Single Cycle Processor
Pipelined

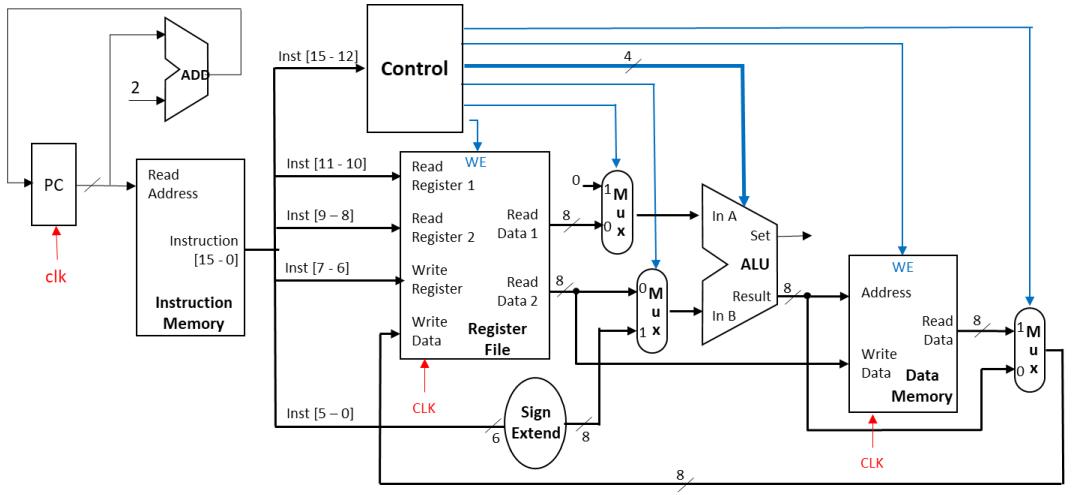
Last updated 2/22/24

Single Cycle Processor - Pipelined

- SCP Timing
  - Block Delays

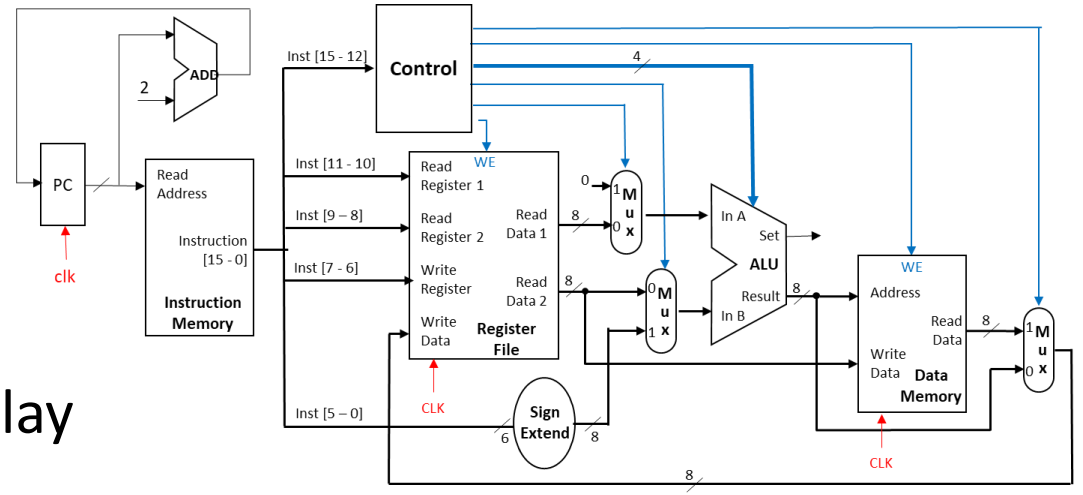
Fetch	20ns
Decode / register access	10ns
Execute (ALU)	10ns
Data Memory r/w	20ns
Writeback	5ns

  - Arithmetic operation
    - F+D+E+WB = 45ns
  - Load/Store
    - Load: F+D+E+M+WB = 65ns, Store: F+D+E+M = 60ns
  - Load Immediate
    - F+E+WB = 35ns



