The Y86 Pipelined Datapath: Data and Control Hazards

CSCI 237: Computer Organization
23rd Lecture, Friday, October 31, 2025

Kelly Shaw

Slides originally designed by Bryant and O'Hallaron @ CMU for use with Computer Systems: A Programmer's Perspective, Third Edition

1

Administrative Details

- Lab #4 due Tuesday at 11pm
- Read CSAPP Ch. 4.4-4.5
- Apply to be a TA by Nov. 7
- Final exam
  - Sunday, Dec. 14, at 1:30-4pm in Clark Hall 105
- Lab #5 partner sign-up
  - Due Wednesday at 8am
  - Both partners must sign up
- Spooky social for colloquium today
  - Start your junk food eating off early!

2

Last Time: The Y86 Datapath

- Construction a single-cycle datapath for Y86

3

Today: The Y86 Pipelined Datapath

- Pipelining Concepts
- Construction of a pipelined datapath for Y86
  - Adding pipeline registers
  - Data hazards
  - Control hazards

4

Our goals for a better processor design:

- Faster clock rate
- Use machine more efficiently
- No longer execute only one instruction at a time

In order to claim we've made an "improvement", we need a way to measure success

5

5

Performance Measurements

- Cycle Time: Time in between clock ticks
- Latency: Time to finish a complete job, start to finish
- Throughput: Average jobs completed per unit time
- CyclesPerJob: Number of cycles between finishing jobs.

6

6

Pipelining Example: Laundry

- Laundry-o-matic washes, dries & folds
- Wash: 30 min
- Dry: 40 min
- Fold: 20 min
- It switches them internally with no delay
- How long to complete 1 load? _____

7

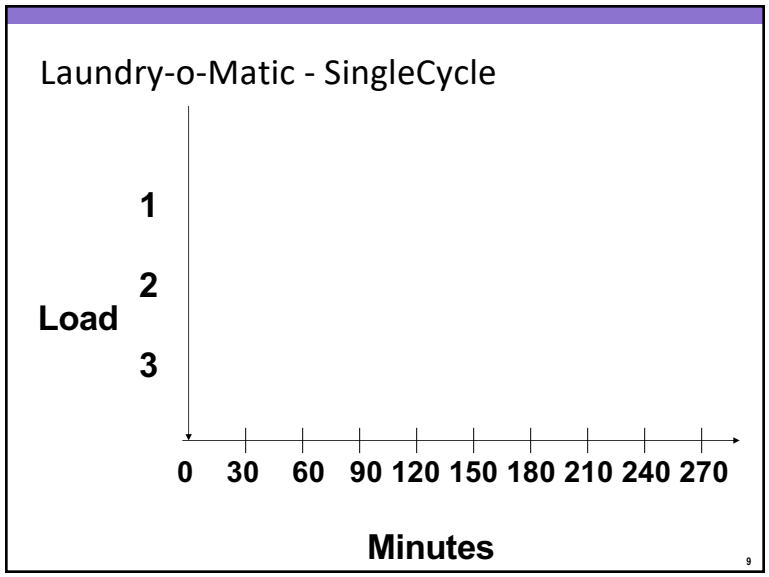
7

Pipelining Example: Laundry

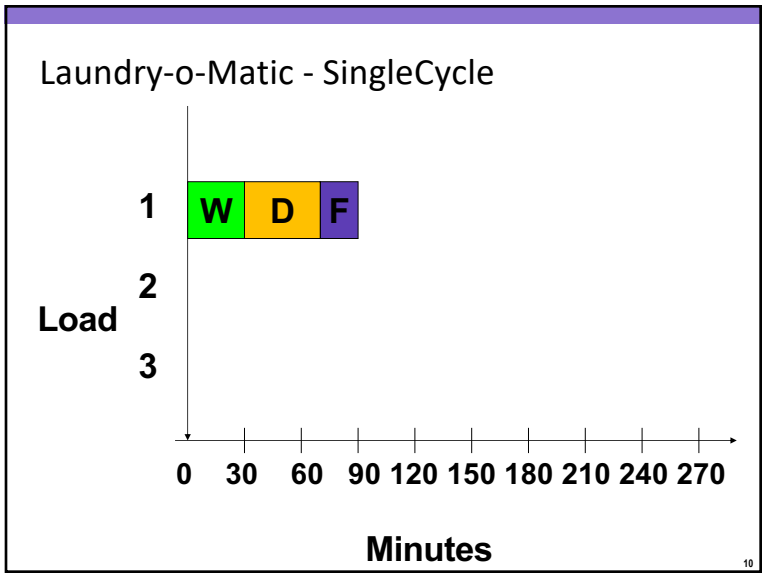
- Laundry-o-matic washes, dries & folds
- Wash: 30 min
- Dry: 40 min
- Fold: 20 min
- It switches them internally with no delay
- How long to complete 1 load? **90 min**

8

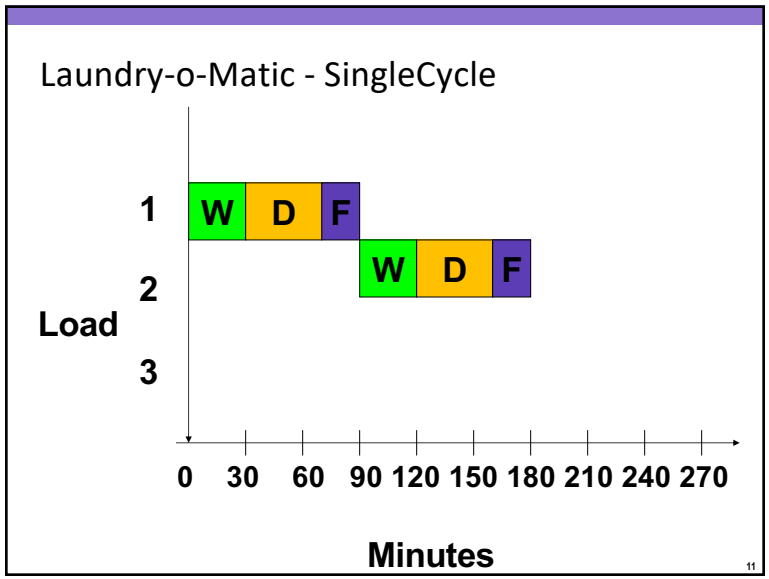
8


Laundry-o-Matic - SingleCycle
Load
1
2
3
0 30 60 90 120 150 180 210 240 270
Minutes
9

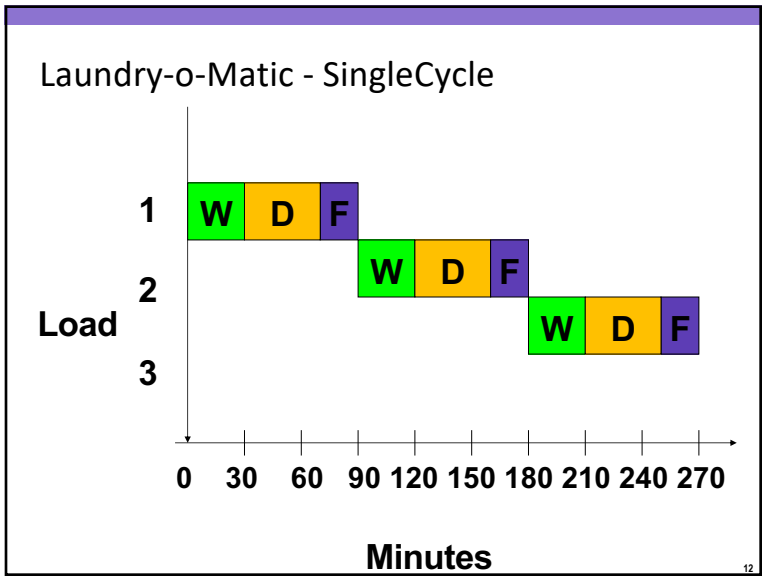
9


Laundry-o-Matic - SingleCycle
Load
1
2
3
W D F
0 30 60 90 120 150 180 210 240 270
Minutes
10

10


Laundry-o-Matic - SingleCycle
Load
1
2
3
W D F
W D F
0 30 60 90 120 150 180 210 240 270
Minutes
11

11


Laundry-o-Matic - SingleCycle
Load
1
2
3
W D F
W D F
W D F
0 30 60 90 120 150 180 210 240 270
Minutes
12

