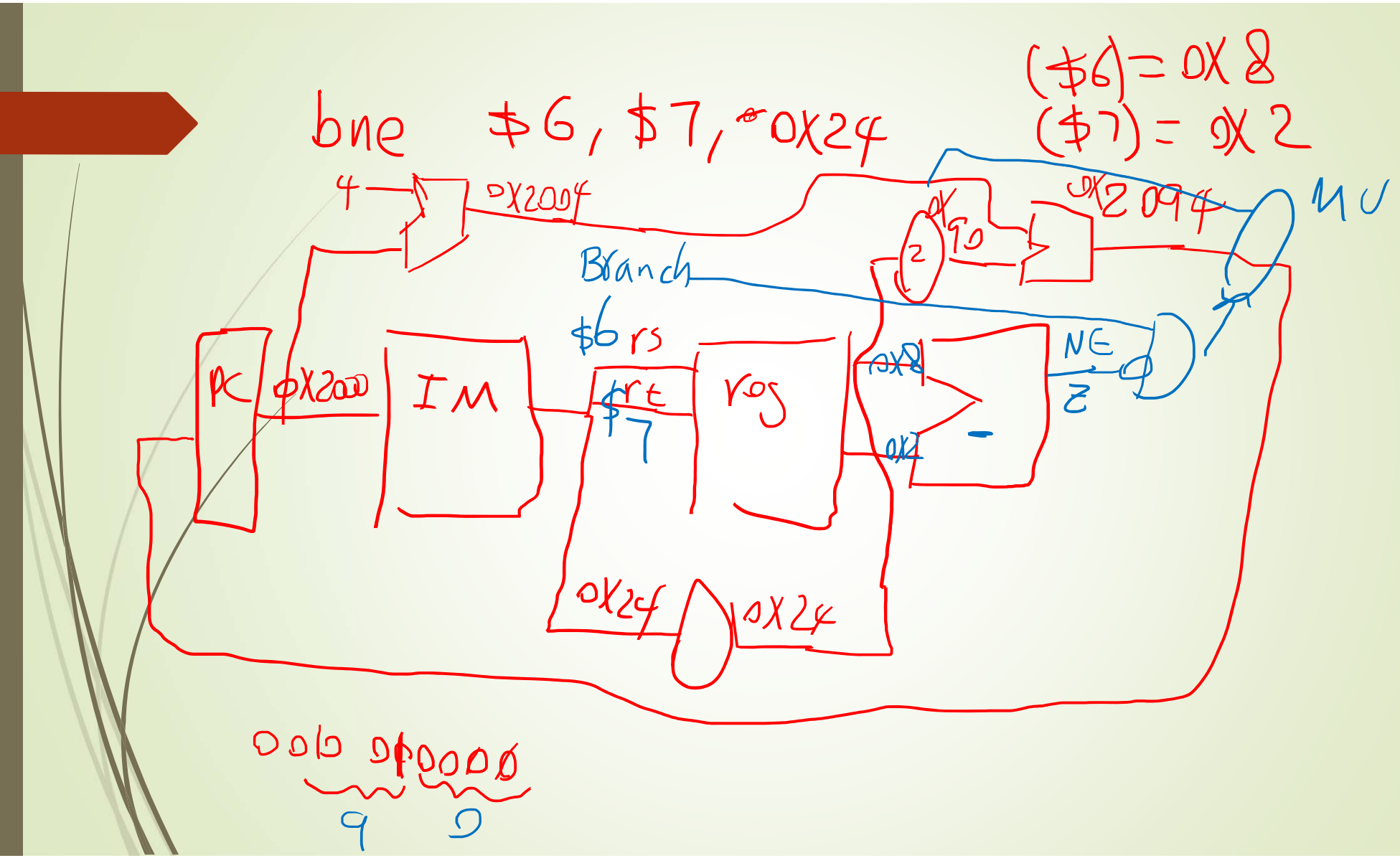
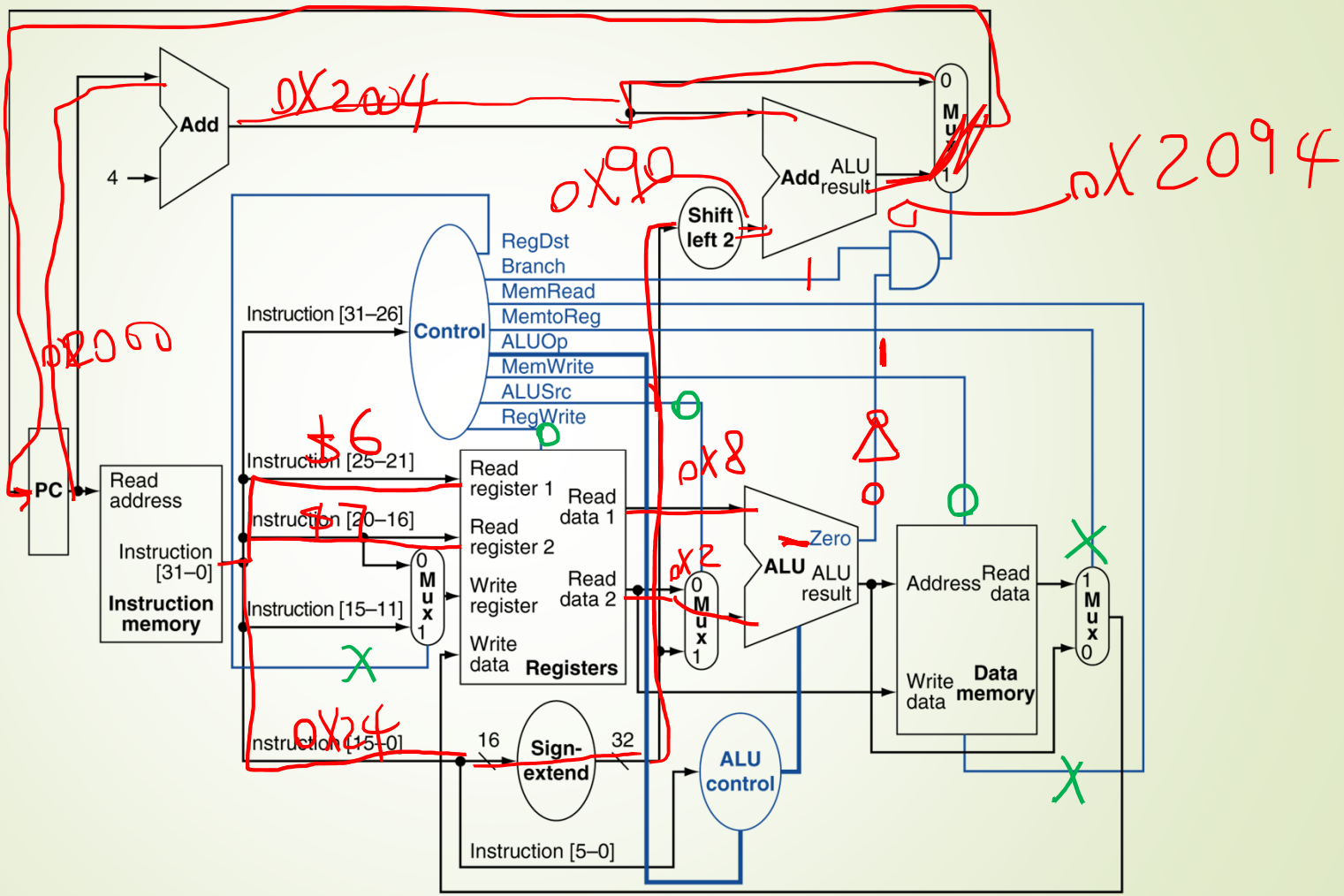
EGC442
Class Notes
3/31/2023

Baback Izadi
Division of Engineering Programs
bai@engr.newpaltz.edu


bne $6, $7, 0x24
($6) = 0x8
($7) = 0x2
4
0x2004
0x2094
Branch
$6 rs
PC
0x200
IM
$rt
$7
rs
0x8
0x2
NE
Z
0x24
0x24
MU
0010 000000
9 0

bne $6, $7, 0x24


Add
4
0x2004
0x90
Add
ALU result
0x2094
Shift left 2
M u x
0
1
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
0x200
$6
$7
0x8
0x2
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
16
Sign-extend
32
ALU control
Instruction [5–0]
0x24

1) How many stages exist in the laundry analogy?

