COMP3211/COMP9211 Computer Architecture

Lab 3 Pipelined Processor

The lab is the extension of your work of Lab 2.

Goals

1. Build a simple pipelined processor.
2. Learn how to model, debug, and evaluate your design.

Tasks:

Task 1: (20 marks)

Based on the single-cycle processor you built in Lab 2, build a pipelined processor model and the pipeline can also handle data hazards for the conditional branch instruction BNE.

Task 2: (20 marks)

Verify the function of your model and determine the minimal clock cycle time of the pipeline.

For each task, you need also to design how to effectively present your work during the peer assessment. For example, your presentation for Task 1 may focus on techniques of building the pipeline model; while for Task 2, your presentation may focus on the strategy of verification and results.

Due Time: your TLB class in Week 7.

Peer Assessment Scheme:

Your work on the lab tasks will be assessed by your peer students and tutor. Similar for Labs 1 & 2, for this lab

- Your TLB class is randomly divided into (up to) two assessement groups, each capped at 12 students. This is to allow sufficient time for the assessment. For a small TLB class, only one assessment group may be required.

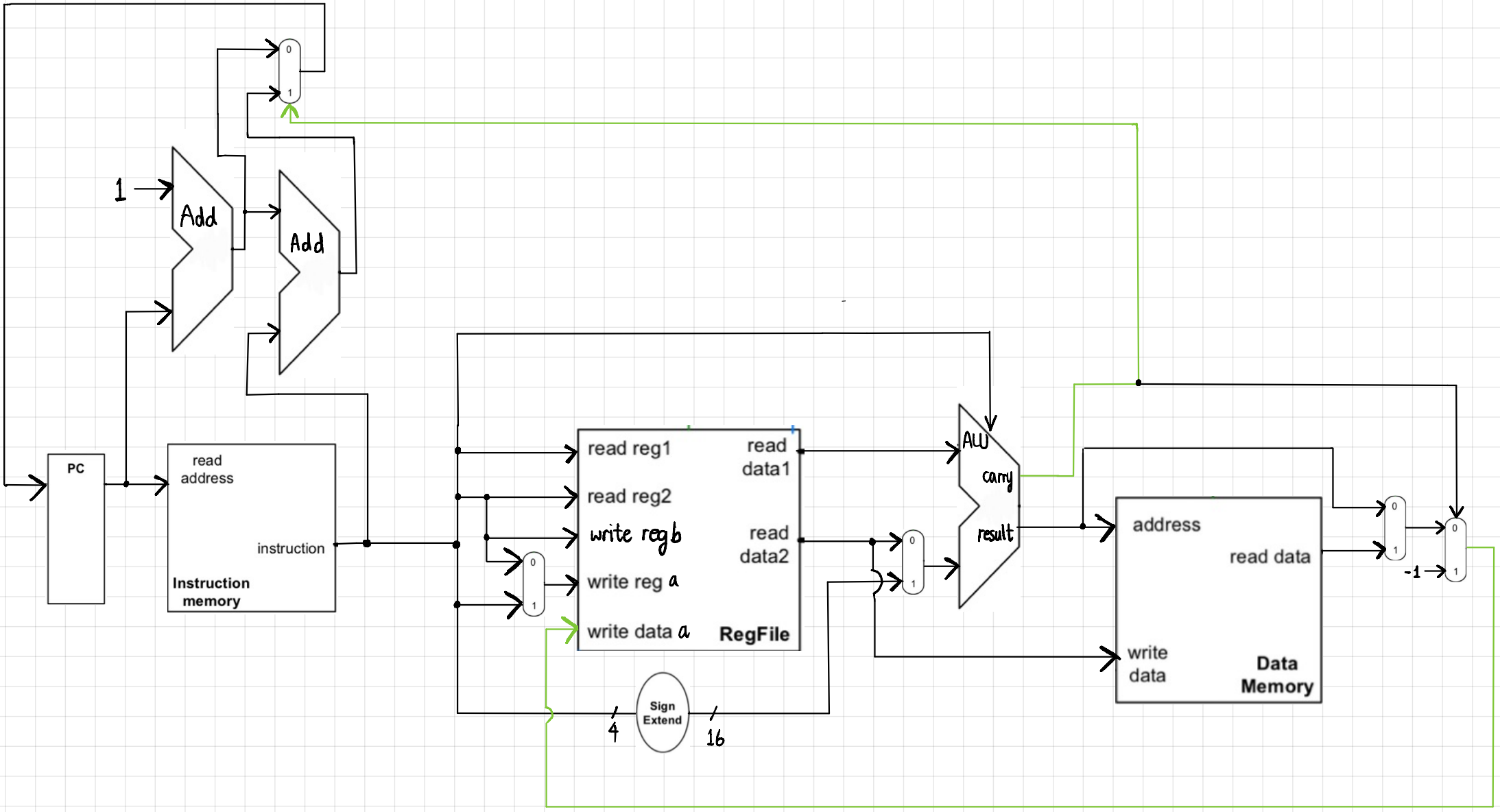
For the f2f classes, the assessement will be conducted in the on-campus classroom and for the online class, the assessment will be conducted in a MS Teams meeting.
The arrangment for each assessment group will be available in the Files folder of your class Teams channel.

