



UNIVERSIDAD INDUSTRIAL DE SANTANDER
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Perfecta combinación entre Energía e Intelecto



***APLICACIÓN DE LA PROGRAMACIÓN GEOMÉTRICA EN EL DISEÑO
DE UN SINTETIZADOR PARA RADIOFRECUENCIA, INTEGRADO EN
TECNOLOGÍA CMOS.***

Genner Andrés Carrillo Rueda

*Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones
Bucaramanga, Enero del 2009*

*APLICACIÓN DE LA PROGRAMACIÓN GEOMÉTRICA EN EL DISEÑO
DE UN SINTETIZADOR PARA RADIOFRECUENCIA, INTEGRADO EN
TECNOLOGÍA CMOS.*

AUTOR

Genner Andrés Carrillo Rueda

DIRECTOR

MSc. Elkim Felipe Roa Fuentes

CO-DIRECTOR

Ing. Fabian Leonardo Cabrera Riaño

*Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones
Bucaramanga, Enero del 2009*

Contenido

1. INTRODUCCIÓN	11
1.1. Aspectos Generales	12
1.1.1. Sintetizador de Frecuencia	12
1.1.2. Programación Geométrica	13
1.2. Estado del Arte	15
1.3. Metodología de diseño	16
1.4. Organización del Documento	17
2. DIVISOR DE FRECUENCIA	19
2.1. Consideraciones iniciales	19
2.1.1. Selección de Arquitectura	20
2.1.2. Especificaciones de Diseño	22
2.2. Aplicación de la Programación Geométrica	22
2.2.1. Formulación del Problema	23
2.2.2. Implementación en un Programa Geométrico	30
2.3. Verificación de Resultados	34
2.3.1. <i>Flip-flop</i> con Lógica <i>TSPC</i>	34
2.3.2. <i>Flip-flop</i> con Lógica <i>E-TSPC</i>	35
2.3.3. Análisis de Resultados	37
3. RESULTADOS	39
3.1. PG como Metodología de Diseño para Divisores de Frecuencia	39
3.2. Automatización de la Metodología	42
3.3. Validación del Diseño del PLL	42
3.4. Layout del PLL	44
3.5. Observaciones y Conclusiones	45
3.6. Recomendaciones para Trabajos Futuros	46

Agradecimientos

Agradezco a Dios, por haberme permitido desarrollar este trabajo. A mi novia, por ser mi motivación y sustento en los momentos mas difíciles. A mi familia, por haberme dado todo para mi formación y para mi vida.

Agradezco a la Universidad Industrial de Santander, por la formación académica y humana recibida durante el desarrollo de mi carrera. A la Vicerrectoria de Investigaciones, por el apoyo económico recibido mediante el programa de “Proyectos Promisorios”.

También agradezco muy sinceramente a todos los integrantes del grupo CIDIC, por toda la colaboración y por demostrarme el verdadero trabajo en equipo. A mi director, el profesor Elkim por el todo apoyo y las enseñanzas que me ayudaron para formarme como profesional y como persona. A mi co-director Fabian Cabrera por las orientaciones en los momentos mas importantes.

Finalmente, muchas gracias a todos aquellos que me colaboraron de alguna u otra forma para desarrollar este proyecto.

RESUMEN

TÍTULO: APLICACIÓN DE LA PROGRAMACIÓN GEOMÉTRICA EN EL DISEÑO DE UN SINTETIZADOR PARA RADIOFRECUENCIA, INTEGRADO EN TECNOLOGÍA CMOS.¹

AUTOR: GENNER ANDRÉS CARRILLO RUEDA²

PALABRAS CLAVES: Sintetizador de frecuencia, *PLL*, *CMOS*, radiofrecuencia, circuito integrado, programación geométrica, optimización convexa, *PG*, *flip-flop*, demora, frecuencia, *TSPC*, *E-TSPC* .

DESCRIPCIÓN

La aplicación de la programación geométrica como metodología de diseño, ha dado excelentes resultados en el campo de los circuitos integrados analógicos y digitales. En el presente trabajo se aplicó, como método de optimización para algunos bloques críticos de un sintetizador de frecuencia.

Entre los bloques optimizados se encuentran *flip-flops* tipo D, con lógica *TSPC* (*True Single Phase Clock*) y con lógica *E-TSPC* (*Extended-True Single Phase Clock*). Además se optimizó un inversor, cuya función objetivo es el tiempo de demora para procesar la señal. Entre las principales restricciones utilizadas en este trabajo para los programas geométricos se encuentran el área y el consumo de potencia.

La metodología utilizada permitió el diseño rápido y sencillo de un *flip-flop* tipo D, con lógica *E-TSPC* adaptado para operar a 5.9 GHz. Adicionalmente, se propuso una alternativa de circuito para resolver algunas de las desventajas de la programación geométrica.

Finalmente se rediseñó un *PLL* ya existente en la literatura, con el objetivo de analizar las ventajas de la optimización en todo el sistema. El *layout* del *PLL* cumple con las actuales reglas de fabricación en la tecnología de 0.35μm C35B4 CMOS para el proceso de *Austria Micro Systems* (*AMS*)

¹Modalidad: Trabajo de grado.

²Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.
Director: MSc. Elkim Felipe Roa Fuentes. Co-Director: Ing. Fabian Leonardo Cabrera Riaño

SUMMARY

TITLE: FREQUENCY SYNTHESIZER DESIGN FOR A TRANSCEIVER INTEGRATED ON CMOS TECHNOLOGY.³

AUTHOR: GENNER ANDRÉS CARRILLO RUEDA⁴

KEY WORDS: Frequency synthesizer, *PLL*, *CMOS*, radio frequency, integrated circuit, geometric programming, convex optimization, *GP*, *flip-flop*, delay, frequency, *TSPC*, *E-TSPC*.

DESCRIPTION

Applying geometric programming as a design methodology for analog and digital circuits has been given excellent results. In this work is applied to the most critical blocks from a frequency synthesizer as a optimization method.

Type D flip-flops with TSPC logic (True Single Phase Clock) and with E-TSPC (Extended-True Single Phase Clock) are some of the blocks that has been optimized. Moreover a digital inverter were optimized, with the delay as objective function. Among the principal restrictions for this geometric program were the area and the power consuming.

The applied methodology allow the quick and simple design of a type D flip-flop with E-TSPC, arranged to operate at 5.9 *GHz*. Furthermore, an circuit technique alternative was implemented in order to solve some disadvantages of the geometric programming.

Finally an already existent *PLL* was redesigned in order to apply and analyze the advantages from the optimization in the complete system. The layout of the PLL accomplish with actual rules for its fabrication in the 0.35 μm C35B4 CMOS process of *Austria Micro Systems (AMS)*.

³Degree Project.

⁴Faculty of Physics-Mechanical Engineering. Electric, Electronic and Telecommunications School. Director: MSc. Elkim Felipe Roa Fuentes. Co-Director: Ing. Fabian Leonardo Cabrera Riaño

Capítulo 1

INTRODUCCIÓN

Durante los últimos años, los equipos utilizados en la implementación de las comunicaciones inalámbricas han presentado un incremento notable en su frecuencia de transmisión-recepción, lo cual impide el uso de cristales piezoeléctricos como fuente directa de frecuencia para los procesos de modulación y demodulación. Debido a este aumento en la frecuencia, las variaciones en la fase de la señal generada por el oscilador local se ven incrementadas, ya sea debido a defectos del proceso o del diseño. Por tal razón, la operación del sistema resulta inadecuada, requiriéndose un método que permita controlar la precisión de la señal generada, medida en partes por millón (*ppm*), de forma tal que el sistema cumpla con el desempeño exigido por los estándares que regulan este campo de las comunicaciones.

En particular, los sistemas de transmisión y recepción (*transceiver*) implementan un método de lazo cerrado conocido como lazo de control de fase *PLLs*¹, el cual tiene como núcleo de operación multiplicadores y divisores de frecuencia tal y como se aprecia en la figura 1.2. El desempeño del *PLL* se ve notablemente afectado por las limitaciones de estos bloques, lo que demanda un diseño robusto y apropiado. En particular, el divisor de frecuencia encargado de sensar la señal de salida del oscilador local y adecuarla para la comparación con la señal de referencia, requiere de gran precisión y velocidad, ya que es el principal responsable en la sintonización de los diversos canales de comunicación.

Este proceso de diseño, efectuado en el método tradicional, donde se selecciona una topología, se realiza una estimación de las variables de diseño y se simula de forma iterada hasta cumplir las especificaciones, implica gran cantidad de tiempo y requiere de un conocimiento profundo de la topología a diseñar. Debido al alto número de variables involucradas, las cuales se asocian mediante diversas restricciones que establecen el comportamiento del circuito, el diseño de este bloque se transforma en un problema n-dimensional.

Por otra parte, las relaciones entre las variables de un circuito se pueden expresar por medio de ecuaciones que determinan de forma aproximada el comportamiento físico del mismo. Mediante dichas ecuaciones es posible expresar el dimensionamiento de circuitos integrados como un problema de optimización matemática, de manera que al implementar un algoritmo se pueda obtener la solución óptima. La programación geométrica es un tipo especial de optimización convexa que ha dado excelentes resultados en el proceso de diseño de circuitos integrados analógicos y digitales. La principal ventaja de utilizar la programación geométrica radica en la existencia de métodos que resuelven un problema geométrico de manera exacta, global y eficiente.

El uso de la programación geométrica en el dimensionamiento de circuitos digitales no es algo nuevo, por ejemplo en 1985 Fishburn y Dunlop [1] propusieron un método para el dimensionamiento de los transistores y las interconexiones basado en la técnica *Elmore Delay*, la cual, tiempo después se identificó como un programa

¹PLL son las iniciales de *Phase Locked Loop*

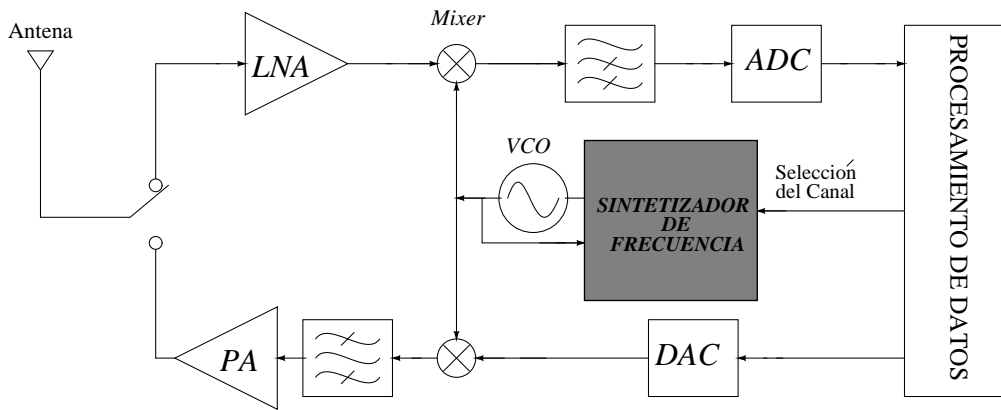


Figura 1.1: *Transceiver* de arquitectura genérica.

geométrico. En los últimos años, la aparición de nuevos algoritmos de solución basados en el método de puntos interiores permiten una alta velocidad de convergencia, incluso si se tienen miles de restricciones [2].

Una muestra de la aplicación de la programación geométrica en el diseño de circuitos analógicos es presentada por Hershenson [3], quien utiliza este método de optimización en el diseño de amplificadores operacionales. Así mismo, a nivel local Cabrera-Oliveros [4] crearon una herramienta para el diseño de tres configuraciones de amplificadores operacionales, incluyendo un paquete de funciones para el modelado de los transistores y una estrategia para la adaptación de la programación geométrica en el diseño de los mismos.

En el presente proyecto se desea utilizar programación geométrica por las ventajas y las potencialidades que posee, para mejorar el desempeño de un sintetizador de frecuencia. Para tal fin se propone optimizar el divisor de frecuencia, como pieza clave en lo referente a las especificaciones de frecuencia de operación, consumo de potencia y área ocupada por el circuito. Esto implica que se optimicen los bloques que conforman los divisores, en particular los basados en *TSPC* y *E-TSPC*. De esta manera se lleva el diseño, a profundizar en dichas celdas extrayendo las mejores especificaciones de cada una y así mejorar el desempeño de todo el sintetizador.

Esto quiere decir que el diseño del *PLL* se va enfocar únicamente en la optimización de las celdas que conforman el divisor de frecuencia, buscando que las características mejoradas del bloque sean transferidas finalmente a todo el sistema. Es importante destacar las bondades de este enfoque, pues adicional a optimizar el *PLL* se tiene la posibilidad de aprovechar las mejoras en otros sistemas reconociendo al *flip-flop* como bloque fundamental de muchos sistemas digitales.

1.1. Aspectos Generales

1.1.1. Sintetizador de Frecuencia

El *transceiver*,² es un sistema que se encarga de la transmisión y recepción de las señales en dispositivos portátiles de comunicaciones basados en radio frecuencia (*RF*). En el *transceiver* genérico presentado en la figura 1.1, la señal captada por la antena atraviesa el filtro multiplexor y luego es acondicionada por el amplificador de bajo ruido (*LNA*), que se encarga de amplificar las señales procedentes de la antena sin

²Se utilizará este término en inglés debido a que la traducción utilizada por algunos autores, la cual corresponde a transceptor, no ha sido plenamente aceptada por la academia internacional y no encierra la esencia del dispositivo al que se hace referencia.

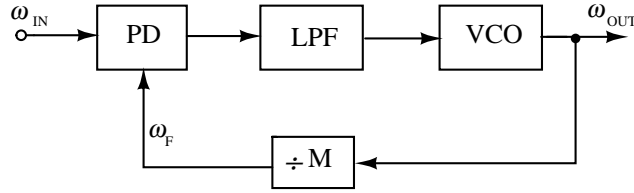


Figura 1.2: *PLL* con un divisor de frecuencia en el lazo de realimentación.

degradar su relación señal a ruido. Posteriormente, la señal *RF* amplificada se combina mediante el mezclador con la señal del oscilador local (*LO*), para ser filtrada y obtener la señal en banda base, la cual se convierte al dominio digital para ser procesada. En el transmisor, la señal proveniente del procesador se convierte en analógica y luego se mezcla con la portadora sintetizada localmente. Como resultado se tiene una señal a la frecuencia de transmisión, que finalmente es llevada a la antena a través del amplificador de potencia.

La exactitud de la frecuencia a la salida del oscilador local en un sistema de comunicaciones, es uno de los factores más importantes para la transmisión y recepción de información, a tal punto, que dicha frecuencia debe tener una exactitud equivalente a unas pocas partes por millón (*ppm*). Para cumplir con esta especificación, es útil implementar un lazo de control de fase o *PLL*,³ el cual en su forma más simple se encarga de igualar la fase entre dos señales de entrada. No obstante, al agregar un bloque extra en el lazo de realimentación del *PLL*, es posible efectuar síntesis de frecuencia.

