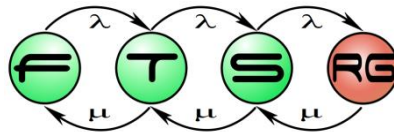


# Safety-critical systems: Evaluation

Systems Engineering course

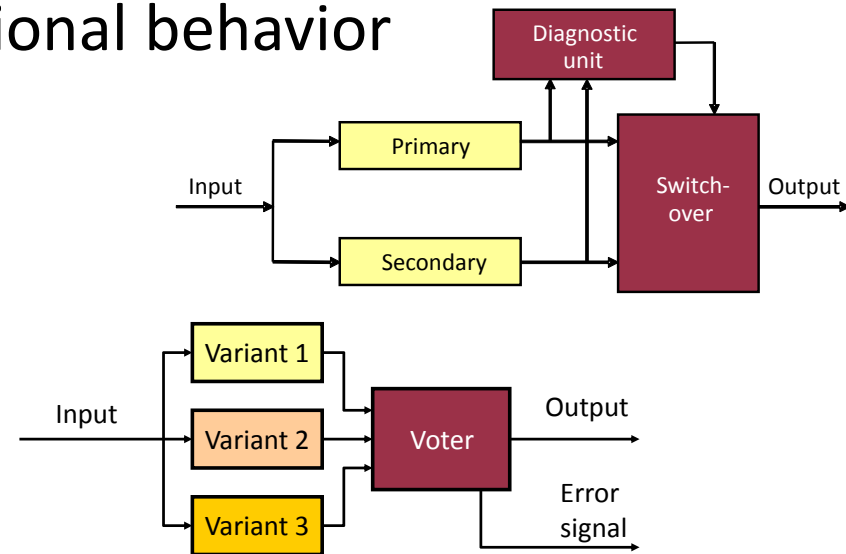
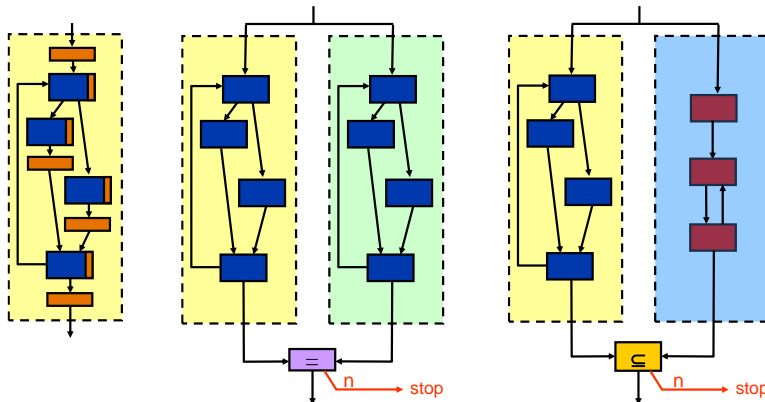
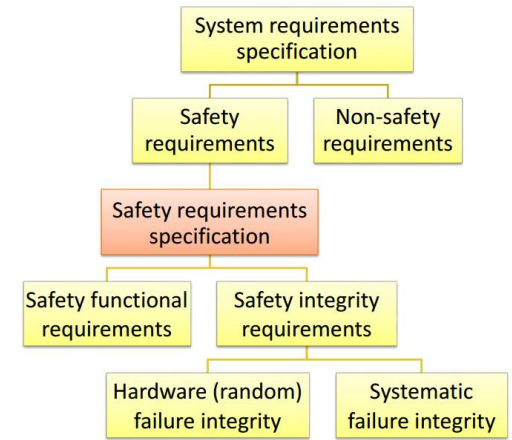
András Vörös  
(slides: István Majzik)



# Overview of the goals

# Previous topics

- Specification in safety-critical systems
  - Safety function requirements
  - Safety integrity requirements
  - Dependability requirements
- Architecture design (patterns)
  - Error detection for fail-stop behavior
  - Fault tolerance for fail-operational behavior



# Goals

- Safety critical systems study block
  1. **Requirements** in critical systems: Safety, dependability
  2. **Architecture design** (patterns) in critical systems
  3. **Evaluation** of system architecture
  
- Focus: Evaluation of the system architecture to ...
  - Analyze the **causes** of potential hazards
  - Analyze the **effects** of component faults
  - **Estimate risk**: Hazards with **rate** (probability) and **severity**
    - check with respect to tolerable hazard rate (THR)
  - Calculate **reliability** and **availability**

# Learning objectives

## Evaluation of hazards and fault effects

- Understand the role of architecture evaluation
- Know the typical techniques for the analysis
- Understand the method of risk estimation
- Perform evaluation of a concrete architecture

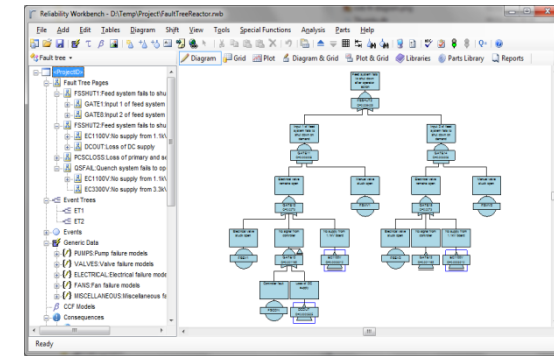
## Evaluation of reliability and availability

- Know the reliability block diagram technique
- Understand the limitations of the technique
- Perform evaluation in canonical systems

# Overview: Evaluation techniques

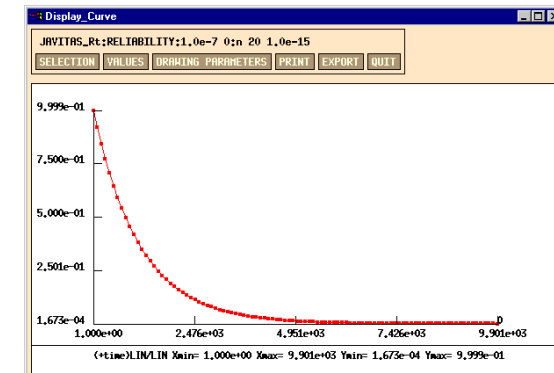
- Systematic analysis of hazard causes and fault effects (with risk estimation):

- **Fault tree analysis (FTA)**
- **Event tree analysis (ETA)**
- **Failure modes and effects analysis (FMEA)**

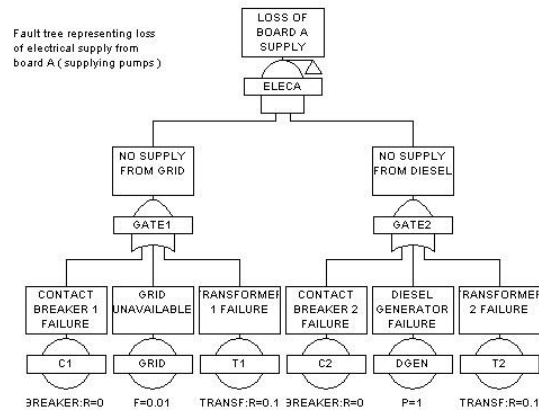


- Quantitative reliability analysis:

- **Reliability block diagram (RBD)** based calculation

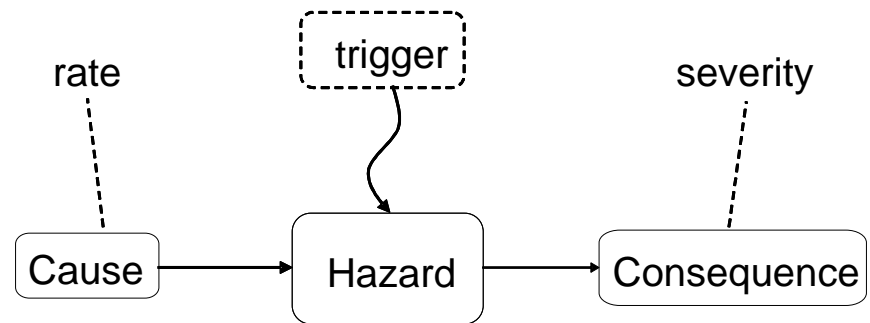


# Fault tree analysis



# Introduction: Hazard analysis

- Goal: Analysis of the **fault effects** and the evolution of **hazards**
  - What are the **causes** for a hazard?
  - What are the **effects** of a component fault?
- Results:
  - Categorization of hazards
    - **Rate** of occurrence
    - **Severity** of consequences
  - Hazard catalogue
  - **Risk matrix**
- These results form the basis for **risk reduction**





