

SISTEMAS DIGITALES II

ACTIVIDAD #3

CAPÍTULO DEL CURSO: CIRCUITOS SECUENCIALES SINCRÓNICOS

TEMA DE LA ACTIVIDAD: IMPLEMENTACIÓN DEL CONVERTIDOR ANALÓGICO-DIGITAL (ADC), Y LA MODULACIÓN POR ANCHO DE PULSO UTILIZANDO LENGUAJE DE DESCRIPCIÓN DE HARDWARE

OBJETIVOS DE APRENDIZAJE:

- Construir el bloque Convertidor Analógico – Digital, utilizando el catálogo de propiedad intelectual, y el bloque Modulador por Ancho de Pulsos en lenguaje de descripción de hardware VHDL, con el fin de obtener un circuito variador de pulsos digitales mediante señales analógicas.
- Analizar el código en lenguaje de descripción de hardware VHDL del bloque PWM, con el fin de cambiar parámetros dependiendo de los requerimientos del circuito.

DURACIÓN: 120 minutos

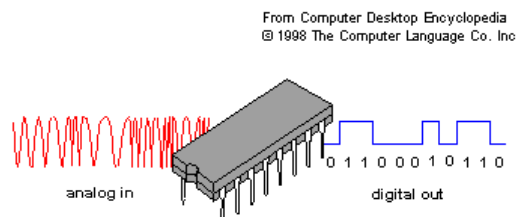
MATERIALES Y HERRAMIENTAS:

- Software Quartus Prime Lite 19.1
- Tarjeta de Desarrollo DE10-Standard.
- 1 potenciómetro: 100 K Ω

INTRODUCCIÓN:

Conversor Analógico – Digital.

Un conversor analógico-digital es un dispositivo electrónico capaz de convertir una señal analógica en un valor binario, en otras palabras, éste se encarga de transformar señales analógicas a digitales (0 y 1).



El dispositivo establece una relación entre su entrada (señal analógica) y su salida (digital) dependiendo de su resolución. La resolución determina la precisión con la que se reproduce la señal original.

Esta resolución se puede saber, siempre y cuando conozcamos el valor máximo de la entrada a convertir y la cantidad máxima de la salida en dígitos binarios.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

Resolución = $+V_{ref}/2^n$ (donde n son bits)

Por ejemplo, un conversor A/D de 8-bits puede convertir valores que van desde 0V hasta el voltaje de referencia (V_{ref}) y su resolución será de: Resolución = $V_{ref}/256$. Lo que quiere decir que mapeará los valores de voltaje de entrada, entre 0 y V_{ref} voltios, a valores enteros comprendidos entre 0 y 255 ($2^n - 1$). [1]

PWM

La modulación de ancho de pulso (PWM, por sus siglas en inglés) de una señal es una técnica que logra producir el efecto de una señal analógica sobre una carga, a partir de la variación de la frecuencia y ciclo de trabajo de una señal digital. El ciclo de trabajo describe la cantidad de tiempo que la señal está en un estado lógico alto, como un porcentaje del tiempo total que esta toma para completar un ciclo completo. La frecuencia determina que tan rápido se completa un ciclo (por ejemplo: 1000 Hz corresponde a 1000 ciclos en un segundo), y por consiguiente que tan rápido se cambia entre los estados lógicos alto y bajo. Al cambiar una señal del estado alto a bajo a una tasa lo suficientemente rápida y con un cierto ciclo de trabajo, la salida parecerá comportarse como una señal analógica constante cuanto esta está siendo aplicada a algún dispositivo.

Ejemplo: Para crear una señal de 3V dada una fuente digital que puede ser alta (5V) o baja (0V), usted podría utilizar un PWM con un ciclo de trabajo del 60%. El cual generaría una señal de 5V el 60% del tiempo. Si la señal es conmutada lo suficientemente rápido, el voltaje visto en las terminales del dispositivo parecerá ser el valor promedio de la señal. Si el estado lógico bajo es 0V (que es el caso más común) entonces el voltaje promedio puede ser calculado multiplicando el voltaje que represente el estado lógico alto por el ciclo de trabajo, o $5V \times 0.6 = 3V$. Seleccionar un ciclo de trabajo del 80% sería equivalente a 4V, un 20% a 1V, y así sucesivamente.

Señales de PWM son utilizadas comúnmente en el control de aplicaciones. Su uso principal es el control de motores de corriente continua, aunque también pueden ser utilizadas para controlar válvulas, bombas, sistemas hidráulicos, y algunos otros dispositivos mecánicos. La frecuencia a la cual la señal de PWM se generará, dependerá de la aplicación y del tiempo de respuesta del sistema que está siendo controlado. A continuación, se muestran algunas aplicaciones y sus respectivas frecuencias:

- Calentar elementos o sistemas con tiempos de respuesta lentos: 10-100 Hz o superior.
- Motores eléctricos de corriente continua: 5-10 kHz o superior.
- Fuentes de poder o amplificadores de audio: 20-200 kHz o superior. [2]

En las figuras 1, 2 y 3 se muestran las señales PWM con ciclos de trabajo del 25%, 50% y 75% respectivamente:

Editado por:

Ing. José Miguel Larrea Gando.
Jefe del Laboratorio Sistemas Digitales Avanzados

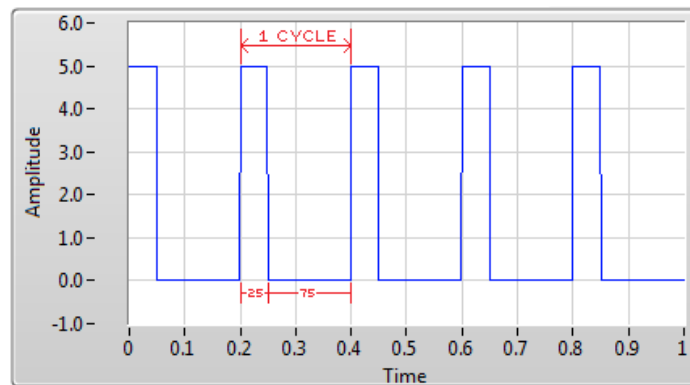


Figura 1. Ciclo de trabajo del 25%.