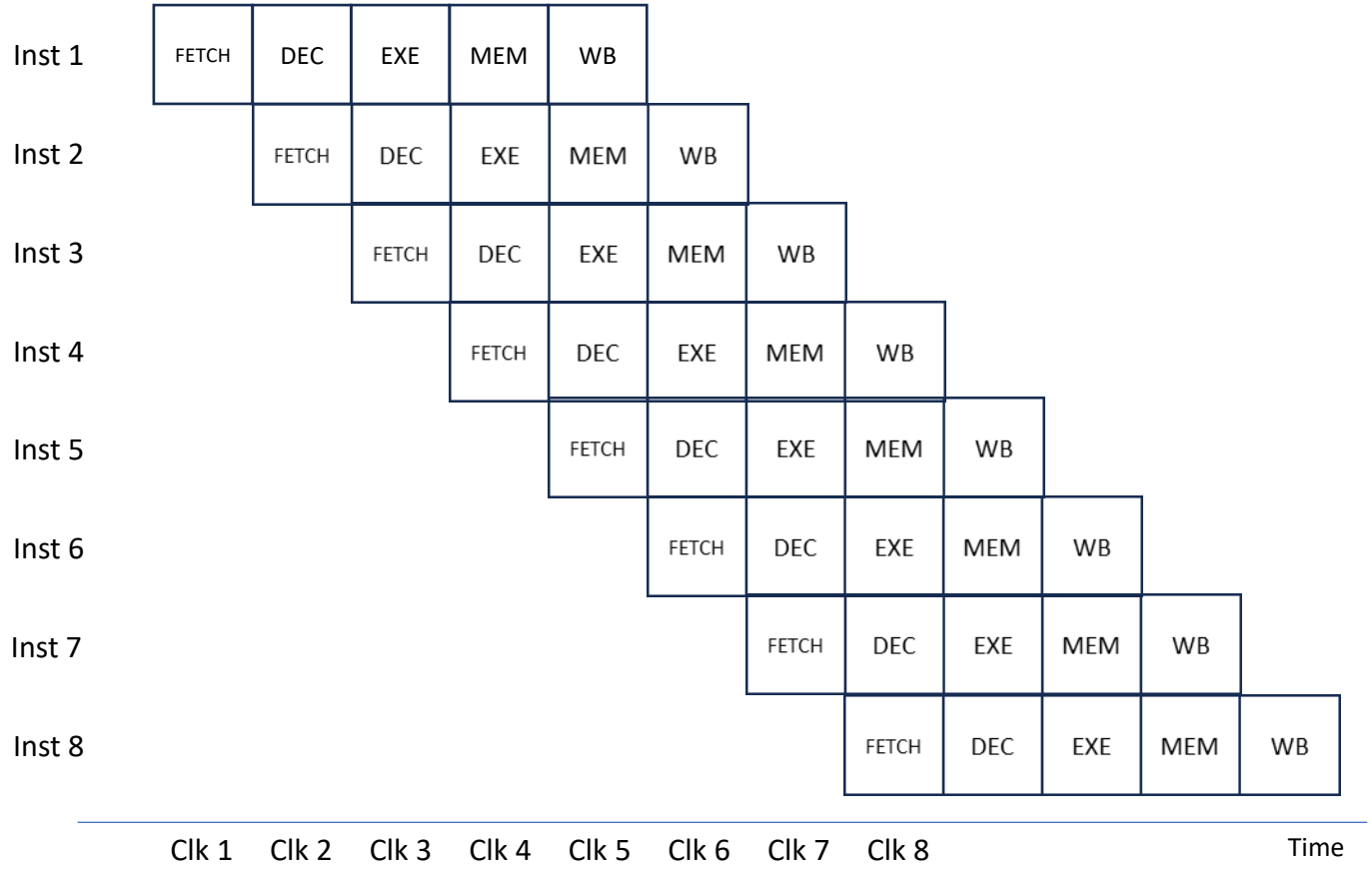
Single Cycle Processor - Pipelined

- SCP Timing
  - Only 1 allowed clock frequency
    - Can’t change frequency every instruction
  - Different operations have different delays
  - Must set clock to account for worst case delay
    - Load instruction = 65nS
    - Max frequency = 15.38MHz



