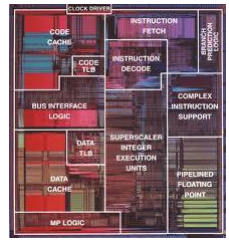
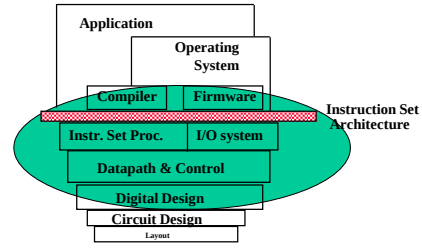
UTD

# CS/SE 3340
# Computer Architecture




## Pipelined Datapath and Control

*Adapted from slides by Profs. D. Patterson, J. Hennessey and M. Irwin*

Arlington GM plant:

*"Approximately 1,200 vehicles are produced daily –3/2017"*

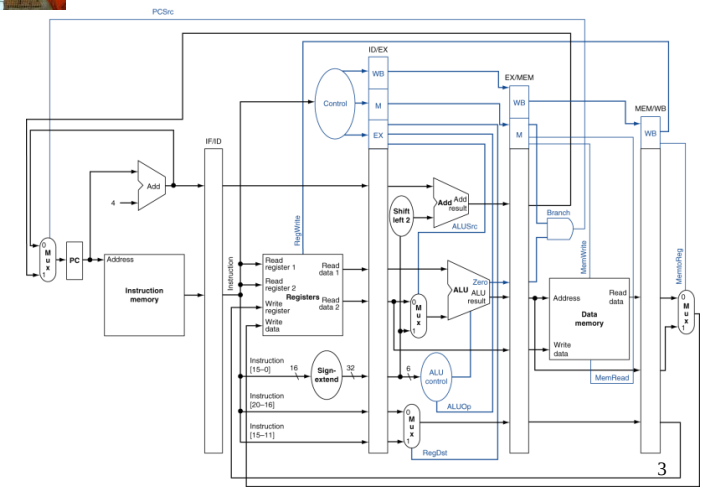
one SUV every ~72 seconds!!!

http://media.gm.com/media/us/en/gm/company_info/facilities/assembly/arlington.html

*"In October, the plant set a production record, building 31,982 SUVs. It expected total output to exceed 300,000 vehicles in 2015"*

one SUV every ~85 seconds!!!

http://www.gosanangelo.com/business/expanded-arlington-gm-plant-rolls-with-suv-sales-28d893d4-dd1a-1ddc-e053-0100007ffc40-364707301.html



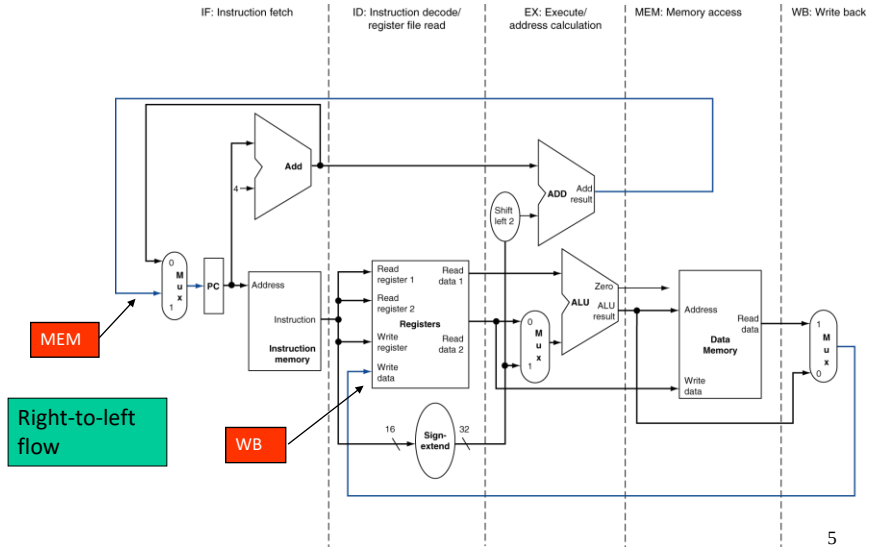

3

# Pipeline Recap

- Pipelining improves performance by increasing instruction throughput
  - Executes multiple instructions in parallel
  - Each instruction has the same latency
- Subject to hazards
  - *Structure, data, control*
- Instruction set design affects complexity of pipeline implementation

