Bertacco Lab
Designing correct heterogeneous systems

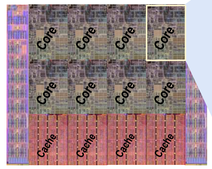
ReDEEM: A Heterogeneous Distributed Microarchitecture

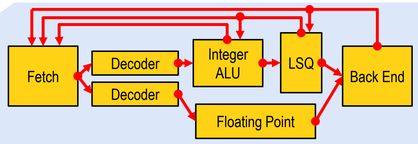
Biruk Mammo, Ritesh Parikh and Valeria Bertacco

EECS ECE CSE — MICHIGAN

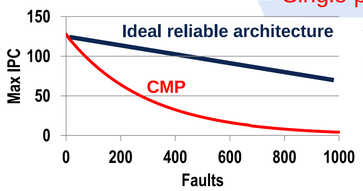
1. Motivation

Traditional CMP





- Rigidly connected hardware modules
- Centralized control logic
- Single points of failure



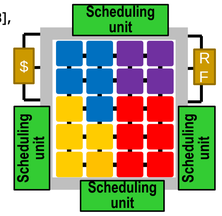
Traditional CMP degrades poorly as fault rates increase

… won't survive high fault rates expected at future technology nodes

2. Decentralized µ-architectures

- **Highly reliable**
  - Fine-grained redundancy
  - Robust interconnect
  - No single-points-of-failure
- **Self-configurable at runtime**
  - Pipelines dynamically configured at runtime
- **Energy efficiency challenges**
  - Communication latencies: ***performance*** ↓
  - Idle redundant modules: ***leakage power*** ↑

StageNet [Micro08], CCA [PACT08], Viper [ISCA12]



energy efficiency ↓

Rethink decentralized µ-architecutres for improving energy efficiency

3. ReDEEM: Techniques evaluated

✓ **Fine-grained circuit-level heterogeneity:** tacking dynamic power
- Components designed to operate at different frequencies
- Enables energy-performance tradeoffs
- Easily plug in heterogeneous components

✓ **Power- and clock-gating:** tackling idle power
- Suitable for fine-grained switching
- Not enough work to fill all pipes

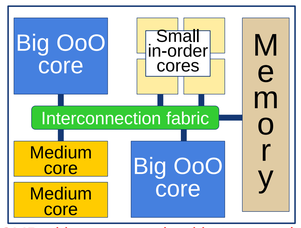
DVFS: to ensure common freq. pipelines
- Not ideal for wide-spread use (diminishing returns)
- Only activated when faults strike

Voltage (V)	28nm Freq (MHz)
1.1	2000
1.0	1800
0.9	1500
0.8	1200
0.7	900
0.6	600

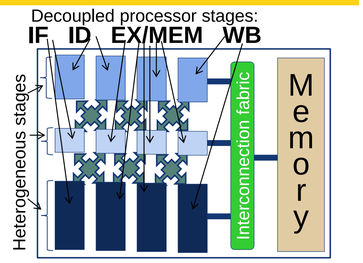
1.8x, 3.3x

Data taken from [Flatresse12]

4. Heterogeneity & power-gating



CMP with coarse-grained heterogeneity



Fine-grained heterogeneity within the core

- Heterogeneous modules are ***architecturally identical*** but synthesized for different VF targets ***(circuit-diverse)***
  - Each module can be power-gated independently
  - Each module can be frequency-scaled to ensure common-frequency pipelines

5. Microarchitecture design

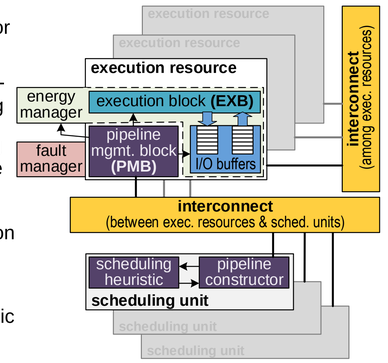
execution resources: heterogeneous components for energy-efficient computing

energy manager: for power-gating and frequency-scaling

fault manager: for detecting and disabling faulty resource

intertconnect: fault-tolerant and low-latency communication between components

scheduling units: monitor performance and build dynamic pipelines



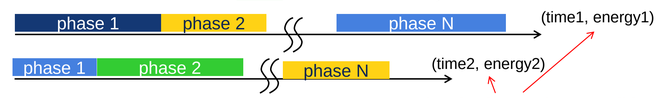
6. Building dynamic pipelines

- Exec. resources ***advertise*** their availability to sched. units
- Sched. units select resources and ***build dynamic pipelines***
- Idle resources go to ***sleep***
  - Can be woken up after they ***recover power-gating costs***

- Application execution grouped into phases and mapped onto different pipelines with different perf./energy characteristics.

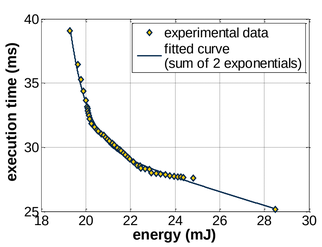
: dynamic pipeline 1 — : dynamic pipeline 3
: dynamic pipeline 2 — : dynamic pipeline 4

diverse perf./energy charac-teristics for different phases

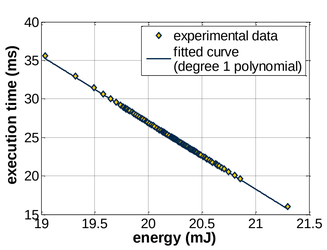


several optimal executions possible

7. Energy-efficiency results



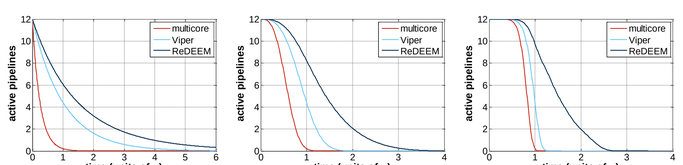
Best case *(gobmk)*: **32%** range in energy savings, **1.55x** range in slowdown



Worst case *(libquantum)*: **11%** range in energy savings, **2.22x** range in slowdown

Up to ***60% total energy savings*** (averaged across all benchmarks) over an equivalent homogeneous decentralized microarchitecture (Viper)

8. Reliability analysis



constant failure rate — gently increasing failure rate — rapidly increasing failure rate

- Statistical model to compare reliability between a multicore, Viper and ReDEEM
  - Each resource has a failure rate defined by a statistical distribution (Weibull)
  - Random variable *(Availability)* to model the number of active pipelines
  - Models shown for different types of failure rates
- ReDEEM is available (with at least one active pipeline) **2.8x** longer than a multicore, and **1.8x** longer than Viper.

9. Conclusions & future work

- ReDEEM is a decentralized µarchitecture that is more reliable and energy-efficient than previously-proposed µarchitectures.
- Scheduling algorithms and communication need to improve to compete with the performance and energy-efficiency of multicores
- Asynchronous pipelines can potentially allow for better energy-efficiency. To be explored.