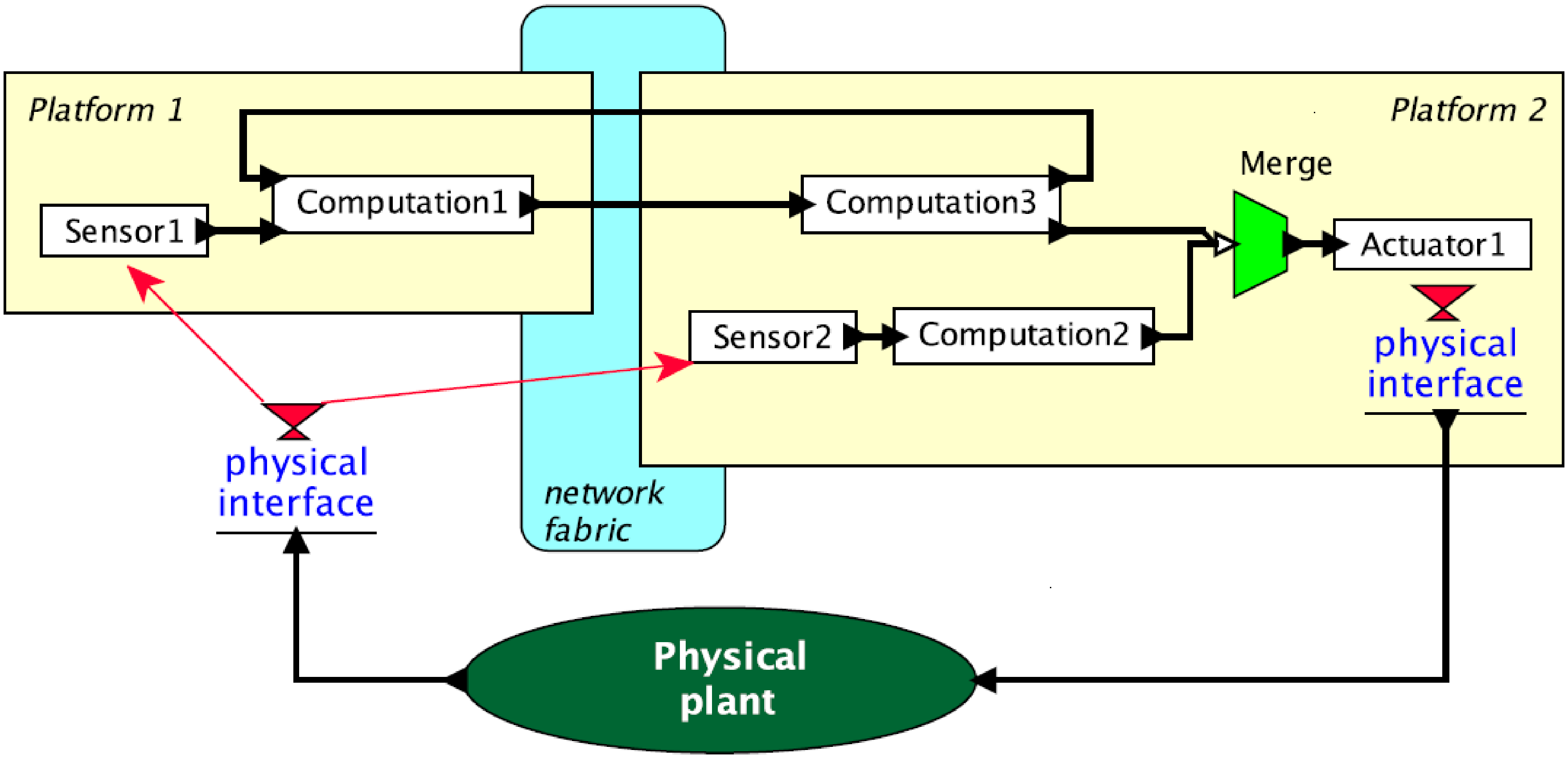
Embedded Communication Networks

Arnab Sarkar
Advanced Technology Development Centre
IIT Kharagpur

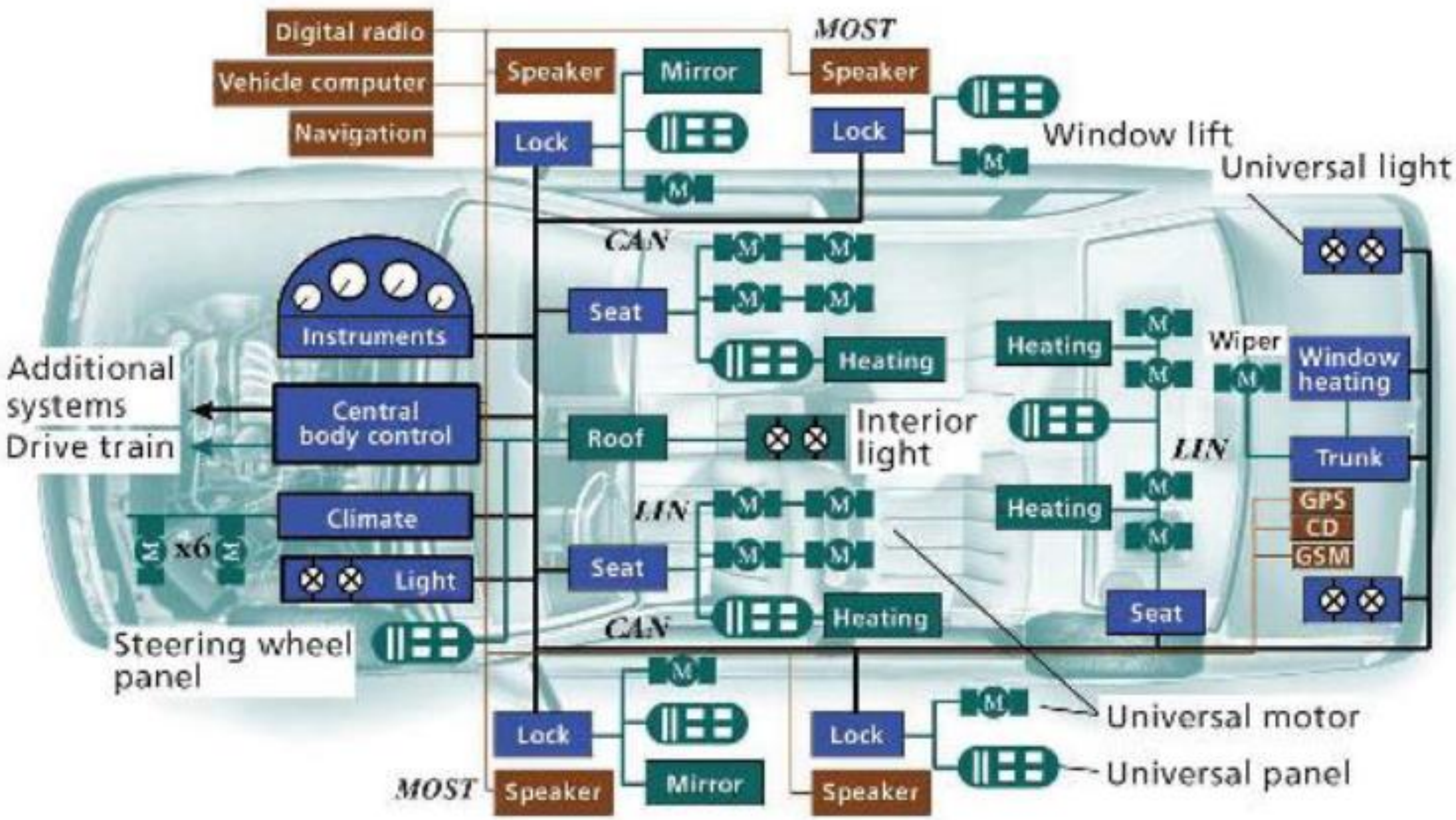
Overview of Topics to be Covered

- Basics of Fault-Tolerant Computing [3-4 Lectures]
- Fault Tolerance Analysis of Embedded Communication Networks [6-7 Lectures]
- Ethernet and Time triggered Ethernet, Time Sensitive Networking (TSN), Industrial Control Network Design using TSN [6-7 Lectures]
- Exam [1], Assignments [2-3]