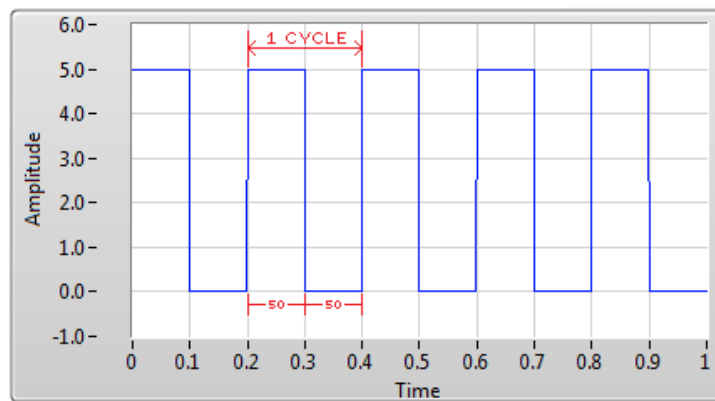


Figura 2. Ciclo de trabajo del 50%.

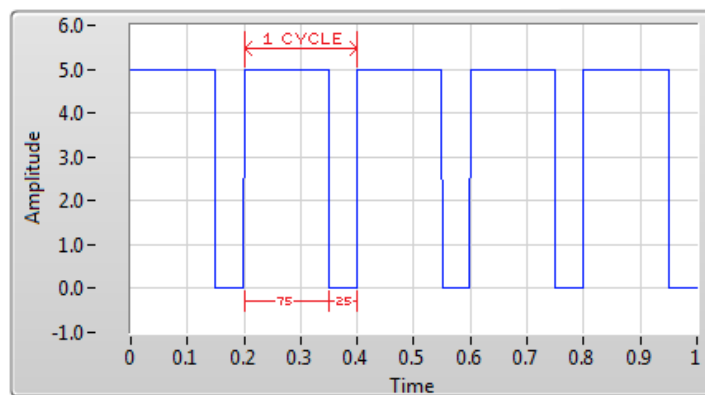



Figura 3. Ciclo de trabajo del 75%.

DESCRIPCIÓN DE LA PRÁCTICA: Se realizará el análisis de un bloque modulador por ancho de pulsos (PWM), donde se podrá modificar el ciclo de trabajo (duty cycle) con un potenciómetro a través de un convertidor analógico – digital.

PROCEDIMIENTO:

1. Proceda a crear una carpeta en el escritorio con el nombre ACTIVIDAD9 (escriba sin espacios). Luego copie y pegue todos los archivos que se encuentran en la carpeta compartida, en la carpeta que fue creada en el escritorio.
2. Ejecute el programa Quartus Prime 17.0 Standard Edition que se encuentra en el escritorio representado por el ícono .
3. Proceda a crear un nuevo Proyecto seleccionando la opción **File**→**New Project Wizard** como se observa en la Figura 4. Aparecerá la ventana de Introducción, luego da clic en **Next**.

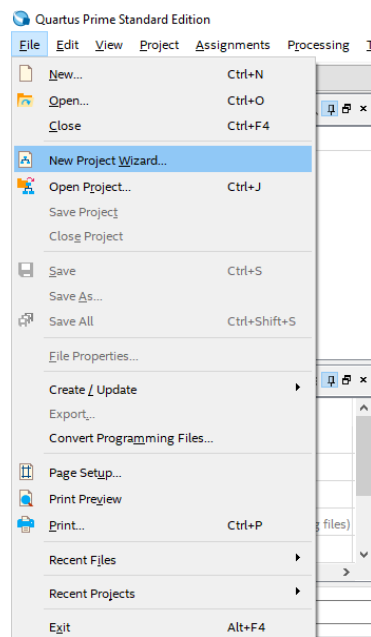


Figura 4. Creación de un nuevo proyecto.

4. En la primera línea escoja la ruta donde va a crear su proyecto, en este caso, la ruta será la carpeta ACTIVIDAD9 que fue creada en el procedimiento 1 y que se encuentra en el escritorio. En la siguiente línea escoge como nombre del proyecto ACTIVIDAD9, tal como se visualiza en la Figura 5. Luego da clic en **Next**.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

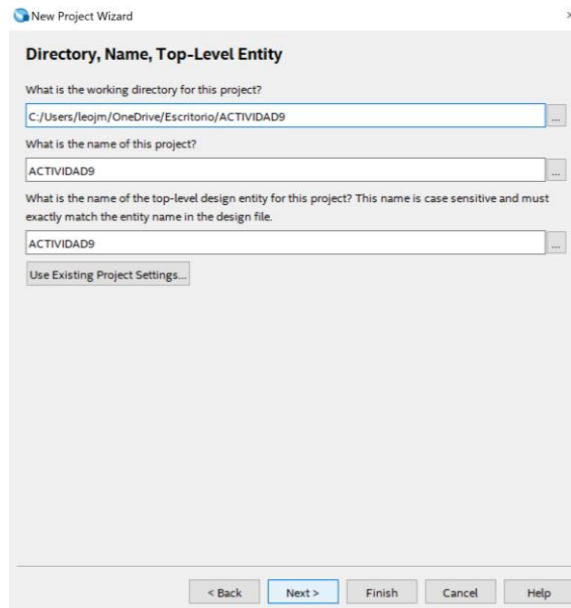


Figura 5. Nombre del proyecto.

