Energy-Efficient Computing
CORE ARCHITECTURE
Concurrent VLSI Architectures, Stanford University
Milad Mohammadi, Subhasis Das
Bill Dally, Tor Aamodt

A disruption to the energy-performance trend is feasible via architectures with simple hardware and more sophisticated software.

Phraseblock

- CFG is split at back-edges, WBB’s, UPLD’s to form code blocks called Phraseblocks (PB).
- Instructions in a PB are statically scheduled, and execute without memory and control interruption.
- Multiple PB’s are scheduled to run concurrently (PB OoO).

Energy Saving Opportunities

Phraseblock Pipeline

Branch Prediction
- dynamic prediction only for WBB

Rename Logic
- only for global registers

Register Files
- Global (Reg. Renaming), Locals (no Reg. Renaming)

Load-Store Unit
- speculative memory access only across Phraseblocks

Out-of-Order Execution

OoO Energy
- 94% of core energy is on bringing instructions and data to the core as well as doing the bookkeeping to achieve high ILP.

OoO Performance
- dynamism and speculation at instruction granularity deliver high ILP

Coarse-grain OoO Execution

Achieving high ILP demands speculation and dynamism support, but NOT at instruction level granularity. Block-level OoO execution delivers similar performance as OoO with 30% less core energy.

Phraseblock Results

Energy Breakdown

- Caches 40%
-Others 5%
-Decide 4%
-RO& Spec 15%

Fetch & Decode 12%

OoO Processor

- Branch Prediction
- Rename Logic
- Out-of-Order Execution

OoO OoO

- 13%
- 25%
- 43%
- 12%
- 2%

Legacy

- 40%
- 10%
- 20%
- 30%
- 40%

Instruction Cache
- 1.0 cache for fast PB reuse for high frequency Phraseblocks

Phraseblock Windows
- multiple FIFO queues (no associative search)

Phraseblock Scheduler
- only access the head of PB Windows to issue and update instructions

Phraseblock ROB
- must only maintain PB ordering; 10x fewer ROB entries