

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 el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

MAESTRÍA EN DISEÑO ELECTRÓNICO



**EN LA MODALIDAD: "FORMACIÓN COMPLEMENTARIA EN
ÁREA DE CONCENTRACIÓN"**

**ÁREA DE CONCENTRACIÓN: "DISEÑO DE CIRCUITOS
INTEGRADOS ANALÓGICOS"**

Trabajo recepcional que para obtener el grado de
MAESTRO EN DISEÑO ELECTRÓNICO

Presenta: Ing. Gustavo Guzman Rosales

Asesor: Dr. Esteban Martínez Guerrero

Asesor: Dr. Ivan Padilla Cantoya

Asesor: Dr. Zabdiel Brito Brito

Tlaquepaque, Jalisco, julio de 2015.

MAESTRO EN INGENIERÍA (2015)
Maestría en Diseño Electrónico

ITESO
Tlaquepaque, Jal., México

ÁREA DE CONCENTRACIÓN: “Diseño de Circuitos Integrados Analógicos”

AUTOR: Gustavo Guzman Rosales.
Ingeniero en mecatrónica (Centro de Enseñanza Técnica
Industrial CETI, México)

REVISORES: Dr. Esteban Martínez Guerrero
Dr. Ivan Padilla Cantoya
Dr. Zabdiel Brito Brito

NÚMERO DE PÁGINAS: vii, 88

Contenido

1. Resumen de los proyectos realizados	3
1.1. PROYECTO 1: DISEÑO DE LAYOUT DEL <i>TIMER 555</i>	4
1.1.1 Introducción	4
1.1.2 Antecedentes	4
1.1.3 Solución Desarrollada	5
1.1.4 Análisis de resultados.....	5
1.1.5 Conclusiones	5
1.2. PROYECTO 2: INTERPOLADOR 8- 32.....	6
1.2.1 Introducción	6
1.2.2 Antecedentes	7
1.2.3 Solución Desarrollada	7
1.2.4 Análisis de resultados.....	8
1.2.5 Conclusiones	8
1.3. PROYECTO 3: DISEÑO DE CP-PLL.....	9
1.3.1 Introducción	9
1.3.2 Antecedentes	9
1.3.3 Solución Desarrollada	9
1.3.4 Análisis de resultados.....	9
1.3.5 Conclusiones	10
2. Conclusiones	13
Apéndices	15
A. REPORTE DE PROYECTO 1: INTERPOLADOR	16
B. REPORTE DE PROYECTO 2: LAYOUT DEL 555	17
C. REPORTE DE PROYECTO 3: PLL.....	18

Introducción

La maestría en diseño electrónico se divide en diferentes áreas de estudio las cuales son relacionadas al diseño de circuitos integrados digitales, circuitos integrados analógicos, sistemas digitales, sistemas embebidos y telecomunicaciones y diseño electrónico de alta frecuencia, de todas ellas se eligió una área en la que los proyectos desarrollados en las materias de esta área fueron sobresalientes debido a los conocimientos aprendidos, el impacto del proyecto y el aporte de los mismos en trabajos futuros.

El presente reporte técnico engloba tres de los proyectos realizados durante la maestría en diseño electrónico en las materias relacionadas al área de diseño de circuitos integrados analógicos, estos proyectos fueron seleccionados debido a que los resultados obtenidos fueron importantes ya que demuestran lo aprendido en las materias siguientes:

- Diseño de circuitos integrados analógicos.
- Diseño físico de circuitos integrados.
- Diseño avanzado de circuitos integrados.

1. Resumen de los proyectos realizados

Los proyectos elegidos fueron desarrollados durante cada una de las materias, por lo cual cada proyecto engloba los conocimientos aprendidos en estas materias de la maestría en diseño electrónico, la metodología de desarrollo de dichos proyectos es muy similar entre si debido a la naturaleza de estos y es posible resumirse en las siguientes etapas:

- Elección del proyecto.
- Diseño.
- Implementación.
- Evaluación de resultados.
- Conclusiones finales.

Para cada una de las materias la elección del proyecto fue diferente y en algunos casos el proyecto fue asignado dando lugar a la etapa siguiente en la cual se diseñó basado en los requerimientos dados por el maestro de cada asignatura. El proceso de implementación de los proyectos del timer y del interpolador fueron similares debido a que estamos hablando de proyectos de layout sin embargo los resultados a evaluar de cada proyecto fueron diferentes.

1.1. Proyecto 1: Diseño de *layout* del *timer* 555

1.1.1 Introducción

Ante la necesidad de obtener circuitos generadores de pulsos, multivibradores (temporizadores) se crearon circuitos basados en amplificadores operacionales en distintas aplicaciones. Sin embargo en 1972 la compañía Signetics introdujo en el mercado un nuevo componente, que no solo cumplía con estas necesidades, sino que mejoraba los resultados obtenidos por los circuitos basados en amplificadores operacionales en muchos aspectos.

Hoy en día el 555 sigue siendo un componente básico en la construcción de circuitos multivibradores, generadores de pulsos, divisores de frecuencia. La principal ventaja del 555 radica en que consigue temporizaciones más precisas. Además, al ser un circuito integrado reduce el número de conexiones así como el precio, factor que todo ingeniero debe tener en cuenta a la hora del diseño.

El objetivo del proyecto es aplicar las diferentes técnicas de *layout full-custom* revisadas en clase, para la creación de un *layout* de un circuito de modo mixto.

1.1.2 Antecedentes

El 555 es un circuito integrado que incorpora dentro de sí dos comparadores de voltaje, un flip flop, una etapa de salida de corriente, divisor de voltaje resistor y un transistor de descarga. Dependiendo de cómo se interconecten estas funciones utilizando componentes externos es posible conseguir que dicho circuito realice un gran número de funciones tales como la del multivibrador astable y la del circuito monoestable. Este integrado se puede usar en diversas aplicaciones, tales como:

- Control de sistemas secuenciales
- Generación de tiempos de retraso
- Divisor de frecuencias
- Modulación por anchura de pulsos
- Repetición de pulsos

- Generación de pulsos controlados por tensión, etc.

1.1.3 Solución Desarrollada

Para este proyecto se implemento el circuito integrado 555, se realizaron los esquemáticos y *layouts* de los diferentes bloques digitales y analógicos que lo componen, así mismo se realizaron pruebas de DRC, LVS y se extrajeron las capacitancias y resistencias parasitas para cada bloque del circuito. Una vez terminados los sub bloques, se implementó el *layout* completo del circuito el cual se realizó pensando en hacer un *layout* simétrico y con una buena distribución de ruteo para que las señales recorrieran la misma distancia entre los bloques.

1.1.4 Análisis de resultados

Se realizo un análisis de las simulaciones del proyecto final comparando los resultados de simulación pre y post *layout*, donde se observaron diferentes resultados, se llego a la conclusión de que estas variaciones son debidas a las resistencias y capacitancias parasitas que son despreciadas en una simulación pre *layout*, con la finalidad de reducir estas variaciones en las mediciones se aplicaron diferentes técnicas de *layout* como las de interdigitación y centroide común.

1.1.5 Conclusiones

Se puede observar de las simulaciones pre *layout* y post *layout*, que en post *layout* la frecuencia disminuye teniendo un cambio o una diferencia de 2.12KHz, este error se redujo gracia a que se han empleado las de técnicas de *layout* y de interdigitación y centroide común así como las de colocar anillos de guarda, *dummys* entre otras. Creo que con más tiempo hubiera sido bueno realizar otro *layout* sin aplicar las técnicas que se aprendieron durante el curso de diseño físico de circuitos integrados.

La diferencia que hay en el pre *layout* y el post *layout* son debido a que en el pre *layout* no se consideran las parasitas, ni los ruteos de las señales, los cuales en algunos casos fueron largos.

Durante el desarrollo de este proyecto me enfrente con algunos problemas asociados con el uso de la herramienta utilizada en la simulación, en específico, con que la simulación en transitorio que realice para estar monitoreando la salida, cuando el tiempo era de 100us la salida oscilaba correctamente, y cuando el tiempo era de 200us la salida no oscilaba, lo único que creo sobre esto es que el muestreo de la señal que hace el simulador afecta.

Por último, con este proyecto me doy cuenta de cómo un sistema, analógico puede convivir con un sistema digital, así como las técnicas de *layout* que se utilizan para cada tipo de sistema son diferentes, por lo cual es indispensable conocer sobre qué tipo de sistema se está realizando el *layout* para poder aplicar las técnicas adecuadas para el mejor funcionamiento del sistema completo.

1.2. Proyecto 2: Interpolador 8- 32

1.2.1 Introducción

En este proyecto se realizo el analisis, diseño e implementación de un circuito interpolador generador de señales a 100MHz en tecnología CMOS, la estructura del interpolador generador implementado, recibe ocho señales igualmente espaciadas en fase provenientes de un oscilador controlado por voltaje (VCO) en la primera etapa, así generando 32 señales con las mismas características en fase. Para obtener las diferencias de fases en las etapas de interpolación, se parte de que la frecuencia de operación es de 100Mhz, el periodo de esta frecuencia de operación es de 10ns, dividido entre 8 bloques generadores de rampa, de los cuales se obtienen 8 rampas con una diferencia de fase de 1.25ns en cada una de ellas. Estas 8 rampas entran a la primera etapa de interpolación, obteniendo a la salida 16 señales de reloj con una diferencia de fase de 625ps entre cada una de ellas. Estas 16 señales de reloj entran a la segunda etapa de generación de rampa para que sean procesadas en la segunda etapa interpoladora

generándose así 32 señales de reloj con una diferencia de fase entre cada una de ellas de 312.5ps requeridos por la especificación de diseño.

1.2.2 Antecedentes

Un interpolador es un circuito generador de señales de reloj, a partir de un número reducido de señales de reloj en su entrada.

En un interpolador la generación de más señales de reloj se logra mediante el retraso de señales a su entrada, donde generalmente se usa una carga RC implementada con transistores y puede ser programable o tener arreglos para cubrir el funcionamiento correcto dentro de especificaciones, el reto en el diseño de interpoladores es cada vez más difícil porque los protocolos a manejar utilizan frecuencias cada vez mayores y errores de fase más pequeños, por otro lado los sistemas utilizan voltajes cada vez más pequeños haciendo los sistemas más sensibles a ruidos y a *jitter* (temblor en las señales).

Típicamente un interpolador debe realizar su función bajo diferentes condiciones de operación, cumpliendo con una frecuencia de operación mínima y máxima, error entre fases y consumo de potencia máximos, es por esto que se investiga el comportamiento y robustez de la topología interpoladora digital para diferentes configuraciones de layout.

Entendamos por interpolador a un circuito generador de señales de reloj, a partir de un número reducido de señales de reloj en su entrada, las señales de salida están en el rango de tiempo delimitado por las señales de entrada

1.2.3 Solución Desarrollada

Debido a que el propósito de este proyecto fue buscar una topología de *layout* que cumpliera con los requerimientos de frecuencia de operación del interpolador, se realizaron dos

layouts para el mismo, la diferencia entre estos *layouts* consiste en la ubicación de las etapas generadoras de rampa, en el primer *layout* la etapa generadora de rampa se ubican todas estas celdas en el centro y juntas, en el segundo *layout* estas celdas se dividieron en dos grupos los cuales se colocaron al centro formando dos columnas permitiendo pasar señales por el centro del *layout* y por los lados de este con la finalidad de tener líneas de metal rectas.

1.2.4 Análisis de resultados

Para los dos *layouts* propuestos se realizaron simulaciones *prelayout* con el fin de obtener un punto de partida al momento en que se realizaron las simulaciones *post layout*, se observó que en las simulaciones *post layout* las diferencias de fase requerida no eran las que se especificaban en los requerimientos del diseño, por tal motivo se aplicaron diferentes técnicas de *layouts* con la finalidad de obtener las diferencias de fase requeridas tales como la técnica de serpentina y la técnica de agregar metal, después de corregir los *layouts* aplicando las estas técnicas, se realizaron las simulaciones *post layout* para verificar como ayudaron a poner en fase las señales de salida del interpolador.

1.2.5 Conclusiones

Un interpolador es un circuito generador de señales de reloj, a partir de un número reducido de señales de reloj en su entrada.

En este proyecto se realizaron 2 propuestas *layout* con sus respectivas simulaciones a partir del *floorplan* realizado para el diseño. Básicamente el *floorplan* es la ubicación física de los diferentes elementos que constituyen el diseño en el Silicio; No obstante, la idea principal del *floorplan* es la optimización del espacio y la ubicación de los elementos de tal forma que minimice el retardo entre bloques para lograr un diseño que se desempeñe de mejor manera.

El ruteo debe ser lo más recto y simple posible, evitando a toda costa caminos innecesarios y demasiados quiebres, además de buscar que las señales recorran la misma distancia, para un desempeño correcto.

En este proyecto se hizo uso de algunas técnicas para compensar las señales cuyas diferencias entre fases quedaran dentro de lo esperado como el uso de serpentinas.

1.3. Proyecto 3: Diseño de CP-PLL

1.3.1 Introducción

Este proyecto tiene como objetivo, diseñar un PLL (*Phase Locked Loop*), obteniendo el modelo transitorio y en AC de un PLL el cual es un sistema realimentado, que en función de una señal de referencia (magnitud y fase) genera un voltaje de control para un VCO, el cual oscila hasta marrarse a la frecuencia de la señal de entrada.

1.3.2 Antecedentes

Un PLL es un circuito que se realimenta con la finalidad de generar una señal de salida con amplitud fija y con una frecuencia basada en la señal de entrada por lo cual es utilizado como sintetizador de frecuencias, osciladores y filtros de seguimiento entre otras aplicaciones.

1.3.3 Solución Desarrollada

Para satisfacer los requerimientos de diseño se implementaron cada uno de los bloques que componen el CP-PLL de manera que se pudiera variar el valor del divisor fácilmente, con el fin de poder sintonizar el PLL en los valores de la especificación, se realizaron los modelos en AC y transitorio, los cuales fueron probados para divisores con valores iguales a 2, 4, 6, 8, 12, 16 y 24.

1.3.4 Análisis de resultados

Para los siete modelos del PLL implementado se realizaron simulaciones que permitieron la medición de los siguientes parámetros:

- *Output skew*
- Tiempos de *rise, fall* y *duty cycle*
- Respuesta transitoria
- *Lock time*
- *Cycle jitter*
- *Phase jitter*
- *Static phase offset*
- Compatibilidad con *spread spectrum*

Se realizaron comparaciones entre los resultados para verificar que sin importar el divisor utilizado el PLL cumpliría con los parámetros de especificados en los requerimientos además de verificar el comportamiento del PLL al variar el divisor.

1.3.5 Conclusiones

El diseño del PLL se divido en dos partes, la primera fue el diseño del modelo en A.C. el cual sirvió para conocer la respuesta en frecuencia del PLL así como la estabilidad y el tiempo de amarre del PLL, este modelo es muy sencillo de realizar o de obtener, tanto el análisis matemático que se desarrolló con ayuda de MatLab, como la simulación que se realizó con el modelo en VerilogA.

La segunda parte de este proyecto, fue el de obtener el modelo transitorio del PLL, este modelo llevo mucho tiempo, debido a la falta de entendimiento sobre el funcionamiento de algunos de los bloques del PLL, las simulaciones de este modelo requirieron de más tiempo, sobre todo la simulación donde se quería conocer si el PLL era compatible con el Spread Spectrum, esta simulación llevo casi 8 horas, las otras simulaciones fueron relativamente más rápidas, aunque comparadas con la simulación del modelo en AC, si hay mucho tiempo de diferencia, así también como el número de simulaciones que se tuvieron que realizar, en el modelo de AC solo se realizaron alrededor de 3, y en el modelo transitorio fueron bastantes.

En resumen se puede decir que el diseño del PLL fue algo no muy sencillo, es muy importante antes de iniciar el diseño entender cada uno de los bloques que componen un PLL, así

como las unidades y dominios de cada uno de ellos, sino es así nunca se llegara a diseñar un PLL que cumpla con las especificaciones de diseño.

2. Conclusiones

En el presente reporte se presenta el resumen de tres proyectos realizados en la maestría en diseño electrónico en el área de concentración de circuitos analógicos. Como se observó la metodología para llevar a cabo cada uno de los proyectos fue muy similar entre sí debido a la naturaleza de los mismos, para todos ellos se planteó el proyecto, se realizó un diseño físico (*layout*) y la implementación del mismo, aplicando las distintas técnicas aprendidas en el curso, se verificó el circuito ejecutando simulaciones que nos permitieron validar los diferentes parámetros de los requerimientos planteados, y al final realizar una comparación de los datos obtenidos.

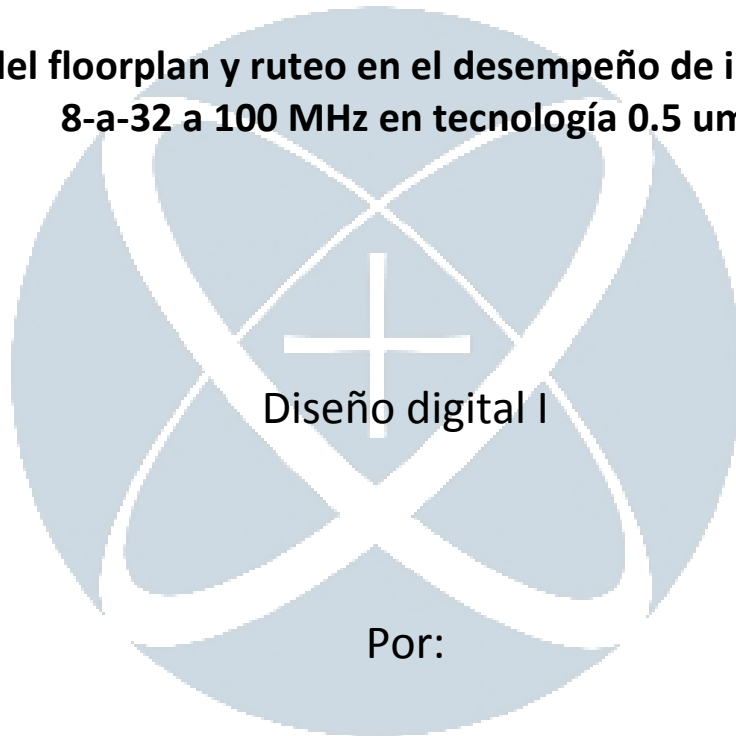
En los diferentes proyectos se puede ver que es muy importante la planeación del ruteo de las señales tanto digitales como analógicas en un *layout* ya que estas provocarían ruido, es importante aislar la parte digital de la parte analógica en un circuito para evitar comportamientos no esperados debido al ruido. Por este motivo en las simulaciones es necesario agregar las resistencias y capacitancias parasitarias que se generan al implementar el circuito para que los resultados de simulación sean más parecidos a los que se tendrán en la realidad.

Apéndices

A. REPORTE DE PROYECTO 1: INTERPOLADOR

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE OCCIDENTE**

**Efecto del floorplan y ruteo en el desempeño de interpolador
8-a-32 a 100 MHz en tecnología 0.5 um**



Gustavo Guzmán Rosales 674630

ITESO
G5

Universidad Jesuita

Prof.: Dr. Manuel Salim Maza

Guadalajara, Jalisco 2009

ÍNDICE

Introducción.....	3
Descripción.....	5
Características y Especificaciones.....	5
Etapas de interpolador.....	6
Simulaciones Pre-Layout.....	12
Propuestas Layout	20
Simulaciones Post-Layout.....	23
Ajuste de Layout.....	28
Layout Compensados.....	33
Conclusiones.....	37
Referencias.....	37

Interpoladores

Introducción:

Un interpolador es un circuito generador de señales de reloj, a partir de un número reducido de señales de reloj en su entrada.

En un interpolador la generación de más señales de reloj se logra mediante el retraso de señales a su entrada, donde generalmente se usa una carga RC implementada con transistores y puede ser programable o tener arreglos para cubrir el funcionamiento correcto dentro de especificaciones, el reto en el diseño de interpoladores es cada vez más difícil porque los protocolos a manejar utilizan frecuencias cada vez mayores y errores de fase más pequeños, por otro lado los sistemas utilizan voltajes cada vez mas pequeños haciendo los sistemas más sensibles a ruidos y a jitter (temblor en las señales).

Típicamente un interpolador debe realizar su función bajo diferentes condiciones de operación, cumpliendo con una frecuencia de operación mínima y máxima, error entre fases y consumo de potencia máximos, es por esto que se investiga el comportamiento y robustez de la topología interpoladora digital para diferentes configuraciones de layout.

Entendamos por interpolador a un circuito generador de señales de reloj, a partir de un número reducido de señales de reloj en su entrada, las señales de salida están en el rango de tiempo delimitado por las señales de entrada, tal como se muestra en la figura siguiente:



Ilustración 1 Esquema Interpolador de 2 a 3 señales

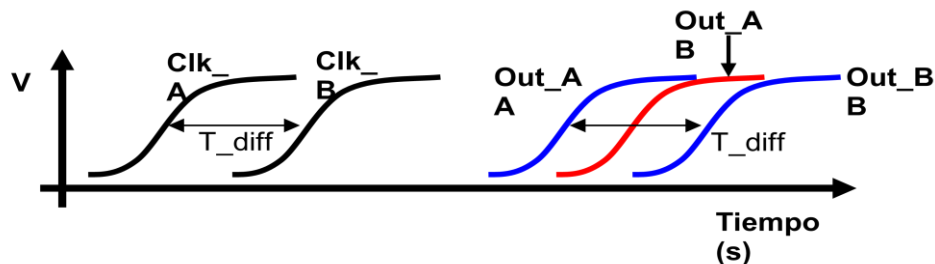


Ilustración 2 Respuesta de Señales Interpolador 2 a 3

De acuerdo a la figura que se muestra a continuación, se dice que dadas dos señales de reloj con la misma frecuencia (Clk_A y Clk_B), con un intervalo de tiempo de diferencia (T_{diff}) entre la transición de pulsos, que pasan a través de un interpolador se obtiene en la salida tres señales de reloj, dos de ellas con las mismas características que las de la entrada (Out_AA y Out_BB) reflejadas exactamente después de un retardo (T_{over}), es decir, la señal $Out_{AA} = T_{diff} + T_{over}$ y $Out_{BB} = T_{diff} + T_{over} + T_{diff}$, la tercera señal de reloj (Out_AB) está situada exactamente a la mitad del tiempo de diferencia de las señales OutAA y OutBB, $Out_{AB} = T_{diff} + T_{over} + (T_{diff} / 2)$, se dice que la señal Out_AB es una señal interpolada en fase debido que se encuentra determinada entre el rango de tiempo de Out_AA y Out_BB, este efecto lo podemos observar tanto como para flancos de subida como de bajada.

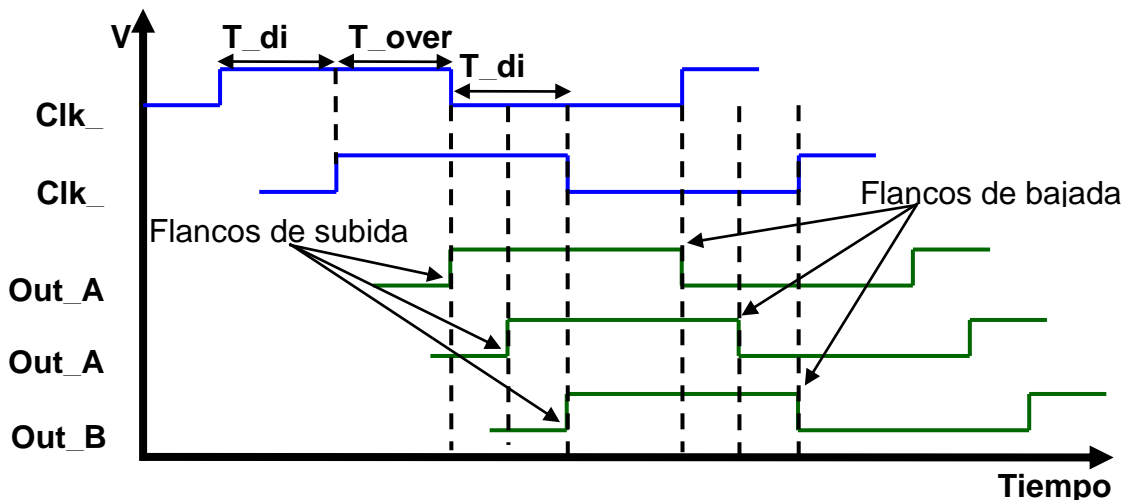


Ilustración 3 Respuesta de señales interpolador 2 a 3

DESCRIPCIÓN DEL PROYECTO:

En este proyecto se realizará el estudio, diseño e implementación de un circuito interpolador generador de señales a 100MHz en tecnología CMOS, la figura siguiente muestra la estructura del interpolador generador implementado, recibe ocho señales igualmente espaciadas en fase provenientes de un oscilador controlado por voltaje (VCO) en la primera etapa, así generando 32 señales con las mismas características en fase. Para obtener las diferencias de fases en las etapas de interpolación, se parte de que la frecuencia de operación es de 100Mhz, el periodo de esta frecuencia de operación es de

10ns, dividido entre 8 bloques generadores de rampa, de los cuales se obtienen 8 rampas con una diferencia de fase de 1.25ns en cada una de ellas. Estas 8 rampas entran a la primera etapa de interpolación, y como se explicó con anterioridad acerca del concepto de interpolación, se obtienen 16 señales de reloj con una diferencia de fase de 625ps entre cada una de ellas. Estas 16 señales de reloj entran a la segunda etapa de generación de rampa para que sea procesada en la segunda etapa interpoladora generándose así 32 señales de reloj con una diferencia de fase entre cada una de ellas de 312.5ps requeridos por la especificación de diseño.

CARACTERÍSTICAS, Y ESPECIFICACIONES DE DISEÑO

- Entradas del Interpolador:
 - 8 señales igualmente espaciadas en fase provenientes de un oscilador controlado por voltaje (VCO).
 - Frecuencia máxima de 100 MHz.
- Salidas del Interpolador:
 - 32 señales igualmente espaciadas en fase (312.5ps +/- 32ps entre sí).

Etapas del Interpolador

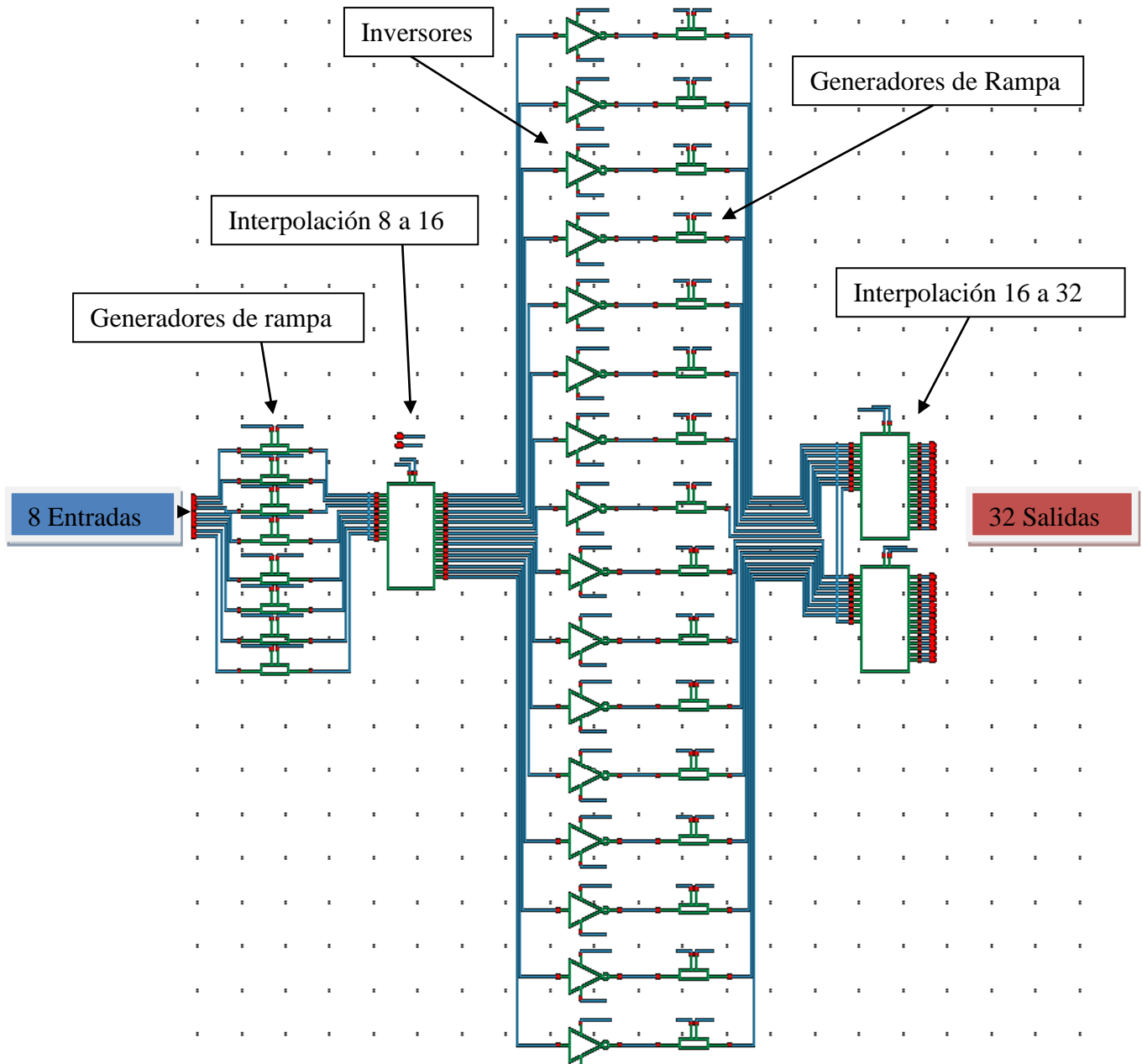


Ilustración 4 Etapas de Interpolación

Etapas 1 Generación de rampas

El circuito interpolador recibe 8 señales las cuales pasan en primer término por la etapa de generación de rampas esta etapa es necesaria para que coincida la interpolación de voltaje con la interpolación de tiempo, debido a que la señal interpolada pasa a través de compuertas que son rápidas, se requiere hacer lento el flanco de subida o bajada de las señales, según se requiera.

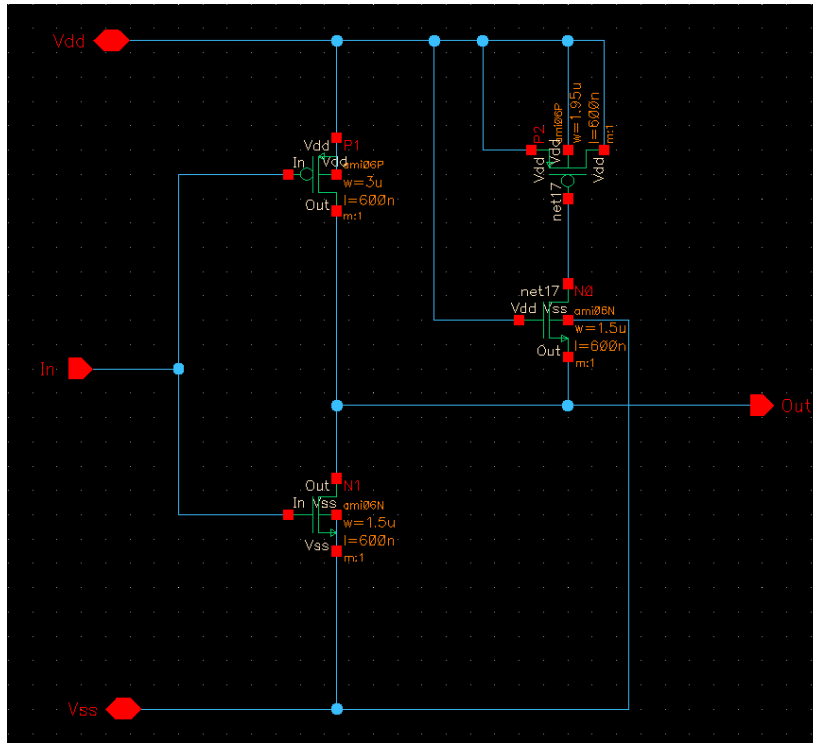


Ilustración 5 Celda de Generador de Rampas

Los dos primeros transistores (MP0 y MN0) conforman un inversor que entrega una señal cuadrada. El transistor MN1 se encuentra trabajando en la región lineal funcionando como resistencia, y el transistor MP1 está conectado como capacitor. Ambos forman un arreglo RC, el cual genera rampas en el flanco de subida y descarga exponencial en el flanco de bajada.

Interpolación

Esquemático de la CELDA DIGITAL BÁSICA DE INTERPOLACIÓN que se emplea en este proyecto:

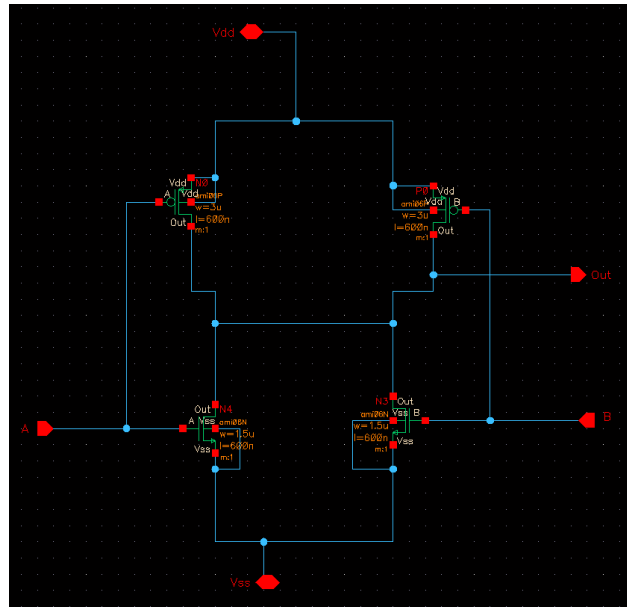


Ilustración 6 Celda Básica de Interpolación

Se implementó la celda básica de interpolación mostrada en la imagen anterior la cual conectada en serie como se muestra en la siguiente figura se pueden generar mayor cantidad de señales de salida.

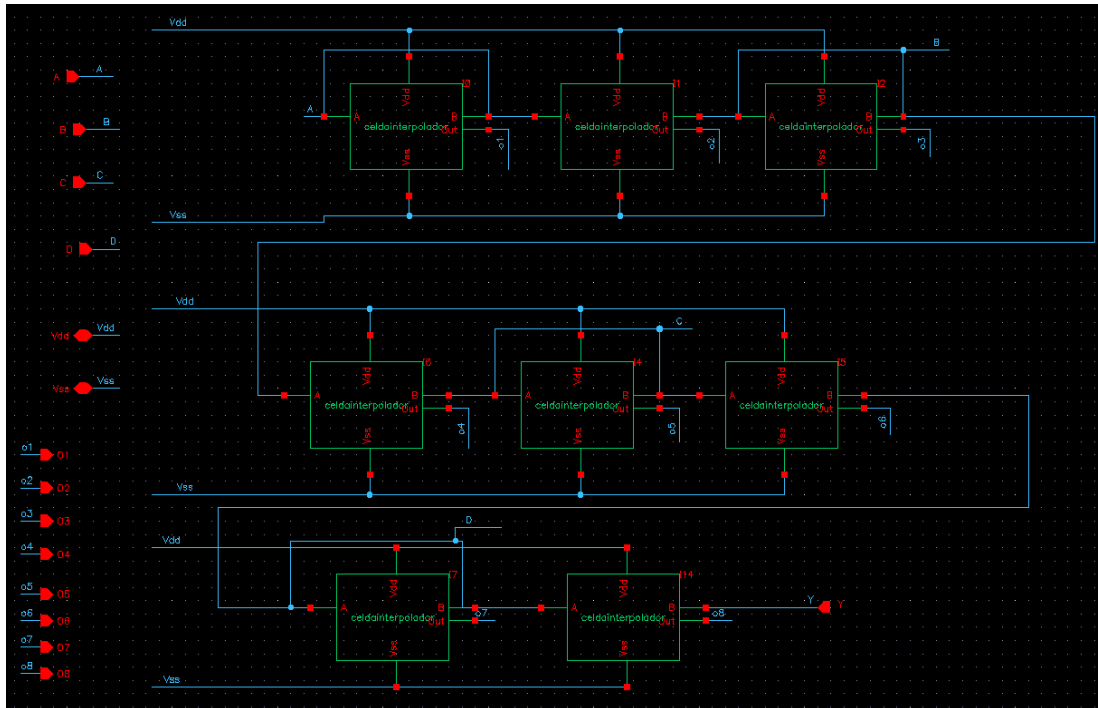


Ilustración 7 Interpolador de 4 a 8 señales

Partiendo de la celda básica y su conexión en serie se genero la Etapa 1 de interpolación 4x2 es decir 4 señales entrantes generando así 8 salidas, de la cual a su vez se podrán generar 16 salidas en la etapa 2 la cual se muestra en la siguiente imagen.

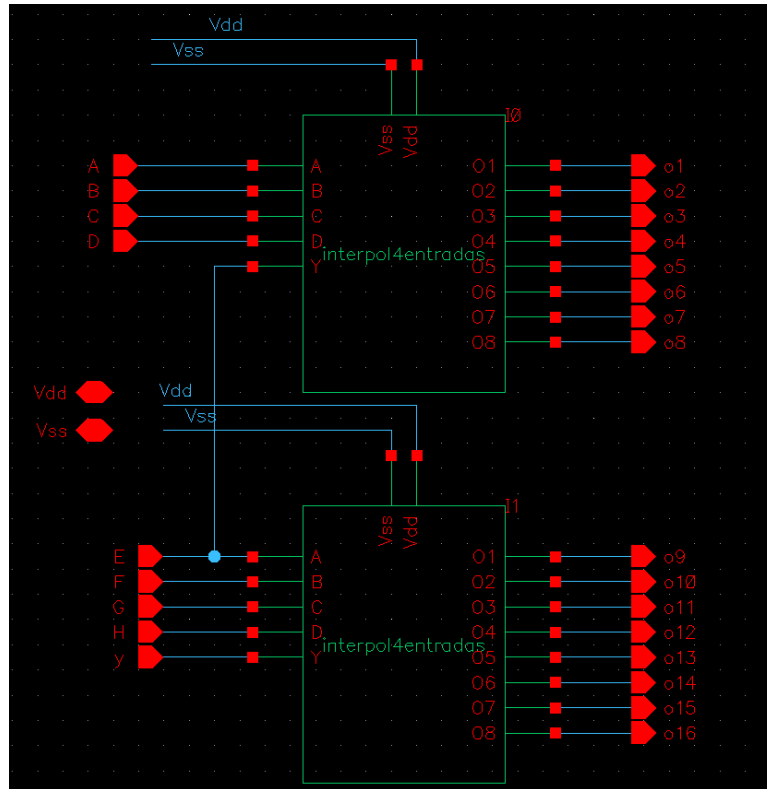


Ilustración 8 Etapa 2 Interpolación de 8 a 16 señales (8x2)

Cabe destacar que las celdas básicas utilizadas en el interpolador son conectadas en serie con la diferencia que un bloque recibe 2 señales idénticas en sus entradas y otra celda recibe señales diferentes, las cuales intercaladas y conectadas en serie como se dice con anterioridad nos permiten generar el interpolador.

Una vez pasada la primer etapa de interpolación y generadas 16 señales pasan nuevamente a una etapa de retardo y generación de rampas formado por inversores y generadores de rampa antes mencionados.

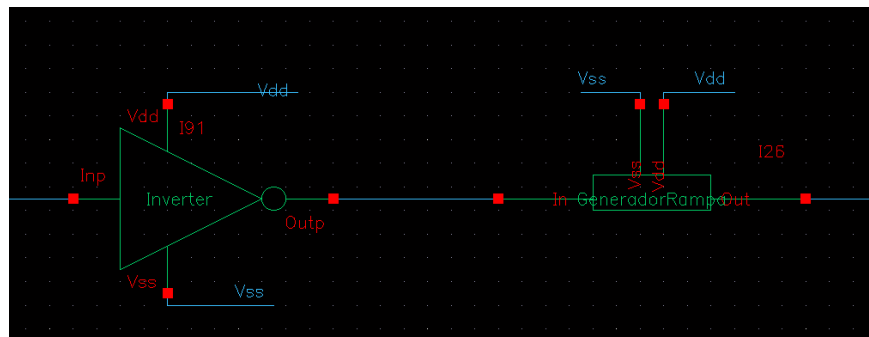


Ilustración 9 Segunda etapa de generación de rampas (retardo de flancos de subida)

Final mente pasan a la segunda etapa de interpolación que se encarga de generar 32 señales a partir de las 16 entrantes generadas por etapas anteriores.

