

DEPENDABLE SYSTEMS AND CRITICAL INFRASTRUCTURES DESIGN

RELIABILITY ENGINEERING AND HARDWARE FAULT-TOLERANCE

DIMITRIS AGIAKATSIKAS, MIHALIS PSARAKIS



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΜΣ ΚΥΒΕΡΝΟΣΦΑΛΕΙΑ
ΚΑΙ ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ

MSc CYBERSECURITY
AND DATA SCIENCE

DEPT OF INFORMATICS
UNIVERSITY OF PIRAEUS

Aviation



Automotive



Telecom



Space



Railway



Petroleum



Energy



Information Technology

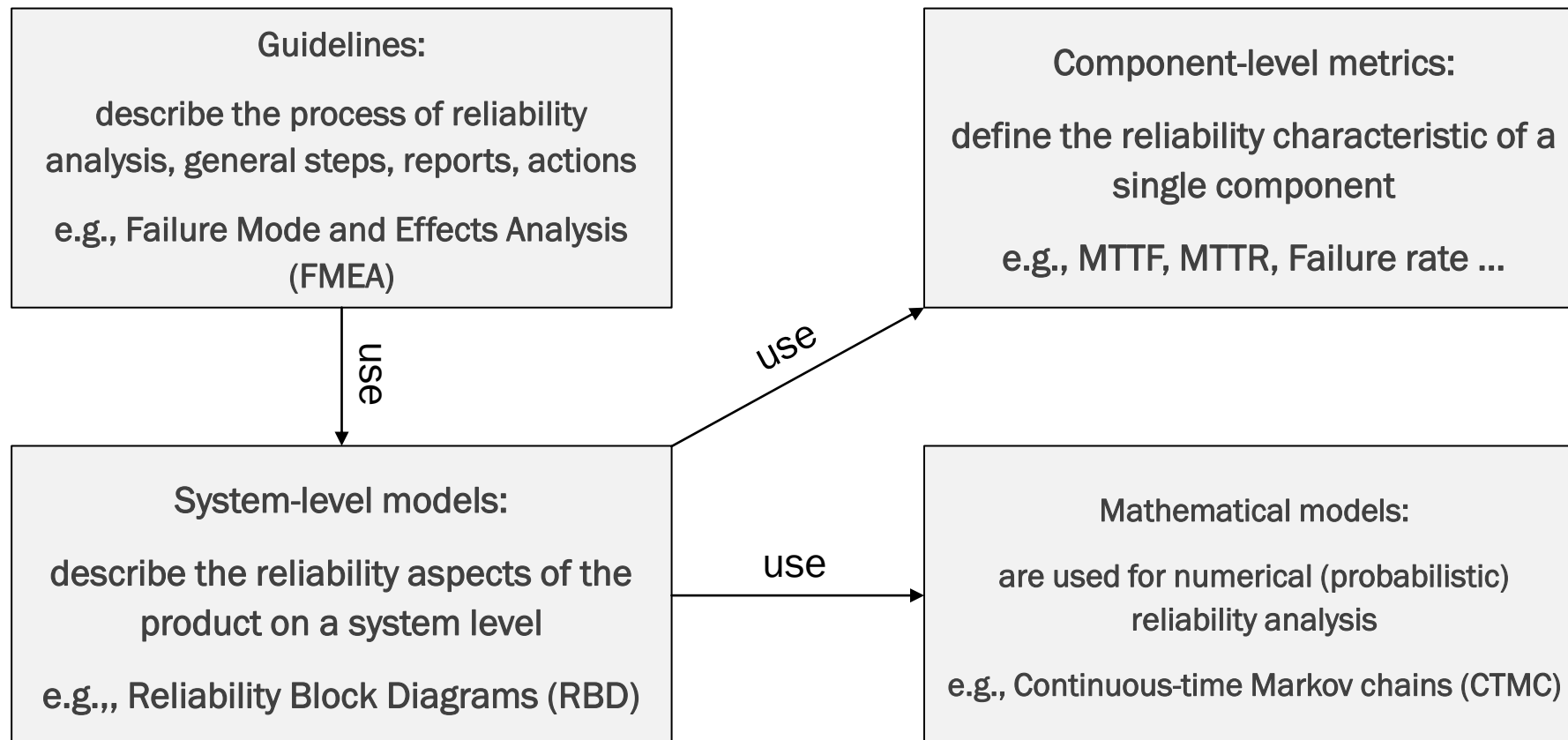


Army

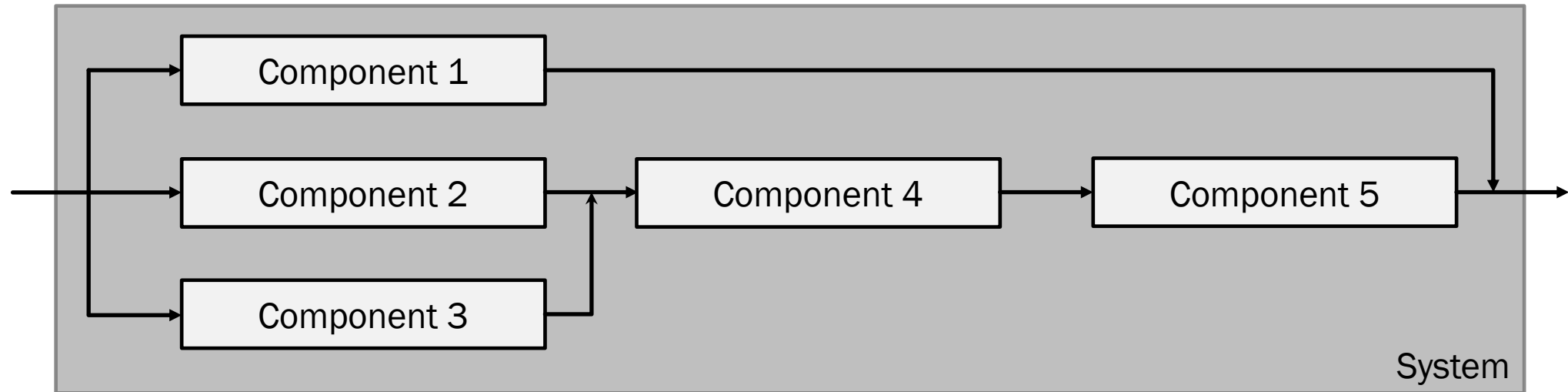



SECTION 1 RELIABILITY ENGINEERING

ROUGH CLASSIFICATION OF RELIABILITY ANALYSIS METHODOLOGIES





SYSTEM-LEVEL MODELS → RELIABILITY BLOCK DIAGRAMS (RBD)



 A RBD is a graphical depiction of the system's components and connectors which can be used to determine the overall system reliability.

 Blocks represent system components. Lines describe the connections between components.

 If any path through the system is successful, then the system succeeds, otherwise it fails.

 Examples of RBD models are the system in series, system in parallel and non-serial-parallel systems you have seen in previous lectures

RELIABILITY BLOCK DIAGRAMS (RBD) STANDARDS, TOOLS AND LIMITATIONS

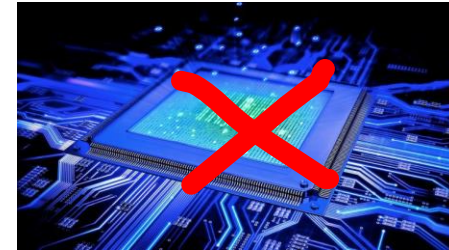
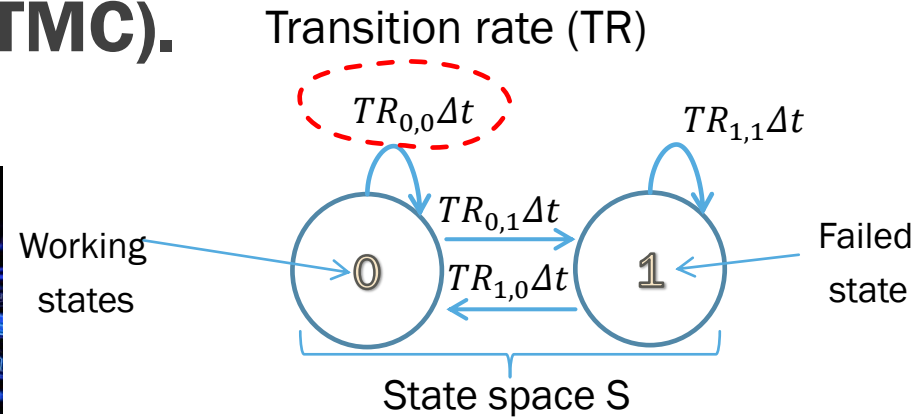
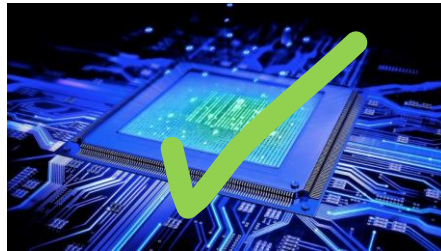
Standards and tools

- Standards: There are many standards that describe the procedure for modelling the dependability of a system and for using the RBD model in order to calculate reliability and availability measures. Examples are the International standard **IEC 61078 Ed. 2.0 b:2006**, the Australian standard **AS IEC 61078-2008** and the European Standard **DIN EN 61078**.
- Tools: There are many software tools to conduct RBD, such as the Relex RBD and the RBD Module.