What is a Cyber-Physical System



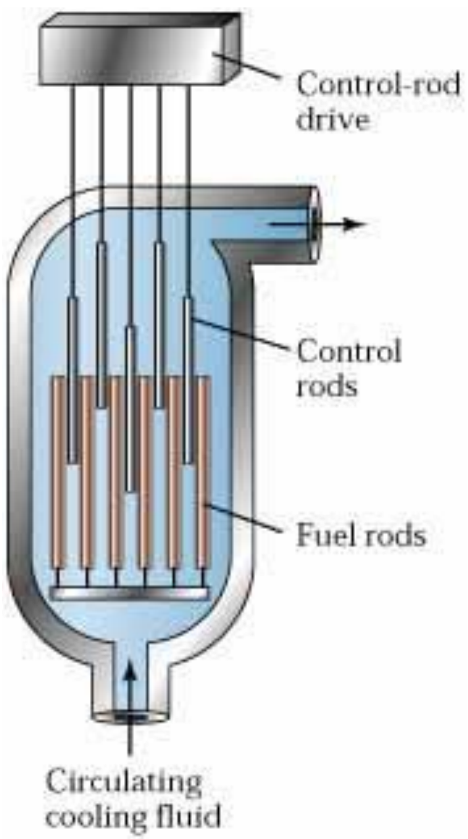
CPS Examples



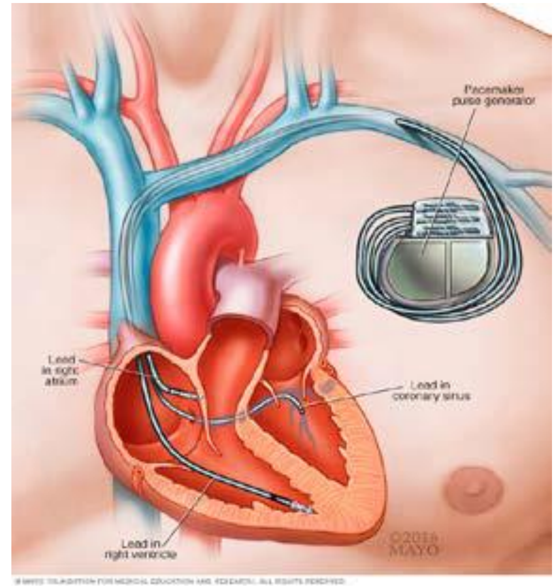
Automotive Control Systems

Ref: Image Taken from Internet Sources

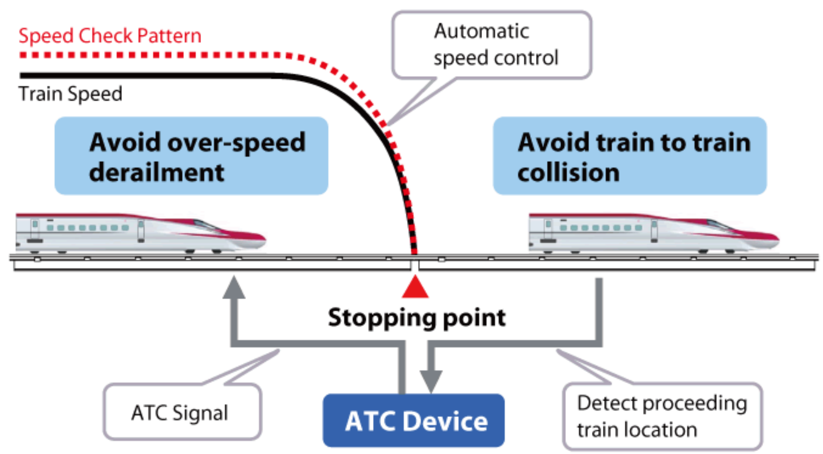
CPS Examples



Atomic Reactors



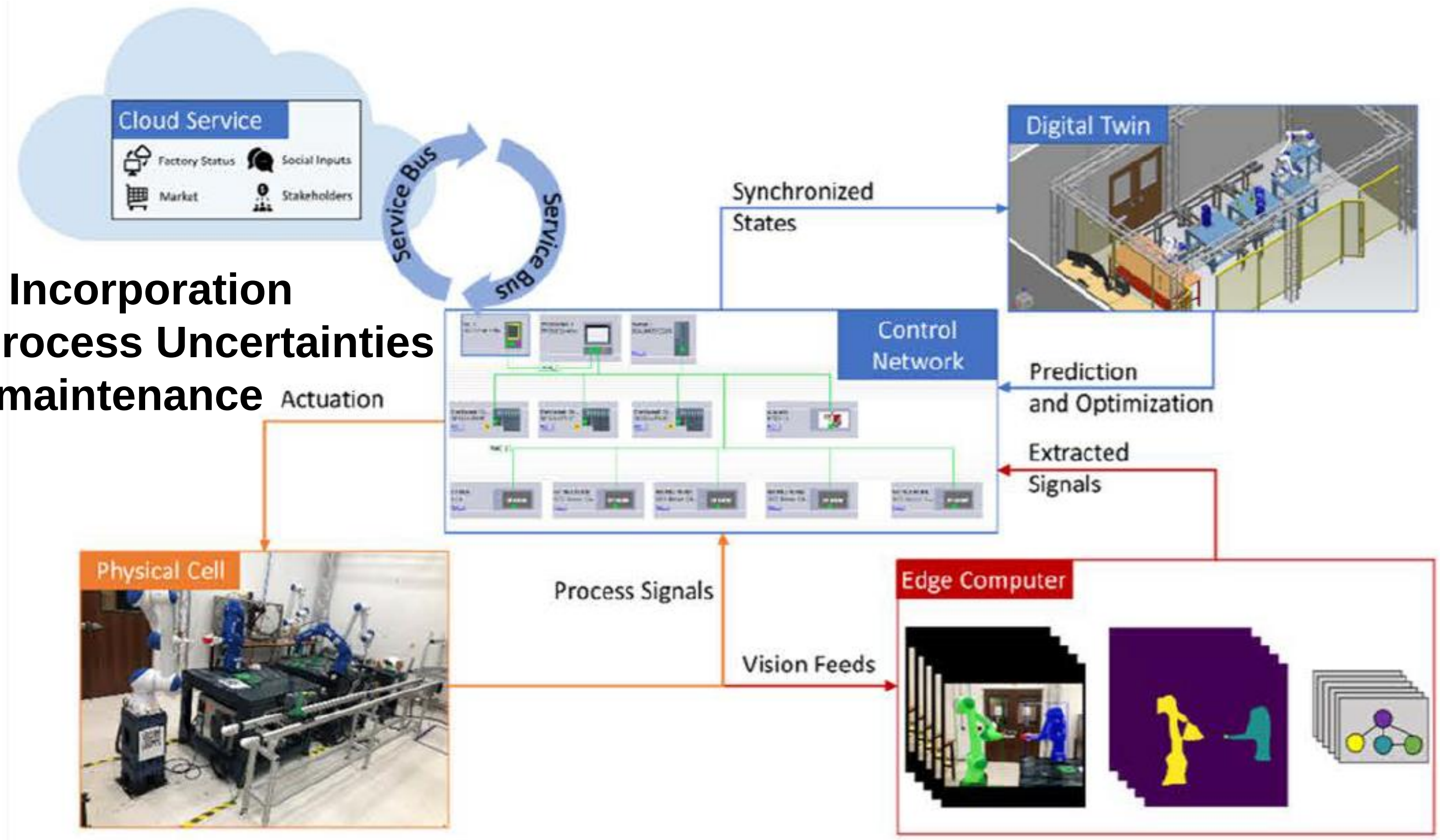
Health Care Devices



Train Control Systems

Ref: Images Taken from Internet Sources

CPS Examples

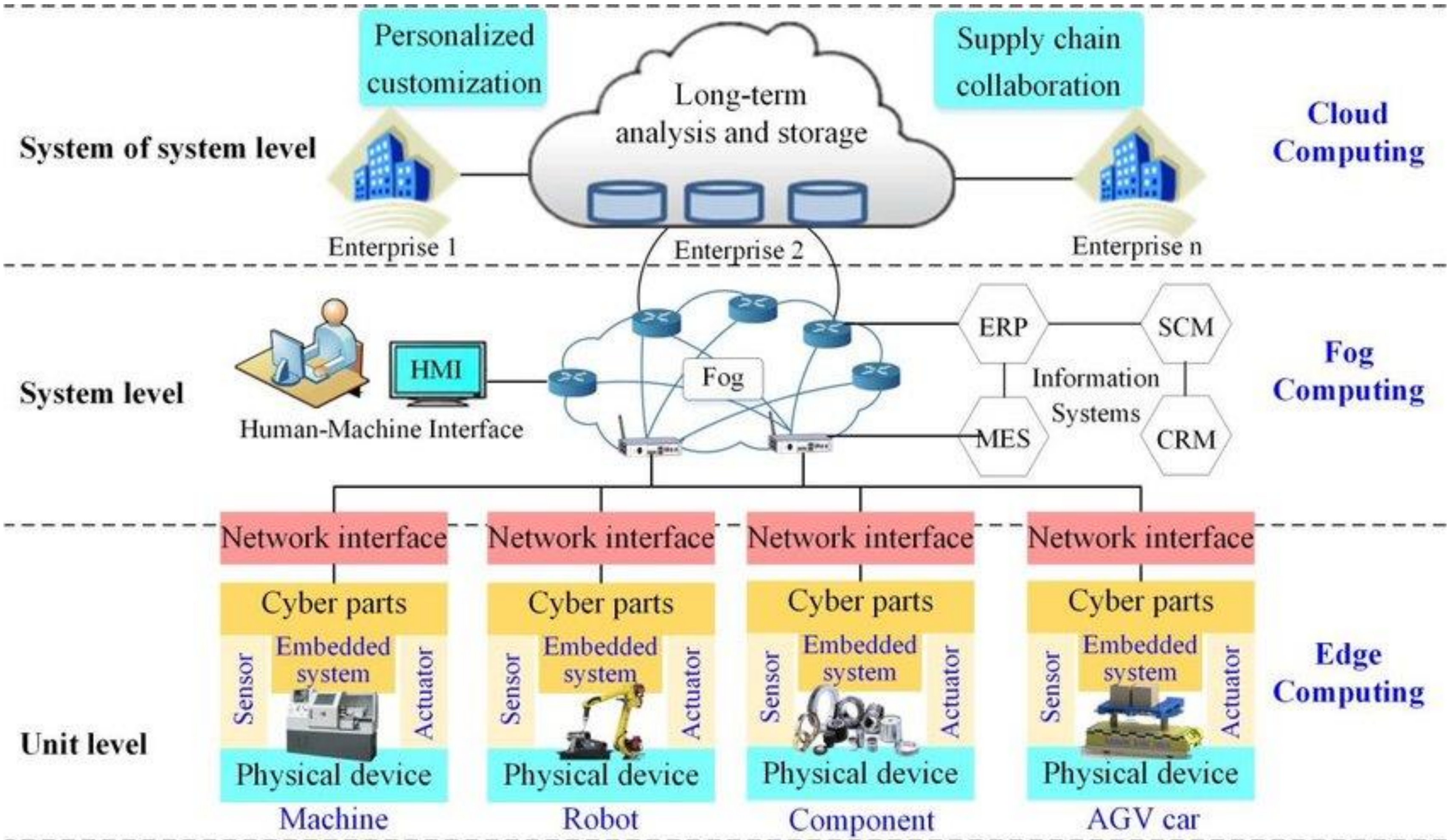


Smartness Incorporation
Handling Process Uncertainties
Predictive maintenance
Learning