12

Laundry-o-Matic

- Cycle Time: Clothing is switched every ____ minutes
- Latency: A single load takes a total of _____ minutes
- Throughput: A load completes each _____ minutes
- CyclesPerLoad: Every ____ cycles, a load completes

13

13

Laundry-o-Matic

- Cycle Time: Clothing is switched every 90 minutes
- Latency: A single load takes a total of _____ minutes
- Throughput: A load completes each _____ minutes
- CyclesPerLoad: Every ____ cycles, a load completes

14

14

Laundry-o-Matic

- Cycle Time: Clothing is switched every 90 minutes
- Latency: A single load takes a total of 90 minutes
- Throughput: A load completes each _____ minutes
- CyclesPerLoad: Every ____ cycles, a load completes

15

15

Laundry-o-Matic

- Cycle Time: Clothing is switched every 90 minutes
- Latency: A single load takes a total of 90 minutes
- Throughput: A load completes each 90 minutes
- CyclesPerLoad: Every ____ cycles, a load completes

16

16

Laundry-o-Matic

- Cycle Time: Clothing is switched every 90 minutes
- Latency: A single load takes a total of 90 minutes
- Throughput: A load completes each 90 minutes
- CyclesPerLoad: Every 1 cycles, a load completes

17

17

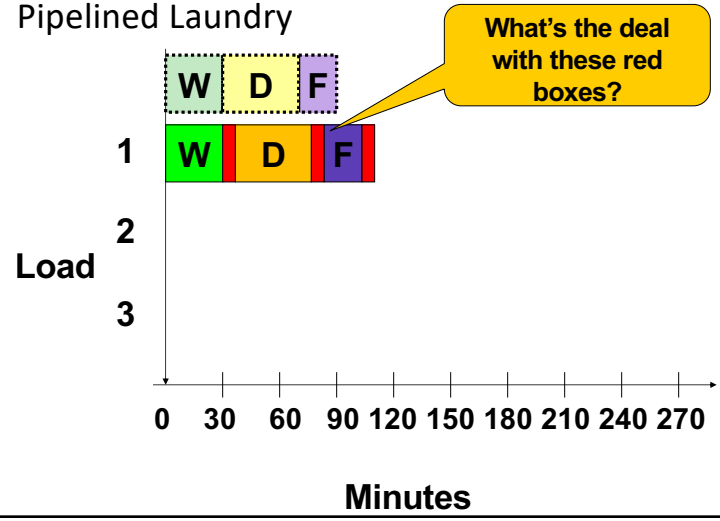
Laundry-o-matic 2.0: Pipelined Laundry

- Suppose we could split the laundry-o-matic unit into a separate washer, dryer, and folder (what a novel concept!!!)
- Moving the laundry from one to another takes 6 minutes

18

18

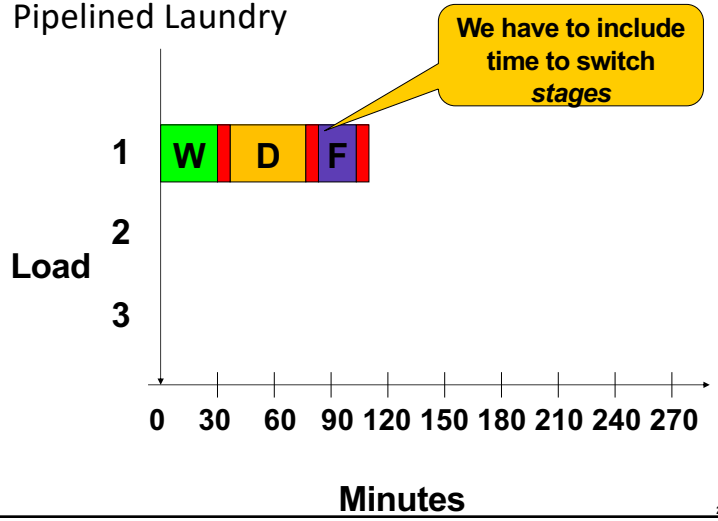
Pipelined Laundry



19

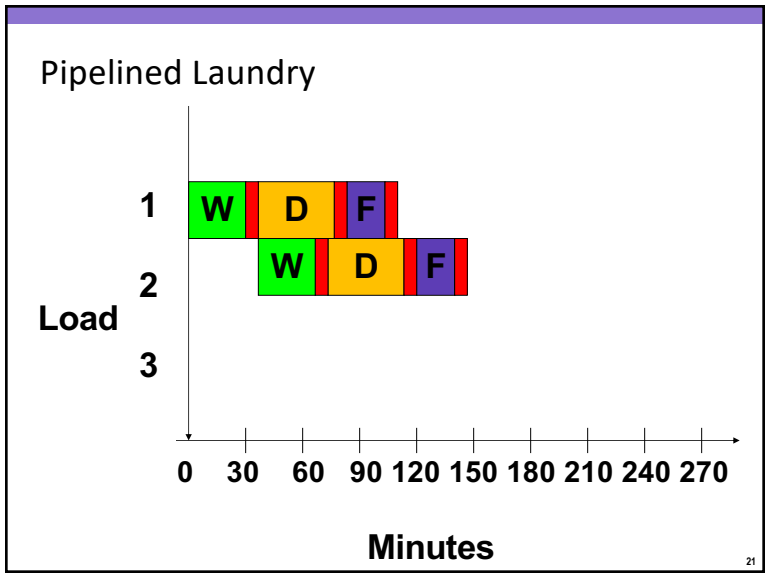
19

Pipelined Laundry

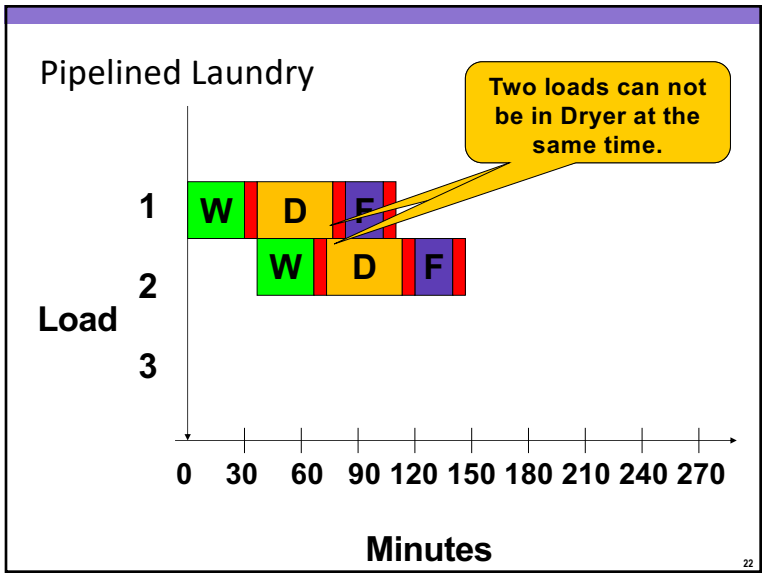


20

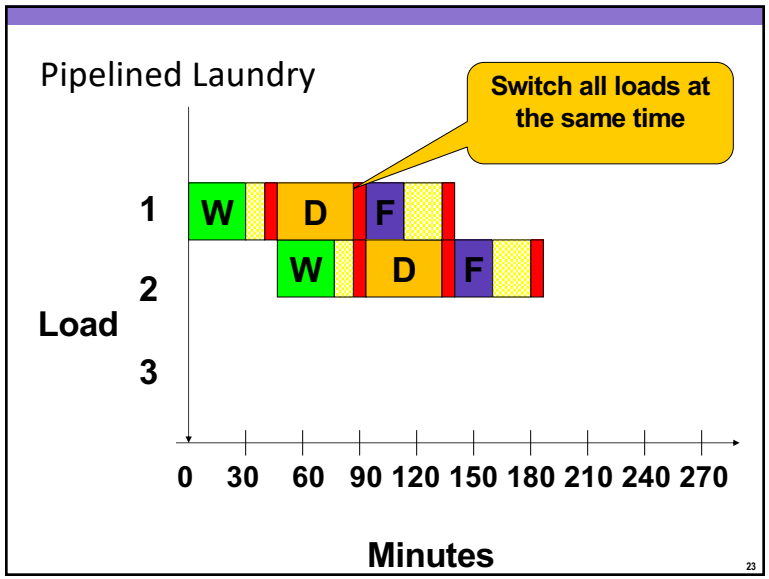
20


Pipelined Laundry
Load
1
2
3
W
D
F
W
D
F
0 30 60 90 120 150 180 210 240 270
Minutes
21

