

TESIS

TESIS

TESIS

TESIS

TESIS



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

CENTRO DE CIENCIAS BÁSICAS

TESIS

DESARROLLO DE UNA ARQUITECTURA DE OPTIMIZACIÓN PARA LA
MINIMIZACIÓN DE LOS EFECTOS DE CANAL CORTO EN TRANSISTORES
MOSFET SUBMICROMÉTRICOS

PRESENTA

Ing. Daniel Yair Nájera Marín
PARA OBTENER EL GRADO EN
“Maestría en Ciencias con Opción a la Computación”

TUTORES

Dr. Angel Eduardo Muñoz Zavala
Dr. Alejandro Román Loera

INTEGRANTE DEL COMITÉ TUTORAL

Dr. José Antonio Guerrero Díaz de León

Aguascalientes, Ags, 30 de junio de 2020

TESIS

TESIS

TESIS

TESIS

TESIS

M. en C. Jorge Martín Alférez Chávez
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como **Miembros del Comité Tutorial** designado del estudiante **DANIEL YAIR NÁJERA MARÍN** con ID **43197** quien realizó la tesis titulada: **DESARROLLO DE UN ARQUITECTURA DE OPTIMIZACIÓN PARA LA MINIMIZACIÓN DE LOS EFECTOS DE CANAL CORTO EN TRANSISTORES MOSFET SUBMICROMÉTRICOS**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia damos nuestro consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que nos permitimos emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla así como continuar con el procedimiento administrativo para la obtención del grado.

Ponemos lo anterior a su digna consideración y sin otro particular por el momento, le enviamos un cordial saludo.

ATENTAMENTE
“Se Lumen Proferre”
Aguascalientes, Ags., a 30 de junio de 2020.


Dr. Ángel Eduardo Muñoz Zavala
Tutor de tesis


Dr. Alejandro Román Loera
Co-Tutor de tesis

Dr. José Antonio Guerrero Díaz de León
Asesor de tesis

c.c.p.- Tesista: Ing. Daniel Yair Nájera Marín
c.c.p.- Secretaría Técnica de la Maestría en Ciencias con Opciones a la Computación, Matemáticas Aplicadas

30/6/2020

Gmail - Voto aprobatorio...



Angel Eduardo Muñoz Zavala <aemzmx@gmail.com>

Voto aprobatorio...

1 message

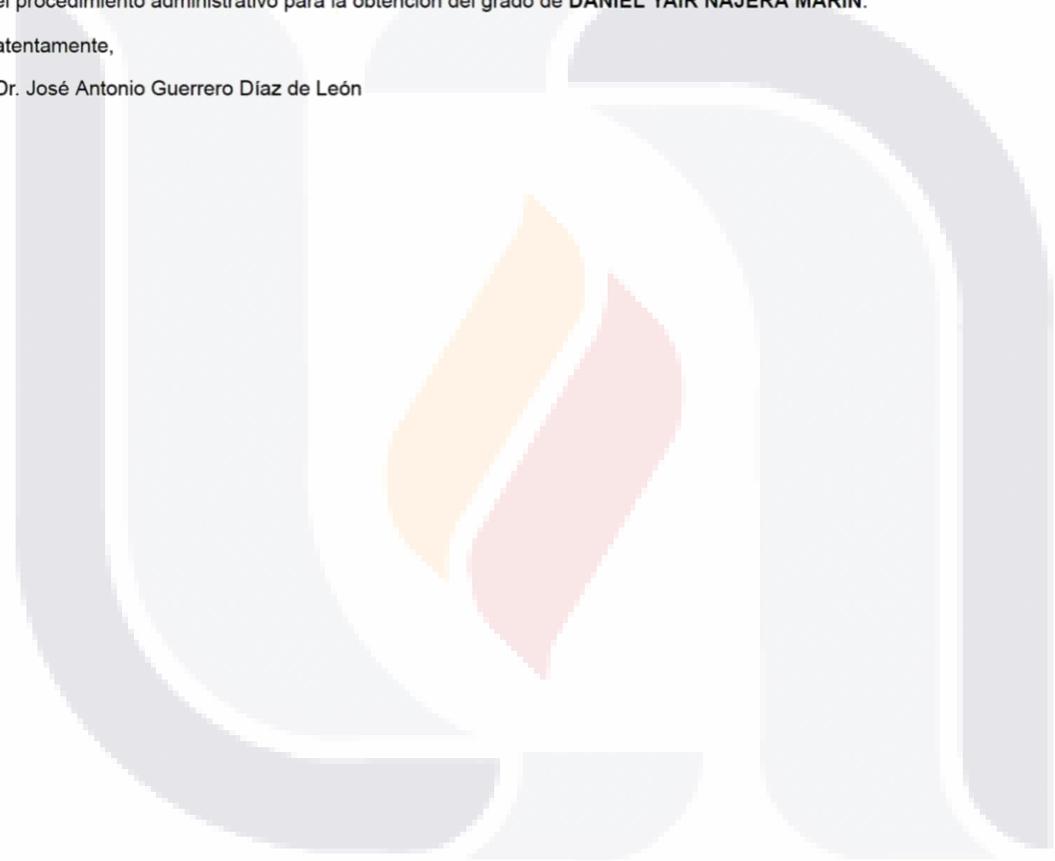
jaguerrero <jaguerrero@correo.uaa.mx>
To: Angel Eduardo Muñoz Zavala <aemzmx@gmail.com>

Tue, Jun 30, 2020 at 4:00 PM

Por medio del presente doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el voto aprobatorio para continuar con el procedimiento administrativo para la obtención del grado de **DANIEL YAIR NÁJERA MARÍN**.

atentamente,

Dr. José Antonio Guerrero Díaz de León



Fecha de dictaminación dd/mm/aaaa: 01/07/2020

NOMBRE: Daniel Yair Nájera Marín ID 43197

PROGRAMA: Maestría en Ciencias con Opción a la Computación, Matemáticas Aplicadas LGAC (del posgrado): Inteligencia Artificial

TIPO DE TRABAJO: (X) Tesis () Trabajo Práctico

TITULO: Desarrollo de una Arquitectura de Optimización para la Minimización de los Efectos de Canal Corto en Transistores MOSFET Submicrométricos

IMPACTO SOCIAL (señalar el impacto logrado): Mejora en el desempeño de los procesos CMOS incorporados en el hardware de diversos dispositivos electrónicos (médicos, móviles, telecomunicaciones, etc.); con lo cual, se verá beneficiado el usuario final en sus actividades laborales, escolares y sociales.

INDICAR SI/NO/N.A.(NO APLICA) SEGÚN CORRESPONDA:

<i>Elementos para la revisión académica del trabajo de tesis o trabajo práctico:</i>	
SI	El trabajo es congruente con las LGAC del programa de posgrado
SI	La problemática fue abordada desde un enfoque multidisciplinario
SI	Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI	Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI	Los resultados presentados en el trabajo son de gran relevancia científica, tecnológica o profesional según el área
SI	El trabajo demuestra más de una aportación original al conocimiento de su área
NO	Las aportaciones responden a los problemas prioritarios del país
NO	Generó transferencia del conocimiento o tecnológica
SI	Cumpe con la ética para la investigación (reporte de la herramienta antiplagio)
<i>El egresado cumple con lo siguiente:</i>	
SI	Cumple con lo señalado por el Reglamento General de Docencia
SI	Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias, estancia, predoctoral, etc)
SI	Cuenta con los votos aprobatorios del comité tutorial, en caso de los posgrados profesionales si tiene solo tutor podrá liberar solo el tutor
N.A.	Cuenta con la carta de satisfacción del Usuario
SI	Coincide con el título y objetivo registrado
SI	Tiene congruencia con cuerpos académicos
NO	Tiene el CVU del Conacyt actualizado
N.A.	Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)
<i>En caso de Tesis por artículos científicos publicados</i>	
N.A.	Aceptación o Publicación de los artículos según el nivel del programa
N.A.	El estudiante es el primer autor
N.A.	El autor de correspondencia es el Tutor del Núcleo Académico Básico
N.A.	En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.
N.A.	Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
N.A.	La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Sí X

No

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado:

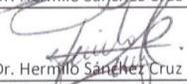
FIRMAS

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCIÓN:


Dr. Hermilio Sánchez Cruz

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO:


Dr. Hermilio Sánchez Cruz

* En caso de conflicto de intereses, firmará un revisor miembro del NAB de la LGAC correspondiente distinto al tutor o miembro del comité tutorial asignado por el Decano

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO:


Dra. Haydee Martínez Ruvalcaba

Autorizó:

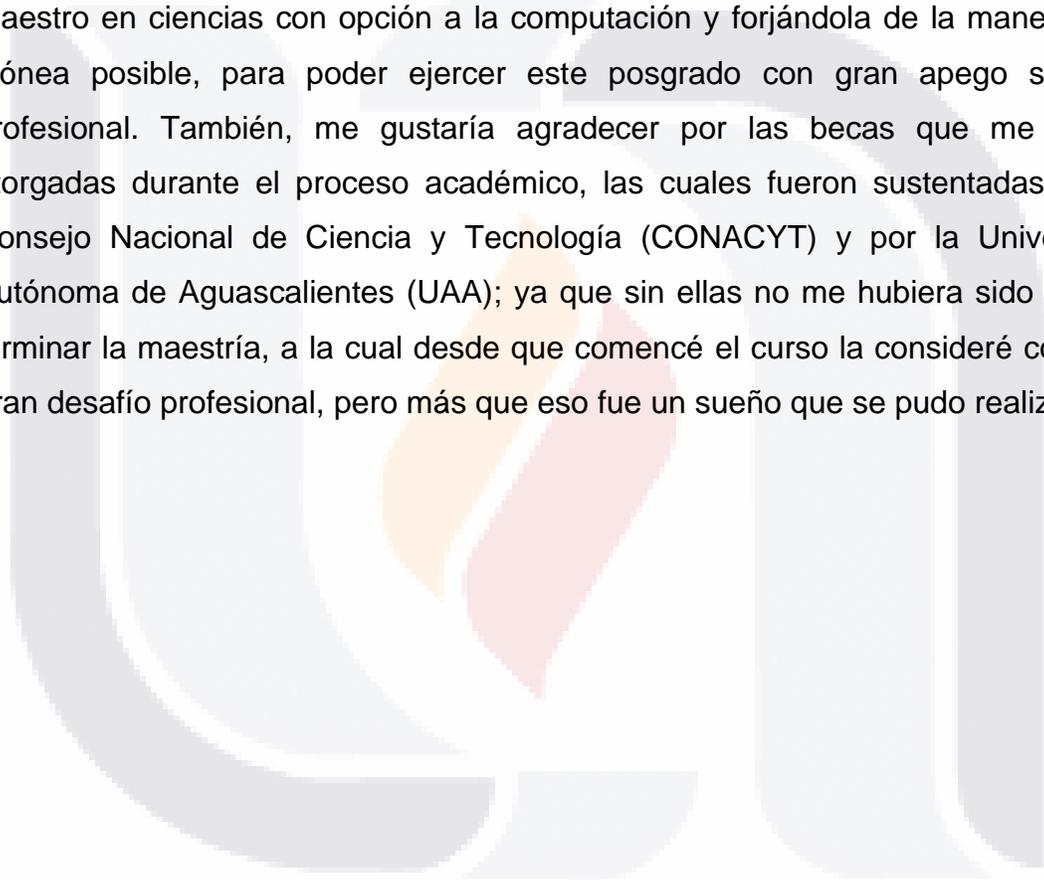
NOMBRE Y FIRMA DEL DECANO:


M. en C. Jorge Martín Alférez Chávez

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 105C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico: Cuidar la eficiencia terminal del programa de posgrado y el Art. 105F las funciones del Secretario Técnico, llevar el seguimiento de los alumnos.

Durante el proceso que llevé en la maestría en ciencias con opción a la computación, experimenté con alegrías, conviví con grandes colegas, me topé con grandes desafíos tanto académicos como personales. Durante estas experiencias personales tuve el apoyo de grandes personas. Debido a esto, estoy profundamente agradecido con mi madre, hermanos y maestros, por haber formado parte en todas estas experiencias. Gracias a su gran apoyo pude superar ciertos obstáculos que surgieron durante el curso de la maestría, contribuyendo a mi formación como maestro en ciencias con opción a la computación y forjándola de la manera más idónea posible, para poder ejercer este posgrado con gran apego social y profesional. También, me gustaría agradecer por las becas que me fueron otorgadas durante el proceso académico, las cuales fueron sustentadas por el Consejo Nacional de Ciencia y Tecnología (CONACYT) y por la Universidad Autónoma de Aguascalientes (UAA); ya que sin ellas no me hubiera sido posible terminar la maestría, a la cual desde que comencé el curso la consideré como un gran desafío profesional, pero más que eso fue un sueño que se pudo realizar.



ÍNDICE

Resumen..... 7

Abstract..... 8

Introducción..... 9

 Hipótesis 13

 Objetivo general..... 14

 Objetivos específicos 14

 Contribución..... 14

Marco teórico 16

 Efectos de canal corto en el *MOSFET*..... 22

 A. Análisis de efectos de canal corto. 23

 I. Corrientes de fuga. 23

 II. Resistencia de salida, transconductancia de salida y capacitancias
 parásitas..... 26

 C. Alternativas de material en los MOSFETs..... 30

 D. Modelos de tecnologías modernas de los MOSFETs..... 33

 E. Algoritmos de optimización implementados en el MOSFET 36

Algoritmos evolutivos para optimización multiobjetivo 48

 Definiciones 48

 Optimización mono objetivo 49

 El problema de optimización multiobjetivo 50

 Variables de decisión 50

 Restricciones 50

 Funciones objetivo 51

Problema de optimización multiobjetivo	51
Optimalidad de Pareto	52
Conceptos básicos de los algoritmos evolutivos	54
Técnicas de clasificación	56
Enfoques del algoritmo evolutivo MOP	57
Diseño de operadores y objetivos genéricos de MOEA	58
Algoritmo genérico MOEA.....	59
Algoritmo genético multi objetivo	60
Fitness escalado linealmente.....	61
Tamaño de nicho	62
Fitness compartido.....	63
Selección	64
Cruzamiento y mutación	64
Dominancia de Pareto	65
Elección de los parámetros del MOGA	66
Manejo de restricciones	69
Metodología	71
Arquitectura de lectura de datos	71
Funciones objetivo del problema de optimización	77
Restricciones del problema de optimización.....	83
Error de truncamiento para los puntos discretos	95
Complejidad computacional de las funciones objetivo y restricciones.....	100
Justificación de la metodología de optimización a utilizar	103
Arquitectura de software del problema de optimización.....	107
Arquitectura del simulador ngspice	108

Arquitectura multiobjetivo para el MOGA..... 110
Arquitectura de comunicación..... 111
Unión de las arquitecturas para el flujo de datos entre ngspice y MOGA 111
Resultados 114
Conclusiones..... 130
Referencias..... 132



ÍNDICE DE TABLAS

Tabla 1 Funciones de prueba numérica para *MOP* (con restricciones laterales).. 68

Tabla 2 Pruebas de convergencia con respecto al tiempo de las tecnologías 180nm y 90nm. 115

Tabla 3 Conjunto de restricciones y parámetros del *MOGA* para obtener el PF_{true} y P_{true} de la tecnología 180nm. 118

Tabla 4 Conjunto de restricciones y parámetros del *MOGA* para obtener el PF_{true} y P_{true} de la tecnología 90nm. 118

Tabla 5 Definición de *Cap* para obtener PF_{true} y P_{true} de la tecnología 180nm. ... 120

Tabla 6 Definición de *EIds* para obtener PF_{true} y P_{true} de la tecnología 180nm... 120

Tabla 7 Definición de *ERds* para obtener PF_{true} y P_{true} de la tecnología 180nm.. 120

Tabla 8 Definición de *limiteIds* para obtener PF_{true} y P_{true} de la tecnología 180nm. 120

Tabla 9 Definición de *Ifug* para obtener PF_{true} y P_{true} de la tecnología 180nm... 121

Tabla 10 Definición de *Cap* para obtener PF_{true} y P_{true} de la tecnología 90nm. ... 122

Tabla 11 Definición de *EIds* para obtener PF_{true} y P_{true} de la tecnología 90nm... 122

Tabla 12 Definición de *ERds* para obtener PF_{true} y P_{true} de la tecnología 90nm.. 122

Tabla 13 Definición de *Ifug* para obtener PF_{true} y P_{true} de la tecnología 90nm... 123

Tabla 14 Resultados de simulación con LTspice XVII para obtener *porarea* y *porgan*. 126

