ECE454/544: Fault-Tolerant Computing & Reliability Engineering

Lecture #3 –

Hardware Redundancy Techniques

Instructor: Dr. Liudong Xing

Administrative Issues (9/14, Wednesday)

- Homework#1
  - Assigned today; please download the problems from the course website
  - https://xingteaching.sites.umassd.edu/
  - Due **Sept. 21, Wednesday**
- Project teams
  - Due **Sept. 14, Wednesday**
- Project Proposal
  - Due **Oct. 5, Wednesday**
  - Refer to Proposal Guideline on the course website

Review of Lecture #2

- Faults, Errors, and Failures
  - Cause-and-effect relationship
  - Three universe model: physical, information, external
- Causes of Faults
  - Specification mistakes, implementation mistakes, component defects, external disturbances
- Characteristics of Faults
  - Nature, duration, extent, value
- Design Philosophies to Combat Faults
  - Fault avoidance, fault masking, fault tolerance

Concept of Redundancy (Revisit)

- Redundancy: the addition of information, resources or time beyond what is needed for normal system operation, to detect and possibly tolerate fault
  - Hardware redundancy
  - Information redundancy
  - Time redundancy
  - Software redundancy

Fault tolerance requires the use of one or more forms of the basic redundancy types

Learning Objectives

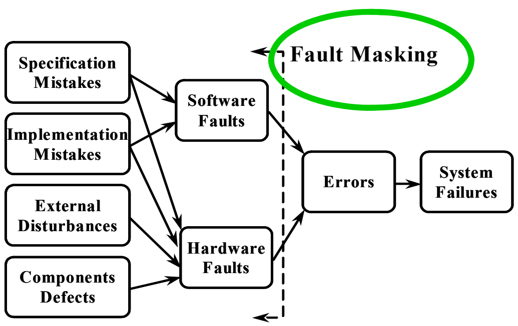
- Describe different types of hardware redundancy techniques for achieving fault tolerance
- Understand the difference between fault masking and fault tolerance

Hardware Redundancy

- Addition of extra hardware, for the purpose of either detecting or tolerating faults
- Three basic types
  - **Passive**
  - Active/dynamic
  - Hybrid

Passive Hardware Redundancy (PHR)

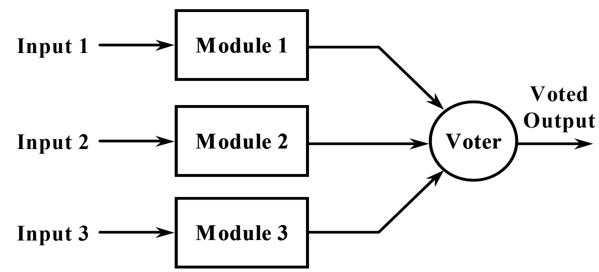
- PHR uses fault masking to hide the occurrence of faults rather than detect them, and prevents the faults from resulting in errors and failures



- PHR relies on **majority voting** mechanisms to mask the occurrence of faults

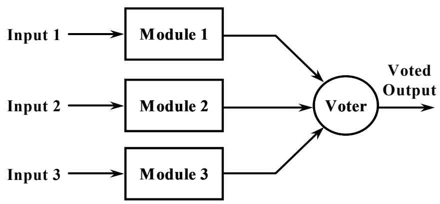
Triple Modular Redundancy (TMR)

- TMR uses three identical modules, performing identical operations, with a majority voter determining the output
- Replicated modules: processors, memories, or any hardware entities.



TMR can be applied to software too!

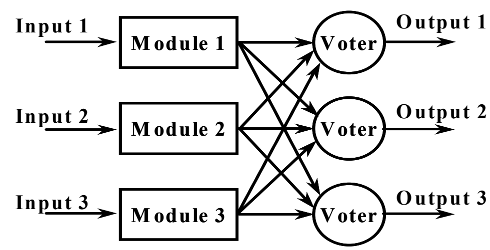
Reliability of TMR



- Reliability of each module: *p*
- Reliability of the voter: *w*
- Reliability of TMR?

