PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO – CHILE
ESCUELA DE INGENIERÍA ELÉCTRICA

EXTENSIÓN DEL ANCHO DE BANDA DE ESPECTROMETROS RADIO ASTRONÓMICOS BASADOS EN FPGA.

ANDRÉS ARTURO ALVEAR CABEZÓN

INFORME FINAL DEL PROYECTO PRESENTADO EN CUMPLIMIENTO DE LOS REQUISITOS PARA OPTAR AL TÍTULO PROFESIONAL DE **INGENIERO CIVIL ELECTRÓNICO**

2014

ESTE TRABAJO HA SIDO PARCIALMENTE FINANCIADO POR: CONICYT, A TRAVÉS DE SU PROGRAMA FONDO BASAL "CENTRO DE ASTROFÍSICA Y TECNOLOGÍAS AFINES", CATA, PROYECTO PBF06. AGRADECEMOS A XILINX INC. POR LA DONACIÓN DE CIRCUITOS INTEGRADOS Y LICENCIAS DE SOFTWARE.

EXTENSIÓN DEL ANCHO DE BANDA DE ESPECTROMETROS RADIO ASTRONÓMICOS BASADOS EN FPGA.

INFORME FINAL

Presentado en cumplimiento de los requisitos
para optar al título profesional de

INGENIERO CIVIL ELECTRÓNICO

otorgado por la

ESCUELA DE INGENIERÍA ELÉCTRICA

de la

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

Andrés Arturo Alvear Cabezón.

Profesor Guía:	Sr. Juan Vignolo Barchiesi
Profesor Correferente:	Sr. Ricardo Finger
Profesor Correferente:	Sr. Sebastian Fingerhuth

2014

ACTA DE APROBACIÓN

La Comisión Calificadora designada por la Escuela de Ingeniería Eléctrica ha aprobado el texto del Informe Final de Proyecto de Titulación, desarrollado entre el Primer Semestre de 2013 y el Segundo Semestre de 2013 y denominado

EXTENSION DEL ANCHO DE BANDA DE ESPECTROMETROS RADIO ASTRONOMICOS BASADOS EN FPGA.

Presentado por el Señor
Andrés Arturo Alvear Cabezón

Sr. Juan Vignolo Barchiesi
Profesor Guía

Sr. Sebastián Fingerhuth
Segundo Revisor

Sr. Héctor Vargas
Secretario Académico

Valparaíso, 2014

Este trabajo está dedicado a mi Padre por brindarme su apoyo y estar siempre cuando más se necesita.

Para mi papá, Milton Alvear.
3 de marzo de 2014.

EXTENSIÓN DEL ANCHO DE BANDA DE ESPECTRÓMETROS RADIO ASTRONÓMICOS BASADOS EN FPGA.

ANDRES ARTURO ALVEAR CABEZON
Profesor Guía:
Sr. Juan Vignolo.

Profesores correferentes miembros de la comisión:
Dr. Ricardo Finger.
Dr. Sebastián Fingerhuth.

RESUMEN

La presente memoria reporta la extensión de ancho de banda de un receptor radioastronómico basado en espectrómetros de Transformada Rápida de Fourier (FFT) implementados sobre un chip *Field Programmable Gate Array* (FPGA). Los FPGA son dispositivos semiconductores que contienen bloques de lógica, cuya interconexión y funcionalidad se puede programar. Para extender el ancho de banda se optimizó el diseño implementado en el chip, con el propósito de mejorar la sincronización entre sus bloques de lógica funcionales.

La expansión del ancho de banda implicó reposicionar manualmente los recursos críticos y generar nuevas áreas dentro del chip. Específicamente se reubicaron los elementos cruciales del diseño, tales como memorias ram y bloques de procesamiento DSP. Esta técnica de optimización se denomina *Floorplanning* y se realizó con el software PlanAhead. Una vez realizado el *Floorplanning* se crearon restricciones con el objetivo de mejorar las estrategias de Ubicación e Interconexión (PAR) durante el proceso de compilación.

Asimismo, este trabajo contempló un análisis sobre la implementación de la estrategia de optimización utilizada y la matemática de los espectrómetros con separación de banda lateral. Además, se estudió la posibilidad de mejorar la resolución espectral y la posibilidad de extensión del ancho de banda necesaria para futuras investigaciones sobre ciencia atmosférica y radioastronomía.

Se logró extender el ancho de banda original del receptor (500 MHz) a 1000 MHz, potenciando así el análisis espectral de la radiación electromagnética captada por la antena del radiotelescopio.

Los resultados de esta memoria se constituyen en una guía para llevar a cabo futuras optimizaciones de la arquitectura FPGA Virtex-5 y Virtex-6, con los softwares que provee Xilinx.

ÍNDICE

APÉNDICE A
TUTORIAL 1: OPTIMIZACION DE VELOCIDAD CON PLANAHEAD

APÉNDICE B
TUTORIAL 2: ARCHIVO DE RESTRICCIONES

APÉNDICE C
TUTORIAL 3: COMPILACIÓN DEL DISEÑO OPTIMIZADO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO DE TÉRMINOS

ADC: acrónimo en inglés de “Analog to Digital Converter”. Es un dispositivo electrónico capaz de convertir una señal analógica de voltaje en una señal digital con un valor binario.

ASIC: Circuito Integrado para Aplicaciones Específicas (ASIC, por sus siglas en inglés). Es un circuito integrado hecho a la medida para un uso en particular, en vez de ser concebido para propósitos de uso general. Se usa para una función específica, por ejemplo, un chip diseñado únicamente para ser usado en un teléfono móvil es un ASIC.

BACK-END: su objetivo es efectuar la detección y procesado posterior de la señal astronómica. Este trabajo se enfoca en esta etapa de los receptores.

CASPER: Center for Astronomy Signal Processing and Electronics Research, UC Berkeley.

DSP: sigla en inglés de “Digital Signal Processing”. Es un área de la Ingeniería Electrónica que se concentra en la representación, transformación y manipulación de señales, así como de la información que ellas contienen.

DFT: sigla en inglés de “Discrete Fourier Transform”. Es una transformación tiempo-frecuencia similar a la DTFT, pero es discreta en ambos dominios y opera sobre un número finito de datos N, entregando N componentes de frecuencia, indexadas mediante la variable entera k.

DELAY: es la cantidad de clocks que necesita un bloque para dar un resultado válido frente a un cambio en la entrada.

FFT: sigla en inglés de “Fast Fourier Transform”. Es un algoritmo (no una transformada) que permite calcular la DFT con menor número de operaciones aritméticas que las que se requieren al aplicar directamente la fórmula de la DFT.

FIR: sigla en inglés de “Finite Impulse Response”. Este es filtro digital de respuesta finita. Cada muestra de la salida depende solamente de un número finito de muestras de la entrada, las cuales pueden incluir muestras pasadas, la muestra presente y muestras futuras.

FLOORPLANNING: en diseño microelectrónicos, el floorplan de un Circuito Integrado es una representación esquemática de la ubicación tentativa de sus bloques de mayor funcionalidad. El concepto Floorplanning corresponde al proceso iterativo donde se busca optimizar espacio y recursos del chip.

FPGA: acrónimo en inglés de “Field Programmable Gate Array”. Es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad se puede programar.

FRONT-END: es la parte del receptor radioastronómico adaptado a la señal que se pretende captar.

IF: acrónimo en inglés de “Intermediate Frequency”, se traduce al español como Frecuencia Intermedia. La IF es el nombre que toma la banda de frecuencias tras el proceso de up-conversion o down-conversion.

VHDL: sigla en inglés compuesta de VHSIC y HDL, donde VHSIC es la sigla de “Very High Speed Integrated Circuit” y HDL es la sigla de “Hardware Description Hardware”. Lenguaje definido por la IEEE (ANSI/IEEE 1076-1993) para describir circuitos digitales. Lenguaje utilizado principalmente para programar dispositivos FPGA.

LATENCIA: es el tiempo que transcurre entre que se alimenta la entrada de un módulo con información y se obtiene el resultado a la salida.

PAR: acrónimo en inglés de “Placement and Routing”, ubicación e interconexión de los bloques funcionales sobre un chip FPGA.

ROACH: acrónimo en inglés de Reconfigurable Open Architecture Computing Hardware. Es una plataforma de procesamiento FPGA independiente.

THROUGHPUT: es la tasa de datos por unidad de tiempo. En general se mide en bits/s, bytes/s, símbolos/s, etc.

XILINX: industria líder de semiconductores que desarrolla FPGA.

INTRODUCCION

Los estudios de las ondas de radio son de gran importancia para científicos e ingenieros que están constantemente buscando nuevas herramientas y creando novedosas técnicas para entender con mayor claridad las señales electromagnéticas que llegan a la Tierra. Esto ha permitido un gran crecimiento de este campo de investigación, siendo hoy en día indispensable para el estudio de distintos problemas astronómicos, tales como la radiación de fondo cósmico, radio galaxias, pulsares, formación y evolución estelar, así como el estudio de nubes moleculares, entre otros.

Durante la última década las observaciones astronómicas han generado nuevos requerimientos para futuras investigaciones, a medida que grandes regiones del espacio han sido estudiadas con los actuales telescopios. Estos nuevos requerimientos deben ser satisfechos con instrumentación de mayor sensibilidad, rango dinámico y rango de visión [1]. Además, se requiere contar con excelentes condiciones atmosféricas, por lo que se deben emplazar los receptores en lugares en que la interferencia de la atmósfera sea mínima. Actualmente al interior de la Segunda Región del país, específicamente en las cercanías de San Pedro de Atacama, se encuentran cinco observatorios radio astronómicos de última tecnología, destacando los proyectos APEX (*Atacama PathFinder Experiment*), ASTE (*Atacama Submillimeter Telescope Experiment*), CBI (*Cosmic Background Imagener*), NANTEN y el proyecto ALMA (*Atacama Large Millimeter Array*). Este último caracterizado por ser el mayor interferómetro en la banda milimétrica a nivel mundial.

Esta situación ha convertido a Chile en una ubicación privilegiada para la realización de proyectos de gran envergadura, planteando a su vez el desafío de generar profesionales capacitados para desempeñarse en la compleja problemática que presenta este tipo de proyectos. En esta línea, el Laboratorio de Ondas Milimétricas-Submilimétricas [2], ubicado en las dependencias del Departamento de Astronomía de la Universidad de Chile, participa activamente en el desarrollo de instrumentación astronómica, teniendo áreas de desarrollo como *Front-Ends* y *Back-Ends*.

Uno de los trabajos realizados en este ámbito corresponde a *"A calibrated digital sideband separating spectrometer for radio astronomy applications",* del PhD Ricardo Finger [3], el cual consiste en el desarrollo de un receptor radioastronómico de configuración 2SB clásica, que es capaz de realizar el proceso de separación de banda lateral de forma digital. El proceso de separación de banda lateral es realizado por un chip FPGA de alto rendimiento equipado con un analizador de espectro.

Esta novedosa tecnología permite separar de forma óptima la banda de interés de la banda de frecuencia adyacente o banda lateral, la cual adiciona ruido y señales indeseadas a las observaciones incrementando el tiempo de observación. Los actuales radiotelescopios, basados en tecnología analógica, pueden cancelar la contaminación de banda lateral en hasta un 90%. La nueva tecnología permite reducir dicha contaminación en un 99,99%.

El diseño del espectrómetro implementado en el chip incurre en una extensiva utilización de los recursos del FPGA, limitando la frecuencia de reloj a 125 MHz y, a su vez, el ancho de banda del sistema a 500 MHz. Sin embargo, el ancho de banda requerido para el análisis espectral de la radiación electromagnética captada por la antena del radiotelescopio debe ser de al menos 1000 MHz.

Por lo anterior surge la necesidad de desarrollar la presente memoria, cuyo objetivo es desarrollar una optimización del chip FPGA para extender el ancho de banda del receptor.

Para estos fines se utilizó un chip FPGA de alto rendimiento Virtex-5 (XC5VSX95T-1) y las herramientas de software que proporciona la compañía Xilinx.

La metodología que se utilizó para el desarrollo de este trabajo consistió en aprovechar los beneficios que proporciona el *Toolflow* estándar [4], así como las capacidades y beneficios que proporciona el uso del software Xilinx PlanAhead para el diseño FPGA. Este documento contiene una guía (véase Apéndice), con una serie de pasos que permiten mejorar el rendimiento del diseño y la sincronización del chip.

CAPÍTULO 1

CONTEXTUALIZACIÓN Y REVISIÓN BIBLIOGRÁFICA

Esta memoria se desarrolla sobre la base de un marco teórico que se focaliza en la radioastronomía e instrumentación para el desarrollo de la ciencia astronómica. Esto permitió orientar la ejecución del trabajo y determinar la relevancia para la radioastronomía.

1.1 La Radioastronomía

La observación astronómica es el método por excelencia que tiene el hombre para alimentar su curiosidad acerca del Universo. La radiación electromagnética se recibe mediante artefactos radicados en tierra o en el espacio. La emisión de esta radiación puede clasificarse según su longitud de onda (revisar Tabla 1-1). Para percibir o captar dichas emisiones se utilizan diferentes herramientas de detección y en distintas localidades (en la Tierra o en el espacio), que permiten descifrar información presente que no es posible obtener sin instrumentos.

Tabla 1-1 Espectro Electromagnético [5]

Nombre	Longitud de Onda	Frecuencia
Rayos gamma	menor que 0.01nm	Mayor que 10 EHZ
Rayos-X	0.01 a 10 nm	30 PHz – 30 EHZ
Ultravioleta	10 nm – 400 nm	30 EHZ – 790 THz
Visible	390 nm – 750 nm	790 THz – 405 THz
Infrarrojo	750 nm – 1 mm	405 THz – 300 GHz
Microonda	1 mm – 1 m	300 GHz – 300 MHz
Radio	1 mm – 1km	300 GHz – 3 Hz

Una de las ventanas electromagnéticas más usadas en los últimos 20 años corresponde a la ventana de radio, que es excepcionalmente transparente para la atmósfera terrestre, por lo que permite la realización de observaciones desde la superficie de la Tierra. Esto se debe a la opacidad de la atmósfera (Fig. 1-1), que filtra señales electromagnéticas con longitudes de onda milimétricas y submilimétricas.

En la banda superior, es decir, entre los 100 GHz y 1.5 THz existen una serie de ventanas de observación, que permiten la realización de mediciones en estas frecuencias. De todas formas, estas ventanas exhiben una fuerte dependencia de las condiciones atmosféricas.

Para llevar a cabo investigaciones científicas con esta información se debe proveer a los astrónomos del medio de observación. La instrumentación astronómica ha permitido desarrollar en las últimas décadas receptores radioastronómicos para tales

fines. La tecnología utilizada debe ser capaz de detectar señales que provienen de fuentes radioemisoras muy lejanas. Los requerimientos de sensibilidad, resolución y precisión de estos equipos son muy superiores a los utilizados en radios comunes. Estos receptores deben ser ubicados en lugares en que la interferencia de la atmósfera sea mínima.

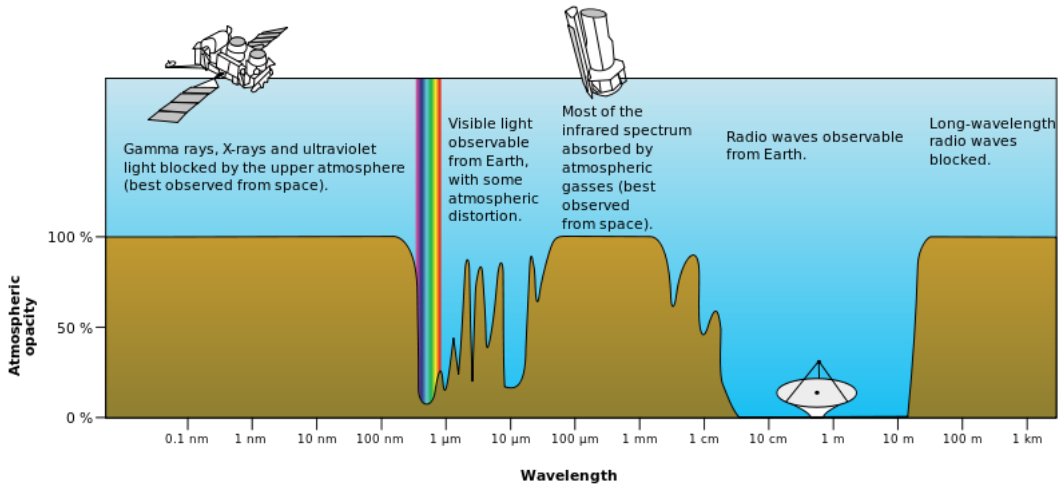

Fig. 1-1 Opacidad de la atmósfera terrestre [5]

1.2 Los receptores radioastronómicos

Los receptores usados para aplicaciones radioastronómicas son los clásicos usados en telecomunicaciones. Estos son llamados receptores heterodinos, que captan ondas de radio mediante un proceso de mezcla de frecuencias o heterodinación para convertir la señal recibida en una frecuencia intermedia fija, la cual puede ser más convenientemente elaborada (filtrada y amplificada) que la frecuencia original. Prácticamente todos los receptores modernos de radio y televisión utilizan el principio heterodino.

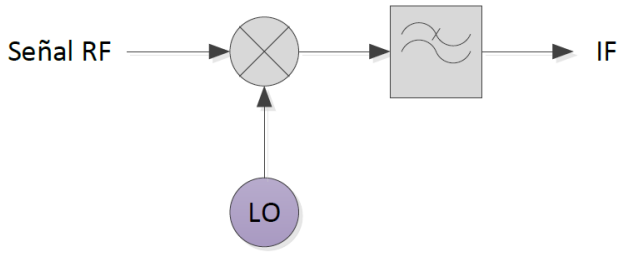

Fig. 1-2 Diagrama de bloques de un receptor heterodino

En la Fig. 1-2 se presenta un diagrama de bloques que explica de manera general el principio de funcionamiento de estos sistemas, los cuales se componen de un mezclador que multiplica la señal RF con una señal de frecuencia conocida denominada oscilador local (LO) y luego un filtro pasabajos, que elimina las componentes de frecuencia innecesarias dentro del diseño realizado. La señal de

entrada corresponde a la que se desea detectar. A ésta se le aplica el desplazamiento en frecuencia, mientras que la señal de LO es una sinusoide de frecuencia fija pero controlable.

A grandes rasgos, el receptor heterodino es un dispositivo que permite desplazar en frecuencia el espectro de una señal, pudiéndose obtener representaciones centradas tanto en frecuencias superiores como en inferiores. El proceso de obtener como resultado un espectro centrado en frecuencias superiores a la original, es conocido como *up-conversion*; mientras que cuando es a la inversa, es decir, cuando el espectro se presenta en frecuencias inferiores, se le conoce por el término *down-conversion*.

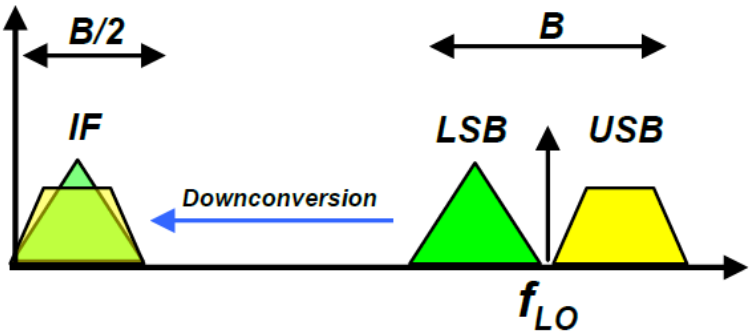

Fig. 1-3 Superposición de espectros ubicados a la misma distancia del Oscilador Local [3]

Por la paridad matemática de la función coseno, el proceso de *down-conversion* también es posible realizarlo cuando el espectro de la señal que está siendo detectada se encuentra en frecuencias inferiores a la del LO. Esta última propiedad resulta ser un problema cuando en una de esas posiciones se encuentra ubicado un espectro no deseado, ya que terminará superponiéndose al de interés tal como se ilustra en la Fig. 1-3.

Para solucionar este problema se utilizan habitualmente técnicas de separación de banda lateral, que permiten separar los espectros de interés con el espectro no deseado.

La configuración preferida para los receptores heterodinos radioastronómicos es la 2SB (en inglés *Dual SideBand*). La principal característica de este receptor es que permite una observación simultánea y esclarecedora de las IF. Sin embargo, este tipo de receptores separadores de banda lateral son difíciles de construir, ya que requieren la implementación de dos híbridos RF, más dos mezcladores y amplificadores en paralelo con una excelente amplitud y balance de fase sobre anchos de banda amplios [3]. La Fig. 1-4 muestra la configuración estándar de una interfaz 2SB.

La primera etapa consiste en un híbrido de RF, que divide la señal entrante en dos caminos con un desfase de 90°. Las salidas del híbrido RF serán desplazadas en

frecuencia por los mezcladores M1 y M2, los cuales están conectados al mismo oscilador local (LO).

Una vez que el LO y la señal RF han sido mezcladas, mediante la utilización de una etapa híbrida IF [6], se obtienen I_1 e I_2, que corresponden a USB y LSB.

En implementaciones reales la separación de la banda lateral es imperfecta, ya que siempre parte de la información de las bandas USB o LSB se fuga en el puerto no deseado. En cada salida se tiene una combinación de ambas bandas laterales. Para un receptor de banda ancha se obtienen ganancias inadecuadas y desequilibrios de fase que limitan fuertemente la relación de rechazo de banda lateral hasta unos 10-20 dB de sensibilidad.

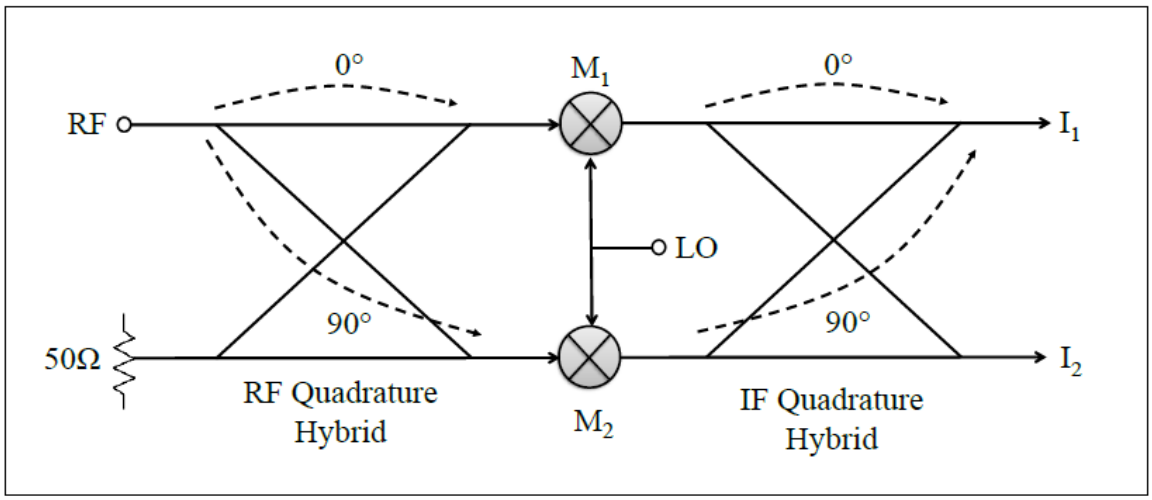

Fig. 1-4 Configuración típica de un receptor 2SB. Idealmente las salidas I_1 e I_2 tienen una de las componentes de la IF, la LSB o la USB [3]

La velocidad creciente de hardware digital ha abierto la puerta a un nuevo enfoque que se basa en la idea de llevar a cabo la recombinación IF digitalmente. En este método las salidas del mezclador son digitalizadas directamente a través de dos ADC de modo que un IF digital puede ser implementado para corregir los desequilibrios de la RF analógico, mediante la implementación de un diseño de un Híbrido IF, basado en procesamiento digital de señales como el que se muestra en la Fig. 1-5.

