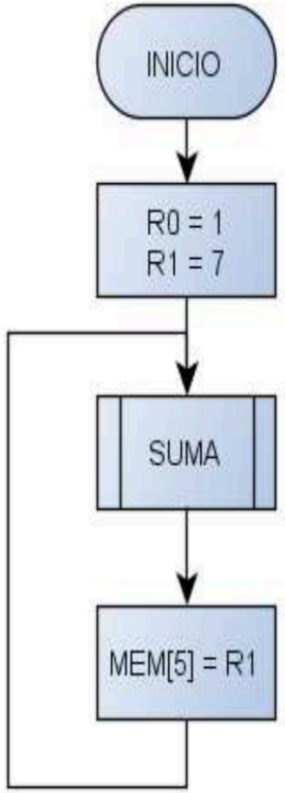
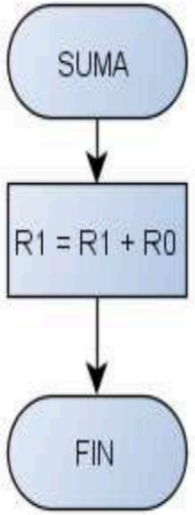
Pila en Hardware

Pila del procesador

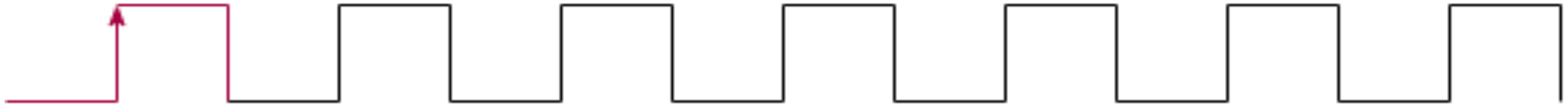
- Guarda las direcciones de retorno en el llamado a subrutinas
- Se compone de un arreglo de contadores y un apuntador para seleccionar uno de ellos






```
        0   Li R0, #1
        1   LI R1, #7
CICLO:  2   CALL SUMA
        3   SWI R1, 0x5
        4   B CICLO

SUMA:   5   ADD R1, R1, R0
        6   RET
```



SP

0

PC(0) = 0
PC(1) = 0
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

PC(SP) = 0

0	**Li R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

LI R0, #1

SP

0

PC(0) = 1
PC(1) = 0
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

PC(SP) = 1

0	**Li R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

LI R1, #7

SP

0

PC(0) = 2
PC(1) = 0
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

PC(SP) = 2

0	**Li R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

CALL SUMA

CALL Lit16

SP ++

PC(SP) = Lit16

SP

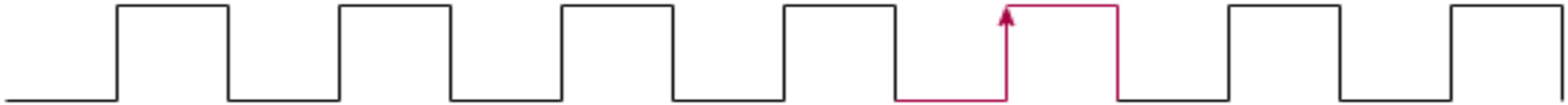
1

PC(0) = 2
PC(1) = 5
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

PC(SP) = 5

0	**Li R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

ADD R1, R1, R0



SP

1

PC(0) = 2
PC(1) = 6
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

PC(SP) = 6

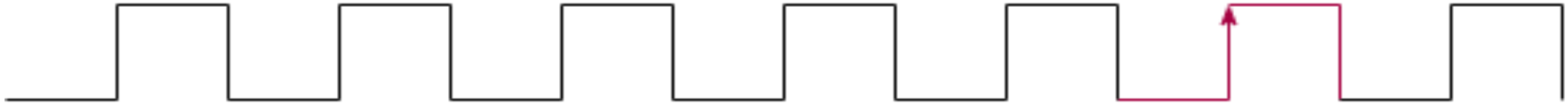
0	**Li R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