Limitations:

The RBD modelling technique is applied primarily to systems without **repair** and **where the order in which failures occur does not matter**. For systems where the order of failures is to be taken into account or where repairs are to be carried out, other modelling techniques, such as **Markov chain analysis**, are more suitable.

MATHEMATICAL DEPENDABILITY MODELS → CONTINUOUS-TIME MARKOV CHAINS (CTMC).



- Markov modelling, named for the Russian mathematician Andrei Markov.
- Markov chain models are well suited for calculating the dependability of systems with conditional probabilities
- The underlying assumption of Markov models is that the state transition depends only on the current state, i.e., transitioning from state i to state j is independent of how it arrived in state i and how long it will stay at state i .
- The conditional probability of transitioning from one state to the next state can be assembled in a square transition matrix T .
- For example, a Markov chain model with two states will have
$$T = \begin{bmatrix} TR_{0,0}\Delta t & TR_{0,1}\Delta t \\ TR_{1,0}\Delta t & TR_{1,1}\Delta t \end{bmatrix}$$

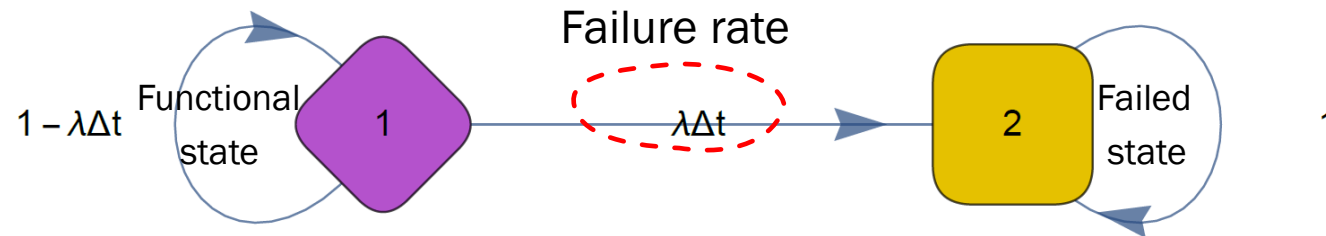
CONTINUOUS-TIME MARKOV CHAINS BASICS

$$T = \begin{bmatrix} TR_{0,0}\Delta t & TR_{0,1}\Delta t \\ TR_{1,0}\Delta t & TR_{1,1}\Delta t \end{bmatrix}$$

$$[p_1(t + \Delta t), p_2(t + \Delta t)] = [p_1(t), p_2(t)] \begin{bmatrix} TR_{0,0}\Delta t & TR_{0,1}\Delta t \\ TR_{1,0}\Delta t & TR_{1,1}\Delta t \end{bmatrix}$$

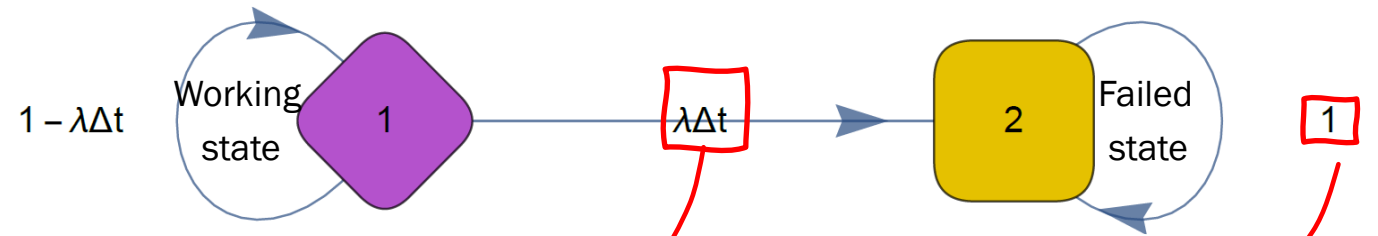
- All entries in the transition matrix T are non-negative
- The entries in each row shall sum to unity (1)
- Multiplying the vector of marginal probabilities of the different states at time t will give the marginal probabilities of the different states for the next time period Δt .

DERIVING THE RELIABILITY FUNCTION OF A NON-REPAIRABLE SIMPLEX SYSTEM (NO REDUNDANCY) WITH MARKOV CHAIN MODELLING



- From previous lectures we know that the Reliability of a non-repairable simplex system is $R(t) = e^{-\lambda t}$.
- In this lecture, we will introduce the Markov chain models by an example. We will derive the reliability $R(t) = e^{-\lambda t}$ equation of a non-repairable simplex system through Markov chain modelling
- The reliability of a non-repairable simplex system can be modelled as a Markov process with two states: Functional (state 1) and Failed (state 2).
- The system transitions functional to failed at a rate of λ , as labelled on the arc from state 1 to state 2.
- The reliability function of the Markov model depicted above can be derived by first converting the process to a continuous-time model and then solving a set of differential equations for the probability the system is in state 2, the failed state, at time t .

MARKOV CHAIN



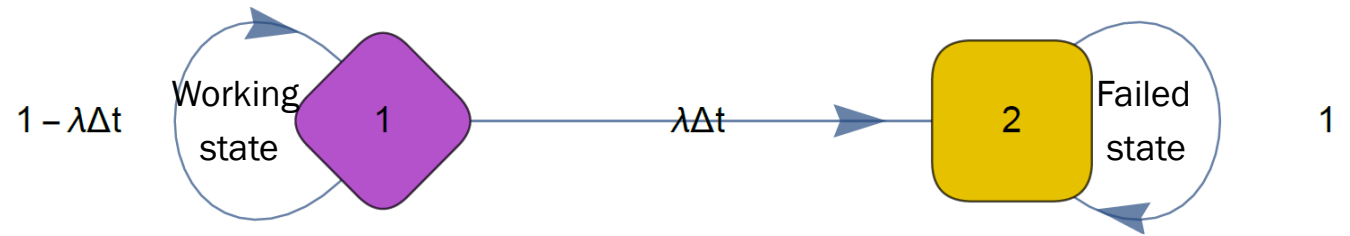
- Step 1: We create a transition matrix T for our model:

$$T = \begin{bmatrix} r(1,1) & r(1,2) \\ r(2,1) & r(2,2) \end{bmatrix} = \begin{bmatrix} 1 - \lambda\Delta t & \lambda\Delta t \\ 0 & 1 \end{bmatrix}$$

- Step 2: From T we define a set of equations that represent the probability of being in state 1 or 2 at time $t + \Delta t$. The distribution of probabilities at time $t + \Delta t$ is the product of the transition matrix T by the distribution of the probabilities at time t :

$$[p_1(t + \Delta t), p_2(t + \Delta t)] = [p_1(t), p_2(t)] \begin{bmatrix} 1 - \lambda\Delta t & \lambda\Delta t \\ 0 & 1 \end{bmatrix}$$

MARKOV CHAIN



- $[p_1(t + \Delta t), p_2(t + \Delta t)] = [p_1(t), p_2(t)] \begin{bmatrix} 1 - \lambda\Delta t & \lambda\Delta t \\ 0 & 1 \end{bmatrix}$

- Which yields

$$p_1(t + \Delta t) = p_1(t) * (1 - \lambda\Delta t) + p_2(t) * 0 \quad \Rightarrow p_1(t + \Delta t) = p_1(t) - \lambda\Delta t p_1(t) \quad \text{Eq. (1)}$$

$$p_2(t + \Delta t) = p_1(t) * \lambda\Delta t + p_2(t) * 1 \quad \Rightarrow p_2(t + \Delta t) = \lambda\Delta t p_1(t) + p_2(t) \quad \text{Eq. (2)}$$

Subtracting $p_1(t)$ from both sides of Eq.1, $p_2(t)$ from both sides of Eq.2, and dividing both (Eq.1 and Eq.2) by Δt yields:

$$\frac{p_1(t+\Delta t) - p_1(t)}{\Delta t} = -\lambda p_1(t) \quad \text{Eq. (3)}$$

and

$$\frac{p_2(t+\Delta t) - p_2(t)}{\Delta t} = \lambda p_1(t) \quad \text{Eq. (4)}$$

Definition of a derivative

MARKOV CHAIN

- Taking the $\lim_{\rightarrow 0}$ of Equations 3 and 4 yields:

$$p'_1(t) = -\lambda p_1(t) \quad \text{Eq.(5)} \quad \text{and} \quad p'_2 = \lambda p_1(t) \quad \text{Eq.(6)}$$

To easily solve the set of equations Eq.(5) and Eq.(6) we first transfer them to the LaPlace domain. The LaPlace transform of Equations 5 and 6 yields

$$\begin{aligned} \mathcal{L}\{p'_1(t)\} &= \mathcal{L}\{-\lambda p_1(t)\} \\ s P_1(s) - P_1(0) &= -\lambda P_1(s) \\ p_1(0) &= s P_1(s) + \lambda P_1(s) \end{aligned} \quad \text{Eq.(7)}$$

MARKOV CHAIN

Similarly, for Eq.(6) we have

$$\begin{aligned} \mathcal{L}\{p'_2(t)\} &= \mathcal{L}\{\lambda p_1(t)\} \\ s P_2(s) - p_2(0) &= \lambda P_1(s) \\ p_2(0) &= s P_2(s) - \lambda P_1(s) \end{aligned} \quad \text{Eq.(8)}$$

where $p_i(0) = p_i(t)$ at $t = 0$, i.e., $p_1(0), p_2(0)$ define the initial distribution of the system

