# Common Cause Failure (CCF)

건국대학교 컴퓨터공학과

UC Lab. 정혁준 & 박경식

amitajung@naver.com, kyeongsik@konkuk.ac.kr

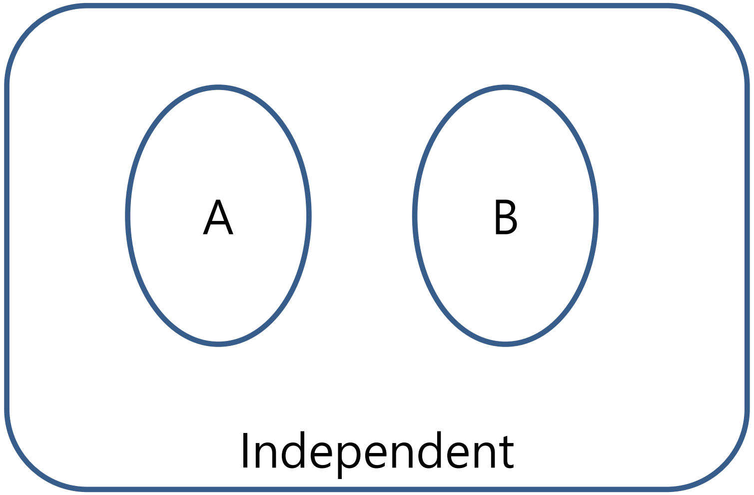
# Contents

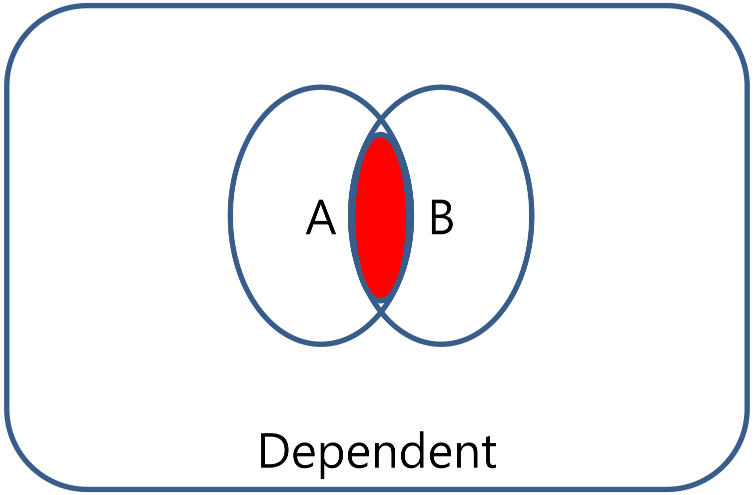
- Common Cause Failure (CCF)
- Types of CCF
- Examples
- Reducing CCF

# Common Cause Failure (CCF)

## Definition of CCF

- *Dependent Failures* in which two or more component fault states exist at the same time, or within a short time interval, as a result of a shared cause.



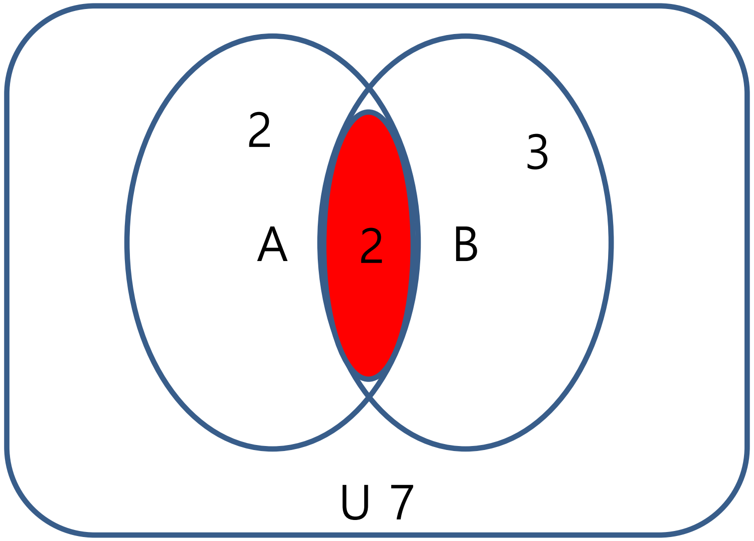



# • Common Cause Failure (CCF)

## Conditional Probability

The probability of an event given that another event has occurred.

"The conditional probability of *A* given *B*", or "the probability of *A* under the condition *B*", is usually written as $P(A|B)$




$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$= \frac{2}{7} \div \frac{5}{7} = \frac{2}{5}$$

# • Common Cause Failure (CCF)

## Independent and Dependent Failures

Consider the event that item Ei is in a failed state. The probability that both items are in a failed state is

$$\Pr(E_1 \cap E_2) = P(E_1|E_2) \cdot P(E_2) = P(E_2|E_1) \cdot P(E_1)$$

**Independent**

$$P(E_1|E_2) = P(E_1)$$

$$P(E_2|E_1) = P(E_2)$$

**Dependent**

$$\Pr(E_1 \mid E_2) \neq \Pr(E_1) \quad \text{and} \quad \Pr(E_2 \mid E_1) \neq \Pr(E_2)$$

Positive dependence

$$P(E_1|E_2) > \Pr(E_1) \cdot \Pr(E_2)$$

Negative dependence

$$P(E_1|E_2) < \Pr(E_1) \cdot \Pr(E_2)$$

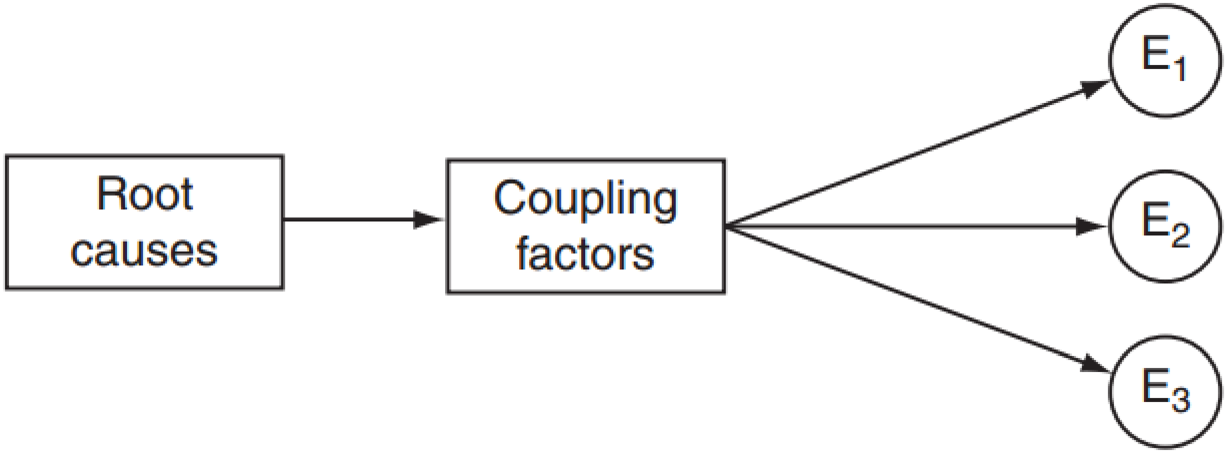
# • Common Cause Failure (CCF)

## Dependent Failures

The shared cause has two elements, a root cause and a coupling factor:

**Root cause :** Most basic cause of item failure that, if corrected, would prevent recurrence of this and similar failures.

