DESIGN OF DEPENDABLE SYSTEMS on CHIPS

Giovanni De Micheli CSL - Stanford University

Outline

- Introduction to dependable design
- Component redundancy
- Reliable interconnect
- Robust design
- Summary and conclusions

Systems on Chips

- Embedded systems market
  - Communication, Consumer, Medical, Vehicle…
- Concerns:

Correctness
Reliability and safety
Robustness

Performance
Energy consumption
Cost

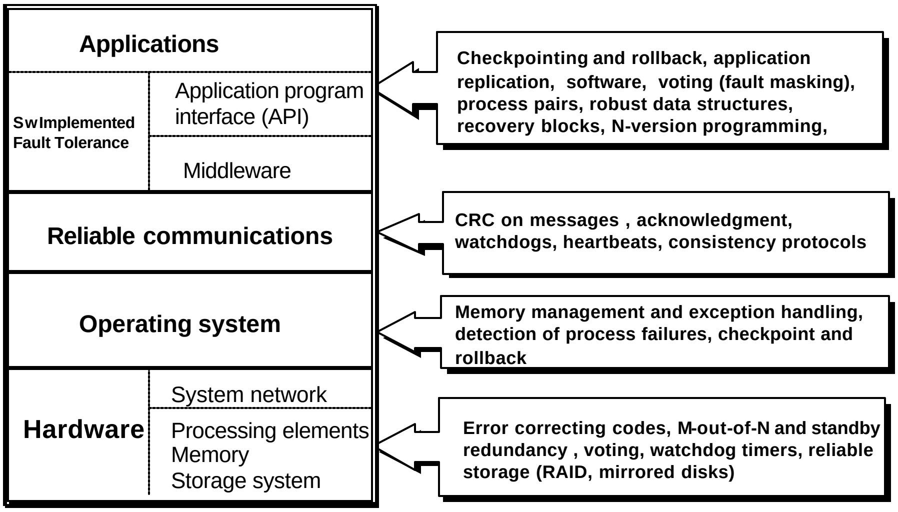
Dependable design
where do we need it ?

- **Traditional applications**
  - Long-life applications (e.g., unmanned and manned space missions)
  - Life-critical, short-term applications (e.g., aircraft engine control, fly-by-wire)
  - Defense applications (e.g., aircraft, guidance & control)
  - Nuclear industry
  - Telecommunications
- **Newer critical-computation applications**
  - Health industry
  - Automotive industry
  - Industrial control systems and production lines
  - Banking, reservations, commerce
- **Very large-scale scientific computing**
  - The new 10 Teraflop machine (IBM)

The economic perspective

- Availability is a critical business metric for commercial systems and services
  - Nearly 100% availability (“five nines+”) is almost mandatory
- Service outages are frequent
  - 65% website managers report outages over a 6-month period
  - 25% report three or more outages [*Internet week 2000*]
- High cost of downtime of systems providing vital services
  - Lost opportunities, lost revenues, non-compliance penalties, potential loss of lives
  - Cost per an hour of downtime varies from $89K for cellular services to $6.5M for stock brokerage [Gartner Group 1998]
- Revenue for HA products in the data/telecom/computer server market is over $100B (˜ $15B for servers alone) [IMEX Research 2003]

Achieving reliable systems



Checkpointing and rollback, application replication, software, voting (fault masking), process pairs, robust data structures, recovery blocks, N-version programming,

CRC on messages , acknowledgment, watchdogs, heartbeats, consistency protocols

Memory management and exception handling, detection of process failures, checkpoint and rollback

Error correcting codes, M-out-of-N and standby redundancy , voting, watchdog timers, reliable storage (RAID, mirrored disks)

Malfunctions

- Manufacturing imperfections
  - More likely to happen as lithography scales down
- Approximations during design
  - Uncertainty about details of design
- Environment-induced
  - Soft-errors, electro-magnetic interference
- Operating-mode induced
  - Extremely-low voltage supply