4

Check Show answer

Correct

4

The four stages are washing, drying, folding, and storing. Each stage is assumed to require 30 minutes.

2) If laundry is done sequentially, how many minutes do 60 loads take to wash, dry, fold, and put away?

7200 minutes

Check Show answer

Correct

7200

One load takes 30 × 4 or 120 minutes, so 60 loads takes 60 × 120 = 7200 minutes. Seems like a shame that the washing machine isn't being used at the same time as the dryer though, right?

3) If laundry is done in a pipelined manner, execution is nearly 4 times faster than if done sequentially. Suppose doing 50 loads sequentially requires 6000 minutes. How long would those 50 loads take if done in a pipelined manner? Assume a 4 times speedup (ignore the fact that some stages are unused for the first few and last few loads).

1500 minutes

Check Show answer

Correct

1500

6000 / 4 = 1500. If pipelining stages are equal in length and the stages can be kept busy, speedup can approach the number of stages, making pipelining very attractive.

4) Each load of laundry takes 4 × 30 = 120 minutes to wash, dry, fold, and store (30 minutes each). How many minutes are required to complete one load of laundry when multiple loads of laundry are done in a pipelined manner?

120 minutes

Check Show answer

Correct

120

Pipelining doesn't affect the time to do one particular load. Each stage still takes 30 minutes, so one load takes 4 × 30 = 120 minutes to wash, dry, fold, and store. What pipelining does is start washing a second load while the first load is drying, and start washing a third load while the second load is drying and the first is being folded, and so on. So multiple loads are being processed at the same time.

1) The nonpipelined datapath implementation has how many stages?

- (●) 1
- ( ) 5

Correct

The implementation has just one long stage. Such a stage is akin to a single laundry person doing laundry by hand, who first washes, then blow dries, then folds, and finally stores each load sequentially.

2) The pipelined datapath implementation has how many stages?

- ( ) 1
- (●) 5

Correct

The one long stage has been divided into 5 shorter stages: Instr fetch, Reg read, ALU op, Data access, and Reg write. Such division is akin to replacing one laundry-by-hand person with four people/machines: a washer, dryer, folder, and storer.

3) The above figure shows the five stages as: Instruction fetch, Reg, ALU, Data access, and Reg. Are the two Regs doing the same thing?

- ( ) Yes
- (●) No

Correct

No, the first Reg represents the register read stage, while the second Reg represents the register write stage. The two just happen to use the same abbreviated word Reg.

4) Does every instruction require all 5 stages?

- ( ) Yes
- (●) No

Correct

beq, for example, only requires the first three stages. However, for simplicity, every instruction will take 5 stages long.

5) Suppose Instr1 is fetched in stage 1. Instr1 then proceeds to stage 2, Reg read. In a pipelined implementation, can Instruction2 be fetched simultaneously with that Reg read?

- (●) Yes
- ( ) No

Correct

Indeed, such overlapping execution is the main point of pipelining.

10) On computer X, a nonpipelined instruction execution would require 12 ns, and thus 12 ns exists between instructions. A pipelined implementation uses 6 equal-length stages of 2 ns each, resulting in _____ ns between instructions.

- ○ 1
- ◉ 2

Correct

After 2 ns, a first instruction Instr1 would be done with stage 1, meaning Instr2 could begin stage 1. 1 ns is too short: Instr1 would only be halfway through stage 1 before Instr2 tried to use stage 1, resulting in a problem. The general equation is 12 ns / 6 stages = 2 ns between instructions.

11) On computer X, a nonpipelined instruction execution would require 12 ns. A pipelined implementation uses 6 equal-length stages of 2 ns each. Assuming one million instructions execute and ignoring empty stages at the start/end, what is the speedup of the pipelined vs. non-pipelined implementation?

- ○ 2
- ◉ 6

Correct

Non-pipelined time: 1,000,000 * 12 ns = 12,000,000 ns.
Pipelined time: 1,000,000 * 2 ns = 2,000,000 (each instruction starts 2 ns after the previous, so each adds 2 ns).
12,000,000 ns / 2,000,000 ns = 6.

Under ideal conditions the speed-up from pipelining is approximately equal to the number of pipe stages.

12) Match the pipeline stage with the physical resource whose icon represents that stage in the stylized datapath depictions.

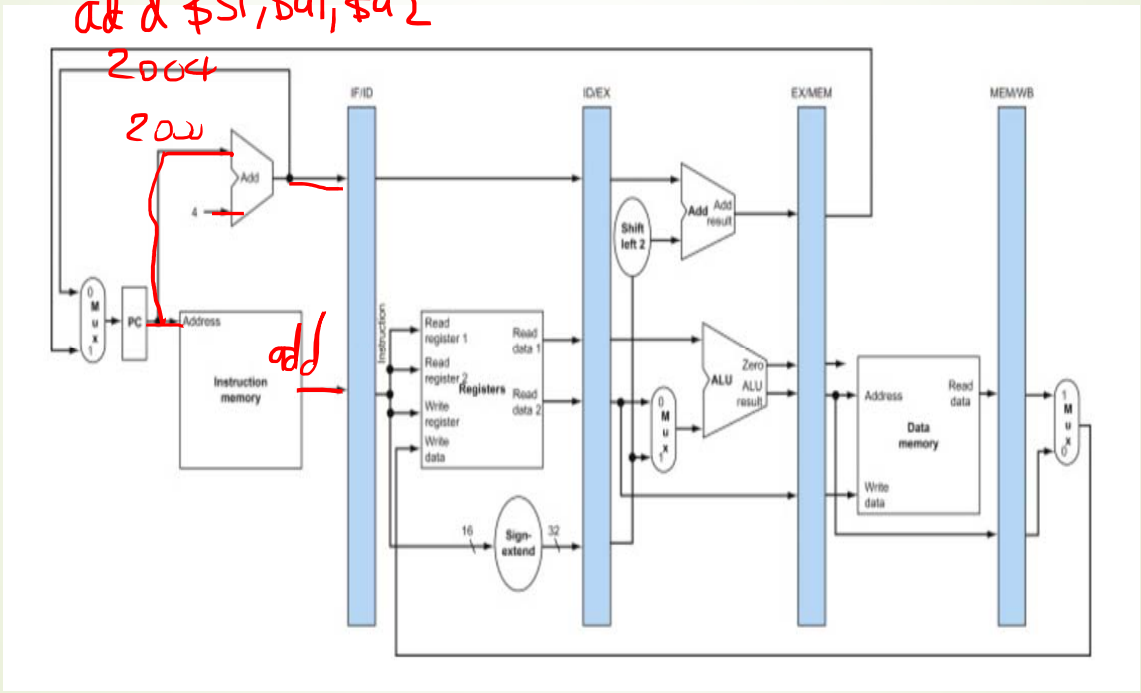
- **IM**
- **Reg (write)**
- **DM**
- **ALU**
- **Reg (read)**