Single Cycle Processor - Pipelined

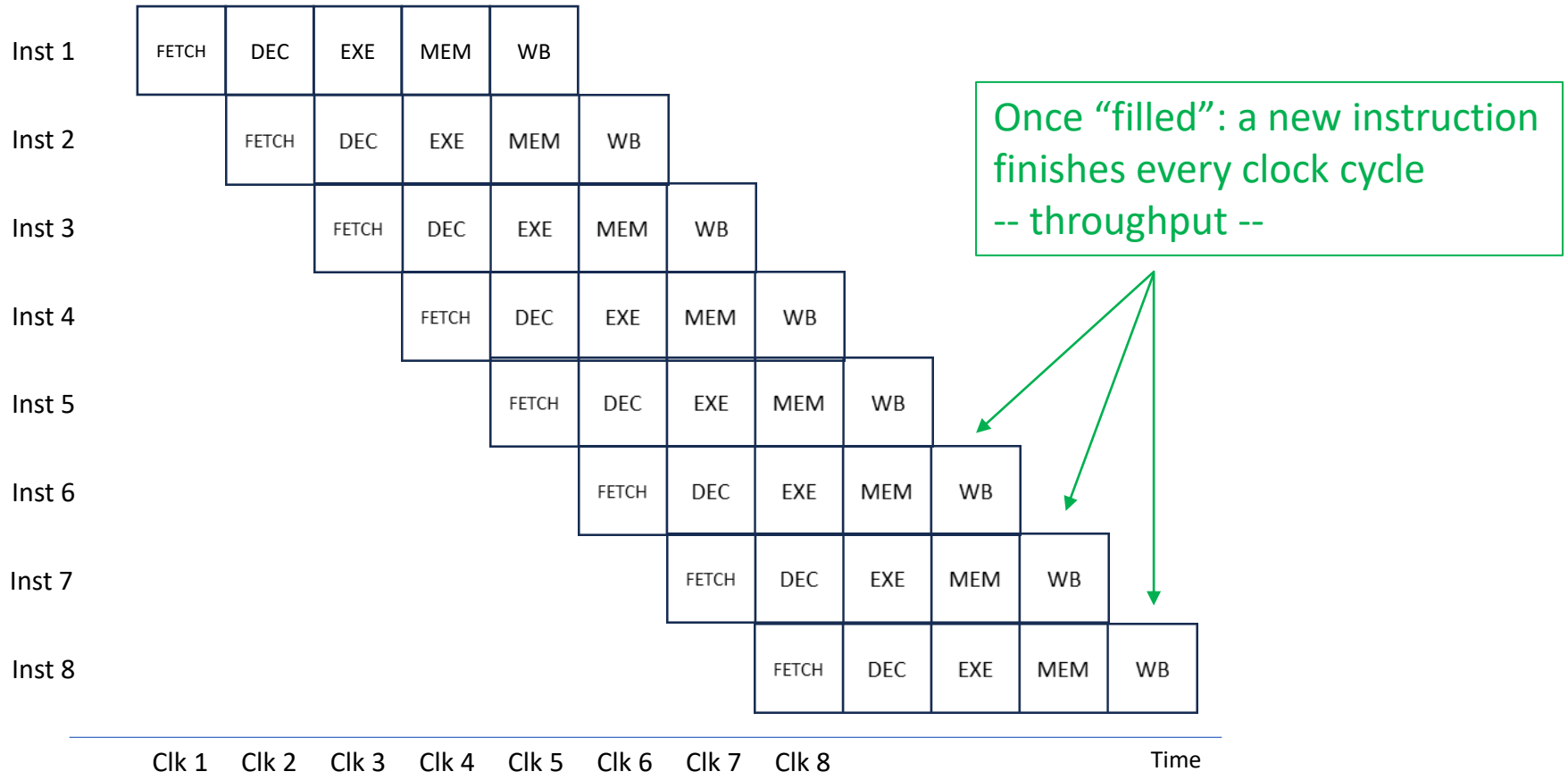
- Pipeline Solution
  - Create a pipeline stage for each step



Single Cycle Processor - Pipelined

- Pipeline Solution
  - Create a pipeline stage for each step

Each instruction takes
5 clock cycles to complete
-- latency --



Once "filled": a new instruction finishes every clock cycle
-- throughput --

Single Cycle Processor - Pipelined

- Pipeline Solution
  - Clock must be set for worst case Stage delay
    - Fetch or Mem = 20ns

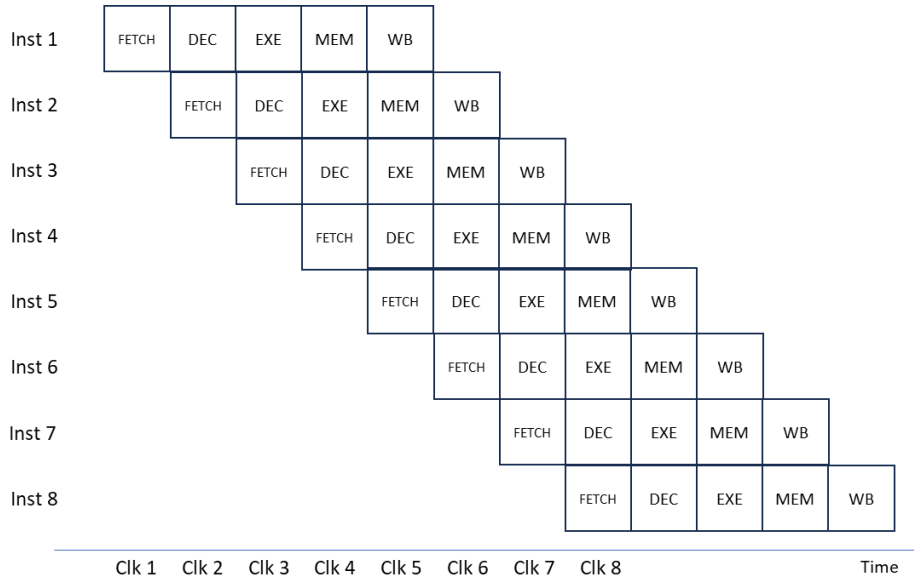
    → Latency = 100ns

    → Throughput = 1inst/20ns = 50M inst/s

    → Fclk = 50MHz

  - Latency only matters at the beginning
  - Throughput matters all the time
    - Single cycle = 1inst/65ns = 15.38M inst/s
    - Pipeline = 1inst/20ns = 50M inst/s