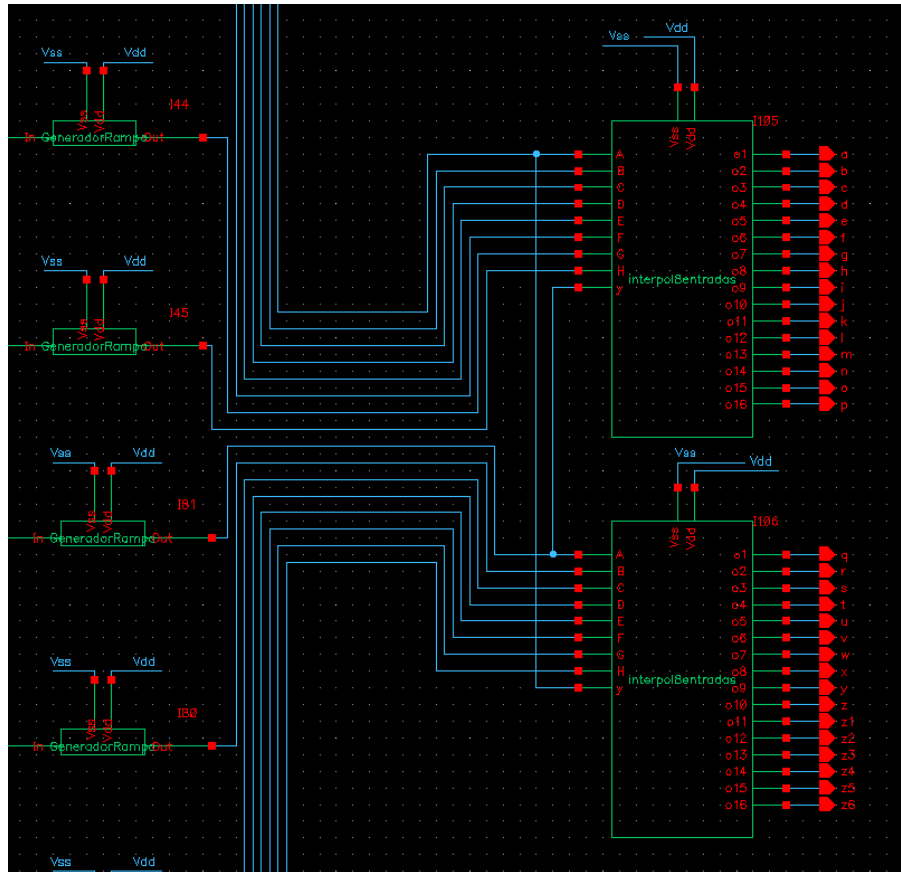


Ilustración 10 Segunda etapa de interpolación

Simulaciones Pre-layout

Para la simulación del interpolador a cada salida se le conecto un capacitor de 25fF, tal y como se ve en la imagen siguiente, para simular la carga que se puede conectar al circuito.

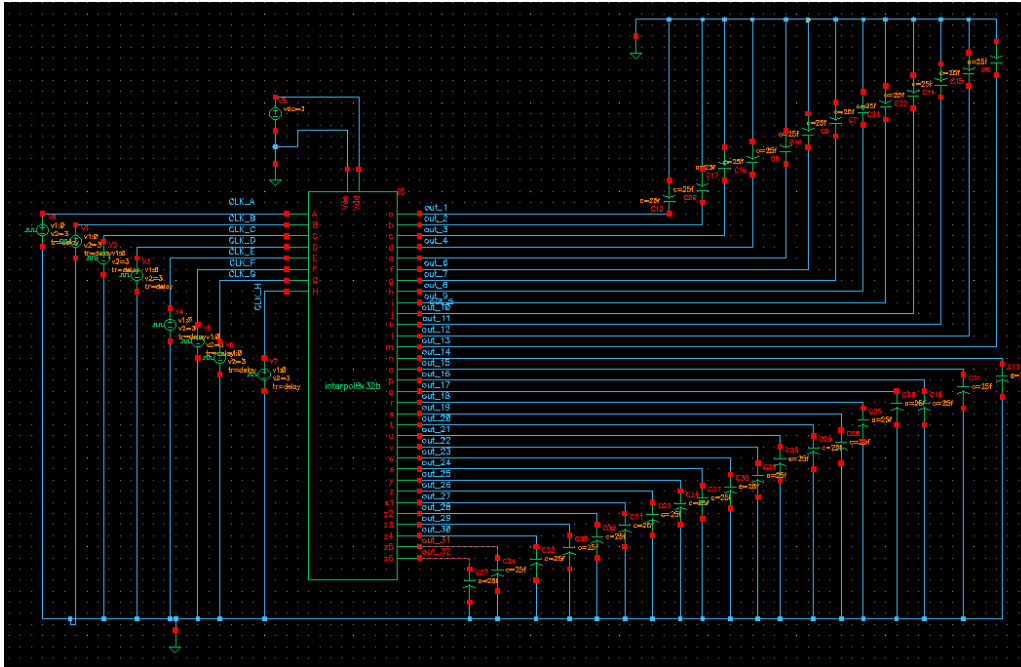


Ilustración 11: Diagrama esquemático para la simulación del interpolador.

A las entradas del circuito se le conectaron 8 fuentes de pulso, a una frecuencia de 100MHz, con 2ns de ancho de pulso y 10ns de periodo, una diferencia de fase de 1.25ns, esto último para que coincidan de forma cíclica: CLK_A, CLK_B, CLK_C, CLK_D, CLK_E, CLK_F, CLK_G, CLK_H, CLK_A, CLK_B, CLK_C, etc. Esto se observa en la ilustración siguiente.

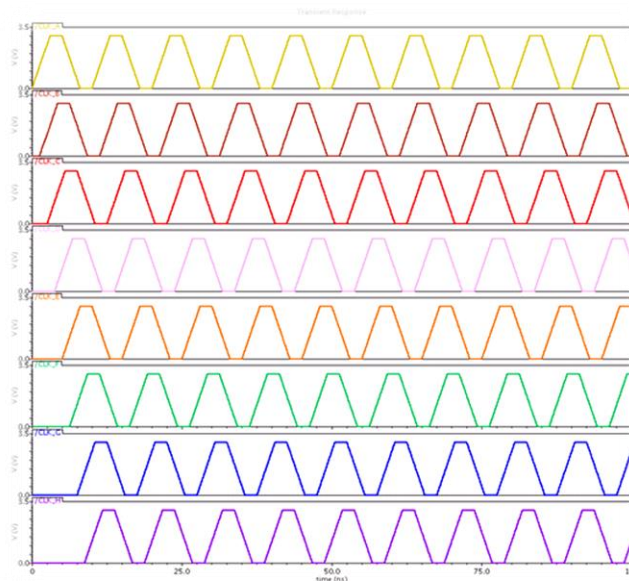


Ilustración 12: Pulsos de reloj a las entradas del interpolador

Se simulo la primera etapa de interpolación, esta etapa se muestra en la ilustración siguiente.

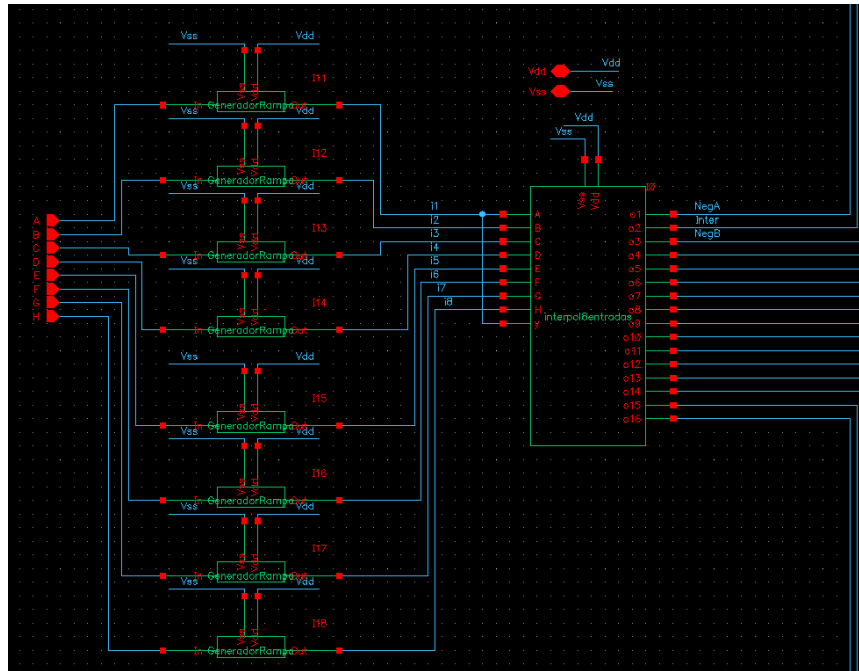


Ilustración 13: Primera etapa de interpolación.

Para esta primera etapa de interpolación se obtienen los resultados siguientes, en primer lugar las entradas A y B se ponen las señales de reloj descritas anteriormente, a la salida del circuito generador de rampa se tienen las señales siguientes tal y como se muestra en la imagen siguiente:

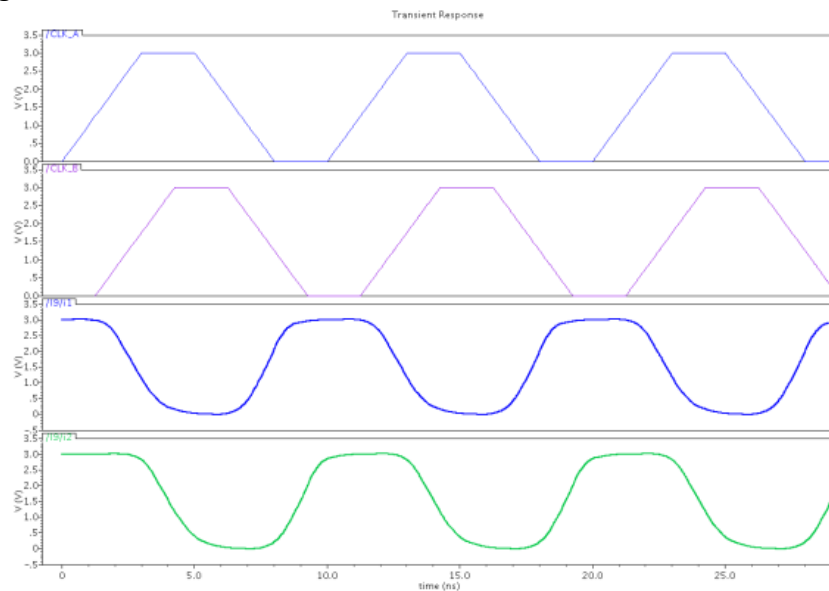


Ilustración 14: Señales de entrada y salidas de la primera etapa de rampa

Estas señales son el resultado del circuito generador de rampas, estas señales salen negadas, eso se observa en la imagen siguiente:

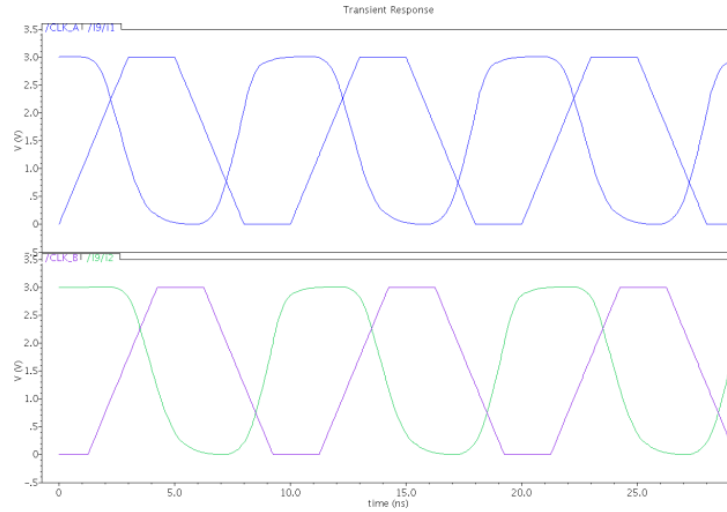


Ilustración 15: Comparación entre las señales de entrada y las salidas de circuito generador de rampa.

Las señales de las salidas del circuito generador de rampa se convierten en las entradas del interpolador, por lo tanto son las señales a interpolar, por lo que a las salidas del interpolador se tiene lo siguiente:

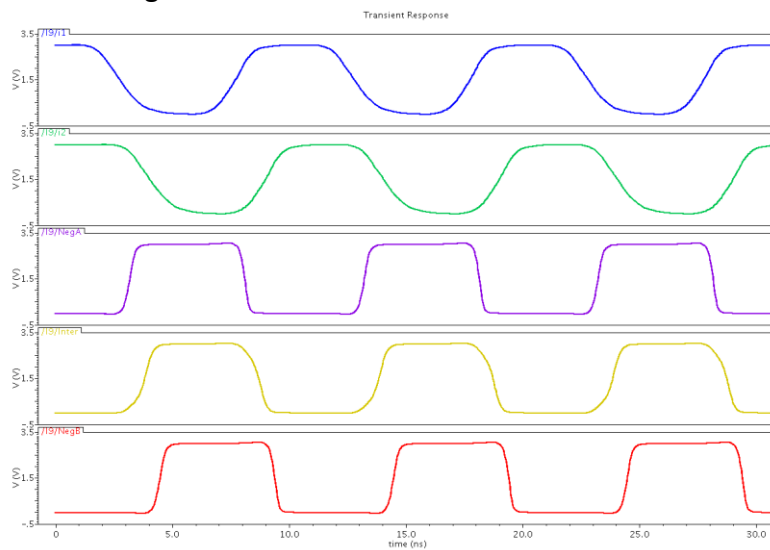


Ilustración 16: señales de entrada y Señales interpoladas

Donde la señal NegA es la señal negada de la entrada, NegB es la señal negada de la segunda entrada e inter es la señal interpolada, tal y como se ve en la imagen siguiente:

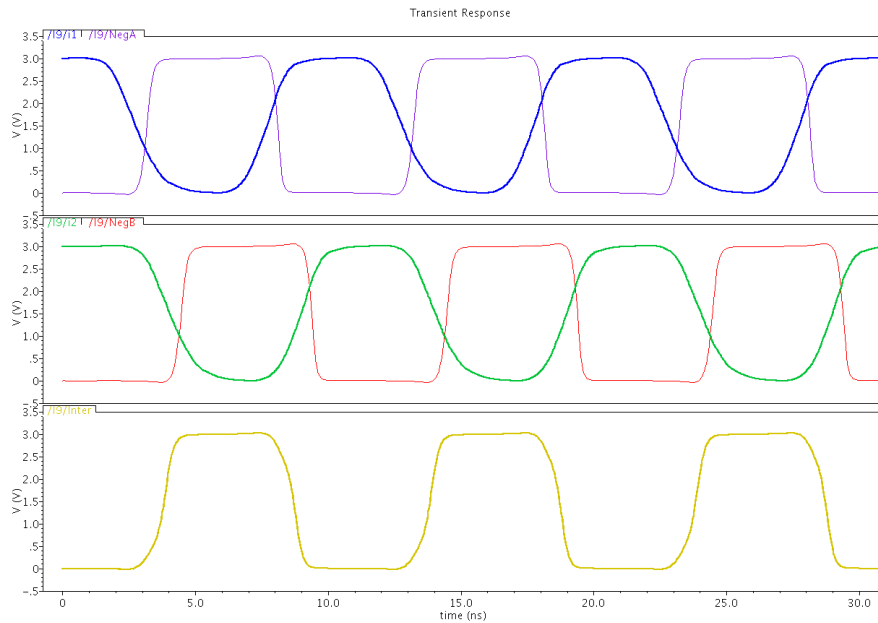


Ilustración 17: Interpolación de 2 señales

Por lo tanto las salidas del la primera etapa del interpolador son las señales que se muestran a continuación, estas señales están igualmente espaciadas tienen una diferencia de fase de 625ps, estas señales se **convierten en las entras de la segunda etapa de interpolación.**

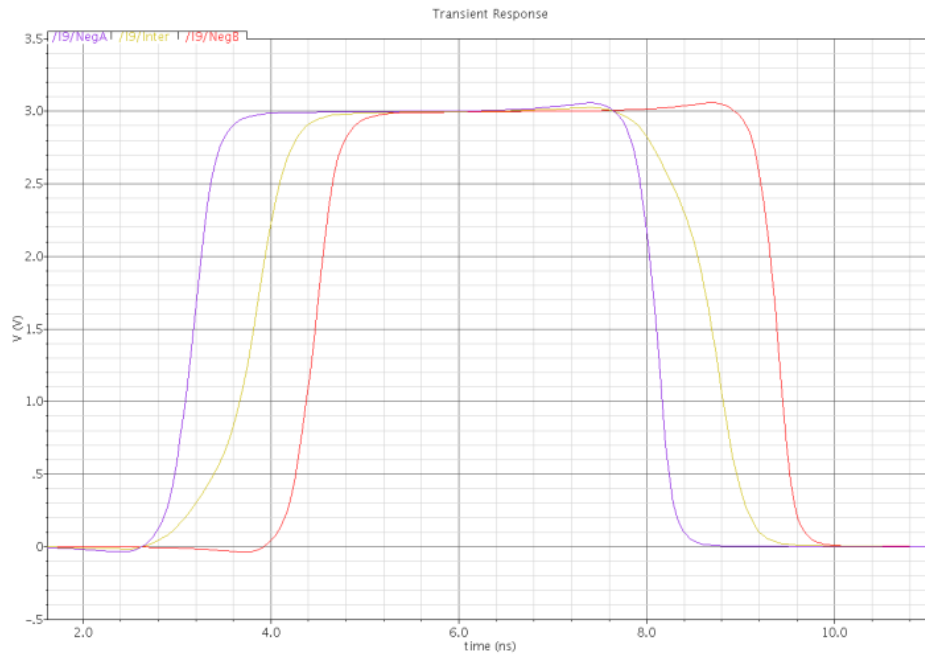


Ilustración 18: Señales interpoladas

De manera general las señales que se obtienen a la salida del interpolador como respuesta a las 8 entradas son las 32 señales reloj con una diferencia de fase entre cada una de ellas de 312.5ps. En la imagen que se presenta a continuación se muestran estas señales en la etapa pre-Layout.

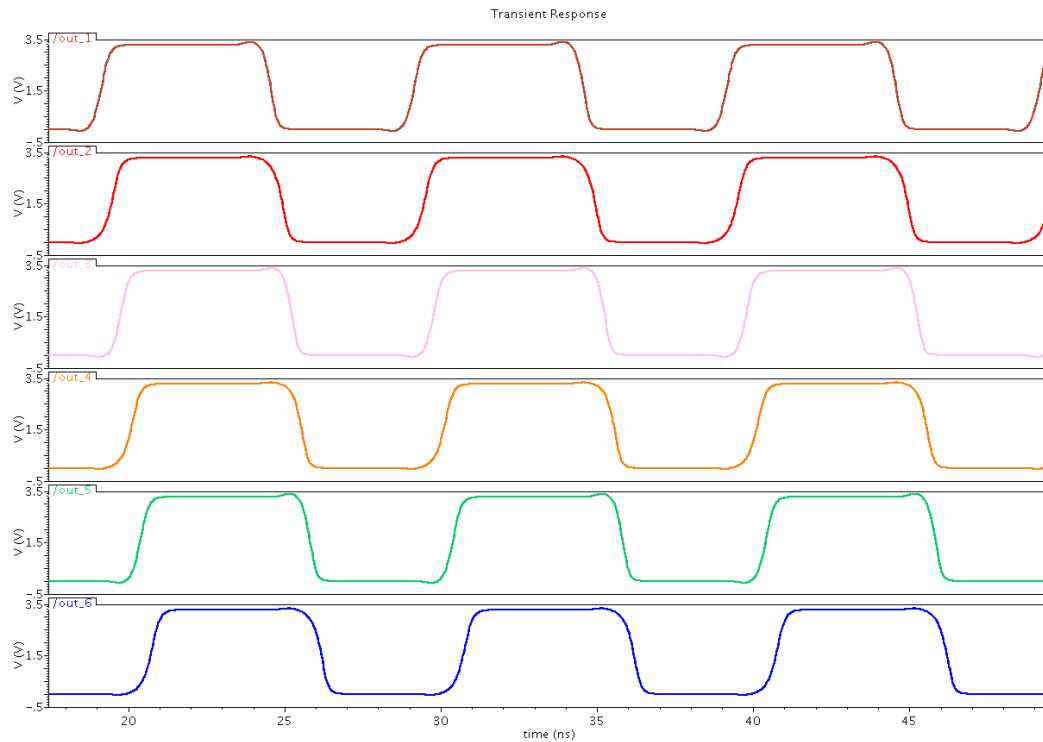


Ilustración 19: Señales de salida del interpolador 1-6

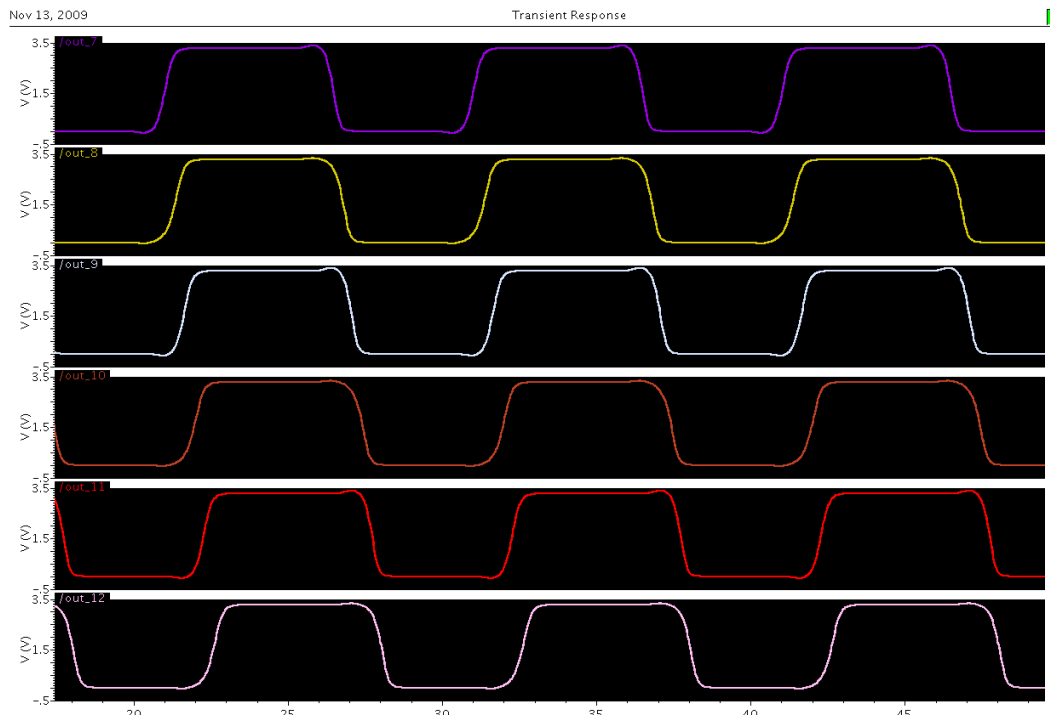


Ilustración 20: Señales de salida del interpolador 7-12

En la diferencia de fase de las señales en la etapa de pre – Layout se obtuvieron los resultados que se presentan en la tabla siguiente:

Tabla 1: Medición de las fases de Pre Layout

FASE	DIFF FASE Pre- Layout	Error =312- fase
1->2	330.4	-17.9
2->3	295.3	17.2
3->4	329.7	-17.2
4->5	294.6	17.9
5->6	330.2	-17.7
6->7	295.6	16.9
7->8	329.6	-17.1
8->9	294.6	17.9
9->10	330.4	-17.9
10->11	295.3	17.2
11->12	329.7	-17.2
12->13	294.6	17.9
13->14	330.2	-17.7
14->15	295.6	16.9
15->16	329.6	-17.1
16->17	294.6	17.9
17->18	330.4	-17.9
18->19	295.3	17.2
19->20	329.7	-17.2
20->21	294.6	17.9
21->22	330.2	-17.7
22->23	295.6	16.9
23->24	329.6	-17.1
24->25	294.6	17.9
25->26	330.4	-17.9
26->27	295.3	17.2
27->28	329.7	-17.2
28->29	294.6	17.9
29->30	330.2	-17.7
30->31	295.6	16.9
31->32	329.6	-17.1
32->1	294.6	17.9

En la tabla anterior se puede observar en la primera columna (FASE), la relación entre las señales, que se van a medir, en la segunda columna se observa medición de las diferencias de fase entre las señales, para este trabajo se tiene la diferencia de fase debe ser de 312.0ps, con una tolerancia de +/- 32.0ps, por lo tanto se calcula el error entre el obtenido como resultado pre layout, y el deseado este error se muestra en la tercera columna de la tabla 1.

En la gráfica anterior se muestra la variación de las diferencias de fase, se observa que una es mayor que 312.0ps, la siguiente menor que 312.0ps, una después a esa es mayor a 312.0ps y así sucesivamente, por lo que se van compensando una con otra para mantener todas las señales dentro de un rango de 312.0 +/- 17.9ps.

Por lo tanto se puede concluir que el circuito en esquemático funciona correctamente y se puede continuar con la elaboración de layout de este circuito.

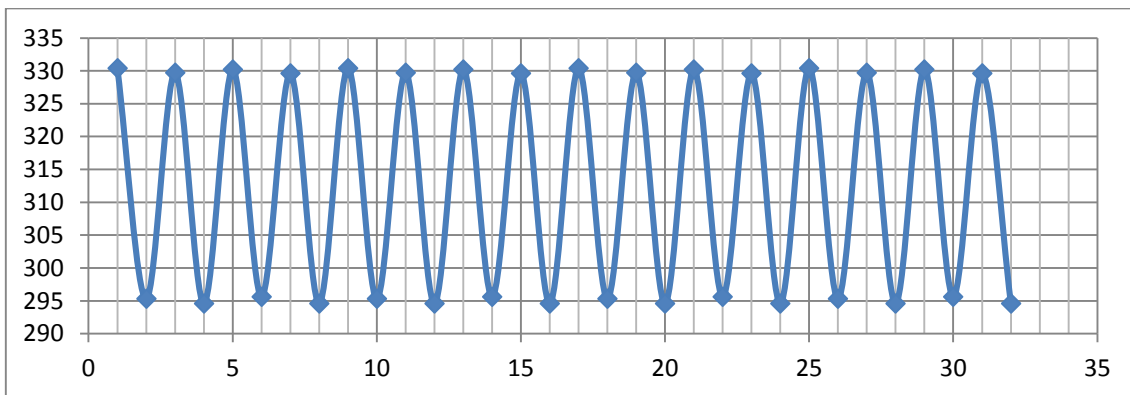


Ilustración 21: Grafica de Diferencias de fase Pre Layout

Las siguientes imágenes muestran los layouts de las celdas básicas utilizadas además de las 2 propuestas completas de layout para este interpolador.

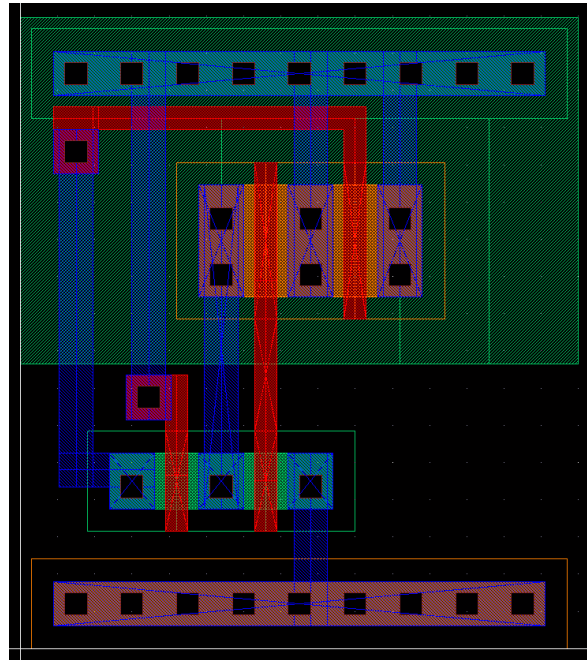


Ilustración 22 Layout Celda generadora de Rampas

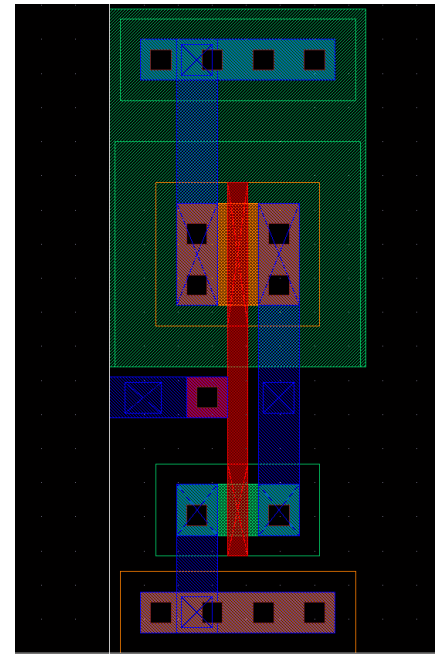
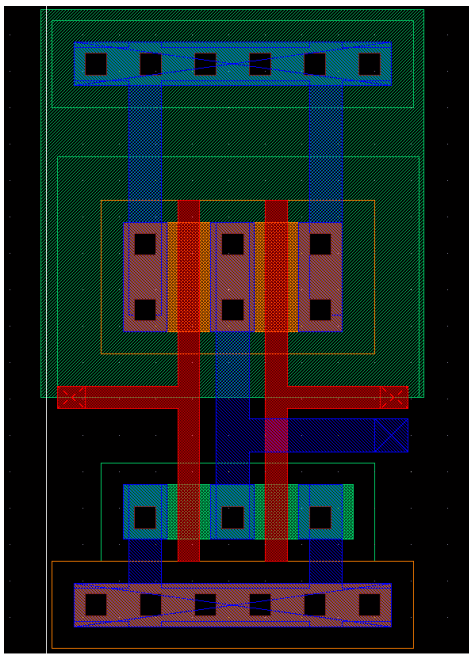


Ilustración 23 Layout de Celda Básica de Interpolación, e Inversor.

Propuesta Layout 1

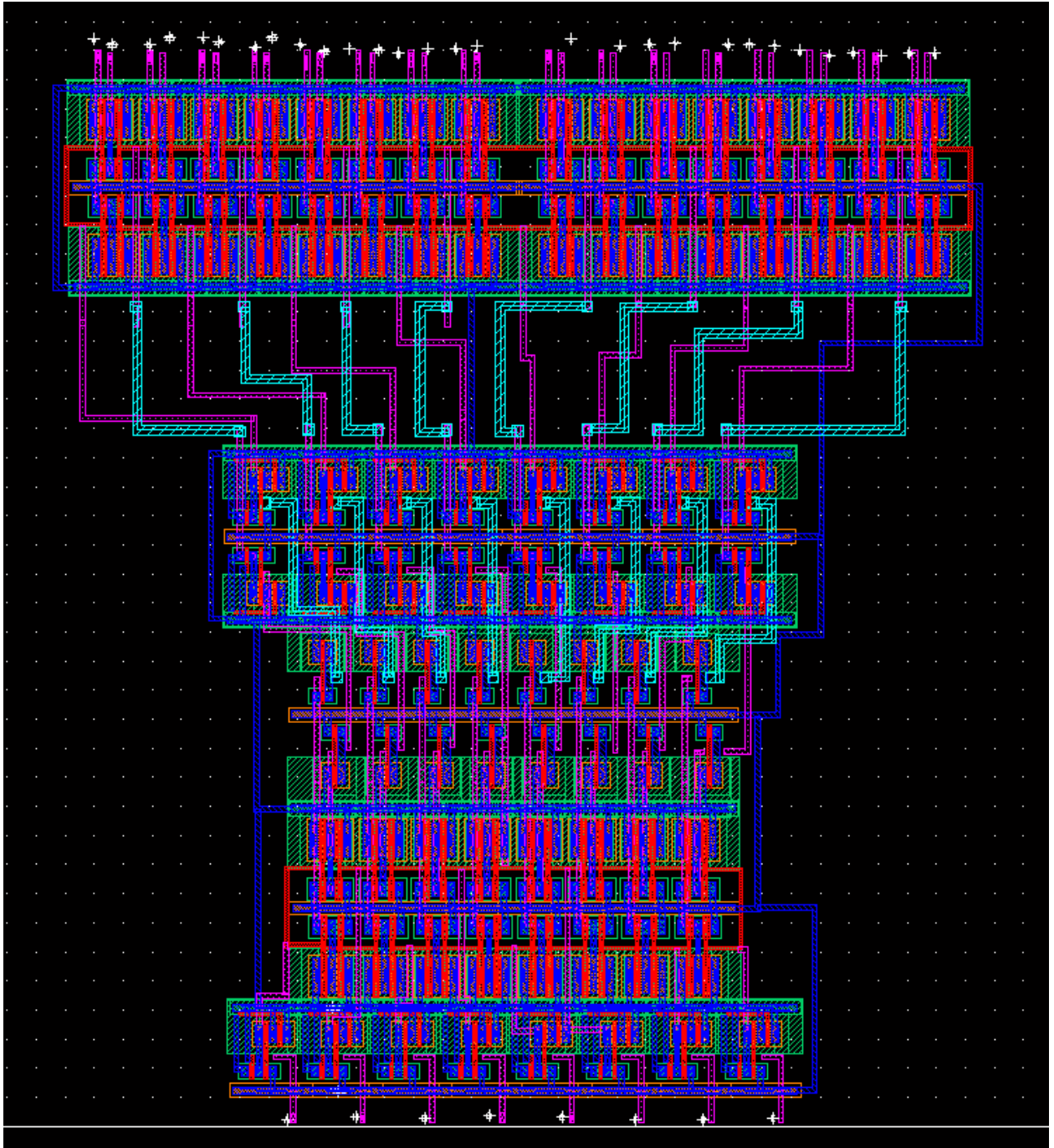


Ilustración 24 Layout 1 interpolador 8 – 32 ((159.50um x 184.3um)

Propuesta Layout 2

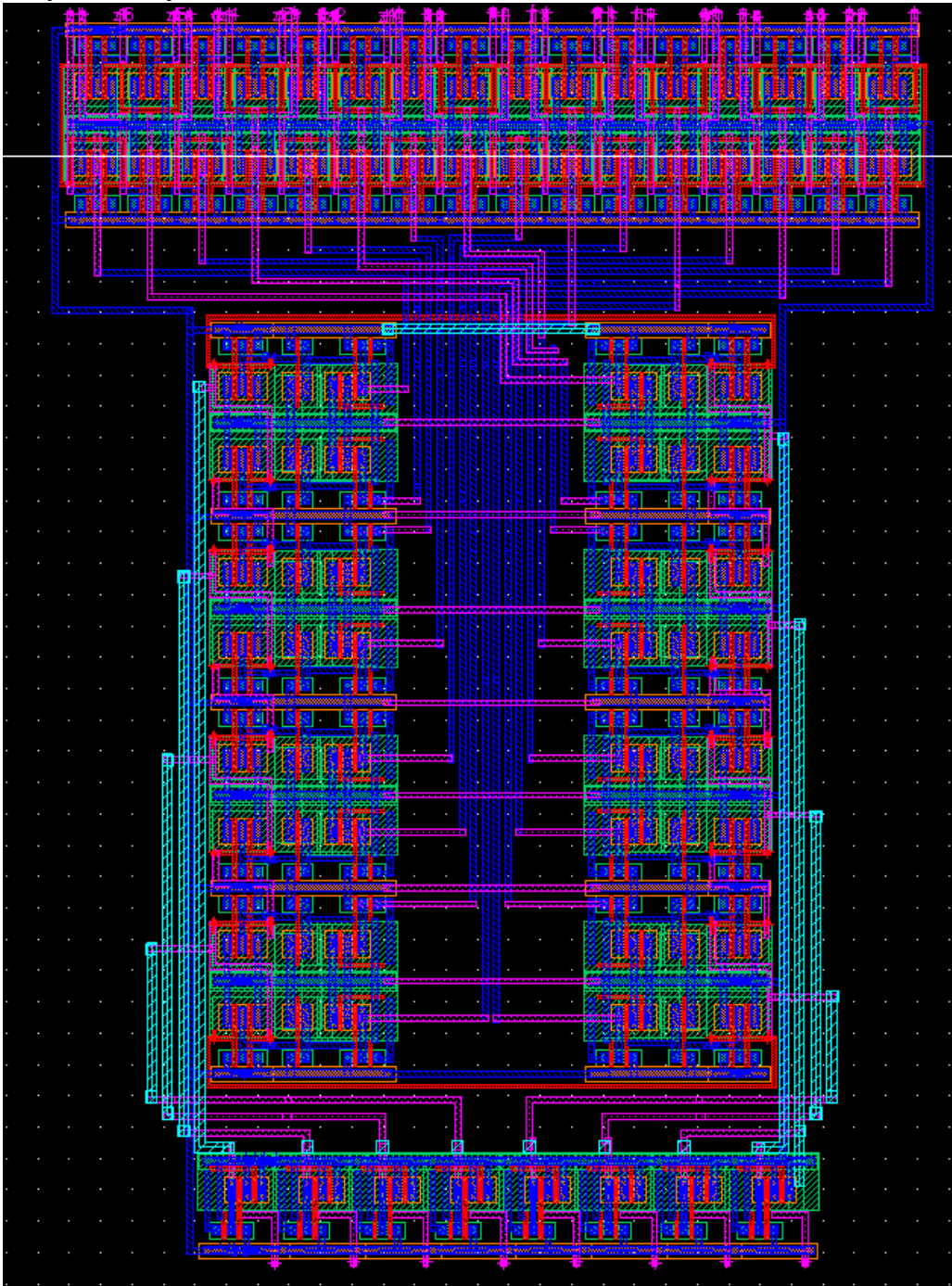


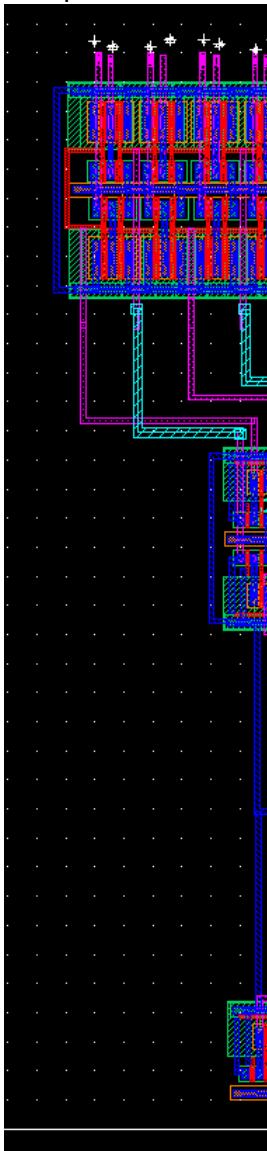
Ilustración 25 Propuesta de Layout 2 para interpolador 8 – 32. Dimensiones (139.35 um x 200.25 um)

1. Simulaciones Post – Layout

Una vez generados los layouts se procedió hacer la simulación de los mismos, además de realizar las correcciones pertinentes en el ruteo encaminados a obtener un buen

desempeño del interpolador donde las señales obtenidas tengan una diferencia entre fases de 312.5 ps o estén dentro del rango de tolerancia de 312.5 +/- 32ps.

En el primer layout se observan las etapas del interpolador de abajo hacia arriba: Primera etapa generadora de rampa, Primera etapa de interpolación 8 x 16, Primera etapa de inversión de señal, Segunda etapa generadora de rampa, Segunda etapa de interpolación 16x32.



FASE	DIFF FASE POST LAYOUT 1 Sin Corregir	Error =312-fase t(+/-32)
1->2	331.5	-19
2->3	287.6	24.9
3->4	317.5	-5

Ilustración 26: layout 1 Interpolador 8 x 32 (159.50um x 184.3um) En la diferencia de fase de las señales en la etapa de POST – Layout 1 se obtuvieron los resultados que se presentan en la tabla siguiente

4->5	274.7	37.8
5->6	325.1	-12.6
6->7	280.3	32.2
7->8	346.3	-33.8
8->9	300	12.5
9->10	347	-34.5
10->11	290.9	21.6
11->12	338.2	-25.7
12->13	276	36.5
13->14	350	-37.5
14->15	296.9	15.6
15->16	348	-35.5
16->17	256	56.5
17->18	355	-42.5
18->19	311	1.5
19->20	266	46.5
20->21	267.3	45.2
21->22	367	-54.5
22->23	324	-11.5
23->24	347	-34.5
24->25	247	65.5
25->26	306	6.5
26->27	280	32.5
27->28	362	-49.5
28->29	311	1.5
29->30	388	-75.5
30->31	271	41.5
31->32	346.1	-33.6
32->1	323.1	-10.6

Tabla 2: Medición de las fases de Post Layout 1 (ps)

De la tabla anterior se puede observar en la segunda columna, que las mediciones de las fases sin corregir el layout, de esta columna se observa las fases tienen mucha variación entre las diferencias de fase, esto se ve con mucha mayor claridad en la gráfica siguiente:

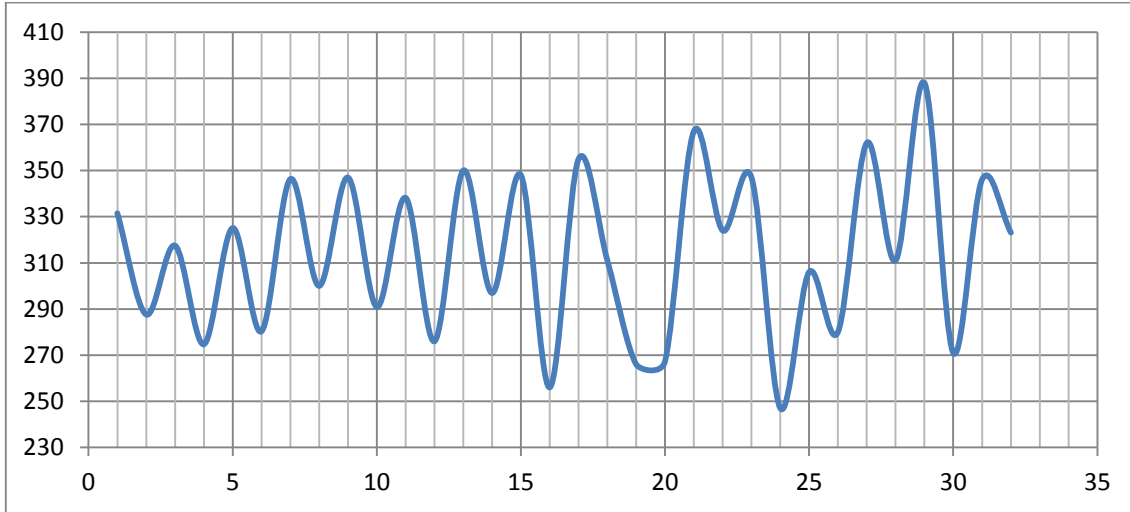


Ilustración 27: Grafica de Diferencias de fase Post Layout sin correcciones.

