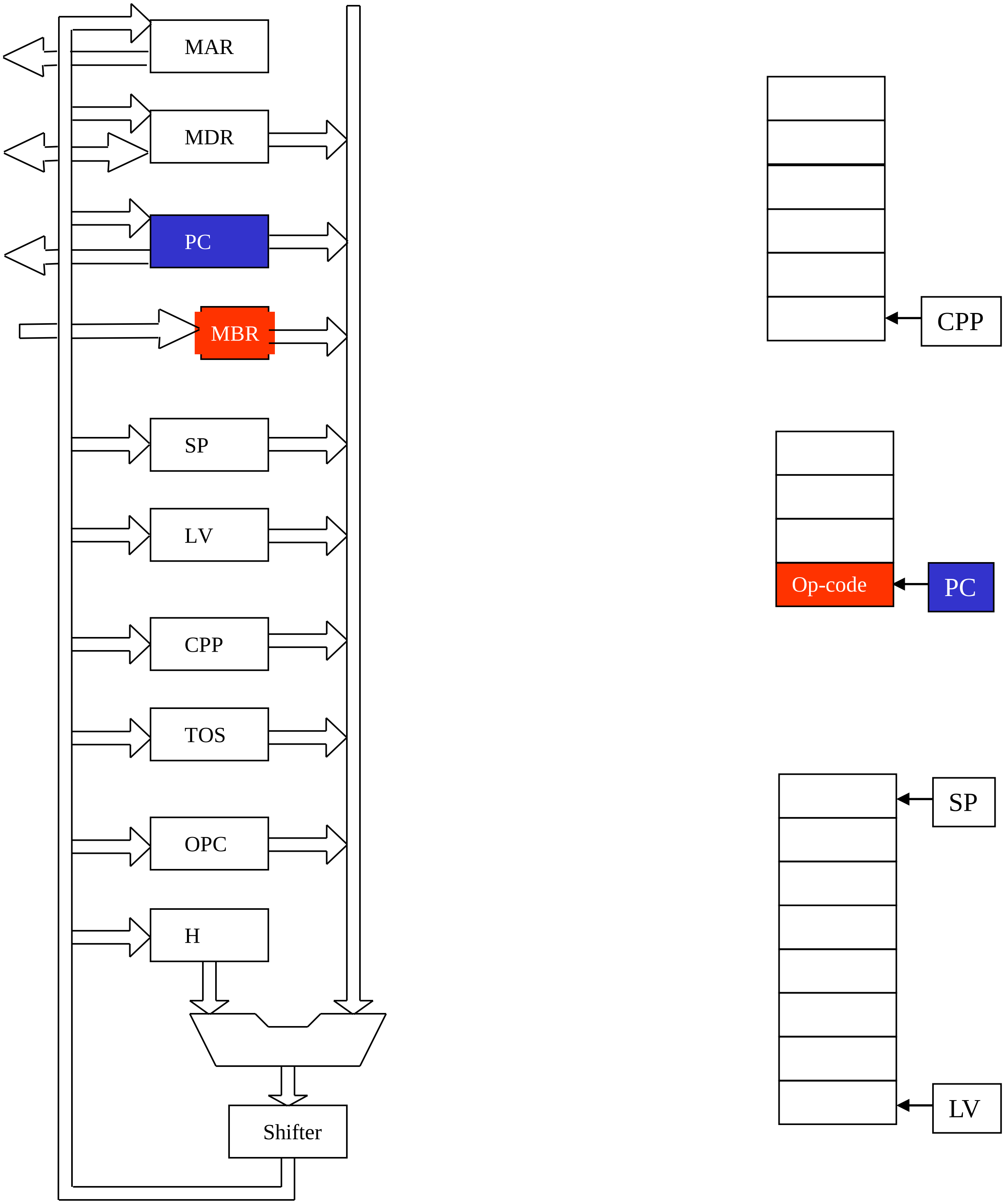
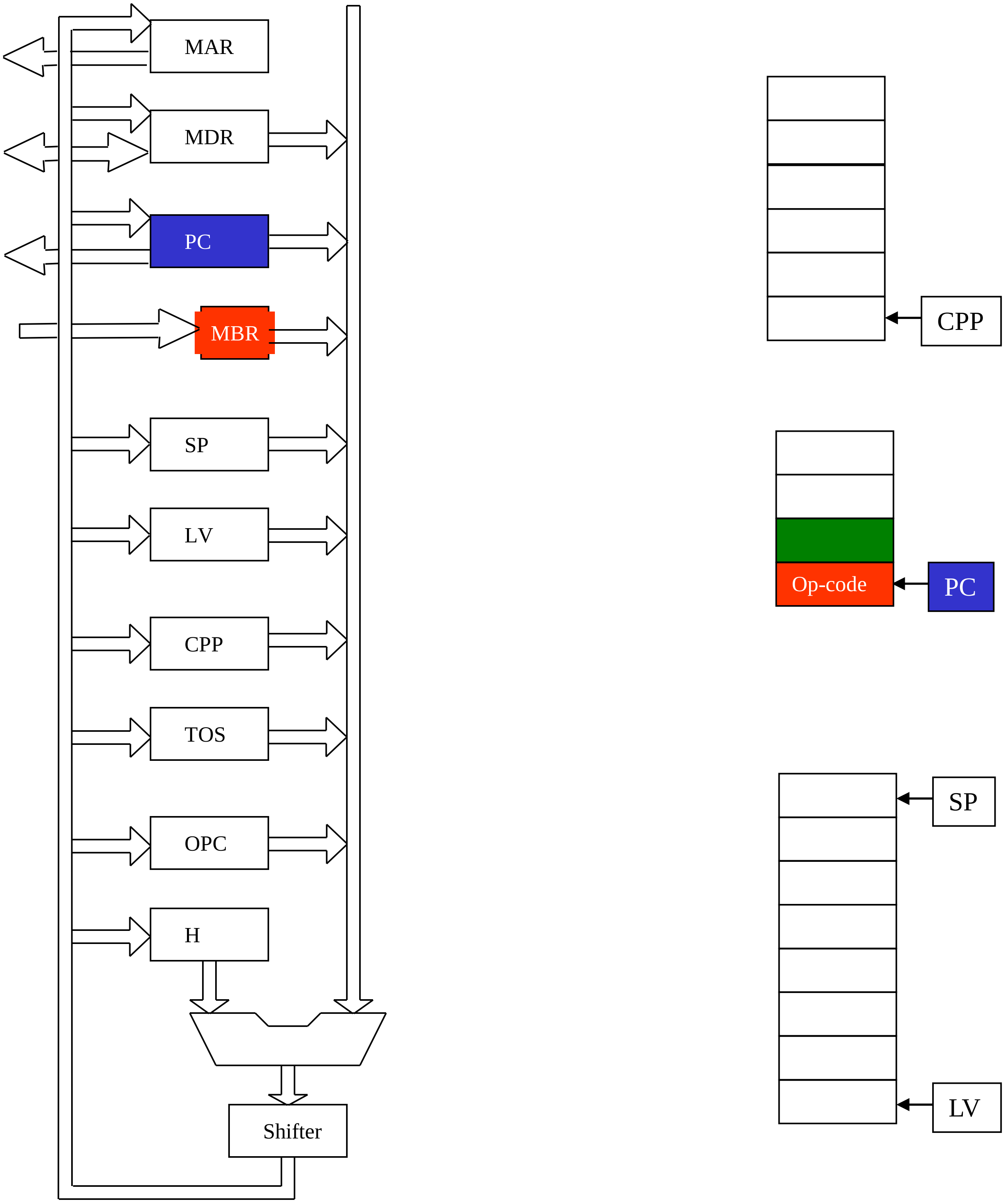
Fetch

