 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

## **ADQUISICIÓN DE SEÑALES EN ZEDBOARD POR MEDIO DEL MÓDULO XADC**

**Fabián David Gómez Bolaños**  
**Ingeniería Electrónica**

**Luis Fernando Castaño**

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**MARZO 2017**

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Resumen

En el presente informe se muestra una descripción del método de programación por bloques del módulo XADC, utilizado mediante las tarjetas de desarrollo ZedBoard como una herramienta auxiliar en la adquisición de señales análogas, logrando realizar por medio de códigos en lenguaje C, posibles análisis y/o filtrado de datos.

Mediante el software de programación VIVADO HLx y Xilinx SDK se realiza la correspondiente programación del módulo y se logra la visualización en tiempo real de los valores de las variables obtenidas. Se realiza además con el fin de servir como insumo en el laboratorio de Sistemas de Control y Robótica para futuros trabajos académicos tanto en la parte de pregrado de Ingeniería Electrónica como para los grupos de investigación en Parque I de la institución.

Palabras claves: XADC, Adquisición de datos, VIVADO, SDK, Xilinx.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## **Reconocimientos**

*El principal reconocimiento se lo quiero realizar a mi madre Piedad Gómez, quien es la principal razón y motivación en la realización de los proyectos a lo largo de mi carrera como persona y desarrollo profesional. Seguido un agradecimiento a mis tíos Teresa y Conrado quienes fueron las personas que me apoyaron en el desarrollo de mi carrera profesional. Un gran agradecimiento a todos mis compañeros de estudio que compartieron conmigo al transcurrir de los años con los cuales formamos excelentes amigos de trabajo, que al final se convirtieron en amigos de vida.*

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Acrónimos

XADC Xilinx Analog to Digital Converter

SDK Software Development Kit

PL Programmable Logic

PS Processing System

MIO (Multiplexed Input/output)

TCL Tools Comand Languages

SoC System on Chip

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Tabla de contenido

Tabla de contenido.....	6
Tabla de figuras.....	7
1. Introducción.....	10
2. Estado del arte .....	11
3. Metodología .....	13
3.1 Desarrollo Instructivo .....	14
3.1.1 Paso 1. Creación del RTL Project.....	15
3.1.3. Paso 3. Adicionar bloques IP.....	22
3.1.4. Paso 4. Organización de bloques e interconexión .....	25
3.1.5. Paso 5. Configuración del bloque XADC .....	26
3.1.6. Paso 6. Generación de HDL Wrapper y Constraints.....	30
3.1.7. Paso 7. Síntesis y Generación Bitstream.....	33
3.1.8. Paso 8. Exportar proyecto y lanzar a SDK.....	37
3.1.9.Paso 9. Creación Application Project, Creación de algoritmo.....	38
4. Resultados .....	43
5. Conclusiones, recomendaciones y trabajo futuro.....	46
6. Referencias .....	47

## Tabla de figuras

Figura 1. Partes de una ZedBoard .....	13
Figura 2. Parte interna de una ZedBoard .....	14
Figura 3. Modulo Físico XADC.....	15
Figura 4. Diagrama de Flujo proyecto.....	18
Figura 5. VIVADO HLx Pantalla Principal.....	19
Figura 6. VIVADO Creación de un nuevo proyecto .....	20
Figura 7. VIVADO Nombre del proyecto.....	20
Figura 8. VIVADO Tipo de proyecto.....	21
Figura 9. VIVADO Agregar fuentes al proyecto.....	21
Figura 10. VIVADO Agregar IP existentes.....	22
Figura 11. VIVADO Agregar constraints existentes.....	22
Figura 12. VIVADO Elección del hardware-ZedBoard .....	23
Figura 13. VIVADO Resumen configuración de Hardware.....	23
Figura 14. VIVADO Administrador de proyecto.....	24
Figura 15. VIVADO Creación diseño de bloques.....	25
Figura 16. VIVADO Espacio Diagram para desarrollo de proyecto.....	25
Figura 17. VIVADO Adicionar bloques IP al proyecto .....	26
Figura 18. VIVADO Búsqueda bloque IP XADC .....	26
Figura 19. VIVADO Bloque XADC con puertos de conexión .....	27
Figura 20. VIVADO Agregar nuevo bloques IP al proyecto.....	27
Figura 21. VIVADO Agregar bloque ZYNQ Processing System.....	28
Figura 22. VIVADO Ejecución Run Connection Automation .....	29
Figura 23. VIVADO Conexión puerto a puerto de los bloques.....	29
Figura 24. VIVADO Bloques organizados con Regenerate Layout.....	30
Figura 25. VIVADO Configuración XADC-Basic.....	30

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Figura 26. VIVADO Configuración XADC-ADC Setup .....	31
Figura 27. VIVADO Configuración XADC-Alarmas.....	31
Figura 28. VIVADO Configuración XADC- Channel Sequenser .....	32
Figura 29. VIVADO Resumen configuración XADC .....	32
Figura 30. VIVADO Creación de puertos.....	33
Figura 31. VIVADO Nombramiento de los puertos .....	33
Figura 32. VIVADO Creación HDL Wrapper .....	34
Figura 33. VIVADO Carpeta HDL Wrapper.....	35
Figura 34. VIVADO Agregar Constraints .....	35
Figura 35. VIVADO Agregar constraints - Add Sources.....	36
Figura 36. VIVADO Nombre Constraints .....	36
Figura 37. VIVADO Código Constraints.....	37
Figura 38. VIVADO Run Synthesis.....	38
Figura 39. VIVADO Synthesis completa.....	38
Figura 40. VIVADO Abrir implementación .....	39
Figura 41. VIVADO Distribución ZedBoard .....	39
Figura 42. VIVADO Como abrir esquemático .....	40
Figura 43. VIVADO Esquemático proyecto .....	40
Figura 44. VIVADO Generación Bitstream .....	41
Figura 45. VIVADO Exportar Hardware .....	41
Figura 46. Software Development Kit SDK .....	42
Figura 47. SDK Creación Application Project .....	42
Figura 48. SDK Nombre Application Project .....	43
Figura 49. SDK Agregar helloworld.c.....	43
Figura 50. SDK Conexión ZedBoard-PC .....	44
Figura 51. SDK Run helloworld.c.....	45
Figura 52. SDK Terminal helloworld.....	45



	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Figura 53. SDK Código Conversión. .... 46

Figura 54. Voltaje de entrada Vp/Vn. 0,354V ..... 46

Figura 55. SDK Terminal Vp/Vn 0,354V ..... 47

Figura 56. Voltaje de entrada Vp/Vn 0,853V..... 47

Figura 57. SDK Terminal Vp/Vn 0,853v ..... 48

Figura 58. Voltaje de entrada Vp/Vn 1,055v ..... 48

Figura 59. SDK Terminal Vp/Vn 0,999v..... 49

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 1. Introducción

En el presente trabajo se muestra un método para la adquisición de señales, empleando el modulo XADC(Xilinx Analog to Digital Converter). Esta herramienta permite ampliar el campo de aplicación de la tarjeta de desarrollo ZedBoard. En este proyecto el conjunto ZedBoard-XADC y su descripción se utiliza con fines académicos, siendo apoyo para la realización de practicas de laboratorio en materias afines con este tipo de dispositivos. En este proyecto se hace uso de los entornos VIVADO HLx Edition y SDK(Software Development Kit), permitiendo el uso del módulo XADC con programación de bloques IP y obteniendo la visualización por medio de códigos en lenguaje C.

El desarrollo del trabajo tiene como principal objetivo realizar la descripción paso a paso para la programación del módulo XADC y obtener así una visualización en tiempo real de las variables adquiridas en forma de voltaje siendo transformadas a valores tipo dato.

Este trabajo cuenta con una sección en la cual se describen algunos artículos, donde presentan como herramienta principal, el modulo XADC y la metodología utilizada. Una sección donde se expone la metodología describiendo la forma de cómo se realizó la implementación y las herramientas empleadas. Una sección donde se presentan los resultados del trabajo realizando una descripción paso a paso el cual es plasmado en un tutorial. Una sección de discusión de los resultados obtenidos y finalmente una sección de conclusiones y recomendaciones del trabajo final.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2. Estado del arte

Desarrollos de diseño haciendo uso de la tarjeta ZedBoard, y que tienen como objetivo realizar aplicaciones con adquisición de señales análogas por medio del bloque XADC, son descritos en artículos como el de *Model-Based Reference Design Projects with MathWorks' HDL Workflow Advisor for Custom-Specific Electronics with the Zedboard* (Versen, Kipfelsberger, & Sökmen, 2016) donde se realiza un acople entre dos entornos de desarrollo como VIVADO (Xilinx) y Matlab (MathWorks). Este trabajo cuenta con dos tipos de lenguaje, por un lado se utiliza el lenguaje HDL Workflow Advisor en Matlab, mientras que en VIVADO se realiza una programación de bloques AXI (Advanced eXtensible Interface).

Versen, Kipfelsberger & Sökmen demuestran su desarrollo mediante dos modelos experimentales. El primer modelo experimental se describe realizando la conversión de los datos entregados por un LDR(Light Dependent Resistor) y un potenciómetro que ingresan a la ZedBoard mediante la programación de los bloques AXI configurando y asignando puertos de entrada y salida, mientras que en el HDL Workflow Advisor se convierte el modelo en una descripción HDL el cual tiene como principal limitación para la implementación las entradas y salidas debido a que estas se definen en un proyecto en VIVADO. Luego de realizada la conversión, los datos son enviados al procesador en Matlab. Después de realizada la conversión se toma los cuatro bits más altos y se utilizan como valores de comparación generando una señal de impulso para un “*Buzzer*” o *Zumbador* por medio de un pin en el FMC como medida para la frecuencia de la señal de salida.

Otro experimento realizado para obtener un análisis más cuantitativo se realiza con los conectores SMA (Conector de cable coaxial) permitiendo entradas analógicas reemplazando así el LDR y potenciómetro anterior por un generador de señales, que entrega una señal sinusoidal con un offset de 0.9Vdc a una frecuencia de 1kHz digitalizada por medio de los AXI siendo enviada al Pc Host de Matlab. Luego de ser descargado y arrancado en el ARM los datos se muestran por medio de Simulink. Este experimento permite conocer los tiempos de muestreo que se pueden obtener variando la resolución de la muestra.

Este desarrollo trae como conclusión su buen aporte para aplicaciones de mecatrónica, usos académicos en cuanto al acople entre plataformas, trabajos futuros con el XADC, y definir según el tipo de muestreo que se requiera la resolución que permitan una mejor visualización de los datos obtenidos en tiempo real.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Otro trabajo que desarrolla aplicaciones con la ZedBoard y el bloque XADC es el llamado: *Utilization of xilinx fpga built-in analog-to-digital converter for data acquisition in power electronics converter* (Ling & Norum, 2016), donde se realiza una descripción de los controladores multinivel y como por medio del bloque XADC realizar la adquisición de señales análoga a la salida de estos dispositivos, logrando un control en lazo cerrado a alta velocidad satisfaciendo la demanda de voltaje en aplicaciones de media y alta tensión.

Como se menciona el principal objetivo del proyecto, es la utilización de un convertidor modular multinivel, logrando sintetizar de mejor manera la salida sinusoidal con varios escalones de voltaje por medio de submódulos. Al tratarse de un convertidor de varios niveles, es necesario realizar mediciones para el control de voltaje y de corriente a la salida, por lo tanto la Zedboard en conjunto con el bloque XADC que se encargara de enviar los datos obtenidos hacia el procesador.

El proyecto hace uso de un multiplexor externo el cual se encargara de modificar los canales diferenciales (VexP, VexN) de lectura y de las direcciones para el almacenamiento de los datos. Con ayuda de programación Round-Robin (toma turnos), el multiplexor dirige y controla los canales que se encuentran en servicio, permitiendo realizar acciones en tiempos determinados. Este modelo permitirá nuevas lecturas mientras que la conversión se está ejecutando y continuará nuevamente el ciclo de lecturas una vez finalice. De lo contrario no se almacenara ese dato y el sistema saltara a otra instrucción.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3. Metodología

En el desarrollo de este trabajo se utiliza herramientas de desarrollo Hardware y Software. Para la parte de hardware se usa la tarjeta ZedBoard, la cual hace parte de los dispositivos Zynq-7000 All Programmable SoC encargada de realizar la adquisición y el procesamiento de los datos. Una fuente de tensión DC y un multímetro con los cuales se realizan las pruebas de campo. Si se requiere obtener más información acerca de la tarjeta Zedboard se puede consultar manuales de fabricante en la página [www.zedboard.org](http://www.zedboard.org).

Para la parte de software se hace uso de entornos como VIVADO HLx, el cual permite la configuración del hardware por medio de los bloques IP, permitiendo realizar la adquisición de datos. Y el entorno SDK que permite la lectura de los datos una vez realizada la conversión por la ZedBoard.

Para mostrar el manejo de la herramienta de conversión análogo digital que brinda la ZedBoard por medio del bloque XADC, se sigue la siguiente serie de pasos como se muestra en el diagrama de flujo (Figura 1), el cual direcciona el procedimiento para una correcta configuración del hardware y periféricos que hacen parte del desarrollo. La configuración se realiza en el software VIVADO HLx Edition 2016.2. Luego en el software SDK se configura y documenta el algoritmo, el cual se encargara de realizar el procesamiento de las señales de voltaje, la conversión de señales y la visualización de los valores que se obtienen en tiempo real. Esto con el fin de presentar un instructivo académico para materias relacionadas con Sistemas de Control y Robótica.