4

# MIPS Pipelined Datapath



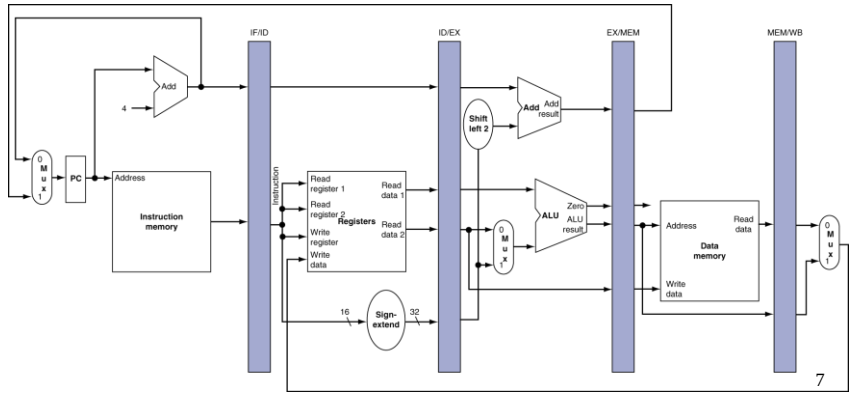



# Pipelined Data Path

- Data flows from left to right on the pipeline
- There are *two exceptions*:
  1. WB that writes the result back into the register file
  2. Selection of the next value of the PC, one input comes from the calculated branch address from the MEM stage
- Later instructions in the pipeline can be influenced by these two right-to-left data movements
  - The first one (WB to ID) leads to data hazards
  - The second one (MEM to IF) leads to control hazards

# Pipeline Registers

- Need registers between stages
  - To hold information produced in previous cycle, e.g. the destination register address, control signals



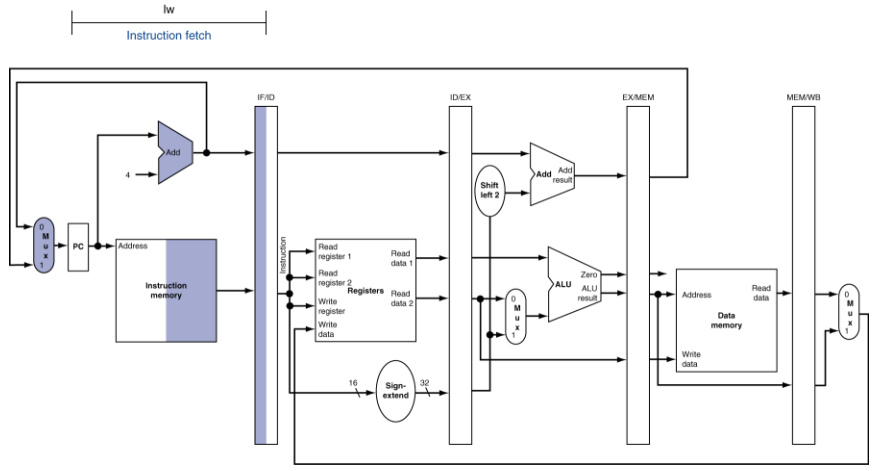

7

# Pipeline Operation

- Cycle-by-cycle flow of instructions through the pipelined datapath
  - "Single-clock-cycle" pipeline diagram
    - Shows pipeline usage in *a single cycle*
    - Highlight resources used
  - c.f. "multi-clock-cycle" diagram
    - Graph of operation over time
- We'll look at "single-clock-cycle" diagrams for *load & store* instructions