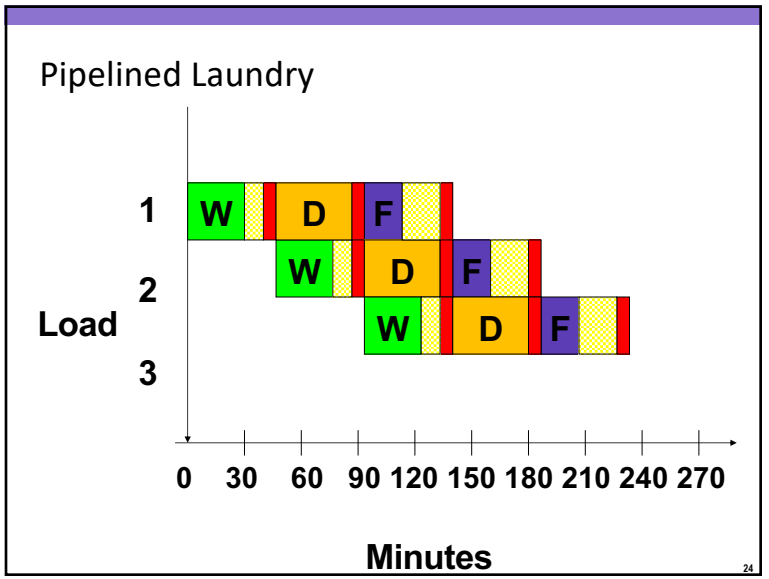
21


Pipelined Laundry
Two loads can not be in Dryer at the same time.
Load
1
2
3
W
D
F
W
D
F
0 30 60 90 120 150 180 210 240 270
Minutes
22

22


Pipelined Laundry
Switch all loads at the same time
Load
1
2
3
W
D
F
W
D
F
0 30 60 90 120 150 180 210 240 270
Minutes
23

23


Pipelined Laundry
Load
1
2
3
W
D
F
W
D
F
W
D
F
0 30 60 90 120 150 180 210 240 270
Minutes
24

24

Pipelined Laundry

- Cycle Time: Clothing is switched every ____ minutes
- Latency: A single load takes a total of _____ minutes
- Throughput: A load completes each _____ minutes
- CyclesPerLoad: Every ____ cycles, a load completes

25

25

Pipelined Laundry

- Cycle Time: Clothing is switched every 46 minutes
- Latency: A single load takes a total of _____ minutes
- Throughput: A load completes each _____ minutes
- CyclesPerLoad: Every ____ cycles, a load completes

26

26

Pipelined Laundry

- Cycle Time: Clothing is switched every 46 minutes
- Latency: A single load takes a total of 138 minutes
- Throughput: A load completes each _____ minutes
- CyclesPerLoad: Every ____ cycles, a load completes

27

27

Pipelined Laundry

- Cycle Time: Clothing is switched every 46 minutes
- Latency: A single load takes a total of 138 minutes
- Throughput: A load completes each 46 minutes
- CyclesPerLoad: Every ____ cycles, a load completes

28

28

Pipelined Laundry

- Cycle Time: Clothing is switched every 46 minutes
- Latency: A single load takes a total of 138 minutes
- Throughput: A load completes each 46 minutes
- CyclesPerLoad: Every 1 cycles, a load completes

29

29

Single-Cycle vs Pipelined

- _____ has the higher cycle time
- _____ has the higher clock rate
- _____ has the higher single-load latency
- _____ has the higher throughput
- _____ has the higher CPL (Cycles per Load)
- More stages makes a _____ clock rate

30

30

Single-Cycle vs Pipelined

- Single has the higher cycle time
- _____ has the higher clock rate
- _____ has the higher single-load latency
- _____ has the higher throughput
- _____ has the higher CPL (Cycles per Load)
- More stages makes a _____ clock rate

31

31

Single-Cycle vs Pipelined

- Single has the higher cycle time
- Pipelined has the higher clock rate
- _____ has the higher single-load latency
- _____ has the higher throughput
- _____ has the higher CPL (Cycles per Load)
- More stages makes a _____ clock rate

32

32

Single-Cycle vs Pipelined

- Single has the higher cycle time
- Pipelined has the higher clock rate
- Pipelined has the higher single-load latency
- _____ has the higher throughput
- _____ has the higher CPL (Cycles per Load)
- More stages makes a _____ clock rate

33

33

Single-Cycle vs Pipelined

- Single has the higher cycle time
- Pipelined has the higher clock rate
- Pipelined has the higher single-load latency
- Pipelined has the higher throughput
- _____ has the higher CPL (Cycles per Load)
- More stages makes a _____ clock rate

34

34

Single-Cycle vs Pipelined

- Single has the higher cycle time
- Pipelined has the higher clock rate
- Pipelined has the higher single-load latency
- Pipelined has the higher throughput
- Neither has the higher CPL (Cycles per Load)
- More stages makes a _____ clock rate

35

35

Single-Cycle vs Pipelined

- Single has the higher cycle time
- Pipelined has the higher clock rate
- Pipelined has the higher single-load latency
- Pipelined has the higher throughput
- Neither has the higher CPL (Cycles per Load)
- More stages makes a Higher clock rate

36

36

Obstacles to speedup in Pipelining

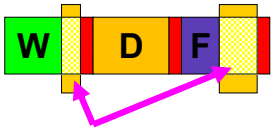
- 1.
- 2.

- Ideal cycle time w/out above limitations with n stage pipeline:

37

37

Obstacles to speedup in Pipelining



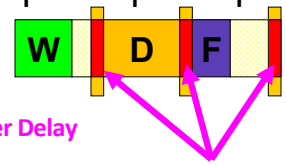
- 1. **Uneven Stages**
- 2.

- Ideal cycle time w/out above limitations with n stage pipeline:

38

38

Obstacles to speedup in Pipelining



- 1. Uneven Stages
- 2. **Pipeline Register Delay**

- Ideal cycle time w/out above limitations with n stage pipeline:

39

39

Obstacles to speedup in Pipelining

- 1. Uneven Stages
- 2. Pipeline Register Delay

- Ideal cycle time w/out above limitations with n stage pipeline:
  - **OldCycleTime / n**

40

40

Practice On Your Own

- Washing = 45
- Drying = 70
- Folding = 15
- Switching = 5

- What is the latency for one load of laundry?
- What is the latency for three loads?

	Latency 1 load	Latency 3 loads
Non-pipelined		
Pipelined		

41

41

Today: The Y86 Pipelined Datapath

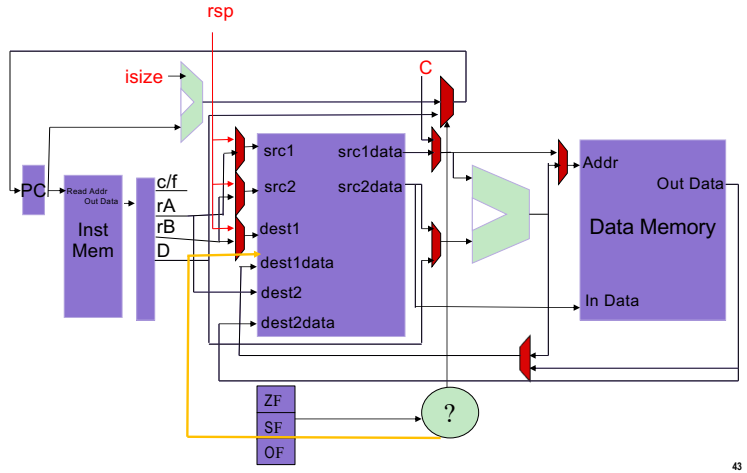
- Pipelining Concepts
- Construction of a pipelined datapath for Y86
  - Adding pipeline registers
  - Data hazards
  - Control hazards