**Coupling factor** : Property that makes multiple items susceptible to the same root cause.



# • Common Cause Failure (CCF)

## Typical Root Causes

**Pre-Operational Root Causes**
Design, manufacturing, construction, installation errors.

**Operational Root Causes**
- **Operation and Maintenance-Related:** Inadequate maintenance and execution, competence and scheduling
- **Environmental Stresses:** Internal and external exposure outside the design envelope or energetic events such as earthquake, fire, flooding

# • Common Cause Failure (CCF)

## Typical Coupling Factors

Same design
Same hardware
Same function
Same software
Same installation staff
Same maintenance and operational staff
Same procedures
Same system/item interface
Same environment
Same (physical) location

# • Common Cause Failure (CCF)

**NUREG/CR-6268 -** Common-Cause Failure Database and Analysis System

**Extrinsic dependency:** A situation where the dependency or coupling is not internal of the system.

Physical or environment stresses.
Human

**Intrinsic dependency:** A situation where the functional status of a component is affected by the functional status of other components.

Functional requirement dependency
Functional input dependency
Cascading failure

# • Common Cause Failure (CCF)

## Cascading Failures

A **cascading failure** is a failure in a system of interconnected parts in which the failure of a part can trigger the failure of successive parts.

Such a failure may happen in many types of systems, including power transmission, computer networking, finance, human bodily systems, bridges even **Finance**!!

# • Common Cause Failure (CCF)

## Attributes of a CCF definition

Smith and Watson (1980) suggest that a definition of CCF should encompass:

1 The items affected are unable to perform as required

2 Multiple failures exist within redundant configurations

*3 The failures are "first-in-line" type of failures and not the result of cascading failures*

4 The failures occur within a defined critical time period
(e.g., the time a plane is in the air during a flight)

5 The failures are due to a single underlying defect or physical phenomenon (the "common cause")

UCLAB UBIQUITOUS COMPUTING LABORATORY KU 건국대학교 KONKUK UNIV.

# • Common Cause Failure (CCF)

## Some different definitions

**Nuclear industry (NEA, 2004)**

A dependent failure in which two or more component fault states exist simultaneously or within a short time interval, and are a <u>direct result of a shared cause</u>

**Space industry (NASA PRA guide, 2002)**

The failure (or unavailable state) of more than one component due to a shared cause <u>during the system mission</u>.

**Process industry (IEC 61511, 2003)**

Failure, which is the result of one or more events, causing failures of two or more separate channels in a <u>multiple channel system, leading to system failure</u>.

# Common Cause Failure (CCF)

## CCF Modeling

1 Develop a system logic model (e.g., a fault tree or a reliability block diagram)

2 Identify relevant common cause component groups (CCCG)

3 Identify relevant root causes and coupling factors/mechanisms

4 Assess the efficiency of CCF defenses

5 Establish explicit models

6 Include implicit models

7 Quantify the reliability and interpret the results

**Common cause component group (CCCG):** A set of system items that may have the same CCF

# Common Cause Failure (CCF)

## Explicit Modeling

The shared cause is identified as a separate basic event/element in the reliability model.

Explicit causes may be:

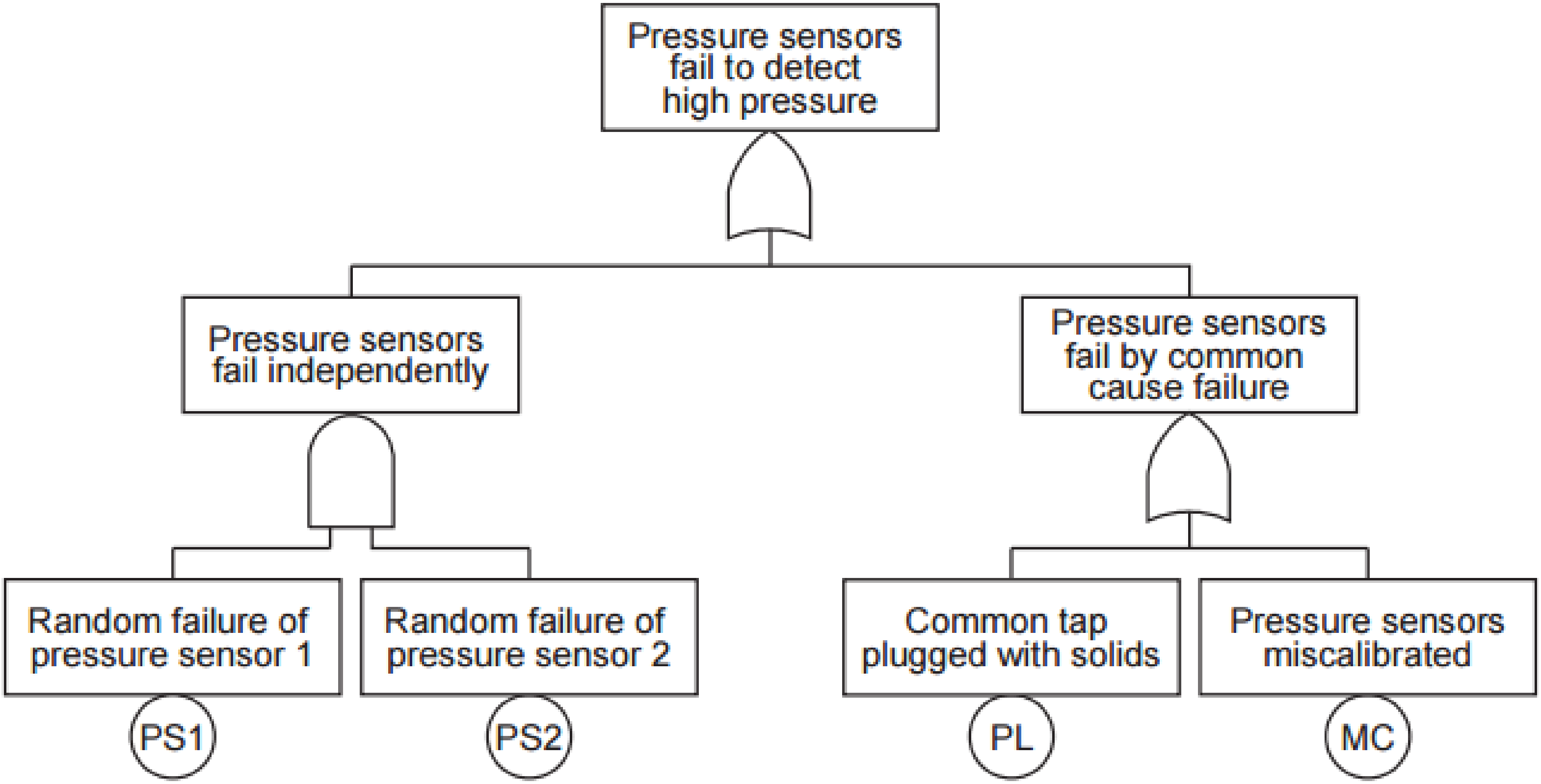
- Human errors
- Utility failures (e.g., power failure, cooling/heating failure, loss of hydraulic power)
- Environmental events (e.g., lightning, flooding, storm)