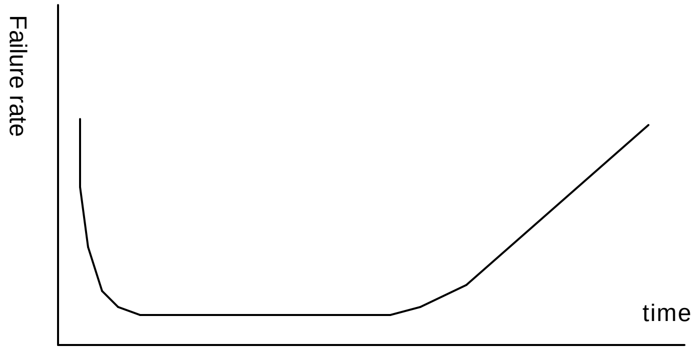
Malfunctions and faults

- Malfunctions are captured by:
  - Faults
    - Abstractions of the malfunctions
  - Failure modes
    - Way in which the malfunction manifests
  - Failure rates
    - Related to failure probability

Defining the problems…

- Failure rate:
  - Assuming a unit works correctly in [0,t], the conditional probability ?(t) that a unit fails in [t, t +? t]
  - Typically the failure ? rate depends on
    - Temperature
    - Time (burn-in and aging)
    - Environmental exposure
      - Soft errors, EMI
  - Often the component failure rate is assumed to be constant for simplicity

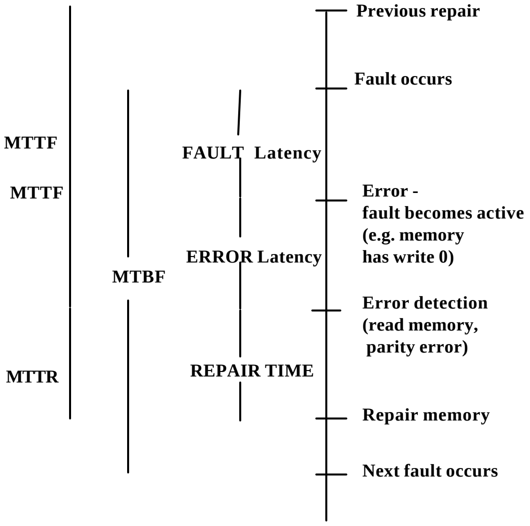
Failure rate
the bathtub curve



Reliability

- The probability function R(t) that a system works correctly in [0, t] without repairs
- Reliability is a function of time
  - If the system consist of a single component with constant failure rate ?, then
    - R(t) = exp (– ?t)
  - The mean time to failure is MTTF = 1/ ?
- In general, the MTTF is E[t] = ?R(t)dt

Dependability Concepts



Reliability:
a measure of the continuous delivery of service;
R(t) is the probability that the system survives (does not fail) throughout [0, t];
expected value: ***MTTF(Mean Time To Failure)***

Maintainability:
a measure of the service interruption
M(t) is the probability that the system will be repaired within a time less than t;
expected value: ***MTTR (Mean Time To Repair)***

Availability:
a measure of the service delivery with respect to the alternation of the delivery and interruptions
A(t) is the probability that the system delivers a proper (conforming to specification)service at a given time t.
expected value: ***EA = MTTF / (MTTF + MTTR)***

Safety:
a measure of the time to catastrophic failure
S(t) is the probability that no catastrophic failures occur during [0, t];
expected value:
MTTCF(Mean Time To Catastrophic Failure)

Reliability of complex systems

- A system is a connection of components
- System reliability depends on the topology
  - Series/parallel configurations
  - N out of K configurations
  - General topologies
- Common mode failures
  - Failure mode that affects all components
  - Examples:
    - Failure of voltage regulator for SoC
    - Failure of scheduler to process exception routines