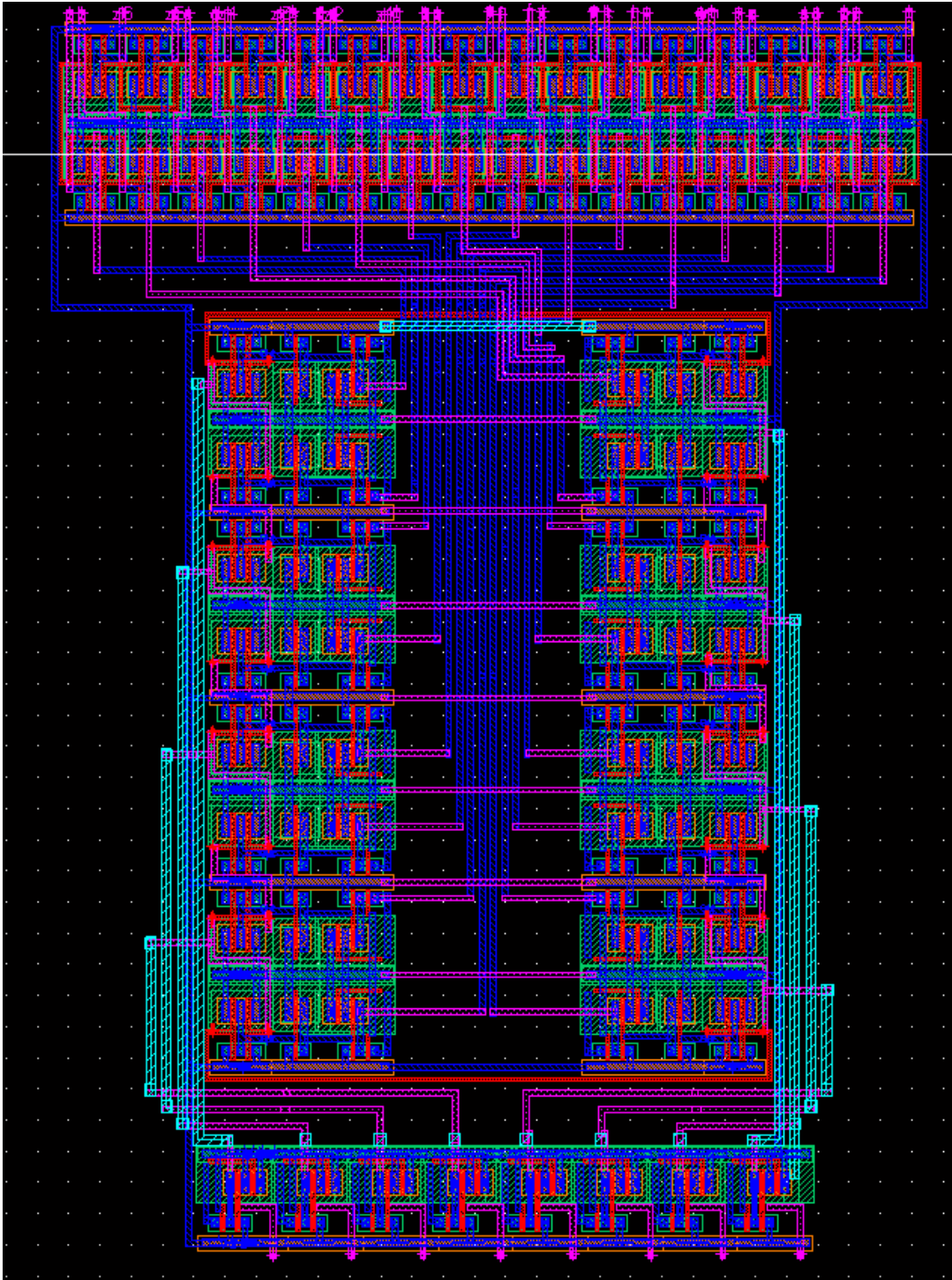


Ilustración 28 Propuesta de Layout 2 para interpolador 8 – 32. Dimensiones (139.35 μm x 200.25 μm)

La imagen muestra el Layout completo del interpolador cuyos resultados de simulación se muestran en la siguiente tabla.

La siguiente tabla muestra los resultados ppst layout 2 donde se muestra el error de de fase entre las señales las cuales tienen una tolerancia de (+/-32 ps)

Tabla 3 Resultados simulación post layout 2

FASE	DIFF FASE POST LAYOUT 1 Sin Corregir	Error =312-fase T(+/-32)
1->2	406	-94
2->3	321	-9
3->4	280	32
4->5	261	51
5->6	351	-39
6->7	265	47
7->8	324	-12
8->9	343	-31
9->10	393	-81
10->11	325	-13
11->12	340	-28
12->13	349	-37
13->14	375	-63
14->15	305	7
15->16	322	-10
16->17	261	51
17->18	312	0
18->19	224	88
19->20	271	41
20->21	252	60
21->22	354	-42
22->23	249	63
23->24	305	7
24->25	305	7
25->26	376	-64
26->27	252	60
27->28	358	-46
28->29	366	-54
29->30	309	3
30->31	206	106
31->32	278	34
32->1	283	29

Los datos en color verde son las señales cuya diferencia de fase está dentro de lo requerido, se observa que la mayoría de las señales no están dentro de los requerimientos por lo que se tendrán que ajustar las señales.

Los datos en color rojo son las señales cuya diferencia de fase está fuera de lo requerido, se observa que la mayoría de las señales no están dentro de los requerimientos por lo que se tendrán que ajustar las señales.

Ajuste de Layouts

Debido a que la mayoría de las fases no se encuentra dentro del rango de especificaciones (312.0ps +/-32.0ps), se procedió a mejorar las diferencias de las fases por medio de técnicas de layout y de ruteo de señales en el layout.

Para poder hacer que las diferencias de fase estuvieran dentro del rango de especificaciones, primero se debe entender el layout, en la imagen siguiente se muestra el layout 1 del interpolador dividiendo a este en las siguientes partes:

- La primera etapa generadora de rampas.
- La primera etapa de interpolación (8-16).
- La primera etapa inversora
- La segunda etapa de generación de rampas.
- La segunda etapa de interpolación (16-32).

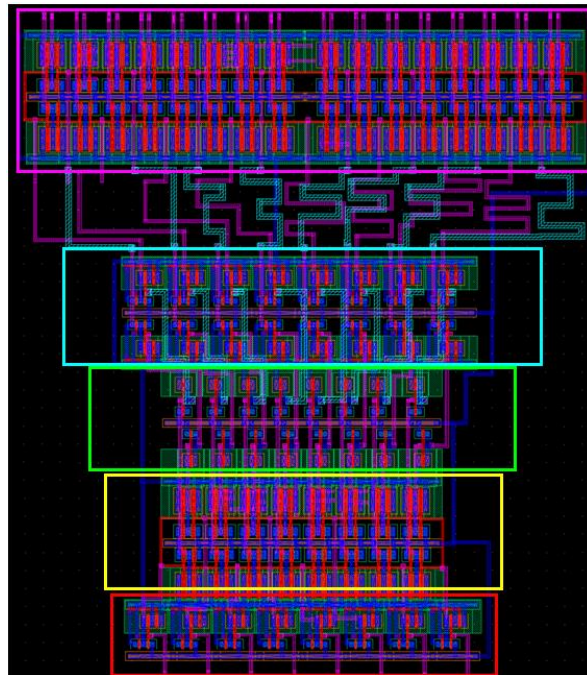


Ilustración 29: Layout del Interpolador

Después de identificar las partes del layout se analizan los resultados obtenidos y mostrados en la tabla 2, de ella se clasifican los datos de la siguiente forma:

- Datos que están por arriba del valor esperado más el valor de tolerancia (312.0ps + 32.0ps), se muestran en color azul.
- Datos que están por abajo del valor esperado menos el valor de tolerancia (312.0ps -32.0ps), se muestran en color rojo.

- Datos que están entre del valor esperado más o menos el valor de tolerancia (312.0ps +/- 32.0ps), se muestran en color verde.

Por lo tanto de la grafica se pueden observar los siguientes casos:

A) Cuando esta una Diferenciad de fase por debajo del valor esperado, entre dos que están en un valor correcto:

B) Cuando esta una Diferenciad de fase por arriba del valor esperado, entre dos que están en un valor correcto:

3->4	317.5
4->5	274.7
5->6	325.1

C) Cuando la diferencia de fase está por debajo del valor esperado, entre dos que están por arriba del valor esperado.

8->9	300
9->10	347
10->11	290.9

29->30	388
30->31	271
31->32	346.1

Para corregir cada uno de los casos anteriores, es necesario entender que estos resultados son debido a que las señales son ruteadas con distancias diferentes, en el caso A, donde se tiene una señal atrasada, esta señal necesita recorrer una mayor distancia que la que recorre en el momento de la medición. En el caso B se tiene una señal adelantada, pero como no es posible hacer que la señal recorra una menor distancia, entonces se procede a adelantar las señales que están a su alrededor, para que la diferencia de fase entre estas señales llegue al valor esperado. En el caso C, donde se tiene una señal atrasada entre dos adelantadas, solamente bastara con hacer que la señal atrasada recorra una mayor distancia, para que se adelante.

Para hacer que una señal recorra una mayor distancia se pueden utilizar muchas técnicas, en este proyecto se utilizaron 2 principalmente, la primera es la de hacer las líneas más largas haciendo una especie de serpiente, como se ve en la imagen siguiente, esto con el fin de que la señal tarde más tiempo en llegar (retardar la señal).

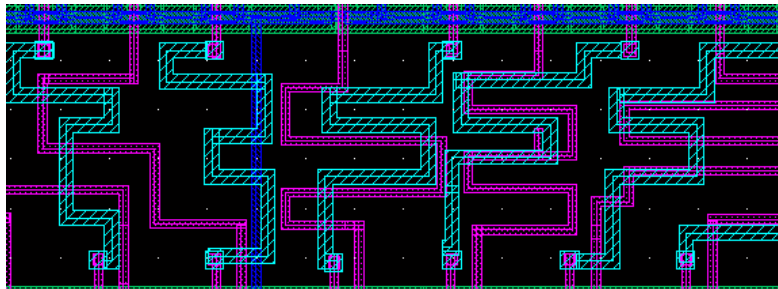


Ilustración 30: Técnicas de ruteo, Serpentina.

La segunda técnica que empleamos para retardar la señal, es agregando un poco más de metal a la línea por donde pasa la señal que se quiere adelantar, un ejemplo de esto se muestra en la imagen siguiente.

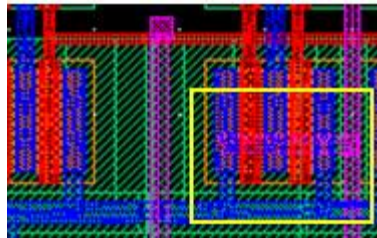


Ilustración 31: Técnicas de ruteo, Se agrega metal.

Todo lo anterior se realizó con cada señal para hacer que todas las señales entraran al rango que se estableció de 312.0 ps +/- 32.0 ps, con esto se obtuvieron las mediciones que se muestran en la tabla siguiente, en ella se puede observar que todas las señales están dentro del rango establecido en el Layout 1 .

Tabla 4: Diferencias de fase después de corregir el layout 1

FASE	DIFF FASE POST LAYOUT 1 Corregido	Error =312-fase	POST LAYOUT 2 Corregido	Error =312-fase
1->2	332.6	-20.1	330	-18
2->3	292.2	20.3	293	19
3->4	327.7	-15.2	306.9	5.1
4->5	283.6	28.9	311.1	0.9
5->6	341	-28.5	336.7	-24.7
6->7	307.9	4.6	254.4	57.6
7->8	343.4	-30.9	339.5	-27.5
8->9	323	-10.5	331.5	-19.5
9->10	341.1	-28.6	364.9	-52.9
10->11	293.4	19.1	342.8	-30.8
11->12	335.6	-23.1	343.3	-31.3
12->13	289.8	22.7	393.3	-81.3
13->14	341.1	-28.6	324.7	-12.7
14->15	289.4	23.1	308.2	3.8
15->16	338.3	-25.8	326.7	-14.7
16->17	288.7	23.8	285.5	26.5
17->18	341.4	-28.9	289.6	22.4
18->19	282.2	30.3	214.9	97.1
19->20	334.2	-21.7	283.3	28.7
20->21	281.9	30.6	280.8	31.2
21->22	314.5	-2	344	-32
22->23	292.3	20.2	299.5	12.5
23->24	333.3	-20.8	333.2	-21.2
24->25	286.2	26.3	332.5	-20.5
25->26	294.1	18.4	339.6	-27.6
26->27	283.3	29.2	280.9	31.1
27->28	344	-31.5	342.6	-30.6
28->29	292	20.5	301.4	10.6
29->30	339.4	-26.9	327	-15
30->31	302.7	9.8	228	84
31->32	328.6	-16.1	301	11
32->1	318.8	-6.3	332	-20

En la segunda columna de la tabla anterior se muestra los valores de las diferencias de fases des pues de que se corrigió el layout, se observa que todas las señales están dentro del rango establecido, en la grafica siguiente se puede ver que ya no hay tanta variación como al principio, por lo que este layout es funcional.

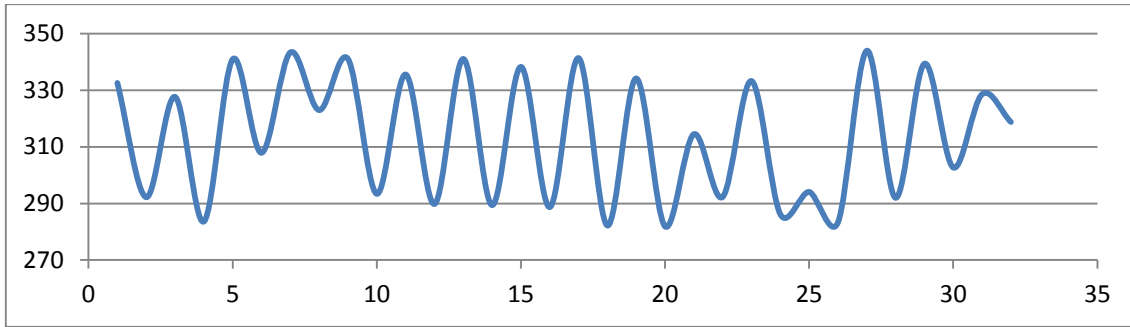


Ilustración 32: Grafica de Diferencias de fase Post Layout 1 con correcciones

El segundo layout tuvo mayores problemas ya que sus señales no estaban dentro de lo establecido, se utilizaron las técnicas antes mencionadas y se logró ajustar la mayoría de las señales quedando solo 5 fuera de los specs.

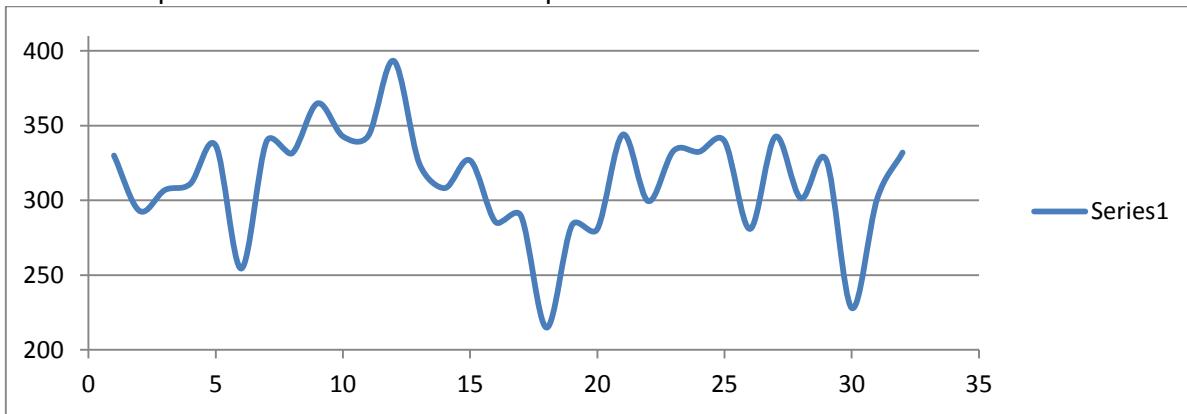


Ilustración 33 Grafica de diferencias de fases post layout 2 con correcciones

Layouts Corregidos

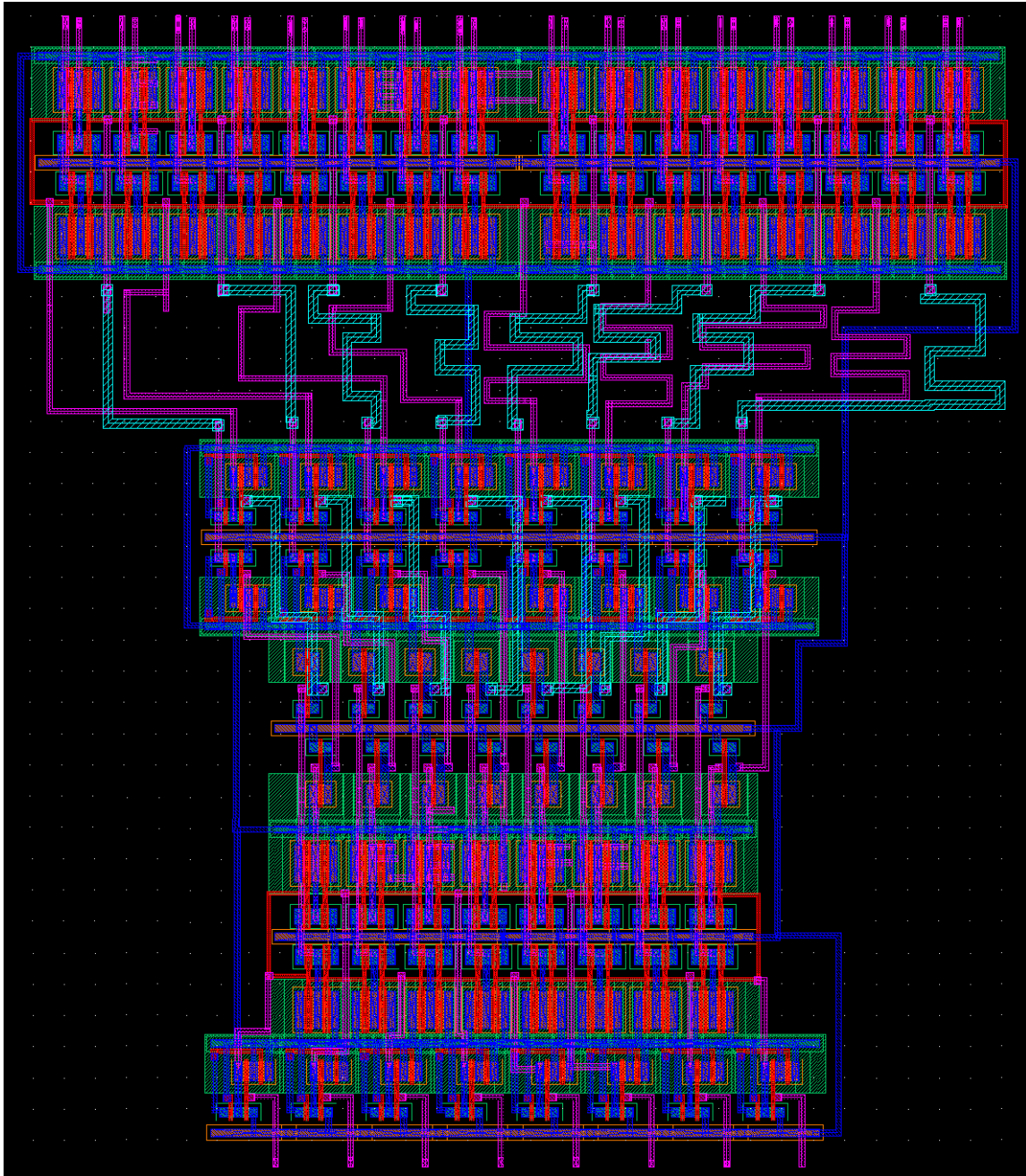


Ilustración 34 Layout 1 con compensaciones para las señales

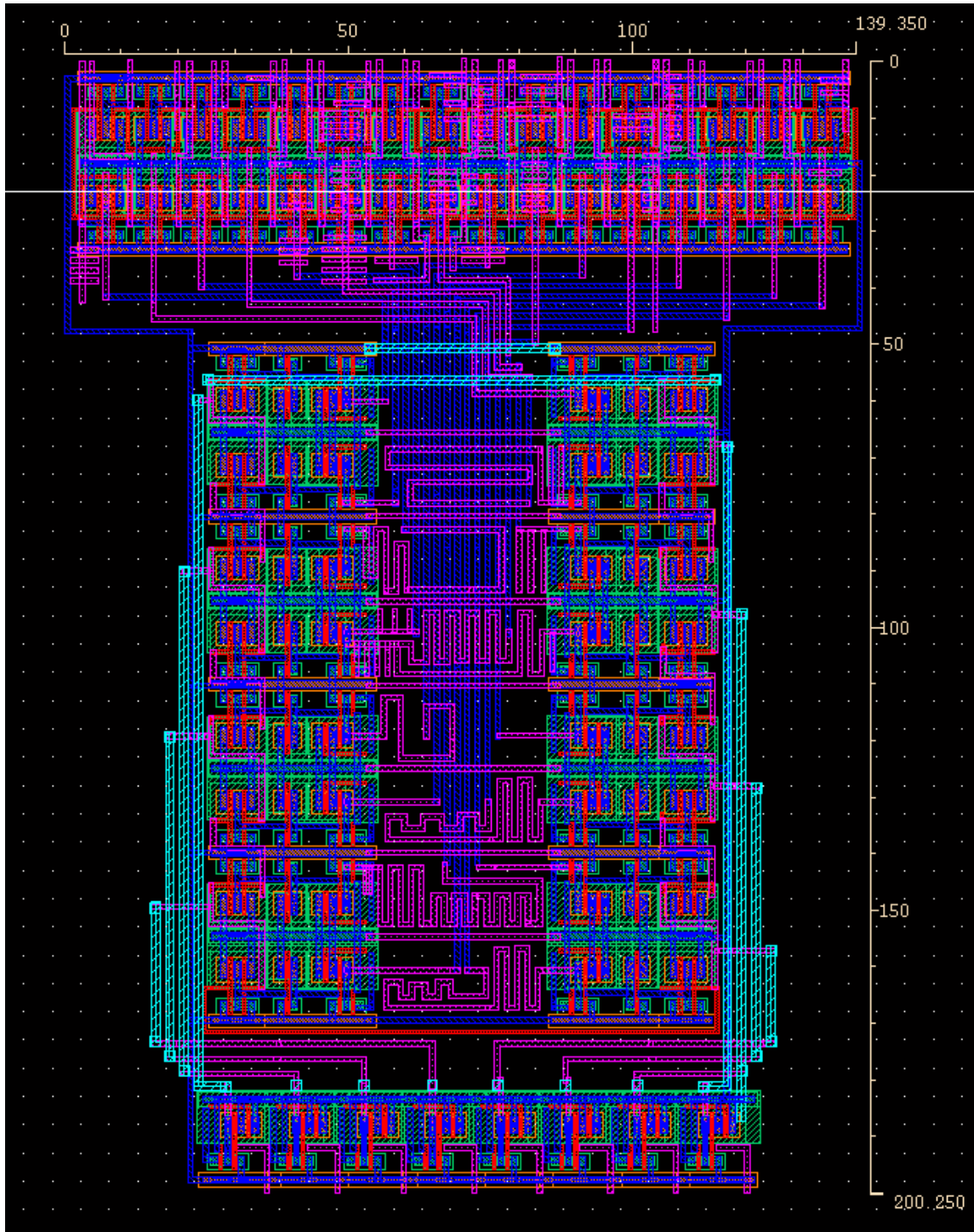


Ilustración 35 Layout 2 con compensaciones para señales

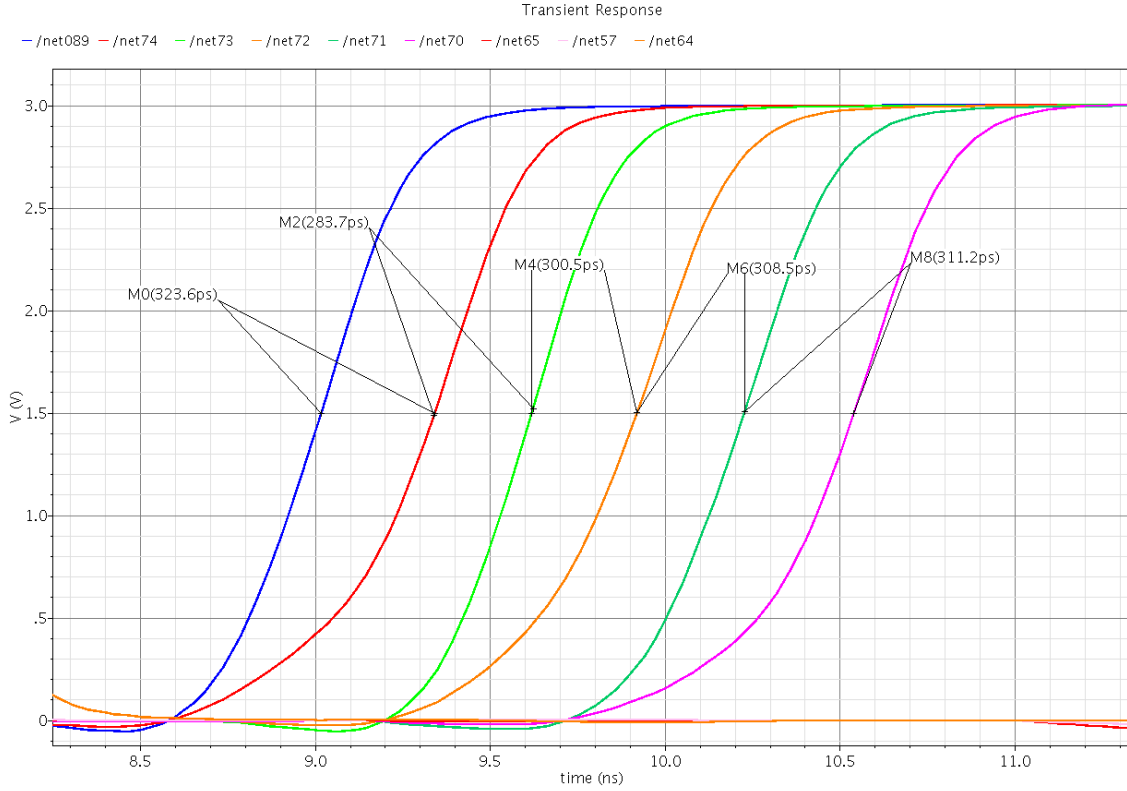


Ilustración 36 6 de 32 señales del interpolador ya compensadas

Los resultados de la simulación post – layout, son los que se muestran en las imágenes siguientes, se observa 32 señales de reloj con una diferencia de fase de 312.5ps con una tolerancia de +32.0ps.

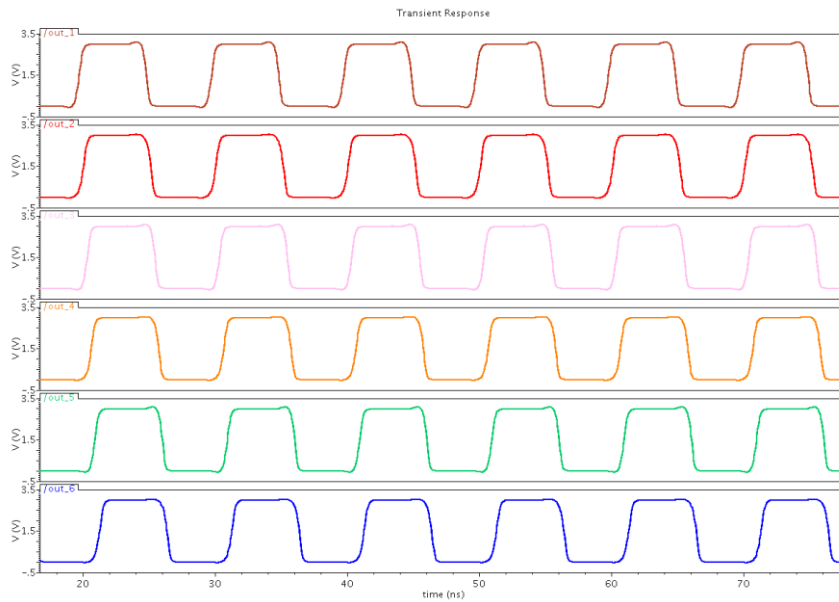


Ilustración 37 Señales de salida del interpolador post layout 1- 6

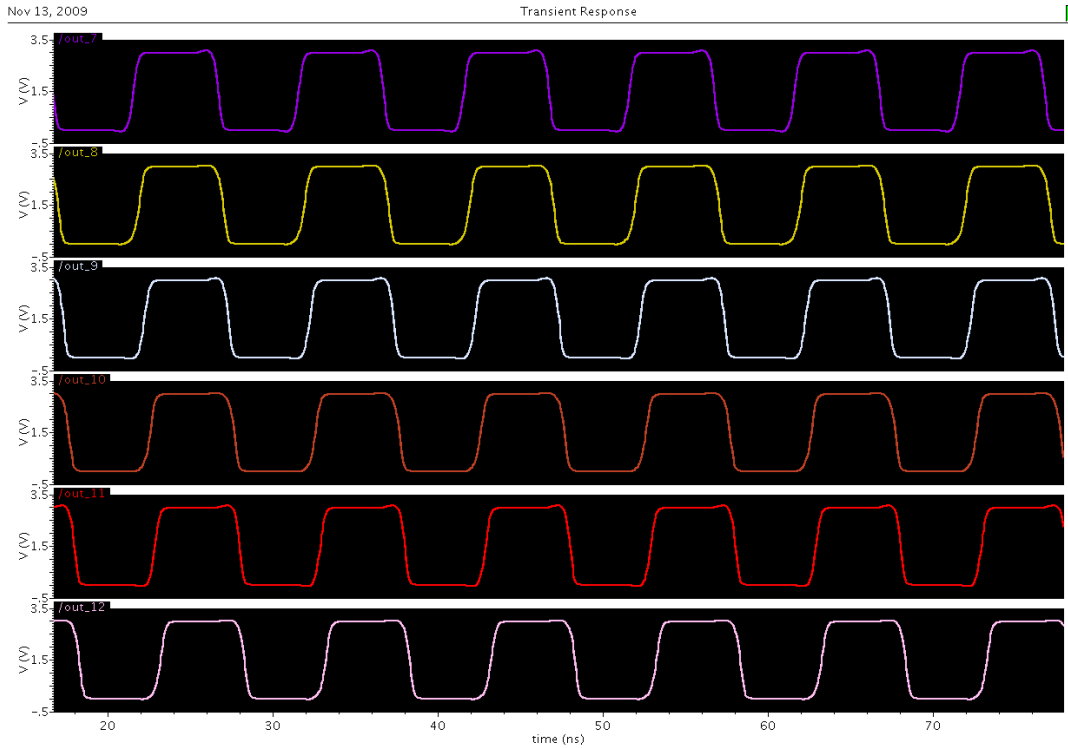


Ilustración 38 Señales de salida del interpolador post layout 7 -12

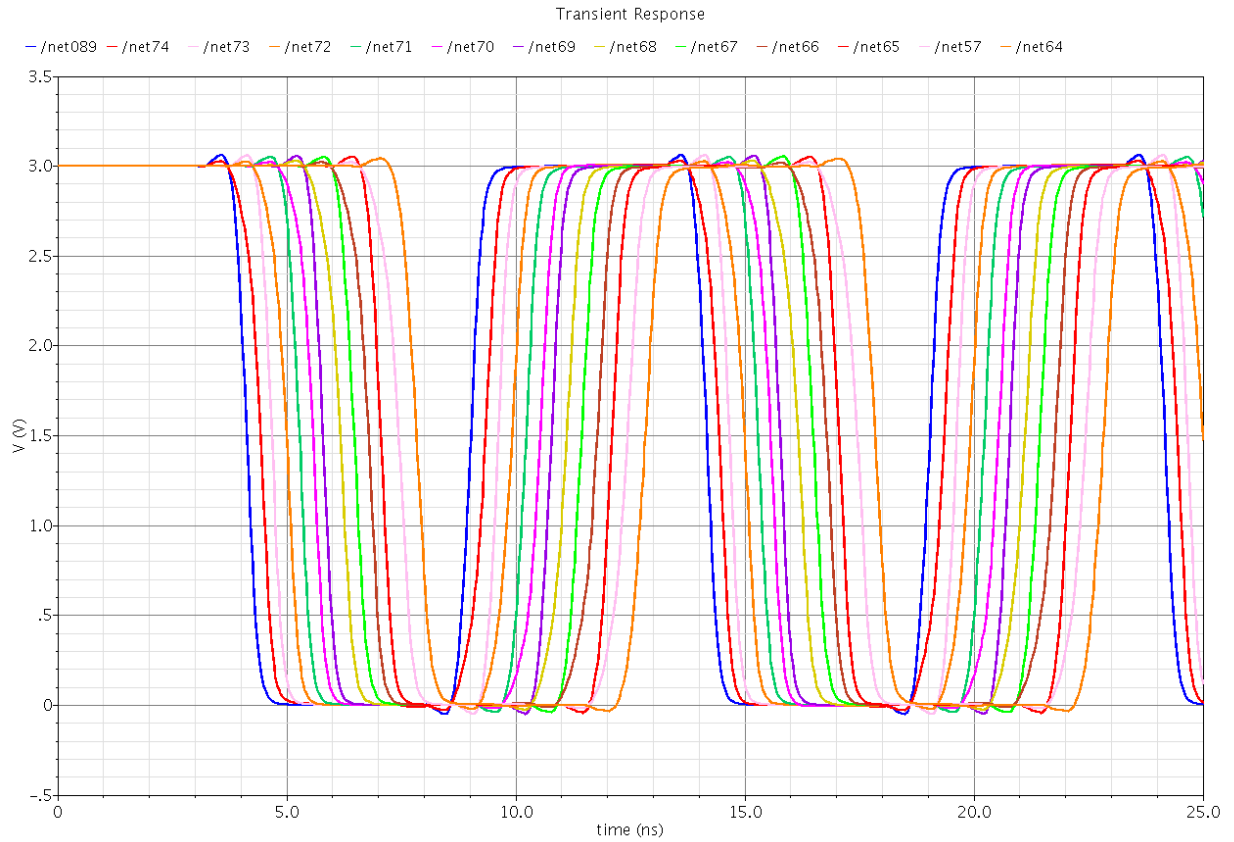


Ilustración 39 Señales interpoladas

Conclusiones:

Un interpolador es un circuito generador de señales de reloj, a partir de un número reducido de señales de reloj en su entrada.

En este proyecto se realizaron 2 propuestas layout con sus respectivas simulaciones a partir del floorplan realizado para el diseño. Básicamente el floorplan es la ubicación física de los diferentes elementos que constituyen el diseño en el Silicio; No obstante, la idea principal del floorplan es la optimización del espacio y la ubicación de los elementos de tal forma que minimice el retardo entre bloques para lograr un diseño que se desempeñe de mejor manera. El ruteo debe ser lo más recto y simple posible, evitando a toda costa caminos innecesarios y demasiados quiebres, además de buscar que las señales recorran la misma distancia, para un desempeño correcto.

En este proyecto se hizo uso de algunas técnicas para compensar las señales cuyas diferencias entre fases quedaran dentro de lo esperado como el uso de serpentinas.

Referencias

- 1) Análisis y Caracterización de Topologías Interpoladoras en Tecnologías Avanzadas VLSI
Edici ITESO.
- 2) Diseño de un Multiplexor Síncrono de Reloj 32-a-1 con Interpolador de 8-a-32 Fases para un Selector de Fase EDCI ITESO 2008.
- 3) Digital Integrated Circuits Analysis and Design John E. Ayers CRC Press 2004.

B. REPORTE DE PROYECTO 2: LAYOUT DEL 555

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

ESPECIALIDAD EN DISEÑO DE CIRCUITOS INTEGRADOS.

Taller de Diseño Físico de Circuitos Integrados.

NOMBRE: Gustavo Guzmán Rosales.

MD674630

03/12/2009

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

1. Introducción:

Ante la necesidad de obtener circuitos generadores de pulsos, multivibradores (temporizadores) se crearon circuitos basados en amplificadores operacionales en distintas aplicaciones.

Sin embargo en 1972 la compañía Signetics introdujo en el mercado un nuevo componente, que no solo cumplía con estas necesidades, sino que mejoraba los resultados obtenidos por los circuitos basados en amplificadores operacionales en muchos aspectos.

Hoy en día el 555 sigue siendo un componente básico en la construcción de circuitos multivibradores, generadores de pulsos, divisores de frecuencia...

La principal ventaja del 555 radica en que consigue temporizaciones más precisas. Además, al ser un circuito integrado reduce el número de conexiones a la vez que el precio, factor que todo ingeniero debe tener en cuenta a la hora del diseño.

1.1. Descripción:

El 555 es un circuito integrado que incorpora dentro de sí dos comparadores de voltaje, un flip flop, una etapa de salida de corriente, divisor de voltaje resistor y un transistor de descarga. Dependiendo de cómo se interconecten estas funciones utilizando componentes externos es posible conseguir que dicho circuito realice un gran número de funciones tales como la del multivibrador astable y la del circuito monoestable. Este integrado se puede usar en diversas aplicaciones, tales como:

- Control de sistemas secuenciales
- Generación de tiempos de retraso
- Divisor de frecuencias
- Modulación por anchura de pulsos
- Repetición de pulsos

- Generación de pulsos controlados por tensión, etc.

2. Objetivo:

El objetivo del proyecto es aplicar las diferentes técnicas de layout full-custom revisadas en clase, para la creación de un layout de un circuito de modo mixto.

3. Diseño:

En la imagen 1 se muestra un diagrama a bloques del timer 555 el cual será implementado.

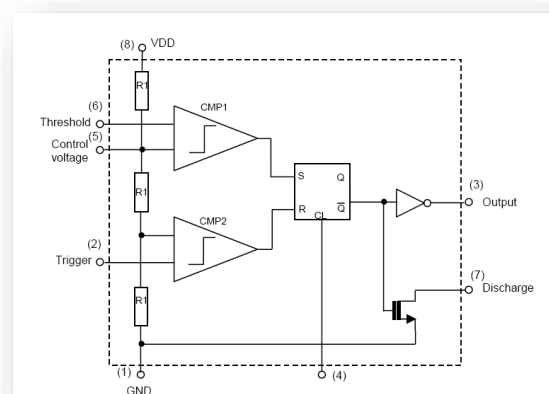


Imagen 1: Diagrama a bloques del timer 555

3.1. Descripción de los pines del timer 555.

GND: es el polo negativo de la alimentación, generalmente tierra.

Disparo/Trigger: es el pin, donde se establece el inicio del tiempo de retardo, si el 555 es configurado como monoestable. Este proceso de disparo ocurre cuando

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

este pin va por debajo del nivel de 1/3 del voltaje de alimentación. Este pulso debe ser de corta duración, pues si se mantiene bajo por mucho tiempo la salida se quedará en alto hasta que la entrada de disparo pase a alto otra vez.

Salida/Out: Aquí veremos el resultado de la operación del temporizador, ya sea que esté conectado como monoestable, astable u otro.

Reset: Si se pone a un nivel bajo o cero lógico, la salida permanece en 0v.

Control de voltaje: Cuando el temporizador se utiliza en el modo de controlador de voltaje, el voltaje en esta patilla puede variar casi desde Vcc (en la práctica como Vcc -1 voltio) hasta casi 0 V (aprox. 2 Voltios). Así es posible modificar los tiempos en que la salida está en alto o en bajo independiente del diseño (establecido por las resistencias y condensadores conectados externamente al 555). El voltaje aplicado a la patilla de control de voltaje puede variar entre un 45 y un 90 % de Vcc en la configuración monoestable.. Modificando el voltaje en esta patilla en la configuración astable causará la frecuencia original del astable sea modulada en frecuencia (FM). Si esta patilla no se utiliza, se recomienda ponerle un condensador de 0.01 μ F para evitar las interferencias.

Umbral/Threshold: Es una entrada a un comparador interno que tiene el 555 y se utiliza para poner la salida a nivel bajo.

Descarga: Utilizado para descargar con efectividad el condensador externo utilizado por el temporizador para su funcionamiento.

Vdd: Es el pin de alimentación positiva, El valor de la fuente de alimentación VCC varía según la tecnología de diseño que se

emplee, en el caso de la Tecnología de 0.5 μ m, VCC = 5V.

4.

