COE 444 – Internetwork Design & Management

Dr. Marwan Abu-Amara
Computer Engineering Department
King Fahd University of Petroleum and Minerals

Basic Forms of Hardware Redundancy

- Masking redundancy
  - relies on voting to mask the occurrence of errors
  - can operate without need for error detection or system reconfiguration
  - triple modular redundancy (TMR)
  - N-modular redundancy (NMR)
- Standby redundancy
  - achieves fault tolerance by error detection, error location, and error recovery
  - standby sparing
    - one module operational; one or more modules serve as standbys or spares
- Hybrid redundancy
  - Fault masking used to prevent system from producing erroneous results
  - fault detection, location, and recovery used to reconfigure system in event of an error.
  - N-modular redundancy with spares.

Evaluation

- Allows comparison of design techniques and subsequent tradeoffs
- Mathematical Models: vital means for system reliability and availability predictions
  - Combinatorial: series/parallel, M-of-N, non-series/nonparallel
  - Markov: time invariant, discrete time, continuous time, hybrid
  - Reward Models
  - Queuing
- Probabilistic/Stochastic models of systems created and used to evaluate reliability and/or availability, Performability

Combinatorial Modeling

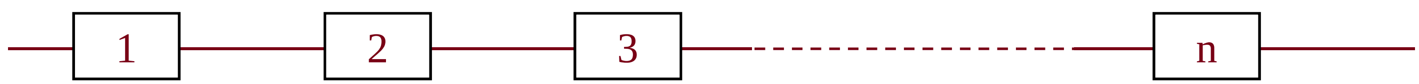
- System is divided into non-overlapping modules
- Each module is assigned either a probability of working, P_i, or a probability as function of time, $R_i(t)$….(Reliability = 1- (area under the failure density curve)
- The goal is to derive the probability, P_{sys}, or function $R_{sys}(t)$: Prob that the system survives until time t
- Assumptions:
  - module failures are independent
  - once a module has failed, it is always assumed to yield incorrect results
  - System considered failed if it does not contain a minimal set of functioning modules
  - once system enters a failed state, other failures cannot return system to functional state
- Models typically enumerate all the states of the system that meet or exceed the requirements for a correctly functioning system
- Combinatorial counting techniques are used to simplify this process

Series Systems

- Assume system has n components, e.g. CPU, memory, disk, terminal
- All components should survive for the system to operate correctly
- Reliability of the system

$$R_{series}(t) = \prod_{i=1}^{n} R_i(t)$$

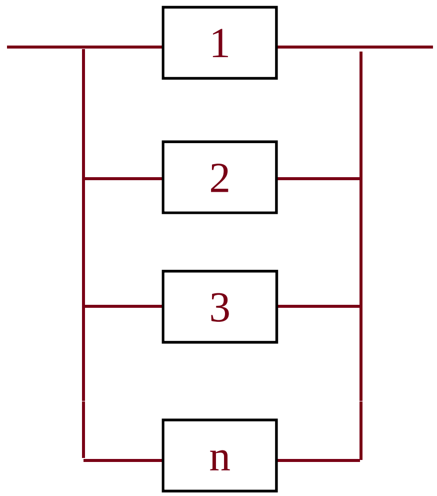
where $R_i(t)$ is the reliability of module i



$$R_{sys} = R_1 \bullet R_2 \bullet R_3 \bullet \ldots \bullet R_n$$

Parallel Systems

- Assume system with spares
- As soon as fault occurs a faulty component is replaced by a spare
- Only one component needs to survive for the system to operate correctly
- Prob. module *i* to survive = R_i
- Prob. module *i* does not survive = $(1 – R_i)$
- Prob. no modules survive = $(1 – R_1)(1 – R_2) \ldots (1 – R_n)$



Prob [at least one module survives] = 1 – Prob [no module survives]

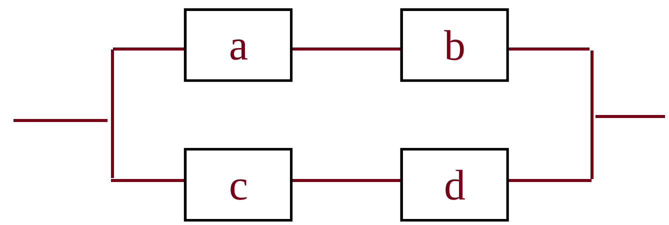
$$R_{parallel}(t) = 1.0 - \prod_{i=1}^{n} \left(1.0 - R_i(t)\right)$$

- Reliability of the parallel system

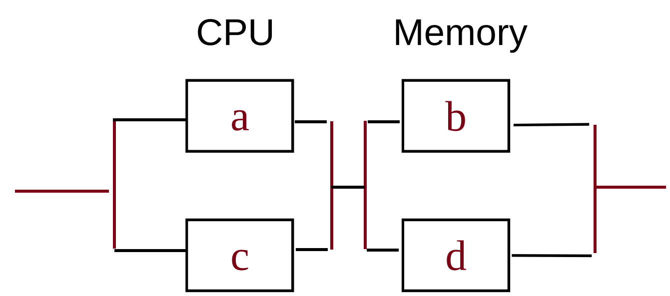
Series-Parallel Systems

- Consider combinations of series and parallel systems
- Example, two CPUs connected to two memories in different ways