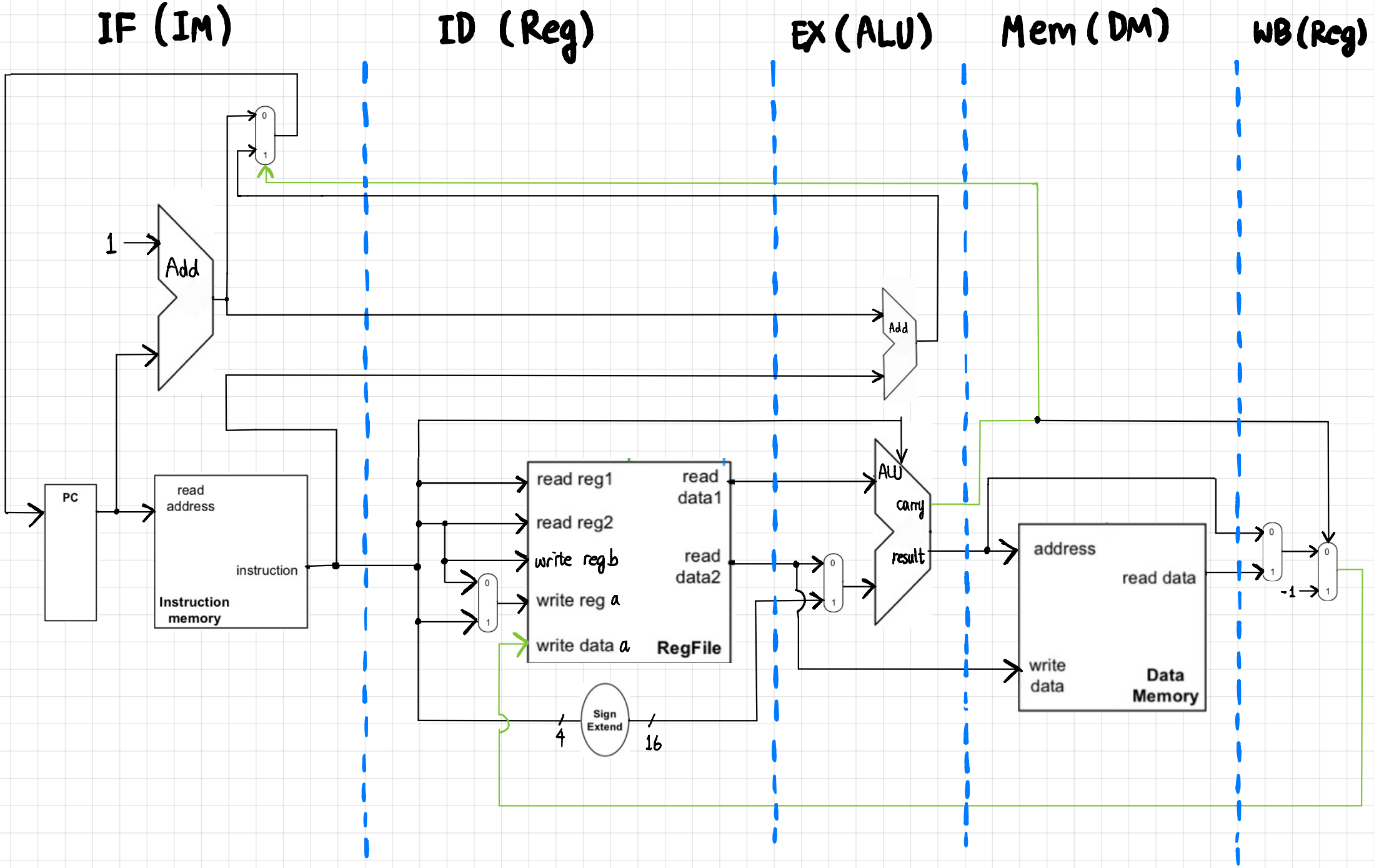
- During assessment, each student is given 6 mins for presentation and 2 mins for Q&A.
    - The presentation covers three areas: design idea, HDL model, simulation results, and any discussion required for each task.
- Your work is assessed based on three categories (Presentation, Completion, and Quality). Among the three categories, Completion is rated in 0-7 and the other two each are rated in 0-5. Optionally you can add comments (your comments will be anonymously available individual assessees). See the guideline below:
    - Presentation (0-5)
        - Clarity (2)
        - Logic (2)
        - Timing management (1)
    - Completion (0-7)
        - For Task 1
            - Design (3)
            - HDL model (4)
        - For Task 2
            - Validation (5)
            - Discussion (2)
    - Quality (0-5)
        - You can give marks based on different aspects demonstrated in the lab work, for example, an interesting design idea, a clever modeling strategy, good design analysis, thorough evaluation, and adequate discussion.

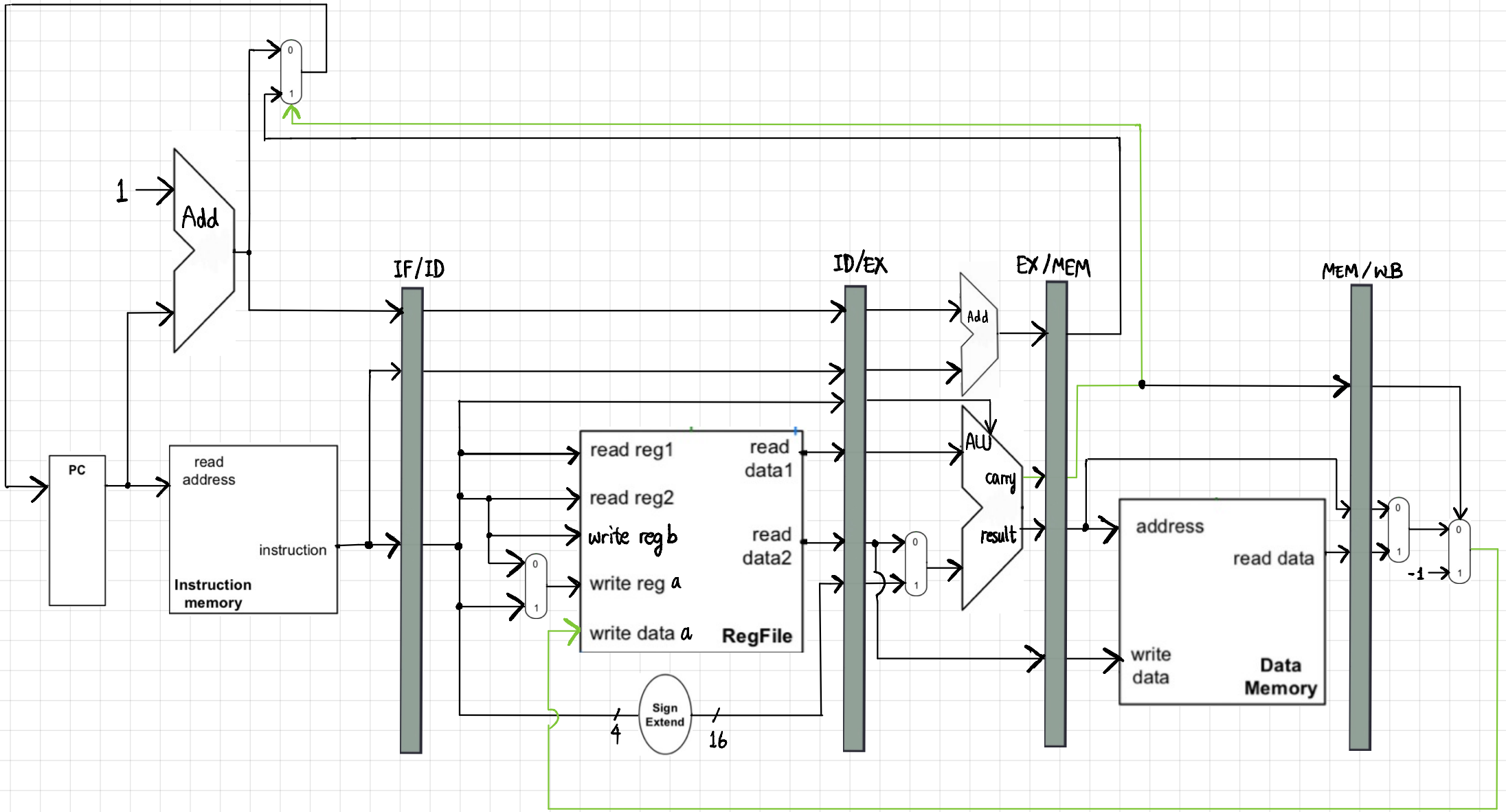
The link of an assessment form will be available in your MS teams channel.

