

Slide 1

Administrivia

- Homework 6 on Web; due next Wednesday.

Slide 2

Minute Essay From Last Lecture

- Question: Looking at figure 5.33, what control signals do we need for the "instruction fetch" step? Give values for all of the following (or as many things on the figure as you have time to decide about):

RegDst	RegWrite	ALUSrcA
MemRead	MemWrite	MemtoReg
IorD	IRWrite	PCWrite
PCWriteCond	ALUSrcB	PCSrc

Pseudocode, again:

```
IR = Memory[PC]
PC = PC + 4
```

- Answer?

The Multiple-Cycle Implementation, Big Picture

- First step was to revise datapath — eliminate duplicate functional units, add multiplexors and “save areas” as needed.
- Next step was to define steps into which we can break instructions and figure out what control signals we need.
- Final step will be to figure out how to generate these control signals.

Slide 3

Instruction Steps for R-Type Instruction

- In pseudocode:
 - (1) `IR = Memory[PC]; PC = PC + 4`
 - (2) `A = Reg[IR[25-21]]; B = Reg[IR[20-16]];`
`ALUOut = PC + (sign-extend(IR[15-0]) << 2)`
 - (3b) `ALUOut = A op B`
 - (4b) `Reg[IR[15-11]] = ALUOut`
- This works, right? Trace through how data moves.

Slide 4

Instruction Steps for `lw`

Slide 5

- In pseudocode:

```
(1) IR = Memory[PC]; PC = PC + 4
(2) A = Reg[IR[25-21]]; B = Reg[IR[20-16]];
    ALUOut = PC + (sign-extend(IR[15-0]) << 2)
(3a) ALUOut = A + sign-extend(IR[15-0])
(4a1) MDR = Memory[ALUOut]
(5) Reg[IR[20-16]] = MDR
```

- This works, right? Trace through how data moves.

Instruction Steps for `sw`

Slide 6

- In pseudocode:

```
(1) IR = Memory[PC]; PC = PC + 4
(2) A = Reg[IR[25-21]]; B = Reg[IR[20-16]];
    ALUOut = PC + (sign-extend(IR[15-0]) << 2)
(3a) ALUOut = A + sign-extend(IR[15-0])
(4a2) Memory[ALUOut] = B
```

- This works, right? Trace through how data moves.

Instruction Steps for beq

- In pseudocode:

```
(1) IR = Memory[PC]; PC = PC + 4
(2) A = Reg[IR[25-21]]; B = Reg[IR[20-16]];
    ALUOut = PC + (sign-extend(IR[15-0]) << 2)
(3c) if (A == B) PC = ALUOut
```

Slide 7

- This works, right? Trace through how data moves. (If time, trace through control signals too.)

Instruction Steps for j

- In pseudocode:

```
(1) IR = Memory[PC]; PC = PC + 4
(2) A = Reg[IR[25-21]]; B = Reg[IR[20-16]];
    ALUOut = PC + (sign-extend(IR[15-0]) << 2)
(3d) PC = PC[31-28] || (IR[25-0] << 2)
```

Slide 8

- This works, right? Trace through how data moves.

Minute Essay

- None — quiz.

Slide 9