Instituto Politécnico Nacional

Escuela Superior de Computo

Práctica 9.

Pila Hardware.

Nombre: Flores Castro Luis Antonio.

Arquitectura de Computadoras.

Profesora: Vega García Nayeli.

Código VHDL de Pila Hardware.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;

entity PilaH is
    generic( m : integer := 16;
              n : integer := 3);--3 bits de 0 a 7 binario
    Port ( pcIn : in STD_LOGIC_VECTOR (m-1 downto 0);
               clk, clr, wPC, UP, DW : in STD_LOGIC;
               pcOut : out STD_LOGIC_VECTOR (m-1 downto 0);
               stackP : out STD_LOGIC_VECTOR (n-1 downto 0));
    end PilaH;

    architecture Behavioral of PilaH is
        type contadores is array (0 to (2**n)-1) of STD_LOGIC_VECTOR(m-1 downto 0);
        signal contador : contadores;
    begin
        process(clk, clr, contador)
            variable auxp : integer range 0 to (2**n)-1;
        begin
            if (clr = '1') then
                auxp := 0;
                contador <= (others => (others => '0'));
            elsif (rising_edge(clk)) then
                if (wPC = '0' and UP = '0' and DW = '0') then
                    contador(auxp) <= contador(auxp) + 1;
                elsif (wPC = '1' and UP = '0' and DW = '0') then
                    contador(auxp) <= pcIn;
                elsif (wPC = '1' and UP = '1' and DW = '0') then
                    auxp := auxp + 1;
                    if(auxp = 2**n) then
                        auxp := 0;
                    end if;
                    contador(auxp) <= pcIn;
                elsif (wPC = '0' and UP = '0' and DW = '1') then
                    auxp := auxp - 1;
                    if(auxp = -1) then
                        auxp := (2**n)-1;
                    end if;
                    contador(auxp) <= contador(auxp) + 1;
                end if;
            end if;

        pcOut <= contador(auxp);
        stackP <= conv_std_logic_vector(auxp, n);

    end process;
end Behavioral;
```

Código VHDL Test-Bench de Pila Hardware.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_arith.all;
use IEEE.STD_LOGIC_unsigned.ALL;
use IEEE.STD_LOGIC_TEXTIO.ALL;
use STD.TEXTIO.ALL;

entity Tbpila is
end Tbpila;

    architecture Behavioral of Tbpila is
        component PilaH is
            Port ( pcIn : in STD_LOGIC_VECTOR (15 downto 0);
                   clk, clr, wPC, UP, DW : in STD_LOGIC;
                   pcOut : out STD_LOGIC_VECTOR (15 downto 0);
                   stackP : out STD_LOGIC_VECTOR (2 downto 0));
        end component;

        --signals
        signal pcIn : STD_LOGIC_VECTOR (15 downto 0)  := (others => '0');
        signal clk, clr, wPC, UP, DW : STD_LOGIC;
        signal pcOut : STD_LOGIC_VECTOR (15 downto 0);
        signal stackP: STD_LOGIC_VECTOR (2 downto 0);
    begin
        stack: PilaH
        Port map (
            pcIn => pcIn,
            clk => clk,
            clr => clr,
            wPC => wPC,
            UP => UP,
            DW => DW,
            pcOut => pcOut,
            stackP => stackP
        );

        reloj : process
        begin
            clk <= '0';
            wait for 5 ns;
            clk <= '1';
            wait for 5 ns;
        end process;

        process
```

```vhdl
            file RES : TEXT;
            variable L_RE : line;
            variable vpcOut : STD_LOGIC_VECTOR (15 downto 0);

            file STIMU : TEXT;
            variable L_E: line;
            variable vpcIn : STD_LOGIC_VECTOR (15 downto 0);
            variable vclr : STD_LOGIC;
            variable vwPC : STD_LOGIC;
            variable vUP : STD_LOGIC;
            variable vDW : STD_LOGIC;
            variable CADENA : string (1 to 2);
        begin

            file_open(STIMU, "C:\Users\Luis FC\Documents\Semestre 21-2\Arquitectura de Computadoras\Pila Hardware\ESTIMULOS.TXT", READ_MODE);
            file_open(RES, "C:\Users\Luis FC\Documents\Semestre 21-2\Arquitectura de Computadoras\Pila Hardware\RESULTADO.TXT", WRITE_MODE);

            CADENA := "SP";
            write(L_RE, CADENA, LEFT,3);

            CADENA := "PC";
            write(L_RE, CADENA, LEFT,5);

            writeline(RES, L_RE);

            for i in 0 to 25 loop--26 Estimulos con reset
                readline(STIMU, L_E);

                Hread(L_E, vpcIn);
                pcIn <= vpcIn;

                read(L_E, vclr);
                clr <= vclr;

                read(L_E, vwPC);
                wPC <= vwPC;

                read(L_E, vUP);
                UP <= vUP;

                read(L_E, vDW);
                DW <= vDW;

                wait until rising_edge(clk);
                wait for 0.1 ns;
                vpcOut := pcOut;

                Hwrite(L_RE, stackP, left, 3);
                Hwrite(L_RE, vpcOut, left, 5);

                writeline(RES, L_RE);
            end loop;

            file_close(STIMU);
            file_close(RES);
            wait;
        end process;
    end Behavioral;
```

