Digitalni sistemi otporni na otkaz

Predavanje V

Lecture Content

- Active redundancy
  - Duplication with comparison
  - Standby sparing
  - Pair-and-a-spare
- Hybrid redundancy
  - Self-purging redundancy
  - N-modular redundancy with spares
  - Triplex-duplex redundancy


entity test_shift is
generic (width : integer := 17);
port (clk : in std_ulogic;
reset : in std_ulogic;
load : in std_ulogic;
en : in std_ulogic;
outp : out std_ulogic);
end test_shift;
Hardware Fault Tolerance Techniques
Active Redundancy

Active Redundancy

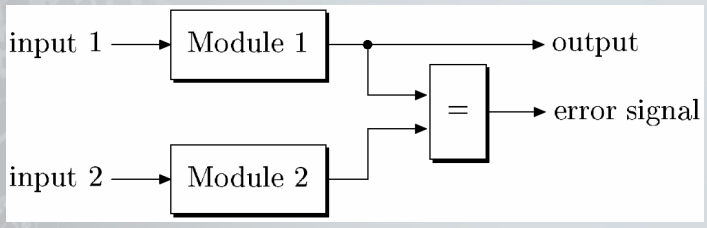
- Active redundancy achieves fault tolerance by first detecting the faults which occur and then performing actions needed to recover the system back to the operational state.
- Active redundancy techniques are common in applications where temporary erroneous results are preferable to the high degree of redundancy required to achieve fault masking.
- Infrequent, occasional errors are allowed, as long as the system recovers back to normal operation in a specified interval of time.
- In this section we consider three common active redundancy techniques:
  - duplication with comparison,
  - standby sparing and
  - pair-and-a-spare

Hardware Fault Tolerance Techniques

Duplication with Comparison

Duplication with Comparison

- The basic form of active redundancy is ***duplication with comparison*** shown in Figure below.
- Two identical modules perform the same computation in parallel.
- The results of the computation are compared using a comparator. If the results disagree, an error signal is generated.
- Depending on the application, the duplicated modules can be processors, memories, I/O units, etc.
- A duplication with comparison scheme can detect only one module fault.
- After the fault is detected, no actions are taken by the system to return back to the operational state.



Duplication with comparison

Reliability Evaluation I

- A duplication with comparison system functions correctly only until both modules operate correctly.
- When the first fault occurs, the comparator detects a disagreement and the normal functioning of the system stops, since the comparator is not capable to distinguish which of the results is the correct one.
- Assuming that the comparator is perfect and that the component failures are mutually independent, the reliability of the system is given by

$$R_{DC} = R_1 \cdot R_2$$

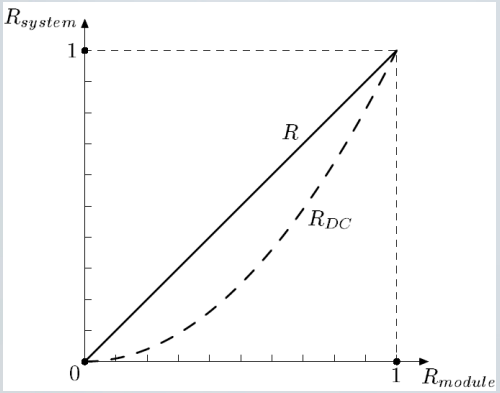
- or,

$$R_{DC} = R^2$$

- if $R_1 = R_2 = R$.

Reliability Evaluation II

- Figure on the right compares the reliability of a duplication with comparison system R_{DC} to the reliability of a simplex system consisting of a single module with reliability R.
- It can been seen that, unless the modules are perfect ($R(t) = 1$), the reliability of a duplication with comparison system is always smaller than the reliability of a simplex system.