El diagrama de bloques correspondiente a un sintetizador de frecuencia se muestra en la figura 1.2, éste se constituye por un detector de fase (*PD*), un filtro pasa bajas (*LPF*), un oscilador controlado por voltaje (*VCO*) y un divisor de frecuencia en el lazo de realimentación ($\div M$). El sintetizador utiliza estos bloques para controlar la fase, de tal manera, que toma la frecuencia de una fuente externa y la compara con la frecuencia de salida del *VCO*, logrando así obtener una frecuencia con la exactitud deseada a la salida del bloque. En aplicaciones *RF* generalmente se utiliza un oscilador de cristal como fuente de referencia externa, debido a que brinda tonos de frecuencia de alta pureza. Aunque podría pensarse en utilizar el cristal como reemplazo del sintetizador, la capacidad de integración y la frecuencia de operación (~ 200 MHz), lo hacen inadecuado para implementarse en dispositivos inalámbricos.

El desempeño de un sintetizador de frecuencia es medido con base en ciertas especificaciones que van desde características generales para cualquier circuito, como el consumo de potencia, hasta algunas muy particulares como la pureza de la señal, el tiempo de establecimiento, el ruido de fase, entre otras [5]. Es claro que algunos bloques del *PLL* tienen mayor influencia sobre ciertas especificaciones, como por ejemplo la operación del divisor de frecuencia que afecta la frecuencia a la cual pueda operar el *PLL*.

1.1.2. Programación Geométrica

La programación geométrica (PG) es una clase de optimización no lineal con muchas propiedades que la hacen buena opción para la implementación computacional. Un programa geométrico requiere de una formulación especial que consta de una función objetivo (la que se desea optimizar) y de unas restricciones que permiten limitar el espacio de optimización. Tanto la función objetivo como las restricciones, pertenecen a un conjunto de funciones que tienen una forma especial, y por tanto cumplen con algunas condiciones específicas.

Una de las principales cualidades de la programación geométrica, es que puede ser convertida mediante

³Por sus siglas en inglés *Phase Locked-Loop*.

transformaciones de las funciones involucradas y cambios de variables en un problema convexo y por lo tanto adquirir las ventajas de la optimización convexa. Entre estas ventajas se puede mencionar que permite encontrar el óptimo global de la función objetivo (siempre y cuando exista solución), con una alta velocidad de convergencia, por medio de algoritmos de puntos interiores [2].

Existen dos formas equivalentes de la programación geométrica: forma estándar y forma convexa. La primera es una optimización condicionada de un tipo de función llamada posinomio, y la segunda forma es obtenida a partir de la primera a través de un cambio de variable logarítmico. La forma estándar es comúnmente usada en problemas de posicionamiento de redes, mientras que la forma convexa se utiliza con modelos estocásticos de problemas teóricos.

A continuación se presentan algunas formas matemáticas utilizadas en la formulación de un programa geométrico y una definición un poco más formal de la programación geométrica junto con algunas observaciones para aplicar a la misma.

Monomios y Posinomios

Una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ con $\text{dom } f = \mathbb{R}_{++}^n$ definida como

$$f(x) = c x_1^{a_1} x_2^{a_2} x_3^{a_3} \cdots x_n^{a_n} \quad (1.1)$$

donde $c > 0$ y $a_i \in \mathbb{R}$, es una función del tipo monomio. Cabe destacar que cualquier variable o constante es un monomio, que mantiene su forma bajo la multiplicación o división.

Una suma de monomios, por ejemplo, una función de la forma

$$f(x) = \sum_{k=1}^K c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}} \quad (1.2)$$

donde $c_k > 0$, es llamada una función posinomial (con K términos), o simplemente un posinomio. Un posinomio mantiene su forma bajo la suma, multiplicación y escalamientos por constantes no negativas.

Formulación del programa geométrico

Un problema de optimización de la forma

$$\begin{aligned} \text{minimizar} \quad & f_0(x) \\ \text{sujeto a} \quad & f_i(x) \leq 1 \quad i = 1, \dots, m \\ & h_i(x) = 1 \quad i = 1, \dots, p \end{aligned} \quad (1.3)$$

donde f_0, \dots, f_m son posinomios y h_1, \dots, h_p son monomios, es denominado un programa geométrico, donde $f_0(x)$ es la función objetivo, $f_i(x)$ restricciones de tipo desigualdad y $h_i(x)$ restricciones de tipo igualdad.

Aunque la teoría de la programación geométrica está muy bien desarrollada y existen algoritmos muy eficientes, los investigadores interesados en usar programación geométrica aún necesitan adquirir una capacidad no trivial de modelar y aproximar problemas de la ingeniería a su forma geométrica. En conclusión, para aplicar programación geométrica en el diseño de circuitos analógicos es necesario:

- Conocer las variables del circuito a optimizar, es decir, tener en cuenta los requerimientos de diseño, para determinar cuáles tienen primacía.
- Representar el funcionamiento del circuito por medio de ecuaciones que relacionen los parámetros de desempeño con las variables de diseño.

Tabla 1.1: Trabajos representativos en sintetizadores de frecuencia.

Trab. ref.	Frecuencia [GHz]	Ref. [MHz]	Consumo[mW]		Área [mm ²]	VDD [V]	Ruido@1MHz [dBc/Hz]	Tecnología <i>CMOS</i>	Año
			con <i>VCO</i>	sin <i>VCO</i>					
[6]	2.4-2.48	32		9.04	0.027 ^a	3.3	-117.6	0.35 μm	2006
[7]	2.4	1	20	7	0.2925 ^b	2.5	-112	0.25 μm	2003
[8]	2.23/2.45	40/50	16	13	0,85 ^c	1.5/2	-109	0.35 μm	2003
[9]	5.14/5.70	10	13.5	7.25	0.495 ^c	2.5	-116	0.25 μm	2004
[10]	2.4/2.6	16	9	7	0.14 ^d	0.5/0.65	-113	90 ηm	2007
[11]	0.5/2	2/8	1.2 ^e		0,020 ^c	1.0	(5,8 ps^f)	90 ηm	2007
[12]	2.4	12	20.9		4,84 ^c	1.8	-126	0.18 μm	2007
[13]	1.2/1.9	50	10.8		0.509	1.8	(2,2 ps^f)	0.18 μm	2003
	0.6/1.6	33	7.3		0.42	1.8	(2,1 ps^f)	0.18 μm	2003
	1.5	10	6.0		0.74	1.8	(4,2 ps^f)	0.18 μm	2003

^aÁrea medida sin incluir el filtro y el VCO.

^bSe implementó una técnica para reducir el área del filtro, este dato incluye el área del filtro y el VCO.

^cÁrea medida incluyendo el filtro y el VCO.

^dÁrea medida incluyendo el VCO, sin el filtro.

^eOperando a 2.6 GHz.

^fPor el tipo de circuito (salida digital), se midió el *Jitter*.

^gMedido a 3 MHz.

- Ejecutar cambios de variables y operaciones necesarias para hacer el modelo del circuito compatible con la programación geométrica.
- Tener en cuenta las restricciones de la optimización convexa (la función objetivo y la restricción de la desigualdad deben ser convexas; mientras que la restricción de la igualdad debe ser lineal ($a_i^T x = b_i$)).
- Aplicar un software de optimización.
- Simular y validar resultados.

1.2. Estado del Arte

En la tabla 1.1 se muestra una breve compilación de trabajos destacables en el diseño de sintetizadores de frecuencia *CMOS*. Para la selección de los elementos en la tabla se tuvo en cuenta la función de cada sintetizador dentro del sistema específico, el estándar para el cual fueron diseñados y las ventajas que presenta cada trabajo. Adicionalmente, se consideró que la frecuencia de salida de los sintetizadores seleccionados no superara los 6 GHz, y que fueran diseñados para aplicaciones RF.

Se destacan los resultados obtenidos por trabajos como el de Pellerano [9], donde se implementaron divisores dinámicos que permiten reducir el consumo de potencia para aplicaciones de alta frecuencia. De la misma manera, el trabajo de Lee [7] propone una técnica de estabilización para los *PLLs*, agregando de manera novedosa un cero en la función de transferencia de lazo abierto, dando mayor amplitud a la relación entre tiempo de establecimiento y magnitud de las bandas laterales de salida. El trabajo de Yu [10] presenta un voltaje de alimentación bajo, utilizando polarización directa del pozo N para reducir el V_{th} y aumentar

la transconductancia; además utiliza arreglos de transistores con el doble de la longitud mínima de canal permitida, para aprovechar la reducción del V_{th} por la inversión de canal corto.

Por otra parte, el trabajo de Jung [11] presenta un consumo de potencia reducido respecto al estado del arte actual, implementando un *PLL* controlado por corriente, con frecuencia de salida independiente del factor multiplicativo y variaciones de proceso, voltaje y temperatura. Los demás trabajos seleccionados presentan ventajas en cuanto a potencia [6, 9, 11], ruido [8] y desempeño en términos generales. En [12] se logra cancelar el ruido de fase con una técnica de calibración adaptativa con base en un conversor digital analógico a la entrada del filtro, haciendo posible un incremento significativo en el ancho de banda.

De manera particular, el trabajo de Cabrera [6] desarrolla e implementa una serie de técnicas dirigidas a mejorar la operación del *PLL*. Técnicas para aumentar el rango del detector de fase mejorando el tiempo de establecimiento, la adaptabilidad de la corriente en la bomba de carga, y demás consideraciones en cuanto a ruido y consumo potencia convierten este trabajo en un aporte destacable. Para el caso del detector de fase se incluyó un detector de rango extendido, el cual se puede asimilar como un segundo detector de fase que se activa cuando el primer detector de fase supera 2π , multiplicando el rango de detección por dos.

Otro de los aportes importantes del trabajo de Cabrera es la implementación de divisores basados en la lógica *TSPC* [14, 15] y *E-TSPC* [16, 17] los cuales permitieron una reducción en el consumo de potencia, junto con un buen desempeño en términos de manejo de frecuencias. Sobre estos divisores se centró la mayoría del desarrollo de este proyecto, dado que su función es crítica al operar a muy altas frecuencias y ser el bloque más influyente en términos de diseño para la especificación del tiempo de establecimiento del *PLL*.

1.3. Metodología de diseño

Para la optimización de *PLLs* con programación geométrica, la literatura reporta el caso de Colleran [13], quien aplica la programación geométrica como método de optimización y automatización para el diseño de una serie de sintetizadores de frecuencia, logrando obtener resultados comparables con el estado del arte en términos de consumo de potencia, área y ruido de fase. Dentro del documento presentado por Colleran, se mencionan los procedimientos ejecutados para elaborar seis sintetizadores, optimizando diversas especificaciones y trabajando con diferentes frecuencias de referencia. Adicional al buen desempeño de los *PLLs*, se mejora el tiempo de diseño, pasando de algo que normalmente tomaría varias semanas a tan sólo unas horas.

Aunque es de resaltar la importancia de aportes como el de Colleran, es necesario profundizar más en la automatización y simplificación de la tarea de diseño del *PLL*. El área, la potencia, el *jitter*, el error de fase estático, el margen de fase, entre otras son las variables a optimizar en el trabajo de Colleran, las cuales corresponden a una optimización a nivel de sistema. Por otra parte, el enfoque de este trabajo es la optimización por bloques, en particular de los *flip-flop* que hacen parte del divisor. Este enfoque brinda más libertad con respecto a las topologías utilizadas porque permite variarlas, si las especificaciones así lo requieren. Incluso es posible emplear los bloques optimizados en aplicaciones distintas al divisor. Permitiendo aumentar la flexibilidad en el diseño y brindando mejor control al diseñador a cambio de la necesidad de un mayor conocimiento de los circuitos.

Así mismo, aunque el sintetizador es en principio un sistema analógico, muchos de sus componentes operan de manera digital lo cual lleva a pensar que es importante tomar en cuenta los aspectos digitales en cuanto a dimensionamiento de los transistores o demoras en la transmisión de señales. Por tal razón optimizar el *PLL* a partir de los bloques digitales que los componen es una alternativa interesante, donde se tienen dos componentes optimizados, el bloque digital que fue directamente optimizado y el *PLL* con las características heredadas de cada uno de los bloques que lo componen.

De esta manera se valida la aplicación de la programación geométrica como herramienta de optimización. Sin embargo la programación geométrica también puede ser utilizada como base para una metodología de diseño, sirviendo como criterio para la selección de los puntos de operación de un circuito determinado. Así, el presente proyecto se busca optimizar el bloque divisor dual 2/3 con lógica *E-TSPC*, el bloque divisor por dos con lógica *TSPC*, los cuales son elementos fundamentales por el consumo de potencia, y junto con el *VCO* son determinantes de la frecuencia fundamental de operación del sintetizador.

Adicionalmente se presenta la aplicación de la programación geométrica como metodología en el diseño del bloque divisor dual 2/3, buscando la frecuencia máxima de operación a la cual puede llegar dicha topología. Finalmente los resultados obtenidos del divisor se trasladan al sintetizador completo, para verificar las mejoras que se estaban buscando al momento de plantear el problema.

1.4. Organización del Documento

Este libro está estructurado de tal manera que se abarca cada tema a medida que es necesario en el desarrollo de las diversas temáticas. Los procedimientos y resultados están organizados de tal manera que sean presentados en forma breve y sencilla. A continuación se puede observar la organización de todo el documento.

El presente capítulo muestra los aspectos más generales de la programación geométrica y su aplicación en circuitos integrados, en particular en sintetizadores de frecuencia.

El capítulo II presenta algunas de las consideraciones más importantes para diseñar divisores de frecuencia y la aplicación de la programación geométrica en los *flip-flops* con lógica *E-TSPC* y con lógica *TSPC*. Al final de este capítulo se presenta una validación de los resultados luego de aplicar programación geométrica.

Finalmente el capítulo III se muestra la integración de los *flip-flops* con el *PLL* diseñado y las respectivas simulaciones del sintetizador presentado las mejoras obtenidas en el desempeño general del sintetizador. Luego se presenta un ejemplo para la aplicación de la programación geométrica como metodología junto con algunos scripts que permiten estructurar esta metodología como una herramienta. Al final de este capítulo se presenta el *layout* del integrado diseñado.

Capítulo 2

DIVISOR DE FRECUENCIA

El problema de optimización de un sintetizador de frecuencia se puede abordar de dos formas: optimizar todo el sistema en conjunto o cada bloque independientemente. La primera opción se reporta en el trabajo de Colleran [13], quien como se comentó en la sección 1.3, obtiene resultados significativos en consumo de potencia, área y ruido de fase luego de integrar en un solo programa geométrico restricciones generales del sistema. Por otra parte, optimizar diversos bloques por separado permite reconocer el aporte de cada bloque en una especificación general del sistema y brinda mayor flexibilidad al diseñador. La flexibilidad es muy importante, ya que permite modificar o reemplazar los bloques en función del flujo de diseño para conseguir alcanzar una determinada especificación.