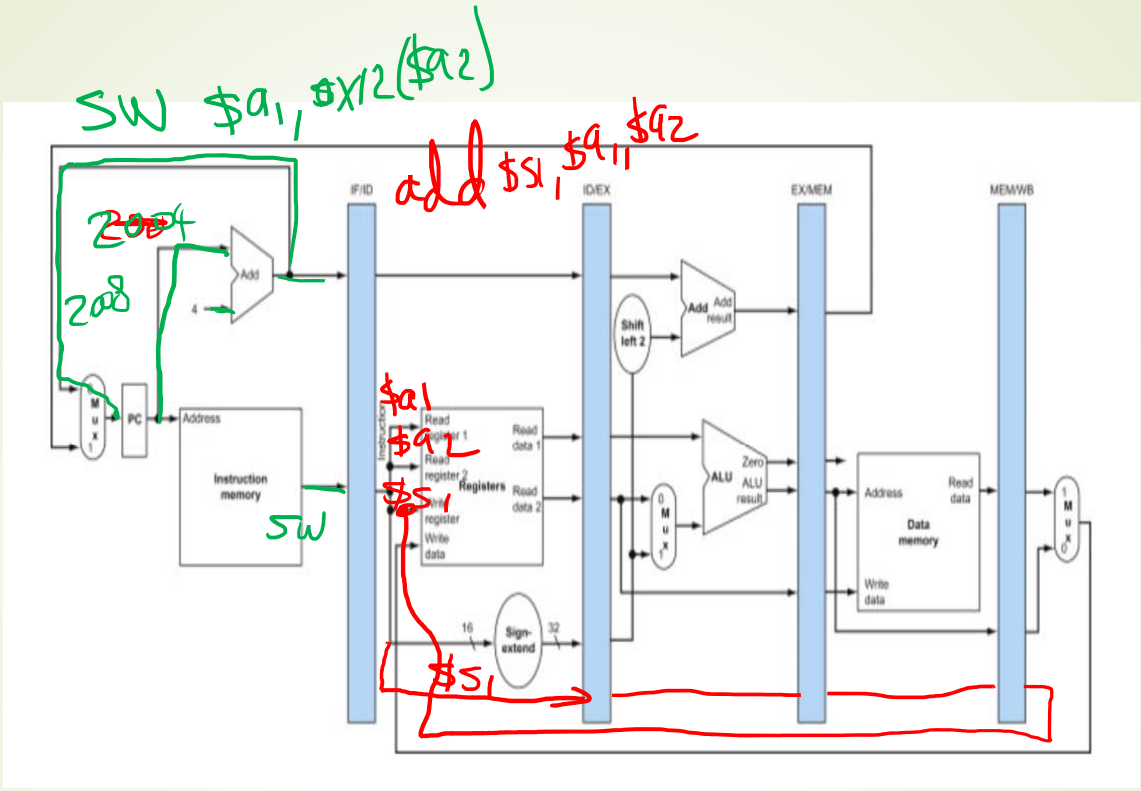
Stage 1: IF (instruction fetch) IM
Stage 2: ID (instruction decode) Reg (read)
Stage 3: EX (execute) ALU
Stage 4: MEM (data memory access) DM
Stage 5: WB (write back) Reg (write)

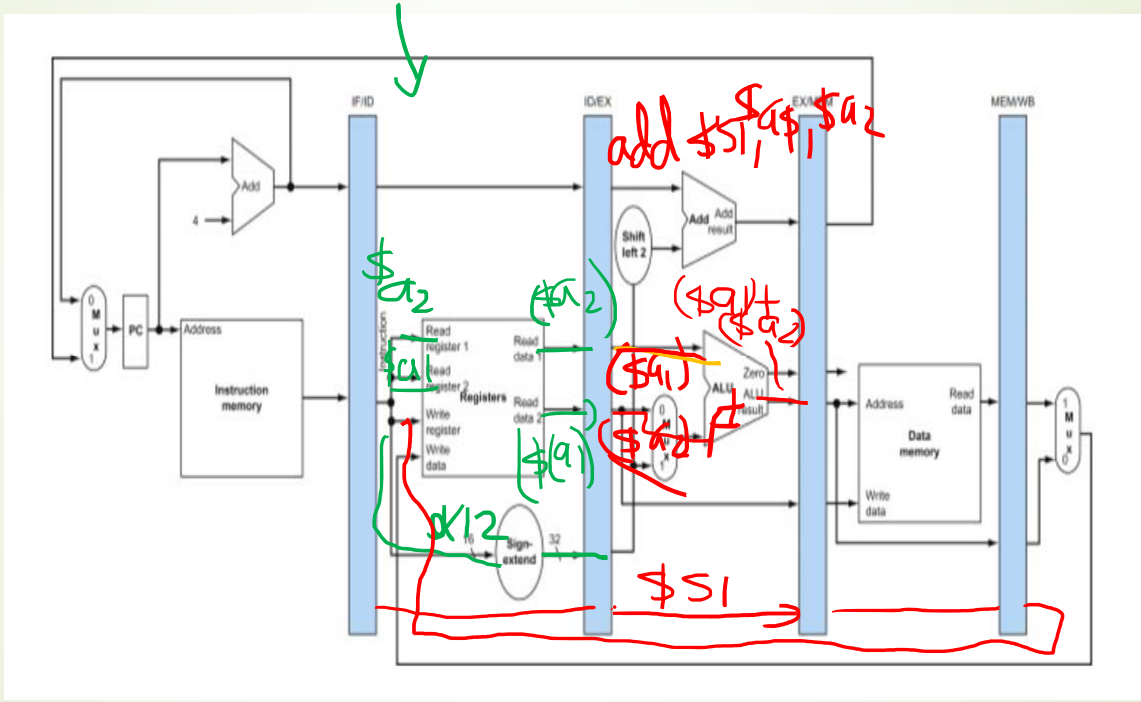
13) Using multiple copies of the following diagram, show the active stages for execution of the following sequence of instruction set

