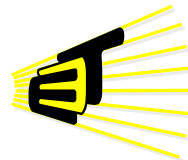


ANÁLISIS PARA LA IMPLEMENTACIÓN DE UN ANALIZADOR  
VECTORIAL DE REDES MEDIANTE SOFTWARE RfNoC EN UN  
USRP X310

BRAYAN LEONARDO BECERRA QUINTERO  
ANDRÉS SEBASTIÁN OLIVARES ÁVILA



**CPS** | RESEARCH GROUP

Universidad Industrial de Santander  
Facultad de Ingenierías Fisicomecánicas  
Escuela de Ingenierías Eléctrica, Electrónica y de  
Telecomunicaciones  
Bucaramanga  
2018

ANÁLISIS PARA LA IMPLEMENTACIÓN DE UN ANALIZADOR  
VECTORIAL DE REDES MEDIANTE SOFTWARE RFNoC EN UN  
USRP X310

Autor:

BRAYAN LEONARDO BECERRA QUINTERO  
ANDRÉS SEBASTIÁN OLIVARES ÁVILA

*Trabajo de grado presentado como requisito para optar al título de*  
Ingeniero Electrónico

Director:

Óscar Mauricio Reyes Torres  
Ph.D en Ingeniería Electrónica

Codirector:

Efrén Darío Acevedo Cárdenas  
Magíster en Ingeniería Electrónica

Universidad Industrial de Santander  
Facultad de Ingenierías Fisicomecánicas  
Escuela de Ingenierías Eléctrica, Electrónica y de  
Telecomunicaciones

Bucaramanga

2018

---

## DEDICATORIA

---

*A mis padres, hermanos, hermana y mi sobrina que siempre han estado presente.*

*A mi compañera de vida, sueños, luchas y resistencia.*

***Brayan Leonardo Becerra Quintero.***

*Especialmente a mis padres y a mis hermanas que fueron y son el gran motor de mi vida para cumplir esta meta y las venideras.*

***Andrés Sebastián Olivares Ávila.***

---

## AGRADECIMIENTOS

---

*Gracias.*

*A mis padres Leonardo Becerra y Nora Quintero por todo el esfuerzo y amor dedicado.*

*A mis hermanos Brandon y Daniel Becerra; a mi hermana Daniela Becerra por el apoyo brindado durante este proceso y a lo largo de mi vida.*

*A Judith Gualdrón por ser compañía, comprensión, amor y siempre haber estado en los momentos más difíciles vividos hasta el día de hoy.*

*A los Profesores Óscar Reyes y Efrén Acevedo por su colaboración y orientación en la elaboración de este proyecto.*

*A mis más llegados amigos de café y aventuras en esta etapa que ya culmina: Mafe, Julián, Jorge, Liliana, Johan entre otros. Por su compañía y los momentos agradables que compartimos.*

**Brayan Leonardo Becerra Quintero.**

*A mi madre Blanca Ávila, a mi padre Fredy Olivares y a mis hermanas Karen y Lorena por todo el esfuerzo enorme que me brindaron; a cada uno de los miembros de mi familia que me apoyaron en todo momento.*

*A mi compañera sentimental Margarita Castellanos, que estuvo siempre a mi lado dándome su cariño y apoyo incondicional. A mi compañero de trabajo de grado Brayan Becerra por toda la colaboración y dedicación.*

*A todos los miembros del inolvidable marco en especial a David por brindarme ese gran apoyo incondicional como amigo.*

*A nuestro director de proyecto el profesor Óscar Reyes por guiarnos y colaborarnos en este proyecto, a los profesores Efrén Acevedo, Carlos Fajardo y Carlos Ángulo por aportarnos también sus experiencia y orientación.*

*Por último, a todos mis compañeros y compañeras que tuve durante toda la carrera que hicieron agradable cada espacio de clase.*

**Andrés Sebastián Olivares Ávila.**

---

# CONTENIDO

---

	Pág.
<b>INTRODUCCIÓN</b>	<b>16</b>
<b>1 ANALIZADOR VECTORIAL DE REDES - VNA</b>	<b>19</b>
1.1. GENERADOR . . . . .	21
1.2. BANCO DE PRUEBAS . . . . .	21
1.2.1. Elemento direccional como banco de prueba . . . . .	23
1.3. RECEPTOR . . . . .	27
1.4. PROCESADOR . . . . .	29
<b>2 SÍNTESIS DE UN VNA SOBRE UN USRP</b>	<b>31</b>
2.1. GENERADOR DE <i>RF</i> EN <i>USRP</i> . . . . .	36
2.1.1. Generador en <i>RFNoC</i> . . . . .	39
2.1.2. Receptor en el <i>USRP</i> . . . . .	40
2.1.3. Procesador con <i>RFNoC</i> . . . . .	41
<b>3 PRUEBAS Y RESULTADOS</b>	<b>44</b>
3.1. CALIBRACIÓN DEL GENERADOR <i>RFNoC</i> . . . . .	44
3.2. VALIDACIÓN GENERADOR DE <i>RF</i> EN <i>USRP</i> . . . . .	48
3.3. RECURSOS <i>FPGA</i> . . . . .	52
3.3.1. Recursos base <i>RFNoC</i> . . . . .	52
3.3.2. Recursos bloque <i>RFNoC:Amp</i> . . . . .	54
3.3.3. Recursos para alternativas de implementación del procesador . . . . .	55
3.4. ACOUPLE DIRECCIONAL 87300B . . . . .	57
3.4.1. Caracterización de un acople direccional 87300B . . . . .	57
3.4.2. Alternativas para realizar el test set . . . . .	61
<b>4 CONCLUSIONES</b>	<b>62</b>

<b>5 RECOMENDACIONES</b>	<b>64</b>
<b>BIBLIOGRAFÍA</b>	<b>66</b>
<b>ANEXOS</b>	<b>70</b>

---

## LISTA DE FIGURAS

---

	Pág.
Figura 1. Esquema de VNA. . . . .	20
Figura 2. Divisor de potencia del <i>test set</i> . . . . .	22
Figura 3. Puente Wheatstone. . . . .	23
Figura 4. Puente <i>VSWR</i> . . . . .	24
Figura 5. Acople direccional y configuraciones (a) Acople direccional (b) configuración en reversa y (c) configuración en seguidor. . . . .	26
Figura 6. Receptor de VNA basado en el principio heterodino . . . . .	27
Figura 7. VNA sobre el USRP X310. . . . .	31
Figura 8. Impulso Gaussiano. . . . .	32
Figura 9. Sinusoidal truncada. . . . .	33
Figura 10. Fase de referencia. . . . .	35
Figura 11. Propuesta implementación de un <i>VNA</i> en un <i>USRP</i> desde GNU Radio. . . . .	36
Figura 12. Transmisor del <i>USRP</i> X310. . . . .	37
Figura 13. Diagrama de bloques funcional del ADL5375. . . . .	38
Figura 14. Diagrama de generador de <i>RF</i> en el <i>USRP</i> X310. . . . .	39
Figura 15. Diagrama Amp. . . . .	40
Figura 16. Receptor de la tarjeta hija <i>SBX – 40</i> . . . . .	41
Figura 17. Diagrama de bloques del procesador. . . . .	42
Figura 18. conexión de los equipos para la prueba de verificación del gene- rador de <i>RF</i> implementado en un <i>USRP</i> . . . . .	45
Figura 19. Curvas de potencia de un generador de <i>RF</i> y del <i>USRP</i> como generador de <i>RF</i> . . . . .	46
Figura 20. USRP como generador de <i>RF</i> a distintas frecuencias de muestreo ( $f_s$ ). . . . .	46
Figura 21. Curvas para ajuste de ganancia para transmisión del <i>USRP</i> . . . . .	47

---

Figura 22. Diagrama de bloques para implementar generador de RF en <i>USRP</i> con ajuste de ganancia. . . . .	49
Figura 23. Comparación de las curvas de potencia de transmisión del <i>USRP</i> con ajuste de ganancia lineal y generador de <i>RF</i> . . . . .	50
Figura 24. Comparación de las curvas de potencia de transmisión del <i>USRP</i> con ajuste de ganancia polinómica y generador de <i>RF</i> . . . . .	51
Figura 25. Comparación de las curvas de potencia de transmisión del <i>USRP</i> con ajuste de ganancia potencial y generador de <i>RF</i> . . . . .	51
Figura 26. Curva de pérdidas para un acople direccional 87300B de 1GHz a 3 GHz. . . . .	58
Figura 27. Curva de factor de acoplamiento para un acople direccional 87300B de 1GHz a 3 GHz. . . . .	59
Figura 28. Curva de <i>SWR</i> para un acople direccional 87300B de 1GHz a 3 GHz. . . . .	60

---

## LISTA DE TABLAS

---

		<b>Pág.</b>
Tabla 1.	Rango de frecuencia de VNA's <i>R&amp;S</i> <sup>®</sup> ZVLx. . . . .	19
Tabla 2.	Características generador del VNA <i>R&amp;S</i> <sup>®</sup> ZVL6. . . . .	21
Tabla 3.	Características del sintetizador ADF4350. . . . .	37
Tabla 4.	<i>Slice Logic</i> mínimas para síntesis de <i>RFNoC</i> . . . . .	53
Tabla 5.	<i>Memory</i> mínima para síntesis de <i>RFNoC</i> . . . . .	53
Tabla 6.	<i>DSP</i> mínima para síntesis de <i>RFNoC</i> . . . . .	53
Tabla 7.	<i>Clocking</i> mínima para síntesis de <i>RFNoC</i> . . . . .	54
Tabla 8.	Recursos <i>slice Logic</i> del bloque <i>Amp</i> . . . . .	55
Tabla 9.	Recursos <i>Memory</i> del del bloque <i>Amp</i> . . . . .	55
Tabla 10.	Recursos <i>DSP</i> del bloque <i>Amp</i> ... . . . .	55
Tabla 11.	Recursos utilizados en los tipos de implementación. . . . .	56
Tabla 12.	Recursos utilizados por cada algoritmo. . . . .	56
Tabla 13.	Especificaciones del acople direccional 87300B. . . . .	57
Tabla 14.	Caracterización del acople direccional 87300B para un rango de 1 GHz a 3 GHz. . . . .	60

---

## LISTA DE ANEXOS

---

	<b>Pág.</b>
Anexo A. Plantillas bloque <i>Amp</i> . . . . .	70
Anexo B. Errores . . . . .	79

---

## RESUMEN

---

<b>TÍTULO:</b>	ANÁLISIS PARA LA IMPLEMENTACIÓN DE UN ANALIZADOR VECTORIAL DE REDES MEDIANTE SOFTWARE RFNoC EN UN USRP X310 *
<b>AUTORES:</b>	Brayan Leonardo Becerra Quintero ** Andrés Sebastián Olivares Ávila **
<b>PALABRAS CLAVE:</b>	USRP, RFNoC, FPGA, generador, banco de pruebas, receptor, procesador.

**DESCRIPCIÓN:** En este trabajo se realizó un análisis de las prestaciones de un *USRP* X310 [9] y un analizador vectorial de redes con el fin de determinar la viabilidad de implementar este último sobre una plataforma *SDR*. Así mismo, se utilizó el entorno *RFNoC* [11] desarrollado por Ettus Research [10] buscando el mayor rendimiento para esta plataforma, interactuando directamente con las *FPGAs* embebidas en los *USRP*[20]. Se describieron y analizaron los cuatro componentes que constituyen el analizador vectorial de redes, la *FPGA Kintex-7 XC710K410T* [30] y las tarjetas hijas compatibles del *USRP* X310. Se plantearon las alternativas para implementar de forma individual tres componentes de un *VNA* bajo el entorno *RFNoC*. Se generó un bloque *RFNoC* para la implementación de un generador de *RF* similar al de un *VNA* y se comparó su respuesta en frecuencia con un generador de *RF* comercial. Se plantea utilizar un acople direccional como banco de pruebas para lo cual se realizó una caracterización de uno de estos, determinando sus parámetros específicos, necesarios para realizar ajustes al momento de implementar un *VNA*. Para el receptor y el procesador se presentan alternativas de implementación con base en las prestaciones de las tarjetas hijas del *USRP* y los bloques que *RFNoC* y *Xilinx* facilitan a los usuarios de sistemas *SDR*. La curva de potencia del generador de *RF* implementado en el *USRP* oscila alrededor de -0.78 dBm con variaciones de  $\pm 0.66$  dB similar a un generador de *RF* comercial con el que se comparó. El bloque construido en *RFNoC* para esta etapa de generador consume menos del 2% de recursos *FPGA* disponibles.

---

\*Trabajo de investigación.

\*\* Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Óscar Mauricio Reyes Torres. Ph.D en Ingeniería Electrónica.

---

## ABSTRACT

---

**TITLE:** ANALYSIS FOR THE IMPLEMENTATION OF A VECTORIAL NETWORK ANALYZER USING RFNoC SOFTWARE IN A USRP X310 \*

**AUTHORS:** Brayan Leonardo Becerra Quintero \*\*  
Andrés Sebastián Olivares Ávila \*\*

**KEYWORDS:** USRP, RFNoC, FPGA, generator, test set, receiver, processor.

### DESCRIPTION:

In this work, a performance analysis of a USRP X310 and a Vector Network Analyzer was carried out in order to determine the feasibility of implementing the latter on a SDR platform. Likewise, RFNoC software, developed by Ettus Research, was used looking forward to a better performance for this platform, interacting directly with FPGA motherboards embedded in USRP. The four components that are part of the Vector Network Analyzer were analyzed and described, as well as the FPGA Kintex-7 XC710K410T and compatible daughter boards of the USRP X310. Additionally, some alternatives were proposed to implement three components of a VNA in an individual form on the RFNoC environment. A RFNoC block was generated to implement a RF generator equivalent to the correspondent VNA generator, and its frequency response was compared with a commercial RF generator. The use of a directional coupling is proposed as a test bench and, in particular, the 87300B was characterized, defining specific parameters required to be adjusted for implementing a VNA. For the receiver and processor, implementation alternatives were presented based on the specifications of the USRP daughter boards and the blocks provided by RFNoC and Xilinx environments. The PSD of the RF generator implemented in the USRP oscillates around -0.78 dBm with variations of  $\pm 0.66$  dB similarly to the RF generator used as a reference. IT is remarkable that the RFNoC block synthesized for this generator stage consumes less than 2% of available FPGA resources.

---

\*Bachelor Thesis

\*\* Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Óscar Mauricio Reyes Torres. Ph.D en Ingeniería Electrónica.

---

## INTRODUCCIÓN

---

En el área de las comunicaciones surge la necesidad de contar con instrumentación que brinden mejoras a los procesos de investigación y desarrollo. Dentro de estos procesos encontramos el análisis y la medición de circuitos de radio frecuencia (*RF*). Los Analizadores Vectoriales de Redes (*vector network analyzer - VNA*) aparecen como dispositivos de alta precisión, eficiencia y capacidad para analizar una gran variedad de sistemas que pueden ir desde sencillos modelos de filtros y amplificadores hasta sistemas de gran complejidad como bases y estaciones de comunicaciones [14].

También se han venido desarrollando los llamados sistemas de radio definido por software (*software defined radio SDR*), que buscan facilitar la implementación de sistemas de radio, a través de la descripción en software de dispositivos que normalmente se encuentran implementados en hardware. Los sistemas *SDR* permiten cubrir gran variedad de sistemas receptores y transmisores sobre un mismo procesador de hardware, mediante la modificación de los algoritmos que se ejecutan con el software [17]. Los sistemas *SDR* se pueden utilizar sobre receptores, transmisores y transceptores de radio como el *USRP* (*universal software radio peripheral*) [20]. Los *USRP* de última generación brindan altas prestaciones, una de ellas es su tarjeta madre, las *FPGA* de séptima familia y tercera generación *XILINX*. Pero, a pesar de la cantidad de recursos con los que cuenta, su uso no se encuentra optimizado al máximo pues el procesamiento digital se realiza sobre el equipo de cómputo con el que se configura el *USRP* a través de software como MATLAB, LabVIEW, Simulink y GNU Radio [19]. Ante esta situación, Ettus Research <sup>5</sup> implementó el entorno *RFNoC*, con el objetivo de implementar bloques o máquinas computacionales sobre la *FPGA* de los radios, en pro de mejorar el tiempo de procesamiento de la información y generar mayor aprovechamiento de estas tarjetas madre.

Pensando en la posibilidad de implementar dispositivos *SDR*, se plantea la posibi-

---

<sup>5</sup>Filial de *National Instruments*, proveedor líder mundial de plataformas *SDR*, incluida la familia de productos *Universal Software Radio Peripheral*.

lidad de implementar instrumentación para telecomunicaciones con base en dispositivos *SDR*, Por lo cual se realizar un análisis de viabilidad para la implementación de un *VNA* sobre un *USRP* utilizando el entorno *RFNoC*, como alternativa para descripción de sistemas sobre *SDR*. Para esto, se realiza un análisis individual de los componentes que constituyen un analizador vectorial de redes con el fin de describir el principio de funcionamiento de cada uno, dado que el análisis en conjunto presenta mayor dificultad. Conocidas las estructuras de los componentes, se realiza una revisión a las prestaciones de un *USRP X310* y se establecen las similitudes que permiten determinar algunas propuestas para la implementación del *VNA*.

Se decide implementar la etapa del generador de *RF* en *RFNoC*, aprovechando un trabajo de grado precedente sobre este entorno. Además la escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones (E3T) de la Universidad Industrial de Santander cuenta con un generador de *RF* comercial que sirve como referencia para validar el generador implementado. En esta comparación se evidencia la necesidad de realizar una calibración pues la respuesta en frecuencia de la etapa transmisora del *USRP X310* presenta diferencias con la del generador de *RF* comercial.

El presente documento está organizado en cinco capítulos. En el capítulo 1 se presentan los cuatro componentes que constituyen un analizador vectorial de redes (generador, test set o banco de pruebas, receptor y procesador); realizando una descripción individual de los mismos. Cumpliendo así, el primer objetivo específico planteado para este trabajo, describir los cuatro componentes fundamentales de un analizador vectorial de redes. Esta descripción permite determinar o identificar los requerimientos para implementar algunos de los componentes mediante *SDR*.

En el capítulo 2 se presentan algunas propuestas para la implementación de un *VNA* sobre un *USRP*, de las cuales se profundizo en una de ellas. Se desarrolla e implementa una máquina computacional o bloque *RFNoC* similar al componente generador de un *VNA*. Se realiza una caracterización de un acople direccional que

permite ser utilizado como banco de pruebas. Se detallan los elementos que constituyen las tarjetas hijas de los *USRP* que permiten la transmisión y recepción de *RF* y se presentaron propuestas para la configuración de los bloques que constituyen la etapa de procesamiento. En el capítulo 3 se documentan las pruebas y resultados de la etapa de generador de *RF* implementado y los recursos que se necesitan para dicha implementación sobre la *FPGA* y la caracterización del acople direccional 87300B y finalmente se cierra con los capítulos de conclusiones y recomendaciones.