La Fig. 1-5 muestra el diagrama de bloques del espectrómetro separador de banda lateral de configuración 2SB clásica, que es capaz de digitalizar las salidas de los mezcladores y hacer una separación de banda lateral de forma digital usando un chip FPGA de alto rendimiento equipado con un analizador de espectro, basado en una PFB [7] [8]. La combinación de un filtro pasabajo y una FFT, es conocido como PFB (Polyphase Filter Bank). El PFB permite que la respuesta en frecuencia de un canal sea plana y además permite suprimir señales que aparecen por fuga (leakage) fuera de la banda (out-of-band signals).

Este tipo de filtros son un desafío debido a la gran cantidad de recursos que utiliza su implementación en un *chip* FPGA, pero la utilización de esta técnica permitirá mejorar el rendimiento del espectrómetro.

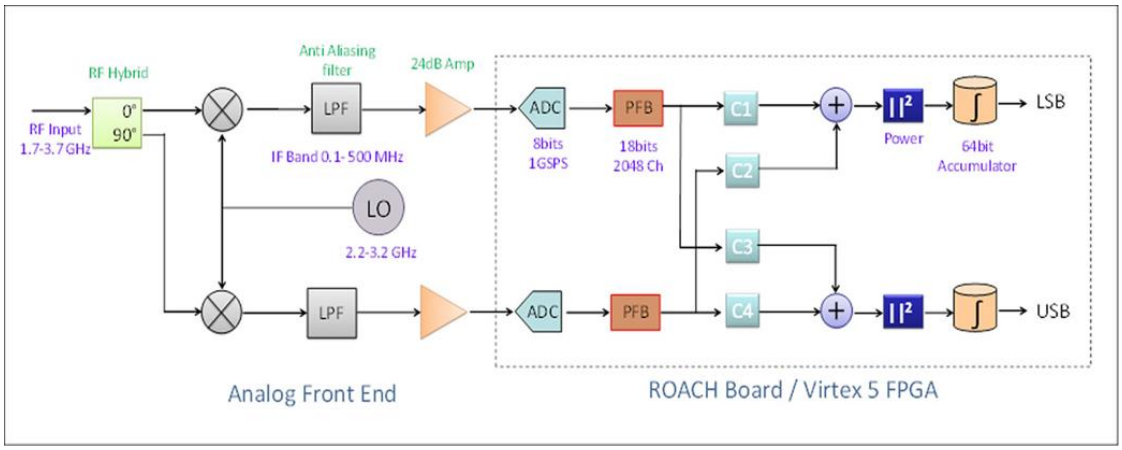

Fig. 1-5 Diagrama de bloques del doble separador de banda lateral digital [3]

Luego que se calcula el espectro en la PFB, las componentes C1 y C4 son posicionadas en 1, mientras que C2 y C3 deben ser ajustadas para calibrar la fase y amplitud y de esta manera corregir los problemas de balance de fase y amplitud que se producen por la no linealidad de los dispositivos. Al igual que el rango de acumulación, C2 y C3 deben ser calibrados por el usuario y cargados desde un computador remoto [3]. Estos receptores son útiles para la observación de espectros astronómicos complejos en un amplio rango de frecuencias y permiten evitar la confusión espectral.

1.3 Dispositivos de interés para el trabajo

En esta sección se abordan temas que tienen especial relevancia en el proyecto de esta memoria. Se revisan dispositivos que forman parte esencial dentro de la instrumentación astronómica, principalmente los FPGA y sus estructuras internas utilizadas en las etapas de entrada/salida y control del sistema. Además se presentan los ADC y el hardware utilizado.

El chip FPGA Virtex-5 (XC5VSX95T-1) está embebido en una plataforma llamada ROACH, desarrollada por CASPER (sigla en inglés Center for Astronomy Signal Processing and Electronics Research at UC Berkeley), que es un centro de investigación y desarrollo científico de la Universidad de Berkeley (EE.UU), para la instrumentación astronómica.

A continuación se presenta la tecnología del FPGA Virtex-5 de Xilinx, que mediante su hardware permite implementar aplicaciones DSP de alto rendimiento.

1.3.1 FPGA

En esta década el campo de la computación está siendo dominado por sistemas embebidos, aplicaciones informáticas y procesadores de aplicación específica. Cuando los requisitos en costo, consumo de potencia, entre otros, no pueden ser cubiertos por los procesadores comerciales, los ingenieros normalmente se ven obligados a diseñar hardwares a medida, empleando circuitos integrados de aplicación específica, conocidos en inglés como *Application Specific Integrated Circuit* (ASIC). Para construir esos sistemas los diseñadores necesitan disponer de tecnologías que permitan implementar sistemas sofisticados y de herramientas de diseño asistido por computadora (CAD), que ayuden en la generación de arquitecturas que cumplan las especificaciones de cada aplicación. La metodología de diseño se basa en la experiencia del diseñador para implementar una arquitectura determinada en hardware. Este trabajo resulta muy tedioso y propenso a la introducción de errores, implicando un tiempo de depuración considerable.

En la década de los ’80 con la aparición de la tecnología FPGA (Field Programmable Gate Array o arreglo de compuertas programable por el usuario) se dio un giro a la industria de los sistemas embebidos. Esto permitió disminuir los tiempos de desarrollo de sistemas embebidos, ya que la nueva tecnología, mediante la reconfiguración de bloques de lógica, da como resultado un chip para alguna aplicación específica.

Los FPGA también significaron un cambio en la forma de pensar la computación, al presentar mayores ventajas frente a los computadores tradicionales. Esta tecnología permite realizar cálculos en forma paralela y funciones tan simples como las compuertas lógicas o tan complejas como las de un microprocesador.

Las aplicaciones donde más comúnmente se utilizan los FPGA incluyen procesamiento digital de señales, radioastronomía, sistemas aeroespaciales y de defensa, sistemas de imágenes para medicina, sistemas de visión para computadoras, reconocimiento de voz, bioinformática, emulación de hardware de computadora, *reconfigurable computing*, prototipos de ASIC, entre otras.

Es importante tener presente que su uso en otras áreas es cada vez mayor, sobre todo en aquellas aplicaciones que requieren un alto grado de paralelismo. Se estima que un FPGA supera 500 veces o más el rendimiento de un procesador DSP, por lo que son usados en sistemas en tiempo real o en aquellos que requieren una alta velocidad de procesamiento.

1.3.2 Arquitectura de un FPGA Virtex-5

Estos arreglos de compuertas lógicas son dispositivos electrónicos semiconductores que internamente están compuestos por bloques lógicos a los cuales se les puede programar, de manera tal que sus unidades funcionales estén interconectadas para generar la configuración deseada. Dichos bloques están compuesto por grupos de compuertas lógicas basadas en LUTs (look up tables) o bien

llamados CLB (Bloques Lógicos Configurables), IOB (Bloques de entrada y salida), memorias de acceso aleatorio (Blocks RAM, Random Access Memory), Bloques DSP (dedicados exclusivamente a realizar operaciones matemáticas) y Administrador de Reloj Digital (DCM).

Es importante mencionar para el futuro diseño que existen conexiones entre los elementos de los CLB y que pueden ser configuradas para obtener una personalidad diferente de la FPGA. Además, existen diferentes tipos de CLB, algunos compuestos por circuitos de baja capacitancias con la finalidad de transmitir señales de sincronización [9].

1.3.3 Slices

En el caso de la arquitectura Virtex-5 existen dos tipos de Slices, las SlicesM y SlicesL. Véase la FIG. 1-6. La Slice tipo M proporciona mayor capacidad de almacenamiento en memoria y la Slice tipo L permite implementar lógica combinacional.

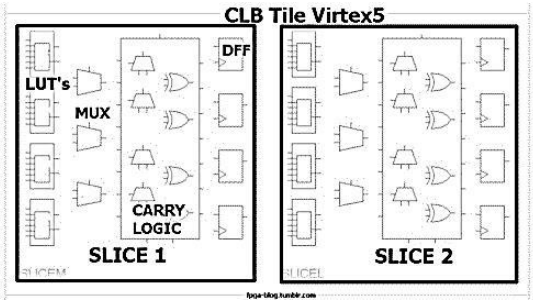

FIG. 1-6 Slices

1.3.4 Bloque Lógico Configurable

Un paso arriba en la jerarquía dentro de la nomenclatura de Xilinx, se encuentra el CLB. Algunos FPGA de Xilinx tienen dos Slices en cada CLB, como se muestra en la Fig. 1-7. La razón para tener este tipo de arquitectura jerárquica, es que es complementada por una arquitectura equivalente en la conexión.

De tal forma, se tienen conexiones más rápidas dentro de un Slice, luego conexiones un poco más lentas entre Slices y conexiones aún más lentas entre CLB. Su finalidad es lograr que las conexiones sean fáciles, sin incurrir en retardos de interconexión.

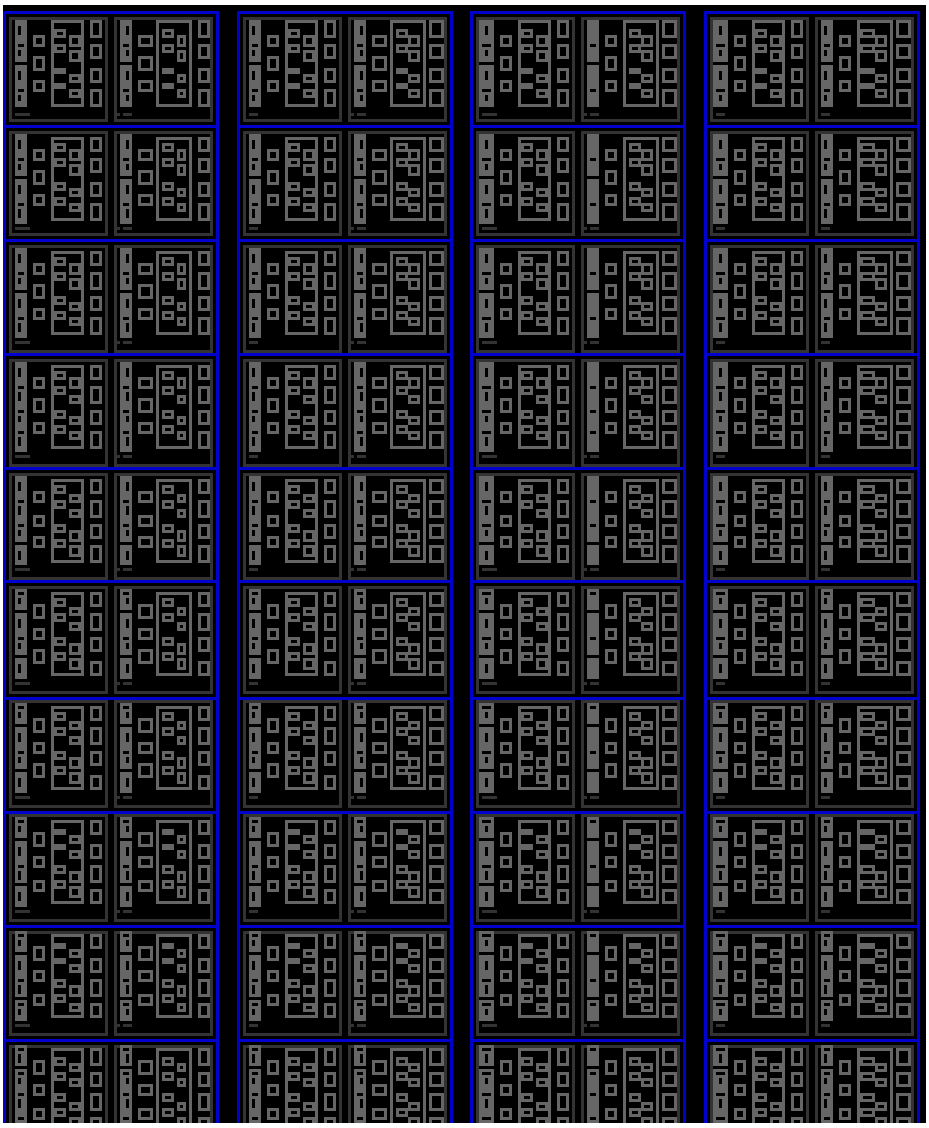

Fig. 1-7 Bloques Lógicos Configurables (CLB)

1.3.5 Bloques DSP48E

Al tener requerimientos de diseño más complejos, como los utilizados en aplicaciones DSP, un hardware basado en sistemas digitales básico se vuelve proporcionalmente más complejo y su desempeño empeora a medida que el diseño va utilizando más recursos. En algunos casos el diseño no es escalable dentro del chip. En particular, el chip Virtex-5 contiene bloques dedicados para aplicaciones DSP llamados DSP48E.

Cuando se requiere realizar operaciones matemáticas complejas, como por ejemplo calcular de formar eficiente una FFT, se puede utilizar bloques DSP48E. Con

esto se pueden implementar operaciones matemáticas, diseñar sistemas sumadores y multiplicadores de números basados en lógica básica.

En la Fig. 1-8 se aprecia una ilustración del diagrama de bloques que describe la funcionalidad de este artefacto, que posee una ALU con la cual realiza operaciones matemáticas en punto fijo, tales como suma, resta, multiplicación, división y operaciones lógicas. Este bloque posee dos entradas A y B, una de 25 bits y otra de 18 bits; y una entrada C que soporta 48 bits. Esta entrada permite conectar varios de estos bloques en cascada o en paralelo y su salida P también es de 48 bits.

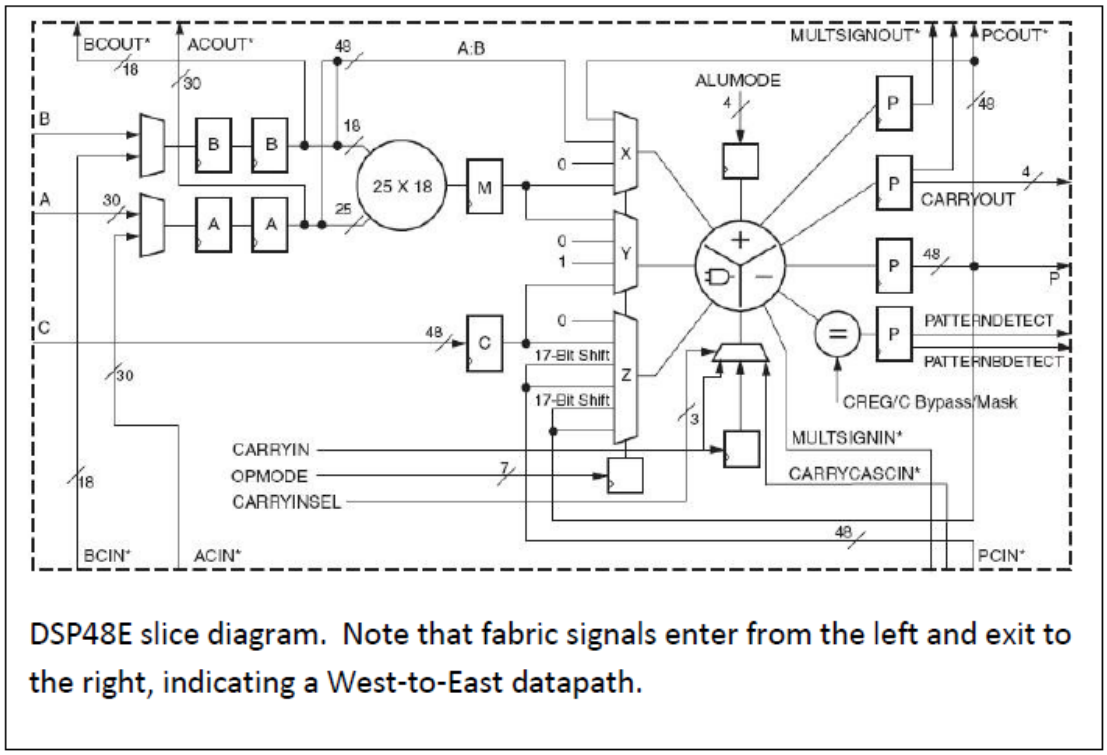

Fig. 1-8 Diagrama de bloque DSP48E de la Arquitectura FPGA Virtex-5 [10]

1.3.6 Bloques BRAM

Cuando se requiere realizar acumulación de datos, integradores o refrescar un vector, se puede utilizar una BRAM. Además se usa para almacenar vectores. Existen dos modos de funcionamientos: Load y Accumulate [10].

Cada BRAM puede almacenar 1024 valores complejos. Si se estipula que el fanout máximo de los bloques DSP48E es cuatro, luego el máximo tamaño de canales para el cálculo de la FFT es 2^{15} [11]. Sin embargo, si se utilizan diseños optimizados de memoria se pueden realizar cálculos de una FFT con 2^{21} puntos [12].

En la Fig. 1-9 se presenta un diseño esquemático de un espectrómetro extraído del software PlanAhead. En las columnas de color magenta se ubican las BRAM y

cada columna contiene 32 de estos bloques. En las columnas de color cian se ubican los sitios de los bloques DSP48E y cada columna contiene 64 de estos bloques. La lógica (en bloques azules) que está contenida dentro de los CLB se distribuye de manera matricial dentro del chip.

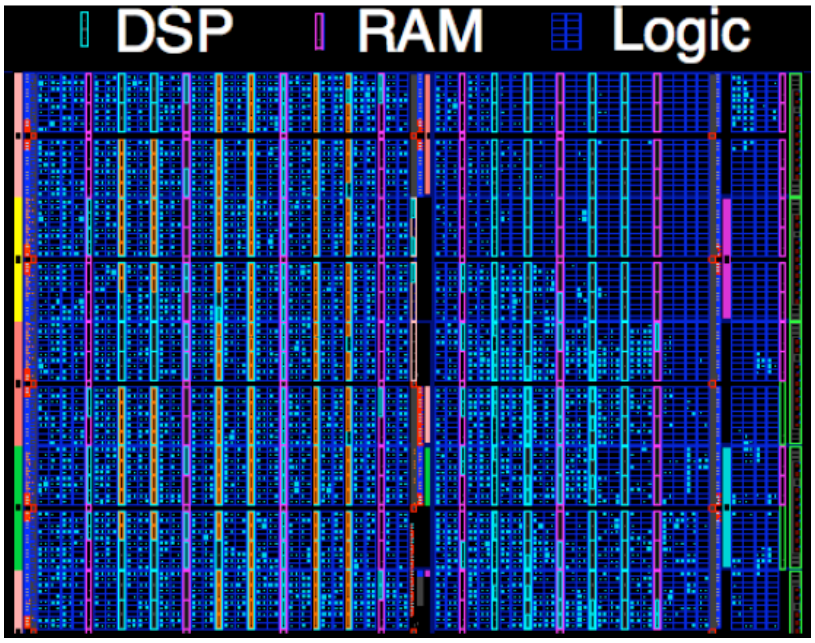

Fig. 1-9 Diseño de espectrómetro, esquemático extraído desde el software PlanAhead

1.4 Hardware utilizado

El hardware usado para el desarrollo de este trabajo es conocido como ROACH (por su sigla en inglés de Reconfigurable Open Architecture Computing Hardware). La plataforma ROACH es un hardware abierto basado en el chip FPGA Virtex-5, producto de un desarrollo colaborativo internacional liderado por CASPER en la Universidad de California Berkeley. La Fig. 1-10 presenta el diagrama de bloques de la tarjeta ROACH.

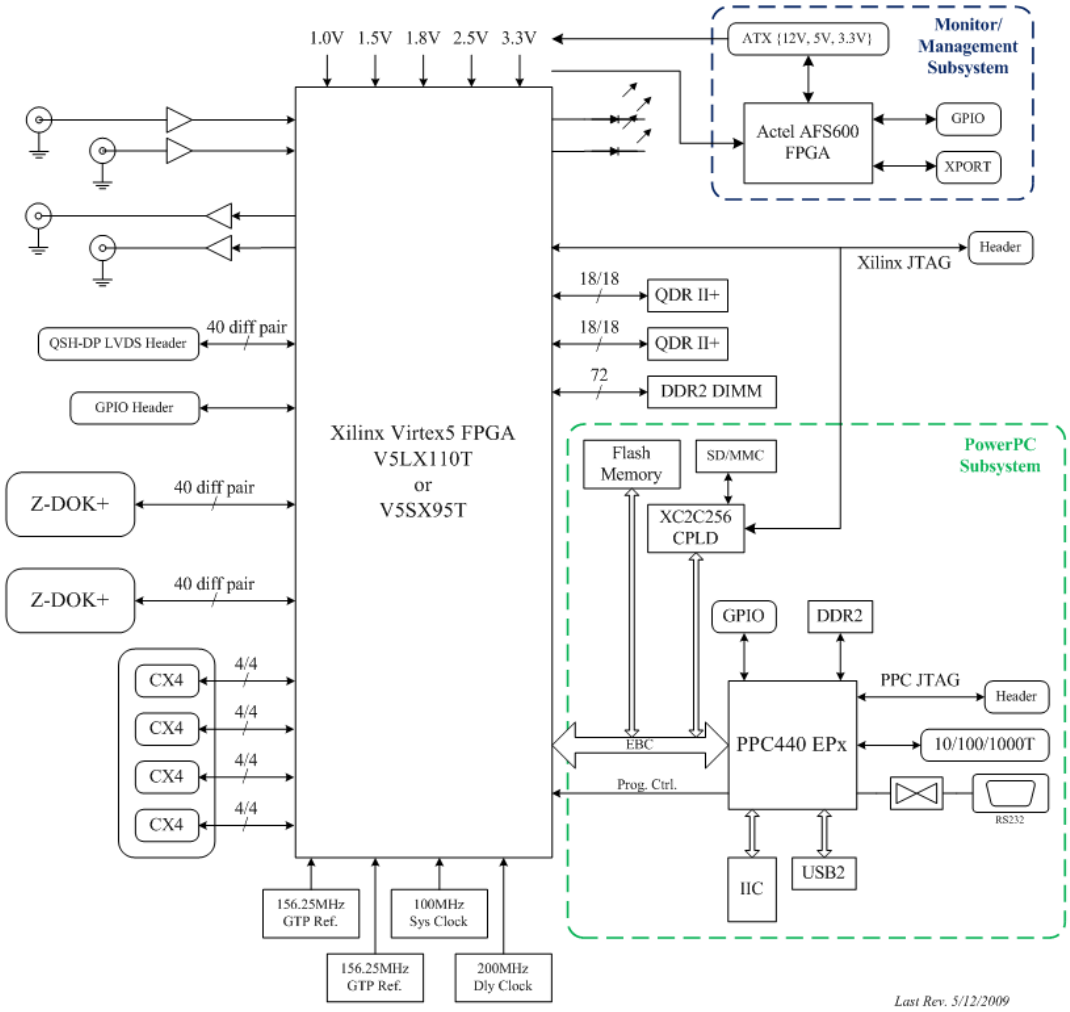

Fig. 1-10 Diagrama de bloques de la plataforma ROACH [13]

Tal como se puede ver en la Fig. 1-10, la pieza central de esta tarjeta es el chip Xilinx Virtex-5 FPGA (LX110T para aplicaciones intensivas en lógica o SX95T para aplicaciones intensivas en procesamiento digital de señales). En la tarjeta hay un chip PowerPC con un sistema operativo basado en Linux, que es usado para controlar la tarjeta y programar el chip FPGA. Además es la interfaz entre dispositivos externos que vía Ethernet pueden leer y escribir los registros, así como leer las memorias del FPGA.