En el caso de este proyecto, se determinó que para optimizar el sintetizador, enfocarse en el divisor de frecuencia es una opción muy atractiva. Esta decisión se tomó en base a: que la frecuencia de operación del *PLL* suele estar limitada por el divisor, que junto con el *VCO* es uno de los bloques que consumen mayor potencia y finalmente que algunos de los elementos constitutivos del divisor podrían presentar mejores características luego de ser optimizados. Estos elementos son los *flip-flops* tipo D basados en la lógica *TSPC* y *E-TSPC*, que conforman el divisor por 2 y el divisor de módulo dual 2/3 respectivamente. Al analizar y discutir con el autor de la técnica *E-TSPC*, Joao Navarro [16] sobre los compromisos de diseño en un *flip-flop*, fue posible concluir que enfocarse en la optimización de los *flip-flop* es una buena estrategia en el diseño de un sintetizador.

Basado en lo anterior este capítulo describe el proceso de diseño del divisor de frecuencia, utilizando programación geométrica. De acuerdo a las especificaciones más relevantes, se desarrolla la formulación e implementación del programa geométrico. Para esta última, se describe el procedimiento para utilizar el optimizador *CVX* [18], enfatizando en la estructura para el ingreso de los datos. Se realiza una verificación funcional de los datos mediante simulación de los circuitos propuestos, ejecutando pruebas asociadas con la frecuencia máxima de operación y los niveles lógicos.

2.1. Consideraciones iniciales

La base de funcionamiento de un *PLL* es la comparación de la fase entre la salida del *VCO* y la frecuencia de referencia. Por ello, un divisor de frecuencia es un elemento fundamental del sintetizador ya que al dividir la frecuencia de salida en el lazo de realimentación, se obtiene una señal cuya fase es comparable, con la fase de salida del oscilador local. La operación de un divisor de frecuencia se presenta en (2.1), donde f_{in} es la frecuencia de la señal de entrada, f_{out} la frecuencia de la señal de salida y el módulo n un valor mayor que uno, entero o fraccionario dependiendo de la arquitectura del divisor.

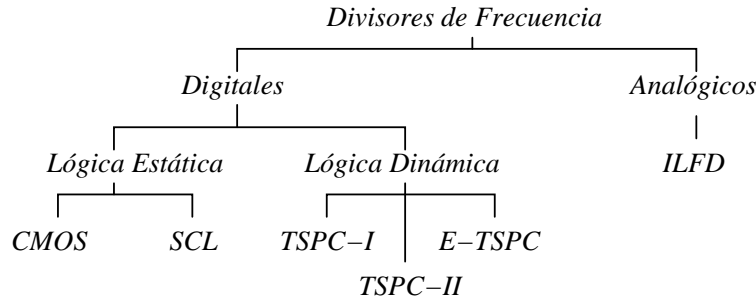


Figura 2.1: Clasificación general de los divisores de frecuencia.

$$f_{out} = \frac{f_{in}}{n} \quad (2.1)$$

2.1.1. Selección de Arquitectura

Existen múltiples tipos de divisores de frecuencia, los cuales se clasifican de acuerdo al rango de frecuencias en las que operan. La figura 2.1 extraída de [6], muestra un resumen de los divisores de frecuencia existentes organizados de izquierda a derecha, en forma ascendente de acuerdo a la frecuencia de operación.

Los divisores analógicos como el caso de los ILFD (*Injection Locked Frequency Dividers*) [19, 20] son diseñados para operar en frecuencias del orden de decenas de giga hertz. Sin embargo, presentan un rango reducido de frecuencia de operación y el área que ocupan dentro del integrado. Por su parte, los divisores digitales operan a menor frecuencia, tienen menor consumo de potencia y ocupan menos área respecto a los divisores analógicos. Entre estos se encuentran los que operan con lógica estática que basan su funcionamiento en celdas de memoria con circuitos biestables, pero con la desventaja de no poder operar a altas frecuencias.

A diferencia de los estáticos, los divisores dinámicos operan a frecuencias mayores ya que funcionan mediante las capacitancias parásitas de los transistores. Por esta razón, se ha popularizado su aplicación en la mayoría de los dispositivos inalámbricos que operan bajo los estándares *GSM* y *Bluetooth*. Entre las contribuciones más representativas a las celdas de tipo dinámico, se encuentran las basadas en la lógica *TSPC* (*True Single Phase Clock*) [14, 15] y en la lógica *E-TSPC* (*Extended-True Single Phase Clock*) [16, 17].

La lógica *TSPC* es una serie de reglas para formar circuitos digitales que opera solamente con una fase de la señal de reloj. Esto trae ventajas significativas al momento de implementar el circuito, ya que una sola fase reduce la aparición de pseudo-fases en el interior del circuito a causa de las demoras por la distribución interna de la señal de reloj. Adicionalmente, el número reducido de transistores la hace una opción muy popular al momento de diseñar circuitos digitales sincrónicos. En la figura 2.2 se observa un *flip-flop* con lógica *TSPC* propuesto por *Huang* [15], el cual es una versión mejorada de la presentada inicialmente por *Yuang* y *Svensson* [14]. La mejora se logra en la frecuencia máxima de operación, que alcanza un valor de 2.97 GHz . Sin embargo presenta condicionalidad en la frecuencia mínima de operación.

De manera similar, la lógica *E-TSPC* es en una extensión de la lógica *TSPC* tradicional y también consiste en una serie de reglas para componer circuitos digitales. La lógica *E-TSPC* hace uso de un conjunto de bloques para efectuar las interconexiones y aplicar las reglas. Algunos de estos bloques son tradicionales basados en lógica estática y dinámica, y algunos nuevos se proponen, como el *N-MOS Like*. Este último marca la principal diferencia entre los dos tipos de lógica, ya que reduce la cantidad de transistores para efectuar las operaciones lógicas. Un ejemplo de esto se presenta en la figura 2.3, donde se muestra un bloque dinámico con lógica *TSPC* y otro con *E-TSPC*, cabe resaltar que ambos circuitos producen el mismo resultado.

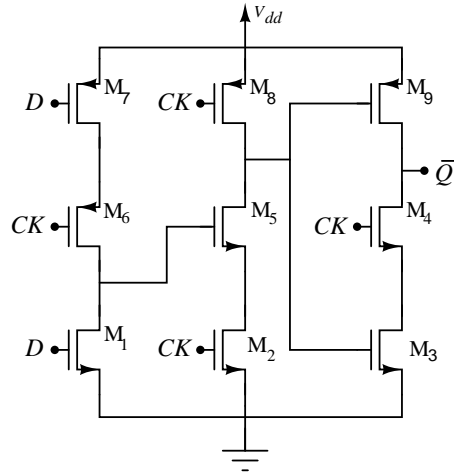


Figura 2.2: *Flip – Flop* tipo D, implementado con lógica *TSPC*.

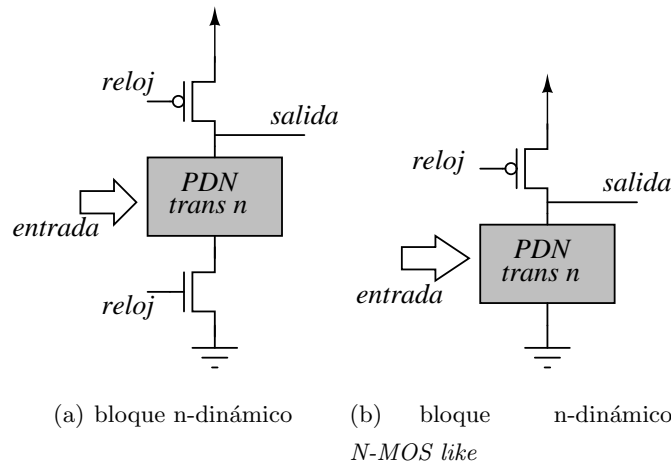


Figura 2.3: Diferencia principal entre *TSPC* y *E-TSPC*.

Las ventajas de la *E-TSPC* para el caso del *flip-flop* tipo D son bastante útiles, ya que tiene solo dos transistores en cada rama de corriente. Para compensar la falta de transistores intermedios uno de los dos transistores debe dominar en cada rama. Una de las principales ventajas de esto es reducir las capacitancias internas que incrementan las demoras propias del circuito, logrando un aumento considerable en la frecuencia máxima de operación. Un *flip-flop* implementado con lógica *E-TSPC* se presenta en la figura 2.4. Cada rama consta de un transistor tipo *NMOS* y otro tipo *PMOS*, que se encargan de efectuar *Pull-up* y *Pull-down* de la señal respectivamente.

Un divisor de frecuencia basado en la lógica *E-TSPC*, es presentado en la figura 2.5. Dicho divisor opera utilizando las demoras de los *flip-flop* tipo D, el cual opera captando la señal en un flanco de subida o bajada del reloj, manteniendola en ese nivel hasta que la entrada se modifique. Dentro del divisor, se interconectan los *flip-flop* en retroalimentación negativa, de tal forma que cada uno opera en niveles lógicos contrarios alternando la señal de salida. Por esta razón el *flip-flop* va a ser objeto de la optimización, pues un *flip-flop* con bajo tiempo de retardo equivale a un divisor con alta frecuencia de operación

En síntesis, tanto la lógica *TSPC* y la lógica *E-TSPC* presentan mejores características respecto a otros tipos de logica para la implementación de divisores de frecuencia. El rango de frecuencias en la cual opera, la menor cantidad de transistores respecto a otras técnicas, el consumo de potencia y el uso de una sola fase del

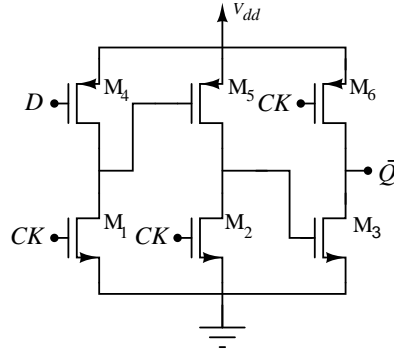


Figura 2.4: *Flip – Flop* tipo D, implementado con lógica *E-TSPC*.

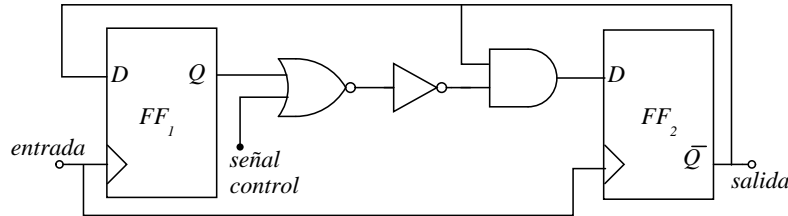


Figura 2.5: Divisor módulo 2/3 basado en *E-TSPC*.

reloj constituyen las principales ventajas. Por lo anterior, resulta adecuado adoptar esta arquitectura para los divisores de frecuencia del presente proyecto.

2.1.2. Especificaciones de Diseño

El divisor de frecuencia diseñado, debe cumplir con una serie de requerimientos que le permitan operar principalmente en el estándar *Bluetooth*. Tales requerimientos se asocian principalmente con el tiempo de establecimiento para el cambio de canales, el rango de frecuencias y el máximo nivel de ruido para cada frecuencia. Así mismo, es necesario considerar que se diseña para una aplicación inalámbrica. Por lo cual se deben tener en cuenta la tensión de alimentación, el consumo de potencia y el área ocupada por el circuito.

Uno de los elementos principales para la construcción de los divisores digitales, es el *flip-flop* tipo D. Por esta razón, al momento de diseñar también es necesario considerar, cual debe ser la carga a la cual va estar sometido y los niveles lógicos de operación. Un resumen de las especificaciones requeridas para el divisor de frecuencia, se presenta en la tabla 2.1.

2.2. Aplicación de la Programación Geométrica

Entre los mayores desafíos para los diseñadores de divisores de frecuencia se encuentra el lograr balancear todos los compromisos entre las diferentes variables, de tal forma que el diseño cumpla con todas las especificaciones requeridas. Dichas especificaciones pueden ser la potencia que consume el divisor, el número de transistores necesarios, la flexibilidad del diseño y la frecuencia máxima a la cual puede operar adecuadamente. Consecuentemente, la dificultad para encontrar el valor adecuado para las variables, radica generalmente en que al buscar mejorar una especificación se está afectando otra.

Por ejemplo en [21], donde utilizando una arquitectura de división basada en *flip-flops* tipo D de lógica *CML*, los cuales en condiciones ideales requieren corrientes suficientemente altas para lograr un amplio

Tensión de alimentación	3.3 [V]
Rango de Frecuencia	2.4 a 2.48 GHz
Consumo de potencia	minimizar
Área ocupada	minimizar
Tiempo de establecimiento	254μs
Carga máxima para los <i>flip-flop</i> *	aprox. 20fF
Margén de ruido *	1.3 [V] a 1.85 [V]

Tabla 2.1: Especificaciones de diseño para el divisor de frecuencia.

* Valores promedio, tomados de verificaciones experimentales.

Pueden variar según la operación del circuito.

rango de linealidad y buena transconductancia para la correcta adquisición de la señal. Sin embargo, una mayor transconductancia significa mayores tamaños de transistores y consecuentemente mayores capacitancias parásitas que deterioran la máxima frecuencia de operación. Presentándose así un fuerte compromiso, entre la ganancia en pequeña señal de los transistores y la máxima frecuencia de operación. Similarmente, en [9] que utilizan tanto la lógica *TSPC*, como la lógica *E-TSPC*, relacionan un fuerte compromiso entre la frecuencia de operación y la potencia consumida. Dicho compromiso se asocia al correcto dimensionamiento de los transistores para mantener el dimensionamiento ideal de los transistores, que evite desperdicios de potencia a altas frecuencias.

Por lo anterior, un buen diseño se constituye obteniendo lo mejor de cada circuito, encontrando el punto exacto para cada una de las variables. Una forma de lograr esto es mediante un profundo conocimiento del circuito, tanto en la operación como en la teoría, lo cual generalmente se traduce en una amplia experiencia por parte del diseñador. Sin embargo, una alternativa bastante interesante al problema de diseño, consiste en utilizar algún tipo de ayuda que indique cuál es la tendencia de las diferentes variables en función de los compromisos.

Tal y como se mencionó en el capítulo 1, una de estas ayudas la constituye la programación geométrica. Ya que, dado un espacio n-dimensional de diseño es posible encontrar un valor óptimo para cada una de las variables involucradas en el problema. Por esta razón, es muy importante asegurar que los datos con los que va a trabajar el programa geométrico sean lo más exactos posible. Así, la formulación debe estar en función de la especificación que se desea optimizar, tomando en cuenta los aspectos importantes y el modelo mas adecuado para cada situación.