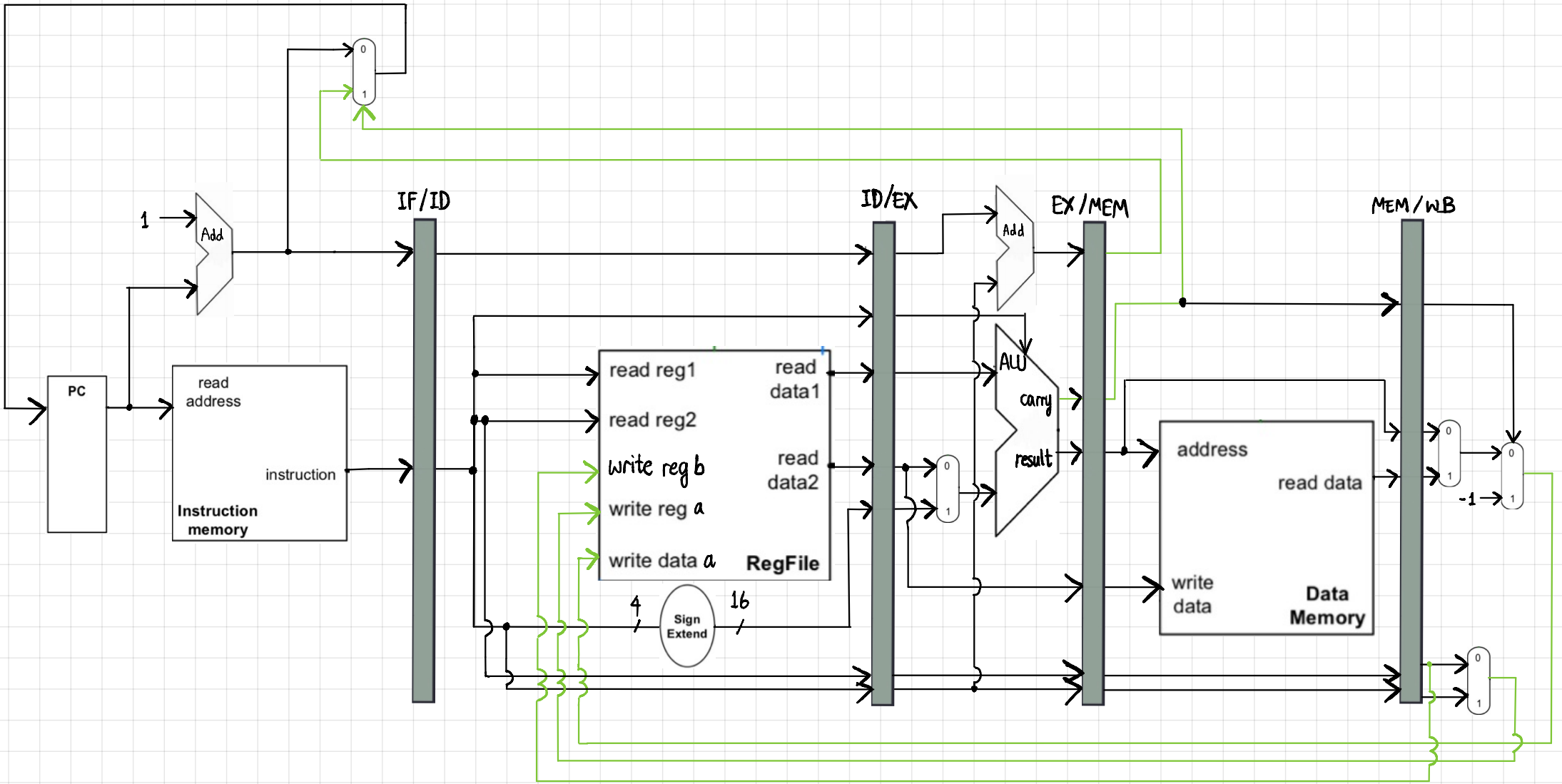
- Your tutor will organize and steer the lab session.
  By the end of the lab class, all students are required to submit their assessment forms. Your participation to the assessment will be taken into account to the overall lab participation marks.

