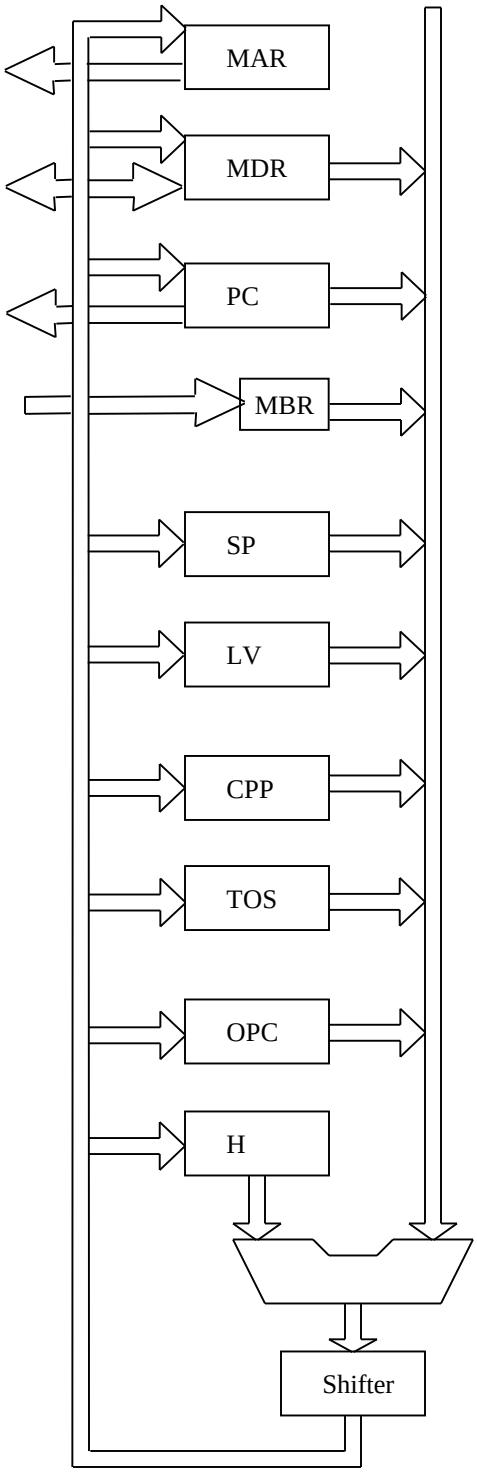
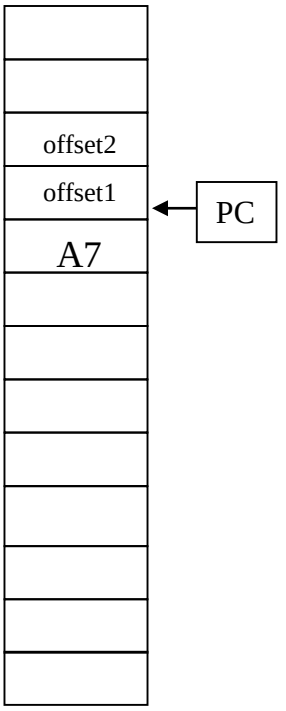
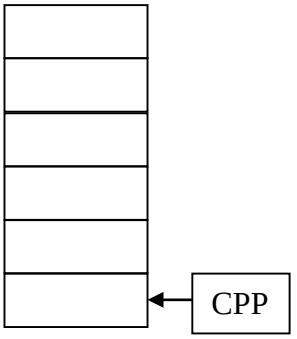
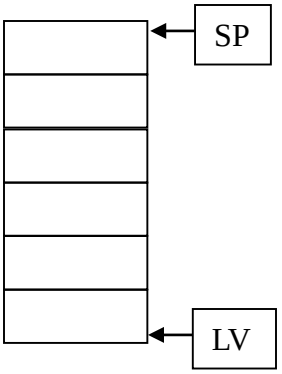
GOTO

