SISTEMAS ELECTRÓNICOS. PRÁCTICA Nº 3.

Grado en Ingeniería en Electrónica, Robótica y Mecatrónica.

Objetivo:

Utilizando el software de Xilinx ISE 14.7 y Aldec Active-HDL, se proporciona un circuito implementado en una FPGA Spartan-6 que realiza la multiplicación de dos números complejos: $(a + jb) \cdot (c + jd)$. La introducción de datos se realiza desde el teclado hexadecimal, y la visualización de los datos introducidos se proporciona en los 4 displays de 7 segmentos de la placa Nexys3 (uno para cada dato), así como el resultado final, usando en este caso 2 dígitos para la parte real y otros 2 para la parte imaginaria del número calculado.

Inicialmente, el sistema esperará la introducción por parte del usuario del primer dato "a", un número entero y positivo de 0 a 9; a continuación, esperará el resto de cifras, "b", "c" y "d". Conforme se vayan tecleando dichos datos se irán mostrando por los displays de 7 segmentos de la Nexys3, y se almacenarán los mismos en 4 registros distintos. Posteriormente, se procederá a realizar la multiplicación compleja de los números, y se almacenará el resultado en 2 registros distintos, uno para la parte real y otro para la parte imaginaria del resultado, con 8 bits cada uno para poder albergar el resultado de las operaciones matemáticas efectuadas. Una vez realizado el cálculo de la multiplicación compleja, se convertirán de binario a BCD esos datos y se mostrarán por los displays de 7 segmentos de la Nexys3, utilizando el conversor de binario a BCD proporcionado, con 2 cifras para cada parte del número complejo final (2 cifras BCD para la parte real y otras 2 para la parte imaginaria).

Por tanto, **el estudiante deberá realizar las siguientes tareas** en el diseño Xilinx proporcionado para la práctica:

A) **Entender el funcionamiento del circuito proporcionado (TAREA A)**:

- Comprender el módulo de introducción de datos proporcionado, que es similar al utilizado en la práctica 2, de manera que ahora acepta solamente los números del 0 al 9 a la entrada.
- Al introducir los 4 números de entrada, el sistema automáticamente comenzará el cálculo de la multiplicación compleja, y el resultado lo mostrará por los displays de 7 segmentos de la placa Nexys3.
- El diseño del circuito de cálculo está realizado aplicando el criterio de minimizar el hardware utilizado para el cálculo de la multiplicación compleja, utilizando por tanto el mínimo número de multiplicadores y sumadores/restadores posibles, y optimizando todo lo posible el número de ciclos de reloj empleados.
- Para realizar el cálculo de la multiplicación compleja se han implementado diversos bloques (multiplicador, sumador/restador) a través de la herramienta CORE Generator. Además, para controlar este proceso de cálculo, se ha construido un circuito de control (máquina de estados, FSM) utilizando la herramienta Aldec Active-HDL, llevando ese código VHDL al Xilinx y situándolo en el diseño.
- A la salida, se multiplexan los datos hacia los displays de 7 segmentos; en el momento de introducir datos se verá en cada display el dato tecleado por el usuario, y tras el cálculo de la multiplicación compleja se verán 2 cifras BCD de la parte real del resultado y otras 2 de la parte imaginaria, separadas por el punto decimal intermedio.

B) **TAREA B, HITO Nº 1, 50% DE LA CALIFICACIÓN PRÁCTICA:**
Realizar las siguientes tareas sobre dicho circuito para **AÑADIR NUEVAS FUNCIONALIDADES**:

- **Posibilidad de introducir 1 cifra decimal adicional a la cifra entera.** Se introducirán siempre números con 1 cifra entera y 1 cifra decimal, por lo que se encenderá el punto decimal adecuado en el display de 7 segmentos de la placa Nexys3 para que así lo vea el usuario. Por tanto, ahora se podrán introducir números desde el 0,0 hasta el 9,9, tecleando siempre el usuario las 2 cifras.

