Open64

A Software Pipelining Framework for Simple Processor Cores

- Juergen Ributzka
- David Stephenson
- Timothy Kong
- Dee Lee
- Fred Chow
- Guang R. Gao

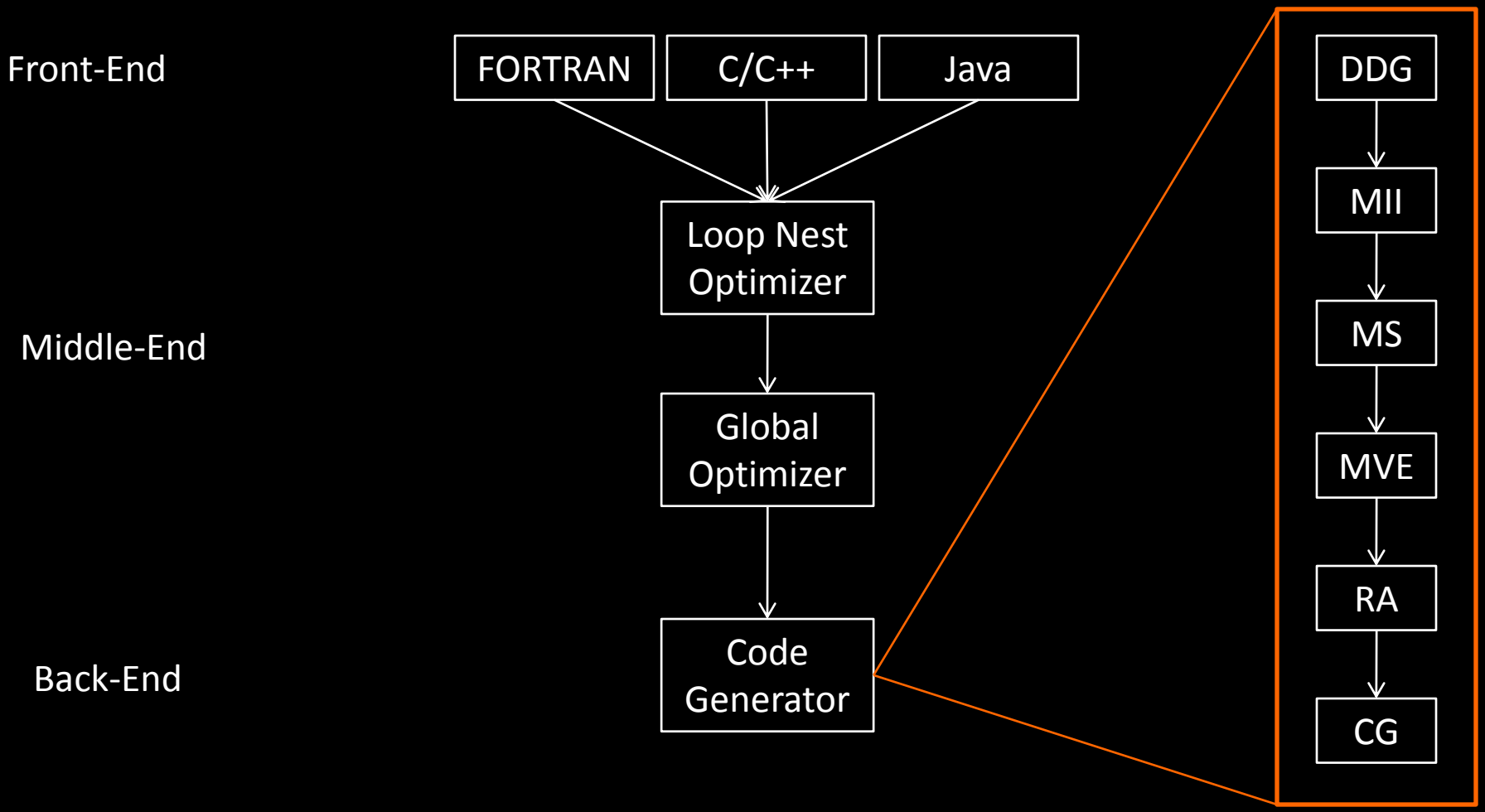
Overview

- SiCortex Multiprocessor
- Software Pipelining Framework
- Results
- Future Work

