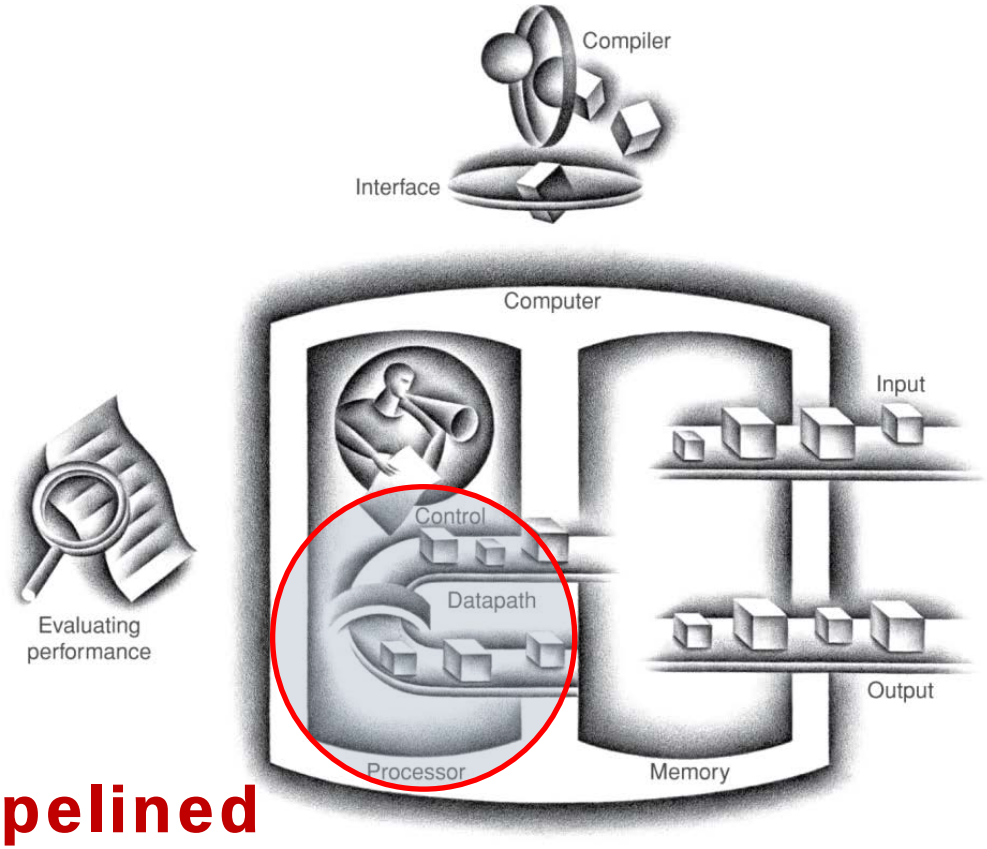
EE2011 Computer Organization

Lecture 10: Enhancing Performance with Pipelining ~ Pipelined Datapath

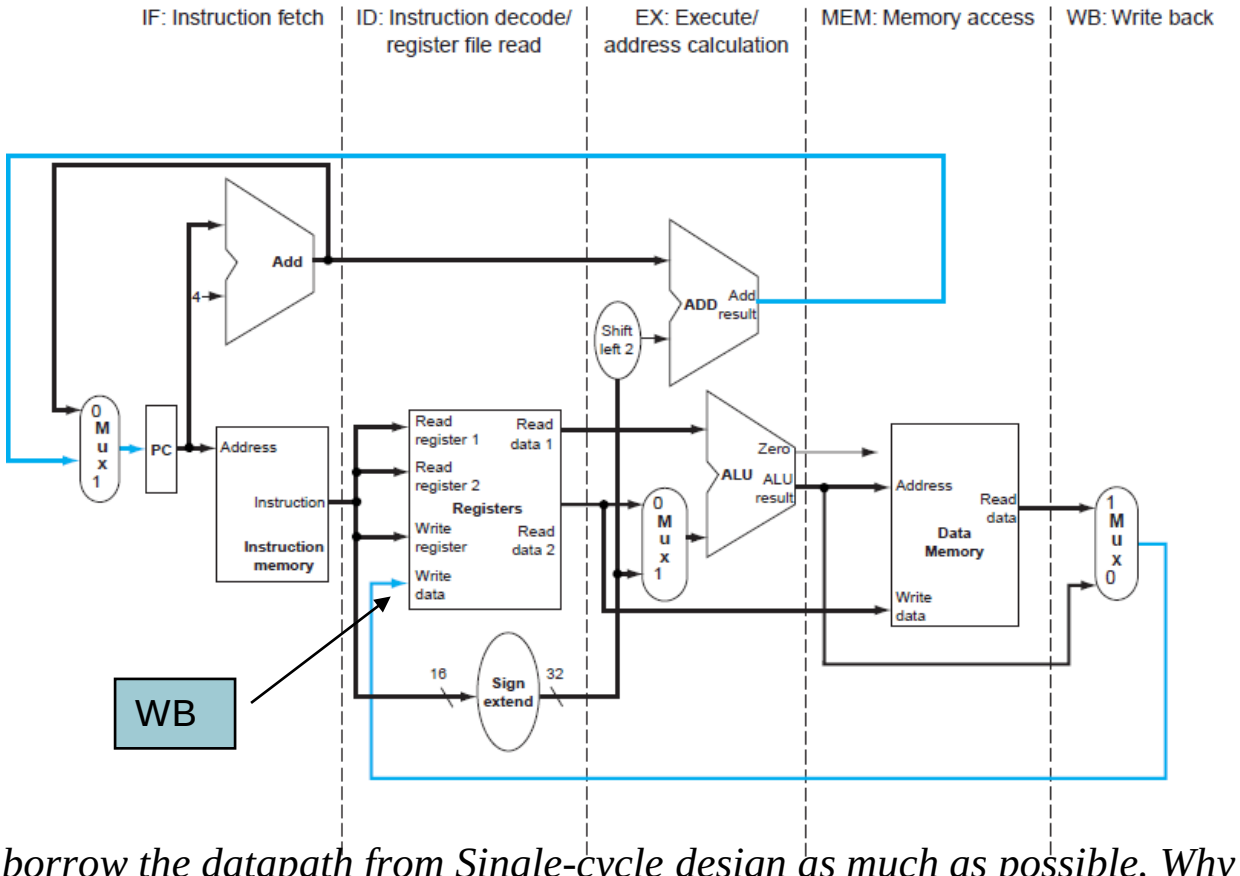
Wen-Yen Lin, Ph.D.
Department of Electrical Engineering
Chang Gung University
Email: wylin@mail.cgu.edu.tw
May 2022

Pipelined Datapath (Ch. 4.7)



Pipelined Datapath

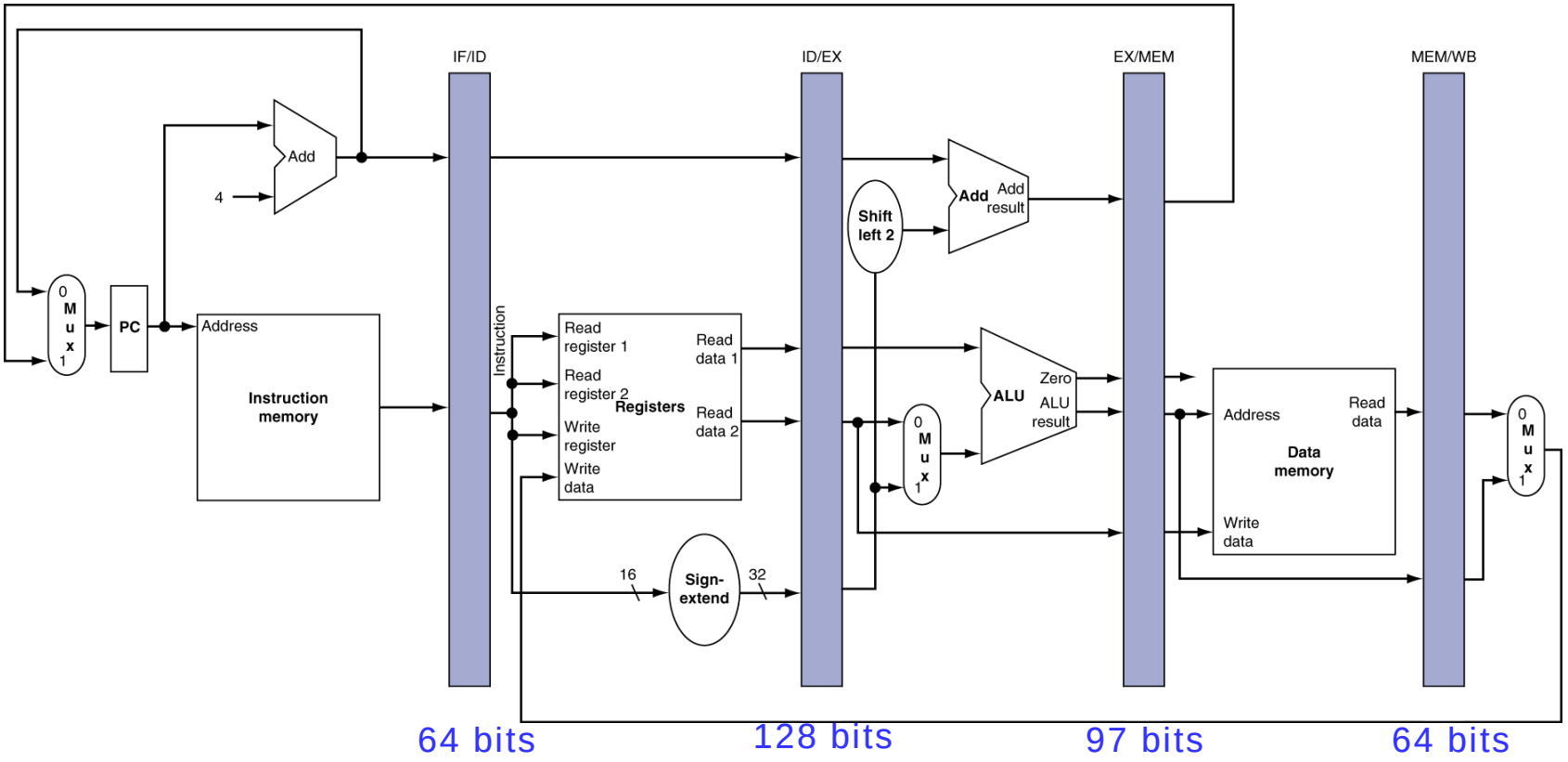
Basic Idea for Pipelined Datapath ~ from Single-cycle Datapath (Fig. 4.33)



- *Let's borrow the datapath from Single-cycle design as much as possible. Why?*
- *What do we need to add to actually split the datapath into stages?*

=> Add internal registers, i.e. Pipeline Registers, to hold internal data between each stage in the pipeline.

