



**UNIVERSIDAD NACIONAL DE RÍO NEGRO**

---

---

## **PROYECTO FINAL INTEGRADOR**

**Diseño de una  
Radio Definida por Software**

---

**Hecho por:**  
**Omar López Cabrera**

**Director: Ing. Franco Alcaraz**

**Codirector: Ing. Federico Tula**

**Para el título de Ingeniería Electrónica**

## Índice

Índice de figuras	IV
Índice de tablas	V
Lista de acrónimos	VI
Resumen	VIII

## Tabla de contenido

<b>1. Introducción</b> .....	0
1.1 ¿Que es una radio definida por software?.....	0
1.2 ¿Por qué utilizar radios definidas por software? .....	0
1.3 Aplicaciones y características de las SDR .....	0
1.4 Comparativa entre Radios Definidas por Software y Radio Definidas por Hardware. ....	1
1.5 Objetivos de la tesis.....	2
1.6 Metodología de diseño .....	4
<b>2. Diseño</b> .....	5
2.1 Arquitectura general según funciones .....	5
2.1.1 Antenas .....	6
2.1.2 RF Front-End.....	7
2.1.3 DAC y ADC. .....	8
2.1.4 Digital Front-End .....	8
2.1.5 CPU .....	9
2.1.6 Memoria.....	11
2.1.7 Conexiones externas .....	11
2.1.8 Interfaces.....	11
2.1.9 Alimentación .....	11
2.2 Arquitectura definida .....	11
2.2.1 RF Front End & Digital Front End.....	12
2.2.2 USB 3.0 .....	15
2.2.2.1 Información general USB 3.0.....	16
2.2.2.2 Controlador USB 3.0 CYUSB3014 .....	17
2.2.3 Memoria DDR3 .....	19
2.2.4 CPU .....	20
2.2.4.1 Selección de FPGA .....	20

2.2.4.2 Pin Out y encapsulado.....	23
2.2.4.3 Función.....	24
2.2.4.4 Configuración FPGA.....	25
2.2.4.5 Alimentación FPGA.....	25
2.2.5 Alimentación SDR.....	25
2.2.6 GPIO y Leds.....	27
2.3 SDR a nivel sistema.....	28
2.3.1 HDL .....	29
2.3.2 Programación FPGA.....	30
2.3.3 SDR transmisión y recepción Half Duplex .....	32
2.3.4 SDR transmisión y recepción Full Duplex .....	34
2.3.5 Software .....	35
2.3.5.1 Drivers .....	35
2.3.5.2 GNU Radio .....	36
2.4 SDR Características.....	37
<b>3. Diseño con herramienta CAD .....</b>	<b>39</b>
3.1 Diseño herramienta CAD.....	39
3.1.1 Esquemático .....	39
3.1.2 Componentes .....	40
3.2 Conexiones .....	40
3.2.1 FPGA .....	41
3.2.1.1 Bancos I/O .....	41
3.2.2 Power .....	42
3.2.3 Transceiver AD9364 .....	42
3.2.4 USB 3.0 .....	42
3.3 Capacidades del fabricante PCB .....	42
3.4 Distribución de componentes y características físicas.....	44
3.5 Stackup .....	45
3.6 Líneas de transmisión.....	48
3.7 Impedancia característica, reflexión y terminación.....	49
3.8 Líneas diferenciales .....	50
3.9 Ajuste de longitudes.....	51
3.10 Consideraciones tomadas al Rutear la placa.....	53
3.10.1 FPGA .....	53
3.10.2 Power .....	53
3.10.3 USB 3.0 .....	53

3.10.4 AD9364.....	53
3.11 Planos de alimentación y tierra.....	54
3.12 Capacitores desacople .....	55
3.13 Modelo 3d .....	57
<b>4. Sumario .....</b>	<b>59</b>
4.1 Simulación de integridad de la señal.....	59
4.2 Fabricación y ensamblado del PCB.....	59
4.3 Software .....	60
4.4 Posibles Mejoras .....	60
<b>Anexos.....</b>	<b>62</b>
A-Hardware Design Guidelines .....	62
A.1 USB 3.0 .....	62
A.2 FPGA.....	64
A.3 Memory DDR3.....	65
A.4 Transceiver AD9364 .....	67
B-Cuestiones generales del diseño .....	67
B.1-Alimentación.....	67
Vías para escape térmico .....	67
Header test por etapa .....	68
B.2-USB 3.0.....	68
Protección ESD .....	68
B.3-AD9364 .....	69
Shielding sección RF .....	69
Interface entre el transceiver y las antenas .....	69
Redes de adaptación del transceiver .....	71
Criterio de interface digital transceiver .....	75
C-Herramientas utilizadas durante el diseño.....	76
C.1-Estimación Consumo FPGA.....	76
C.2-Programa ADP5054 criterio más eficiente.....	78
C.3-Calculo de corriente por vías. ....	79
C.4-Calculo Impedancia diferencial.....	79
C.5-Uso de herramienta xSignals .....	81
C.6-Coplanar waveguide y ajuste de impedancia .....	83
D- Simulaciones .....	84
D.1-Simulaciones Power.....	84
D.2-Simulaciones etapa de RF AD9364 con ADS.....	85

E- Información general y hoja de datos .....	87
E.1-Recomendaciones por parte de los principales IC.....	87
E.2-Capacidades Fabricante PCB Prototype.....	87
E.3-Información sobre conexiones de cada sección .....	88
E.4-Esquemático, Gerbers, BOM.....	88
<b>Referencias .....</b>	<b>89</b>
<b>Bibliografia.....</b>	<b>91</b>

# Índice de Figuras

Figura 1: Etapas del proyecto .....	2
Figura 2: Esquema de funcionamiento receptor FM .....	3
Figura 3: Esquema de funcionamiento comunicación punto a punto .....	3
Figura 4: Diagrama metodología de diseño Top-Down .....	4
Figura 5: Diagrama en bloques de la arquitectura general de la SDR.....	6
Figura 6: Ejemplo patrón de radiación de una Antena .....	7
Figura 7: Proceso de diezmado .....	9
Figura 8: Proceso de interpolación .....	9
Figura 9: Comparativa entre los posibles CPUs.....	10
Figura 10: Diagrama en bloques de la Arquitectura definida de la SDR .....	12
Figura 11: Diagrama de arquitectura definida, sección transceiver .....	13
Figura 12: Cables USB 3.0.....	16
Figura 13: Diagrama en bloques controlador USB 3.0 .....	17
Figura 14: Diagrama en bloques receptor QAM .....	21
Figura 15: Modelo de FPGA seleccionado.....	23
Figura 16: Pin Out I/O FPGA .....	24
Figura 17: Diagrama distribución de tensiones.....	27
Figura 18: Pines de propósito general.....	27
Figura 19: Esquema funcional SDR.....	28
Figura 20: Programación de la FPGA.....	30
Figura 21: Diagrama en bloques configuración FPGA .....	31
Figura 22: Diagrama de tiempos, configuración FPGA.....	32
Figura 23: Esquema transmisión de la SDR .....	33
Figura 24: Esquema Recepción de la SDR .....	33
Figura 25: Esquema de operación Full Duplex .....	34
Figura 26: Hardware y drivers del sistema .....	35
Figura 27: Receptor de FM, GNU Radio .....	37
Figura 28: Potencia salida transceiver.....	38
Figura 29 : Bandas y asignaciones de frecuencia .....	38
Figura 30: Esquemático en forma jerárquica, primer sheet .....	39
Figura 31: Información componentes del esquemático .....	40
Figura 32: Interface de los bancos de la FPGA .....	41
Figura 33: Restricción distancia mínima en PCB .....	43
Figura 34: Restricciones tamaño agujero en PCB.....	43
Figura 35: Espacio disponible en componentes BGA.....	44
Figura 36: Ejemplo de ruteo, componente BGA .....	44
Figura 37: Distribución de los componentes en la placa.....	45
Figura 38: Stack up del PCB .....	48
Figura 39: Microstrip .....	49
Figura 40: Strip line .....	50
Figura 41: Propiedad de las líneas diferenciales .....	51
Figura 42: Líneas diferenciales en PCB .....	51
Figura 43: Ejemplo ruteo memoria DDR3 .....	52

Figura 44: Plano1 de power .....	54
Figura 45: Plano2 de power .....	54
Figura 46: Plano sólido GND .....	55
Figura 47: Ubicación capacidores desacople FPGA .....	56
Figura 48: Modelo 3D placa, vista superior.....	57
Figura 49: Modelo 3D placa, vista inferior .....	58
Figura 50: Diagrama de conexión, mejora etapa RF .....	61

**Falta agregar figuras de anexos. También que cada figura diga a que capítulo pertenece**

## Índice de Tablas

Tabla 1: Transceivers disponibles en mercado.....	12
Tabla 2: Opciones de Booting USB 3.0 .....	18
Tabla 3: Selección de clock USB 3.0 .....	19
Tabla 4: Estimación de recursos FPGA .....	21
Tabla 5: Estimación de consumo por etapa .....	26
Tabla 6: Corriente necesaria por tensión .....	26
Tabla 7: Selección del modo de booting FPGA.....	31
Tabla 8: Cantidad de señales por sección .....	47
Tabla 9: Impedancias de cada sección .....	47
Tabla 10: Longitud de líneas en la interfaz DDR3.....	52

**Falta agregar tablas de anexos. También que cada tabla diga a que capítulo pertenece**

## **Lista de Acrónimos**

<b>ADC</b>	Analog to Digital Converter
<b>AMBA</b>	Advanced Microcontroller Bus Architecture
<b>ASICS</b>	Application Specific Integrated Circuit
<b>API</b>	Application Programming Interface
<b>AXI</b>	Advanced eXtensible Interface
<b>BB</b>	Base Band
<b>BBA</b>	Base Band Amplifier
<b>BBP</b>	Base Band Processor
<b>BGA</b>	Ball Gate Array
<b>BOM</b>	Bill of Materials
<b>CPU</b>	Central Processing Unit.
<b>CAD</b>	Computer Assisted Design
<b>DAC</b>	Digital to Analog Converter
<b>DDC</b>	Digital Down Converter
<b>DDR</b>	Double Data Rate
<b>DDS</b>	Digital Direct Synthesis
<b>DMA</b>	Direct Memory Access
<b>DUC</b>	Digital Up Converter
<b>DSP</b>	Digital Signal Processor
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory
<b>ENACOM</b>	Ente Nacional de Comunicaciones
<b>FAQ</b>	Frequently Asked Questions
<b>FDD</b>	Frequency
<b>FIFO</b>	First In First Out
<b>FPGA</b>	Field Programmable Gate Array
<b>HDL</b>	Hardware Description Language
<b>GPIO</b>	General Purpose Input Output

<b>GPP</b>	General Purpose Processor
<b>IC</b>	Integrated Circuit
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>ITU</b>	International Telecommunication Union
<b>JTAG</b>	Joint Test Action Group
<b>LPF</b>	Low Pass Filter
<b>LAPAC</b>	Laboratorio de Procesamiento de Señales Aplicadas y Computación de Alto Rendimiento
<b>MIMO</b>	Multiple Input Multiple Output
<b>QAM</b>	Quadrature Amplitude Modulation
<b>QPSK</b>	Quadrature Phase Shift Keying
<b>PCB</b>	Printed Circuit Board
<b>RF</b>	Radio Frequency
<b>RFFE</b>	Radio Frequency Front End
<b>ROM</b>	Read Only Memory
<b>SDR</b>	Software Defined Radio
<b>SPI</b>	Serial Peripheral Interface
<b>SRRC</b>	Square Raised Root Cosine
<b>USB</b>	Universal Serial Bus
<b>VCO</b>	Voltage Controlled Oscillator
<b>XPE</b>	Xilinx Power Estimator

## Resumen

Esta tesis describe el diseño del hardware para una Radio Definida por Software (SDR). Partiendo de una arquitectura modular genérica, pasando a una arquitectura con componentes definidos, hasta el desarrollo del PCB de la misma.

Esta SDR cubre un rango de frecuencia de 70 MHz hasta 6 GHz, tiene una conexión USB 3.0 como interface externa y una FPGA Artix-7 como unidad de control y procesamiento. Dicha FPGA puede ser programable por USB; esta característica permite realizar experimentos en un rango de las siguientes señales: FM, TV broadcast, celular, Wi-Fi entre otros. El Front End de RF está compuesto por el transceiver AD9364 de Analog Devices el cual tiene 56 MHz de ancho de banda instantáneo. Este dispositivo puede ser alimentado por USB o por medio de un conector tipo Jack.

Esta placa puede utilizarse en los laboratorios de todos los cursos de comunicaciones de la carrera, además de los cursos de procesamiento de señales aleatorias y determinísticas. Solo requiere un par de antenas y contar con una computadora donde se programa el tipo de procesamiento que se desea hacer con los datos, utilizando software abierto (GNU Radio). Este tipo de equipos está presente en todas las carreras de Telecomunicaciones de las Universidades más prestigiosas del mundo, debido a la facilidad con que permiten realizar experimentos complejos y entender, rápidamente, el estado del arte. Al ser un desarrollo propio se pretende que el proyecto se pueda distribuir como hardware libre. De este modo, cualquier persona o institución que lo desee puede utilizarla sin más que tener que mencionar el origen. Esto dará visibilidad a la UNRN y a la carrera de Ingeniería Electrónica en particular.

# Capítulo 1

## 1. Introducción

### 1.1 ¿Qué es una radio definida por software?

El término radio definida por software fue utilizado por Josep Mitola en 1991 en referencia a las radios reprogramables o reconfigurables [1]. En otras palabras, un mismo hardware puede realizar diferentes funciones en momentos diferentes. El foro de SDR las define como una radio en la cual se puede programar el tráfico y control de la información, soportando un amplio rango de frecuencias y aplicaciones de software. El usuario puede cambiar de un formato de comunicación a otro en cuestión de milisegundos, usar el GPS para localización o recibir una transmisión satelital.

### 1.2 ¿Por qué utilizar radios definidas por software?

En el mundo de las comunicaciones inalámbricas, los estándares y protocolos avanzan a una velocidad tal que los equipos de comunicaciones quedan obsoletos poco después de su producción. Para remediar esto, las SDR representan un cambio en el paradigma comparado con las radios tradicionales, en donde gran parte de su funcionalidad es implementada mediante software en lugar de hardware. Este cambio, reduce la cantidad de componentes analógicos y de radio frecuencia (RF), haciendo énfasis en el procesamiento de señales y la programación, logrando una gran versatilidad en el receptor.

### 1.3 Aplicaciones y características de las SDR

**Interoperabilidad:** Una SDR puede operar como puente entre múltiples sistemas de radios incompatibles. Interoperabilidad es una de las principales razones por las cuales los militares estaban interesados en ellas. Diferentes ramas de los militares y los policías usan decenas de radios incompatibles. Durante comunicaciones en operaciones conjuntas una SDR multicanal puede actuar como intérprete de las diferentes radios. Un proyecto militar de los Estados Unidos que aplica esta característica es el SpeakEasy [2]. Este utiliza una SDR para realizar comunicaciones entre 10 diferentes tipos de radio militares en un solo sistema.

**Eficiencia en el uso de recursos ante condiciones variables:** Una SDR puede adaptar la forma de onda para maximizar algún parámetro clave. Por ejemplo, una señal de baja potencia o una modulación más lenta puede ser seleccionada si la radio está baja de batería. Una señal de alta tasa puede ser seleccionada si se requiere bajar un archivo rápido. Eligiendo la forma de onda más apropiada en cada escenario, las radios pueden proveer una mejor experiencia al usuario (por ejemplo, mayor durabilidad de la batería). Siempre y cuando haya una negociación posible entre las radios de la red.

**Uso conveniente de las frecuencias (Radio Cognitiva):** Una SDR puede sacar ventaja del espectro de frecuencia que no está siendo utilizado. Si el dueño de la porción de espectro no está utilizando la misma, una SDR puede pedir prestado hasta que el dueño vuelva. Esta técnica tiene un gran potencial para incrementar la cantidad de espectro disponible.

**Aumento de tiempo de vida útil:** Una SDR puede actualizarse en el campo para soportar los últimos estándares de comunicaciones. Esta capacidad es de especial importancia para el ciclo de vida de las radios en aplicaciones militares, comunicaciones o aeroespaciales. Por ejemplo, un nuevo estándar de celular que sale al mercado puede ser cargado remotamente mediante un nuevo software en la estación base (SDR), ahorrando costos de un nuevo hardware y trabajo de instalación.

**Bajo costo:** Una SDR puede adaptarse para usarse en diferentes mercados y múltiples aplicaciones. Algunas radios pueden ser vendidas a compañías de celulares o automóviles, pudiendo reducir los costos de mantenimiento y entrenamiento.

**Investigación y desarrollo:** Las SDR pueden ser utilizadas para implementar diferentes formas de ondas en tiempo real para análisis de performance.

#### [1.4 Comparativa entre Radios Definidas por Software y Radio Definidas por Hardware.](#)

A continuación se muestra una tabla específica de ventajas (sombreado gris) y desventajas entre los dos tipos de radio.

Es importante aclarar que esta comparativa no es válida para todos los casos de uso de las radios.

Radio definida por software	Radio definida por hardware
Mayor tiempo de desarrollo	Menor tiempo de desarrollo
Adaptabilidad a diferentes escenarios	Se debe utilizar para un escenario específico
Gran variedad de modulaciones	Modulaciones dependen del HW adquirido
Simulaciones en PC	Necesidad de laboratorio y equipamiento para simulaciones
Adaptable a nuevas tecnologías	No adaptable a nuevas tecnologías
Mayor costo económico al principio	Menor costo al principio
Menor Sensibilidad en frecuencias específicas	Mayor sensibilidad en frecuencias específicas
El consumo es menor ya que se adapta a momentos y formas de funcionamiento.	Sólo tiene dos modos de consumo On y Stand By
Reprogramable	Única funcionalidad desde que se graba el firmware

Tabla 1: Tabla comparativa entre radios

## 1.5 Objetivos de la tesis

La finalidad de esta tesis, es el diseño de una Radio Definida por Software. Esta representa las primeras etapas de un proyecto el cual contempla la construcción y puesta en marcha de la misma. La siguiente figura muestra las diferentes etapas del proyecto completo:

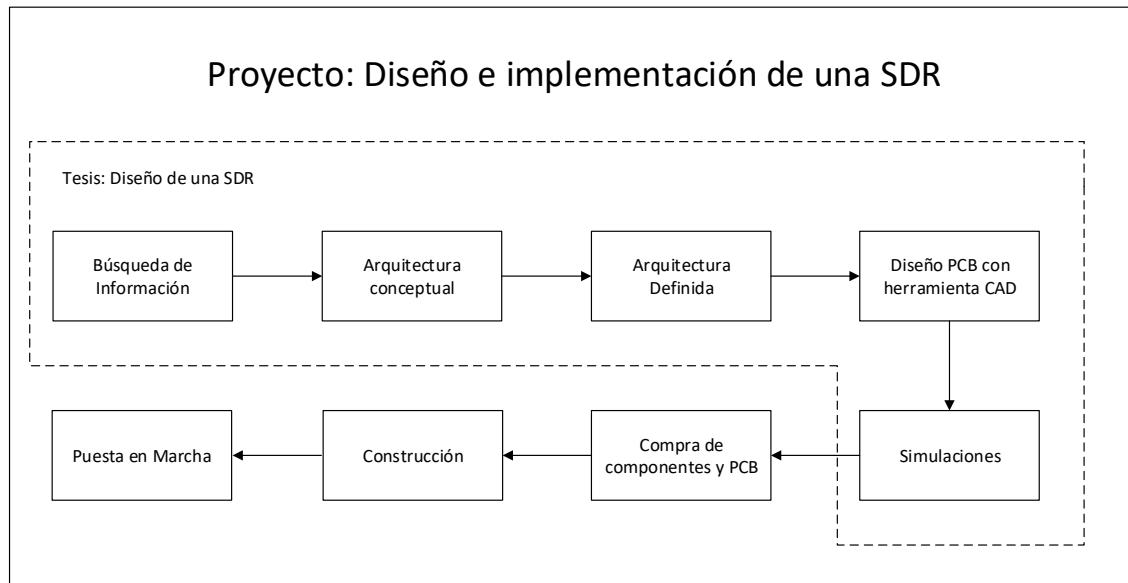


Figura 1: Etapas del proyecto

Este proyecto tiene los siguientes requerimientos:

- Poder escuchar las radios comerciales de FM mediante el hardware a desarrollar y una computadora utilizando el programa open source GNU-RADIO.

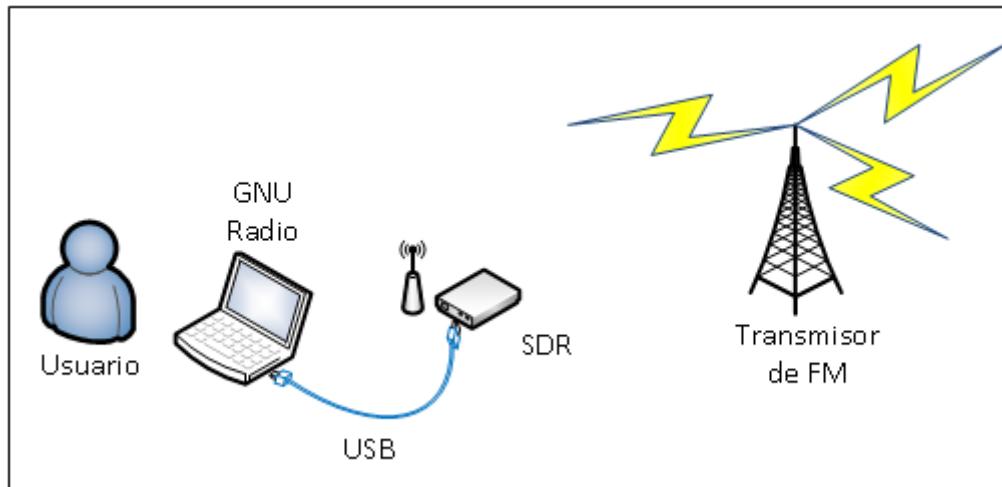


Figura 2: Esquema de funcionamiento receptor FM

- Establecer comunicaciones digitales punto a punto mediante la SDR a desarrollar y cualquier otra SDR, por medio de diferentes tipos de modulaciones establecidas a través de GNU-RADIO.

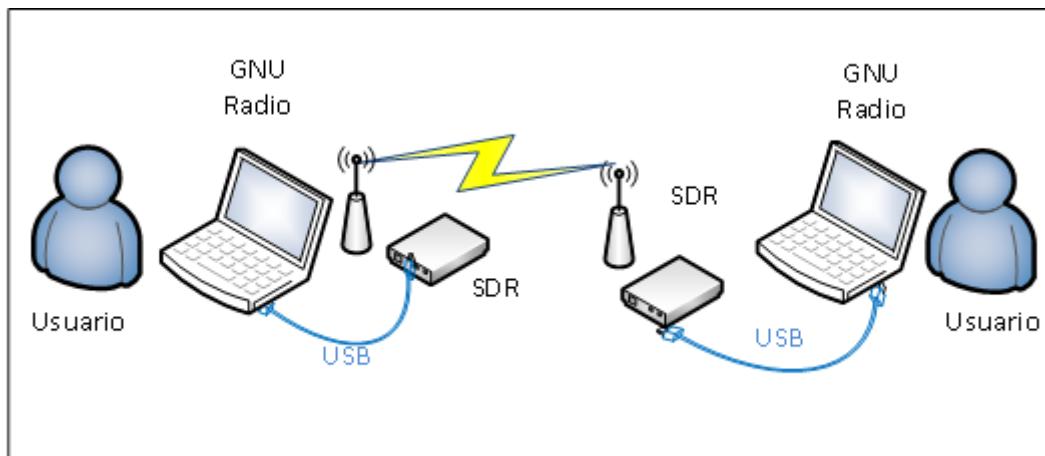


Figura 3: Esquema de funcionamiento comunicación punto a punto

- Conexión USB 3.0
- Frecuencia de operación de 100 MHz hasta 3 GHz
- Ancho de banda de 40 MHz
- Memoria DDR3

Por lo tanto el requerimiento de esta tesis es diseñar una SDR la cual tenga las características enunciadas anteriormente.

Las etapas de construcción y puesta en marcha quedan como trabajo a futuro.

## 1.6 Metodología de diseño

Para realizar este proyecto se utilice una metodología de diseño Top-Down. La cual consiste en capturar una idea con un alto nivel de abstracción, implementarla partiendo de la misma, e incrementar el nivel de detalle según sea necesario. El sistema inicial se va subdividiendo en módulos, estableciendo una jerarquía. Cada módulo se subdivide hasta llegar a los componentes primarios del diseño.

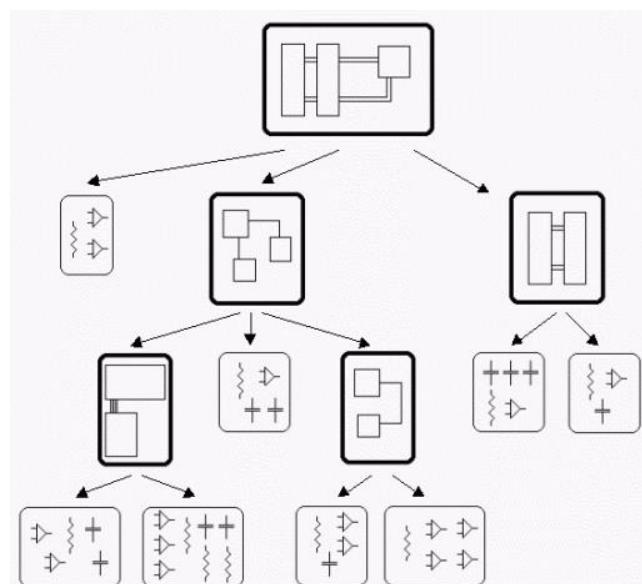


Figura 4: Diagrama metodología de diseño Top-Down

Utilizando la idea de este método se desarrolló el diseño de la SDR dividiendo al mismo en las siguientes etapas:

- Diseño Modular según funciones
- Diseño Modular específico
- Selección de Componentes
- Diseño del esquemático
- Diseño del PCB
- Simulaciones

## Capítulo 2

### 2. Diseño

#### 2.1 Arquitectura general según funciones

La arquitectura de la SDR debe soportar la evolución a nuevas plataformas de servicios, softwares y hardware.

La arquitectura de las SDR debe tener las siguientes propiedades:

- Permitir que el funcionamiento del sistema sea definido por software, esto significa, que una porción del sistema se implementara mediante programación y/o reconfiguración del hardware. Permitiendo que el hardware del sistema pueda reutilizarse para distintas aplicaciones sin la necesidad de modificaciones físicas del mismo.
- Tener la capacidad de captar y transmitir las señales de RF en el rango de operación de las mismas. Estas señales son transmitidas en una frecuencia intermedia (por cuestiones de tamaño de antena, potencia etc.). Por lo tanto la arquitectura debe permitir bajar esta frecuencia a una frecuencia de operación (por ejemplo banda base).
- Poder digitalizar las señales recibidas para su posterior procesamiento y almacenado.

La figura 5 muestra una arquitectura en donde todos sus bloques son lógicos. En las arquitecturas lógicas se da todo el detalle posible sin especificar como es implementada la misma. En cambio en las arquitecturas físicas se detalla cómo esta implementada la misma, componentes, software etc.

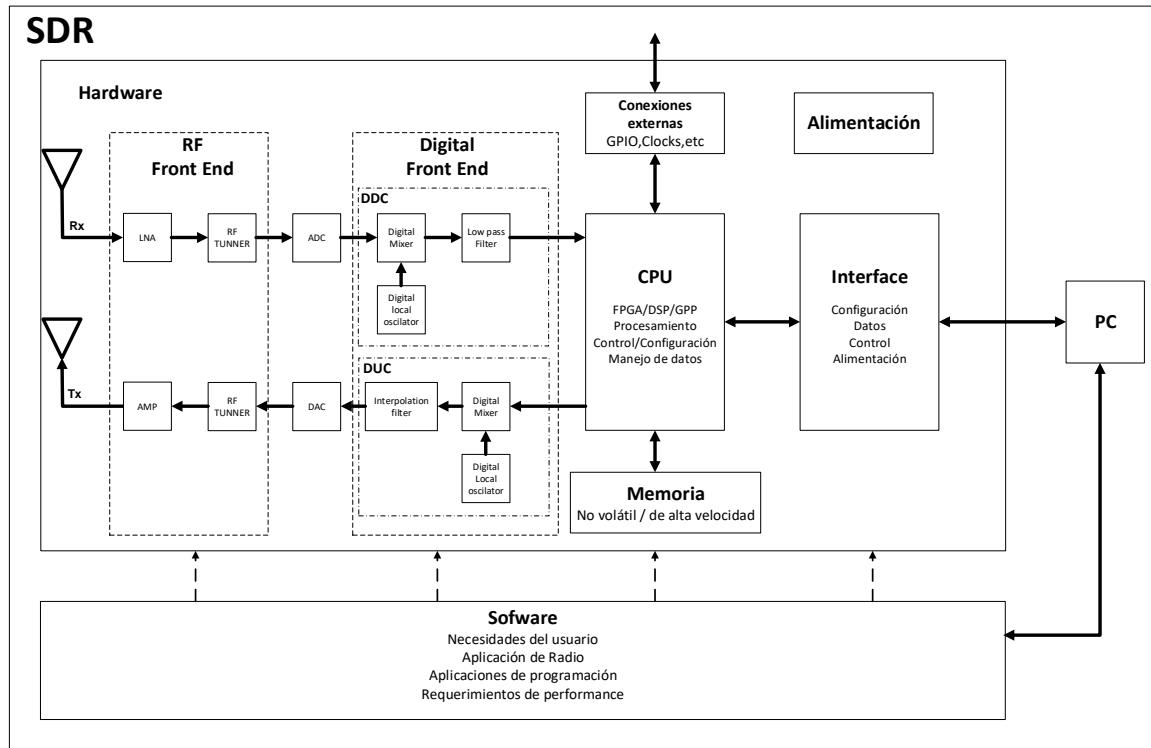


Figura 5: Diagrama en bloques de la arquitectura general de la SDR

### 2.1.1 Antenas

Una antena transmisora transforma energía eléctrica en ondas electromagnéticas, y una receptora realiza la función inversa. Para las SDR que son diseñadas para múltiples modos y/o una banda amplia de frecuencia la selección de la antena es algo crucial. En la mayoría de los casos para un rango amplio de frecuencia o múltiples modos, es necesario utilizar más de una antena para cubrir estos rangos/modos.

Las características fundamentales de una antena son su ganancia y al ancho del rayo al cual su potencia cae a la mitad (-3 dB). La ganancia es una medida de cuento de la potencia de entrada es radiada en una dirección particular. La siguiente figura muestra un ejemplo de patrón de radiación de una antena en función del ángulo.