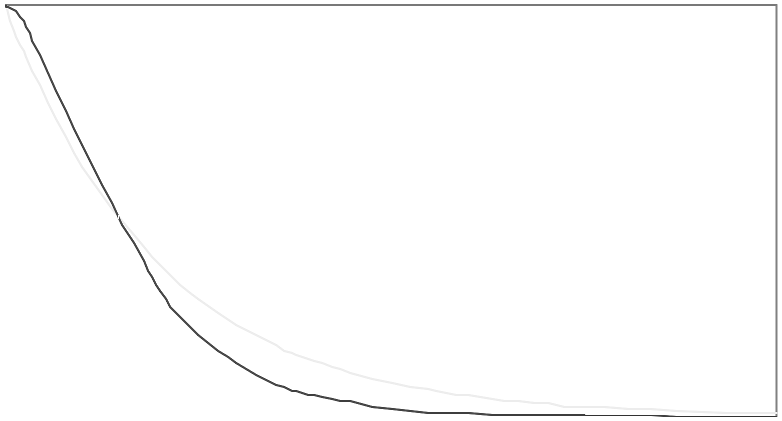
Very simple example

- For reliability analysis, a system consists of three components:
  - Processor, memory, bus
- All components have to be up at the same time to accomplish the mission
- The three components form a series configuration
- The system reliability is the product of the component reliabilities (if the failure rates are independent)
- Assume failure rates constant:
  - The system failure rate is the sum of the failure rates
  - The MTTF is its inverse

Example (2)

- For reliability analysis, a system consists of two processors:
  - A working processor suffices to accomplish the mission
- The two components form a parallel configuration
- The system unreliability is the product of the component unreliabilities (if the failure rates are independent)
  - $R(t) = 1 – [1-R_1(t)]\ [1-R_2(t)]$
  - Assume failure rates constant
  - The MTTF is $1/?_1 + 1/?_2 +1/ (?_1 +?_2)$
- Other relevant configurations:
  - Standby
  - Triple modular redundancy

TMR vs simplex reliability

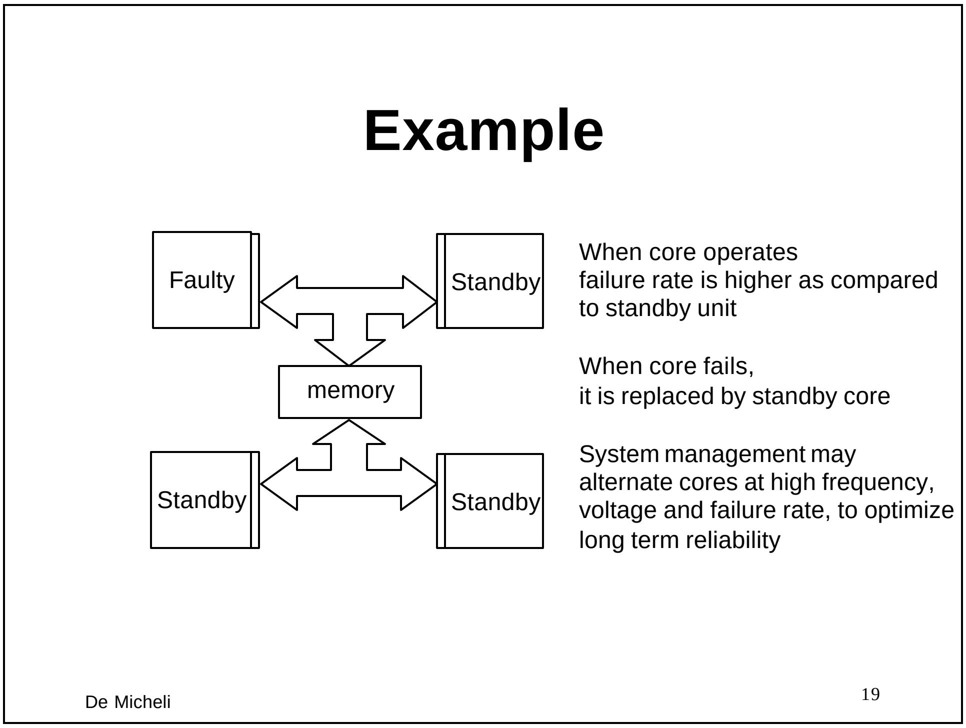


Outline

- Introduction to dependable design
- Component redundancy
- Reliable interconnect
- Robust design
- Summary and conclusions

Providing component redundancy

- Component redundancy for enhanced reliability
  - Energy consumption penalty may be severe
- Power-managed standby components
  - Provide for temporary/permanent back-up
  - Provide for load and stress sharing
- Power management and reliability are intertwined:
  - PM allows reasonable use of redundancy on chip
  - Failure rates depend on effect of PM on components
- A programmable and flexible interconnection means is required