Pipelined Datapath (Fig. 4.35)



- Each pipeline register has to be large enough to hold all the data produced in previous cycle and being passing to the next stage.
  Can you find a problem even if there are no dependencies?
  What instructions can we execute to manifest the problem?

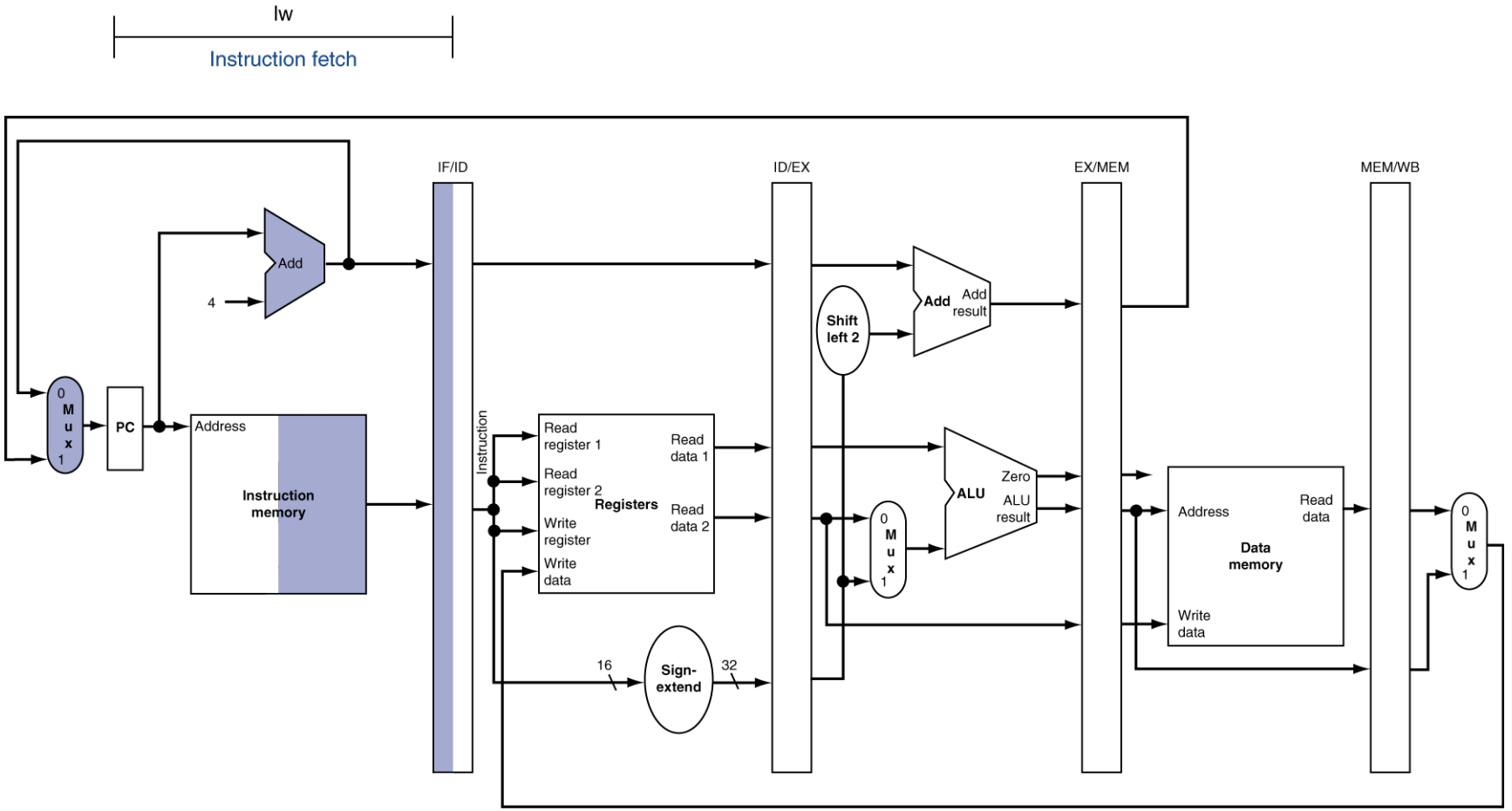
Details of Pipeline Registers

- IF/ID (64 bits)
  - IF/ID.IR : register for instruction code (32 bits)
  - IF/ID.PC : register for Incremented Program Counter (32 bits)
- ID/EX (128 bits)
  - ID/EX.RegRS : register for Rs value (32 bits)
  - ID/EX.RegRT : register for Rt value (32 bits)
  - ID/EX.PC : register for the passage of incremented Program Counter (32 bits)
  - ID/EX.S32 : register for 32-bit sign-extension (32 bits)
- EX/MEM (97 bits)
  - EX/MEM.BrAddr = register for the Computed Branch Target Address (32 bits)
  - EX/MEM.Zero = register for register compared result (1 bit)
  - EX/MEM.ALUResu = register for ALU result (32 bits)
  - EX/MEM.RegRT = register for the passage of Rt value (32 bits)
- MEM/WB (64 bits)
  - MEM/WB.DMData = register for the data from data memory (32 bits)
  - MEM/WB.ALUResu = register for the ALU result passage (32 bits)