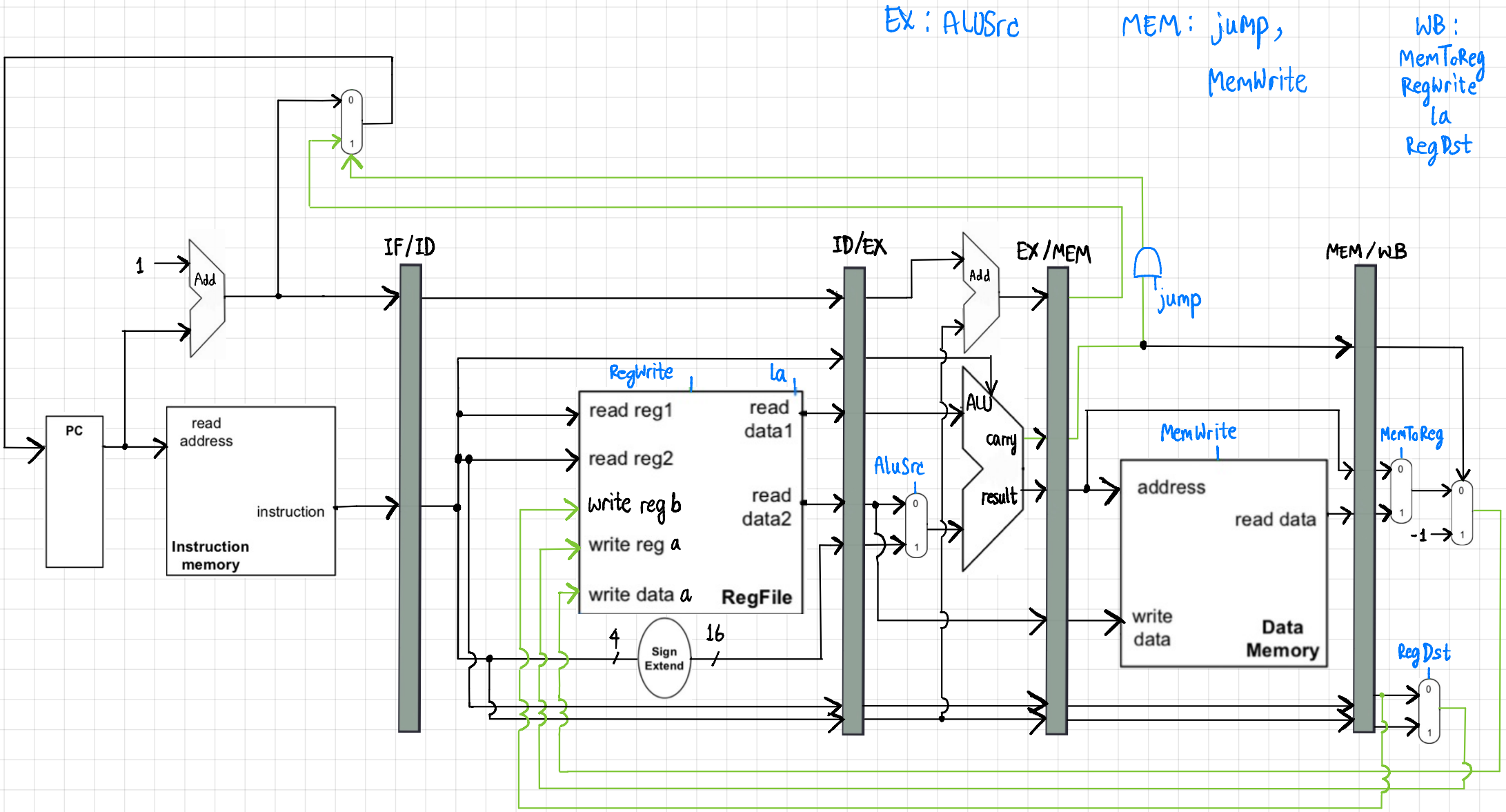
- Your marks are determined by
    - 80% from peers, and
    - 20% from tutor