Dos QDR proporcionan alta velocidad y capacidad media de memoria. Una DDR2 DIMM proporciona alta capacidad de almacenamiento en memoria para el FPGA. El PowerPC tiene una DDR2 DIMM independiente, que le permite bootear una distribución Linux, llamada BORPH [14] (por su sigla en inglés de Berkeley Operating System for ReProgrammable Hardware). Este es un sistema operativo para computadores reconfigurables basados en FPGA.

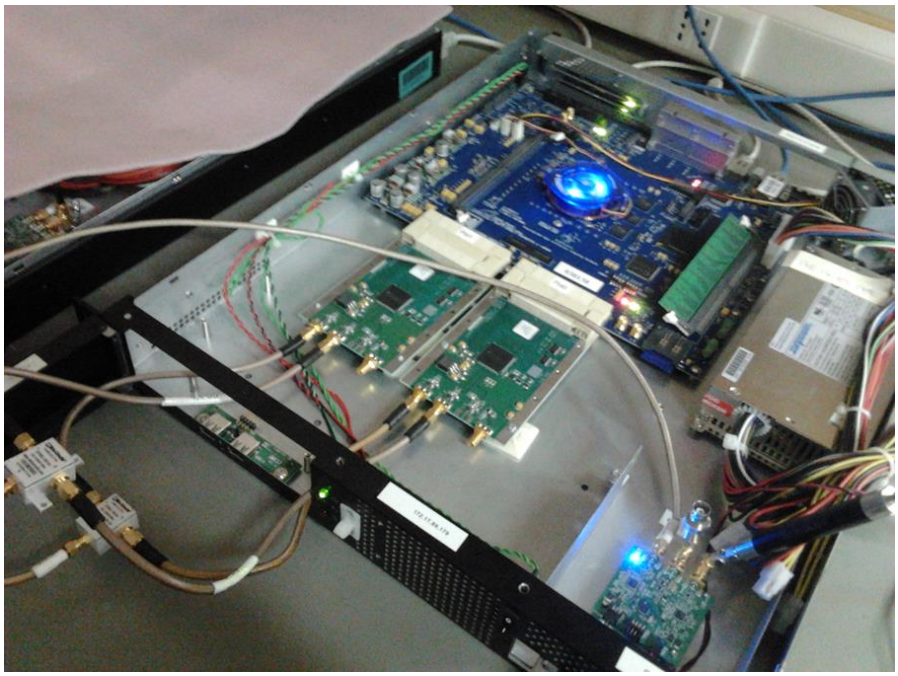

Fig. 1-11 ROACH (tarjeta azul), ADC0 y ADC1 (par de tarjetas verdes), en la esquina se observa sintetizador, que entrega una señal de referencia externa a los ADC

Dos conectores Z-DOK permiten la conexión de tarjetas ADC, que permiten llevar a cabo la conversión análogo digital [13].

La tarjeta ADC está basada en el chip ADC083000 a National Semiconductor [15] que permite tasas de muestreos hasta 3GSPS. La Fig. 1-11 muestra dos tarjetas verdes idénticas, éstas corresponden a los ADC. Cada muestra es de 8 bits. La entrada de los ADC posee una impedancia de 50 [Ω], por lo que se puede calcular el error de cuantización para la potencia a la cual serán utilizados estos ADC:

$$P_{\max\ de\ entrada} = 0\ [dBm] <=> 1\ [mW] \tag{1-1}$$

$$1\ [mW] = \frac{v^2}{R} \tag{1-2}$$

$$v = \sqrt{50 \cdot 0.001} \approx 0.22[v] \tag{1-3}$$

$$Error\ de\ cuantización = \pm\frac{0.22[v]}{2} = \pm 0.11[v] \tag{1-4}$$

CAPÍTULO 2

PROYECTO

Este proyecto de título lleva por nombre "Extensión del ancho de banda de espectrómetros radio astronómicos basados en FPGA". A continuación se presentan los objetivos y alcances del proyecto, además de las especificaciones del receptor y el diseño a optimizar.

2.1 Objetivos

2.1.1 Objetivo General:

Optimización del diseño físico de FPGA para aumentar el ancho de banda de dos espectrómetros implementados sobre plataformas ROACH con un FPGA Virtex-5 SX95T de Xilinx, usando la herramienta de Software PlanAhead de Xilinx Inc.

2.1.2 Objetivos Específicos:

- Instalar el toolflow de CAPSER. Ubuntu 12.04 LTS con Xilinx 14 y Matlab R2012b y librerías para trabajar con ROACH.
- Realizar el tutorial de CASPER-ROACH y PlanAhead.
- Determinar por qué aplicaciones DSP de alto rendimiento sobre FPGA, usando el Toolflow Matlab/Simulink/System_Generator, tienen un bajo rendimiento una vez que la utilización de los recursos del FPGA supera el ~70%; y encontrar soluciones para este problema.
- Estudio de la arquitectura del FPGA Virtex-5 y alternativas para la optimización del diseño físico de FPGA para espectrómetros de alto ancho de banda.
- Con el uso de PlanAhead aumentar el ancho de banda de los espectrómetros actuales de 500 MHz a 1000 MHz manteniendo el número de canales en 2048.

Para el cumplimiento de los objetivos, se realizó un estudio previo de arquitectura de los FPGA, además de un repaso de DSP con énfasis en análisis espectral mediante el uso de la Transformada Rápida de Fourier (FFT), que corresponde a la base para este tipo de instrumentación. Junto con ello se adquirió un conocimiento general de microondas, que permitió entender los procesos de las ondas milimétricas en los receptores heterodinos.

2.1.3 Descripción del sistema a optimizar

En la Fig. 2-3 se presenta lo que se denomina gateware, que es la representación de un algoritmo de procesamiento de datos corriendo en un FPGA. En este caso se utiliza un entorno de alto nivel de desarrollo, como lo es Simulink de Matlab. Todos los bloques relacionados con el hardware son de color amarillo y se pueden encontrar en la biblioteca CASPER XPS, la cual contiene todos los componentes específicos de la tarjeta llamados Yellow Blocks. Los bloques relacionados con DSP se encuentran en la librería CASPER DSP y pueden tener otros colores.

Con estas librerías se desarrolló un espectrómetro con separación de banda lateral que aparece en la Fig. 2-3, con las especificaciones de funcionamiento de un ancho de banda de 500MHz para IF-USB y 500MHz para IF-LSB, con 2048 canales y una resolución espectral de 244KHz. El diseño consta de 2 ADC de 8 bits, con una tasa de muestreo de 1GSPS, 2 PFB de 2048 canales (Filtro Fir más Pipelined FFT), constantes complejas, bloques de cálculo de magnitud y bloques acumuladores. Los bloques amarillos sirven para implementar una interfaz de comunicación entre la FPGA y un computador remoto, para que pueda leer y escribir registros/memorias vía Ethernet.

2.1.4 Especificaciones del receptor previas a la optimización

En la Fig. 2-2se presenta el diagrama de bloques con el que se somete a prueba el receptor con las especificaciones que fueron desarrolladas en un trabajo anterior, presentando por el PhD Ricardo Finger [3]. En el diagrama de bloques se destacan los subsistemas Front-End y Back-End, que en su conjunto conforman el receptor.

- Ancho de banda (IF): 500 MHz
- Número de canales: 2048
- Resolución espectral: 244 kHz
- Velocidad de conversión (desde IF hasta espectro digital): 0.53 segundos. Dependiendo del largo de acumulación que se vaya a utilizar.

$$\text{Velocidad de Conversión de datos} = \frac{N^o Ch}{4}(largo_{acc})(T_{clk}) \tag{2-1}$$

$$\text{Velocidad de Conversión de datos} = \frac{2048}{4}(131072)(8[ns]) \approx 0.53\ [s] \tag{2-2}$$

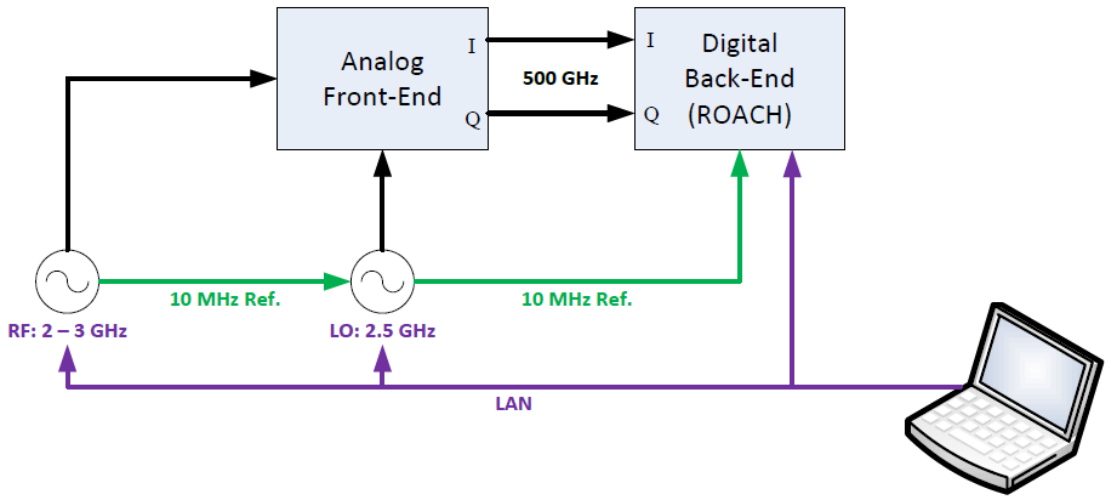

Fig. 2-1 Diagrama de bloques del receptor desarrollado en [3]

2.1.5 Nuevas especificaciones para el receptor

El objetivo final de este trabajo consiste en extender el ancho de banda original del receptor (500 MHz) a 1000 MHz, potenciando así el análisis espectral de radiación electromagnética.

En la Fig. 2-2 se presenta el diagrama de bloques del receptor de pruebas con ancho de banda IF extendido. El *Front-End* de configuración 2SB se encarga de realizar el proceso de *down conversion*, trasladando desde un ancho de banda entre 2 y 4 GHz, a banda base de 1GHz. Esta etapa entrega a la salida las señales I y Q.

Estas señales están listas para ser procesadas por la etapa digital, denominada Back-End, y serán muestreadas por un ADC de 8 bit de resolución, a una tasa de 2GSPS. Luego, una vez digitalizadas, se realiza el proceso de separación de banda lateral mediante un chip FPGA de alto rendimiento equipado con un analizador de espectro.

- Ancho de banda (IF): 1000 MHz
- Número de canales: 2048
- Resolución espectral: 488 kHz
- Velocidad de conversión (desde IF hasta espectro digital): 0.27 segundos. Dependiendo del largo de acumulación que se vaya a utilizar.

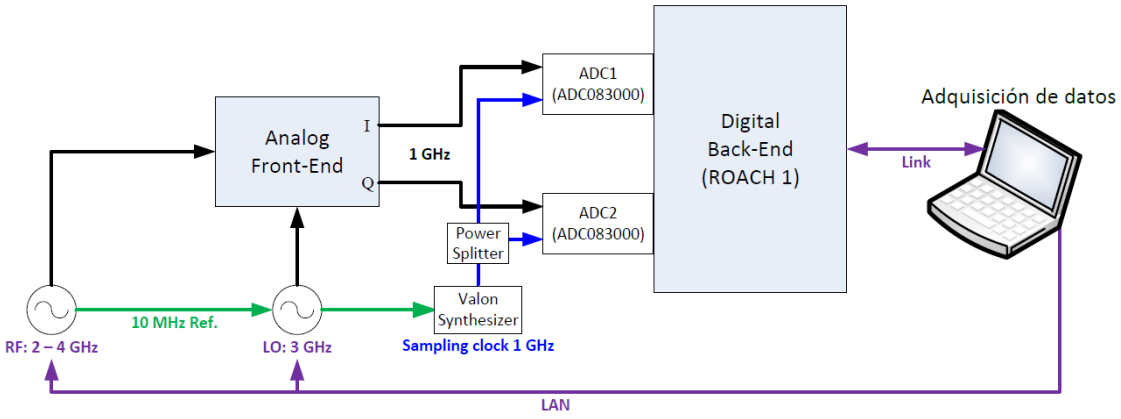

Fig. 2-2 Diagrama de bloques del receptor de pruebas con ancho de banda IF extendido

Para llevar a cabo lo presentado en la figura anterior se presenta una metodología de optimización de velocidad de un hardware implementado en un chip FPGA. Esta metodología de trabajo utiliza como principal herramienta el software PlanAhead, que se basa en el enfoque de Xilinx para realizar Floorplanning. Los siguientes capítulos describen como utilizar este software para hacer un Floorplanning sobre el diseño presentado en la Fig. 2-3.

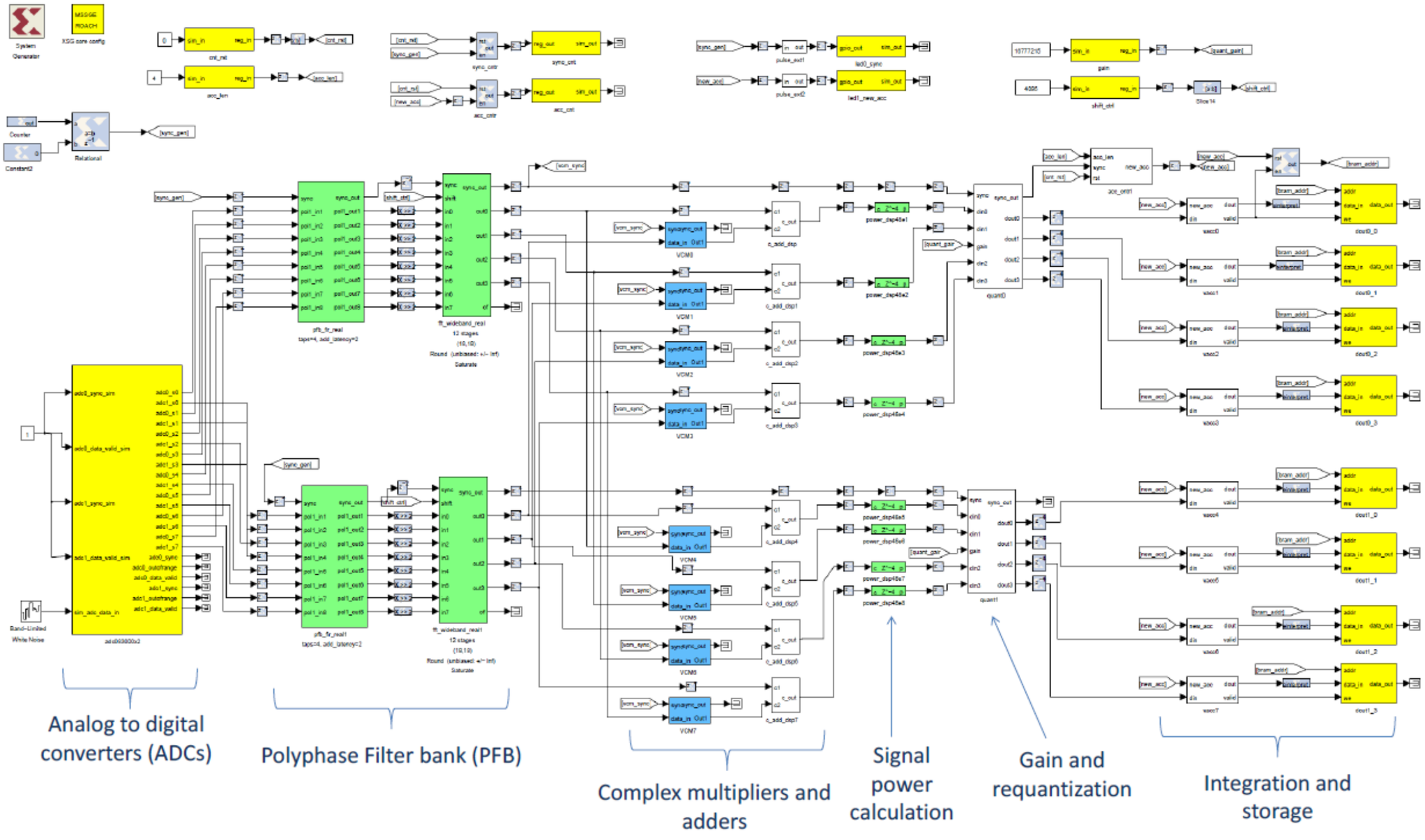

Fig. 2-3 Gateware del espectrómetro con separador de banda lateral digital [3]

CAPÍTULO 3

FLOORPLANNING CON PLANAHEAD

En los capítulos previos se presentó el hardware utilizado y su funcionamiento. Los principales dispositivos del instrumento corresponden a los ADC y al FPGA. Cada ADC está diseñado para obtener ocho muestras por cada ciclo de reloj del FPGA, es decir, el chip FPGA posee una arquitectura preparada para procesar estos datos de forma paralela. Con dos tarjetas ADC sincronizadas con el chip FPGA es posible procesar 16 muestras simultáneas por ciclo de reloj (1/4 de la frecuencia de los ADC). Para que el FPGA pueda procesar un ancho de banda de 1000 MHz debe operar con un reloj de 250 MHz.

En el presente capítulo se exhibe la metodología de optimización que fue empleada al chip FPGA, a fin de ampliar el ancho de banda (500 MHz) del receptor radioastronómico a 1000 MHz.

3.1 Optimización del diseño físico con PlanAhead

Para extender el ancho de banda se debe optimizar el diseño implementado en el chip mejorando la sincronización entre sus bloques funcionales. Además es necesario reposicionar manualmente los recursos críticos y generar nuevas áreas dentro del chip. Esta técnica de optimización se denomina *Floorplanning* y se lleva a cabo con el software PlanAhead [16]. En la Fig. 3-1 se presenta una vista del diseño implementado sobre el FPGA Virtex-5 usando el software PlanAhead.

PlanAhead es una interfaz gráfica que despliega la arquitectura del chip y permite al usuario ingresar restricciones, permitiendo reducir los grados de libertad del algoritmo heurístico que se encarga del posicionamiento de recursos a lo largo del chip. Esto permite mejorar las estrategias de Ubicación e Interconexión (Acrónimo en inglés PAR por Placement And Routing) de recursos durante el proceso de compilación.

3.1.1 Estrategia general de reubicación de recursos

La forma más general y recomendada de hacer Floorplanning es generando áreas dentro del chip, esta es la forma de introducir restricciones para mejorar la sincronización entre sus bloques críticos.

Para generar estas áreas se deben asignar recursos a un P-Block, estos permiten agrupar los recursos en áreas y sitios que el usuario debe designar dentro del chip.

Para guiar la reubicación de recursos del diseño se debe evaluar la sincronización del chip. Mediante la herramienta Timing Analysis se pueden revisar los reportes de sincronización, que reportan información acerca de cada ruta. Estos reportes especifican cuales son las rutas críticas, holguras (Slack), máximos delays y latencias distribuidas dentro del chip.

Para más detalles de cómo usar esta herramienta ver Apéndice A, donde se encuentra una guía completa para llevar a cabo una optimización.

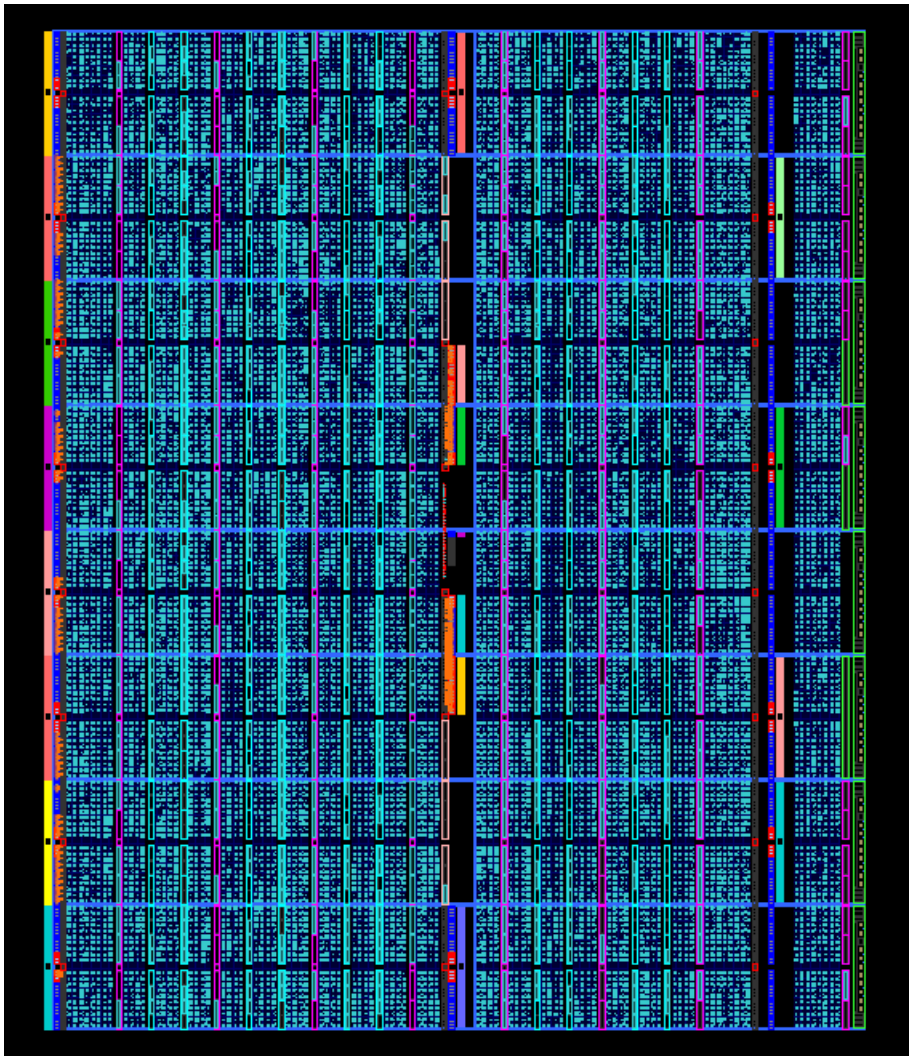

Fig. 3-1 Floorplan de dos espectrómetros de 2048 canales cada uno, generado por las herramientas de Xilinx (no optimizado)

3.1.2 Implementaciones de Hardware con Simulink

El FPGA Virtex-5 tiene varias peculiaridades que deben ser atendidas, en especial cuando se trabaja con System Generator.

Es importante hacer notar que los delays implementados por Xilinx System Gerator se implementan utilizando un bloque SRL16 por bit. Un bloque SRL16 es un elemento de hardware FPGA que permite retrasar un bit hasta 16 ciclos de reloj. Esto tiene dos consecuencias importantes:

En primer lugar, después de la adición de un delay a una ubicación en el hardware, puede añadir hasta 15 delays más sin costo de hardware adicional.

En segundo lugar, si se está tratando de mejorar el enrutamiento (routing), la adición de multiples delays no va a mejorar el *routing* del proceso, porque estos delays serán implementados como un solo SRL16.

Si múltiples latencias son necesarias para mejorar la ruta se deben agregar múltiples delays que funcionen durante un solo ciclo de reloj. Si se quiere implementar un delay con latencia "1", para una cantidad de N bits, la mejor manera de implementarlo es mediante un Flip-Flop tipo D, el cual (de acuerdo a Xilinx) tiene un tiempo de configuración más bajo que un SRL16.

Una vez que se revisó completamente el diseño implementado, se mejoraron los problemas de sincronización agregando delays en el diseño Simulink de acuerdo a la información que entrego el reporte de sincronización.

3.1.3 Floorplanning y la creación de restricciones de área

A continuación se procedió a revisar la implementación de los recursos sobre el chip. En la Fig. 3-2 se presenta la ruta que siguen los datos desde el pin de entrada del ADC hasta el primer Tap del filtro FIR. Los Pines de Entrada de los ADC están ubicados en la columna ubicada a la izquierda de la ilustración.

