



**UNIVERSIDAD NACIONAL DE RÍO NEGRO**

---

**PROYECTO FINAL INTEGRADOR**

**Diseño e implementación de una  
Radio Definida por Software**

---

**Hecho por:**

**Omar López Cabrera**

**Director: Franco Alcaraz**

**Codirector: 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>	11
1.1 Que es una radio definida por software	11
1.2 Por qué utilizar radios definidas por software?	11
1.3 Aplicaciones y características de las SDR	11
1.4 Requerimientos/ objetivos de la tesis	2
1.5 Metodología de diseño	3
<b>2. Diseño</b>	5
2.1 Arquitectura general según funciones	5
2.1.1 Antenas	5
2.1.2 RF Front-End	5
2.1.3 Digital Front-End	6
2.1.4 CPU	6
2.1.5 Memoria	6
2.1.6 Conexiones externas	6
2.1.7 Interfaces	6
2.1.8 Alimentación	7
2.2 Arquitectura definida	7
2.2.1 CPU	8
2.2.1.1 Selección de FPGA (Falta completar)	9
2.2.1.2 Encapsulado y PinOut	11
2.2.1.3 Función	11
2.2.1.4 Configuración FPGA	12
2.2.1.5 Alimentación FPGA	12
2.2.2 RF Front End & Digital Front End	12

2.2.2.1 Modos de funcionamiento del transceiver .....	16
2.2.3 USB 3.0 .....	20
2.2.3.1 Información general USB 3.0.....	20
2.2.3.2 Controlador USB 3.0 CYUSB3014 .....	21
2.2.4 Alimentación SDR .....	23
2.2.5 Memoria DDR3 .....	25
2.2.6 GPIO y Leds.....	26
2.2.7 SDR Características.....	26
<b>3. Implementación del Diseño .....</b>	<b>28</b>
3.1 Diseño herramienta CAD.....	28
3.1.1 Esquemático .....	28
3.1.2 Componentes .....	29
3.2 Conexiones .....	30
3.2.1 FPGA.....	30
3.2.1.1 Bancos I/O .....	30
3.2.1.2 Booting .....	31
3.2.2 Power .....	33
3.2.3 Transceiver AD9364 .....	33
3.2.4 USB 3.0 .....	33
3.3 PCB información básica .....	34
3.3.1 PCB definición .....	34
3.3.2 Mascara de soldadura (Solder Mask).....	35
3.3.3 Mascara de Leyendas (Silk Mask).....	35
3.3.4 Pistas .....	35
3.3.5 Pads .....	35
3.3.6 Perforación Platinada (Plated Hole) .....	35
3.4 Capacidades del fabricante PCB.....	36
3.5 Distribución de componentes y características físicas.....	37
3.6 Stackup .....	38
3.7 Líneas de transmisión.....	41
3.8 Impedancia característica, reflexión y terminación .....	41
3.9 Líneas diferenciales .....	43
3.10 Ajuste de longitudes.....	44
3.11 Algunas Consideraciones.....	46

3.11.1 FPGA .....	46
3.11.2 Power .....	46
3.11.3 USB 3.0 .....	46
3.11.4 AD9364 .....	46
3.12 Planos de alimentación y tierra.....	46
3.13 Capacitores desacople .....	47
3.14 Modelo 3d .....	49
<b>4. Sumario .....</b>	<b>51</b>
4.1 Simulación de integridad de la señal.....	51
4.2 Fabricación y ensamblado de la PCB.....	51
4.3 Implementación en VHDL y drivers.....	52
4.4 Posibles Mejoras .....	52
4.5 Conclusiones.....	53
<b>Anexos.....</b>	<b>54</b>
A-Cuestiones generales del diseño .....	54
Alimentación .....	54
Estimación Consumo FPGA .....	54
Programa ADP5054 criterio más eficiente .....	56
Vías para escape térmico .....	57
Header test por etapa .....	58
Calculo de corriente por vías.....	58
Simulaciones Power .....	58
USB 3.0 .....	60
Calculo Impedancia diferencial .....	60
Protección ESD .....	62
Uso de herramienta xSignals.....	62
AD9364.....	63
Shielding sección RF .....	63
Coplanar waveguide y ajuste de impedancia.....	64
Balun.....	<b>¡Error! Marcador no definido.</b>
Criterio de interface digital .....	66
B- Información general y hoja de datos .....	67
Recomendaciones por parte de los principales IC .....	67
Capacidades Fabricante PCB Prototype .....	68

Información sobre conexiones de cada sección .....	68
Esquemático, Gerbers, BOM .....	69
Referencias.....	70
Bibliografía .....	71

## Indice de Figuras

## Índice de Tablas

## **Lista de Acrónimos**

**ADC:** Analog to Digital Converter

**ASICS:** application specific integrated circuit

**BB:** Base Band

**BBA:** Base Band Amplifier

**BBP:** Base Band Processor

**BGA:** Ball Gate Array

**BOM:** Bill of Materials

**CPU:** Central processing unit.

**CAD:** Computer Assisted Design

**DAC:** Digital to Analog Converter

**DDC:** Digital Down Converter

**DDR:** Double Data Rate

**DMA:**

**DUC:** Digital Up Converter

**DSP:** Digital Signal Processor

**EEPROM**

**FAQ:** Frequently Asked Questions

**FDD:** Frequency

**FPGA:** Field Programmable Gate Array

**GPIO:** General purpose input output

**GPP:** general purpose processor

**IC**

**JTAG**

**LPF**

**LAPAC**

**MIMO:**



**PCB**

**RF**

**RFFE**

**ROM**

**SDR** Software Defined Radio

**SPI**

**USB**

**XPE**

## 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 y una FPGA Artix-7. 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 si 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 Que es una radio definida por software

El termino radio definida por software fue utilizado por Joe Mitola en 1991 en referencia a las radios reprogramables o reconfigurables. 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, escondiendo

comunicaciones durante operaciones conjuntas. Una SDR multicanal puede actuar como intérprete de las diferentes radios.

**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 forma de onda de baja potencia 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).

**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, reduciendo 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 Requerimientos/ objetivos de la tesis

La finalidad de este Proyecto Final integrador es el diseño e implementación de una Radio Definida por Software (SDR: Software Defined Radio). Además de este objetivo principal el plan de trabajo estipulaba 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.

Falta imagen ilustrativa

- Entablar 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.

Falta imagen ilustrativa

-Conexión USB 3.0

Al comenzar el diseño se agregaron otros requerimientos al proyecto:

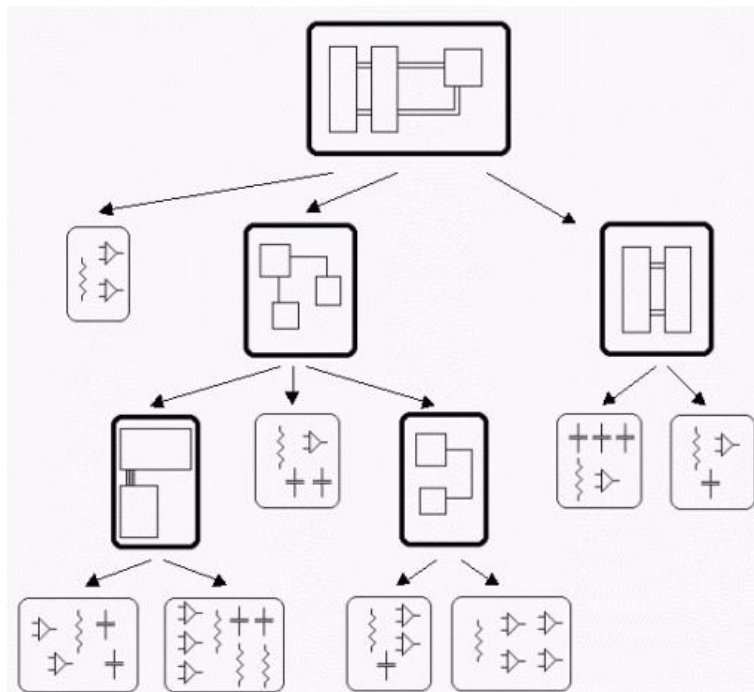
-Frecuencia de operación de 100 MHz hasta 3 GHz

-Ancho de banda de 40 MHz

-Memoria DDR3

### 1.5 Metodología de diseño

Para realizar este proyecto se utilizó 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.



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
- Fabricación y ensamblado de la placa
- Pruebas

Debido a la envergadura del proyecto no se pudieron realizar las últimas tres etapas, quedando estas como trabajo a futuro.

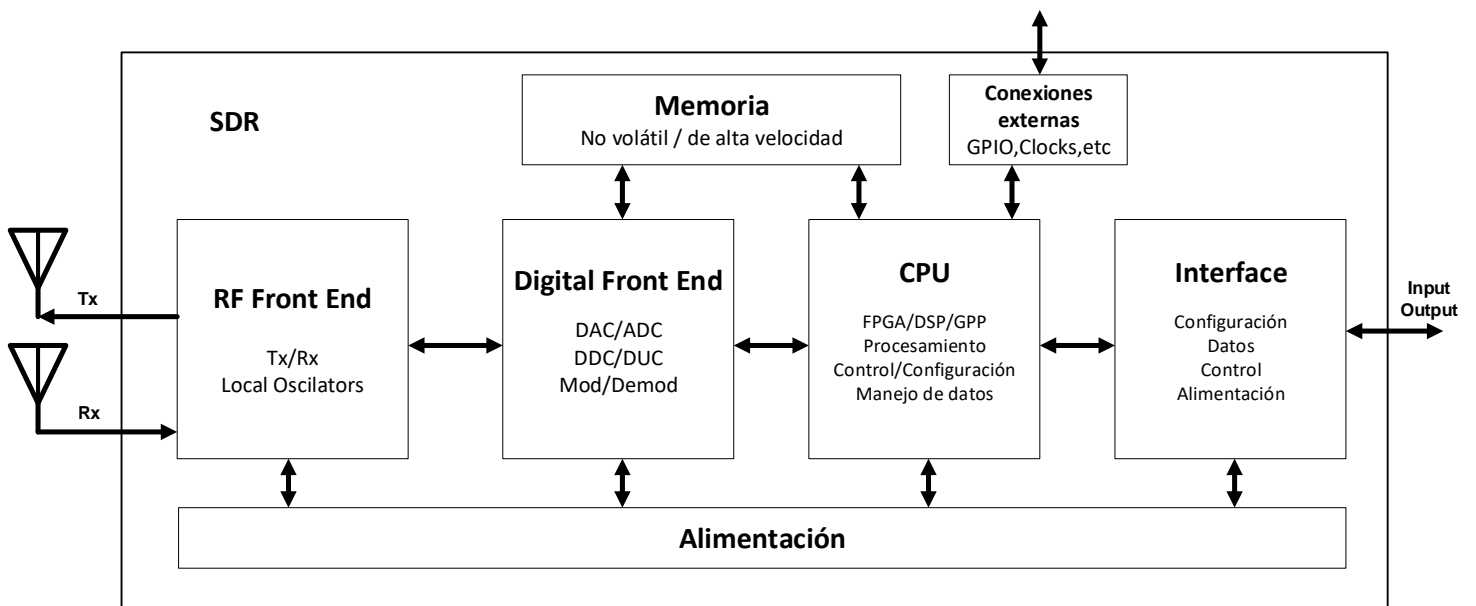
**Agregar sección comparando radio tradicional con SDR. Graficos de las dos y tabla de ventaja y desventaja?**

## Capítulo 2

### 2. Diseño

#### 2.1 Arquitectura general según funciones

La arquitectura de las SDR debe 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.



##### 2.1.1 Antenas

Conjuntos de antenas de transmisión y recepción. Las mismas deben ser capaces de operar en el rango de frecuencia requerido.

##### 2.1.2 RF Front-End

Su función es el acondicionamiento tanto de las señales recibidas como las transmitidas por las antenas correspondientes. En dicho acondicionamiento comprende: filtrado, amplificación, conversión de frecuencia etc.

### 2.1.3 Digital Front-End

La función de este módulo es la conversión analógica-digital en la recepción y digital-analógica en la transmisión.

También es posible realizar procesamientos digitales sobre las señales como por ejemplo:

- Modulación
- Demodulación
- Diezmado
- Interpolación

### 2.1.4 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/des encriptación etc.

### 2.1.5 Memoria

La memoria del sistema puede estar compuesta por memorias de alta velocidad y/o memorias no volátiles.

### 2.1.6 Conexiones externas

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

### 2.1.7 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.

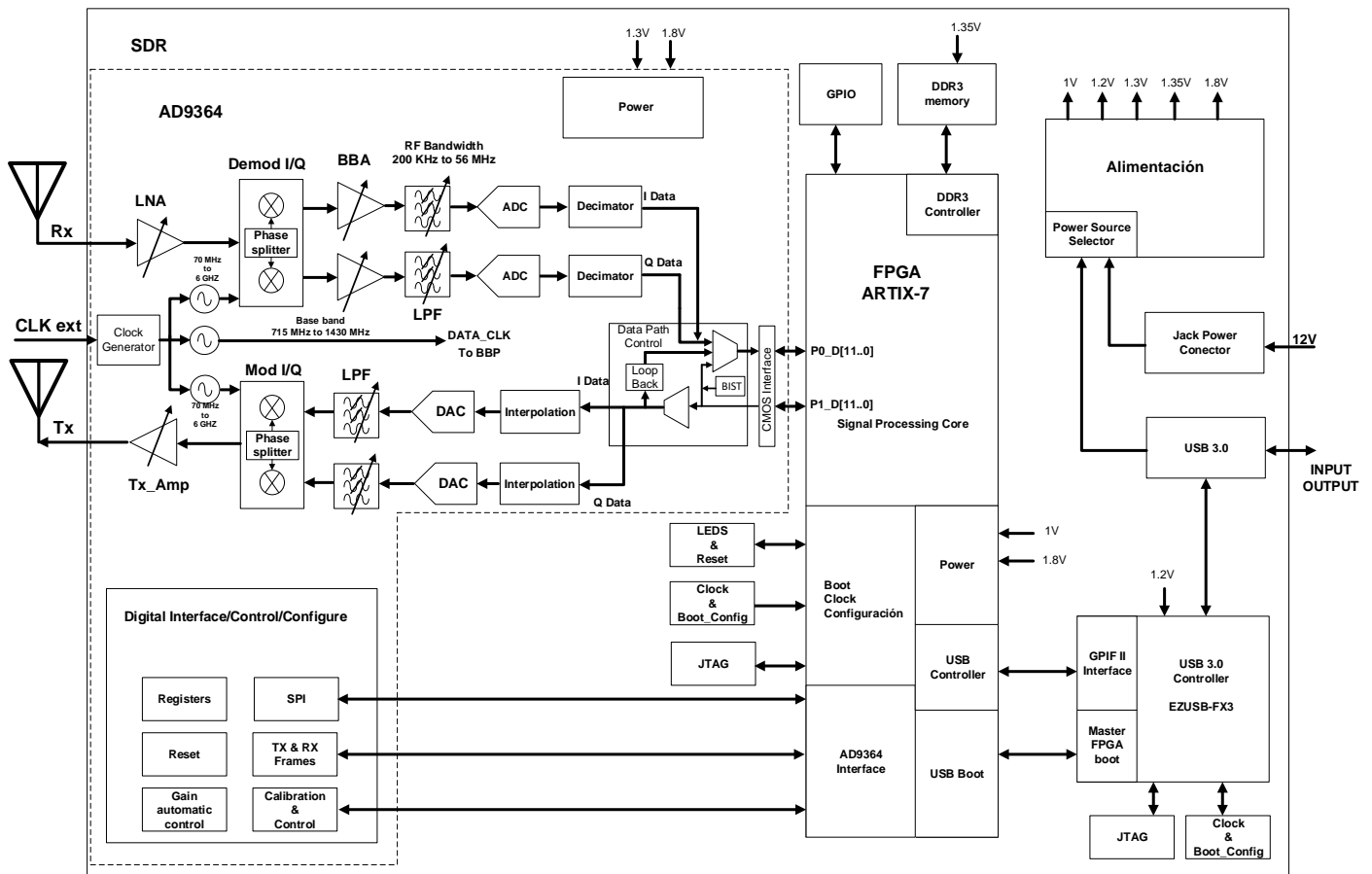


### 2.1.8 Alimentación

Su función es proveer la alimentación a los módulos nombrados anteriormente.