ÍNDICE DE FIGURAS

Figura 1. Variación de la corriente I_{bs} con respecto a L , V_{ds} y V_{gs} 13

Figura 2. Estructura del *MOSFET*: (a) acumulación, (b) agotamiento, e (c) inversión (Neil H. E. Weste, 2011). 16

Figura 3. Transistor *nMOS*: (a) apagado, (b) y (c) lineal y (d) saturación (Neil H. E. Weste, 2011). 18

Figura 4 Variación de la impedancia de salida R_{ds} con respecto a L , V_{ds} y V_{gs} 23

Figura 5 Frente de Pareto(PF_{true}) y soluciones de Pareto(P_{true}) de un conjunto de Funciones de prueba con restricciones para *MOP*. 67

Figura 6 Circuito con un transistor *nMOS*. 74

Figura 7 Variación de la corriente I_{ds} con respecto a L , V_{ds} y V_{gs} 75

Figura 8 Variación de la transconductancia de salida g_m con respecto a L , V_{ds} y V_{gs} 76

Figura 9 Variación de la capacitancia C_{bg} con respecto a L , V_{ds} y V_{gs} 76

Figura 10 Funciones objetivo de ganancia de I_{ds} , g_m y R_{ds} 90

Figura 11 Error relativo de I_{ds} , g_m y R_{ds} 93

Figura 12 Formato general y nomenclatura para el error de truncamiento en varios segmentos. 98

Figura 13 Error de truncamiento de las variables dependientes. 101

Figura 14 Diagrama de ngspice/XSPICE (Holger Vogt, 2018). 109

Figura 15 Arquitectura de alto nivel (Arcelli, Cortellessa, Emidio, & Pompeo, 2018). 111

Figura 16 Modelo de comunicación basado en el código de comunicación de NGSPICE (Holger Vogt, 2018). 112

Figura 17 PF_{true} y P_{true} de la tecnología 180nm para determinar g_{max} y m 116

Figura 18 PF_{true} y P_{true} de la tecnología 90nm para determinar g_{max} y m 117

Figura 19 PF_{true} y P_{true} de la tecnología 180nm con los valores de la Tabla 3. ... 118

Figura 20 PF_{true} y P_{true} de la tecnología 90nm con los valores de la Tabla 4. 119

Figura 21 PF_{true} y P_{true} de la tecnología 180nm para cada ecuación (13), (15), (17), (26) y (27). 121

Figura 22 $P_{F_{true}}$ y P_{true} de la tecnología 90nm para cada ecuación (13), (15), (26) y (27)..... 123

Figura 23 Amplificador de dos etapas..... 124

Figura 24 Amplificador *Rail to Rail*..... 125



Resumen

Los efectos de canal corto degradan el funcionamiento de los *MOSFETs*, lo cual influye en el desempeño de las variables intrínsecas de este transistor, ocasionando incrementos en algunas variables que deben minimizarse; en consecuencia dichas variables entran en conflicto. Es de particular interés en este trabajo, obtener la mínima área, el mínimo consumo de potencia, la máxima ganancia de voltaje y el máximo ancho de banda del *MOSFET* con aplicaciones en circuitos analógicos, estas características del transistor se ven afectadas por los efectos de canal corto. Se configuraron dos circuitos analógicos basados en tecnología *CMOS* (*MOSFETs* tipo N y P), observándose un comportamiento inversamente proporcional en algunas respuestas de estos circuitos analógicos. Por esta razón, se obtuvo un conjunto de funciones objetivo y restricciones que tomaron en cuenta los principales efectos de canal corto del *MOSFET*. Este conjunto de funciones objetivo y restricciones se implementaron en una arquitectura de optimización cuyos núcleos principales son el algoritmo genético multi objetivo (*MOGA*), y el simulador Ngspice para obtener un rango de las longitudes de canal del *MOSFET* donde exista una mínima área, un mínimo consumo de potencia, etcétera del transistor. Al implementar la arquitectura de optimización en un *MOSFET Bulk* de tecnología 180nm (BSIM3) y 90nm (BSIM4), se obtuvieron diferentes conjuntos de soluciones (P_{true}) y frentes de Pareto (PF_{true}). Después de obtener los P_{true} y PF_{true} , se optó por probarlos en los amplificadores operacionales (*op-amp*, por su término en inglés) dos etapas y *rail to rail*. Al comparar los P_{true} y PF_{true} con los resultados de simulación, se observó que el conjunto de funciones objetivo y restricciones se definieron de manera adecuada. Además, se realizaron análisis teórico-práctico para reducir la complejidad computacional de la arquitectura de optimización, reduciendo el tiempo computacional de manera significativa.

Abstract

The short channel effects deteriorate the MOSFETs operation, which influences the performance of the intrinsic variables of these transistors, causing an increase in some variables that must be minimized; consequently, these variables come into conflict each other. It is of particular interest in this work, to obtain the minimum area, the minimum power consumption, the maximum voltage gain and the maximum bandwidth of the MOSFET with applications in analog circuits, these characteristics of the transistor are affected by the short channel effects. Two analog configurations were implemented using CMOS (N and P MOSFET) transistors, observing a proportional inversely behavior in some responses of these analog circuits. For this reason, a set of objective functions and constraints was obtained that considered the main short channel effects of MOSFET. This set of objective functions and constraints were implemented in an optimization architecture whose main cores are the Multi Objective Genetic Algorithm (MOGA) and the Ngspice simulator, to obtain a range of MOSFET gate lengths where there is a minimum area, a minimum power consumption, etc. of transistor. When implementing the optimization architecture in a MOSFET Bulk of 180nm (BSIM3) and 90nm (BSIM4) technology, different sets of solutions (P_{true}) and Pareto fronts (PF_{true}) were obtained. After that, P_{true} and PF_{true} were chosen to be tested in the two stages opamp and rail to rail opamp. When comparing the P_{true} and PF_{true} with the simulation results, it was observed that the set of objective functions and constraints were properly defined. Furthermore, theoretical-practical analysis was made to reduce the computational complexity of architecture optimization, reducing significantly computational time.

Introducción

El gran avance en la tecnología desde los años 50 ha contribuido en la resolución de las tareas que cotidianamente llevan a cabo las personas, por ejemplo, la rapidez al resolver un problema en la escuela, en la comodidad de sus hogares, en la eficiencia en el trabajo, etcétera. Esto se ha debido en parte a la investigación científica y tecnológica de los transistores que se utilizan en la mayoría de los dispositivos electrónicos que usamos a diario, como, computadoras, televisiones, dispositivos programables, etcétera. Los primeros circuitos integrados estaban contruidos con pocos transistores, con el paso del tiempo, la tasa de crecimiento desde los años 1950 ha sido enorme encontrando microprocesadores actuales con más de 2 billones de transistores en su interior. La gran composición de los transistores por circuito integrado se ha podido realizar por la miniaturización de los transistores y los procesos de manufactura (Neil H. E. Weste, 2011).

En 1947, John Bardeen y Walter Brattain construyeron el primer transistor de contacto de punto en los laboratorios Bell, en poco tiempo, el transistor *BJT* fue construido. Este transistor era más confiable que el transistor de contacto de punto en cuanto a ruido y eficiencia de la potencia, pero consumía potencia aún en reposo. En 1935, Oskar Heil propuso un diseño de un transistor *MOSFET*, pero no fue construido debido a los problemas de materiales en esa época. En la década de los 60s, los transistores efecto campo de metal-oxido-semiconductor (*MOSFETs*) empezaron a construirse. A diferencia de los *BJT*, estos transistores disipaban una potencia de casi cero en reposo. Hay dos tipos de transistores *MOSFETs*: *nMOS* y *pMOS* los cuales están conformados por un sustrato de silicio tipo-n y un sustrato de silicio tipo-p, respectivamente (Neil H. E. Weste, 2011).

En 1963, Frank Wanlass diseñó y construyó las primeras compuertas lógicas usando transistores *nMOS* y *pMOS*, obteniendo el nombre metal-oxido-semiconductor complementario o *CMOS*. Los primeros circuitos integrados utilizaban transistores *pMOS*, los cuales tenían los inconvenientes de pobre

desempeño, rendimiento y fiabilidad. Los transistores *nMOS* se volvieron comunes en 1970, aunque el proceso de construcción de los circuitos integrados con tecnología *nMOS* fue menos costosa que la tecnología *CMOS*, los transistores *nMOS* todavía consumían potencia estando en reposo. Este problema se acrecentó cuando un circuito integrado contenía cientos de miles de transistores. En consecuencia a las dificultades de las tecnologías *nMOS* y *BJT*, la tecnología *CMOS* ha sido utilizada más ampliamente (Neil H. E. Weste, 2011).

Los transistores pueden comportarse como dispositivos conmutadores (entre otras aplicaciones), los cuales tienen una terminal de control que dependiendo del voltaje que se le aplique, otras dos terminales conmutan para dejar pasar o no corriente eléctrica. Los transistores están contruidos de uniones múltiples de materiales semiconductores, y por medio de métodos de dopado de los materiales se les puede dar polaridad P o N. Además, los transistores *MOSFETs* están conformados por materiales aislantes y conductores. Así mismo, los transistores se pueden utilizar en aplicaciones digitales (conmutador) y analógicas. Por ejemplo, en conjunto con dispositivos eléctricos pueden usarse para amplificar señales eléctricas.

La gran necesidad de una alta integración de sistemas en un circuito integrado conllevó a la aparición de tecnologías modernas de los *MOSFETs* con longitud de canal menores a $.2\mu\text{m}$ por el año 2000 (Razavi, 2001); a consecuencia de esto, cuando los transistores tienden a pequeñas dimensiones existen efectos que degradan su correcto funcionamiento. Esta disminución del tamaño del canal provoco la aparición de efectos de mayor orden que minimizan la eficiencia de los transistores. Esto ha afectado principalmente a la ciencia y tecnología, ya que el desempeño de los microprocesadores actuales no es suficiente para llevar a cabo la tarea de algunos softwares con alta complejidad computacional. Además, los circuitos analógicos como, los Amplificadores operacionales (*opamps*) sufren de corrientes de fuga; en consecuencia, disminuye la ganancia de voltaje afectando el ancho de banda (Weixun, Kolm, & Zimmermann, 2008). Otro problema que surge, es la necesidad de modelos de simulación y análisis más complejos para

proporcionar la suficiente precisión en los diseños de sistemas basados en tecnología *CMOS* (Razavi, 2001). A estos impactos de mayor orden se les suele llamar efectos de canal corto. Una de las líneas de investigación en el diseño de circuitos integrados es disminuir estos efectos de canal corto para obtener el mayor desempeño posible de estos procesos *CMOS*.

La variación de las dimensiones de los transistores *MOSFETs* afectan la transconductancia del transistor y la conductancia de salida, y por consecuencia la velocidad de operación del dispositivo (Baek et al., 2013) (Chen, Mohamed, Jo, Ravaioli, & Xu, 2013); también, las capacitancias parásitas de un transistor *MOSFET* disminuyen la frecuencia de operación del dispositivo (Lam & Chan, 2013). Además, el incremento de las fugas de corriente del dispositivo aumenta el consumo de potencia (Mukhopadhyay et al., 2018). Como, se mencionó anteriormente estas características que degradan el funcionamiento del dispositivo se les suele llamar efectos de canal corto. Los efectos de canal corto son:

- a) La variación del voltaje de disparo.
- b) La degradación de la movilidad de los portadores.
- c) La velocidad de saturación de los portadores.
- d) Las cargas excitadas.
- e) La variación de la impedancia de salida con el voltaje Drenaje-Fuente
- f) La modulación de la longitud de canal.

Se han desarrollado modelos matemáticos y computacionales que incluyen los efectos de canal corto. Sin embargo, estos modelos dependen del tipo de tecnología *CMOS* que se trate. De acuerdo con lo anterior, un modelo eficiente para una tecnología, no lo será para otra tecnología (por ejemplo, tecnología de $0.5\mu\text{m}$ a $0.35\mu\text{m}$ a $0.25\mu\text{m}$) (Razavi, 2001). Asimismo, se ha llevado a cabo una gran investigación para reducir los efectos de canal corto, por ejemplo, transistor con doble Compuerta (Srivastava, 2015) (Frank, Laux, & Fischetti, 1992), transistores de sustrato ultradelgado y óxido incrustado completamente agotado (UTBB FD-SOI, por sus siglas en inglés) (Cui et al., 2018), correlación de variables microscópicas

TESIS TESIS TESIS TESIS TESIS

con macroscópicas del *MOSFET* (Mattausch et al., 2009), etcétera; esta investigación implica la aparición de más modelos para las nuevas tecnologías (Jaafar, Aouaj, Bouziane, & Iñiguez, 2018) (Huang & Li, 2015) (Bera, Mondal, & Biswas, 2017). Por lo tanto, esto representa un impedimento para cualquier estrategia implementada para minimizar los efectos de canal corto.

Todo circuito integrado analógico tiene asociados cuatro parámetros importantes, las cuales son tamaño del circuito integrado, consumo de potencia, ganancia de V/I y ancho de banda. Un buen circuito integrado analógico es aquel que sus dimensiones tienden a cero, el consumo de potencia es pequeño, la ganancia tiende a grandes valores, y el ancho de banda tiende a infinito (Cantürk & Kahraman, 2015) (Jafari, Zekri, Sadri, & Mallahzade, 2010). Sin embargo, los efectos de canal corto reducen la posibilidad de obtener un buen circuito integrado analógico. La Figura 1 muestra uno de los efectos de canal corto; el cual es la variación de la corriente de fuga del Substrato a Fuente (I_{bs}) con respecto a el voltaje de Drenaje a Fuente (V_{ds}), el voltaje de Compuerta a Fuente (V_{gs}), la longitud del canal del transistor (L), manteniendo la relación W/L constante. Se puede observar que I_{bs} aumenta con respecto a V_{ds} , también con respecto a la longitud L . Los datos de I_{bs} se obtuvieron de un *MOSFET* de tipo *Bulk* de tecnología 180nm utilizando el software de simulación Ngspice.

Para un buen circuito analógico es importante reducir los efectos de canal corto. Debido a esto, se han diseñado o implementado un gran número de algoritmos que minimizan los efectos de canal corto de los circuitos analógicos, como la programación geométrica para la optimización del diseño de amplificadores de ruido bajo (*LNA*) con transistores *CMOS* (Cheung & Wong, 2006), el algoritmo genético híbrido para la optimización de un filtro analógico (Liu & He, 2009), el algoritmo de optimización numérica para el diseño de un oscilador de control de voltaje *LC CMOS* diferencial (Abdel-Bary & Khalil, 2003), el algoritmo búsqueda armónica basada en oposición para el diseño óptimo de los circuitos amplificadores diferenciales (Maji, Jaiswal, Kar, Mandal, & Ghoshal, 2015), etcétera. En todas estas metodologías de

optimización primero se diseñó el circuito analógico, luego se analizó para obtener sus parámetros, tales como ruido térmico, potencia de disipación, resistencias parasitas, frecuencia de corte, longitudes del canal, ganancia de voltaje, la relación de rechazo en modo común, el *Slew Rate*, el voltaje de disparo, la transconductancia del *NMOS* y *PMOS*, etcétera; es de resaltar que cada circuito analógico tiene sus propias características y que no necesariamente son todas las anteriores. Estos parámetros han sido implementadas como funciones objetivo y/o restricciones en diversos algoritmos de optimización, tomando en cuenta que cada circuito analógico cuenta con sus propias características; y, cada algoritmo conlleva una adaptación distinta de una aplicación a otra (Kotti et al., 2011) (Jin & Hoe, 2012) (Aggarwal & O'Reilly, 2007). Una problemática, es diseñar un algoritmo de optimización que tome en cuenta los efectos de canal corto para optimizar cualquier circuito analógico. Además, otra problemática es la adaptación del algoritmo a los modelos de tecnología *CMOS*.