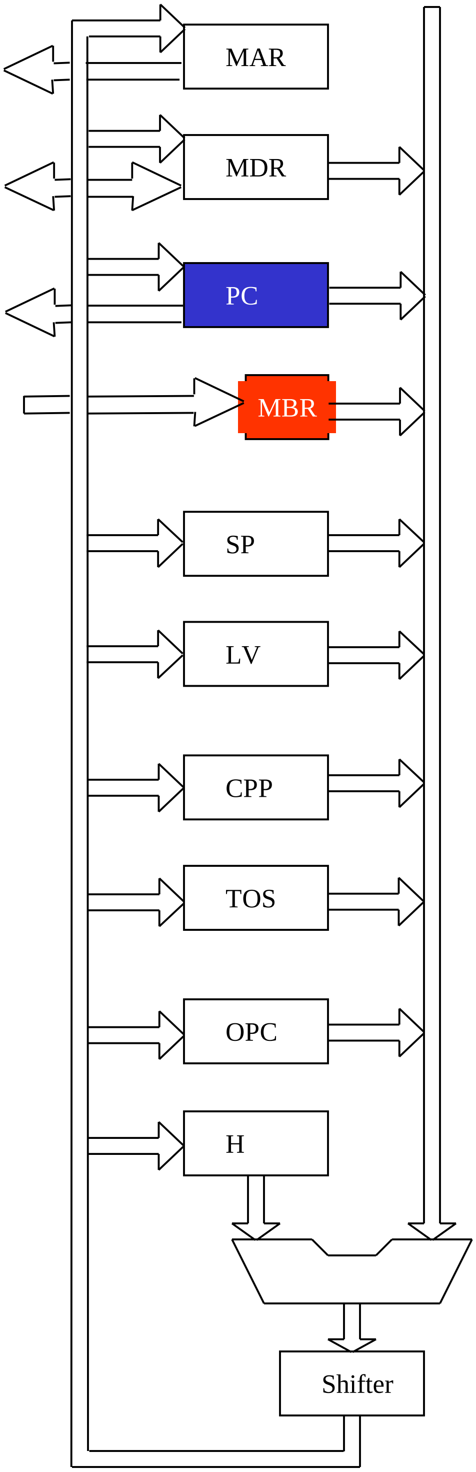


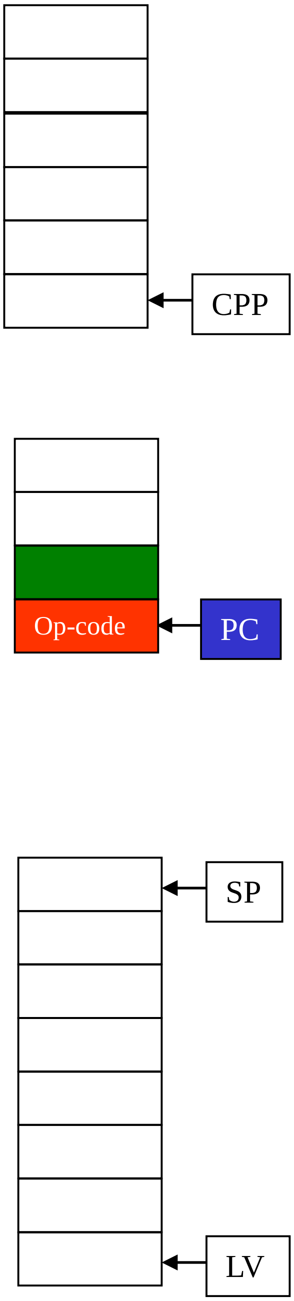
At this point the program counter PC contains the address of the instruction that is about to be executed and a copy of the value at that address (the op-code for the current instruction) was already retrieved and is in the MBR. What we will actually do here is to change the value of the PC to PC+1 and retrieve the content at that address, which si the op-code of the next instruction to be executed.