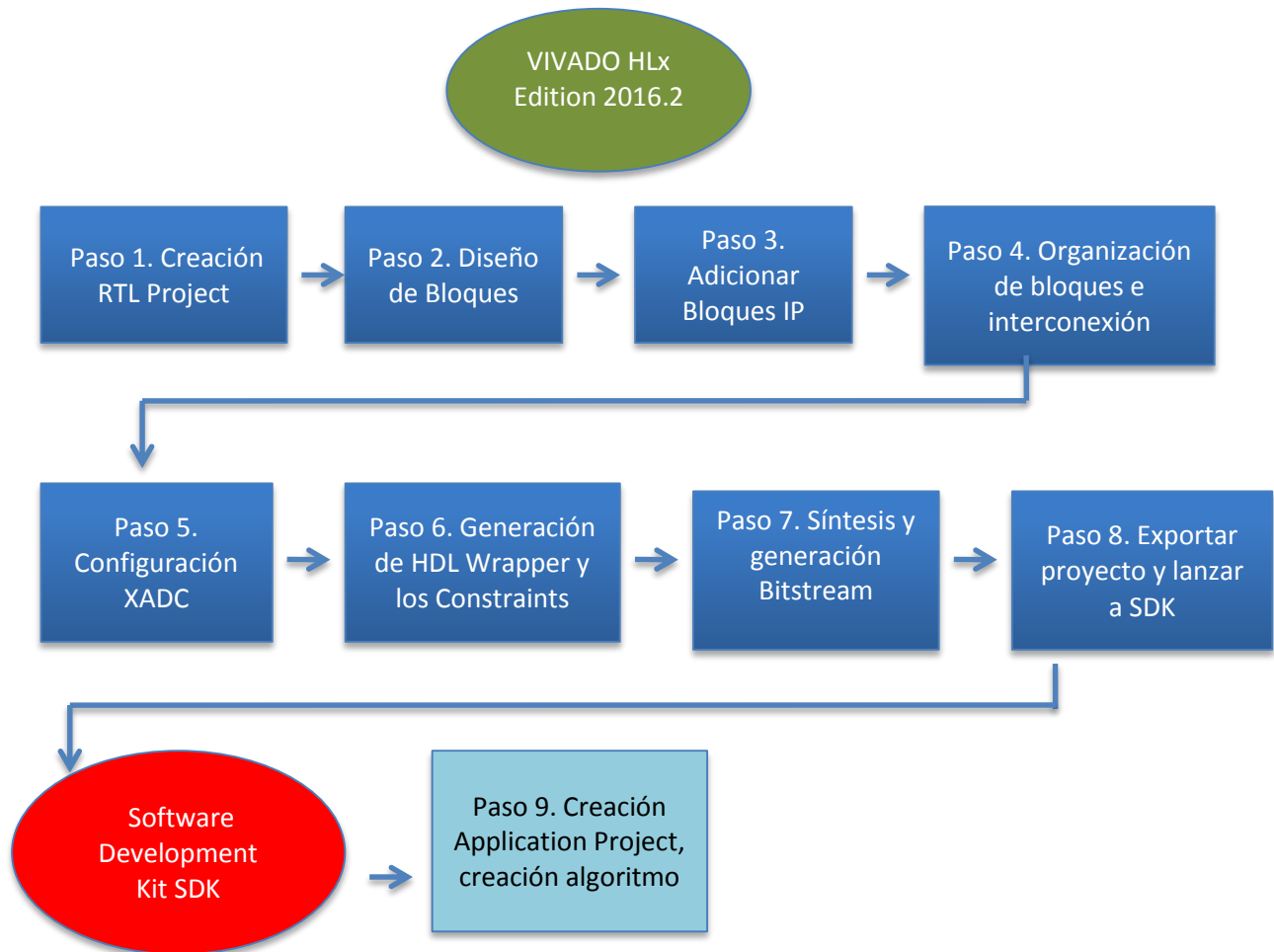


Figura 1. Diagrama de flujo proyecto

### 3.1 Desarrollo Instructivo

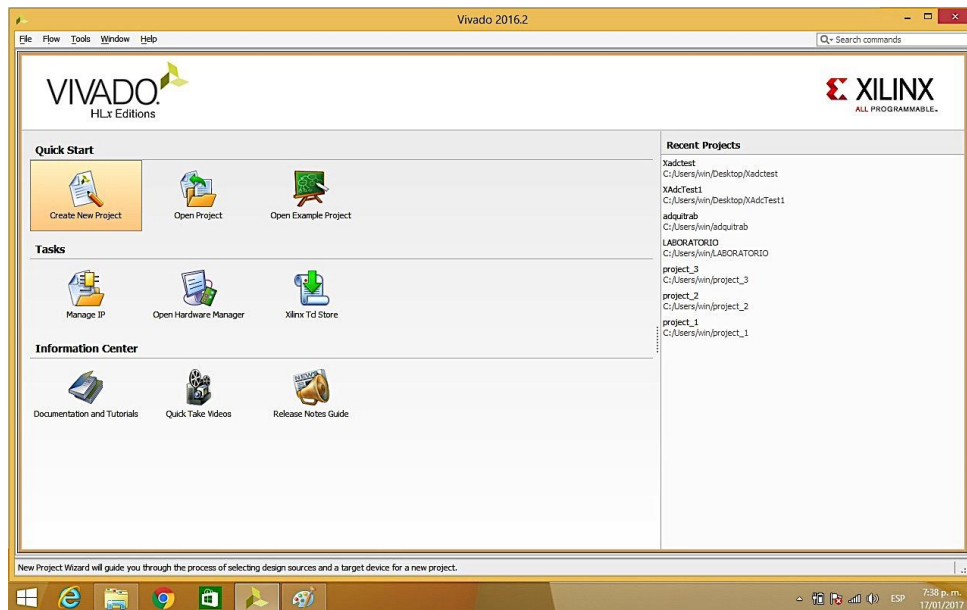
En este instructivo se utilizara la versión 2016.2 del software de desarrollo VIVADO, siendo una de las versiones más actuales. Si no se cuenta con el software instalado en el computador, esta versión y/o versiones anteriores pueden ser descargadas por medio de la página de Xilinx.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## VIVADO HLx

En la ventana de inicio de VIVADO, se encuentran las diferentes opciones que el usuario puede ejecutar como se muestra en la Figura 2.

Se crea un nuevo proyecto mediante la etiqueta “Create New Project” donde se configura el tipo de hardware utilizado y las características que el proyecto requiere.

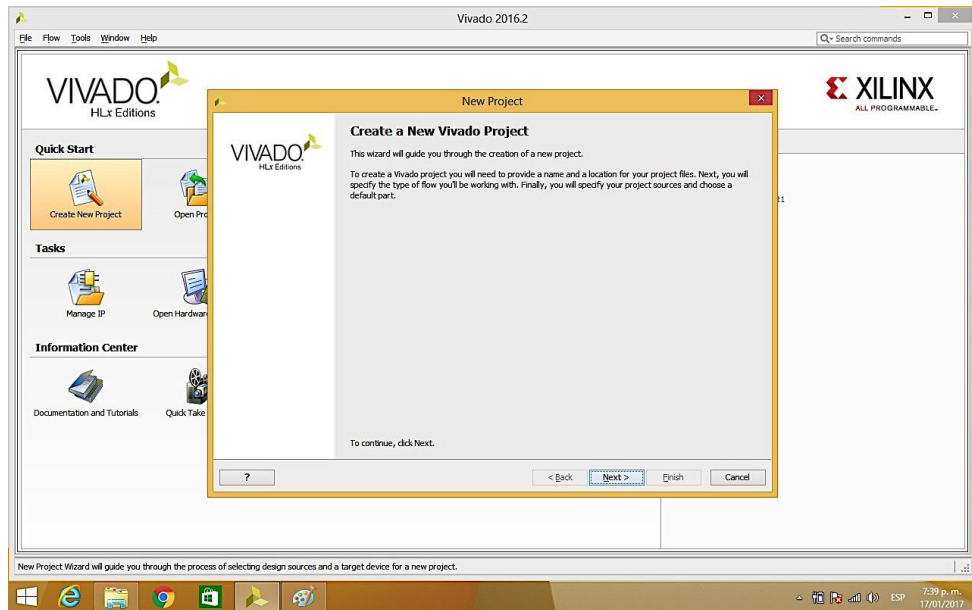


**Figura 2. VIVADO HLx Pantalla Principal**

### **3.1.1 Paso 1. Creación del RTL Project.**

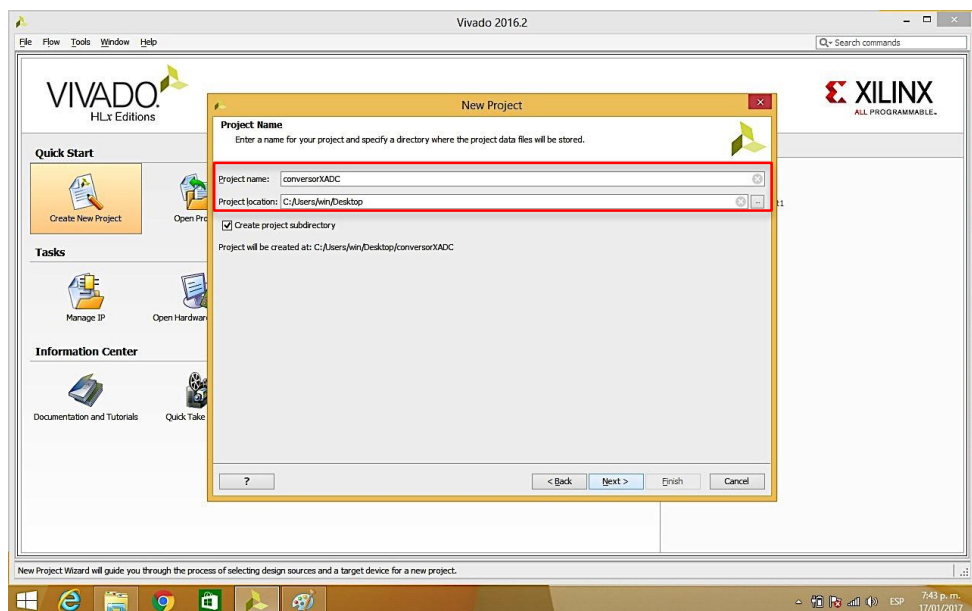
Por tratarse de un nuevo proyecto, VIVADO acompañará a los usuarios para elegir las opciones de parametrización en subventanas y diferentes menús con base a los dispositivos compatibles a este software.

 <b>Institución Universitaria</b>	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



**Figura 3. VIVADO Creación de un nuevo proyecto**

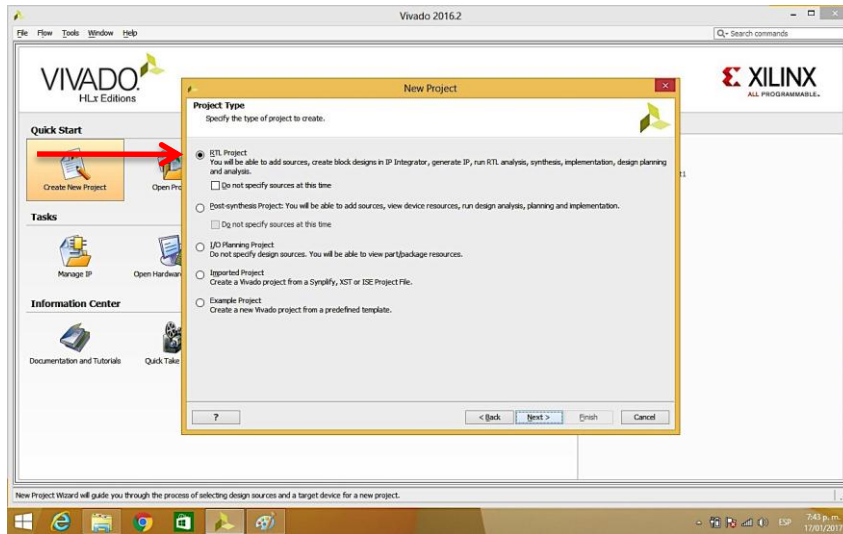
Se asigna un nombre al proyecto fácil de identificar, y se asigna la carpeta donde el proyecto será guardado con todos los archivos generados con la configuración.



**Figura 4. VIVADO Nombre del proyecto**



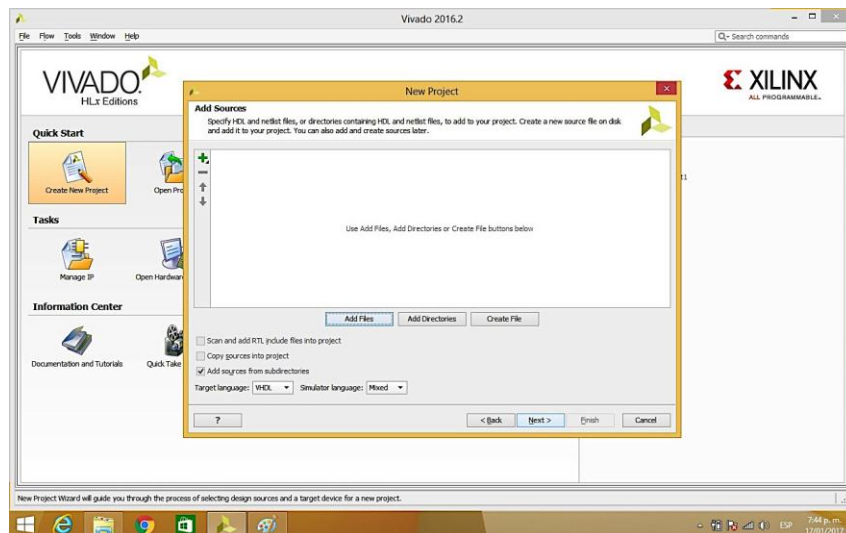
Se escoge la opción RTL Project. Esta opción permite agregar códigos fuente, generación de bloques IP, realizar síntesis de proyectos entre otros. Se continúa la configuración con el botón “Next”



**Figura 5. VIVADO Tipo de proyecto**

### “Add Source”

Esta ventana permite agregar archivos realizados con anterioridad, que pueden servir en el proyecto. Tratándose de un proyecto nuevo existirán varias ventanas a las cuales no se les realiza modificaciones y se continúa con la opción “Next”.

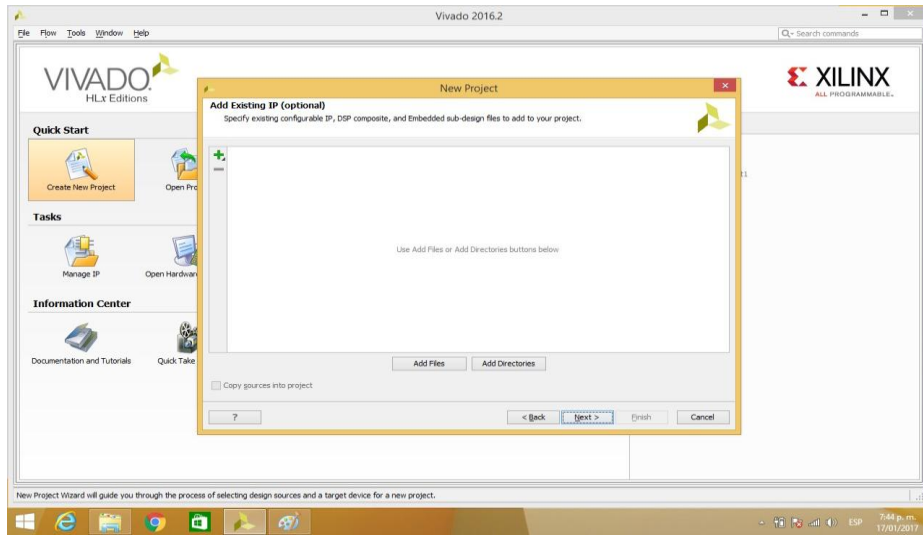


**Figura 6. VIVADO Agregar fuentes al proyecto**

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

**“Add Existing IP”**

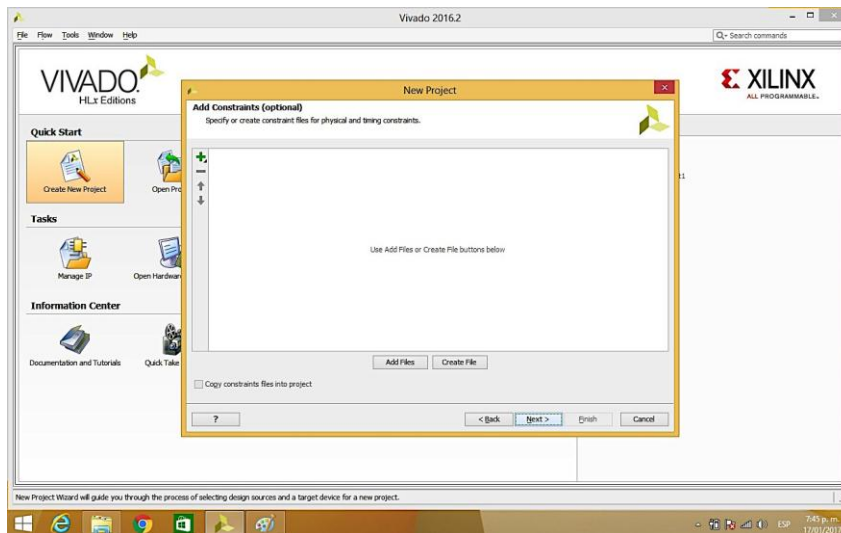
Esta ventana permite agregar archivos IP realizados anteriormente, que pueden servir en el proyecto. Tratándose de un proyecto nuevo existirán varias ventanas a las cuales no se les realiza modificaciones y se continúa con la opción “Next”



