ECE454/544: Fault-Tolerant Computing & Reliability Engineering

Lecture #2 –

Fault, Error, and Failure

Instructor: Dr. Liudong Xing

Administrative Issues (9/12, Monday)

- Today is the **Last day to Add/Drop**
- Project teams
  - Due **September 14, Wednesday**

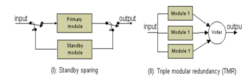
Review of Lecture #1

In the first lecture, we covered the

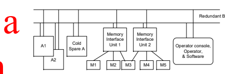
- Course syllabus & operational details
- Basic concepts of FTC & RE
  - Fault-tolerant systems, fault-tolerance, fault-tolerant computing, fault avoidance
  - Reliability, availability, safety, maintainability, testability, performability, & graceful degradation, and dependability
- Applications of fault tolerance
  - Long-life applications; Critical computation applications; High availability applications
- Significance of fault tolerance and reliability analysis via five examples

Review of Lecture #1 (Cont'd) - Examples

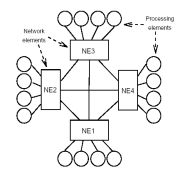
1) Comparing alternative fault-tolerant designs (TMR vs Standby Sparing)



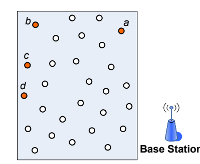
2) Evaluate the reliability of a complex computer system



3) Comparing alternative configurations for a fault-tolerant parallel processing system

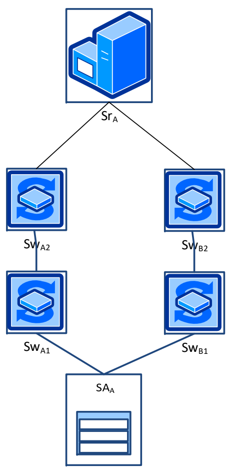


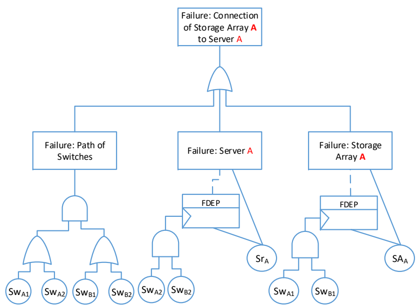
4) Topology design for wireless sensor networks (Mesh vs. Hierarchical Clustering vs. Tree)



5) **Reliability vs. Cost**

Example (5): revisit

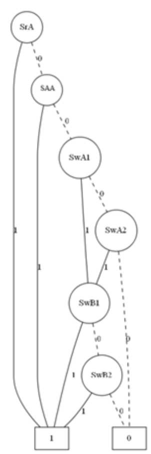




Component	Reliability
Switch	99.999%
Storage Array	99.999%
Server	97.81 %

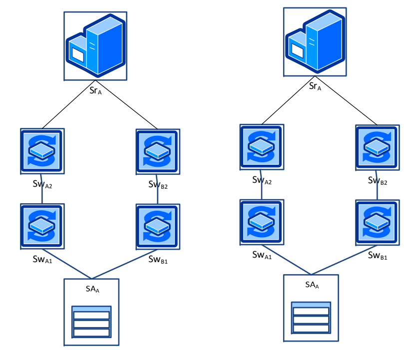
0.0219097
(~ 8 days/year)

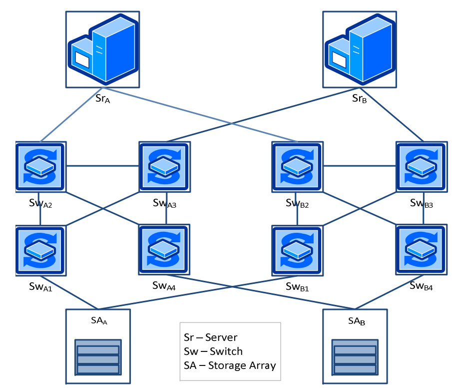
Sr – Server
Sw – Switch
SA – Storage Array



