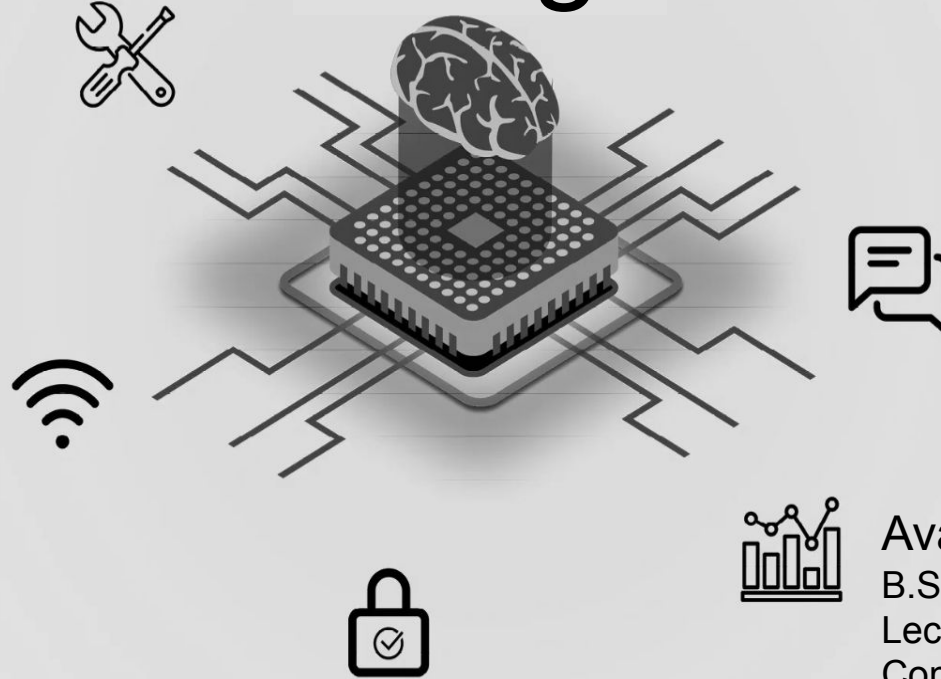


# EE6352 - Embedded System Design



Avanthi Jayasundara  
B.Sc.Eng.(Hons)(RUH, SL)  
Lecturer (probationary)  
Computer Engineering | Dept. of  
Electrical and Information Eng. |  
Faculty of Engineering

Email address: [avanthi@eie.ruh.ac.lk](mailto:avanthi@eie.ruh.ac.lk)

# Learning Outcome

LO5 : Design, assemble and build embedded systems for domestic and industrial applications

- **Design considerations in using embedded systems in industrial applications**
- **Embedded system design challenges**
- Bootloader, Watchdog timer
- Introduction to real-time operating systems

# We Are going to talk about ?

- System Reliability
- Dependability Models
- Reliability Block Diagram

# System Reliability



# Failure of Embedded Systems



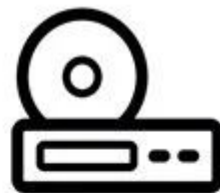
Industrial Robots



GPS Receivers



Digital Cameras



DVD Players



Wireless Routers

Embedded Systems



MP3 Players



Set top Boxes



Gaming Consoles



Photocopiers



Microwave Ovens

Different devices have different target lifetimes. However, a device may fail any time.

What is reliability ?

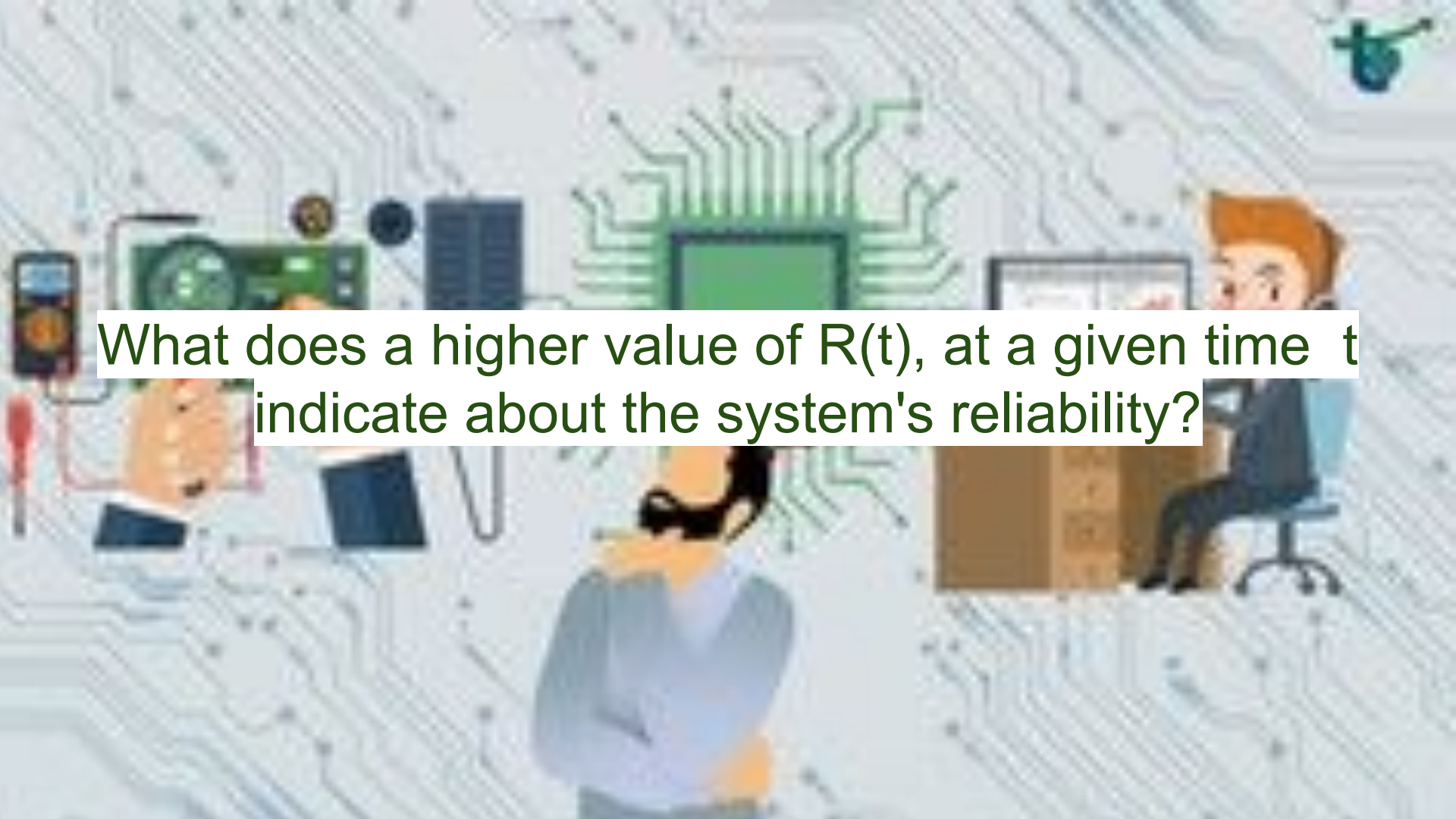
**“the probability that a device will last at least a specified time under specified conditions”**

$$R(t) = P(T > t)$$

$R(t)$  is the reliability function, which represents the probability that a system or component will operate without failure up to time  $t$ .

$T$  is a random variable representing the time to failure.

$P(T > t)$  denotes the probability that the time to failure  $T$  is greater than a specific time  $t$ .

The background is a light blue-grey color with a pattern of white circuit lines. In the center, there is a green computer monitor with a green screen. To the left of the monitor, there is a smartphone and a laptop. To the right, there is a person with brown hair sitting at a desk, looking at a laptop. In the foreground, there is a person with a beard and a blue shirt, looking at a laptop. The overall scene is a stylized illustration of a tech office or workspace.