TMR (Cont'd)

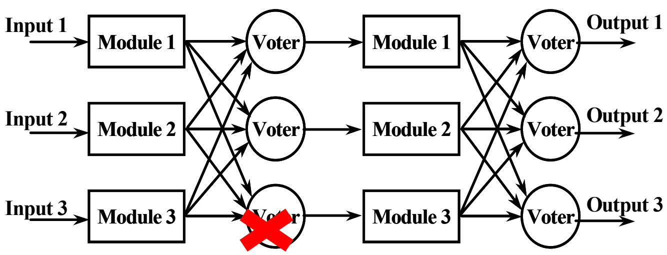
- The voter is a single-point of failure
  - Any single component within a system whose failure leads to the system failure
- Triplicated voters can overcome the effects of voter failure
  - Called a "restoring organ"



The voter is no longer a single-point of failure!

Multi-Stage Triplicated TMR

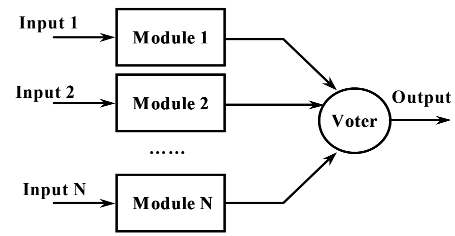
- Several stages of triplicated TMR can be interconnected so that errors are corrected before being passed to a subsequent module



- If a voter fails in one stage, the subsequent stage sees the failure as one input becoming corrupted. Voting at the output of the stage that gets the erroneous input corrects the erroneous result

N-Modular Redundancy (NMR)

- A generalization of TMR: uses *N* modules as opposed to three



- *N* is an ODD number so that a majority voting arrangement can be used
- More module faults can be tolerated

  To tolerate 2 faults, *N*=?

- Primary tradeoff is the fault tolerance achieved vs. the hardware required (power, weight, cost, size limitations)

Reliability of NMR

- Reliability of each module: *p*
- Reliability of the voter: *w*
- $N = 2n+1$
- Reliability of NMR?

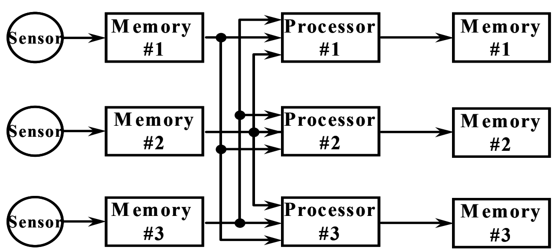
Voting Techniques in NMR

- Hardware voting & software voting
- **Hardware voting** uses a hardware voter
  - Logic gates, using digital logic design technique
- **Exercise:** design a 1-bit TMR voter that produces an output of 1 when at least 2 out of 3 inputs are 1
  - Truth table
  - Karnaugh map
  - Logic function for the voter
  - Implementation: circuit

An 8-bit or 16-bit majority voter can be constructed using 8 or 16 of the above circuits

Software Voting

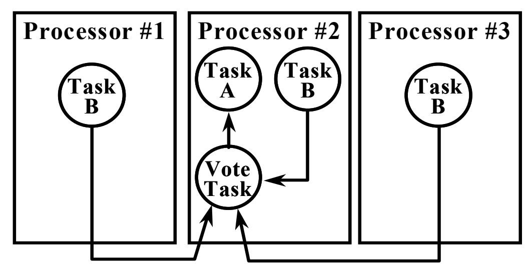
- A mechanism must be available to provide the software routine with the data on which to vote
- **Example I:** each processor performs a majority vote on three inputs to determine the appropriate value to use in calculation



A microprocessor system using software voting

Software Voting (Cont'd)

- **Example II**
  - Task B is executed on three separate processors.
  - Point-to-point links between processors to share data.
  - Results of task B are voted upon in processor #2 before being used as input to task A.



Hardware vs. Software Voting

- Hardware voting
  - Using a dedicated hardware voter → fast!
  - The hardware required for the voter increases the system's power consumption, weight, and size
- Software voting
  - A software voter performs the voting process within a minimum amount of additional hardware, by taking advantage of a processor's computational capabilities
  - By simply modifying the software, the software voter can modify the manner in which the voting is performed
  - The voting process requires more time!