2.2.1. Formulación del Problema

La formulación es una parte fundamental en la aplicación de la programación geométrica, ya que involucra un conocimiento adecuado de la operación del circuito. En cierta forma, implica traducir los datos de un modelo físico a un modelo matemático que pueda interpretar el programa geométrico. De allí, es importante conocer muy bien el circuito con el que se va a trabajar y cuáles son los aspectos más relevantes en la operación.

Como se mencionó anteriormente la frecuencia máxima de operación es una de las principales especificaciones al momento de definir la operación de un divisor de frecuencia. Como tal, es importante reconocer como

se define y cuales son las variables involucradas en lograr el mejor valor para esta. Dado que la frecuencia es una medida de la rapidez con la que se transmiten las señales a través del circuito, se hace necesario abordar el análisis a partir del tiempo que demora cada señal en pasar de un punto a otro en el circuito. Es entonces el tiempo de demora la función que se desea optimizar con el objetivo de mejorar la frecuencia máxima de operación, ya que a menor demora, mayor es la frecuencia.

Como consecuencia de tomar la demora como la función objetivo del problema de optimización, otras especificaciones como la potencia y el área se convierten en restricciones del programa geométrico. Además de estas, otras restricciones propias del comportamiento del *flip-flop* deben ser incluidas dentro de la programación para lograr una correcta formulación cercana al comportamiento del circuito. Entre tales restricciones es posible encontrar el ancho de los transistores, la corriente de cada rama del circuito y la tensión que debe permanecer entre el drenaje y la fuente de cada transistor.

En términos formales se tiene la ecuación (2.2) que presenta a modo general como se pretende abordar el problema de optimización, donde D es la demora¹, P es la potencia a consumir, y A es el área.

$$\begin{aligned} \text{minimizar} \quad & D \\ \text{sujeto a} \quad & P \leq P_j \\ & A \leq A_j \end{aligned} \tag{2.2}$$

De acuerdo a la formulación presentada en (2.2) es necesario plantear los modelos matemáticos de circuito que van a permitir extraer las diversas relaciones necesarias para el programa geométrico. Estos modelos deben tener la capacidad de relacionar mediante ecuaciones el tiempo de demora de cada una de las señales, asociando esto con parámetros propios del transistor o de la operación del circuito. En las siguientes secciones se muestra la forma como se extraen estos modelos, para los *flip-flop* basados en *TSPC* y *E-TSPC*.

Flip-flop con lógica TSPC

Inicialmente en cualquier circuito digital es importante identificar las rutas que toma una señal para cada transición específica. Es decir, realizar una revisión de los transistores activos en cada rama de acuerdo a las señales que entran y salen de cada uno de los dispositivos. A partir de esta identificación, es posible determinar cuales son los nodos críticos para la transmisión de la señal de una rama a otra. Una vez identificados estos nodos, se hace una revisión de las capacitancias asociadas a cada uno con el objetivo de obtener un estimado de la demora asociada a cada rama del circuito.

Dado que en este caso se busca la formulación de un *flip-flop* basado en *TSPC*, es necesario implementar un modelo que me permita cuantificar la velocidad de respuesta del circuito y en particular de cada transistor frente a las diferentes señales. Tomando en cuenta lo anterior, un circuito RC sirve de modelo para conocer el tiempo de demora de un circuito, ya que la combinación de las capacitancias y resistencias determina el tiempo de carga y descarga de las señales en cada nodo del circuito [22].

Para lograr obtener un circuito RC se modela eléctricamente cada transistor con tres elementos: una resistencia, una capacitancia y un interruptor. La resistencia representa la oposición al paso de la corriente que presenta cada transistor, la capacitancia determina la carga que puede acumular el transistor luego de alguna transición o cambio de estado y finalmente el interruptor muestra si el transistor está conduciendo o no ($V_{GS} \geq V_T$). De acuerdo a esto se representa cada transistor con su equivalente RC, lo que finalmente resulta en el circuito presentado en la figura 2.6.

¹Se define como el tiempo que toma la salida de un dispositivo para cambiar a un nuevo valor, luego de que la entrada ha cambiado.

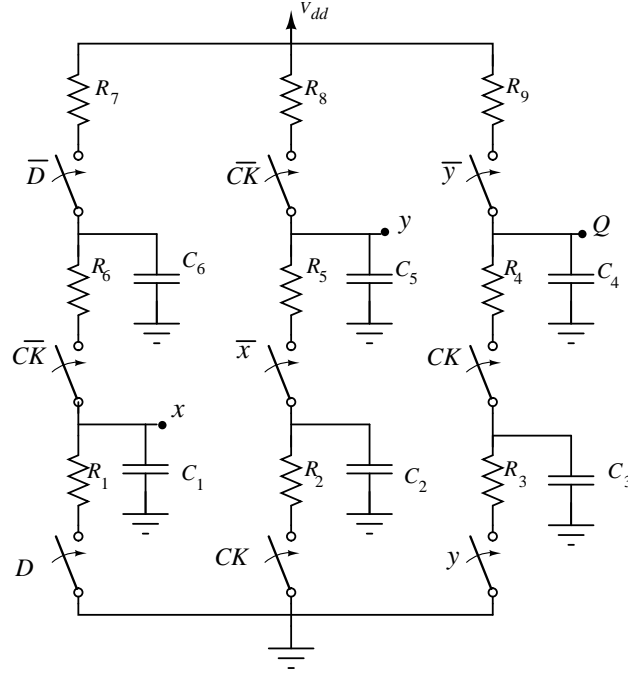


Figura 2.6: Equivalente RC de un *flip-flop* tipo D con lógica *TSPC*.

De acuerdo a Rabaey [23], es adecuado asumir en un circuito digital una resistencia infinita cuando no esta conduciendo y una resistencia de un valor finito cuando esta conduciendo. Sin embargo, el principal problema de esta afirmación radica en que cuando el transistor esta activo, la resistencia no es constante. Por el contrario la resistencia presenta un comportamiento variable en el tiempo y es no lineal, pues depende del punto de operación del transistor.

Aún así, resulta razonable trabajar en circuitos digitales con una resistencia lineal y constante, ya que permite una relativa sencillez en los cálculos y las aproximaciones realizadas tienen una exactitud aceptable. De esta manera, en un circuito digital donde no existen cambios muy fuertes en la linealidad resulta adecuado utilizar un valor promedio de la resistencia operando sobre la región de interés, o lo que equivale a la ecuación (2.3).

$$R_{eq} = \text{promedio}_{t=t_1, t_2, \dots, t_n}(R_{enc}(t)) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} R_{enc}(t) dt = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \frac{V_{DS}(t)}{I_D(t)} dt \quad (2.3)$$

En el caso mas común de operación de circuitos digitales las resistencias actúan como elemento para la carga y descarga de un capacitor de V_{DD} hacia tierra o viceversa. Tomando como base lo anterior y asumiendo una descarga desde V_{DD} a $V_{DD}/2$ (el cual es el punto de interes para el calculo de la demora), se tiene la ecuación (2.4).

$$R_{eq} = \frac{1}{-V_{DD}/2} \int_{V_{DD}}^{V_{DD}/2} \frac{V}{I_{DSAT}(1 - \lambda V)} = \frac{3V_{dd}}{4I_{DSAT}} \left(1 - \frac{5}{6} \lambda V_{dd} \right) \\ \text{con } I_{DSAT} = k' \frac{W}{L} \left((V_{DD} - V_T) V_{DSAT} - \frac{V_{DSAT}^2}{2} \right) \quad (2.4)$$

De la ecuación (2.4) es importante destacar que la resistencia obtenida es inversamente proporcional a la relación de aspecto (W/L) del transistor y por lo tanto a mayor ancho, menor es la resistencia. Para efectos de cálculo, el voltaje V_{DSAT} que representa la tensión critica a la cual la velocidad de los portadores de carga

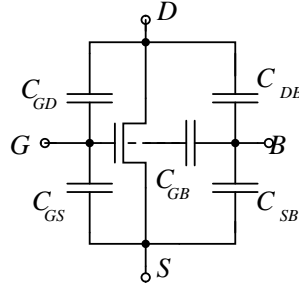


Figura 2.7: Modelo capacitivo de un transistor MOSFET.

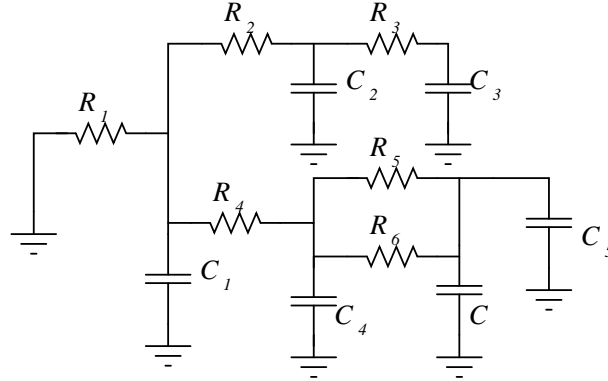


Figura 2.8: Ejemplo de un árbol RC.

se convierte en una constante proporcional al campo eléctrico (lo que representa una movilidad constante de los electrones), se puede calcular en función del largo del transistor L , la velocidad de saturación v_{sat} y la constante de movilidad de los electrones μ_n . La ecuación (2.5) muestra como se calcula este valor. Los demás elementos presentes en la ecuación (2.4) son constantes que pueden ser obtenidas a partir del proceso de fabricación, o de circunstancias propias del circuito.

$$V_{DSAT} = \frac{Lv_{sat}}{\mu_n} \quad (2.5)$$

Por otra parte, para hallar las capacitancias equivalentes de cada nodo denotadas en la figura 2.6 como $C_1 - C_6$, es necesario tomar en cuenta los aportes capacitivos que realiza cada transistor hacia el nodo. Una forma de visualizar los aportes capacitivos de cada transistor es utilizar un modelo capacitivo, como el de la figura 2.7. En dicha figura se observan de forma concentrada las capacitancias que presenta un transistor, en toda su geometría. En el modelo, se observan las capacitancias que van conectadas en cada nodo y de acuerdo a ello se determina la capacitancia equivalente.

Por ejemplo para la capacitancia C_1 , se observan los aportes del drenos transistor M_1 , del drenos transistor M_6 y de la puerta del transistor M_5 . El resultado de la capacitancia equivalente C_1 se observa en la ecuación (2.6).

$$C_1 = C_{db1} + C_{db6} + 2(C_{gd6} + C_{gd1}) + C_{g5} \quad (2.6)$$

De la observación del circuito en la figura 2.6, la identificación de los nodos críticos y las ecuaciones obtenidas del circuito RC, es posible mediante la técnica *Elmore Delay* [24] calcular manualmente un estimado para el tiempo de demora asociado con cada rama del circuito en una transición específica.

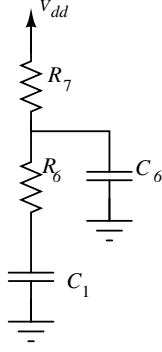


Figura 2.9: Circuito equivalente de la rama 1 para cuando $CK = 0$ y $D = 0$.

La técnica *Elmore Delay* dice: “Sea $v_i(t)$ el voltaje en el nodo i y suponer que todos los capacitores de un árbol RC (figura 2.8) están inicializados con un voltaje $v_i(0) = 1$. Entonces, se puede demostrar que cada uno de los voltajes del árbol RC se decreta monótonicamente hasta cero, mientras $t \rightarrow \infty$. Así, el *Elmore Delay* en el nodo i , está definido por la ecuación (2.7)” [25].

$$D_i = \int_0^t v_i(t) dt \quad (2.7)$$

$$D_i = \sum_{j \in \text{Ruta}(i)} R_j C_j^{\text{tot}} \quad (2.8)$$

La ecuación (2.7) también puede ser interpretada de manera discreta como una sumatoria de la demora en cada uno de los nodos del árbol. Por lo tanto, se tiene la ecuación (2.8), y aplicándola en el ejemplo de la figura 2.8, se obtiene como resultado la ecuación (2.9).

Retomando el análisis para el *flip-flop* de la figura 2.9, en el caso de la primera rama se tiene que la demora es la presentada en la ecuación (2.10). Este mismo procedimiento debe ser utilizado en cada una de las ramas del circuito y para cada transición específica. De esta manera es necesario calcular todas las posibles combinaciones de circuitos, para los diferentes niveles lógicos que toma el reloj y las demás entradas. Haciendo la combinatoria para cada rama, con dos opciones para cada transistor y tres transistores por rama, se tiene $2^3 = 8$ posibles circuitos por cada rama, y con tres ramas se tiene un total de 24 posibles transiciones. Cabe destacar que algunos circuitos pueden ser directamente eliminados del análisis, si por simple inspección es claro que no presentan el peor caso de demora para el circuito. Esta inspección se fundamenta en la aparición de resistencias en paralelo, las cuales disminuyen notablemente el valor final de la demora.

$$D_3 = R_1(C_1 + \dots + C_6) + R_2(C_2 + C_3) + R_3C_3 \quad (2.9)$$

$$D_{1_{R1}} = R_6C_1 + R_7(C_1 + C_6) \quad (2.10)$$

De la misma forma, utilizando como base el cálculo de las capacitancias para cada nodo, es posible obtener una estimación del consumo de potencia dinámico del *flip-flop*. La ecuación (2.11) muestra esta relación, donde f_i es la frecuencia de operación del circuito y C_i^L es la capacitancia de cada nodo.

La potencia estática, es la potencia disipada aún cuando las entradas del circuito son constantes, y es determinada por la ecuación (2.12). De esta ecuación \bar{I}_i^{perd} corresponde a la corriente que consume cada rama con relación de aspecto mínimo y puede ser determinada mediante simulación, mientras que x_i es un factor de escalamiento de acuerdo al ancho de cada transistor en las ramas (puede ser equivalente a W/L).

Componente	Forma compatible con PG	Equivalencia
R_{eq}	$R_{eq} = \frac{R_i}{x_i}$	$R_i = \frac{3}{2}V_{dd} \left(1 - \frac{7}{9}V_{dd}\right) \left[\frac{1}{k' [2(V_{dd}-V_i)V_{DSAT} - V_{DSAT}^2]} \right]$ $x_i = \frac{\omega}{L}$
C_{db}	$C_{db} = C_{db_i}x_i$	$C_{db_i} = k_{eq}AD CJ + k_{eqsw}2CJSW$ $x_i = \omega$
C_{gd}	$C_{gd} = C_{gd_i}x_i$	$C_{gd_i} = C_{ox}\frac{L}{2} + CGDO$ $x_i = \omega$
C_{gs}	$C_{gs} = C_{gs_i}x_i$	$C_{gs_i} = \frac{2}{3}LC_{ox} + CGSO$ $x_i = \omega$
C_{gb}	$C_{gb} = C_{gb_i}x_i$	$C_{gb_i} = C_{ox}L$ $x_i = \omega$

Tabla 2.2: Compatibilidad de las ecuaciones del *flip-flop* tipo D con la programación geométrica.