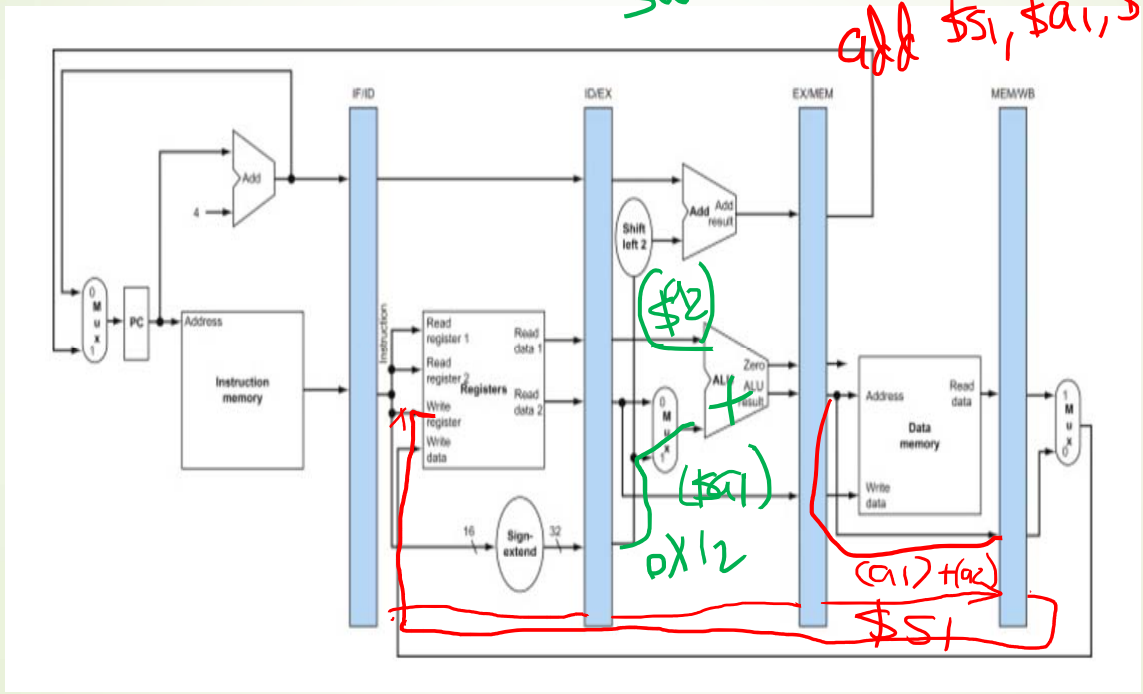
2000 add $s1, $a1, $a2

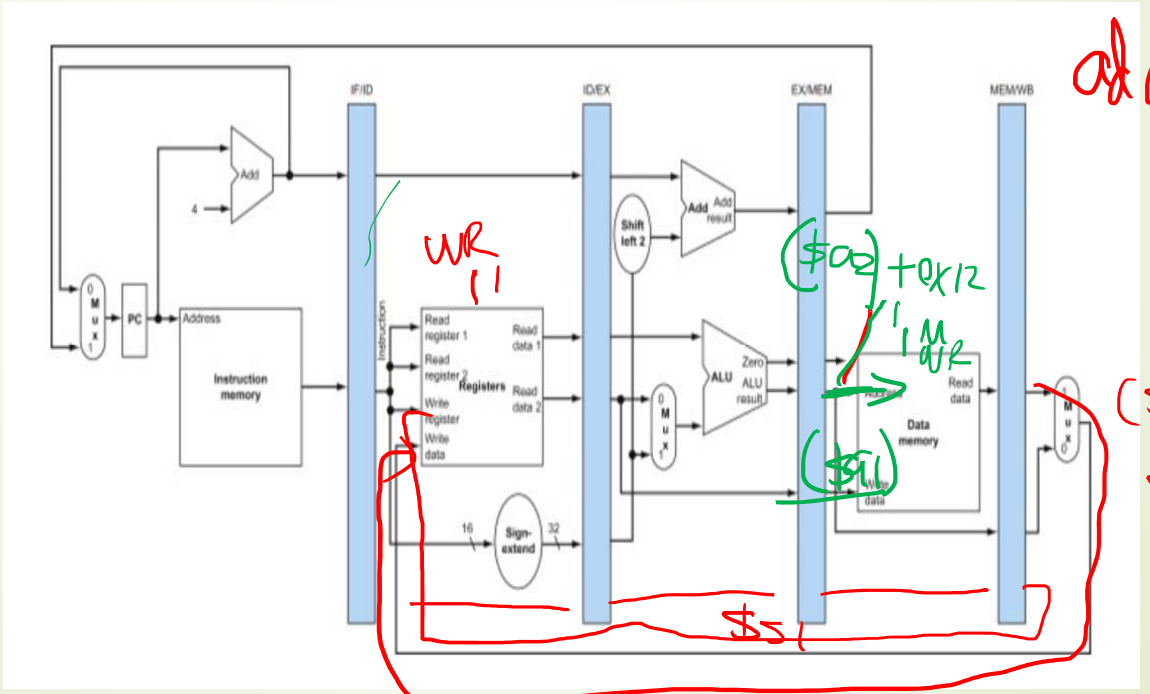
2004 sw $a1, 0x12 ($a2)




IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
M u x
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
Data memory
Read data
Write data
16
Sign-extend
32


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


IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
M u x
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
Data memory
Read data
Write data
16
Sign-extend
32



sw $q1, 0x12($q2)

add $s1, $q1, $q2

WR 1

($q2) + 0x12

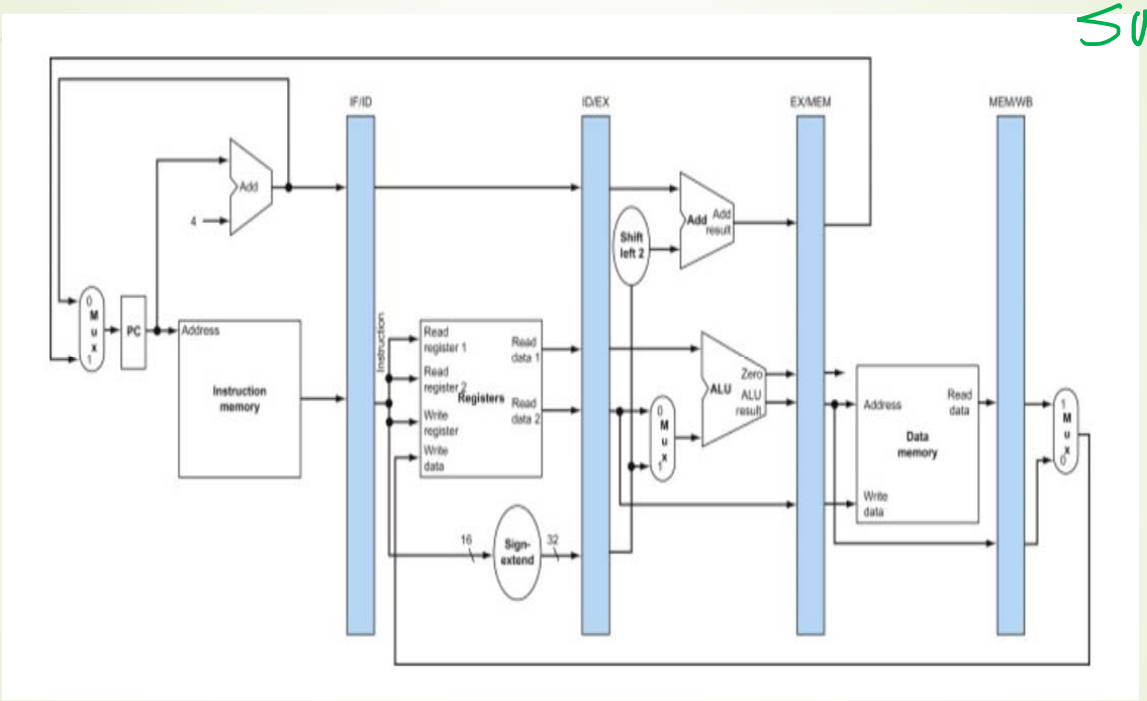
1, MDR

($q1)

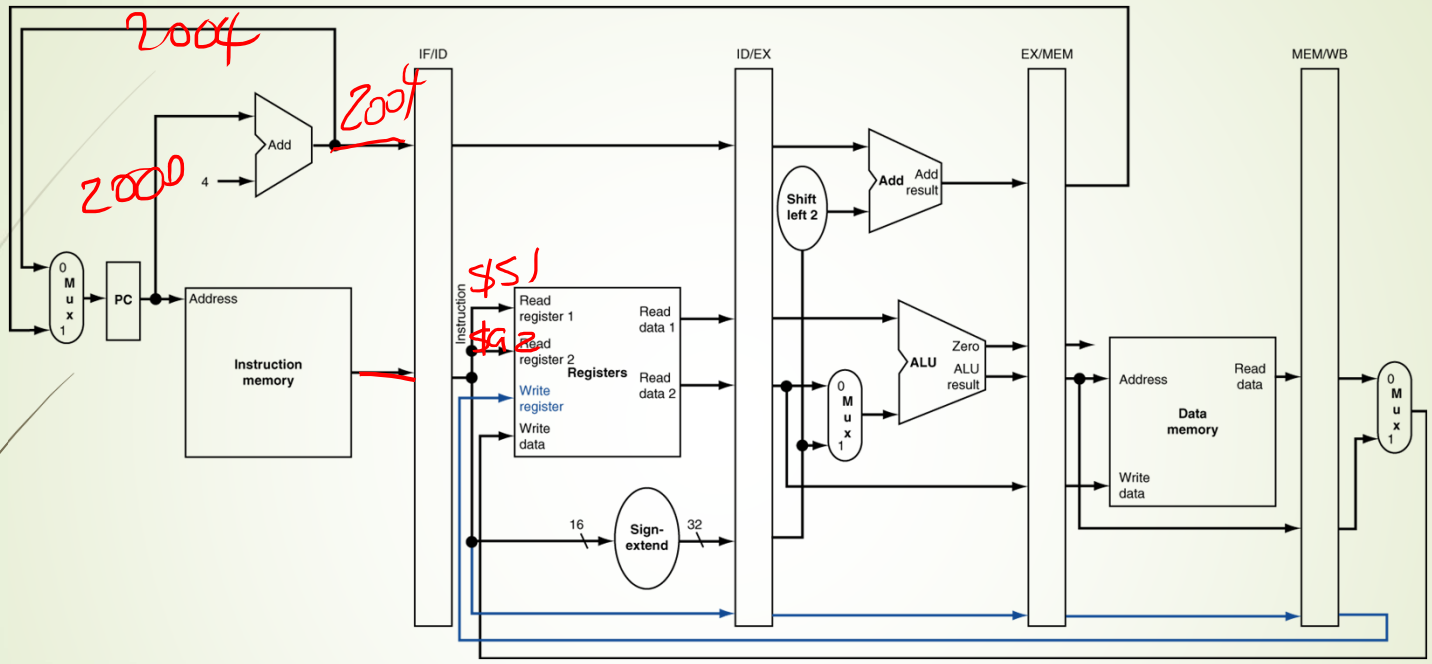
($q1) + ($q2)