What does a higher value of  $R(t)$ , at a given time  $t$  indicate about the system's reliability?

# Calculation...

For an embedded system component with an exponential failure distribution with a failure rate  $\lambda = 0.001$  failures per hour, the reliability function is:

$$R(t) = e^{-\lambda t} = e^{-0.001t}$$

Take  $t=100$



# Calculation

- At  $t = 100$  hours, the reliability  $R(100)$  is:

$$R(100) = e^{-0.001 \times 100} = e^{-0.1} \approx 0.905$$

This means there is approximately a 90.5% probability that the component will function without failure for at least 100 hours.

# Guarantee Time Period

## Definition:

- The guarantee (or warranty) time period is the duration for which a manufacturer promises that a product will perform without failure. If a failure occurs within this period, the manufacturer typically offers repair, replacement, or refund.

## Relationship Between Reliability and Guarantee Time:

- The guarantee period is often based on reliability data. Manufacturers set a guarantee time period to balance customer satisfaction and cost considerations.
- Manufacturers use reliability data to determine a time period during which the likelihood of failure is acceptably low.
- For an exponential distribution with failure rate  $\lambda$ , if a manufacturer guarantees the product for time  $t_g$ , they might aim for  $R(t_g)$  to be above a certain threshold, e.g., 95% reliability:

### Example:

- If  $\lambda = 0.001$  per hour and the desired reliability is 95% ( $R(t_g) = 0.95$ ):

$$0.95 = e^{-0.001t_g}$$

$$\ln(0.95) = -0.001t_g$$

$$t_g = -\frac{\ln(0.95)}{0.001} \approx 51.3 \text{ hours}$$

The manufacturer might set a guarantee period slightly above this to ensure high customer satisfaction, possibly at 50 hours.



Suppose that the probability density function (PDF) of the Failure rate of a device is given by

$$f(t) = \frac{200}{(t + 10)^3}$$

where  $t$  is in years

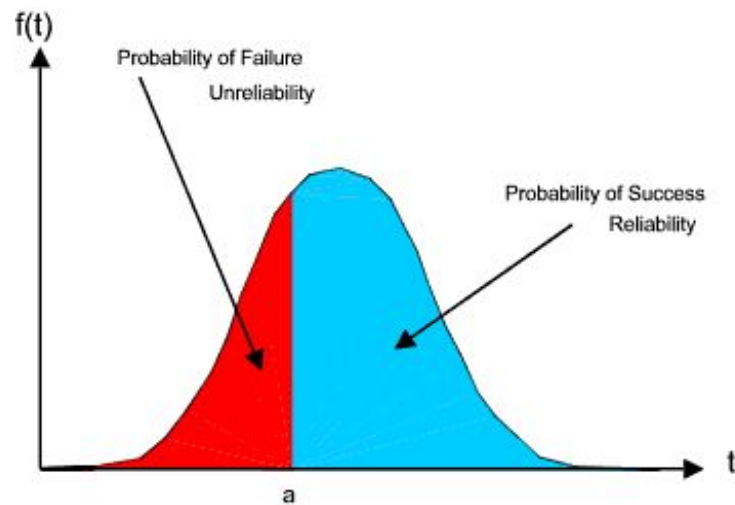
Suppose that the probability density function (PDF) of the Failure rate of a device is given by

$$f(t) = \frac{200}{(t + 10)^3} \quad \text{where } t \text{ is in years}$$

If the warranty period is given as 1 Year, what is the probability that the device will last until the expiry of the warranty period?

$$P(t > 1) = \int_1^{\infty} \frac{200}{(t + 10)^3} dt = 0.8264$$

More than 1 year

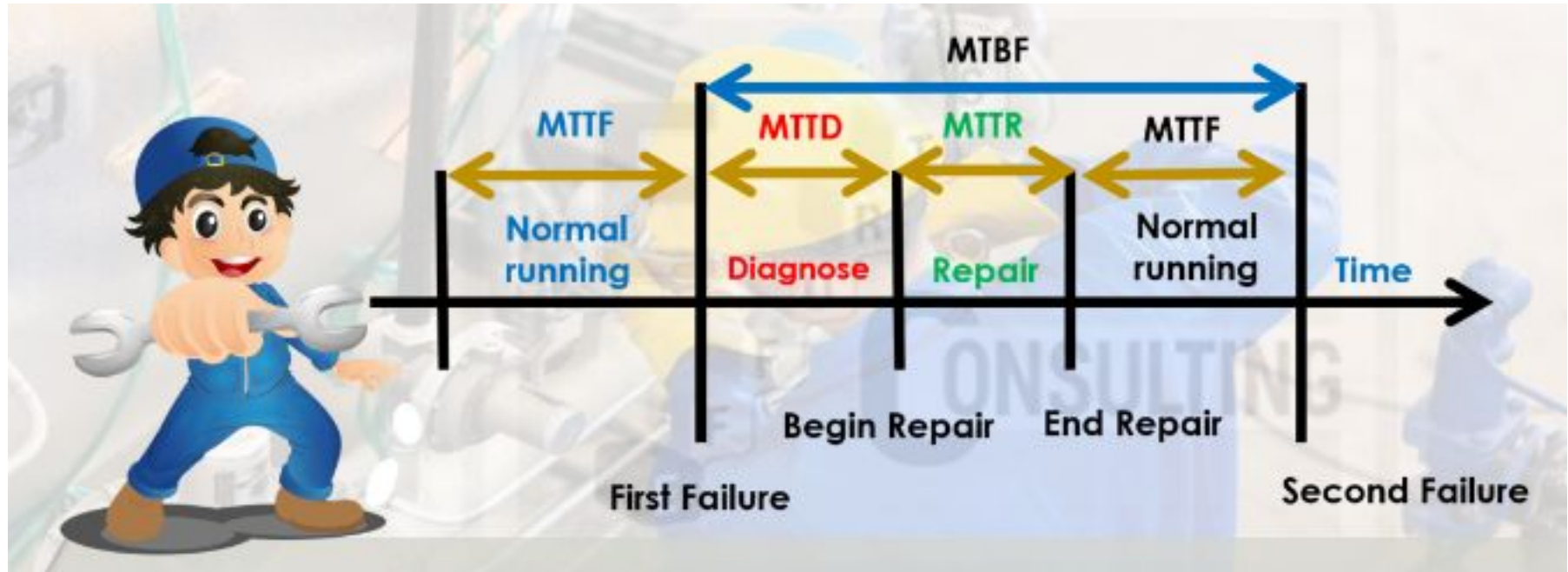


Time-to-failure (Random Function)

# Reliability Can be given using a number of measures

**MTTF (Mean Time to Failure):** Average time to failure for **non-repairable** systems.

**MTBF (Mean Time Between Failures):** Average time between failures for **repairable** systems.



The MTTF is the integral of time  $t$  multiplied by the pdf  $f(t)$  over the range from 0 to infinity:

$$\text{MTTF} = \int_0^{\infty} t \cdot f(t) dt$$



# Failure Rates in Embedded Systems:

- Two Types:
  - **Hardware Failure Rates**
  - **Software Failure Rates**

## Hardware Failure Rates:

### Probability Curve:

- Characterized by a well-known probability curve.
- Consists of three distinct periods: high initial failure, constant useful life, and increasing wear out period.

### Initial High Failure Period:

- Newly manufactured devices have a short period with a high probability of failure.
- Manufacturers often perform a "burn-in" process during this period.
- Devices that fail during burn-in are reworked and only those that survive are shipped.

## Useful Life Period:

- After burn-in, the hardware enters a useful life period.
- Probability of failure is constant and typically low during this period.

