Projeto com Circuitos Reconfiguráveis
Projeto de Sistemas em Chip

Fault Tolerant System Design

Prof. Daniel M. Muñoz Arboleda

FGA - UnB

Lecture plan

- Introduction
  - Definition of fault tolerance
  - Applications of fault tolerant system design
- Fundamentals of dependability
  - dependability attributes: reliability, availability, safety
  - dependability impairments: faults, errors, failures
- Some dependability evaluation techniques
  - common measures: failure rate, MTTF, MTTR
- Redundancy techniques
  - space redundancy
    - hardware redundancy
    - information redundancy
    - software redundancy
  - time redundancy

Fault Tolerance is the ability of a system to continue performing its function in spite of faults

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

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

Applications of fault tolerance

- **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 of fault tolerance

- **mission-critical** applications
  - – it is important to complete the mission
  - – repair is impossible or prohibitively expensive
    - Aerospace applications such as satellite, spacecraft
- 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 of fault tolerance

- **business-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 of fault tolerance

- **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)

Lecture plan

- Introduction
  - Definition of fault tolerance
  - Applications of fault tolerant system design
- **Fundamentals of dependability**
  - dependability attributes: reliability, availability, safety
  - dependability impairments: faults, errors, failures
- Some dependability evaluation techniques
  - common measures: failure rate, MTTF, MTTR
- Redundancy techniques
  - space redundancy
    - hardware redundancy
    - information redundancy
    - software redundancy
  - time redundancy

Goals of fault tolerance

The main goal of fault tolerance is
to increase the **dependability (confiabilidade)** of a system

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)

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)
    - Airplane: R(several hours) = 0.999 999 9 = 0.9_7
  - no repair possible (satellite, spacecraft)
    - Spacecraft: R(several years) = 0.95

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

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
- Often the availability assumes a time-indepentent value after some initial time interval
- 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

Availability

- High availability 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)

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

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

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

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

Summary

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

Lecture plan

- Introduction
  - Definition of fault tolerance
  - Applications of fault tolerant system design
- **Fundamentals of dependability**
  - dependability attributes: reliability, availability, safety
  - dependability impairments: faults, errors, failures
- Some dependability evaluation techniques
  - common measures: failure rate, MTTF, MTTR
- Redundancy techniques
  - space redundancy
    - hardware redundancy
    - information redundancy
    - software redundancy
  - time redundancy

Fault (Falta)

Fault is a physical defect, imperfection or flaw that occurs in hardware or software

Example:
- short between wires
- break in transistor
- infinite program loop

Error

Error is a deviation from correctness or accuracy

Example: Suppose a line is physically shortened to 0 (there is a fault). As long as the value on line is supposed to be 0, there is no error.

Errors are usually associated with incorrect values in the system state.

Failure (Falha)

Failure is a non-performance of some action that is due or expected

Example: Suppose a circuit controls a lamp (0 = turn off, 1 = turn on) and the output is physically shortened to 0 (there is a fault). As long as the user wants the lamp off, there is no failure.

A system is said to have a failure if the service it delivers to the user deviates from compliance with the system specification.

Cause-and-effect relationship

- Faults can result in errors. Errors can lead to system failures.

- Errors are the effect of faults. Failures are the effect of errors.
  Fault → Error → Failure