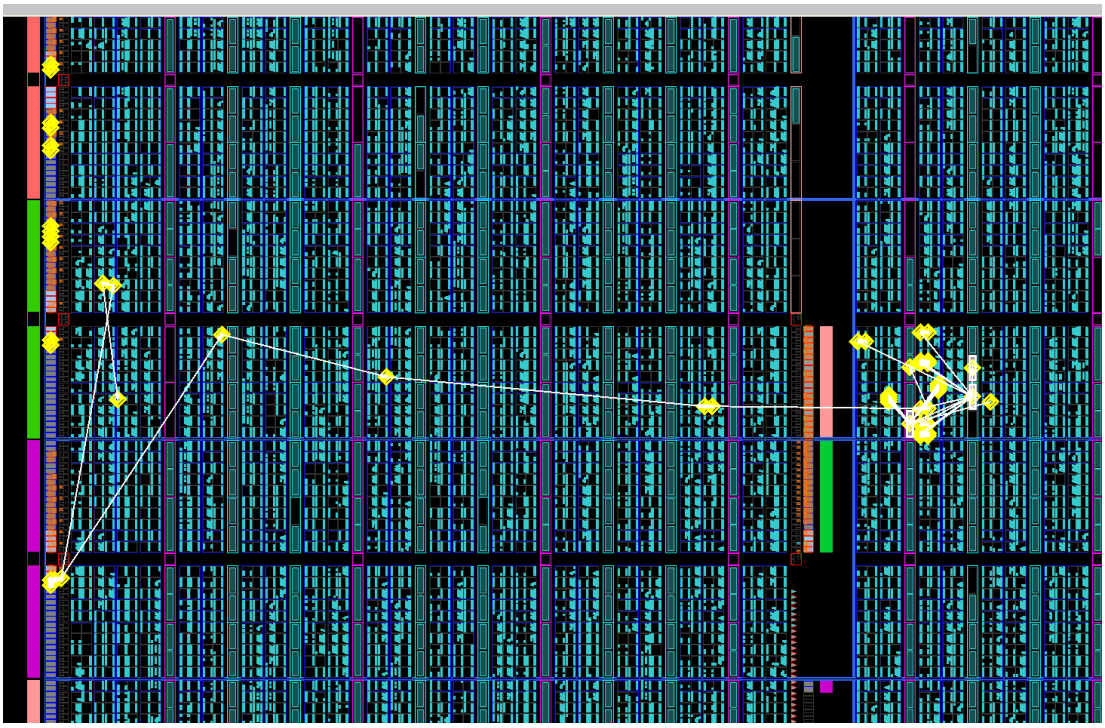

Fig. 3-2 En amarillo se aprecia el camino que siguen los datos desde el pin de entrada del ADC hasta el primer Tap del filtro FIR

La implementación que realizó el compilador System Generator no es óptima, ya que abarca una distancia muy grande dentro del chip. Se puede observar en la Fig. 3-2 que la implementación atraviesa un poco más de la mitad del ancho del chip. Sin embargo, para mejorar la sincronización se deben reducir estas distancias.

A continuación se procedió a realizar Floorplanning tal como aparece en la Fig. 3-3, donde se crea un P-Block para el filtro fir. La idea general consiste en restringir el área donde se desea que queden implementados recursos específicos.

En particular se hizo énfasis en reagrupar los bloques DSP48E y los bloques BRAM. Se movieron de manera manual estos bloques agrupándolos en áreas denominadas P-Blocks. En el caso del bloque del pfb_fir_real, que se destaca en la Fig. 3-3, se crea un P-Block a continuación de los pines de entrada de datos del ADC.

Como se puede apreciar a la derecha de la Fig. 3-3, que presenta el Gateware (representación del algoritmo de procesamiento de datos corriendo en un FPGA) implementado en Simulink. El ADC (bloque grande amarillo), va al principio del diseño, luego prosigue el bloque pfb_fir_real, es decir, para llevar a cabo la optimización se debe plasmar una representación fiel del Gateware sobre el chip FPGA.

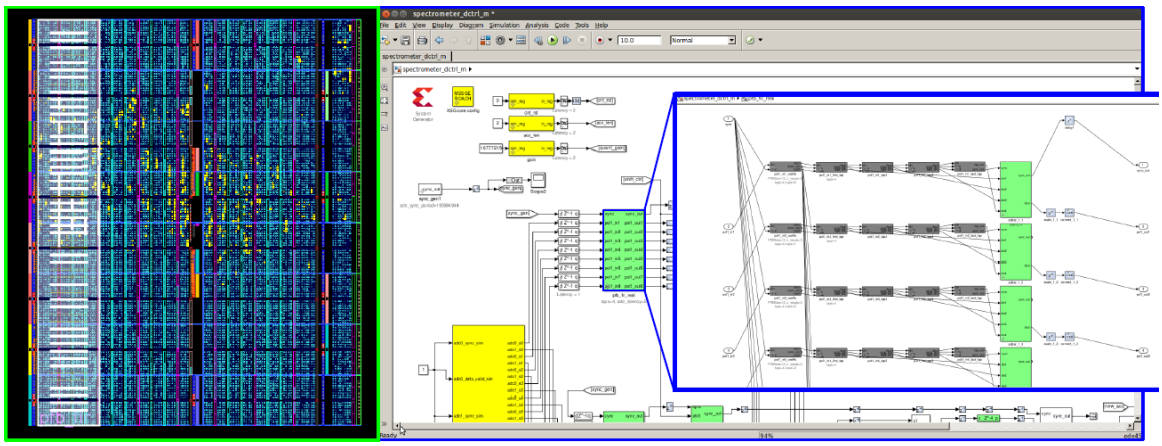

Fig. 3-3 pfb_fir_real floorplan

Este método debe ser acompañado con un estudio de sincronización entre los bloques que implementan los procesos críticos del diseño, de esta forma se puede mejorar la estrategia de optimización. La idea es generar una representación esquemática sobre el chip de sus bloques de mayor funcionalidad.

Como se señaló anteriormente se realiza la reubicación de los elementos cruciales para el procesamiento de datos sobre el chip. Por lo tanto los bloques que se reubican son los DSP48E y BRAMs, mientras que el compilador se encargara de reubicar la lógica restante.

Este proceso llamado Floorplanning es iterativo y busca optimizar el espacio y recursos del chip para mejorar la implementación de ubicación e interconexión de componentes que realiza el compilador System Generator durante el proceso de compilación. En los Anexos se presenta una guía para trabajar con esta herramienta.

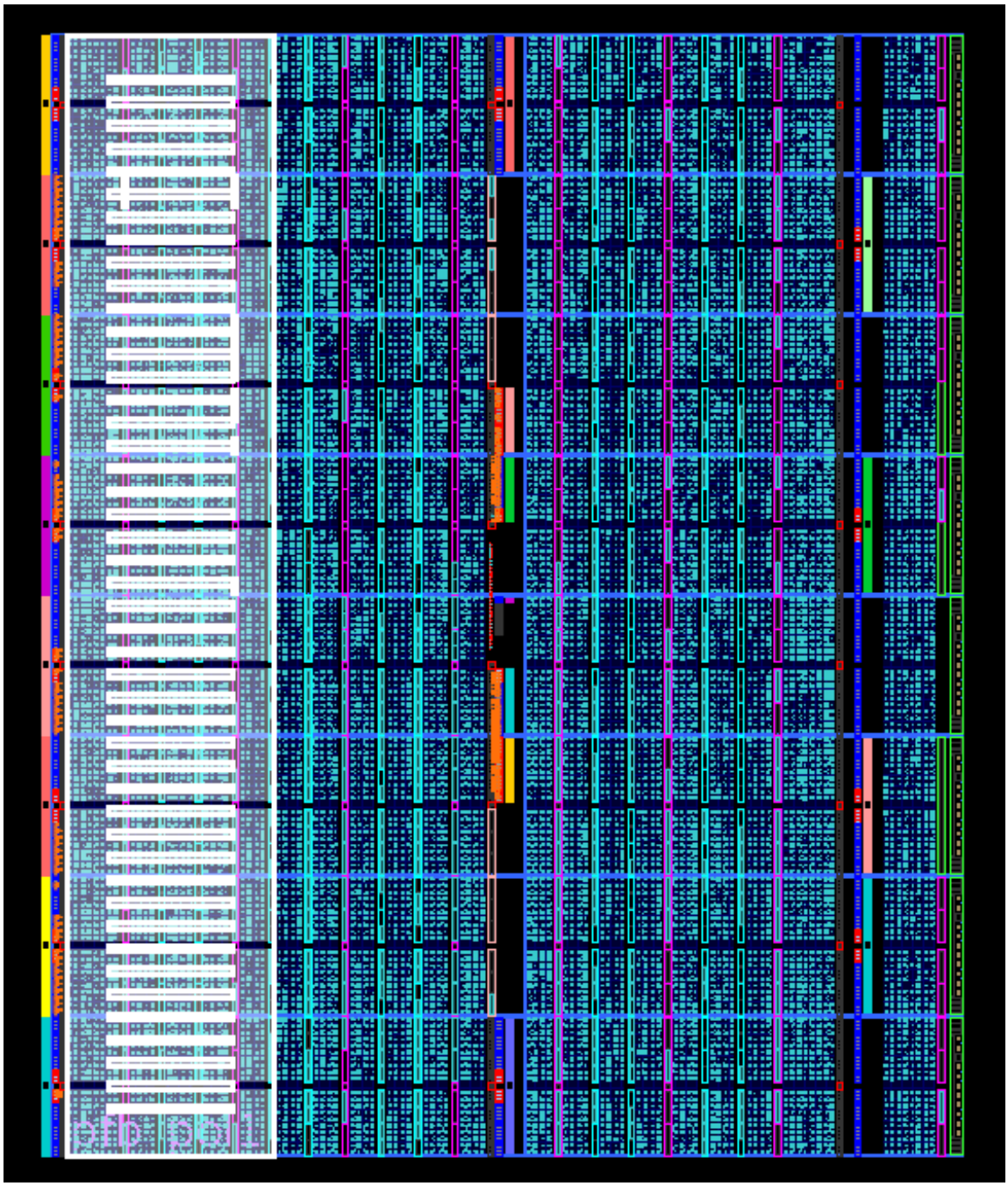

Fig. 3-4 Pblock del bloque pfb_fir_real

En la Fig. 3-4 se presenta el Floorplanning realizado para el bloque pfb_fir_real, donde se puede apreciar la nueva ubicación que tomaran los recursos de este bloque.

3.1.4 Estrategia de Floorplanning para el chip

A continuación se presenta el esquema que muestra la nueva ruta que seguirán los datos a lo largo del chip. Siguiendo la propagación de los datos que se presentan en el Gateware del diseño presentado en la Fig. 2-3.

En la Fig. 3-5 se puede observar la distribución de los PBlocks que se diseñaron para mejorar la sincronización de las rutas de procesamiento de datos.

Siguiendo la misma idea que se desarrolló para el bloque pfb_fir_real, se prosigue con restringir la ubicación del bloque Wideband_fft_real, descomponiéndola en sus etapas internas.

El Gateware cuenta con dos ramas idénticas, es decir, se tienen dos bloques pfb_fir_real, wideband_real_fft, ocho bloques power_dsp48e, dos quant0, ocho vacc y ocho Shared Brams. Sin embargo, el chip se subdivide en tres grandes subgrupos: ADC I/O con pfb_fir_real, etapas de la wideband_fft_real y etapa de acumulación que incluye principalmente los ocho bloques power_dsp48e, dos quant0, ocho vacc y ocho Shared Brams. Se escogió esta distribución siguiendo la ruta de procesamiento de datos desde los pines de entrada ADC, hasta los pines de salida que permiten leer las memorias.

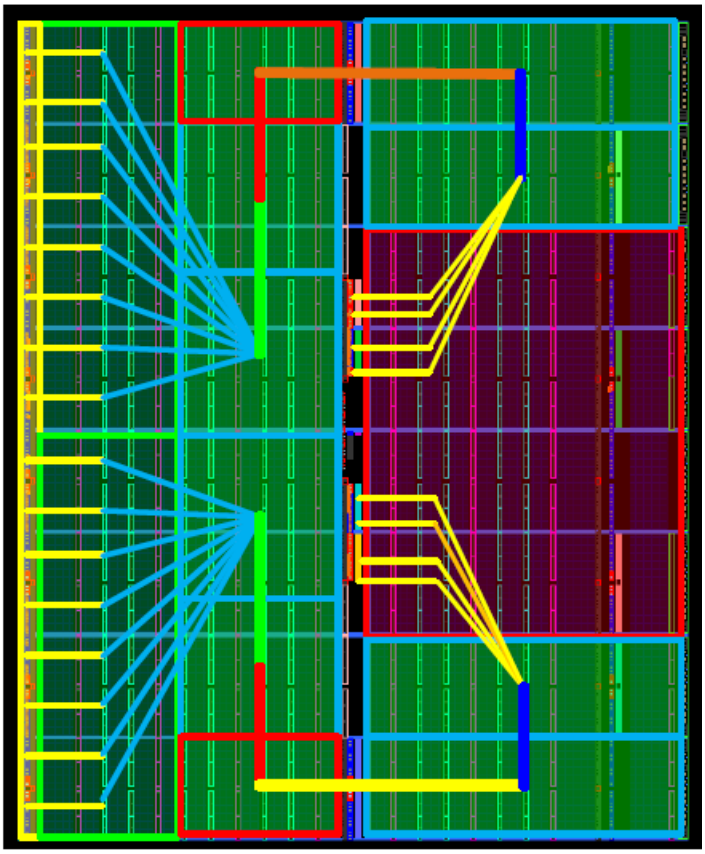

Fig. 3-5 Floorplan del Espectrómetro con separación de banda lateral

CAPÍTULO 4

RESULTADOS

En el presente capítulo se exhiben los resultados de la optimización del chip FPGA. Se amplió el ancho de banda (500 MHz) del receptor radioastronómico a 1000 MHz, mediante la utilización de la estrategia de Floorplanning descrita en el capítulo anterior.

4.1 Resultados del floorplan

En la Fig. 4-1se presenta un esquema del chip optimizado mediante las técnicas descritas de Floorplanning.

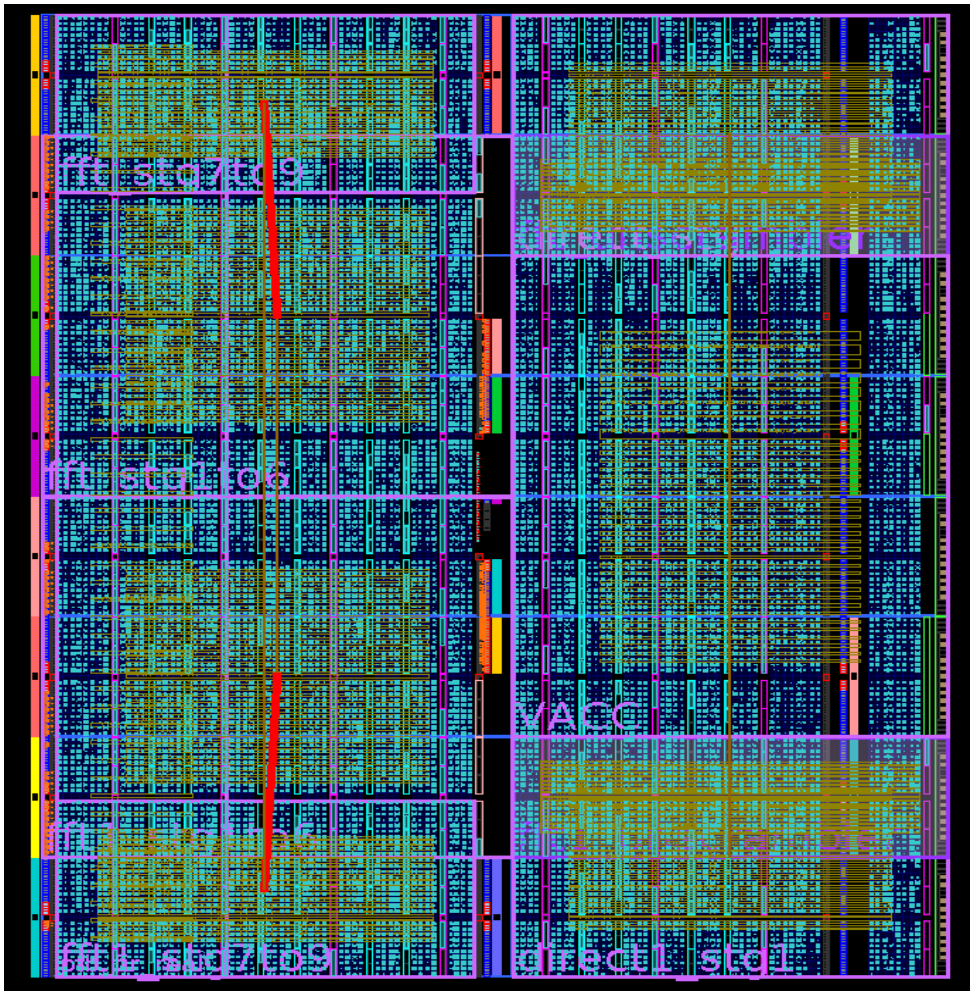

Fig. 4-1 Esquema del chip optimizado

Antes de aplicar los cambios al chip se debe modificar el archivo "system.ucf" con el fin de cargar las restricciones de áreas y en general todas las modificaciones que se añadieron. Se sugiere recompilar el diseño creado con Simulink esta vez con el clock para los ADCs de 1000 MHz, sin marcar la opción "*EDK/ISE/Bitgen*", antes de agregar el archivo de restricciones, tal como se sugiere en el Apendice-C.

Una vez que se realizó lo anterior se procedió a agregar las restricciones en el archivo /…/data/system.ucf. Luego se recompiló el diseño solo con la opción "*EDK/ISE/Bitgen*", obteniendo un diseño compilado con un periodo de reloj de 4 ns.

Tras la compilación se procede a revisar la implementación con el software de análisis PlanAhead y se obtiene el esquema del chip optimizado que se presenta en la Fig. 4-1.

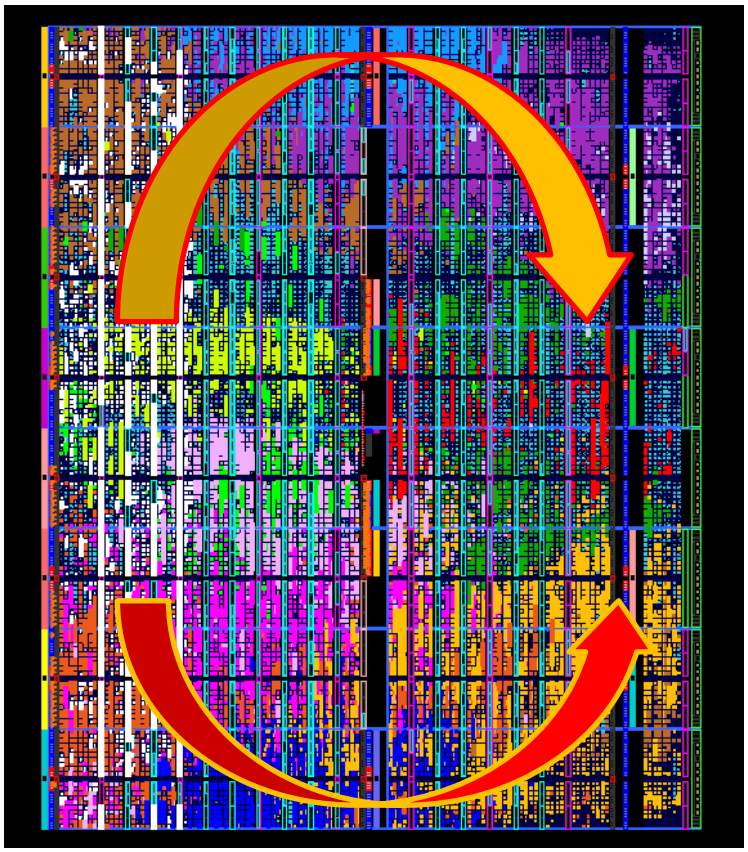

Fig. 4-2 Optimización lograda con el floorplan aplicado al chip

En la Fig. 4-2 se presenta la estrategia utilizada para propagar los datos muestreados por los ADCs desde el Oeste del chip, a través de la PFB de los cuales el filtro fir aparece en blanco, las etapas de la pipeline-FFT de colores en el sentido de la flecha, los bloques vectores complejos y los bloques de DSP48 para calcular la magnitud en verde, y finalmente hasta los bloques Vectores Acumuladores de color rojos en el extremo Este central del chip, donde finalmente serán leídos por el bus de datos epb del PowerPC, que tiene su banco de I/O pins en la parte central del chip.

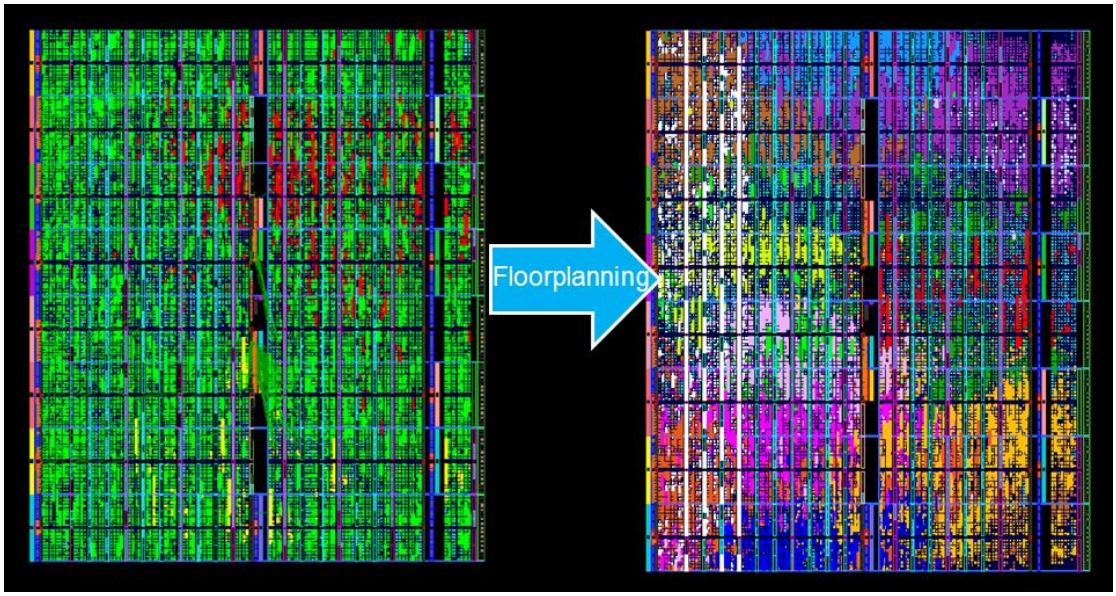

Fig. 4-3 Implementación de Xilinx vs diseño floorplanned con PlanAhead

En la Fig. 4-3 se presenta a la izquierda el diseño tras la compilación de las herramientas de Xilinx, donde se destaca en color verde la pipeline FFT, en rojo la PFB y en amarillo se aprecia el Vector Acumulador (parte inferior del chip), este diseño fue compilado como se señaló anteriormente con un requerimiento de frecuencia de reloj de 125MHz.

Luego de realizar el Floorplanning con la estrategia de generar áreas dentro del chip que interactúen entre sí, de acuerdo a la propagación de los datos, que provienen de los pines I/O del ADC que se sitúan en el lado oeste del chip. Finalmente los datos se propagan de Oeste-Este, hasta llegar a los Vectores Acumuladores marcados en rojo, permitiendo que el diseño pueda operar a una frecuencia de reloj de 250 MHz.