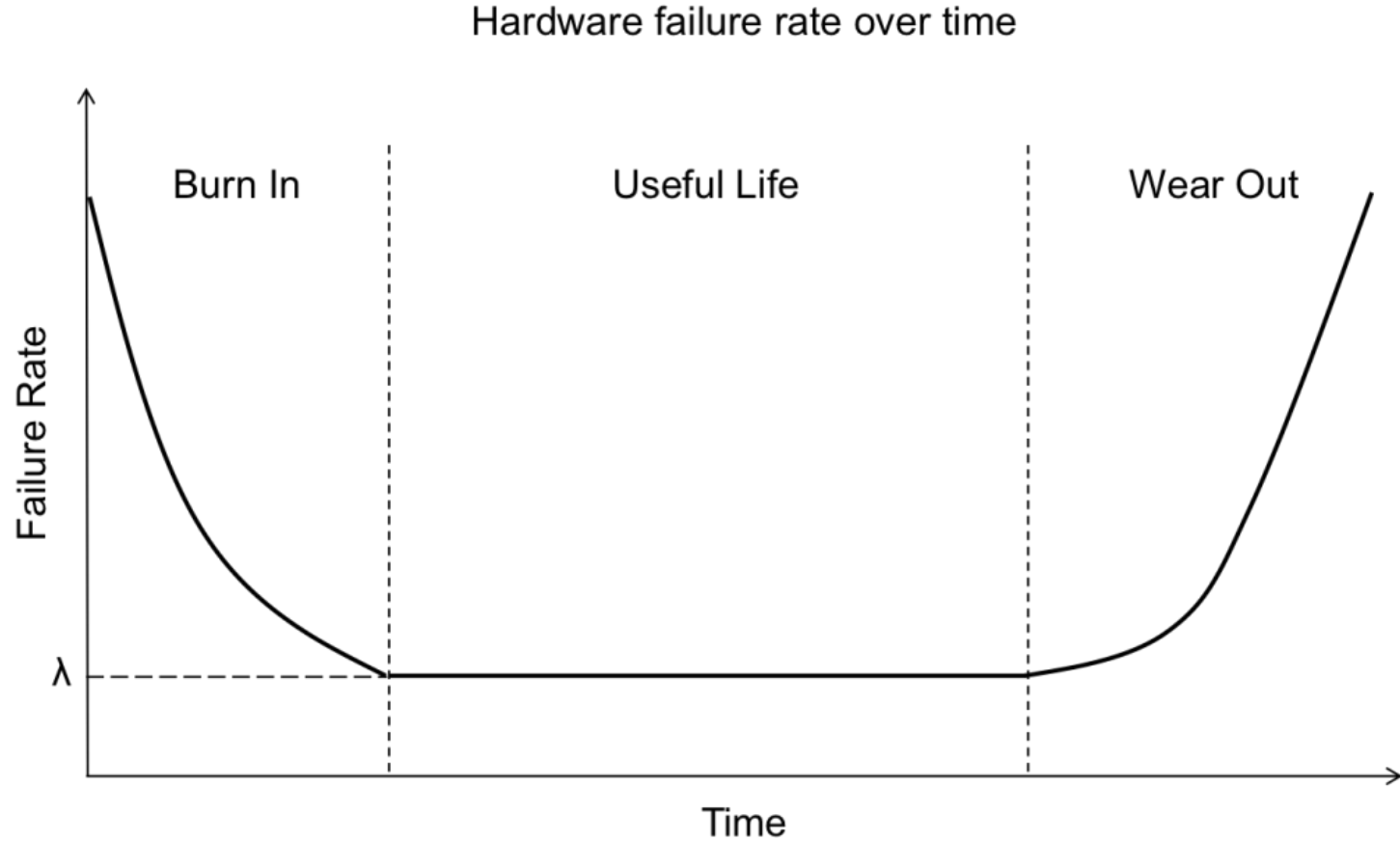
## Wear Out Period:

- After the useful life, the device enters the wear out period.
- Probability of failure increases due to aging and the finite lifetimes of physical components.

## Key Points:

- **Burn-in Process:** Used to eliminate early failures and ensure only reliable devices are shipped.
- **Useful Life:** Period with the lowest and most stable failure rates.
- **Wear Out:** Increasing failure rates due to physical component degradation.

# Failure Rate



# Software Failure Rates:

## 1. **Probability Curve:**

- Varies drastically compared to hardware failure rates.
- Starts with a high probability of failure during the Test/Debug phase.

## 2. **Test/Debug Phase:**

- Equivalent to the hardware burn-in phase.
- High initial probability of failure as bugs and issues are identified and resolved.

## 3. **Decreasing Failure Rate:**

- As software undergoes testing, the probability of failure decreases.
- This process continues until the failure rate is at an acceptable level for the product to ship.

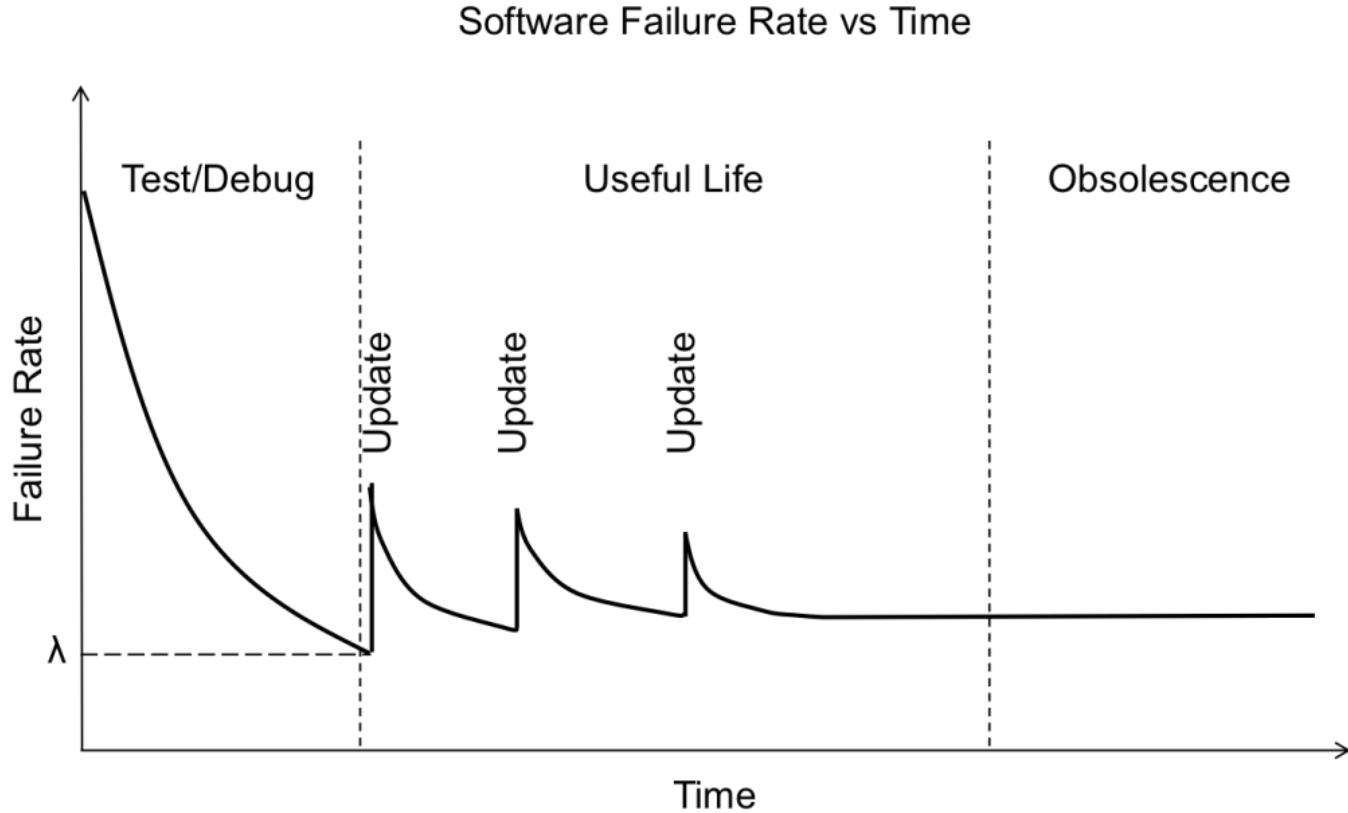
## 4. **Useful Life:**

- Once the software is stable and the failure rate is acceptable, the product enters its useful life stage.
- During this stage, the software maintains a low probability of failure, similar to the hardware useful life stage.