Modern Manufacturing - Industry 4.0

Ref: Image Taken from Internet Sources

Generic AI based CPS Architecture

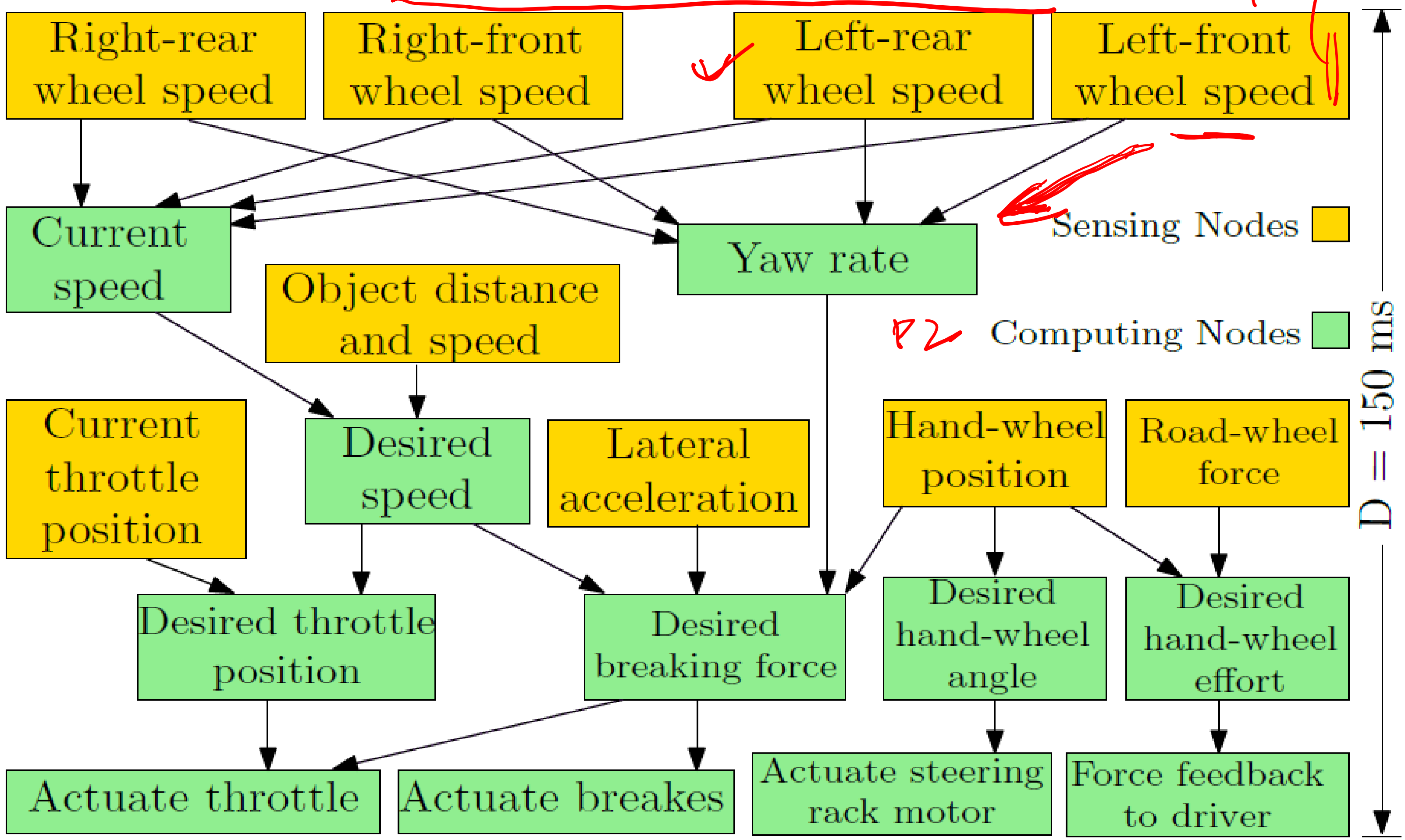


Ref: Image Taken from Internet Sources

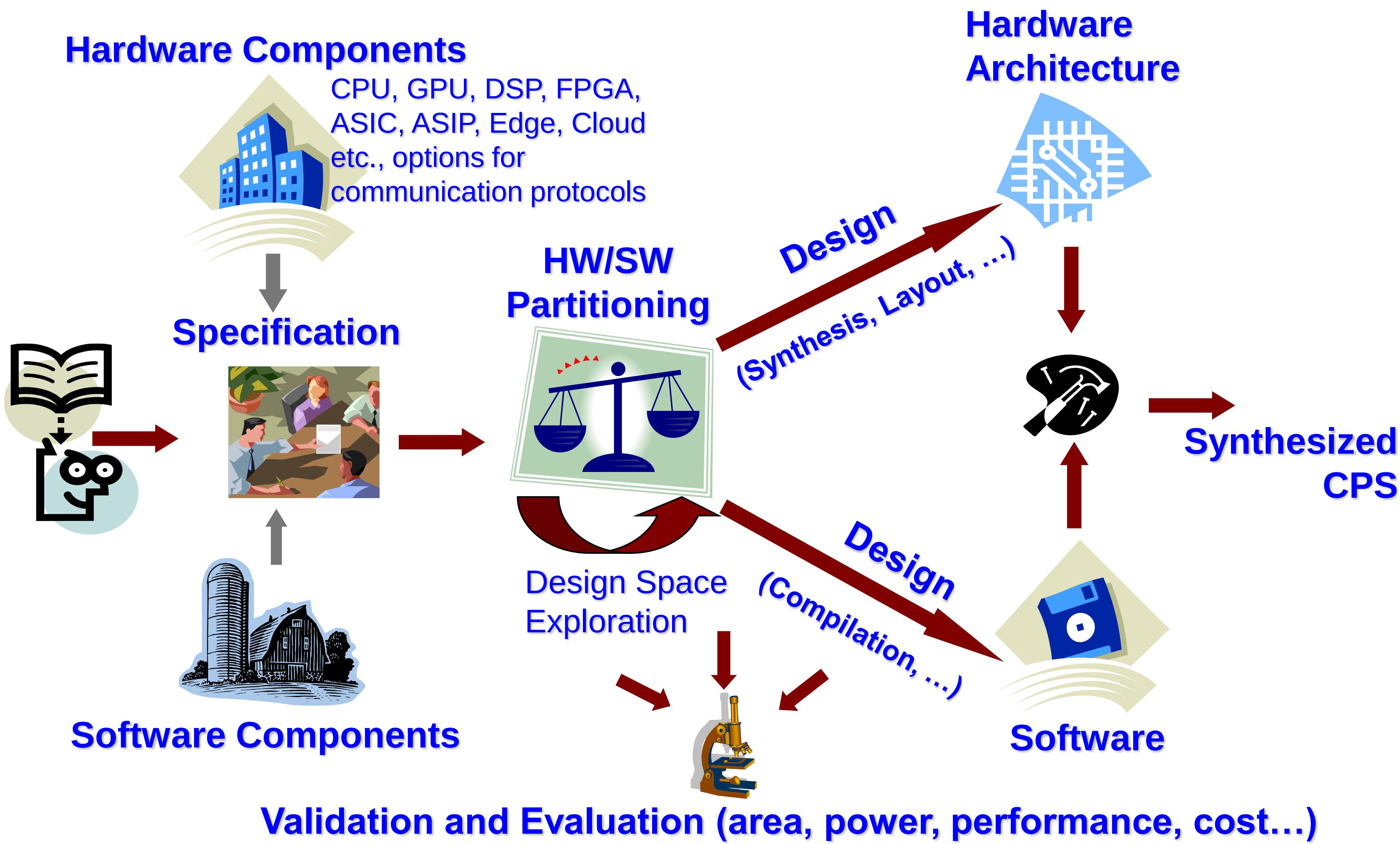
CPS Specification - Schematic

A Simple Adaptive Cruise Control (ACC) System

Codes



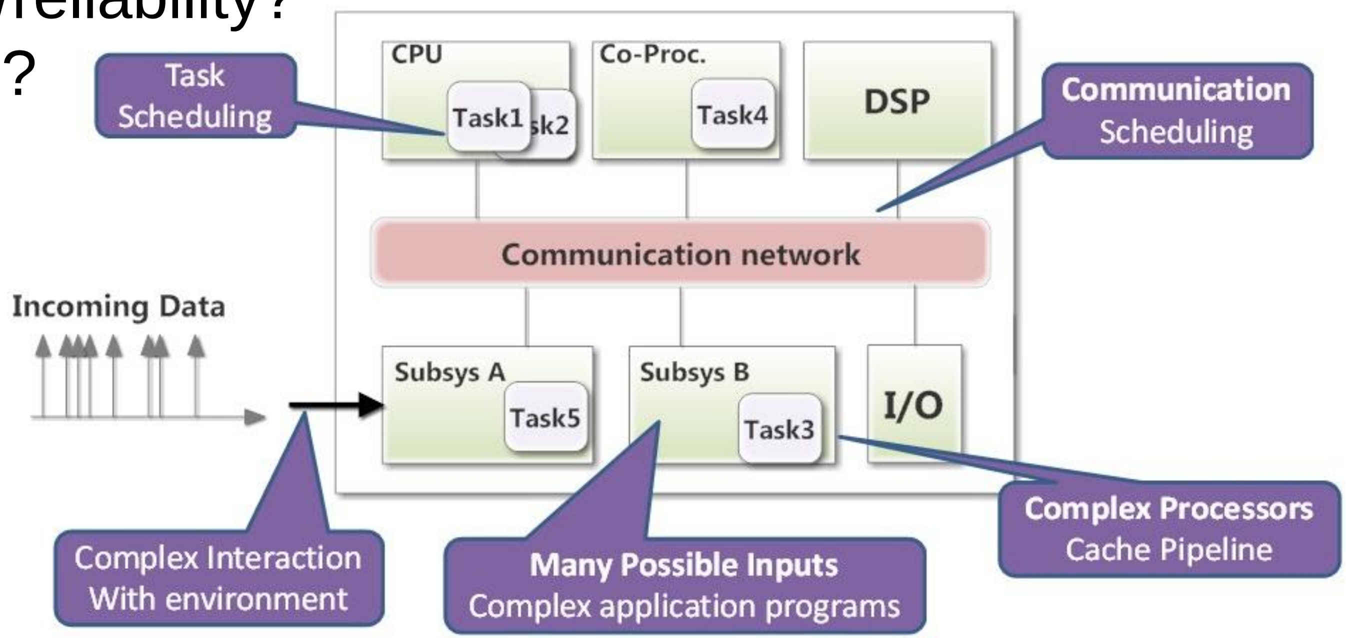
CAD Flow for CPS



Schematic of a Synthesized CPS

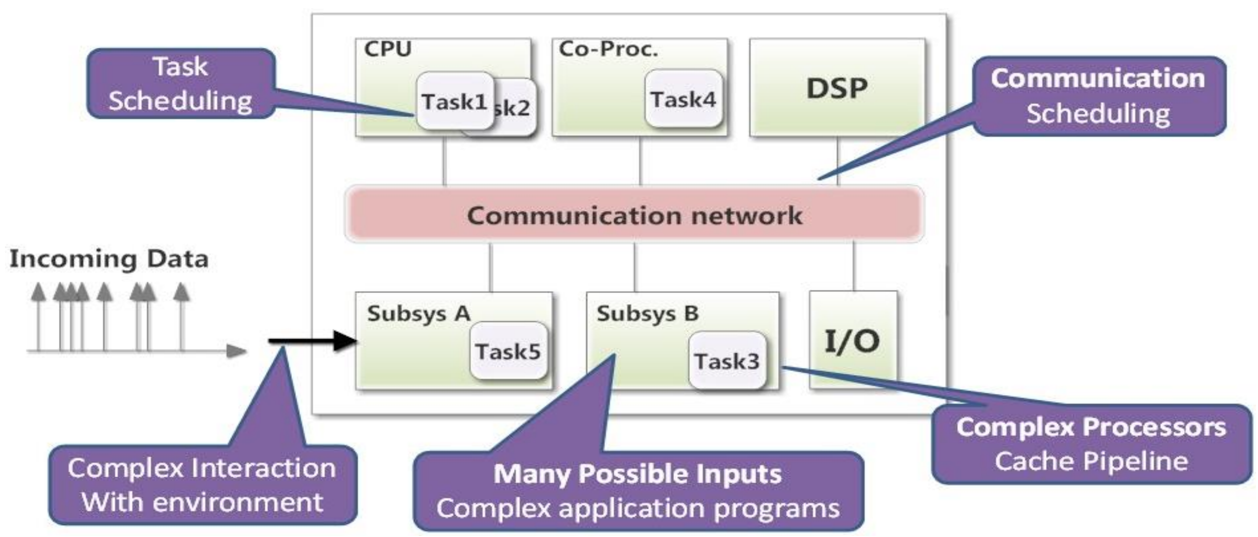
- Validation is important
  - A control system acquires data at the rate on *n* samples per second
    - Can we ensure that each data item will be processed in less than 1/*n* seconds?

- Fault-tolerance/reliability?
- Security Issues?