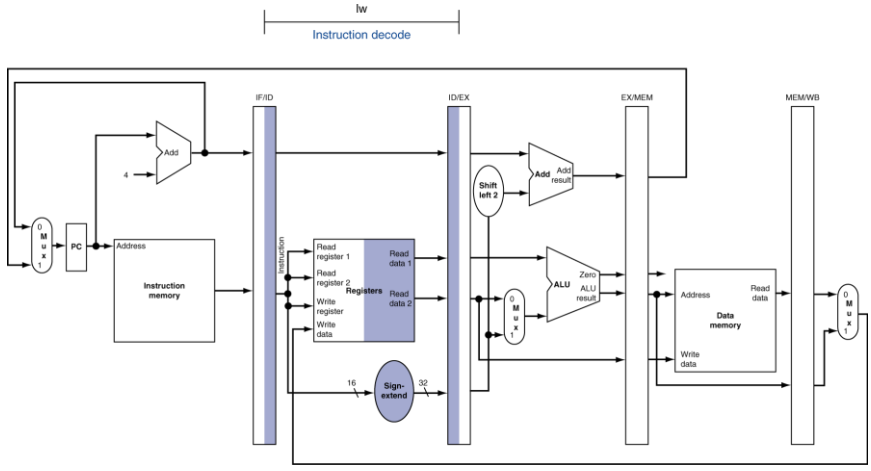
8

# IF for Load, Store, …



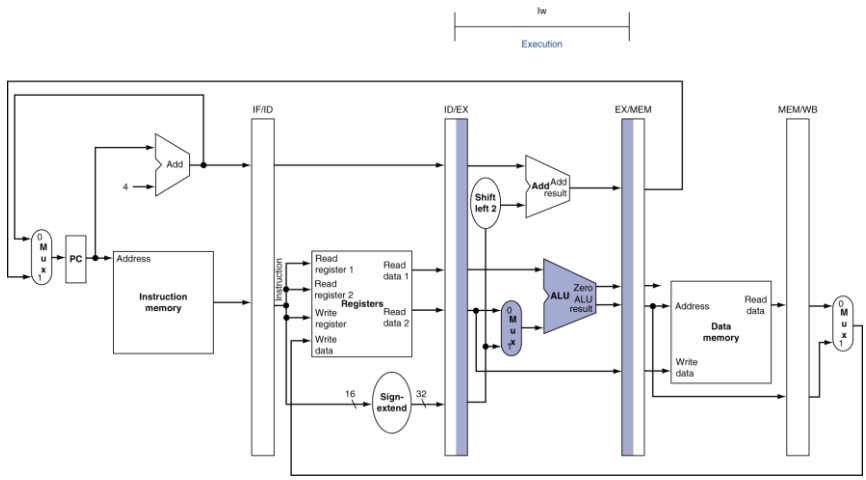

9

# ID for Load, Store, …



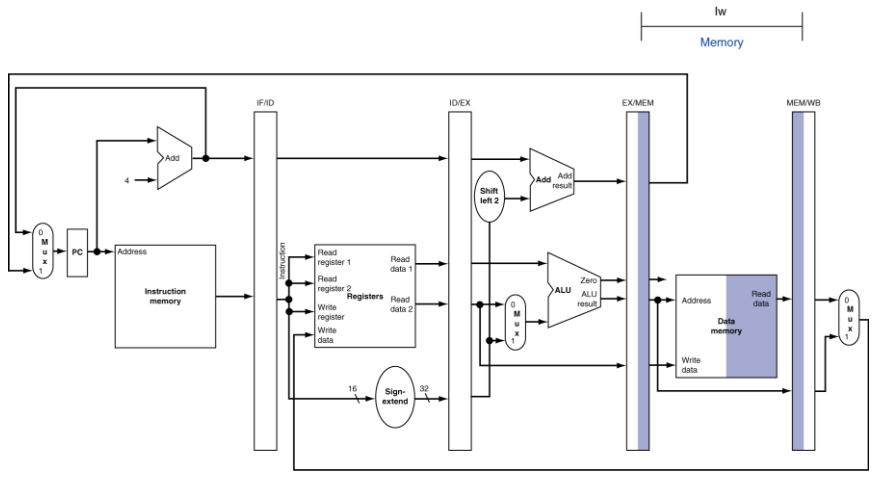