With doubling budget, improve the design with down time < 5 hours per year

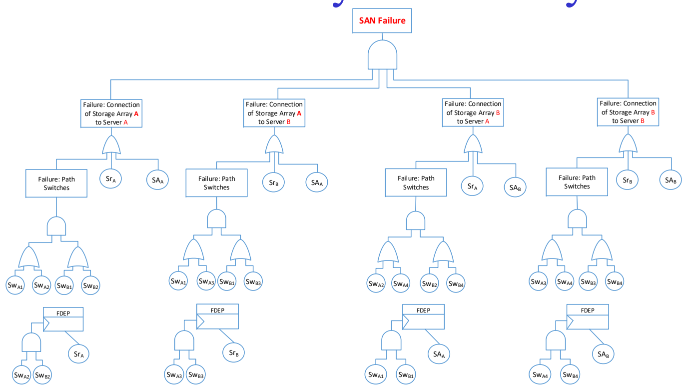
With Doubled Budget

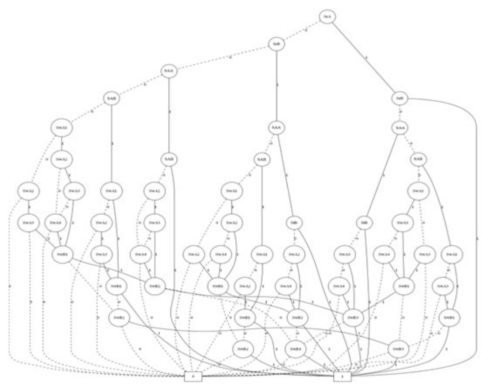




6

SAN: Reliability-Cost Analysis





0.00047961 (~ 4.2hours/year) after doubling budget

vs. 0.0219097 (~ 8 days/year)

Effects of Link Failures

Link Failure Probability	Reliability	MDT Per Year (hrs)
0	0.999520389895	4.20138451
0.0000011	**0.999520389894**	**4.20138452**
0.000011	0.999520389881	4.20138464
0.00011	0.999520389282	4.20138988
0.0002	**0.999520388009**	**4.20140104**
0.0011	0.999520336967	4.20184816
0.011	0.999515023235	4.24839645
0.1	0.998603856838	12.230214

- Practical range for link unreliability: [1.1e-6, 200e-6]
- Effects of link failures are negligible

L#2 - Learning Objectives

- Understand the difference between the concepts of fault, error, and failures
- Understand the fault, error, and failure cause-and-effect relationships

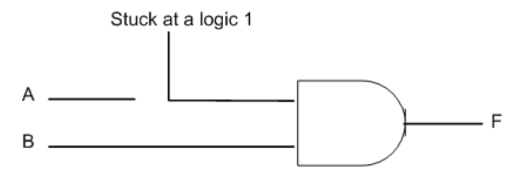
Topics

- Faults, Errors, and Failures
- Causes of Faults
- Characteristics of Faults
- Design Philosophies to Combat Faults

Faults, Errors, and Failures

- Fault -- a physical defect, imperfection, or flaw that occurs in HW or SW component, for example,
  - Shorts between electrical conductors
  - A program loop that when entered can never be exited
- Error -- the occurrence of an incorrect value in some unit of information
  - The manifestation of a fault
  - A deviation from accuracy or correctness
- Failure -- a deviation from the expected performance of a system

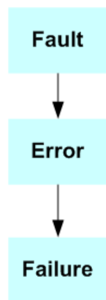
Example (1)



- Suppose a physical short results in line A being permanently stuck at logic 1
- Some condition occurs requiring line A to transit to a logic 0, the value on the line will be in **error.**
  - The correct value for line A should be logic 0, the existence of the fault has caused the line to have an erroneous value → **Faults are the cause of errors!**
- The output F is logic 1 whenever input B is 1, regardless of the actual value applied to line A
  - The error in line A results in the system performing its function incorrectly → a system failure occurs

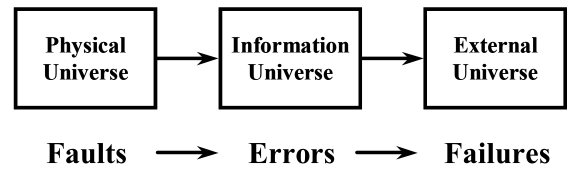
Errors are the cause of failures!