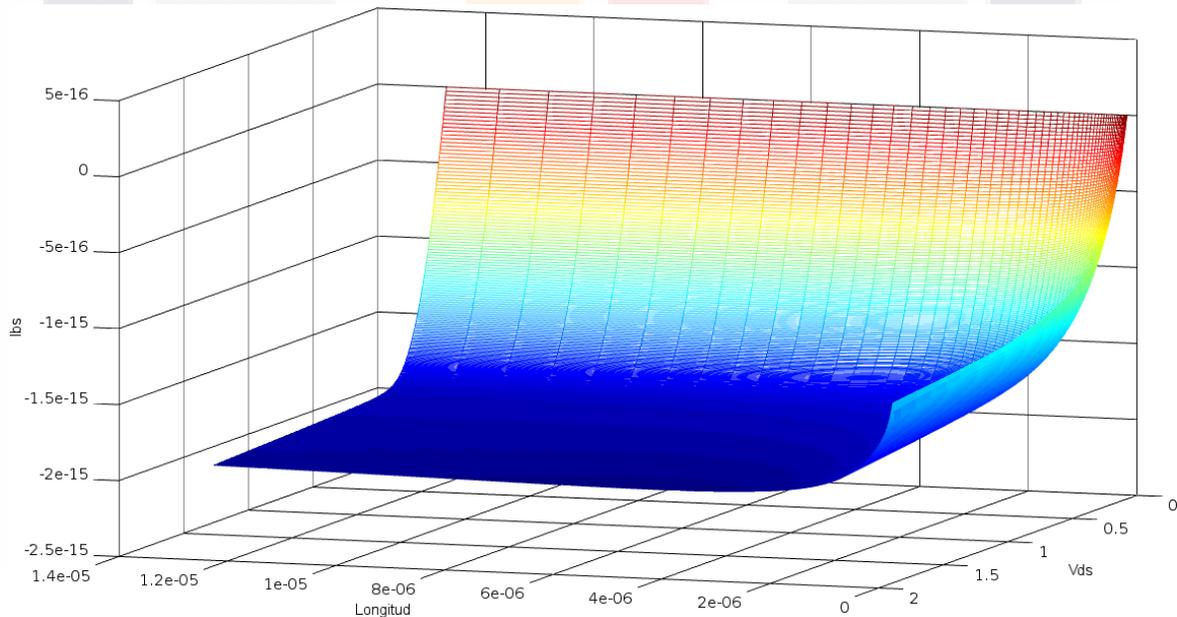


Figura 1. Variación de la corriente I_{ds} con respecto a L , V_{ds} y V_{gs} .

Hipótesis

Al tomar en cuenta los principales efectos de canal corto que degradan el desempeño del transistor *MOSFET* con aplicaciones en circuitos integrados analógicos, se podrán optimizar las características del *MOSFET*, las cuales son la

mínima área, el mínimo consumo de potencia, la máxima ganancia de voltaje y el máximo ancho de banda; con respecto a las arquitecturas implementadas con tecnología *CMOS* moderna.

Objetivo general

Implementar una arquitectura que tome en cuenta los efectos de canal corto aplicado a tecnologías *MOSFET*, el cual optimice las características del transistor (el área, el consumo de potencia, la ganancia de voltaje y el ancho de banda) con aplicaciones en circuitos integrados analógicos.

Objetivos específicos

1. Desarrollar un análisis de las funciones de los efectos de canal corto para obtener el tipo de comportamiento que tienen al tratar de resolver las funciones objetivo. Con lo cual, se podrá tomar la decisión correcta del tipo de algoritmo que se utilizará ya sea matemático, heurístico o metaheurístico.
2. La complejidad computacional de la arquitectura tiene que ser la mínima posible. Con lo cual, se llevará a cabo un enfoque teórico para analizar el pseudocódigo antes de implementarlo, y con esto obtener las funciones que nos garanticen una complejidad computacional eficiente.
3. Desarrollar un análisis sobre los cambios que hay en los modelos de cada tecnología para ver el comportamiento de sus efectos de canal corto; así como, para determinar si aparecen nuevas funciones de degradación del funcionamiento de los *MOSFETs*. Con lo cual, al tomar en cuenta este análisis para el desarrollo del algoritmo de optimización el algoritmo podrá ser adaptado a cualquier modelo de tecnología, tomando en cuenta que hay modelos más precisos que otros.

Contribución

La contribución de esta tesis será desarrollar e implementar una arquitectura que optimice las características de las tecnologías *CMOS*, los resultados de esta metodología serán utilizados en la simulación de un circuito analógico y así,

determinar si su desempeño es mejor que el de aquellas implementadas con una tecnología *CMOS* moderna, tomando en cuenta los efectos de canal corto. La arquitectura empleada en la optimización será de una complejidad computacional eficiente, y podrá ser utilizada con cualquier modelo de tecnologías *CMOS* modernas. La optimización del *MOSFET* tendrá un impacto en la industria electrónica de manera significativa; principalmente el usuario final, que utilice los productos desarrollados por estas empresas, se verá beneficiado en gran manera.



Marco teórico

El transistor de tipo MOS es un dispositivo de portadores mayoritarios en el cual la corriente en el canal de conducción entre la Fuente y el Drenaje es controlada por el voltaje de Compuerta. En un transistor *nMOS*, los portadores mayoritarios son electrones, y en un transistor *pMOS*, los portadores mayoritarios son los huecos. El comportamiento de un *MOSFET* puede ser entendido si primero se examina una estructura del *MOSFET* aislada con Compuerta y Substrato, pero sin Fuente y Drenaje. La Figura 2 muestra la estructura de un *MOSFET* simple. La capa superior de la estructura está denominada como la compuerta de transistor y, está construida con un material conductor, la capa intermedia es una muy delgada película aislante de dióxido de silicio (SiO_2) y la capa inferior es un Substrato de silicio dopado (Neil H. E. Weste, 2011).

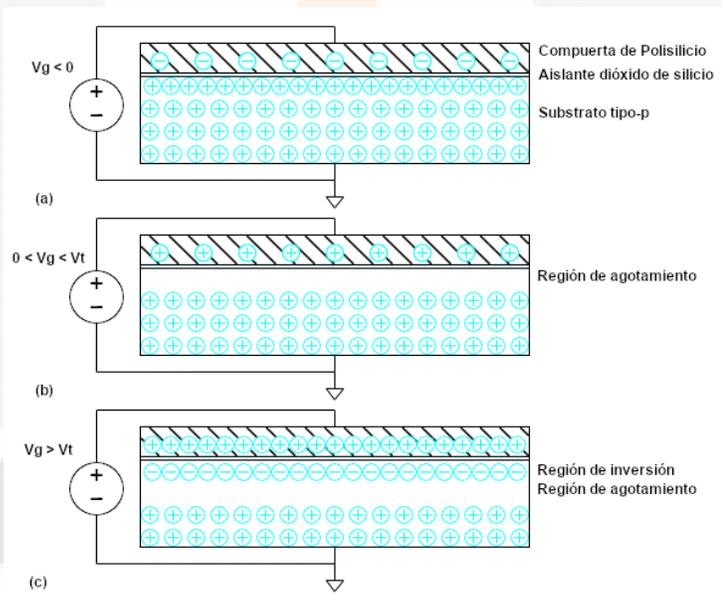


Figura 2. Estructura del MOSFET: (a) acumulación, (b) agotamiento, e (c) inversión (Neil H. E. Weste, 2011).

En la Figura 2(a), un voltaje negativo es aplicado a la Compuerta, entonces hay cargas negativas en la Compuerta. Los huecos cargados positivamente son atraídos a la región debajo de la Compuerta. Esto es llamado modo de acumulación. En la Figura 2(b), un pequeño voltaje positivo es aplicado a la Compuerta, resultando en algunas cargas positivas en la Compuerta; los huecos en el Substrato son repelidos de la región directamente debajo de la Compuerta, resultando en una

región de agotamiento formada debajo de la Compuerta. En la Figura 2(c), un voltaje positivo más grande que excede el voltaje de disparo crítico (V_t) es aplicado a la terminal de Compuerta, atrayendo más cargas positivas a esta terminal; los huecos son repelidos más lejos y algunos electrones libres en el Substrato son atraídos a una región debajo de la Compuerta. Esta capa conductora de electrones en el Substrato tipo-p es llamada la región de inversión (Neil H. E. Weste, 2011).

La Figura 3 muestra un transistor *nMOS*, el transistor consiste en las terminales Fuente (*s*) y Drenaje (*d*) conformadas de material tipo-n. En la Figura 3(a) el voltaje de Compuerta (*g*) a Fuente (V_{gs}) es menor que el voltaje de disparo (V_t). La *s* y el *d* tienen electrones libres; el Substrato tiene huecos libres, pero no electrones libres. Se dice que el transistor está apagado, y esta manera de operación es llamado modo apagado. En la Figura 3(b), el voltaje de Compuerta (V_g) es mayor que V_t . Ahora, una región de inversión de electrones (portadores mayoritarios) llamado el canal conecta a la Fuente y el Drenaje, creando un camino de conducción y por consecuencia, el transistor se enciende. El número de portadores minoritarios y la conductividad incrementa con el incremento de V_g . La diferencia de potencial entre Drenaje y Fuente es V_{ds} y, esta es igual a $V_{gs} - V_{gd}$. Si $V_{ds} = 0$ (ejemplo, $V_{gs} = V_{gd}$), no se establece un campo eléctrico lateral que empuje los portadores del Drenaje a la Fuente y por tanto la corriente entre estas dos terminales es 0. Cuando un pequeño voltaje positivo V_{ds} es aplicado al Drenaje (Figura 3(c)), la corriente de Drenaje a Fuente (I_{ds}) fluye a través del canal de Drenaje a Fuente. Este modo de operación es llamado lineal, resistivo, triodo, no saturado, o insaturado; la corriente incrementa con ambos voltajes de Drenaje y de Compuerta. Si V_{ds} se vuelve suficientemente grande tal que $V_{ds} > V_{gs} - V_t$, el canal ya no está invertido cerca del Drenaje y se vuelve una región de estrangulamiento, este voltaje es denominado voltaje de saturación ($VDSSAT$) (Figura 3(d)). Sin embargo, la conducción es aún producida por el amontonamiento de electrones bajo la influencia del voltaje de Drenaje positivo; cuando, los electrones alcanzan el final del canal, estos son introducidos en la región de agotamiento cerca del Drenaje y acelerados hacia el Drenaje. Por encima de $VDSSAT$ la corriente I_{ds} es controlada únicamente por el

voltaje de Compuerta y deja de ser influenciada por el Drenaje. Este modo es llamado saturación (Neil H. E. Weste, 2011).

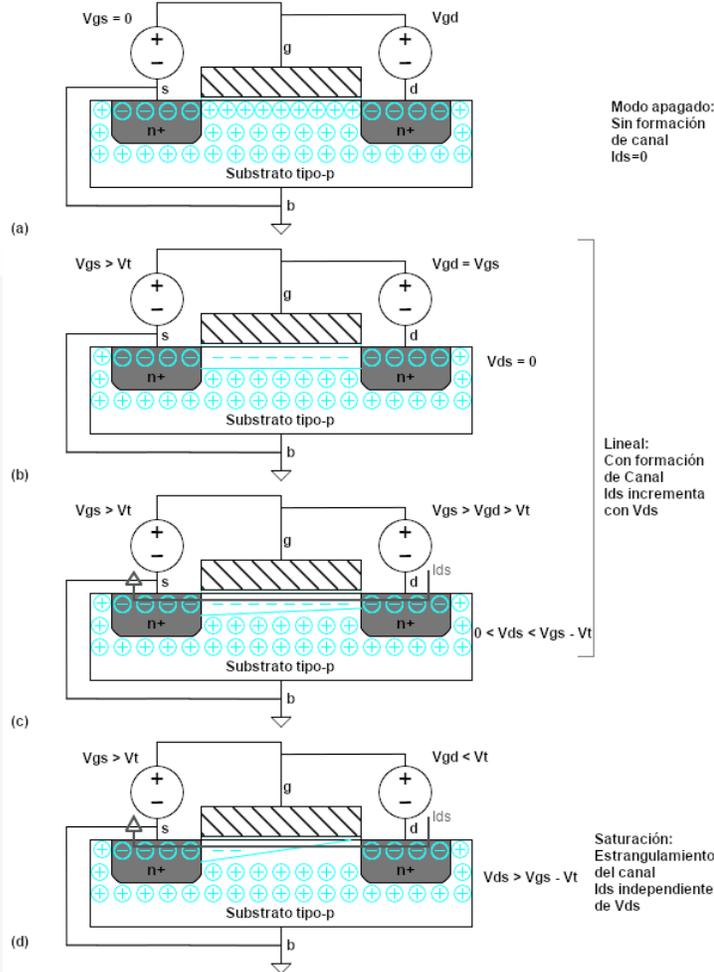


Figura 3. Transistor nMOS: (a) apagado, (b) y (c) lineal y (d) saturación (Neil H. E. Weste, 2011).

Por otra parte, existe un modelo que relaciona la corriente y el voltaje (I-V) para un transistor nMOS en las regiones apagado, lineal y saturación. El modelo asume que la longitud del canal es lo suficientemente larga que el campo eléctrico lateral (el campo entre el s y d) es relativamente bajo, lo que ya no aplica a dispositivos nanométricos. Este modelo es conocido como modelo de canal-largo, ideal, de primer orden, o Shockley (Neil H. E. Weste, 2011).

De acuerdo con este modelo, la corriente I_{ds} ideal en saturación es:

$$I_{ds} = \frac{\mu C_{OX} W}{2L} (V_{gs} - V_t)^2 \quad (1)$$

tal que C_{OX} es la capacitancia por unidad de área de la capa de dióxido de silicio, μ es la movilidad de los portadores, W es el ancho del canal del transistor *MOSFET*, L es la longitud del canal del transistor *MOSFET*, V_{gs} es el voltaje entre las terminales Compuerta y Fuente, y V_t es el voltaje de disparo (Neil H. E. Weste, 2011).

Los modelos de I-V ideales para los *MOSFETs* omiten muchos efectos que son importantes para dispositivos con longitud de canal por debajo de 1 micrómetro (canal corto). La corriente de saturación incrementa menos que el modelo cuadrático ideal con el incremento de V_{gs} . Esto es causado por dos efectos: saturación de velocidad y degradación de la movilidad (Neil H. E. Weste, 2011).

La degradación de la movilidad de los portadores disminuye con el incremento del campo eléctrico transversal perpendicular al óxido de la compuerta. Además, para muy altos campos eléctricos longitudinales en la región de estrangulación, la velocidad de los portadores se satura. Para transistores de canal corto, los portadores viajan a velocidad saturada sobre la mayoría del canal, en este caso, la corriente de Drenaje no incrementa cuadráticamente como el modelo ideal; pero muestra una dependencia lineal. Debido a lo anterior, la degradación de la movilidad y la velocidad saturada de los portadores degradan el comportamiento de la respuesta I-V (Ben G. Streetman, 2015). Además, de la degradación de la movilidad y la saturación de la velocidad de los portadores existen otros efectos de canal corto que degradan el desempeño general de un transistor de tipo *MOS*, estos efectos son:

- Variación del voltaje de disparo: La elección del voltaje de disparo (V_t) se basa en el desempeño del dispositivo en aplicaciones de circuitos típico. El límite superior del voltaje de disparo es aproximadamente igual a $\frac{V_{DD}}{4}$, donde V_{DD} es el voltaje de la fuente que se aplica en la terminal del Drenaje, esta cota superior fue establecida para evitar la degradación de la velocidad de

las compuertas CMOS digitales. El límite inferior del voltaje de disparo depende de la región llamada *subthreshold*, de la temperatura, y de la longitud del canal. La pendiente de la región *subthreshold* indica la razón de cambio de I_{ds} con respecto a V_{gs} . Por ejemplo, cuando V_{gs} disminuye para apagar el transistor I_{ds} decrece. Además, la pendiente indica que el límite inferior se aproxima al límite superior. El límite inferior es afectado por la temperatura más severamente. Es difícil reducir V_t por debajo de unos cientos de milivolts (Razavi, 2001).

La longitud del canal no puede ser controlada exactamente durante la fabricación de los dispositivos, este efecto introduce variaciones adicionales en V_t . Esto afecta en los diseños analógicos ya que si la longitud del canal es incrementada para lograr impedancias de salida altas, entonces el V_t también incrementa (Razavi, 2001).

Otro efecto de canal corto relacionado al voltaje de disparo es la reducción de la barrera inducida por el drenaje (*DIBL*, por sus siglas en ingles). El impacto principal de *DIBL* en el diseño de circuitos es la degradación de la impedancia de salida (Razavi, 2001).

- Degradación de la movilidad de los portadores: La región de inversión es inducida por un campo eléctrico vertical que aparece en la terminal de Compuerta, en un transistor *nMOS* la fuerza de un voltaje positivo aplicado en la terminal de Compuerta dirige a los portadores en la región de inversión hacia el Substrato. Cuando los electrones viajan en el canal hacia la terminal de Drenaje, ellos son atraídos hacia el Substrato, pero después son repelidos por fuerzas coulómbicas. Este efecto es llamado dispersión del Substrato, el efecto reduce la movilidad de los portadores. También, si se encuentra una carga positiva de óxido fija cerca de la interfaz de óxido-semiconductor, la movilidad será aún más reducida debido a fuerzas coulómbicas adicionales (Neamen, 2012).