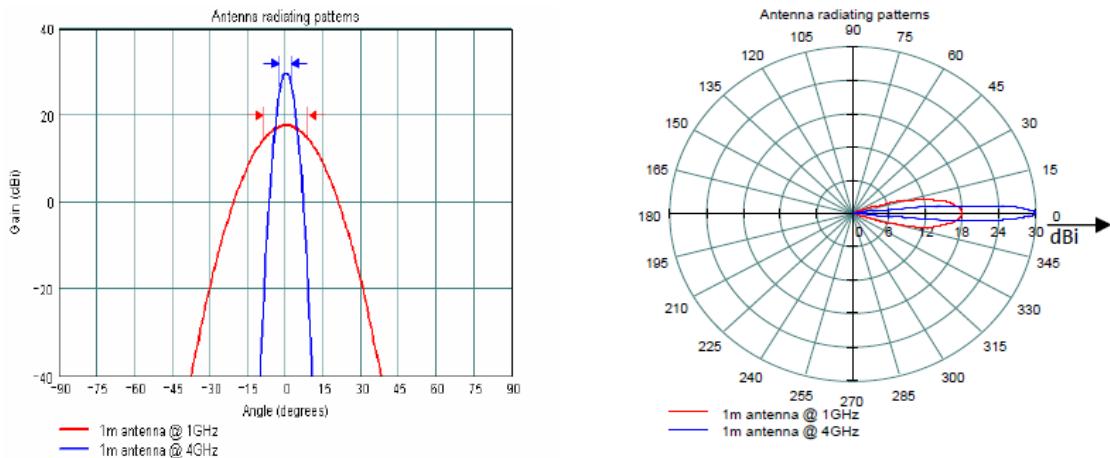


Figura 6: Ejemplo patrón de radiación de una Antena

Algunas de las antenas que pueden ser utilizadas en una SDR las siguientes: mini whip antenna, monopole antenna, telescopic antenna, omnidirectional antenna, adaptative array smart antenna, etc.

A menudo las antenas son denominadas inteligentes, debido a su capacidad de seleccionar una banda de frecuencia, seguimiento móvil o cancelación de interferencia.

Las antenas isotrópicas conceptualmente radian de igual modo en todas las direcciones.

### 2.1.2 RF Front-End

Su principal función es transmitir y recibir señales en varias frecuencias de operación. Otras de sus funciones dependiendo el sentido de la señal son las siguientes:

- En modo Tx: La señal analógica es llevada a una frecuencia intermedia (IF) por medio de un mixer y oscilador local (LO). Esta es amplificada para luego ser transmitida.
- En modo Rx: En el camino de recepción la señal de RF es capturada. Luego la señal pasa a través de un amplificador de bajo ruido (LNA) el cual se encuentra lo más cercano posible a la antena, para amplificar señales débiles y minimizar el nivel de ruido. Esta señal amplificada es llevada a una frecuencia intermedia (IF) por medio de un mixer y un oscilador local (LO).

Las antenas son conectadas con el Front End de RF utilizando circuitos de adaptación para obtener una transferencia óptima de potencia.

### 2.1.3 DAC y ADC.

El Conversor Analógico-Digital (ADC) en el camino de recepción y el Conversor Digital-Analógico (DAC) en el camino de transmisión forman un puente entre el Front End Digital y el Front End de RF.

La performance de un ADC puede ser descripta por varios parámetros, por ejemplo alguno de ellos:

- Signal-to-Noise Ratio (SNR).
- Resolución: número de bits por muestra.
- Spurious Free Dynamic Range (SFDR).
- Disipación de potencia.

La performance de un DAC puede ser descripta por varios parámetros, por ejemplo alguno de ellos:

- Resolution
- Full scale range (FSR)
- Offset error
- Gain error
- Differential Nonlinearity
- SNR

### 2.1.4 Digital Front-End

El Digital Front-End se lleva a cabo las funciones de Digital Down Converter (DDC) y Digital Up Converter (DUC).

Un DDC convencional está compuesto por tres secciones:

- Un mixer digital
- Un oscilador local digital
- Un filtro pasa bajos

Mediante el mixer digital y el oscilador local se traslada las muestras digitales en IF a BB. El filtro pasa bajo limita el ancho de banda de la señal y actúa como un diezmador. Los DDC incluyen una gran cantidad de hardware para realizar dicho trabajo, como por ejemplo: multiplicadores, sumadores y shift registers.

El proceso de diezmado consiste en quedarse con una muestra de cada N muestras de la señal, como muestra la figura 7. Si la tasa de muestras diezmadas se mantiene mayor al doble del ancho de banda de salida no se perderá información (Nyquist)

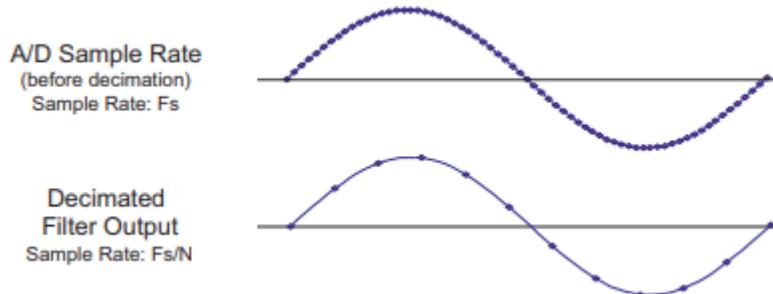


Figura 7: Proceso de diezmando

Un DUC convencional está compuesto por tres secciones:

- Un mixer digital
- Un Oscilador local digital
- Un filtro de interpolación

Mediante el mixer digital y el oscilador local se traslada la señal de BB a una IF. Esta IF es determinada por el oscilador local. Por lo general la frecuencia de muestreo ( $F_s/N$ ) de la señal proveniente del BBP es menor a la frecuencia ( $F_S$ ) que necesita el DAC. Para compensar este problema se utiliza un filtro de Interpolación mediante el cual se incrementa la frecuencia de muestreo de la señal en banda base en un factor  $N$ . La figura 8 muestra el efecto del filtro de interpolación.

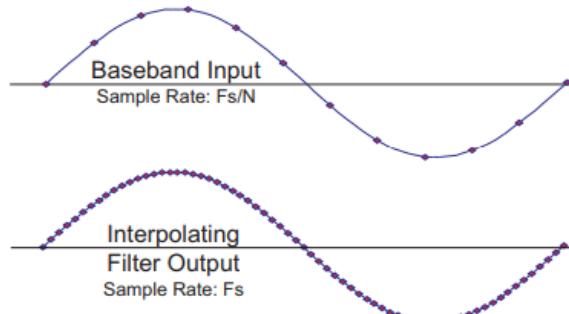


Figura 8: Proceso de interpolación

### 2.1.5 CPU

Sus principales funciones son: control, configuración, manejo de datos e interconexión entre los diferentes módulos (memoria, gpio, external interface, digital front-end).

Además también se puede realizar procesamientos, algoritmos, encriptación/desencriptación etc.

Entre los posibles CPU a utilizar en un SDR se encuentran las siguientes opciones:

**ASIC:** Es un circuito integrado hecho a la medida para un uso en particular, en vez de ser concebido para propósitos de uso general. Son utilizados cuando se necesita una alta tasa de datos a un consumo de energía razonable.

**FPGA:** Son integrados capaces de ser reprogramados al contener una matriz de bloques lógicos reprogramables. Pueden soportar altas tasas y altos niveles de consumo de energía.

**GPP:** Son procesadores de propósito general los cuales pueden procesar tasas bajas/medias de señales en tiempo real. Estos son ideales cuando el consumo de energía no es de gran importancia.

**DSP:** Un procesador digital de señales, es un sistema basado en un procesador o microprocesador que posee un conjunto de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Por lo general no ofrecen suficiente tasa o bajo consumo como las demás opciones. Son utilizados cuando los GPP no son lo suficientemente rápidos, el consumo es demasiado o los desarrolladores no tienen experiencia en FPGA.

En la siguiente figura se muestra una comparativa en cuanto: tasa de datos, consumo, esfuerzo de desarrollo y costo entre las diferentes soluciones para la CPU:

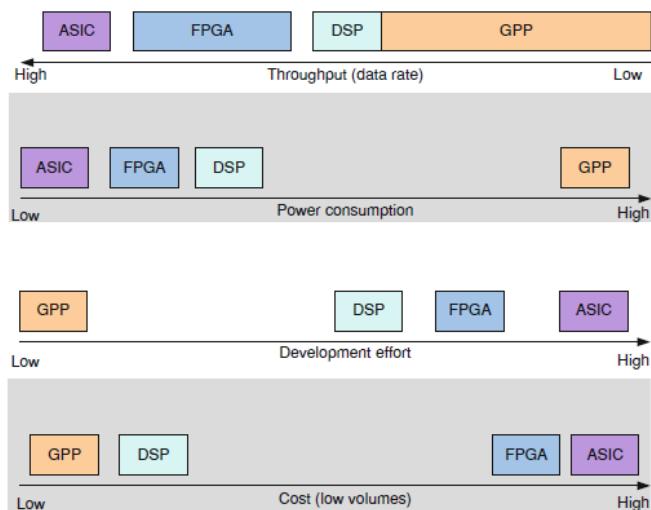


Figura 9: Comparativa entre los posibles CPUs

### 2.1.6 Memoria

La memoria del sistema puede estar compuesta por memorias de alta velocidad y/o memorias no volátiles. Las mismas proveerán recursos de cómputo y capacidad de almacenamiento de datos para el sistema.

### 2.1.7 Conexiones externas

Este módulo contiene las conexiones físicas de la SDR con elementos externos, como por ejemplo: GPIO, alimentación, Clock externo, etc.

### 2.1.8 Interfaces

Es el bloque que encargado de las interconexiones de la SDR con el resto de sistema, para las líneas de datos, configuración, programación y control.

### 2.1.9 Alimentación

Su función es proveer la alimentación a los módulos nombrados anteriormente. En el mismo se realizaran las diferentes conversiones de DC a DC que sean necesarias.

## 2.2 Arquitectura definida

En base a las funciones expuestas en la sección anterior, los requerimientos y los componentes disponibles en el mercado, diseñe la siguiente arquitectura para la implementación de la SDR.

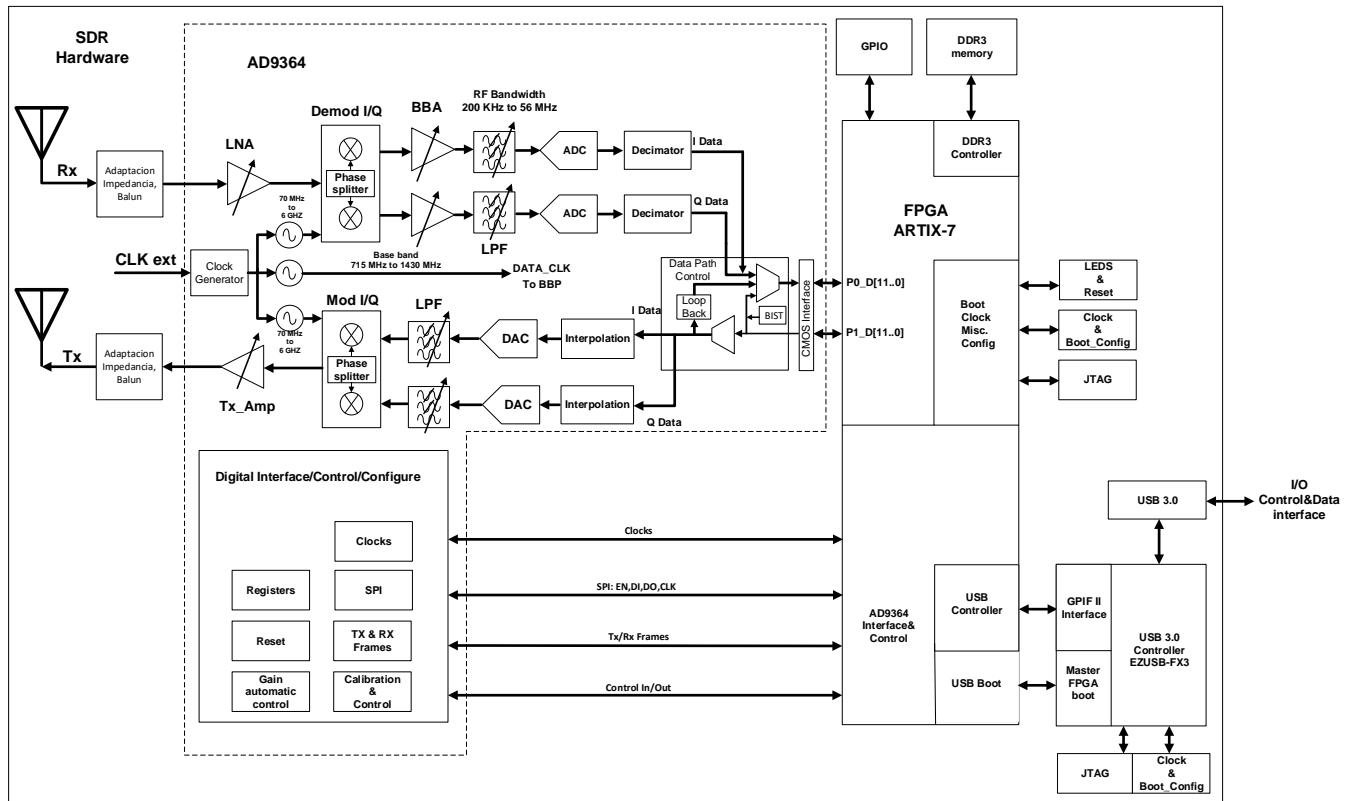


Figura 10: Diagrama en bloques de la Arquitectura definida de la SDR

A continuación se detalla la información sobre cada módulo de esta arquitectura.

### 2.2.1 RF Front End & Digital Front End

Optó en utilizar un transceiver integrando ambos Front End en un solo IC, lo que significa una menor cantidad de componentes y superficie. Este fue el primer componente seleccionado, ya que el mismo determina las principales características de la SDR (ancho de banda, rango de frecuencia, etc.). Además el mismo fija requerimientos para las demás etapas del sistema (alimentación a utilizar, tasa de datos necesaria, etc.).

En base a los requerimientos estipulados los posibles transceiver encontrados en el mercado fueron los siguientes:

Integrado	fabricante	Ancho de banda	Rango de Frecuencia	Costo(\$USD)
AD9361	Analog Devices	56 MHz	70 MHz-6GHz	214.31
AD9364	Analog Devices	56 MHz	70 MHz-6GHz	159.20
LMS6002DFN	Lime MicroSystems	28MHz	0.3-3.8GHz	35.00

Tabla 2: Transceivers disponibles en mercado.

Descarte el transceiver LMS6002 por los siguientes motivos:

- No cumple con el ancho de banda (requerimiento que podría ser flexible debido a su bajo precio).
- Ya existe una SDR open source que utiliza este transceiver (proporcionan: gerbers, esquemáticos y proyecto entero en Altium Designer).
- Escases de información en cuanto a funcionamiento y guías de diseño.

Decidí utilizar el AD9364 y dejar al AD9361 para una posible mejora a futuro convirtiendo a la SDR en una MIMO. En particular ya existen SDRs open source con este transceiver (empresa Ettus) pero solo comparten información sobre el esquemático.

Este dispositivo combina un Front End de RF con un flexible mezclador de señales en banda base y un sintetizador de frecuencias integrado. Simplificando el diseño a una interface digital de configuración por medio de un procesador.

Por medio del bus SPI se tiene todo el control de toda la parte digital del AD9364. Cada registro SPI es de 8 bit de ancho y contienen: bits de control, monitores de estado y otras configuraciones que controla todas las funciones del dispositivo (filtrado, ganancia, etc.).

En la siguiente figura exponemos que porción de la arquitectura de la SDR será cubierta por este IC:

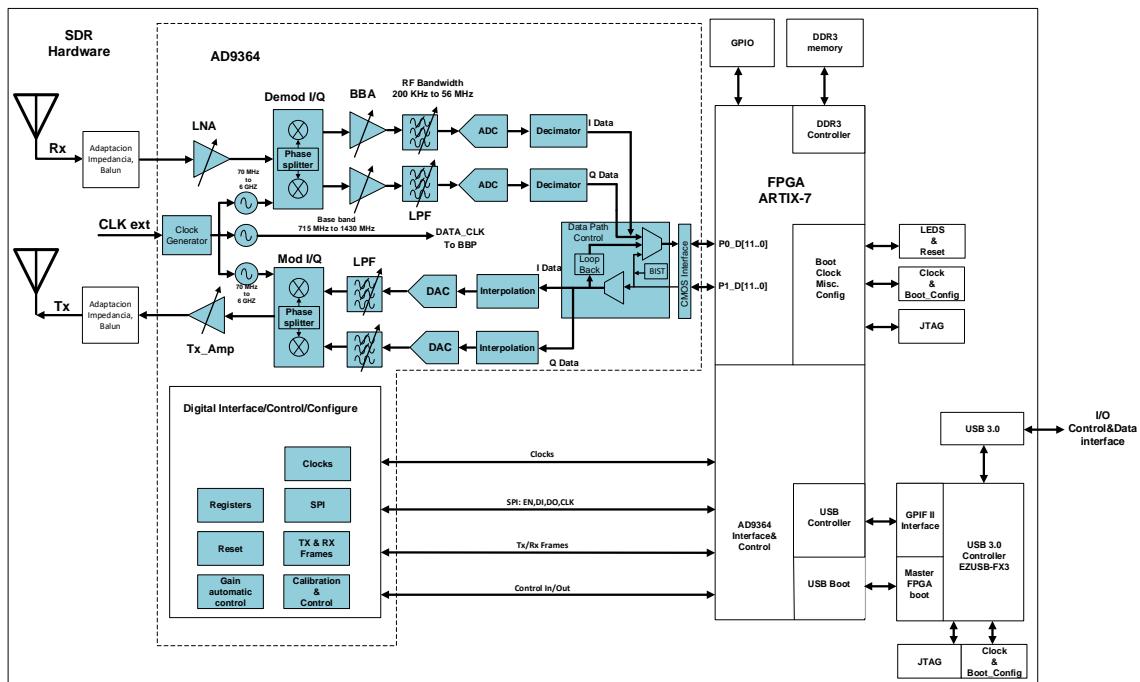


Figura 11: Diagrama de arquitectura definida, sección transceiver

Este conjunto de módulos tendrá las siguientes funciones:

- Convertir las señales de RF a BB para la recepción
- Convertir las señales BB a RF para la transmisión
- Acondicionamiento de las señales

A continuación se explicara la función y características de cada uno de los módulos que componen al transceiver:

- **Analog to Digital conversión (ADC):** Convierte al dominio digital de las señales en banda base entregadas por todo el camino de recepción. El mismo es de 12 bits.
- **Digital to Analog conversión (DAC):** Conversión al dominio analógico de las señales en banda base a ser entregadas al camino de transmisión (up-conversión) por la antena de la SDR. El mismo es de 12 bits.
- **Demod I/Q:** Convierte la señal de RF que entrega la cadena de recepción del RFFE a banda base para su posterior digitalización y procesamiento.
- **Mod I/Q:** Convierte la señal en banda base a transmitir a RF para ser luego amplificadas por el RFFE en su cadena de transmisión.
- **Diezmado:** El proceso de diezmado consiste en quedarse con una de cada n muestras, siempre dentro de los límites de Nyquist. Este límite es determinado por el ancho de banda de los filtros pasa bajo. Este proceso es útil ya que baja la tasa de datos hacia procesador y demanda menos almacenamiento.
- **Interpolación:** El proceso de interpolación consiste en aumentar la tasa de muestreo en n. Este proceso no cambia la frecuencia de la señal solo su tasa de muestreo.
- **Low noise amplifier (LNA):** Los amplificadores de bajo ruido amplifican señales de bajo poder sin degradar su relación señal a ruido. Este es un componente clave en el Front-End de RF para reducir el ruido, ya que la figura de ruido es dominada por las primeras etapas del receptor.
- **Low Pass Filter (LPF):** Los filtros pasa bajo dejan pasar las señales con una frecuencia menor a la frecuencia corte y atenúa las señales mayores a la misma. Algunos los filtros pasa bajos que componen esta etapa:

En Rx:

- Un filtro de tercer orden Butterworth, el cual tiene una frecuencia de corte es programable en el rango de 200 kHz a 39.2 MHz.
- Un filtro con un polo simple, el cual tienen una frecuencia de corte programable en el rango de 1 MHz a 70MHz

En Tx:

- Un filtro de tercer orden Butterworth, la frecuencia de corte es programable en el rango de 625 kHz a 32 MHz.
- Un filtro con un polo simple, la frecuencia de corte es programable en el rango de 2.7 MHz a 100MHz

- **Generador de señales BITS:** Este integrado dispone en un generador de señales interno con el cual se puede injectar señales tanto en el puerto de salida de datos de Tx como en el de entrada de Rx. Este es útil como herramienta de debugging a la hora de probar la conectividad de los puertos de Tx y Rx.
- **Digital Interface:** Por medio este módulo se realiza la interface con el BBP, pudiendo configurar completamente el transceiver. Cada uno de los módulos nombrados anteriormente (ADC, filtros, etc.) esta mapeado a una dirección memoria (registros), los cuales se van modificando por comandos a través de SPI. Mediante estas modificaciones se puede configurar entre otras cosas:
  - Ganancia de los amplificadores.
  - Frecuencia de corte filtros.
  - Frecuencia de los osciladores locales.
  - Modo de operación.
  - Calibración.
- **Adaptación de impedancia:** Mediante circuitos con componentes discretos y/o distribuidos, se logra una red de adaptación para obtener máxima transferencia de potencia tanto en la transmisión como en la recepción.

Información más detallada del transceiver puede ser encontrada en [3].

### 2.2.2 USB 3.0

La mayor tasa de datos del sistema está determinada por el transceiver, el cual tiene una tasa máxima de aproximadamente 188 MB/seg (66.44 MHz de clock máximo y 24 líneas de dato). Para este proyecto una conexión USB 2.0 (60

MB/seg idealmente) no es suficiente para operar a la máxima tasa de datos por lo tanto se decidió utilizar una conexión USB 3.0 (300 MB/seg).

### 2.2.2.1 Información general USB 3.0

Las especificaciones del USB 3.0 permiten una tasa de transferencia teórica máxima de 5 Gbps (velocidad práctica de hasta 300 Megabytes), la cual es 10 veces más que la tasa del USB 2.0 (High Speed). La arquitectura del USB 3.0 contiene tres capas: physical layer, link layer y protocol layer.

### Physical Layer

La capa física hace referencia al puerto de conexión y al cable de conexión, mediante el cual se realiza el upstream y downstream. Los cables diferenciales de transmisión y recepción de datos del USB 3.0 están mallados y separados. Los cables USB 3.0 incluyen a los utilizados en el USB 2.0 como se muestra en la siguiente figura:

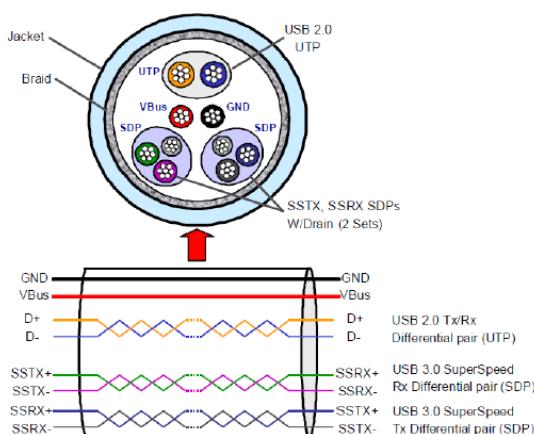


Figura 12: Cables USB 3.0

El bus SuperSpeed emplea un enfoque dual y simple, el cual permite transmitir y recibir paquetes simultáneamente. Por ejemplo durante una ráfaga de transacciones, el dispositivo puede recibir datos desde el host y devolver una confirmación asociada con los datos recibidos.

### Link Layer

La link layer es la responsable de mantener una comunicación confiable y robusta entre el host y el dispositivo. Este mantenimiento lo realiza utilizando una máquina de estados llamada “Link Training and Status State Machine” (LTSSM).

## Protocol Layer

La capa de protocolo maneja las reglas de comunicación entre el host y el dispositivo.

### 2.2.2.2 Controlador USB 3.0 CYUSB3014

El dispositivo que decidí utilizar como controlador del USB3.0 es el CYUSB3014 de la compañía Cypress. Este es un controlador de periféricos de SuperSpeed que provee características integradas y flexibles.

Este controlador es totalmente configurable, paralelo y posee una interface programable llamada GPIF II, mediante la cual se puede conectar a cualquier tipo de procesador, ASIC o FPGA.

A continuación se mostrara una vista general de este dispositivo:

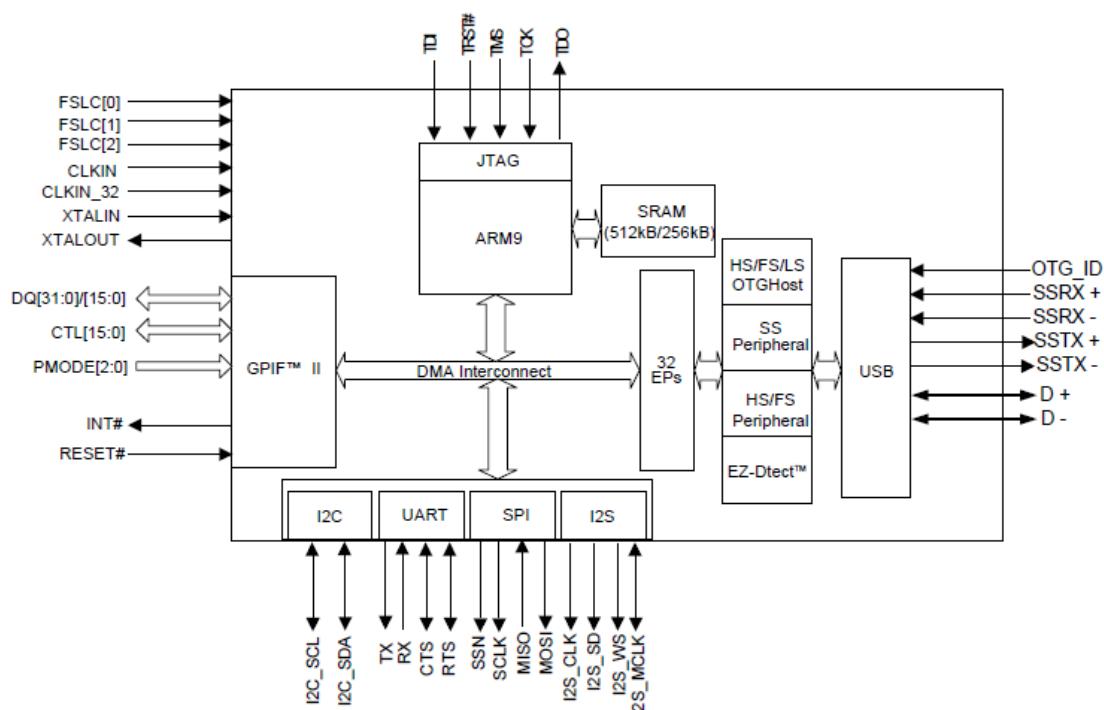


Figura 13: Diagrama en bloques controlador USB 3.0

## CPU

El dispositivo cuenta con un CPU de 32 bits, un ARM92EJ-S de 200 MHz. Este core tiene acceso directo a una memoria de instrucciones de 16 KB y una de 8 KB para datos. Además tiene dos memorias cache (una para instrucciones y otra para datos) de 8KB cada una.

## DMA

El dispositivo habilita eficientes y flexibles transferencias del DMA entre varios periféricos (USB, GPIF II, SPI, I2S y UART).

## JTAG Interface

Es una interface estándar de 5 pines para conectar el JTGA con el debug firmware a través del CPU. Cualquier herramienta estándar de debugging para ARM926EJ-S puede ser utilizada para este dispositivo.

## Boot Options

El dispositivo tiene integrado una ROM de 32 KB, la cual contiene el bootloader, permitiendo que se baje la imagen desde diferentes fuentes. El modo de boot se selecciona mediante los pines PMODE, la siguiente tabla muestra las diferentes configuraciones de boot.

PMODE[2:0]			Boot Option
PMODE[2]	PMODE[1]	PMODE[0]	
Z	0	0	Sync ADMUX (16-bit)
Z	0	1	Async ADMUX (16-bit)
Z	0	Z	Async SRAM (16-bit)
Z	1	1	USB Boot
1	Z	Z	I2C
Z	1	Z	I2C; on failure, USB Boot is enabled
0	Z	1	SPI; on failure, USB Boot is enabled
0	0	0	S0-port (eMMC); On Failure, USB Boot is enabled

Tabla 3: Opciones de Booting USB 3.0

Z = Pin is floating; left unconnected.

En este caso se decidió que el boot sea por medio de una memoria EEPROM y en caso de fallo por medio del USB (PMODE [2:0] = 000).

## CLOCKING

Este dispositivo permite conectar un cristal entre los pines XTALIN y XTALOUT o conectar un clock externo en pin CLKIN. Estos pines pueden dejarse sin conectar de no ser usados. La frecuencia de cristal soportada es de 19.2 MHz.

Mediante los pines FSLC [2:0] se determina que opción fue seleccionada: cristal o clock externo. La siguiente tabla muestra las diferentes opciones de cristal o clock externo:

FSLC[2]	FSLC[1]	FSLC[0]	Crystal/Clock Frequency
0	0	0	19.2-MHz Crystal
1	0	0	19.2-MHz Input CLK
1	0	1	26-MHz Input CLK
1	1	0	38.4-MHz Input CLK
1	1	1	52-MHz input CLK

Tabla 4: Selección de clock USB 3.0

Como fuente de clock opté por la opción de utilizar el cristal de 19.2 MHz (FSCL[2:0] = 000).

## GPIF II

La interface GPIF II es una máquina de estado programable la cual permite una interface flexible con la CPU, esta puede funcionar como master o slave.

GPIF II tiene las siguientes características:

- Funciona como master o slave.
- Ofrece 256 estados de firmware programables.
- Soporta 8-bit, 16-bit, and 32-bit de bus de datos paralelo
- Frecuencia de interface hasta 100 MHz
- Soporta 14 pines de control configurables cuando se utiliza un bus de datos de 32-bits; todos los pines de control pueden ser entrada, salida o bidireccionales
- Soporta 16 pines de control configurables cuando se utiliza un bus de datos de 16 o 8 bits; todos los pines de control pueden ser entrada, salida o bidireccionales

Una de las implementaciones más populares de GPIF II es la interface sincrónica: Slave FIFO. Esta interface es utilizada en aplicaciones en donde el dispositivo externo que está conectado a este controlador lee o escribí datos desde el.