**Figura 7. VIVADO Agregar IP existentes**

**“Add Constraints”**

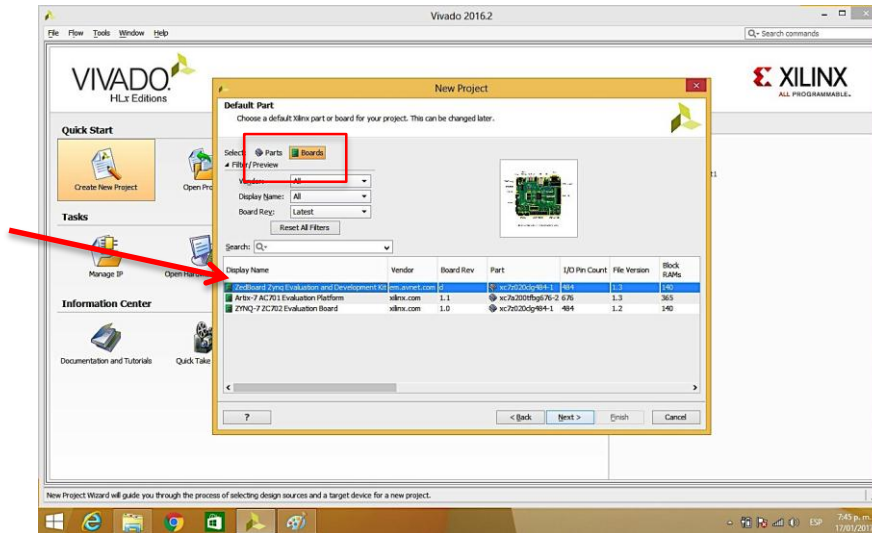
Esta ventana permite agregar archivos ‘constraints’ realizados con anterioridad, que pueden servir en el proyecto. Este tipo de archivos realizan restricciones o pone requisitos que se deben de tener para realizar la compilación y lograr ser ejecutado en la ZedBoard. Tratándose de un proyecto nuevo no se realizan modificaciones y se continúa con la opción “Next”



**Figura 8. VIVADO Agregar constraints existentes**

**“Default Part”**

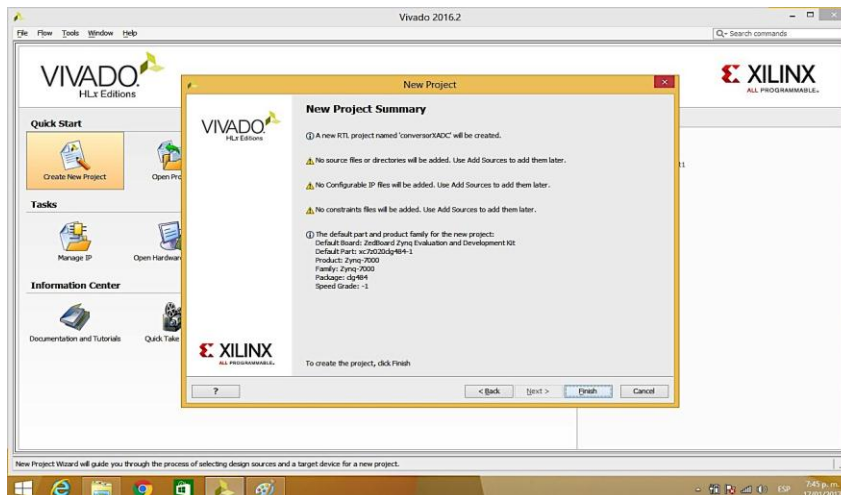
En la ventana de la Figura 9 se elige el tipo de Board o placa de desarrollo a utilizar. El software tiene diferentes algunas referencias predeterminadas. En este caso en la opción “Boards” se escoge la placa “ZedBoard Zynq Evaluation and Development kit”, la cual contiene el módulo XADC. Esta opción puede ser modificada posteriormente si se requiere.



**Figura 9. VIVADO Elección del hardware-ZedBoard**

**“New Project Summary”**

Se muestra un resumen del hardware escogido. Para terminar con este paso se escoge “Finish” y permitir continuar.



**Figura 10. VIVADO Resumen configuración de Hardware**

Una vez realizada la configuración, aparecerá la siguiente ventana (Figura 11), donde se realizará la creación de los archivos necesarios para la implementación. Las herramientas del VIVADO permiten realizar diferentes actividades en el proyecto, en cuanto a archivos y configuraciones que se requiera modificar.

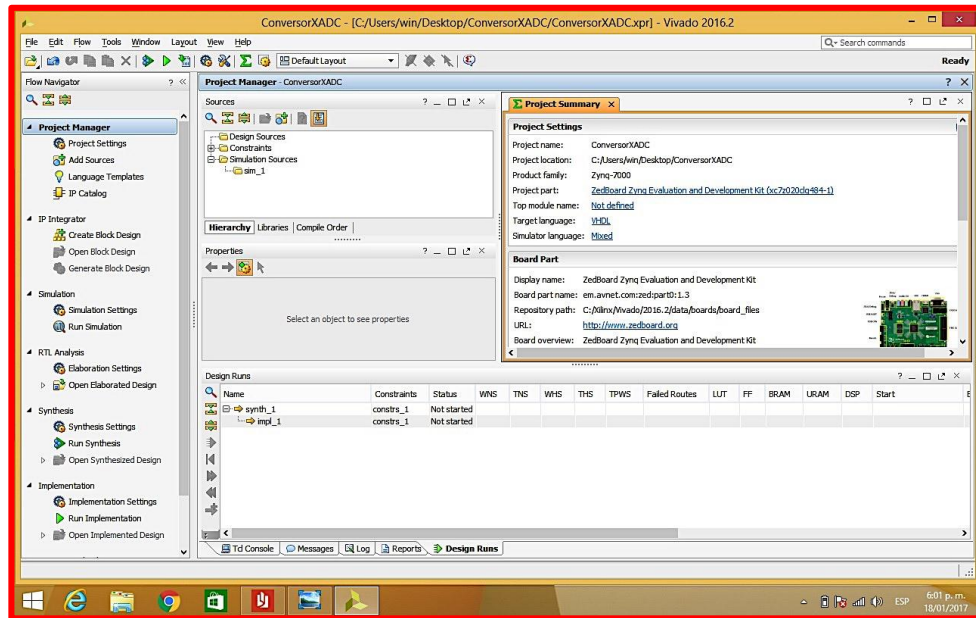
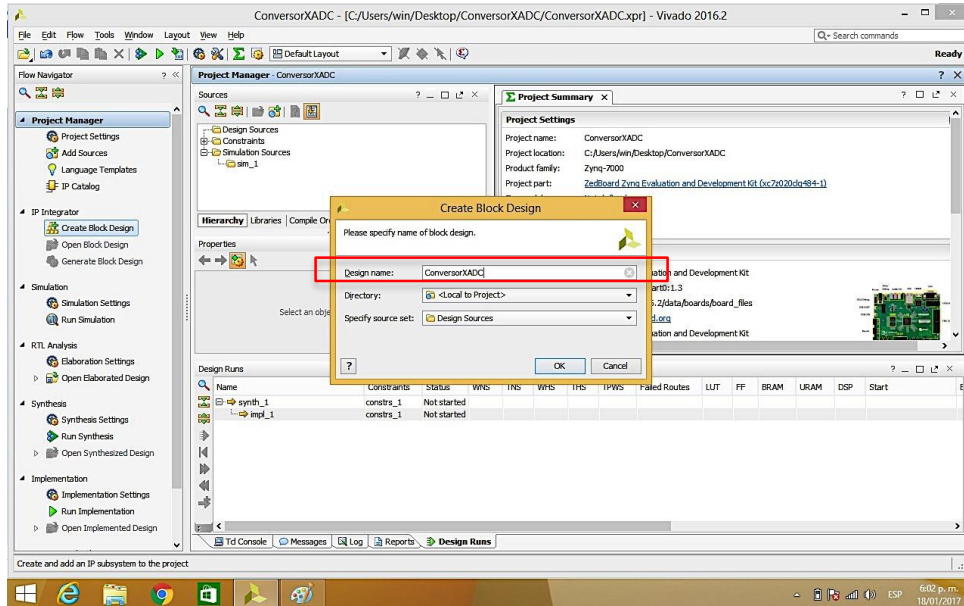


Figura 11. VIVADO Administrador de proyecto

### 3.1.2. Paso 2. Diseño de proyecto en bloques.

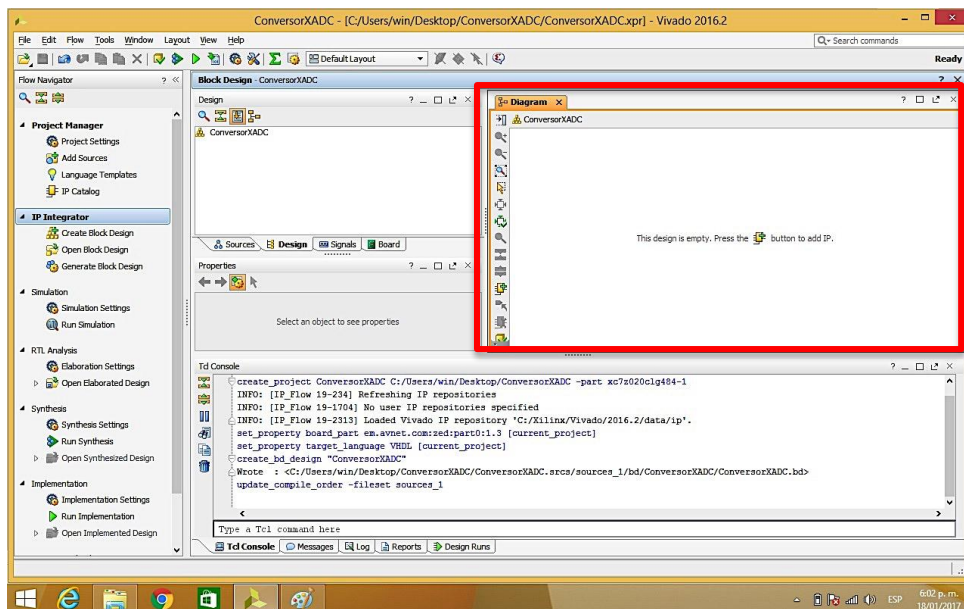
En este paso permite que el proyecto sea realice en lenguaje tipo bloques. Siendo esta una forma fácil, rápida y didáctica ofrecida por VIVADO a diferencia de la configuración por medio de líneas de códigos que sería un poco más tediosa en su realización.

El primer paso es escoger la opción “Create Block Desing”, el cual se encuentra en el integrador de IP, asignándole un nombre de fácil identificación y la carpeta donde se almacenará el archivo.



**Figura 12. VIVADO Creación diseño de bloques**


Creado el archivo aparece la siguiente ventana (Figura 13), mostrando el “Diagrama” o espacio de trabajo, donde se llamarán todos los bloques IP requeridos para el desarrollo.



**Figura 13. VIVADO Espacio Diagram para desarrollo de proyecto**



### 3.1.3. Paso 3. Adicionar bloques IP

Los bloques IP se agregan ubicando el botón  , donde se muestra todos bloques disponibles en el repositorio de VIVADO.

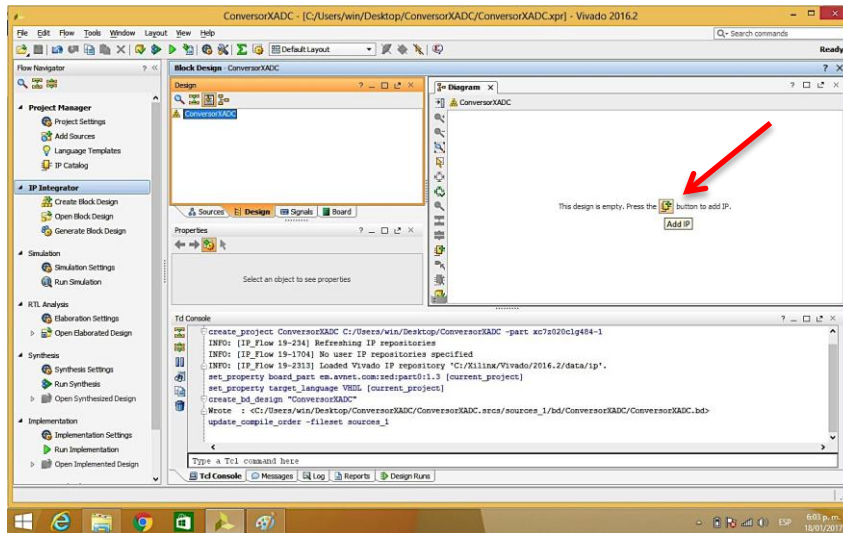


Figura 14. VIVADO Adicionar bloques IP al proyecto

En la subventana emergente (Figura 15), se escribe el nombre del bloque requerido, el bloque correspondiente al XADC aparece con el nombre de “XADC Wizard”. Se selecciona y este bloque se ubicara en el “Diagram”.

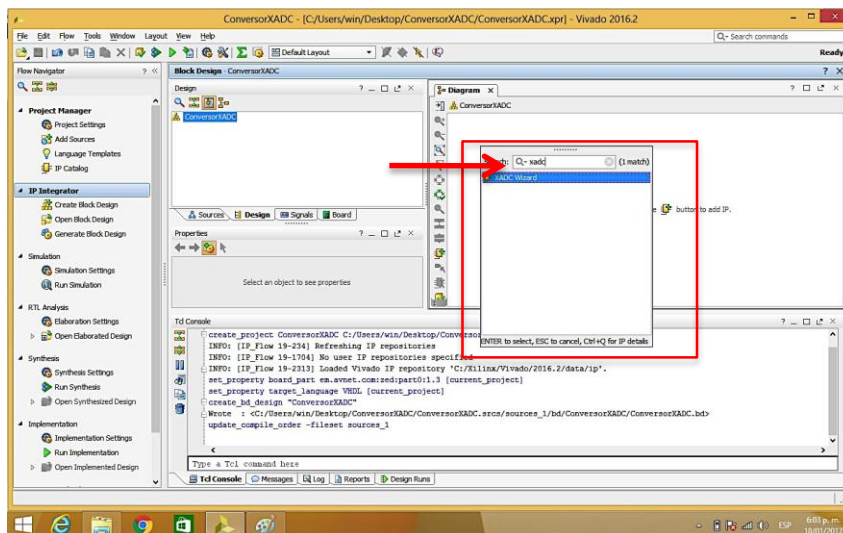


Figura 15. VIVADO Búsqueda bloque IP XADC

El bloque del XADC, se presenta de la forma como aparece en la Figura 16 con sus respectivos pines de conexión, para poderse luego interconectar con otros bloques.

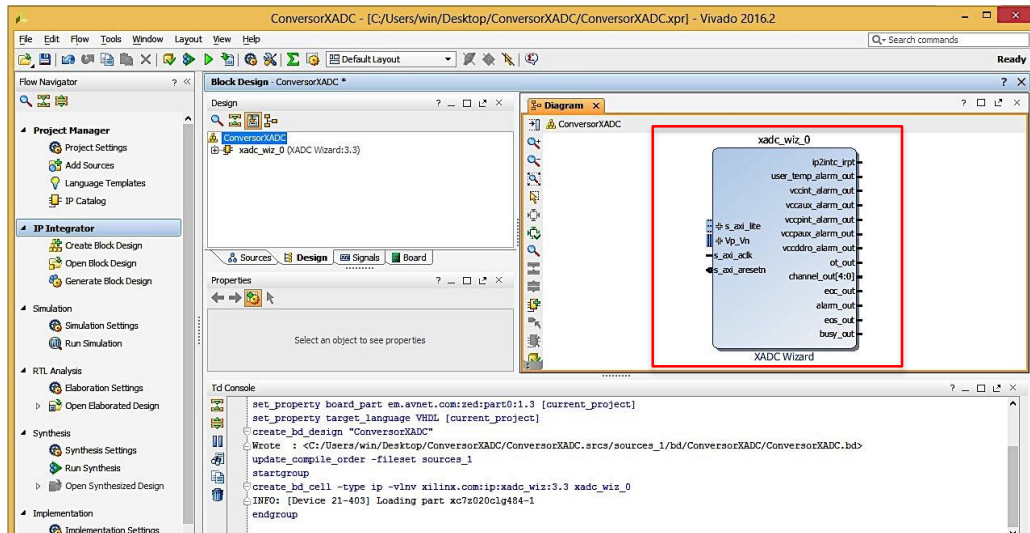


Figura 16. VIVADO Bloque XADC con puertos de conexión

Para agregar un nuevo bloque IP, en el espacio de trabajo “Diagram” de presiona clic derecho y emerge un menú con diferentes opciones, simplemente se debe seleccionar la opción “Add IP...”.

