

Práctica 2: Captura, simulación e implementación en CPLD de un sistema combinacional

Universidad de Alcalá

Curso Académico 2019/2020
Curso 1º – Cuatrimestre 2º

Alumno(s)	Grupo	Puesto

Índice:

1	INTRODUCCIÓN.....	3
2	OBJETIVOS.....	3
3	RESUMEN DE LA PRÁCTICA.....	3
4	CONCEPTOS TEÓRICOS.....	4
5	DESARROLLO DE LA PRÁCTICA.....	5
5.1	DESCRIPCIÓN DEL SISTEMA y metodología de diseño.....	5
5.2	metodología de diseño.	5
5.3	REALIZACIÓN DEL DISEÑO.....	7
5.3.1.	Módulo MULTIPLEXOR.	7
5.3.2.	Módulo selector del mayor o menor entre dos números.....	9
5.3.3.	visualización del mayor o menor entre dos números.....	11
5.3.4.	diseño completo. esquema final.	12
5.4	PROGRAMACIÓN DEL CPLD Y PRUEBAS SOBRE LA PLACA.....	13

1 INTRODUCCIÓN

En esta práctica se realiza el diseño circuito electrónico digital combinacional mediante una herramienta de diseño asistido por ordenador (*CAD-Computer-Aided Design*), utilizado para configurar un dispositivo lógico programable, en particular un CPLD (*Complex Programmable Logic Device*). Con ello se pretende avanzar en la comprensión del análisis y diseño de circuitos digitales explicados en las sesiones de teoría. El uso de estas herramientas junto con los nuevos dispositivos se traduce en una evolución en la metodología de diseño.

Este guion incluye inicialmente una explicación general del sistema global y a continuación una descripción de cada uno de los bloques que lo componen, con el fin de servir de guía al alumno. No obstante, en alguno de los puntos del guion, se deja al alumno la tarea de completar la información necesaria, porque se considera que ese esfuerzo favorece la mejor comprensión del trabajo realizado.

A lo largo de la práctica se han incluido varias cuestiones que el alumno debe resolver, y que tienen un doble objetivo: servir al alumno de ayuda para ir alcanzando resultados intermedios hasta completar la práctica en su totalidad, y también como herramienta de autoevaluación para comprobar el nivel alcanzado en la realización de dicha práctica.

2 OBJETIVOS

El objetivo principal de esta práctica consiste en la iniciación en el diseño y análisis de circuitos digitales mediante herramientas CAD, y familiarización con la metodología de diseño (captura, simulación, implementación y descarga en placa) de circuitos digitales. Además, se pretende alcanzar los objetivos recogidos en varias competencias de la asignatura, con especial énfasis en los siguientes:

- Profundizar en la capacidad de análisis y diseño de circuitos combinacionales, mediante la aplicación práctica de los conocimientos adquiridos en las sesiones teóricas.
- Adquirir la capacidad de comprender y dominar los principios básicos de la electrónica digital y de las tecnologías de dispositivos digitales utilizados en la actualidad en el diseño electrónico profesional.
- Tener una primera toma de contacto con los lenguajes de descripción hardware y con las herramientas de diseño para la simulación e implementación de circuitos combinacionales.

3 RESUMEN DE LA PRÁCTICA

La práctica 2 se basa en un diseño similar al de la práctica 1 pero utilizando la metodología de diseño basada en dispositivos de lógica programable e incluyendo algunos componentes que se modelan mediante VHDL. El objetivo final es la selección del mayor o menor entre dos números dados en binario natural de 4 bits y su representación en un *display* de 7 segmentos con punto decimal.

El sistema tiene las siguientes entradas y salidas (ver Figura 1):

- **A[3..0] y B[3..0]**: entradas de datos de 4 bits, en binario natural que, en el circuito final, tomarán el valor especificado por unos interruptores. Uno de los dos datos de cuatro bits se visualizará en un *display* de ánodo común.

- **M_m**: entrada de un bit que indica que el dato que se debe visualizar es el mayor de **A[3..0]** y **B[3..0]**, para **M_m=1** (H) o el menor de ambos en caso contrario, para **M_m=0** (L).
- Salidas de los segmentos del display, **SEG_A**, **SEG_B**, **SEG_C**, **SEG_D**, **SEG_E**, **SEG_F**, **SEG_G** junto con el punto decimal **DP**.

La visualización se realiza en formato decimal mostrándose las unidades en el *display* y las decenas en el punto decimal del mismo. Por ejemplo, para visualizar el dato 0101_b en el *display* se mostrará el 5 estando el punto decimal apagado. Si el número de entrada no representa un dígito BCD, en el *display* se mostrarán las unidades encendiéndose el punto decimal. Por ejemplo, si se considera el valor de entrada a representar como el 1100_b, en el *display* se mostrará el 2 y punto decimal estará encendido para indicar que las decenas son 1.

El diseño, una vez simulado e implementado, se va a programar en el dispositivo CPLD existente en la placa de desarrollo **EPM240Board** mostrada en la Figura 1, cuyo núcleo central es el CPLD **EPM240T100C4**.

