Topics covered

Book: chapters 1, 2, and 4 (mostly 1 and 4).
Italics are those topics with questions currently in the practice midterm.

- Security
  - Meltdown and Spectre
  - Security as a design constraint
- History of computing
  - *Moore's law*
  - Dennard scaling
- CMOS devices
  - Energy
  - Power
- Trends in computing
- *Instruction set architectures (ISAs)*
  - RISC versus CISC
- RISC-V
  - Instruction types
  - Extensions
- From code to execution
  - Going from higher-level languages to machine code
  - Executing an instruction
- Single-cycle CPU
  - *Datapath*
  - Adding new instructions
- *Performance*
  - *Measuring performance*
  - *Iron law*
- *Pipelined CPU*
  - *Pipeline performance*
  - *Pipeline control*
  - *Hazards (data, control, structural)*
  - *Branch prediction*
  - *Forwarding*
- *Instruction-level parallelism (covered on Wed. 2/4 and Fri. 2/6)*
  - *Static (VLIW, SIMD)*
  - *Dynamic (superscalar, out of order execution)*
  - *Limits on ILP*

1. _____ claims that the number of transistors per chip will double every 18-24 months.

2. Increasing the frequency of the processor leads to increased performance and increased _____.

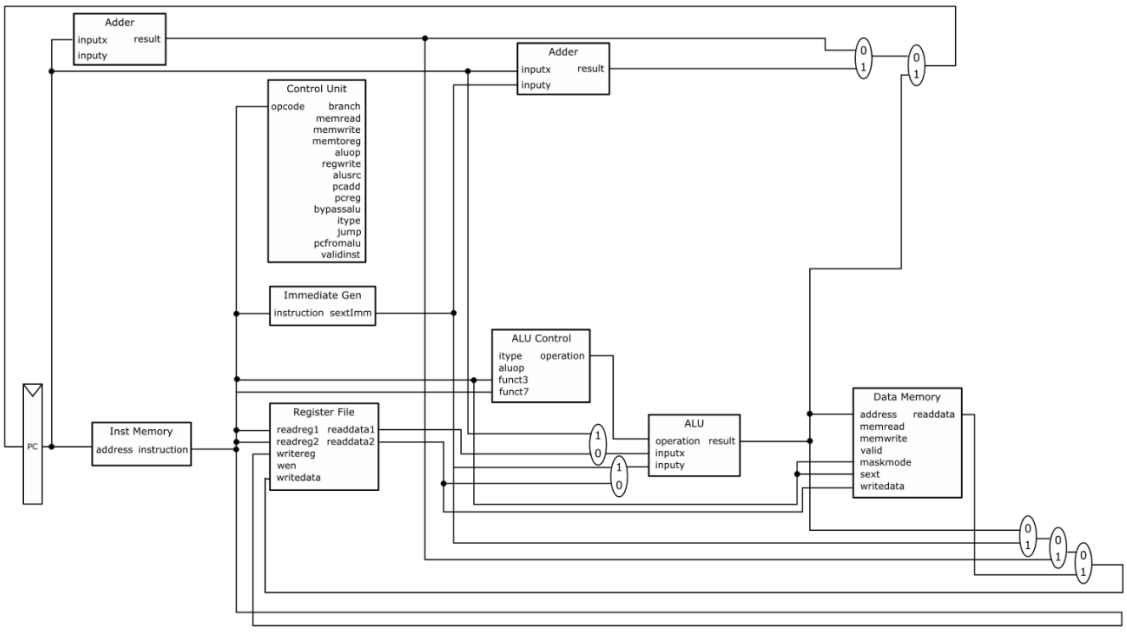
3. What are three ways to compare two systems to determine which is the "better" system?

4. A _____ predicts the performance of a real-world application, but often has a shorter runtime and is standardized to allow comparisons across systems.

5. The ISA defines the interface between _____ and _____.

6. Processor A can run 1 billion instructions in 0.5 seconds at a frequency of 500 MHz. You are proposing a pipelined design for a new processor, design B. Your processor has a CPI of 4. The company you work for will not release a new design unless it has at least a 1.5x speedup compared to processor A. What is your target frequency?

For the rest of the problems on this page, use the single cycle DINO CPU diagram from Lab 2 below.




Single cycle DINO CPU

7. What does the *ControlUnit* block do?

8. Explain the *funct7* and *funct3* wires going into the ALU control unit, and what they do.

9. Shade the wires and structures used for a jump instruction on the diagram above.

10. Give a short assembly example that would have a RAW hazard in a pipelined processor. Describe why this hazard exists in your example.

11. Give a short assembly example that has a WAW dependence. Is this a hazard in your pipelined processor, what about in an out of order design?

12. What are the benefits of VLIW compared to superscalar designs? What are the drawbacks of VLIW that led to most systems being implemented as out of order designs instead?

13. How does the number of pipeline stages affect the CPI of an in-order design?

14. Explain why predicting all branches as taken is better than predicting all branches as not taken. Writing down a code snippet may help you answer this question.

15. What are the two pieces of information that a branch predictor predicts?

16. Fill in the table with the data hazards in the RISC-V code below. List the instruction address and the register that causes the hazard. Assume the basic 5 stage pipeline with **no forwarding**. Write the addresses of the younger and older instructions. Older instructions are executed before younger instructions (i.e., in cycle 2, addi is younger than lw).

