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 DE CIRCUITOS INTEGRADOS

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

Presenta: **GUILLERMO SOTO RAMIREZ**

Asesor **DR. OMAR HUMBERTO LONGORIA GÁNDARA**

Tlaquepaque, Jalisco. 27 de agosto de 2021.

Contenido

1. Introducción

El presente reporte contiene documentados tres proyectos relacionados al diseño de filtros en las principales áreas de concentración en la Maestría en Diseño Electrónico del ITESO. Las áreas mencionadas son: Diseño digital, Diseño Analógico y Diseño en Alta Frecuencia. A continuación, se enlistan los proyectos aquí reunidos:

- **Diseño de un ecualizador gráfico,** correspondiente a la asignatura de Procesamiento Digital de Señales.
- **Diseño de un amplificador operacional para PCI Express Gen II,** correspondiente a la asignatura de Diseño de Circuitos Integrados Analógicos
- **Diseño de Filtro Rechaza-Banda,** correspondiente a la asignatura de Diseño Electrónico en alta Frecuencia.

El conocimiento y experiencia que se adquirió al realizar estos proyectos ha sido de gran utilidad en el ámbito laboral debido a que el alumno se desempeña como ingeniero de validación funcional en una empresa de diseño de procesadores, dentro de sus tareas está el análisis y resolución de problemas a nivel sistema, esto contempla capaz de diseño analógico como por ejemplo PCIe, diseño digital que es lógica de un procesador y señales en alta frecuencia como son protocolos de comunicación que utilizan las memorias.

2. Resumen de los proyectos realizados

Los proyectos realizados tienen como objetivo común el diseño de filtros, agregando valor a cada uno de este seleccionado enfoque de diseño diferente en cada caso como se menciona a continuación:

Enfoque de Diseño Digital: Para este proyecto se diseñó un ecualizador dentro de un sistema embebido, tomando como entrada señales analógicas y la ganancia en el punto de frecuencia indicado, convirtiendo estos datos para posteriormente procesarlos de forma algorítmica calculando lo coeficientes de los filtros para finalmente aplicarlos y obtener una salida modulada.

Enfoque de Diseño Analógico: En este proyecto se diseñó un amplificador que cumpliera las especificaciones para PCIe Gen II, para lo cual filtro es completamente analógico implementado una topología de par diferencial usando tecnología MOSFET 130nm.

Enfoque de Diseño en alta Frecuencia: En este proyecto se diseñó un filtro para aplicaciones de microondas en ambientes de comunicaciones, esto se logra utilizando una metodología de transformación de filtros para una implementación final utilizando microcintas.

2.1. Filtro digital – Ecualizador gráfico

2.1.1 Introducción

Este apartado considera el diseño y desarrollo de un ecualizador gráfico basado en un esquema dual, como primera etapa una unidad de control y motor de procesamiento en donde se computan los coeficientes de los filtros dependiendo de las frecuencias de corte especificadas, la segunda parte es un sistema embebido en el cual recibiendo como entradas los coeficientes calculados aplica el algoritmo de filtrado a una señal de audio teniendo como resultado esta misma señal ecualizada.

El propósito de este proyecto es mostrar en forma funcional la operación de un ecualizador digital corriendo en un sistema embebido controlado remotamente y el uso de Matlab para el diseño de filtros.

2.1.2 Antecedentes

Ecualización digital es el procedimiento para ajustar el contenido en frecuencia de una señal de tal forma que la ganancia efectiva es independiente de la frecuencia. En términos de respuesta en frecuencia una respuesta plana o ganancia constate contra frecuencia es deseable para una modulación de alta calidad en la señal generada (Equalization, n.d., p. 30).

Un ecualizador gráfico es un controlador de datos que permite al usuario visualizar y controlar de forma gráfica individualmente un número diferente de bandas de frecuencias en un sistema.

Un ecualizador gráfico típico consiste en una gran cantidad de datos, filtros y amplificadores, cada uno centrado en un rango de frecuencia específica. La mayoría de los ecualizadores gráficos tiene dos conjuntos de filtros/amplificadores idénticos.

2.1.3 Solución desarrollada

La propuesta en este proyecto consiste en dividir la solución en dos partes, una unidad de control y un sistema embebido que funciona como un motor de ecualización, comunicando ambas instancias a través del protocolo ethernet.

La unidad de control permite la selección de frecuencias de corte y ganancias de voltaje, utilizando estos datos para el cálculo de los coeficientes de cada uno de los filtros del sistema.

El sistema embebido recibe datos provenientes de la unidad de control utilizándolos para configurar el controlador de audio Advanced Linux Sound Architecture (ALSA), este último procesa la señal de audio dando como resultado una señal ecualizada.

El sistema está arquitectado utilizando un esquema distribuido como se describe en los párrafos anteriores, la interacción y conexión de todos los elementos del sistema se muestra en la Figura 1-1.

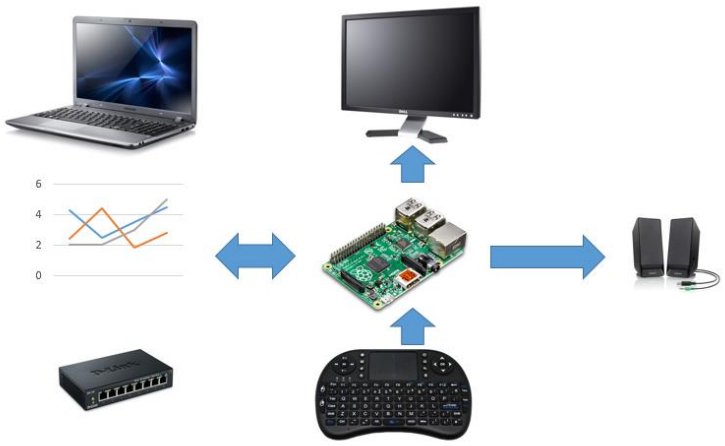


Fig.1- 1 Interacción del sistema de ecualización.

El sistema de control inicia su procesamiento con valores de coeficientes por defecto; estos son cambiados de acuerdo con lo especificado por el usuario en la interfaz de control.

El sistema embebido toma como entrada los datos enviados por la unidad de control (coeficientes) generando dos salidas:

1.- Señal de audio analógica.

2.- Datos ecualizados enviados a la unidad de control como retroalimentación.

El motor de ecualización es el corazón del sistema, se divide en cinco etapas mostradas en la Figura 1-2, este procesa datos aplicando filtros que son configurados con las frecuencias, ganancias y coeficientes provenientes de la unidad de control.

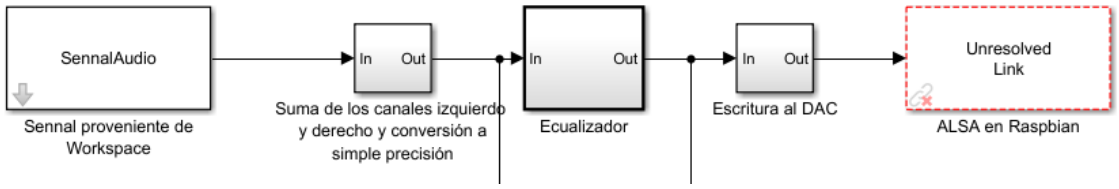


Fig.1- 2 Etapas del ecualizador.

Señal de entrada: Se refiere a la información que será procesada y que podrá provenir de cualquier tipo de sensor.

Ecualizador: Filtro paramétrico de tres bandas derivado en un filtro IIR (Infinite Impulse Response).

DAC: Convertidor analógico digital que provee una nueva señal compensada y digitalizada proveniente del ecualizador.

2.1.4 Análisis de resultados

En el sistema embebido corre una distribución de Linux en la cual se carga el controlador de audio (ALSA)

La Figura 1-3 muestra la respuesta del ecualizador, el usuario puede mover el punto de operación de cada filtro recalculando coeficientes en cada movimiento, enviándolos al sistema embebido y obteniendo como respuesta una señal de audio ecualizada.

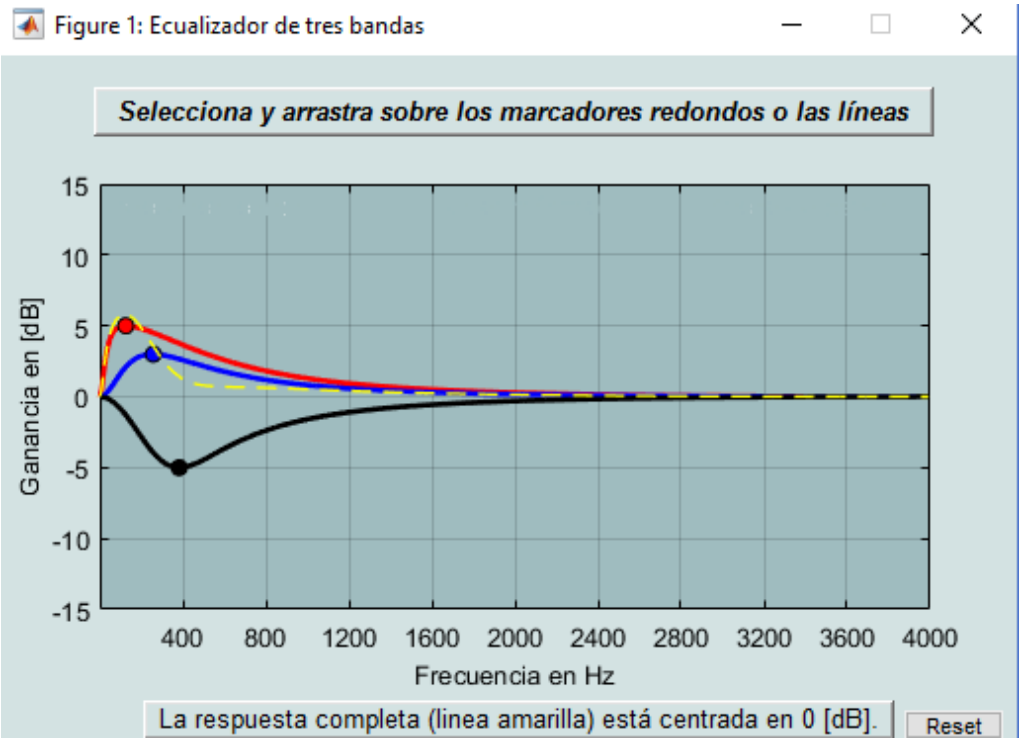


Fig. 1-3. Interfaz de control para los coeficientes de ecualización.

2.1.5 Conclusiones

Conclusiones

El ecualizador diseñado en este proyecto cumplió con el funcionamiento esperado al procesar señales de audio. Sin embargo, esta dificultad puede solucionarse en trabajos futuros implementando filtros de orden superior a dos, de tal manera que la velocidad de caída aumente considerablemente; aunque esta operación incrementa la dificultad del diseño, además de acarrear costos adicionales.

Mejoras

- Hacer un filtro más general para procesar un mayor rango de frecuencias.
- Incrementar el número de bandas.
- Mejorar el algoritmo para hacer un procesamiento en tiempo real.
- Mejorar el protocolo de comunicación.

- Incrementar la autonomía del algoritmo.
- Implementar un control distribuido.
- Mejorar la interfaz gráfica.

2.2. Diseño de un amplificador operacional para PCI Express Gen II

2.2.1 Introducción

Un amplificador se describe como un circuito capaz de procesar señales adecuándolas para alguna aplicación. El amplificador permite mantener o mejorar la presentación de una señal suministrada en un sistema .

Se llama amplificador multietapa a los circuitos o sistemas que tienen múltiples transistores y además pueden ser conectadas entre sí para mejorar sus respuestas tanto en ganancia, Zin, Zout o ancho de banda. Las aplicaciones pueden ser tanto de cc como de ca. (R. Carrillo, s.f.)

Este proyecto se enfoca en diseñar un amplificador diferencial que cumpla con las especificaciones de diseño eléctrico para PCIe generación II, la cual está definida por los parámetros mostrados en la tabla 2-1, (Eads, s.f.)

V_{DD} = + 1.2 V	V_{icm} = 0.7× V_{DD}
V_{SS} = 0 V	V_{ocm} = 0.7× V_{DD}
A_v = 40 dB	BW = 5 GHz
$P_{Dmáx}$ <= 1 mW	C_L = 20 fF
K_n = 200 µA/V^2	V_{TH} = 0.35 V

Tabla 2-1. Especificaciones de diseño para un amplificador PCIe Gen II.

2.2.2 Antecedentes

La mayoría de los amplificadores integrados tienen entrada diferencial. Para realizar esta entrada diferencial, casi todos los amplificadores usan lo que comúnmente se le llama un par diferencial de transistores. Un par diferencial junto con una fuente de corriente es mostrado en la Figura 2-1, los transistores Q1 y Q2 son de tamaño idéntico y son saturados al mismo voltaje de DC de compuerta. Un modelo de baja frecuencia y de pequeña señal del par diferencial es mostrado en la Figura 2-1. Este circuito equivalente de pequeña señal está basado en un modelo T para transistores MOS. (Carusone, 2012)

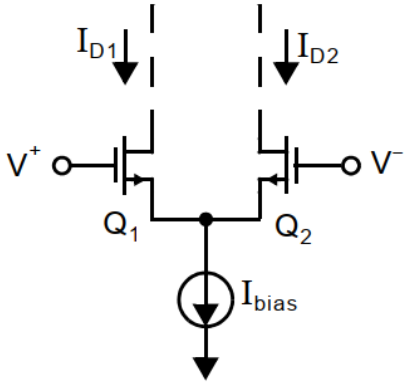


Fig. 2-1. Par diferencial MOS.

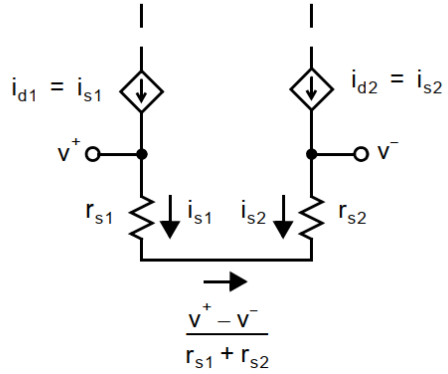


Fig. 2-2. Modelo de pequeña señal de un par diferencial MOS.

2.2.3 Solución desarrollada

Como punto inicial se eligió una arquitectura OTA self-CASCODE que se deriva a partir en un amplificador diferencial mostrado en la Figura 2-3. Un amplificador PCIe II tiene los

siguientes requerimientos mostrados en la tabla 2-1 (Para los transistores del par diferencial se considera L=0.24um).

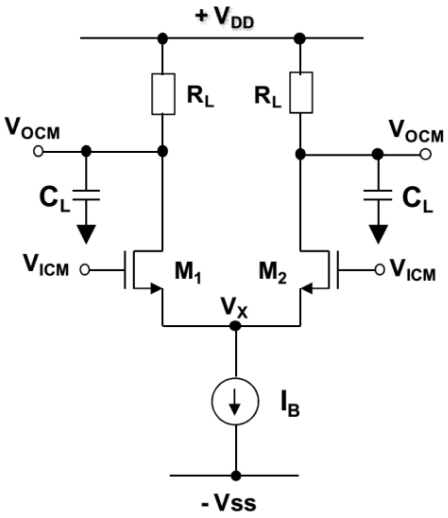


Fig. 2-3. Amplificador diferencial.

Una vez que obtenidas las especificaciones se procede a realizar los cálculos correspondientes para cumplir con el diseño. El desarrollo detallado de cado uno de los cálculos necesarios se encuentra en el apéndice B.

La tabla 2 -2 muestra de forma resumida los resultados calculados para cada uno de los parámetros necesarios en la implementación de este diseño.

Parámetro	Valor
I_{Bmax}	833 µA
R_L	1591.55 Ω
V_{icm}	0.84 V
V_{ocm}	0.84 V
I_B	452.39 µA
G_m	1.254 mS
K_n	315.398 µ $\frac{A}{V^2}$
W	2.64 µm

Tabla 2-2. Resultados de cálculos parámetros de diseño.

Como segunda parte se procede a realizar la verificación de la respuesta en frecuencia usando la herramienta CADENCE Virtuoso usando tecnología de 130 nm. La Figura 2-4 muestra la implementación del circuito esquemático.

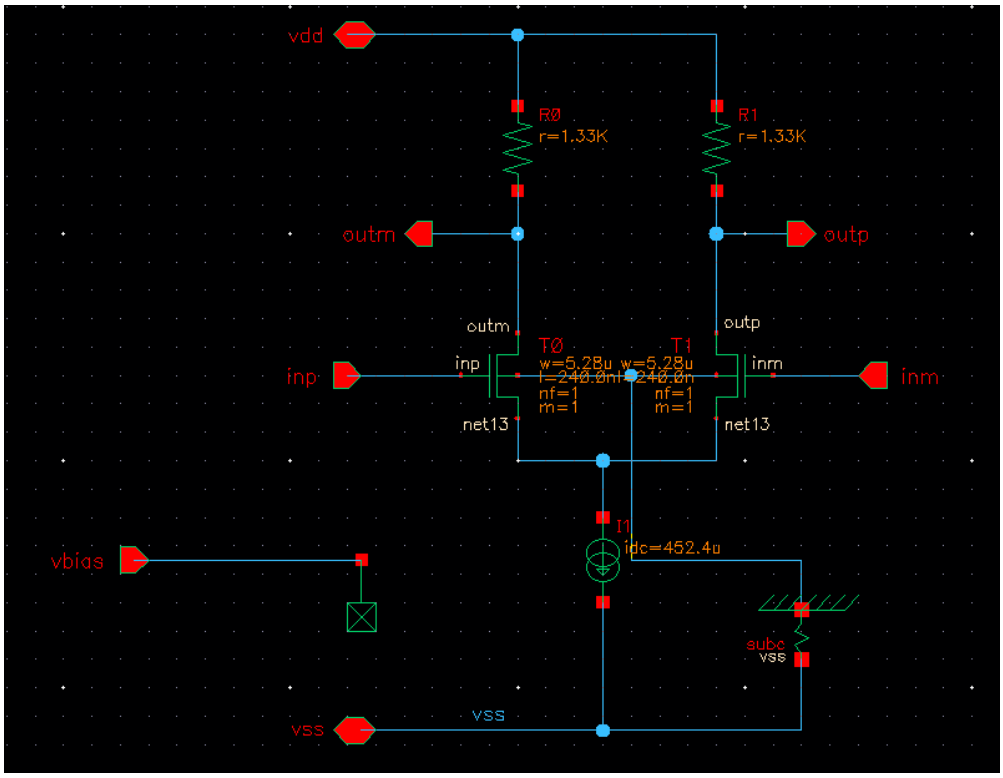


Fig. 2-4. Implementación del amplificador diferencial.

La simulación se realizó mediante un análisis en AC con barrido en frecuencia iniciando en l 1Hz y finalizando en 10 Ghz, tomando 100 puntos por década. La Figura 2-5 muestra el diagrama bode de frecuencia contra ganancia de voltaje.

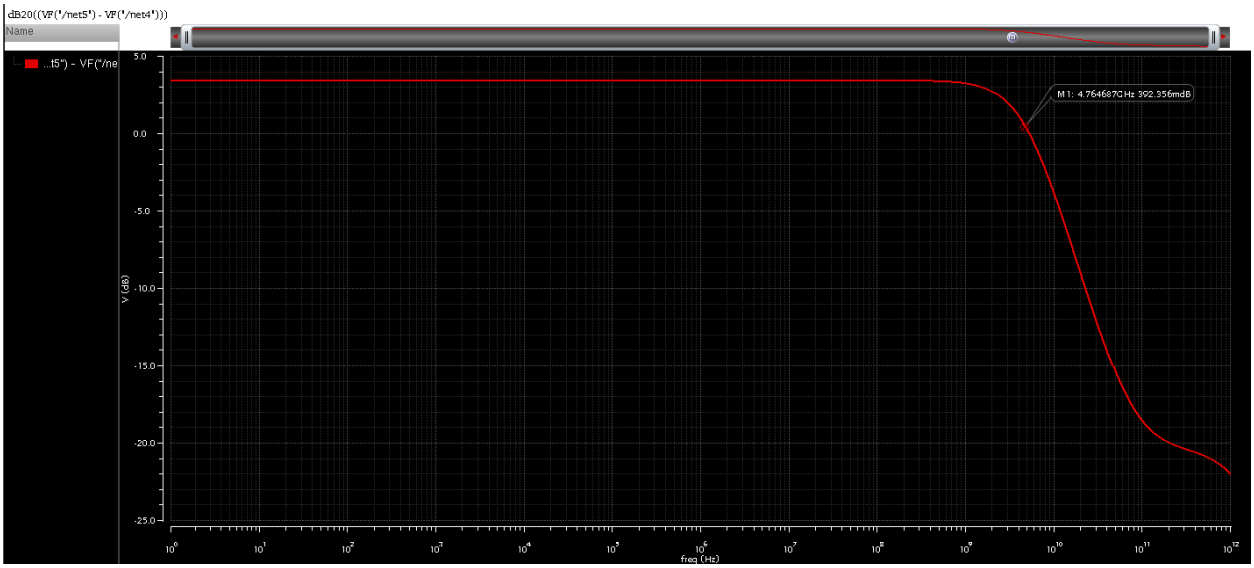


Fig. 2-5. Diagrama bode frecuencia contra ganancia de voltaje.

En esta primera aproximación no se cumple con el criterio de ganancia de voltaje y el ancho de banda, por lo que se continúa recalculando parámetros.

La herramienta de simulación tiene la capacidad de obtener el punto de operación usando los valores calculados de W y L (consultar tablas en apéndice B, apartado “Operating point analysis results”).

Tomando como base las fórmulas de ancho de banda (2-1) y ganancia (2-1) es posible identificar los parámetros adecuados para que al realizar su variación se obtenga BW y Av.

$$BW = \frac{1}{R_L \cdot C_L} \qquad (2\text{-}1)$$

$$A_v \cong g_m R_L = \sqrt{K_n I_B \left(\frac{W}{L}\right)} \cdot R_L \qquad (2\text{-}2)$$

En la formula (2-1) C_L está dentro de la especificación, por lo cual el único parámetro que debe ser modificado es R_L, un caso similar para de la formula A_v es posible identificar a W como el parámetro variable.

Después de varias iteraciones al variar R_Ly W fue posible obtener el valor correcto de A_v. Consultar detalle de dichas iteraciones es mostrado en el apéndice B.

La Figura 2-6 muestra el resultado final de la simulación incluyendo los cálculos finales de cada uno de los parámetros de diseño.

