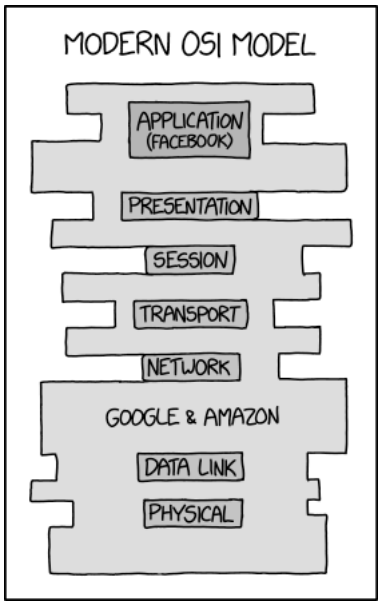
Lecture 10: Review

Wednesday, January 30, 2019 10:38 AM



7. 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?

$t = \frac{sec}{cycle} \cdot \frac{cycles}{inst} \cdot inst$

$Speedup = \frac{old}{new}$

$1.5x = \frac{0.5}{t_{new}} \rightarrow t_{new} = \frac{1}{3}s$

B $\frac{1}{3} = x \cdot 4 \cdot 10^9$

B $x = \frac{1}{12}\ ns/cycle \rightarrow 12\ GHz$

$0.5 = \frac{1}{500 \cdot 10^6} \cdot CPI \cdot 10^9$

A $CPI = 0.25$

25

25. RISC instructions tend to be less "powerful" than CISC instructions, thus it takes more RISC instructions to write a program compared to the same program written for a CISC architecture. How are RISC architectures faster than CISC architectures despite this problem?

freq can be higher w/ RISC → longest inst less time for RISC than longest inst for CISC

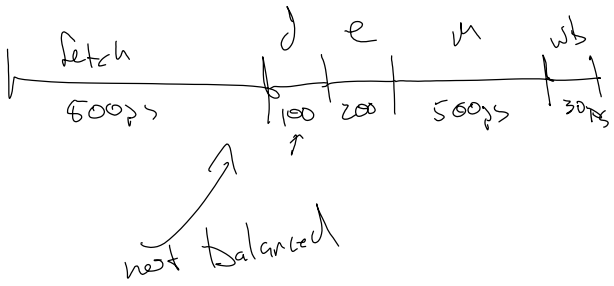
CPI is lower for RISC

↳ CISC inst might take many cycles to complete

RISC → can better utilize hardware (nothing is idle)

Balanced pipeline

→ When splitting into stages you want all stages to be equal in time



need balanced pipeline since freq/cycle time set by longest stage

11. With our canonical five-stage pipeline, what's the maximum number of instructions that we can run at once? How many could we potentially execute at once with a twenty-stage pipeline?

For the next two questions, examine the following five-stage pipeline. The time it takes for each stage to execute is below.

- Fetch: 220 ps
- Decode: 180 ps
- Execute: 420 ps
- Memory: 210 ps
- Writeback: 190 ps

sets cycle time

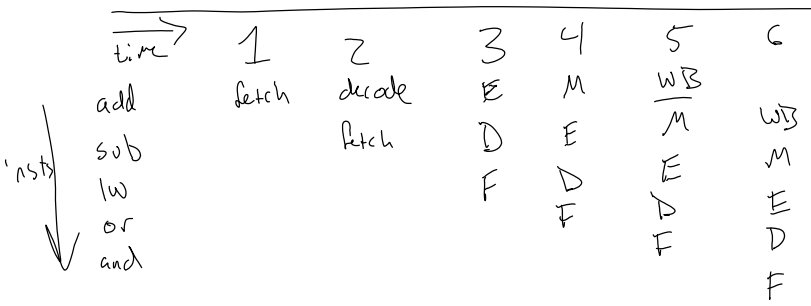
fetch 220
decode 180
Ex1 210+ε
Ex2 210+ε
M 210
WB 190