RET

RET

SP - -

PC(SP) ++



SP

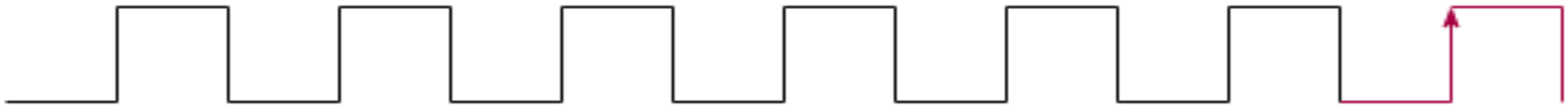
0

PC(0) = 3
PC(1) = 6
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

PC(SP) = 3

0	**Li R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

SWI R1, 0x5



SP

0

PC(0) = 4
PC(1) = 6
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

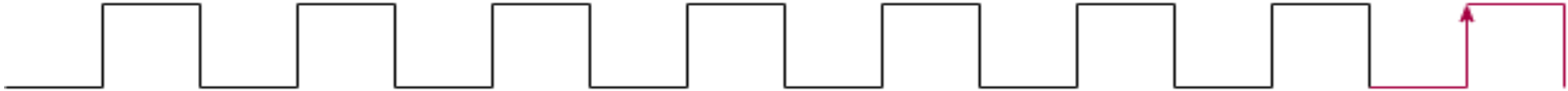
PC(SP) = 4

0	**Li R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

SWI B CICLO

B Lit16

PC(SP)= Lit16



SP

0

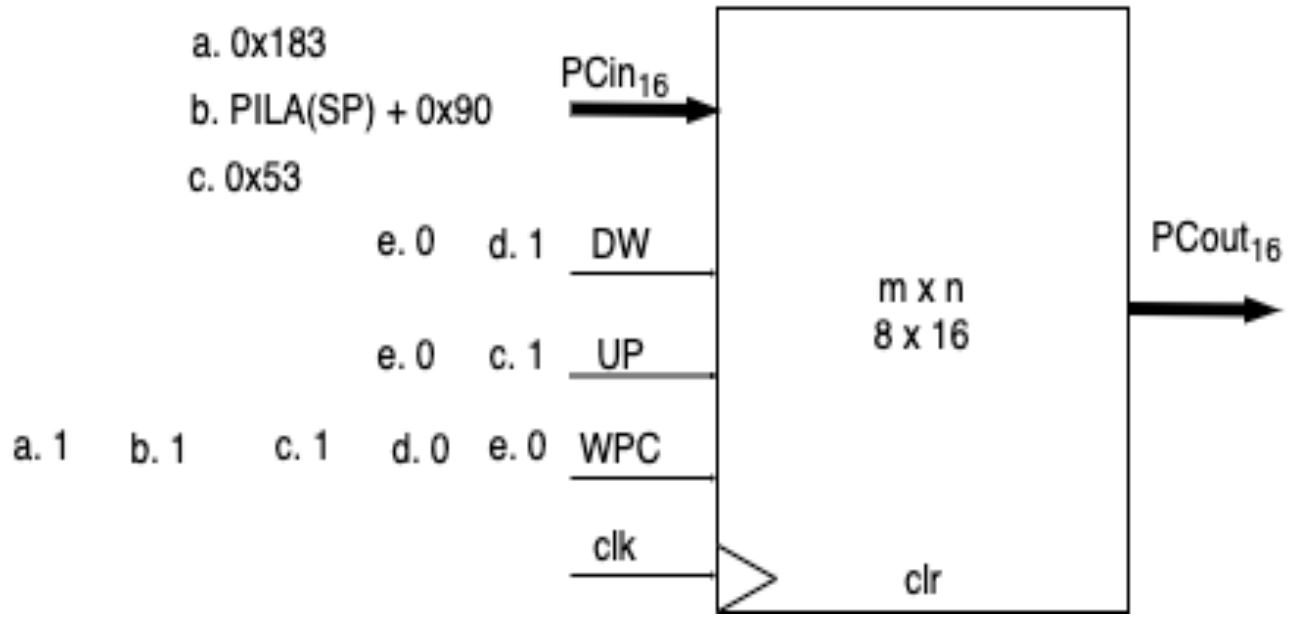
PC(0) = 2
PC(1) = 6
PC(2) = 0
PC(3) = 0
PC(4) = 0
PC(5) = 0
PC(6) = 0
PC(7) = 0