SiCortex Multiprocessor

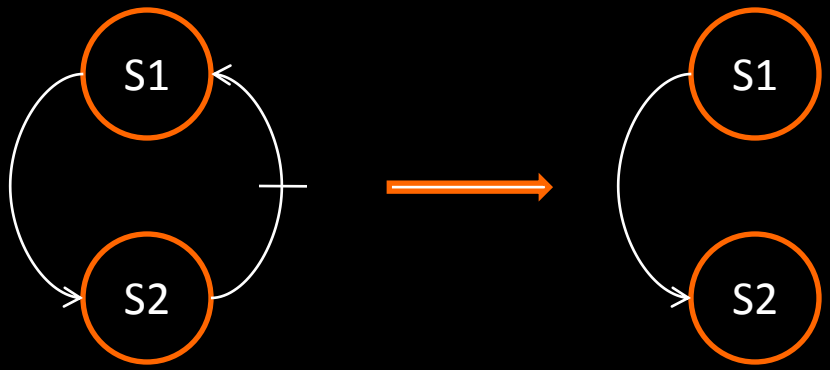
- RISC
- 6 cores (MIPS 5KF)
- 500 – 700 MHz
- In-Order Execution
- Limited-Dual Issue
- Low Power (12 – 16 Watt)

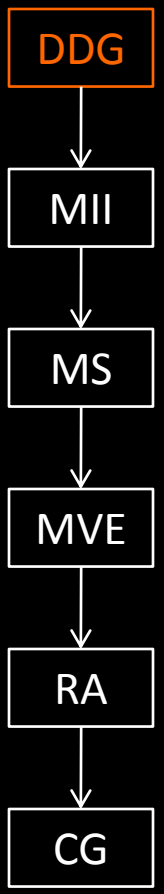
Software Pipelining Framework



Data Dependence Graph (DDG)

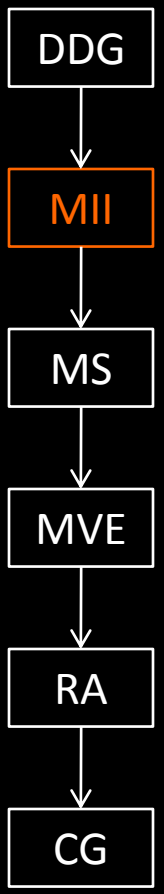
- Cyclic Data Dependence Graph
- No register anti- and output-dependencies
- Only single BBs





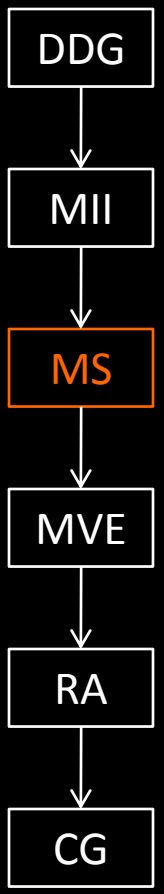
Minimum Initiation Interval

- Limited by:
  - Resource Requirements
  - Recurrence Requirements



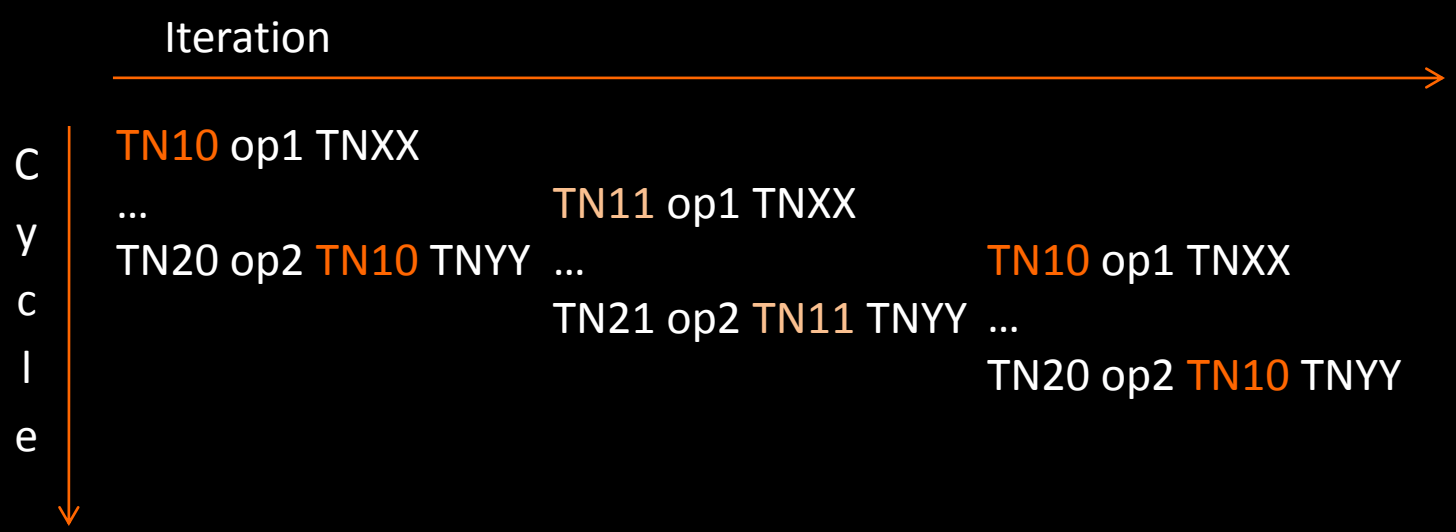
Modulo Scheduling (MS)