Finalmente, la potencia total consumida por la celda corresponde a la suma de la potencia estática y la potencia dinámica, tal y como se observa en la ecuación (2.13).

$$P_{din} = \sum f_i C_i^L V_{dd}^2 \quad (2.11)$$

$$P_{est} = \sum \bar{I}_i^{perd} x_i V_{dd} \quad (2.12)$$

$$P_{tot} = P_{din} + P_{est} \quad (2.13)$$

Con respecto al área, se definió una aproximación en base a una constante multiplicada por la relación de aspecto de cada transistor. Dicha constante A_i , se utiliza para aproximar la relación de aspecto al área total para un transistor del tamaño mínimo impuesto por las reglas de fabricación, por lo tanto solo es adecuada mientras los transistores diseñador mantengan un tamaño reducido. Tal relación se presenta en la ecuación (2.14).

$$A_{tot} = \sum x_i A_{const} \quad (2.14)$$

Una vez obtenidas las ecuaciones que definen parcialmente el comportamiento del *flip-flop*, es necesario llevar cada una a forma compatible con programación geométrica. Es decir, cada ecuación debe quedar en forma monomial y/o posinomial, según sea el caso. En la tabla 2.2.1 se presentan las equivalencias y relaciones utilizadas para llevar a forma compatible con programación geométrica las diferentes ecuaciones.

Para el caso de las resistencias R_i es una constante que depende de parámetros del proceso y de la tensión de alimentación del circuito y x_i es la relación de aspecto del transistor. Para las diferentes capacitancias x_i representa el ancho de cada transistor y la constante que lo acompaña esta dada por los parámetros específicos de cada capacitancia, se define mediante datos del proceso de fabricación y algunas condiciones de polarización del circuito. Es importante destacar que todos los modelos aquí propuestos son de tipo monomial, y los datos para el cálculo de las constante se han obtenido a partir de los datos de proceso que suministra el fabricante que en este caso es *AMS* para el proceso C35B4 de $0.35\mu m$ CMOS.

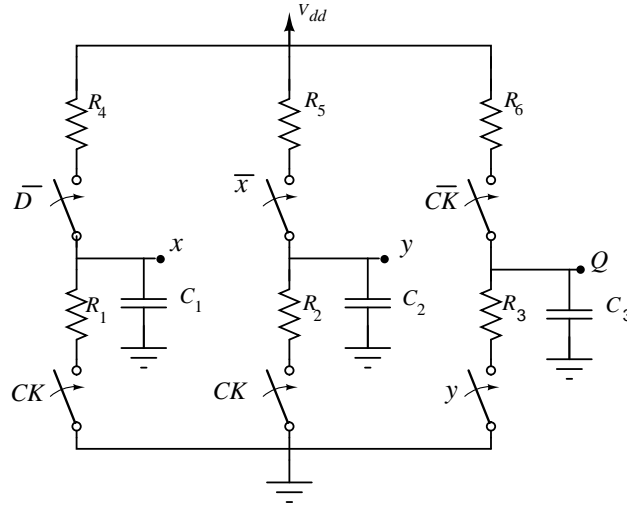


Figura 2.10: Equivalente RC de un *flip-flop* tipo D con lógica *E-TSPC*

Como se puede notar en la tabla 2.2.1 tanto el modelo para las resistencias como para las capacitancias, es de tipo monomial. Donde cada uno es función de una constante por el ancho del transistor, elevado a la potencia 1 o -1 según corresponda. Este tipo de modelos, según se determina en [25] es adecuado para trabajar con circuitos digitales, donde se busque minimizar la demora del circuito.

Flip-flop* con lógica *E-TSPC

En el *flip-flop* tipo D, basado en la lógica *E-TSPC* la mayoría de las consideraciones son similares al caso de *TSPC*, ya que estas relaciones son propias de cualquier transistor MOS. No obstante, en esta sección se destacaran los aspectos distintivos para este tipo de lógica, resaltando las diferencias cuando sea pertinente.

Inicialmente se debe realizar el circuito RC equivalente, tal y como se observa en la figura 2.10. Para el caso de la resistencia, es posible utilizar la expresión (2.4). Con respecto a las capacitancias, es necesario obtener las capacitancias asociadas con cada nodo. Dado que todas las ramas se componen tanto de un transistor NMOS y un transistor PMOS en serie, es posible establecer una capacitancia genérica, de la cual solo varían las relaciones entre los componentes. De tal manera, la capacitancia resultante se observa en la ecuación (2.15).

$$C_{equ} = 2(C_{gd_n} + C_{gd_p}) + C_{db_n} + C_{db_p} + C_{fanout} \quad (2.15)$$

En este caso particular no es necesaria la búsqueda de nodos críticos, ya que por simple inspección es posible decir que los tres nodos donde se unen los drenos de los tres pares de transistores, son nodos de interés que afectan el paso de la señal por cada una de las ramas y por lo tanto son nodos críticos. Para el caso cuando están conduciendo ambos transistores de cada rama el circuito equivalente resulta con dos resistencias en paralelo, eliminando este estado como el peor caso de demora. Esto deja tan solo dos posibles transiciones por rama para analizar, un caso con el transistor tipo NMOS activo y el PMOS apagado como se observa en la figura 2.11a, y otro con el PMOS activo y el NMOS apagado como se observa en la figura 2.11b.

El análisis de las transiciones deja como resultado las ecuaciones de demora presentadas en 2.16, el cual por definición, equivale al tiempo requerido para que el voltaje a la salida del circuito RC llegue al punto medio de los niveles lógicos de voltaje (p.e. 0 y V_{dd}). A diferencia del caso con *TSPC*, acá no es necesario utilizar la técnica *Elmore Delay* ya que la topología del circuito permite realizar un análisis similar al de un

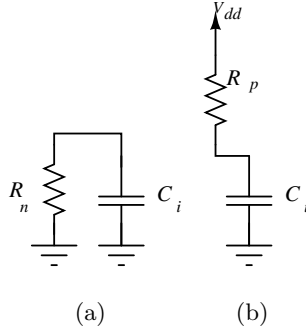


Figura 2.11: (a) Circuito equivalente para PMOS=0 NMOS=1. (b) Circuito equivalente para PMOS=1 NMOS=0.

inversor, del cual son bastante conocidas las relaciones para los tiempos tanto de subida como de bajada [23].

Con respecto al consumo de potencia y la estimación del área ocupada por el circuito, es factible utilizar las ecuaciones (2.11 - 2.14). Así mismo, la información contenida en la tabla 2.2.1 es compatible tanto en el caso *TSPC*, como de *E-TSPC*.

$$\begin{aligned}
 D_1 &= 0,69R_pC_i \\
 D_2 &= 0,69R_nC_i
 \end{aligned}
 \tag{2.16}$$

2.2.2. Implementación en un Programa Geométrico

Una vez obtenida la formulación del problema como programa geométrico, se utiliza un software que permita ejecutar el algoritmo de solución. Existen algoritmos muy eficientes disponibles a través de paquetes de software (p.e. *MOSEK* [26], *CVX* [18], *TOMLAB* [27], *COPL-GP*, *YALMIP* [28]). La mayoría de ellos posee una sintaxis compleja para el ingreso de datos, convirtiendo su ejecución en una labor delicada y tediosa.

Por su parte, *CVX* es un sistema de modelado basado en MATLAB®, permitiendo especificar las restricciones y la función objetivo mediante la sintaxis estándar. Adicionalmente, permite escribir programas geométricos en forma no convexa, lo cual reduce la complejidad en el proceso de implementación. Adicional a esto, *CVX* se distribuye para ser utilizado libremente, tanto en fines académicos como comerciales. Gracias a lo anterior no es necesario el pago de licencias para utilizar el programa, permitiendo que las aplicaciones basadas en *CVX* se puedan emplear en cualquier entorno. Dados los argumentos expuestos anteriormente, se optó por *CVX* como herramienta para la solución de los programas geométricos facilitando el proceso de automatización del diseño.

Programación geométrica con *CVX*

El flujo seguido para la construcción de un programa geométrico se inicia con la definición de las constantes envueltas en el planteamiento, las cuales provienen del proceso de fabricación, y de las especificaciones propuestas al inicio del capítulo. Ejemplo de ellas son la movilidad de los portadores μ_n y μ_p , la capacitancia por unidad de área del oxido de la puerta del transistor C_{ox} , la velocidad del electrón en saturación v_{sat} , la tensión de umbral V_{TH} o la constante para el calculo de las capacitancias $CGDO$, CJ , entre otras. Dichas constantes fueron extraídas de los documentos proporcionados por AMS y en otros casos con base en algunos cálculos que involucran la geometría del transistor o la polarización del mismo.

A partir de estas constantes se crean expresiones que representan las variables relevantes en el cálculo del desempeño del circuito, en este caso, las capacitancias y resistencias asociadas a un transistor NMOS. En la ecuación (2.17) se muestra un ejemplo donde se define la capacitancia C_{gs} intrínseca del transistor. Como se puede notar en la ecuación no se incluyó el ancho del transistor W , ya que lo se busca es calcular el elemento C_{gs_i} mencionado en la tabla 2.2.1. Cabe aclarar que las capacitancias tampoco son elementos lineales, y varían en función de las regiones de operación. Para tomar en cuenta dicho efecto se utilizó siempre el valor más alto de capacitancia, que representa el peor caso. La tabla 2.3 desarrollada por [23], muestra una distribución promedio de las capacitancias de canal para las diferentes regiones de operación.

Tabla 2.3: Distribución promedio de la capacitancia de canal, para un transistor MOS.

Región de operación	C_{GCB}	C_{GCS}	C_{GDC}	C_{GC}	C_G
Corte	$C_{ox}WL$	0	0	$C_{ox}WL$	$C_{ox}WL + 2C_0W$
Resistivo	0	$C_{ox}WL/2$	$C_{ox}WL/2$	$C_{ox}WL$	$C_{ox}WL + 2C_0W$
Saturación	0	$(2/3)C_{ox}WL$	0	$(2/3)C_{ox}WL$	$(2/3)C_{ox}WL + 2C_0W$

$$C_{gs_n} = CGSO_n + \frac{2}{3}LC_{ox} \quad (2.17)$$

De manera similar, en el caso de la resistencia se realiza el cálculo para obtener el valor de R_i . Respecto a la capacitancia de carga del circuito, se incluyó en la definición de capacitancias de nodo. Tomandola como si se tuviera otro *flip-flop* conectado a la salida, es decir la carga esta definida por las capacitancias C_{gs} y C_{gb} de un transistor.

Una vez definidas estas variables se crea un arreglo de datos tipo estructura que permitió asignar a cada transistor sus elementos correspondientes, proporcionando la base para la implementación formal del problema de optimización caracterizado por la sentencia `cvx_begin gp`. Un ejemplo de dicha estructura, se puede observar a continuación:

```

NMOS=struct('R',Ri_n,'Cdb',Cdbc_n,'Cgd',Cgdc_n,'Cgs',Cgsc_n,'Cgb',Cgbc);
PMOS=struct('R',Ri_p,'Cdb',Cdbc_p,'Cgd',Cgdc_p,'Cgs',Cgsc_p,'Cgb',Cgbc);

```

Cabe notar que los datos incluidos posterior a esta sentencia forman parte del algoritmo a optimizar por *CVX* y por lo tanto deben estar expresados en forma matemáticamente clara. Para este ejemplo, se ubicaron dentro del algoritmo de optimización la definición de las capacitancias de nodo y las resistencias (incluyendo el ancho W), junto con las expresiones para calcular el retardo, la potencia y el área.

Finalmente se definió la función a optimizar junto con el conjunto de restricciones a tomar en cuenta. Para definir las restricciones, es necesario que el diseñador tenga conocimiento previo de la forma como funciona el circuito. Por ejemplo, para el caso de los anchos es necesario definir un rango a partir del cual *CVX* va a buscar el mejor valor para optimizar la función objetivo. El límite inferior para este rango esta dado por las reglas del proceso, que en este caso es de $0,6\mu m$, mientras que el límite superior se decidió tomando el máximo valor del diseño propuesto por [6].

Para el caso de la longitud de canal, se tomo siempre el valor minimo para maximizar la relación de aspecto (W/L). A continuación se presenta un ejemplo del código implementado para optimizar el *flip-flop* con lógica *E-TSPC*.

```
% divisoretspc.m
```

```

%El presente archivo ejecuta el algoritmo de optimización con programación geométrica
%usando el software para optimización CVX de Stanford
%***** Flip-Flop D *****

%calculo de los valores constantes para las capacitancias
Cdbc_n=keq_n*Adc_n*CJ_n+keqsw_n*2*CJSW_n; Cdbc_p=keq_p*Adc_p*CJ_p+keqsw_p*2*CJSW_p;
Cgdc_n=Cox*L/2+CGD0_n; Cgdc_p=Cox*L/2+CGD0_p;
Cgsc_n=CGS0_n+2/3*L*Cox; Cgsc_p=CGS0_p+2/3*L*Cox;
Cgbc=Cox*L; %se toma este valor para cuando esta cortado o en triodo el transistor ya que es el peor caso
%Calculo de los valores constantes para las resistencias
VDSAT_n=(L*vsat_n)/u_n; VDSAT_p=(L*vsat_p)/u_p;
Ri_n=L*3/2*Vdd*(1-7/9*lambda_n*Vdd)*(1/(k_n*((Vdd-vth_n)*VDSAT_n-VDSAT_n*VDSAT_n)));
Ri_p=L*3/2*Vdd*(1-7/9*lambda_p*Vdd)*(1/(k_p*((Vdd-vth_p)*VDSAT_p-VDSAT_p*VDSAT_p)));

N=6;

%Definicion de estructuras para los diferentes valores
NMOS=struct('R',Ri_n,'Cdb',Cdbc_n,'Cgd',Cgdc_n,'Cgs',Cgsc_n,'Cgb',Cgbc);
PMOS=struct('R',Ri_p,'Cdb',Cdbc_p,'Cgd',Cgdc_p,'Cgs',Cgsc_p,'Cgb',Cgbc);

%INICIO DEL PROGRAMA GEOMETRICO
cvx_begin gp

variables w(N) Vov(N) I(N)

device(1:3)=NMOS; device(4:6)=PMOS;% definicion del tipo de cada dispositivo o estructura

for num=1:N

device(num).R=device(num).R/w(num);
device(num).Cdb=device(num).Cdb*w(num);
device(num).Cgd=device(num).Cgd*w(num);
device(num).Cgs=device(num).Cgs*w(num);
device(num).Cgb=device(num).Cgb*w(num);
end

%Capacitancias
C1=2*(device(4).Cgd+device(1).Cgd)+device(4).Cdb+device(1).Cdb+device(2).Cgb+device(5).Cgb+device(2).Cgs+device(5).Cgs;
C2=2*(device(5).Cgd+device(2).Cgd)+device(5).Cdb+device(2).Cdb+device(3).Cgb+device(6).Cgb+device(3).Cgs+device(6).Cgs;
C3=2*(device(6).Cgd+device(3).Cgd)+device(6).Cdb+device(3).Cdb+device(3).Cgb+device(6).Cgb+device(3).Cgs+device(6).Cgs;

%Resistencias
R=[device.R]';

%Area
area=sum(w)*L;

%Demoras asociadas a cada una de las transiciones de interes

D1=R(3)*C3;
D2=R(4)*C1;
D3=R(5)*C2;
D4=R(6)*C3;
D5=R(1)*C1;
D6=R(2)*C2;

```

```
minimize(max([D1 D2 D3 D4 D5 D6]))
subject to
w<=9e-6;

w(1)>=2e-6;
w(2)>=1e-6;
w(3)>=1e-6;
w(4)>=2e-6;
w(5)>=1e-6;
w(6)>=2e-6;

w(4)>=3.2*w(1);
w(5)<=3.3*w(2);
w(6)<=3.2*w(3);

cvx_end
```

El *script* generado se guarda con extensión `.m` MATLAB. Una vez terminado el proceso de optimización, se presentan en la consola los resultados de la optimización y el valor optimizado de la función objetivo que en este caso fue el tiempo de retardo. Los valores para las restricciones y demás cálculos son almacenados como variables de MATLAB.

