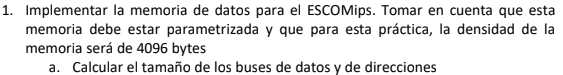
# Práctica 7

# Memoria de datos

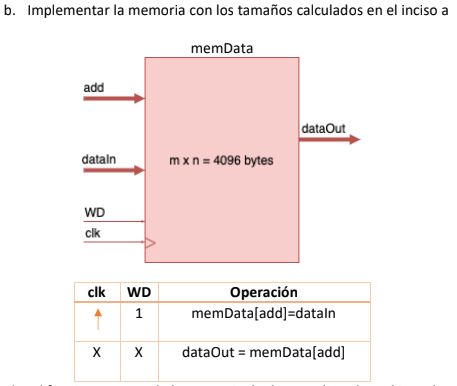


Dado que se trata de una densidad de 4096 bytes tenemos que la densidad convertida a bits es de **.**

Ahora que conocemos la densidad en bits, y dado que también conocemos la longitud de la literal de la instrucción, la cual es de 16 bits, lo siguiente será dividir la densidad entre la longitud de palabra.

Así tenemos que ; por lo cual ahora conocemos el valor tanto de ***m*** como de ***n****,* siendo ***m = 16*** y ***n = 2048.***

Cabe aclarar que, dado que se poseerá un número de palabras igual a 2048, se requiere de un bus de direcciones de tamaño **.**



1. **Código de implementación**



1. **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 tb\_Arreglo is

end tb\_Arreglo;

architecture tb of tb\_Arreglo is

component Arreglo

    port (addrs   : in std\_logic\_vector (10 downto 0);

          dataIn  : in std\_logic\_vector (15 downto 0);

          clk     : in std\_logic;

          wd      : in std\_logic;

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

end component;

signal addrs   : std\_logic\_vector (10 downto 0);

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

signal clk     : std\_logic;

signal wd      : std\_logic;

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

constant CLK\_period : time := 10 ns;

begin

dut : Arreglo

port map (addrs   => addrs,

          dataIn  => dataIn,

          clk     => clk,

          wd      => wd,

          dataOut => dataOut);

CLK\_process :process

begin

    CLK <= '0';

    wait for CLK\_period/2;

    CLK <= '1';

wait for CLK\_period/2;

end process;

stimuli : process

file ARCH\_RES : TEXT;

variable LINEA\_RES : LINE;

VARIABLE VAR\_DATAOUT : STD\_LOGIC\_VECTOR(15 DOWNTO 0);

file ARCH\_VEC : TEXT;

variable LINEA\_VEC : line;

VARIABLE VAR\_RDATAIN :  STD\_LOGIC\_VECTOR(15 DOWNTO 0);

VARIABLE VAR\_RADDRS : STD\_LOGIC\_VECTOR(10 DOWNTO 0);

VARIABLE VAR\_RWD : STD\_LOGIC;

VARIABLE CADENA : STRING(1 TO 4);

VARIABLE CADENA\_I : STRING(1 TO 6);

VARIABLE CADENA\_X : STRING(1 TO 5);

VARIABLE CADENA\_W : STRING(1 TO 7);

begin

  file\_open(ARCH\_RES, "D:\ESCOM\ARQUITECTURA\MemoriaDatos\MemoriDatos.srcs\sim\_1\new\resultado.txt", WRITE\_MODE);

file\_open(ARCH\_VEC, "D:\ESCOM\ARQUITECTURA\MemoriaDatos\MemoriaDatos.srcs\sim\_1\new\entradas.txt", READ\_MODE);

    CADENA\_X := " ADD ";

    write(LINEA\_RES, CADENA\_X, right, CADENA\_X'LENGTH+1);

    CADENA := " WD ";

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

    CADENA\_W := "dataIN ";

    write(LINEA\_RES, CADENA\_W, right, CADENA\_W'LENGTH+1);

    CADENA\_W := "dataOUT";

    write(LINEA\_RES, CADENA\_W, right, CADENA\_W'LENGTH+1);

    writeline(ARCH\_RES,LINEA\_RES);

  wait for 100 ns;

  FOR I IN 0 TO 11 LOOP

        readline(ARCH\_VEC,LINEA\_VEC);

        Hread(LINEA\_VEC, VAR\_RADDRS);

        addrs <= VAR\_RADDRS;

        read(LINEA\_VEC, VAR\_RWD);

        wd <= VAR\_RWD;

