1)a) Ambas son SOP.

b) Matriz de entrada AND programable y OR de salida fija, pero la OR no es reprogamable.

c)Depende del integrado.

d)Solo la GAL es reprogramable.

2)a) Falso ya solo se cumple para las Full Custom.

b) Verdadero.

c) Falso, el FPGA.

d) Verdadero.

e) Verdadero.

f) Verdadero.

g) Verdadero.

h) Falso, al contrario.

i) Falso.

3)a) LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY sumador\_completo IS

PORT

( a : IN STD\_LOGIC;

b : IN STD\_LOGIC;

carry\_in : IN STD\_LOGIC;

suma : OUT STD\_LOGIC;

carry\_out : OUT STD\_LOGIC );

END sumador\_completo;

ARCHITECTURE behavioral OF sumador\_completo IS

BEGIN

PROCESS (a, b, carry\_in)

BEGIN

suma <= a XOR b XOR carry\_in;

carry\_out <= (a AND b) OR (carry\_in and(a xor b));

END PROCESS;

END behavioral;

b) LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY sumador\_completo IS

PORT (

a : IN STD\_LOGIC;

b : IN STD\_LOGIC;

carry\_in : IN STD\_LOGIC;

suma : OUT STD\_LOGIC;

carry\_out: OUT STD\_LOGIC );

END sumador\_completo;

ARCHITECTURE behavioral OF sumador\_completo

IS

BEGIN

WITH (a & b & carry\_in) SELECT

suma <=

'0' WHEN "000", -- a=0, b=0, carry\_in=0

'1' WHEN "001", -- a=0, b=0, carry\_in=1

'1' WHEN "010", -- a=0, b=1, carry\_in=0

'0' WHEN "011", -- a=0, b=1, carry\_in=1

'1' WHEN "100", -- a=1, b=0, carry\_in=0

'0' WHEN "101", -- a=1, b=0, carry\_in=1

'0' WHEN "110", -- a=1, b=1, carry\_in=0

'1' WHEN "111"; -- a=1, b=1, carry\_in=1

WITH (a & b & carry\_in) SELECT

carry\_out <=

'0' WHEN "000", -- a=0, b=0, carry\_in=0

'0' WHEN "001", -- a=0, b=0, carry\_in=1

'0' WHEN "010", -- a=0, b=1, carry\_in=0

'1' WHEN "011", -- a=0, b=1, carry\_in=1

'0' WHEN "100", -- a=1, b=0, carry\_in=0

'1' WHEN "101", -- a=1, b=0, carry\_in=1

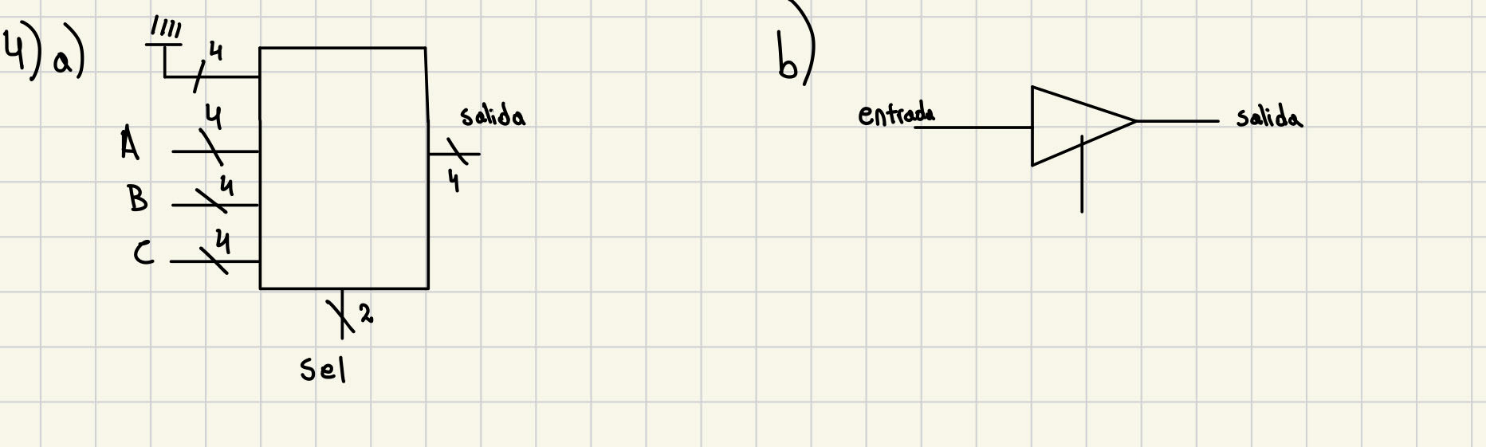
'1' WHEN "110", -- a=1, b=1, carry\_in=0

'1' WHEN "111"; -- a=1, b=1, carry\_in=1

END behavioral;

4) Se sintetiza un MUX con 4 entradas de 4 bits cada una, donde la primera entrada esta “conectada a tierra”. La segunda entrada es “a”, la tercera es” b” y la cuarta es “c”. La entrada se selección es “sel” y es de dos bits. Por otro lado, la salida también cuenta con 4 bits.

b) Se sintetiza un Buffer tri-state, es decir tiene un estado más que es el de alta impedancia.