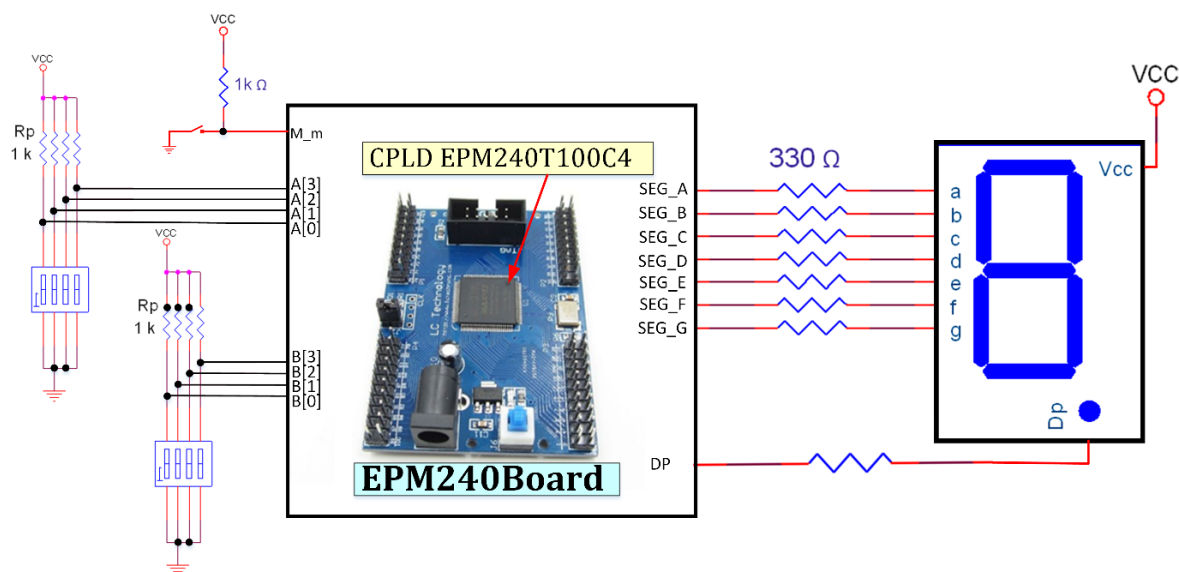


Figura 1.- Estructura del diseño.

Como se puede observar en la figura anterior, el valor de los datos **A[3..0]**, **B[3..0]** y **M_m** se seleccionarán con los *microswitches* de la placa utilizados en la práctica 1. Las salidas de los segmentos y el punto decimal, se conectarán, a través de resistencias, al *display* de la práctica 1.

Para la realización del diseño se va a utilizar el sistema de desarrollo *Quartus Prime Lite Edition* de Intel, empleándose para especificar la funcionalidad del mismo, tanto bloques basados en esquemas como en código VHDL. De este modo el alumno podrá comparar las dos estrategias de diseño. Por tanto, es imprescindible haber realizado el tutorial de la herramienta de desarrollo.

4 CONCEPTOS TEÓRICOS

Los conceptos teóricos esenciales para poder llevar a cabo con éxito el desarrollo de esta práctica, son los siguientes:

- Fundamentos de lógica combinacional.
- Conocimiento de los circuitos combinacionales básicos y de aritmética binaria (multiplexores, comparadores, decodificadores, sumadores, etc...).
- Conexiones básicas de un *display* de 7 segmentos.
- Codificación en binario y en BCD.
- Conocimiento básico del modelado de circuitos combinacionales en VHDL.
- Metodología de diseño de circuitos digitales para dispositivos CPLDs.

5 DESARROLLO DE LA PRÁCTICA

A continuación, se muestra una descripción detallada del diseño a realizar, junto con los bloques que lo componen y la metodología a seguir que se basa en el flujo de diseño para dispositivos de lógica programable.

5.1 DESCRIPCIÓN DEL SISTEMA Y METODOLOGÍA DE DISEÑO.

En la Figura 2 se muestran los bloques que componen el diseño donde se han omitido las conexiones entre ellos. Las entradas y salidas del mismo ya han sido explicadas en el apartado 3. Sin entrar en detalles, que en apartados sucesivos se incluirán, el diseño consta de los siguientes bloques:

- Un multiplexor de dos entradas de datos de 4 bits. Para modelar este bloque se utiliza VHDL siendo el fichero ***mux4bit2to1.vhd*** el que contiene el código que describe su funcionalidad.
- Un comparador de dos números de 4 bits (***A[3..0]*** y ***B[3..0]***) con una única salida (***GT***) que se activa a nivel alto cuando la entrada ***A*** es mayor que ***B***. La funcionalidad del comparador se especifica en VHDL, siendo el fichero ***comp_gt.vhd*** el que contiene el código que lo modela. Como se puede ver en la Figura 2, este comparador se utiliza dos veces en el diseño final. La función de cada uno de ellos se explicará en detalle en los apartados siguientes, correspondientes al desarrollo de la práctica.
- Un circuito combinacional, (CC).
- Un sumador de dos números de cuatro bits basado en el circuito integrado 7483.
- Un decodificador BCD a 7 segmentos basado en el circuito integrado 7447.

