*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Arquitectura de Computadoras*

***Practica 7: Memoria de Datos***

***Nombre:*** *Sampayo Hernández Mauro*

***Grupo:*** *3CV8*

***Profesor:*** *Nayeli Vega García*

***Fecha de entrega:*** *9 de abril del 2020*

**Código de implementación:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity MemoriaDatos is

generic (d : integer := 16);

Port (dataIn: in STD\_LOGIC\_VECTOR (d-1 downto 0);

add : in STD\_LOGIC\_VECTOR (d-1 downto 0);

clk, WD : in STD\_LOGIC;

dataOut : out STD\_LOGIC\_VECTOR (d-1 downto 0));

end MemoriaDatos;

architecture Behavioral of MemoriaDatos is

type banco is array (0 to (2\*\*d)-1) of std\_logic\_vector(d-1 downto 0);

signal memData : banco;--memoria del banco de memoria

begin

process(clk)

begin

if (clk' event and clk = '1') then

if (WD = '1') then

memData(conv\_integer(add)) <= dataIn;

end if;

end if;

end process;

dataOut <= memData(conv\_integer(add));

end Behavioral;

**Código de simulación:**

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 MemoriaDatos\_tb is

end MemoriaDatos\_tb;

architecture Behavioral of MemoriaDatos\_tb is

component MemoriaDatos is

Port (dataIn: in STD\_LOGIC\_VECTOR (15 downto 0);

add : in STD\_LOGIC\_VECTOR (15 downto 0);

clk, WD : in STD\_LOGIC;

dataOut : out STD\_LOGIC\_VECTOR (15 downto 0));

end component;

--Inputs

signal dataIn : std\_logic\_vector(15 downto 0) := (others => '0');

signal add : std\_logic\_vector(15 downto 0) := (others => '0');

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

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

--Outputs

signal dataOut : std\_logic\_vector(15 downto 0);

-- Clock period definitions

constant clk\_period : time := 10 ns;

begin

uut: MemoriaDatos Port Map(

dataIn => dataIn,

add => add,

clk => clk,

wd => wd,

dataOut => dataOut);

-- 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

file ARCH\_RES : TEXT;--Archivo de resultados

variable LINEA\_RES : line;--Linea de resultado

file ARCH\_VEC : TEXT;--Archivo de vectores

variable LINEA\_VEC : line;--Linea de vectores

--Variables

variable V\_DATAIN, V\_ADD, V\_DATAOUT: STD\_LOGIC\_VECTOR(15 DOWNTO 0);

variable V\_WD: STD\_LOGIC;

--Cadena

variable CADENA : STRING(1 TO 4);

begin

file\_open(ARCH\_VEC, "VECTORES.txt", READ\_MODE);

file\_open(ARCH\_RES, "RESULTADO.txt", WRITE\_MODE);

--Impresión de Encabezado

CADENA := " ADD";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH+1); --"ADD"

CADENA := " DIN";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH+1); --"DATAIN"

CADENA := " WD";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH+1); --"WRITE DATA"

CADENA := "DOUT";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH+1); --"DATAOUT"

writeline(ARCH\_RES,LINEA\_RES);--Escribe la linea en el archivo

--Impresión de Resultados

wait for 100 ns;

for i in 0 to 11 loop

--lectura de VECTORES.TXT--

readline(ARCH\_VEC,LINEA\_VEC);--Lee una linea completa

Hread(LINEA\_VEC, V\_ADD);

ADD <= V\_ADD;

Hread(LINEA\_VEC, V\_DATAIN);

DATAIN <= V\_DATAIN;

read(LINEA\_VEC, V\_WD);

WD <= V\_WD;

wait until RISING\_EDGE(CLK);--Espera al Flanco de Subida

V\_DATAOUT := DATAOUT;--Asignación de salidas

--Escritura de Resultados

Hwrite(LINEA\_RES, V\_ADD, right, 5);

Hwrite(LINEA\_RES, V\_DATAIN, right, 5);

write(LINEA\_RES, V\_WD, right, 5);

Hwrite(LINEA\_RES, V\_DATAOUT, right, 5);

writeline(ARCH\_RES,LINEA\_RES);--Escribe la linea en el archivo

end loop;

file\_close(ARCH\_VEC);--Cierra el archivo

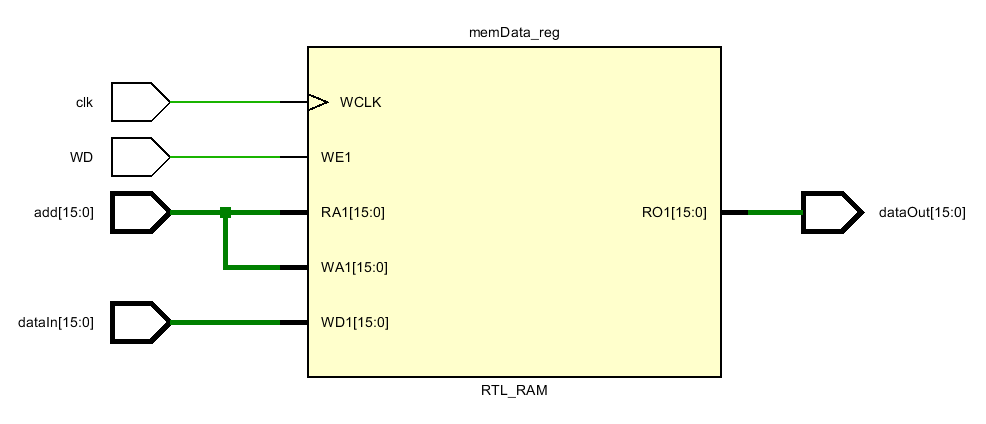
file\_close(ARCH\_RES);--Cierra el archivo

wait;

end process;--Stimulus process

end Behavioral;

**Diagrama RTL:**



**Simulación:**