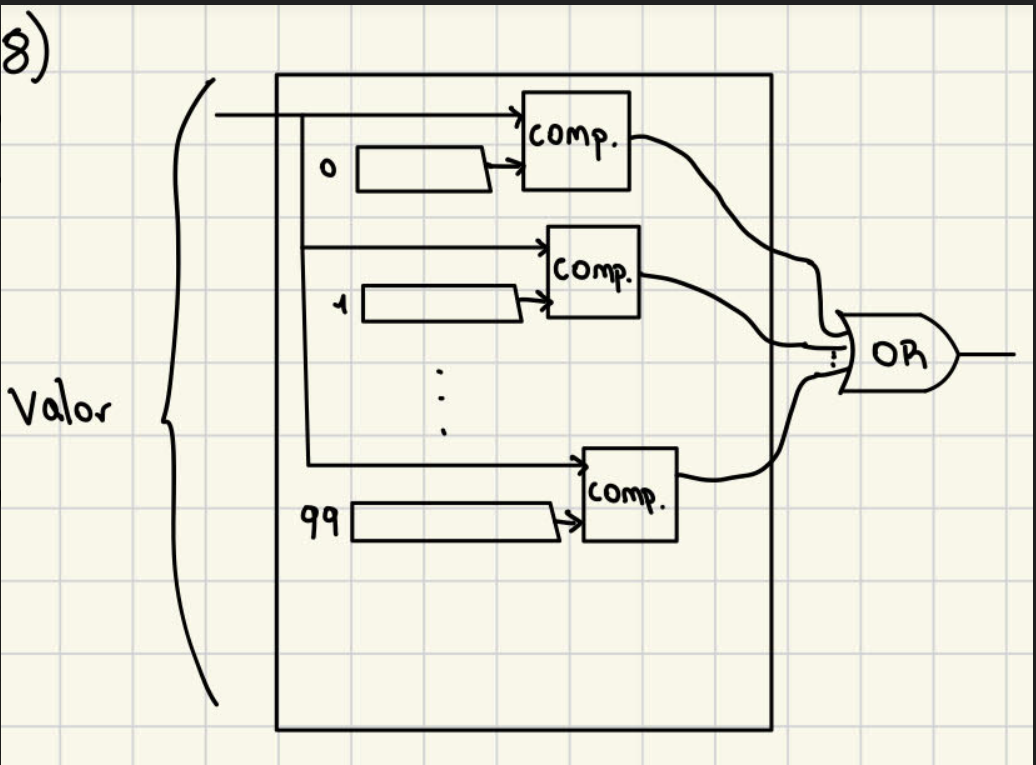
5) Una FPGA posee tres tipos principales de elementos programables: LUTs (Look-Up Tables), que implementan funciones lógicas combinacionales al almacenar tablas de verdad; Flip-Flops, que permiten lógica secuencial y almacenamiento de datos sincronizados por una señal de reloj; y una Matriz de Conexiones Programables, que interconecta las LUTs y flip-flops, permitiendo la configuración y comunicación entre los distintos bloques lógicos de la FPGA.

Un bloque lógico en una FPGA contiene los siguientes componentes básicos: una LUT (Look-Up Table), un Flip-Flop, Multiplexores, acarreo y Conexiones Programables.

6) Un lenguaje de descripción de hardware (HDL), como VHDL o Verilog, se utiliza para modelar y describir circuitos electrónicos y su funcionamiento, permitiendo definir conexiones y operaciones de hardware digital. A diferencia de los lenguajes de programación tradicionales, que se enfocan en describir algoritmos secuenciales ejecutados por un procesador, los HDL describen el hardware. Además, los HDL permiten simular y luego sintetizar el diseño para implementarlo físicamente en hardware, mientras que los lenguajes de programación se ejecutan en software.

7) Se sintetiza un decodificador de 3 a 8, es decir 3 entradas y 8 salidas.

8)



b) Es combinacional ya que cuenta con compuerta OR y comparadores, los cuales modifican su salida solo en función de la entrada en ese momento sin almacenar memoria o estados.

c) El tiempo mínimo es el tiempo que le lleva a un solo comparador comparar y a la compuerta OR cambiar su salida en función de la entrada, es decir hay dos retardos. La frecuencia máxima es la inversa de la suma de los dos retardos.

d) Al menos debería tener 100 bits ya que son todos los bits del valor de entrada.