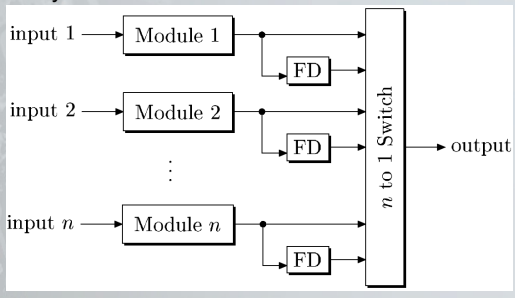
Duplication with comparison reliability compared to simplex system reliability

Hardware Fault Tolerance Techniques

Standby Sparing

Standby Sparing I

- ***Standby sparing*** is another scheme for active hardware redundancy. The basic configuration is shown in Figure below.
- Only one of *n* modules is operational and provides the system's output. The remaining *n*-1 modules serve as spares.
- A ***spare*** is a redundant component which is not needed for the normal system operation.
- A switch is a device that monitors the active module and switches operation to a spare if an error is reported by fault-detection unit FD.



Standby sparing redundancy

Standby Sparing II

- There are two types of standby sparing: hot standby and cold standby.
- In the ***hot standby sparing***, both operational and spare modules are powered up. The spares can be switched into use immediately after the operational module has failed.
- In the ***cold standby sparing***, the spare modules are powered down until needed to replace the faulty module.
- A disadvantage of cold standby sparing is that time is needed to apply power to a module, perform initialization and re-computation.
- An advantage is that the stand-by spares do not consume power. This is important in applications like satellite systems, where power consumption is critical.
- Hot standby sparing is preferable where the momentary interruption of normal operation due to reconfiguration needs to be minimized, like in a nuclear plant control system.

Standby Sparing III

- A standby sparing system with *n* modules can tolerate *n* -1 module faults.Here by “tolerate” we mean that the system will detect and locate the faults, successfully recover from them and continue delivering the correct service.
- When the *n*th fault occurs, it will still be detected, but the system will not be able to recover back to normal operation.

- The standby sparing redundancy technique is used in many systems. One example is the Apollo spacecraft’s telescope mount pointing computer. In this system, two identical computers, an active and a spare, are connected to a switching device that monitors the active computer and switches operation to the backup in case of a malfunction.

- Another example of using standby sparing is Saturn 5 launch vehicle digital computer (LVDC) memory section. The section consists of two memory blocks, with each memory being controlled by an independent buffer register and parity-checked.
- Initially, only one buffer register output is used. When a parity error is detected in the memory being used, operation immediately transfers to the other memory. Both memories are then re-generated by the buffer register of the "correct" memory, thus correcting possible transient faults.

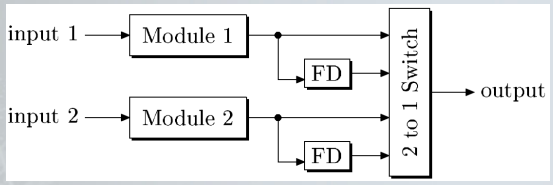
- Standby sparing is also used in Compaq’s NonStop Himalaya server. The system is composed of a cluster of processors working in parallel. Each processor has its own memory and copy of the operating system. A primary process and a backup process are run on separate processors. The backup process mirrors all the information in the primary process and is able to instantly take over in case of a primary processor failure.

Reliability Evaluation

- By their nature, standby systems involve dependency between components, since the spare units are held in reserve and only brought to operation in the event the primary unit fails.
- Therefore, standby systems are best analyzed using Markov models.
- We first consider an idealized case when the switching mechanism is perfect.
- We also assume that the spare cannot fail while it is in the standby mode.
- Later, we consider the possibility of failure during switching.

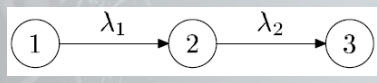
Perfect Switching Case I

- Consider a standby sparing scheme with one spare, shown in Figure below.



Standby sparing system with one spare

- Let module 1 be a primary module and module 2 be a spare.
- The state transition diagram of the system is shown in Figure below. The states are numbered according to the Table shown below.