## 2.2 Arquitectura definida

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



A continuación se detallara información sobre cada módulo de esta arquitectura.

### 2.2.1 CPU

A la hora de elegir la CPU algunas de las opciones pueden ser las siguientes:

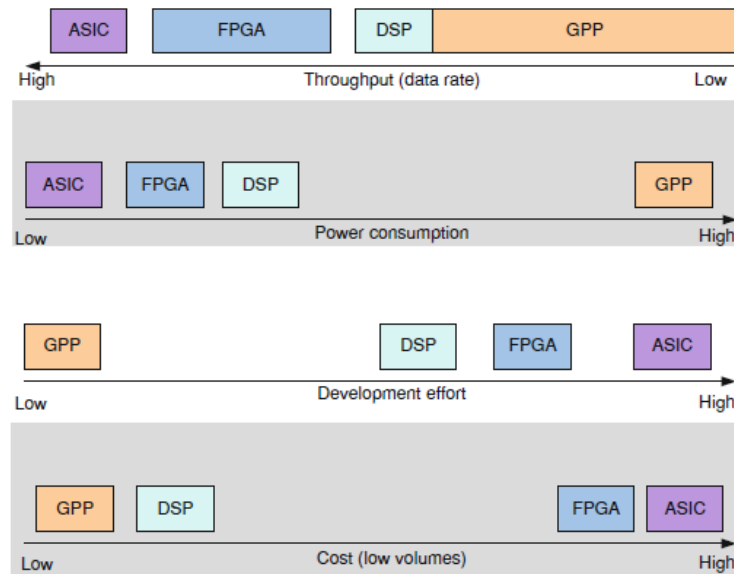
**ASIC:** Es un circuito integrado hecho a la medida para un uso en particular, en vez de ser concebido para propósitos de uso general. Son utilizadas cuando se necesita una alta tasa de datos a un consumo de energía razonable.

**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.

A continuación se mostrara una comparativa entre las diferentes soluciones para la CPU:



Falta comentar imagen

Se decidió utilizar una FPGA como CPU por las siguientes razones:

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

### 2.2.1.1 Selección de FPGA (Falta completar)

Primero se debe seleccionar que proveedor utilizar, en el mercado de las FPGAs existen 2 grandes empresas Xilinx y Altera. Debido a previo conocimiento de las FPGAs de Xilinx se decidió utilizar este proveedor.

Una vez seleccionado el proveedor, se decidió utilizar la serie 7 de Xilinx ya que es la más actual y utilizan el entorno Vivado para ser programadas. 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

[<https://www.xilinx.com/support/documentation/selection-guides/7-series-product-selection-guide.pdf>]

(Pasar el link/nombre como referencia y número correspondiente)

Luego para seleccionar el modelo adecuado de FPGA Artix-7 se tuvo en cuenta las siguientes características:

**Recursos FPGA:** En principio, lo ideal es tener muy bien definido lo que se va a programar en ella, tener implementado todo el código en VHDL. Además, se debe disponer de recursos extras para darle la libertad al usuario de implementar otros procesamientos en la FPGA.

Sin embargo, en el contexto de este proyecto esto no es aplicable, principalmente por falta de: tiempo y dispositivos con características a fines a lo propuesto.

Por lo tanto, para estimar los recursos necesarios de la FPGA se realizó un estudio de las capacidades de las SDR comerciales:

Modelo	FPGA	Programmable Logic Gates
USRP B200mini	Xilinx Spartan-6 XC6SLX75	75k
LimeSDR	Altera Cyclone IV EP4CE40F23	40k
BladeRF xA4	Altera Cyclone V	49K

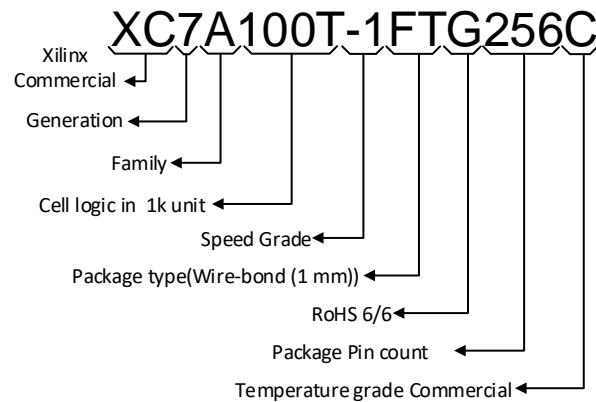
**Temperatura de operación:** Comercial industrial (0° a 85 °C), extended (0° a 100 °C), industrial (-40° a 100 °C) o expanded (-40° a 125 °C)

**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 número del modelo seleccionado.



### 2.2.1.2 Pin Out

Mostrar imagen del pin out?

### 2.2.1.3 Función

## Procesamiento 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 brindarán 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

### 2.2.1.4 Configuración FPGA

Las FPGA son configuradas 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.

### 2.2.1.5 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 (para más información ir al **Anexo A- Alimentación**). 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**

### 2.2.2 RF Front End & Digital Front End

Se decidió utilizar un transceiver integrando ambos Front End en un solo IC, lo que significa una menor cantidad de componentes y superficie. 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
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

Se descartó el transceiver LMS6002 por los siguientes motivos:

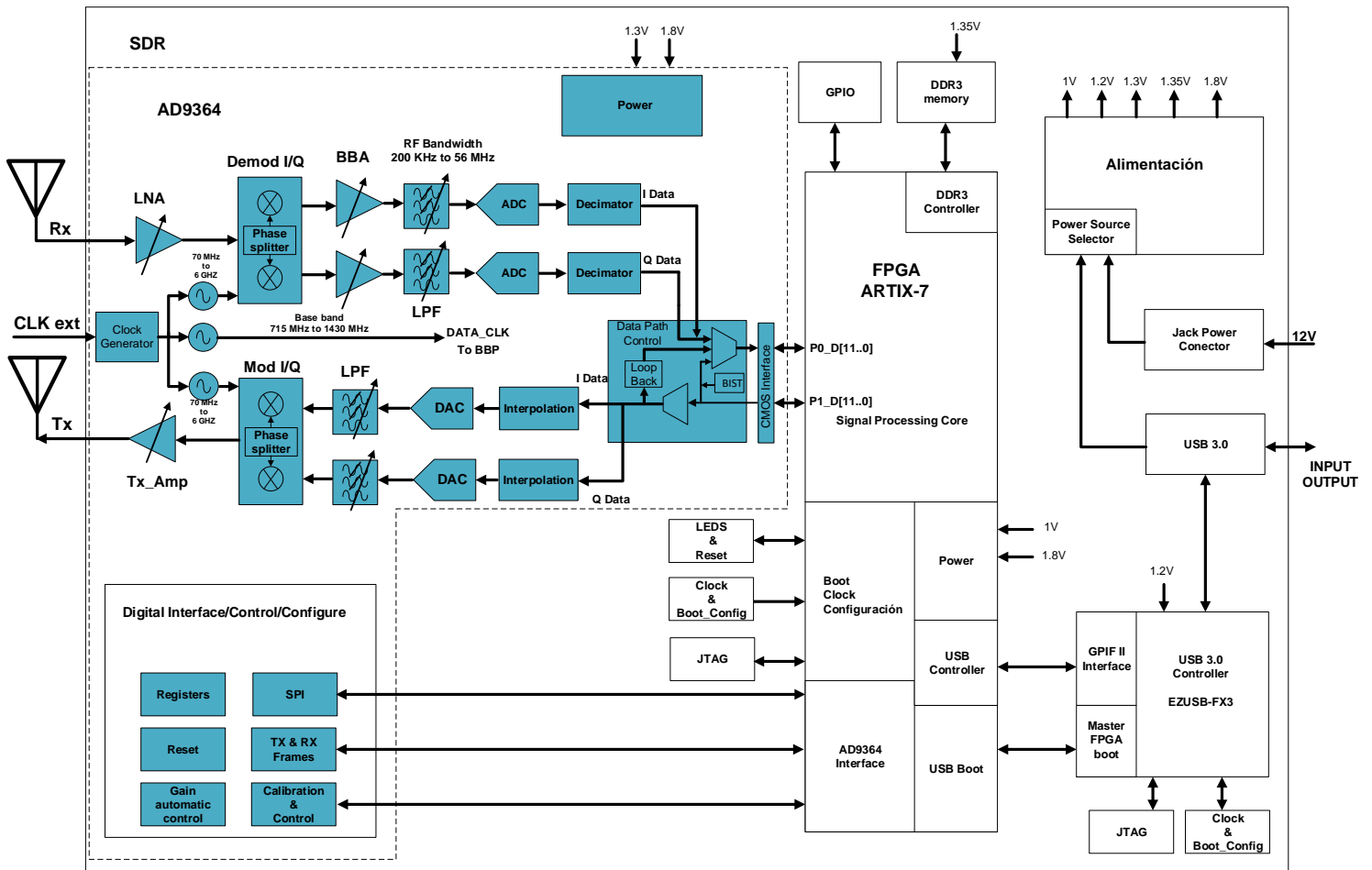
- No cumple con el ancho de banda (requerimiento que podría ser flexible debido a su bajo precio).
- Ya existe una SDR open source que utiliza este transceiver (proporcionan: gerber, esquemáticos y proyecto entero en altium).
- Escases de información en cuanto a funcionamiento y guías de diseño.

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

Este dispositivo combina un RF Front End con un flexible mezclador de señales en banda base y un sintetizador de frecuencia 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:



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 que cumple 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. En particular estos filtros son de tercer orden Butterworth y su frecuencia de corte es programable en el rango de 200 kHz a 39.2 Mhz.

### **Generador de señales BITS**

Este integrado dispone en un generador de señales interno con el cual se puede inyectar 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.

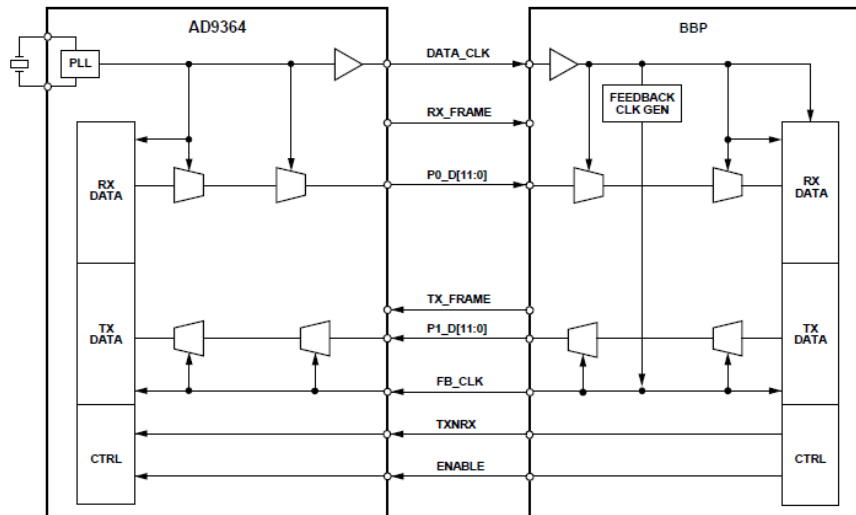
#### 2.2.2.1 Modos de funcionamiento del transceiver

Este transceiver puede ser utilizado en los siguientes modos de operación:

#### Dual port full duplex mode.

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 las conexiones entre el transceiver AD9364 y el BBP (FPGA) para este modo de operación.



Los datos transmitidos en P1\_D[11:0] que van desde el BBP hacia el AD9364 son capturados por medio del FB\_CLK. Mientras que los datos recibidos en P0\_D[11:0] que van desde el AD9364 hacia el BBP son capturados por medio de DATA\_CLK.

Un pulso en el pin de ENABLE (un flanco ascendente) dispara el comienzo de transferencia de datos y otro pulso (un flanco descendente) da fin a la transferencia de datos.

Las señales Rx\_FRAME y Tx\_FRAME (con un flanco ascendente) indican el comienzo de un conjunto de muestras (frames).

Para los datos de Tx, P1\_D[11] es el bit más significativo y P1\_D[0] es el bit menos significativo. Para los datos de Rx, P0\_D[11] es el bit más significativo y P0\_D[0] es el bit menos significativo.

Los datos I/Q son transportados en el mismo bus en cada dirección. Para un solo camino de RF en cada dirección los datos son transportados de la siguiente manera:

Datos de Tx o Rx : I , Q , I , Q , .....

A continuación se mostrarán unos diagramas de tiempo de las distintas señales tanto para la transmisión como la recepción, en el modo de operación Full Duplex.



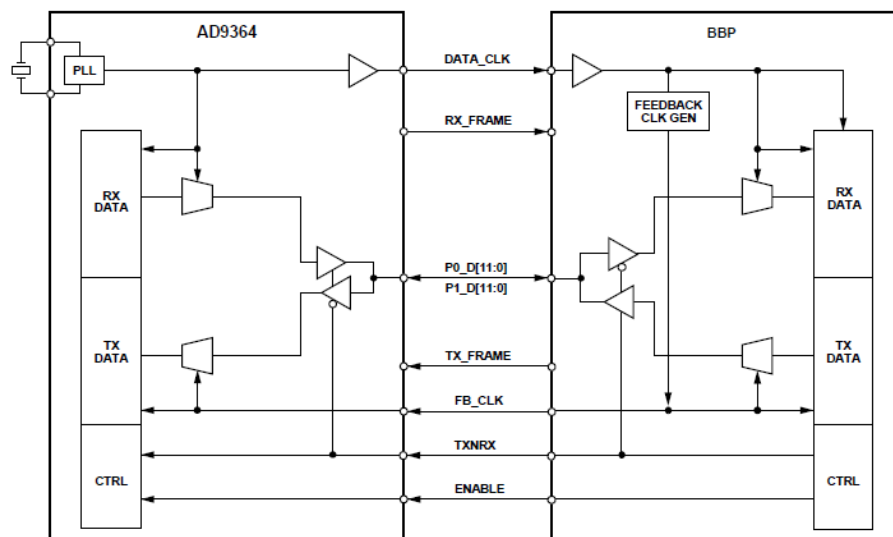
Falta comentar imagen



Falta comentar imagen

### Dual port half duplex mode.

El modo de operación Dual Port Half Duplex es utilizado en aplicaciones que requieren operaciones TDD (time-division duplex) y una tasa de datos hasta 122.88 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 ilustra las conexiones entre el AD9364 y el BBP en este modo de operación.



