Lecture 7: Pipelining basics

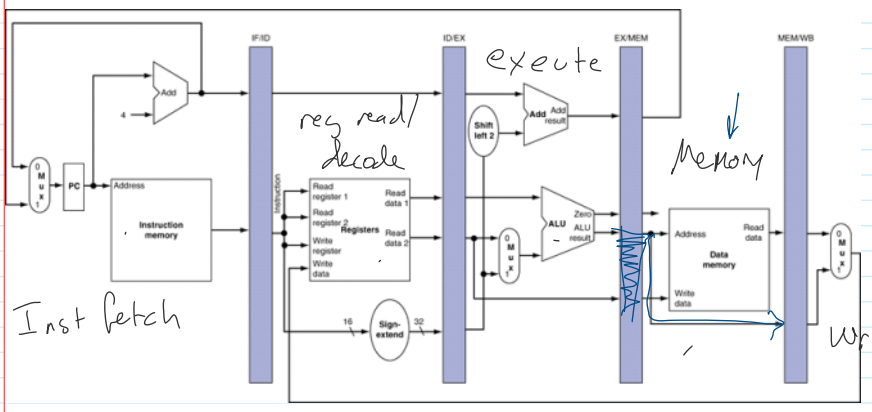
Monday, January 29, 2018 11:20 AM

Outline

- Finish multi-cycle processor design.
- Pipelining metaphor
- Pipelining in processors
- Hazards

1) PC to inst memory
→ inst memory 1kB
PC can address 4GB
↳ low-order bits to access inst. mem.

2) Inst is word addr. not byte



Single/multicycle performance

Single cycle design

Fetch: 100 ps
Decode/reg read: 80ps
Execute: 80ps
Memory read: 200ps
Writeback: 60ps

100ps for multicycle clock rate

Cycle time for single cycle? 520ps (1000/520=1.9231 GHz)

Time for program with 1 billion instructions?
CPI = 1
Time = instructions * CPI * cycle time
Time = 1*10^9 * 1 * 520 / 10^12 = 0.52s

Multicycle design

Assuming memory is 2 cycles: What is the cycle time? 100ps → 10GHz

Now, different instructions could take different cycles

- Loads = 6 cycles
- Stores = 5 cycles
- R-type = 4 cycles
- Branch = 3 cycles
- Jump = 2 cycles

loads: 20% R-type: 50%
Stores: 10% Branches: 15%
jumps: 5%

What is the time for the same number of instructions (1 billion)? need to know inst. Breakdown

cycles per inst
Time = inst X CPI X cycle time
1 billion 4.25 100 ps
$10^9 \cdot 4.25 \cdot \frac{100}{10^{12}} = 0.425$ s

CPI = 6 · .2 + 5 · .1 + 4 · .5 + 3 · .15 + 05 · 2
= 4.25 cycles per inst.

What is the cycle time required for meeting the same performance of single cycle?

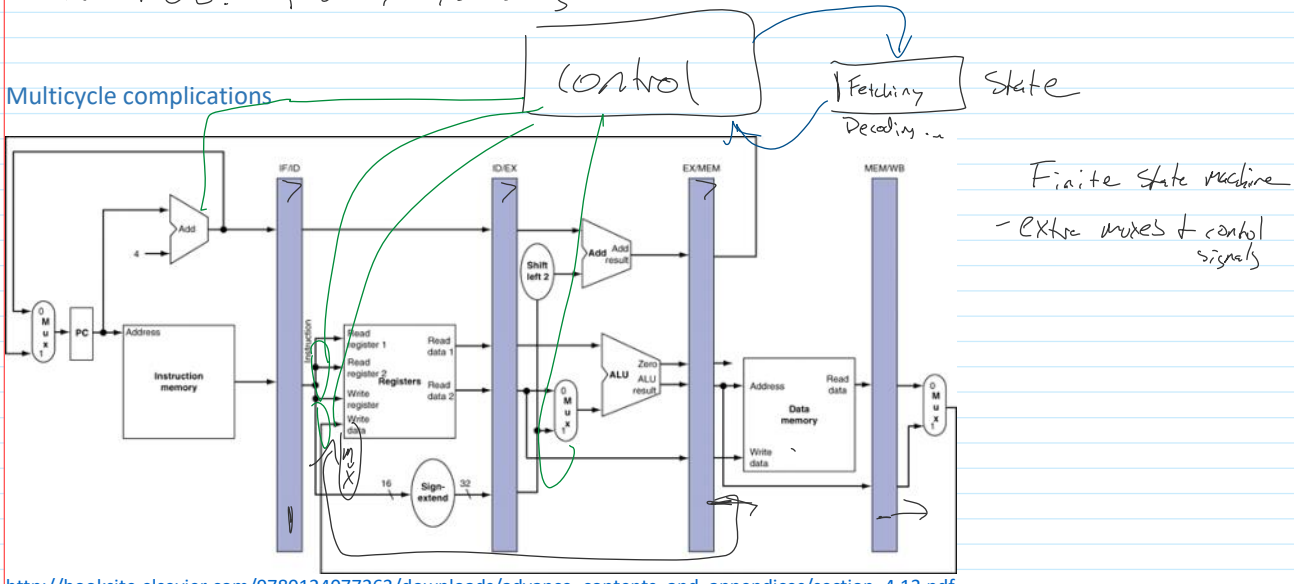
Time = inst · CPI · cycle time
0.52s = 10^9 4.25 x