$$R_{sys} = 1 - (1 - R_a R_b)(1 - R_c R_d)$$

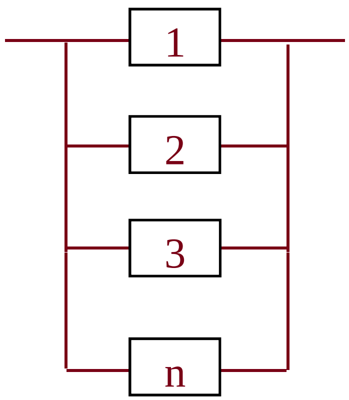


$$R_{sys} = (1 - (1 - R_a)(1 - R_c))(1 - (1 - R_b)(1 - R_d))$$



A Simple Example

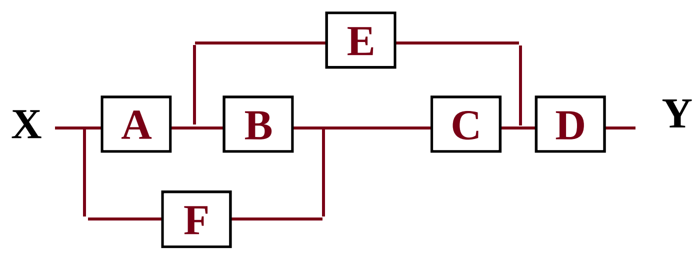
- Consider dynamic redundant system with spares (dynamic redundancy)
- As soon as fault occurs, a faulty component is replaced by a spare
- Up to n-1 spare modules
- $R_{sys} = 1 – (1 – R_1) (1 – R_2)... (1 – R_n)$



- Consider identical modules with $R_i = 0.9$
- How can you increase R_{sys} to $0.999999 = 1\text{-}10^{-6}$
- Prob. of module *i* to survive = R_i
- Number of modules $n = \ln 10^{-6} / \ln (1\text{-}R_i) = 6$
- Hence, need 5 spares to make reliable system

Non-Series-Parallel-Systems

- "Success" diagram used to represent the operational modes of the system



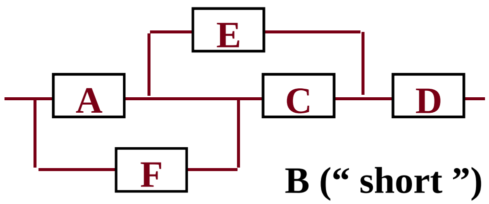
Each path from X to Y represents a configuration that leaves the system successfully operational

- Reliability of the system derived by expanding around a single module *m*

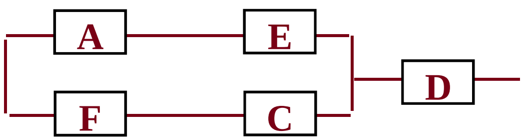
$R_{sys} = R_m$ P (system works | *m* works) + $(1 - R_m)$ P (system works | *m* fails)

where the notation P(s | m) denotes the conditional probability "s given, *m* has occurred"

Non-Series-Parallel-Systems (cont.)

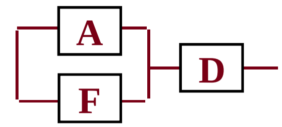


B (“ short ”) works



B (“ open ”) fails

Reduced model with B replaced



B (“ short ”) works, C (“ short ”) works

B (“ short ”) works, C (“ open ”) fails

Reduction with B and C replaced

$R_{sys} = R_B$ P(system works|B works)

$+ (1 - R_B) \{R_D[1 - (1 - R_A R_E)(1 - R_F R_C)]\}$

P(system works|B works) =

$R_C\{R_D[1 - (1 - R_A)(1 - R_F)]\}$

$+ (1 - R_C)(R_A R_D R_E)$

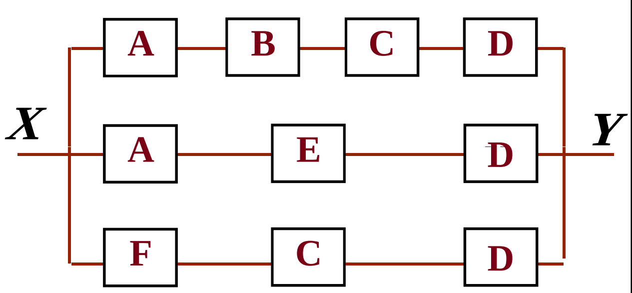
Letting $R_A = \ldots = R_F = R_m$ yields $R_{sys} = R^6_m - 3R^5_m + R^4_m + 2R^3_m$

Non-Series-Parallel-Systems (cont.)

- For complex success diagrams, an upper-limit approximation on R_{sys} can be used
- An upper bound on system reliability is:

$$R_{sys} \le 1 - \prod \left(1 - R_{path\ i}\right)$$

$R_{path\ i}$ is the serial reliability of path i



Reliability block diagram (RBD) of a system

The above equation is an upper bound because the paths are not independent.
That is, the failure of a single module affects more than one path.

$$R_{sys} \le 1 - \left(1 - R_A R_B R_C R_D\right)\left(1 - R_A R_E R_D\right)\left(1 - R_F R_C R_D\right)$$

$$R_{sys} \le 2R_m^3 + R_m^4 - R_m^6 - 2R_m^7 + R_m^{10}$$