# Categorization of the techniques

- On the basis of the development phase (tasks):
  - **Design phase**: Identification and analysis of hazards
  - Delivery phase: Demonstration of safety
  - Operation phase: Checking the modifications
- On the basis of the analysis approach:
  - Cause-consequence view:
    - **Forward (inductive)**: Analysis of the **effects** of faults and events
    - **Backward (deductive)**: Analysis of the **causes** of hazards
  - System hierarchy view:
    - **Bottom-up**: From the components (subsystems) to system level
    - **Top-down**: From the system level down to the components
- **Systematic** techniques are needed

# Fault tree analysis

Analysis of the **causes** of system level hazards

- **Top-down** analysis
- Identifying the component level **combinations** of faults/events that may lead to hazard

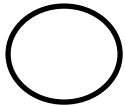
Construction of the fault tree

1. Identification of the foreseen **system level hazard**: on the basis of environment risks, standards, etc.
2. Identification of **intermediate events (pseudo-events)**: Boolean (AND, OR) combinations of lower level events that may cause upper level events
3. Identification of **primary (basic) events**: no further refinement is needed/possible

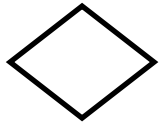
# Set of elements in a fault tree



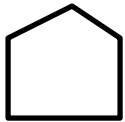
Top level or intermediate event



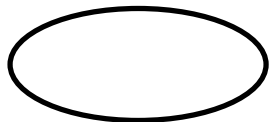
Primary (basic) event



Event without further analysis



Normal event (i.e., not a fault)