## Key Points:

- **Test/Debug Phase:** Critical for identifying and fixing bugs, reducing the failure rate.
- **Useful Life:** Software maintains low failure probability once it reaches an acceptable stability level for shipping.

# Failure Rate



## Software Update Characteristics:

- Unlike hardware, software can be updated at any time in the field.
- Each firmware update can cause a spike in the system's failure probability.

## Reasons for Increased Failure Rates Post-Update:

- **Incomplete Updates:** Failure to successfully complete the update.
- **Untested Features:** New features may not be fully tested.
- **Incomplete Regression Testing:** Regression testing might not be completed.
- **Field Variations:** Devices in the field may be in different states.
- **Security Vulnerabilities:** New updates might introduce new security issues.
- **Insufficient Testing Time:** Not spending enough time testing the update.

# Practical Recommendations for Developers

- 1. Make Small Incremental Updates:**
  - Smaller updates reduce the risk of introducing significant issues and make it easier to isolate and fix problems.
- 2. Perform Full Regression Testing:**
  - Ensure that the new update does not negatively impact existing functionalities by thoroughly testing all aspects of the software.
- 3. Test on Diverse Devices:**
  - Test updates on multiple devices, including at least one from each manufactured batch, to account for variations in hardware.
- 4. Extended Testing Duration:**
  - Run the updated devices continuously for at least 72 hours to identify any issues that might not be immediately apparent.
- 5. Use Automated Testing Systems:**
  - Implement automated systems to continuously operate and monitor the device during testing, ensuring comprehensive and consistent testing processes.

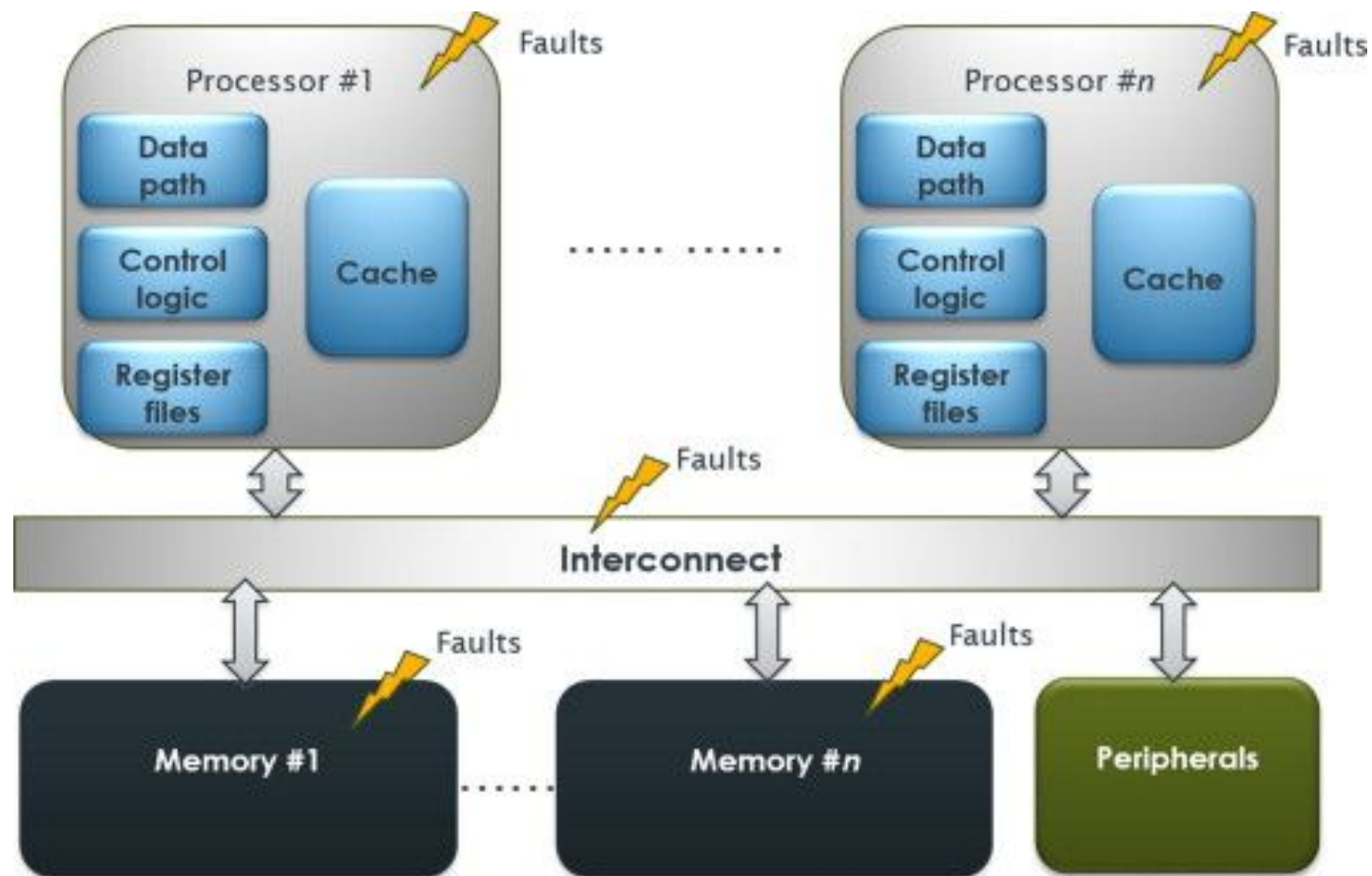
# Dependability Models



Dependability models are to **capture the conditions that make a system fail**, in terms of the **structural relationships** between the **system components**.

For a system consisting of  $n$  components,  
Every component can be in one of the two conditions:  
**working or failed**

- How many possible combinations of the status of these  $n$  components?
- How do you calculate the reliability of the system given the probability of each component being working (or failed)?