- TESIS TESIS TESIS TESIS TESIS
- Saturación de velocidad de los portadores: La saturación de velocidad de los portadores esta acotada por la dependencia del campo eléctrico, esta cota disminuye cuando el campo eléctrico aumenta su fuerza, además, para dispositivos de canal corto este problema es más grave debido a que el campo eléctrico horizontal generalmente es más fuerte. También, la velocidad de saturación disminuye debido al voltaje aplicado en la terminal de Compuerta por el campo eléctrico vertical, y al efecto de dispersión del sustrato. La I_{ds} de saturación depende de la velocidad de saturación, con lo cual si la velocidad de saturación disminuye, también I_{ds} lo hará, e igualmente el V_{DSSAT} (Neamen, 2012).
 - Efecto de inyección de portadores calientes: Para transistores *MOSFET* de canal corto es probable que experimenten con campos eléctricos horizontales altos, con lo cual, la velocidad de las cargas se satura, y la velocidad instantánea se satura también. Por lo tanto, la energía cinética de las cargas continúa incrementándose, especialmente cuando ellas aceleran hacia la terminal de Drenaje. Esto es llamado portadores calientes (Razavi, 2001).

Cerca de la terminal de Drenaje los portadores calientes posiblemente colisionen con los átomos de silicio a velocidades altas, creando impactos de ionización. Como resultado nuevos electrones y huecos aparecen, los electrones son absorbidos por la terminal de Drenaje y los huecos por el sustrato, apareciendo una corriente de Drenaje-Substrato. Además, si los portadores obtienen la suficiente energía pueden alojarse en el óxido de silicio (aislante), y probablemente salgan de la terminal de Compuerta, apareciendo una corriente de Compuerta (Razavi, 2001).

- Variación de la impedancia de salida con el voltaje de Drenaje a Fuente: Cuando el voltaje de Drenaje-Fuente (V_{ds}) incrementa y la región de

estrangulamiento se mueve hacia la terminal de la Fuente, la tasa de disminución de la región de agotamiento cerca de la terminal Fuente da como resultado una impedancia de salida (R_{ds}) más alta (Razavi, 2001).

En dispositivos de canal corto, cuando V_{ds} tiende a valores grandes, el *DIBL* se vuelve significativo, reduciendo el voltaje de disparo e incrementando la corriente de drenaje, este efecto da una R_{ds} constante. A voltajes de Drenaje Suficientemente altos los impactos de ionización cerca de la terminal de Drenaje produce corrientes grandes (flujo del Drenaje al Substrato), con lo cual disminuye la impedancia de salida (Razavi, 2001).

La variación de R_{ds} se comporta de manera no lineal en muchos circuitos. En el amplificador cascode, por ejemplo, cuando el voltaje de salida varia, también la impedancia de salida del cascode y por lo tanto la ganancia de voltaje del circuito. Además, los impactos de ionización limitan la máxima ganancia que puede ser obtenida por el dispositivo cascode porque este efecto introduce resistencias en pequeña señal del Drenaje al Substrato más que a la Fuente (Razavi, 2001). En la Figura 4 se puede observar la variación de R_{ds} con respecto a V_{ds} , V_{gs} , la longitud del canal del transistor (L), manteniendo la relación W/L constante. Se puede observar que R_{ds} aumenta cuando la longitud (L) incrementa, y con respecto a V_{ds} sucede lo mismo. Los datos de R_{ds} se obtuvieron para un *MOSFET* de tecnología 180nm.

Efectos de canal corto en el *MOSFET*

Como se habló anteriormente, los efectos de canal corto surgen en los transistores *MOSFET* cuando estos tienden a dimensiones pequeñas, lo cual afecta el funcionamiento del *MOSFET* de una manera drástica comparado con los transistores de canal largo. Por consiguiente, se ha realizado una extensa investigación para disminuir los efectos de canal corto en estos transistores para poder aprovechar al máximo sus características. Algunos de los estudios que se han llevado a cabo se listan a continuación.

A. Análisis de efectos de canal corto.

- I. Corrientes de fuga.
 - II. Resistencia de salida, transconductancia de salida y capacitancias parásitas.
- B. Alternativas de material en los *MOSFETs*.
 - C. Modelos de tecnologías modernas de los *MOSFETs*.
 - D. Algoritmos de optimización implementados en el *MOSFET*.

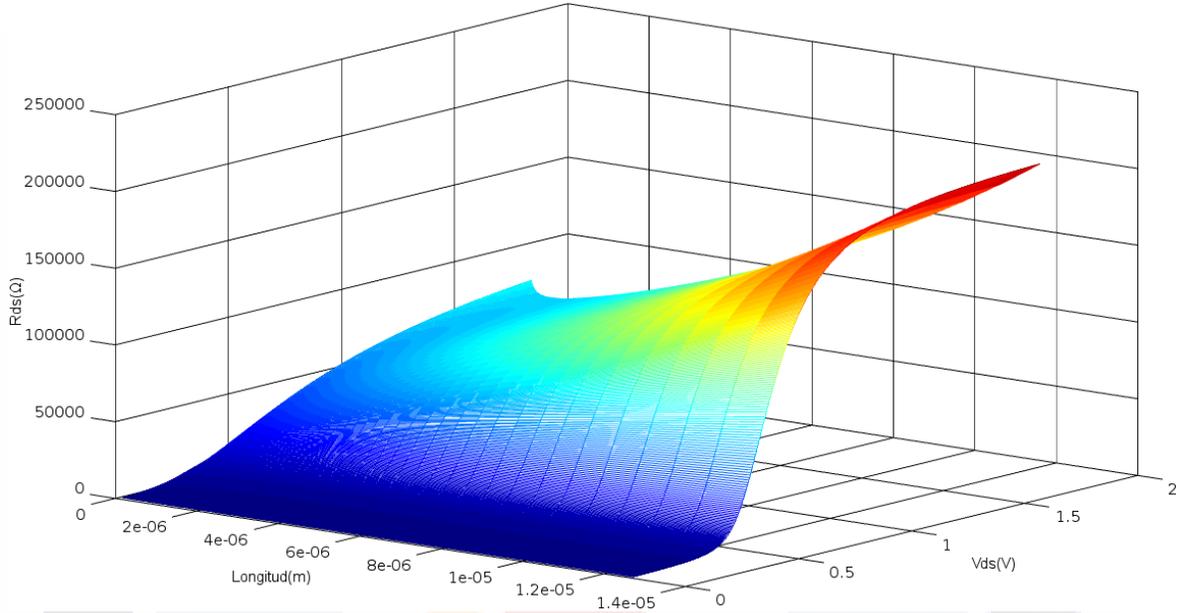


Figura 4 Variación de la impedancia de salida R_{ds} con respecto a L , V_{ds} y V_{gs} .

A. Análisis de efectos de canal corto.

Un efecto de canal corto es la corriente de fuga, la cual surge por diversas zonas del transistor *MOSFET*. Las corrientes de fuga degradan el comportamiento del transistor *MOSFET*, ya que reducen la ganancia de voltaje afectando el ancho de banda de los *opamps*, crece el consumo de potencia del *MOSFET* y por consecuencia de los *opamps*. Otros efectos de canal corto son la variación de la resistencia de salida (R_{ds}), la transconductancia de salida (g_m) y las capacitancias parásitas. Estos efectos de canal corto reducen la frecuencia de operación del dispositivo *MOSFET*. A continuación, se presentan algunos artículos donde investigan estos efectos de canal corto.

I. Corrientes de fuga.

La degradación del desempeño de un circuito, como la pérdida de potencia no puede ser modelada de forma precisa únicamente por los efectos de canal corto. Está demostrado que la eficiencia de potencia puede lograrse reduciendo las corrientes de fuga causadas por la contribución del canal corto (Mukhopadhyay et al., 2018).

En 2018, Mukhopadhyay et al. utilizaron resultados de simulación dispositivo-2D. La razón es que la información física microscópica puede ser extraída junto con el desempeño del dispositivo macroscópico. Para el modelado del *MOSFET*, HiSIM2 es utilizado ya que el modelo es compatible con simulación dispositivo-2D (Mukhopadhyay et al., 2018).

Las características de corriente de drenaje vs voltaje de compuerta ($I_{ds}-V_{gs}$) son calculadas con diferentes longitudes de canal (L_{ch}), para la menor longitud de canal se observa la degradación del disparo y la razón es el efecto de fuga. Para la extracción del voltaje de disparo (V_t), los autores utilizaron la extrapolación lineal g_m . La extracción del cambio del voltaje de disparo ($\Delta V_t = V_{t,largo} - V_t$) como una función de L_{ch} se realizó con simulación dispositivo-2D para dos generaciones de tecnología, es decir diferentes valores del espesor del óxido de compuerta (T_{ox}). Con esta simulación, los autores se percataron que las dos tecnologías son similares, con lo cual descartaron la corriente de fuga de Compuerta de su investigación. Los autores simularon ΔV_t vs L_{ch} y observaron que sigue la teoría; esto concluye que los efectos de canal corto son inducidos por campos eléctricos laterales adicionales debido a la reducción de la longitud del canal. Para evaluar la dinámica de los portadores, su movilidad es simulada con respecto a la L_{ch} . La simulación demostró que la movilidad se reduce cuando L_{ch} es reducida. También, los autores demostraron que la corriente incrementa cuando L_{ch} es reducida en la región por debajo de saturación; esto es atribuido a la contribución de las corrientes de fuga. Sin embargo, en la región de condición de inversión fuerte la corriente disminuye (Mukhopadhyay et al., 2018).

Para el estudio de la eficiencia de potencia, se utilizó un circuito inversor *CMOS* para el análisis. Los resultados de simulación muestran que en un rango de la longitud del canal, el consumo de potencia es casi constante. Pero cuando se reduce la longitud de canal el consumo de potencia aumenta drásticamente. También, los autores analizaron el consumo de potencia sin corriente de fuga. Se ha visto que el principal origen del incremento de consumo de potencia es la corriente de fuga (Mukhopadhyay et al., 2018).

Los dispositivos *CMOS* nanométricos tienen propiedades de corriente directa (*DC*, por sus siglas en inglés) muy pobres y más potencia es consumida para mantener el mismo rendimiento analógico. Los transistores en tecnología *CMOS* 65nm poseen una conductancia de salida muy grande, fugas de corrientes de Compuerta, corriente por efecto túnel y *voltaje Early*. Estos problemas hacen difícil el diseño de *opamps* de alta ganancia y bajo consumo de potencia, especialmente cuando el voltaje mínimo de operación (*Headroom*) es pequeño (Weixun et al., 2008).

La estructura de tres etapas es ampliamente utilizada para el diseño de *opamps*, porque la mayoría de los esquemas de compensación frecuencia eficiencia potencia populares son optimizados para tres etapas. Sin embargo, como se mencionó anteriormente, el diseño de *opamps* de alta ganancia y potencia es desafiante en tecnología *CMOS* 65nm debido a las propiedades de *DC* pobres de los transistores. De acuerdo con lo mencionado anteriormente, en 2008, Weixun, Kolm y Zimmermann diseñaron un *opamp* de compensación *nested-Miller* de cuatro etapas con una resistencia de anulación. La estructura eficiente del *opamp* de cuatro etapas *nested-Miller* con una resistencia de anulación está adecuadamente compuesto para lograr una alta ganancia con consumo de potencia bajo y una buena estabilidad (Weixun et al., 2008).

La mayor desventaja de los efectos de canal corto es la corriente de fuga, la cual fluye por diferentes partes del transistor debido a diferentes factores. Esto afecta la característica (I-V) de cualquier manera. Las corrientes por efecto túnel son una de

las principales fugas de corriente que tiene 5 componentes: la fuga de capacitores parásitos de compuerta, de la Compuerta a la región de traslape de fuente/drenaje (I_{go}) la cual crea dos corrientes: (I_{gso}) y (I_{gdo}), la fuga de Compuerta a canal invertido (I_{gc}) que tiene dos corrientes (I_{gcs}) y (I_{gcd}), y la fuga de corriente de Compuerta al Substrato (I_{gb}) (Rjoub, Taradeh, & Al-Mistarihi, 2014).

Los modelos que propusieron Rjoub, Taradeh y Al-Mistarihi incluyen ecuaciones simples con el objetivo de conseguir un comportamiento más rápido que el modelo BSIM4. Los autores tomaron en cuenta las variables que contribuyen a las corrientes de fuga como el voltaje del óxido, el potencial electrostático del óxido, la longitud efectiva, el campo eléctrico; se basaron en el modelo analítico BSIM4 y un modelo basado en potencial. Además, aproximaron las ecuaciones usando métodos matemáticos y físicos para tener ecuaciones más simples con alta exactitud. La corriente de fuga de Compuerta total se resolvió usando ley de corrientes de Kirchoff (*KCL*, por sus siglas en ingles) para ser igual a la suma de todos los tipos de fuga de Compuerta. Los resultados de simulación en las mismas condiciones para este tipo de fuga proveen un error promedio igual 2.87% entre el modelo propuesto por los autores y otros modelos (Rjoub, Taradeh, & Al-Mistarihi, 2014).

II. Resistencia de salida, transconductancia de salida y capacitancias parásitas.

En 2013, Baek et al. propusieron una nueva metodología de diseño de circuito analógico utilizando estructuras de *self-cascode* (SC) optimizadas independientemente. El *MOSFET* de Compuerta de función de trabajo doble (*DWFG*, por sus siglas en inglés) está construido con una Compuerta de poli silicio asimétricamente dopado para incrementar la resistencia de salida (R_{ds}) y la transconductancia (g_m). La longitud de la Compuerta lateral de la Fuente con una más alta función de trabajo es el mayor factor para optimizar R_{ds} y g_m . Asombrosamente, el *MOSFET DWFG* es un concepto similar a la estructura SC y puede ser aplicado a el diseño de circuitos analógicos, pero esta técnica exige pasos de proceso complicado. La tecnología CMOS estándar utilizada en el método

propuesto y la longitud del canal de la estructura SC fueron optimizadas por separado para obtener grande R_{ds} y g_m (Baek et al., 2013).

Los autores simularon g_m y g_{ds} ($=1/R_{ds}$) del SC *MOSFET* contra la longitud del canal, el cual se diseñó con dos transistores *MOSFETs* de canal p denominados como MS y MD en el artículo. El voltaje de polarización directa del Substrato para MD es aplicado para reducir V_t sin usar *MOSFETs* de bajo V_t para evitar pasos de procesos de fabricación adicional. Se logro obtener un grande R_{ds} y oscilación de voltaje de salida con condiciones del voltaje de polarización directa del Substrato de MD; El SC con $L_{MS} = 0.8 \mu m$ mostró las características de g_{ds} más bajas comparado con la configuración de un único transistor *pMOSFET* ($L = 1.0 \mu m$) y el SC *MOSFET* con $L_{MS} = 0.5 \mu m$. Además, las características de g_m mejoradas fueron también observadas en la simulación del SC. La g_m efectiva de la estructura SC está cerca de la g_m del transistor MS. Se observó en la simulación que la longitud del canal del transistor MS es un parámetro crítico para la optimización de g_m y g_{ds} del SC (Baek et al., 2013).

Los autores fabricaron un *opamp* convencional con únicamente *pMOSFETs* y el *opamp* con SC *pMOSFETs* propuesto usando tecnología CMOS $0.18 \mu m$ TSMC. La ganancia de DC medida del *opamp* usando el SC propuesto y el *opamp* convencional fueron 78.4 y 66.0 dB , respectivamente. El g_m y R_{ds} mejorado de la estructura del SC propuesto resultó en una mayor ganancia de DC del *opamp* (Baek et al., 2013).

Los *MOSFETs* sin unión (*JL*, por sus siglas en inglés) tienen muchas ventajas sobre el *MOSFET* en modo de inversión (*IM*, por sus siglas en inglés) (con uniones), tales como: supresión del *DIBL*, mejoras en la pendiente del subtrheshold, operación de potencia de modo bajo, bajo *DIBL*. Sin embargo, los *MOSFETs JL* tienen transconductancia (g_m) y frecuencia de corte (f_t) más bajas en comparación con el *MOSFET IM*, las cuales restringen las aplicaciones analógicas/RF de los dispositivos *JL* (Chen et al., 2013).

