PS1 Review

Section 4
2/15/2013
Yunsup Lee
“All problems in computer science can be solved by another level of indirection”

• David Wheeler

• Examples of indirection?
Pset #1, Lab #1

- Solutions online after section
- Two free extensions allowed for the entire semester
  - Must turn in one class after original due date
- Must turn in all labs
  - Cannot pass class otherwise
- Interested in helping us out?
  - Talk to me after class
Quiz #1

• Next Tuesday (2/19)
• Completely closed books (no cheat sheets, only pencils)
• Will look just like the Pset
Quiz #1 Cont.

- Possible Topics
  - ISA design
  - Microcoding
  - Pipelining
    - Bypassing/interlocking
    - Precise exceptions
    - Control hazards
    - Branch speculation (BTB,BHT,return stacks)
  - Iron Law

- How to study?
  - Fully understand Pset#1
  - Go over past tests
  - Read book! Go over lecture slides
  - Argue with a friend
• I’m at a conference to present @ Shenzhen
  – 2/22: Lab 2 Overview (Andrew Waterman)
  – 2/26: No OH
  – 3/1: Memory hierarchy and PS 2 (Chris Celio)
Microrcode Problem

• For microcode problems, key is to get the pseudocode right
  – Control signals follow readily from pseudocode

• Sanity checks:
  – Only one device may drive the bus
  – The bus probably should be driven every cycle
  – Don’t read from a register whose write-enable was a don’t-care
• don’t cares
  – If you won’t read A/B/MA registers again, their write-enables should be don’t-cares
  – If enMem is off, Mem Wr is a don’t-care
  – If enReg is off, Reg Wr is a don’t-care
  – If you *will* read rd,rs1,rs2 in the future, keep ldIR == 0
P2 ADDm

• M[rd] <-~ M[rs1] + M[rs2]
  – MA <- R[rs1]
  – A <- Mem
  – MA <- R[rs2]
  – B <- Mem
  – MA <- R[rd]
  – Mem <- ALU (A+B);
  – uBR=J to Fetch

• Note efficiency: 10 cycles vs. 30 for ld,ld,add,st
P2 STRCPY

• STRCPY
  – MA ← Rs; A ← Rs
  – B ← Mem
  – MA ← Rd
  – Mem ← B
  – If (B == 0) uBr to FETCH0
  – Rs ← A + 4
  – A ← Rd
  – Rd ← A+4, J to STRCPY
Microcoding Hints

• Correct Psuedo-code is Essential
• ldIR stays 0, if I still need to access rd,rs1,rs2
• S-subr whenever you access memory
  – must also keep ldmA == 0 to keep the address value
    locked in (likewise for other registers as necessary)
• If nothing is driving the bus, set all en* signals to 0
  – (yes, I realize you could technically set 3 to 0, and 1 to *)
• do not need to enable enALU when only doing branch comparisions
• You can look at Lab1d_ucode’s “microcode.scala” for inspiration

• Caveat Emptor:
  – there may be bugs, or suboptimal logic
  – (Chris is only human)

• Chisel doesn’t have “*” don’t care