5. Se mostrará la ventana para seleccionar el tipo de proyecto. Se escogerá la opción de Proyecto vacío (**Empty project**), tal como se visualiza en la Figura 6. Luego da clic en **Next**.

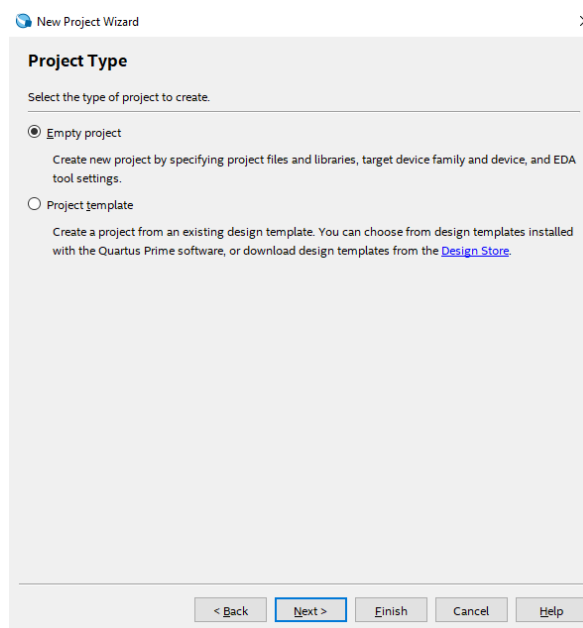


Figura 6. Tipo de proyecto.

6. Se mostrará la ventana para adhesión de archivos. En esa ventana seleccione la opción **Add All**, para que se agreguen todos los archivos que fueron copiados a la carpeta del proyecto y que se realizó en el procedimiento 1. En la Figura 7 se observan los archivos añadidos. Por último, haga clic en **Next**.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

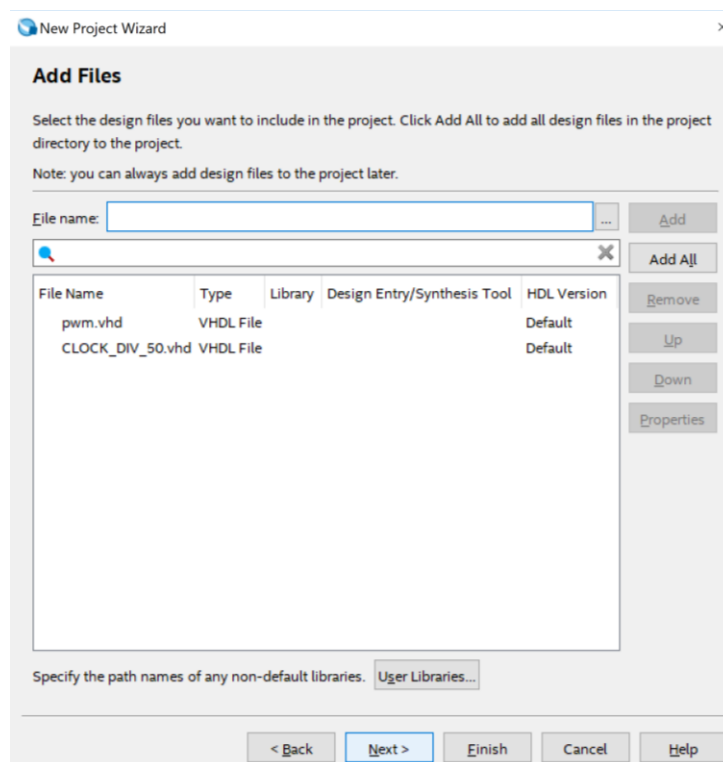


Figura 7. Adhesión de archivos.

- En la siguiente ventana aparecerán todas las familias de chips FPGA. Proceda a escoger la familia **Cyclone V (E/GX/GT/SX/SE/ST)** y seleccione el nombre del chip **FPGA SoC 5CSXFC6D6F31C6**, tal como se muestra en la Figura 8. Luego haga clic en **Next**.