42

42

Seq Datapath

Entire instruction takes 1 cycle



43

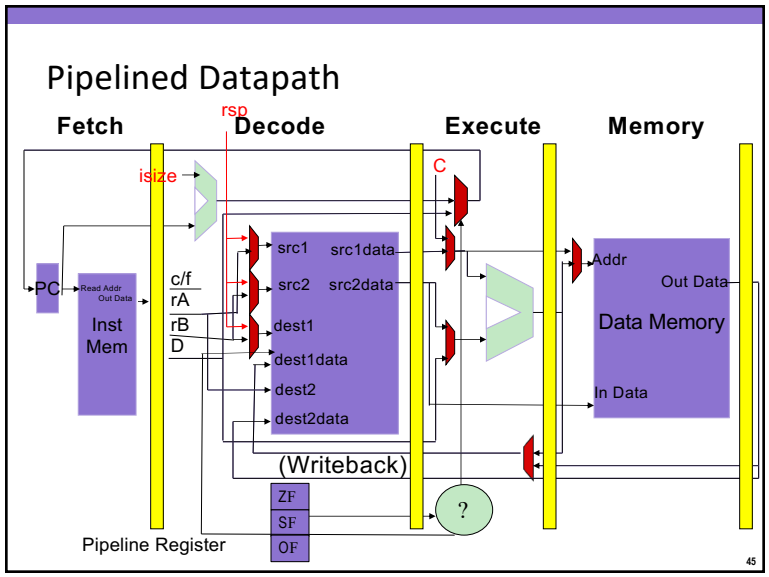
43

Creating Stages to transform Seq Design

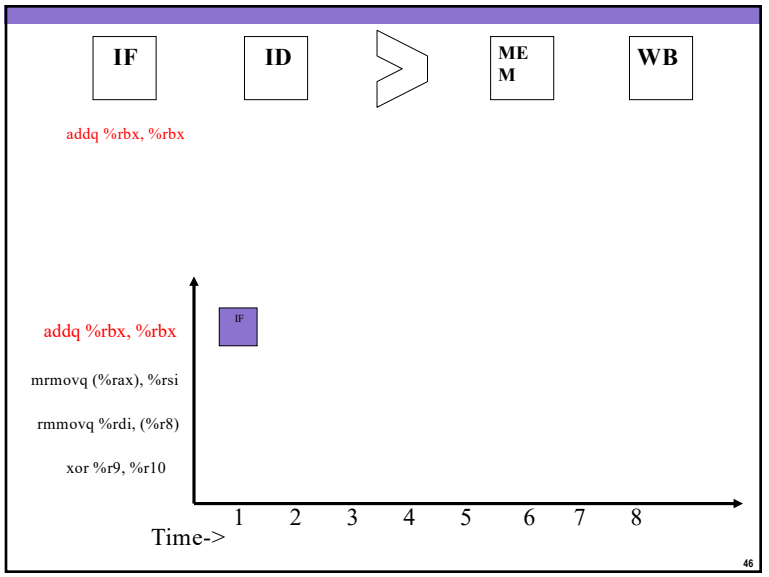
- Fetch – get instruction
- Decode – read registers
- Execute – use ALU
- Memory – access memory
- WriteBack – write registers

44

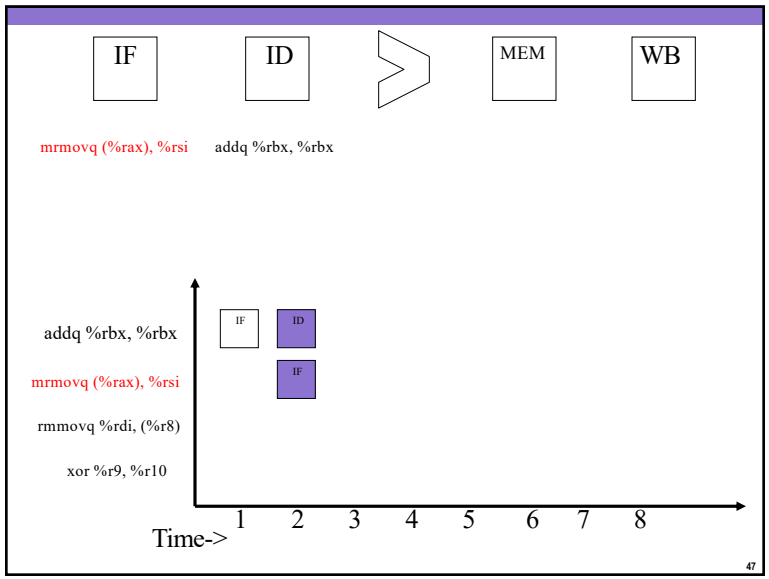
44


Pipelined Datapath
Fetch
Decode
Execute
Memory
rsp
isize
C
PC
Read Addr
Out Data
Inst
Mem
c/f
rA
rB
D
src1
src1data
src2
src2data
dest1
dest1data
dest2
dest2data
(Writeback)
ZF
SF
OF
?
Addr
Out Data
Data Memory
In Data
Pipeline Register
45

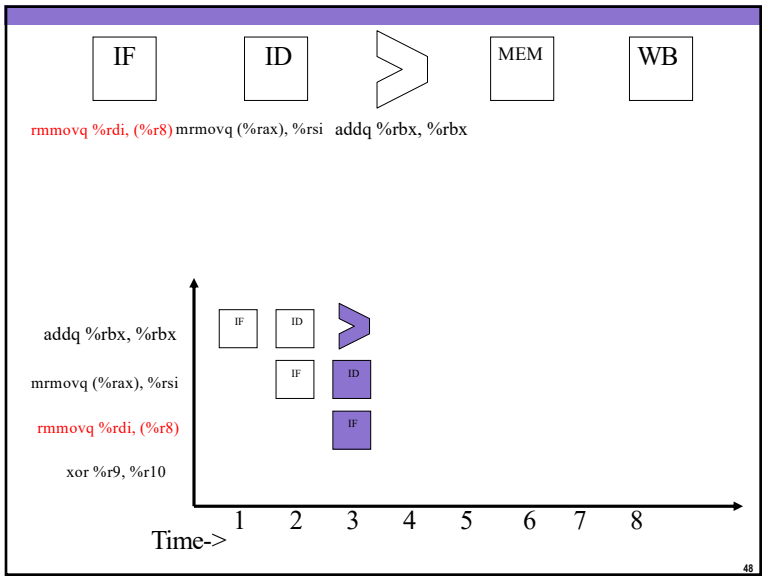
45


IF
ID
ME
M
WB
addq %rbx, %rbx
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
IF
1 2 3 4 5 6 7 8
Time->
46

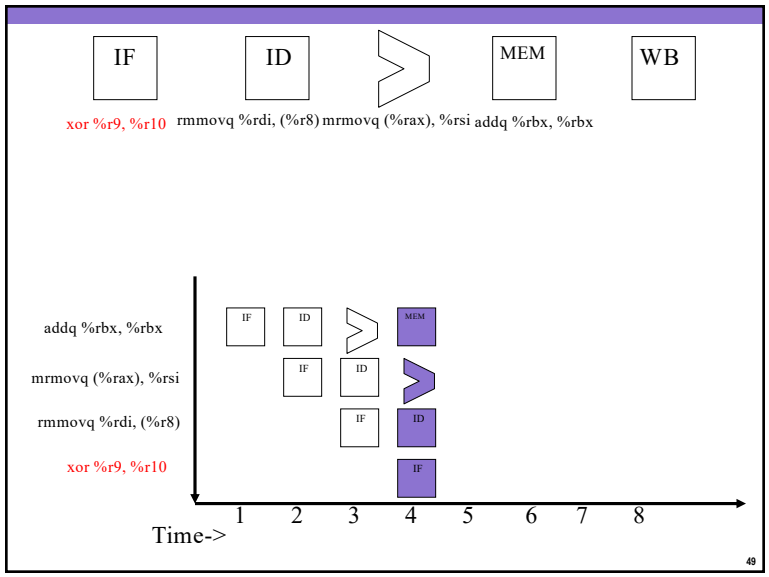
46


IF
ID
MEM
WB
mrmovq (%rax), %rsi
addq %rbx, %rbx
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
IF
ID
IF
1 2 3 4 5 6 7 8
Time->
47