4.2 Evaluación del funcionamiento del sistema

En la Fig. 4-4 se muestra el espectro de una señal de radiofrecuencia con un ancho de banda de 1 GHz. Se observa la fundamental en 250 MHz y su contenido armónico.

Esta evaluación consiste en probar el sistema solo como espectrómetro, no se realiza el proceso de separación de banda digital, ya que el objetivo de este trabajo es la extensión del ancho de banda.

Se puede apreciar que el rango dinámico de ambos espectrómetros es de 50dB, ya que el tono fundamental tiene una magnitud de 80 dB, mientras que los armónicos más grandes poseen una magnitud de hasta 30 dB (la diferencia corresponde al rango dinámico).

Finalmente, en base a esta evidencia se comprueba que la optimización del diseño físico del FPGA permitió mejorar la sincronización entre sus bloques funcionales y a

consecuencia de esto, se logró extender el ancho de banda del receptor radioastronómico a 1000 MHz.

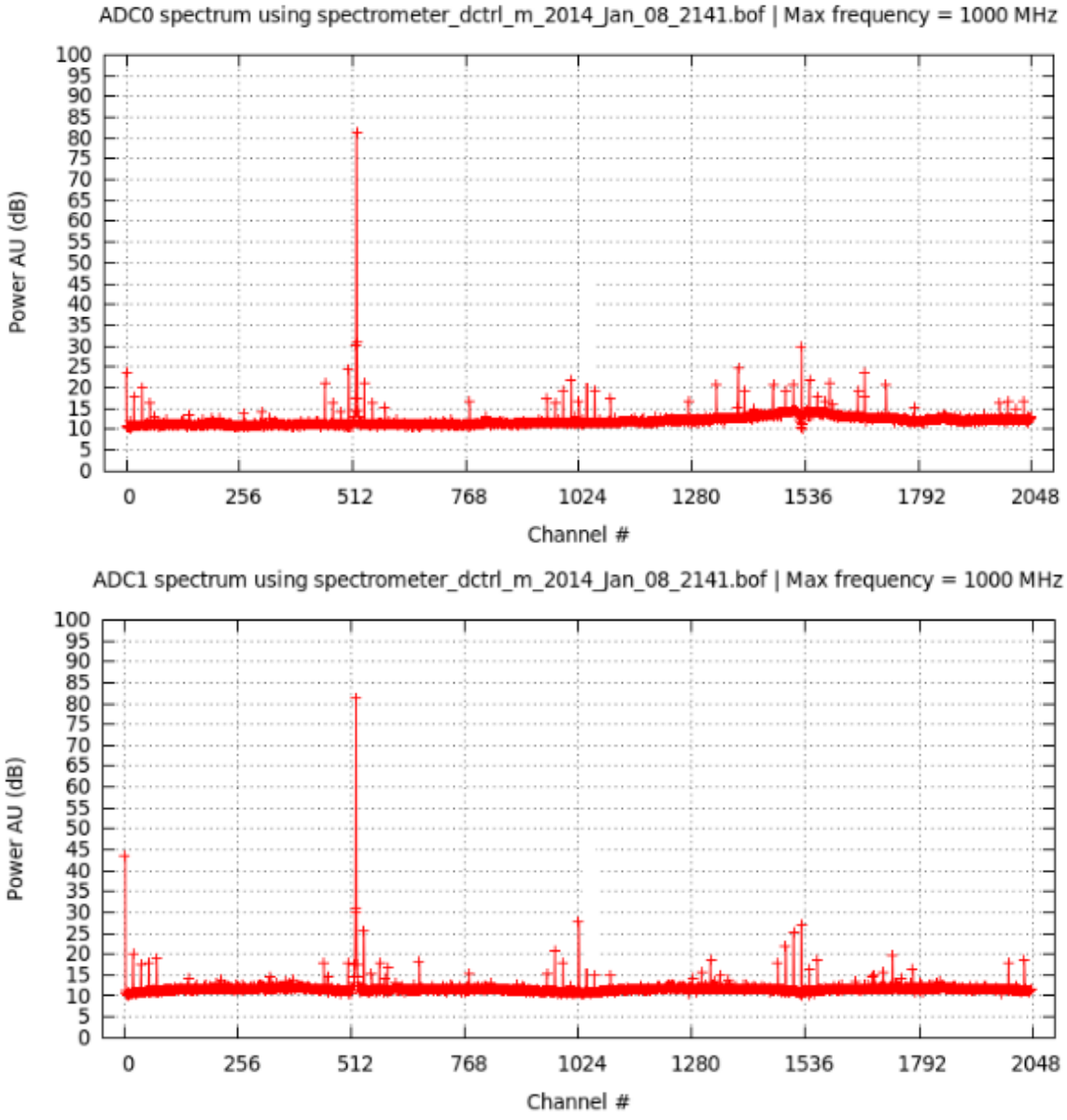

Fig. 4-4 Espectro de una señal de radiofrecuencia con un ancho de banda de 1 GHz

4.3 Trabajo futuro

El trabajo a futuro consiste en seguir aumentando el ancho de banda de la IF con un objetivo de 1.5 GHz por cada banda lateral. Se estudiará la posibilidad de incrementar el número de canales del espectrómetro, para mejorar la resolución espectral.

Además se debe llevar a cabo la optimización del modelo Calibrador para una IF de 1000 MHz, para poder cargar las constantes al diseño que fue optimizado en esta memoria. El siguiente paso consiste en hacer una medición de SRR con un ancho de banda de 1 GHz y obtener rechazos de banda sobre a 40 dB.

Por otro lado el back-end que fue optimizado será integrado al radiotelescopio mini [17], con el cual se realizarán estudios de mecanismos de emisión de radiación continua, con observaciones simultáneas de las bandas USB y LSB. Se provechará el gran ancho de banda de las IF del receptor, para observar las emisiones de radiación electromagnéticas que son generadas por transiciones electrónicas de monóxido de carbono ^{12}CO y ^{13}CO [3]. Para más detalle acerca de estos procesos consultar [18].

CONCLUSIONES

El objetivo de esta memoria se cumplió, puesto que se logró extender el ancho de banda original del receptor (500 MHz) a 1000 MHz, potenciando así el análisis espectral de la radiación electromagnética captada por la antena del radiotelescopio.

Sobre el uso de los *softwares* que pertenecen al *Toolflow Standard*, se observó que para hacer correr los diseños de *Simulink*, que utilizan más del ~90% de recursos de lógica sobre 250 MHz, es necesario restringir la ubicación de primitivas en la estructura del FPGA. Para realizar este trabajo es necesario utilizar complementariamente el software *Timing Analyzer* junto con *PlanAhead*, para guiar la reubicación de recursos del *chip*.

Para abordar el problema de optimización fue necesario entender cada etapa del proceso de diseño y desarrollo, dado que el trabajo implicaba modificar los parámetros del diseño en *Simulink* de manera reiterativa para maximizar el rendimiento. Además se necesitaba comprender cómo se propagarían los datos dentro del *chip*.

En la mayoría de los casos el reposicionamiento de los bloques de mayor funcionalidad, tales como, BRAMs y DSP48Es funciona correctamente, ya que PAR se desenvuelve adecuadamente posicionando la lógica complementaria. Se constató que al mover lógica se obtienen resultados subóptimos.

Dejar fijos los bloques BRAM y DSP48E en ubicaciones específicas no funcionó bien. Sin embargo, restringir un grupo de ellos en áreas específicas (PBlocks) si logró óptimos resultados, aunque esta técnica es limitada en cuanto a la cantidad de recursos que se le puede asignar a cada PBlock. Como máximo se recomienda llenar los PBocks hasta ~70%, pero esto va depender de la situación. En algunas ocasiones fue necesario llenarlos al 100% y funcionó correctamente.

Los PBlocks se ubican de Oeste-Este dentro del *chip*, a continuación de los pines de entrada del ADC. Cada PBlock agrupa una sección específica del diseño como la pfb_fir_real, etapas de la FFT y la etapa de cálculo de la magnitud al cuadrado y acumulación. Luego, siguiendo la propagación de los datos, se van agrupando los bloques en los respectivos PBlocks.

En resumen, se modifica el archivo de restricciones plasmando el diseño del Floorplan en líneas de código de acuerdo a la sintaxis que provee Xilinx. Una vez guardadas las restricciones, se compila y se verifica que el modelo funciona correctamente, por lo que se puede concluir que la optimización del diseño físico del FPGA permitió mejorar la sincronización entre sus bloques funcionales y a consecuencia de esto, se logró extender el ancho de banda del receptor radioastronómico a 1000 MHz.

La utilización de estos *softwares* para llevar a cabo optimizaciones conforman una metodología de trabajo, que busca de manera iterativa plasmar la estrategia del desarrollador. Cabe mencionar que los resultados obtenidos aseguran que la metodología empleada en esta memoria es adecuada para llevar a cabo optimizaciones, por lo que se puede usar como pauta de referencia para diseños futuros.

REFERENCIAS

[1] J. R. Fisher y M. A. Morgan, «Experiments with calibrated digital sideband separating downconversion,» *Publications of the Astronomical Society of the Pacific,* vol. 122, nº 889, pp. 326-335, 2010.

[2] Observatorio Astronómico Nacional, «Laboratorio de Ondas Milimétricas y Submilimétricas,» [En línea]. Available: http://www.das.uchile.cl.

[3] R. Finger, «A calibrated digital sideband separating spectrometer for radio astronomy applications,» *Publications of the Astronomical Society of the Pacific,* 2013.

[4] «Projects-CASPER,» [En línea]. Available: http://casper.berkeley.edu/wiki/Projects.

[5] NASA (original); SVG by Mysid., "Telescope," Wikipedia, The Free Encyclopedia. , 26 March 2014. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Telescope&oldid=601314057. [Accessed 2 April 2014].

[6] KOOI, «Advanced receiver implementations,» 2012, p. Chapter 8.

[7] Chennamangalam y Jayanth, «The Polyphase Filter Bank Technique,» CASPER Memo 41, 2011.

[8] J. Proakis y D. Manolakis, Digital Signal Processing, Third edition, PRENTICE-HALL INTERNATIONAL, INC, 1996.

[9] Xilinx Inc., «Virtex-5 FPGA Family,» [En línea]. Available: www.xilinx.com/support/documentation/virtex-5_data_sheets.htm.

[10] Xilinx, «UG193 Virtex-5 FPGA XtremeDSP Design Considerations,» 2010.

[11] S. Gowda, «A 3Gigahertz Bandwidth FPGA Spectrometer,» October 7, 2010.

[12] CASPER, "Mega Channel Spectrometer," [Online]. Available: https://casper.berkeley.edu/wiki/Toolflow.

[13] CASPER, «Hardware ROACH,» [En línea]. Available: https://casper.berkeley.edu/wiki/ROACH.

[14] H. K.-H. So, «BORPH: An Operating System for FPGA-Based Reconfigurable Computers,» UNIVERSITY OF CALIFORNIA, BERKELEY, 2010.

[15] National Semiconductor, «ADC083000: 8-Bit, 3 GSPS, High Performance, Low Power A/D Converter from the PowerWise Family,» [En línea]. Available: http://www.national.com/pf/AD/ADC083000.html.

[16] Xilinx Inc, «Design Analysis and Floorplanning Tutorial, "PlanAhead Design Tool", UG676 (v14.1),» May 8, 2012.

[17] mm-wave Laboratory, "The 1.2m Southern Millimeter Wave Telescope is a 85-115GHz," Observatorio Astronómico Nacional Cerro Calán.

[18] R. y. H. Wilson, «Chapter 10: Emission Mechanisms of Continuous Radiation.,» de *Tools of Radioastronomy, 5th Ed.*, Springer, 2009, pp. 239 - 274.

[19] K. R. S. H. T.L. Wilson, Tools of Radio Astronomy, 5th Edition, Springer, 2010.

[20] CASPER, "BORPH," [Online]. Available: http://bee2.eecs.berkeley.edu/wiki/Bee2OperatingSystem.html.

[21] CASPER, «Nyquist Sampling,» [En línea]. Available: https://casper.berkeley.edu/astrobaki/index.php/Nyquist_Sampling.

[22] Xilinx Inc., [En línea]. Disponible: http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_4/cgd.pdf.

APÉNDICE A

TUTORIAL 1: OPTIMIZACION DE VELOCIDAD CON PLANAHEAD

APÉNDICE A

TUTORIAL 1: OPTIMIZACIÓN DE VELOCIDAD CON PLANAHEAD.

Contenidos:

- Introducción a la herramienta PlanAhead.
- Importar diseño a PlanAhead.
- Floorplanning.
- Estrategia general de la ubicación de Bloques.

A.1 Introducción a la herramienta PlanAhead.

A continuación se presenta una guía para trabajar con la herramienta de optimización de diseños de hardware de FPGA PlanAhead 14.5. Cuando por requerimiento de diseño es necesario que el FPGA tenga un *clock* superior a 250 MHz, se debe utilizar PlanAhead.

Primero que todo se debe tener bien definido el diseño a optimizar y compilado a la máxima velocidad de reloj posible. Se recomienda usar casper_xps para compilar los diseños a 200 MHz. No importa si falla durante PAR (placing and routing). Si su diseño no logra compilar a esta velocidad, es muy probable que haya algo erróneo en su diseño que deberá ser reparado antes de proceder a usar PlanAhead. Analice su diseño con el software Timing Analyzer.

Para ejecutar Timing Analyzer ejecute el siguiente comando en el prompt de matlab: `>>dos('timingan')`

Muchos problemas de sincronización del FPGA pueden ser solucionados, mediante la introducción de delays en las rutas críticas para igualar las holguras de los netlists del diseño. Se le recomienda al desarrollador tener en mente que los resultados de las modificaciones salen a la luz tras una nueva compilación, por lo que vale la pena dedicar más tiempo a planificar una estrategia que probar a ensayo y error hasta obtener buenos resultados.

Para elaborar una estrategia de optimización se debe seguir una idea similar a la descrita en la memoria.

Nota: Si su diseño utiliza más del 70% de los siguientes recursos: slices, slice registers y LUT flip flops, va tener problemas para alcanzar las requerimientos de timing.

A.2 Importar diseño a PlanAhead.

A continuación se presenta una guía con los pasos preliminares para trabajar con PlanAhead. Primero que todo abra una consola de Linux, ctrl+alt+t y registrarse como administrador tal como se presenta en la FIG. A-1. Para registrarse como administrador ingrese el siguiente comando e ingrese contraseña.: `~$ sudo bash`

El nombre de usuario debe cambiar a "root", y además debe aparecer el signo "#" que indica que se ingresó como súper usuario, tal como se puede apreciar en la FIG. A-1.

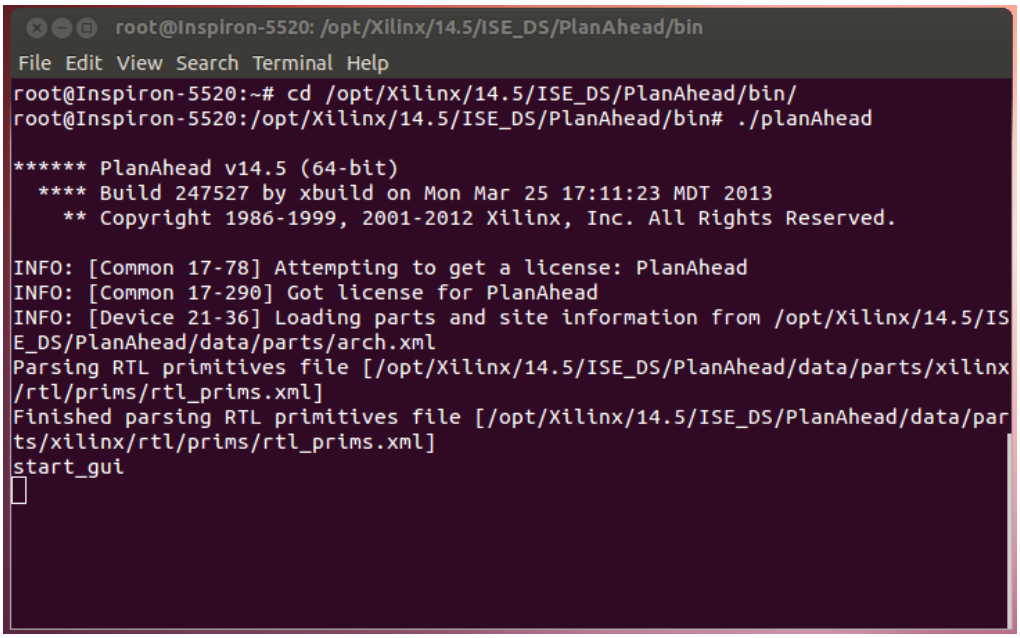

FIG. A-1 Terminal de Ubuntu 12.04 LTS.

Segundo, ingrese a la siguiente ruta para poder ejecutar PlanAhead, luego presione "Enter":

```
~# cd /.../Xilinx/14.5/ISE_DS/PlanAhead/bin/
```

Una vez que se ingresó a /.../bin/ inicialice Planahead con el siguiente comando tal como se presenta en la figura A-1. Luego se desplegara Planahead v14.5. Con una interfaz como se presenta en la Fig. A-2.

```
~#./planAhead
```

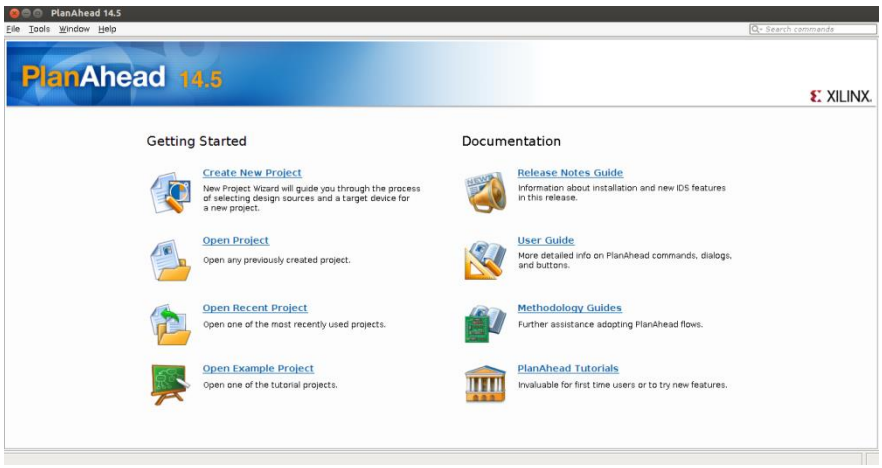

Fig. A-2 Interfaz de usuario PlanAhead 14.5

Para importar el *netlist* y restricciones generadas tras la compilación, siga los siguientes pasos, presione en *File* y luego *New Project*.

(1) Haga *click* en ***Create a New Project*** y luego presione ***Next***, tal como se ve en la Fig. A-3.

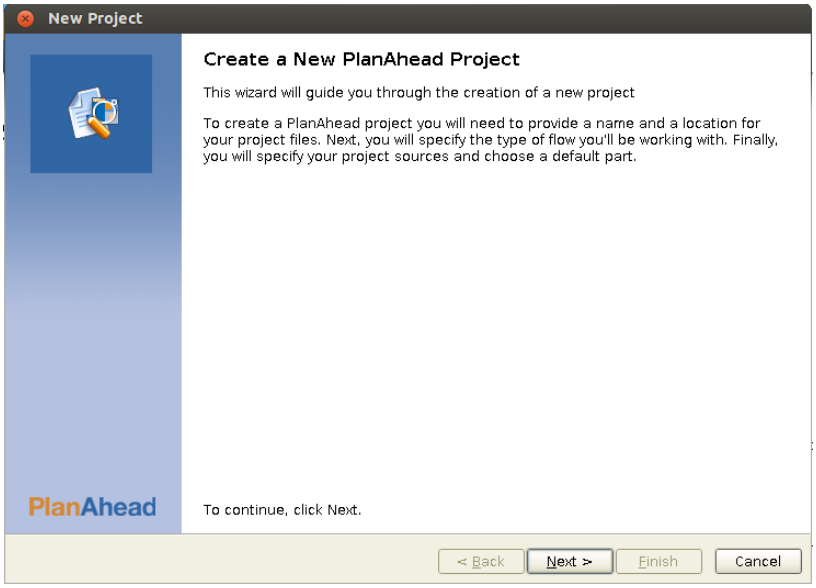

Fig. A-3 Creando un proyecto nuevo

(2) Project Name:

Escoja un nombre y seleccione la ubicación del nuevo proyecto. Escoja el nombre proyecto_1 en *Project name*. Luego escoja un directorio donde quiera ubicar el proyecto, tal como se presenta en la Fig. A-4.

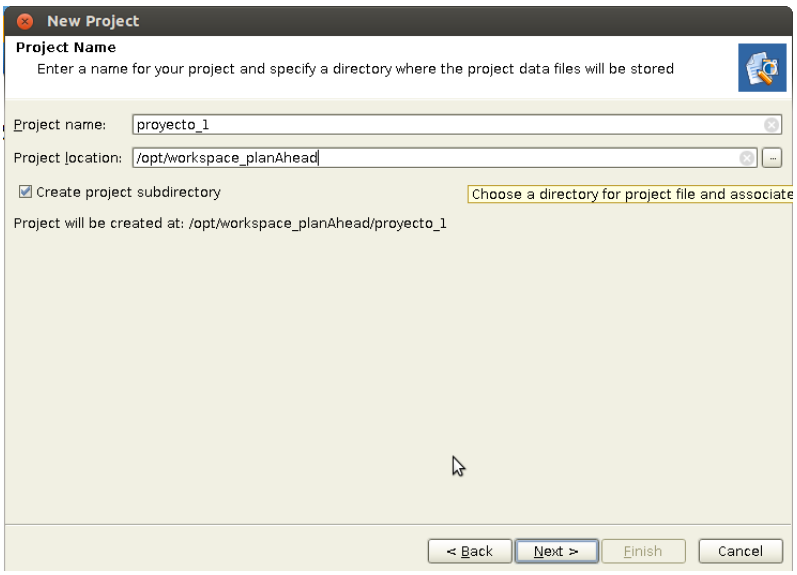

Fig. A-4 Nombre del proyecto

(3) Tipo de proyecto:

Hay dos opciones, Post-Synthesis project o Import ISE Place & Route results. El primer caso corresponde a la primera parte "*Xilinx System Generator"***.** Esta parte genera un equivalente en VHDL al modelo y luego lo sintetiza, esta parte es muy lenta. En cambio la segunda opción permite incluir más etapas como Place and Route y la fase de Mapping y además permite incluir el archivo de restricciones el cual finalmente será utilizado para tener todas estas etapas incluidas seleccione en CASPER XPS todas las etapas de la compilación del *Toolflow* estandar, tal como se aprecia en la Fig. A-5.

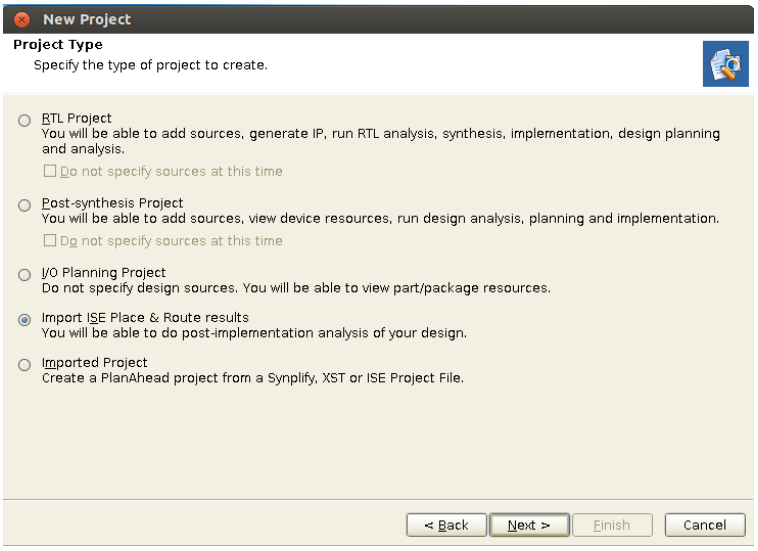

Fig. A-5 Seleccione el tipo de proyecto de acuerdo al proceso de compilación

Seleccione "Import ISE Place and Route results (EDIF or NGC) netlist", tal como aparece en la Fig. A-5.

