

CONFIGURACION DEL PROGRAMA QUARTUS II PARA EL FPGA CYCLONE IV

Gerardo Duran Medrano

20 de Marzo de 2015

1 Introducción

En la actualidad, cada día y a cada instante, se generan datos que se requieren interpretar y adquirir para diversos usos. La adquisición de estos datos se hace cada vez más renuente, ya que existe una gran diversidad de dispositivos que son capaces de realizar estas tareas.

Los dispositivos utilizados deben tomar los datos en el tiempo más real posible, ya que en algunos casos la adquisición en tiempo real hace más efectivo el trabajo y por ende más realista.

Los microcontroladores son dispositivos confiables para adquirir datos, ya que pueden tomar muestras muy acercadas a lo real, la desventaja es que su adquisición de tiempo no lo hace en tiempo real, más al momento de ejecutar una acción inmediata después de obtener el resultado esperado, por ejemplo un microcontrolador de Microchip dura cuatro ciclos de reloj para cada acción ejecutada.

El dispositivo más confiable para lograr esto son los FPGA (Field Programmable Gate Array, por sus siglas en inglés), estos son campos de arreglos de compuertas logicas que son programables y lo mejor de estos dispositivos es que cada ejecución lo hace en tiempo real en tan solo un ciclo de reloj momento de adquirir el dato, además de que cada vez que adquiere el dato lo hace en tiempo muy real, obviamente dependiendo del FPGA.

El presente documento es una práctica que muestra como se configura el FPGA Cyclone IV en el programa Quartus II, con el fin de realizar más adelante prácticas de adquisición de datos para analizar diversos comportamientos y realizar diferentes tareas que se programarán en el dispositivo, además con la ayuda de elementos electrónicos desarrollar más tareas avanzadas.

2 Desarrollo

PRIMER PASO. El software a utilizar Quartus II, es un programa que se puede descargar de manera gratuita de la página de Altera. Ya instalado el programa se procede con lo siguiente: En el Menú superior en el apartado FILE, buscamos la opción que dice NEW PROJECT WIZARD, ahí se abrirá una ventana. La primer ventana dice INTRODUCTION, ya que solo es informativa, se puede seleccionar la opción en la parte inferior DON'T SHOW ME THIS INTRODUCTION AGAIN, de igual manera se presiona la opción NEXT para continuar con la configuración del proyecto nuevo. En la siguiente ventana muestra tres cuadros para introducir texto, en el primero se selecciona la ubicación donde se guardará el proyecto, es recomendable tener una carpeta asignada para cada proyecto, ya que se generarán algunos archivos y todos son importantes para la ejecución del proyecto; en el siguiente cuadro se asignará un nombre al proyecto y en el tercer cuadro debe estar el mismo nombre que se asignará al proyecto, esto lo hace de manera automática, después de realizar estas acciones se da clic en NEXT. A la siguiente ventana solo damos clic en NEXT. En la tercer ventana de configuración, se seleccionará la familia y configuración del dispositivo, en el apartado de FAMILY, seleccionamos la opción CYCLONE IV E, y en el NAME FILTER, seleccionamos la opción del FPGA a utilizar que es el modelo EP4CE22F17C6, por último en el apartado de AVAILABLE DEVICES, se selecciona la única opción, y por último se da clic en NEXT. En la ventana cuatro solamente se da clic en NEXT, y en la última ventana se selecciona la opción FINISH.

SEGUNDO PASO. Es poner los pines que no se van a utilizar en alta impedancia. Para esto se selecciona el menú ASSIGNMENTS la opción DEVICE, abrirá una ventana ahora se selecciona la opción DEVICE AND PIN OPTIONS y en el apartado CATEGORY, seleccionamos la opción UNUSED PINS, y por último en la opción de RESERVE ALL UNUSED PINS seleccionamos la opción AS INPUT TRI-STATED, y por último damos clic en OK.

TERCER PASO. Insertar los archivos generados anteriormente del código VHDL que se generó en MATLAB, con la ayuda de SIMULINK. Para lograr esto, en la ventana de PROJECT NAVIGATOR, se selecciona la pestaña de FILES, luego se da clic derecho dentro de la ventana en la opción de FILES, y se da clic en ADD/REMOVE FILES IN PROJECT, esto abrirá una nueva ventana. En la opción de BROWSE, que esta marcada con un cuadro con tres puntos, abrirá otra ventana, y buscamos los

archivos generados por el SIMULINK, primero el ADC y después el DAC. Y por último damos clic en OK.