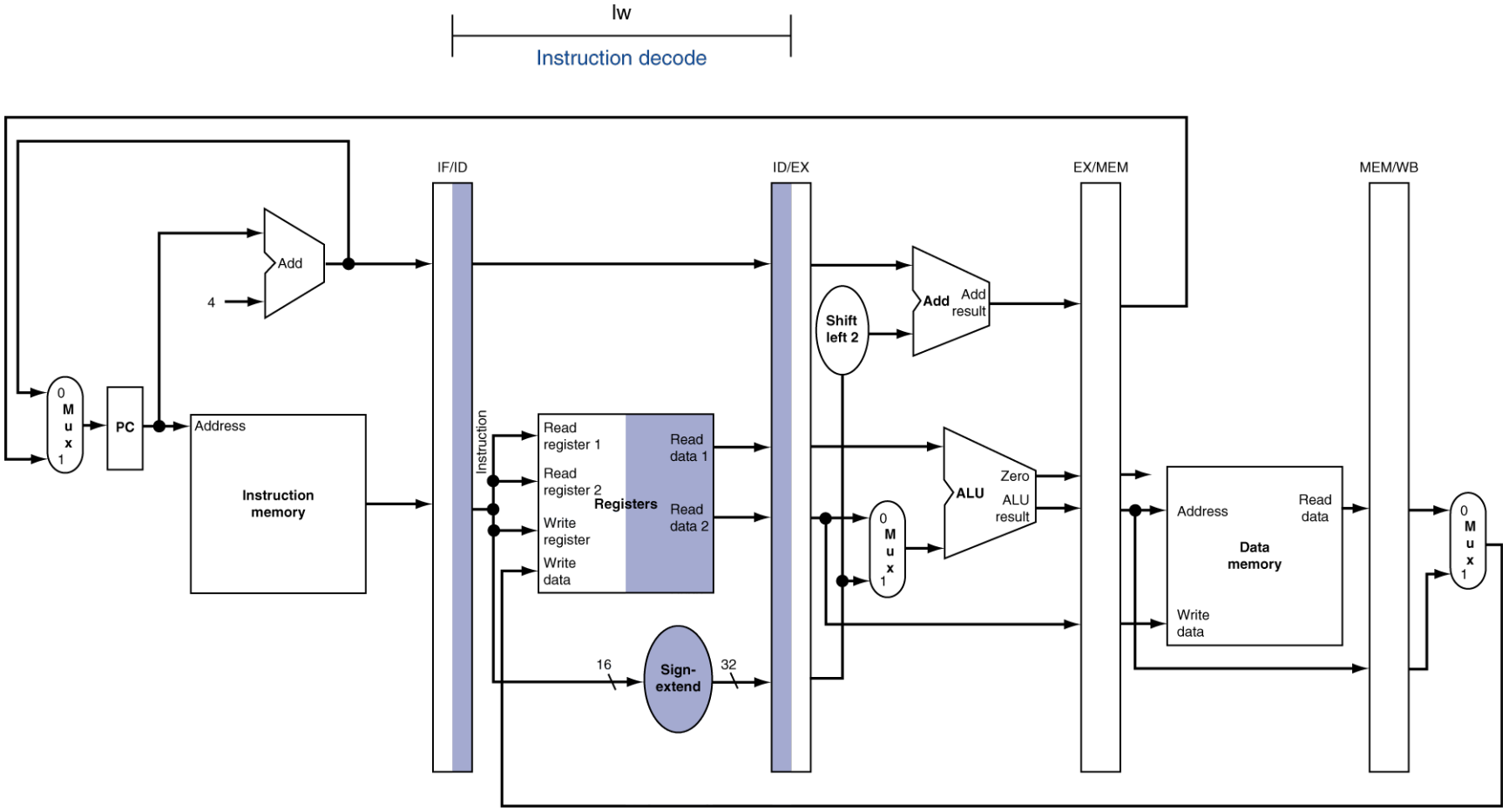
Pipeline Operation

- 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

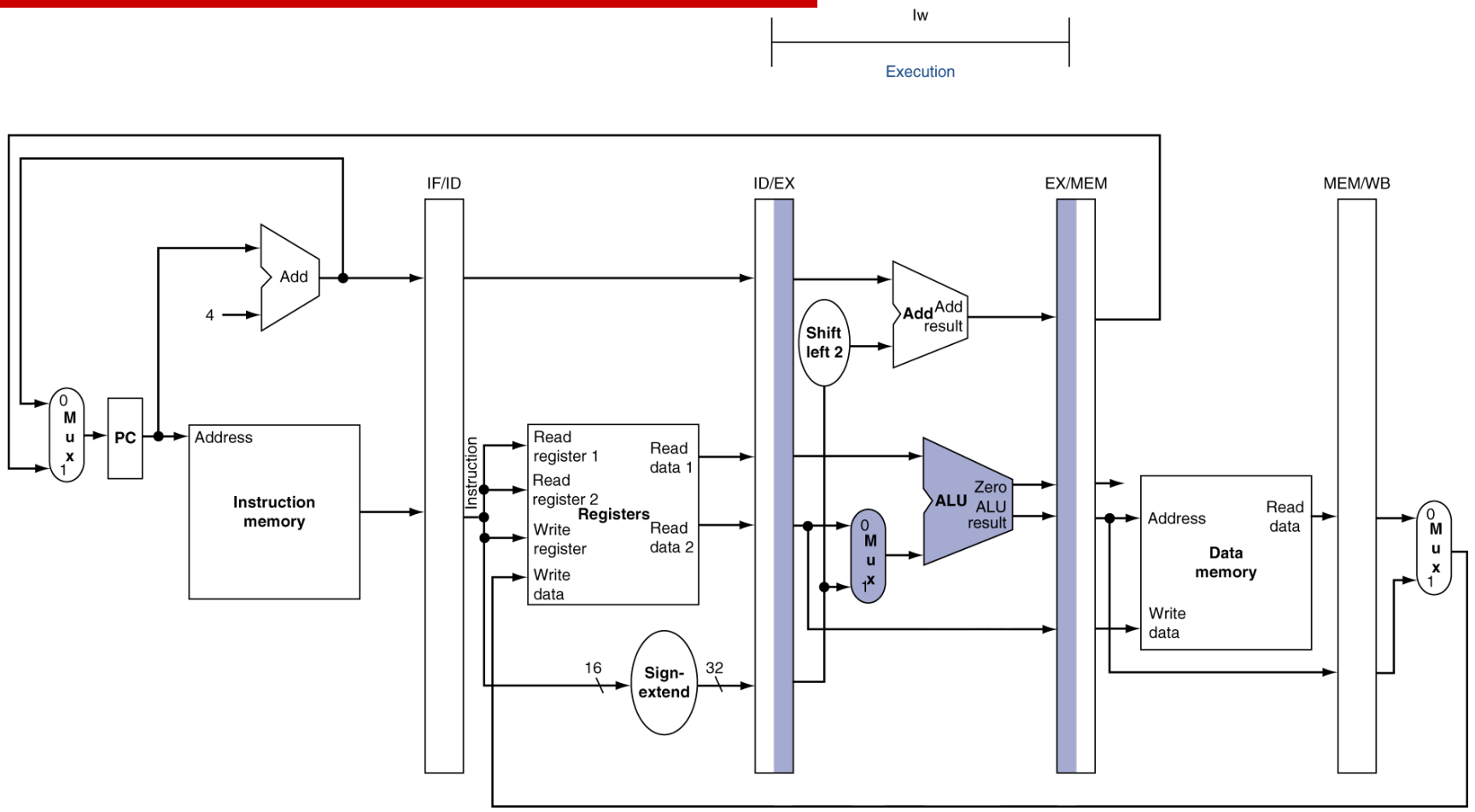
IF for Load, Store, ... (Fig. 4.36)



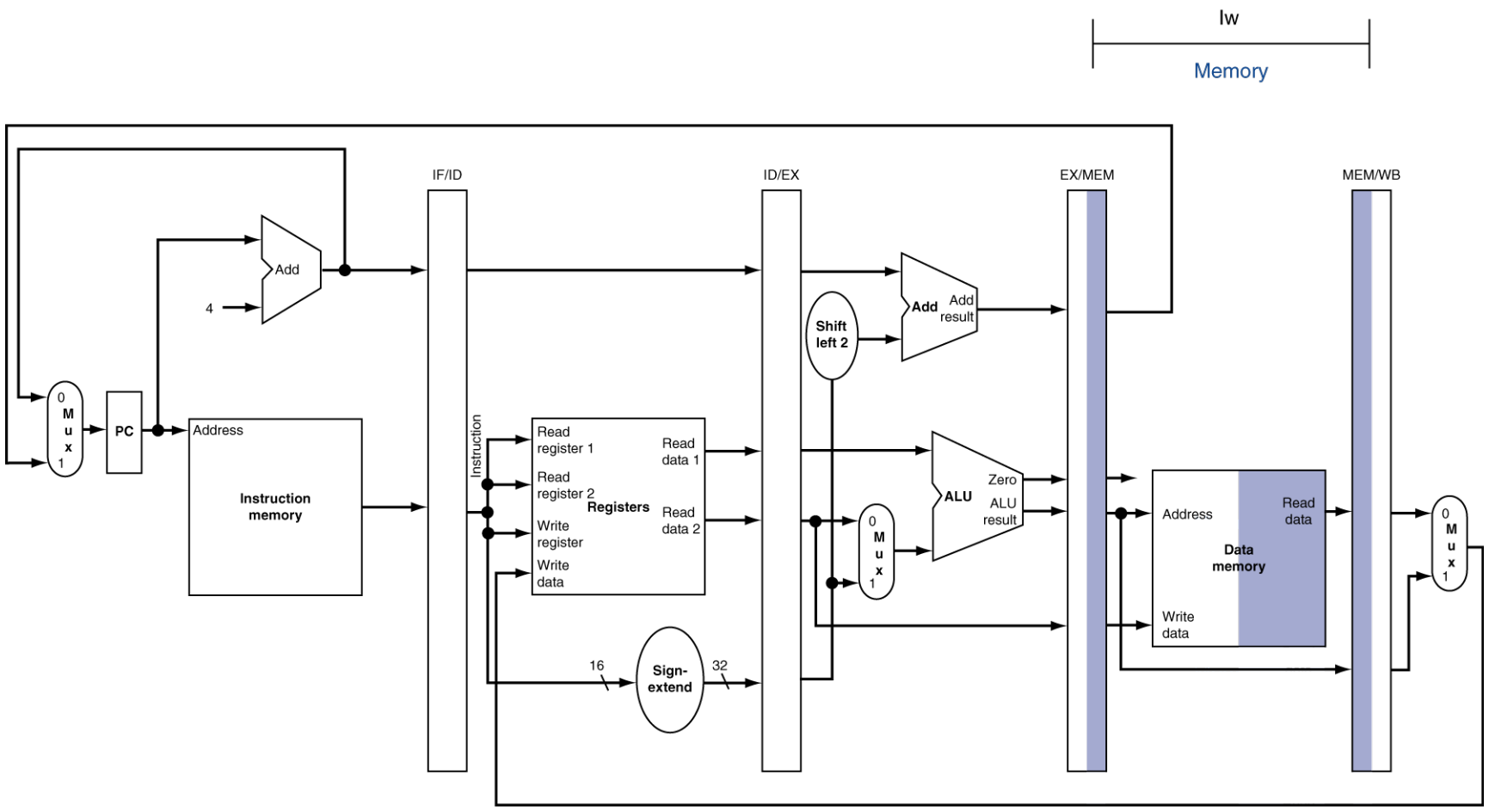
ID for Load, Store, ... (Fig. 4.36)



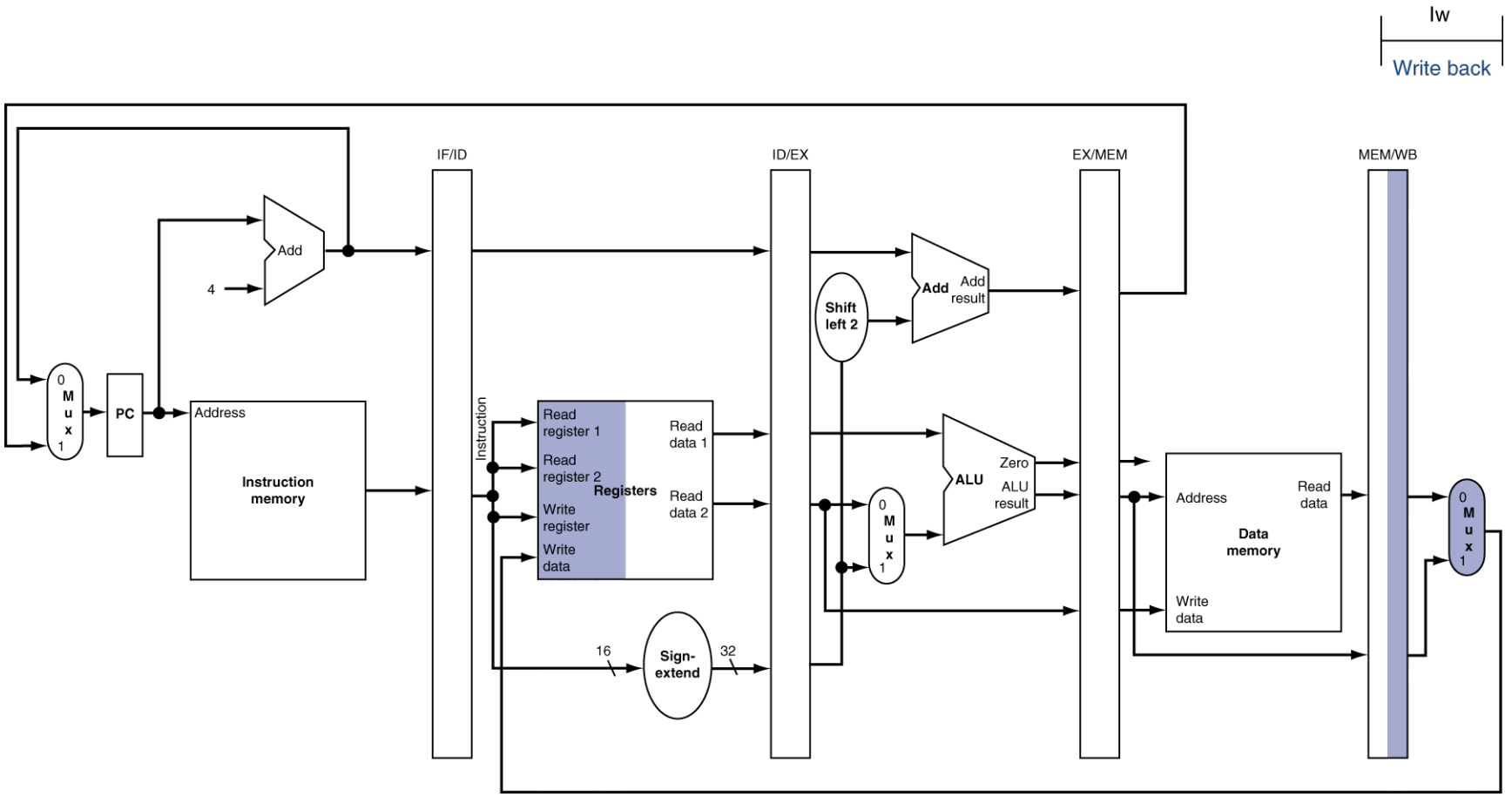
EX for Load (Fig. 4.37)