State transition diagram of a standby sparing system with one spare

Component 1	Component 2	State Number
O	O	1
F	O	2
F	F	3

Markov states of the state transition diagram of a standby sparing system with one spare

Perfect Switching Case II

- When the primary component fails, there is a transition between state 1 and state 2.
- If a system is in state 2 and the spare fails, there is a transition to state 3.

- Since we assumed that the spare cannot fail while in standby mode, the combination (*O*, *F*) cannot occur.
- The states 1 and 2 are operational states. The state 3 is the failed state.

- The transition matrix for the state transition diagram shown on the previous Slide is given by

$$\mathbf{M} = \begin{bmatrix} -\lambda_1 & 0 & 0 \\ \lambda_1 & -\lambda_2 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix}$$

- So, we get the following system of state transition equations

$$\frac{d}{dt}\begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} = \begin{bmatrix} -\lambda_1 & 0 & 0 \\ \lambda_1 & -\lambda_2 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix}$$

Perfect Switching Case III

- By solving the system of equations, we get

$$P_1(t) = e^{-\lambda_1 t}$$
$$P_2(t) = \frac{\lambda_1}{\lambda_2 - \lambda_1}(e^{-\lambda_1 t} - e^{-\lambda_2 t})$$
$$P_3(t) = 1 - \frac{1}{\lambda_2 - \lambda_1}(\lambda_2 e^{-\lambda_1 t} - \lambda_1 e^{-\lambda_2 t})$$

- Since $P_3(t)$ is the only state corresponding to system failure, the reliability of the system is the sum of $P_1(t)$ and $P_2(t)$

$$R_{SS}(t) = e^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_2 - \lambda_1}(e^{-\lambda_1 t} - e^{-\lambda_2 t})$$

- This can be re-written as

$$R_{SS}(t) = e^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_1 t}(1 - e^{-(\lambda_2 - \lambda_1)t})$$

- Assuming $(\lambda_2 - \lambda_1)t << 1$, we can expand the term $e^{-(\lambda 2 - \lambda 1)t}$ as a power series of $-(\lambda_2 - \lambda_1)t$ as

$$e^{-(\lambda_2 - \lambda_1)t} = 1 - (\lambda_2 - \lambda_1)t + 1/2(\lambda_2 - \lambda_1)^2 t^2 - \ldots$$

- Substituting it in the expression for the $R_{ss}(t)$, we get

$$R_{SS}(t) = e^{-\lambda_1 t} + \lambda_1 e^{-\lambda_1 t}(t - 1/2(\lambda_1 - \lambda_2)t^2 + \ldots)$$

- Assuming $\lambda_2 = \lambda_1$, the above can be simplified to

$$R_{SS}(t) = (1 + \lambda t)e^{-\lambda t}$$

Perfect Switching Case IV

- Next, let us see how the last equation from previous slide would change if we would ignore the dependency between the failures.
- If the primary and spare module failures are treated as mutually independent, the reliability of a standby sparing system is a sum of two probabilities:
  1. The probability that module 1 operates correctly, and
  2. The probability that module 2 operates correctly, while module 1 has failed and has been replaced by module 2.
- Then, we get the following expression:

$$R_{SS} = R_1 + (1 - R_1)R_2$$

- If $R_1 = R_2 = R$, then

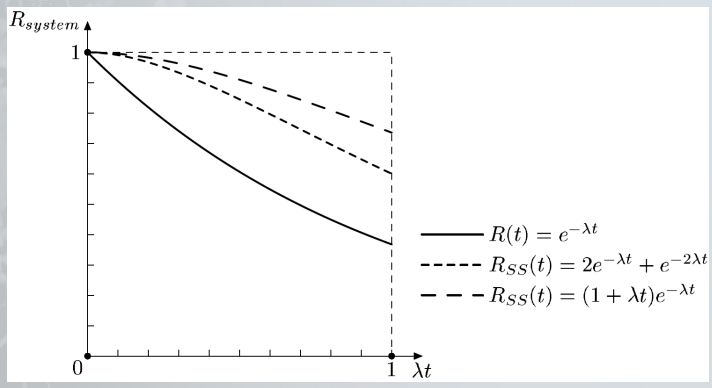
$$R_{SS} = 2R - R^2$$

- or

$$R_{SS}(t) = 2e^{-\lambda t} - e^{-2\lambda t}$$

Perfect Switching Case V

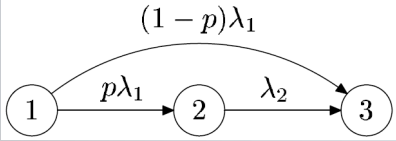
- Figure below compares the plots of the reliabilities for dependent and independent case.
- One can see that neglecting the dependencies between failures leads to underestimating the standby sparing system reliability.



Standby sparing reliability compared to simplex system reliability

Non-perfect Switching Case I

- Next, we consider the case when the switch is not perfect.
- Suppose that the probability that the switch successfully replaces the primary unit by a spare is *p*.
- Then, the probability that the switch fails to do it is 1 - *p*.
- The state transition diagram with these assumptions is shown in Figure below.
- The transition from state 1 is partitioned into two transitions.
- The failure rate is multiplied by *p* to get the rate of successful transition to state 2.
- The failure rate is multiplied by 1 - *p* to get the rate of the switch failure.



State transition diagram of a standby system with one spare

Non-perfect Switching Case II

- The state transition equations corresponding to the state transition diagram shown on the previous slide are

$$\begin{cases} \frac{d}{dt}P_1(t) = -\lambda_1 P_1(t) \\ \frac{d}{dt}P_2(t) = p\lambda_1 P_1(t) - \lambda_2 P_2(t) \\ \frac{d}{dt}P_3(t) = \lambda_2 P_2(t) + (1-p)\lambda_1 P_1(t) \end{cases}$$

- By solving this system of equations, we get

$$P_1(t) = e^{-\lambda_1 t}$$
$$P_2(t) = \frac{p\lambda_1}{\lambda_2 - \lambda_1}(e^{-\lambda_1 t} - e^{-\lambda_2 t})$$
$$P_3(t) = 1 - (1 + \frac{p\lambda_1}{\lambda_2 - \lambda_1})e^{-\lambda_1 t} + \frac{p\lambda_1}{\lambda_2 - \lambda_1}e^{-\lambda_2 t}$$

- As before, $P_3(t)$ corresponds to system failure. So, the reliability of the system is the sum of $P_1(t)$ and $P_2(t)$

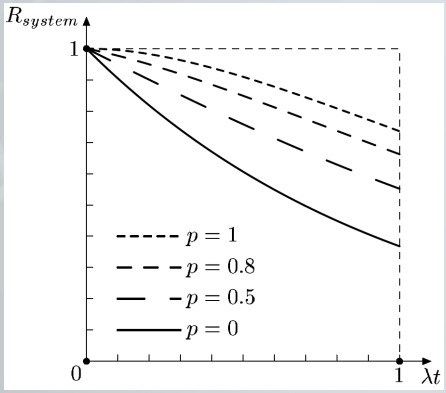
$$R_{SS}(t) = e^{-\lambda_1 t} + \frac{p\lambda_1}{\lambda_2 - \lambda_1}(e^{-\lambda_1 t} - e^{-\lambda_2 t})$$

- Assuming $\lambda_2 = \lambda_1$, the above can be simplified to

$$R_{SS}(t) = (1 + p\lambda t)e^{-\lambda t}$$

Non-perfect Switching Case III

- Figure below compares the reliability of a standby sparing system for different values of *p*.
- As *p* decreases, the reliability of the standby sparing system decreases.
- When *p* reaches zero, the standby sparing system reliability reduces to the reliability of a simplex system.