Información más detallada sobre este controlador puede ser encontrada en [4].

### 2.2.3 Memoria DDR3

La inclusión de una memoria externa no estuvo en las consideraciones iniciales, se decidió agregarla ya que dota al sistema con la posibilidad de almacenar los datos, ya sean procesados por la FPGA o directamente los datos crudos capturados por el transceiver. El costo de esta inclusión fue de un incremento de en la complejidad del sistema en: una mayor densidad de ruteo de señales en la placa,

una ramificación extra de tensión de alimentación y un pequeño coste económico extra.

Se utilizó una de memoria DDR3L con una capacidad total de 256MB .El modelo de la memoria es el MT41J128M16JT-125 fabricado por Micrón. El mismo cuenta con 14 líneas de dirección (A[13..0]), 3 líneas para selección de banco (BA[2..0]), 16 líneas de datos [15..0]. Al utilizar un solo chip de memoria no es necesario utilizar terminaciones (on die terminations). La frecuencia máxima de operación es de 333 MHz.

Para un correcto funcionamiento de la memoria, un controlador de memoria y una interface con la capa física debe ser incluido en la FPGA. El modo más fácil de implementar esto en una FPGA artix-7, es utilizando el generador de cores para una interface de memoria con la herramienta MIG (Memory Interface Generator) Wizard.

Información más detallada sobre esta memoria puede ser encontrada en [5].

### 2.2.4 CPU

Entre las opciones mencionadas en la sección 2.1.5, se decidió utilizar una FPGA como CPU por las siguientes razones:

- Gran tasa de datos,
- Gran ancho de banda
- Flexibilidad.
- Paralelismo.
- Reconfigurable

#### 2.2.4.1 Selección de FPGA

Primero se seleccionó que fabricante utilizar, en el mercado de las FPGAs existen 2 grandes empresas Xilinx y Altera. Se decidió utilizar Xilinx debido a que posee la mayoría del mercado de las FPGA.

Una vez seleccionado el fabricante, se decidió utilizar la serie 7 de Xilinx ya que es la más actual y utilizan Vivado como entorno de programación. Esta serie tiene las siguientes gamas:

- Spartan-7
- Artix-7
- Kintex-7
- Virtex-7

Se decidió utilizar la Artix-7 ya que su costo era mucho menor que las de gama alta (Kintex-7 y Virtex-7). La Spartan-7 se descartó debido a que la Artix-7 poseía

mayor número de DSP slices a un precio similar. Para ver las diferentes características de estas FPGAs ver [6]. Luego para seleccionar el modelo adecuado de FPGA Artix-7 se tuvo en cuenta las siguientes características:

**Recursos FPGA:** Para determinar los recursos necesarios de la FPGA se realizaron estimaciones (sobredimensionando la misma), tomando en cuenta los siguientes parámetros:

- Se realizó un estudio de las capacidades de las SDR comerciales:

Modelo	FPGA	Programmable Logic Gates
USRP B200mini [7]	Xilinx Spartan-6 XC6SLX75	75k
LimeSDR [8]	Altera Cyclone IV EP4CE40F23	40k
bladeRF xA4 [9]	Altera Cyclone V	49K

Tabla 5: FPGA de SDR comerciales

- La cantidad de recursos utilizados por la implementación de un receptor QAM como el mostrado en la siguiente figura:

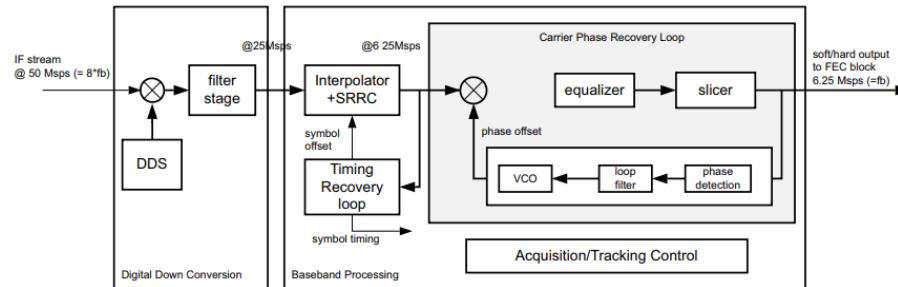


Figura 14: Diagrama en bloques receptor QAM

Los recursos necesarios para esta implementación son los siguientes:

Recursos utilizados	
SLICE	304
LUT	987
FF	322
DSP	5
BRAM	2
SRL	86

Tabla 6: Recursos necesarios para receptor QAM

Para más detalle sobre esta implementación y especificaciones del receptor ver la referencia [10].

- La cantidad de recursos necesarios para implementar un controlador MicroBlaze application processor [11]. Los recursos necesarios para implementar este procesador son los siguientes:

ModeloFPGA	XC7A200T-3
Logic Cells	7000
% Utilizacion	4%

Tabla 7: Recursos necesarios para application processor

**Temperatura de operación:** Comercial industrial (0º a 85 ºC), extended (0º a 100 ºC), industrial (-40º a 100 ºC) o expanded (-40º a 125 ºC)

**Cantidad de I/O:** la cantidad de pines debe ser suficiente para poder interconectar todos los módulos.

**Grado de velocidad:** Representa el tiempo que tardan los circuitos lógicos en atravesar el dispositivo, es decir determina la frecuencia máxima que soporta la FPGA. Se seleccionó el grado -1 (gama baja, frecuencia máxima = 464 MHz), ya que a medida que se mejora este grado de velocidad el precio de la FPGA aumenta.

**Pitch:** El mismo determina la separación entre cada bola del encapsulado BGA, esta puede ser de; 1mm, 0.8mm y 0.5mm. Se decidió utilizar un pitch de 1.mm para tener más espacio entre las vías a la hora de realizar el ruteo del PCB y además es más sencilla de soldar en comparación a las de pitch más chico.

Teniendo en cuenta todas estas características se decidió utilizar el modelo de FPGA XC7A100T-1FTG256C. La siguiente figura muestra el significado de cada letra y numero del modelo seleccionado.

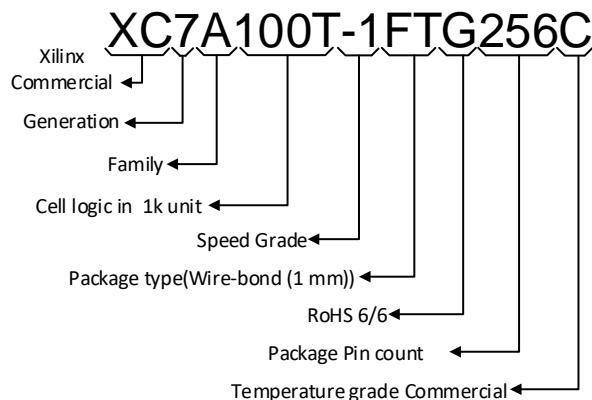


Figura 15: Modelo de FPGA seleccionado

Una estimación del porcentaje de ocupación que representan las consideraciones previas (MicroBlaze application processor, modulador y demodulador QPSK) en la FPGA seleccionada es:

FPGA XC7A100T	
Modulo	%Utilización
AP MicroBlaze	8%
Mod QPSK	2%
Demod QPSK	2%
Total	12%

Tabla 8: Estimación de recursos a utilizar

Se estimó que los recursos utilizados por el transmisor QPSK son similares a los del receptor. Como se puede apreciar en la tabla 8, la cantidad de recursos de la FPGA están sobredimensionados, estimando que estos recursos serán suficientes para implementar otros módulos tales como: protocolos, procesamientos, interfaces, controladores, etc.

#### 2.2.4.2 Pin Out y encapsulado

El modelo de FPGA seleccionado está compuesto por un encapsulado Wire-Bond Fine-Pitch Thin BGA de alta densidad con un total de 256 pines, que se encuentran distribuidos en forma de rejilla cuadrada de 16 filas por 16 columnas. El pitch o separación entre los pines es de 1mm. Este tipo de encapsulado es apropiado cuando se dispone de un gran número de I/O, en este caso el modelo dispone de un total de 170. El total de pines de I/O está distribuido en bancos. La figura 16 muestra el mapa de pines de I/O de la FPGA. Se pueden distinguir los diferentes bancos por su color y por su numeración.

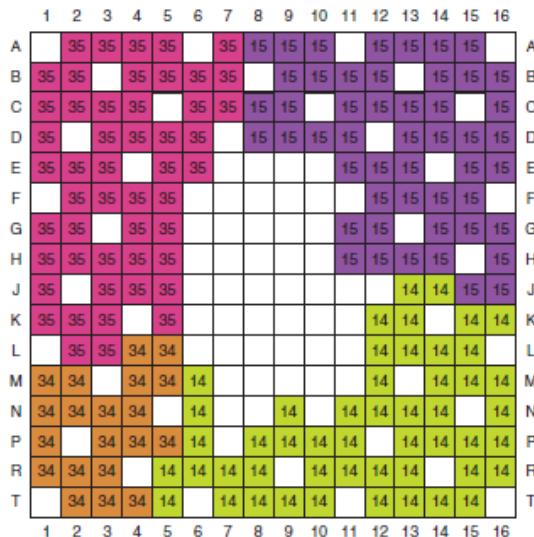


Figura 16: Pin Out I/O FPGA

#### 2.2.4.3 Función

### Control de aplicación.

Esta tarea se refiere a la implementación de la lógica y máquinas de estados necesarios que definen la o las aplicaciones de la SDR. Definiéndose como la unidad central de control que implementa los modos de operación de la SDR. A través de esta unidad, se brindaran las interfaces lógicas para el control, configuración y monitoreo externo de la SDR.

### Procesamiento de señal

Compuesto principalmente por una FPGA con una gran capacidad para el procesamiento digital de señales, memoria rápida y de gran volumen, gran cantidad de pines, interfaces de alta velocidad y capacidad para ser reprogramada.

Estará orientado, como su nombre lo indica, al procesamiento digital que demande la aplicación que quiera embeberse en la SDR. Algunas de estas aplicaciones pueden ser:

- Procesamiento en banda base.
- Manejo de datos.
- Encriptar/Des encriptar.
- Aplicar diferentes tipos de modulación a los datos I/Q.

### 2.2.4.4 Configuración FPGA

La FPGA seleccionada es configurada cargando las aplicaciones en su memoria interna. En este caso la configuración de esta memoria puede ser cargada por medio de otro procesador (ubicado en el controlador del USB 3.0) de forma serial. Este modo de configuración se lo conoce como “Slave-serial configuration mode”.

También se dispone de un JTAG, el cual es una interfaz de 4 pines usada frecuentemente con finalidades de testeo y depuración. La conexión JTAG no sólo sirve para configurar la FPGA sino que es totalmente compatible con el protocolo de testeo de dispositivos definido en el estándar IEEE 1149.1.

Las configuraciones de las FPGA de Xilinx son almacenadas en “CMOS configuration latches” (CCLs). Por lo tanto deben ser reconfigurada después de cada apagado.

### 2.2.4.5 Alimentación FPGA

Para determinar la alimentación necesaria de la FPGA artix-7 se utilizó la herramienta “Xilinx Power Estimator”, de estimación de consumo que provee Xilinx. Otorgando como datos de salida la corriente necesaria en cada banco para su correcto funcionamiento.

También se debe tener en cuenta la secuencia de encendido recomendada:

**VCCINT-> VCCBRAM ->VCCAUX ->VCCO**

### 2.2.5 Alimentación SDR

La función que se define en este módulo es la de proveer las tensiones secundarias que serán utilizadas por los demás módulos de la SDR.

Dado que el principal elemento de la SDR, el transceiver AD9364, recomienda utilizar convertidores de DC a DC de Analog Devices, se decidió que la mayoría de los integrados de alimentación sean de la misma empresa. Permitiendo así realizar una simulación casi completa de la sección de alimentación ([Anexo D1](#)). Para alimentar la mayor parte del sistema se utilizó una de las soluciones que ofrece Analog Devices para la alimentación de FPGAs. Dicha solución utiliza el integrado ADP5054, el cual combina 4 buck regulators en un solo integrado.

Además se estimaron los consumos de los principales integrados, como se indica en la siguiente tabla:

Integrado	Tension	Corriente	Potencia
FPGA	1.0 V	2.919 A	2.919 W
	1.8 V	0.274 A	0.493 W
	1.35 V	0.125 A	0.168 W
Transceiver	1.8 V	0.008 A	0.014 W
	1.3 V	0.55 A	0.715 W
USB 3.0	1.2 V	0.127 A	0.1524 W
	1.8 V	0.065 A	0.117 W
DDR3	1.35 V	0.609 A	0.822 W

Tabla 9: Estimación del consumo por etapa

En donde:

- La estimación de consumo de energía de la FPGA se realizó utilizando la herramienta XPE de Xilinx. Para esta estimación se consideraron casos extremos, como el 100% del uso de los recursos y alta tasa de cambio en las señales (Toggle rate). Para más información ir al [Anexo C1](#) y [12].
- Para la estimación del transceiver se utilizó el peor caso de consumo provisto por la hoja de datos del mismo. En este caso la trasferencia de datos entre el BBP y el transceiver se realizan con un clock de 30.72 MHz (1.8V@ 8mA). La transmisión/ recepción es en modo FDD a 5 Ghz, con un ancho de banda de 5 MHz y 7 dBm de potencia (1.3V@550 mA) [13].
- Para el USB3.0 se utilizó una tabla de consumos típicos que provee Cypress (ver referencia [14]).
- Para obtener el consumo de la memoria DDR3 se utilizó una hoja de cálculo que provee Micrón y se tomaron las corrientes máximas de la misma (ver referencia [15]).

La siguiente tabla indica la corriente necesaria por cada tensión:

Tension	Corriente
1.0 V	2.919 A
1.8 V	0.2885 A
1.35 V	0.734 A
1.3 V	0.55 A
1.2 V	0.127 A

Tabla 10: Corriente necesaria por tensión

A partir de todas las estimaciones y consideraciones enunciadas anteriormente se diseñó la siguiente distribución de tensiones:

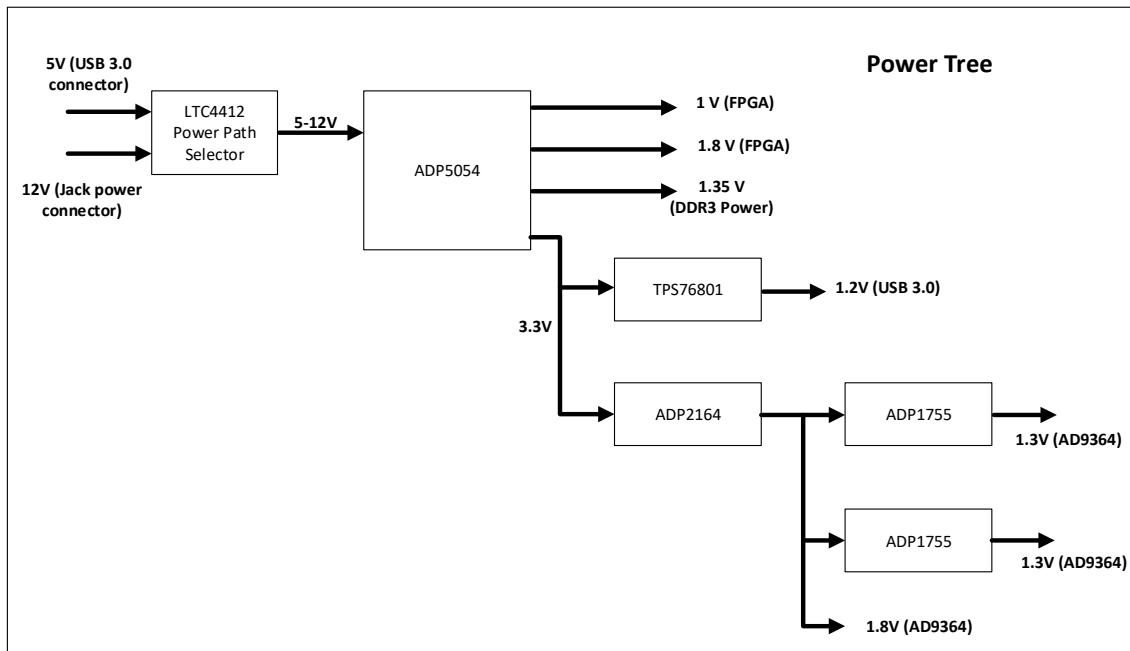


Figura 17: Diagrama distribución de tensiones.

Como se puede apreciar en el diagrama, la SDR puede ser alimentada tanto por USB 3.0 como por medio un jack externo. La selección entre ambas tensiones es realizada automáticamente por el selector de camino LTC4412, el cual prioriza el de mayor tensión. Ante la presencia de ambas conexiones el sistema será alimentado por el Jack.

La potencia total del sistema, con consideraciones extremas, es de 5.4W con lo cual es posible que la potencia entregada por el USB3.0 (5W) no sea suficiente. Este posible problema con la potencia necesaria y la posibilidad de funcionamiento por medio de un puerto USB 2.0, son los principales motivos por los cuales se agregó la opción de alimentación por Jack.

#### 2.2.6 GPIO y Leds

Para aprovechar los pines sin uso en la FPGA, se utilizaron los mismos para armar un GPIO como muestra la siguiente figura:

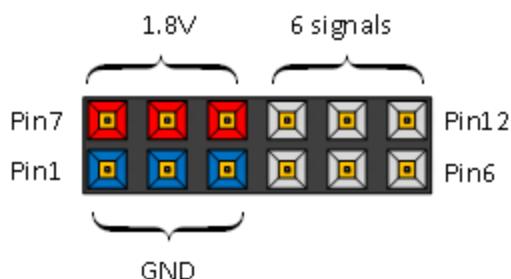


Figura 18: Pines de propósito general

GPIO Pin	FPGA Pin
4	M16
5	K15
6	K13
10	R12
11	N14
12	K12

Tabla 11: Asignación de pines GPIO-FPGA

Además también se agregaron 8 leds a la placa, los cuales van a ser de futura ayuda a la hora del debugging, para poder visualizar cualquier registró en forma de byte.

### 2.3 SDR a nivel sistema

En esta sección se presenta un posible esquema funcional de la SDR diseñada. Haciendo enfoque en el flujo de datos del sistema, HDL, interfaces, software, drivers, etc. Esto servirá como guía para quien ejecute la siguiente etapa de este proyecto. Este esquema se realizó en base a las siguientes referencias [16],[17],[18].

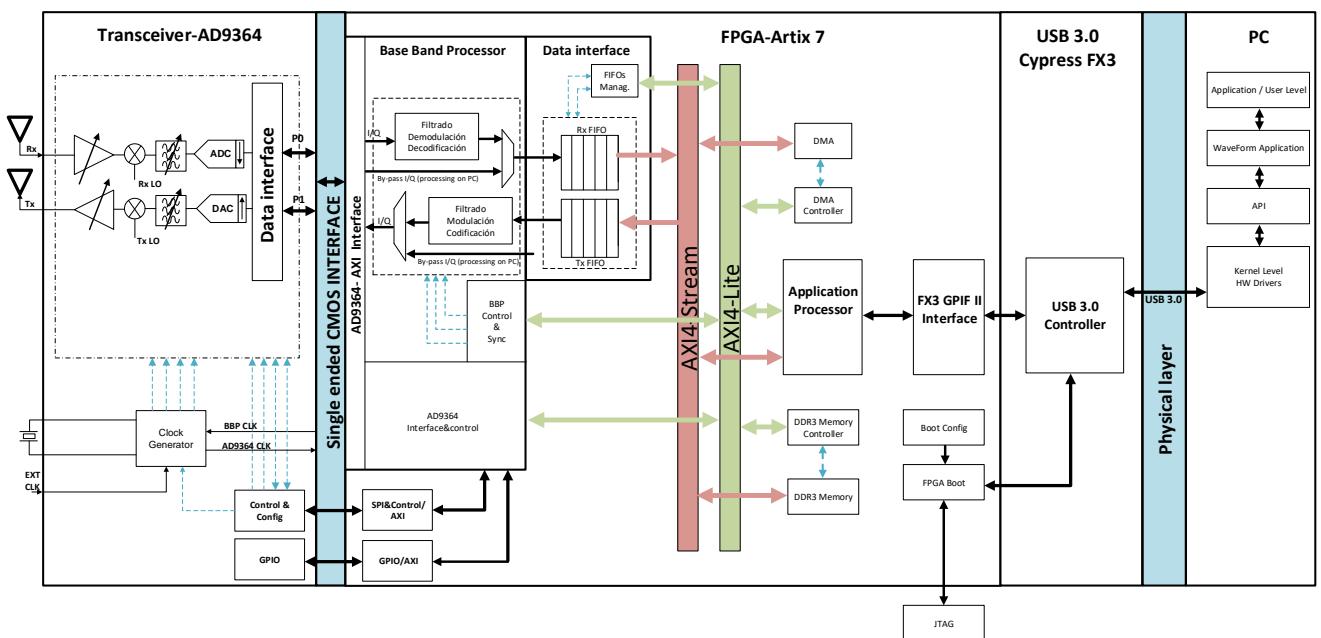


Figura 19: Esquema funcional SDR

### 2.3.1 HDL

A continuación se detallara la función de cada uno de los módulos a implementar en HDL:

#### **Application Processor**

El procesador implementado en la FPGA están destinados a:

- Control y configuración de: hardware y datos de la SDR para implementar la aplicación deseada.
- Encargarse interface del hardware con la aplicación en la computadora.

Para simplificar la interface entre la PC y el hardware de la SDR, el application processor puede correr Linux. Una opción para este app processor puede ser un MicroBlaze application processor (del cual se estimó la cantidad de recursos necesarios de la FPGA en la sección 2.2.4.1) el cual puede correr Linux. Este tipo de procesador tiene como requerimiento mínimo disponer de una memoria RAM de 32 MB o más.

#### **Base Band Processor**

Este procesador es el encargado de realizar los procesos de filtrado, modulación/demodulación, codificación/decodificación, scrambling/descrambling, sincronización/recuperación de portadora y demás procesos necesarios para el procesamiento. En el mismo se selecciona el camino de datos IQ dependiendo si el procesamiento se realiza en la PC (by-pass) o en el BBP. Este procesador actúa como esclavo del application processor (AP), es posible que a la hora de entregar datos al AP, el mismo esté ocupado por lo tanto es necesario la utilización de FIFOs para no perder información.

En este bloque se realizará el control, interface, sincronismo y configuración del AD9364 dependiendo las instrucciones enviadas por el AP. La configuración del transceiver dependerá del modo de funcionamiento (halfduplex Tx o Rx, fullduplex) en el que se encuentre y la información de los datos (BW, IF, gain, etc.).

#### **FPGA AXI**

Advanced eXtensible interface (AXI) es parte de las especificaciones de ARMs AMBA. El AXI es una interconexión punto a punto diseñada para sistemas alta performance y microcontroladores de alta velocidad. Esta interconexión punto a punto evita compartir el bus, permitiendo un ancho de banda alto y baja latencia. En la segunda versión de AXI (AXI4) existen tres tipos de interfaces:

- AXI4: Para requerimientos de alta performance en mapeo de memoria.
- AXI4-Lite: Para comunicaciones de mapeo de memoria de bajo rendimiento. Por ejemplo control y estado de registros.
- AXI4-Stream: Para transmisiones de datos de alta velocidad.

Para más información ver [19]

### 2.3.2 Programación FPGA

La configuración de la FPGA se realiza por medio controlador de USB 3.0 o por medio del JTGA, eliminando la necesidad de una memoria flash para programar la FPGA. La siguiente figura muestra el camino utilizado para la programación la FPGA.

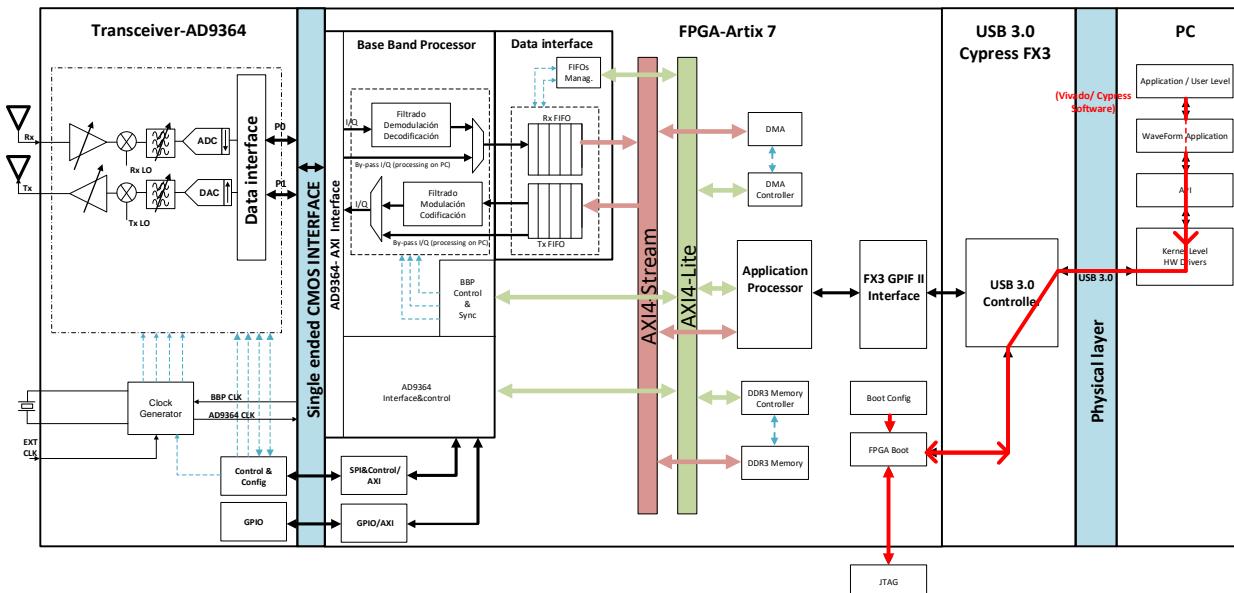


Figura 20: Programación de la FPGA

En el caso del controlador USB 3.0, actuando como master, configura la FPGA en modo Slave Serial.

Para poder realizar esta configuración, es necesario realizar las siguientes conexiones entre el controlador de USB 3.0 y la FPGA:

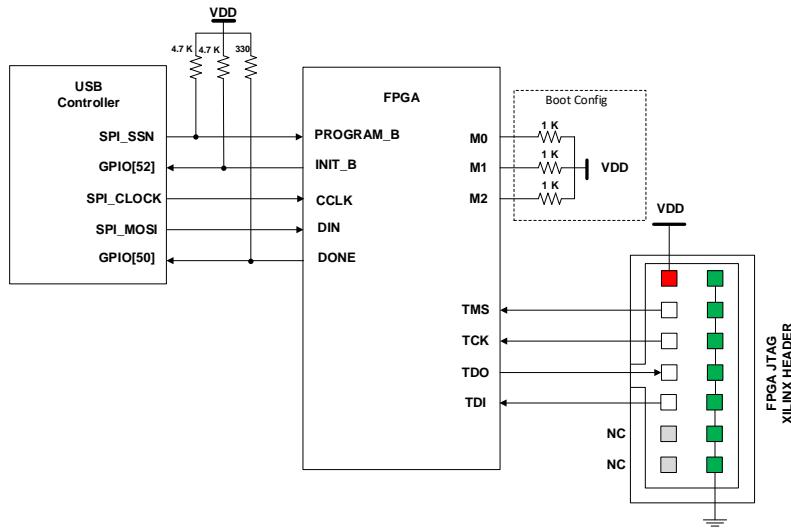


Figura 21: Diagrama en bloques configuración FPGA

En donde:

- **Program\_B**: Cuando esta LOW por 500 ns o más, fuerza a la FPGA a reiniciar el proceso de configuración, limpiando la memoria de configuración.
- **INIT\_B**: Low después de power on reset (POR) o cuando la FPGA está limpiando la memoria de configuración.
- **CCLK**: Clock de configuración.
- **DIN**: Entrada de datos serial para la configuración de la FPGA. Estos datos son capturados en los flancos ascendentes de CCLK
- **DONE**: Low durante la configuración y HIGH cuando la configuración es terminada exitosamente.
- **M [2:0]**: Determinan la forma de configuración que tendrá la FPGA, dependiendo de los pull-down / pull-up que se conecten a estos pinos.

Tablita de los diferentes valores de M[2:0] y algo sobre el JTAG

Configuration Mode	M [2:0]	Bus Width	CCLK Direction
Master Serial	000	x1	Output
Master SPI	001	x1,x2,x4	Output
Master BPI	010	x8,x16	Output
Master SelectMap	100	x8,x16	Output
JTAG	101	x1	Not Applicable
Slave SelectMAP	110	x8,x16,x32	Input
Slave Serial	111	x1	Input

Tabla 12: Selección del modo de booting FPGA

El controlador de USB comienza a configurar la FPGA pulsando PROGRAM\_B (t1) y monitoreando el pin INIT\_B. Cuando el pin INIT\_B está en HIGH (t2) la FPGA está lista para recibir los datos de configuración. El master empieza a suministrar los datos de configuración y la señal de clock hasta (t3):

- Que el pin DONE este en HIGH indicando una configuración exitosa (t4).
- Que el pin INIT\_B este en LOW indicando un error en la configuración.

La siguiente figura muestra un diagrama de tiempo de las señales involucradas en la configuración de la FPGA.

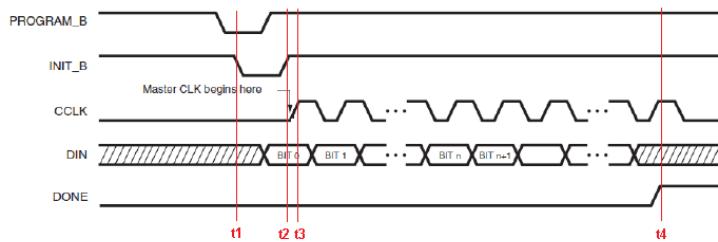


Figura 22: Diagrama de tiempos, configuración FPGA

Para más información sobre la programación de la FPGA vía USB ver [\[\]](#)

### 2.3.3 SDR transmisión y recepción Half Duplex

Una vez programada y configurada la FPGA, la SDR se encontraría lista para correr la aplicación. El modo de operación dual port half dúplex es utilizado en aplicaciones que requieren operaciones TDD (time-division dúplex) y una tasa de datos hasta 61.44 MHz. En este modo ambos puertos son utilizados, el puerto P0 es asignado a los datos I y el puerto P1 es asignado a los datos Q. En este modo los puertos operan de forma bidireccional, la dirección es determinado por el canal que está activo: transmisión o recepción.

