Projeto com Circuitos Reconfiguráveis
Projeto de Sistemas em Chip

Fault Tolerant System Design

Prof. Daniel M. Muñoz Arboleda

FGA - UnB

Overview

- 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

Techniques for fault tolerance

- Fault masking “hides” faults that occur. Do not require detecting faults, but require containment of faults (the effect of all faults should be local)

- Another approach is to first to detect, locate and contain faults, and then to recover from faults using reconfiguration.

Redundancy

- hardware redundancy
    - 2nd CPU, 2nd ALU, ...
- software redundancy
    - validation test...
- information redundancy
    - error-detecting and correcting codes, ...
- time redundancy
    - repeating tasks several times, ...

Redundancy

- NOTHING FOR FREE!
- costs
  - HW: components, area, power, ...
  - SW: development costs, ...
  - information: extra HW to code / decode
  - time: faster CPUs, components
- trade-off against increase in dependability

Overview

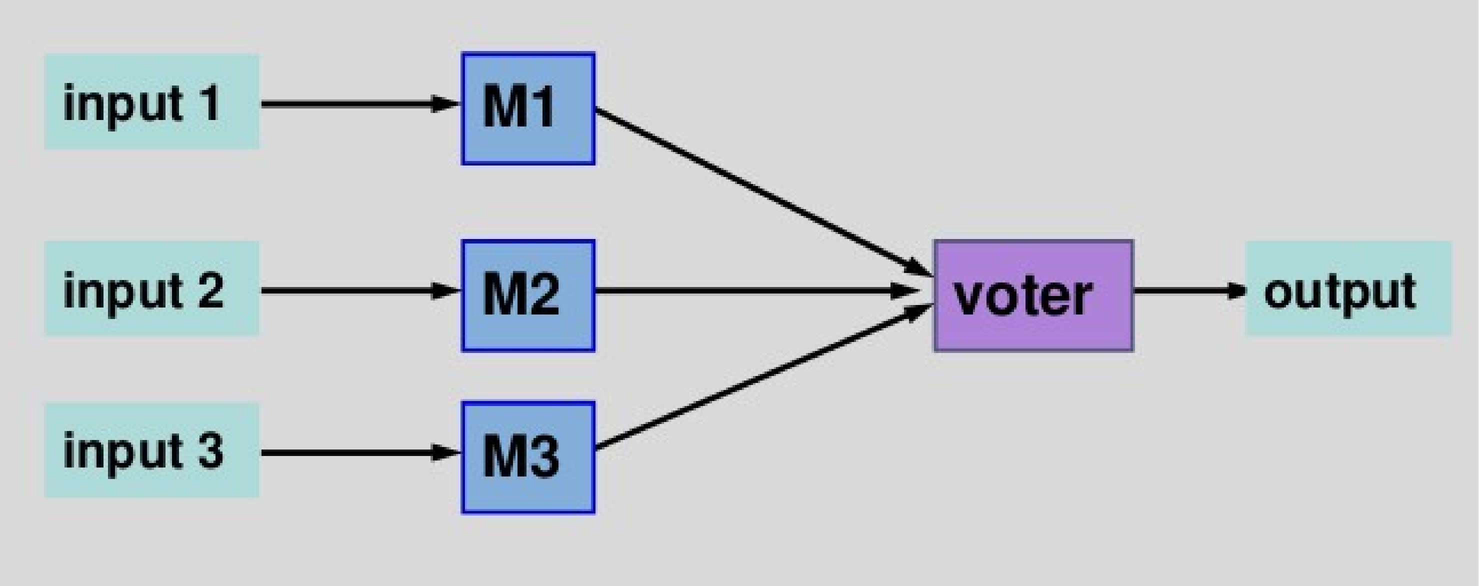
- Introduction
  - – Definition of fault tolerance
  - – Applications of fault tolerant system design
- Fundamentals of dependability
  - – dependability attributes: reliability, availability, safety
  - – dependability impairments: faults, errors, failures
- Some dependability evaluation techniques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

HW redundancy: overview

- passive redundancy techniques
  - fault masking
- active redundancy techniques
  - detection, localization, containment, recovery
- hybrid redundancy techniques
  - static + dynamic
  - fault masking + reconfiguration

Passive HW redundancy
Triple Modular Redundancy (TMR)

- – 3 active components
- – fault masking by voter
- – Problem: voter is a single point of failure