Voting Techniques Selection

- The decision to use HW or SW voting depends on
  - Availability of a processor to perform the voting
  - Speed at which voting must be performed
  - Criticality of space, power, and weight limitations
  - Number of different voters that must be provided
  - Flexibility required of the voter *w.r.t.* future changes in the system

Problem in Voting

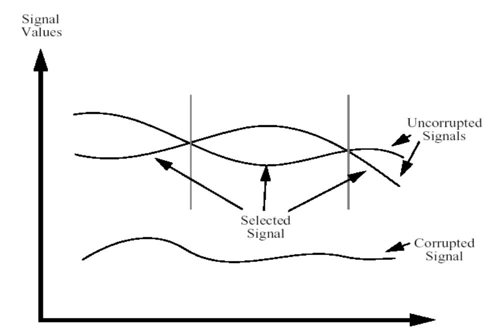
- In practical application of voting, three results in a TMR system may not completely agree even in a fault-free environment → the majority voter may find no two results agree exactly!

- **Solutions:**
  - Mid-value select technique
  - Voting on *k* msb of the data

msb: most significant bit
lsb: least significant bit

Solution (1): Mid-Value Select Technique

- Chooses a value from the three available in a TMR by selecting the value that lies between the remaining two
- Can be applied to any systems with an odd number of modules



Solution (2): Voting on Part of Data

- Often used when quantities never exactly agree and acceptable disagreement will occur only in the *lsb*
  - An AD converter can produce quantities that disagree in the *lsb*, even if the exact signal is passed through the same converter multiple times.
- Ignore the *lsb*; performing a majority vote only on the *k msb* of the data
- Number of bits ignored depends on the application; a function of the accuracy of components being used

Agenda

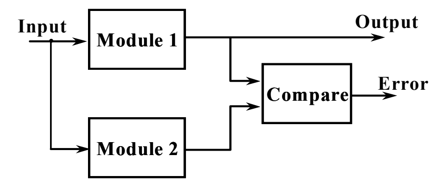
- Hardware Redundancy
  - √ Passive redundancy
    - Basic concept, TMR & multi-stage triplicated TMR, NMR
    - Hardware and software voting techniques
    - Voting problem and solutions
  - **Active redundancy**
  - Hybrid redundancy

Active Hardware Redundancy (AHR)

- Attempt to achieve fault tolerance by fault/error detection, location, and recovery
- Not attempt to prevent faults from producing errors within the system
- **Common examples**
  I. Duplication with comparison
  II. Standby sparing

Example I: Duplication with Comparison (DWC)

- Basic idea: to develop two identical pieces of HW modules performing the same computations in parallel, in the event of disagreement, an error message is generated



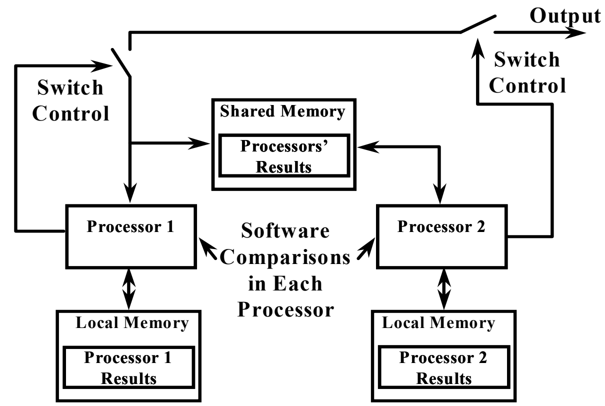
- DWC can only detect faults, not tolerate them → used as fundamental fault detection technique in AHR
- Inefficient use of hardware (>100% redundancy)
- Efficient use of time

Problem of DWC

- The comparator can fail such that
  - Faults in duplicated modules are never detected
  - An error indication is caused when no error exists
- Approach: duplicate the comparison process

Enhanced DWC

- **Example:** to implement the comparison process in software that executes in each of the two microprocessors



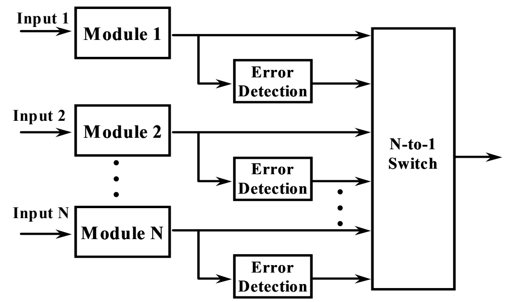
Both processors must agree that results match before an output is produced!