Cause-and-Effect Relationship



- Faults result in errors; errors are the effect of faults
- Errors lead to system failure; failures are the effect of errors

Three Universe Model



- Proposed by Lapire (France), Johnson (UVa)
- Physical universe
  - Semiconductor devices, mechanical elements, displays, printers, power supplies, other physical entities that make up a system
  - A fault is a physical defect or alternation of some component within the physical universe
- Information universe
  - An error occurs when some unit of information (data words within a computer, digital voice or image information) becomes incorrect
- External universe (user’s universe)
  - Where the user of a system ultimately sees the effect of faults and errors
  - Where failures occur

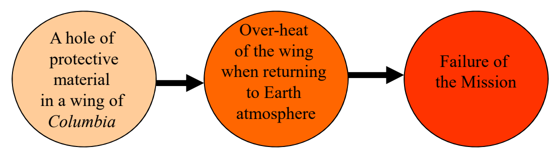
Example (2)

- Accident of shuttle Columbia, Feb. 1, 2003

- There was a hole of protective material in a wing of the Columbia shuttle. When the shuttle returned into earth atmosphere, the hole cause the overheat of the wing. Lastly, the overheat of the wing caused the explosion of the shuttle.

Explanation

- Fault in <u>physical universe</u>: a hole (a physical defect) of protective material in a wing of the shuttle
- Error in <u>informational universe</u>: the hole cause the overheat (a deviation from the correctness) of the wing when the shuttle returned into earth atmosphere
- Failure in <u>external universe</u>: the overheat caused the explosion of the shuttle

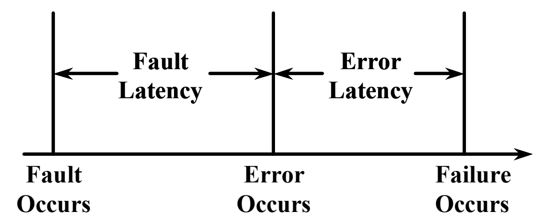


Hands-On Problem

- Devise an original example to illustrate the difference between faults, errors, and failures. As you illustrate these concepts, relate them to the three-universe model.

Definitions of Latency

- Fault latency – the length of time between the occurrence of a fault and the appearance of an error due to that fault
  - Latent fault: a fault that is present but not yet visible. In other words, the fault has not yet produced an error.
- Error latency – the length of time between the occurrence of an error and the appearance of the resulting failure
  - Latent error: an error that is present but not yet visible. In other words, the error has not yet produced a failure.



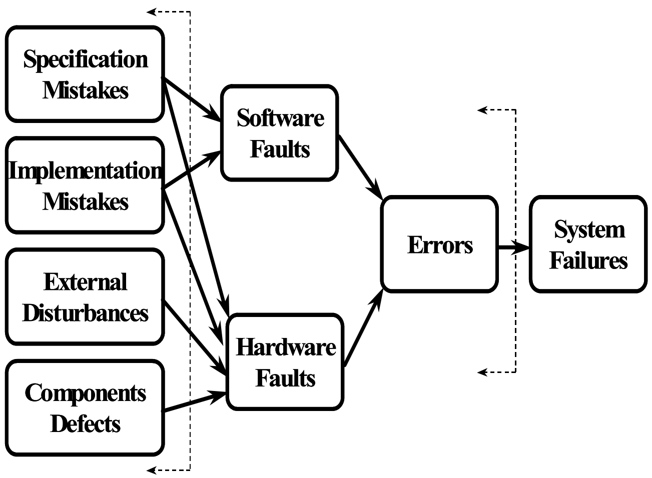
Topics

✓ Faults, Errors, and Failures
- Causes of Faults
- Characteristics of Faults
- Design Philosophies to Combat Faults