La siguiente figura muestra el flujo de datos en una transmisión en modo half-duplex:

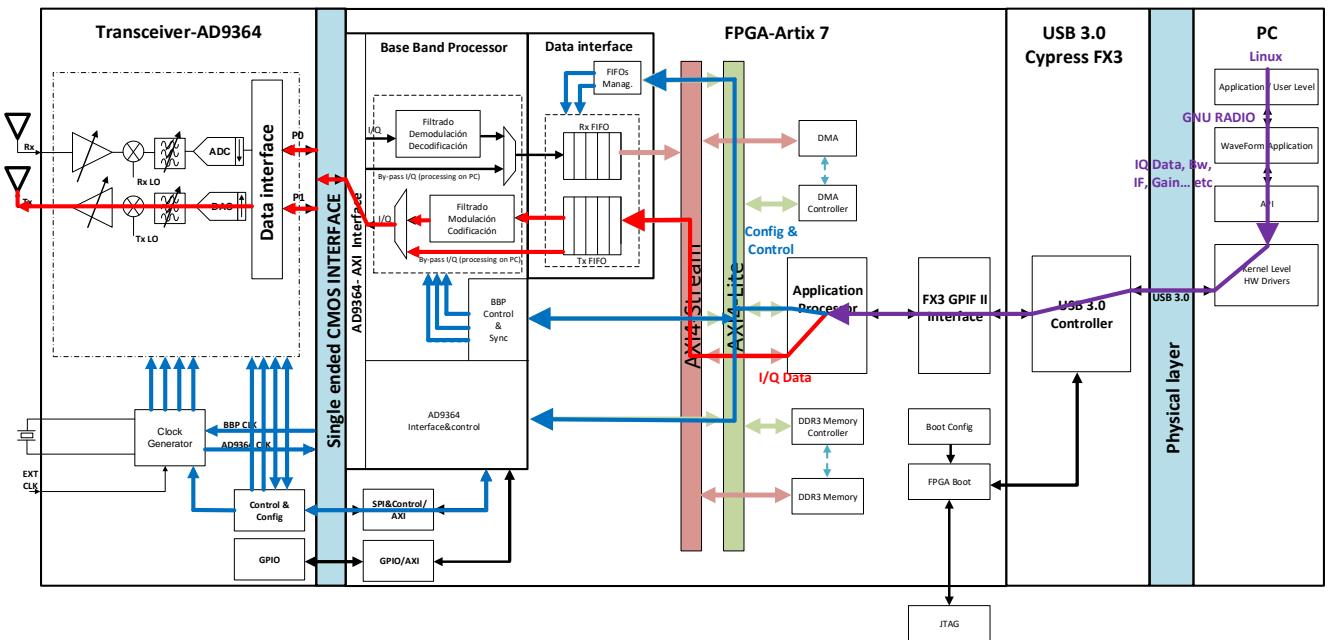


Figura 23: Esquema transmisión de la SDR

Primero se debe realizar la configuración (líneas azules) de los diferentes módulos (AD9364, BBP, FIFOs data interface, etc.) a partir de los datos de la señal (BW, IF, Gain, etc) enviada desde GNU RADIO. Para luego poder transmitir los datos (líneas rojas) por medio del transceiver.

La siguiente figura muestra el flujo de datos en una recepción en modo half duplex:

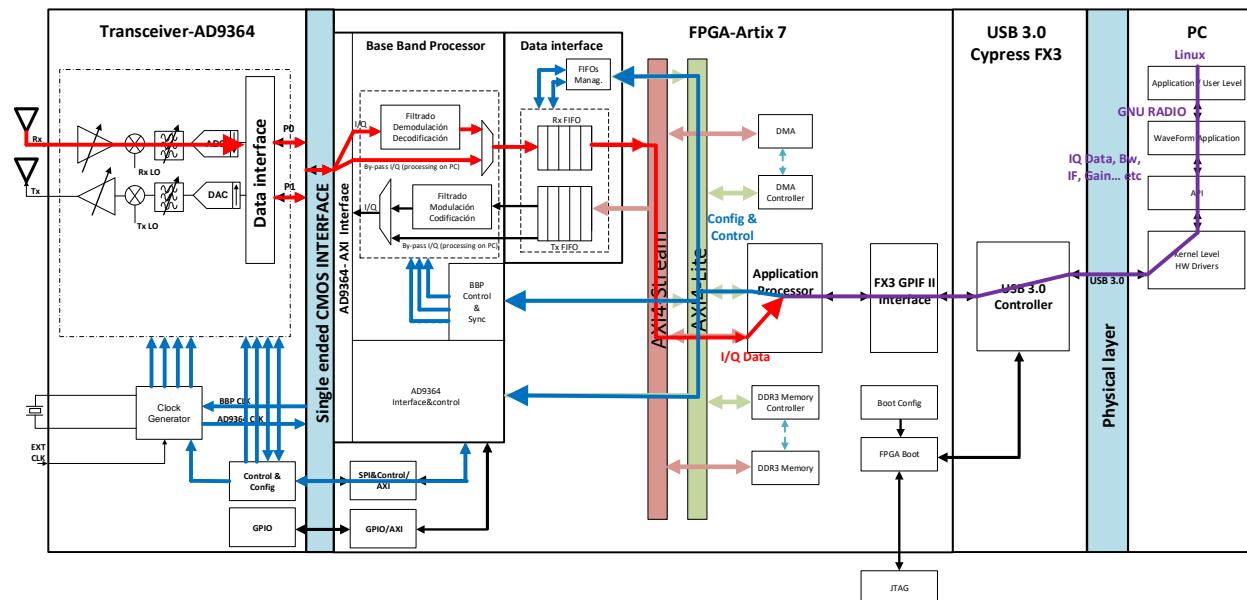


Figura 24: Esquema Recepción de la SDR

Para poder realizar la recepción el application processor, se debe realizar debe realizar la configuración necesaria (líneas azules) de los diferentes módulos (AD9364, BBP, FIFOs data interface, etc.) dependiendo de la señal que se vaya a detectar. La señal recibida (líneas rojas) es amplificada, filtrada, pasada a banda base y digitalizada. Luego los datos IQ que son enviados a la PC, pueden ser procesados en el BBP o en la PC misma.

### 2.3.4 SDR transmisión y recepción Full Duplex

El modo de operación Dual Port Full Duplex es usado en aplicaciones que requieren operar en FDD (frequency-division duplex) y tasas de dato menores a 61.44 MHz. En este modo ambos puertos de datos son utilizados, P0 es asignado a los datos recibidos y P1 es asignado a los datos transmitidos. Cada bus de datos opera simultáneamente, permitiendo una transferencia full dúplex para los datos de transmisión y recepción entre el BBP y el AD9364.

La siguiente figura muestra el flujo de datos en una recepción en modo full duplex:

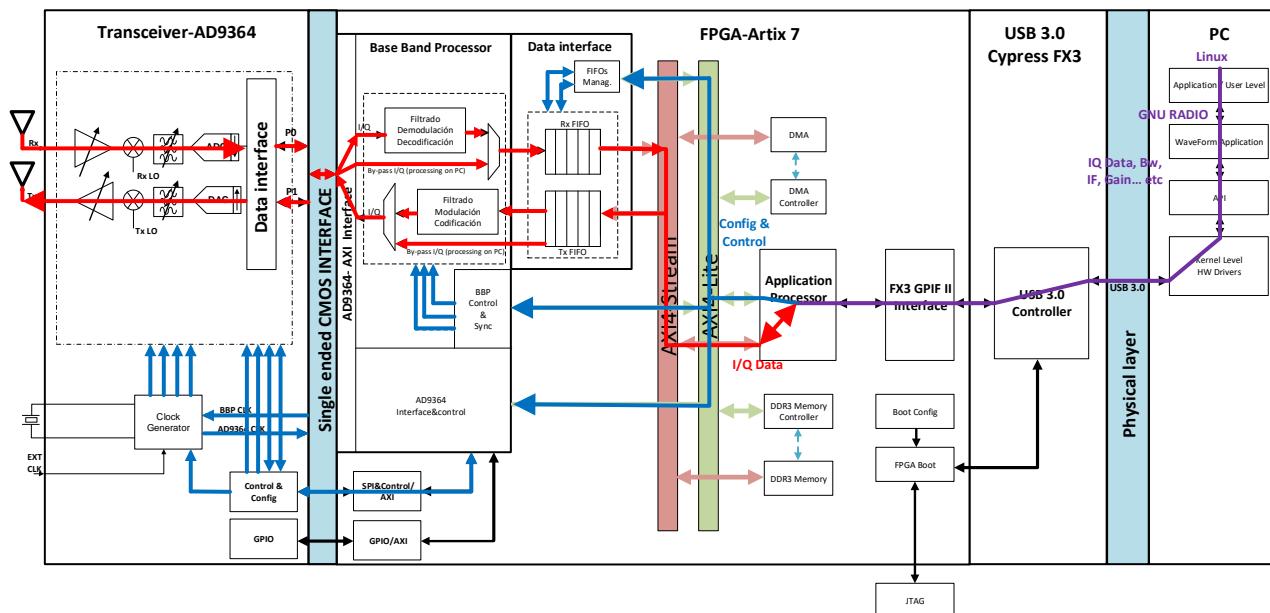


Figura 25: Esquema de operación Full Duplex

### 2.3.5 Software

#### 2.3.5.1 Drivers

Si se embebe Linux en el application processor (por ejemplo utilizando un Micro-Blaze Application Processor) entonces un posible esquema de utilización de drivers puede ser el siguiente:

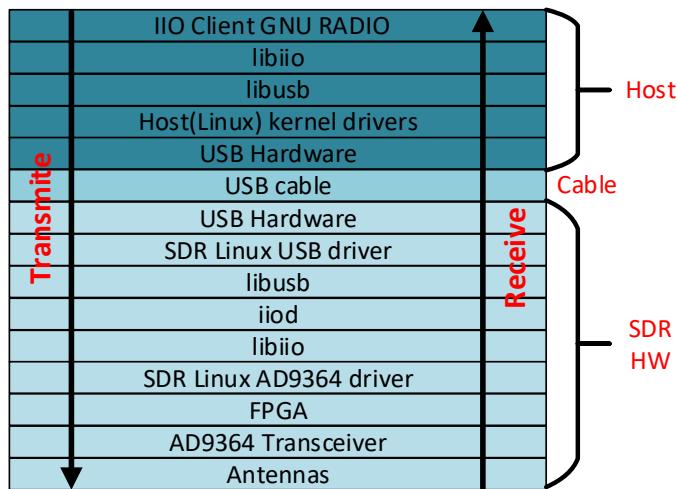


Figura 26: Hardware y drivers del sistema

### Linux Industrial Input/Output (libiio)

El subsistema industrial Input/output (IIO) dentro del Linux kernel está destinado a proveer soporte para dispositivos. El IIO es un open-source estándar adoptado por diferentes fabricantes ya que el mismo provee un API en común para tratar con una gran variedad de diferentes dispositivos. Entre esta variedad de dispositivos podemos encontrar: ADCs, Accelerometers, capacitance to digital converters (CDCs), phase-locked loops (PLLs), etc.

El libiio es utilizado como interface con el subsistema de IIO. Este puede ser utilizado en un dispositivo con Linux embebido o para comunicarse remotamente con el mismo dispositivo desde un host Linux, Windows o Mac a través de USB, Ethernet o Serial.

Libiio fue desarrollado principalmente por Analog Devices. Es una librería open-source Publicada bajo: GNU Lesser General Public Licence. Esta licencia es open source, permitiendo que cualquiera utilice esta librería en cualquier procesador/FPGA/Soc que pueda estar controlando a cualquier dispositivo periférico como por ejemplo: ADC, DAC, hasta incluso un transceiver como el AD9364.

Para más información sobre libiio de Analog Devices y los drivers del AD9364 ver [20][21]

## **Libusb**

Libusb es una librería de c que provee un acceso genérico a dispositivos USB. Su propósito es para ser utilizado por desarrolladores, facilitando la producción de aplicaciones las cuales requieran comunicación vía USB con el hardware.

El firmware de FX3, contiene la librería Cyusb\_Linux\_1.0.4 en la cual se encuentran funciones para realizar una comunicación con un Linux Host. Estas funciones están basadas en los paquetes de la librería libusb. Para más información ver [22][23]

## **Industrial Input/Output liod**

El IIO Daemon es el responsable de permitir la conexión remota con el cliente IIO en este caso el application processor.

### *2.3.5.2 GNU Radio*

GNU Radio es un software gratuito, licenciado bajo GPL, el cual consta con un kit de herramienta para la implementación de radios definidas por software. Inicialmente era utilizado por radio amateurs, pero luego obtuvo un gran interés por los investigadores en el área de wireless, teniendo hoy en día una gran comunidad de usuarios y contribuidores. Este programa fue diseñado para Linux y sus paquetes están pre-compilados para la mayoría de distribuciones de Linux.

GNU Radio proporciona medios para realizar parte del procesamiento digital de un sistema de comunicaciones. Como también algoritmos para definir: filters, channel codes, synchronization elements, equalizers, demodulators, decoders y muchos otros elementos. Estos componentes pueden ser utilizados como bloques para armar un sistema de comunicación, por ejemplo la siguiente figura muestra un receptor de FM en GNU Radio:

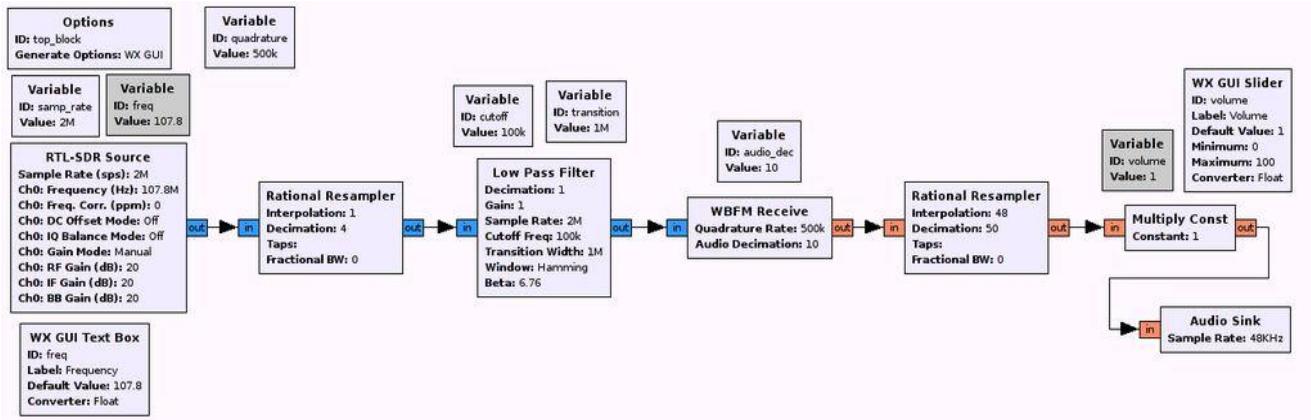


Figura 27: Receptor de FM, GNU Radio

El manejo de datos es realizado por el software y es transparente al usuario. El tipo de dato tanto en la recepción como en la transmisión son muestras complejas en banda base. La aplicación primaria de GNU Radio está escrita en el lenguaje de programación Python, los bloques utilizados en este programa pueden estar escritos en Python o en C++.

### 2.4 SDR Características

Resumiendo la SDR tendrá las siguientes características:

#### Capacidades RF:

- 1xTX & 1xRX Half o Full Duplex
- 70 MHz a 6 GHz de rango de frecuencia
- Hasta 56 MHz de ancho de banda

#### Interface y alimentación:

- USB 3.0
- Alimentación por USB o Jack

#### Hardware:

- FPGA Artix-7 XC7A100T
- DDR3 Memory 256 MB
- GPIO
- JTAG & USB boot

### ¿Qué tan lejos?:

Por lo general tanto en esta SDR como en la mayoría, no es posible determinar que tan lejos puede transmitir. Ya que la distancia depende de muchos factores: carrier frequency, tipo de modulación, ancho de banda, antenas, medio ambiente local (topografía, vegetación, clima), etc.

Por ejemplo la siguiente figura muestra como varia la potencia de salida del transceiver AD9364 en función de la frecuencia y la temperatura.

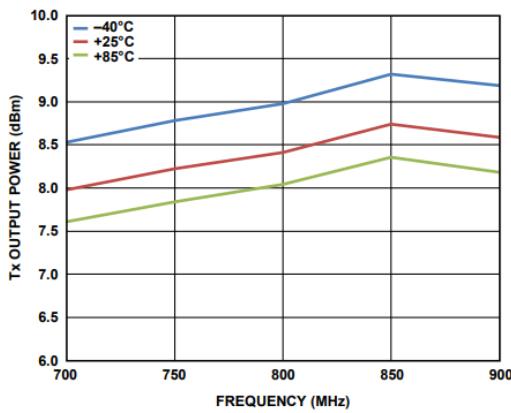


Figura 28: Potencia salida transceiver

### Espectro de frecuencia

La siguiente figura muestra las bandas que se encuentran dentro del espectro de la SDR. Además también se brinda algunas asignaciones de frecuencia dentro del espectro frecuencia argentino, regulado por ENACOM [24].

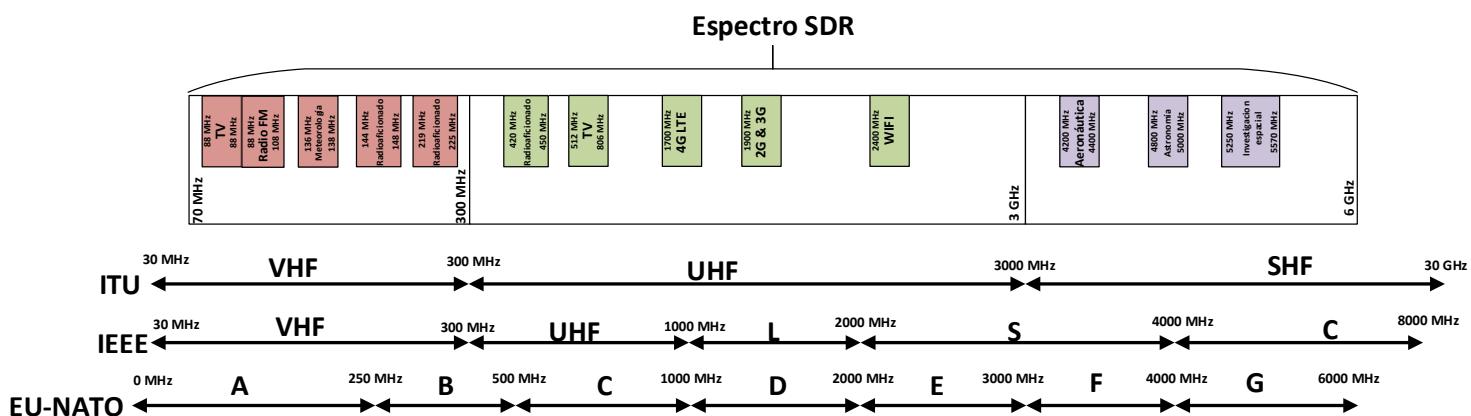


Figura 29 : Bandas y asignaciones de frecuencia

## Capítulo 3

### 3. Diseño con herramienta CAD

#### 3.1 Diseño herramienta CAD

La herramienta imprescindible para llevar a cabo un buen diseño electrónico es un programa informático CAD. Existen muchas alternativas a gusto del diseñador, aunque las mejores opciones son alguna de las desarrolladas por las principales compañías que llevan desarrollando este tipo de software: Cadence, Altium Designer, Eagle o MentorGraphics están entre las más usadas.

En este proyecto se decidió usar Altium Designer debido a previo conocimiento de esta herramienta.

#### 3.1.1 Esquemático

Este programa permite realizar el diseño del esquemático en forma jerárquica, pudiendo dividir el proyecto en módulos funcionales. Cada uno de estos módulos será desarrollado en sheets (con sus correspondientes circuitos), logrando un mejor seguimiento, orden según funcionalidad y facilidad a la hora de diseñar.

Las conexiones entre estos sheets son indicadas en un primer archivo en el cual se muestran todas las conexiones entre las diferentes sheets como se muestra en la figura 30. Estas conexiones se realizan por medio de puertos los cuales pueden ser unidireccionales o bidireccionales.

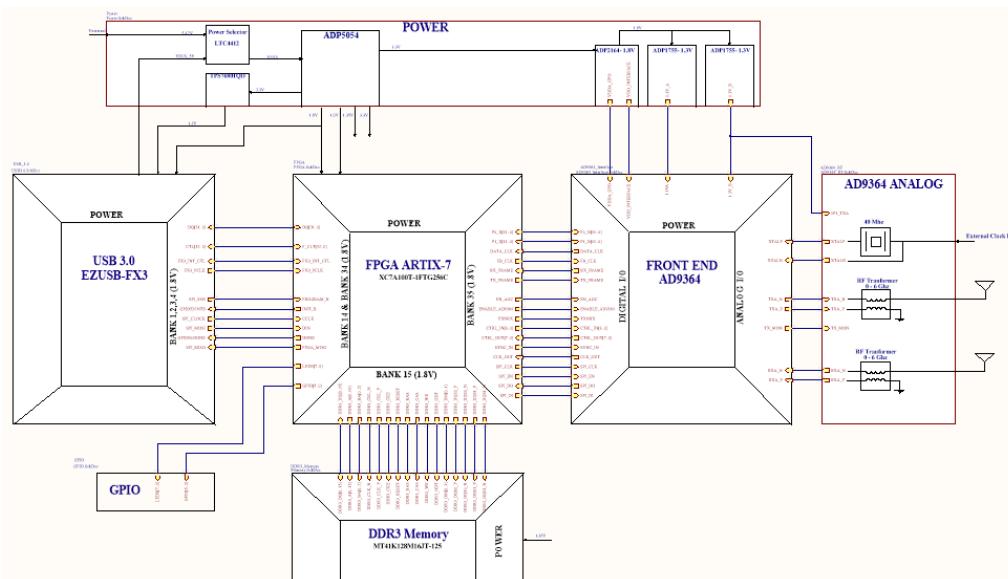


Figura 30: Esquemático en forma jerárquica, primer sheet.

Por otra parte, tanto a los componentes como a las conexiones se les pueden asignar diferentes clases o directivas, las cuales pueden estar sujetas a determinadas reglas de diseño que el programa tendrá en cuenta en la etapa del ruteo del PCB y serán aplicadas automáticamente. Por ejemplo, se pueden seleccionar un conjunto de líneas de datos y asignarles una netclass para luego aplicarle reglas de ruteo a la misma (impedancia, separación entre líneas, largo máximo, .etc.)

### 3.1.2 Componentes

Para una mejor organización del proyecto a cada componente se le agrego información extra en sus propiedades dentro del esquemático. La siguiente figura muestra dichas propiedades:

Parameters			
Visible	Name	Value	Type
<input type="checkbox"/>	Vendedor2	Digi-Key	STRING
<input type="checkbox"/>	Vendedor1	Mouser	STRING
<input type="checkbox"/>	Value	Value	STRING
<input type="checkbox"/>	Tolerance		STRING
<input type="checkbox"/>	Stock2	0 en pedido	STRING
<input type="checkbox"/>	Stock1	1200 en pedido 5 de junio	STRING
<input type="checkbox"/>	Requerimiento		STRING
<input type="checkbox"/>	PartNumber	ADP5054ACPZ-R7	STRING
<input type="checkbox"/>	Package	LFCSP-48	STRING
<input type="checkbox"/>	Manufacturer	Analog Devices	STRING
<input type="checkbox"/>	LinkVendedor2	<a href="https://www.digikey.com/product-detail">https://www.digikey.com/product-detail</a>	STRING
<input type="checkbox"/>	LinkVendedor1	<a href="https://fr.mouser.com/ProductDetail/AI">https://fr.mouser.com/ProductDetail/AI</a>	STRING
<input type="checkbox"/>	Descripción	Reguladores de voltaje, SMD, 48pins	STRING
<input type="checkbox"/>	DataSheet	<a href="http://www.mouser.com/ds/2/609/ADP1">http://www.mouser.com/ds/2/609/ADP1</a>	STRING
<input type="checkbox"/>	CostoVendedor2	10.76	STRING
<input type="checkbox"/>	CostoVendedor1	9.88	STRING

Figura 31: Información componentes del esquemático

Este agregado de información ayuda a la futura construcción de la lista de materiales (BOM), el pedido y compra de los diferentes componentes.

Además estos cada uno de los componentes utilizados contienen otro tipo de información como puede ser: el símbolo, footprint y modelo 3D. La información de todos los componentes utilizados en este proyecto fueron almacenados en una sola librería.

### 3.2 Conexiones

Las conexiones de cada dispositivo fueron realizadas según la indicación de diferentes data sheets y guías de diseño, las cuales se encuentran en el [Anexo E3](#).

Además se tuvo en cuenta los “absolute maximum ratings” de todos los dispositivos, para no superar ninguna tensión máxima en los pines de cada integrado. Por ejemplo en un enabel del ADP5054 se agregó un divisor resistivo en su entrada para no superar su tensión máxima de entrada (+6.5V) cuando es alimentado por medio del Jack.

En las siguientes secciones se comentaran algunas otras particularidades que se tuvieron en cuenta a la hora de realizar las conexiones.

#### 3.2.1 FPGA

##### 3.2.1.1 Bancos I/O

Para realizar las conexiones entre los diferentes bancos del FPGA y los componentes principales (transceiver, controlador USB 3.0 y memoria), se tuvo en cuenta los niveles de tensión e impedancia en cada una de las interfaces. En la siguiente figura se indica la asignación de pines, tensión e impedancia de las diferentes conexiones:

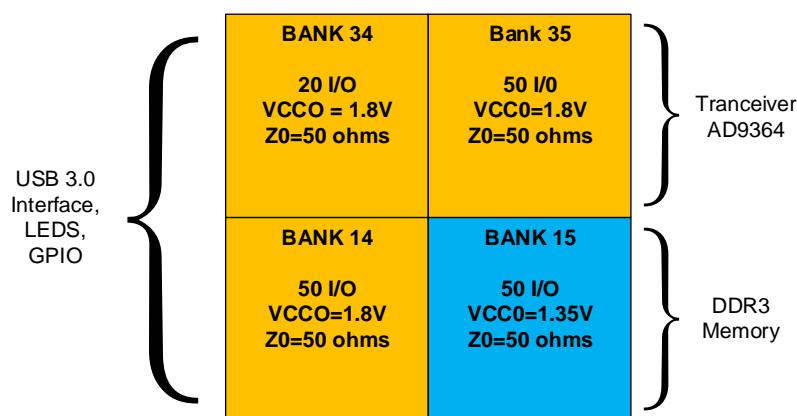


Figura 32: Interface de los bancos de la FPGA

Además cada uno de estos bancos tiene pines que sirven para determinadas funciones, como por ejemplo los pines MRCC. En estos pines se conectarán la mayoría de las señales de clock, ya que por medio de estos se puede distribuir dichas señales a múltiples regiones de clock dentro de la FPGA.

La FPGA también tiene un banco 0, este es un banco específico que contiene los pines necesarios para configuración, programación y reseteo. Además este banco puede ser alimentado con diferentes tensiones, en este caso 1.8V para tener los mismo niveles de tensión que su programador, el controlador de USB 3.0 (el JTAG tendrá la misma tensión también).

#### 3.2.2 Power

Para realizar las conexiones en la sección de alimentación se tuvieron en cuenta las siguientes consideraciones:

- Se agregaron headers para separar por la alimentación por etapas.
- Se realizaron cálculos de para obtener la tensión de salida de algunos reguladores.
- Se realizaron simulación del funcionamiento de la mayoría de los reguladores y el selector de camino de alimentación.
- Utilización de la aplicación “ADP505x BuckDesigner” tanto para las conexiones como para la selección de los componentes del regulador ADP5054.

La mayoría de estos ítems están detallados en el [Anexo B](#) y [Anexo C](#).

#### 3.2.3 Transceiver AD9364

Para realizar la mayoría de las conexiones del transceiver se utilizó: la placa de evaluación del AD9364 “AD-FMCOMMS4-EBZ” (por recomendación del fabricante) y la guía de diseño de hardware que provee el fabricante.

#### 3.2.4 USB 3.0

Para realizar la mayoría de las conexiones del controlador de USB 3.0 se utilizó: la placa de desarrollo “SUPERSPEED EXPLORER KIT” (por recomendación del fabricante) y la guía de diseño de hardware que provee el fabricante.

### [3.3 Capacidades del fabricante PCB](#)

El diseño del PCB se adecuo a las capacidades de fabricación de la compañía “PCB Prototype”. Estas capacidades establecen algunos requerimientos principales para el diseño PCB:

- Mínimo ancho de pista y espaciado

Estos requerimientos estipulan la distancia mínima entre dos pistas y el ancho de las mismas. Dicha distancia mínima es de 3 mil (0.076 mm) para ambos requerimientos.

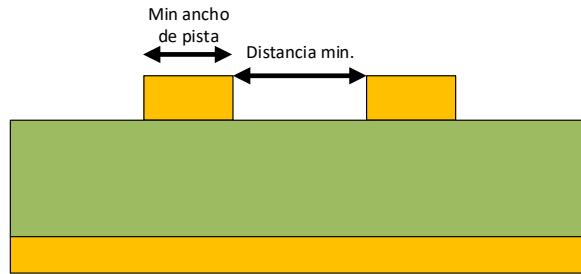


Figura 33: Restricción distancia mínima en PCB

- Mínimo tamaño de agujero y annular ring

El mínimo tamaño de agujero es de 0.15 mm, mientras que el annular ring es de 0.1 mm.

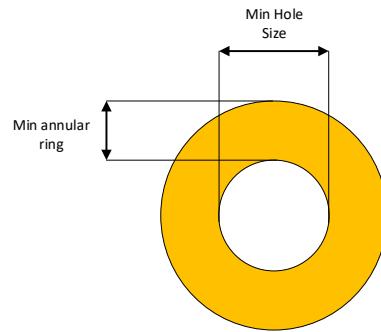


Figura 34: Restricciones tamaño agujero en PCB

Estos requerimientos son esenciales para poder determinar si va ser posible realizar el “fanout” (llevar cada uno de los pinos de un componente BGA hacia fuera del integrado) de los componentes BGA. Ya que a partir de estos requerimientos se podrá determinar si el espacio entre las vías es suficiente para que pasen pistas a través del mismo.