El diseño completo se debe almacenar en un fichero de esquemas (***bdf***=*block design file*) con el mismo nombre que el del proyecto y con extensión ***bdf***. El nombre del proyecto debe ser de la forma ***PR2_GRPX_PY***. El término ***X*** se corresponde con el grupo de laboratorio e ***Y*** con el puesto dentro de laboratorio. Con respecto al puesto, cuando este sea menor que 10 se pondrá un cero delante. Por ejemplo, el nombre del proyecto ***PR2_GRPX1_P05*** indica que el proyecto pertenece a alumnos del **grupo C1** y **puesto P5** del laboratorio.

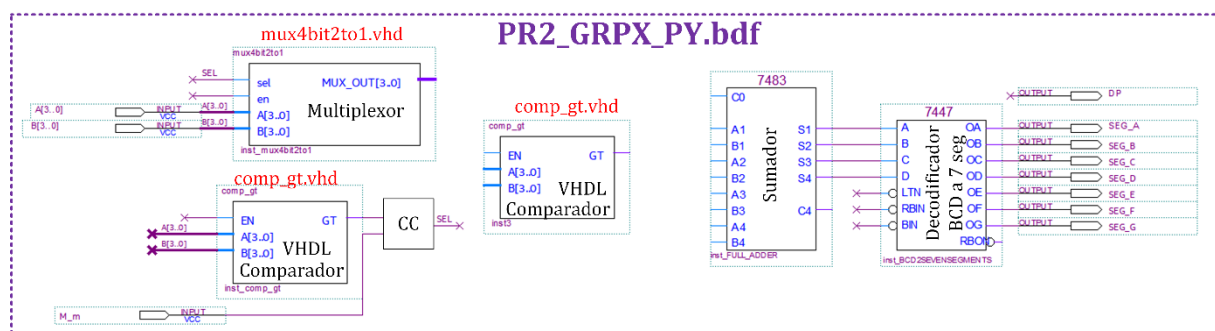


Figura 2.- Diagrama de bloques del diseño a realizar.

5.2 METODOLOGÍA DE DISEÑO.

La metodología de diseño se basa en el diagrama de flujo de la Figura 3 donde se incluyen las fases por las que el diseño debe pasar desde la especificación del mismo hasta su descarga en placa. Aunque esta metodología se explica en las clases de teoría, se ha incluido en este enunciado por conveniencia y claridad.

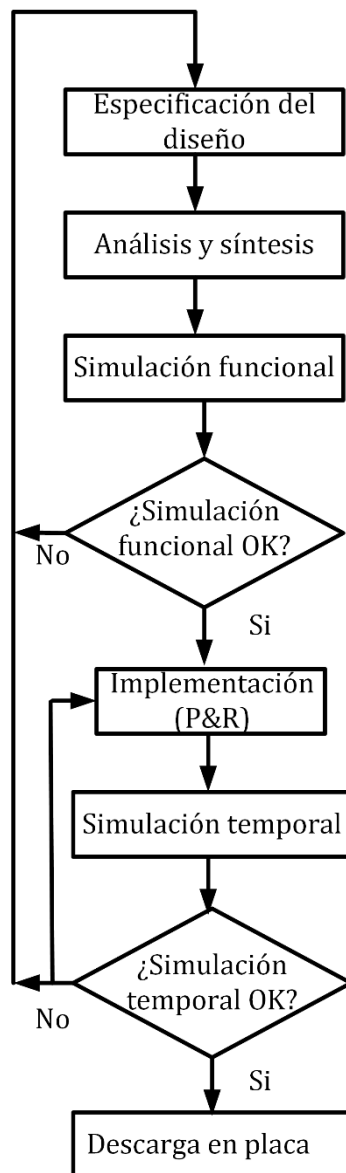


Figura 3.- Flujo de diseño para CPLDs.

Se trata del flujo de diseño que se utiliza para dispositivos programables en el que se comienza con la especificación del diseño, habitualmente utilizando lenguajes de descripción hardware. En el siguiente paso se realiza el análisis y la síntesis del modelo para verificar que las construcciones utilizadas son sintetizables, como paso previo a la simulación funcional. En ella se comprueba que el diseño funciona correctamente. Sin embargo, en esta fase de simulación, no se dispone de información de retardos. Para ello, se realiza la implementación, que a través de las fases de emplazamiento y ruteado (*Place and Route*) permite obtener información temporal precisa imprescindible para realizar una simulación más realista, ya que incluye los retardos que tienen los distintos elementos del circuito. La fase final del proceso de diseño consiste en la verificación de la funcionalidad en placa una vez corregidos todos los errores y comprobado que el diseño funciona de acuerdo a las especificaciones funcionales y cumpliendo las restricciones temporales si las hubiere. Finalmente, en el último apartado se realizará la descarga del diseño completo. Durante el desarrollo de la práctica, no todas las fases serán necesarias. En algunos casos, se llegará hasta la simulación funcional, mientras que en otros será obligatorio llevar a cabo la simulación temporal.

5.3 REALIZACIÓN DEL DISEÑO

Una vez descrito el funcionamiento del sistema y la metodología a seguir para completar la práctica, en este apartado se explican cada uno de los bloques que integran diseño junto con las cuestiones que alumno debe completar.

Cuestión 1. En primer lugar, imprime este guion de la práctica para cumplimentar las diferentes cuestiones planteadas en la misma.

