UCLA

ERSA: Error-Resilient System Architecture

Liangzhen Lai

L. Leem, H. Cho, J. Bau, Q. A. Jacobson, and S. Mitra. ERSA: Error resilient system architecture for probabilistic applications. In DATE, 2010.

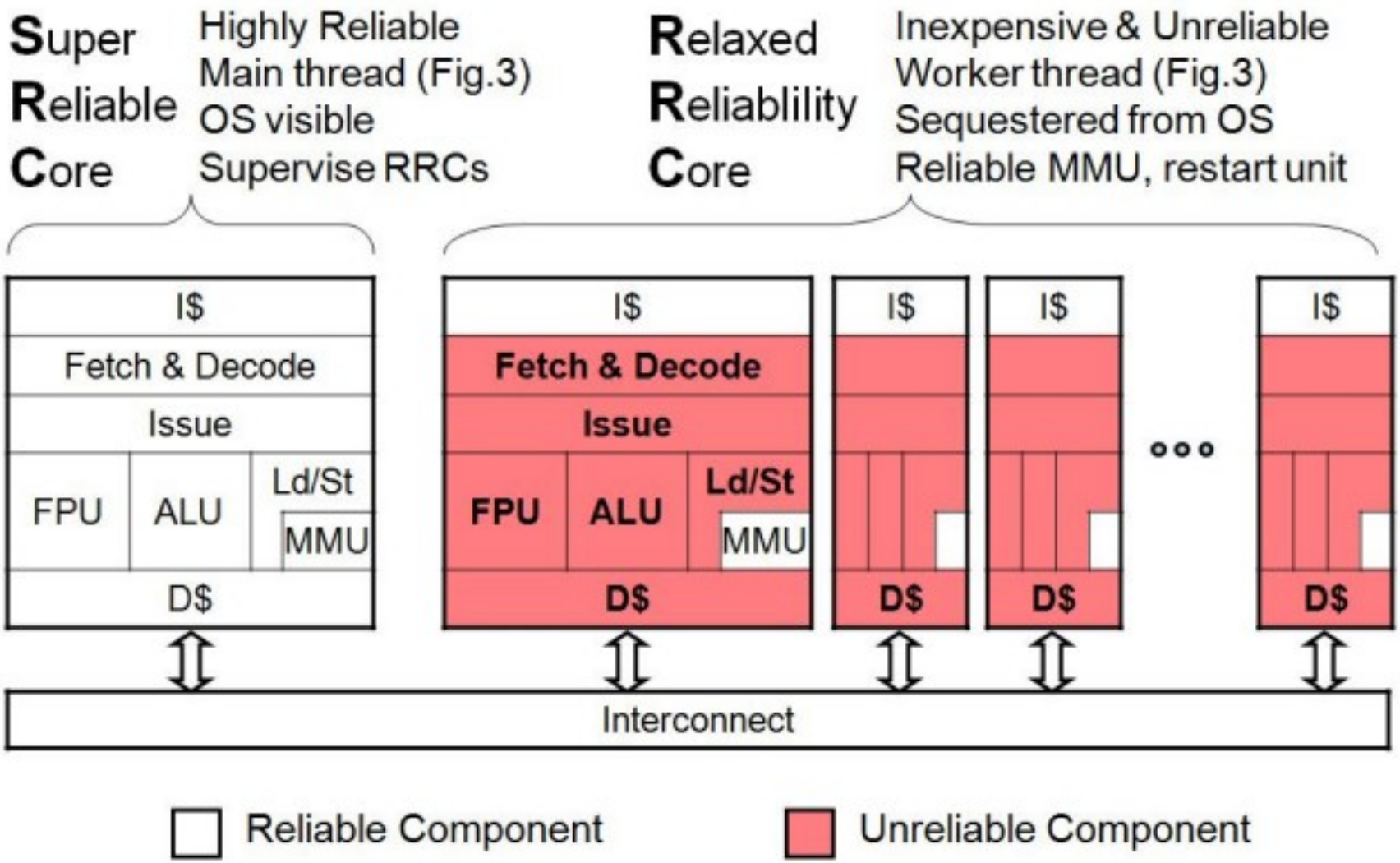
Outline

- Probabilistic Applications
- ERSA Overview
- SRC and RRC
- ERSA experiments

Probabilistic Applications

- Some probabilistic applications such as Recognition, Mining and Synthesis (RMS) applications have the following properties:
  - Massive parallelism
  - Algorithmic resilience (e.g. iterative refinement, relying on convergence)
  - Cognitive resilience (e.g. qualitative results,)
- Key Challenges:
  - Control flow is hardly error-tolerant
  - Asymmetric tolerance: low-order bits vs. high-order bits
  - Surviving from high error rates

Hardware Architecture



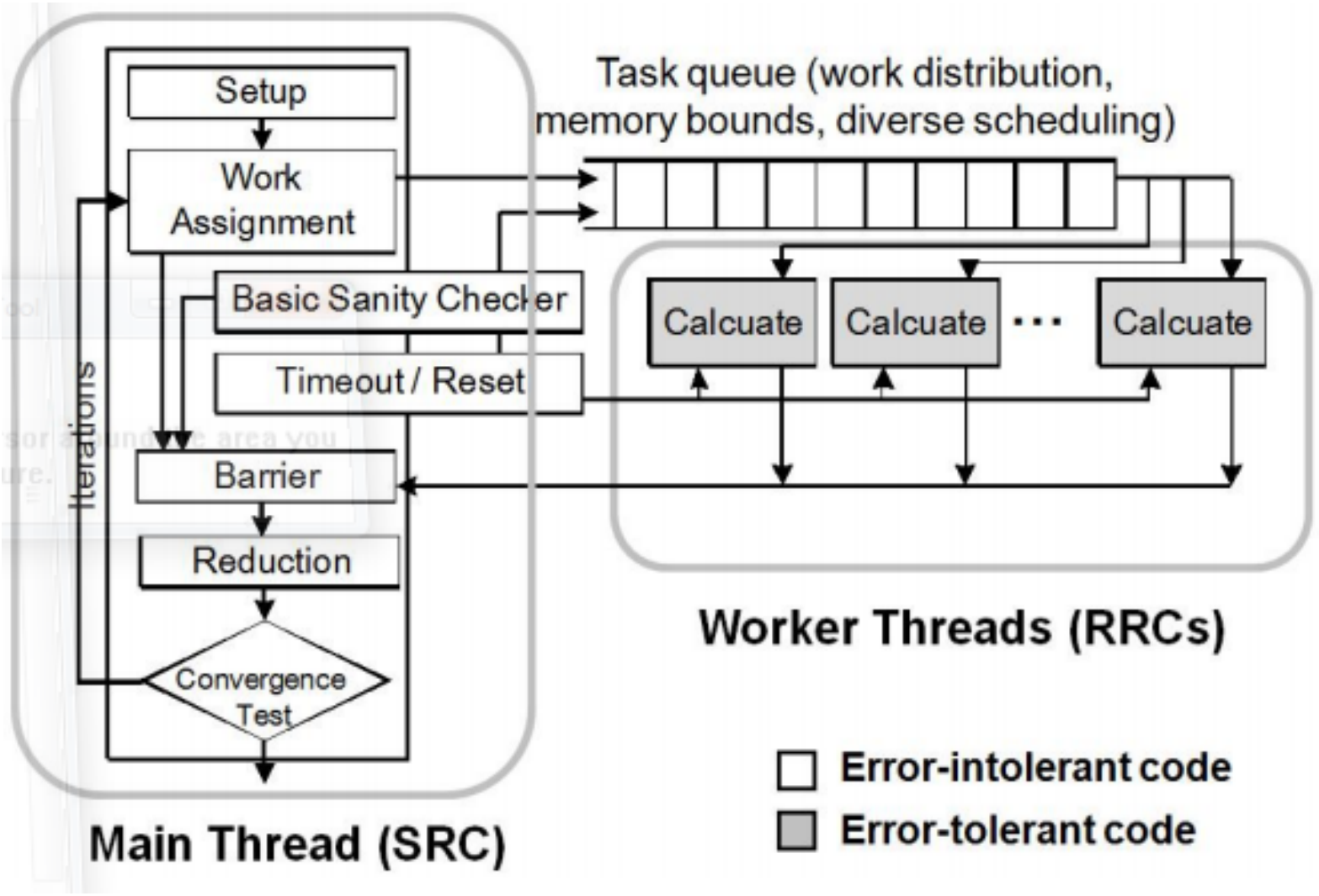
Super Reliable Core (SRC)

