# TDTS08: Advanced computer architecture

Lesson

2010

LiU

expanding reality

## Outline

➜ **Lab organization and goals**

▪ **SimpleScalar architecture and tools**

▪ **Assignment on multiprocessor systems**

▪ **Break**

▪ **Exercises**

2

## Organization

- **Assistants**
  - **Group A: Soheil Samii**
  - **Group B: Ke Jiang**
  - **Group C: Jakob Rosén**
- **Web page**
  - **http://www.ida.liu.se/~TDTS08**
  - **Check the lab pages**

3

## Organization

- **Sign up in Webreg (you have already done that!)**
- **Deadline for all lab assignments: January 9, 2011**
- **Rules: Read them! (linked from the lab pages)**

4

Linköpings universitet

## Examination

- **Written report for each lab**
  - **Hand in the report enclosed in a lab cover; it must be signed by all group members**
    - **Hand in at a lab session**
    - **Put in the box outside your assistant's office**

  - **Returned in the box outside your assistant's office**

5

## Labs

- **5 lab assignments**
  1. **Cache memories**
  2. **Pipelining**
  3. **Superscalar architectures**
  4. **VLIW processors**
  5. **Multiprocessor systems**

- **Labs homepage**
  - **http://www.ida.liu.se/~TDTS08/labs**

6

## Goals

- **Obtain knowledge about computer organization and architectures**
- **Insights in various trade-offs involved in the design of a processor**
- **Become familiar with a set of tools necessary for evaluation of computer architectures**
    - **Simulation tools!**

TDTS08: Advanced Computer Architec...

7

## Environment

- **Unix**
- **Simulations are started from the command line**

! **If you are not familiar with the Unix environment:**
- **Search the Internet**
    - **Tutorials**
    - **List of basic commands**
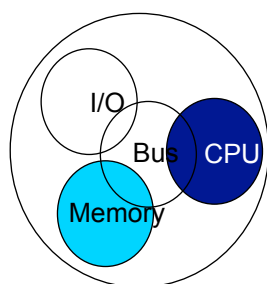- **Make sure that you learn the basic commands in order to be able to work in a command-line environment**

8

## Outline

- **Lab organization and goals**
- → **SimpleScalar architecture and tools**
- **Assignment on multiprocessor systems**
- **Break**
- **Exercises**

9

## Introduction

I/O

Bus  CPU

Memory
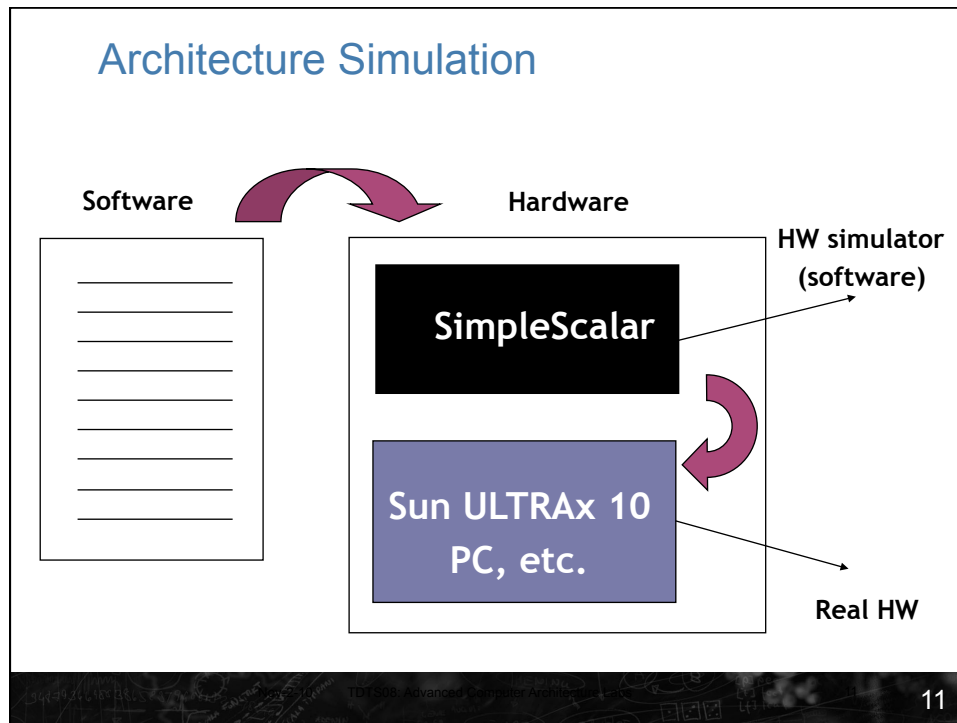
Issues covered in our lab:

- Cache memories
- Instruction pipelining
- Superscalarity
- VLIW (Very Long Instruction Word) processors
- Multiprocessor and Multi-computer systems

10

## Architecture Simulation

Software      Hardware

**SimpleScalar**

HW simulator (software)

