Reporte de practica 10

González Pardo Adrian

Abril 2020

1. Código fuente:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;

entity stackHardware is
    Port(PCin:in STD_LOGIC_VECTOR(15 downto 0);
         DW, UP, WPC, CLK,CLR:in STD_LOGIC;
         PCout:out STD_LOGIC_VECTOR(15 downto 0);
         SP_out:out STD_LOGIC_VECTOR(2 downto 0));
end stackHardware;

architecture Behavioral of stackHardware is
    type banco is array (0 to 7) of std_logic_vector(15 downto 0);
    signal pila: banco;
begin
    process(CLK,CLR)
        variable sp: integer range 0 to 7;
    begin
        if(CLR = '1') then
            sp := 0;
            pila <= (others =>(others => '0'));
        elsif(CLK'event and clk = '1') then
            if(WPC = '0' and UP = '0' and DW = '0') then
                pila(sp) <= pila(sp)+1;
            elsif(WPC = '1' AND UP = '0' AND DW = '0') then
                pila(sp) <= PCin;
            elsif(WPC = '1' and UP = '1' and DW = '0')then
                sp := sp + 1;
                pila(sp) <= PCin;
            elsif(WPC = '0' and UP = '0' and DW = '1') then
                sp:=sp-1;
                pila(sp)<=pila(sp)-1;
            end if;
        end if;
        SP_out <= conv_std_logic_vector(sp,3);
        PCout <= pila(sp);
    end process;
end Behavioral;
```

2. Test-Bench:

```
LIBRARY ieee;
LIBRARY STD;
USE STD.TEXTIO.ALL;
USE ieee.std_logic_TEXTIO.ALL;  --PERMITE USAR STD_LOGIC
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_UNSIGNED.ALL;
```

```
USE ieee.std_logic_ARITH.ALL;

entity pilaHW_TB is
end pilaHW_TB;

architecture Behavioral of pilaHW_TB is
    component stackHardware
        Port(PCin:in STD_LOGIC_VECTOR(15 downto 0);
         DW, UP, WPC, CLK,CLR:in STD_LOGIC;
         PCout:out STD_LOGIC_VECTOR(15 downto 0);
         SP_out:out STD_LOGIC_VECTOR(2 downto 0));
    end component;
    --In
    signal PCin:STD_LOGIC_VECTOR(15 downto 0);
    signal DW,UP,WPC,CLK,CLR:STD_LOGIC;
    --Out
    signal PCout:STD_LOGIC_VECTOR(15 downto 0);
    signal SP_out:STD_LOGIC_VECTOR(2 downto 0);
    --CLK
    constant CLK_period:time:=10ns;
begin
    pilaHWTB:stackHardware port map(
        PCin=>PCin,
        DW=>DW,
        UP=>UP,
        WPC=>WPC,
        CLK=>CLK,
        CLR=>CLR,
        PCout=>PCout,
        SP_out=>SP_out
    );
    clk_process: process
    begin
        CLK<='0';
        wait for CLK_period/2;
        CLK<='1';
        wait for CLK_period/2;
    end process;
    tb:process
        --OUT
        file ARCH_SAL:TEXT;
        variable LINEA_SAL:line;
        variable VAR_PCout:STD_LOGIC_VECTOR(15 downto 0);
        variable VAR_SP_out:STD_LOGIC_VECTOR(2 downto 0);
        --IN
        file ARCH_ENT:TEXT;
        variable LINEA_ENT:line;
        variable VAR_PCin:STD_LOGIC_VECTOR (15 downto 0);
        variable VAR_DW:STD_LOGIC;
        variable VAR_UP:STD_LOGIC;
        variable VAR_WPC:STD_LOGIC;
        variable VAR_CLR:STD_LOGIC;
        VARIABLE CADENA:STRING(1 TO 6);
    begin
        file_open(ARCH_ENT, "/run/media/d3vcr4ck/externData/materias-Sem20_2/
    arquitecturaDeComputadoras/arquitecturaDeComputadoras/practicasVivado/pilaHardware_1/in
    .txt", READ_MODE);
        file_open(ARCH_SAL, "/run/media/d3vcr4ck/externData/materias-Sem20_2/
    arquitecturaDeComputadoras/arquitecturaDeComputadoras/practicasVivado/pilaHardware_1/
    out.txt", WRITE_MODE);
        CADENA:="SPout_";
        write(LINEA_SAL, CADENA, right, CADENA'LENGTH+1);
        CADENA:=" PCout";
        write(LINEA_SAL, CADENA, right, CADENA'LENGTH+1);
        writeline(ARCH_SAL,LINEA_SAL);
        wait for 100 ns;
        for I in 0 to 22 loop
```

```
            readline(ARCH_ENT,LINEA_ENT);

            Hread(LINEA_ENT,VAR_PCin);
            PCin <= VAR_PCin;

            read(LINEA_ENT,VAR_DW);
            DW <= VAR_DW;

            read(LINEA_ENT,VAR_UP);
            UP <= VAR_UP;

            read(LINEA_ENT,VAR_WPC);
            WPC <= VAR_WPC;

            read(LINEA_ENT,VAR_CLR);
            CLR <= VAR_CLR;

            --wait until rising_edge(CLK);
            wait for 10ns;
            VAR_PCout:=PCout;
            VAR_SP_out:=SP_out;
            HWRITE(LINEA_SAL,VAR_SP_out, right, 4);
            HWRITE(LINEA_SAL,VAR_PCout, right, 9);
            writeline(ARCH_SAL,LINEA_SAL);
        end loop;
        file_close(ARCH_ENT);
        file_close(ARCH_SAL);
        wait;
    end process;
end Behavioral;
```

3. Archivos de entrada y salida

3.1. Entrada

```
0000 0 0 0 1
0000 0 0 0 0
A0FF 0 0 0 0
0034 0 0 1 0
0000 0 0 0 0
0000 0 0 0 0
0061 0 1 1 0
0000 0 0 0 0
0000 0 0 0 0
0100 0 1 1 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 1 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 1 0 0 0
0300 0 0 1 0
0889 0 1 1 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 1 0 0 0
0000 1 0 0 0
```

3.2. Salida

```
SPout_  PCout
  0     0000
  0     0000
  0     0001
  0     0002
  0     0034
  0     0035
  1     0000
  1     0061
  1     0062
  2     0000
  2     0100
  2     0101
  2     0102
  1     0063
  1     0062
  1     0063
  0     0036
  0     0035
  1     0064
  1     0889
  1     088A
  1     088B
  0     0300
```

4. Diagrama RTL