Equations 7 and 9 can be written as :

$$[p_1(0), p_2(0)] = [P_1(s), P_2(s)] \begin{bmatrix} s + \lambda & -\lambda \\ 0 & s \end{bmatrix},$$

in matrix form

Solving for $[P_1(s), P_2(s)]$ gives

$$[P_1(s), P_2(s)] = [p_1(0), p_2(0)] \begin{bmatrix} s + \lambda & -\lambda \\ 0 & s \end{bmatrix}^{-1}$$

The inverse of the matrix yields

$$[P_1(s), P_2(s)] = [p_1(0), p_2(0)] \begin{bmatrix} \frac{1}{s+\lambda} & \frac{\lambda}{(s+\lambda)s} \\ 0 & \frac{1}{s} \end{bmatrix} \quad \text{Eq.(13)}$$

Assuming that the system starts at state 1 (working state), the initial probability distribution is

$$[p_1(0), p_2(0)] = [1, 0] \quad \text{Eq.(14)}$$

Substituting Eq.(14) into Eq.(13) yields

$$[P_1(s), P_2(s)] = [1, 0] \begin{bmatrix} \frac{1}{s+\lambda} & \frac{\lambda}{(s+\lambda)s} \\ 0 & \frac{1}{s} \end{bmatrix}$$

$$P_1(s) = \frac{1}{s+\lambda} \quad \text{and} \quad P_2(s) = \frac{\lambda}{(s+\lambda)s}$$

MARKOV CHAIN

- Last, we transfer back to the time domain by taking the inverse Laplace of $P_1(s)$ which gives the reliability function $R(t)$ of the non-repairable simplex system:

$$\begin{aligned}\mathcal{L}^{-1}\{P_1(s)\} &= \mathcal{L}^{-1}\left\{\frac{1}{s+\lambda}\right\} \\ R(t) = p_1(t) &= e^{-\lambda t}\end{aligned}$$

Similarly, we can find the probability of being in state 2 by taking the inverse Laplace of $P_2(s)$

$$\begin{aligned}\mathcal{L}^{-1}\{P_2(s)\} &= \mathcal{L}^{-1}\left\{\frac{\lambda}{(s+\lambda)s}\right\} \\ p_2(t) &= 1 - e^{-\lambda t}\end{aligned}$$

- Since the probabilities of all states in a Markov chain model should add to 1 (i.e., $\sum_{i=0}^K p_i(t) = 1$) we can subtract $p_2(t)$ from unity (1), which gives the probability of being in state 1

$$R(t) = 1 - p_2(t) = 1 - (1 - e^{-\lambda t}) = e^{-\lambda t}$$

MARKOV CHAIN

- Markov chains provide a nice structural representation of the system's states, but the derivation of dependability expressions is quite tedious and becomes more demanding as the states of the chain increase.
- Fortunately, there are plenty of Markov chain software tools, such as Relex Markov, SHARPE and Isograph Markov. Most of these tools perform numerical Markov chain analysis rather than symbolic.
- Wolfram Mathematica is a nice tool that focuses on symbolic computation of Markov chain models. For example, the reliability of a non-repairable simplex system can be derived with the following code:

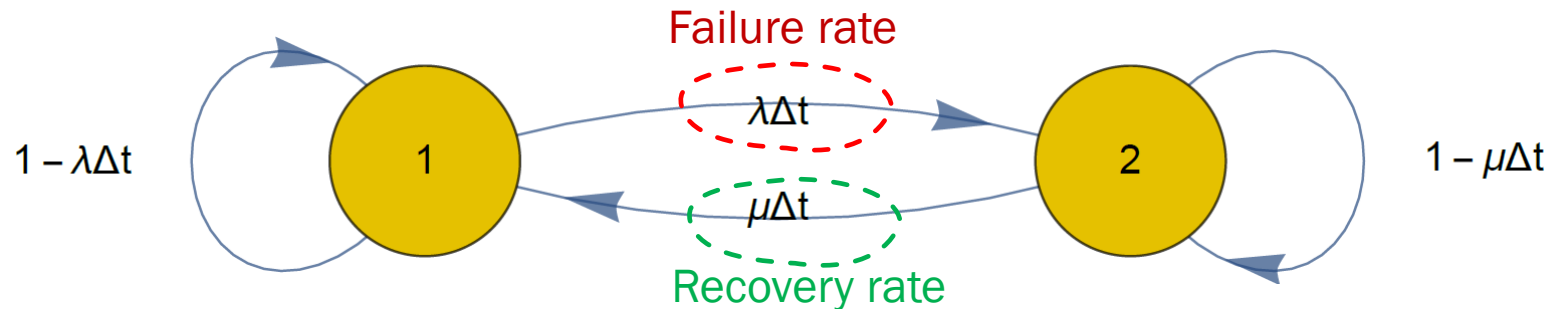
code:

```
T = { {1 - λ}, λ}, {0, 1} }  
initialDistribution = {1, 0};  
proc = ContinuousMarkovProcess [initialDistribution ,T] ;  
R = FullSimplify[PDF[ proc [ t ] , 1 ] ]
```

output:

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

DERIVING THE AVAILABILITY AND STEADY-STATE AVAILABILITY OF A REPAIRABLE SIMPLEX SYSTEM WITH MARKOV CHAIN MODELLING



- Now that we know how to use Mathematica, let's accelerate the derivation of the availability function with the following code, which finds the probability distribution of the system being in state 1 (working)

code:

```
T = { {1 - λ}, λ}, {μ, 1 - μ} }  
initialDistribution = {1, 0};  
proc = ContinuousMarkovProcess [initialDistribution ,T] ;  
A = FullSimplify[PDF[ proc [ t ] , 1 ] ]
```

output:

$$A = \frac{\mu}{\lambda + \mu} + \frac{\lambda e^{-t(\lambda + \mu)}}{\lambda + \mu}$$

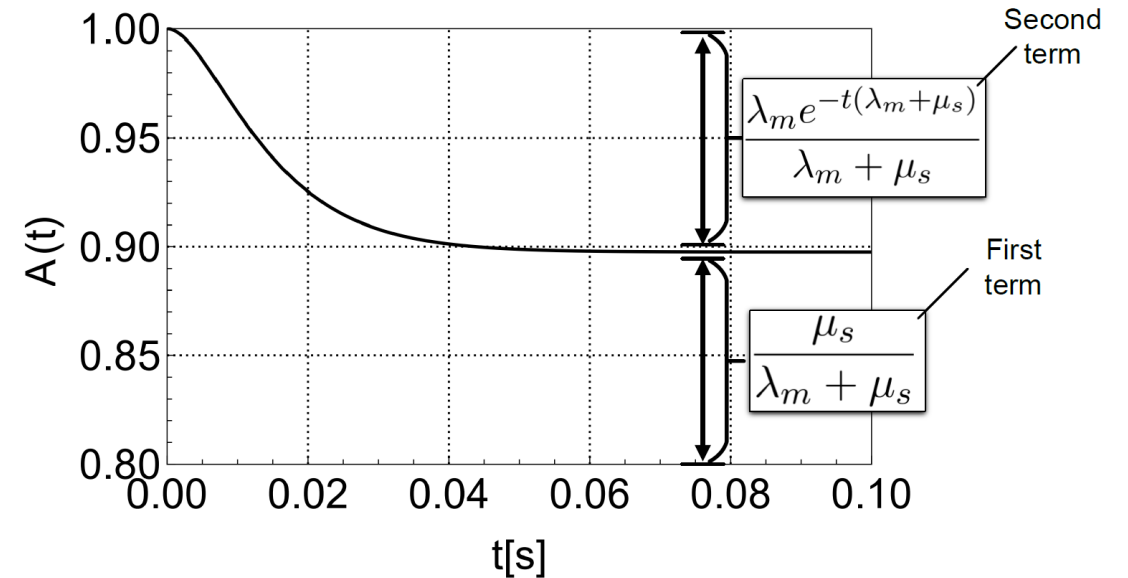
AVAILABILITY VS STEADY AVAILABILITY

- Let's examine the Availability $A(t)$ function

$$A(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda e^{-t(\lambda + \mu)}}{\lambda + \mu} \quad \text{Eq. (14)}$$

- The system starts at state 1 (working) and at time $t=0$, the availability $A(0) = 1$
- As the time passes, the system reaches the steady state availability
- In Eq. (14), the first term captures the steady state expression, and the second term captures the transitory behaviour as the system approaches steady state.
- Therefore, the steady state A of the system is

$$A = \lim_{t \rightarrow \infty} A(t) = \frac{\mu}{\lambda + \mu} + \frac{\cancel{\lambda} e^{-t(\lambda + \mu)}}{\cancel{\lambda} + \mu} = \frac{\mu}{\lambda + \mu}$$