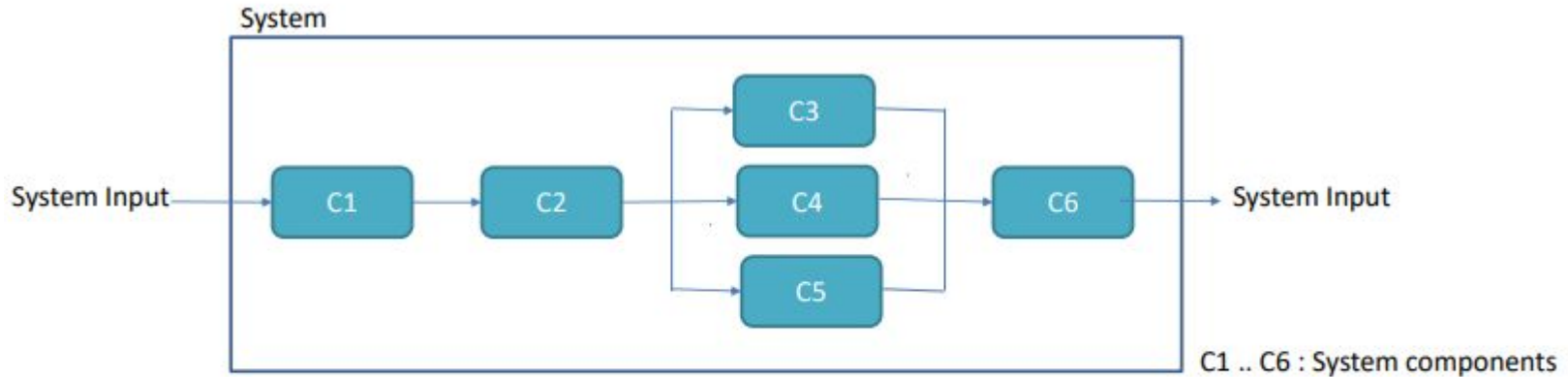


# Dependability Models

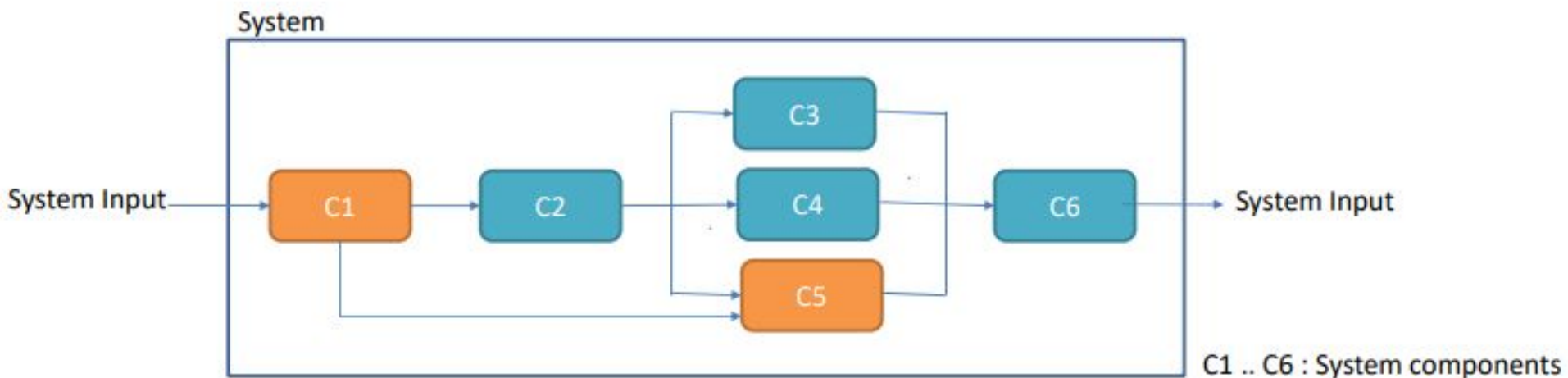
1. Reliability Graph
2. Fault Tree Model
- 3. Reliability Block Diagram**

# Reliability Block Diagram

Reliability Block Diagram (RBD) is a graphical representation of how the components of a system are connected from reliability point of view.



Components may have a straightforward structural relationship



Components may have a **complex** structural relationship

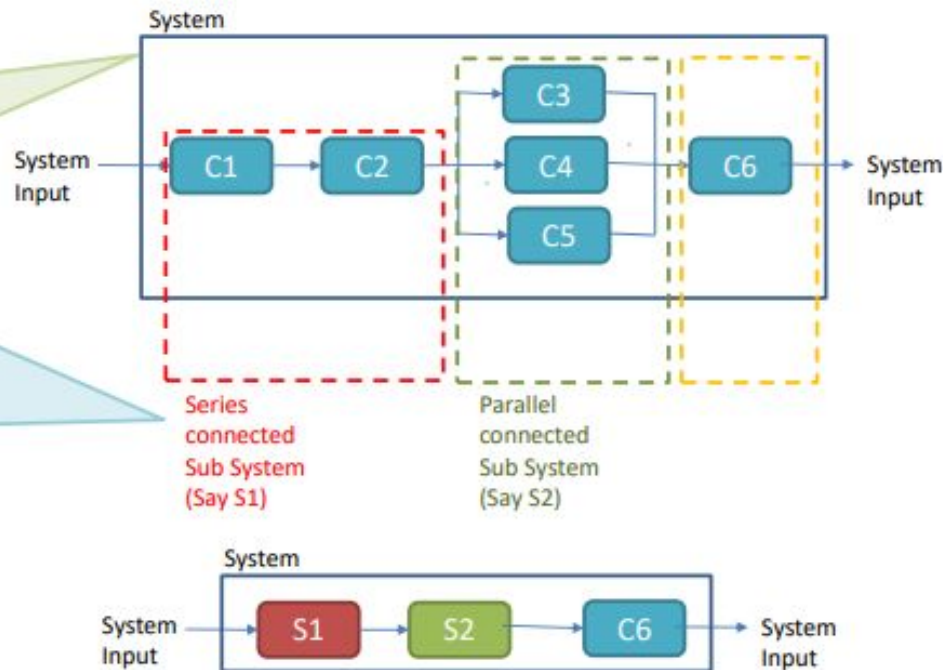
# Reliability Block Diagram

Reliability of the system is derived in terms of reliabilities of its individual components.

System needs to be broken down into simple structural configurations

The most common configurations of an RBD are the series and parallel configurations

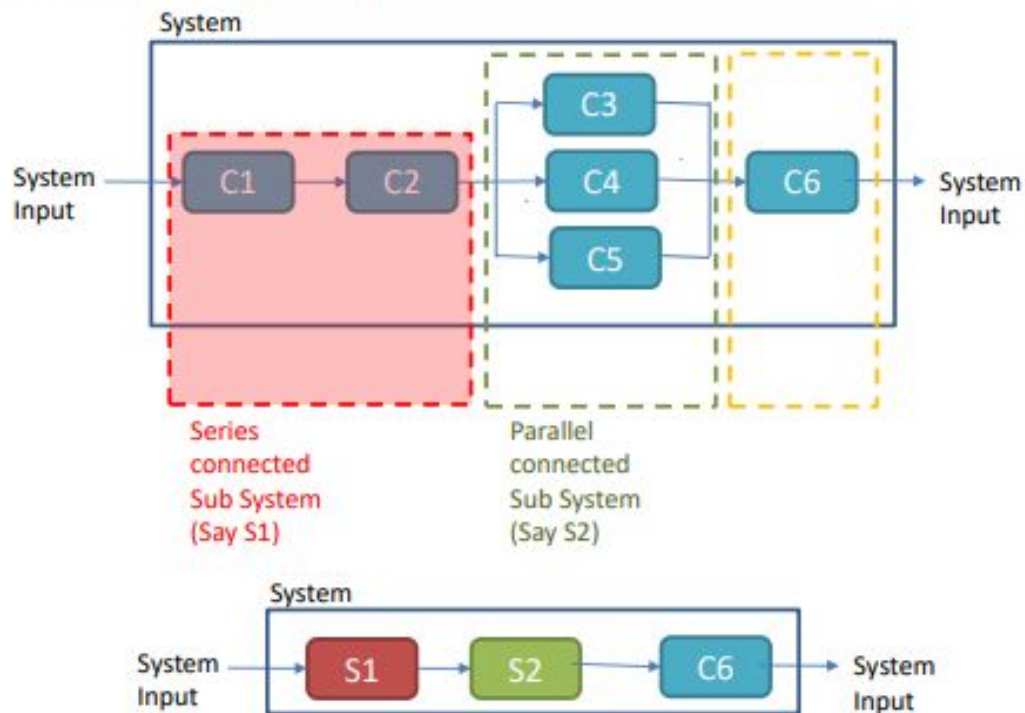
RBD analysis is typically used for determining reliability, availability and down time of the system



# Reliability Block Diagram : serial Configuration

In a serial system configuration, the elements must all work for the system to work and the system fails if one of the components fails.

The overall reliability of a serial system is lower than the reliability of its individual components.