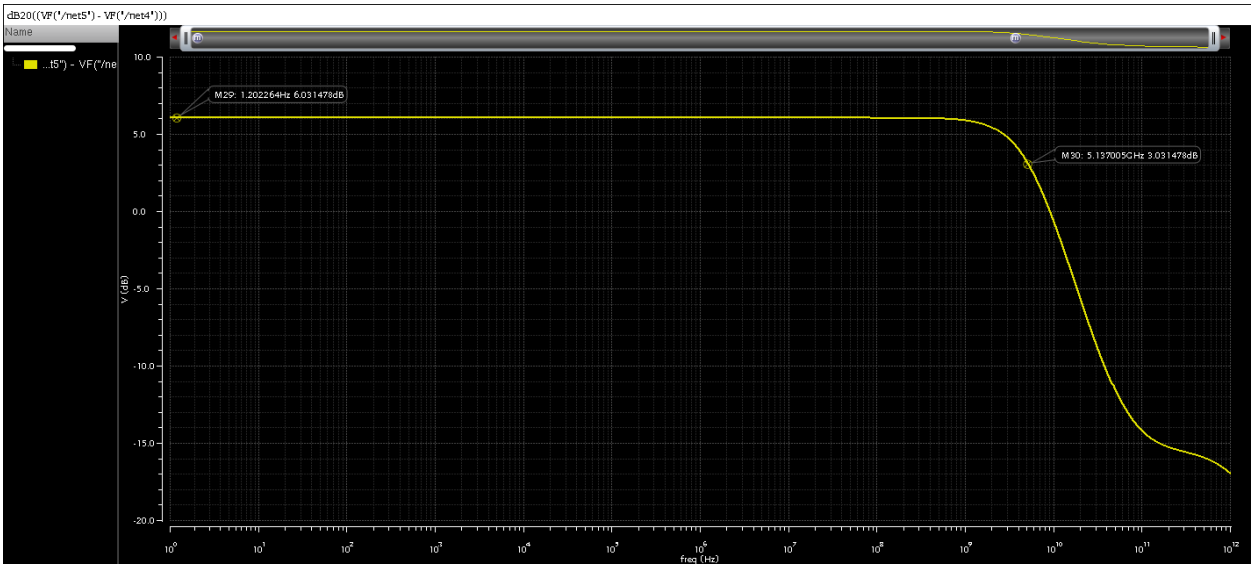


Fig. 2-6.

2.2.4 Análisis de resultados

Con todos los parámetros teóricamente calculados la respuesta en frecuencia no fue como se esperaba, el ancho de banda fue reducido a 4.78 GHz y la ganancia de voltaje menor que 5db.

En la mayoría de los casos será necesario ajustar los parámetros de diseño para lograr las salidas deseadas pero lo importante es saber que necesita ser cambiado con base en los modelos matemáticos en el caso de $R_L$esta ligada con BW y W con la ganancia.

Todas las variaciones son proporciones, no valores absolutos, esto obedece a uno de los principios de diseño.

2.2.5 Conclusiones

De acuerdo con los resultados obtenidos el parámetro de diseño con mayor impacto es R_L porque el impacto no solo es en BW, si no, también en A_v esto es debido a las fórmulas matemáticas.

Una de las cosas más importantes para tomar en cuenta es caracterizar correctamente todos los elementos de diseño, en este caso fue necesario extraer el punto de operación de los transistores para obtener el valor correcto de Kn sin que el resto del diseño pueda estar incorrecto.

Hay una diferencia entre los valores calculados y los resultados de la simulación, esto es porque las herramientas de simulación toman en cuenta parámetros físicos que normalmente los cálculos teóricos no.

Esto remarca que todas las fórmulas son aproximaciones simplistas de un comportamiento real, es por esto por lo que el uso de simuladores resulta en una herramienta sumamente poderosa para llegar a obtener un comportamiento más apegado a la realidad.

2.3. Filtro en alta frecuencia – Diseño de Filtro Rechaza-Banda

2.3.1 Introducción

El proyecto presenta el diseño de un filtro rechaza-banda para aplicaciones de microondas, la implementación de este diseño se realiza utilizado líneas de microcintas (substrato RF4) y centradas en una frecuencia de 3.4 Ghz. El filtro ha sido simulado usando la herramienta de software “ADS”.

2.3.2 Antecedentes

La diferencia entre teoría de circuitos y teoría de líneas de transmisión es la cantidad de energía eléctrica. Análisis de circuitos asume que las dimensiones físicas de las redes son mucho más pequeñas que las longitudes de onda, mientras que las líneas de transmisión podrían ser consideradas una fracción de la longitud de onda o muchas longitudes de onda en tamaño. Así una línea de transmisión es una red de parámetros distribuido, donde los voltajes y corrientes pueden variar en magnitud y fase sobre su longitud, mientras que el análisis de un circuito ordinario lidia con elementos concentrados, donde los voltajes y corrientes no varían apreciablemente sobre las dimensiones físicas de los elementos. (Pozar)

2.3.3 Solución desarrollada

Para el desarrollo de este filtro se tiene las especificaciones mostradas en la tabla 3-1.

Frecuencia central	Fracción de ancho de banda	Orden del filtro	Tipo de respuesta en frecuencia	Impedancia de referencia
3.4 GHz	5%	5	0.1dB Chebyshev	50Ω

Tabla 3-1. Especificaciones de filtro rechaza-banda.

Microcinta ROGER RO4003

h	Cladding	er
0.81 mm	0.5 oz	3.55

Diseño

Nota: El diseño detallado del filtro se puede consultar en el apéndice C en la sección de diseño.

- Como primer paso se obtiene un filtro pasa-bajas normalizado con valores correspondientes a una n=5.
- Una primera aproximación para el filtro pasa-bajas se realiza usando elementos discretos y como la “n” es impar la topología del circuito inicia con una carga

resistiva. La Figura 3-1 muestra la implementación de un filtro pasa-bajas normalizado en impedancia y con una n=5.

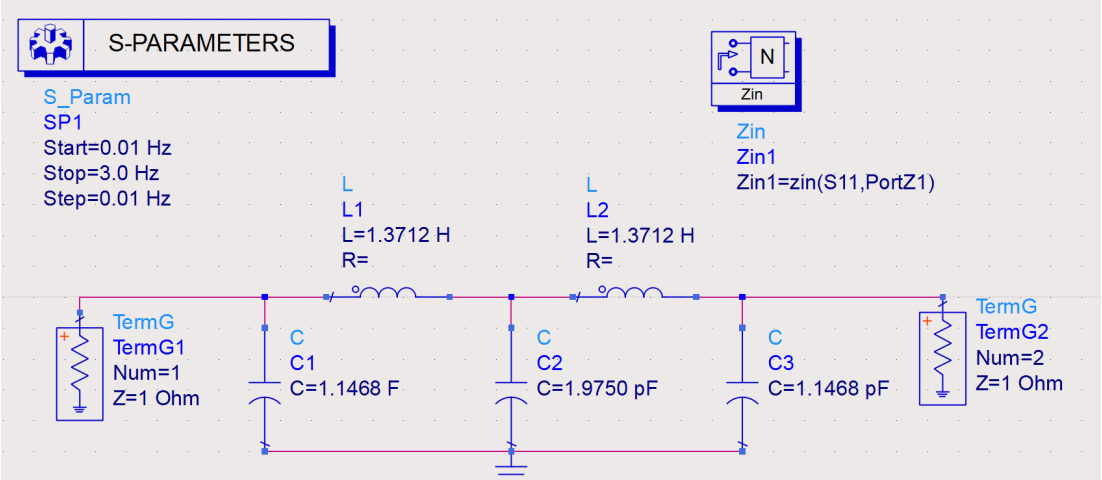


Fig. 3-1. Filtro pasa-baja normalizado en impedancia con n=5.

Simulación: El circuito fue simulado de 0.01Hz a 0.3Hz, el resultado es mostrado en la Figura 3-2.

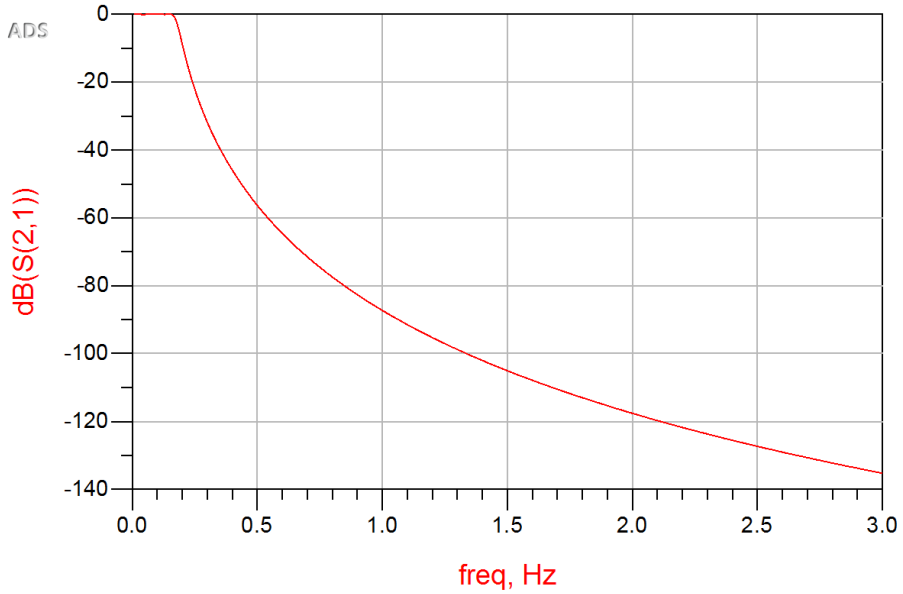


Fig. 3-2. Respuesta del filtro pasa-baja con impedancia normalizada.

Transformación de filtro

- El siguiente paso es una desnormalización de cada elemento en impedancia y frecuencia.
- La Figura 3-3 muestra el filtro pasa-bajas después de habérsele aplicado la escala en impedancia y frecuencia. Puede observarse valores correspondientes a cada elemento a magnitudes más realistas.

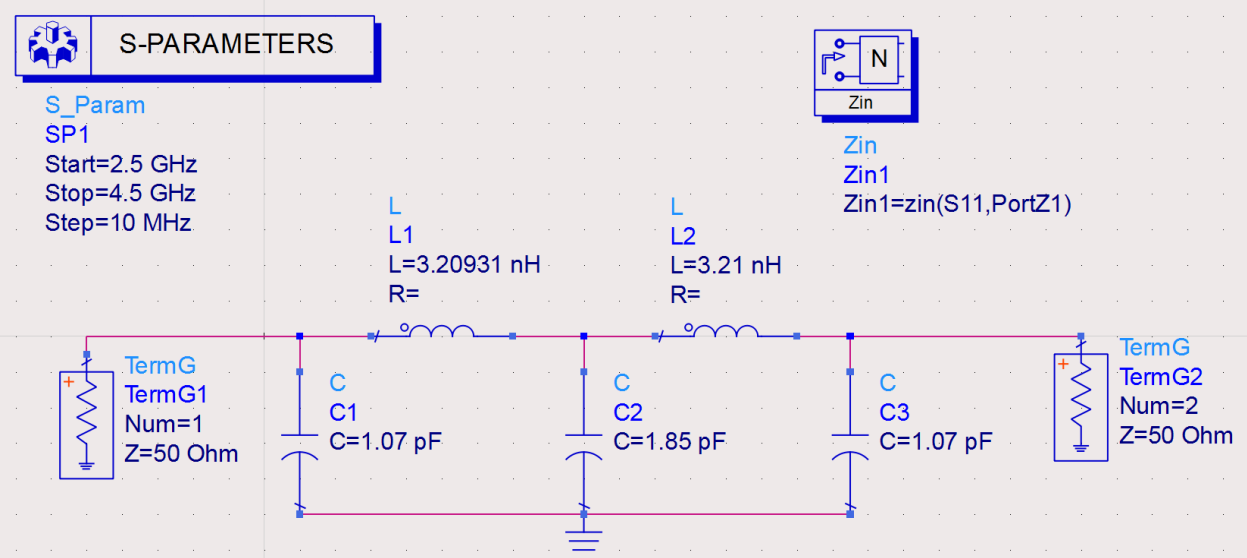


Fig. 3-3. Filtro pasa-baja después de escalamiento en impedancia y frecuencia.

Simulando el circuito de la Figura 3-3 para obtener la respuesta en frecuencia de 2.5 Ghz a 4.5 Ghz es posible verificar el ancho de banda correspondiente a -3dB en magnitud. El resultado es mostrado en la Figura 3-4, se observa que el filtro tiene 3.8 Ghz de ancho de banda y casi 0% de perdida a 3.4ghz.

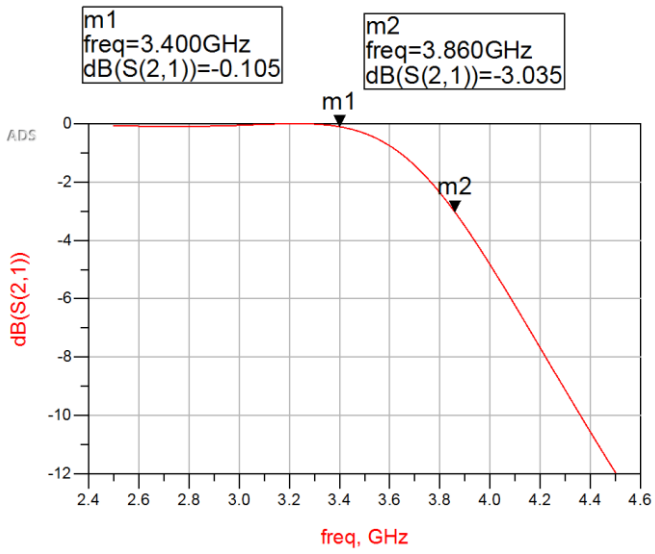


Fig. 3-4. Respuesta en frecuencia de 2.5 Ghz a 4.5 Ghz.

Transformación a rechaza-banda.

Si ω_1 y ω_2 denotan los flancos de un filtro pasa-banda, entonces la respuesta de un filtro rechaza-banda puede obtenerse la siguiente sustitución en frecuencia (3-1). La frecuencia central es ω_0.

$$\Delta = \frac{\omega_2}{\omega_0} \qquad (3\text{-}1)$$

Acerca el resto de los elementos ene l circuito, existe una conversión predefinida. Libro "Microwave Engineering" pagina 404, tabla 8.6. La transformación de la topología es implementada en forma circuital como se muestra en la Figura 3-5.

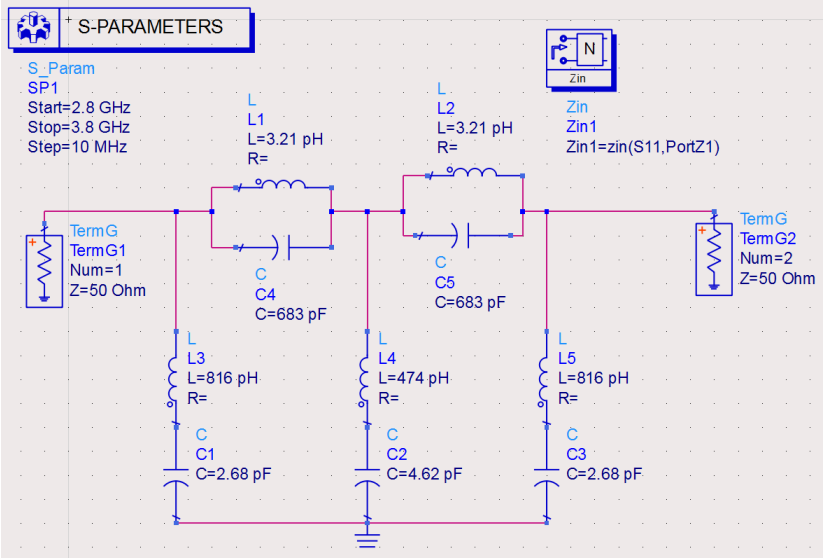


Fig. 3-5. Transformación filtro pasa-baja a rechaza-banda.

Respuesta en frecuencia de filtro rechaza-banda con frecuencia central en 3.4 Ghz y un ancho de banda de 170 Mhz, se muestra en Figura 3-6.

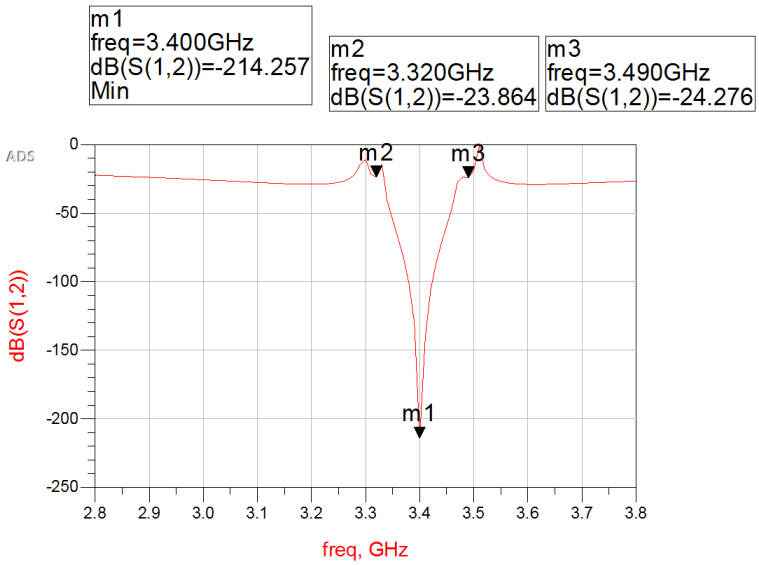


Fig. 3-6. Respuesta del filtro rechaza-banda.

A continuación, se realiza una primera aproximación con líneas de microcintas ideales. Implementación Figura 3-7 (Microwave engineering, David Pozar, tercera edición pag. 427)

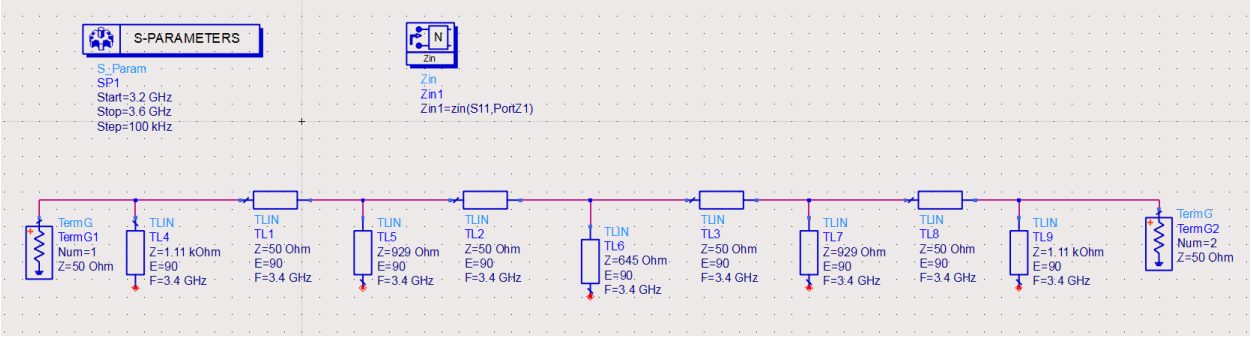


Fig. 3-7. Primera aproximación con microcintas ideales.

El comportamiento del filtro usando microcintas ideales cumple con las especificaciones, tal como se muestra en la Figura 3-8.

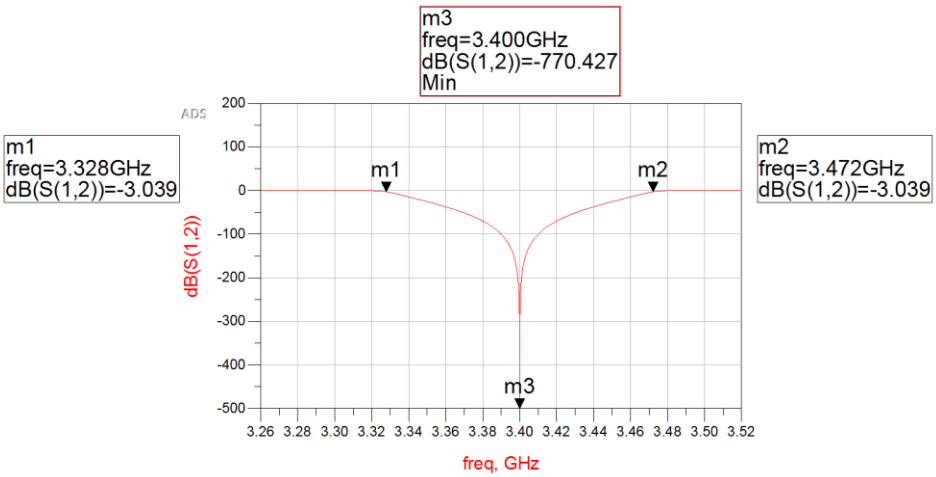


Fig. 3-8. Respuesta filtro rechaza-banda usando microcintas ideales.

Las dimensiones de L y W son calculadas usando la herramienta de cálculo del simulador ADR, el resultado de estos cálculos es mostrado en la Figura 3-9.

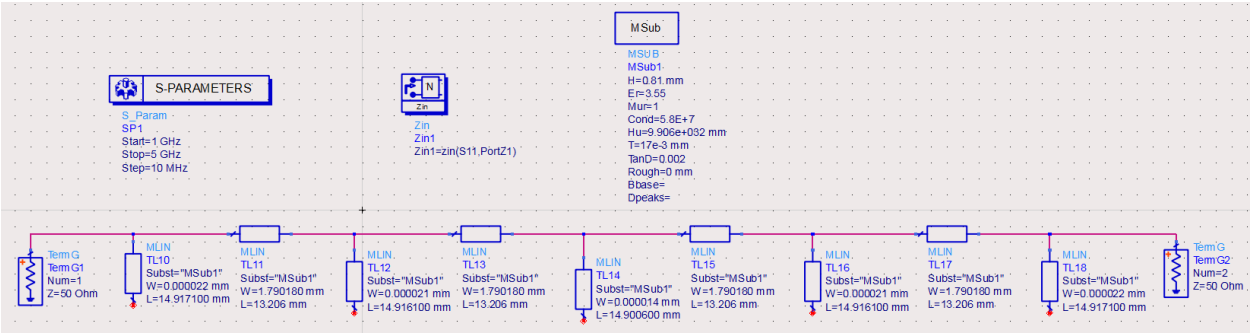


Fig. 3-9. Implementación con segmentos de microcintas para el filtro rechaza-banda.

2.3.4 Análisis de resultados

Para incrementar el ancho de banda se agregaron resonadores L de la misma dimensión. La separación entre la línea principal se conserva mientras se varia la longitud de cada uno de los resonadores L. Los resonadores L mantienen el tamaño su segmento horizontal y tienen pequeñas variaciones en el segmento horizontal. Figura 3-10 muestra el resultado final con las variaciones en cada resonador.