Fetch



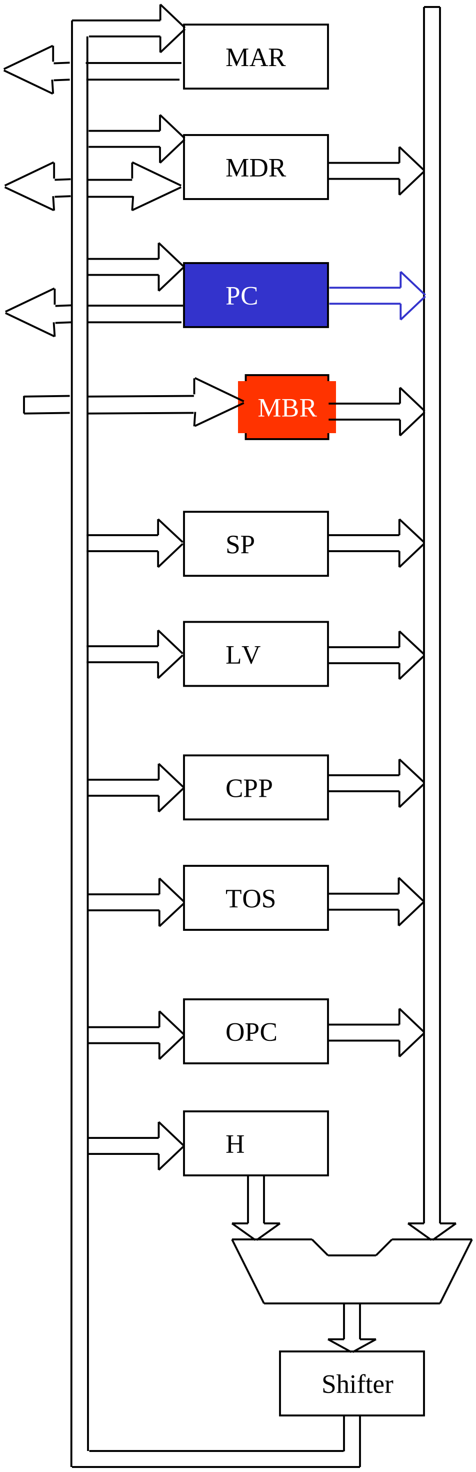
Fetch

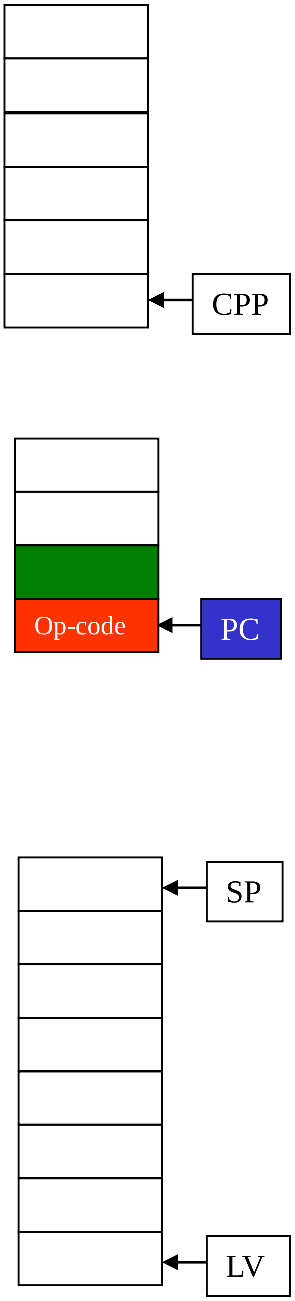




PC = [PC] + 1

