Pipelined Control

Tom Kelliher, CS 240

Apr. 17, 2002

1 Administrivia

Announcements

Assignment

Read 6.4 and 6.5.

Homework due 4/24, **beginning of class**: 6.3, 6.4, 6.7, 6.22.

From Last Time

Pipelined datapath.

Outline

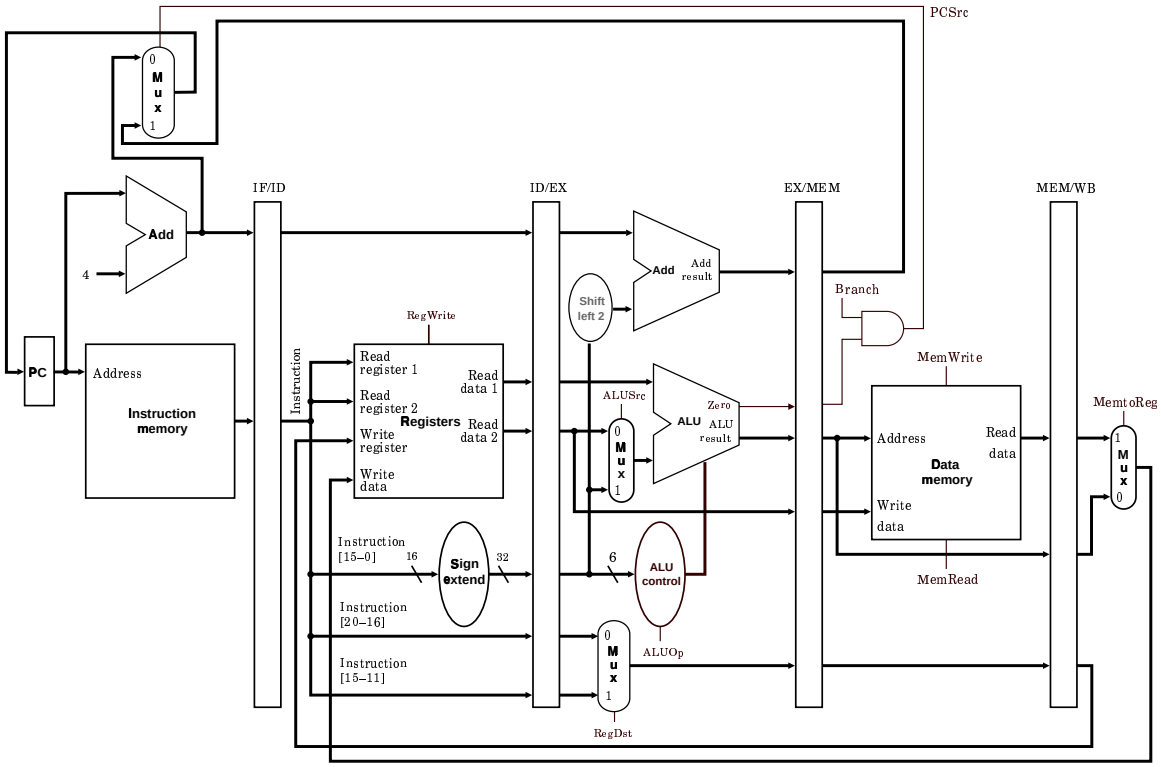
1. Pipeline control.
2. A real-world example of the pipeline in use.

Coming Up

Data hazards: forwarding and stalling.

2 Pipeline Control

Here's the datapath with control signals shown:

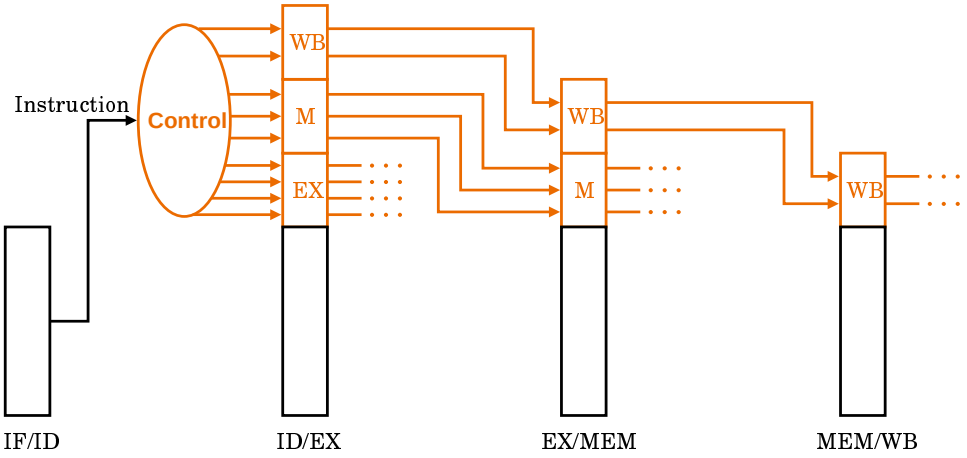


Questions:

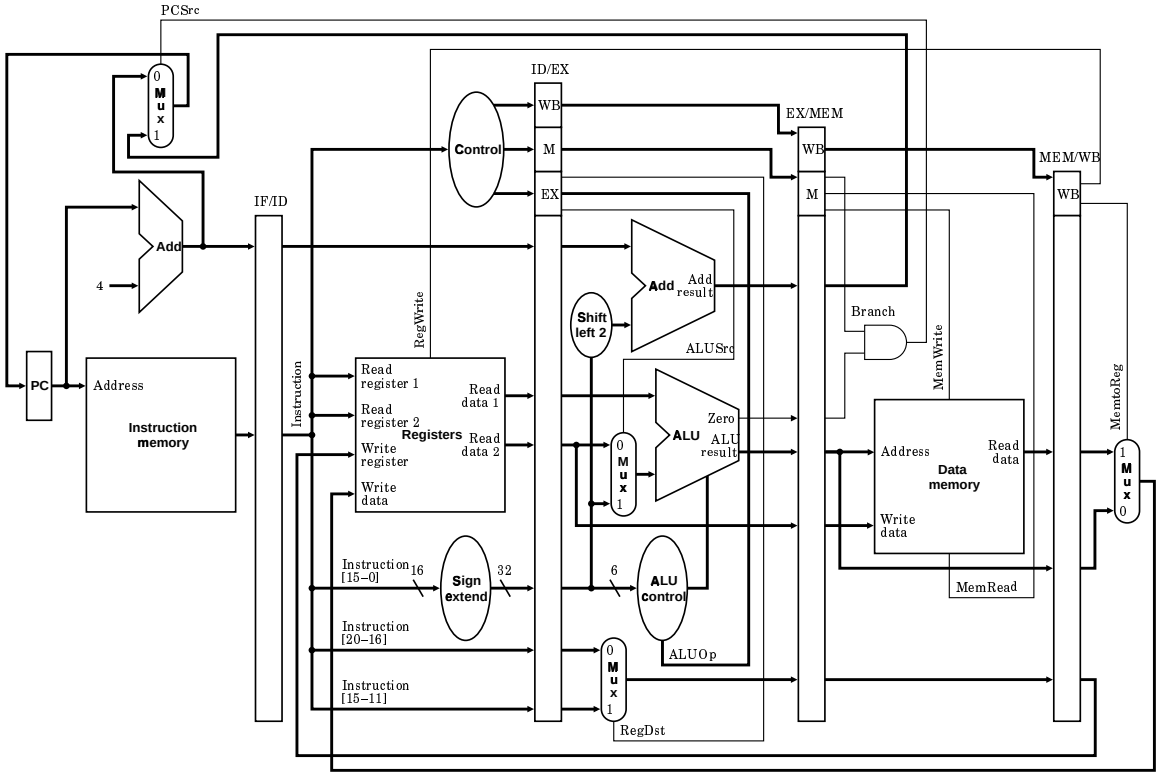
1. Consider an instruction in, say, the execute stage. Does it need to exert control over any other stages during that cycle?

2. What organizational options doe we have for the control? What are the tradeoffs?

This shows the control signals being generated during the instruction decode stage and being passed down the pipeline in parallel with the data:



The final, simple, pipelined processor:



3 A Real-World Example

This will demonstrate shortcomings of the simple pipelined processor. Can we find them all?

Consider the following MIPS code segment, assuming the following:

1. The code segment begins at address 1000.

2. Initially, `$7` = 4, `$8` = 2000, `$10` = 12, and `$11` = 20.

3. Also initially, memory location 2000 has the word value 16 and memory location 2004 has the word value 6.

```
Top:    lw $10, 0($8)
        lw $11, 4($8)
        slt $9, $11, $10
        beq $9, $0, Label
        sw $10, 4($8)
        sw $11, 0($8)
Label:  add $8, $8, $7
```

1. What does this code do?

2. Can it be optimized? Does what follows the `add` affect the optimization?

3. What happens if we run it through the pipeline? Do we get the correct results? Why or why not?