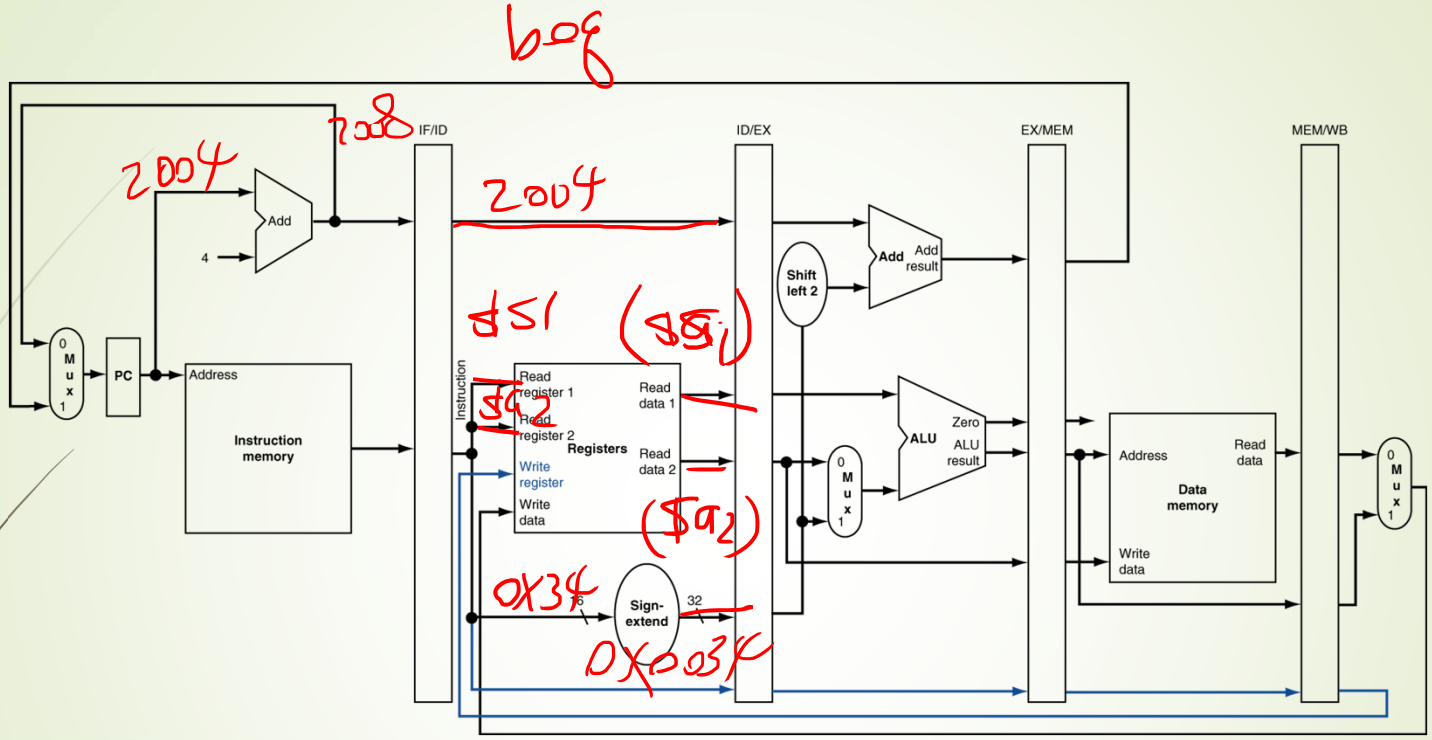
$s1

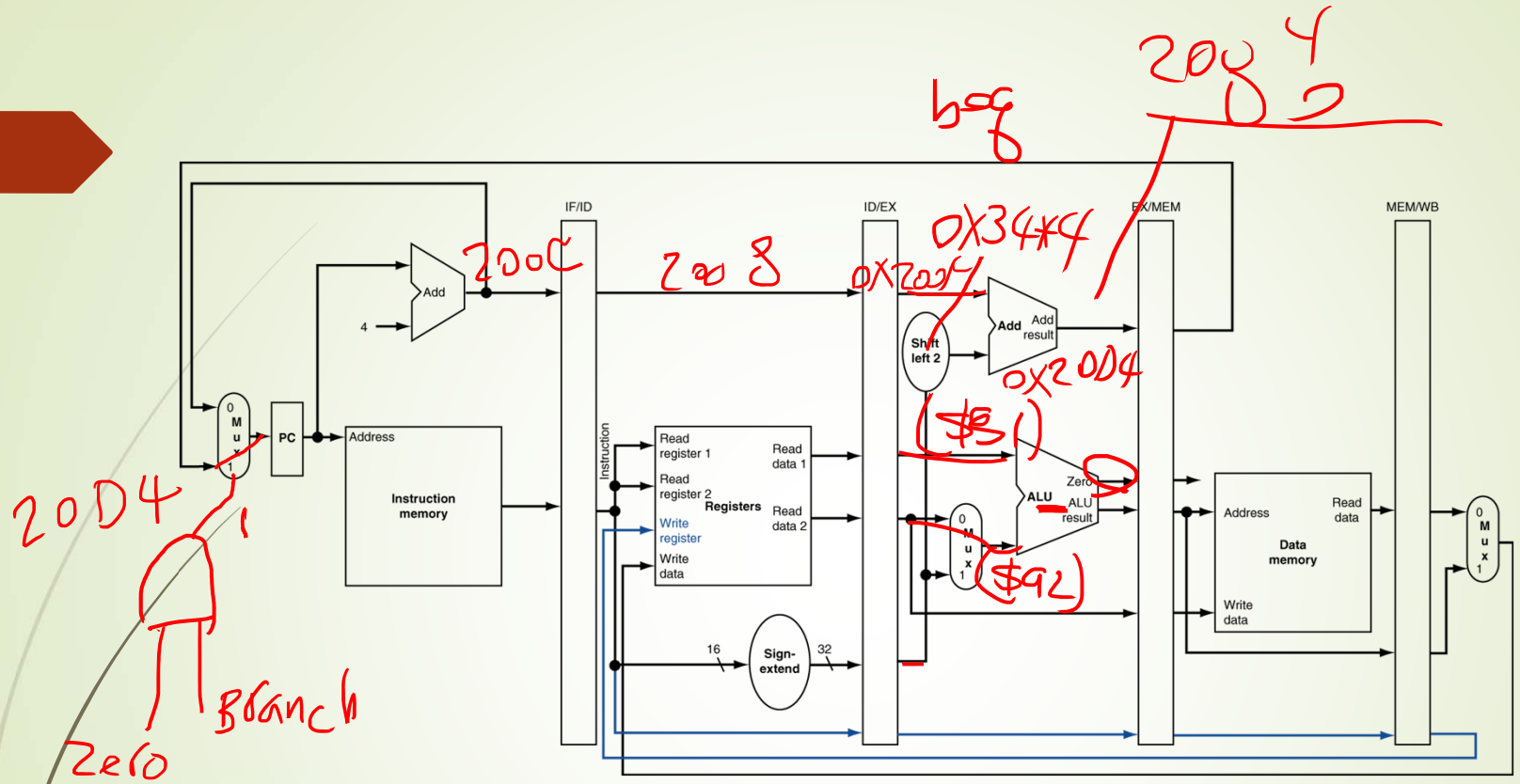
sw $a1, 0x12($a2)


IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
M u x
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