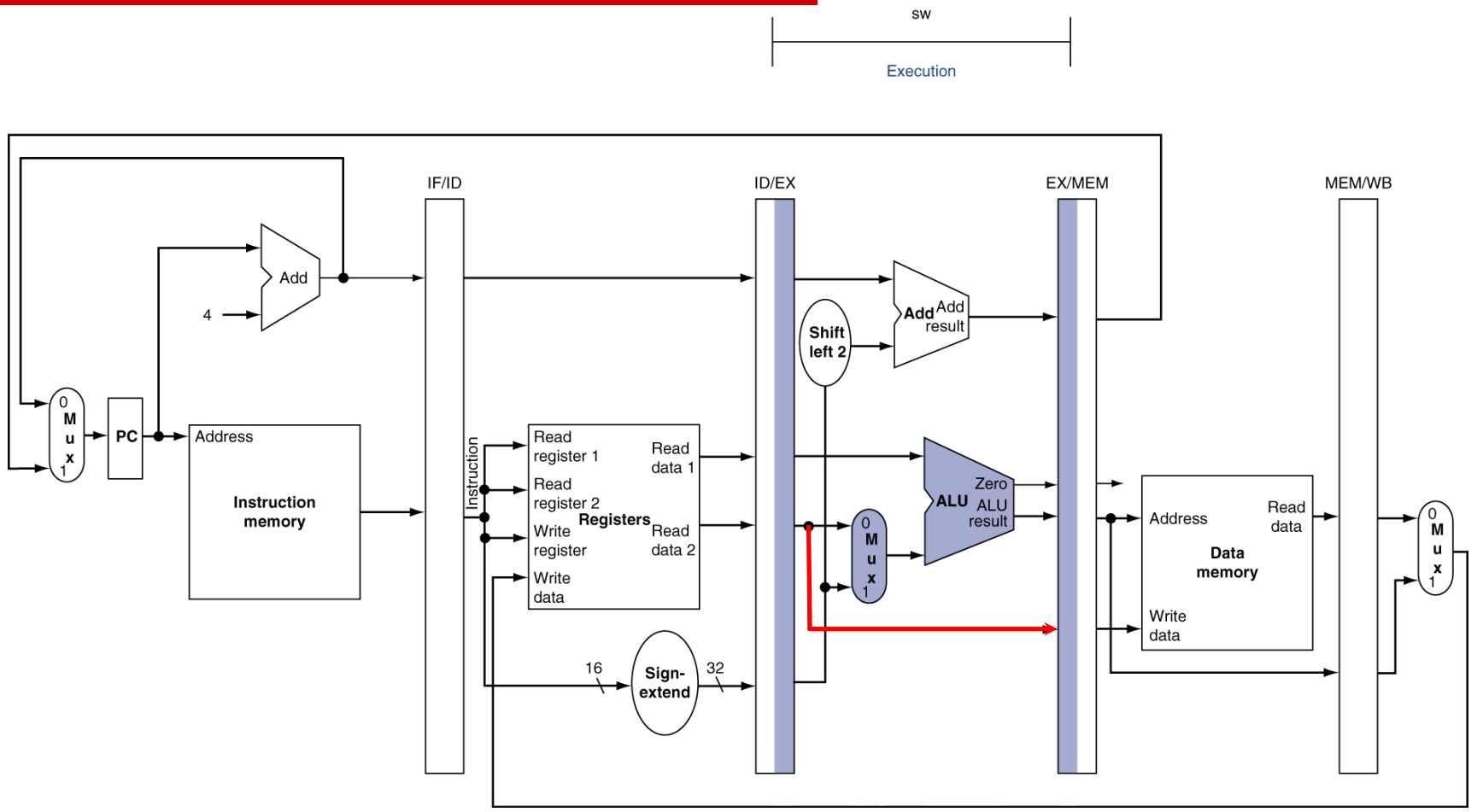
MEM for Load (Fig. 4.38)



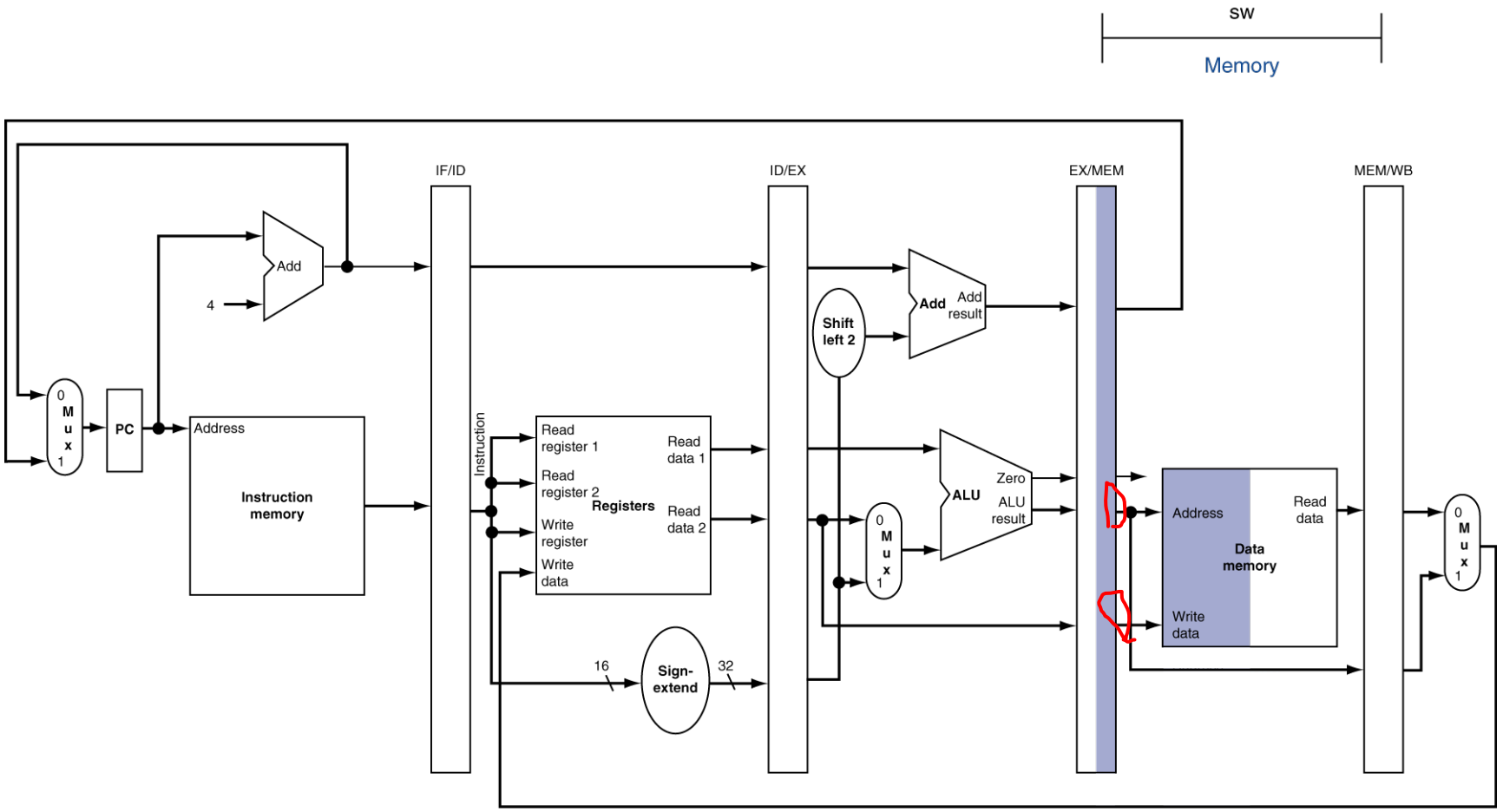
WB for Load (Fig. 4.38)



