CpE643/CS791F
FAULT-TOLERANT
COMPUTING

Introduction
Bojan Cukic
Lane Department of Computer Science and Electrical Engineering
West Virginia University

Overview
- Motivation
- About the Course and the Instructor
  - Syllabus
  - Introduction
  - Terminology
  - Fundamental Principles
  - Fault-Error-Failure concept

Motivation
- Informal Definition
- Key Attributes
- Who, What and Why Study
- Examples
Motivation

What is Fault-Tolerance?

A "fault-tolerant system" is one that continues to perform at desired level of service in spite of failures in some components that constitute the system.

Motivation (contd.)

Key attributes

Fault - Error - Failure
Performance, Availability, Reliability, etc.

Inclusions of these constraints at design stage is likely to be more cost effective.

Motivation (contd.)

Who is concerned about fault-tolerance?
- System Users

Who is concerned at design stages?
- Universities
  - R, d, and a (Research, development, applications)
- Industry
  - r, D, and A (research, Development, Applications)

Issues
- Design, Analysis/Validation, Implementation, Testing/Validation, Measurement & Evaluation
Motivation (contd.)

Examples
- General Purpose Systems
  - PCs: RAMs with parity checks
  - Workstations: error detection (HW), occasional corrective action (SW), keeping logs (SW)

Motivation (contd.)

Examples
- Reliable Systems
  - Telephone systems
  - Banking systems, e.g. ATM
  - Stock market

Motivation (contd.)

Examples
- Critical and Life Critical Systems
  - Manned and unmanned space borne systems
  - Aircraft control systems
  - Nuclear reactor control systems
  - Life support systems
Motivation (contd.)

Examples
- Reliable -> Critical Systems
  - 911 telephone switching system
  - Traffic light control system
  - Automobile control system (ABS, Fuel injection system)

Introduction

- Historical perspective
- Goals of fault-tolerance
- Applications of fault-tolerance

Introduction (contd.)

- Historical Perspective
  - not a new concept
  - first use by J. van Neumann 1956
    - Probabilistic logic and synthesis of reliable organism from unreliable components, Annals of Mathematical Studies, Princeton University Press

- Major push
  - Space program
  - HW Fault tolerance - then
  - SW Fault tolerance later
  - Merge the two
Introduction (contd.)

- Goals - different goals for different applications
- Intuitive explanations
  - Dependability
  - Service
  - Specification

Introduction (contd.)

- Intuitive concepts
  - Reliability
  - Availability
  - Safety
  - Performability
  - Maintainability
  - Testability

Introduction (contd.)

- Applications
  - Space borne system
    - long life system
  - Airplane control system
    - critical system
  - Transaction processing system
    - high availability system
  - Switching system
    - high availability over certain level of performance
Terminology
- Reliability and concept of probability
- Availability
- Performability
- Dependability

Terminology (contd.)
- Reliability and concept of probability
  - $R(t)$: conditional probability that a system provides continuous proper service in the interval $[0,t]$ given that it provided desired service at time 0.
- Availability
- Performability
- Dependability

Fundamental Principles
- Hardware redundancy
  - Low level
  - High level
- Software Redundancy
- Time Redundancy
- Information Redundancy
Fundamental Principles (contd.)

- Hardware Redundancy - Low level
  - logic level
    - Example 1 - Self checking circuits
    - Example 2 - Arithmetic code
      A modular adder using the mathematical principle
      \[(A+B) \mod k = ((A \mod k) + (B \mod k)) \mod k\]
  - Hardware Redundancy - High level
    - Triples or 5-copies as in Space Shuttle

- Software Redundancy
  - Use two different programs/algorithms

- Time Redundancy
  - Re-compute or redo the task and compare the results
  - May or may not use the same hardware/software

- Information Redundancy
  - Backup information

Question - What level of FT is achieved?

Fault-Error-Failure concept

- Intuitive definitions
- Origins of faults
- Methods to break FEF chain
- Attributes of faults
Fault-Error-Failure concept (contd.)

Intuitive definitions
- **Fault**
  - An anomalous physical condition caused by a manufacturing problem, fatigue, external disturbance (intentional or unintentional), design flaw, coding flow
- **Error**: Effect of activation of a fault
- **Failure**: Visible system effect of an error
  
  Fault -> Error -> Failure

Fault-Error-Failure concept (contd.)

Origins of faults
- Physical device level (HW)
- Logic level (HW)
- Chip level (HW)
- System level (HW/SW)
  - interfacing, specifications, ...
- Why systems fail

Fault-Error-Failure concept (contd.)

Methods to break FEF chain
- Flow FEF
- Barriers
  - Fault avoidance
  - Fault removal
  - Fault forecasting
- **Fault tolerance**
### Fault/Error/Failure concept (contd.)

<table>
<thead>
<tr>
<th>Attribute of faults</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause</td>
<td></td>
</tr>
<tr>
<td>Nature</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td></td>
</tr>
<tr>
<td>Extent</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td></td>
</tr>
</tbody>
</table>