Diagrama lógico de la Pila Hardware.

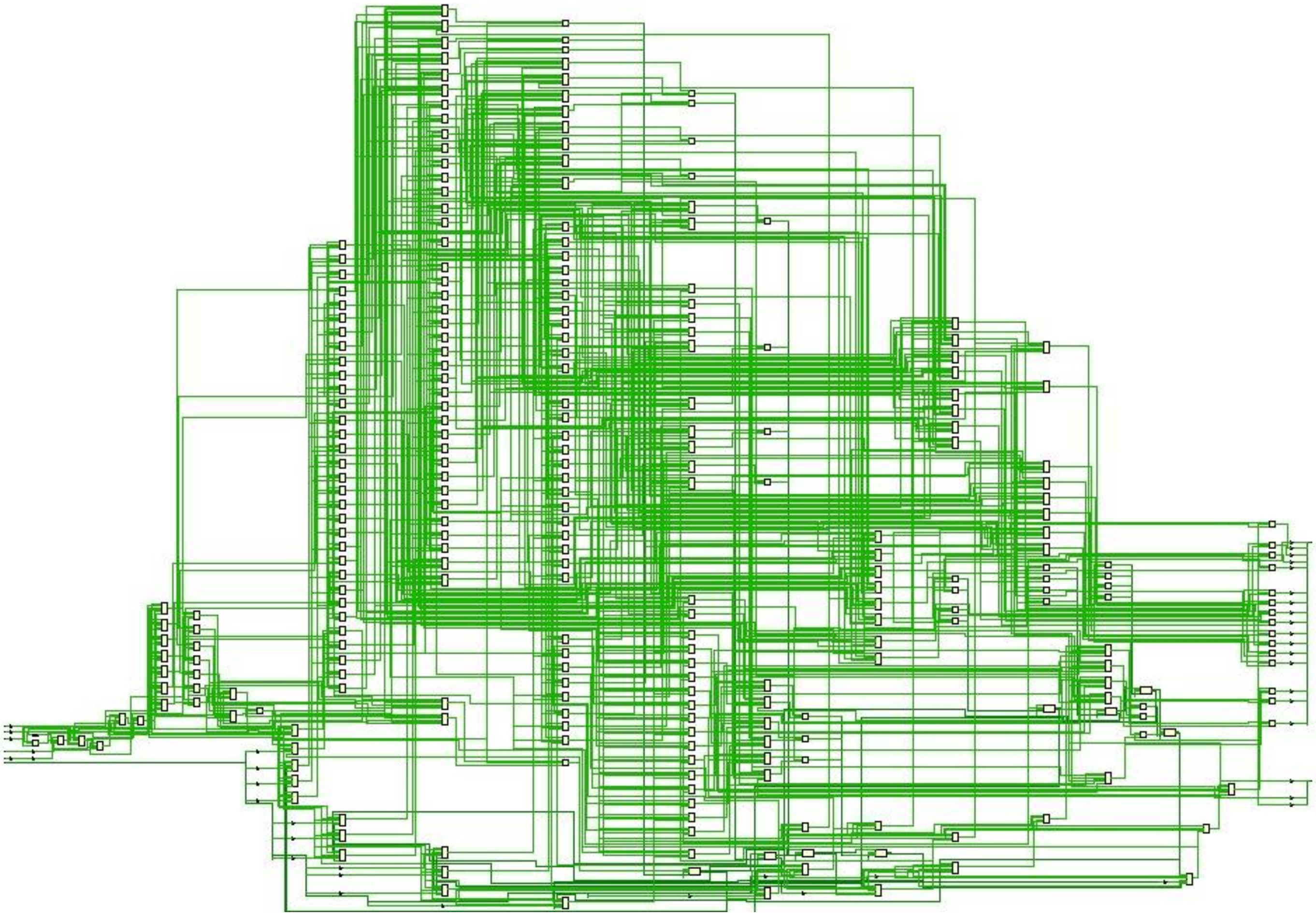


Figura 1. Diagrama lógico de la Pila Hardware.