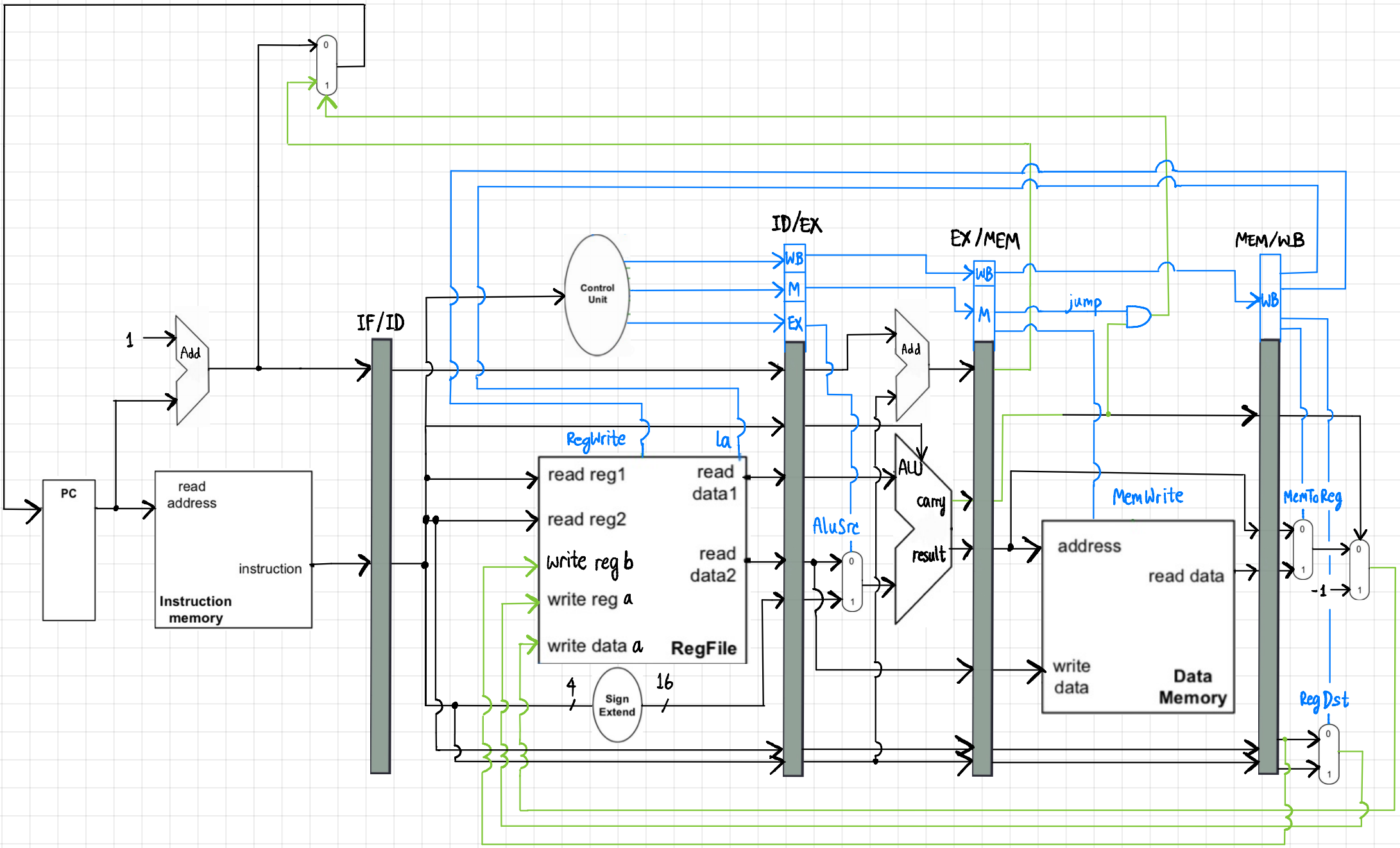

0
1
1
Add
Add
PC
read
address
instruction
Instruction
memory
read reg1
read reg2
write reg b
write reg a
write data a
read
data1
read
data2
RegFile
0
1
ALU
carry
result
0
1
address
read data
write
data
Data
Memory
0
1
-1
0
1
Sign
Extend
4
16

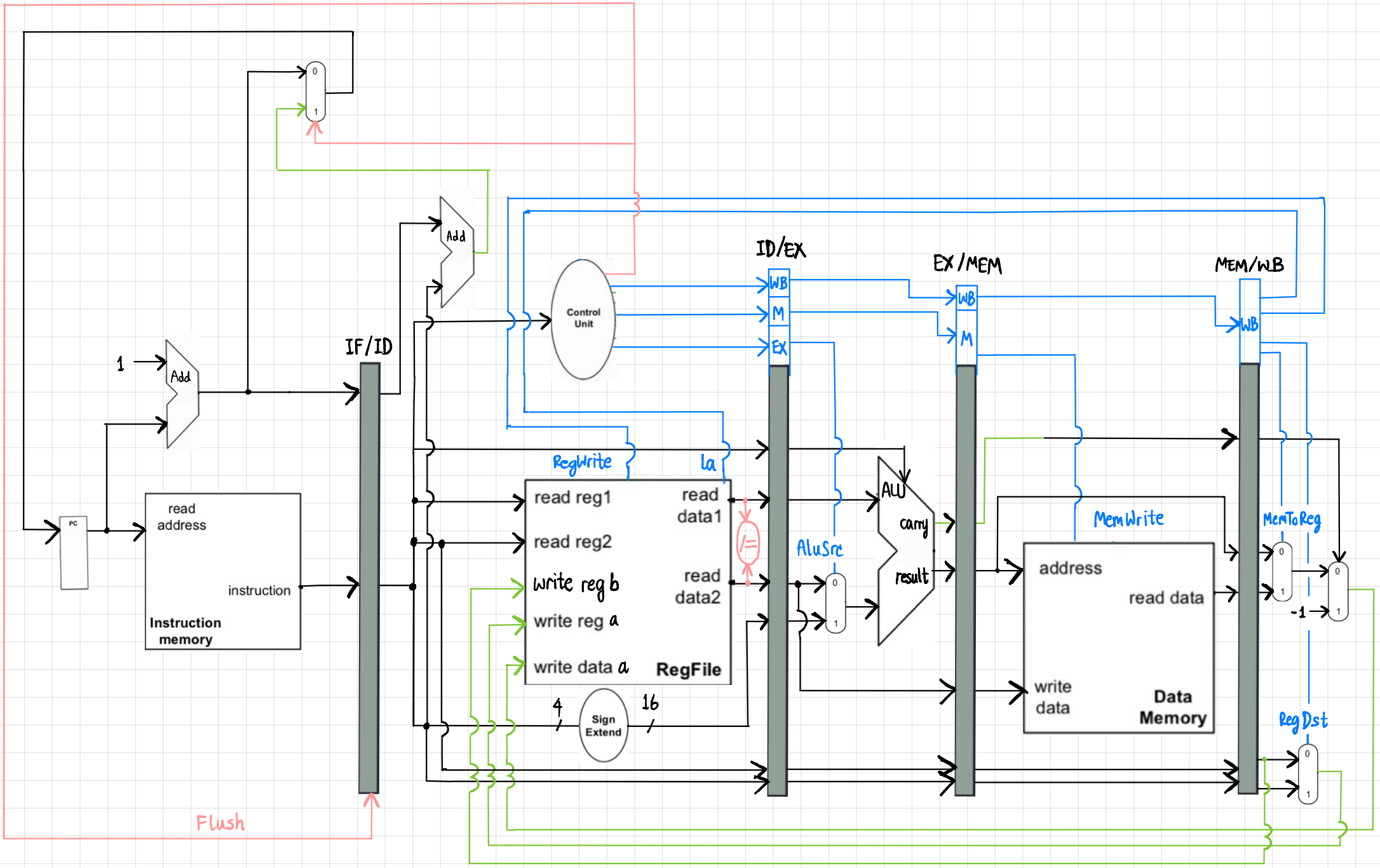

IF (IM)
ID (Reg)
EX (ALU)
Mem (DM)
WB (Reg)
1
Add
Add
PC
read address
instruction
Instruction memory
read reg1
read reg2
write reg b
write reg a
write data a
read data1
read data2
RegFile
Sign Extend
4
16
ALU
carry
result
address
read data
write data
Data Memory
-1

