VETENSKAP OCH KONST

KUNGL TEKNISKA HÖGSKOLAN

Inte
Master Pr
in System-on-Chip Design

Fault Tolerant System Design

Lecturer

Elena Dubrova
Electronic Systems (ES)
ICT/KTH

dubrova@kth.se
http://www.ict.kth.se/~dubrova

Office hours

- No fixed time
- Send me an email with your questions or ask for a meeting

Teaching Assistant

Shohreh Sarif Mansouri
PhD Student, Electronic Systems
ICT/KTH

shsm@kth.se

Text book

- E. Dubrova, **Fault-Tolerant Design: An Introduction**, draft
- Available from my homepage

Course evaluation

- Midterm exam (20%)
- Final exam (60%)
- 5 assignments (20%)

Assignments

- 5 assignments, worth 20% of the final grade
  - each consists of 5-6 tasks, worth 1-3 points
  - should be handled to me on the due date
  - late assignments will get reduced points (-25% per day)

Examinations

- Midterm exam, 45 min, worth 20% of the final grade
  - will be done during 45 min on a lecture in the middle of the course, 4-5 tasks
  - cannot be re-done
- Final exam, 4 hours, worth 60% of the final grade
  - 4 hours, 10-12 tasks

PhD students

- Additional component for PhD students:
  – select 2 interesting papers/problems, related to the course material
  – bring them to me for discussion
  – I will select one of them
  – you will read this paper/solve the problem, write a 2-page report and give a 20 min talk at the last lecture

Objectives

- understanding fault tolerance
  – faults and their effects (errors, failures)
  – redundancy techniques
  – evaluation of fault-tolerant systems
- balance
  – concepts, underlying principles
  – applications

Overview

- Introduction
  - definition of fault tolerance, applications
- Fundamentals of dependability
  - dependability attributes: reliability, availability, safety
  - dependability impairments: faults, errors, failures
  - dependability means
- Dependability evaluation techniques
  - common measures: failure rate, MTTF, MTTR
  - reliability block diagrams
  - Markov processes

Overview

- Redundancy techniques
  - space redundancy
    - hardware redundancy
    - information redundancy
    - software redundancy
  - time redundancy

KUNGL TEKNISKA HÖGSKOLAN

Inte

Master Pr

in System-on-Chip Design

Introduction to Fault Tolerance

Fault tolerance

fault-tolerance is the ability of a system to continue performing its function in spite of faults

broken connection	**hardware**
bug in program	**software**

Easily testable system

- Easily testable system is one whose ability to work correctly can be verified in a simple manner

Why do we need fault-tolerance?

- It is practically impossible to build a perfect system
  - suppose a component has the reliability 99.99%
  - a system consisting of 100 non-redundant components will have the reliability 99.01%
  - a system consisting of 10.000 components will have the reliability 36.79%
- It is hard to forsee all the factors

Redundancy

- Redundancy is the provision of functional capabilities that would be unnecessary in a fault-free environment
  - replicated hardware component
  - parity check bit attached to digital data
  - a line of program verfiying the correcntess of the resut

History

- early computer systems
  - basic components had very low reliability
  - fault-tolerant techniques were need to overcome it
    - redundant structures with voting
    - error-detection and error correction codes

History

- early computer systems
  - EDVAC (1949)
    - duplicate ALU and compare results of both
    - continue processing if agreed, else report error
  - Bell Relay Computer (1950)
    - 2 CPU's
    - one unit begin executing the next instruction if the other encounts an error
  - IBM650, UNIVAC (1955)
    - parity check on data transfers

History

- Advent of transistors
  - more reliable components
  - led to temporary decrease in the emphasis on fault-tolerant computing
  - designers thought it is enough to depend on the improved reliability of the transistor to guarantee correct computations

History

- last decades
  - more critical applications
    - space programs, military applications
    - control of nuclear power stations
    - banking transactions
  - VLSI made the implementation of many redundancy techniques practical and cost effective
  - Other than hardware component faults need to be tolerated:
    - transient faults (soft errors) caused by environmental factors
    - software faults

Applications

- safety-critical applications
  - critical to human safety
    - aircraft flight control
  - environmental disaster must be avoided
    - chemical plants, nuclear plants
  - requirements
    - 99.99999% probability to be operational at the end of a 3-hour period

Applications

- mission-critical applications
  - it is important to complete the mission
  - repair is impossible or prohibitively expensive
    - Pioneer 10 was launched 2 March 1970, passed Pluto 13 June 1983
- requirements
    - 95% probability to be operational at the end of mission (e.g. 10 years)
    - may be degraded or reconfigured before (operator interaction possible)