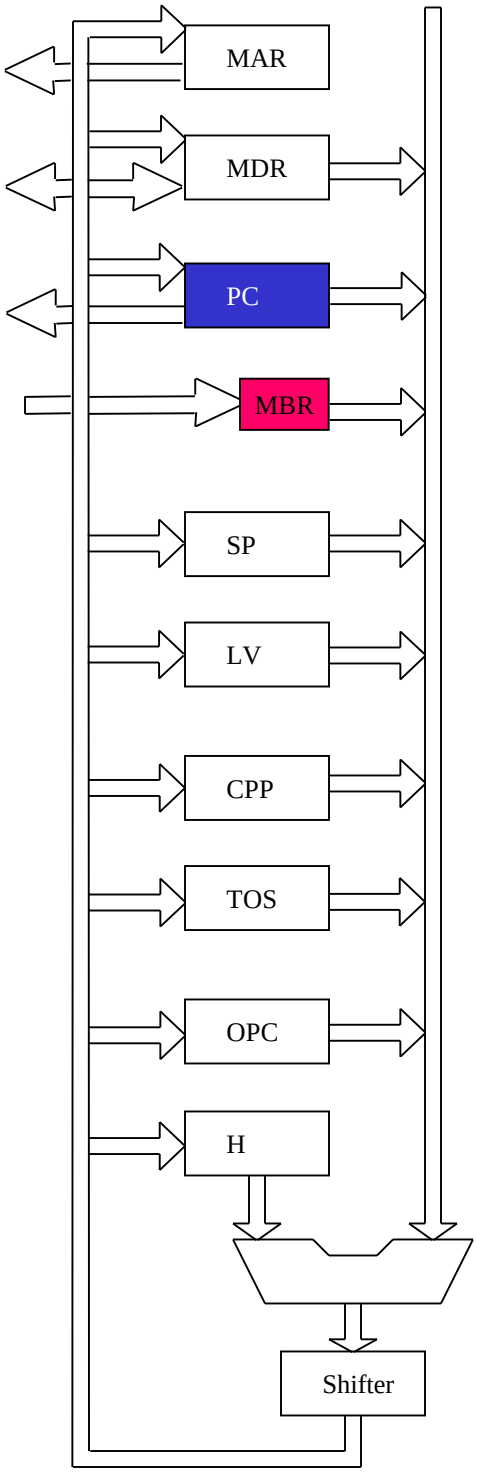

MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

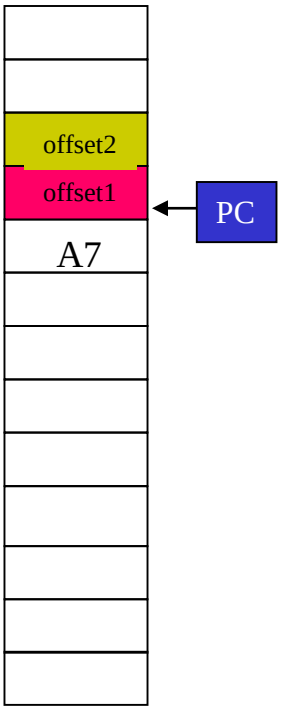

offset2
offset1
A7
PC

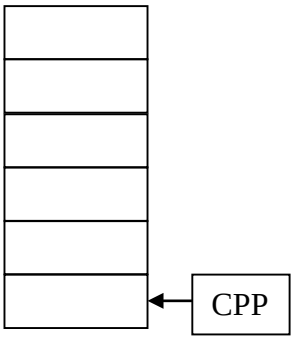

CPP

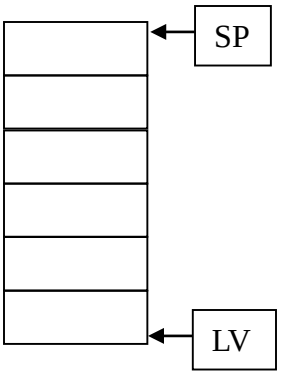

SP
LV

GOTO

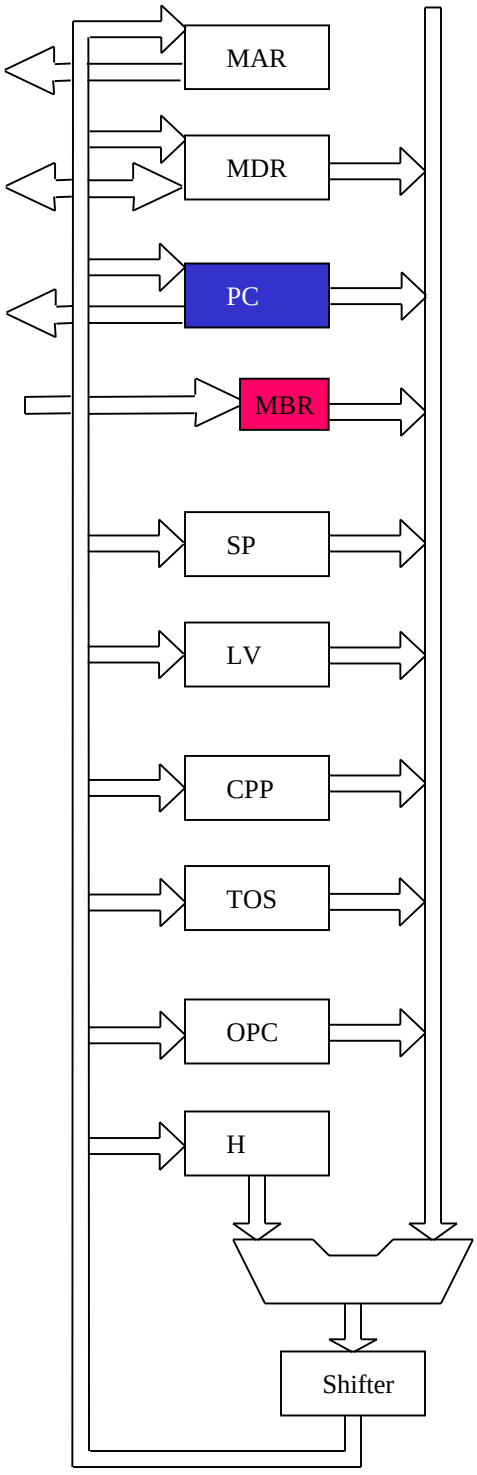

MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

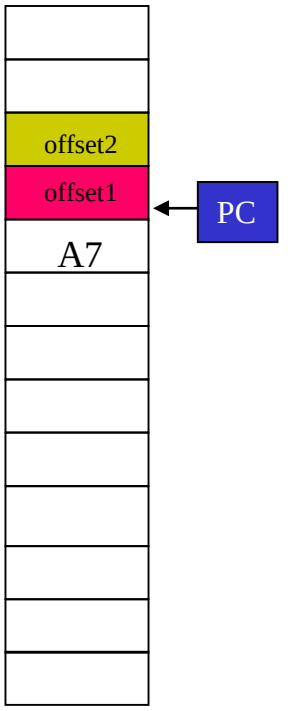

offset2
offset1
PC
A7

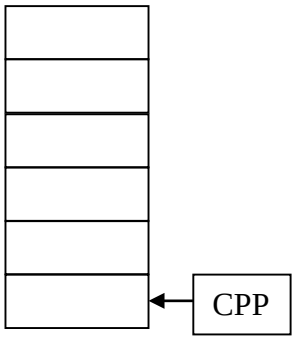

CPP

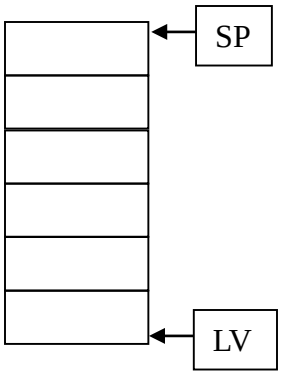

SP
LV

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

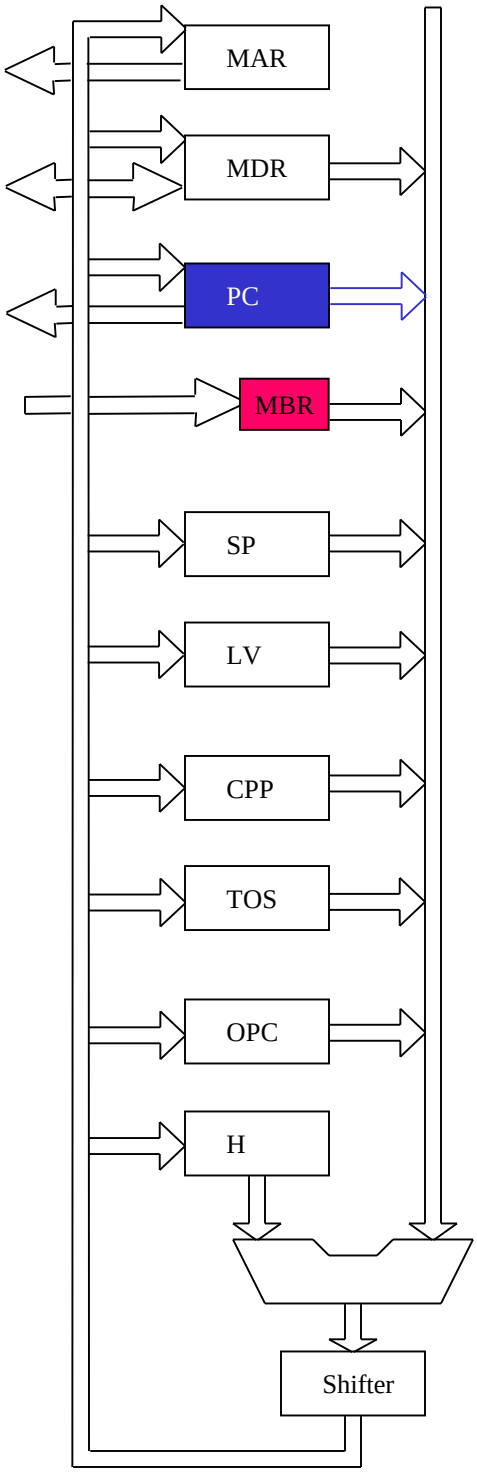

offset2
offset1
PC
A7

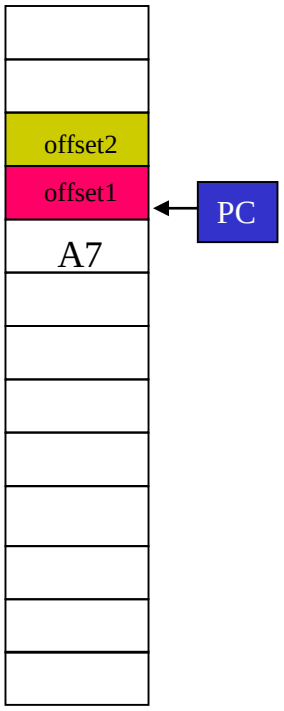

CPP

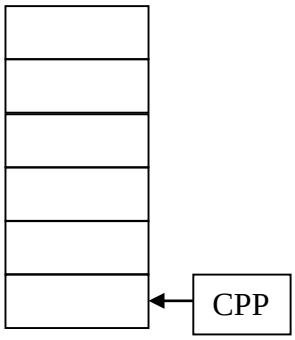

SP
LV

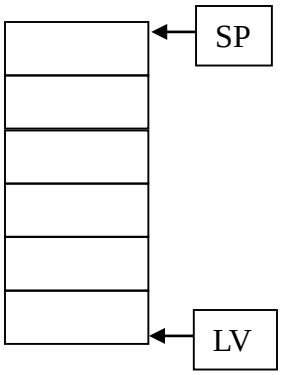
OPC = [PC] - 1

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

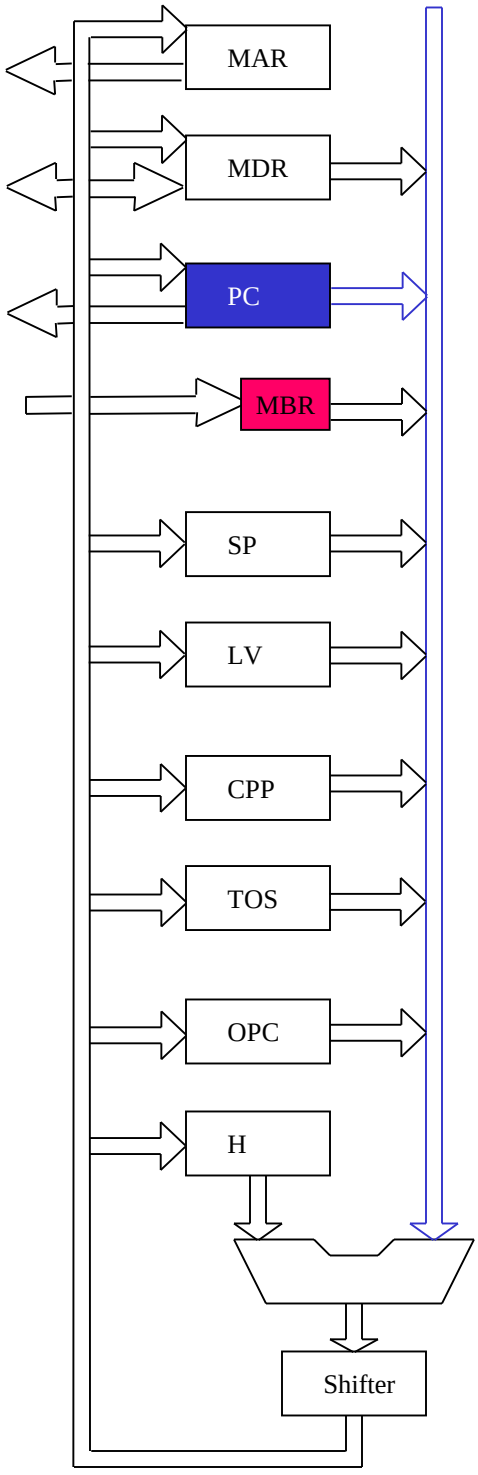

offset2
offset1
PC
A7

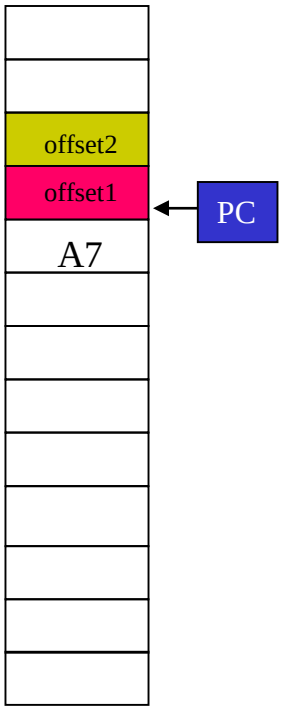

CPP

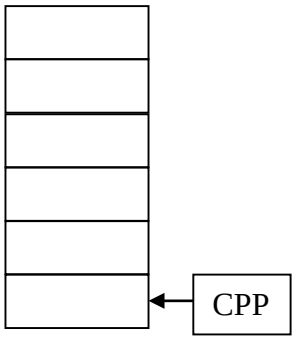

SP
LV

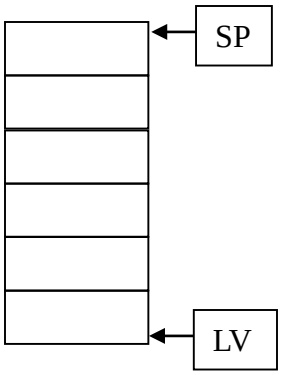
OPC = [PC] - 1

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

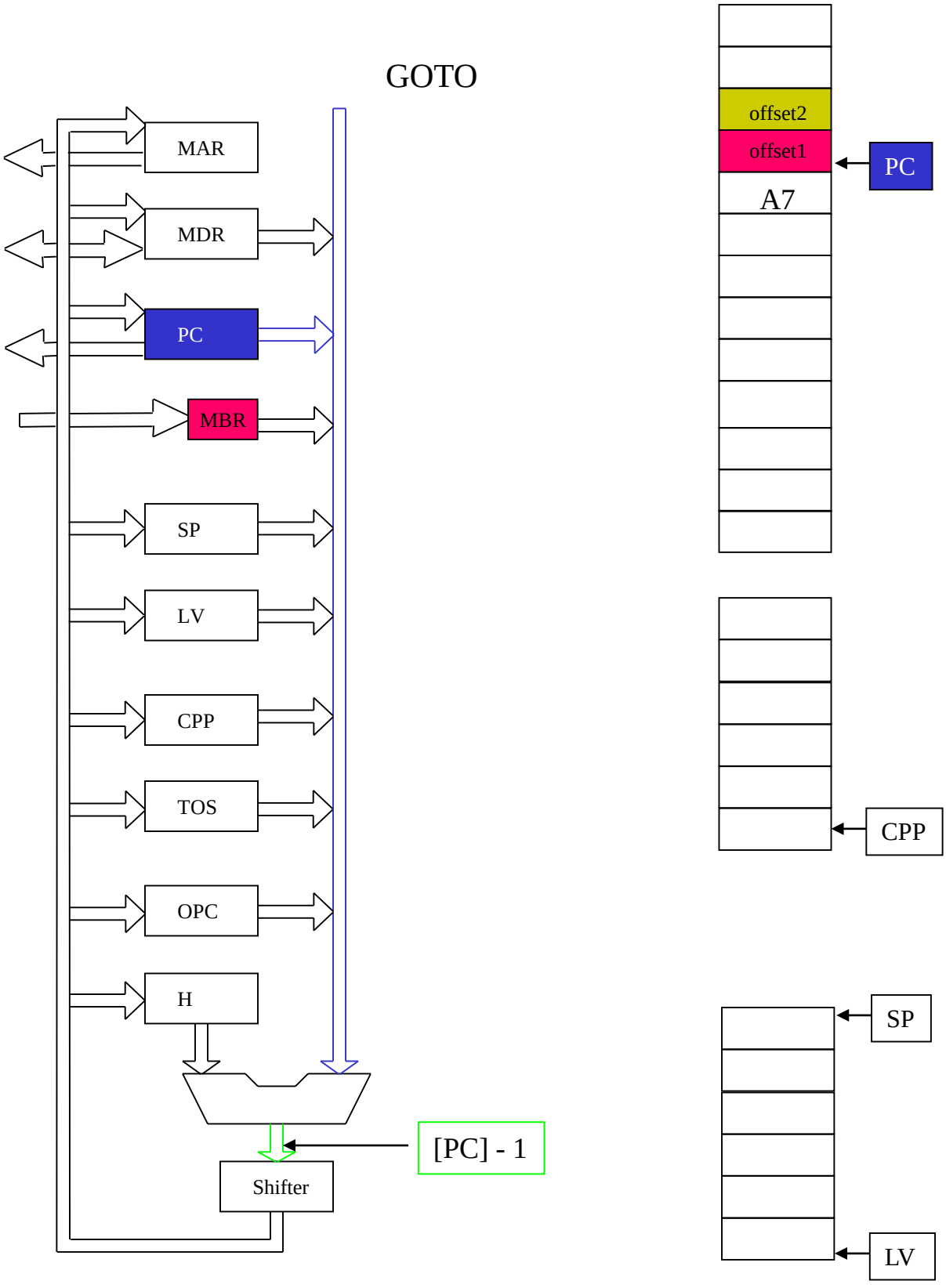

offset2
offset1
PC
A7


CPP


SP
LV

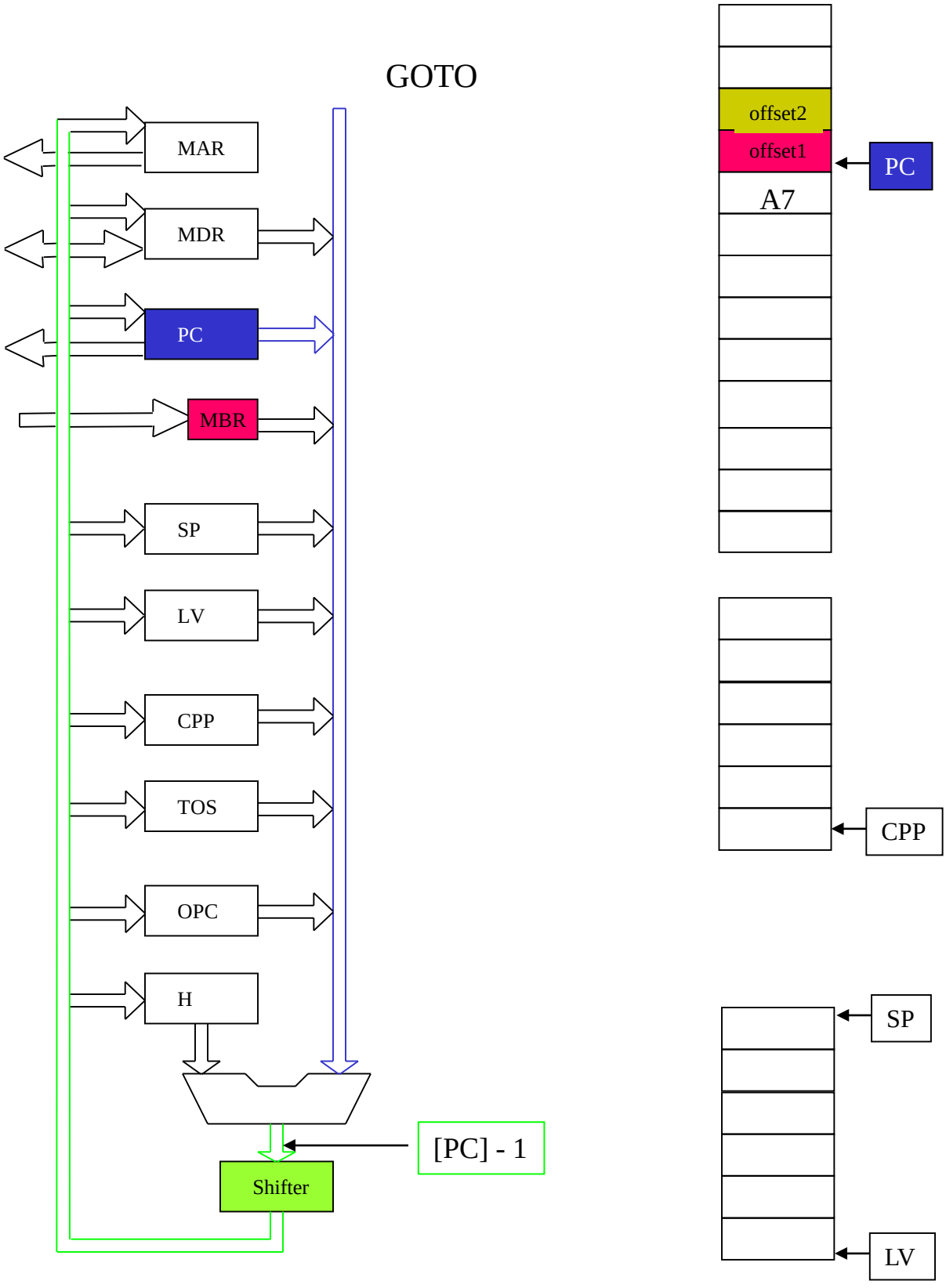
OPC = [PC] - 1

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter
[PC] - 1
offset2
offset1
PC
A7
CPP
SP
LV

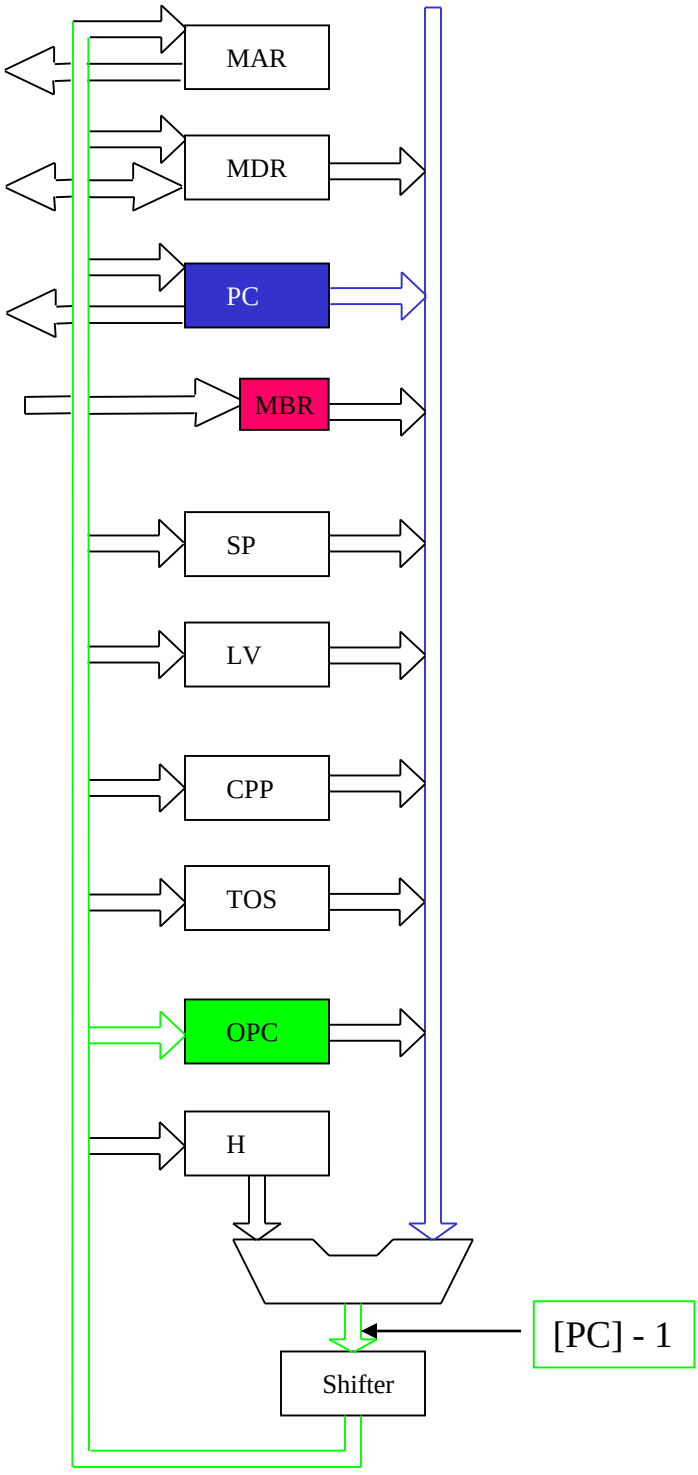
OPC = [PC] - 1

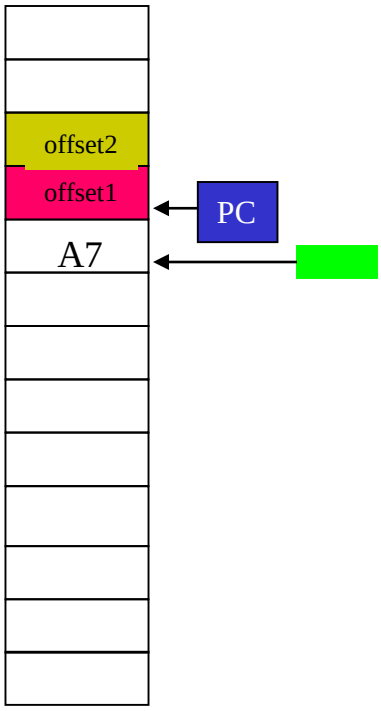
GOTO

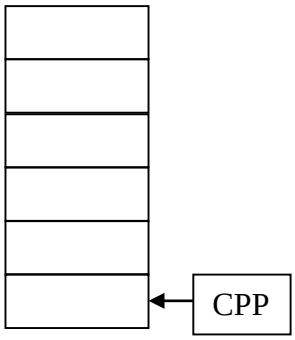

MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
[PC] - 1
Shifter
offset2
offset1
PC
A7
CPP
SP
LV