En la sección de anexos se documentan las plantillas que se utilizaron para implementar el bloque Amp del generador de *RF* Y los errores que se presentan con el entorno *RFNoC* al momento de crear el bloque y algunas propuestas para solucionarlos.

---

## 1. ANALIZADOR VECTORIAL DE REDES - VNA

---

En este capítulo se describen las características funcionales de los cuatro componentes que constituyen un analizador vectorial de redes y los requerimientos necesarios para implementar cada uno de estos sobre FPGA. El capítulo se divide en cuatro subsecciones, una para cada componente. Para la realización de este trabajo de investigación se tuvo como referencia el VNA  $R\&S^{\text{®}}$  ZVL6 con el que cuenta el laboratorio de comunicaciones de la Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones de la Universidad Industrial de Santander, por lo que las características funcionales de los componentes que se presentan en este capítulo corresponden a las del analizador mencionado. El VNA  $R\&S^{\text{®}}$  ZVL6 pertenece a una familia de VNA's fabricados por la compañía Rohde & Schwarz GmbH & Co. KG [23], de los cuales las series ZVL3, ZVL3-75, ZVL6 y ZVL13; son VNA's de dos puertos y difieren entre si por los rangos de frecuencias que operan para realizar mediciones [24], como se ve en la Tabla 1. De la cual es posible resaltar al ZVL6 pues su rango de frecuencia se asemeja al del USRP X310. Por otra parte, los de las series ZVA, ZVC, ZVL, ZVT, ZNBT, ZN y ZNB son VNA's de más de dos puertos y los ZNxx ofrecen mejores prestaciones como la representación de datos en la pantalla.

**Tabla 1:** Rango de frecuencia de VNA's  $R\&S^{\text{®}}$  ZVLx.

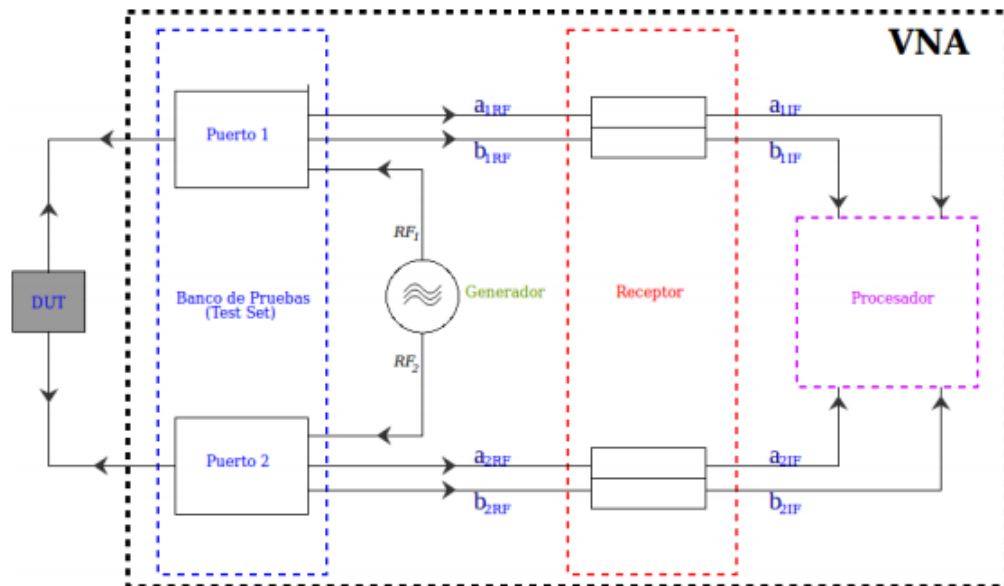
VNA	Rango de frecuencias
$R\&S^{\text{®}}$ ZVL3 y $R\&S^{\text{®}}$ ZVL3-75	9 kHz - 3 GHz
$R\&S^{\text{®}}$ ZVL6	9 kHz - 6 GHz
$R\&S^{\text{®}}$ ZVL—3	9 kHz - 13.6 GHz

Un *VNA* es un instrumento que permite determinar las características de una red de  $N$  puertos en una determinada banda de frecuencias. Estas características son: los parámetros de dispersión de la red y las impedancias de entrada y salida de la misma.

Una de las principales diferencias entre un analizador vectorial de redes y otros instrumentos de análisis como el analizador de espectros o el analizador de redes escalar, es que el primero ofrece información de magnitud y fase de una señal relativa a la señal de prueba. Sus resultados se pueden visualizar en diagramas polares, carta de Smith e incluso en diagramas de Bode [13].

El analizador vectorial de redes se encuentra estructurado en cuatro etapas o componentes fundamentales, que son: un generador de señales, un banco de pruebas (*test set*), un receptor y un procesador; que se describen a continuación.

**Figura 1:** Esquema de VNA.



En la Figura 1 se presenta un esquema de la estructura del *VNA* en donde se visualiza mediante un diagrama de bloques el principio de funcionamiento del analizador, que consiste en aplicar señales sinusoidales de radio frecuencia (*RF*) producidas por un generador, realizando un barrido de frecuencias en determinada banda sobre un dispositivo bajo prueba (*device under test* - *DUT*), como una red de dos puertos. En el cable (comúnmente coaxial) utilizado para la conexión entre el *DUT* y el *VNA* se presenta superposición de las señales incidentes  $a_{1,2}$  y las reflejadas  $b_{1,2}$  desde los puertos 1 y 2. Es por esto que surge la necesidad de separarlas antes de iniciar

el procesamiento haciendo uso del banco de pruebas. Posteriormente, son entregadas al receptor, cuya función es llevar las señales de RF a frecuencias intermedias ( $IF$ ) para aplicar una conversión Analógico- Digital a las señales y así facilitar el procesamiento digital (*digital signal processing - DSP*) que ocurre en el procesador del analizador. En la Figura 1 se agregan los subíndices  $RF$  e  $IF$  para aclarar en qué rango de frecuencias se encuentran las señales  $a_{1,2}$  y  $b_{1,2}$ .

### 1.1 GENERADOR

Un analizador vectorial de redes utiliza un generador de señales sinusoidales, este se basa en un oscilador electrónico sintonizable con rango de sintonización amplio y se encuentra en un  $PLL$  (*phase locked loop*) en el que su oscilador de referencia presenta estabilidad a largo plazo y pureza espectral. Entre los osciladores de cristal comúnmente utilizados en estos dispositivos, se encuentran el de temperatura compensada (*temperature – compensated crystal oscillators – TCXO*) [7] y el de temperatura controlada (*oven controlled temperature crystal oscillators – OCXO*) [27]. El  $PLL$  vincula la fase y la señal de oscilador sintonizable con el de referencia para generar señales de estímulo con la frecuencia deseada para la prueba. Las características del generador del VNA  $R\&S^{\text{®}}$  ZVL6 [22] se presentan en la Tabla 2.

**Tabla 2:** Características generador del VNA  $R\&S^{\text{®}}$  ZVL6.

<b>Características</b>	<b>Valores</b>
<b>Rango de frecuencia</b>	9kHz a 6GHz
<b>Resolución de frecuencia</b>	1 Hz
<b>Rango de potencia</b>	-50dBm a 0dBm
<b>Resolución de potencia</b>	0.01dB
<b>Precisión de potencia</b>	>2dB (fuente a 10MHz y -10dB)

### 1.2 BANCO DE PRUEBAS

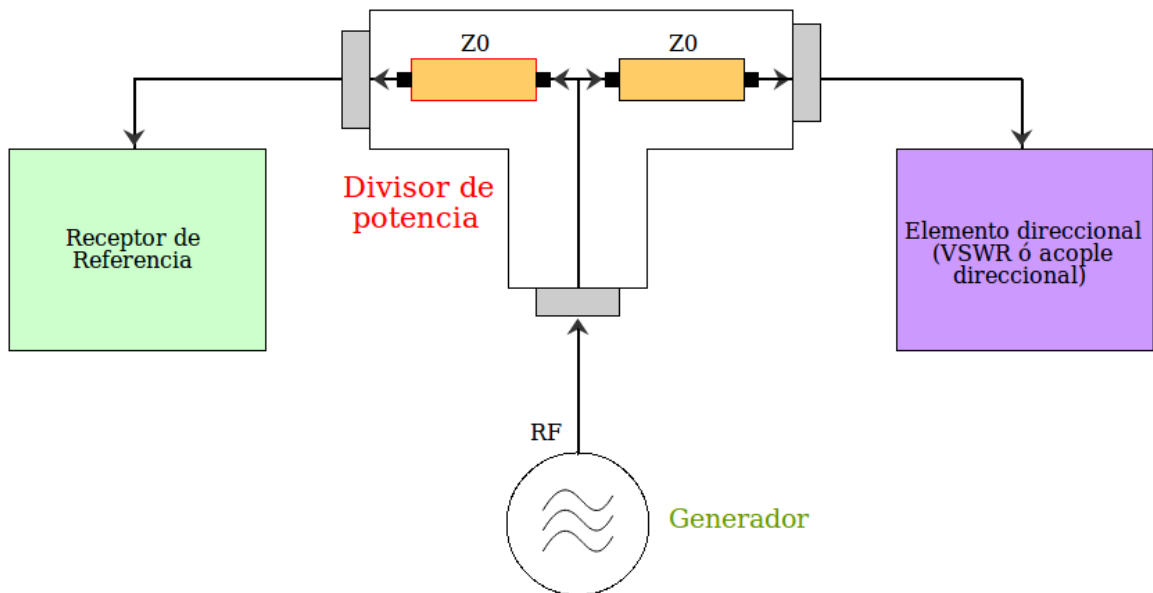
O *test set* es otro de los cuatro componentes fundamentales que constituyen un analizador vectorial de redes. Su función, se centra en la separación de las ondas incidente y reflejada de los  $DUT$  para entregar a los puertos de pruebas. Cuando se

diseña este dispositivo, es importante hacerlo bajo ciertos parámetros, tales como:

- Constancia de la onda.

La constancia de la onda es de suma importancia, pues si ésta no mantiene un nivel aproximadamente constante, con un tolerancia aceptable de  $\pm 0,3dB$  el resultado de la medición puede presentar errores, que pueden ser controlados por medio de un receptor de referencia al cual se le aplica la señal de entrada mediante un divisor de potencia como el de la Figura 2, el cual garantiza que la señal aplicada al receptor de referencia y la del elemento direccional sea la misma, la emitida por el generador de señales del analizador vectorial.

**Figura 2:** Divisor de potencia del *test set*.

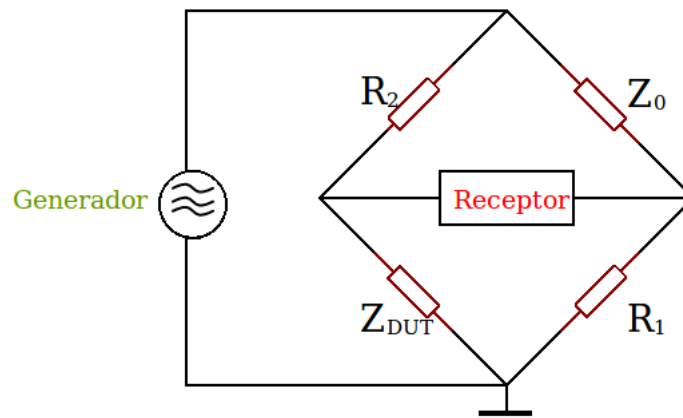


- Rango de frecuencias.

El rango de frecuencias del *test set* debe satisfacer la banda de frecuencias en la que el *VNA* realiza el respectivo análisis. Este rango está dado por los elementos, circuitos o sistemas que componen el *test set*, lo cual implica que éstos se deben seleccionar por su respuesta en frecuencia; buscando que no se afecte la medición.

**1.2.1 Elemento direccional como banco de prueba** Para el diseño de este dispositivo se utiliza un elemento direccional, que puede ser implementado a través de un puente *VSWR* (*Voltage Standing Wave Ratio*) y acoples direccionales. Éstos se utilizan en conjunto debido a que sus rangos de frecuencias son diferentes, pero que al operar en un mismo arreglo, se complementan. Un puente *VSWR* posee un rango de frecuencias que usualmente va desde algunos kHz hasta 4GHz aproximadamente. Por otra parte los acoples direccionales pueden trabajar desde frecuencias de 1GHz y según la necesidad, se pueden diseñar para trabajar a frecuencias de hasta 100GHz.

**Figura 3:** Puente Wheatstone.



#### ■ El puente *VSWR*

El puente *VSWR* se puede ver como el circuito de un puente Wheatstone de la Figura 3 conformado por 2 resistencias  $R_1$ ,  $R_2$  y 2 impedancias:  $Z_0$  y  $Z_{DUT}$  (impedancia del *DUT*). Si se busca obtener un puerto de prueba ideal, el coeficiente de reflexión de la tensión del puerto de salida  $S_{11}$  debe ser igual a cero, así mediante el análisis de circuitos se obtiene:

$$S_{11} = \frac{R_1 R_2 - Z_0^2}{2((R_1 + Z_0)(R_2 + Z_0))} \quad (1.1)$$

Por lo cual se debe cumplir que  $R_1 R_2 = Z_0^2$ . Con esta condición, los otros

parámetros de dispersión son:

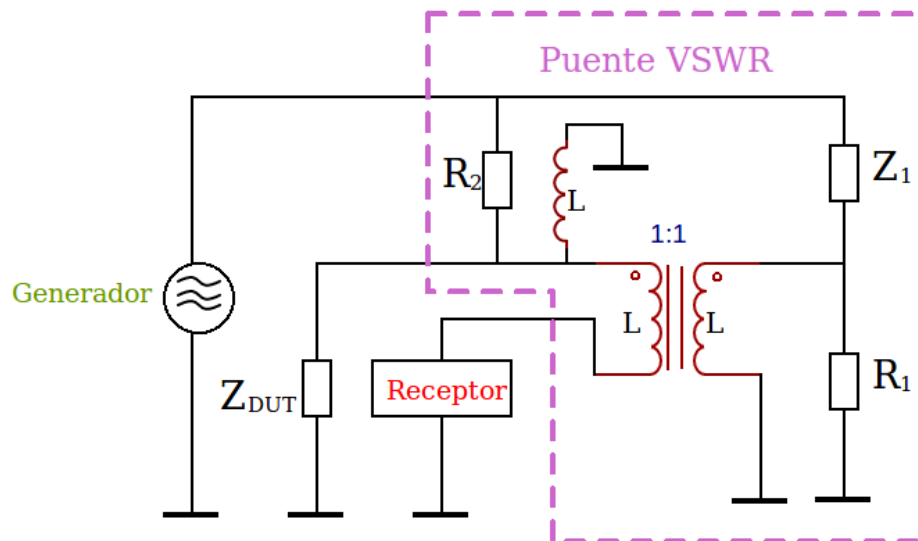
$$S_{12} = \frac{Z_0}{R_2 + Z_0} \quad (1.2)$$

$$S_{21} = \frac{Z_0}{R_2 + Z_0} \quad (1.3)$$

$$S_{22} = 0 \quad (1.4)$$

Con este circuito se puede presentar un problema al tomar la medida de tensión en la diagonal del puente, ya que es posible que se presente un corto circuito si esta medición no se hace de manera balanceada. Es por esto que se utiliza un transformador con relación de espiras de 1:1 para obtener este voltaje con referencia a tierra y recordando que el puente Wheatstone requiere que sus resistencias  $R_1$  y  $R_2$  sean iguales, es necesario colocar una inductancia del mismo valor en el nodo en el que se coloca el primario del transformador y así se compensa el circuito asimétrico que genera el transformador. Ahora, este transformador restringe el ancho de banda y por ende este elemento direccional se utiliza para señales de hasta 4GHz, aproximadamente. El nuevo sistema es lo que se conoce como puente *VSWR* y se ve en la Figura 4.

**Figura 4:** Puente *VSWR*.



### ■ Acoples direccionales

Los acoples direccionales son dispositivos lineales de cuatro puertos (entrada, transmisión, acople y aislado) y consisten en dos líneas con longitud de  $\lambda/4$  estrechamente separadas y acopladas como se ve en la Figura 5a. Al ser un dispositivo pasivo lineal, resulta indiferente cuál es el puerto al que se le inyecta la señal y cuál es el puerto por el que se transmite, lo que resulta importante es la conexión en los puertos de la línea de acople, pues según su configuración es posible contar con acoples seguidores o acoples en reversa. El acople en reversa utiliza el puerto adyacente al de entrada de la señal, como puerto de salida aislada y el diagonal como puerto de acople como se ve en la Figura 5b. Para el acople seguidor el puerto adyacente al de entrada es el de acople y el diagonal es puerto aislado como se ve en la Figura 5c. Esto para acoples con terminación de puerto aislado externo, ya que algunos acoples comerciales se encuentran por defecto con puerto aislado interno, el cual trae conectada una carga interna que limita la operación del puerto. De acuerdo a la necesidad se utiliza alguna de las configuraciones mencionadas, por ejemplo para las guías de onda es recomendable el uso de acoples seguidores.

### Características generales de acoples direccionales [21].

Para la explicación de las características generales se utiliza el acople en reversa, por lo cual la notación utilizada corresponde al del acople en reversa de la Figura 5b.

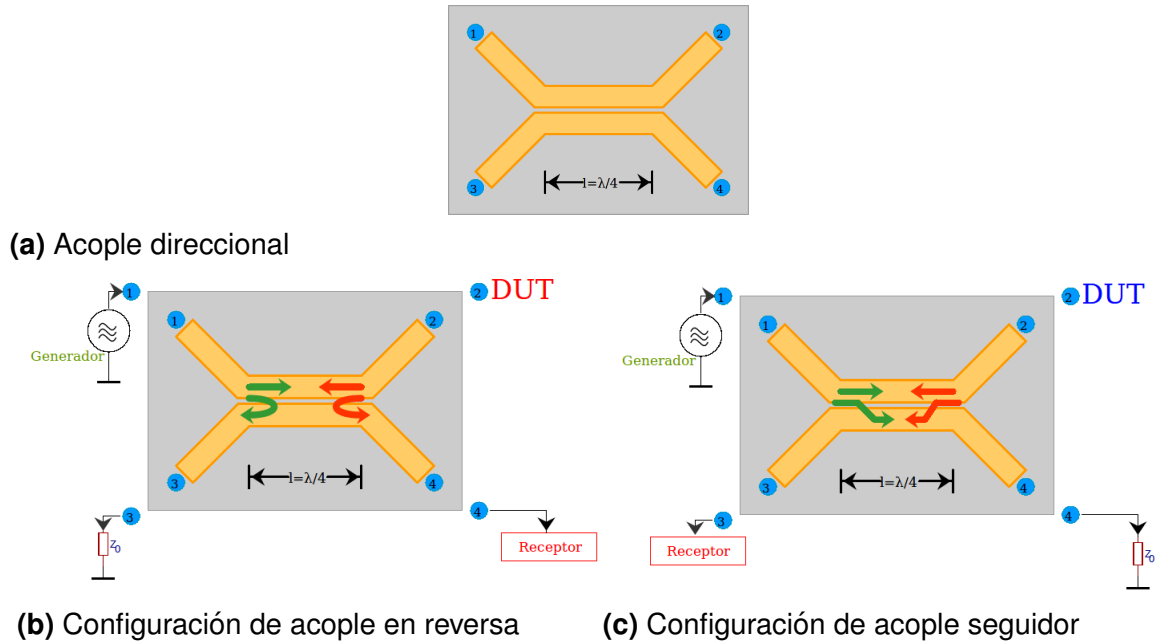
- Factor de acoplamiento (*Coupling Factor*).