Time = Inst · CPI · cycle time

$0.525 = 10^9 \cdot 4.75 \cdot x$

$x = 122.3\ ps \rightarrow 1.2 \times 10^{-11}\ s$

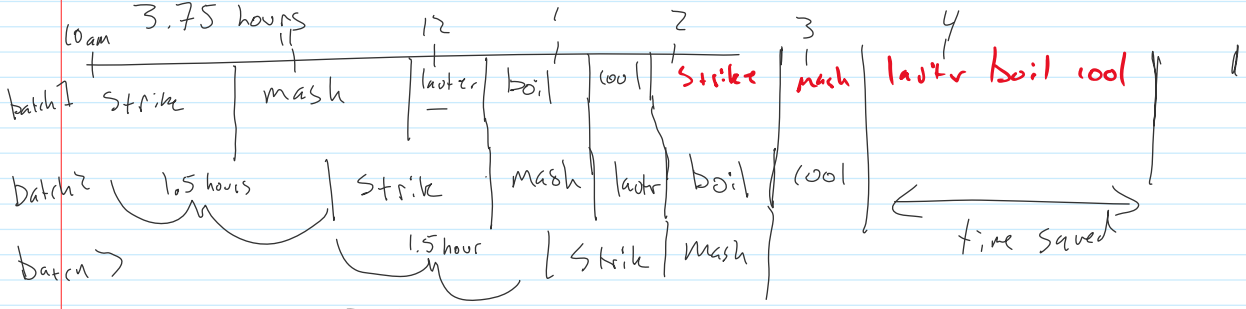
Multicycle complications



http://booksite.elsevier.com/9780124077263/downloads/advance_contents_and_appendices/section_4.13.pdf

Pipelining analogy

1) heat water → strike water ~45 min
2) mash combine grain & hot water sugar is extracted ~60 min
3) lauter → getting sugary water ~30
4) boil 1) sterilizes (wort) 2) hops → bitterness & flavor ~60 min
5) cool the wort & pitch yeast ~30 min



throughput? rate of brewing? 2/3 batch/hr (1 batch / 1.5 hr)

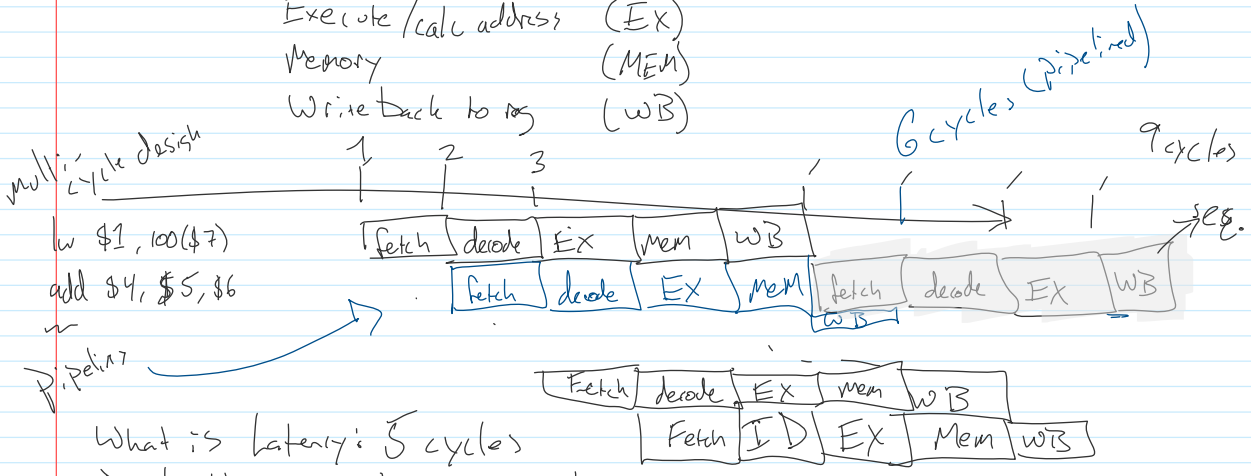
Latency: 3.75 hours

long term speedup of pipeline? (compared to sequential) → 3.75 / 1.5 =

Pipelined processor

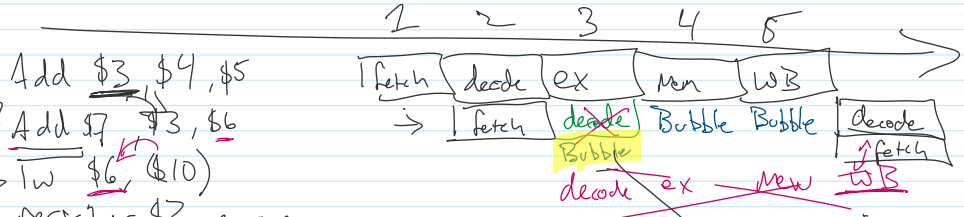
5 stages → Inst fetch (IF)
Inst decode & reg. read (ID)

5 stages → Inst fetch (IF)
Inst decode & reg. read (ID)
Execute / calc address (EX)
Memory (MEM)
Write back to reg (WB)



What is latency: 5 cycles
Peak throughput: 1 inst per cycle
↳ IPC not CPI
Cycles per inst = 1/IPC

Single cycle → ~2 GHz CPI = 1 .5 s for 1 billion
Multi cycle → 10 GHz CPI = 4.25 .425
pipeline proc. → 10 GHz CPI = 1 → .1 for 1 billion



cycle
0 1 2 3 4 5
3: 7 7 26
4: 11 11
5: 15 15
6: 21
7: 0

register $3 causes hazard
data hazard
read after write
↳ inserted when we need to stall

Workload:

- 10% store
- 20% load
- 15% branch
- 5% jump
- 50% ALU/R-type