10

# EX for Load



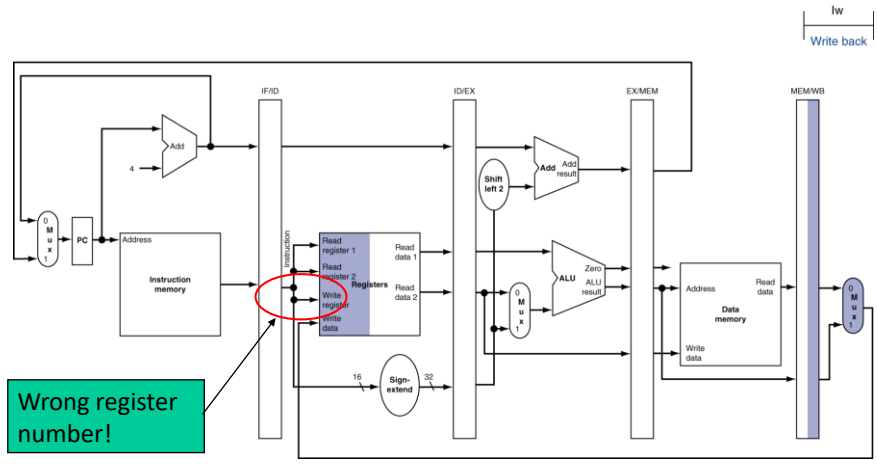

11

# MEM for Load




12

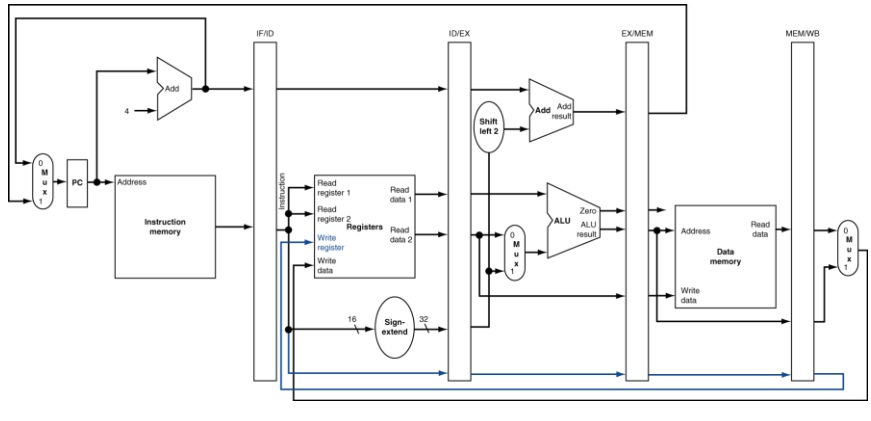
# WB for Load




Wrong register number!

