

PRACTICA 1

“Modelado en VHDL de un Decodificador”

MEMORIA DESCRIPTIVA DE LOS RESULTADOS DE LA PRACTICA

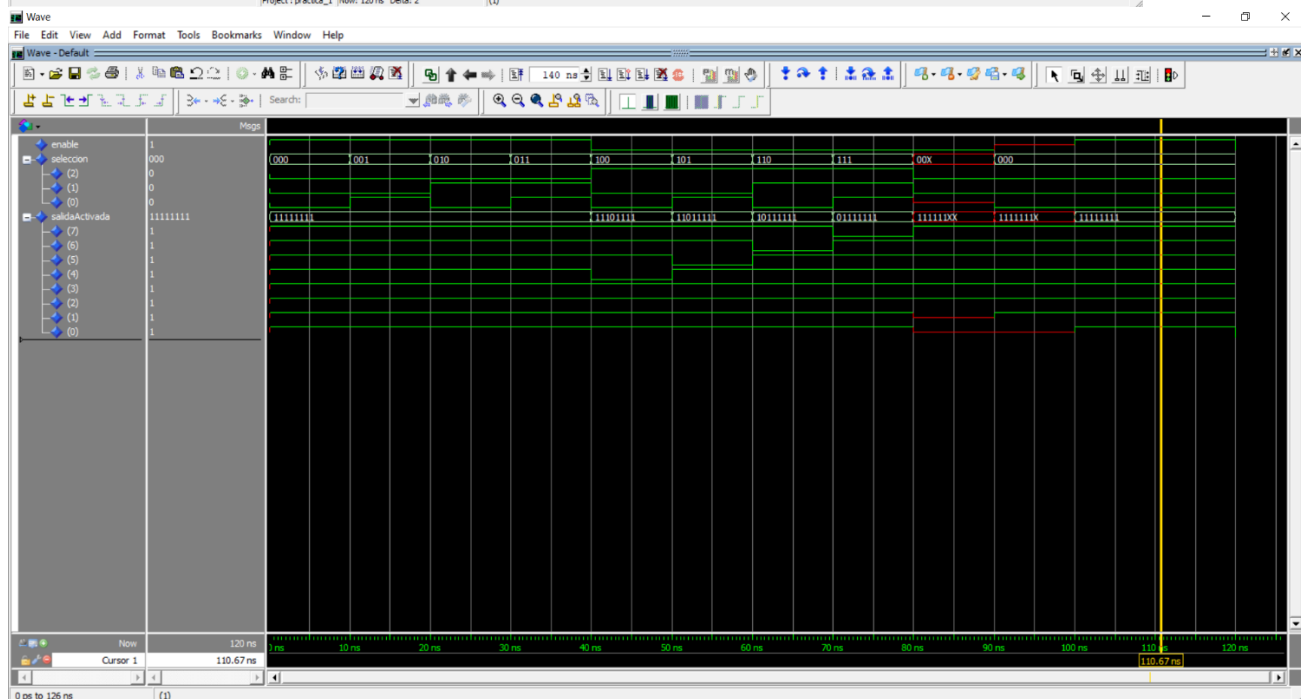
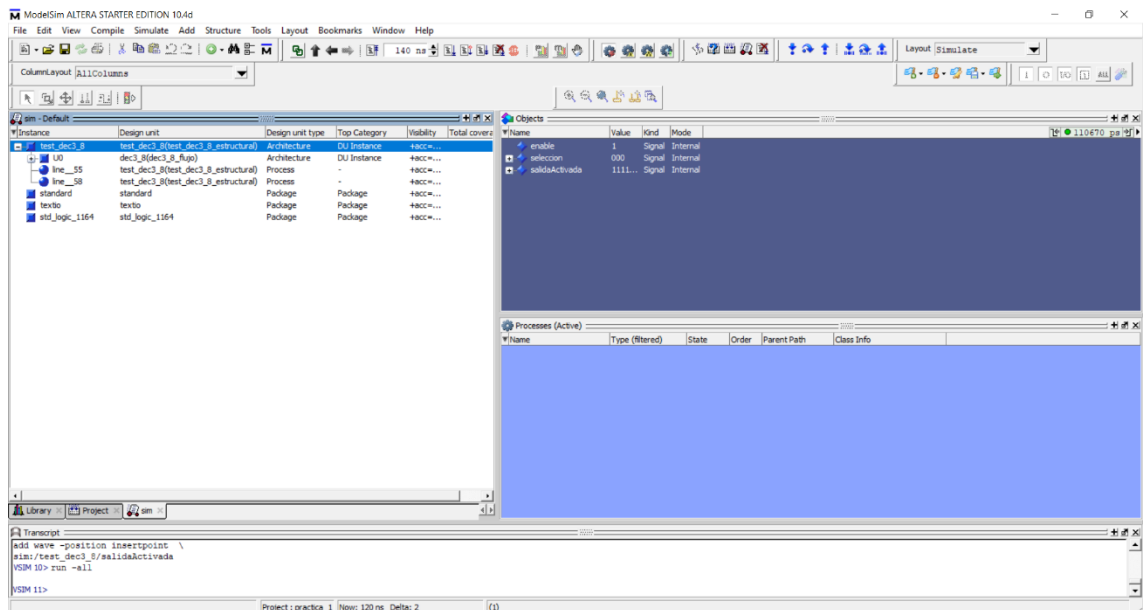
1. Ecuaciones lógicas de cada una de las salidas: S(0).....S(7)

