Módulo 4: Memoria

Lady Johanna Trejos Hernández

Andrés Felipe Betancurt Rivera

Docente Ramiro Andrés Barrios Valencia

Universidad Tecnológica de Pereira

Facultad de Ingenierías FEECC

Asignatura Laboratorio de Electrónica Digital

27 de abril de 2017

La tarjeta de desarrollo Nexys 2 cuenta con dos tipos de memoria RAM: la interna y la externa.

La memoria externa es propia de la tarjeta de desarrollo, es una RAM de 128Mbit organizados como 8 Mbytes x 16 bits.

La memoria interna es proporcionada por la FPGA Xilinx Spartan 3E, sobre la cual está basada la Nexys 2. Proporciona almacenamiento de datos en forma de bloques de doble puerto de 18 Kbit. Puede ser de dos clases:

* **Por bloques:** Utilizando los bloques de memoria RAM proporcionados por la FPGA.
* **Distribuida:** Configurar los *CLB (Configurable Logic Blocks)* para que funcionen como memoria. Funciona de manera asíncrona y su desventaja es que se tendrían menos CLB para la lógica que se esté desarrollando.

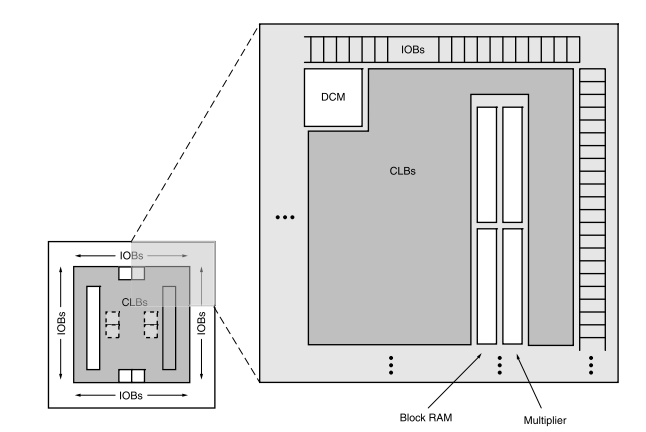


Ilustración 1. Arquitectura de la FPGA Spartan 3E

Para este módulo se ha escogido la memoria externa, ya que ofrece mayor capacidad de almacenamiento (8 MB x 16 bits).

El código VHDL es el siguiente:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Memoria is

port( clk, lectura\_escritura, habilitador: in STD\_LOGIC;

direccion: in STD\_LOGIC\_VECTOR(3 downto 0);

dato\_entrada: in STD\_LOGIC\_VECTOR(2 downto 0);

dato\_salida: out STD\_LOGIC\_VECTOR(2 downto 0));

end Memoria;

architecture Behavioral of Memoria is

constant bits\_direccion : integer := 4;

constant bits\_dato : integer := 3;

type Block\_ram is array(2\*\*bits\_direccion-1 downto 0) of STD\_LOGIC\_VECTOR(bits\_dato-1 downto 0);

signal RAM: Block\_ram;

begin

process(clk)

begin

if clk'event and clk = '1' then

if habilitador = '1' then

if lectura\_escritura = '1' then

RAM(conv\_integer(direccion)) <= dato\_entrada;

else

dato\_salida <= RAM(conv\_integer(direccion));

end if;

end if;

end if;

end process;

end Behavioral;

En el código se ha definido un nuevo tipo de señal que es un símil a una matriz de 8x8 para este módulo.

Las constantes bitsdir y bitsdata son para definir, o cambiar, los bits de la dirección y los bits del dato respectivamente.

Para la simulación se presenta el siguiente Test Bench:

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY memoria\_tb IS

END memoria\_tb;

ARCHITECTURE behavior OF memoria\_tb IS

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

COMPONENT Memoria

PORT(

clk : IN std\_logic;

lectura\_escritura : IN std\_logic;

habilitador : IN std\_logic;

direccion : IN std\_logic\_vector(3 downto 0);

dato\_entrada : IN std\_logic\_vector(2 downto 0);

dato\_salida : OUT std\_logic\_vector(2 downto 0)

);

END COMPONENT;

--Inputs

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

signal lectura\_escritura : std\_logic := '0';

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

signal direccion : std\_logic\_vector(3 downto 0) := (others => '0');

signal dato\_entrada : std\_logic\_vector(2 downto 0) := (others => '0');

--Outputs

signal dato\_salida : std\_logic\_vector(2 downto 0);

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: Memoria PORT MAP (

clk => clk,

lectura\_escritura => lectura\_escritura,

habilitador => habilitador,

direccion => direccion,

dato\_entrada => dato\_entrada,

dato\_salida => dato\_salida

);

-- 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

begin

habilitador <= '1';

lectura\_escritura <= '1'; -- Escritura

wait for 10 ns;

direccion <= "0000"; -- Guardando dato

dato\_entrada <= "101";

wait for 30 ns;

direccion <= "0100"; -- Guardando dato

dato\_entrada <= "001";

wait for 30 ns;

lectura\_escritura <= '0'; -- Lectura

direccion <= "0000"; -- Leyendo dato

wait for 30 ns;

direccion <= "0100"; -- Leyendo dato

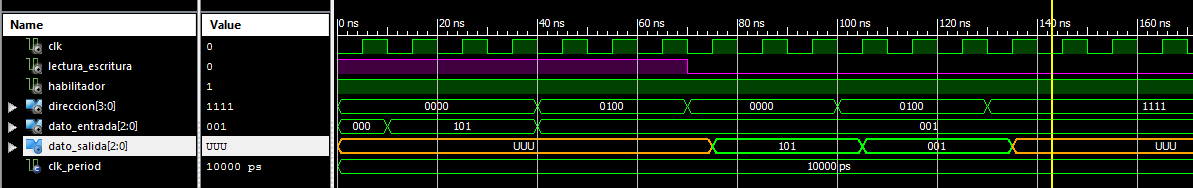
wait for 30 ns;

direccion <= "1111"; -- Leyendo de una posición sin asignación

wait;

end process;

END;

**IMÁGEN DE LA SIMULACIÓN:**

**WEBGRAFÍA**

*Digilent Nexys2 Board Reference Manual*. (2008). *Department of Software Technology*. Recuperado el 19 de abril de 2017, de <http://www.st.ewi.tudelft.nl/~gemund/Courses/In4073/Resources/nexys2_reference_manual.pdf>

*Spartan-3E FPGA Family Data Sheet*. (2013). *Xilinx*. Recuperado el 19 de abril de 2017, de <https://www.xilinx.com/support/documentation/data_sheets/ds312.pdf>