(4) Importando el Netlist del diseño:

Ubique el archivo system.ngc y luego haga *click* en el botón Next. En la ruta que se describe a continuación:

```
$YOUR_SIMULINK_PROJECT_PATH/XPS_ROACH_base/implementation/system.ngc
```

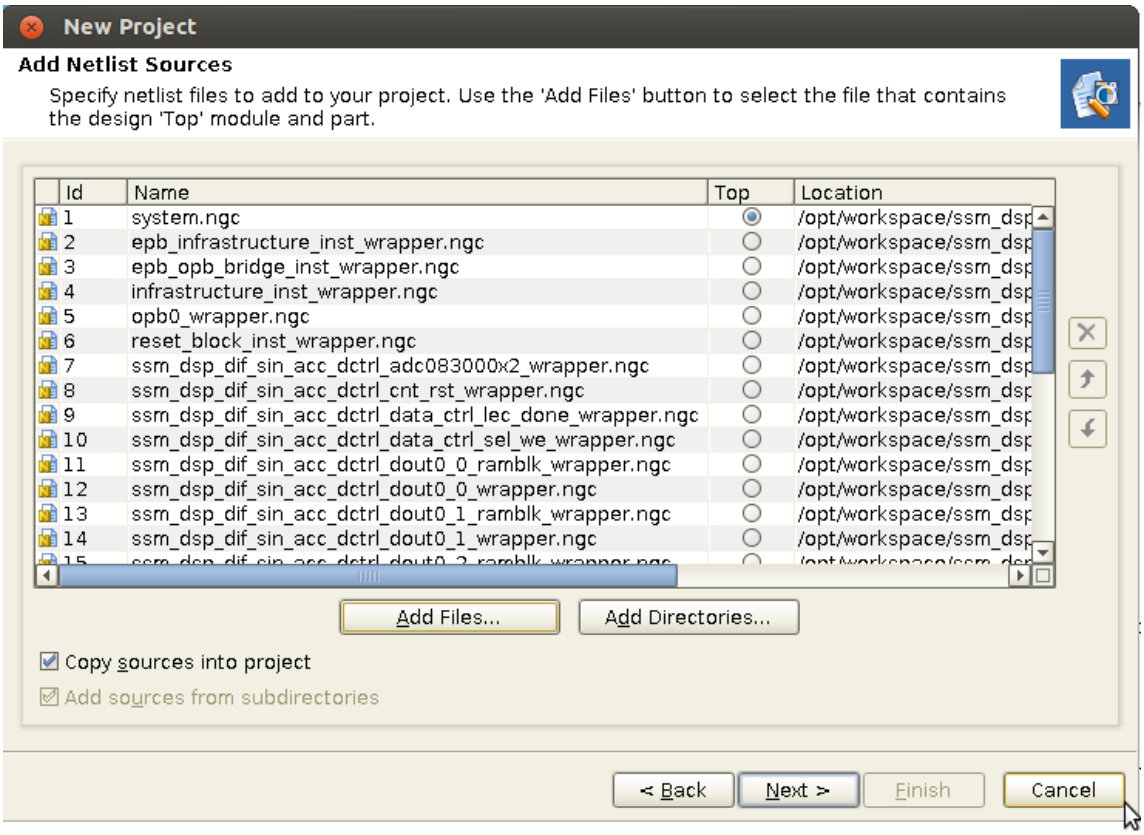

Fig. A-6 Agregando los archivos Netlists del diseño

Luego, seleccione como "Top", el archivo system.ngc como se muestra en la Fig. A-6, ya que este es el archivo principal (Top o principal) que contiene todo el netlist del diseño, los demás archivos de extensión ngc contienen los sub-netlists del diseño, en el archivo Top se llaman todos esos sub-netlists.

(5) Escoja el modelo del chip y el nombre de Floorplan:

Esto debería ocurrir automáticamente, verificar que aparezcan los siguientes datos según corresponda, luego haga *click* en next, tal como aparece en la Fig. A-7.

Product Family: Virtex5
Choose Part: xc5vsx95tff1136-1.

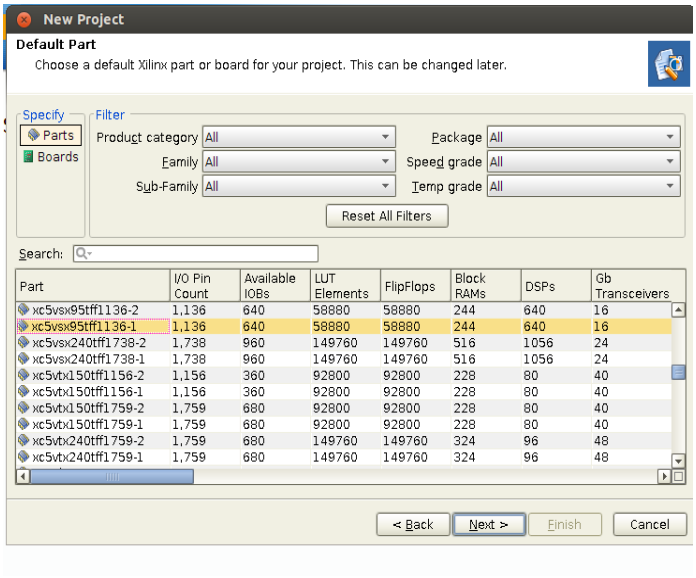

Fig. A-7 Seleccionando modelo del FPGA

(6) Importando archivo de restricciones system.ucf:

Ubique el archivo que contiene las restricciones de la compilación, system.ucf, en la ruta que se describe a continuación:

```
$YOUR_SIMULINK_PROJECT_PATH/XPS_ROACH_base/implementation/system.ucf
```

Luego haga *click* en *Next*, tal como se muestra en la Fig. A-8.

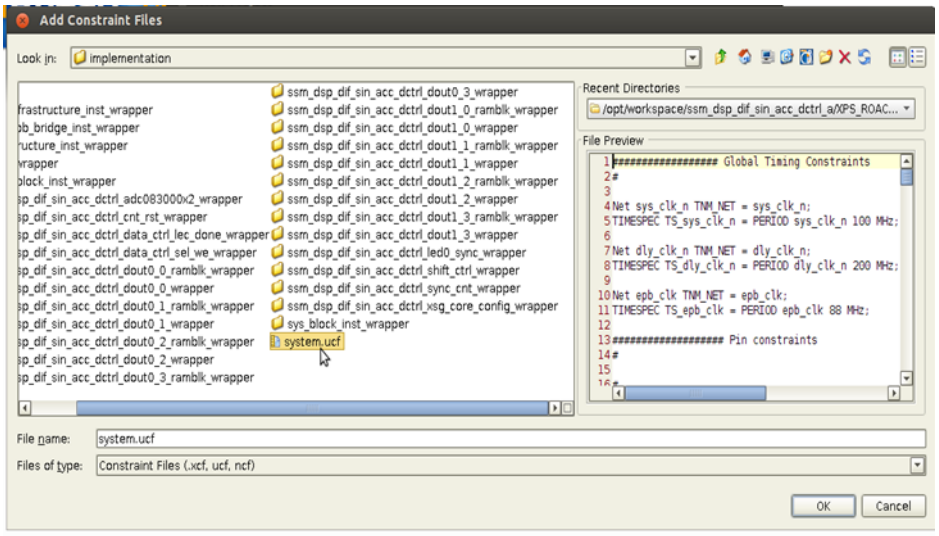

Fig. A-8 Archivo de restricciones

Es importante ubicar el archivo de restricciones y hacer una copia una vez finalizada la compilación, ya que este archivo será utilizado más tarde para guardar los cambios que se desean aplicar al diseño, aquí en este archivo de extensión ".ucf" es donde

se deben agregar restricciones de acuerdo a las estrategia *Floorplanning*, definición de áreas, y reubicación manual de recursos del FPGA. Una vez seleccionado el archivo de restricciones, aparecerá una ventana como se presenta en la Fig. A-9, si no desea agregar más archivos de restricciones, haga *click* para continuar, en caso contrario agregue un nuevo archivo ".ucf" haciendo *click* en "Add Files…".

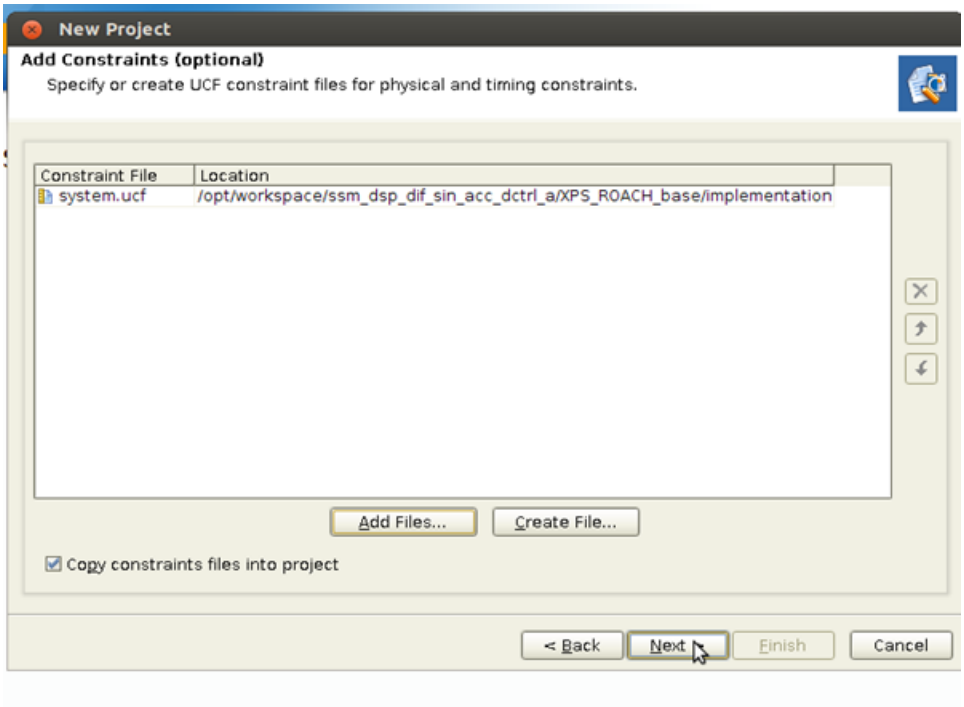

Fig. A-9 Archivo system.ucf

En la Fig. A-10, se agregan los archivos de extensión NCD que corresponde a los resultados del algoritmo de *Placement*, el cual consiste en hacer una asignación de recursos de acuerdo al diseño con el que se esté trabajando, una vez asignados los recursos, el algoritmo de *Placement* se encarga de ubicarlos en algún lugar dentro del chip, luego finalmente se realiza el proceso donde se unen todos estos recursos tratando de minimizar las distancias entre los diferentes bloques del FPGA.

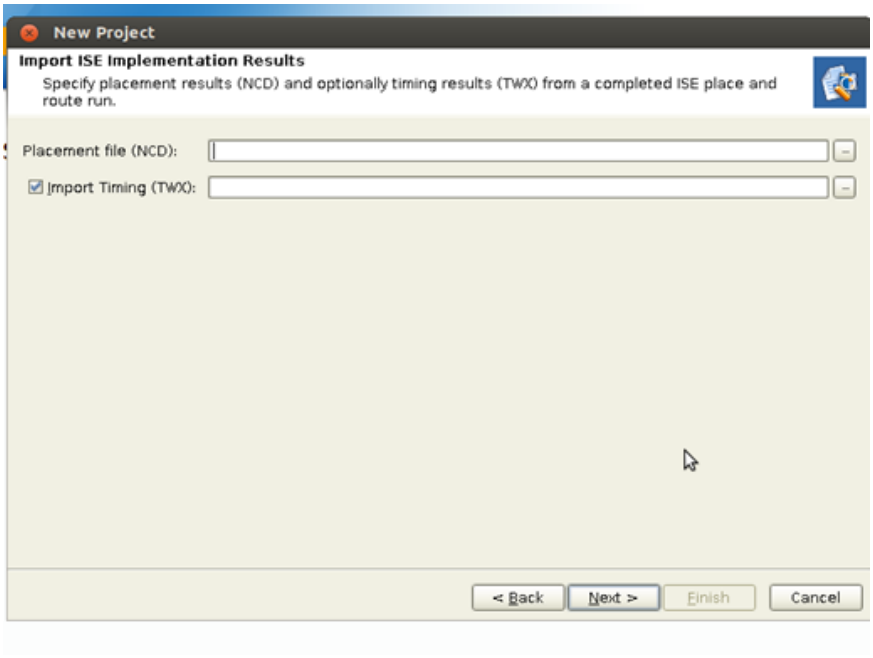

Fig. A-10 Agregar archivo de Placement y Timing Report

En la Fig. A-11 se muestra seleccionado el archivo system.ncd con los resultados del P&R y en la Fig. A-12 se presenta el resultado del analisis de tiempo de las señales de reloj de los datos.

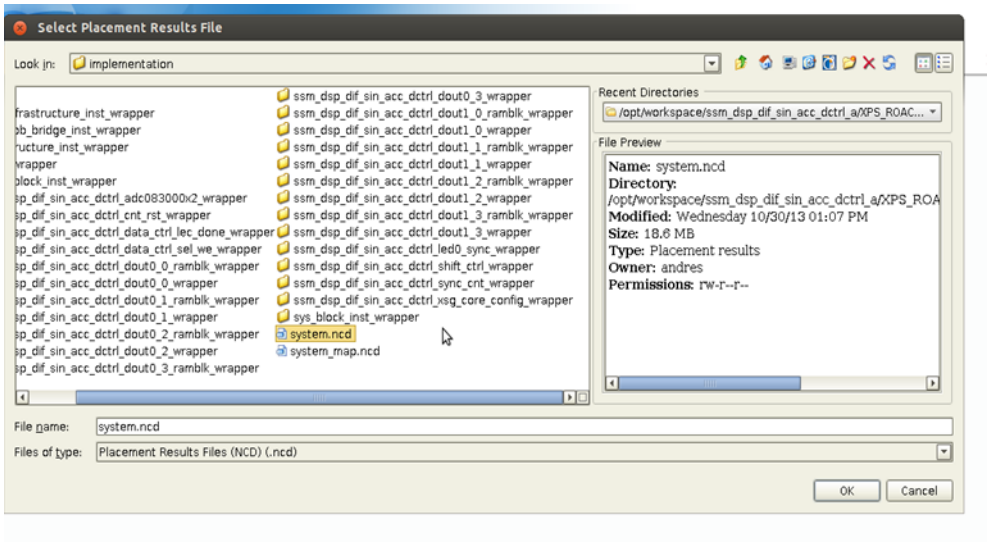

Fig. A-11 Archivo system.twx

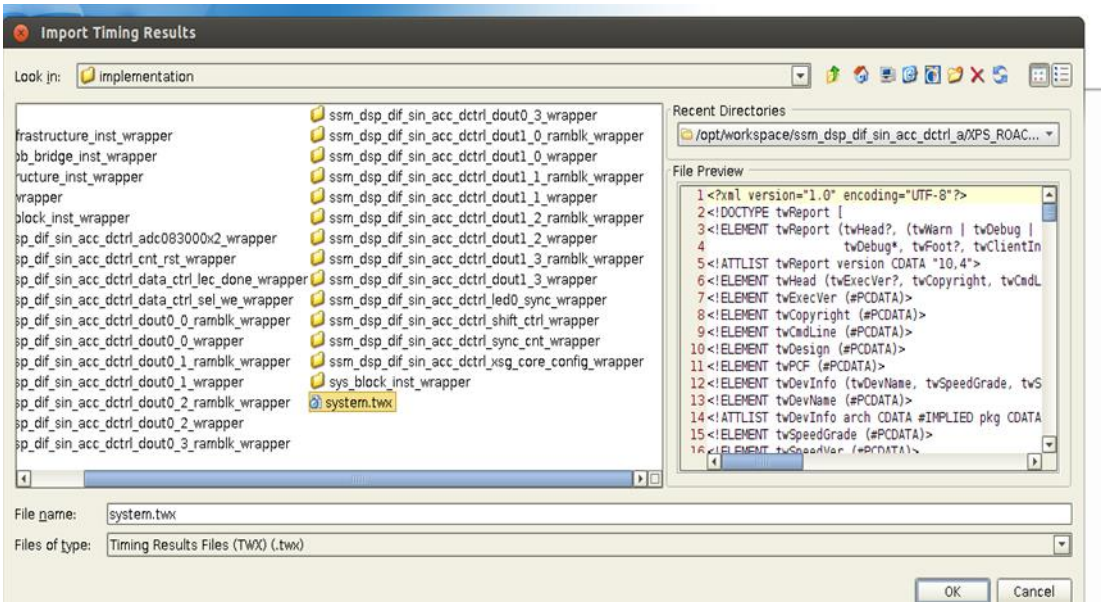

Fig. A-12 Archivo system.ncd

(7) Resumen del Nuevo Proyecto:

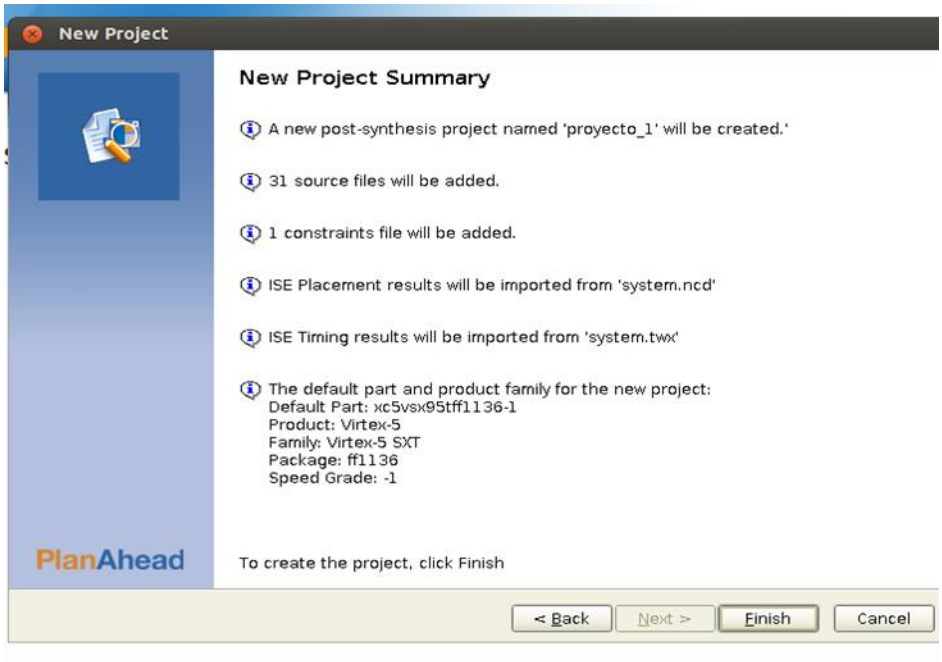

Fig. A-13 Resumen del Proyecto creado con PlanAhead 14.5

Una vez finalizada la creación del proyecto, aparecerá una ventana que presenta un resumen del proyecto, como el que aparece en la Fig. A-13. Luego, presione *Finish* para continuar.

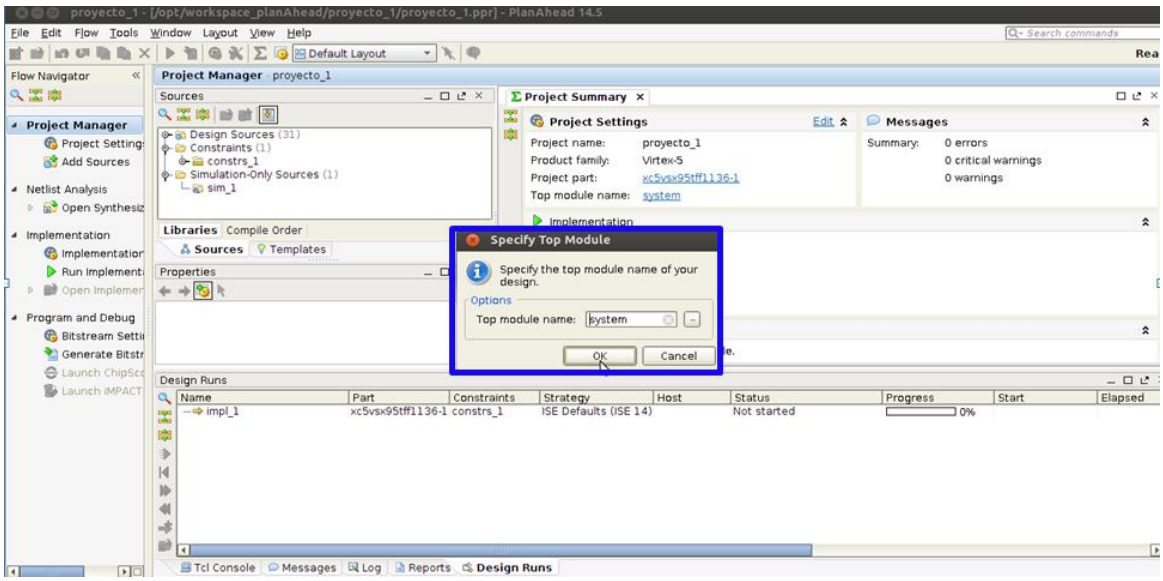

Fig. A-14 Interfaz gráfica de PlanAhead 14.5

Una vez creado el proyecto de PlanAhead, se abrirá una interfaz gráfica como la que aparece en la Fig. A-13, se desplegara una ventana para confirmar el archive top, que debe llamarse system, tal como aparece en la Fig. A-14.

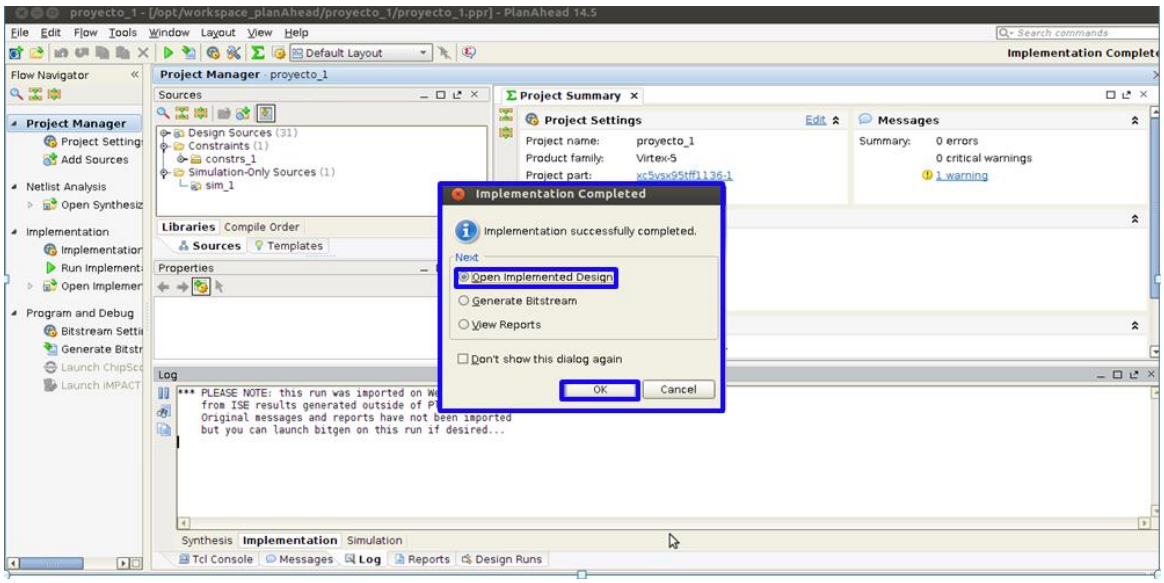

Fig. A-15 Abrir diseño implementado

Luego de esperar varios minutos mientras se cargan los archivos en PlanAhead, aparecerá un recuadro con un mensaje de que la implementación fue completada satisfactoriamente, tal como se presenta en la Fig. A-15. Aquí se pueden proseguir tres opciones, pero no haga cambios mantenga la opción por defecto.