Cuestión 2. Como primera fase del diseño se debe crear un proyecto con el entorno Quartus de nombre **PR2_GRPX_PY**. Como ya se ha explicado anteriormente, el termino **X** se corresponde con el grupo de laboratorio e **Y** con el puesto dentro de laboratorio. Por ejemplo, el nombre del proyecto **PR2_GRPX1_P05** indica que el proyecto pertenece a alumnos del **grupo C1** y **puesto P5** del laboratorio.

A la hora de crear el proyecto se deben tener en cuenta las siguientes consideraciones:

- Se debe crear previamente un directorio donde almacenar el proyecto.
- El nombre de la entidad de mayor nivel en la jerarquía de diseño será el mismo que el nombre del proyecto (ver Figura 2).
- Se debe crear un proyecto vacío al que inicialmente no se añadirá ningún fichero.
- En dispositivo CPLD a utilizar será el **MAXII EPM240T100C4**.
- Se debe configurar como herramienta de simulación *ModelSim-Altera* para VHDL, en la ventana *EDA Tool Settings*.

5.3.1. MÓDULO MULTIPLEXOR.

La funcionalidad del multiplexor se muestra en siguiente tabla.

Tabla 1.- Tabla de funcionamiento del bloque MULTIPLEXOR.

EN	SEL	MUX_OUT(3:0)
1	X	0000
0	0	A(3:0)
0	1	B(3:0)

El código VHDL que modela la funcionalidad del multiplexor se muestra en el Listado 1. Este código se debe almacenar en el fichero ***mux4bit2to1.vhd***.

```

library ieee;
use ieee.std_logic_1164.all;

entity mux4bit2to1 is
  port(
    sel: in std_logic;
    en: in std_logic;
    A: in std_logic_vector(3 downto 0);
    B: in std_logic_vector(3 downto 0);
    MUX_OUT: out std_logic_vector(3 downto 0));
end entity;

architecture for_mux4bit2to1 of mux4bit2to1 is
begin
  process (en, sel,A,B)
  begin
    if en='0' then
      case sel is
        when '0' => MUX_OUT <= A;
        when '1' => MUX_OUT <= B;
        when others => MUX_OUT <= "0000";
      end case;
    else
      MUX_OUT<= "0000";
    end if;
  end process;
end architecture;

```

Listado 1.- Código VHDL del multiplexor.

Cuestión 3. Crea un fichero nuevo para introducir el código del Listado 1. Almacena el fichero con el nombre ***mux4bit2to1.vhd***. Utilizando la opción de University Program VWF (Vector Waveform File), simula el código VHDL del multiplexor y completa el diagrama de formas de onda de la Figura 4 en la que se tendrá que definir el valor adecuado para las entradas **EN** y **SEL** para comprobar que el multiplexor funciona correctamente, es decir, que cumple la tabla de verdad anterior (Tabla 1). La salida a completar será **MUX_OUT**. Almacena el fichero ***vwf*** con el nombre ***mux4bit2to1.vwf***. Realiza solamente la simulación funcional.

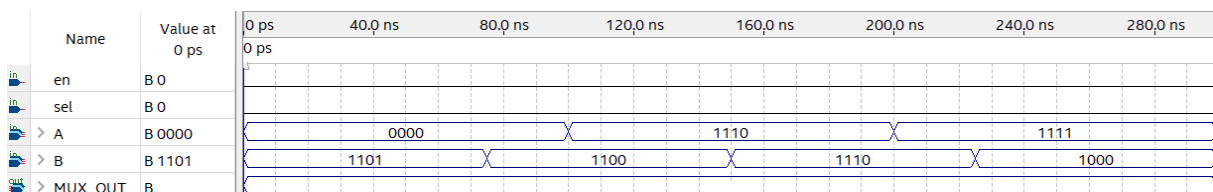


Figura 4.- Diagrama de formas de ondas para la simulación del multiplexor.

Cuando se compruebe que el modelo de simulación del multiplexor funciona adecuadamente crea un componente a través de la opción ***Create Symbol Files for Current File***. Para ello, dentro del proyecto, el fichero debe estar configurado con ***Set as Top-Level Entity***. El componente tendrá el mismo nombre que el fichero ***vhd*** donde se especifica su funcionalidad.

5.3.2. MÓDULO SELECTOR DEL MAYOR O MENOR ENTRE DOS NÚMEROS.

En este módulo se pretende simular el comportamiento funcional y temporal de módulo que se encarga de seleccionar el mayor o menor entre dos números, **A** y **B**, de 4 bits en función de una entrada de un bit, **M_m**. El esquema completo del módulo se muestra en la Figura 5 donde se puede observar que está formado por tres bloques: el multiplexor especificado en el apartado anterior, un comparador de dos números de 4 bits (**comp_gt**) y un circuito combinacional, **CC**.