La siguiente figura muestra cómo queda restringido el espacio entre los pad de los componentes BGA para el ruteo de las señales:

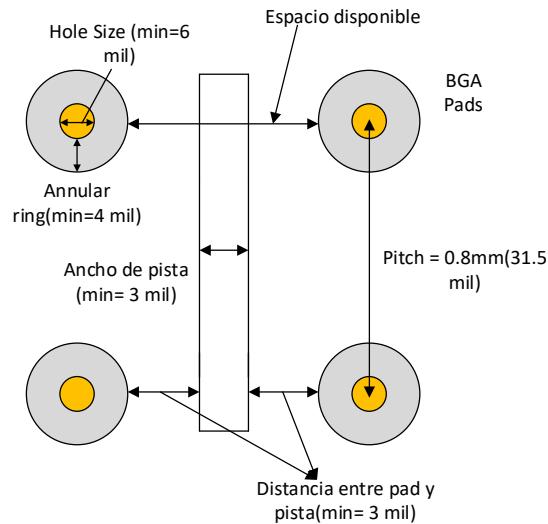


Figura 35: Espacio disponible en componentes BGA

En la siguiente figura se puede apreciar una parte del ruteo de uno de los componentes BGA con pitch 0.8mm (USB 3.0 controller):

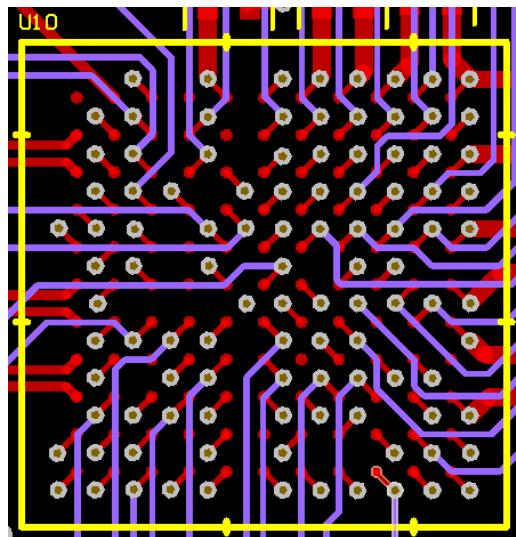


Figura 36: Ejemplo de ruteo, componente BGA

Para más información sobre todos las capacidades del fabricante ir al [Anexo E2](#)

### 3.4 Distribución de componentes y características físicas

Antes de comenzar con el ruteo de la placa, hay que estimar si las dimensiones de la misma son suficiente para dar cabida a todos los componentes y permitir su interconexión. Así como mantener a cierta distancia aquellos que tiendan a generar más calor.

La siguiente figura muestra una de las capas mecánicas en donde se puede distinguir la colocación de los diferentes componentes. La posición ideal para la FPGA es en el centro, ya que las líneas que salen de ella se distribuyen radialmente hacia los componentes que la rodean.

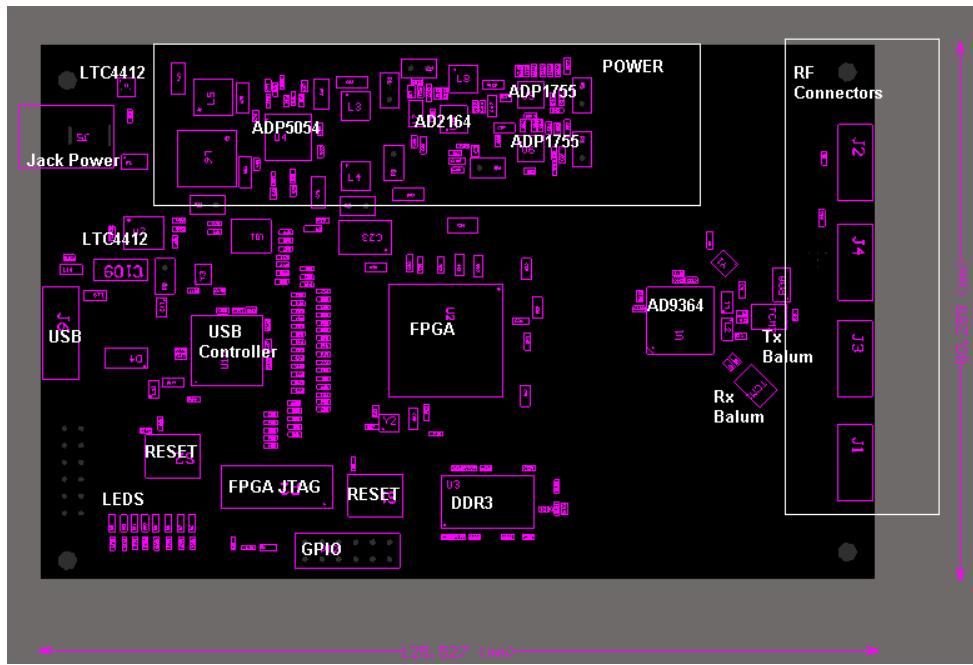


Figura 37: Distribución de los componentes en la placa

En el extremo izquierdo de la placa se encuentra la conexión USB y el conector Jack para alimentación externa. Mientras que en el extremo derecho se encuentran las conexiones de RF para transmisión y recepción.

Las interfaces de usuario se han colocado en lugares accesibles para tener buena operatividad y además lejos de la FPGA ya que las líneas no se consideran críticas.

Las dimensiones de la placa son de aproximadamente 12.5x8 centímetros y tiene un grosor de 1.6mm. Además tiene 4 agujeros de montaje los cuales están conectados a GND.

#### 3.5 Stackup

La estructura de una PCB está constituida por un conjunto de láminas de cobre apiladas y separadas por un material aislante. La manera en la que se distribuyen estas capas, junto con sus respectivos grosores y materiales utilizados se conoce como el stack-up de la placa.

Cuando se va a diseñar una placa con varias capas y con líneas de alta velocidad el primer paso a dar es definir el stack-up o una buena aproximación de él. Para ello se deben tener en cuenta factores como presupuesto, integridad de la señal, impedancias y distribución de la energía.

Para definir el stackup se tuvo en cuenta:

- Por recomendación del transceiver AD9364 la segunda capa y la ante ultima deben ser GND (plano de cobre sin ninguna división) .Esto es crucial para mantener integridad de señal de RF y performance del AD9364.Ade mas tambien estipula que las señales digitales deben ser ruteadas por las capas internas.
- Debido a la cantidad de tensiones diferentes, un solo plano no basta para distribuir todas las tensiones a los módulos pertinentes, por lo que se decidió utilizar 2 planos de tensión.
- La cantidad de capas de señal deben ser las suficientes para poder realizar el ruteo de todos los componentes .En este caso el componente con mayor densidad de señales es la FPGA, por lo tanto esta determinara la cantidad de capas necesarias para el ruteo de todas las señales.  
Para determinar la cantidad de capas necesarias Xilinx provee una fórmula para estimar la misma:

$$\text{Layers} = \frac{\text{Signals}}{\text{Routing Channels} \times \text{Routes per Channel}}$$

En donde:

**Routing Channels:** son el número de caminos de salida disponibles hacia fuera del BGA. Es decir el número de pines de un lado del BGA menos uno, cuatro veces (debido a los cuatro lados del componente). En este caso el encapsulado es de 16x16, obteniendo un total de 60 Routing Channels (15x4).

**Router per Channel:** Puede ser uno o dos dependiendo si uno o dos señales son ruteadas entre los pads del BGA. En este caso solo se realizara el ruteo de una señal entre los pads.

**Señales:** La siguiente tabla indica la cantidad de señales que corresponden a la FPGA:

Sección	Nº Señales
FPGA-USB	53
FPGA-AD9364	48
FPGA-DDR3	46
FPGA-GPIO	16
FPGA Misc.	13
Total	176

Tabla 13: Cantidad de señales por sección

$$Layers = \frac{176}{60 \times 1} = 3 \text{ (redondeado)}$$

Debido a que el número de capas de un PCB deben ser par se decidió utilizar 4 capaz de señales.

- Las impedancias necesarias para cada sección:

Sección	Tipo	Impedancia	Layer	Medidas
AD9364	Diferencial	100 Ω	Top	W=0.229mm,S=0.152mm
			Bot	W=0.229mm,S=0.152mm
	Individual	50 Ω	Top	W=0.236mm
			Bot	W=0.128mm
			Signal 1	W=0.14mm
			Signal2	W=0.236mm
	Diferencial	90 Ω	Top	W=0.203mm, S=0.152mm
	USB 3.0	50 Ω	Top	W=0.236mm
			Bot	W=0.128mm
			Signal 1	W=0.14mm
			Signal 2	W=0.236mm
	FPGA	100 Ω	Top	W=0.229mm,S=0.152mm
			Bot	W=0.229mm,S=0.152mm
		50 Ω	Top	W=0.236mm
			Bot	W=0.128mm
			Signal 1	W=0.14mm
			Signal 2	W=0.236mm
	Diferencial	100 Ω	Top	W=0.229mm,S=0.152mm
	Bot	W=0.229mm,S=0.152mm		
Memory	Individual	50 Ω	Top	W=0.236mm
			Bot	W=0.128mm
			Signal 1	W=0.14mm
			Signal 2	W=0.236mm
	Diferencial	100 Ω	Top	W=0.229mm,S=0.152mm
	Bot	W=0.229mm,S=0.152mm		

Tabla 14: Impedancias de cada sección

Teniendo en cuenta toda las cuestiones mencionadas anterior mente se diseñó el siguiente stackup:

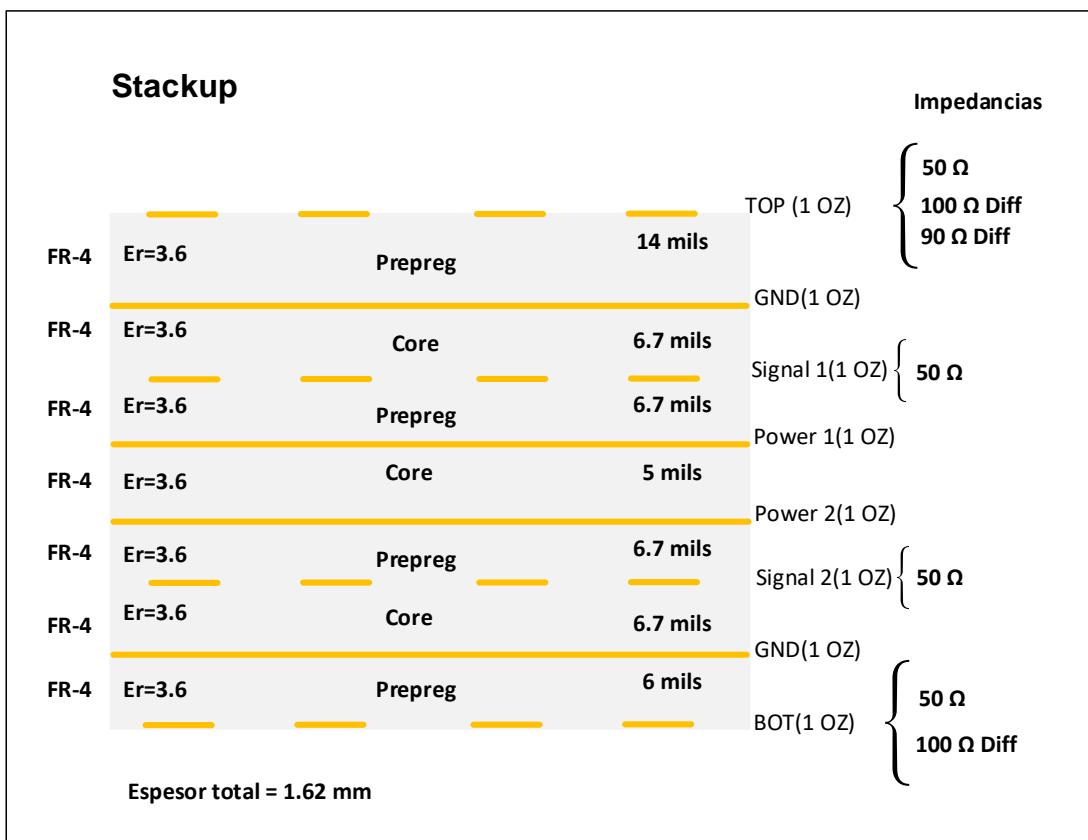


Figura 38: Stack up del PCB

Para obtener la constante dieléctrica del material se contactó vía mail con el fabricante del PCB “PCB Prototypes”, los cuales suministraron la información necesaria para realizar el stackup.

### 3.6 Líneas de transmisión

Cuando las señales son del orden de las decenas de los MHz estas pueden ser tratadas como señales de alta velocidad.

Una señal puede ser considerada de alta velocidad cuando la relación entre su frecuencia o longitud de onda y la distancia que esta señal recorre son de magnitudes similares. Cuando en una pista de cobre o cable por donde viaje una señal si cumple con dicha relación se consideran como una línea de transmisión.

### 3.7 Impedancia característica, reflexión y terminación

Una de las propiedades de una línea de transmisión es su impedancia característica,  $Z_0$ . Cuando se diseña una línea de transmisión entre un emisor y un receptor, las impedancias: de salida del receptor, de la línea y de entrada del receptor deben estar adaptadas. Si no se ajusta bien esta adaptación de impedancias se producen reflexiones y oscilaciones dentro de la misma línea provocando una degradación de la señal o incluso el fallo completo de la comunicación.

La impedancia característica es una propiedad de la línea de transmisión que depende únicamente de su geometría (alto, ancho y forma) y de las propiedades del material dieléctrico que la componen (aire, FR4, etc.) siendo independiente de la longitud de la línea. Por tanto es importante que la línea de transmisión preserve su forma y geometría durante todo el recorrido de la señal para mantener una continuidad en su impedancia característica.

En un PCB los parámetros de fabricación como el grosor del laminado de cobre, el tamaño del material dieléctrico (distancia al próximo plano sólido) y constante dieléctrica, son los parámetros que van a permitir calcular la geometría de la línea para conseguir la impedancia característica deseada.

A continuación se muestran los dos casos que podemos encontrar en el diseño de esta SDR.

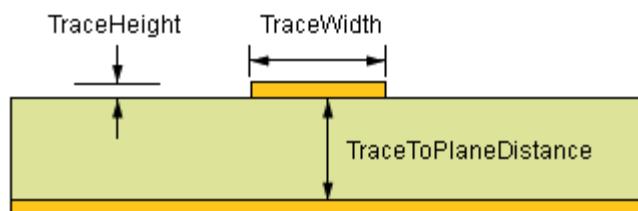


Figura 39: Microstrip

Fórmula utilizada por Altium Designer para el cálculo de impedancia en Microstrip:

$$Z_0 = \frac{87}{\sqrt{\epsilon_r + 1.41}} \times \ln\left(\frac{5.98 \times H}{0.8 \times W + T}\right)$$

En donde:

$H$  = Distancia entre pista y plano

$W$  = Ancho de pista

T= Grosor de pista

Er= Constante dieléctrica del material dieléctrico.

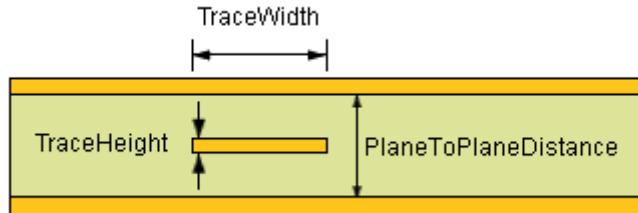


Figura 40: Strip line

Fórmula utilizada por Altium Designer para el cálculo de impedancia en Stripline:

$$Z_0 = \frac{60}{\sqrt{e_r}} \times \ln\left(\frac{1.9 \times B}{0.8 \times W + T}\right)$$

En donde:

B = Distancia entre planos

### 3.8 Líneas diferenciales

Las señales que se transmiten de modo diferencial viajan a través de dos líneas acopladas, presentando la misma magnitud en ambas y signo opuesto.

Una de las principales ventajas de las señales diferenciales es su superior inmunidad al ruido frente a las señales individuales. Cuando una interferencia electromagnética afecta a la señal diferencial, afecta a las dos líneas por igual y, puesto que en el receptor se mide la diferencia entre ambas, se suprime el efecto del ruido y se puede recuperar fácilmente la señal original

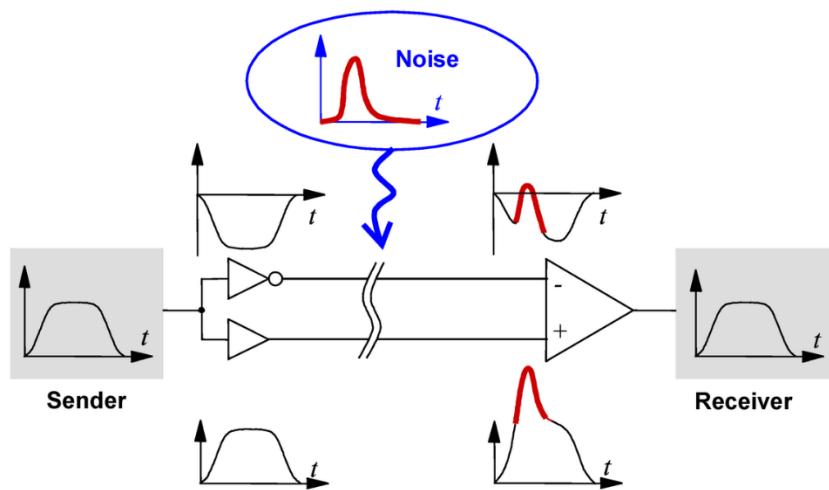


Figura 41: Propiedad de las líneas diferenciales

En la SDR la mayoría de las señales de alta velocidad se transmiten en modo diferencial. Al igual que las líneas individuales, las líneas diferenciales también presentan una impedancia característica,  $Z_0$ . En este caso también se calcula en función de la geometría del par diferencial y de las propiedades del stackup, interviniendo además la separación entre las líneas como muestra la siguiente figura:

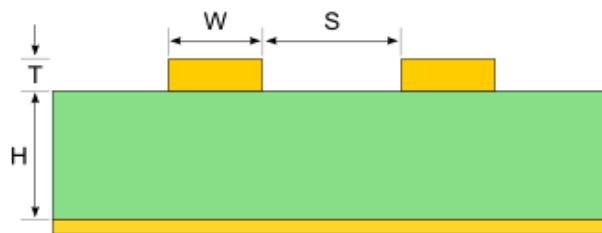


Figura 42: Líneas diferenciales en PCB

La herramienta Altium Designer no hace cálculos de impedancias diferenciales, por lo que se utilizó un programa realizar dicho cálculo. Los cálculos realizados con este programa auxiliar se encuentran en el anexo [C4]

### 3.9 Ajuste de longitudes

Cuando varias señales de alta velocidad pertenecen a un mismo grupo o bus de datos que funcionan de forma síncrona, es importante que todas tengan la misma distancia eléctrica, es decir, que todas las señales eléctricas lleguen dentro de

un pequeño intervalo de tiempo que garantice el correcto funcionamiento circuito (lo ideal sería que todas lleguen al mismo tiempo).

A menudo, estas señales se encuentran, en línea recta, a diferentes distancias cada una de su destino. Por ello es preciso forzar a que las líneas más cortas realicen un recorrido extra para igualar al recorrido de la línea más larga del grupo.

Este criterio se tuvo en cuenta en varias secciones del diseño de la SDR:

- Interface GPIF II entre el controlador de USB3.0 y la FPGA
- Líneas de dato, address y control en la memoria DD3
- Líneas de datos y clocks entre la FPGA y el transceiver AD9364

La siguiente figura muestra como mediante la utilización de serpentinas o ruteo en forma de laberinto se equiparan los largos de las líneas de interface entre la memoria DDR3 y la FPGA.

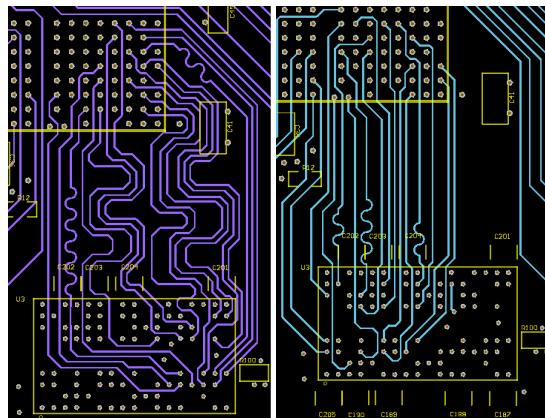


Figura 43: Ejemplo ruteo memoria DDR3

Nombre	Largo (mil)	Nombre	Largo (mil)
DDR3_DQ0	894,669	DDR3_A1	1310,066
DDR3_DQ15	898,977	DDR3_A0	1312,581
DDR3_DQ8	899,062	DDR3_A6	1318,087
DDR3_DQ6	901,442	DDR3_A7	1320,843
DDR3_DQ5	901,852	DDR3_A10	1321,725
DDR3_DQ9	902,812	DDR3_A9	1321,832
DDR3_DQ1	903,108	DDR3_A2	1323,624
DDR3_DQ10	903,831	DDR3_A11	1323,735
DDR3_DQ13	904,941	DDR3_A12	1324,045
DDR3_DQ2	907,448	DDR3_A5	1333,306
DDR3_DQ4	908,487	DDR3_A4	1343,608
DDR3_DQ11	909,391	DDR3_A8	1344,844
DDR3_DQ3	913,043	DDR3_A13	1353,309
DDR3_DQ14	916,676	DDR3_A3	1361,219
DDR3_DQ12	919,22		
DDR3_DQ7	939,541		

Tabla 15: Longitud de líneas en la interfaz DDR3

#### 3.10 Consideraciones tomadas al Rutear la placa

El ruteo de cada módulo se hizo de acuerdo a las guías de diseño provistas por cada fabricante. Algunas de las recomendaciones, criterios y herramientas utilizadas se encuentran en los [Anexo A](#), [Anexo B](#) y [Anexo C](#).

##### 3.10.1 FPGA

- Todas las pistas tienen impedancia de  $50 \Omega$ .
- El plano de power de mayor necesidad se encuentra más cerca de la FPGA (1V FPGA)
- Cristal lo más cercano a la FPGA posible

##### 3.10.2 Power

- Se calculó de tamaño de vías para determinar si pueden suministrar la corriente necesaria.
- Uso de vías de escape térmico

##### 3.10.3 USB 3.0

- Ajuste de largo de pista utilizando la herramienta xSignals.
- Utilización de protección ESD
- Calculo impedancia Diferenciales

##### 3.10.4 AD9364

- Criterio de cero diferencia de longitud entre las líneas de interface digital con la FPGA.
- Interface entre el transceiver y las antenas.
- Coplanar Waveguide y ajuste de impedancia  $50 \Omega$  para lograr un ancho similar al conector de RF y no tener desadaptación de impedancia.
- Pista en el top para posible shielding.

### 3.11 Planos de alimentación y tierra

Como se ha explicado en la sección 3.5, de las 8 capas con que cuenta el PCB, 2 son planos de tierra y 2 son para planos de alimentación. Las siguientes figuras mostraran como están distribuidas las diferentes tensiones y la solidez de los planos de GND.

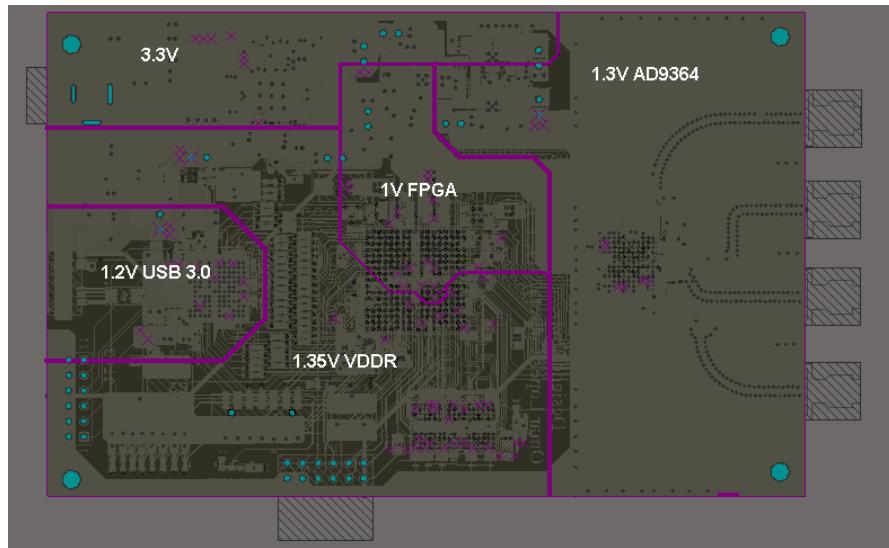


Figura 44: Plano1 de power

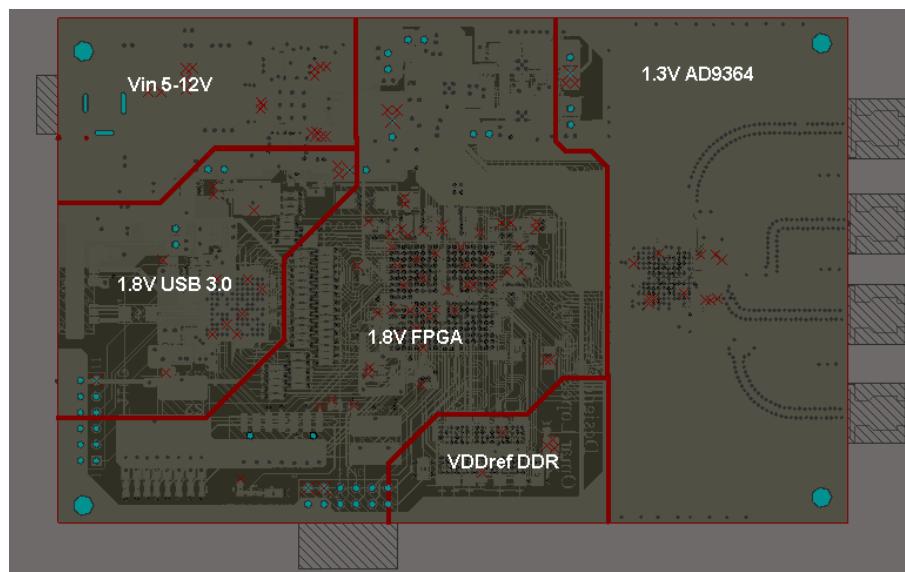


Figura 45: Plano2 de power

Una de las recomendaciones principales para los planos de tierra es que sean superficies de cobre ininterrumpidas que cubran toda la placa. En la siguiente figura se muestra dicho plano sólido.

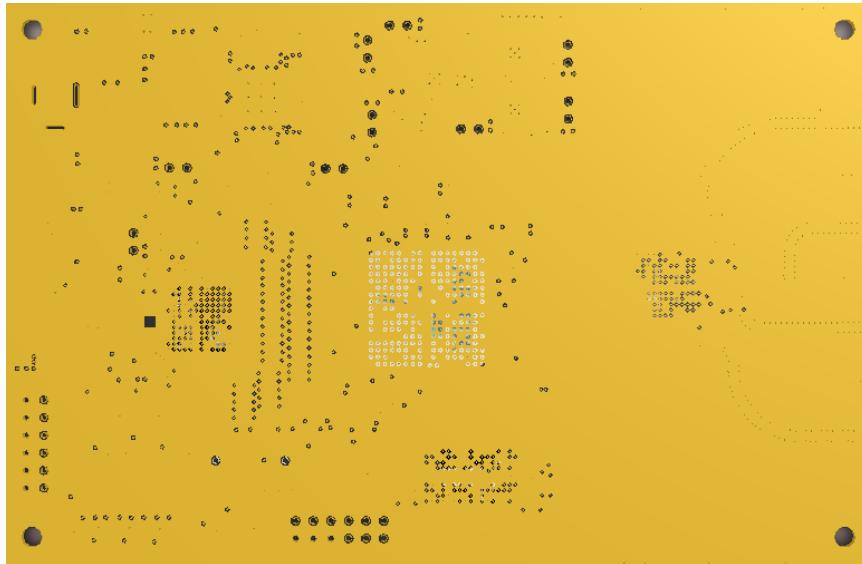


Figura 46: Plano sólido GND

### 3.12 Capacitores desacople

Los capacitores de desacople funcionan como un fuente de poder local muy chica para los circuitos integrados. Si el suministro de poder cae temporalmente en voltaje, estos capacitores pueden entregar voltaje durante un breve periodo de tiempo. Es por esto que estos capacitores también son llamados capacitores de bypass, ya que pueden temporalmente actuar como fuente de poder haciendo un bypass la fuente suministradora.

Todos los componentes cuentan con sus condensadores de desacoplo. En particular tanto en el controlador del USB 3.0 y el transceiver AD9364 los capacitores de desacople fueron seleccionados y colocado como en sus placas de desarrollo (por recomendación del fabricante).

Por otra parte, en el caso de la FPGA, Xilinx directamente recomienda la cantidad de condensadores de desacoplo que se deben proveer, incluso sugiere los fabricantes y modelos, ya que cumplen con requisitos de calidad en cuanto a ESR, ESL y degradación de la capacidad con el tiempo y la temperatura.

Lo más importante es que cada uno rango de valores de capacitancia son útiles en unos márgenes de frecuencia, influyendo a la distancia a la que se deben colocar de la FPGA. Por ejemplo los capacitores grandes actúan cuando el regulador de voltaje deja de funcionar en el rango de frecuencia baja.

Los capacitores grandes pueden ser ubicados en cualquier parte en los alrededores de la FPGA en cambio los pequeños deben ser colocados lo más cerca posible de los pines de alimentación. Esto es complicado para los integrados que tienen encapsulado BGA pues hay una gran cantidad de vías justo debajo del chip y cuesta mucho ubicar todos los condensadores en sus lugares óptimos.

En la siguiente figura se muestra la parte posterior de la FPGA en donde se puede observar que los capacitores más pequeños ocupan el centro mientras que los más grandes los alrededores.