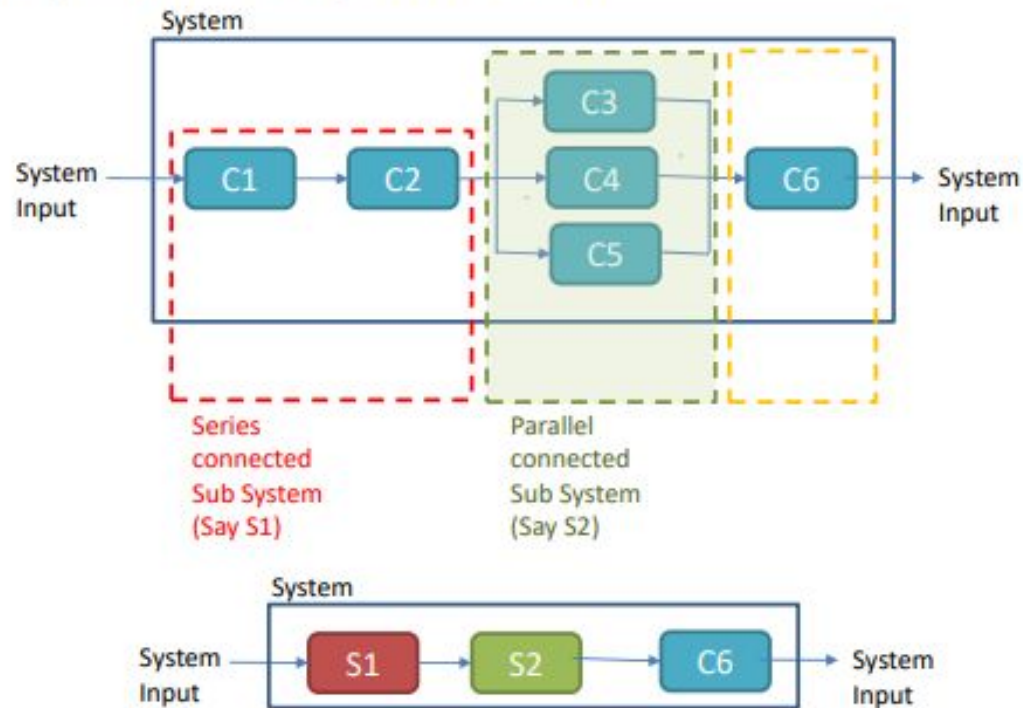




# Reliability Block Diagram : Parallel Configuration

In parallel configuration, the components are considered to be redundant and the system will still cease to work if all the parallel components fail.

The overall reliability of a parallel system is higher than the reliability of its individual components.





# Steps to follow ?

- Define boundary of the system for analysis
- Break system into functional components
- Determine serial-parallel combinations
- Represent each components as a separate block in the diagram
- Draw lines connecting the blocks in a logical order for mission success

# Disadvantages

Some complex constructs, such as standby, branching and load sharing, etc., cannot be clearly represented using the traditional RBD constructs.

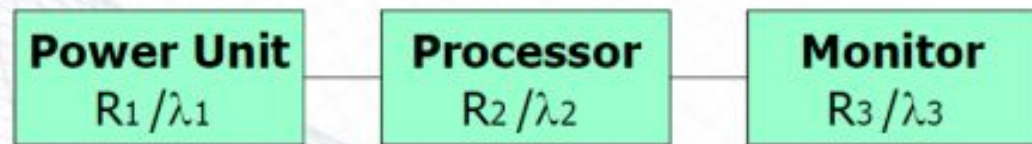
There are a number of automated tools, integrated with the other methods, such as Fault Tree Analysis (FTA), to generate the diagram and to analyze it.

# Serial System Reliability

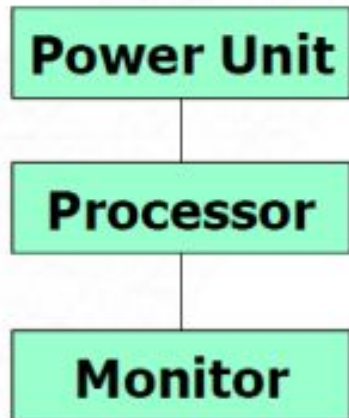
Consider the given computer system given in the figure

It consists of three blocks in series.

The reliability block diagram can be given as



System block diagram

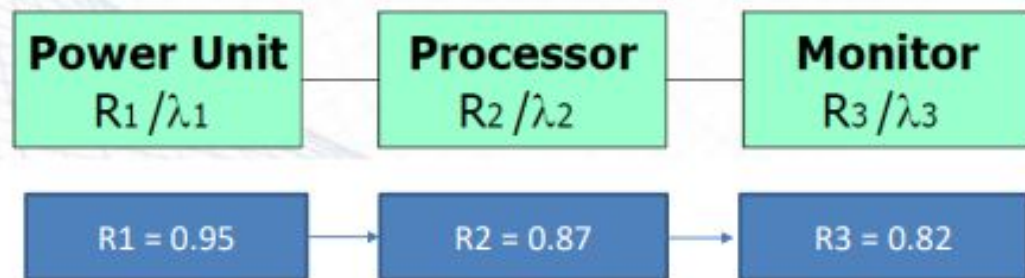


$R_i$  : Reliability of block  $i$

$\lambda_i$  : Failure rate of block  $i$

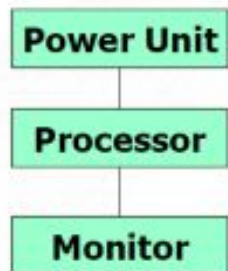
# Serial System Reliability

$$R = \prod_{k=1}^N R_k$$

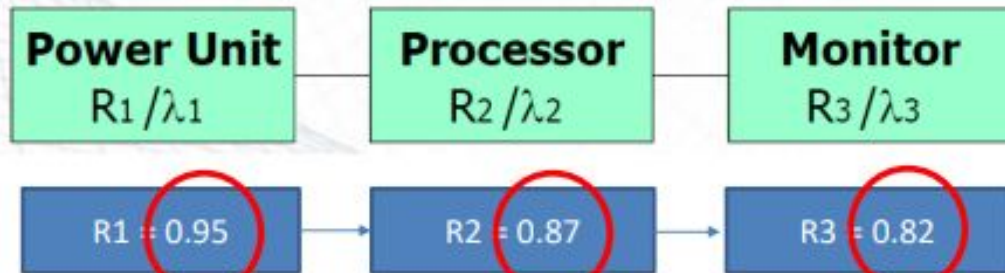


$$\begin{aligned} R &= \prod_{k=1}^3 R_k = R_1 R_2 R_3 \\ &= 0.95 \times 0.87 \times 0.82 \\ &= 0.677 \end{aligned}$$

# Serial System Reliability



$$R = \prod_{k=1}^N R_k$$



The overall reliability of a serial system is lower than the reliability of its individual components

$$R = \prod_{k=1}^3 R_k = 0.95 \times 0.87 \times 0.82 = 0.677$$

# Parallel System Reliability

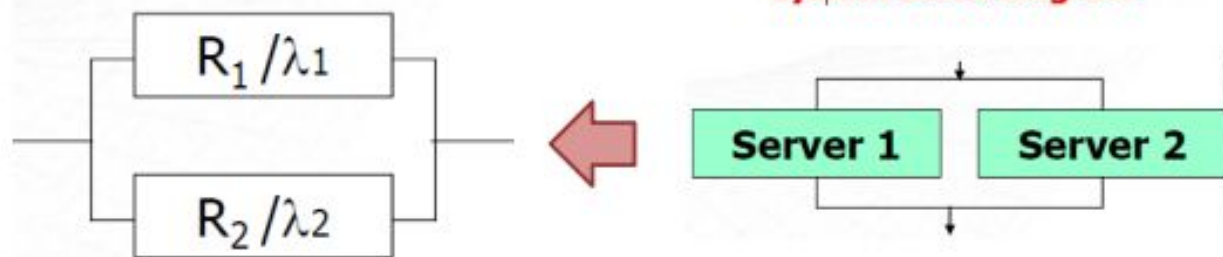
Consider the given computer system given in the figure

It consists of two blocks in parallel.