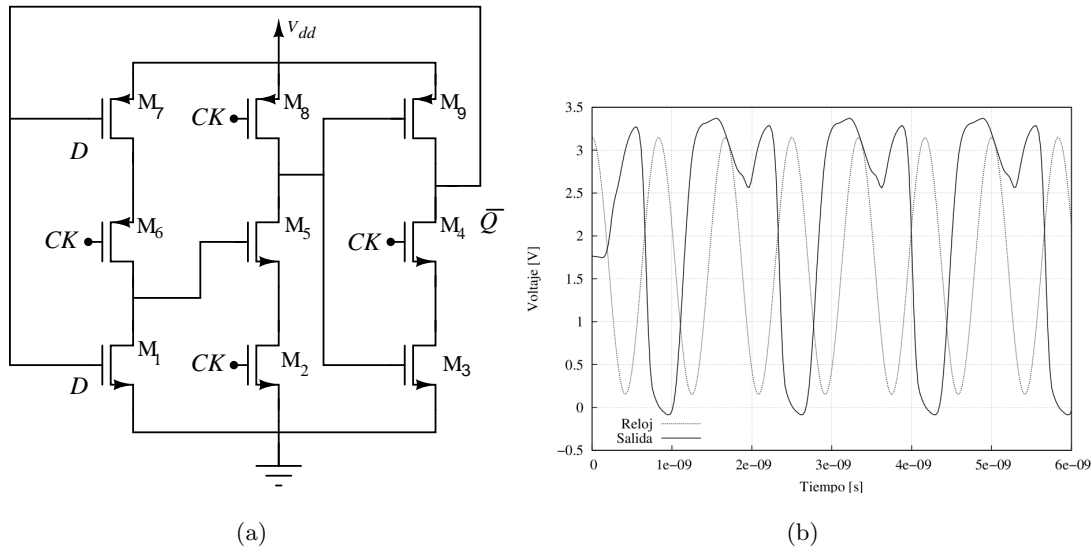


Figura 2.12: a) Divisor dual basado en la lógica *TSPC*. b) Respuesta de un *flip-flop* tipo D *TSPC*, configurado como divisor dual.

2.3. Verificación de Resultados

Luego plantear y resolver los programas geométricos para el *flip-flop* con lógica *TSPC* y el *flip-flop* con lógica *E-TSPC*, se hace necesario utilizar un simulador con el fin de comparar y validar los resultados obtenidos mediante la programación geométrica. Las simulaciones se efectuaron con *Hspice* de acuerdo a la especificaciones de la sección 2.1.2, utilizando los modelos de transistores BSIM3V3 de nivel 49 proporcionados por *AMS* para el proceso C35B4.

2.3.1. *Flip-flop* con Lógica *TSPC*

Bluetooth

Con base en la optimización del *flip-flop* con lógica *TSPC*, se obtuvieron los anchos de los transistores que se presentan en la tabla 2.4. Utilizando estos datos realiza una simulación con el *flip-flop* configurado como un divisor de onda dual como se muestra en la figura 2.12a, conectando la salida \bar{Q} a la entrada *D*.

Dada la posición del divisor módulo 2 en el divisor de frecuencia completo, operando bajo el estándar *Bluetooth* la máxima frecuencia a la que deba trabajar sera de 1.2 GHz. Por lo tanto, se simula con un reloj senoidal de frecuencia 1.2 GHz y amplitud de 1.5 [V] con un *offset* de 1.65 [V]. La gráfica 2.12b muestra los resultados, donde se observa que la salida tiene la mitad de la frecuencia que la señal de reloj.

Máxima Frecuencia

Una vez verificada la correcta operación del circuito a 1.2 GHz se realiza un barrido con un paso 0.2 GHz en la frecuencia de entrada, con el fin de observar cual es el límite superior para la frecuencia de operación dada la función objetivo del PG que buscaba optimizar la máxima frecuencia de operación. En cada punto del barrido se observó la onda de salida del divisor, notandose que para frecuencias superiores a 3.6 GHz comienza a observarse un deterioro en la simetría de la onda, pero que continua siendo aceptable dentro del margen de ruido establecido. Finalmente para frecuencias superiores a los 3.8 GHz, el circuito opera de manera completamente inadecuada como se observa en la figura 2.13, dejando de operar como un divisor. La

Anchos de los transistores [μm] *								
M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9
4.78	8.99	2.00	8.93	4.03	2.03	4.06	2.18	2.00
Consumo de Potencia								
PG			<i>Hspice</i>			%error		
1.44 mW			1.52 mW			5.26 %		
Tiempo de demora								
PG			<i>Hspice</i>			%error		
46.42ps			51.50ps			9.86 %		

Tabla 2.4: Aplicación de la programación geométrica de un *flip-flop* tipo D, con lógica *TSPC*.

* Para el largo de todos los transistores se utilizo el L_{min} , permitido por la tecnología, de $0.35\mu m$.

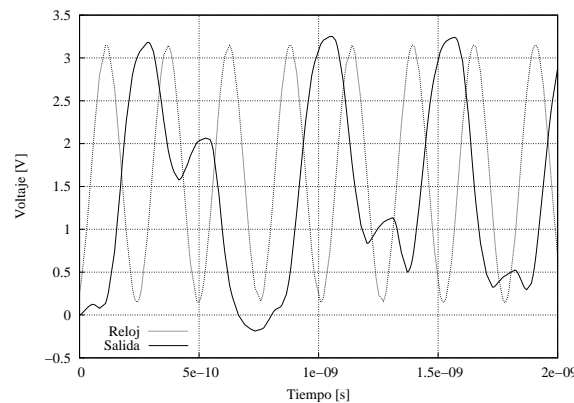


Figura 2.13: Operación incorrecta de un *flip-flop* basado en *TSPC* operando a $3.9 GHz$.

causa principal de esta falla en la operación, se le atribuye a la primera rama del circuito que es la encargada de captar la señal en el flanco de bajada del reloj.

De acuerdo a los resultados obtenidos en la tabla 2.4 y los gráficos 2.12b y 2.13, es posible afirmar que la optimización funciono adecuadamente mejorando la máxima frecuencia de operación del circuito, pasando de $1.2 GHz$ que era la frecuencia de diseño inicial planteada por [6], hasta aproximadamente $3.6 GHz$.

2.3.2. *Flip-flop* con Lógica *E-TSPC*

Bluetooth

En el caso del *flip-flop* con lógica *E-TSPC*, se simula en el dominio del tiempo. Como este bloque está directamente conectado con el *VCO* su frecuencia máxima de operación bajo el estandar *Bluetooth* es de $2.48 GHz$, por ello se simula con un reloj senoidal de frecuencia $2.48 GHz$. La amplitud del reloj es de $1.5 [V]$ con un offset de $1.65 [V]$, con los anchos presentados en la tabla 2.5. En la entrada *D* se introduce un pulso de amplitud $3 [V]$, con una duración aproximada equivalente a 5 periodos de la señal de reloj y en la salida una carga de $15,8fF$ (utilizando un promedio de las resistencias observadas en los nodos del circuito).

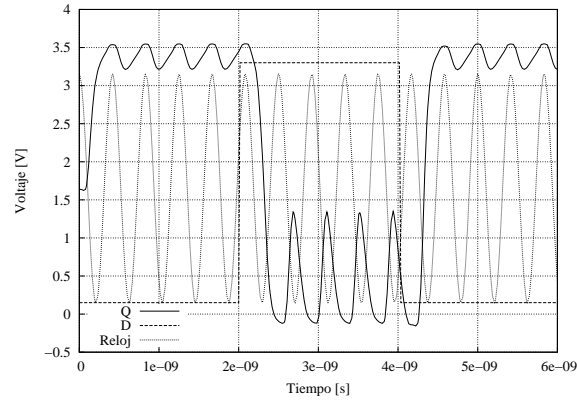


Figura 2.14: Respuesta de un *flip-flop* tipo D, implementado con lógica *E-TSPC*.

Los resultados de la simulación se pueden observar en la figura 2.14. Cabe destacar que este tipo de *flip-flop* capta los cambios de la señal de datos en los flancos de bajada de la señal de reloj (*fall edge triggered*).

Anchos de los transistores [μm] *					
M_1	M_2	M_3	M_4	M_5	M_6
2.81	1.11	1.36	8.99	1.0	2.0
Consumo de Potencia					
PG		<i>Hspice</i>	%error		
4.321 mW		4.587 mW	5.80 %		
Tiempo de demora					
PG		<i>Hspice</i>	%error		
51,81 ps		55,89 ps	7,3 %		

Tabla 2.5: Aplicación de la programación geométrica de un *flip-flop* tipo D, con lógica *E-TSPC*.

* Para el largo de todos los transistores se utilizo el L_{min} permitido por la tecnología, y es de $0.35\mu m$.

A pesar de la existencia de cierto rizado a la salida del circuito de la figura 2.14, este no deteriora la correcta operación del *flip-flop* ya que no supera los niveles lógicos fijados en las especificaciones. Esto se comprobó, mediante la implementación de un *buffer* con dos inversores en cascada, observando que el nivel de la señal a la salida no variara en dicho rizado. De esta forma se garantiza la correcta operación del *flip-flop*, en otros circuitos de los que pueda hacer parte.

Máxima Frecuencia

Similar al caso con *TSPC*, se realiza un barrido en la frecuencia del reloj con un paso de $0.1 GHz$ buscando la máxima frecuencia de operación. Se encontró, que para frecuencias superiores a $2.8 GHz$ la operación del

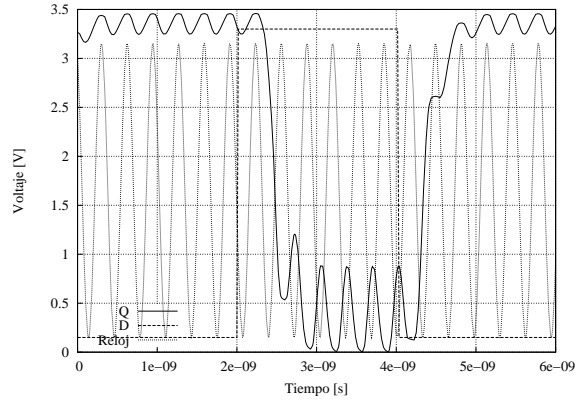


Figura 2.15: Operación incorrecta de un *flip-flop* basado en *E-TSPC* operando a 2.9 GHz .

flip-flop comenzó a verse afectada retrasándose la carga inicial, luego de haber detectado una variación en la entrada. La falla en la operación del *flip-flop*, se presenta en la gráfica 2.15.

La principal causa de la falla en la operación ocurre por la "lentitud" de respuesta del circuito para captar la señal en la fase de evaluación. Siendo la principal causa de este efecto, la cantidad de carga por unidad de tiempo que circula en la primera rama del circuito con M_1 y M_4 . Se atribuye el efecto a la primera rama ya que es la encargada de efectuar la precarga de los datos para realizar la evaluación.

De acuerdo a las predicciones obtenidas en la simulación, es posible decir que el *flip-flop* con lógica *E-TSPC* opera bajo las condiciones previstas inicialmente. El tiempo de demora, el cual era la función objetivo del programa geométrico se logró mejorar de 75.2 ps (sin optimizar) [16] a 55.89 ps (con optimización), logrando una mejora de aproximadamente %25. También es de resaltar el correcto modelado que permitió una correcta correlación de los datos en el algoritmo de optimización, con una exactitud en los modelos de aproximadamente 7.3%.

2.3.3. Análisis de Resultados

De acuerdo a los resultados obtenidos, es posible validar la programación geométrica como herramienta para el diseño de divisores de frecuencia, basados en *flip-flops* con lógica *TSPC* y *E-TSPC*. Esto se puede afirmar tomando en cuenta los resultados operando para el estándar *Bluetooth*, tal y como se definió en la sección 2.1.2.

Cabe destacar que las restricciones con las que se trabajó, fueron tomadas de acuerdo a un diseño previamente establecido, específicamente el diseño propuesto en [6]. Dichas restricciones establecen limitaciones para el espacio de diseño con el que trabaja *CVX*. Por lo tanto se desconocen los resultados que pueda traer efectuar variaciones a las restricciones. Es decir, el *flip-flop* con lógica *E-TSPC* operó correctamente hasta 2.9 GHz . Sin embargo, ¿sí modificó las restricciones del PG pueden mejorar esta especificación?.

El anterior cuestionamiento motiva utilizar la programación geométrica para encontrar hasta que especificaciones es válida una topología. Por esa razón, sería interesante y de utilidad, explorar la aplicación de la programación geométrica en este aspecto.

Capítulo 3

RESULTADOS

En un proceso de diseño es complejo fijar los límites de una determinada topología, para saber si es necesario modificarla o reemplazarla. ¿Cómo determinar la frecuencia máxima de operación, de un divisor que opera bajo *E-TSPC*? La respuesta a este interrogante se encuentra a partir del análisis de los resultados obtenidos en el capítulo anterior, donde se plantea el uso de programación geométrica como herramienta para identificar los límites de diseño.