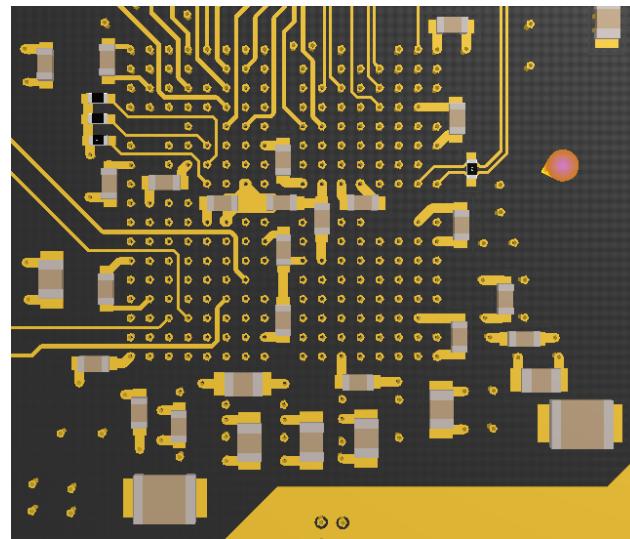


Figura 47: Ubicación capacitores desacople FPGA

## 3.13 Modelo 3d

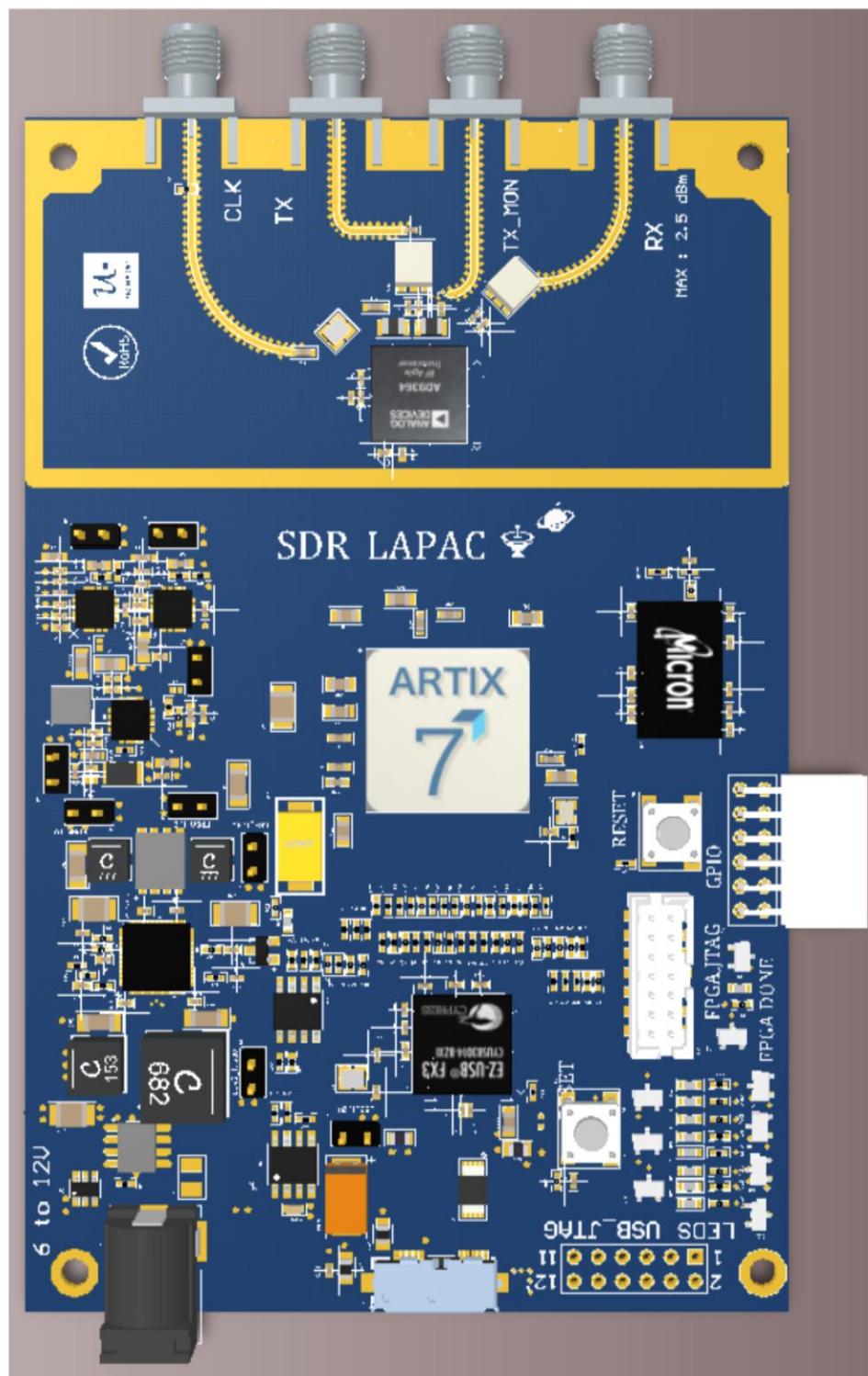


Figura 48: Modelo 3D placa, vista superior

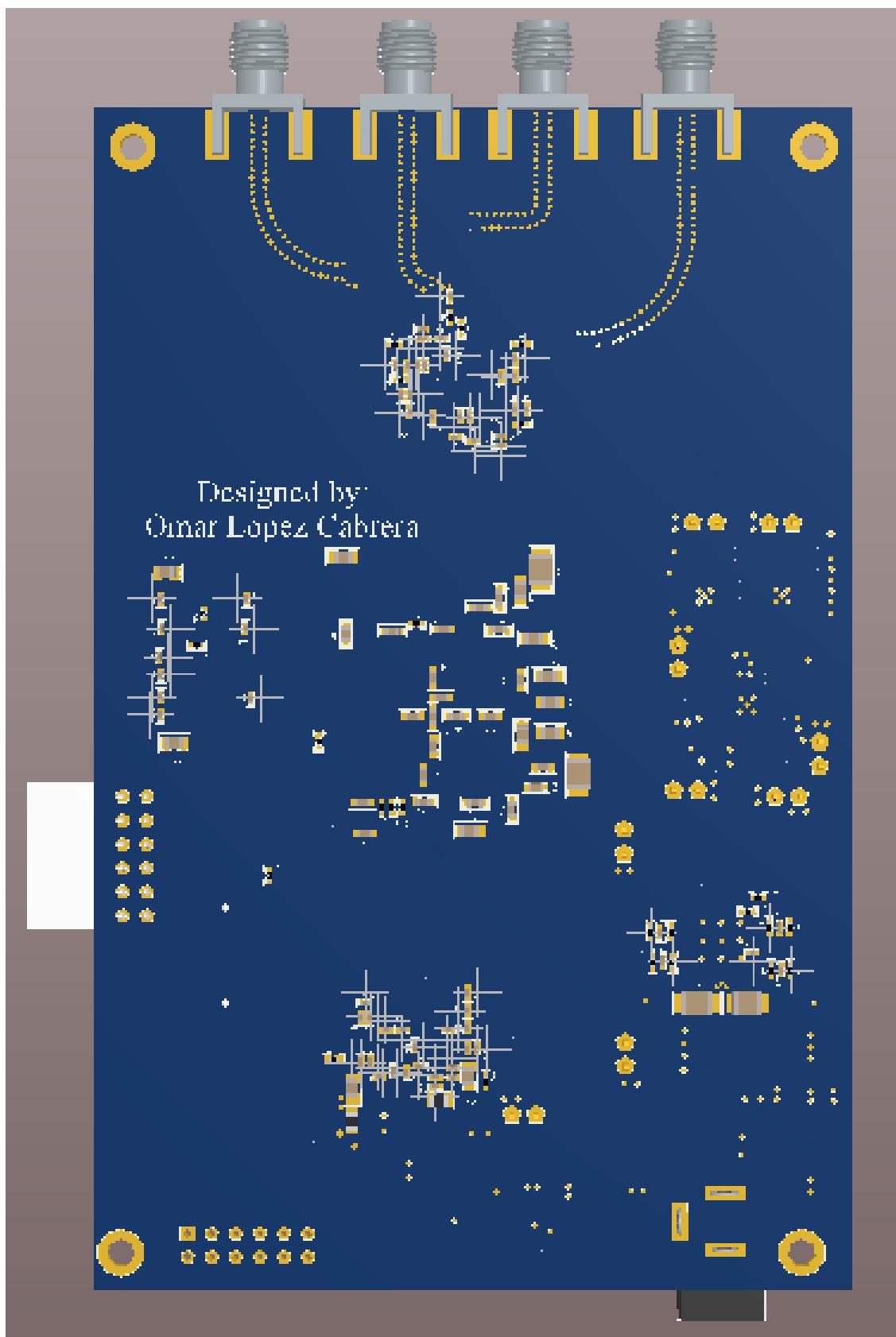


Figura 49: Modelo 3D placa, vista inferior

## Capítulo 4

### 4. Sumario

#### 4.1 Simulación de integridad de la señal

Por otra parte, también sería muy interesante realizar simulaciones de integridad de la señal. Este tipo de simulaciones no son de carácter funcional sino de carácter físico. Actualmente los fabricantes proporcionan modelos de simulación de sus circuitos integrados en unos archivos llamados IBIS. Estos archivos contienen información de niveles de tensión y corriente en los pines de entrada y salida del chip, obtenidas de mediciones o de simulaciones del circuito.

Para realizar este tipo de simulaciones se pueden utilizar los siguientes programas:

- Mentor Graphics
- HyperLynx
- Cadence Design System

Dichos simuladores con estos archivos son capaz de simular cómo aparece un impulso eléctrico en un pin de salida, qué alteraciones sufre durante su recorrido por la placa y cómo es recibido en el pin del receptor.

#### 4.2 Fabricación y ensamblado del PCB

Una vez que terminado las simulación y los requerimientos por parte del fabricante son cumplidos es momento de enviar el diseño a fabricación. Al tratarse de un primer prototipo por lo general se realizan una tirada pequeña de placas, lo cual aumenta el costo por placa.

Durante el desarrollo del proyecto se contactó con el fabricante “PCB Prototype” para pedir los datos necesarios para el diseño (datos sobre el stack up, costos y requerimientos de fabricación), por lo cual es recomendable realizar la fabricación con dicho proveedor.

La segunda operación a realizar es la compra de componentes. Si bien todos los componentes discretos y la mayoría de circuitos integrados están en permanente

disponibilidad, algunos componentes más exclusivos como la FPGA o el transceiver pueden estar sujetos a fabricación bajo pedido o sin stock por lo tanto puede llevar a una demora en la recepción de varias semanas. Al tratarse de una gran cantidad de componentes se debió utilizar tres proveedores (Digikey, Mouser y Mini-Circuits) para cubrir la totalidad de los mismos.

La última operación es el ensamblado de la placa. El cual consiste en montar y soldar todos los componentes de la placa. El proceso más común consiste en utilizar un esténcil (plantilla de metal con perforaciones, provisto por PCB Prototype) como mascara sobre el PCB para depositar la pasta de soldado. Luego los componentes se sueltan en su sitio de la placa donde quedan sujetos por esta pasta aún semilíquida. Finalmente, el PCB se introduce en un horno especial con curvas de temperaturas controladas, produciéndose la soldadura dentro del mismo.

Debido a la gran cantidad de componentes y el uso de circuitos integrados en encapsulados BGA es recomendable que se contrate un servicio de ensamblado.

### 4.3 Software

Se debe realizar la programación y configuración de los diferentes módulos de la SDR para poder correr la aplicación deseada. Como así también la utilización de los diferentes drivers para la correcta interface entre la SDR y la PC. Se puede utilizar como ayuda la información detallada en la sección 2.3.

### 4.4 Posibles Mejoras

- **Transceiver 2x2:** Para transformar esta SDR a una Multi Input Multi Output (MIMO), solo hay que cambiar el transceiver AD9364 por el AD9361. El AD9361 tiene el mismo encapsulado, pin out e interface digital que el AD9364 por lo que solo habría que modificar la parte de RF. Los sistemas MIMO son una parte importante de los estándares de comunicación Wireless modernos como por ejemplo: IEEE 802.11(Wi-Fi), 4G, 3GPP Long Term Evolution, WiMAX and HSPA+.
- **Posibilidad de boot dual de la FPGA:** Se puede agregar el hardware necesario para poder seleccionar el modo de boot de la FPGA (memoria Flash o vía USB) y selección del camino de datos adecuado. Realizando esta mejora la SDR puede llegar a operar de forma autónoma sin la necesidad de una PC para ciertas aplicaciones.

- Balun y red de adaptación según rango de frecuencia:** La impedancia de los puertos de RF del transceiver varían según la frecuencia. Por este motivo es difícil obtener un ajuste óptimo de impedancia y performance para un amplio rango de frecuencia (70Mhz to 6Ghz). En particular el transceiver utilizado en este proyecto tienen 3 pares de puertos diferenciales para Rx (Rxa, Rxb, Rxc) y 2 pares para Tx (Txa, Txb). Se puede aprovechar esta cantidad de puertos para dividir el rango de frecuencia en varios más acotados. Utilizando un mejor balun y red de adaptación para cada rango de frecuencia [25]. Luego la salida de cada balun iría conectada a un selector de camino de RF, el cual a su vez se conectaría a la antena.

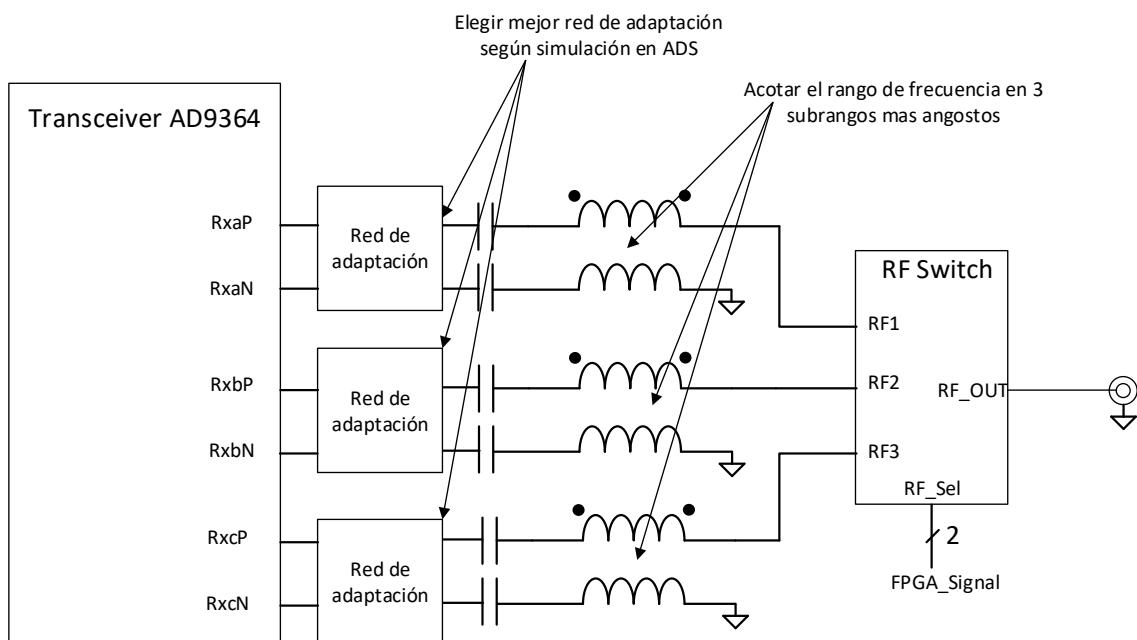


Figura 50: Diagrama de conexión, mejora etapa RF

## Anexos

### A-Hardware Design Guidelines

En esta sección se detallaran tablas que indican las recomendaciones en cuanto al layout del PCB. Como así también la implementación en el Altium de algunas ellas.

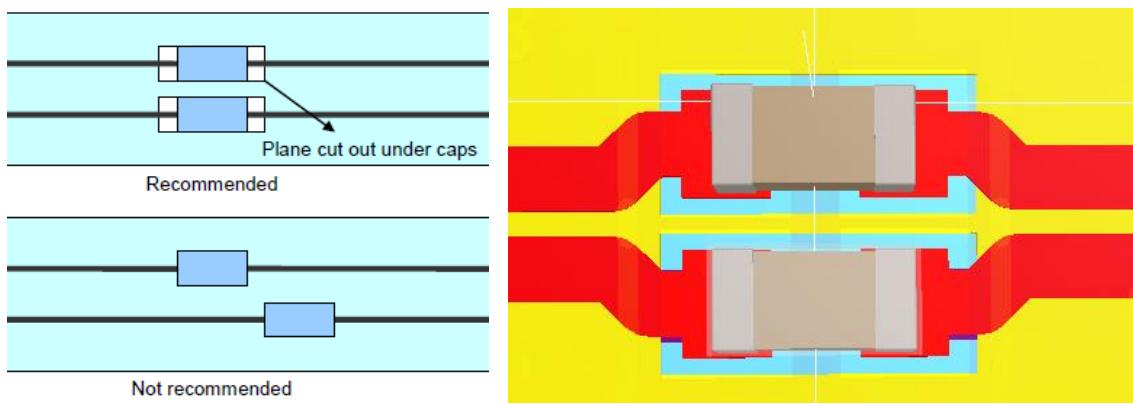
Estas recomendaciones con respecto al layout y otras con respecto al esquemático están disponibles en [Anexo E1](#) y [Anexo E3](#).

#### A.1 USB 3.0

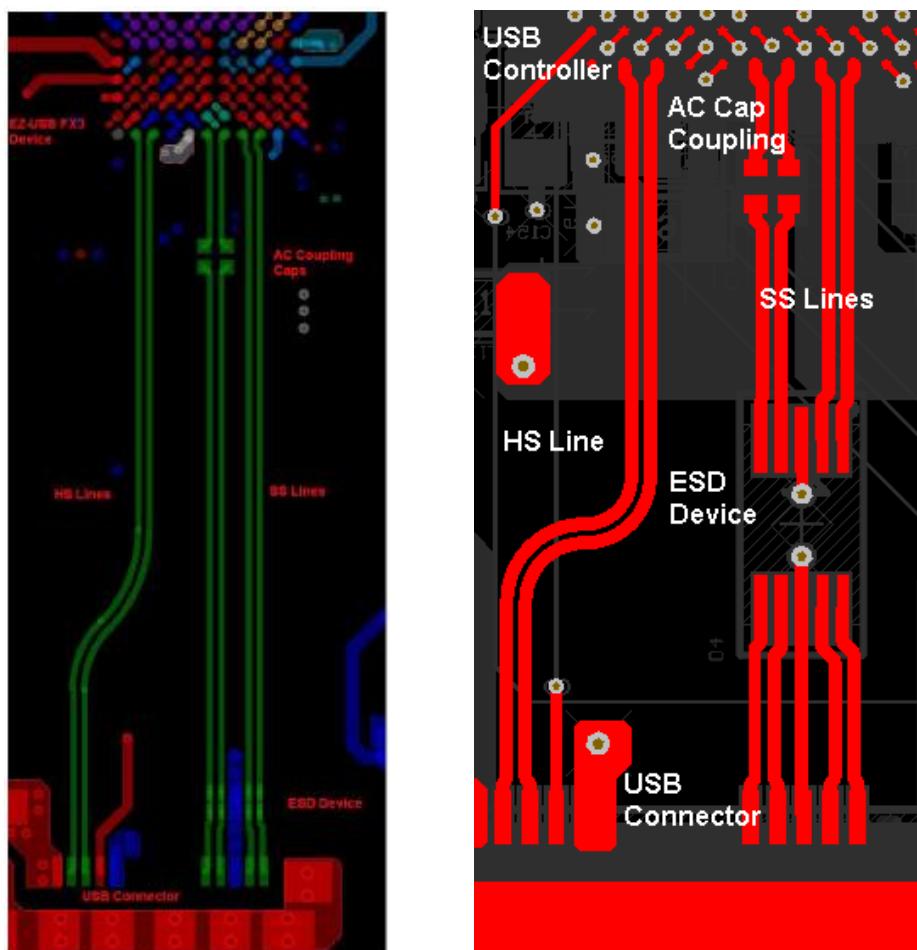
Nº	Layout Design guide
1	Poner el cristal cerca del chip (menos de 2 cm)
2	Capacitores de desacople cercano al los pines de power del FX3
3	Rutear las pistas de las señales de clock lejos de las señales de alta velocidad
4	Rutear las pistas de power lejos de las señales de alta velocidad y las señales de clock
5	Capacitor del circuito de reset cerca del pin de reset
6	Las señales USB SS y USB HS deben tener impedancia de 90 ohms
7	Las señales USB SS y USB HS deben cumplir con el match length
8	Proveer un plano sólido de GND de bajo de las señales de USB data
9	No utilizar Stubs en ninguna pista de USB
10	Proveer las pistas de USB SS con capacitores de desacople de 0.1 uF
11	No utilizar vias en las pistas de USB SS
12	No utilizar codos a 90 grados en las pistas de USB
13	Planos vacíos debajo de los capacitores de acople AC
14	Asegurarse que ruteo de las pistas de SS y HS tengan un espaciado consistente
15	Ruteo de las líneas de USB lo mas cortas posible (<3 inches)

A continuación se mostrara la implementación de algunos de estos puntos (todos fueron implementados) en el PCB diseñado (recomendación/diseño):

- Por lo menos el plano inmediato debajo de los capacitores de AC coupling debe tener un corte con la forma de estos capacitores para evitar capacidades extra a causa de las capacidades de los pads.



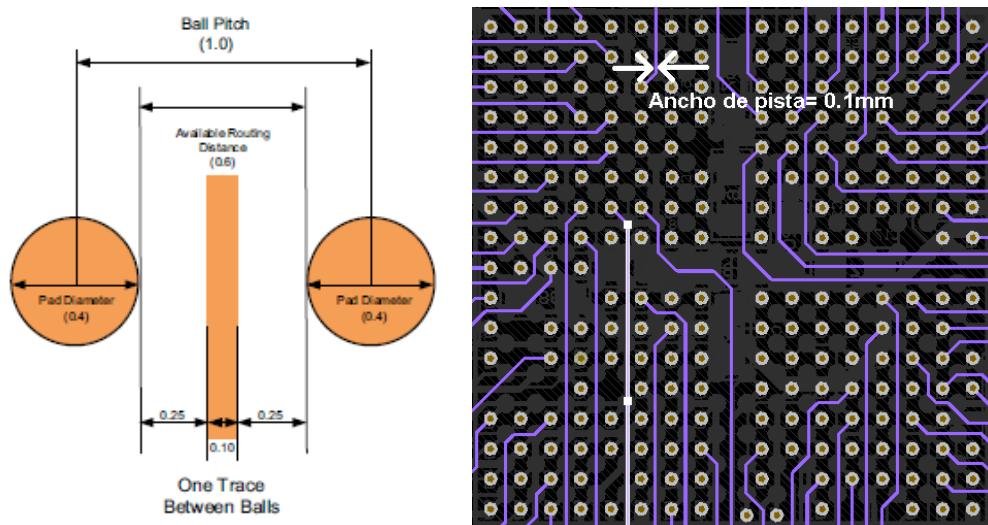
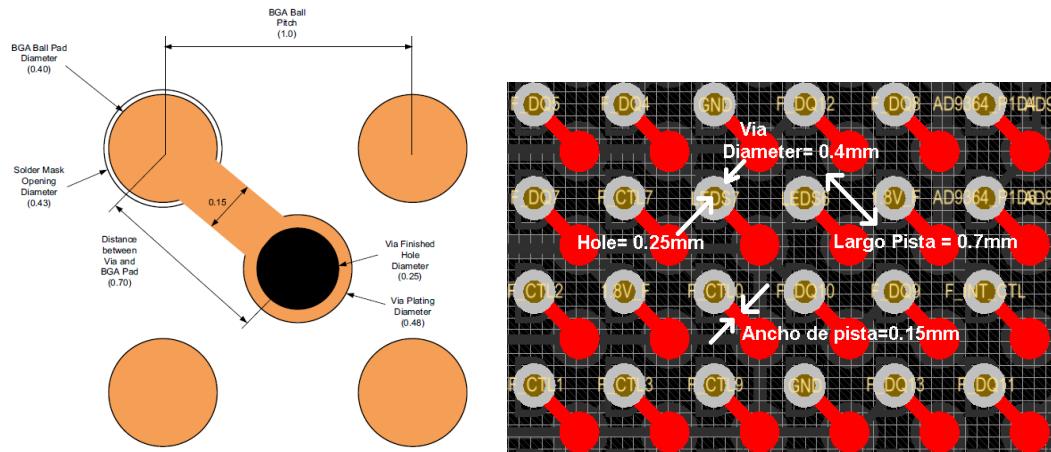
- La siguiente figura muestra un ejemplo e implementación (CAD) del ruteo de las Señales de USB desde el controlador hasta su conector. Las señales diferenciales mantienen una forma uniforme en todo el ruteo de las líneas. Los capacitores de AC coupling se encuentran cerca del controlador.



## A.2 FPGA

Nº	Layout Design guide
1	Realizar el layout del area del BGA de acuerdo a las recomendaciones
2	Realizar ruteo dentro del area del BGA de acuerdo a las recomendaciones
3	Estimacion de cantidad de capas PCB de acuerdo a formula de Xilinx
4	Colocacion de Capacitores cerca de la FPGA dependiendo tamaño
5	Utilizar land pattern recomendados para los capacitores
6	Utilizar los capacitores recomendados por Xilinx dependiendo modelo de la FPGA
7	Alimentaciones de baja prioridad ubicarlas en la mitad inferior del stack up
8	Alimentaciones de alta prioridad ubicarlas en la mitad superior del stack up

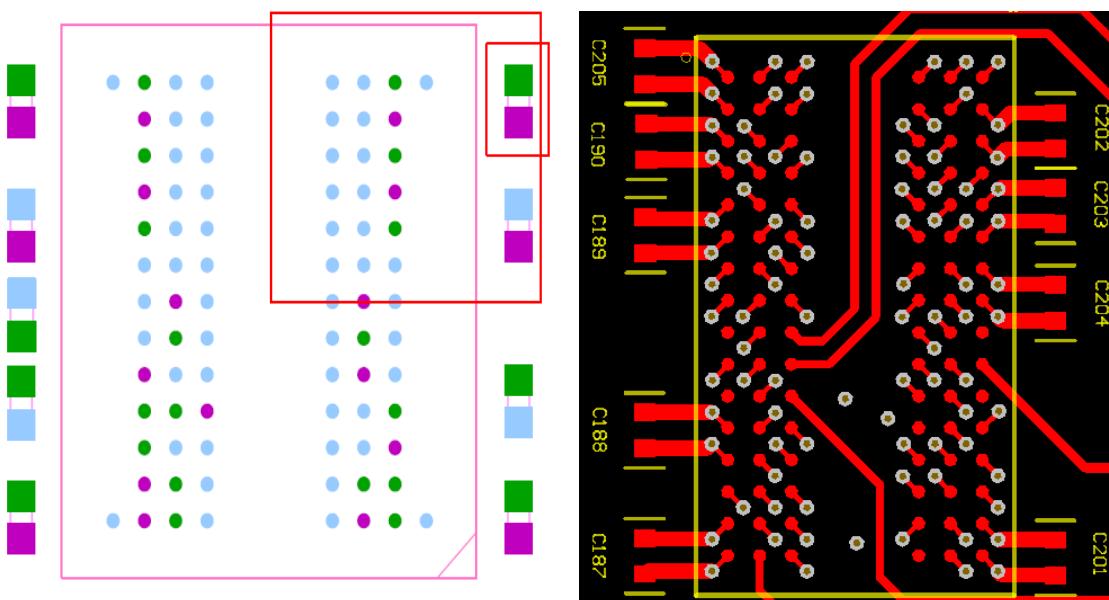
A continuación se mostrara la implementación de algunos de estos puntos (todos fueron implementados) en el PCB diseñado (recomendación/diseño):



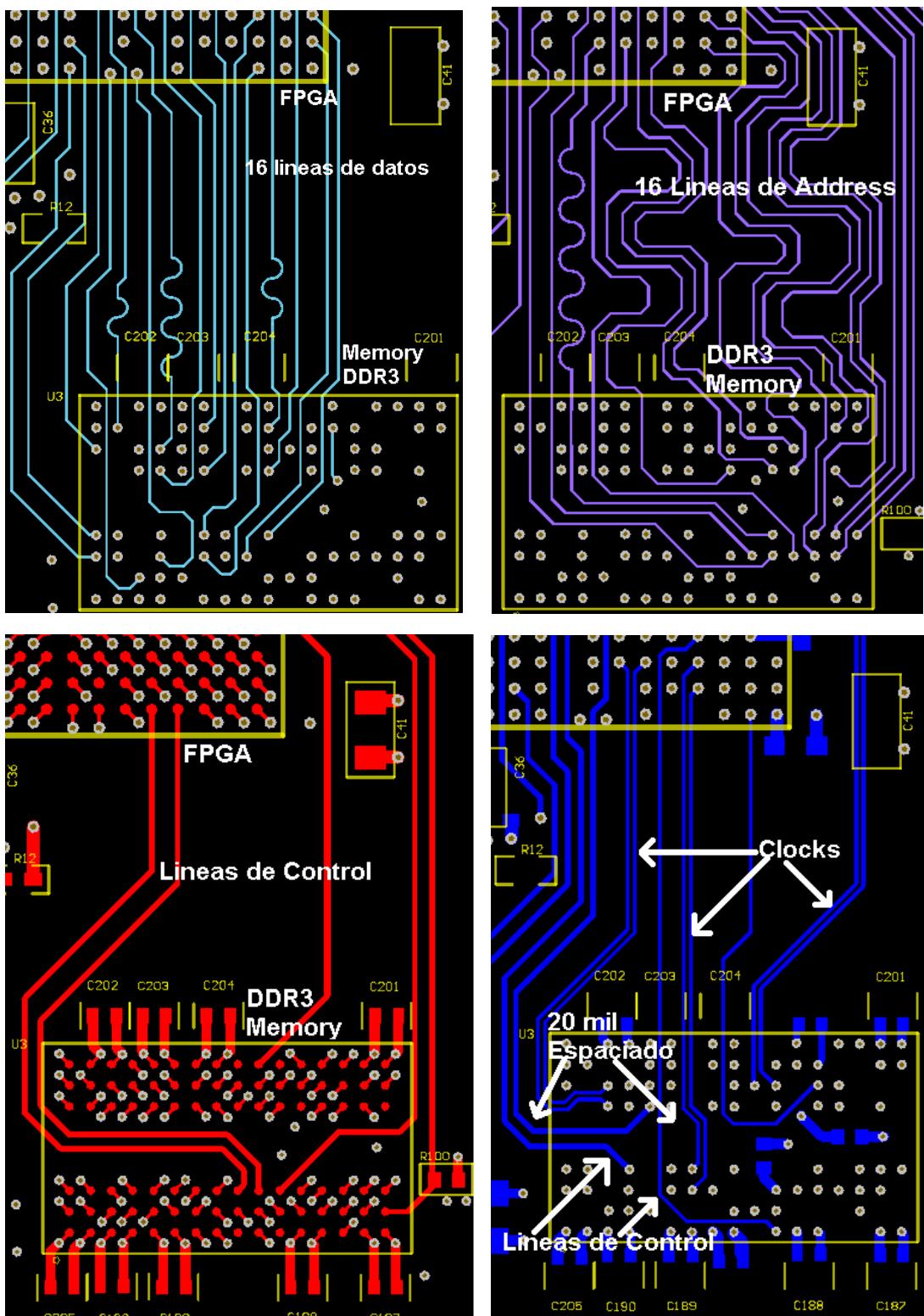
## A.3 Memory DDR3

Nº	Layout Design guide
1	Colocar capacitores de desacople según recomendaciones.
2	Rutear las líneas de datos, Address y control en diferentes capas, de no ser posible dejar un espacio mínimo entre grupos de 20 mils.
3	La máxima diferencia de largo entre las líneas de datos es de +/- 50 mils.
4	Utilizar ruteo en forma de "S" como patrón para incrementar el largo de pista.
5	Las líneas de clock CK y CK# deben ser ruteadas en la misma capa.
6	La máxima diferencia de largo entre las líneas CK y CK# es de +/- 20 mils.
7	La máxima diferencia entre las líneas de CK, Ck# y las líneas de datos es de +/- 500 mils.
8	El largo total de cada línea, desde la memoria hasta el controlador de la misma debe ser menor a 2 pulgadas.
9	Respetar el espaciado entre líneas de un mismo grupo y de diferentes grupos.
10	Las líneas de DDR3 single-ended tienen una impedancia de 50 Ohm.
11	Las líneas DDR3 Differential tienen una impedancia diferencial de 100 Ohm.