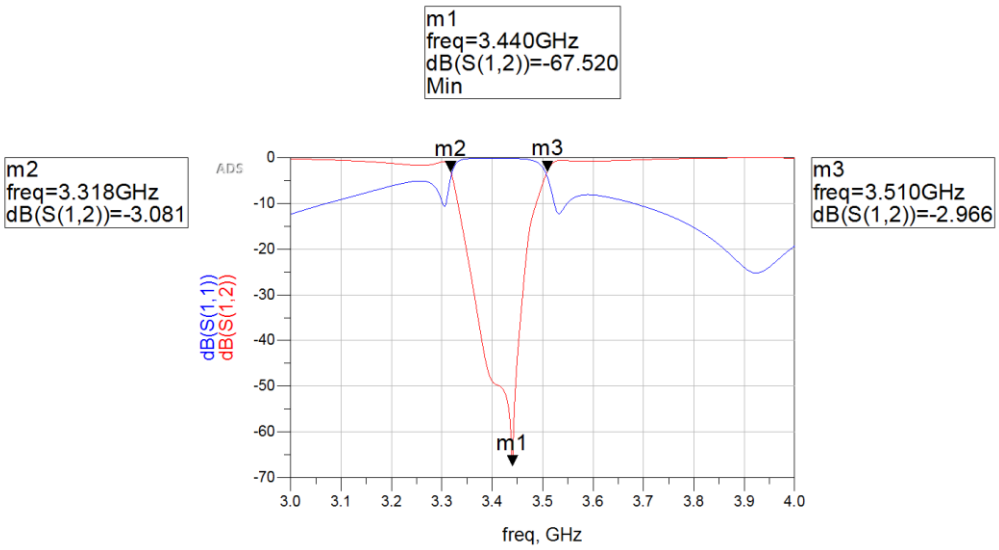


Fig. 3-10. Resultado final filtro rechaza-banda.

2.3.5 Conclusiones

El diseño del filtro se realizó siguiendo un método tradicional de conversión de tipo de filtros, las diferencias se presentan en al momento de implementación con microcintas que por características físicas tiene una variación en la respuesta con respecto a lo teóricamente calculado, para compensar esto es necesario hacer adecuaciones de forma “experimental” en el tamaño de los segmentos de cada resonador, estos ajustes deben obedecer a una proporción en sus segmentos verticales y horizontales. Es de suma importancia contar con las herramientas adecuadas para el tipo de simulación que se realiza, en este caso el simulador ADR permitió debido a que toma en cuenta parámetros físicos que impactan en el resultado final y esto brinda mayor certeza en cómo será el comportamiento en la implementación real.

3. Conclusiones

Con elección de los proyectos documentados en este reporte se logró abarcar las principales vertientes de diseño, digital, analógico y alta frecuencia resolviendo un mismo tipo de problema que fue diseño de filtros, dado esto, en todos los casos el principio de funcionamiento es el mismo permitiendo soluciones desde tres perspectivas diferente.

El ecualizador digital tiene la ventaja de ser altamente configurable lo cual permite observar su comportamiento a diferentes frecuencias de operación. Este proyecto se basa en las ventajas que ofrece el procesamiento digital de señales, una vez que una señal eléctrica se digitaliza se transforma en datos los cuales pueden ser manipulados con gran facilidad utilizando software o lógica digital, una pieza clave para la implementación fue la utilización de herramientas como Simulik MatLab permite enforcase más en la aparte algorítmica.

El amplificador para PCIe Gen II proporciono un enfoque innovador en el diseño de un Amplificadores Operacionales de Transconductancia (OTA) al implementar un diseño de par diferencial, el amplificador debe cumplir con especificaciones muy estrictas, para lograrlo se realizaron los cálculos a mano como una primera aproximación, pero el uso de las herramientas de simulación (*Virtuoso*) permite mover parámetros con conocimiento de causa y observar la respuesta que produce.

El filtro rechaza-banda fue un reto al momento de la implementación con microcintas, esto es debido a que para el rango de frecuencias en ultra ancho de banda las dimensiones físicas cobran una vital importancia en la respuesta del filtro, el uso de herramientas de simulación especializadas como él (ADS) permite aplicar variaciones en parámetros necesarios para obtener el resultado deseado.

Apéndices

APÉNDICE A

Graphic equalizer

3.1. Introduction

What is digital equalization?
Digital equalization is the process of adjusting the frequency content of the signal so that the effective gain is independent of the frequency. In terms of frequency response, a flat response or constant gain versus frequency is desirable for highest modulation quality in the generated signal. Reference: NI-Response and Software Equalization

For many physical channels, such as telephone lines, not only are they bandlimited, but they also introduce distortions in their passbands. Such a channel can be modeled by an LTI filter followed by an additive white gaussian noise (AWGN) source.
Reference: Intersymbol Interference and Equalization

Types of equalizers

- Shelving EQ
  In shelving equalization, all frequencies above or below a certain point are boosted or attenuated the same amount. This creates a "shelf" in the frequency spectrum.

- Bell EQ
  Bell equalization boosts or attenuates a range of frequencies centered around a certain point. The specified point is affected the most, frequencies further from the point are affected less.

- Graphic EQ
  Graphic equalizers provide a very intuitive way to work — separate slider controls for different frequencies are laid out in a way which represents the frequency spectrum. Each slider adjusts one frequency band so the more sliders you have, the more control.

- Parametric EQ
  Parametric equalizers use bell equalization, usually with knobs for different frequencies, but have the significant advantage of being able to select which frequency is being adjusted.

Parametric are found on sound mixing consoles and some amplifier units (guitar amps, small PA amps, etc.).

IIR Filters

Infinite impulse response (IIR) filters are the most efficient type of filter to implement in DSP (digital signal processing). They are usually provided as "biquad" filters. For example, in the parametric EQ block, each peak/notch or shelving filter is a single biquad. In the crossover blocks, each crossover uses up to 4 biquads. Each band of a graphic EQ is a single biquad, so a full 31-band graphic EQ uses 31 biquads per channel.

The amount of processing that is required to compute a biquad is relatively small. This is what enables the low-cost products to implement a full active crossover with parametric EQ on all input and output channels. The DSP (digital signal processor) on each board can compute a certain number of biquads, and this is the primary thing that determines how many filters are available.
Reference: FIR vs IIR filtering

Control

A control system (CS) is a platform for automated control and operation of a plant or industrial process. A CS combines the following into a single automated system: human machine interface (HMI), logic solvers, historian, common database, alarm management, and a common engineering suite. Reference: Yokogawa

A traditional control system (CS) is built for only one purpose, which is process control. In the competitive economy today, you must consider forward-thinking possibilities and use technology and innovation to your advantage. It is time to rethink what to expect from a CS. To drive productivity, increase efficiencies, and reduce costs, you must integrate all your automation operations to achieve The Connected Enterprise. You can accomplish this integration with the technology that a modern CS.
Reference: Rockwell

Embedded system
An Embedded system is a combination of computer hardware and software. As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller. The Embedded system hardware includes elements like user interface, Input/Output interfaces, display, and memory, etc. Generally, an embedded system comprises power supply, processor, memory, timers, serial communication ports and system application specific circuits. Reference: classification of embedded systems

3.2. Goal and objetives

3.2.1 Functional description

A graphic equalizer is a data control that allows the user to see graphically and control individually several different frequency bands in a system. A typical graphic equalizer consists of several data filter/amplifiers, each centered at a specific frequency range. Most graphic equalizers have two identical sets of filter/amplifiers.
The system described in this project consist of an embedded system that servers as equalizer engine, a control host connected through Ethernet with the engine.
The embedded system gets the data from some IO pin (any kind of sensor) and processed through sending the output data to the control host.
The control host can show the data coming from equalizer engine and modifying over run time the parameter (coefficients) of the equalizer.

3.2.2 Project purpose

The purpose in this project is to show in functional way the operation of the equalizer running over an embedded system controlled remotely, the use of matlab design filters and to generate source code.

3.3. Architecture

3.3.1 High Level architecture

This system is architected for to divide an equalizer system in two sections data processing and control, figure 1 show the interaction between all elements in the system.

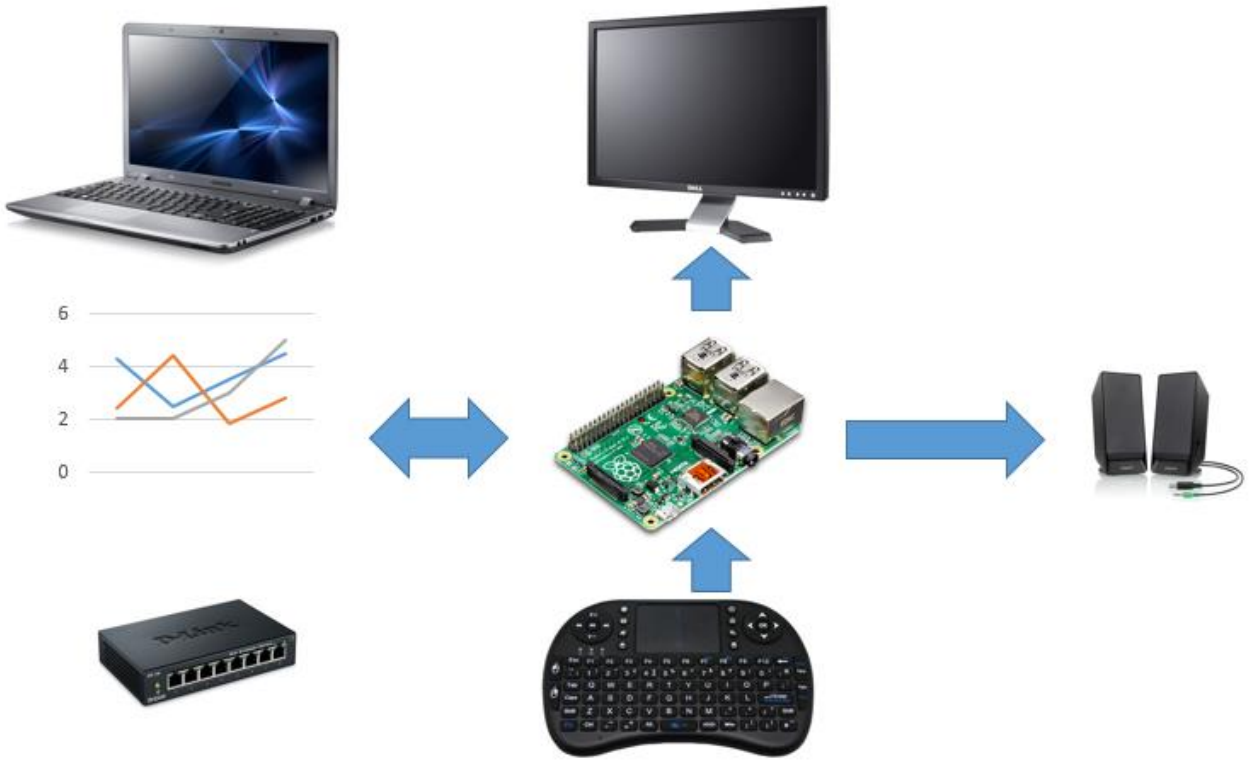


Figure 1. System interaction.

The equalizer engine star its process with a default coefficient values and immediately these are changed by the control according with the specifying in the interface.

The embedded system took the two inputs it means data from a sensor and the coefficients to control to generate two outputs:

1. Analog signal to inject to another system.
2. Equalized data sent to feedback to the remote-control host.

3.3.2 Micro-architecture

This case only describes the heart of the processing which is the equalizer engine divided into five stage showing in figure

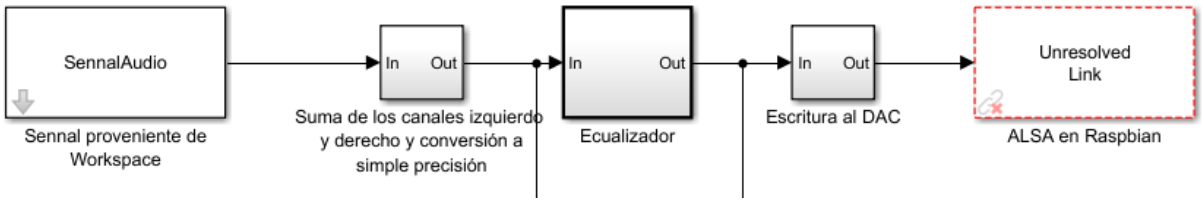


Figure 2. 5 stage equalizers.

Input signal: This is referent to the information to be processed, it can come from any kind of sensors.
Equalizer: Three band parametric filter derivate from IIR filter.
DAC: Digital to analog converter to provide a new compensated signal from the equalizer to a system that that use the new signal to some purpose.