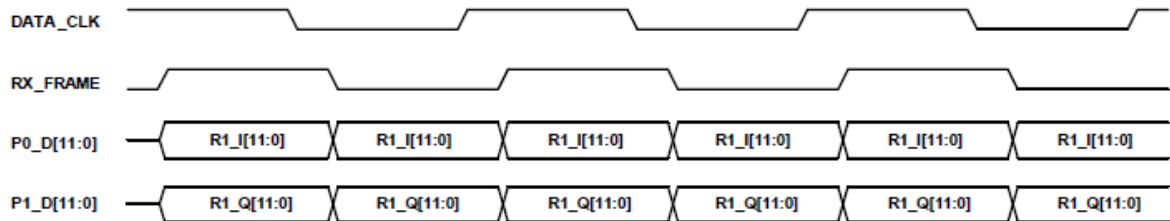
Durante una ráfaga de recepción, los datos suministrados por el AD9364 hacia el BBP son capturados por medio del DATA\_CLK. La transferencia de datos empieza cuando una señal de pulso alto aparece en el pin de ENABLE y la misma termina cuando el pulso vuelve a bajo.

La dirección de la transferencia de datos es determinada por la señal TXNRX. Cuando esta señal es baja el AD9364 está en modo de recepción, configurándose el bus en la dirección de recibir, los datos son transferidos desde el AD9364 hacia el BBP. Cuando la señal de TXNRX es alta, la dirección de transmisión del bus es cambiada, los datos son transferidos desde el BBP hacia el AD9364. Durante una ráfaga de transmisión los datos suministrados desde el BBP hacia el AD9364 son capturados por medio de FB\_CLK. Las señales Rx\_FRAME y Tx\_FRAME indican el comienzo de un conjunto de muestras.

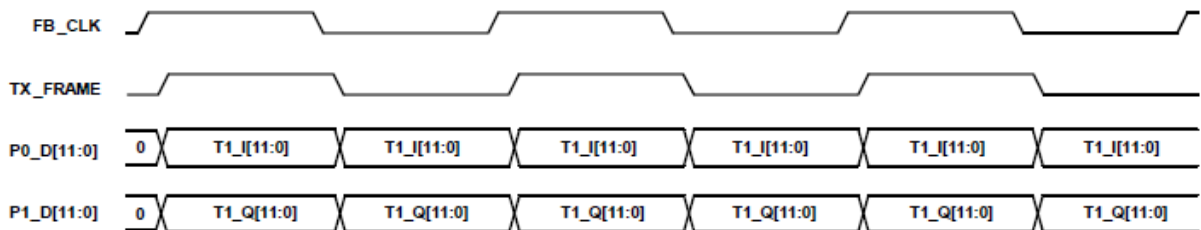
## 2. Diseño

Para las líneas de datos D[11] es el bit más significativo mientras que D[0] es el bit menos significativo. Las muestras son separadas de la siguiente manera: los datos I en el puerto P0\_D[11:0] y los datos Q en el puerto P1\_D[11:0].

A continuación se mostrarán unos diagramas de tiempo de las distintas señales tanto para la transmisión como la recepción para el modo de operación half duplex.



Falta comentar imagen



Falta comentar imagen

### 2.2.3 USB 3.0

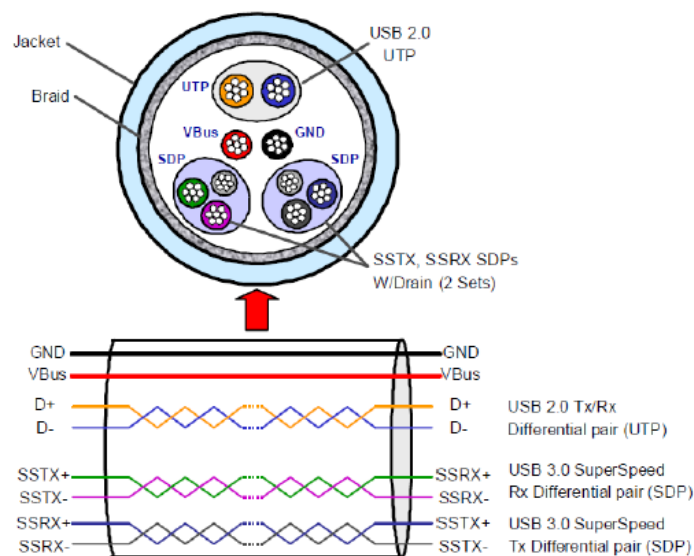
(Falta completar función, selección del mismo)

#### 2.2.3.1 Información general USB 3.0

Las especificaciones del USB 3.0 permite una tasa de transferencia máxima de 5 Gbps (SuperSpeed), 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 esta mallados y separados. Los cables USB 3.0 incluyen a los utilizados en el USB 2.0 como se muestra en la siguiente figura.



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

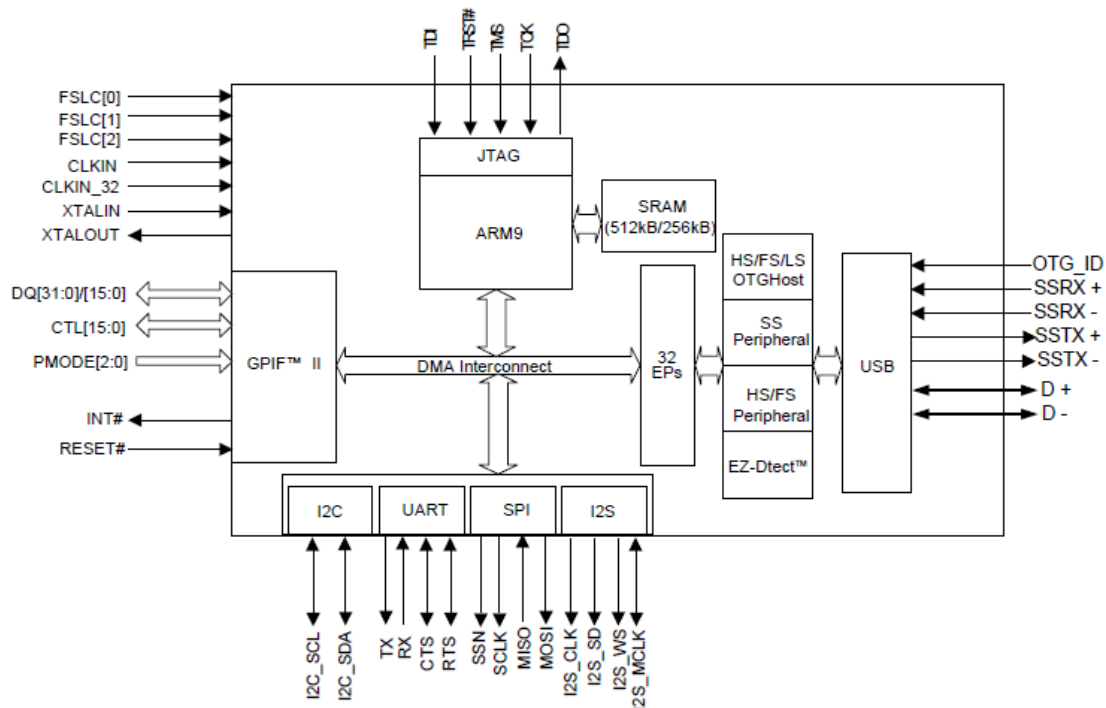
La capa de protocolo maneja las reglas de comunicación entre el host y el dispositivo.

### 2.2.3.2 Controlador USB 3.0 CYUSB3014

El dispositivo que se decidió 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:



## 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 JTAG debugger 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

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

Como fuente de clock se utilizó un cristal de 19.2 MHz (FSLC[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.

Faltaria hablar sobre el modo de funcionamiento "FIFO SLAVE"??

### 2.2.4 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 una futura simulación casi completa de la sección de alimentación.

Para alimentar la mayor parte del circuito 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

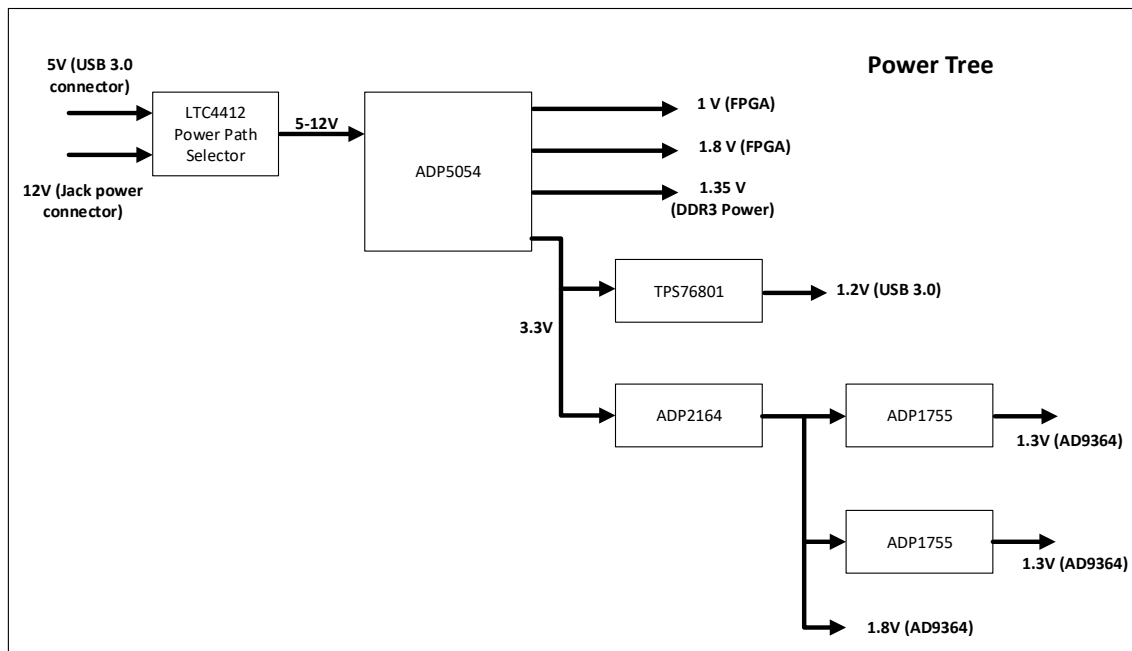
En donde:

- 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 A**.
- Para la estimación del transceiver se utilizó el peor caso de consumo provisto por la hoja de datos del mismo. En este caso la transferencia 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).
- Para el USB3.0 se utilizó una tabla de consumos típicos que provee cypress (ver referencia <https://www.cypress.com/file/145131/download> ).
- Para obtener el consumo de la memoria DDR3 se utilizó una hoja de cálculo que provee micron y se tomaron las corrientes máximas de la misma (ver referencia [https://www.micron.com/-/media/client/global/documents/products/power-calculator/ddr3\\_ddr3l\\_power\\_calc.xlsm?la=en](https://www.micron.com/-/media/client/global/documents/products/power-calculator/ddr3_ddr3l_power_calc.xlsm?la=en) ).

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

A partir de todas las estimaciones y consideraciones enunciadas anteriormente se diseñó la siguiente 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.

### 2.2.5 Memoria DDR3 (Falta completar)

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 un solo chip de memoria con una capacidad total de 256MB .El modelo del chip es el MT41J128M16JT-125 fabricado por Micron.

Cuenta con 14 líneas de dirección (A[13..0]), 3 líneas para selección de banco (BA[2..0]), 16 líneas de datos [15..0].

Además para esta memoria, la herramienta de programación Vivado, provee controladores para la misma. Ahorrando tiempo de programación en el futuro.

**Hablar sobre velocidad de clock máxima,minima , código del modelo**

**Encapsulado y pines?**

### 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 DEL GPIO

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.

### 2.2.7 SDR Características

Resumiendo la SDR LAPAC 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

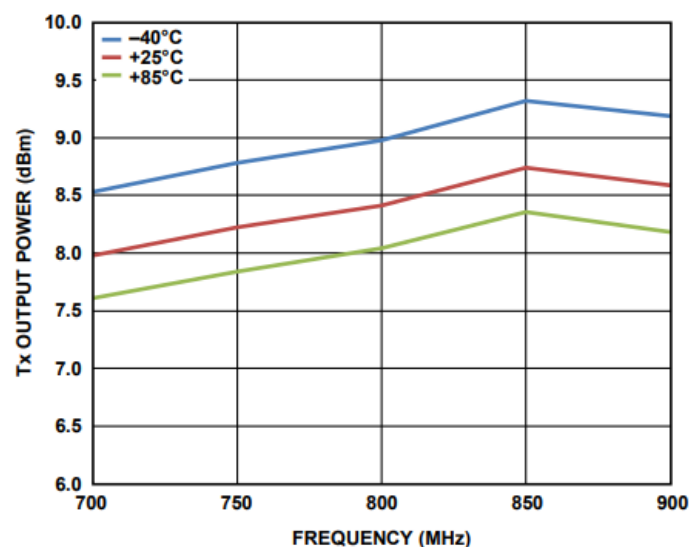
### Hardware:

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

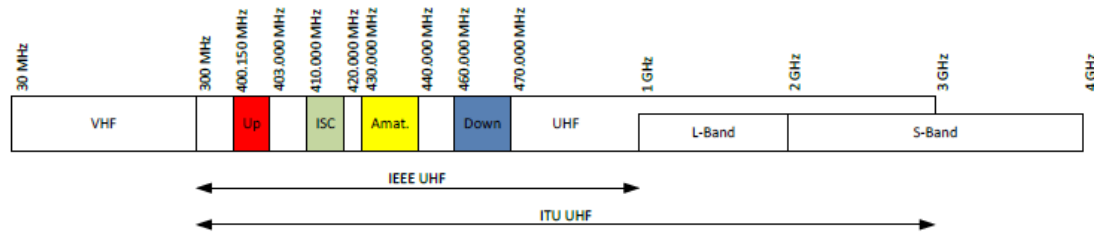
### Que tan lejos?:

Por lo general tanto en esta SDR como en la mayoría, no es posible determinar que tal 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.



