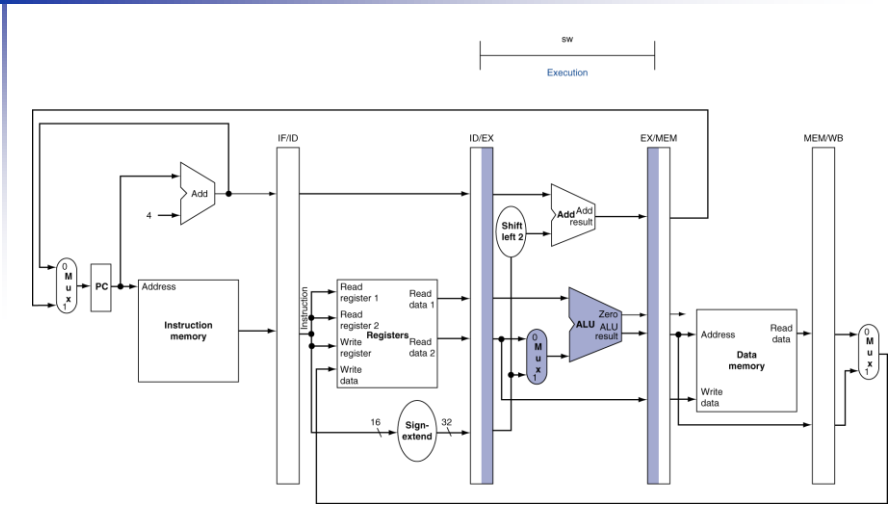


EX for Store



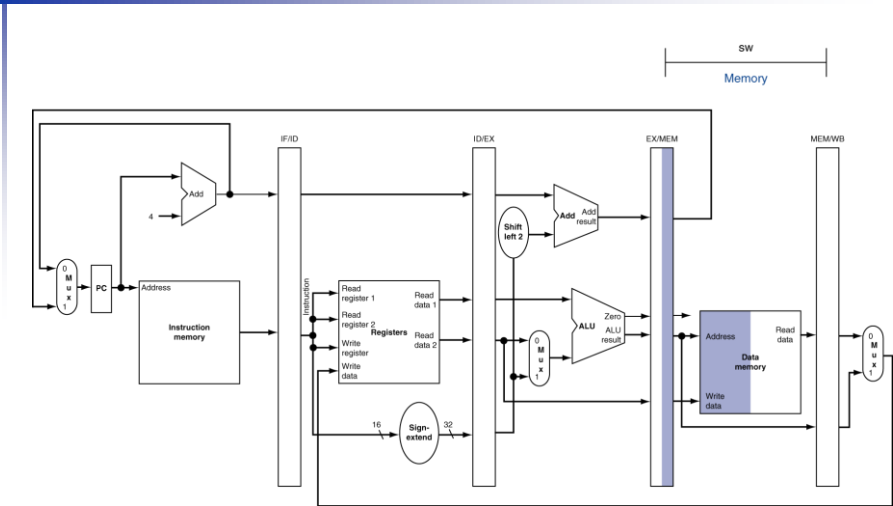
Assignment Project Exam Help

Chapter 4 — The Processor — 88

<https://powcoder.com>

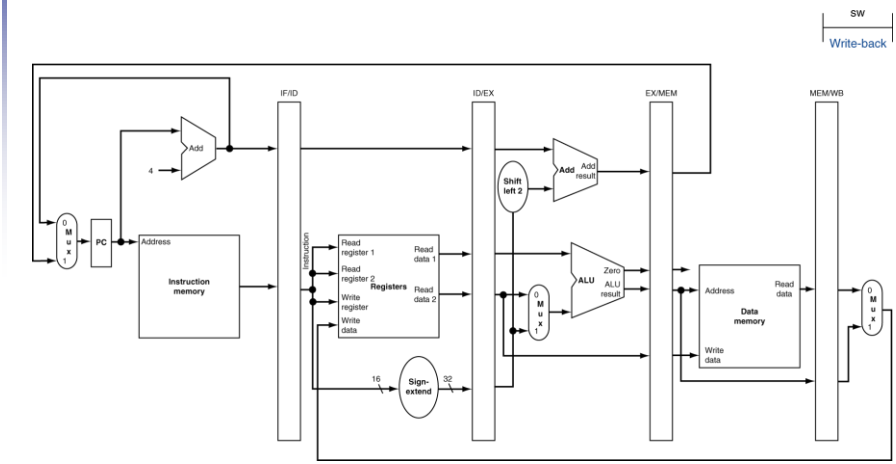
Add WeChat powcoder

MEM for Store



Chapter 4 — The Processor — 89

WB for Store

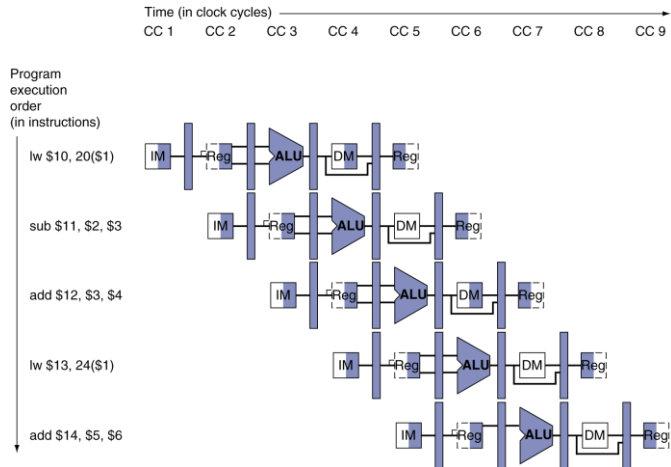


Assignment Project Exam Help

<https://powcoder.com>

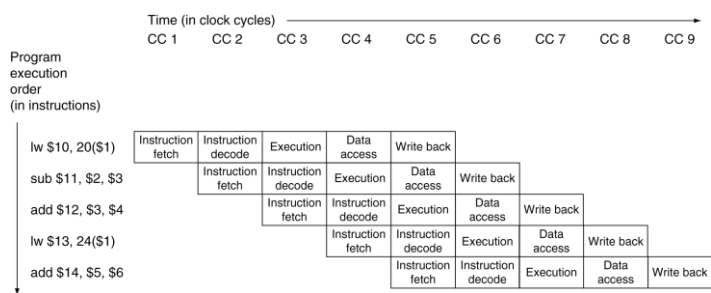
Add WeChat powcoder
Multi-Cycle Pipeline Diagram

■ Form showing resource usage



Multi-Cycle Pipeline Diagram

■ Traditional form



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Pop Quiz

- How many cycles would it take the following instructions to complete?

■ A: 5

■ B: 6

■ C: 7

■ D: 8

add \$t2, \$t0, \$t1

lw \$t3, 0(\$t2)

add \$t4, \$t5, \$t3

Pop Quiz (answer)