5. Esquemáticos de bloques analógicos y digitales:

En este proyecto se realizaron por separado, algunos circuitos analógicos y digitales, los cuales componen el circuito del timer 555. En la imagen siguiente se muestra un diagrama de cómo se compone este circuito.

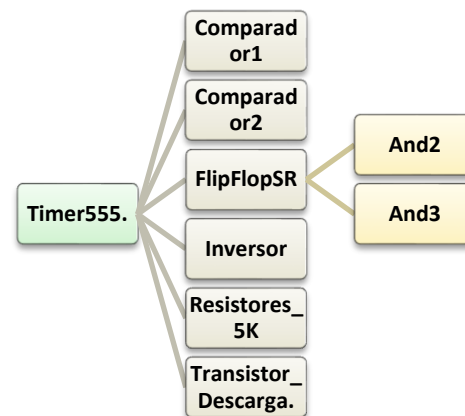


Imagen 2: Esquema de: Esquemáticos del Timer 555

5.1.

5.2. Comparador 1 y 2:

5.2.1 Esquemático:

Ofrecen a su salida dos estados perfectamente diferenciados (alto y bajo)

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

en función de las tensiones aplicadas a sus entradas (+ y -), de tal forma que:

Si $V(+)>V(-)$, la salida toma un nivel alto

Si $V(+)<V(-)$, la salida toma un nivel bajo

No se contempla el caso $V(+)=V(-)$, ya que una muy pequeña variación entre ambas haces que la salida adopte el nivel determinado por el sentido de dicha variación.

El diagrama esquemático realizado el que se muestra en la imagen que se presenta a continuación:

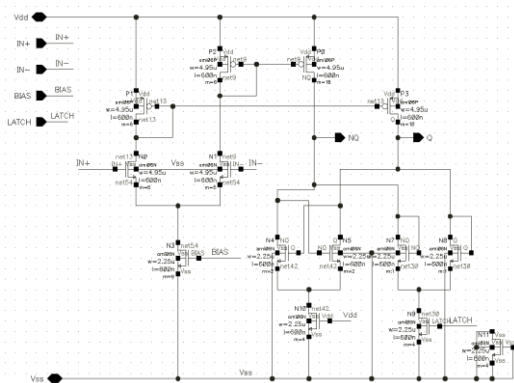


Imagen 3: Diagrama esquemático del comparador.

En la tabla siguiente se muestran las dimensiones de los transistores utilizados en el esquemático de la imagen 3, en la columna de multiplicador se muestra el número de transistores en los que se divido para poder hacer un mejor layout.

Tabla 1: Dimensiones de los transistores del Comparador.

	W	L	Multiplicador
M1	4.95um	0.6um	5
M2	4.95um	0.6um	5
M3	4.95um	0.6um	6
M4	4.95um	0.6um	6
M5	4.95um	0.6um	18
M6	4.95um	0.6um	18

M7	2.25um	0.6um	2
M8	2.25um	0.6um	2
M9	2.25um	0.6um	1
M10	2.25um	0.6um	1
M11	2.25um	0.6um	9
M12	2.25um	0.6um	4
M13	2.25um	0.6um	4
Dum	2.25um	0.6um	4

5.2.2 Layout:

El layout del comparador se realizo utilizando las técnicas de interdigitación y centroide común, las cuales fueron revisadas en clase, en la imagen 4 se puede observar el layout de este comparador, el cual paso las pruebas de LVS y DRC.

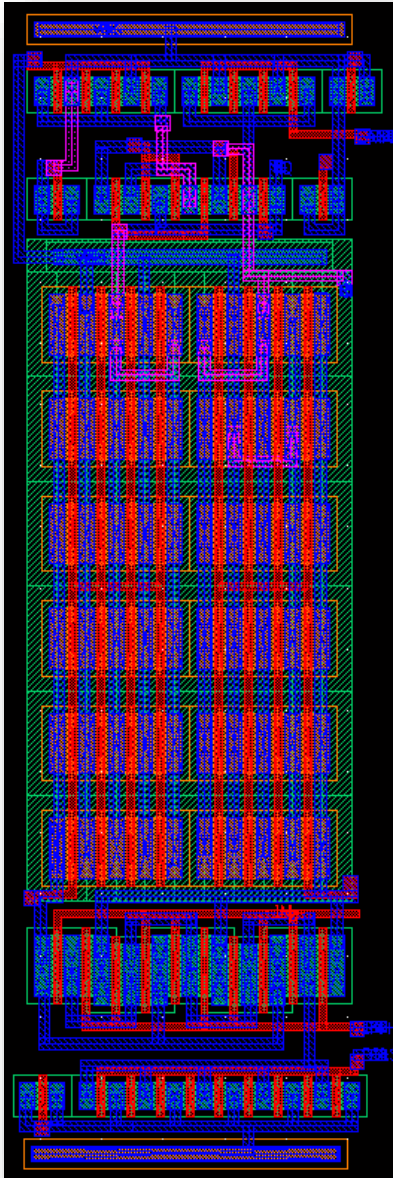


Imagen 4: Layout del comparador. 94.05um x29.95um

5.2.3

5.2.4 Extracción:

Para realizar el análisis post layout se realizó la extracción de las resistencias y capacitancias parasitas, el resultado de esta extracción se muestra en la imagen 5,

en dicha imagen se puede ver el valor de las capacitancias parasitas, las cuales puede o no afectar el funcionamiento del circuito, con la finalidad de reducir estas parasitas, y que no afecten tanto, se aplicaron las técnicas antes mencionadas en el layout.

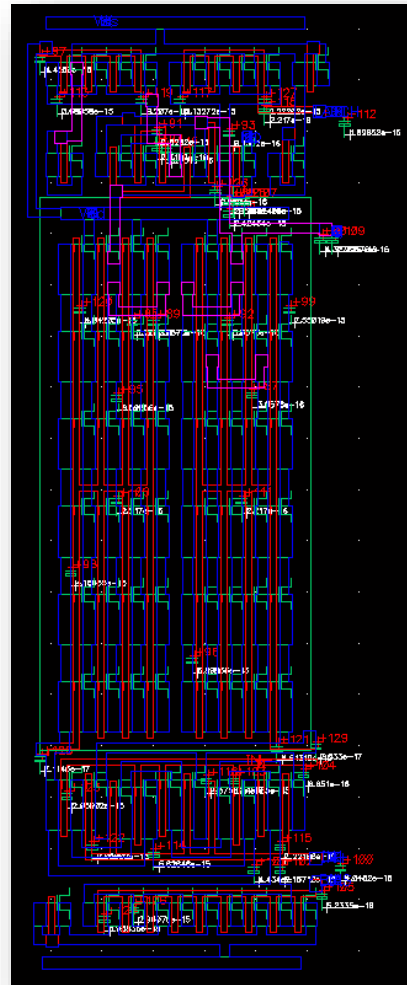


Imagen 5: Extracción de parasitas del comparador.

5.3. Flip Flop SR:

5.3.1 Esquemático:

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

Su funcionamiento responde al de cualquier biestable, ofreciendo dos estados permanentes. Presenta dos entradas de activación R y S, que condicionan su salida Q: Si R pasa de nivel bajo a alto, hace que el biestable pase a nivel bajo. Si S pasa de nivel bajo a alto, el biestable pasa a nivel alto. El paso de R o S de estado alto a bajo no influye al biestable.

El diagrama esquemático realizado el que se muestra en la imagen que se presenta a continuación:

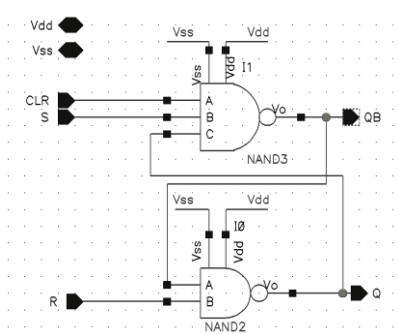


Imagen 6: Diagrama esquemático del Flip Flop SR.

Se observa de la figura 6 que este flip flop está compuesto por 2 compuertas NAND una de 2 y otra de 3 entradas, los diagramas esquemáticos de estas compuertas se muestran en las imágenes 7 y 8.

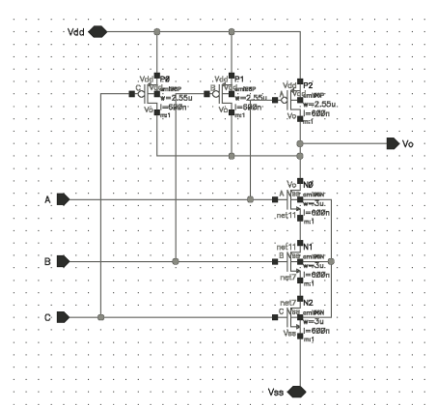


Imagen 7: Diagrama esquemático: NAND 3.

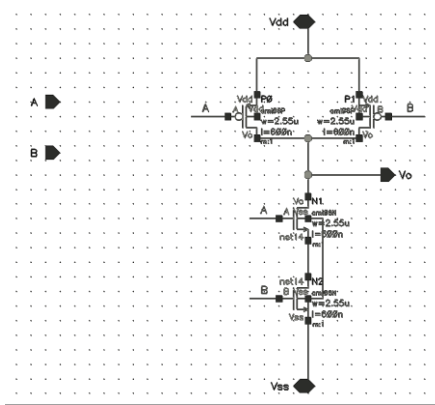


Imagen 8: Diagrama Esquemático: NAND 2

En las tablas siguientes se muestran las dimensiones de los transistores utilizados en el esquemático de las compuertas NAND de 2 y de 3 entradas.

Tabla 2: Dimensiones de los transistores de la compuerta NAND de 2 entradas.

	W	L
MN	2.55um	0.6um
MP	2.55um	0.6um

Tabla 3: Dimensiones de los transistores de la compuerta NAND de 3 entradas

	W	L
MN	3.00um	0.6um
MP	2.55um	0.6um

5.3.2 Layout:

El layout del flip flop SR se realizo de manera jerárquica a partir de los layout de las compuertas, el layout de cada compuerta son mostrados en las imágenes siguientes, así también como el layout final del flip flop SR. Para este layout, se cuidó que fuese un layout compacto, además de que las compuertas de los transistores tuvieran la misma orientación.

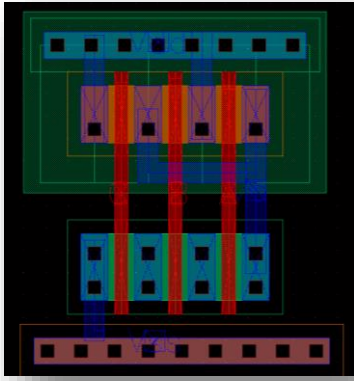


Imagen 9: Layout: NAND3 13.5um x 16.5um

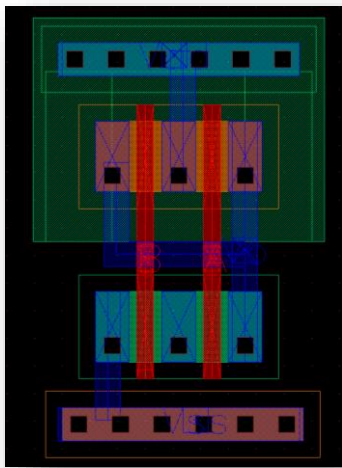


Imagen 10: Layout: NAND2 10.5um x 16.5um

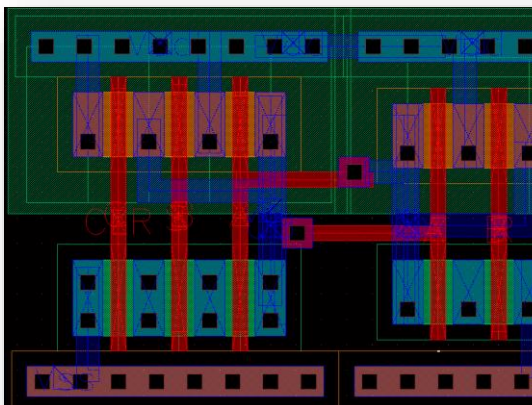


Imagen 11: Layout: Flip Flop SR. 23.55um x 16.5um

5.3.3

5.3.4 Extracción:

Para realizar el análisis post layout se realizó la extracción de las resistencias y capacitancias parásitas, el resultado de esta extracción se muestra en la imagen 12, en dicha imagen se puede ver el valor de las capacitancias parásitas.

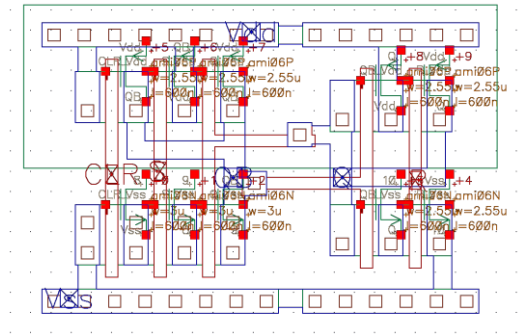


Imagen 12: extracción de parásitos, del flip flop sr

5.4. Inversor:

5.4.1 Esquemático:

Debido a que la señal de salida debe ser la negada de la señal de entrada, se pone un inversor antes de la salida. el diagrama esquemático de este inversor es el que se muestra en la imagen 13.

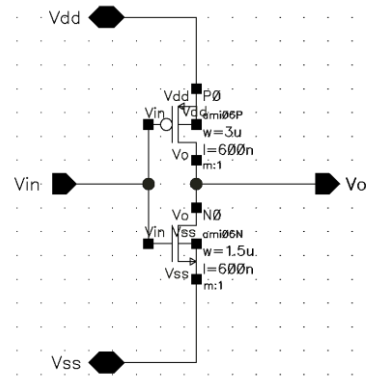


Imagen 13: Diagrama esquemático del inversor.

Tabla 4: Dimensiones de los transistores de la compuerta inversora.

	W	L
MN	1.5um	0.6um
MP	3.0um	0.6um

5.4.2 Layout:

El layout del inversor es muy sencillo, este layout se muestra en la figura siguiente, para su realización, se cuidó, que fuera un layout compacto, además de que las compuertas de los transistores tuvieran la misma orientación.

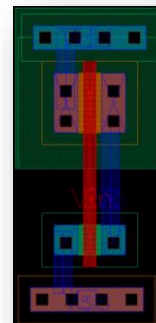


Imagen 14: Layout del inversor.7.5um x 16.5um

5.4.3 Extracción:

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

Para realizar el análisis post layout se realizó la extracción de las resistencias y capacitancias parasitas, el resultado de esta extracción se muestra en la imagen 15, en dicha imagen se puede ver el valor de las capacitancias parasitas.

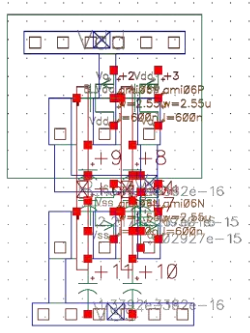


Imagen 15: Extracción de parasitas del Inversor.

5.5. Resistores 5K:

5.5.1 Esquemático:

Los resistores sirven para formar un divisor de tensión, este divisor formado por tres resistencias iguales de 5k. Su comportamiento caracteriza el estado de los comparadores. Sitúa $1/3V_{cc}$ en la entrada no inversora del comparador I, y $2/3 V_{cc}$ en la inversora del comparador II.

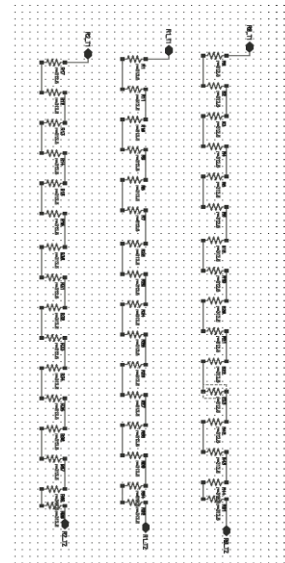


Imagen 16: Diagrama esquemático de los resistores de 5K.

Cada una de las 3 resistencias de 5K se dividió en 16 resistencias de 312.5. Con el fin de hacer un arreglo interdigitado, para lograr un mejor apareamiento entre los tres resistores en el layout.

5.5.2 Layout:

En el layout de los resistores, se utilizó un arreglo interdigitado, además de que se agregó un anillo de guarda, con el fin de proteger del ruido a los resistores. Dicho layout se muestra en la imagen siguiente.

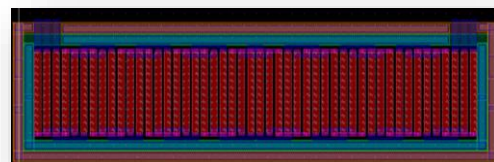


Imagen 17: Layout de las 3 resistencias de 5K. 203.4umx 57.75um

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

5.5.3

5.5.4 Extracción:

Para realizar el análisis post layout se realizo la extracción de las resistencias y capacitancias parasitas, el resultado de esta extracción se muestra en la imagen 18, en dicha imagen se puede ver el valor de las capacitancias y resistencias parasitas, en la imagen 19 se observa el valor del cada uno de los resistores unitarios.

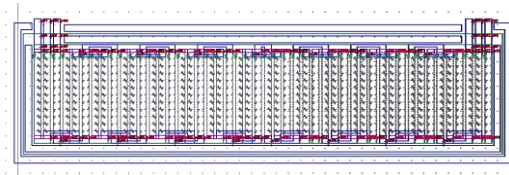


Imagen 18: Extracción de las resistencias de 5K

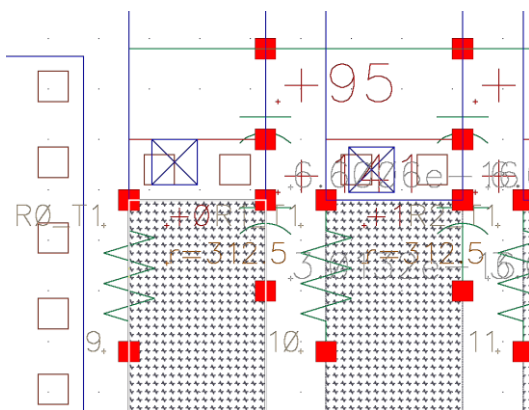


Imagen 19: Valor del resistor unitario

5.6. Timer 555:

5.6.1 Esquemático:

El diagrama esquemático del 555 corresponde al mostrado en la imagen 1, para la simulación de este proyecto se realizo el mostrado en la imagen 20.

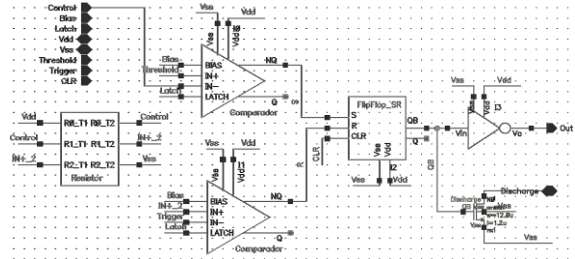


Imagen 20: Esquemático del circuito 555.

Algunas características de este circuito son:

Los tamaños del transistor de descarga:

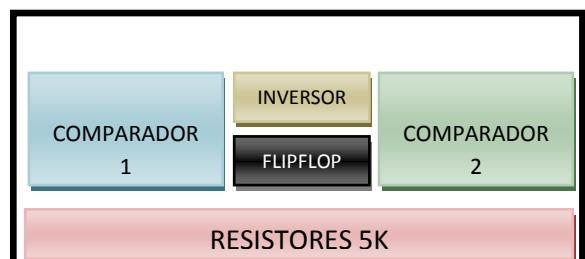
- ◆ $W=12.0\mu m$.
- ◆ $L=2.00\mu m$.

Los voltajes de polarización para los comparadores:

- ◆ $V_{bias} = 1.8v$
- ◆ $V_{latch} = 3.6v$
- ◆ $V_{dd} = 5v$
- ◆ $V_{ss} = 0v$

5.6.2 Layout:

El layout completo del circuito se realizó pensando en hacer un layout simétrico y con una buena distribución del ruteo para que las señales no recorran mucha distancia. El layout de este proyecto se baso en el floorplan mostrado en la imagen siguiente.



Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

El layout completo, en el cual se integran de manera jerárquica los layout de las etapas anteriores, se muestra en la imagen siguiente, se puede observar que es un layout simétrico con dimensiones de 203.55um por 97.650um.

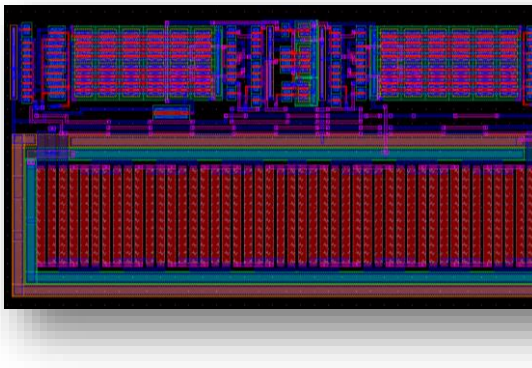


Imagen 21: Layout del Timer 555.

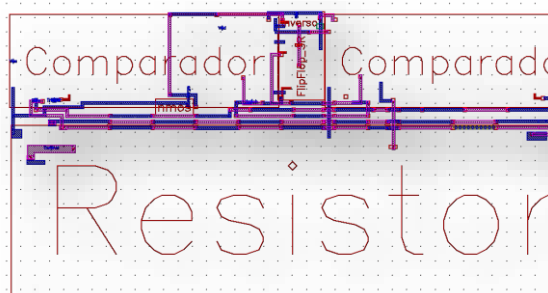


Imagen 22: Bloques que componen el layout del timer 555

5.6.3

5.6.4

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

5.6.5 Extracción:

Para realizar el análisis post layout se realizo la extracción de las resistencias y capacitancias parasitas, el resultado de esta extracción se muestra en la imagen 23, en dicha imagen se puede ver el valor de las capacitancias y resistencias parasitas. Estas parasitas son las que hacen que las características del sistema cambien durante el análisis post layout.

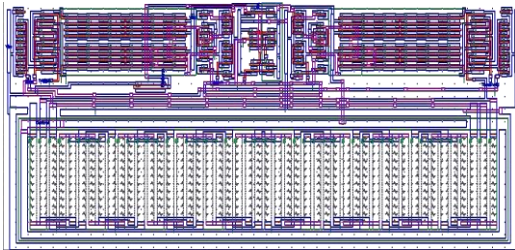


Imagen 23: Extracción de parasitas del timer 555.

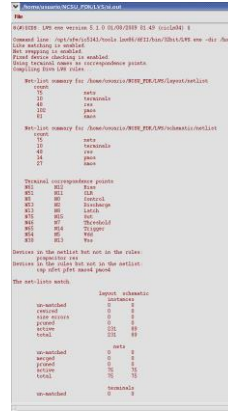


Imagen 25: Informe del LVS

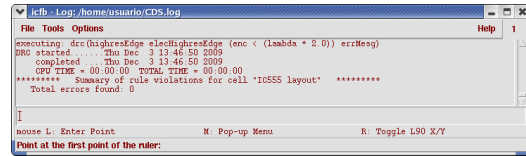


Imagen 26: DRC con 0 errores.

5.6.6 Pruebas LVS y DRC:

A este layout se le aplicaron las pruebas de DRC y LVS, las cuales pasaron satisfactoriamente, para comprobar esto se muestran las imágenes de resultados.

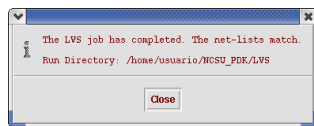


Imagen 24: LVS correcto.

5.7.

5.8. SIMULACION:

Para la simulación del timer 555 en modo astable se realizo el siguiente diagrama esquemático de pruebas.

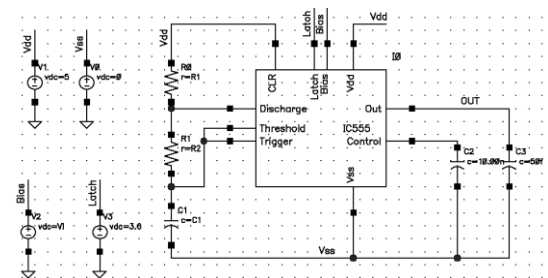


Imagen 27: Diagrama esquemático del 555 como multivibrador astable.

Los valores de los componentes utilizados en esta simulación para que el 555 funcione como multivibrador astable a una frecuencia de 100KHz, son los siguientes:

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

$R1=2.15K\Omega$, $R2=380K\Omega$,
 $C1=10pF$, $C2=0.01\mu F$, $C3=50fF$.

5.8.1

5.8.2 Simulación Pre Layout:

Los resultados de la simulación pre layout son mostrados en la grafica siguiente, en esta grafica se muestra la salida del 555, donde la forma de onda tiene un periodo de: 10.08us, por lo que la frecuencia es de: 99.20KHz

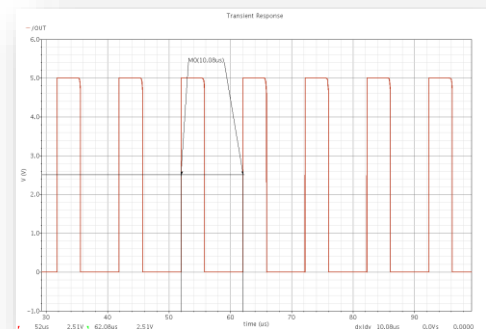


Imagen 28: salida del 555 en pre layout.

5.8.3

5.8.4 Simulación Post Layout:

Los resultados de la simulación post layout son mostrados en la grafica siguiente, en esta grafica se muestra la salida del 555, donde la forma de onda tiene un periodo de: 10.3us, por lo que la frecuencia es de: 97.08KHz

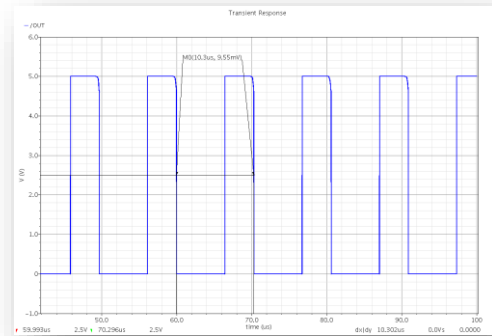


Imagen 29: Salida del 555 en Post Layout.

5.9. Conclusiones:

Se puede observar de las simulaciones pre layout y post layout, que en post layout la frecuencia disminuye teniendo un cambio o una diferencia de 2.12KHz, creo que este error no es demasiado, pudo ser mucho mayor si no se hubieran empleado técnicas e layout de interdigitacion y centroide común así como las de colocar anillos de guarda, dummies entre otras. Creo que con más tiempo hubiera sido bueno realizar otro layout sin aplicar las técnicas que se aprendieron durante el curso de diseño físico de circuitos integrados.

La diferencia que hay en el pre layout y el post layout son debido a que en el pre layout no se consideran las parasitas, ni los ruteos de las señales, los cuales en algunos casos fueron largos.

Durante el desarrollo de este proyecto me enfrente con algunos problemas asociados con el uso de la herramienta utilizada en la simulación, en especifico, con que la simulación en transitorio que realice para ver monitorear la salida, cuando el tiempo era de 100us la salida oscilaba

Reporte: Proyecto Final: DISEÑO DE LAYOUT DEL TIMER 555.

correctamente, y cuando el tiempo era de 200us la salida no oscilaba, lo único que creo sobre esto es que el muestro de la señal que hace el simulador afecta.

Por último, con este proyecto me doy cuenta de cómo un sistema, analógico puede convivir con un sistema digital, así como las técnicas de layout que se utilizan para cada tipo de sistema son diferentes, por lo cual es indispensable conocer sobre qué tipo de sistema se está realizando el layout para poder aplicar las técnicas adecuadas para el mejor funcionamiento del sistema completo.

5.10. Referencias:

CMOS digital Integrate Circuits. Analysis and Design. Sung-Mo Kang Yusuf Leblebici

<http://www.monografias.com/trabajos11/contact/contact.shtml#circuito>

<http://www.uv.es/marinjl/electro/555.htm>

C. REPORTE DE PROYECTO 3: PLL

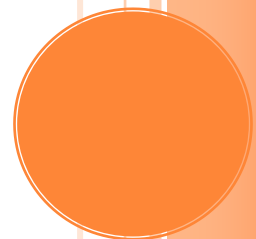
DISEÑO DE CP-PLL

Reporte de proyecto final de la materia Analog Design II.

Este proyecto tiene como objetivo, diseñar un PLL (Phase Locking Loop), obteniendo el modelo transitorio y en AC. Un PLL es un sistema realimentado, que en función de una señal de referencia (magnitud y fase) genera un voltaje de control para un VCO, el cual oscila para hasta marrarse a la frecuencia de la señal de entrada.

Nombre: Gustavo Guzmán Rosales MD674630

22/04/2010



CONTENIDO

OBJETIVO:	3
Diseño del modelo AC del CP-PLL.....	4
Resultados de Simulación en MatLab.....	7
Resultados de Simulación en Cadence.....	10
Diseño del modelo Transitorio del CP-PLL.....	14
Modulo VerilogA del PFD	14
Modulo VerilogA del Flip Flop	15
Modulo VerilogA del AND.....	16
Modulo VerilogA del Ip	17
Modulo VerilogA del LPF.....	19
Modulo VerilogA del VCO.....	20
Modulo VerilogA del N.....	21
Integración del Soc.....	22
Modulo VerilogA del buffer.....	22
Modulo VerilogA del multiplexor 2-1.....	23
Esquemático del SoC:.....	24
Resultados de Simulación:.....	25
Medicion del Output Skew.....	29
Medicion del tiempo de Rise & Fall	31
Resultado de Simulación Transitoria:.....	32
Con Divisor de 2:.....	32
Con Divisor de 4:.....	33
Con Divisor de 6:.....	33
Con Divisor de 8:.....	34
Con Divisor de 24:.....	35
Medición del Lock Time:	35
Con Divisor de 2:.....	35
Con Divisor de 4:.....	36
Con Divisor de 6:.....	37
Con Divisor de 8:.....	37
Con Divisor de 12:.....	38
Con Divisor de 16:.....	39
Con Divisor de 24:.....	40
Medición del Cycle jitter:	42

Con Divisor de 2:.....	42
Con Divisor de 4:.....	42
Con Divisor de 6:.....	43
Con Divisor de 8:.....	43
Con Divisor de 12:.....	44
Con Divisor de 16:.....	45
Con Divisor de 24:.....	46
Medición del PHASE jitter:.....	46
Con Divisor de 2:.....	46
Con Divisor de 4:.....	47
Con Divisor de 6:.....	47
Con Divisor de 8:.....	48
Con Divisor de 12:.....	49
Con Divisor de 16:.....	50
Con Divisor de 24:.....	50
Medición del STATIC PHASE OFFSET:.....	51
Con Divisor de 2:.....	51
Con Divisor de 4:.....	52
Con Divisor de 6:.....	53
Con Divisor de 8:.....	53
Con Divisor de 12:.....	54
Con Divisor de 16:.....	55
Con Divisor de 24:.....	56
Referencias:.....	57

OBJETIVO:

Diseñar un PLL basado en Charge-Pump (CP-PLL) que va inmerso en un SoC

- 1:6 PLL LVCMOS clock generator.
- 3.0 power supply
- PLL generates clock signals up to 1.6 GHz
- Max. output skew of 50ps
- On-chip crystal oscillator reference clock
- LVCMOS PLL reference clock input
- Internal and external PLL Feedback
- Ambient temperature range: 0C to +70C
- Spread Spectrum Compatible (30KHz – 33KHz, Sinusoidal, Triangular and Hershey's Kiss)
- < 50 ps max cycle–cycle jitter
- < 20 ps max I/O Phase jitter
- < 20 ps max static phase offset (SPO)
- < 50 ps max output–output skew
- PLL Lock time < 4us
- Power Down Current < 10uA
- Current Consumption < 20mA
- Output Duty Cycle = 50% +/- 1%
- Rise & Fall time: 100ps
- Output Load = 10pF per output
- $V_x = V_{DD}/2 \pm 1\%$ (Cross Voltage)
- Generar Modelo de AC (PLL) y modelo Transitorio (PLL + Todo el sistema, incluir jitter & SSC)

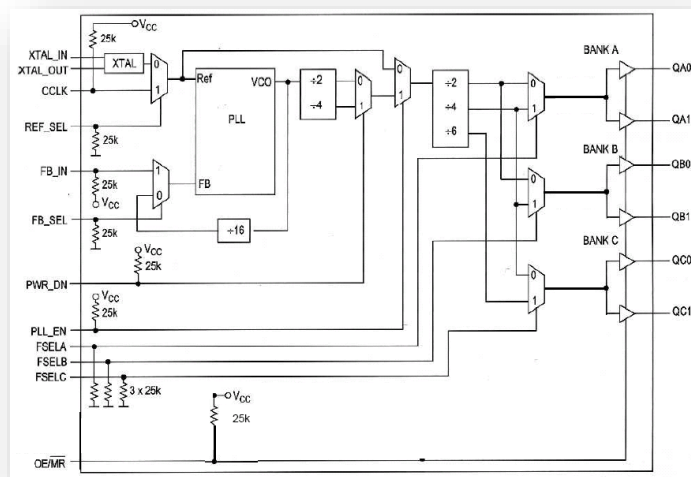


Figura 1: Diagrama a bloques del SoC.

1. DISEÑO DEL MODELO AC DEL CP-PLL.

Para encontrar el modelo en AC del PLL, primero es necesario definir las relaciones de las frecuencias, en la grafica siguiente se muestran las relaciones de las frecuencias, a partir de los divisores se encuentra el ancho de banda y la ubicación del cero de la función de transferencia.

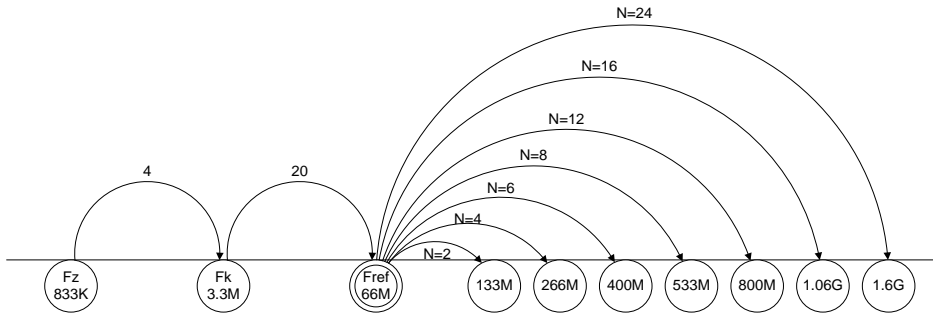


Figura 2: Relationship of critical frequencies.

De la figura anterior se determina el rango de sintonía del VCO, esto a partir de las frecuencias máximas y mínimas del PLL, por lo tanto el rango para este VCO es de 133M a 1.6G.

Se determina la relación de división de lazo debido a que se tienen múltiples divisores, se obtiene la media aritmética de la forma siguiente.

$$N_{mean} = (N_{min} * N_{max})^{1/2} = (2 * 24)^{1/2} = 6.92$$

Se determina el valor de Damping $\zeta_{mean} = 0.7071$ debido a que se quiere que se tenga un sistema rápido, pero sin tanto sobre impulso.

Se determina la frecuencia natural conociendo que se va a diseñar un CP-PLL con un valor de $\zeta_{mean} = 0.7071$

$$\omega_n = \frac{\omega_{3dB}}{2.06} = \frac{3.3MHz * 2\pi}{2.06} = 1.02E+07$$

Se determina la ganancia del VCO, de la forma siguiente:

$$K_O = \frac{2\pi(F_{max}-F_{min})}{V_{sat}} = \frac{2\pi(1.6G-133M)}{200mV} = 7.33E+09$$

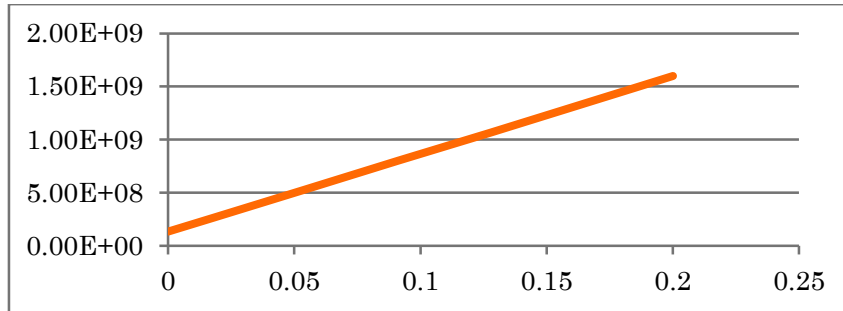


Figura 3: Grafica de respuesta teórica del VCO

El diagrama a bloques del CP-PLL se muestra en la imagen siguiente:

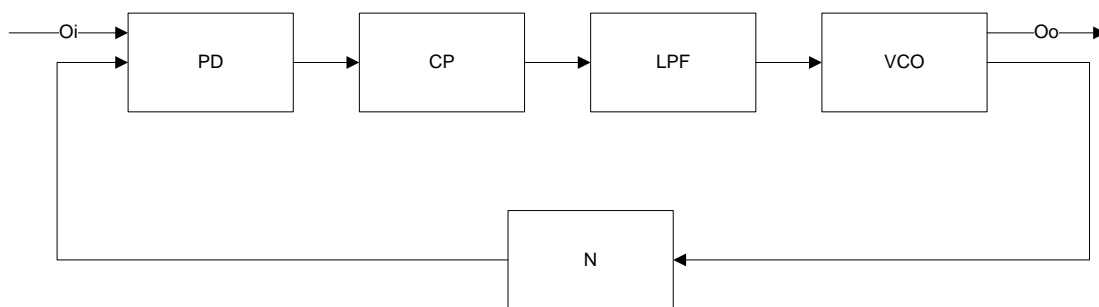


Figura 4: Diagrama a Bloques del CP-PLL

Donde las funciones de transferencia en lazo abierto de cada bloque son las siguientes:

PD:

$$= \frac{1}{2\pi}$$

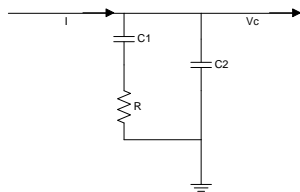
CP:

$$= Ip$$

VCO:

$$= \frac{Ko}{s}$$

LPF:



$$= \frac{RC1s + 1}{s^2RC1C2 + s(C1 + C2)}$$

Por lo tanto la función de transferencia del CP-PLL en lazo abierto es:

$$= \left(\frac{1}{2\pi}\right) * (Ip) * \left(\frac{Ko}{s}\right) * \left(\frac{RC1s+1}{s^2 RC1C2+s(C1+C2)}\right)$$

$$= \frac{\frac{1}{2\pi} * Ip * Ko * R * C1 * s + \frac{1}{2\pi} * Ip * Ko * R * C1}{s^3 RC1C2 + s^2 (C1 + C2)}$$

Y en lazo cerrado:

$$= \left(\frac{\frac{1}{2\pi} * Ip * Ko * R * C1 * s + \frac{1}{2\pi} * Ip * Ko * R * C1}{s^3 RC1C2 + s^2 (C1 + C2)}\right) / \left(1 + \frac{\frac{1}{2\pi} * Ip * Ko * R * C1 * s + \frac{1}{2\pi} * Ip * Ko * R * C1}{s^3 RC1C2 + s^2 (C1 + C2)}\right)$$

$$= \frac{\frac{1}{2\pi} * Ip * Ko * R * C1 * s + \frac{1}{2\pi} * Ip * Ko * R * C1}{s^3 RC1C2 + s^2 (C1 + C2) + \frac{1}{2\pi} * Ip * Ko * R * C1 * s + \frac{1}{2\pi} * Ip * Ko * R * C1}$$

$$\omega_n^2 = \frac{I_p K_{VCO}}{2\pi C_1}$$

$$2\xi\omega_n = \frac{I_p R K_{VCO}}{2\pi}$$

$$G_o(s) = \frac{2\xi\omega_n s + \omega_n^2}{m \frac{2\xi}{\omega_n} s^3 + (m+1)s^2}$$

$$G_c(s) = \frac{2\xi\omega_n s + \omega_n^2}{m \frac{2\xi}{\omega_n} s^3 + (m+1)s^2 + 2\xi\omega_n s + \omega_n^2}$$

De las ecuaciones anteriores se calculan los valores de R, C1, C2, con una corriente Ip=1uA, si se sabe que C2 debe ser menor que C1 para que el tercer polo sea muy pequeño, además C2 debe tener un valor tal que disminuya el rizo. Se propone una relación de C1/C2=0.05.

Por lo tanto:

	Hz	rad/s
W3dB	3.33E+06	2.09E+07
Wz	8.33E+05	5.24E+06
Wn	1.62E+06	1.02E+07
Kv	7.33E+09	-----
R	12319.14788	-----
C1	1.12912E-11	-----
C2	5.64558E-13	-----
N	6.92	-----

Por lo que la función de transferencia en lazo cerrado con retroalimentación unitaria es de:

$$\frac{Oo}{Oi} = \frac{1.437e^{007} s + 1.033e^{014}}{3.617e^{-009}s^3 + 1.05s^2 + 1.437e^{007}s + 1.033e^{014}}$$

Ecuación 1.

Por lo que la función de transferencia en lazo cerrado con retroalimentación N es de:

$$\frac{Oo}{Oi} = \frac{1.437e^{007} s + 1.033e^{014}}{3.617e^{-009}s^3 + 1.05s^2 + 2.074e^{006}s + 1.491e^{013}}$$

Ecuación 2.

1.1. Resultados de Simulación en MatLab.

Las ecuaciones 1 y 2 obtenidas, se obtuvieron para hacer un análisis de cómo la realimentación afectaba a la estabilidad y rapidez del sistema, a continuación se presentan unas graficas comparativas relazadas en MatLab.