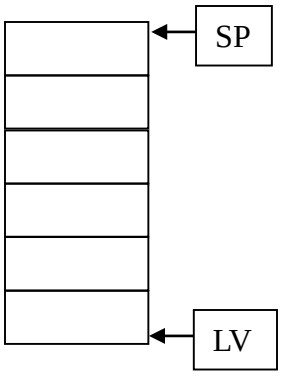
OPC = [PC] - 1

GOTO


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

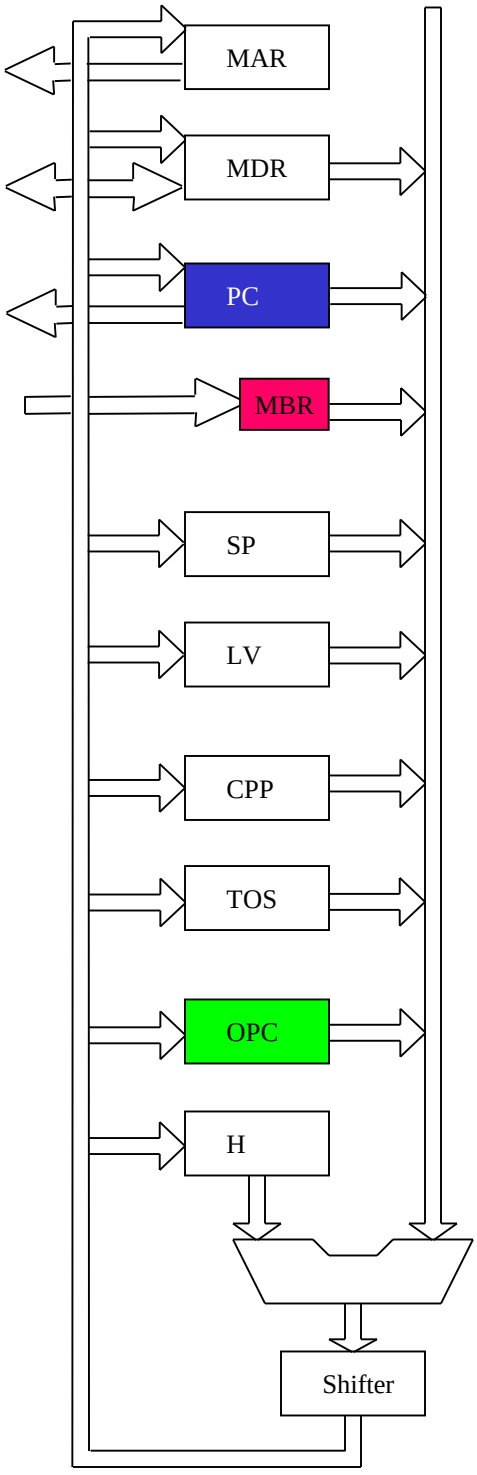

offset2
offset1
PC
A7

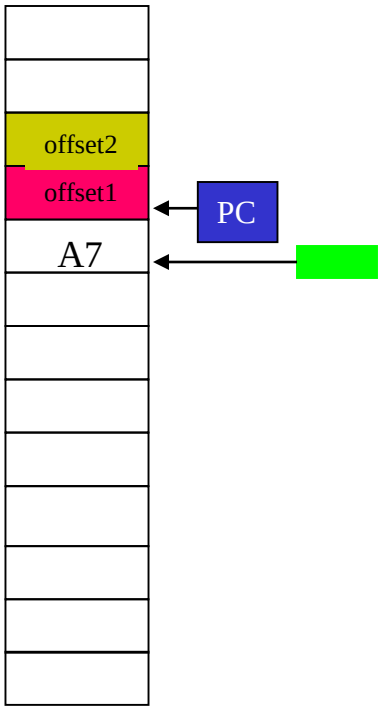

CPP

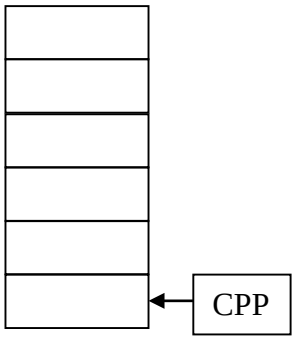

SP
LV

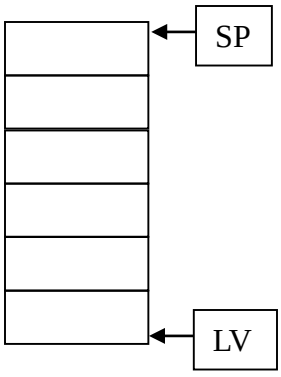
OPC = [PC] - 1

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

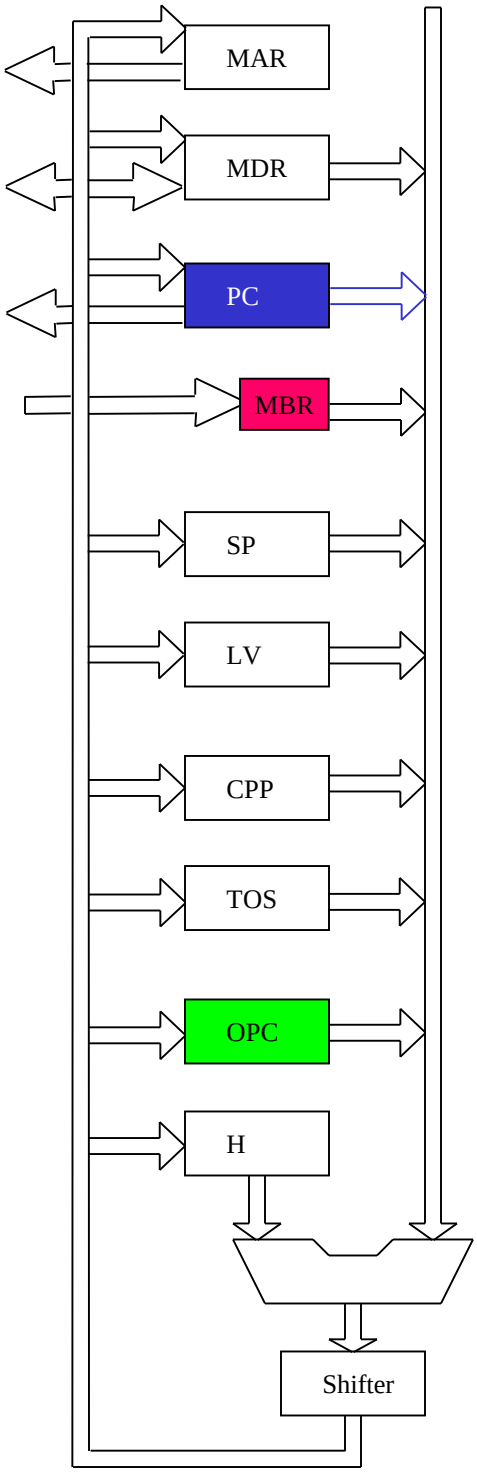

offset2
offset1
PC
A7

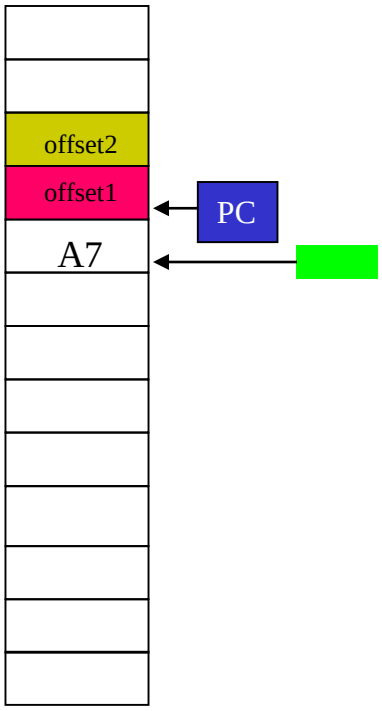

CPP

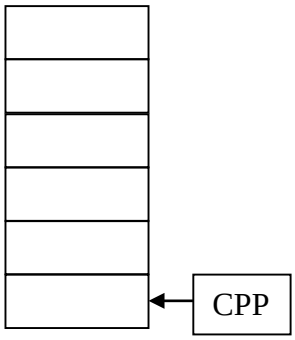

SP
LV

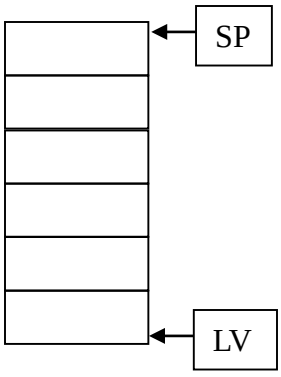
PC = [PC] + 1

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

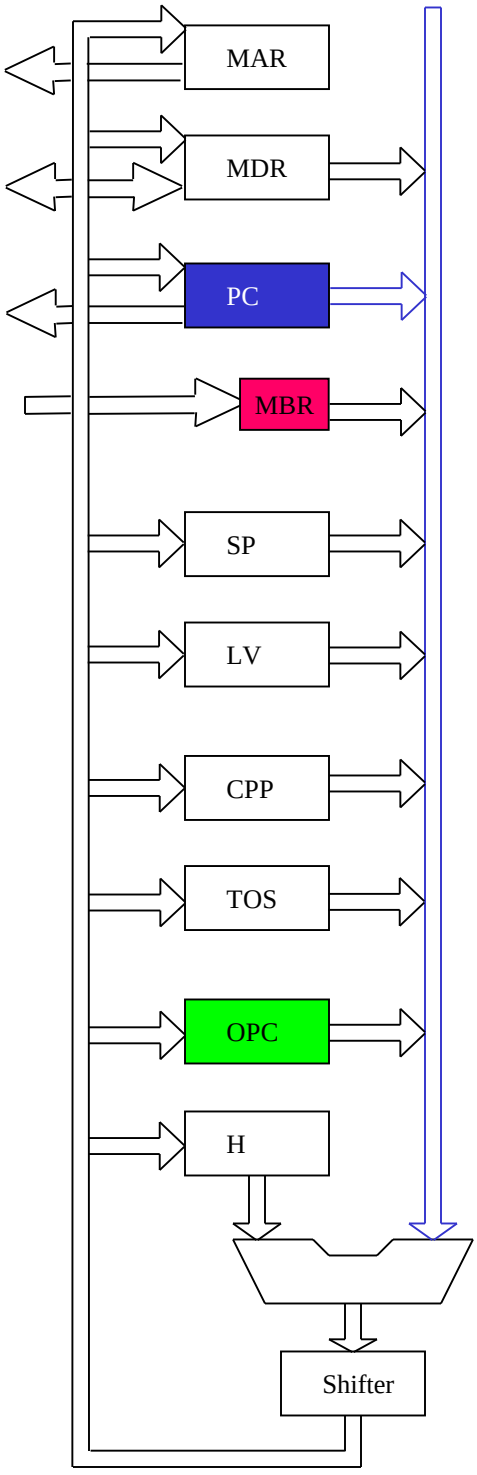

offset2
offset1
PC
A7

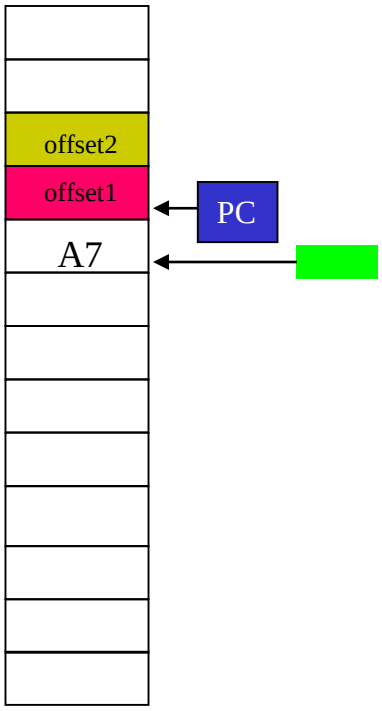

CPP

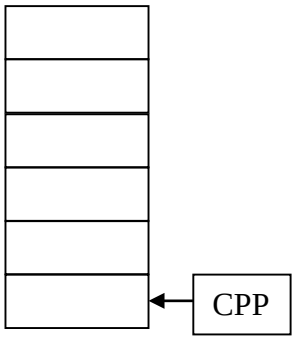

SP
LV

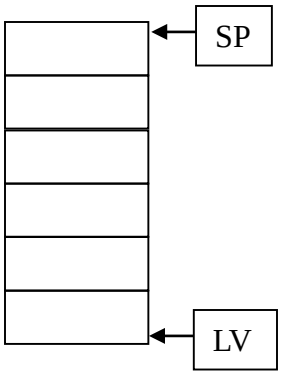
PC = [PC] + 1

GOTO



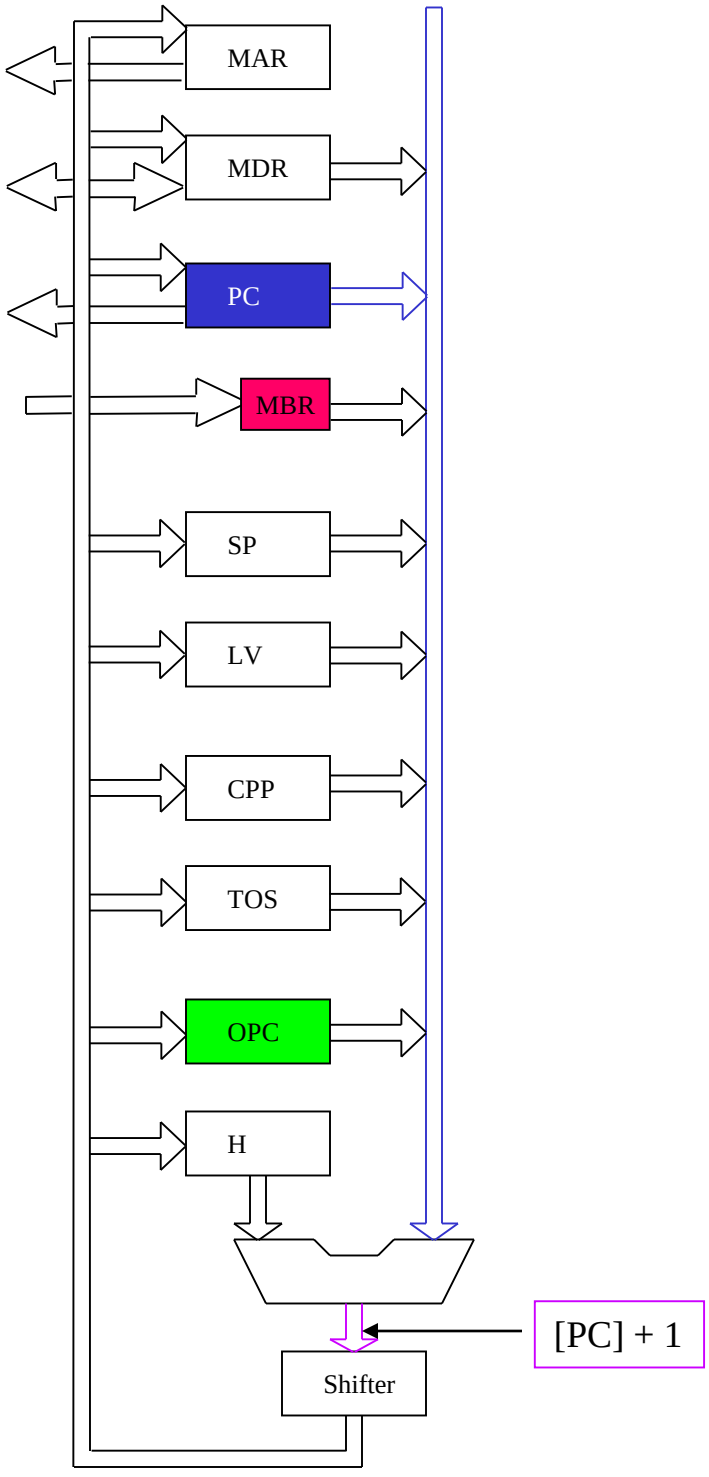


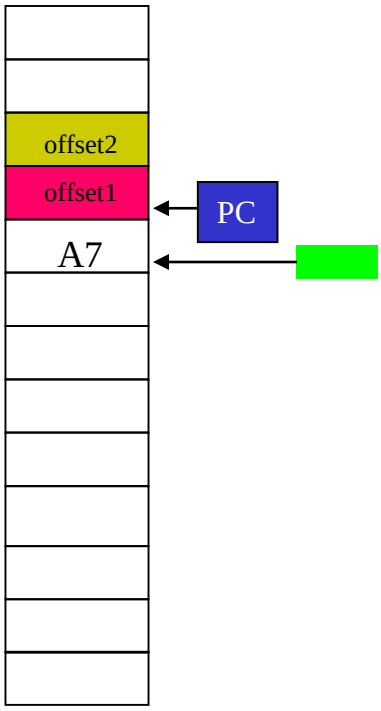