S(0)	$(e \text{ OR } c(2) \text{ OR } c(1) \text{ OR } c(0))$
S(1)	$(e \text{ OR } c(2) \text{ OR } c(1) \text{ OR } (\text{NOT } c(0)))$
S(2)	$(e \text{ OR } c(2) \text{ OR } (\text{NOT } c(1)) \text{ OR } c(0))$
S(3)	$(e \text{ OR } c(2) \text{ OR } (\text{NOT } c(1)) \text{ OR } (\text{NOT } c(0)))$
S(4)	$(e \text{ OR } (\text{NOT } c(2)) \text{ OR } c(1) \text{ OR } c(0))$
S(5)	$(e \text{ OR } (\text{NOT } c(2)) \text{ OR } c(1) \text{ OR } (\text{NOT } c(0)))$
S(6)	$(e \text{ OR } (\text{NOT } c(2)) \text{ OR } (\text{NOT } c(1)) \text{ OR } c(0))$
S(7)	$(e \text{ OR } (\text{NOT } c(2)) \text{ OR } (\text{NOT } c(1)) \text{ OR } (\text{NOT } c(0)))$

*Las ecuaciones lógicas de este apartado son las que hemos implementado a nuestro código de modelsim, si bien se quisieran representar las ecuaciones sin los operadores lógicos OR y NOT se cambiarían por +/OR y (/)/NOT.

2. Captura de pantalla con los resultados de la simulación del circuito (visualizándose todas las entradas y salidas, sin que aparezcan los nombres cortados). Para el vector correspondiente a las entradas de selección y el asociado a las salidas se recomienda elegir la opción de representación como valores binarios [En la ventana Wave se seleccionan las señales anteriores y pulsamos el botón derecho del ratón.

A continuación, en el menú contextual elegimos la opción *Radix* y, por último, *Binary*].



MÁS AMPLIADA:



- Breve análisis de los resultados anteriores donde se justifiquen los diferentes valores que toman las salidas y, además, se verifique que la implementación desarrollada para el decodificador cumple de forma exacta las especificaciones descritas anteriormente. Comentar en detalle los resultados entre los 70 y los 130 ns.

Acorde a las gráficas aportadas en el apartado anterior, nos hemos dado cuenta de que en ciertos intervalos de tiempos no se dan los resultados obtenidos, y esto se debe al código implementado en la arquitectura del tester previamente descargado de la plataforma Moodle y adjuntadas las capturas de dicha parte del código al final de este apartado.

Como se observar en nuestro apartado 1, hemos implementado las ecuaciones lógicas de las salidas como suma de productos en vez de producto de sumas que es como lo representa la herramienta en la ventana de “Wave” donde se representan las gráficas asociadas a cada uno de los valores lógicos de las salidas. En nuestro código al estar expresado como suma de productos todos los enable están a 1 y por tanto los resultados esperados según la rúbrica o tabla de los valores lógicos de las salidas adjuntada al final del apartado, pues solo van a cambiar a los valores esperados cuando el enable sea 0, pero en el código del tester especifica que empiece en 1 y a partir de 40 ns la señal enable sea 0, por tanto, de 0 ns a 40 ns el valor de todas las salidas va a ser 1.

A partir 40 ns todas las salidas toman los valores correspondientes a los valores lógicos de las señales de control hasta los 90 ns donde la señal enable toma 'X' y todas las salidas a 1 a excepción de la s(0) y s(1) que toman 'X'. A partir de los 100 ns la señal enable otra vez es 1 y por tanto todas las salidas a 1 y por último al llegar a los 120 ns la señal enable toma 0 pero como las señales de control toman a partir de los 90 ns los valores 000 (establecido por el tester) pues las salidas toman todas 1 a excepción de la s(0) que toma 0* acorde con la tabla de valores lógicos de las salidas.