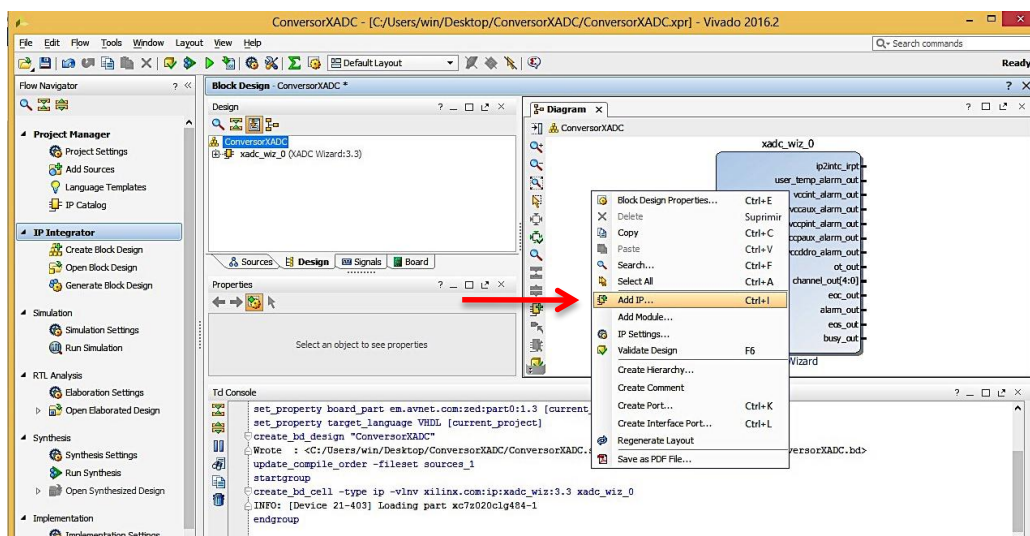
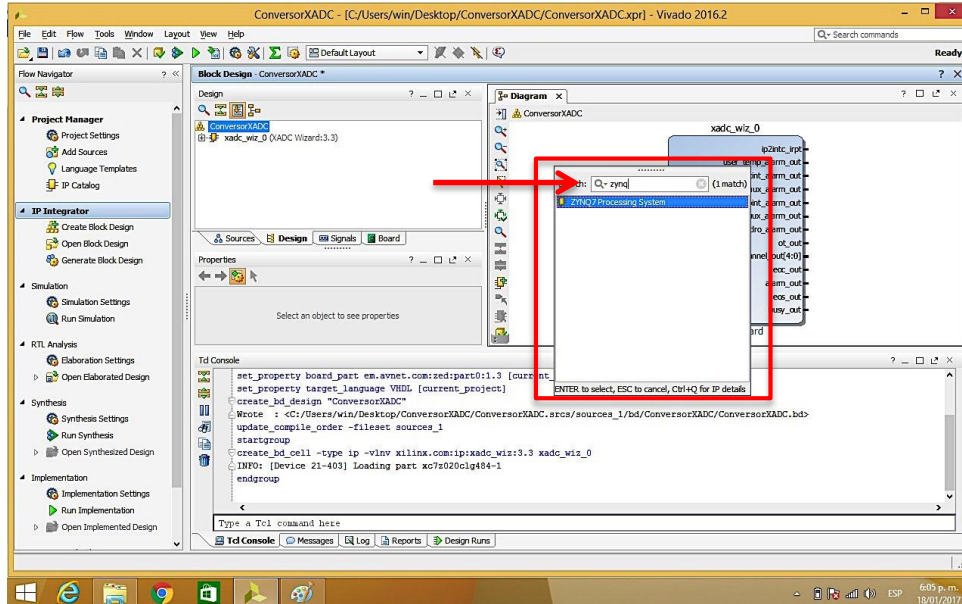


Figura 17. VIVADO Agregar nuevo bloques IP al proyecto

Ahora se busca un bloque llamado “ZYNQ Processing System”, el cual se encarga de integrar los bloques IP.



**Figura 18. VIVADO Agregar bloque ZYNQ Processing System**

Una vez agregado el Bloque de “Processing System”, se agrega el bloque “AXI interconnect” automáticamente por medio de la opción “Run Connection Automation” que aparece en la ventana “Diagram”, estos bloques se auto conectan entre el dispositivo maestro con otros dispositivos esclavos, “Processor System Reset” el cual se encarga del reinicio del sistema del procesador dependiendo los parámetros establecidos por el usuario.



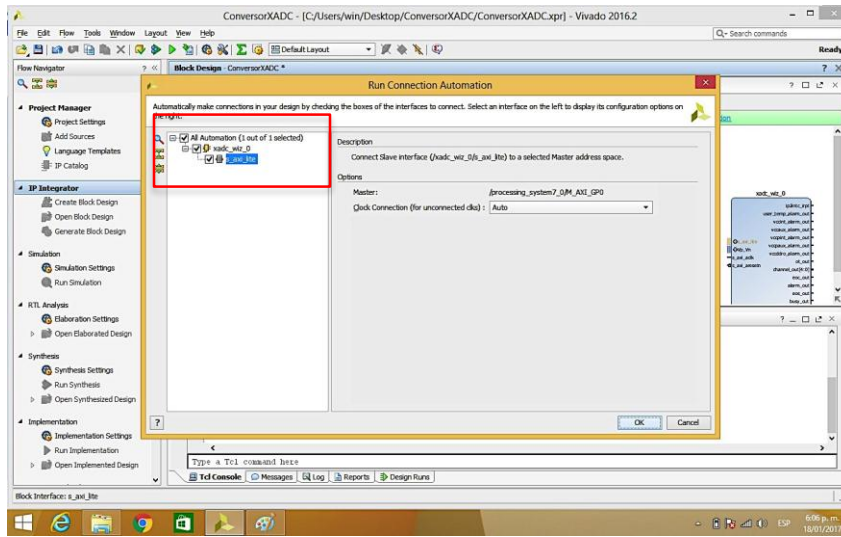


Figura 19. VIVADO Ejecución Run Connection Automation

### 3.1.4. Paso 4. Organización de bloques e interconexión

Los bloques conectados aparecen en una manera desorganizada. Por lo tanto para un mejor orden en los proyectos y una mejor visualización de las conexiones, se busca la opción “Regenerate Layout”. Y automáticamente se cambia de ubicación los bloques y sus conexiones como se observa en la Figura 20 y Figura 21.

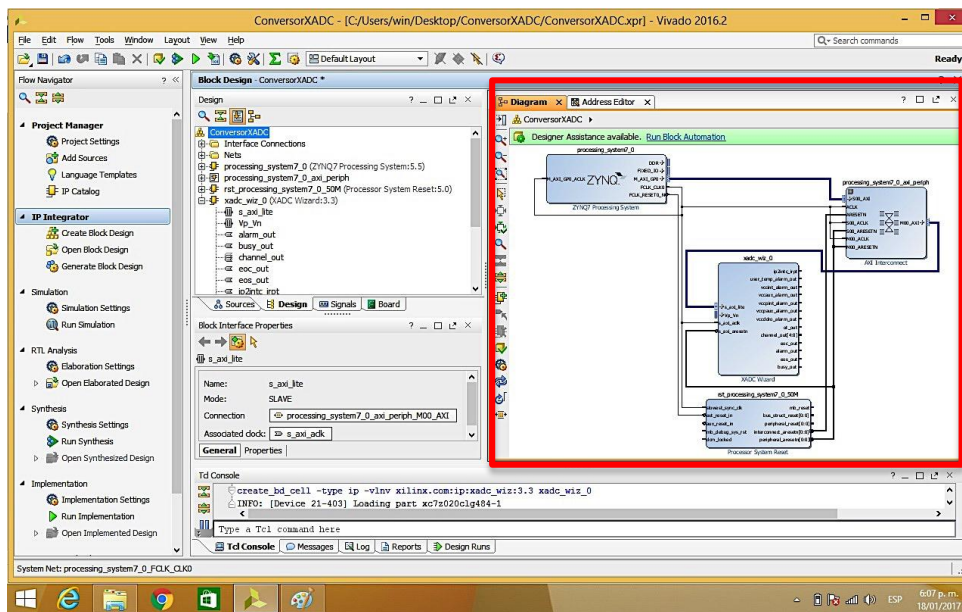


Figura 20. VIVADO Conexión puerto a puerto de los bloques

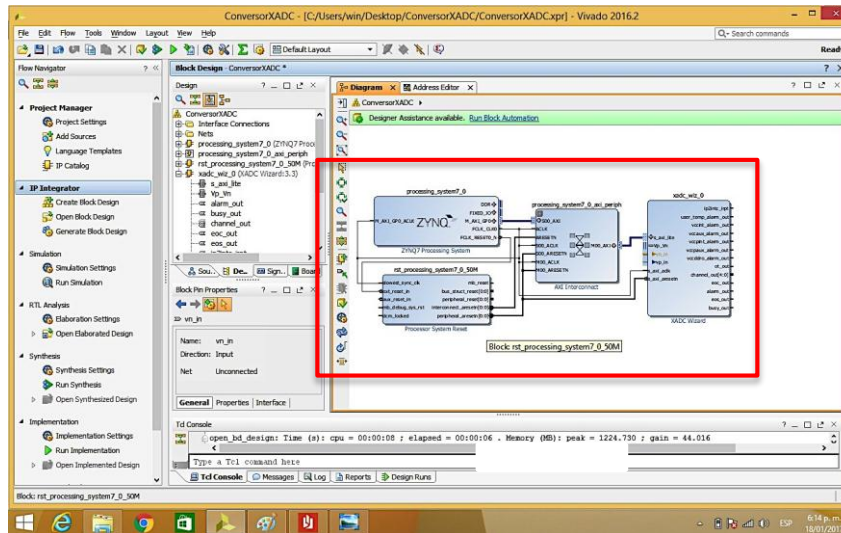


Figura 21. VIVADO Bloques organizados con Regenerate Layout

### 3.1.5. Paso 5. Configuración del bloque XADC

En este paso se realiza la configuración del bloque XADC, donde se agregan o quitan funciones dependiendo de los requerimientos y la complejidad del proyecto. Para nuestro objetivo se trabajara con una configuración básica del módulo de la siguiente manera.

En la etiqueta BASIC, se eligen las opciones que se encuentran en los recuadros (Figura 22), para hacer uso del XADC con varias entradas y a una frecuencia de reloj de 100MHz

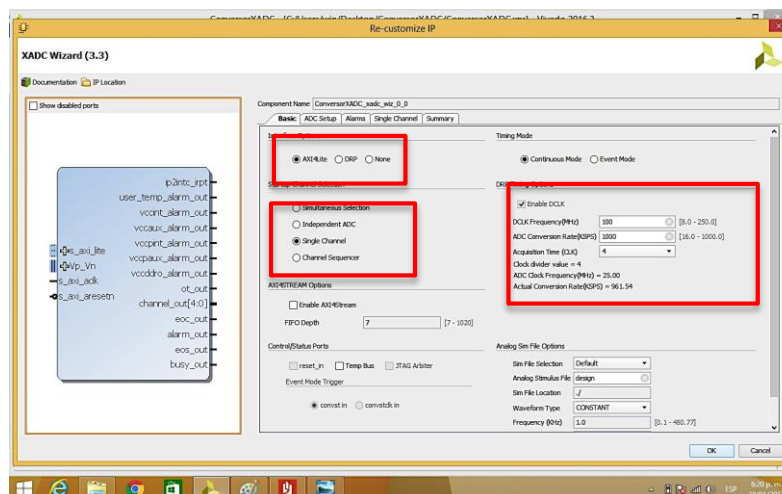


Figura 22. VIVADO Configuración XADC-Basic

En la etiqueta “ADC SETUP”, se seleccionan las opciones que se encuentran en los recuadros de la Figura 23.

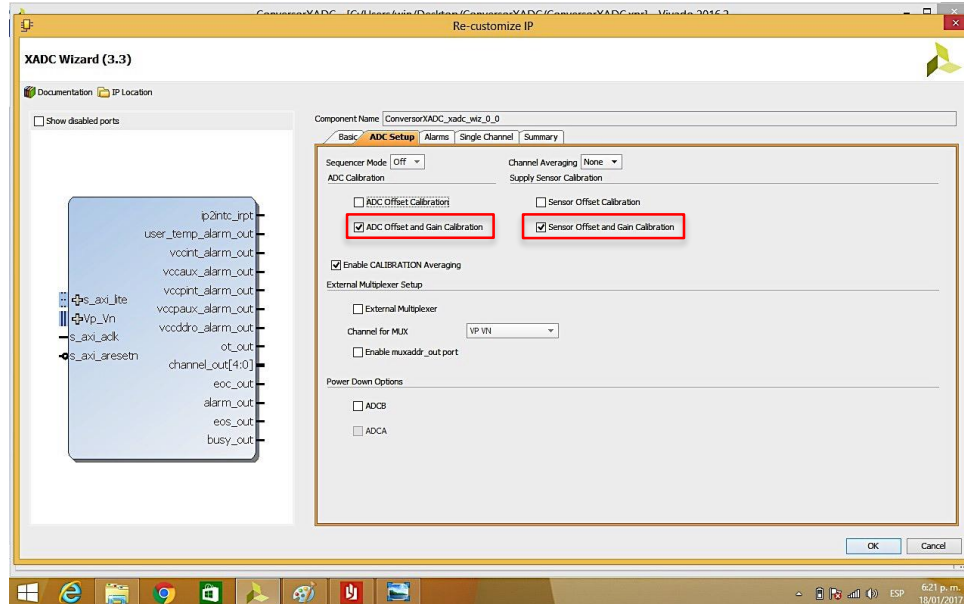


Figura 23. VIVADO Configuración XADC-ADC Setup

En la etiqueta “ALARMS”, en este caso se desactivan todas las alarmas que tenga seleccionadas, dado a que en nuestro proyecto no se utilizaran estas advertencias que harán que el modulo entre en reinicio.

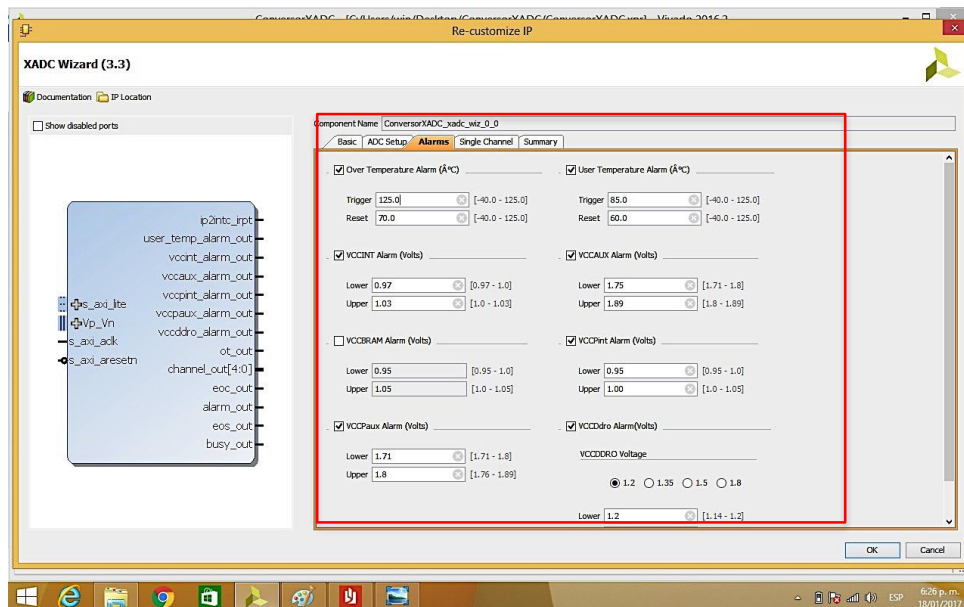


Figura 24. VIVADO Configuración XADC-Alarmas

En la etiqueta “Channel Sequencer” se habilitan o deshabilitan las opciones de entrada que se utilizarán, en este caso se habilitarían las opciones Vp/Vn ya que solo se trabajará en un solo canal.

