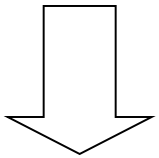
Quantitative evaluation of Dependability

Quantitative evaluation of Dependability

- Faults are the cause of errors and failures. Does the arrival time of faults fit a **probability distribution**?
  If so, what are the parameters of that distribution?

- Consider the time to failure of a system or component.
  It is not exactly predictable - **random variable**.



probability theory

Quantitative evaluation of failure rate, Mean Time To Failure (MTTF), Mean Time To Repair (MTTR), Reliability function (R(t)), Availability function (A(t)) and Safety function (S(t))

Quantitative definition of dependability attributes

Reliability - R(t)

conditional probability that the system performs correctly throughout the *interval of time* [t0, t], given that the system was performing correctly at the *instant* of time t0

Availability - A(t)

the probability that the system is operating correctly and is available to perform its functions at the *instant* of time t

Safety – S(t)

the probability that the system either behaves correctly or will discontinue its functions in a manner that causes no harm throughout the *interval of time* [t0, t], given that the system was performing correctly at the *instant* of time t0

Definitions

Reliability R(t)

$$R(0) = 1 \quad R(\infty) = 0$$

Failure probability Q(t)

$$Q(t) = 1 - R(t)$$

Failure probability density function f(t)

the failure density function f(t) at time t is the number of failures in Δt

$$f(t) = \frac{dQ(t)}{dt} = \frac{-\,dR(t)}{dt}$$

Failure rate function $\lambda(t)$

the failure rate $\lambda(t)$ at time t is defined by the number of failures during Δt in relation to the number of correct components at time t

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{-\,dR(t)}{dt}\,\frac{1}{R(t)}$$

Hardware Reliability

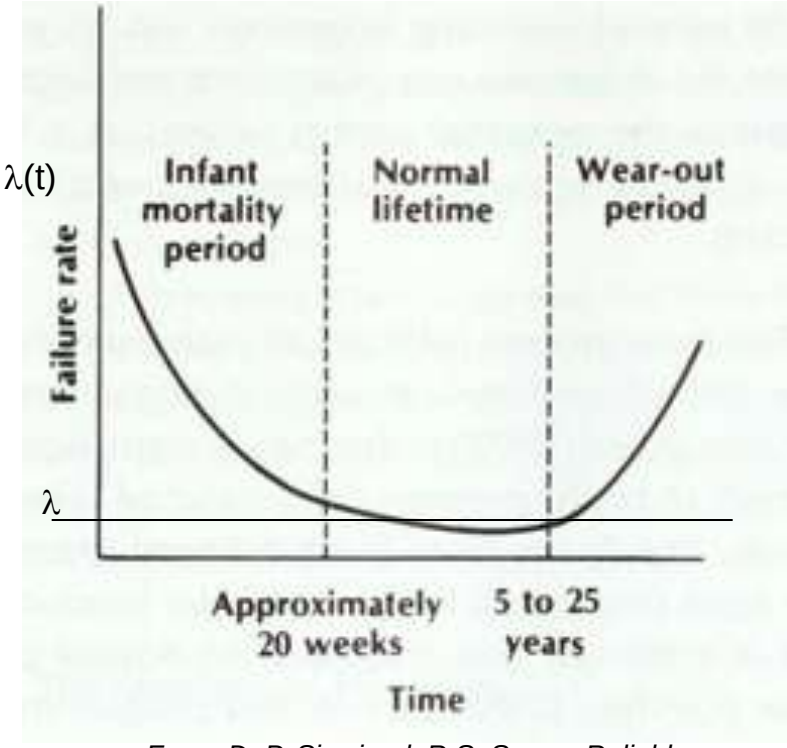
$\lambda(t)$ is a function of time
(bathtub-shaped curve)

$\lambda(t)$ constant > 0 in the useful life period

Constant failure rate λ

(usually expressed in number of failures for million hours)

λ = 1/2000
one failure every 2000 hours



From: D. P. Siewiorek R.S. Swarz, Reliable Computer Systems, Prentice Hall, 1992

Early life phase: there is a higher failure rate, calleld infant mortality, due to the failures of weaker components. Often these infant mortalities result from defetct or stress introduced in the manufacturing process.

Operational life phase: the failure rate is approximately constant.

Wear-out phase: time and use cause the failure rate to increase.

Hardware Reliability

Constant failure rate

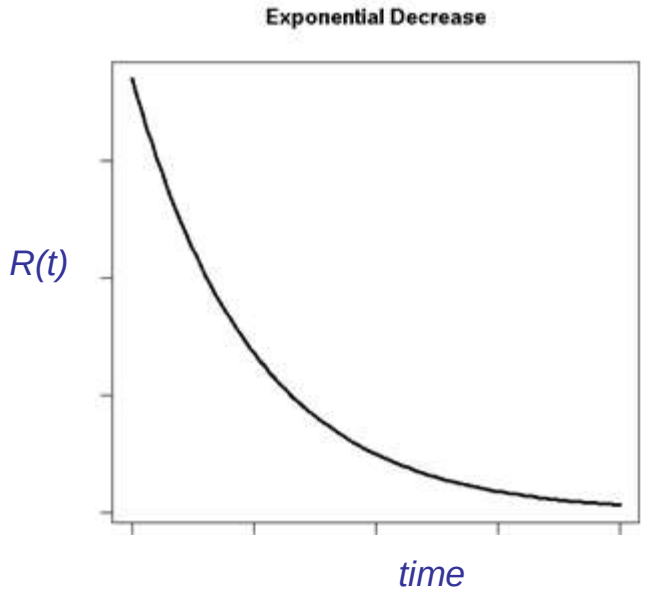
$\lambda(t) = \lambda$

Probability density function

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

Reliability function

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



the exponential relation between reliability and time is known as ***exponential failure law***