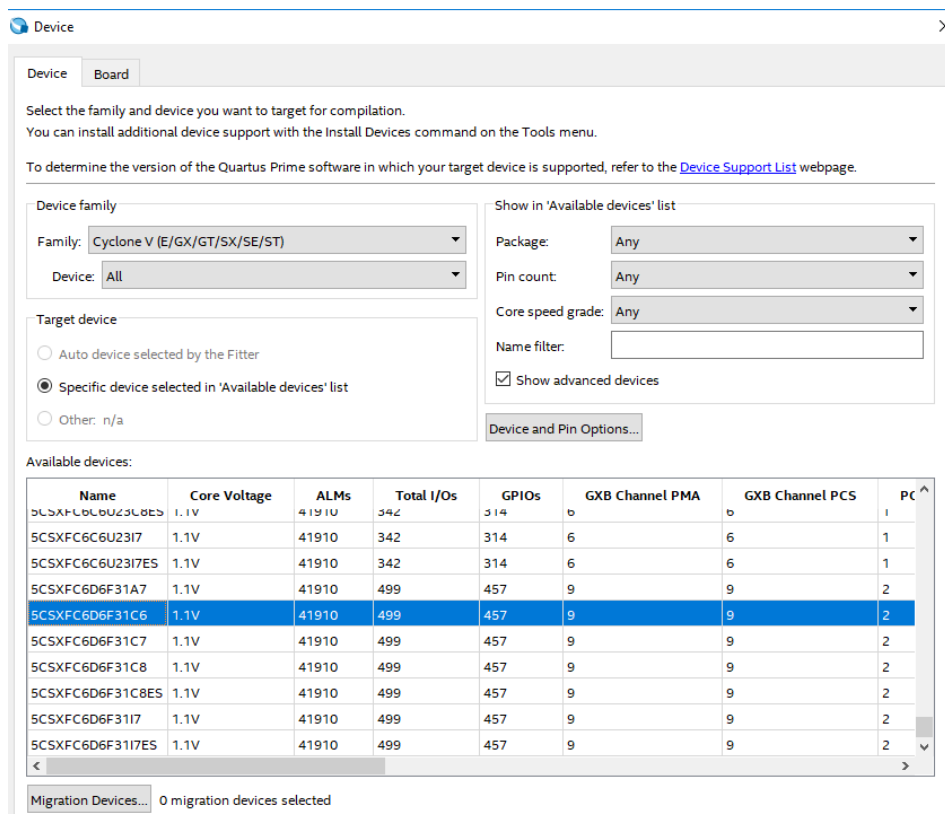


Figura 8. Familia y modelo del chip FPGA SoC.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

8. Debido a que no se realizarán simulaciones en la presente actividad, no se seleccionará alguna herramienta de simulación como se muestra en la Figura 9. Haga clic en **Next**.

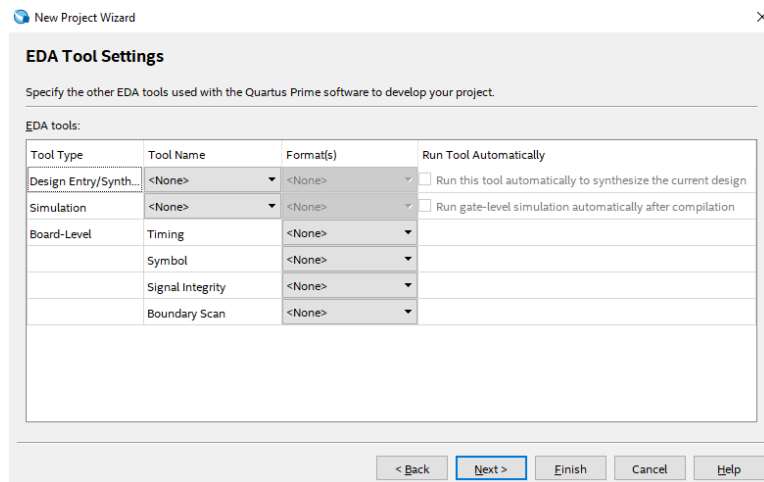


Figura 9. Herramientas para simulación.

9. Al finalizar la creación del proyecto, aparecerá la ventana **Summary** para resumir todas las especificaciones que tendrá nuestro proyecto. Para finalizar haga clic en **Finish**.
10. Una vez creado el proyecto, proceda a crear el símbolo de los archivos **pwm.vhd** y **CLOCK_DIV_50.vhd** para agregarlos en el editor gráfico. Para crear el símbolo de cada uno, haga clic derecho en cada archivo y seleccione la opción **Create Symbol Files for Current File**, tal como se observa en la Figura 10.

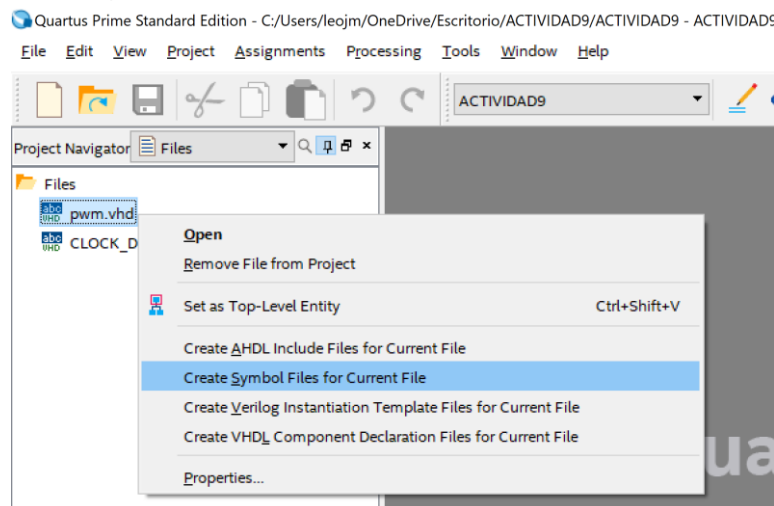


Figura 10. Creación del símbolo para el archivo PWM.vhd.

11. Proceda a crear un archivo editor gráfico, para ello haga clic en **File**→**New**, y seleccione **Block Diagram/Schematic File** tal como se muestra en la Figura 11.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

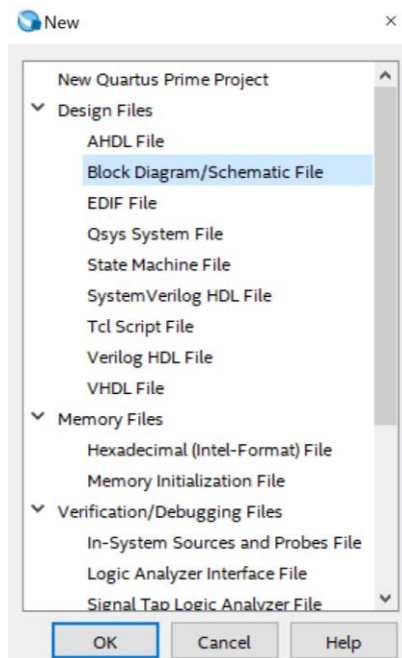


Figura 11. Creación de un archivo Editor Gráfico

12. En el editor gráfico construya el circuito digital que se encuentra en la Figura 12, es de importancia mencionar que el bloque **ADC** va a ser elaborado por medio del catálogo de propiedad intelectual de Quartus (**IP Catalog**).