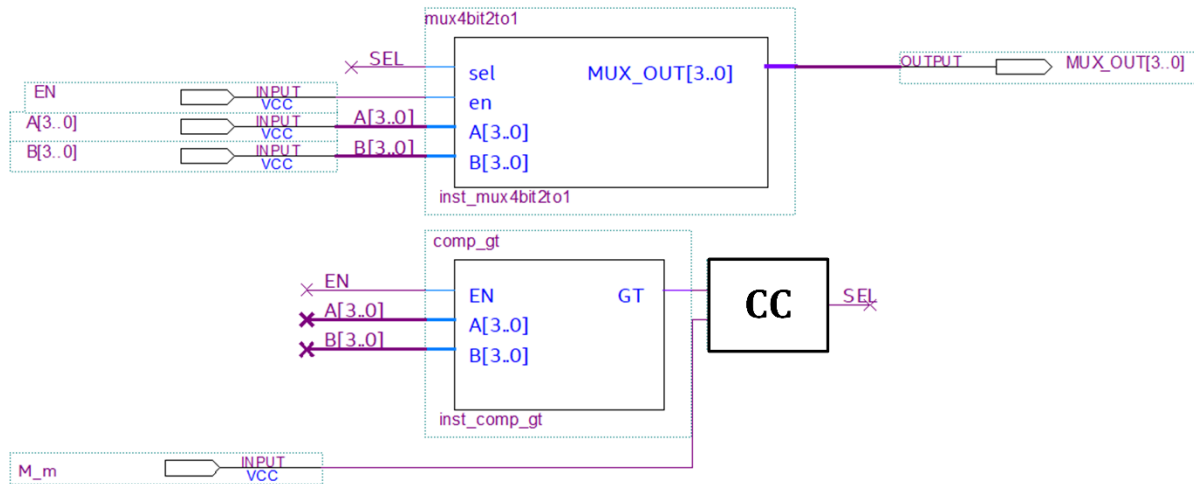


Figura 5.- Esquema del módulo selector del mayor o menor entre dos números.

Inicialmente, se debe especificar la funcionalidad del comparador de dos números de 4 bits, **A[3..0]** y **B[3..0]**, el cual dispone de una única salida (**GT**) que indica, a nivel alto, si **A > B**. El código VHDL incompleto que modela el comparador se muestra en el Listado 2. El fichero que contiene este código debe llevar como nombre **comp_gt.vhd**.

```
library ieee;
use ieee.std_logic_1164.all;
entity comp_gt is
port
(
    A: in std_logic_vector(3 downto 0);
    B: in std_logic_vector(3 downto 0);
    GT: out std_logic);
end entity;

architecture for_comp_gt of comp_gt is
begin
    process(
    )
    begin

        if
            then
                GT<= ;
            else
                GT<= ;
            end if;

        end process;
    end architecture;
```

Listado 2.- Código VHDL incompleto del comparador de dos número de 4 bits.

El alumno deberá modificar el código anterior incluyendo una entrada de habilitación, **EN**, que funcione de acuerdo a la siguiente tabla de verdad.

Tabla 2.- Tabla de funcionamiento del bloque comparador **comp_gt**.

EN	A/B	GT
1	X	0
0	$A > B$	1
0	$A \leq B$	0

Cuestión 4. Crea un nuevo fichero para introducir el código del Listado 2. Almacena el fichero con nombre **comp_gt.vhd**. Modifica el código del Listado 2 para que implemente la funcionalidad representada en la Tabla 2. Utilizando la opción de University Program VWF (Vector Waveform File), simula el código VHDL del comparador y completa el diagrama de formas de onda de la Figura 6. Almacena el fichero vwf con el nombre **comp_gt.vwf**. Realiza solamente la **simulación temporal**.

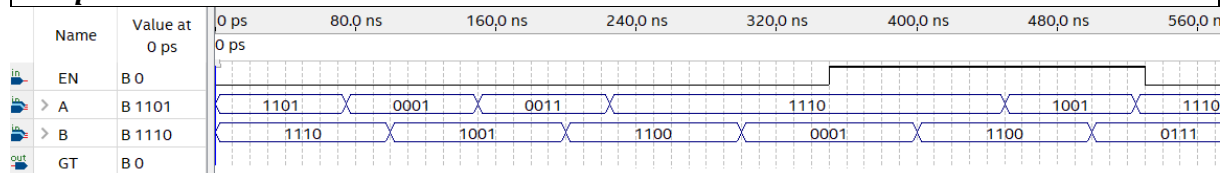


Figura 6.- Diagrama de formas de onda para la simulación del comparador **comp_gt**.

Cuestión 5. ¿Cuál es el retardo máximo del comparador? Razona la respuesta haciendo referencia al diagrama de tiempos de la Figura 6.

Como en el caso del multiplexor, cuando se compruebe que el modelo de simulación del comparador funciona adecuadamente, crea un componente a través de la opción **"Create Symbol Files for Current File"**. Para ello, dentro del proyecto, el fichero debe estar configurado con **"Set as Top-Level Entity"**.

Cuestión 6. Crea el esquema de la Figura 5 en el que tienes que añadir el circuito correspondiente al bloque **CC** y realiza la simulación funcional del mismo completando el diagrama de tiempos de la Figura 7. Ten en cuenta que cuando la entrada **M_m='1'**, la salida del multiplexor **MUX_OUT** debe ser el mayor de **A** y **B**. En caso contrario **MUX_OUT** debe tomar el valor del menor de ambas entradas. El nombre del fichero que almacena el esquema debe ser **PR2_MAYOR_MENOR.bdf**. Asimismo, almacena el fichero **vwf** con el nombre **PR2_MAYOR_MENOR.vwf**.