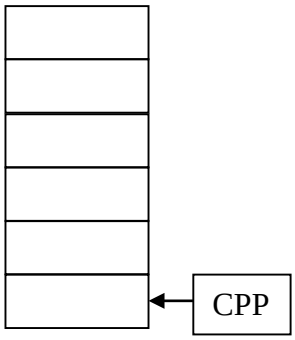


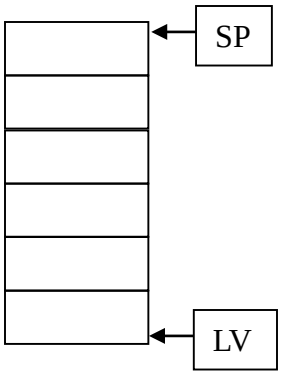
PC = [PC] + 1; fetch

GOTO



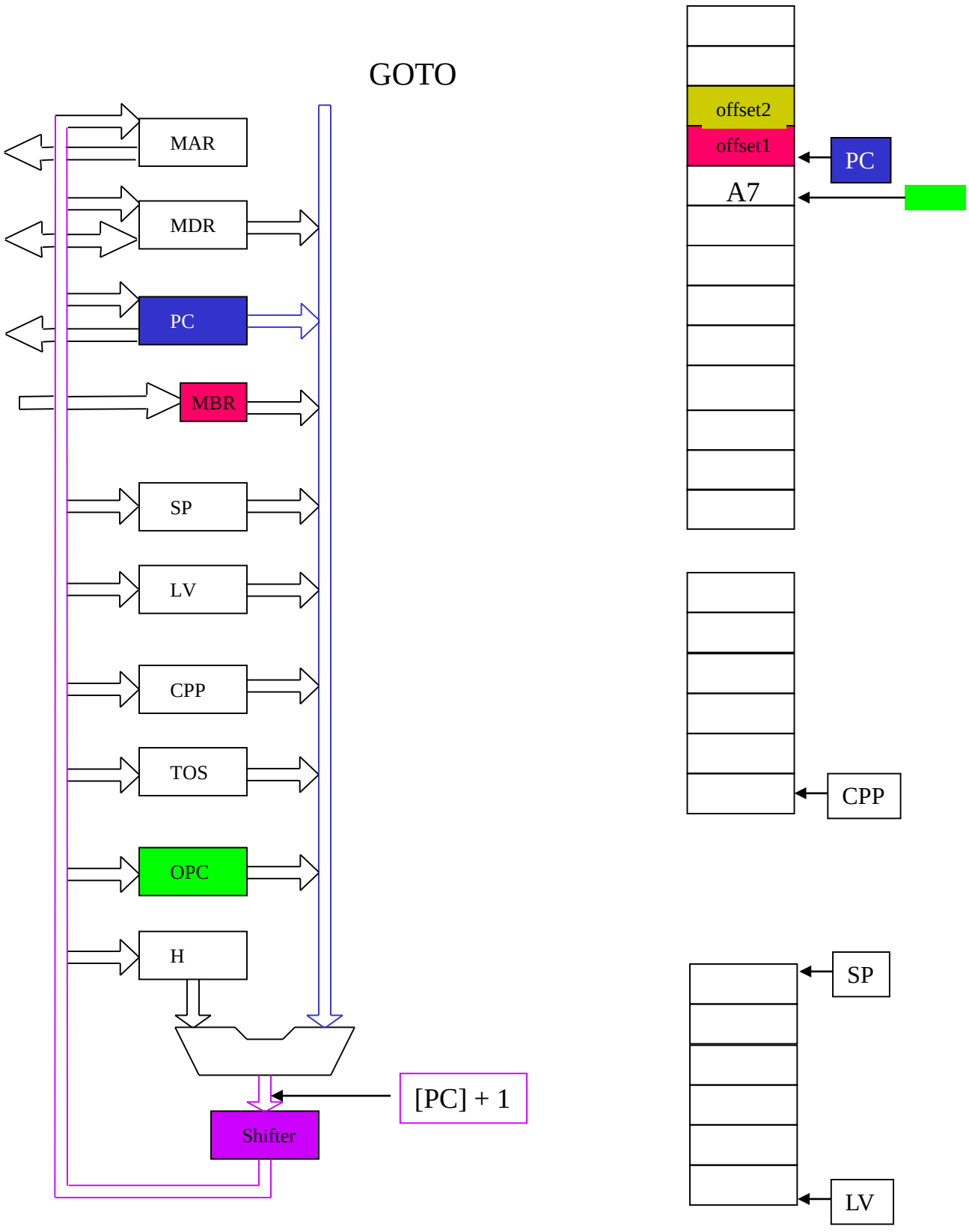






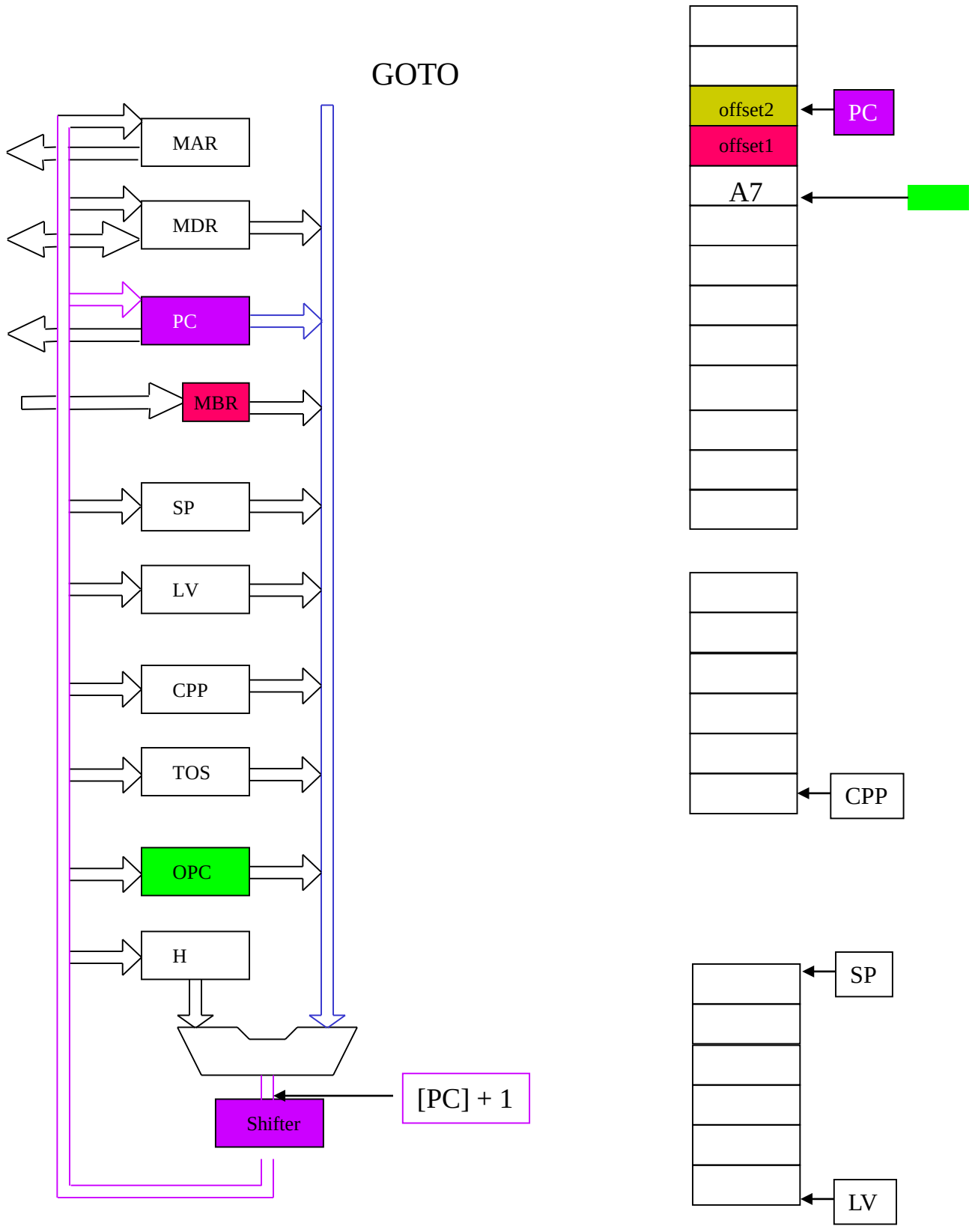
PC = [PC] + 1; fetch

GOTO



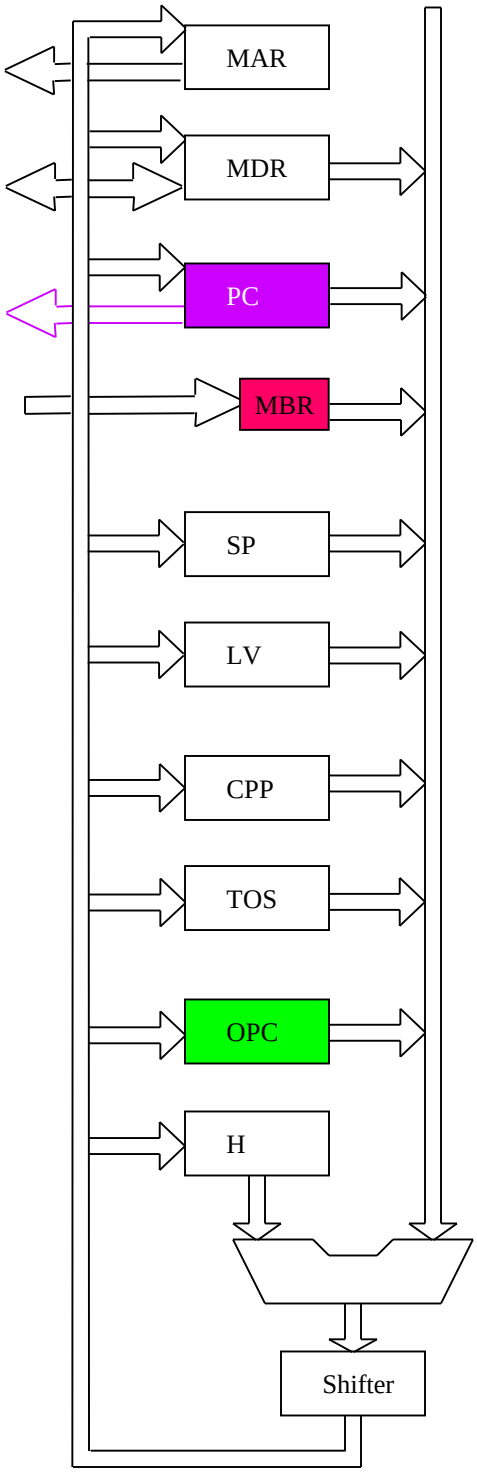
PC = [PC] + 1; fetch

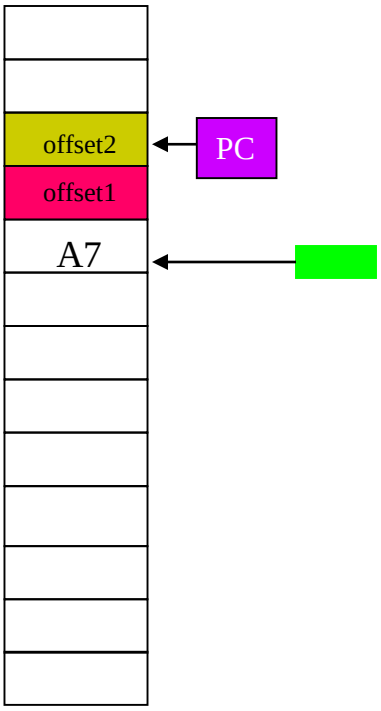
GOTO

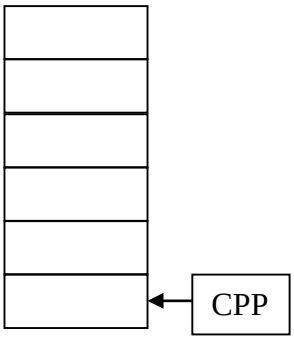


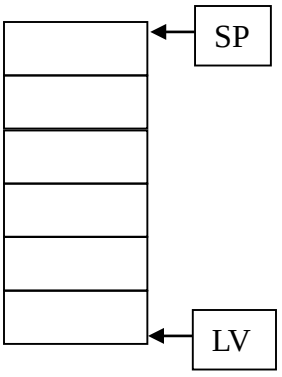
PC = [PC] + 1; fetch

GOTO



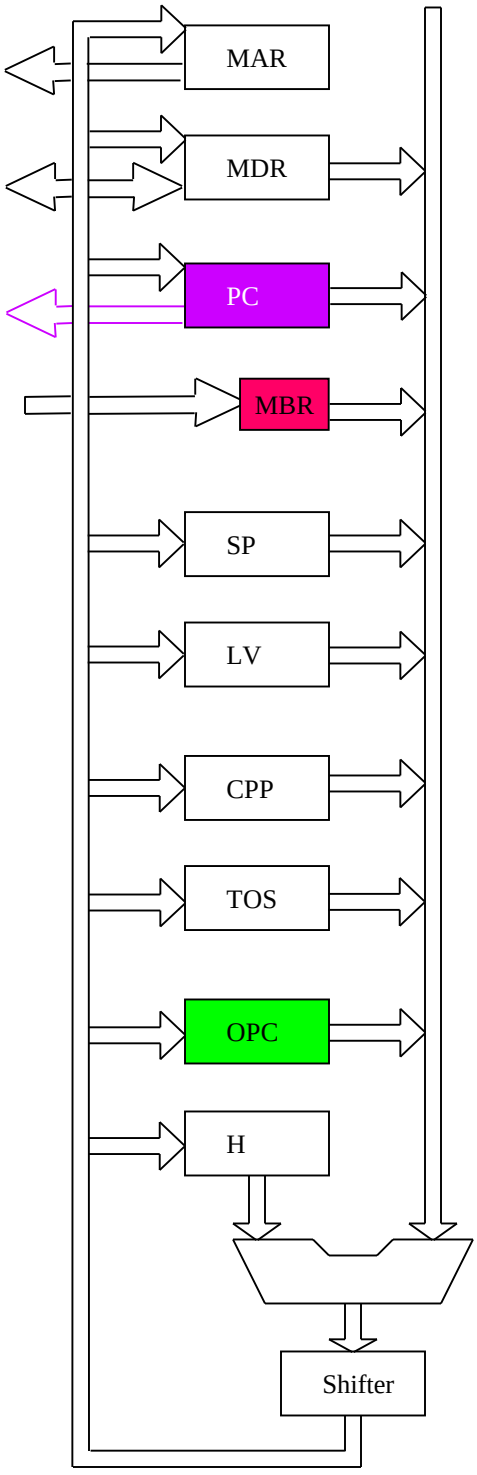


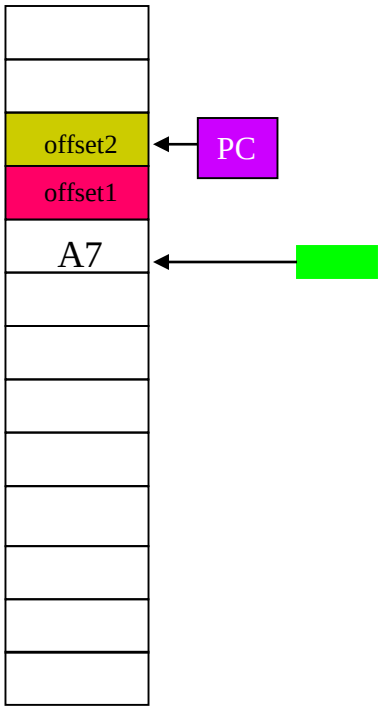


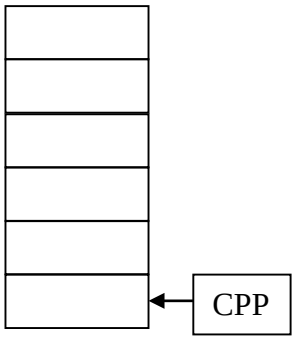


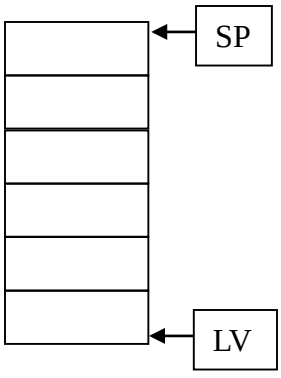
PC = [PC] + 1; **fetch**

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

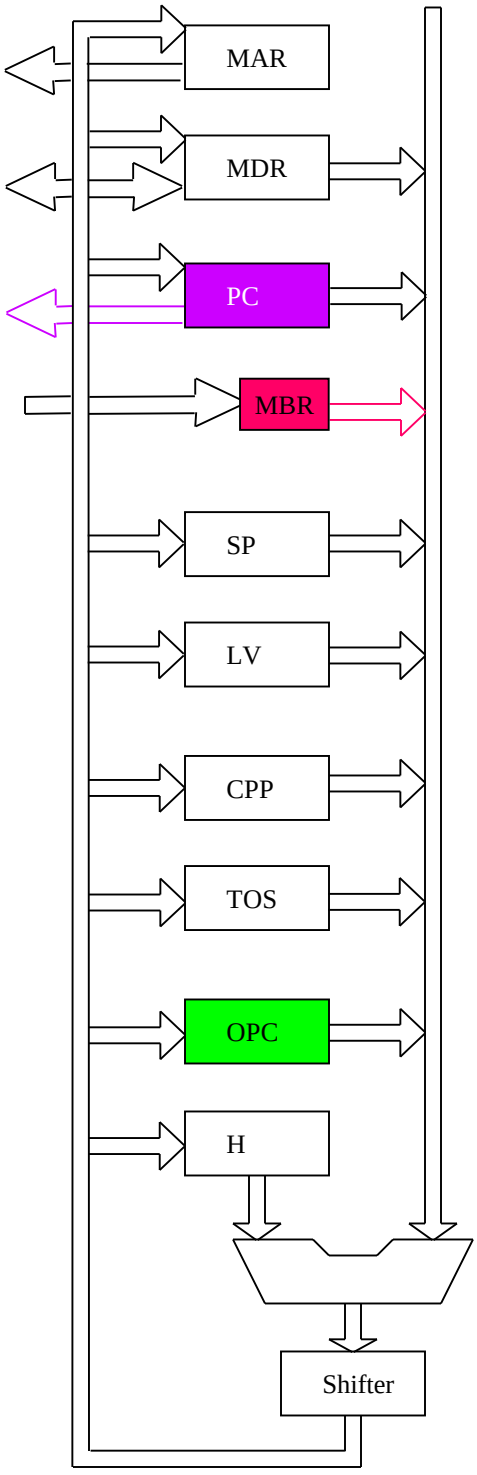

offset2
offset1
A7
PC

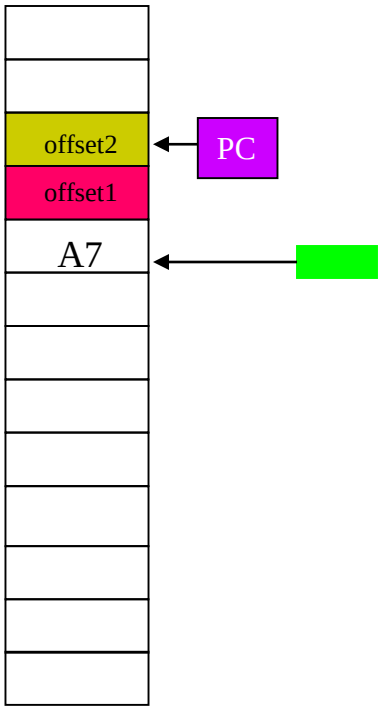

CPP

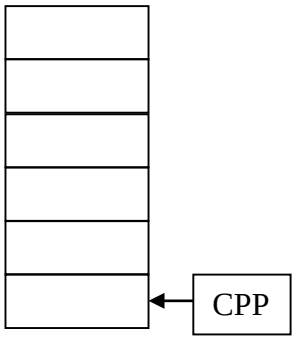

SP
LV

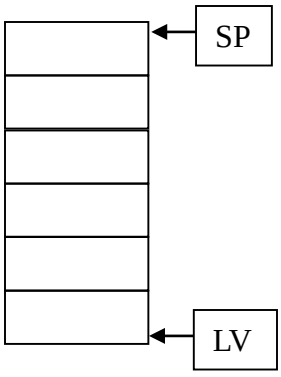
$$H = [MBR]_S << 8$$

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter

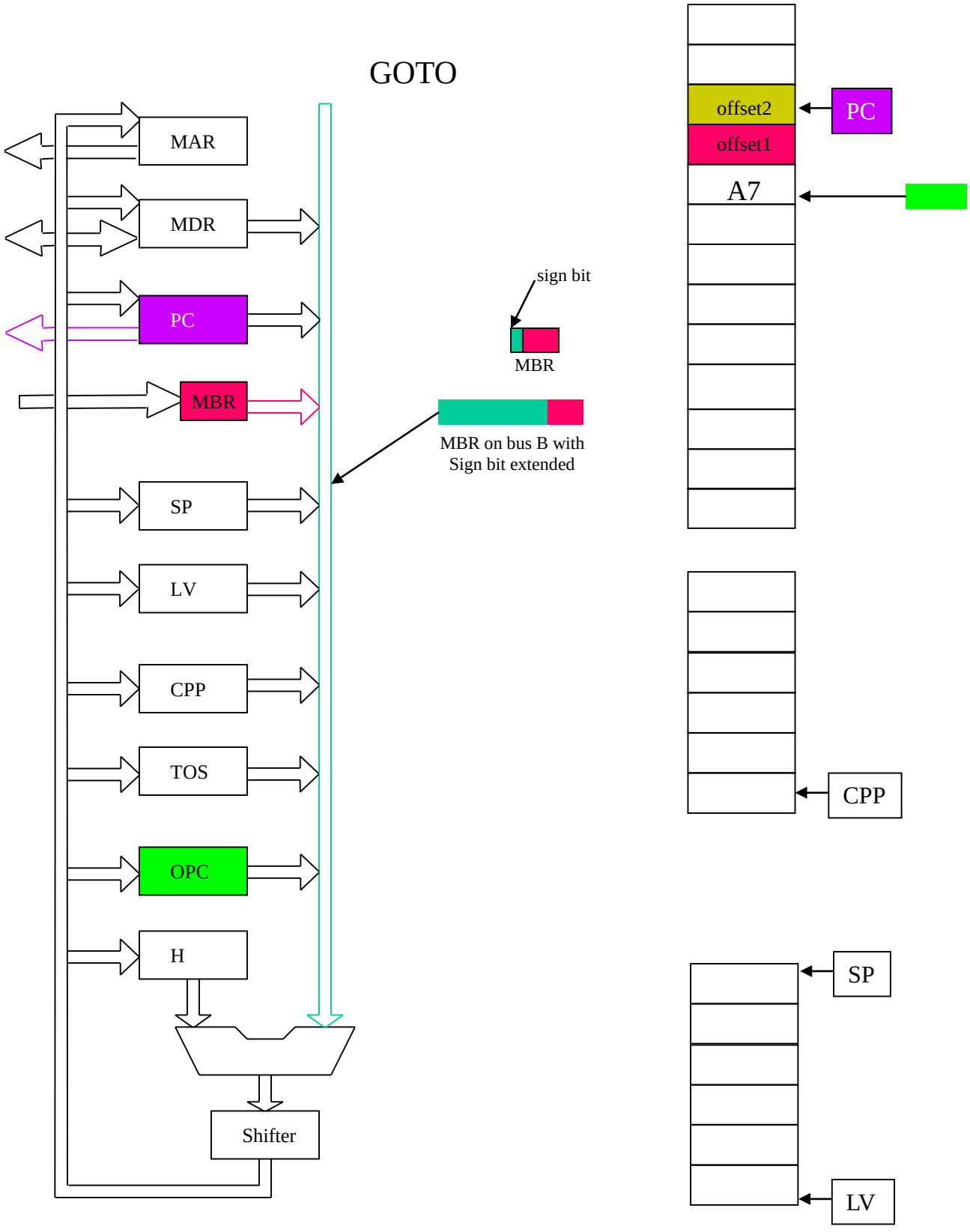

offset2
offset1
A7
PC


CPP


SP
LV

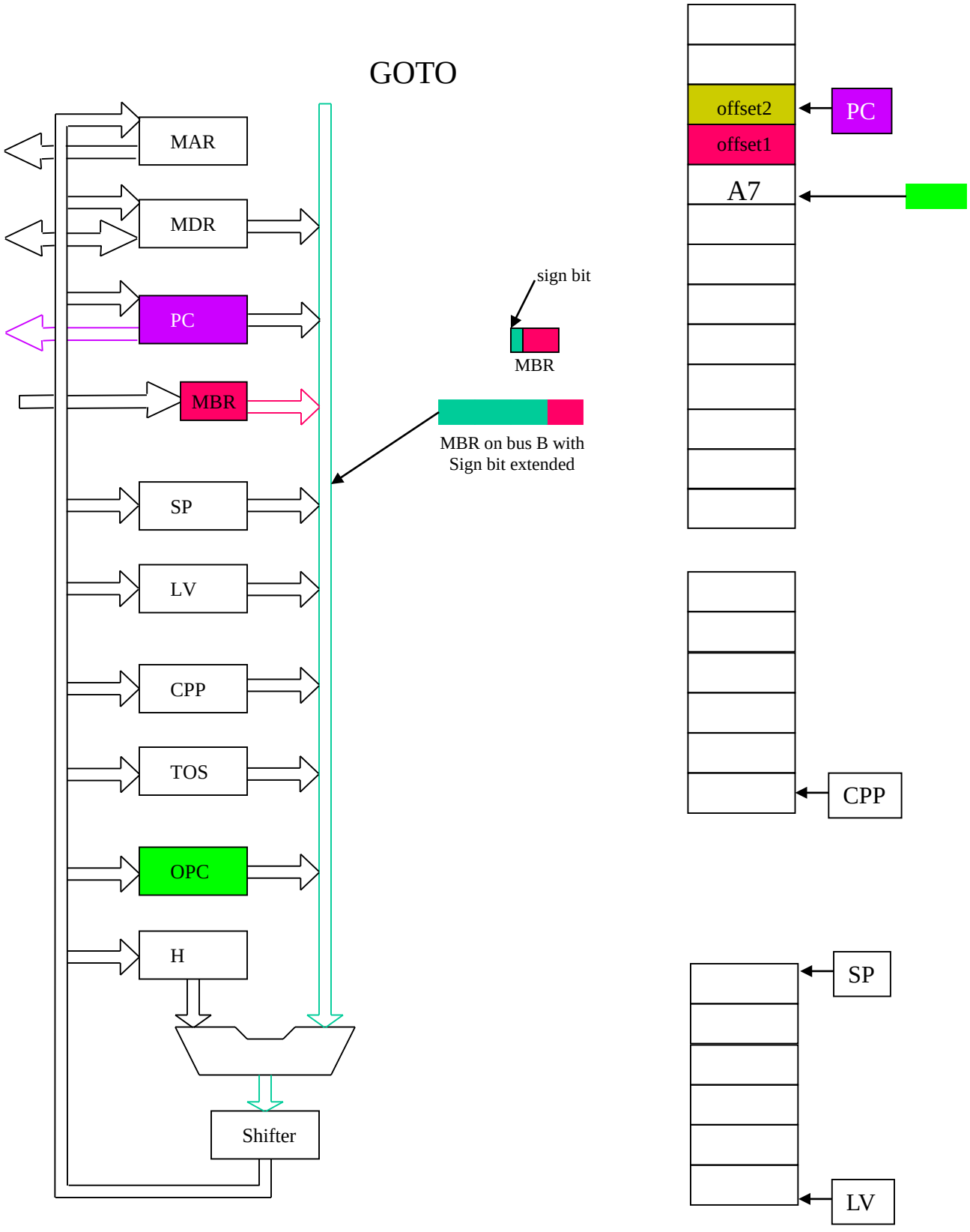
$$H = [MBR]_S << 8$$

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter
sign bit
MBR
MBR on bus B with
Sign bit extended
offset2
offset1
A7
PC
CPP
SP
LV

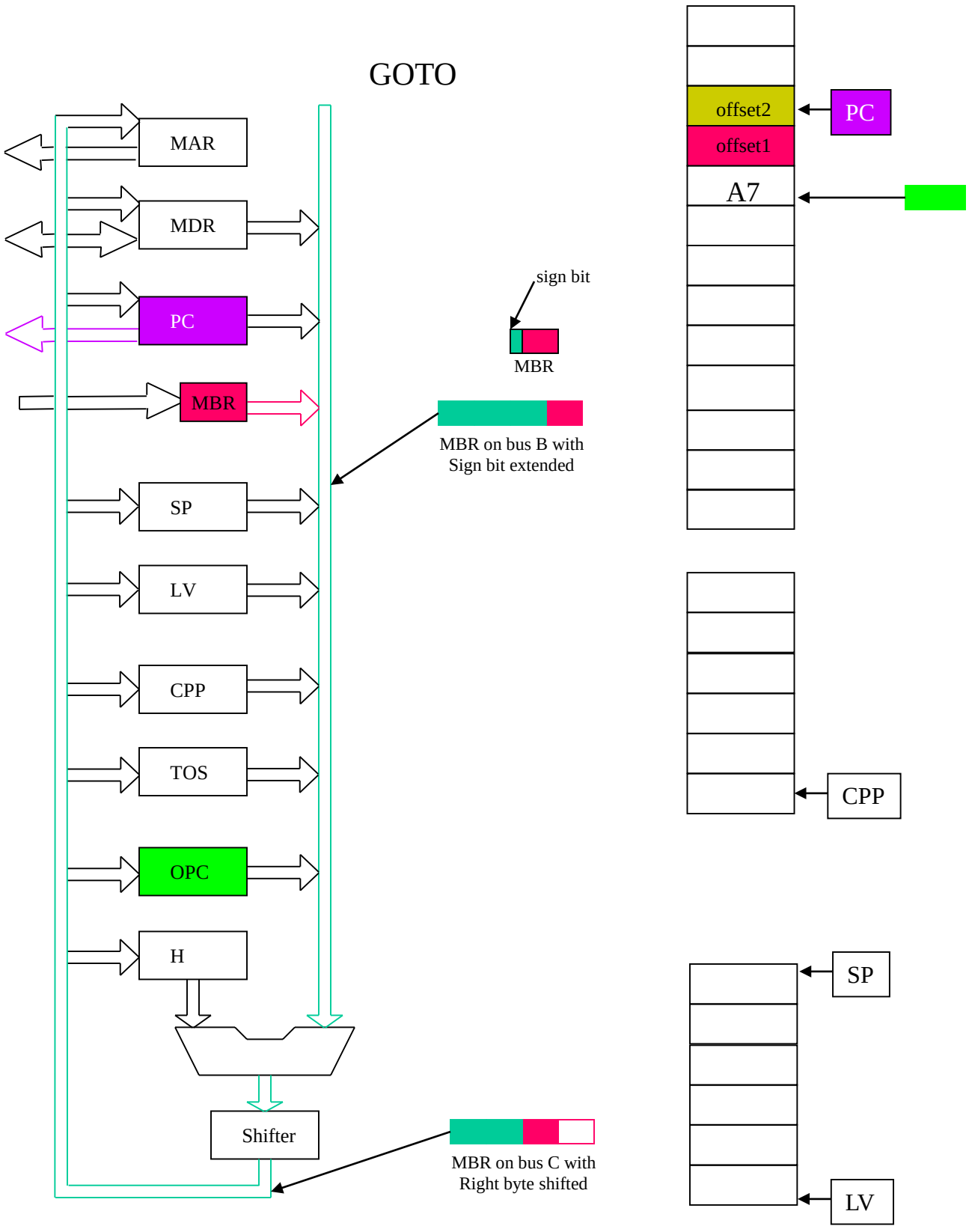
$$H = [MBR]_S << 8$$

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter
sign bit
MBR
MBR on bus B with
Sign bit extended
offset2
offset1
A7
PC
CPP
SP
LV

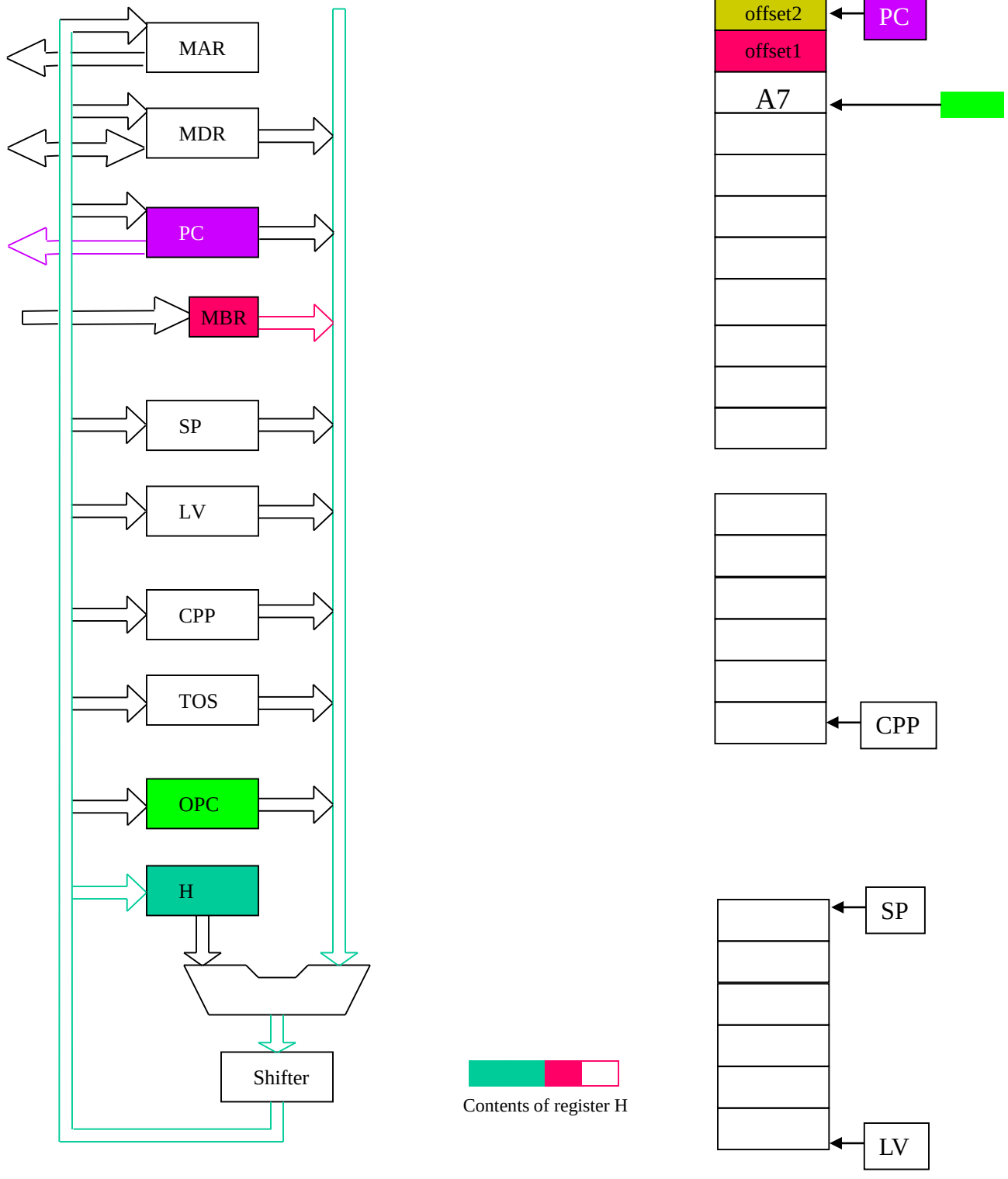
$$H = [MBR]_S << 8$$

GOTO


MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter
sign bit
MBR
MBR on bus B with
Sign bit extended
MBR on bus C with
Right byte shifted
offset2
offset1
A7
PC
CPP
SP
LV

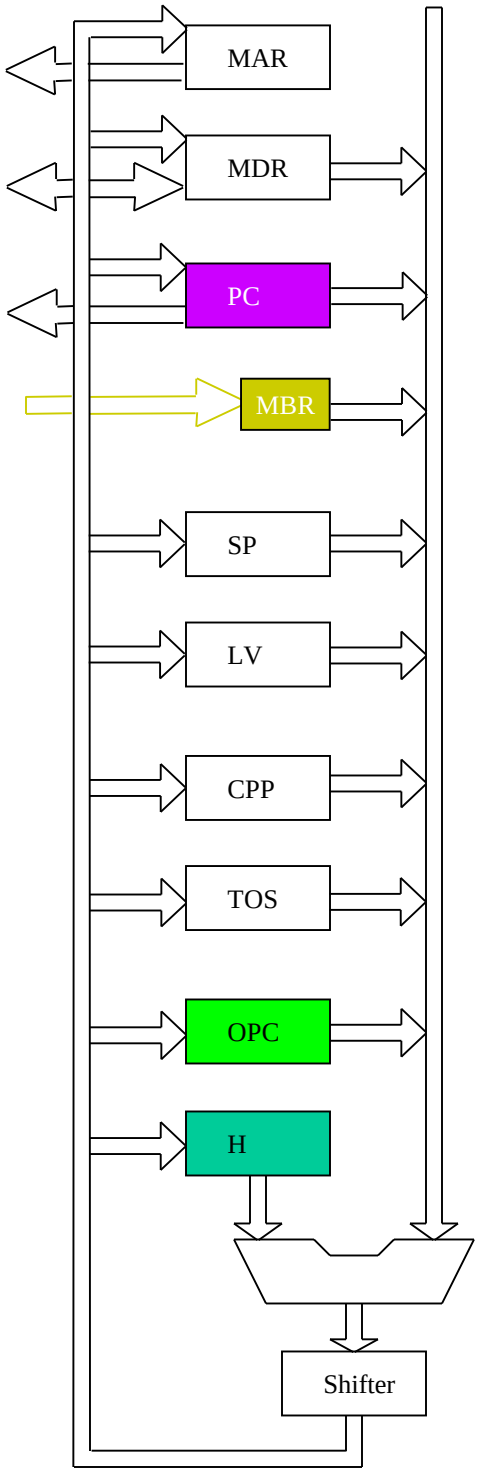
$$H = [MBR]_S << 8$$

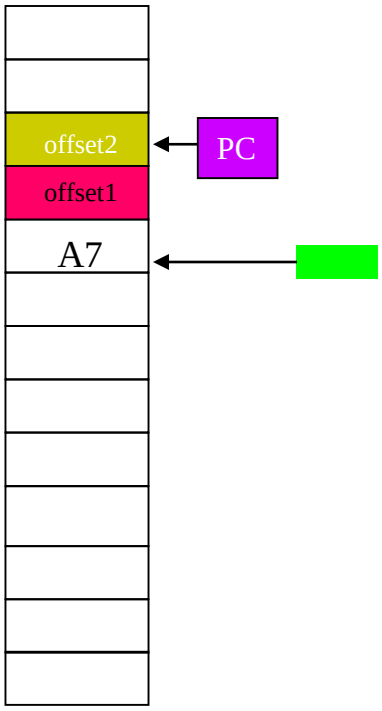
GOTO

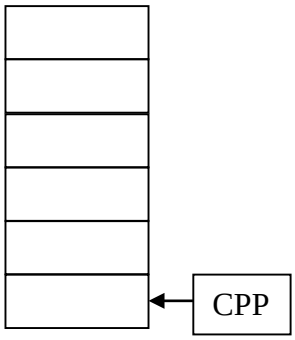

MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter
offset2
offset1
A7
PC
CPP
SP
LV
Contents of register H