(e is the base of the natural log e= 2.718)

Time to failure of a component

Time to failure of a component can be modeled by a random variable X

$F_X(t)$ cumulative distribution function $P[X<=t]$

Unreliability of the component at time t is given by

$$Q_X(t) = P[X <= t] = F_X(t)$$

Reliability of the component at time t is given by

$$R_X(t) = P[X > t] = 1 - P[X <= t] = 1 - F_X(t)$$ **reliability function**

R(t) is the probability of not observing any failure before time t

Hardware Reliability

Mean time to failure (MTTF)

is the expected time that a system will operate before the first failure occurs (e.g., 2000 hours)

$$MTTF = \int_0^\infty t\, f(t)\, dt = \int_0^\infty t\, \lambda e^{-\lambda t}\, dt = \frac{1}{\lambda}$$

- λ = 1/2000 — 0.0005 per hour
- MTTF = 2000 — time to the first failure 2000 hours

Failure in time (FIT)

measure of failure rate in 10^9 device hours

- 1 FIT means 1 failure in 10^9 device hours

Failure Rate

- Handbooks of failure rate data for various components are available from government and commercial sources.

- Reliability Data Sheet of product

- **Commercially available databases**
  - Military Handbook MIL-HDBK-217F
  - Telcordia,
  - PRISM User's Manual,
  - International Eletrotechnical Commission (IEC) Standard 61508
  - …

  Databases used to obtain reliability parameters in
  ''Traditional Probabilistic Risk Assessment Methods
  for Digital Systems'',
  U.S. Nuclear Regulatory Commission,
  NUREG/CR-6962, October 2008

Distribution model for permanent faults

MIL-HBDK-217 (*Reliability Prediction of Electronic Equipment* -Department of Defence) is a model for chip failure. Statistics on electronic components failures are studied since 1965 (periodically updated).

Typical component failure rates in the range 0.01-1.0 per million hours.

Failure rate for a single chip :

$$\lambda = \pi_L \pi_Q (C_1 \pi_T \pi_V + C_2 \pi_E)$$

π_L = learning factor, based on the maturity of the fabrication process

π_Q = quality factor, based on incoming screening of components

π_T = temperature factor, based on the ambient operating temperature and the type of semiconductor process

π_E = environmental factor, based on the operating environment

π_V = voltage stress derating factor for CMOS devices

C_1, C_2 = complexity factors, based on the number of gates, or bits for memories in the component and number of pins.

From Reliable Computer Systems.D. P. Siewiorek R.S. Swarz, Prentice Hall, 1992

Model-based evaluation of dependability

MODEL-BASED evaluation of dependability
(a model is an abstraction of the system that highlights the important features for the objective of the study)

Dependability of a system is calculated in terms of the dependability of individual components

“divide And conquer approach”: the solution of the entire model is constructed on the basis of the solutions of individual sub-models

Methodologies that employ combinatorial models
Reliability Block Diagrams, Fault tree, ….

State space representation methodologies
Markov chains, Petri-nets, SANs, …

Packages for dependability evaluation

SHARPE

http://people.ee.duke.edu/~kst/

SHARPE, (Symbolic Hierarchical Automated Reliability and Performance Evaluator) is a tool for specifying and analyzing performance, reliability and performability models.

SURF-2

http://www.laas.fr/surf/surf-uk.html

SURF-2 tool for hardware and software systems, based on numerical resolution of Markov models. System behaviour is modelized by either a Markov Chain or a Generalized Stochastic Petri Net (GSPN).

UltraSAN

http://www.crhc.uiuc.edu/UltraSAN/UltraSAN.html

UltraSAN is a software package for hierarchical model-based

evaluation of systems represented as stochastic activity networks (SANs).

MOBIUS

http://www.mobius.uiuc.edu/

stochastic extensions to Petri nets, Markov chains and extensions, and stochastic process algebras

……. ….