Faltaría comentar algo sobre rango de frecuencias, como por ejemplo:



## Capítulo 3

### 3. Implementación del Diseño

#### 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. Estas

conexiones se realizan por medio de puertos los cuales pueden ser unidireccionales o bidireccionales.

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 (falta completar)

Para una mejor organización del proyecto a cada componente se le agrego información extra en sus propiedades dentro del esquemático. La siguiente figura muestras 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	LF CSP-48	STRING
<input type="checkbox"/>	Manufacturer	Analog Devices	STRING
<input type="checkbox"/>	LinkVendedor2	<a href="https://www.digikey.com/product-detail/A">https://www.digikey.com/product-detail/A</a>	STRING
<input type="checkbox"/>	LinkVendedor1	<a href="https://ar.mouser.com/ProductDetail/A">https://ar.mouser.com/ProductDetail/A</a>	STRING
<input type="checkbox"/>	Descripción	Reguladores de voltaje, SMD, 48 pins	STRING
<input type="checkbox"/>	DataSheet	<a href="http://www.mouser.com/ds/2/609/ADP5054ACPZ-R7">http://www.mouser.com/ds/2/609/ADP5054ACPZ-R7</a>	STRING
<input type="checkbox"/>	CostoVendedor2	10.76	STRING
<input type="checkbox"/>	CostoVendedor1	9.88	STRING

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

Hablar sobre la unificación de librerías de todos los componentes en una sola

### 3.2 Conexiones

Las conexiones de cada dispositivo fue realizada según la indicación de diferentes data sheets y guías de diseño, las cuales se encuentran en el anexo [Conexiones componentes]

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 ADP505 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:

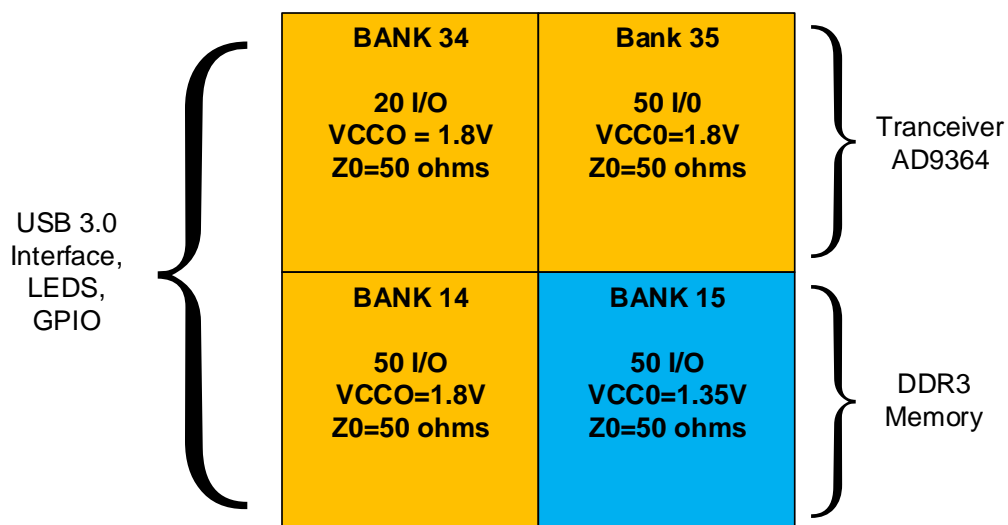




Figura asignación de bancos FPGA.

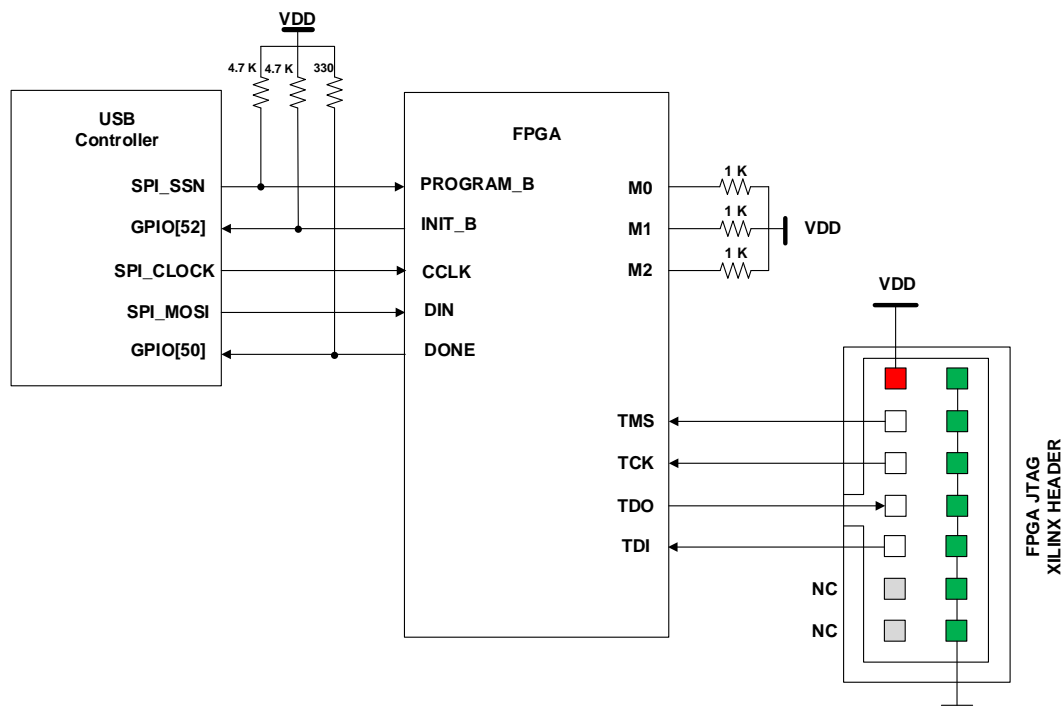
Además cada uno de estos bancos tiene pines que sirven para determinadas funciones como por ejemplo los pines MRCC. En estos pines se conectarán la mayoría de las señales de clock, ya que por medio de estos pines se puede distribuir dichas señales a múltiples regiones de clock dentro de la FPGA.

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).

#### 3.2.1.2 Booting

La configuración de la FPGA se realiza por medio controlador de USB 3.0, lo cual elimina la necesidad de una memoria para programar la FPGA. Actuando como master este controlador 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:



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 seria 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 pines.

Tablita de los diferentes valores de m[2:0] y algo sobre el Jtag

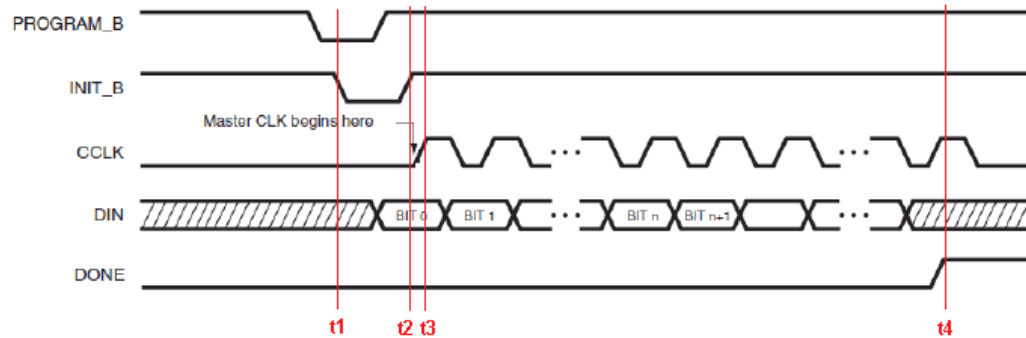
Configuration Mode	M[2:0]	Bus Width	CCLK Direction
Master Serial	000	x1	Output
Master SPI	001	x1, x2, x4	Output
Master BPI	010	x8, x16	Output
Master SelectMAP	100	x8, x16	Output
JTAG	101	x1	Not Applicable
Slave SelectMAP	110	x8, x16, x32 <sup>(1)</sup>	Input
Slave Serial <sup>(2)</sup>	111	x1	Input

Falta cambiar imagen a tabla

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.



#### 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 los componentes del regulador ADP5054.

La mayoría de estos ítems están detallados en el Anexo A- Alimentación.

#### 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 PCB información básica

#### 3.3.1 PCB definición

Un PCB es usado para crear conexiones entre componentes, tales como resistencias, circuitos integrados y conectores. Los mismos están compuesto de una capa de dieléctrico y delgadas hojas de cobre añadidas en ambos lados de una placa. Esta placa (dieléctrico) puede ser flexibles o rígidas.

Un PCB multicapa (una placa con dos o más capas de cobre) puede estar compuesto por:

- Core
- Prepeg
- Láminas de Cobre

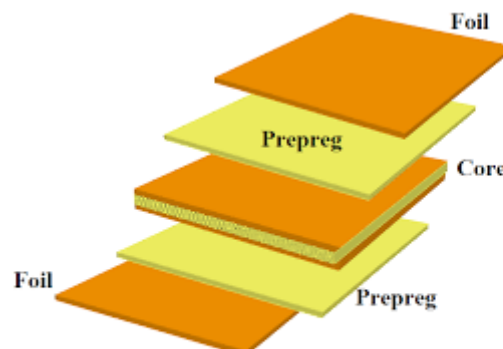
El Prepeg es una placa dieléctrica con pegamento en ambas caras.

El Core es una placa de material dieléctrico con una lámina de cobre en cada cara.

El laminado de cobre (copper foil), es una delgada hoja de cobre la cual se coloca en/o entre el material de pre-preg y lo une en el adhesivo que forma parte del pre-preg. Los grosores más comunes de estas láminas son los siguientes:

- 1/2 oz (0.0007"[.01778mm])
- 1 oz (.0014"[0.03556mm])
- 2 oz(.0028"[0.07112mm])

La siguiente figura muestra cómo puede estar compuesto un PCB multicapa



### 3.3.2 Mascara de soldadura (Solder Mask)

La máscara de soldadura es un material que se utiliza para proteger a la placa de:

- Los alrededores del ambiente
- Aislar eléctricamente la misma
- Proteger los puntos de soldadura

### 3.3.3 Mascara de Leyendas (Silk Mask)

La silk mask es una capa de pintura que se utiliza para indicar referencias, como por ejemplo: polaridad, posición y datos de los componentes.

Por lo general se utiliza el color blanco, amarillo, negro y rojo.

### 3.3.4 Pistas

La función de estas es conectar eléctricamente un punto con otro. Estas son formadas removiendo el cobre no necesario y dejando solamente los trazos y pads requeridos.

Las características que definen a una pista son el ancho, el largo y el grosor de la misma. Con estas características se pueden determinar la corriente que puede circular por la misma. Además el grosor y el ancho son parámetros que determinara la impedancia de la misma (también depende del stack-up y las constantes dieléctricas como se mostrara más adelante).

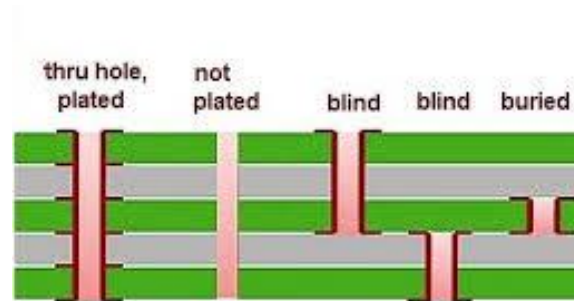
### 3.3.5 Pads

El pad es una región de cobre encargada de hacer contacto entre el PCB y el componente

### 3.3.6 Perforación Platinada (Plated Hole)

Una perforación platinada, como se muestra en la siguiente figura, consiste en un pad con una perforación en el centro. Las paredes de esta perforación son

platinadas con cobre. El platinado del agujero comprende desde la superficie del pad y continúa a través de la perforación hasta llegar al otro extremo de la placa.



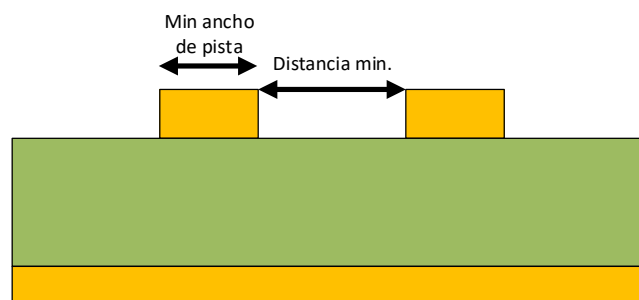
Este tipo de agujeros son las vías más comunes ya que son simples de fabricar y tienen un costo bajo en comparación con las “blind Via Hole” o “Buried Via Holes”. Son utilizadas para conectar las diferentes capas en un PCB multicapa.

#### 3.4 Capacidades del fabricante PCB

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

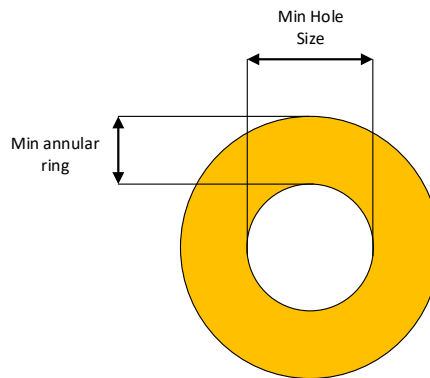
- Mínimo ancho de pista y espaciado

Este requerimiento estipula la distancia mínima entre dos pistas y el ancho de las mismas. Dicha distancia mínima es de 3 mils (0.076 mm) para ambos requerimientos.

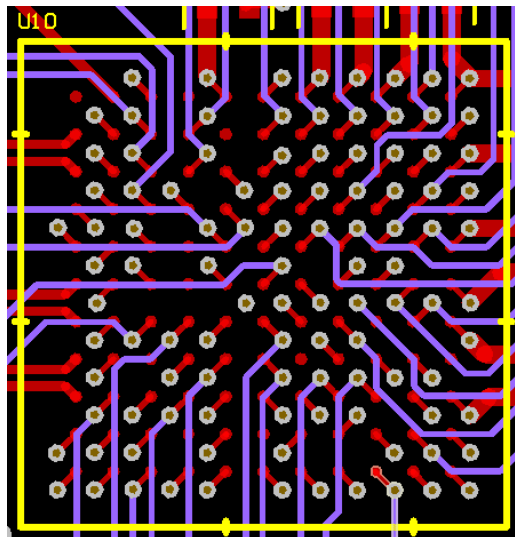


- Mínimo tamaño de agujero y annular ring

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



Estos requerimientos son esenciales para poder determinar si va ser posible realizar el “Fanout” (llevar cada uno de pines 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.

