

# **Instituto Tecnológico y de Estudios Superiores de Occidente**

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática  
**Maestría en Diseño Electrónico**



## **REPORTE DE FORMACIÓN COMPLEMENTARIA EN ÁREA DE CONCENTRACIÓN EN DISEÑO ELECTRÓNICO EN ALTA FRECUENCIA**

---

**TRABAJO RECEPCIONAL** que para obtener el **GRADO** de  
**MAESTRO EN DISEÑO ELECTRÓNICO**

Presenta: **IVONNE ALFARO FLORES**

Asesor: **OMAR HUMBERTO LONGORIA GANDARA**

Tlaquepaque, Jalisco. 31 de Marzo de 2021.



# Contenido

<b>Maestría en Diseño Electrónico .....</b>	<b>i</b>
<b>REPORTE DE FORMACIÓN COMPLEMENTARIA EN ÁREA DE CONCENTRACIÓN EN DISEÑO ELECTRONICO EN ALTA FRECUENCIA .....</b>	<b>i</b>
<b>1. Introducción .....</b>	<b>5</b>
<b>2. Resumen de los proyectos realizados .....</b>	<b>6</b>
2.1. CIRCUITO RECUPERADOR DE RELOJ Y DATOS .....	7
2.1.1 Circuito de decisión.....	8
2.1.2 Circuito de detección de fase, eliminación de glitches .....	8
2.1.3 Sincronía entre reloj y dato .....	9
2.1.4 Circuito de prioridad .....	9
2.1.5 Sincronización entre datos y reloj .....	10
2.1.6 Implementación y Resultados .....	10
2.1.7 Conclusiones .....	12
2.2. PROCESAMIENTO DIGITAL DE SEÑALES .....	12
2.2.1 Detección de imágenes manipuladas mediante el análisis biespectral .....	13
2.2.2 Implementación y Resultados .....	15
2.2.3 Conclusiones .....	16
2.3. DISEÑO DE UN FILTRO RECHAZA BANDA .....	16
2.3.1 Especificaciones de diseño.....	17
2.3.2 Simulación y Resultados .....	18
2.3.3 Conclusiones .....	19
<b>3. Conclusiones .....</b>	<b>20</b>
<b>4. Bibliografía .....</b>	<b>21</b>
<b>Apéndices .....</b>	<b>23</b>
APENDICE A .....	24
APENDICE B .....	38
APÉNDICE C .....	52



# 1. Introducción

El presente documento muestra los resultados de trabajos que se realizaron en áreas de concentración de electrónica avanzada como diseño de circuitos digitales, diseño en alta frecuencia y procesamiento digital de señales. Mediante el uso de herramientas de diseño electrónico como Cadence, Matlab y ADS se obtuvieron resultados para los siguientes proyectos:

- Diseño a nivel transistor para  $0.35\mu\text{m}$  de un circuito recuperador de datos.
- Uso de análisis biespectral para la detección de imágenes manipuladas
- Diseño de un filtro rechaza banda mediante líneas de transmisión

La importancia de profundizar en áreas de electrónica avanzada radica principalmente en que, los circuitos y sistemas electrónicos han sido la base para el desarrollo de los actuales sistemas de comunicación y procesamiento de información. El interés y necesidad creciente de sectores del ramo industrial, científico, comercial y doméstico está llevando a la electrónica a un proceso de desarrollo y avance continuo, en donde día a día se innovan tecnologías y se desarrollan conceptos y aplicaciones de la electrónica en múltiples campos interdisciplinarios.

En muchos sistemas la información se transmite o se recibe sin una referencia de la señal de reloj. Por ejemplo, en sistema de comunicación óptica (SONET), la información se transmite sin una señal de reloj que la acompañe, pero el receptor requiere que esta información se procese de forma síncrona. Por lo tanto, una señal de reloj se debe recuperar a partir de la información recibida. Este trabajo se presenta en el proyecto 1 para una tecnología de  $0.35\mu\text{m}$

La manipulación de imágenes es un problema creciente en el ámbito de casos de crimen y de disputas públicas. La manipulación de imágenes puede ser utilizada para promover un artículo en una revista, para la difamación de un político adversario, o para otros objetivos. El proyecto 2, consistió en crear un algoritmo para la detección de imágenes que son manipuladas de manera digital.

Las aplicaciones de sistemas de comunicación cubren un amplio rango de aplicaciones como localización de personas, predicción del tiempo, transmisión de información, comunicaciones satelitales. Todas estas aplicaciones requieren de la implementación de filtros cuya función es dejar pasar un rango definido de frecuencias con una pérdida mínima de potencia y con un nivel de distorsión bajo. El proyecto 3, presenta el diseño de un filtro Chebyshev rechaza banda usando líneas de transmisión.

La formación especializada que se cubre en la maestría de diseño electrónico ha facilitado la disposición de conocimientos y destrezas que coinciden con el perfil que es requerido en empresas de alta tecnología. Cada uno de los proyectos mencionados anteriormente han fortalecido las competencias propias de la profesión con habilidades como el análisis y la toma de decisiones

## 2. Resumen de los proyectos realizados

Durante mis estudios en la Maestría en Diseño Electrónico obtuve conocimientos teórico-prácticos que me permitieron abrirme paso en mi carrera profesional en el área de diseño de VLSI, validación eléctrica y diseño de hardware. Cada una de las materias cursadas me dio los fundamentos para analizar y finalizar de manera efectiva las áreas en donde me he desarrollado profesional y personalmente.

Las materias cubiertas en este reporte, por orden cronológico, son:

1. Diseño digital, donde se trabajó en el diseño de un circuito recuperador de datos. El trabajo cubrió simulaciones y layout del circuito elaborado
2. Procesamiento digital de señales, se exploró y trabajó en una técnica llamada análisis biespectral que nos permitió identificar imágenes que han sido trabajadas digitalmente, uno de los usos propuestos en el trabajo fue la detección de imágenes falsas.
3. Diseño en alta frecuencia, a través del uso de herramientas como ADS, se diseñó un filtro rechaza banda de orden 3 con resonadores tipo L usando líneas de transmisión  $\lambda/4$ . El trabajo requirió una segunda iteración de diseño, lo que permitió cumplir con las especificaciones de diseño tomando en cuenta el *tradeoff* o compromiso entre los parámetros como longitud de canal del transistor,  $L$ , y ancho de banda del filtro.

## 2.1. CIRCUITO RECUPERADOR DE RELOJ Y DATOS

Un circuito recuperador de datos y reloj, CDR en inglés, es ampliamente utilizado en el área de comunicaciones. Los datos transmitidos entre el receptor y transmisor son, la mayoría de las ocasiones, contaminados por señales de ruido tanto externas como internas lo cual es causa de *jitter* o *skew*. Además, en transmisiones de alta velocidad la comunicación entre el receptor y el transmisor se realiza de manera asíncrona. Por lo tanto, al finalizar la recepción de la información el uso de un CDR es necesario para extraer tanto los datos como la señal de reloj para éstos y eliminar fenómenos como *jitter*, *skew* o *glitches*. La información transmitida es procesada nuevamente con respecto al reloj recuperado

El CDR implementado en este proyecto tuvo como especificaciones de diseño el que trabajara a una frecuencia de 200 MHz, con una topología DDR (*Double Data Rate*). La tecnología es de 0.35  $\mu\text{m}$ . El bloque del VCO, que es el que provee las señales de reloj necesarias para la recuperación del dato, se programó usando VerilogA. No se hace uso del PLL, ya que en este proyecto no se planteó como objetivo. La Tabla 1 muestra con mayor detalle las especificaciones de diseño que el circuito CDR debe cumplir

<i>PVT</i>	<i>Worst Case</i>	<i>Typical</i>	<i>Best Case</i>
<i>Vdd(V)</i>	2.7	3	3.3
<i>T (°C)</i>	125	65	-40
<i>Frecuencia del reloj recuperado (MHz)</i>	200	200	200
<i>Datos recuperados</i>	2	2	2

Tabla 1 Especificaciones de diseño circuito CDR

La implementación del circuito recuperador de datos y reloj requirió el diseño de los siguientes bloques (Figura 1)

- Circuito de decisión
- Circuito de detección de fase, eliminación de glitches
- Circuito de prioridad
- Bloque para la Sincronía entre reloj y dato

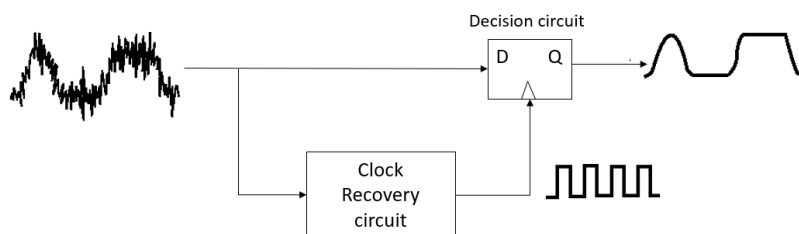


Figura 1 Implementación del CDR

### 2.1.1 Circuito de decisión

El primer bloque del que se compone el CDR es un circuito de decisión, éste lee el dato de entrada y genera una señal de reloj periódica además de que muestrea el valor de la amplitud de la señal en el instante  $t$  y decide el valor de este elemento. El circuito se describe en la Figura 2

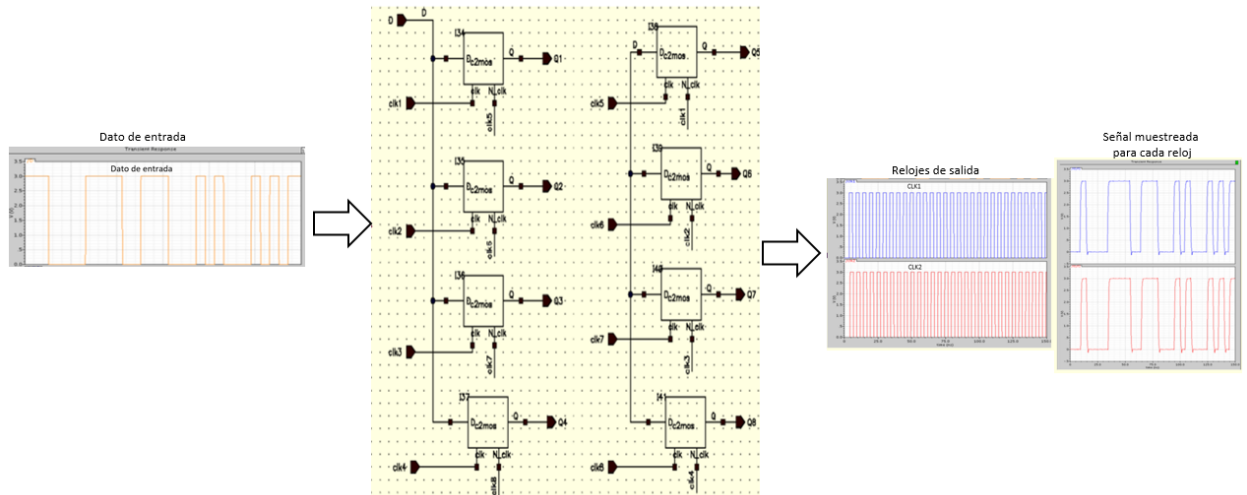


Figura 2 CDR - Circuito de decisión

### 2.1.2 Circuito de detección de fase, eliminación de glitches

Este bloque detecta las variaciones de fase entre el reloj de salida del VCO y la frecuencia del dato de entrada para así determinar si el PLL opera a su frecuencia establecida. Las salidas del módulo sirven como señales de entrada a un retenedor para eliminar los glitches que se generan en la compuerta; el circuito encargado de realizar esta eliminación de glitches y retención de datos consta de 8 Flip-Flop tipo D. Figura 3 y Figura 4 describen los diagramas de los circuitos implementados.

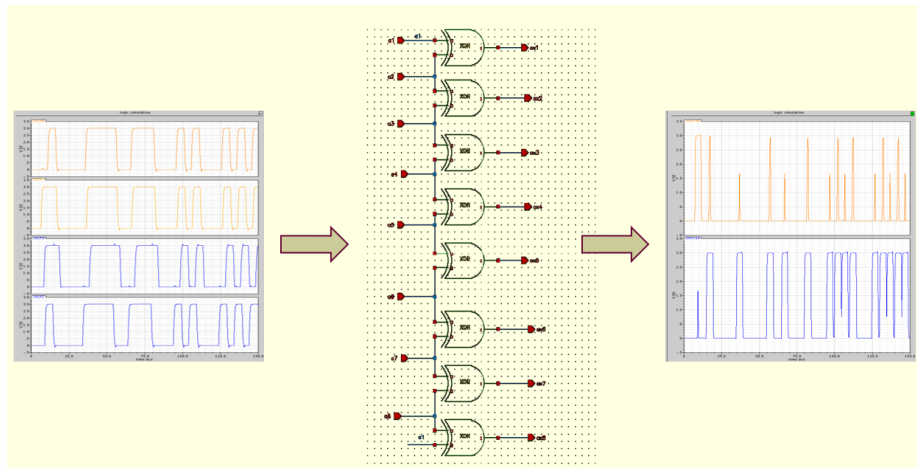


Figura 3 CDR- Detector de fase

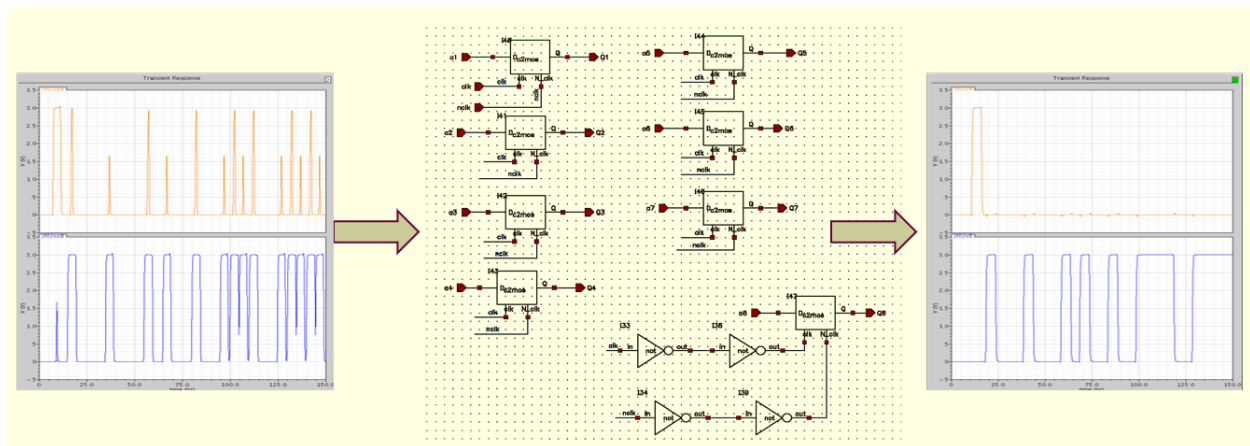


Figura 4CDR - Retención del dato y cambio de nivel

### 2.1.3 Sincronía entre reloj y dato

Una compuerta OR de 8 entradas 1 salida es la encargada de detectar en el ciclo de reloj correspondiente un cambio en los datos de entrada. Es decir, indica si por ejemplo en la señal del clk1 se detecta algún cambio en el dato de entrada que junto con su reloj asignado activa los Flip-Flop de memoria ya que su objetivo es guardar el valor de la entrada anterior

### 2.1.4 Circuito de prioridad

El objetivo de este bloque es dar prioridad al dato que llegue primero a sus terminales de entrada e ignorar los valores de las siguientes entradas, es decir si en la entrada de dato que corresponde al clk1 hay un cambio de nivel el circuito deshabilita todos los demás valores y ya no

seguir leyendo el dato de las demás entradas, esto lo hace para los siguientes 7 datos de entrada. Este circuito es mostrado en la Figura 5

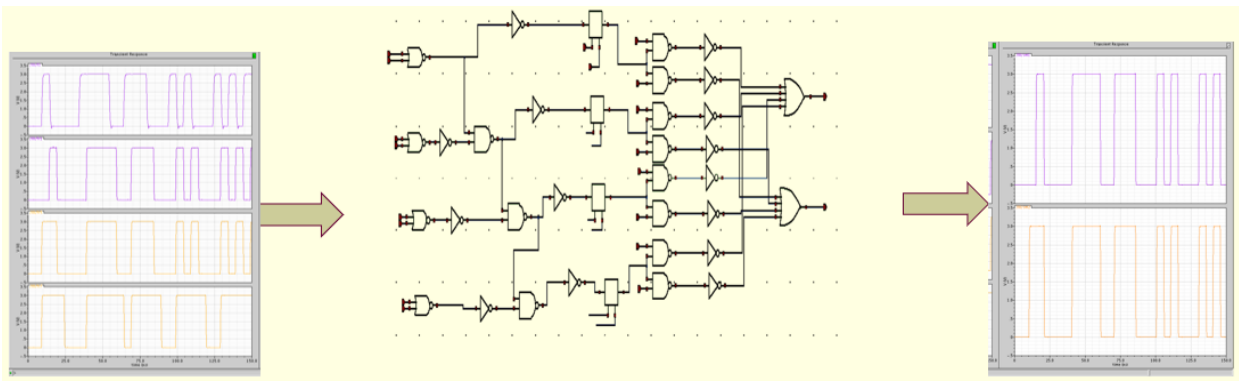


Figura 5 CDR - Circuito de prioridad

### 2.1.5 Sincronización entre datos y reloj

A la entrada del circuito de prioridad se tiene un bloque el cual almacena la señal de reloj que corresponde para cada señal de entrada. Trabaja en sincronía con los relojes de salida clk1 y clk5 para que junto con el bloque de mayor precedencia sincronicen el dato recuperado con su señal de reloj.