Es la relación existente entre la potencia entregada por el puerto acoplado ( $P_4$ ) con respecto a la potencia del puerto de entrada de la señal ( $P_1$ ) y se calcula como:

$$\text{Coupling factor}(dB) = -10 \log(P_4/P_1) \quad (1.5)$$

representa la capacidad de acoplamiento existente entre las líneas del acople direccional, en el cual influye la separación de las pistas y la frecuencia de operación del dispositivo, por lo cual el factor de acoplamiento varía con respecto a la frecuencia bajo la que se encuentra operando, pues resulta imposible teóricamente implementar un acople direccional con respuesta plana

**Figura 5:** Acople direccional y configuraciones (a) Acople direccional (b) configuración en reversa y (c) configuración en seguidor.



para toda la banda de frecuencia de trabajo.

- Pérdidas (*Loss*).

El factor de la pérdidas en la línea de principal (*puerto<sub>1</sub> – puerto<sub>2</sub>*) se presenta por la acumulación de pérdidas del acoplamiento, pérdida dieléctrica, pérdida del conductor y pérdida de *VSWR*. El factor de pérdida depende del rango de frecuencia y se vuelve menos significativa cuando el factor de acoplamiento es alto. El factor de *insertion loss* está dado por la ecuación 1.6.

$$Insertionloss(dB) = 10 \log(1 - P_4/P_1) \quad (1.6)$$

- Aislamiento (*Isolation*)

Existen dos factores de aislamientos, uno dado por la diferencia de los niveles de potencia de las señales del puerto de entrada ( $P_1$ ) y el puerto aislado ( $P_3$ ) cuando se encuentran conectados a cargas combinadas. El otro está dado

por los dos puertos de salida ( $P_2$  y  $P_4$ ), utilizando uno de estos como entrada y el otro como salida mientras  $P_1$  y  $P_3$  terminan con cargas combinadas. Este factor se calcula mediante la ecuación 1.7 para el primer caso y con la ecuación 1.8 para el segundo. Un aislamiento considerable permite buenos acoplamientos.

$$Isolation(dB) = -10 \log(P_3/P_1) \quad (1.7)$$

$$Isolation(dB) = -10 \log(P_4/P_2) \quad (1.8)$$

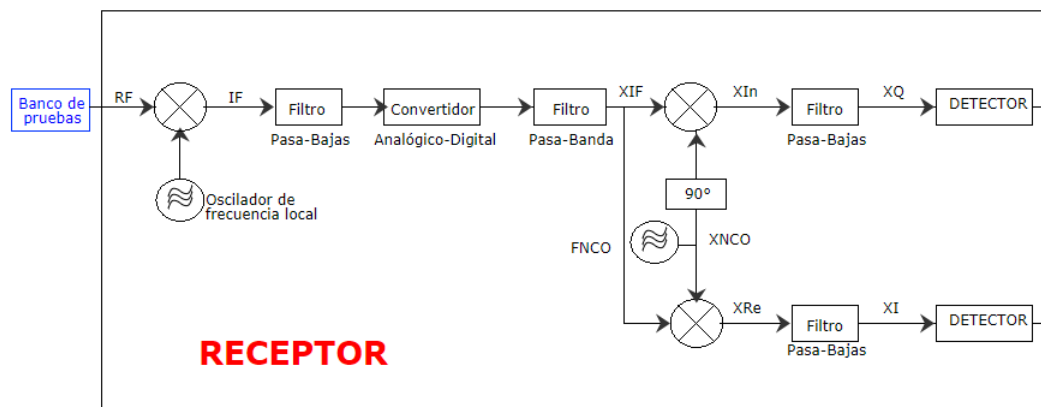
- Directividad (*Directivity*)

Se relaciona con el aislamiento y es la diferencia entre los puertos de acoplamiento  $P_4$  y el aislado  $P_3$  como se ve en la ecuación 1.9. No es posible medir este parámetro de forma directa, por lo cual se realizan las mediciones de aislamiento y acoplamiento, es decir, que la directividad está dada como:  
 Directividad (dB) = Aislamiento (dB) - Acoplamiento (dB).

$$Directivity(dB) = -10 \log(P_3/P_4) = -10 \log(P_3/P_1) + 10 \log(P_4/P_1) \quad (1.9)$$

### 1.3 RECEPTOR

Figura 6: Receptor de VNA.



La etapa receptora del VNA se fundamenta en el principio heterodino, el cual convierte una señal de  $RF$  recibida, en una señal de frecuencia intermedia  $IF$ , mediante la mezcla de frecuencias de dos señales, permitiendo mejor su implementación (filtrado y amplificado); por consiguiente, en este sistema se traslada la frecuencia central de la señal recibida ( $f_{RF}$ ) a una frecuencia intermedia ( $f_{IF}$ ), antes de pasar a la parte del procesamiento de la señal, con el fin de llevar la señal RF a banda base al multiplicarla con la señal local que tiene frecuencia ( $f_{LO}$ ), obteniendo también características como la disminución del ruido y el fácil filtrado de la señal, seguidamente se aplica un filtro pasa bajas que deja pasar la señal con frecuencia  $IF$ , obtenida por el mezclador. La frecuencia intermedia ( $f_{IF}$ ), es la frecuencia de la señal  $IF$  dada por  $f_{IF} = |f_{RF} - f_{LO}|$ . Seguidamente se pasa la señal por un filtro pasa bajas para eliminar el ruido teniendo la señal acondicionada para pasarla al procesamiento digital ( $DSP$ ). En el  $DSP$  se realiza una conversión Análogica/Digital y se filtra nuevamente con un filtro pasa bandas con frecuencia central ( $f_{IF}$ ), de esta manera se obtiene una señal  $X_{IF}$  que se mezcla con una señal generada por un oscilador controlado numéricamente ( $NCO$ ) de frecuencia  $f_{NCO}$  con el fin de llevar a ( $f_{IF}$ ) a una frecuencia igual a cero y realizar una modulación  $I \setminus Q$ . En el proceso se utilizan dos multiplicadores que se encargan de multiplicar a  $X_{IF}$  con  $X_{NCO}$  y  $X_{NCO}$  desplazada  $90^\circ$  [reflibro]. El diagrama de bloques que representa este proceso se ve en la Figura 6.

Conocida la estructura y las características del receptor se describen los componentes que constituyen y permiten obtener las características del receptor. Los *Down – converters* digitales ( $DDC$ ) se encargan de trasladar la banda de frecuencias recibida para realizar la respectiva modulación y recuperar toda la información. Éste se encuentra compuesto por el oscilador controlado numéricamente, por un mezclador y un filtro pasa bajas. El oscilador controlado numéricamente ( $NCO$ ) es el componente que permite la transferencia a banda base de la información recogida para ser demodulada. Este oscilador genera una señal sinusoidal muestreada (por sucesión de muestras), que, al mezclarse con la señal recibida, causa el desplazamiento a las frecuencias deseadas en el espectro. El desplazamiento que se produce se realiza con respecto a la frecuencia de la señal portadora. Para tener

un diseño apropiado del *NCO* se debe tener especial precisión y exactitud con la señal sinusoidal, de no ser así, es posible una degradación de la relación señal a ruido e incluso tener falsos espectros en el dominio de la frecuencia. Consiste en un sintetizador de frecuencia basado en un acumulador de fase y una conversión de ángulo a la forma de onda. El mezclador del receptor se encarga de multiplicar la señal recibida por la señal que genera el oscilador controlado, buscando la modificación de la banda de frecuencia del espectro de la señal. Su implementación se puede realizar utilizando dos multiplicadores uno para función seno y otro para la función coseno. La señal recibida en el mezclador se multiplica por:

$$\cos(t) - j \sin(t) = e^{-j\omega t} \quad (1.10)$$

Generando un fasor complejo el cual produce la rotación de la señal recibida por un ángulo de fase que se modifica continuamente. Siguiendo el esquema del receptor la etapa que sigue es la del filtrado en donde después de mezclada la señal se filtra para eliminar todas las frecuencias innecesarias y conservar el espectro o la banda de frecuencias de interés. El filtro a trabajar debe ser de un orden elevado y de banda estrecha especialmente para eliminar todas las componentes que no se necesiten. Esto llega a tener una responsabilidad entre la relación SNR y el orden del filtro [6].

#### 1.4 PROCESADOR

Este componente del *VNA* esta dedicado a realizar las operaciones para determinar principalmente los parámetros-*S* a partir de las amplitudes de las señales entregadas por el receptor, arrojando los datos en forma de valores numéricos complejos para ser vistos desde el software del dispositivo. Está constituido por dos componentes: Un convertidor análogo-digital (*ADC*) y un procesador digital de señales (*DSP*). El *ADC* se encarga de convertir las señales que salen del receptor en señales digitales, con el fin de que éstas sean compatibles con el *DSP*. El *DSP* es el componente principal del procesador debido a que es el encargado de realizar las operaciones

principales que dan como resultado los parámetros- $S$ .

El propósito del procesador del VNA es determinar la magnitud y la fase de los parámetros *scattering*; pero para obtener estos resultados el *DSP* debe estar compuesto de operaciones ya sean algebraicas o de procesamiento digital de señales; que logren hallar las variables esperadas. Un procesador tiene muchas formas o maneras diferentes de diseñarlo, debido a que cada persona tiene un pensamiento distinto pero lo importante es que el resultado que se obtenga debe ser igual para todos, además también viene ligado el componente del generador.

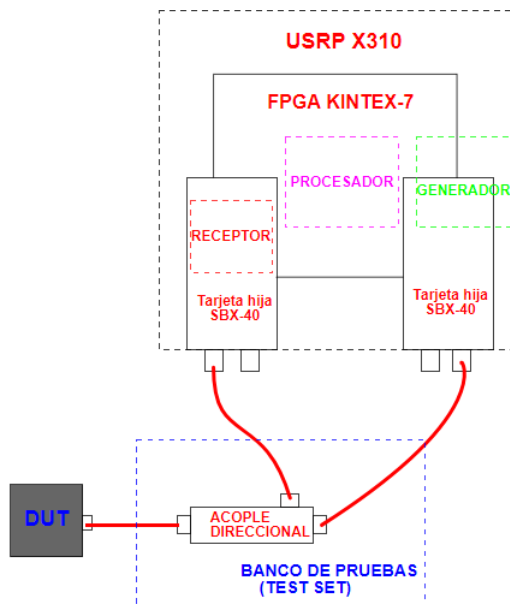
---

## 2. SÍNTESIS DE UN VNA SOBRE UN USRP

---

Con base en la descripción de los componentes del *VNA* y su principio de funcionamiento, se realiza un análisis general para determinar si es posible la implementación de un *VNA* en el *USRP X310*. Se proponen algunas alternativas para dicho fin y las implicaciones que éstas conllevan. Posteriormente se profundiza en una de estas alternativas como opción para realizar la implementación de cada componente de un *VNA* en un *USRP*. En la Figura 7 se presenta un breve esquema del *USRP* en el que se resaltan sus tarjetas hijas y madre (*FPGA*). Sobre el mismo se superponen bloques de las etapas del *VNA* que representan sobre que componente del *USRP* se desarrollan estas etapas. Se presenta un *DUT* de un puerto y la respectiva conexión utilizando un acople direccional como alternativa para el banco de pruebas. El receptor se encuentra en la tarjeta hija, el generador de *RF* comparte la etapa de transmisión de la misma tarjeta y se configura desde la *FPGA*. EL procesador se implementa sobre la *FPGA* y este va ligado a la forma de la generación de la señal.

**Figura 7:** VNA sobre el USRP X310.

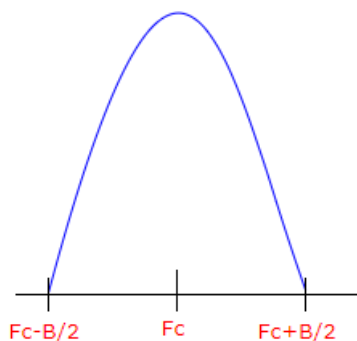


A continuación, se presentan algunas propuestas de señales para realizar el análisis y finalmente se desarrolla una de estas.

### **Pulso gaussiano:**

Consiste en realizar una transmisión de una campana de Gauss como la de la Figura 8, la cual permite incluir información de magnitud y fase para cada componente de frecuencia incluida dentro de la campana gaussiana. Sin embargo, para realizar un análisis en rangos de frecuencia amplios como los analizados en este trabajo en el rango de GHz, se requiere enviar múltiples impulsos gaussianos variando la frecuencia central dado que las tarjetas hijas compatibles con los *USRP X310* cuentan con anchos de banda de 40 MHz y 120 MHz. Además, la etapa de procesamiento requiere implementar la Transformada Rápida de Fourier (*FFT*) con el fin de determinar la amplitud y fase para cada componente de frecuencia tanto de transmisión como de recepción.

**Figura 8:** Impulso Gaussiano.

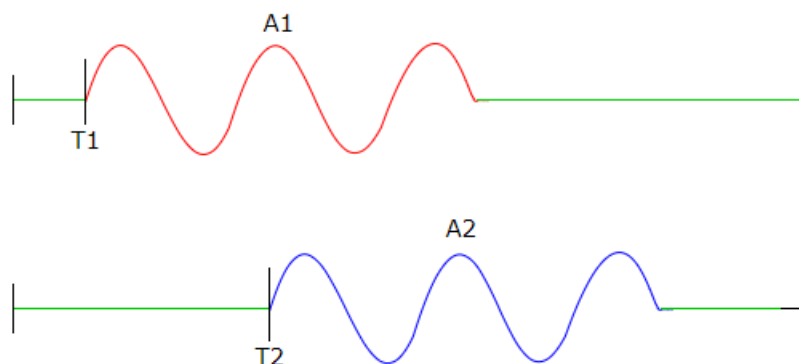


### **Sinusoidal troncada:**

Se basa en la generación de una señal sinusoidal troncada en  $N$  periodos como se presenta en la Figura 9, para un análisis en el dominio del tiempo, con el fin de determinar el desfase presentado por medio del diferencial de tiempo ( $T_2 - T_1$ ) y la magnitud del parámetro-S mediante la división de las amplitudes  $A_2/A_1$ , que se introducen por el DUT y la conexión. Esta alternativa requiere precisión en la frecuencia de muestreo pues se pueden presentar errores al determinar los cambios

de la señal (paso de señal en cero a la variación de la señal sinusoidal y el valor máximo de amplitud de la misma).

**Figura 9:** Sinusoidal truncada.



#### Transmisión de fase de referencia:

La modulación y demodulación en cuadratura puede presentar desviaciones de frecuencia si no cuentan con un sistema de sincronización en sus VCOs, lo cual introduce variaciones entre la señal modulada y la de modulada, además estas introducen fases en el proceso de conversión de  $IF$  a  $RF$  como se ve en la ecuación 2.1. La cual representa la señal  $RF$  transmitida por el  $USR P$  con fase  $\theta_0$  agregada en el proceso de modulación.

$$S_i(t) = x(t) \cos(2\pi f_0 t + \theta_0) + y(t) \sin(2\pi f_0 t + \theta_0) \quad (2.1)$$

Al pasar esta señal por el  $DUT$  este introduce variaciones de amplitud y fase a la señal incidente. Y estas variaciones son las que se requieren determinar para el cálculo de los parámetros-S. La ecuación 2.2 representa la señal reflejada por el  $DUT$ , en donde el  $t_0$  representa el desfase y  $\alpha$  la variación de amplitud que genera el dispositivo bajo prueba a la señal incidente de la ecuación 2.1.

$$S_{DUT}(t) = \alpha[x(t - t_0) \cos(2\pi f_0(t - t_0) + \theta_0) + y(t - t_0) \sin(2\pi f_0(t - t_0) + \theta_0)] \quad (2.2)$$

A partir de esta ecuación se analiza que señales  $x(t)$  y  $y(t)$  facilitaran el proceso para determinar las variaciones introducidas por el  $DUT$ . Una propuesta se muestra en la ecuación 2.3. Con lo cual se busca enviar una señal con fase de referencia y mediante esta, calcular la fase introducida por el sistema, descartando la variación que introduce el diferencial de frecuencia.

$$x(t) = A + \cos(2\pi f_m t) \quad (2.3)$$

$$y(t) = 0$$

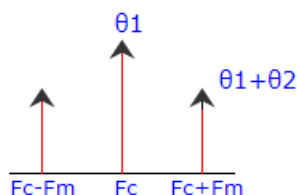
Con estas señales finalmente en la recepción se obtiene la señal representada por la ecuación 2.4. En donde  $2\pi \Delta f$  representa la desviación de frecuencia obtenida por la des sincronización de los VCOs en las etapas de modulación y demodulación y el  $\theta_1$  son los desfases introducidos en el proceso. Con estas señales finalmente en la recepción se obtiene la señal representada por la ecuación 2.4. En donde  $2\pi \Delta f$  representa la desviación de frecuencia obtenida por la des sincronización de los VCOs en las etapas de modulación y demodulación y el  $\theta_1$  son los desfases introducidos en el proceso.

$$S_{rec}(t) = \alpha[(A + \cos 2\pi f_m(t - t_0)) \cos(\theta_1 - 2\pi \Delta f t) + j(A + \cos 2\pi f_m(t - t_0)) \sin(\theta_1 - 2\pi \Delta f t)] \quad (2.4)$$

En el dominio de la frecuencia esto se representa como un impulso en la frecuencia central ( $F_c$ ) dado por el offset ( $A$ ) y dos simétricos resultantes de la señal coseno, como se puede ver en la Figura 10. El impulso generado por  $A$  contiene la fase  $\theta_1$  introducida por el sistema de modulación y demodulación y los impulsos de la señal coseno contienen la fase de todo el proceso (modulación, demodulación y

variaciones introducidas por el *DUT*) por lo cual es posible determinar la fase  $\theta_2$  generada por el *DUT*, calculando las fases de los impulsos y restandolas.