- **Confirmación del dato de entrada con la tecla A.** Ahora, al tener más de una cifra cada dato, se pedirá confirmación de los mismos utilizando la tecla A. Por tanto, se teclean el número entero, la cifra decimal, y se confirmará su introducción al circuito con la tecla A.
- **El resultado podrá ser de 2 cifras enteras a la salida con 2 decimales, y positivo o negativo en cada una de las partes de salida (real e imaginaria)**. Por tanto, los datos real e imaginario de la multiplicación compleja tendrán hasta **2 cifras enteras BCD y 2 decimales, y se multiplexarán a la salida** utilizando el switch **SW(0)** de la Nexys3 (0 → Parte Real, 1 → Parte Imaginaria). **Para indicar el signo para cada una de las partes se encenderá algún led de la placa Nexys3 para indicarlo, y también se indicará el punto decimal en el lugar adecuado**.

C) **TAREA C, HITO Nº 2, 50% DE LA CALIFICACIÓN PRÁCTICA:**

Una vez que esté operativo el diseño, introducir un MicroBlaze en el mismo que realice las tareas de entrada y salida de datos, utilizando el teclado y pantalla del PC. De esta manera, el teclado del ordenador hará las veces de teclado hexadecimal proporcionando las pulsaciones de las teclas por parte del usuario, y los resultados los mostrará por la pantalla del PC el MicroBlaze, en este caso sin limitaciones dado que podemos usar todo el espacio requerido para mostrar los datos de salida simultáneamente. Por tanto, las tareas a realizar serán:

- **Sustituir la entrada de datos a través del teclado hexadecimal y la salida de la información a través de los displays 7 segmentos por un MicroBlaze, de manera que se utilizará el teclado del PC como entrada de datos y la pantalla del PC como salida de información**, funcionando éste en modo de terminal con el programa "Putty". La comunicación entre el micro y la parte de la FPGA se realizará mediante los distintos GPI y GPO (puertos de entrada y salida genéricos del MicroBlaze) para enviar y recibir los datos, así como para alguna señal de sincronismo necesaria para el envío de datos desde el MicroBlaze hacia la FPGA, y las señales de interrupciones "intc_interrupt(0:0)" e "intc_irq" para avisar al MicroBlaze desde la FPGA que tiene los datos de la multiplicación compleja calculados y debe leerlos y actualizarlos por pantalla. **Será necesario**, por tanto, **modificar el código proporcionado del MicroBlaze** para incluir la funcionalidad de entrada y salida de datos, **así como realizar cambios en el esquemático de nivel superior** añadiendo un circuito en la parte de la FPGA para comunicar adecuadamente el micro con el resto del sistema, y reconectando todo de manera correcta para que el micro sea el encargado de introducir las pulsaciones de las teclas así como de recibir los datos de parte real y parte imaginaria del resultado de salida que hay que mostrar por la pantalla del PC. Se recomienda dejar operativos los bloques para la salida de la información los displays de 7 segmentos para así tener una referencia de lo que se debería ver por la pantalla del PC al introducir el MicroBlaze como dispositivo de Entrada/Salida.
- **Anadir al bloque del CORE Generator del MicroBlaze una señal de interrupción externa "intc_interrupt(0:0)", configurando ésta como activa por flanco de subida**.
- **Añadir al bloque del CORE Generator del MicroBlaze un temporizador de tipo FIT (temporizador fijo) de 10 milisegundos de duración** para que, a partir del mismo, se conmute cada 0,5 segundos el estado de un led de la placa Nexys3 y así comprobar que el MicroBlaze está funcionando adecuadamente.