3.4. Design

3.4.1 Matlab source code

```
function GUI_ecualizador_audio(state)
%GUI_ecualizador_audio

persistent Fs NdB axisStyle

axisStyle = 'normal';

%Definición de los colores para la presentación en la interfaz de usuario
color1 = [255 0 0]/255;     % Red
color2 = [0 0 255]/255;     % Blue
color3 = [0 0 0]/255;       % Black
axisColor = [159 188 191]/255;
figColor = [211 226 226]/255;

if nargin==0 || strcmp(state,'reset')

    Owner=gcs;
    NdB = 0.707;

    Fs = 8000;
    if evalin('base', 'exist(''audiosamplingrate'')')
        Fs = evalin('base', 'audiosamplingrate');
    end

    if ~exist('eq_state')
        eq_state.BandWidth1 = .3 ;
        eq_state.CenterFreq1 = .1;
        eq_state.peakgain1 = 5 ;
```

```
    eq_state.BandWidth2 =.3  ;
    eq_state.CenterFreq2 = .2;
    eq_state.peakgain2 =  3;

    eq_state.BandWidth3 =.3  ;
    eq_state.CenterFreq3 = .3;
    eq_state.peakgain3 = -5 ;
end

bandw1 = eq_state.BandWidth1;
centerfreq1 =eq_state.CenterFreq1;
pk1= eq_state.peakgain1;

bandw2= eq_state.BandWidth2;
centerfreq2 = eq_state.CenterFreq2;
pk2=eq_state.peakgain2;

bandw3 =eq_state.BandWidth3;
centerfreq3 = eq_state.CenterFreq3;
pk3=eq_state.peakgain3;

%Creación inicial de los filtros
%Filtro 1
[b1 a1] = peq(0,pk1,centerfreq1,bandw1,NdB);
[h1,w1] = freqz(b1,a1);
%Filtro 2
[b2 a2] = peq(0,pk2,centerfreq2,bandw2,NdB);
[h2,w2] = freqz(b2,a2);
%Filtro 3
[b3 a3] = peq(0,pk3,centerfreq3,bandw3,NdB);
[h3,w3] = freqz(b3,a3);

coeffsMatrix1 = [b1(1) b1(2) b1(3) a1(2) a1(3)]';
coeffsMatrix2 = [b2(1) b2(2) b2(3) a2(2) a2(3)]';
coeffsMatrix3 = [b3(1) b3(2) b3(3) a3(2) a3(3)]';

% Write initial Filter values as to workspace as MPT Objects after
% creating a variable first.
assignin('base', 'CoeffsMatrix1', coeffsMatrix1);
%evalin(  'base', 'CoeffsMatrix1 = mpt.Parameter;');
%evalin(  'base', 'CoeffsMatrix1.Value = coeffsMatrix1;');

assignin('base', 'CoeffsMatrix2', coeffsMatrix2);
%evalin(  'base', 'CoeffsMatrix2 = mpt.Parameter;');
%evalin(  'base', 'CoeffsMatrix2.Value = coeffsMatrix2;');

assignin('base', 'CoeffsMatrix3', coeffsMatrix3);
%evalin(  'base', 'CoeffsMatrix3 = mpt.Parameter;');
%evalin(  'base', 'CoeffsMatrix3.Value = coeffsMatrix3;');

%Convert back to dB
h1 = 20*log10(abs(h1));
h2 = 20*log10(abs(h2));
h3 = 20*log10(abs(h3));
```

```
    %% ------------------BAND ONE----------------------------------------
    %Creación de la GUI
    figureName = 'Ecualizador de tres bandas';
    % Si la Figura ya existe, sobreescribe en ella
    hFig = findall(0, 'type', 'figure', 'Name', figureName );
    if ~isempty(hFig)
        figure(hFig);
    end

    if strcmp(axisStyle,'log')
        line1 = semilogx(w1,h1,'tag','line1','linewidth',2); hold on;
    elseif strcmp(axisStyle, 'normal')
        line1 = plot(w1,h1,'tag','line1','linewidth',2); hold on;
    end;

    set(line1,'color',color1);
    %Ejes off;
    grid on;
    xlim(([0 1])*pi);
    ylim([-15 15]);

    %Customize the looks of the Figure Window
    set(gcf,'menubar','none');
    set(gcf,'Name',figureName);

    % and the axis
    xlabels=[.1:.1:3.0];   % setup 1 2 5 sequence for xlabels
    set(gca,'xtick',xlabels*pi);

    for k=1:length(xlabels)
        xstring{k} = sprintf('%3.0f',(xlabels(k)*Fs/2));
    end;
    set(gca,'xticklabel',xstring)

    set(gcf,'color',figColor);
    set(gca,'color',axisColor);
    ScreenSize = get(0,'screensize');
    if ( exist('state') && ~strcmp(state,'reset')) || (nargin==0)
        set(gcf,'position',[min(ScreenSize(3)*2/3,507) min(ScreenSize(4)/2,417) 516
350]);
    end
    set(gca,'position',[0.1000    0.2000    0.8150    0.6150]);
    set(gcf,'resize','off');
    set(gca,'xcolor','k','ycolor','k');

    set(get(gca,'xlabel'), 'String', 'Frecuencia en Hz', 'color','k', 'FontSize',
10);
    set(get(gca,'ylabel'), 'String', 'Ganancia en [dB]', 'color','k', 'FontSize',
10);

    %Create Help Text at the top of the Figure Window
    htext = uicontrol(gcf);
```

```
    set(htext,'position',[50 310 425 25], 'BackgroundColor', figColor,
'fontangle','oblique','fontweight','bold', ...
        'FontSize', 10, 'tag', 'text1', 'string', 'Selecciona y arrastra sobre los
marcadores redondos o las líneas');

    %Create Text box for Parameter Value display
    freqText  = text(0.0057,13.65, 'Frecuencia central:', 'color', axisColor, 'tag',
'freqText', 'FontSize', 10);
    peakText  = text(1.30,13.65, 'Valor máximo:', 'color', axisColor, 'tag',
'peakText', 'FontSize', 10);
    bandwText = text(2.30,13.65, 'Ancho de banda:', 'color', axisColor, 'tag',
'bandwText', 'FontSize', 10);

    %Create the Note Text
    noteText = uicontrol(gcf);
    set(noteText,'position',[75 5 380 20], 'BackgroundColor', figColor, ...
        'string', 'La respuesta completa (linea amarilla) está centrada en 0 [dB].',
'FontSize', 10);

    %Create a Reset Button
    reset = uicontrol(gcf);
    set(reset,'position',[460 5 50 15],'string','Reset');
    set(reset,'callback','GUI_ecualizador_audio(''reset'')');
    %Add Markers that will be used for manipulating the parameters
    if pk1>0
       [h1max,indexh1] = max(h1);
   else
       [h1max,indexh1] = min(h1);
   end

    w1max = w1(indexh1);
    % the marker is plotted as a line object
    gain1 = line(w1max,h1max,'Marker','o','MarkerEdgeColor','k', ...

'MarkerFaceColor',color1,'tag','peakgain1','LineStyle','none','markersize',6);
    %-------------------END CODE FOR BAND ONE-------------------------------------

%% -------------------BAND TWO-------------------------------------------------

    %Create the plot
    %line2 = plot(w2*fs/(2*pi),h2,'b','tag','line2');
    %line2 = semilogx(w2,h2,'tag','line2');
    if strcmp(axisStyle,'log')
        line2 = semilogx(w2,h2,'tag','line2','linewidth',2);
    elseif strcmp(axisStyle, 'normal')
        line2 = plot(w2,h2,'tag','line2','linewidth',2);
    end;
    set(line2,'color',color2);

    %Add Markers that will be used for manipulating the parameters
    if pk2>0
        [h2max,indexh2] = max(h2);    % this does not work, it can be a max or a min!
    else
        [h2max,indexh2] = min(h2);    % this does not work, it can be a max or a min!
```

```
    end;

    w2max = w2(indexh2);
    % the marker is plotted as a line object
    gain2 = line(w2max,h2max,'Marker','o','MarkerEdgeColor','k', ...

'MarkerFaceColor',color2,'tag','peakgain2','LineStyle','none','markersize',6);
    %------------------END CODE FOR BAND TWO-------------------------------------
%% -------------------------BAND THREE-----------------------------------------
    %Create the plot
    %line3 = plot(w3*fs/(2*pi),h3,'m','tag','line3');
    %line3 = semilogx(w3,h3,'tag','line3');
    if strcmp(axisStyle,'log')
        line3 = semilogx(w3,h3,'tag','line3','linewidth',2);
    elseif strcmp(axisStyle, 'normal')
        line3 = plot(w3,h3,'tag','line3','linewidth',2);
    end;
    set(line3,'color',color3);

    %Add Markers that will be used for manipulating the parameters
    if pk3>0
       [h3max,indexh3] = max(h3);
    else
       [h3max,indexh3] = min(h3);
    end
    w3max = w3(indexh3);
    % the marker is plotted as a line object
    gain3 = line(w3max,h3max,'Marker','o','MarkerEdgeColor','k', ...

'MarkerFaceColor',color3,'tag','peakgain3','LineStyle','none','markersize',6);
    %------------------END CODE FOR BAND THREE-----------------------------------
%% Sum of all Bands
    sumH = h1+h2+h3;
    %Normalize the sum of responses to value bet -15 & +15
    normSumH = sumNorm(sumH);
    %Plot sum of all responses
    line4 = plot(w2,normSumH,'y--','tag','line4','linewidth',1,'hittest','off');
    hold off;

    %Store sum of all response in GCF to display sum of responses
    allH = normSumH;
    setappdata(gcf,'allH', allH);

    %Also save the equalizer responses and current parameters in a
    %structure: eqData.H ,eqData.centerFreq, eqData.bandW
    eqData.H = [h1 h2 h3];
    eqData.centerFreq = [centerfreq1 centerfreq2 centerfreq3];
    eqData.bandW = [bandw1 bandw2 bandw3];
    setappdata(gcf,'eqData', eqData);

    if nargin~=0
        try
            warning('off', 'Simulink:Engine:LineWithoutDst');
```

```
            warning('off', 'Simulink:Engine:LineWithoutSrc');
            set_param(bdroot,'SimulationCommand','update');
            warning('on', 'Simulink:Engine:LineWithoutDst');
            warning('on', 'Simulink:Engine:LineWithoutSrc');
        end
    end

    %Set the button down function
    set(gcf,'WindowButtonDownFcn','GUI_ecualizador_audio(''down'')');

    %Set the mouse move while button down function
    set(gcf,'WindowButtonMotionFcn','','WindowButtonUpFcn','');

    % Execute the WindowButtonDownFcn
elseif strcmp(state,'down')

    %Get the current complete info on filter responses
    EqData = getappdata(gcf,'eqData');
    cfreq = EqData.centerFreq;

    %Identify the Band that was clicked
    htype = get(gco,'type');

    %If Line is clicked, then set the Point Down information in the Figure
    if strcmp(htype,'line')
        tag = get(gco,'tag');
        tagIndex = eval(tag(end));

        set(gcf,'WindowButtonMotionFcn','GUI_ecualizador_audio(''move'')', ...
            'WindowButtonUpFcn','GUI_ecualizador_audio(''up'')');

        cp = get(gca,'CurrentPoint');
        xDown = cp(1,1);
        yDown = cp(1,2);

        setappdata(gcf,'pointDown',[xDown cfreq(tagIndex)]);

        text1 = findobj(gcf,'tag','text1');
        line4 = findobj(gcf,'tag','line4');
        if strcmp(tag,'peakgain1')|strcmp(tag,'peakgain2')|strcmp(tag,'peakgain3')
            set(text1,'string','Arrástralo para mover el valor máximo');
        elseif strcmp(tag,'line1')|strcmp(tag,'line2')|strcmp(tag,'line3')
            set(text1,'string','Arrastralo para cambiar el ancho de banda');
        elseif strcmp(tag,'line4')
            set(text1,'string','Esta es la respuesta actual. Cambiála al mover las
bandas individualmente.');
        end
    end

% Execute the WindowButtonMotionFcn
elseif strcmp(state,'move')

    %Get the current complete info on filter responses
    EqData = getappdata(gcf,'eqData');
```

```
    cfreq = EqData.centerFreq;
    H = EqData.H;
    bandw = EqData.bandW;

    %Find handles of blocks in Simulink Model whose value is set here
    text1 = findobj(gcf,'tag','text1');

    line1 = findobj(gcf,'tag','line1');
    gain1 = findobj(gcf,'tag','peakgain1');

    line2 = findobj(gcf,'tag','line2');
    gain2 = findobj(gcf,'tag','peakgain2');

    line3 = findobj(gcf,'tag','line3');
    gain3 = findobj(gcf,'tag','peakgain3');

    line4 = findobj(gcf,'tag','line4');

    freqText = findobj(gcf,'tag','freqText');
    peakText = findobj(gcf,'tag','peakText');
    bandwText = findobj(gcf,'tag','bandwText');

    cp = get(gca,'CurrentPoint');

    %-----------------------------------
    %If the user drags the point out of the axis reset
    %the corresponding point to the axis limits
    x = cp(1,1); xlims = get(gca,'xlim');
    %if x<xlims(1), x = xlims(1);end;
    if x<(45*2*pi/Fs), x = (45*2*pi/Fs) ;end; % Center frequency cannot be lower than
45 Hz
    if x>xlims(2), x = xlims(2);end;

    y = cp(1,2); ylims = get(gca,'ylim');
    if y<ylims(1), y = ylims(1);end;
    if y>ylims(2), y = ylims(2);end;
    %-----------------------------------

    tag = get(gco,'tag');
    if isempty(tag)
        return
    end
    tagIndex = eval(tag(end));

    switch tag(end)
        case '1'
            col = color1;
        case '2'
            col = color2;
        case '3'
            col = color3;
    end
    %Get the original ButtonDown x,y coordinates
```

```
    xyDown = getappdata(gcf,'pointDown');
    allH = getappdata(gcf,'allH');

    %IF THE GAIN POINT IS MOVED ==>
    if strcmp(tag,'peakgain1')||strcmp(tag,'peakgain2')||strcmp(tag,'peakgain3')
        set(text1,'string','Cambio de la frecuencia central y ganancia máxima de la
banda de frecuencia');
        myH = H(:,tagIndex);

        %Change the Shape and color of Marker while it's moving
        set(eval(['gain'
tag(end)]),'xdata',x,'ydata',y,'marker','diamond','markersize',12,'markerfacecolor','
y');
        [newb, newa] = peq(0,y,x,bandw(tagIndex),NdB);
        [newh,neww] = freqz(newb,newa);
        newh = 20*log10(abs(newh));
        [watever maxW] = max(newh);

        set(eval(['line' tag(end)]),'ydata',newh,'xdata',neww,'linewidth',2);
        H(:,tagIndex) = newh;

        %Update the sum response plot to improve readability
        allH = sumNorm(sum(H,2));
        set(line4,'ydata',allH,'xdata',neww);

        %Set the new parameters for the modified band
        EqData.H = H;
        cfreq(tagIndex) = x;
        EqData.centerFreq = cfreq;
        EqData.bandW = bandw;
        setappdata(gcf,'eqData',EqData);
        setappdata(gcf,'allH',allH);

        %Display the parameters as the user moves the UIobjects
        numFreq = numFormat(x,Fs);
        numBandw = numFormat(bandw(tagIndex),Fs);
        peakVal = sprintf('%0.1f',y);
        set(freqText,'string',['Frecuencia central: ',numFreq],'color',col);
        set(peakText,'string',['Valor máximo: ',peakVal,'dB'],'color',col);
        set(bandwText,'string',['Ancho de banda: ',numBandw],'color',col);

        drawnow

    %IF THE LINE (Bandwidth) IS MOVED ==>
    elseif strcmp(tag,'line1')||strcmp(tag,'line2')||strcmp(tag,'line3')
        set(text1,'string','Al arrastrar las líneas de color se modifica en ancho de
banda de la banda de frecuencia');
        %Get the response of the current band
        myH = H(:,tagIndex);
        %Change the Shape and color of line while it's moving
        set(eval(['line' tag(end)]),'color','c','linewidth',3);

        %Get the value of the peak gain
        peakG = get(eval(['gain' tag(end)]),'ydata');
```

```
        bwDiff = (x-xyDown(2));
        bandww = 2*abs(bwDiff);
        if bandww<50*2*pi/Fs
            bandww = 50*2*pi/Fs; %Minimum Bandwidth allowed is 50 hertz
        elseif bandww>(20000*2*pi/Fs) %Maximum Bandwidth allowed is 20000 hertz
            bandww = 20000*2*pi/Fs;
        end
        bandw(tagIndex) = bandww;

        %Center Frequency cfreq does not change

        %Compute New Filter parameters
        [newb, newa] = peq(0,peakG,cfreq(tagIndex),bandww,NdB);
        [newh,neww] = freqz(newb,newa);
        newh = 20*log10(abs(newh));

        %Update Line for the chosen band
        set(eval(['line' tag(end)]),'ydata',newh,'xdata',neww);
        set(eval(['gain' tag(end)]),'Marker','o','MarkerEdgeColor','k',...
             'MarkerFaceColor',col,'LineStyle','none','markersize',6);

        H(:,tagIndex) = newh;
        EqData.H = H;
        %EqData.centerFreq = cfreq;
        EqData.bandW = bandw;
        setappdata(gcf,'eqData',EqData);

        allH = sumNorm(sum(H,2));
        setappdata(gcf,'allH',allH);
        set(line4,'ydata',allH,'xdata',neww);

        peakVal = sprintf('%.1f',peakG);
        numFreq = numFormat(cfreq(tagIndex),Fs);
        numBandw = numFormat(bandw(tagIndex),Fs);

        set(freqText,'string',['Frecuencia central: ',numFreq],'color',col);
        set(peakText,'string',['Valor máximo: ',peakVal,' dB'],'color',col);
        set(bandwText,'string',['Ancho de banda: ',numBandw],'color',col);

        drawnow
    end;

% Execute the WindowButtonUpFcn
elseif strcmp(state,'up')

    tag = get(gco,'Tag');
    if isempty(tag)
        return
    end
    tagIndex = eval(tag(end));

    switch tag(end)
```

```
        case '1', col = color1;
        case '2', col = color2;
        case '3', col = color3;
    end

    eqData = getappdata(gcf,'eqData');
    H = eqData.H;
    myH = H(:,tagIndex);
    hmax = max(myH);
    hmin = min(myH);

    if abs(hmax) < 0.00001
        hmax = hmin;
    end

    cf = eqData.centerFreq;
    cfreq = cf(tagIndex);
    bw = eqData.bandW;
    bandw = bw(tagIndex);

    text1 = findobj(gcf,'tag','text1');
    gain1 = findobj(gcf,'tag','peakgain1');
    line1 = findobj(gcf,'tag','line1');

    gain2 = findobj(gcf,'tag','peakgain2');
    line2 = findobj(gcf,'tag','line2');

    gain3 = findobj(gcf,'tag','peakgain3');
    line3 = findobj(gcf,'tag','line3');

    % Call the PEQ function to get the filter coefficients
    [b a] = peq(0,hmax,cfreq,bandw,0.707);

%     % Overwrite the initial Filter and Scale values of the appropriate band
%     [newSos, newg]= tf2sos(num,den);

    disp(['Se han modificado los coeficientes del filtro para la banda: ',tag(end)]);
    varName = ['CoeffsMatrix',tag(end)];
    %newCoeffMatrix = evalin('base',varName);
    newCoeffMatrix = [b(1) b(2) b(3) a(2) a(3)]';
    assignin('base', varName, newCoeffMatrix);
   % evalin('base',['CoeffsMatrix',tag(end),'.Value = ',varName,';']);

    % Update Model with new parameters.
    try
        warning('off', 'Simulink:Engine:LineWithoutDst');
        warning('off', 'Simulink:Engine:LineWithoutSrc');
        set_param(bdroot,'SimulationCommand','update');
        warning('on', 'Simulink:Engine:LineWithoutDst');
        warning('on', 'Simulink:Engine:LineWithoutSrc');
    end

    set(text1,'string','Selecciona y arrastra en los marcadores o sobre la línea');
    set(eval(['gain' tag(end)]),'Marker','o','MarkerEdgeColor','k',...
```

```
            'MarkerFaceColor',col,'LineStyle','none','markersize',6);
    set(eval(['line' tag(end)]),'color',col,'linewidth',2);

    set(gcf,'WindowButtonMotionFcn','');
    set(gcf,'WindowButtonUpFcn','');

elseif strcmp(state,'save_state')
    %eq_state.peakgain1 =
    %str2num(get_param([Owner,'/PeakGain1'],'value'));
    %eq_state.CenterFreq1= str2num(get_param([Owner,'/CenterFreq1'],'value'));
    %eq_state.BandWidth1=  str2num(get_param([Owner,'/BandWidth1'],'value'));

    %eq_state.peakgain2   =  str2num(get_param([Owner,'/PeakGain2'],'value'));
    %eq_state.CenterFreq2 =  str2num(get_param([Owner,'/CenterFreq2'],'value'));
    %eq_state.BandWidth2  =  str2num(get_param([Owner,'/BandWidth2'],'value'));

    %eq_state.peakgain3   =  str2num(get_param([Owner,'/PeakGain3'],'value'));
    %eq_state.CenterFreq3 =  str2num(get_param([Owner,'/CenterFreq3'],'value'));
    %eq_state.BandWidth3  =  str2num(get_param([Owner,'/BandWidth3'],'value'));

    %eq_state.AF_Gain   =    str2num(get_param([Owner,'/AF_Gain'],'gain'));

   %save EQ_State  eq_state


end

function out = numFormat(x,Fs)
%This function is for formatting the display of the frequency values
x = x*Fs/(2*pi); %Convert from radians/sec to absolute frequency in Hz
if (x/1000)<1
    out = [num2str(round(x)),'Hz'];
else
    s = sprintf('%0.3g',x/1000);
    out = [s,'kHz'];
end

function normSumH = sumNorm(x)
% Look for the center of the freq response range, and force this to be 0 dB.
normSumH= x; % -(max(x)+min(x))/2;

%% PEQ - Calculate Parametric Equalizer Coefficient
function [b, a] = peq(Gref, G0, w0, BW, NdB)

GBW = G0*NdB;
f = fdesign.parameq(...
      ['N,' 'F0,' 'BW,' 'Gref,' 'G0,' 'GBW'],...
        2,   w0,   BW,   Gref,   G0,   GBW, 2*pi);
myfilt = design(f);
[b,a]= tf(myfilt);
```

3.4.2 Parametric EQ with matching gain at Nyquist frequency

```
% peq.m - Parametric EQ with matching gain at Nyquist frequency
% Author: Sophocles J. Orfanidis,
% Usage:  [b, a] = peq(G0, G, w0, Dw, NdB)
%
% G0 = reference gain at DC in dB
% G  = boost/cut gain in dB
% GB = bandwidth gain in dB
%
% w0 = center frequency in Hz
% Dw = bandwidth in Hz
% NdB = amount of db less than the peak/notch gain as the bandwidth frequencies
% b  = [b0, b1, b2] = numerator coefficients
% a  = [1,  a1, a2] = denominator coefficients

% Modified by Arvind using the equations from IIR Filter Design Chapter in
% the book
function [b, a, beta] = peq(G0, G, w0, Dw, NdB)

%Convert from Decibels to real values
GB = G * NdB;
G  = (10^(G/20));
GB = (10^(GB/20));
G0 = (10^(G0/20));

%Convert absolute frequencies to rads/sec
%w0 = w0*2*pi/fs;
%Dw = Dw*2*pi/fs;

beta = sqrt((GB^2-G0^2)/(G^2-GB^2)) * tan(Dw/2);
b = [(G0+G*beta)/(1+beta) -(2*G0*cos(w0))/(1+beta)  (G0-G*beta)/(1+beta)];
a = [1 -(2*cos(w0))/(1+beta) (1-beta)/(1+beta)];

 %**********************
% double beta;
%
% beta = sqrt((GB[0]^2-G0[0]^2)/(G^2[0]-GB[0]^2)) * tan(Dw[0]/2)
% Num[0] = (G0[0]+G[0]*beta)/(1+beta);
% Num[1] =  -(2*G0[0]*cos(w0[0]))/(1+beta) ;
% Num[2] = (G0[0]-G[0]*beta)/(1+beta);
%
% Den[0] = 1;
% Den[1] = -(2*cos(w0[0]))/(1+beta);
% Den[2] = (1-beta)/(1+beta);
```

3.5. Results

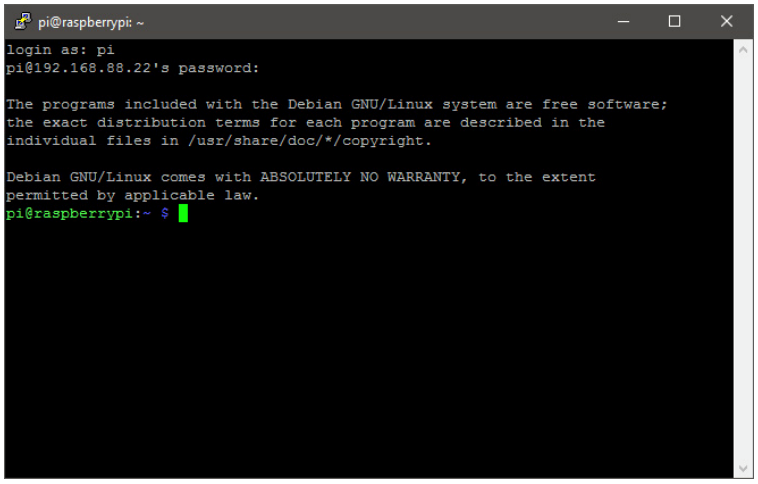


Figure 3.

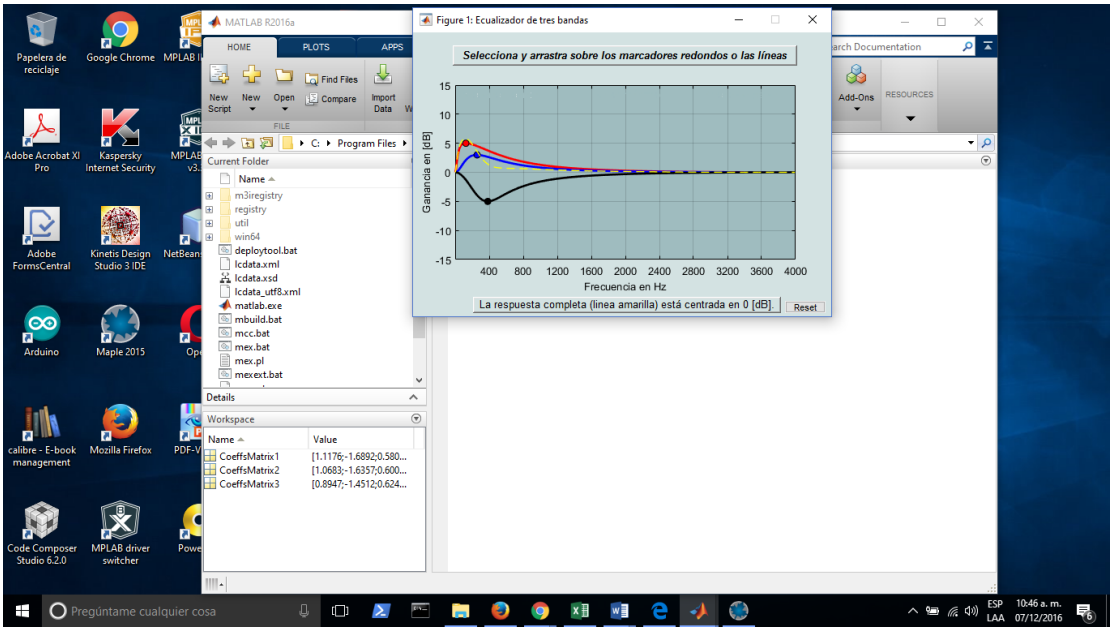


Figure 4.

3.6. Conclusions

3.6.1 Improvements

- Make a more general filter to process a wider range of signals.
- Increment the numbers of bands.
- Improve the algorithm to make a real time processing.
- Improve communication protocol.
- Increment the autonomy of the algorithm.
- Implement distributed control.

- Improve the graphic interface.

3.6.2 Success factors for remainder of project

- The most valuable aspect is the generation of source code from matlab tool, the allow to the engineer to focus on the algorithm without take care about development code, the advantage in this is to save time (engineering hours) and for a company it represents money.

Contents

3.7. References

- [NI-Response and Software Equalization] http://zone.ni.com/reference/en-XX/help/371025N-1/rfsg/if_response_and_equalizer/
- [Intersymbol Interference and Equalization] http://wireless.ece.ufl.edu/twong/Notes/Comm/ch4.pdf
- [FIR vs IIR filtering] https://www.minidsp.com/applications/dsp-basics/fir-vs-iir-filtering
- Reference: Rockwell
- Reference: Yokogawa
- Reference: classification of embedded systems
- Reference: Matlab 3-Band Parametric Equalizer

APÉNDICE B

Diseño de un amplificador para PCIe Gen II

- **1. Design differential amplifier show in figure 1, to achieve the define specifications from table 1. For the transistors of the differential pair consider L=0.24um, Kn and VTN parameters listed.**

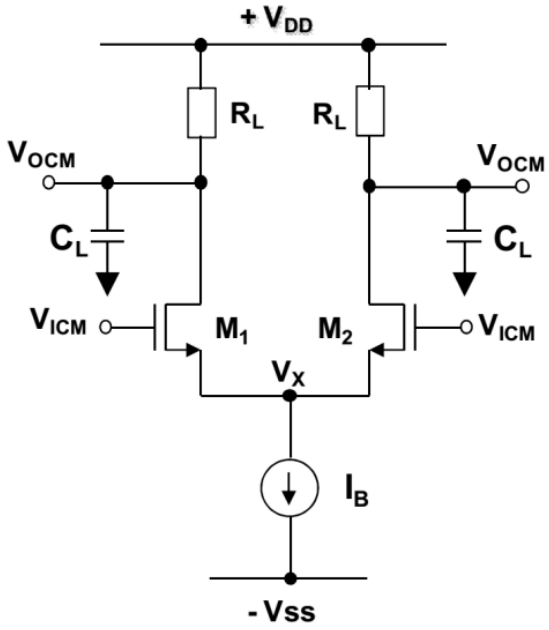


Figure 5.Differential amplifier.

V_{DD} = + 1.2 V	V_{icm} = 0.7× V_{DD}
V_{SS} = 1 V	V_{ocm} = 0.7× V_{DD}
A_v = 6 dB	BW = 5 GHz
$P_{Dmáx}$ <= 1 mW	C_L = 20 fF
K_n = 200 µA/V^2	V_{TH} = 0.35 V

Table 2

Once that specification is defined the first step is make corresponding theoretical calculous to accomplish with the design.

Step 1. Calculate I_{Bmax}.

$$P_{Dmáx} = V_{DD} I_{Bmáx} = 1\ \text{mW}$$

$$\Rightarrow I_{Bmáx} = \frac{P_{Dmáx}}{V_{DD}} = \frac{1\ \text{mW}}{1.2\ \text{V}} = 833\ \mu\text{A}$$

Step 2. Calculate R_L from bandwidth and capacitive specified.

$$BW = \frac{1}{R_L \cdot C_L}$$

$$\Longrightarrow R_L = \frac{1}{BW \cdot C_L} = \frac{1}{(2\pi \times 5 \times 10^9\ \text{Hz})(20 \times 10^{-15}\ \text{F})} = 1591.55\ \Omega$$

Step 3. Calculate V_{icm} and V_{ocm}.

$$Vicm = Vocm = 0.7 \times V_{DD} = 0.84\ \text{V}$$

Step 4. Calculate I_B from V_{ocm} and R_L.

$$\frac{I_B}{2} = \frac{V_{DD} - V_{ocm}}{R_L}$$

$$\Rightarrow I_B = 2\left(\frac{1.2\ V - 0.84\ V}{1591.55\ \Omega}\right) = 452.4\ \mu A$$

Step 5. Calculate Gm from voltage gain Av.

$$A_v \cong g_m R_L$$

$$\Rightarrow g_m = \frac{A_v}{R_L} = \frac{10^{\frac{6\ dB}{20}}}{1591.55\ \Omega} = \frac{2}{1591.55\ \Omega} = 1.254\ mS$$

Step 6. Calculate $\left({}^{W}/_{L}\right)$ and W y L from the data obtained in previous steps.

$$g_m = \sqrt{2K_n I_D\left(\frac{W}{L}\right)} = \sqrt{2K_n \frac{I_B}{2}\left(\frac{W}{L}\right)} = \sqrt{K_n I_B\left(\frac{W}{L}\right)} \Rightarrow g_m^2 = k_n I_B\left(\frac{W}{L}\right)$$

$$\Rightarrow \left(\frac{W}{L}\right) = \frac{g_m^2}{k_n \cdot I_B}$$

From the last equation is necessary to get Kn parameter, for this reason parametric analysis was performed to get the operating point.