**Sun ULTRAx 10 PC, etc.**

Real HW

11

---

## SimpleScalar: Literature

- **"The SimpleScalar Tool Set, Version 2.0" by Doug Burger and Todd M. Austin**
    - **Very important!! Read it as a preparation for the labs**
    - **Used all the time during all labs!**

- **User's and Hacker's guide (slides by Austin)**

- **Linked from the lab pages**

12

---

# SimpleScalar Architecture

- **Virtual architecture derived from MIPS-IV**
- **SimpleScalar ISA semantics are a superset of MIPS**
  - **Control (j, jr,..., beq, bne,...)**
  - **Load/Store (lb, lbu, ...)**
  - **Integer Arithmetic (add, addu, ...)**
  - **Floating Point Arithmetic (add.s, add.d, ...)**
  - **Miscellaneous (nop, syscall, break)**
- **little/big-endian instruction set definition**

13

# SimpleScalar Architecture (cont'd)

Instruction encodings (64 bits)

|  | 16-annote | 16-opcode | 8-rs | 8-rt | 8-rd | 8-ru/shamt |
|---|---|---|---|---|---|---|
| Register format | | | | | | |

|  | 16-annote | 16-opcode | 8-rs | 8-rt | 16-imm |
|---|---|---|---|---|---|
| Immediate format | | | | | |

|  | 16-annote | 16-opcode | 6-unused | 26-target |
|---|---|---|---|---|
| Jump format | | | | |

14

## SimpleScalar Architecture (cont'd)

- **Registers**
    - **32 integer registers + PC, HI, LO**
    - **32 floating-point registers + FCC**
- **Virtual memory:**
    - `0x00000000 – 0x003fffff`    **unused**
    - `0x00400000 – 0x0fffffff`    **text (code)**
    - `0x10000000 – ..........`    **data**
    - `.......... – 0x7fffc000`    **stack**
    - `0x7fffc000 – 0x7ffffff`    **Args and Env**

TDTS08: Advanced Computer Architec

15

## SimpleScalar: tools

- **Several simulators**
    - <u>**Sim-fast:**</u> **Fast, only functional simulation (no timing)**
    - <u>**Sim-safe:**</u> **Sim-fast + memory checks**
    - <u>**Sim-cache:**</u> **Sim-safe + cache simulation and various timing properties (simulation time, measured time, …)**
    - <u>**Sim-cheetah:**</u> **Simulation of multiple cache configurations**
    - <u>**Sim-outorder:**</u> **Superscalar simulator**

16

## SimpleScalar Tools (cont'd)

- **Tool set installed in ~TDTS08/bin**

- **Configurable through command-line arguments or files (recommended):**
  - **-dumpconfig <filename>**
  - **-config <file-name>**

- **gcc cross-compiler available for generating binaries to be executed on SimpleScalar**
  - **Binaries have been generated and are available in the course directory**

17

## Demonstration

1. **Autoset the path**

```
gedrix <348> echo setenv PATH \"\$PATH\":/home/TDTS08/bin >> .login
```

2. **Assignment 2**

a) Copy the files from the course folder

```
gedrix <319> mkdir lab1
gedrix <320> cd lab1/
gedrix <321> cp ~TDTS08/www-pub/labs/cache_memories/2/* ./
gedrix <322> ls
cache1.cfg  Makefile    test1.ss    test2.ss
cache2.cfg  test1.c     test2.c
gedrix <323> []
```

(use "cp -r" to copy the whole contents of a directory, including all subdirectories)

18

## Demonstration

**b) Run simulation of *test1.ss* with configuration *cache1.cfg***

```
gedrix <354> sim-cache -config cache1.cfg test1.ss
```

**c) You will get the following output**

```
sim: ** simulation statistics **
sim_num_insn            505292260 # total number of instructions executed
sim_num_refs             87822850 # total number of loads and stores executed
sim_elapsed_time               89 # total simulation time in seconds
sim_inst_rate         5677441.1236 # simulation speed (in insts/sec)
dl1.accesses          87823201.0000 # total number of accesses
dl1.hits                 87022637 # total number of hits
dl1.misses                 800564 # total number of misses
dl1.replacements           799540 # total number of replacements
dl1.writebacks             274082 # total number of writebacks
dl1.invalidations               0 # total number of invalidations
dl1.miss_rate              0.0091 # miss rate (i.e., misses/ref)
dl1.repl_rate              0.0091 # replacement rate (i.e., repls/ref)
dl1.wb_rate                0.0031 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate               0.0000 # invalidation rate (i.e., invs/ref)
```

19

## Demonstration
### 3. Assignment 3

**a) Dump the default configuration of *sim-cheetah* to *config -file* and look into the default configuration**

```
gedrix <355> sim-cheetah -q -dumpconfig config-file
```
**Part of the configuration file:**
```
# reference stream to analyze, i.e., {inst|data|unified}
-refs                   data

# replacement policy, i.e., lru or opt
-R                      lru

# cache configuration, i.e., fa, sa, or dm
-C                      sa

# min number of sets (log base 2, line size for DM)
-a                      7

# max number of sets (log base 2, line size for DM)
-b                      14

# line size of the caches (log base 2)
-l                      4

# max degree of associativity to analyze (log base 2)
-n                      1
```