1. add \$t2, \$t0, \$t1
2. lw \$t3, 0(\$t2)
3. add \$t4, \$t5, \$t3

1	2	3	4	5	6	7	8

F	D	X	M	W			
	F	D	X	M	W		
		*	*	*	*	*	
			F	D	X	M	W

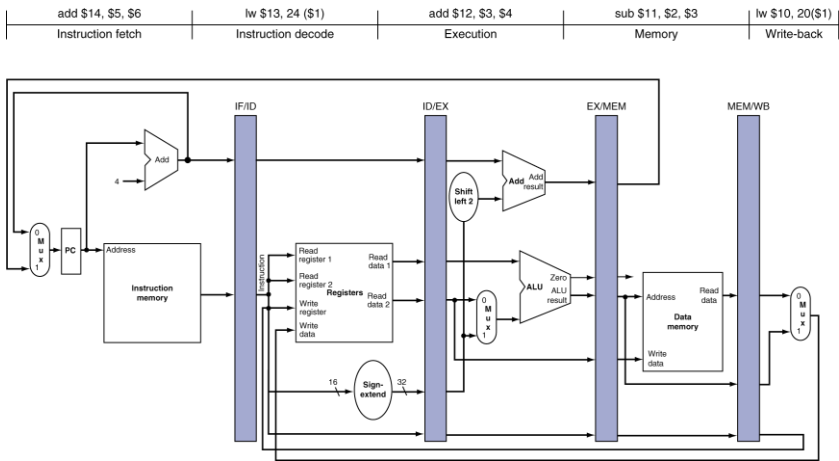
Assignment Project Exam Help

<https://powcoder.com>

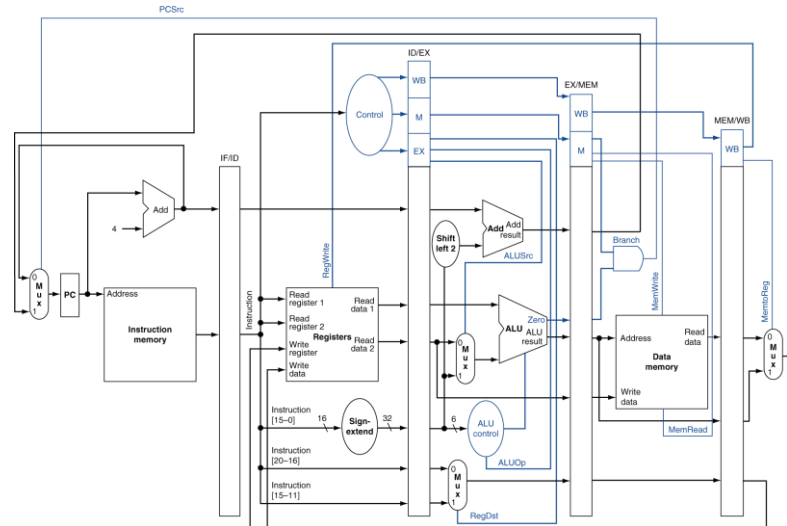
Add WeChat powcoder

Single-Cycle Pipeline Diagram

State of pipeline in a given cycle



Pipelined Control



Assignment Project Exam Help

<https://powcoder.com>

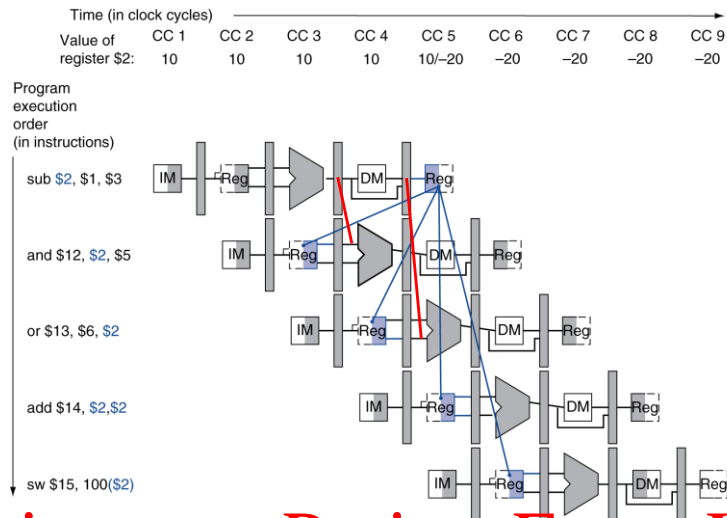
Add WeChat powcoder Data Hazards in ALU Instructions

- Consider this sequence:

```
sub $2, $1,$3
and $12,$2,$5
or  $13,$6,$2
add $14,$2,$2
sw  $15,100($2)
```
- We can resolve hazards with forwarding
 - How do we detect when to forward?