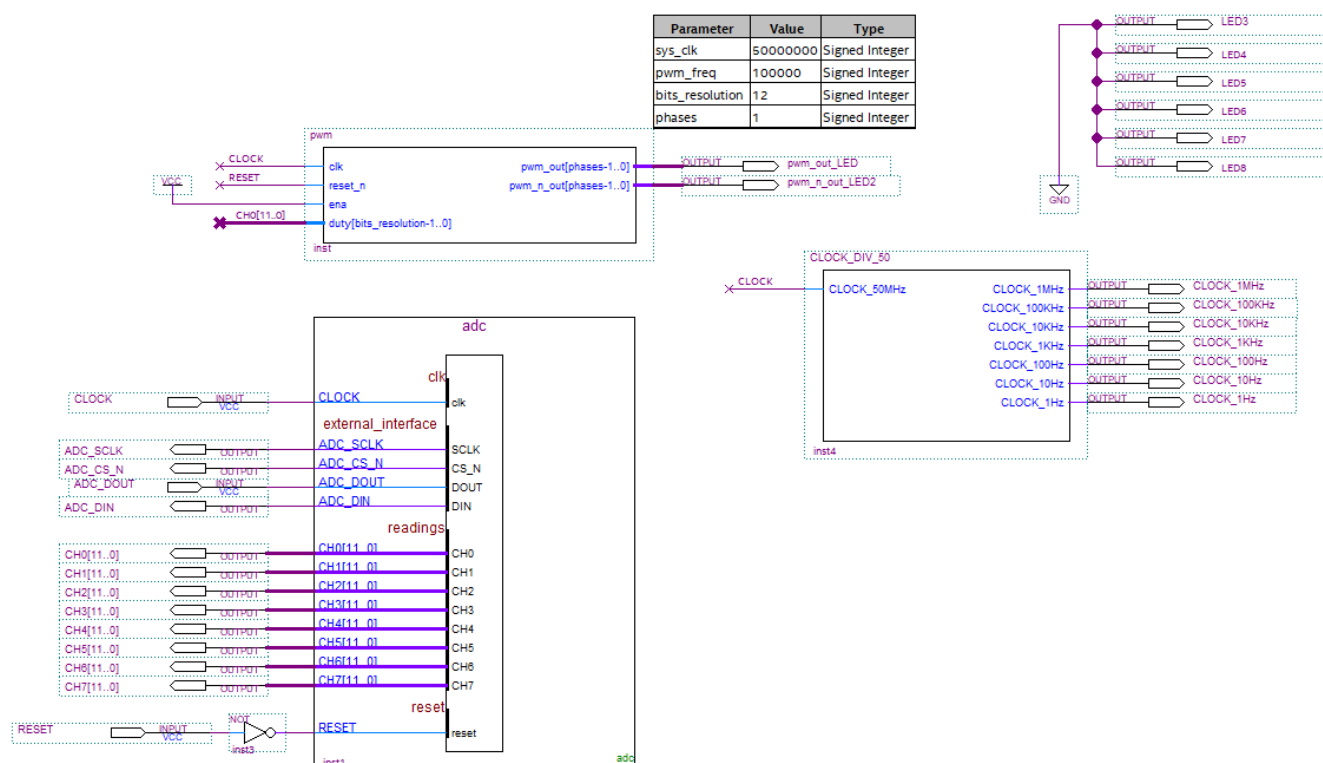


Figura 12. Circuito Digital

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

13. Para agregar el bloque **ADC**, seleccione en el menú **Tools** → **IP Catalog**, como se visualiza en la Figura 13.

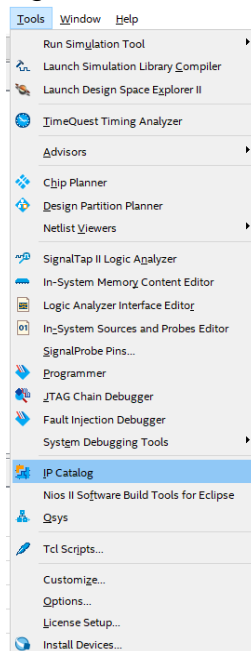


Figura 13. Selección de Catálogo de Propiedad Intelectual.

14. En el buscador de la ventana **IP Catalog** escriba “ADC”, y seleccione haciendo doble clic en “**ADC Controller for DE-series Boards**”, como se visualiza en la Figura 14.

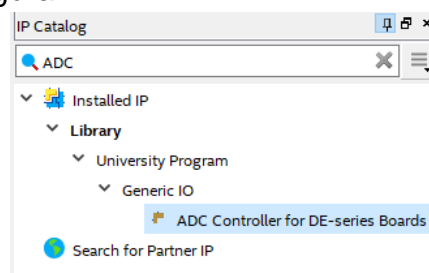


Figura 14. Ventana IP Catalog.

15. Se abrirá la ventana de **Qsys** para establecer diferentes parámetros a nuestro bloque convertidor analógico – digital. Proceda a colocar el nombre “**ADC**” a nuestro archivo de **Qsys**, tal como se visualiza en la Figura 15. Asegúrese que la familia del chip sea **Cyclone V** y el modelo **5CSXFC6D6F31C6**, haga clic en **OK**.