**Figura 10:** Fase de referencia.



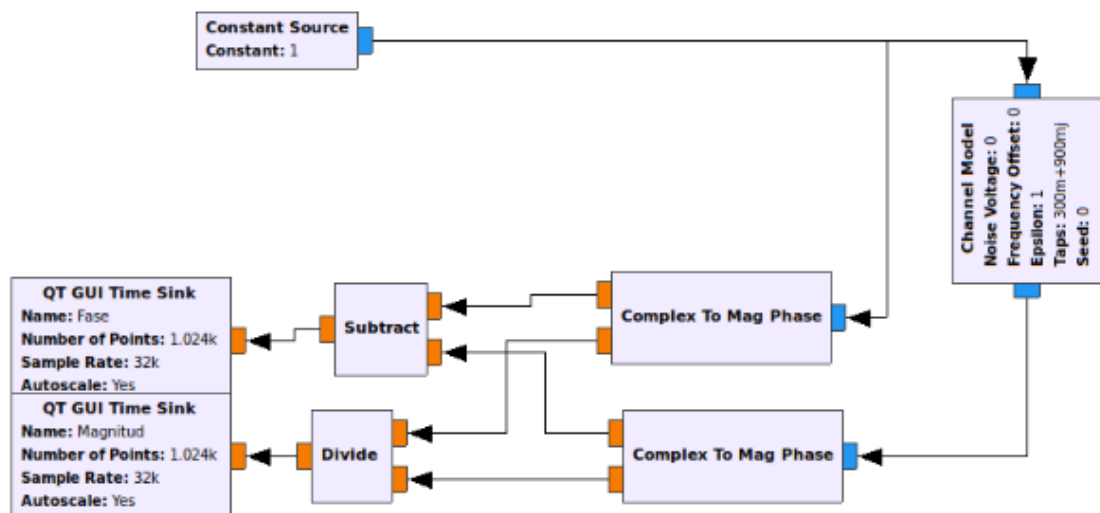
**Pulso de *RF*:** Esta opción consiste en implementar un generador de *RF* y realizar un barrido por las frecuencias en las que se requiere el análisis del *DUT*. En la generación de las señales de *RF* se utiliza una constante como señal de referencia para compararla con la obtenida por la etapa de recepción de *RF*. El análisis se resume a operaciones algebraicas entre constantes complejas.

Esta es la opción que se desarrolla a continuación dado que es el principio de funcionamiento básico de un *VNA* y su construcción sirve como base a algunas de las propuestas presentadas anteriormente.

En la Figura 11 se presenta un diagrama de bloques de GNU RADIO en el que se modela esta propuesta. Esta consiste en generar constantes que al pasar por la etapa de transmisión del *USR*P modula la amplitud de la señal de *RF*, con el bloque *Cannel Model* se modelan los efectos que introduce el sistema de modulación y demodulación. Con las constantes complejas de las etapas de transmisión y recepción se realiza el procesamiento para determinar la magnitud y la fase del parámetro a medir. El banco de pruebas, dada su ubicación entre las etapas de transmisión y recepción, acorde a su función, es un dispositivo que se debe agregar como auxiliar externo al *USR*P y que por ende no se presenta en el diagrama.

Los *USR*P cuentan con una etapa de transmisión y recepción de *RF*, incorporada en las tarjetas hijas. Para un *USR*P X310 existe la posibilidad de cambiar estas tarjetas de acuerdo a la aplicación y el rango de frecuencias de operación a trabajar.

Figura 11: Propuesta implementación de un VNA en un USRP desde GNU Radio.



Esta plataforma *SDR* permite placas secundarias de los modelos *WBX* [10], *SBX* [7] y *CBX* [6] que tienen como principal característica su ancho de banda, con un filtro banda base que puede ser de 120 MHz o 40MHz. Las principales diferencias entre las tres series es el rango de frecuencia de trabajo y la sincronización de fase. Para este trabajo se utilizó la tarjeta *SBX – 40* dado que presenta un mayor rango de operación (400 MHz a 4400 MHz) y cuenta con sincronización de fase mientras que la *WBX* tienen un desfase de  $180^\circ$  y la *CBX* no trabaja con sincronización.

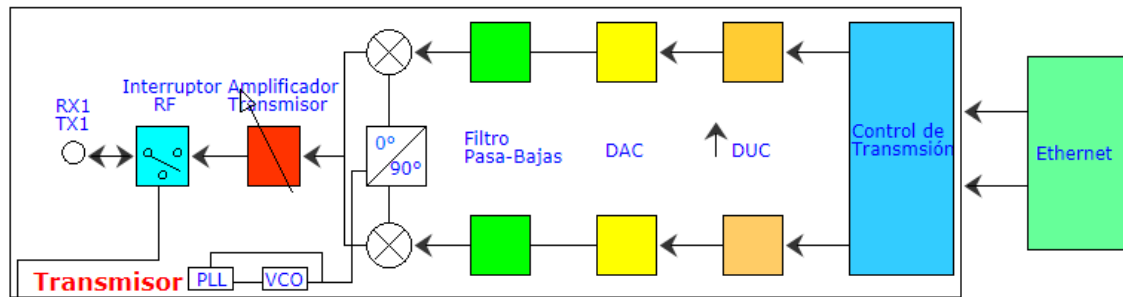
## 2.1 GENERADOR DE RF EN USRP

En la construcción de la etapa del generador de RF se realizó un estudio de la etapa de transmisión de la tarjeta *SBX – 40*, presentada en la Figura 12 la cual se encuentra constituida con los siguientes circuitos:

- Sintetizador de banda ancha con VCO integrado – ADF4350.,

El ADF4350 es un sintetizador de *PLL* con *VCO* integrado, frecuencia de referencia y filtro de bucle externo. Cuenta con circuitos que permiten dividir su frecuencia de salida fundamental, la frecuencia fundamental oscila entre 2200 MHz a 4400 MHz [2]. Algunas características importantes de este circuito se

Figura 12: Transmisor del *USRP X310*.



presentan en la Tabla 3. La frecuencia de salida del ADF4350 es la frecuencia de modulación utilizada por el modulador en cuadratura ADL4375-05.

Tabla 3: Características del sintetizador ADF4350.

Características	Valor
Rango de frecuencia de salida	137.5 MHz - 4400 MHz
Rango de frecuencia de entrada	10 MHz - 200MHz
Actuación ante el ruido	-89 dBc/Hz a -145 dBc/Hz. varia con la frecuencia de la portadora
Divisores de frecuencia	1, 2, 4, 8 y 16

- Filtro pasa bajas - LFCN-1200.

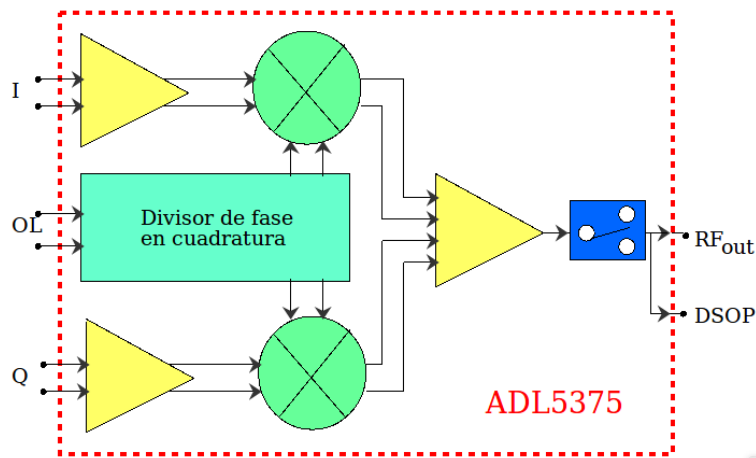
El LFCN-1200 es un filtro pasa bajas con frecuencia de corte de 1200 MHz, este se encuentra conectado entre la salida del DAC y la entrada del modulador en cuadratura. Puede presentar atenuación 1 dB en su banda de operación y hasta 30 dB para frecuencias cercanas a 5000 MHz [18]. Presenta impedancia de salida de 50  $\Omega$ .
- Modulador de cuadratura de banda ancha – ADL5375-05.

El ADL5375 es un modulador de cuadratura con rango de operación desde 400 MHz a 6 GHz. Presenta gran precisión de fase y estabilidad de amplitud. Su respuesta en frecuencia ubica la banda plana en el rango de 450 MHz a 3.5 GHz con atenuación de 1 dB y su impedancia de salida también es de 50  $\Omega$  [1].

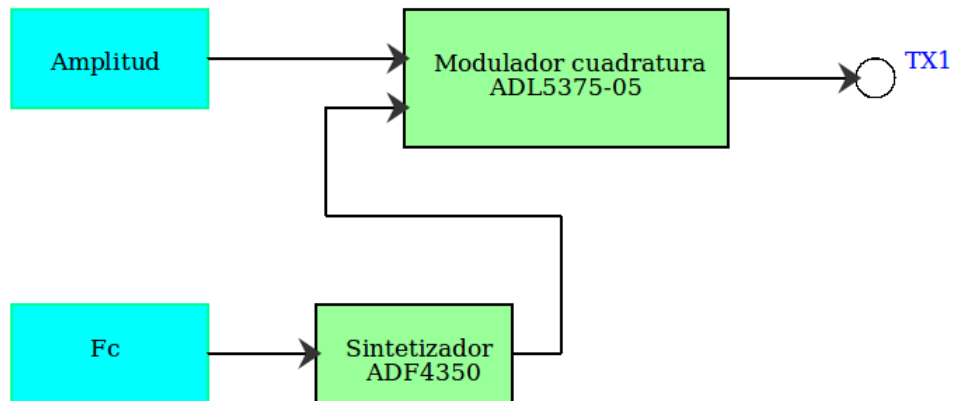
La estructura de los moduladores en cuadratura se presenta en la Figura 13 y la ecuación que representa la señal transmitida es la Ecuación 2.5. En donde I y Q son las señales de entrada a modular en cuadratura, OL es el oscilador local proveniente del ADF4350 y  $RF_{out}$  es la señal de salida en RF; además el ADL5375 cuenta con un pin de salida (DSOP) que deshabilita o aísla el circuito del puerto de salida en los instantes que no se desee transmitir.

$$s(t) = I(t) \cos(2\pi f_0 t) + Q(t) \sin(2\pi f_0 t) \quad (2.5)$$

**Figura 13:** Diagrama de bloques funcional del ADL5375.

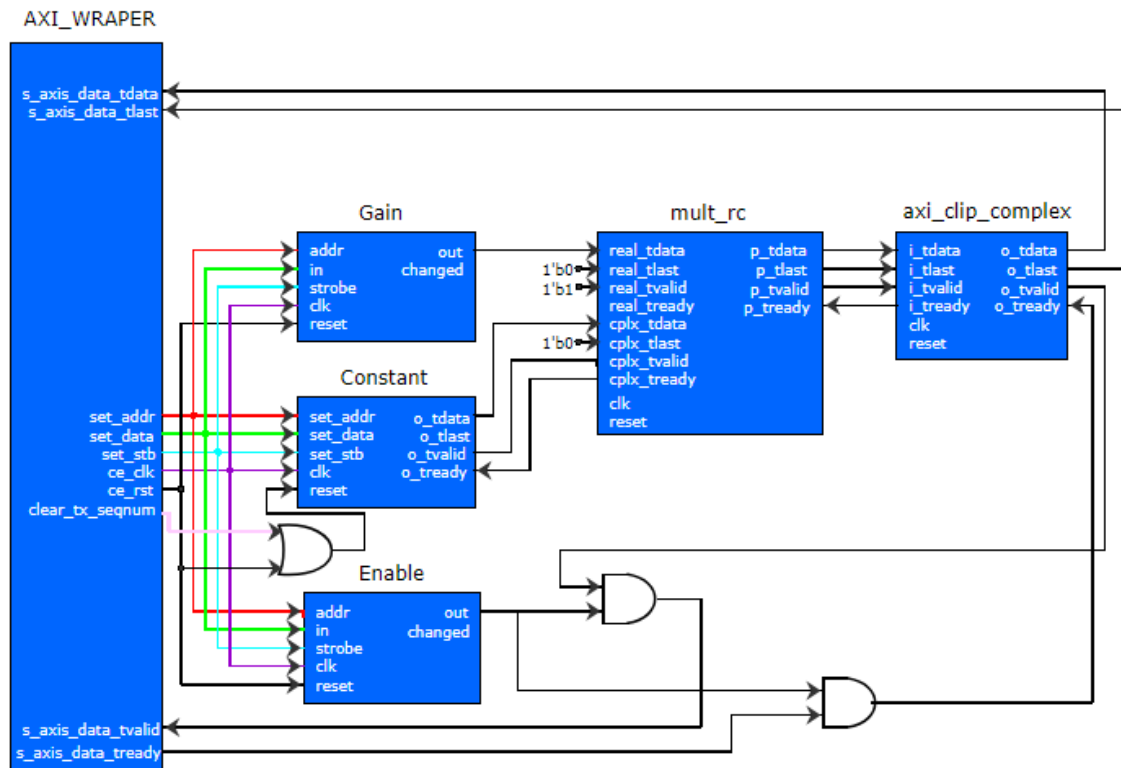


Contando con este sistema y su similitud en cuanto a la estructura del generador de un VNA (rango de frecuencias, PLL y VCO). Se planteó la estructura de la Figura 14 para la construcción del generador de RF, la cual presenta un diagrama de bloques de esta etapa, en donde los parámetros para la configuración de esta fueron la frecuencia central de modulación o portadora ( $F_c$ ) y cumpliendo con el objetivo de implementar un generador de señales sinusoidales de RF, el mensaje a modular es un valor constante, el parámetro *Amplitud*. A partir de esta estructura se procedió a la construcción del mismo sobre RFNoC. A partir de esta estructura se procedió a la construcción del mismo sobre RFNoC, el cual se explica con detalle en la subsección 2.1.1.

**Figura 14:** Diagrama de generador de  $RF$  en el  $USRP$  X310.

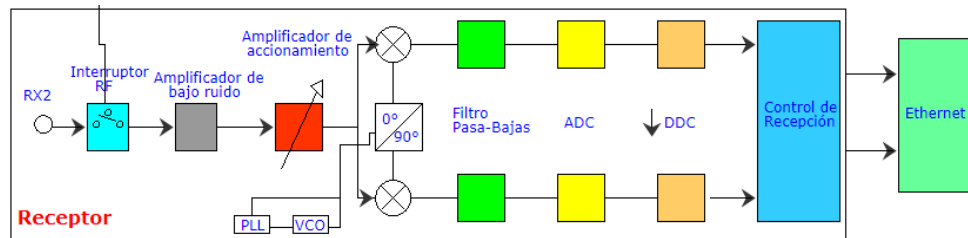
**2.1.1 Generador en  $RFNoC$**  La etapa del generador implementada en  $USRP$  con  $RFNoC$  se estructuró mediante dos bloques del entorno. El bloque  $RFNoC$  : RADIO, el cual permite acceder a las tarjetas hijas a través de  $UHD$  para configurarlas como transmisor o receptor, la frecuencia central de operación, ganancia de las etapas, entre otros parámetros. Este bloque cuenta con un puerto que se configura como *sink* si se desea utilizar para transmitir y como *source* para recibir. Para ésta etapa se configuró como transmisor, permitiendo el control del sintetizador ADF4350 y el modulador ADL5375-05. Al puerto se conectó el bloque encargado de suministrar el valor de la constante. El bloque  $RFNoC$  : RADIO se incluye por defecto al sintetizar las imágenes *.bit*, el bloque del mensaje se diseñó e implementó sobre éste entorno. En la Figura 15 se presenta el diseño del bloque Amp, complementario en la construcción de la etapa del generador. En este bloque se utiliza la interfaz  $AXI$ , con la cual se construyen los bloques  $RFNoC$  y cuenta con tres parámetros: *Enable Constant* y *Gain*. El parámetro *Enable* permite hacer una función on/off o start/stop para habilitar o deshabilitar este componente. Con el parámetro *Constant* se introduce el valor de la constante o amplitud a modular y *Gain* permite configurar el bloque de ganancia digital. Los bloques *mult\_rc* y *axi\_clip\_complex* son módulos que constituyen el bloque *Digital Gain* de  $RFNoC$  que se encuentran disponibles a los usuarios del entorno.

Figura 15: Diagrama Amp.



Se describió este bloque en las plantillas de los niveles *FPGA*, *UHD* y *GNU*; acorde a la estructura del entorno RFNoC [14].

**2.1.2 Receptor en el USRP** Este componentes en el *USRP* se localiza en las placas hijas constituyendo la etapa receptora mencionada, que a través del demodulador de cuadratura lleva la señal recibida de *RF* a *IF* como se puede observar en el esquema general del receptor del *USRP* (ver Figura 16), debido a esto, se facilita el proceso del análisis de esta etapa, puesto que el receptor viene integrado al *USRP* y sólo es necesario ajustar a las características que presenta el receptor del *VNA*. Al igual que en la etapa de transmisión, el bloque *RFNoC : RADIO* permite el control de esta etapa configurándolo como *source* y adicionalmente es necesario la conexión en serie de un bloque *RFNoC : DDC* (un bloque digital down converter propio del entorno *RFNoC*).

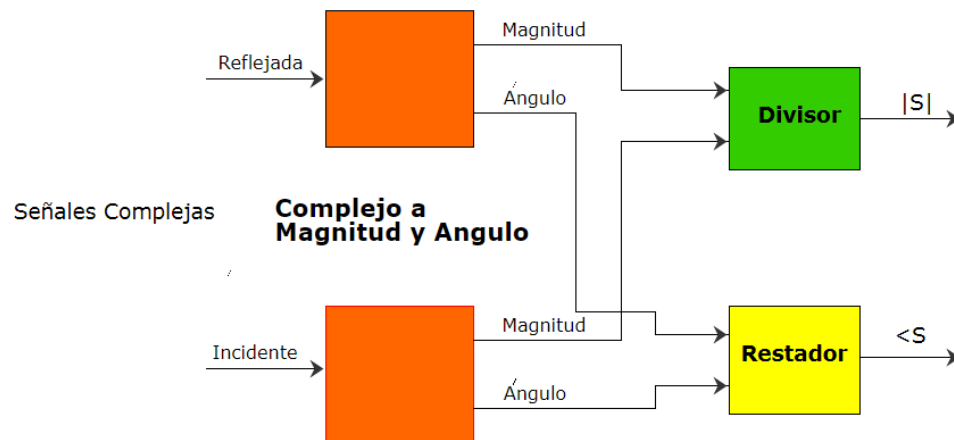
Figura 16: Receptor de la tarjeta hija *SBX* – 40

A continuación, se mencionan los elementos de la etapa receptora que contiene la *SBX* – 40:

- Controlador ADC diferencial de realimentación de corriente – ADA4927 [5].
- Demodulador de cuadratura – ADL5380 [4].
- Amplificador – AD8591 [3].
- Sintetizador de banda ancha con VCO integrado – ADF4350.
- Filtro pasa bajas – LFCN-1200.

**2.1.3 Procesador con RFNoC** En la figura 17 está representado el diagrama de bloques que compone el procesador, asumiendo que la etapa receptora está recibiendo señales constantes complejas. Para implementar este *DSP* en el *USRP* se requiere realizar el bloque en *RFNoC* que contengan las instancias de los bloques de complejo a magnitud y fase, del divisor y del restador. El entorno *RFNoC* trae en sus plantillas ya los bloques de complejo a magnitud y fase, y el restador, sin embargo, estos se pueden generar a partir de los catálogos IP *core* que presentan Vivado XILINX como se puede apreciar a continuación.