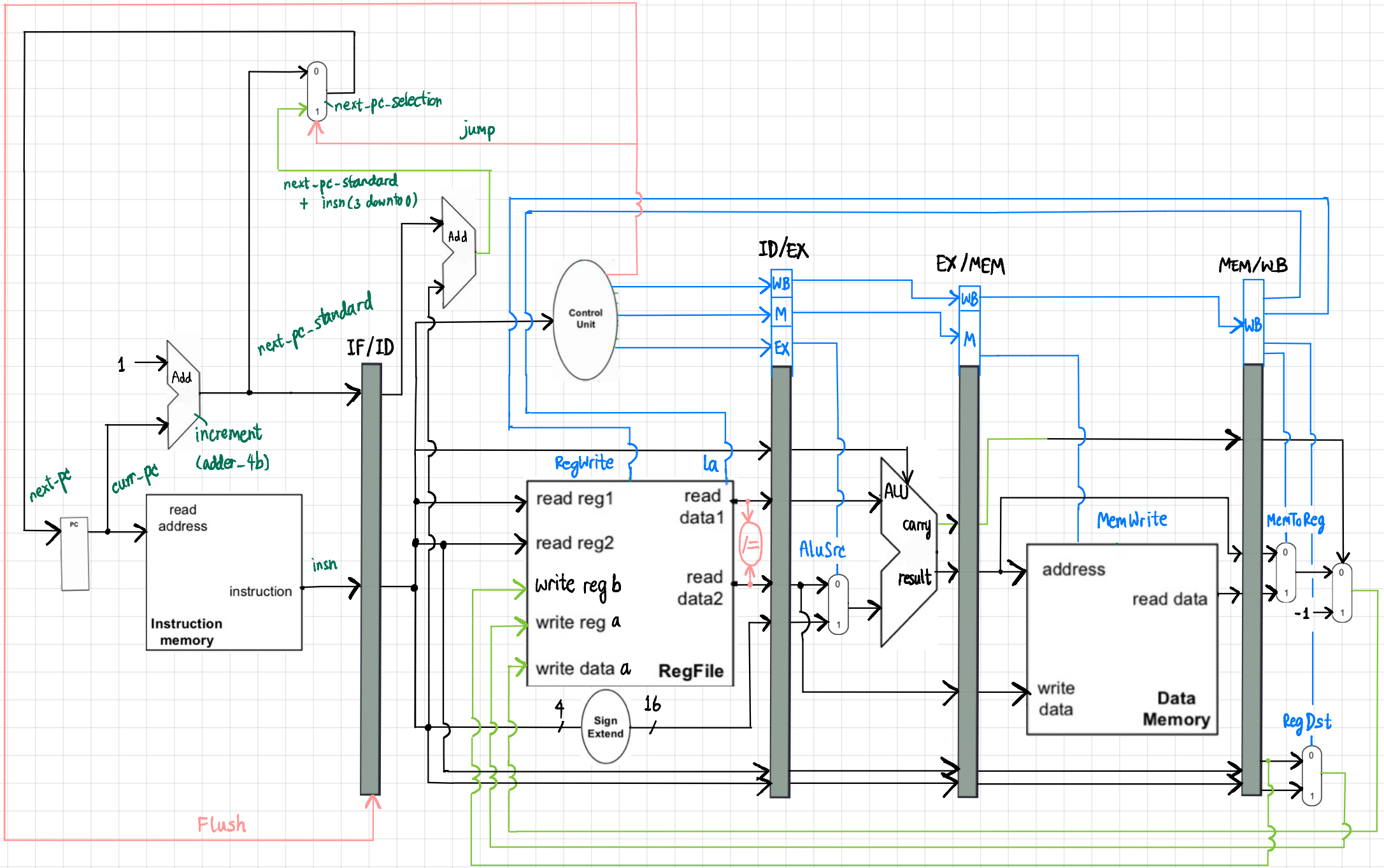

0
1
1
Add
IF/ID
ID/EX
EX/MEM
MEM/WB
Add
PC
read
address
instruction
Instruction
memory
read reg1
read reg2
write reg b
write reg a
write data a
RegFile
read
data1
read
data2
0
1
ALU
carry
result
address
read data
write
data
Data
Memory
Sign
Extend
4
16
-1