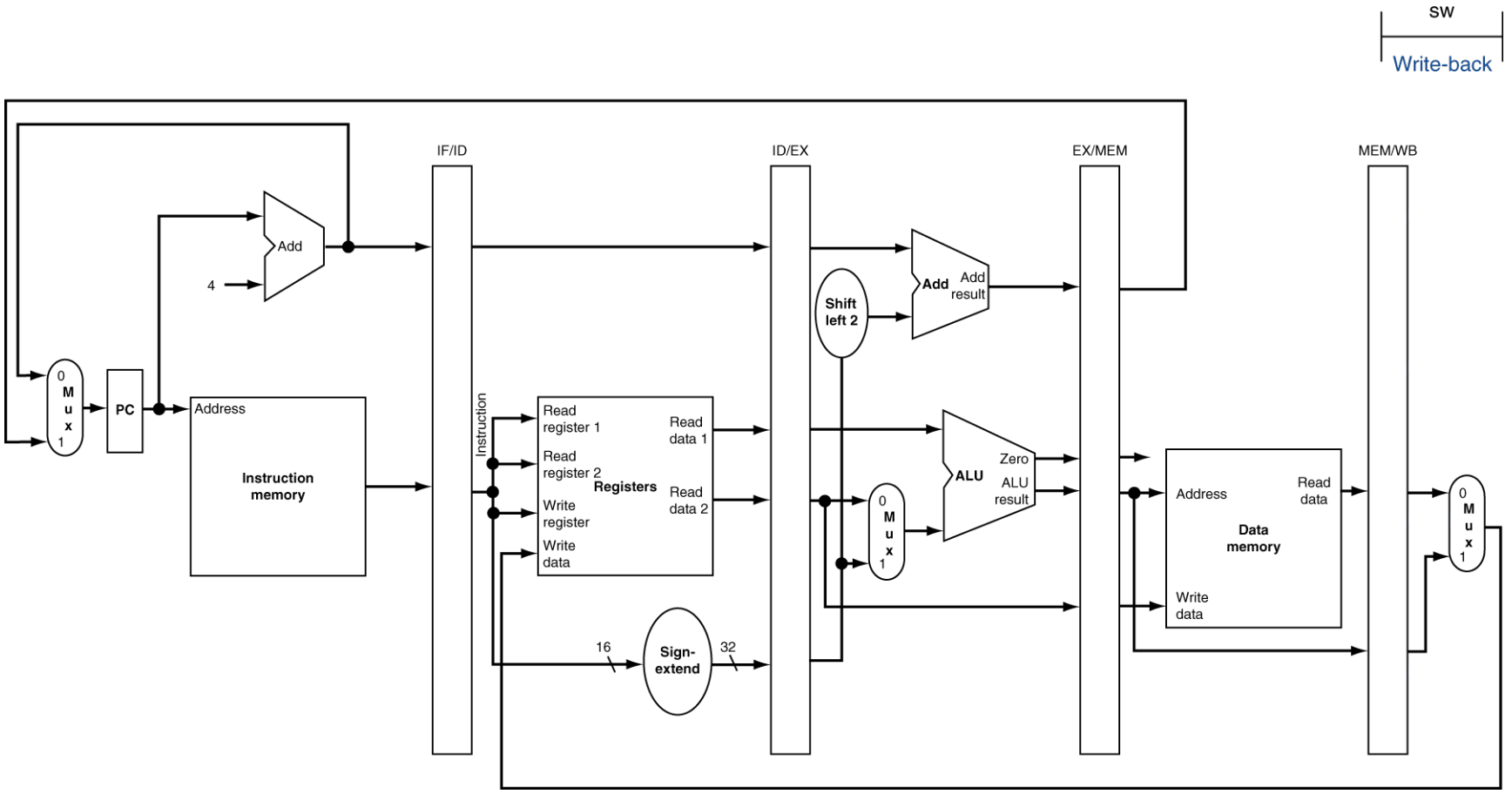
EX for Store (Fig. 4.39)



MEM for Store (Fig. 4.40)



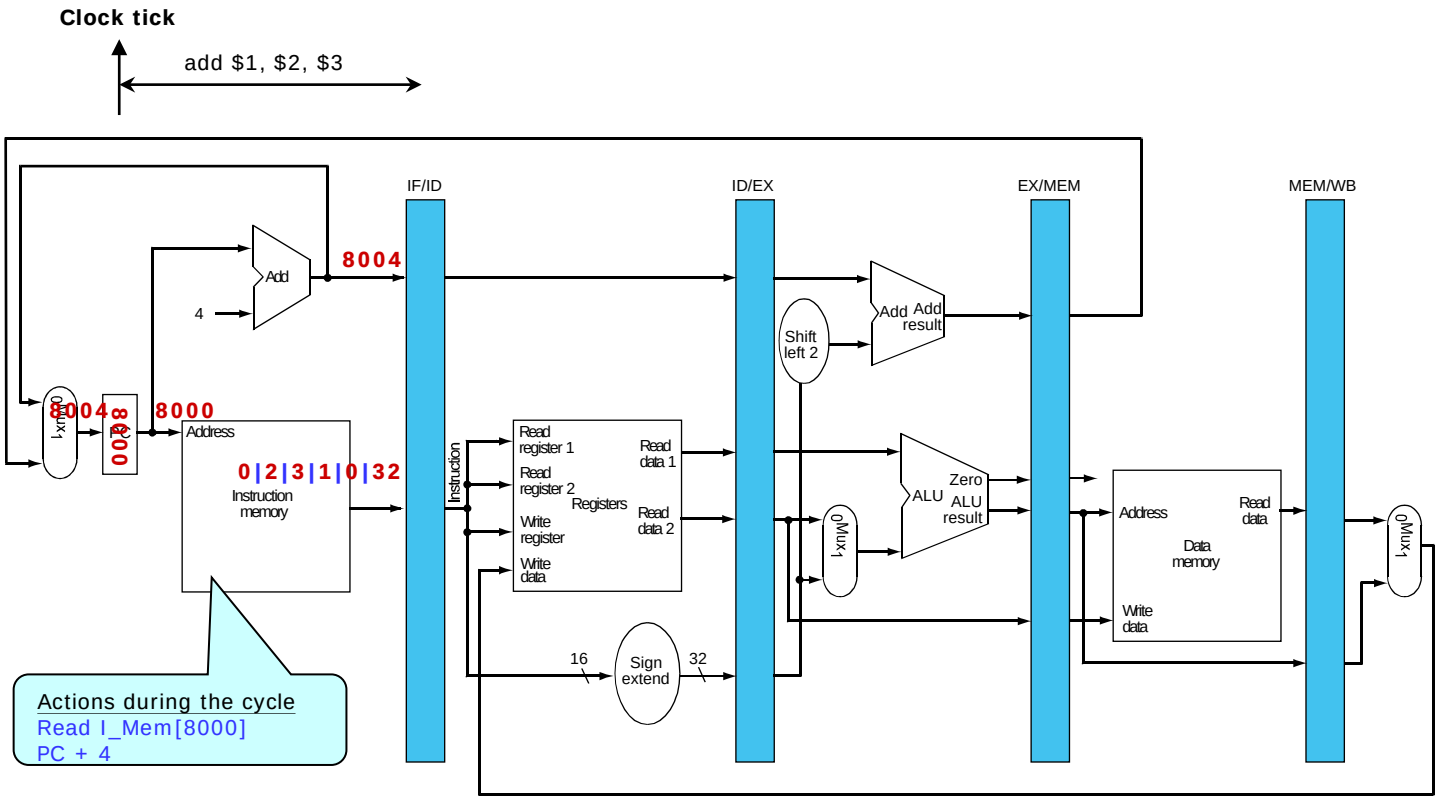
WB for Store (Fig. 4.40)



Example – Cycle 1

Addr: Instruction
8000: add $1, $2, $3
8004: lw $4, 4($5)
8008: sw $6, 8($5)
8012: and $7, $2, $3
8016: sub $8, $0, $9

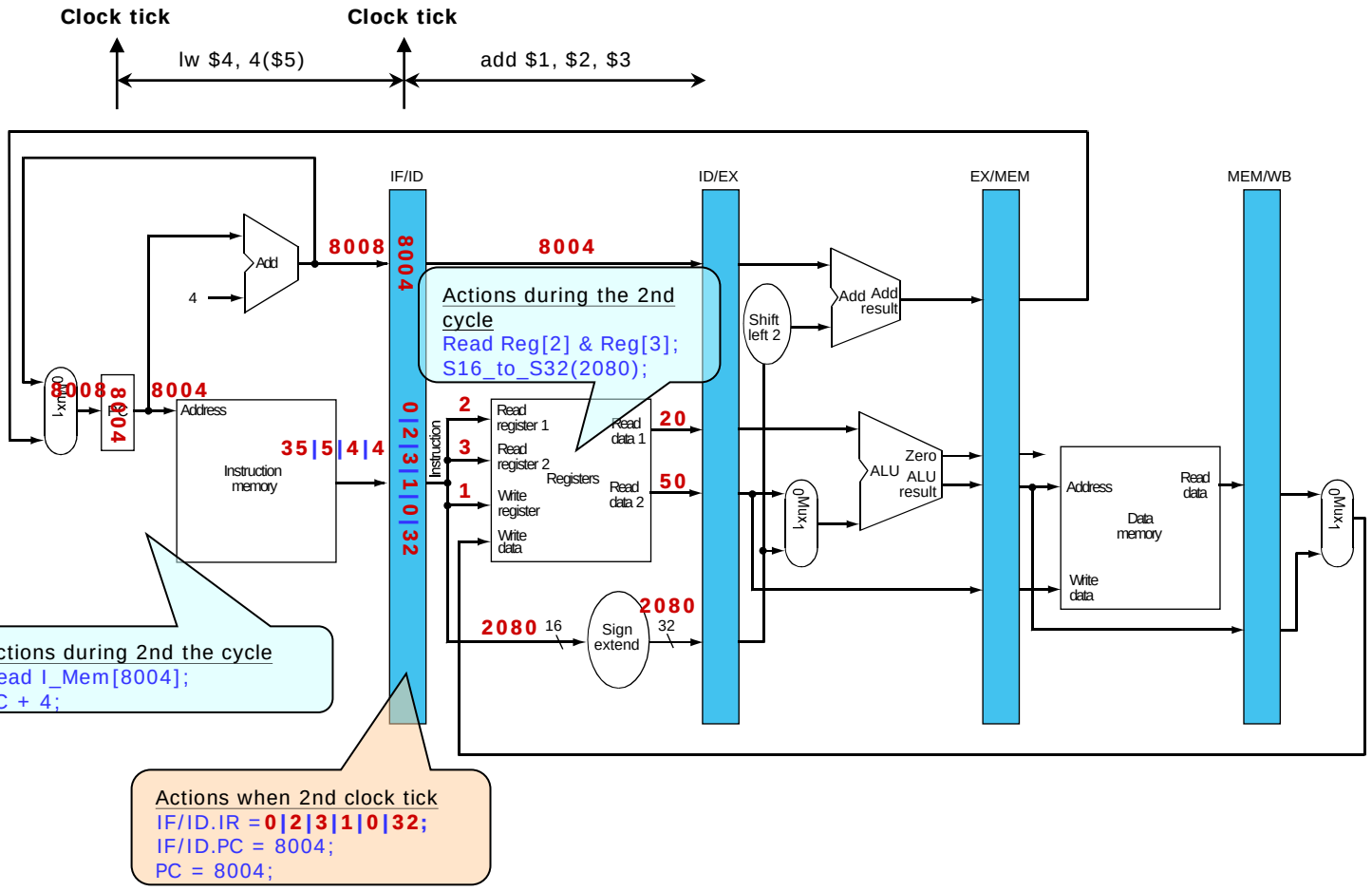
Reg#	Value
0	0
1	0
2	20
3	50
4	10
5	600
6	88
7	1
8	2
9	3