Agenda

- Hardware Redundancy
  - ✓ Passive redundancy
    - Basic concept, TMR & multi-stage triplicated TMR, NMR
    - Hardware and software voting techniques
    - Mid-value select technique; voting on part of data
  - ✓ Active redundancy
    - Duplication with comparison
    - **Standby sparing: hot, cold, warm**
  - – Hybrid redundancy

Example II: Standby Sparing

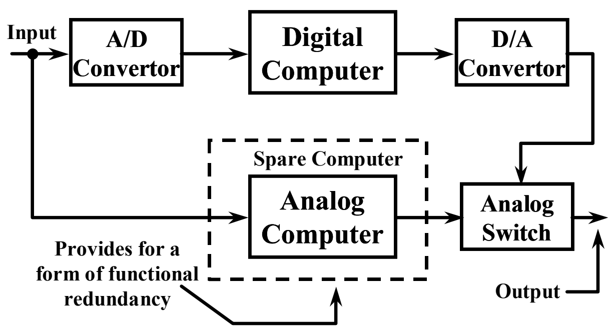
- Also called "standby replacement"



- – One module is operational and others serve as standbys or spares.
- – Error location & detection techniques identify faulty modules so that a fault-free module is always selected to provide the system's output
- – The switch examines error reports from error detection circuitry associated with each module to decide which module's output to use

https://en.wikipedia.org/wiki/Fault_detection_and_isolation

Application -- X-29 Flight Control System



http://en.wikipedia.org/wiki/Analog_computer

Sparing Approaches

- Standby sparing can bring a system back to a full operational capability after a fault occurs
- But it requires different levels of disruption in system operation

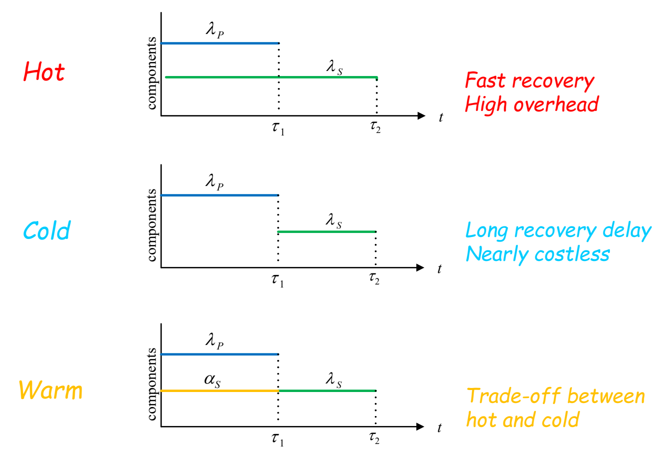
- Types
  - Hot standby sparing
  - Cold standby sparing
  - Warm standby sparing

Sparing Approaches (Cont'd)

- Hot standby sparing -- spares remain powered at all times to perform operations and to minimize the reconfiguration and recovery times following a fault
  - Example: a process control system that controls a chemical reaction
- Cold standby sparing -- spares remain unpowered until needed in the reconfiguration and recovery processes
  - Long time required to apply power and perform initialization prior to bringing the module into active service
  - Example: satellite applications where power consumption is critical
- Warm standby sparing – a trade-off between cold and hot
  - Example: players waiting outside the field while play is going on

Sparing Approaches (Cont'd)

- Dynamic failure rate behavior of the standby sparing system
- λ_P: failure rate of primary unit
- λ_S, α_S: failure rate of spare unit



Agenda

- Hardware Redundancy
  - ✓ Passive redundancy
    - Basic concept, TMR & multi-stage triplicated TMR, NMR
    - Hardware and software voting techniques
    - Mid-value select technique
  - ✓ Active redundancy
    - Duplication with comparison
    - Standby sparing: hot, cold, warm
  - – **Hybrid redundancy**