La estructura propuesta por Chen et al. (2013) llamada transistor *MOSFET* seudo sin unión de canal graduado (*GPJL*, por sus siglas en inglés) tiene como objetivo mejorar el desempeño analógico/RF de los *MOSFETs JL*. La única diferencia entre las estructuras *JL* y *GPJL* es que el transistor *GPJL* tiene una alta área de dopado n ($N_{gd} = 2.5 \times 10^{19} \text{ cm}^{-3}$) cerca del fin de canal del Drenaje, mientras el resto del substrato es de dopado n constante, con un $N_d = 2 \times 10^{19} \text{ cm}^{-3}$ (Chen et al., 2013).

En esta investigación los autores fijaron la longitud de la región de alto dopado (L_{gd}) a la mitad de L_g . Los autores examinaron el comportamiento dinámico del *MOSFET GPJL* y lo compararon con el *MOSFET JL* convencional usando una modificación del método de simulación Monte Carlo (MC). En las simulaciones de f_t vs I_{ds} y la máxima frecuencia de oscilación (f_{max}) vs I_{ds} se puede observar más altas f_t y f_{max} a lo largo de I_{ds} para el transistor *GPJL*. La mejora de f_t se debe al incremento de g_m y a la disminución de la capacitancia de Compuerta total ($C_g = C_{gs} + C_{gd}$), donde C_{gs} es la capacitancia de Compuerta a Fuente y C_{gd} es la capacitancia de Compuerta a Drenaje. El *MOSFET GPJL* no proporciona mejoras obvias de f_{max} porque el incremento de C_{gd} y el decrecimiento de g_{ds} se compensan mutuamente. La eficiencia de la generación de la transconductancia (g_m/I_{ds}) del transistor, la cual representa la ganancia disponible por unidad de valor de disipación de potencia es una de las figuras clave de mérito para aplicaciones analógicas de baja potencia. Una más alta g_m/I_{ds} se obtiene en el *MOSFET GPJL*, debido a la mejora de I_{ds} y g_m . El voltaje *Early* ($V_{ea} = I_{ds}/g_{ds}$) es otro parámetro analógico importante ya que una vez multiplicado por g_m/I_{ds} , este determina la ganancia de voltaje intrínseca ($A_{vo} = V_{ea} \times g_m/I_{ds}$) del dispositivo. El transistor *GPJL* también presenta altos valores de V_{ea} en comparación con el *JL*. Finalmente, la más grande g_m/I_{ds} y V_{ea} para el *MOSFET GPJL* resulta en una mejor A_{vo} (Chen et al., 2013).

En 2013 Chen et al. examinaron el caso cuando el canal entero bajo la Compuerta del *MOSFET GPJL* está altamente dopado ($L_{gd} = L_g$). En este caso, los valores de I_{ds} del *MOSFET GPJL* son ligeramente menores que los del dispositivo *JL*. Por lo

anterior, uno esperaría que hubiera un valor óptimo para L_{gd} (entre 0 y L_g). Además, incluso con el uso de un óptimo L_{gd} , mejoras sustanciales del *MOSFET GPJL* no serán observadas si la altura del gradiente de dopaje ($N = N_{gd} - N_d$) no es lo suficientemente grande. Consecuentemente, ambos L_{gd} y N_{gd} tienen valores óptimos. Debe señalarse que la elección del óptimo L_{gd} y N_{gd} podría variar cuando se trate de diferentes figuras de mérito (f_t , A_{vo} , etcétera) (Chen et al., 2013).

El transistor *MOSFET* nanométrico de Arseniuro de galio (*InGaAs*) con recrecimiento de las regiones Fuente y Drenaje fue diseñado para producir un rendimiento del dispositivo aparentemente excelente, es decir alta corriente I_{DSmax} y transconductancia g_m . Sin embargo, existen impactos de elementos parásitos del circuito asociados con tal estructura del *MOSFET* sofisticada. Cuando se utiliza para aplicaciones de radio frecuencia (RF) el impacto negativo de las resistencias y capacitancias parásitas aparecerán; también, otras estructuras de *MOSFET* de nanoescala mostrarán problemas similares. Para evaluar el rendimiento de RF de un dispositivo activo, es común usar la frecuencia máxima de oscilación (f_{max}), la cual depende de la resistencia de la Compuerta (R_G), resistencia de la Fuente (R_S), resistencia del Drenaje (R_D), capacitancia de Compuerta-Drenaje (C_{gd}), y otras variables; además, de la frecuencia de corte de ganancia de corriente (f_T), la cual depende de la capacitancia de Compuerta-Drenaje (C_{gd}), capacitancia de Compuerta-Fuente (C_{gs}), y otros parámetros (Lam & Chan, 2013).

En 2013, Lam y Chan asumieron un *MOSFET* de $W/L = 1000$, que es típicamente usado como transistor de RF. Por ejemplo, en un amplificador de ruido bajo. Usando una rama múltiple de layout de 30 ramas de Compuerta, las resistencias y capacitancias parásitas del transistor pueden ser estimadas por cálculos geométricos simples usando material revelado y datos estructurales. Con lo cual, resulta en $f_T = 184 \text{ GHz}$, y $f_{max} = 1.05 \text{ THz}$ que es impresionantemente alto. Determinaciones más exactas de f_T y f_{max} pueden ser obtenidas por cálculos de MATLAB de las matrices de impedancia y admitancia del circuito equivalente de

pequeña señal del transistor estudiado. Obteniendo, $f_T \approx 320 \text{ GHz}$ y $f_{max} \approx 530 \text{ GHz}$ (Lam & Chan, 2013).

Para evaluar los efectos de cada elemento del circuito equivalente de pequeña señal en el rendimiento de RF de un *MOSFET* de nanoescala, f_T y f_{max} son determinados para varios valores de los elementos del circuito. El parámetro f_T disminuye constantemente con el incremento de las capacitancias de Compuerta, mientras f_{max} también decrece con el incremento de la capacitancia de Compuerta, pero el impacto negativo es menos significativo. Además, los autores observaron que C_{gd} tiene un muy pequeño impacto en f_T y f_{max} comparado con C_{gs} . Ambas, f_T y f_{max} decrecen con el incremento de las resistencias en serie de Fuente y Drenaje, R_S y R_D . Como las resistencias disipan potencia, es intuitivo ver que f_{max} cae mucho más bruscamente que f_T . La R_G no cambia a f_T en absoluto; además, si R_G tiene un muy pequeño valor, también tiene un impacto mínimo en f_{max} . Sin embargo, cuando R_G incrementa a cierto valor, esta causa una caída repentina en f_{max} . Por lo tanto, estructuras de dispositivo optimizadas para reducir las resistencias parasitas son esperadas a pesar del incremento de las capacitancias parasitas (Lam & Chan, 2013).

C. Alternativas de material en los MOSFETs.

Debido a los efectos de canal corto, los transistores *MOSFETs* tienen características de *DC* muy bajas. Por lo cual, se ha realizado una gran investigación en las propiedades de los materiales de estos transistores para mejorar sus características de *DC*. Algunos ejemplos, se presentan en los siguientes artículos.

La reducción de los dispositivos semiconductores requiere alternativas de material semiconductor para SiO_2 (dieléctrico de la compuerta). Al cambiar la estructura del *MOSFET* una-compuerta a *MOSFET* doble-compuerta, el voltaje de disparo puede cambiarse, y con lo cual, la corriente de fuga y los efectos de canal corto pueden reducirse. En 2015, Srivastava diseñó un modelo *MOSFET* doble-compuerta con un material dieléctrico alto-k HfO_2 para la terminal de cada compuerta; con lo cual logró

minimizar el espesor del óxido de la compuerta, disminuir las corrientes de fuga, disminuir los efectos de canal corto, mayor ganancia de voltaje, una frecuencia de oscilación más alta, mayor ganancia de corriente, una excelente transconductancia y el voltaje de disparo no es afectado por los efectos del sustrato (Srivastava, 2015).

Los transistores de sustrato ultradelgado y óxido incrustado completamente agotado (*UTBB FD-SOI*) con capa de silicio delgada y óxido oculto (*BOX*) han resuelto el escalamiento de dimensiones, fugas, y problemas de variabilidad de la miniaturización de la tecnología *CMOS*. Comparando con las tecnologías de silicio-*bulk*, las tecnologías *UTBB FD-SOI* son más resistentes a los efectos de evento único (*SEE*, por sus siglas en inglés) porque el volumen de acumulación de los portadores en *UTBB FD-SOI* es extremadamente pequeño. Pero por otro lado, los efectos de irradiación de la dosis de ionización total (*TID*, por sus siglas en inglés) en los transistores *UTBB FD-SOI* son más complicados que los dispositivos silicio-*bulk*. Los *n-MOSFETs* usados en el experimento fueron fabricados en un proceso *UTBB FD-SOI*, con una longitud de canal de 20nm, un ancho de canal de 400nm y 160nm. Para cada tamaño de canal, cuatro transistores fueron elegidos para muestras experimentales, una muestra nueva sin irradiación, otras tres muestras fueron respectivamente irradiadas, con una dosis total de 300 krad (Si). El estrés de portadores excitados con la misma condición ($V_G = V_D = 1.5V$) fue realizado en todas las muestras. El cambio en las gráficas de las características de I-V son mayores para las muestras irradiadas que la muestra sin irradiación. La degradación de I_{dssat} y V_t es también más grande para las muestras irradiadas que la muestra sin irradiación (Cui et al., 2018).

Los efectos de las variaciones de procesos en el desempeño del *MOSFET* incrementan la dificultad para reducir las dimensiones del dispositivo. El análisis de variación está convencionalmente basado en las medidas macroscópicas de las propiedades del *MOSFET*, como, voltaje de disparo V_t o corriente de encendido I_{on} . Está demostrado que las variaciones macroscópicas pueden correlacionarse con

TESIS TESIS TESIS TESIS TESIS

las propiedades de variación del *MOSFET* microscópicas (concentración de dopado, movilidad de los portadores, etc.) usando modelos de potencial de superficie compactos como HiSIM2, y que estas variaciones microscópicas pueden ser determinadas cuantitativamente (Mattausch et al., 2009).

En 2009, Mattausch et al. verificaron que la extracción de los parámetros macroscópicos del HiSIM2 en un chip con valores medianos I_{on} y V_t y 2 longitudes de compuerta L_g con respecto a una oblea con concentración de impureza normal, es suficiente para predecir la distribución de variación de las variables $I_{on} - V_t$. También, la extracción de las variables macroscópicas fue correcto, cuando, se utilizaron otras dos obleas, pero con diferente impureza bajo y alto; y únicamente variando *NSUBC* (Dopado del sustrato de dispositivos grandes). Para la correlación de las variables macroscópicas con los parámetros microscópicos del HiSIM2, los parámetros microscópicos que determinaron cambios más grandes en los parámetros macroscópicos se tomaron en cuenta para el análisis. Estos parámetros son, *NSUBC* y *MUESR1* (Degradación de la movilidad con oxido rugoso , dispositivo grande) para $L_g = 10\mu m$; *XLD* (Cambios de longitud de canal) y *NSUBP* (concentración de dopado *peak pocket*) para $L_g = 180nm$. Después, los autores compararon las variaciones de medida de $I_{on} - V_t$ para L_g largo y corto de un n-*MOSFET*, con respecto a datos simulados del HiSIM2 tomando en cuenta sus parámetros intrínsecos y 16 combinaciones de límites de variación de los 4 parámetros microscópicos (*NSUBC*, *MUESR1*, *XLD* y *NSUBP*), y confirmaron la buena relación entre las variaciones medidas y simuladas dentro de la oblea para todas las longitudes de canal (Mattausch et al., 2009).

Por lo tanto, esto confirma que únicamente estas 4 más sensibles variaciones de modelos de parámetros microscópicos son suficientes para confirmar la dependencia de L_g con respecto a las variaciones macroscópicas medidas (Mattausch et al., 2009).

En 1992, Frank, Laux y Fischetti utilizaron el método de simulación Monte Carlo para explorar las características de un *MOSFET* canal-n. Está demostrado que los *FETs* doble compuerta y *FETs* con una compuerta superior y un nivel de tierra posterior son menos inmunes a los efectos de canal corto y por lo tanto sus dimensiones pueden ser más pequeñas. También, Los *FETs* doble compuerta deberían tener un desempeño mejor, debido a la menor capacitancia parasita a tierra. Por lo tanto, este estudio se basa en la tecnología doble compuerta (Frank et al., 1992).

En el diseño de *FETs* de longitud de compuerta muy pequeña (por debajo de $0.1\mu\text{m}$) el voltaje de drenaje debe estar limitado, evitar los túneles banda a banda, limitar la ionización por impacto, y prevenir la degradación de los portadores del óxido. Los dos primeros casos fueron considerados en este estudio. Los autores calcularon el voltaje de disparo con respecto a la longitud de compuerta para 3 diferentes anchos de canal con el modelo de deriva difusión. No se espera que este modelo produzca corrientes de disparo exactas en los dispositivos de canal corto, pero las corrientes de disparo del modelo dominado por difusión deberían ser más exactas que las obtenidas por el método de simulación Monte Carlo. Por último, los autores se basaron en tres aspectos físicos para saber la mínima dimensión del canal los cuales son, cuantificación, resistencia de la fuente y dopado discreto (Frank et al., 1992).

Además, para probar las predicciones anteriores los autores usaron el método de simulación Monte Carlo, tomando en cuenta las características de disparo ya descritas anteriormente de un *MOSFET* doble compuerta de longitud de canal 30nm y ancho de canal de 5nm. De acuerdo con la simulación Monte Carlo, este dispositivo se comporta como un *FET* propio, adecuado para la lógica digital. Además, se usó el método de simulación Monte Carlo para calcular el tiempo de retardo de encendido y apagado de un circuito inversor (Frank et al., 1992).

D. Modelos de tecnologías modernas de los MOSFETs.

Ya que han surgido nuevas tecnologías de los *MOSFETs* por el estudio de los materiales de estos transistores. Entonces, se requieren nuevos modelos para estas tecnologías. En las siguientes investigaciones, los autores obtuvieron un modelaje para las tecnologías analizadas.

En 2015, Huang y Li analizaron los impactos del ancho fin(W_{fin}) y altura fin(H_{fin}) del canal en compuerta múltiple III-V *MOSFETs*. Los autores simularon numéricamente las características de transferencia y salida, y los efectos de canal corto. Sin embargo, las características eléctricas óptimas de dispositivos de III-V no habían sido computacionalmente investigadas cuando se realizó este estudio. Por lo tanto, los autores estudiaron las características de *DC* del dispositivo *MOSFET* de triple-compuerta *InGaAs* con una longitud de compuerta de 14 nm. Las características de *DC* del dispositivo investigado fueron simuladas resolviendo numéricamente un conjunto de ecuaciones de gradiente de densidad de difusión-acumulación. Para validar la precisión de la simulación del dispositivo, Huang y Li calibraron la simulación de dispositivos 3-D a los datos experimentales. Primero, se mantuvo constante H_{fin} y se examinaron las variaciones de características cambiando W_{fin} , las características que se analizaron fueron la densidad de electrones del canal del dispositivo, I_{ON} , I_{OFF} . Después, se mantuvo constante W_{fin} y se varió H_{fin} y las características que se analizaron fueron la banda de conducción, confinamiento de electrones, acomodamiento de electrones, I_D - V_G , la reducción del *DIBL*. Para lograr las características de *DC* superiores, el confinamiento cuántico más fuerte y la mejor supresión de los efectos de canal corto, dispositivos con $W_{fin} = 10\text{nm}$ y H_{fin} entre 14 y 21 nm podrían ser aceptados (Huang & Li, 2015).

Como solución para los efectos de canal corto (*SCEs*, por sus siglas en inglés), una variedad de dispositivos *FETs* Multi-Compuerta fueron propuestos estos últimos años. El *TMOS* con Compuerta cilíndrica (*SRG MOSFET*, por sus siglas en inglés) representa uno de los mejores candidatos para continuar el escalamiento. Los *SCEs* no pueden ignorarse más para longitudes de canal más bajas que 50nm. En este contexto, algunos modelos son desarrollados para *SRG MOSFET* con canal corto y

TESIS TESIS TESIS TESIS TESIS

para el modelado de los SCEs. En este manuscrito, una Compuerta con doble material (*DMG*, por sus siglas en inglés) y Canal gradual (*GC*, por sus siglas en inglés) con dos diferentes espesores de oxido (*DOT*, por sus siglas en inglés) son combinados en esta estructura de dispositivo nuevo de *MOSFET* de Compuerta circundante (*SG*, por sus siglas en inglés). De las ecuaciones de Poisson y la aproximación del Canal gradual (*GCA*, por sus siglas en inglés); el potencial de superficie, el voltaje de disparo y por debajo del voltaje de disparo son calculados (Jaafar et al., 2018).