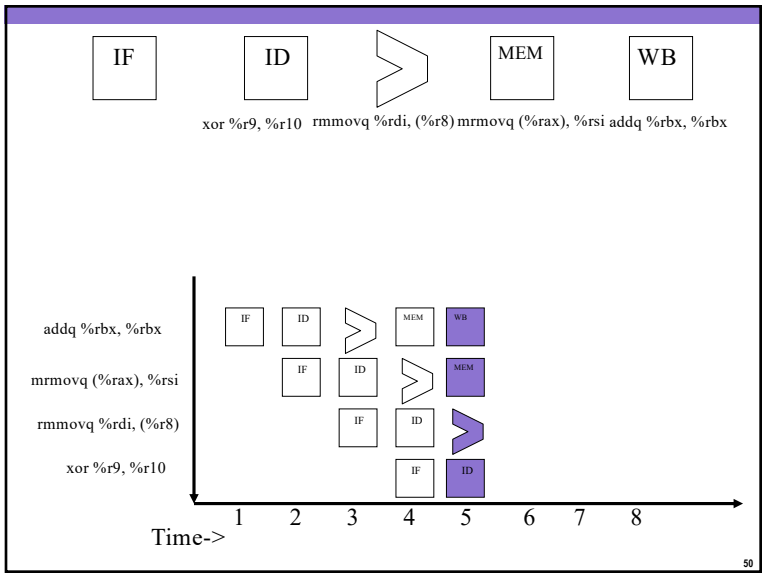
47


IF
ID
MEM
WB
rmmovq %rdi, (%r8)
mrmovq (%rax), %rsi
addq %rbx, %rbx
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
IF
ID
IF
ID
IF
1 2 3 4 5 6 7 8
Time->
48

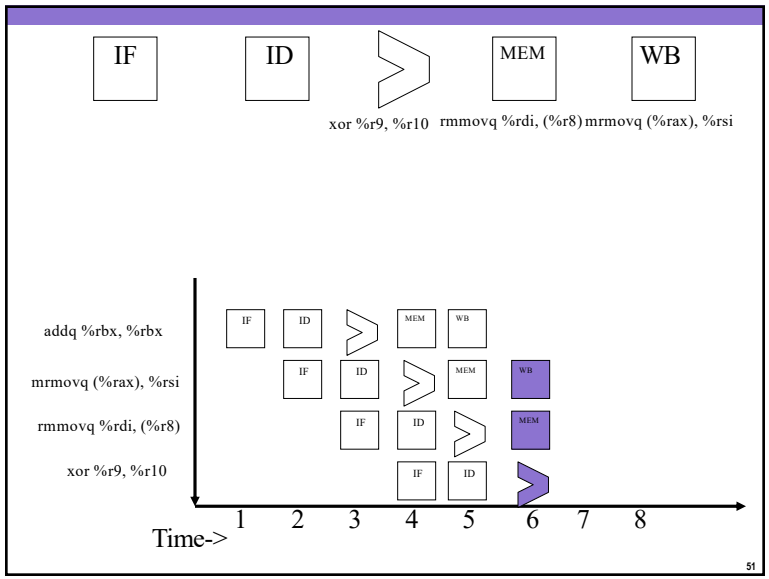
48


IF
ID
MEM
WB
xor %r9, %r10
rmmovq %rdi, (%r8)
mrmovq (%rax), %rsi
addq %rbx, %rbx
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
49

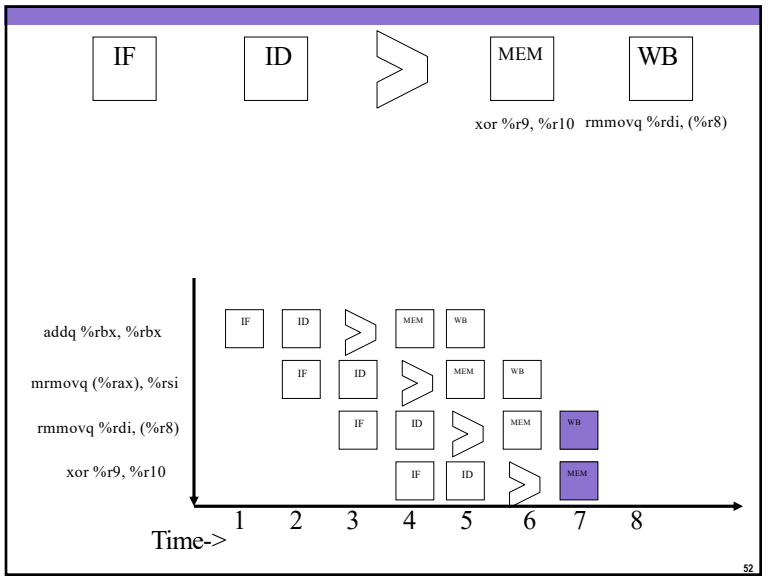
49


IF
ID
MEM
WB
xor %r9, %r10
rmmovq %rdi, (%r8)
mrmovq (%rax), %rsi
addq %rbx, %rbx
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
50

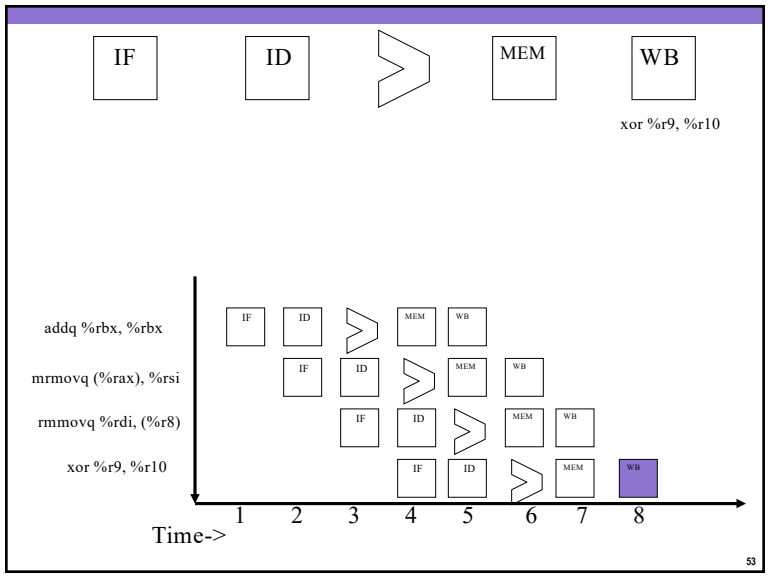
50


IF
ID
MEM
WB
xor %r9, %r10
rmmovq %rdi, (%r8)
mrmovq (%rax), %rsi
addq %rbx, %rbx
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
51

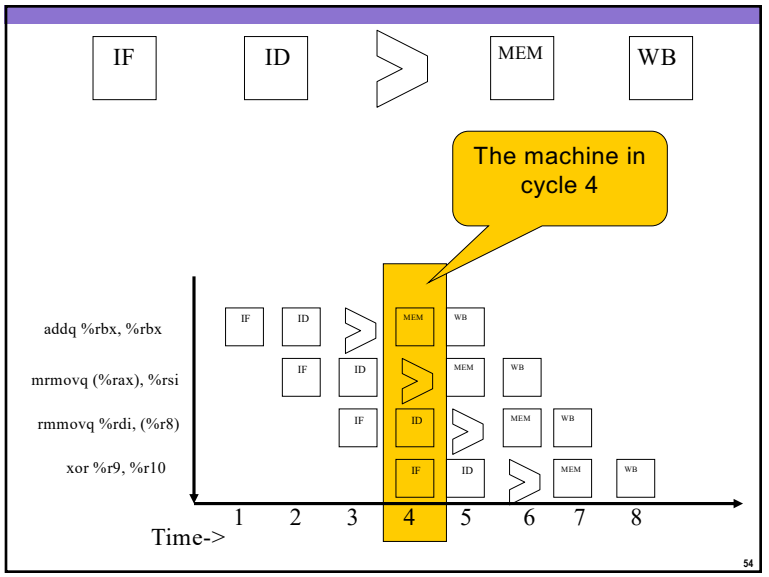
51


IF
ID
MEM
WB
xor %r9, %r10
rmmovq %rdi, (%r8)
addq %rbx, %rbx
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
52

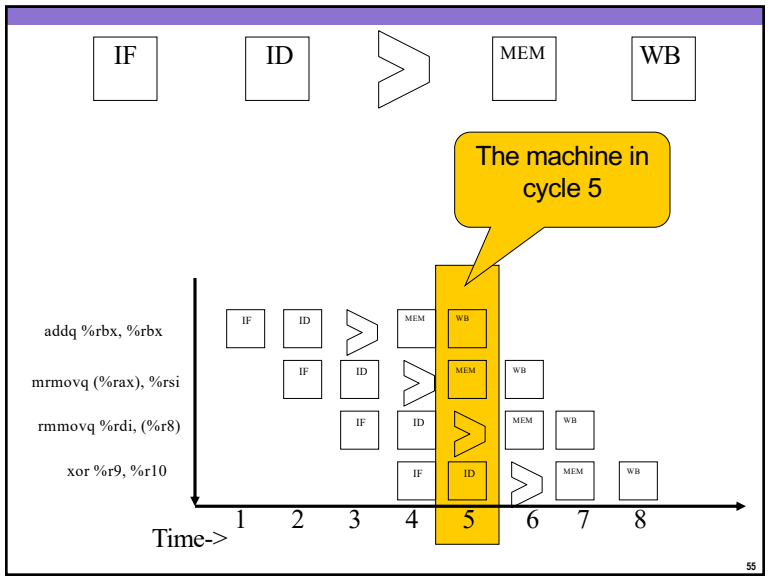
52


IF
ID
MEM
WB
xor %r9, %r10
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
53

