Welcome to CS152 Computer Architecture and Engineering

Section 1
1/25/2013
Yunsup Lee
• yunsup@eecs
  – Please add [CS152] in the subject
  – Will try to get back in a day
  – Please post questions regarding lecture, problem sets, labs, and quizzes on piazza

• Piazza
  – I only see 38 enrolled

• Sections
  – Fridays, 10:30AM-noon, 9 Evans

• Office hours
  – Tuesdays, 1-2PM (right before class), 751 Soda
• Class account forms
• Github.com ID
  – If you don’t have one please sign up at github.com
• Signup sheet
Power of abstraction

UCB EECS Courses
- CS170
- CS164
- CS162

Application
Algorithm
Programming Language
Operating System/Virtual Machines
Instruction Set Architecture (ISA)
Microarchitecture
Gates/Register-Transfer Level (RTL)
Circuits
Devices
Physics

315° NW
Brief introduction of what’s below the Microarchitecture abstraction layer

- Transistors
- INVERTER, NAND, AND, NOR, OR gates
- Tri-state gates
- 2:1, 4:1 Muxes (Multiplexers)
- Latches, Flipflops (Registers)
- Register files
- Adders (Half adders, Full adders)
- RAMs (SRAM, DRAM)
6 Cores @ 5.5 GHz

Special-purpose coprocessors on each core

48MB of Level-3 cache on chip

32nm SOI Technology
2.75 billion transistors
23.7mm x 25.2mm
15 layers of metal
7.68 miles of wiring!
10,000 power pins (!)
1,071 I/O pins

[From IBM HotChips24 presentation, August 28, 2012]
Microcode Machine: Control (Start)

- Machine
  - Control
    - (Start)

- Control Signals (17)
- Control ROM
  - address
  - data
- μPC (state)

- Control Signals (17) → next-state encoding reduces ROM width
Microcode Machine: Control (N)

- Control Signals (17)
- Control ROM
- Next-state encoding reduces ROM width

\[ \mu PC \rightarrow \mu PC+1 \]

\[ \mu PC \text{ (state)} \]

\[ +1 \]

\[ \text{address} \]

\[ \text{data} \]
Microcode Machine: Control (S)

- Control Signals (17)
- Control ROM
- next-state encoding reduces ROM width
- busy
- jump logic
Microcode Machine: Control (D)

- Opcode
- ext
- op-group
- μPC
- μPC+1
- μPC (state)
- +1
- μPCSsrc
- jump logic
- busy
- Control ROM
- address
- data
- Control Signals (17)
- next-state encoding reduces ROM width
Microcode Machine: Control (J)

- Opcode
  - ext
  - op-group
  - absolute

- $\mu\text{PC}$
  - $\mu\text{PC}+1$
  - +1

- $\mu\text{PC} (\text{state})$

- Control ROM
  - address
  - data

- Control Signals (17)

- next-state encoding reduces ROM width

- jump logic
  - busy
Microcode Machine: Control

- Opcode → ext
- absoluṭe
- op-group
- µPC → µPC + 1
- µPCSṛc
- jump logic
- zero
- busy
- Control ROM
  - address
  - data
- Control Signals (17)

input encoding reduces ROM height

next-state encoding reduces ROM width

13