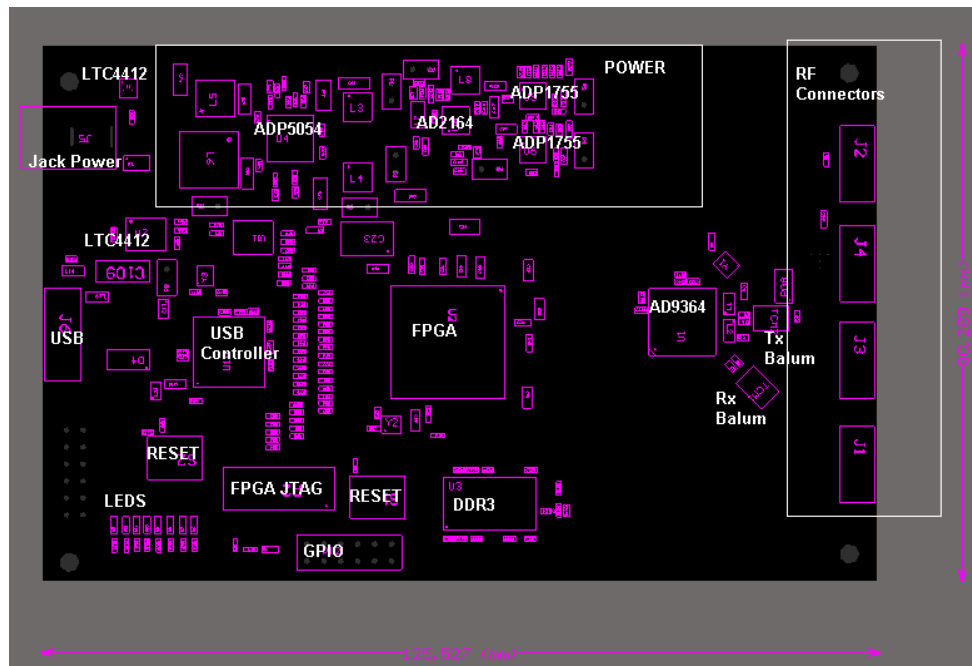


Para más información sobre todas las capacidades del fabricante ir al anexo..

#### 3.5 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 Figura TATATA muestra una de las capas mecánicas 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



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.

#### 3.6 Stackup

Como se mencionó en la sección 3.3.1, 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. Además también 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 rutear 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:

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

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 ruteara 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

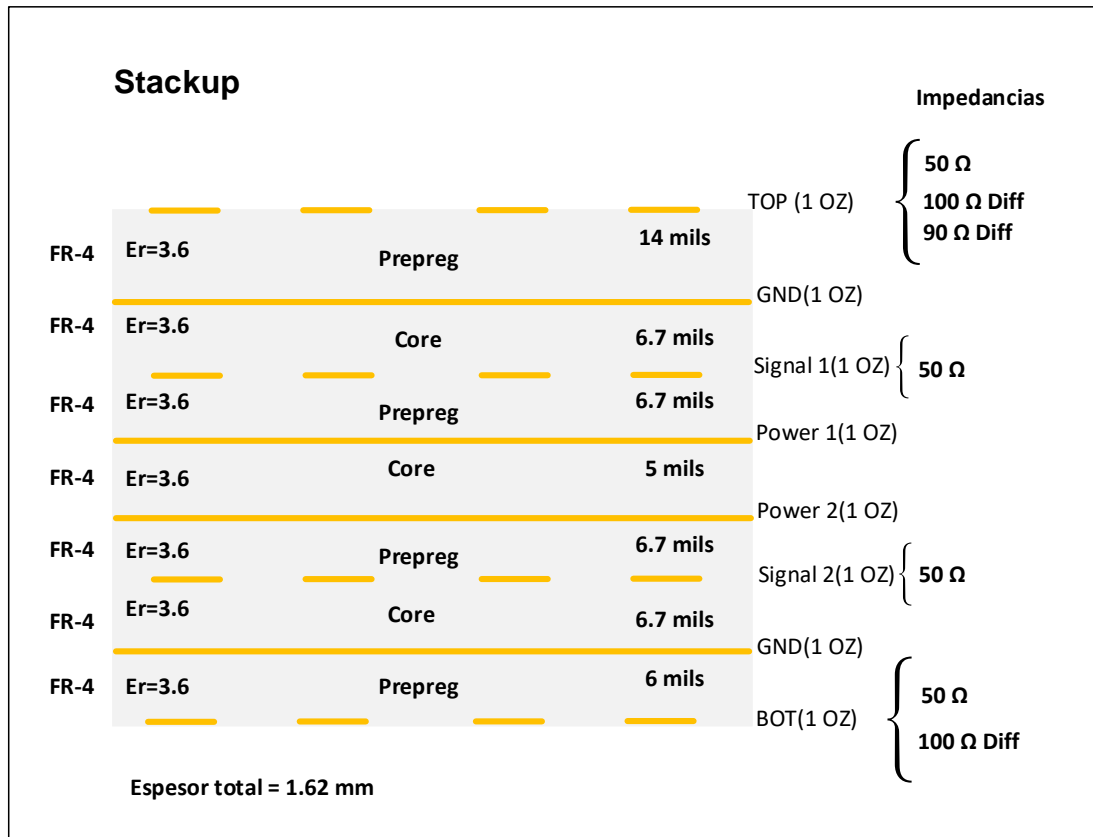
$$Layers = \frac{176}{60 \times 1} = 3 \text{ (redondeado)}$$

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
<b>AD9364</b>	Diferencial	100 Ω	Top
			Bot
	Individual	50 Ω	Top
			Bot
			Signal 1
			Signal2
<b>USB 3.0</b>	Diferencial	90 Ω	Top
			Bot
	Individual	50 Ω	Top
			Bot
			Signal 1
			Signal 2
<b>FPGA</b>	Diferencial	100 Ω	Top
			Bot
	Individual	50 Ω	Top
			Bot
			Signal 1
			Signal 2
<b>Memory</b>	Diferencial	100 Ω	Top
			Bot
	Individual	50 Ω	Top
			Bot
			Signal 1
			Signal 2

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



Para determinar 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.

#### 3.7 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 dependiendo de la relación entre su frecuencia o longitud de onda y la distancia que esta señal recorre (cuando 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.

#### 3.8 Impedancia característica, reflexión y terminación

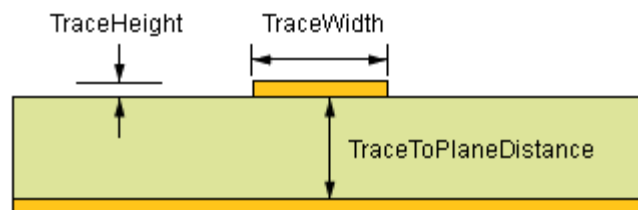
Una de las propiedades de una línea de transmisión es su impedancia característica,  $Z_0$ . Cuando se diseña una línea de transmisión entre un emisor y un receptor, las impedancias: de salida del receptor, de la línea y de entrada

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.



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

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

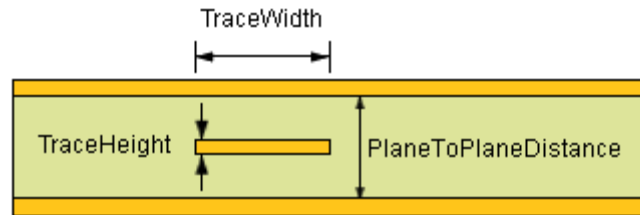
En donde:

H = Distancia entre pista y plano

W = Ancho de pista

T= Grosor de pista

Er= Constante dieléctrica del material dieléctrico.



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

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

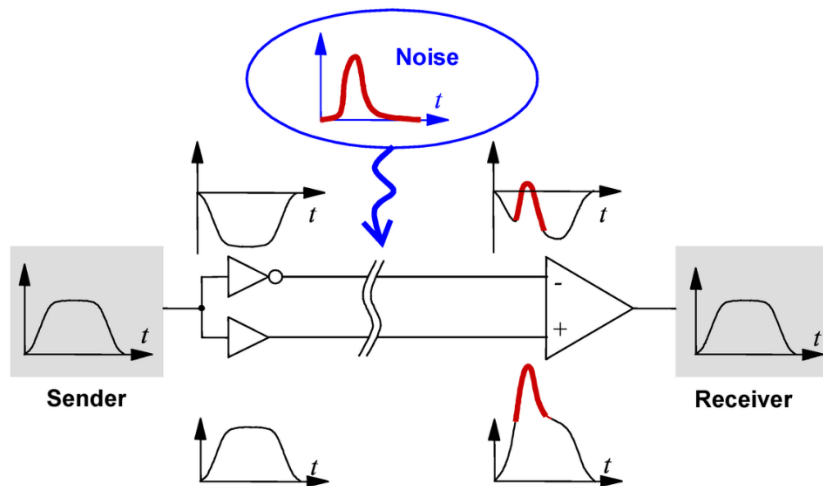
En donde:

B = Distancia entre planos

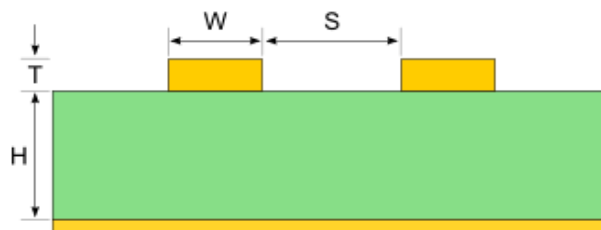
#### 3.9 Líneas diferenciales

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.



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:



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

### 3.10 Ajuste de longitudes

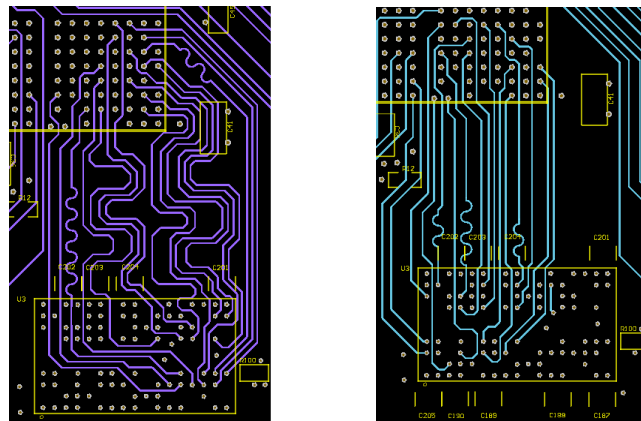
Cuando varias señales de alta velocidad pertenecen a un mismo grupo o bus de datos que funcionan de forma síncrona, es importante que todas tengan la misma distancia eléctrica, es decir, que todas las señales eléctricas lleguen dentro de un pequeño intervalo de tiempo que garantice el correcto funcionamiento 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, adres 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.



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		

Comentar los requerimientos por parte del fabricante de la memoria Micro en cuanto long etc

### 3.11 Algunas Consideraciones

#### 3.11.1 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

#### 3.11.2 Power

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

#### 3.11.3 USB 3.0

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

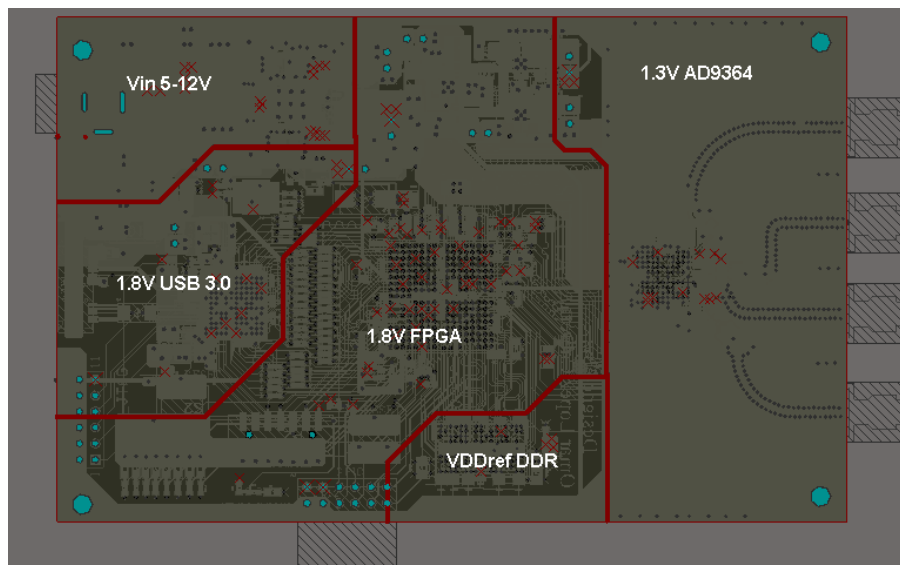
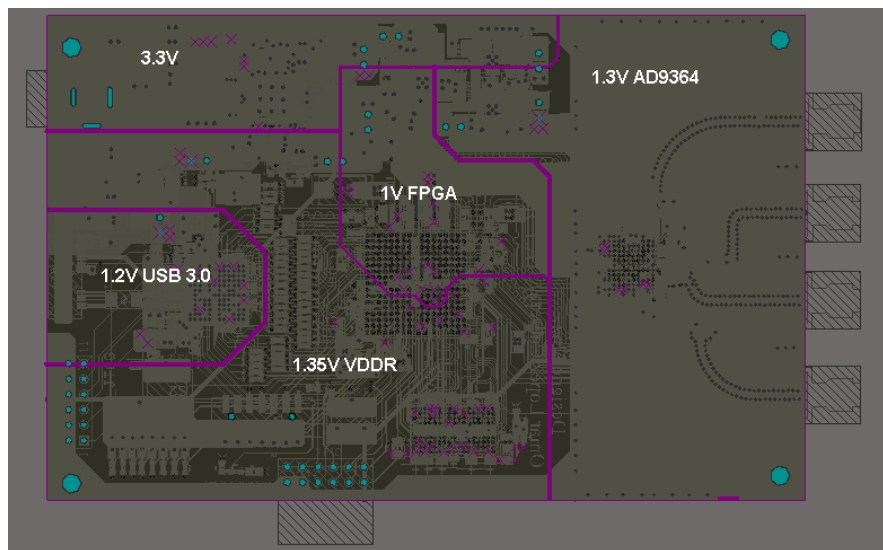
#### 3.11.4 AD9364

- Criterio de cero diferencia de longitud entre las líneas de interface digital con la FPGA.
- Coplanar Waveguide y ajuste de impedancia 50ohms para lograr un ancho similar al conector de RF y no tener desadaptación de impedancia
- Conexiones de Tx y Rx
- Pista en el top para posible shielding

### 3.12 Planos de alimentación y tierra

Como se ha explicado en la Sección 4.1, de las 8 capas con que cuenta la PCB, 2 son planos de tierra y 2 son para planos de alimentación. Las siguientes figuras mostraran como están distribuidas las diferentes tensiones y algunas particularidades en los planos de GND.





Una de las recomendaciones principales para los planos de tierra es que sean superficies de cobre ininterrumpidas que cubran toda la placa. En las siguientes figuras se muestran dichos planos sólidos.

Falta figura planos solidos GND

#### 3.13 Capacitores desacople

Los capacitores de desacople funciona 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.