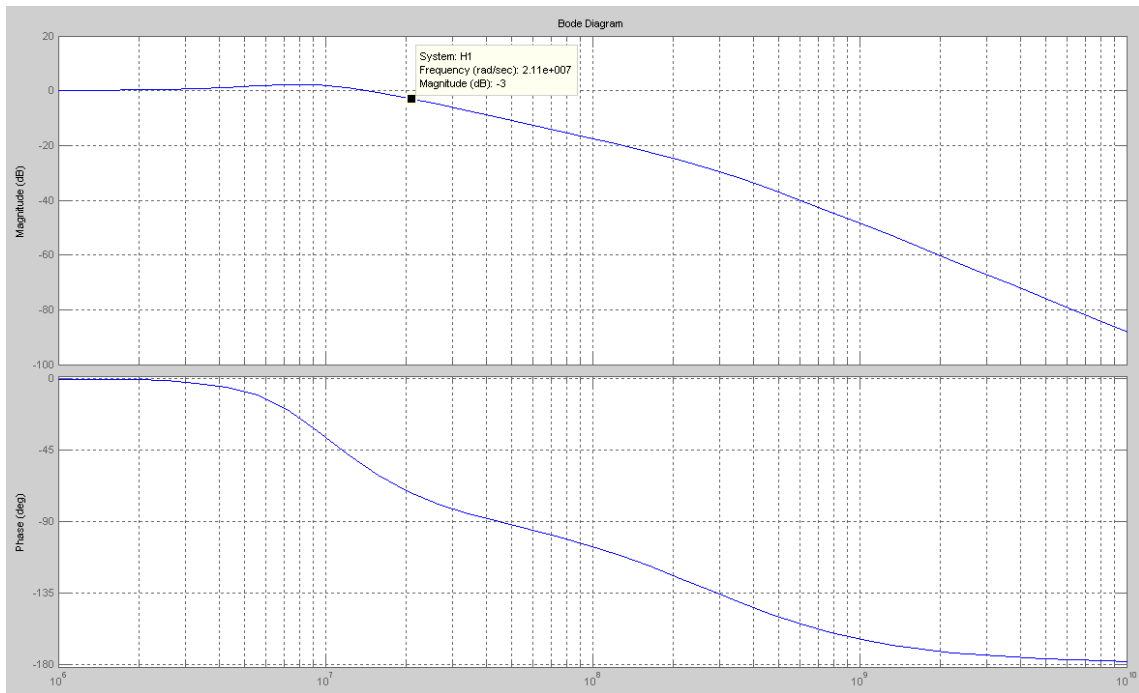


Figura 5: Respuesta en frecuencia del sistema con realimentación N=1, ancho de banda =2.11e7

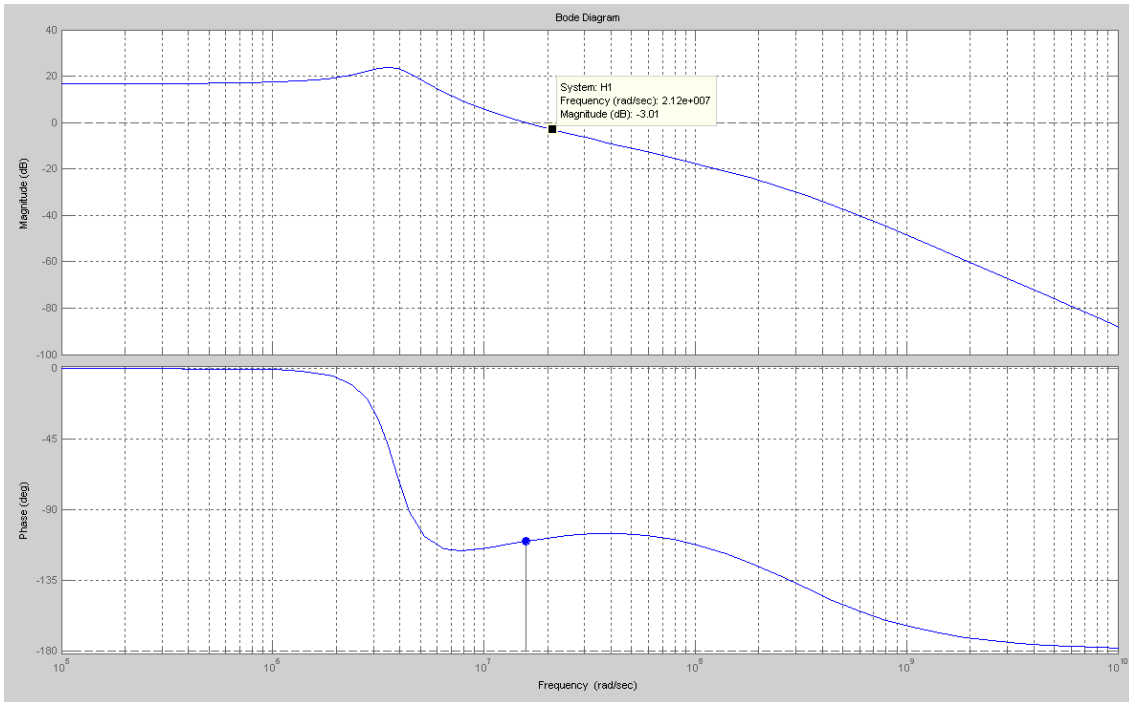


Figura 6: Respuesta en frecuencia del sistema con realimentación $N=6.92$, ancho de banda $=2.12e7$

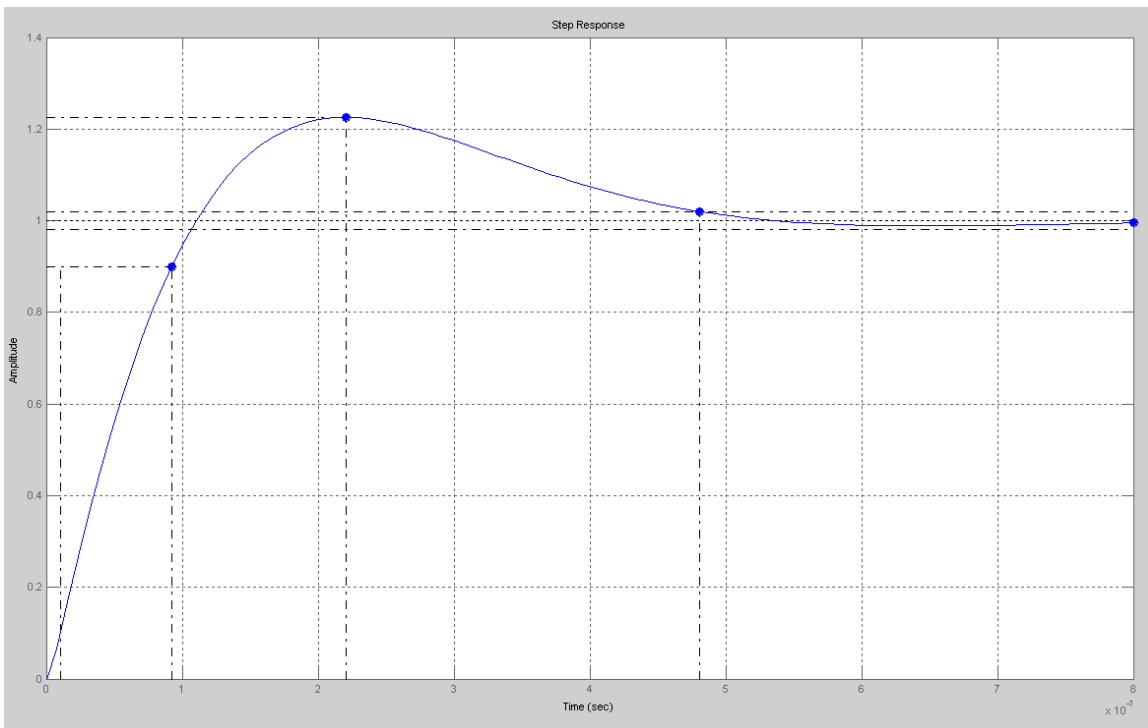


Figura 7: Respuesta a la entrada escalón del sistema con $N=1$

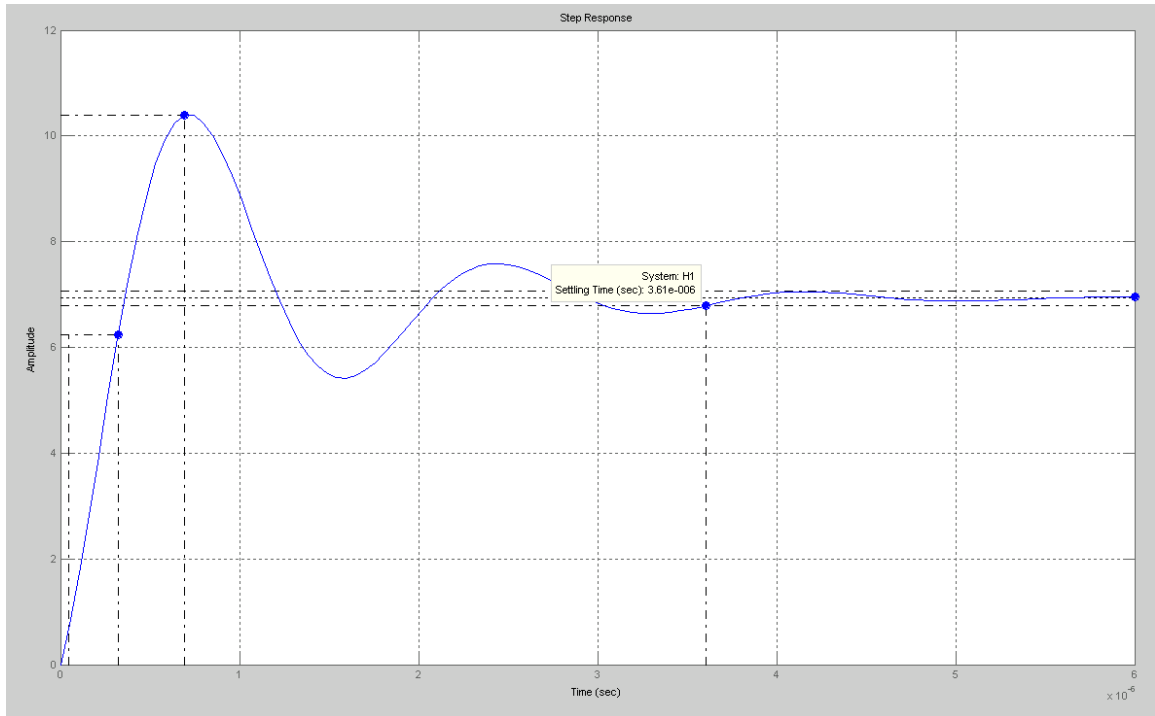


Figura 8: Respuesta a la entrada escalón del sistema con $N= 6.92$

Para verificar la estabilidad del sistema se obtuvo las graficas del lugar geométrico de las raíces, donde se observa que los polos y ceros de la función de transferencia se encuentran en el semiplano izquierdo, por lo tanto el sistema es estable, además se observa que el tercer polo se encuentra muy alejado, por lo que se puede considerar como un sistema de segundo orden.

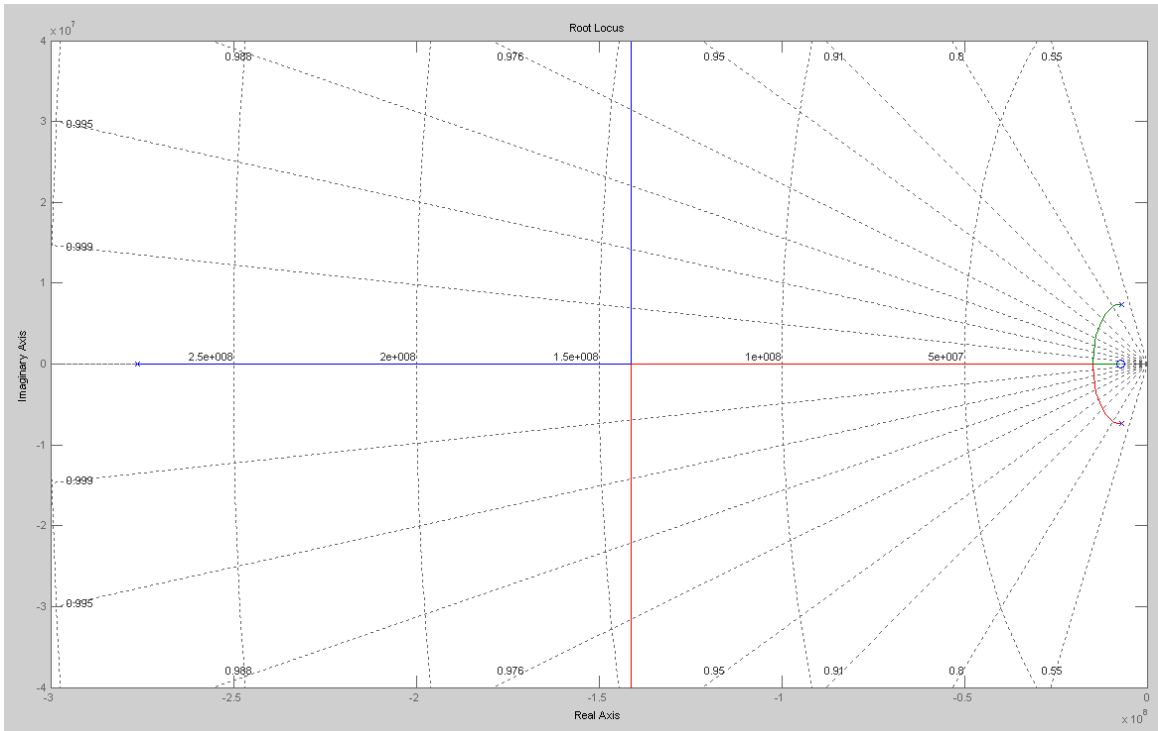


Figura 9: Grafica del LGR, para el sistema con $N=1$

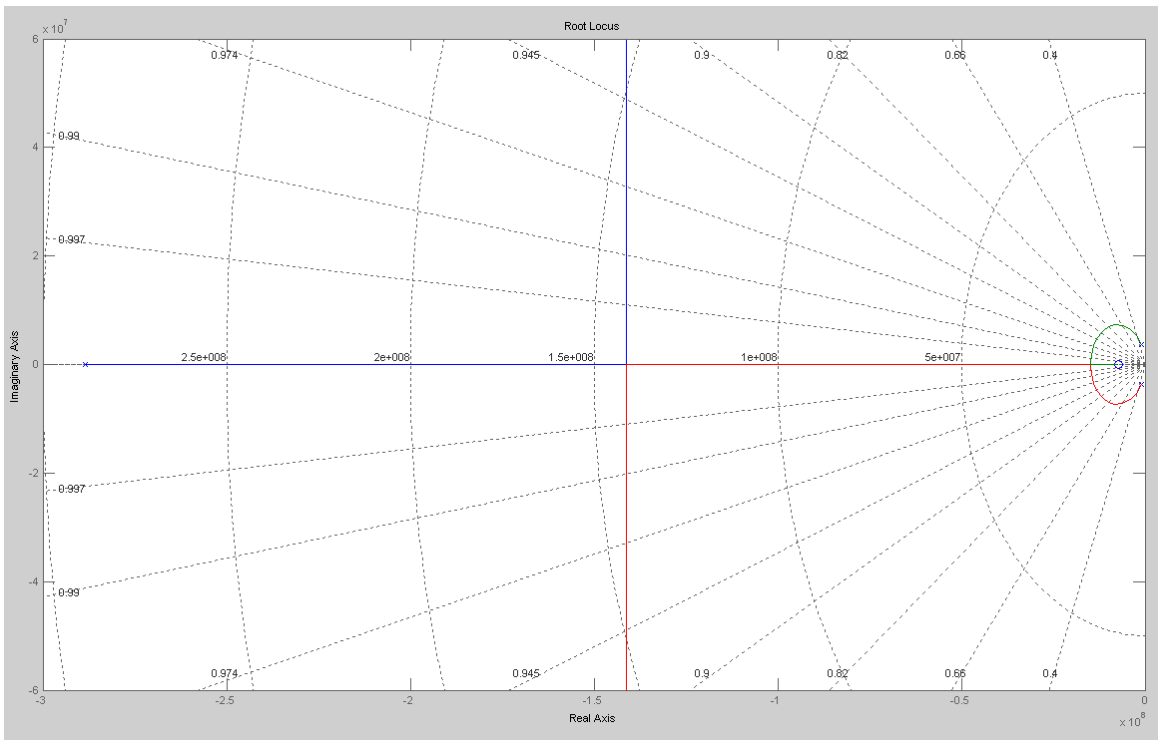


Figura 10: Grafica del LGR, para el sistema con $N=6.92$

1.2. Resultados de Simulación en Cadence.

El modelo en VerilogA del CP-PLL en AC es el que se describe a continuación, se puede observar que en el código se modela la función de transferencia de la ecuación 1, utilizando la función `laplace_nd`, utilizando el parámetro `N` se puede simular también la función de transferencia de la ecuación 2:

```
// VerilogA for PLL_Proyecto, PLL_AC, veriloga
// Nombre: PLL_AC
// Descripción: Función de transferencia en lazo cerrado con N=6.92
// POR: Gustavo Guzmán Rosales.

`include "constants.vams"
`include "disciplines.vams"

module PLL_AC(In , Out);
input In;
output Out;
electrical In,Out;

parameter real N=6.92;
analog begin
V(Out)<+
laplace_nd(V(In),{1.033e14,1.437e7},{1.033e14/N,1.437e7/N,1.05,3.617e-
9});
end

endmodule
```

Los resultados de la simulación de este modulo se muestran en las imágenes siguientes, ya que el modulo se puede parametrizar, se aprovecha y se realizan dos simulaciones, tal y como se realizo en MatLab, una con $N=1$ y otra con $N=6.92$.

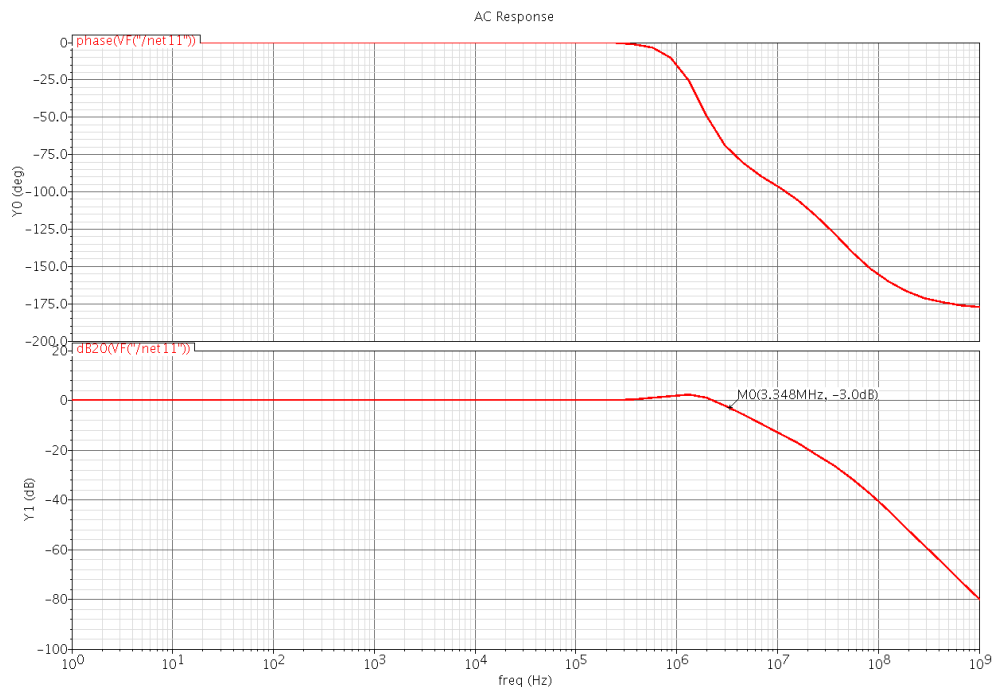


Figura 11: respuesta en frecuencia del sistema con $N=1$ $K=3.348\text{MHz}$

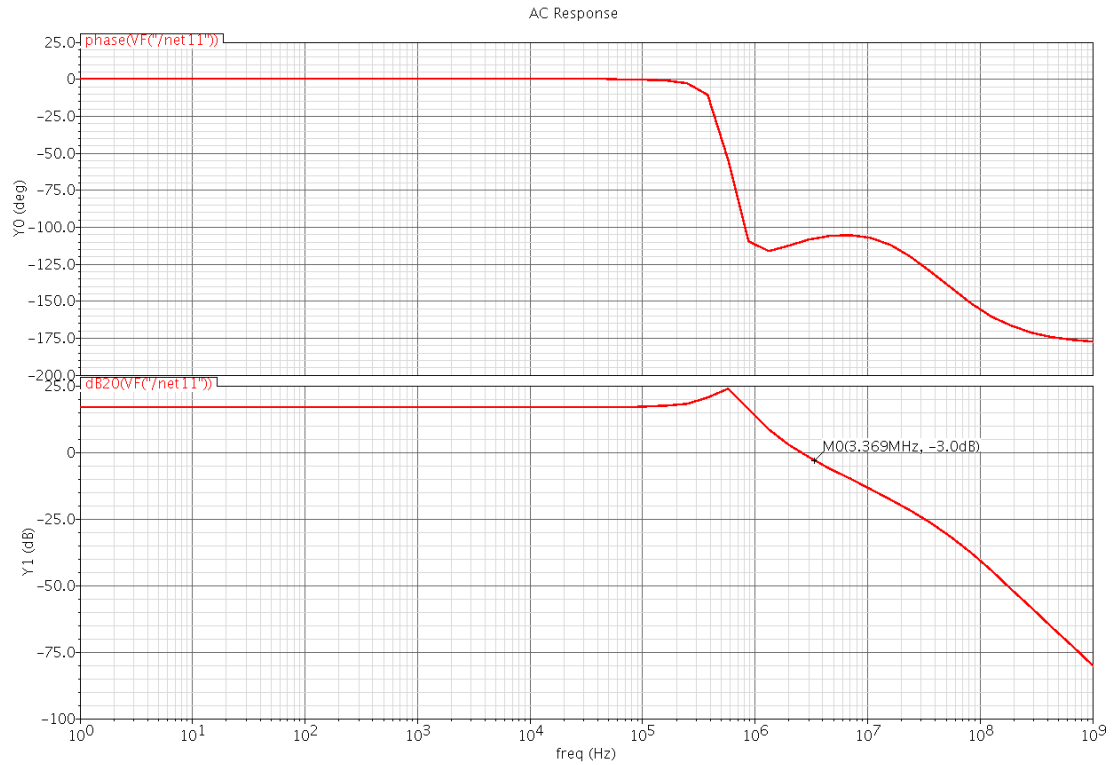


Figura 12: respuesta transitoria del sistema con $N=6.92$, $K=3.69\text{MHz}$

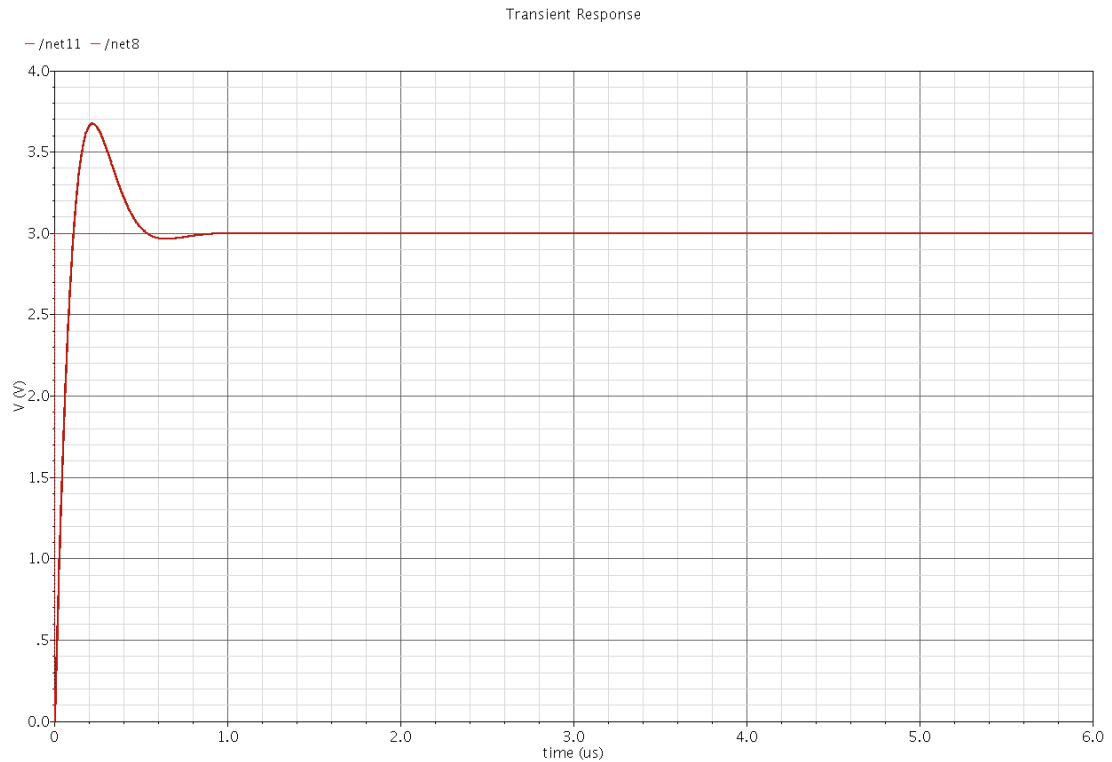


Figura 13: respuesta a la entrada escalón del sistema con $N=1$, tiempo de set = 1μ

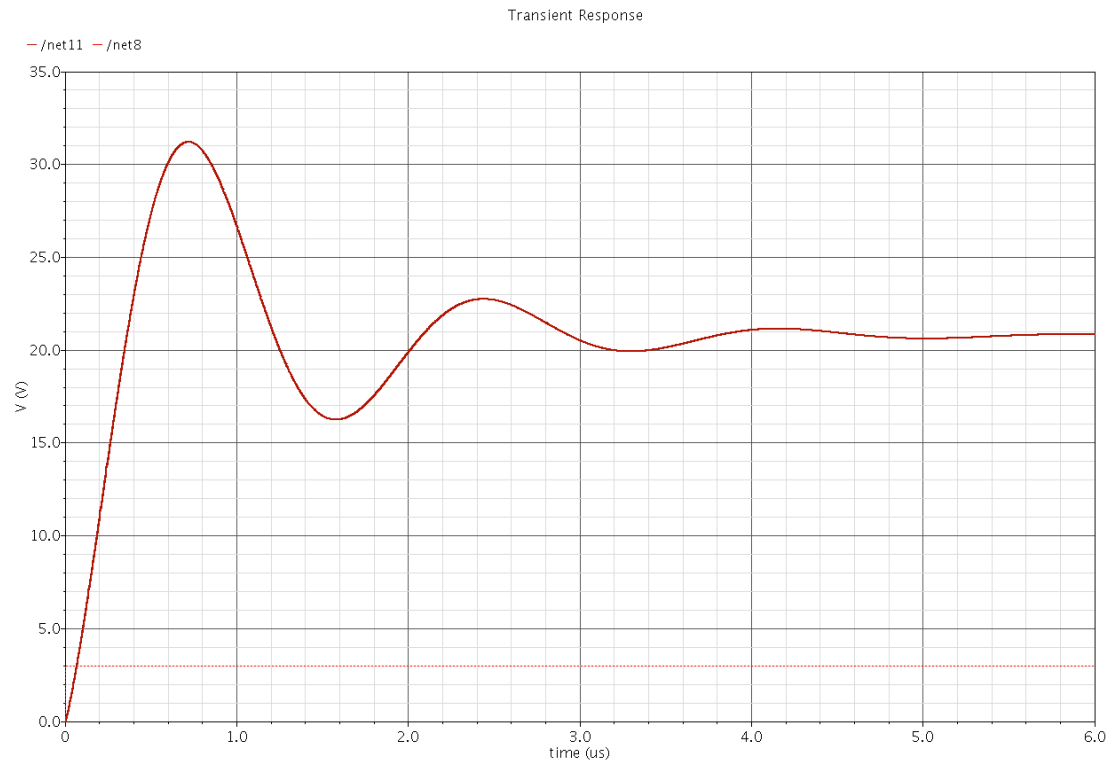


Figura 14: respuesta a la entrada escalón del sistema con $N= 6.92$. Tiempo de set 3.3μ

2. DISEÑO DEL MODELO TRANSITORIO DEL CP-PLL.

Con los parámetros obtenidos (K_o , C_1 , C_2 , R etc.) En el modelo en AC es posible utilizarlos para crear los modelos transitorios.

2.1. Modulo VerilogA del PFD

El modelo del PFD está compuesto por 2 flip flops y una compuerta and, ya que se quiere construir un PFD de cuatro estados, el diagrama de estados se muestra en la figura siguiente:

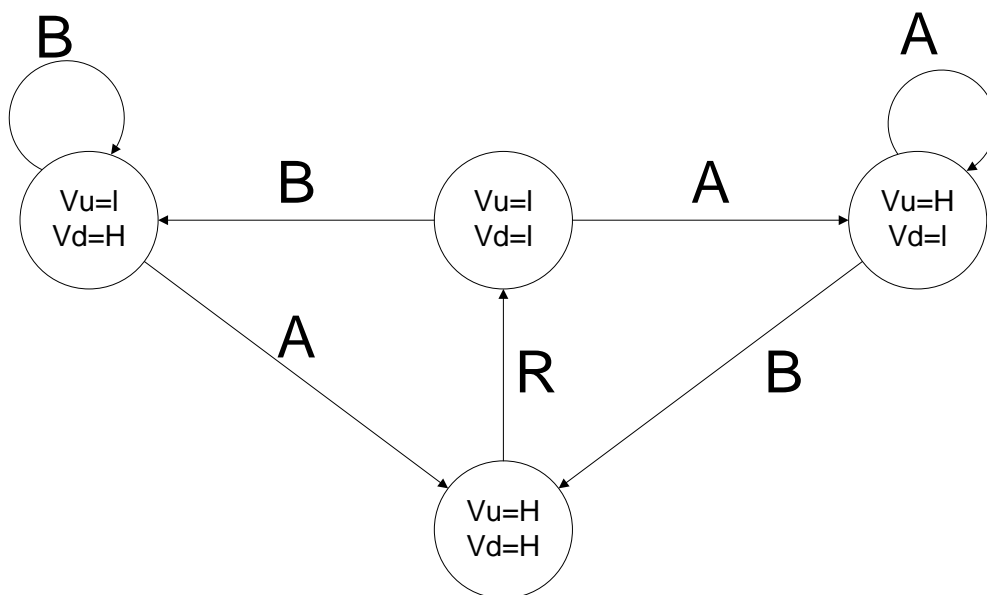


Figura 15: Diagrama de estados del PFD.

El diagrama esquemático del PFD se muestra a continuación, y más adelante se muestra el código verilogA de cada uno de estos bloques.

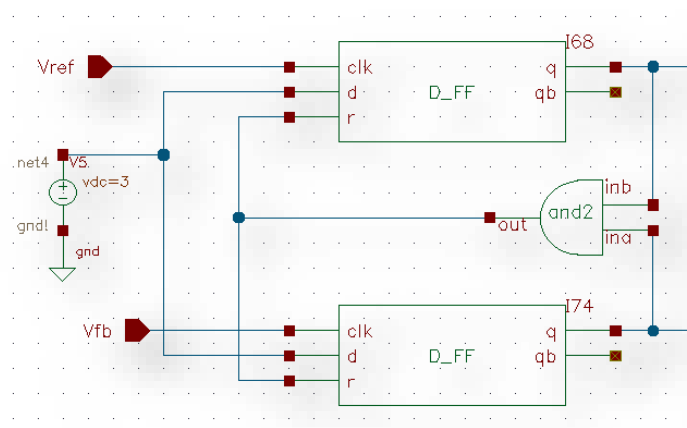


Figura 16: PFD de tres estados.

2.1.1 Modulo VerilogA del Flip Flop

```
// VerilogA for PLL_proyecto3, PFD_MaqEdo3, veriloga
// POR: Gustavo Guzmán Rosales.
```

```
`include "constants.vams"
`include "disciplines.vams"

module PFD_MaqEdo3(d, clk, r, q, qb);
input d, clk, r;
output q, qb;
electrical d, clk, r, q, qb;

parameter real vdd=3;
parameter real rf_time=15p;
integer reset, data, clockdata;

analog begin
    @(timer(0)) begin
        if (V(r)>(vdd/2))
            begin
                reset=0;
                clockdata=data*reset;
            end
        else
            reset=1;
            if (V(d)>(vdd/2))
                data=1;
            else
                data=0;
            if (V(clk)>(vdd/2))
                clockdata=data*reset;
            end
        @(cross((V(r) - (vdd/2)), +1))
        begin
            reset=0;
            clockdata=data*reset;
        end
    end
end
```

```

end
@(cross((V(r)-(vdd/2)),-1))
    reset=1;
@(cross((V(d)-(vdd/2)),+1))
    data=1;
@(cross((V(d)-(vdd/2)),-1))
    data=0;
@(cross((V(clk)-(vdd/2)),+1))
    clockdata=data*reset;
V(q)<+transition((clockdata*vdd),0, rf_time);
V(qb)<+transition(!clockdata*vdd),0, rf_time);
end
endmodule

```

2.1.2

2.1.3

2.1.4

2.1.5 Modulo VerilogA del AND

```

// VerilogA for PLL_proyecto3, and2, veriloga
// POR: Gustavo Guzmán Rosales.

`include "constants.vams"
`include "disciplines.vams"
//`include "std.vams"
//`include "const.vams"

module and2(ina, inb ,out);
input ina, inb;
output out;
electrical ina, inb, out;
parameter real vdd=3;
parameter real rf_time=0.0001p;
integer dataa, datob, andfl;
analog begin
    @(timer(0)) begin
        if (V(ina)>(vdd/2))
            dataa=1;
        else
            dataa=0;
        if (V(inb)>(vdd/2))
            datob=1;
        else
            datob=0;
        end
    @(cross(V(ina)-vdd/2,+1))

```

```
        dataa=1;
    @(cross(V(ina)-vdd/2,-1))
        dataa=0;
    @(cross(V(inb)-vdd/2,+1))
        datob=1;
    @(cross(V(inb)-vdd/2,-1))
        datob=0;
    andfl=(dataa&&datob);
    V(out) <+ transition(andfl*vdd,0,rf_time, rf_time);
end
endmodule
```

2.2.

2.3.

2.4.

2.5.

2.6.

2.7.

2.8.

2.9.

2.10.

2.11.

2.12. Modulo VerilogA del CP

```
// VerilogA for PLL_proyecto3, CP_n, veriloga
// POR: Gustavo Guzmán Rosales.

`include "constants.vams"
`include "disciplines.vams"
```

```
module CP (Iout, Down, Up);

input Up,Down;
output Iout;

electrical Up,Down,Iout;

parameter real v_th =0.0001;
parameter real v_max =0.3;
parameter real v_min =0.0001;
parameter real Ip =1u;
parameter real Delay =50p;
parameter real tt = 50.0p;
integer state;

analog begin
    @(cross(V(Up)-v_th, 1))
    begin
        state = -1;
    end
    @(cross(V(Up)-v_th, -1))
    begin
        state =0;
    end
    @(cross(V(Down)-v_th, 1))
    begin
        state = 1;
    end
    @(cross(V(Down)-v_th, -1))
    begin
        state =0;
    end
    @(cross(V(Iout)-v_max, 1))
    begin
        state =0;
    end
    @(cross(V(Iout)-v_min, -1))
    begin
        state =0;
    end
    I(Iout)<+transition(Ip*state, Delay, tt);
end
endmodule
```

2.13.

2.14.

2.15.

2.16.

2.17.

2.18. Modulo VerilogA del LPF

```
// VerilogA for PLL_Proyecto, LF, veriloga
// POR: Gustavo Guzmán Rosales.

`include "constants.vams"
`include "disciplines.vams"

module LF(In, Out);
input In;
output Out;

electrical In,Out;

parameter R =2.16e3;
parameter C1=6.45e-11;
parameter C2=3.22e-12;

analog
begin
    V(Out)<+laplace_nd(V(In),{1,R*C1},{0,(C1+C2),(R*C1*C2)});
end

endmodule
```

2.19. Modulo VerilogA del VCO

```
// VerilogA for PLL_proyecto3, VCO, veriloga
// POR: Gustavo Guzmán Rosales.

`include "constants.vams"
`include "disciplines.vams"

module VCO(out, in);
input in;
output out;
electrical in,out;

parameter real Vmin=0; //de la grafica 0 -> Wo
parameter real Vmax= 0.2 from (Vmin:inf);
parameter real Fmin=133M from (0:inf);
parameter real Fmax = 1.6G from (0:inf);
parameter real Vlo=0, Vhi=3;
parameter real tt=100p from(0:inf);
parameter real jitter=3.7p;
parameter real ttol=0.1p ;

real freq;
real fase;
real dt;
integer n , seed,Vout;

analog begin

    @(initial_step) seed=-561;
    freq=(V(in)-Vmin)*(Fmax-Fmin)/(Vmax-Vmin)+Fmin;

    dt=abs(1.414*jitter*$rdist_normal(seed,0,1));
    fase=idtmod(freq,0.0,1.0,-0.5);

    @(cross(fase - 0.25,1,ttol))
    begin
        Vout=Vhi;
    end
    @(cross(fase+0.25,1,ttol))
    begin
        Vout=Vlo;
    end

    V(out)<+transition(Vout,0+dt,tt);
end

endmodule
```

2.20. Modulo VerilogA del N

```
// VerilogA for PLL_Proyecto2, Div, veriloga
// POR: Gustavo Guzmán Rosales.

`include "constants.vams"
`include "disciplines.vams"

module Div(in,out);
input in;
output out;

electrical in,out;
parameter real v_high=3;
parameter real v_low =0;
parameter real td = 100p ;
parameter real tt = 100p from (0:inf) ;
parameter real ttol=1p from (0:inf);
parameter real N_max =25;
parameter real N_min = 1;
parameter real T_jump=8e-6;
parameter real jitter =0 from [0:inf);
integer cont;
integer n;
integer M,seed;
real dt;
analog begin
    @(initial_step) seed=-311;
    @(cross(V(in),-1))begin
        cont = cont+1;
        if($abstime>T_jump)M=N_max;
        else M=N_min;
        if(cont>=M) cont=0;
        n=(2*cont>=M);
        dt=jitter* $rdist_normal(seed,0,1);
    end
V(out) <+ transition (n ? v_high : v_low, dt,tt);
end
endmodule
```

Con los módulos anteriores es posible hacer un solo bloque el cual será el PLL en lazo abierto, para que se pueda realimentar de forma interna y externa al Soc, ya que es una de las especificaciones de diseño.

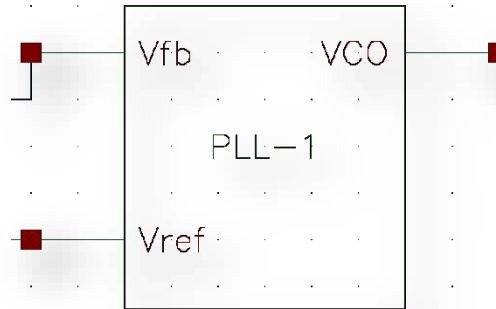


Figura 17: Símbolo del CP-PLL

3. INTEGRACIÓN DEL SOC.

Con el modelo transitorio del CP-PLL anteriormente, es posible integrarlo en un SoC, para hacer esto es necesario crear algunos bloques mas, como son, buffers, multiplexores y el cristal interno.

3.1. Modulo VerilogA del buffer.

```
// VerilogA for PLL_proyecto3, Buffer_Tri, veriloga
`include "constants.vams"
`include "disciplines.vams"

module Buffer_Tri(I,O,S);
input I,S;
output O;

electrical I,O,S;
real Out;

analog begin

if(V(S)>0)
    Out=V(I);
else
    Out=0;

V(O)<+ Out;

end

endmodule
```

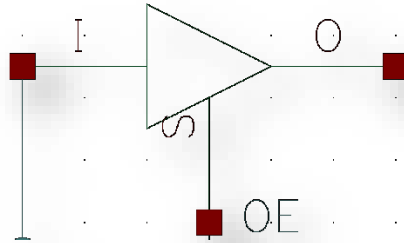



Figura 18: Simbolo del Buffer

3.2.

3.3.

3.4. Modulo VerilogA del multiplexor 2-1.

```
// VerilogA for PLL_proyecto3, MUX, veriloga
`include "constants.vams"
`include "disciplines.vams"

module MUX(In0,In1 ,Out, Sel);
input In0,In1,Sel;
output Out;
electrical In0,In1,Out,Sel;

real OutV;

analog begin
if (V(Sel)>0)
  OutV = V(In1);
else
  OutV = V(In0);

V(Out) <+ OutV;
end

endmodule
```

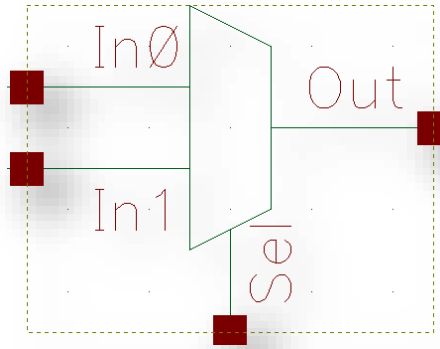


Figura 19: Símbolo del Multiplexor.