Figura 17: Diagrama de bloques del procesador.



**Adder/Subtractor** Xilinx LogiCORE en su IP Catálogo contiene un módulo Adder/Subtractor que permite la creación de bloques sumadores o restadores configurables, que operan datos con signo o sin signo. Pueden ser implementados usando la opción de fábrica donde utilizan *LUT* y *FFs* o por *DSP48* [28].

**Divider Generator XILINX IP-Core** El Divider Generator de XILINX logiCORE es un circuito para la división de enteros basado en la división Radix-2, o la división por el algoritmo High Radix que utiliza segmentos DSP a un rendimiento menor, mientras que el algoritmo Radix-2 utiliza la lógica *FPGA* logrando un rango de opciones de rendimiento además de ser ciclo único [29].

Cada uno de los tres algoritmos con los que cuenta la implementación de la división sirven para ciertas características específicas. El LUTMult utiliza segmentos *DSP*, bloques *RAM* y cantidades pequeñas de lógicas *FPGA* (registro y *LUT*). De uso recomendable para operaciones con ancho de bits menores o iguales a 12, utilizando menos recursos a diferencia de Radix-2 por el uso de *DSP* y bloque *RAM*. El algoritmo Radix-2 utiliza operandos enteros por lo cual permite generar resto fraccionario o un entero, se recomienda para anchos de operandos de menos de 16 bits

o para aplicaciones con alto endimamiento, a diferencia del algoritmo LUTMult, éste sólo utiliza implementación lógica *FPGA* (registros y *LUT*). Por último el High Radix con preescalado, utiliza segmentos *DSP* y bloques *RAM*, usado principalmente para anchos de operandos superiores a 16 bits.

Se realizaron unas pruebas de validación del generador con las que se buscó comprobar que un *USRP X310* permite implementar este tipo de dispositivo con similitud a uno comercial. Se sintetizaron imágenes *.bit* con el objetivo de conocer los recursos *FPGA* consumidos por los bloques que constituyen esta propuesta planteada. Dado que los resultados de estas pruebas permiten determinar la viabilidad de implementar un *VNA* y la factibilidad de implementar alguna de las otras propuestas que consumen más recursos dada la complejidad en la etapa de procesamiento y generación de la señal. Se caracterizó un acople direccional presentado como alternativa para el banco de pruebas porque es necesario conocer los parámetros específicos para posteriores calibraciones al implementar el *VNA*.

---

## 3. PRUEBAS Y RESULTADOS

---

En este capítulo se presentan los resultados obtenidos en la prueba del generador de *RF* implementado sobre un *USRP X310* con *RFNoC* para verificar el funcionamiento de la máquina computacional, también se listan los recursos utilizados por el bloque complementario en la construcción de esta y los posibles recursos utilizados por la etapa del procesador; los resultados de una calibración de un acople direccional como propuesta para la implementación del banco de pruebas, puesto que no es posible la descripción de este sobre una plataforma *SDR*.

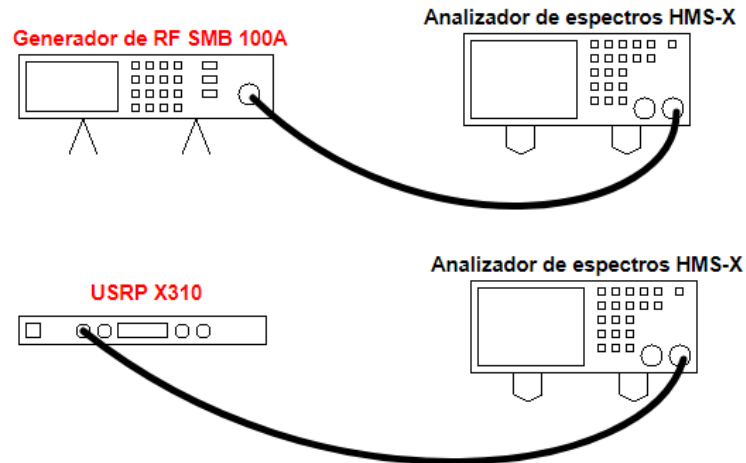
### 3.1 CALIBRACIÓN DEL GENERADOR *RFNoC*

Para la calibración se utilizaron los siguientes dispositivos:

- Analizador de espectros (*spectrum analyzer - SA*) HMS - X de *Rohde&Schwarz* [25].
- Generador de *RF* SMB 100A *Rohde&Schwarz* [26].
- *USRP X310*.
- Acople direccional Keysight 87300B.

Se realizó la conexión de los equipos como se muestra en la Figura 18. Con el generador de *RF* SMB 100A se realizó un barrido en el rango de 400 MHz a 3 GHz con pasos de 100 MHz, generando señales de radio frecuencia con potencia de 0 dBm. Con el *SA* se registraron los valores de potencia. Posteriormente se generaron señales en el mismo rango de frecuencias y con el mismo paso, desde el *USRP* y con los bloques *RFNoC* que constituyen la etapa del generador de *RF* para la implementación del VNA. Los datos de potencia obtenidos también se obtuvieron con el *SA*.

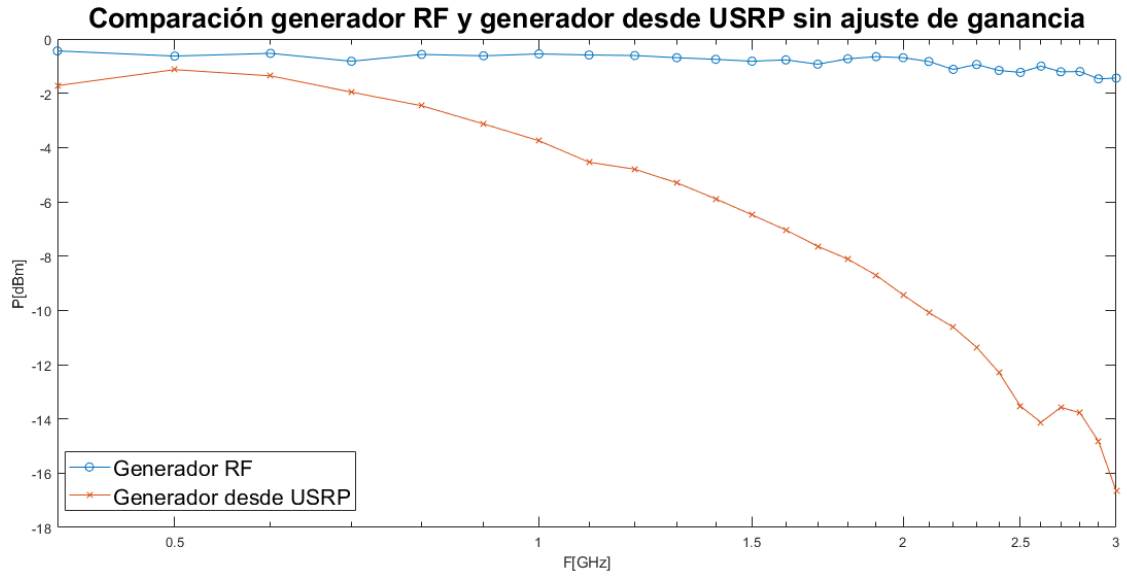
**Figura 18:** conexión de los equipos para la prueba de verificación del generador de *RF* implementado en un *USRP*.



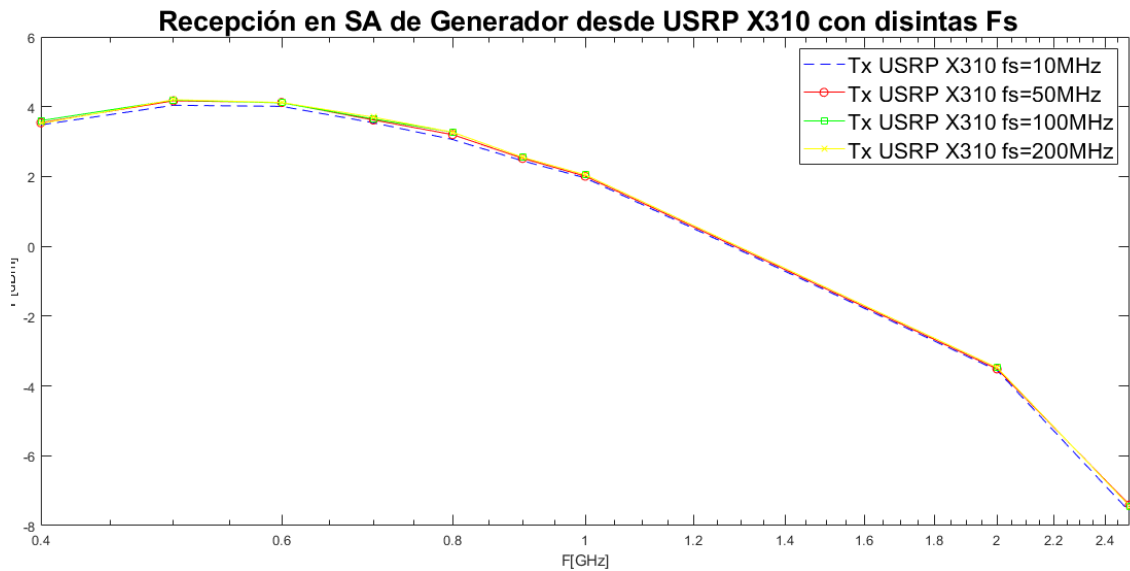
Con los valores de potencia recibidos en el *SA* a partir del generador de *RF* y del *USRP* se trazaron las gráficas de la potencia, presentadas en la Figura 19 en la que se evidencia que el generador de *RF* presenta un comportamiento casi constante en el rango de frecuencias utilizado en la prueba, con potencias alrededor de  $-0.85$  dBm con variaciones de  $\pm 0.45$  dBm mientras que la respuesta del generador implementado en el *USRP* presenta una curva decreciente comprendida en un rango de  $-1.13$  dBm a  $-16.66$  dBm.

La anterior prueba de calibración se realizó con una frecuencia de muestreo de 10 MHz, esta frecuencia de muestreo es un parametro de configuración del Bloque *RFN*<sub>o</sub>*C* : RADIO por lo cual se repitió la prueba ajustando la frecuencia de muestreo a 10 MHz, 50 MHz, 100 MHz y 200 MHz. Los resultados obtenidos se trazaron y se presentan en la Figura 20. Los resultados de esta prueba permiten determinar que la frecuencia de muestreo no influye en la respuesta que se está obteniendo del transmisor del *USRP* pues no se corrigió el efecto y las pequeñas variaciones entre las distintas frecuencias de muestreo se presentan por errores en la medición, debido a que no se toman todas al mismo tiempo.

**Figura 19:** Curvas de potencia de un generador de  $RF$  y del  $USRP$  como generador de  $RF$ .

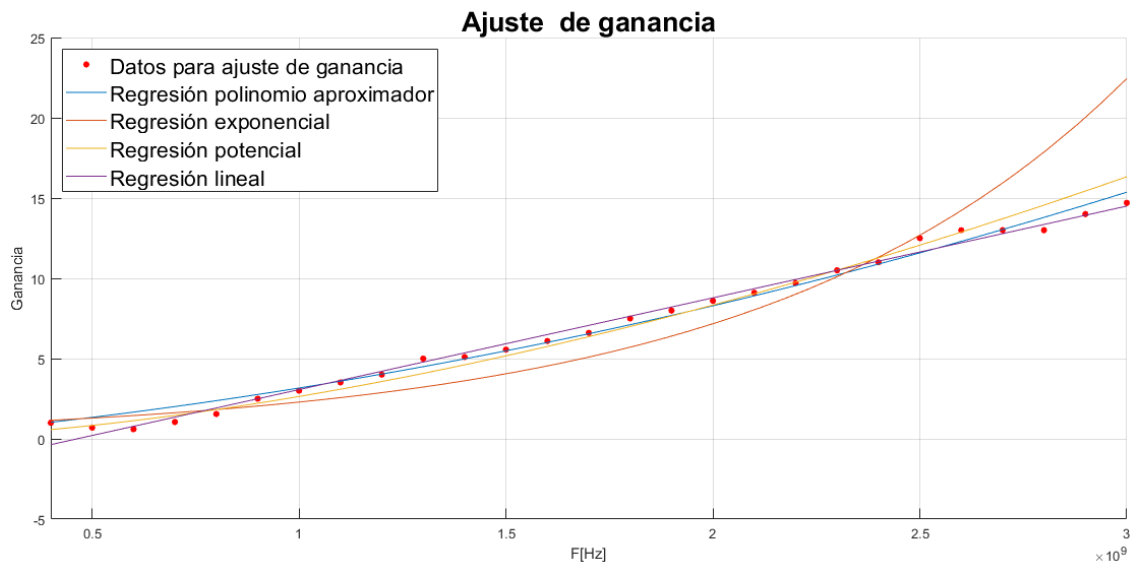


**Figura 20:** USRP como generador de  $RF$  a distintas frecuencias de muestreo ( $f_s$ ).



Se propuso agregar un bloque para la compensación de la curva de potencia de transmisión del *USRP* por lo cual se repitió la prueba con parámetros fijos: Amplitud = 1, Ganancia=0.5 (del bloque de la constante) y  $F_s=10$  MHz. El parámetro de ganancia del bloque *RFNoC* : RADIO se dejó variable con el fin de ajustar manualmente este y así corregir el error que se estaba presentando. Generando señales entre los 400 MHz y 3 GHz con pasos de 100 MHz se ajustó el parámetro de ganancia del bloque para cada una de estas señales, buscando obtener valores similares a los obtenidos por el generador de *RF*. Los valores de ganancia se registraron y con el uso de MATLAB a través de la función *polyfit* se generaron regresiones lineal, exponencial y polinómica, para determinar la curva que mejor se ajustara a los datos y así obtener una mejor compensación. En la Figura 21 se trazan los datos obtenidos al ajustar manualmente la ganancia con respecto a la frecuencia y las respectivas curvas de las funciones que se ajustan a los datos. Gráficamente evidencia una mejor aproximación con la regresión lineal, la polinómica y la potencial que con la exponencial.

**Figura 21:** Curvas para ajuste de ganancia para transmisión del *USRP* .



Las ecuaciones obtenidas por estas regresiones se presentan a continuación, la Ecuación 3.1 representa la aproximación polinómica, la 3.2 corresponde a la ex-

ponencial, la 3.3 a la regresión potencial y la 3.4 a la lineal. Estas se utilizan para ajustar la curva de ganancia, presentada en la Subseccion 3.2.

$$G(F_c) = 9,77441e^{-19} * F_c^2 + 2,1904e^{-09} * F_c \quad (3.1)$$

$$G(F_c) = e^{-0,31153+1,1408-09*F_c} \quad (3.2)$$

$$G(F_c) = 10^{-14,5061} * F_c^{1,6586} \quad (3.3)$$

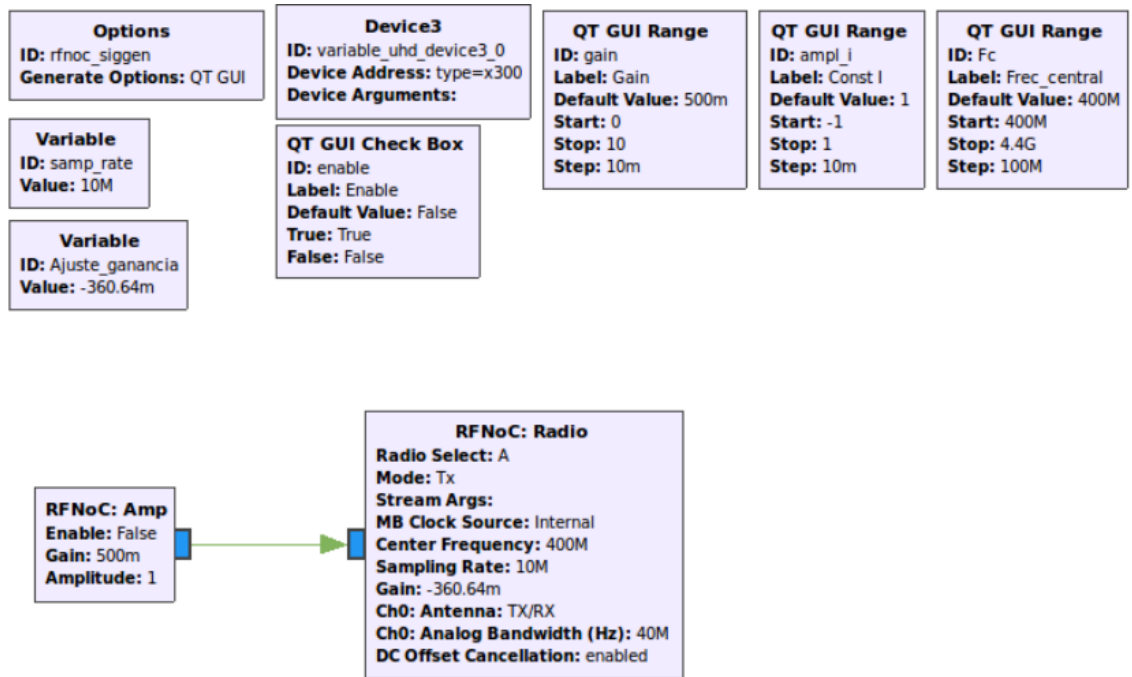
$$G(F_c) = 5,7154e^{-09} * F_c - 2,6468 \quad (3.4)$$

### 3.2 VALIDACIÓN GENERADOR DE RF EN USRP.

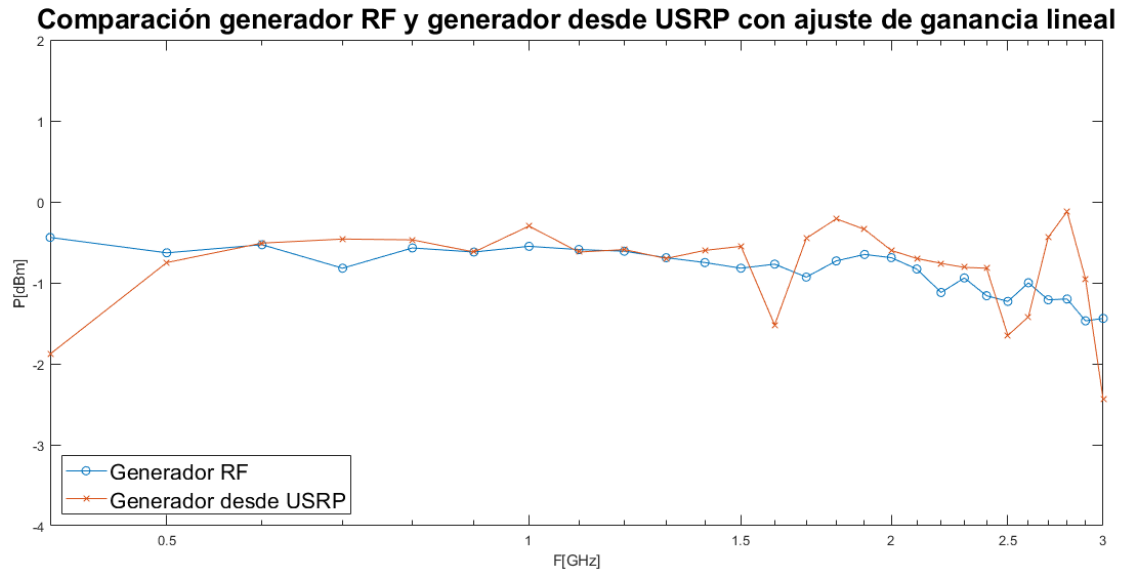
En la seccion 3.1 mediante regresiones se determinaron funciones que permitieron calibrar la curva de ganancia de transmisión del generador implementado en el *USRP X310* con *RFNoC*. Se repitió la prueba elaborada para la calibración, pero incluyendo un bloque variable en GNU Radio que modificaba el parámetro de ganancia del bloque *RFNoC* : Radio con respecto a la frecuencia de transmisión ajustada en el mismo. La Figura 22 es el diagrama de bloques implementado para esta prueba en GNU y en primera instancia se implementó la regresión lineal (ver Ecuación 3.4) para el ajuste de la ganancia, los resultados de ésta se evidencian en la Figura 23 en donde el comportamiento se asemejó al del generador de *RF*. En el rango de 500 MHz a 1.5 GHz la curva de la calibración se asemeja más a la del generador de *RF* con respecto a todo el rango de frecuencias utilizado para la prueba, pues en frecuencias como 400 MHz se tienen diferencias de 1.44 dBm, y de -1dBm y 1dBm en 2.8 GHz y 3 GHz respectivamente. Con este ajuste de ganancia, la curva de potencia del generador de *RF* implementado en el *USRP* oscila alrede-

dor de -0.78 dBm con variaciones de  $\pm 0.66$  dBm.

