RISC V Pipeline

RISC V Pipeline

Five Stages of the Pipeline:

- Instruction Fetch
- Instruction Decode
- Execute
- Memory Access
- Writeback

Instruction Fetch

1. Take instruction from PC
2. **if** *non-branch statement*:
   PC -> PC + 4
   else:
   Stall for cycle
   Use updated PC

Instruction Decode

`'ADD' 'x2', 'x1', 'x3'`
During the decode stage, the **indexes of these two registers** are identified within the instruction, and the **indexes are presented to the register memory**, as the address. At the same time the register file is read, **instruction issue logic in this stage determines if the pipeline is ready to execute the instruction in this stage**.

`'BEQ' 'x1', 'x2', 'OFFSET'`
If the **instruction decoded is a branch or jump**, the **target address of the branch** or jump is computed in parallel with reading the register file. The **branch condition is computed in the following cycle** (after the register file is read), and if the branch is taken or if the instruction is a jump, the PC in the first stage is assigned the branch target, rather than the incremented PC that has been computed.
If branch results in true then `PC -> PC + OFFSET`
If branch results in false then `PC -> PC + 4`

Execute

CASE 1:

REG - REG Operations:

If we use Instructions that make use of only the regs then *we simply used the values read from decoded registers and perform operation and get output by the end of the stage.*

CASE 2:

MEM Ref:

If we use instructions that use memory then *we simply add the value of the reg to offset both read in ID stage*.

At the end of this stage we get **OUTPUT** of the operation.

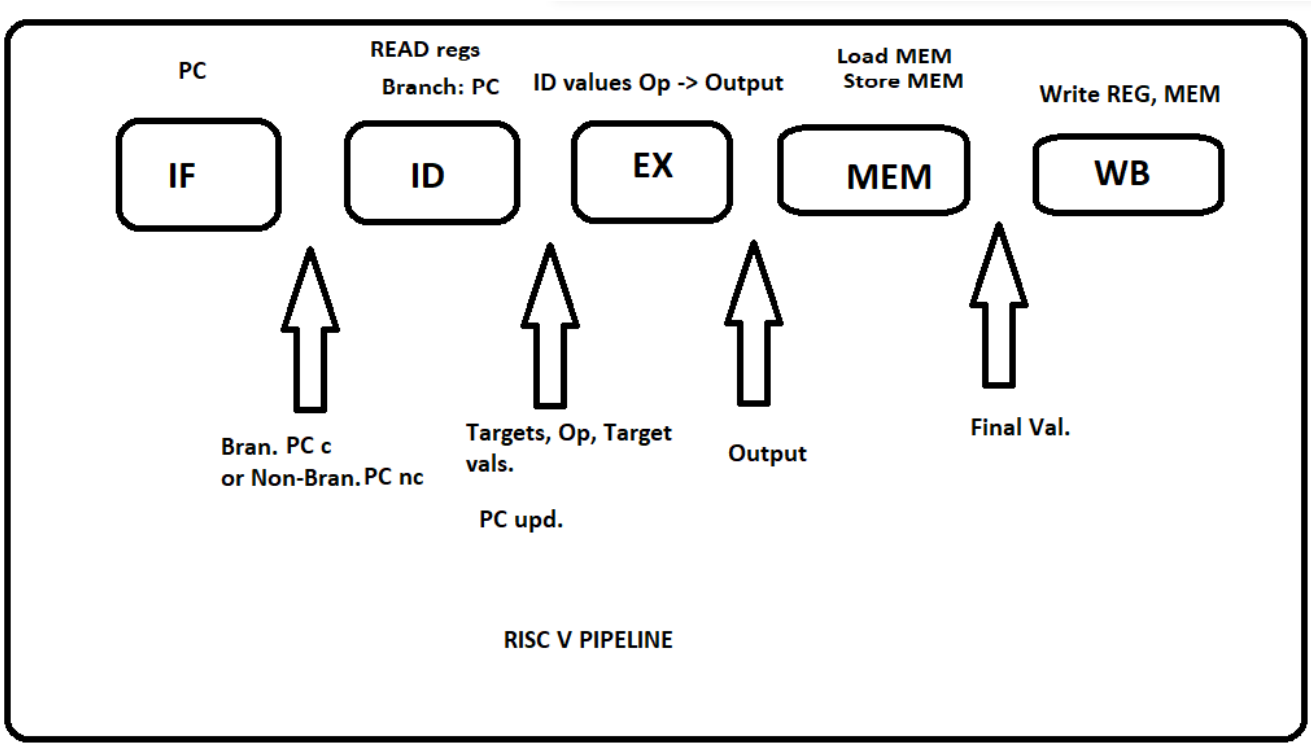
Memory Access:

In this stage we simply get values from memory if any instruction needs it.