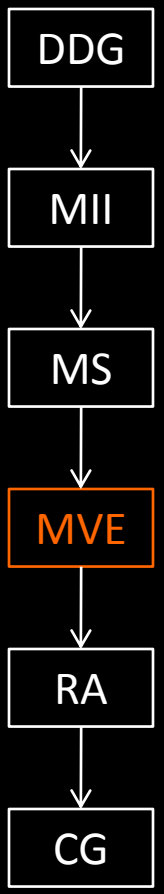
- Huff Modulo Scheduling
  - Lifetime sensitive
  - Uses backtracking
- Hyper Node Reduction Scheduling
  - Lifetime sensitive
  - No backtracking



Modulo Variable Expansion (MVE)

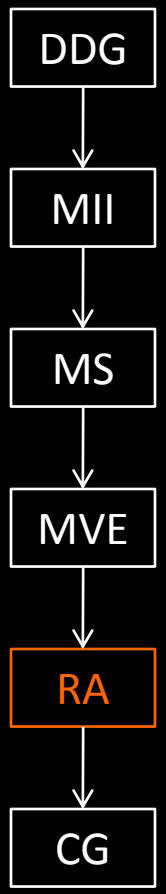
- Separate TN set for every iteration
- # of unrollings depends on longest lifetime





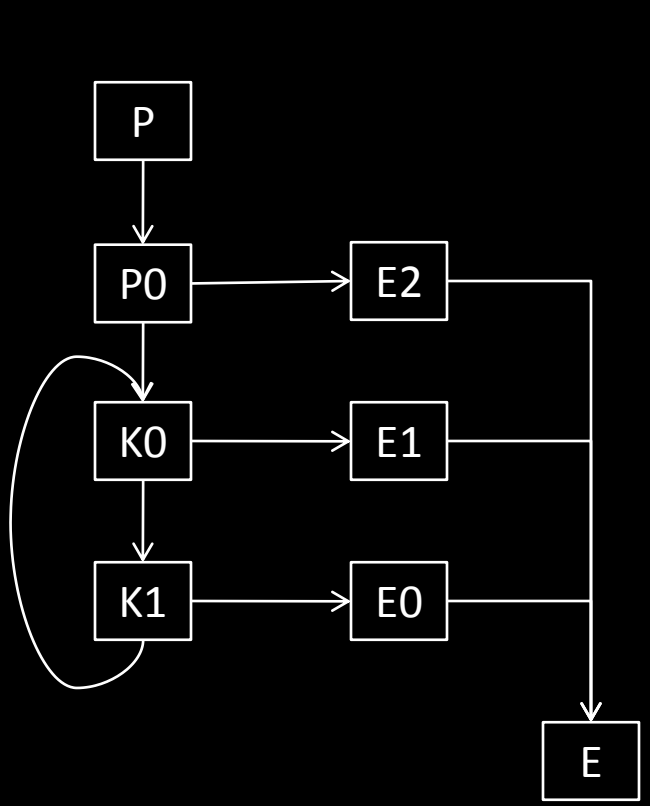
Register Allocation (RA)

