

Escuela Técnica Superior de Ingeniería Industrial y de Telecomunicación.

Grado en Ingeniería de Tecnologías de Telecomunicación. Curso 2021/22.

Electrónica Digital I.

Se valorará la corrección del trabajo así como la presentación del mismo. La presentación puede hacerse a mano o por ordenador. Deben entregarse también (via e-mail, por ejemplo) los ficheros de Circuito Maker o Quartus II (todo los proyectos, no sólo los ficheros VHDL) generados al realizar la práctica (por ejemplo, comprimidos en un .zip o en un .rar).

Incluir una referencia al tiempo utilizado para hacer todo el trabajo (resolución y presentación), y el trabajo realizado por cada miembro del grupo indicando el tanto por ciento del trabajo total y las tareas realizadas. Última fecha de entrega: 29-Oct-2021.

Trabajo nº 3:

1º. Dada las funciones lógicas 1º a) y 1º b) que se indican:

(VER HOJA POR GRUPO)

- Diseñar con Circuit Maker en ficheros diferentes para cada función el circuito digital que se corresponda directamente con las funciones lógicas y obtener mediante simulación su tabla de verdad.

Atención: En Circuit Maker comprobar, haciendo doble-click sobre las puertas lógicas, que su modelo no es de tipo OC (“open-collector”, Circuit Maker lo indica como O/C en la descripción del modelo). Estas puertas no operan como puertas convencionales ya que cuando deben producir 1 lógico en realidad dejan el circuito desconectado (al usar el modo traza con el icono  en la simulación no aparecen las líneas en color rojo del 1 lógico sino en verde). En algunas instalaciones de Circuit Maker al seleccionar puertas lógicas desde la clase *Digital Basics*, las carga directamente con un modelo OC. Si seleccionan las puertas lógicas desde otras clases como *Original Symbols*->*Gates*, *Digital by Function* o *Digital by Number*, deberían funcionar bien.

- Simplificar la expresión lógica en lo posible mediante los teoremas y postulados del álgebra de conmutación.

- Diseñar con Circuit Maker un circuito digital que se corresponda con la expresión lógica reducida y comprobar que su tabla de verdad coincide con la del circuito generado de la función lógica sin simplificar. Se sugiere diseñar el circuito simplificado sobre el mismo fichero que el circuito original para comprobar que producen la misma tabla de verdad. En el caso de que las tablas de verdad sean distintas buscar la causa del error (errores en la simplificación, errores en el diseño del circuito, etc) y solventarla.

2º. Un sistema tiene tres entradas de datos X2, X1 y X0 y dos señales de control C1, C0 binarias (valores 0 o 1) y una única salida Z. En función de las entradas de datos se realizan los siguientes cálculos aritméticos (VER HOJA POR GRUPO 2º a). En base a los resultados aritméticos se quiere saber si se cumple una de la siguientes cuatro condiciones (VER HOJA POR GRUPO 2º b), seleccionando la condición mediante los valores de las entradas de control C1 y C0, lo que activará (si se cumple) o no (si no se cumple) la salida Z del sistema.

a) Realizar una descripción VHDL del problema lógico, usando una arquitectura generada directamente de la especificación (usando sentencias del tipo IF-ELSE, CASE, LOOP, operadores aritméticos, incluyendo ABS: valor absoluto, y MOD o REM: resto de una división, operaciones relacionales, etc, pero sin generar directamente la tabla de verdad del problema). Para este problema la descripción puede hacerse dentro de un único *process*, en varias partes formadas por sentencias secuenciales. Por ejemplo, la primera parte debe generar los resultados de las operaciones usando variables intermedias (F0, F1, ...) de tipo entero para obtener los valores de las operaciones aritméticas en función del valor lógico de las entradas Xs: usar sentencias para asignar las variables Fs primero el valor del término independiente, y luego para cada entrada una sentencia IF que añade en cada F el valor del coeficiente correspondiente si la entrada es 1. Puede que sea cómodo definir las variables intermedias F como un *array* de enteros, para poder recorrerlos mediante lazos *loop*. Para definir un *array* unidimensional de 4 enteros hay que declarar un nuevo tipo, por ejemplo, en la zona de definiciones de la arquitectura (antes de su *begin*), mediante la sentencia VHDL:

```
type nom_type is array (3 downto 0) of integer range -10 to 10;
```

donde *nom_type* es el nombre del tipo que elija cada uno. El rango de valores asociado a los enteros es opcional. Hay que tener cuidado de que los valores máximo o mínimo de las operaciones están dentro del rango indicado. Las señales o variables de ese tipo se pueden definir, por ejemplo

```
variable F: nom_type;
```

que corresponden a 4 variables enteras F(0), F(1), F(2), F(3).