Conditional event

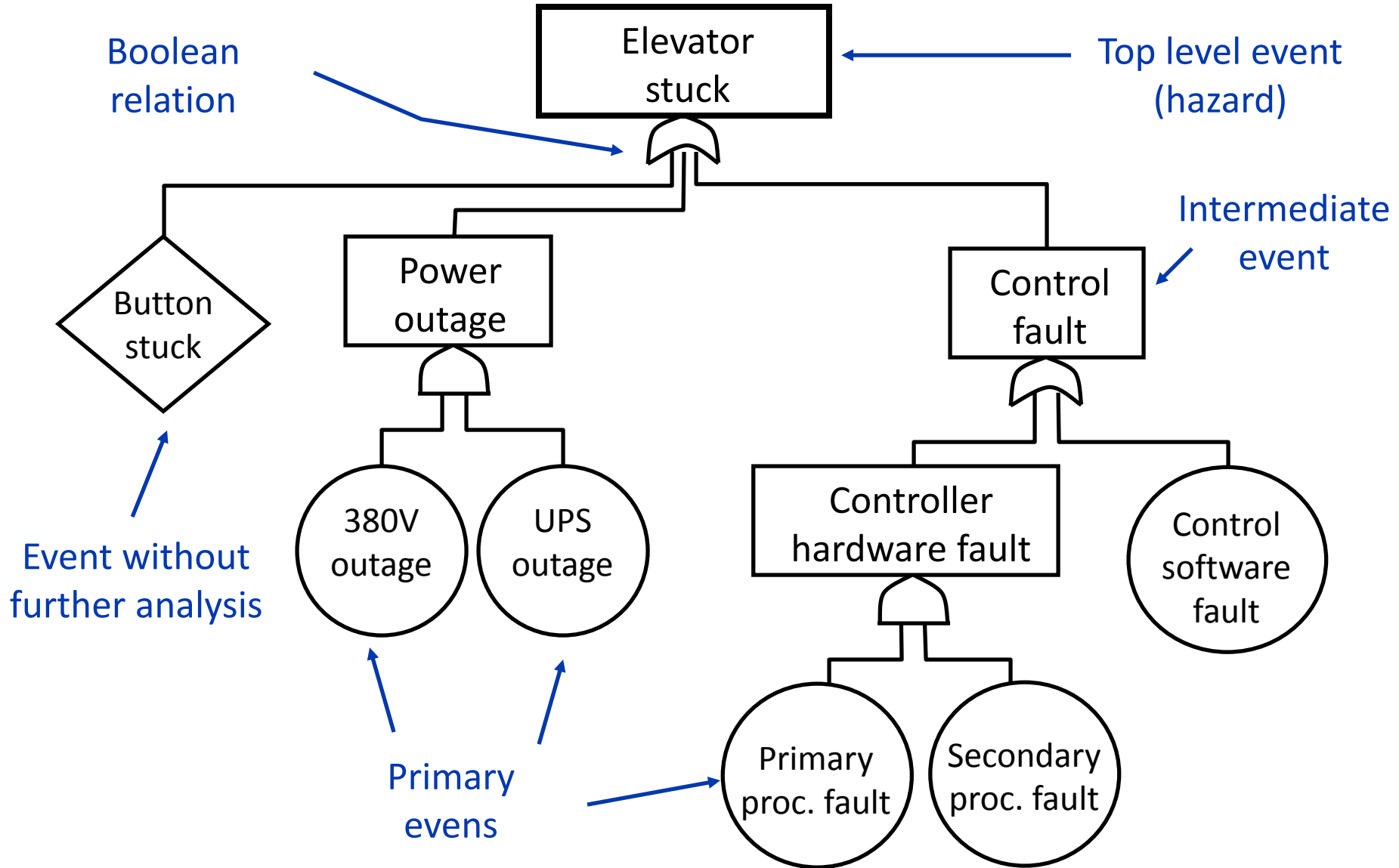


AND combination of events



OR combination of events

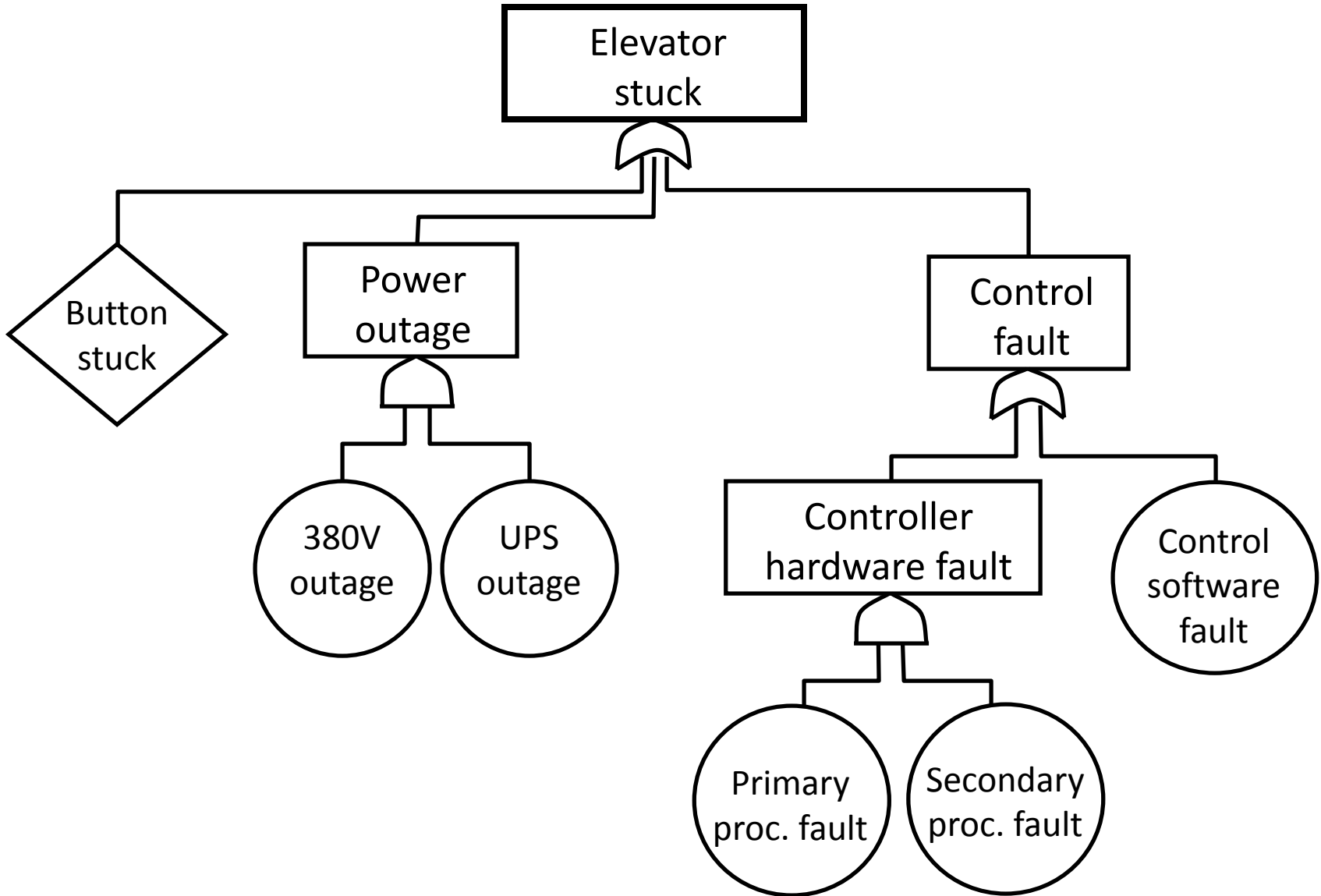
# Fault tree example: Elevator



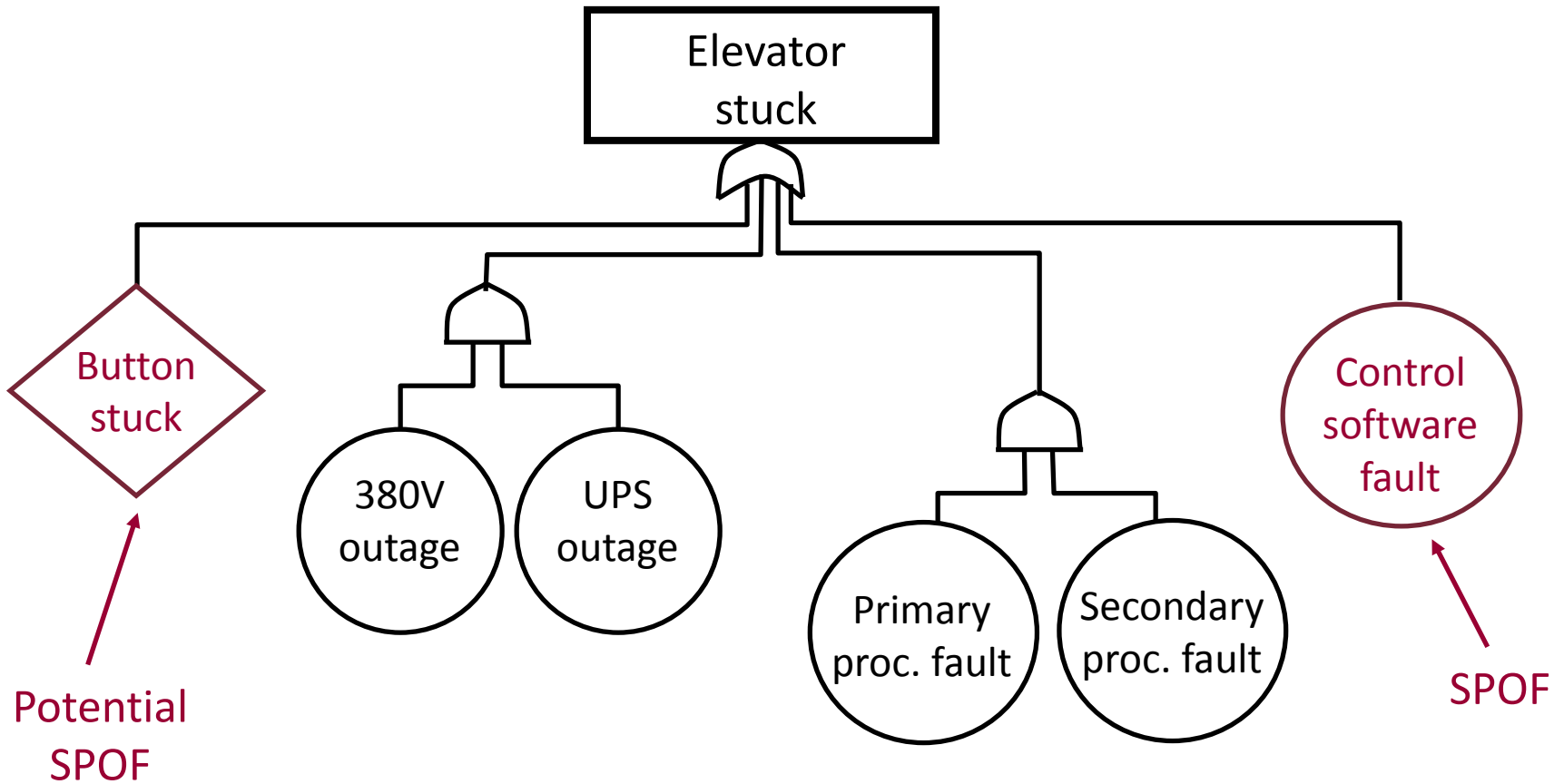
# Qualitative analysis of the fault tree

- Fault tree **reduction**: Resolving intermediate events/pseudo-events using primary events  
→ **disjunctive normal form** (OR on the top of the tree)
- **Cut** of the fault tree:  
AND combination of primary events
- **Minimal cut set**: No further reduction is possible
  - Minimal cut: There is no other cut that is a subset
- Outputs of the analysis of the reduced fault tree:
  - **Single point of failure** (SPOF)
  - Critical events that appear in several cuts

# Original fault tree of the elevator example



# Reduced fault tree of the elevator example

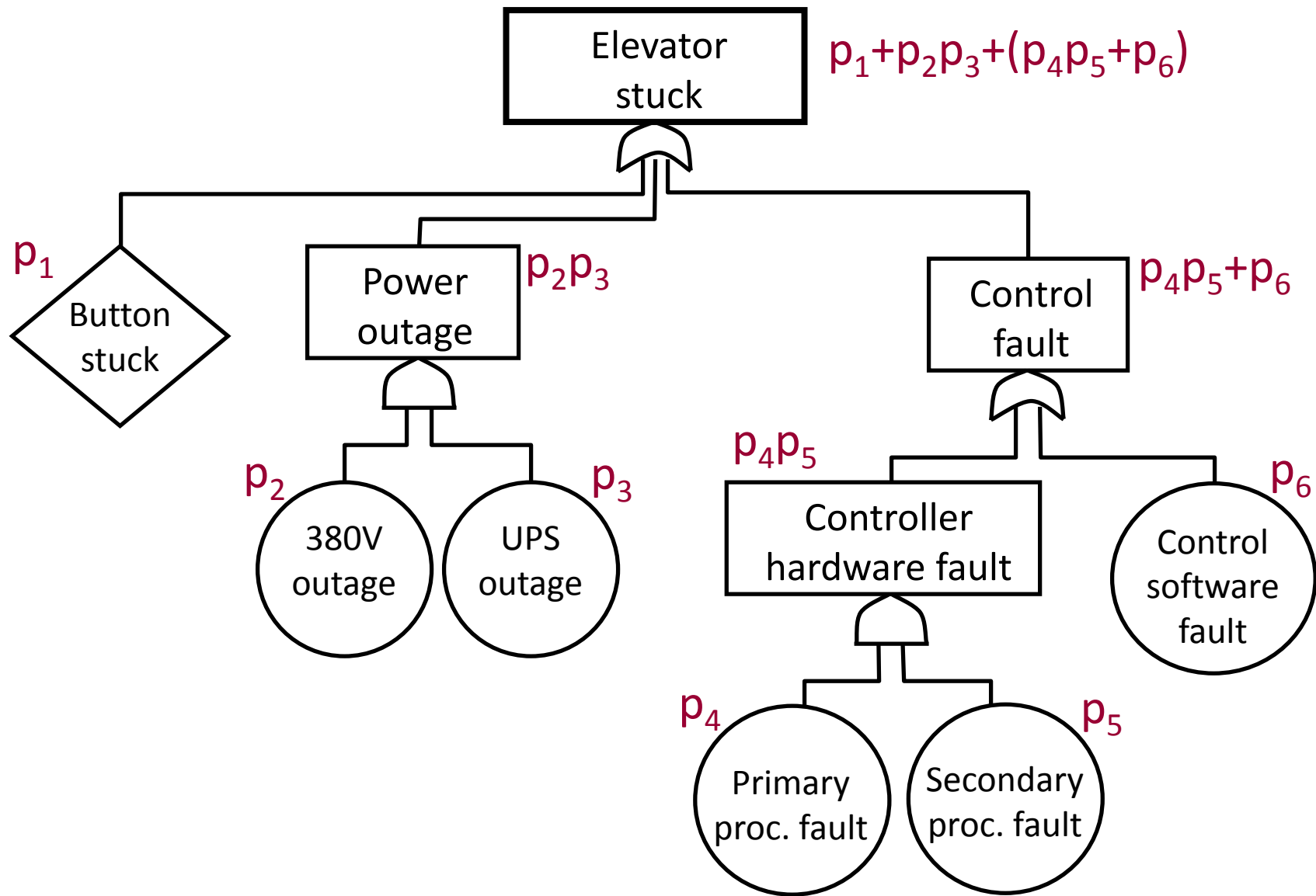


# Quantitative analysis of the fault tree

- Basis: **Probabilities** of the primary events
  - Component level data, experience, or estimation
- Result: Probability of the **system level hazard**
  - Computing probability on the basis of the probabilities of the primary events, depending on their combinations
  - AND gate: **Product** (if the events are independent)
    - Exact calculation:  $P\{A \text{ and } B\} = P\{A\} \cdot P\{B | A\}$
  - OR gate: **Sum** (worst case estimation)
    - Exactly:  $P\{A \text{ or } B\} = P\{A\} + P\{B\} - P\{A \text{ and } B\} \leq P\{A\} + P\{B\}$
  - Probability as time function can also be used in computations (e.g., reliability, availability)
- Typical problems:
  - Correlated faults (not independent)
  - Handling of fault sequences



