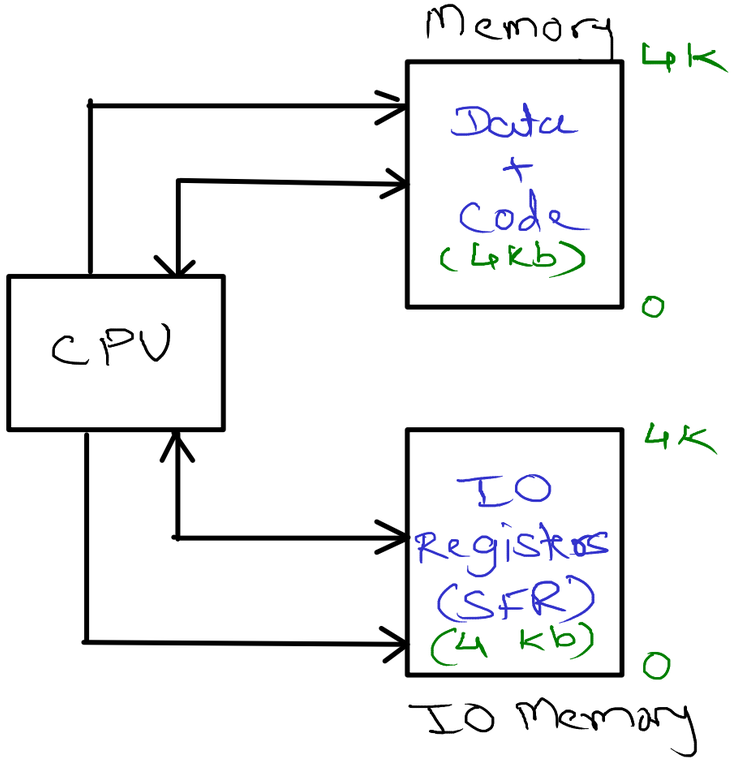
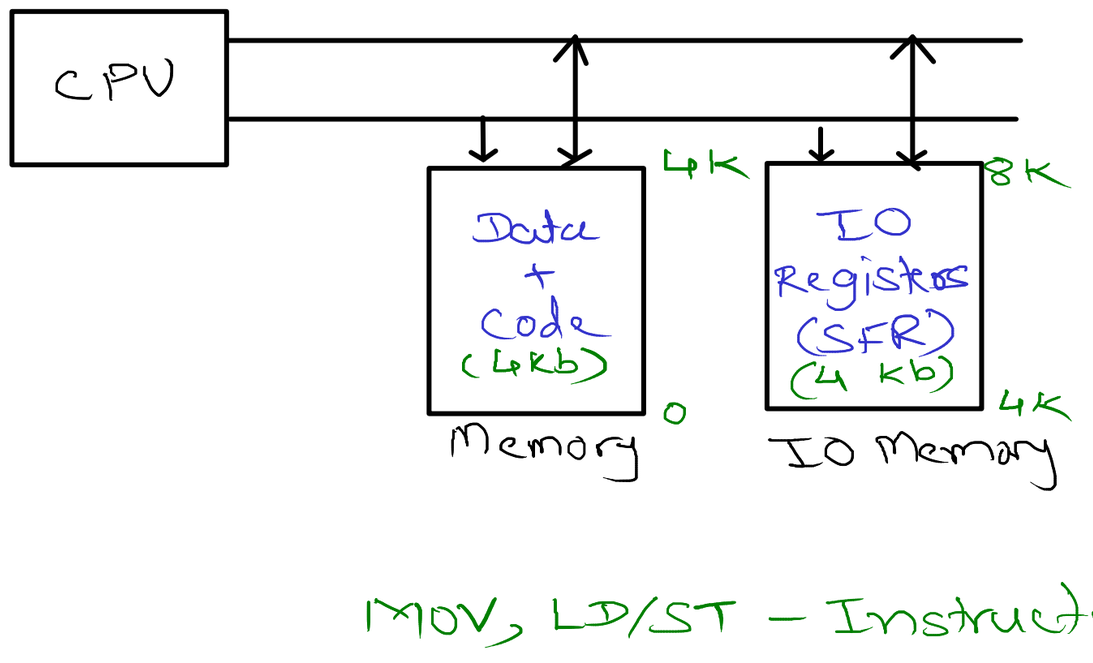
IO mapped IO Vs Memory mapped IO