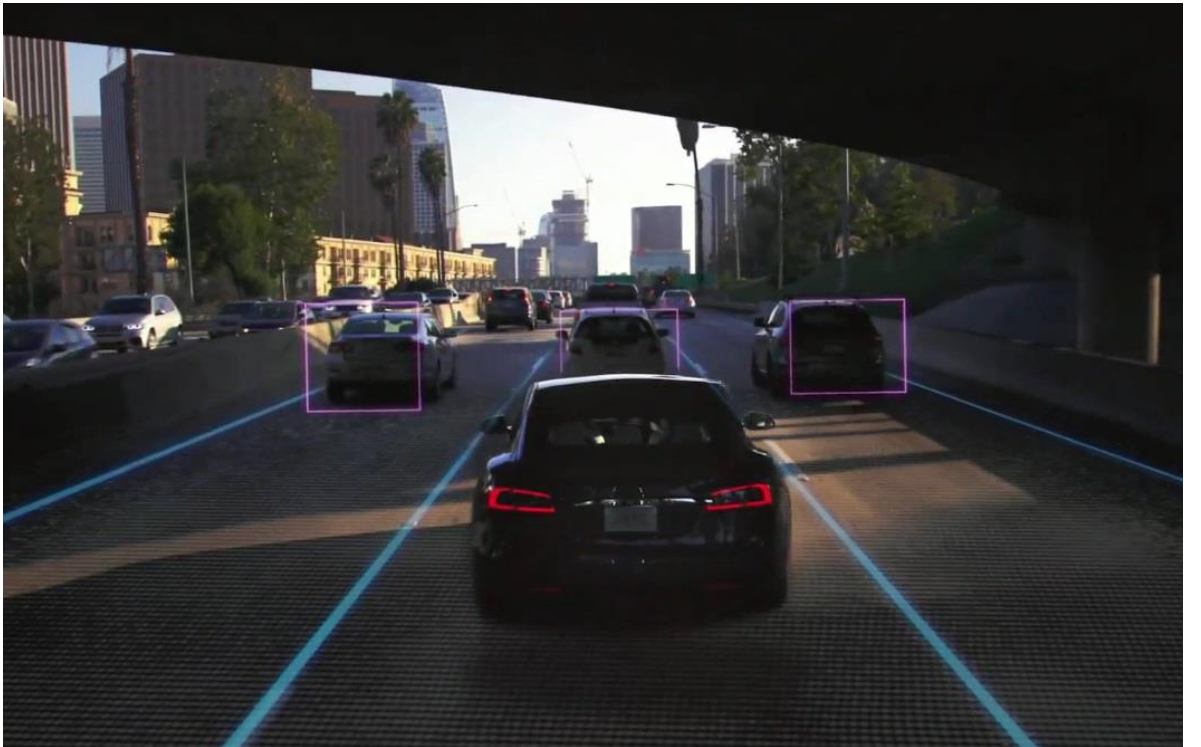
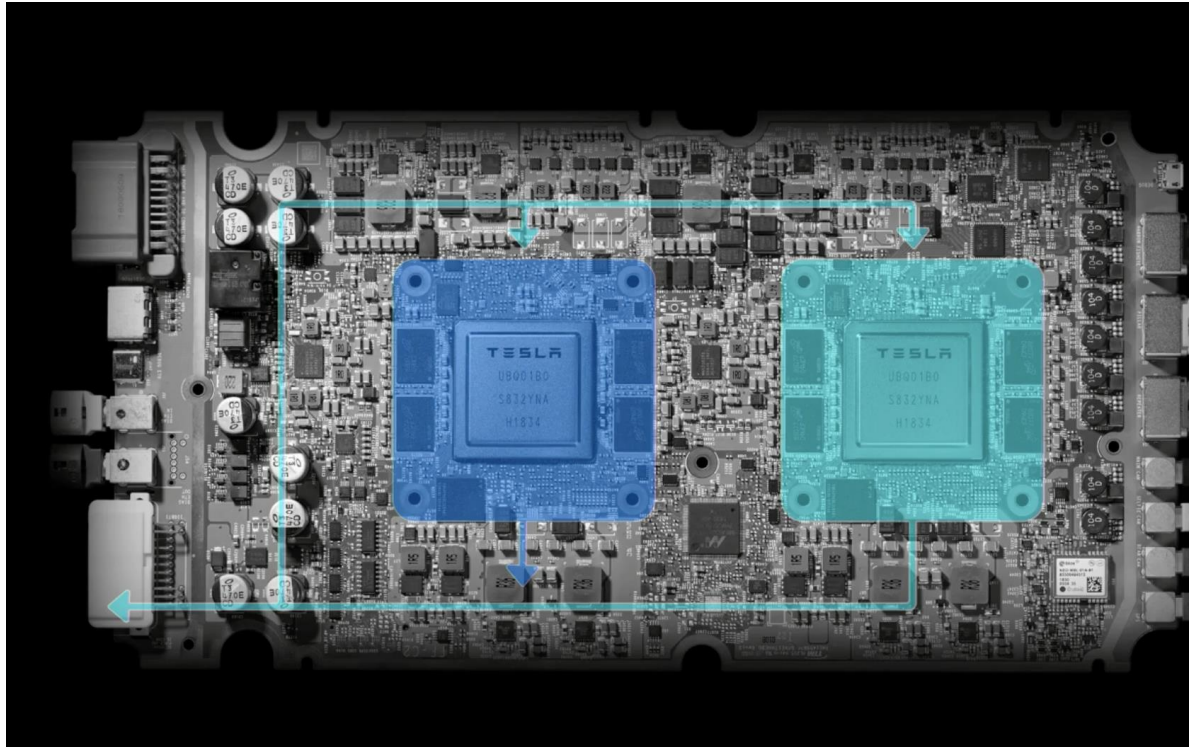
SECTION 2

FAULT-TOLERANCE

FAULT-TOLERANCE INCORPORATES SOME FORM OF REDUNDANCY TO DETECT AND/OR MASK ERRORS

Types of redundancy:

- **HARDWARE** → Example: Instead of having one CPUs, we can use two or three CPUs, each performing the same function, to detect and/or mask errors, respectively.
- **INFORMATION** → Example: Extra (redundant) bits are added to the original data bits of a RAM so that an error in the data bits can be detected and corrected. These are usually called Error Correction Codes (ECC).
- **TIME** → Example: Re-execute the same program on the same CPU to detect transient faults.
- **SOFTWARE** → Example: Mitigate software faults (bugs) by independently producing (from disjoint teams of programmers) two versions of a software in the hope that the different versions will not fail on the same input.



HARDWARE FAULT-TOLERANCE

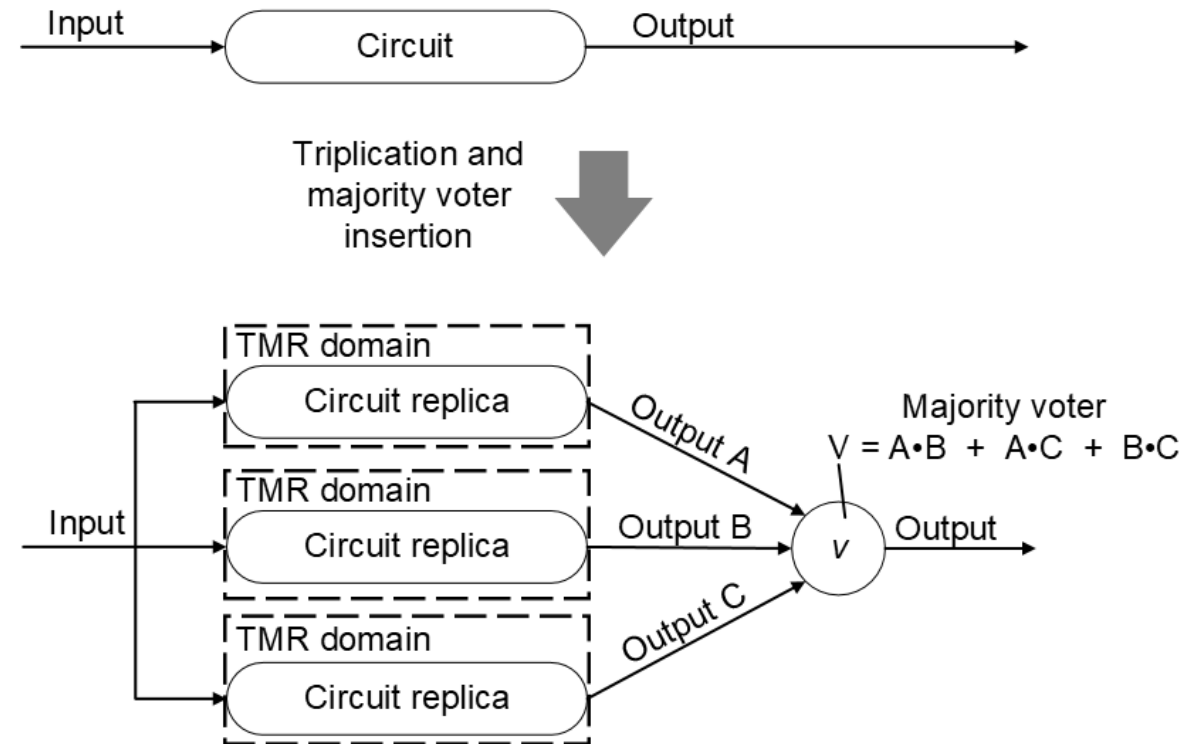
NON-REPAIRABLE M-OF-N SYSTEM

- An M-of-N system consists of N modules and needs at least M of them for proper operation.
- The key assumption for a high reliability system of M-of-N system is the proper isolation of faults between modules (or otherwise fault containment). Even a slightly extend of positive correlated (i.e., common-mode) failures has a greatly impact on overall reliability.
- The best-known example is the triplex system or Triple Modular Redundant (TMR) system that will be presented in detail in the next slide.

