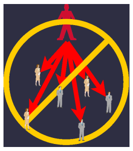
Embedded Systems: An Application-Centered Approach

Robert Dick

http://robertdick.org/esaca/
Office: 2417-E EECS
Department of Electrical Engineering and Computer Science
University of Michigan



Types of reliability

- Algorithm correctness: Does the specification have the desired properties?
- Robustness in the presence of transient faults: Can the system continue to operate correctly despite temporary errors?
- Robustness in the presence of permanent faults: Can the system continue to operate correctly in the presence of permanent errors?

Conventional software testing

- Implement and test
- Number of tests bounded but number of inputs huge
- Imperfect coverage

Model checking

- Use finite state system representation
- Use exhaustive state space exploration to guarantee desired properties hold for all possible paths
- Guarantees properties
- Difficulty with variables that can take on many values
  - Symbolic techniques can improve this
- Difficulty with large number of processes

Critical barriers to use

- For simple systems, manual proofs possible
- For very complex systems, state space exploration intractable
- May require new, more formal, specification language

Overcoming barriers to use

- Automatic abstraction techniques permitting use on more complex systems
  - Difficult problem
- Target moderate-complexity systems where reliability is important
  - Medical devices
  - Transportation devices
  - Electronic commerce applications
- Give users a high-level language that is actually easier to use than their current language, and provide a path to a language used in existing model checkers

Cross-talk

- Shielding
- Bus encoding

Particle impact

- Temporal redundancy
- Structural redundancy
- Voltage control

Random background offset charge

- Improvements to fabrication
- Temporal redundancy
- Structural redundancy

Temperature-induced timing faults

- Preemptive throttling
- Global planning

Checkpointing: a tool for robustness in the presence of transient faults

- Periodically store system state
- On fault detection, roll back to known-good state
- Should system-wide or incremental, as-needed restores be used?
- When should checkpoints be taken?

Electromigration

- Reduce temperature
- Reduce current
- Spatial redundancy

Manufacturing defects

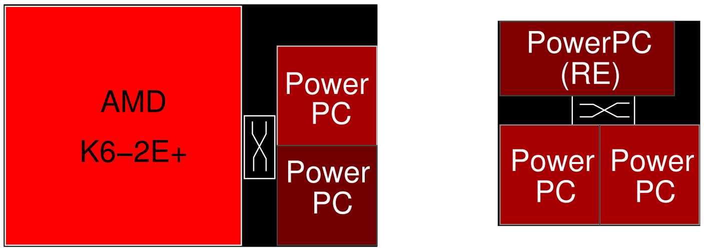
- Spatial redundancy

Example lifetime failure aware synthesis flow

Changyun Zhu, Z. P. Gu, Robert P. Dick, and Li Shang. Reliable multiprocessor system-on-chip synthesis.
In *Proc. Int. Conf. Hardware/Software Codesign and System Synthesis*, pages 239–244, October 2007

- Use temperature reduction and spatial redundancy to increase system MTTF
- System MTTF: the expected amount of time an MPSoC will operate, possibly in the presence of component faults, before its performance drops below some designer-specified constraint or it is no longer able to meet it functionality requirements

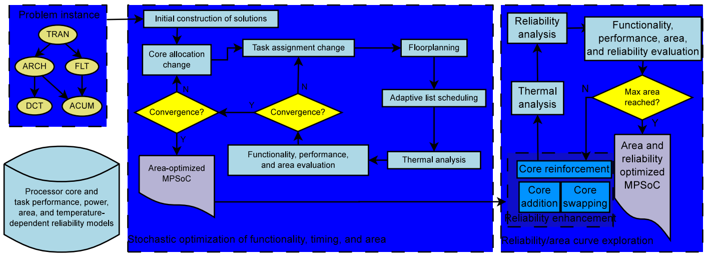
Motivating example for reliability optimization



Solution I

Solution II

Reliability optimization flow


Problem instance
TRAN
ARCH
FLT
DCT
ACUM
Processor core and task performance, power, area, and temperature-dependent reliability models
Initial construction of solutions
Core allocation change
Task assignment change
Floorplanning
Adaptive list scheduling
Thermal analysis
Functionality, performance, and area evaluation
Convergence?
Convergence?
N
Y
N
Y
Area-optimized MPSoC
Stochastic optimization of functionality, timing, and area
Reliability analysis
Functionality, performance, area, and reliability evaluation
Max area reached?
N
Y
Thermal analysis
Core reinforcement
Core addition
Core swapping
Reliability enhancement
Area and reliability optimized MPSoC
Reliability/area curve exploration

Lifetime reliability optimization challenges

- Accurate reliability models
- Efficient system-level reliability models
- Efficient fault detection and recovery solutions
- Optimization

Importance of understanding fault class

- Many reliability techniques attempt to deal with arbitrary fault processes
- However, the properties of the fault process most significant for a particular appliation may be important
  - Considering them can allow more efficient and reliable designs

What to do before Monday

1. Adjust your project definition based on customer interviews so far and prepare a page-long description of why it is valuable and how it will be prototyped and evaluated.
2. Complete at least another five interviews of people who might value what you are trying to provide and take detailed notes.