Reliability Analysis - Issues

- Reliability – Probability of correct system functionality even in the presence of possible component failures:
  - One or more processors / memories / I/O channels, may completely crash or malfunction transiently
    - Transient malfunction – *Due to bit-flips caused by ion impinge*
  - The software tasks may produce incorrect outputs on certain inputs
  - The bus / communication network, may fail
  - The sensors / actuators interacting with the environment, may fail
  - Even the interfaces between say the processor and buses, may fail



Fault Tolerance

- A **fault** is an underlying defect
  - Example: A frozen memory bit, an uninitialized variable in software
- An **error** is the manifestation of a fault as an unexpected behaviour within our system
  - Example: Incorrect result of a computation
  - A fault may (or may not) lead to error
- A **failure** is a situation in which a system (or part of a system) is not performing according to intended specification
  - An error may (or may not) lead to failure

- *A low level failure in a small component of the system can be viewed as a fault at an higher level. This fault can lead to errors, and such errors can trigger failures at the higher level*

Types of Faults

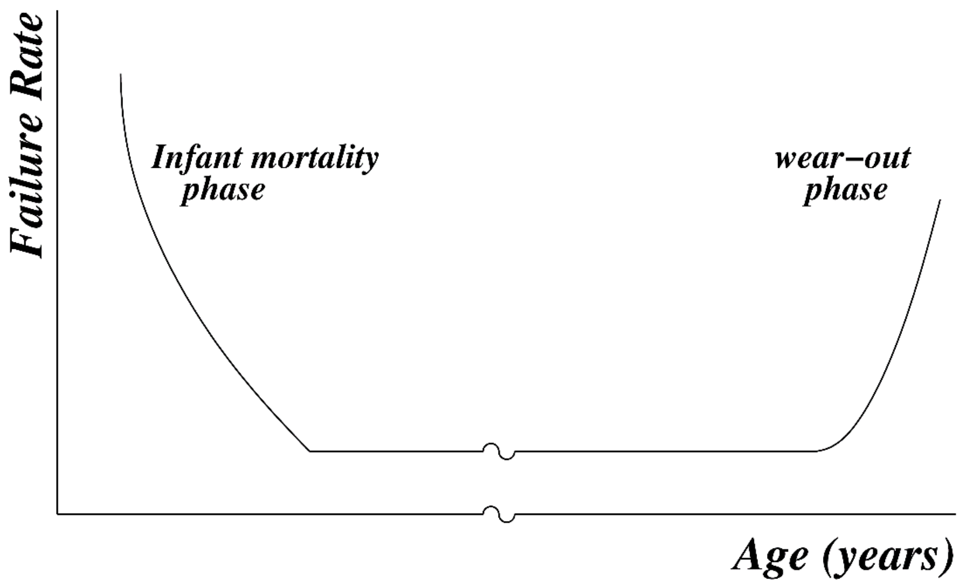
- There are three main types of ‘fault’:
- *Transient Fault* – appears once, then disappears.
- *Intermittent Fault* – occurs, vanishes, reappears; but: follows no real pattern (worst kind).
- *Permanent Fault* – once it occurs, only the replacement/repair of the faulty component will allow the system to function normally.

Reliability and Availability

- Reliability at time t, $R(t)$
  - Conditional probability that the system performs correctly during the period $[0,t]$, given that the system was performing correctly at time 0
- Unreliability, $F(t)$
  - $1 - R(t)$. Often referred to as the *probability of failure*
- Availability at time t, $A(t)$
  - Probability that a system is operating correctly and is available to perform its functions at time t. Unlike reliability, availability is defined at an instant of time
    - The system may incur failures but can be repaired promptly, leading to high availability
    - *A system may have very low reliability, but very high availability*

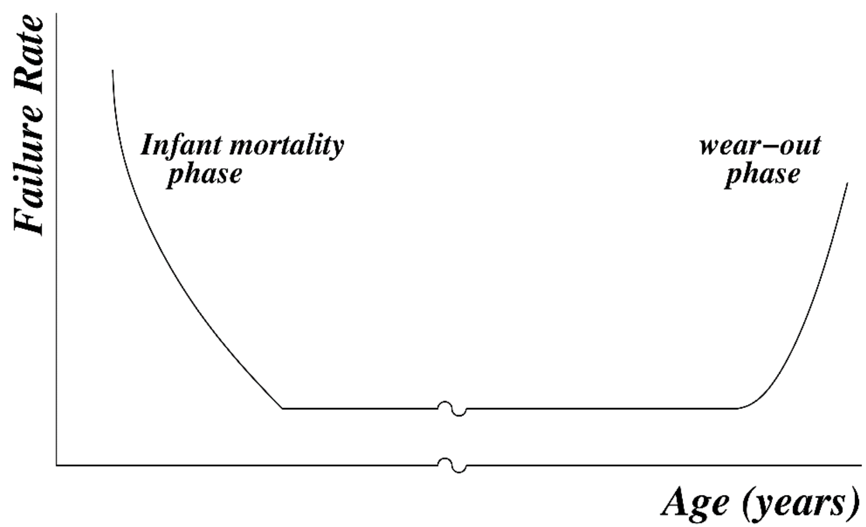
Failure Rate

- Rate at which a component suffers faults
- Depends on age, ambient temperature, voltage or physical shocks that it suffers, and technology
- Dependence on age is usually captured by the bathtub curve:

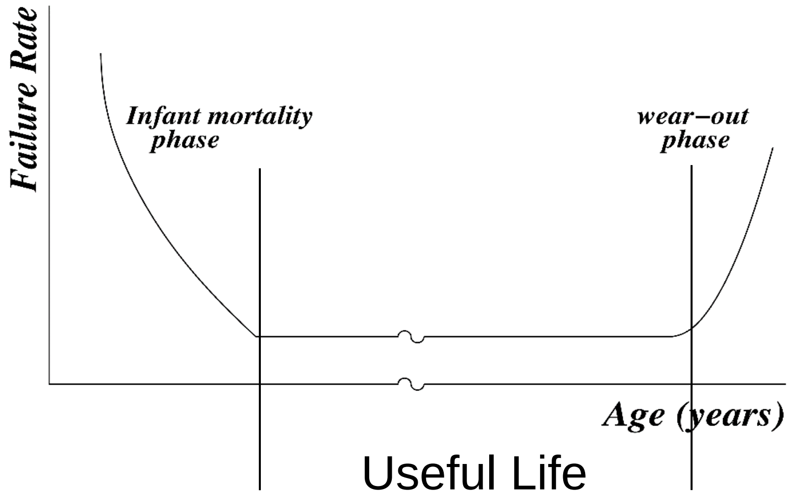


Bathtub Curve

- Young component – high failure rate
  - Good chance that some defective units slipped through manufacturing quality control and were released
- Later - bad units weeded out – remaining units have a fairly constant failure rate
- As component becomes very old, aging effects cause the failure rate to rise again



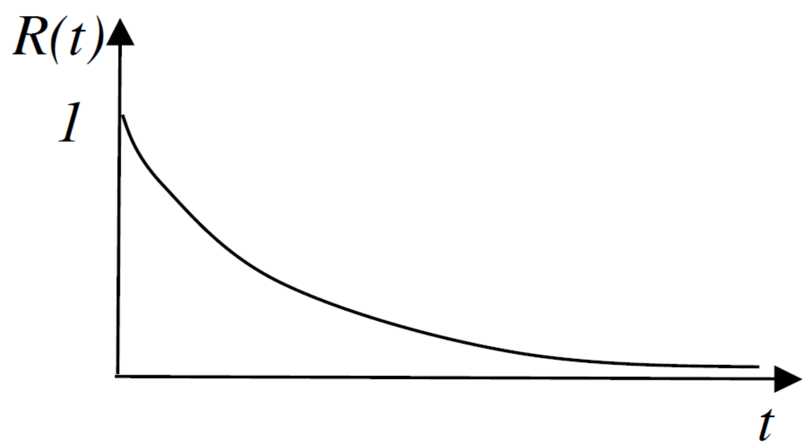
Failure Rate, Reliability and MTTF



- During useful life, components exhibit a constant failure rate λ
- Mean Time To Failure, MTTF = 1/ λ

- Reliability can be modelled using an exponential distribution: $R(t) = e^{-\lambda t}$

- $F(t) = 1 - e^{-\lambda t}$



Redundancy

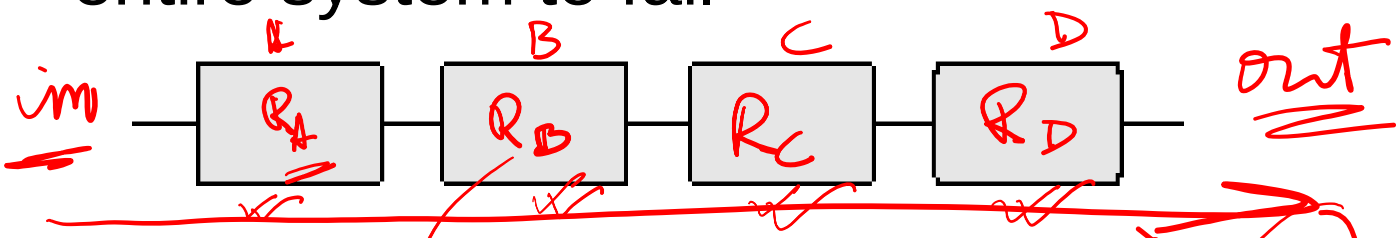
- *Fault-tolerance* is achieved by incorporating more (redundant) resources in the system than is strictly required
  - *Hardware Redundancy:* Based on physical replication of hardware.
  - *Software Redundancy:* The system is provided with different software versions of tasks, preferably written independently by different teams
  - *Time Redundancy:* Based on multiple executions on the same hardware at different times
  - *Information Redundancy:* Based on coding data in such a way that a certain number of bit errors / omissions can be detected and/or corrected
  - *Hybrid Redundancy:* A mixture of two or more of the above strategies

Canonical Structures

- A canonical structure is constructed out of N individual modules
- The basic canonical structures are
  - A series system
  - A parallel system
  - A mixed system
- *We will assume statistical independence between failures in the individual modules*

Reliability of a Series System

- A series system - set of modules so that the failure of any one module causes the entire system to fail

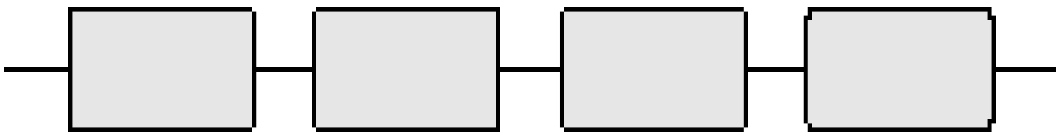


$R_{sys} = R_A \times R_B \times R_C \times R_D$

Property of independence

Reliability of a Series System

- A series system - set of modules so that the failure of any one module causes the entire system to fail



- Reliability of a series system - $R_s(t)$ - product of reliabilities of its N modules

$$R_s(t) = \prod_{i=1}^{N} R_i(t)$$

- $R_i(t)$ is the reliability of module i

Reliability of a Series System

- Every module *i* has a constant failure rate λ_i

$$R_i(t) = e^{-\lambda_i t}$$

$$R_s(t) = e^{-\lambda_s t} = e^{-\Sigma \lambda_i t}$$

- $\lambda_s = \Sigma \lambda_i$ is the constant failure rate of the series system
- Mean Time To Failure of a series system -

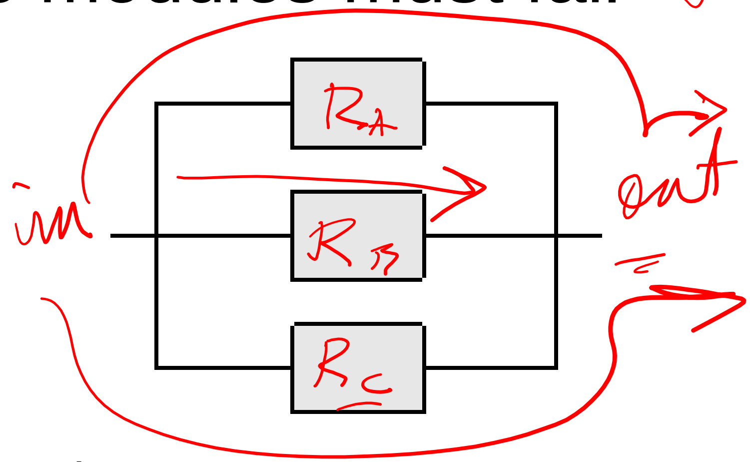
$$MTTF_s = \frac{1}{\lambda_s} = \frac{1}{\Sigma \lambda_i}$$

Reliability of a Parallel System

- A Parallel System - a set of modules connected so that all the modules must fail before the system fails