CUARTO PASO. El siguiente paso es crear los símbolos de los códigos generados, principalmente el de los dos sistemas, que son de los convertidores ADC y DAC que se vieron anteriormente y que serán de ayuda para el FPGA. En la ventana de PROJECT NAVIGATOR, se da clic derecho en cualquiera de los dos (no importa el orden), y se selecciona la opción CREATE SYMBOL FOR CURRENT FILE, al terminar de compilar esta opción deberá aparecer una ventana que dice CREATE SYMBOL FILE WAS SUCCESSFUL. Realizamos esta misma acción para el que quedara pendiente de los convertidores.

QUINTO PASO. Ahora vamos a insertar un diagrama a bloques, para poder trabajar con el diseño. Damos clic en el menú FILE, luego en NEW y se abrirá una ventana, y ahí seleccionaremos BLOCK DIAGRAM/SCHEMATIC FILE. Damos clic en OK y tendremos el área de trabajo.

SEXTO PASO. Insertaremos los símbolos creados. Damos clic derecho dentro del área de trabajo y en la opción INSERT, luego en la opción SYMBOL. Ahora se abrirá una ventana, en el apartado de LIBRARIES en la carpeta que dice PROJECT, seleccionamos el primero, ya sea el ADC o el DAC. Luego hacemos lo mismo para el que se quedó pendiente.

SEPTIMO PASO. Crear el reloj que utilizarán el DAC y el ADC. Se da clic derecho sobre el área de trabajo y se selecciona la opción INSERT, luego en SYMBOL, y en el cuadro de texto de NAME, se escribe lo siguiente ALTPLL, damos clic en OK. Se abre una ventana para configurar, dejamos la opción VHDL y se da clic en NEXT. Ahora en la ventana 3 de 14, en la opción de WHAT IS THE FRECUENCY OF THE INCLK0 INPUT?, en el cuadro de texto le ponemos a 50 MHZ, que es la frecuencia con la que trabaja nuestro FPGA, ahora damos clic en NEXT. En la ventana 4 de 14, quitamos lo que se encuentre seleccionado y damos clic en NEXT varias veces hasta llegar a la ventana 8 de 14, que será para configurar los relojes externos. Ya que se quedó seleccionada la pestaña de clk c0, se empieza a configurar el primer reloj, en la opción de ENTER OUTPUT CLOCK FRECUENCY, le ponemos 2.1 Mhz, que es la frecuencia de reloj con la que trabaja nuestro ADC, damos clic en NEXT. En esta ventana seleccionamos la opción de USE THIS CLOCK, para activar el siguiente reloj, e insertamos la frecuencia de 20Mhz, que es la frecuencia con la que trabaja el DAC, luego damos clic en NEXT hasta la ventana 14 para dar clic por último en FINISH. Y de ahí insertamos el bloque del generador de reloj.

OCTAVO PASO. Vamos a generar los pines de salida. Para esto damos clic derecho sobre el símbolo del ALTPLL y seleccionamos la opción GENERATE PINS FOR SYMBOL PARTS. Y le podemos cambiar el nombre a los pines para identificarlos más fácilmente. Y los relojes generados para cada dispositivo se debe conectar hacia los CLK de cada dispositivo, pero se deben conectar con inversores. Se da clic derecho sobre el área de trabajo y en el cuadro de texto NAME escribimos NOT, para que aparezca una compuerta not que nos ayudará a negar las entradas hacia el dispositivo. Ahora generamos los pines para el ADC y el DAC de la misma manera que se generaron para el altpll.

NOVENO PASO. Ahora vamos a conectar en arreglo el DAC y el ADC, y dejar los pines que se utilizaran a la salida. Los pines clk y clock van conectados juntos, en ambos dispositivos. A la entrada RESET se le niega la entrada, antes del inversor se conectan CLK ENABLE de cada dispositivo y después del inversor se conectan a ambos RESET. Por último se le cambian los nombres a CS, SCLK y DOUT por ADC y DAC, seguidos del nombre respectivamente. Y por último guardamos el avance.