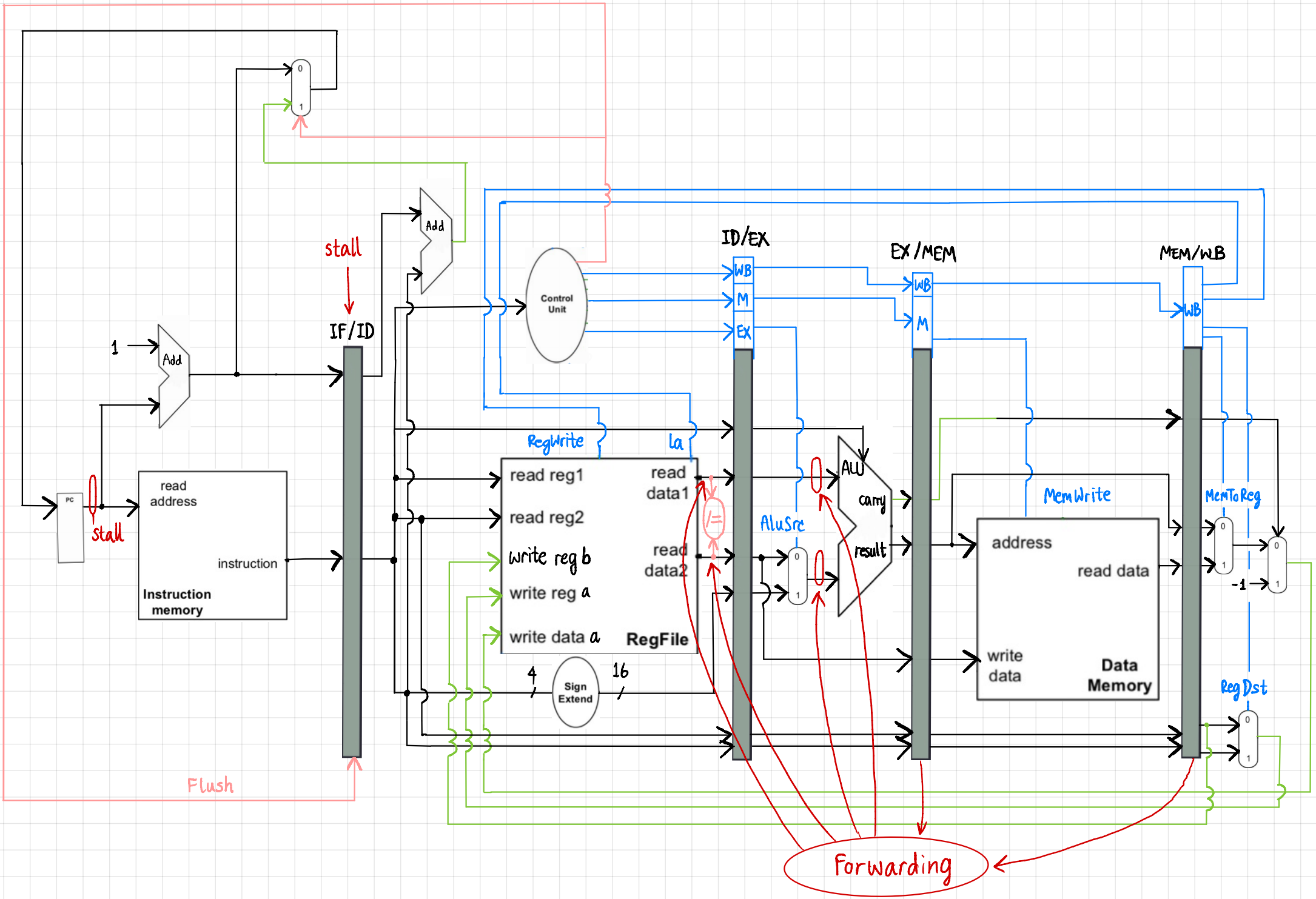

IF/ID
ID/EX
EX/MEM
MEM/WB
Add
1
PC
read address
instruction
Instruction memory
read reg1
read reg2
write reg b
write reg a
write data a
read data1
read data2
RegFile
4
Sign Extend
16
ALU
carry
result
address
read data
write data
Data Memory
-1
0
1

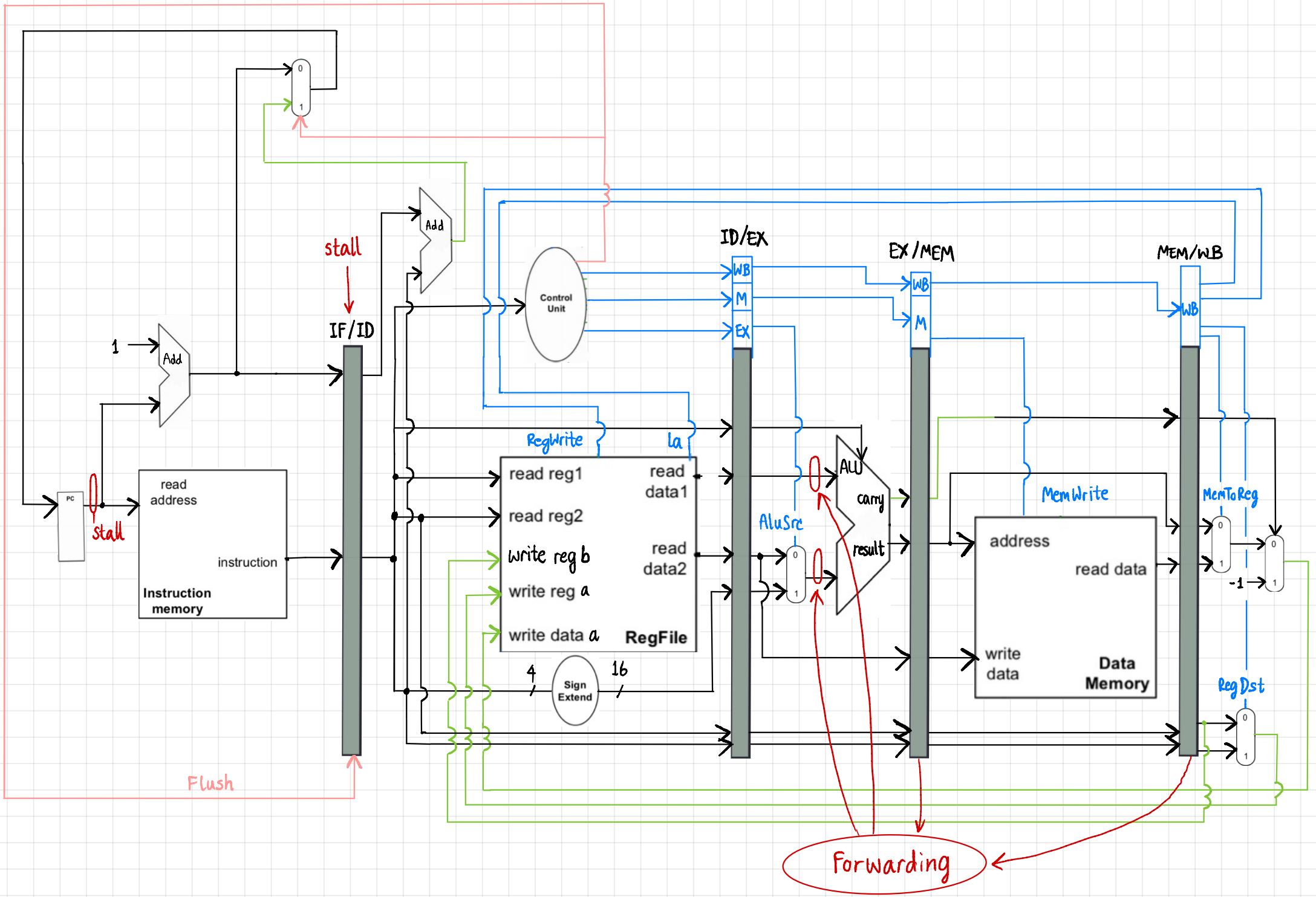

EX : ALUSrc
MEM : jump, MemWrite
WB : MemToReg RegWrite la RegDst
IF/ID
ID/EX
EX/MEM
MEM/WB
1
Add
PC
read address
instruction
Instruction memory
RegWrite
la
read reg1
read reg2
write reg b
write reg a
write data a
read data1
read data2
RegFile
4
Sign Extend
16
AluSrc
ALU
carry
result
Add
jump
MemWrite
address
read data
write data
Data Memory
MemToReg
-1
RegDst