Restricciones para el diseño:

- Los datos de entrada serán de 8 bits cada uno (2 cifras en BCD, parte entera y parte decimal), por tanto los datos que albergarán serán de 0,0 a 9,9 como máximo: A(7:0), B(7:0), C(7:0) y D(7:0). Cada vez que se introduzca un dato nuevo se generará una señal "carga_cifra" que le indicará al bloque de almacenamiento (registro de desplazamiento, ya implementado en el diseño) que debe guardar este dato proporcionado.
- Estos datos en BCD con una cifra entera y otra decimal se convertirán a binario utilizando el conversor de BCD a binario proporcionado en el diseño de Xilinx ISE de la práctica, llamado "*Conversor_BCD_Bin_1decim*". Este conversor incluye la parte decimal, que la convierte a binario de 4 bits de precisión, por lo que el dato de salida de 8 bits tendrá 4 para la parte entera, y otros 4 para la parte decimal, en binario puro (sin signo).
- A la salida del circuito de cálculo se ofrecerá el resultado con 16 bits cada uno, Dato_Real(15:0) y Dato_Imag(15:0), junto a la señal de sincronismo "Multip_OK" que indica que el resultado está listo para ser mostrado hacia el exterior. Se debe tener en cuenta que cada dato de salida incluirá los bits necesarios para la parte decimal y el signo, ya que serán datos en complemento a 2 a la salida (pueden salir resultados positivos o negativos).
- Se convertirán estos resultados de la multiplicación compleja de complemento a 2 a binario puro para que el siguiente conversor de binario a BCD llamado "*Conversor_Bin_BCD_decim*" pueda procesarlos, ya que sólo acepta números positivos.
- Se convertirán los datos en binario puro de 8 bits para la parte entera y 8 bits para la parte decimal a BCD, quedando finalmente 2 cifras BCD enteras y 2 cifras decimales para cada uno de los resultados. Por tanto, se podrán representar a la salida datos desde el -99,99 hasta el 99,99.
- Hay que tener en cuenta que alguno de los datos de salida podría ser negativo, por lo que a la hora de mostrarlo habría que encender un led de la placa para indicarlo al usuario.
- Como ya se ha indicado (**importante**), el diseño del circuito de cálculo está realizado aplicando el criterio de minimizar el hardware utilizado para la multiplicación compleja, utilizando el mínimo número de multiplicadores y sumadores/restadores que sea posible, y optimizando a la vez el número de ciclos de reloj empleados. El resto del circuito se debe construir de la misma manera.

Nota: La multiplicación compleja consiste en calcular $(a + jb) \cdot (c + jd)$, por tanto, la parte real e imaginaria del resultado consistirá en:

Parte Real: $(a \cdot c - b \cdot d)$
Parte Imaginaria: $(a \cdot d + b \cdot c)$ j

En el circuito original, como los resultados se muestran sólo con 2 dígitos para la parte real y otros 2 dígitos para la parte imaginaria, no se podrán usar valores muy grandes para los datos "a", "b", "c" y "d" para que los resultados no se desborden.

Ejemplos:
$(5 + 3j) \cdot (2 + 1j) = 7 + 11j$
$(7 + 4j) \cdot (5 + 2j) = 27 + 34j$

Fases de trabajo (TAREA B, HITO Nº 1, 50% DE CALIFICACIÓN PRÁCTICA):

1ª.- **En primer lugar, se modificará el diagrama de bloques funcionales proporcionado para que incluya los nuevos módulos del diseño.** Este diagrama se entregará en una tarea de forma previa a los resultados finales.

2ª.- **Se desarrollarán las partes del circuito a implementar utilizando el editor de esquemáticos** incluido en la herramienta ISE, junto con los bloques ya proporcionados en el mismo diseño, los elementos de la librería que sean necesarios (biestables, puertas lógicas, comparadores, multiplexores, etc), y bloques creados con la herramienta **"CORE Generator"**. Asimismo, se utilizará el editor de máquinas de estados (FSM) **Active-HDL** para la elaboración y/o modificación de los circuitos de control (máquinas de estados, FSM) que sean necesarios.

3ª.- **Se simulará el diseño** de manera funcional hasta que trabaje correctamente. Para la simulación será imprescindible la modificación del "Test Bench" en VHDL proporcionado en la práctica ("sim_bloque_calculo.vhd") **de tal manera que se refleje el funcionamiento completo del circuito**. En la hoja de resultados final se adjuntarán diversas capturas de las señales más importantes del circuito que muestren que se está realizando correctamente el cálculo de la multiplicación compleja, junto con los comentarios que el estudiante estime oportunos.