Model-based evaluation of dependability

Combinatorial methods

offer simple and intuitive methods of the construction and solutions of models

independent components

each component is associated a failure rate

model construction is based on the structure of the systems (series/parallel connections of components)

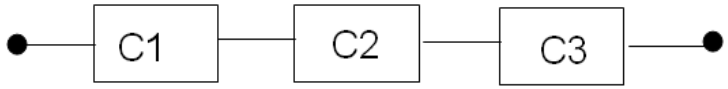
inadequate to deal with systems that exhibits complex dependencies among components and repairable systems

Combinatorial models

Series: all components must be operational (a)

$R_i(t)$ reliability of module i at time t

$R_{series}(t) = \Pi_{i=1}^{n} R_i(t)$
where Π is the product



(a)

If each individual component i satisfies the exponential failure law with constant failure rate λ_i:

$$R_{series}(t) = e^{-\lambda_1 t} ... e^{-\lambda_n t} = e^{-\sum_{i=1}^{n} \lambda_i t}$$

Unreliability function

$$Q_{series}(t) = 1 - R_{series}(t) = 1 - \Pi_{i=1}^{n} R_i(t) = 1 - \Pi_{i=1}^{n} [1 - Q_i(t)]$$

Combinatorial models

If the system does not contain any redundancy, that is any component must function properly for the system to work, and if component failures are independent, then

- the **system reliability** is the product of the component reliability, and it is exponential

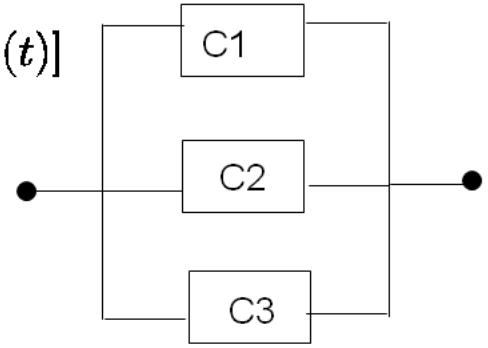
- the **failure rate of the system** is the sum of the failure rates of the individual components

Combinatorial models

Parallel: at least one of the components must be operational (b)

$$Q_{parallel}(t) = \Pi_{i=1}^{n} Q_i(t)$$
$$R_{parallel}(t) = 1 - Q_{parallel}(t) = 1 - \Pi_{i=1}^{n} Q_i(t) = 1 - \Pi_{i=1}^{n}[1 - R_i(t)]$$

Note the duality between Q and R in the two cases



(b)

M-of-N systems - a generalisation of parallel model
at least M modules of N are required to function

Assume N identical modules and M of those are required for the system to function properly, the expression for reliability of M-of-N substems can be written as:

$$R_{M-of-N}(t) = \Sigma_{i=0}^{N-M} \frac{N!}{(N-i)!i!} R^{N-i}(t)(1 - R(t))^i$$

$$\binom{n}{i} = \frac{n!}{(n-i)!\,i!}$$

i number of faulty components

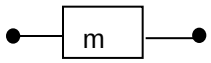
Combinatorial models

If the system contain redundancy, that is a subset of components must function properly for the system to work, and if component failures are independent, then

- the **system reliability** is the reliability of a series/parallel combinatorial model

TMR

Simplex system



λ failure rate of module m

$$R_m = e^{-\lambda t}$$

$$R_{simplex} = e^{-\lambda t}$$

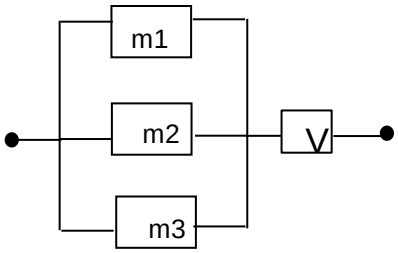
TMR system

$$R_V(t) = 1$$

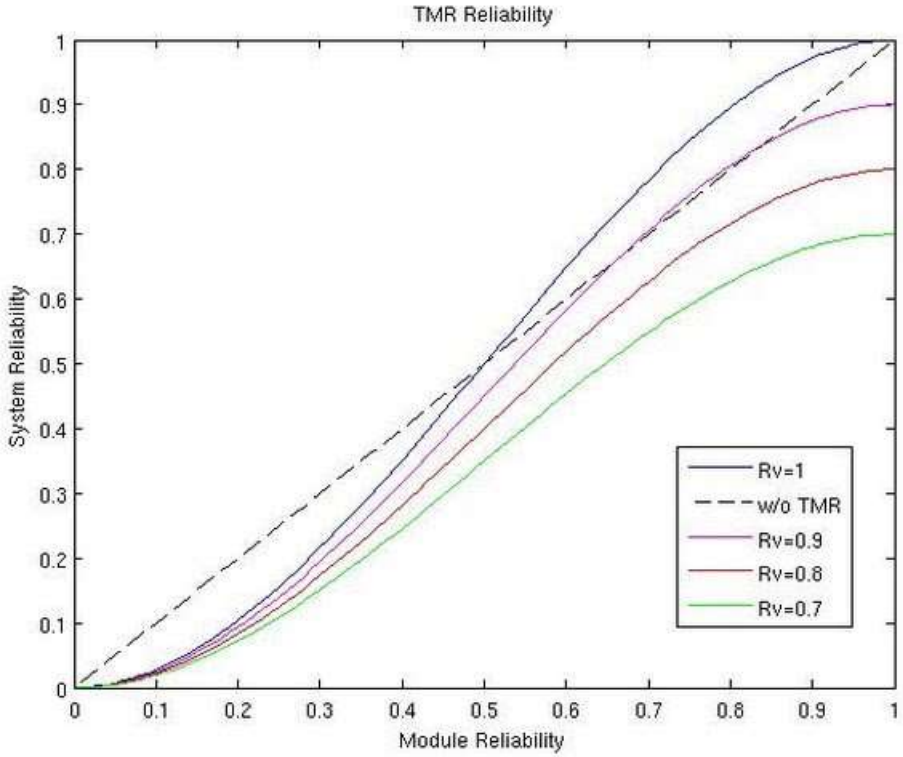
$$R_{TMR} = \sum_{i=0}^{1} \binom{3}{i} (e^{-\lambda t})^{3-i} (1 - e^{-\lambda t})^i$$