Finalmente, las salidas de este bloque van a un módulo que sincroniza las señales de reloj generadas con el dato recuperado correspondiente Figura 6

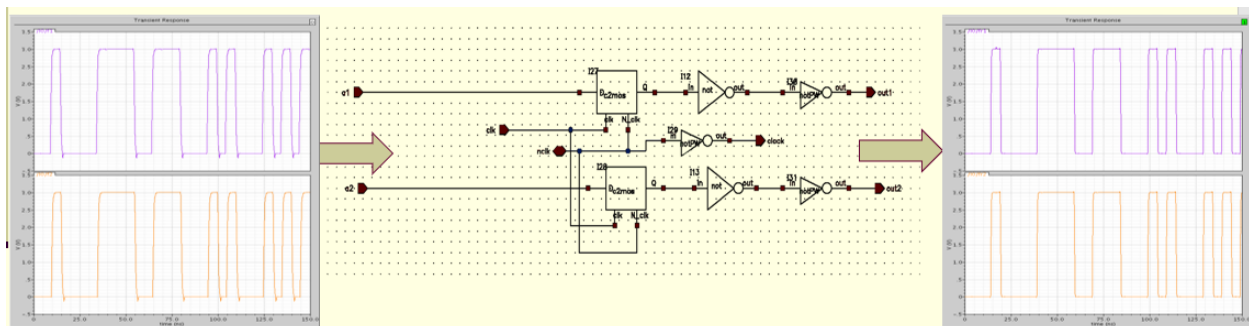


Figura 6 CDR - Sincronización entre datos y reloj

### 2.1.6 Implementación y Resultados

El esquema VCO – CDR se muestra en la Figura 7. De igual forma en la Figura 8 se muestran los bloques internos de los que se compone el circuito recuperador de datos

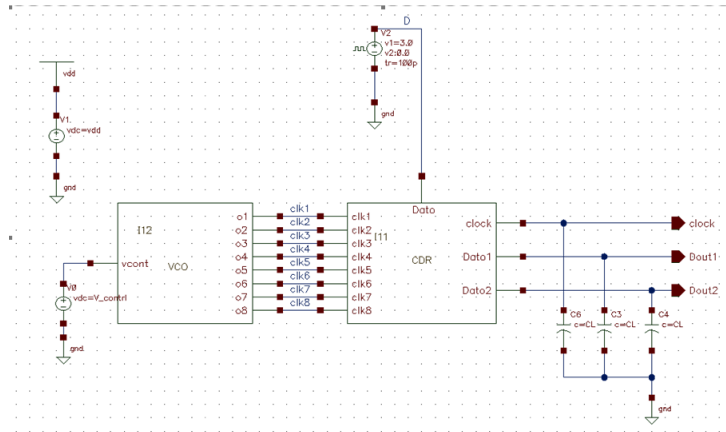


Figura 7 CDR - Implementación CDR con VCO

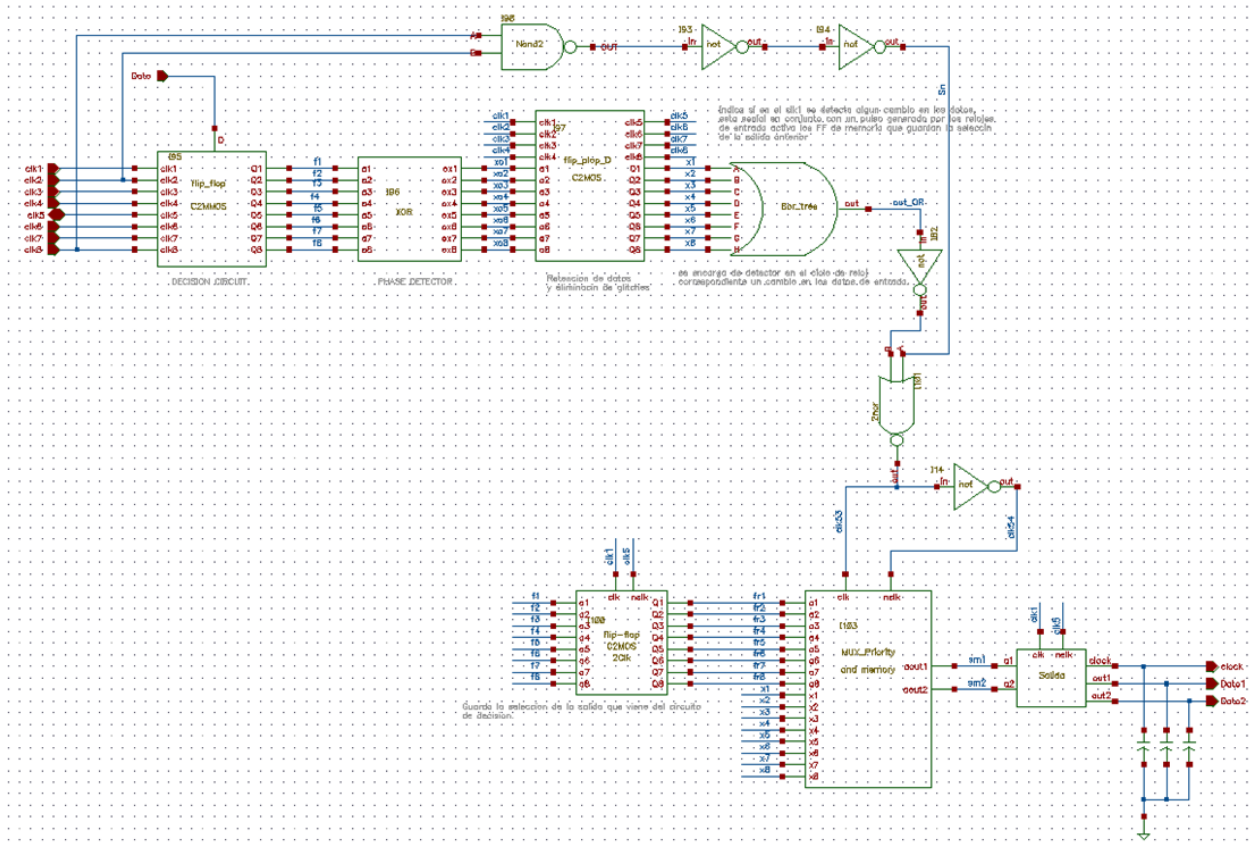


Figura 8 CDR - Implementación final

Los resultados del análisis de potencia se muestran en la Tabla 2

Potencia pre-layout (mW)			Potencia post-layout (mW)		
WC	TC	BC	WC	TC	BC
5.306	6.915	9.384	5.406	7.306	10.743

*Tabla 2 CDR - Análisis de potencia*

### 2.1.7 Conclusiones

Algunas de las ventajas del diseño implementado es que al tener una topología DDR se tenga una doble transferencia durante un ciclo de reloj permitiendo alcanzar un mayor rango de frecuencia de operación lo que lo hace muy útil para comunicaciones de alta frecuencia.

La mayor desventaja de este diseño es la falta de un lazo de retroalimentación por lo que no se tiene una señal de reloj predefinida.

Durante la elaboración de este proyecto, se definieron algunos trabajos futuros para las siguientes generaciones que tomen el curso de diseño digital. El principal trabajo es la implementación de un lazo de retroalimentación para que el diseño sea capaz de sincronizar el dato de entrada con cualquier señal de reloj.

## 2.2. PROCESAMIENTO DIGITAL DE SEÑALES

Este curso abarcó temas sobre el análisis y procesamiento de señales digitales, análisis de señales y sistemas en tiempo discreto, transformada discreta de Fourier, transformada Z, diseño de filtros, procesamiento de voz, procesamiento de imágenes y procesamiento de video.

El proyecto que se realizó tuvo como meta inicial realizar una investigación sobre las técnicas de procesamiento de audio, imágenes y video para después proponer una nueva metodología o algoritmo que resolviera problemas relacionados con estas áreas como puede ser la creación de imágenes falsas.

El proyecto seleccionado consistió en detectar imágenes que han sido manipuladas digitalmente; la técnica que se usó fue el análisis biespectral.

## 2.2.1 Detección de imágenes manipuladas mediante el análisis biespectral

La manipulación de imágenes es un problema creciente en el ámbito de casos de crimen y de disputas públicas. La manipulación de imágenes puede ser utilizada para promover un artículo en una revista, para la difamación de un político adversario, o para otros objetivos. Las herramientas forenses para el análisis de imágenes digitales son utilizadas para ayudar a resolver crímenes. El desarrollo de estas herramientas es esencial para el futuro del análisis forense de imágenes manipuladas, como la mostrada en la Figura 9.



Figura 9 Fomontaje de Saddam Hussein con Bill Clinton en la Casa Blanca. (Tomado de Fridrich, Soukal, & Lukás, 2003)

Una técnica utilizada para determinar la autenticidad de una imagen digital es conocida como “firma” o “marca de agua”, ésta consiste en agregar una codificación a la imagen (la cual no es visible a simple vista), y que es posible decodificarla posteriormente. Sin embargo, esta técnica presenta desventajas porque la mayoría de las cámaras digitales no agregan la marca de agua, incluso esta técnica tampoco funcionaría para aquellas imágenes provenientes de una digitalización.

Existen otras técnicas para la detección de imágenes manipuladas, una de ellas consiste en detectar inconsistencias entre bloques generados durante la compresión de una imagen JPEG.

Sin embargo, esta técnica también tiene limitantes, si la imagen se vuelve a codificar después de su manipulación, el algoritmo de detección se vuelve vulnerable (S. Ye, 2007)

Durante la investigación de las distintas técnicas de modificación de imágenes se optó por utilizar el análisis biespectral, pues al no depender de una codificación o de una marca de agua, sino únicamente de la información de la señal misma, parece ser la más robusta. Este análisis biespectral es una técnica utilizada para el análisis de no linealidades causadas por un sistema no natural.

Cuando una señal pasa a través de un sistema no lineal, se crean armónicos a frecuencias y fases correlacionadas (las cuales no son creadas por sistemas lineales). El objetivo del análisis biespectral es medir y cuantificar dichas correlaciones de alto orden de frecuencia y fase. Para el análisis de señales en el dominio de frecuencia normalmente se recurre al “espectro de potencia”. Sin embargo, esta técnica únicamente es capaz de detectar la presencia de correlaciones de hasta segundo orden y no de orden mayor, las cuales son de interés para nuestro estudio. Para solucionar este problema se recurre al biespectro, el cual es capaz de detectar correlaciones de tercer orden.

Una vez obtenido el biespectro se procede a obtener la bicoherencia, el cual es la versión normalizada del biespectro. Cuando una no linealidad es detectada, el espectro de potencia no varía, pero la magnitud e histograma de fase de la bicoherencia sí. Figura 10

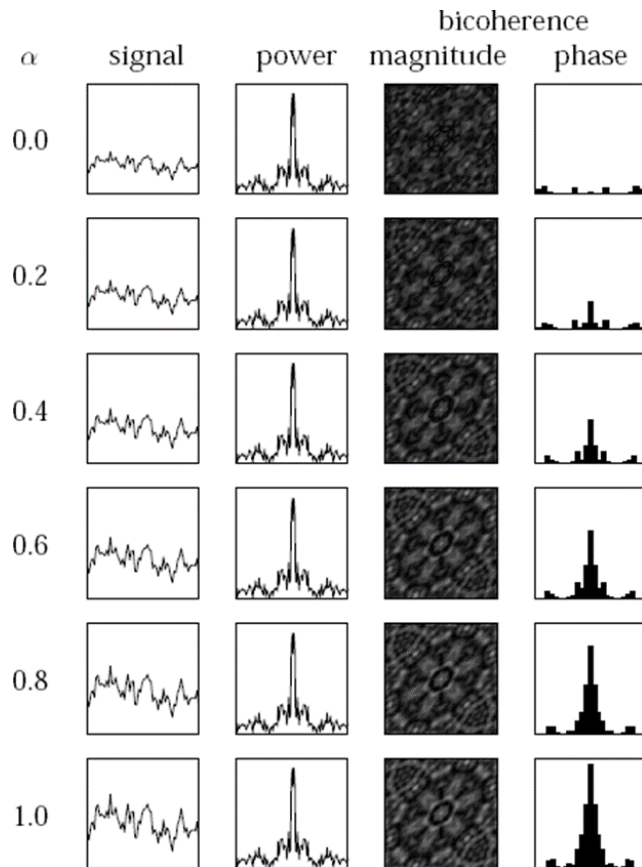


Figura 10 Comparación entre espectro de potencia y magnitud y fase de la bicoherencia de una señal  $x(n)$  sometida a un sistema no lineal  $y(n) = \alpha x^2(n) + x(n)$ . (Tomado de Farid, 1999)

Una de las dificultades de este proyecto consistió en que las imágenes son señales bidimensionales y el cálculo de la bicoherencia de la imagen completa requiere mayor capacidad de procesamiento. Para ello se probó un nuevo algoritmo; éste consistió en dividir la imagen en múltiples vectores horizontales y verticales, posteriormente se calculó la bicoherencia de cada uno de los vectores, finalmente se promediaron los valores de la magnitud de la bicoherencia. Este proyecto se implementó con la ayuda de la herramienta de simulación matemática Matlab, además fue necesaria la instalación de una librería llamada HOSA (High-Order Spectral Analysis) para la obtención de la bicoherencia.

### 2.2.2 Implementación y Resultados

La detección de imágenes mediante esta técnica se implemento de la siguiente manera:

Se crearon dos imágenes de prueba. La primera se dividió en cuatro secciones para después invertir dos de ellas de manera diagonal; la segunda consistió en sustituir un pequeño segmento de la imagen con una copia de otra sección de la imagen. Figura 11



Figura 11 Imágenes de prueba: Lena cruzado, y Lena desplazado

Posteriormente se realizaron los cálculos de la bicoherencia mediante dos métodos. El primero de ellos consistió en aplicar el cálculo de la magnitud de la bicoherencia directamente al archivo de imagen y obtener su valor máximo; el segundo método consistió en segmentar la imagen en pequeños vectores horizontales y verticales, para después calcular el promedio de la magnitud de la bicoherencia máxima de cada segmento. La Tabla 3 compara los resultados de las mediciones de la magnitud de la bicoherencia por ambos métodos.

Imagen	Método 1	Método 2
	Valor máximo de la bicoherencia de la imagen completa	Promedio de los valores máximos de la bicoherencia de cada segmento de la imagen
Lena original	0.4430	0.7811
Lena cruzado	0.8029	1.0315
Lena desplazado	0.4434	1.0854

*Tabla 3 Resultados de la magnitud de la bicoherencia*

Se puede observar que a la hora de cruzar los segmentos de imagen provocaron un aumento significativo en la magnitud de la bicoherencia; mientras que en el caso del 20 desplazamiento de una pequeña sección de la imagen, ocasiona también un pequeño incremento en la magnitud de la bicoherencia.

### 2.2.3 Conclusiones

Pese a que se pudo comprobar la teoría con la práctica, aun se cuestiona la eficiencia de este método si consideramos que la magnitud de la bicoherencia de la imagen original es desconocida, por lo tanto, una pequeña alteración de la imagen ocasionaría un mínimo cambio en la magnitud de la bicoherencia.

Cabe resaltar que esta técnica es mayormente usada en archivos de audio, se partió del mismo principio y se adaptó para que fuera capaz de detectar imágenes que han sido manipuladas digitalmente.

Este proyecto fue por demás retador, ya que implicó un arduo trabajo de investigación para poder definir la técnica a usar e implementar, además de que como resultado principal fue la redacción de un artículo técnico, que, aunque no se publicó, cumplía con los requisitos de la IEEE. Sin dejar a un lado que el uso de herramientas de simulación fue una constante durante este trabajo de investigación.

## 2.3. DISEÑO DE UN FILTRO RECHAZA BANDA

Las aplicaciones de los sistemas de comunicación actuales van desde el radar, transmisión de información mediante enlaces terrestres y espaciales, sistemas inalámbricos de comunicaciones entre otras más. Todas estas aplicaciones requieren la implementación de filtros que dejen pasar o

rechacen un cierto rango definido de frecuencias con una pérdida mínima de potencia y con un nivel de distorsión bajo.

### 2.3.1 Especificaciones de diseño

El objetivo de este proyecto es la implementación de un filtro rechaza banda Chebyshev de tercer orden. La frecuencia de rechazo es de 3.4GHz con un ancho de banda del 5%. Las especificaciones de la microcinta son Rogers RO4003 con una altura  $h = 0.81 \text{ mm}$ . El filtro por diseñar está basado en una topología  $\lambda/4$  resonador-L.

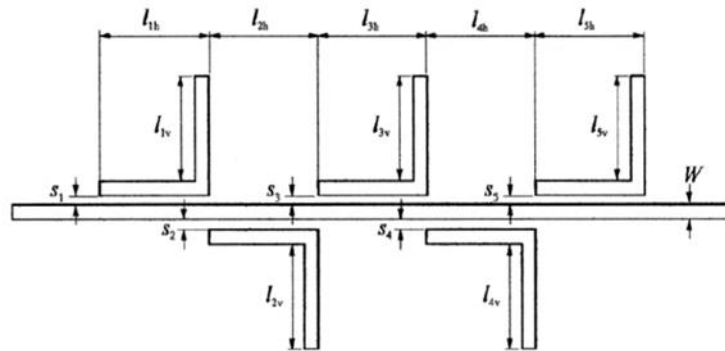


Figura 12 Topología resonador-L

El filtro propuesto se basa en una topología Shunt, los valores de L y C se muestran en la Tabla 4

	L (H)	C (F)
1	4.5376e-08	4.8289e-14
2	1.3428e-10	1.6319e-11
3	4.5376e-08	4.8289e-14

Tabla 4 Filtro Rechaza banda - Element value

Para estos valores de L y C y gracias al uso de ADS se pueden obtener las dimensiones del transistor,  $W=1.811700\text{mm}$  y  $L=13.154\text{mm}$ . Con estas características la especificación de frecuencia central no logró alcanzarse, lo que hace necesario un ajuste en la longitud del ancho del canal del transistor. En base a una serie de iteraciones manuales, el valor de L para lograr  $f_0$  es de  $L=12.7\text{mm}$ .