Example – Cycle 2

Addr: Instruction
8000: add $1, $2, $3
8004: lw $4, 4($5)
8008: sw $6, 8($5)
8012: and $7, $2, $3
8016: sub $8, $0, $9

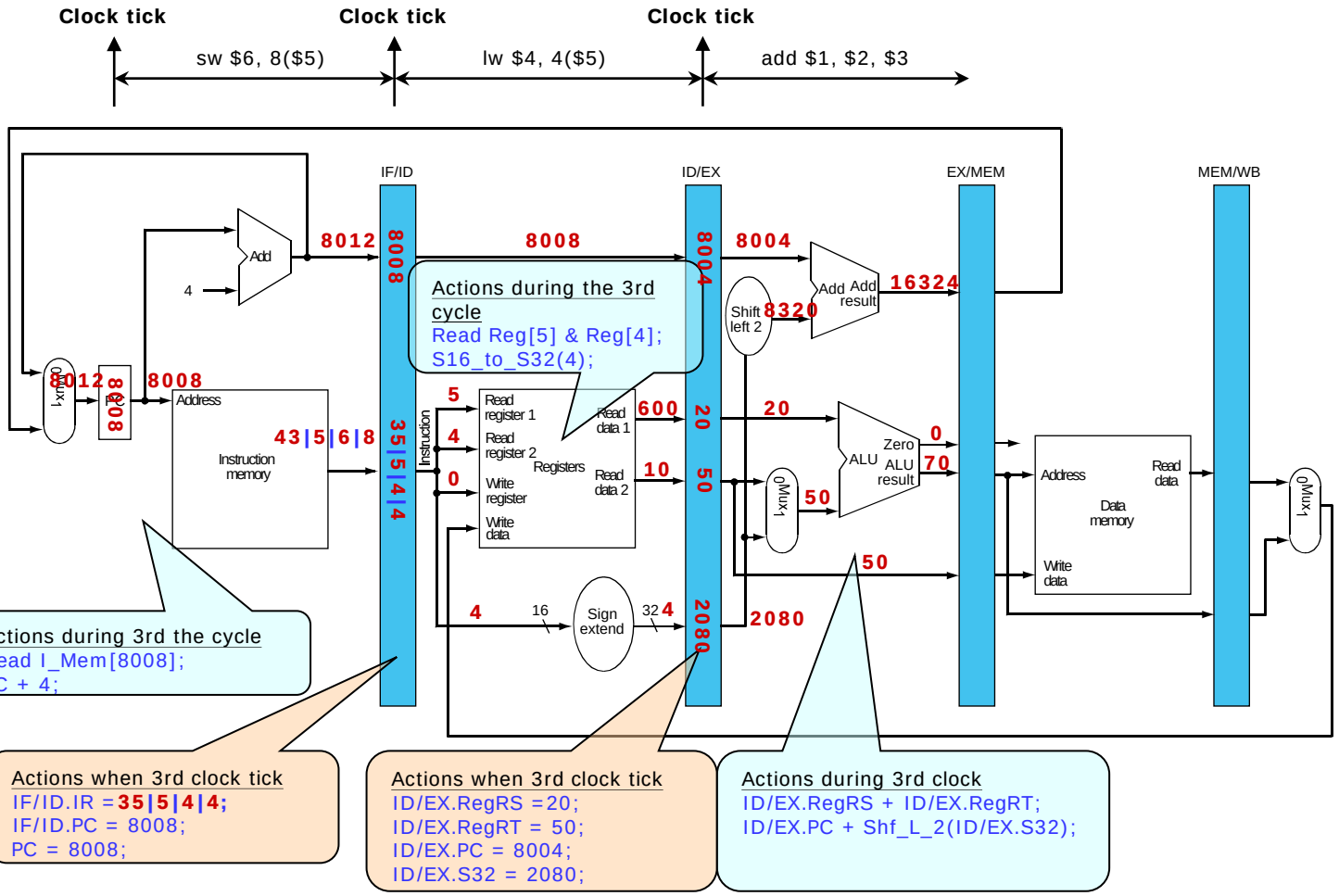
Reg#	Value
0	0
1	0
2	20
3	50
4	10
5	600
6	88
7	1
8	2
9	3



Example – Cycle 3

Addr: Instruction
8000: add $1, $2, $3
8004: lw $4, 4($5)
8008: sw $6, 8($5)
8012: and $7, $2, $3
8016: sub $8, $0, $9

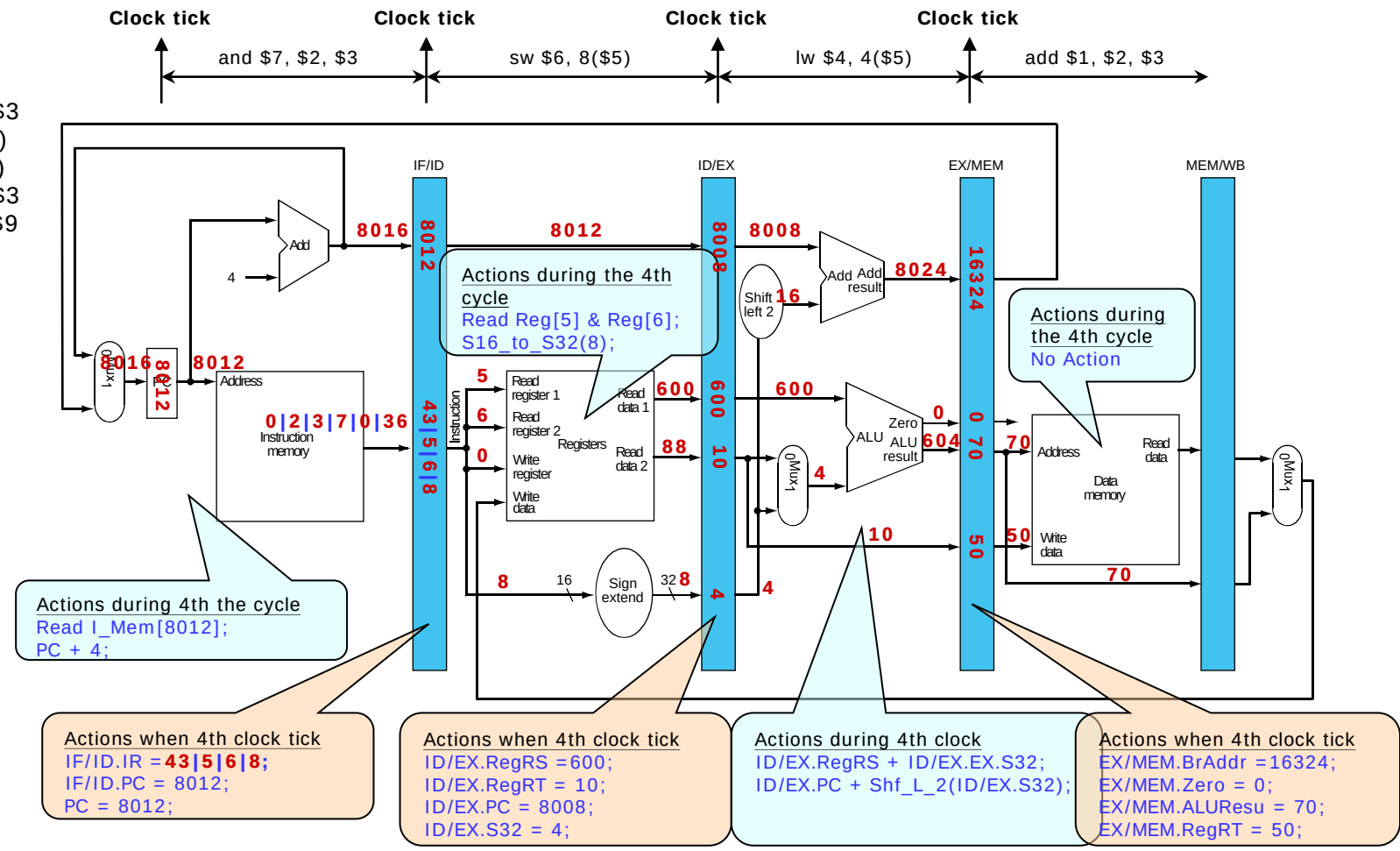
Reg#	Value
0	0
1	0
2	20
3	50
4	10
5	600
6	88
7	1
8	2
9	3



Example – Cycle 4

Addr: Instruction
8000: add $1, $2, $3
8004: lw $4, 4($5)
8008: sw $6, 8($5)
8012: and $7, $2, $3
8016: sub $8, $0, $9