EXAMPLE OF HARDWARE REDUNDANCY TO MASK FAULTS/ERRORS (SPATIAL) TRIPLE MODULAR REDUNDANCY (TMR)

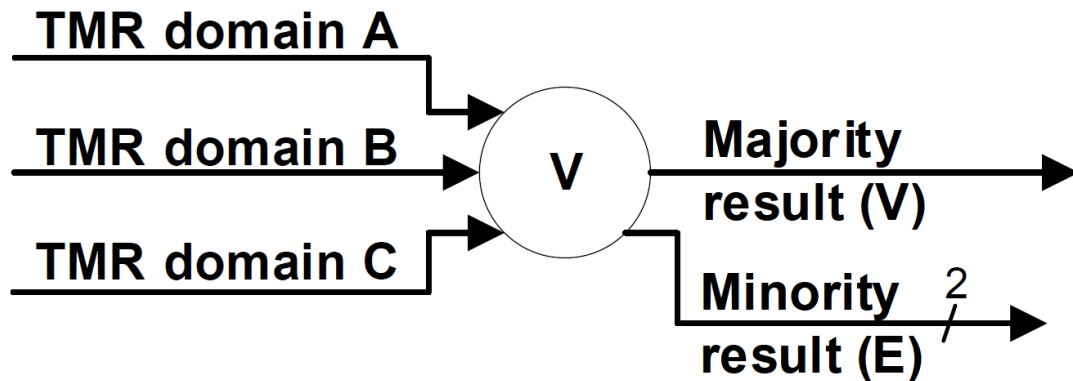
TMR Basics

- The basic concept of TMR is to replicate a circuit three times, provide the three identical circuits with the same input stimulus, and perform bit-wise majority voting on the outputs of each circuit replica
- The majority voter (V) simply masks any erroneous result from a faulty TMR by outputting the result corresponding to at least two of its inputs.
- Each circuit replica is referred to as TMR domain or module of the TMR scheme.



BITWISE TMR MAJORITY VOTER WITH ERROR LOCALISATION

- The majority voter is used to mask any number of errors from a single TMR domain
- The minority voter is used to detect and report which TMR domain is faulty
- N voters are used for each bit of a N-bit signal

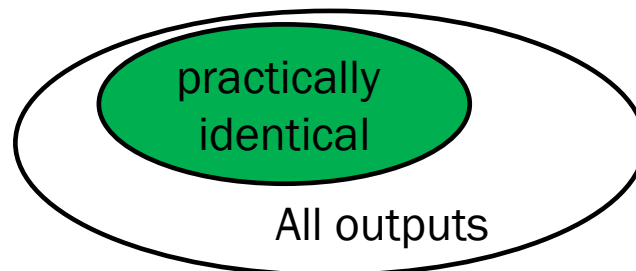


Truth table of bitwise the majority voter (V) with error localisation (E)

| A | B | C | V | E | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

PLURALITY VOTERS

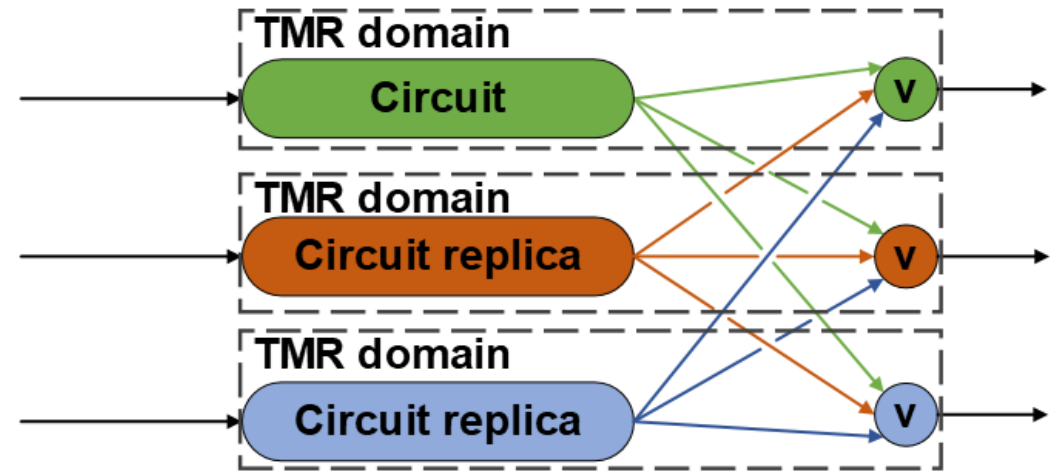
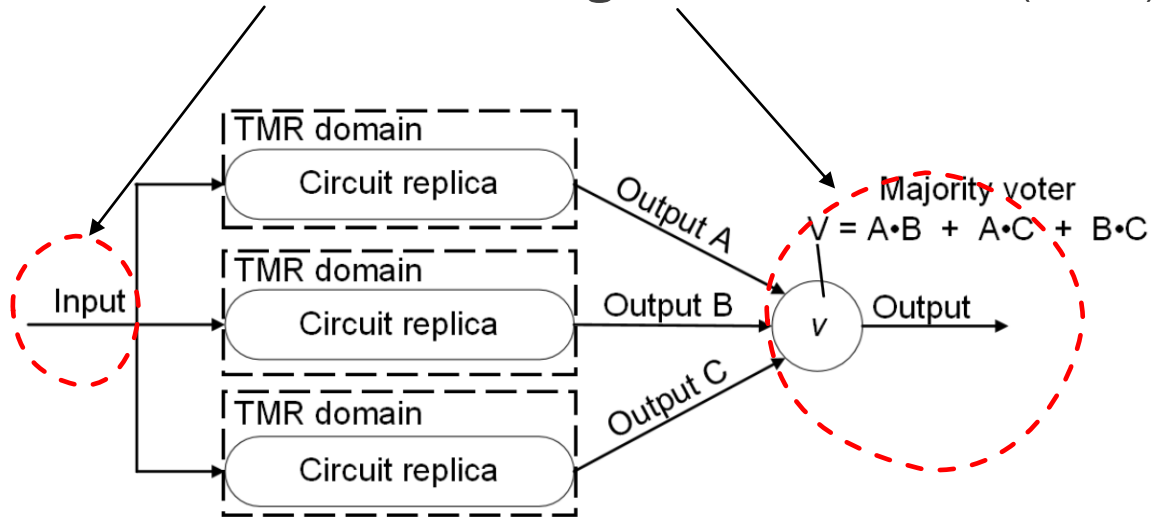
- Bitwise voters work well when we can guarantee that every module will generate an output that matches the output of every other functional module, bit-by-bit.
Example → M identical processors that use identical inputs and identical software, and have tightly synchronized clocks
- To implement M-of-N redundancy in systems with different modules (e.g., different CPUs or different software for the same problem), we can declare two outputs x and y as “practical identical” if $|x - y| < \delta$ for some specified δ
- The plurality voter looks for a set of k practical identical outputs and picks their median as the representative
- For example, if we set $\delta = 0.1$, and the five outputs were **1.10, 1.11, 1.32, 1.49, 3.00**, then the subset **{1.10, 1.11}** would be selected by a **K=2** plurality voter and the representative **1.105**



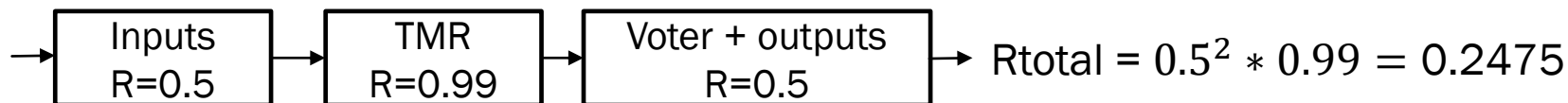
TMR WITH TRIPLICATED INPUTS, MAJORITY VOTERS AND OUTPUTS

I/Os and majority voter shall be triplicated, otherwise the circuit is prone to Common Mode Failures or otherwise Single Point of Failure (SPFs)

Fully triplicated circuit, no SPFs



SPFs dictate the total reliability of the TMR system and shall be eliminated.



RELIABILITY OF A NON-REPAIRABLE TMR SYSTEM

Reliability by using M-of-N general function of a Reliability Block Diagram (RBD) analysis

- $$R_{M \text{ out of } N}(t) = \sum_{i=M}^N \binom{N}{i} R^i(t) [1 - R(t)]^{N-i}$$

where $\binom{N}{i} = \frac{N!}{(N-i)!i!}$

- $$R_{2 \text{ out of } 3}(t) = \sum_{i=2}^3 \binom{3}{i} R^i(t) [1 - R(t)]^{3-i}$$

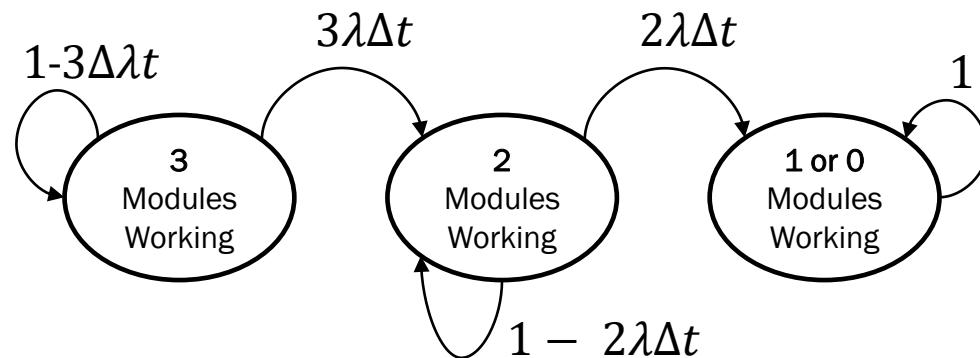
$$= \binom{3}{2} R^2(t) [1 - R(t)]^{3-2} + \binom{3}{3} R^3(t) [1 - R(t)]^{3-3}$$

$$= 3 R^2(t) - 2 R^3(t)$$

- For constant failure rate $e^{-\lambda t}$ we therefore have

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

Reliability by using Markov chain modeling



code:

```

T = {{(1 - 3 λ), 3 λ, 0}, {0, (1 - 2 λ), 2 λ}, {0, 0, 1}};
Init = {1, 0, 0};
proc = ContinuousMarkovProcess[Init, T];
rTMRNR = FullSimplify[1 - PDF[proc[t], 3]]
  