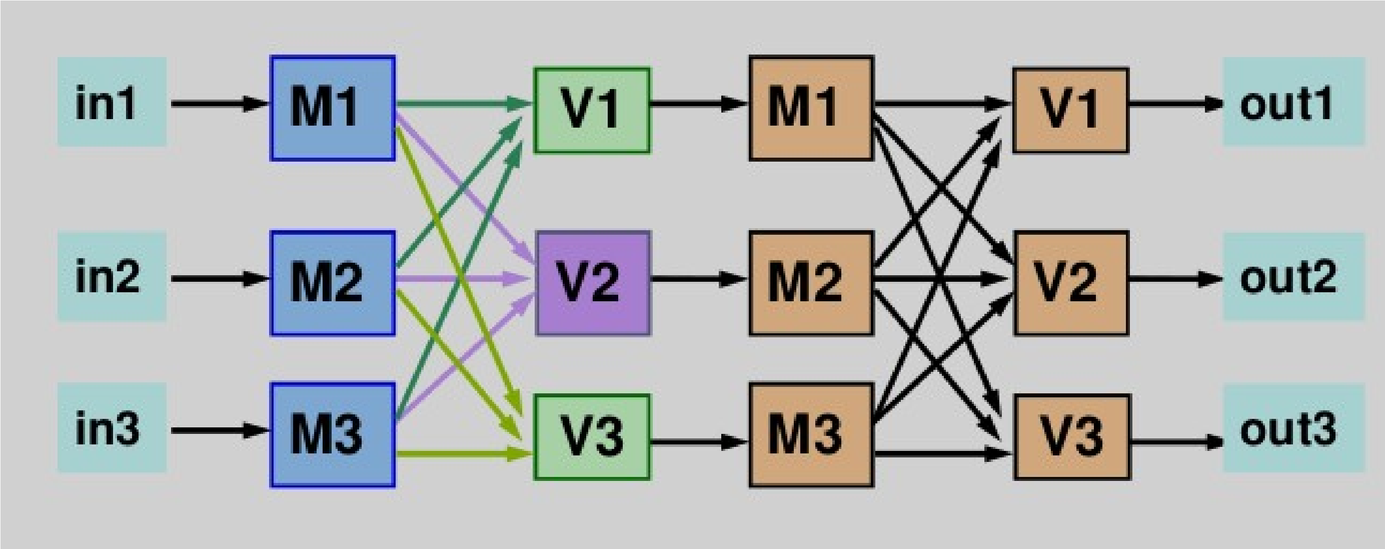
Passive HW redundancy
N Modular Redundancy (NMR)

- *N*-modular redundancy (NMR)
  - *N* active components
  - *N* odd, for majority voting
  - tolerates $\lfloor N/2 \rfloor$ module faults

- example Apollo
  - N=5
  - 2 faults can be tolerated (masked)

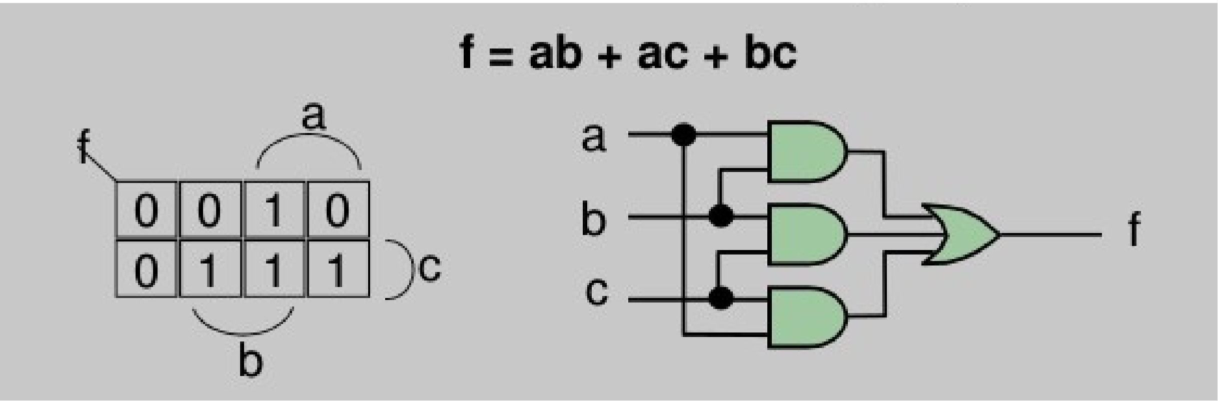
Passive HW redundancy
N Modular Redundancy (NMR)

- *N*-modular redundancy (NMR)
  - – *N* active components
  - – *N* odd, for majority voting
  - – tolerates $\lfloor N/2 \rfloor$ module faults



HW voting

- hardware realisation of 1-bit majority voter



f = ab + ac + bc

0	0	1	0
0	1	1	1

- *n*-bit majority voter: *n* times 1-bit
- requires 2 gate delays

SW voting

- Voting can be performed using software
- voter is software implemented by a microprocessor
- voting program can be as simple as a sequence of three comparisons, with the outcome of the vote being the value that agrees with at least on on the other two

- HW: fast, but expensive
  - 32-bit voter: 128 gates and 256 flip-flops
- SW: slow, but more flexible
  - use existing CPUs

Problem with voting

- Major problem with practical application of voting is that the three results may not completely agree
  - sensors, used in many control systems, can seldom be manufactured so that their values agree exactly
  - analog-to-digital converter can produce quantities that disagree in the least significant bits