Applications

- bisness-critical applications
  – users want to have a high probability of receiving service when it is requested
  – transaction processing (banking, stock exchange or other time-shared systems)
    - ATM: < 10 hours/year unavailable
    - airline reservation: < 1 min/day unavailable

Applications

- maintenance postponement applications
  - avoid unscheduled maintenance
  - should continue to function until next planned repair (economical benefits)
  - examples:
    - remotely controlled systems
    - telephone switching systems (in remote areas)

Goals of fault tolerance

The main goal of fault tolerance is

to increase the dependability of a system

Dependability

Dependability
is the ability of a system to deliver its intended level of service to its users

Dependability tree

- **dependability**
  - **attributes**
    - **availability**
    - **reliability**
    - **safety**
  - **means**
    - **fault tolerance**
    - **fault prevention**
    - **fault removal**
    - **fault forecasting**
  - **impairments**
    - **faults**
    - **errors**
    - **failures**

Reliability

- R(t) is the probability that a system operates without failure in the interval [0,t], given that it worked at time 0

- We need high reliability when:
  - even momentary periods of incorrect performance are unacceptable (aircraft, heart pace maker)
  - no repair possible (satellite, spacecraft)

High reliability examples

- airplane:
  - R(several hours) = 0.999 999 9 = 0.9_7

- spacecraft:
  - R(several years) = 0.95

Reliability versus fault tolerance

- Fault tolerance is a technique that can improve reliability, but
  - a fault tolerant system does not necessarily have a high reliability
  - a system can be designed to tolerate any single error, but the probability of such error to occur can be so high that the reliability is very low

Reliability versus fault tolerance

- A highly reliable system is not necessarily fault tolerant
  - a very simple system can be designed using very good components such that the probability of hardware failing is very low
  - but if the hardware fails, the system cannot continue its functions

How fault tolerance helps

- Fault tolerance can improve a system's reliability by keeping the system operational when hardware or software faults occur
  - a computer system with one redundant processor can be designed to continue working correctly even if one of the processors fails
  - **QUESTION:** Will a fault-tolerant system always be more reliable than an individual component?

Availability

- A(t) is the probability that a system is functioning correctly at the instant of time t

- depends on
  - how frequently the system becomes non-operational
  - how quickly it can be repaired

Steady-state availability

- Often the availability assumes a time-indepentent value after some initial time interval
- This value is called steady-state availability A_{ss}
- Steady-state availability is often specified in terms of downtime per year

A_{ss} = 90%, downtime = 36.5 days/year

A_{ss} = 99%, downtime = 3.65 days/year

Reliability versus availability

- reliability depends on an interval of time
- availability is taken at an instant of time
- a system can be highly available yet experience frequent periods of being non-operational as long as the length of each period is extremely short

High availability examples

- examples
  - transaction processing
    - ATM: $A_{ss}=0.9_3$ (< 10 hours/year unavailable)
    - banking: $A_{ss}=0.997$ (< 10 s/hour unavailable)
  - computing
    - supercomputer centres
      $A_{ss}=0.997$ (< 10 days/year unavailable)
  - embedded
    - telecom: $A_{ss}=0.9_5$ (< 5 min./year unavailable)

How fault tolerance helps

- Fault tolerance can improve a system's availability by keeping the system operational when a failure occur
  - a spare processor can perform the functions of the system, keeping its available for use, while the primary processor is being repaired

Safety

- Safety is the probability that a system will either perform its function correctly or will discontinue its operation in a safe way

- System is safe
  – if it functions correctly, or
  – if it fails, it remains in a safe state

High safety examples

- railway signalling
  - all semaphores red
- nuclear energy
  - stop reactor if a problem occur
- banking
  - don’t give the money if in doubt

Reliability versus safety

- Reliability is the probability that a system will perform its functions correctly
- Safety is the probability that a system will either work correctly or will stop in a manner that causes no harm

How fault tolerance helps

- Fault tolerance techniques can improve safety by turning a system off if a failure of a certain sort is detected
  - in a nuclear power plant the reaction process should be stopped if some discrepancy is detected

Summary: attributes of dependability

- reliability:
  - continuity of service
- availability:
  - readiness for usage
- safety:
  - non-occurrence of catastrophic consequences on environment

Next lecture

- Faults, error and failures
- Design philosophies to combat faults