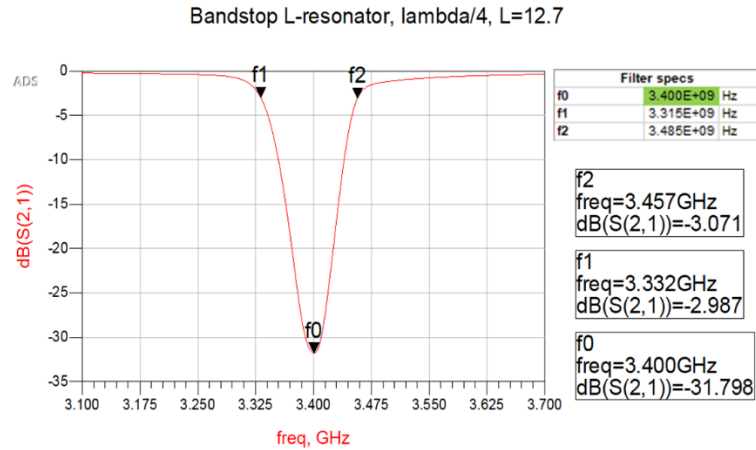


Figura 13 Filtro rechaza banda lambda/4 resonador-L

### 2.3.2 Simulación y Resultados

La Figura 14 muestra el diseño del filtro simulado mediante ADS.

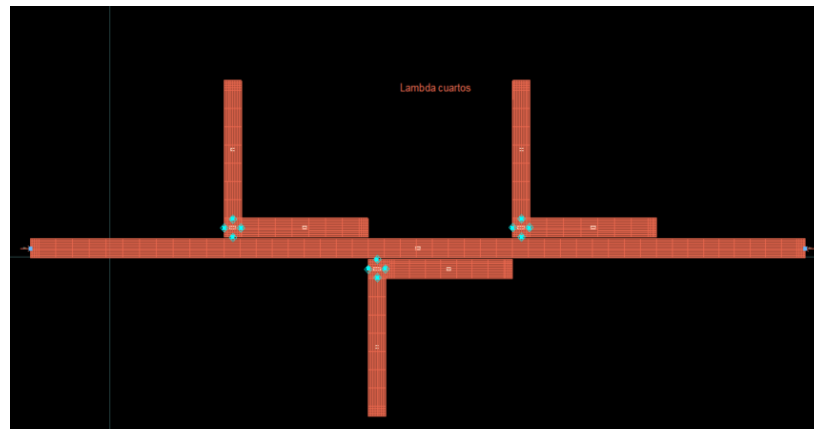


Figura 14 Filtro rechaza banda - ADS

Gracias al ajuste que se hizo en  $L$ , se logro cumplir con los valores de frecuencia central pero no de ancho de banda. Esta optimización se propone como trabajo futuro.

La Tabla 5 muestra una comparación entre la especificación de diseño y los resultados logrados en simulación, como anteriormente se mencionó, tuvo que hacerse un tradeoff entre frecuencia central y ancho de banda. Se eligió cumplir con la especificación de frecuencia ya que el ajuste de ancho de banda requiere una serie de optimizaciones que esta fuera del contenido del curso.

<b>Resonador – L, <math>\lambda/4</math>, n=3</b>			
<b>Parámetro</b>	<b>Simulado</b>	<b>Especificación</b>	<b>Unidades</b>
f0	3.40	3.40	GHz
f1	3.33	3.32	GHz
f2	3.45	3.49	GHz
Delta (3dB)	125.00	170.00	MHz

*Tabla 5 Filtro rechaza banda, simulado vs especificación*

### 2.3.3 Conclusiones

Uno de los retos del presente proyecto fue cumplir con las especificaciones de frecuencia de operación con respecto a la geometría del filtro,  $S$ . Para lograr cumplir con tal especificación fue necesario un ajuste en la longitud del canal del transistor  $L$  lo que implica un cambio en la constante dieléctrica del material.

Una reducción en  $L$  ayuda a cumplir con el requerimiento de frecuencia de operación y a disminuir la diferencia entre el ancho de banda esperado vs simulado.

Como trabajo futuro se propone la optimización del diseño para poder cumplir con la especificación del ancho de banda, ya que este trabajo requiere de ciertas iteraciones que solo son viables mediante el uso de herramientas de optimización y se encuentra fuera del alcance del curso de diseño de circuitos en alta frecuencia.

### 3. Conclusiones

Cada uno de los conocimientos adquiridos en los trabajos mostrados en este reporte, permiten tener cimientos sólidos para poder desarrollarse en cualquier área de diseño de circuitos electrónicos, ya que brindan fundamentos teóricos y conocimientos prácticos sobre el uso de herramientas CAD que permiten analizar y facilitar la toma de decisiones sobre los compromisos o *tradeoffs* que deben hacerse mientras se diseñan, verifican o validan circuitos digitales y/o analógicos para cumplir con las especificaciones de diseño propuestas inicialmente. Además, estos conocimientos, permiten explorar y trabajar en técnicas de análisis de información de datos para, por ejemplo, la detección de trabajos manipulados de manera digital.

El diseño de circuitos en alta frecuencia es un tema que cada día cobra mayor relevancia y que es importante entender y analizar los efectos en el diseño de circuitos. El curso de diseño en alta frecuencia me permitió conocer y entender los conceptos que pudieran afectar la integridad de señales en una plataforma de validación y como pueden ser minimizados mediante la implementación de filtro rechaza banda.

El trabajo de CDR me permitió tener fundamentos teóricos-prácticos sobre el diseño de VLSI permitiéndome entender cuales son las etapas y herramientas de diseño que se usan a nivel transistor. Este conocimiento me ha permitido trabajar con grupos de alto desempeño orientados 100% al diseño digital.

Este documento pretende ser una guía que facilite la escritura del documento del reporte. Puedo ser utilizado para trabajos de nivel licenciatura o de nivel posgrado. El documento es revisado con cierta frecuencia y se procura mejorarlo siempre que sea posible. Si el lector encuentra alguna inconsistencia o error, o bien encuentra alguna forma de mejorar la implementación de alguna funcionalidad del documento, mucho le agradeceré su retroalimentación.

## 4. Bibliografía

(s.f.). Obtenido de [www.rogerscorp.com](http://www.rogerscorp.com).

Jia-Sheng Hong, M. J. (s.f.). *Microstrip Filters for RF/Microwave Applications*.

Pozar, D. M. (s.f.). *Microwave Engineering*.

S. Ye, Q. S.-C. (2007). Detecting digital image forgeries by measuring inconsistencies of blocking artifact. Institute for Infocomm Research Singapore.

Zabdiel, B. B. (s.f.). *ITESO*. Obtenido de O2020\_MDE2546A Material Course: <https://canvas.iteso.mx/courses/9073>



## **Apéndices**

## **APENDICE A**

*CIRCUITO RECUPERADOR DE RELOJ Y DATOS*

# Circuito recuperador de reloj y datos

## OBJETIVO

Diseñar a nivel transistor para tecnología de 0.35  $\mu\text{m}$  un circuito recuperador de datos.

## INTRODUCCIÓN

En muchos sistemas la información se transmite o se recibe sin una referencia de la señal de reloj. Por ejemplo, en sistema de comunicación óptica (SONET), la información se transmite sin una señal de reloj que la acompañe, pero el receptor requiere que ésta información se procese de forma síncrona. Por lo tanto, una señal de reloj se debe recuperar a partir de la información recibida.

## ARQUITECTURA INTERNA DEL CDR

### **Datos aleatorios**

Para generar los datos de entrada se utilizó un programa generador de pulsos aleatorios en VerilogA mediante la instrucción random, que va a ser el dato a recuperar, este se muestra en la figura 1



Figura 1. Señal de los datos aleatorios

### **Oscilador controlado por voltaje**

El VCO es un dispositivo no lineal que genera una salida periódica. La frecuencia de ésta señal se fija a través de un voltaje de control. Para nuestro caso cuando el voltaje de control es de 0.5 V se tiene una señal con una frecuencia de 60 MHz y para 200 MHz dicho voltaje de control es de 2.5 V.

Este bloque fue diseñado en verilogA para tener las señales de reloj, su necesidad radica en el hecho de que el sistema desarrollado no es retroalimentado, es decir, debido a la falta de un PLL o DLL no se puede generar una señal de reloj a partir de la información obtenida del CDR; sino que el reloj con el cual sincronizar los datos ya está establecido desde el inicio mediante el módulo del VCO.

Las señales de salida de éste bloque se muestran en la figura 2

En la primera ventana se observa el valor del ancho de pulso mínimo del dato de entrada, este valor es de 5ns.

A partir de [1] el reloj generado debe tener una frecuencia igual al ancho de pulso; es decir si el ancho de pulso mínimo es de 5 ns entonces el reloj debe tener una frecuencia de 200MHz. Como se muestra en la figura 2, el periodo es de 5ns, el tiempo de subida y bajada para esta tecnología es de 100 ps (figura 3) para cada flanco de subida o bajada

Para que los 8 relojes generados a partir del VCO puedan alcanzar a leer este dato el valor entre flanco de subida de clk1 y clk8 deben entrar los clk2 hasta clk7. Esto se muestra en las ventanas 2 y 3 de la figura 2.

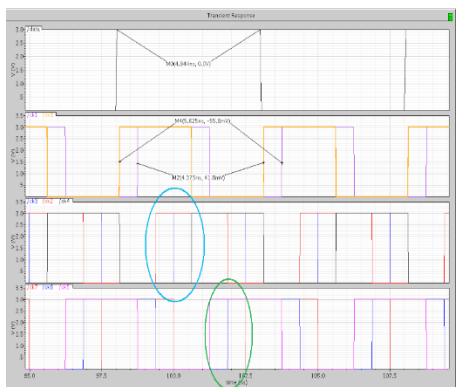


Figura 2. Señales de reloj generadas por el módulo del VCO

Los niveles de voltaje de salida se muestran en la siguiente figura 3

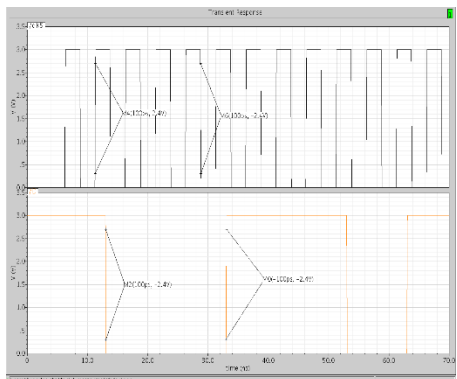


Figura 3. Tiempos de subida y bajada de las señales de reloj del VCO

El circuito de prueba del módulo del VCO es el siguiente;

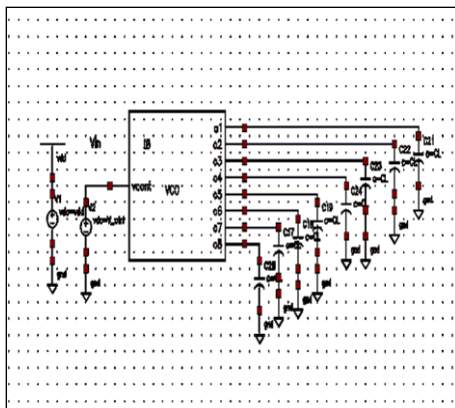


Figura 4. Testbench para el VCO

### Circuito de decisión

Para poder tener un comportamiento síncrono en el muestreo del dato aleatorio los receptores deben generar una señal de reloj para cada dato de entrada.

Como se ve en la figura 5 un circuito recuperador de reloj lee el dato de entrada y genera una señal de reloj periódica. Este comportamiento se puede lograr mediante el uso de un flip flop tipo D.

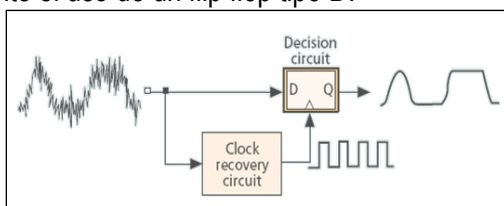


Figura 5. Comportamiento ideal de un circuito

Una vez generados los datos y los 8 pulsos de reloj, se llevan a un arreglo de flip flop tipo D con 8 señales de reloj y una entrada. Éste módulo asigna el dato aleatorio a una de las señales de reloj. A su salida muestra 8 datos donde cada uno ya se encuentra sincronizado con una señal de reloj. Además mediante éste bloque se asegura que los datos generados en el flanco de subida vean el dato con el ancho de pulso más pequeño de entre todos los datos aleatorios generados.

El circuito a nivel transistor se muestra en la figura 6

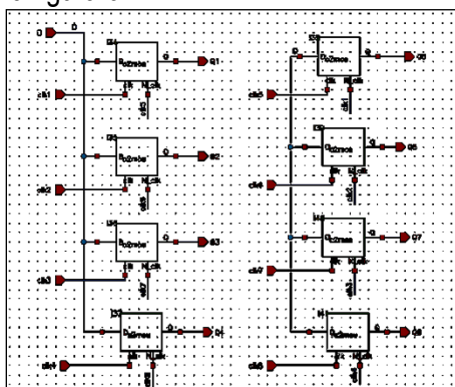


Figura 6. Circuito de decisión

### ***Detector de fase***

El bloque de detección de fase consiste en una compuerta XOR, detecta las variaciones entre el dato sincronizado con la señal del reloj y la velocidad de entrada del dato. Asimismo, la señal de salida indica cual es el dato que entra primero al detector de fase.

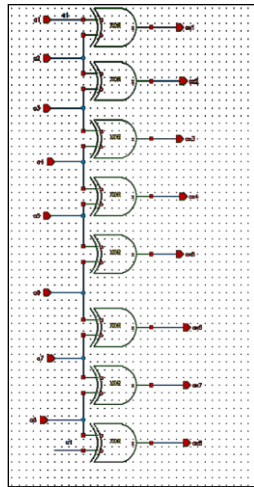


Figura 7. Circuito de detección de fase

### ***Eliminación de glitches, retención del dato y cambio de dato***

Las salidas de este bloque entran a una etapa encargada de retener los datos de entrada y eliminar los glitches generados por las transiciones de las XOR, se muestra en la figura 8

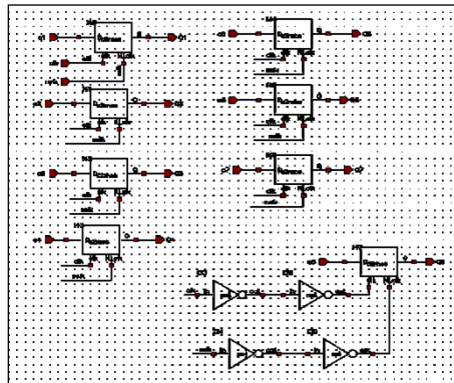


Figura 8. Retención de datos y eliminación de glitches

Estos datos entran en una compuerta OR de 8 entradas una salida la cual se encarga de detectar en ese ciclo de reloj un cambio en los datos de entrada en caso de que exista.

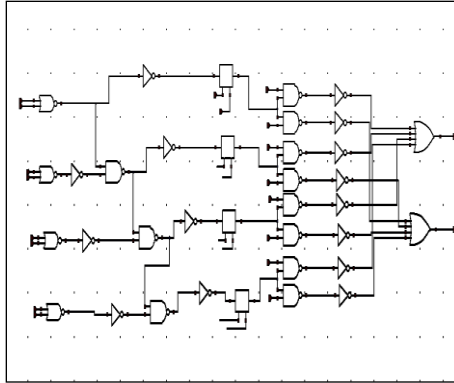


Figura 9. Circuito de sincronización

### **Circuito de prioridad**

Éste circuito detecta el dato que llega primero a alguna de sus entradas, estos datos se obtienen a partir del circuito de detección. En caso de que en la entrada 1 se detectará un flanco de subida entonces el circuito le da prioridad a ésta entrada ignorando el valor de las entradas restantes.

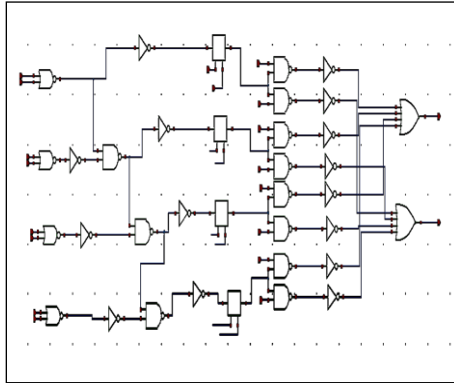


Figura 10. Circuito de prioridad

### **Circuito de retención**

Para este módulo el objetivo a cumplir es que retenga el dato que se obtiene del circuito de decisión para sincronizarlo con las señales de reloj asignada para cada entrada.

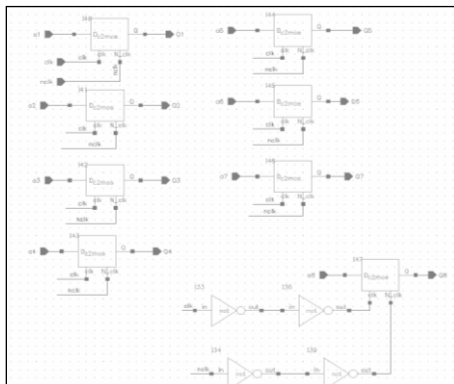


Figura 11. Circuito de retención

## Bloque de salida

El objetivo de este módulo es tener el dato recuperado junto con su señal de reloj, son dos las señales que esta unidad entrega, el primer dato se sincroniza con los flancos de subida de la señal de reloj, mientras que el dato 2 lo hace para los flancos de bajada.

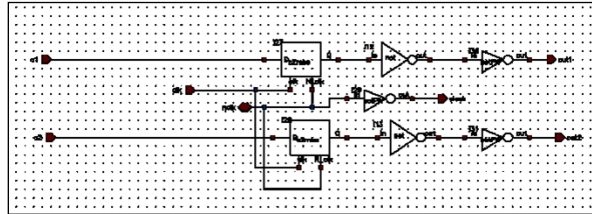


Figura 12. Sincronización datos - reloj

Finalmente, el circuito recuperador de información completo queda como sigue;

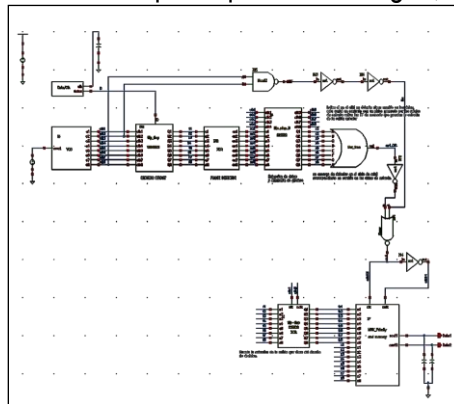


Figura13. Circuito recuperador de datos y reloj

## RESULTADOS

El layout que se realizó solo fue del circuito de prioridad, ya que es éste el que entrega finalmente la señal del dato sincronizada con su reloj respectivo.