Diagrama RTL de la Pila Hardware.

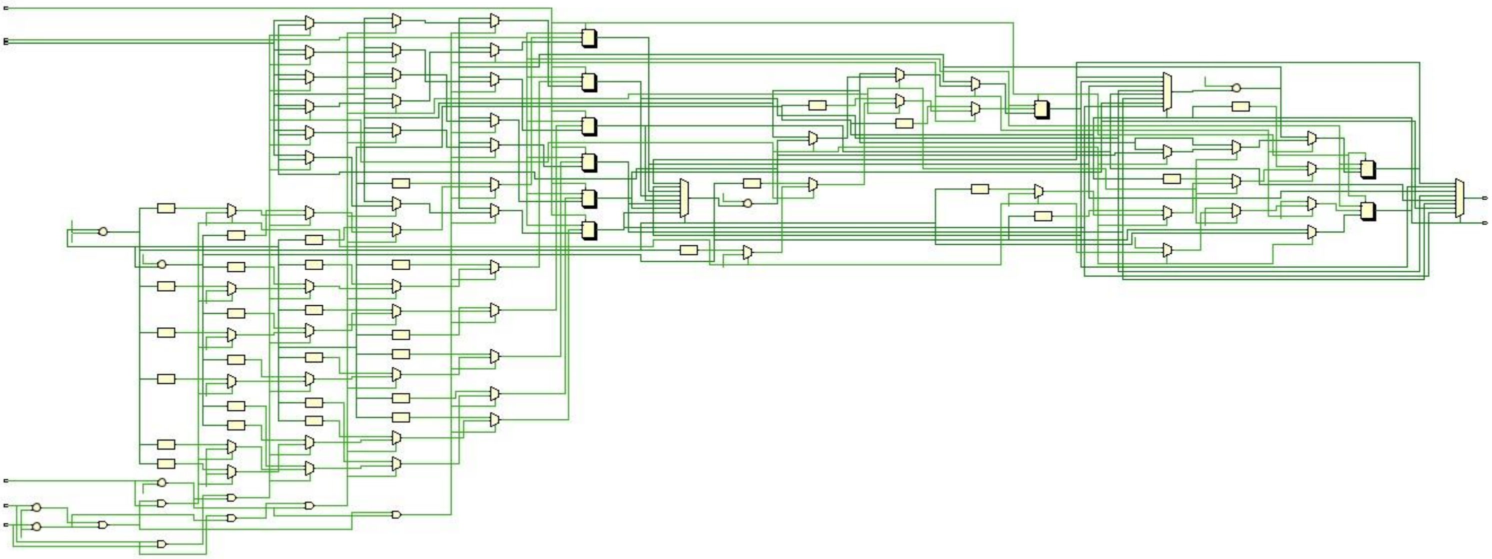


Figura 1.1 Diagrama RTL de la Pila Hardware.

Simulación de Onda de la Pila Hardware.