Example

Faulty

Standby

memory

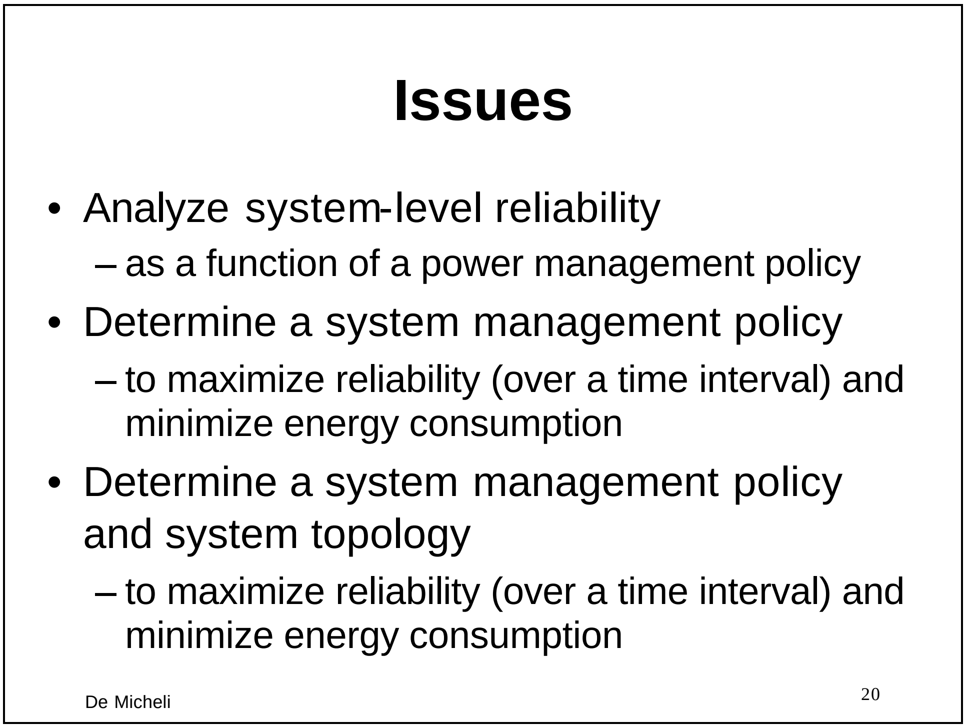
Standby

Standby

When core operates
failure rate is higher as compared
to standby unit

When core fails,
it is replaced by standby core

System management may
alternate cores at high frequency,
voltage and failure rate, to optimize
long term reliability



Issues

- Analyze system-level reliability
  - as a function of a power management policy
- Determine a system management policy
  - to maximize reliability (over a time interval) and minimize energy consumption
- Determine a system management policy and system topology
  - to maximize reliability (over a time interval) and minimize energy consumption

Outline

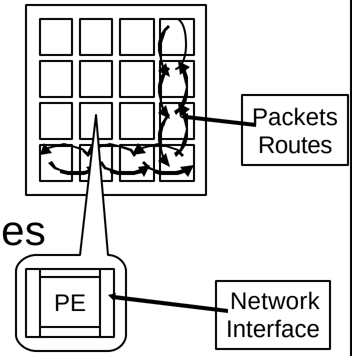
- Introduction to dependable design
- Component redundancy
- Reliable interconnect
- Robust design
- Summary and conclusions

Why on-chip networking ?