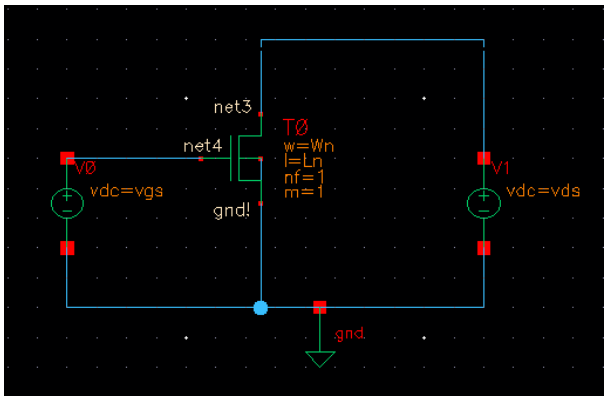


Figure 6. Characterization circuit to get operating point.

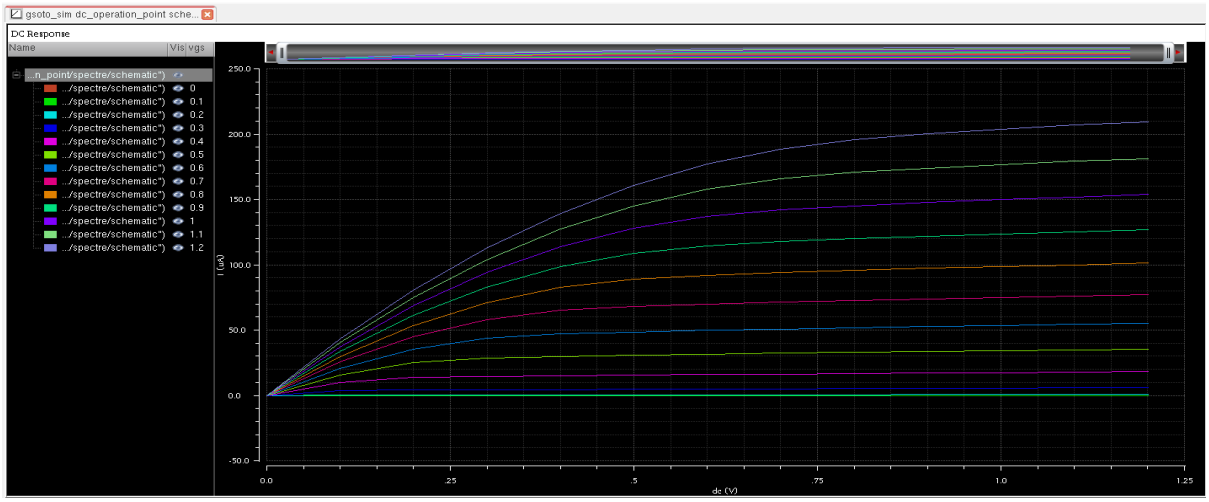


Figure 7. DC response - Multiparametric analysis waveforms Ib vs Vds.

Parameter	Value
betaeff	511.5u
cbb	386.9a
cbd	-170e888
cbdbo	-170e888
cbg	-2.248f
cbgbo	-2.229f
cbs	2.031f
cbsbo	2.031f
cdb	-977.8a
cdd	906.2a
cddbo	382.5a
cdg	-6.195f
cdgbo	-5.671f
cds	6.266f
cdsbo	6.266f
cgb	2.028f
cgd	-846.7a
cgdbo	-323.1a
cgg	17.13f
cggbo	16.06f
cgs	-18.31f
cgsbo	-17.78f
cjd	1.346f
cjs	1.665f

Parameter	Value
Covlgb	19.38a
Covlgd	523.6a
Covlgs	524.6a
Csb	-1.437f
Csd	110.5a
Csg	-8.683f
Css	10.01f
Gbd	1.00E+122
Gbs	1.00E+122
Gds	27.79u
Gm	229.4u
Gmbs	37.11u
gmoverid	2.274
Ibulk	-644.1z
Id	100.9u
Ids	100.9u
Igb	756.3y
Igbacc	756.3y
Igbinv	756.3y
Igcd	1.511p
Igcs	2.38E+122
Igd	5.845f
Igdt	4.017p
Igidl	0
Igisl	0

Parameter	Value
igs	120.2f
pwr	59.81u
qb	-3.288f
qbi	-3.288f
qd	-3.936f
qdi	-3.936f
qg	13.31f
qgdovl	128e888
qgi	13.31f
qgsovl	438.9a
qjd	-886.7a
qjs	-5.967a
qs	-6.086f
qsi	-6.086f
rdeff	35.49
region	2
reversed	0
rgbd	0
ron	5.876K
rseff	35.49
vbs	0
vdb	600m
vds	600m
vdsat	545.5m
vgb	840m
vgd	240m

vgs	840m
vth	157.1m

$$V_{BS} = 0\ V \Longrightarrow V_{Th} = 157.1\ mV$$

Using this result is possible calculate Kn

$$g_m = \kappa_n \frac{W}{L} V_{DSat}$$

$$\Longrightarrow \kappa_n = \frac{g_m}{\frac{W}{L} V_{DSat}}$$

$$\kappa_n = \frac{229.4\ \mu \frac{A}{V}}{\frac{1.6}{1.2}(545.5\ mV)} = 315.398\ \mu \frac{A}{V^2}$$

$$\Rightarrow \left(\frac{W}{L}\right) = \frac{g_m^2}{k_n \cdot I_B} = \frac{(1.254\ mS)^2}{\left(315.398\ \frac{\mu A}{V^2}\right)(452.39\ \mu A)} \cong 11$$

$L = 0.24\ \mu m$:

$$W = 11 \times L = 11 \times 0.24\ \mu m$$

$$W = 2.64\ \mu m$$

Table 2 summarize the theoretical results calculated from the specification.

Parameter	Value
I_{Bmax}	833 µA
R_L	1591.55 Ω
V_{icm}	0.84 V
V_{ocm}	0.84 V
I_B	452.39 µA
G_m	1.254 mS
K_n	$315.398\ \mu \frac{A}{V^2}$
W	2.64 µm

Table 3. Calculated values.

• 2. – Verify the design in CADENCE using 130nm technology

A) Frequency response of the amplifier accomplish with the design specification.

1. To simulate the differentia amplifier the first step is implement schematic in virtuoso tool, figure 4.

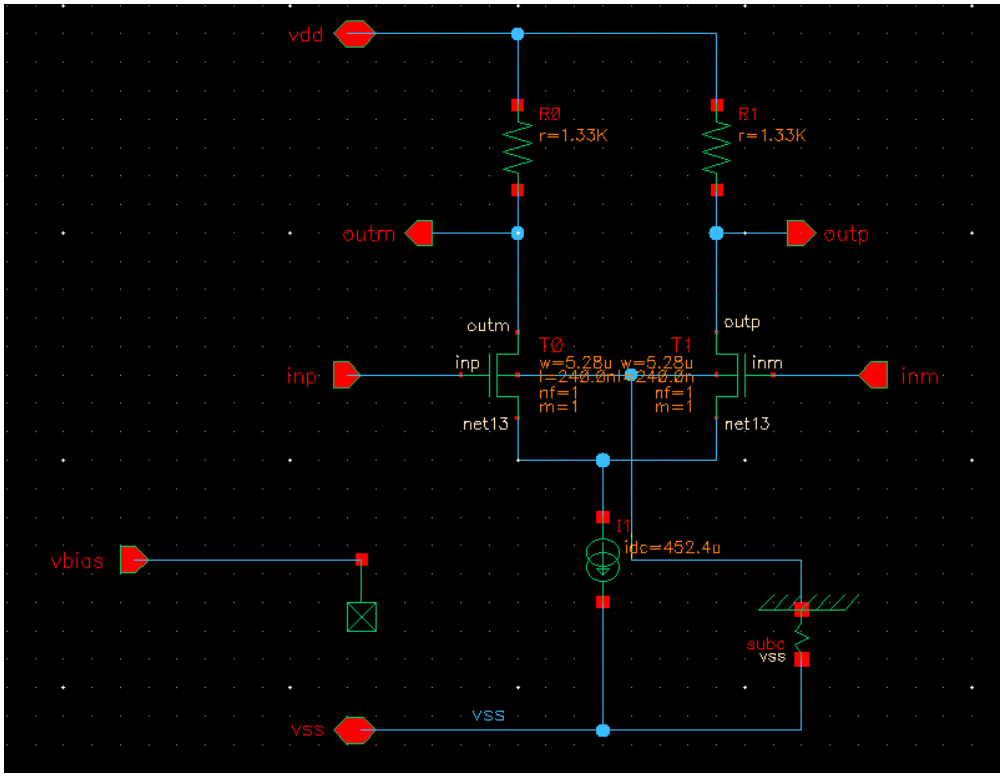


Figure 8. Schematic implemented to simulate differential amplifier.

2. Test bench implementation, figure 5.

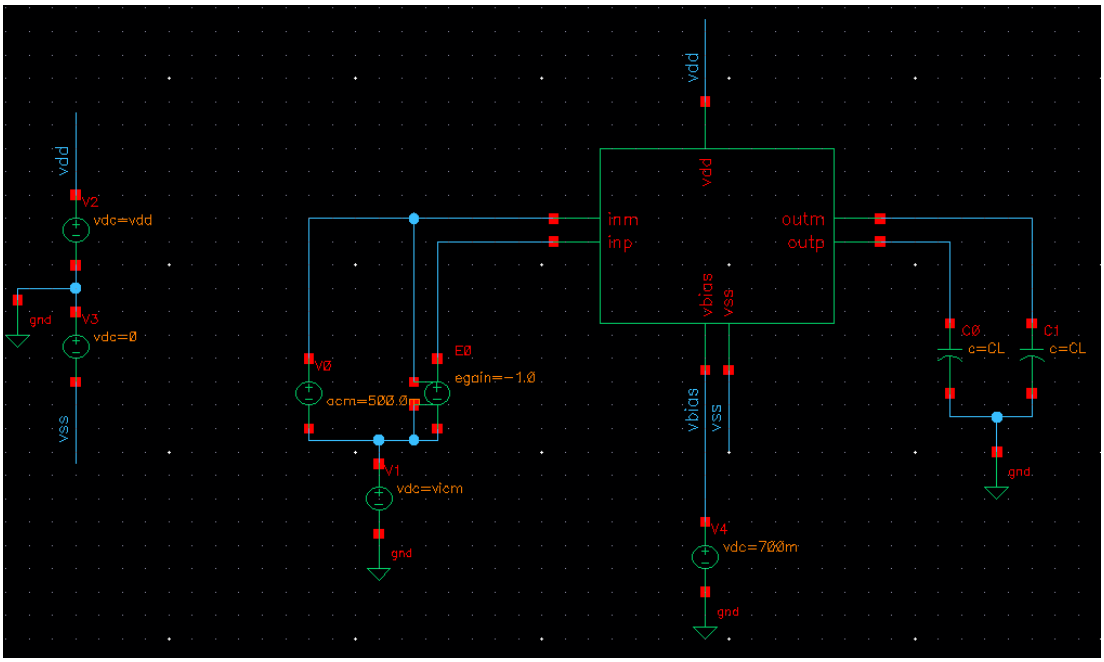


Figure 9. Test bench differential amplifier.

3. Frequency response: AC analysis with frequency sweep.

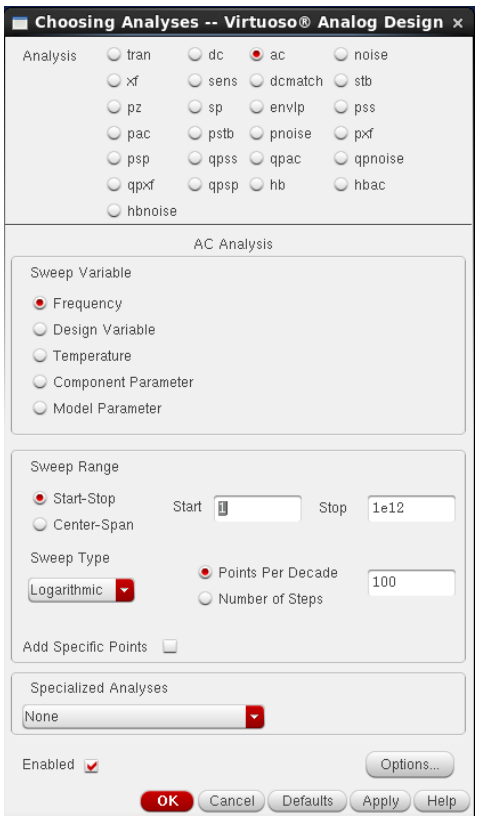


Figure 10. AC analysis configuration.

Figure 7 show the bode diagram frequency vs voltage gain as a frequency analysis result.

Figure 11. Bode diagram freq. vs Av.

b) Operating point analysis results.

Parameter	Value
Betaeff	511.5u
Cbb	386.9a
Cbd	-170e888
Cbdbo	-170e888
Cbg	-2.248f
Cbgbo	-2.229f
Cbs	2.031f
Cbsbo	2.031f
Cdb	-977.8a
Cdd	906.2a
Cddbo	382.5a
Cdg	-6.195f
Cdgbo	-5.671f
Cds	6.266f
Cdsbo	6.266f
Cgb	2.028f
Cgd	-846.7a
Cgdbo	-323.1a
Cgg	17.13f
Cggbo	16.06f
Cgs	-18.31f
Cgsbo	-17.78f
Cjd	1.346f
Cjs	1.665f

Parameter	Value
Covlgb	19.38a
Covlgd	523.6a
Covlgs	524.6a
Csb	-1.437f
Csd	110.5a
Csg	-8.683f
Css	10.01f
Gbd	1.00E+122
Gbs	1.00E+122
Gds	27.79u
Gm	229.4u
Gmbs	37.11u
gmoverid	2.274
Ibulk	-644.1z
Id	100.9u
Ids	100.9u
Igb	756.3y
Igbacc	756.3y
Igbinv	756.3y
Igcd	1.511p
Igcs	2.38E+122
Igd	5.845f
Igdt	4.017p
Igidl	0
Igisl	0

Parameter	Value
igs	120.2f
pwr	59.81u
qb	-3.288f
qbi	-3.288f
qd	-3.936f
qdi	-3.936f
qg	13.31f
qgdovl	128e888
qgi	13.31f
qgsovl	438.9a
qjd	-886.7a
qjs	-5.967a
qs	-6.086f
qsi	-6.086f
rdeff	35.49
region	2
reversed	0
rgbd	0
ron	5.876K
rseff	35.49
vbs	0
vdb	600m
vds	600m
vdsat	545.5m
vgb	840m
vgd	240m
vgs	840m
vth	157.1m

C) Comparative table showing the final dimensions of the amplifier vs theoretical values.

From the BW formula is possible identify what need to change to achieve the corresponding BW value, C_L is under specification for that is immovable, then R_L is the only parameter that can be modifying.

$$\mathrm{BW} = \frac{1}{\mathrm{R_L} \cdot \mathrm{C_L}}$$

Like BW case from A_v formula is possible identify W like the parameter that can be changing to achieve the A_v expected value.

$$\mathrm{A_v} \cong \mathrm{g_m R_L} = \sqrt{\mathrm{K_n I_B}\left(\frac{\mathrm{W}}{\mathrm{L}}\right)} \cdot \mathrm{R_L}$$

Next table show all values obtained from R_L and W variation, the first approximation it was to get A_v value, this was possible at the first iteration just duplicated the value from W. For BW was necessary other four iterations

Iteration	RL'	W'	GM'	Av'	Av'dB	BW (Rad) calculated	BW (Hz) calculated	Av sim (dB)	BW sim (GHz)
1	1591.5	0.00000264	0.001253	1.993835	5.993783	31416902293	5032179378	3.392	4.764
2	1591.5	0.00000528	0.001772	2.819708	9.004083	31416902293	5032179378	7.33	4.373
3	1432.35	0.00000528	0.001772	2.537737	8.088933	34907669215	5591310420	6.574	4.802
4	1273.2	0.00000528	0.001772	2.255766	7.065882	39271127867	6290224223	5.706	5.346
5	1300	0.00000528	0.001772	2.303249	7.246817	38461538462	6160548831	5.86	5.245
6	1330	0.00000528	0.001772	2.356401	7.444982	37593984962	6021589083	6.031	5.137

Iteration 2: First approximation duplicated W value with that A_v value was good enough, figure 8.

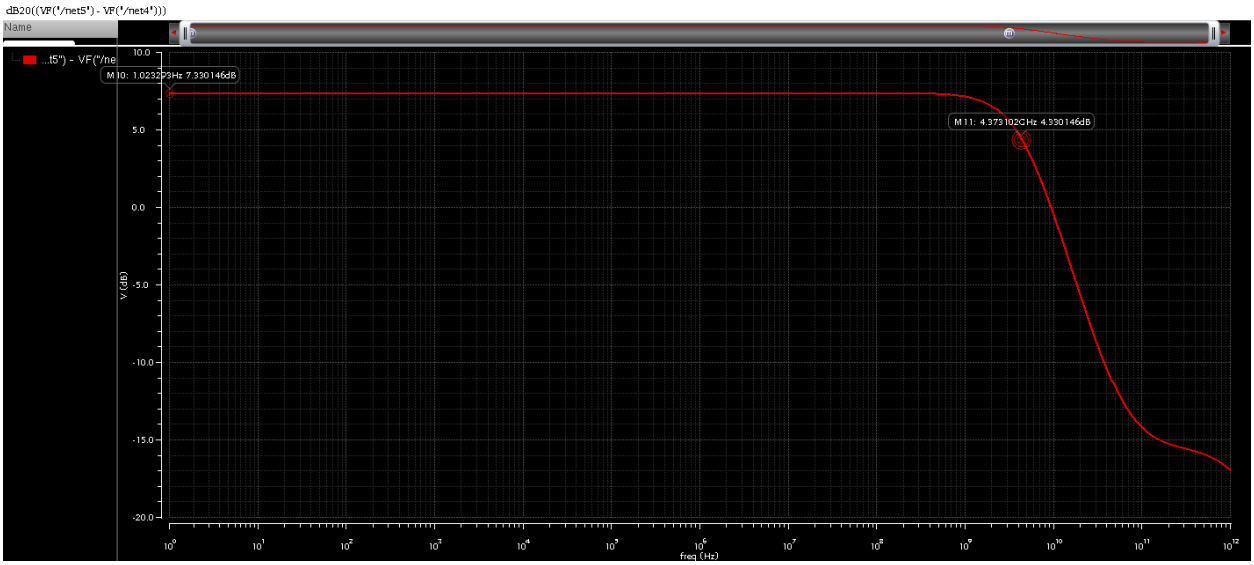


Figure 12.W=5.28um and Rl=1.591K ohms

Iteration 3: RL Decrement 10 %

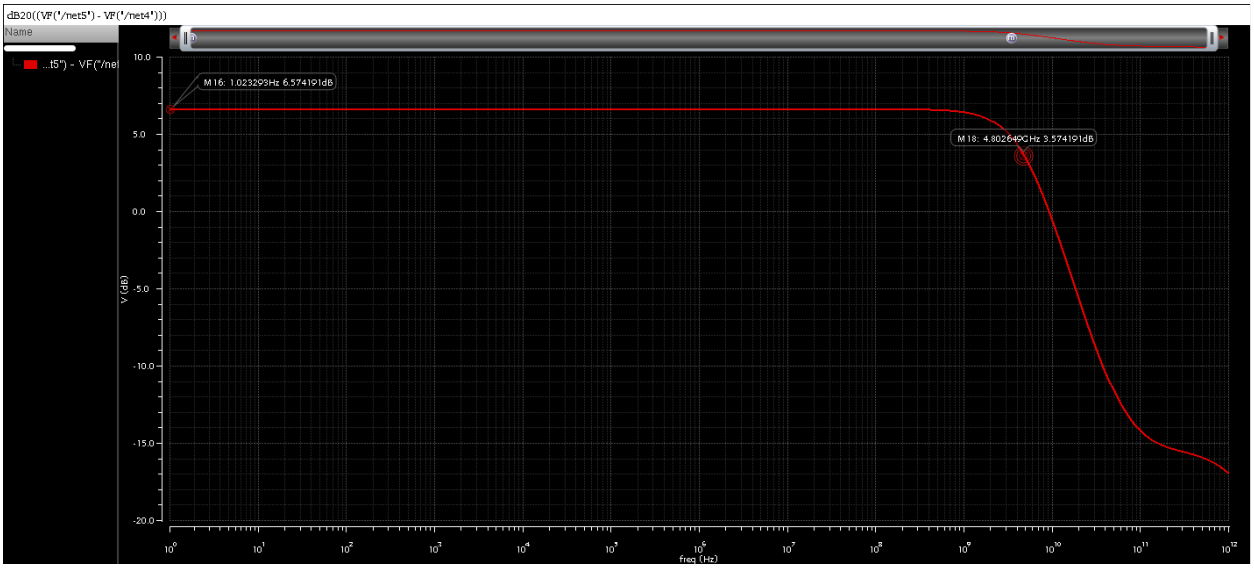


Figure 13. W=5.28um and Rl=1432.35 ohms.

Iteration 4: RL decrement 20 %

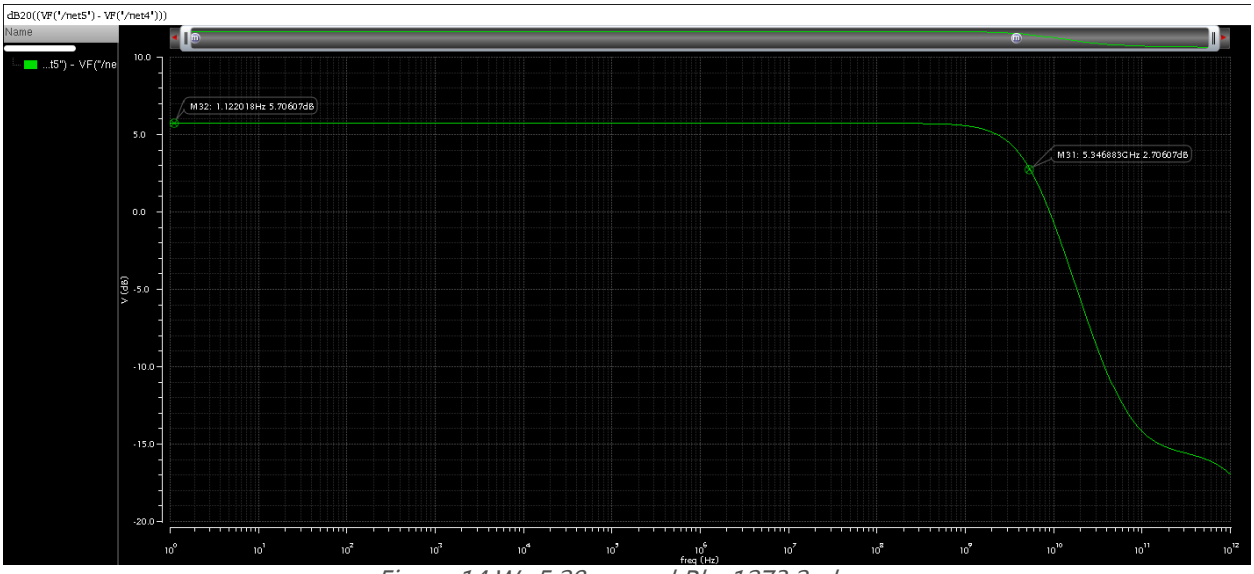


Figure 14.W=5.28um and Rl= 1273.2 ohms

Iteration 5: In the previous iteration the gain was less than expected, then 20% of decrement is too much, for that RL is readjusted to 18 %.