§4.7 Data Hazards: Forwarding vs. Stalling

Dependencies & Forwarding



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Detecting the Need to Forward

- Pass register numbers along pipeline
 - e.g., ID/EX.RegisterRs = register number for Rs sitting in ID/EX pipeline register
- ALU operand register numbers in EX stage are given by
 - ID/EX.RegisterRs, ID/EX.RegisterRt
- Data hazards when
 - 1a. EX/MEM.RegisterRd = ID/EX.RegisterRs
 - 1b. EX/MEM.RegisterRd = ID/EX.RegisterRt
 - 2a. MEM/WB.RegisterRd = ID/EX.RegisterRs
 - 2b. MEM/WB.RegisterRd = ID/EX.RegisterRt

Fwd from
EX/MEM
pipeline reg

Fwd from
MEM/WB
pipeline reg

Detecting the Need to Forward

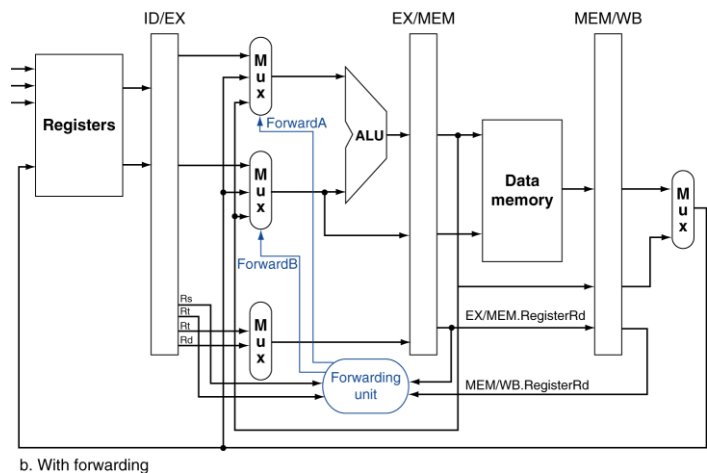
- But only if forwarding instruction will write to a register!
 - EX/MEM.RegWrite, MEM/WB.RegWrite
- And only if Rd for that instruction is not \$zero
 - EX/MEM.RegisterRd ≠ 0, MEM/WB.RegisterRd ≠ 0

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Forwarding Paths



Forwarding Conditions

- EX hazard
 - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0) and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
ForwardA = 10
 - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0) and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
ForwardB = 10
- MEM hazard
 - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0) and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
ForwardA = 01
 - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0) and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
ForwardB = 01

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Pop Quiz

- We can always “forward” our way around instruction dependencies:
- A: True
- B: False

Double Data Hazard

- Consider the sequence:
 - add \$1, \$1, \$2
 - add \$1, \$1, \$3
 - add \$1, \$1, \$4
- Both hazards occur
 - Want to use the most recent
- Revise MEM hazard condition
 - Only fwd if EX hazard condition isn't true

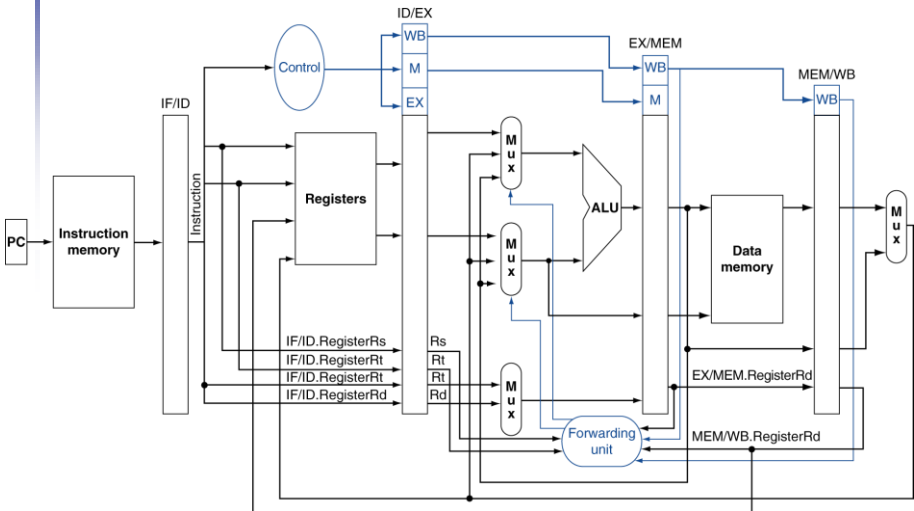
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder Revised Forwarding Condition