The reliability block diagram can be given as



**System block diagram**



$R_i$  : Reliability of block  $i$

$\lambda_i$  : Failure rate of block  $i$

# Parallel System Reliability

Consider the given computer system given in the figure

It consists of two blocks in parallel.

The reliability block diagram can be given as



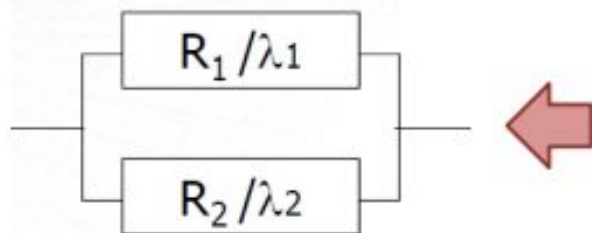
## System block diagram

Consider the given computer system given in the figure  
It consists of two blocks in parallel.

The reliability block diagram can be given as

Unreliability  $F = \prod_{i=1}^N F_i$

Reliability  $R = 1 - F = 1 - \prod_{i=1}^N (1 - R_i)$



Unreliability  $F = \prod_{k=1}^N F_i$

Reliability  $R = 1 - F = 1 - \prod_{k=1}^N (1 - R_i)$



# Parallel System Reliability

$$R_1 = 0.6777$$

$$R_2 = 0.6777$$



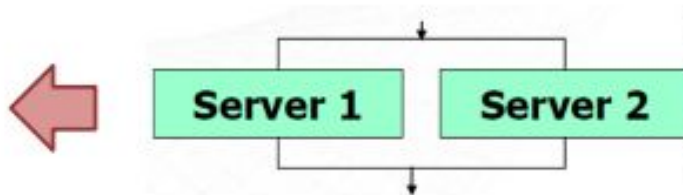
$$\text{Reliability } R = 1 - \prod_{k=1}^N (1 - R_i)$$

$$= 1 - (1 - 0.6777) \times (1 - 0.6777)$$

$$= 0.8961$$

$$R_1 = 0.6777 \quad R_2 = 0.6777$$

$$\begin{aligned} \text{Reliability } R &= 1 - \prod_{k=1}^N (1 - R_i) \\ &= 1 - (1 - 0.6777) \times (1 - 0.6777) \\ &= 0.8961 \end{aligned}$$





# Parallel System Reliability

$$R_1 = 0.6777$$

$$R_2 = 0.6777$$

$$\text{Reliability } R = 1 - \prod_{k=1}^N (1 - R_i)$$

$$= 1 - (1 - 0.6777) \times (1 - 0.6777)$$

$$= 0.8961$$

$$R_1 = 0.6777 \quad R_2 = 0.6777$$

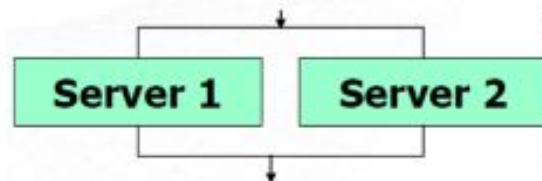
$$\text{Reliability } R = 1 - \prod_{k=1}^N (1 - R_i)$$

$$= 1 - (1 - 0.6777) \times (1 - 0.6777)$$

$$= 0.8961$$



The overall reliability of a serial system ??? than the reliability of its individual components



# System Reliability

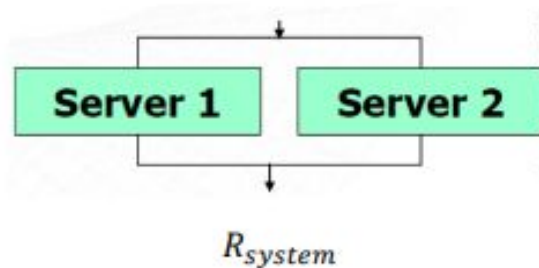
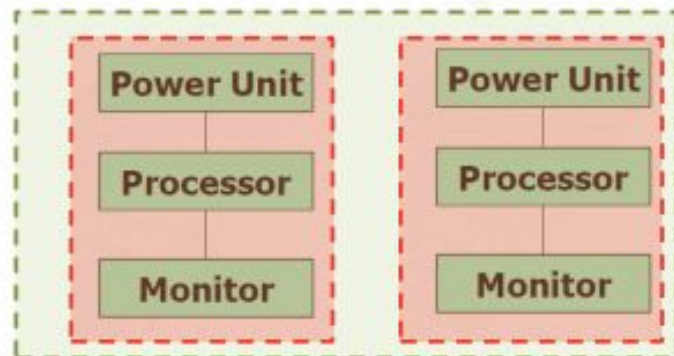
$$R_{power\_unit} = 0.95$$

$$R_{processor} = 0.87$$

$$R_{monitor} = 0.82$$

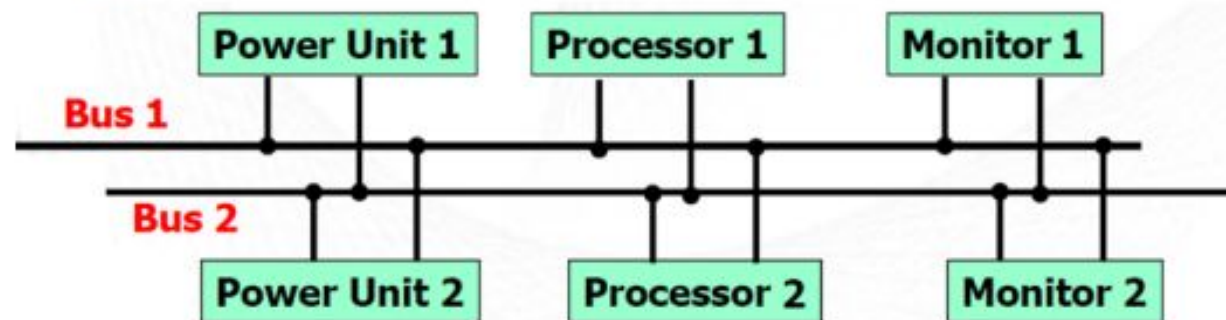
$$R_{server} = 0.95 \times 0.87 \times 0.82 = 0.6777$$

$$\begin{aligned} R_{system} &= 1 - (1 - R_{server}) \times (1 - R_{server}) \\ &= 1 - (1 - 0.6777) \times (1 - 0.6777) \\ &= 0.8961 \end{aligned}$$

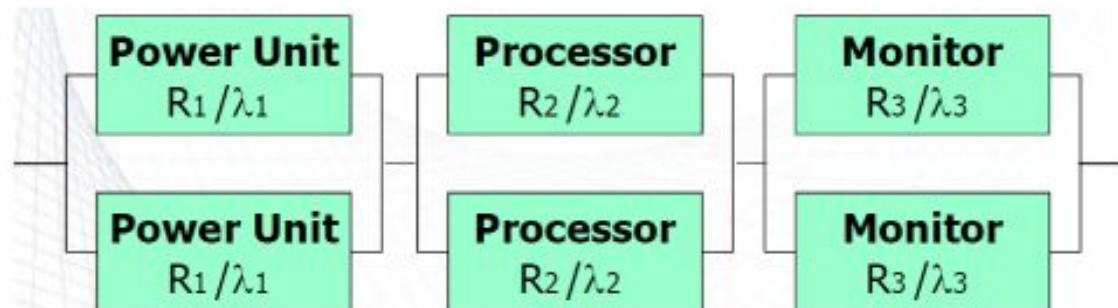


# Series-parallel Configuration

Two busses having redundant components

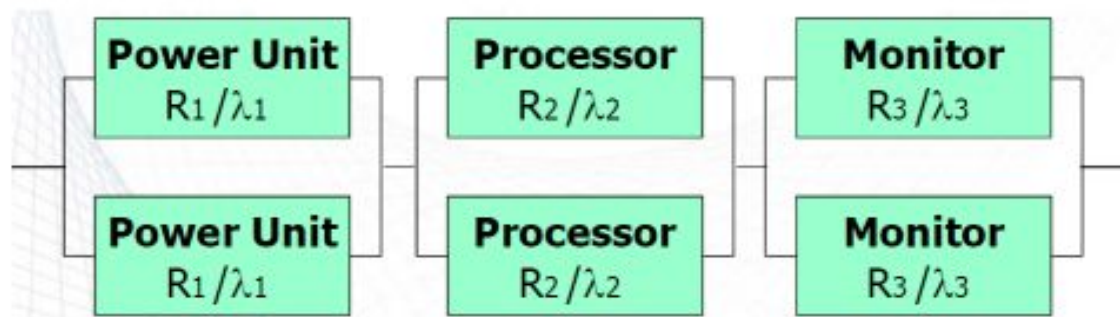


Reliability Block Diagram for Series-Parallel configuration (for each bus: busses are not redundant)