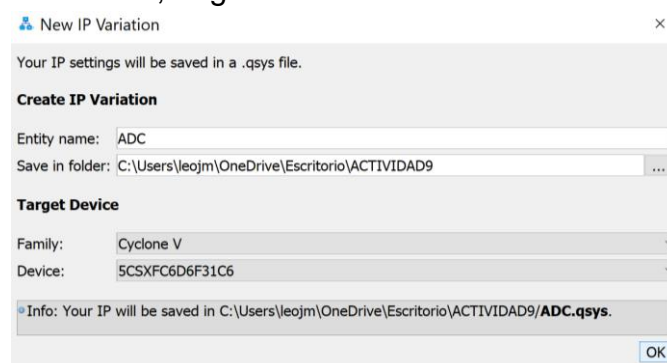


Figura 15. Creación de un bloque IP.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

16. En los parámetros del bloque ADC deben de estar colocado el nombre de la tarjeta DE10-Standard, la frecuencia de muestreo del ADC a 12.5 MHz y el número de canales a usar que son 8, como se visualiza en la Figura 16.

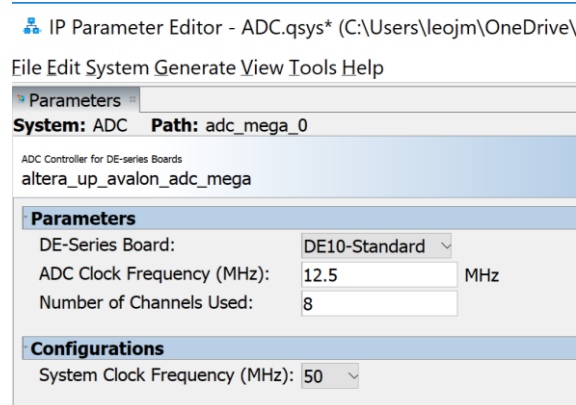


Figura 16. Establecimiento de parámetros en el bloque IP.

17. Una vez asignado todos los valores, haga clic en **Generate HDL** que se encuentra en la parte inferior derecha de la ventana **IP Parameter Editor**.
18. Asegúrese que la casilla para crear el .bsf (**Block Symbol File**) se encuentre activada, tal como se visualiza en la Figura 17. Para finalizar haga clic en **Generate**.

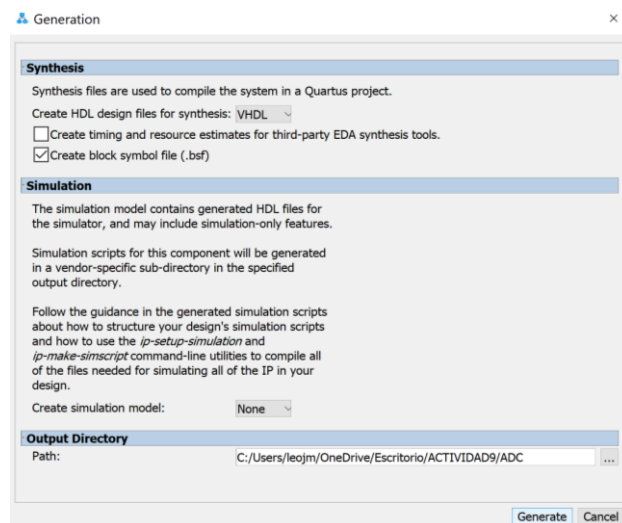


Figura 17. Generación del bloque IP.

19. Haga clic en **Close** y para finalizar haga clic en **Finish**.
20. Guarde el archivo Editor Gráfico como ACTIVIDAD9.bdf y agregue el archivo **ADC.qsys** en la ventana **Project Navigator**, tal como se visualiza en la Figura 18.

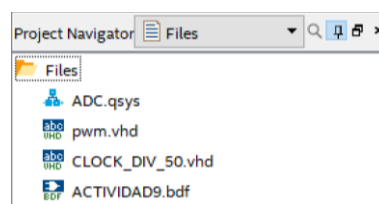


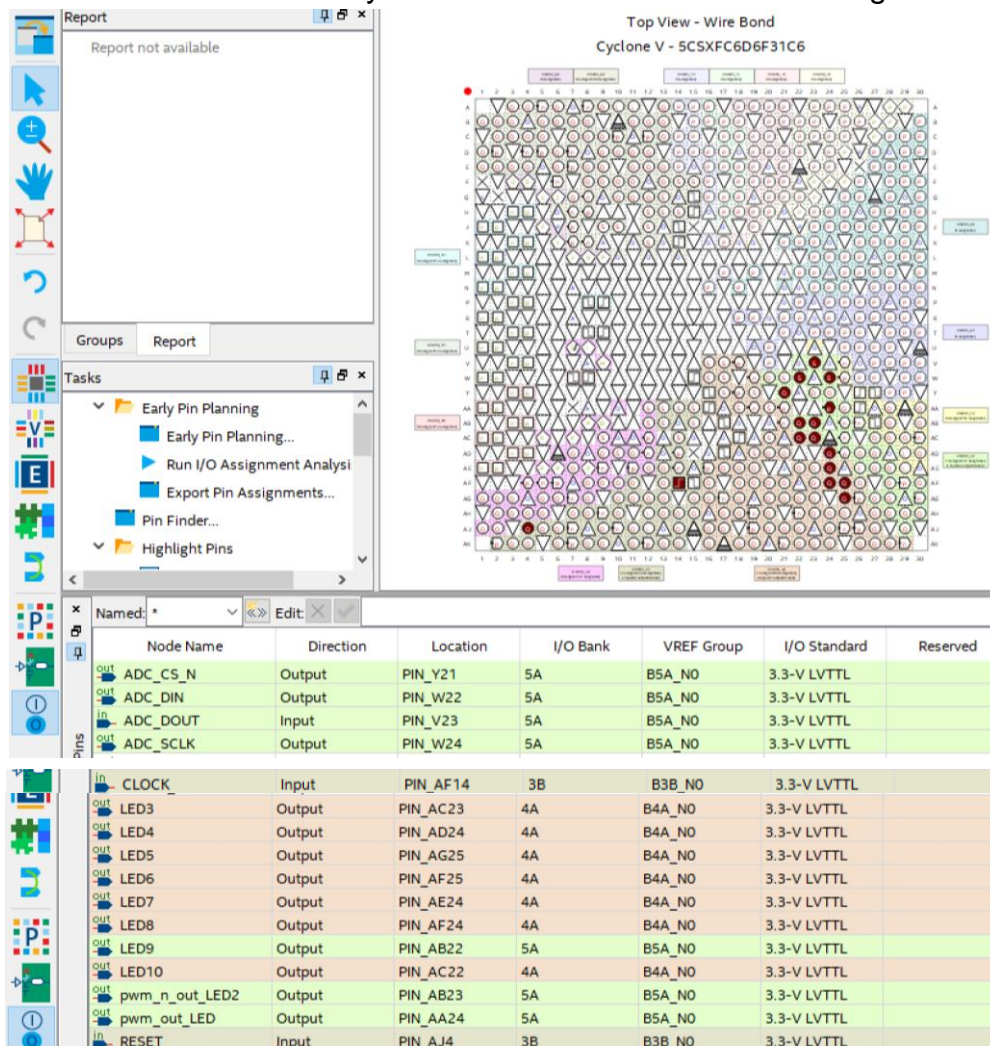
Figura 18. Bloques en la ventana Project Navigator.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