PC(SP) = 2

0	**LI R0, #1**
1	LI R1, #7
2	CALL SUMA
3	SWI R1, 0x5
4	B CICLO
5	ADD R1, R1, R0
6	RET

LI R0, #1

Entidad de la Pila en Hardware



B Lit16
PILA(SP) = Lit16

a. B 0x183
PILA(SP) = 0x183

b. BEQI R4, R7, x90
if(R4 = R7) =>
PILA(SP) = PILA(SP) + 0x90

CALL lit16
SP ++
PILA(SP) = lit16

c. CALL 0x53
SP ++
PILA(SP) = 0x53

RET
SP --
PILA(SP) ++

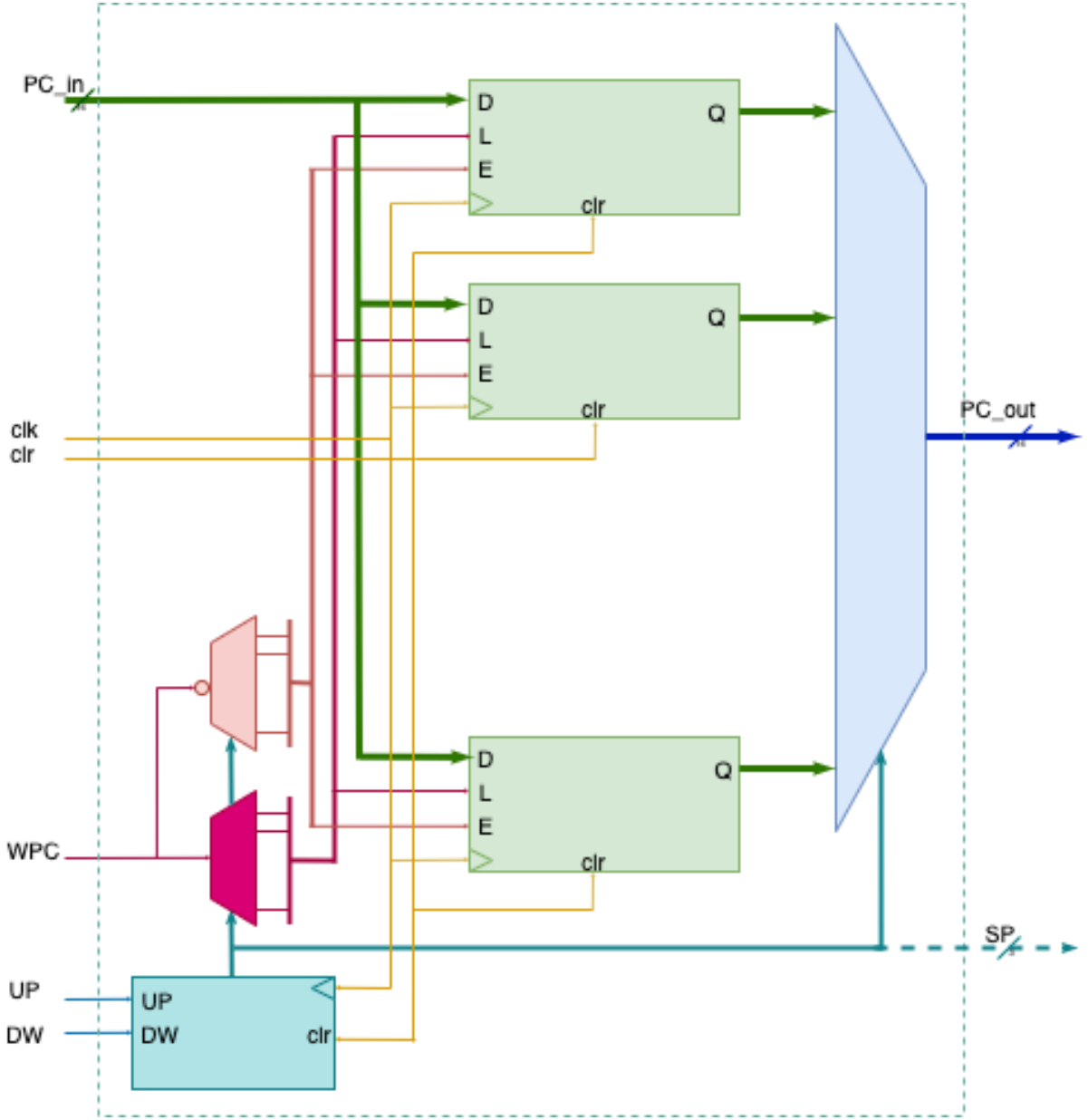
d. RET

e. LI Rd, #Slit16

Tabla de control de la pila

clr	clk	WPC	UP	DW	Operación
1	x	x	x	x	SP = 0 PILA(0, 1, …, 7) = 0
0	↑	0	0	0	SP = SP PILA(SP) ++
0	↑	1	0	0	SP = SP PILA (SP) = PCin
0	↑	1	1	0	SP ++ PILA(SP) = PCin
0	↑	0	0	1	SP – PILA(SP) ++
x	x	x	x	x	Pcout = PILA(SP)

Estructura interna de la Pila en Hardware



clr	clk	L	E	Operación
1	X	X	X	Q = 0
0	↑	0	1	Q = Q + 1
0	↑	1	0	Q = D
0	↑	0	0	Q = Q

clr	clk	UP	DW	Operación
1	X	X	X	SP = 0
0	↑	0	0	SP = SP
0	↑	1	0	SP ++
0	↑	0	1	SP --

SP = 3 && WPC = 1 => “00001000”, “00000000”

SP = 3 && WPC = 0 => “00000000”, “00001000”