# Series-parallel Configuration

Reliability Block Diagram for Series-Parallel configuration (for each bus: busses are not redundant)



$$R_{Power} = 1 - (1 - R_1) \times (1 - R_1) = 1 - (1 - 0.95) (1 - 0.95) = 0.9975$$

$$R_{Processor} = 1 - (1 - R_2) \times (1 - R_2) = 1 - (1 - 0.87) (1 - 0.87) = 0.9831$$

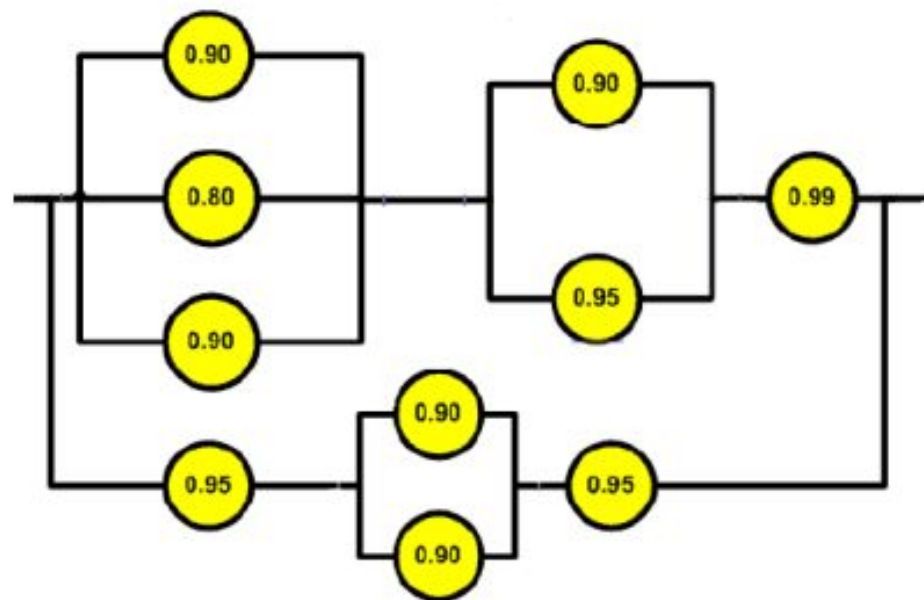
$$R_{Monitor} = 1 - (1 - R_1) \times (1 - R_1) = 1 - (1 - 0.82) (1 - 0.82) = 0.9676$$

$$R = R_{Power} \times R_{Processor} \times R_{Monitor} = 0.9975 \times 0.9831 \times 0.9676 = 0.9488694411$$

# Reliability Block Diagram : Exercise 2

A mixed mode system composed of serial and parallel components is shown in the figure.

Calculate the overall system reliability given the reliability for each module.





# Reliability Block Diagram : Exercise 2

A mixed mode system composed of serial and parallel components is shown in the figure.

Calculate the overall system reliability given the reliability for each module.

Identify the set of subsystems R1 to R5

$$R1 = 1 - (1 - 0.9)(1 - 0.8)(1 - 0.9) = 0.998$$

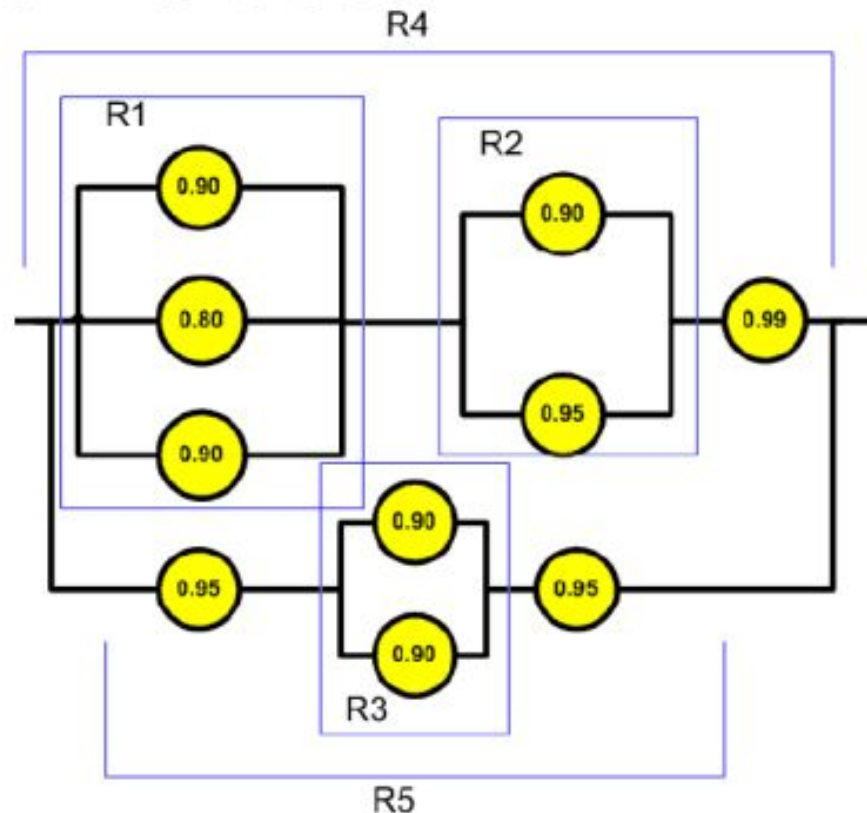
$$R2 = 1 - (1 - 0.9)(1 - 0.95) = 0.995$$

$$R3 = 1 - (1 - 0.9)(1 - 0.9) = 0.990$$

$$R4 = R1 \cdot R2 \cdot (0.99) = 0.983$$

$$R5 = (0.95) \cdot R3 \cdot (0.95) = 0.893$$

$$R = 1 - (1 - R4)(1 - R5) = 0.998$$



# Hazard Analysis in Embedded Systems

## Goal:

- **Identify Events Leading to Accidents:** The primary goal is to detect potential hazards that could cause accidents within embedded systems.
- **Identify Vulnerable Elements/Operations:** Determine specific components or operations in the system prone to failure (Single Point Failures).
- **Determine Impact on System:** Assess how these failures could affect the overall system functionality and safety.

# Techniques:

- **FMEA (Failure Modes and Effects Analysis):** This technique involves systematically identifying potential failure modes within the system and their effects on system operations, prioritizing them based on severity, occurrence, and detection.
- **FMECA (Failure Modes, Effects, and Criticality Analysis):** An extension of FMEA, this method adds a criticality analysis to evaluate the significance of each failure mode in terms of its impact on system reliability and safety.
- **ETA (Event Tree Analysis):** This technique uses a graphical representation to map out potential event sequences following an initiating event, helping to identify possible outcomes and their probabilities.
- **FTA (Fault Tree Analysis)**
- **HAZOP (HAZard and OPerability Studies)**