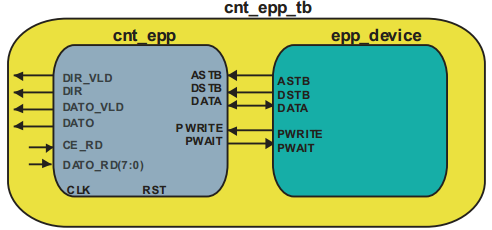
2014/2015

Modelado y síntesis de sistemas electrónicos digitales

Practica Laboratorio

christopher HYDE PEINADO 09046047D

**1.1.1.-** Código VHDL de la entidad ***cnt\_epp*** y del banco de pruebas utilizado en su simulación.  
También se debe proporcionar el código de la entidad *epp\_device*.



**A)** Código VHDL de la entidad ***cnt\_epp***:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity cnt\_epp is

port (

CLK : in std\_logic;

RST : in std\_logic;

ASTRB : in std\_logic;

DSTRB : in std\_logic;

DATA : inout std\_logic\_vector(7 downto 0);

PWRITE : in std\_logic;

PWAIT : out std\_logic;

DATO\_RD : in std\_logic\_vector(7 downto 0);

CE\_RD : out std\_logic;

DIR : out std\_logic\_vector (7 downto 0);

DIR\_VLD : out std\_logic;

DATO : out std\_logic\_vector (7 downto 0);

DATO\_VLD : out std\_logic);

end ;

architecture rtl of cnt\_epp is

signal S1:std\_logic;

signal S11:std\_logic;

signal Q:std\_logic;

signal S2:std\_logic;

signal S22:std\_logic;

signal Q2:std\_logic;

begin

------------------ADDRESS---------------------------------

AddrBiestableD1:process (CLK,RST)

begin

if (RST='1') then

Q <= '0';

elsif (CLK'event and CLK='1')then

Q <= ASTRB;

end if;

end process;

S1<= ASTRB and not Q;

S11<=S1 and not PWRITE;

AddrBiestableD2:process (CLK,RST)

begin

if (RST='1') then

DIR\_VLD <= '0';

elsif (CLK'event and CLK='1')then

DIR\_VLD <= S11;

end if;

end process;

AddrBiestableD3:process (CLK,RST,S11)

begin

if (RST='1') then

DIR <= (others=>'0');

elsif (CLK'event and CLK='1')then

if (S11='1') then

DIR <= DATA;

end if;

end if;

end process;

-----------------DATA------------------------------------

DataBiestableD1:process (CLK,RST)

begin

if (RST='1') then

Q2 <= '0';

elsif (CLK'event and CLK='1')then

Q2 <= DSTRB;

end if;

end process;

S2<= DSTRB and not Q2;

S22<=S2 and not PWRITE;

DataBiestableD2:process (CLK,RST)

begin

if (RST='1') then

DATO\_VLD <= '0';

elsif (CLK'event and CLK='1')then

DATO\_VLD <= S22;

end if;

end process;

DataBiestableD3:process (CLK,RST,S22)

begin

if (RST='1') then

DATO <= (others=>'0');

elsif (CLK'event and CLK='1')then

if (S22='1') then

DATO <= DATA;

end if;

end if;

end process;

--tristate buffer

DATA<= DATO\_RD when ((PWRITE= '1') and (DSTRB='1')) else (others => 'Z');

PWAITbiestableD:process (CLK,RST)

begin

if (RST='1') then

PWAIT <= '0';

elsif (CLK'event and CLK='1')then

PWAIT <= (not(ASTRB)) and (not(DSTRB));

end if;

end process;

CE\_RDbiestableD:process (CLK,RST,PWRITE)

begin

if (RST='1') then

CE\_RD <= '1';

elsif (CLK'event and CLK='1')then

if (PWRITE='1') then

CE\_RD <= DSTRB;

end if;

end if;

end process;

end rtl;

**B)** Código VHDL de la entidad ***epp\_device*** modificado para realizar 3 escrituras y una lectura:

Process

procedure epp\_cicle ( address : in std\_logic\_vector(7 downto 0);

data\_io : inout std\_logic\_vector(7 downto 0);

r\_w : in character) is

begin

wait until clk\_epp = '1';

PWRITE <= '0';

wait until clk\_epp = '1';

ASTRB <= '0';

data <= address;

wait for T\_clk\_epp\*EPP\_cicle\_length;

ASTRB <= '1';

wait until clk\_epp = '1';

data <= (others => 'Z');

PWRITE <= '1';

wait until clk\_epp = '1';

wait for T\_clk\_epp\*EPP\_cicle\_length;

--------------------------------------------------------------------------

if r\_w = 'w' then -- write cicle

PWRITE <= '0';

data <= data\_io;

end if;

--------------------------------------------------------------------------

wait until clk\_epp = '1';

DSTRB <= '0';

wait for T\_clk\_epp\*EPP\_cicle\_length;

--------------------------------------------------------------------------

if r\_w = 'r' then -- read cicle

data\_io:= data;

end if;

--------------------------------------------------------------------------

DSTRB <= '1';

wait until clk\_epp = '1';

data <= (others => 'Z');

PWRITE <= '1';

wait until clk\_epp = '1';

end procedure;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use std.textio.all;

use ieee.std\_logic\_textio.all;

entity epp\_device is

port (

DATA : inout std\_logic\_vector(7 downto 0);

PWRITE : out std\_logic;

DSTRB : out std\_logic;

ASTRB : out std\_logic;

PWAIT : in std\_logic);

end epp\_device;

architecture sim of epp\_device is

constant T\_clk\_epp : time := 100 ns; -- Internal clock period.

signal clk\_epp : std\_logic := '0'; -- Internal clock signal.