- MEM hazard
 - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
ForwardA = 01
 - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
ForwardB = 01

Datapath with Forwarding

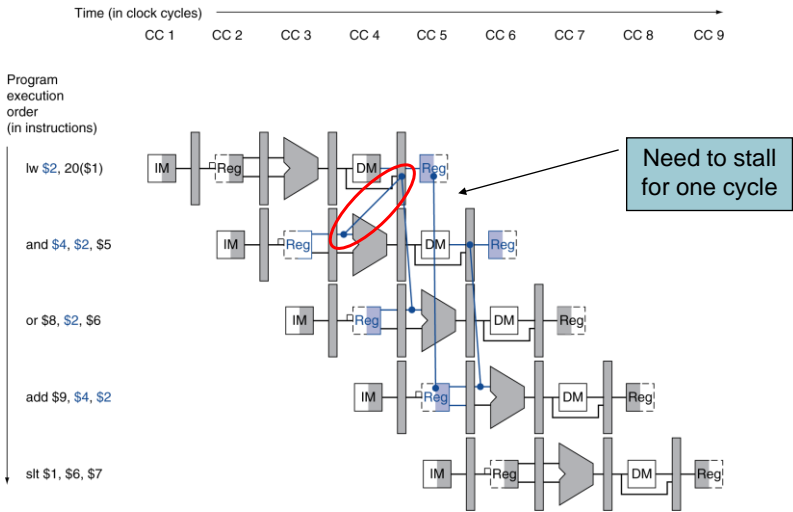


Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Load-Use Data Hazard



Chapter 4 — The Processor — 109

Load-Use Hazard Detection

- Check when using instruction is decoded in ID stage
- ALU operand register numbers in ID stage are given by
 - IF/ID.RegisterRs, IF/ID.RegisterRt
- Load-use hazard when
 - ID/EX.MemRead and $((ID/EX.RegisterRt = IF/ID.RegisterRs)$ or $(ID/EX.RegisterRt = IF/ID.RegisterRt))$
- If detected, stall and insert bubble

Assignment Project Exam Help

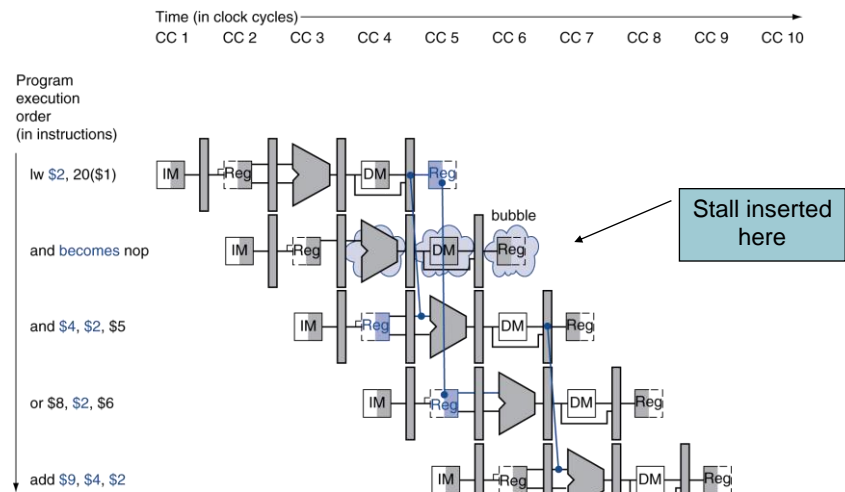
<https://powcoder.com>

Add WeChat powcoder

How to Stall the Pipeline

- Force control values in ID/EX register to 0
 - EX, MEM and WB do nop (no-operation)
- Prevent update of PC and IF/ID register
 - Using instruction is decoded again
 - Following instruction is fetched again
 - 1-cycle stall allows MEM to read data for 1w
 - Can subsequently forward to EX stage