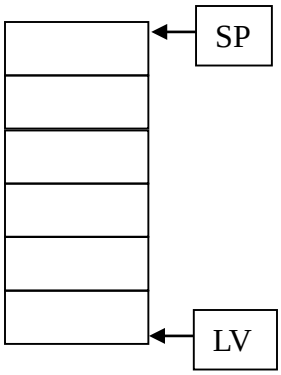
$$H = [MBR]_S << 8$$

GOTO







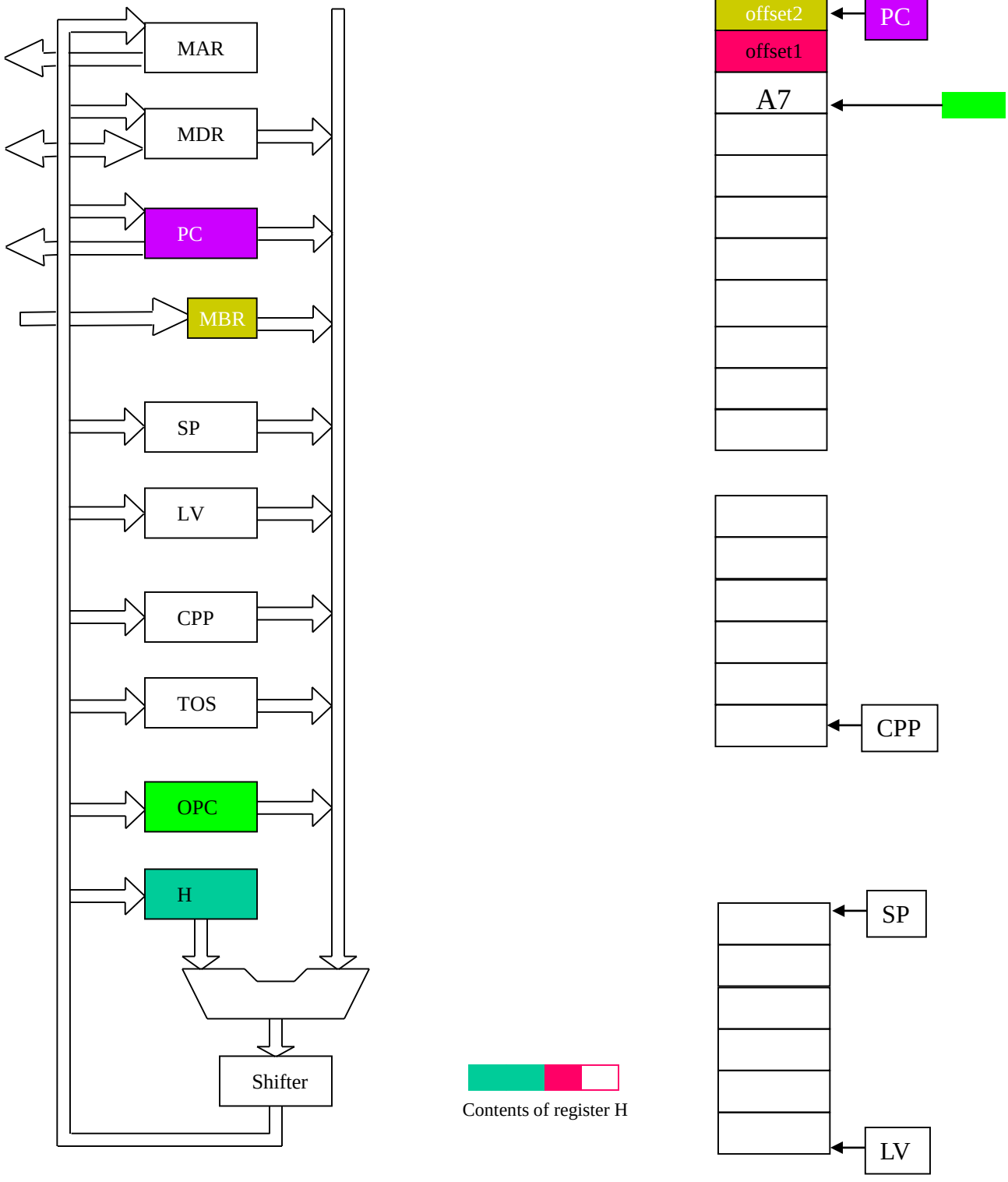


Contents of register H

$H = [MBR]_S << 8$

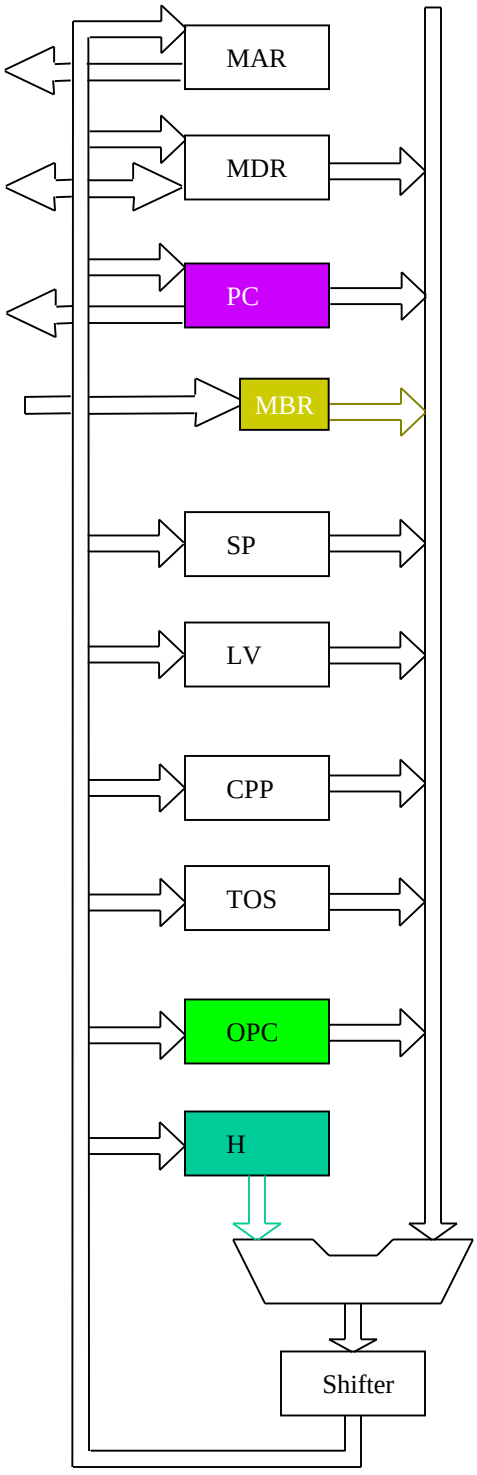
(previous fetch completes)