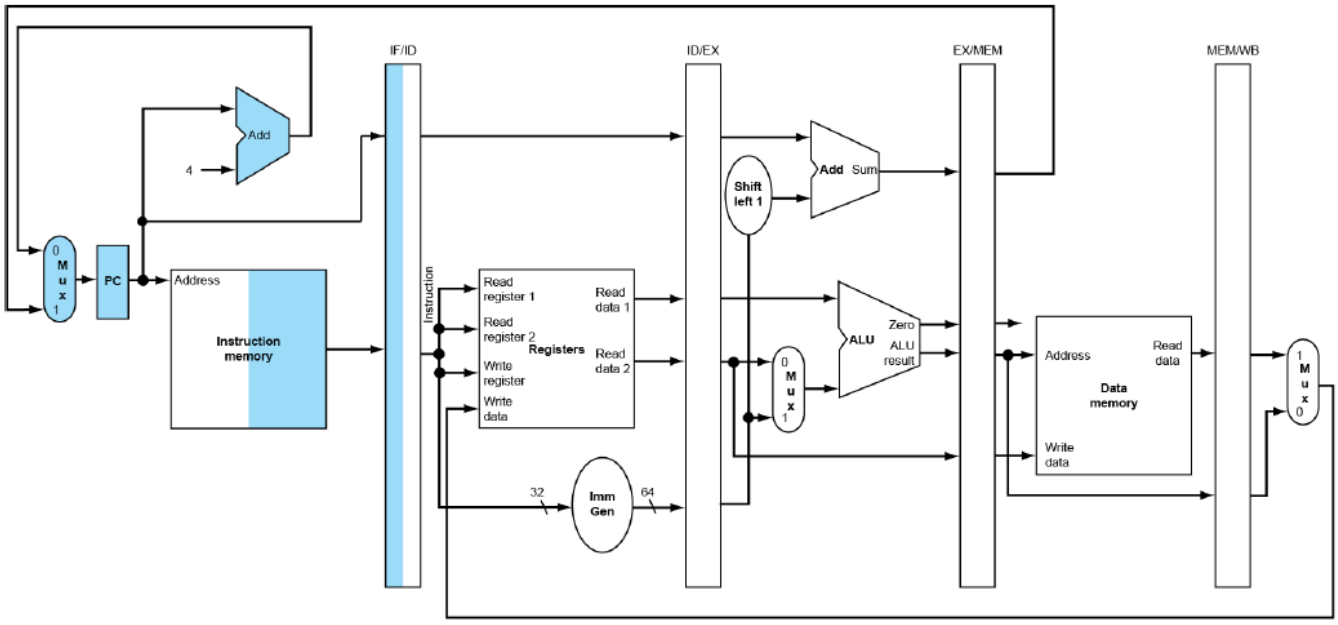
Writeback:

In this stage we simply write back the value to **memory** or **register**.