21. Termine de completar el circuito digital que se encuentra en el paso 12. Proceda a compilar el editor gráfico **ACTIVIDAD9.bdf**.
22. En el menú **Assignments**, seleccione **Pin Planner**. Asigne los pines de las señales de entrada y salida como se encuentra en la Figura 19.



The screenshot shows the Pin Planner interface. On the left, a 'Report' window indicates 'Report not available'. Below it, a 'Tasks' panel lists actions like 'Early Pin Planning', 'Run I/O Assignment Analysis', 'Export Pin Assignments', 'Pin Finder', and 'Highlight Pins'. The main area displays a 'Top View - Wire Bond' of a 'Cyclone V - 5CSXFC6D6F31C6' chip. On the right, a table lists the assigned pins and their configurations.

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
ADC_CS_N	Output	PIN_Y21	5A	B5A_NO	3.3-V LVTTTL	
ADC_DIN	Output	PIN_W22	5A	B5A_NO	3.3-V LVTTTL	
ADC_DOUT	Input	PIN_V23	5A	B5A_NO	3.3-V LVTTTL	
ADC_SCLK	Output	PIN_W24	5A	B5A_NO	3.3-V LVTTTL	
CLOCK_	Input	PIN_AF14	3B	B3B_NO	3.3-V LVTTTL	
LED3	Output	PIN_AC23	4A	B4A_NO	3.3-V LVTTTL	
LED4	Output	PIN_AD24	4A	B4A_NO	3.3-V LVTTTL	
LED5	Output	PIN_AG25	4A	B4A_NO	3.3-V LVTTTL	
LED6	Output	PIN_AF25	4A	B4A_NO	3.3-V LVTTTL	
LED7	Output	PIN_AE24	4A	B4A_NO	3.3-V LVTTTL	
LED8	Output	PIN_AF24	4A	B4A_NO	3.3-V LVTTTL	
LED9	Output	PIN_AB22	5A	B5A_NO	3.3-V LVTTTL	
LED10	Output	PIN_AC22	4A	B4A_NO	3.3-V LVTTTL	
pwm_n_out_LED2	Output	PIN_AB23	5A	B5A_NO	3.3-V LVTTTL	
pwm_out_LED	Output	PIN_AA24	5A	B5A_NO	3.3-V LVTTTL	
RESET	Input	PIN_AJ4	3B	B3B_NO	3.3-V LVTTTL	

Figura 19. Asignación de Pines.

23. Una vez que terminó de asignar los pines, cierre el **Pin Planner** y proceda a compilar nuevamente el editor gráfico **ACTIVIDAD9.bdf**.
24. Proceda a conectar la tarjeta de desarrollo DE10-Standard a la PC.
25. En el menú **Tools**, seleccione la herramienta **SignalTap II LOGIC ANALYZER**. En el programador embebido en el SignalTap coloque en **Hardware**→**USB-BLASTER**; en **SOF Manager** agregue el archivo **ACTIVIDAD9.SOF** de su proyecto que se encuentra en la carpeta **output_files**. En la Figura 20 se muestra los valores anteriormente mencionados.



Figura 20. Programador embebido en el SIGNALTAP II LOGIC ANALYZER.

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

26. Seleccione como muestreo a la señal **CLOCK_1MHz**. Debe de recordar que la señal **CLOCK_1MHz** es una señal de salida, por lo tanto, debe de filtrar su búsqueda como **Pins: all** y haga clic en **List** para observar todas las entradas y salidas del bloque como se visualiza en la Figura 21.

