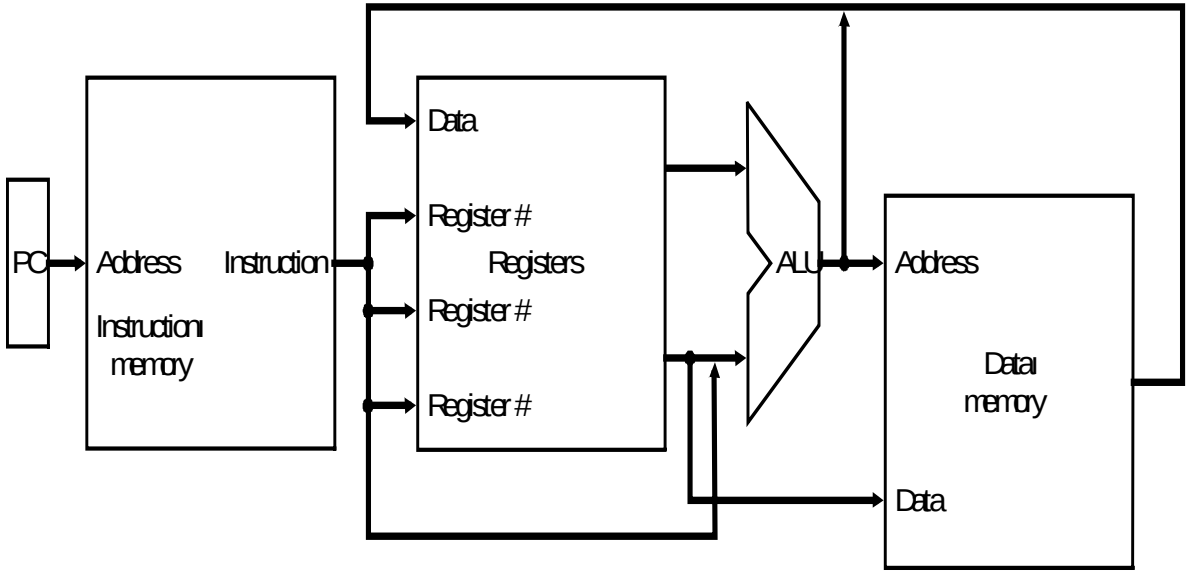
Datapath

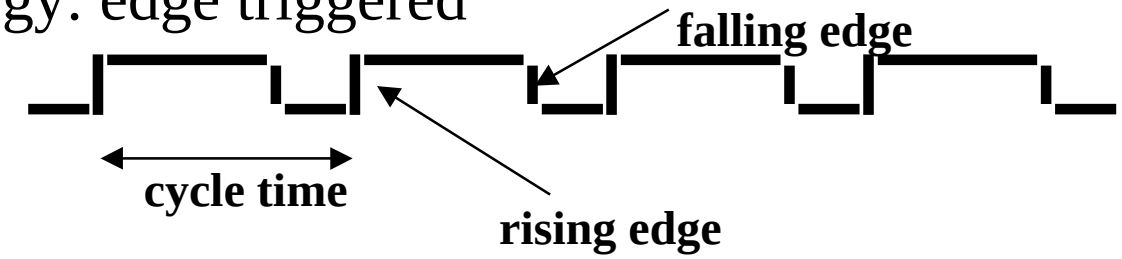
- Abstract / Simplified View:



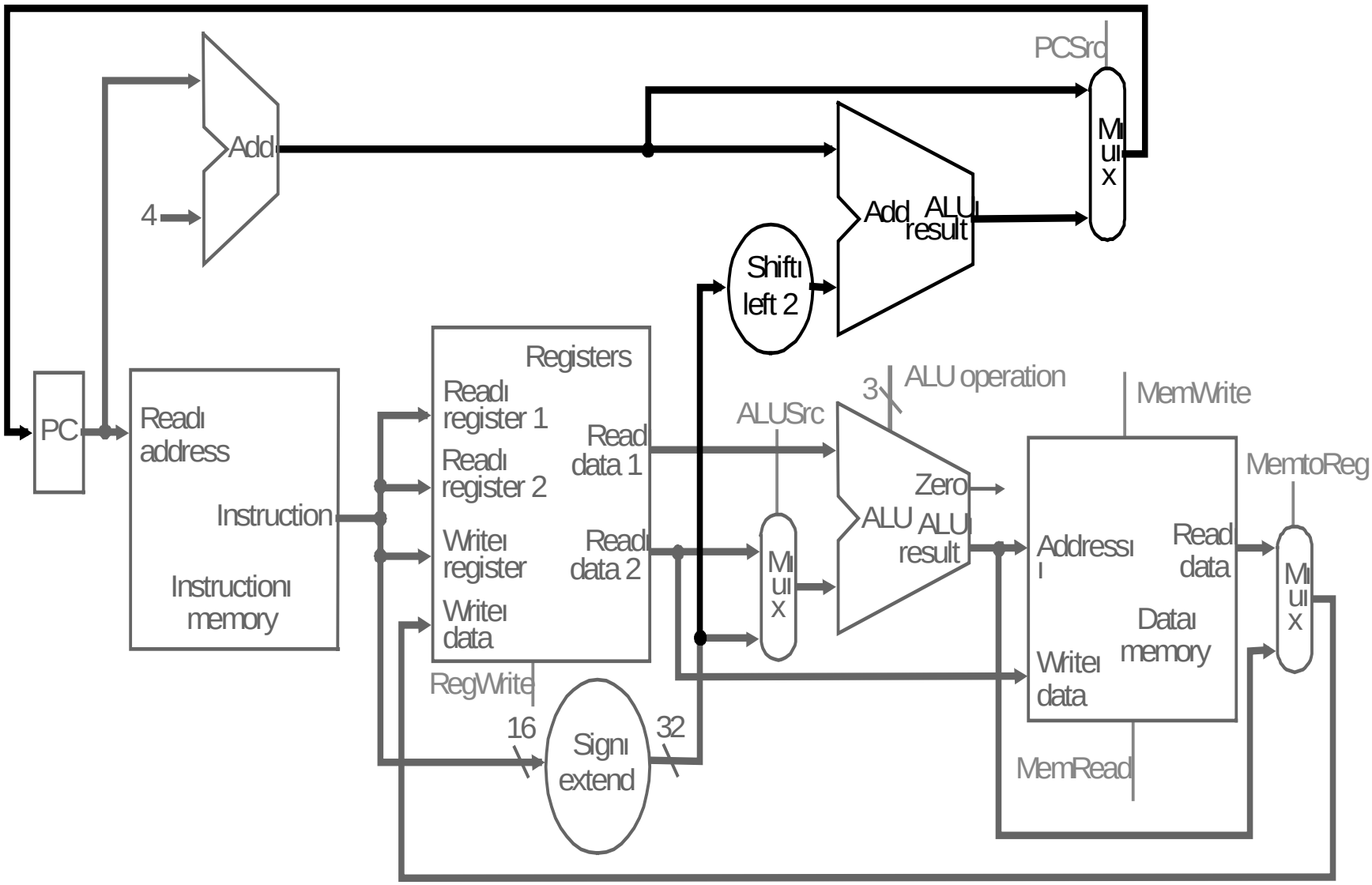
Two types of functional units:

- Combinational logic
- State elements: D-lathes and D flip-flops

- Clocking methodology: edge triggered



Building the Datapath



Control

