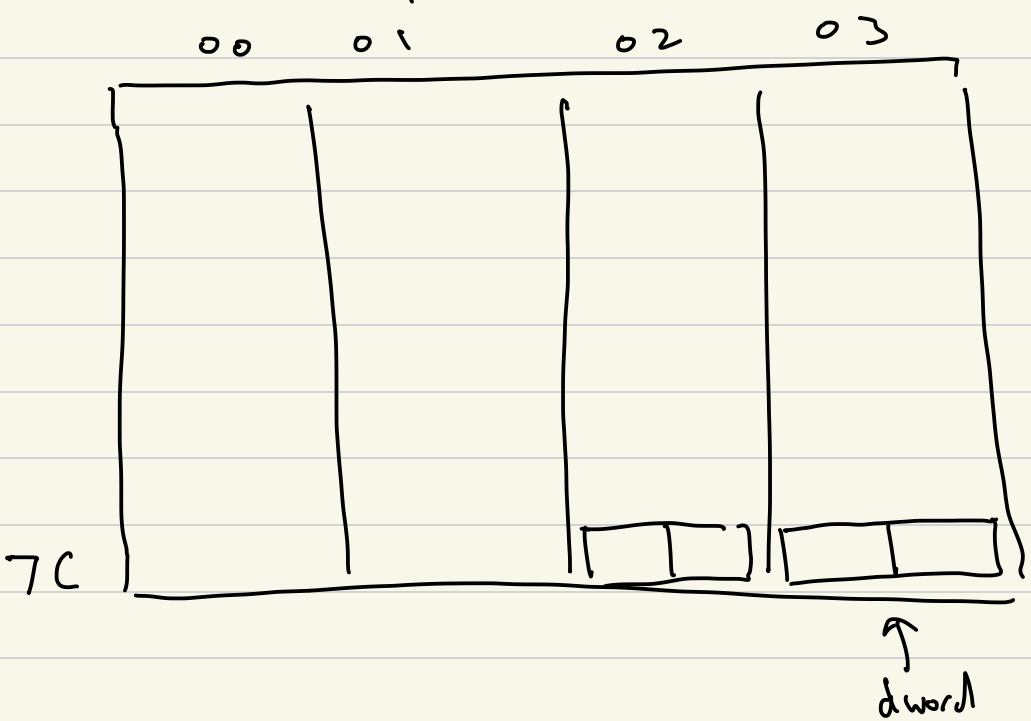
CS315-02 Processor Design Pipelining

Past scores and make up points
75% back on Project02, Project03, Project04
Address code quality for project03, project04
10 additional points per project
Last day of class

Project 06 Q & A

Data Memory



Single-cycle processor

Multi-cycle processor

Pipelined processor

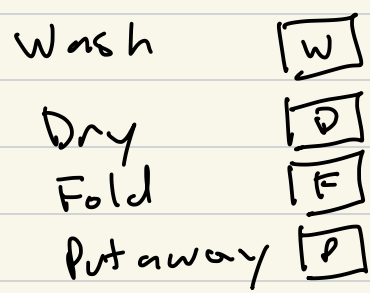
```
x = x * 4
  ||
  \/
slli t0,t0,2

x = x * 7
```

Laundry

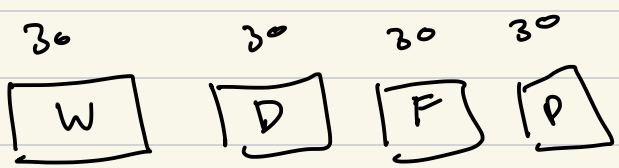
Laundry steps

1) Wash
2) Dry
3) Fold
4) Put away

Assume: each step takes 30 mins

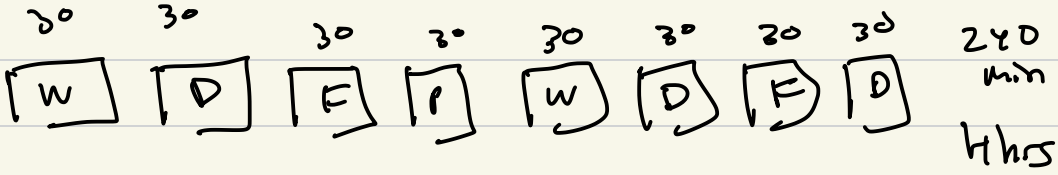


load

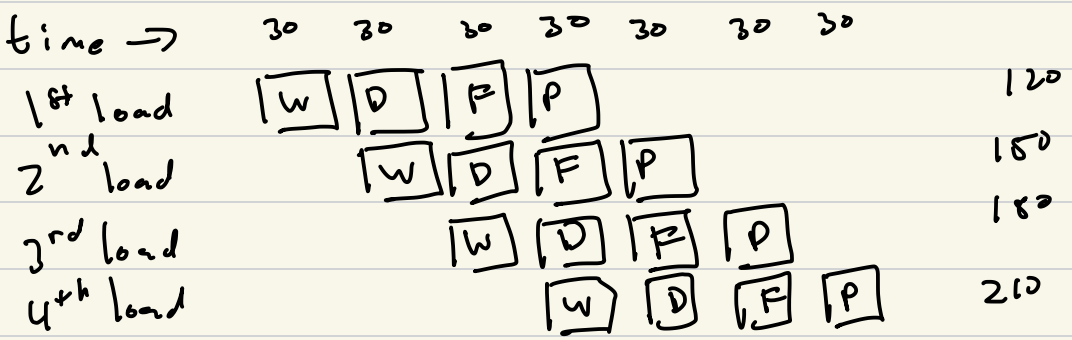


= 120 min

2 hrs



Laundry Pipeline



1 load = 120 mins (2 hrs)
2 loads = 150 mins (2.5 hrs)
4 loads = 210 mins (3.5 hrs)

100 loads?