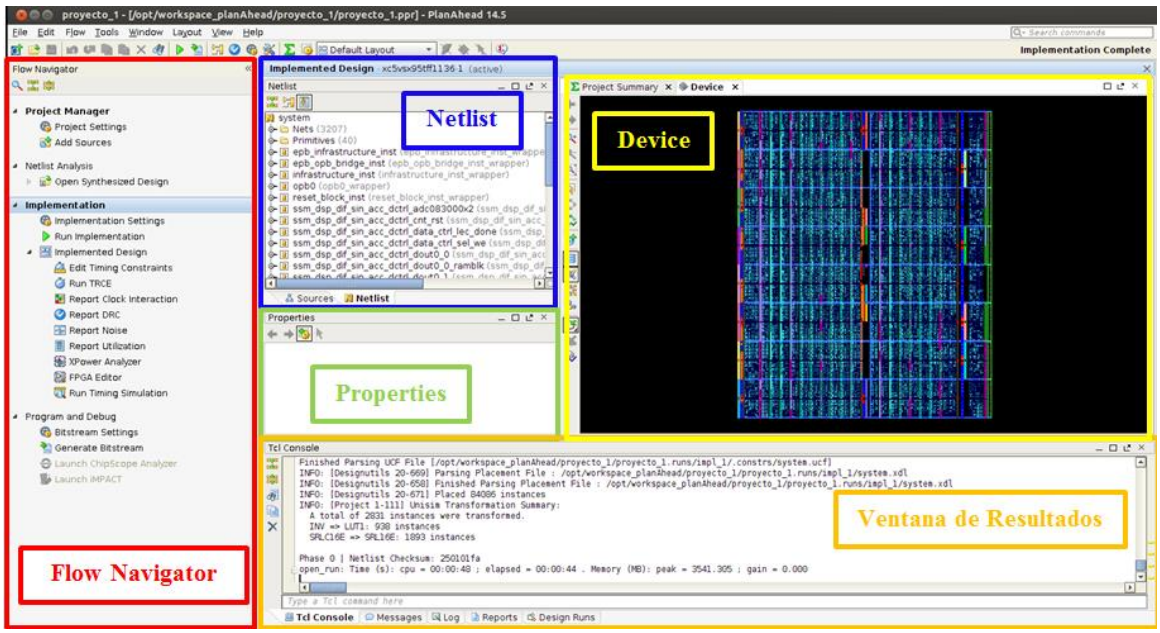

Fig. A-176 Entorno de trabajo de PlanAhead

A continuación se prosigue con los pasos para llevar a cabo una optimización de hardware, el lector podrá dirigirse a los tutoriales de Planahead si requiere mas información sobre esta herramienta de software.

En la Fig. A-16 se presenta el ambiente de trabajo de PlanAhead, que se subdivide de izquierda a derecha en 5 subventanas. A la izquierda de la figura se encuentra una ventana llamada "*Flow Navigator*" que permite abrir el diseño implementado, además permite agregar restricciones de tiempo, en la subsección "*Edit Timing Constraints*", las demás opciones por lo general no fueron utilizadas, mayoritariamente son para realizar análisis y simulaciones del diseño, si requiere mas información acerca de estas herramientas de análisis diríjase a la guía "*Design Analysis and Floorplanning Tutorial*", que puede ser descargado directamente desde la página web de Xilinx.

A.3 Floorplanning.

1. Presione Ctrl+F, que corresponde al comando *"Find"*, se desplegara una ventana que deberá ser configurada como se sugiere a continuación:

   i) Haga *click* en "More" y se sugiere realizar búsquedas por nombre y tipo de componente del FPGA.

   ii) Para el tipo, seleccione "Block Memory" para buscar BRAMs o "Block Arithmetic" para DSP48s.

   iii) Para referirse al nombre, busque algún nombre que le interese de los bloques que se encuentran en el diseño de Simulink. Por ejemplo: pfb_fir_real.

   iv) *Click* "OK".

Repita este proceso para ubicar los bloques que les interese, luego haga *click* sobre alguno de ellos, y luego para seleccionar todos presione en el teclado Ctrl+a. En la ventana *Device* de la Fig. A-16, se ven que los bloques que han sido ubicados tienen color celeste, al seleccionarlos estos se tornan de color blanco. Se recomienda trabajar con dos monitores, ya que los nombres de los bloques poseen dirección y nombres, y se requiere trabajar con una vista clara del dispositivo.

Presione con el botón izquierdo del mouse sobre los resultados de la búsqueda en la ventana llamada “Find Results” (DSP48 y BRAMs) que fueron encontrados y cambie el color con la opción highlight blocks, la idea es obtener un diagrama tal como se presenta en la Fig. A-18, con esto se podrá ver claramente donde están ubicados los bloques del diseño de Simulink, como la fft_wideband_real, pfb_fir_real, VCM, quant0, vacc, del diseño a optimizar.

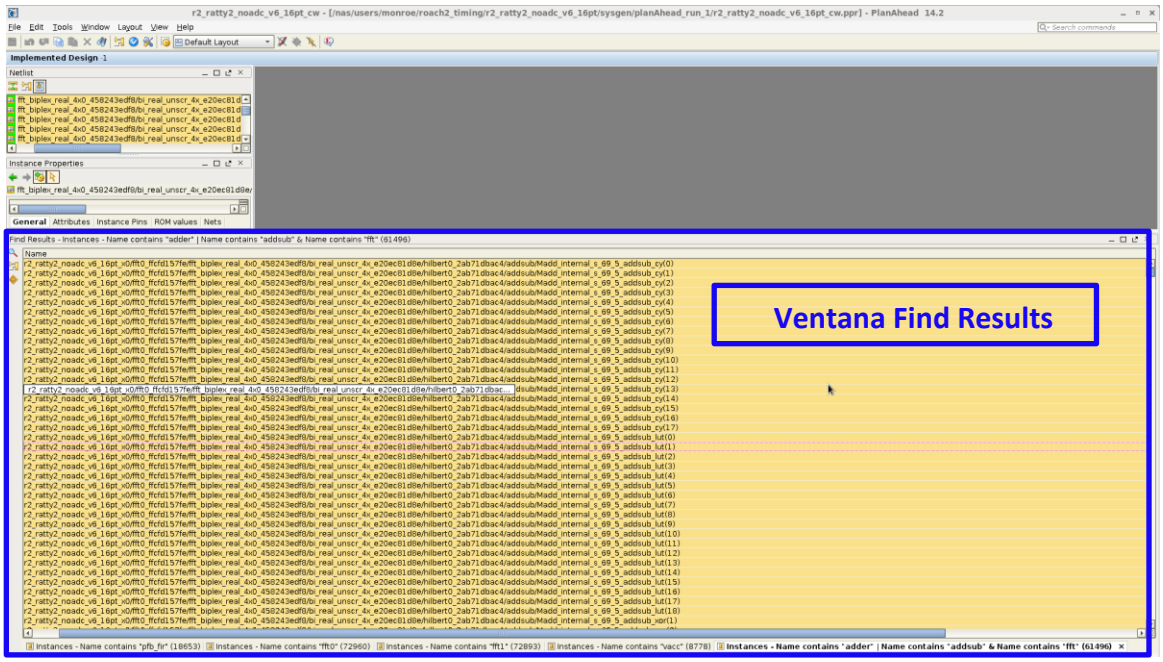

Fig. A-18 Ctrl+f para buscar los bloques del diseño implementados

2. En la ventana "Find Results" que aparece a la izquierda inferior de la Fig. A-18, busque algún bloque de del diseño realizado en Simulink, como por ejemplo “pfb_fir_real”, luego haga *Click* sobre algún bloque encontrado y finalmente presione Ctrl+a, para seleccionar simultáneamente todos los bloques encontrados.

   i) Seleccione algunas primitivas.

   ii) *Click*-Derecho sobre estas.

   iii) *Click* en la opción "Draw Pblock".

3. En la ventana “Device”, que muestra una ilustración del chip como el que aparece en la Fig. A-19, dibuje un Pblock rectangular en una nueva región donde serán reubicados.

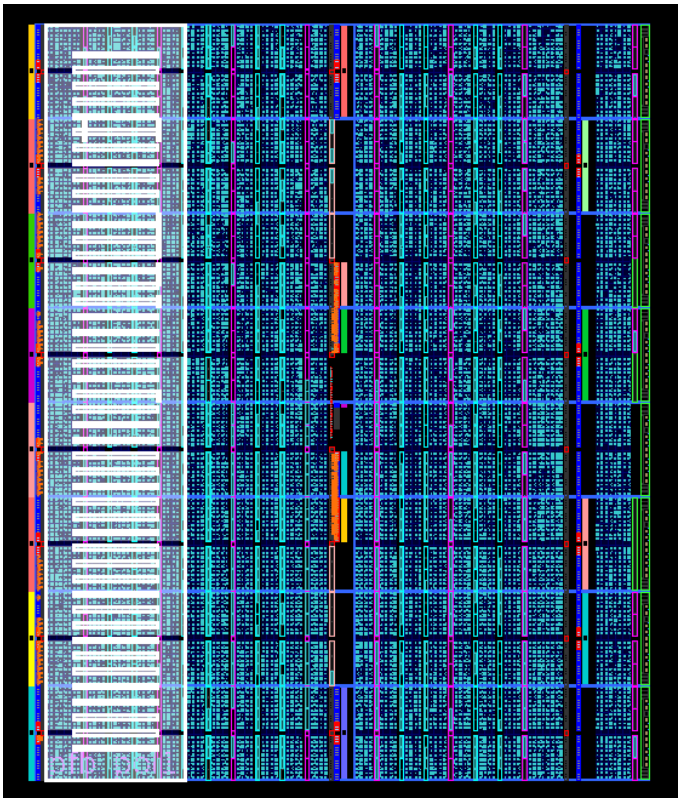

Fig. A-19 Ventana Device de PlanAhead. Pblock del bloque pfb_fir_real.

4. A continuación se desplegara una ventanita llamada "New Pblock".

   i) Escoja un nombre para el PBlock.

   ii) Seleccione el tipo de primitivas que a usted le gustaría restringir dentro del Pblock.

   iii) Verificar que la opción "*Assign selected instances*" está marcada.

   iv) *Click* "OK".

5. En la ventana "*Pblock Properties*":

   a) Seleccione "*Statistics*" en el menú.

   b) Verificar que su Pblock no exceda el 70% de utilización de los recursos.

6. Repita estos procesos tantas veces como sea necesario.

En la Fig. A-20, se presenta una representación esquemática de la reubicación realizada con la idea de propagar los datos desde los bancos de pines de entrada y salida ubicada al Oeste del chip, hasta el lado Este, donde se encuentran los Vectores Acumuladores y las memorias RAM, que son cargadas con los resultados finales y que permiten la descarga del espectro vía Ethernet.

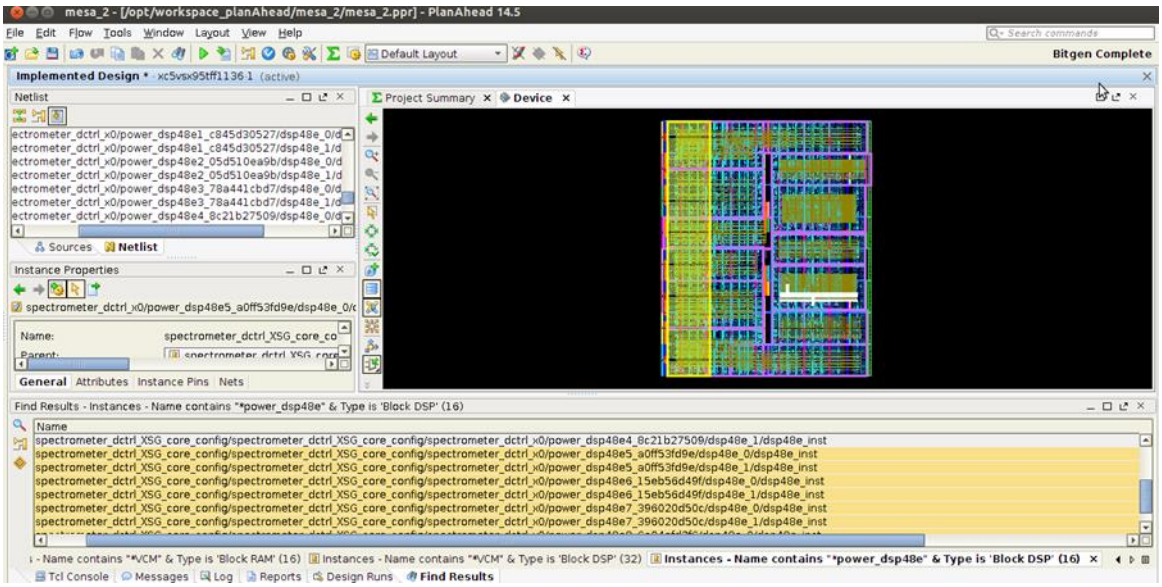

Fig. A-20 Floorplanning del diseño implementado en el chip Virtex-5

A.4 Estrategia general de la ubicación de Bloques.

Lamentablemente no hay una receta para que un diseño alcance los requerimientos de altas velocidades. Como se ha mencionado en reiteradas ocasiones durante este trabajo, este es un proceso iterativo que puede desarrollarse de manera intuitiva, y mediante práctica se puede llevar a cabo la confección de alguna estrategia de optimización. A continuación se describe de manera general la estrategia utilizada.

Cuando PAR falle, revise los problemas de sincronización sobre el chip, mediante un software llamado Timing Analyzer y use los resultados para guiar la nueva ubicación (placement) de los bloques. Para expandir el ancho de banda fue necesario hacer un reposicionamiento manual de los recursos críticos y generar áreas dentro del chip FPGA. Específicamente se reubicaron los elementos cruciales del diseño tales como memorias ram (BRAM) y bloques de procesamiento DSP (DSP48E). Se recomienda dejar que la herramienta PAR ubique la lógica auxiliar, ya que hacerlo a mano provoca resultados suboptimos. Sólo mueva LUTs individuales, si usted tiene una razón específica para hacerlo (por ejemplo, encaminamiento de bits de estado en todo el perímetro de la FFT).

No se recomienda mover recursos y dejarlos fijos dentro del chip, ya que se reporta que no funciona bien. Es mejor restringir un grupo de estos bloques DSP48Es y BRAMs en áreas llamadas Pblocks. Cuando agregue recursos de la FPGA a los Pblocks, procure dejar espacio disponible, asegúrese de no llenar en más de un 70% para que PAR tenga espacio suficiente para poder encontrar alguna posición óptima para los recursos, sin

embargo esto depende del problema en particular que se esté tratando. Eventualmente podría ser necesario utilizar un Pblock al 100% y es probable que funcione de manera óptima. Esto se deberá evaluar durante el proceso, es por esta razón que se dice que estas herramientas dependen de la experiencia del usuario para obtener buenos resultados.

Existen otras formas de crear un Pblock. La primera de ellas consiste en definir un área vacía, para luego ir agregando elementos que usted considere más tarde, esto eventualmente podría permitir una mejor distribución de los bloques, analizando los reportes de sincronización antes de proceder a mover algún recurso de la FPGA. La segunda forma consiste en definir un área en el archivo de restricciones "system.ucf", siguiendo la sintaxis propia de este archivo [22]. Otra forma, es hacerlo vía menú de Floorplanning que permite generar Pblocks de manera automática o manual [16]. Sin embargo, estas formas que acabamos de mencionar no serán desarrolladas en esta guía, si necesita información de estos métodos, puede encontrarlo en los artículos de referencia señalados.

APÉNDICE B

TUTORIAL 2: ARCHIVO DE RESTRICCIONES.

APENDICE B

TUTORIAL 2: ARCHIVO DE RESTRICCIONES.

Contenidos:

- Restricciones.
- Wildcards.
- Modificando las “timing constraints”.
- Estrategia general de la ubicación de Bloques.

B.1 Restricciones.

Una vez finalizado el proceso de Floorplanning, y los Pblocks han sido distribuidos satisfactoriamente, dentro del Chip. Es hora de crear las restricciones de diseño, para que el proceso de compilación con casper_xps pueda llegar a compilar a una velocidad mayor.

Se debe tener en cuenta que todos los cambios que se generen son responsabilidad del desarrollador, que tomo todas las precauciones respecto a la estrategia utilizada. Debe considerar que en la mayoría de los casos el periodo objetivo no se logra, ya que generalmente, se generan slacks de tiempo del orden de -1.5ns a -2ns. El signo menos del slack time indica que el path no logra cumplir la restricción del clock, y que requiere al menos de la cantidad de nanosegundos señalada para que el diseño funcione correctamente. Por ejemplo, si el periodo objetivo era 2.5ns, y timing report de PlanAhead le indica que tiene un slack de tiempo de -1.77ns, el resultado final será la suma de ambos, es decir, un periodo de 4.2740ns = 234 MHz!

Para corregir esto se deben corregir los worst paths del diseño, esto requiere mucho trabajo y se recomienda tratar uno a uno los casos, para que se pueda establecer una estrategia que permita optimizar el diseño, de manera que los slacks tiendan a cero ns.

B.2 Wildcards:

Para asignar áreas y crear Pblocks de manera más eficiente, PlanAhead soporta el uso de WildCards, para poder buscar palabras dentro de los Netlists. Se puede usar “*” o “?”. El signo “*” sirve para buscar palabras dentro de frases, para poder abarcar más posibilidades. Y el signo “?” sirve para que al momento de recompilar, las etiquetas hash que usa el compilador queden como comodín y se puedan reubicar en nuevos lugares. A continuación en la figura 4-23 se presenta un ejemplo de uso de “*”. Y finalmente, en la figura 4-24 se presenta un ejemplo de uso de “?”.

```
INST "*dout0_0*" AREA_GROUP = "VACC";
INST "*dout0_1*" AREA_GROUP = "VACC";
INST "*dout0_2*" AREA_GROUP = "VACC";
INST "*dout0_3*" AREA_GROUP = "VACC";
INST "*dout1_0*" AREA_GROUP = "VACC";
INST "*dout1_1*" AREA_GROUP = "VACC";
INST "*dout1_2*" AREA_GROUP = "VACC";
INST "*dout1_3*" AREA_GROUP = "VACC";
###INST "*vcm0*" AREA_GROUP = "VACC";
###INST "*vcm1*" AREA_GROUP = "VACC";
###INST "*vcm2*" AREA_GROUP = "VACC";
###INST "*vcm3*" AREA_GROUP = "VACC";
###INST "*vcm4*" AREA_GROUP = "VACC";
###INST "*vcm5*" AREA_GROUP = "VACC";
###INST "*vcm6*" AREA_GROUP = "VACC";
###INST "*vcm7*" AREA_GROUP = "VACC";
INST "*c_add_dsp_*" AREA_GROUP = "VACC";
INST "*c_add_dsp1_*" AREA_GROUP = "VACC";
INST "*c_add_dsp2_*" AREA_GROUP = "VACC";
INST "*c_add_dsp3_*" AREA_GROUP = "VACC";
INST "*c_add_dsp4_*" AREA_GROUP = "VACC";
INST "*c_add_dsp5_*" AREA_GROUP = "VACC";
INST "*c_add_dsp6_*" AREA_GROUP = "VACC";
INST "*c_add_dsp7_*" AREA_GROUP = "VACC";
INST "*power_dsp48e1_*" AREA_GROUP = "VACC";
INST "*power_dsp48e2_*" AREA_GROUP = "VACC";
INST "*power_dsp48e3_*" AREA_GROUP = "VACC";
INST "*power_dsp48e4_*" AREA_GROUP = "VACC";
INST "*power_dsp48e5_*" AREA_GROUP = "VACC";
INST "*power_dsp48e6_*" AREA_GROUP = "VACC";
INST "*power_dsp48e7_*" AREA_GROUP = "VACC";
INST "*power_dsp48e8_*" AREA_GROUP = "VACC";
INST "*quant0_*" AREA_GROUP = "VACC";
INST "*quant1_*" AREA_GROUP = "VACC";
AREA_GROUP "VACC" RANGE=CLOCKREGION_X1Y2:CLOCKREGION_X1Y5;
```

FIG. B-1 Ejemplo de uso de "*"

```
INST "spectrometer_dctrl_m_XSG_core_config/spectrometer_dctrl_m_XSG_core_config/
spectrometer_dctrl_m_x0/pfb_fir_real_??????????/pol1_in1_tap2_??????????/mult/
Maddsub_mult_46_56" AREA_GROUP = "pfb_fir";

INST "spectrometer_dctrl_m_XSG_core_config/spectrometer_dctrl_m_XSG_core_config/
spectrometer_dctrl_m_x0/pfb_fir_real_??????????/pol1_in3_tap2_??????????/mult/
Maddsub_mult_46_56" AREA_GROUP = "pfb_fir";
```

FIG. B-2 Ejemplo de uso de "?"

B.3 Modificando las restricciones de sincronización en PlanAhead.

Para cambiar las timing constriants o restricciones de sincronización en PlanAhead. Diríjase a la ventana "Constraints", y seleccione lo siguiente:

```
Constraints -> Clk period -> Basic period
```

Cambie las cuatro restricciones NET PERIOD llamadas "adc[01]clk_[pn] " al tiempo de reloj deseado, por ejemplo en nuestro caso para alcanzar un reloj de 250 MHz, asignamos un periodo de 4 ns.

B.4 Guardando los cambios.

Una vez hecho todos los cambios que se requieren, y después de haber editado agregando wildcards en el archivo "system.ucf" se deben guardar los cambios. Ya ha

finalizado de crear un archivo de restricciones. Para finalizar debe guardar el archivo en la carpeta:

```
/Ruta_del_Archivo…/Nombre_del_Diseño/XPS_ROACH_base/Data/
```

B.5 Archivo de restricciones y su sintaxis.