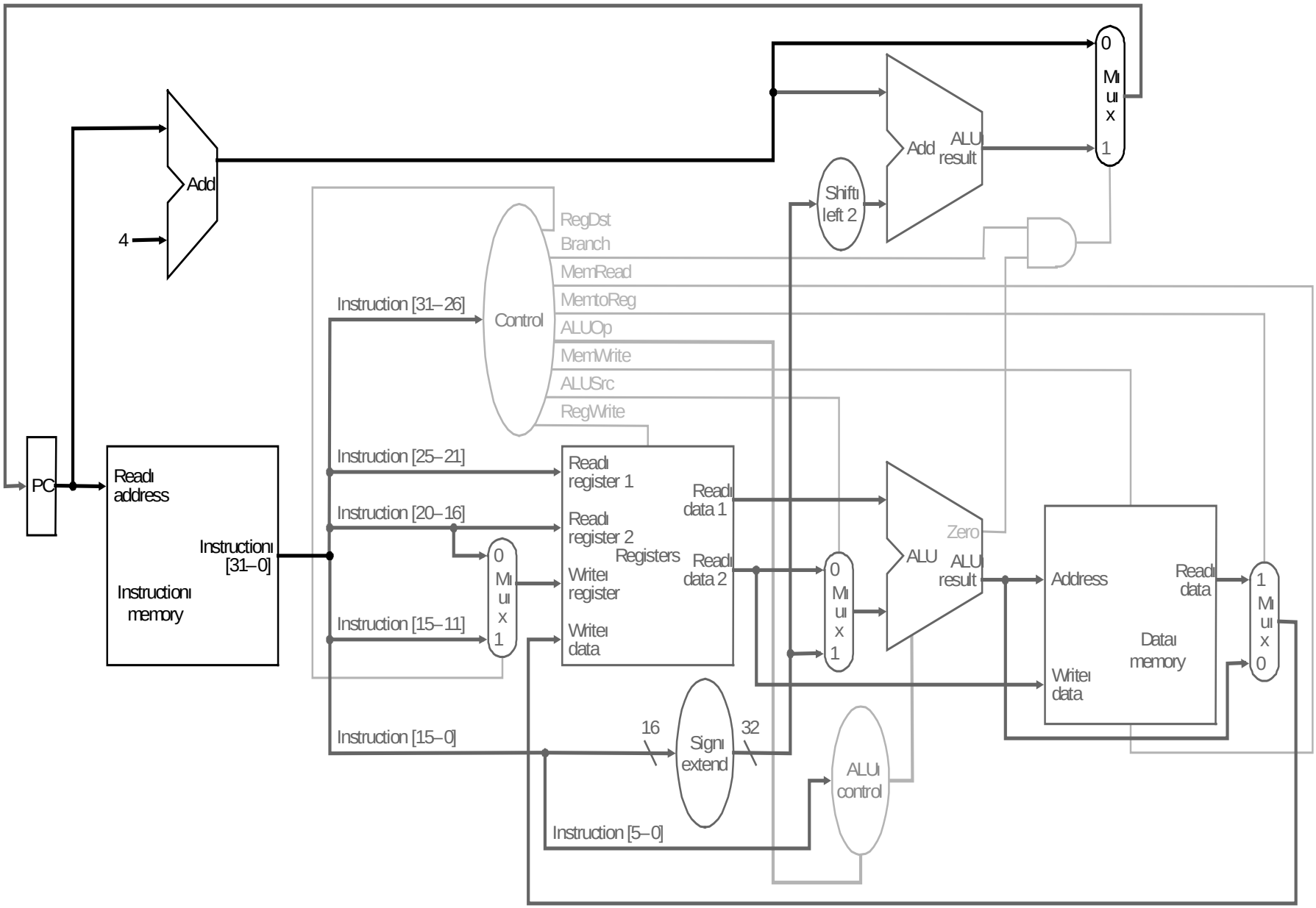
- Selecting the operations to perform (ALU, read/write, etc.)
- Controlling the flow of data (multiplexor inputs)
- Information comes from the 32 bits of the instruction
- Example:

add $8, $17, $18 Instruction Format:

000000	10001	10010	01000	00000	100000
op	rs	rt	rd	shamt	funct

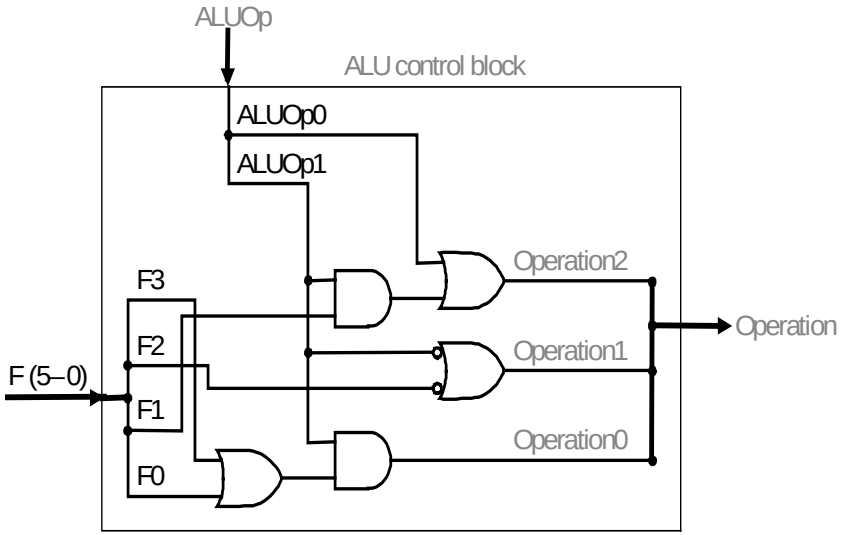
- ALU's operation based on instruction type and function code

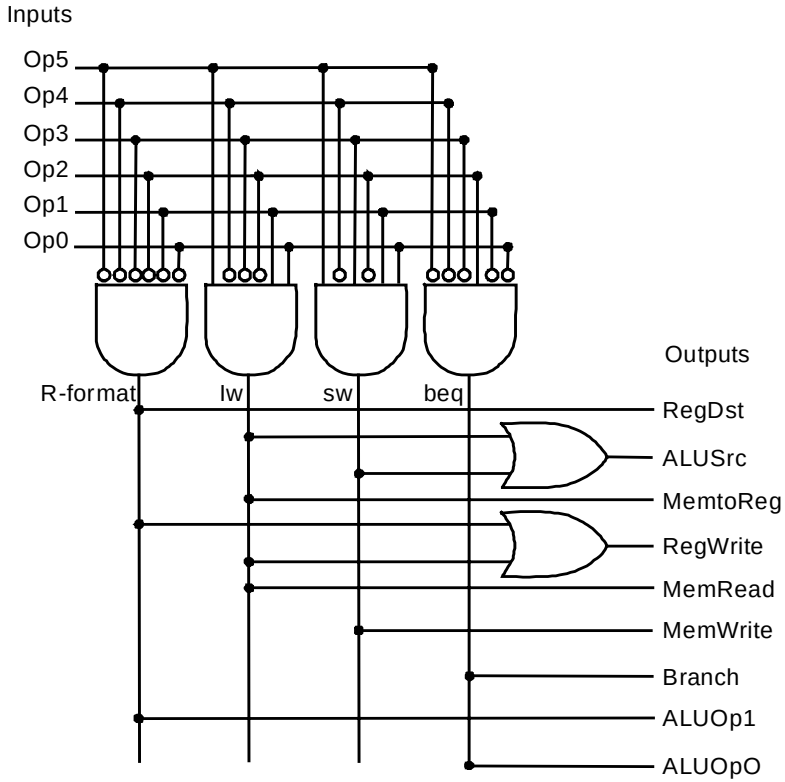
Control


Add
4
M
u
x
0
1
Add
ALU result
Shift left 2
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
16
Sign extend
32
ALU control
Zero
ALU
ALU result
Address
Read data
Data memory
Write data