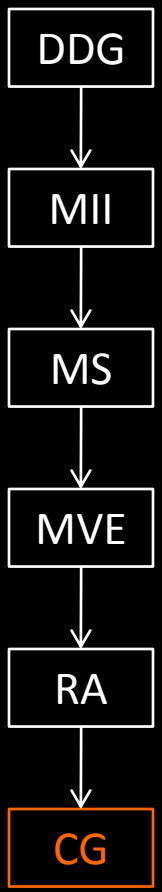
- Only kernels are fully register allocated
- No regions
  - SWP kernels are not a black box for GRA and LRA anymore
- Currently no register spilling support



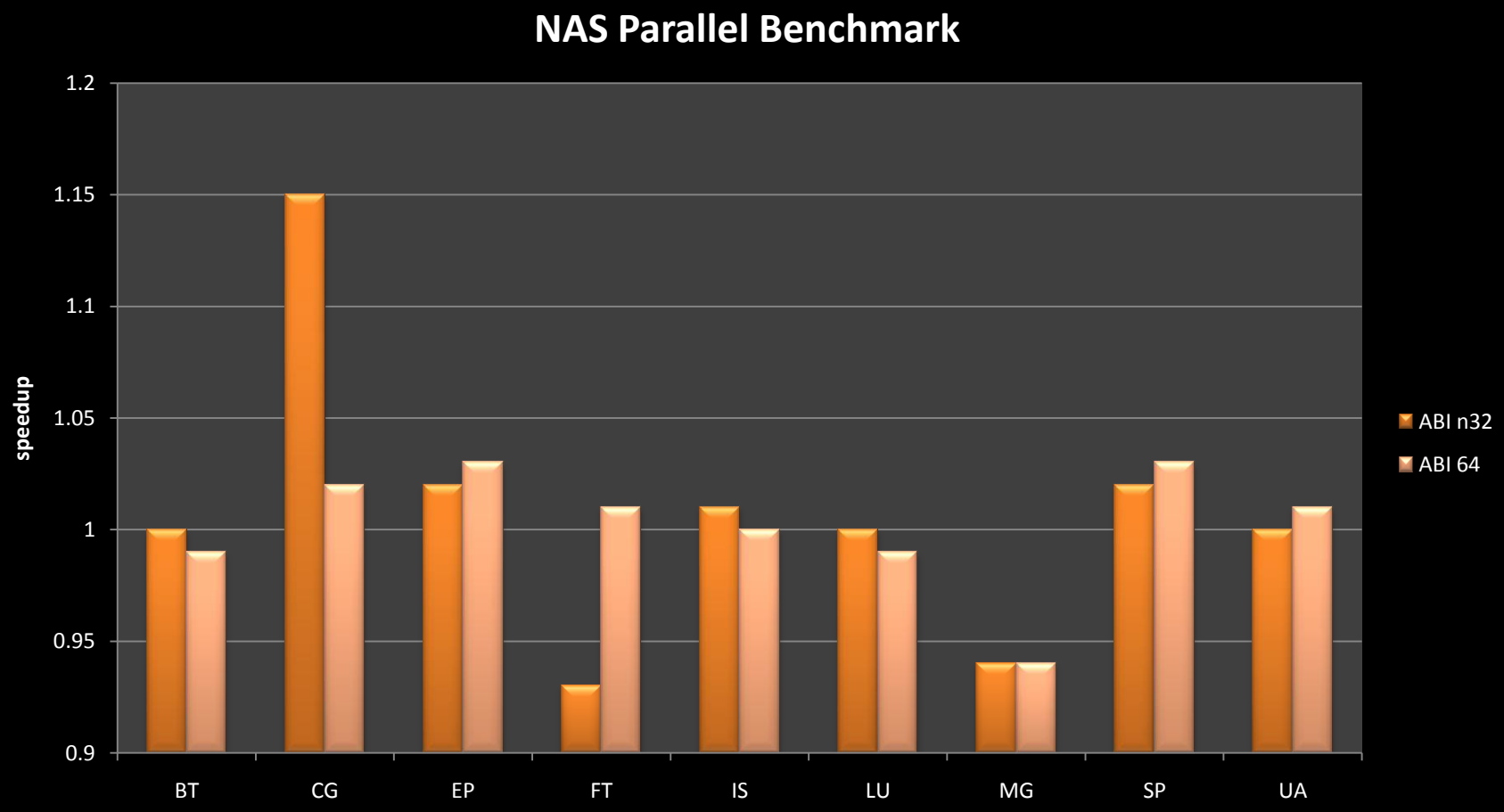
Code Generation (CG)

