1) Identifique qué elementos constituyen los LEs de la FPGA Cyclone III y qué estructura tienen las LABs

Las LEs contienen: una LUT (Look-Up Table) que cumple la función de implementar funciones lógicas combinacionales, una lógica de acarreo, multiplexores internos, y una serie de flip-flops tipo D generalmente que sirven para dar un comportamiento secuencial.

Las LABs, son agrupaciones de LEs que comparten recursos, permiten brindar comportamientos mas complejos a los sistemas que el usuario quiera generar

2) ¿De qué se trata el Nios® II?

Es el procesador que se encuentra dentro de las FPGA, que permiten que estas sean configurables y programables. Es un procesador de 32 bits.

3) ¿Qué diferencia existe entre IP cores y los bloques embebidos (ej multiplicador embebido) disponibles en la FPGA?

La diferencia es que los IP cpres son bloques funcionales diseñados previamente, que se configuran por software. Mientras tanto, los bloques embebidos disponibles, son bloques físicos ya dispuestos en la misma FPGA, lo que hace que sean mas rapidos y eficientes.

4) ¿Qué tipo de celda de programación posee el dispositivo FPGA Cyclone III?

Posee una celda de programación del tipo SRAM. Tiene la característica de tener una memoria externa.

5) Realice la descripción en VHDL de un Flip Flop JK.

6) Realice la descripción en VHDL de un sumador completo de un bit.

library ieee;

use ieee.std\_logic\_1164.all;

entity Sumador\_Completo is

port( A: in std\_logic;

B: in std\_logic;

Cin: in std\_logic;

Cout: out std\_logic;

Sum: out std\_logic);

end Sumador\_Completo;

architecture comportamiento of Sumador\_Completo is

Signal Sig1,Sig2,Sig3,Sig4: std\_logic;

begin

Sig1 <= B xor Cin;

Sum <= Sig1 xor A;

Sig2 <= Cin and B;

Sig3 <= A and B;

Sig4 <= A and Cin;

Cout<= Sig2 or Sig3 or Sig4;

end comportamiento;

7) Realice la descripción en VHDL del test bench del sumador completo de un bit.

library ieee;

use ieee.std\_logic\_1164.all;

entity TB\_Sumador\_Completo is

end TB\_Sumador\_Completo;

architecture comportamiento of TB\_Sumador\_Completo is

component Sumador\_Completo

port(

A : in std\_logic;

B : in std\_logic;

Cin : in std\_logic;

Cout : out std\_logic;

Sum : out std\_logic

);

end component;

signal A, B, Cin : std\_logic := '0';

signal Sum, Cout : std\_logic;

begin

uut: Sumador\_Completo

port map (

A => A,

B => B,

Cin => Cin,

Cout => Cout,

Sum => Sum

);

stim\_proc: process

begin

A <= '0'; B <= '0'; Cin <= '0'; wait for 10 ns;

A <= '0'; B <= '0'; Cin <= '1'; wait for 10 ns;

A <= '0'; B <= '1'; Cin <= '0'; wait for 10 ns;

A <= '0'; B <= '1'; Cin <= '1'; wait for 10 ns;

A <= '1'; B <= '0'; Cin <= '0'; wait for 10 ns;

A <= '1'; B <= '0'; Cin <= '1'; wait for 10 ns;

A <= '1'; B <= '1'; Cin <= '0'; wait for 10 ns;

A <= '1'; B <= '1'; Cin <= '1'; wait for 10 ns;

wait;

end process;

end comportamiento;