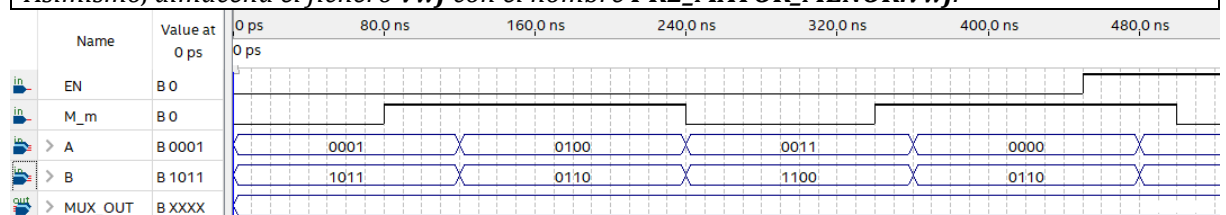


Figura 7.- Diagrama de formas de onda para la simulación del modelo **PR2_MAYOR_MENOR.bdf**.

5.3.3. VISUALIZACIÓN DEL MAYOR O MENOR ENTRE DOS NÚMEROS.

En este apartado se va a incluir, en el modelo de la Figura 5, un decodificador BCD a 7 segmentos para *displays* de ánodo común. El objetivo principal es comprobar la funcionalidad del mencionado decodificador a través de simulaciones. El diseño que se deber realizar se muestra en la Figura 8, siendo una ampliación del diseño especificado en el apartado anterior (Figura 5).

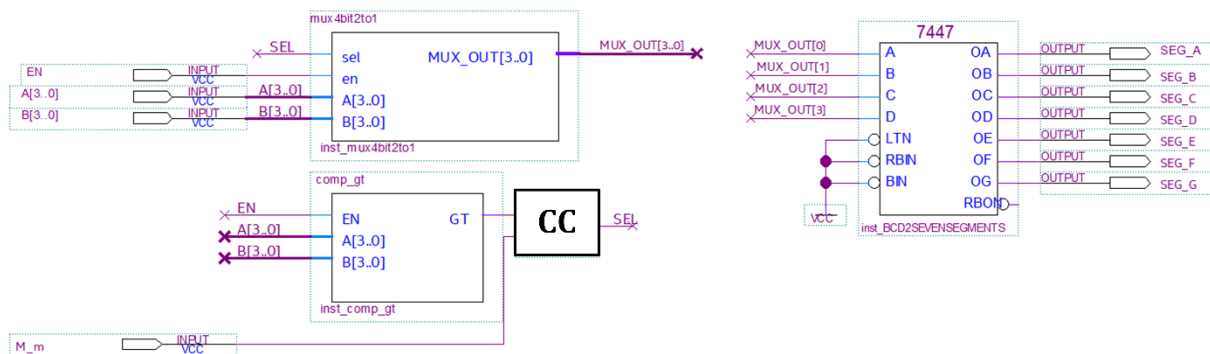


Figura 8.- Esquema para representar el mayor o menor entre dos números.

El componente 7447 tiene la misma funcionalidad que el circuito integrado del mismo nombre que se utilizó en la práctica 1.

Cuestión 7. Crea el esquema de la Figura 8 partiendo del esquema de la Figura 5, añadiendo el decodificador BCD a 7 segmentos (CI 7447). Realiza la simulación funcional y completa, en el cronograma de la Figura 9, las salidas desde **SEG_A** hasta **SEG_G**. El nombre del fichero que almacena el esquema debe ser **PR2_DISPLAY_MAYOR_MENOR.bdf**. Asimismo, almacena el fichero **vwf** con el nombre **PR2_DISPLAY_MAYOR_MENOR.vwf**.

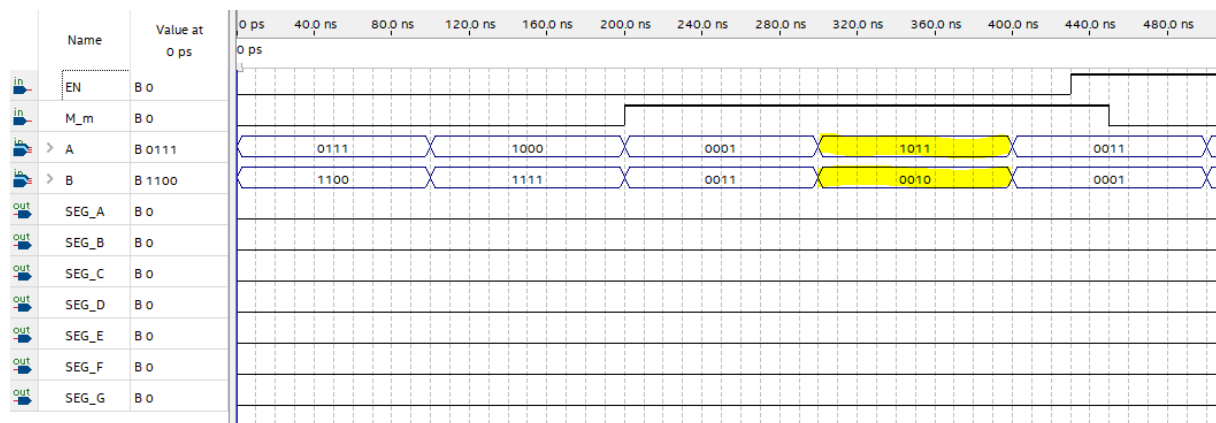


Figura 9.- Diagrama de tiempos para comprobar la funcionalidad del decodificador BCD a 7-seg.