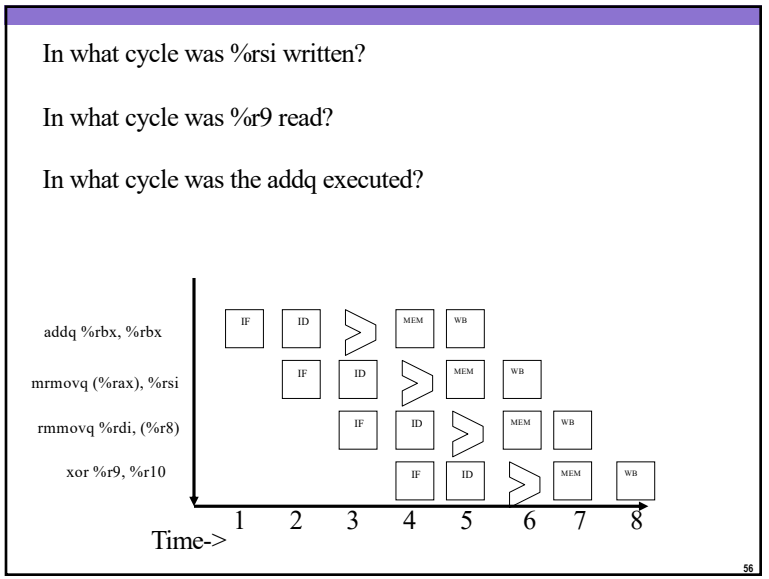
53


IF
ID
MEM
WB
The machine in cycle 4
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
54

54


IF
ID
MEM
WB
The machine in cycle 5
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
55

55

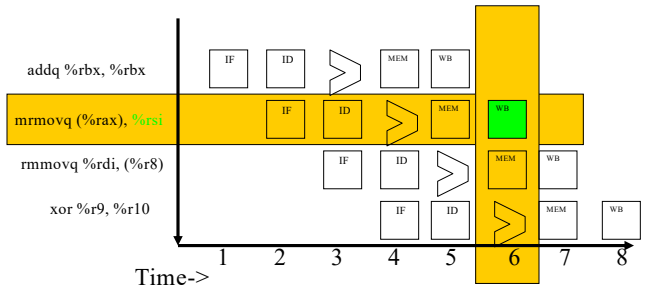

In what cycle was %rsi written?
In what cycle was %r9 read?
In what cycle was the addq executed?
addq %rbx, %rbx
mrmovq (%rax), %rsi
rmmovq %rdi, (%r8)
xor %r9, %r10
1 2 3 4 5 6 7 8
Time->
56

56

In what cycle was %rsi written? 6

In what cycle was %r9 read?

In what cycle was the addq executed?

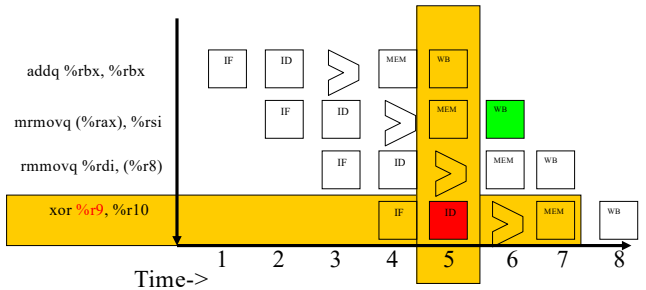


57

In what cycle was %rsi written? 6

In what cycle was %r9 read? 5

In what cycle was the Add executed?

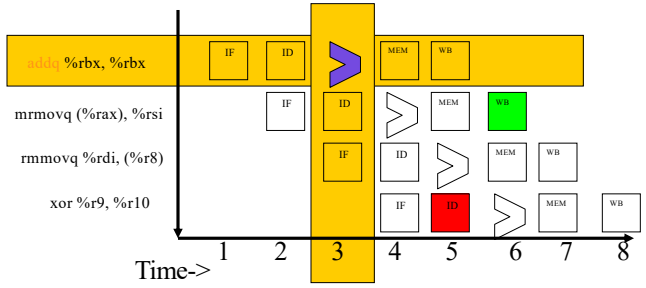


58

In what cycle was %rsi written? 6

In what cycle was %r9 read? 5

In what cycle was the addq executed? 3

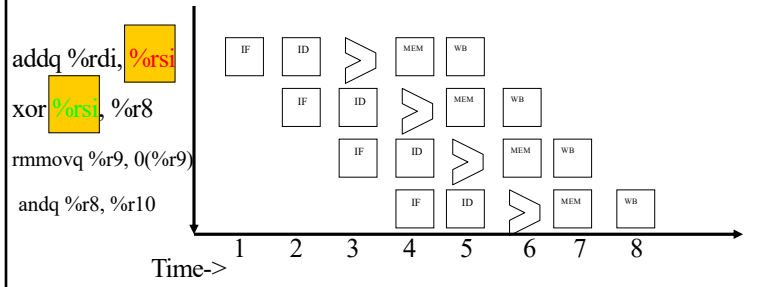


59

Incorrect Execution

Easy Right? Not so fast.

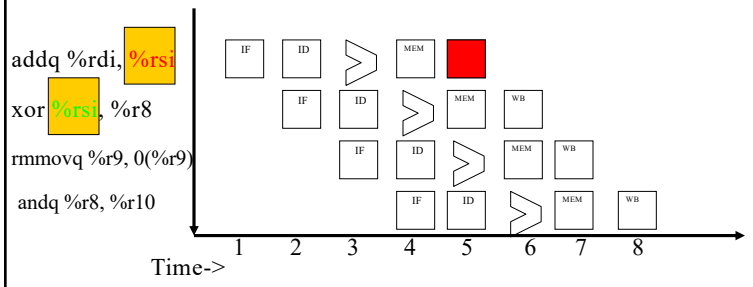
In what cycle does the addq write %rsi?
In what cycle does the xor read %rsi?



60

Easy Right? Not so fast.

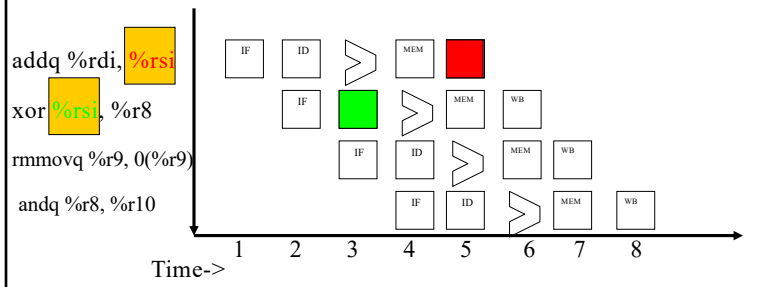
In what cycle does the addq write %rsi? cycle 5
In what cycle does the xor read %rsi?



61

Easy Right? Not so fast.

In what cycle does the addq write %rsi? cycle 5
In what cycle does the xor read %rsi? cycle 3

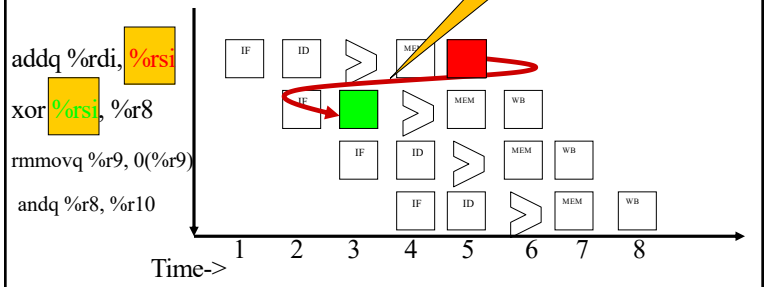


62

Easy Right? Not so fast.