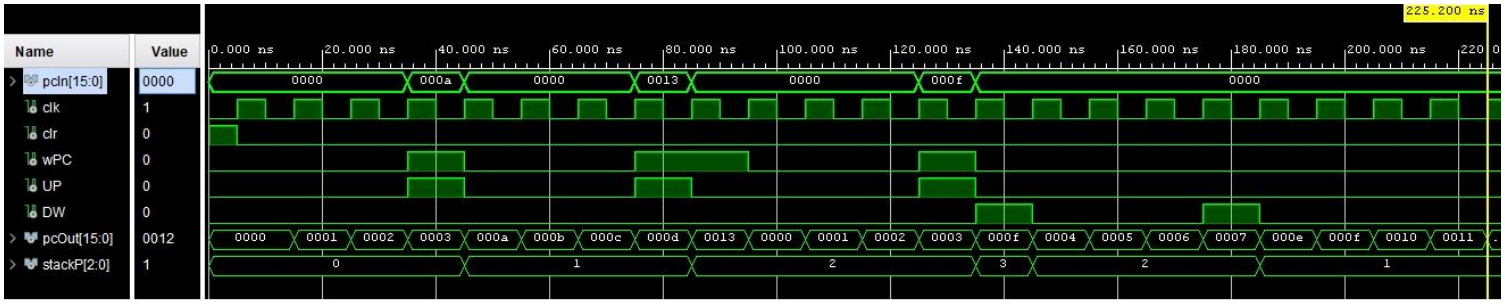
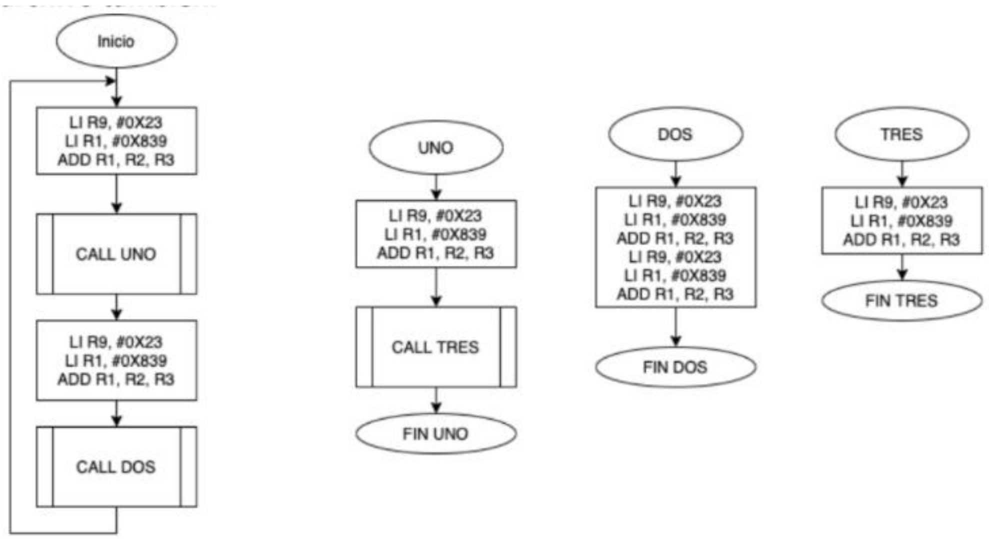


Figura 1.2 Diagrama de Onda de la Pila Hardware.

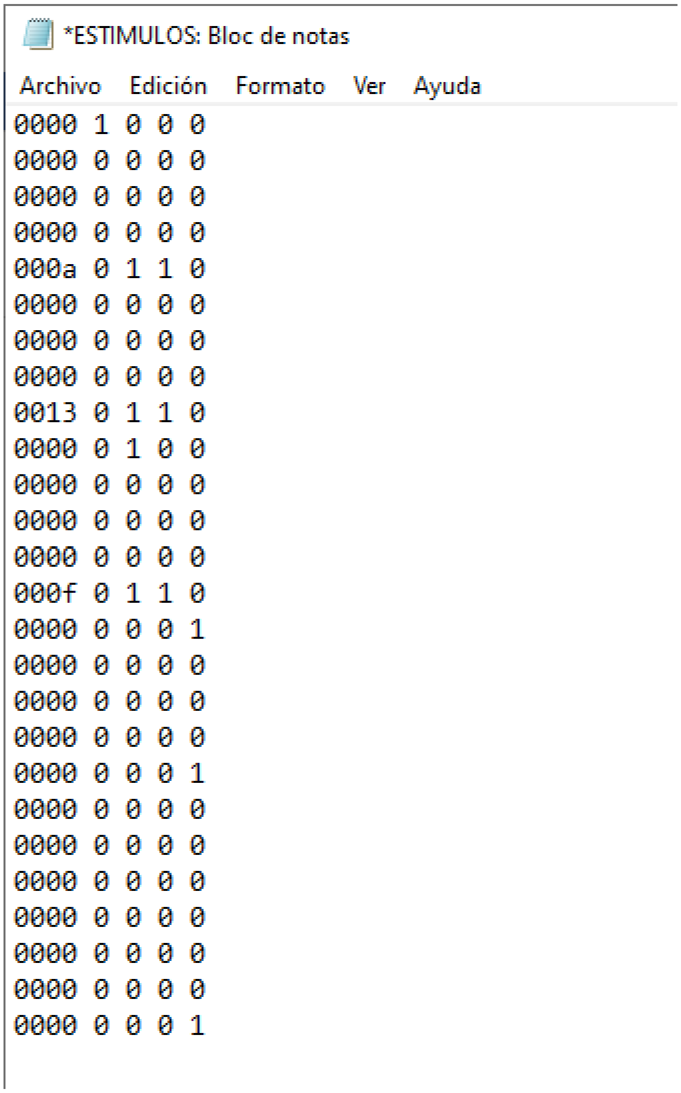
Programa.