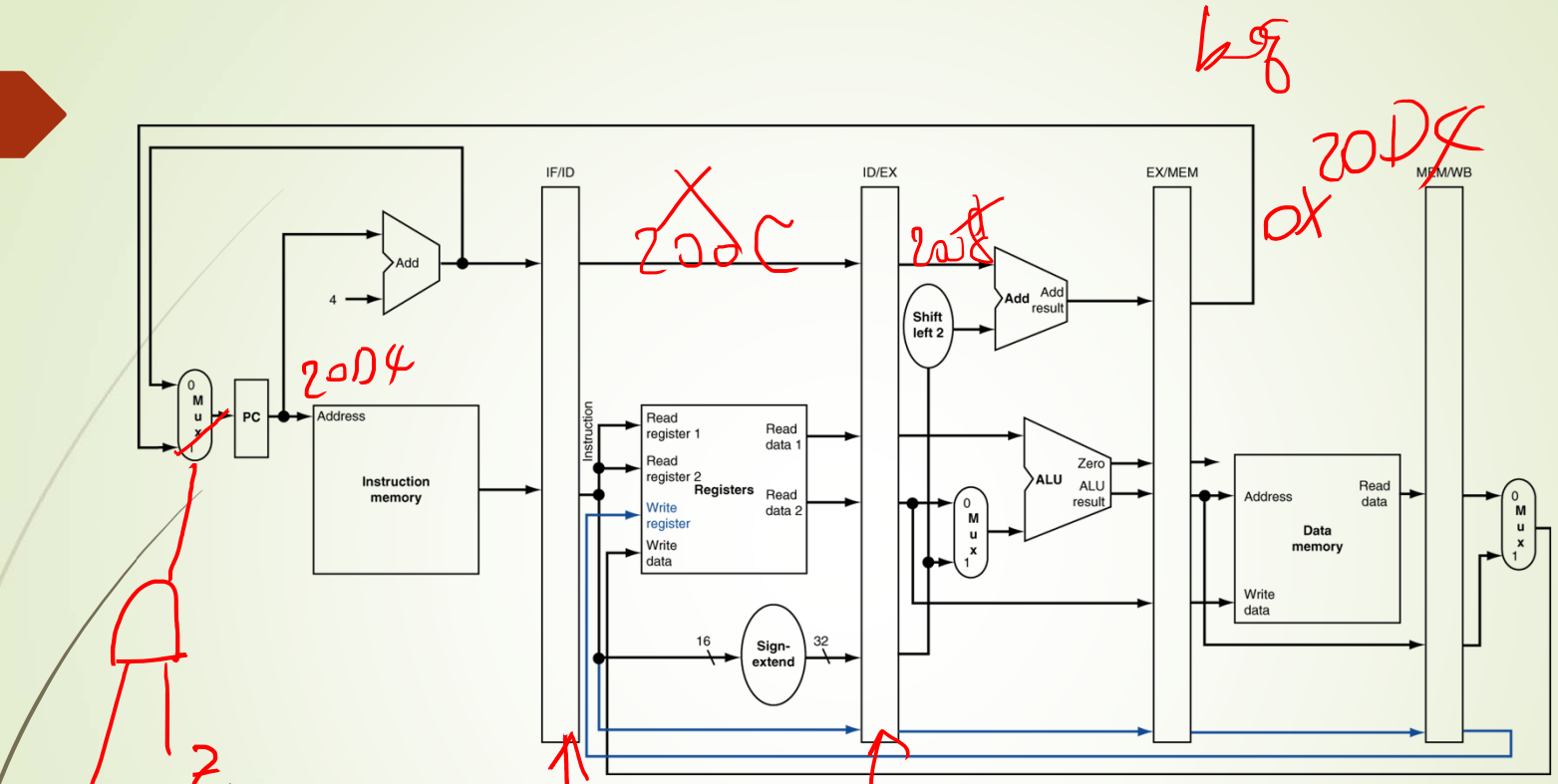
beq $s1, $a2, 0x34

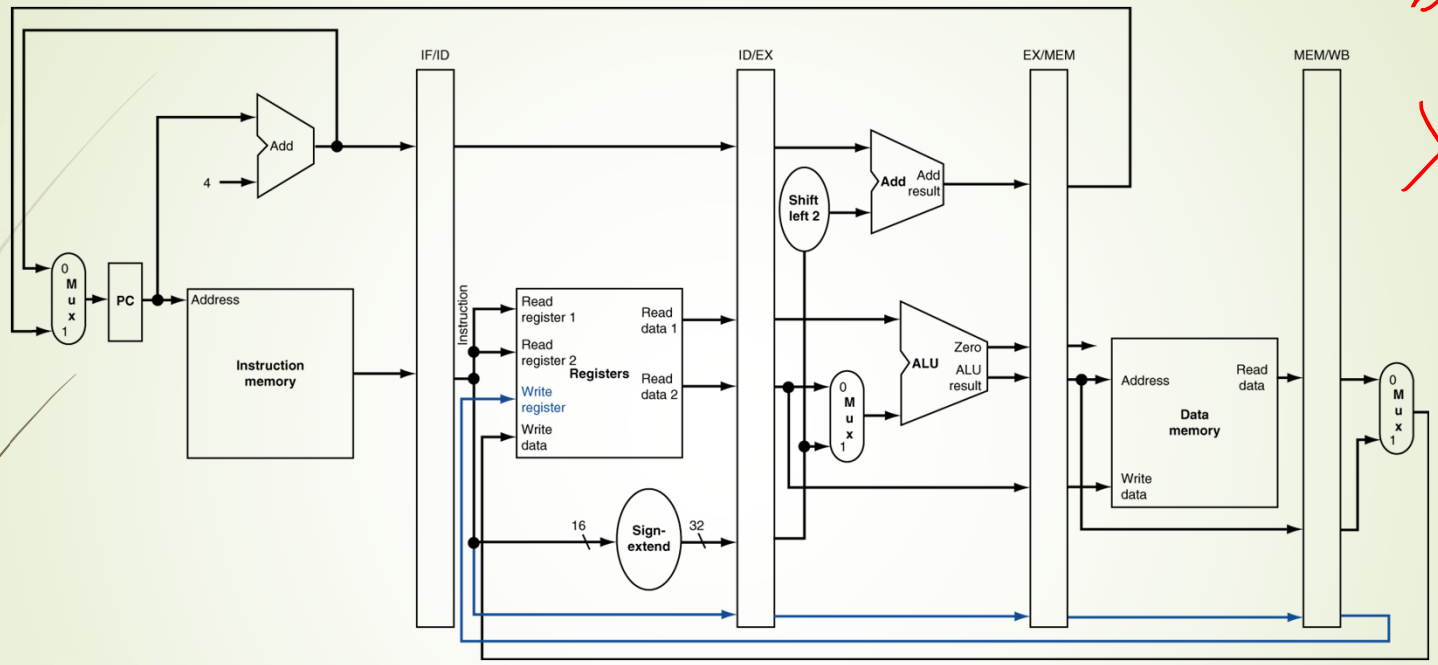



IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
M u x
0
1
PC
Address
Instruction memory
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 2
Add
Add result
ALU
Zero
ALU result
Address
Read data
Data memory
Write data
Sign-extend
16
32