- Provide a structured methodology for realizing on-chip communication schemes
  - Modularity
  - Flexibility
- Cope with inherent limitations of busses
  - Performance and power of busses do not scale up
- Support reliable operation
  - Layered approach to error detection and correction

Interconnect design in a multi-processing environment

- Most SoCs are multi-processors
  - Homogeneous
    - High performance computation
  - Heterogeneous
    - Application specific solutions
- Classic and *ad hoc* topologies
- Different QoS requirements
  - Best-effort services
  - Guaranteed performance



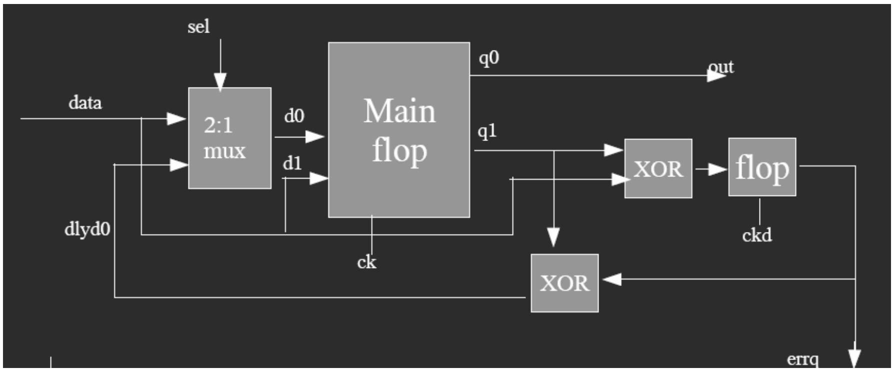
Micro-network characteristics

- Micro-networks bring networking techniques to silicon
  - Combination of networking and VLSI design techniques
- Differences between micro and macro networks:
  - Problem characterization:
    - SoCs have more predictable parameters than macro networks
  - Objectives and constraints:
    - Low communication latency
      - Standards like TCP/IP are impractical
    - Low communication energy consumption
    - Limited adherence to standards
      - SoCs are typically proprietary designs
    - Tailoring micro-networks to specific applications
  - Synthesis of micro-networks

Providing communication reliability

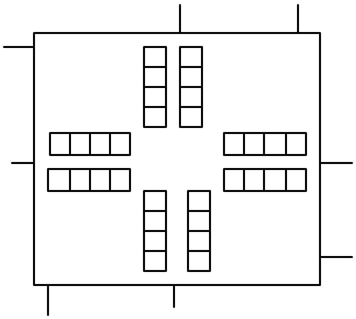
- Some network topologies support multiple source/destination paths
  - Tolerate transient congestion, transient and permanent link malfunctions
- Error detection and correction
  - Physical links
    - Timing-errors detection by shadow latches
  - Switches and routers
    - Flit-level error detection and correction with CRCs
  - Network interface
    - Packet integrity check
  - Processor cores
    - Software data correctness check

Shadow latch



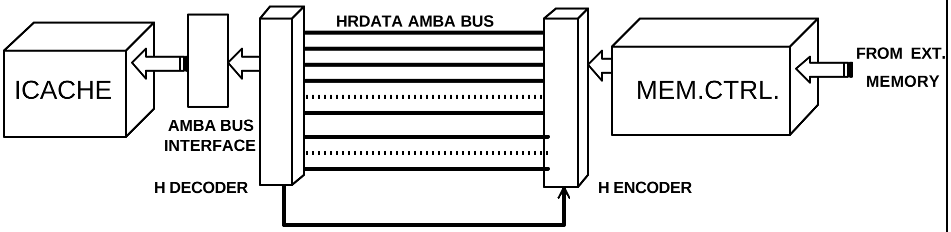
Error-resilient switches

- Switches are pipelined
- Forward flow control
  - – A flit is transmitted only when storage is available at the receiving end
- Buffering at output with virtual channels
- Distributed error detection logic
  - – CRC decoders work in parallel with switch