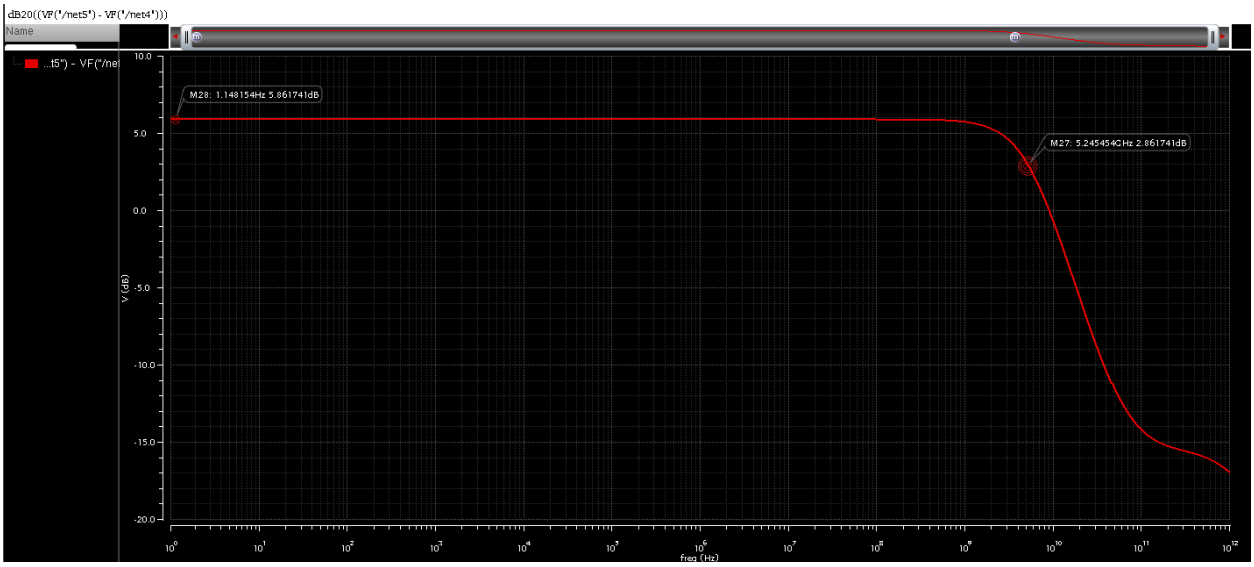


Figure 15. W=5.28um and Rl=1.3K ohms

Iteration 6: In the previous iteration the gain was less than expected, then 18% of decrement is too much, for that RL is readjusted to 16 %.

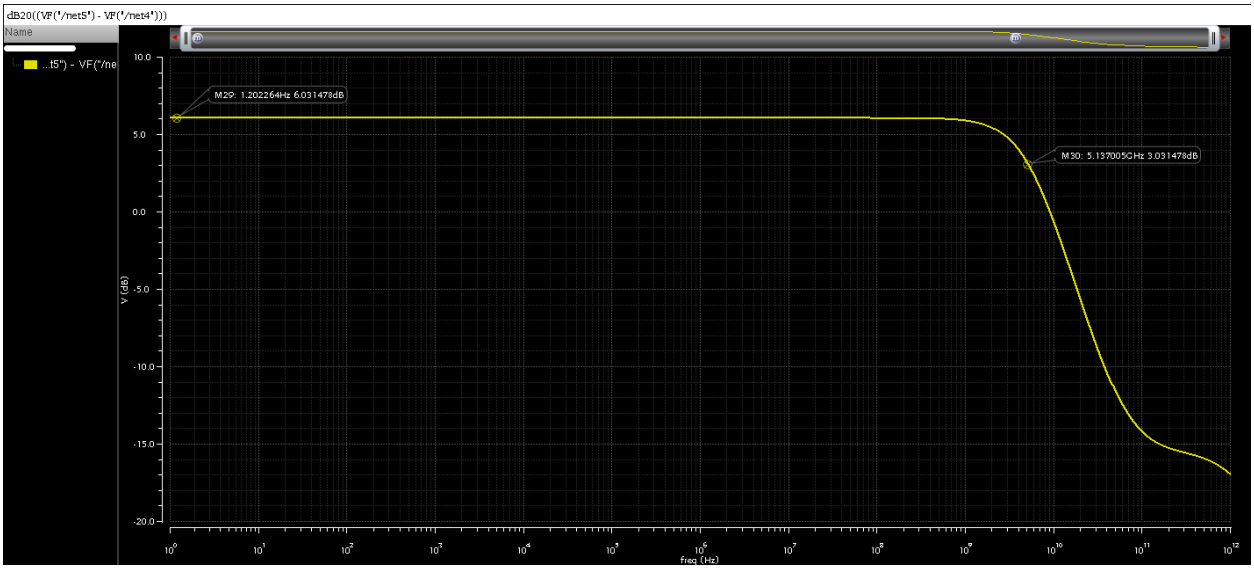


Figure 16.W=5.28um and Rl=1.3301K ohms

With W= 5.28 um and $R_L = 1.330K\Omega$ the expected behavior according to A_v and BW was achieved.

D) Conclusions

With all parameters theoretically calculated the frequency response was not like expected, the bandwidth was reduced to 4.76 GHz and the voltage gain less than 5 db.

- Av is around 30 % less than expected
- BW is around 20% less than expected

In most of the cases will be necessary fit design parameters to achieve the desired outputs, but the important thing is to know what need to be change based in the mathematical models, in this case R_L is link up with the BW and W with the gain.

All variations are proportions not absolute values according to one of the design principles. According to the obtained results the design parameter with more impact is R_L because it has impact not only in BW but also in A_v, this is due to the mathematical formulas that describe the model decrementing R_L to increase the BW at the same time the A_v is decrementing that is an expected behavior.

One of the most important things to take care is characterize correctly all the element used in the design, in this case was needed to extract its operating point of the NFET transistor to get the correct Kn value, without that the rest of the design can be wrong.

There is a difference between calculated values and simulation results, this is because simulation tool takes care of physical parameters that in normal calculous not.

This remark that all formulas are a simplistic approximation of the real behavior, that sounds some disappointing but not at all, thanks to this approximation is possible to speed up any design, in other way will be tedious and very slow to do all the math for any design. Ones that the first approximation is available is when the simulation tools are very useful to get a more realistic models and behaviors saving time and effort.

CONTENTS

APÉNDICE C

• Abstract

This project presents the design of band-stop filter for microwave applications; this design is implemented in micro-strips lines with center frequency 3.4 GHz. The filter has been simulated using ADS design software and implementation on RF4 substrate.

• Filter specifications

Center frequency	Fractional Bandwidth	Filter Order	Type of frequency response	Reference impedance
3.4 GHz	5%	5	0.1dB Chebyshev	50Ω

Microstrip ROGER RO4003

h	Cladding	er
0.81 mm	0.5 oz	3.55

• Design

First step is to get normalized filter with values corresponding to n=5, the next table show normalized values for Chebyshev filter with 0.1dB ripple, the bold letters values are selected to implement the normalized low pass filter.

Normalized Chebyshev element values, 0.1 dB ripple										
Order	**C1**	**L2**	**C3**	**L4**	**C5**	**L6**	**C7**	**L8**	**C9**	**R_{Load}**
2	0.8431	0.6220								0.7378
3	1.0316	1.1474	1.0316							1
4	1.1088	1.3062	1.7704	0.8181						0.7378
5	**1.1468**	**1.3712**	**1.9750**	**1.3712**	**1.1468**					**1**
6	1.1681	1.4040	2.0562	1.5171	1.9029	0.8618				0.7378

7	1.1812	1.4228	2.0967	1.5734	2.0967	1.4228	1.1812			1
8	1.1898	1.4346	2.1199	1.6010	2.1700	1.5641	1.9445	0.8778		0.7378
9	1.1957	1.4426	2.1346	1.6167	2.2054	1.6167	2.1346	1.4426	1.1957	1
	L1	**C2**	**L3**	**C4**	**L5**	**C6**	**L7**	**C8**	**L9**	**R_{Load}**

As a first approximation for low pass filter, the circuit is modeling using lumped elements, as "n" is odd the topology of this circuit beginning with a shunt element. Figure 1 show implementation of a low pass filter normalized on impedance with n=5.

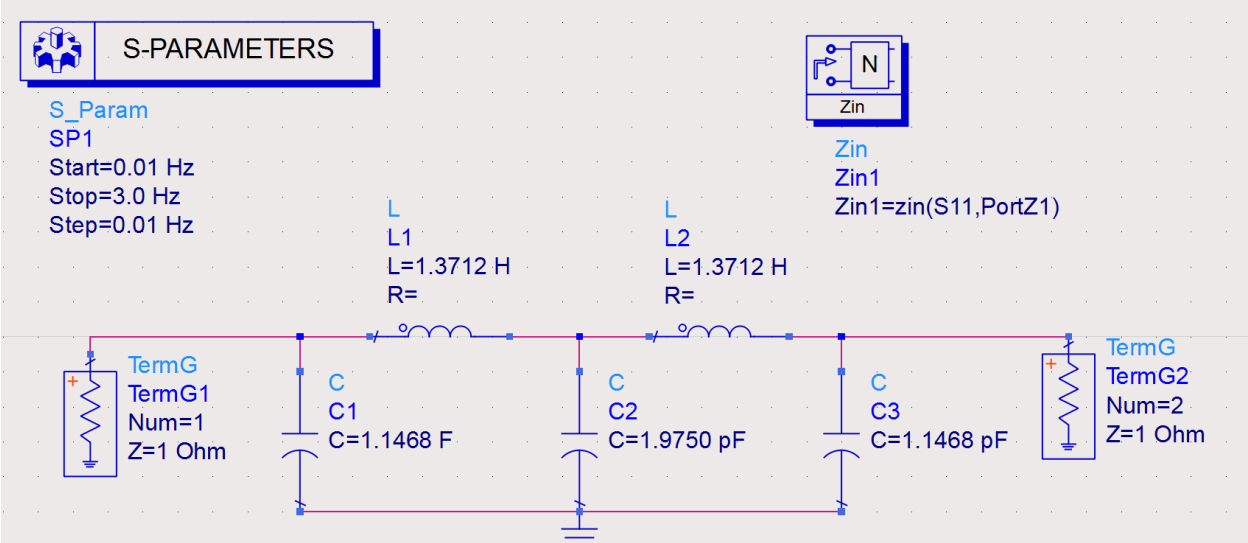


Figure 17.low pass filter normalized in impedance with n=5.

This circuit was simulated from 0.01 Hz to 3.0 Hz, the result is show in figure 2.

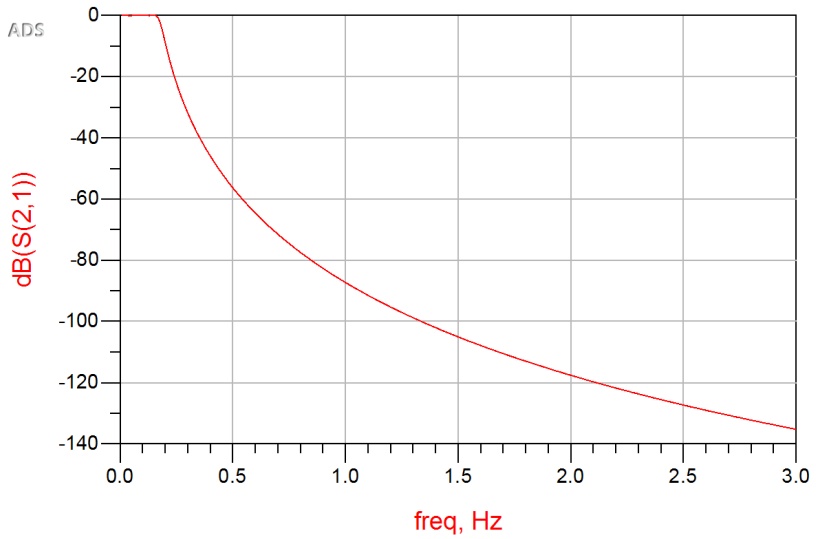


Figure 18. Low pass filter response with impedance normalization.

- Filter transformation

Low-pass filter prototypes of the previous section where normalized design gave a source impedance R=1 and w=1.

Next step is de-normalized each circuit element on impedance and frequency, applying next scaling formulas.

$$L_k = \frac{Z_0 L_k}{\omega_c} \qquad C_k = \frac{C_k}{Z_0 \omega_c}$$

Applying these formulas to corresponding circuit element is possible to get the new scaling values.

Circuit element	Value
C1	1.07E-12
L2	3.20931E-09
C3	1.85E-12
L4	3.21E-09
C5	1.07E-12

Figure 3 show the low pass filter with the elements after apply impedance and frequency scaling, can be observed that all values corresponding to a more realistic magnitude for a capacitors and inductors.

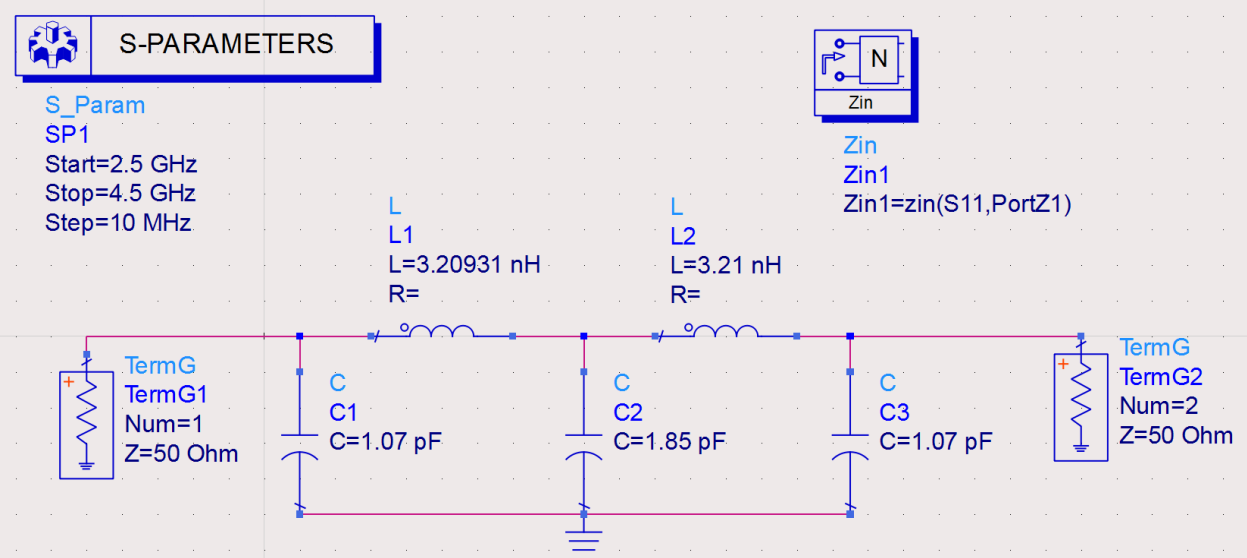


Figure 19. Low pass filter after impedance and frequency scaling.

Simulating circuit above to get the frequency response from 2.5 to 4.5 GHz, is possible to verify the BW corresponding to -3dB in magnitude. The result is show in figure 4, is observed that filter has 3.8 GHz of band width and almost 0% of losses for 3.4 GHz.

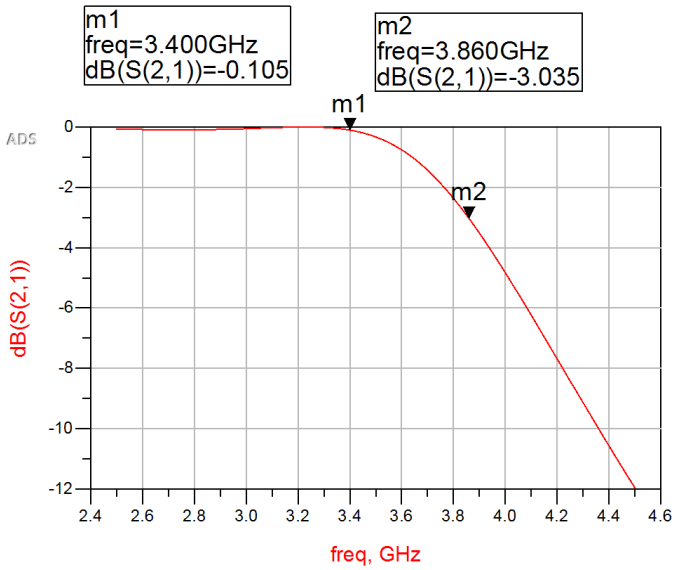


Figure 20. Frequency response from 2.5 to 4.5 GHz.

- Bandstop transformation

Low-pass prototype filter design can also be transformed to have the bandpass or bandstop responses, if w1 and w2 denote the edges of bandpass, then a stopband response can be obtaining using the following frequency substitution:

$$\Delta = \frac{\omega_{2-\omega_1}}{\omega_0}$$

Is the fractional bandwidth of bandstop, then the center is ω_0 .

About the rest of elements in the circuit exist a predefined conversion. From Microwave engineering, page 404, table 8.6 – Summary of prototype filter transformation.

Delta	**5.00E-02**
Serie LC g1	
L1_k	8.16E-10
C1_k	2.68E-12
Parallel LC g2	
L2_k	3.21E-12
C2_k	6.83E-10
Serie LC g3	
L3_k	4.74E-10
C3_k	4.62E-12
Parallel LC g4	
L4_k	3.21E-12
C4_k	6.83E-10
Serie LC g5	
L5_k	8.16E-10
C5_k	2.68E-12

Transformation topology is implemented in circuital manner like is show in figure 5.

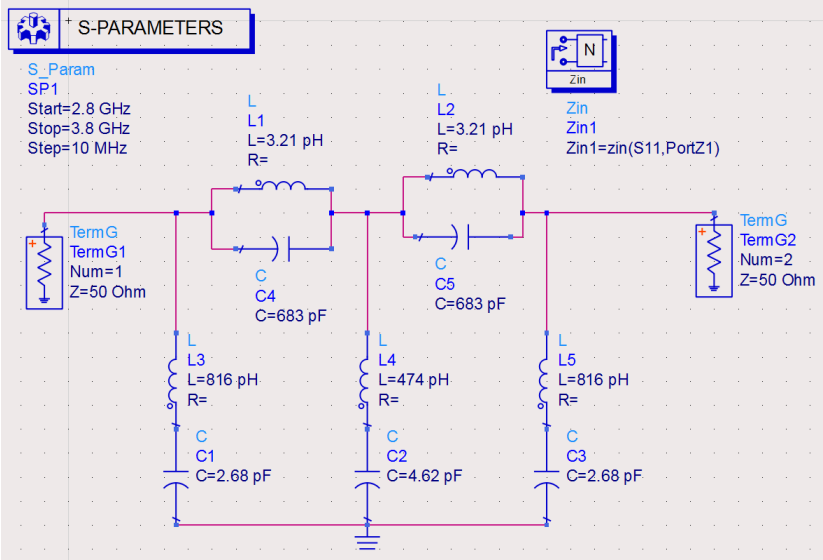


Figure 21.Low pass to band stop transformation.

Simulation response:

The frequency response corresponding to bandstop filter with the center frequency at 3.4 GHz and BW = 170 MHz, this meets the design specification.

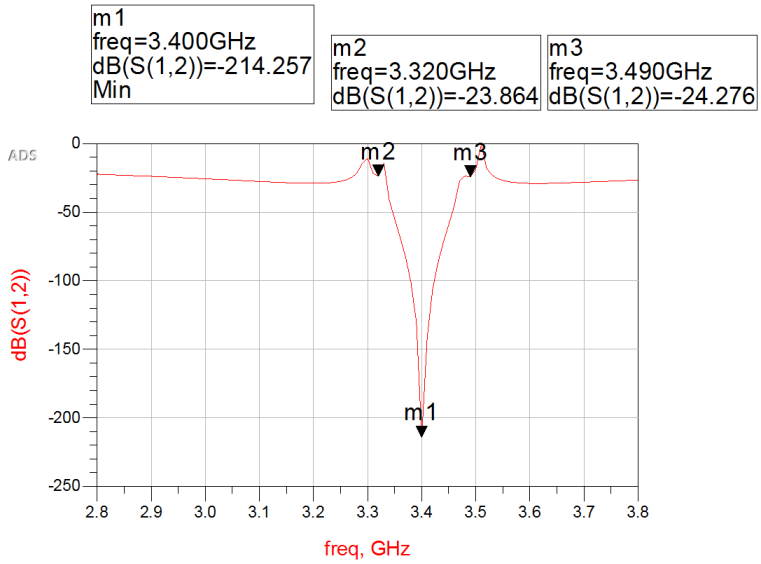
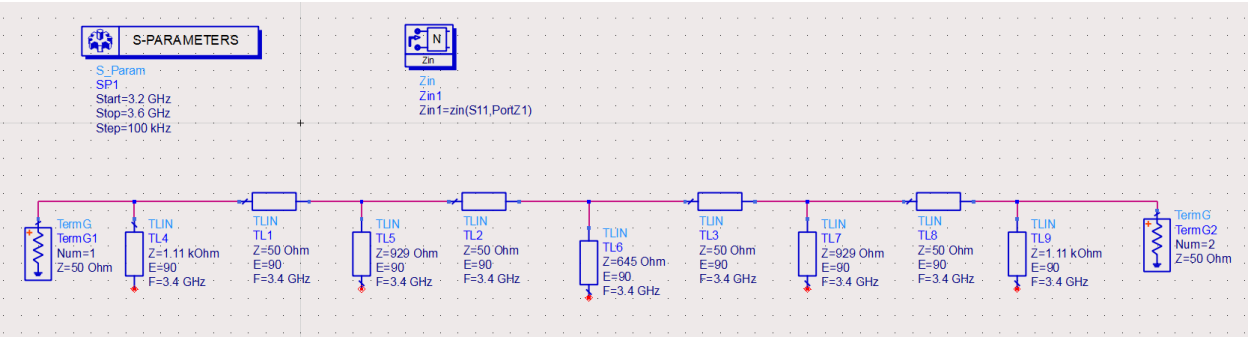


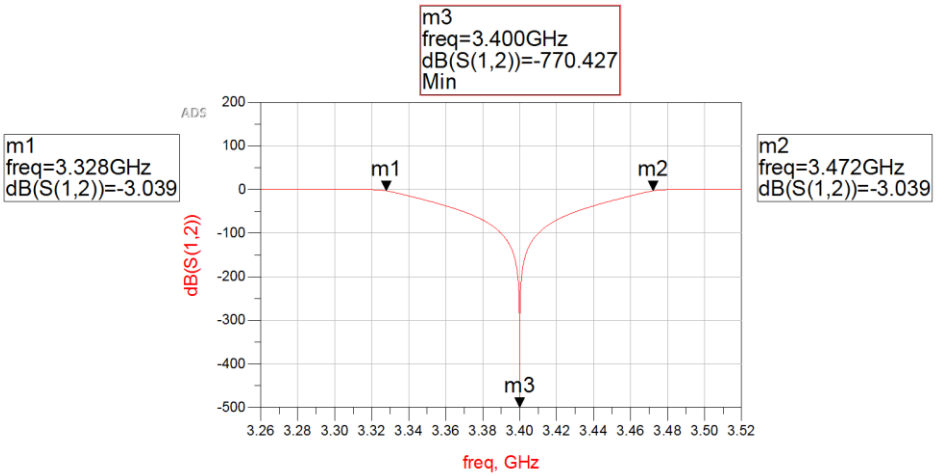
Figure 22.Band stop filter response.