A continuación se mostrara la implementación de algunos de estos puntos (todos fueron implementados) en el PCB diseñado (recomendación/diseño):

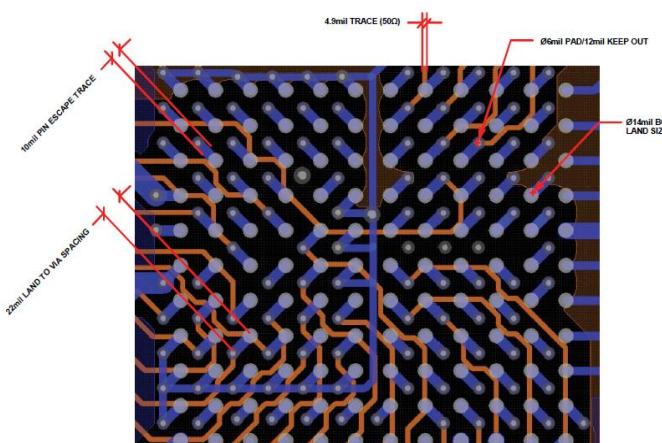


Ajuste de largo de pista utilizando ruteo en forma de "S", ruteo en diferentes capas



#### A.4 Transceiver AD9364

Nº	Layout Design guide
1	Realizar el layout del area del BGA de acuerdo a las recomendaciones.
2	La segunda layer y la ante ultima deben ser planos de GND solidos sin divisiones, esto es fundamental para asegurar la integridad de RF y la performance del integrado.
3	Las señales de RF de Rx y Tx deben ser ruteadas en el top sin la utilizacion de vias.
4	El ruteo de las pistas de Rx y Tx hasta el balun deben ser lo mas cortas posibles.
5	El ruteo de la interface digital debe realizarce en las capas interntas del PCB.
6	La impedancia de las lineas de la interface digital deben ser de 50 Ohm.
7	Deben utilizarse microstrip lines para el ruteo de las señales de RF.
8	Los planos de GND deben ser ruteados hasta los bordes del PCB debajo del conector SMA para mantener la integridad de señal.
9	Mantener alejadas las señales de RF de las señales digitales de alta velocidad.
10	Colocar capacitores de bypass cerca de sus respectivos pines

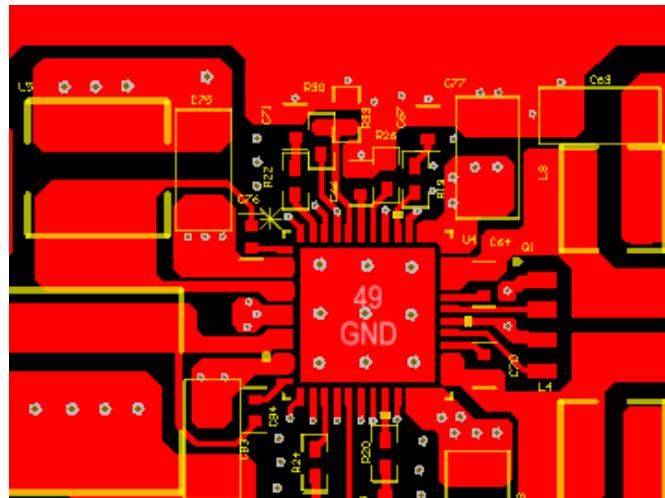


#### B-Cuestiones generales del diseño

##### B.1-Alimentación

##### Vías para escape térmico

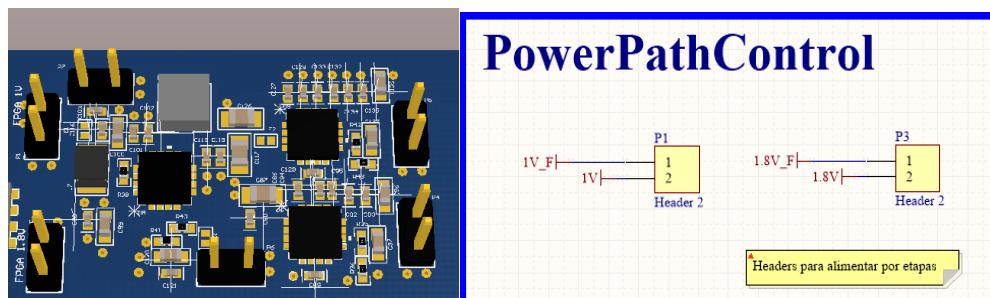
Las vías térmicas son agujeros mecánicos que atraviesan toda la placa. Estás vías son idealmente colocadas directamente bajo los componentes de montaje superficial que disipan mucho calor. Este método hace que el calor que fluye a través de las vías sea disipado en los planos solidos de GND.



#### Header test por etapa

La alimentación de cada sección está separada por un header, de esta manera se puede ir probando cada sección por separado. Una vez probada el correcto funcionamiento de todas las etapas, estos headers pueden ser reemplazados por resistencias de 0 Ohm.

En la siguiente figura se puede ver dichos headers tanto en el modelo 3D como en el esquemático.



#### B.2-USB 3.0

#### Protección ESD

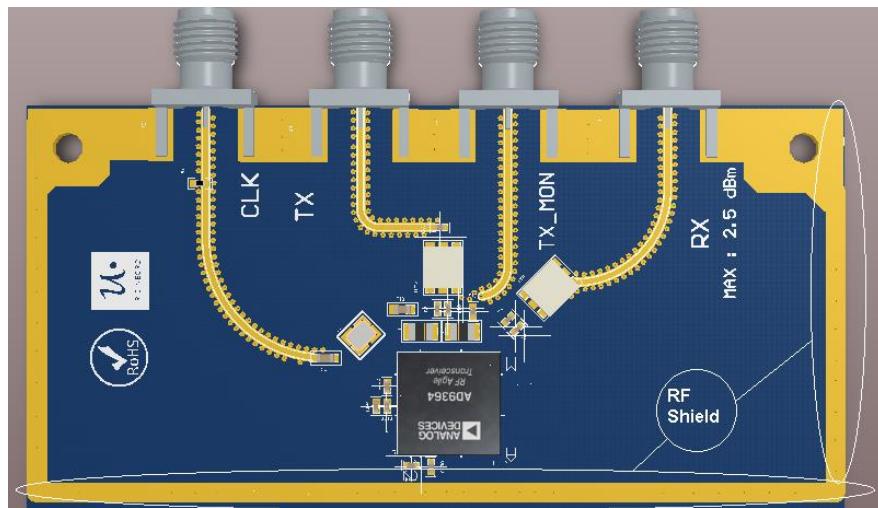
Falta completar

## B.3-AD9364

## Shielding sección RF

Se añadió la opción de poder aislar la sección de RF del transceiver AD9364 mediante la utilización de una jaula. La misma iría soldada sobre la parte expuesta de cobre, la cual tiene vías a GND.

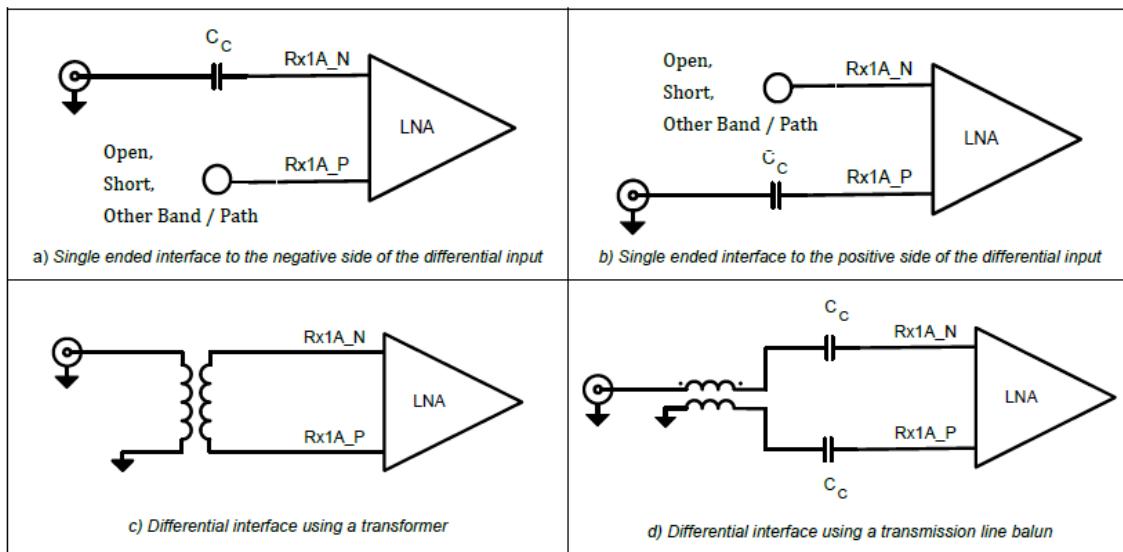
La siguiente figura muestra dicha sección:



## Interface entre el transceiver y las antenas

**Sección Rx**

Los amplificadores de bajo ruido en los puertos de Rx (Rxa, Rxb, Rxc) son funcionales en el rango de frecuencia de 70 MHz – 6 GHz. Cuando estos operan por debajo de los 3 GHz, todos los puertos de entrada de los LNA proveen una performance óptima. Cuando la frecuencia de operación está por encima de los 3GHz es conveniente utilizar los puertos de entrada Rxa y Rxb para una performance óptima. Estos tres puertos son diferenciales pero también pueden ser configurados como single ended. La recomendación por parte del fabricante es utilizarlos en modo diferencial para lograr la mejor performance en: figura de ruido y punto de intersección de segundo orden (IP2). El máximo nivel de entrada es de +2.5 dBm de pico (single ended, ideal match, 50 ohm source). La siguiente figura muestra las posibles configuraciones single ended y diferenciales para la interface:



De las dos opciones con interface diferencial se decidió implementar la opción “D”, la cual es utilizada en la placa de desarrollo de Analog Devices (modelo de referencia).

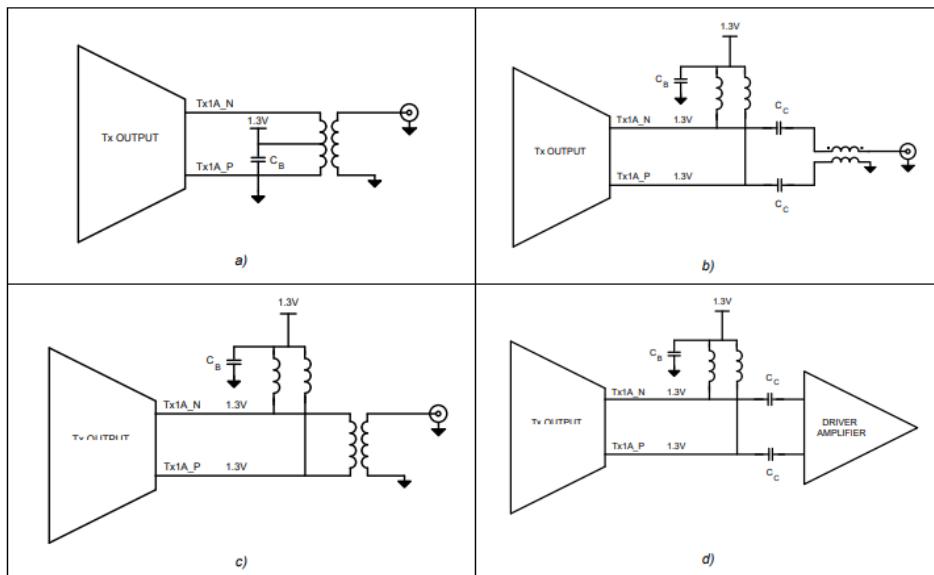
## Sección Tx

El camino de transmisión cubre todo el rango de frecuencia desde 70 MHz hasta 6 GHz, mediante 2 puertos de salida Tx los cuales tienen una performance similar. Estos puertos de salida son diferenciales y requieren una polarización externa. Esta polarización externa requiere un voltaje de 1.3V DC utilizando como conexión: inductor de choque (wire-wound inductor) o transformer center tal.

Para determinar los componentes y la topología a utilizar se tuvieron en cuenta las siguientes recomendaciones por parte del fabricante:

- La inductancia de choque debe ser elegida de tal manera que su impedancia sea alta en comparación con la impedancia de carga, para que no afecte la respuesta en frecuencia.
- Las salidas de Tx deben estar acopladas en AC debido a la presencia del voltaje de polarización de 1.3V DC.
- Una red de adaptación, balun single-ended port, es la mejor para conseguir una performance óptima.
- Utilización de componentes con baja resistencia DC para minimizar la caída de tensión debida a resistencias en serie parasitas.

La siguiente figura muestra las posibles configuraciones single ended y diferenciales para la interface:

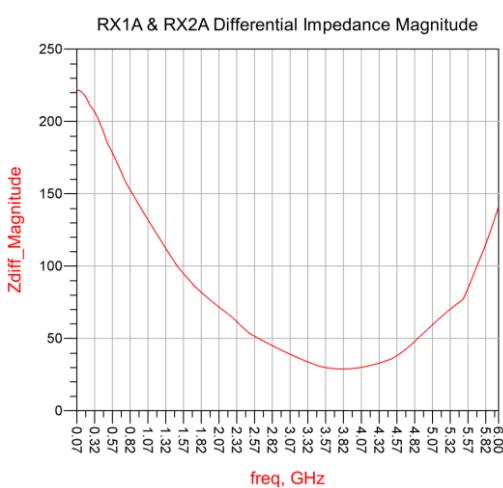


En base a las recomendaciones del fabricante, de las dos opciones que utilizan balun se decidió implementar la opción “b”.

#### Redes de adaptación del transceiver

#### Adaptación Rx

La impedancia de los puertos de RF Rx del transceiver varía según la frecuencia como se muestra en la siguiente figura:

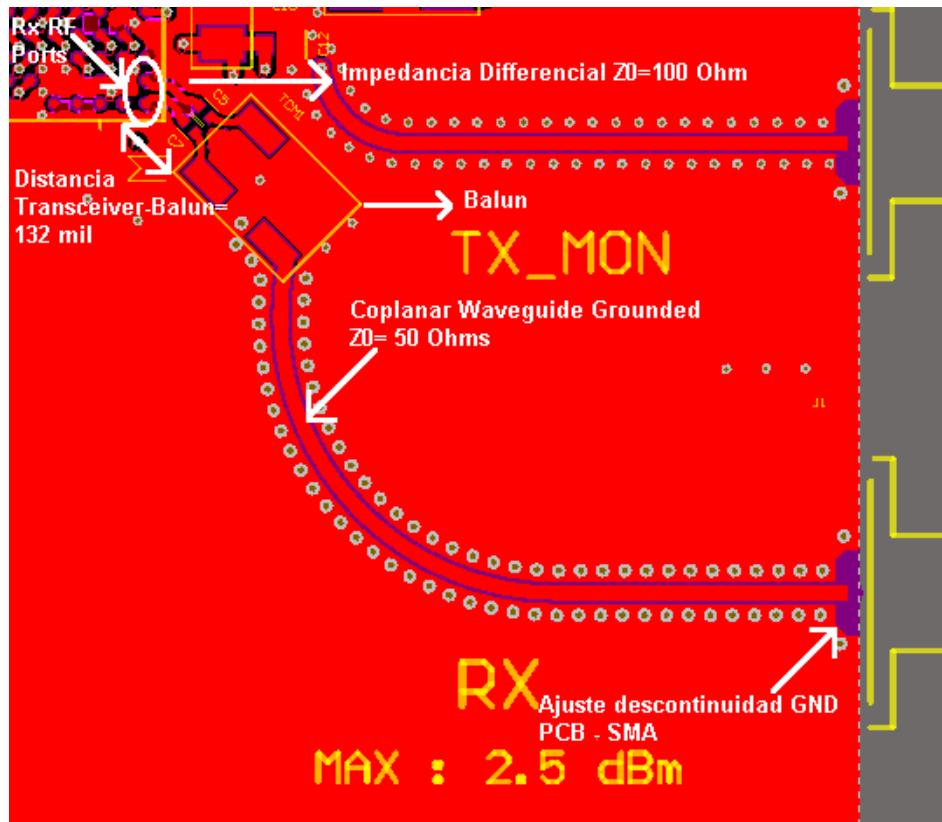


Por este motivo la interconexión entre estos puertos y el balun debe ser adaptada para su óptimo funcionamiento, estas conexiones se realizaron según el FAQ "AD9361 RF Port Interface" [25] (este FAQ es válido para el AD9364).

Esta guía establece las siguientes recomendaciones cuando se trabaja en un ancho de banda grande (70 MHz to 6 GHz):

- Conexiones lo más cortas posibles.
- Impedancia diferencial Rx de 100 Ohm.
- Utilizar Balun (1:N, Z<sub>0</sub>= 50 Ohm).

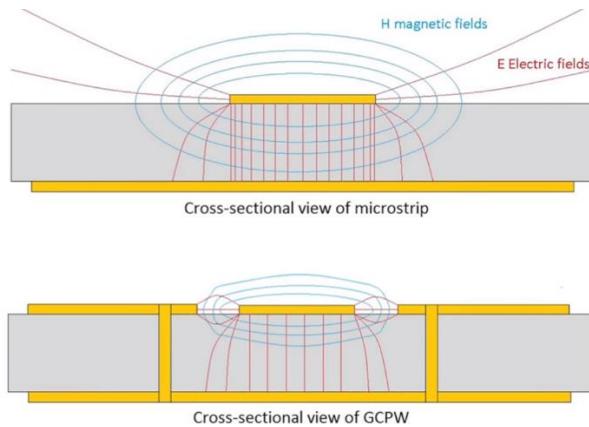
En la siguiente figura se muestra la aplicación de estas recomendaciones en el ruteo del PCB:



Para realizar la interconexión entre el balun y la antena se utilizó una coplanar waveguide Grounded (CPWG) en lugar de una microstrip.

Las coplanar waveguide tienen las siguientes cualidades en comparación a las microstrip line:

- Más simples de fabricar (menos cobre a eliminar cuando se realiza el printed circuit).
- Reducción de pérdida por radiación.
- Reducción de cross talk (campos eléctricos y magnéticos confinados).



El cálculo de impedancia de la CPWG se encuentra en el [Anexo C6](#). Para más información ver sobre CPWG [ ]

Además se ajustó de stackup y ancho de pista 50 Ohm RF para tener mismo ancho que el pin del conector SMA. Reduciendo la desadaptación de impedancia al pasar del PCB al conector.

Para determinar si la distancia de conexión entre el transceiver y el balun (132 mil) es corta se debe hacer un análisis de cuanta distancia en Lambda esta conexión representa para las diferentes frecuencias.

Para calcular Lambda (longitud de onda) se realizaron las siguientes cuentas [Pozar]:

$$\lambda = \frac{v_p}{f}$$

$$v_p = \frac{c}{\sqrt{\epsilon_{eff}}}$$

$$\epsilon_{eff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \frac{1}{\sqrt{1 + \frac{12d}{W}}}$$

En donde:

d = Altura del dieléctrico

W = Ancho del microstrip

Er = Constante dieléctrica.

Eeff = Constante dieléctrica efectiva

Vp= velocidad de propagación.

C= Velocidad de la luz.

Resolviendo estas ecuaciones:

$$\epsilon_{eff} = 2.607$$

$$V_p = 1.85 \times 10^8 \frac{m}{seg}$$

Frecuencia[GHz]	Lambda[mm]	Lambda[mil]	Distancia de 132 mil en lambdas
1	185	7283,45	0,018
2	92,5	3641,73	0,036
3	61,6	2425,19	0,054
4	46,25	1820,86	0,072
5	37	1456,69	0,091
6	30,83	1213,78	0,109

A partir de estos cálculos se puede determinar que la distancia es corta. Además se puede ver que a medida que se aumenta la frecuencia la distancia aumenta su valor en lambdas. Por lo tanto en las altas frecuencias una desadaptación de impedancia tiene más peso que en las bajas frecuencias.

Este puede ser uno de los motivos por el cual se recomienda utilizar una impedancia diferencial de 100 Ohm, ya que en las altas frecuencias la impedancia diferencial en los puertos de RF es cercana a los 100 Ohm.

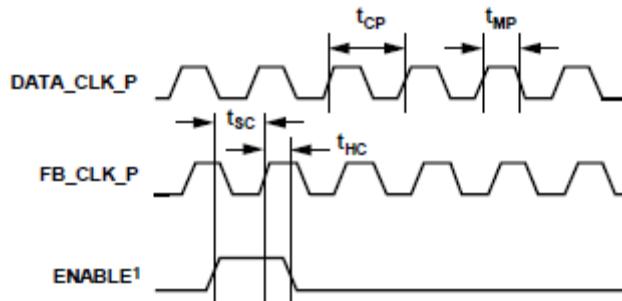
Se realizó una simulación de RF de la sección de Rx en el ADS para ver cuán adaptado esta la antena al transceiver en el rango de frecuencia de 70 MHz a 6 GHz [Anexo D2].

De forma similar se realizó el ruteo de la sección de Tx tomando como referencia el FAQ y la placa de desarrollo del AD9364.

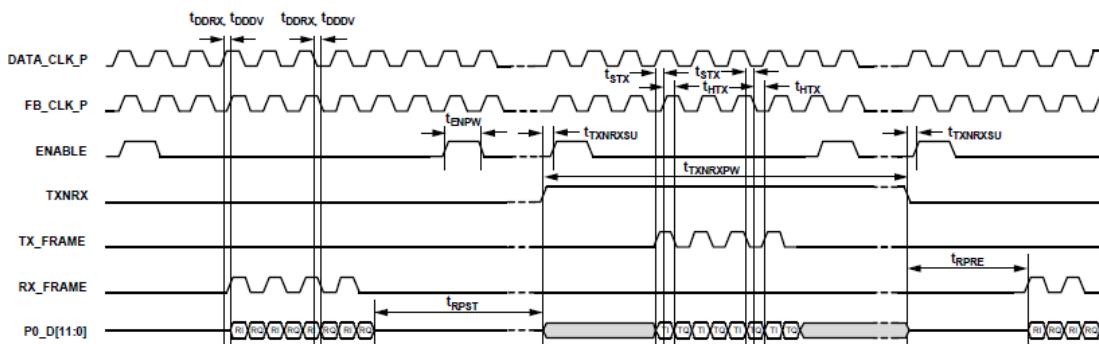
## Criterio de interface digital transceiver

En particular este dispositivo no impone explícitamente requerimientos en cuanto a la máxima diferencia de largo entre cada línea de su interface digital con el BBP. En su lugar da restricciones en cuanto a tiempos en el camino de datos y la interface SPI como muestran las siguientes figuras:

Parameter	Min	Typical	Max	Description
$t_{CP}$	16.276 ns			DATA_CLK cycle time (clock period)
$t_{MP}$	45% of $t_{CP}$		55% of $t_{CP}$	DATA_CLK and FB_CLK high and/or low minimum pulse width (including effects of duty cycle distortion, period jitter, cycle-cycle jitter and half-period jitter)
$t_{SC}$	1 ns			Control signal setup time to FB_CLK at AD9364 inputs (ENABLE, TXNRX)
$t_{HC}$	0 ns			Control signal hold time from FB_CLK at AD9364 inputs (ENABLE, TXNRX)
$t_{STX}$	1 ns			Tx data setup time to FB_CLK at AD9364 inputs
$t_{HTX}$	0 ns			Tx data hold time from FB_CLK at AD9364 inputs
$t_{DDRX}$	0 ns	1.5 ns	1.2 ns	Rx data delay from DATA_CLK to D[11:0] outputs - 1.8 V supply
				Rx data delay from DATA_CLK to D[11:0] outputs - 2.5 V supply
$t_{DDDV}$	0 ns		1.0 ns	Rx data delay from DATA_CLK to Rx_FRAME
$t_{ENPW}$	$t_{CP}$			ENABLE pulse width (edge-detected by FB_CLK)
$t_{TXNRXPW}$	$t_{CP}$			TXNRX pulse width (edge-detected by FB_CLK)
$t_{TXNRXSU}$	0 ns			TXNRX setup time to ENABLE
$t_{RPRE}$	$2 \times t_{CP}$			Time at which BBP stops driving D[11:0] before a receive burst, TDD
$t_{RPST}$	$2 \times t_{CP}$			Time at which BBP starts driving D[11:0] after a receive burst, TDD



Data Port Timing Parameter Diagrams – Data Reference Clocks and Hardware Control Inputs (CMOS Bus Configuration)



Data Port Timing Parameter Diagrams – CMOS Bus Configuration (Example shown is Single Port, DDR, TDD Operation)

En lugar de realizar cálculos de cuanto es la velocidad de propagación de las señales dentro de la placa y cuanta distancia representa estos tiempos. Se optó por ajustar a “cero” (unos pocos mils) la diferencia de largo de estas líneas.

Faltaría mostrar layout y tabla de largos cumpliendo este criterio

## C-Herramientas utilizadas durante el diseño.

### C.1-Estimación Consumo FPGA

Para estimar el consumo de la FPGA se utilizó la herramienta “Xilinx® Power Estimator (XPE)”. Esta herramienta considera los recursos utilizados en el diseño, como por ejemplo: toggle rates, I/O loading y otros factores que combinados son utilizados para estimar la distribución de power.

La estimación de power para dispositivos programables es un proceso complejo porque depende en gran medida de la cantidad de lógica en el diseño y la configuración de la misma.

Para estimar este consumo, esta herramienta pide como mínimo los siguientes parámetros de entrada:

- Dispositivo, encapsulado y grado de velocidad.
- Cantidad de recursos utilizados (flip-flops, look-up tables. Block RAM, etc.).
- Frecuencia de clock.
- Una estimación del data toggle rate.
- Uso de memoria externa.
- Temperatura del medio en el que va a operar

### **Toggle rate**

Obtener un toggle rate preciso es esencial para que la estimación de power sea correcta. Esta información no está disponible en esta etapa del diseño ya que depende de la lógica programada y el funcionamiento de la misma. Para estimar este valor Xilinx da como guía los siguientes puntos a tener en cuenta:

- Para un camino sincrónico, el toggle rate refleja que tan seguidas las salidas cambian con respecto al clock de entrada. Este cambio es modelado de forma porcentual entre 0-100% (200% para DDR). Un máximo de toggle rate en los datos significa que la salida cambia en cada flanco ascendente de clock.

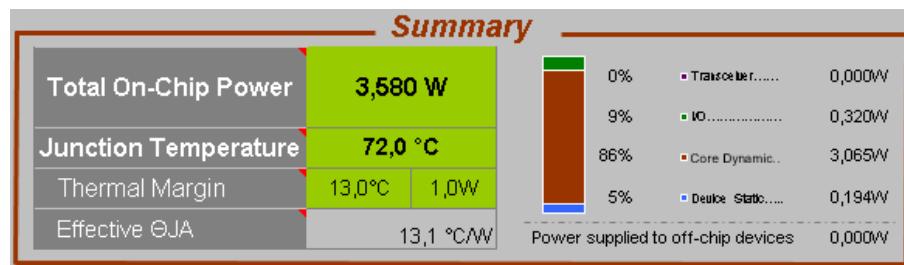
- Para señales no periódicas, el toggle rate no es fácil de predecir. Un método efectivo para estimar el promedio de tasa de cambio para un determinado diseño, consiste en dividir al mismo en secciones por funcionalidad. Luego calcular el toggle rate en cada sección y obtener un promedio de todos. En general la mayoría de los diseños con una gran cantidad de lógica tienen un 12.5% de promedio de toggle rate (este porcentaje es el que trae por default la herramienta).

Teniendo en cuenta esta guía y sobredimensionando, se decidió utilizar los siguientes parámetros para obtener una estimación del consumo de la FPGA:

- 100% de uso de recursos de la FPGA (peor caso).
- 40% de toggle rate (a una frecuencia de 150 Mhz y 50% de uso)
- DDR3I external memory de 16 bits de datos (memoria usada).
- Temperatura ambiente de 25 °C (environement still air).
- Modelo de la FPGA: Artix-7-XC7A100T-1FTG256C.

Con estos parámetros la herramienta nos da los siguientes datos de salida:

- Consumo y temperatura de la FPGA:



- Corriente necesaria por cada tensión de alimentación de la FPGA:

Power Supply		
Source	Voltage	Total (A)
V <sub>CCINT</sub>	1,000	2,899
V <sub>CCBRAM</sub>	1,000	0,020
V <sub>CCUX</sub>	1,800	0,254
V <sub>CCUX_IO</sub>		
V <sub>CCO 3.3V</sub>	3,300	
V <sub>CCO 2.5V</sub>	2,500	
V <sub>CCO 1.8V</sub>	1,800	
V <sub>CCO 1.5V</sub>	1,500	
V <sub>CCO 1.35V</sub>	1,350	0,125
V <sub>CCO 1.2V</sub>	1,200	
V <sub>CCADP</sub>	1,800	0,020

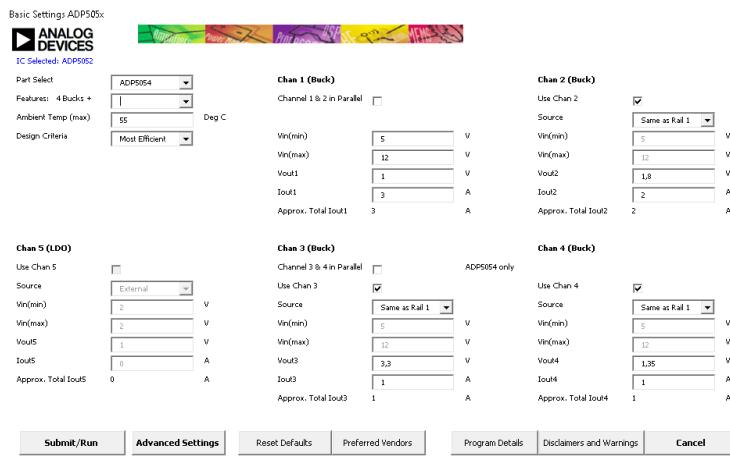
Esta última tabla ayudo a determinar las corrientes necesarias para cada buck regulator del integrado ADP5054.