Event tree and fault tree analysis

Consideration of functional interdependencies

# • Common Cause Failure (CCF)

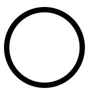
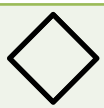
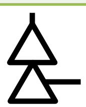
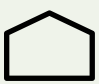
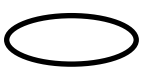
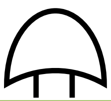
**Explicit Modeling Example: Two pressure sensors**




Summers and Raney (1999)

# • Fault Tree

## Fault Tree

| Symbol | Description |
| --- | --- |
|  | Event : Symbol indicates a case arises in the combination of the case through the logic gate |
|  | Basic Event : More symbols representing the basic error event does not require the development |
|  | Undeveloped Event : Not analyzed by the lack of information or analysis is required or not is a symbol representing the abbreviation phenomena |
|  | Sign indicating the electric information between the other part is the same as in Fault Tree |
|  | Symbols indicating events that can be expected to occur normally |
|  | Symbol showing the state that must be considered in the production of the gate of the output |
|  | AND Gate : A logic gate that is used to satisfy all of the lower case |
|  | OR Gate : A logic gate that  is used to satisfy any one of the sub case |

# • Common Cause Failure (CCF)

## Implicit Modeling

Where a set of items share a number of root causes and coupling factors, and where the explicit modeling would be unmanageable, the (residual) shared causes are modeled as a "combined" basic event/element.

The implicit modeling implies approach of the use of a CCF modeling.

Marshall-Olkin-Model "2-out-of-3-system", *b*-Factor-Model, MGL-Model (Multiple Greek Letter), BFR-Model (Binominal Failure Rate)

# • Common Cause Failure (CCF)

## Multiplicity

Consider a system of three components 1, 2, and 3, and let Ei be the event that component i is in a failed state.

A failure event can have 3 different multiplicities:

A single failure, where only one component fails, can occur in 3 different ways as:

$(E_1 \cap E_2^* \cap E_3^*)$, $(E_1^* \cap E_2 \cap E_3^*)$, or $(E_1^* \cap E_2^* \cap E_3)$

A double failure can also occur in three different ways as:

$(E_1 \cap E_2 \cap E_3*)$, $(E_1 \cap E_2 * \cap E_3)$, or $(E_1^* \cap E_2 \cap E_3)$

A triple failure occurs when

$(E_1 \cap E_2 \cap E_3)$

# Common Cause Failure (CCF)

## Multiplicity

Probability of a specific combination for a system of 3 identical channels:

$$
\begin{aligned}
g_{1,3} &= \Pr(E_1 \cap E_2^* \cap E_3^*) = \Pr(E_1^* \cap E_2 \cap E_3^*) \\
&= \Pr(E_1^* \cap E_2^* \cap E_3)
\end{aligned}
$$

$$
\begin{aligned}
g_{2,3} &= \Pr(E_1 \cap E_2 \cap E_3*) = \Pr(E_1 \cap E_2 * \cap E_3) \\
&= \Pr(E_1^* \cap E_2 \cap E_3)
\end{aligned}
$$

$$
g_{3,3} = \Pr(E_1 \cap E_2 \cap E_3)
$$

# • Common Cause Failure (CCF)

## Multiplicity

Probability of a specific multiplicity

$$Q_{1:3} = \binom{3}{1} \cdot g_{1,3} = 3 \cdot g_{1,3}$$

$$Q_{2:3} = \binom{3}{2} \cdot g_{2,3} = 3 \cdot g_{2,3}$$

$$Q_{3:3} = \binom{3}{3} \cdot g_{3,3} = g_{3,3}$$

# • Common Cause Failure (CCF)

## 2-out-of-3 system

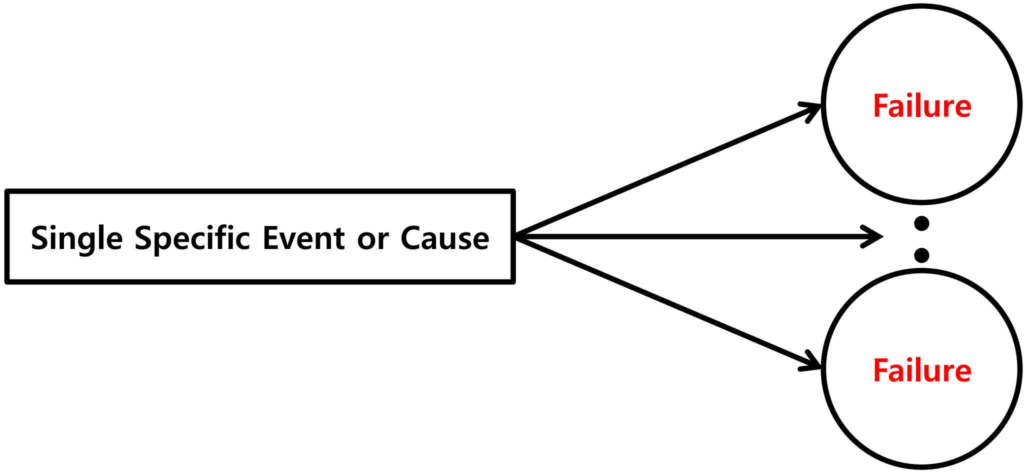
N-M 시스템에서 M개의 시스템 중 N개가 고장이 났을 경우 전체 시스템이 고장날 확률

$$\begin{aligned}\Pr(\text{System failure}) &= Q_{2:3} + Q_{3:3} \\ &= 3 \cdot g_{2,3} + g_{3,3}\end{aligned}$$

# Common Cause Failure (CCF)

## Definition

- High technology industries with high failure costs commonly use **redundancy** as a means to reduce risk
- Redundant systems, whether similar or dissimilar, are susceptible to **Common Cause Failures (CCF)**
- **Common Cause Failure (CCF)** is "A failure of two or more components, system, or structures due to a single specific event or cause."