In what cycle does the addq write %rsi? cycle 5
In what cycle does the xor read %rsi? cycle 3

Ahhhh! Values can not pass backwards in time



63

How Could We Solve this Problem?

- Compiler could add `nop` instructions before later instruction
- We can add circuitry to detect the problem and stall the second instruction

64

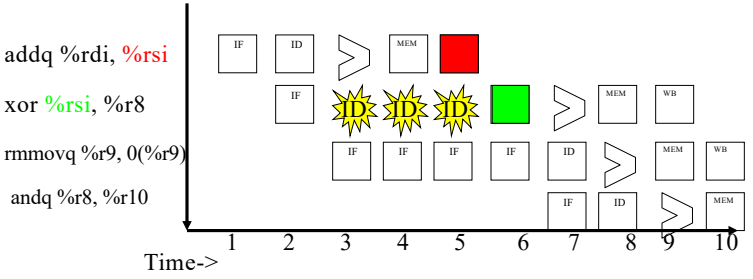
64

Correct, Slow Execution

Easy Right? Not so fast.

In what cycle does the addq write %rsi? cycle 5
In what cycle does the xor read %rsi? cycle 6

Stall - wasted cycles



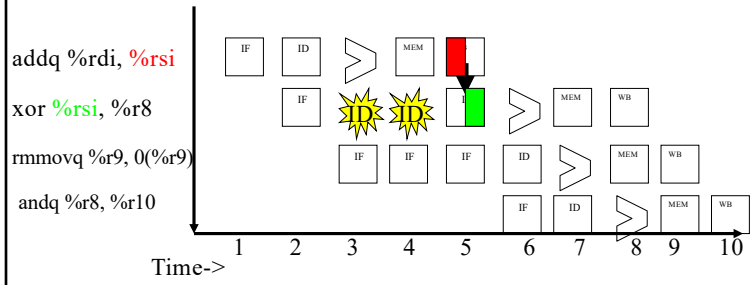
65

Correct, Slow Execution

Easy Right? Not so fast.

In what cycle does the addq write %rsi? 1st half of cycle 5
In what cycle does the xor read %rsi? 2nd half of cycle 5

Stall - wasted cycles



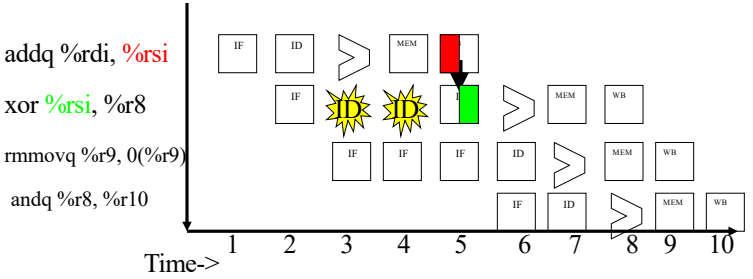
66

Correc

Only Register File rd/wr in half a cycle. All other stages take a full cycle – this is because of shared hardware

In what cycle does the addq write %rsi? 1st half of cycle 5
In what cycle does the xor read %rsi? 2nd half of cycle 5

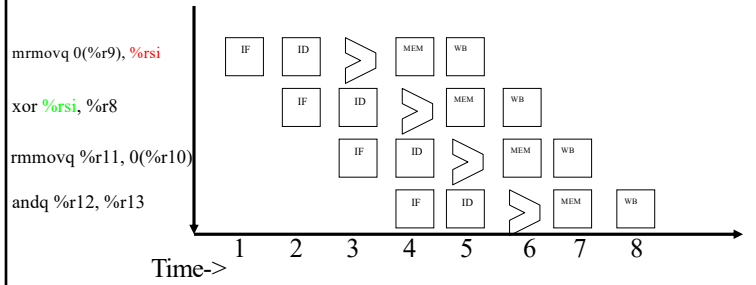
Stall - wasted cycles



67

Incorrect Execution caused by Data Hazard

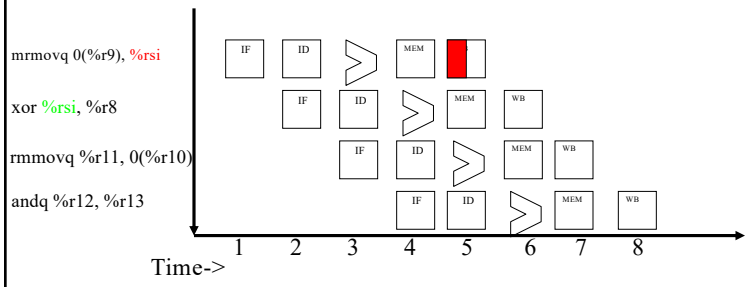
In what cycle does the mrmovq write %rsi?
In what cycle does the xor read %rsi?



69

Incorrect Execution caused by Data Hazard

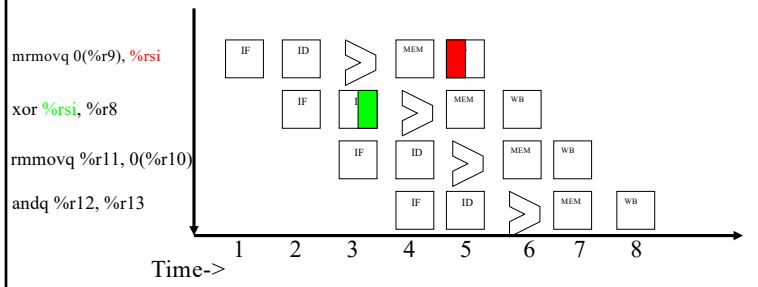
In what cycle does the mrmovq write %rsi? 1st half of cycle 5
In what cycle does the xor read %rsi?



70

Incorrect Execution caused by Data Hazard

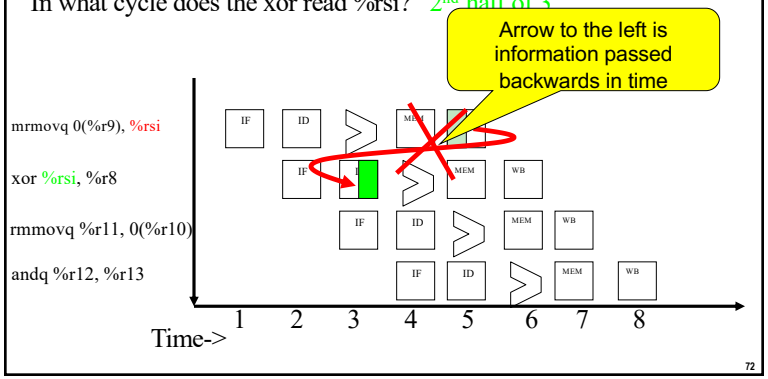
In what cycle does the mrmovq write %rsi? 1st half of cycle 5
In what cycle does the xor read %rsi? 2nd half of cycle 3



71

Incorrect Execution caused by Data Hazard

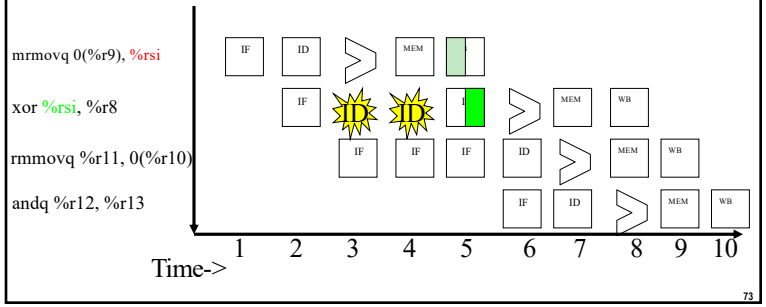
In what cycle does the mrmovq write %rsi? 1st half of 5
In what cycle does the xor read %rsi? 2nd half of 3



72

Incorrect Execution caused by Data Hazard

In what cycle does the mrmovq write %rsi? 1st half of 5
In what cycle does the xor read %rsi? 2nd half of 3

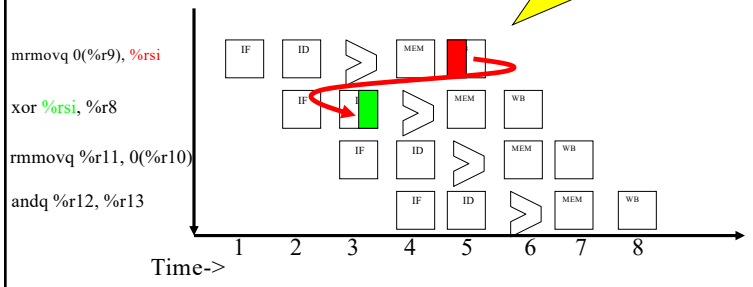


73

Data Hazard

In what cycle does the mrmovq write %rsi? 1
In what cycle does the xor read %rsi? 2nd ha

Remember!!! Only WB and ID take ½ cycle. All other stages take a full cycle!!!!!!!!!



