16.482 / 16.561: Computer Architecture and Design
Fall 2013

Lecture 5: Key Questions
October 7, 2013

1. Describe the basic operation of a pipelined datapath.

2. Does pipelining improve latency or throughput?

3. What is the maximum potential speedup of pipelining?

4. If one pipeline stage can run faster than the others, how does that affect the speedup?

5. Draw a basic pipeline diagram and describe the 5 stages.

6. What are the implications (in terms of what hardware is needed) of:
a. Incrementing the PC and adding two registers in the same cycle?

b. Reading memory on a load and fetching? a new instruction in the same cycle?

7. Every pipelined instruction takes the same number of cycles. Why?

8. <u>Example:</u> Say we have the following code:

```
loop:           add $t1, $t2, $t3
                lw  $t4, 0($t1)
                beq $t4, $t3, end
                sw  $t3, 4($t1)
                add $t2, $t2, 8
                j   loop
end:            ...
```

Assume each pipeline stage takes 4 ns. How much time will it take on an ideal pipelined datapath (i.e., no delays between instructions)?

9. Describe how a pipelined datapath is divided into stages and how data is transferred between stages.
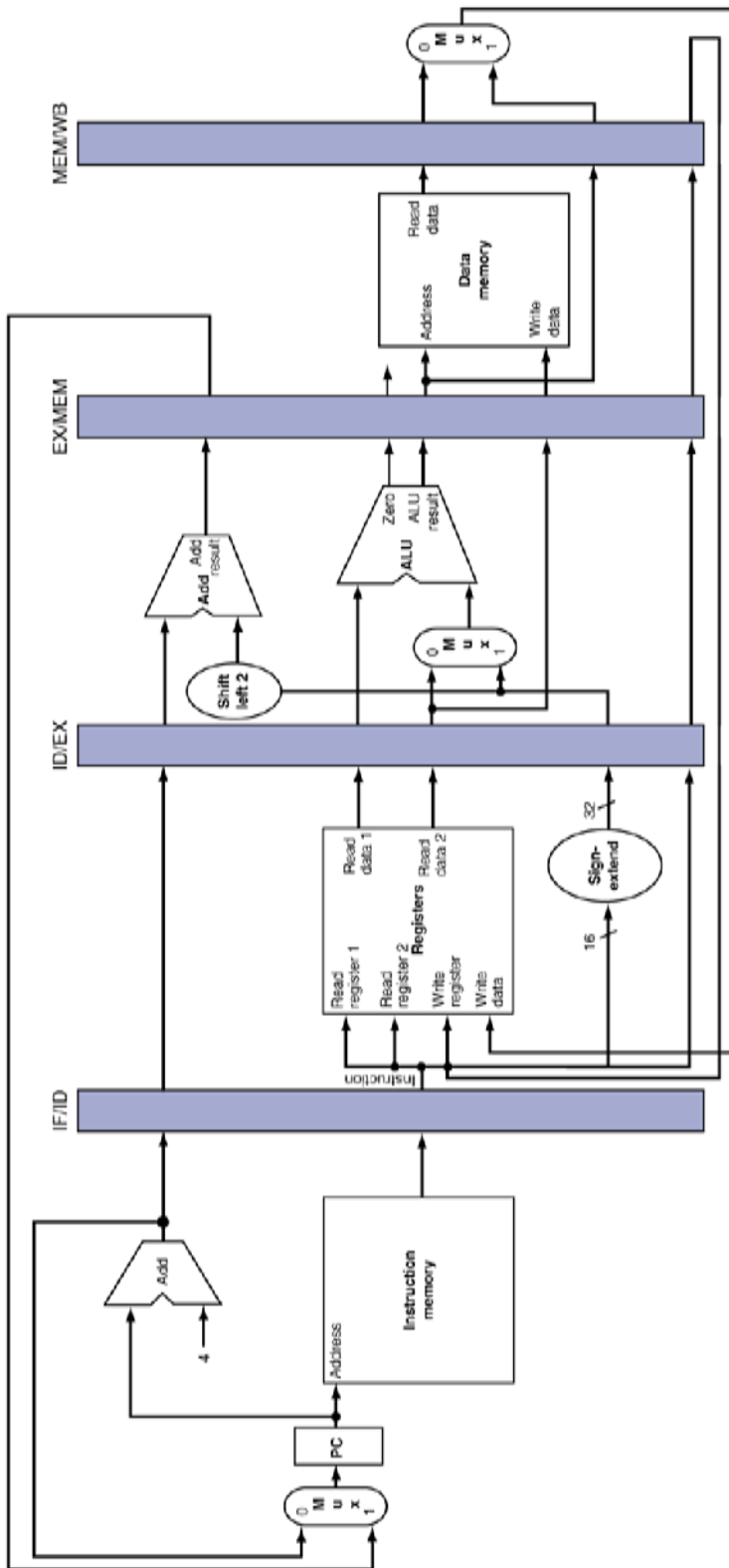
10. Describe the three different types of pipeline hazards.

11. <u>Example</u>: What are no-ops? Given the following code, where are no-ops needed?

```
add $t2, $t3, $t4
sub $t5, $t1, $t2
or  $t6, $t2, $t7
slt $t8, $t9, $t5
```

12. Explain how forwarding works in a pipelined datapath.

13. Describe a case in which forwarding alone is not enough to resolve a data hazard.

14. Describe the reason processors may experience branch delays.

15. Describe the basics of branch prediction.