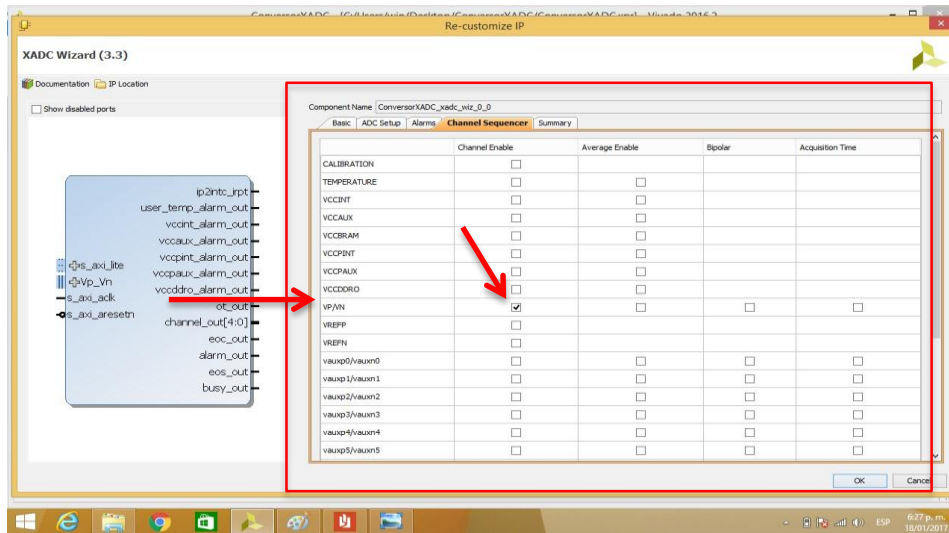


Figura 25. VIVADO Configuración XADC- Channel Sequencer

En la etiqueta “SUMMARY” únicamente se presenta un resumen de toda la configuración realizada.

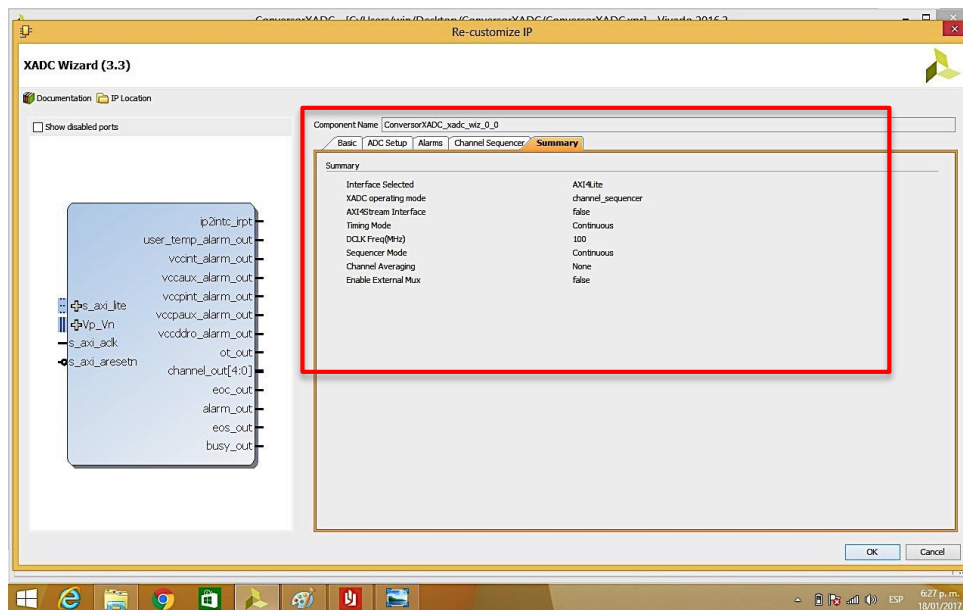


Figura 26. VIVADO Resumen configuración XADC

Una vez realizada la configuración del XADC, el siguiente paso es crear los puertos para cada una de las entradas, en la Figura 27. Se muestra como agregar los puertos.

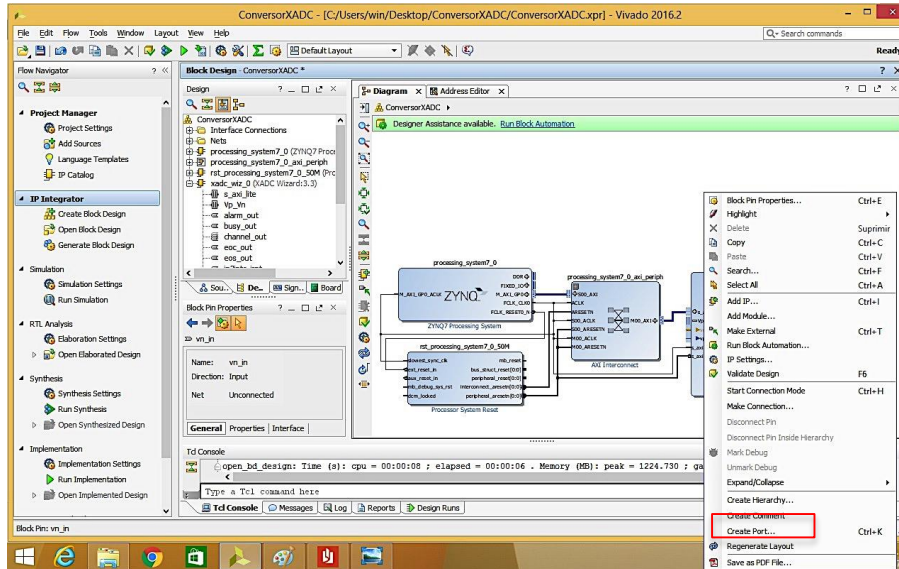


Figura 27. VIVADO Creación de puertos

Se ubica los puertos individualmente, dando clic derecho en cada uno de los puertos, Vp\_in, Vn\_in se elige la opción “Create ports”.

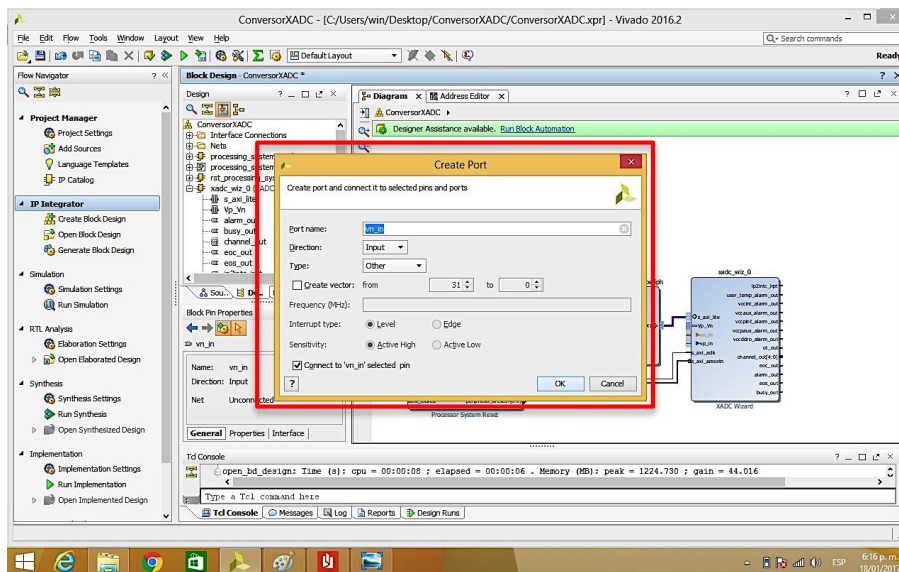


Figura 28. VIVADO Nombramiento de los puertos



Se asigna un nombre para una fácil identificación y la dirección ( Input or Output). Luego se repite la opción de “Regenerate Layout”, para reorganizar los nuevos objetos.

Se confirma la dirección que tiene el XADC en la etiqueta “Address Editor”— 0x43C0\_0000 y 0x43C0\_FFFF.

### 3.1.6. Paso 6. Generación de HDL Wrapper y Constraints

En este paso se mostrará la creación del HDL Wrapper, el cual crea un archivo VHD el cual se asigna como módulo a “top module” con el fin de lograr la síntesis e implementación. Para hacer este procedimiento se debe ubicar en el nombre del diseño de bloques, desplegar las opciones con clic derecho y escoger “Create HDL Wrapper”.

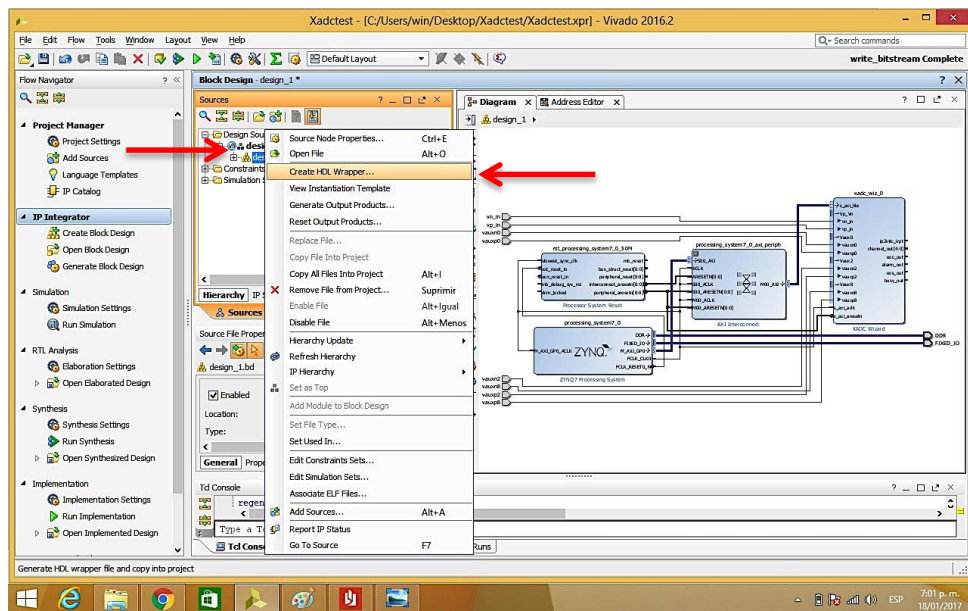


Figura 29. VIVADO Creación HDL Wrapper

Entonces este se volverá un módulo de mayor jerarquía en el diseño, de la siguiente manera, indicando que el Wrapper se creó correctamente.

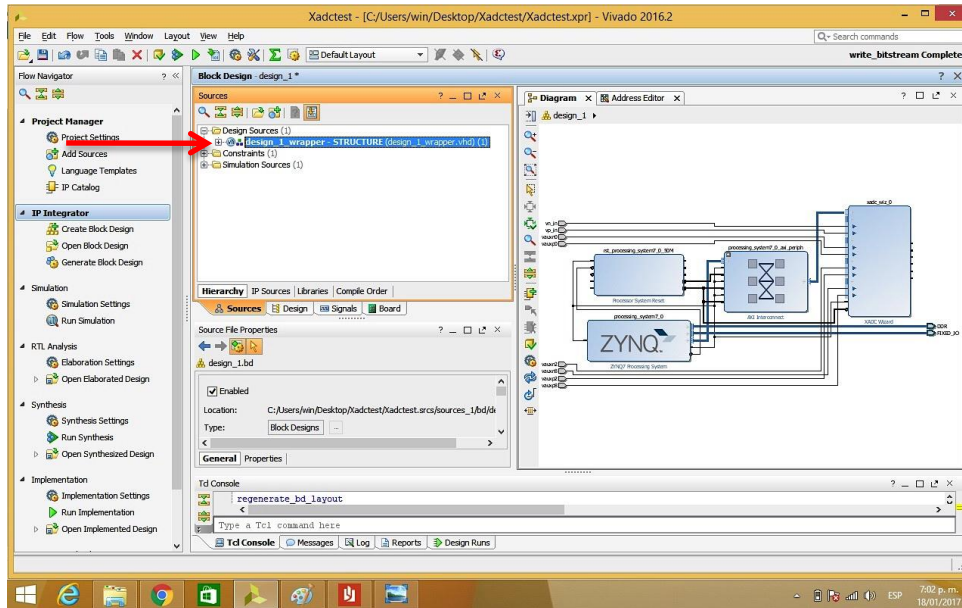


Figura 30. VIVADO Carpeta HDL Wrapper

Para las “constraints”, se da clic derecho en la carpeta raíz agregando una fuente “Add Source” o para más facilidad con las teclas Alt+A.

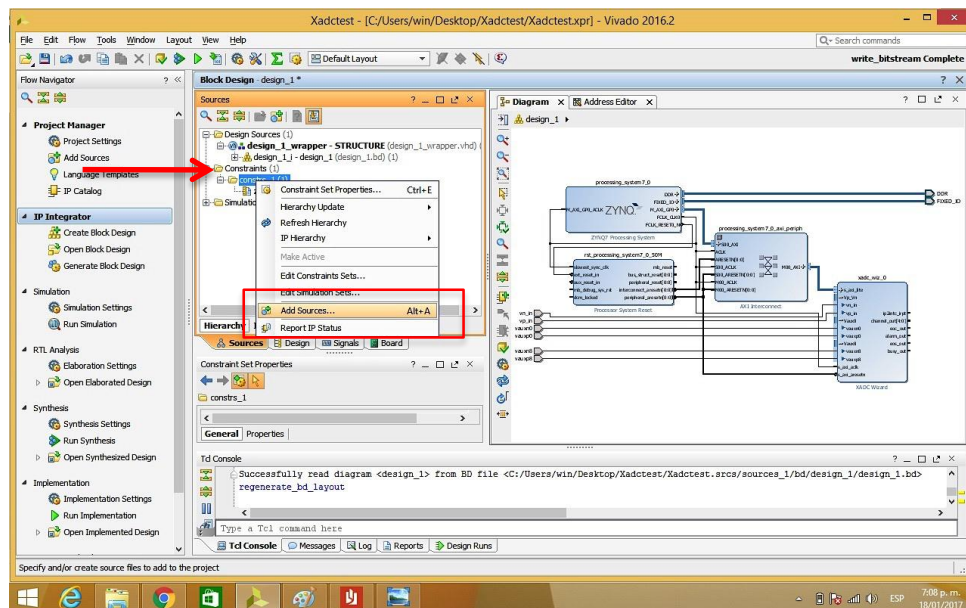


Figura 31. VIVADO Agregar Constraints

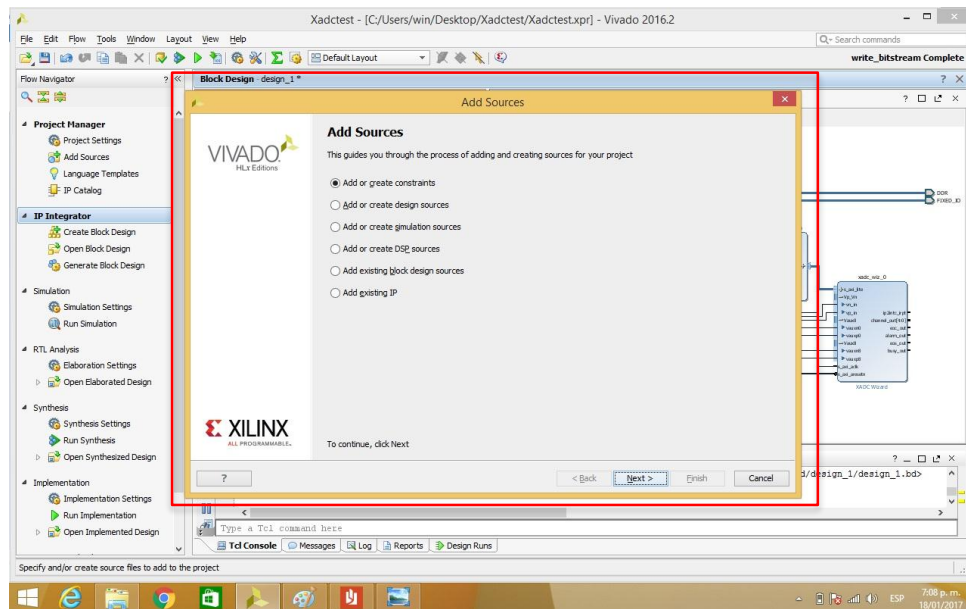


Figura 32. VIVADO Agregar constraints - Add Sources

Se elige el nombre de tal manera que sea fácil su identificación en los archivos del proyecto.