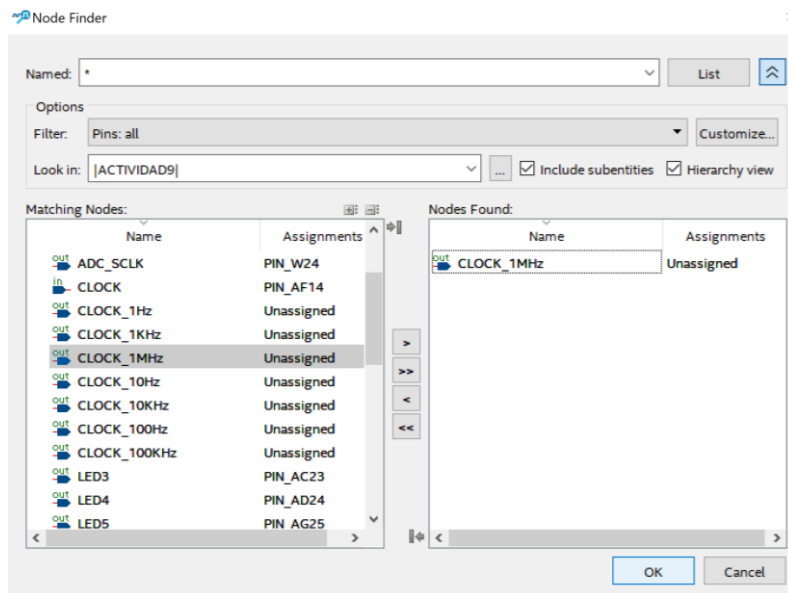


Figura 21. Filtro de señales a través de Pins: all.

27. Agregue para analizar las señales de entrada y salida como se observa en la Figura 22. Debe de recordar que, para las señales de entrada y salida, se puede filtrar su búsqueda como **Pins: all** y hace clic en **List** para observar todas las entradas, salidas del bloque.



Figura 22. Señales de entrada y salida.

28. Guarde el archivo SignalTap con el nombre que desee sin espacios, y luego proceda a compilar el editor gráfico **ACTIVIDAD9.bdf**.
29. Una vez que terminó de compilar, programe la tarjeta de desarrollo desde el programador embebido ubicado en el SignalTap haciendo clic en el ícono de la flecha hacia abajo (**Program Device**) como se muestra en la Figura 23.

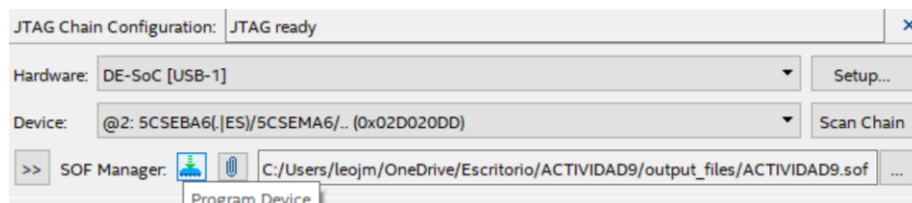


Figura 23. Programación del chip FPGA (Program Device).

Editado por:

Ing. José Miguel Larrea Gando.

Jefe del Laboratorio Sistemas Digitales Avanzados

30. Presione el ícono **Autorun Analysis** (símbolo de reproducir con un círculo color verde) como se muestra en la Figura 24. También puede aplastar la tecla F6 que tiene el mismo resultado.

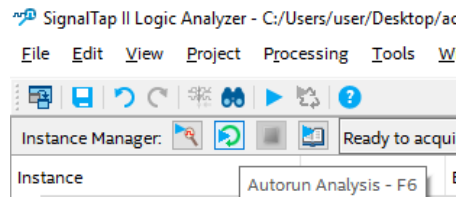
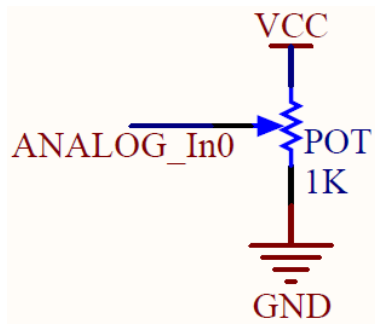


Figura 24. Compilación desde el SignalTap (Autorun Analysis).

31. Proceda a realizar la configuración del circuito de la Figura 25, tomando en consideración que las señales son conectadas al **conector J15 ADC_CON** de la tarjeta de desarrollo DE10-Standard.



PIN FPGA	FUNCIÓN
1	VCC
2	ANALOG_In0
10	GND

Figura 25. Conexión física del circuito.

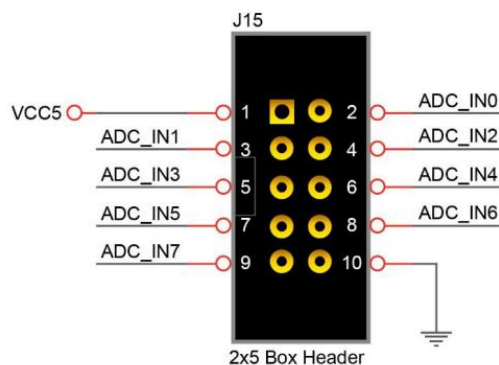


Figura 26. Distribución de pines del conector J15 visto desde la parte inferior de la tarjeta.

32. Conteste la pregunta #1.
 33. Conteste la pregunta #2.
 34. Conteste la pregunta #3.
 35. Conteste la pregunta #4.
 36. Determine las conclusiones y recomendaciones de la práctica.

Bibliografía:

[1]. Arduino. Conversor Analógico – Digital (A/D). Enlace:
<https://playground.arduino.cc/ArduinoNotebookTraduccion/Appendix6>.
Consulta: 7 de Noviembre de 2022.

[2]. National Instruments. Tema: ¿Qué es una Señal Modulada por Ancho de Pulso (PWM) y Para Qué es Utilizada?. Enlace:
<http://digital.ni.com/public.nsf/allkb/AA1BDEA4AA224E3E86257CE400707527>.
Consulta: 7 de noviembre de 2022.