3.5. Esquemático del SoC:

Con todos los bloques construidos anteriormente se construye el diagrama esquemático final. En la imagen siguiente se muestra el diagrama final del circuito.

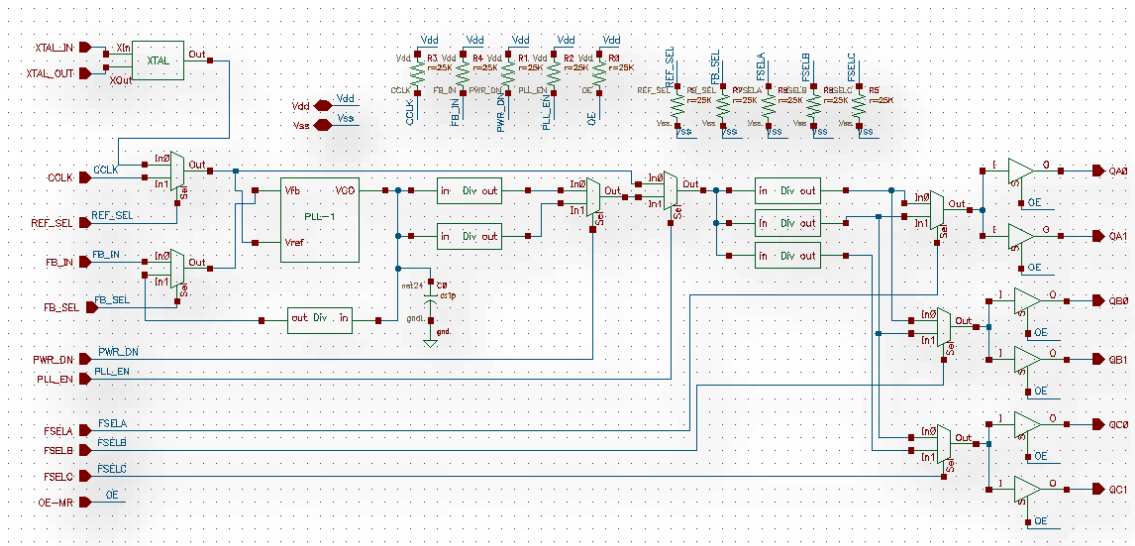


Figura 20: Diagrama esquemático del SoC.

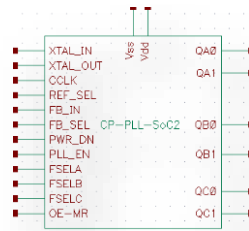


Figura 21: Símbolo del SoC.

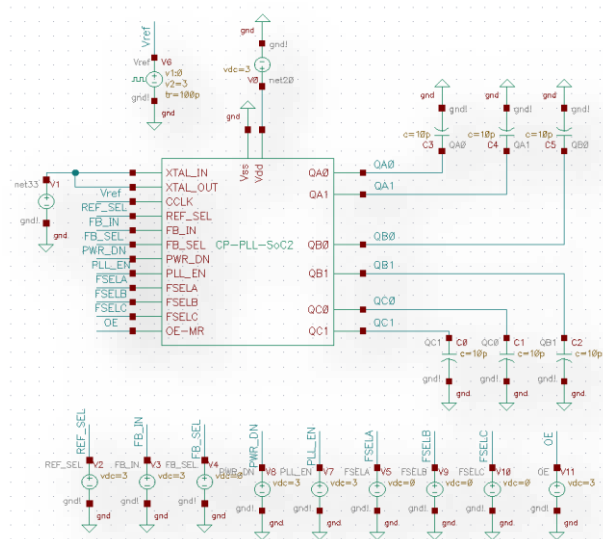
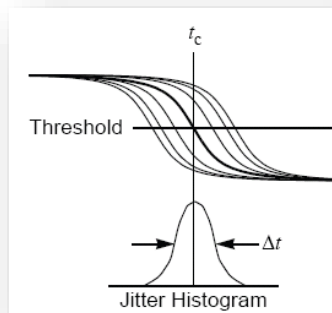
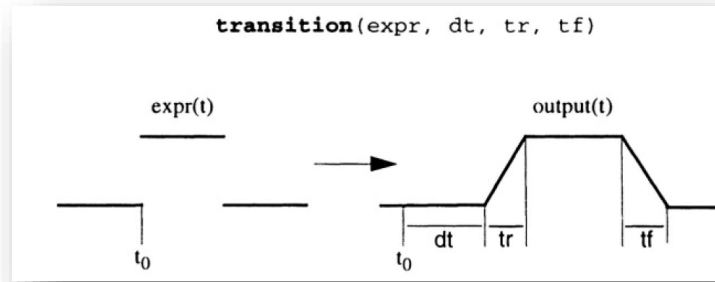


Figura 22: Diagrama esquemático del TestBench. Se observa una capacitancia de carga de 10pF

4. MODELOS CON JITTER

Para la construcción de los modelos incluyendo el jitter, primero fue necesario conocer que el jitter va a mover el periodo de la señal, por lo tanto se puede poner un tiempo aleatorio en el retardo de la señal, esto en la función **transition**, que genera la señal de salida, debido a que el jitter es aleatorio, se deberán generar números aleatorios, pero desacuerdo a la imagen siguiente



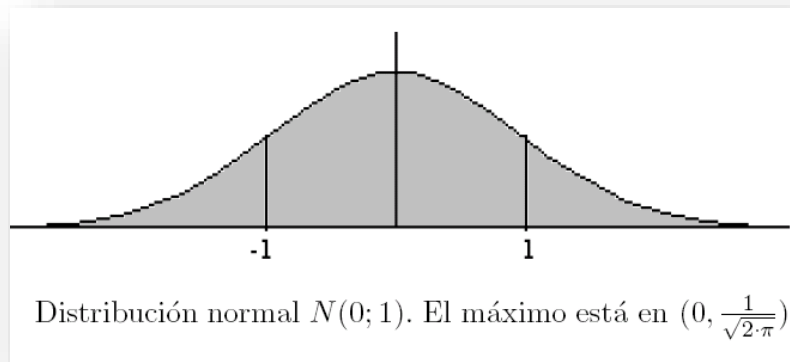


Los números aleatorios generados deben ser muchos cerca del valor ideal, y pocos lejos del valor ideal. Para hacer uso de la instrucción **dist_normal** esta instrucción genera números aleatorios o pseudo aleatorio.

dist_normal(seed, mean, standard_deviation)

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}}$$

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{x^2}{2}}$$



Donde:

mean_expression value of zero (0) y *stdev_expression* en uno (1) genera Gaussian distribution. Para que en cada simulación se generen los mismos números aleatorios, se utiliza un valor de seed igual en cada simulación.

5. RESULTADOS DE SIMULACIÓN:

Los resultados de las mediciones se muestran en las tablas siguientes:

DIVISOR	Periodo	Frecuencia	Lock Time <4us	Volt-Lock	Volt-Lock-Real
2	7.50E-09	1.33E+08	1.19	0	4.60E-05
4	3.75E-09	2.67E+08	1.2	0.01826	1.82E-02
6	2.50E-09	4.00E+08	2	0.0364	3.54E-02
8	1.88E-09	5.33E+08	2.25	0.05453	5.46E-02
12	1.25E-09	8.00E+08	2.5	0.09093	9.09E-02
16	9.38E-10	1.07E+09	3	0.1277	1.27E-01
24	6.25E-10	1.60E+09	3.8	0.2	2.00E-01
PROMEDIO	-----	-----	2.277142857	0.075402857	0.075177857

DIVISOR	Periodo	Frecuencia	phse jitter <20ps	SPO <20ps/ jitter	SPO Sin Jitter	SKEW <50ps
2	7.50E-09	1.33E+08	18.58	13.04	15	13.03
4	3.75E-09	2.67E+08	15.71	16.24	15	12.74
6	2.50E-09	4.00E+08	11.78	18.03	15	13.65
8	1.88E-09	5.33E+08	9.969	12.51	15	13.65
12	1.25E-09	8.00E+08	12.47	12.02	15	13.65
16	9.38E-10	1.07E+09	18.95	7.955	15	12.8
24	6.25E-10	1.60E+09	12.32	16.78	15	13
PROMEDIO	-----	-----	14.25414286	13.79642857	15	13.2171429

DIV	T1	T2	T3	T4	T5	T6	T7	T8	J1	J2	J3	J4	J5	J6	J7
2	7504	7498	7506	7503	7494	7500	7500	7502	6	8	3	9	6	0	2
4	3740	3752	3750	3753	3746	3756	3747	3754	12	2	3	7	10	9	7
6	2507	2492	2498	2500	2504	2506	2490	2505	15	6	2	4	2	16	15
8	1871	1878	1875	1878	1868	1889	1869	1868	7	3	3	10	21	20	1
12	1241	1253	1258	1245	1244	1250	1253	1248	12	5	13	1	6	3	5
16	935.3	937.3	937.9	942.2	934.3	940	938.6	935.3	2	0.6	4.3	7.9	5.7	1.4	3.3
24	627.8	637.6	619.5	625.7	630.7	626.7	638.2	633.3	9.8	18.1	6.2	5	4	11.5	4.9

Promedio de J

4.857142857

7.142857143

8.571428571

9.285714286

6.428571429

3.6

5.1. Resultados de simulación transitoria.

Las frecuencias de las señales de salida del SoC(Q0A, Q1A, Q0B, Q1B, Q0C, Q1C), son dependientes del estado o activación de las señales de entrada(PWR_DN, EO, FSEL_C, FSEL_A, FSEL_B, PLL_EN) para esta simulación las señales de entrada se encuentran de la siguiente manera: PWR_DN=0, EO=3, FSEL_C=0, FSEL_A=0, FSEL_B=0, PLL_EN=3. Por lo tanto a la salida las señales tienen los siguientes periodos:

Q0A=3.749ns, Q1A=3.749ns, Q0B=3.749ns, Q1B=3.749ns, Q0C=7.5ns, Q1C=7.5ns, los resultados de esta simulación se observan en las graficas siguientes:



Figura 23: simulación transitoria del Soc, donde se observa las señales de salida Q0A, Q1A, Q0B, Q1B, Q0C, Q1C

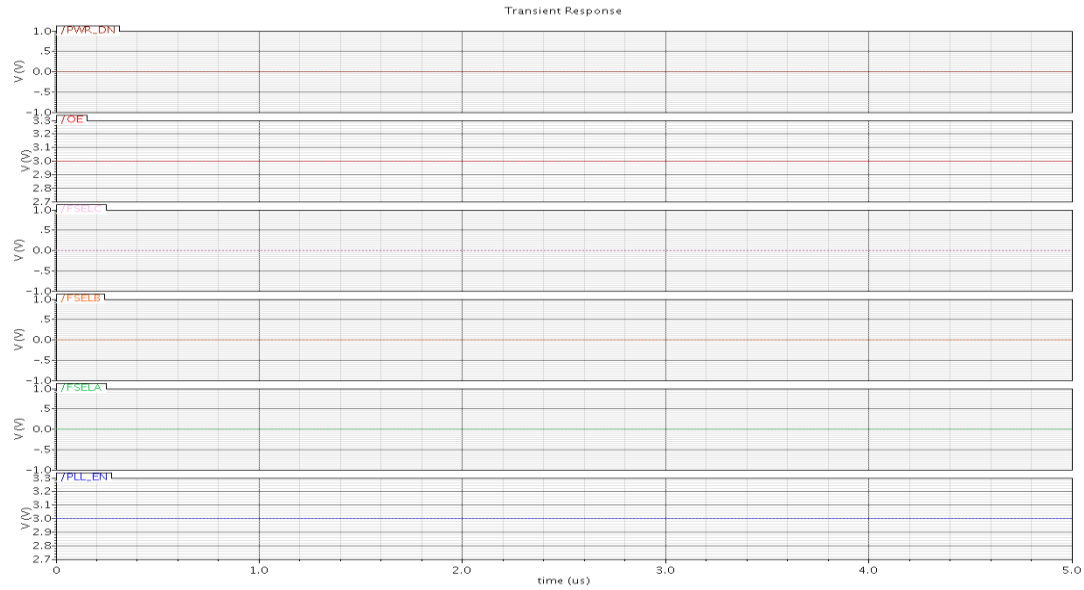
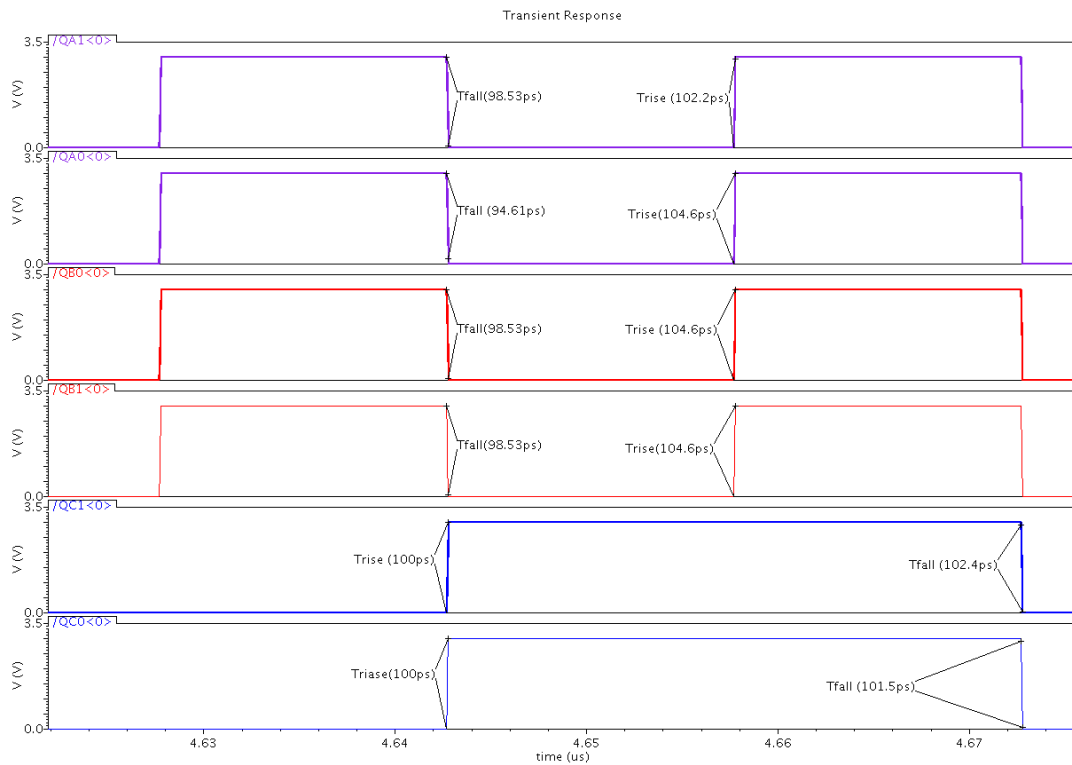


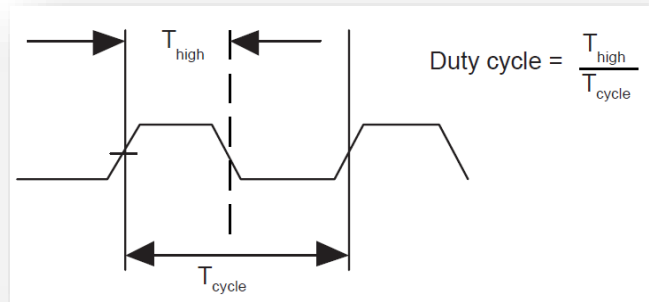
Figura 24: Simulación transitoria del SoC, donde se observa el estado de las señales de entrada.

5.2. Medición del Duty Cycle

En la grafica mostrada a continuación:

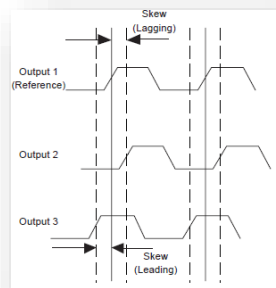


En la grafica anterior se observa que el duty cycle, es del 50%, siendo el duty cycle la razón en la que la señal de salida se encuentra en alto, expresada en porcentaje.



5.3. Medición Del Output Skew

El Skew es la variación en la llegada de dos señales que se esperaba que llegaran al mismo tiempo.



En la grafica siguiente se muestra la medición del Skew de las señales de salida del SoC.

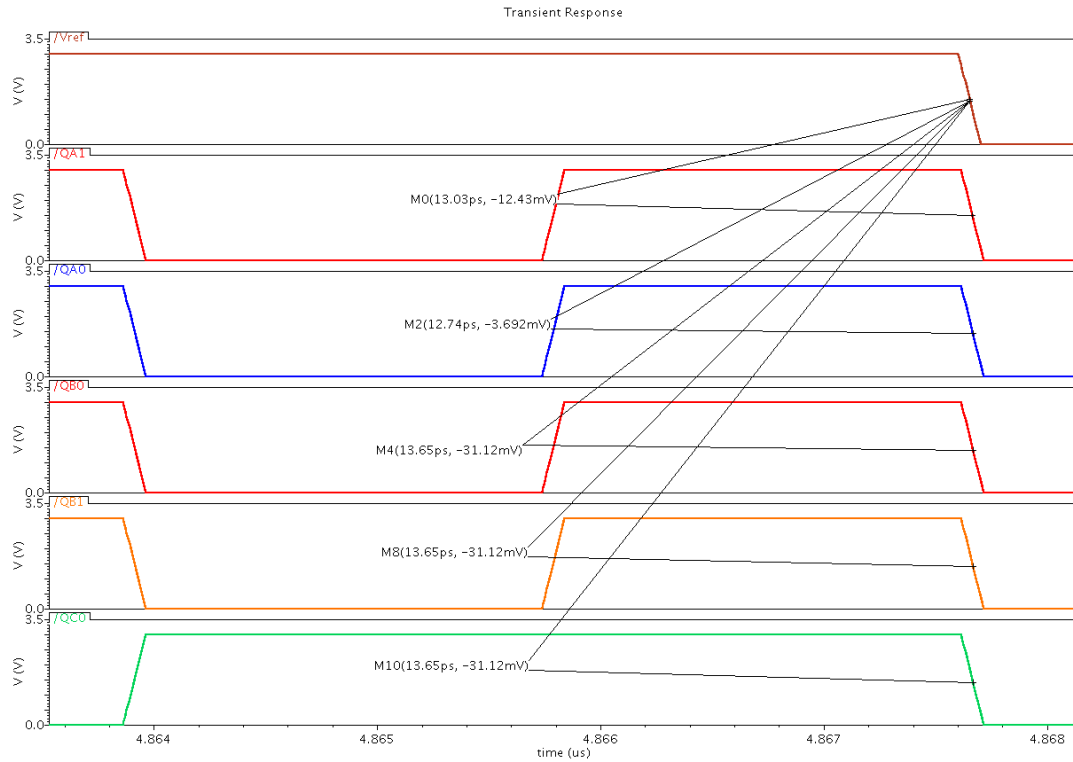


Figura 25: Medición del Skew =13ps

5.4. Medición del tiempo de Rise & Fall.

En la grafica mostrada a continuación se muestra la medición de de los tiempos de Rise y de Fall, para las señales de salida.

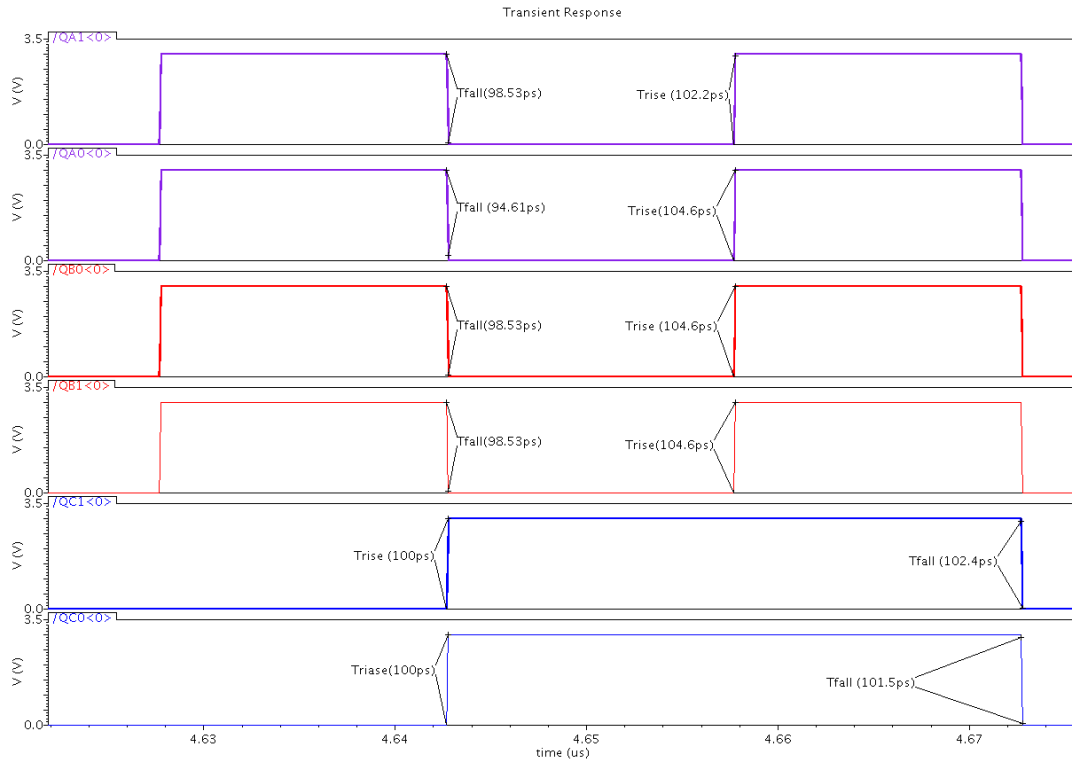


Figura 26: Medicon de los tiempos de fall y rise =100p y Duty Cycle de 50%

5.5. Resultado de Simulación Transitoria:

En las próximas graficas se muestra como responde el PLL en el tiempo cuando se varia el valor de N.

5.5.1 Con Divisor de 2:

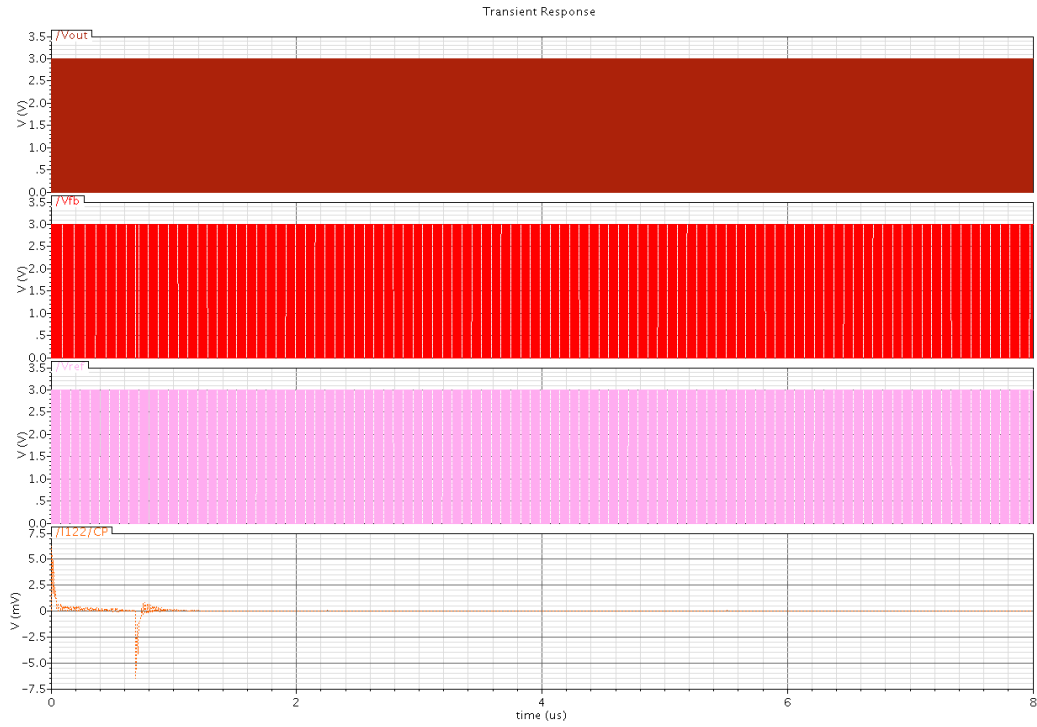


Figura 27: respuesta transitoria.

5.5.2 Con Divisor de 4:

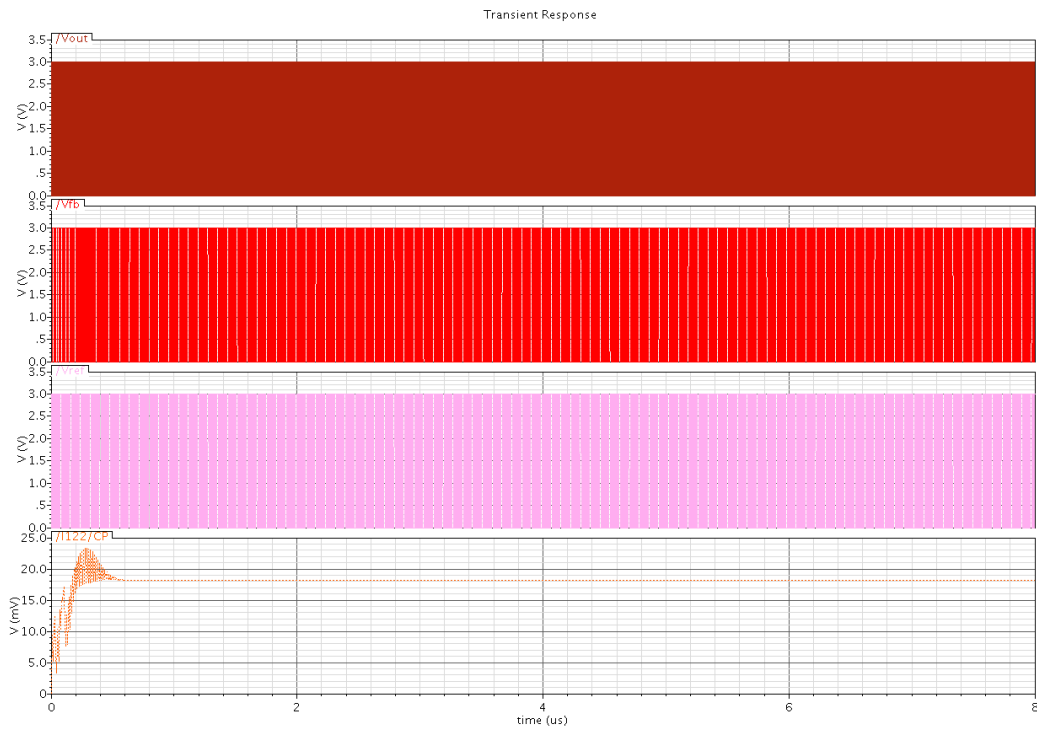


Figura 28 respuesta transitoria.

5.5.3 Con Divisor de 6:

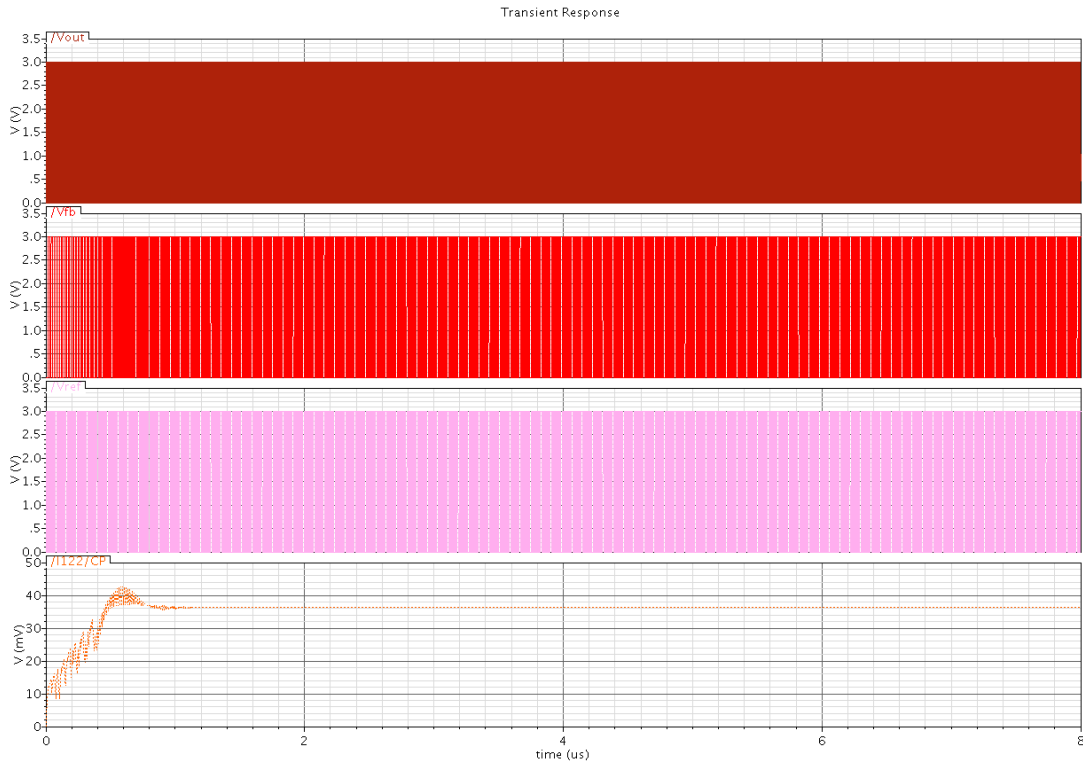


Figura 29 respuesta transitoria.

5.5.4 Con Divisor de 8:

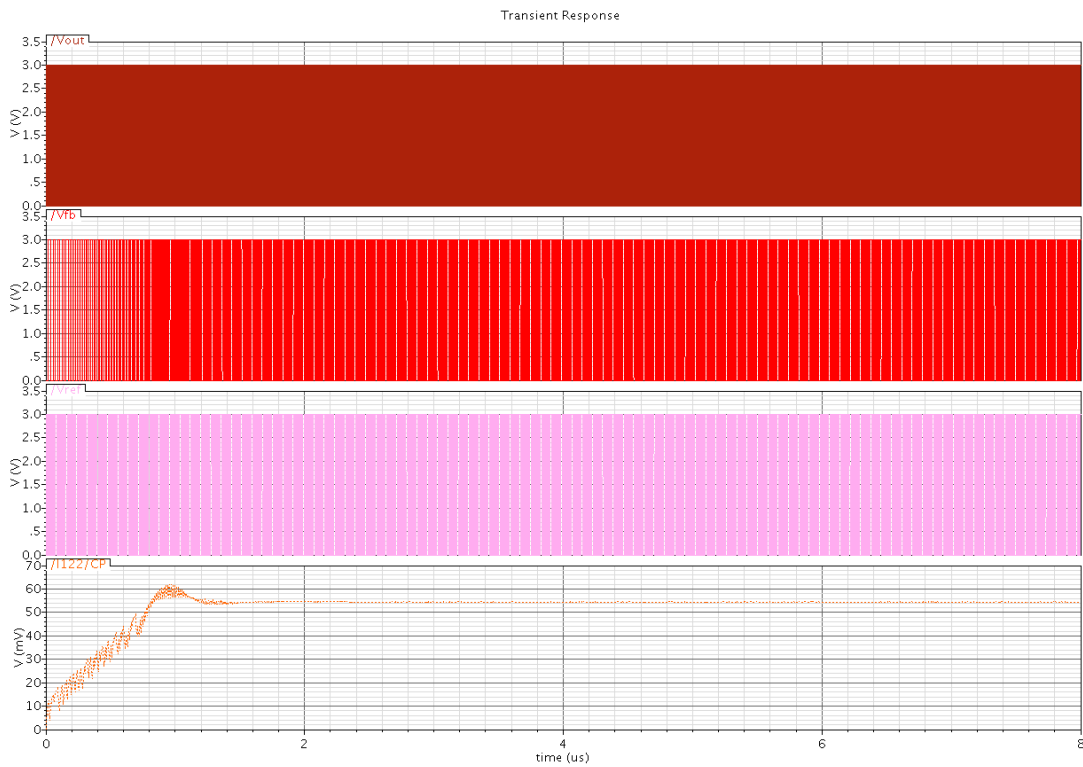


Figura 30 respuesta transitoria.

5.5.5

5.5.6 Con Divisor de 24:

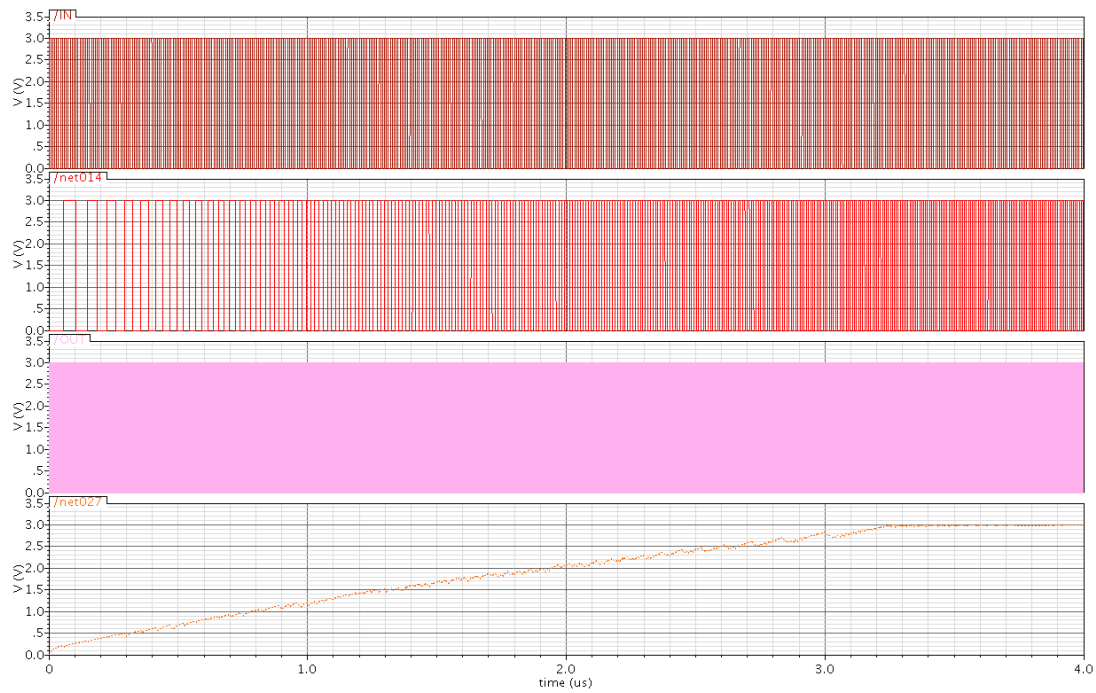


Figura 31 respuesta transitoria.

