

Reporte de practica 10

González Pardo Adrian

Abril 2020

1. Código fuente:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_UNSIGNED.ALL;
4 use IEEE.STD_LOGIC_ARITH.ALL;
5
6 entity stackHardware is
7     Port(PCin:in STD_LOGIC_VECTOR(15 downto 0);
8         DW, UP, WPC, CLK,CLR:in STD_LOGIC;
9         PCout:out STD_LOGIC_VECTOR(15 downto 0);
10        SP_out:out STD_LOGIC_VECTOR(2 downto 0));
11 end stackHardware;
12
13 architecture Behavioral of stackHardware is
14     type banco is array (0 to 7) of std_logic_vector(15 downto 0);
15     signal pila: banco;
16 begin
17     process(CLK,CLR)
18         variable sp: integer range 0 to 7;
19     begin
20         if(CLR = '1') then
21             sp := 0;
22             pila <= (others =>(others => '0'));
23         elsif(CLK'event and clk = '1') then
24             if(WPC = '0' and UP = '0' and DW = '0') then
25                 pila(sp) <= pila(sp)+1;
26             elsif(WPC = '1' AND UP = '0' AND DW = '0') then
27                 pila(sp) <= PCin;
28             elsif(WPC = '1' and UP = '1' and DW = '0')then
29                 sp := sp + 1;
30                 pila(sp) <= PCin;
31             elsif(WPC = '0' and UP = '0' and DW = '1') then
32                 sp:=sp-1;
33                 pila(sp)<=pila(sp)-1;
34             end if;
35         end if;
36         SP_out <= conv_std_logic_vector(sp,3);
37         PCout <= pila(sp);
38     end process;
39 end Behavioral;
```

2. Test-Bench:

```
1 LIBRARY ieee;
2 LIBRARY STD;
3 USE STD.TEXTIO.ALL;
4 USE ieee.std_logic_TEXTIO.ALL;  --PERMITE USAR STD_LOGIC
5 USE ieee.std_logic_1164.ALL;
6 USE ieee.std_logic_UNSIGNED.ALL;
```

```

7  USE ieee.std_logic_ARITH.ALL;
8
9  entity pilaHW_TB is
10 end pilaHW_TB;
11
12 architecture Behavioral of pilaHW_TB is
13     component stackHardware
14         Port(PCin:in STD_LOGIC_VECTOR(15 downto 0);
15             DW, UP, WPC, CLK,CLR:in STD_LOGIC;
16             PCout:out STD_LOGIC_VECTOR(15 downto 0);
17             SP_out:out STD_LOGIC_VECTOR(2 downto 0));
18     end component;
19     --In
20     signal PCin:STD_LOGIC_VECTOR(15 downto 0);
21     signal DW,UP,WPC,CLK,CLR:STD_LOGIC;
22     --Out
23     signal PCout:STD_LOGIC_VECTOR(15 downto 0);
24     signal SP_out:STD_LOGIC_VECTOR(2 downto 0);
25     --CLK
26     constant CLK_period:time:=10ns;
27 begin
28     pilaHWTB:stackHardware port map(
29         PCin=>PCin,
30         DW=>DW,
31         UP=>UP,
32         WPC=>WPC,
33         CLK=>CLK,
34         CLR=>CLR,
35         PCout=>PCout,
36         SP_out=>SP_out
37     );
38     clk_process: process
39     begin
40         CLK<='0';
41         wait for CLK_period/2;
42         CLK<='1';
43         wait for CLK_period/2;
44     end process;
45     tb:process
46         --OUT
47         file ARCH_SAL:TEXT;
48         variable LINEA_SAL:line;
49         variable VAR_PCout:STD_LOGIC_VECTOR(15 downto 0);
50         variable VAR_SP_out:STD_LOGIC_VECTOR(2 downto 0);
51         --IN
52         file ARCH_ENT:TEXT;
53         variable LINEA_ENT:line;
54         variable VAR_PCin:STD_LOGIC_VECTOR (15 downto 0);
55         variable VAR_DW:STD_LOGIC;
56         variable VAR_UP:STD_LOGIC;
57         variable VAR_WPC:STD_LOGIC;
58         variable VAR_CLR:STD_LOGIC;
59         VARIABLE CADENA:STRING(1 TO 6);
60     begin
61         file_open(ARCH_ENT, "/run/media/d3vcr4ck/externData/materias-Sem20_2/
arquitecturaDeComputadoras/arquitecturaDeComputadoras/practicasVivado/pilaHardware_1/in
.txt", READ_MODE);
62         file_open(ARCH_SAL, "/run/media/d3vcr4ck/externData/materias-Sem20_2/
arquitecturaDeComputadoras/arquitecturaDeComputadoras/practicasVivado/pilaHardware_1/
out.txt", WRITE_MODE);
63         CADENA:="SPout.";
64         write(LINEA_SAL, CADENA, right, CADENA'LENGTH+1);
65         CADENA:=" PCout.";
66         write(LINEA_SAL, CADENA, right, CADENA'LENGTH+1);
67         writeline(ARCH_SAL,LINEA_SAL);
68         wait for 100 ns;
69         for I in 0 to 22 loop

```

```

70         readline(ARCH_ENT,LINEA_ENT);
71
72         Hread(LINEA_ENT,VAR_PCin);
73         PCin <= VAR_PCin;
74
75         read(LINEA_ENT,VAR_DW);
76         DW <= VAR_DW;
77
78         read(LINEA_ENT,VAR_UP);
79         UP <= VAR_UP;
80
81         read(LINEA_ENT,VAR_WPC);
82         WPC <= VAR_WPC;
83
84         read(LINEA_ENT,VAR_CLR);
85         CLR <= VAR_CLR;
86
87         --wait until rising_edge(CLK);
88         wait for 10ns;
89         VAR_PCout:=PCout;
90         VAR_SP_out:=SP_out;
91         HWRITE(LINEA_SAL,VAR_SP_out, right, 4);
92         HWRITE(LINEA_SAL,VAR_PCout, right, 9);
93         writeline(ARCH_SAL,LINEA_SAL);
94     end loop;
95     file_close(ARCH_ENT);
96     file_close(ARCH_SAL);
97     wait;
98 end process;
99 end Behavioral;

```

3. Archivos de entrada y salida

3.1. Entrada

```

1 0000 0 0 0 1
2 0000 0 0 0 0
3 A0FF 0 0 0 0
4 0034 0 0 1 0
5 0000 0 0 0 0
6 0000 0 0 0 0
7 0061 0 1 1 0
8 0000 0 0 0 0
9 0000 0 0 0 0
10 0100 0 1 1 0
11 0000 0 0 0 0
12 0000 0 0 0 0
13 0000 0 0 0 0
14 0000 1 0 0 0
15 0000 0 0 0 0
16 0000 0 0 0 0
17 0000 1 0 0 0
18 0300 0 0 1 0
19 0889 0 1 1 0
20 0000 0 0 0 0
21 0000 0 0 0 0
22 0000 0 0 0 0
23 0000 1 0 0 0
24 0000 1 0 0 0

```

3.2. Salida

1	SPout_	PCout
2	0	0000
3	0	0000
4	0	0001
5	0	0002
6	0	0034
7	0	0035
8	1	0000
9	1	0061
10	1	0062
11	2	0000
12	2	0100
13	2	0101
14	2	0102
15	1	0063
16	1	0062
17	1	0063
18	0	0036
19	0	0035
20	1	0064
21	1	0889
22	1	088A
23	1	088B
24	0	0300

4. Diagrama RTL