    → 225% speedup



Single Cycle Processor - Pipelined

- Pipeline Solution w/ penalty
  - Pipelining may increase delays
    - Assume 25% delay increase
    - Clock must be set for worst case Stage delay
    - Fetch or Mem = 25ns

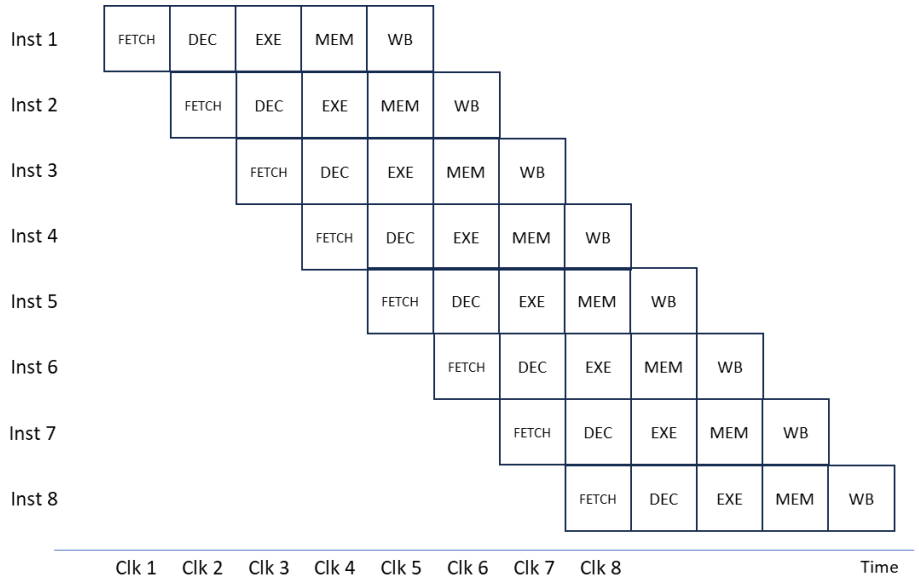
    → Latency = 125ns

    → Throughput = 1inst/25ns = 40M inst/s

    → Fclk = 40MHz

  - Latency only matters at the beginning
  - Throughput matters all the time
    - Single cycle = 1inst/65ns = 15.38M inst/s
    - Pipeline = 1inst/25ns = 40M inst/s