En concreto de los 70 ns a los 130 ns, de 70 - 80 ns la señal enable es 0, las señales de control toman 111 y las salidas todas a 1 a excepción de la s(7) que está a 0. En los siguientes 10 ns (80-90ns) la señal enable sigue a 0 pero las señales de control pasan a ser 00X por tanto las salidas pasan todas a 1 excepción de s(0) y s(1) que toman 'X'. De los 90 a los 100 ns la señal enable cambia a 'X' entonces como consecuencia las señales de control pasan a 000 y las salidas pasan a 1 a excepción de la s(0) que en vez de tomar 0 debido a que el valor del enable es 'X' pasa a ser también 'X'. De 100 a 120 ns la señal enable pasa a ser de nuevo 1 entonces las salidas pasaran a ser todas 1. Y por último del tramo de 120-130 ns la señal enable toma 0 las señales de control 000 y por tanto acorde a la tabla de valores lógicos las salidas toman todas 1 a excepción de la s(0) que toma 0*.

e	c(2)	c(1)	c(0)	s(7)	s(6)	s(5)	s(4)	s(3)	s(2)	s(1)	s(0)
0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	0	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1
1	x	x	x	1	1	1	1	1	1	1	1

```

BEGIN

-- Conexionado entre los puertos del componente y las señales externas

U0: Dec3_8 PORT MAP(e=>enable, c=>seleccion, s => salidaActivada);

-- Drivers de las señales de entrada
-- Habilitación del chip

enable<= '1', '0' AFTER 40 ns, 'X' AFTER 90 ns, '1' AFTER 100 ns, '0' AFTER 120 ns;

-- Seleccion de salida

seleccion<= "000", "001" AFTER 10 ns, "010" AFTER 20 ns, "011" AFTER 30 ns, "100" AFTER 40 ns, "101" AFTER 50 ns,
"110" AFTER 60 ns, "111" AFTER 70 ns, "00X" AFTER 80 ns, "000" AFTER 90 ns;

END Test_Dec3_8_Estructural;
```

Código del tester que condiciona las salidas en los ns correspondientes.

- En el fichero de test proporcionado, indicar que cambios habría que hacer en el código, si la señal que se conecta a la salida del componente se llamase salidadecodificador.

Lo primero se tendría que cambiar sería la entidad para llamarse s: OUT STD_LOGIC_VECTOR (7 DOWNT0 0); SIGNAL salidadecodificador: std_logic_vector (7 DOWNT0 0); = "11111111"; y después especificar (s=> salidadecodificador) junto con las señales de control y la señal enable.

Y ya no haría falta hacer más cambios en el código tester ya que no se desea cambiar los valores del enable ni los valores de las señales de control en los ns específicos simplemente renombrar la señal que se conecta a la salida del componente.

ASÍ QUEDARÍA EL CÓDIGO VHDL DEL TESTER: (cambios en negrita)

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY Test_Dec3_8 IS
END Test_Dec3_8;

ARCHITECTURE Test_Dec3_8_Estructural OF Test_Dec3_8 IS

-- Parte de declarativa de la arquitectura
-- Declaracion del componente a testear

    COMPONENT Dec3_8
        PORT (e: IN std_logic; c: IN std_logic_vector (2 DOWNTO 0);
s: OUT std_logic_vector (7 DOWNTO 0));
    END COMPONENT;

-- Declaracion de las sennales externas que se conectan a los puertos
del componente

    SIGNAL enable: std_logic;
    SIGNAL seleccion: std_logic_vector (2 DOWNTO 0);
    SIGNAL salidadecodificador: std_logic_vector (7 DOWNTO 0) :=
"11111111";

BEGIN

-- Conexionado entre los puertos del componente y las sennales
externas

    U0: Dec3_8 PORT MAP(e=>enable, c=>seleccion,
s => salidadecodificador);

-- Drivers de las sennales de entrada

-- a). - Habilitacion del decodificador
    enable<= '1', '0' AFTER 40 ns, 'X' AFTER 90 ns, '1' AFTER 100 ns,
'0' AFTER 120 ns;

-- b). - Seleccion de salida
    seleccion<= "000", "001" AFTER 10 ns, "010" AFTER 20 ns, "011"
AFTER 30 ns, "100" AFTER 40 ns, "101" AFTER 50 ns, "110" AFTER 60 ns,
"111" AFTER 70 ns, "00X" AFTER 80 ns, "000" AFTER 90 ns;

END Test_Dec3_8_Estructural;

```

5. Observaciones:

*La única observación por mencionar es que al simular y correr el código del decodificador en 140 ns cuando muestra la gráfica en la ventana de “Wave” solo representa los valores de las señales hasta 120 ns. Dado que según el código del tester a partir de los 120 ns la señal enable toma 0 y las señales de control a partir de 90 ns toman 000 pues todas las salidas tomarían 1 a excepción de la s(0) que

tomaría 0 pero la herramienta por mucho que incrementes los ns no representa más de los 120 ns, pero si lo hiciese las salidas tomarían esos valores como también se menciona en apartado 3.