# Fault tree of the elevator with probabilities



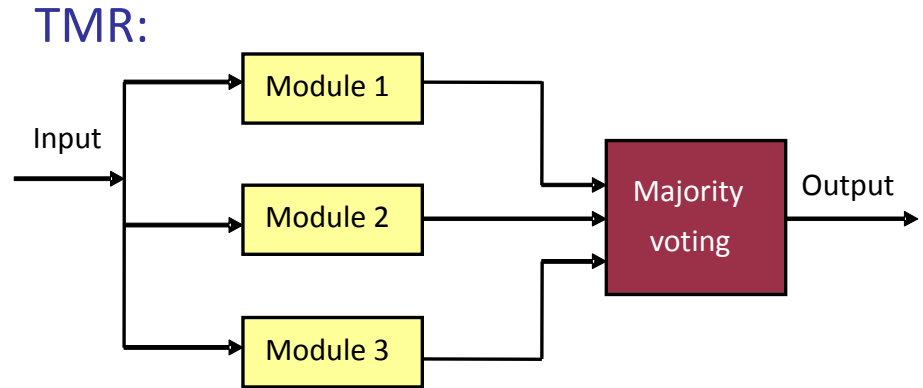
# Exercise: Evaluation of an intrusion detection system

The intrusion detection system of a flat includes as detectors a **door opening sensor**, a **pressure detector** on the floor and a **sound detector** with an analogue **sound filter**.

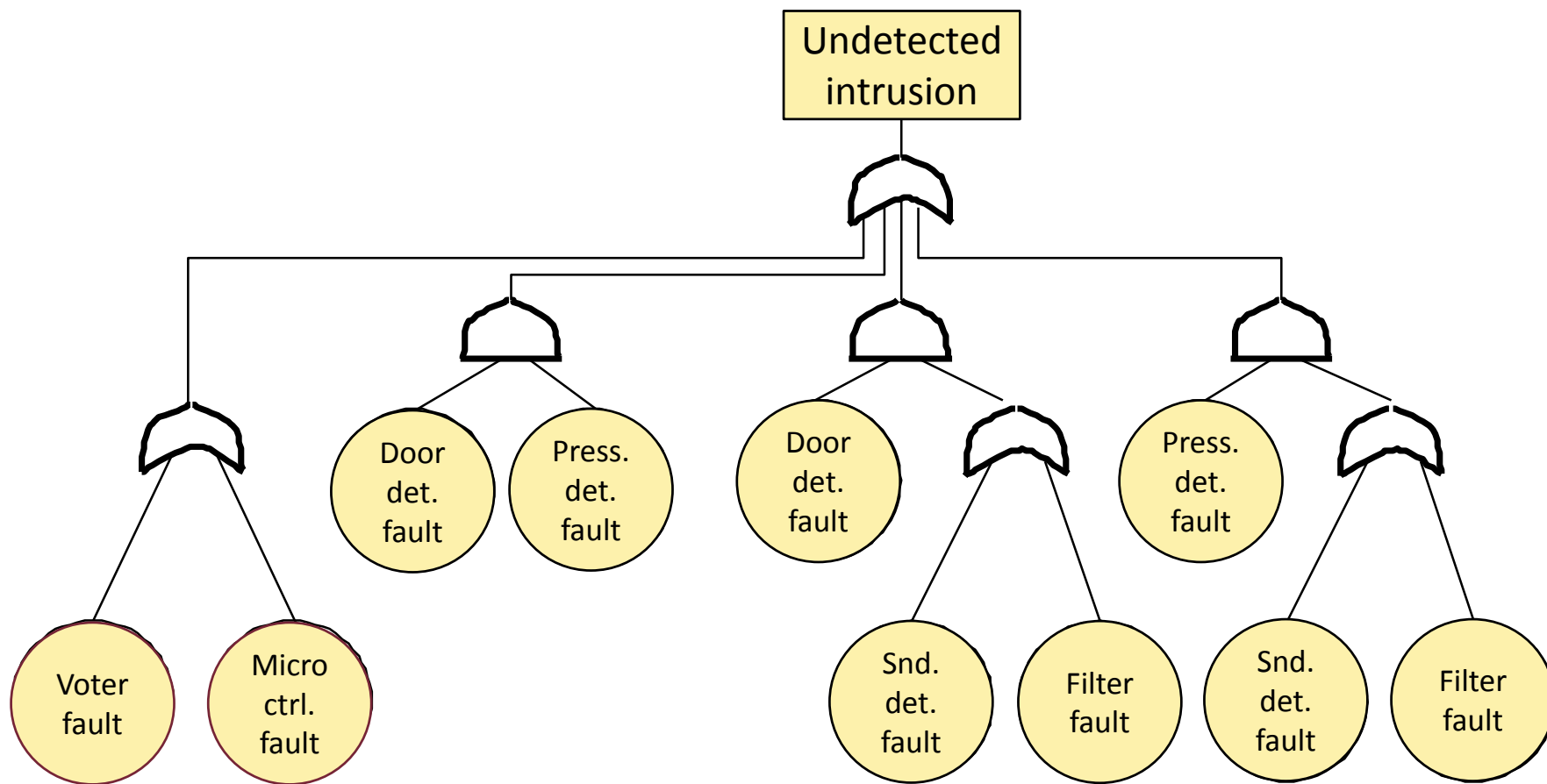
These detectors are operated in a **TMR structure** with a **voter** component that is implemented using a **microcontroller**.

## Exercise:

- Draw up the **fault tree** that belongs to the **undetected intrusion** as the top level hazard. The basic events are the faults of the above mentioned components (these faults are considered as independent).
- Indicate the **single point of failure** (if any).
- Is it possible to implement the **recovery block structure** on the microcontroller in order to tolerate the faults of the detectors?