74

74

Barriers to pipelined performance

- Uneven stages
- Pipeline register delays

75

75

Barriers to pipelined performance

- Uneven stages
- Pipeline register delays
- Data Hazards

76

76

Barriers to pipeline performance

- Uneven stages
- Pipeline register delays
- Data Hazards
  - An instruction depends on the result of a previous instruction still in the pipeline and that dependence has the potential to cause erroneous computation

77

77

Read After Write (RAW) Data Dependences

- When a later instruction depends on the result of an earlier instruction
- If instructions are close enough in pipeline, later instruction may need to be stalled to ensure correctness

78

78

RAW – Read after Write

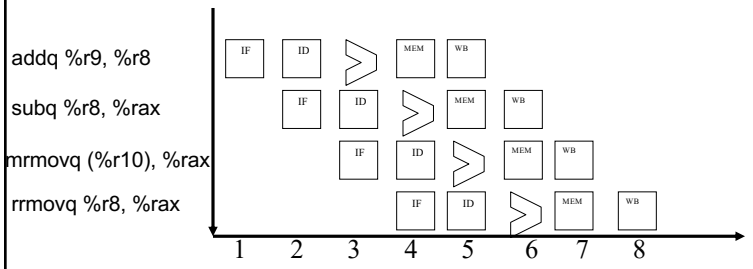
```
addq   %r8, %rsi
subq   %rsi, %r9
xor    %rax, %rax
addq   %rdi, %rdi
```

79

79

Identify the RAW dependences

RAW

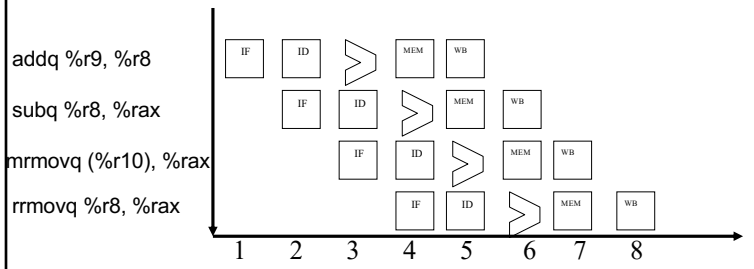


80

80

Identify the RAW dependences

RAW addq/subq %r8
RAW addq/rrmovq %r8



81

81

Practice Together

- Consider the Y86-64 code below. Are there any potential problems due to data hazards in this code? Draw the pipeline diagram, assuming the 5 stage stalling pipeline.

```
mrmovq      (%rdi), %r8
irmovq      $4, %r9
addq        %r9, %r8
rmmovq      %r8, (%rdi)
mrmovq      (%rdi), %rax
```

82

82

Today: The Y86 Pipelined Datapath

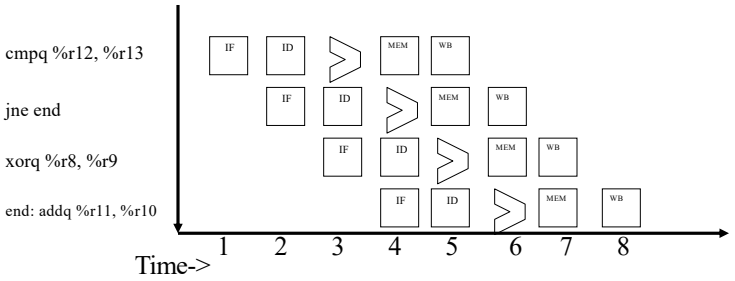
- Pipelining Concepts
- Construction of a pipelined datapath for Y86
  - Adding pipeline registers
  - Data hazards
  - Control hazards

83

83

Control Hazard

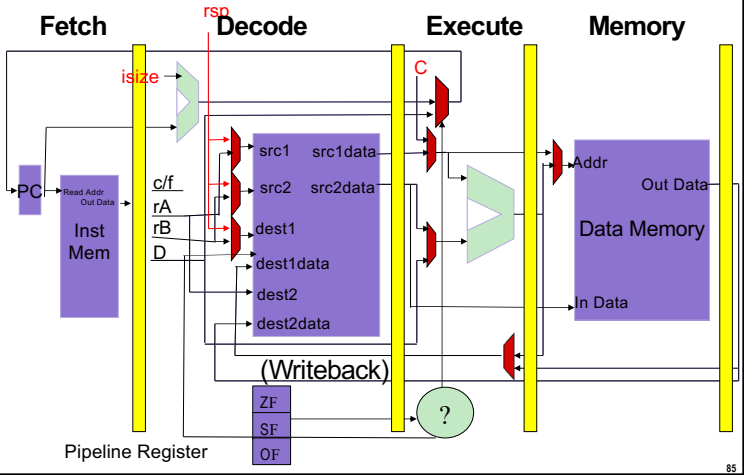
In what cycle does the nextPC get calculated for the `jne`?
In what cycle does the `xorq` get fetched?



84

84

Pipelined Datapath

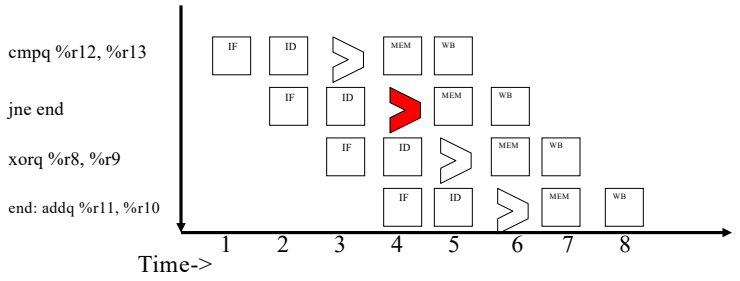


85

85

Control Hazard

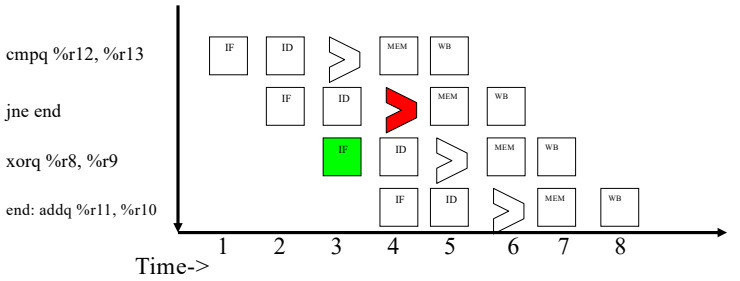
In what cycle does the nextPC get calculated for the `jne`? End of 4
In what cycle does the `xorq` get fetched?



86

Control Hazard

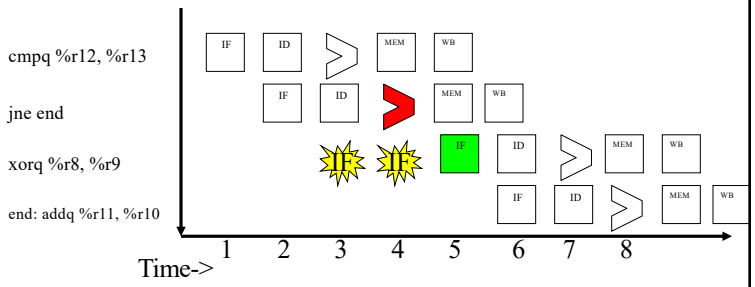
In what cycle does the nextPC get calculated for the `jne`? End of 4
In what cycle does the `xorq` get fetched? Beginning of 3



87

Control Hazard: Stall until target known

In what cycle does the nextPC get calculated for the `jne`? End of 4
In what cycle does the `xorq` get fetched? Beginning of 3



88

Barriers to Pipeline Performance

- Uneven stages
- Pipeline register delays
- Data Hazards
- Control Hazards
  - ***Whether*** an instruction will execute depends on the outcome of a control instruction still in the pipeline

89

89