# • Common Cause Failure (CCF)

## Safety Instrumented System (SIS)

- Common Cause Failures (CCFs) are an important part of reliability analysis, and engineers have been aware of these type of failures
- **Safety Instrumented System (SIS)** is a system which consists of **sensors**, **logic solvers** and **actuating items**
- A fire and gas detection system with an alarm or a sprinkler system is an example of a **SIS**
- A **SIS** is constructed to take the process into a **safe state** if a dangerous event occurs

Gas Detectors

→ **Logic Solver** →

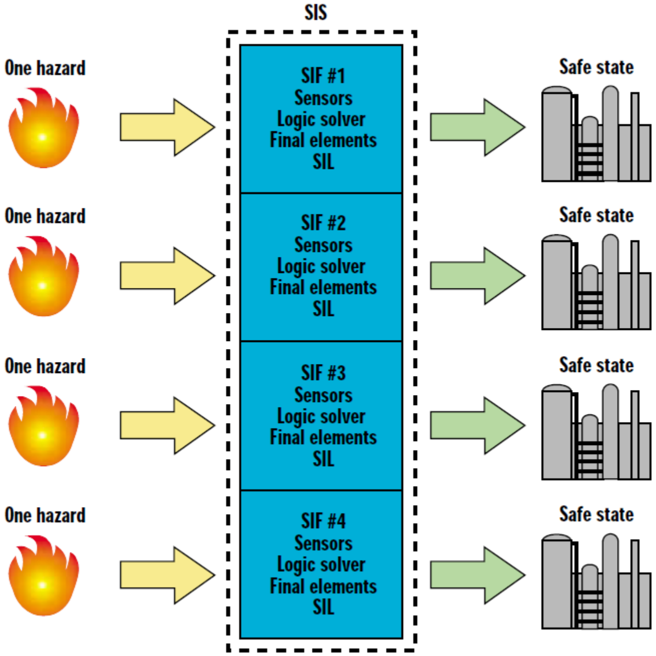
Shut down valves

# • Common Cause Failure (CCF)

## Safety Instrumented System (SIS)

- **Safety Instrumented Function (SIF)** is a function that is implemented by a **SIS**, <u>**SIS** may consist of several **SIFs**</u>

- Each **SIF** has to fulfill a requirement which is called **Safety Instrumented Level (SIL)**



# • Common Cause Failure (CCF)

## Safety Instrumented System (SIS)

- Safety integrity is defined as

> The probability of a safety-related system satisfactorily performing the required Safety functions under all the stated conditions within a stated period of time
>
> IEC 61508 (2000, Part 4)

- The measure is classified into four different discrete levels defined as **Safety Integrity Levels (SIL)**

| SIL | Low Demand Mode | High Demand Mode |
|---|---|---|
| 4 | $\geq 10^{-5}$ to $< 10^{-4}$ | $\geq 10^{-9}$ to $< 10^{-8}$ |
| 3 | $\geq 10^{-4}$ to $< 10^{-3}$ | $\geq 10^{-8}$ to $< 10^{-7}$ |
| 2 | $\geq 10^{-3}$ to $< 10^{-2}$ | $\geq 10^{-7}$ to $< 10^{-6}$ |
| 1 | $\geq 10^{-2}$ to $< 10^{-1}$ | $\geq 10^{-6}$ to $< 10^{-5}$ |

# • Common Cause Failure (CCF)

## Safety Instrumented System (SIS)

- **Low Demand Mode** : The frequency of demands for operation made on a safety-related system is **no greater than one per year** and **no greater than twice the proof-test frequency**

Ex. Shut down valves, Heat detector

- **High Demand Mode** : The frequency of demands for operation made on a safety-related system is **greater than one per year** or **twice the proof-test frequency**

Ex. Braking system of a car

# Types of CCF

- There are several contributing **factors** or **causes** for a CCF
- The following is a brief list of **causes** which can take out **redundant components** or **systems**

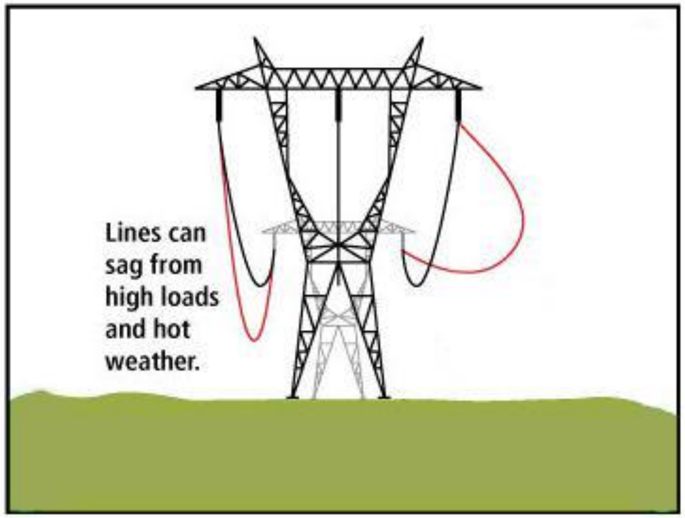
| | | |
|---|---|---|
| *System or Component Requirements* | | *Loss of Power* |
| *Wear Out* | | *Software* |
| *Contamination* | | *Saturation of Signals* |
| *Corrosion* | | *Design Deficiency* |
| Environment | *Weather* | *Transportation/Shipping* |
| | *Lightning/Electromagnetic Interference* | *Human Error/System Complexity* |
| | *Earthquake* | *Cascading* |
| | *Thermal Conditions* | *Single Physical Point where Redundant Items Meet* |
| *Lack of Process Control/Manufacturing Deficiency* | | |

# • Examples

*NASA Marshall Space Flight Center, Huntsville, Alabama, USA*

## Power Grid (Cascading)

- Hot summer day
  - Led to increased power consumption
  - Led to power lines sagging
- One set of power lines were lost -> Increasing load on remaining lines
  - Those lines sagged

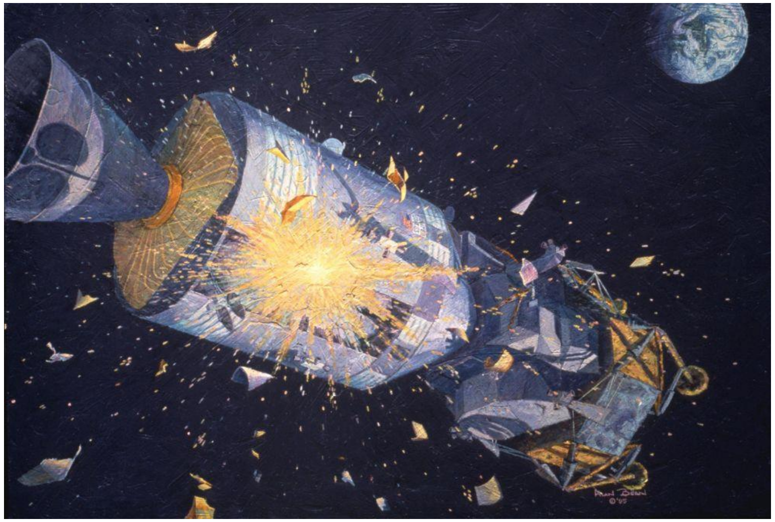


## • Examples

*NASA Marshall Space Flight Center, Huntsville, Alabama, USA*

### Apollo 13 Explosion (Single Physical Point)

- Oxygen Tank 1 and its redundant supply, Oxygen Tank 2, were located directly adjacent to each other
- Oxygen Tank 2 blast
  - The concussion from the blast also damaged Oxygen Tank 1
  - Causing it to leak, Emptying its entire supply to space

