Automated construction of Markov models for redundant sub-designs, enabling fast reliability assessment of design candidates

- Individual components are modeled by their rate of permanent ($\lambda_P$) and temporary ($\lambda_T$) failures
- Recoveries from temporary failures are considered by rate $\mu$.

**Abstract** — This thesis presents an automated toolset for exploring design choices which provide fault tolerance by means of hardware redundancy. Based on a given VHDL model, various fault tolerant implementations can be automatically created and evaluated regarding their overhead and reliability improvement.

**Contributions**

- Automated insertion of static and dynamic hardware redundancy including the required administrative logic into a given VHDL model
- Evaluation of costs and benefits of resulting fault tolerant design candidates
- Reliability modeling considering transient and permanent faults
- Estimation of costs in terms of area, power and performance using state of the art synthesis tools
- Fast predictions of synthesis results, offering a trade-off between runtime and accuracy

**Exploring the Design Space**

- Evaluation of various redundancy strategies for four ALUs in a superscalar processor

**Reliability Modeling**

- Automated construction of Markov models for redundant sub-designs, enabling fast reliability assessment of design candidates
- Individual components are modeled by their rate of permanent ($\lambda_P$) and temporary ($\lambda_T$) failures
- Recoveries from temporary failures are considered by rate $\mu$.

Example: Markov model for TMR

\[ LIF = \frac{MTTF_{FT}}{MTTF_{ORG}} \]

\[ RIF(t) = 1 - \frac{1 - R_{ORG}(t)}{1 - R_{FT}(t)} \]

\[ State = i \iff i \text{ functioning units} \]

\[ R(t) = 1 - Prob(State(t) = i) \]

**Cost Estimation Methods**

1. Sub-design synthesis (reference method):
   - Synthesize generated VHDL model of a fault-tolerant sub-design
2. Component synthesis:
   - Infer which administrative components are required and only synthesize these
   - Build a database of results for typical components (e.g., multiplexers with varying structural parameters)
   - Reuse results for other instances of this component (in the same or other design candidates)
3. Result interpolation:
   - After a sufficient number of results are available, properties of new components may be derived by interpolation

**Accuracy and Runtime Trade-off**

- Evaluation for 1000 randomly generated design candidates
- Component synthesis:
  - Mean estimation error of 1.4% (area) and 2.6% (power)
  - 13x speedup over reference method (increasing with number of candidates)
- Additional error by result interpolation:
  - Depending on size of result database
  - Insignificant after component synthesis for ~50 candidates
  - 70x speedup in this case