    → 160% speedup

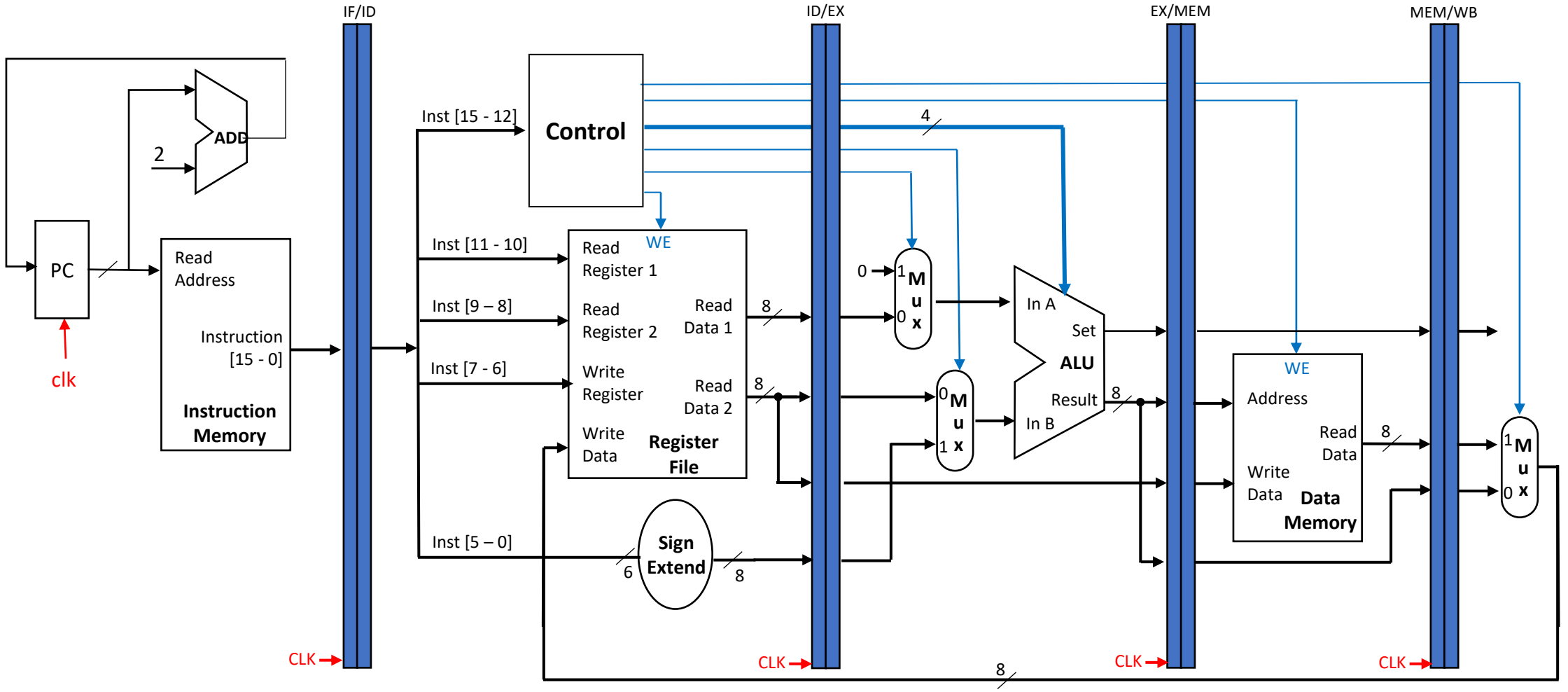


Single Cycle Processor - Pipelined

- Configuration
  - Insert registers between each execution stage
    - Fetch, Decode, Execute, Memory Access, Writeback

Single Cycle Processor - Pipelined

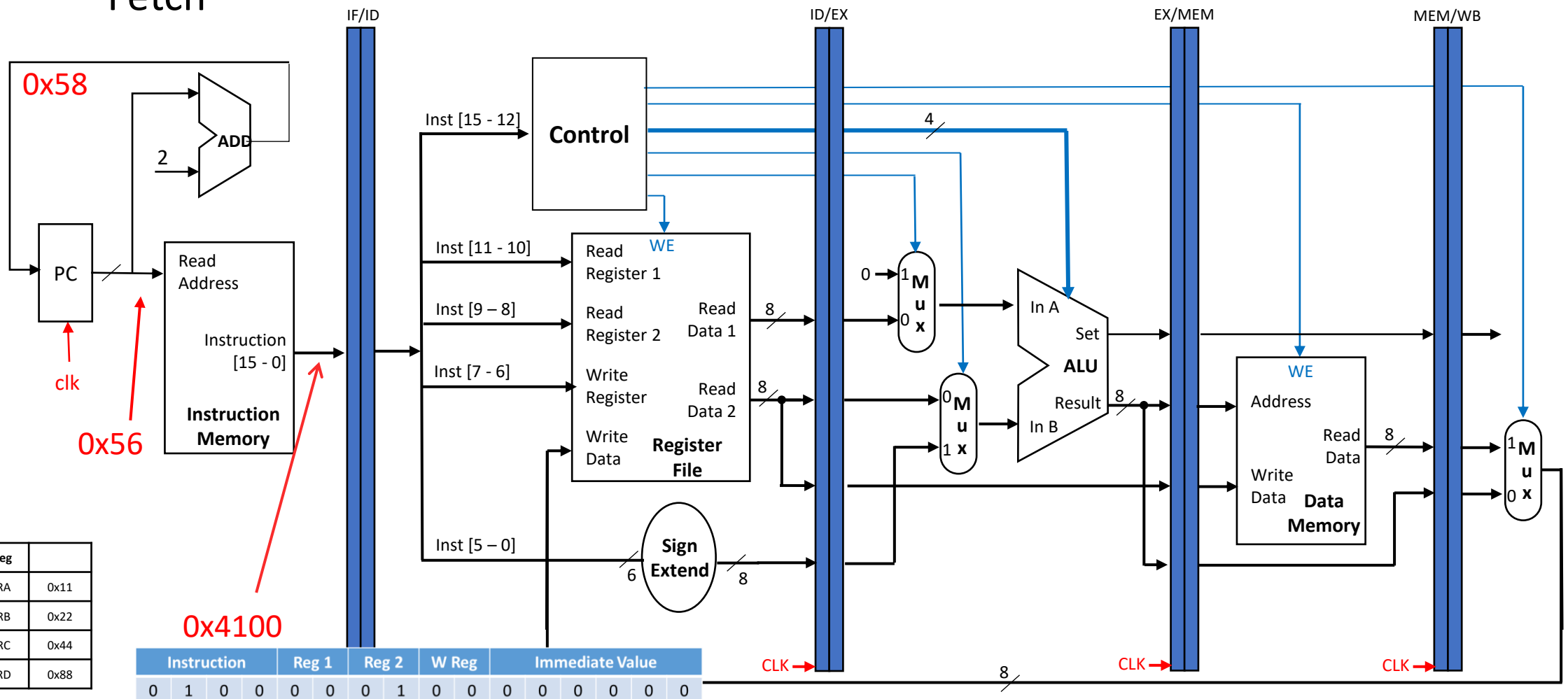
- Configuration



Single Cycle Processor - Pipelined

- Follow 1 instruction
  - Fetch

Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
Execute: add RA,RB,RA, located in program memory at 0x56