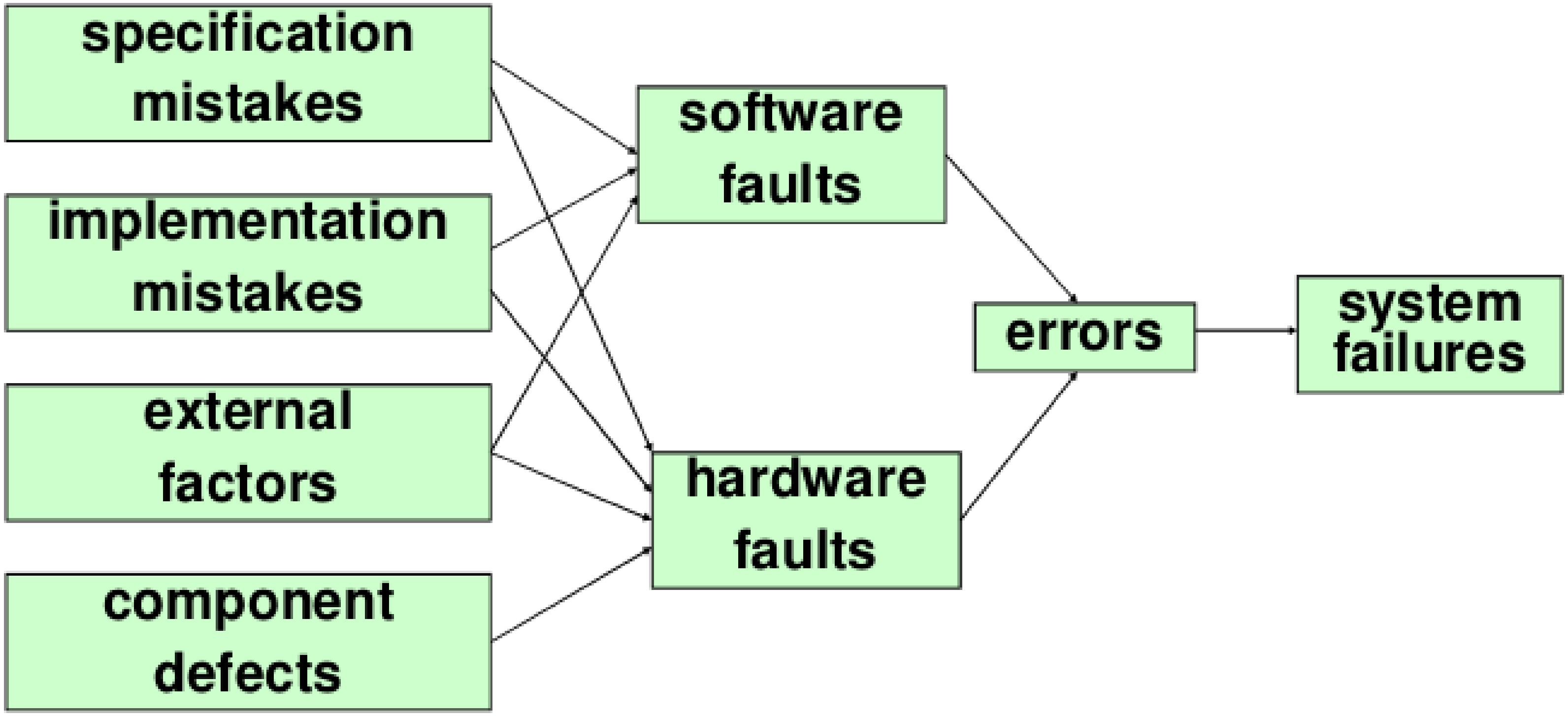
- Example in context of software:

Bug in a program is a fault. Possible incorrect values caused by this bug is an error. Possible crush of the operating system is a failure.

Origins of faults

- specification mistakes
  - incorrect algorithms, incorrectly specified requirements (timing, power, environmental)
- implementation mistakes
  - poor design, software coding mistakes
- component defects
  - manufacturing imperfections, random device defects, components wear-outs
- external factors
  - **radiation**, lightning, operator mistakes

Cause-and-effect relationship



Lecture plan

- Introduction
  - – Definition of fault tolerance
  - – Applications of fault tolerant system design
- Fundamentals of dependability
  - – dependability attributes: reliability, availability, safety
  - – dependability impairments: faults, errors, failures
- **Some dependability evaluation techniques**
  - – common measures: failure rate, MTTF, MTTR
- Redundancy techniques
  - – space redundancy
    - hardware redundancy
    - information redundancy
    - software redundancy
  - – time redundancy