PSATSim

An Interactive Graphical Superscalar Architecture Simulator for Power and Performance Analysis

> Clint W. Smullen, IV Tarek M. Taha

Clemson University Department of Electrical and Computer Engineering

WCAE 2006

Purposes

- I. Tool for instructors
 - Demonstrate superscalar architectures
 - Use in-class
- 2. Framework for students
 - Explore the power and performance
- 3. Interactive execution
- 4. Wide range of configuration options

Power Modeling

Uses the Wattch power model
D. Brooks, V. Tiwari, and M. Martonosi 2000
High-level modeling of major components

Power Modeling

- Tracks activity use of each component
- Average activity use scales maximum energy consumption
- Averages the sum of component energy usage over length of execution

Capabilities

Uses SimpleScalar ISA
 Related to MIPS ISA
 Easy to understand instruction format
 Statistically models branch misspeculation
 This improves accuracy of power model

Capabilities

- Statistically models cache hierarchy
- Uses trace files
 - SPEC benchmark traces are provided with program
 - Reduces overhead in demonstrations
 - Shortens iteration latency for students

Execution Interface

-	PSATSim	
Eile <u>C</u> ycle <u>H</u> elp		
🗅 × 🗶 🖨 🐉 🄇		
 ✓ Reorder Buffer 13 I.d F0 IMM R2 	Cycle: 15 Committed: 13 IPC: 0.867 Fetch Latency: 0 I-Cache Misses: 1 D-Cache Misses: 0 Branches: 2 Misspeculated Branches: 0	
14 mul.d F2 F2 F0	Fetch: 28 525540 29 525541	30 525542
15 I.d F0 IMM R5	Decode: 25 I.d 26 div.d	27 addiu
16 sub.d F0 F0 F2	R2 R9 IMM R2 R2 R13	F2 IMM R2
17 addiu R4 R4 IMM	▷ Register Mappings	
18 addiu R6 R6 IMM	✓ Rename Table	
19 slti R2 R6 IMM	0 13 1 15 2 18 3 19 4 16 5	17 6 7 14
20 s.d F0 R5	Execution	
21 bne R2 R0		s.d
		bne I.d
	19 18 14 12 13 16 15 14	19 15
	Integer Integer Floating Floating	Branch Memory
	FU 19 14	
		15
	Commit: 13 I.d	

In-Order Front-end

Fetch:	28		525540	29	5	25541	30		52.	5542
Decode:	25		l.d	26		div.d	27		а	ddiu
Dispatch:	22		sll	23		addu	24			l.d
	R2	R9	IMM	R2	R2	R13	F2	IMI	V I	R2

Misspeculated instructions displayed with strikethrough

Fetch:	84	525531	85	525532	86	525533
Decode:	81	sub.d	82	addiu	83	addiu
Dispatch:	80	bgtz				
	R8	3 IMM				

Coloration



• Makes it easy to see dependencies

Renaming Table

⊲ Rer	name Ta	able													
0	520	1	528	2	529	3	516	4	517	5	518	6	519	7	521
8	522	9	523	10	524	11	525	12	531	13	532	14	533	15	

• Provides false hazard resolution

 Instructions without color have already produced a value

Reorder Buffer



13

.c

- Provides in-order completion of instructions
- Uncolored opcodes have finished and await commit
- Up to the superscalar width in instructions are committed each cycle

Commit:

Reservation Stations

Distributed:

Centralized:

Hybrid:





RS slti 103 104 RS addiu sub.d l.d 103 99 100 10 100 RS addiu mul.d .d 99 96 102 98 98 Integer Integer Floating Floating Branch Memory

Functional Units



IAdd IAdd	IMult	IDi∨	FPAdd	FPMult	FPDi∨	FPSqrt	Branch	Load	Store
FU 115 116								113	
FU								111	

Configuration

New Simulation	New Simulation
General Execution Memory/Branching	General Execution Memory/Branching
3 🛉 Superscalar Factor (1-16)	Execution Unit Architecture Standard 🗢
8 🚔 # of Rename Entries (1-512)	Reservation Architecture: Distributed 🗢
12 🚔 # of Reorder Entries (1-512)	2 A # of Entries per Reservation Station (1-8)
Separate Decode and Dispatch	
Enter the path for the trace file:	2 🗘 # of Integer Execution Units (1-8)
traces/applu.tra Browse	2 🚔 # of Floating Point Execution Units (1-8)
Enter the path for the output file:	1 🚔 # of Branch Execution Units (1-8)
output.xml Browse	1 🚔 # of Memory Execution Units (1-8)
Den □Save ▲ Cancel ▲ Apply	©pen <u>S</u> ave X Cancel √ Apply

Interactive Use



- User can force a branch misspeculation
- Single- and auto-step through the execution
- Pause automated execution
- Quickly finish execution

Project Use

- Use of traces gives shorter simulation time
- Wide range of architectural options
- Exploration within a given set of constraints

Simulation Results

Instructions: 71771

Cycles: 47363

Power: 21.8652 W

IPC: 1.51534

Execution Time: 78938.3 ns

(Cycle time = 1.66667 ns)

X Close



Use in the Classroom

- Used in the undergraduate and graduate courses at Clemson
- Used for demonstration
- Students are asked to maximize performance within a given power envelope

Implementation

Written in C++
Uses GTK+2
LibXML2, LibPCRE, PThreads
I4 K lines code

Availability

 Software available from: <u>http://www.ces.clemson.edu/~tarek/psatsim/</u>

Currently available for Windows
 Linux version should be available soon