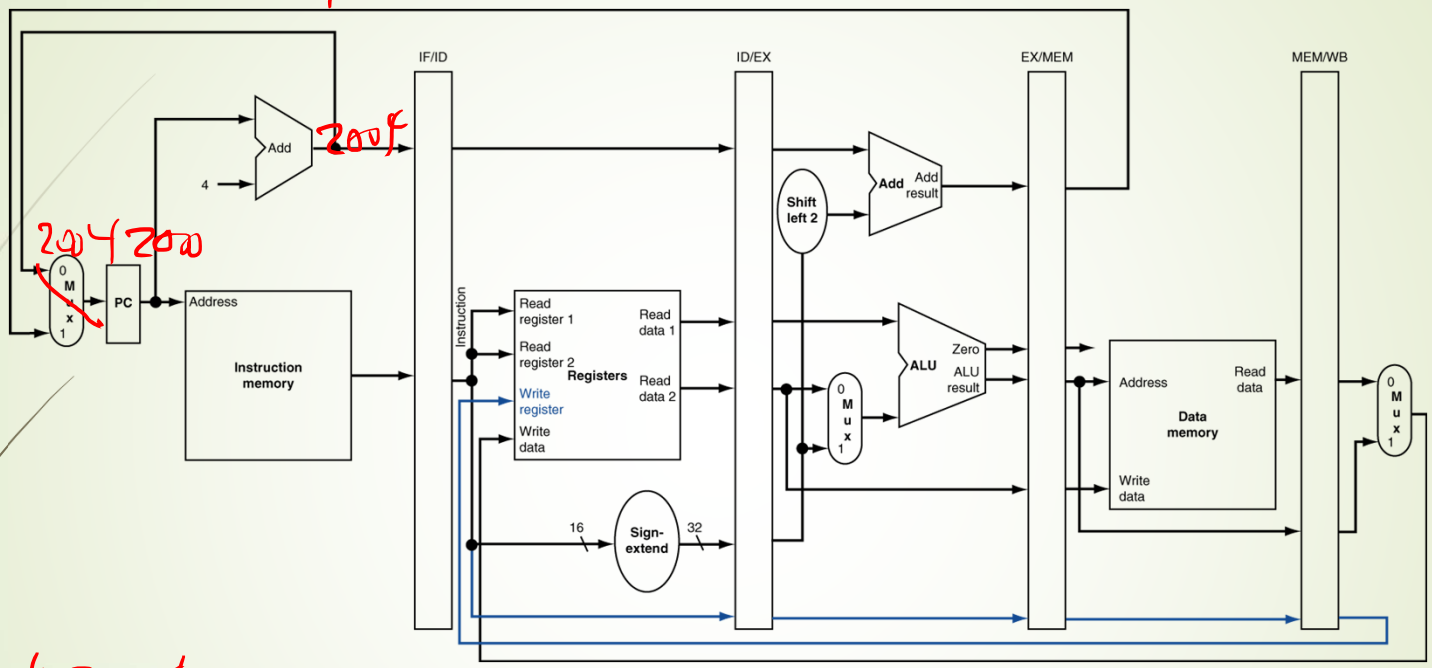

IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
PC
Address
Instruction memory
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Sign-extend
16
32
Shift left 2
Add
Add result
ALU
Zero
ALU result
Address
Data memory
Read data
Write data
0
M
u
x
1

200C
2008
0X2004
0X34X4
0X2004
($s1)
($s2)
2004
1
Branch
Zero
2011 0100 0000
D
0


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
Data memory
Read data
Write data
16
Sign-extend
32