Es claro que la programación geométrica funciona como herramienta para optimizar los *flip-flops* con lógica *TSPC* y *E-TSPC*. Sin embargo, no se conoce un límite definitivo para las especificaciones, pues el óptimo global hallado con programación geométrica está restringido al espacio de diseño fijado por las restricciones. Esto quiere decir que al variar las restricciones del PG es posible encontrar otro óptimo para el nuevo espacio de diseño. Por lo anterior, explorar las ventajas de modificar las restricciones, se convierte en una alternativa interesante para mejorar las especificaciones y conocer el verdadero potencial de una determinada topología.

Consecuentemente, el presente capítulo explora las ventajas de modificar las restricciones buscando mejorar la especificación de máxima frecuencia para el *flip-flop* con lógica *E-TSPC*. Con este fin, se presenta una metodología que permite utilizar la programación geométrica para identificar límites de diseño, junto con una herramienta que automatiza el proceso de aplicación de la programación geométrica en divisores de frecuencia. Luego se presentan los resultados de simulación para el *PLL* diseñado. Finalmente se presenta el *layout* del *PLL* diseñado, aplicando las consideraciones básicas para el diseño de circuitos integrados.

3.1. PG como Metodología de Diseño para Divisores de Frecuencia

Una vez se obtiene un primer diseño del circuito con programación geométrica, es conveniente identificar un nuevo rango en las variables que permita mejorar alguna especificación. Con este fin, se modifican las restricciones del programa geométrico implementado inicialmente, con el propósito de identificar las tendencias de la optimización con respecto al dimensionamiento de los transistores. Estas modificaciones consisten en variar el ancho de los transistores, la corriente máxima por cada rama y el área total del circuito, determinando así el comportamiento del circuito con cada nueva restricción.

Con base en la optimización desarrollada en el capítulo II, se observa que el ancho del transistor M_4 de la figura 2.4 tiende hacia el límite superior de la restricción. Mientras que los demás transistores, se mantienen en valores intermedios. Esto da un primer indicio sobre la tendencia de la PG, por lo tanto se procede a variar las restricciones que limitan el ancho máximo de los transistores. Se comienza a partir de $9 \mu m$, el cual es el valor límite para la primera implementación. A partir de allí, se continua aumentando el ancho y

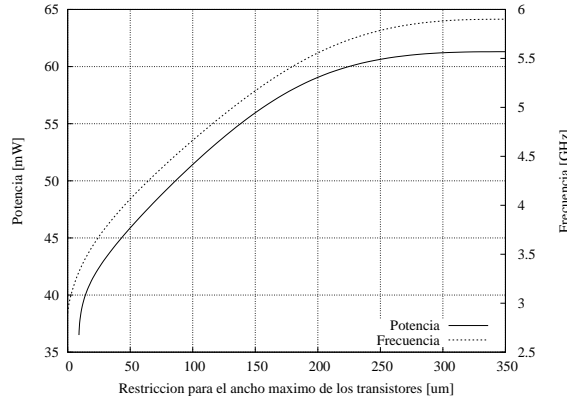


Figura 3.1: Comportamiento de la máxima frecuencia y el consumo de potencia de un *flip-flop* optimizado para diferentes anchos.

simulando los resultados en cada una de las variaciones, hasta $200\mu\text{m}$. El resultado de dichas variaciones se puede observar en la figura 3.1, el cual contiene en el eje de las abscisas el valor del ancho máximo permitido por las restricciones del programa geométrico y en el eje de las ordenadas los valores de máxima frecuencia de operación y de potencia consumida por el circuito.

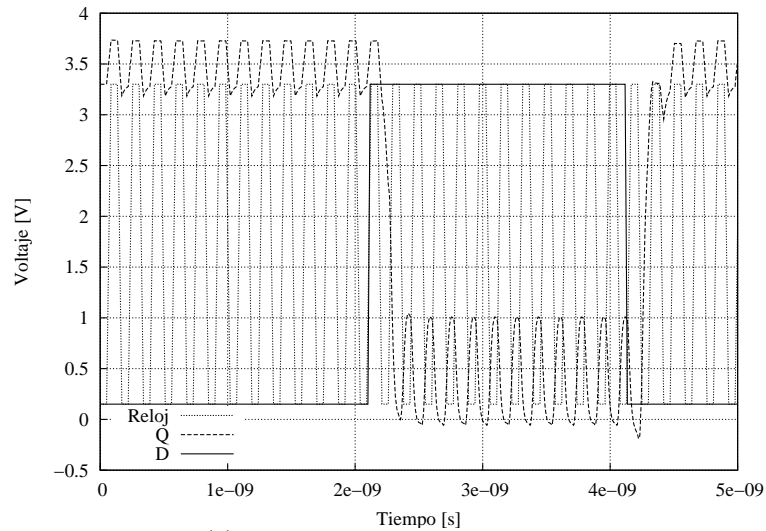
Observando la figura 3.1 se observa que tanto la potencia, como la frecuencia máxima se mantienen constantes luego de la que restricción supera los $200\mu\text{m}$. En todos los datos recopilados se observa la tendencia del transistor M_4 a mantenerse en el límite máximo del ancho que fuera impuesto para la optimización. Es decir, el ancho de este transistor siempre tiende a $9\mu\text{m}$ que es el límite superior de la restricción, mientras que el ancho de los otros dos transistores tipo PMOS (M_5 y M_6) se mantiene cercano límite inferior de la restricción, de $1\mu\text{m}$.

Según la tendencia del transistor M_4 , es posible afirmar que la primera rama de transistores requiere un menor valor de resistencia, para incrementar la corriente por la rama. Esto sucede debido a que la primera rama es un bloque n-dinámico *N-MOS like* que opera en la etapa de evaluación de la señal, cuya rapidez es crítica en la operación de todo el *flip-flop*.

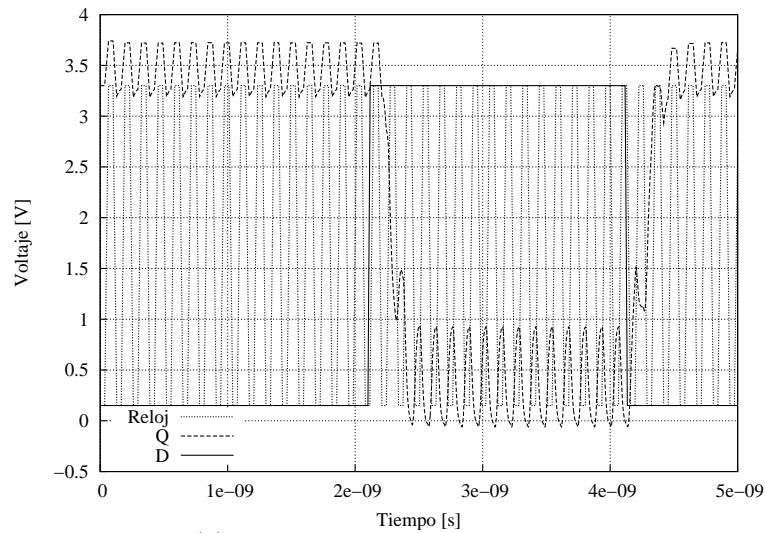
Al intentar superar los 5.9GHz de frecuencia de operación, se comienzan a presentar ciertas anomalías en la onda de salida y un deterioro en el tiempo de respuesta del *flip-flop*. Por ejemplo como se observa en la figura 3.2(b), que el *flip-flop* no alcanza completamente el nivel lógico esperado sino hasta luego de un par de flancos de bajada del reloj. Esto sucede a pesar de aumentar hasta un valor de $350\mu\text{m}$ la restricción para el ancho máximo de los transistores. Con base en lo anterior, se puede decir respecto al circuito de la figura 2.4 que la máxima frecuencia de operación que se alcanza modificando las variables de diseño, es de 5.9GHz .

Una vez alcanzado el límite, se simula para validar los resultados. El reloj consiste en una onda cuadrada de 3.15[V] de amplitud con un offset de 1.65 [V] , cada flanco de subida/bajada va de 0.15[V] a 3.3[V] en un tiempo de 35ps , las partes planas de la onda corresponden a un tiempo de 50ps , con un periodo total de 170ps , que equivale a 5.9GHz . La carga corresponde a un promedio de las resistencias observadas en los nodos del circuito, de $15,8\text{fF}$. La señal de prueba D tiene una pendiente de 17.5ps y va de 0.15[V] a 3.3[V] en 2.1ns y vuelve a bajar a los 4.1ns . Este *flip-flop* opera captando la señal de entrada D, en los flancos de bajada de la señal de reloj (*fall edge triggered*). En la figura 3.2(a) se observa la operación del *flip-flop* operando a 5.9GHz .

El proceso descrito anteriormente constituye la identificación de límites de diseño para una determinada topología. A modo general se debe hacer lo siguiente, se comienza con el diseño inicial mediante programación



(a) Operación correcta a 5.9 GHz.



(b) Operación incorrecta a 6.5 GHz.

Figura 3.2: Operación de *Flip-flop* con lógica *E-TSPC*.

geométrica como punto de partida. Basándose en el conocimiento del circuito, se modifican las restricciones del programa geométrico para mejorar la especificación deseada, en este caso particular fue la máxima frecuencia de operación. Es decir, se varía moderadamente el tamaño del espacio de diseño buscando mejoras en el comportamiento del circuito. Se ejecuta la optimización y luego se simulan los resultados para verificar el nuevo valor que tomaron las especificaciones. Finalmente se repite este procedimiento hasta encontrar una falla en el comportamiento del circuito o un límite físico en el desempeño del transistor.

La idea del procedimiento descrito anteriormente es aplicarlo dentro en un flujo de diseño tradicional. Enmarcado principalmente en la etapa de diseño, luego de haber determinado la arquitectura y las variables de diseño. Por ejemplo, si un diseñador tiene las siguientes especificaciones de entrada para un divisor de frecuencia:

- Potencia máxima a consumir: 9 mW.
- Área ocupada por el circuito: 0.009mm²
- Módulo de división: 3

- Frecuencia de entrada: 3.3 GHz

Es posible utilizar el procedimiento anteriormente descrito, para determinar si con el divisor módulo dual es posible llegar a estas especificaciones. De no lograrse este objetivo, el diseñador deberá buscar otro circuito, pero se evitó la tarea de iterar para darse cuenta que las especificaciones no eran posibles de manera simultánea.

De esta manera, la metodología presentada en esta sección puede servir como herramienta de validación para una determinada topología. Encontrando los límites en las variables para la especificación que se desee, bien sea potencia, área, frecuencia o cualquier otro parámetro que cumpla con las condiciones de la programación geométrica.

3.2. Automatización de la Metodología

Tomando como base la metodología propuesta en la sección anterior, a continuación se presenta el desarrollo de una herramienta que efectúa la automatización de la metodología de diseño. Facilitando la selección de topologías, la introducción de especificaciones, la aplicación de la programación geométrica y la simulación.

El objetivo de dicha herramienta es reducir el tiempo que implica aplicar la metodología descrita en la sección 3.1. Para esto, se utiliza una serie de *scripts* que permiten estructurar de manera ordenada los diferentes procedimientos para completar el flujo de diseño propuesto. El elemento principal para esta estructura es un archivo para *bash*¹ el cual inicialmente toma la entrada de datos, y de acuerdo a esto construye un archivo .m que será la entrada para *CVX*.

Una vez ejecutada la optimización se ejecuta una simulación en *Hspice* para verificar los datos de simulación. Posteriormente, el *script* compara los datos de la simulación para verificar si corresponden con lo solicitado en la entrada, en caso de ser negativo, envía un mensaje al usuario para solicitar nuevas especificaciones. De lo contrario, muestra el circuito con las medidas propuestas, y algunos datos y gráficos que permiten observar dicho comportamiento.

El resultado que entrega dicha herramienta es un diseño de un divisor de frecuencia, módulo 2, 3, 4 ó 5 basado en la lógica *E-TSPC*, con especificaciones como potencia, área y frecuencia de operación fijadas por el usuario. Las topologías para los divisores de frecuencia han sido previamente presentadas por [16, 17], la herramienta permite adaptar estos diseños a las especificaciones entregadas por el usuario de la herramienta.

Las ventajas de utilizar la automatización radican en la organización de proceso, el cual permite seguir un flujo de diseño para validar los resultados. Además puede reducir significativamente el tiempo de diseño, ya que mediante el uso de programación geométrica no es necesario iterar para encontrar un mejor valor porque el óptimo está garantizado.

3.3. Validación del Diseño del PLL

Con el propósito de verificar la operación en el *PLL* de los *flip-flops* diseñados aplicando programación geométrica, a continuación se presentan los resultados de las simulaciones en el dominio del tiempo, para el lazo cerrado, del *PLL* completo. Para realizar estas simulaciones se utilizaron los circuitos del comparador de fase y frecuencia, bomba de carga y del filtro propuestos en [6]. Así mismo, se utilizó un *VCO* en cuadratura suministrado con la herramienta de simulación *hspice*, el cual opera a una frecuencia de 2.6 GHz.

La simulación se ejecutó para un tiempo de 10ns, en el cual fue posible observar el enganche de la señal proveniente del lazo directo. Así mismo, se verificó la correcta operación de los bloques en el sistema

¹Bash es un terminal interprete de comandos para sistemas tipo Unix/Linux.

Tabla 3.1: Resultados de la simulación del divisor en el *PLL*.

Consumo de potencia		
Estática	Dinámica	Total
3.287 mW	5.61 mW	8.89 mW

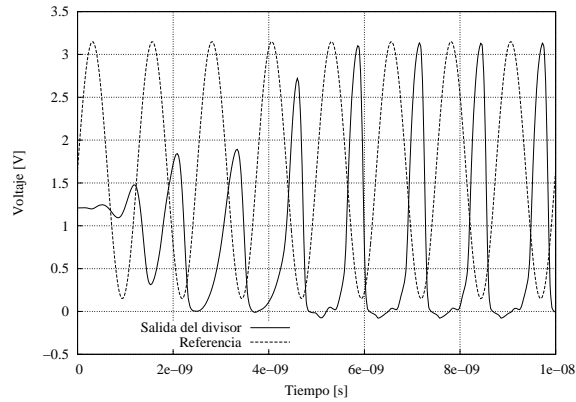


Figura 3.3: Salida del divisor de frecuencia, operando en el *PLL*.

detallando cuales eran las entradas y salidas para cada uno. Se midió la potencia consumida por el divisor de frecuencia tanto de manera dinámica como estática, el resultado se observa en la tabla 3.1. En la gráfica 3.3 se observa el comportamiento de la señal de salida del divisor, comparada con la frecuencia de referencia.