Memoria de Programa de los Programas.

```
0. Reset
1. LI R9, #0x23
2. LI R1, #0x839
3. ADD R1, R2, R3
4. CALL UNO
5. LI R9, #0x23
6. LI R1, #0x839
7. ADD R1, R2, R3
8. CALL DOS
9. B inicio lit=0
10.     LI R9, #0x23
11.     LI R1, #0x839
12.     ADD R1, R2, R3
13.     CALL TRES
14.     RET
15.     LI R9, #0x23
16.     LI R1, #0x839
17.     ADD R1, R2, R3
18.     RET
19.     LI R9, #0x23
20.     LI R1, #0x839
21.     ADD R1, R2, R3
22.     LI R9, #0x23
23.     LI R1, #0x839
24.     ADD R1, R2, R3
25.     RET
```

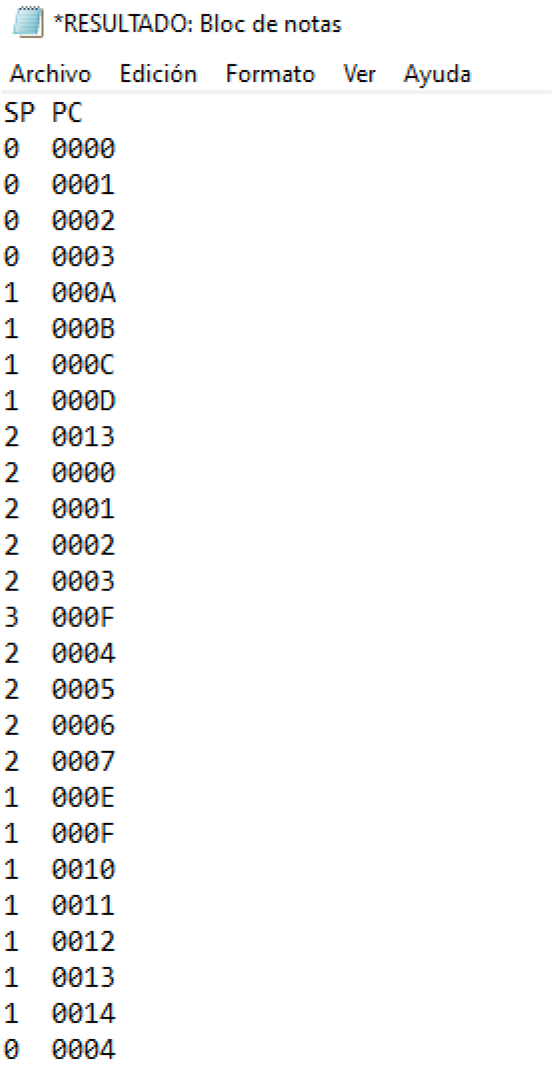
Archivo de estímulos de entrada de la Pila Hardware.



```
0000 1 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
000a 0 1 1 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0013 0 1 1 0
0000 0 1 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
000f 0 1 1 0
0000 0 0 0 1
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 1
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 0
0000 0 0 0 1
```

Figura 1.3 Archivo de Texto de estímulos de entrada de la Pila Hardware.

Archivo de salida de la Pila Hardware.



```
SP  PC
0   0000
0   0001
0   0002
0   0003
1   000A
1   000B
1   000C
1   000D
2   0013
2   0000
2   0001
2   0002
2   0003
3   000F
2   0004
2   0005
2   0006
2   0007
1   000E
1   000F
1   0010
1   0011
1   0012
1   0013
1   0014
0   0004
```

Figura 1.4 Archivo de resultado de la Pila Hardware generado al realizar el Test Bench.