$R_A \cup R_B \cup R_C$

Demorgans law



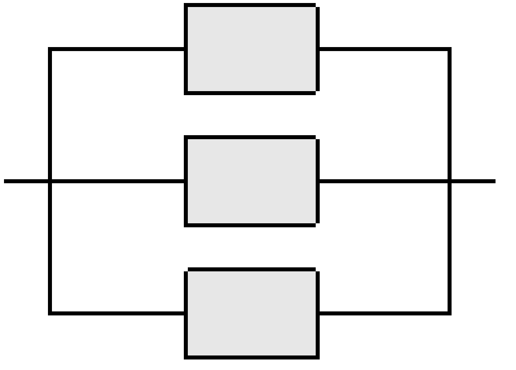
- Reliability of a parallel system - $R_p(t)$

$$R_A \cup R_B \cup R_C = 1 - \boxed{F_A \cap F_B \cap F_C}$$

$$\left(1 - (1-R_A)(1-R_B)(1-R_C)\right)$$

Reliability of a Parallel System

- A Parallel System - a set of modules connected so that all the modules must fail before the system fails



- Reliability of a parallel system - $R_p(t)$

$$R_p(t) = 1 - \prod_{i=1}^{N} [1 - R_i(t)]$$

- $R_i(t)$ is the reliability of module *i*

Low-level Vs. High-level Redundancy

$\text{in} \rightarrow \boxed{A}\cdot\boxed{B} \rightarrow \text{out}$

$R_{sys} = 0.987$

- Low-level – Component level redundancy

$R_A = 0.9 \qquad R_B = 0.95$

$$\left[1-(1-0.9)^2\right] \times \left[1-(1-0.95)^2\right]$$

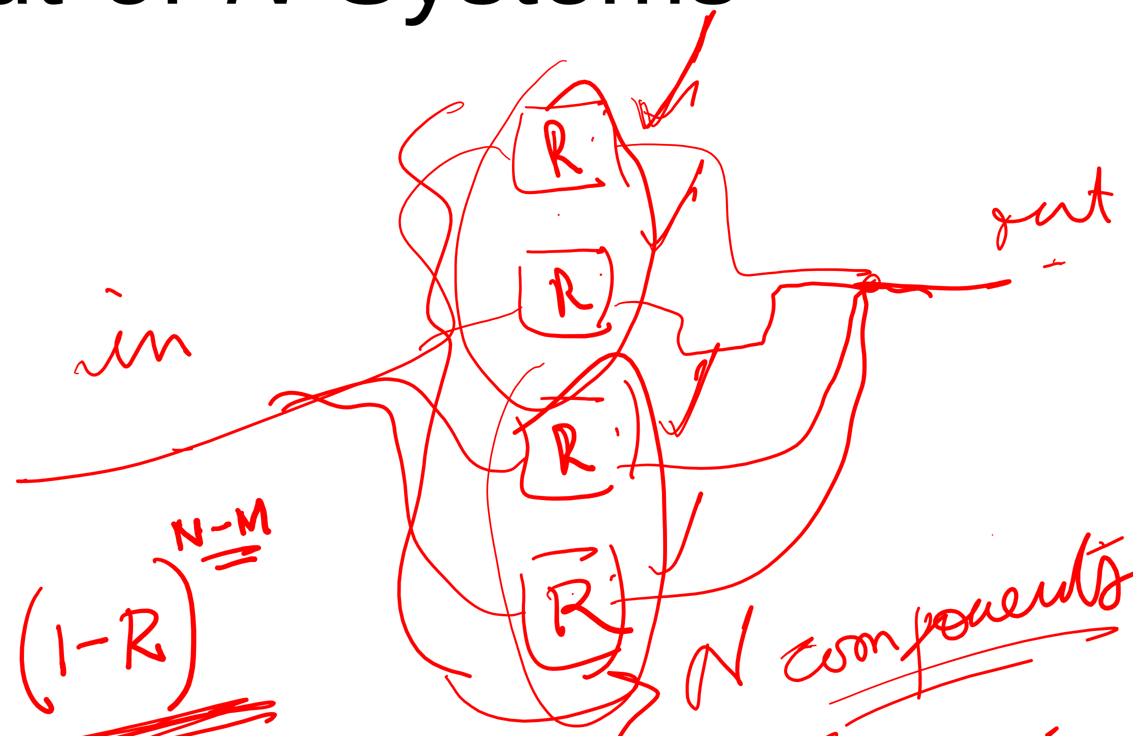
0.99

$\Rightarrow 0.9875$

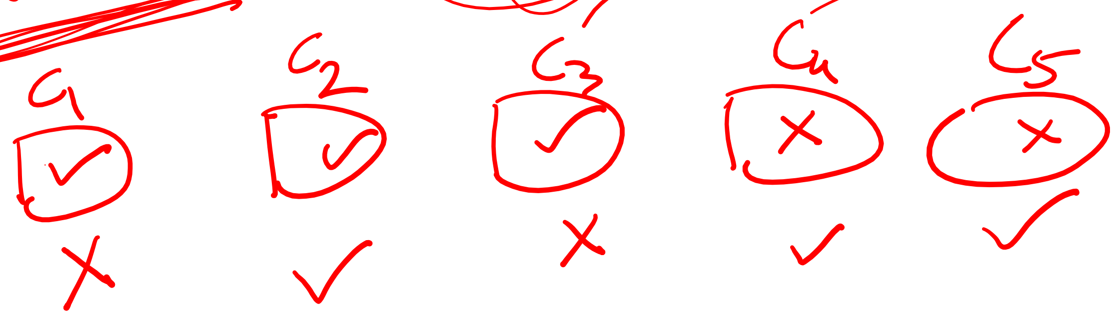
- High-level – System-level redundancy

$$\left[1-(1-R_A R_B)^2\right] \rightarrow R_{sys} = 1-(1-\underbrace{0.9\times 0.95}_{F_{AB}})^2 = 0.73$$

Exactly M-out-of-*N* Systems



$$\binom{N}{M} R^{M} \times (1-R)^{N-M}$$



At Least M-out-of-*N* Systems

$$\sum_{i=M}^{N} \binom{N}{i} R^{i} (1-R)^{N-i}$$

0.9 ✓

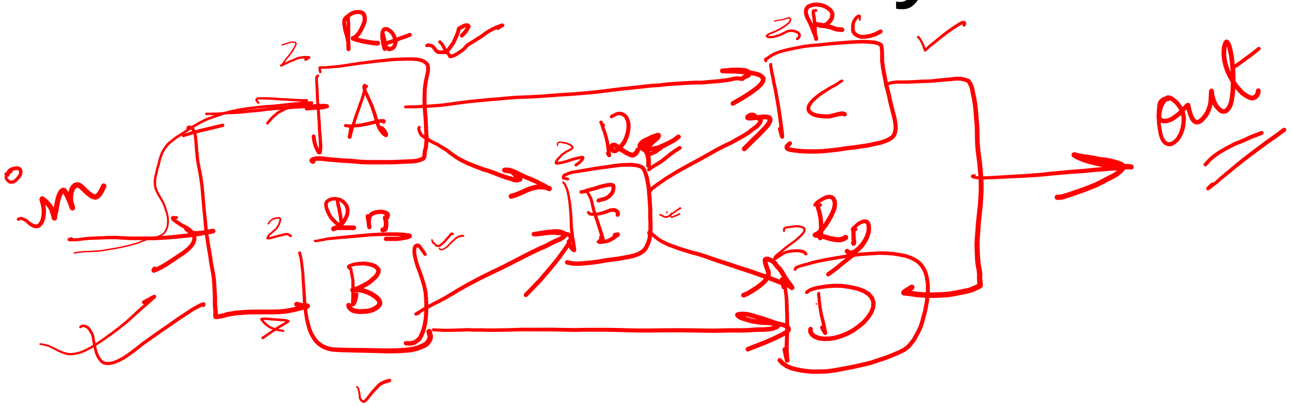
0.9 ✓

0.9

$${}^{3}C_{2} \cdot R^{2} (1-R) + {}^{3}C_{3} R^{3}$$

$$R_{sys} = \boxed{3 \times (0.9)^{2} \times (0.1) + 0.9^{3}}$$

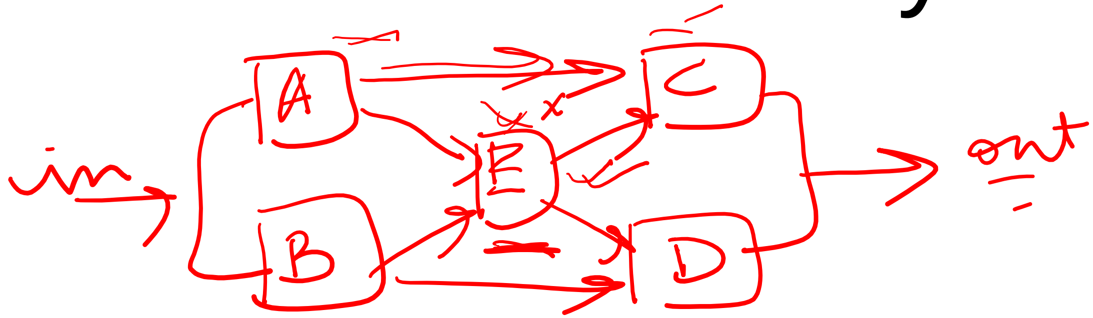
Non Series / Parallel Systems

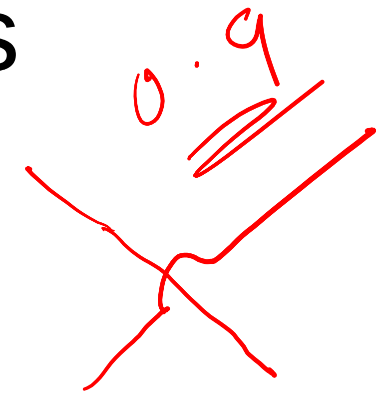


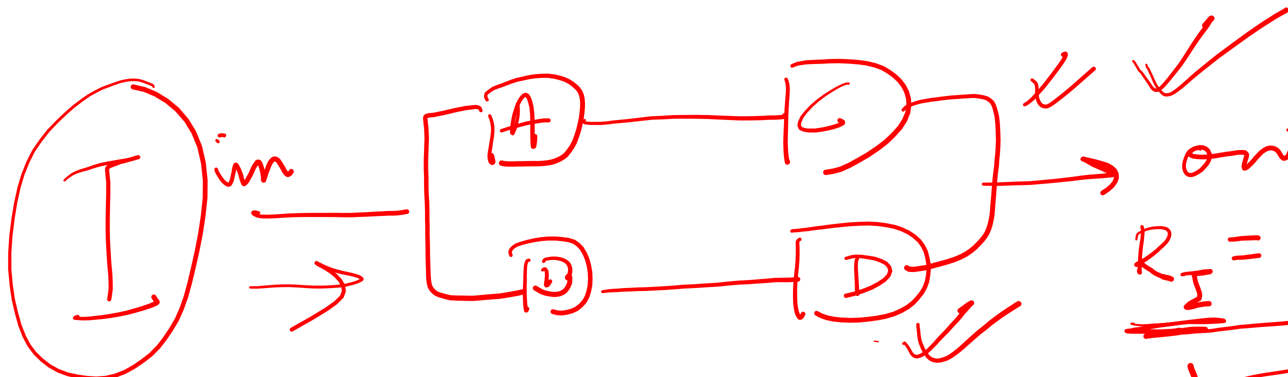
A	B	C	D	E	
✓	✓	✓	✓	✓	$= R \ldots$
✓	✗	✗	✓	✓	$=$
✗	✗	✓	✓	✓	$= 0 = 0$