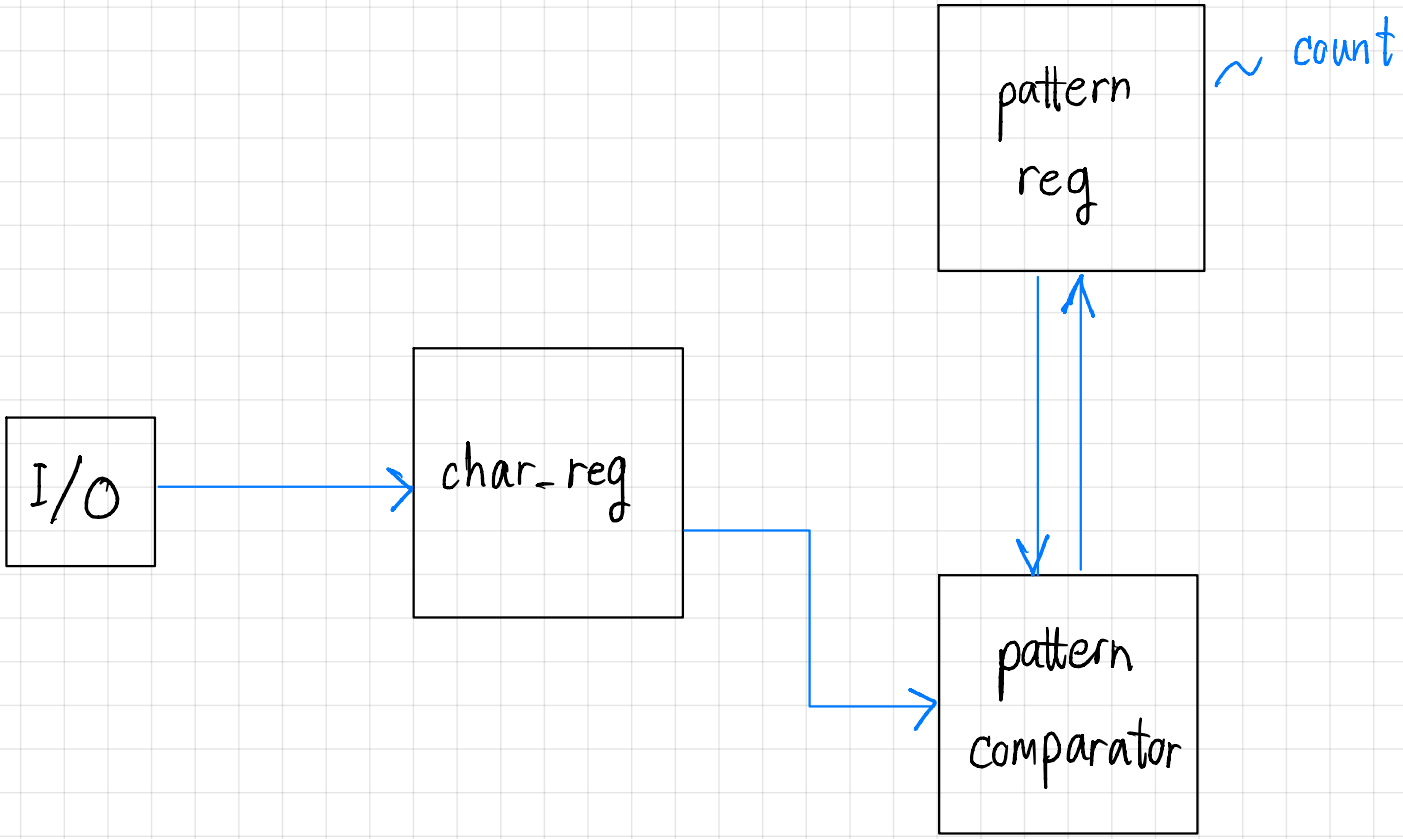

0
1
1
Add
PC
read
address
instruction
Instruction
memory
IF/ID
Control
Unit
ID/EX
WB
M
EX
EX/MEM
WB
M
MEM/WB
WB
jump
Add
RegWrite
la
read reg1
read reg2
write reg b
write reg a
write data a
read
data1
read
data2
RegFile
4
Sign
Extend
16
AluSrc
ALU
carry
result
MemWrite
address
read data
write
data
Data
Memory
MemToReg
-1
RegDst


0
1
Add
1
Add
IF/ID
Control Unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
WB
M
WB
PC
read address
instruction
Instruction memory
RegWrite
la
read reg1
read reg2
write reg b
write reg a
write data a
read data1
read data2
RegFile
=
AluSrc
ALU
carry
result
MemWrite
address
read data
write data
Data Memory
MemToReg
-1
RegDst
4
Sign Extend
16
Flush


next-pc-selection
jump
next-pc-standard
+ insn(3 downto 0)
Add
next-pc-standard
1
Add
increment
(adder_4b)
next-pc
curr-pc
PC
read
address
instruction
Instruction
memory
insn
IF/ID
Flush
Control
Unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
RegWrite
la
read reg1
read reg2
write reg b
write reg a
write data a
read
data1
read
data2
RegFile
4
Sign
Extend
16
/=
AluSrc
ALU
carry
result
MemWrite
address
read data
write
data
Data
Memory
MemToReg
-1
RegDst


0
1
Add
1
Add
stall
IF/ID
Control Unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
WB
M
WB
PC
stall
read address
instruction
Instruction memory
RegWrite
la
read reg1
read reg2
write reg b
write reg a
write data a
RegFile
read data1
read data2
/=
AluSrc
ALU
carry
result
MemWrite
MemToReg
address
read data
write data
Data Memory
-1
4
Sign Extend
16
RegDst
Flush
Forwarding


0
1
Add
stall
IF/ID
Add
1
Control Unit
ID/EX
EX/MEM
MEM/WB
WB
M
EX
WB
M
WB
PC
stall
read address
instruction
Instruction memory
RegWrite
la
read reg1
read reg2
write reg b
write reg a
write data a
read data1
read data2
RegFile
4
Sign Extend
16
AluSrc
ALU
carry
result
MemWrite
address
read data
write data
Data Memory
MemToReg
-1
RegDst
Flush
Forwarding


pattern
reg
count
I/O
char_reg
pattern
comparator