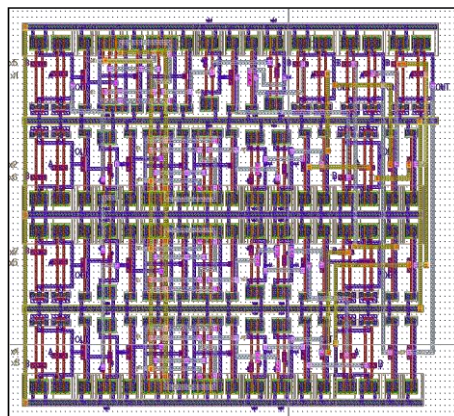


Figura 14. Layout del circuito de prioridad

Un análisis prelayout entrega los siguientes resultados

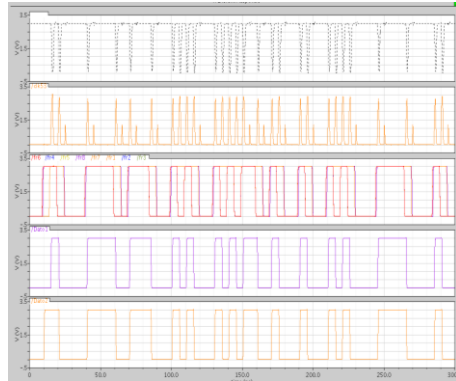


Figura 15. Análisis prelayout

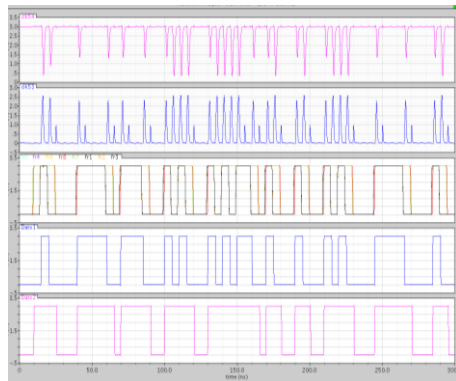


Figura 16. Análisis post layout

La extracción de las capacitancias parásitas arroja el siguiente resultado;

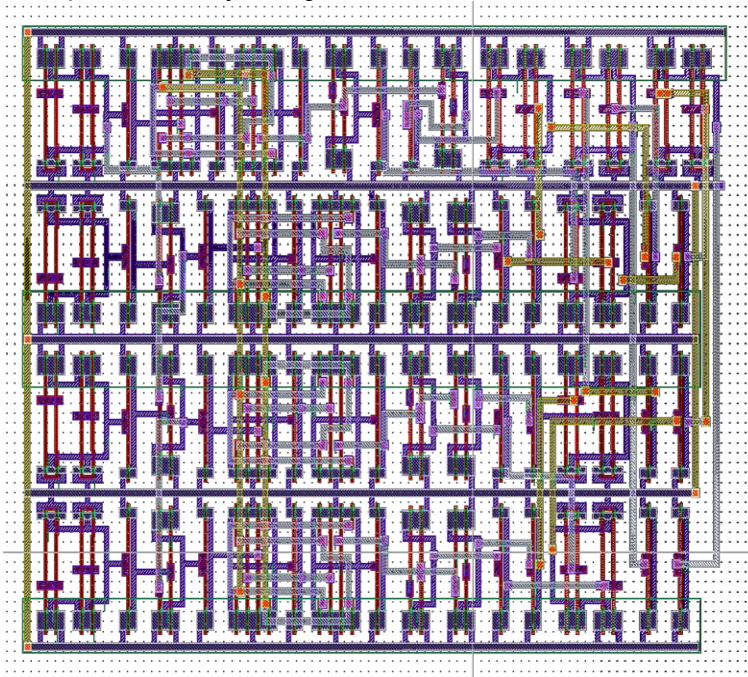


Figura 17. Capacitancias parásitas (líneas en verde)

Para las pruebas PVT a diferentes valores se muestran en las siguientes figuras.

Para el caso típico

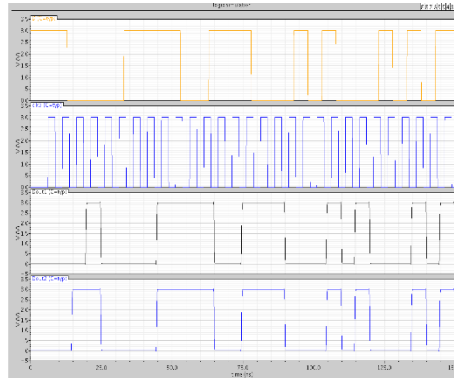


Figura 18. Respuesta del CDR para un caso típico

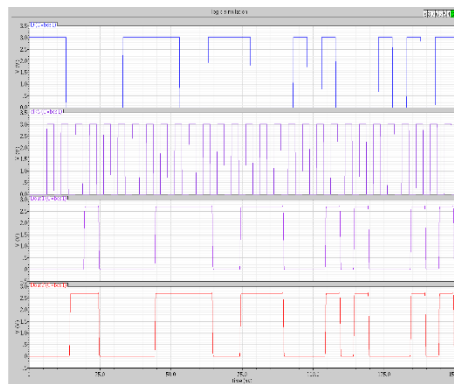


Figura 19. Respuesta al mejor de los casos

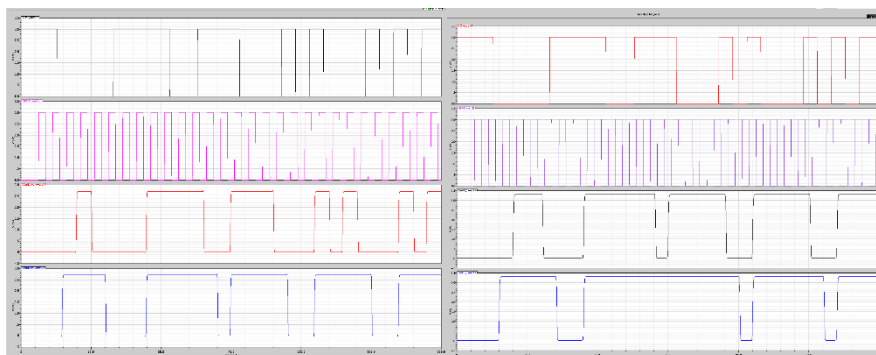


Figura 20. Respuesta del CDR para un caso extremo

En la tabla 1 se muestran los valores de potencia para los análisis pre y post layout. Aunque como se muestra en la figura 21 en este caso la mejor respuesta de potencia se obtuvo para uno de los peores casos, el mejor valor de potencia que se obtuvo fue de 5.306 mW en comparación con los otros valores para diferentes casos.

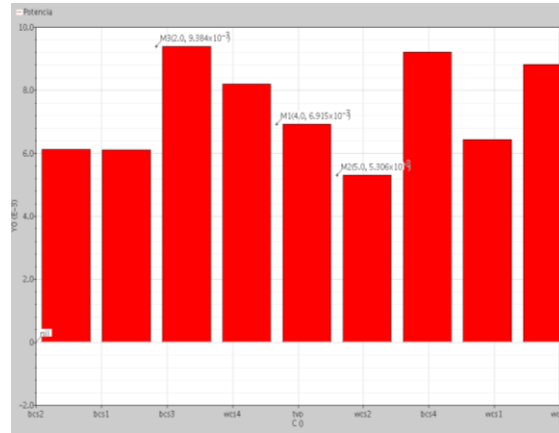


Figura 21. Valores de potencia para diferentes PVT

Potencia pre-layout (mW)			Potencia post-layout (mW)		
WC	TC	BC	WC	TC	BC
5.306	6.915	9.384	5.406	7.306	10.743

Tabla 1. Análisis de potencia

## CONCLUSIONES

El uso de un recuperador de datos y reloj es de gran ayuda sobre todo cuando se tiene que enviar y/o recibir información a grandes velocidades que es en donde la información es contaminada con señales de ruido.

El uso de retroalimentación no se planteó como objetivo para este proyecto y debido a esta especificación, el reloj con el cual los datos serían sincronizados fueron generados en verilogA, esto limita el uso de este proyecto ya que en la vida real a partir de la información recuperada es de donde se obtienen los datos sincronizados con una señal de reloj.

Debido a la falta de acoplamiento ideal entre transistores los tiempos de retardos entre etapas no fueron los mismos, lo que hizo necesario el uso de buffer lo que aumentaba el tiempo de retardo entre los relojes y datos. Si el acoplamiento entre transistores fuera el ideal éstos retardos se presentarían pero no causando un efecto marcado en los transistores.

# Circuito recuperador de reloj y datos

Ivonne Alfaro, Martín González  
Especialidad en diseño de circuitos integrados  
Instituto Tecnológico de Estudios Superiores de Occidente

## Resumen

Un circuito recuperador de datos y reloj es ampliamente utilizado en el área de comunicaciones. El CDR implementado en este proyecto funciona a una frecuencia de 200 MHz, la topología utilizada para la recuperación del dato es DDR (double data rate). No se hace uso del PLL, ya que en este proyecto no se planteó como objetivo. EL CDR fue implementado en una tecnología de 0.35  $\mu\text{m}$  consta de un VCO programado en verilogA que es el que provee las señales de reloj necesarias para la recuperación del dato.

## I. INTRODUCCIÓN

Un recuperador de datos (CDR) juega un papel importante en la comunicación de información. Los datos transmitidos entre el receptor y transmisor son, la mayoría de las ocasiones, contaminados por señales de ruido tanto externas como internas lo cual es causa de jitter o skew. Además, en transmisiones de alta velocidad la comunicación receptor-transmisor se realiza de manera asincrónica. Por lo tanto, al finalizar la recepción de la información el uso de un CDR es necesario para extraer tanto los datos como la señal de reloj para éstos y eliminar fenómenos como jitter, skew o glitches. La información transmitida es remedida en el tiempo de acuerdo con el reloj recuperado.

En la referencia [1] se habla acerca de los retos que existen en el diseño de un CDR y su arquitectura para transmisiones de alta velocidad.

En la segunda parte de éste artículo se explica la arquitectura propuesta para un CDR. El sistema fue puesto a prueba para distintas esquinas PVT.

## II. ARQUITECTURA DEL CDR

El diseño del CDR debe cumplir con las siguientes especificaciones:

Tabla 6. Especificaciones de diseño

PVT	WCS vel	WCS Pot	TYP	BCS
Vdd(V)	2.7	3.3	3	3.3
T (°C)	125	125	65	-40
Frec del reloj recuperado(MHz)	200	200	200	200
Datos recuperados	2	2	2	2

En este caso el dato de entrada es generado de manera aleatoria por un módulo construido en verilogA, por lo que la información a recuperar no presenta señales de ruido.

Las señales de reloj necesarias para recuperar el dato son generadas en un script desarrollado en verilogA. Éstas salidas pertenecen al VCO y van directamente conectadas a las entradas del CDR.

### Circuito de decisión

Para poder tener una sincronía en el muestreo de un dato aleatorio los receptores deben de ser capaces de generar una señal de reloj.

El primer bloque del que se compone el CDR es un circuito de decisión, éste lee el dato de entrada y genera una señal de reloj periódica además de que lee el valor de la amplitud de la muestra en el instante t y decide el valor de este elemento, el circuito se muestra en la figura 1.

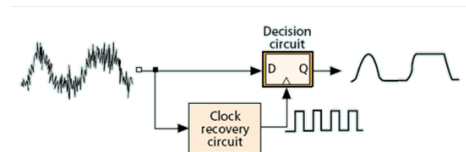


Figura 1. Ejemplo de la implementación del CDR

En la figura 2 muestra la implementación a nivel transistor de éste bloque

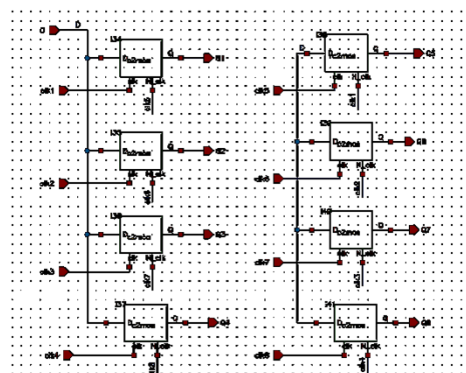


Figura 2. Circuito de decisión

Circuito de detección de fase, eliminación de glitches

La topología utilizada en el CDR es tipo XOR, éste bloque detecta las variaciones de fase entre el reloj de salida del VCO y la velocidad de entrada del dato (figura 3).

Las salidas del módulo sirven como señales de entrada a un retenedor para ser eliminados los glitches que se generan en la compuerta, el circuito encargado de realizar esta eliminación de glitches y retención de datos esta compuesto por 8 flip flop tipo D usando la topología C<sup>2</sup>MOS (figura 4).

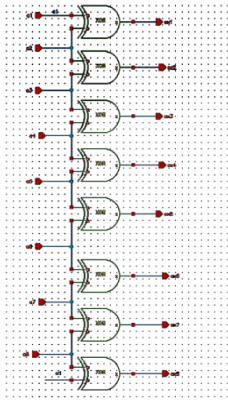


Figura 3. Circuito de detección de fase

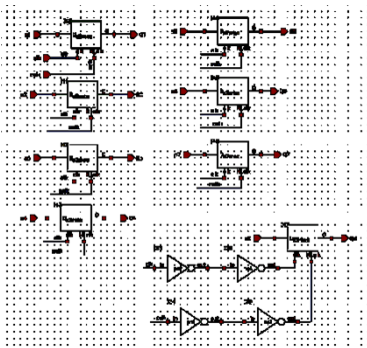


Figura 4. Retención de datos y eliminación de glitches

### Sincronía entre reloj y dato

Una compuerta OR de 8 entradas 1 salida es la encargada de detectar en el ciclo de reloj correspondiente un cambio en los datos de entrada. Es decir, indica si por ejemplo en la señal del clk1 se detecta algún cambio en el dato de entrada que junto con su reloj asignado activa los flip flop de memoria y que su objetivo es guardar el valor de la entrada anterior (figura 5).

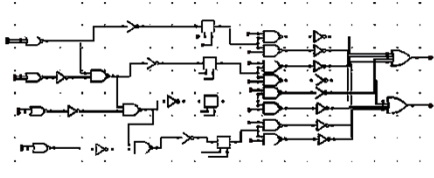


Figura 5. Circuito de sincronización

### Circuito de prioridad

El objetivo de este bloque es dar prioridad al dato que llegue primero a sus terminales de entrada e ignorar los valores de las siguientes entradas, es decir si en la entrada de dato que corresponde al clk1 hay un cambio de nivel el circuito deshabilita todos los demás valores y ya no seguir leyendo el dato de las demás entradas, esto lo hace para los siguientes 7 datos de entrada (figura 6).

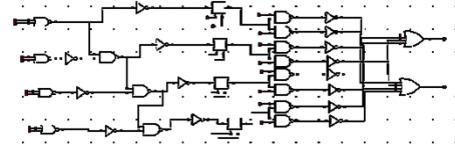


Figura 6. Circuito de prioridad

A la entrada del circuito de prioridad se tiene un bloque el cual almacena la señal de reloj que corresponde para cada señal de entrada. Trabaja en sincronía con los relojes de salida clk1 y clk5 para que junto con el bloque de mayor precedencia sincronicen el dato recuperado con su señal de reloj.

Finalmente, las salidas de este bloque van a un módulo que sincroniza las señales de reloj generadas con el dato recuperado correspondiente (figura 7).

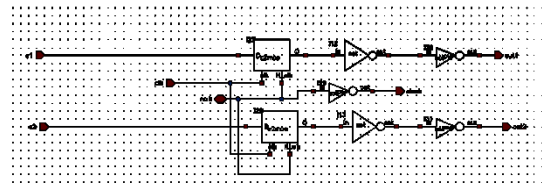


Figura 7. Sincronización datos - reloj

## III. RESULTADOS EXPERIMENTALES

El esquema VCO – CDR se muestra en la figura 8. De igual forma se muestran en la figura 9 los bloques internos de los que se compone el circuito recuperador de datos.

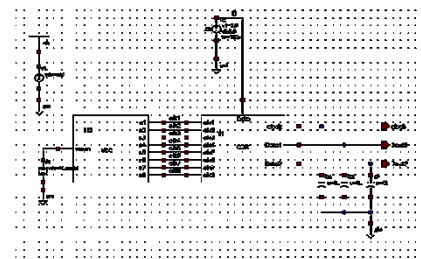


Figura 8. Implementación del CDR con VCO

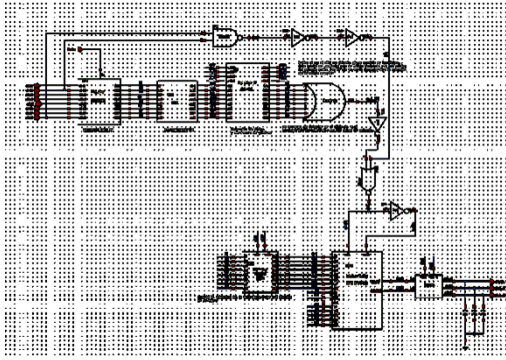


Figura 9. Esquema del CDR

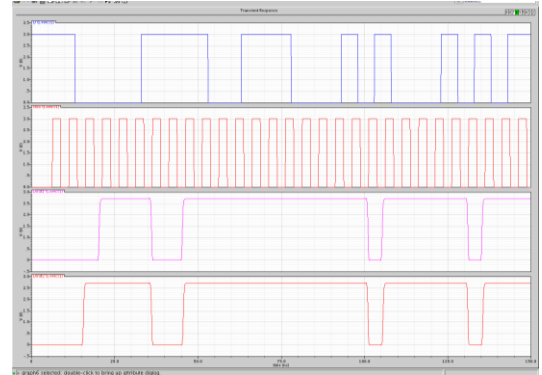


Figura 12. Salida del dato para el peor de los casos

Los resultados de simulación, tanto de las pruebas PVT como de las señales de salida se muestran en las siguientes figuras.