A continuación se presenta un ejemplo de un archivo "*system.ucf*" de restricciones. Una vez realizado el *Floorplanning* se crean restricciones con el objetivo de mejorar las estrategias PAR (Acrónimo inglés de *placing and routing*) durante el proceso compilación. El siguiente archivo de restricciones, permite optimizar el diseño propuesto en la memoria, permitiendo extender el ancho de banda de los espectrómetros de 500 MHz a 1000 MHz.

```
NET "adc0clk_p" LOC = H19 | IOSTANDARD = LVDS_25 | DIFF_TERM = TRUE
| PERIOD = 4 ns ;
NET "adc0clk_n" LOC = H20 | IOSTANDARD = LVDS_25 | DIFF_TERM = TRUE
| PERIOD = 4 ns ;
NET "adc1clk_p" LOC = K17 | IOSTANDARD = LVDS_25 | DIFF_TERM = TRUE
| PERIOD = 4 ns ;
NET "adc1clk_n" LOC = L18 | IOSTANDARD = LVDS_25 | DIFF_TERM = TRUE
| PERIOD = 4 ns ;

############################   P-Bocks   ############################


INST "*fft_wideband_real_*/fft_stage_1*" AREA_GROUP = "fft_stg1to6";
INST "*fft_wideband_real_*/fft_stage_2*" AREA_GROUP = "fft_stg1to6";
INST "*fft_wideband_real_*/fft_stage_3*" AREA_GROUP = "fft_stg1to6";
AREA_GROUP "fft_stg1to6" RANGE=CLOCKREGION_X0Y4:CLOCKREGION_X0Y6;


INST      "*fft_wideband_real1_*/fft_stage_1*"      AREA_GROUP      =
"fft1_stg1to6";
INST      "*fft_wideband_real1_*/fft_stage_2*"      AREA_GROUP      =
"fft1_stg1to6";
INST      "*fft_wideband_real1_*/fft_stage_3*"      AREA_GROUP      =
"fft1_stg1to6";
AREA_GROUP "fft1_stg1to6" RANGE=CLOCKREGION_X0Y3:CLOCKREGION_X0Y1;
#AREA_GROUP "fft1_stg1to3" RANGE=SLICE_X0Y60:SLICE_X47Y80;
#AREA_GROUP "fft1_stg1to3" RANGE=DSP48_X0Y24:DSP48_X5Y31;
#AREA_GROUP "fft1_stg1to3" RANGE=RAMB36_X0Y12:RAMB36_X3Y15;

#INST      "*fft_wideband_real1_*/fft_stage_4*"      AREA_GROUP      =
"fft1_stg4to6";
#INST      "*fft_wideband_real1_*/fft_stage_5*"      AREA_GROUP      =
"fft1_stg4to6";
#INST      "*fft_wideband_real1_*/fft_stage_6*"      AREA_GROUP      =
"fft1_stg4to6";
#AREA_GROUP "fft1_stg4to6" RANGE=SLICE_X0Y30:SLICE_X47Y59;
```

```
#AREA_GROUP "fft1_stg4to6" RANGE=DSP48_X0Y12:DSP48_X5Y23;
#AREA_GROUP "fft1_stg4to6" RANGE=RAMB36_X0Y6:RAMB36_X3Y11;

#INST "*fft_wideband_real*/fft_stage_4*" AREA_GROUP = "fft_stg4to6";
#INST "*fft_wideband_real*/fft_stage_5*" AREA_GROUP = "fft_stg4to6";
#INST "*fft_wideband_real*/fft_stage_6*" AREA_GROUP = "fft_stg4to6";
#AREA_GROUP "fft_stg4to6" RANGE=SLICE_X0Y99:SLICE_X47Y129;
#AREA_GROUP "fft_stg4to6" RANGE=DSP48_X0Y40:DSP48_X5Y51;
#AREA_GROUP "fft_stg4to6" RANGE=RAMB36_X0Y20:RAMB36_X3Y25;

INST "*fft_wideband_real_*/fft_stage_7*" AREA_GROUP = "fft_stg7to9";
INST "*fft_wideband_real_*/fft_stage_8*" AREA_GROUP = "fft_stg7to9";
INST "*fft_wideband_real_*/fft_stage_9*" AREA_GROUP = "fft_stg7to9";
AREA_GROUP "fft_stg7to9" RANGE=SLICE_X0Y130:SLICE_X47Y159;
AREA_GROUP "fft_stg7to9" RANGE=DSP48_X0Y52:DSP48_X5Y63;
AREA_GROUP "fft_stg7to9" RANGE=RAMB36_X0Y26:RAMB36_X3Y31;


INST      "*fft_wideband_real1_*/fft_stage_7*"      AREA_GROUP      =
"fft1_stg7to9";
INST      "*fft_wideband_real1_*/fft_stage_8*"      AREA_GROUP      =
"fft1_stg7to9";
INST      "*fft_wideband_real1_*/fft_stage_9*"      AREA_GROUP      =
"fft1_stg7to9";
AREA_GROUP "fft1_stg7to9" RANGE=SLICE_X0Y0:SLICE_X47Y29;
AREA_GROUP "fft1_stg7to9" RANGE=DSP48_X0Y0:DSP48_X5Y11;
AREA_GROUP "fft1_stg7to9" RANGE=RAMB36_X0Y0:RAMB36_X3Y5;

##
INST  "*fft_wideband_real1_*/fft_direct_*/butterfly1_*"  AREA_GROUP  =
"direct1_stg1";
INST  "*fft_wideband_real1_*/fft_direct_*/butterfly2_*"  AREA_GROUP  =
"direct1_stg1";
AREA_GROUP "direct1_stg1" RANGE=CLOCKREGION_X1Y0:CLOCKREGION_X1Y1;

INST  "*fft_wideband_real_*/fft_direct_*/butterfly1_*"  AREA_GROUP  =
"direct_stg";
INST  "*fft_wideband_real_*/fft_direct_*/butterfly2_*"  AREA_GROUP  =
"direct_stg";
AREA_GROUP "direct_stg" RANGE=CLOCKREGION_X1Y7:CLOCKREGION_X1Y6;

INST    "*fft_wideband_real1_*/fft_unscrambler_*"    AREA_GROUP    =
"fft1_unscrambler";
AREA_GROUP                                          "fft1_unscrambler"
RANGE=CLOCKREGION_X1Y1:CLOCKREGION_X1Y1;

INST    "*fft_wideband_real_*/fft_unscrambler_*"    AREA_GROUP    =
"fft_unscrambler";
AREA_GROUP                                          "fft_unscrambler"
RANGE=CLOCKREGION_X1Y6:CLOCKREGION_X1Y6;
##

INST "*dout0_0*" AREA_GROUP = "VACC";
```

```
INST "*dout0_1*" AREA_GROUP = "VACC";
INST "*dout0_2*" AREA_GROUP = "VACC";
INST "*dout0_3*" AREA_GROUP = "VACC";
INST "*dout1_0*" AREA_GROUP = "VACC";
INST "*dout1_1*" AREA_GROUP = "VACC";
INST "*dout1_2*" AREA_GROUP = "VACC";
INST "*dout1_3*" AREA_GROUP = "VACC";
###INST "*vcm0*" AREA_GROUP = "VACC";
###INST "*vcm1*" AREA_GROUP = "VACC";
###INST "*vcm2*" AREA_GROUP = "VACC";
###INST "*vcm3*" AREA_GROUP = "VACC";
###INST "*vcm4*" AREA_GROUP = "VACC";
###INST "*vcm5*" AREA_GROUP = "VACC";
###INST "*vcm6*" AREA_GROUP = "VACC";
###INST "*vcm7*" AREA_GROUP = "VACC";
INST "*c_add_dsp_*" AREA_GROUP = "VACC";
INST "*c_add_dsp1_*" AREA_GROUP = "VACC";
INST "*c_add_dsp2_*" AREA_GROUP = "VACC";
INST "*c_add_dsp3_*" AREA_GROUP = "VACC";
INST "*c_add_dsp4_*" AREA_GROUP = "VACC";
INST "*c_add_dsp5_*" AREA_GROUP = "VACC";
INST "*c_add_dsp6_*" AREA_GROUP = "VACC";
INST "*c_add_dsp7_*" AREA_GROUP = "VACC";
INST "*power_dsp48e1_*" AREA_GROUP = "VACC";
INST "*power_dsp48e2_*" AREA_GROUP = "VACC";
INST "*power_dsp48e3_*" AREA_GROUP = "VACC";
INST "*power_dsp48e4_*" AREA_GROUP = "VACC";
INST "*power_dsp48e5_*" AREA_GROUP = "VACC";
INST "*power_dsp48e6_*" AREA_GROUP = "VACC";
INST "*power_dsp48e7_*" AREA_GROUP = "VACC";
INST "*power_dsp48e8_*" AREA_GROUP = "VACC";
INST "*quant0_*" AREA_GROUP = "VACC";
INST "*quant1_*" AREA_GROUP = "VACC";
AREA_GROUP      "VACC"      RANGE=CLOCKREGION_X1Y2:CLOCKREGION_X1Y5;
```

APÉNDICE C

TUTORIAL 3: COMPILACIÓN DEL DISEÑO OPTIMIZADO.

APÉNDICE C

TUTORIAL 3: COMPILACIÓN DEL DISEÑO OPTIMIZADO.

Contenidos:

- Introducción a EDK.
- Compilación con un ancho de banda de 1000 MHz.
- Resultado de la compilación.

C.1 Introducción a EDK.

Si el diseño que se está optimizando no cumple las restricciones de sincronización, el diseño arrojará un error. A menudo los diseños fallan incluso cuando solo una estadística de tiempo está fuera de los parámetros. Para evitar que este tipo de problemas ocurran, se debe seguir las siguientes instrucciones.

Abra la aplicación de Xilinx XPS, cuando se inicialice aparecerá una ventana como la que aparece en la FIG. C-1.

FIG. C-1 Inicialización de EDK de Xilinx.

Una vez abierto el software EDK, desplegara una interfaz de usuario tal como se presenta en la Fig. C-2.

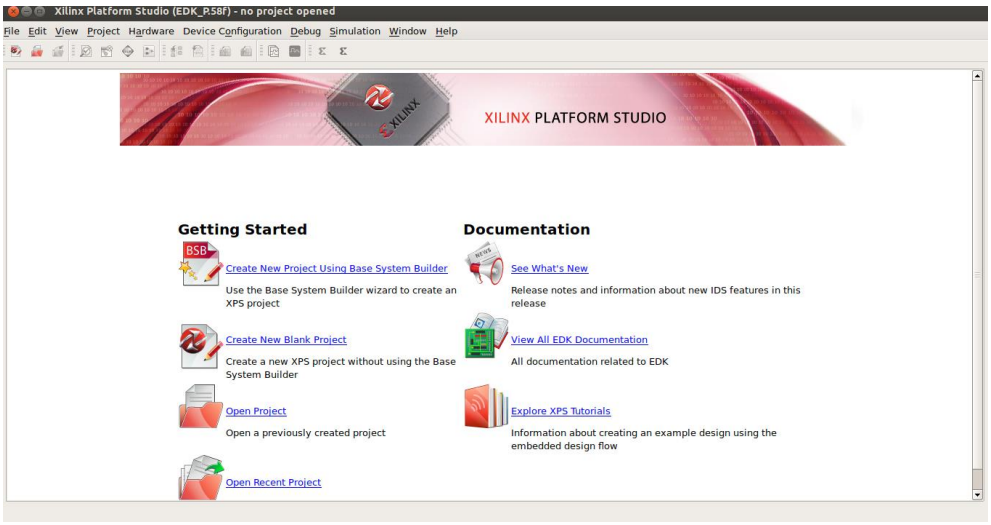

Fig. C-2 Interfaz de usuario de EDK.

A continuación haga *click* en "*Open Recent Project*" y busque el proyecto que debió haber sido compilado previamente y que además debe estar abierto en *Simulink*.

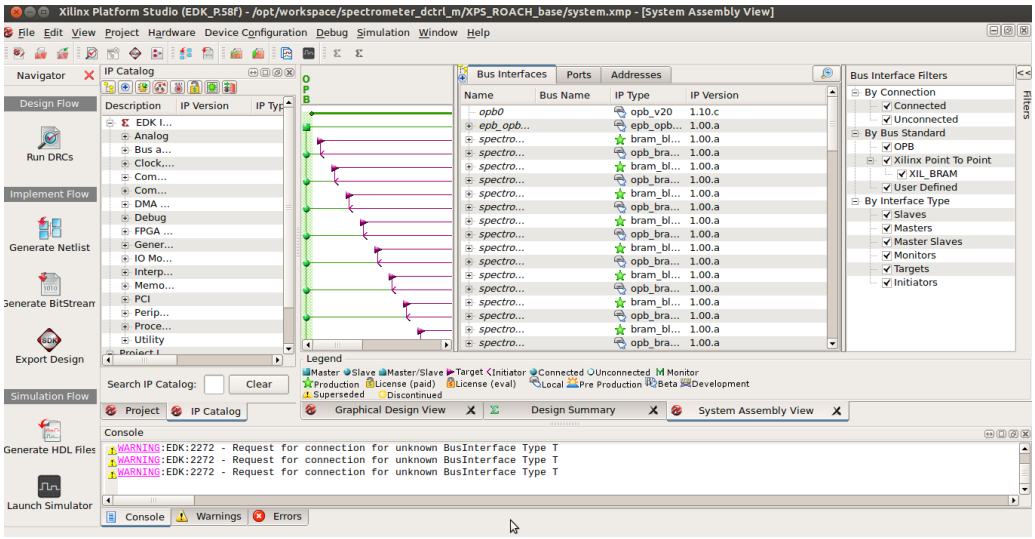

Fig. C-3 Ambiente de trabajo de EDK.

En la Fig. C-3se presenta el ambiente de trabajo de EDK. Haga *click* en "*Project*", y luego en "*Project Options*", luego haga *click* en la pestaña "*Design Flow*", para luego Inhabilitar la opción "*Treat timing closure failure as an error*", tal como aparece en la Fig. C-4. Una vez hecho esto presione OK, y cierre EDK.

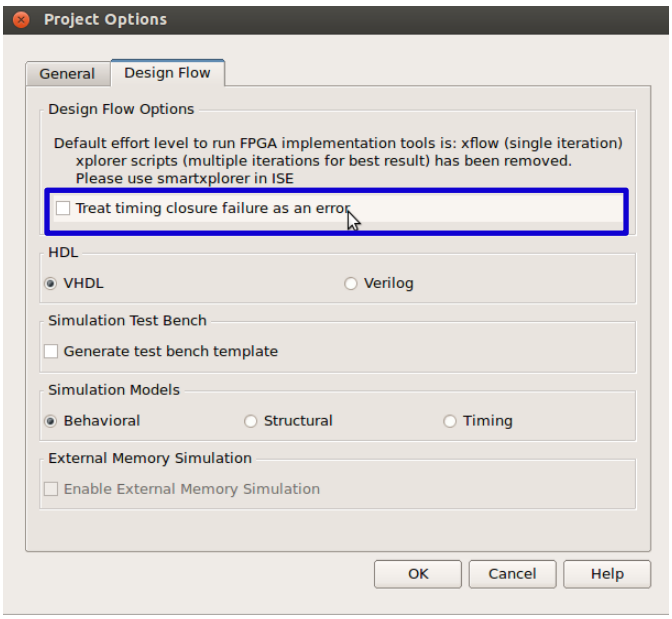

Fig. C-4 Inhabilitar la opción "Treat timing closure failure as an error"

Nota: Con esos pasos al momento de compilar, las fallas por sincronización (*timing*) no se tomaran en cuenta, tenga precaución y siga estos pasos solo en el caso que este seguro, que tiene todos los aspectos de timing controlados en el diseño.

C.2 Compilación con un ancho de banda de 1000 MHz.

Finalmente, el último paso cosiste en llevar a cabo la compilación con todos los cambios realizados, y el nuevo archivo de restricciones.

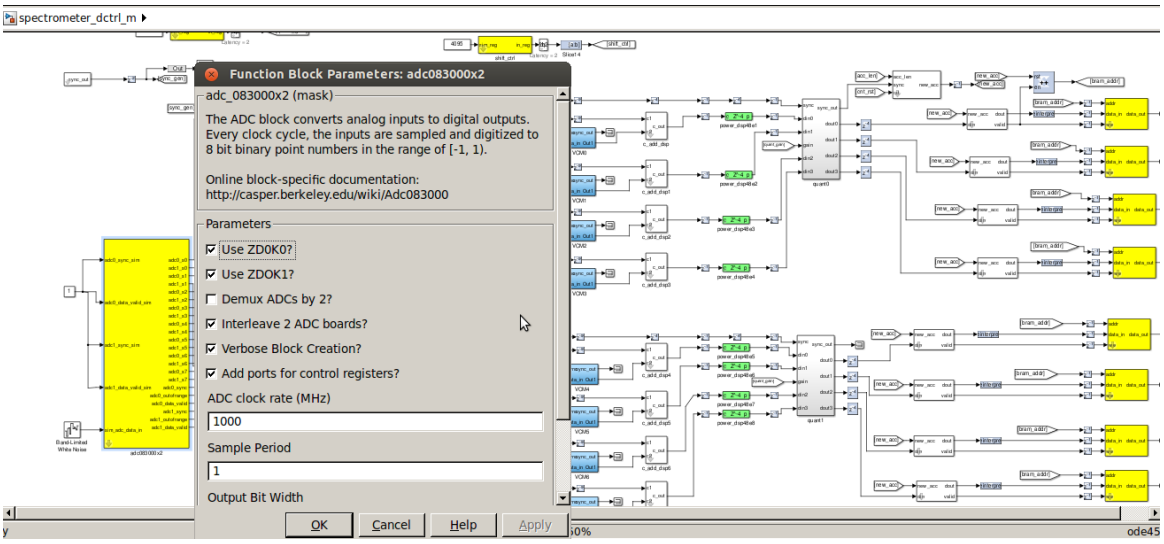

Fig. C-5 aumentando ADC clock rate a 1000MHz

Primero que todo, se deben cambiar algunos parámetros. En la Fig. C-5, se cambia el *ADC clock rate* a 1000 MHz. Luego, se hace *click* en *OK*.

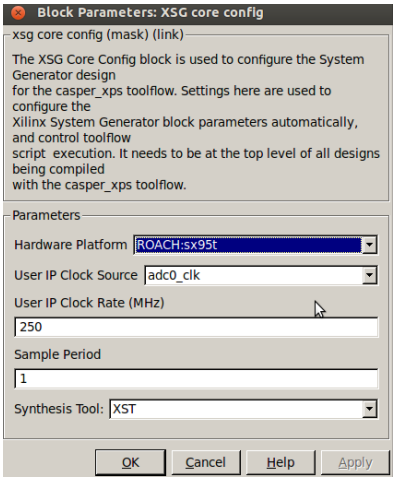

FIG. C-6 aumentando el clock rate del FPGA a 250 MHz

Segundo, haga *click* en el *XSG core config*, se debe cambiar la opción *User IP Clock Rate* a 250 MHz, tal como se presenta en la FIG. C-6. Luego, presione Ctrl+d, para propagar los tipo de datos como aparece en la FIG. C-7, esto asegura que la compilación se realizara correctamente.

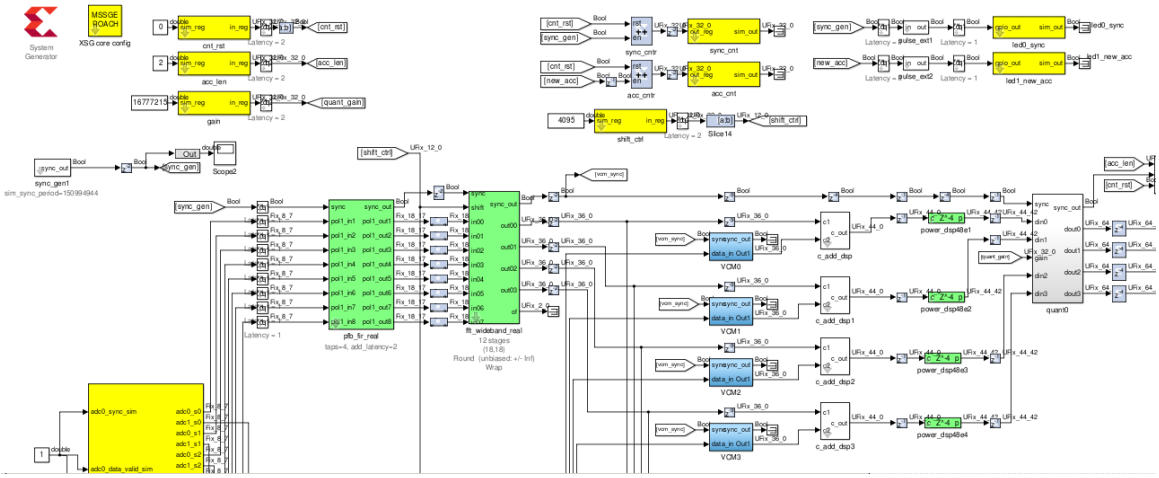

FIG. C-7 Ctrl+d para cargar los tipos de datos del diseño

Finalmente, en la consola de Matlab, escriba *casper_xps*, y luego inhabilite todas las opciones del *System Generator* con excepción de *EDK/ISE/Bitgen*. Luego presione, *RUN XPS*, tal como se aprecia en la FIG. C-8.

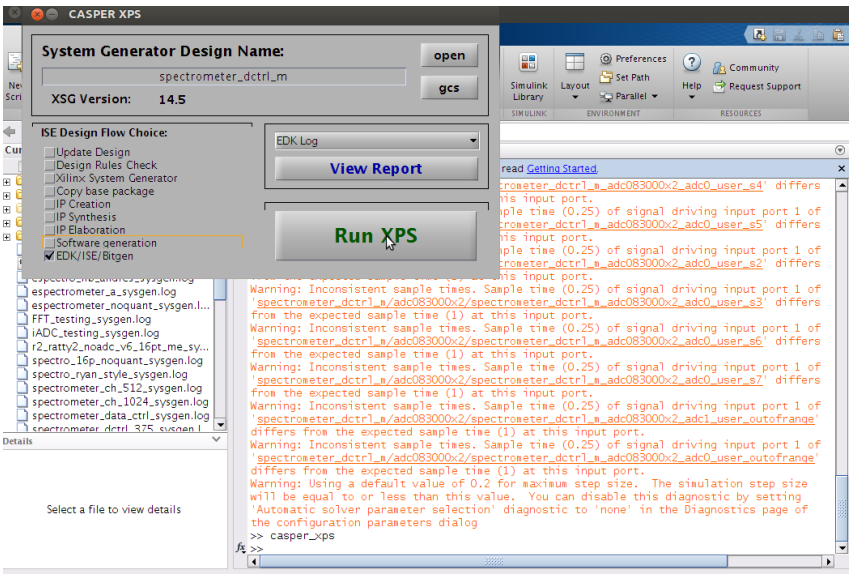

FIG. C-8 Habilitar solo EDK/ISE/Bitgen

Una vez hecho esto comenzara a correr el compilador, se leerá el archivo de restricciones “system.ucf”, y luego si todo fue realizado correctamente, se debería obtener un diseño optimizado y compilado satisfactoriamente.

C.3 Resultado de la compilación.

En la FIG. C-9 se presenta el resultado de la compilación del diseño optimizado, presenta el porcentaje de dispositivos que se utilizó. Como se puede apreciar este diseño utilizo sobre un 90% en lógica y un 60% de DSP48Es y un 40% de BRAMs. Lo que permitió, poder realizar una reubicación de estos dos últimos tipos de bloques, luego el compilador se hizo cargo se reubicar la lógica de una manera eficaz.

Además se obtiene una compilación con un clock rate de 250MHz, permitiendo hacer funcionar el ADC a una tasa de muestreo de 2GSPS.

Sin embargo, se recomienda no utilizar más del 70% de los recursos del FPGA, si se desea aumentar la velocidad de funcionamiento del reloj.

```
Device Utilization Summary:

   Number of BUFGs                          8 out of 32     25%
   Number of DCM_ADVs                       4 out of 12     33%
   Number of DSP48Es                      384 out of 640    60%
      Number of LOCed DSP48Es              10 out of 384     2%

   Number of ILOGICs                      111 out of 800    13%
   Number of External IOBs                190 out of 640    29%
      Number of LOCed IOBs                190 out of 190   100%

   Number of OLOGICs                       19 out of 800     2%
   Number of RAMB18X2s                     99 out of 244    40%
      Number of LOCed RAMB18X2s            15 out of 99     15%

   Number of RAMB36_EXPs                   80 out of 244    32%
   Number of Slices                     14501 out of 14720  98%
   Number of Slice Registers            53623 out of 58880  91%
      Number used as Flip Flops         53623
      Number used as Latches                0
      Number used as LatchThrus             0

   Number of Slice LUTS                 44431 out of 58880  75%
   Number of Slice LUT-Flip Flop pairs  55518 out of 58880  94%
```

FIG. C-9 Resultado de la compilación. Archivo “xflow”