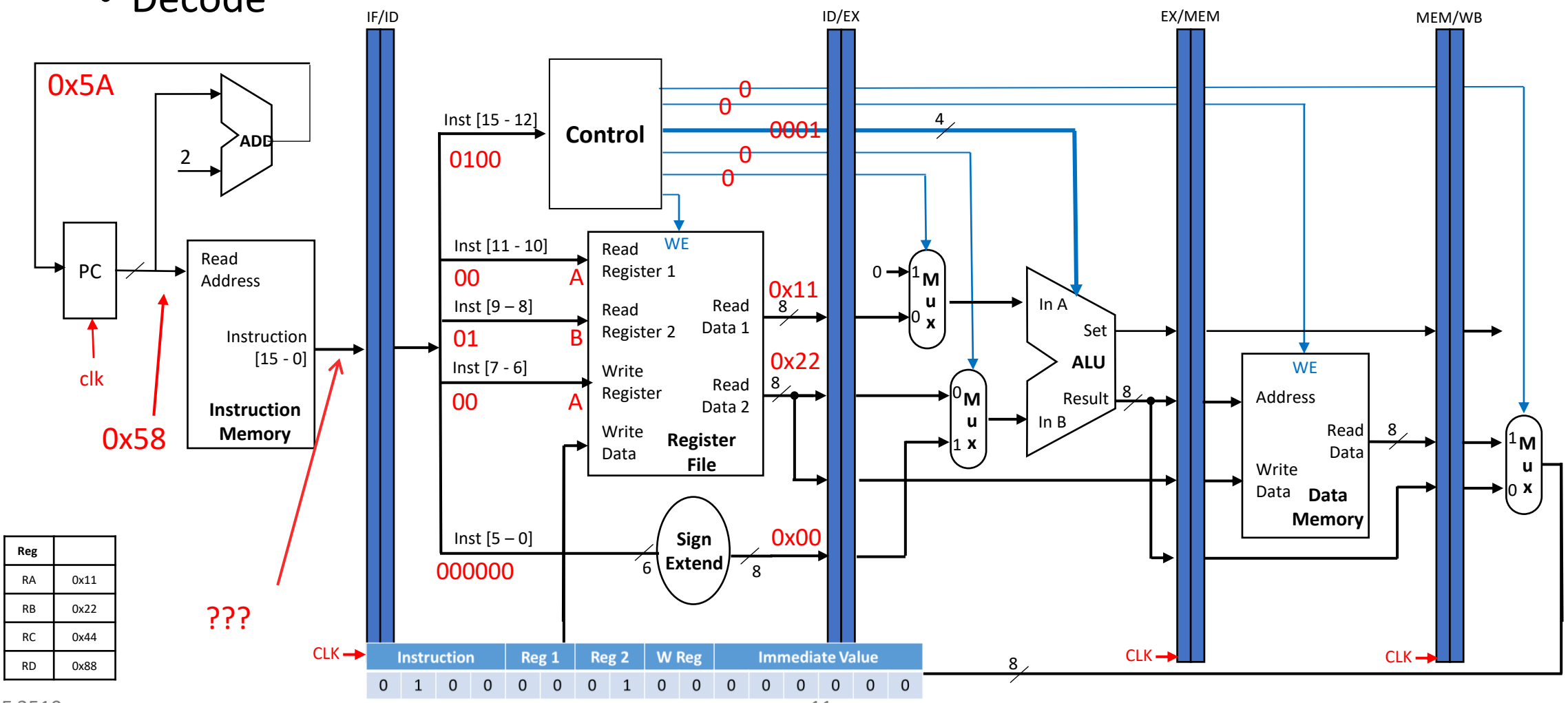
Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Single Cycle Processor - Pipelined

- Follow 1 instruction
  - Decode

Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
Execute: add RA,RB,RA, located in program memory at 0x56