**Figura 22:** Diagrama de bloques para implementar generador de RF en *USRP* con ajuste de ganancia.



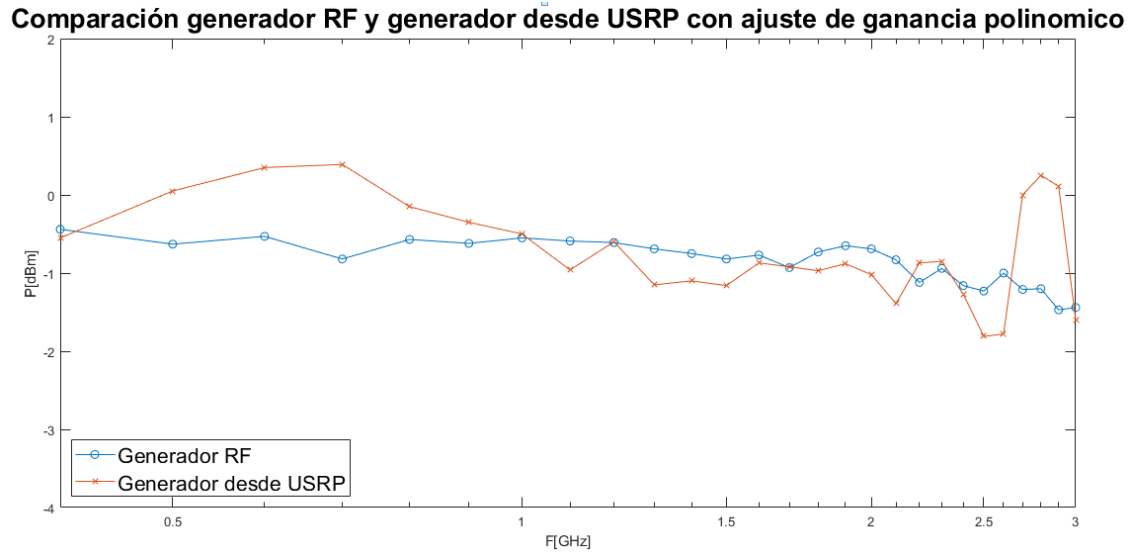
**Figura 23:** Comparación de las curvas de potencia de transmisión del *USRP* con ajuste de ganancia lineal y generador de *RF*.



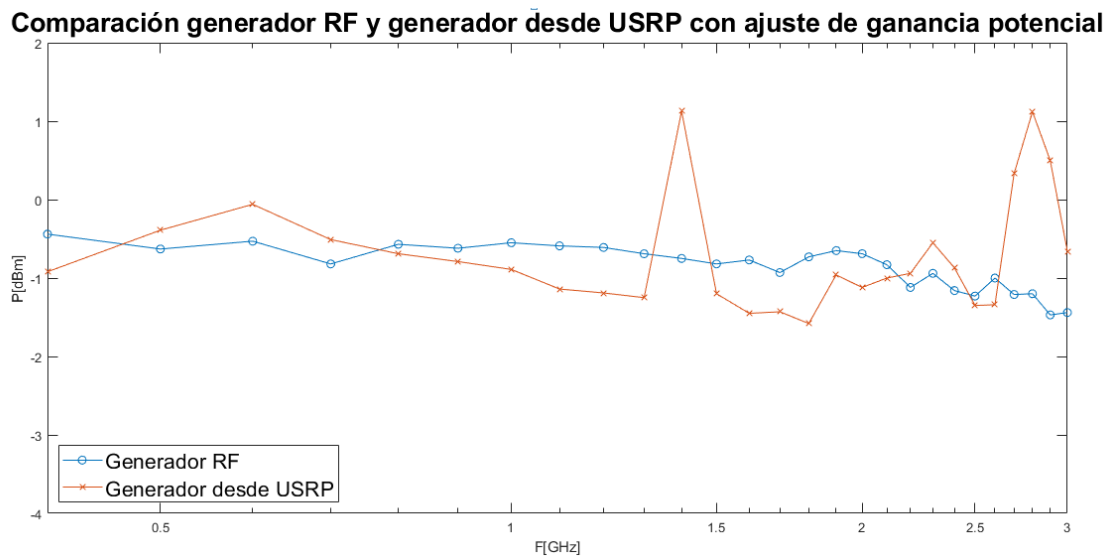
La curva de ganancia con ajuste polinómico se presenta en la Figura 24 con la cual se presentan mayores variaciones en comparación con la curva de ajuste de ganancia lineal. Para el ajuste polinómico se obtuvo una curva de potencia con oscilaciones al rededor de  $-0.78$  dBm al igual que en la lineal pero con variaciones hasta de  $\pm 1.11$  dBm. Con respecto a la curva del generador de *RF* los puntos más críticos en diferencias de potencia se presentan por encima de  $1.21$  dBm a  $700$  MHz y  $1.21$  dBm a  $1.58$  dBm entre  $2.7$  GHz y  $2.9$  GHz.

En la Figura 25 se presenta la curva de potencia con ajuste de ganancia potencial en la cual las diferencias con respecto a la curva del generador de *RF* es más significativa que en las curvas obtenidas con los ajustes realizados anteriormente. Las potencias más cercanas a la curva del generador de *RF* tienen diferencias mínimas de  $0.12$  dBm en  $800$  MHz y  $2.5$  GHz. La potencia promedio para este ajuste de ganancia es de  $-0.71$  dBm con variaciones de  $\pm 1.84$  dBm, alcanzando diferencias de hasta  $1.88$  dBm a  $1.4$  GHz, de  $1.54$  dBm en  $2.7$  GHz,  $2.3$  dBm en  $2.8$  GHz y  $1.97$  dBm en  $2.9$  GHz.

**Figura 24:** Comparación de las curvas de potencia de transmisión del *USRP* con ajuste de ganancia polinómica y generador de *RF*.



**Figura 25:** Comparación de las curvas de potencia de transmisión del *USRP* con ajuste de ganancia potencial y generador de *RF*.



La calibración resultó satisfactoria, pues se corrigió el error obtenido en la primer

prueba, sin el bloque de ajuste de ganancia. Aunque la corrección de la curva es satisfactoria, aun se presentan algunas diferencias dadas por la sensibilidad del amplificador de la etapa transmisora de la tarjeta SBX-40, para lo cual se recomienda realizar una calibración en la que intervengan el valor de la constante a modular, la ganancia del bloque Amp y el parámetro utilizado para esta calibración, la ganancia del bloque *RFNoC* : Radio. De las regresiones obtenidas, la regresión lineal presenta una mejor aproximación a los datos obtenidos del generador de *RF*, por lo cual se sugiere usar esta en la implementación de la etapa del generador de un *VNA*.

### 3.3 RECURSOS FPGA

A continuación se listan los recursos base para la síntesis de una imagen *.bit* con *RFNoC*, los utilizados por el bloque *Amp* implementado para la construcción de la etapa del generador de *RF* y los utilizados por los módulos que constituyen la etapa del procesador para la propuesta analizada.

**3.3.1 Recursos base *RFNoC*** El entorno *RFNoC* utiliza algunos recursos mínimos de *FPGA*, como base para el desarrollo del mismo. Estos recursos son: *slice lógicas (LUTs, registros, y mux)*, memoria (bloques de *RAM*), *DSP*, características específicas, recursos de reloj (*clocking*) y entradas y salidas. Estos recursos se presentan en las Tablas 4 a 7; obtenidos del reporte que se generó al sintetizar una imagen *.bit* sin bloques *RFNoC* excepto los que se generan por defecto (dos bloques *RADIO* y un *Dma-FIFO*)

**Tabla 4:** *Slice Logic* mínimas para síntesis de *RFNoC*.

<b>Site Type</b>	<b>Used</b>	<b>Available</b>	<b>Util %</b>
<i>Slice LUTs</i>	82921	254200	32.62
<i>LUT as Logic</i>	72332	254200	28.45
<i>LUT as Memory</i>	10589	90600	11.69
<i>LUT as Distributed RAM</i>	4774	–	–
<i>LUT as Shift Register</i>	5815	–	–
<i>Slice Registers</i>	103797	508400	20.42
<i>Register as Flip Flop</i>	103783	508400	20.41
<i>Register as Latch</i>	0	508400	0.00
<i>Register as AND/OR</i>	14	508400	< 0.01
<i>F7 Muxes</i>	576	127100	0.45
<i>F8 Muxes</i>	9	63550	0.01

**Tabla 5:** *Memory* mínima para síntesis de *RFNoC*.

<b>Site Type</b>	<b>Used</b>	<b>Available</b>	<b>Util %</b>
<i>Block RAM Tile</i>	265	795	33.33
<i>RAMB36/FIFO</i>	248	795	31.19
<i>RAMB36E1 only</i>	248	–	–
<i>RAMB18</i>	34	1590	2.14
<i>RAMB18E1 only</i>	34	–	–

**Tabla 6:** *DSP* mínima para síntesis de *RFNoC*.

<b>Site Type</b>	<b>Used</b>	<b>Available</b>	<b>Util %</b>
<i>DSPs</i>	71	1540	4.61
<i>DSP48E1 only</i>	71	–	–

**Tabla 7:** *Clocking* mínima para síntesis de *RFNoC*.

<b>Site Type</b>	<b>Used</b>	<b>Available</b>	<b>Util %</b>
<i>BUFGCTRL</i>	20	32	62.50
<i>BUFIO</i>	1	40	2.50
<i>BUFIO only</i>	1	–	–
<i>MMCME2_ADV</i>	4	10	40.00
<i>PLLE2_ADV</i>	3	10	30.00
<i>BUFMRCE</i>	0	20	0.00
<i>BUFHCE</i>	7	168	4.17
<i>BUFR</i>	2	40	5.00

Conocer los recursos base permite determinar los utilizados para la síntesis de los distintos bloques *RFNoC*, debido a que el reporte de recursos no distingue los utilizados por la base de los utilizados por los bloques, lo cual fue necesario tener claro en la realización de este trabajo de investigación.

**3.3.2 Recursos bloque *RFNoC:Amp*** Se generó una imagen *.bit* en la se sintetizó el bloque complementario de la etapa generadora de *RF*, y con la referencia de los recursos base del entorno, se obtuvieron los utilizados por este bloque. Los resultados se presenta en las Tablas 8 a 10.

**Tabla 8:** Recursos *slice Logic* del bloque *Amp*.

<b>Site Type</b>	<b>Used</b>	<b>Util %</b>
<i>Slice LUTs</i>	3843	1.51
<i>LUT as Logic</i>	3522	1.39
<i>LUT as Memory</i>	321	0.35
<i>LUT as Distributed RAM</i>	268	–
<i>LUT as Shift Register</i>	53	–
<i>Slice Registers</i>	7882	1.55
<i>Register as Flip Flop</i>	7882	1.55
<i>Register as Latch</i>	0	0.00
<i>Register as AND/OR</i>	0	0.00
<b>F7 Muxes</b>	100	0.08
<b>F8 Muxes</b>	32	0.05

**Tabla 9:** Recursos *Memory* del generador del bloque *Amp*.

<b>Site Type</b>	<b>Used</b>	<b>Util %</b>
<i>Block RAM Tile</i>	8.5	1.07
<i>RAMB36/FIFO</i>	8	1.01
<i>RAMB36E1 only</i>	8	–
<i>RAMB18</i>	1	0.06
<i>RAMB18E1 only</i>	1	–

**Tabla 10:** Recursos *DSP* del bloque *Amp*.

<b>Site Type</b>	<b>Used</b>	<b>Util %</b>
<i>DSPs</i>	2	0.13
<i>DSP48E1 only</i>	2	–

**3.3.3 Recursos para alternativas de implementación del procesador** Los recursos utilizados en la generación del bloque *Adder/Subtractor* dependen del tipo

de implementación que se escoja, la Tabla 11 muestra los recursos que se necesitan para generarlo en modo *Fabric* y en modo *DSP48*.

**Tabla 11:** Recursos utilizados en los tipos de implementación.

<b>Recursos</b>	<b>Fabric</b>	<b>DSP48</b>
DSPs	0	1
BRAMs	0	0
LUTs	15	0
FFs	15	0

Para obtener estos resultados se tuvo que sintetizar el IP-CORE del Adder/Subtractor seleccionando cada uno de las implementaciones, la duración de cada compilación es de aproximadamente 30 a 40 minutos. Se recomienda la construcción de este bloque en arquitectura *DSP48* dado que este solo consume un recurso *DSP* y la Kintex7 cuenta con 1540 bloques de este tipo como se ve en la Tabla 6.

Los recursos que utiliza el *Divider Generator* dependen principalmente de la elección del algoritmo con el cual se desee generar y de la configuración general que se establezca. En la Tabla 12 se pueden observar los recursos que se utilizan con cada algoritmo, eligiendo un ancho de 16 bits en los buses y las otras configuraciones por defecto. Cabe resaltar que al igual que el adder/subtractor la compilación para cada uno de los algoritmos tiene un tiempo estimado de 30 a 40 minutos que dependen del procesador del equipo de computo que se este utilizando.

**Tabla 12:** Recursos utilizados por cada algoritmo.

<b>Recursos</b>	<b>Radix2</b>	<b>High Radix</b>	<b>Lutmult</b>
DSPs	0	11	2
BRAMs	0	0.5	0.5
LUTs	401	486	68
FFs	938	924	209

El módulo complejo a magnitud y fase que dispone *RFNoC* se sintetiza por defecto y los recursos que este utiliza se encuentran inmersos en los recursos base del entorno.

### 3.4 ACOPLE DIRECCIONAL 87300B

El acople 87300B es un acople direccional de banda ancha con puerto aislado interno fabricado por Keysight [16], utilizado para el monitoreo y nivelación de señales. Algunas especificaciones de este acople se presentan en la Tabla 13 [15]. Estos parámetros están definidos para todo el rango de frecuencias de operación, por lo cual se caracterizó este acople para el rango de frecuencias de 1GHz a 3 GHz, dado por la intersección del rango trabajado con el generador (400MHz a 3GHz), el del acople direccional y el rango de operación del analizador de espectros HMS – X.

**Tabla 13:** Especificaciones del acople direccional 87300B.

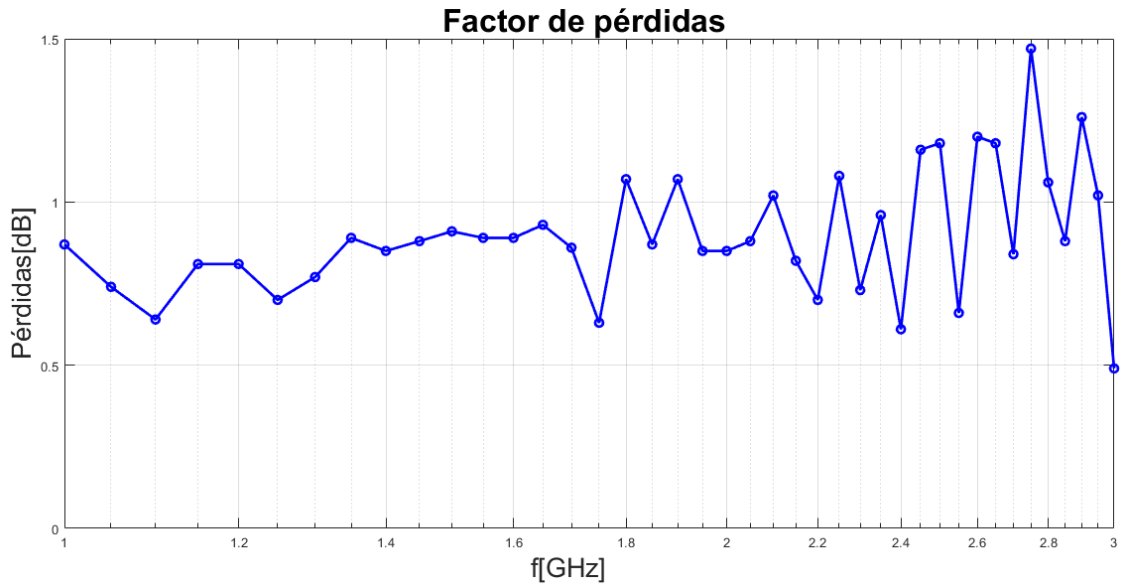
<i>Especificación</i>	<i>Valor</i>
Rango de frecuencia	1GHz - 20GHz
Acoplamiento y desviación	$10 \pm 0.5$ dB
Directividad	$> 16$
Pérdidas	$< 1.5$
Sensibilidad de frecuencia	$\pm 0.35$ dB
Máximo <i>SWR</i>	1.35 dB

**3.4.1 Caracterización de un acople direccional 87300B** Se realizó una caracterización del acople 87300B con base en la descripción de características realizada en la subsección 1.2.1. A continuación se presentan algunos parámetros obtenidos en la caracterización dado que por ser un acople de puerto aislado interno, no es posible determinar parámetros que dependen de este puerto, como la directividad.

Se realizó la medición de las pérdidas del acople, conectando el puerto de entrada del acople al generador de *RF*, el puerto de salida al analizador de espectros y el puerto de acople a una carga resistiva de  $51.4 \Omega$ . La curva de este parámetro se

presenta en la Figura 26, en la cual se evidencia que este factor es menor a 1.5 dB, con pérdida mínima de 0.49 dB en 3 GHz y máxima de 1.47 dB en 2.75 GHz.