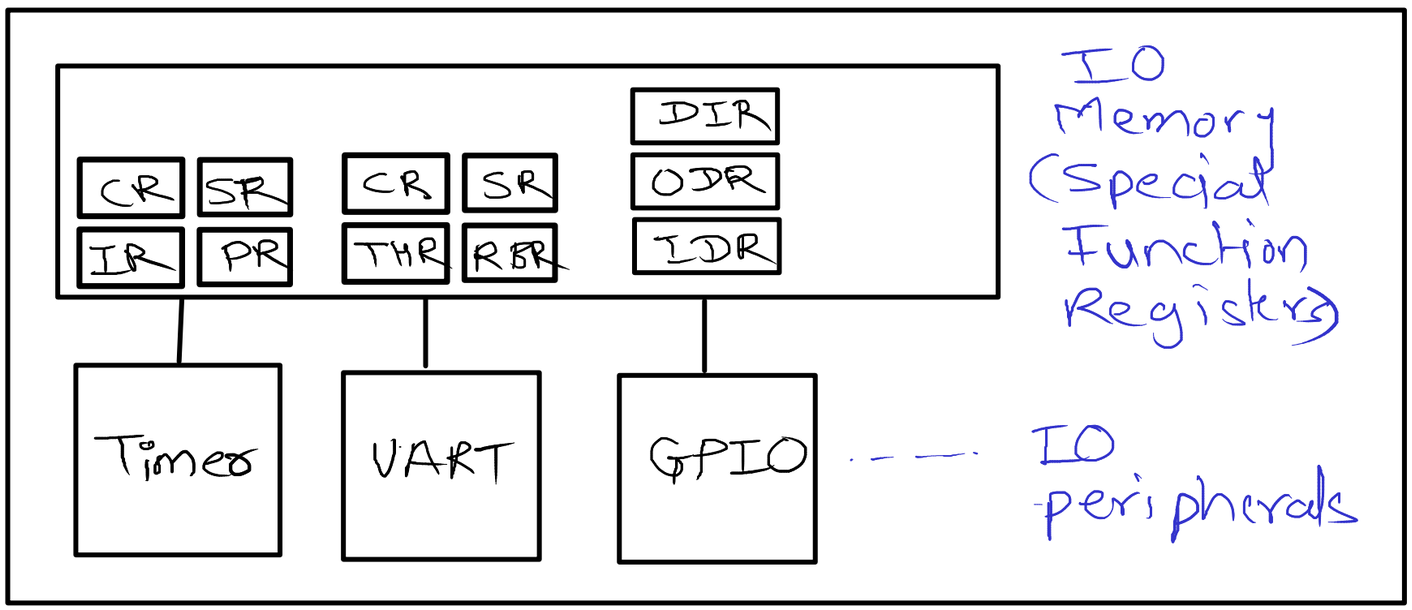
IO/M

MOV, LD/ST - Memory - 0

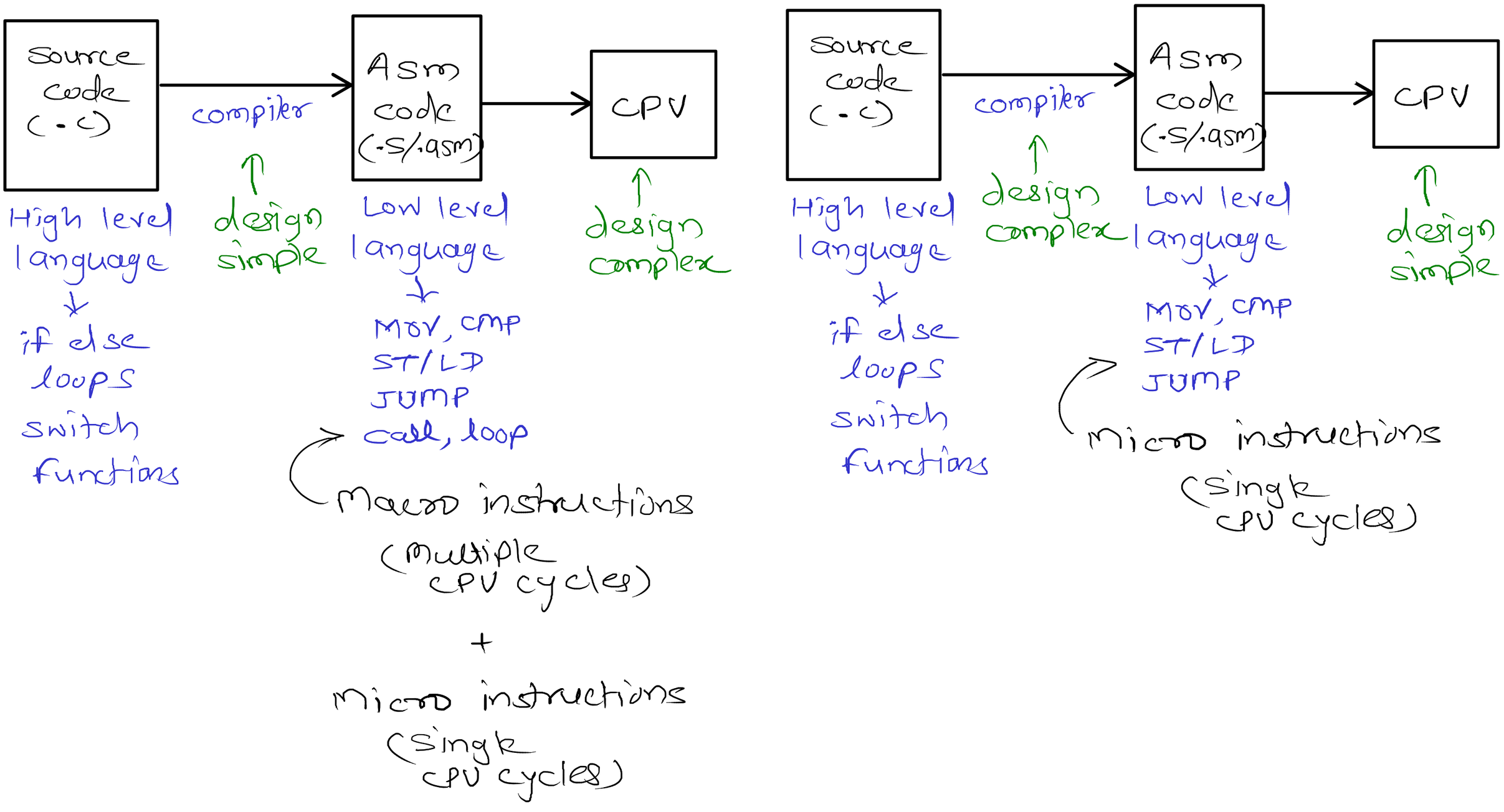
IN/OUT - IO - 1

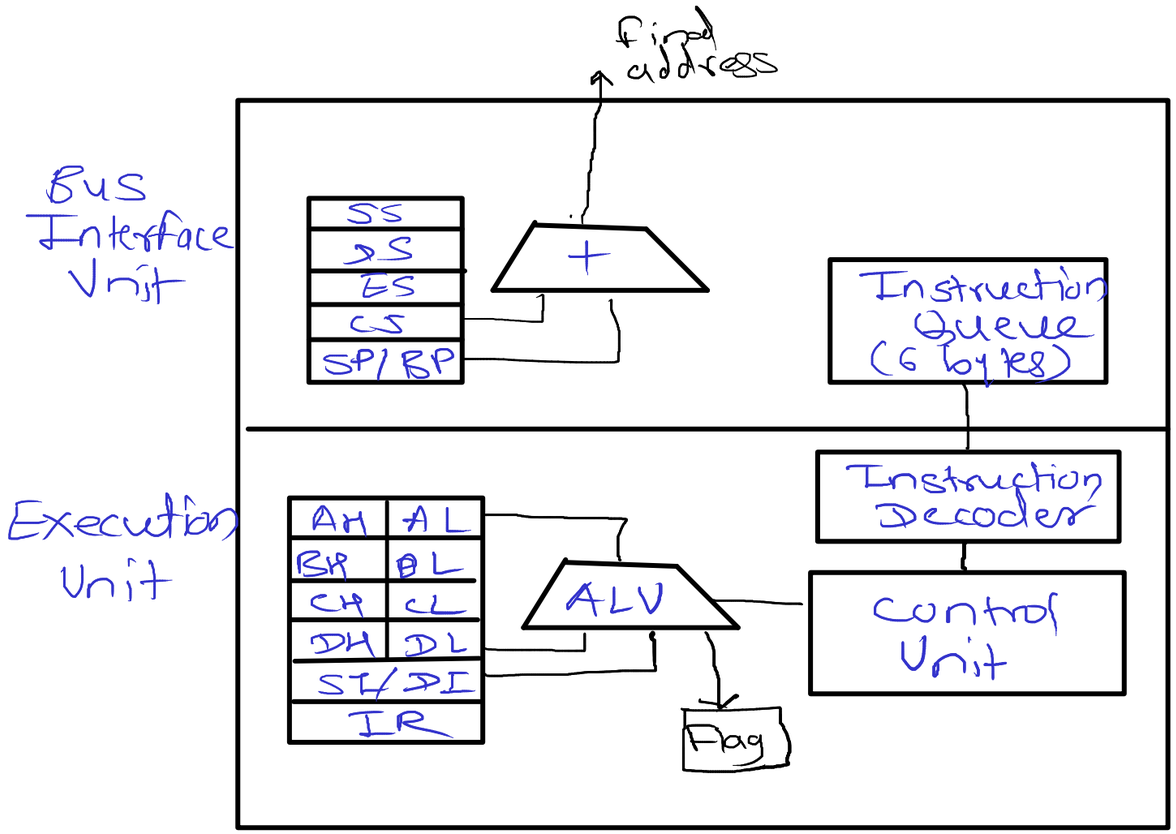


MOV, LD/ST - Instructions

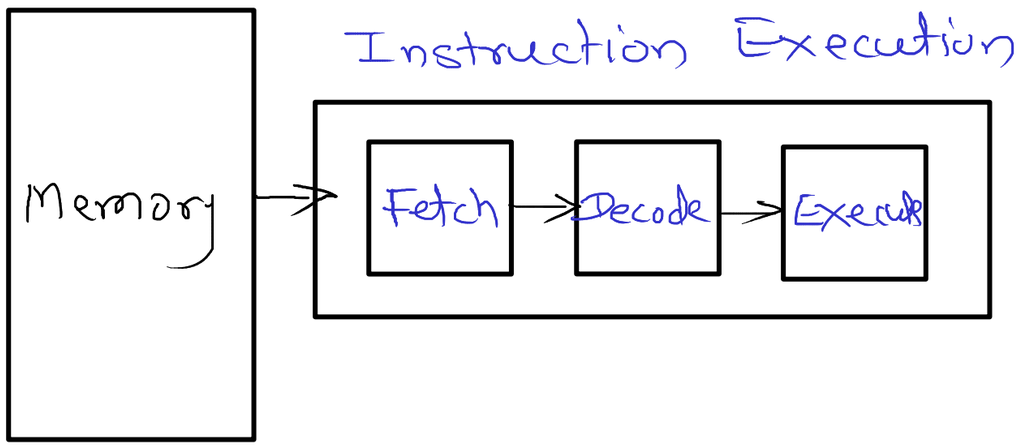


CISC Vs RISC


Source code (.c)
compiler
Asm code (.s/.asm)