20

## Demonstration

**b) Modify the configuration file according to the instructions using any text editor you prefer**

**c) Run sim-cheetah with modified config-file**

```
gedrix <358> sim-cheetah -config config-file ~TDTS08/spec95-big/go.ss 3 7
```

**d) Sample results:**

```
libcheetah: ** end of simulation **
Addresses processed: 394081
Line size: 16 bytes

Miss Ratios


              Associativity
              1             2
No. of sets
128           0.122619      0.096127
256           0.109566      0.093854
512           0.102443      0.092651
1024          0.097429      0.085947
2048          0.083881      0.064152
4096          0.065504      0.061119
8192          0.061401      0.058600
16384         0.059216      0.056511
```
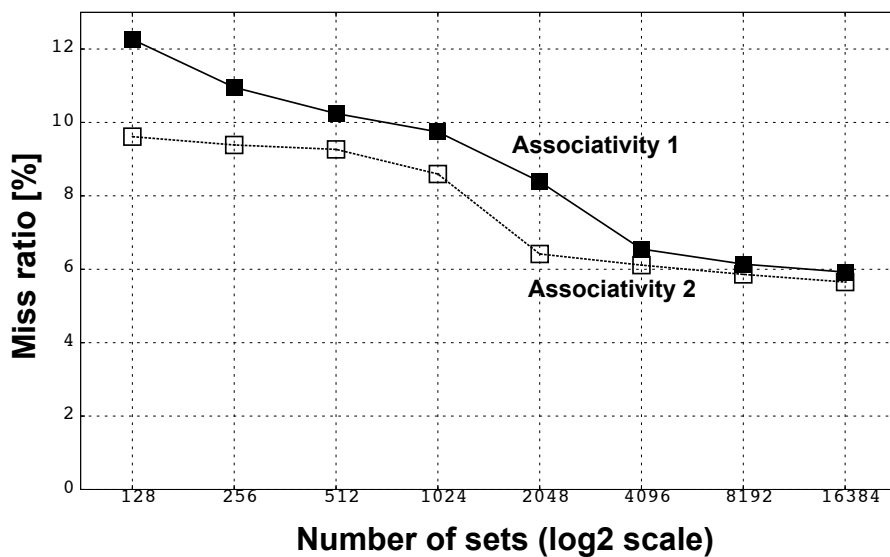
- **Plot the results**
  - **OpenOffice**
  - **Gnuplot**
  - **Matlab**
  - **Excel**
  - **Your favorite plotting tool**

21

## Demonstration



22

## Outline

- **Lab organization and goals**
- **SimpleScalar architecture and tools**
→ **Assignment on multiprocessor systems**
- **Break**
- **Exercises**

23

## Lab 5: Multiprocessor Systems

- **Assignment:**
  - **Select an article on a multi-core, multiprocessor, multi-computer system, or a graphics processor**
  - **Review the selected article**
  - **Write a review report on the article**

- **List of papers is available on the course page**
  - **You may select other articles if your lab assistant agrees**

24

## Multiprocessor Systems (cont'd)

- **Read and understand the paper**
  - **If the course literature does not help you, investigate the referenced papers.**
  - **Searching the Internet can help you find explanations of abbreviations and terms**

25

## Multiprocessor Systems (cont'd)

- **Analyze the paper**
  - **Classify the architecture (MIMD, SIMD, NUMA)**
  - **Possible questions to ask:**
    - **Why has the actual method/approach been selected?**
    - **What are the advantages and disadvantages?**
    - **What is the application area?**
    - **What has been demonstrated?**
    - **…**

26

## Multiprocessor Systems (cont'd)

- **Write a summary**
  - **~1000 words**
- **Submit, in PDF format, to your lab assistant**
  - **sohsa65.liu@analys.urkund.se (Soheil Samii)**
  - **kejiang.liu@analys.urkund.se (Ke Jiang)**
  - **jakro62.liu@analys.urkund.se (Jakob Rosén)**

27

## Outline

- **Lab organization and goals**
- **SimpleScalar architecture and tools**
- **Assignment on multiprocessor systems**
- ➜ **Break**
- **Exercises**

28

## Exercises

- **We have selected some assignments from the 8th edition of the course book (Stallings)**
  - **Review questions (page 146)**
    - **4.4 and 4.8**
  - **Problems (pages 147–149)**
    - **4.8 (preparation for lab 1, mandatory)**
      - **Include your solution for this assignment in the report for lab 1**
    - **Additional exercises in this order:**
      - **4.15 (locality)**
      - **4.22 (average memory-access time)**
      - **4.17 (replacement policy)**

29