Cuestión 8. ¿Qué ocurre en las salidas de circuito cuando en las entradas A y B se introducen los valores resaltados en la figura anterior? Razona la respuesta.

5.3.4. DISEÑO COMPLETO. ESQUEMA FINAL.

Para finalizar el modelado y simulación de la práctica 2 se deberán incluir los circuitos necesarios para poder representar en el *display* números que no sean BCD dentro de los posibles valores que pueden tomar las entradas **A** y **B**. En concreto deberá incluirse un sumador y un comparador. El sumador se implementará mediante el modelo basado en el circuito integrado 7483. Por el contrario, el comparador será idéntico al utilizado en apartados anteriores. Finalmente, las unidades se deben representar en los segmentos del *display* y las decenas en el punto decimal del mismo, estando este último encendido cuando las decenas tomen el valor 1.

La Figura 10 muestra el esquema final del diseño a completar. Como se puede ver en la figura, no hay un puerto de entrada para controlar señales de habilitación (**EN**). Por tanto, todas las entradas de habilitación de los bloques que la contenga, deberán conectarse a masa para que los circuitos correspondientes estén habilitados (ver Tabla 1 y Tabla 2).

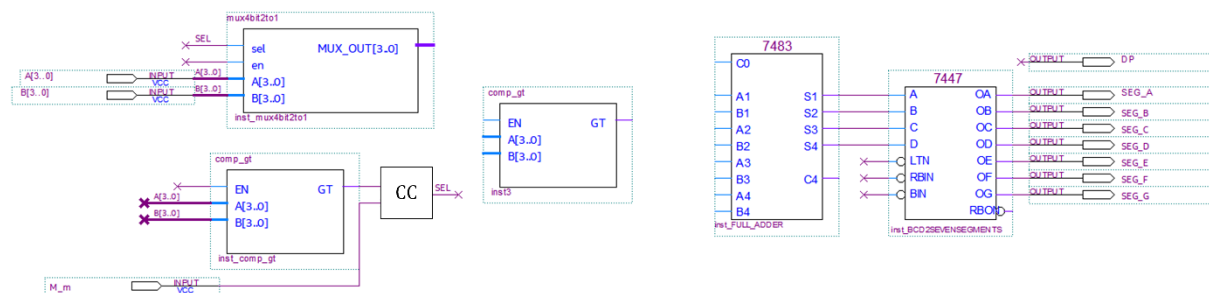


Figura 10.- Esquema final del diseño.

Cuestión 9. Completa el esquema de la Figura 10 para que implemente la funcionalidad del diseño completo. Completa, asimismo, el cronograma de la Figura 11 correspondiente a la simulación funcional del diseño completo. El nombre del fichero que almacena el esquema debe ser **PR2_GRPX_PY.bdf**. Asimismo, almacena el fichero **vwf** con el nombre **PR2_GRPX_PY.vwf**. Recuerda que, en los nombres anteriores, **X** representa el nombre del grupo de laboratorio e **Y** el identificador del puesto dentro del laboratorio.

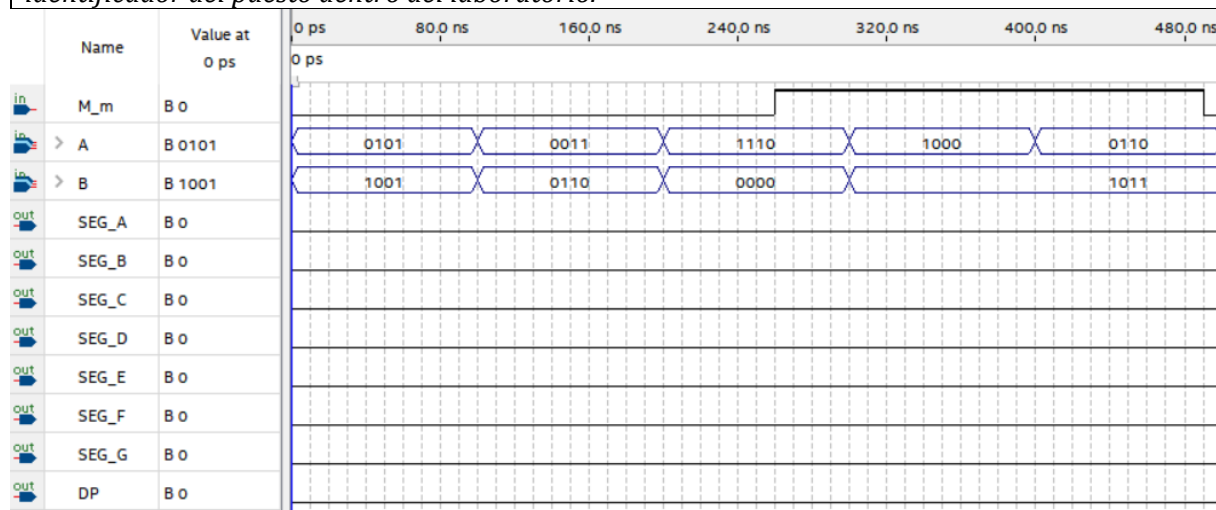


Figura 11.- Diagrama de formas de ondas del diseño final.