5.6. Medición del Lock Time:

5.6.1 Con Divisor de 2:

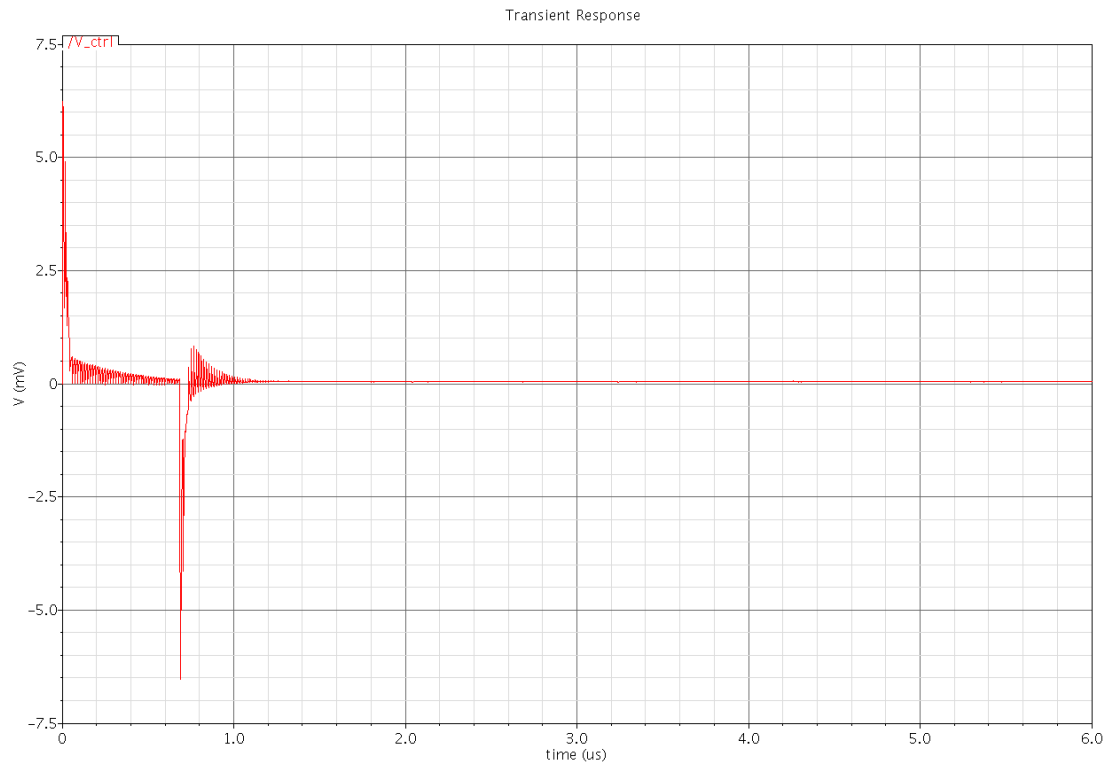


Figura 32: Medición del Lock Time =1.25us

5.6.2 Con Divisor de 4:

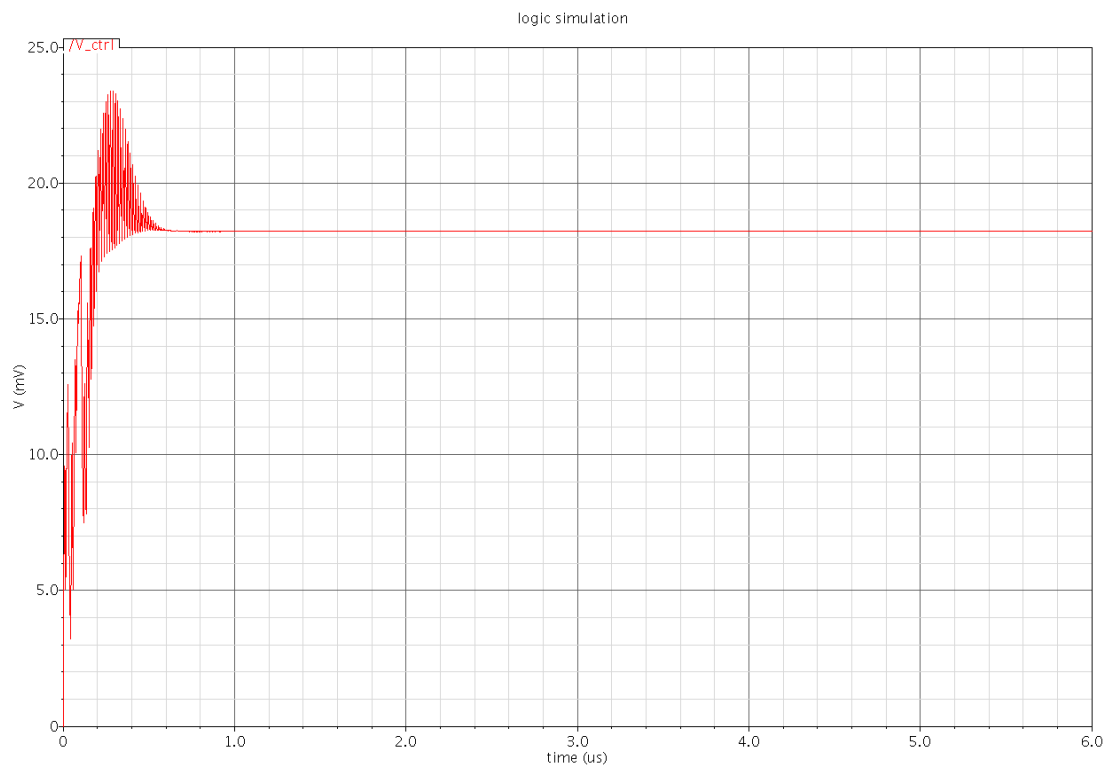


Figura 33 Medición del Lock Time=0.75us

5.6.3 Con Divisor de 6:

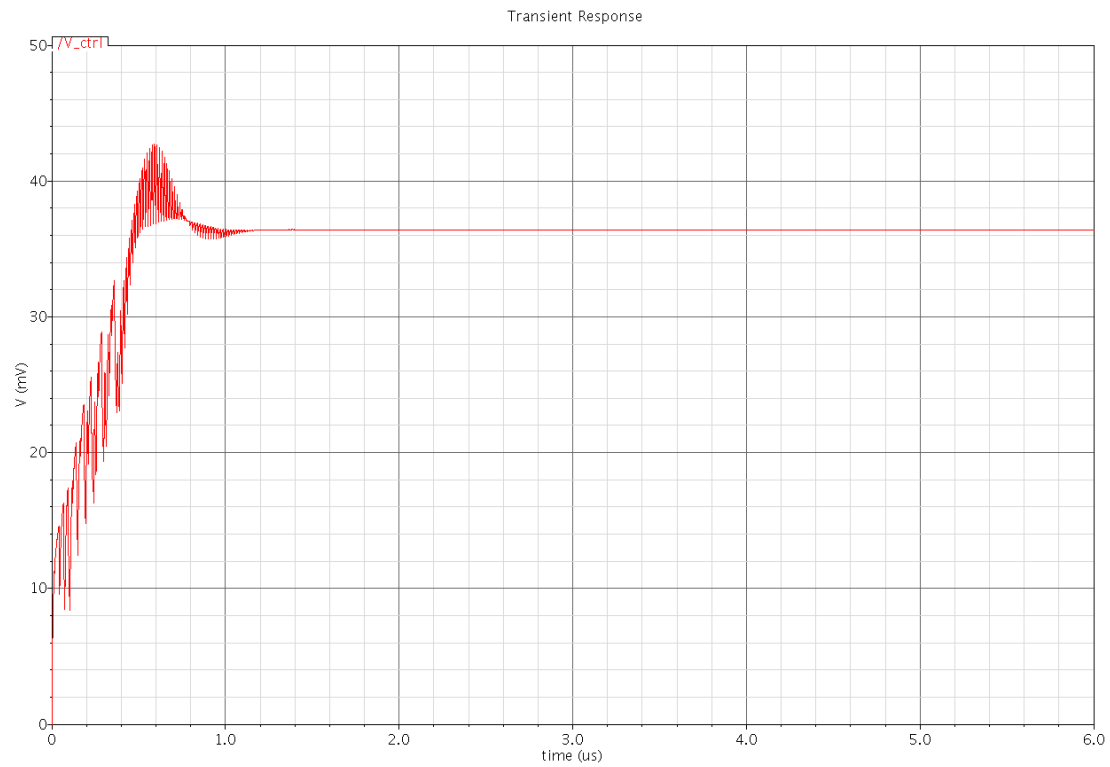


Figura 34 Medición del Lock Time=1.5us

5.6.4 Con Divisor de 8:

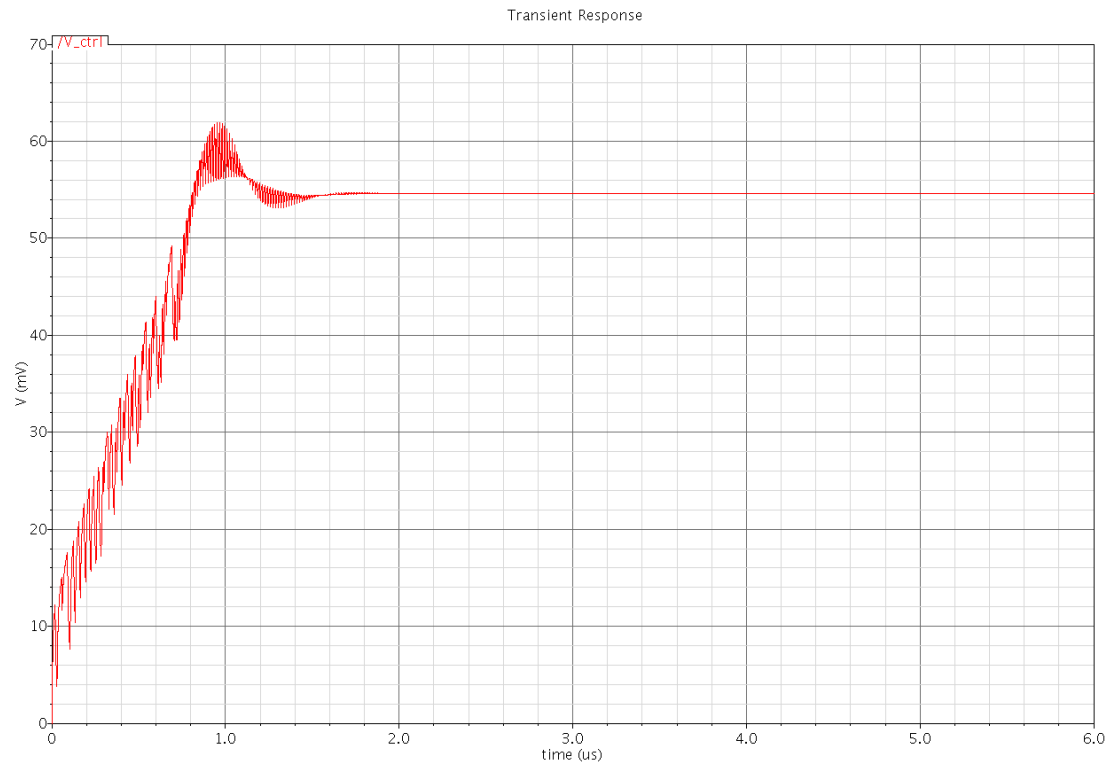


Figura 35 Medición del Lock Time=2us

5.6.5 Con Divisor de 12:

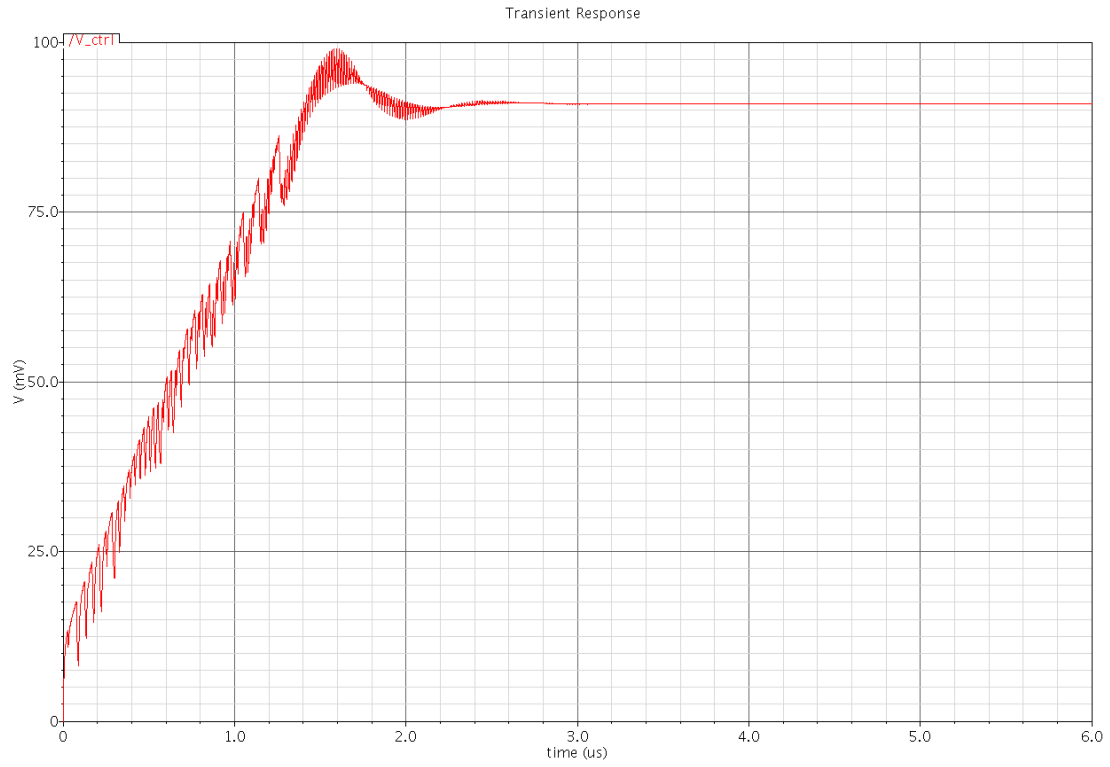


Figura 36 Medición del Lock Time= 2.75us

5.6.6 Con Divisor de 16:

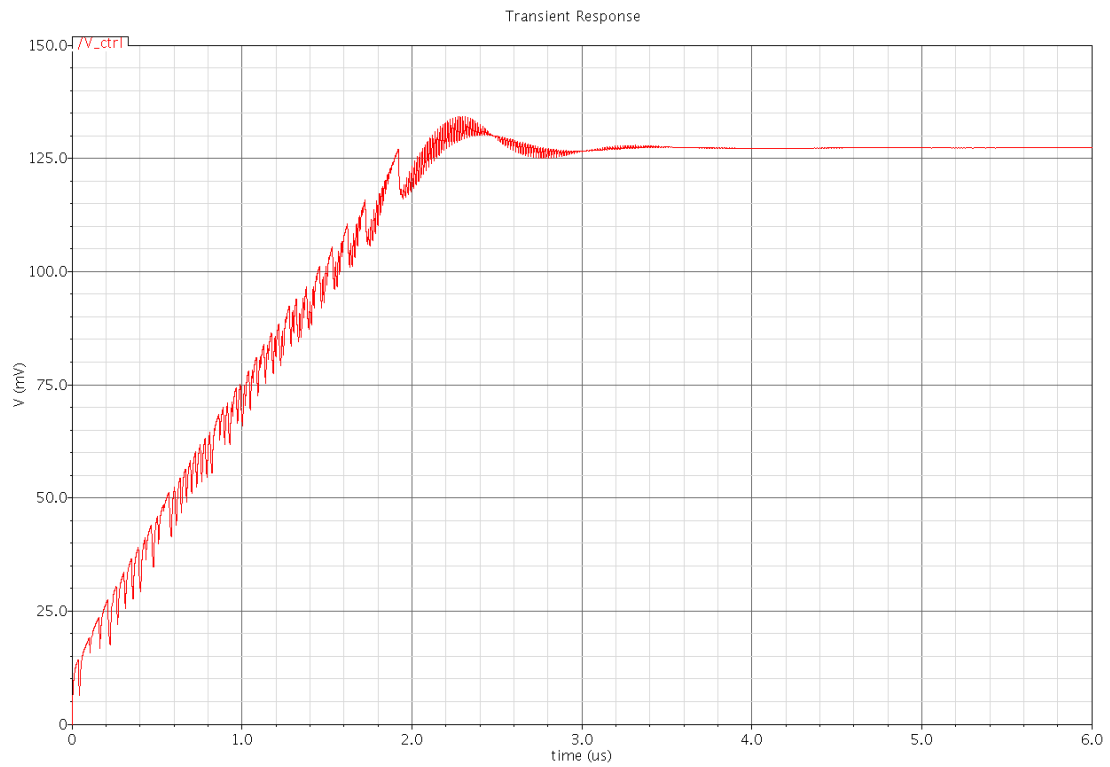
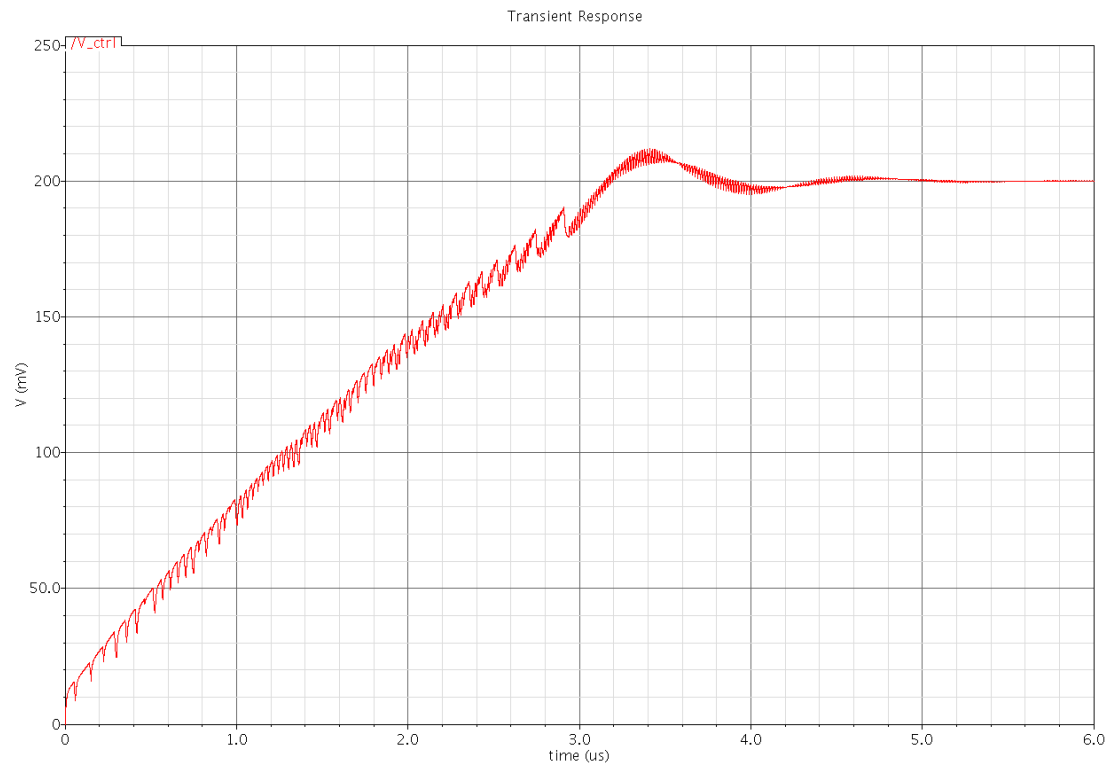


Figura 37 Medición del Lock Time=3.6us

5.6.7 Con Divisor de 24:

**Figura 38 Medición del Lock Time=3.9us**

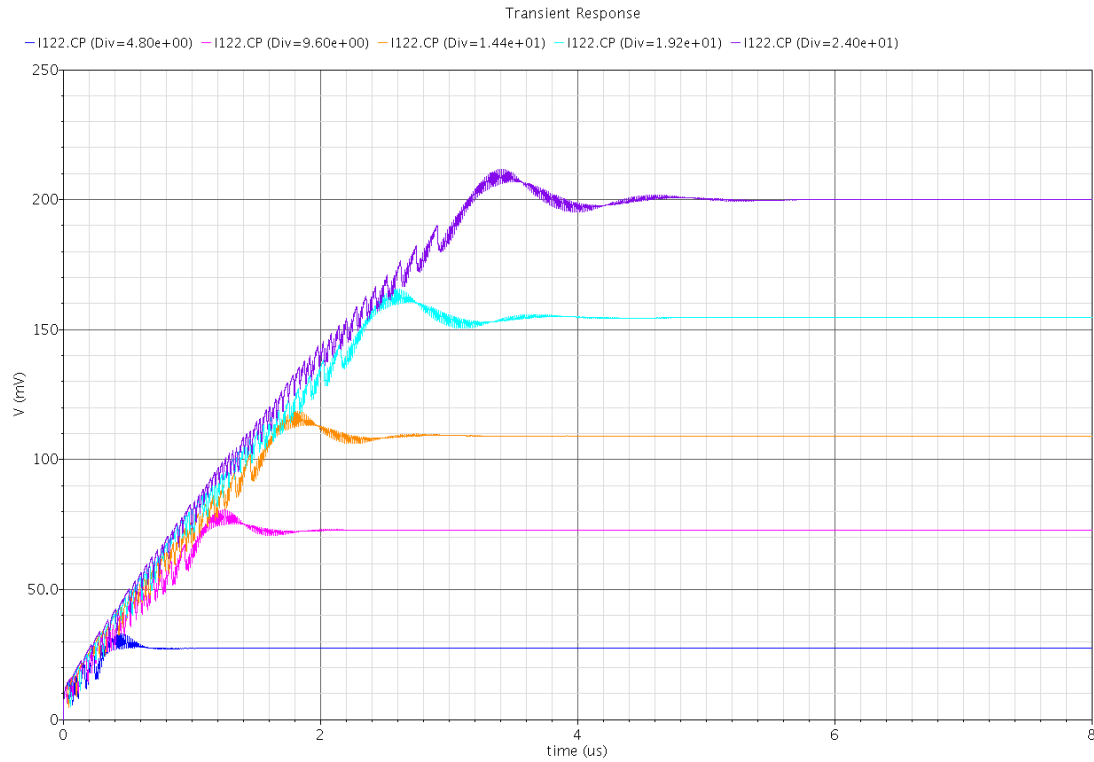
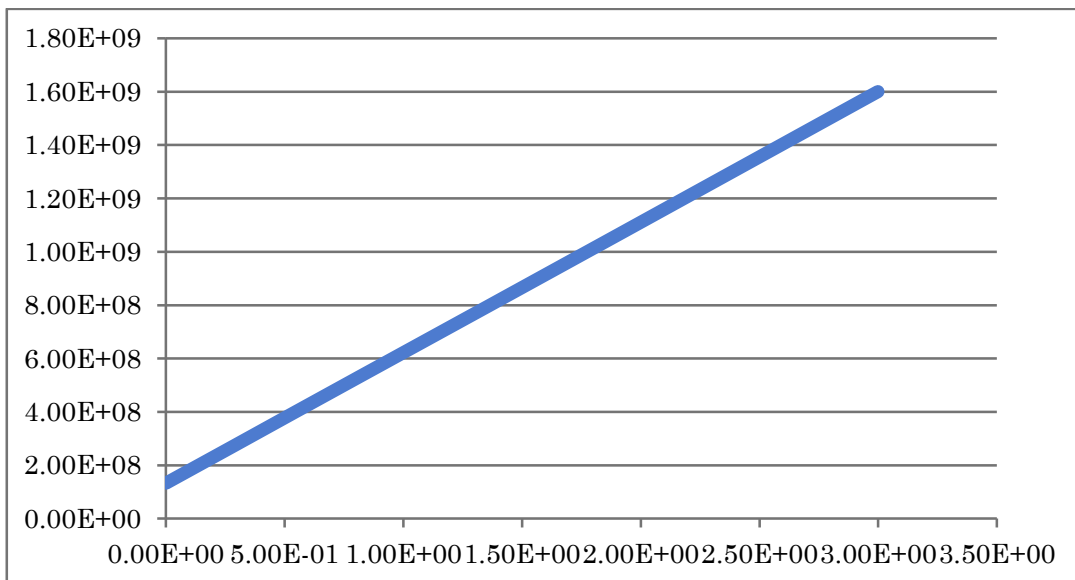


Figura 39: Grafica de comparación entre diferentes valores de N, se observa la linealidad del VCO.

En la grafica siguiente se muestra como funciona el VCO en simulación, lo cual demuestra que responde tal y como se diseño, muy lineal en el rango que se requiere.



6.

7.

8. MEDICIÓN DEL CYCLE JITTER:

El jitter ciclo a ciclo es la variación que hay de un periodo a otro, en las graficas que se presentan a continuación se realizaron las mediciones de los periodos, de la se la de salida del PLL, con diferentes valores de N, para después en base a estos datos determinar el jitter ciclo a ciclo.



Con Divisor de 2:

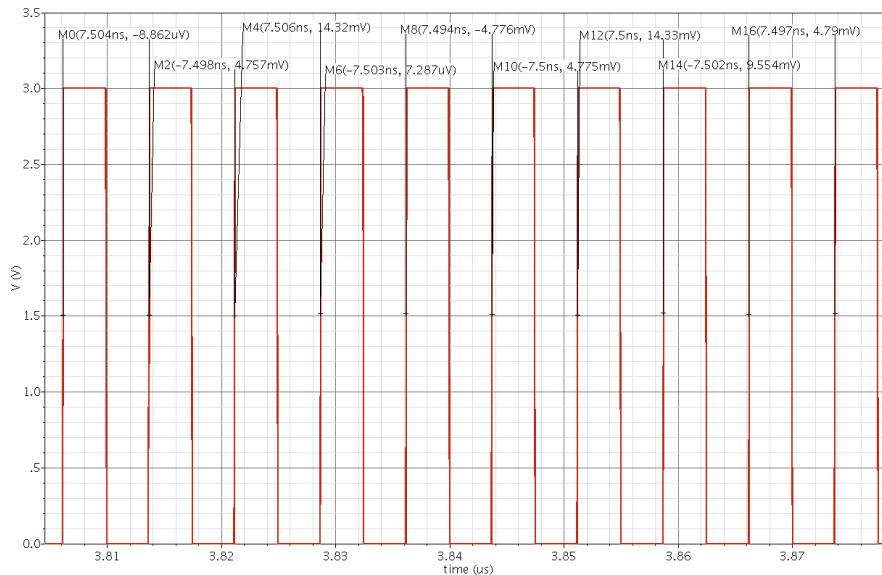


Figura 40: Medición del Cycle Jitter

8.1.1 Con Divisor de 4:

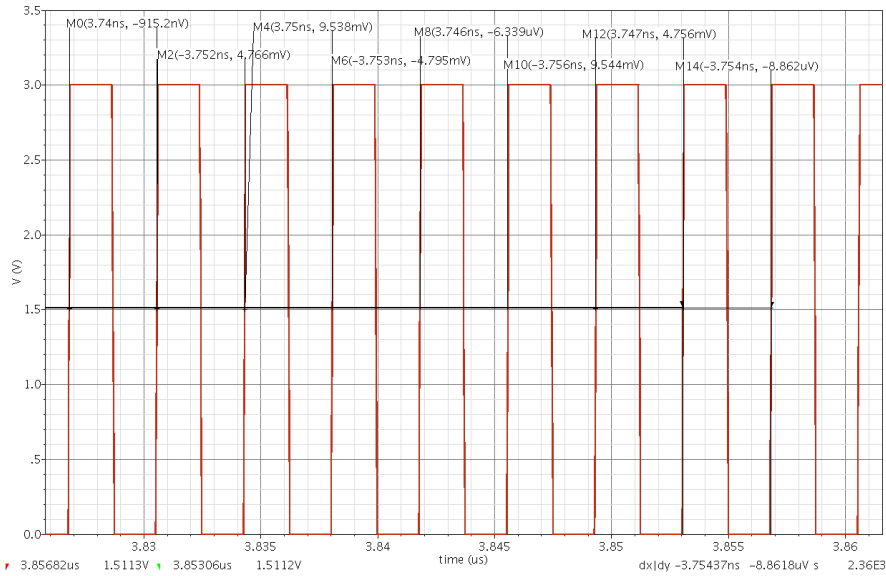


Figura 41 Medición del Cycle Jitter

8.1.2 Con Divisor de 6:

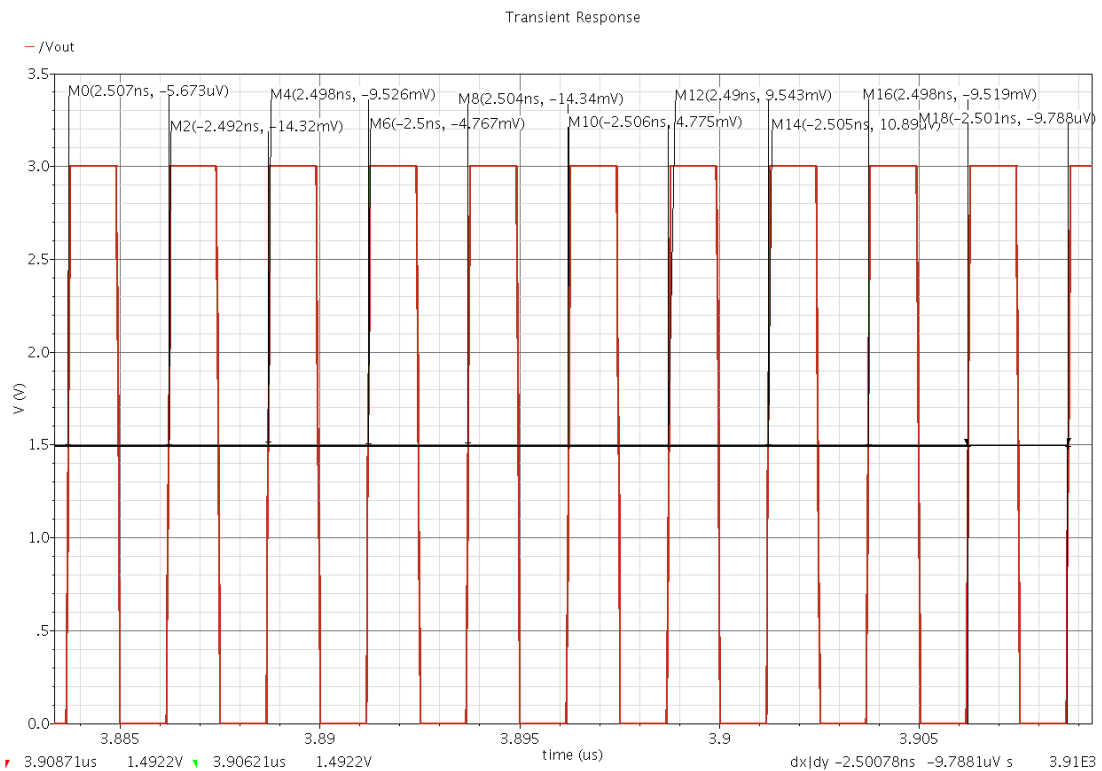


Figura 42: Medición del Cycle Jitter

8.1.3 Con Divisor de 8:

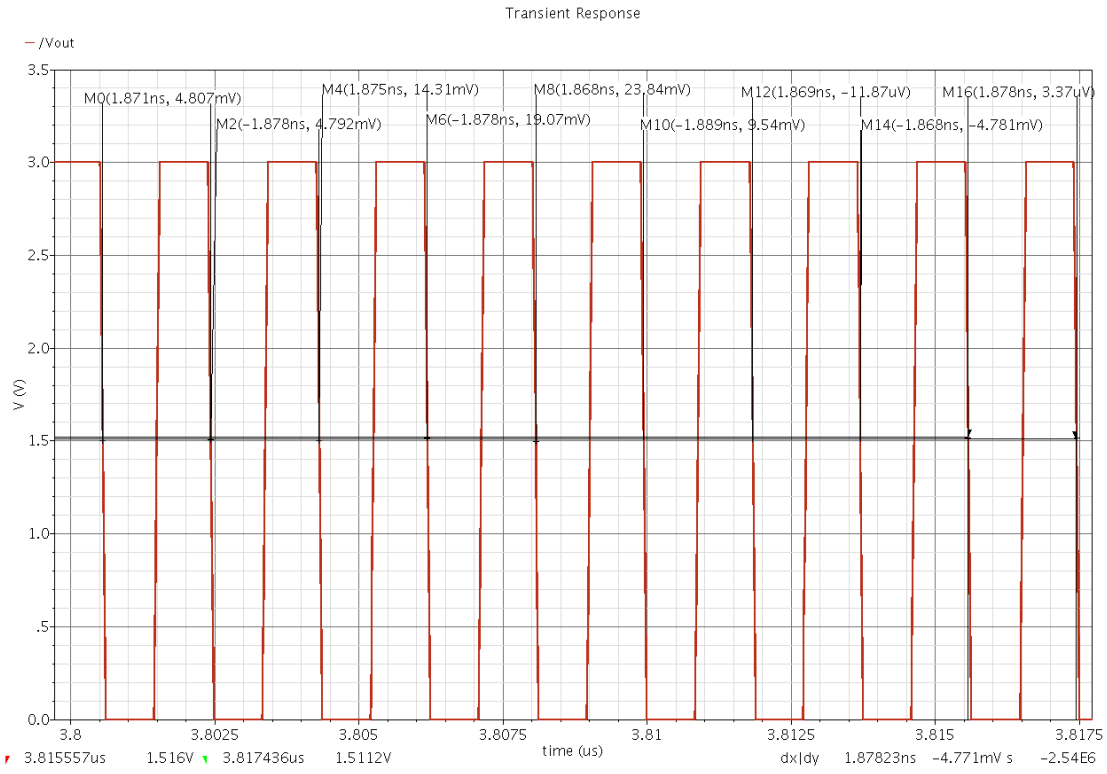


Figura 43: Medición del Cycle Jitter

8.1.4 Con Divisor de 12:

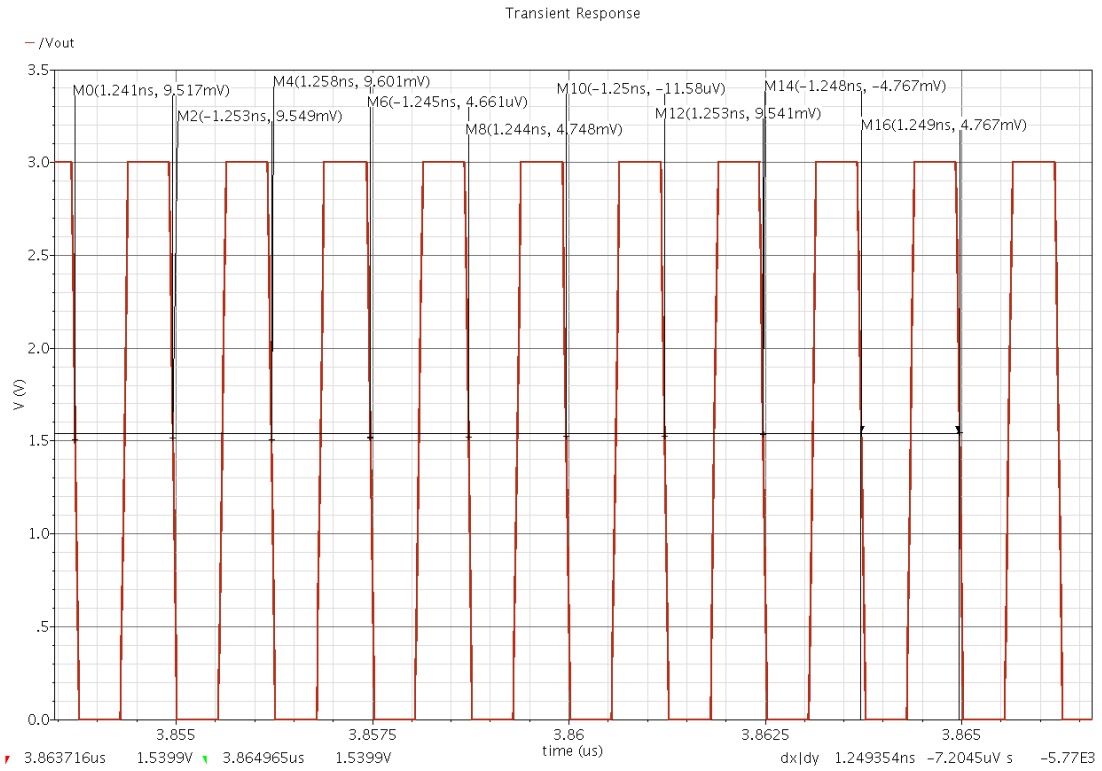


Figura 44: Medición del Cycle Jitter

8.1.5 Con Divisor de 16:

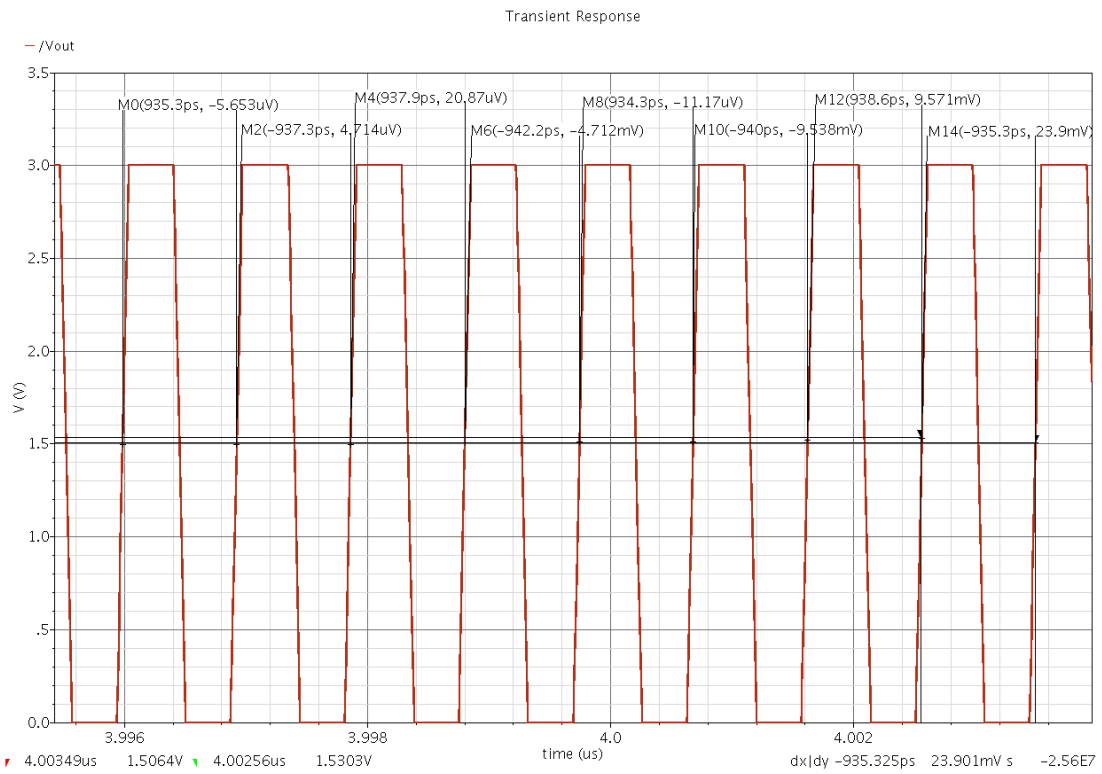


Figura 45: Medición del Cycle Jitter

8.1.6 Con Divisor de 24:

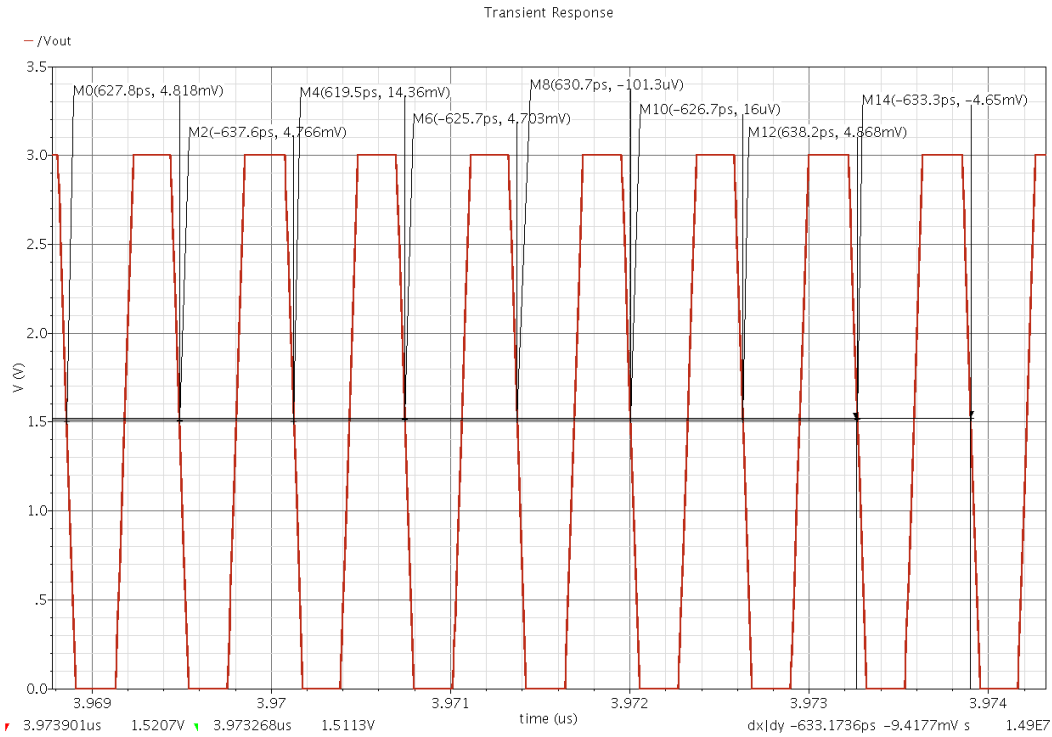


Figura 46: Medición del Cycle Jitter

9. MEDICIÓN DEL PHASE JITTER:

Para medir el jitter de fase, se realizó el diagrama de ojo, para las señales de salida del PLL, para varios valores de N. en las graficas siguientes se muestran los diagramas de ojo para cada señal, así como la medición de su jitter.