$$= (e^{-\lambda t})^3 + 3(e^{-\lambda t})^2 (1 - e^{-\lambda t})$$

$$R_{TMR} > R_m \text{ if } R_m > 0.5$$



2 of 3



From www.google.com

TMR: reliability function and mission time

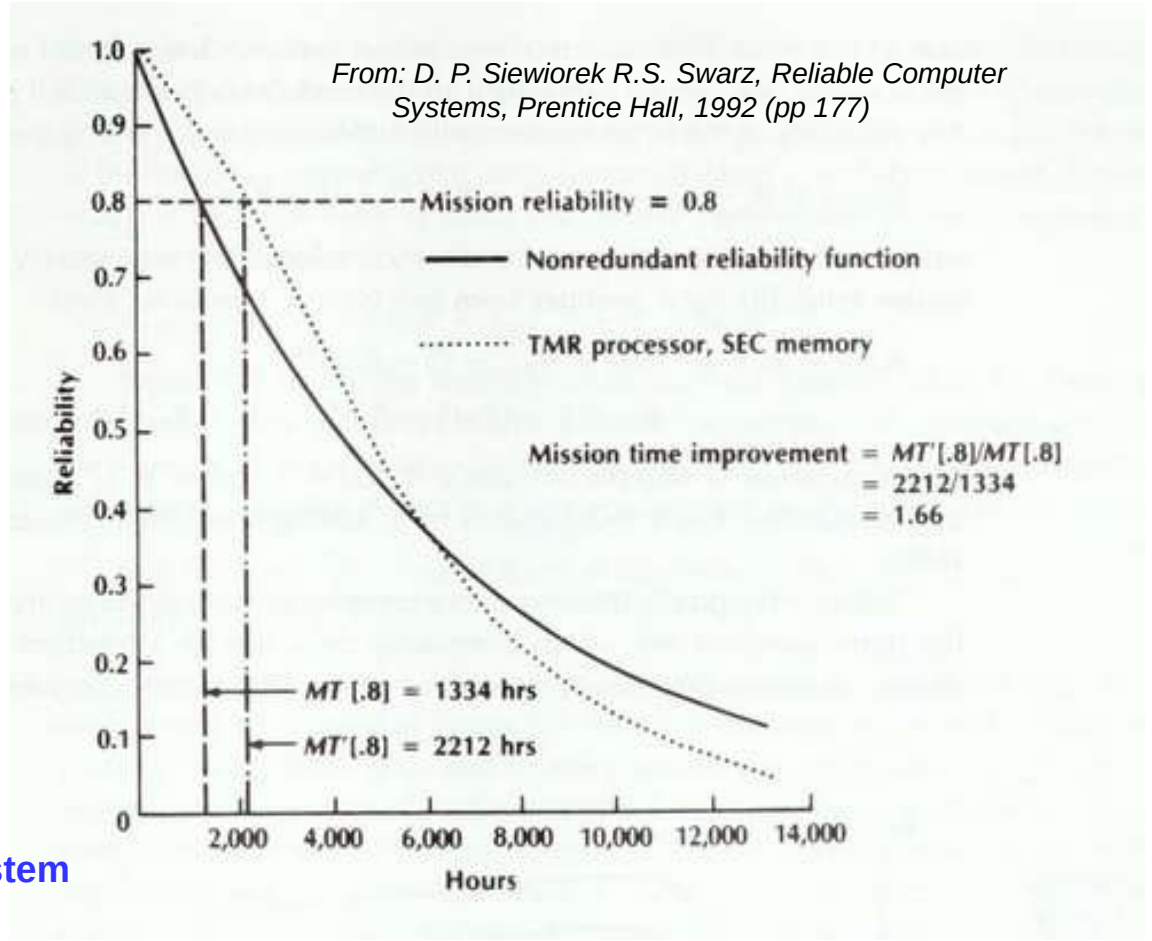
$R_{simplex} = e^{-\lambda t}$

$MTTF_{simplex} = \frac{1}{\lambda}$

TMR system

$R_{TMR} = 3e^{-2\lambda t} - 2e^{-3\lambda t}$

$MTTF_{TMR} = \frac{3}{2\lambda} - \frac{2}{3\lambda} = \frac{5}{6\lambda} < \frac{1}{\lambda}$



TMR worse than a simplex system !

