

INSTITUTO POLITECNICO NACIONAL

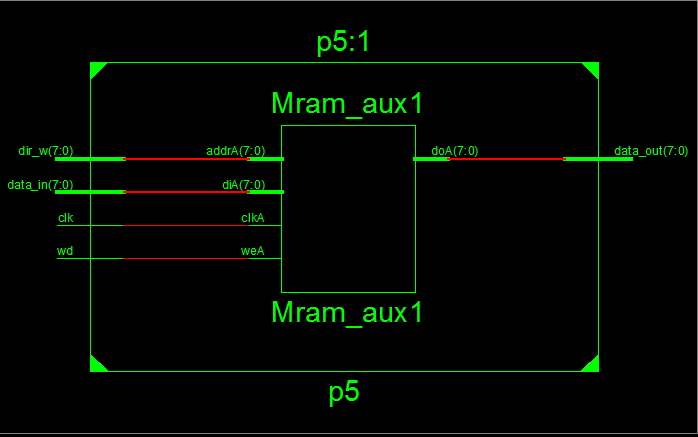
ESCUELA SUPERIOR DE CÓMPUTO

ARQUITECURA DE COMPUTADORAS

PRACTICA 5

MEMORIA DE DATOS

CRUZ MONDRAGON DIEGO



**Código Practica 5**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_arith.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

entity p5 is

generic (

bus\_dir : integer := 8;

bus\_datos : integer:= 8

);

Port ( --dir\_r : in STD\_LOGIC\_VECTOR (bus\_dir-1 downto 0);

dir\_w : in STD\_LOGIC\_VECTOR (bus\_dir-1 downto 0);

data\_in : in STD\_LOGIC\_VECTOR (bus\_datos-1 downto 0);

data\_out : out STD\_LOGIC\_VECTOR (bus\_datos-1 downto 0);

clk, wd : in STD\_LOGIC);

end p5;

architecture Behavioral of p5 is

type ejemplo is array (0 to (2\*\*bus\_dir)-1) of std\_logic\_vector(bus\_Datos-1 downto 0);

signal aux : ejemplo := (others=>(others=> '0'));

begin

process(clk)

begin

if (rising\_edge(clk)) then

if (WD = '1') then --escritura

aux(conv\_integer(dir\_w)) <= data\_in;

end if;

else

end if;

end process;

data\_out <= aux(conv\_integer(dir\_w));

end Behavioral;

Código Simulación

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

use std.textio.all;

use ieee.Numeric\_std.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_textio.all;

ENTITY t\_p5 IS

END t\_p5;

ARCHITECTURE behavior OF t\_p5 IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT p5

PORT(

dir\_w : IN std\_logic\_vector(7 downto 0);

data\_in : IN std\_logic\_vector(7 downto 0);

data\_out : OUT std\_logic\_vector(7 downto 0);

clk : IN std\_logic;

wd : IN std\_logic

);

END COMPONENT;

--Inputs

signal dir\_w : std\_logic\_vector(7 downto 0) := (others => '0');

signal data\_in : std\_logic\_vector(7 downto 0) := (others => '0');

signal clk : std\_logic := '0';

signal wd : std\_logic := '0';

--Outputs

signal data\_out : std\_logic\_vector(7 downto 0);

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: p5 PORT MAP (

dir\_w => dir\_w,

data\_in => data\_in,

data\_out => data\_out,

clk => clk,

wd => wd

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

-- Stimulus process

stim\_proc: process

--Entradas

FILE archivoEntrada : TEXT;

VARIABLE lineaEntrada : LINE;

VARIABLE var\_wd : STD\_LOGIC;

VARIABLE var\_Din, var\_Dir: STD\_LOGIC\_VECTOR ( 7 DOWNTO 0 );

--Salidas

FILE archivoSalida : TEXT;

VARIABLE lineaOut : LINE;

VARIABLE var\_Dout: STD\_LOGIC\_VECTOR ( 7 DOWNTO 0 );

VARIABLE cadena : STRING ( 1 TO 4 );

VARIABLE cadena1 : STRING ( 1 TO 5 );

begin

FILE\_OPEN (archivoEntrada, "in.txt", READ\_MODE);

FILE\_OPEN (archivoSalida, "out.txt", WRITE\_MODE);

cadena := " DIN";

WRITE ( lineaOut, cadena, RIGHT, cadena'LENGTH+1 );

cadena := " DIR";

WRITE ( lineaOut, cadena, RIGHT, cadena'LENGTH+1 );

cadena := " WD";

WRITE ( lineaOut, cadena, RIGHT, cadena'LENGTH+1 );

cadena := "DOUT";

WRITE ( lineaOut, cadena, RIGHT, cadena'LENGTH+1 );

WRITELINE (archivoSalida, lineaOut);

WAIT FOR 10 NS;

--Se lee hasta encontrar el fin del archivo

WHILE NOT ENDFILE (archivoEntrada) LOOP

READLINE (archivoEntrada,lineaEntrada);

HREAD(lineaEntrada, var\_Din);

data\_in <= var\_Din;

HREAD(lineaEntrada, var\_Dir);

dir\_w <= var\_Dir;

READ(lineaEntrada, var\_wd);

wd <= var\_wd;

WAIT UNTIL RISING\_EDGE ( clk );

var\_Dout := data\_out;

--Se escriben entradas

HWRITE(lineaOut, var\_Dir, RIGHT, 5 );

HWRITE(lineaOut, var\_Din, RIGHT, 5 );

WRITE(lineaOut, var\_wd, RIGHT, 5 );

--Se escriben salidas

HWRITE(lineaOut, var\_Dout, RIGHT, 5 );

WRITELINE(archivoSalida, lineaOut);

END LOOP;

--Se cierran archivos de texto

FILE\_CLOSE(archivoEntrada);

FILE\_CLOSE(archivoSalida);

WAIT;

end process;

END;

Imágenes simulaciones