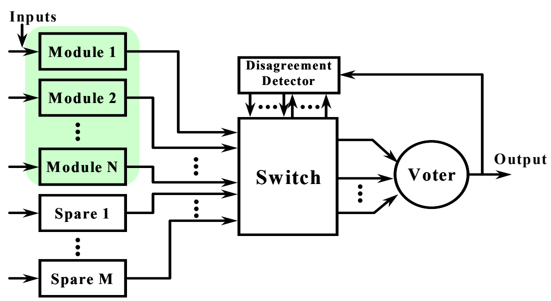
Hybrid Hardware Redundancy

- Combine attractive features of both active and passive techniques
  - – uses passive redundancy to prevent errors, but also uses active redundancy to provide enhanced fault tolerance
  - – requires enough hardware to use voting and for spares

- **Example approaches**
  - I. N-Modular Redundancy (NMR) with Spares
  - II. Self-Purging Redundancy

Example I: NMR with Spares

- Combines **NMR** and **standby sparing**
- To provide a basic core of *N* modules arranged in a voting configuration, spares are provided to replace faulty modules in the NMR core



  - The system remains in the basic NMR configuration until disagreement detector determines the existence of a faulty unit
  - Fault detection: compare output of the voter with individual outputs of the modules. A module that disagree with the majority output is labeled as faulty and removed from NMR core
  - A spare unit is switched in to replace the faulty module

NMR with Spares (Cont'd)

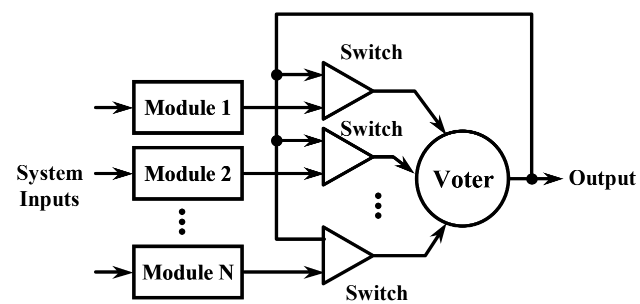
- How many module faults can be tolerated using a TMR with one spare design (4 modules)?
- To tolerate two faults, how many modules must be configured in a passive fault masking configuration?

NMR with Spares (Cont'd)

- Advantages
  - Can accomplish the same results using **fewer** hardware modules than passive approaches, but with fault detection/location/recovery schemes
  - The voting configuration (core NMR) can be restored after a fault has occurred
  - Reliability of the core NMR system is maintained as long as the pool of spares is not exhausted

Example II: Self-Purging Redundancy

- Each module is designed with capability to remove itself from the system in the event that its output disagrees with the voted output



  - **Switch:** to remove/purge its associated module from the system when the module fails
  - **Voter:** to produce the system output and provide masking of any fault that occur

Summary of Lecture #3 (1)

- **Passive** redundancy uses fault masking to hide the occurrence of faults and prevent the faults from resulting in errors and failures
  - **TMR** is the most common form of passive hardware redundancy, triplicated TMR can overcome the effects of the single-point of failure (voter)
  - Hardware and software voting have their pros and cons, the decision must be made based on several factors
  - Mid-value select technique and voting on part of data technique can be used to alleviate the problem of disagreeing results in a NMR system (N is an odd number)
- **Active** redundancy uses detection, location, and recovery techniques (reconfiguration)
  - **Duplication with comparison** can only detect faults, not tolerate them
  - **Hot standby sparing** can minimize the disruption in performance but consume more power than **cold standby sparing**

Summary of Lecture #3 (2)

- **Hybrid** redundancy employs both fault masking and reconfiguration
  - **NMR with spare** technique can accomplish the same results using fewer hardware modules than passive approaches, but with fault detection/location/recovery schemes
  - **Self-purging redundancy** technique uses the system output to remove modules whose output disagrees with the system output

Things to DO

- Homework#1
  - Due **Sept. 21, Wed.**
- Project Proposal
  - Due **Oct. 5, Wed.**

Next topic:
Information Redundancy Techniques!