DECIMO PASO. Ya que todos los pines ya están asignados previamente en el FPGA, debemos asignar cada uno de los pines. Primero debemos analizar y sintetizar para poder configurar los pines, para esto damos clic en el menú PROCESSING, luego START y luego START ANALYSIS N SYNTHESIS. Y lo comienza a compilar, hasta que sale una ventana que muestra que fue satisfactorio. Si muestra errores, no se puede continuar hasta corregirlos, y si muestra advertencias, no hay problema se puede continuar con la asignación de pines. Luego vamos al menú ASSIGNMENTS en la opción PIN PLANNER. Emergerá una ventana para configurar los pines, y muestra los pines que se asignaron en la columna NODE NAME y para asignarle sus pines de salida se modificará en LOCATION, asignando así cada uno de los pines más cercanos para los convertidores ADC y DAC:

- ADCCS quedará en el pin P16.
- ADDIN quedará en el pin L15.
- ADCDOUT quedará en el pin K16.
- ADCSCLK quedará en el pin N11.
- CLK quedará en el pin R8.
- DACCS quedará en el pin P9.

- DACSCLK quedará en el pin R10.
- DACDI quedará en el pin R11.
- RESET quedará en el pin J15.

Por último cerramos esa ventana.

ONCEAVO PASO. Damos clic en el icono de PROGRAMMER, y seleccionamos en HARDWARE SETUP la opción de USB-BLASTER (USB-0). Luego damos clic en START y en la barra de PROGRESS debe aparecer en SUCCESSFUL.

DOCEAVO PASO. Verificamos el comportamiento de los convertidores con la ayuda del QUARTUS. Una vez que el FPGA ya está programado, vamos a corroborar que la programación quedará de manera correcta con los datos introducidos desde simulink. En el menú TOOLS, seleccionamos la opción SIGNAL TAP II LOGIC ANALYZER. En esta ventana nos da automáticamente el hardware con el que estamos trabajando, así como el dispositivo y algunas características del mismo tales como la memoria ocupada. En la ventana AUTO-SIGNALTAP-0 hacemos doble clic en la parte en blanco. Ahí sale otra ventana para seleccionar los nodos de los cuales se requiere información de su comportamiento, ya sea los relojes de los convertidores o alguno de los pines que se quieran analizar. Ya que seleccionamos los nodos o pines a analizar, en la opción de INSTANCE MANAGER, damos clic en RUN ANALYSIS para ejecutar una vez y en AUTORUN ANALYSIS para ejecutar indeterminado número de veces, y debajo de esas opciones se podrá observar el comportamiento de las señales que se están buscando analizar.

3 Resultado

Lo que se obtiene después de configurar de manera correcta el dispositivo FPGA es que los datos que se introducen desde el simulink, si son muy cercanos a lo que se puede medir con el multímetro. Se puede observar que los convertidores analógico-digital y digital-analógico realizan las tareas asignadas de conversión, además que si se modifican los datos introducidos, se verá reflejado en la respuesta del multímetro y del signal tap. El signal tap no da la respuesta análoga, sino que da una respuesta en números en base decimal, con límites a binario según la conversión correspondiente, pero realizando la conversión a lo que debe dar en la salida, da una respuesta aproximada. En el convertidor análogo-digital, puse una entrada de 3.1415 como dato, y después de pasar por el ADC luego por el FPGA y después por el DAC dió una respuesta de 3.15, siendo un resultado muy aproximado a lo real.

4 Conclusiones

La realización de esta práctica, se concluye como exitosa, ya que cada configuración quedó de una manera correcta, la ayuda de los FPGA para el análisis de señales es muy adecuada por la adquisición de los datos en tiempo real. El aprender a configurar los convertidores es de gran ayuda para obtener los datos que se requieren en cierto comportamiento de señales. La conversión de los programas en VHDL con la ayuda de MATLAB, ayuda en el ahorro de tiempo de programación y solo se configura según la hoja de datos del fabricante del convertidor en el diagrama a bloques de simulink, así se puede ganar más tiempo en lo que se necesita que es la adquisición de datos para realizar las técnicas de control necesarias. Como conclusión estos elementos utilizados en la práctica serán de gran aporte para los temas de investigación que se puedan ir generando en el futuro.