# Solution of the exercise



Single point of failure: Voter fault, microcontroller fault

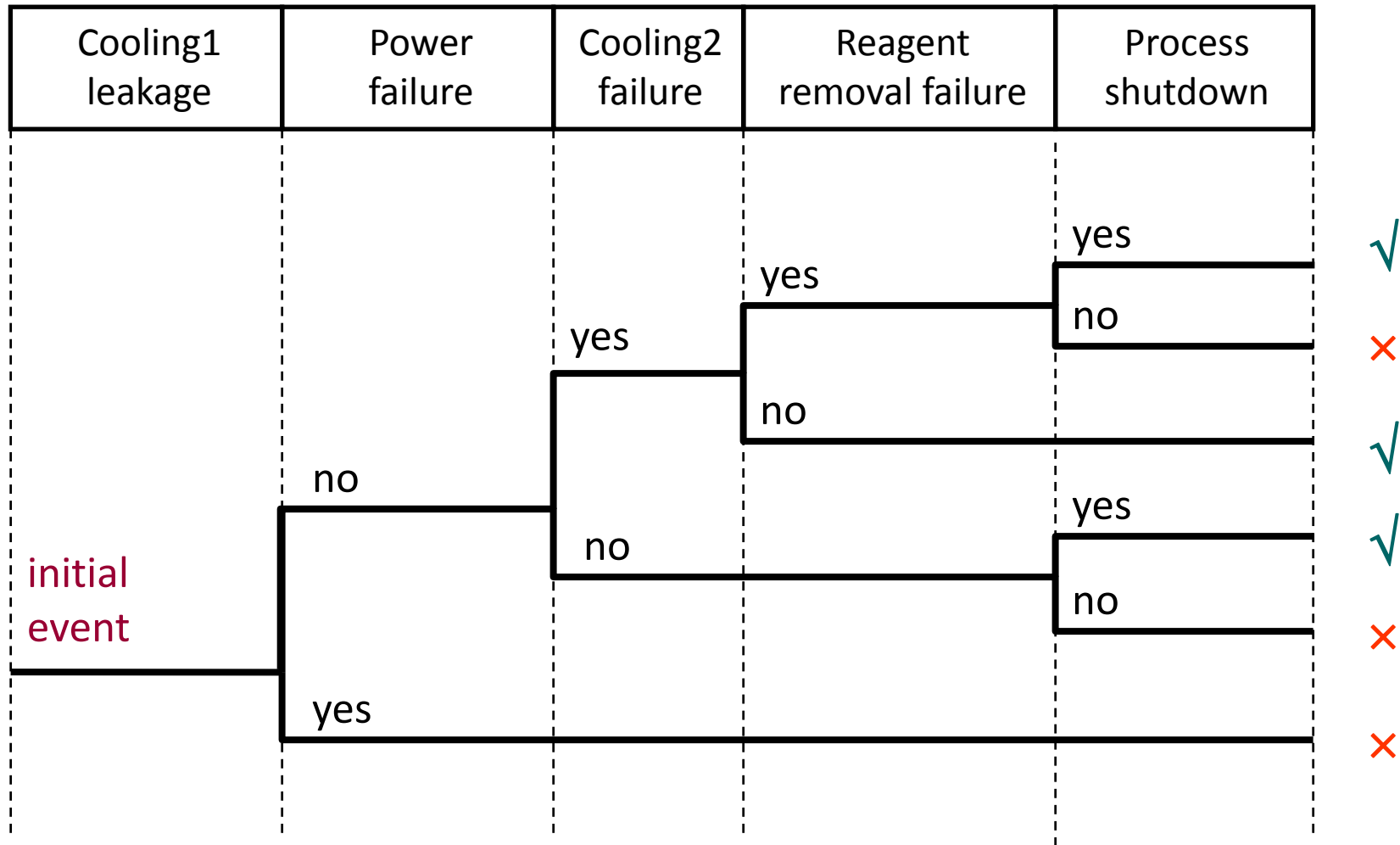
# Event tree analysis

Fire Starts	Fire Detected	Fire Alarm Starts	Sprinkler System Starts	Consequence	Result
			B9: Q=2.7208e-5	<i>Minimum Damage</i> W-1:R-3.02121e-17:	Seq-Q-3.02121e-17
		B6: Q=2.7208e-5	B10: Q=0.999973	<i>Damage No Loss of Life</i> W-2:R-2.22079e-12:	Seq-Q-1.11038e-12
	B2: Q=2.7208e-5		B11: Q=2.7208e-5	<i>Limited Damage / Wet People</i> W-7:R-7.77267e-12:	Seq-Q-1.11038e-12
		B6: Q=0.999973	B12: Q=0.999973	<i>Major Damage and Loss of Life</i> W-9b:R-0.134998:	Seq-Q-4.00090e-8
B1: Q=0.0015					
		B3: Q=0.999973	B6: Q=0.999973	<i>Major Damage and Loss of Life</i> W-9b:R-0.134998:	Seq-Q-0.00149988

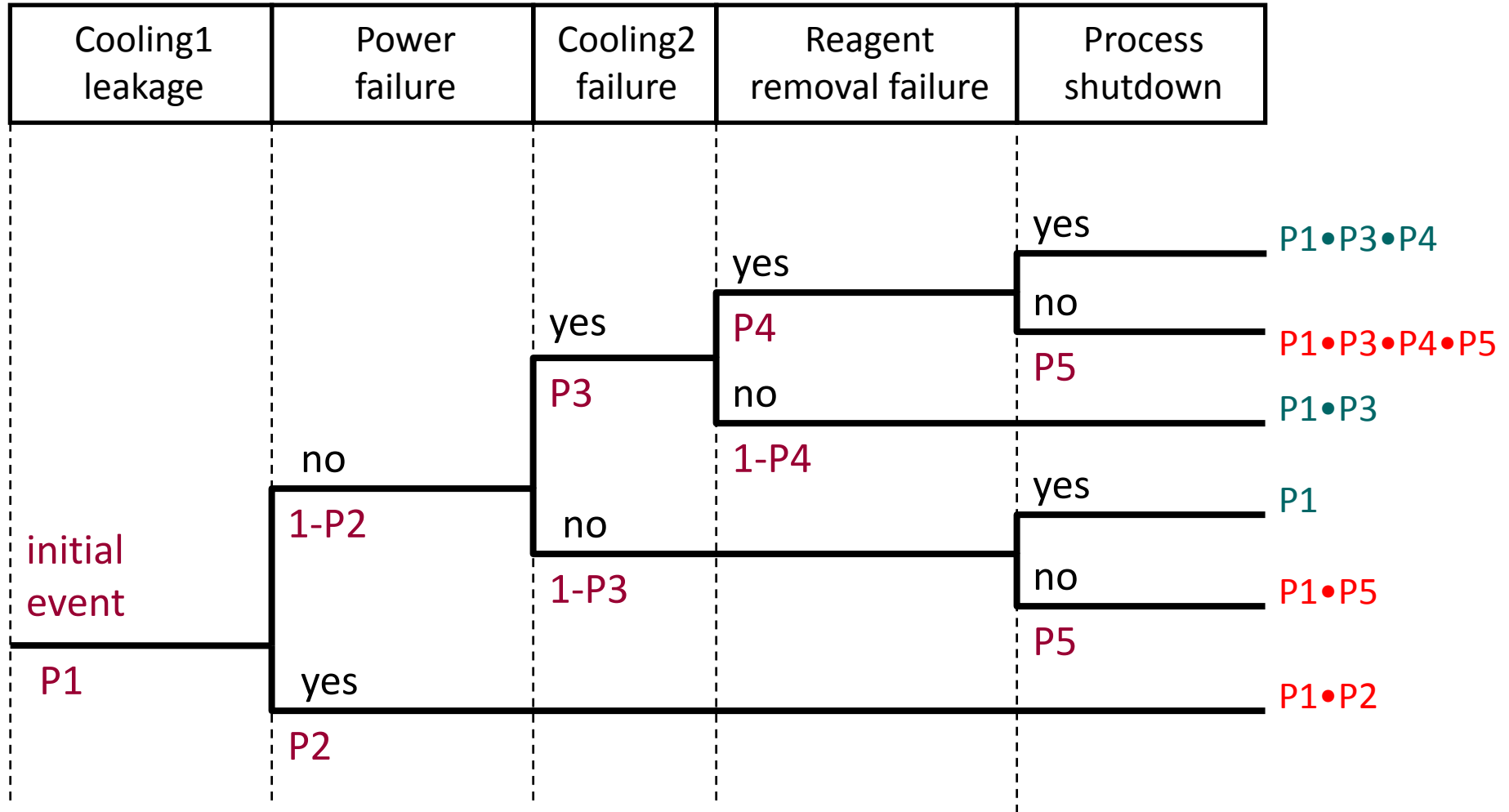
# Event tree analysis

- Forward (inductive) analysis:  
Investigates the **effects** of an initial event
  - **Initial event:** component level fault/event
  - Related events: faults/events of other components
  - Ordering: causality, timing
  - Branches: depend on the occurrence of events
- Investigation of **hazard occurrence „scenarios“**
  - Path **probabilities** (on the basis of branch probabilities)
- Advantages: Investigation of **event sequences**
  - Example: Checking protection systems (protection levels)
- Limits: Complexity, multiplicity of events

# Event tree example: Reactor cooling



# Event tree example: Reactor cooling



# Exercise: Evaluation of sensor subsystem

The temperature of a hot water storage is measured using **two sensors**.

- The two sensors may be faulty with probability **p1** and **p2**, in this case they report the invalid temperature +255°C.
- The faults of the sensors are checked by the **controller** performing an **acceptance check**.
- The sensor with **p1** fault probability is the **primary sensor**. The secondary sensor is read only in case of detecting the fault of the primary sensor.
- In case of a **faulty sensor**, the acceptance check always detects the fault.

However, due to a program bug, the acceptance check detects a sensor fault with probability **pe** even in case of a **non-faulty sensor**.



# Exercise: Evaluation of sensor subsystem

The temperature of a hot water storage is measured using **two sensors**.

- The two sensors may be faulty with probability **p1** and **p2**, in this case they report the invalid temperature +255°C.
- The faults of the sensors are checked by the **controller** performing an **acceptance check**.
- The sensor with **p1** fault probability is the **primary sensor**. The secondary sensor is read only in case of detecting the fault of the primary sensor.
- In case of a **faulty sensor**, the acceptance check always detects the fault. However, due to a program bug, the acceptance check detects a sensor fault with probability **pe** even in case of a **non-faulty sensor**.

Draw the event tree belonging to this system and calculate the probabilities of the scenarios.