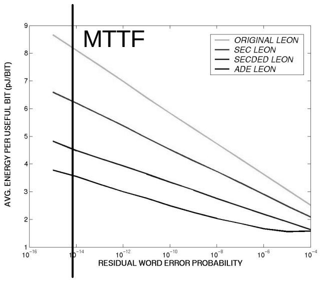


Example of a 4-in, 4-out switch with 2 virtual channels per output

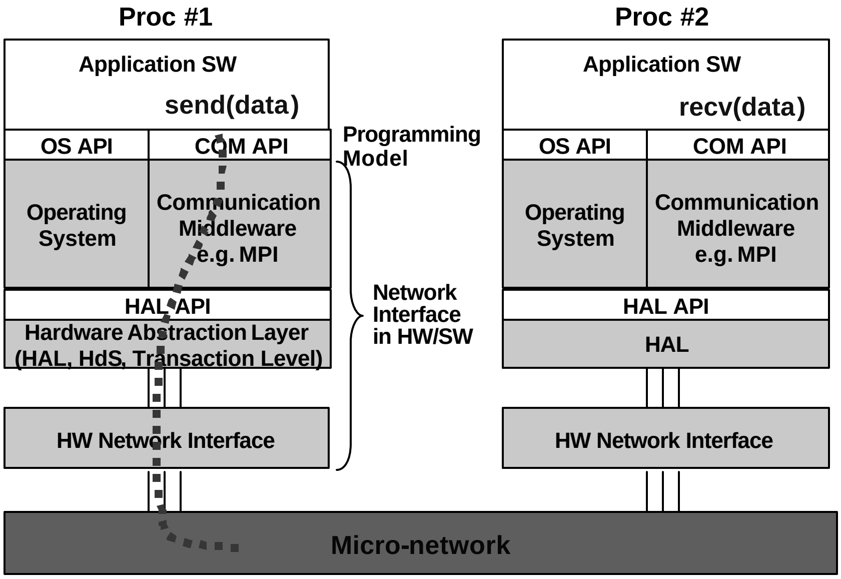
Boundary correction



- Compare original AMBA bus to extended bus with error detection and correction or retransmission
  - – SEC coding
  - – SEC-DED coding
  - – ED coding
- Explore energy efficiency



System view of communication



Proc #1

Application SW

send(data)

OS API

COM API

Operating System

Communication Middleware e.g. MPI

HAL API

Hardware Abstraction Layer (HAL, HdS, Transaction Level)

HW Network Interface

Programming Model

Network Interface in HW/SW

Proc #2

Application SW

recv(data)

OS API

COM API

Operating System

Communication Middleware e.g. MPI

HAL API

HAL

HW Network Interface

Micro-network

Outline

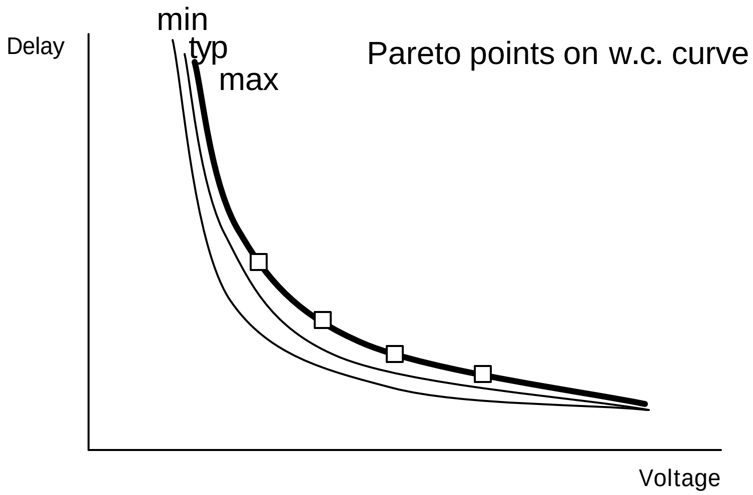
- Introduction to dependable design
- Component redundancy
- Reliable interconnect
- Robust design
- Summary and conclusions

Robust design

- A system must preserve correct operation:
  - Under varying environmental conditions
  - Under changes of design assumptions
- Informal definition of robustness:
  - Ability of a system to work correctly in presence of invalid input parameters
- There is a close link to sensitivity analysis
  - Robustness relates to the size of the input parameter space under which the system performs correctly