En 2018, Jaafar, Aouaj, Bouziane e Iñiguez presentaron expresiones analíticas de la densidad de corriente de Drenaje, transconductancia g_m y conductancia de Drenaje g_{ds} para esta nueva estructura de *MOSFET* de Compuerta cilíndrica. Los resultados se comparan con los de *DMG MOSFET* y *DMGDOT MOSFET*. El simulador 2D ATLAS fue usado para validar los resultados del modelo. En este trabajo los autores concluyeron que hay un muy cercano acuerdo entre los resultados obtenidos usando el modelo analítico y los resultados obtenidos usando el simulador de dispositivos. Esto apunta a la exactitud del modelo analítico propuesto y en general el *MOSFET DMG-GC-DOT* ofrece una mejora significativa de las características del dispositivo (Jaafar et al., 2018).

El uso de modelos exactos para dispositivos *MOSFETs* se vuelve eficaz en la obtención de circuitos de rendimiento confiable. Debido a esto, ha habido innumerables modelos *MOSFET* tal como MOS1, MOS2, MOS3, BSIM1, BSIM2, BSIM3, etcétera. El modelo BSIM ha sido desarrollado para incorporar los efectos de canal corto y los efectos de ancho estrecho (Jaafar et al., 2018).

En años recientes, los transistores con sustrato de germanio (Ge) han sido el foco de investigación para alto rendimiento de los *pMOSFETs*, ya que el Ge exhibe una alta movilidad de huecos en todos los materiales. Lo extraordinario de las propiedades de transporte de los huecos es aprovechada para fabricar *pMOSFETs* de canal de Ge de alto rendimiento para obtener alta frecuencia analógica y lógica

de alta conmutación. Sin embargo, los parámetros BSIM3v3 pertenecientes a los dispositivos de canal de Ge aún no han sido reportados (Bera et al., 2017).

En esta investigación, en 2017, Bera, Mondal y Biswas intentaron extraer el conjunto de parámetros BSIM3v3.2.2 para los *MOSFETs* de canal de Ge usando ante todo características experimentales y sus resultados de simulación *TCAD*. El algoritmo de los autores escrito en MATLAB se usa para emparejar las ecuaciones del modelo BSIM3v3.2.2 con las características de simulación *TCAD* o experimentales para diferentes geometrías de dispositivos y condiciones de polarización, y de esta manera extraer los parámetros BSIM3v3.2.2 para dispositivos de canal de Ge. Habiendo extraído los parámetros, los autores usaron el simulador *ADS* para obtener las características de transferencia, y hacer una comparación con las características experimentales y sus características de simulación *TCAD*. En base a la comparación, Bera, Mondal y Biswas concluyeron que su algoritmo de extracción es correcto y exacto. Las características del *pMOSFET* de canal de Ge con los parámetros BSIM3v3.2.2 extraídos en una longitud del canal de 70 nm fueron usadas para obtener los parámetros del dispositivo; tales como la transconductancia, la conductancia de salida, la ganancia de voltaje y la frecuencia de corte de ganancia unitaria (Bera et al., 2017).

E. Algoritmos de optimización implementados en el MOSFET.

Los efectos de canal corto degradan el correcto funcionamiento del *MOSFET*, y por lo cual las características de los circuitos analógicos tienen pérdidas significativas. Debido a esto, se han diseñado o implementado muchos algoritmos que obtienen el circuito analógico o *MOSFET* idóneo. En las siguientes investigaciones, los autores optimizaron un circuito analógico o *MOSFET* con un algoritmo evolutivo, una metaheurística o un algoritmo matemático.

Cuando las dimensiones de la tecnología *CMOS* tienden a escalas pequeñas de submicrométricas a nanométricas, la complejidad de los modelos de ruido plantea desafíos para la formulación de la función objetivo y restricciones de diseño en la

forma requerida por la programación geométrica. Los autores utilizaron un diseño de amplificador de bajo ruido con arquitectura “cascode” y también un inductor con un valor pequeño, con el fin de tener una respuesta del amplificador óptima. La ecuación de ruido de los amplificadores de ruido bajo (*LNA*, por sus siglas en inglés) puede ser calculada con análisis de pequeña señal; el ruido térmico es el mayor problema en las frecuencias intermedias RF para los *MOSFETs* (Jin & Hoe, 2012).

Además, en 2012 Jin y Hoe obtuvieron el factor de ruido con respecto al ruido térmico de la resistencia de la fuente de alimentación, el ruido térmico del canal, el ruido de Compuerta inducido y el ruido térmico de la resistencia de salida; luego, obtuvieron este factor de ruido en la frecuencia de resonancia, el cual dependía del ruido blanco que aumentaba cuando la longitud del canal disminuía, de la corriente I_{ds} con efectos de canal corto (Jin & Hoe, 2012).

En este trabajo se tomaron en cuenta ciertas cotas de diseño de acuerdo con las variables de los *LNAs*, como factor de calidad del circuito de entrada, emparejamiento de la impedancia de entrada, potencia de consumo y máxima potencia de disipación permitida; también, tomaron en cuenta para los *LNAs* que la potencia de disipación puede ser demasiado alta cuando el ruido es minimizado. Además, realizaron un análisis fisicomatemático de las variables intrínsecas del amplificador para obtener las ecuaciones a optimizar, las cuales incluían el factor de ruido y las cotas de diseño (Jin & Hoe, 2012).

Además, se utilizó programación geométrica para resolver problemas de optimización donde la función objetivo está en la forma de una función posinomial, y las cotas están expresadas como desigualdades posinomiales e igualdades monomiales. La función objetivo para este problema de optimización es minimizar la ecuación de ruido, y las cotas de diseño mencionadas anteriormente son tomadas en cuenta como restricciones en la programación geométrica. De acuerdo con el tiempo promedio para obtener los valores óptimos de los *LNAs*, se concluye que la programación geométrica tiene una eficiencia computacional alta (Jin & Hoe, 2012).

Los valores óptimos fueron validados con resultados de simulación del software sistema de diseño avanzado (*ADS*). Para el proceso de 90nm, el ancho de la Compuerta y el mínimo en la ecuación de ruido obtenidos con programación geométrica y *ADS* tuvieron discrepancias, esto se debe probablemente al exceso de ruido térmico presente en dispositivos con longitudes de Compuerta menores que 100nm. Para el proceso de 180nm, el ancho de la Compuerta y el mínimo en la ecuación de ruido obtenidos con los dos métodos muestran una tendencia muy similar, esto confirma que la programación geométrica es un método eficiente para el diseño de *LNAs CMOS* de canal corto (Jin & Hoe, 2012).

En 2006, Cheung y Wong utilizaron programación geométrica (*GP*) para la optimización del diseño de amplificadores de ruido bajo (*LNA*) con transistores *CMOS*. Métodos basados en simulación como recocido simulado, algoritmos genéticos, y algoritmos evolutivos inteligentes sufren de una carga computacional alta debido a muchas simulaciones iterativas (Cheung & Wong, 2006).

Optimización basada en ecuaciones como programación geométrica produce parámetros de circuito óptimos globales en un muy corto tiempo. La diferenciación matemática es utilizada para la minimización de las ecuaciones de ruido de los *LNAs*. Sin embargo, tal enfoque es ineficiente en el cálculo de la solución óptima global, y rápidamente se vuelve intratable cuando restricciones de diseño adicionales son agregadas (Cheung & Wong, 2006).

La programación convexa se refiere a la minimización de una función objetivo convexa sobre un conjunto convexo. La *GP* es un tipo especial de técnicas de programación convexa que consiste en polinomios positivos llamados posinomios. La propiedad de la *GP* es si existe su óptimo global puede ser resuelto con gran eficiencia usando algoritmos de punto interior estándares por ejemplo, resuelve en segundos en una PC un problema con 10^3 variables y 10^4 restricciones. Por la eficiencia computacional, la *GP* se implementó en un *LNA* con respecto a la longitud

del canal, el ancho del canal, factor de calidad, emparejamiento de la impedancia de entrada, la frecuencia de operación y la transconductancia; estas variables formaron parte de las restricciones de la *GP*. La función objetivo fue minimizar el ruido del *LNA* (Cheung & Wong, 2006).

Los resultados de la *GP* obtenidos con diferentes factores de calidad son precisos con respecto al método de diferenciación parcial. Además, los autores compararon en una gráfica la función de ruido y el ancho del transistor con respecto a diferentes factores de calidad, ambas funciones mostraron una tendencia de decrecimiento al aumentar el factor de calidad, con un comportamiento de ambas funciones casi idéntico (Cheung & Wong, 2006).

En 2010, Jafari, Zekri, Sadri y Mallahzade utilizaron un algoritmo genético (*GA*) para la optimización del amplificador de transconductancia operacional *CMOS* dos etapas (*OTA*). En el *GA*, cada parámetro desconocido es llamado gen, el vector de parámetros es llamado cromosoma y una función objetivo llamada función fitness. El objetivo del *GA* es determinar los elementos del vector desconocido (cromosoma) para maximizar la función fitness definida. Uno de los problemas más graves en el diseño de circuitos integrados analógicos y especialmente en los *opamps* es el bajo voltaje, la baja potencia, la alta ganancia y el gran ancho de banda. En el problema presentado por los autores el vector de parámetros desconocido contiene W (ancho del canal), L (longitud del canal), Voltaje de sobremarcha (V_{gs}) y una capacitancia del *OTA*. Además, los autores aplicaron *GA* multiobjetivo para optimizar la ganancia, la ganancia de ancho de banda, el *slew rate*, el margen de fase y el consumo de potencia en el *OTA*. El desempeño total del circuito es evaluado por la función fitness en problema multiobjetivo. Los autores presentaron una gráfica donde se muestra la convergencia del algoritmo genético. Después de usar el *GA*, las características de desempeño que se obtuvieron fueron incluso mejores que las del objeto deseado (la ganancia, el ancho de banda, el *slew rate*, etcétera) (Jafari et al., 2010).

El algoritmo genético y el recocido simulado son utilizados para la optimización de circuitos analógicos. Sin embargo, fue comprobado que estos algoritmos presentan algunos inconvenientes, como tiempo de cálculo excesivo, demasiados parámetros para ajustar y demasiadas restricciones. Las metaheurísticas basadas en la inteligencia de bandadas de animales como optimización de bandadas de partículas (*PSO*), y optimización de colonias de hormigas (*ACO*) superan los inconvenientes de los algoritmos antes mencionados, y ofrecen flexibilidad de manipulación lo cual permite que el algoritmo se adapte más fácil de una aplicación a otra (Kotti et al., 2011).

En 2011, Kotti et al. utilizaron *PSO* y *ACO* para la optimización de un convertidor de corriente de segunda generación (*CCII+*) y un amplificador operacional *CMOS* dos etapas (*OP-AMP*). En la optimización del (*CCII+*), las funciones objetivo fueron disminuir la resistencia parasita de entrada y maximizar la frecuencia de corte alta, y las variables que fueron usadas para el rendimiento de optimización son la longitud del canal y el ancho de la compuerta respetando las condiciones de saturación para cada transistor *MOSFET*. En la optimización del *OP-AMP*, se optimizó de acuerdo con los siguientes objetivos: maximizar la ganancia de voltaje de lazo abierto, minimizar el área del transistor, maximizar la relación de rechazo en modo común y minimizar la disipación de potencia. Los autores demostraron en estos casos en particular que el algoritmo *ACO* ofrece mejores resultados en términos de robustez, mientras que el *PSO* es más rápido y requiere menos parámetros para su manejo (Kotti et al., 2011).

En 2015, Maji et al. utilizaron el algoritmo búsqueda armónica basada en oposición (*OHS*) para encontrar el diseño óptimo de los circuitos amplificadores diferenciales. Los autores utilizaron el concepto de aprendizaje basado en la oposición (*OBL*). El *OBL* se basa en la consideración simultánea de una estimación y su correspondiente estimación opuesta es decir conjetura y conjetura opuesta con el fin de lograr una mejor aproximación para la solución candidata actual. Para

incrementar la tasa de convergencia de el algoritmo búsqueda armónica (*HS*) utilizaron *OBL* (Maji et al., 2015).

Las especificaciones de diseño son la parte más importante de cualquier diseño de circuito analógico. Las especificaciones del amplificador diferencial propuesto por los autores son el *Slew Rate*, la ganancia de voltaje *DC* de pequeña señal, la frecuencia de corte, el máximo y mínimo rango de modo común de entrada y la potencia de disipación. La razón de cambio (*W/L*) de los transistores *MOS*, la capacitancia de carga y la capacitancia de compensación son consideradas como las variables de diseño del amplificar diferencial. En el *OHS* se tomó en cuenta la relación entre las especificaciones del amplificador y las variables de diseño del amplificador. Las especificaciones del amplificador y la capacitancia de carga fueron utilizadas por el *OHS* para optimizar el amplificador; además, el voltaje de alimentación positivo y negativo del amplificador, el voltaje de disparo del *NMOS* y *PMOS*, y la transconductancia del *NMOS* y *PMOS*. La función objetivo fue minimizar el total de área que ocupan los transistores en el amplificador diferencial (Maji et al., 2015).

Los resultados de simulación comprueban que el diseño realizado en *OHS* cumple con las especificaciones y parámetros de diseño, por ejemplo, minimizo el total de área ocupada por los transistores en el circuito propuesto y minimizo el total de potencia disipada en comparación con el algoritmo genético, el algoritmo de optimización de bandadas de partículas (Maji et al., 2015).

En 2015, Cantürk y Kahraman compararon las ventajas de dos algoritmos evolutivos multiobjetivo los cuales son el algoritmo de búsqueda armónica (*HSA*), y el algoritmo genético de clasificación no predominante (*NSGA-II*) para el diseño de *opamps* de dos etapas. En un diseño de un *opamp*, la parte más importante es tener baja disipación de potencia, alta ganancia y gran ancho de banda. Estos parámetros son ajustados por optimización paramétrica en el diseño de circuitos integrados analógicos. En *HSA*, la disipación de potencia es definida como la función objetivo;

TESIS TESIS TESIS TESIS TESIS

además, la ganancia, el ancho de banda, y el *slew rate* son definidos como restricciones. Mientras el *HSA* intenta minimizar la disipación de potencia, en todo momento verifica las restricciones para ver si se cumplen o no. Si no se cumplen, el algoritmo usa una función de penalización para encargarse de esto. En *NSGA-II* todas las funciones son definidas como funciones objetivo. El *NSGA-II* intenta maximizar la ganancia, el ancho de banda y el *slew rate* a su punto más alto y al mismo tiempo el algoritmo trabaja para minimizar la disipación de potencia. Además, el *HSA* y el *NSGA-II* utilizan restricciones de voltajes de operación del amplificador operacional de dos etapas, la longitud y el ancho del canal para el correcto desempeño del amplificador. Los resultados de simulación comprueban que todas las especificaciones de diseño del amplificador se satisfacen usando ambos algoritmos (Cantürk & Kahraman, 2015).

En 2009, Liu y He tomaron en cuenta un algoritmo genético (GA) para el diseño de circuitos analógicos automatizados, en el cual, el cruzamiento es usado para desarrollar la topología del circuito, y la mutación es usada para desarrollar los valores de los componentes del circuito. La tasa de cruzamiento del GA fue de (.6 – 1.0) y la tasa de mutación fue de (.001 - .1). Como consecuencia con la baja tasa de mutación, los autores propusieron un algoritmo genético híbrido con hipermutación y estrategias elitistas. La hipermutación es usada para incrementar la tasa de mutación de los valores de los componentes del circuito, y la estrategia elitista es usada para mantener soluciones óptimas y ayudar a dirigir la búsqueda de la hipermutación a regiones más cercanas al óptimo del espacio solución. El problema de comprobación del experimento es para el diseño automatizado de un filtro analógico. Se compararon las funciones fitness del GA y el algoritmo propuesto sin estrategia elitista, con incrementos de hipermutación de .1 en el rango [.1, 1]. Los resultados fueron que la hipermutación dota al algoritmo propuesto con mejor convergencia que el GA. Además, se volvieron a comparar el GA y el algoritmo propuesto, pero esta vez con estrategia elitista. En este caso el algoritmo con hipermutación tuvo mejor convergencia que el GA, excepto con hipermutación de .8 (Liu & He, 2009).