TMR has a higher reliability for the first 6.000 hours of system life

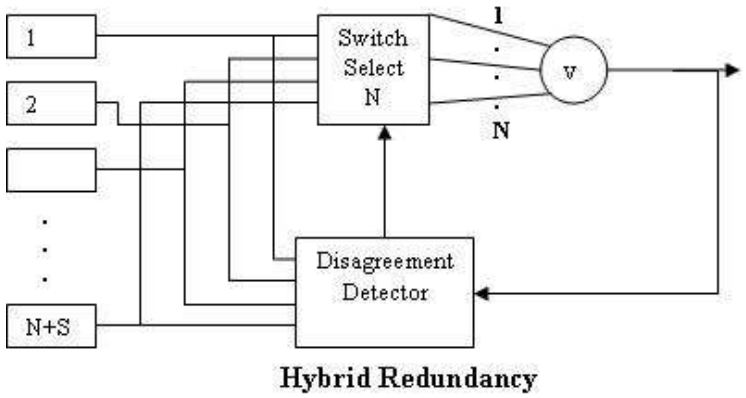
TMR operates at or above 0.8 reliability 66 percent longer than the simplex system

- **S shape curve is typical of redundant systems (there is the well known knee): above the knee the redundant system has components that tolerate failures; after the knee there is a sharper decrease of the reliability function in the redundant system (the system has exhausted redundancy, there is more hardware to fail than in the non redundant system)**

Hybrid redundancy with TMR

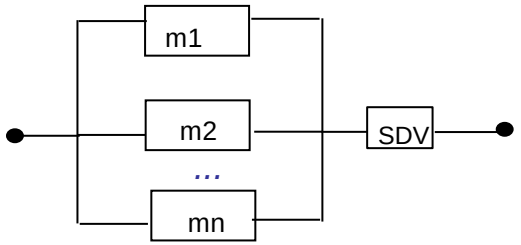
Symplex system
λ failure rate m
$R_m = e^{-\lambda t}$
$R_{sys} = e^{-\lambda t}$

Hybrid system
n total number of components
S number of spares



Hybrid Redundancy

$R_V(t) = 1 \quad R_{sw}(t) = 1$
λ failure rate of on line comp
λ failure rate of spare comp



The first system failure occurs if 1) all the modules fail; 2) all but one modules fail

$R_{Hybrid} = R_V R_{sw} (1- Q_{Hybrid})$

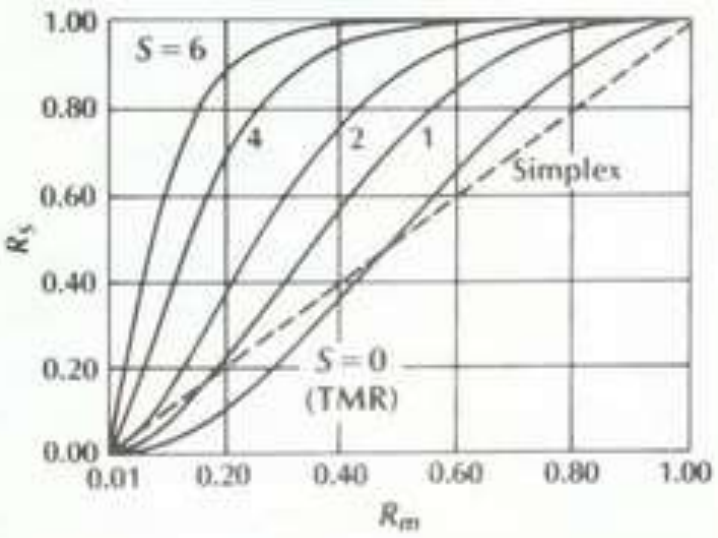
$R_{Hybrid} = (1 – ((1-R_m)^n + n(R_m)(1-R_m)^{n-1}))$

$R_{Hybrid(n+1)} – R_{Hybrid(n)} > 0$

adding modules increases the system reliability under the assumption Rsw independent of n

Hybrid redundancy with TMR

Hybrid TMR system reliability R_S vs individual module reliability R_m



S is the number of spares
R_V =1 R_{sw} =1

Figure 1. standby failure rate equal to on-line failure rate

the TMR with one spare is more reliable than simplex system if R_m>0.23

From: D. P. Siewiorek R.S. Swarz, Reliable Computer Systems, Prentice Hall, 1992 (pp 177)