R_{sys}

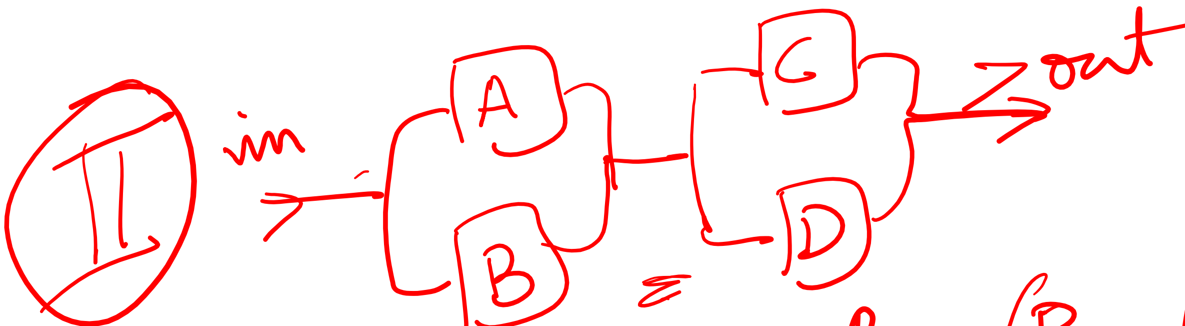
Non Series / Parallel Systems







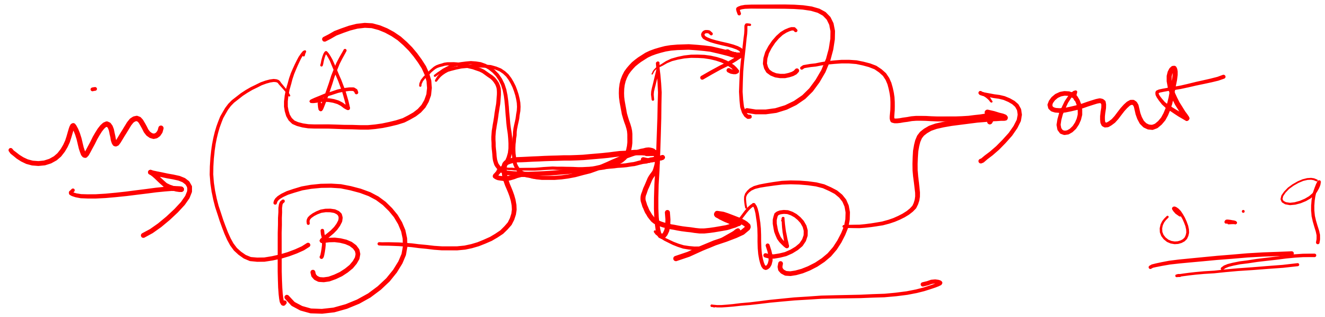
$$R_I = 1 - (1 - R_A R_C)(1 - R_B R_D)$$



$$\boxed{R_{sys} = (1 - R_E) \times \boxed{R_I} + R_E \times \boxed{R_{II}}}$$

$$R_{sys} = (R_{sys} \mid E \text{ fails}) + (R_{sys} \mid E \text{ works})$$

Non Series / Parallel Systems

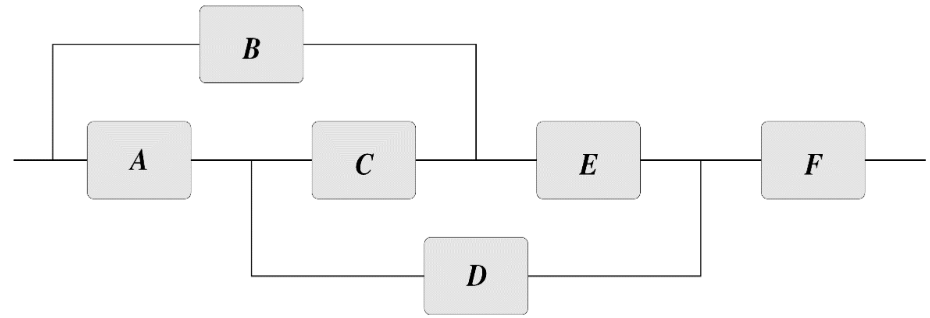


$$R_{\pi} = \left[1-(1-R_A)(1-R_B)\right]\left[1-(1-R_C)(1-R_D)\right]$$

$$= \left[1-(0.1)^2\right]\left[1-(0.1)^2\right]$$

$$= (0.99)^2 = 0.9801$$

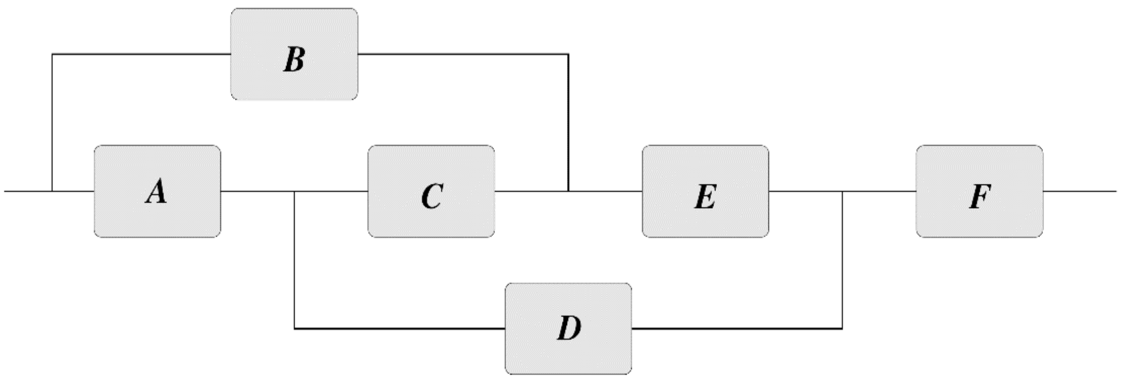
Non Series / Parallel Systems



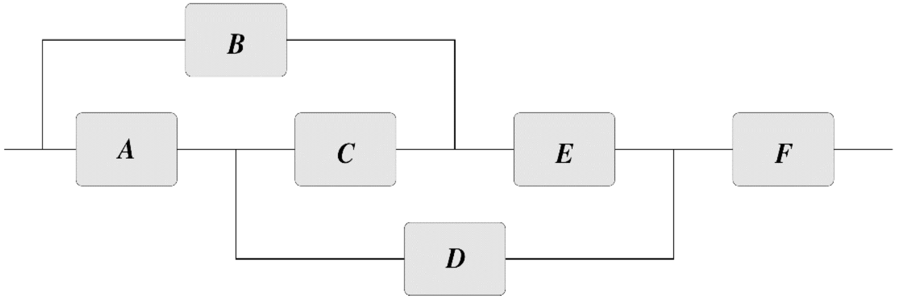
- Each path represents a configuration allowing the system to operate successfully, e.g., ADF
- The reliability can be calculated by expanding about a single module *i* :
- $R_s = R_i$ *Prob{System works | i is fault-free}*

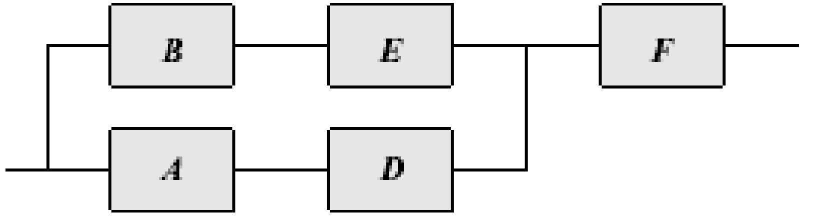
  $+(1-R_i)$ *Prob{System works | i is faulty}*
- Draw two new diagrams: in (a) module *i* is operational; and (b) module *i* is faulty
- Module *i* is selected so that the two new diagrams are closer to simple series/parallel structures

Expanding about C



Expanding about C

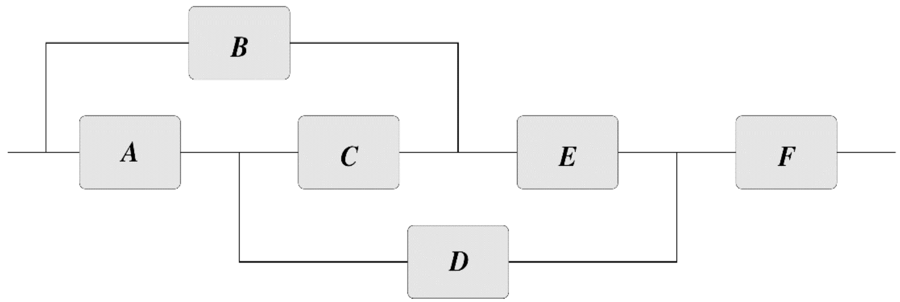


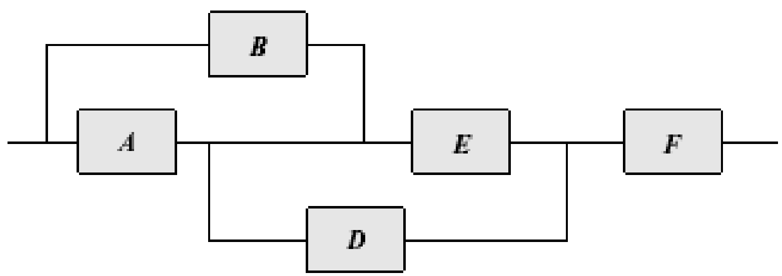


When C is faulty

Effective Reliability ?

Expanding about C

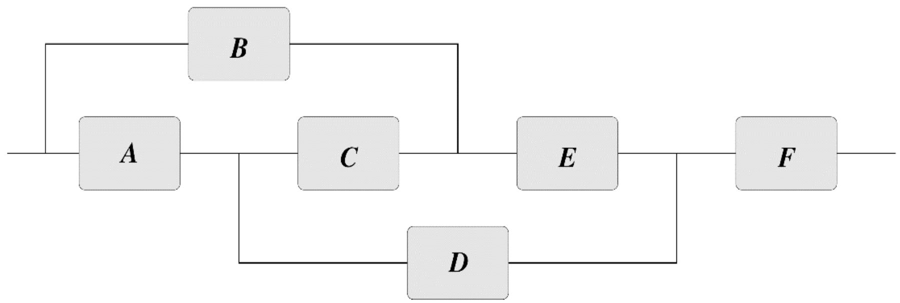


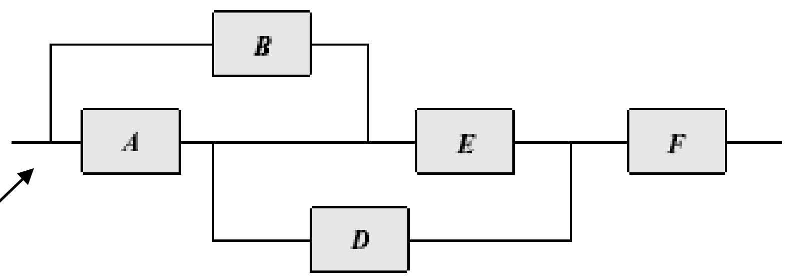


When C is operational

Can you directly find the effective reliability for this case?