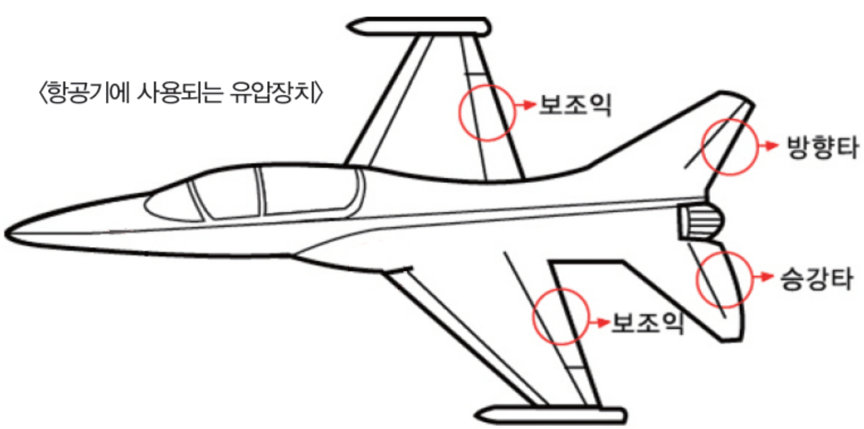


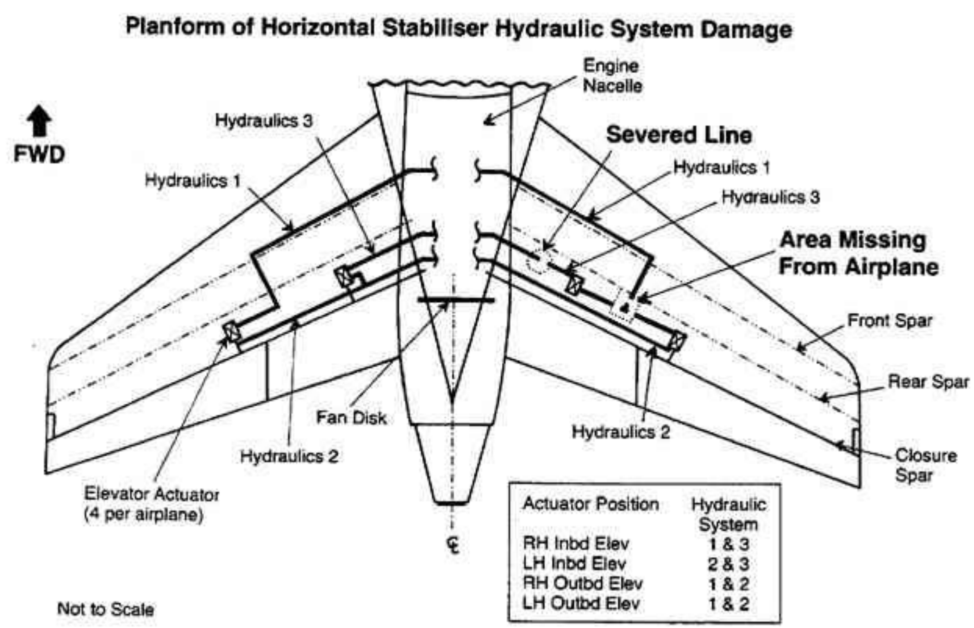
# • Examples

*NASA Marshall Space Flight Center, Huntsville, Alabama, USA*

## Airlines Flight 232 (Single Physical Point)

- All 3 redundant hydraulic systems were cut by single engine failure
- Non designed in redundancy
  - Using remaining two engines to control the plane, saved many lives



# • Examples

*NASA Marshall Space Flight Center, Huntsville, Alabama, USA*

## Japan’s Fukushima Daiichi Power Plant (Environmental)

- Backup generators used to generate power if an earthquake interrupted power failed
  - Due to the water from a tsunami flooding the system
- The thought of the CCF of an earthquake both causing power loss and a tsunami of sufficient size to overcome the wall created to protect the plant was not envisioned

# • Examples

## RAID System 1

- When two disks are purchased online and are installed in a computer
  - There can be many Common Cause Failure
- The disks are likely from the same manufacturer and of the same model
  - They share the same design flaws **(Design Deficiency)**
- The disks are likely to have similar serial numbers
  - They may share any manufacturing flaws affecting production of the same batch **(Manufacturing Deficiency)**
- The disks are likely to have been shipped at the same time
  - They are likely to have suffered from the same transportation damage **(Transportation/Shipping)**

UCLAB UBIQUITOUS COMPUTING LABORATORY KU 건국대학교 KONKUK UNIV.

# • Examples

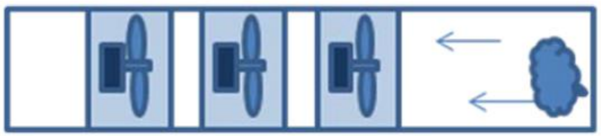
## RAID System 2

- As installed, both disks are attached to the same power supply
  - Making them vulnerable to the same power supply issues **(Loss of Power)**
- As installed, both disks are in the same case
  - Making them vulnerable to the same overheating events **(Thermal Conditions)**
- They will be both attached to the same card or motherboard, and driven by the same software
  - May have the same bugs or viruses **(Software)**
- Both disks will be subjected to the same workload and to very repetitive similar access patterns, stressing them in the same way.
  - stressing them in the same way **(Wear Out)**

# • Examples of Reducing CCF

## Environmental Control Fan (Cascading)

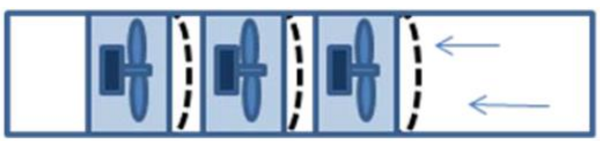
- On orbit, air flow is required to maintain life



All three fans could be susceptible to dirt/debris from cabin

A screen could prevent this

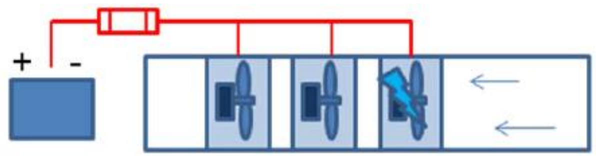
One fan can fail, sending debris into other fans, a cascading failure



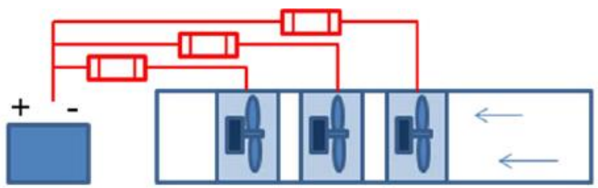
Each fan having a screen will limit this

