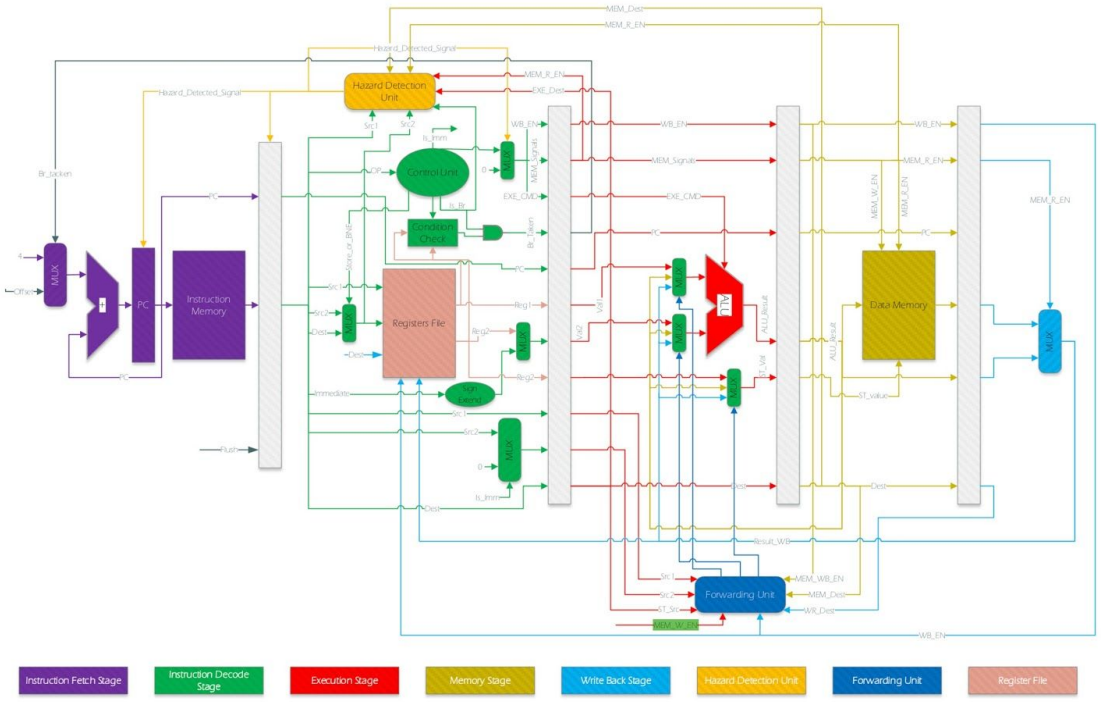
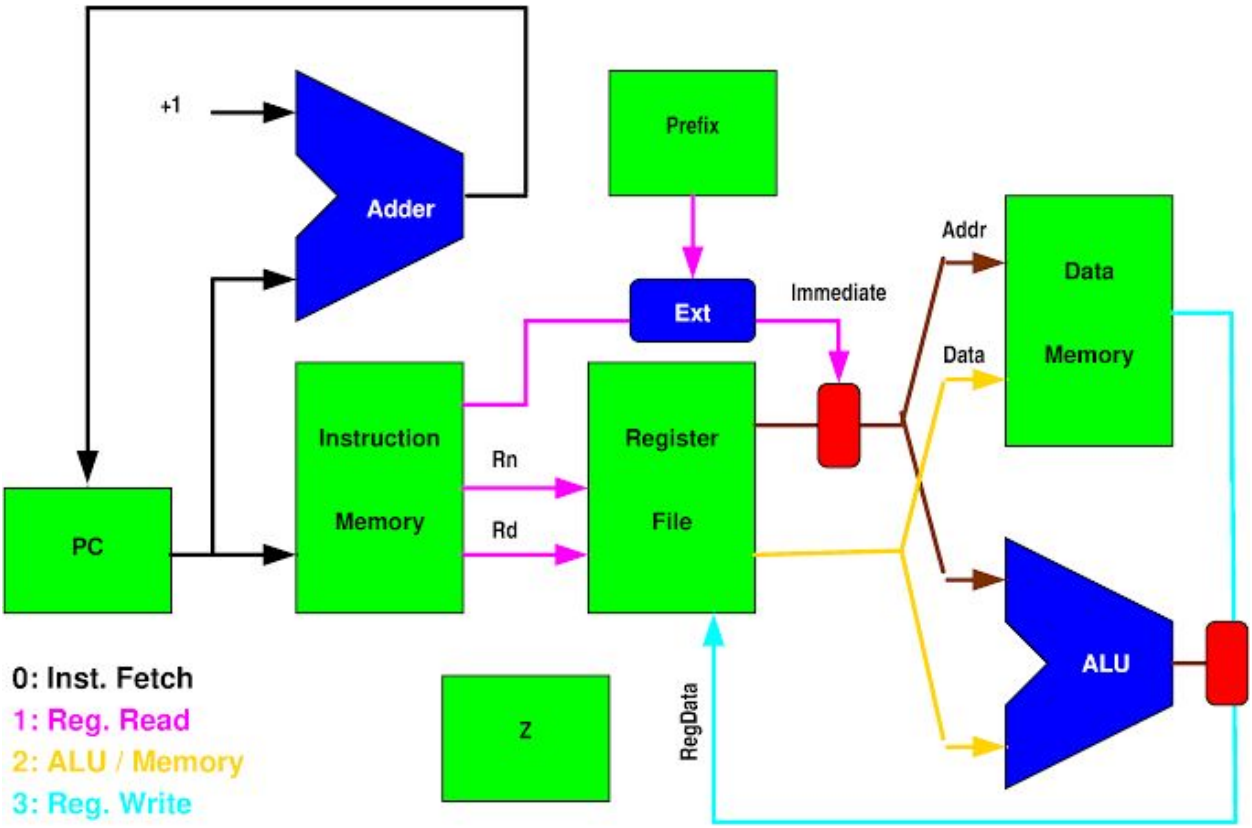
TOP LEVEL CIRCUIT OF PIPELINING



We use the **Program Counter(PC)** to act as Instruction memory.We get the instruction from there.
We look at the instruction to **decode** it to figure out which type of instruction it is.
Meanwhile we could be reading the registers .Once we have read the registers we can feed them to the **ALU** where we are going to do Add,subtract,xor or whatever the

instruction wants us to do.The decoding logic is going to tell the ALU what to do.
Once we get the results of ALU we are done and **write the results to the registers.**

Or we could have a load or store instruction in which case what the ALU computed is really the address that we use to access the **data memory** and what comes out of data memory is what we end up writing to our registers and of course there is stuff to do with branches and so on but basically we can do one instruction per cycle by starting at PC,fetching the instruction,accessing the registers,doing the operation ,accessing the data memory and then writing the results back to the registers.
So there are basically **5 phases** :
1.**Fetching the instruction**
2.**Decoding (Reading the registers)**
3.**ALU**
4.**Access the memory**
5.**Write Back.**


+1
Adder
Prefix
Ext
Immediate
Addr
Data
Data
Memory
Instruction
Memory
Rn
Rd
Register
File
PC
ALU
RegData
Z
0: Inst. Fetch
1: Reg. Read
2: ALU / Memory
3: Reg. Write