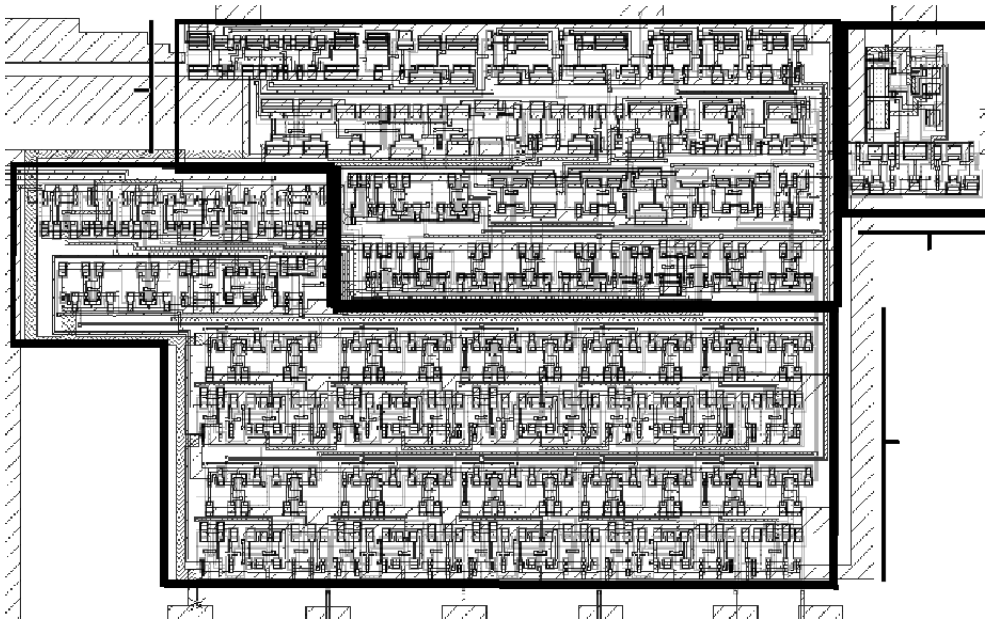


Figura 3.4: *Layout* del sintetizador de frecuencia.

3.4. Layout del PLL

En la figura 3.5 se presenta el *layout* del divisor de frecuencia, con los *flip-flop* diseñados para mejorar la operación del divisor de frecuencia. La zona resaltada de la figura, corresponde al divisor dual y al divisor módulo fijo con los *flip-flop* diseñados. Una vez elaborado el divisor se integro al conjunto de todo el sintetizador, obteniéndose como resultado el *layout* de la figura 3.4.

Para el desarrollo del *layout* se tuvieron en cuenta ciertas consideraciones que diferencian un *layout* de un circuito eléctrico. Las difusiones laterales, el desgaste por efectos del proceso (*etching*) y los efectos tridimensionales son algunos ejemplos de estas consideraciones. Así mismo, efectos parásitos de los materiales de fabricación como por ejemplo la resistencia en serie que aparece entre el dren y la fuente o las capacitancias verticales entre las diferentes capas de metal, fueron minimizados. Elementos críticos como la línea que lleva el reloj fueron blindadas con estructuras *dummy* para reducir el acople de las señales.

El *layout* se elaboró utilizando *IC flow* de la compañía *Mentor Graphics*, que es un software especializado en el diseño de circuitos integrados. Luego de elaborado el *layout* de la figura 3.5 se procedió a realizar la extracción de todas las capacitancias extrínsecas que no fueron tomadas en cuenta para el diseño eléctrico. Los resultados la extracción arrojaron valores de capacitancias inferiores a los $12fF$.

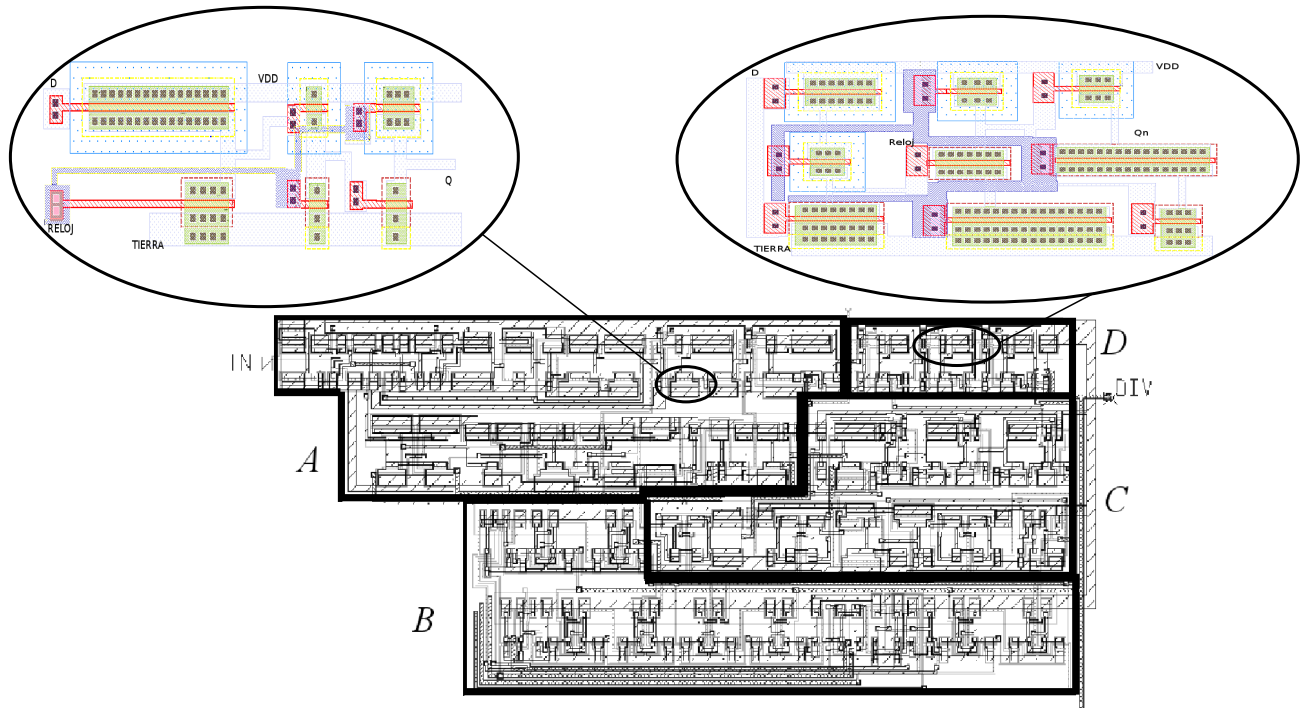


Figura 3.5: *Layout* del divisor de frecuencia.

3.5. Observaciones y Conclusiones

La fortaleza principal de este trabajo se centró principalmente en optimizar los bloques *TSPC* y *E-TSPC* encargados de realizar la división de frecuencia en el *PLL*. Dicha optimización se basó en la aplicación de la programación geométrica como herramienta de apoyo en la tarea de diseño.

Las especificaciones logradas en los *flip-flop* implementados, permiten validar la programación como una ayuda importante en el diseño de compuertas digitales. No obstante, es importante reconocer las limitaciones de los circuitos y los compromisos existentes entre las especificaciones, los cuales no se pueden obviar en ninguna instancia del diseño.

Una vez implementado el programa geométrico, la tarea de mejorar las especificaciones es más sencilla y rápida que el diseño manual. Esto se evidencia en el hecho de lograr pasar de una frecuencia máxima de 2.4 GHz a 5.9 GHz en tan solo unas horas. Sin embargo, no se pueden descuidar los demás pasos en el flujo de diseño esperando llegar a buenos resultados dependiendo únicamente del optimizador.

Utilizar la programación geométrica para identificar los límites de diseño en determinada topología, permite conocer todas las potencialidades que posee un determinado circuito. De esta manera se facilita el trabajo de los diseñadores, en la búsqueda de opciones para solucionar problemas relacionados con especificaciones.

Es importante destacar el uso de herramientas que faciliten el diseño de circuitos integrados, permitiendo obtener buenos diseños en menor tiempo con respecto a un diseño tradicional. No obstante, el diseñador no debe tomar la optimización como una caja negra y desconocer por completo la operación del circuito, ya que sigue siendo necesario el factor humano para determinar el espacio de diseño y las variables más críticas.

Una de las principales ventajas de aplicar la optimización por bloques radica en la posibilidad de utilizar los bloques optimizados en diversas aplicaciones. Aprovechando las ventajas de la optimización en varios sistemas.

Se demostró que existe un límite en la frecuencia máxima de operación de los *flip-flop* basados en *E-TSPC* de 5.9 GHz, con un ancho para el transistor M_4 de 200 μm , para la tecnología de 0.35 μm de AMS.

Contrario a lo que se puede pensar inicialmente, los resultados de la programación geométrica demostraron la tendencia de los transistores tipo PMOS de la lógica *E-TSPC* a aumentar el tamaño, para reducir el tiempo de demora asociado con la transmisión de señales al interior del circuito.

Dentro de un flujo de diseño tradicional, es bastante práctico incluir algún tipo de optimización como herramienta de decisión. Reduciendo así, la cantidad de iteraciones necesarias para encontrar el valor de alguna especificación.

3.6. Recomendaciones para Trabajos Futuros

La metodología propuesta para la identificación de límites, sea aplicada en diversas topologías de divisores como las basadas en *CML*, *SCL*, entre otras para explorar las posibilidades que tiene.

Se recomienda explorar el uso de los *flip-flop* diseñados, en sistemas distintos al *PLL*. Aprovechando las características optimizadas para mejorar otros sistemas.

Con el fin de verificar los resultados obtenidos, se recomienda la fabricación de los divisores de frecuencia implementando los *flip-flop* propuestos en este trabajo.

Aumentar la abstracción de la automatización, llevando la optimización al nivel de las reglas de construcción de topologías sería recomendable. Como el caso de la técnica *E-TSPC* en la que se puede optimizar la forma como se interconectan los bloques lógicos.

Bibliografía

- [1] J. P. Fishburn and A. E. Dunlop, “TILOS: a posynomial programming approach to transistor sizing,”
- [2] S. Boyd and L. Vanderberghe, “Convex optimization,” 2004.
- [3] Maria del Mar Hershenson and Stephen P. Boyd and Thomas H. Lee, “GPCAD: A Tool for CMOS Op-Amp Synthesis,” in *IEEE Transactions on Computer-Aided Design*, Nov. 1998.
- [4] J. Oliveros and D. Cabrera, “Aplicación de la programación geométrica en el diseño de amplificadores operacionales integrados en tecnología cmos,” 2006.
- [5] B. Razavi, *RF Microelectronics*. McGraw-Hill, 1998.
- [6] F. Cabrera, “Diseño de un sintetizador de frecuencia par aun transceiver, integrado en tecnología cmos,” 2006.
- [7] T. Lee and B. Razavi, “A stabilization technique for phase-locked loop frequency synthesizers,” *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 888–894, 2003.
- [8] K. Shu, E. Sanchez-Sinencio, J. Silva-Martinez, and S. H. K. Embabi, “[a 2.4-ghz monolithic fractional-n frequency synthesizer with robust phase-switching prescaler and loop capacitance multiplier,”
- [9] S. Pellerano, S. Levantino, C. Samori, and A. L. Lacatia, “A 13.5 mw 5-ghz frequency synthesizer with dynamic-logic frequency divider,” *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 378–383, 2004.
- [10] S. Yu and P. Kinget, “A 0.65v 2.5ghz fractional-n frequency synthesizer in 90nm cmos,” vol. SESSION 17, no. 17.2, pp. 304–305, 2007.
- [11] W. Jung, H. Choi, C. Jeong, K. Kim, W. Kim, H. Jeon, G. Koo, J. Kim, J. Seo, M. Ko, and J. Kim, “A 1.2mw 0.02mm² 2ghz current-controlled pll based on a self-biased voltage-to-current converter,” vol. SESSION 17, no. 17.5, pp. 310–311, 2007.
- [12] A. Swaminathan, K. J. Wang, and I. Galton, “A Wide-Bandwidth 2.4 GHz ISM-Band Fractional-N PLL with adaptative Phase-Noise Cancellation,” vol. SESSION 17, no. 17.1, pp. 302–303, 2007.
- [13] D. M. Colleran, C. Portmann, A. Hassibi, C. Crusius, S. S. Mohan, S. Boyd, T. Lee, and M. M. Hershenson, “Optimization of phase-locked loop circuits via geometric programming,” 2003.
- [14] J. Yuan and C. Svensson, “High-speed CMOS circuit technique,” *IEEE J. Solid-State Circuits*, February 1989.
- [15] Q. Huang and R. Rogenmoser, “Speed optimization of edge-triggered CMOS circuits for gigahertz single-phase clocks,” *IEEE J. Solid-State Circuits*, March 1996.
- [16] J.Ñ. Soares and W. V. Noije, “A 1.6-GHz dual modulus prescaler using the extended true-single-phase-clock CMOS circuit technique (*E-TSPC*),” *IEEE J. Solid-State Circuits*, January 1999.

- [17] F. de Miranda, J.Ñ. Soares, and W. V. Noije, "A 4.1 GHz dual modulus prescaler using the *E-TSPC* technique and double data throughput structures," in *Proc. IEEE ISCAS*, May 2007.
- [18] M. Grant and S. Boyd, "<http://www.stanford.edu/boyd/cvx/>, matlab software for disciplined convex programming.," September 2008.
- [19] J. Lee and B. Razavi, "A 40-GHz frequency divider in 0.18- μm CMOS technology," *IEEE J. Solid-State Circuits*, April 2004.
- [20] T. Lee *et al.*, "A CMOS frequency synthesizer with an injection-locked frequency divider for a 5GHz wireless LAN receiver," *IEEE J. Solid-State Circuits*, May 2000.
- [21] R. Mohanavelu and P. Heydari, "A novel ultra high-speed flip-flop-based frequency divider," *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 4, pp. IV-169-72 Vol.4, May 2004.
- [22] D. A. Hodges, H. G. Jackson, and R. A. Saleh, *Analysis and design of digital integrated circuits*. McGraw-Hill, 2004.
- [23] J. M. Rabaey, A. Chandrakasan, and B.Ñikolic, *Digital Integrated Circuits, a design perspective*. Prentice Hall, second ed., 2002.
- [24] E. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *Journal of Applied Physics*, January 1948.
- [25] S. P. Boyd, S.-J. Kim, D. D. Patil, and M. A. Horowitz, "Digital circuit optimization via geometric programming," *Oper. Res.*, vol. 53, no. 6, pp. 899-932, 2005.
- [26] "<http://www.mosek.com>, specialized solvers for linear programming, mixed integer programming and many types of nonlinear convex optimization problems.."
- [27] "<http://tomopt.com/tomlab>, optimization environment for fast and robust large-scale optimization in matlab.."
- [28] "<http://control.ee.ethz.ch/joloef/yalmip.php>, a matlab toolbox for rapid prototyping of optimization problems.."