En 2003, Abdel-Bary y Khalil utilizaron una técnica de optimización numérica para el diseño de un oscilador de control de voltaje *LC CMOS* diferencial, la función objetivo es minimizar el ruido de fase junto con las restricciones que se obtuvieron por análisis de ecuaciones intrínsecas del oscilador. Los autores realizaron un análisis de las variables del oscilador para obtener las fuentes de ruido que causan el deterioro de la respuesta en el dominio de fase (ruido de fase) del oscilador. Además, se realizó análisis de pequeña señal para obtener los parámetros necesarios que se utilizaron como restricciones en la optimización numérica. Los autores utilizaron programación cuadrática secuencial para minimizar el ruido de fase tomando en cuenta las restricciones. Los resultados que se obtuvieron del algoritmo de optimización son exactos con respecto a los resultados de simulación (Abdel-Bary & Khalil, 2003).

Como es conocido, la industria de semiconductor desarrolla de acuerdo con la ley de Moore. Este desarrollo necesita modelos matemáticos para ser rápidamente modificado y completado con nuevos parámetros y ecuaciones, que es logrado por el uso de los llamados modelos matemáticos “flexibles”. Uno de los ejemplos de aplicación de los modelos flexibles es el uso de los algoritmos genéticos, para la solución del problema clásico *CAD* de variación en la longitud del transistor en un circuito (Melik-Adamyan, 2009).

En este trabajo, en 2009 Melik y Adamyan consideraron la técnica de optimización multiobjetivo. El algoritmo de los autores da flexibilidad de selección a los ingenieros de diseño para la optimización y la búsqueda de compensación de soluciones. De la tecnología de 90 nm e inferiores, hasta el 50% del consumo total de potencia cae dentro de las fugas de la región por debajo del voltaje de disparo. De acuerdo con los autores, otro problema relevante es proporcionar la razón de rendimiento, el cual involucra la formación de una nueva disciplina de plano de diseño con respecto a los procedimientos de operación. Uno de los métodos relativamente simples que puede ser usado en ambos casos es el dimensionamiento del transistor. Los autores

indican las siguientes ventajas de la optimización multiobjetivo sobre la mono-objetivo: optimiza varios objetivos simultáneamente, no requiere la interferencia de un usuario en el mejor de los casos y en ausencia de una solución, da soluciones cerca de los óptimos de Pareto. La aplicación de los algoritmos genéticos sirve para solucionar los problemas multiobjetivo en sistemas *CAD* (Melik-Adamyán, 2009).

Para el uso de optimización en un ciclo de diseño estándar, los autores unieron el optimizador con el programa SPICE para la evaluación explícita de los resultados. El total de corriente de fuga para un circuito es obtenido por la simple suma de valores para todos los transistores, y este total multiplicado por una constante, la cual representa la acumulación y la relación de dimensiones de los transistores. El total de corriente de fuga puede ser usado por todo tipo de transistores. El modelo de área se obtiene por la simple suma del área de los transistores multiplicado por un coeficiente de peso, este coeficiente es determinado por su participación en el seguimiento y muestra la influencia de la variación de su correspondiente transistor en el área total. El algoritmo fue probado para algunas librerías de circuitos estándar y un subconjunto de circuitos de la familia *ISCAS*. El análisis estadístico del tiempo fue hecho por medio de Berkley PTM para modelos de transistor 130 nm. A pesar del número diferente de transistores, el algoritmo da algunas soluciones de consumo de potencia vs área y optimización de retraso vs área para un ingeniero de diseño, para seleccionar, las cuales constituyen las soluciones óptimas de Pareto. Los autores concluyeron que la optimización un-objetivo resulta desigual para la búsqueda de las soluciones. Sin embargo, el algoritmo genético permite seleccionar cualquiera del conjunto de soluciones óptimas de Pareto. Además el consumo de potencia y área son mejorados por 12-18% (Melik-Adamyán, 2009).

El comportamiento de los parámetros de pequeña señal del *MOSFET* de Compuerta doble apilada (*GSDG*), son estudiados y optimizados usando algoritmos genéticos multiobjetivo (*MOGAs*) para aplicaciones de circuitos analógicos *CMOS* nanométricos. Para el año 2013 la optimización multiobjetivo ya se había introducido para estudiar los sistemas complejos y no lineales, y ya se habían encontrado

TESIS TESIS TESIS TESIS TESIS

aplicaciones útiles en el campo de la ingeniería. Debido al mecanismo simple y alto rendimiento proporcionado por *MOGAs* para optimización global multiobjetivo, *MOGA* puede ser aplicado para estudiar los *MOSFETs GSDG* nanométricos. En el estudio presente, el enfoque de Pareto debería ser una opción adecuada, debido a que busca soluciones no dominantes llamadas soluciones optimas de Pareto en el espacio objetivo (Bendib & Djefal, 2013).

El primer paso del enfoque de Bendib y Djefal consiste en modelos compactos de los parámetros de transconductancia (g_m) y corriente de apagado (I_{OFF}) para los *MOSFETs GSDG* propuestos. Usando el software SILVACO, fue observado que los modelos de parámetros formulados pueden ser usados como funciones objetivo, que se dan como funciones de variables de diseño de entrada. Con lo cual, el diseño obtenido puede proporcionar los mejores parámetros de saturación y la región llamada por debajo del voltaje de disparo satisfaciendo las siguientes funciones objetivo: $g_m(x)$ (Maximización) y $I_{OFF}(x)$ (Minimización), tal que el vector de variables de entrada x contiene parámetros geométricos y eléctricos. Los parámetros de *MOGA* fueron variados y se registró el error de optimización asociado. Para esta configuración, la función objetivo global fue de 28.877 y casi el 99% de los casos presentados fueron aprendidos correctamente. Esto resultó en 10'000 evaluaciones de conjunto de parámetros, y tomó alrededor de 20 s para completar usando Windows XP con Pentium IV (1.5 GHz). Con el fin de mostrar el impacto de la metodología de diseño de los autores en un diseño de circuito nanométrico, en 2013, Bendib y Djefal propusieron estudiar el comportamiento eléctrico de un circuito oscilador de anillo, con y sin optimización de transistor. Se utilizó el simulador numérico SILVACO para la simulación de la frecuencia del circuito oscilador de anillo, donde una gran mejora de la frecuencia de oscilación ha sido registrada para el dispositivo optimizado, en comparación con el dispositivo sin optimizar (Bendib & Djefal, 2013).

La tecnología basada en *FET* es una alternativa prometedora para superar el alto consumo de potencia no deseado para dispositivos ópticos submicrométricos de

alto desempeño, debido a las propiedades ópticas y eléctricas superiores ofrecidas por la tecnología *CMOS*, en ambas, analógico así como en aplicaciones digitales. Recientemente, el *FET* controlado ópticamente (*OC*), ha sido impulsado agresivamente a escalas submicrométricas con el fin de lograr la mejora deseada en unidad de corriente, consumo de potencia bajo y velocidad de operación. Sin embargo, los dispositivos *FET-OC* desarrollados presentan los problemas conocidos de baja velocidad de conmutación y baja ganancia óptica, que pueden degradar el desempeño óptico y eléctrico del dispositivo. La mayoría de los trabajos publicados en *FETs-OC* tratan principalmente con investigaciones experimentales y simulaciones numéricas de las estructuras *FETs-OC* de Compuerta única (*SG*), en el que la compuerta consiste en una sola región dopada de Ge tipo-n (Djeffal & Ferhati, 2016).

En 2016, Djeffal y Ferhati investigaron y optimizaron analíticamente un nuevo *FET-OC* doble Compuerta texturizada (*DTG*) usando *MOGAs* para mejorar el desempeño del dispositivo, para comunicaciones ópticas basadas en *CMOS* de alta velocidad y bajo consumo de potencia. La Compuerta del *FET-OC DTG* se compone de dos regiones de Ge diferentes con dos concentraciones de dopaje diferentes llamadas $N_1\text{-Ge}$ y $N_2\text{-Ge}$, donde la superficie de la Compuerta es texturizada para ambas regiones. En este trabajo, los autores se centraron en el papel de nuevos parámetros de diseño y condiciones de operación utilizando un modelo analítico. Los resultados analíticos para ambos diseños (*DTG* y *SG*) concuerdan con los resultados numéricos del simulador de dispositivos ATLAS 2D. El dispositivo *FET-OC DTG* propuesto proporciona un mejor comportamiento de la corriente de drenaje frente a la potencia óptica, un mejor consumo de potencia y un mejor desempeño de ganancia óptica (G), en comparación con el diseño *FET-OC SG* convencional (Djeffal & Ferhati, 2016).

Una alta razón I_{ON}/I_{OFF} tiende a aumentar la sensibilidad del dispositivo y disminuir el consumo de potencia del dispositivo. Por los resultados de simulación de los autores, el diseño propuesto proporciona un dispositivo optoelectrónico con alta

TESIS TESIS TESIS TESIS TESIS

sensibilidad óptica y mejor consumo de potencia en comparación con el dispositivo *FET-OC SG*. Para un dispositivo fototransistor la capacidad de respuesta R_{ph} y la relación señal/ruido (SNR) son dos parámetros clave para caracterizar el desempeño del dispositivo. Contrariamente al diseño propuesto, el diseño *FET-OC SG* convencional presenta bajos valores de la capacidad de respuesta y la relación señal/ruido para el rango de potencia completo. Djeflal y Ferhati eligieron los *MOGAs* porque difieren de la mayoría de las técnicas de optimización ya que su búsqueda global ofrece una población de soluciones en lugar de una sola solución. En la investigación de los autores, cinco funciones objetivo son consideradas: $R_{ph}(X)$, $G(X)$, $I_{ON}/I_{OFF}(X)$, $I_{DMAX}(X)$ y $SN(x)$, donde X representa el vector de variables de diseño que contiene parámetros geométricos y físicos del *FET-OC DTG*, tal que todas las funciones objetivo son de maximización. Los autores seleccionaron cinco puntos de la superficie que representa el frente de Pareto, los cuales representan los casos extremos en el espacio de diseño. Es de resaltar que los resultados obtenidos de la región del frente de Pareto permiten al diseñador obtener un resumen exhaustivo de las compensaciones de diseño, con respecto a las diferentes figuras de mérito del dispositivo. Los autores claramente demostraron que su dispositivo optimizado proporciona un mejor rendimiento en comparación con la estructura *FET-OC SG*, en términos de rendimiento eléctrico y óptico (Djeflal & Ferhati, 2016).

Durante la última década, el *MOSFET* de carburo de silicio 4H (4H-SiC) ha sido ampliamente propuesto para aplicaciones de potencia alta, temperatura alta, y aplicaciones de conmutación. Aunque, varios estudios han tratado con la compensación entre las características eléctricas de un *MOSFET* en 4H-SiC, ninguna investigación en optimización de rendimiento global basada en algoritmos evolutivos ha sido llevada a cabo. En 2019, Bencherif et al. investigaron el diseño optimizado de un *MOSFET* doble implantación 4H-SiC (*DMOSFET*) para una aplicación específica por medio del *MOGA*. Primeramente, los modelos analíticos y numéricos son usados para investigar el comportamiento de corriente-voltaje del *DMOSFET*. Después, estos modelos son asumidos como funciones objetivo en

MOGA. Las técnicas basadas en MOGA permiten la identificación de un conjunto de soluciones optimizadas, comúnmente llamadas “frente de Pareto” (Bencherif, Dehimi, Pezzimenti, De Martino, & Della Corte, 2019).

Los autores utilizaron dos funciones objetivo en MOGA, las cuales son $BV_{DS}(Y)$ (voltaje de ruptura) y $R_{ON}(Y)$ (Resistencia del estado de encendido) tal que Y contiene parámetros geométricos y físicos del *DMOSFET*, $BV_{DS}(Y)$ fue de maximización y $R_{ON}(Y)$ fue de minimización. Únicamente, se implementó el modelo numérico ya que los autores se percataron que convergía más rápido. El frente de Pareto obtenido por MOGA muestra que hay un incremento de $R_{ON}(Y)$ con respecto a un incremento de $BV_{DS}(Y)$. Los resultados del comportamiento de R_{ON} con el diseño propuesto por los autores se compararon con los resultados del *DMOSFET* (sin resultados por algoritmos evolutivos). Se puede observar, para un dispositivo con el mismo valor de BV_{DS} , la optimización basada en MOGA logra una R_{ON} más baja en todo el rango de V_{GS} (Bencherif, Dehimi, Pezzimenti, De Martino, & Della Corte, 2019).

Algoritmos evolutivos para optimización multiobjetivo

El uso de algoritmos evolutivos (*EAs*) para resolver problemas de optimización multiobjetivo (*MOPs*) ha sido principalmente motivado, debido a la naturaleza basada en población de los *EAs* que permite la generación de varios elementos del conjunto óptimo de Pareto en una sola ejecución. Los *MOPs* son atacados hoy usando *EAs* por ingenieros, científicos de la computación, biólogos, e investigadores de operaciones por igual (Coello et al., 2007).

Definiciones

Para desarrollar un entendimiento de *MOPs* y la habilidad de diseñar algoritmos evolutivos multi objetivos (*MOEAs*) para resolverlos, una serie de definiciones formales no ambiguas son requeridas. Estas definiciones permiten un análisis adecuado de las estructuras de los *MOEAs* (Coello et al., 2007).

Para comprender los *MOPs* es necesario comenzar con una breve descripción de los problemas de optimización mono objetivo.

Optimización mono objetivo

El problema de optimización mono objetivo, como se presenta en la Definición 1, continúa siendo abordado por muchas técnicas de búsqueda, incluidos numerosos algoritmos evolutivos (Coello et al., 2007).

Definición 1 (**Problema general de optimización mono objetivo**): se define como la optimización (minimización o maximización) de $f(x)$ sujeto a $g_i(x) \leq 0$, $i = \{1, \dots, m\}$, y $h_j(x) = 0$, $j = \{1, \dots, p\}$ donde x es un vector de variable de decisión n -dimensional $x = (x_1, \dots, x_n)$ de algún universo Ω .

Observe que $g_i(x) \leq 0$ y $h_j(x) = 0$ representan restricciones que deben cumplirse, mientras se optimiza (minimiza o maximiza) $f(x)$. Ω contiene todas las posibles x que se pueden usar para satisfacer una evaluación de $f(x)$ y sus restricciones. Claro que, x puede ser un vector de variables continuas o discretas, así como f puede ser continua o discreta (Coello et al., 2007).

El método para encontrar el óptimo global (puede no ser único) de cualquier función, se denomina Optimización Global. En general, el mínimo global de un problema mono objetivo es presentado en la Definición 2 (Veldhuizen, 1999):

Definición 2 (**Optimización mínima global mono objetivo**): Dada una función $f: \Omega \subseteq R^n \rightarrow R$, $\Omega \neq \emptyset$, para $x \in \Omega$, el valor f^* , tal que $f(x^*) > -\infty$ se llama un mínimo global si y solo si

$$\forall x \in \Omega: f(x^*) \leq f(x).$$

Entonces, x^* es la solución (es) mínima global, f es la función objetivo, y el conjunto Ω es la región factible de x . El objetivo de determinar la (s) solución (es) mínima (s)

global (es) se denomina **problema de optimización global** para un problema mono objetivo.

Aunque, los problemas mono objetivo pueden tener una única solución óptima, los *MOPs* (como regla) presentan un posible conjunto incontable de soluciones. Cuando se evalúa, producen vectores cuyos componentes representan compensaciones en el espacio objetivo. Entonces, el *DM* elige implícitamente una solución aceptable (o soluciones) seleccionando uno o más de estos vectores (Coello et al., 2007).

El problema de optimización multiobjetivo

El problema de optimización multiobjetivo, también, es llamado problema de optimización multicriterio, rendimiento múltiple u optimización vectorial.

Variables de decisión

Las variables de decisión son cantidades numéricas cuyos valores se eligen en un problema de optimización. Estas cantidades se denotan como $x_j, j = 1, 2, \dots, n$.

El vector x de n variables de decisión se representa por:

$$x = [x_1, x_2, \dots, x_n]^T$$

Restricciones

En la mayoría de los problemas de optimización siempre hay restricciones impuestas por las características del entorno o los recursos disponibles (por ejemplo, limitaciones físicas, restricciones de tiempo, etc.). Estas restricciones se deben satisfacer para que una determinada solución sea aceptable. Estas restricciones son expresadas como desigualdades matemáticas (Coello et al., 2007):

$$g_i(x) \leq 0 \quad i = 1, \dots, m$$

o igualdades:

$$h_j(x) = 0 \quad j = 1, \dots, p$$

Funciones objetivo

Para saber que tan “buena” es una cierta solución, es necesario tener algunos criterios para evaluarla. Estos criterios se expresan como funciones computables de las variables de decisión, llamadas funciones objetivo (Coello et al., 2007).