Control

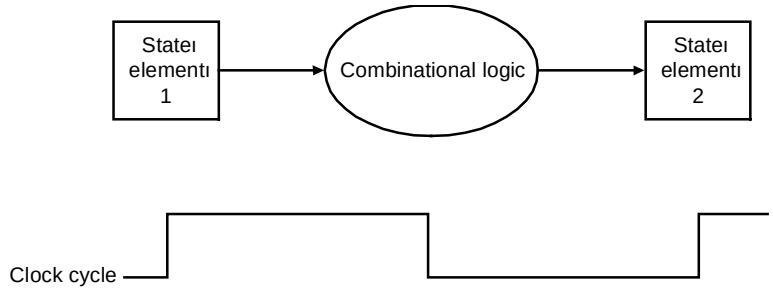
- Simple combinational logic (truth tables)





Our Simple Control Structure

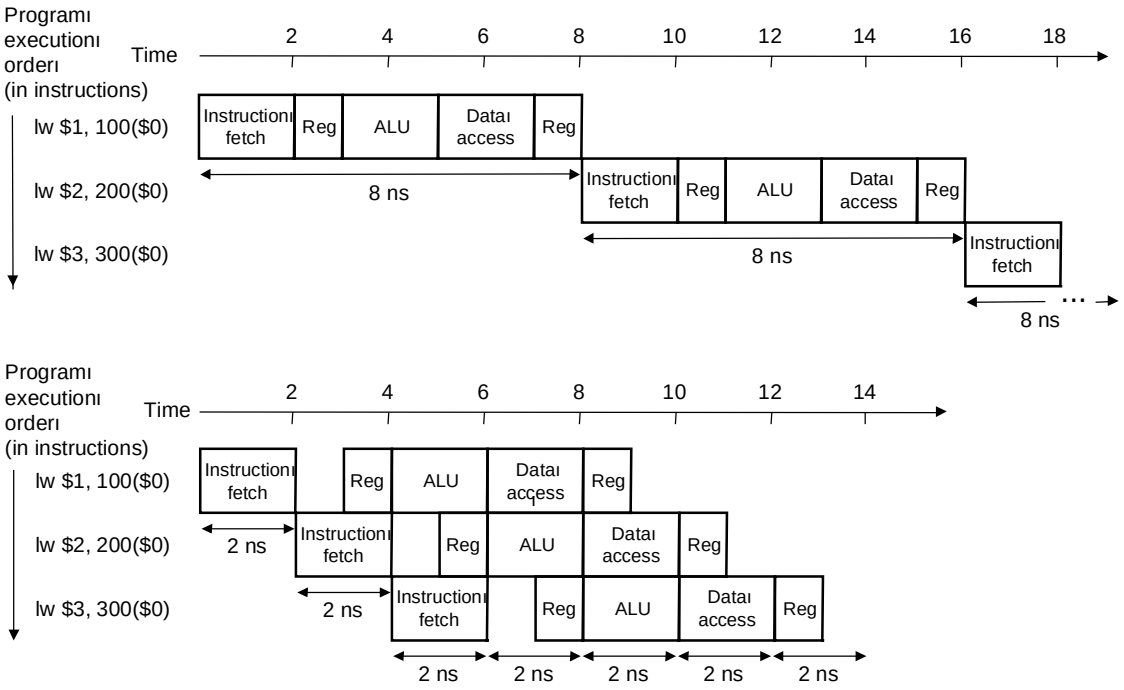
- All of the logic is combinational
- We wait for everything to settle down, and the right thing to be done
  - ALU might not produce “right answer” right away
  - we use write signals along with clock to determine when to write
- Cycle time determined by length of the longest path