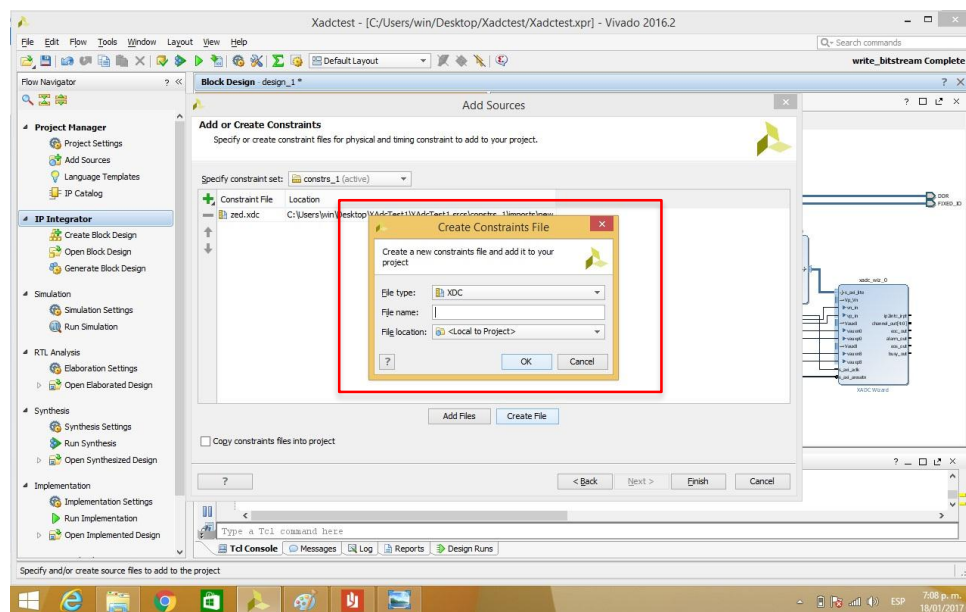


Figura 33. VIVADO Nombre Constraints



Una vez se crea el archivo “Constraints”, en la ventana del archivo Zed.xdc, se van a definir los puertos asignados a las entradas que se están utilizando. Como solo se utilizará las entradas Vp y Vn, los puertos se habilitan quitando el símbolo “#” a las líneas de código que lo tienen, como se observa en Figura 34. El código se almacenara como anexo.

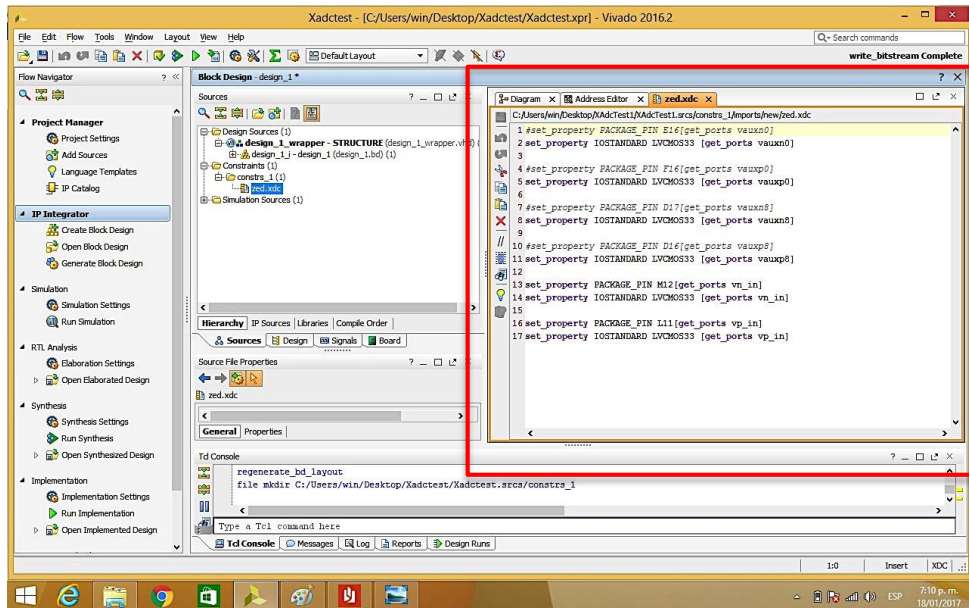
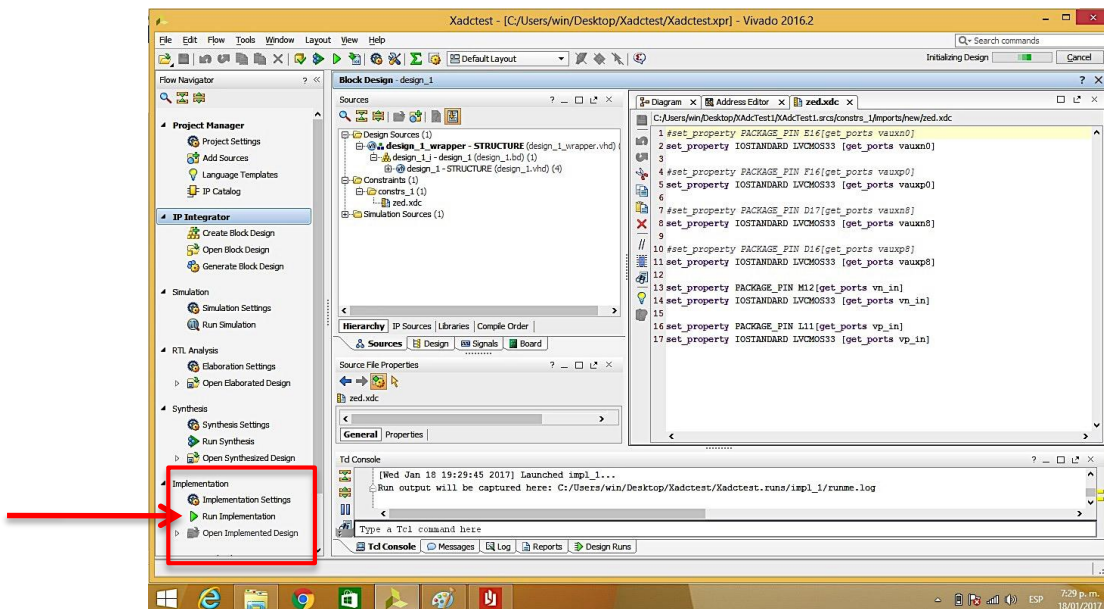


Figura 34. VIVADO Código Constraints

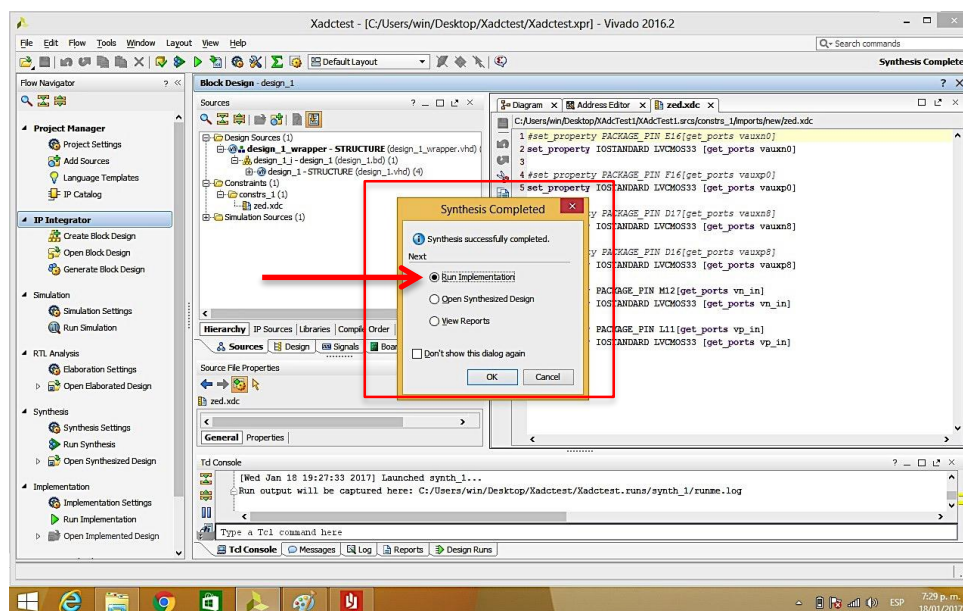
### 3.1.7. Paso 7. Síntesis y Generación Bitstream

Para la síntesis del proyecto en la sección “Flow Navigator”, etiqueta “Synthesis” se escoge la opción “Run Synthesis”, estos pasos se hacen con el fin de poder generar el Archivo Bitstream para luego poder exportar el archivo a SDK.



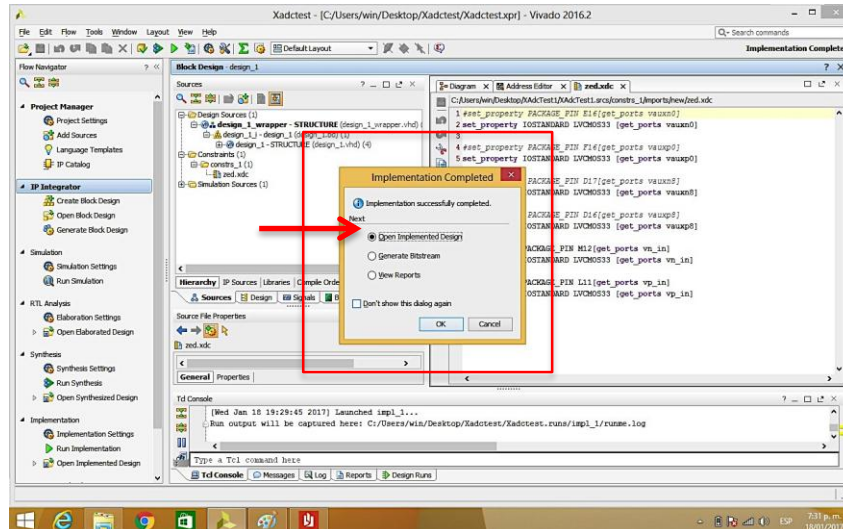
**Figura 35. VIVADO Run Synthesis**

Una vez finalizada la síntesis emerge una ventana como en la Figura 36. confirmando la correcta ejecución y realizar la implementación.



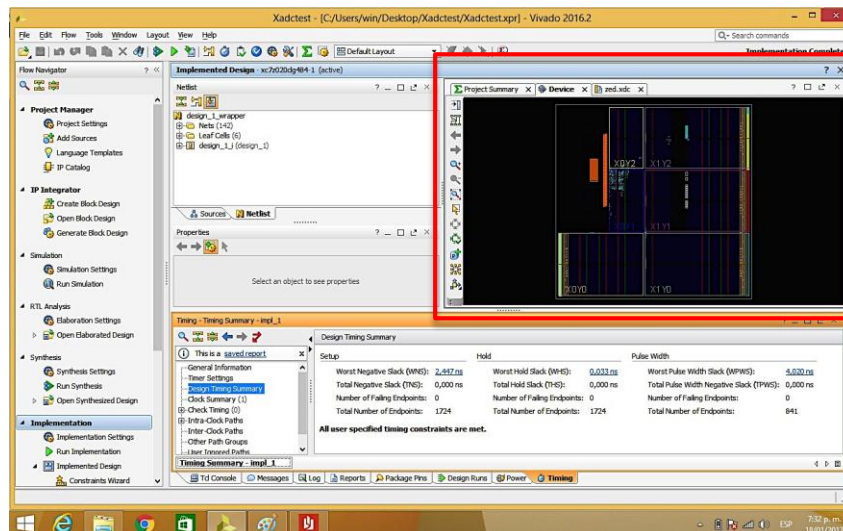
**Figura 36. VIVADO Synthesis completa**

Una vez completa la implementación, emerge una ventana como en la Figura 37. confirmando la ejecución y permitiendo la apertura del diseño implementado.



**Figura 37. VIVADO Abrir implementación**

Cuando se abre el diseño, se puede visualizar las divisiones de la tarjeta y el esquemático.



**Figura 38. VIVADO Distribución ZedBoard**

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para abrir el esquemático del proyecto que se ha realizado, se presiona clic derecho sobre la imagen de la tarjeta, seleccionando “Schematic” como se muestra en la Figura 39. y Figura 40.

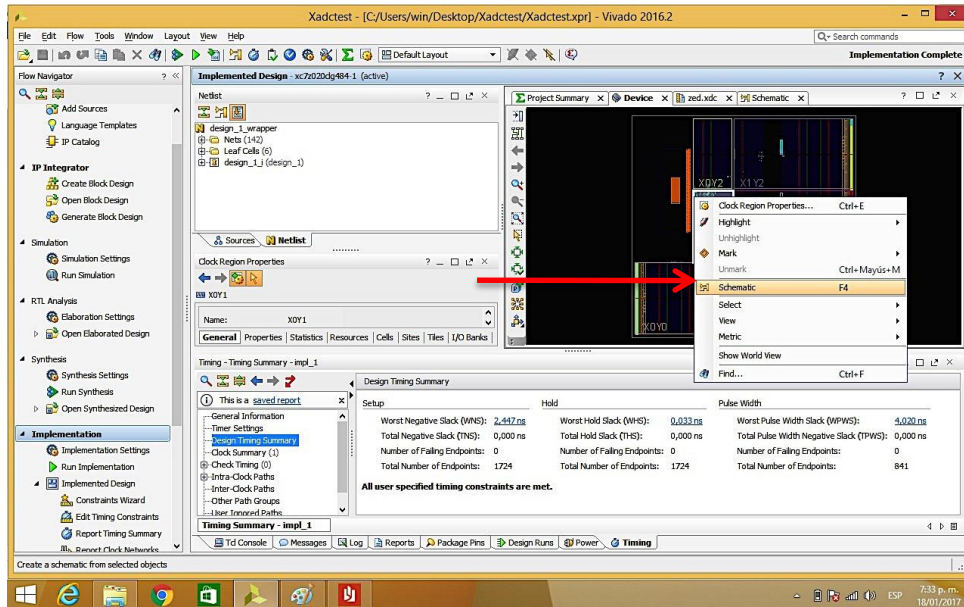


Figura 39. VIVADO Como abrir esquemático

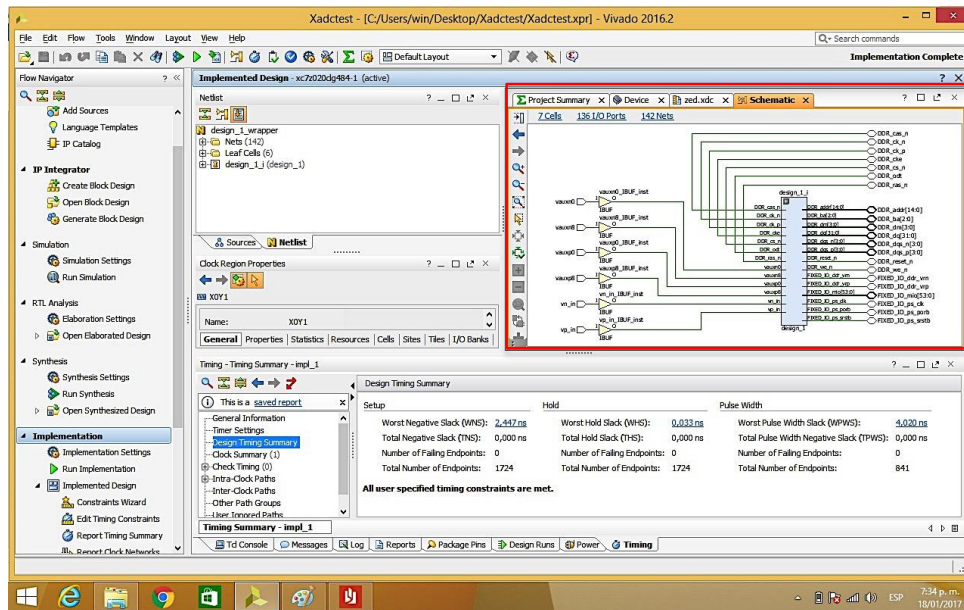


Figura 40. VIVADO Esquemático proyecto



Para la generación del fichero “Bitstream”, solo se ubica la opción “Generate Bitstream”