The events:

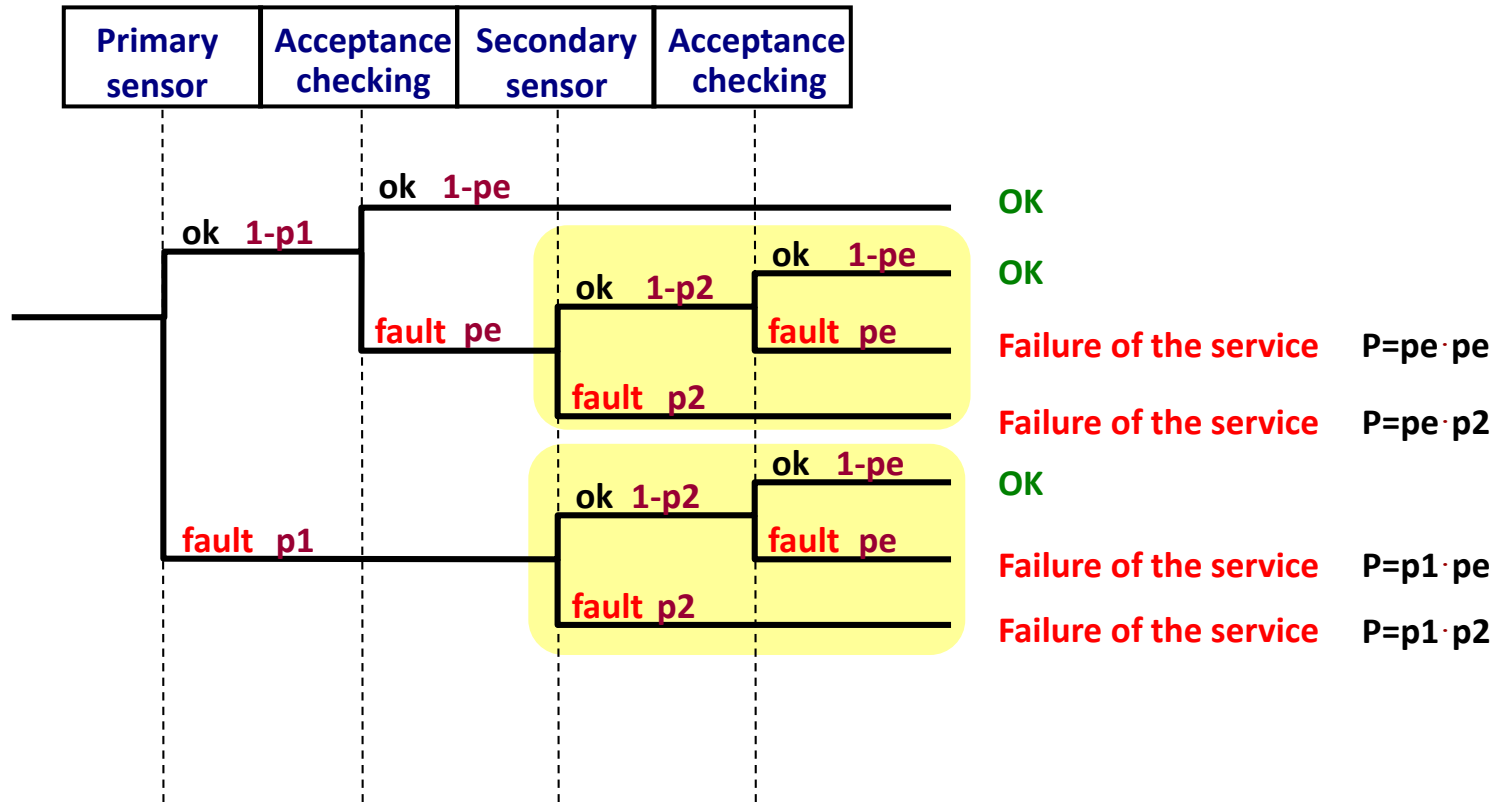
- Initial event: Starting the temperature measurement
- Further events: Faults of the sensors, fault of the acceptance checking

Ordering of events:

- Primary sensor ← may be faulty with probability **p1**
- Acceptance checking ← may be faulty with probability **pe** (in case of a non-faulty sensor)
- Secondary sensor ← may be faulty with probability **p2**
- Acceptance checking ← may be faulty with probability **pe** (in case of a non-faulty sensor)

# Solution of the exercise

Event tree:



Failure of the service at system level:  $pe \cdot pe + pe \cdot p2 + p1 \cdot pe + p1 \cdot p2$

# Failure modes and effects analysis

Item and (% chance of failure)	Failure mode		Effect of failure mode		Criticality of effect by severity type x 10 <sup>6</sup>			
	Description	Chance	Description	Chance	V.Hi	High	Med	Low
Main stack (0.2%)	Corruption	15%	Data loss	24%	180	495	2700	1225
	Overflow	60%	System crash	66%				
	Underflow	25%	Shutdown	90%				
			System crash	10%		300		
			Warning	98%				
Total					180	795	2700	1225

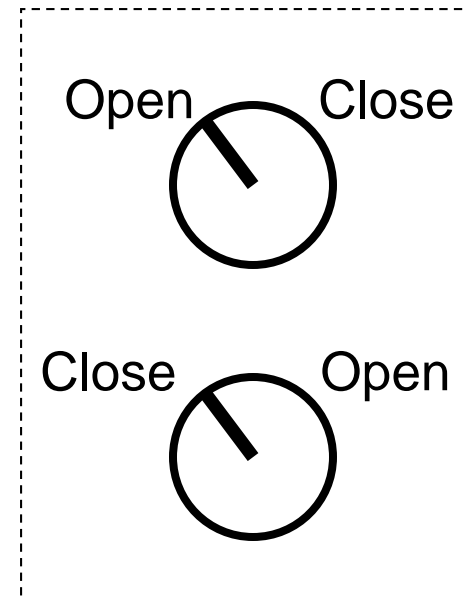
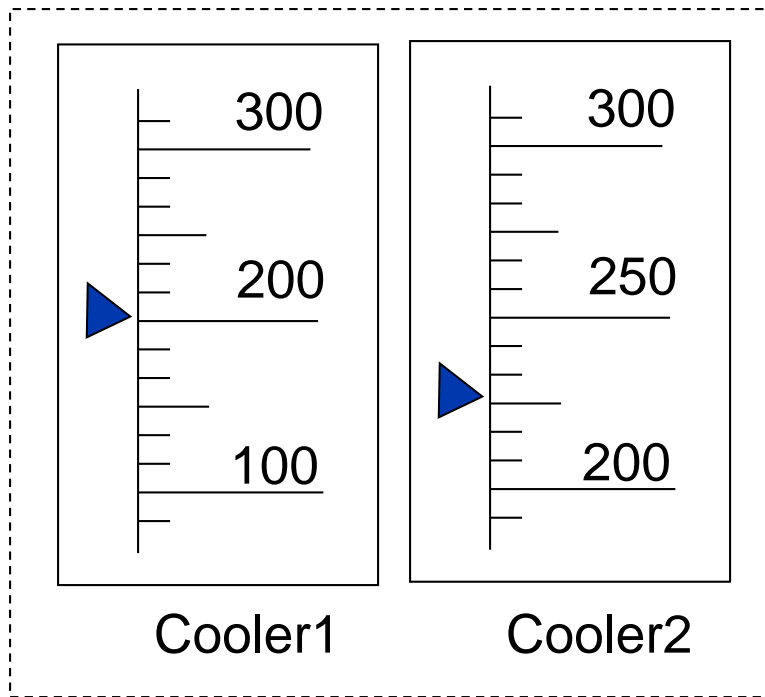
# Failure modes and effects analysis (FMEA)

- Systematic investigation of **component failure modes** and their **effects**
- Advantages:
  - Known faults of components are included
  - Criticalities of effects can also be estimated (FMECA)

Component	Failure mode	Probability	Effect
D1 diode	Open circuit	65%	Over-heating
	Short circuit	35%	Missing output
...	...	...	...