### C.2-Programa ADP5054 criterio más eficiente

Las conexiones y componentes de este integrados se obtienen mediante el programa “505x BuckDesigner” brindado por Analog Devices. Este programa requiere como datos de entrada:

- Modelo del Regulador
- Temperatura máxima del ambiente.
- Criterio de diseño (más eficiente, menor espacio, menor costo): en este caso se decidió priorizar la eficiencia.
- Tensión de alimentación de cada buck.
- Tensión y corriente de salida de cada buck.

La siguiente figura muestra el menú en donde ingresar dichos parámetros de entrada.



A partir de estos parámetros se obtuvo:

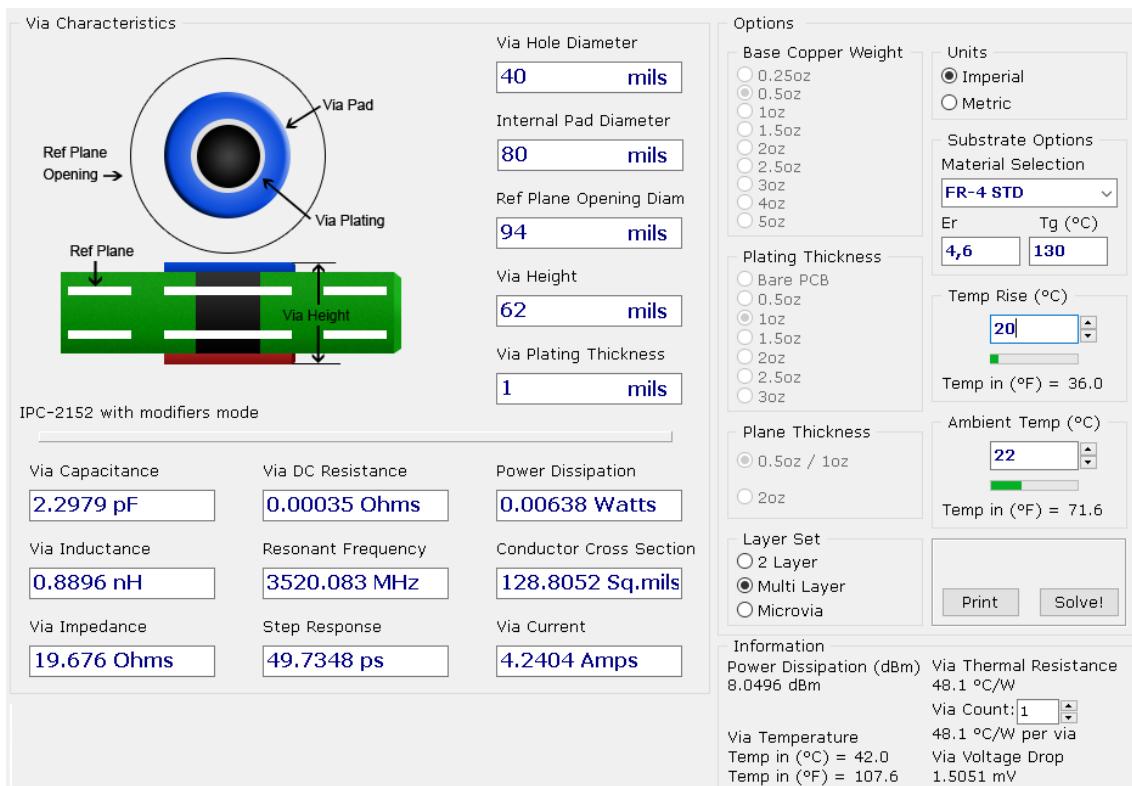
- Conexiones del circuito
- Componentes a utilizar
- Diagramas de eficiencia, perdida de potencia y ganancia

**Agregar imagen de circuito componentes y diagramas?**

### C.3-Cálculo de corriente por vías.

Se corrobora que el tamaño de vías de los header va a ser suficientes para soportar la corriente máxima de cada fuente de alimentación (peor caso 3 A para la tensión de 1V FPGA).

Utilizando la herramienta: Saturn PCB Design, se calculó la corriente máxima que soportan las vías.



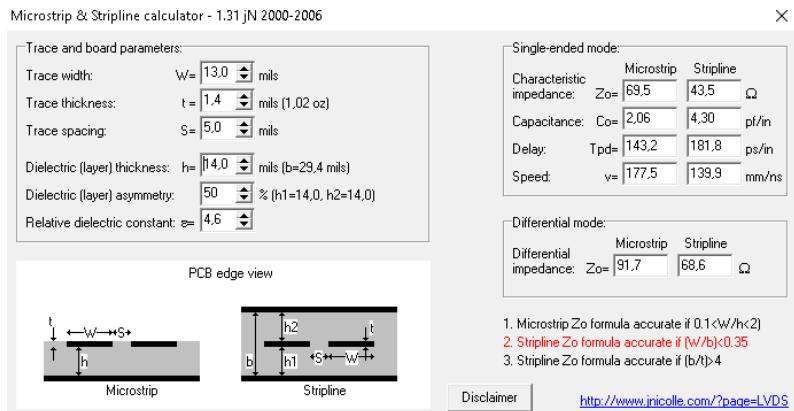
Se estimó que los headers (tamaño de agujero = 40 mils y largo =62 mils) soportan 4.2 A de corriente máxima.

### C.4-Cálculo Impedancia diferencial

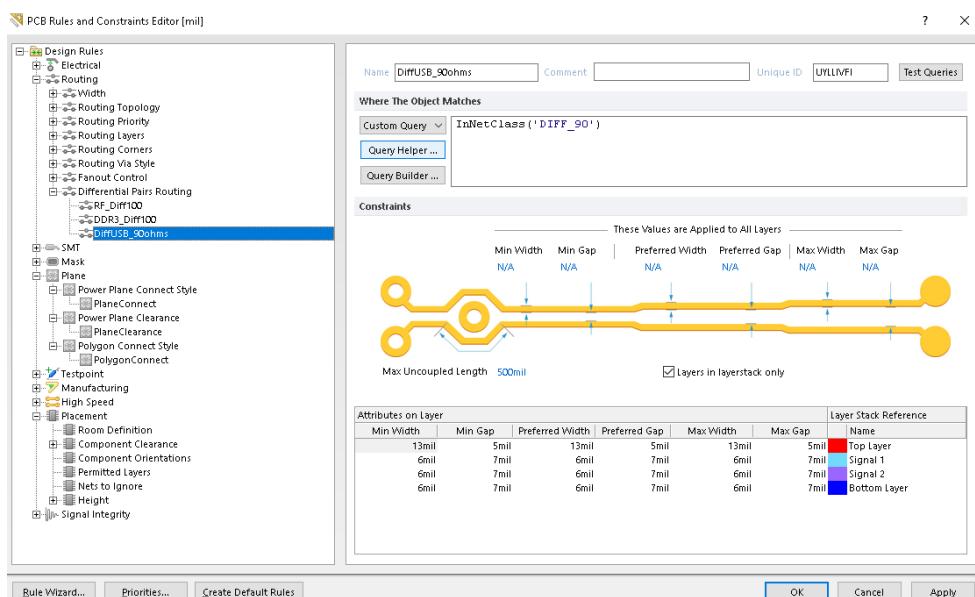
Para calcular las impedancias diferenciales se utilizó el programa “Microstrip & Stripline calculator” desarrollado por Jean Nicolle.

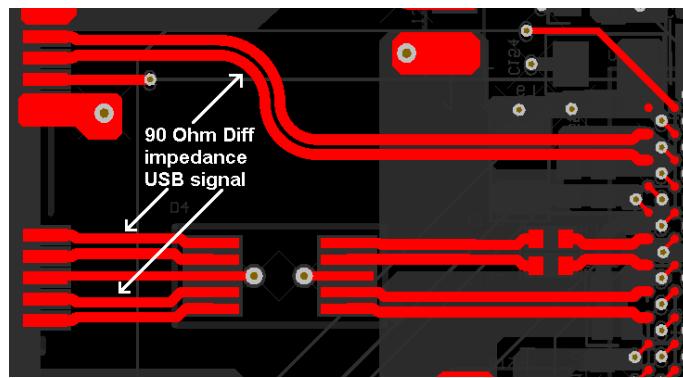
El mismo solo necesita como datos de ingreso la información del stack-up y variando el ancho pista y la separación entre las mismas se puede obtener la impedancia deseada.

Resultados para la impedancia de 90 Ohm +/- 10% requerida por el controlador USB 3.0



Con estos valores de ancho de pista y separación de las mismas se creó una regla de ruteo en el Altium Designer para señales diferenciales, como se muestra en las siguientes figuras.





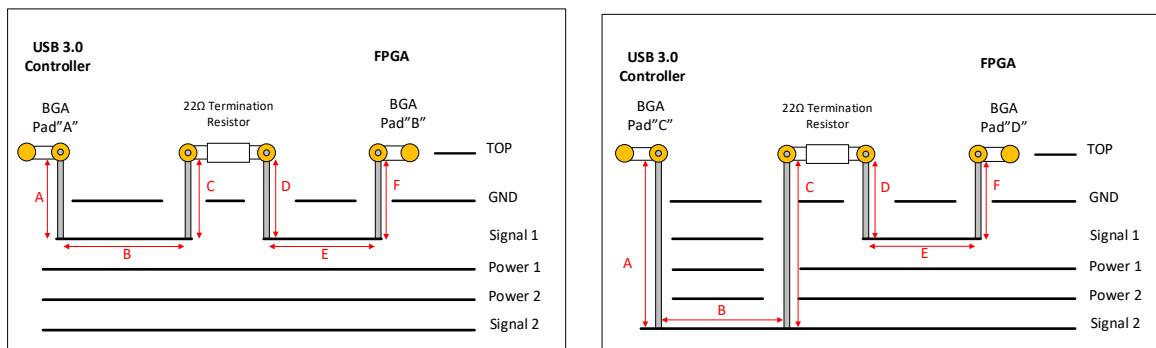
Del mismo modo se calculó las demás impedancias diferenciales.

#### C.5-Uso de herramienta xSignals

Algunos de los requerimientos de las conexiones entre el controlador del USB 3.0 y la FPGA son los siguientes:

- Utilización de resistencia de 22 ohm para evitar reflexiones
- Largo máximo de cada línea 5 pulgadas (5000 mil).
- Max diferencia de largo entre las líneas 500 mil.

El principal problema es llevar un seguimiento del largo de cada línea, ya que el ruteo de las mismas se lleva a cabo en diferentes capas y tramos (del controlador a la resistencia y de la resistencia a la FPGA). Por ejemplo las siguientes figuras muestran dos posibles escenarios de conexión entre la FPGA y el controlador.



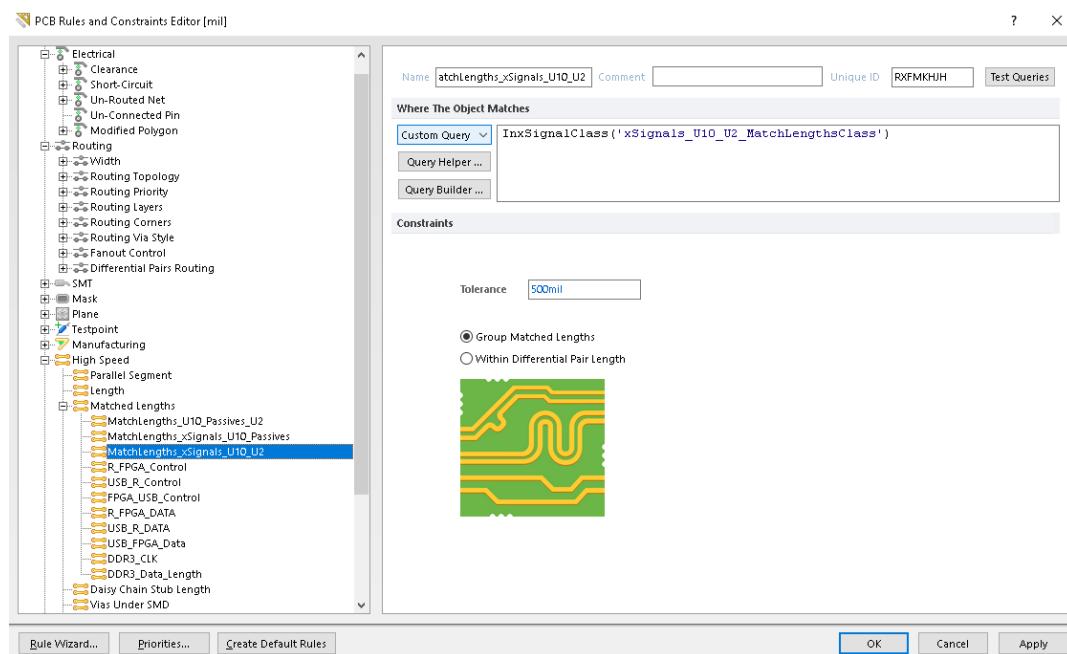
Para cumplir los requerimientos de máxima diferencia entre líneas, la distancia  $A+B+C+D+E+F$  de ambas figuras deben ser iguales o menor a 500 mils. Para poder controlar estas distancias se utilizó la herramienta xSignals del Altium Designer, la cual lleva un control de las distancias totales de conexiones entre ambos componentes (el controlador y la FPGA). Esta herramienta en particular está disponible a partir de la versión de Altium Designer 16.0. Si se dispone de una

versión anterior hay que sumar manualmente (con una tabla en Excel) cada tramo.

Se descartó la interconexión por medio de la top layer por los siguientes motivos:

- Al ser esta distancia muy corta, crearía una gran restricción para los pad internos de la FPGA ya que su fanout se realiza por vías y por ende su camino será mucho más largo.
- Si se rutea por top algunas señales y otras por capas internas, estas estarían viajando por medios diferentes (microstrip y stripline) por lo que sus velocidades de propagación serían diferentes.
- No hay suficiente espacio en la top layer debido a la presencia de los capacitores de desacople cercanos al integrado.

Mediante la utilización de la misma se genera una regla de ruteo, con la cual se controla el largo de las interconexiones entre ambos integrados. La siguiente figura muestra una de estas reglas:



En donde U10 es el controlador del USB 3.0, U2 es la FPGA y la tolerancia de 500 mil es el requerimiento.

En las siguientes tablas muestran las distancias de las conexiones entre el controlador de USB 3.0 y la FPGA:

Name	Routed Length (mil)	Name	Routed Length (mil)	Name	Routed Length (mil)
DQ22	1464	DQ30	1570	CTL0	1476
DQ4	1480	DQ26	1576	CTL1	1483
DQ5	1487	DQ12	1579	CTL2	1470
DQ8	1488	DQ25	1579	CTL3	1552
DQ21	1491	DQ23	1580	CTL4	1486
DQ20	1494	DQ19	1581	CTL5	1569
DQ7	1499	DQ27	1585	CTL6	1559
DQ16	1515	DQ3	1590	CTL7	1472
DQ15	1517	DQ28	1593	CTL8	1542
DQ13	1524	DQ14	1608	CTL9	1599
DQ1	1527	DQ17	1624	CTL10	1509
DQ2	1534	DQ31	1635	CTL11	1537
DQ18	1538	DQ6	1637	CTL12	1557
DQ11	1550	DQ10	1653	PCLK	1724
DQ29	1568	DQ0	1671		
DQ9	1569	DQ24	1676		

Como se puede apreciar en las tablas, la máxima diferencia de longitud entre las líneas de interface USB-FPGA es de 260 mil (1724-1464), cumpliendo con el requerimiento de 500 mil de máxima diferencia de longitud.

#### C.6-Coplanar waveguide y ajuste de impedancia

Para calcular la impedancia de las Coplanar Waveguide Grounded (CPWG) se utilizó la herramienta: Saturn PCB Design, la misma pide como datos de entrada: el ancho de la pista (W), separación entre GNDs adyacentes (G), distancia al plano de GND más cercano (H) y la constante dieléctrica del material (Er). Obteniendo como resultado la impedancia de la coplanar waveguide, la siguiente figura muestra el resultado obtenido con esta herramienta:

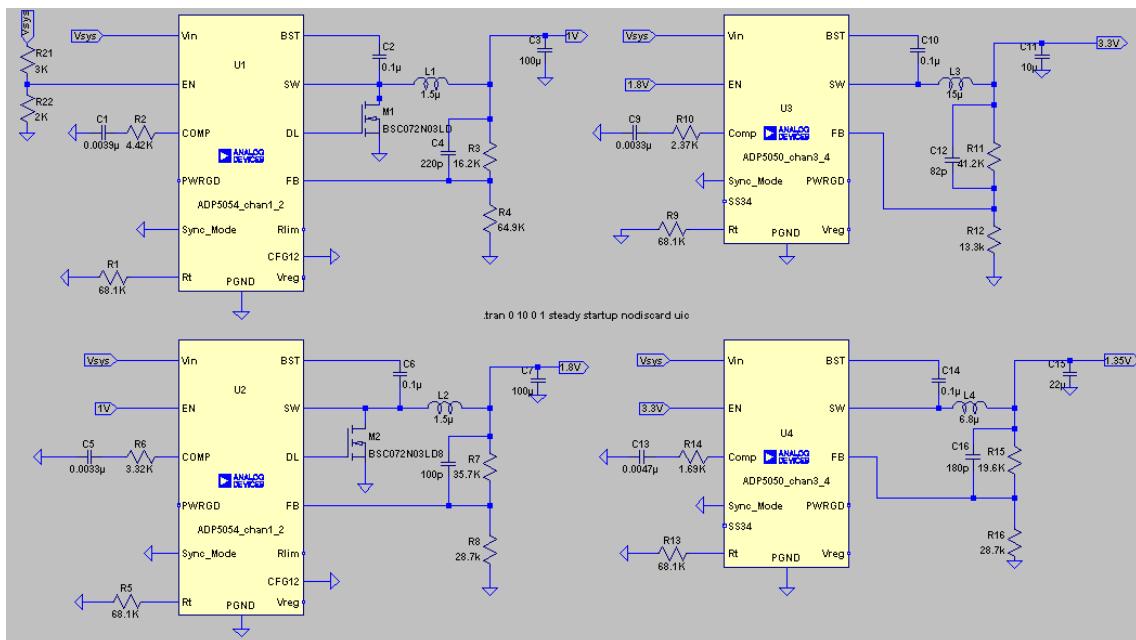


## D- Simulaciones

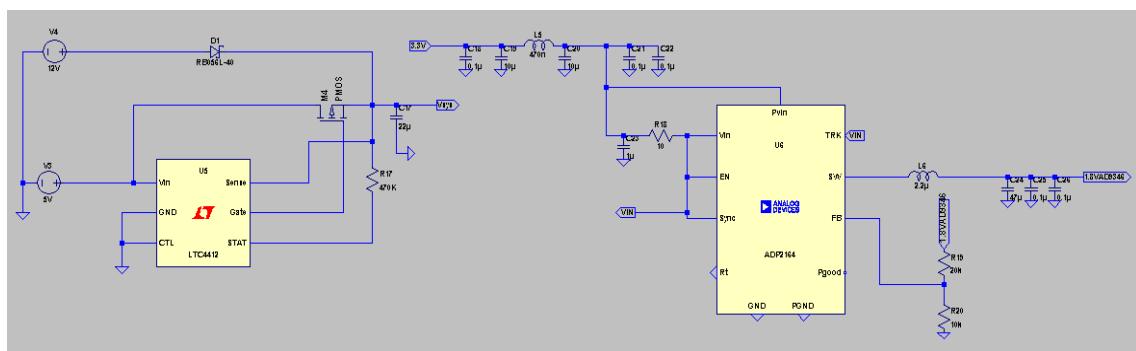
### D.1-Simulaciones Power

Se simulo una gran parte de la sección de alimentación utilizando la herramienta LTspice. En esta simulación se pudo observar: el correcto funcionamiento del selector de alimentación LTC4412, como la de los convertidores DC-DC: ADP5054, ADP2164.

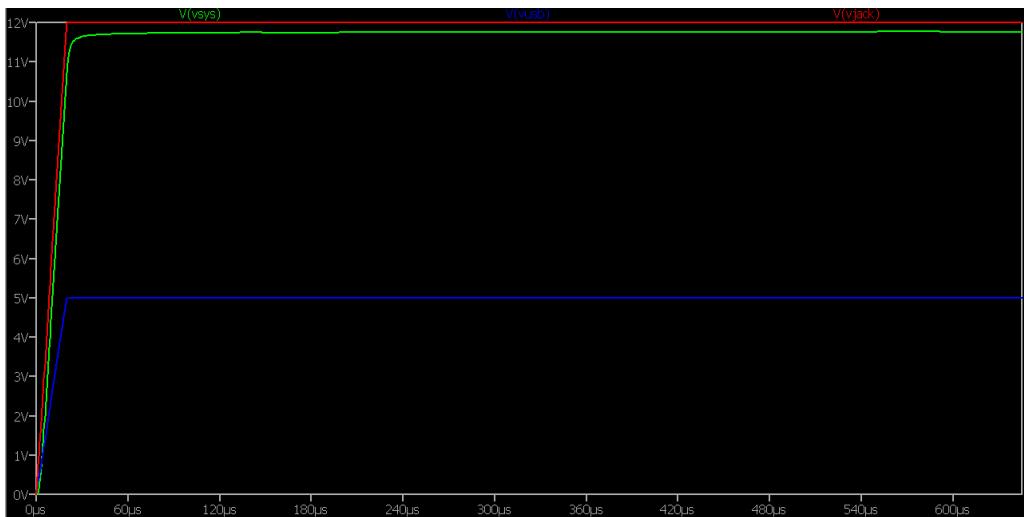
Las siguientes figuras muestran el circuito y los resultados obtenidos durante la simulación



Esquemático de los 4 buck reguladores del adp5054

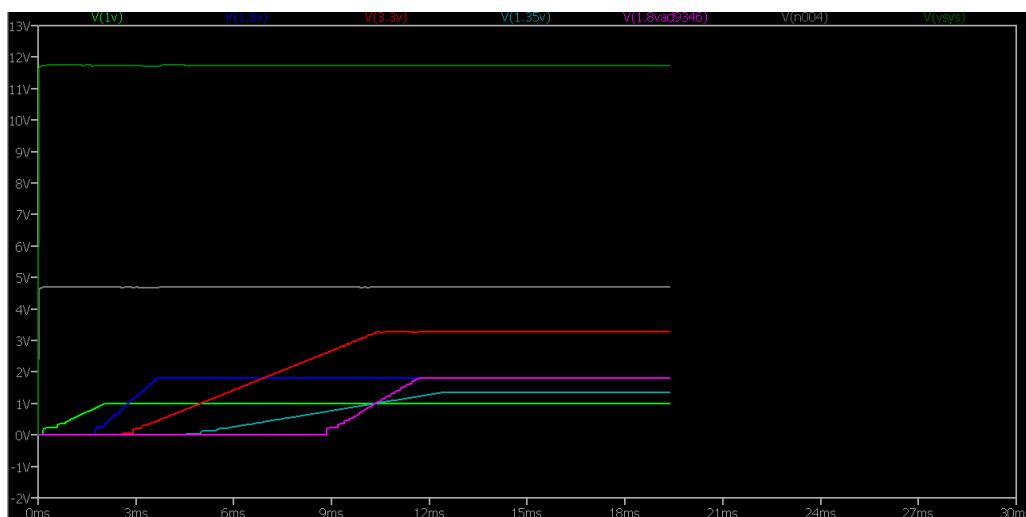


Esquemático del selector de power y el regulador ADP2164



En esta imagen se puede apreciar el correcto funcionamiento del selector de power LTC4412, ya que ante la presencia de ambas tensiones de alimentación (USB y Jack), la tensión de salida es la provista por el Jack.

Imagen de tensiones:



En esta figura se puede apreciar que los niveles de tensión y la secuencia de encendido de la FPGA son los correctos.

#### D.2-Simulaciones etapa de RF AD9364 con ADS

Simulación de la sección de Rx, desde los puertos diferenciales Rx del AD9364 hasta el puerto SMA para las antenas.

La simulación consistió en sacar los parámetros S de la sección de RF Rx del PCB, utilizando la herramienta Momentum del ADS. Luego se realizó una simu-

lación conjunta con parámetros S del PCB y los parámetros S de los componentes (puerto diferencial AD9364, capacitores en serie y balun). Obteniendo como resultado una estimación del parámetro S11 (en el puerto del SMA). El mismo indica el nivel de desadaptación en el rango de frecuencia entre 70Mhz y 6Ghz.

**Falta Completar**

## E- Información general y hoja de datos

### E.1-Recomendaciones por parte de los principales IC

<b>Integrado</b>	<b>Nombre</b>	<b>Comentarios</b>
<b>FPGA</b>	Recommended Design Rules and Strategies for BGA Devices-UG1099	<ul style="list-style-type: none"> <li>• Fanout del BGA</li> </ul>
	7 Series FPGAs PCB Design Guide -UG483	<ul style="list-style-type: none"> <li>• Capacitores a usar y sus posiciones.</li> <li>• Orden power layer en stackup.</li> <li>• Impedancia de pistas.</li> </ul>
<b>AD9364</b>	AD9364 Reference Manual UG-673. FAQ AD9364 RF Port Interface	<ul style="list-style-type: none"> <li>• PCB material y selección del stack</li> <li>• RF layout.</li> <li>• Fan-out and trace-space layout guidelines</li> <li>• Colocación de componentes y guía de ruteo</li> <li>• Power management y consideraciones de ruido</li> </ul>
<b>USB 3.0</b>	Hardware Design Guidelines and Schematic Checklist- AN70707	<ul style="list-style-type: none"> <li>• Recomendaciones a seguir tanto en hardware como en el esquemático.</li> </ul>
<b>DDR3 Memory</b>	Hardware Tips for Point-to-Point System Design: Termination, Layout, and Routing	<ul style="list-style-type: none"> <li>• Recomendaciones a seguir tanto en hardware como en el esquemático.</li> </ul>

### E.2-Capacidades Fabricante PCB Prototype

Toda la información en cuanto a las capacidades de fabricación de PCB Prototypes se encuentra en el siguiente link:

<https://wwwpcbprototype.com/en/pcb-capabilities>

La mayoría de estas capacidades fueron impuestas como requerimientos del PCB en forma de reglas de ruteo.

## E.3-Información sobre conexiones de cada sección

Integrado	Sección	Guía de usuario
FPGA	Slave Serial Configuration. JTAG	7 Series FPGAs Configuration-UG470
	DDR3 Memory	7 Series FPGAs Memory Interface Solutions-UG586
	I/O	7 Series FPGAs SelectIO Resources-UG471
	Clock	7 Series FPGAs Clocking Resources-UG472
USB 3.0	Interface GPIF FPGA	Designing with the EZ-USB® FX3™ Slave FIFO Interface AN65974
	Boot FPGA	Configuring an FPGA Over USB Using Cypress EZ-USB® FX3™ AN84868
	Interface USB	EZ-USB® FX3™/FX3S™ Hardware Design Guidelines and Schematic Checklist AN70707
	Boot USB 3.0	EZ-USB® FX3™/FX3S™ Boot Options-AN76405
AD9364	Interface	AD9364 Reference Manual UG-673

## E.4-Esquemático, Gerbers, BOM

Todos los archivos del Proyecto fueron almacenados en el siguiente repositorio de Github:  
<https://github.com/OmarLopezCabrera/Tesis-SDR-LAPAC-Open-Source>

## Referencias

- [1]. Mitola J. and Maguire G. Q. Cognitive radio: Making software radios more personal. IEEE Personal Communications, 6, 13–18, August 1999.
- [2]. R. J. Lackey and D. W. Upmal, “Speakeasy: the Military Software Radio”, IEEE Communications Magazine, vol. 33, pp. 56-61, 1995.
- [3]. AD9364 Reference Manual UG-673. [[Link](#)]
- [4]. EZ-USB FX3 Technical Reference Manual. [[Link](#)]
- [5]. 2 Gb: x4, x8, x16 DDR3L SDRAM Description. [[Link](#)]
- [6]. All Programmable 7 Series Product Selection Guide. [[Link](#)]
- [7]. Información sobre USRP B200 mini. [[Link](#)]
- [8]. Información sobre Lime SDR. [[Link](#)]
- [9]. Información sobre BladeRF xA4. [[Link](#)]
- [10]. Implementing Carrier Phase Recovery Loop Using Vivado HLS Author: Alex Paek, Duncan Mackay, May 2, 2013. [[Link](#)]
- [11]. Quick start guide MicroBlaze. [[Link](#)]
- [12]. Xilinx Power Estimator User Guide. [[Link](#)]
- [13]. RF Agile Transceiver AD9364 data sheet. [[Link](#)]
- [14]. USB 3.0 Core IO Power. [[Link](#)]
- [15]. DDR3 Power Calculation. [[Link](#)]
- [16]. R.Machado and A. Wyglinski, “Software-defined radio: Bridging the analog digital divide” Proceeding of the IEEE, vol. 103 no. 3, pp 409-423, march 2015. [[Link](#)]
- [17]. Software-Defined Radio for Engineers, by Travis F. Collins, Robin Getz, Di Pu, and Alexander M. Wyglinski, 2018, pp 171-190.
- [18]. FPGA-based Implementation of Multiple PHY Layers of IEEE 802.15.4 Targeting SDR Platform. [[Link](#)]
- [19]. AXI Reference Guide. [[Link](#)]
- [20]. Información sobre libiio Analog Devices. [[Link](#)]
- [21]. Información drivers transceiver AD9364. [[Link](#)]
- [22]. EZ-USB FX3 Software Development Kit. [[Link](#)]
- [23] Using the FX3 SDK on Linux Platforms [[Link](#)]

[24]. Cuadro de atribución de bandas de frecuencias de la República Argentina  
[[Link](#)]

[25].FAQ AD9364 RF Port Interface [[Link](#)]

[]]. Informacion sobre CPWG [[Link](#)]

[]]. Informacion sobre Balun: Application Note on Transformers (AN-20-002)  
Minicircuits.

## Bibliografia

- 1-Eugene Grayver-Implementing Software Defined Radio
- 2-Software-Defined Radio for Engineers, by Travis F. Collins, Robin Getz, Di Pu, and Alexander M. Wyglinski
- 3-Software-Defined R ware-Defined Radio Handbook 14th Edition (Pentek)
- 4-W. Tuttlebee, Software Defined Radio: Origins, Drivers and International Perspectives. John Wiley & Sons, 2002
- 5- Software-defined Radios: Architecture, State-of-the-art, and Challenges Rami Akeela, and Behnam Dezfouli
- 6- Microwave Engineering- David M.Pozar