Causes of Faults

- Specification mistakes
  - Incorrect algorithms, architectures
  - Hardware/software design mistakes
- Implementation mistakes
  - Poor design, component selection
  - Software coding mistakes
- Component defects
  - Manufacturing imperfections
  - Component wearout
- External disturbances
  - Radiation, electromagnetic interference
  - Operator mistakes
  - Environmental extremes (lightning)

Causes of Faults (Cont'd)



Topics

- General Motivation
- Faults, Errors, and Failures
- Causes of Faults
- Characteristics of Faults
  - Nature
  - Duration
  - Extent
  - Value
- Design Philosophies to Combat Faults

Characteristics of Faults (1)

- Nature: specifies the type of fault
  - Hardware
    - Analog: a power supply fault
    - Digital: a short circuit in a microprocessor
  - Software: a loop when entered can never be exited

Characteristics of Faults (2)

- Duration: specifies the length of time that a fault is active
  - Permanent fault: a fault that remains in existence indefinitely
    - A logic line that is physically stuck at logic 1
  - Transient fault: a fault that appears and disappears in a very short period of time
    - A fault resulting from external disturbance (lightning)
  - Intermittent fault: a fault that appears and disappears repeatedly
    - A fault resulting from a weak solder joint in a circuit

Characteristics of Faults (3)

- Extent: specifies whether the fault is localized to a given HW or SW module or globally affects HW, or SW, or both
  - Local fault: a fault that is confined to a defined area of a system
    - A fault in a memory
  - Global fault: a fault that encompasses an entire system
    - A power supply fault

Characteristics of Faults (4)

- Value:
  - Determinate fault: a fault whose impact does not change with time
    - A voltage that is always "stuck at" ground
  - Indeterminate fault: a fault whose impact can change with time
    - A voltage that oscillates from ground to +5 v

Topics

- General Motivation
- Faults, Errors, and Failures
- Causes of Faults
- Characteristics of Faults
  - Nature
  - Duration
  - Extent
  - Value
- Design Philosophies to Combat Faults

Design Philosophies to Combat Faults (1)

- Three techniques to improve or maintain a system's performance in an environment where faults are of concern
  - Fault avoidance
  - Fault masking
  - Fault tolerance

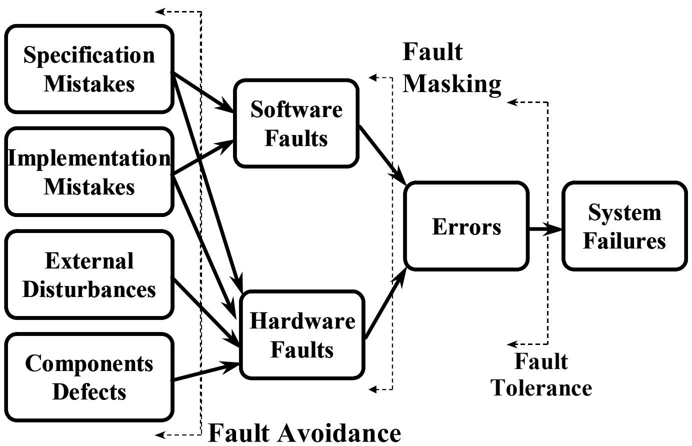
Design Philosophies to Combat Faults (2)

- Fault avoidance: any technique used to prevent faults in the first place
  - Design reviews and testing
- Fault masking: any process that prevents faults in a system from introducing errors into the informational structure of the system
  - Majority voting

Design Philosophies to Combat Faults (3)

- Fault tolerance: the ability of a system to continue to perform its tasks after the occurrence of faults
  - To detect and locate the fault that has occurred and reconfigure the system to remove the faulty component
  - **Reconfiguration**: the process of eliminating a faulty entity from a system and restoring the system to some operational condition or state
    - Fault detection
    - Fault location: where
    - Fault containment: isolating
    - Fault recovery: remaining operational or regaining operational status

Design Philosophies to Combat Faults (4)



Summary of Lecture #2

- Faults, Errors, and Failures
  - Cause-and-effect relationship
  - Three universe model
- 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

Next Topic

- Hardware redundancy techniques