We are ignoring some details like setup and hold times

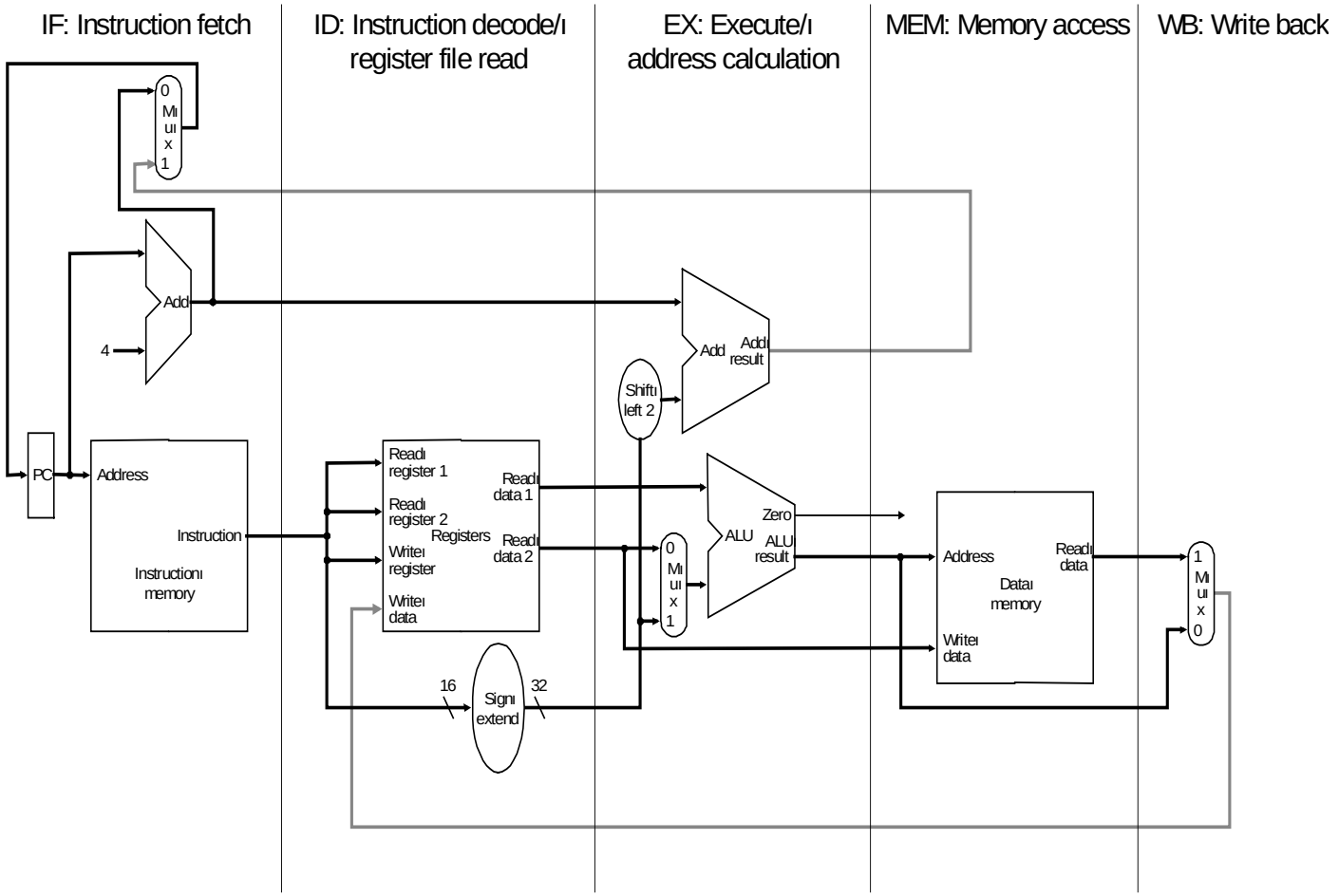
Pipelining

- Improve perfomance by increasing instruction throughput

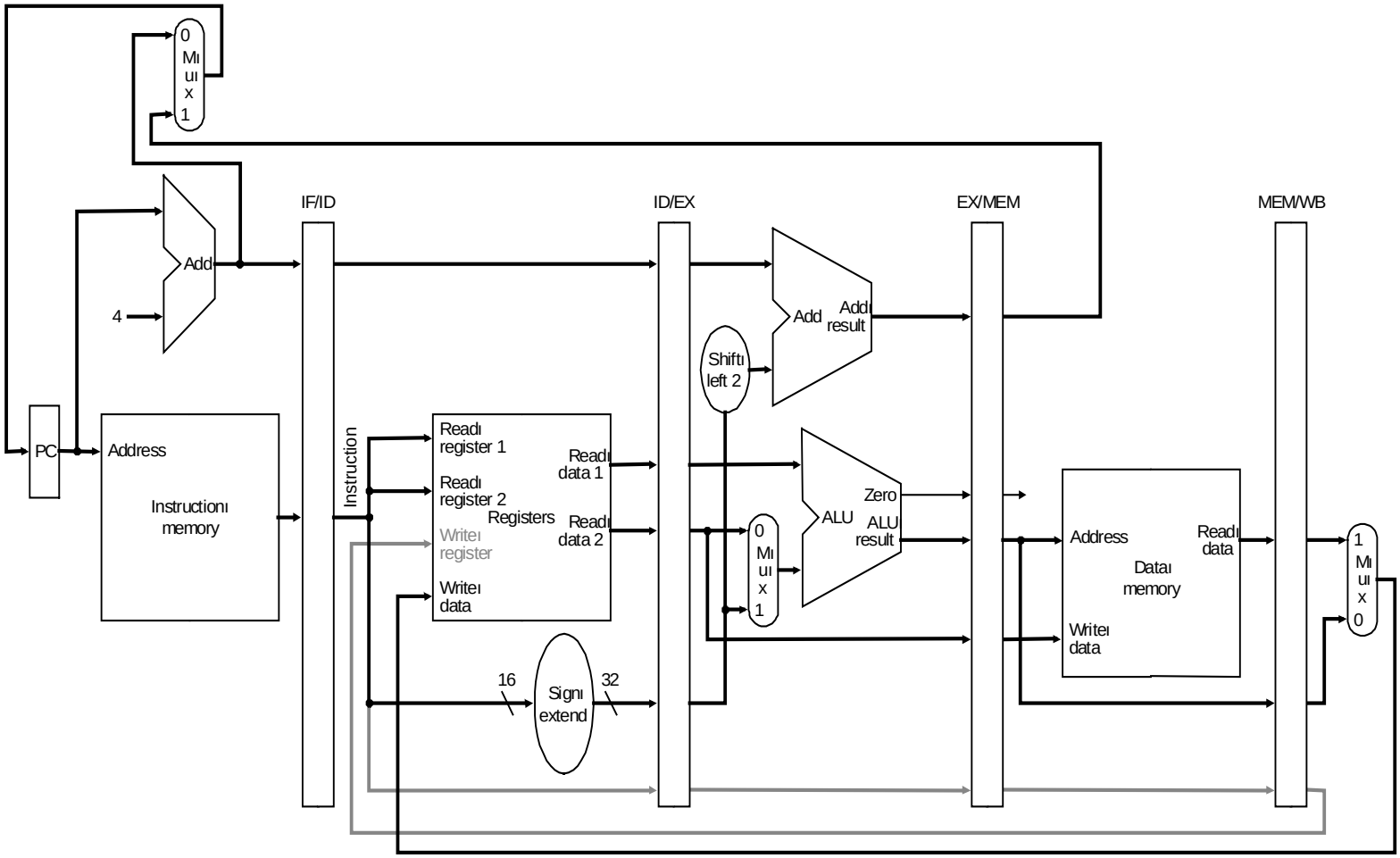


Ideal speedup is number of stages in the pipeline. Do we achieve this?