En este proyecto solo se realizó el layout del circuito de prioridad, el cual se muestra en la siguiente imagen

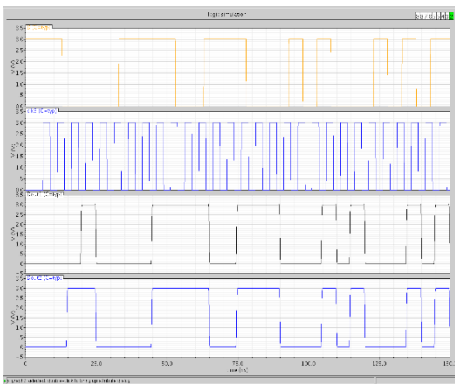


Figura 10. Resultados del CDR para un caso típico

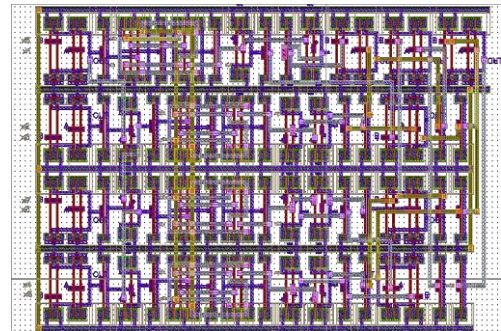


Figura 13. Layout del circuito de prioridad

Para las pruebas en el mejor caso a 3.3 V, -40°C con una frecuencia de 200 MHz se obtuvieron los resultados que se muestran en la figura 11.

Los resultados que arroja el circuito de prioridad prelayout son los que se ven en la figura 14 mientras que los resultados postlayout se muestran en la figura 15.

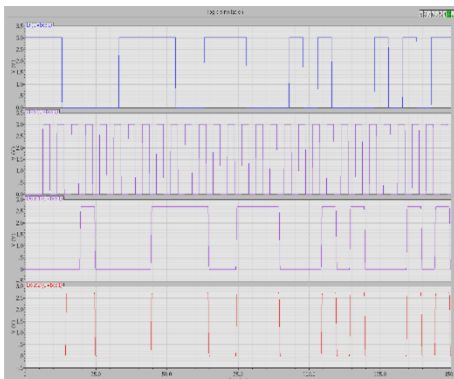


Figura 11. Salida del dato para el mejor de los casos

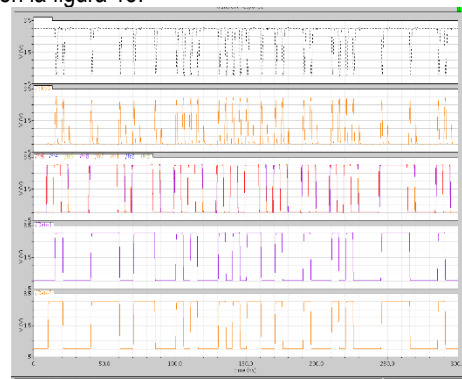


Figura 14. Análisis prelayout del circuito de prioridad

Para las pruebas en el peor caso a 2.7 V, 125°C con una frecuencia de 200 MHz se obtuvieron los resultados que se muestran en la figura 12.

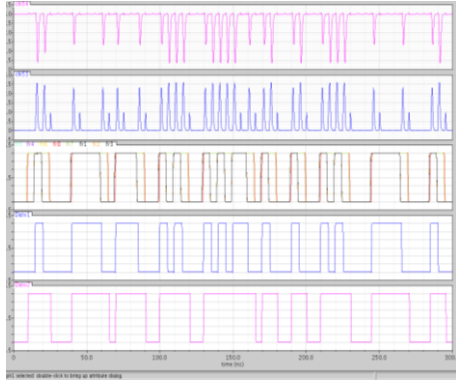


Figura 14. Análisis postlayout del circuito de prioridad

#### IV. CONCLUSIONES

Como se mencionó anteriormente, un CRD es de bastante utilidad sobre todo en el área de comunicaciones a alta velocidad ya que en nuestra vida diaria es difícil encontrar señales que se hayan enviado sin algún factor de ruido inherente.

El esquema que se ve en la figura 9 se llevó a cabo en verilogA y los resultados fueron mejores, esto en base a que no se presentaron fenómenos como glitches, jitter o skew. El uso de un código en verilogA para proporcionar las señales de reloj provenientes del VCO ayudó a eliminar desde un inicio los fenómenos anteriormente mencionados.

#### V. REFERENCIAS Y BIBLIOGRAFÍA

- [1] B. Razavi, *Design of Analog CMOS Integrated Circuits*, New York: McGraw Hill, 2001.
- [2] Philipus Oh, Win Chaivipas, Akira Matsuzawa. "A Study on Full Digital Clock Data Recovery (CDR)." Tokyo Institute of Technology, Tokyo, Japan.  
[http://www.ssc.pe.titech.ac.jp/publications/2006/20060526\\_philipus\\_ppt.pdf](http://www.ssc.pe.titech.ac.jp/publications/2006/20060526_philipus_ppt.pdf).
- [3] "Clock and Data Recovery"  
[http://analog.postech.ac.kr/3.Class/1.Classes/06\\_695v/CDR.pdf](http://analog.postech.ac.kr/3.Class/1.Classes/06_695v/CDR.pdf).
- [4] Razavi B. "Challenges in the design high-speed clock and data recovery circuits." *Communications Magazine*, IEEE, vol. 40, pp. 94-101, Aug 2002.
- [5] Alexander, J. D. H., "Clock Recovery from Random Binary Signals", *Electronics Letters* 11, 22 (30th October 1975), 541-542.
- [6] "Lecture 200 – CLOCK AND DATA RECOVERY CIRCUITS" [http://users.ece.gatech.edu/~jskenney/L200-CDR-I\(2UP\).pdf](http://users.ece.gatech.edu/~jskenney/L200-CDR-I(2UP).pdf).

## **APENDICE B**

*DETECCION DE IMAGENES MANIPULADAS MEDIANTE EL  
ANALISIS BIESPECTRAL*

## **Detección de imágenes manipuladas mediante el análisis biespectral**

Este proyecto consiste en la generación de un algoritmo para la detección de imágenes manipuladas digitalmente. La detección se hace usando el análisis biespectral. Este proyecto se realizó en equipo junto con Juan Carlos Romero Ramirez.

La manipulación de imágenes puede ser utilizada para promover un artículo en una revista, para la difamación de un político adversario, o para otros objetivos. Las herramientas forenses para el análisis de imágenes digitales son utilizadas para ayudar a resolver crímenes. El desarrollo de estas herramientas es esencial para el futuro del análisis forense de imágenes manipuladas, como la mostrada en la **Error! Reference source not found..**

Una técnica utilizada para determinar la autenticidad de una imagen digital es conocida como “firma” o “marca de agua”, ésta consiste en agregar una codificación a la imagen (la cual no es visible a simple vista), y que es posible decodificarla posteriormente. Sin embargo, esta técnica presenta desventajas porque la mayoría de las cámaras digitales no agregan la marca de agua, incluso esta técnica tampoco funcionaría para aquellas imágenes provenientes de una digitalización.

Existen otras técnicas para la detección de imágenes manipuladas, una de ellas consiste en detectar inconsistencias entre bloques generados durante la compresión de una imagen JPEG. Sin embargo, esta técnica también tiene limitantes, si la imagen se vuelve a codificar después de su manipulación, el algoritmo de detección se vuelve vulnerable [3]



Figura 15 Fotomontaje de Saddam Hussein con Bill Clinton en la Casa Blanca. (Tomado de Fridrich, Soukal, & Lukás, 2003)

Para este proyecto se decidió utilizar el análisis biespectral, pues al no depender de una codificación o de una marca de agua, sino únicamente de la información de la señal misma, esta técnica se consideró como la más robusta. El análisis biespectral es una técnica utilizada para el análisis de no linealidades causadas por un sistema no natural.

Cuando una señal pasa a través de un sistema no lineal, se crean armónicos a frecuencias y fases correlacionadas (las cuales no son creadas por sistemas lineales). El objetivo del análisis biespectral es medir y cuantificar dichas correlaciones de alto orden de frecuencia y fase. Para el análisis de señales en el dominio de frecuencia normalmente se recurre al “espectro de potencia”. Sin embargo, esta técnica únicamente es capaz de detectar la presencia de correlaciones de hasta

segundo orden y no de orden mayor, las cuales son de interés para nuestro estudio. Para solucionar este problema se recurre al biespectro, el cual es capaz de detectar correlaciones de tercer orden.

Una vez obtenido el biespectro se procede a obtener la bicoherencia, el cual es la versión normalizada del biespectro. Cuando una no linealidad es detectada, el espectro de potencia no varía, pero la magnitud e histograma de fase de la bicoherencia sí.

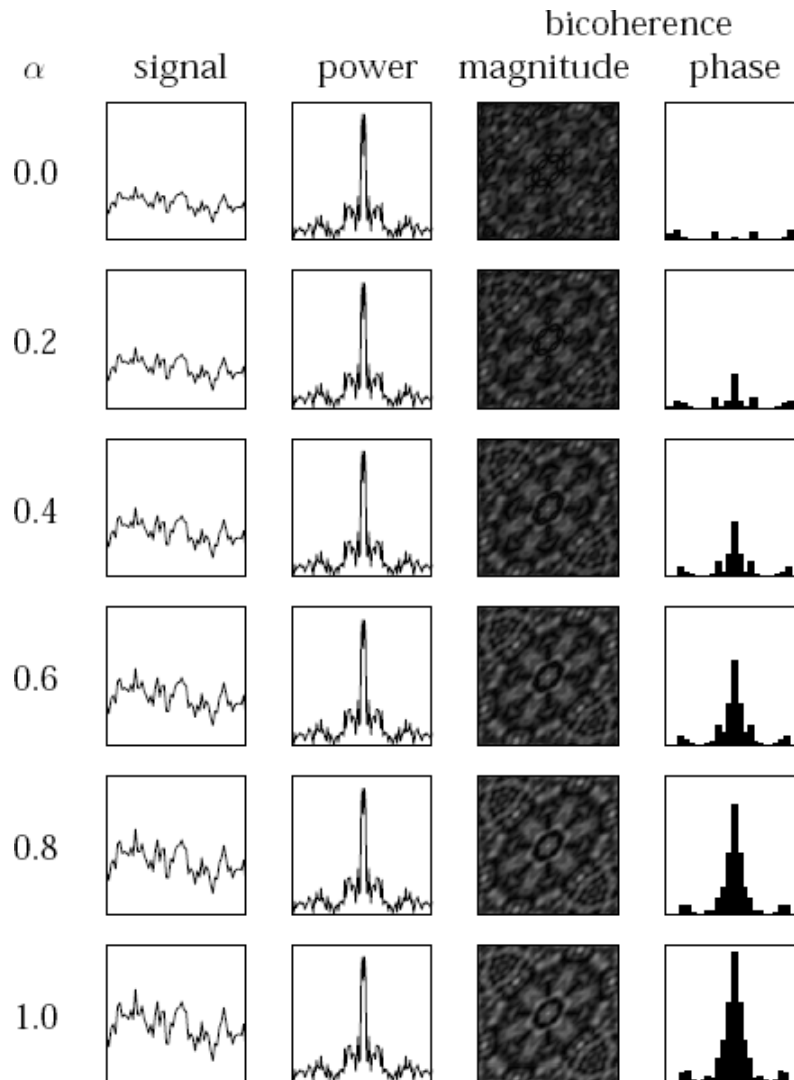


Figura 16 Comparación entre espectro de potencia y magnitud y fase de la bicoherencia de una señal  $x(n)$  sometida a un sistema no lineal  $y(n)=ax^2(n)+x(n)$ . (Tomado de Farid, 1999)

Una de las dificultades de este proyecto consistió en que las imágenes son señales bidimensionales y el cálculo de la bicoherencia de la imagen completa requiere de mayor

capacidad de procesamiento. Para ello se probó un nuevo algoritmo; éste consistió en dividir la imagen en múltiples vectores horizontales y verticales, posteriormente se calculó la bicoherencia de cada uno de los vectores, finalmente se promediaron los valores de la magnitud de la bicoherencia.

Este proyecto se implementó con la ayuda de la herramienta de simulación matemática Matlab, además fue necesaria la instalación de una librería llamada HOSA (*High-Order Spectral Analysis*) para la obtención de la bicoherencia.

## Resultados

Para probar nuestro algoritmo de detección de imágenes, se crearon dos imágenes de prueba. Una consistió en dividir una imagen en cuatro secciones e invertir dos de ellas de manera diagonal; la segunda consistió en sustituir un pequeño segmento de la imagen con una copia de otra sección de la imagen.



Figura 17 Imágenes de prueba: Lena cruzado, y Lena desplazado

Posteriormente se realizaron los cálculos de la bicoherencia mediante dos métodos distintos: el método 1 consistió en aplicar el cálculo de la magnitud de la bicoherencia directamente al archivo de imagen y obtener su valor máximo; el método 2 consistió en segmentar la imagen en pequeños vectores horizontales y verticales, para después calcular el promedio de la magnitud de la bicoherencia máxima de cada segmento. En la **Error! Reference source not found.** se muestra una comparación entre los resultados de las mediciones de la magnitud de la bicoherencia por ambos métodos. Se puede observar que a la hora de cruzar los segmentos de imagen provocaron un aumento significativo en la magnitud de la bicoherencia; mientras que en el caso del desplazamiento de una pequeña sección de la imagen, ocasiona también un pequeño incremento en la magnitud de la bicoherencia.

<b>Imagen</b>	<b>Método 1</b>	<b>Método 2</b>
	<b>Valor máximo de la bicoherencia de la imagen completa</b>	<b>Promedio de los valores máximos de la bicoherencia de cada segmento de la imagen</b>
Lena original	0.4430	0.7811
Lena cruzado	0.8029	1.0315
Lena desplazado	0.4434	1.0854

*Tabla 7 Resultados de la magnitud de la bicoherencia*

Pese a que se pudo comprobar la teoría con la práctica, aun se cuestiona la eficiencia de este método si consideramos que la magnitud de la bicoherencia de la imagen original es desconocida, por lo tanto, una pequeña alteración de la imagen ocasionaría un mínimo cambio en la magnitud de la bicoherencia.

Los principales resultados de este trabajo fueron los siguientes:

1. La idea de utilizar el análisis biespectral surgió de un artículo que explicaba cómo se podía usar tal algoritmo para la detección de manipulaciones en un archivo de audio. Nosotros partimos del mismo principio y adaptamos este algoritmo a imágenes digitales.
2. El proceso de investigación que se realizó permitió redactar un artículo con formato IEEE, donde se pudo mencionar toda la investigación que este trabajo requirió ya que se partió audio a imágenes. Dicho artículo se presenta enseguida

# Image forgeries detection using Biespectral Analysis

Juan Carlos Romero Ramírez  
Instituto Tecnológico y de Estudios Superiores de Occidente  
Tlaquepaque, Jalisco, México  
[lagart@gmail.com](mailto:lagart@gmail.com)

Ivonne Alfaro Flores  
Instituto Tecnológico y de Estudios Superiores de Occidente  
Tlaquepaque, Jalisco, México  
[ivalfl@yahoo.com.mx](mailto:ivalfl@yahoo.com.mx)

**Abstract**—The advent of the modern digital technology has not only brought about the prevalent use of digital images in our daily activities but also the ease of creating image forgery such as digital photomontages using publicly accessible and user friendly image processing tools such as Adobe Photoshop. Among all operations involved in image photomontage, image splicing can be considered the most fundamental and essential operation. In this paper we address the problem of authenticating digital signals assuming no explicit prior knowledge of the original. The basic approach that we take is to assume that in the frequency domain a “natural” signal has weak higher-order statistical correlations. We then show that “un-natural” correlations are introduced if this signal is passed through a non-linearity (which would almost surely occur in the creation of a forgery). Techniques from polyspectral analysis are then used to detect the presence of these correlations.

**Keywords**;bispectrum, bicoherence

## I. INTRODUCTION

Photomontage refers to a paste-up produced by sticking together photographic images. Actually, creating photomontage can be as easy as performing a cut and paste with specific tools provided by image publishing software such as Adobe Photoshop. Nowadays, this type of software is widely accessible to us. The ease of creating digital photomontage, with a quality that could manipulate belief, would certainly make us to think twice before accepting an image as authentic. This becomes a serious issue when it comes to photographic evidence presented in the court or for insurance claims. Image authenticity is also a critical for news photographs as well as for the scanned image of checks in the electronics check clearing system.

With the rapid increase in low-cost and sophisticated digital technology the need for techniques to authenticate digital material will become more urgent. In this paper we address the problem of determining whether a digital signal has been tampered with from the time of its recording. As illustrated in Figure 1 our work differs from recent work in digital watermarking where the goal is to embed in a signal an imperceptible and distinctive pattern that is impervious to a broad class of manipulations; the watermark is stored and later used for authentication. Such techniques have primarily been developed for protecting copyrights. Our work is also distinct from the vast authentication literature

which encodes and encrypts certain aspects of what is assumed to be the authentic signal; this signature is later used for authentication. These techniques have been popular in trying to ensure that signals are not tampered with during transmission over insecure lines.

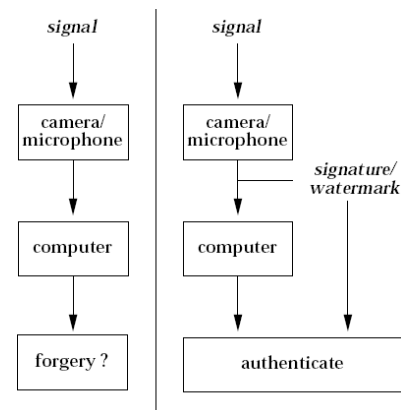


Figure 1. Unlike other authentication techniques (right) we are interested in verifying the authenticity of digital signals (left) with no explicit prior knowledge of the original

Unlike both of these approaches we assume no explicit prior knowledge of the original signal. The basic approach that we take to detecting digital forgeries is to assume that in the frequency domain a “natural” signal has weak higher-order statistical correlations. We then show that if this signal is passed through a non-linearity (which would almost surely occur in the creation of a forgery), that “un-natural” higher-order correlations are introduced (in magnitude and phase). Techniques from polyspectral analysis [4] are then used to detect the presence of these correlations, and used as an indication of digital tampering.