# Examples of Reducing CCF

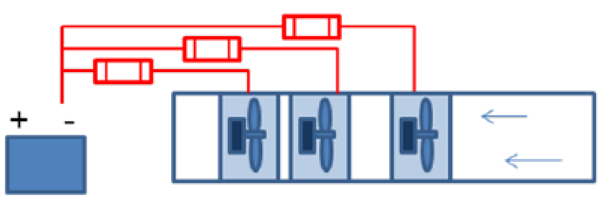
## Environmental Control Fan (Loss of Power)



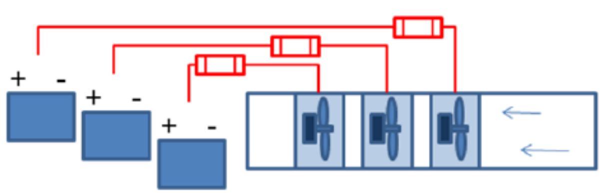

All three fans could be susceptible to loss of power if one fan has a short




Each fan having a fuse will limit this




One battery could fail causing all fans to fail

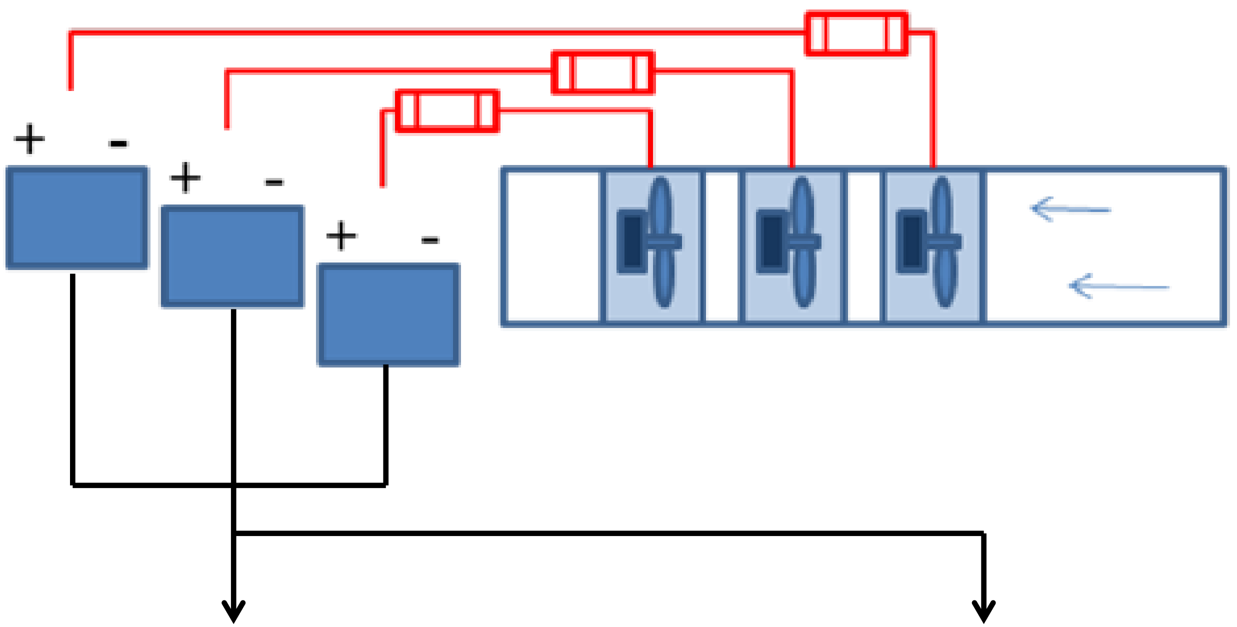



Redundant batteries could prevent this

# • Examples of Reducing CCF

## Environmental Control Fan

- Using Diverse(Unlike) Redundancy

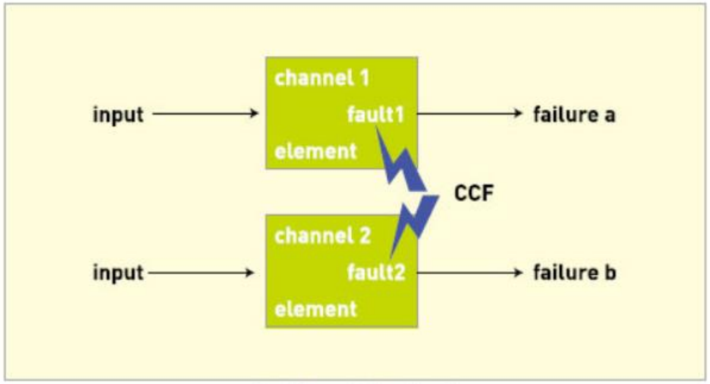


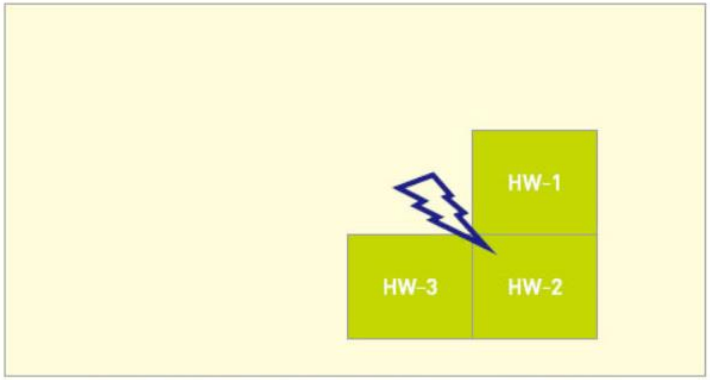

Physically Separated

Manufactured by different vendor

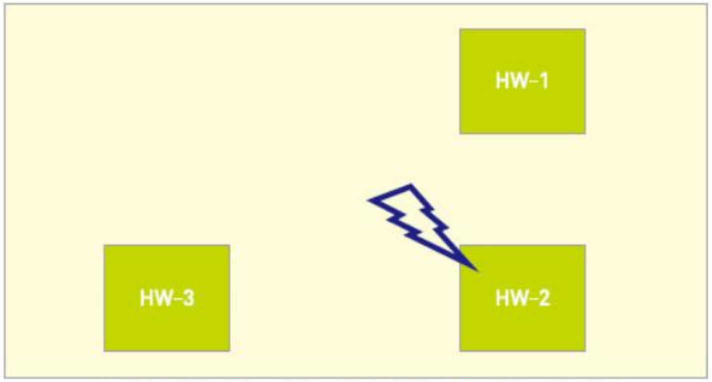
# Examples of Reducing CCF

## Closely Located Hardware Device (Single Physical Point)




Closely located hardware device

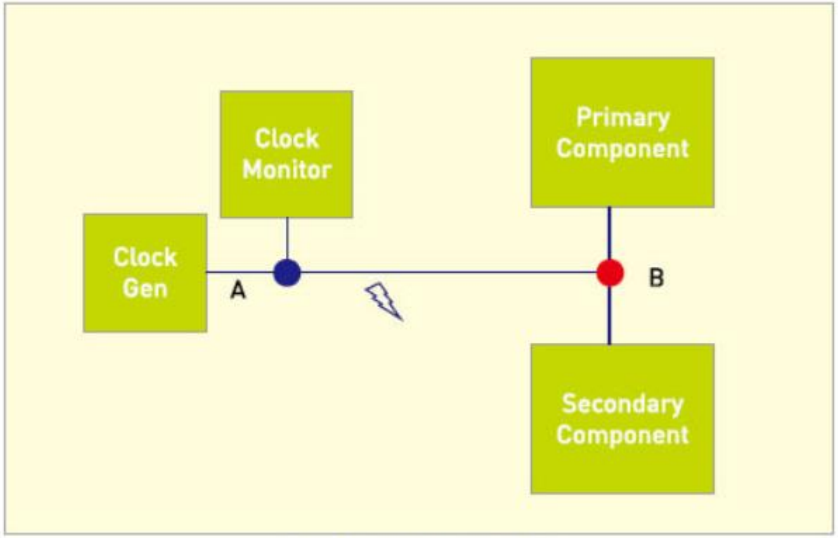



Separately located hardware device

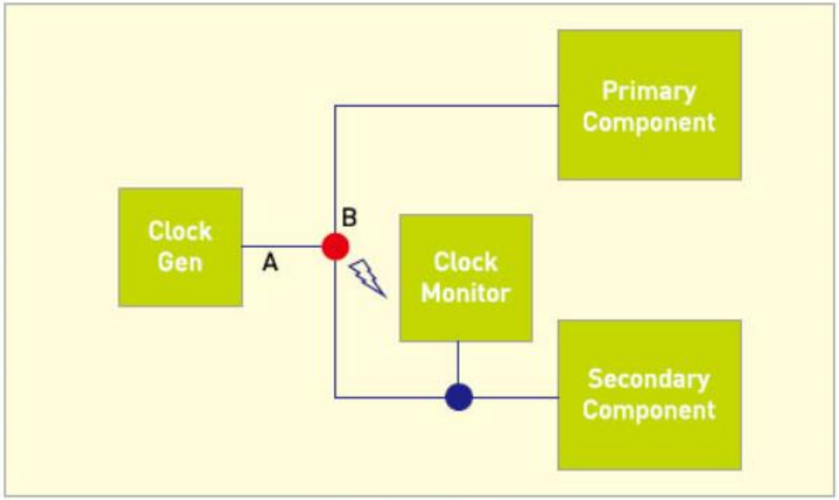
# • Examples of Reducing CCF

## Clock Tree & Clock Monitoring (Design Deficiency)

- Split point(Red Dot) before the monitoring point(Blue Dot) is not found failures that occur in the Clock Tree path
- Failures in the path influence Spare parts




Clock Tree vulnerable to CCF




Improved Clock Tree

# • Analysis for Reducing CCF

**Use a Common Cause Failure list (Check List, IEC 61508)**

**Use diverse(unlike) redundancy when possible**

**Perform a Fault Tree Analysis (FTA)**

**The $\beta$-Factor Model, The C-Factor Model, Others**

UCLAB UBIQUITOUS COMPUTING LABORATORY | KU 건국대학교 KONKUK UNIV.

# • Analysis for Reducing CCF

## Use a Common Cause Failure list (Check List, IEC 61508)

| Hardware | Software | ASICs and FPGAs |
|---|---|---|
| <u>During design and implementation</u><br>1. Robust project management and documentation (throughout)<br>2. Structured specification, design<br>3. Observance of guidelines and standards<br>4. Functional testing, analysis<br>5. Operation and maintenance instructions, user- and maintenance-friendly<br>6. Interference testing<br>7. Fault insertion testing<br><br><u>During operation</u><br>1. Program sequence monitoring and on-line monitoring or testing<br>2. Power supply monitoring and protection<br>3. Spatial separation<br>4. Ambient temperature protection<br>5. Modification protection | 1. **<u>Functional safety assessment</u>**: checklists, truth tables, failure analysis, CCF analysis, reliability block diagrams<br>2. **<u>Software requirements specification</u>** – formal or semi-formal methods, traceability, software tools<br>3. Fault detection, error detecting codes<br>4. Diverse monitoring techniques<br>5. Recovery mechanisms or graceful degradation<br>6. Modular design<br>7. Trusted/verified software elements<br>8. **<u>Forwards/backwards traceability