Expanding about C

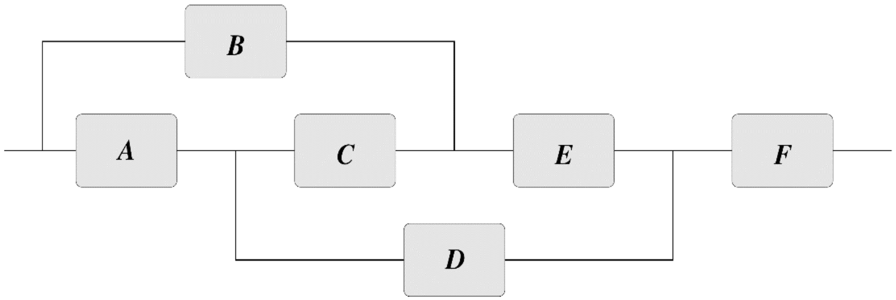


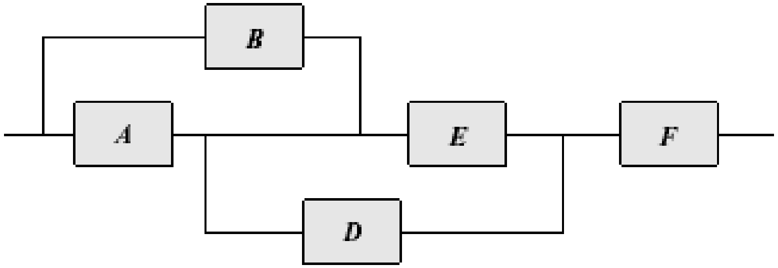


When C is operational

Could this system be looked upon as a simpler structure?

Expanding about C

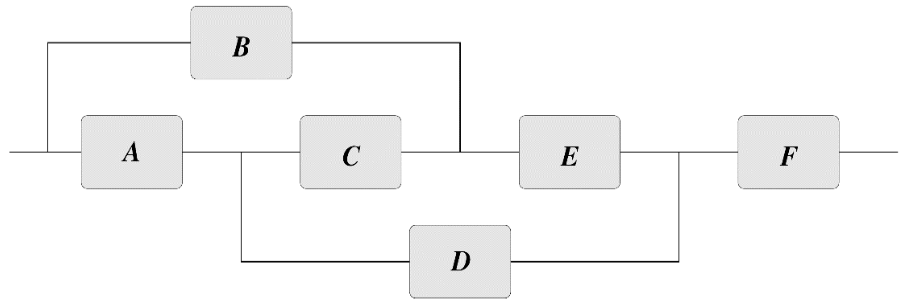


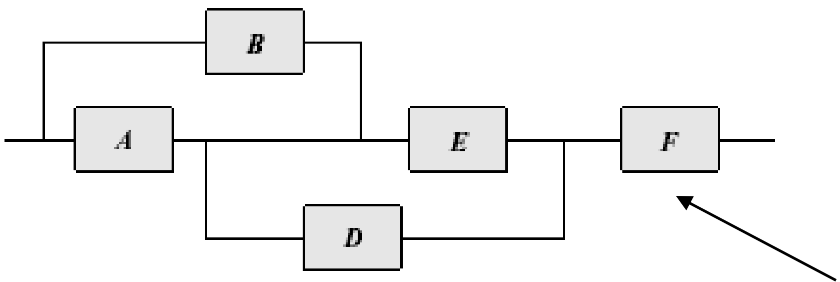


When C is operational

This figure should not be viewed as a parallel connection of A and B, connected serially to D and E in parallel. Such a diagram will have the path BCDF which is not a valid path

Expanding about C



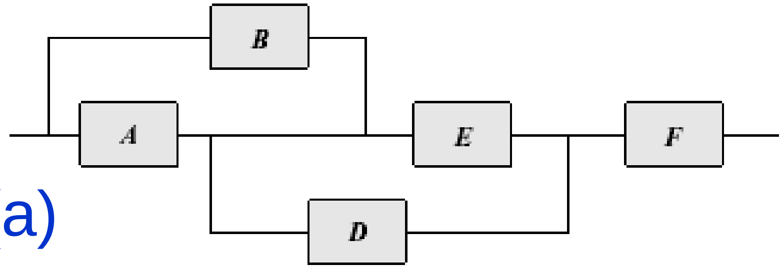


The above figure needs further expansion about E

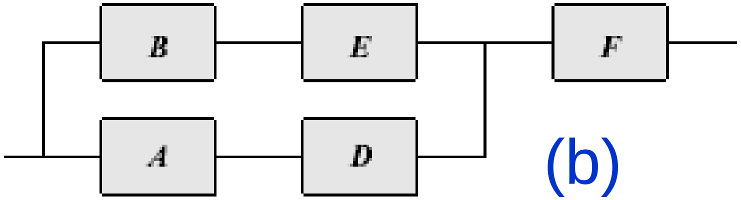
Draw the derived figures when the above figure is expanded about E

- The process of expanding can be repeated until the resulting diagrams are of the series/parallel type

Expanding about C and E



(a)



(b)

- $R_s = R_C$ Prob {System works | C is operational}
  $+(1-R_C)\ R_F\ [1-(1-R_A\ R_D)(1-R_B\ R_E)]$

- Expanding about E yields
- Prob {System works | C is operational} =
  $R_E\ R_F\ [1-(1-R_A)(1-R_B)] +(1-R_E)R_A\ R_D\ R_F$
- Substituting results in
- $R_s = R_C\ [R_E\ R_F(R_A+R_B-R_A\ R_B)+(1-R_E)\ R_A\ R_D\ R_F]\ +$
  $(1-R_C)\ [R_F(R_A\ R_D+R_B\ R_E-R_A\ R_D\ R_B\ R_E)]$

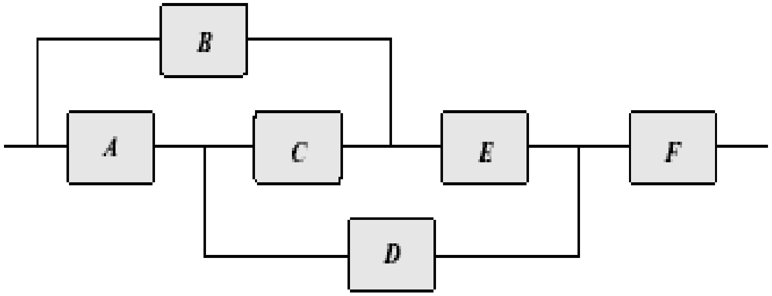
Upper Bound on Reliability

- If structure is too complicated - derive upper and lower bounds on R_{system}

- An upper bound - $R_{system} \leq 1 - \prod (1-R_{path\_i})$
  - R_{path_i} - reliability of modules in series along path i
  - Assuming all paths are in parallel

Determine the paths in our example system:

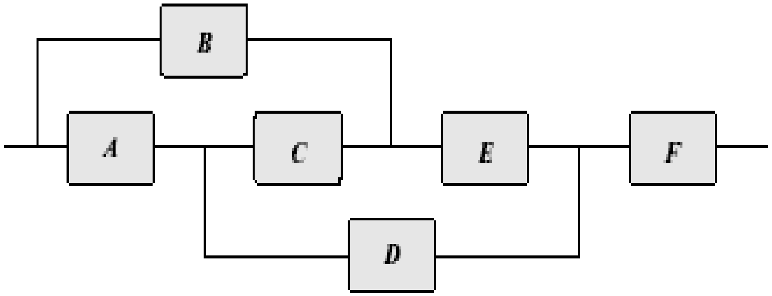
What is the value of R_{system}?



Upper Bound on Reliability

- If structure is too complicated - derive upper and lower bounds on R_{system}
- An upper bound - $R_{system} \le 1 - \prod (1-R_{path\_i})$
  - R_{path_i} - reliability of modules in series along path i
  - Assuming all paths are in parallel
- Example - the paths are ADF, BEF and ACEF
- $R_S \le 1 - (1-R_A R_D R_F)(1-R_B R_E R_F)(1-R_A R_C R_E R_F)$
- If $R_A=R_B=R_C=R_D=R_E=R_F=R$

$$R_s \le R^3(R^7 - 2R^4 - R^3 + R + 2)$$

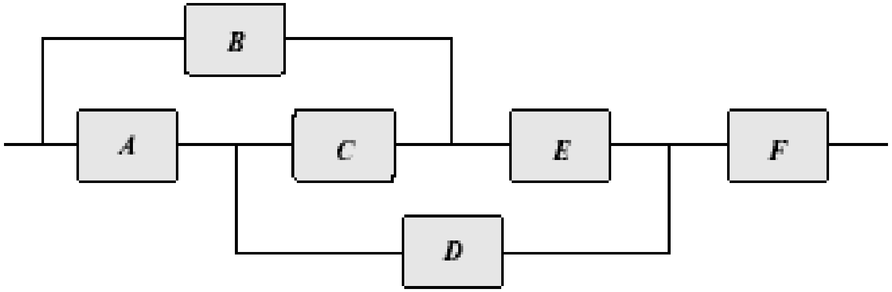


Lower Bound on Reliability

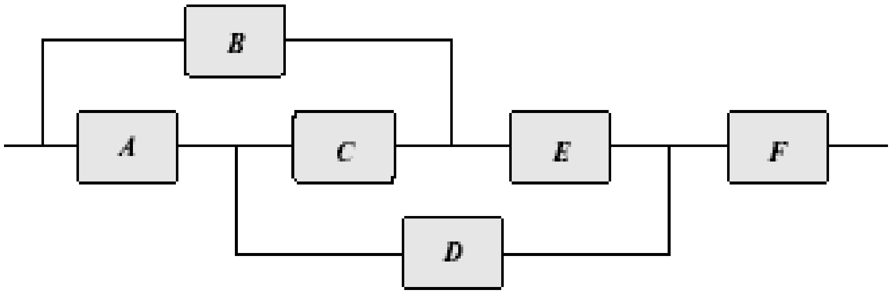
- Lower bound calculated based on minimal cut sets
- A minimal cut set:
  - Minimal list of modules such that the removal of all modules will cause a working system to fail
- The lower bound is: $R_{system} \geq \prod (1-Q_{cut\_i})$
  - Q_{cut_i} - probability that the minimal cut i is faulty (i.e., all its modules are faulty)

Determine the minimal cut sets in our example system:

What is the value of R_{system}?



Lower Bound on Reliability

- Lower bound calculated based on minimal cut sets
- A minimal cut set:
  - Minimal list of modules such that the removal of all modules will cause a working system to fail
- Minimal cut sets: F, AB, AE, DE and BCD
- 

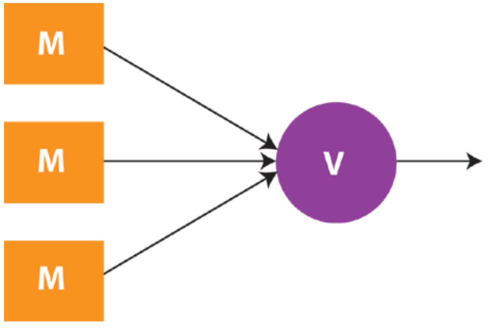
- $R_S \geq R_F \times [1-(1-R_A)(1-R_B)] \times [1-(1-R_A)(1-R_E)] \times [1-(1-R_D)(1-R_E)] \times [1-(1-R_B)(1-R_C)(1-R_D)]$

M-of-*N* Systems

- An *N* module system which needs at least *M* of them for proper functioning
  - $$R_{M\_of\_N}(t) = \sum_{i=M}^{N} \binom{N}{i} R^{i}(t)[1 - R(t)]^{N-i}$$

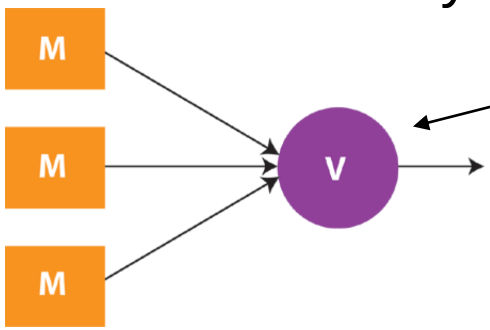
M-of-*N* Systems

- An *N* module system which needs at least *M* of them for proper functioning
  - $$R_{M\_of\_N}(t) = \sum_{i=M}^{N} \binom{N}{i} R^i(t)[1 - R(t)]^{N-i}$$
  - Assumption: Failures are independent
    - Eg. What happens if the entire system fails due to a common point of failure?
    - What is reliability of the *2-of-3* system shown below?