Split execute into 2 stages → add extra registers

12. Why is the pipeline above not balanced?
13. Suggest two ways that we can make this pipeline balanced.

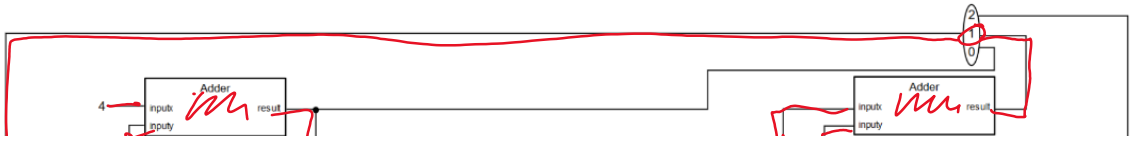
factor 2 is a good rule of thumb

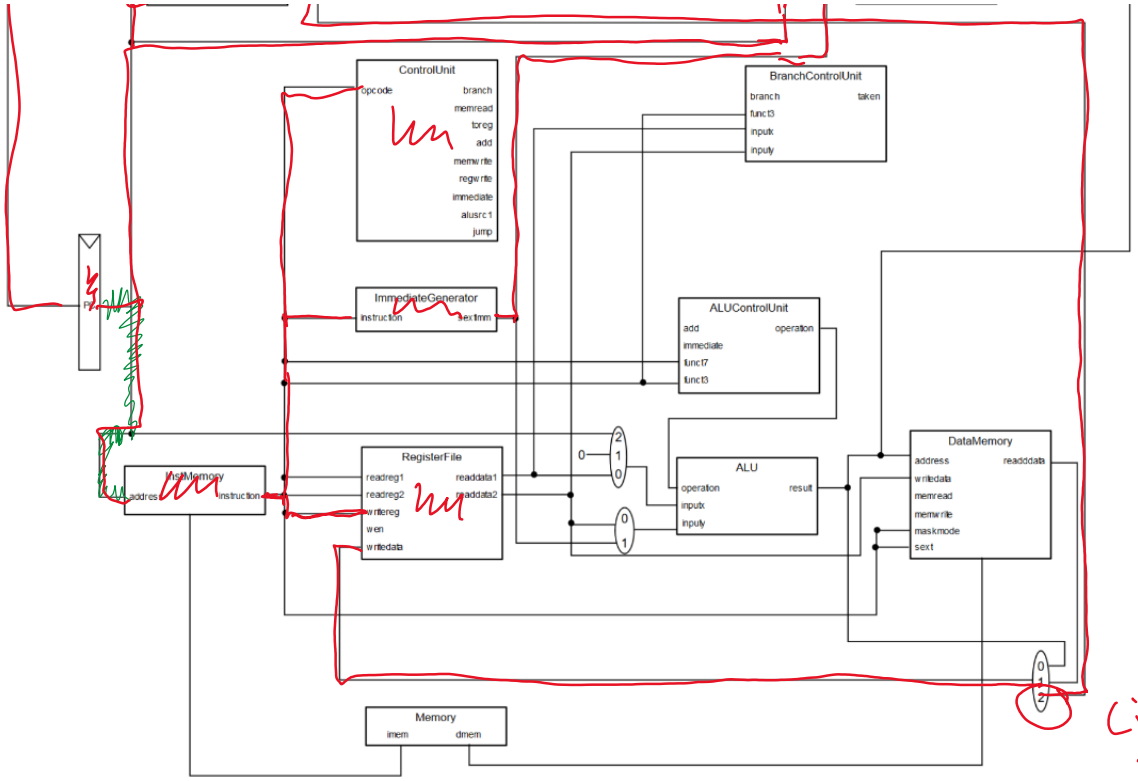
F+D 400ps
E 420ps
M+WB 400ps



Inst per cycle → 1
↳ throughput

latency → time from start to end 5 cycles





16. Why should security be considered as a first-order design constraint when designing a new chip or architecture?

very difficult to fix H/W security bugs

Don't want to lose data

↳ privacy

integrity of data

14. Amdahl's law is a mathematical representation of what common computer design principle?

parallization is OK

↳ make the common case fast