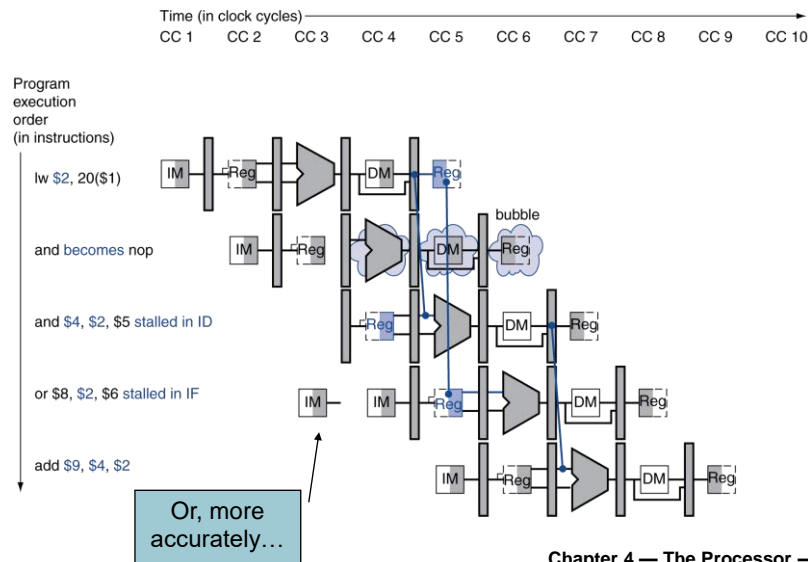
Stall/Bubble in the Pipeline



Assignment Project Exam Help

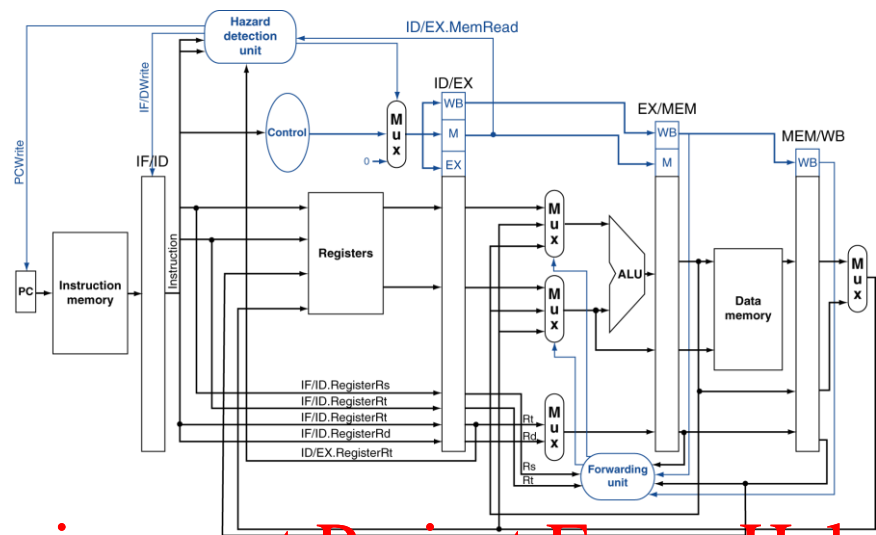
<https://powcoder.com>

Add WeChat powcoder Stall/Bubble in the Pipeline



Chapter 4 — The Processor — 113

Datapath with Hazard Detection



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Stalls and Performance

The BIG Picture

- Stalls reduce performance
 - But are required to get correct results
- Compiler can arrange code to avoid hazards and stalls
 - Requires knowledge of the pipeline structure