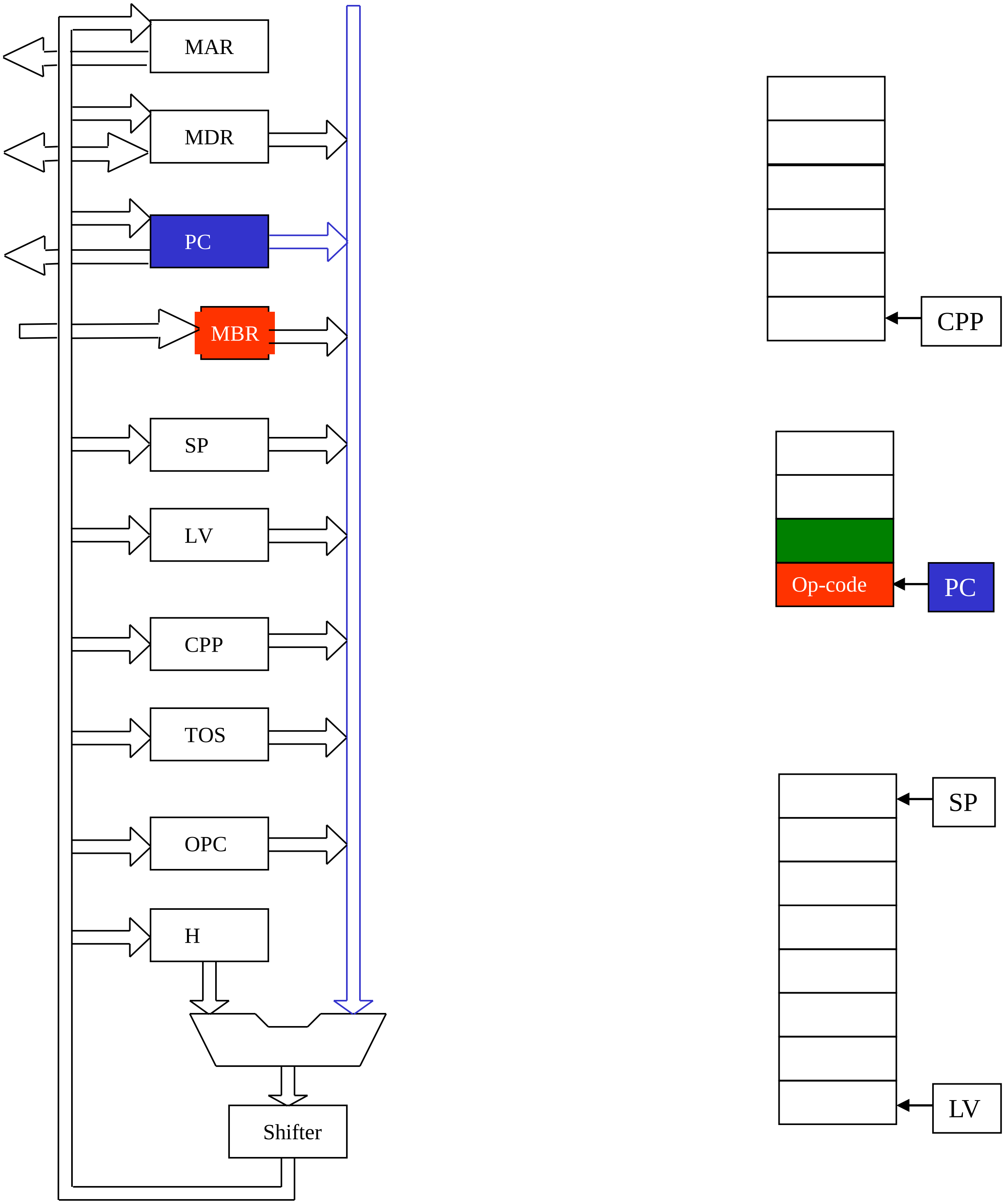
Fetch





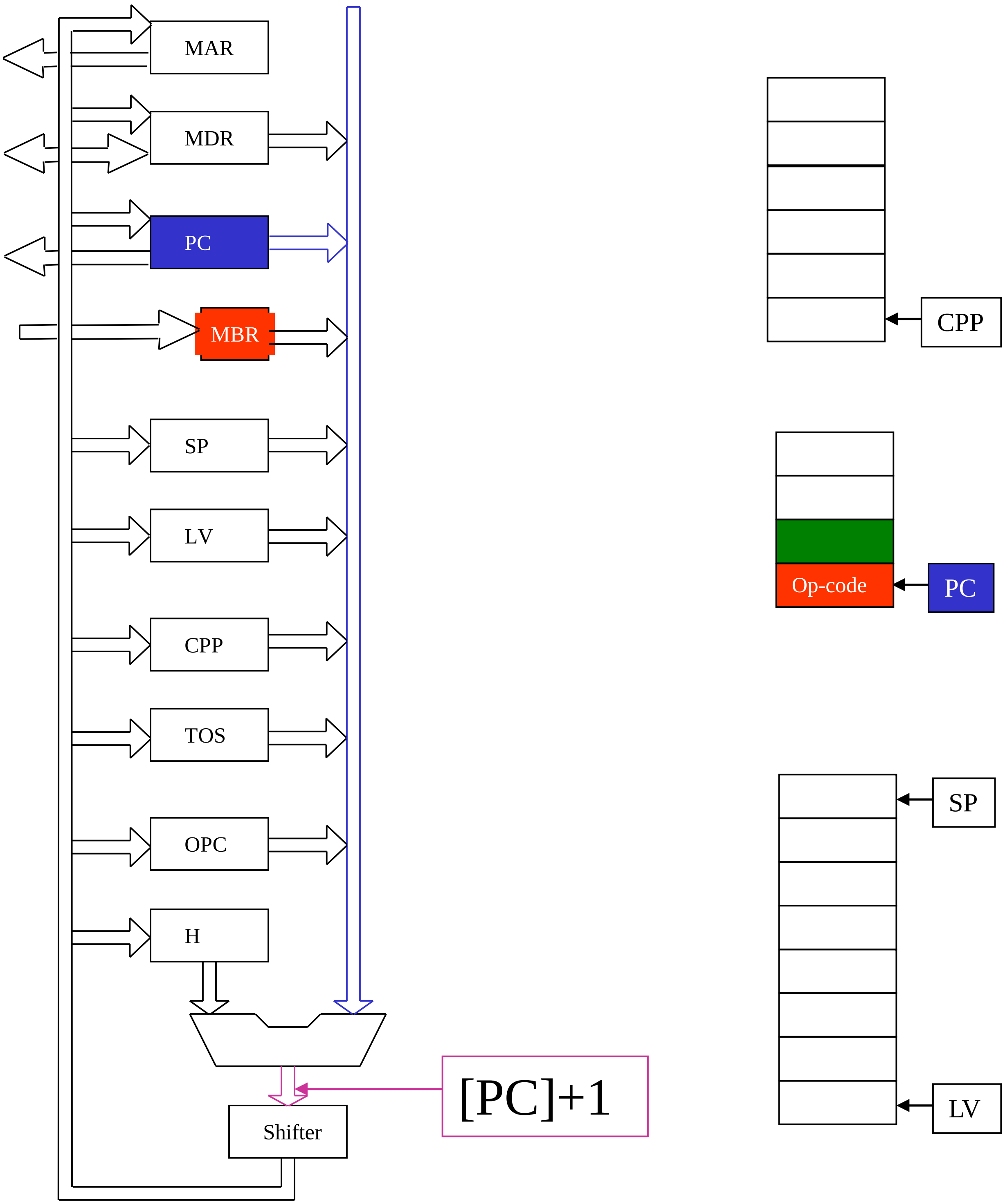
PC = [PC] + 1

Fetch



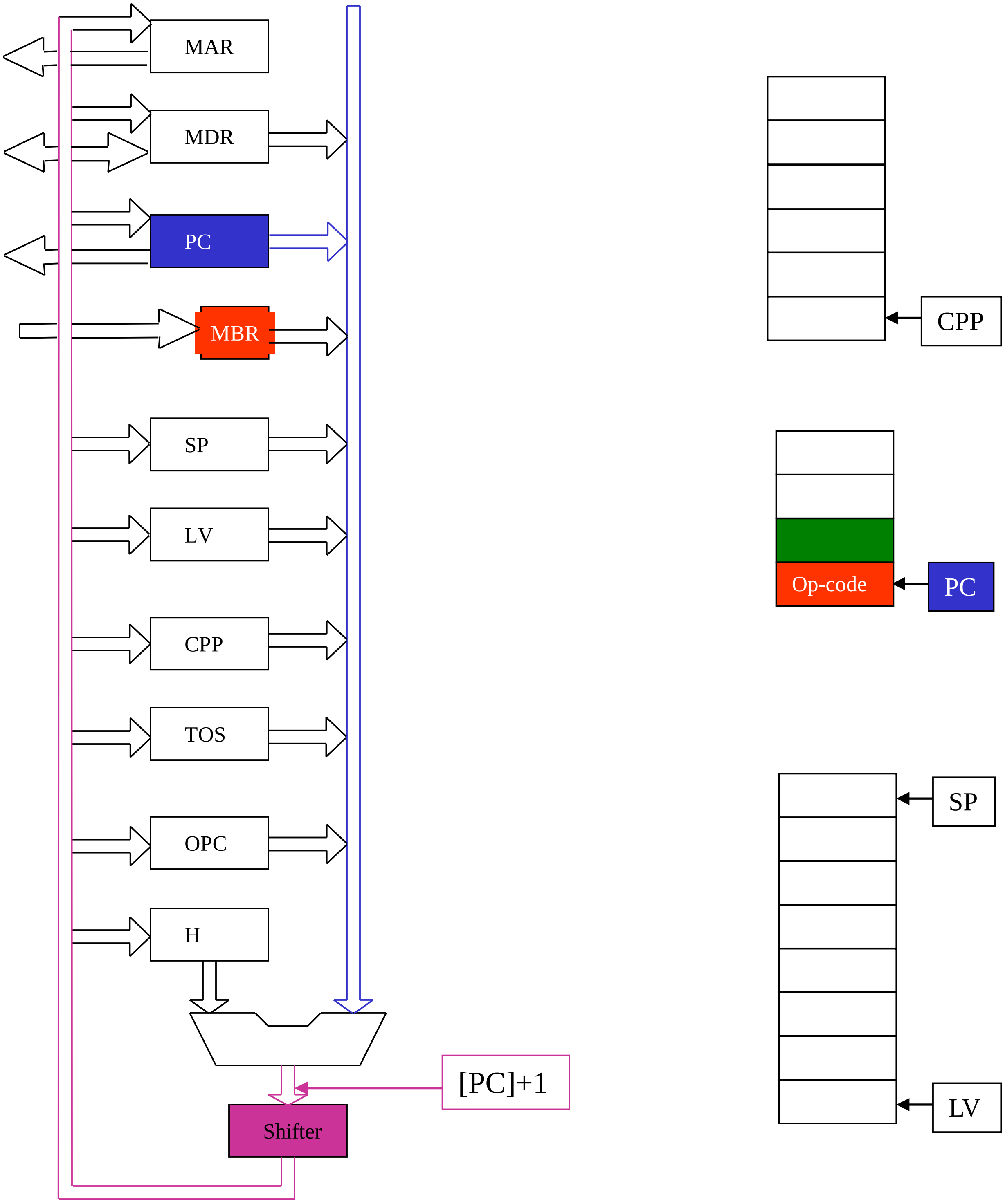
PC = [PC] + 1

Fetch


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter
[PC]+1
CPP
Op-code
PC
SP
LV