Implementation of the pipelined datapath

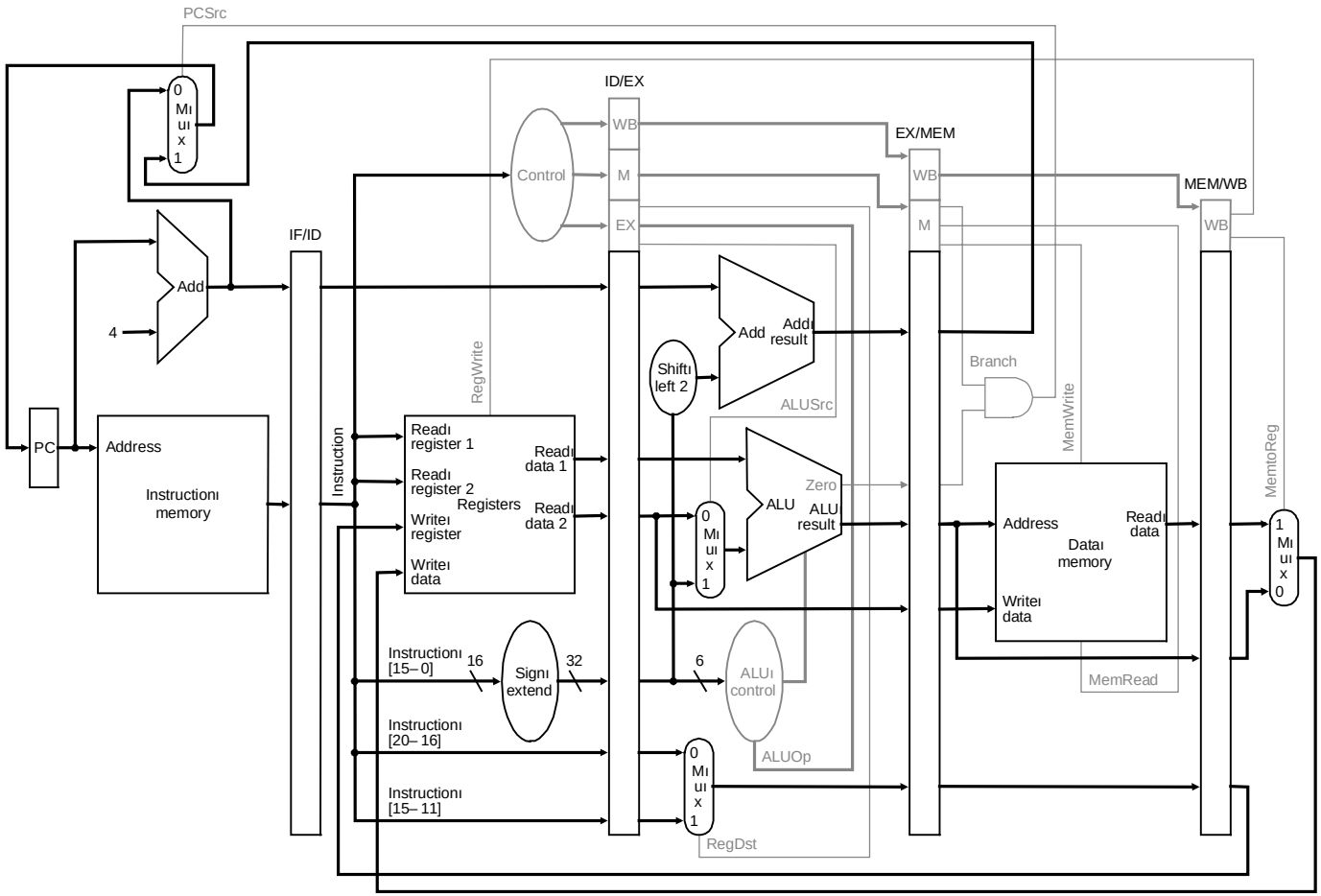


Datapath

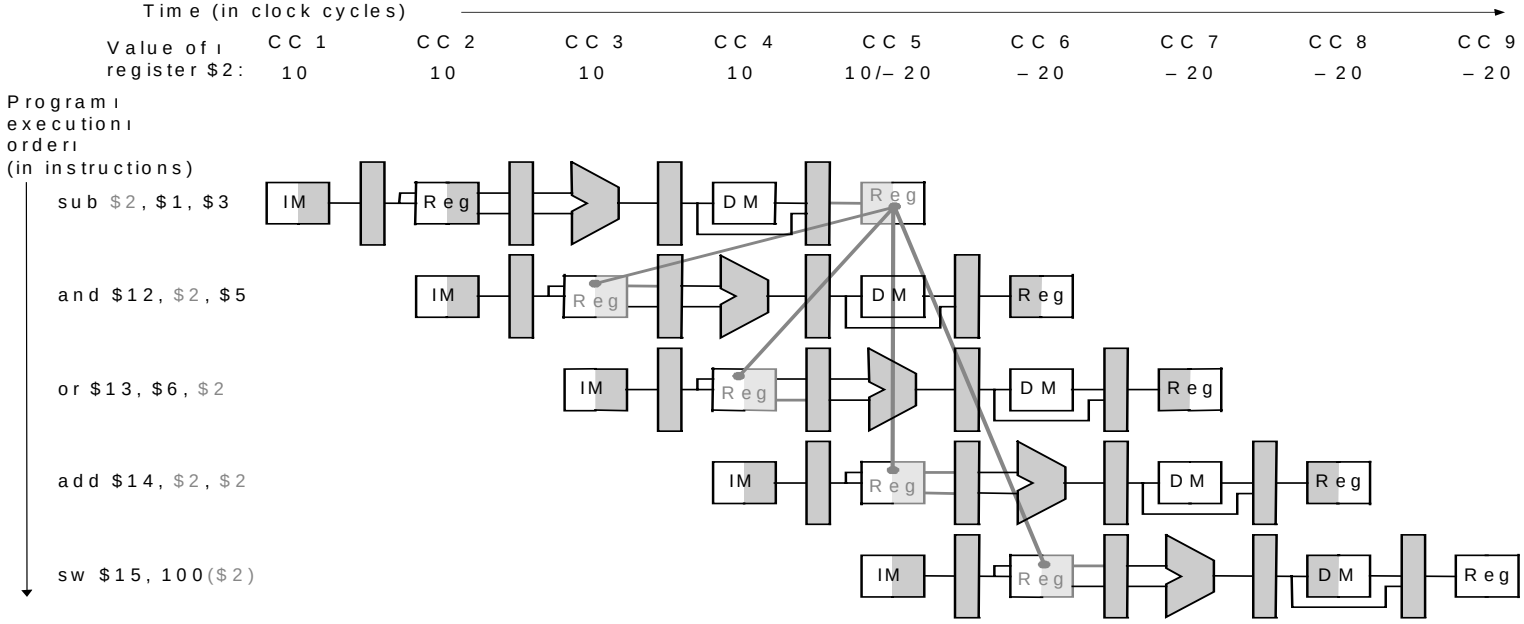

0
M
u
x
1
Add
4
IF/ID
ID/EX
EX/MEM
MEM/WB
Add
Add
result
Shift
left 2
PC
Address
Instruction
memory
Instruction
Read
register 1
Read
register 2
Registers
Write
register
Write
data
Read
data 1
Read
data 2
0
M
u
x
1
Zero
ALU
ALU
result
Address
Data
memory
Read
data
Write
data
1
M
u
x
0
16
Sign
extend
32