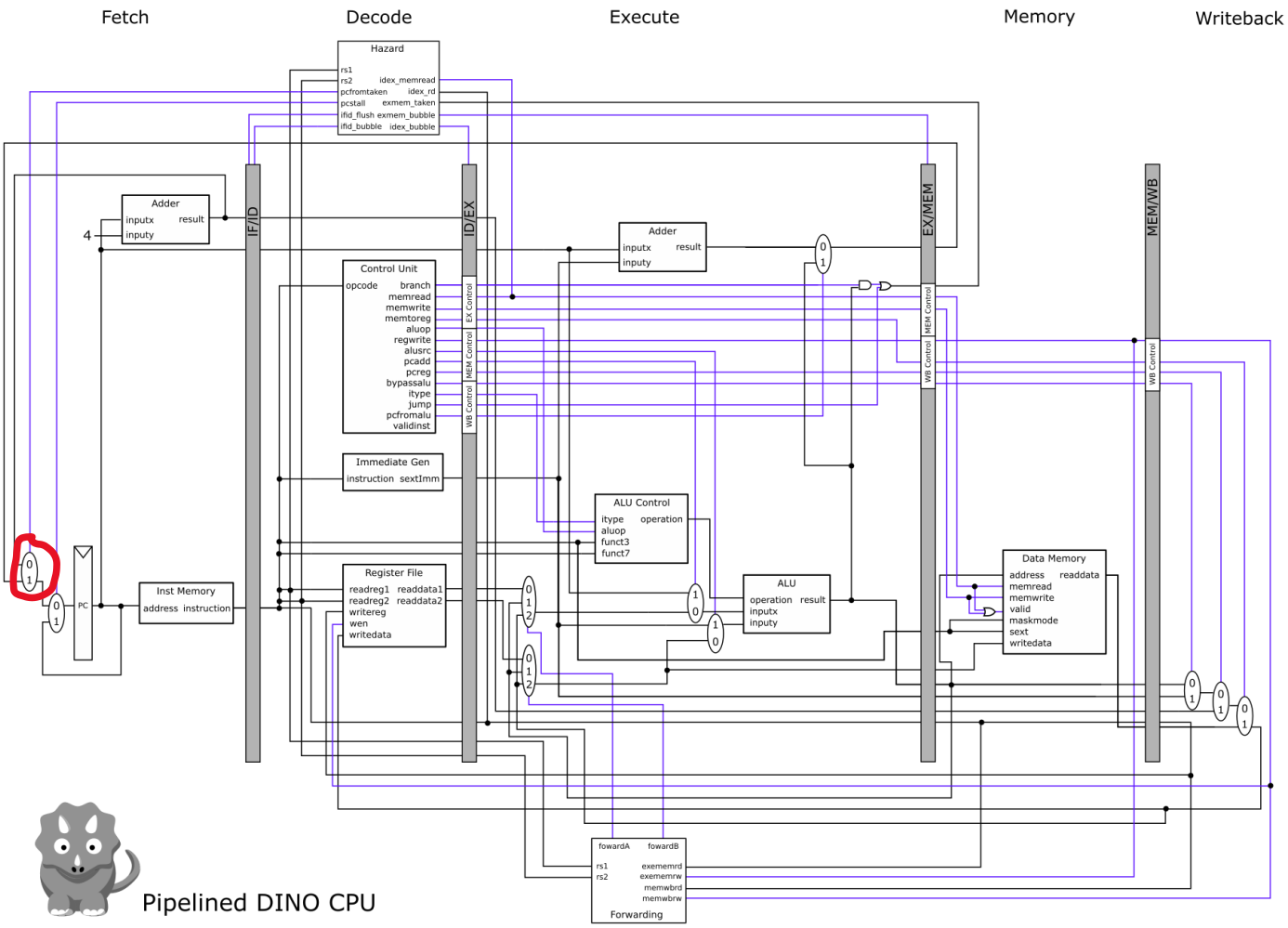
```
0:   lw    x2, 0(x1)
4:   addi  x7, x2, -5
8:   sw    x4, 12(x3)
12:  add   x10, x7, x4
16:  lw    x8, 100(x10)
20:  sub   x10, x9, x8
```

Register number	Younger instruction	Depends on	Older instruction
		➔	
		➔	
		➔	
		➔	
		➔	
		➔	

17. Use the code snippet from the previous problem. Now assume that we are executing that code on an out-of-order processor. Are there any other hazards that we need to worry about? For each hazard, list the hazard type and the instructions involved.

18. Assume we executed the code snippet from the previous two problems on a processor that implemented register renaming. In this processor, why do we not need to worry about the hazards from the previous problem?

Use the pipelined DINO CPU diagram below to answer the following questions.



19. When would we select the 1 input on the circled MUX in the IF stage?

20. When would we bubble the ID/EX pipeline register?

21. Fill in the per-cycle pipeline diagram table for the following code. Show all cycles where forwarding occurs and the stages between which the data is forwarded. Assume there is no branch prediction and branches are resolved in the decode stage. The branch is resolved not taken in the decode stage.

    Use a pencil for this question!

```
0:  add  x3, x7, x11
4:  sub  x2, x4, x5
8:  add  x6, x3, x2
12: lw   x8, 20(x6)
16: beq  x8, x0, 4
20: add  x4, x7, x9
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

22. Can we move instruction 20 up between instructions 12 and 16 to potentially remove a stall? Why or why not?