CPU
High level language
if else
loops
switch
functions
design simple
Low level language
MOV, CMP
ST/LD
JUMP
call, loop
Macro instructions
(Multiple CPU cycles)
+
Micro instructions
(Single CPU cycles)
design complex
Source code (.c)
compiler
Asm code (.s/.asm)
CPU
High level language
if else
loops
switch
functions
design complex
Low level language
MOV, CMP
ST/LD
JUMP
Micro instructions
(Single CPU cycles)
design simple



Instruction pipeline (parallelism)



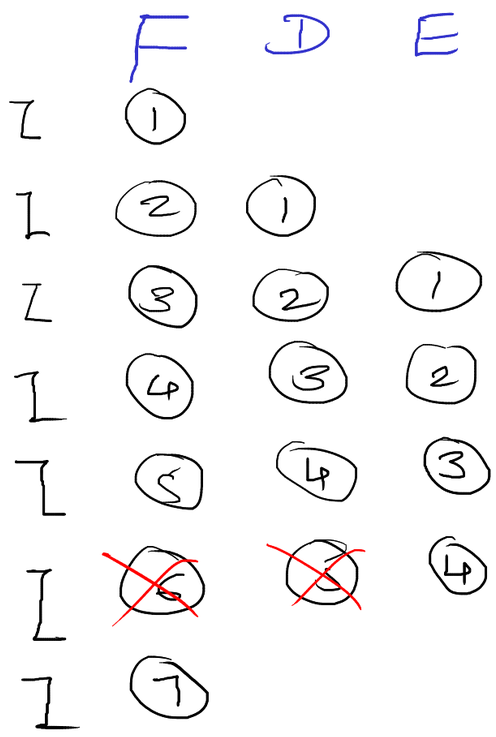
CPU cycles	Fetch	Decode	Execute
1	I1		
2	I2	I1	
3	I3	I2	I1
4	I4	I3	I2
5	I5	I4	I3