$$\text{edge-to-edge jitter} \\ J_{ee}(t) = \sqrt{\text{var}(\delta t_i)}$$

9.1.1 Con Divisor de 2:

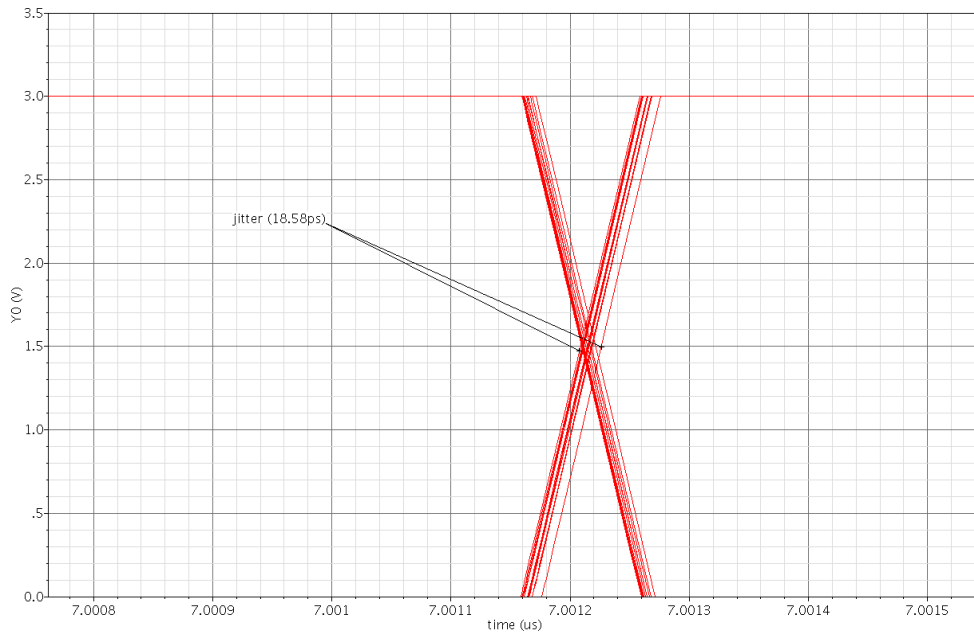


Figura 47: Medición del Phase Jitter= 18.5ps

9.1.2 Con Divisor de 4:

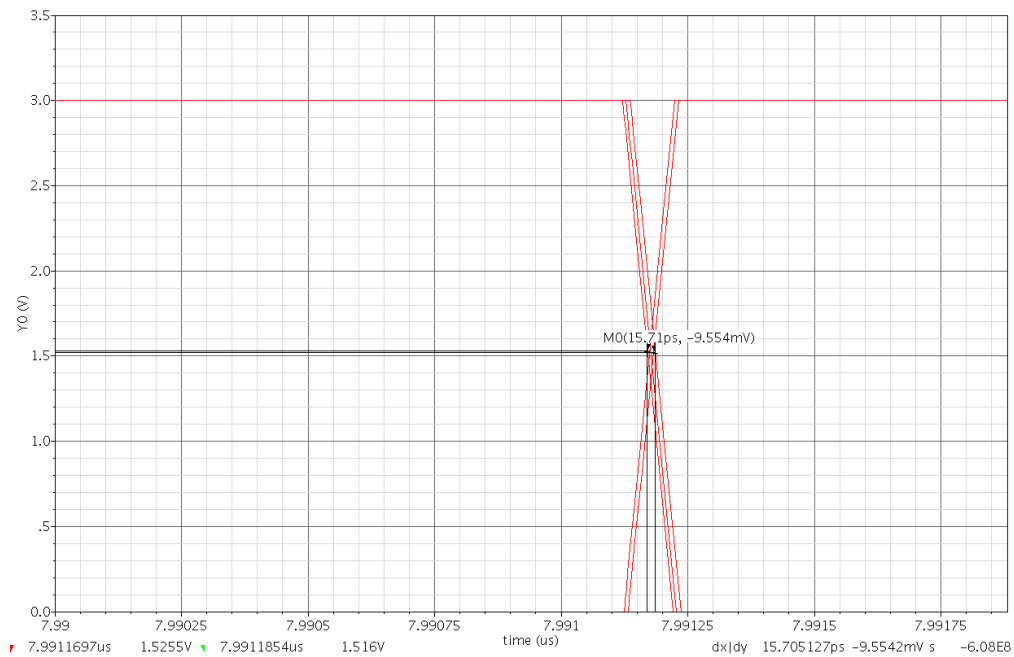


Figura 48 Medición del Phase Jitter=15.71ps

9.1.3 Con Divisor de 6:

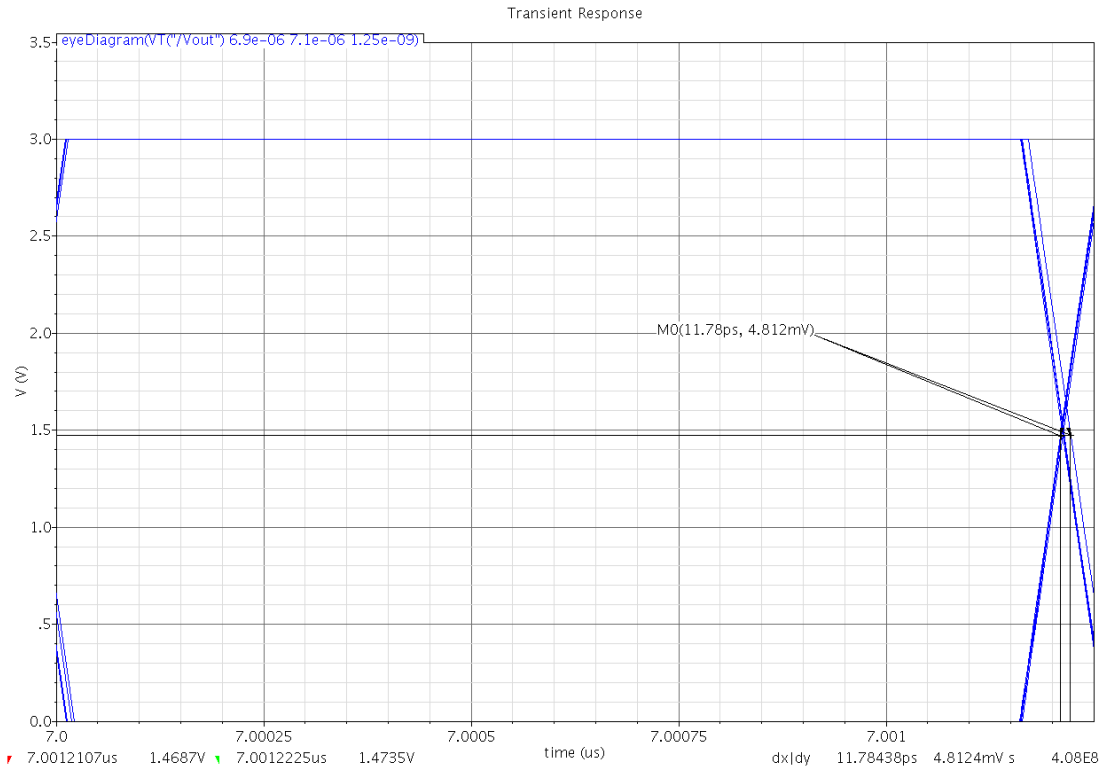


Figura 49 Medición del Phase Jitter=11.78ps

9.1.4 Con Divisor de 8:

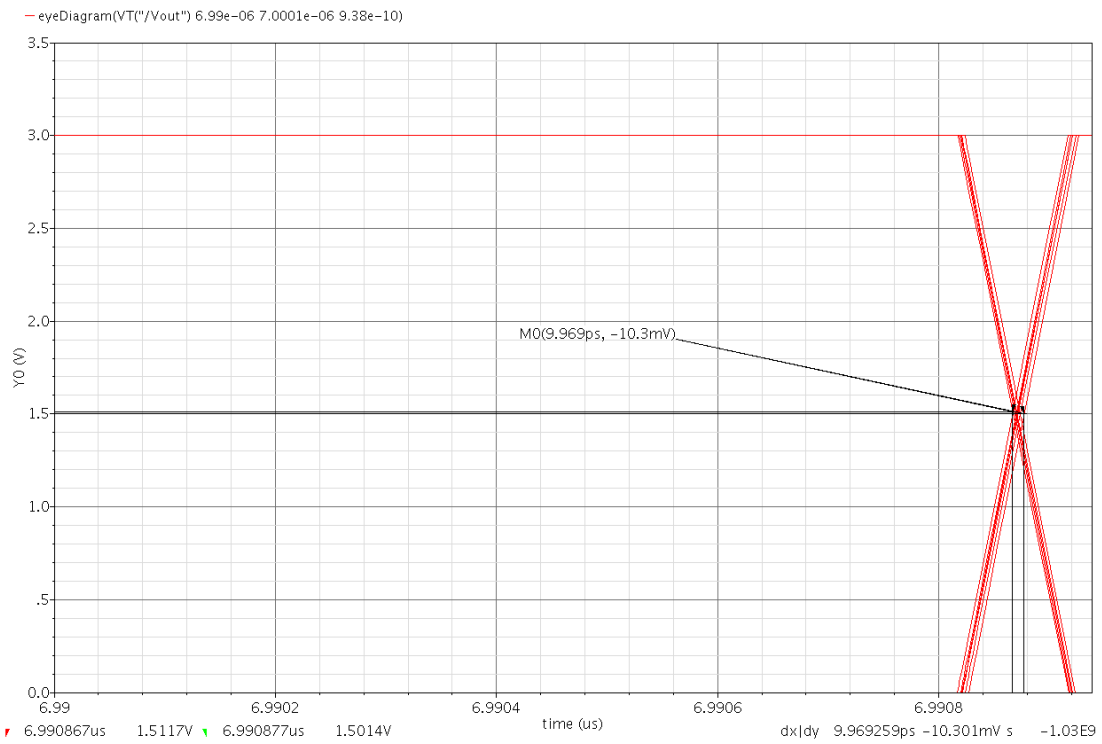


Figura 50 Medición del Phase Jitter=9.9ps

9.1.5 Con Divisor de 12:

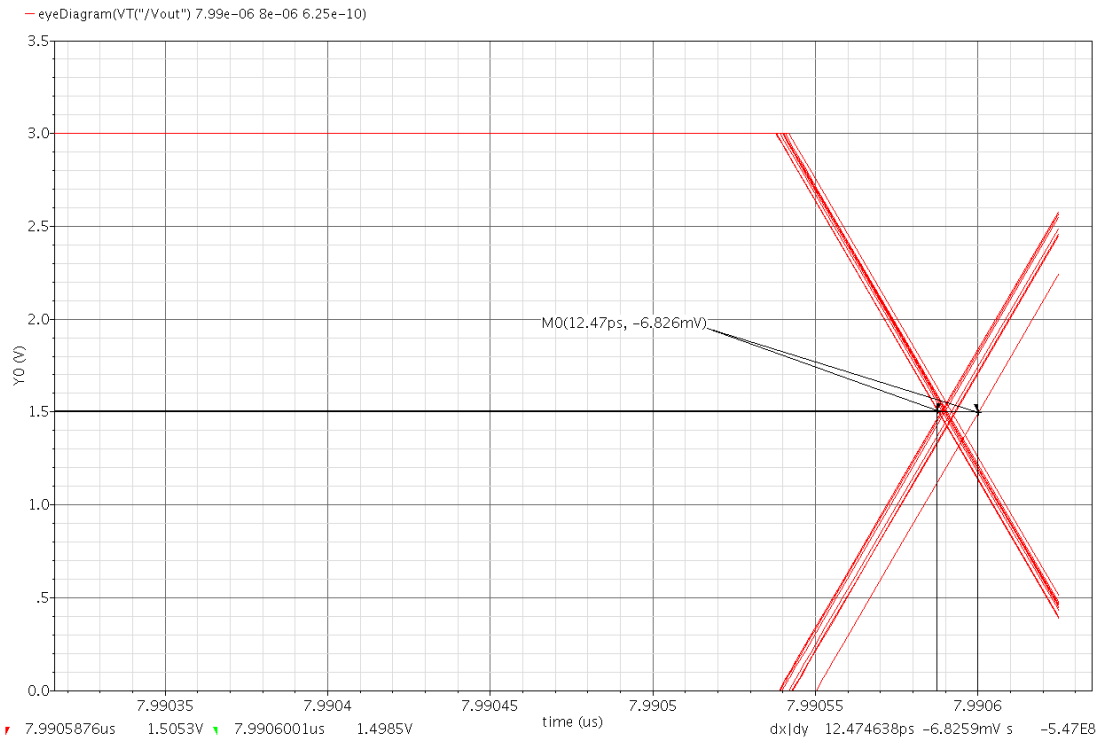
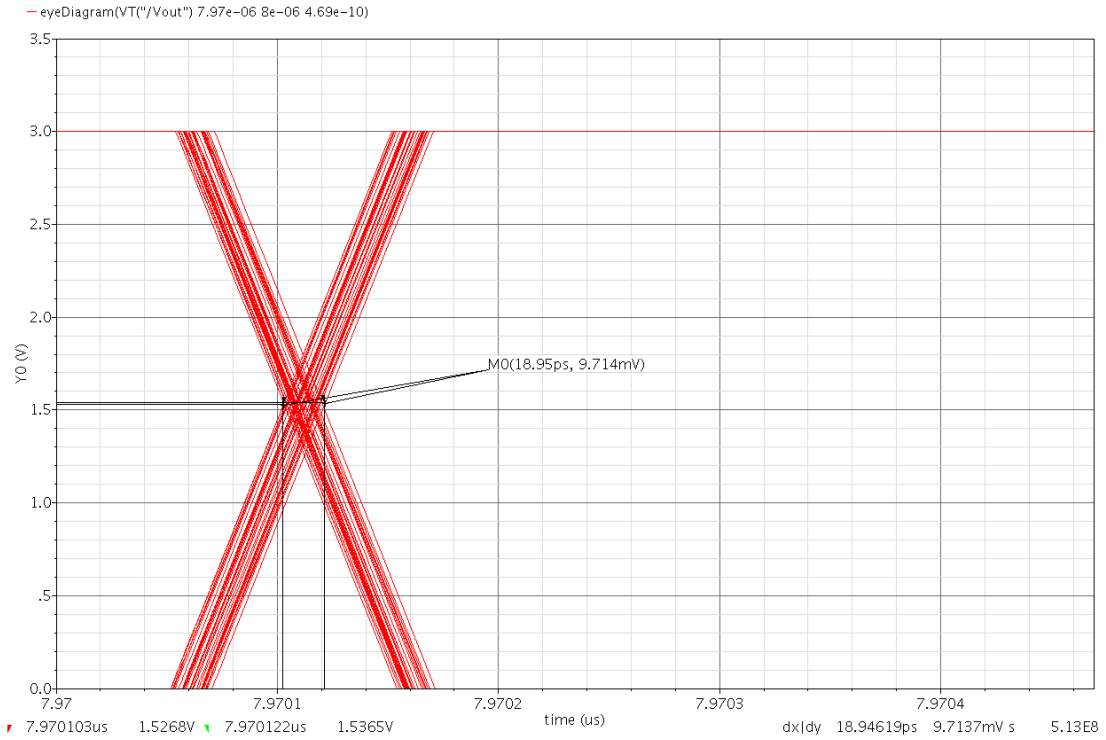


Figura 51 Medición del Phase Jitter=12.47ps

9.1.6 Con Divisor de 16:

**Figura 52 Medición del Phase Jitter=18.95ps**

9.1.7 Con Divisor de 24:

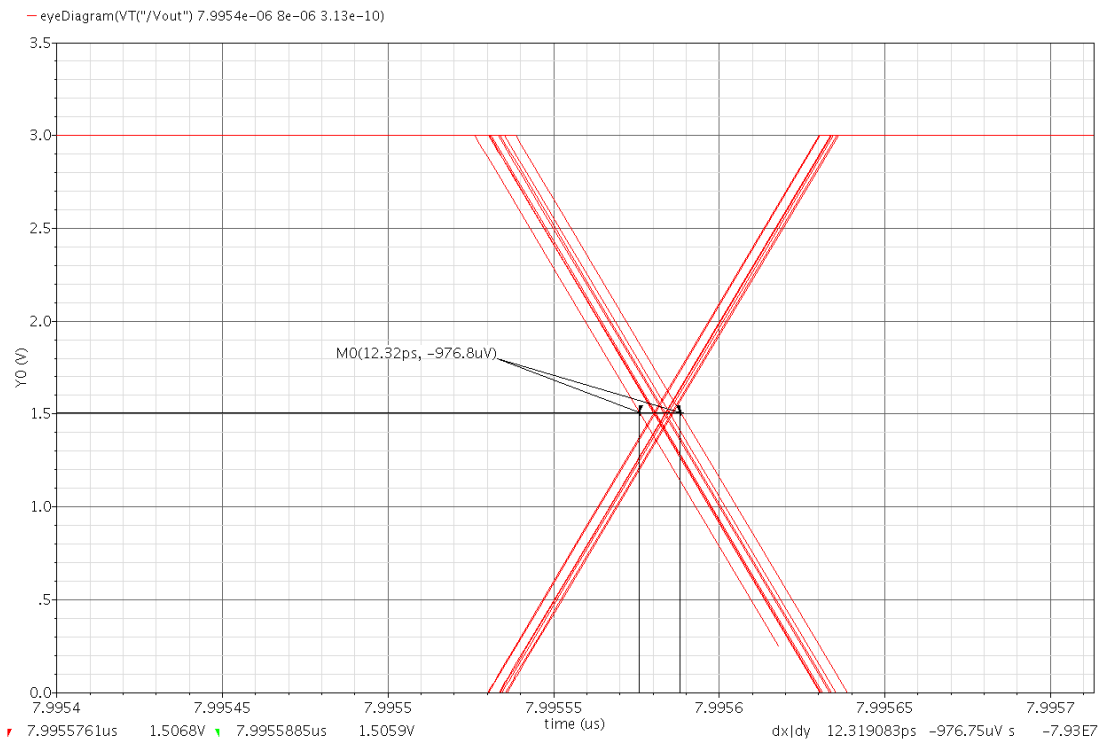


Figura 53 Medición del Phase Jitter=12.32ps

10. MEDICIÓN DEL STATIC PHASE OFFSET:

10.1.1 Con Divisor de 2:

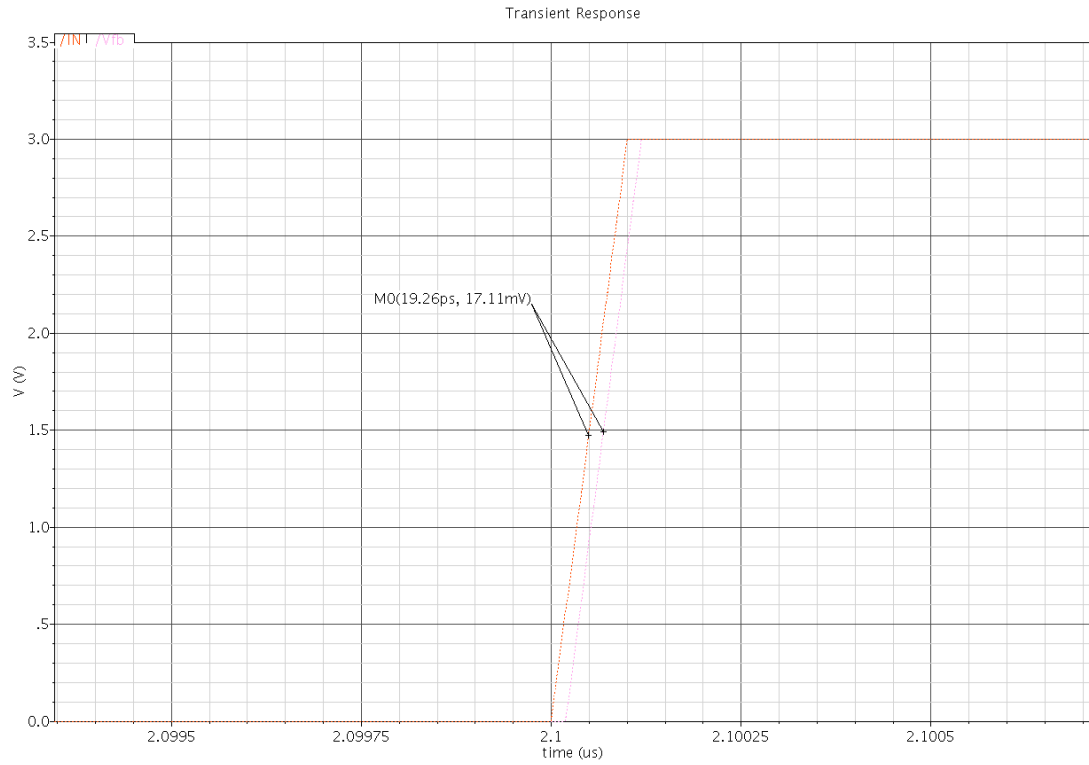


Figura 54: Medición del SPO=19.26ps

10.1.2 Con Divisor de 4:

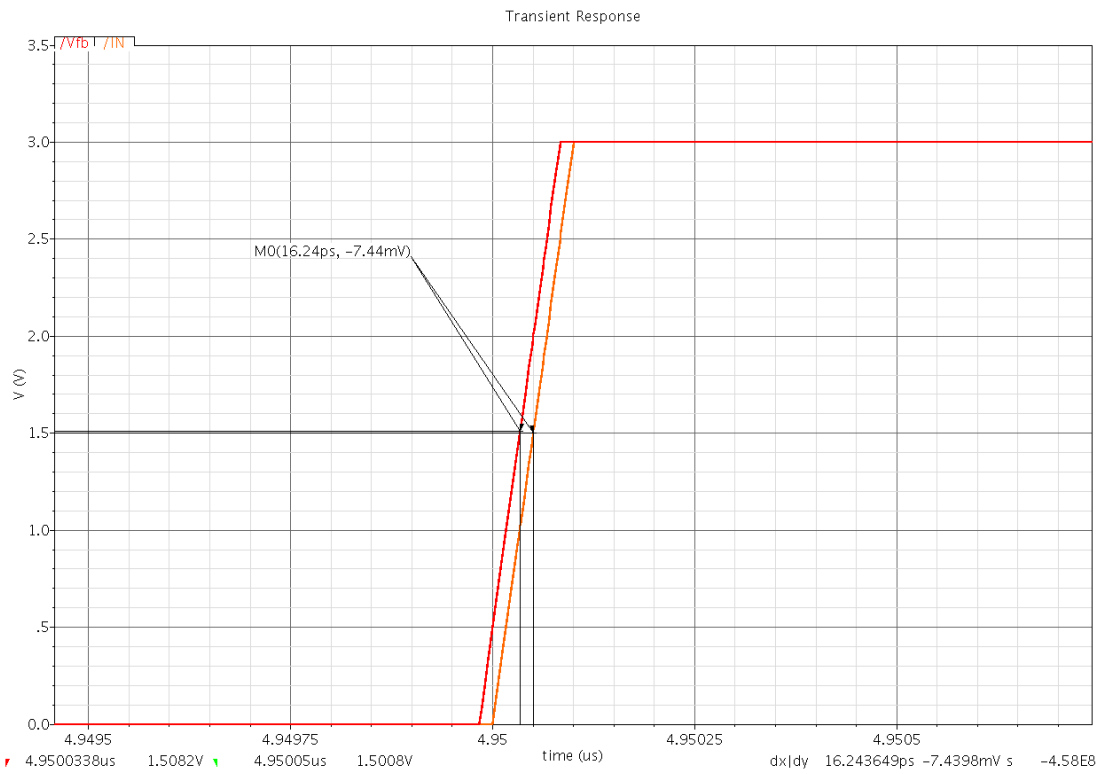


Figura 55: Medición del SPO=16.24ps

10.1.3 Con Divisor de 6:

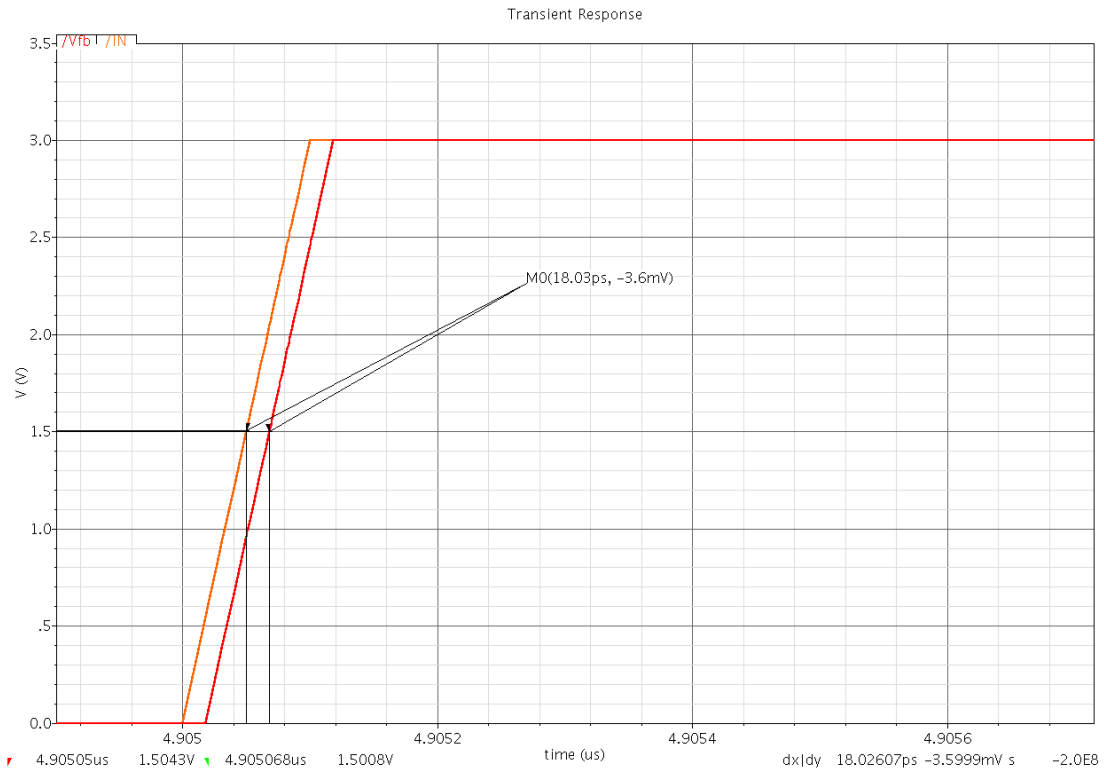


Figura 56: Medición del SPO=18ps

10.1.4 Con Divisor de 8:

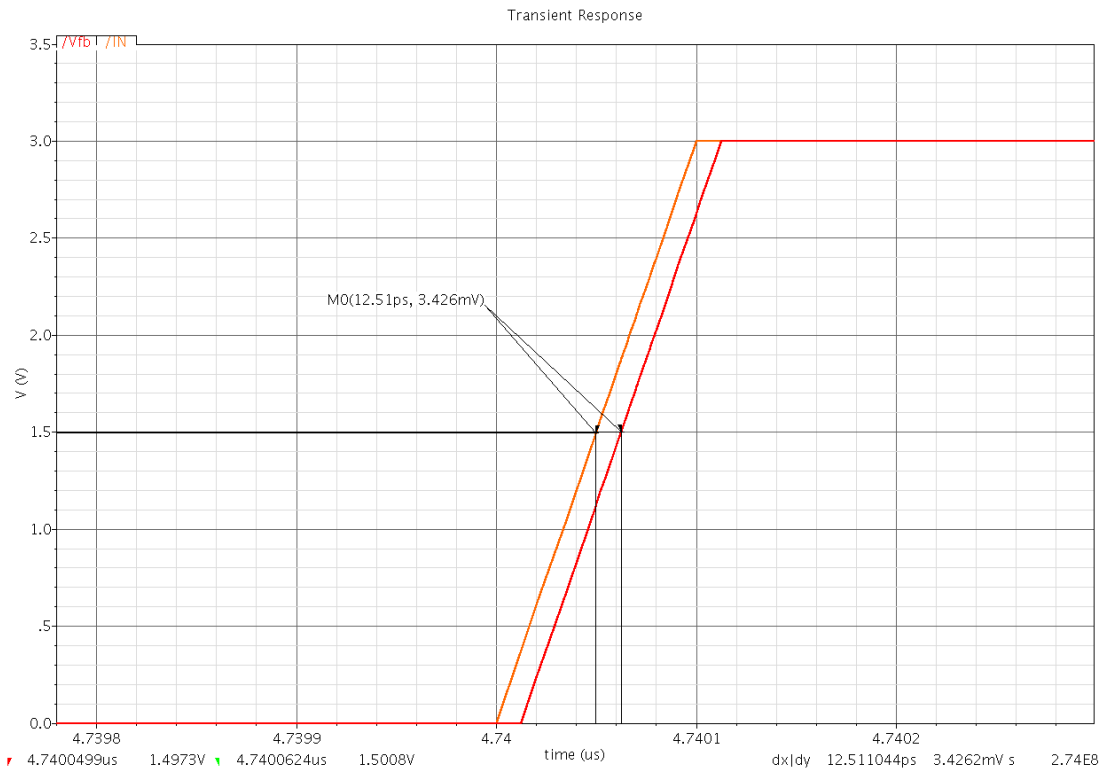


Figura 57: Medición del SPO=12.51ps

10.1.5 Con Divisor de 12:

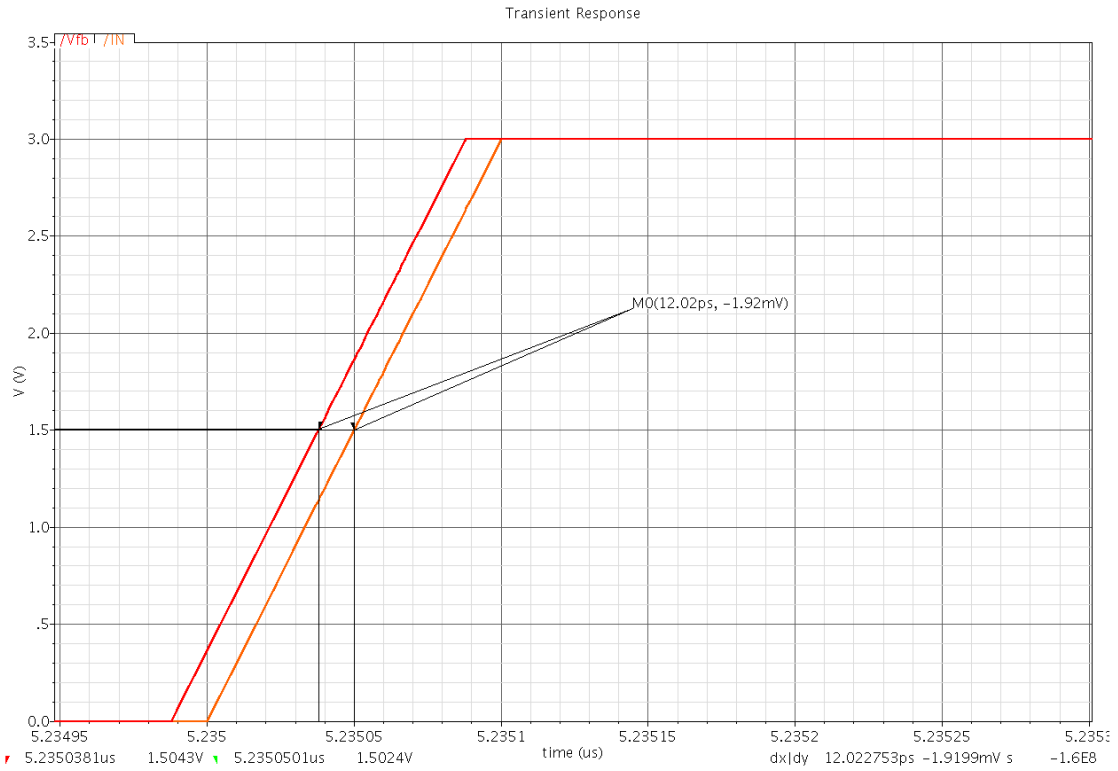


Figura 58: Medición del SPO=12ps

10.1.6 Con Divisor de 16:

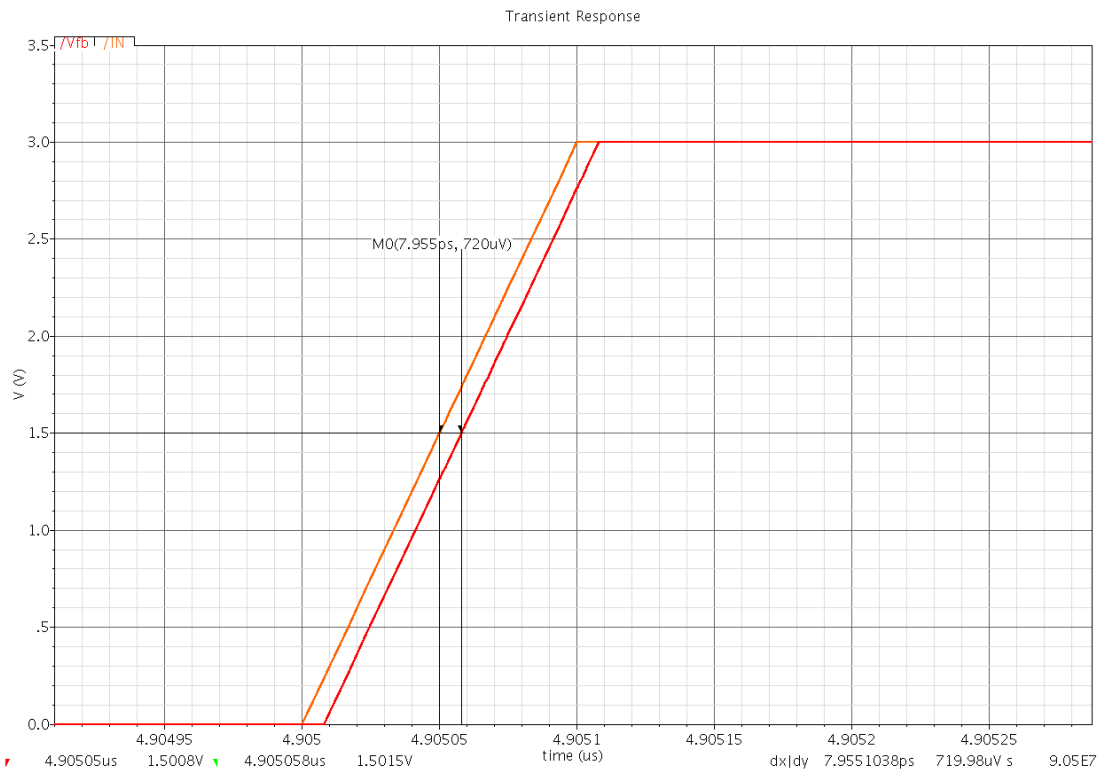


Figura 59: Medición del SPO=8ps

10.1.7 Con Divisor de 24:

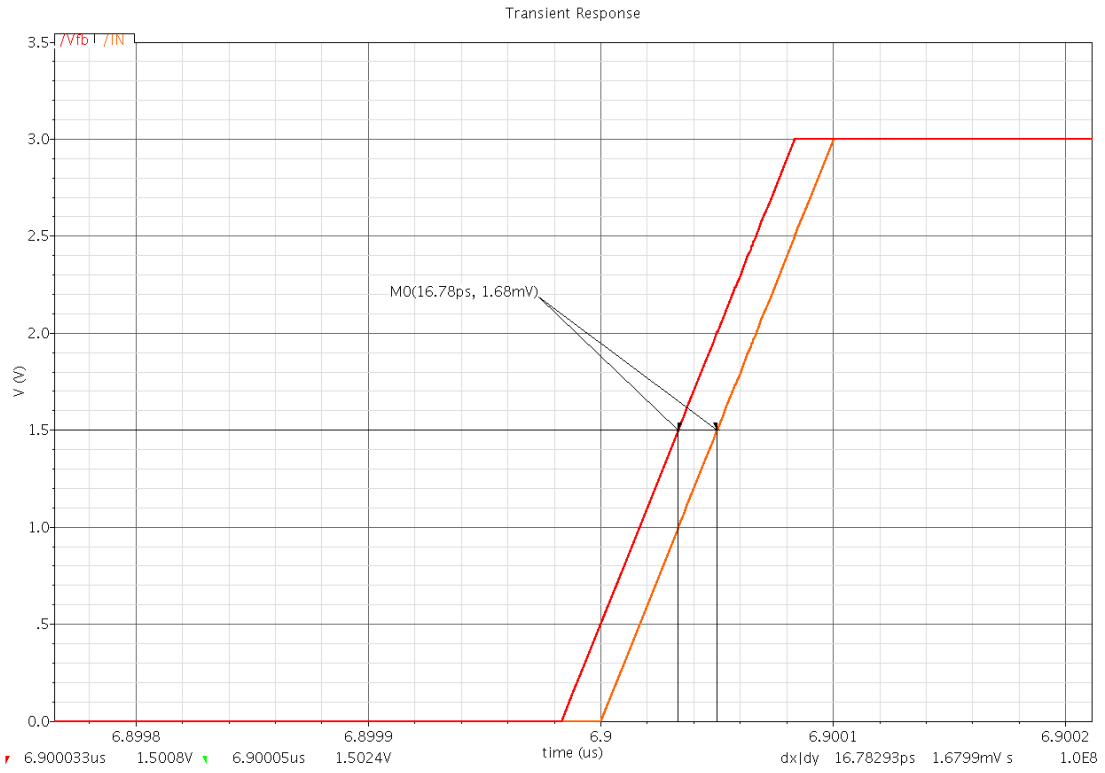


Figura 60: Medición del SPO=16.78ps

11. SPREAD SPECTRUM COMPATIBLE.

El PLL será compatible con spread spectrum si el ancho de banda del PLL esta una década arriba en frecuencia que la frecuencia de modulación del spread spectrum, por lo tanto si se recuerda que el ancho de banda del PLL es de 3.3MHz y la frecuencia de modulación del spread spectrum es de 30kHz, 31KHz, la relación se cumple y el PLL es compatible con spread spectrum. Para comprobarlo se realizo una simulación los resultados se muestran en las graficas siguietes:

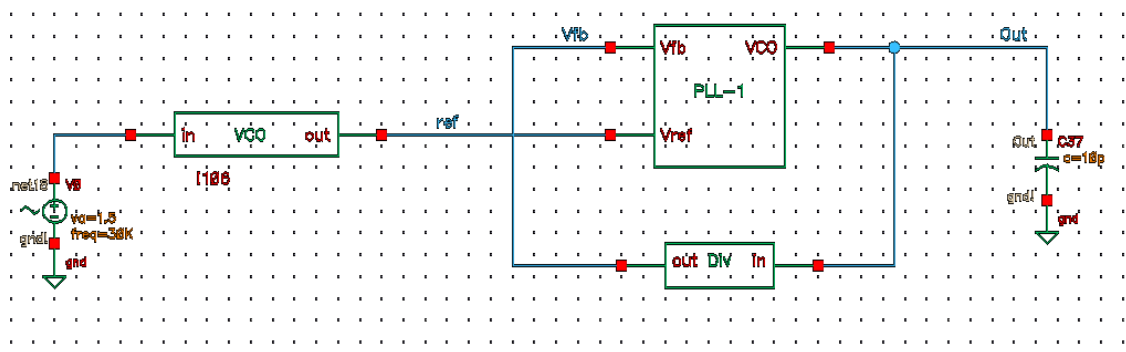


Figura 61: Diagrama esquemático del TestBench.

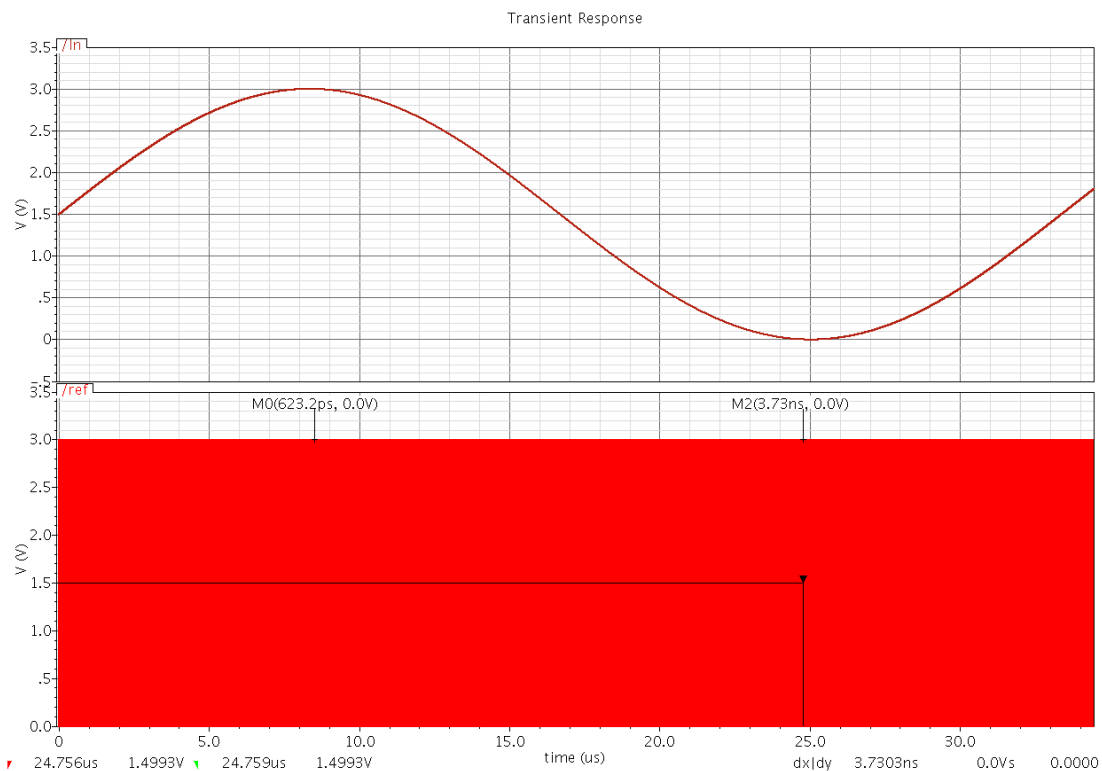


Figura 62: Grafica de la señal de entrada al modulador, y señal de referencia del PLL.

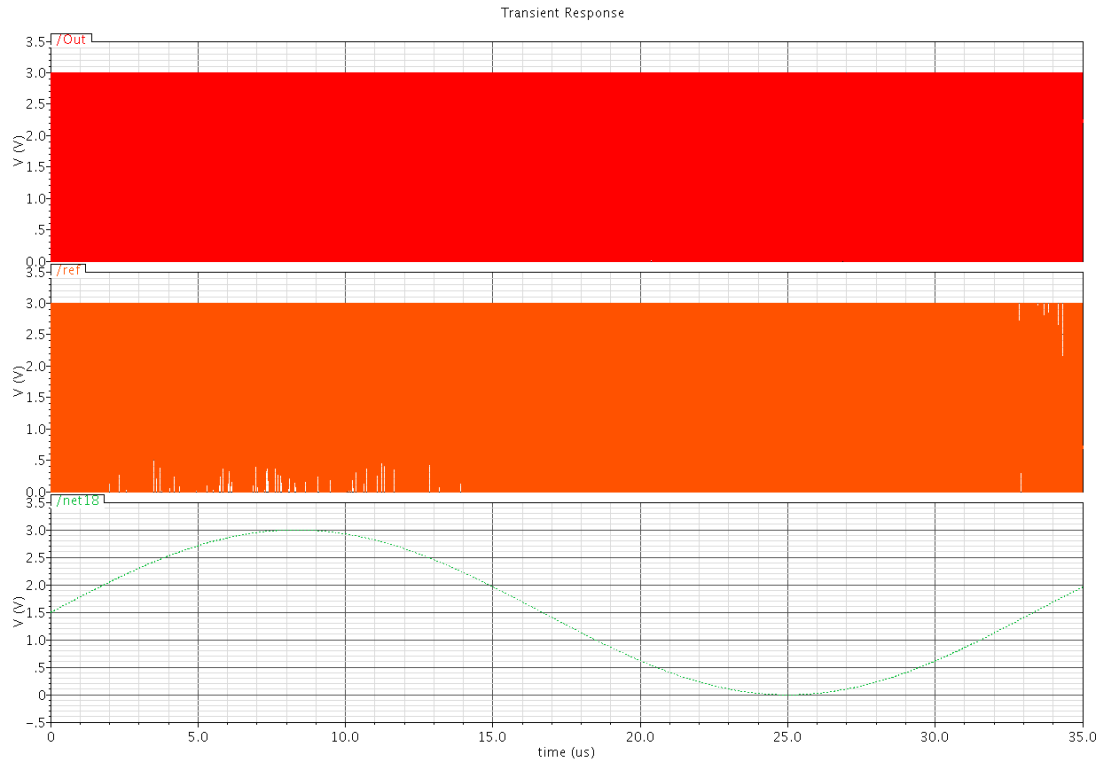


Figura 63: Grafica de la señal de referencia, señal de entrada y señal de salida del PLL.

12. CONCLUSIONES:

El diseño del PLL se dividió en dos partes, la primera fue el diseño del modelo en A.C. el cual sirvió para conocer la respuesta en frecuencia del PLL así como la estabilidad y el tiempo de amarre del PLL, este modelo es muy sencillo de realizar o de obtener, tanto el análisis matemático que se desarrolló con ayuda de MatLab, como la simulación que se realizó con un modelo en VerilogA.

La segunda parte de este proyecto, fue el de obtener el modelo transitorio del PLL, este modelo llevó mucho tiempo, debido a la falta de entendimiento sobre el funcionamiento de algunos de los bloques del PLL, las simulaciones de este modelo requirieron de más tiempo, sobre todo la simulación donde se quería conocer si el PLL era compatible con el Spread Spectrum, esta simulación llevó casi 8 horas, la verdad no se porque duró tanto tiempo. Pero las otras simulaciones fueron relativamente más rápidas, aunque comparadas con la simulación del modelo en AC, si hay mucho tiempo de diferencia, así también como el número de simulaciones que se tuvieron que realizar, en el modelo de AC solo se realizaron alrededor de 3, y en el modelo transitorio fueron bastantes.

En resumen se puede decir que el diseño del PLL fue algo no muy sencillo, es muy importante antes de iniciar el diseño entender cada uno de los bloques que componen un PLL, así como las unidades y dominios de cada uno de ellos, sino es así nunca se llegará a diseñar un PLL que cumpla con las especificaciones de diseño.

13. REFERENCIAS:

Optimal Loop Parameter Design of Charge Pump PLLs For Jitter Transfer Characteristic Optimization

Hanjun Jiang, Chengming He, Degang Chen and Randall Geiger

Department of Electrical and Computer Engineering

Iowa State University Ames, IA, 50011, USA

Behavioral Modeling and Simulation of Jitter and Phase Noise in Fractional-N PLL Frequency Synthesizer

Xiaojian Mao Huazhong Yang Hui Wang

Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers Ken Kundert Designer's Guide Consulting, Inc.

Analog behavioral modeling with the Verilog-A language

Escrito por Dan FitzPatrick, Ira Miller