## II. PRIOR WORK

One way to examine the authenticity of images is to check the internal consistencies within a single image, such as whether or not all the objects in an image are in correct perspective or the shadow is consistent with the lighting [2]. This technique inspects the minor detail of the image to locate the possible inconsistencies, which is likely to be overlooked by forgers.

Another approach of asserting image authenticity is through extracting digital signature [3] or content-based signature from an image at the moment it is taken by a secure or trustworthy camera. Alternatively, watermark data can be embedded into an image to achieve the same purpose. In general, watermarking techniques such as fragile watermarking [4-8], semi-fragile watermarking [9-12] or content-based watermarking [13,14] are used for the image authentication application. Fragile watermark is impractical for many real-life applications as compression and transcoding of images are very common in the chain of multimedia delivery. In this case, fragile watermarking techniques will declare a transcode or compressed image as inauthentic even though the transcoding or compression operation is a content preserving.

As the watermarking and the signature extraction technique for content authentication have to work together with a secure camera the security of these approaches depends on the security of the camera as well as the security of watermarking or the signature extraction. A secure camera has to ensure that a watermarking is only embedded or a signature is only extracted at the every moment an image is captured.

Unfortunately, until today, there is still not a fully secure watermarking scheme; the watermarking secret for ensuring the security of a watermarking scheme can be hacked given sufficient conditions such as when sufficient number of images with the same secret watermark key are available or the embedded watermark can be removed by exploiting the weak points of a watermarking scheme.

### III. BISPECTRAL ANALYSIS

For purpose of illustration consider a signal composed of a sum of two sinusoids:

$$x(n) = \sin(\omega_1 n + \phi_1) + \sin(\omega_2 n + \phi_2) \quad (1)$$

Consider now passing this signal through a simple non linearity

$$y(n) = x^2(n) + x(n) \quad (2)$$

Expanding the polynomial and rewriting in terms of the harmonics using basic trigonometric identities gives:

$$\begin{aligned} y(n) = & \frac{1}{2} \left( 1 + \sin(2\omega_1 n + 2\phi_1 - \frac{\pi}{2}) \right) \\ & + \frac{1}{2} \left( 1 + \sin(2\omega_2 n + 2\phi_2 - \frac{\pi}{2}) \right) \\ & + 2 \sin((\omega_1 + \omega_2) + (\phi_1 + \phi_2)) \\ & + 2 \sin((\omega_1 - \omega_2) + (\phi_1 - \phi_2)) \\ & + \sin(\omega_1 n + \phi_1) + \sin(\omega_2 n + \phi_2) \end{aligned} \quad (3)$$

The power spectrum is blind to higher order correlations of the sort of interest to us. However these correlations can be detected by turning to higher-order spectral analysis. For example the

bispectrum is used to detect the presence of third order correlations

$$B(\omega_1, \omega_2) = Y(\omega_1)Y(\omega_2)Y^*(\omega_1 + \omega_2) \quad (4)$$

From an interpretative stance it will be convenient to express the complex bispectrum with respect to its magnitude:

$$|B(\omega_1, \omega_2)| = |Y(\omega_1)| \cdot |Y(\omega_2)| \cdot |Y(\omega_1 + \omega_2)| \quad (5)$$

And phase

$$\angle B(\omega_1, \omega_2) = \angle Y(\omega_1) + \angle Y(\omega_2) - \angle Y(\omega_1 + \omega_2) \quad (6)$$

Also from an interpretative stance it is helpful to work with the normalizaed bispectrum, the bicoherence:

$$B_c(\omega_1, \omega_2) = \frac{Y(\omega_1)Y(\omega_2)Y^*(\omega_1 + \omega_2)}{\sqrt{|Y(\omega_1)Y(\omega_2)|^2 |Y(\omega_1 + \omega_2)|^2}} \quad (7)$$

### IV. ALGORITHM FOR THE BICOHERENCE

1. The data y are segmented in records without overlapping.
2. The mean is remained for each record.
3. A time-domain Hanning window is applied.
4. FFT Calculated.
5. The power spectrum of the kth record is computed as

$$P_K(m) = |X_K(m)|^2 \quad (8)$$

where :

$X_K(m)$  denotes the FFT of the kth record.

6. The bispectrum is computed as

$$B_K(m, n) = X_K(m)X_K(n)X_K^*(m+n) \quad (9)$$

7. And the bicoherence is then estimated as

$$bic(f_1, f_2) = \frac{|B(f_1, f_2)|^2}{P(f_1)P(f_2)P(f_1 - f_2)} \quad (10)$$

#### A. Method 1 for forgeries detection

1. Data y is an image matrix.
2. The magnitude of bicoherence is estimated without overlapping, since the bicoherence algorithm does not allow overlapping when its input is a matrix.
3. The maximum magnitude and mean magnitude of bicoherence are listed.

#### B. Method 2 for forgeries detection

1. The image matrix is decomposed in one set of columns and other for rows. Thus we could compute vectors with overlapping.
2. From each vector (columns and rows) is estimated its bicoherence.
3. The final bicoherence is then estimated as

$$bic(f_1, f_2) = \sqrt{\left(\frac{1}{N_h} \sum_k M_i^{Horizontal}\right)^2 + \left(\frac{1}{N_v} \sum_k M_i^{Vertical}\right)^2} \quad (11)$$

4. The maximum magnitude and mean magnitude of bicoherence are listed.

## V. RESULTS

We use a couple of probe images: Lena and Boat. To both, spliced forgeries were applied. One of them consisted in swap to quarters of the image, and the other consisted in splice a small portion of the image in other point. Below it is showed the probe images.



Figure 2. Test for forgery

After applied both methods, we obtained the graphics of bicoherence magnitude, showed in fig.3. and fig.4. This kind of graphics are called fractals and represents the frequencies where exist a high order correlation.

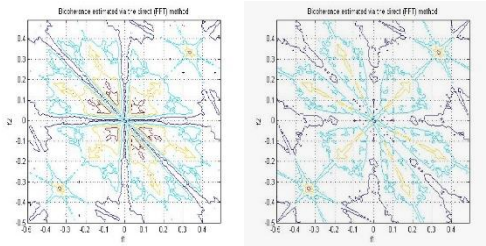


Figure 3. Bicoherence of Lena Image; (a) from original image, (b) from cross forged image

The Table I shows comparative results of the average and maximum bicoherence magnitude of each image.

TABLE I. TABLE OF RESULTS

Image	Method 1		Method 2	
	Maximum	Average	Maximum	Average
Lena original	0.44297	0.0155	1.9374	0.7811
Lena crossed	0.80278	0.0114	2.6215	1.0315
Lena spliced	0.44342	0.0155	1.9218	1.0854
Boat original	1.72960	0.0087	1.3357	0.5909
Boat crossed	0.43373	0.0249	1.5664	0.6748
Boat spliced	1.75680	0.0087	1.3409	0.6038

We could observe an increment of 39% in bicoherence average of Lena spliced image, and 2% in Boat, when method 2 is applied.

## VI. REFERENCES

- [1] Hanneschlänger Günther, "HOSA Higher Order Spectra Analysis", Nov. 24, 2004 pp 1-9
- [2] W. J. Mitchell, "When is seeing believing?", in Scientific American, 1994, pp. 44-49
- [3] G. L. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image", IEEE Transactions on Consumer Electronics, vol. 39, pp. 905-910, Nov. 1993.
- [4] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification", IEEE International Conference on Image Processing, Oct 26-29, 1997
- [5] M. Wu and B. Liu, "Watermarking for image authentication", IEEE International Conference on Image Processing, Oct 4-7, 1998
- [6] I. J. Cox, M. L. Miller and J. A. Bloom, Digital Watermarking: Morgan Kaufmann, 2002
- [7] J. Fridrich, M. Goljan, and B. A. C., "New Fragile Authentication Watermark for Images", IEEE International Conference on Image Processing, Vancouver, Canada, Sept 10-13, 2000
- [8] P. W. Wog, "A watermark for Image Integrity and Ownership Verification", IS&T Conference on Image Processing, Image Quality and Image Capture Systems, Portland, Oregon, May, 1998.
- [9] E. T. Lin, C. I. Podilchuk, and E. J. Delp, "Detection of Image Alterations Using Semi-fragile watermarks", SPIE International Conference on Security and watermarking of multimedia contents II, San José, CA, Jan 23-28, 2000
- [10] C.-Y. Lin and S. F. Chang "A Robust Image Authentication Method Surviving JPEG Lossy Compression", SPIE Storage and Retrieval of Image/Video Database, San José, Jan, 1998.
- [11] J. Fridrich, "Image Watermarking for Tamper Detection", IEEE International Conference on Image Processing, Chicago, Oct., 1998
- [12] N. Memon and P. Vora, "authentication Techniques for Multimedia Content", SPIE Multimedia Systems and Applications, Boston, MA. Oct, 1998
- [13] J. Fridrich and M. Goljan, "Protection of Digital Images Using Self Embedding" Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, May 14, 1999
- [14] C. Rey and J. L. Dugelay, "Blind Detection of Malicious Alterations on Still Images using Robust Watermarks", IEEE Seminar Secure Images and Image Authentication, Apr, 2000

## Algoritmo para el cálculo de la bicoherencia (Matlab)

```
function [bic,waxis] = bicoher (y, nfft, wind, nsamp, overlap)
%BICOHER - Direct (FD) method for estimating bicoherence
% [bic,waxis] = bicoher (y, nfft, wind, segsamp, overlap)
% y - data vector or time-series
% nfft - fft length [default = power of two > segsamp]
% actual size used is power of two greater than 'nsamp'
% wind - specifies the time-domain window to be applied to each
% data segment; should be of length 'segsamp' (see below);
% otherwise, the default Hanning window is used.
% segsamp - samples per segment [default: such that we have 8 segments]
% - if x is a matrix, segsamp is set to the number of rows
% overlap - percentage overlap, allowed range [0,99]. [default = 50];
% - if x is a matrix, overlap is set to 0.
% bic - estimated bicoherence: an nfft x nfft array, with origin
% at the center, and axes pointing down and to the right.
% waxis - vector of frequencies associated with the rows and columns
% of bic; sampling frequency is assumed to be 1.

% Copyright (c) 1991-2001 by United Signals & Systems, Inc.
% $Revision: 1.7 $
% A. Swami January 20, 1995

% RESTRICTED RIGHTS LEGEND
% Use, duplication, or disclosure by the Government is subject to
% restrictions as set forth in subparagraph (c) (1) (ii) of the
% Rights in Technical Data and Computer Software clause of DFARS
% 252.227-7013.
% Manufacturer: United Signals & Systems, Inc., P.O. Box 2374,
% Culver City, California 90231.
%
% This material may be reproduced by or for the U.S. Government pursuant
% to the copyright license under the clause at DFARS 252.227-7013.

% ----- parameter checks -----

[ly, nrecs] = size(y);
if (ly == 1) % Si datos son un vector
renglon
y = y(:); % lo convierte a columna
ly = nrecs;
nrecs = 1;
end

if (exist('nfft') ~= 1) % Tamaño default de FFT
nfft = 128;
end

if (exist('overlap') ~= 1) % Valor default de translape
overlap = 50;
end
overlap = max(0,min(overlap,99)); % Delimita rango de translape (0 a 99)

if (nrecs > 1)
```

```

        overlap = 0; % Translape cero si datos es
matriz
    end

    if (exist('nsamp') ~= 1)
        nsamp = 0; % Si no se especifica el numero de
muestras
    end

    if (nrecs > 1) % Si datos es matriz, numero de
muestras % igual a numero de filas
        nsamp = ly;
    end

    if (nrecs == 1 & nsamp <= 0) % Si datos son un vector y no se
especifica numero de muestras % divide los datos en 8
        nsamp = fix(ly/ (8 - 7 * overlap/100));
segmentos
    end

    if (nfft < nsamp) % Si el tamaño de la FFT es menor
al numero de muestras % tamaño FFT aumenta a la
        nfft = 2^nextpow2(nsamp);
potencia de 2 inmediata superior
    end

    overlap = fix( nsamp * overlap/100); % Distribuye uniformemente el
translape en todos los segmentos
    nadvance = nsamp - overlap;
nrecs = fix ( (ly*nrecs - overlap) / nadvance); % Una columna para cada
segmento

% -----
    if (exist('wind') ~= 1)
        wind = hanning(nsamp); % Ventana por default: Hanning
    end

    [rw,cw] = size(wind);
    if (min(rw,cw) ~= 1 | max(rw,cw) ~= nsamp) % Si la ventana no es del mismo tamaño
que el segmento
        disp(['Segment size is ',int2str(nsamp)])
        disp(['"wind" array is ',int2str(rw),' by ',int2str(cw)]) disp(['Using
        default Hanning window'])
        wind = hanning(nsamp); % se utiliza la ventana por
default
    end
    wind = wind(:); % Ventana es una columna
% ----- accumulate triple products -----

    bic = zeros(nfft,nfft); Pyy
        = zeros(nfft,1);

    mask = hankel([1:nfft],[nfft,1:nfft-1] ); % the hankel mask (faster) (matriz
con diagonales paralelas)
    Yf12 = zeros(nfft,nfft);
    ind = [1:nsamp]; % indice = primer segmento

```

```

for k = 1:nrecs
    y(ind);
    ys = (ys(:) - mean(ys)) .* wind;
segmento y se multiplica

    Yf = fft(ys,nfft) / nsamp;
    conj(Yf);
    Pyy = Pyy + Yf .* CYf;
potencia
    Yf12(:) = CYf(mask);
coeficientes del conjugado
    bic = bic + (Yf * Yf.') .* Yf12;
    ind = ind + nadvance;
end

    bic = bic / nrecs;
    Pyy = Pyy / nrecs;
potencia
    mask(:) = Pyy(mask);
coeficientes del especto de potencia
    bic = abs(bic).^2 ./ (Pyy * Pyy.' .* mask);
    fftshift(bic);

% ----- coutout plot of magnitude bispectrum -----

if (rem(nfft,2) == 0)
    waxis = [-nfft/2:(nfft/2-1)]/nfft;
else
    waxis = [-(nfft-1)/2:(nfft-1)/2]/nfft;
end

hold off, clf
% contour(bic,4,waxis,waxis),grid
contour(waxis,waxis,bic,4), grid on
title('Bicoherence estimated via the direct (FFT) method') xlabel('f1'),
ylabel('f2')
set(gcf,'Name','Hosa BICOHER')

[colmax,row] = max(bic) ;
[maxval,col] = max(colmax); row =
row(col);
disp(['Max: bic(',num2str(waxis(row)),',',num2str(waxis(col)),') = ', ... num2str(maxval) ])

return

```

```

% Repite para cada segmento ys =
% Valores del primer segmento
% A cada valor se le resta la media del
% por la ventana (Hanning)
% Se aplica la FFT CYf =
% Conjugado de la FFT
% Sumatoria de los Espectros de
% Crea mascara de hankel con los
% Sumatoria de los Biespectros
% Indice = siguiente segmento
% Promedio de los biespectros
% Promedio de los Espectros de
% Mascara hankel con
% Estimación de la bicoherencia bic =

```

```

% Frecuencias normalizadas

```

```

% Grafica de fractales.

```

## **Prueba 1. Cálculo de la bicoherencia de toda la imagen (Matlab)**

```

% En este archivo se calcula el bicoherencia de toda la imagen y se calcula
% la magnitud de la bicoherencia promedio.

```

```

clc
clear all
close all

```

```

x=double(imread('lena_original.png','png'));

```

```

% Asigna imagen a una matriz

```

```

y=double(imread('boat_original.png','png'));

x_f=double(imread('lena_forger.png','png'));
y_f=double(imread('boat_forger.png','png'));

x_c = x; % Intercambia cuadrante (1:1) por (2:2)
x_c(1:256,1:256) = x(257:512,257:512);
x_c(257:512,257:512) = x(1:256,1:256);

y_c = y;
y_c(1:256,1:256) = y(257:512,257:512);
y_c(257:512,257:512) = y(1:256,1:256);

[bic_x, waxis_x] = bicoher(y); % Bicoherencia de Lena
figure
[bic_x_c, waxis_x_c] = bicoher(y_c); % Bicoherencia de Lena cruzado figure
[bic_x_f, waxis_x_f] = bicoher(y_f); % Bicoherencia de Lena forger

average=0;
for i = 1:26144
    average=average+bic_x(i);
end
average_x = average/26144 % Magnitud promedio de bicoherencia Lena

average=0;
for i = 1:26144
    average=average+bic_x_c(i);
end
average_x_c = average/26144 % Magnitud promedio de bicoherencia Lena
cruzado

average=0;
for i = 1:26144
    average=average+bic_x_f(i);
end
average_x_f = average/26144 % Magnitud promedio de bicoherencia Lena
forger

```

## **Prueba 2. Cálculo de la bicoherencia por filas y columnas (Matlab)**

```

%Este archivo calcula la bicoherencia de las filas y de las columnas
%de la imagen, luego las promedia, y estima la magnitud de la bicoherencia,
%finalmente calcula el promedio de la magnitud de la bicoherencia.

clc
clear all
close all

x=double(imread('lena_original.png','png')); % Asigna imagen a una matriz
y=double(imread('boat_original.png','png')); z=double(imread('shine_original.png','png'));

```

```

x_f=double(imread('lena_forger.png','png'));
y_f=double(imread('boat_forger.png','png'));

x_c = x;
x_c(1:256,1:256) = x(257:512,257:512);
x_c(257:512,257:512) = x(1:256,1:256);

y_c = y;
y_c(1:256,1:256) = y(257:512,257:512);
y_c(257:512,257:512) = y(1:256,1:256);

z_c = z;
z_c(1:256,1:256) = z(257:512,257:512);
z_c(257:512,257:512) = z(1:256,1:256);

matriz = x; % Seleccionar una matriz de imagen
% NOTA: LA PRUEBA TARDA VARIOS
% SEGUNDOS EN CORRER

%Lena Original

bic_hor = 0;
bic_ver = 0; for i
= 1:512
    [bic, waxis] = bicoher2(matriz(i,:)); % Bicoherencia horizontal bic_hor
    = bic_hor + bic; % Bic. horizontal acumulada
end

bic_hor = (bic_hor/512).^2;
bic_ver = (bic_ver/512).^2;
bic_matriz = sqrt(bic_hor + bic_ver); % Bicoherencia estimada
average=0;
for i = 1:16384
    average=average+bic_matriz(i);
end
average_matriz = average/16384 % Promedio de la bicoherencia estimada

```

## **APÉNDICE C**

*FILTRO CHEBYSHEV RECHAZA BANDA CON LÍNEAS DE TRANSMISIÓN*

## Filtro Chebyshev Rechaza Banda con líneas de transmisión

Este trabajo se hizo en conjunto con Lizbeth De La Mora Hernandez como parte del proyecto final para acreditar la materia de Diseño en Alta Frecuencia.