• Layout

First approximation with ideal strips lines, implementation (Microwave engineering, David Pozar, third edition pag. 427)



Frequency response using ideals strips lines, is observed the behavior of the filter meets the specifications.



Using quarter wave resonators, the impedance for each stub:

$$Z_{0n} = \frac{4Z_0}{\pi g_{n\Delta}}$$

Where

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

n	gn	Zo
1	**1.1468**	1.11E+03
2	**1.3712**	9.29E+02
3	**1.975**	6.45E+02
4	**1.3712**	9.29E+02
5	**1.1468**	1.11E+03

Figure 7 show circuital implementation with ideal strip lines, L and W dimension are calculated using calc tool in ADR simulator.

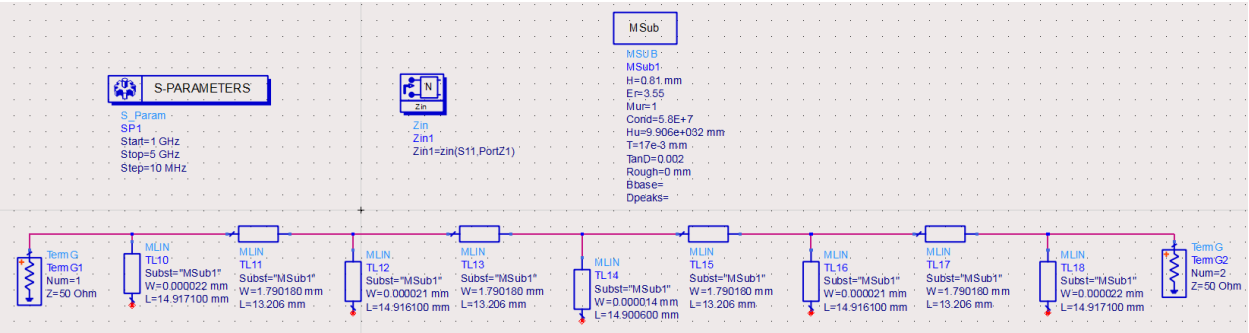


Figure 23.Stubs implementation for bandstop filter.

Frequency response is show in figure 8, this trace is not like expected. Need deep analysis to find root cause of this issue.

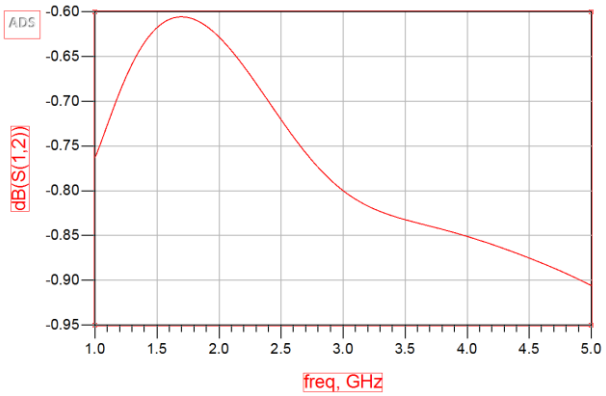


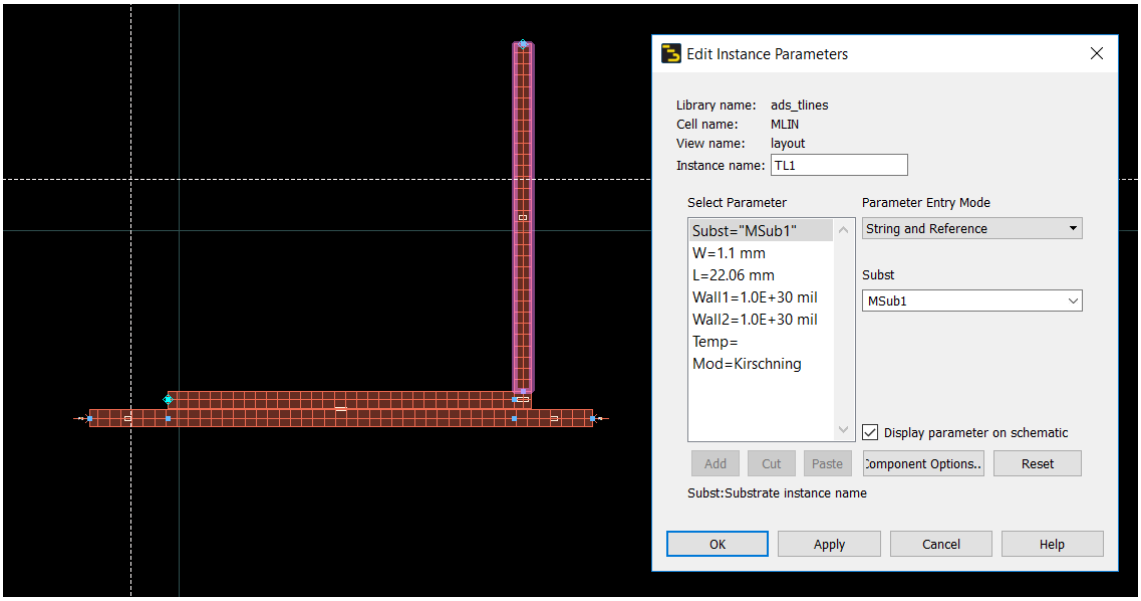
Figure 24. Frequency responds

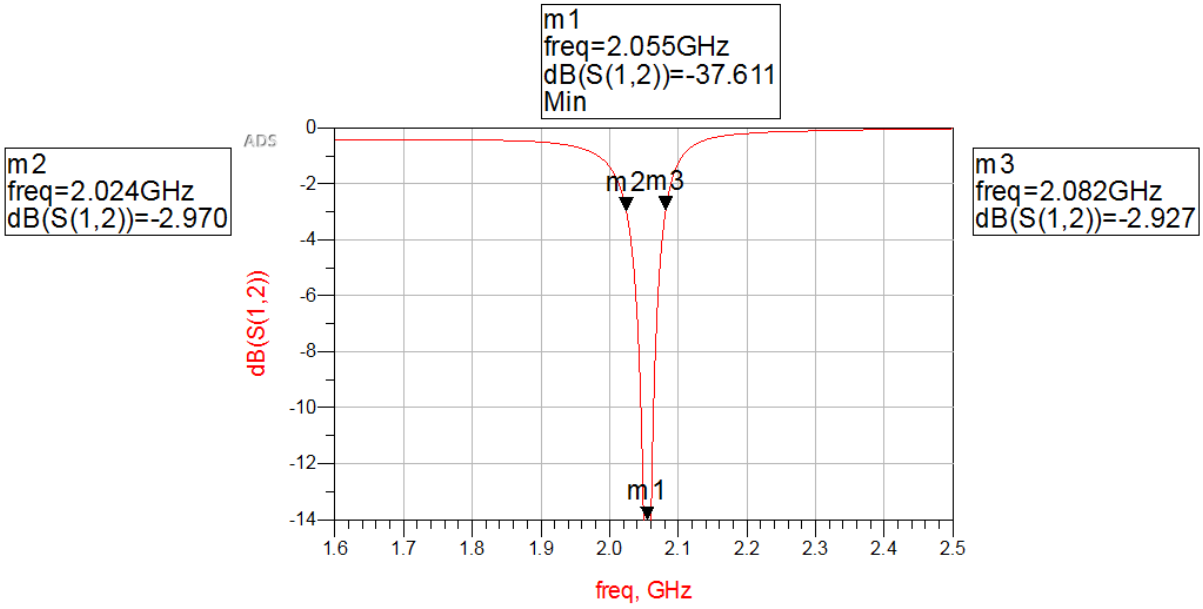
- Last section was not possible to get a satisfactory response.

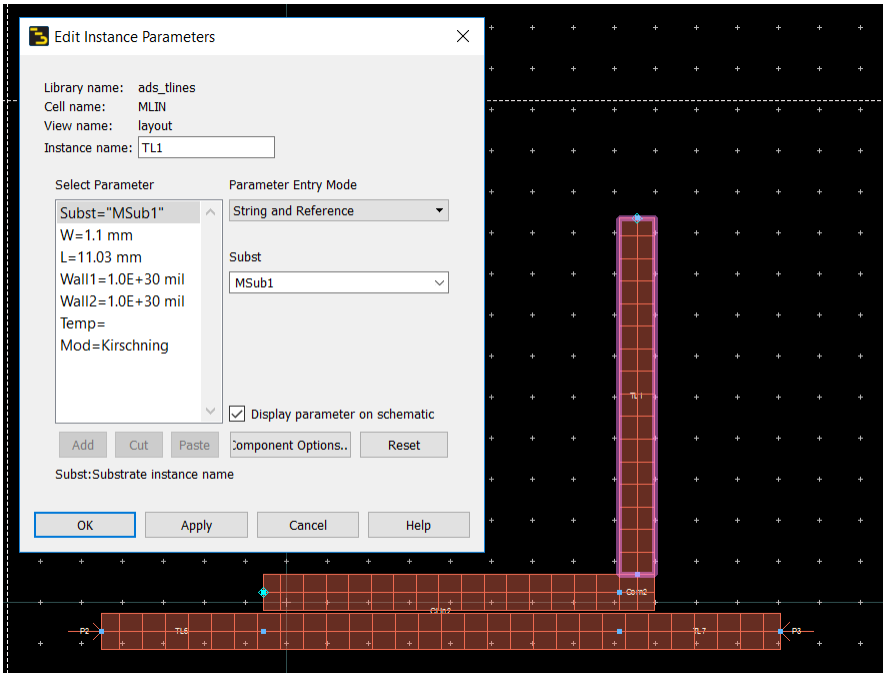
Mitigation: Find a practical example/implementation that could help to find a solution. Review material from a previous class get an example for a similar design where filter is implemented using L-resonator.

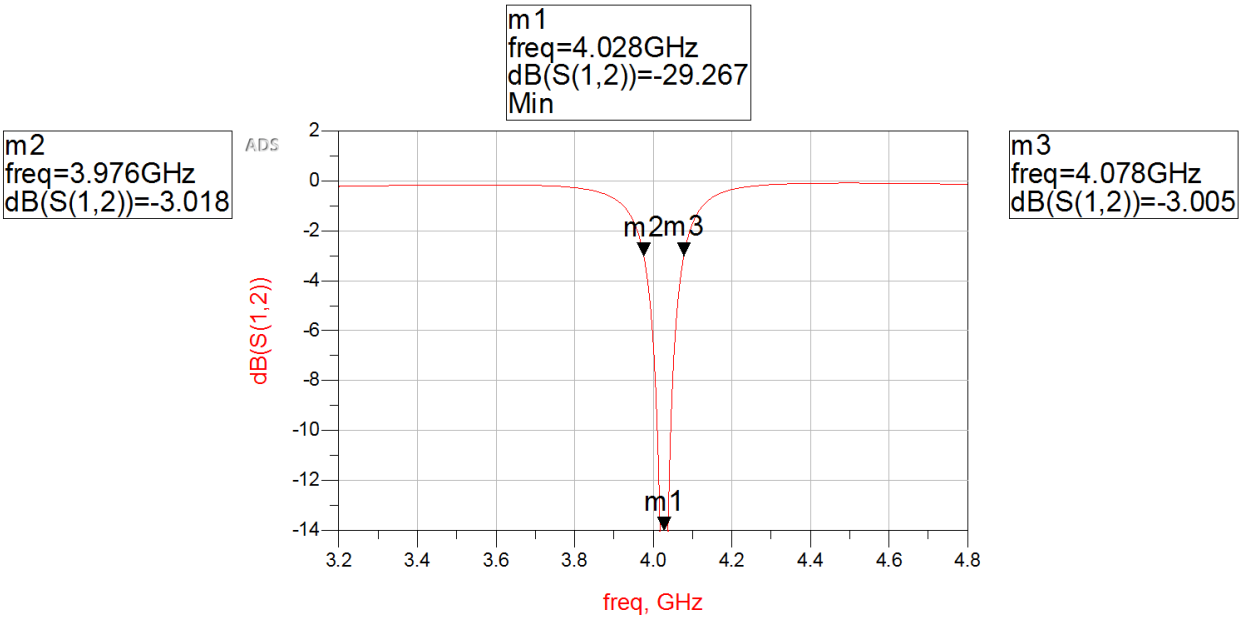
$$\lambda = \frac{c}{\omega_0} = 0.088235294$$

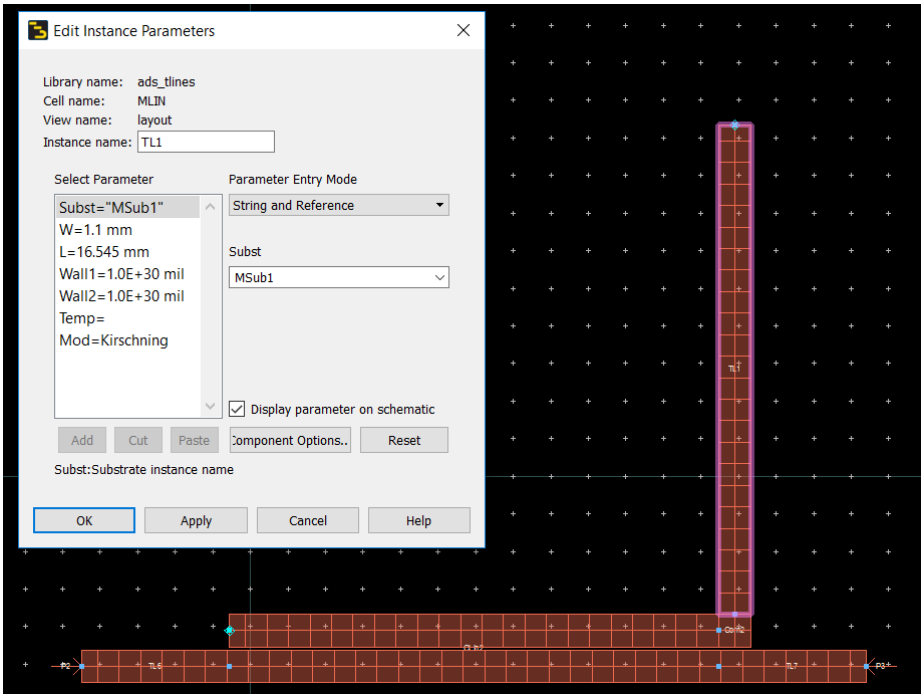
$$\frac{\lambda}{4} = 0.022058824$$

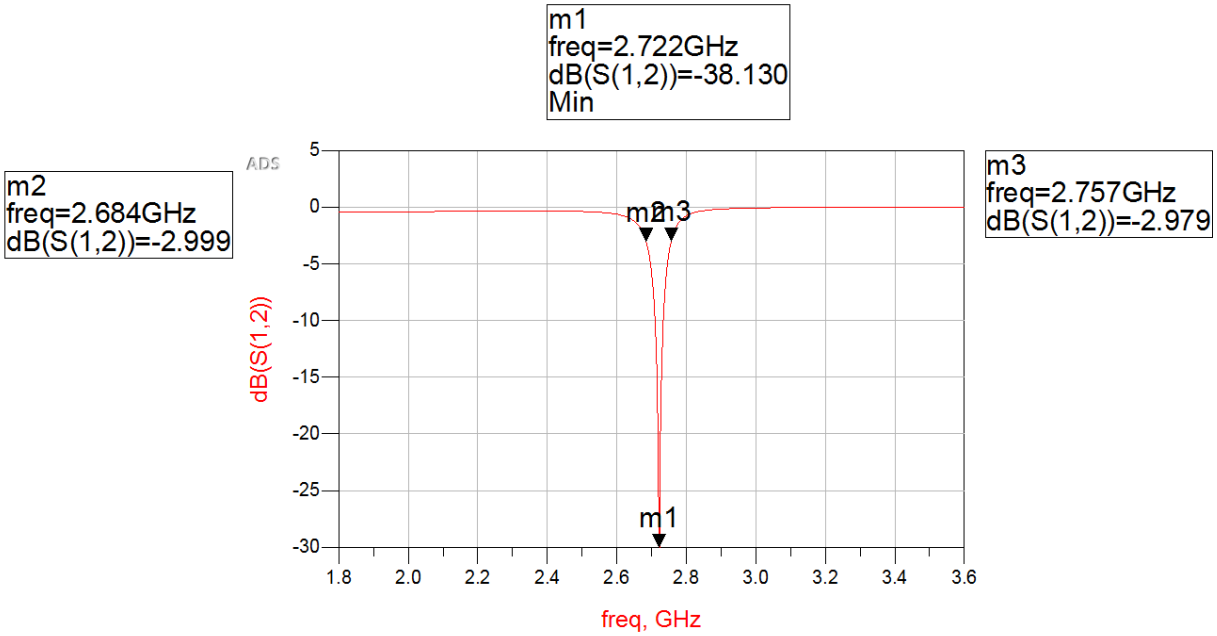

Edit Instance Parameters
Library name: ads_tlines
Cell name: MLIN
View name: layout
Instance name: TL1
Select Parameter
Parameter Entry Mode
Subst="MSub1"
W=1.1 mm
L=22.06 mm
Wall1=1.0E+30 mil
Wall2=1.0E+30 mil
Temp=
Mod=Kirschning
String and Reference
Subst
MSub1
Display parameter on schematic
Add
Cut
Paste
Component Options..
Reset
Subst:Substrate instance name
OK
Apply
Cancel
Help

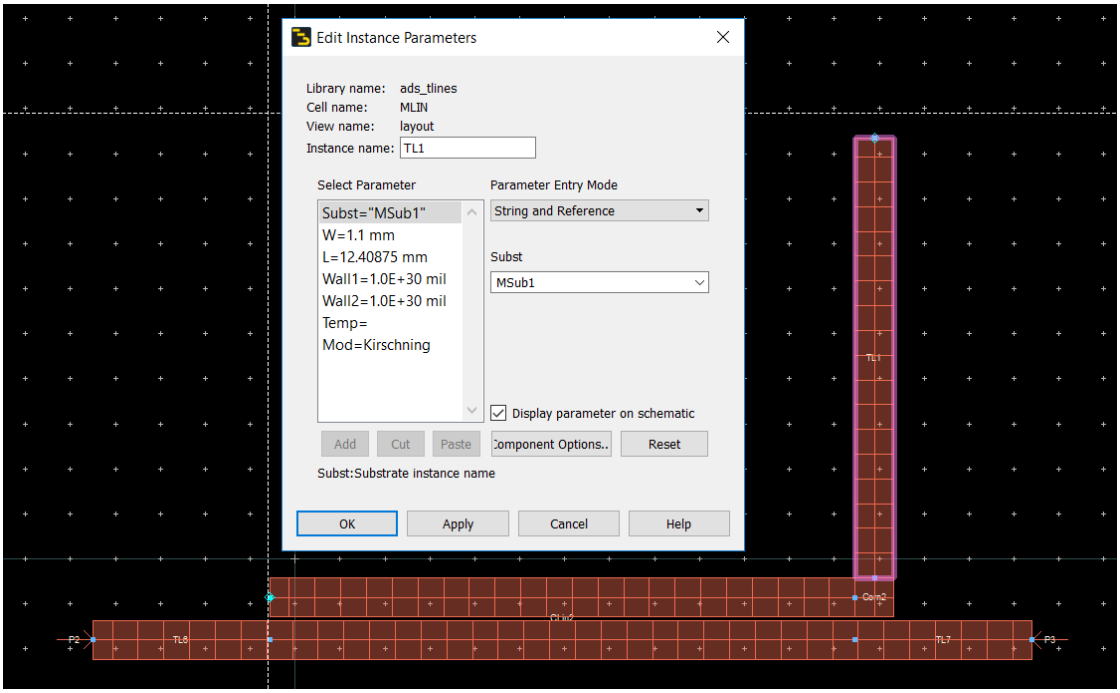

m1
freq=2.055GHz
dB(S(1,2))=-37.611
Min
m2
freq=2.024GHz
dB(S(1,2))=-2.970
m3
freq=2.082GHz
dB(S(1,2))=-2.927
ADS
dB(S(1,2))
freq, GHz
0
-2
-4
-6
-8
-10
-12
-14
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4
2.5

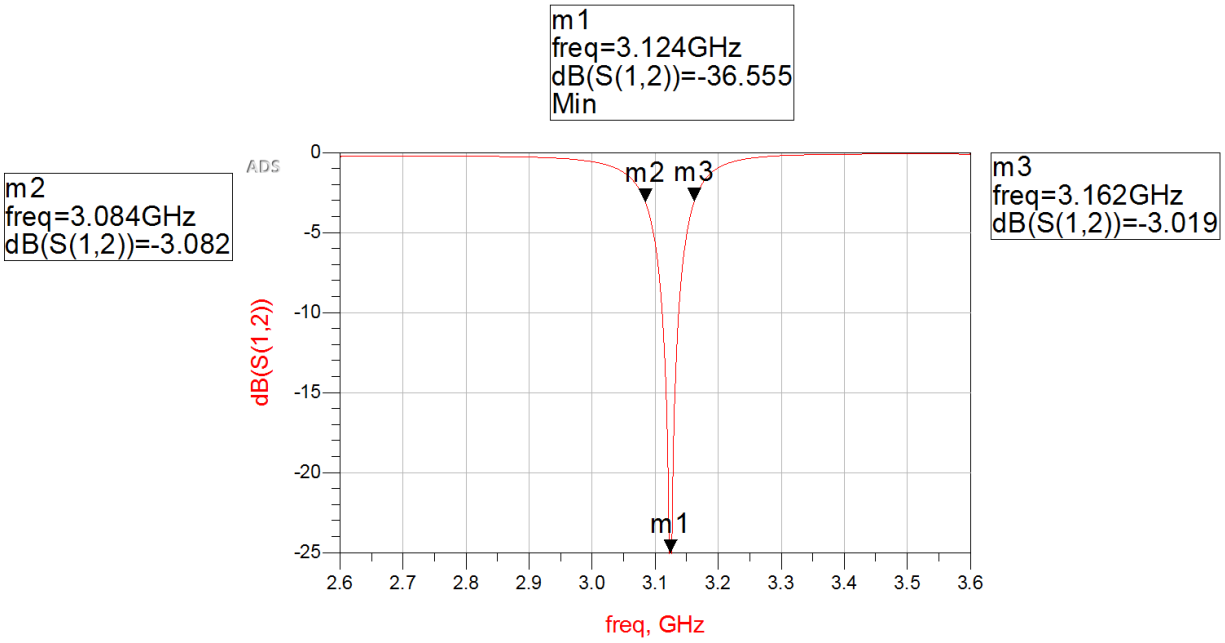

Edit Instance Parameters
Library name: ads_tlines
Cell name: MLIN
View name: layout
Instance name: TL1
Select Parameter
Parameter Entry Mode
Subst="MSub1"
W=1.1 mm
L=11.03 mm
Wall1=1.0E+30 mil
Wall2=1.0E+30 mil
Temp=
Mod=Kirschning
String and Reference
Subst
MSub1
Display parameter on schematic
Add
Cut
Paste
Component Options..
Reset
Subst:Substrate instance name
OK
Apply
Cancel
Help