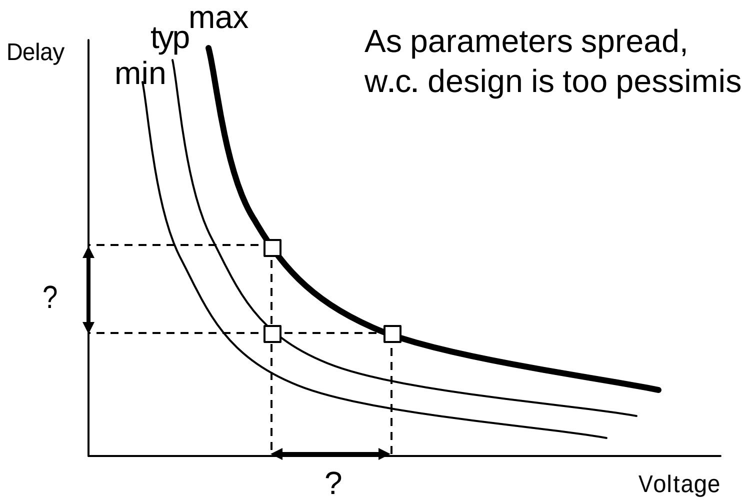
Design space exploration

worst case analysis



Adaptive design space

worst case analysis



As parameters spread,
w.c. design is too pessimistic

Self-calibrating circuits

- The operating points of a circuit should be determined on-line
  - Variation from chip to chip
  - Operation close to the edge of failure
- Analogy
  - Sailing boat tacking upwind
  - Max gain when sailing close to wind
    - When angle is too close, large loss of speed

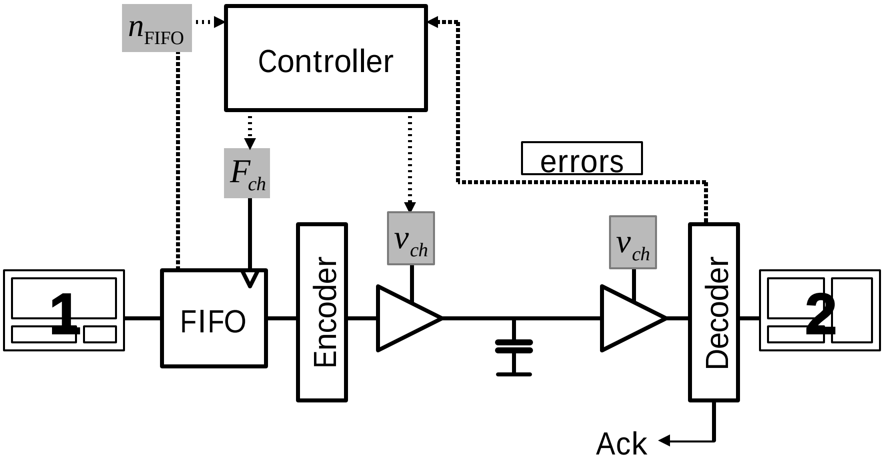
© Photo wave / Louis Vuitton

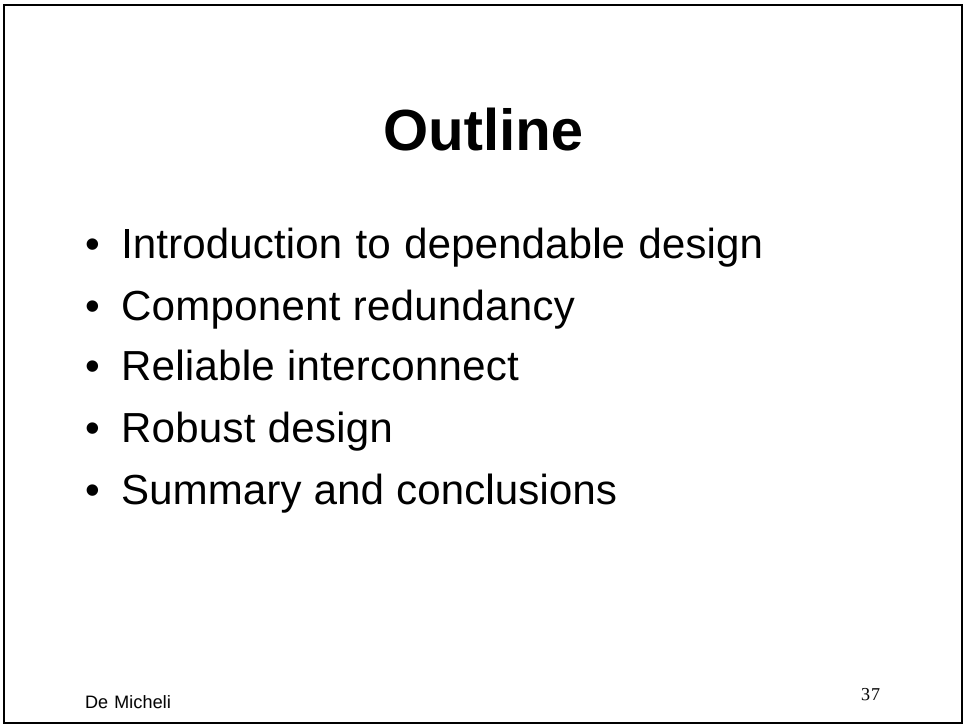
How to calibrate?