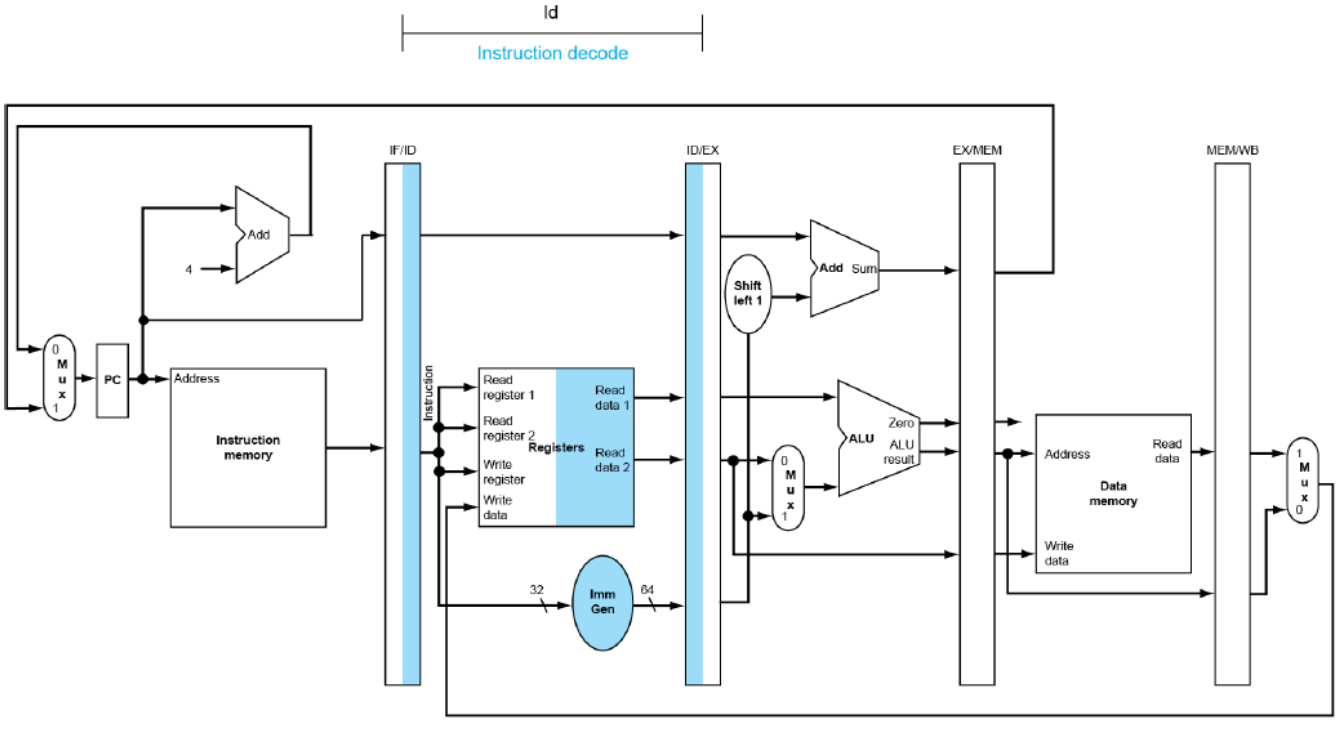
RISC-V PIPELINE:



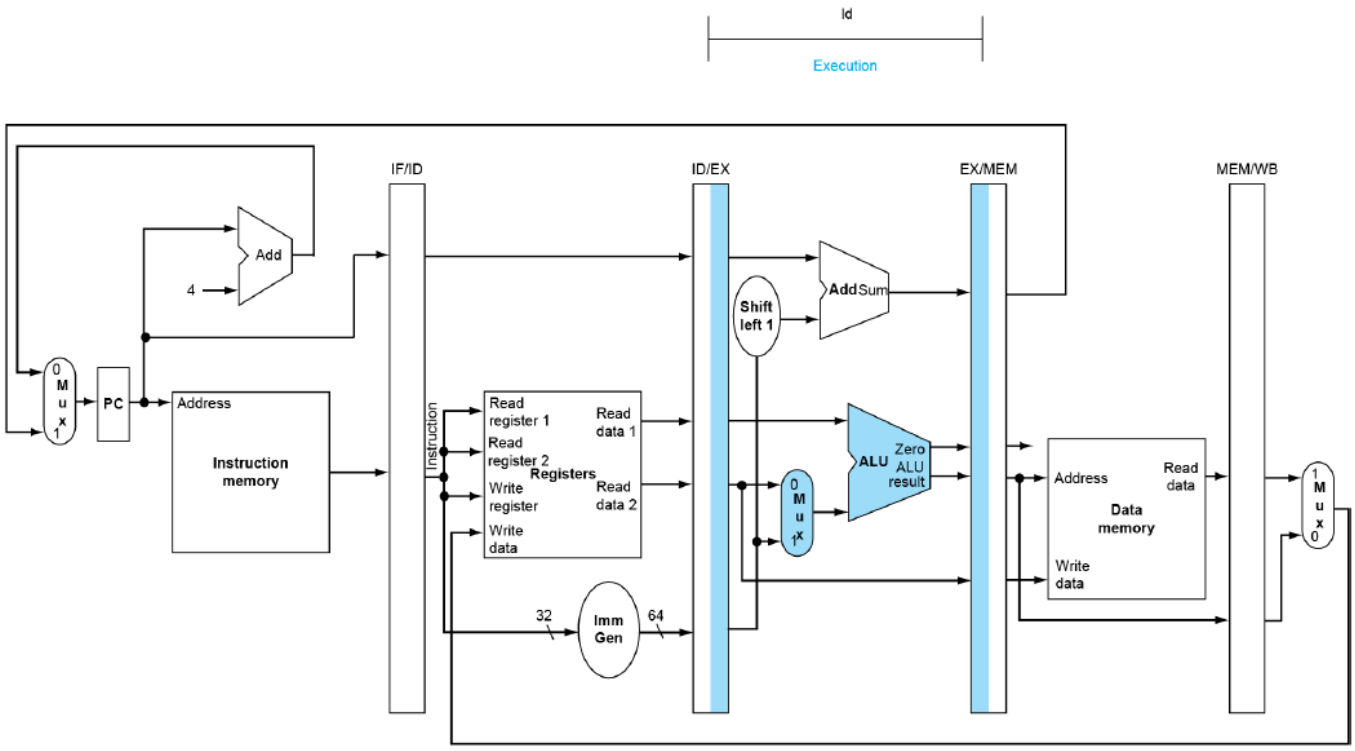
Instruction Fetch



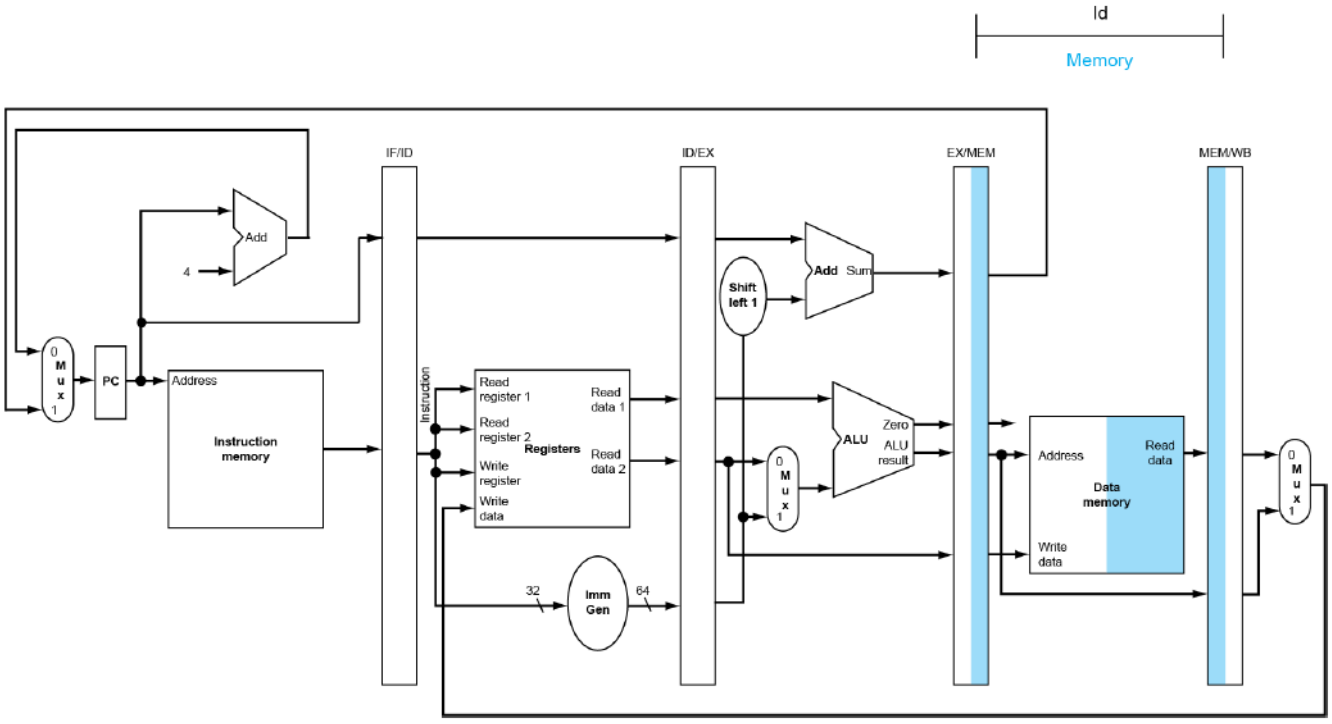
Instruction Decode



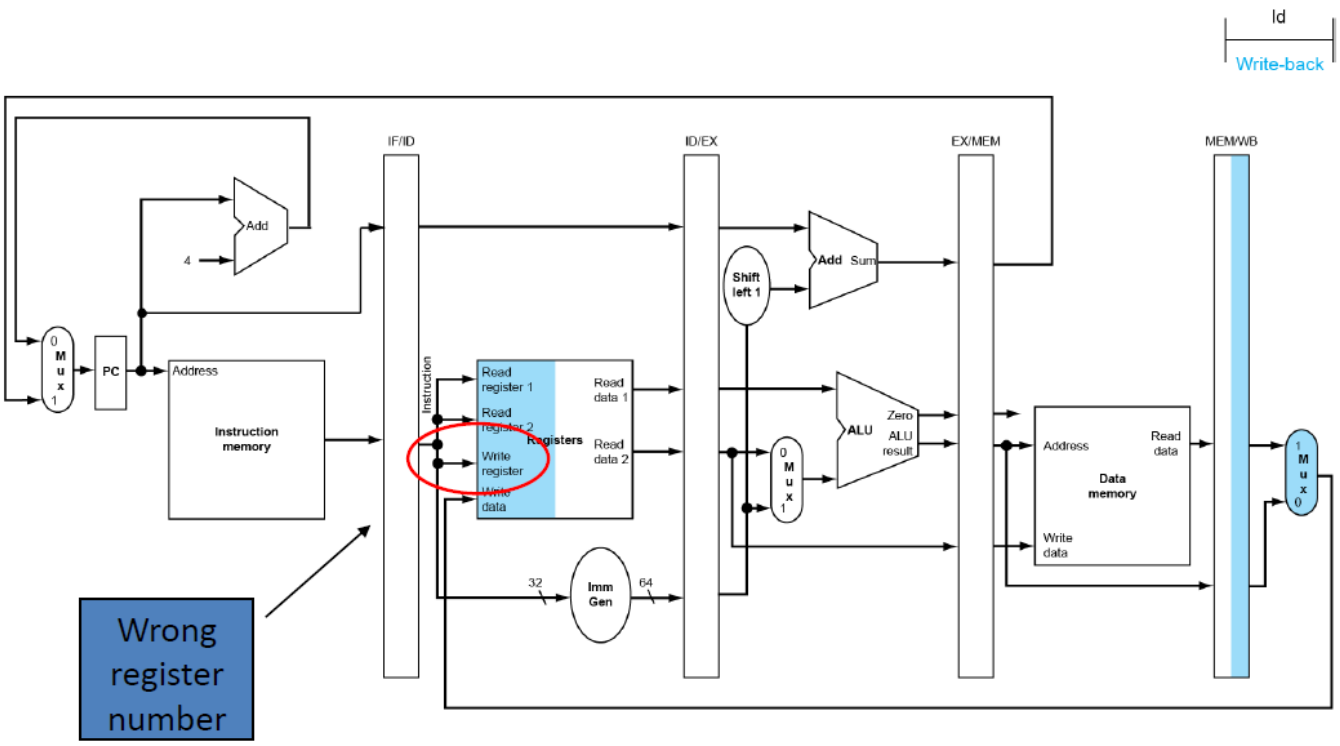
Execute



Memory Access



Writeback


ld
Write-back
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 1
Add
Sum
ALU
Zero
ALU result
Address
Read data
Data memory
Write data
32
Imm Gen
64
Wrong register number