- Prologues and epilogues are partially register allocated
- Need several epilogues due to different register sets

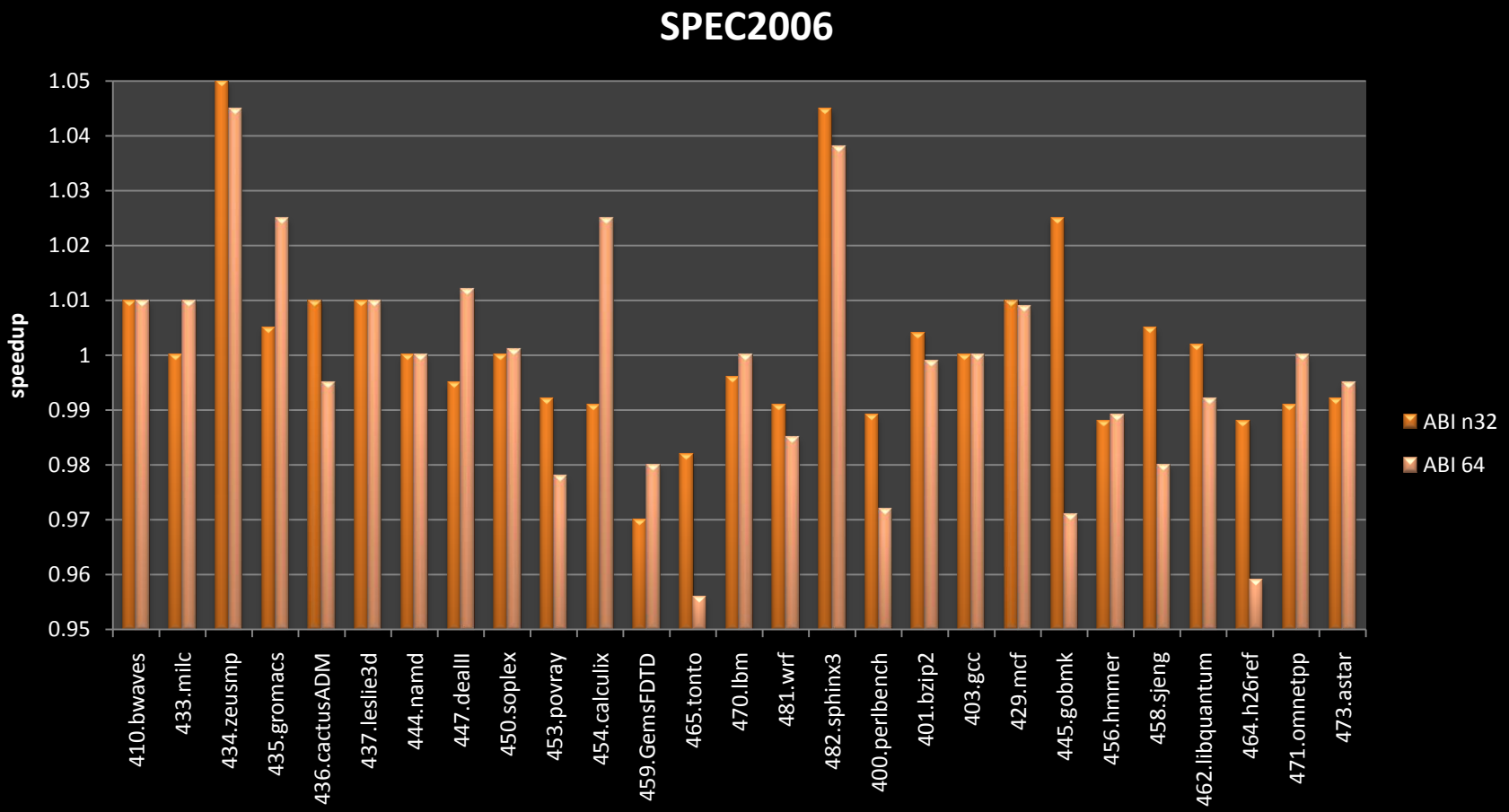




Benchmark Results



Benchmark Results



Open64

Conclusion and Future Work

- Spilling support needed to enable more loops for SWP
- Screen out loops with small trip count during runtime to reduce SWP overhead
- Hit-under-miss support to overcome current hardware limitations

Questions ?