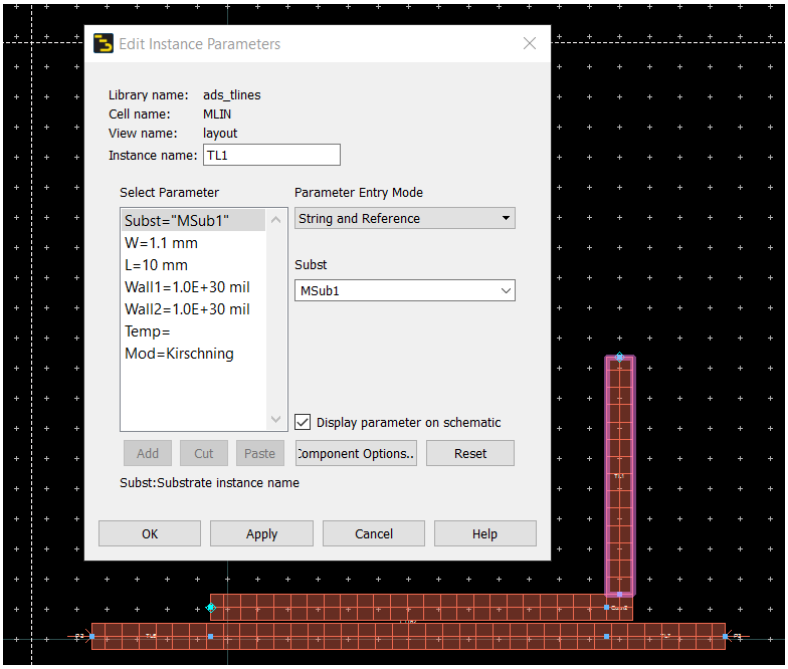

m1
freq=4.028GHz
dB(S(1,2))=-29.267
Min
m2
freq=3.976GHz
dB(S(1,2))=-3.018
m3
freq=4.078GHz
dB(S(1,2))=-3.005
ADS
m2m3
m1
dB(S(1,2))
freq, GHz
2
0
-2
-4
-6
-8
-10
-12
-14
3.2
3.4
3.6
3.8
4.0
4.2
4.4
4.6
4.8

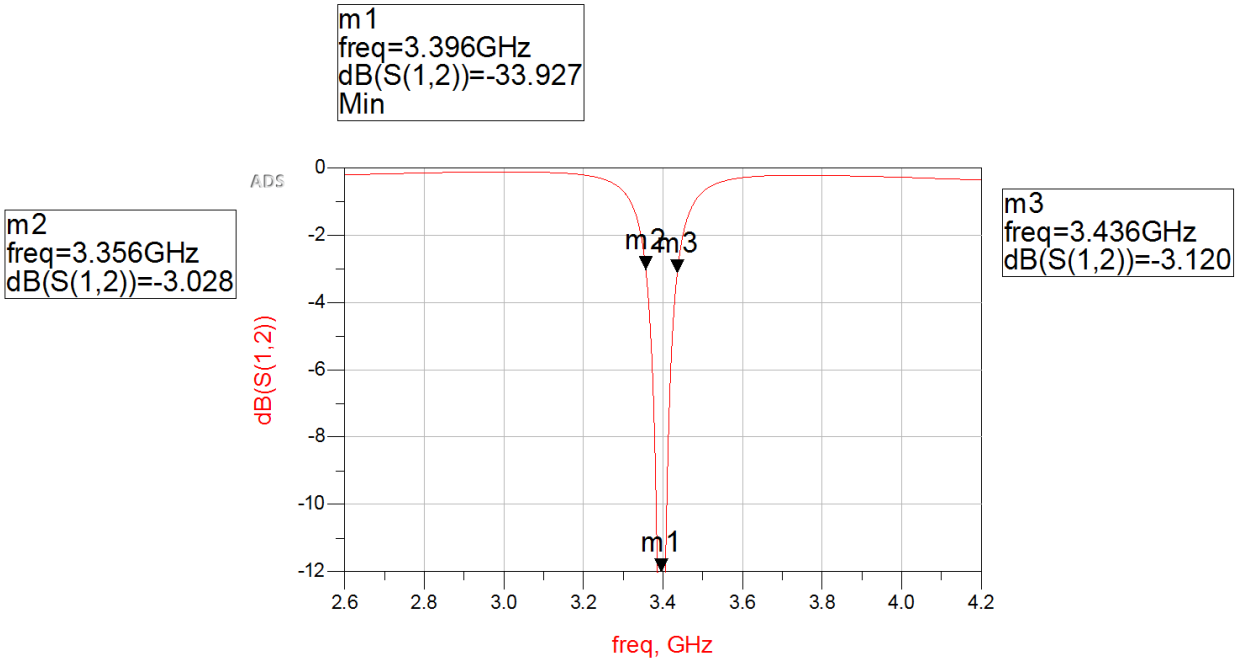

Edit Instance Parameters
Library name: ads_tlines
Cell name: MLIN
View name: layout
Instance name: TL1
Select Parameter
Parameter Entry Mode
Subst="MSub1"
W=1.1 mm
L=16.545 mm
Wall1=1.0E+30 mil
Wall2=1.0E+30 mil
Temp=
Mod=Kirschning
String and Reference
Subst
MSub1
Display parameter on schematic
Add
Cut
Paste
Component Options..
Reset
Subst:Substrate instance name
OK
Apply
Cancel
Help


m1
freq=2.722GHz
dB(S(1,2))=-38.130
Min
m2
freq=2.684GHz
dB(S(1,2))=-2.999
m3
freq=2.757GHz
dB(S(1,2))=-2.979
ADS
dB(S(1,2))
freq, GHz
5
0
-5
-10
-15
-20
-25
-30
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6


Edit Instance Parameters
Library name: ads_tlines
Cell name: MLIN
View name: layout
Instance name: TL1
Select Parameter
Parameter Entry Mode
Subst="MSub1"
W=1.1 mm
L=12.40875 mm
Wall1=1.0E+30 mil
Wall2=1.0E+30 mil
Temp=
Mod=Kirschning
String and Reference
Subst
MSub1
Display parameter on schematic
Add
Cut
Paste
Component Options..
Reset
Subst:Substrate instance name
OK
Apply
Cancel
Help
TL1
Term2
P2
TL8
TL7
P3


m1
freq=3.124GHz
dB(S(1,2))=-36.555
Min
m2
freq=3.084GHz
dB(S(1,2))=-3.082
m3
freq=3.162GHz
dB(S(1,2))=-3.019
ADS
m2 m3
m1
0
-5
-10
-15
-20
-25
dB(S(1,2))
2.6
2.7
2.8
2.9
3.0
3.1
3.2
3.3
3.4
3.5
3.6
freq, GHz


Edit Instance Parameters
Library name: ads_tlines
Cell name: MLIN
View name: layout
Instance name: TL1
Select Parameter
Parameter Entry Mode
Subst="MSub1"
W=1.1 mm
L=10 mm
Wall1=1.0E+30 mil
Wall2=1.0E+30 mil
Temp=
Mod=Kirschning
String and Reference
Subst
MSub1
Display parameter on schematic
Add
Cut
Paste
Component Options..
Reset
Subst:Substrate instance name
OK
Apply
Cancel
Help


m1
freq=3.396GHz
dB(S(1,2))=-33.927
Min
m2
freq=3.356GHz
dB(S(1,2))=-3.028
m3
freq=3.436GHz
dB(S(1,2))=-3.120
ADS
m2
m3
m1
0
-2
-4
-6
-8
-10
-12
dB(S(1,2))
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
4.2
freq, GHz

Finally, to increase bandwidth as we saw in class is necessary to add extra L-resonator using the same dimensions.

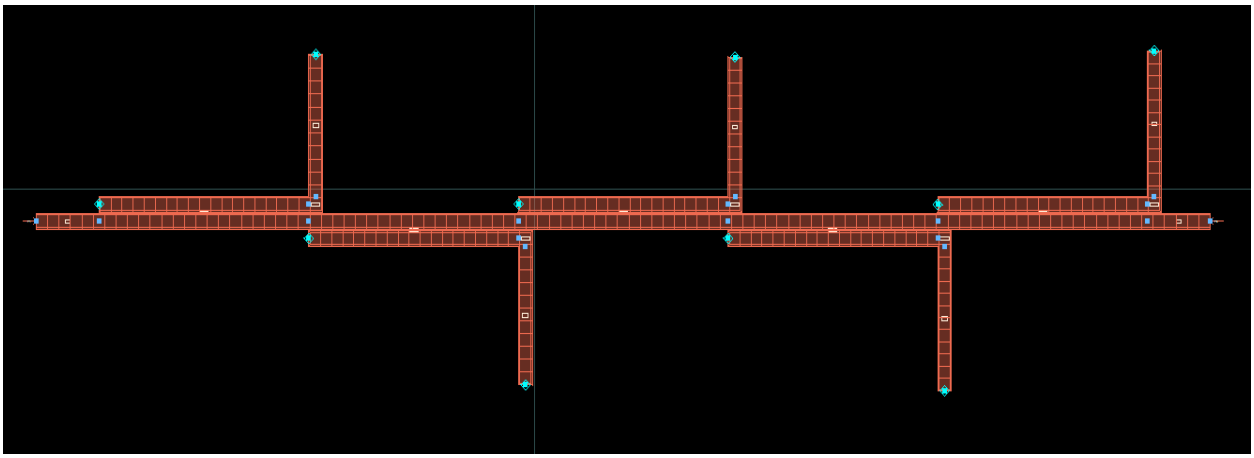
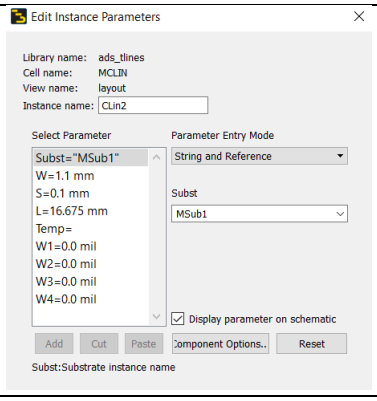
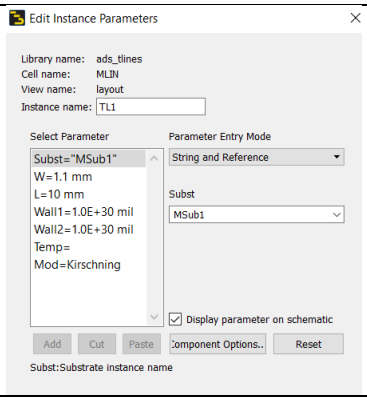
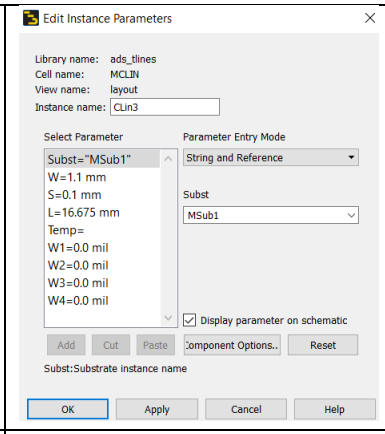
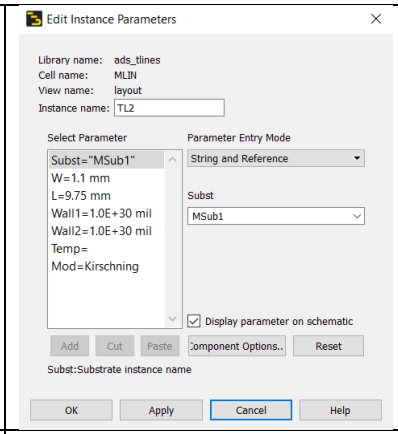
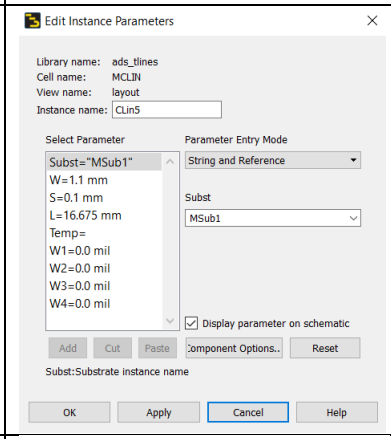
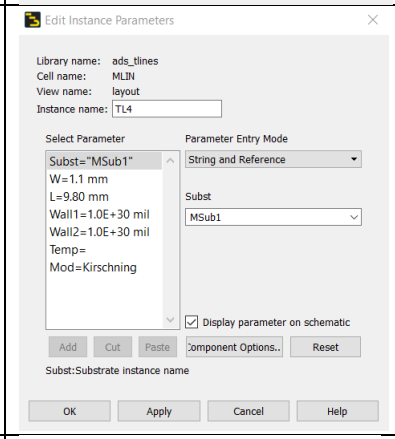
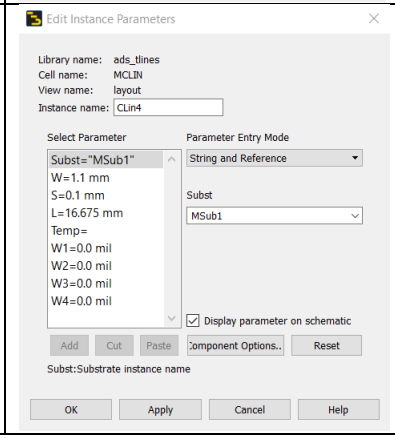
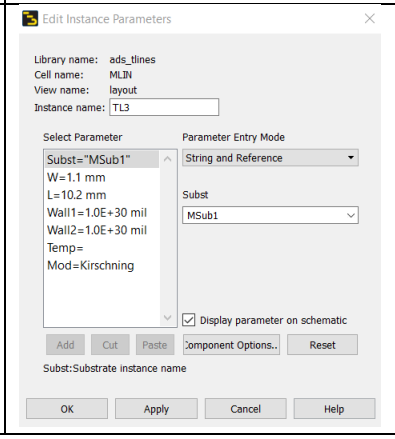
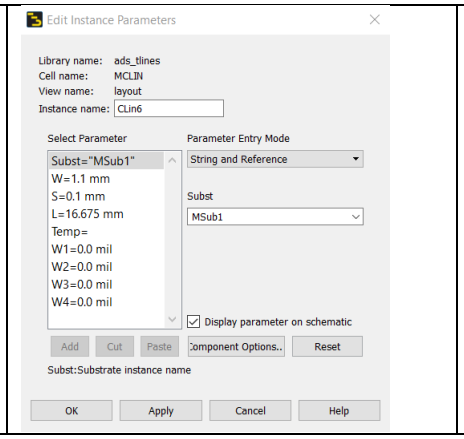
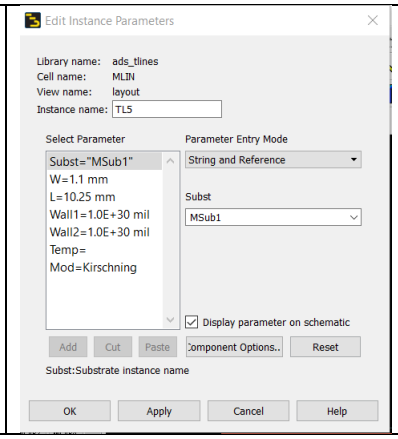


Figure 25.Filter layout.

An extra data from paper [1] the separation between principal line and the L-resonators are conserved and varying the longitude of each resonator. L- Resonators have the same horizontal proportion and little variation in verticals segments

Segment	Horizontal	Vertical
1		

2	 Edit Instance Parameters Library name: ads_tlines Cell name: MCLIN View name: layout Instance name: CLin3 Select Parameter Parameter Entry Mode Subst="MSub1" W=1.1 mm S=0.1 mm L=16.675 mm Temp= W1=0.0 mil W2=0.0 mil W3=0.0 mil W4=0.0 mil String and Reference Subst MSub1 Display parameter on schematic Add Cut Paste Component Options.. Reset Subst:Substrate instance name OK Apply Cancel Help	 Edit Instance Parameters Library name: ads_tlines Cell name: MLIN View name: layout Instance name: TL2 Select Parameter Parameter Entry Mode Subst="MSub1" W=1.1 mm L=9.75 mm Wall1=1.0E+30 mil Wall2=1.0E+30 mil Temp= Mod=Kirschning String and Reference Subst MSub1 Display parameter on schematic Add Cut Paste Component Options.. Reset Subst:Substrate instance name OK Apply Cancel Help
3	 Edit Instance Parameters Library name: ads_tlines Cell name: MCLIN View name: layout Instance name: CLin5 Select Parameter Parameter Entry Mode Subst="MSub1" W=1.1 mm S=0.1 mm L=16.675 mm Temp= W1=0.0 mil W2=0.0 mil W3=0.0 mil W4=0.0 mil String and Reference Subst MSub1 Display parameter on schematic Add Cut Paste Component Options.. Reset Subst:Substrate instance name OK Apply Cancel Help	 Edit Instance Parameters Library name: ads_tlines Cell name: MLIN View name: layout Instance name: TL4 Select Parameter Parameter Entry Mode Subst="MSub1" W=1.1 mm L=9.80 mm Wall1=1.0E+30 mil Wall2=1.0E+30 mil Temp= Mod=Kirschning String and Reference Subst MSub1 Display parameter on schematic Add Cut Paste Component Options.. Reset Subst:Substrate instance name OK Apply Cancel Help
4	 Edit Instance Parameters Library name: ads_tlines Cell name: MCLIN View name: layout Instance name: CLin4 Select Parameter Parameter Entry Mode Subst="MSub1" W=1.1 mm S=0.1 mm L=16.675 mm Temp= W1=0.0 mil W2=0.0 mil W3=0.0 mil W4=0.0 mil String and Reference Subst MSub1 Display parameter on schematic Add Cut Paste Component Options.. Reset Subst:Substrate instance name OK Apply Cancel Help	 Edit Instance Parameters Library name: ads_tlines Cell name: MLIN View name: layout Instance name: TL3 Select Parameter Parameter Entry Mode Subst="MSub1" W=1.1 mm L=10.2 mm Wall1=1.0E+30 mil Wall2=1.0E+30 mil Temp= Mod=Kirschning String and Reference Subst MSub1 Display parameter on schematic Add Cut Paste Component Options.. Reset Subst:Substrate instance name OK Apply Cancel Help

5		

[1] Source: Design and Implementation of a Microstrip Band-Stop Filter for Microwave Applications L. BalaSenthilMurugan*, S. Antony Anbu Raja, S. Deeban Chakravarthy, N.Kanniyappan

Bandstop filter response

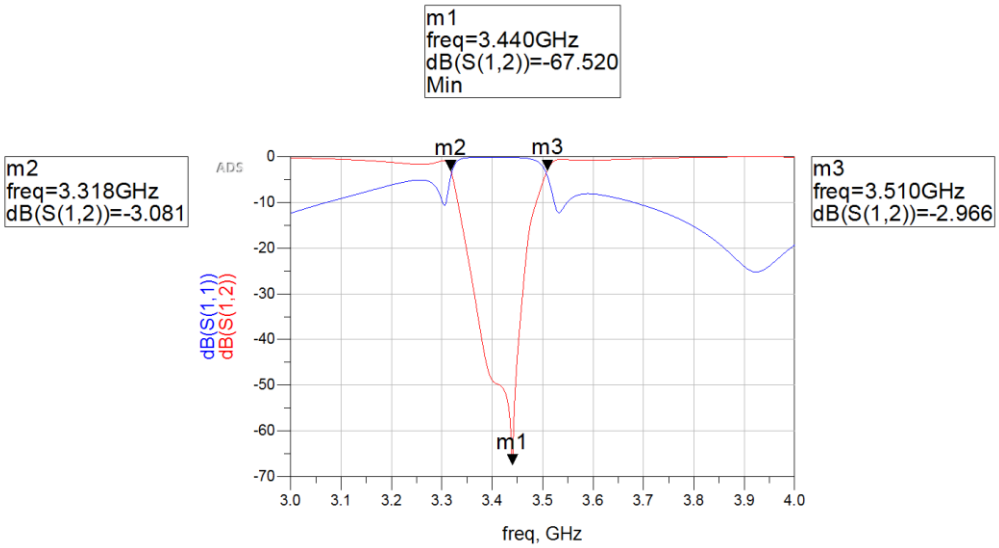


Figure 26

CONTENTS

4. Referencias

(s.f.). Obtenido de http://146.83.206.1/~jhuircan/PDF_CTOI/MultIee2.pdf

Carusone, T. C. (2012). ANALOG INTEGRATED CIRCUIT DESIGN. En D. A. Tony Chan Carusone, *ANALOG INTEGRATED CIRCUIT DESIGN.*

Eads, R. (s.f.). *PCI Express Electrical Basics*. Obtenido de PCI Express Electrical Basics: https://pcisig.com/sites/default/files/files/PCI_Express_Electrical_Basics.pdf

Equalization, R. a. (s.f.). *Response and Software Equalization*. Obtenido de National Instruments: http://zone.ni.com/reference/en-XX/help/371025N-01/rfsg/if_response_and_equalizer/

Instruments, N. (s.f.). *Response and Software Equalization.*

Pozar, D. (s.f.). 3.8 MOS DIFFERENTIAL PAIR AND GAIN STAGE. En D. Pozar, *Microwave Engineering.*

R. Carrillo, J. H. (s.f.). *Amplificadores Multietapa*. Obtenido de https://studylib.es/doc/5675361/amplificadores-multietapa