El objetivo del proyecto diseñar, analizar a través de resultados de simulaciones en ADS un filtro Chebyshev rechaza banda usando un resonador tipo-L con una frecuencia central de 3.4GHz, con un ancho de banda de 5% (170MHz), un rizo de 0.1dB y un sustrato RO4003 (h=0.081mm; cladding of 0.5oz,  $\epsilon_r = 3.55$ ) (www.rogerscorp.com, n.d.)

El filtro rechaza banda son usados rechazar una banda de frecuencias de una señal, como se muestra en la **Error! Reference source not found.** La banda de frecuencias a rechazar es la banda de rechazo entre  $\Omega_1$  y  $\Omega_2$ .

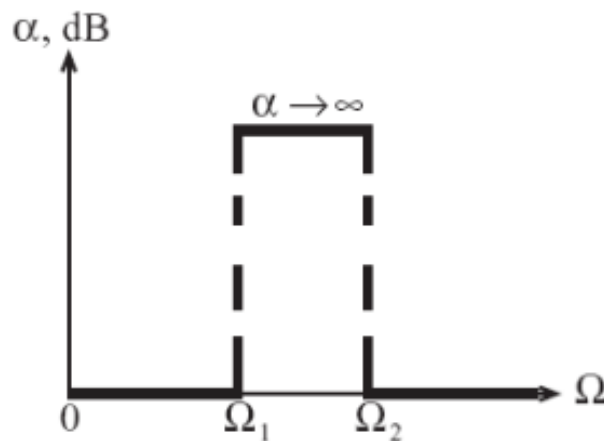


Figura 18 Respuesta de un filtro rechaza banda

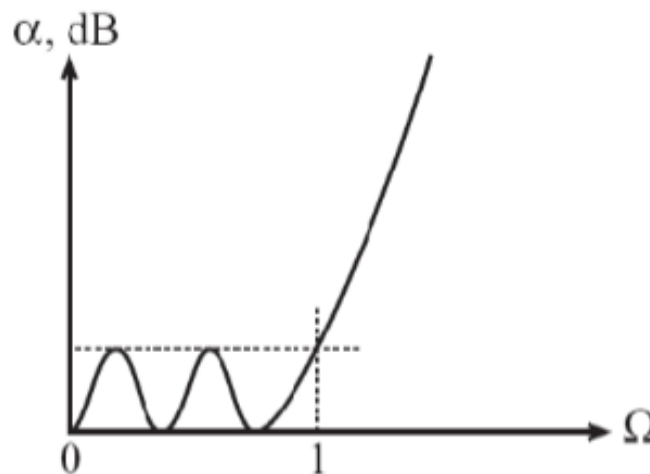


Figura 19 Respuesta filtro Chebyshev

Un filtro rechaza banda es una transformación de un filtro pasa bajas

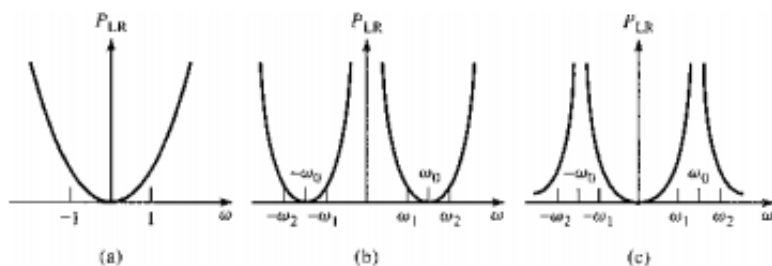


Figura 20 (a) Respuesta de un filtro pasa bajas para  $\omega_c=1$  (b) Filtro Pasa bandas (c) Filtro rechaza banda

El filtro se diseñará usando resonadores cuarto de onda

## Diseño del filtro

### Especificaciones

- Frecuencia central = 3.4GHz
- Ancho de banda = 5%
- Orden del filtro = 3
- Respuesta en frecuencia = 0.1dB Chebyshev
- Impedancia =  $50\Omega$
- Sustrato Roger RO4003 h=0.81mm, cladding of 0.5oz,  $\epsilon_r=3.55$

## Proceso de diseño

El flujo de diseño está basado en el material que se vio en la clase de diseño de altas frecuencias (Zabdiel, n.d.)

**Paso 1** – En base al orden del filtro deseado, es necesario encontrar los valores L-P (Chebyshev):

For passband ripple  $L_{dr} = 0.1$  dB

$n$	$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	$g_7$	$g_8$	$g_9$	$g_{10}$
1	0.3052	1.0								
2	0.8431	0.6220	1.3554							
3	1.0316	1.1474	1.0316	1.0						
4	1.1088	1.3062	1.7704	0.8181	1.3554					
5	1.1468	1.3712	1.9750	1.3712	1.1468	1.0				
6	1.1681	1.4040	2.0562	1.5171	1.9029	0.8618	1.3554			
7	1.1812	1.4228	2.0967	1.5734	2.0967	1.4228	1.1812	1.0		
8	1.1898	1.4346	2.1199	1.6010	2.1700	1.5641	1.9445	0.8778	1.3554	
9	1.1957	1.4426	2.1346	1.6167	2.2054	1.6167	2.1346	1.4426	1.1957	1.0

Tabla 8 Valor para un pasa bajas con un rizo de  $L_{AR}=0.1$ dB

Basados en (Pozar) se puede obtener la transformada a un filtro rechaza banda

$$\Omega = \frac{\Omega_c FBW}{\left(\frac{\omega_0}{\omega} - \frac{\omega}{\omega_0}\right)}$$

Ecuación 1

$$\omega_0 = \sqrt{\omega_1 \omega_2}$$

*Ecuación 2*

$$FBW = \frac{\omega_2 - \omega_1}{\omega_0}$$

*Ecuación 3*

Donde:

$\omega_2$  representa el ancho de banda deseado

Los elementos L-C se calculan en base a las siguientes ecuaciones:

$$C_{paralelo} = \left( \frac{1}{FBW \omega_0 \Omega_c} \right) \frac{1}{\gamma_0 g}$$

*Ecuación 4*

$$L_{paralelo} = \left( \frac{\Omega_c FBW}{\omega_0} \right) \gamma_0 g$$

*Ecuación 5*

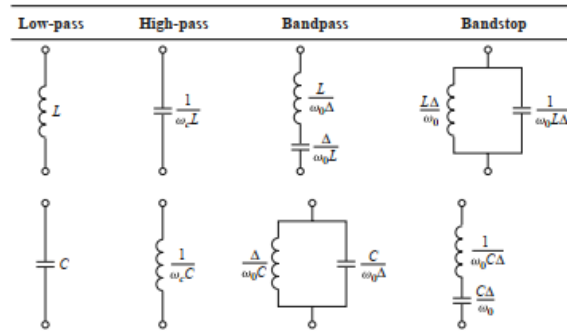
$$L_{serie} = \left( \frac{1}{FBW \omega_0 \Omega_c} \right) \frac{\gamma_0}{g}$$

*Ecuación 6*

$$C_{serie} = \left( \frac{\Omega_c FBW}{\omega_0} \right) \frac{g}{\gamma_0}$$

*Ecuación 7*

En componentes L-C, las ecuaciones anteriores se traducen de la siguiente manera:



*Tabla 9 Transformaciones entre filtros ( $\Delta = \frac{\omega_2 - \omega_1}{\omega_0}$ )*

## Paso 2 – Calcular los componentes L-C

La topología que se usa es Shunt (**Error! Reference source not found.**). Usando las **Error! Reference source not found.** hasta la **Error! Reference source not found.**, los elementos L-C serie y paralelo se muestran en la tabla

	L (H)	C (F)
1	4.5376e-08	4.8289e-14
2	1.3428e-10	1.6319e-11
3	4.5376e-08	4.8289e-14

Tabla 10 Elementos escalados del filtro rechaza banda

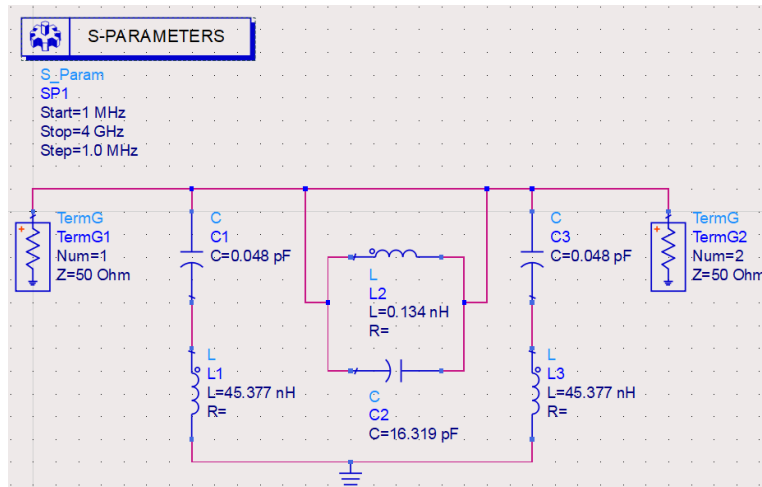


Figura 21 Esquemático de un filtro rechaza banda - Shunt

## Simulaciones ADS

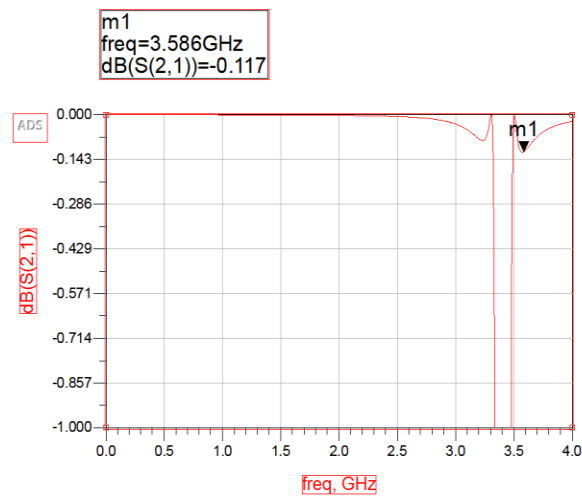


Figura 22 Simulación del rizo obtenido

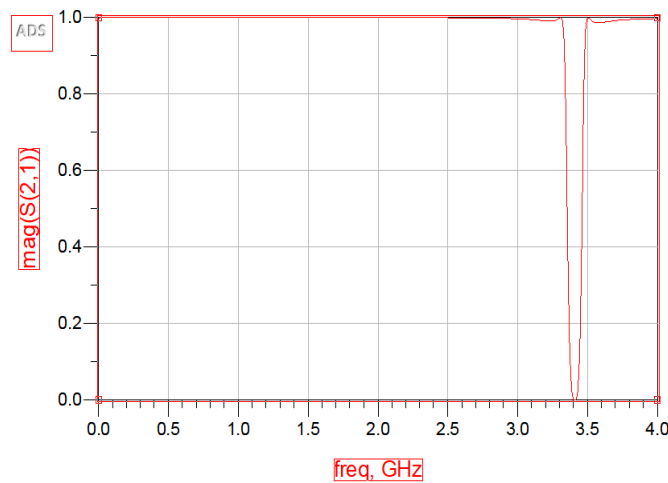


Figura 23 Respuesta del filtro rechaza banda

El código de Matlab usado para la obtención de los valores L-C se muestra a continuación

```
f = 3400*10^6; %frequency
BW = f * 0.05; % Bandwidth
FBW = 0.05;

w0 = 2 * pi * f;
w1 = w0 - (BW/2);
w2 = w0 + (BW/2);
delw = (w2 - w1)/f;
R0= 50;

%N = 3
omgC = 1;
g0=1;
g1 = 1.0316;
g2 = 1.1474;
g3 = 1.0316;
g4 = 1;

%%SHUNT
L1 = R0/(w0*delw*g1)
C1 = (delw*g1)/(w0*R0)

L3 = R0/(w0*delw*g3)
C3 = (delw*g3)/(w0*R0)

%%SERIE
L2 = (g2*delw*R0)/w0
C2 = 1/(w0*g2*delw*R0)
```

Código 1 Código Matlab para el calculo de L-C

**Paso 3** – Cálculo de las dimensiones físicas del filtro rechaza banda ( $W/L$ )

$W/L$  pueden calcularse a través de la opción LineCal de ADS

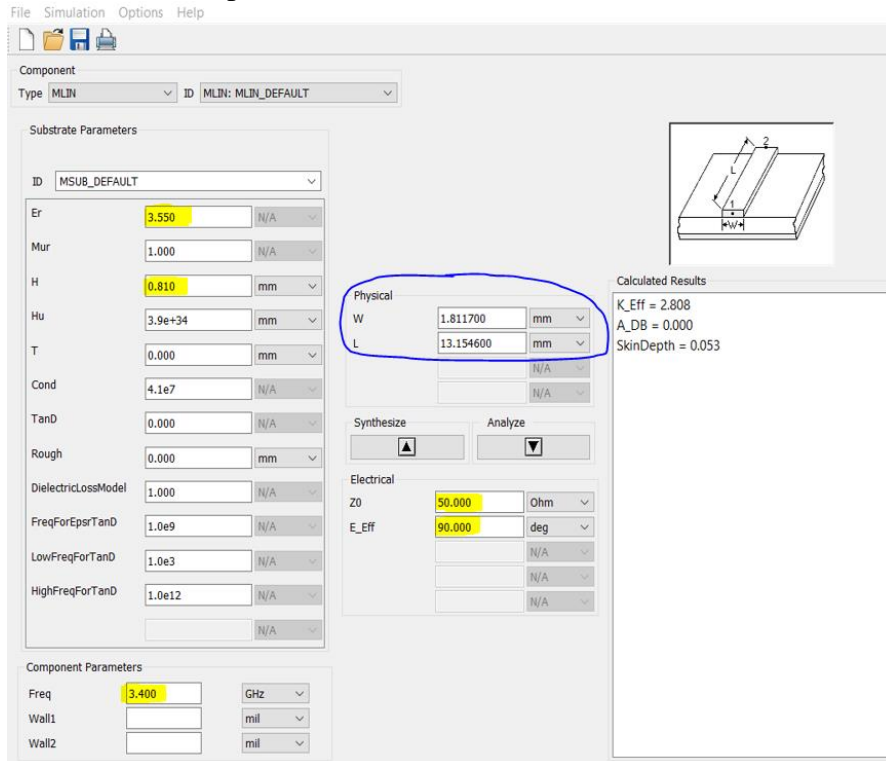


Figura 24 LineCal de ADS para obtener los valores  $W/L$

La topología que se usará para el diseño del filtro rechaza banda será  $\lambda/4$  L-resonador.