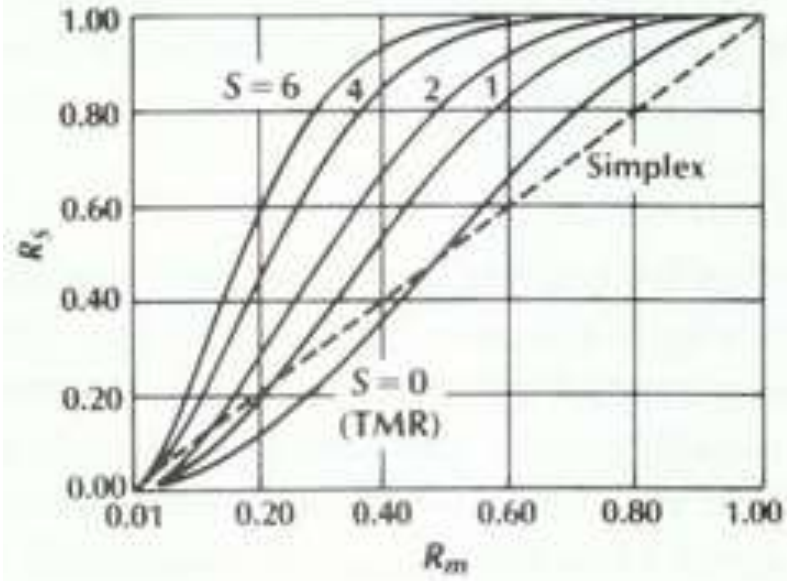
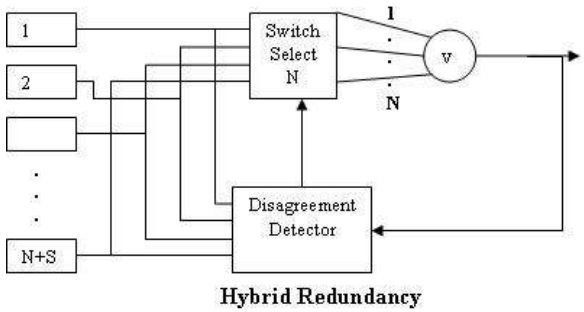


Figure 2. standby failure rate equal to 10% of on line failure rate

the TMR with one spare is more reliable than simplex system if R_m>0.17

Hybrid redundancy with TMR



Hybrid Redundancy

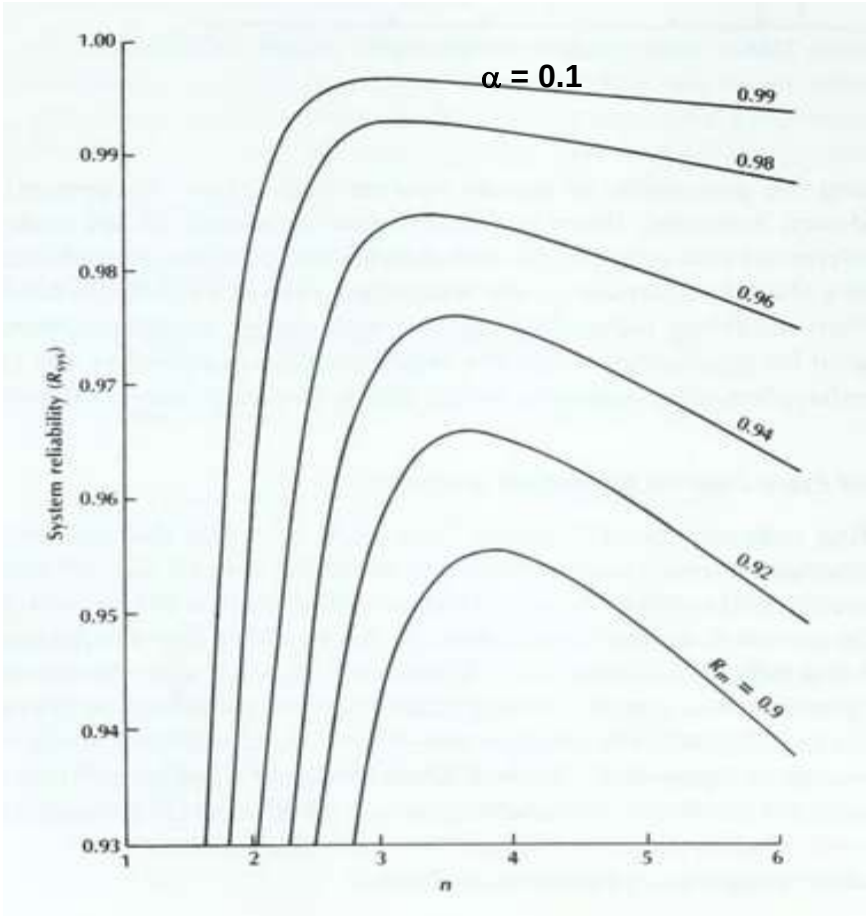
Typically complexity of the additional logic grows linearly with n

Let p the complexity of switch when a component is added

α relative complexity of the incremental switch with respect to a module

$$p = R_m^{\alpha}$$

$$R_{Hybrid} = R_m^{n(\alpha}(1 - ((1-R_m)^n + n(R_m)(1-R_m)^{n-1}))$$



Variation of R_{Hybrid} as a function of n, R_m and α

n= 1, …, 6

R_m = 0.9, 0.92, 0.94, 0.96, 0.98, 0.99

α = 0.1

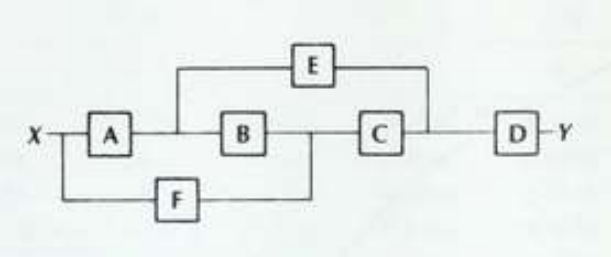
Maximum system reliability :