**Figura 26:** Curva de pérdidas para un acople direccional 87300B de 1GHz a 3 GHz.



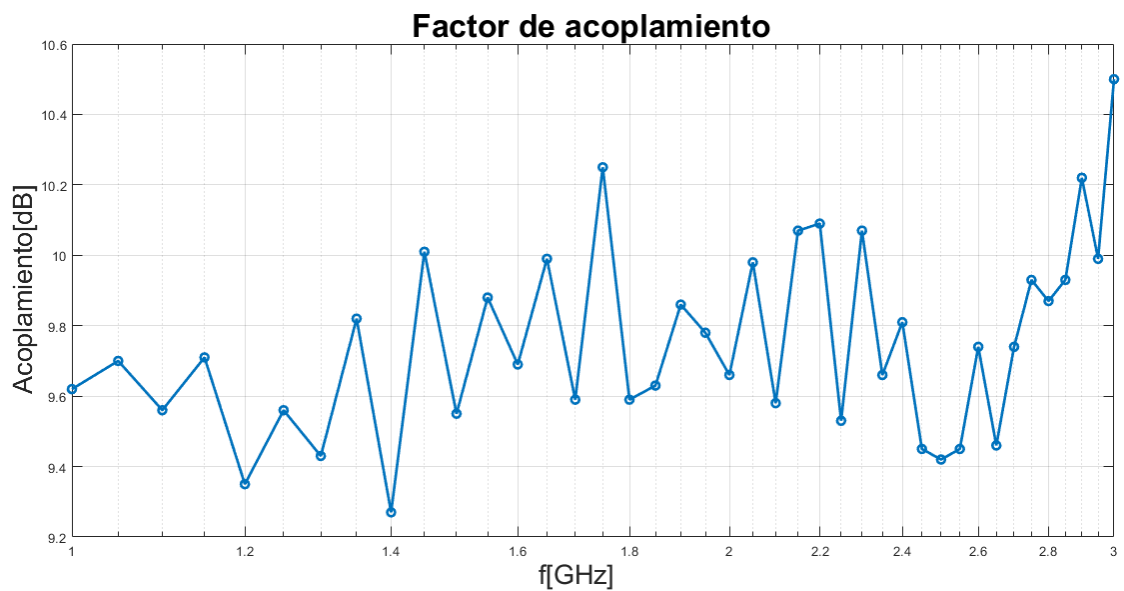
Para determinar el factor de acoplamiento se conectó el puerto de entrada del acople direccional al generador de *RF* y el puerto de salida se terminó con la carga resistiva. Se midió la potencia en el puerto de entrada y en el de acoplamiento para trazar la curva del factor de acoplamiento aplicando la ecuación 1.5 la cual se presenta en la Figura 27. El factor de acoplamiento para este rango de frecuencias es de 9.88 dB con variaciones de 0.61 dB, el punto máximo de acoplamiento es de 10.5 dB, el cual se encuentra en 3 GHz y el mínimo es de 9.27 dB en 1.4 GHz.

La sensibilidad de frecuencia se determinar a partir de los puntos máximos y mínimos del factor de acoplamiento mediante la ecuación 3.5. La sensibilidad obtenida en la calibración es de  $\pm 0.31$  dB.

$$Frequency\_sensitivity(dB) = \frac{min\_punto\_acoplamiento - max\_punto\_acoplamiento}{2} \tag{3.5}$$

El factor de SWR se determinó entre el nivel de potencia del puerto de acople y el del puerto de salida, al puerto de entrada se le conectó la carga resistiva. La curva de este factor se presenta en la Figura 28. El factor de SWR se encuentra por debajo de -15.31 dB en 2.95 GHz, alejados del límite máximo especificado, el mínimo es de -24.02 dB que se presenta en el extremo inferior (1 GHz) del rango de frecuencias de operación de este acople.

**Figura 27:** Curva de factor de acoplamiento para un acople direccional 87300B de 1GHz a 3 GHz.

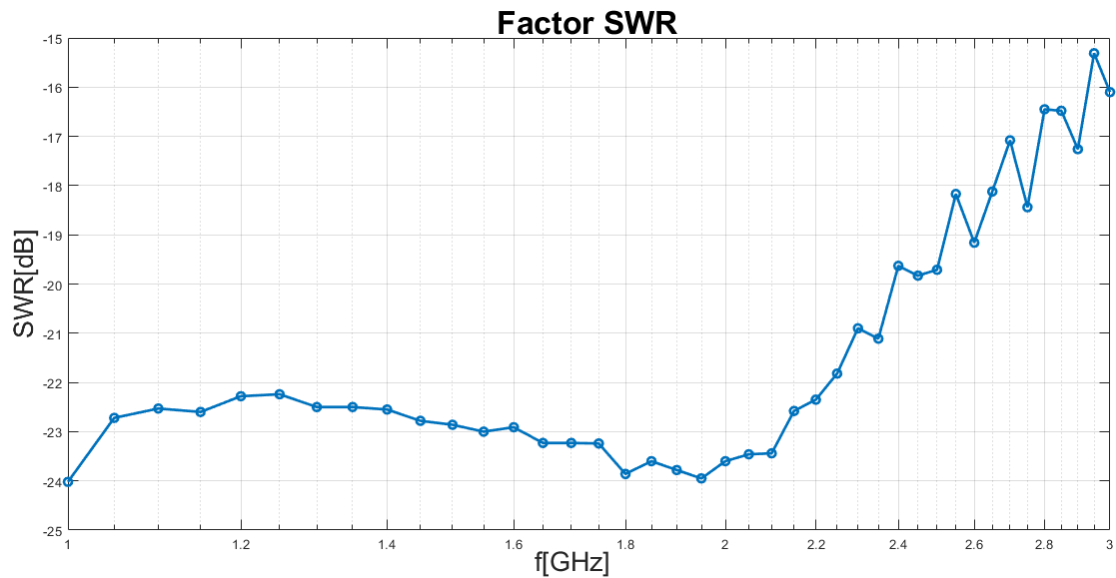


Los datos resultantes de la caracterización se presentan en la Tabla 14. Los parámetros obtenidos se ajustan a los suministrados por la fábrica(ver Tabla 13).

De esta caracterización se obtiene:

- El error del factor de acoplamiento es de 1.2% y la variación presenta error del 20%.
- El factor de pérdidas se encuentra por debajo del límite especificado, presentado una pérdida máxima con diferencia de 0.03dB del límite.

**Figura 28:** Curva de  $SWR$  para un acople direccional 87300B de 1GHz a 3 GHz.



- El factor  $SWR$  se encuentra alejado 16,66 dB del punto máximo, dado que el rango de caracterización es del 10,52 % ubicado en el límite inferior del rango de operación, los datos específicos de fábrica resultan de la caracterización de todo el rango (1GHz a 20 GHz) y se evidencia que a medida que aumenta la frecuencia, acercándose a la frecuencia central de operación del acople, este factor aumenta.

**Tabla 14:** Caracterización del acople direccional 87300B para un rango de 1 GHz a 3 GHz.

<b>Especificación</b>	<b>Valor</b>
Acoplamiento y variación	$9.88 \pm 0.6$ dB
Pérdidas	$< 1.47$ dB
Sensibilidad de frecuencia	$\pm 0.31$ dB
Máximo $SWR$	-15.31 dB

**3.4.2 Alternativas para realizar el test set** En la sección anterior se realizó la caracterización de un acople direccional adquirido, el cual cuenta con un rango de frecuencias de operación diferente al que presenta el *USRP X310*, este es útil para la implementación de un *VNA* pero repercutiría en el rango de frecuencias, por lo que otras alternativas que se pueden contemplar son:

- El diseño y desarrollo de un acople direccional que comprenda el rango de frecuencias adecuado.
- La implementación del circuito puente *VSWR* (ver subsección 1.2).

---

## 4. CONCLUSIONES

---

- Se implementó un generador de  $RF$  sobre un *USRP* X310 con rango de operación de 400 MHz a 3 GHz utilizando el entorno *RFNoC* y los recursos físicos de la etapa transmisora de la tarjeta SBX-40, con potencia de salida promedio de -0.78 dBm y variaciones de  $\pm 0.66$  dB aplicando un ajuste de ganancia lineal.
  - La construcción de un instrumento de este tipo sobre *RFNoC* sólo consume el 1.5% de recursos *slice*, 1.07% de bloques de memoria *RAM* y 0.13% de *DSPs* de los disponibles tras la síntesis de bloques *RFNoC*, por lo cual la gran mayoría de los recursos quedan disponibles para la implementación de otras etapas que constituyen en conjunto un analizador vectorial de redes.
- *RFNoC* cuenta con una variedad de módulos de nivel *FPGA* disponibles a los usuarios del entorno, para la creación de sistemas similares a los que se pueden construir en GNU Radio, además permite incorporar módulos *IP-Core* Xilinx con diferentes configuraciones de acuerdo a las necesidades de los usuarios.
  - Esto facilita la descripción de los bloques planteados como alternativas para la implementación de un *VNA* sobre *USRP* u otro dispositivo que permita ser descrito sobre *FPGA*, utilizando el entorno *RFNoC*.
- El implementar instrumentación sobre un dispositivo como un *USRP* presenta ventajas como:
  - La relación costo beneficio, dado que un *USRP* X310 tiene un costo de \$5,290.00 y un *VNA* ZVL6 de \$14,500.00, de los cuales el *USRP* al ser un dispositivo reconfigurable permite implementar distintos sistemas como un *VNA*, mientras que un *VNA* comercial sólo brinda una función específica.

- Estos equipos *SDR* son un conjunto de etapas, desde nivel *FPGA* hasta configuraciones de radio frecuencia, con integración de diferentes softwares como Vivado y GNU Radio, del cual el último es un software libre.
- Es posible controlar y acceder a las prestaciones del *USRP* mediante los controladores *UHD* y con la integración de estos softwares mediante *RFNoC* se facilita el diseño y desarrollo de sistemas, los cuales serían complejos de implementar si se desea construir desde cero, acoplando *FPGAs*, etapas de *RF* y demás periféricos que constituyen un *USRP*.
- Finalmente es viable la implementación de un analizador vectorial de redes con base en la arquitectura *USRP*.
  - Estos deben cumplir algunos requerimientos específicos como: sincronización de *VCOs* en las etapas de transmisión y recepción, banco de filtros que permitan una respuesta en frecuencia con banda plana en el rango de frecuencias seleccionado para operar, *FPGAs* de última generación que ofrezcan una cantidad de recursos físicos que permitan implementar bloques de las etapas del *VNA* y no presenten limitaciones de recursos.
  - Así mismo, el contar con licenciamiento Xilinx para el correcto uso de estas tarjetas, optimizaría los procesos de desarrollo e implementación.

---

## 5. RECOMENDACIONES

---

- Ettus incorporó la serie N a la última actualización del entorno *RFNoC*, con doce imágenes *.bit* para los *USRP* N300 y N310, seis para cada uno. Estos *USRP* disponen de cuatro canales de *RF* y rango de frecuencia de 10 MHz a 6 GHz (con tarjetas hijas fijas, no cambiables) [8]. La modulación en cuadratura que se implementa en los X310 introduce variaciones que no son posibles de determinar en tiempo real, lo cual puede introducir errores al determinar de los parámetros de *Scattering*. El uso de un radio de cuatro puertos permitiría implementar una mejor configuración de *VNA*, dado que es posible contar con la adquisición simultánea de las dos señales (incidente y reflejada) en dos puertos de  $R_x$ , utilizando *power splitter* a la salida de las señales incidentes ( $T_x$ ), además, estos equipos cuentan con reloj de referencia para sincronizar los osciladores, con lo cual se podría corregir el error introducido por el diferencial de frecuencia que presentan los moduladores y demoduladores de cuadratura al no tener sincronización en los *VCOs*. Contando con estas características resulta pertinente extrapolar este trabajo de investigación a estos *USRP*, visualizando incluso la implementación de *VNA* para redes de cuatro puertos.
- Durante la ejecución de este proyecto se presentó una actualización del entorno *RFNoC*, en la cual se evidenció el desarrollo acorde a las necesidades de los usuarios de radio definida por software, como la incorporación de nuevas series *USRP* y optimización de recursos (disminución de recursos base en 4.25 % de *slice*, 5.6 % en bloques de memoria RAM y 18.75 % en recursos de reloj). El desarrollo que está presentando *RFNoC* permite visualizar la integración o cooperación de las asignaturas de comunicaciones y sistemas digitales del programa académico Ingeniería Electrónica, en la construcción de sistemas de comunicación sobre *SDR*, para lo cual es importante que la escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones de la Universidad Industrial de Santander, adquiera las licencias correspondientes de Vivado, en pro de contar con todos los recursos que Xilinx ofrece para la

construcción y síntesis de estos bloques en nivel *FPGA*. Esto dado porque en el proceso de síntesis se presentan algunos *Critical Warning* con motivo de licenciamiento que no permiten un proceso más limpio, pues los productos *IP-Core* que el entorno utiliza no cuentan con la licencia requerida.

- Para la implementación de un analizador vectorial de redes con un mayor rango de frecuencias de operación, se sugiere la construcción de un acople direccional o puente *VSWR* que cubra todo el rango de operación del *USRP* (400 MHz a 4.4 GHz) dado que el adquirido para la caracterización, aunque es útil, disminuye el rango de operación para el *VNA* que se podría implementar.
- Se propone la actualización de los equipos e implementos del laboratorio de comunicaciones de la escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones de la Universidad Industrial de Santander, dado que los equipos de cómputo no presentan los requerimientos necesarios para el uso de *RFNoC*, los analizadores de espectros y osciloscopios presentan frecuencias de operación inferiores a 3 GHz y 400 MHz respectivamente, factores que limitaron el análisis presentado. Además, adquirir nuevas series *USRP* que presentan mejores prestaciones como mayores rangos de operación, sincronización de reloj, bancos de filtro, entre otros; permitirían avanzar y optimizar la implementación de instrumentación para comunicaciones sobre *SDR*.

---

## BIBLIOGRAFÍA

---

- [1] ANALOG DEVICES. 400 mhz to 6 ghz broadband quadrature modulator, data sheet adl5375 [en línea]. <http://www.analog.com/media/en/technical-documentation/data-sheets/ADL5375.pdf>. Recuperado: 1 de julio del 2018.
- [2] ANALOG DEVICES. Wideband synthesizer with integrated vco, data sheet adf4350 [en línea]. <https://co.mouser.com/datasheet/2/609/ADF4350-878384.pdf>. Recuperado: 1 de julio del 2018.
- [3] Analog Devices. Operational amplifiers with shutdown [en línea]. [http://www.analog.com/media/en/technical-documentation/data-sheets/AD8591\\_8592\\_8594.pdf](http://www.analog.com/media/en/technical-documentation/data-sheets/AD8591_8592_8594.pdf). Recuperado: 24 de julio del 2018.
- [4] Analog Devices. Quadrature demodulator [en línea]. <https://www.mouser.mx/datasheet/2/609/ADL5380-878580.pdf>. Recuperado: 24 de julio del 2018.
- [5] Analog Devices. Ultralow distortion, current feedback differential adc driver [en línea]. [https://www.mouser.com/datasheet/2/609/ADA4927-1\\_ADA4927-2-878322.pdf](https://www.mouser.com/datasheet/2/609/ADA4927-1_ADA4927-2-878322.pdf). Recuperado: 24 de julio del 2018.
- [6] Benavente, C., Manuel Pardo Martín, J., Javier Ortega González, F., and Jiménez Trillo, J. Consideraciones sobre la arquitectura de receptores para software radio mediante fpga.
- [7] Buroker, G. E., and Frerking, M. E. A digitally compensated tcxo. In *27th Annual Symposium on Frequency Control* (June 1973), pp. 191–198.
- [8] Ettus Research. Usrp n310 [en línea]. <https://www.ettus.com/product/details/USRP-N310>. Recuperado: 17 de julio del 2018.
- [9] Ettus Research. Usrp x310 [en línea]. <https://www.ettus.com/product/details/X310-KIT>, 2017. Revisado: 11 de Marzo del 2018.

- [10] Ettus Research. About ettus research [en línea]. <https://www.ettus.com/site/about>, 2018. Revisado: 14 de Marzo del 2018.
- [11] Ettus Research. Rfnoc (red de rf en chip) [en línea]. <https://www.ettus.com/sdr-software/detail/rf-network-on-chip>, 2018. Revisado: 17 de Marzo del 2018.
- [12] GNU Radio. Outoftreemodules [en línea]. <https://wiki.gnuradio.org/index.php/OutOfTreeModules>. Recuperado: 14 de julio del 2018.
- [13] Helfrich., A. D. *Electrical Spectrum and Network Analyzers: A Practical Approach*, 1 st ed. ISBN 0-12-338250-5. Academic Press, Inc., 1991.
- [14] Hiebel., M. *Fundamentals of Vector Network Analysis*, 1 st ed. ISBN 978-3-939837-06-0. RohdeSchwarz GmbH Co.KG, 2007.
- [15] Keysight. Keysight couplers [en línea]. <https://literature.cdn.keysight.com/litweb/pdf/5990-5353EN.pdf?id=1850620>. Recuperado: 19 de julio del 2018.
- [16] Keysight. Keysight technologies [en línea]. <https://www.keysight.com/zz/en/home.html>. Recuperado: 19 de julio del 2018.
- [17] Krishnan, R., Babu, R. G., Kaviya, S., Kumar, N. P., Rahul, C., and Raman, S. S. Software defined radio (sdr) foundations, technology tradeoffs: A survey. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (Sept 2017), pp. 2677–2682.
- [18] Mini-Circuits. Ceramic low pass filter, data sheet lfcn-1200 [en línea]. <https://www.minicircuits.com/pdfs/LFCN-1200.pdf>. Recuperado: 1 de julio del 2018.
- [19] Mishra, M., Potnis, A., Dwivedy, P., and Meena, S. K. Software defined radio based receivers using rtl x2014; sdr: A review. In *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)* (Oct 2017), pp. 62–65.