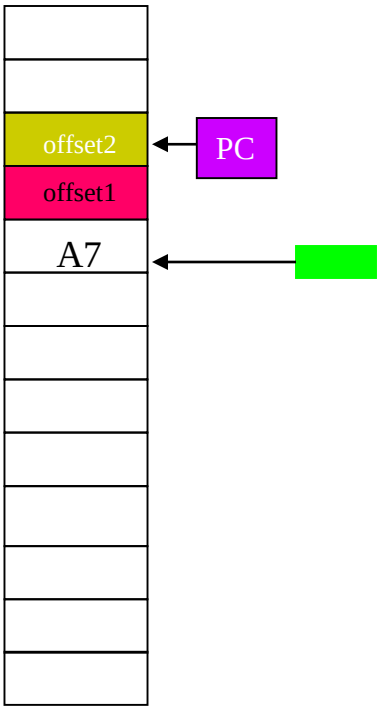
GOTO

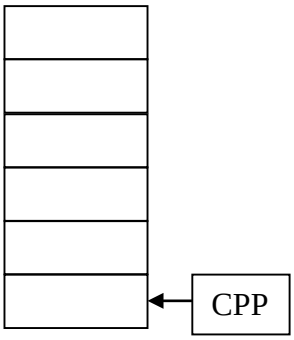


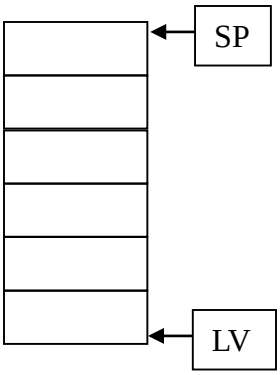
$$H = [H] \text{ OR } [MBR]_U$$

GOTO



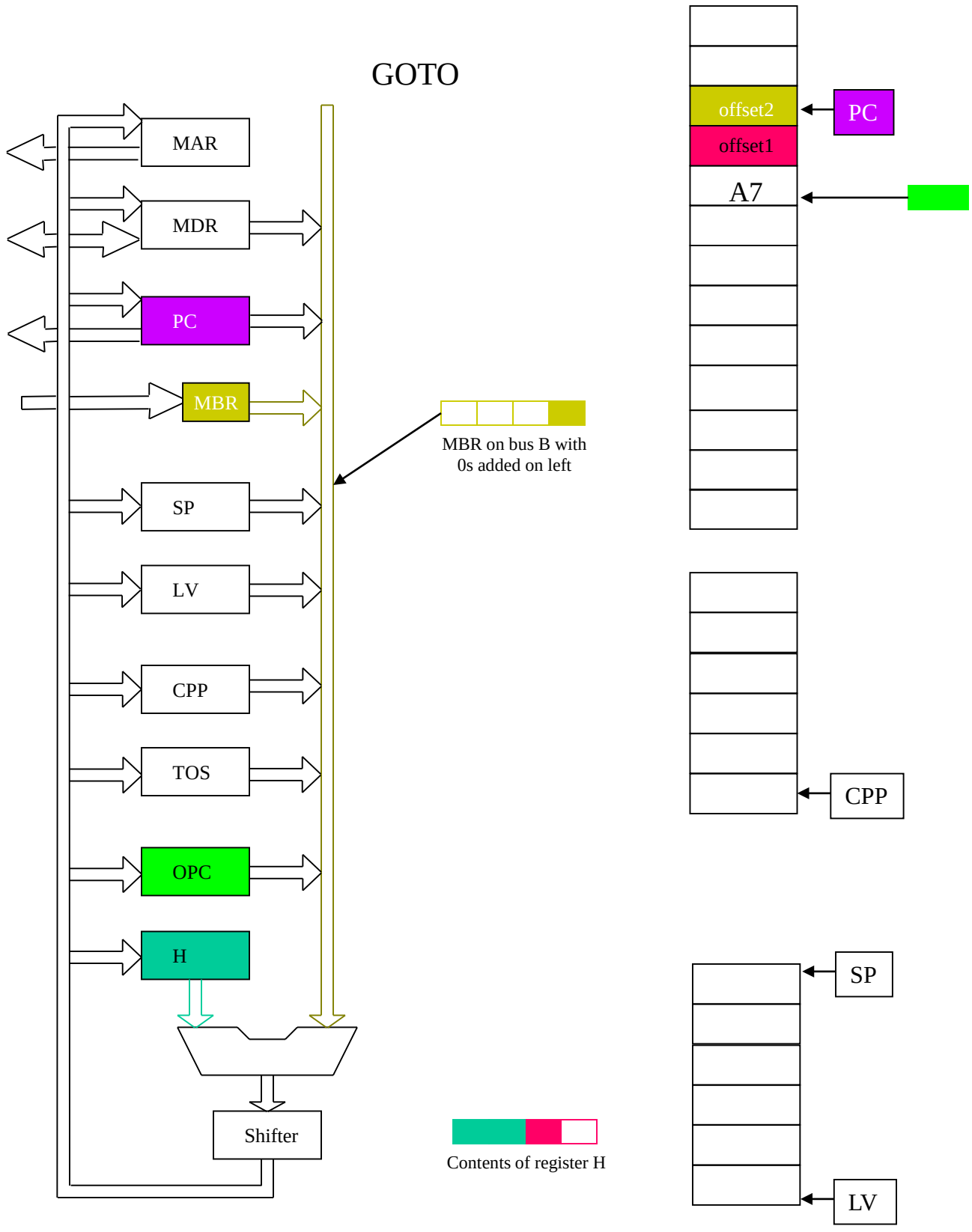


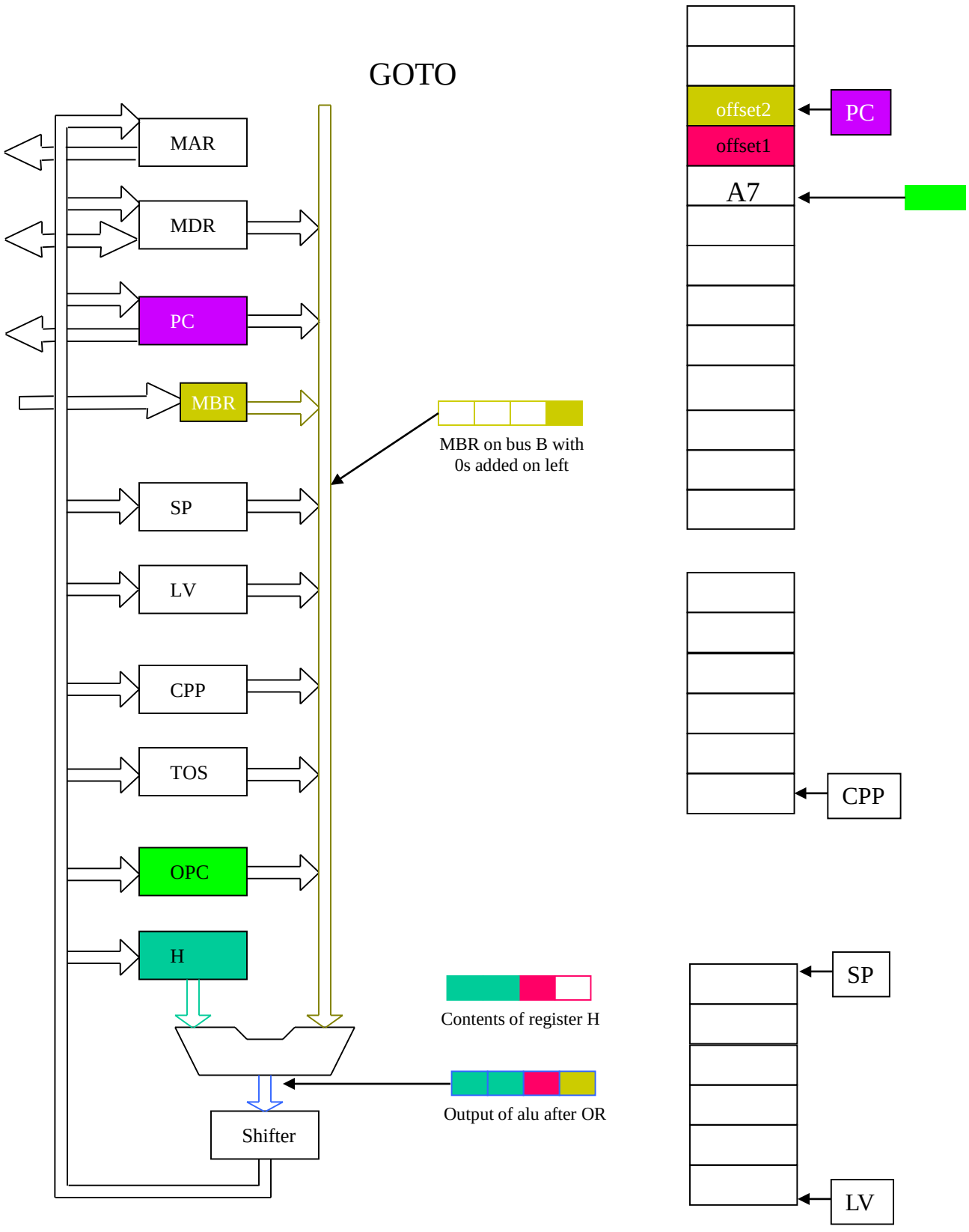




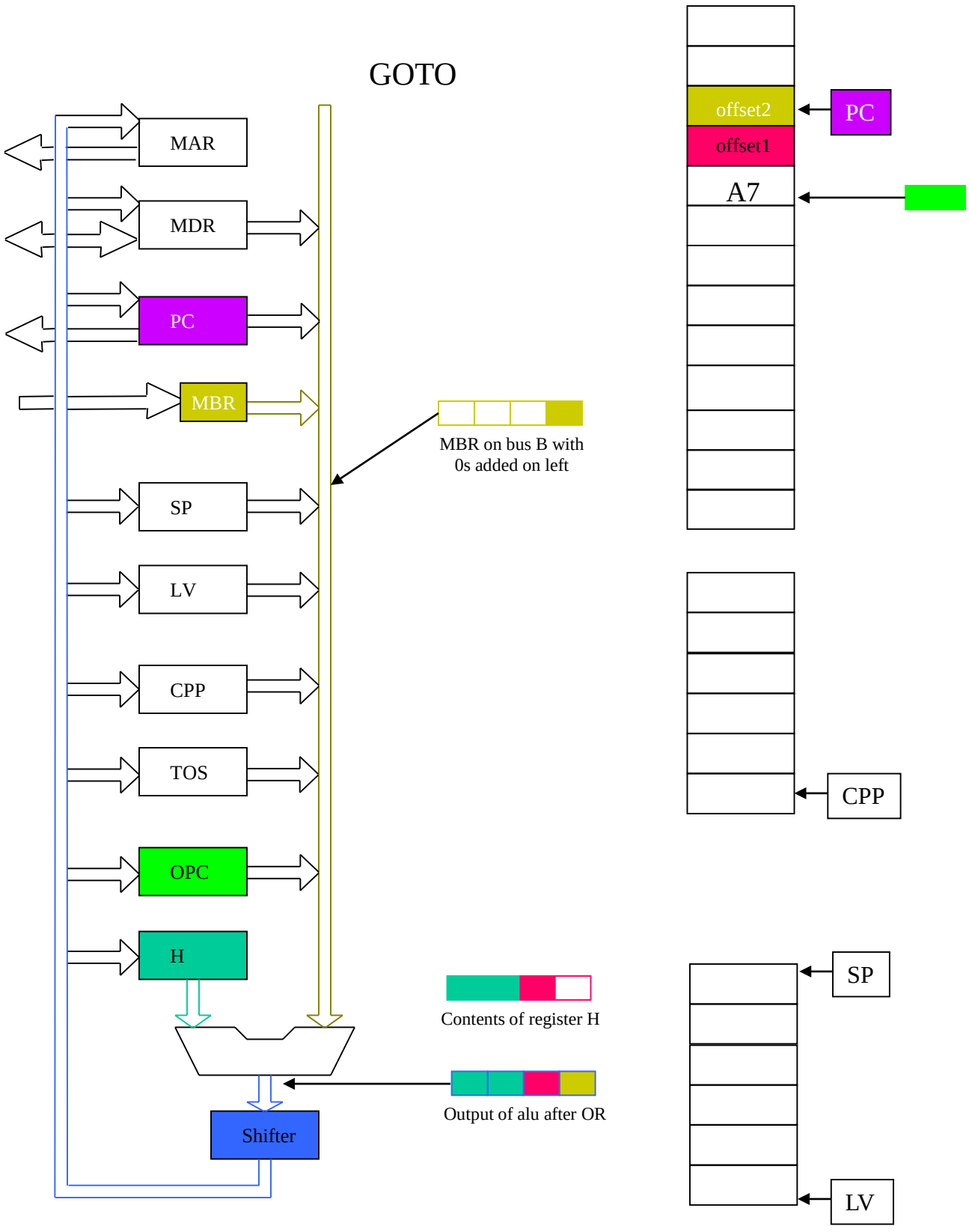
Contents of register H

$$H = [H] \text{ OR } [MBR]_U$$

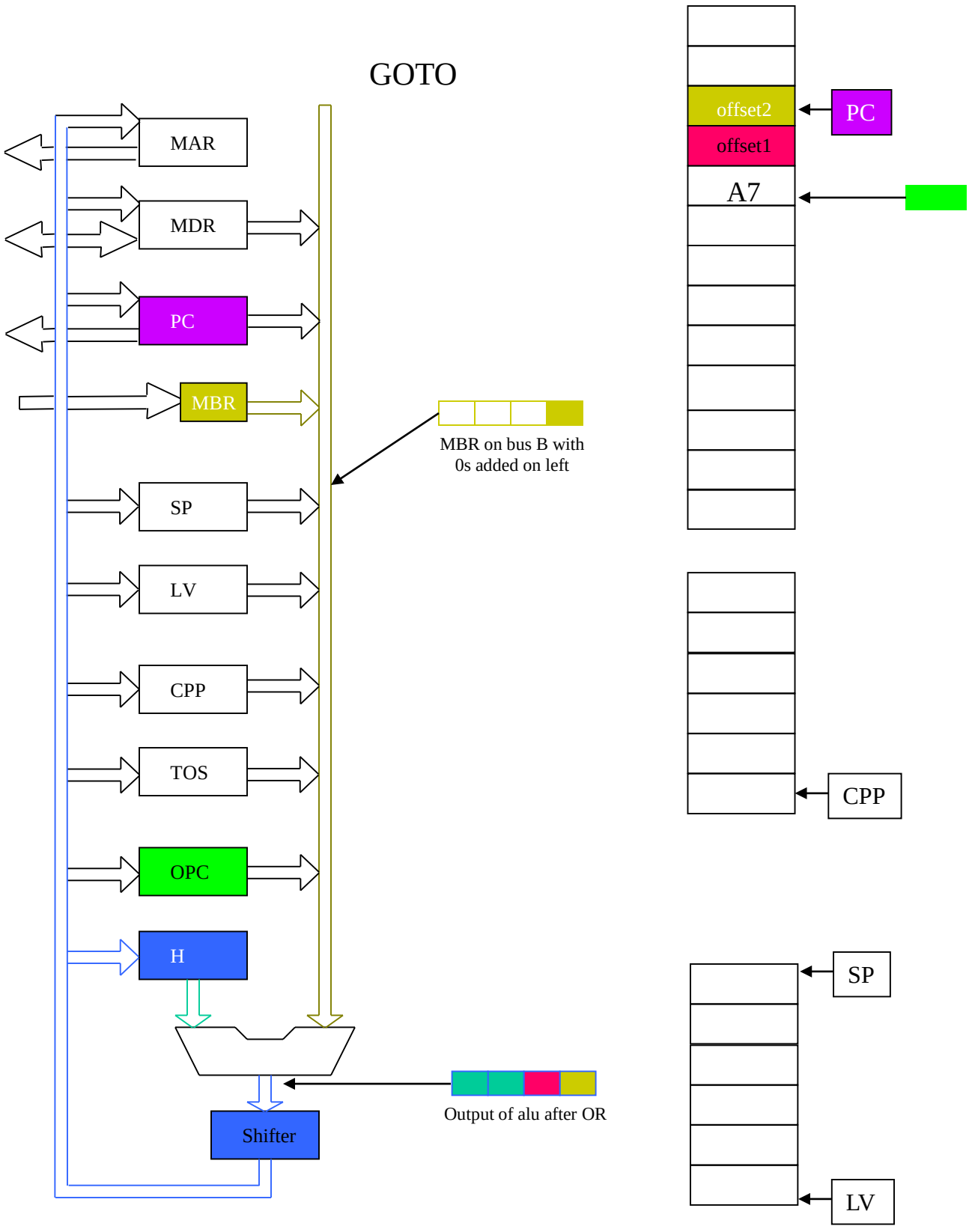

GOTO
MAR
MDR
PC
MBR
SP
LV
CPP
TOS
OPC
H
Shifter
MBR on bus B with
0s added on left
Contents of register H
offset2
offset1
A7
PC
CPP
SP
LV
H = [H] OR [MBR]U



$$H = [H] \text{ OR } [MBR]_U$$

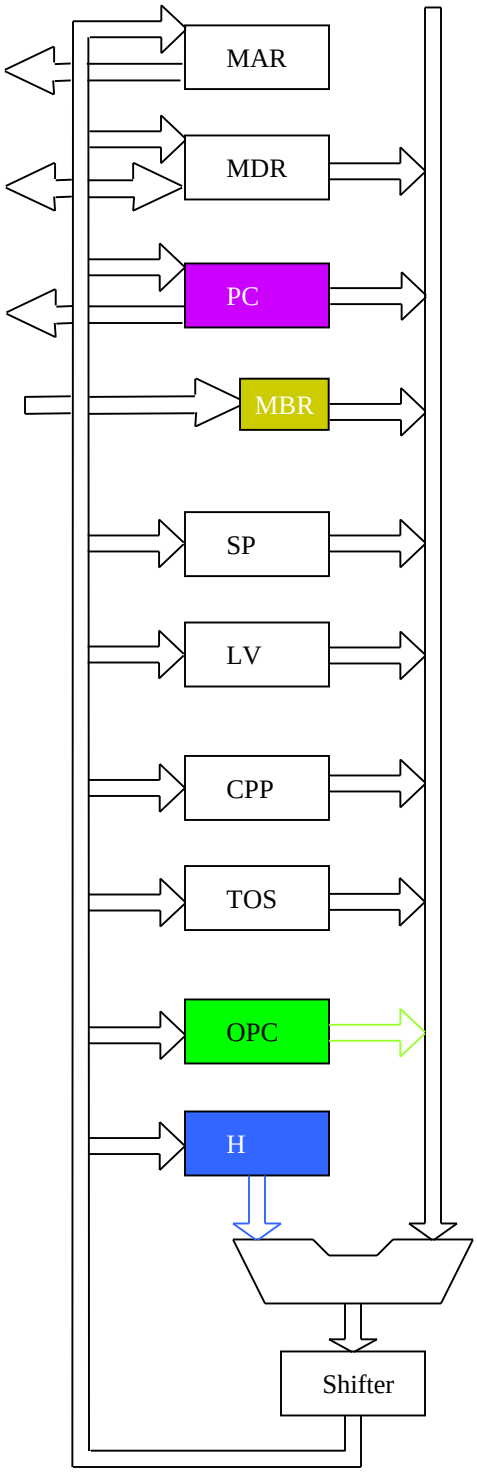


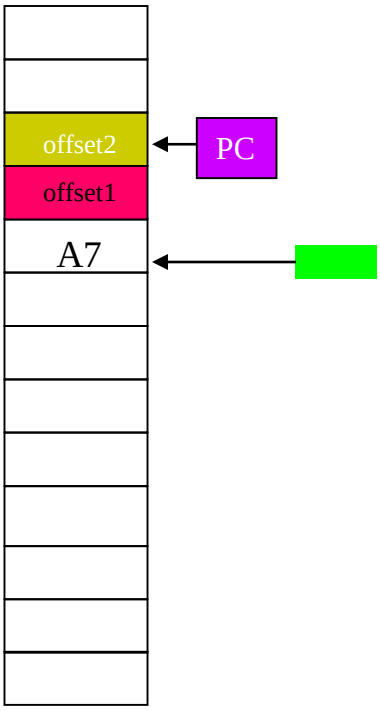
$$H = [H] \text{ OR } [MBR]_U$$

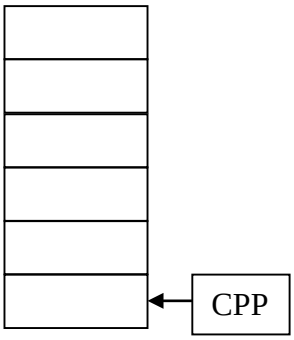


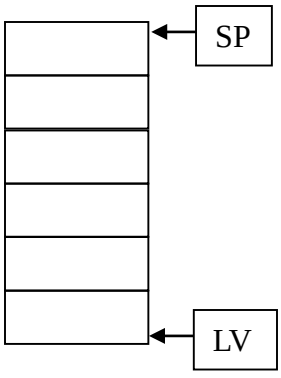
$$H = [H] \text{ OR } [MBR]_U$$

GOTO





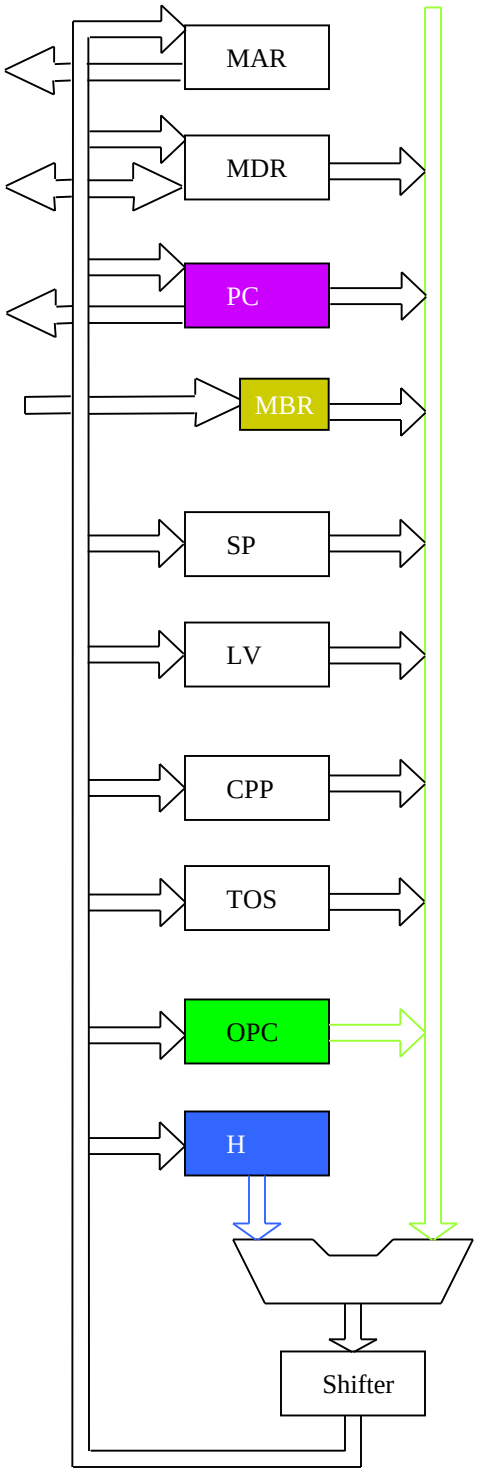


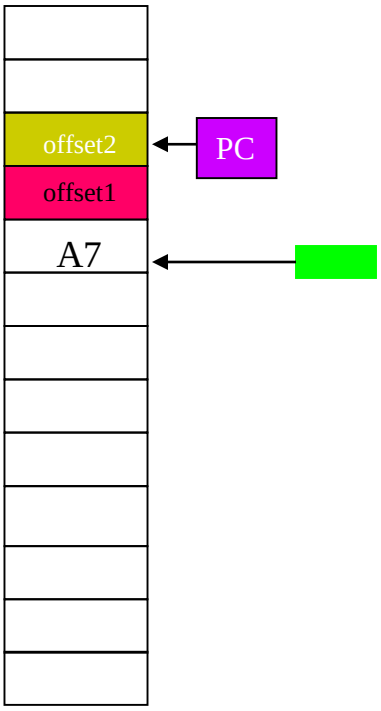


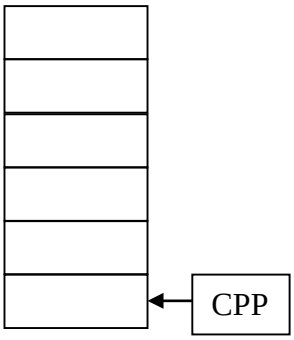
Contents of register H
(has full offset as 16-bit signed value)

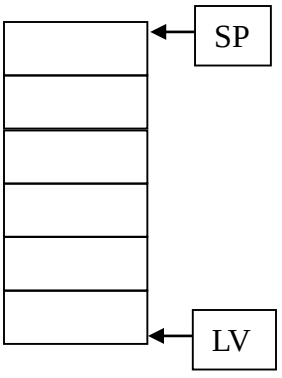
PC = [H] + [OPC]; fetch

GOTO





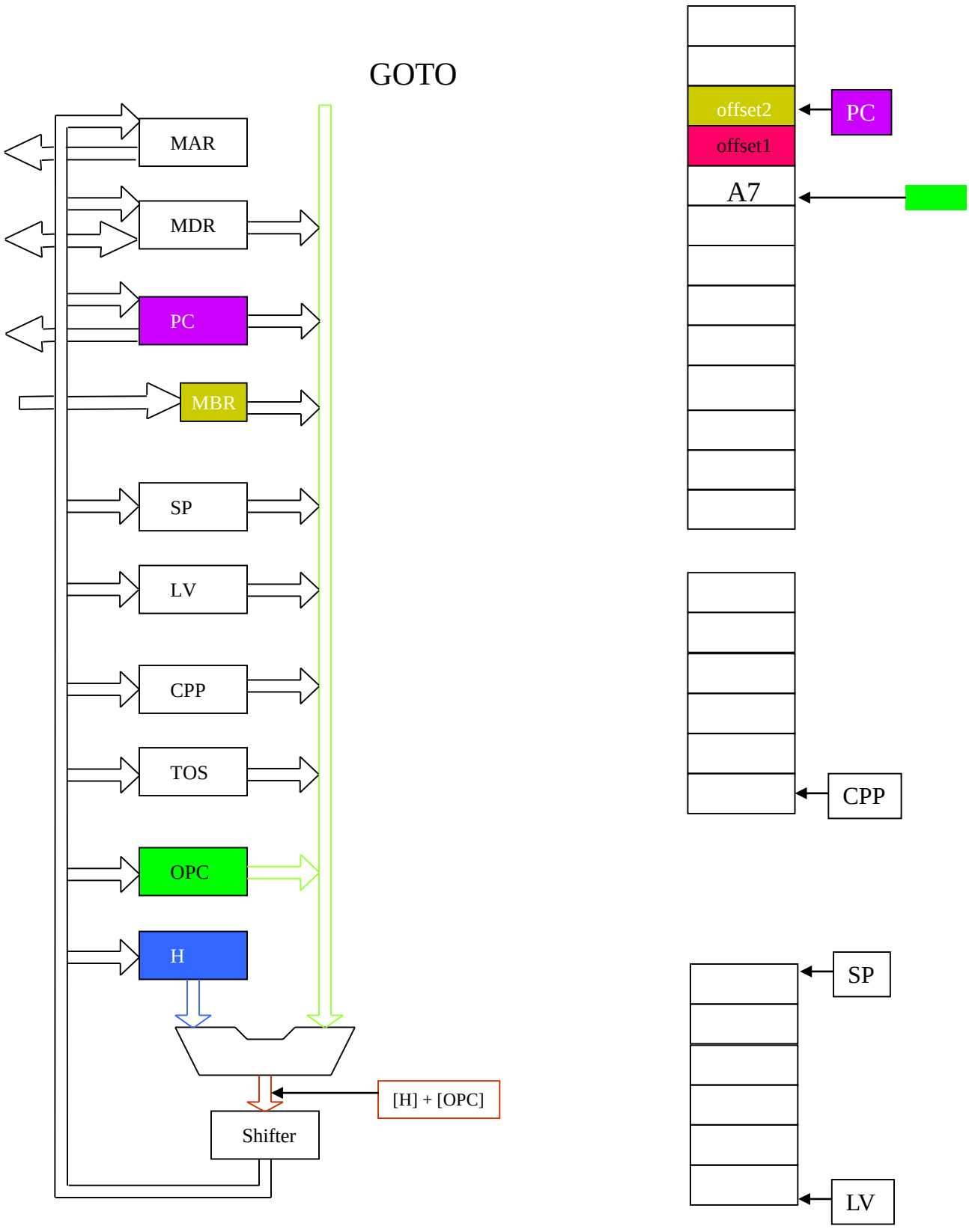




Contents of register H
(has full offset as 16-bit signed value)