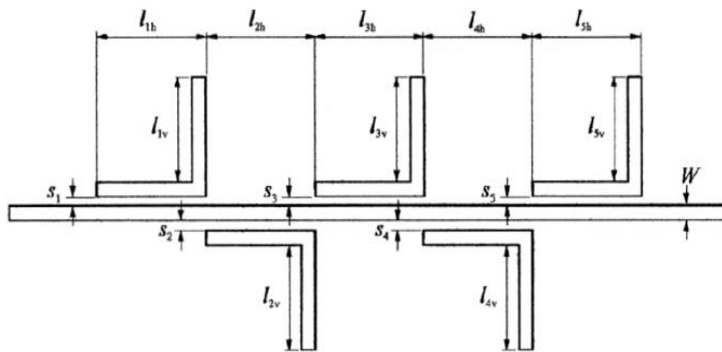


Figura 25 Topología de un resonador tipo L

	Valor	Unidades
W	1.8117	mm
L $\lambda/2$	26.3091	mm
L $\lambda/4$	13.1546	mm

Tabla 11 Valores de  $W$  y  $L$  obtenidos de LineCal (ADS)

Para obtener la relación  $X_i/Z_o$  usaremos las siguientes ecuaciones:

$$\left(\frac{Z_U}{Z_0}\right)^2 = \frac{1}{g_0 g_{n+1}}$$

Ecuación 8

$$x_i = \omega_0 L_i = \frac{1}{\omega_0 C_i} = Z_0 \left(\frac{Z_U}{Z_0}\right)^2 \frac{g_0}{g_i \Omega_c FBW}$$

Ecuación 9

Los resultados son:

Target		
g1=g3	1.0316	19.38735944
g2	1.1474	17.43071292

Tabla 12 Xi/Z0

Código de Matlab:

```
f = 3400*10^6; %freq
BW = f * 0.05; %bandwidth
FBW = 0.05;

w0 = 2 * pi * f;
w1 = w0 - (BW/2);
w2 = w0 + (BW/2);
delw = (w2 - w1)/f;
R0= 50;
Zu = Zo = R0;

%N = 3
omgC = 1;
g0=1;
g1 = 1.0316;
g2 = 1.1474;
g3 = 1.0316;
g4 = 1;

%separaciones x/Z0
x1Z0 = g0/(g1*omgC*FBW)
x2Z0 = g0/(g2*omgC*FBW)
x3Z0 = g0/(g3*omgC*FBW)

%%Resonadores
Qe1 = (g0*g1)/(FBW)
Qen = (g3*g4)/(FBW)

M12 = (FBW)/sqrt(g1*g2)
M23 = (FBW)/sqrt(g2*g3)
```

Código 2 Líneas de código en Matlab para obtener la relación Xi/Z0

Los resultados para un solo resonador son:

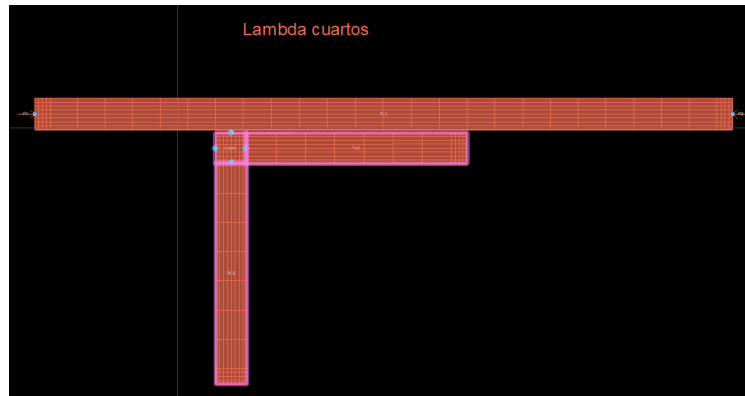


Figura 26 Layout para un resonador tipo L

Ajustando de manera manual la distancia de las líneas de transmisión para cumplir con la relación de  $x/Z_0$  deseada, se llega a los siguientes resultados:

S	0.149	mm
f0	3.2900	GHz
f1	3.2480	GHz
f2	3.3330	GHz
Delta (3dB)	0.0850	
x/Z0	19.35	

Tabla 13 Valor de S para resonadores 1 y 3

S=0.149mm, x/Z0=19.353

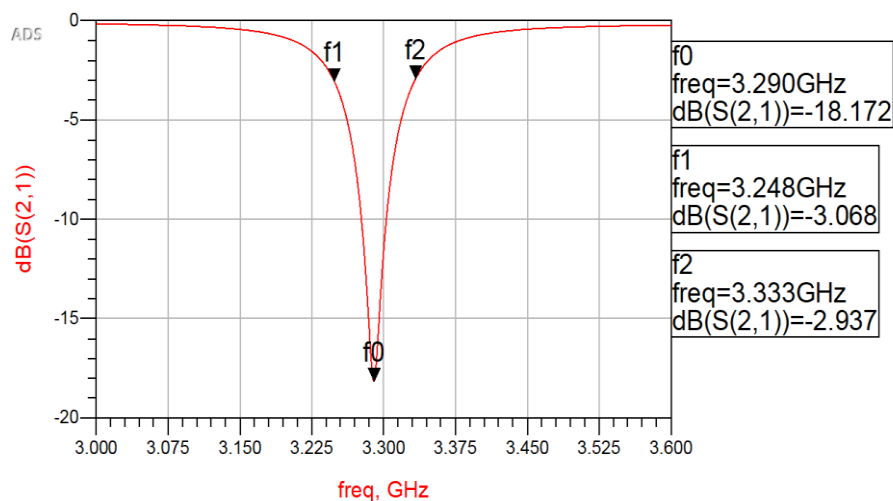


Figura 27 Simulación para un resonador tipo L  $g1=g3$

El cálculo para el resonador  $g2$  con una distancia  $S2 = 0.149\text{mm}$  es:

S	0.121	mm
f0	3.2920	GHz
f1	3.2450	GHz
f2	3.3400	GHz
Delta (3dB)	0.0950	
x/Z0	17.33	

Tabla 14 Valor S para el resonador g2

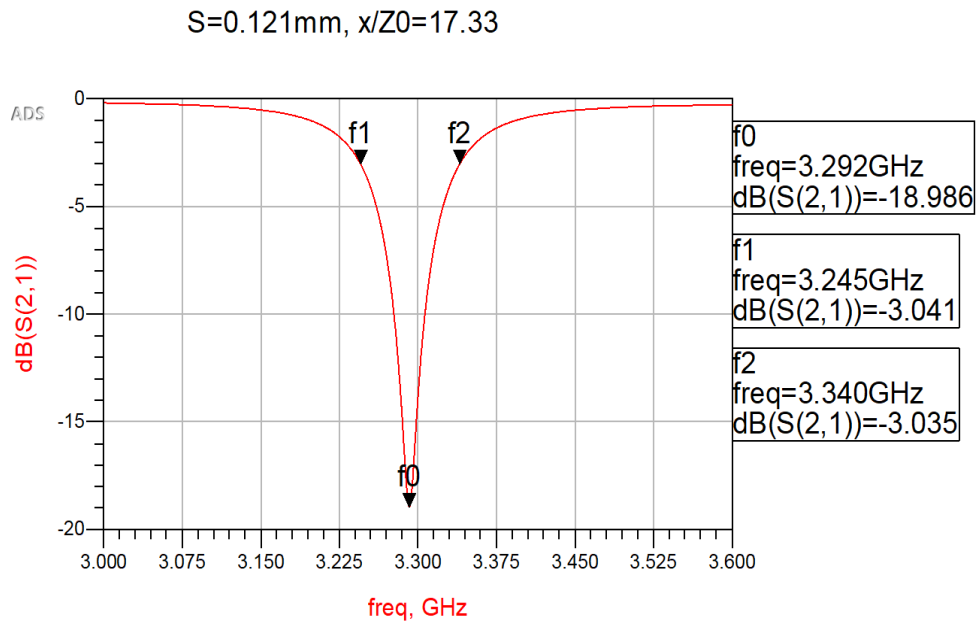


Figura 28 Simulación en ADS del resonador g2

Finalmente, todos los resonadores deben colocarse a la distancia que se muestra en **Error! Reference source not found.** y **Error! Reference source not found.**

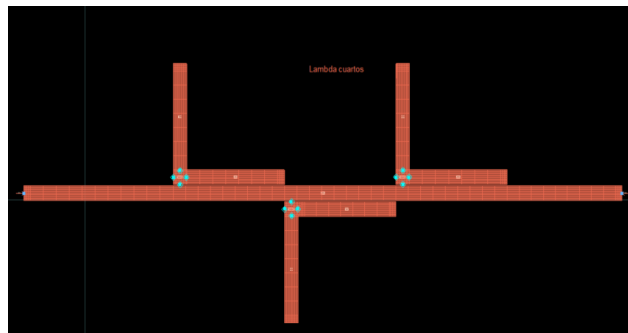


Figura 29 Layout de un filtro rechaza banda completo n-3

La respuesta en frecuencia del filtro se muestra a continuación:

	Valor	Unidades
f0	3.2920	GHz

f1	3.2240	GHz
f2	3.3450	GHz
Delta (3dB)	0.1210	

Tabla 15 Respuesta en frecuencia esperada de la 1ª iteración

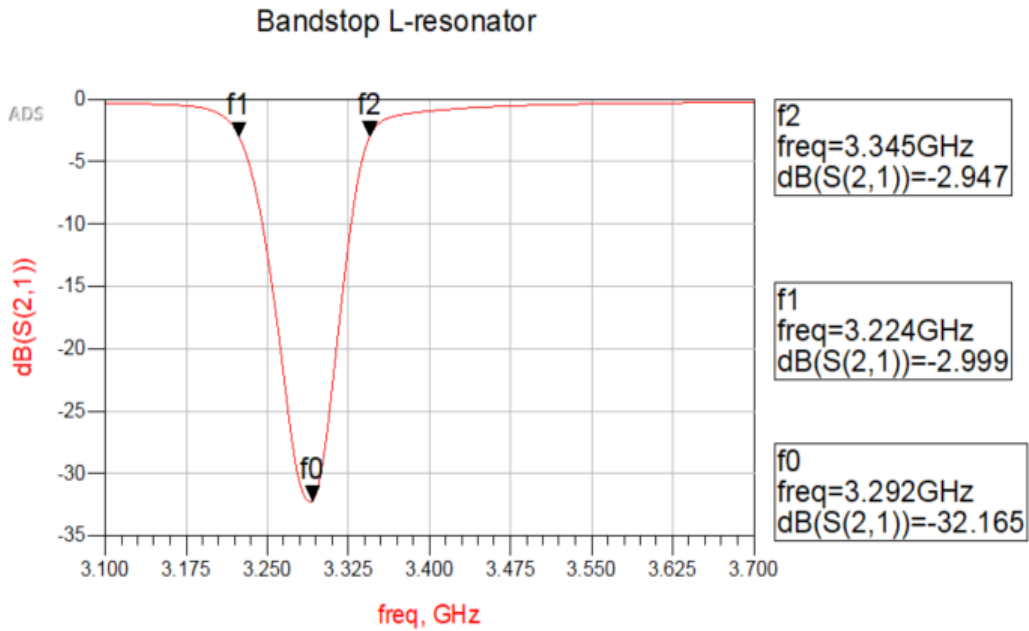


Figura 30 Simulación ADS de un filtro rechaza banda n-3

Como se muestra, la primera iteración de diseño muestra que la simulación, **Error! Reference source not found.** coincide con la respuesta esperada que se muestra en **Error! Reference source not found.** Sin embargo, no cumple con la frecuencia central que se tiene en la especificación (3.4MHz) de diseño mostrada en **Error! Reference source not found.** lo que hace necesario una segunda iteración de diseño para cumplir con el ancho de banda esperado.

Para la segunda iteración, se propuso tomar  $L-\lambda/4$  como 12.7mm. Mediante el uso de LineCal, este cambio afecta el parámetro conocido como permitividad efectiva  $E_{EFF}$  con un cambio de 90 a 86.89deg. El uso de LineCal se muestra en **Error! Reference source not found.**

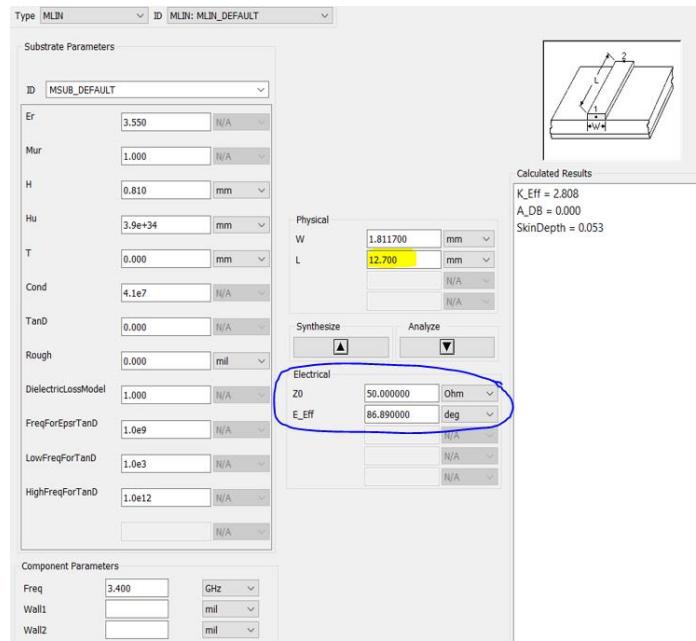


Figura 31 LineCal para L=12.7mm

Debido al cambio en L es necesario ajustar el valor de la distancia S tomando en cuenta que se debe cumplir los valores de  $g_1$  y  $g_3$  mostrados en **Error! Reference source not found.** para lo cual se elige  $S=0.144\text{mm}$

S	0.145	mm
f0	3.4010	GHz
f1	3.3580	GHz
f2	3.4460	GHz
Delta (3dB)	0.0880	
x/Z0	19.32	

Tabla 16 Valores de frecuencia con  $S=0.145\text{mm}$

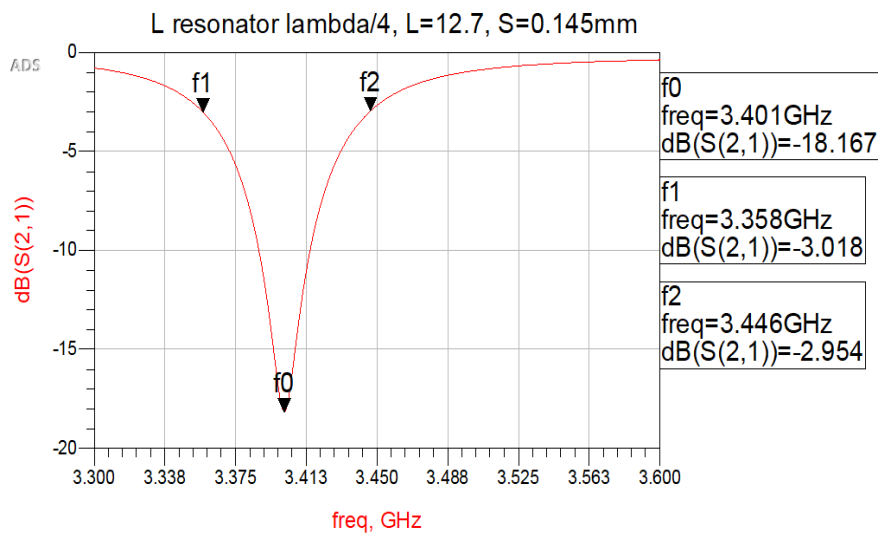


Figura 32 Respuesta en frecuencia de 1 solo resonador tipo L con  $L=12.7\text{mm}$  y  $S=0.145\text{mm}$

Para g2, S se toma como 0.12mm, lo que sigue cumpliendo con **Error! Reference source not found.**

S	0.12	mm
f0	3.4030	GHz
f1	3.3550	GHz
f2	3.4530	GHz
Delta (3dB)	0.0980	
x/Z0	17.36	

Tabla 17 Resonador-L con L=12.7mm para S=0.12mm

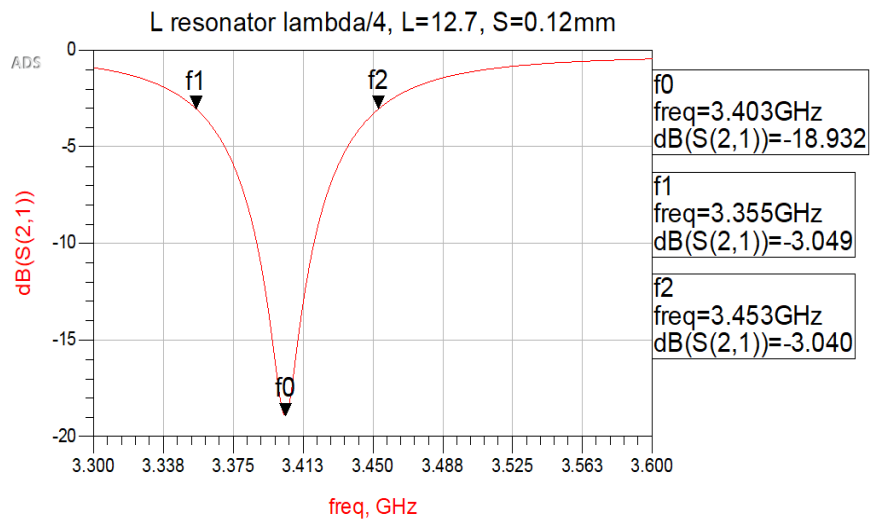


Figura 33 Respuesta en frecuencia para L=12.7mm, g2

Los valores esperados de frecuencia para el resonador tipo L n=3 se muestran a continuación

Resonador-L $\lambda/4$ n=3		
S	Valor	Unidades
f0	3.4000E+09	GHz
f1	3.3320E+09	GHz
f2	3.4570E+09	GHz
Delta (3dB)	1.2500E+08	GHz
Delta BW espec vs sim	4.50E+07	Hz

Tabla 18 Valores esperados para la segunda iteración de diseño

### Bandstop L-resonator, lambda/4, L=12.7

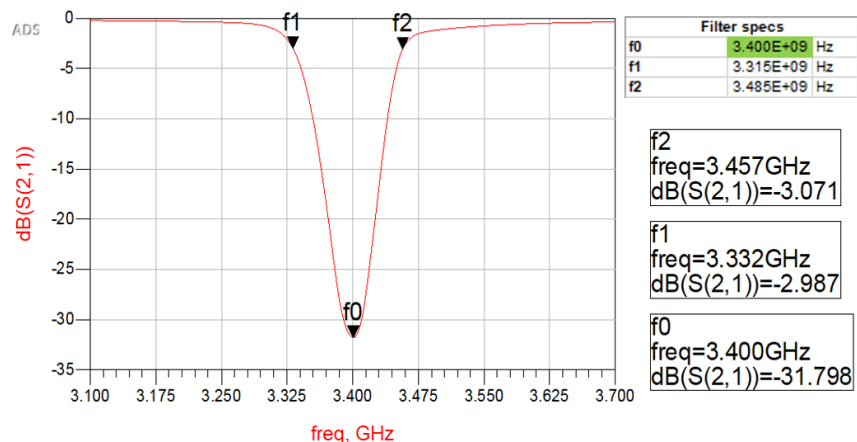


Figura 34 Respuesta en frecuencia del resonador-L con L=12.7mm

## Resultados

Finalmente, ajustando el valor de L, se logra cumplir con la frecuencia central esperada. La siguiente tabla muestra los resultados de diseño de la primera iteración comparados con la segunda corrida de diseño.

Parámetro	L=13.1546, λ/4	L=12.7, λ/4	Unidades
S, g1=g3	0.149	0.145	mm
S, g2	0.121	0.12	mm
f0	3.2920	3.4	GHz
f1	3.2240	3.332	GHz
f2	3.3450	3.457	GHz
Delta (3dB)	121	125	MHz
Delta BW espec vs Sim	49	45	MHz

Tabla 19 Tabla final de resultados

## Conclusiones

Para el filtro rechaza banda mostrado en este reporte, el cambio de L= 13.1546mm a L=12.7mm ayudo a cumplir con la especificación de diseño de la respuesta en frecuencia. Esto muestra el efecto de la longitud L del transistor en el diseño de circuitos integrados.

El cambio en la longitud L, mejoro la frecuencia central del filtro mas no ayudo a cumplir con la expectativa de BW, esta mejora requiere otras técnicas de optimización que están fuera del alcance de la materia de diseño en alta frecuencia.

Herramientas de simulación como ADS, siguen siendo de gran ayuda para el diseño de circuitos electrónicos. De manere rápida y efectiva pueden realizarse varias iteraciones de diseño para lograr las especificaciones deseadas.