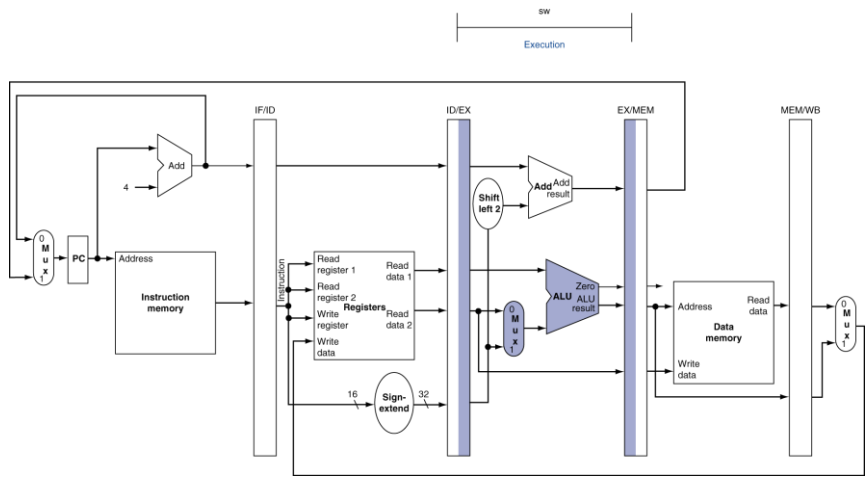
13

# Corrected Datapath for Load



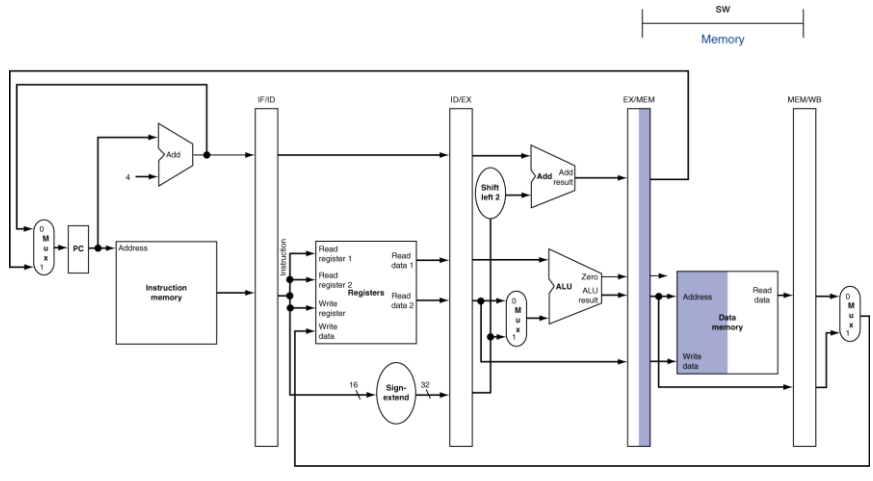

14

# EX for Store



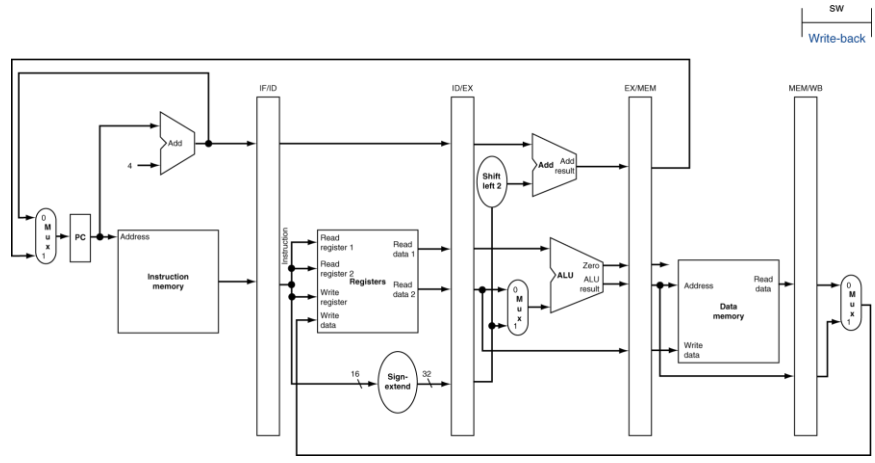

15

# MEM for Store




16

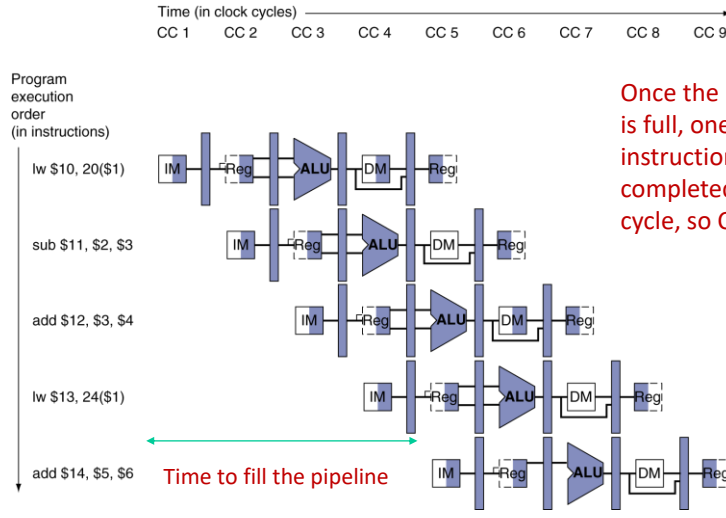
# WB for Store






# Multi-Cycle Pipeline Diagram

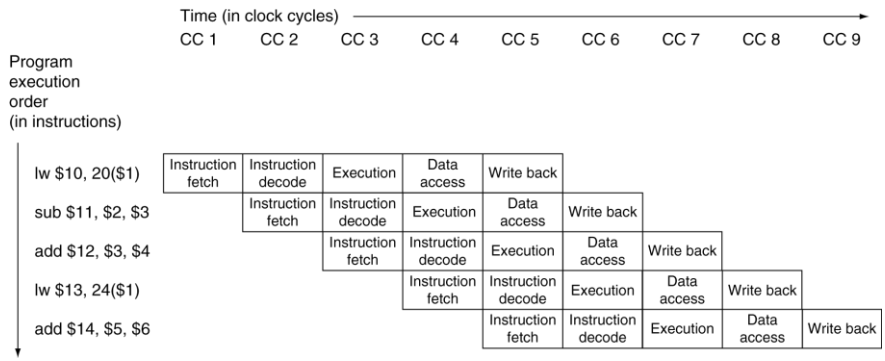
- Form showing resource usage




Once the pipeline is full, one instruction is completed every cycle, so CPI = 1

# Multi-Cycle Pipeline Diagram

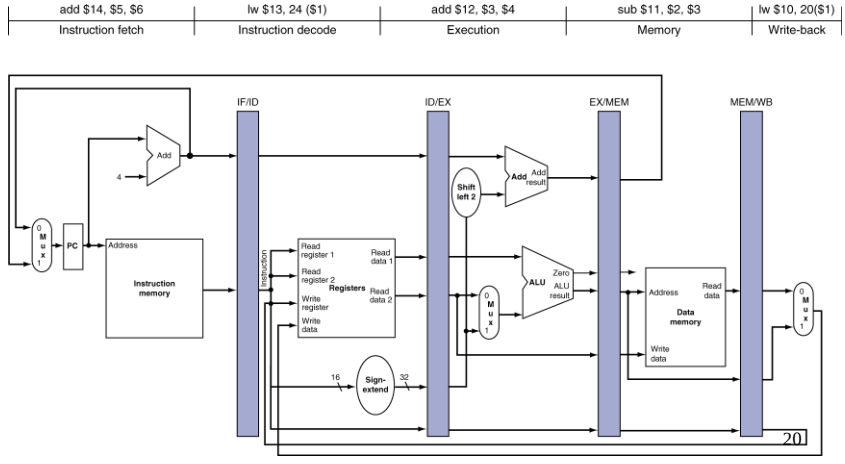
- Traditional form



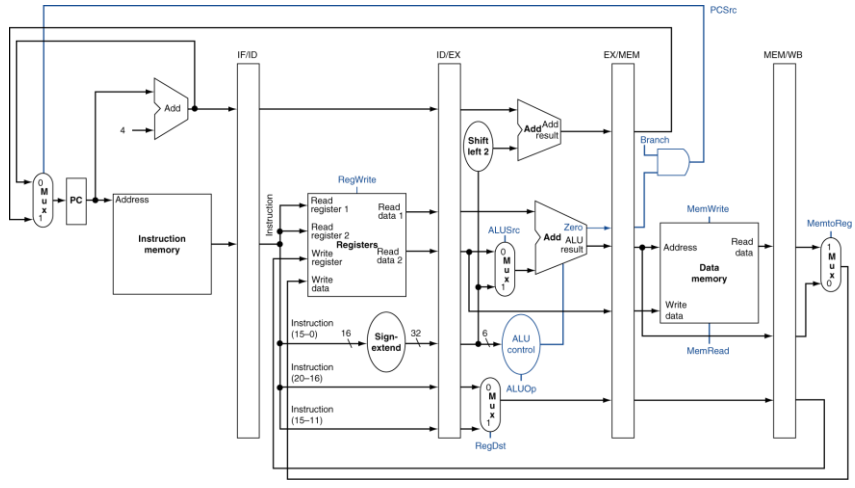



# Single-Cycle Pipeline Diagram

- State of pipeline in cycle CC5



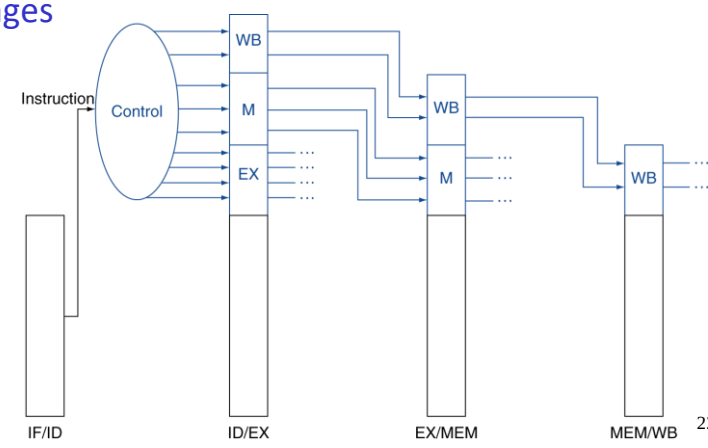
# Pipelined Control (Simplified)






# Pipelined Control

- Control signals derived from instruction in ID stage and held in pipeline registers between stages