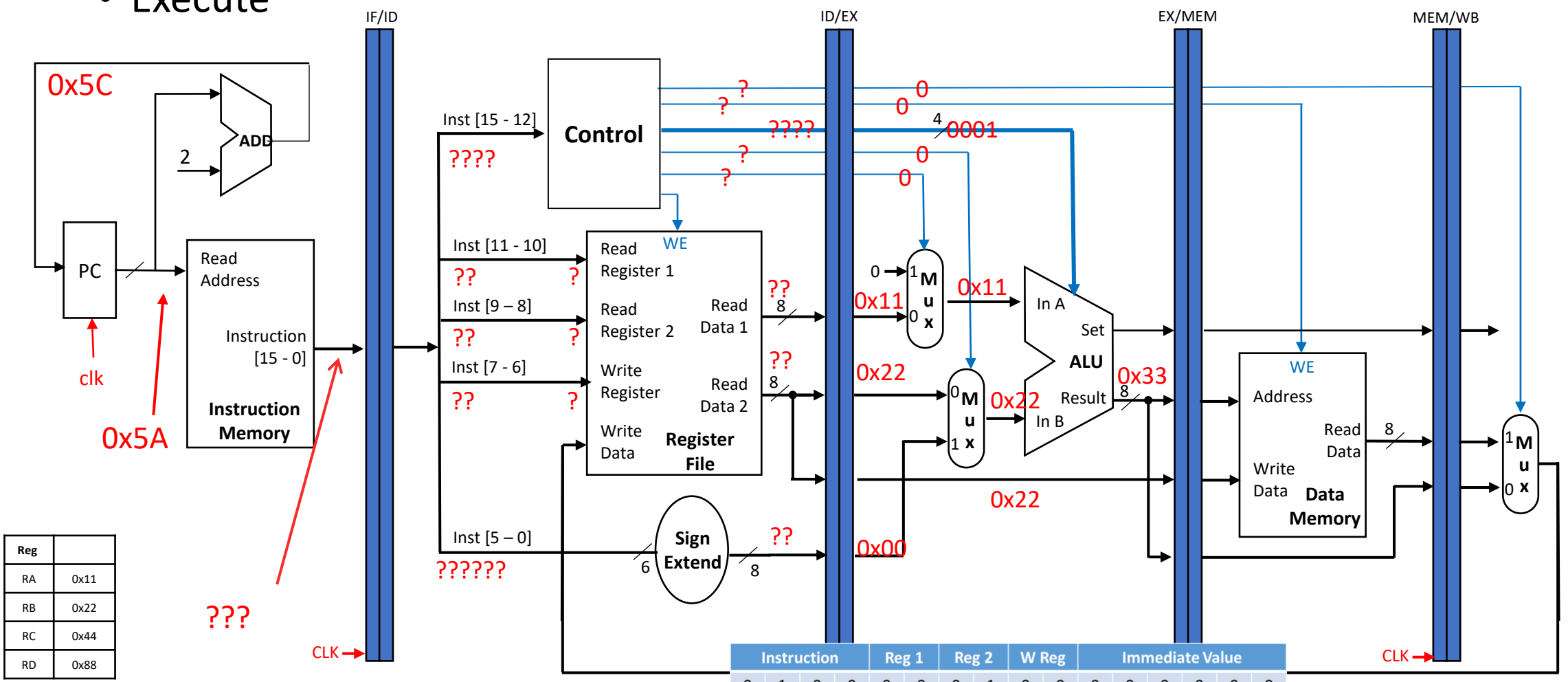


Reg	
RA	0x11
RB	0x22
RC	0x44
RD	0x88

Single Cycle Processor - Pipelined

- Follow 1 instruction
  - Execute

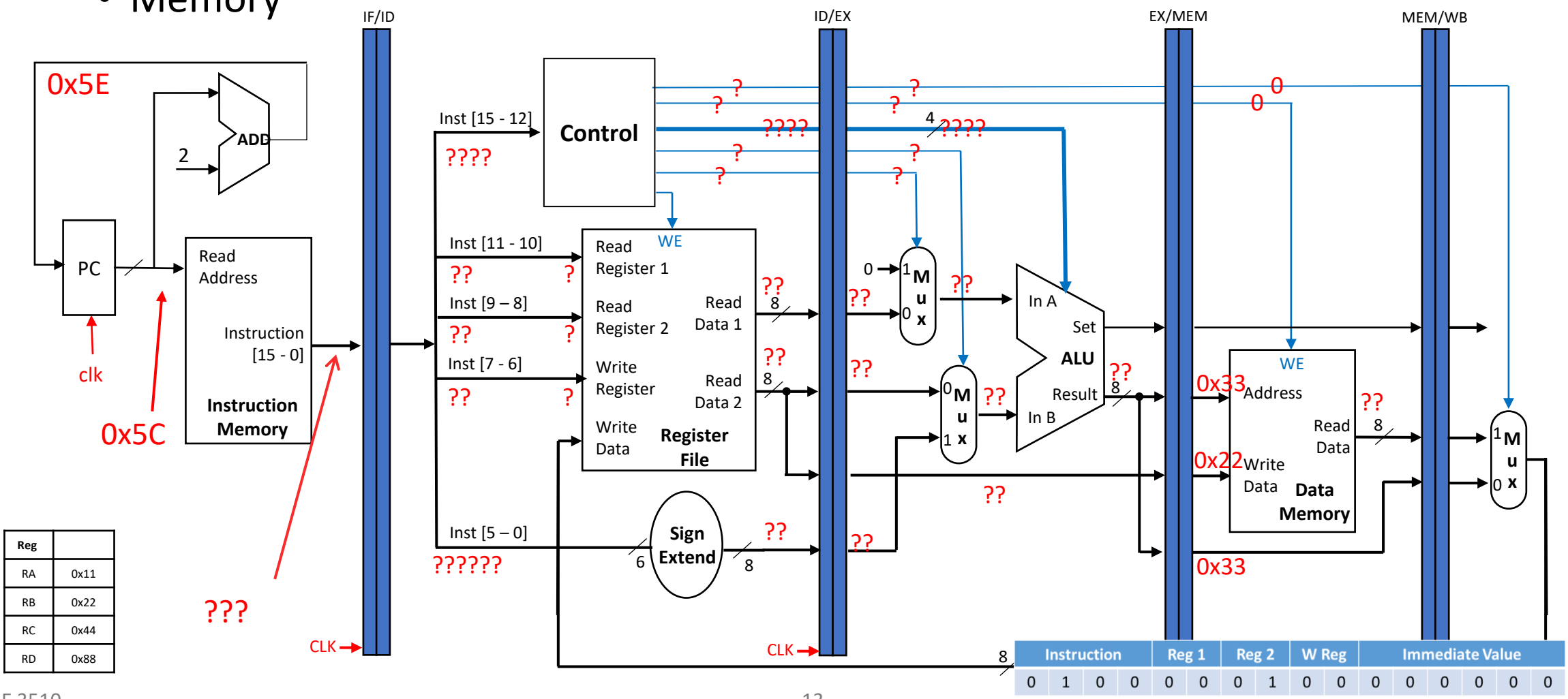
Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
Execute: add RA,RB,RA, located in program memory at 0x56



