CSCE 5760: Design for Fault Tolerance

You May Want To Know

My Name: Krishna M. Kavi
My Office: F278
My Phone Number: 940-369-7216
My Office Hours: Tuesdays: 10-12 noon
                Thursdays: 1:30-3:30pm
                Other times by appointment
My Email: kavi@cse.unt.edu
Grader: None

Tentative Breakdown of Course Grade

1 Mid-semester exam  25%
Final              25%
Homework Assignments (including presentations) 30%
Project            15%
Discretion         5%

This course consists of both formal lectures and discussions of papers read. I will provide background on concepts and mathematical formulations. I will assign problems as homework assignments based on the material covered in these lectures. In addition, I will assign papers as reading assignments and students are required to make presentations, write reports and participate in class discussions.

Prerequisites: Understanding of hardware design, architecture, designing complex software systems, and good probability and statistics.
CSCE 5760: Design for Fault Tolerance
Course Outline

1. Introduction  
   Motivation  
   Understanding the need for fault tolerance  
   Metrics and Models  
   Performability  
   3 hours

2. Hardware Fault Tolerance  
   Redundancy  
   Watchdog timers  
   Byzantine failures  
   3 hours

3. Software Fault Tolerance  
   N version programming  
   Acceptance tests  
   Exception handling  
   3 hours

4. Information Redundancy  
   Error Correcting Codes  
   Arithmetic codes  
   3 hours

5. Fault Tolerant Networks  
   Measuring reliability  
   Example of networks  
   3 hours

6. Check-pointing  
   Roll back recovery  
   Roll forward recovery  
   3 hours

7. Case Studies  
   6 hours

8. State of the art  
   Impact of fault tolerance on security  
   Hardware design approaches  
   Trading performance for reliability  
   12 hours

Text:  
I. Koren and C. Mani Krishna. "Fault Tolerant Systems"

Other Useful Books:

2. Elena Dubrova. Design of Fault Tolerant Systems
3. Papers from recent conferences on Dependable Computing