```

output:

$$3 e^{-2\lambda t} - 2 e^{-3\lambda t}$$

RELIABILITY OF A REPAIRABLE TMR SYSTEM

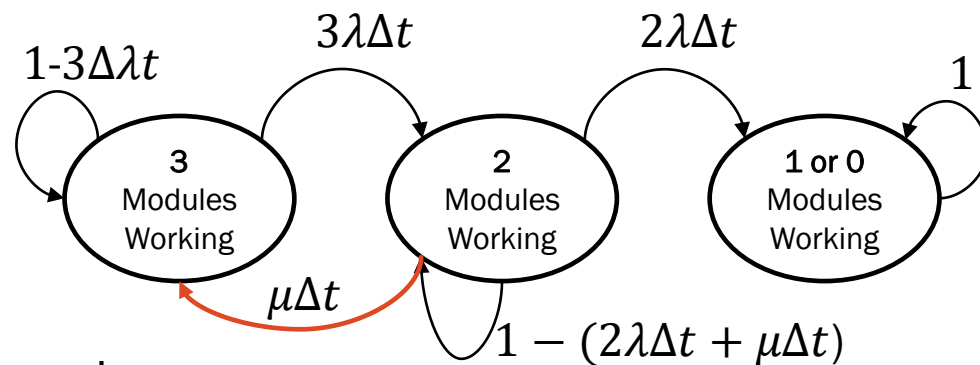
Reliability by using M-of-N general function

- Very difficult to evaluate in this way repairable systems
- We use Markov chain modelling instead

As the states increase and repair techniques are incorporated, the R(t) functions become tedious.

In practice, we only evaluate R(t) functions numerically and don't care of symbolic expressions

Reliability by using Markov chain modeling



code:

```

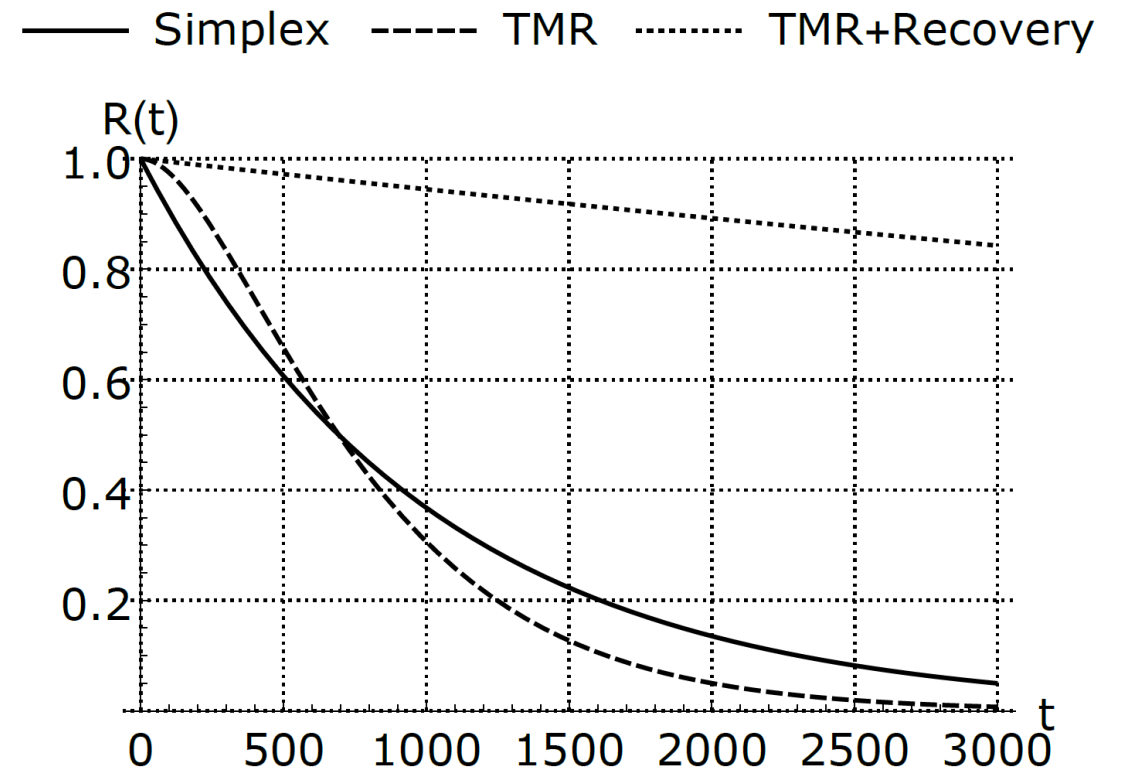
T = {{(1 - 3 λ), 3 λ, μ}, {0, (1 - (2 λ + μ)), 2 λ}, {0, 0, 1}};
Init = {1, 0, 0};
proc = ContinuousMarkovProcess[Init, T];
rTMRNR = FullSimplify[1 - PDF[proc[t], 3]]
  
```

output:

$$e^{-\frac{1}{2}t(5\lambda+\mu)} \left(\cosh\left[\frac{1}{2}t\sqrt{\lambda^2+10\lambda\mu+\mu^2}\right] + \frac{(5\lambda+\mu)\sinh\left[\frac{1}{2}t\sqrt{\lambda^2+10\lambda\mu+\mu^2}\right]}{\sqrt{\lambda^2+10\lambda\mu+\mu^2}} \right)$$

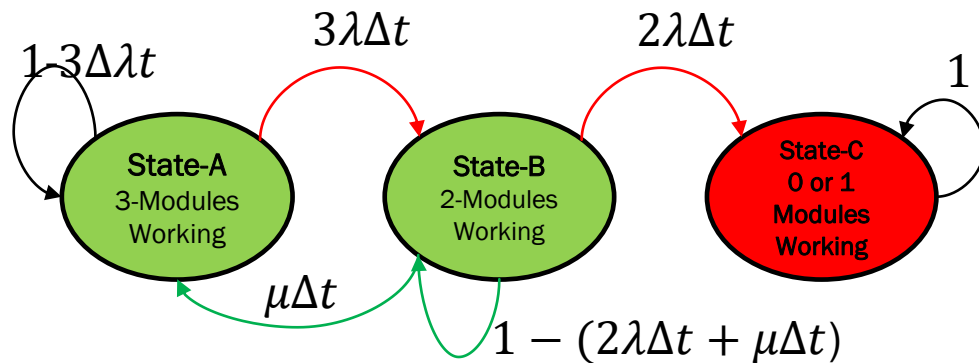
RELIABILITY: SIMPLEX VS NON-REPAIRABLE TMR VS REPAIRABLE TMR

- The reliability of a non-repairable TMR system is only higher than its functionally equivalent simplex circuit in its early operation.
- The reason for this counter-intuitive fact is that once one module of the TMR circuit fails, the remaining two healthy modules have a higher probability of failing than the simplex circuit itself.
- The reliability of a TMR FPGA circuit increases considerably when combined with error recovery mechanisms