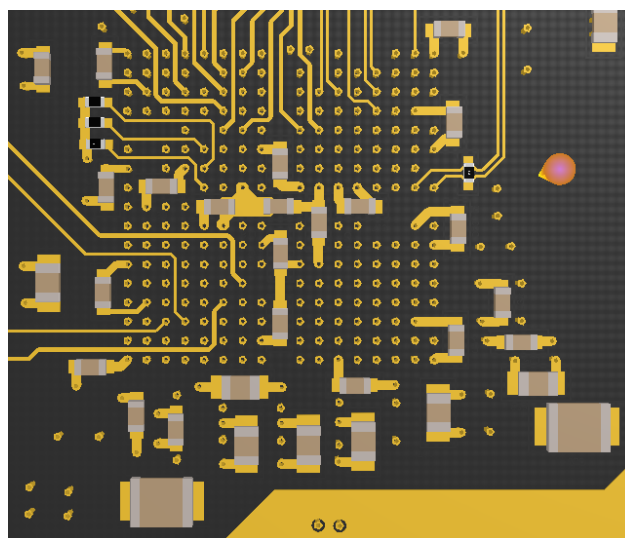
Todos los componentes cuentan con sus condensadores de desacoplo. En particular tanto en el controlador del USB 3.0 y el transceiver AD9364 los capacitores de desacople fueron seleccionados y colocado como en sus placas de desarrollo (por recomendación del fabricante).

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.

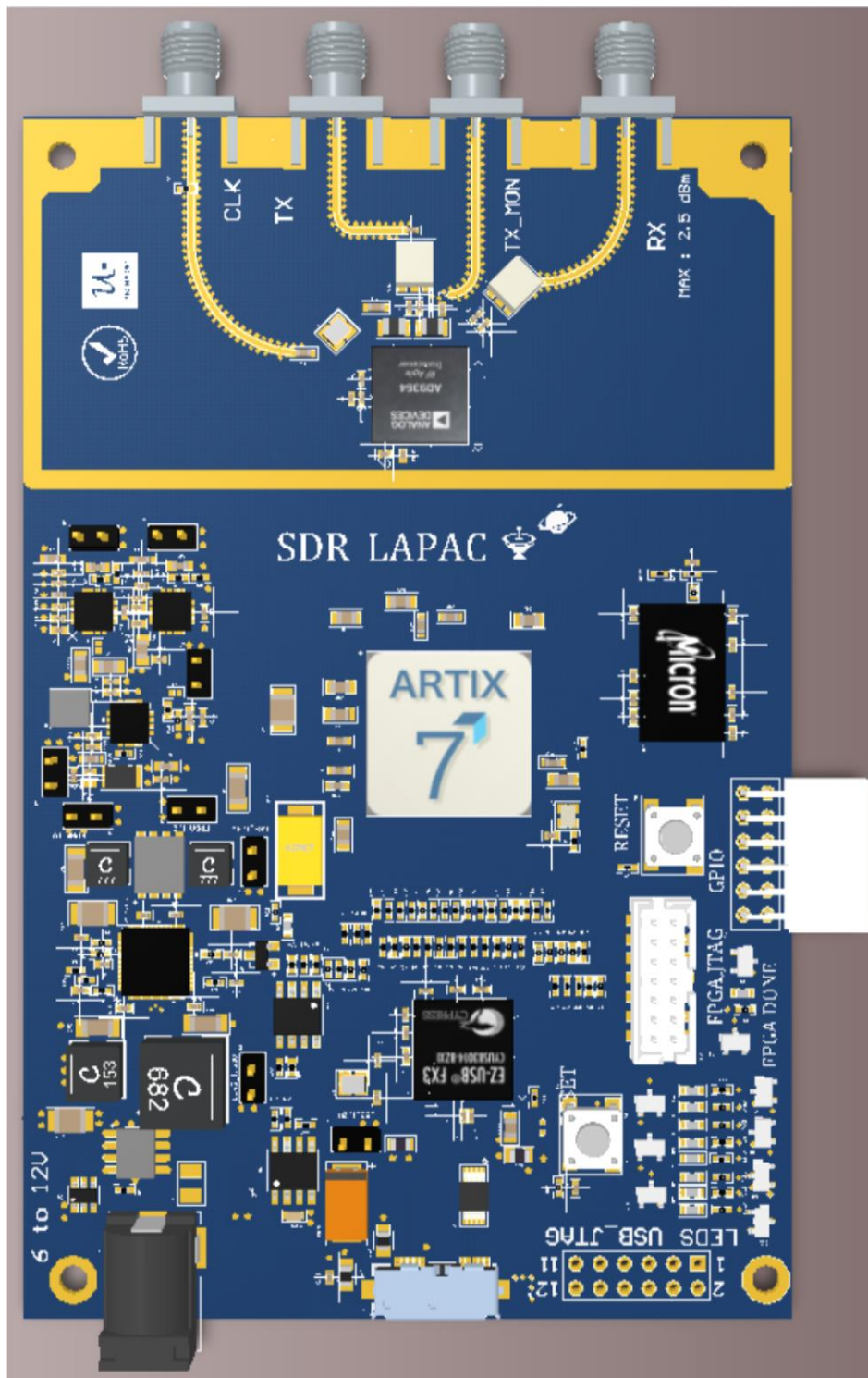
Lo más importante es que cada uno rango de valores de capacitancia son útiles en unos márgenes de frecuencia, influyendo a la distancia a la que se deben colocar de la FPGA. Por ejemplo los capacitores grandes actúan cuando el regulador de voltaje deja de funcionar en el rango de frecuencia baja.

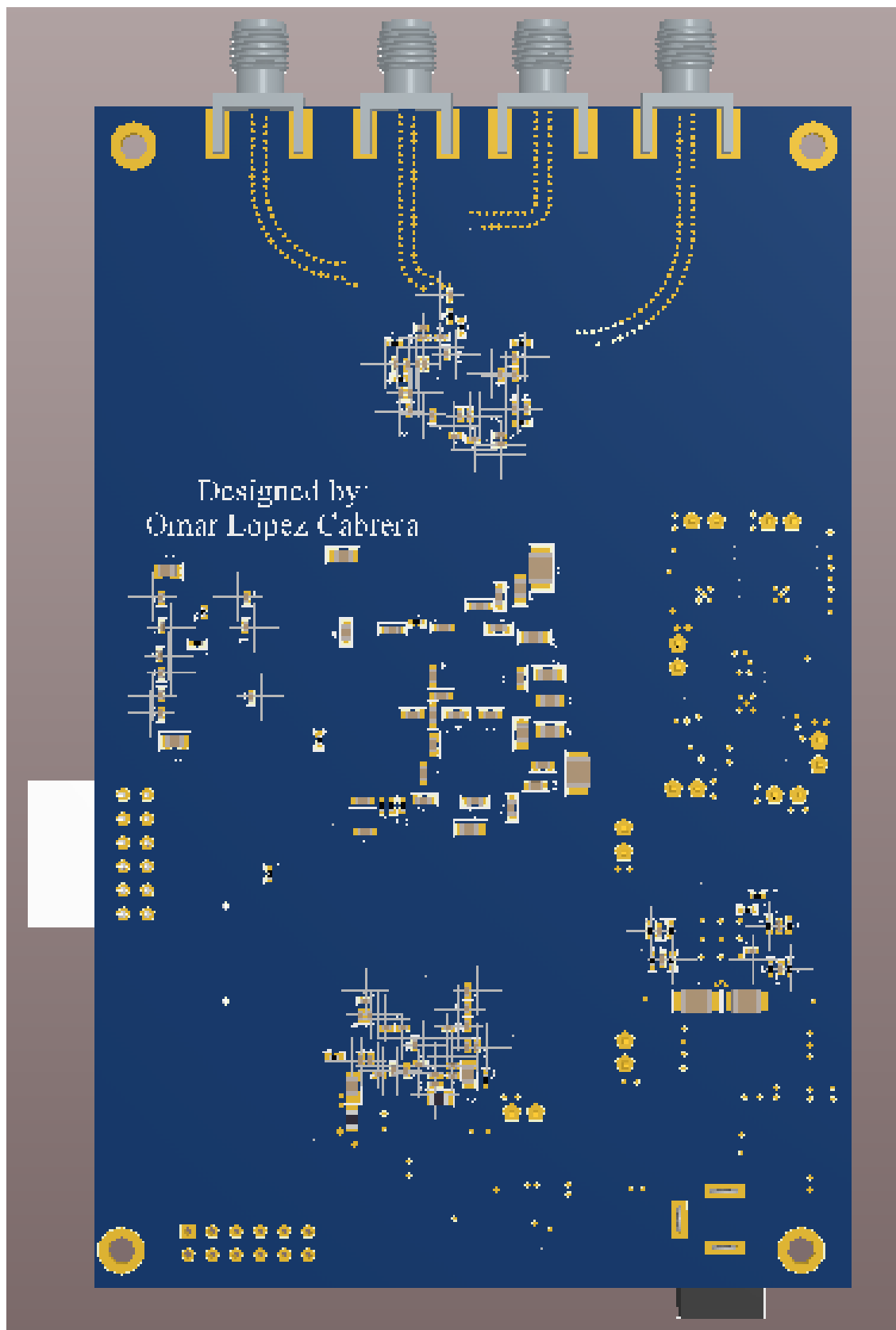
Los capacitores grandes pueden ser ubicados en cualquier parte en los alrededores de la FPGA en cambio los pequeños deben ser colocados lo más cerca posible de los pines de alimentación. Esto es complicado para los integrados que tienen capsulado BGA pues hay una gran cantidad de vías justo debajo del 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.



#### 3.14 Modelo 3d





## Capítulo 4

### 4. Sumario (Falta completar)

#### 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 de la PCB

Una vez que terminado las simulación y requerimientos por parte del fabricante son cumplidas 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 stencil (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 Implementación en VHDL y drivers

Un primer paso importante sería poder contar con algunos de los algoritmos implementados en VHDL, para así poder estimar con más precisión los requisitos que la FPGA debe cumplir en cuanto a recursos lógicos.

Falta comentar drivers, librerías , gnu radio etc

### 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.
- Amplificador en la parte de tx rf
- **Baluns 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 para cada rango de frecuencia. 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.

Falta Gráfico de posible conexión y baluns

### 4.5 Conclusiones

## Anexos

### A-Cuestiones generales del diseño

#### Alimentación

#### 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 síncronico, 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.

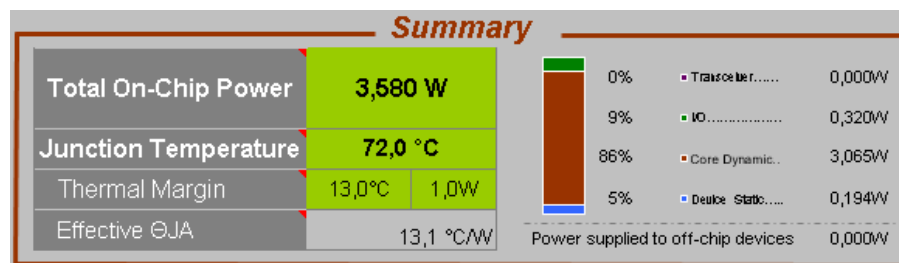
- Para señales no periódicas, el toggle rates no es fácil de predecir. Un método efectivo para estimar el promedio de tasa de cambio para un determinado diseño, consiste en dividir al mismo en secciones por funcionalidad. Luego calcular el toggle rate en cada sección y obtener un promedio de todos. En 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)
- DDR3l external memory de 16 bits de datos (memoria usada).
- Temperatura ambiente de 25 °C (enviroment 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:

<b>Power Supply</b>		
<b>Source</b>	<b>Voltage</b>	<b>Total (A)</b>
V <sub>CCINT</sub>	1,000	2,899
V <sub>CCBRAM</sub>	1,000	0,020
V <sub>CCaux</sub>	1,800	0,254
V <sub>CCaux_ID</sub>		
V <sub>CC0</sub> 3.3V	3,300	
V <sub>CC0</sub> 2.5V	2,500	
V <sub>CC0</sub> 1.8V	1,800	
V <sub>CC0</sub> 1.5V	1,500	
V <sub>CC0</sub> 1.35V	1,350	0,125
V <sub>CC0</sub> 1.2V	1,200	
V <sub>CCADC</sub>	1,800	0,020

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

[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.

Basic Settings ADP505x

**ANALOG DEVICES**

IC Selected: ADP5052

Part Select: ADP5054

Features: 4 Buck+ +

Ambient Temp (max): 55 Deg C

Design Criteria: Most Efficient

**Chan 1 (Buck)**

Channel 1 & 2 in Parallel: ☐

Use Chan 1: ☒

Source: Same as Rail 1

Vin(min): 5 V

Vin(max): 12 V

Vout1: 1 V

Iout1: 3 A

Approx. Total Iout1: 3 A

**Chan 2 (Buck)**

Use Chan 2: ☒

Source: Same as Rail 1

Vin(min): 5 V

Vin(max): 12 V

Vout2: 1.8 V

Iout2: 2 A

Approx. Total Iout2: 2 A

**Chan 3 (Buck)**

Channel 3 & 4 in Parallel: ☐

Use Chan 3: ☒

Source: Same as Rail 1

Vin(min): 5 V

Vin(max): 12 V

Vout3: 3.3 V

Iout3: 1 A

Approx. Total Iout3: 1 A

**Chan 4 (Buck)**

Use Chan 4: ☒

Source: Same as Rail 1

Vin(min): 5 V

Vin(max): 12 V

Vout4: 1.35 V

Iout4: 1 A

Approx. Total Iout4: 1 A

**Chan 5 (LDO)**

Use Chan 5: ☐

Source: External

Vin(min): 2 V

Vin(max): 2 V

Vout5: 1 V

Iout5: 0 A

Approx. Total Iout5: 0 A

Buttons: Submit/Run, Advanced Settings, Reset Defaults, Preferred Vendors, Program Details, Disclaimers and Warnings, Cancel

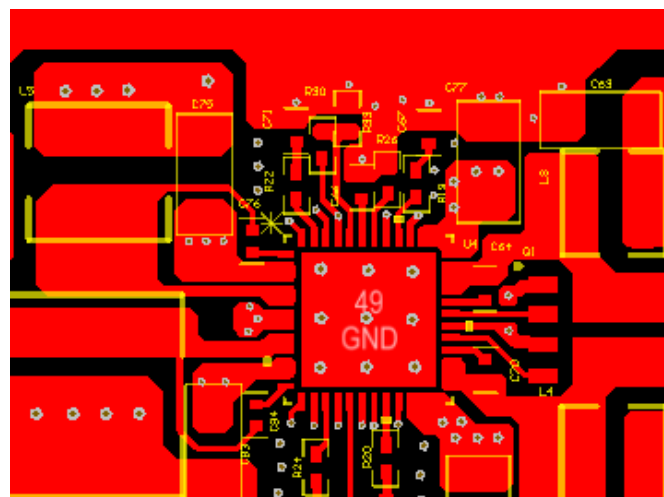
A partir de estos parámetros se obtiene:

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

Agregar imagen de circuito componentes y diagramas ?