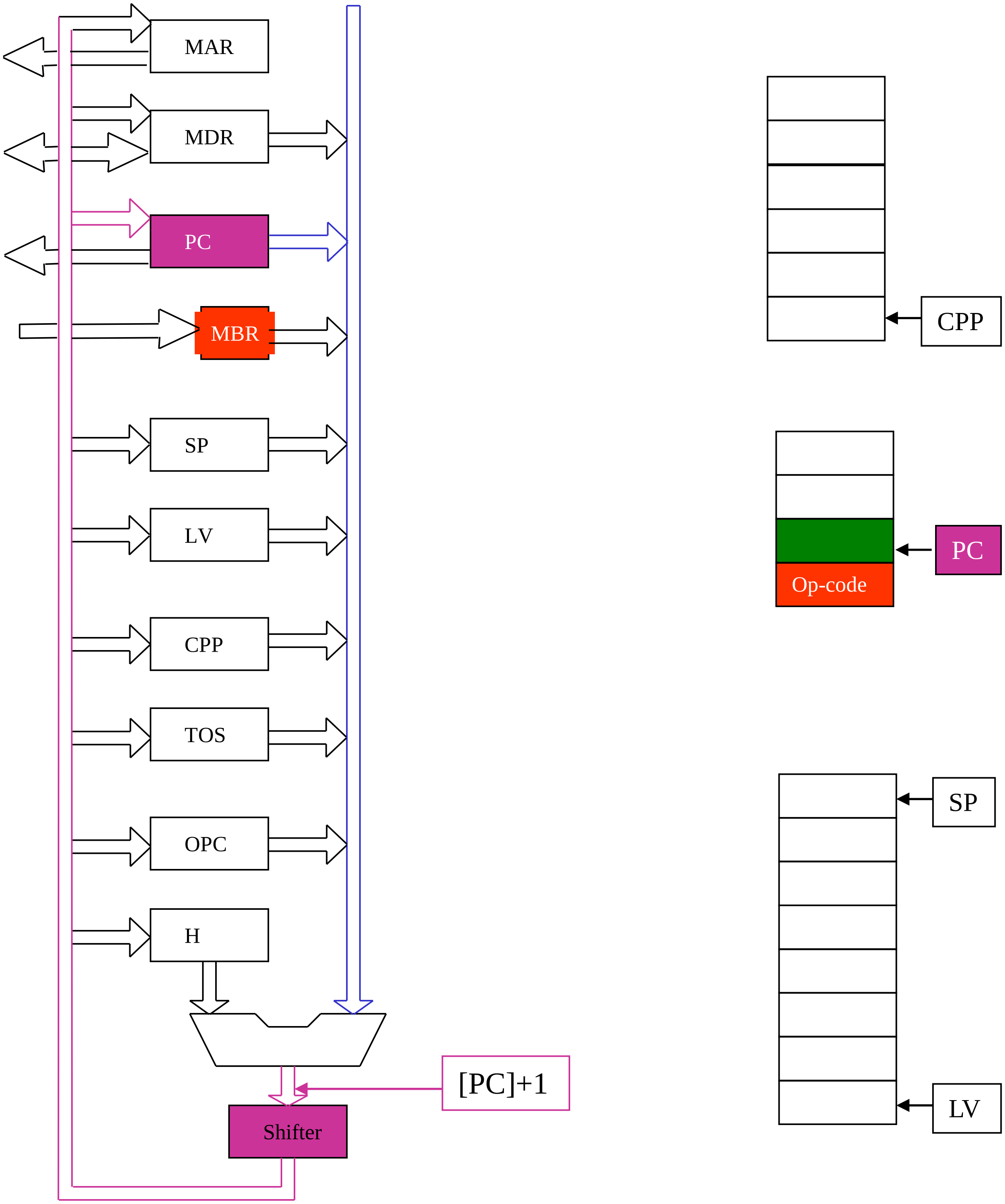
PC = [PC] + 1

Fetch


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
[PC]+1
Shifter
CPP
Op-code
PC
SP
LV