# Pipelined Control

- IF stage: read instruction memory (always asserted) and write PC (on system clock edge)
- ID stage: no optional control signals to set

| | EX Stage | | | | MEM Stage | | | WB Stage | |
|---|---|---|---|---|---|---|---|---|---|
| | Reg Dst | ALU Op1 | ALU Op0 | ALU Src | Brch | Mem Read | Mem Write | Reg Write | Mem toReg |
| R | | | | | | | | | |
| `lw` | | | | | | | | | |
| `sw` | | | | | | | | | |
| `beq` | | | | | | | | | |

23

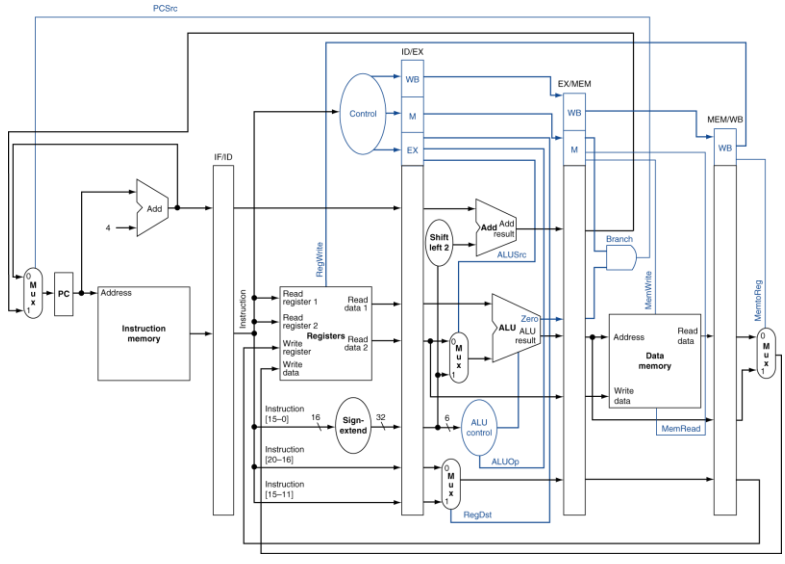
# Pipelined Control – con't

- IF stage: read instruction memory (always asserted) and write PC (on system clock edge)
- ID stage: no optional control signals to set

| | EX Stage | | | | MEM Stage | | | WB Stage | |
|---|---|---|---|---|---|---|---|---|---|
| | Reg Dst | ALU Op1 | ALU Op0 | ALU Src | Brch | Mem Read | Mem Write | Reg Write | Mem toReg |
| R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| `lw` | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| `sw` | X | 0 | 0 | 1 | 0 | 0 | 1 | 0 | X |
| `beq` | X | 0 | 1 | 0 | 1 | 0 | 0 | 0 | X |

24

# Pipelined Control




25

# Summary

- MIPS pipeline is consisted of five stages
  - Registers between stages to hold information produced in previous cycle
  - Data flows from left to right with exception of WB and MEM
- Two types of pipeline diagrams
  - Multi-cycle pipeline diagram shows resource usage
  - Single-cycle diagram shows state of pipeline in a given cycle
- Control signals derived from instruction in ID stage and held in pipeline registers between stages

26