Single Cycle Processor - Pipelined

- Follow 1 instruction
  - Memory

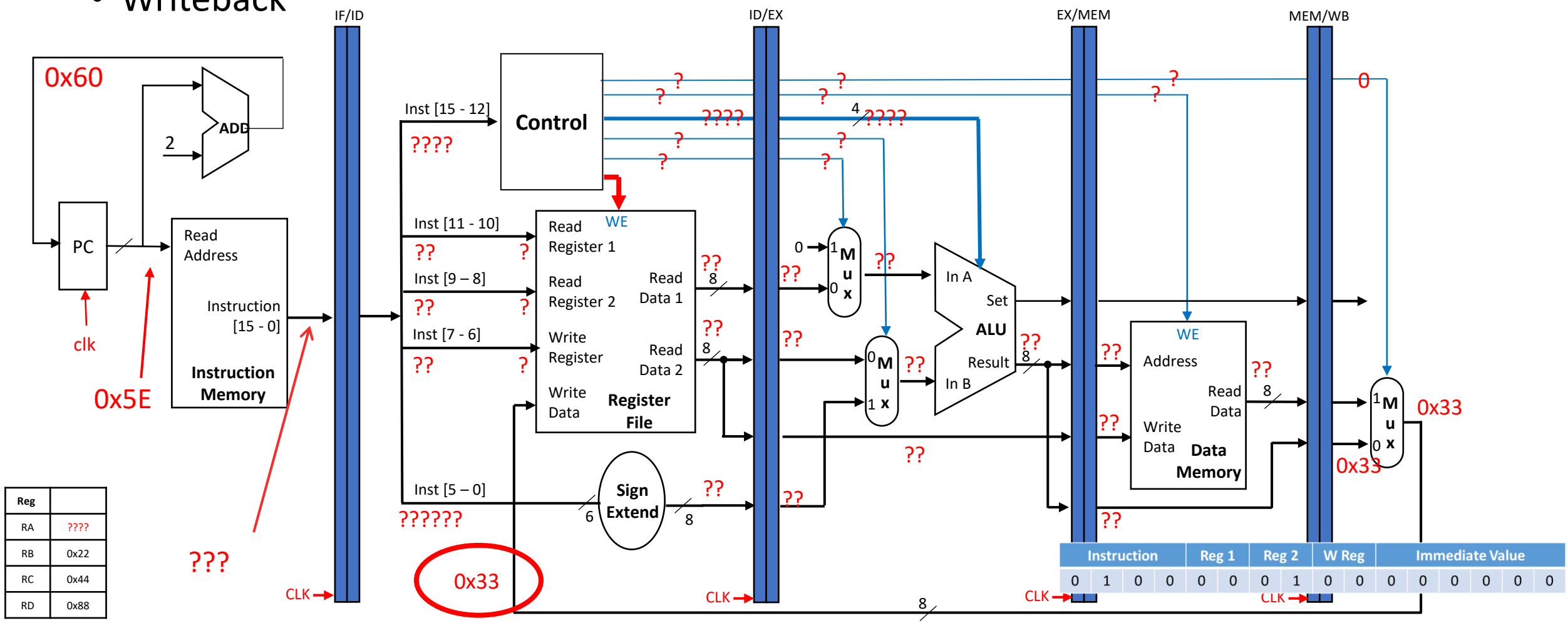
Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
Execute: add RA,RB,RA, located in program memory at 0x56



Single Cycle Processor - Pipelined

- Follow 1 instruction
  - Writeback

Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively
Execute: add RA,RB,RA, located in program memory at 0x56



Single Cycle Processor - Pipelined

Assume RA, RB, RC, RD contain the values 0x11, 0x22, 0x44, 0x88 respectively

Execute: add RA,RB,RA, located in program memory at 0x56

- Follow 1 instruction
  - Corrected Writeback