

# IF

## Instruction Fetch

Fetch the current instruction from memory using the program counter (PC) as the address,  
add 4 to the PC,  
and store to NPC

# ID

## Instruction Decode/ Register Fetch

Determine which instruction we are holding,  
fetch the register values (two, always, in this  
instruction set),  
compare the two registers and set the **EQUAL**  
flag if equal.

# EX

## Execution/ Effective Address

Depending on the instruction type:

- . **Memory reference:** add the base register and the offset to form the effective address
- . **Register-Register ALU instruction:** perform the operation (e.g., add, multiply, logic operation)
- . **Register-Immediate ALU instruction:** perform op on first register and the immediate value

# MEM

## Memory

If the instruction is a **LOAD** or a **STORE**, do the appropriate thing.

Update the **PC** using either **NPC** or the output of the **ALU** operation.

# WB

## Write-Back

Write result back to register file.