- An SRC is responsible for:
  1. Executing non-error-tolerant codes
     - OS
     - Application main thread
  2. Supervising RRC
     - Workload distribution
     - Sanity checks
     - Timeout / Reset
     - Computation results checking

Relaxed Reliability Core(RRC)

- RRC is the main execution units that can be unreliable
  - A reliable memory management unit is used to detect memory access bound violations

Computation Model



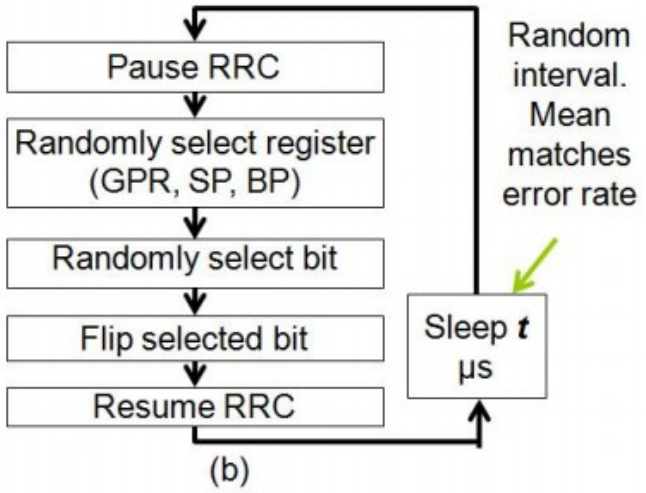
Software Optimization

- Convergence Damping
  - If Δ > threshold, let Δ = threshold
- Convergence Filtering
  - If Δ_i > threshold, discard Δ_i

Experiment Platform

- 2 Processor cores in FPGA
  - One for SRC
  - One for RRC with time-multiplexing
- Bit-error in injected randomly in registers
  - 32 general purpose registers
  - Stack and base pointers

(a)



(b)

Results

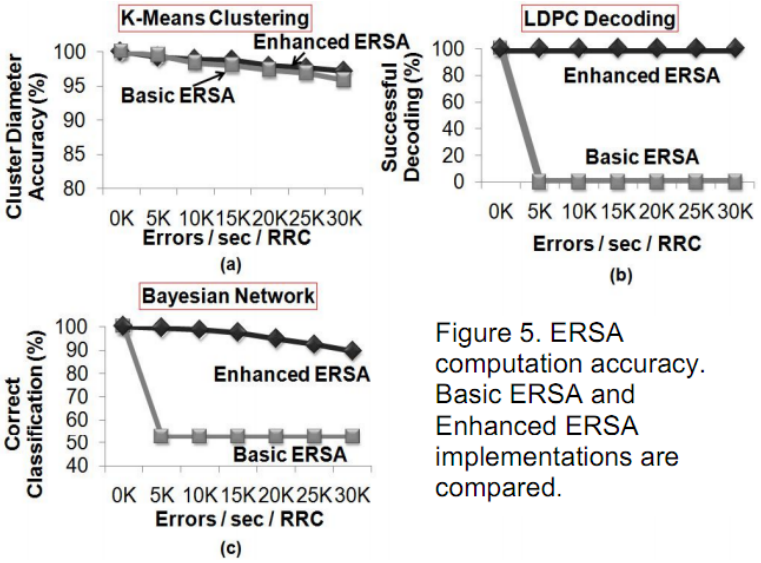


Figure 5. ERSA computation accuracy. Basic ERSA and Enhanced ERSA implementations are compared.