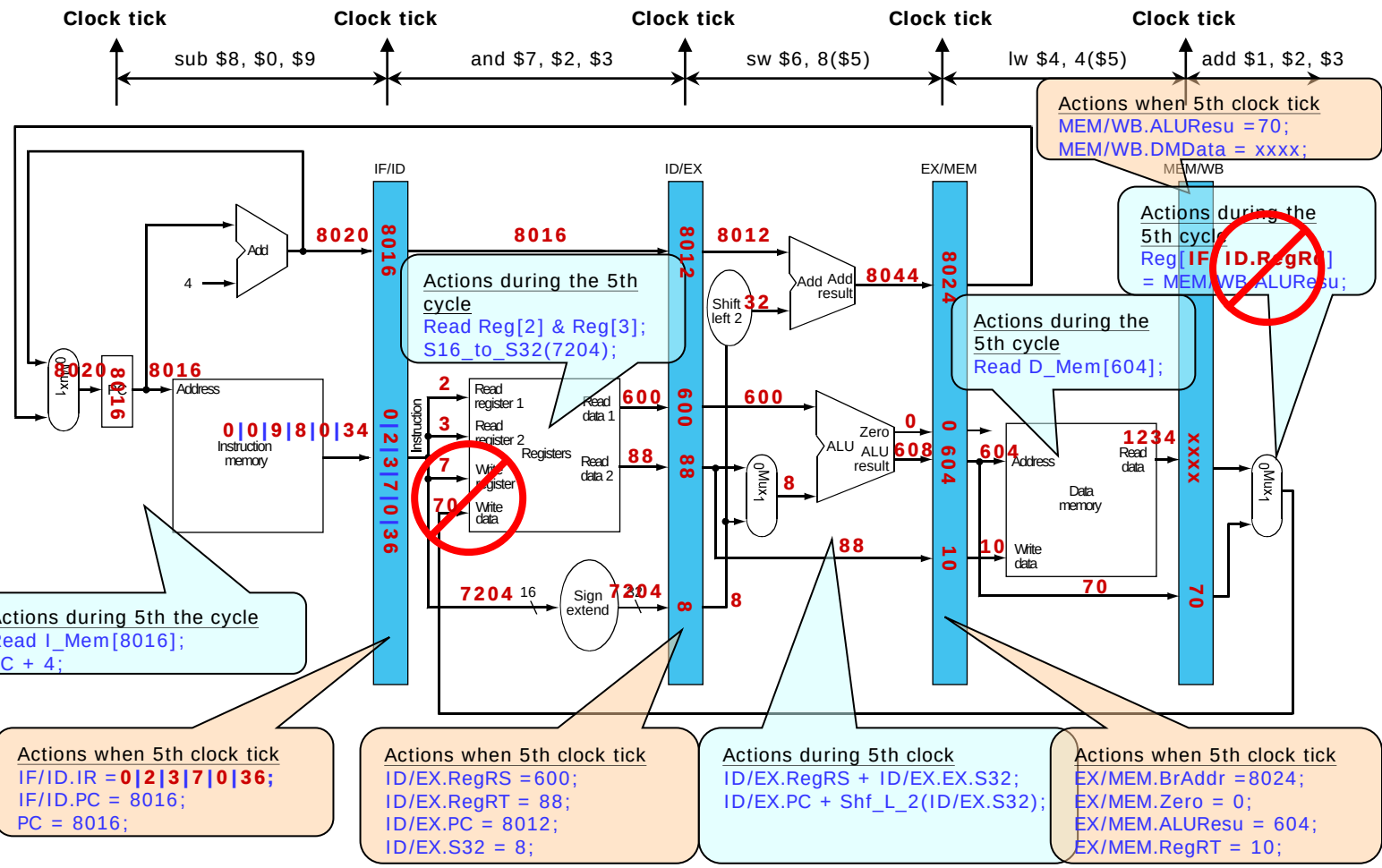
Reg#	Value
0	0
1	0
2	20
3	50
4	10
5	600
6	88
7	1
8	2
9	3



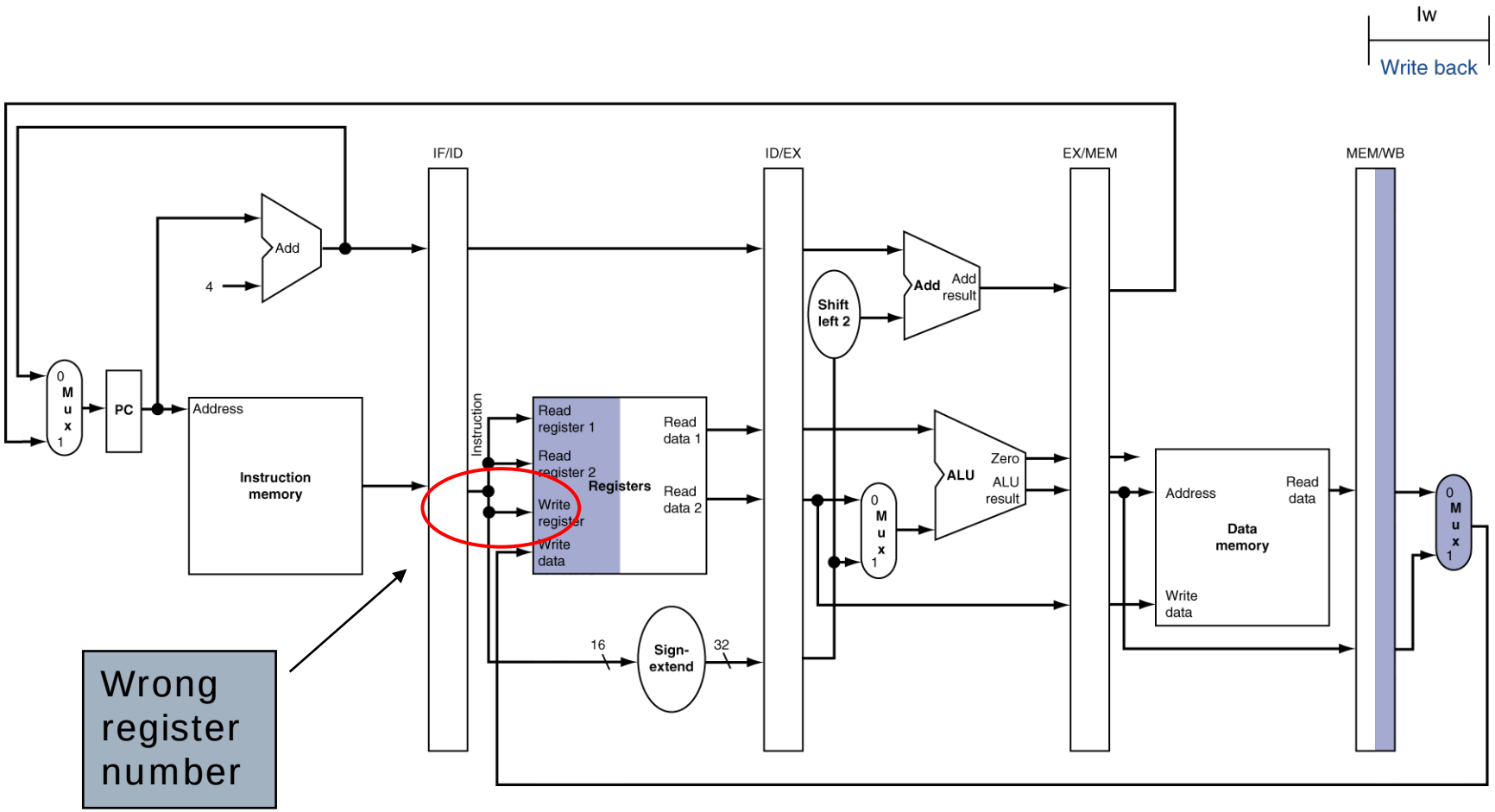
Example – Cycle 5

Addr: Instruction
8000: add $1, $2, $3
8004: lw $4, 4($5)
8008: sw $6, 8($5)
8012: and $7, $2, $3
8016: sub $8, $0, $9

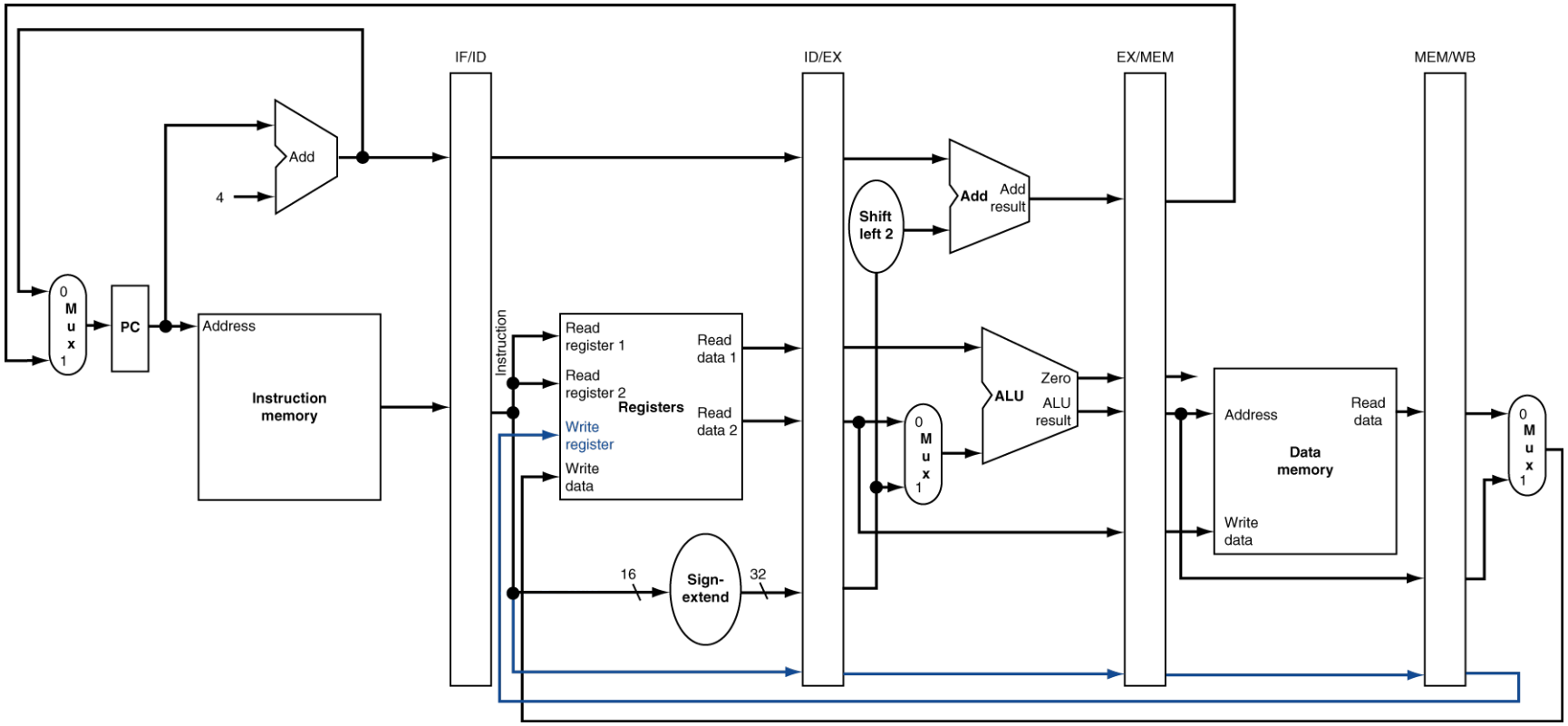
Reg#	Value
0	0
1	0
2	20
3	50
4	10
5	600
6	88
7	1->70
8	2
9	3