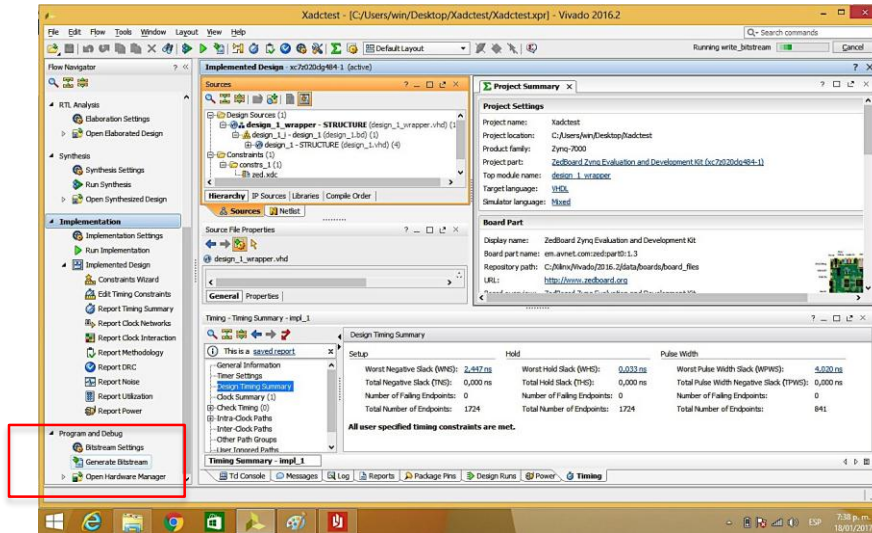


Figura 41. VIVADO Generación Bitstream

### 3.1.8. Paso 8. Exportar proyecto y lanzar a SDK

Para exportar el diseño a SDK se selecciona en *File>Export>Export Hardware* y en la ventana que emergente asegurarse que la opción "Include bitstream" se encuentre activada.

Por último se selecciona en file > Launch SDK y abre el programa de SDK.

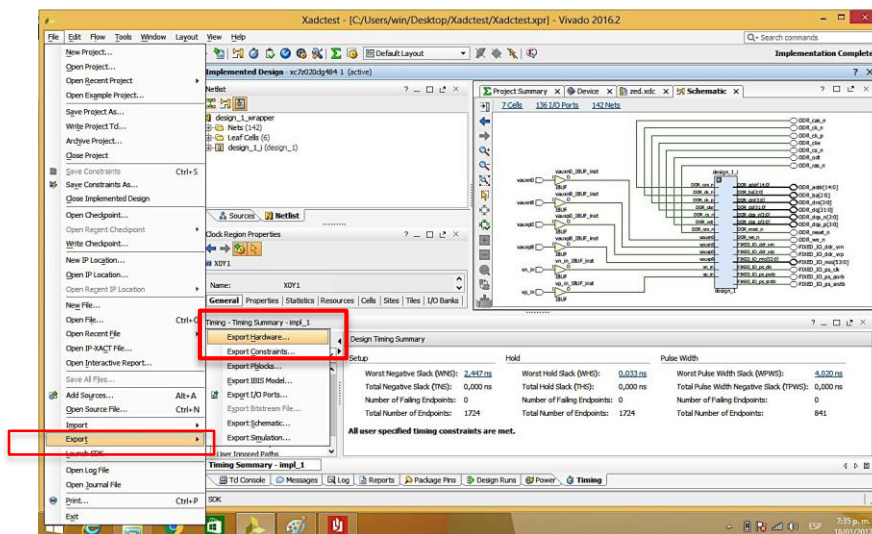


Figura 42. VIVADO Exportar Hardware

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



Figura 43. Software Development Kit SDK

### 3.1.9. Paso 9. Creación Application Project, Creación de algoritmo

#### SDK (Software Development Kit)

El objetivo de lanzar el proyecto a SDK, es poder realizar la conversión de la señal mediante operaciones definidas por líneas de código, que finalmente por medio de la ventana llamada “Terminal” permite observar en tiempo real el valor de los datos que se están leyendo mediante el módulo XADC.

Para crear el programa en SDK, el primer paso es seguir la ruta *File > New > Application Project*, esto con el fin de crear el espacio de trabajo para el código logrando poder interactuar con el hardware XADC y realizar la adquisición de los datos. En las Figura 44 y Figura 45 se muestran los pasos.

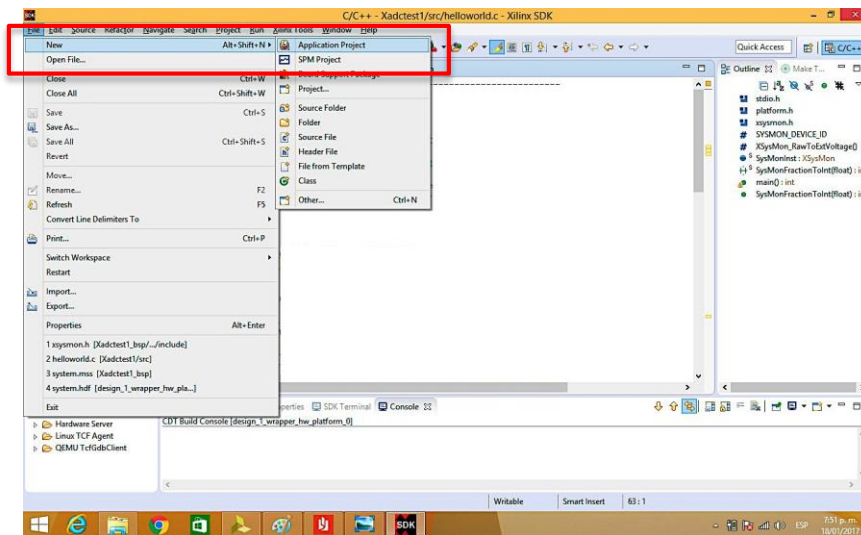
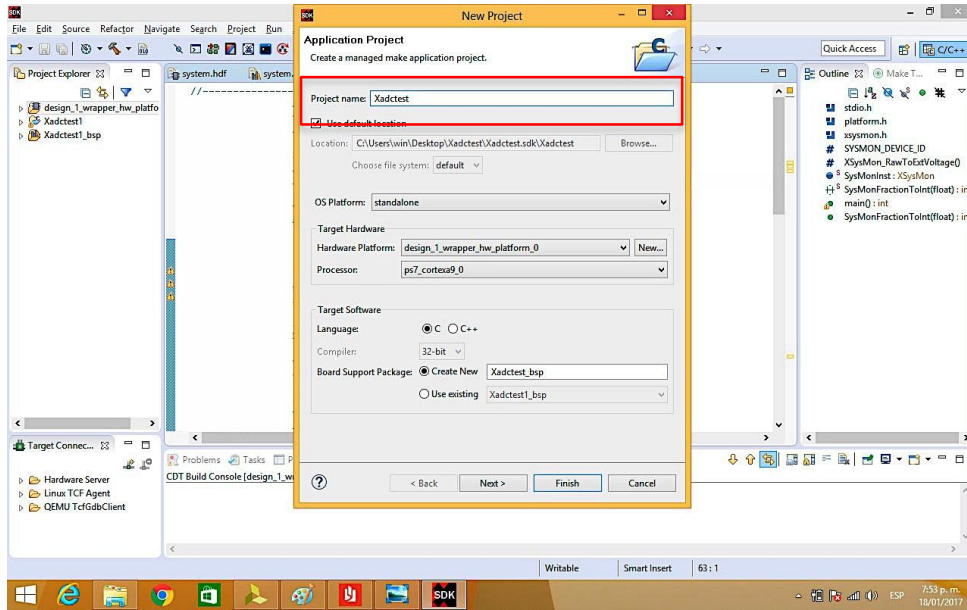
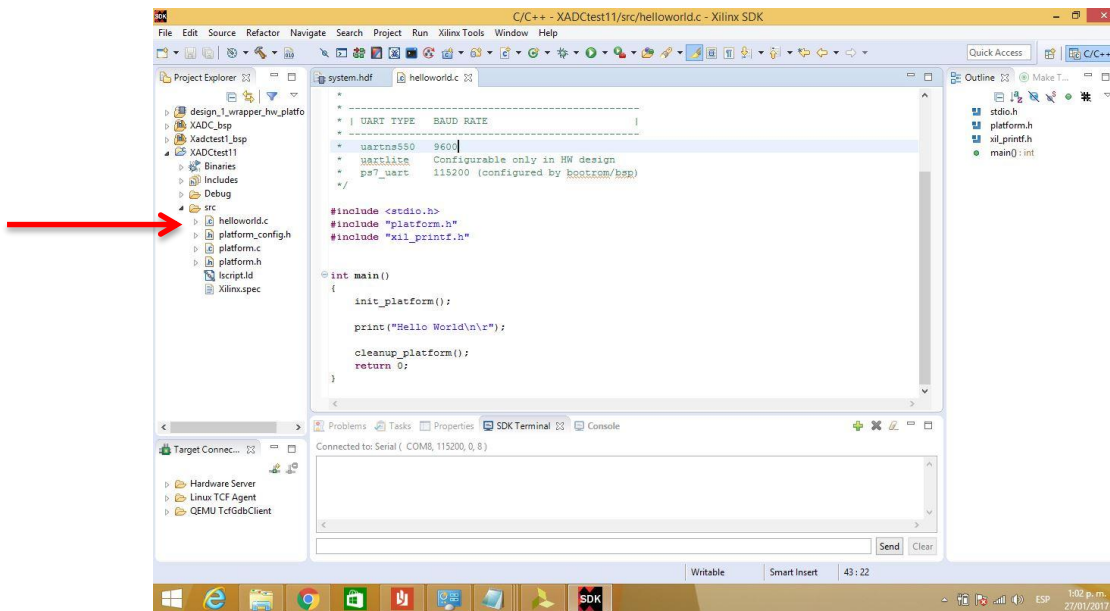


Figura 44. SDK Creación Application Project




**Figura 45. SDK Nombre Application Project**

Una vez creado el “Application Project”, se busca el código fuente “helloworld.c” para realizar una prueba de conexión con el hardware utilizado.



**Figura 46. SDK Agregar helloworld.c**

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para la prueba de conexión una vez compilado el código, en la etiqueta “SDK Terminal” se establece el puerto, modificándolo con la etiqueta . En la Figura 47. Se muestra el proceso.

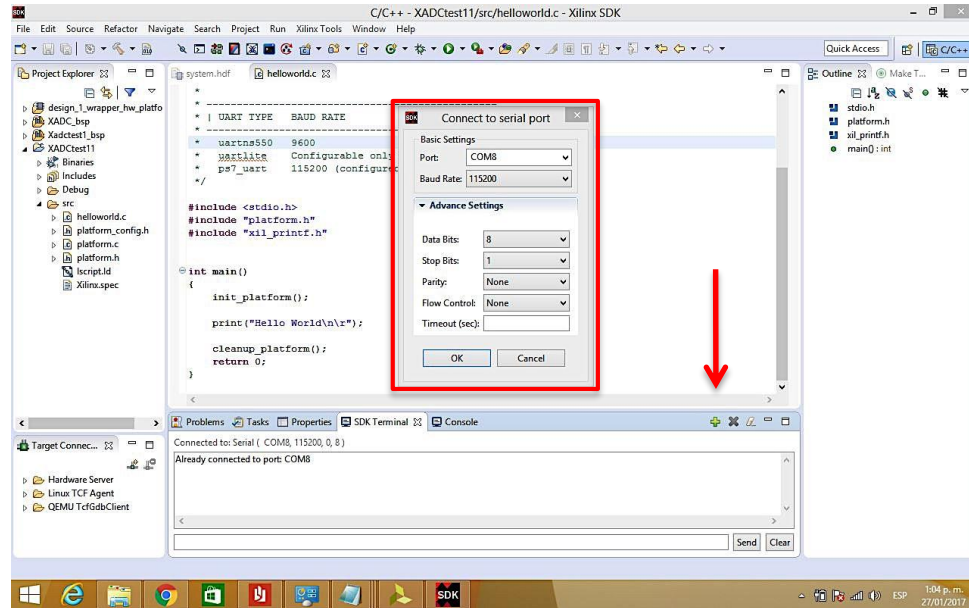



Figura 47. SDK Conexión ZedBoard-PC

La conexión se confirma con el mensaje “*Already connected to port: COM#*”. Este puerto se puede conocer mediante del panel de control en el PC, luego se programa la ZedBoard mediante el botón  (*Program FPGA*). En la Tarjeta se enciende un LED azul confirmando la descarga. Para la ejecución del código y la visualización de los datos en la ventana “SDK terminal” se debe tener seleccionada las opciones como se muestra en la Figura 48.