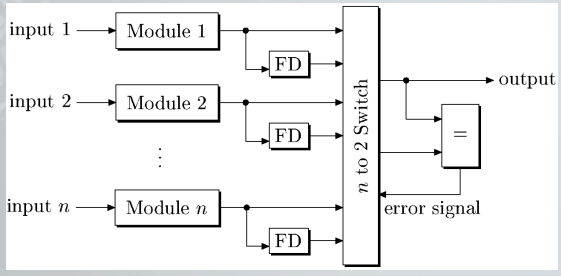


Reliability of a standby sparing system for different values of *p*


entity test_shift is
generic (width : integer := 17);
port (clk : in std_ulogic;
reset : in std_ulogic;
load : in std_ulogic;
en : in std_ulogic;
outp : out std_ulogic);
end test_shift;
Hardware Fault Tolerance Techniques
Pair-and-a-Spare

Pair-and-a-Spare I

- Pair-and-a-spare technique combines standby sparing and duplication and comparison approaches (see Figure below).
- The idea is similar to standby sparing, however two modules instead of one are operated in parallel.
- As in the duplication with comparison case, the results are compared to detect disagreement.
- If an error signal is received from the comparator, the switch analyzes the report from the fault detection block and decides which of the two modules' output is faulty.
- The faulty module is removed from operation and replaced with a spare module.



Pair-and-a-spare redundancy

Pair-and-a-Spare II

- A pair-and-a-spare system with *n* modules can tolerate *n* - 1 module faults.
- When the *n* - 1th fault occurs, it will be detected and located by the switch and the correct result will be passed to the system's output.
- However, since there will be no more spares available, the switch will not be able to replace the faulty module with a spare module.
- The system's configuration will be reduced to a simplex system with one module. So, the *n*th fault will not be detected.


entity test_shift is
generic (width : integer := 17);
port (clk : in std_ulogic;
reset : in std_ulogic;
load : in std_ulogic;
en : in std_ulogic;
outp : out std_ulogic);
end test_shift;
Hardware Fault Tolerance Techniques
Hybrid Redundancy

Hybrid Redundancy

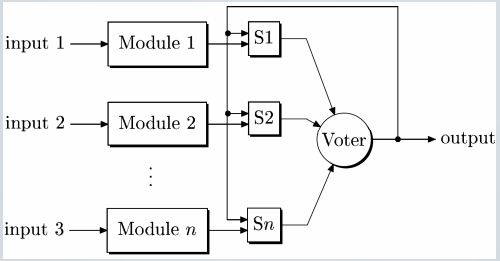
- The main idea of hybrid redundancy is to combine the attractive features of passive and active approach.
- Fault masking is used to prevent system from producing momentary erroneous results.
- Fault detection, location and recovery are used to reconfigure the system after a fault occurs.
- In this section, we consider three basic techniques for hybrid redundancy:
  - self-purging redundancy,
  - N-modular redundancy with spares and
  - triplex-duplex redundancy.

Hardware Fault Tolerance Techniques

Self-Purging Redundancy

Self-Purging Redundancy

- Self-purging redundancy consists of *n* identical modules which are actively participating in voting.
- The output of the voter is compared to the outputs of individual modules to detect disagreement.
- If a disagreement occurs, the switch opens and removes, or *purges*, the faulty module from the system.
- The voter is designed as a threshold gate, capable to adapt to the changing number of inputs.
- The input of the removed module is forced to zero and therefore do not contribute to the voting.
- A self-purging redundancy system with *n* modules can mask *n* - 2 module faults.
- When *n* - 2 modules are purged and only two are left, the system will be able to detect the next, *n*-1 th fault, but, as in the duplication with comparison case, the voter will not be able to distinguish which one of the two results is the correct one.