optimum number of components is higher for lower component reliability R_m

From: D. P. Siewiorek R.S. Swarz, Reliable Computer Systems, Prentice Hall, 1992

Non-series/nonparallel models

Succes diagram



System successfully operational
for each path from X to Y

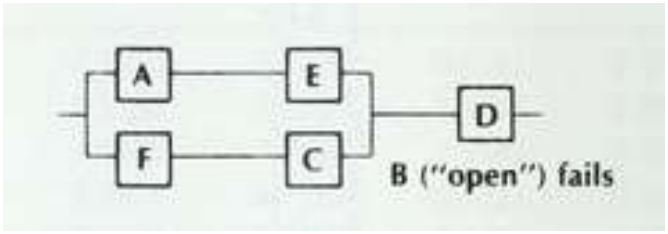
From: D. P. Siewiorek R.S. Swarz, Reliable Computer Systems, Prentice Hall, 1992

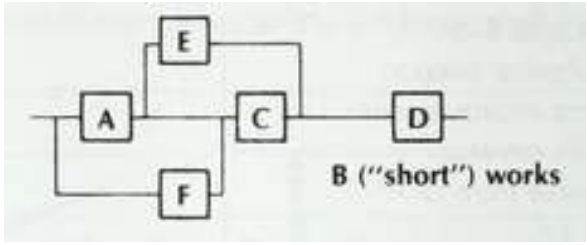
Reliability computed expanding around one module m:

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

Let m = B

R_{sys} = R_B x P(system works | B works) + (1- R_B) x P(system works | B fails)





P(system works | B fails) =
{ R_D [1 – (1- $R_A R_E$) (1- $R_F R_C$)]}

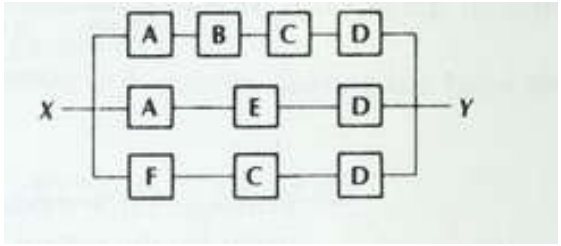
R_i=R_m

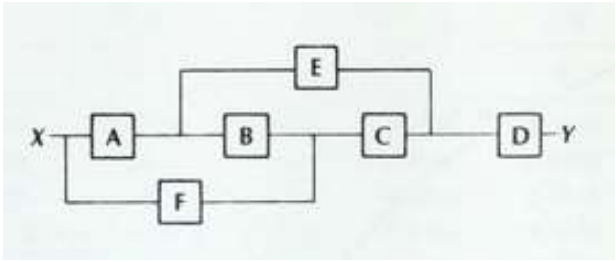
P(system works | B works)
must be further reduced
……

R_{Sys} <= $(R_m)^6$ - 3 $(R_m)^5$ + $(R_m)^4$ + 2$(R_m)^3$

Non-series/nonparallel lower-limit

Reliability Block Diagram: all path in parallel





From: D. P. Siewiorek R.S. Swarz, Reliable Computer Systems, Prentice Hall, 1992

Upper-bound:

$$R_{Sys} <= 1 - \Pi_i (1 - R_{path\ i})$$

Upper-bound because paths are not independent, the faiure of a single module affects more than one path (close approximation if paths are small)

Upper-bound:

$$R_{Sys} <= 1 - (1 - R_A R_B R_C R_D)(1 - R_A R_E R_D)(1 - R_F R_C R_D)$$

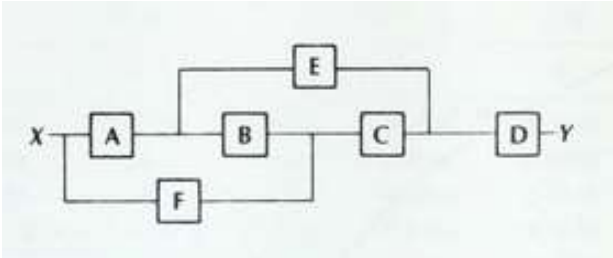
$R_i = R_m$

$$R_{Sys} <= 2 (R_m)^3 + (R_m)^4 - (R_m)^6 - 2 (R_m)^7 + (R_m)^{10}$$

Non-series/nonparallel lower-limit

Minimal cut sets of the system

Minimal cut set : is a list of components such that removal of any component from the list will cause the system to change from operational to failed



From: D. P. Siewiorek R.S. Swarz, Reliable Computer Systems, Prentice Hall, 1992

Minimal cut sets:
{D}{A,F}{E,C}{A,C}{BEF}

Lower-bound:

$$R_{Sys} >= \Pi_i (1 - Q_{cut\ i}) = \Pi_i\ R_{cut\ i}$$

where $Q_{cut\ i}$ is the probability that the cut i does not occur

Lower-bound:
R_i=R

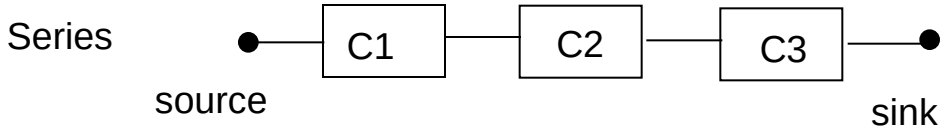
$$R_{Sys} >= R\ (1- (1-R)^2)^3\ (1- (1-R)^3)$$