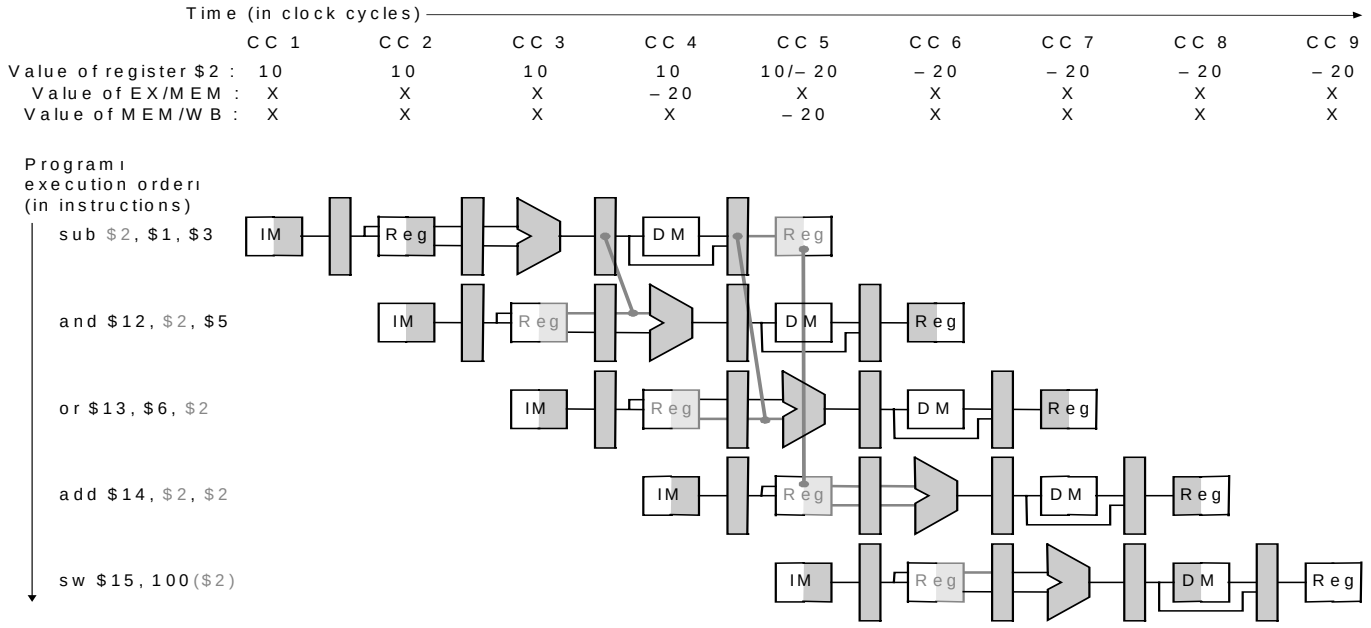
Datapath with Control

Pass control signals along just like the data



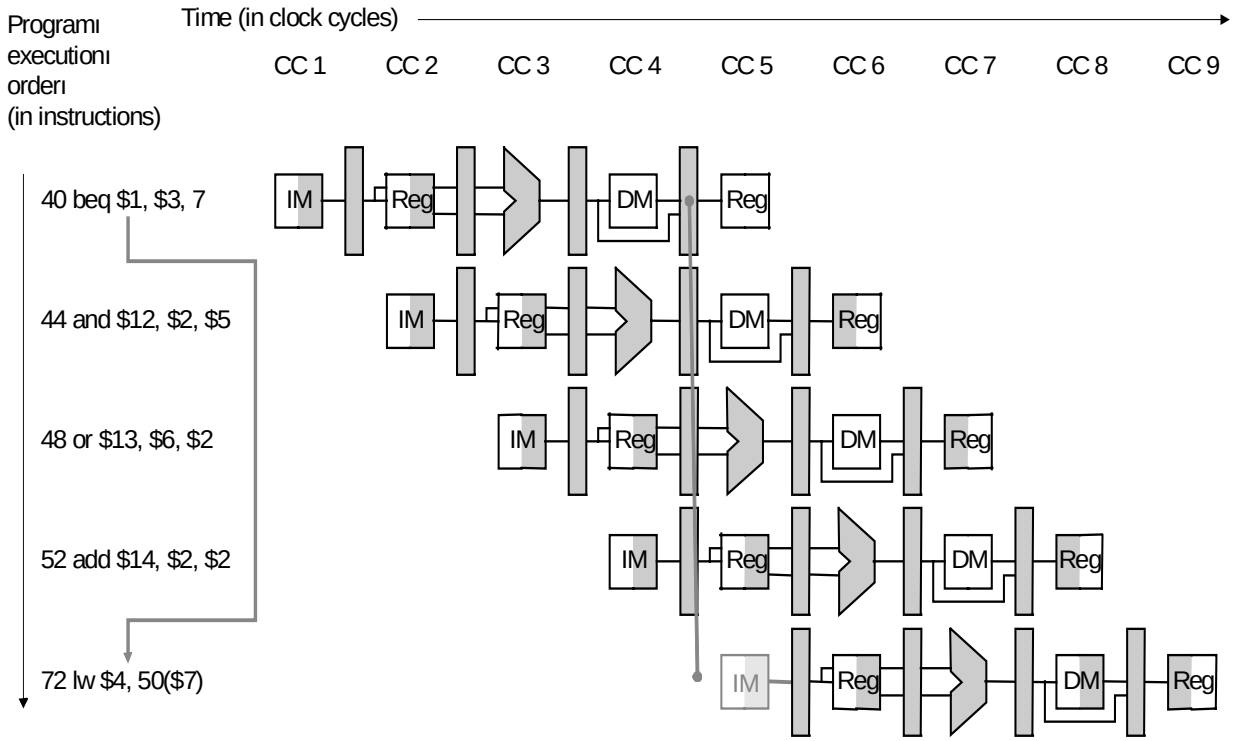
Dependencies and forwarding





Branch Hazards

- When we decide to branch, other instructions are in the pipeline!



Pipeline with hazard detection and forwarding