effectively per CPU cycle one instruction will be completed.

Pipeline Hazards

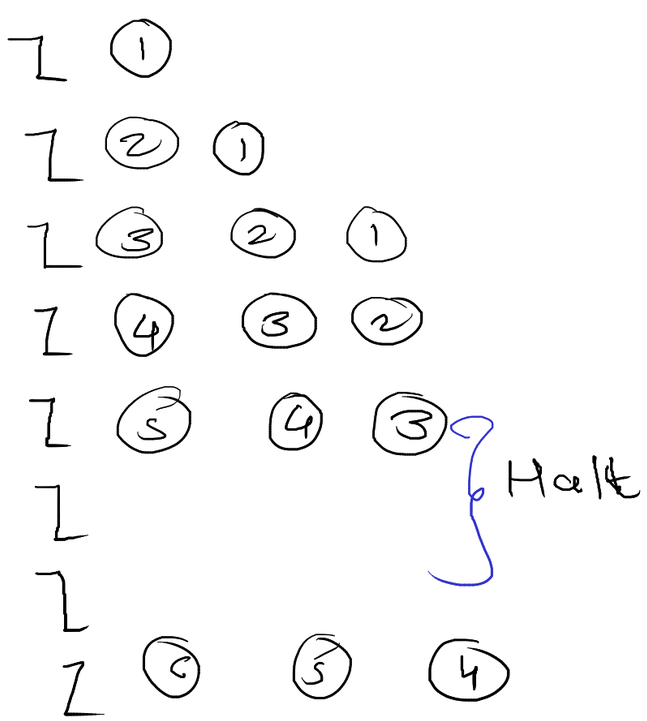
Control Hazard

```
1   A = 10
2   B = 20
3   CMP
4   Branch → T
5   A is greater
6   Jump end
7   T: B is greater
8   end:
```



Data Hazard

```
A = 10
B = 20
IMUL A, B
STA 0x---
=
```



Structural Hazard