Vías para escape térmico

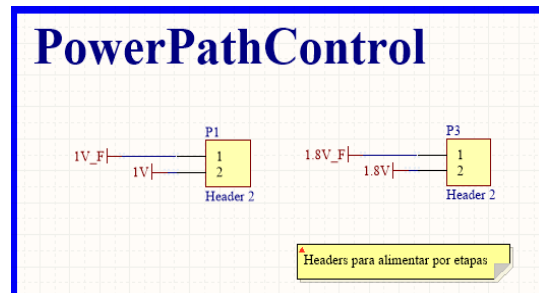
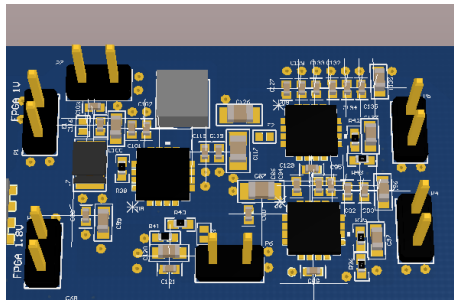
Las vías térmicas son agujeros mecánicos que atraviesan toda la placa. Estás vías son idealmente colocadas directamente bajo los componentes de montaje superficial que disipan mucho calor. Este método hace que el calor que fluye a través de las vías sea disipado en los planos solidos de GND.



## 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 ohms

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



## Calculo de corriente por vías.

Se corroboro 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 de cálculo de corriente en vías de la siguiente página:

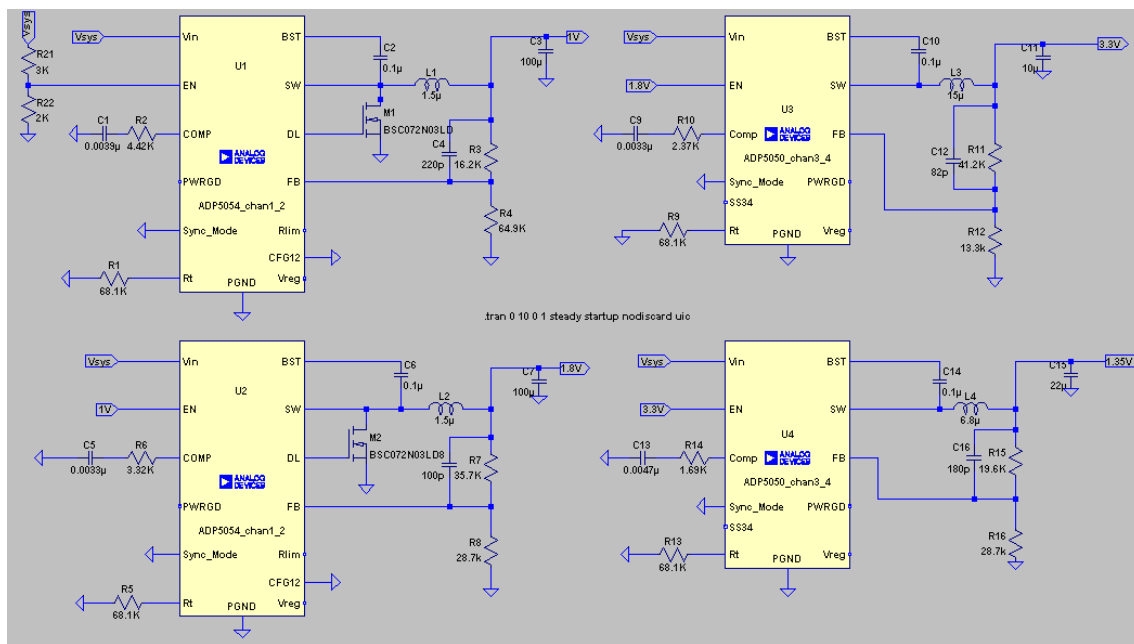
<http://circuitcalculator.com/wordpress/2006/03/12/pcb-via-calculator/>

Se estimó que los headers (tamaño de agujero = 39 mils y largo =31 mils) soportan 4 A de corriente máxima.

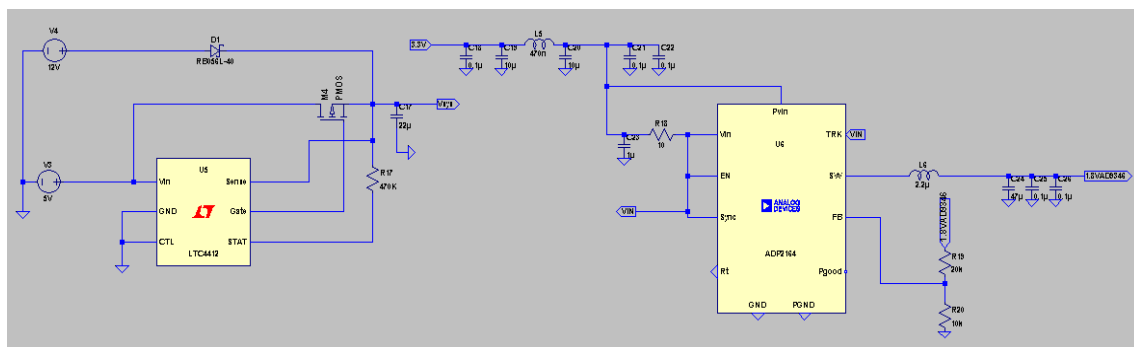
## Simulaciones Power

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.

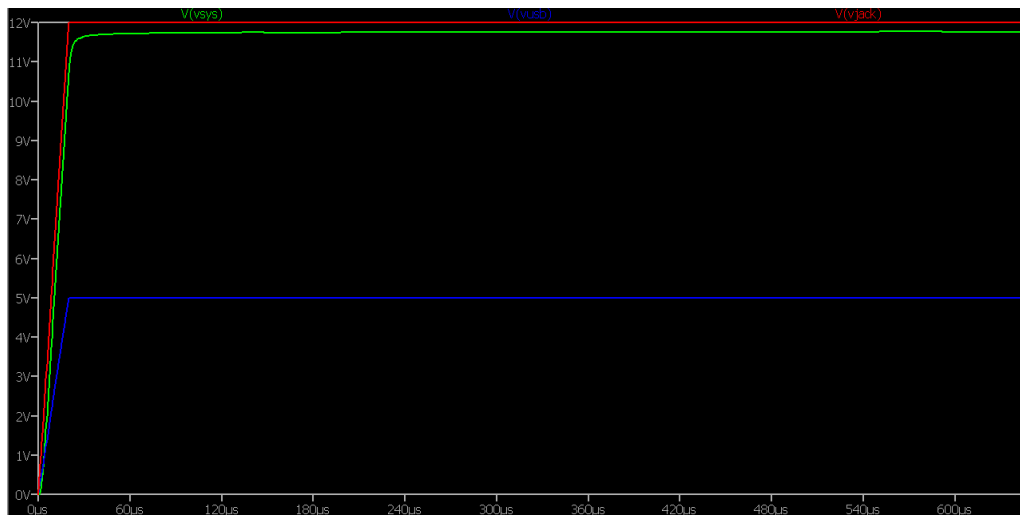
Las siguientes figuras muestran el circuito y los resultados obtenidos durante la simulación



Esquemático de los 4 buck reguladores del adp5054

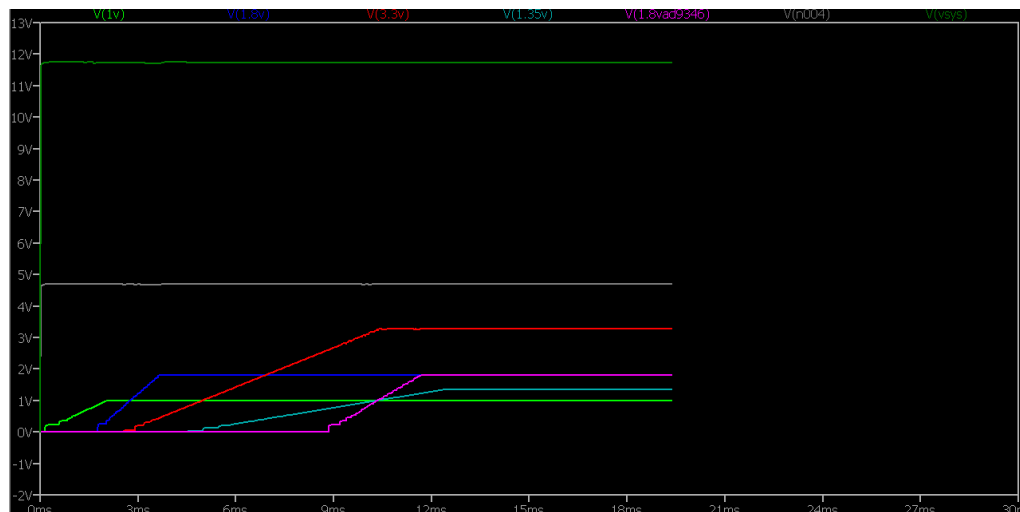


Esquemático del selector de power y el regulador ADP2164



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.

## USB 3.0

Calculo Impedancia diferencial (Cambiar utilizando la herramienta de ADS para cálculo de 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 Ohms +/- 10% requerida por el controlador USB 3.0

Microstrip & Stripline calculator - 1.31 jN 2000-2006

Trace and board parameters:

Trace width:  $W=$  13.0 mils

Trace thickness:  $t=$  1.4 mils (1.02 oz)

Trace spacing:  $S=$  5.0 mils

Dielectric (layer) thickness:  $h=$  14.0 mils ( $b=29.4$  mils)

Dielectric (layer) asymmetry: 50 % ( $h1=14.0$ ,  $h2=14.0$ )

Relative dielectric constant:  $\epsilon=$  4.6

Single-ended mode:

Characteristic impedance:  $Z_o=$  69.5 43.5  $\Omega$

Capacitance:  $C_o=$  2.06 4.30 pF/in

Delay:  $T_{pd}=$  143.2 181.8 ps/in

Speed:  $v=$  177.5 139.9 mm/ns

Differential mode:

Differential impedance:  $Z_o=$  91.7 68.6  $\Omega$

PCB edge view

Microstrip

Stripline

1. Microstrip  $Z_o$  formula accurate if  $0.1 < W/h < 2$

2. Stripline  $Z_o$  formula accurate if  $(W/b) < 0.35$

3. Stripline  $Z_o$  formula accurate if  $(b/h) > 4$

Disclaimer <http://www.jnicolle.com/?page=LVD5>

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.

PCB Rules and Constraints Editor [mil]

Design Rules

- Electrical
  - Routing
    - Routing Topology
      - Routing Priority
        - Routing Layers
          - Routing Corners
            - Routing Via Style
              - Fanout Control
                - Differential Pairs Routing
                  - RF\_Diff100
                    - DDR3\_Diff100
                      - DiffUSB\_90ohms

Where The Object Matches

Custom Query: InNetClass ('DIFF\_90')

Query Helper ...

Query Builder ...

Constraints

These Values are Applied to All Layers

Min Width: N/A, Min Gap: N/A, Preferred Width: N/A, Preferred Gap: N/A, Max Width: N/A, Max Gap: N/A

Max Uncoupled Length: 500mil

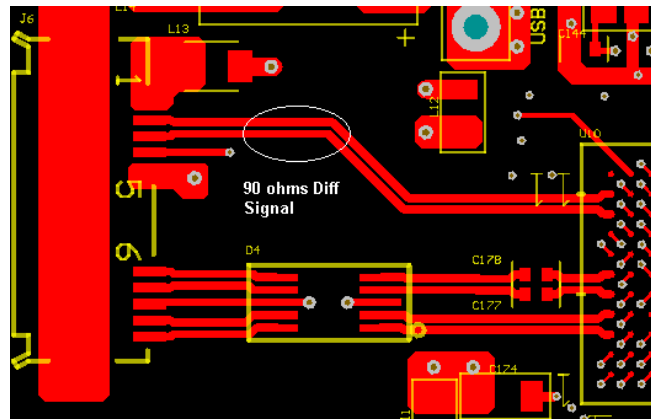
☒ Layers in layerstack only

Attributes on Layer

| Min Width | Min Gap | Preferred Width | Preferred Gap | Max Width | Max Gap | Layer Stack Reference |
|-----------|---------|-----------------|---------------|-----------|---------|-----------------------|
| 13mil     | 5mil    | 13mil           | 5mil          | 13mil     | 5mil    | Top Layer             |
| 6mil      | 7mil    | 6mil            | 7mil          | 6mil      | 7mil    | Signal 1              |
| 6mil      | 7mil    | 6mil            | 7mil          | 6mil      | 7mil    | Signal 2              |
| 6mil      | 7mil    | 6mil            | 7mil          | 6mil      | 7mil    | Bottom Layer          |

Rule Wizard... Priorities... Create Default Rules

OK Cancel Apply



Del mismo modo se calculó las demás impedancias diferenciales.