Self-purging redundancy

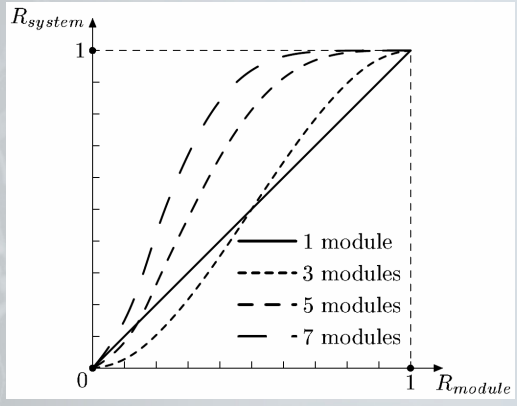
Reliability Evaluation I

- Since all the modules of the system operate in parallel, we can assume that the modules' failures are mutually independent.
- It is sufficient that two of the modules of the system function correctly for the system to be operational.
- If the voter and the switches are perfect, and if all the modules have the same reliability $R_1 = R_2 = \ldots = R_n = R$, then the system is *not* reliable if all the modules have failed (probability $(1 - R)^n$), or if all but one modules have failed (probability $R(1 - R)^{n-1}$).
- Since there are n choices for one of n modules to remain operational, we get the equation

$$R_{SP} = 1 - ((1 - R)^n + nR(1 - R)^{n-1})$$

Reliability Evaluation II

- Figure below compares the reliabilities of self-purging redundancy systems with three, five and seven modules.



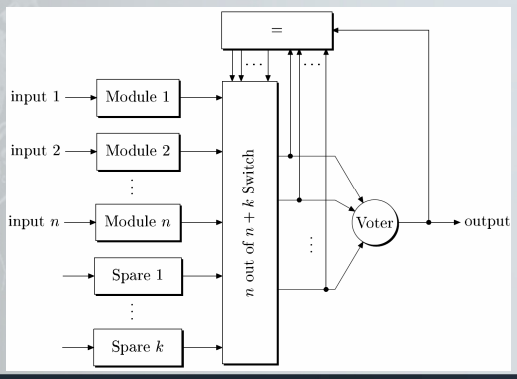
Reliability of a self-purging redundancy system with 3, 5 and 7 modules

Hardware Fault Tolerance Techniques

N-modular Redundancy with Spares

N-modular Redundancy with Spares I

- N-modular redundancy with *k* spares is similar to self-purging redundancy with *k* + *n* modules, except that only *n* modules provide input to a majority voter (see Figure below). Additional *k* modules serve as spares.
- If one of the primary modules becomes faulty, the voter will mask the erroneous result and the switch will replace the faulty module with a spare one.
- Various techniques are used to identify faulty modules. One approach is to compare the output of the voter with the individual outputs of the modules, as shown in Figure below. A module which disagrees with the majority is declared faulty.



N-modular redundancy with spares

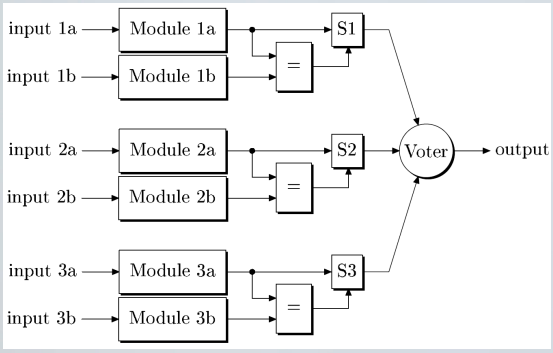
N-modular Redundancy with Spares II

- The fault-tolerant capabilities of an N-modular redundancy system with *k* spares depend on the form of voting used as well as the implementation of the switch and comparator.
- One possibility is that, after the spares are exhausted, the disagreement detector is switched off and the system continues working as a passive NMR system.
- Then, such a system can mask $\lfloor n/2 \rfloor + k$ faults, i.e. the number of faults a NMR system can mask plus the number of spares.
- Another possibility is that the disagreement detector remains on, but the voter is designed to be capable to adjust to the decreasing number of inputs.
- In this case, the behavior of the system is similar to the behavior of a self-purging system with *n* + *k* modules, i.e. up to *k* + *n* - 2 module faults can be masked.
- Suppose the spares are exhausted after the first *k* faults, and the *k* + 1th fault occurred.
- As before, the erroneous result will be masked by the voter, the output of the voter will be compared to the individual outputs of the modules, and the faulty will be removed from considerations.
- A difference is that it will not be replaced with a spare one, but instead the system will continue working as *n* - 1-modular system. Then a *k* + *i*th fault occurs, the voter votes on (*n* - *i)* modules.