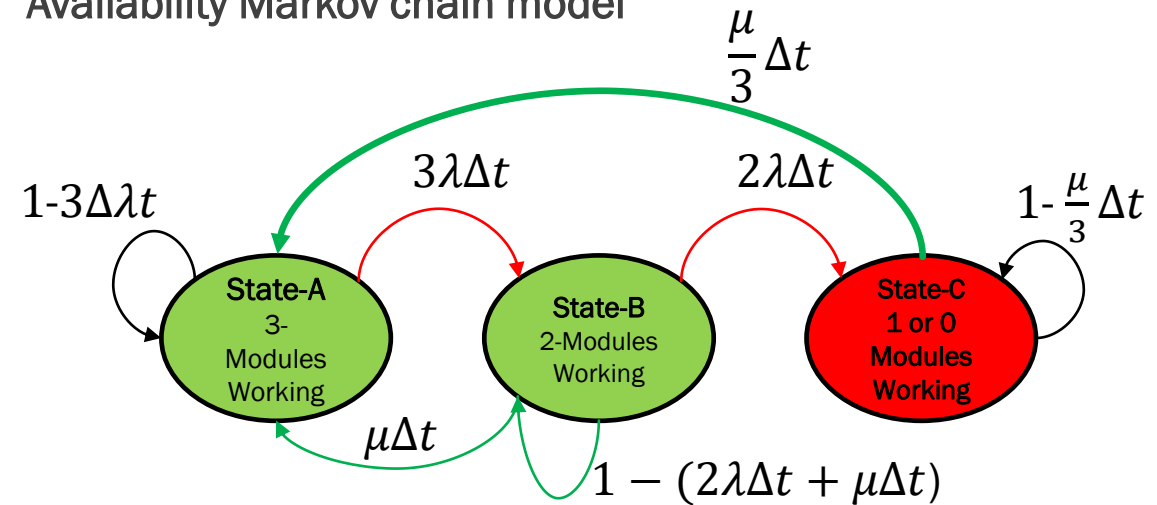


CONVERTING THE RELIABILITY MARKOV MODEL OF A REPAIRABLE TMR SYSTEM TO AN AVAILABILITY MARKOV MODEL

Reliability Markov chain model



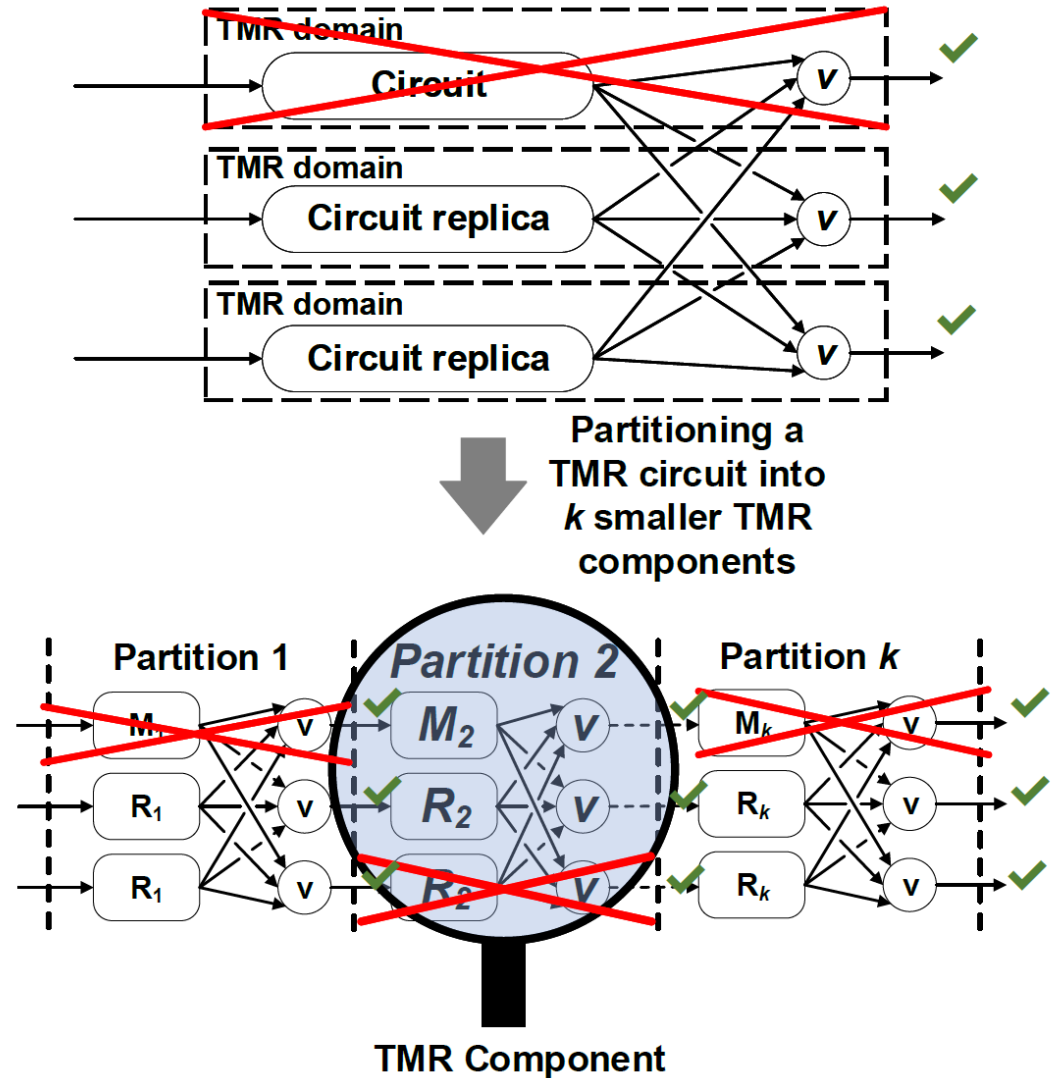
Availability Markov chain model



- The reliability of the TMR system is modeled by “trapping” the chain to State-C once it fails (see transition with rate 1) and summing the probability distribution of state-A and State-B
- The availability of the TMR system is modeled by adding a transition arrow from State-C to State-A with rate $\frac{\mu}{3}\Delta t$ to capture the fact that all three modules need to be recovered since in State-C one cannot know which modules are faulty. Similarly, the availability is found by summing the probability distribution of state-A and State-B

IMPROVING RELIABILITY THROUGH TMR PARTITIONING

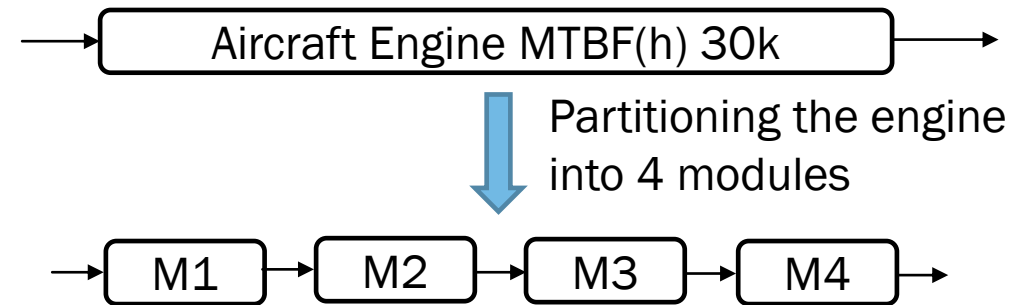
- TMR is a 2-out-of-3 redundancy scheme, which means that a TMR circuit can withstand faults in only one TMR domain at a time.
- However, if the same circuit is partitioned into k smaller TMR components it can then mask faults in k TMR domains, assuming that each partition (i.e., TMR component) has no more than one faulty TMR domain.
- The more partitions a TMR FPGA has, the less the likelihood of soft errors affecting the one TMR component and the higher the total reliability of the circuit.
- However, when k becomes very large the benefits of circuit partitioning are overwhelmed by the area and performance overheads of the added voters used between the TMR components.



SYSTEM PARTITIONING CAN ALSO REDUCE MAINTAINING COSTS

Example

- An aircraft engine has MTBF = 1K hours (scheduled and unscheduled)
- With a total annual flying rate of 30K hours and an average replacement cost of 10K €, the annual repair bill is $(30K/1K) * 10K € = 300000 €$
- The manufacturer redesigned the engine so it could be separated into four modules.
- What would be the new annual cost?



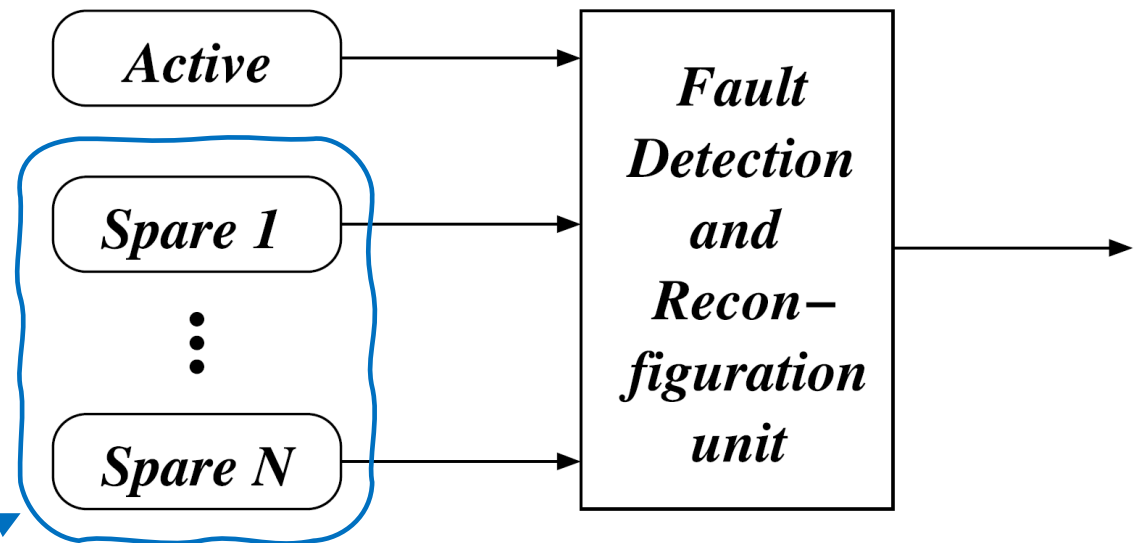
| Module | MTBF(h) | Replacement cost (€) | Replacements per year (30K hours) | Yearly cost (€) |
|----------|---------|----------------------|-----------------------------------|-----------------|
| M1 | 2.5K | 3.0K | $30K/2.5K=12$ | $12*3K =36K$ |
| M2 | 4.0K | 2.0K | $30K/4K=7.5$ | $7.5*2K=15K$ |
| M3 | 4.0K | 2.5K | $30K/4K=7.5$ | $7.5*2K=18.75K$ |
| M4 | 10.0K | 1.0K | $30K/10K=3$ | $3*10K=3K$ |
| TOTAL(€) | | | | 72750 |

COLD REDUNDANCY

- This best suited for non-critical systems where down-time and reliability is not a big concern. A cold redundant system consists of an ACTIVE module and N SPARE modules that are inactive (powered off). The fault-detector and reconfiguration unit (DRU) replaces any failed ACTIVE module with a SPARE module and powers it on (i.e., the SPARE becomes the ACTIVE module).