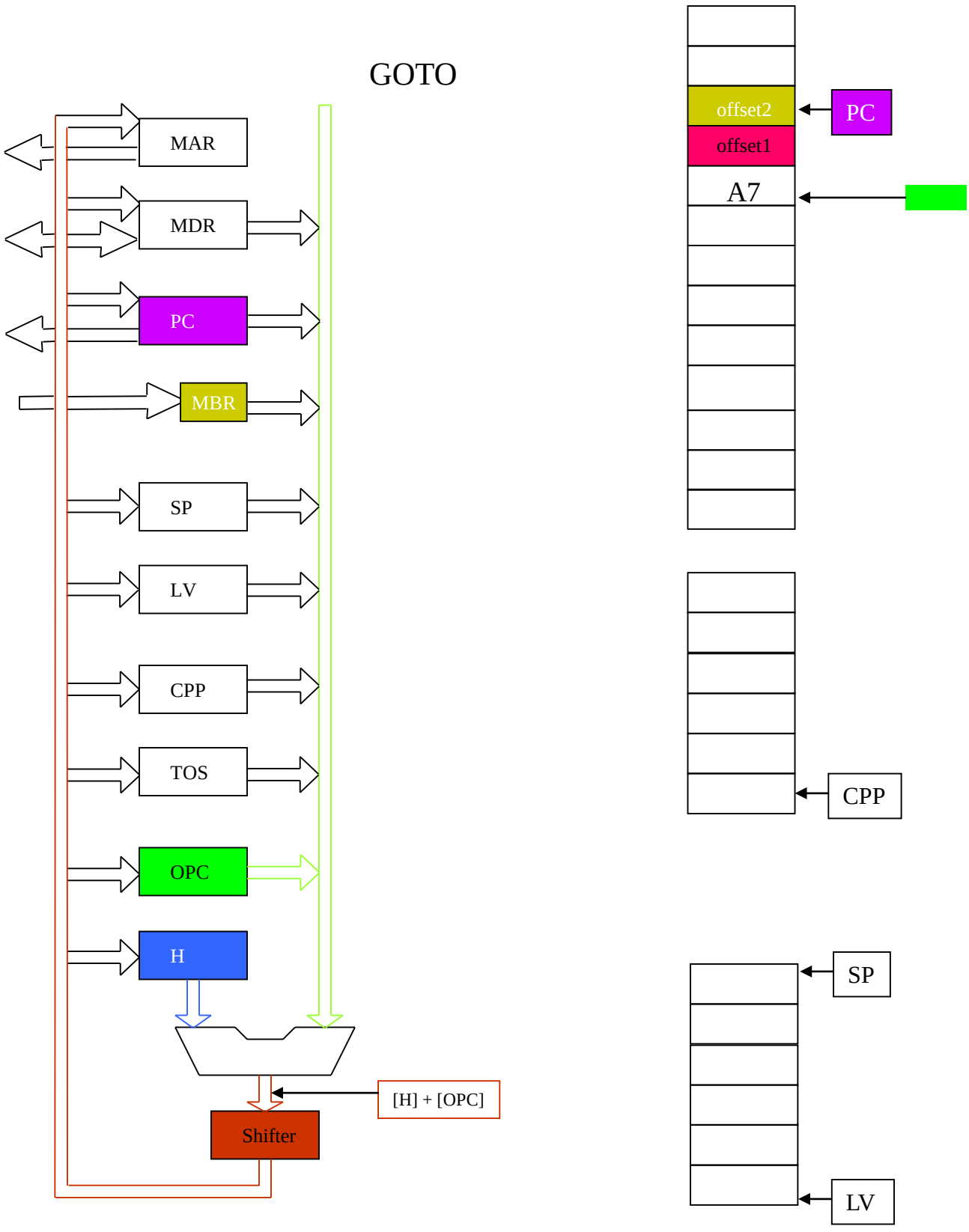
PC = [H] + [OPC]; fetch

GOTO



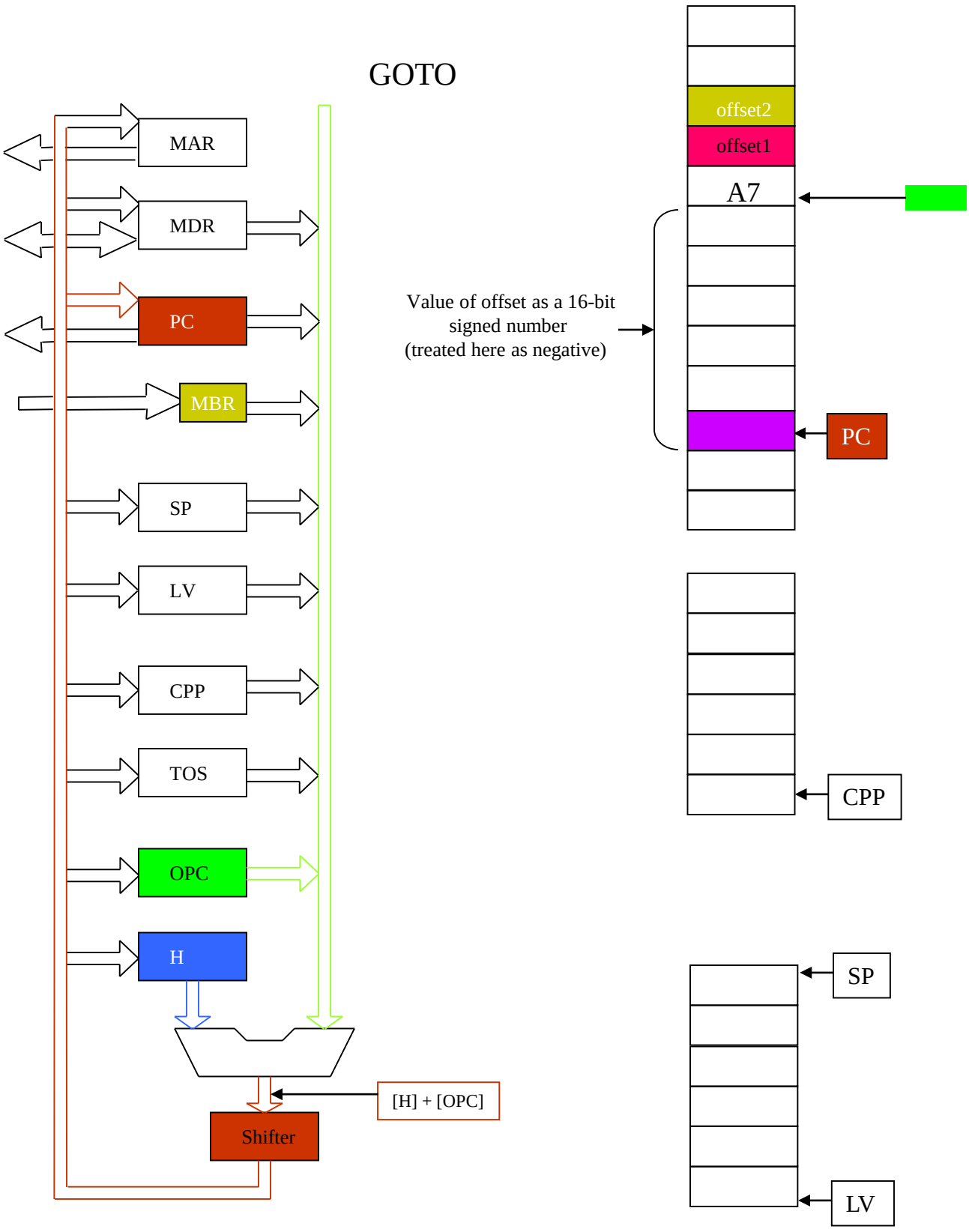
PC = [H] + [OPC]; fetch

GOTO



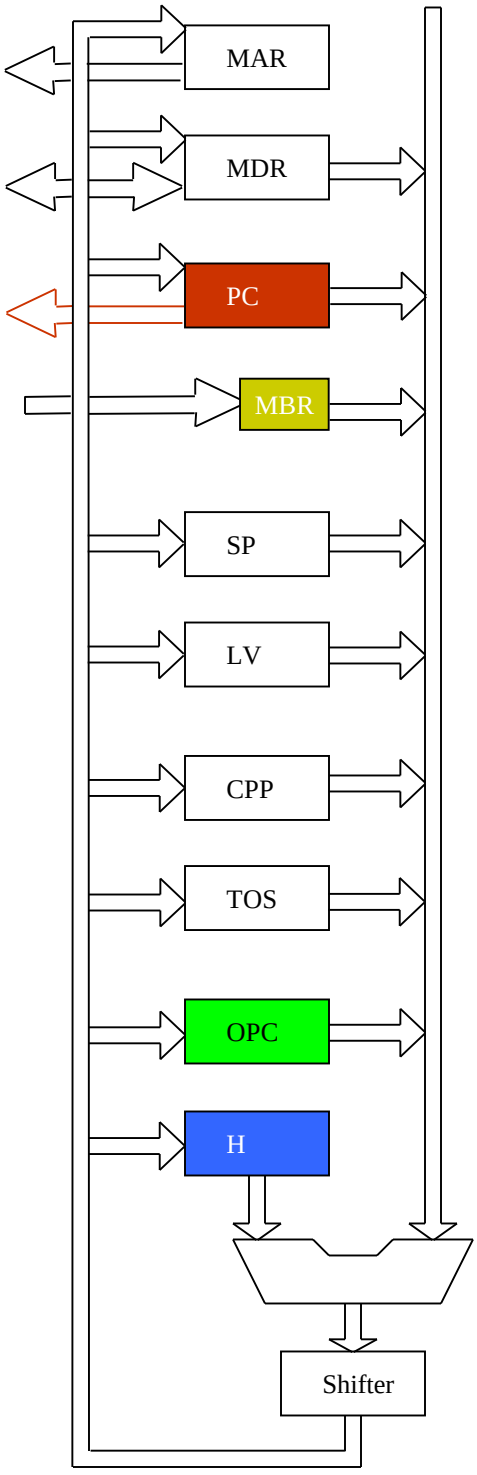
PC = [H] + [OPC]; fetch

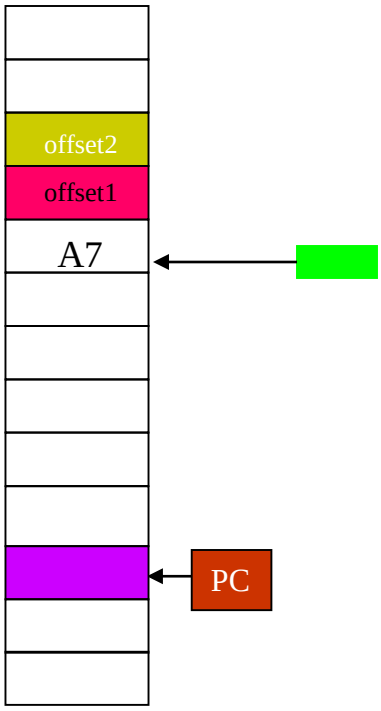
GOTO

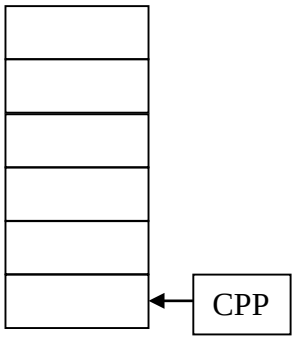


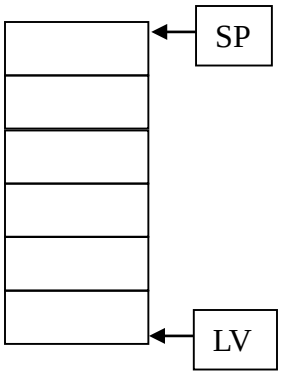
PC = [H] + [OPC]; fetch

GOTO



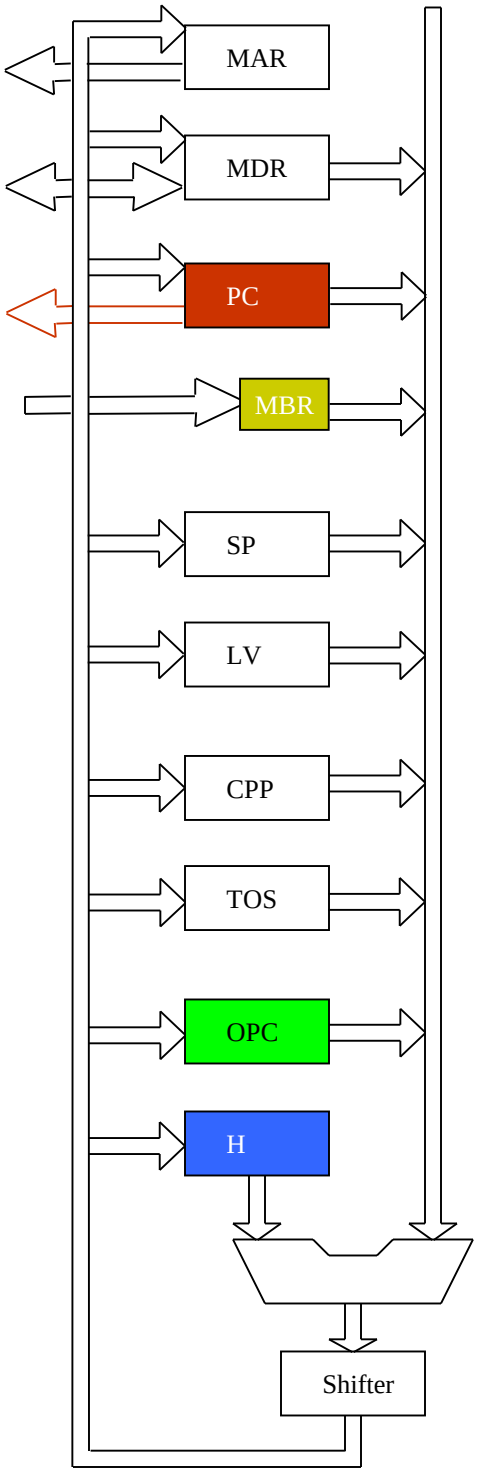


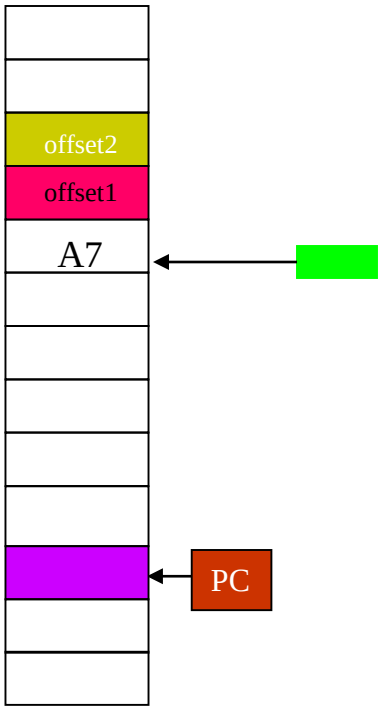


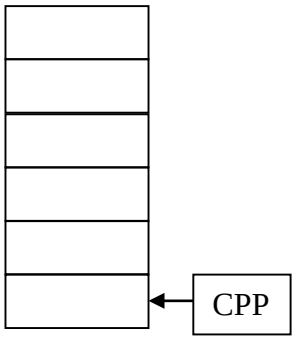


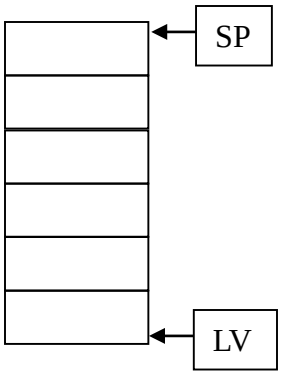
PC = [H] + [OPC]; **fetch**

GOTO



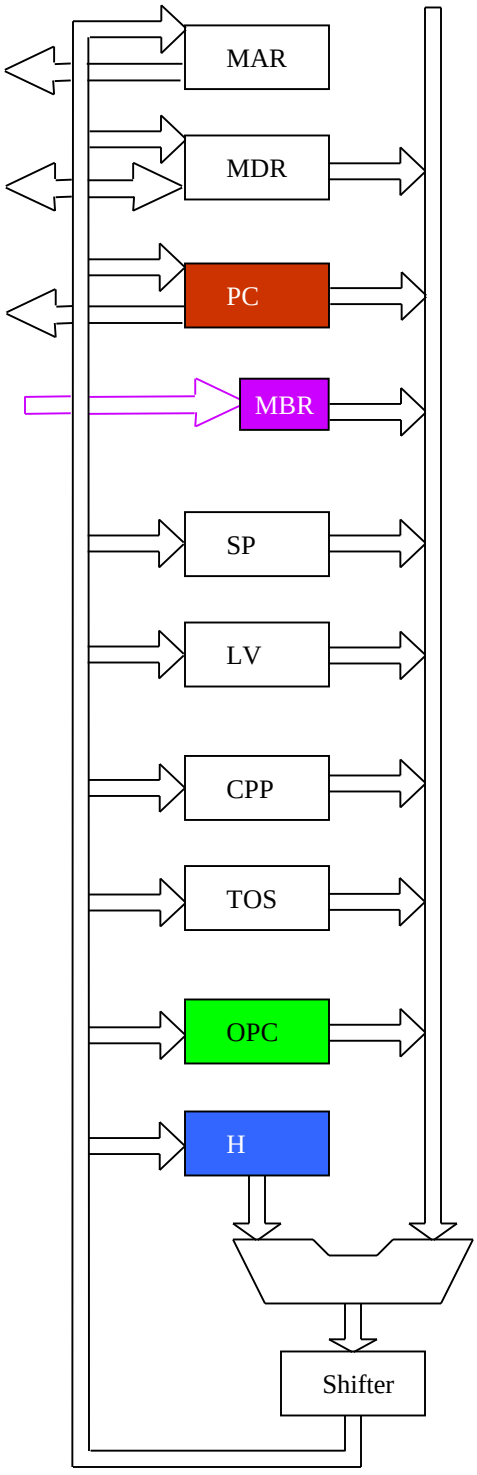


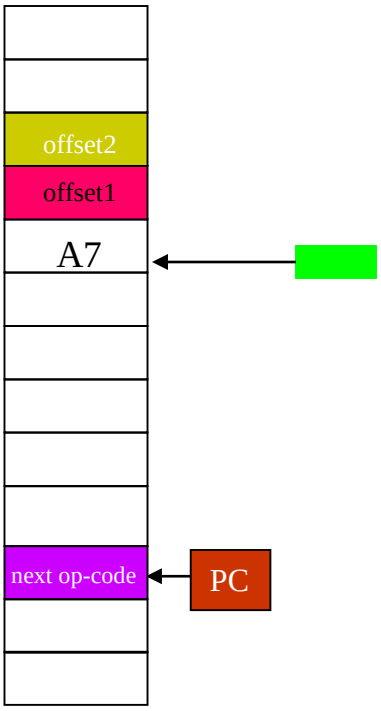


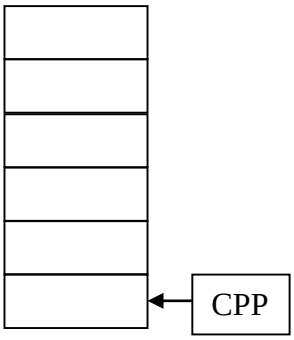


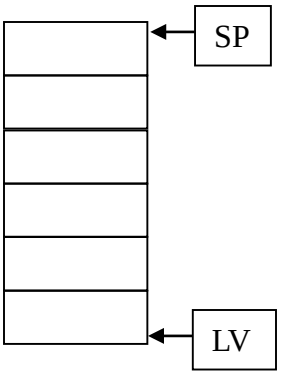
wait until end of
next clock cycle

GOTO









previous fetch completes