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

constant dir\_frec : std\_logic\_vector( 7 downto 0) := x"F0";

constant dir\_dpram1 : std\_logic\_vector( 7 downto 0) := x"A1";

constant dir\_dpram2 : std\_logic\_vector( 7 downto 0) := x"A2";

constant EPP\_cicle\_length: natural:= 10;

begin

-- internal clock signal generation.

clk\_epp <= not(clk\_epp) after T\_clk\_epp/2;

file arch\_in : text ;

variable bf : line;

variable dato : std\_logic\_vector(7 downto 0);

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

begin

--inicialización

data <= (others => 'Z');

PWRITE <= '1';

DSTRB <= '1';

ASTRB <= '1';

dir := (others => '0');

-----------------first write values----------

wait for 130 ns;

DIR:=dir\_dpram1;

DATO:=X"34";

epp\_cicle ( address => dir,

data\_io => dato,

r\_w => 'w');

-----------------second write values----------

wait for 130 ns;

DIR:=dir\_dpram2;

DATO:=X"44";

epp\_cicle ( address => dir,

data\_io => dato,

r\_w => 'w');

-----------------third write values----------

wait for 130 ns;

DIR:=dir\_frec;

DATO:=X"54";

epp\_cicle ( address => dir,

data\_io => dato,

r\_w => 'w');

-----------------first read values----------

wait for 130 ns;

DIR:=X"FF";

epp\_cicle ( address => dir,

data\_io => dato,

r\_w => 'r');

read\_value<=dato;

wait for 1 us;

report "FIN CICLO R/W" severity failure;

end process;

end sim;

**C)** Código VHDL del testbench de simulación ***cnt\_epp\_tb:***

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: cnt\_epp PORT MAP (

CLK => CLK,

RST => RST,

ASTRB => ASTRB\_comm,

DSTRB => DSTRB\_comm,

DATA => DATA\_comm,

PWRITE => PWRITE\_comm,

PWAIT => PWAIT\_comm,

DATO\_RD => DATO\_RD\_comm,

CE\_RD => CE\_RD,

DIR => DIR,

DIR\_VLD => DIR\_VLD,

DATO => DATO,

DATO\_VLD => DATO\_VLD

);

eppDevice:epp\_device

port map(

DATA=>DATA\_comm,

PWRITE=>PWRITE\_comm,

DSTRB=>DSTRB\_comm,

ASTRB=>ASTRB\_comm,

PWAIT=>PWAIT\_comm

);

-- Clock process definitions

CLK\_process :process

begin

CLK <= '0';

wait for CLK\_period/2;

CLK <= '1';

wait for CLK\_period/2;

end process;

RST\_process :process

begin

wait for 5 ns;

RST <= '0';

end process;

END rtl;

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY cnt\_epp\_tb IS

END cnt\_epp\_tb;

ARCHITECTURE rtl OF cnt\_epp\_tb IS

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

COMPONENT cnt\_epp

PORT(

CLK : IN std\_logic;

RST : IN std\_logic;

ASTRB : IN std\_logic;

DSTRB : IN std\_logic;

DATA : INOUT std\_logic\_vector(7 downto 0);

PWRITE : IN std\_logic;

PWAIT : OUT std\_logic;

DATO\_RD : IN std\_logic\_vector(7 downto 0);

CE\_RD : OUT std\_logic;

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

DIR\_VLD : OUT std\_logic;

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

DATO\_VLD : OUT std\_logic

);

END COMPONENT;

COMPONENT epp\_device

port (

DATA : inout std\_logic\_vector(7 downto 0);

PWRITE : out std\_logic;

DSTRB : out std\_logic;

ASTRB : out std\_logic;

PWAIT : in std\_logic);

END COMPONENT;

signal ASTRB\_comm:std\_logic;

signal DSTRB\_comm:std\_logic;

signal PWRITE\_comm:std\_logic;

signal PWAIT\_comm:std\_logic;

signal DATA\_comm:std\_logic\_vector(7 downto 0);

--cnt\_epp

signal DATO\_RD\_comm :std\_logic\_vector(7 downto 0):=x"AA";

signal CE\_RD :std\_logic;

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

signal DIR\_VLD : std\_logic;

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

signal DATO\_VLD :std\_logic;

-- Clock period definitions

constant CLK\_period : time := 10 ns;

signal clk : std\_logic;

signal rst : std\_logic:='1';