Inactive/not powered to conserve energy and mitigate aging effects

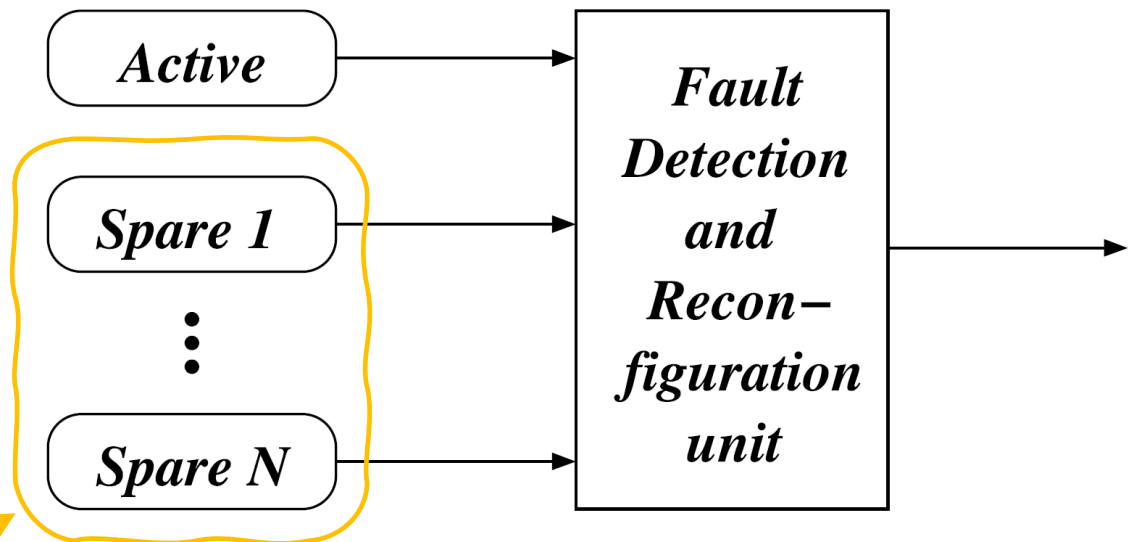
Both, fault detection and ACTIVE/SPARE swap can be slow, down-time can be tolerated.



WARM REDUNDANCY

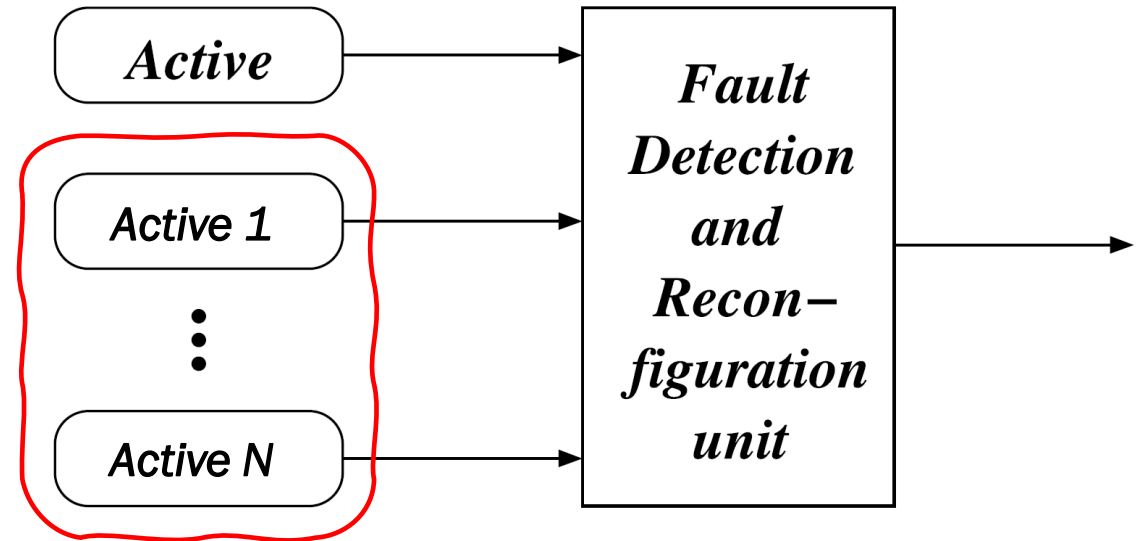
- This is suited to systems where time and response are important, but a momentary outage is still acceptable.

Spare modules are in standby mode and are periodically powered on to synchronize their state with the Active module

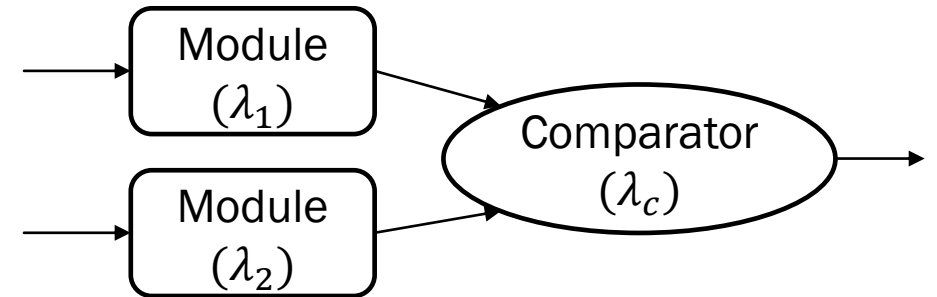


HOT REDUNDANCY

- Hot redundancy is the best solution for critical systems that cannot tolerate an outage for even a brief moment.
- Spare modules are powered on and their state is always synchronized with the state of the Active module.
- The NMR (e.g., TMR) systems that we examined follow the HOT redundancy methodology



DUPLEX OR DOUBLE MODULAR REDUNDANT (DMR) SYSTEMS



- In contrast with M-of-N (e.g., TMR) systems that detects, localises and mask faults, the Duplex systems are only used for error detection.
- The reliability and availability function of the Duplex system or Double Modular Redundant (DMR) system is similar with that of a simplex system assuming a perfect comparator (or $\lambda_1 + \lambda_c \gg \lambda_c$):
 $R(t) = e^{-(\lambda_1 + \lambda_2)t}$, and $R(t) = e^{-2\lambda t}$, for $\lambda_1 = \lambda_2$

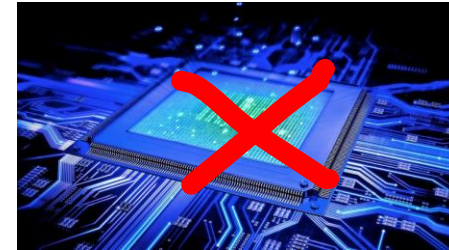
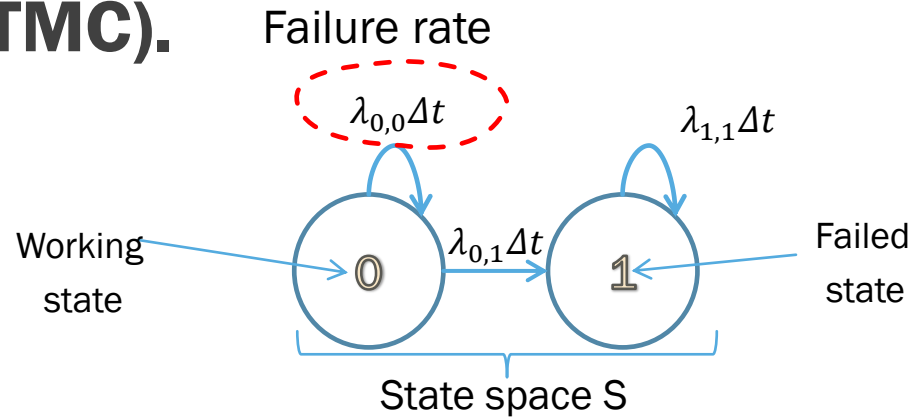
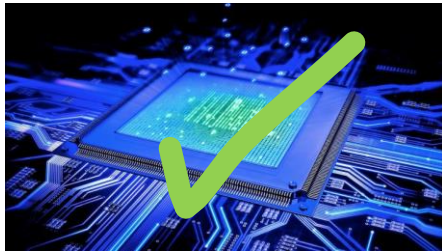
BIBLIOGRAPHY

- McMurtrey, Daniel, et al. "Estimating TMR reliability on FPGAs using Markov models." (2008).
- Koren, Israel, and C. Mani Krishna. *Fault-tolerant systems*. Morgan Kaufmann, 2020.
- Agiakatsikas, Dimitris. *High-level synthesis of triple modular redundant FPGA circuits with energy efficient error recovery mechanisms*. Diss. University of New South Wales, Sydney, Australia, 2019.
- Morozov, Andret. Reliability: Methodologies, Standards, and Tools, Presentation at Fraunhofer, 2013



BACKUP SLIDES

MATHEMATICAL DEPENDABILITY MODELS → CONTINUOUS-TIME MARKOV CHAINS (CTMC).



- A discrete state continuous time Markov chain is a stochastic process with infinite random variables $X(t)$ indexed by time t ($0 \leq t \leq 1$), which take states from a finite state space $S = \{0, 1, \dots, K\}$, where ($K > 0$).
- The main assumptions of a Markov chain are:
 - Given that a system just entered state i , it will remain for exponentially distributed time $e^{-\lambda t}$ in state i until it moves to the next state j with rate $\lambda_{i,j}\Delta t \in \mathbb{R}, \lambda_{i,j} > 0$ and constant over time.
 - **Memoryless:** The rate $r_{i,j}\Delta t$ of the system transitioning from state i to state j is independent of how it arrived in state i and independent of the time it will stay at state i .
- A Markov chain essentially consists of a set of transitions, which are determined by some probability distribution. Metrics such as reliability and availability are obtained by this probability distribution
- See <https://setosa.io/ev/markov-chains/> for visual/intuitive explanation of Markov-chain models