Superscalar Processor with Dynamic Branch Prediction

Computer Science 152 – Final Project
University of California, Berkeley

The Paper Ceiling (PC)
Jack Kang, Benjamin Lee, David Lee, Lyle Takacs

May 15, 2003
Outline

- Two-Way Superscalar Processor
  - Prefetching
  - Instruction Level Parallelism & Dual Issue
  - Hazard Detection & Forwarding

- Dynamic Branch Prediction
  - Branch Target Buffer
  - Branch History Table

- Testing
- Performance
- Future Work
- Summary
Superscalar – Dual Issue (1/2)

- Multi-ported Instruction Cache
  - Fetches up to 4 instructions per cycle

- FIFO Instruction Buffer
  - Prefetches up to 8 instructions
  - Maximizes pipeline utilization

- Issue Unit
  - Performs data dependency analysis
  - Issues up to 2 instructions per cycle
  - Stalls or swaps instructions to resolve control hazards and data hazards not handled by pipeline forwarding
Superscalar – Dual Issue (2/2)
Superscalar - Pipelines

(ID) (EX) (MEM) (WB)

(ID) (EX) (WB)

(...From issue unit)
Superscalar – Forwarding (1/2)

- Forwarding between pipeline stages
  - EX and MEM values forwarded to ID stage for branch comparisons and jumps
  - EX and MEM values forwarded to EX stage for arithmetic, shift, and compare instructions
  - Memory bypass forwards loaded data to proceeding store word word instructions
Superscalar – Forwarding (2/2)

- Forwarding between pipelines
  - EX and MEM values from bottom/top pipeline forwarded to ID stage in top/bottom pipeline
  - EX and MEM values from bottom/top pipeline forwarded to EX stage in bottom/top pipeline
  - EX values from top pipeline “forwarded” into MEM stage in bottom pipeline
Branch Translation Buffer

- 256-entry direct mapped buffer
  - Provides a target address for the PC in the fetch stage
  - Indexed by the PC

- Buffer Entry
  - At most 2 branch instructions per 4 instructions fetched
  - 32-bit tag corresponding to PC of branch instruction
  - 32-bit PC1, PC2 correspond to target addresses for 2 branches
Branch History Table

- 256-entry direct mapped table
  - Provides a taken signal for the PC in the fetch stage
  - Indexed by the PC

- Table Entry
  - 2-bit dynamic branch prediction with hysteresis
Testing Methodology

- **Incremental Testing**
  - Issue Unit – fake instruction cache
  - Superscalar Pipelines – forwarding
  - Branch Translation Buffer – proper PC selection
  - Branch History Table – finite state machine

- **Integrated Testing**
  - Boot loader – ability to run instructions, corner cases
  - Single pipeline forwarding– basic forwarding
  - Superscalar pipeline forwarding– advanced forwarding

- **General Testing**
  - Monitor and console outputs – access to data flow
  - Trace files – instructions, cycle count, time, etc...
Performance (1/2)

- Prefetch and Instruction Buffer
  - Improves instruction throughput
  - Allows for dual issue

- Two-way Set Associative Cache
  - Reduces cache miss rate

- Dynamic Branch Prediction
  - Reduces delay slot penalties
  - Hysteresis for increased prediction accuracy
Performance (2/2)

- **Current Status**
  - Verify and Base tests are fully functional in simulation

- **Hardware Statistics**
  - Number of Block RAMS: 67 out of 160 (41%)
  - Number of Slices: 6669 out of 19200 (34%)

- **Timing Statistics**
  - Critical Path: 69.646 ns
  - Clock Frequency: 14.358 MHz

- **CPI**
  - Lab 6: 1.728 cycles per instruction
  - Final: <to be determined>
Possible Continued Work

- Jump prediction
  - Reduce the penalties of extra delay slots
- Load value prediction
  - Reduce the penalties of extra delay slots
- Separate decode and issue into 2 stages
  - Increase the clock frequency
- Hide memory stalls
  - Reduce the penalties of data cache misses
Summary

- Two-way superscalar processor with dynamic branch prediction and prefetch
  - Design, implementation, verification, presentation in one week
- Significant concepts
  - Modular design with interface specifications
  - Implementation with Schematic vs. Verilog
  - Implementation with different clock frequencies
  - Incremental testing
  - Integrated testing
  - Hardware testing
  - Lab notebook and coordination
  - Division of labor
Acknowledgements

- Jack Kang
  - Design and implementation of the dual issue unit
  - General verification in simulation and board

- Benjamin Lee
  - Design and implementation of BHT and BTB
  - Data path assembly
  - General verification in simulation

- David Lee
  - Design and implementation of the dual issue unit
  - General verification in simulation and board

- Lyle Takacs
  - Design and implementation of forwarding unit
  - Modifications to monitor module
  - Data path assembly
  - General verification in simulation
Acknowledgements

- John Kubiatowicz, Professor
  - Design consultation in office hours
  - Reference material in lecture notes
- Henry Lam, Section TA
  - Design consultation in section
  - Check-offs
- Jason Ding, Section TA
  - Check-offs
- Ben Liao, Project Development TA
  - Assistance with development tools
  - Assistance with hardware
Questions