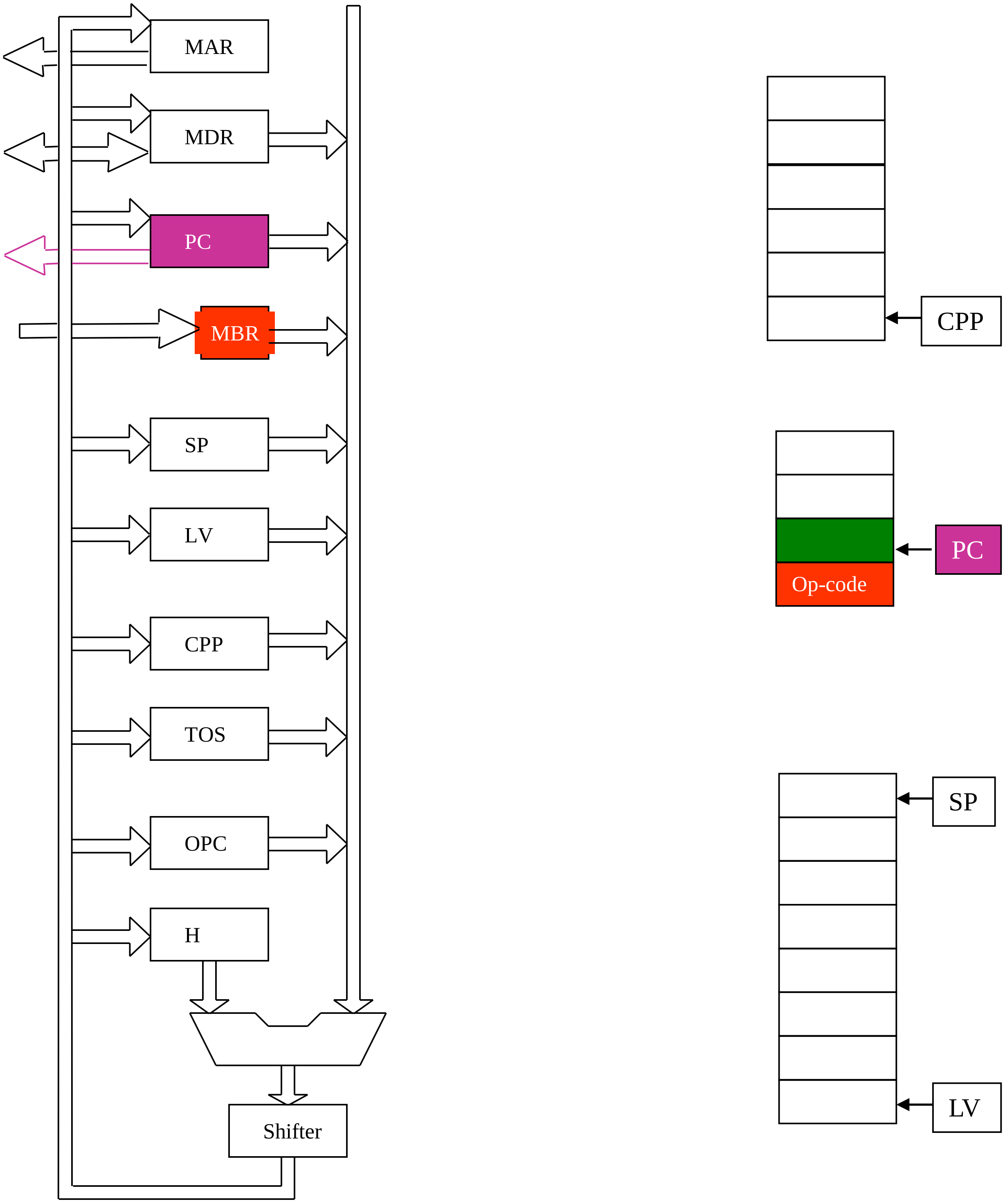
$$PC = [PC] + 1$$

Fetch


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
[PC]+1
Shifter
CPP
PC