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

add $a1, $a2, $t1



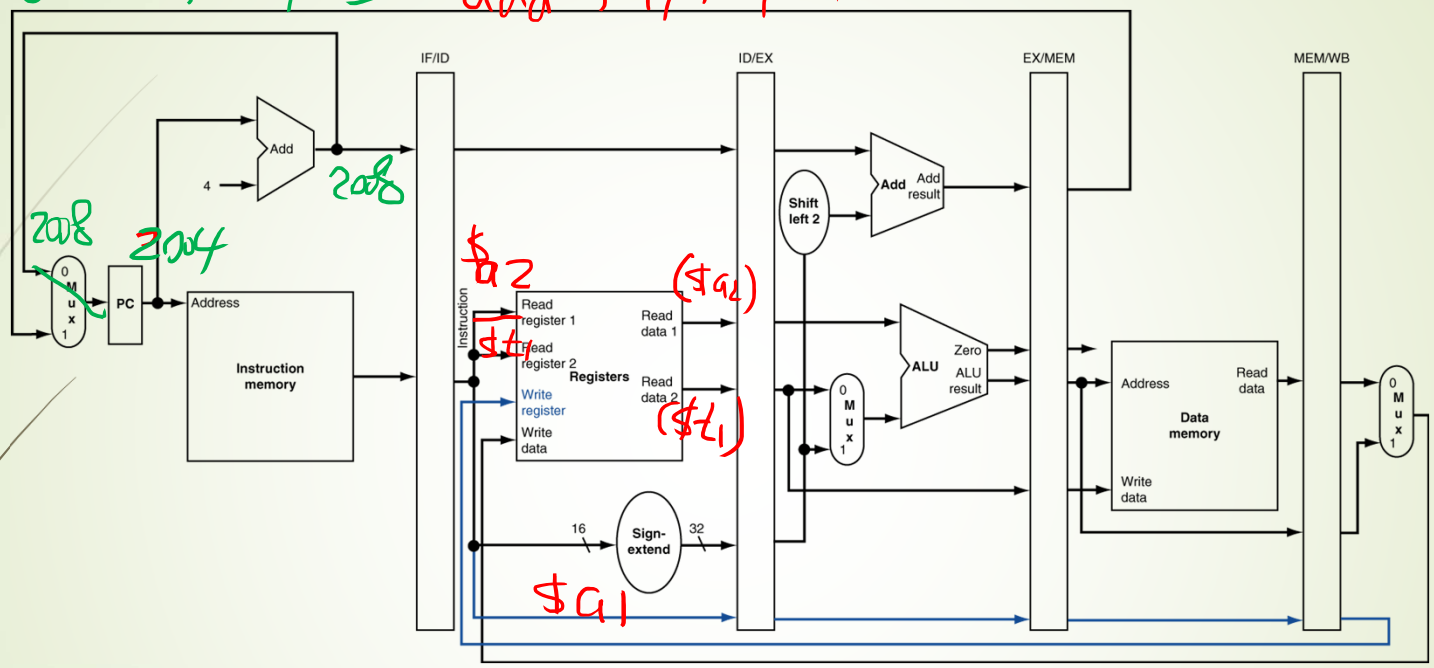
2000 add $a1, $a2, $t1

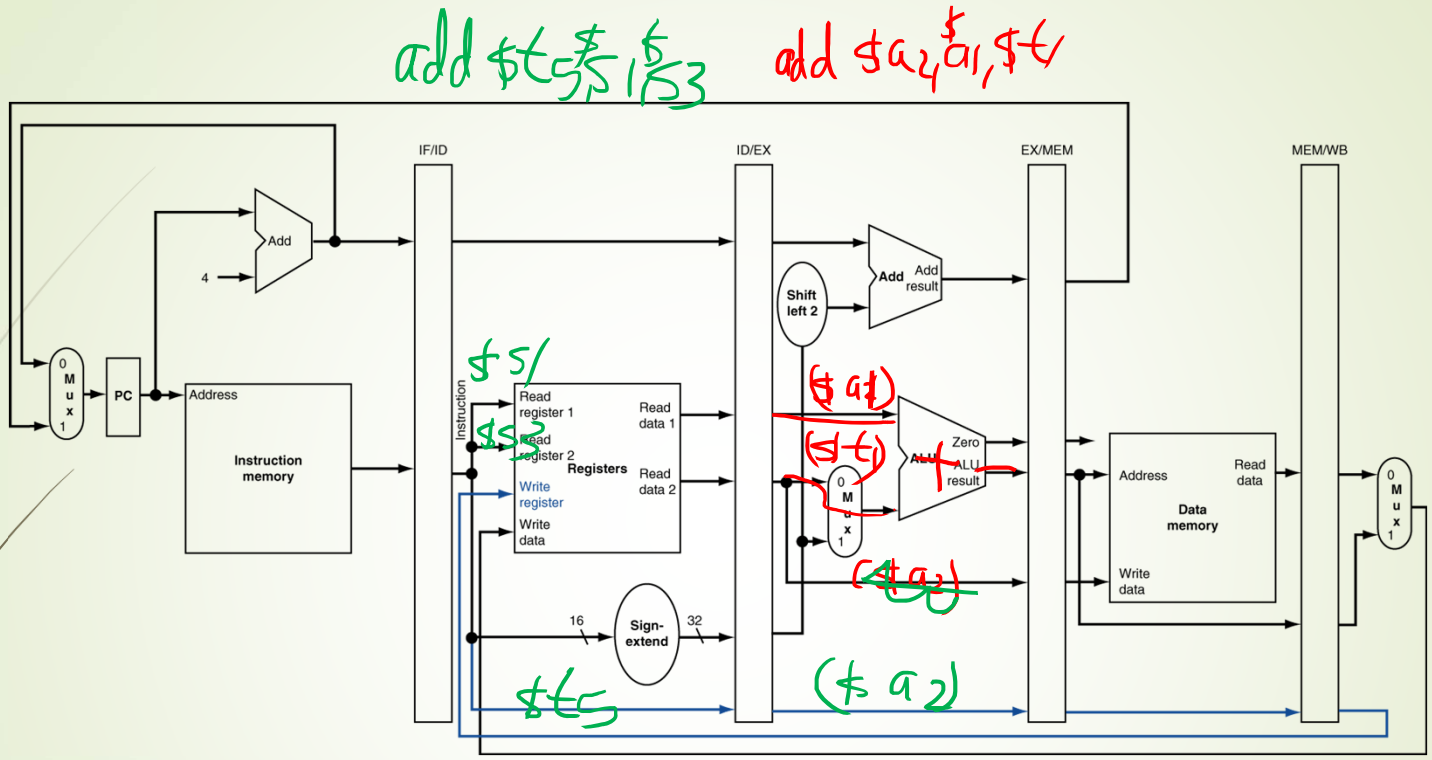
2004 add $t5, $s1, $s3

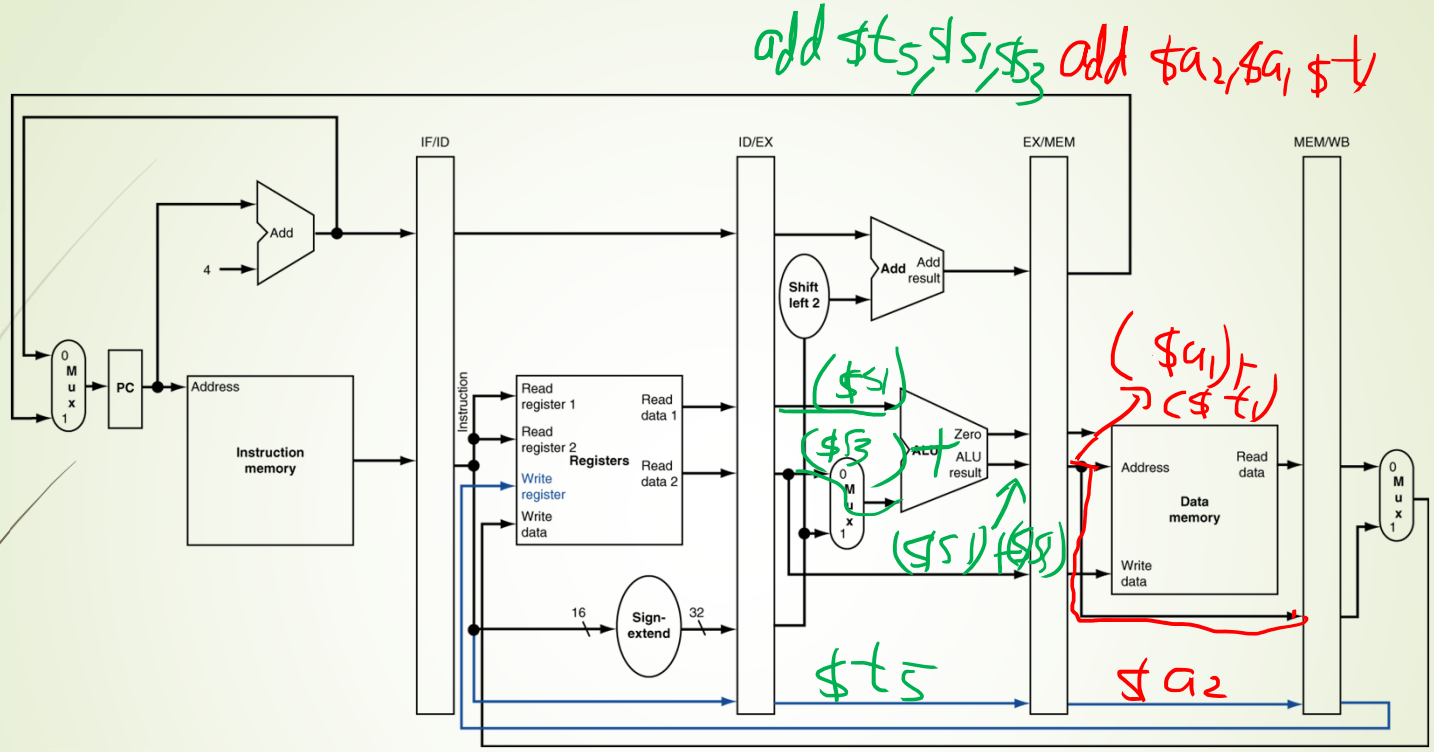
2008 sub $s5, $a4, $t3

add $t5, $s1, $s3

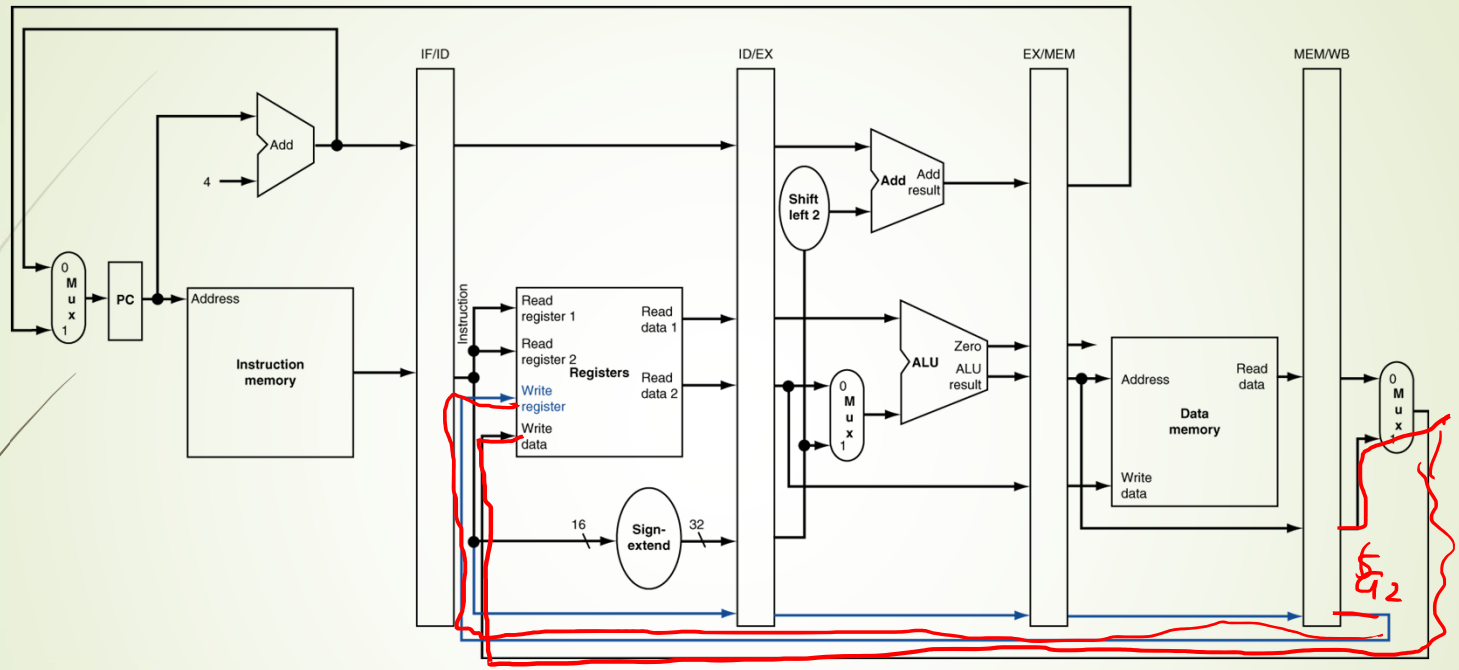
add $a1, $a2, $t1




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


IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
M u x
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
Zero
ALU
ALU result
Address
Data memory
Read data
Write data
16
Sign-extend
32

add $a2, $a1, $t4



($a2) + ($t4)

$a2