4ª.- **Se procederá a la implementación del diseño**, de manera que se asegure que funciona a la máxima velocidad posible, **y se probará sobre la placa Nexys3 junto con la placa auxiliar del teclado hexadecimal**). Se podrá utilizar el Remote Lab para probar el funcionamiento del diseño igualmente.

Fases de trabajo (TAREA C, HITO Nº 2, 50% DE LA CALIFICACIÓN PRÁCTICA):

1ª.- **Se realizará un diagrama de bloques funcionales solamente del esquemático "top" del diseño (el nivel superior)**, en el cual se establezca el camino de los datos ("*datapath*") desde que vienen del MicroBlaze, se procesan en el módulo "Bloque_calculo" y se devuelven al MicroBlaze para mostrarlos por pantalla. Este diagrama se elaborará en primer lugar y se adjuntará a la entrega final.

2ª.- **Se creará un MicroBlaze mediante el CORE Generator, incluyendo el temporizador FIT1 de 10 milisegundos (activando sus interrupciones), y además se proveerá al micro de una interrupción externa por flanco de subida** (ver ejemplos en el Campus Virtual para observar cómo está configurado el MicroBlaze).

3ª.- **Se realizará el código C del MicroBlaze utilizando el entorno de desarrollo SDK de Xilinx y, además, se utilizará el editor de esquemáticos** incluido en la herramienta ISE, los bloques ya proporcionados en el mismo diseño, los elementos de la librería y los bloques del "**CORE Generator**" que sean necesarios (biestables, puertas lógicas, comparadores, etc). Asimismo, nos serviremos del editor de máquinas de estados (FSM) **Active-HD**L para la elaboración de los circuitos de control (máquinas de estados, FSM) que sean necesarios.

4ª.- **Se simularán los bloques de la FPGA que sean necesarios introducir para completar el diseño hasta que trabajen correctamente. También se creará un script simple para simular el circuito completo**, y dado que el **MicroBlaze** realiza las funciones de entrada/salida de datos, **haremos que éste funcione de manera que al principio de su operación realice algún intercambio de datos con la parte de la FPGA para que se vea en dicha simulación, y se pruebe el funcionamiento del temporizador también**. Por ejemplo, haremos que al principio el MicroBlaze simule que el usuario ha introducido algunos datos de números complejos y deberemos ver cómo se calcula la multiplicación compleja y la salida se envía de manera correcta al MicroBlaze de nuevo, todo sin intervención del usuario. Para la simulación será imprescindible la creación de un "Test Bench" en VHDL para la simulación del circuito completo.

5ª.- **Se procederá a la implementación del diseño modificado**, de manera que se asegure que funciona a la máxima velocidad posible, y se probará sobre la placa Nexys3 conectándola al PC mediante un cable USB de manera que el teclado y pantalla del PC hagan las funciones de entrada y salida de datos, respectivamente.

Trabajo a presentar por el estudiante (OBLIGATORIO):

1.- Como se indicó en los objetivos, será necesario modificar la práctica para incluir toda la funcionalidad requerida, es decir, cálculo de la multiplicación compleja con datos de 1 cifra y 1 decimal positivos, así como la multiplexación de los resultados hacia los displays de 7 segmentos de la Nexys3. Se realizará y se mostrará en clase al profesor, en primer lugar, los diagramas de bloques, y se entregarán los mismos en la tarea indicada. A continuación, se modificará el diseño de Xilinx ISE para incluir los nuevos bloques, y se mostrará en clase al profesor el diseño realizado. Se podrá utilizar el programa Active-HDL para realizar nuevas máquinas de estado si son necesarias, o para modificar las existentes.

2.- Modificar el script de simulación para incluir las nuevas situaciones a simular, es decir, introducción de diversos datos para el cálculo de la multiplicación compleja, incluyendo los casos más "extremos" (números muy pequeños y números muy grandes), observando correctamente los valores de salida. Se mostrará en clase al profesor la simulación del circuito realizada.