Op-code
SP
LV

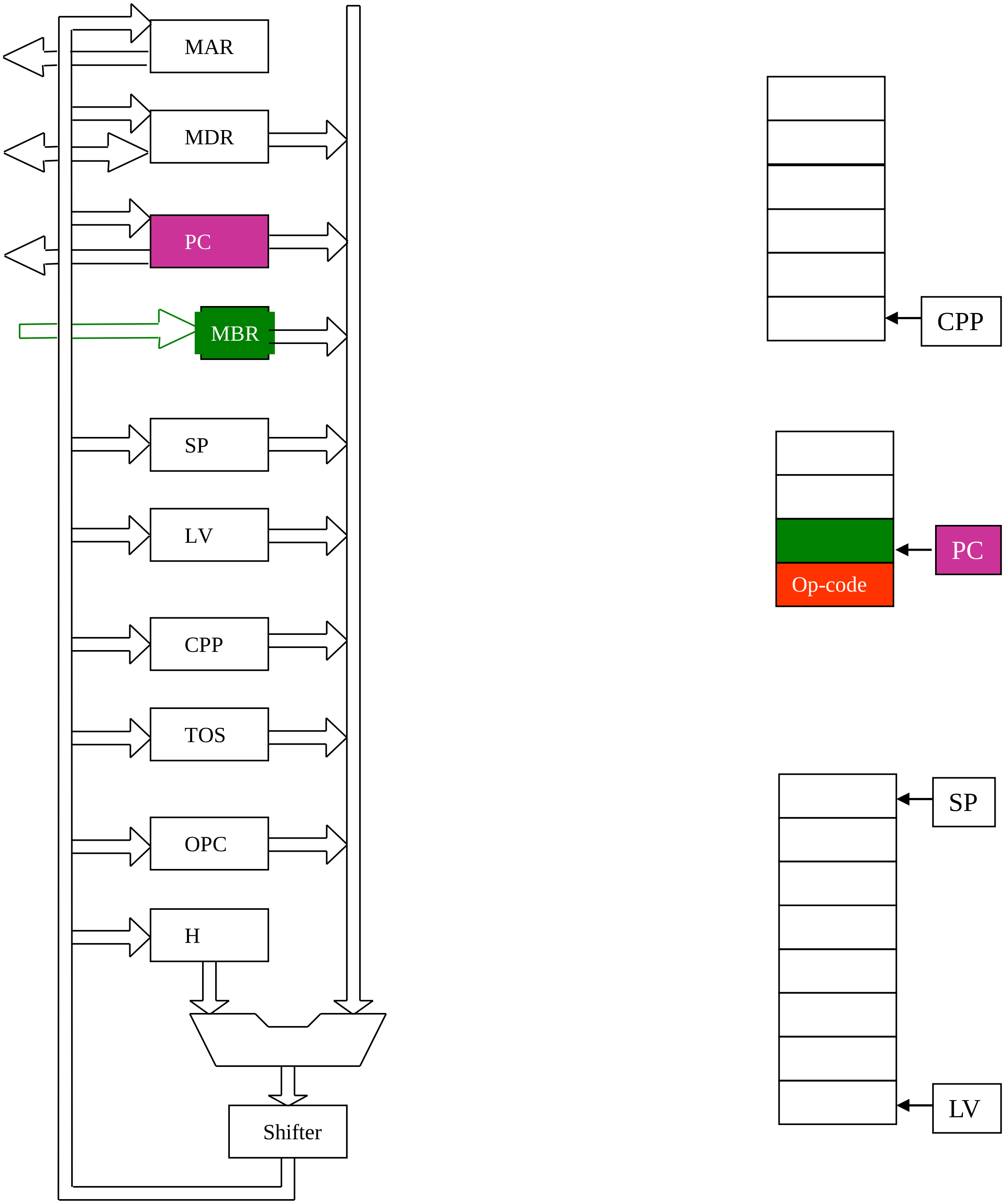
PC = [PC] + 1

Fetch



PC = [PC] + 1; fetch

Fetch



At the end of the next clock cycle