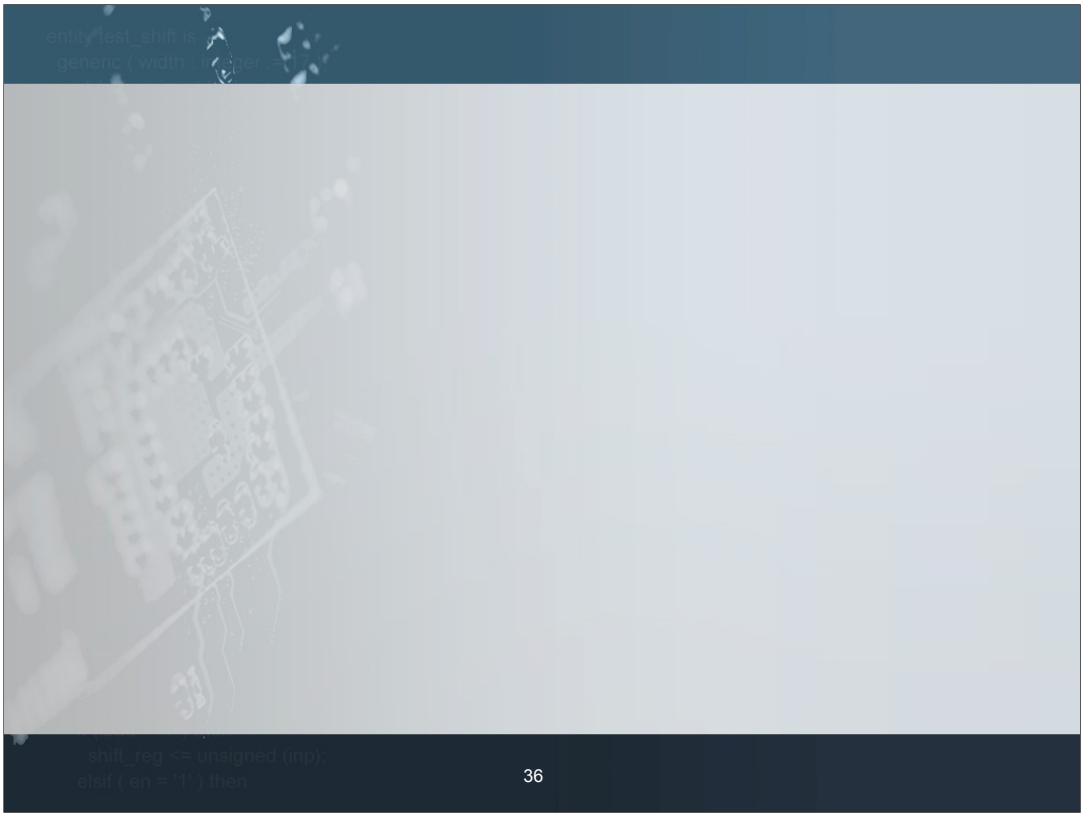

entity test_shift is
generic (width : integer := 17);
port (clk : in std_ulogic;
reset : in std_ulogic;
load : in std_ulogic;
en : in std_ulogic;
outp : out std_ulogic);
end test_shift;
Hardware Fault
Tolerance Techniques
Triplex-Duplex Redundancy

Triplex-Duplex Redundancy

- Triplex-duplex redundancy combines triple modular redundancy and duplication with comparison.
- A total of six identical modules, grouped in three pairs, are computing in parallel.
- In each pair, the results of the computation are compared using a comparator.
- If the results agree, the output of the comparator participates in the voting.
- Otherwise, the pair of modules is declared faulty and the switch removes the pair from the system.
- In this way, only faulty-free pair participates in voting.



Triplex-duplex redundancy


entity test_shift is
generic (width : integer :=
shift_reg <= unsigned (inp);
elsif (en = '1') then
36