WB for Load

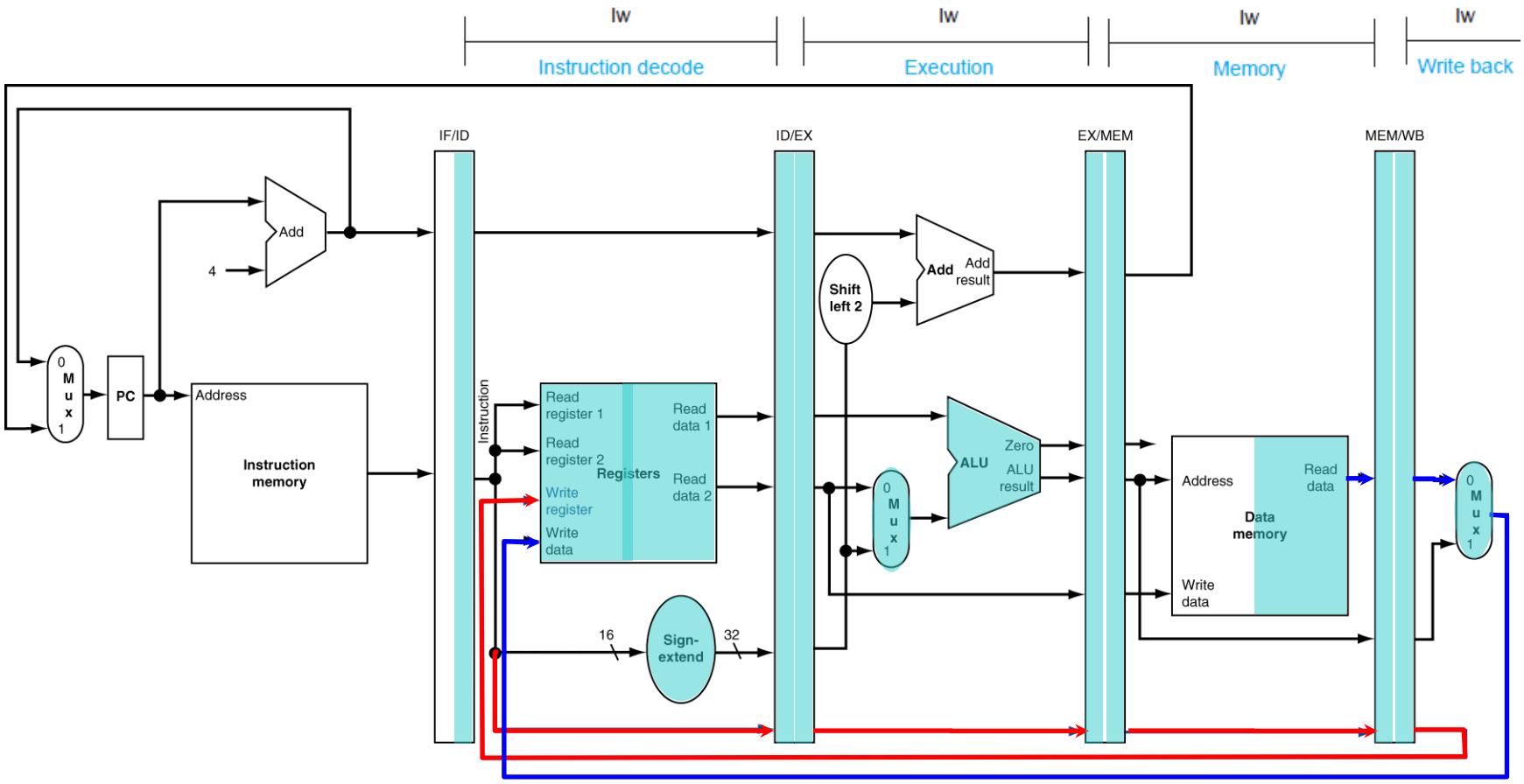

lw
Write back
IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
0
M
u
x
1
PC
Address
Instruction memory
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 2
Add
Add result
ALU
Zero
ALU result
Address
Read data
Data memory
Write data
16
Sign-extend
32
Wrong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

Corrected Datapath (Fig. 4.41)



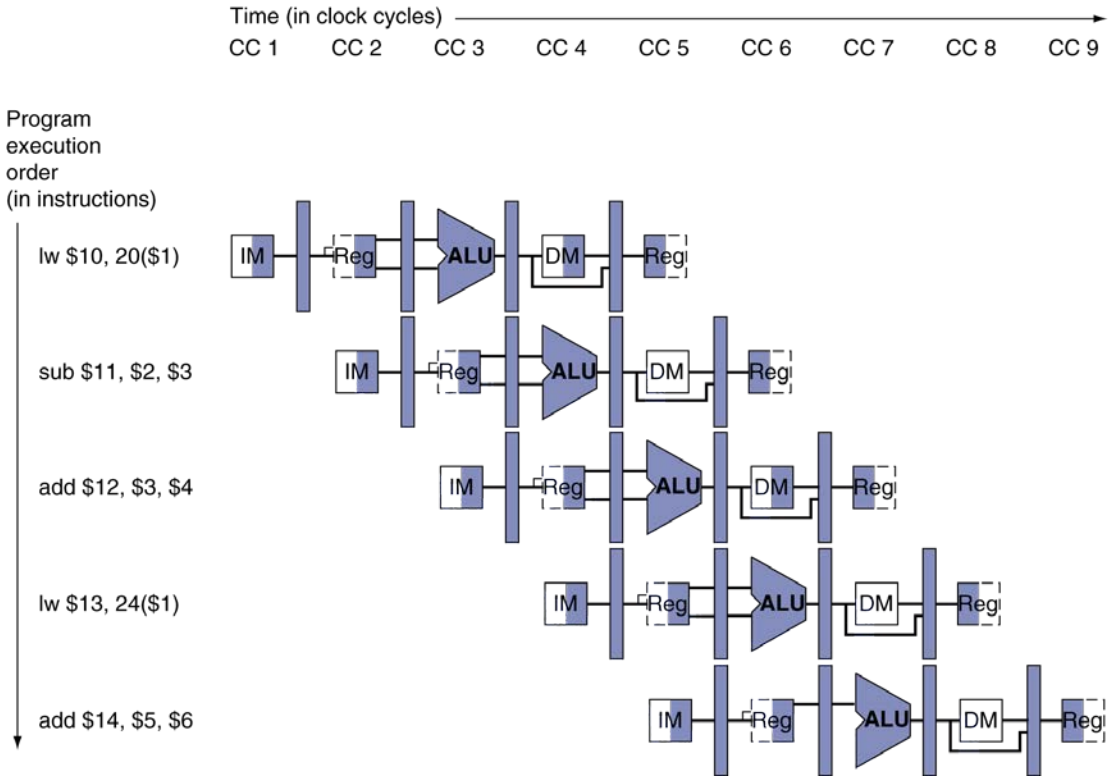
All the information (including control signals) required by all instructions on the executing stages have to be carried !!

Corrected Datapath for Load



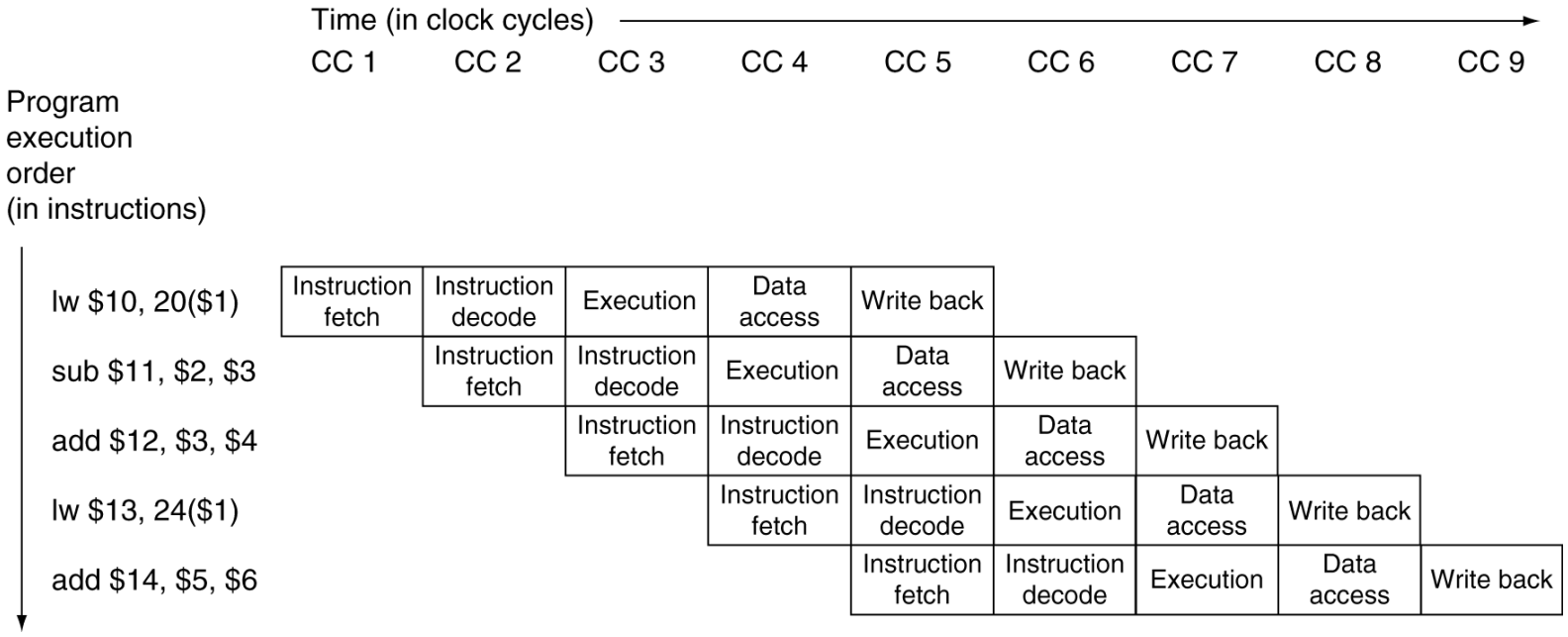
Graphically Representing Pipelines (p. 310, Fig. 4.43)

- Multiple-clock-cycle pipelining diagram of five instructions
- Form showing resource usage



Graphically Representing Pipelines (Fig. 4.44)

- Traditional multiple-clock-cycle pipelining diagram of five instructions



Graphically Representing Pipelines (Fig. 4.45)

- The single-clock-cycle diagram corresponding to clock cycle 5 of the pipeline.