Cuestión 10. Sube al aula virtual el fichero **PR2_GRPX_PY.bdf** que contiene el diseño completo y el fichero **PR2_GRPX_PY.vwf** que hayas utilizado para realizar la simulación de la Figura 11. También se deberá enviar el resto de los ficheros **bdf** y **vwf** generados en las diferentes cuestiones. Para ello, se abrirá una actividad en el aula virtual.

5.4 PROGRAMACIÓN DEL CPLD Y PRUEBAS SOBRE LA PLACA

Una vez se ha comprobado, a través de las correspondientes simulaciones, que el diseño funciona correctamente, la última etapa del flujo de diseño consiste en la verificación del funcionamiento del mismo en placa (ver Figura 3).

La placa utilizada en el laboratorio tiene 4 conectores en los que vamos a conectar las entradas del circuito a través de interruptores y las salidas que excitarán a través de resistencias, los segmentos y el punto decimal de un *display*. Tanto los interruptores como el *display* estarán ubicados en la placa de pruebas del laboratorio, placa que se ha utilizado en la práctica 1.

Para poder realizar la descarga, hay que indicar sobre que pines del CPLD se conectan las entradas y salidas anteriormente comentadas. Para ello se utiliza la herramienta *Pin Planner* (ver Tutorial). Esta herramienta nos permite observar la localización física de los pines en el CPLD.

Dentro de la placa de pruebas hay cuatro conectores. El conector P1 se utilizará para las entradas y el conector P2 para las salidas del circuito diseñado. En la Figura 12 se puede ver la ubicación de los conectores de la placa junto con los pines del CPLD que se conectan a cada uno de ellos. Finalmente, los circuitos conectados en la placa de pruebas (interruptores y *display* entre otros) se alimentarán con una tensión de 3,3 V que se debe obtener de la placa de desarrollo **EPM240Board** a través conector P1.

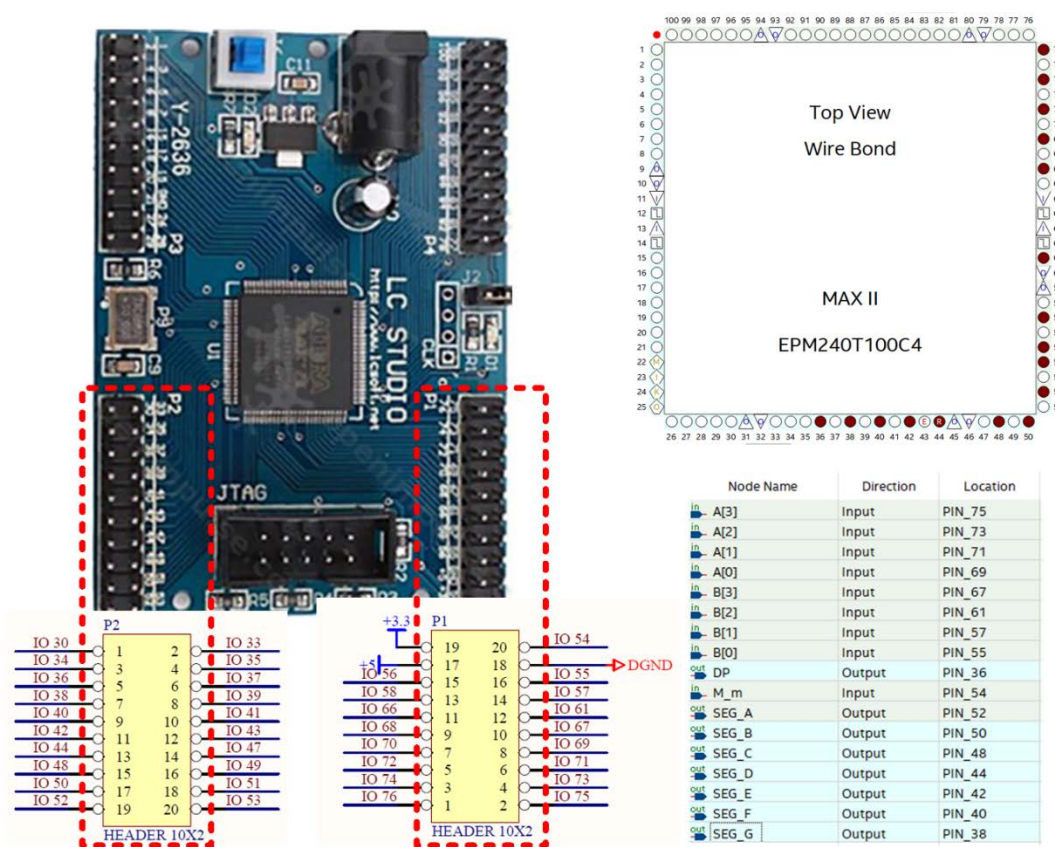


Figura 12.- Ubicación de pines dentro de la placa de pruebas.

Cuestión 11. Descarga el fichero en la placa y comprueba que funciona correctamente. Enseña al profesor correspondiente el resultado de la descarga.