55

m/'' [www.bosch.co](http://www.bosch.co)