- 
- [20] NATIONAL INSTRUMENTS. Dispositivo de radio definido por software [en línea]. <https://www.ni.com/es-co/shop/select/usrp-software-defined-radio-device>. Recuperado: 1 de agosto del 2018.
- [21] Naval air warfare center, N. a. s. c. *Electronic Warfare and Radar Systems Engineering Handbook*, 2 nd ed. RohdeSchwarz GmbH Co.KG, 1999.
- [22] Rohde&Schwarz. *R&S<sup>®</sup>ZVL Vector Network Analyzer Specifications*. Rohde&Schwarz. Revisado: 1 de Febrero del 2018.
- [23] RohdeSchwarz GmbH Co.KG. Network analyzer accessories [en línea]. [https://www.rohde-schwarz.com/lat/producto/network\\_analyzer\\_accessories-pagina-de-inicio-producto\\_63493-41856.html](https://www.rohde-schwarz.com/lat/producto/network_analyzer_accessories-pagina-de-inicio-producto_63493-41856.html). Recuperado: 11 de Abril del 2018.
- [24] RohdeSchwarz GmbH Co.KG. *Rs<sup>®</sup>zvl6 vector network analyzer* [en línea]. [https://www.rohde-schwarz.com/lat/producto/zvl6-pagina-de-inicio-producto\\_63493-9016.html](https://www.rohde-schwarz.com/lat/producto/zvl6-pagina-de-inicio-producto_63493-9016.html). Recuperado: 11 de Abril del 2018.
- [25] *R&S<sup>®</sup>ZVL*. *Hms-x spectrum analyzer* [en línea]. [https://www.rohde-schwarz.com/us/product/hms-productstartpage\\_63493-42689.html](https://www.rohde-schwarz.com/us/product/hms-productstartpage_63493-42689.html). Recuperado: 27 de julio del 2018.
- [26] *R&S<sup>®</sup>ZVL*. *r&s<sup>®</sup>zvl smb100a rf and microwave signal generator* [en línea]. [https://www.rohde-schwarz.com/us/product/smb100a-productstartpage\\_63493-9379.html](https://www.rohde-schwarz.com/us/product/smb100a-productstartpage_63493-9379.html). Recuperado: 27 de julio del 2018.
- [27] Weiss, K., Gniewinska, B., and Nafalski, L. The modern oxco quartz oscillators requirements and parameters. In *6th International Conference on Laser and Fiber-Optical Networks Modeling, 2004. Proceedings of LFNM 2004*. (Sept 2004), pp. 209–212.
-

- [28] XILINX. Adder/subtracter v12.0. logicore ip product guide [en línea]. [https://www.xilinx.com/support/documentation/ip\\_documentation/addsub/v12\\_0/pg120-c-addsub.pdf](https://www.xilinx.com/support/documentation/ip_documentation/addsub/v12_0/pg120-c-addsub.pdf). Recuperado: 7 de julio del 2018.
- [29] XILINX. Divider generator v5.1. logicore ip product guide [en línea]. [https://www.xilinx.com/support/documentation/ip\\_documentation/div\\_gen/v5\\_1/pg151-div-gen.pdf](https://www.xilinx.com/support/documentation/ip_documentation/div_gen/v5_1/pg151-div-gen.pdf). Recuperado: 7 de julio del 2018.
- [30] XILINX. 7 series fpgas data sheet: Overview [en línea]. [https://www.xilinx.com/support/documentation/data\\_sheetsds180\\_7Series\\_Overview.pdf](https://www.xilinx.com/support/documentation/data_sheetsds180_7Series_Overview.pdf), 2018. Revisado: 14 de Marzo del 2018.

---

## ANEXOS

---

### A. Plantillas bloque *Amp*

Instalado el entorno *RFNoC* se creó el módulo *Amplitud* de tipo *Out of tree* (*OOT*) sobre el cual está creada la arquitectura del entorno [12], mediante la herramienta *rfnocmodtool newmod* desde la terminal con el comando:

```
$ cd ~/rfnoc/src/ && rfnocmodtool newmod Amplitud
```

Después se agregó el bloque *Amp* al módulo con la herramienta *rfnocmodtool add* con:

```
$ cd ~/rfnoc/src/rfnoc-Amplitud/ && rfnocmodtool add Amp
```

Creado el bloque y con base en la estructura de la Figura 15, con *gedit* se editaron los siguientes archivos:

```
$ gedit ~/rfnoc/src/rfnoc-Amplitud/rfnoc/fpga-src/noc_block_Amp.v
$ gedit ~/rfnoc/src/rfnoc-Amplitud/rfnoc/blocks/Amp.xml
$ gedit ~/rfnoc/src/rfnoc-Amplitud/grc/Amplitud_Amp.xml
```

- Plantilla de Vivado para bloque *Amp*.

Se edita la plantilla de nivel *FPGA* creada por el entorno al agregar el bloque *Amp*. De la interfaz *AXI*, sólo se utilizan las señales *s\_axis\_data* como se muestra a continuación, dado que este bloque cuenta únicamente con puerto de salida.

```
...
////////////////////////////////////
//
// AXI Wrapper
// Convert RFNoC Shell interface into AXI stream interface
//
////////////////////////////////////

wire [31:0] s_axis_data_tdata;
```

```
wire      s_axis_data_tlast;
wire      s_axis_data_tvalid;
wire      s_axis_data_tready;
wire      s_axis_data_tvalid_int;
wire      s_axis_data_tready_int;
wire [31:0] s_axis_const_tdata;
wire      s_axis_const_tvalid;
wire      s_axis_const_tready;
wire      enable;
...
axi_wrapper #(
    .SIMPLE_MODE(1))
axi_wrapper (
    .clk(ce_clk), .reset(ce_rst),
    .bus_clk(bus_clk), .bus_rst(bus_rst),
    .clear_tx_seqnum(clear_tx_seqnum),
    .next_dst(next_dst_sid),
    .set_stb(set_stb), .set_addr(set_addr), .set_data(set_data),
    .i_tdata(), .i_tlast(), .i_tvalid(), .i_tready(),
    .o_tdata(str_src_tdata), .o_tlast(str_src_tlast), .o_tvalid(
        str_src_tvalid), .o_tready(str_src_tready),
    .m_axis_data_tdata(),
    .m_axis_data_tlast(),
    .m_axis_data_tvalid(),
    .m_axis_data_tready(1'b1),
    .m_axis_data_tuser(),
    .s_axis_data_tdata(s_axis_data_tdata),
    .s_axis_data_tlast(s_axis_data_tlast),
    .s_axis_data_tvalid(s_axis_data_tvalid),
    .s_axis_data_tready(s_axis_data_tready),
    .s_axis_data_tuser(),
    .m_axis_config_tdata(),
    .m_axis_config_tlast(),
    .m_axis_config_tvalid(),
    .m_axis_config_tready(),
    .m_axis_pkt_len_tdata(),
    .m_axis_pkt_len_tvalid(),
    .m_axis_pkt_len_tready());
```

Se asignan las direcciones de registro para cada parámetro, posteriormente se configuran con los `setting_reg` y `axi_setting_reg` para la construcción del bloque.

```

...
// NoC Shell registers 0 – 127,
// User register address space starts at 128
localparam SR_GAIN = 128;
localparam SR_ENABLE = 129;
localparam SR_CONSTANT = 130;
...
// Start/stop functionality
// Enable_block
setting_reg #(
    .my_addr(SR_ENABLE), .awidth(8), .width(1), .at_reset(0))
set_enable (
    .clk(ce_clk), .rst(ce_rst),
    .strobe(set_stb), .addr(set_addr), .in(set_data),
    .out(enable), .changed());

// Constant_Block

axi_setting_reg #(
    .ADDR(SR_CONSTANT), .AWIDTH(8), .WIDTH(32), .REPEATS(1))
const_block (
    .clk(ce_clk), .reset(ce_rst | clear_tx_seqnum),
    .set_stb(set_stb), .set_addr(set_addr), .set_data(set_data),
    .o_tdata(s_axis_const_tdata), .o_tlast(), .o_tvalid(
        s_axis_const_tvalid), .o_tready(s_axis_const_tready));

// Gain_Block

wire [15:0] gain;
setting_reg #(
    .my_addr(SR_GAIN), .awidth(8), .width(16))

```

```

set_gain (
    .clk(ce_clk), .rst(ce_rst),
    .strobe(set_stb), .addr(set_addr), .in(set_data), .out(gain),
    .changed());

```

La etapa de ganancia digital está conformada por dos módulos que brinda *RFNoC*, por lo cual se realiza la instanciación de los mismos y se conectan para la construcción de este bloque, basados en la plantilla del bloque *Digital Gain* de *RFNoC*.

```

// Enable logic
assign s_axis_data_tvalid = s_axis_data_tvalid_int & enable;
assign s_axis_data_tready_int = s_axis_data_tready & enable;

wire [31:0] mult_tdata;
wire      mult_tlast;
wire      mult_tvalid;
wire      mult_tready;

mult_rc #(.WIDTH_REAL(16), .WIDTH_CPLX(16), .WIDTH_P(32), .
        DROP_TOP_P(5), .LATENCY(4)) mult_rc (

    .clk(ce_clk), .reset(ce_rst | clear_tx_seqnum),
    .real_tdata(gain), .real_tlast(1'b0), .real_tvalid(1'b1), .
        real_tready(),
    .cplx_tdata(s_axis_const_tdata), .cplx_tlast(1'b0),
    .cplx_tvalid(s_axis_const_tvalid), .cplx_tready(s_axis_const_tready),
    .p_tdata(mult_tdata); .p_tlast(mult_tlast);,
    .p_tvalid(mult_tvalid);, .p_tready(mult_tready););

axi_clip_complex #(.WIDTH_IN(32), .WIDTH_OUT(16))
axi_clip_complex (
    .clk(ce_clk), .reset(ce_rst | clear_tx_seqnum),
    .i_tdata(mult_tdata);, .i_tlast(mult_tlast);,
    .i_tvalid(mult_tvalid), .i_tready(mult_tready),

```

```

        .o_tdata(s_axis_data_tdata), .o_tlast(s_axis_data_tlast), .o_tvalid(
            s_axis_data_tvalid_int),
        .o_tready(s_axis_data_tready_int));

endmodule

```

- Plantilla nivel UHD de Amp.

Se procede a la edición de la plantilla de integración *UHD*, las líneas que se encuentran a continuación, son agregadas por defecto al momento de crear el bloque. Sin embargo se debe verificar que el NOC ID coincida con el asignado en el momento de la creación del bloque y al que aparece en la plantilla de Vivado (el cual no se muestra en la plantilla de Vivado presentada anteriormente, por reducción del *script* y porque igualmente se asigna por defecto a esta plantilla al momento de crear el bloque).

```

<nocblock>
  <name>Amp</name>
  <blockname>Amp</blockname>
  <ids>
    <id revision="0">DACC4C1758AA3096</id>
  </ids>
  ...

```

Para este nivel de integración se definen nuevamente los registros y sus respectivas direcciones, los cuales deben coincidir con los asignados en la plantilla de Vivado.

```

...
<!--One input, one output. If this is used, better have all the
      info the C++ file.-->
<registers>
  <setreg>

```

```

    <name>GAIN</name>
    <address>128</address>
  </setreg>
  <setreg>
    <name>ENABLE</name>
    <address>129</address>
  </setreg>
  <setreg>
    <name>CONSTANT</name>
    <address>130</address>
  </setreg>
</registers>
...

```

A continuación se definen los argumentos que posteriormente son utilizados para el enlace o la integración de nivel GNU. En estos argumentos se define: el tipo de argumento, un valor, rango de operación, se asigna un mensaje y se ejecuta la acción correspondiente a cada uno.

```

...
<!-- Args -->
<args>
  <arg>
    <name>gain</name>
    <type>double</type>
    <value>1.0</value>
    <check>GE($gain,0.0) AND LE($gain,32767.0)</check>
    <check_message>Invalid gain.</check_message>
    <action>
      SR_WRITE( "GAIN" , IROUND($gain) )
    </action>
  </arg>
  <arg>
    <name>enable</name>
    <type>int</type>
    <value>0</value>

```

```

    <check>EQUAL($enable,0) OR EQUAL($enable, 1) </check>
    <check_message>Enable is either 0 or 1. </check_message>
    <action>SR.WRITE("ENABLE", $enable) </action>
</arg>
<arg>
    <name>amplitude_i</name>
    <type>double</type>
    <value>1.0</value>
    <check>GE($amplitude_i, -1.0) AND LE($amplitude_i,1.0) </check>
    <check_message>Invalid amplitude. </check_message>
</arg>
<arg>
    <name>amplitude_q</name>
    <type>double</type>
    <value>1.0</value>
    <check>GE($amplitude_q, -1.0) AND LE($amplitude_q*,(*1.0) </check>
    <check_message>Invalid amplitude. </check_message>
    <action>
        SR.WRITE("CONSTANT", ADD(MULT(65536, IROUND(MULT(32767.0,
            $amplitude_i))), IROUND(MULT(32767.0, $amplitude_q))))
    </action>
</arg>
</args>

```

Finalmente se definen los puertos y el tipo. Para este tipo de bloque que es una fuente, el puerto de entrada (*sink*) se declara como *dump* para deshabilitarlo y evitar problemas de puertos desconectados al momento de utilizar el bloque.

```

<!-- Ports -->
<ports>
    <sink>
        <name>dump</name>
    </sink>
    <source>
        <name>out0</name>

```

```

    <type>sc16</type>
  </source>
</ports>
</nocblock>

```

- Plantilla nivel GNU de Amp.

Se procede a la edición de la plantilla de nivel GNU, en la cual se deben inicializar dentro del *make* los argumentos definidos en la plantilla de nivel *UHD* con la estructura que se presenta a continuación.

```

...
<make>Amplitud.Amp (
    ...
    $block_index ,
    $device_index
)
self.$(id).set_arg("gain", $gain)
self.$(id).set_arg("enable", $enable)
self.$(id).set_arg("amplitude_i", $(amplitude).real)
self.$(id).set_arg("amplitude_q", $(amplitude).imag)

</make>

```

Con los argumentos definidos, se hace la asignación del espacio dentro del bloque GNU a través del comando *callback*.

```

<callback>set_arg("gain", $gain)</callback>
<callback>set_arg("enable", $enable)</callback>
<callback>set_arg("amplitude_i", $(amplitude).real)</callback>
<callback>set_arg("amplitude_q", $(amplitude).imag)</callback>

```

A continuación se definen los parámetros que modifican el bloque desde el nivel GNU, con la etiqueta `< param >` y las sub etiquetas `< name >` que

asigna el nombre del parámetro dentro del bloque, `< key >` guarda el valor asignado al parámetro, `< value >` es el valor que se asigna por defecto cuando se llama el bloque desde la interfaz de GNU, `< type >` define el tipo de variable que será ingresado para este parámetro y `< option >` permite listar posibles valores.

```
<param>
  <name>Enable</name>
  <key>enable</key>
  <type>int</type>
  <option>
    <name>True</name>
    <key>1</key>
  </option>
  <option>
    <name>False</name>
    <key>0</key>
  </option>
</param>

<param>
  <name>Gain</name>
  <key>gain</key>
  <value>1.0</value>
  <type>real</type>
</param>

<param>
  <name>Amplitude</name>
  <key>amplitude</key>
  <value>1.0</value>
  <type>complex</type>
</param>
```

Finalmente se asigna el puerto de salida con la etiqueta `< source >`, se define el tipo de datos permitidos con `< type >` y la longitud del vector con `< vlen >`,

el puerto de entrada no se define ya que en los niveles inferiores se dejó establecido que para este bloque no existe.

```
...  
  
<source>  
  <name>out</name>  
  <type>complex</type>  
  <vlen>$grvlen</vlen>  
  <domain>rfnoc</domain>  
</source>  
</block>
```

## B. Errores

Durante la ejecución del proyecto se presentaron problemas con el equipo de computo, lo que condujo a reinstalar el entorno *RFNoC*, ésta instalación descargó paquetes de la última actualización del entorno. Como se estaba trabajando con la versión anterior, se presentaron algunos errores con el bloque generado, que serán expuestos a continuación, con alguna alternativa de solución.

- **Error:** Si se realiza la actualización del entorno RFNoC, las operaciones realizadas con herramientas *RFNoC* que involucren a Vivado, generan errores.  
**Solución:** Se debe actualizar a la versión del vivado que satisfaga el requerimiento del entorno (actualmente versión 2017.4).
- **Error:** La ultima actualización de *RFNoC* (actual) requiere Xilinx Vivado tools version 2017.4 o mayor. Los bloques generados en la versión anterior se ejecutaban en Xilinx Vivado tools version 2015.4 o mayor pero menor a versiones 2017 por lo cual al momento de simular o intentar generar la imagen *.bit* se presentaron errores de sintaxis.  
**Solución:** Corregir el archivo o los archivos *.v* que componen el bloque *RFNoC* con las reglas de sintaxis de la versión de vivado que se este utilizando.
- **Error:** Cuando se estaba simulando el bloque del generador apareció el si-

guiente error:

```
ERROR: [Simtcl 6 – 50] Simulation engine failed to start:
A valid license was not found for simulation. Please run
the Vivado License Manager for assistance in determining
which features and devices are licensed for your system
```

Desde el *Vivado License Manager* se había copiado la licencia pero no fue reconocida, los métodos de licenciamiento de Vivado no solucionaron el error y al entrar a la carpeta *.Xilinx/* efectivamente no aparecía el archivo licencia *Xilinx.lic*.

**Solución:** Abrir el archivo *trial.lic* con

```
$ gedit .Xilinx/trial.lic
```

Estando en el *gedit* crear un archivo nuevo, copiar en éste el contenido del archivo *Xilinx.lic* descargado desde la página de *Xilinx* y nombrarlo de esta manera (*Xilinx.lic*). Al cerrar el *gedit* y abrir el *Vivado License Manager* aparecen los permisos para simulación de la licencia. Correr nuevamente la simulación. Si las fuentes *.v* y *.sv* no presentan errores la simulación será exitosa.

- **Error:** Si se instaló *Vivado* sin permisos de *root*, al generar la imagen *.bit* el siguiente *Critical Warning* hará que el proceso falle.

```
CRITICAL WARNING: [Common 17–183] Failed to open handle
vivado.jou. Please check access permission of directory
'/opt/Xilinx/Vivado/2017.4/bin'. You should restart the
application from a writable working directory.
```

Esto ocurre porque *Vivado* sólo tiene permisos de lectura más no de escritura.

**Solución:** Crear un *temp* que de accesos de lectura y escritura a *Vivado*. Ejecutar en la terminal:

```
mv /home/<username>/Xilinx/Vivado/2017.4
/home/<username>/Xilinx/Vivado/2017.4_temp

cd /home/<username>/opt/Xilinx/Vivado/2017.4/bin/vivado &
```

- **Error:** Cuando se elimina un bloque *RFNoC* que anteriormente había sido agregado a la interfaz para crear las imágenes .bit se genera un error si se intenta crear una nueva imagen, pues no encuentra las fuentes del bloque eliminado.

**Solución:** Eliminar la carpeta del bloque *RFNoC* creada y reiniciar el equipo. Con cerrar la interfaz y volverla abrir no se soluciona el problema, pues después de cerrarla ésta no vuelve a abrir debido a los errores en la búsqueda de las fuentes del bloque eliminado.

- **Error:** Creado el bloque en *RFNoC* y llevado a simulación en GNU Radio, aparece el siguiente error.

```
AttributeError: ''module'' object has not attribute ''<name_block
>''
```

**Solución:** Esto se debe a que no se le asignaron controladores al bloque del módulo al momento de crearlo, en la versión anterior estos no eran necesarios, ahora se deben agregar los bloques de control al momento de crear el bloque, no saltando la generación de estos.

```
RFNoC module name identified: <name_module>
Block/code identifier: <name_block>
...
Skip Block Controllers Generation? [UHD block ctrl files] [y/N] N
Skip Block interface files Generation? [GRC block ctrl files] [y/N
] N
```