        Hread(LINEA\_VEC, VAR\_RDATAIN);

        dataIn <= VAR\_RDATAIN;

        WAIT UNTIL RISING\_EDGE(CLK);

        VAR\_Dataout := dataOut;

        Hwrite(LINEA\_RES, VAR\_RADDRS,   right, 5);

        write(LINEA\_RES, VAR\_RWD,   right, 5);

        Hwrite (LINEA\_RES, VAR\_RDATAIN, right, 7);

        Hwrite (LINEA\_RES, VAR\_DATAOUT,     right, 8);

        writeline(ARCH\_RES,LINEA\_RES);

    end loop;

    file\_close(ARCH\_VEC);

    file\_close(ARCH\_RES);

*-- EDIT Add stimuli here*

    wait for CLK\_period\*10;

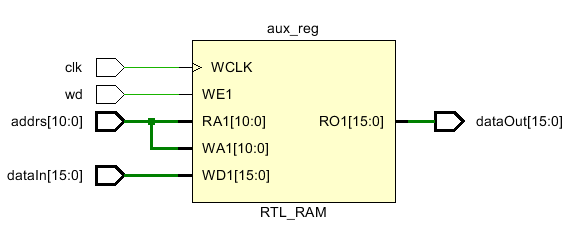
*-- Stop the clock and hence terminate the simulation*

    wait;

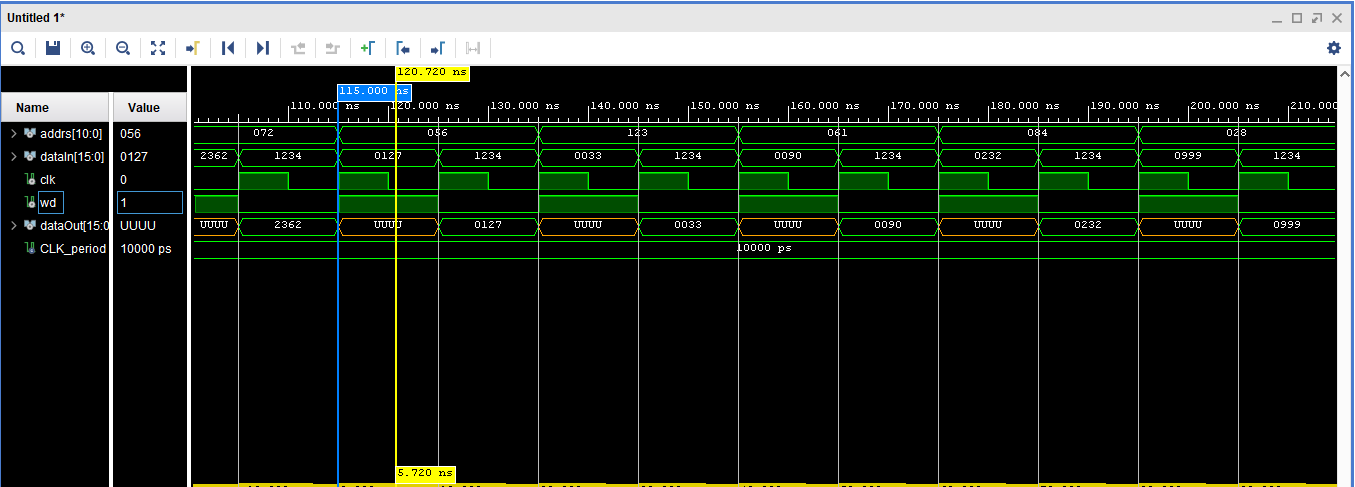
end process;

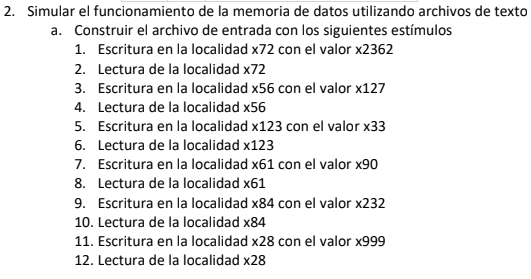
end tb;

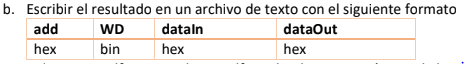
1. **Diagrama RTL**



1. **Forma de onda de simulación**







Al introducir las instrucciones en el archivo ***entradas.txt***, obtenemos los siguientes resultados en el archivo de salida ***resultados.txt.***