Serial: 100 x 2hrs = **200 hrs**

Pipeling:

2hrs 30m

1 x (2.0) + (100-1) x (0.5)

2 + (99) x (0.5)

2 + 49.5 = **51.5 hrs**

50

1000 loads

1 x (2.0) + (1000-1) x (0.5)

2 + (999) x (0.5)

2 + 499.5 = **501.5 hrs**

500

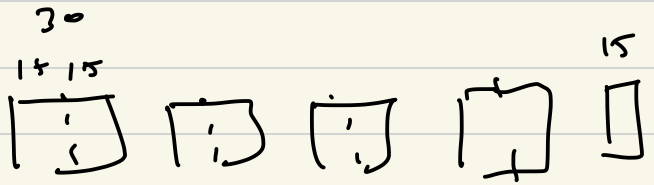
Latency

Throughput

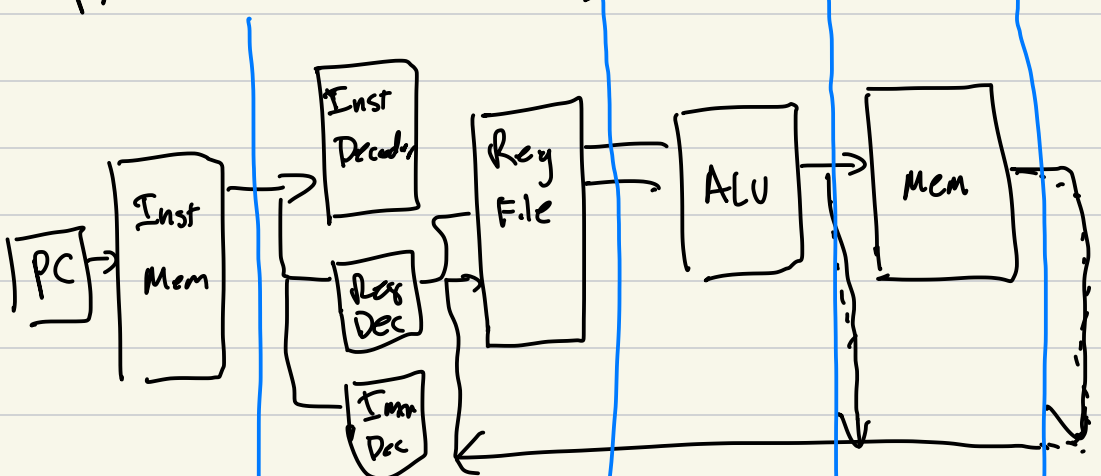
In principle

An n-stage pipeline can speed up execution by

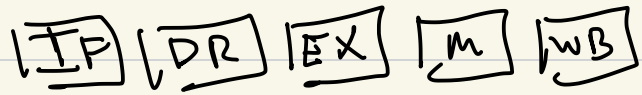
$$\frac{1}{n}$$



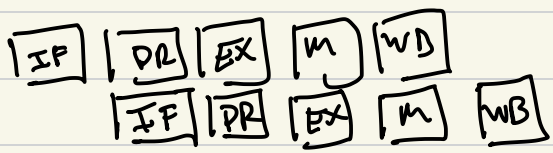
Processor Pipelining



Instruction Pipeline Stage




```
addi a0, zero, 1
addi a1, zero, 2
```



Pipeline Hazards

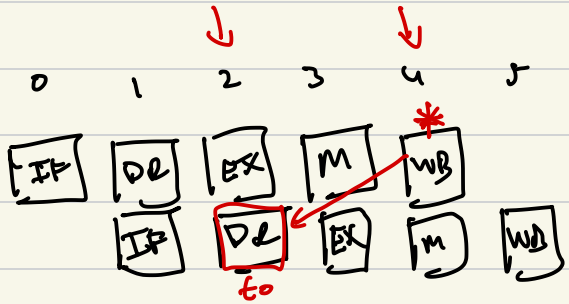
Data Hazards

Control Hazards (jumps and branches)

Data Hazards

time →

```
addi t0, zero, 7
addi t1, t0, 3
```

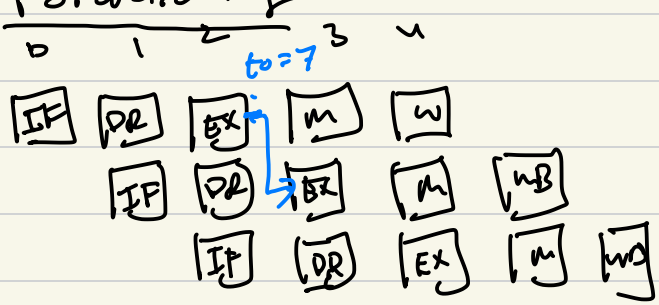


Solution #1 insert nops

→ add zero, zero, zero

```
add: t0, zero, 7
nop
nop
nop
add: t1, t0, 3
```

Solution #2 Forwarding




```
add: t0, zero, 7
add: t1, t0, 3
add t2, t0, t0
```

```
ld t0, (t1)
add t2, t0, t0
```

Stall

Control → Flush