 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

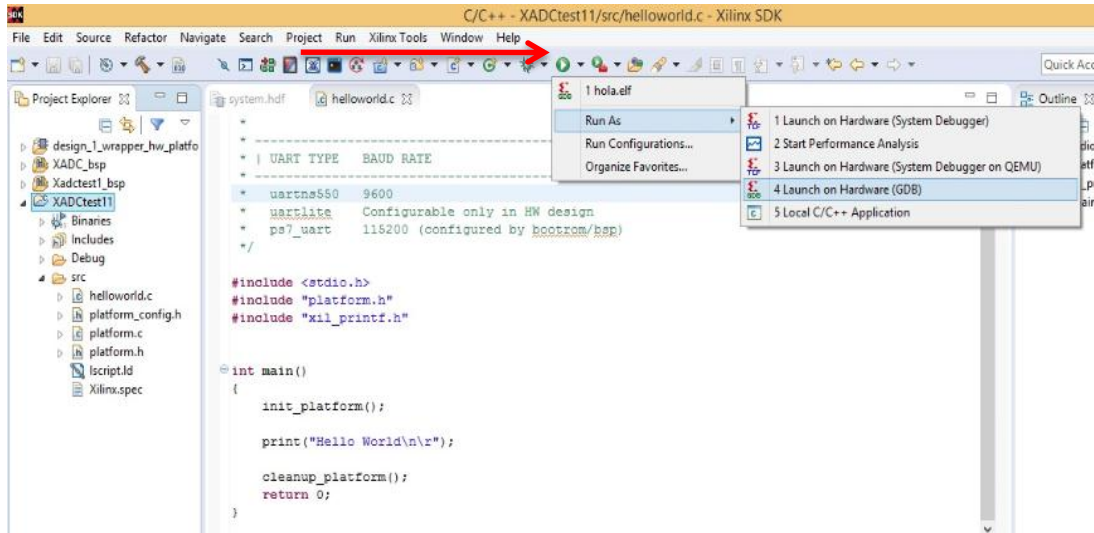


Figura 48. SDK Run helloworld.c

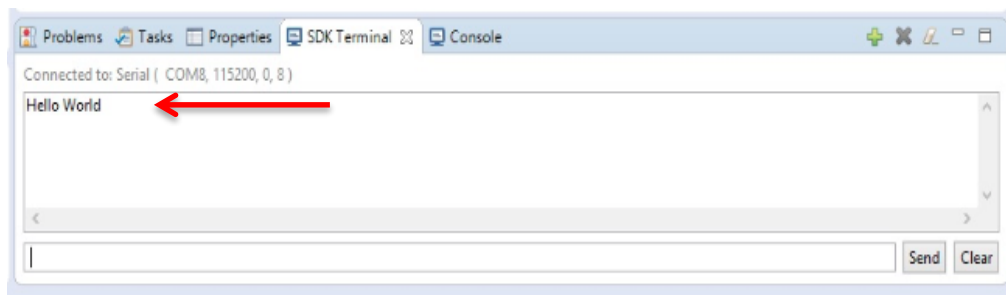


Figura 49. SDK Terminal helloworld

Finalizada la ejecución del archivo “helloworld.c”, se debe modificar por el código para realizar la adquisición. Una buena práctica para el proyecto será renombrar el fichero *helloworld.c* con la opción *Rename*. Es importante que en cada cambio realizado el proyecto sea guardado nuevamente, así en el momento de reprogramar la ZedBoard los cambios se transfieran.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

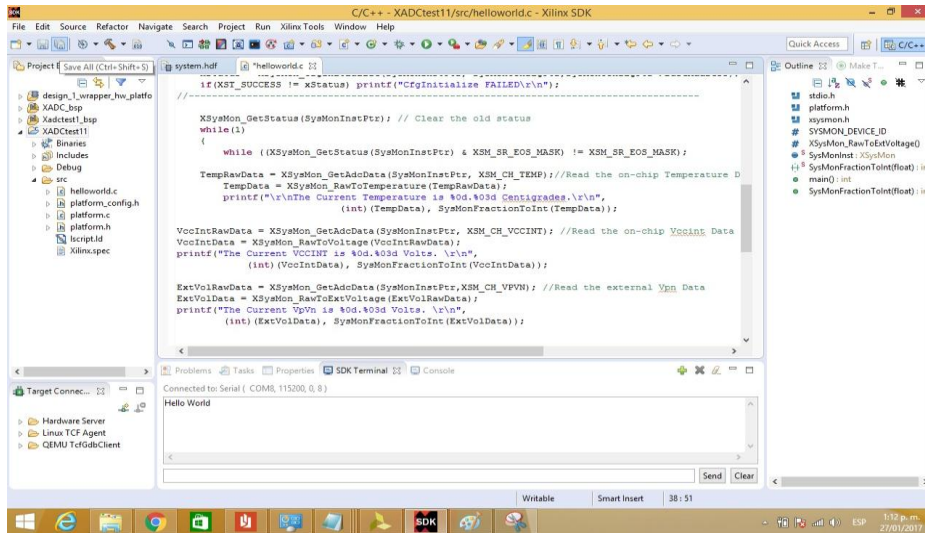


Figura 50. SDK Código Conversión.

Se realizan los mismos pasos en la ejecución de helloworld.c. *Programar ZedBoard y Launch on Hardware.*

La ventana terminal en el SDK permite la visualización de los resultados obtenidos como se muestra en la siguiente sección.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 4. Resultados

Finalizados los anteriores pasos se conectan físicamente los pines VP\_VN del “XADC header” por donde la señal ingresará. Se realizan 3 pruebas variando el voltaje entregado por la fuente de corriente continua y se comprueba el valor con la medición visualizada en el display del multímetro. En la etiqueta “SDK terminal” del programa SDK se observa el siguiente comportamiento.

La primera prueba se realiza con 0.354V entregados por la fuente.

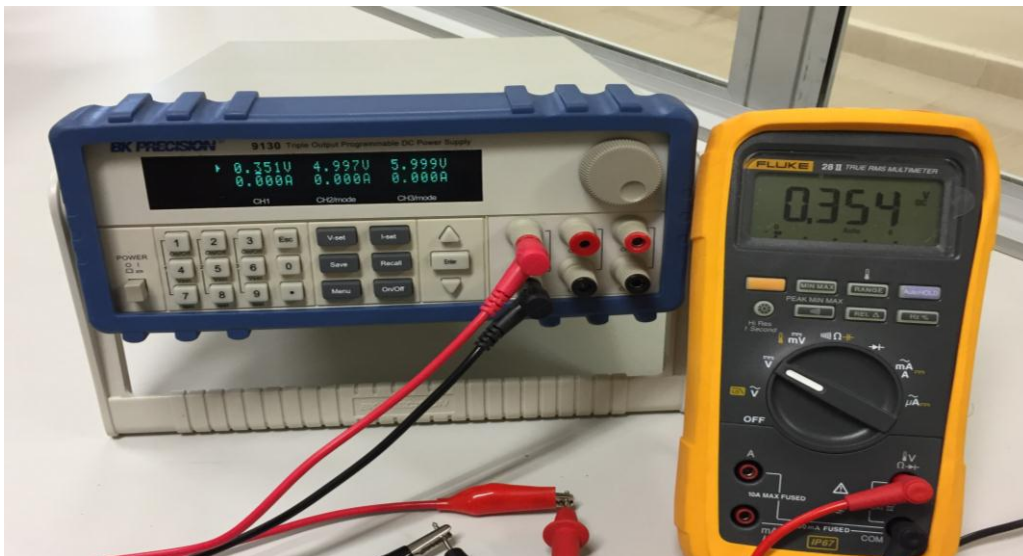


Figura 51. Voltaje entrada Vp/Vn 0,354v

```

Problems Tasks Properties SDK Terminal Console
Connected to: Serial ( COM8, 115200, 0, 8 )

PRUEBA EN ZEDBOARD MODULO XADC:
El Voltaje en Vp y Vn es 0.356 Voltios.

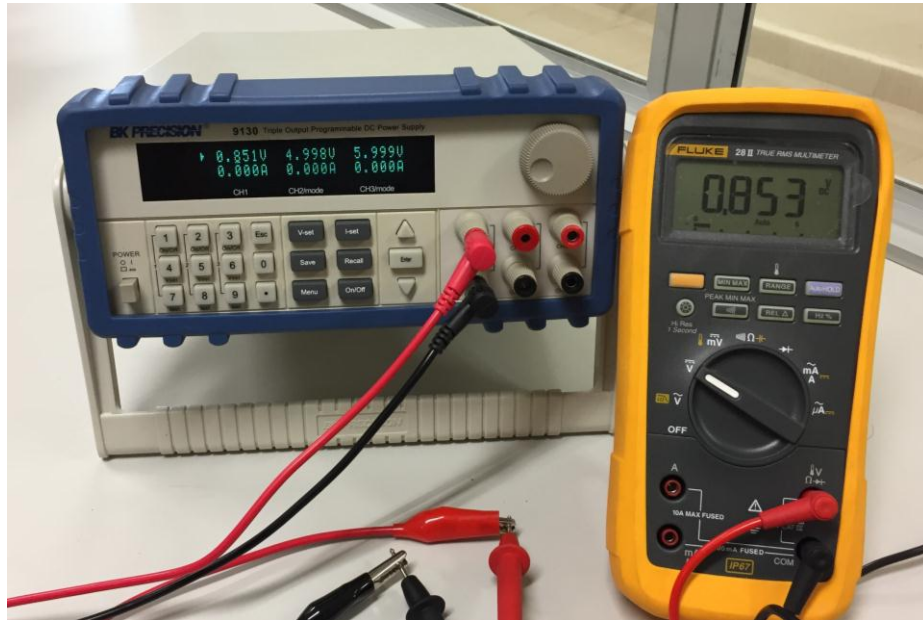
PRUEBA EN ZEDBOARD MODULO XADC:
El Voltaje en Vp y Vn es 0.356 Voltios.

```

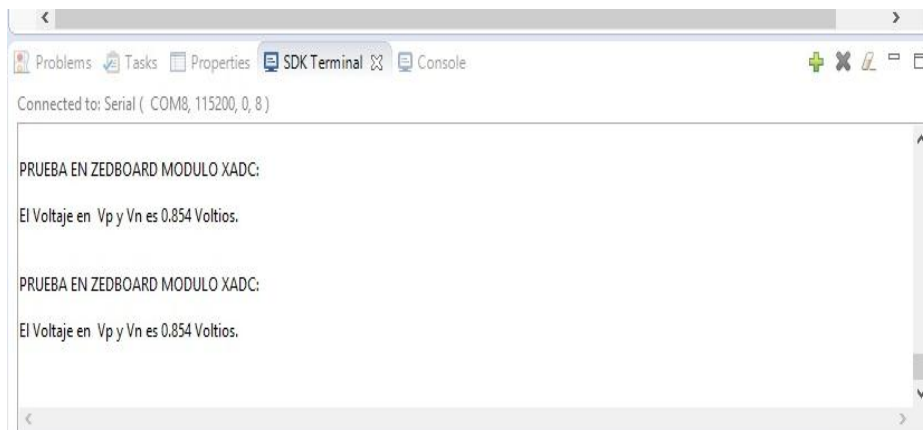
Figura 52. SDK Terminal Vp/Vn 0,354V

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La segunda prueba se realiza con un voltaje de 0.853v entregados por la fuente.



**Figura 53. Voltaje de entrada Vp/Vn 0,853V**



**Figura 54. SDK Terminal Vp/Vn 0,853v**

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La tercera prueba se realiza con 1.055v entregados por la fuente.

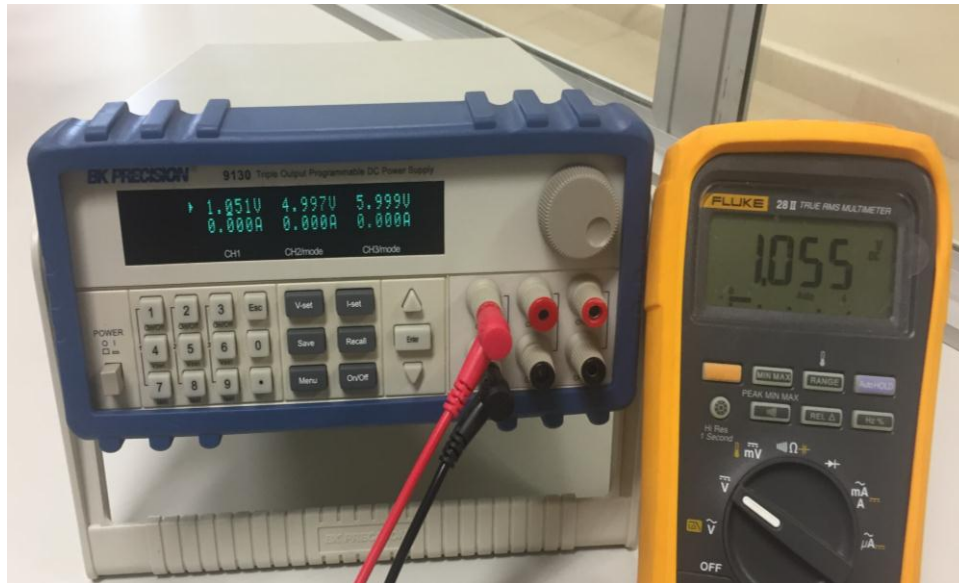


Figura 55. Voltaje de entrada Vp/Vn 1,055v

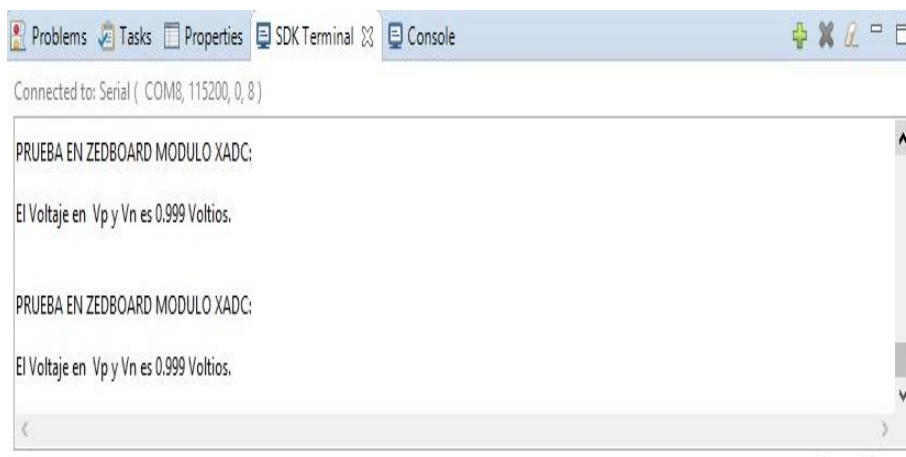


Figura 56. SDK Terminal Vp/Vn 0,999v

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 5. Conclusiones, recomendaciones y trabajo futuro

El modulo que se conecta en el “XADC header”, es un componente adicional que no se encuentra incluido en la ZedBoard. Por lo tanto, la implementación del trabajo práctico en laboratorio, se desarrolló con la conexión directa de los pines programados para la adquisición de las señales análogas. Esto trajo consigo realizar un estudio previo de los manuales ofrecidos por el fabricante, posibilitando la implementación del código sin el modulo físico. Este tipo de implementación requiere una buena concentración en el momento de cablear las referencias para voltajes positivos, ya que podría existir un cortocircuito, dañando el fusible principal de la tarjeta.

Las pruebas presentan una consistencia entre los valores entregados por la fuente, el valor medido en el Multímetro y el valor mostrado en la ventana de visualización del SDK, confirmando una correcta codificación y configuración de la tarjeta.

Para una implementación del XADC con sensores externos, se debe realizar primero un escalamiento del voltaje entregado por los sensores, ya que el módulo tiene un rango de trabajo de 0v hasta 1v, por lo tanto si no se realiza este procedimiento se pueden, generar daños sobre el modulo.

El resultado obtenido en la prueba número 3, muestran el máximo voltaje de lectura en el canal Vp/Vn, por lo cual se debe tener bastante cuidado en este tipo de Voltajes de entrada, por lo tanto, en caso alguno de que el voltaje pueda tener valores superiores, se debe agregar un circuito externo limitador de voltaje, posiblemente con diodos zener o integrados reguladores de tensión.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 6. Referencias

Castrejón, J. (2013). *Evaluación y desarrollo de módulo hardware en SoC programable* . Obtenido de Universidad Carlos III de Madrid : <https://core.ac.uk/download/pdf/29404841.pdf>

Ling, C., & Norum, L. (2016). Utilization of Xilinx FPGA built in Analog to Digital Converter for data acquisition in power electronics converter. *Journal of Electrical Engineering* , 16 (3), 1-6.

Ortega, S. (2014). *Implementación de periféricos en vivado para dispositivos ZYNQ*. Obtenido de Universidad de Alcalá Escuela Politécnica Superior : <http://hdl.handle.net/10017/20821>

Versen, M., Kipfelsberger, S., & Sökmen, F. (2016). Model-Based Reference Design Projects with MathWorks'HDL Workflow Advisor for Custom-Specific Electronics with the Zedboard. 104-107.



	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES           Fabray Gómez          

FIRMA ASESOR           Luis Fernando Castro Landín          

FECHA ENTREGA: 15 de Marzo de 2017

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD \_\_\_\_\_

RECHAZADO \_\_\_\_\_ ACEPTADO \_\_\_\_\_ ACEPTADO CON MOD. \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

FIRMA CONSEJO DE FACULTAD \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_