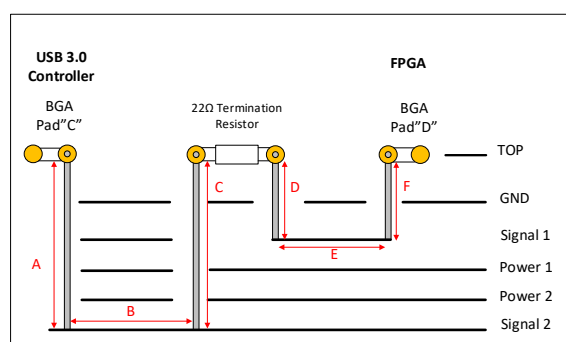
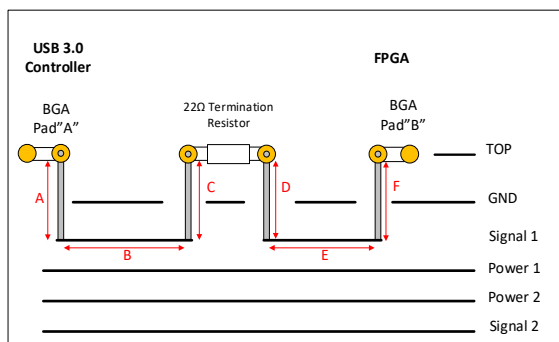
#### Protección ESD

#### 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.
- Max diferencia de largo entre las líneas 500 mils.

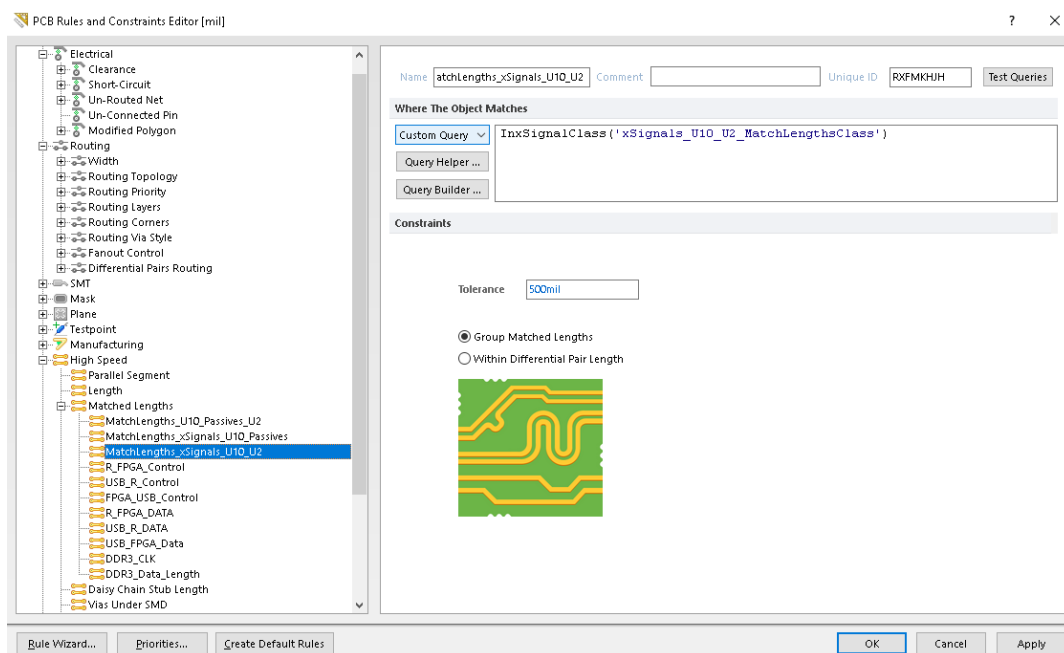
El principal problema es llevar un seguimiento del largo de cada línea, ya que el ruteo de las mismas se lleva a cabo en diferentes capas y tramos (del controlador a la resistencia y de la resistencia a la FPGA). Por ejemplo las siguientes figuras muestran dos posibles escenarios de conexión entre la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





Para cumplir los requerimientos de max 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 línea entre ambos componentes (el controlador y la FPGA).

La utilización de esta herramienta genera una regla de ruteo, mediante 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 mils es el requerimiento.

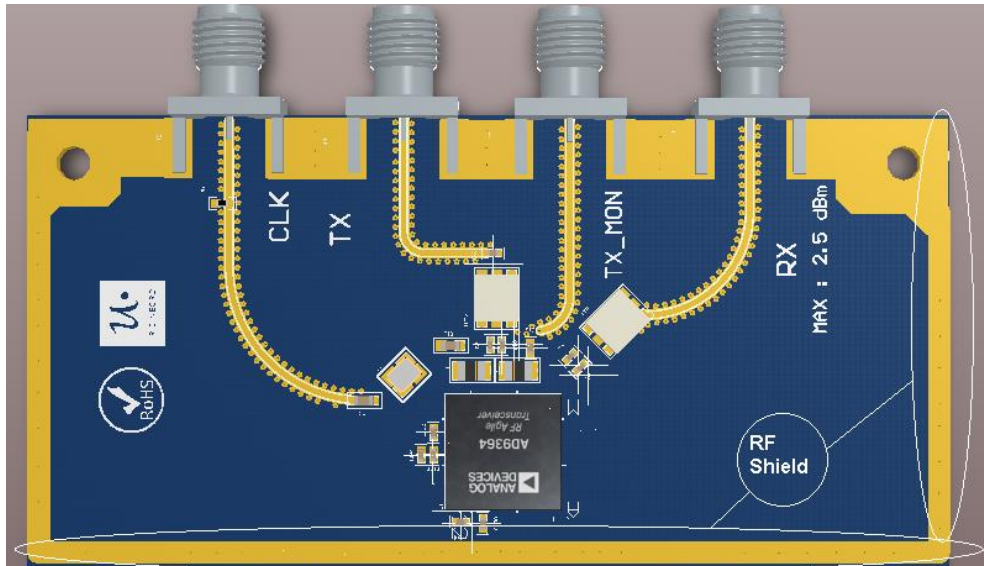
Esta herramienta en particular está disponible a partir de la versión de Altium Designer 16.0. Si se dispone de una versión anterior hay que sumar manualmente (con una tabla en Excel) cada tramo.

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:



#### Coplanar waveguide y ajuste de impedancia

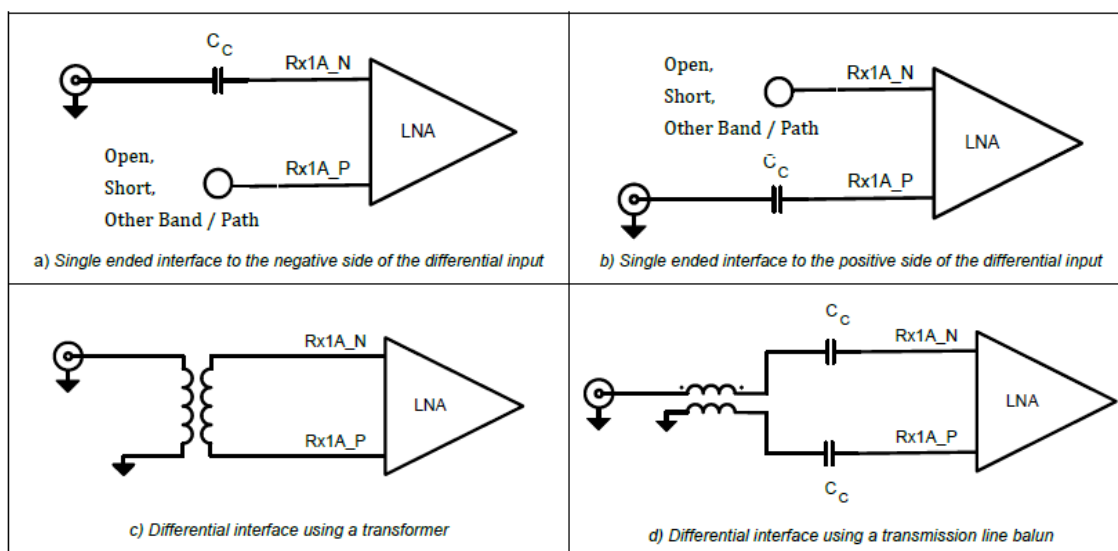
Hablar sobre el cálculo de impedancia en coplanar y sus ventajas: confinamiento campos, mejorando crosstalk etc.

Mostrar utilización de herramienta

Ajusto de stackup y ancho de pista 50 ohms rf para tener mismo ancho que conector sma y reducir la desadaptación de impedancia al pasar del pcb al conector

#### Conexiones Tx y 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 orden par de distorsión (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 los dos opciones con interface diferencial se decidió implementar la opción “D”, la cual es utilizada en la placa de desarrollo de analog devices (modelo a seguir).

(Ver y terminar de completar, Faltaría hacer alguna simulación con el ADS como muestra el FAQ )

(FAQ Link : [https://ez.analog.com/cfs-file/\\_key/telligent-evolution-components-attachments/00-441-00-00-07-90-49/AD9361RFPortInterface\\_5F00\\_v2.1.pdf](https://ez.analog.com/cfs-file/_key/telligent-evolution-components-attachments/00-441-00-00-07-90-49/AD9361RFPortInterface_5F00_v2.1.pdf) )

La impedancia de los puertos de RF (Tx y Rx) del transceiver varía según la frecuencia. 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”. Esta guía establece las siguientes recomendaciones:

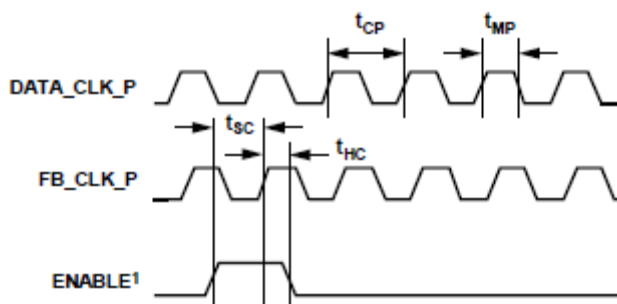
- Pistas lo más cortas posibles
- Impedancia diferencial Rx de 100 ohms
- Impedancia diferencial de Tx de 50 ohms (Problema con ancho sugerido por el FAQ de 35 mil width, la placa de desarrollo utiliza 100 ohm diff. Ante esta duda en el foro dice que al tratarse de un wideband es muy difícil encontrar la impedancia optima, por lo que indica que la conexión entre el AD9364 y el balun sea lo más corto posible)

## Criterio de interface digital transceiver

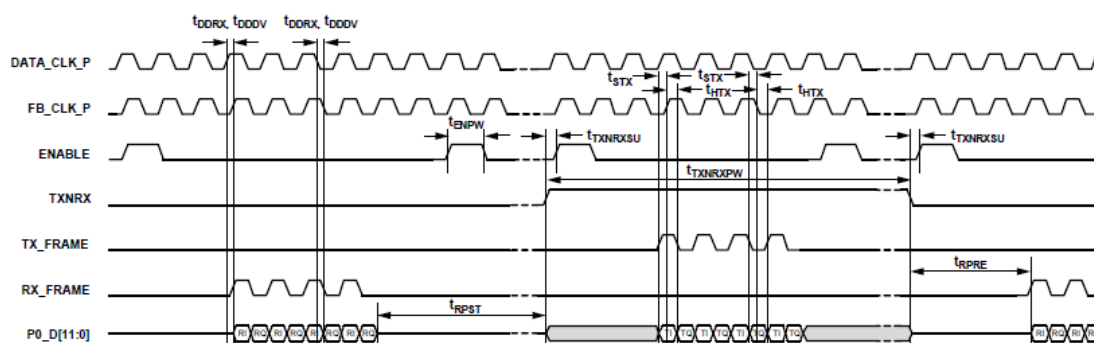
En particular este dispositivo no impone explícitamente requerimientos en cuanto a la máxima diferencia de largo entre cada línea de su interface digital con el BBP. En su lugar da restricciones en cuanto a tiempos en el camino de datos y la interface SPI como muestran las siguientes figuras:

Constrains al camino de datos:

| 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          | Rx data delay from DATA_CLK to D[11:0] outputs – 1.8 V supply  |
|               |                   |         | 1.2 ns          | Rx data delay from DATA_CLK to D[11:0] outputs – 2.5 V supply  |
| $t_{DDOV}$    | 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  |

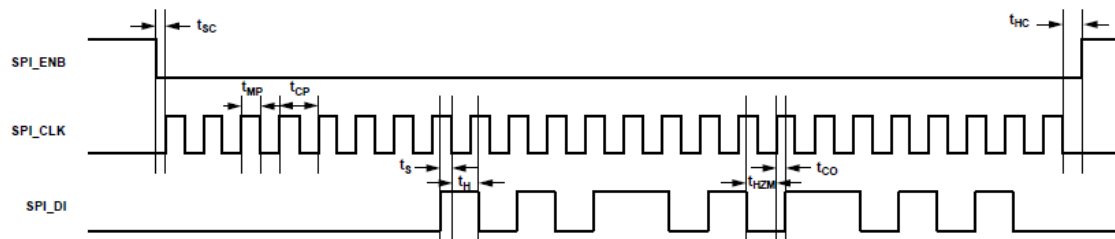


Data Port Timing Parameter Diagrams – Data Reference Clocks and Hardware Control Inputs (CMOS Bus Configuration)



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

| Parameter | Min   | Typ | Max            | Description  |
|-----------|-------|-----|----------------|--|
| $t_{CP}$  | 20 ns |     |                | SPI_CLK cycle time (clock period)                                |
| $t_{MP}$  | 9 ns  |     |                | SPI_CLK pulse width  |
| $t_{SC}$  | 1 ns  |     |                | SPI_ENB setup time to first SPI_CLK rising edge                  |
| $t_{HC}$  | 0 ns  |     |                | Last SPI_CLK falling edge to SPI_ENB hold                        |
| $t_S$     | 2 ns  |     |                | SPI_DI data input setup time to SPI_CLK                          |
| $t_H$     | 1 ns  |     |                | SPI_DI data input hold time to SPI_CLK                           |
| $t_{CO}$  | 3 ns  |     | 8 ns           | SPI_CLK rising edge to output data delay (3-wire or 4-wire mode) |
| $t_{HZM}$ | $t_H$ |     | $t_{CO} (max)$ | Bus turnaround time after BBP drives the last address bit        |
| $t_{HZS}$ | 0 ns  |     | $t_{CO} (max)$ | Bus turnaround time after AD9364 drives the last data bit        |



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

## B- Información general y hoja de datos

Recomendaciones por parte de los principales IC

| Integrado | Nombre   | Comentarios  |
|-----------|--|--|
| FPGA      | Recommended Design Rules and Strategies for BGA Devices-UG1099 | <ul style="list-style-type: none"> <li>Fan out 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<br>UG-673   | <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-<br>AN70707                      | • Recomendaciones a seguir tanto en hardware como en el esquemático.   |
| <b>DDR3 Memory</b> | Hardware Tips for Point-to-Point System Design:<br>Termination, Layout, and Routing | • Recomendaciones a seguir tanto en hardware como en el esquemático.   |

### Capacidades Fabricante PCB Prototype

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

<https://www.pcbprototype.com/en/pcb-capabilities>

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

### Información sobre conexiones de cada sección

| Integrado | Sección                             | Guía de usuario                                       |
|-----------|-------------------------------------|---|
| FPGA      | Slave Serial Configuration.<br>JTAG | 7 Series FPGAs<br>Configuration-UG470                 |
|           | DDR3 Memory                         | 7 Series FPGAs<br>Memory Interface<br>Solutions-UG586 |

|         |                        |  |
|---------|------------------------|--|
|         | I/O                    | 7 Series FPGAs<br>SelectIO Resources-UG471   |
|         | Clock                  | 7 Series FPGAs<br>Clocking Resources-UG472   |
| USB 3.0 | Interface GPIF<br>FPGA | Designing with the EZ-USB®<br>FX3™ Slave FIFO Interface<br>AN65974                     |
|         | Boot FPGA              | Configuring an FPGA Over USB<br>Using Cypress EZ-USB® FX3™<br>AN84868                  |
|         | Interface USB          | EZ-USB® FX3™/FX3S™<br>Hardware Design Guidelines<br>and Schematic Checklist<br>AN70707 |
|         | Boot USB 3.0           | EZ-USB® FX3™/FX3S™ Boot<br>Options-AN76405   |
| AD9364  | Interface              | AD9364 Reference Manual<br>UG-673  |

Esquemático, Gerbers, BOM

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

### Referencias

Application Note on Transformers (AN-20-002) Minicircuits



## Bibliografia

Eugene Grayver-Implementing Software Defined Radio

Software Defined Radio Evolution with MIMO (Multi Input Multi Output)  
Technique:

<https://www.ijser.org/paper/Software-Defined-Radio-Evolution-with-MIMO-Multi-Input-Multi-Output-Technique.html>