at all stages</u>**<br>9. Structured or semi-formal or formal methods, auto-code generation<br>10. Software tools<br>11. Guaranteed maximum cycle time, | 1. Structured description, VHDL design description and simulation, Boolean design description<br>2. Proven in use VHDL simulators and design environment<br>3. Functional testing on module and top levels, and embedded in system environment<br>4. Avoid asynchronous constructs, synchronised primary inputs<br>5. Design for testability; modularisation<br>6. Code guidelines adherence, code checker, defensive programming<br>7. Documentation of simulation results<br>8. Code inspection, walk-through<br>9. Validation of soft-cores<br>10. Internal consistency checks<br>11. Simulation of gate netlist to check timing constraints; static timing analysis of propagation delay |

# • Analysis for Reducing CCF

## Use a Common Cause Failure list (Check List, IEC 61508)

| | | |
|---|---|---|
| | time-triggered architecture, maximum response time<br>12. Static resource allocation, synchronisation<br>13. Language selection, suitable tools<br>14. Defensive programming, modular approach, coding standards, structured programming<br>15. <u>**Testing**</u>: dynamic, functional, black box, performance, model-based, interface, probabilistic<br>16. Process simulation, modelling<br>17. <u>**Modification/change control**</u>: impact analysis, re-verification, revalidation, regression testing, configuration management, data recording and analysis<br>17. <u>**Verification**</u>: Formal proof, static analysis, dynamic analysis, numerical analysis | 12. Verification of gate netlist<br>13. Check ASIC vendor requirements and constraints<br>14. Documentation of synthesis constraints, results and tools; use of proven in use tools and target libraries<br>15. Script based procedures<br>16. Test insertion and test pattern generation<br>17. Placement, routing, layout generation<br>18. Proven in use chip technology and manufacturing, QA, QC<br>19. Test coverage of manufacturing test; final verification and validation<br>20. Burn-in test |

# • Analysis for Reducing CCF

## Use diverse(unlike) redundancy when possible

- For example, **Nuclear Reactor Protection Systems**
- The diverse system design should be developed by a different team, using independently derived safety functional requirements
- The diverse system should be electrically and physically separated
- It should use different input sensors measuring diverse operating parameters
- Its signals should pass via separate routes and be processed by diverse types of logic solver
- Its final actuating devices (usually electrical breakers) should be from a different manufacturer
- Its means of shutdown should use different physical principles