3.- Se implementará el circuito correctamente sin errores, y se probará en la placa Nexys3, mostrando igualmente al profesor el resultado final. Este diseño funcionando de manera correcta será el que se deba presentar como resultado de la práctica, junto con la hoja de resultados que se incluye aparte.

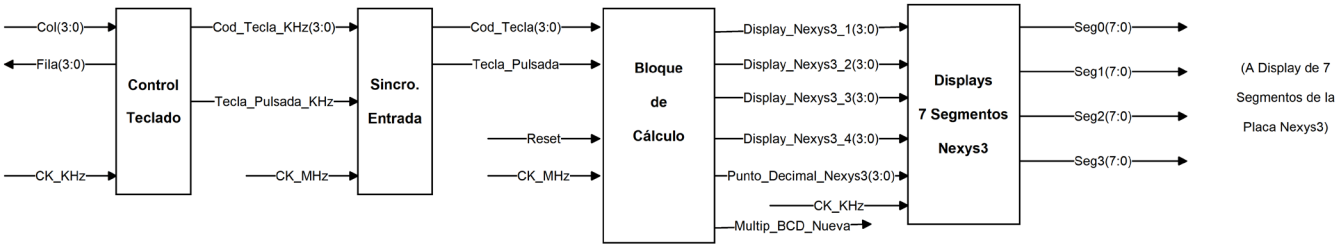
4.- Se introducirá en el circuito el MicroBlaze con la funcionalidad indicada de hacer las veces de entrada/salida de datos. Se simulará el circuito y se implementará en la placa Nexys3. Se mostrará en clase al profesor todo este trabajo realizado y se adjuntará este nuevo diseño a la entrega final también.

5.- Se subirá a la tarea de la "Práctica 3 – Entrega de Diseño y Hoja de Resultados" del Campus Virtual un archivo comprimido que contenga la carpeta con la práctica completa de la TAREA B (HITO Nº 1). Si se ha implementado la TAREA C (HITO Nº 2), constará de 2 diseños, el primero utilizando teclado hexadecimal y displays 7 segmentos como entrada/salida de datos, y el segundo con el MicroBlaze como elemento de entrada/salida (será el mismo diseño por cada pareja que trabaje junta en el laboratorio), incluyendo diseño de Xilinx ISE y el de Active-HDL para las máquinas de estado. Aparte, en un segundo archivo separado, la "Hoja de Resultados" que se incluye en el archivo comprimido de la práctica, una vez rellena (en formato PDF o Word) y completada de manera INDIVIDUAL.

Diagrama de Bloques General:

El siguiente diagrama muestra el recorrido de los datos de entrada (introducidos desde el teclado hexadecimal, a través del bloque "Control Teclado") desde que entran al circuito, se sincronizan adecuadamente, se introducen en el bloque de cálculo, y salen del mismo los datos en BCD a mostrar por los 4 displays de 7 segmentos de la placa Nexys3:

Diagrama de Bloques General (Top)
Práctica 3



Mirando dentro del interior del "Bloque de Cálculo", vemos que dentro del mismo se detecta si la pulsación de la tecla corresponde a un número o a una letra (para realizar la acción correspondiente), se almacenan en un registro de desplazamiento (sólo las cifras numéricas) para su envío al bloque de cálculo del multiplicador complejo iniciándose el cálculo al recibir los 4 datos, y al acabar se ofrece los datos a su salida ("Dato_Real(7:0)" y "Dato_Imag(7:0)") así como una señal de sincronismo "Multip_OK" que indica que se ha acabado el proceso, por lo que el dato en binario se puede convertir a BCD, multiplexar después los datos BCD según proceda, y darle salida a través de los displays de 7 segmentos. También se incluye una pequeña máquina de estados "Control entZ_sal" encargada de gestionar si se está en momento de introducción de datos (entZ_sal = '0') o en momento de salida de datos (entZ_sal = '1'):

Diagrama de Bloques "Bloque de Cálculo"
Práctica 3