M-of-*N* Systems

- An *N* module system which needs at least *M* of them for proper functioning
  - $$R_{M\_of\_N}(t) = \sum_{i=M}^{N} \binom{N}{i} R^{i}(t)[1-R(t)]^{N-i}$$
  - Assumption: Failures are independent
    - Eg. What happens if the entire system fails due to a common point of failure?
    - What is reliability of the *2-of-3* system shown below?



- A Triple Modular Redundancy (TMR)
- In general: *N*-Modular Redundancy (NMR)
  - *M-of-N* cluster with *N* odd and *M = (N+1)/2*

M-of-N Systems

- An N module system which needs at least M of them for proper functioning
  - $$R_{M\_of\_N}(t) = \sum_{i=M}^{N} \binom{N}{i} R^i(t)[1 - R(t)]^{N-i}$$
  - Assumption: Failures are independent
    - What happens when:
      - The entire system fails due to a common point of failure?
        - The correlated failure factor R_{voter} can dominate the overall failure probability

M-of-*N* Systems

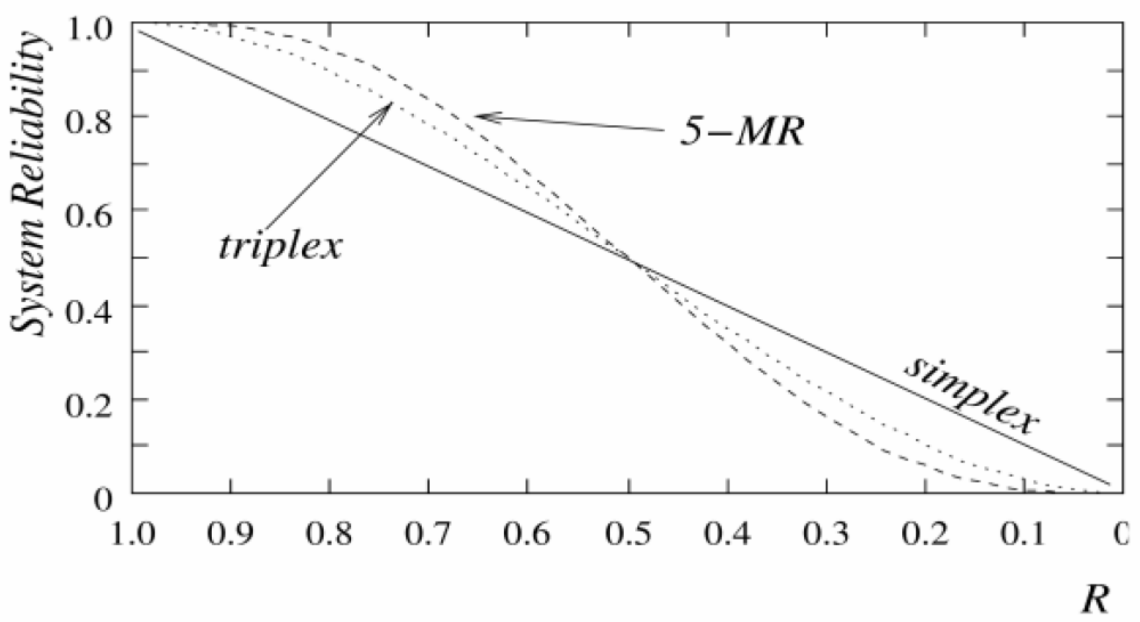
- An *N* module system which needs at least *M* of them for proper functioning
  - $$R_{M\_of\_N}(t) = \sum_{i=M}^{N} \binom{N}{i} R^{i}(t)[1 - R(t)]^{N-i}$$
  - Assumption: Failures are independent
    - What happens when:
      - The entire system fails due to a common point of failure?
        - The correlated failure factor R_{voter} can dominate the overall failure probability
      - Not all but certain subsets of the *N* modules can suffer correlated failures?

M-of-*N* Systems

- An *N* module system which needs at least *M* of them for proper functioning
  - $$R_{M\_of\_N}(t) = \sum_{i=M}^{N} \binom{N}{i} R^{i}(t)[1 - R(t)]^{N-i}$$
  - Assumption: Failures are independent
    - What happens when:
      - Individual modules have very poor reliabilities
      - What is R_s of a TMR structure when all modules have a reliability of R=0.25?

M-of-*N* Systems

- What happens when individual modules have very poor reliabilities



Comparison of NMR reliability (N=3 and 5) to that of a single module (Voter failure rate considered negligible)

- Below R=0.5: Redundancy becomes a disadvantage
- Usually R >> 0.5: Triplex offers significant reliability gains

Voters

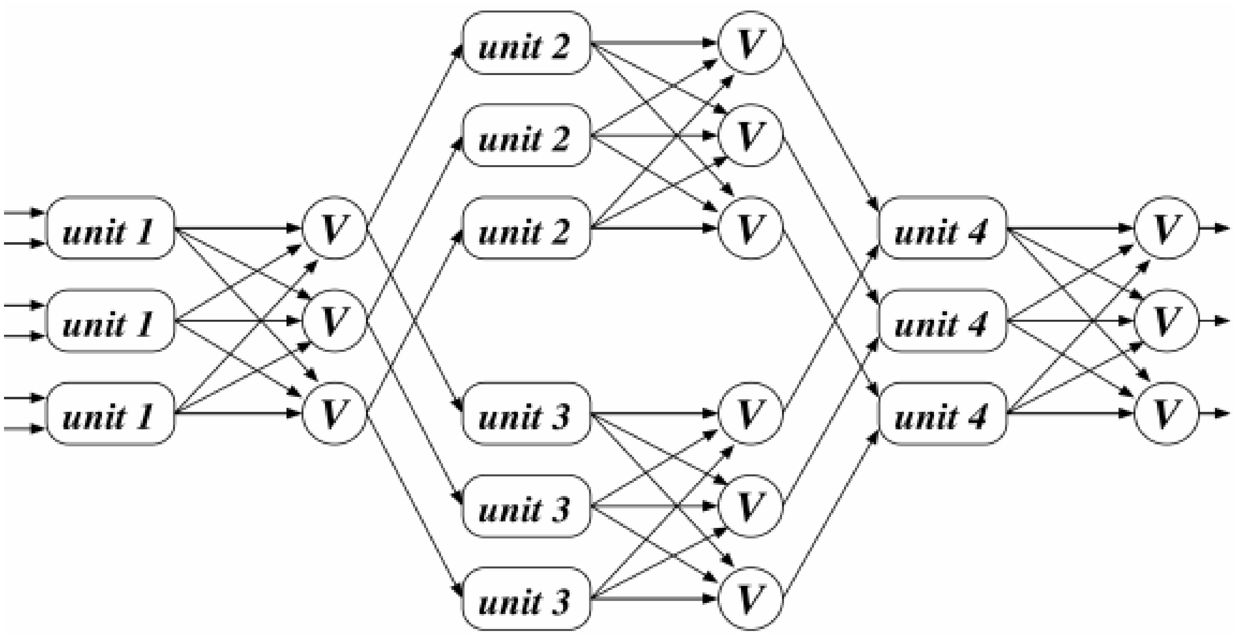
- A voter receives inputs X_1, X_2,...,X_N from an *M-of-N* cluster and generates a representative output
- Simplest voter: *bit-by-bit* comparison of the outputs producing the majority vote
- This only works when all functional processors generate outputs that match bit by bit
  - Processors must be identical and use the same software
  - Otherwise - two correct outputs can diverge slightly, in the lower significant bits

Plurality Voters

- We declare two outputs X and Y as practically identical if $|x-y| < \delta$ for some specified δ
- A *k*-plurality voter looks for a set of at least *k* practically identical outputs, and picks any of them (or their median) as the representative

- Example: $\delta = 0.1$, five outputs
  - 1.10, 1.11, 1.32, 1.49, 3.00
  - The subset {1.10, 1.11} would be selected by a 2-plurality voter

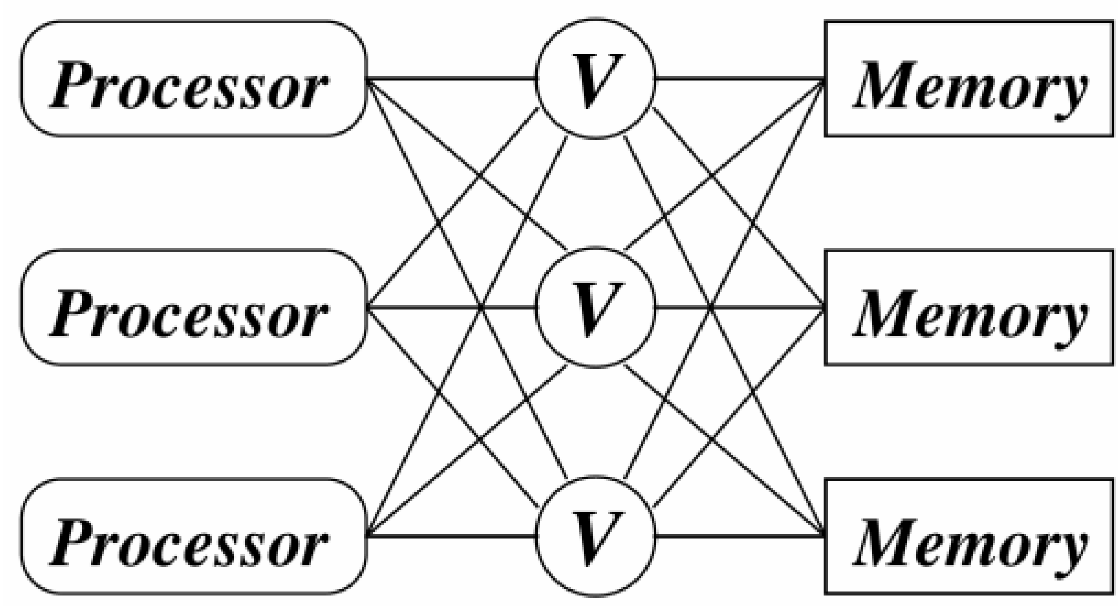
Unit-level Modular Redundancy



- Voters are no longer as critical as in NMR; a single faulty voter will be no worse than a single faulty unit
  - Effect of a fault will not propagate beyond the next level of units

- The level at which the replication and voting are applied can be further lowered at the expense of additional voters increasing the size and delay of the system

Triplicated Processor/Memory System



- All communications (in either direction) between the triplicated processors and triplicated memories go through majority voting
- This organization has a higher reliability than a single majority voting of triplicated processor/ memory structure