# Analysis of operator faults

- Qualitative techniques:
  - Operation – hazards – effects – causes – mitigations
  - Analysis of physical and mental demands
  - Fault causes ← **human-machine interface** problems



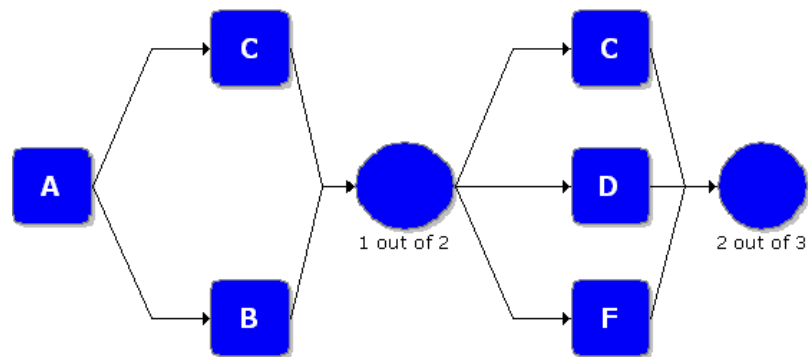
# Outcome of hazard analysis

- Categorization of hazards on the basis of hazard analysis (e.g., MIL-STD-822b, NASA):
  - **Probability / rate** of hazard occurrence calculated: Frequent, probable, occasional, remote, improbable, incredible
  - **Severity level** of hazard consequences estimated: Catastrophic, critical, marginal, insignificant
- Identification of **risks**
- Output of the rate and severity analysis:
  - **Risk matrix**
  - **Protection level**: Identifies the risks to be handled

# Example: Risk matrix (railway control systems)

	Frequency of Occurrence of a Hazardous Event	RISK LEVELS			
Daily to monthly	<b>FREQUENT (FRE)</b>	Undesirable (UND)	Intolerable (INT)	Intolerable (INT)	Intolerable (INT)
Monthly to yearly	<b>PROBABLE (PRO)</b>	Tolerable (TOL)	Undesirable (UND)	Intolerable (INT)	Intolerable (INT)
Between once a year and once per 10 years	<b>OCCASIONAL (OCC)</b>	Tolerable (TOL)	Undesirable (UND)	Undesirable (UND)	Intolerable (INT)
Between once per 10 years and once per 100 years	<b>REMOTE (REM)</b>	Negligible (NEG)	Tolerable (TOL)	Undesirable (UND)	Undesirable (UND)
Less than once per 100 years	<b>IMPROBABLE (IMP)</b>	Negligible (NEG)	Negligible (NEG)	Tolerable (TOL)	Tolerable (TOL)
	<b>INCREDIBLE (INC)</b>	Negligible (NEG)	Negligible (NEG)	Negligible (NEG)	Negligible (NEG)
		<b>INSIGNIFICANT (INS)</b>	<b>MARGINAL (MAR)</b>	<b>CRITICAL (CRI)</b>	<b>CATASTROPHIC (CAT)</b>
		<b>Severity Levels of Hazard Consequence</b>			

# Reliability block diagrams





# Boole model for calculating dependability

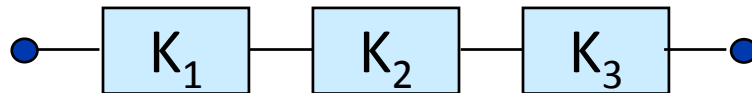
- Boole model of components
  - Two states: **Fault-free** (good) or **faulty** (bad)
  - No dependences regarding faults or repairing
- Relation of components from the point of view of dependability: What kind of **redundancy** is used?
  - **Serial connection:**
    - If both components are necessary for the operation of the system
    - I.e., the components are **not redundant**
  - **Parallel connection:**
    - If the components may replace each other in case of their failure
    - I.e., the components are **redundant**

The connection may depend on the failure modes

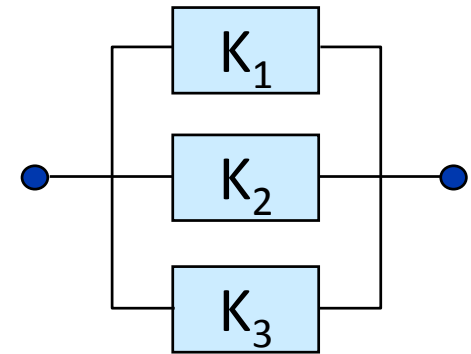
# Reliability block diagram

- **Blocks:** Components
- **Connections:** Serial or parallel (redundancy)
- **Paths:** Operational system configurations
  - The system is **operational** (correct) if **there is a path** from the start point to the end point of the diagram through fault-free components

**Serial:**

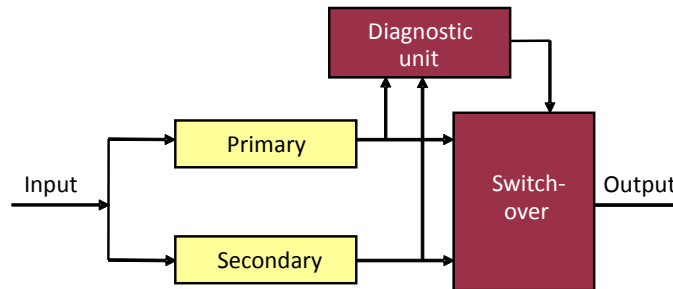


**Parallel:**

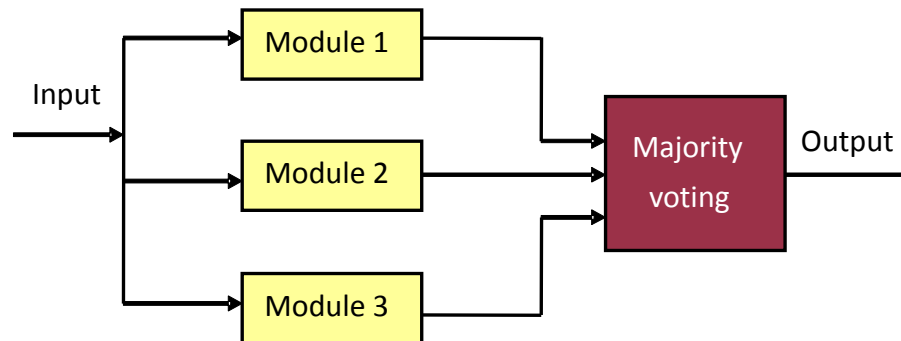


# Overview: Typical system configurations

- Serial system model: **no redundancy**
- Parallel system model: **redundancy** (replication)



- Complex canonical system: redundant subsystems
- M faulty out of N components: **Majority voting** (TMR)



# Previous topic: Attributes of components

- Data from product sheet / reliability handbook:

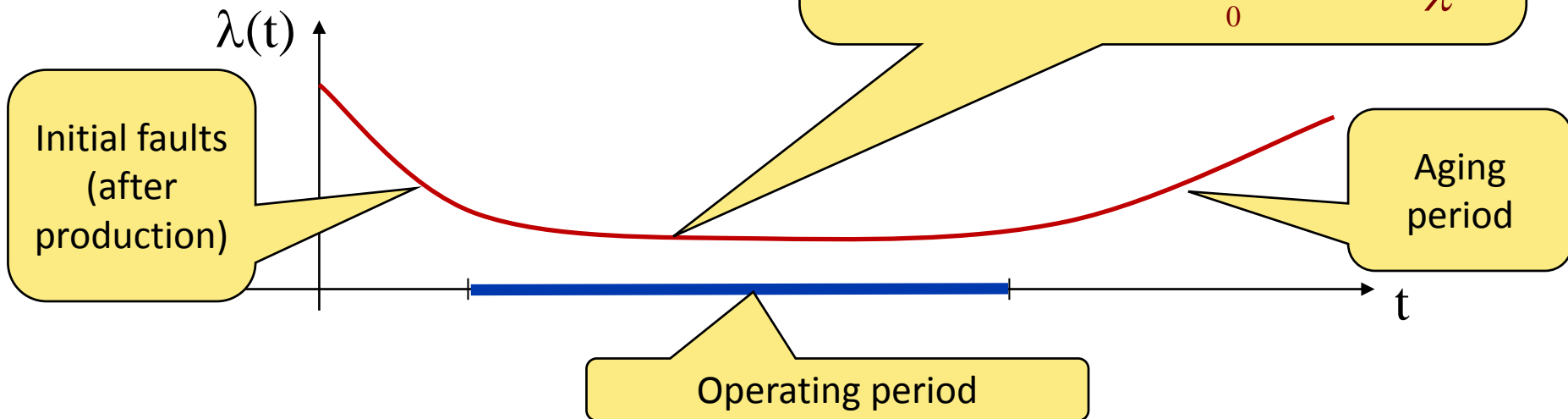
**Fault rate:**  $\lambda(t)$

- Reliability of components:  $r(t) = e^{-\int_0^t \lambda(t) dt}$

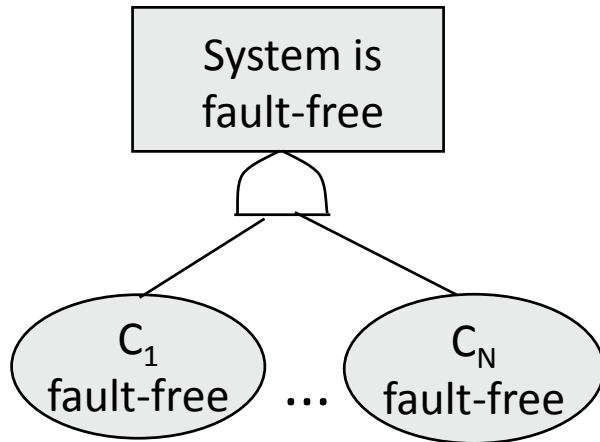
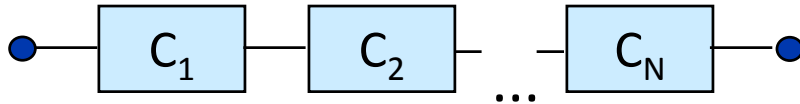
- For electronic components:

Here  $r(t) = e^{-\lambda t}$

$$MTFF = E\{u1\} = \int_0^{\infty} r(t) dt = \frac{1}{\lambda}$$



# Serial system



$P(A \wedge B) = P(A)P(B)$   
if independent

## Reliability:

$$r_R(t) = \prod_{i=1}^N r_i(t)$$

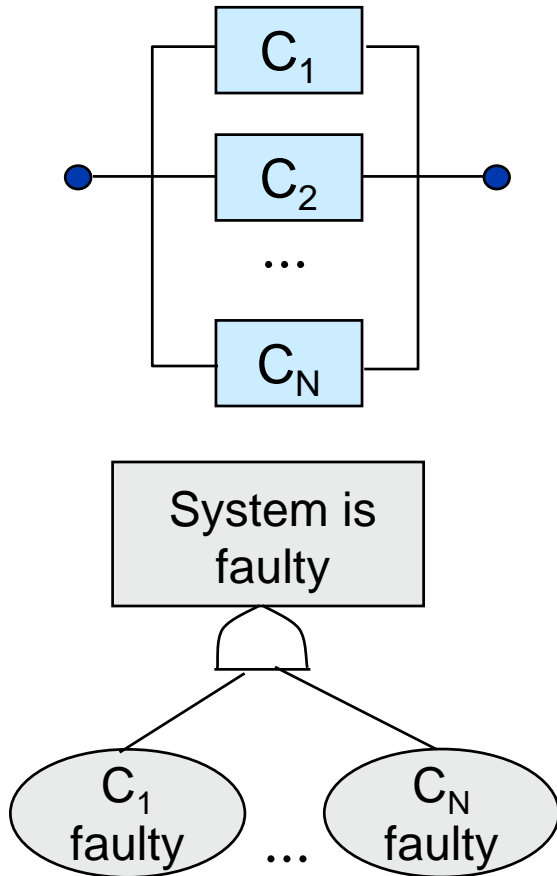
System level  
reliability

Component  
reliability

## MTFF:

$$MTFF = \frac{1}{\sum_{i=1}^N \lambda_i}$$

# Parallel system



$P(A \wedge B) = P(A)P(B)$   
if independent

- Reliability:

$$1 - r_R(t) = \prod_{i=1}^N (1 - r_i(t))$$

- Uniform N components:

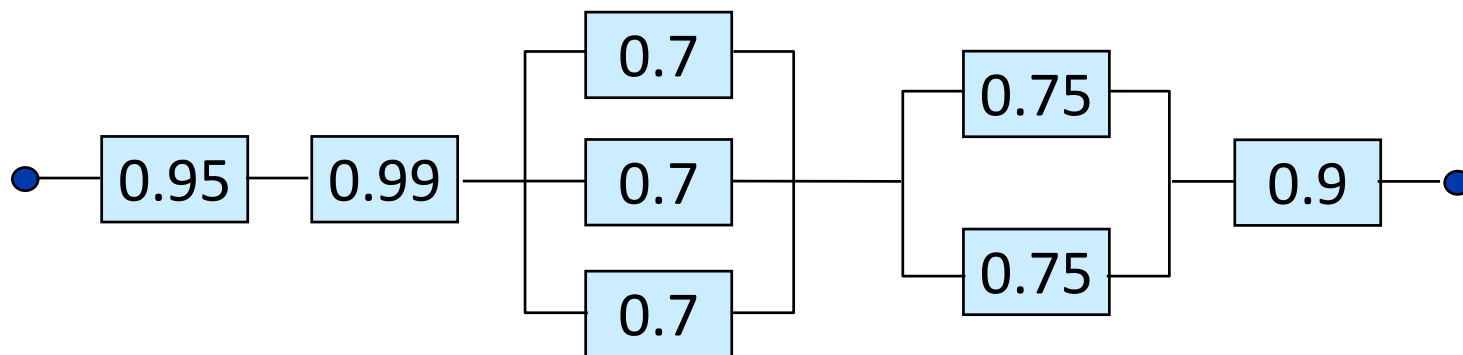
$$r_R(t) = 1 - (1 - r_K(t))^N$$

- MTFF (without explanation):

$$MTFF = \frac{1}{\lambda} \sum_{i=1}^N \frac{1}{i}$$

# Complex canonical system

- Calculation on the basis of parts with basic connections
  - Example: Calculation of asymptotic availability



$$K_R = 0.95 \cdot 0.99 \cdot \left[ 1 - (1 - 0.7)^3 \right] \cdot \left[ 1 - (1 - 0.75)^2 \right] \cdot 0.9$$

# M faulty out of N components

- **N** replicated components;

If **M** or more components faulty: the system is faulty

$$r_R = \sum_{i=0}^{M-1} P \{ \text{"there are } i \text{ faults"} \}$$

$$r_R = \sum_{i=0}^{M-1} \binom{N}{i} (1-r)^i \cdot r^{N-i}$$

- Application: Majority voting (TMR): N=3, M=2

$$r_R = \sum_{i=0}^1 \binom{3}{i} (1-r)^i \cdot r^{3-i} = \binom{3}{0} (1-r)^0 \cdot r^3 + \binom{3}{1} (1-r)^1 \cdot r^2 = 3r^2 - 2r^3$$

$$MTFF = \int_0^{\infty} r_R(t) dt = \int_0^{\infty} (3r^2 - 2r^3) dt = \frac{5}{6} \cdot \frac{1}{\lambda}$$

Less than in the case of a single component!



# Exercise: Availability of a SCADA system

A SCADA system consists of the following components:

4 data collector units, 3 control units, 2 supervisory servers, 1 logging server and the corresponding network

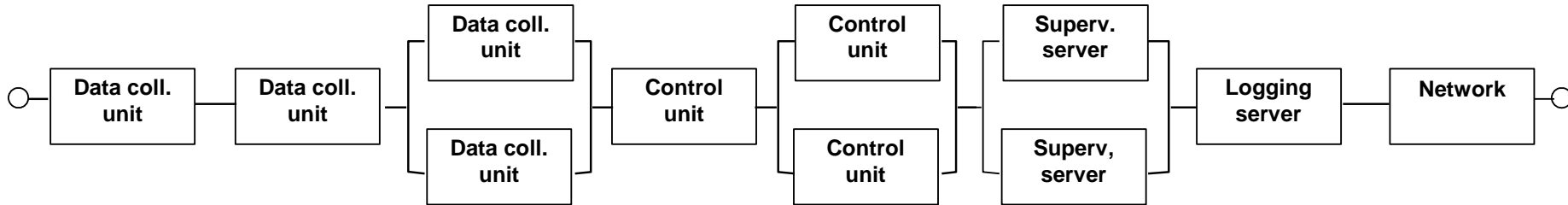
- The 2 supervisory servers are in a hot redundancy structure.
- Critical data collector and control units are in a hot redundancy structure: 2 data collector units and 2 control units are hot redundant units
- The reliability data of the system components are given as follows (measured in hours, with independent repairs in case of faults):

	Data coll. unit	Control unit	Superv. server	Logging server	Network
MTTF	9000	12000	4500	2000	30000
MTTR	2	3	5	1	2

- Evaluate the system level availability using a reliability block diagram.
- Compute the asymptotic availability of the system using the above given parameters of the system components.
- How many hours is the system out of service per year?

# Solution of the exercise

Reliability block diagram:



Component level asymptotic availability:  $K = \text{MTTF} / (\text{MTTF} + \text{MTTR})$

	Data coll. unit (D)	Control unit (C)	Superv. server (S)	Logging server (L)	Network (N)
MTTF	9000	12000	4500	2000	30000
MTTR	2	3	5	1	2
K	KD=0.99977	KC=0.99975	KS=0.99889	KL=0.9995	KN=0.99993

System level asymptotic availability:

$$KD * KD * (1 - (1 - KD) * (1 - KD)) * KC * (1 - (1 - KC) * (1 - KC)) * (1 - (1 - KS) * (1 - KS)) * KL * KN = 0.9987362$$

Approx. 11 hours out of service per year

# Summary

## ■ Hazard analysis

- Fault tree analysis
- Event tree analysis
- Failure modes and effects analysis (FMEA)
- Risk matrix:
  - Severity level of hazard consequences
  - Rate of hazard occurrence

## ■ Reliability analysis

- Reliability block diagrams