- General paradigm
  - A circuit may be in *correct* or *faulty* operational state, depending on a parameter (e.g., voltage)
  - Computed/transmitted data need checks
    - If data is faulty, data is recomputed and/or retransmitted
  - Error rate is monitored on line
  - Feedback loop to control *operational state parameter* based on error rate

Circuits can generate errors:
- Errors must be detected and corrected
- Correction rate is used for calibration

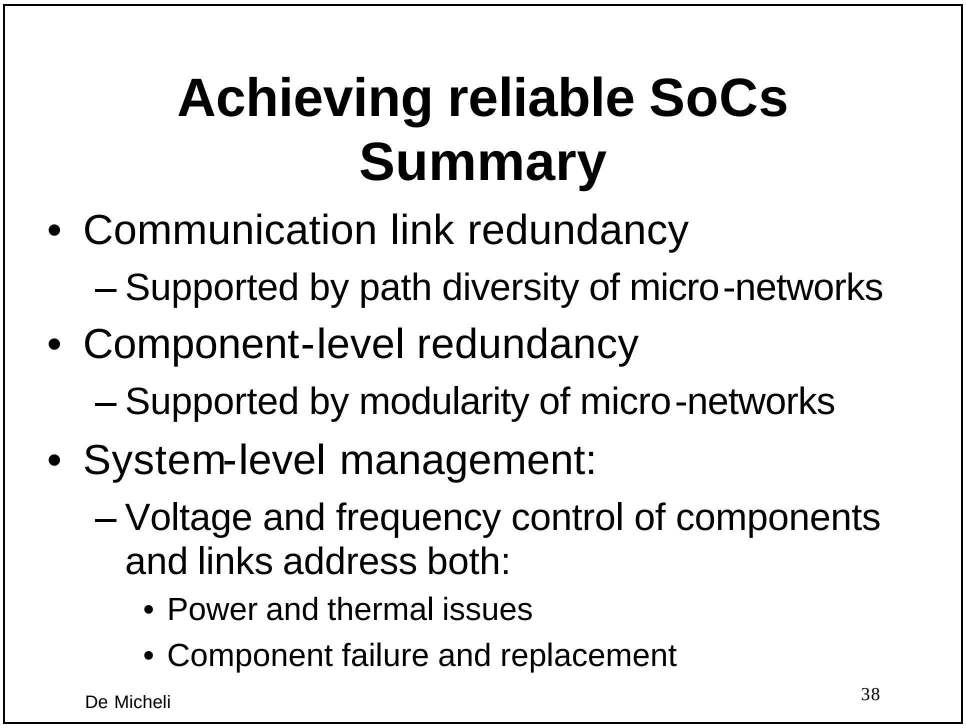
Adaptive packet-based link





Outline

- Introduction to dependable design
- Component redundancy
- Reliable interconnect
- Robust design
- Summary and conclusions



Achieving reliable SoCs Summary

- Communication link redundancy
  - – Supported by path diversity of micro-networks
- Component-level redundancy
  - – Supported by modularity of micro-networks
- System-level management:
  - – Voltage and frequency control of components and links address both:
    - Power and thermal issues
    - Component failure and replacement

Conclusions

- Micro-networks are the ideal platform to integrate multi-processor systems
- Micro-networks support seamlessly redundant stand-by components for achieving high reliability
- Micro-networks provide fault-tolerant communication by supporting alternative paths

Thank You