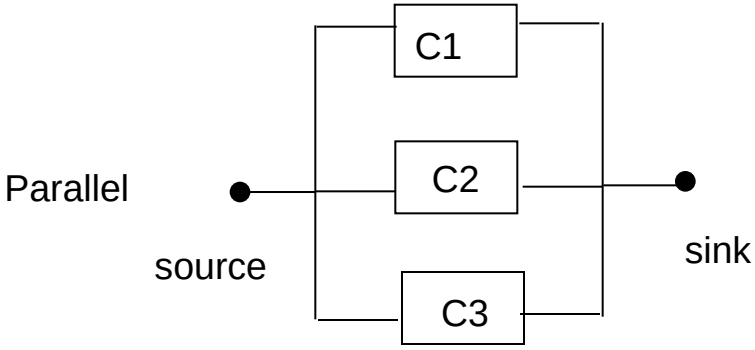
$$R_{Sys} >= 24\ R^5 - 60\ R^6 + 62\ R^7 - 33\ R^8 + 9\ R^9 - R^{10}$$

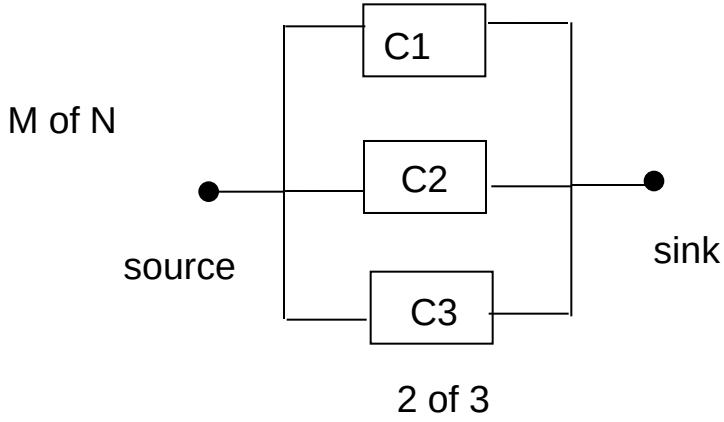
SHARPE tool
Reliability Blocks diagrams

- Blocks are components
  connected among them to represent the temporal order with which
  the system uses components, or the management of redundancy schemes
  or the success critera of the system

- System failure occurs if there is no path from source to sink



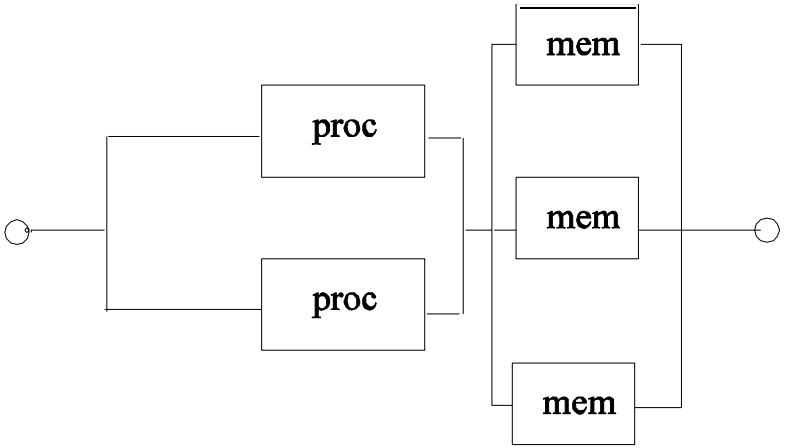




Example

Multiprocessor with 2 processors and three shared memories

-> analysis under different conditions



Series/Parallel

Assume $Q_p(t) = 0.0138$ with $t = 10$ days

$Q_m(t) = 0.00692$ with $t = 10$ days

```
1  block arch1(k,n,pfail,mfail)

2  comp proc prob(pfail)
3  comp mem prob(mfail)

4  parallel procs proc proc
5  kofn mems k,n,mem
6  series top procs mems
7 end

8  loop k,1,3,1
9  expr 1 - sysprob(arch1;k,2,.0138,.00692)
10 end
11 end
```

k operational memories, n operational processors,
pfail value of failure probablity of processor,
mfail value of failure probability of memories

bottom-up description of the system (top: serie of parallel modules)

sysprob(…) computes system failure probability

1-sysprob(…) reliability

Output:

```
K=1   1 - sysprob(arch1;k,2,.0138,.00692) :  0. 99981
K=2   1 - sysprob(arch1;k,2,.0138,.00692) :  0. 99967
K=3   1 - sysprob(arch1;k,2,.0138,.00692) :  0. 97920
```

We note that: increment of reliability is significant from three to two operational memories requirement

(after ten days, one memory: 99.8%; two memories: 99.7%;
three memories: 97.8%)

A failure probability function can be assigned to components by specifying the failure rate: the exponential failure law is assumed.