# • Analysis for Reducing CCF

## Perform a Fault Tree Analysis (FTA)

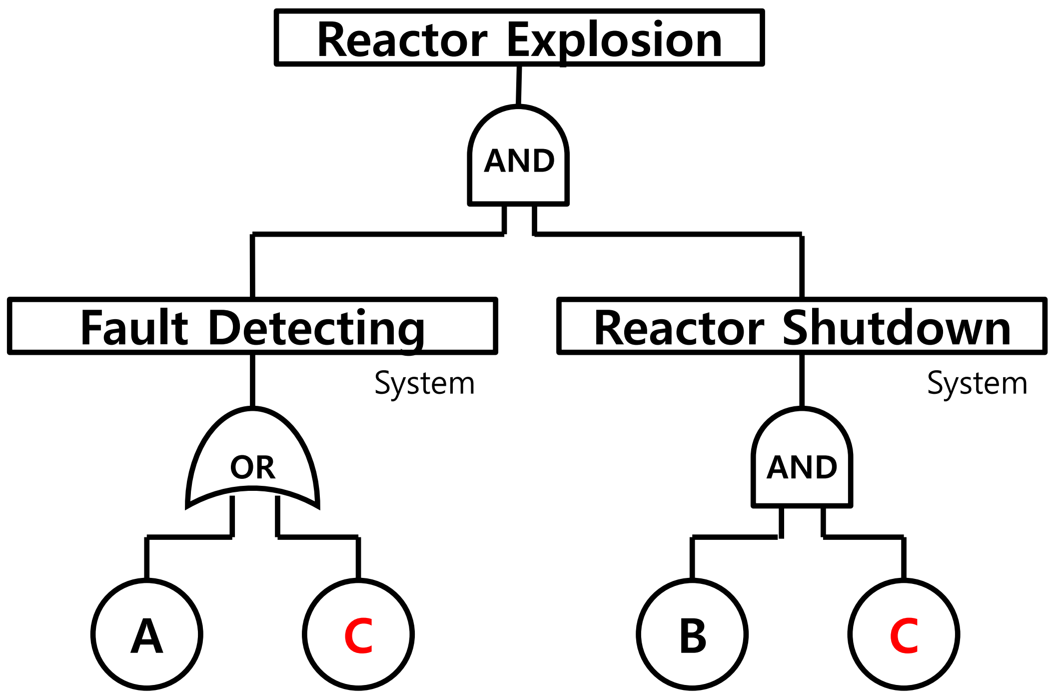
- Defines interactions and common failure paths
- Can be done on system level and can performed on subsystems or components that contain redundant items which are deemed susceptible

B & C occur

↓

Reactor Explosion

**Common Cause : C**



# • Analysis for Reducing CCF

## The $\boldsymbol{\beta}$-Factor Model

- The $\boldsymbol{\beta}$**-factor** model is the most commonly used CCF model
- This model assumes that a certain percentage of all failures are CCFs
- The **total failure rate** $\boldsymbol{\lambda}$ is split into an **independent part** $\boldsymbol{\lambda_I}$ and a **dependent part** $\boldsymbol{\lambda_C}$, such that

$$\lambda = \lambda_I + \lambda_C$$

- A $\boldsymbol{\beta}$**-factor** is defined as

$$\beta = \frac{\lambda_C}{\lambda}$$

- The value $\boldsymbol{\beta}$ can also be expressed as

$$\beta = P(CCF|Failure)$$

# • Analysis for Reducing CCF

## The $\beta$-Factor Model

- Consider a system of ***m*** similar items
- Each item failure can have two distinct causes :
  - An independent cause (i.e., a cause that only affects the specific item)
  - A shared cause that will affect all the *m* items – and cause all *m* to fail at the same time
- This means that the multiplicity of each CCF event must be either **1** or ***m***
- **It is not possible to have CCF events with intermediate multiplicities**

# • Analysis for Reducing CCF

## The $\beta$-Factor Model

- Consider a system of $m$ identical channels and assume that we have observed that a channel has failed

- The conditional probability that this is, in fact a CCF of multiplicity $k$ is

$$f_{1,m} = 1 - \beta$$

$$f_{k,m} = 0$$

$$f_{m,m} = \beta$$

for $k$ = 2, 3, ……, $m$ - 1

# Analysis for Reducing CCF

## The $\beta$-Factor Model

- The $\boldsymbol{\beta}$**-factor** model is simple and easy to understand and use
  - Since it has only one extra parameter ($\boldsymbol{\beta}$)
  - And it is easy to understand the meaning of this parameter
- The $\boldsymbol{\beta}$**-factor** model is the most commonly used CCF model
- The $\boldsymbol{\beta}$**-factor** model is preferred CCF model in IEC 61508

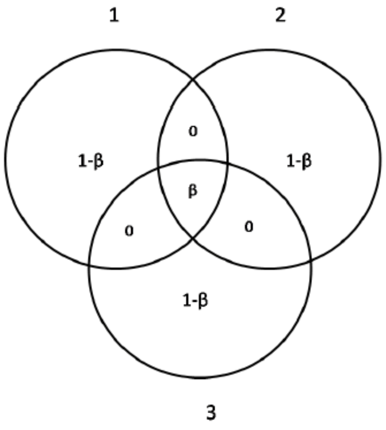
# • Analysis for Reducing CCF

## The $\beta$-Factor Model

- An effort to reduce an item's susceptibility to CCFs will reduce the parameter $\boldsymbol{\beta}$
  - But will at the same time increase the **rate of independent failures $\boldsymbol{\lambda_I}$**
  - Since $\boldsymbol{\lambda_I}$ is defined as

$$\lambda_I = (1 - \beta) \cdot \lambda$$

- If we have a system consisting of more than two components, the **$\boldsymbol{\beta}$-factor** model doesn't allow for the possibility that more than one
  - But not all components fail due to a CCF



# • Analysis for Reducing CCF

## The C-Factor Model

- The **C-Factor** model is mainly the same model as the $\boldsymbol{\beta}$**-factor** model
  - But the **rate of dependent failures,** $\boldsymbol{\lambda_C}$ is defined as **a fraction (C) of the independent failure rate,** $\boldsymbol{\lambda_I}$
  - Instead of as a fraction of the total failure rate (as is done in the $\boldsymbol{\beta}$**-factor** model), such that

$$\lambda = \lambda_I + C \cdot \lambda_I$$

- This means that an effort to reduce the item's susceptibility to CCFs will reduce the **total failure rate** $\boldsymbol{\lambda}$
  - And not as in the $\boldsymbol{\beta}$**-factor** model to increase the independent failure rate

# Analysis for Reducing CCF

## Others

- Basic Parameter Model
- Alpha-Factor Model
- Shock Models
  - The Multinomial Failure Rate Model
  - The Random Probability Shock Model
  - The Random Probability Shock Model
- Markov Analysis
  - The Matrix Multiplication method
  - The differential equations method

- ***Q & A***

- ***Thank You***