La segunda parte debe obtener el valor de las condiciones pedidas en función de los valores de Fs ya calculados, y se pueden usar sentencias de alto nivel (*for-loop*, *if-else*, etc) y operadores relacionales (=, /=, >, <, etc), para calcular otras variables intermedias que indiquen si se cumplen las condiciones indicadas en el enunciado (2º-b).

Por último, la tercera parte genera el valor lógico de la salida en la salida Z en base a los valores de esas últimas variables y de las señales de control, por ejemplo, con una sentencia *case*.

- Utilizando la herramienta de diseño Quartus II, versión 9.1, generar un proyecto para ese fichero utilizando el dispositivo EP2C35F672C6 de la familia Cyclone II.

- Compilar la descripción depurando los errores que aparezcan. Obtener el número de celdas utilizadas en el diseño y obtener el tiempo máximo de propagación del circuito dado por la herramienta de análisis temporal.

- Simular el circuito comprobando que el funcionamiento es el correcto. Utilizar unos estímulos que comprueben las 32 combinaciones válidas de las entradas. Se pueden cargar los valores fácilmente en X0 como señal periódica con X_0 , y doblar sucesivamente el periodo para X0, X1, X2, C0 y C1 (o usar X_C si X es un vector, y programarlo como una cuenta de 00000 a 11111). Los cambios en las entradas deben hacerse con un tiempo mayor que el tiempo de propagación máximo del circuito (aproximadamente 5 veces, por ejemplo) para poder medir correctamente las salidas.

** No todos los ejercicios han sido comprobadas exhaustivamente: si alguno diese un resultado anómalo (salidas siempre 0 ó siempre 1, siempre el mismo valor, etc) hablar con el profesor.

- Obtener la potencia disipada dada por la herramienta *PowerPlay*. Visualizar y describir la estructura general del circuito.

- Programar las opciones de compilación para intentar reducir el tiempo de propagación en, al menos, un 20%. Volver a compilar, comprobar si se cumple el objetivo fijado y volver a simular el circuito bajo estas condiciones. Medir de nuevo el tiempo máximo de propagación y la potencia disipada.

b) Realizar una descripción VHDL del problema lógico, usando una arquitectura generada a partir de la tabla de verdad del problema. Obtener la tabla de verdad de este problema de 5 entradas y una salida, y utilizar una descripción basada en una sentencia de tipo *case* dentro de un *process*.

- Utilizando la herramienta de diseño Quartus II, versión 9.1, generar un proyecto para ese fichero utilizando el dispositivo EP2C35F672C6 de la familia Cyclone II.

- Compilar la descripción depurando los errores que aparezcan. Obtener el número de celdas utilizadas en el diseño y obtener el tiempo máximo de propagación del circuito dado por la herramienta de análisis temporal.

- Simular el circuito comprobando que el funcionamiento es el correcto. Se puede copiar el fichero de estímulos del apartado anterior.

- Obtener la potencia disipada dada por la herramienta *PowerPlay*. Visualizar y describir la estructura general del circuito.

- Programar las opciones de compilación para intentar reducir el tiempo de propagación en, al menos, un 20%. Volver a compilar, comprobar si se cumple el objetivo fijado y volver a simular el circuito bajo estas condiciones. Medir de nuevo el tiempo máximo de propagación y la potencia disipada.

c) Comparar los resultados obtenidos entre las dos descripciones, comprobando que los resultados de la simulación en las salidas son iguales en las dos descripciones. Indicar las ventajas e inconvenientes de utilizar una u otra descripción.