Las funciones objetivo son designadas: $f_1(x), f_2(x), \dots, f_k(x)$, donde k es el número de funciones objetivo en el *MOP* siendo resuelto. Por lo tanto, las funciones objetivo forman un vector función $f(x)$ que se define por (Coello et al., 2007):

$$f(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T$$

El conjunto de todas las n -tuplas de números reales denotado por R^n es llamado n -espacio euclidiano. Se consideran dos espacios euclidianos en los *MOPs* (Coello et al., 2007):

- El espacio n -dimensional de las variables de decisión en el cual cada eje coordenado corresponde a una componente del vector x .
- El espacio k -dimensional de las funciones objetivo en el cual cada eje coordenado corresponde a una componente del vector $f(x)$.

Cada punto en el primer espacio (espacio decisión) representa una solución y da un cierto punto en el segundo espacio (espacio objetivo), que determina una calidad de esta solución en términos de los valores de la función objetivo.

Problema de optimización multiobjetivo

La formulación mono objetivo es extendida para reflejar la naturaleza de los problemas multi objetivos donde no hay una única función objetivo para optimizar, sino muchas. Por lo tanto, los *MOPs* son esos problemas donde el objetivo es optimizar simultáneamente k funciones objetivo, con lo cual, no hay una única solución pero un conjunto de soluciones. Este conjunto de soluciones se encuentra a través del uso de la teoría de optimización de Pareto (Ehrgott, 2005). Tenga en

cuenta que los problemas multi objetivos requieren un tomador de decisiones para hacer una elección de los valores de x_i^* (Coello et al., 2007). Un problema mínimo global de un *MOP* es formalmente definido en Definición 3 (Veldhuizen, 1999).

Definición 3 (*MOP* general): Un ***MOP* general** es definido como minimizar (o maximizar) $F(x) = (f_1(x), \dots, f_k(x))$ sujeto a $g_i(x) \leq 0, i = \{1, \dots, m\}$, y $h_j(x) = 0, j = \{1, \dots, p\}$ $x \in \Omega$. Una solución *MOP* minimiza (o maximiza) los componentes de un vector $F(x)$ donde x es un vector de variables de decisión n-dimensional $x = (x_1, \dots, x_n)$ de algún universo Ω . Se observa que, $g_i(x) \leq 0$ y $h_j(x) = 0$ representan restricciones que deben cumplirse mientras se minimiza (o maximiza) $F(x)$, y Ω contiene todos los posibles x que se pueden utilizar para satisfacer una evaluación de $F(x)$.

Por lo tanto, Un *MOP* consiste en n variables de decisión, $m+p$ restricciones, y k funciones objetivo. Las k funciones objetivo pueden ser lineales o no lineales y continuas o discretas en naturaleza. Además, el vector de variables de decisión puede ser continuo o discreto.

Optimalidad de Pareto

Al tener varias funciones objetivo, el concepto de “óptimo” cambia, porque en *MOPs*, el objetivo es encontrar buenos compromisos (o “compensaciones”) más que una única solución como en la optimización global. La noción de “óptimo” más comúnmente adoptada es el óptimo de Pareto (Coello et al., 2007). La definición formal se proporciona a continuación (Veldhuizen, 1999).

Definición 4 (Óptimo de Pareto): Se dice que una solución $x \in \Omega$ es un **óptimo de Pareto** con respecto a Ω si y solo si no existe $x' \in \Omega$ para el cual $v = F(x') = (f_1(x'), \dots, f_k(x'))$ domina $u = F(x) = (f_1(x), \dots, f_k(x))$. El concepto **óptimo de Pareto** tiene un significado con respecto al espacio entero de las variables de decisión a menos que se especifique lo contrario.

Definición 5 (**Dominancia de Pareto** (Veldhuizen, 1999)): Se dice que un vector $u = (u_1, \dots, u_k)$ domina a otro vector $v = (v_1, \dots, v_k)$ (denotado por $u < v$), si y solo si u es parcialmente menor que v , es decir, $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\}: u_i < v_i$.

Definición 6 (**Conjunto óptimo de Pareto** (Veldhuizen, 1999)): Para un *MOP* determinado, $F(x)$, el **conjunto óptimo de Pareto**, P^* , se define como:

$$P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega F(x') < F(x)\}$$

Las soluciones óptimas de Pareto son aquellas soluciones dentro del espacio de búsqueda del genotipo (espacio decisión) cuyos correspondientes componentes del vector objetivo del fenotipo no pueden ser todos simultáneamente mejorados, este conjunto de soluciones es representado por P^* . Sus vectores correspondientes se denominan no dominados; seleccionar un vector (es) de este conjunto de vectores (El conjunto del frente de Pareto PF^*) indica implícitamente que son soluciones óptimas de Pareto aceptables (Coello et al., 2007).

Definición 7 (**Frente de Pareto** (Veldhuizen, 1999)): Para un *MOP*, $F(x)$, y el conjunto óptimo de Pareto, P^* , el **Frente de Pareto** PF^* se define como:

$$PF^* := \{u = F(x) \mid x \in P^*\}$$

Cuando se grafica en el espacio objetivo, los vectores no dominados son colectivamente conocidos como el **frente de Pareto**. Nuevamente, P^* es un subconjunto de algún conjunto solución. Sus vectores objetivos evaluados forman PF^* , de los cuales cada uno es no dominado con respecto a todos los vectores objetivo. En general, no es fácil encontrar una expresión analítica que exprese una línea o superficie que contenga estos puntos y en la mayoría de los casos, resulta imposible. El procedimiento normal para generar el frente de Pareto es computar muchos puntos en Ω y su correspondiente $f(\Omega)$. Cuando, hay suficientes valores de

estos, entonces, es posible determinar los puntos no dominados y producir el frente de Pareto. Cada solución asociada con un punto en el frente de Pareto es un vector cuyas componentes representan compensaciones en el espacio decisión o en el espacio solución de Pareto. En el dominio computacional, el conjunto solución de Pareto verdadero (denominado P^*) es definido como P_{true} que generalmente es un subconjunto de P^* , y su asociado frente de Pareto se define como PF_{true} (Coello et al., 2007).

Además, el tomador de decisiones (DM) a menudo selecciona soluciones a través de la elección del desempeño de objetivos aceptables, representados por el frente de Pareto. Identificar un conjunto de soluciones optimas de Pareto es la clave para que el DM seleccione una solución “compromiso” que satisfaga los objetivos “lo mejor” posible. En cada generación, el DM típicamente elige pocos puntos del Frente de Pareto. Las soluciones optimas de Pareto asociadas, entonces, son las soluciones de compromiso “aceptables” (por parte del tomador de decisiones).

Conceptos básicos de los algoritmos evolutivos

La computación evolutiva se refiere a sistemas computacionales que resuelven un problema en específico usando modelos de procesos evolutivos naturales, como selección natural, supervivencia del más apto y reproducción, los cuales son los componentes fundamentales de tales sistemas computacionales. Por lo cual, los algoritmos evolutivos (EAs) basados en computación evolutiva están fundamentados en la evolución natural de una población elegida aleatoriamente. Un EA utiliza una búsqueda estocástica a través del espacio para obtener una solución óptima (Engelbrecht, 2007).

En la naturaleza, los individuos tienen ciertas características que influyen en su habilidad para sobrevivir y reproducirse. Estas características son representadas por cadenas largas de información contenidas en los cromosomas de los organismos. Cada cromosoma contiene un gran número de genes, donde un gen es la unidad de herencia (Engelbrecht, 2007). Cada individuo del $MOGA$

representará una solución candidata para encontrar una solución óptima, por ejemplo la máxima ganancia de corriente. Como se mencionó las características de un individuo son representadas por un cromosoma, estas características serán las variables de decisión V_{ds} y A del *MOGA*. A las variables de decisión se les llama genes.

Algo que se debe tomar en cuenta es la representación apropiada de las soluciones candidatas (por ejemplo, los cromosomas). La eficiencia y complejidad del algoritmo de búsqueda depende en gran medida del esquema de representación. Diferentes *EAs* de distintos paradigmas usan diferentes esquemas de representación (Engelbrecht, 2007). Algunas representaciones de las soluciones candidatas son el código binario, el código Gray, números reales, máquinas de estado, etcétera; donde cada representación tiene sus ventajas y desventajas dependiendo de la naturaleza de nuestro problema de optimización. Cuando incrementa la complejidad de nuestra representación, el diseño exitoso de los operadores de cruzamiento y mutación incrementa (Hinterding, 2000), los cuales se definirán más adelante. La elección de la representación de los genes del *MOGA* se basará en el paradigma de Programación C++, en el tipo de datos de las funciones objetivo, en la precisión de la solución óptima, y en la complejidad del *MOGA*.

Los algoritmos evolutivos se basan en la generación de una población inicial (Cierta cantidad de individuos), una función objetivo (*fitness*), operadores de selección, operadores de reproducción y condiciones de paro; el Algoritmo 1 muestra como estos componentes son combinados para formar un *EA* genérico. El primer paso de nuestro *EA* es seleccionar una población inicial, una opción es generar números aleatorios de los genes (V_{ds} y A). Con lo cual, se tendrá una distribución uniforme del espacio de búsqueda. Si se selecciona una población pequeña, no se cubrirá mucho espacio de búsqueda, pero se tendrá menos complejidad computacional. Para evaluar la habilidad de supervivencia de los individuos del *EA* se utilizará una función *fitness* (función objetivo) para determinar qué tan buena es la solución representada por los cromosomas. Por la naturaleza del problema de optimización

la función *fitness* estará basada en el frente de Pareto. El operador de selección debe asegurar que los mejores individuos sobrevivan para la siguiente generación y así sucesivamente, con lo cual aseguraremos la convergencia hacia la solución óptima. Las siguientes generaciones (descendencia) son creadas a través de la aplicación de operadores de cruzamiento y mutación, a los cuales se les llama operadores de reproducción. El operador de cruzamiento combina los genes seleccionados aleatoriamente de dos o más padres para obtener uno o más individuos, y el operador de mutación cambia aleatoriamente los valores de los genes. Antes de mencionar, algunos tipos de operadores de selección, se definirá la presión selectiva, la cual es la velocidad en que la mejor solución ocupara a la población entera por repetidas aplicaciones de los operadores de selección. Un operador con una alta presión selectiva reduce la diversidad en la población más rápidamente que con una baja presión selectiva, que puede dirigir a convergencia prematura de subóptimos. Algunos operadores de selección son: selección aleatoria (baja presión selectiva), selección proporcional (alta presión selectiva), selección de torneo (baja presión selectiva), etcétera. El último paso en nuestro EA será la condición de paro, la cual puede ser un límite de iteraciones o hasta que la población converja a un valor óptimo (Engelbrecht, 2007).

Algoritmo 1 Algoritmo Evolutivo Genérico (Engelbrecht, 2007).

Sea $t = 0$ el contador de generaciones;

Crear e inicializar una población n_x -dimensional, $C(0)$, que consiste en n_s individuos;

Mientras la condición(s) de paro no sea verdadera **hacer**

 Evaluar la función *fitness*, $f(x_i(t))$, de cada individuo, $x_i(t)$;

 Realizar la reproducción para crear la descendencia;

 Seleccionar la nueva población, $C(t + 1)$;

 Avanzar a la nueva generación, es decir, $t = t + 1$;

Fin

Técnicas de clasificación

Ha habido varios intentos para clasificar las muchas técnicas de optimización multi objetivo actualmente en uso. Ante todo, es bastante importante distinguir dos etapas en que la solución de un problema de optimización multiobjetivo se puede dividir: la optimización de varias funciones objetivos involucradas y el proceso de decidir, que tipo de “compensaciones” son apropiadas desde la perspectiva del tomador de decisiones (el llamado proceso de toma de decisiones multicriterio) (Coello et al., 2007).

Una de las más populares clasificaciones de técnicas dentro de la comunidad de investigación de operaciones es (Coello et al., 2007):

1. Articulación de preferencias a priori: Este grupo de técnicas incluyen aquellos enfoques que asumen que ciertos objetivos deseados o cierto preordenamiento de objetivos se puede realizar por el tomador de decisiones antes de la búsqueda (toma decisiones antes de buscar).
2. Articulación de preferencias a posteriori: Estas técnicas no requieren información de preferencia previa del *DM* (busca antes de tomar decisiones).
3. Articulación de preferencia progresiva: Estas técnicas normalmente operan en tres etapas: (1) buscar una solución no dominada, (2) obtener la reacción del *DM* con respecto a esta solución no dominada, y modificar las preferencias de los objetivos en consecuencia, y (3) repetir los dos pasos anteriores hasta que el *DM* este satisfecho o no es posible una mejora adicional. Lo cual significa que integra búsqueda y toma de decisiones.

Enfoques del algoritmo evolutivo MOP

Desde la perspectiva del tomador de decisiones (*DM*), se desea que únicamente “pocas” soluciones estén disponibles para facilitar la decisión. Por lo tanto, se está intentando optimizar una función objetivo vector posiblemente con restricciones resultando en compensaciones entre los múltiples objetivos. Se aspira a que un *MOEA* genere soluciones *MOP* en P_{true} que provean una compensación de rendimiento (efectividad, eficiencia) para objetivos específicos del modelo del sistema (costo / beneficio, restricciones, etc.) que pueden mutuamente estar en

conflicto. Lograr el frente de Pareto exacto de un *MOP* es generalmente bastante difícil. Sin embargo, las aproximaciones razonablemente buenas de PF_{true} son generalmente aceptables dentro del límite de tiempo computacional. Los *MOEAs* por definición intentan encontrar este frente de Pareto aceptable pero aproximado (Coello et al., 2007).

La clasificación de técnicas a priori, a posteriori y progresiva se ha adoptado por las técnicas de solución *MOP* basadas en *MOEA*. Los métodos específicos que los *MOEAs* incorporan en cada clasificación de técnica son (Coello et al., 2007):

- Técnicas a priori: Lexicográfico, combinación de *fitness* lineal, y combinación de *fitness* no lineal.
- Técnicas progresivas: Técnicas progresivas o dirección computacional interactiva.
- Técnicas a posteriori: Muestreo independiente, selección de criterio, selección de agregación, selección basada en Pareto, selección de Pareto basada en rangos y nichos, selección de Pareto basada en deme, selección de Pareto basada en elitismo, y selección híbrida.

Diseño de operadores y objetivos genéricos de MOEA

El concepto de diseño básico del algoritmo es usar la asignación *fitness* basada en Pareto para identificar los vectores no dominados de la población actual de un *MOEA*. Con respecto a esto y a la discusión previa, los cuatro objetivos principales de alto nivel de tales algoritmos para resolver *MOPs* son (Coello et al., 2007):

Objetivo 1. Preservar los puntos no dominados generación tras generación.

Objetivo 2. Progresar o guiar generación tras generación hacia PF_{true} .

Objetivo 3. Generar y mantener diversidad de: puntos en el frente de Pareto generación tras generación y/o soluciones óptimas de Pareto generación tras generación.

Objetivo 4. Proporcionar al tomador de decisiones (*DM*) con un número limitado de puntos generación tras generación.

Por lo tanto, un *MOEA* debería guiar la búsqueda hacia PF_{true} , generar y mantener una diversidad de puntos generación tras generación, y evitar la pérdida de “buenas” soluciones a través del almacenamiento (Coello et al., 2007).

La integración de un *MOP* específico con el software *MOEA* seleccionado requiere una comprensión no únicamente en el dominio del problema, sino también en la implementación de los operadores de *MOEA*. Un espectro de *MOEAs* incluye numerosos operadores que se numeran a continuación de acuerdo con su apoyo de los cuatro objetivos principales de *MOEA* (Coello et al., 2007).

Objetivo 1. Preservar puntos no dominados

- Dominancia basada en rango – asignación *fitness*
- No-Pareto vs enfoques de Pareto
- Almacenamiento + elitismo de la población de cromosoma

Objetivo 2. Progreso hacia los puntos en PF_{true}

- Converger hacia el frente de Pareto computacional, PF_{true}
- Generar puntos de fenotipos no dominados
- Manipulación explícita / no explícita de bloques de construcción
- Modelos probabilísticos de *MOEA*; incorporación de búsqueda local, etc.

Objetivo 3. Mantener diversidad de: puntos en el frente de Pareto generación tras generación y/o soluciones óptimas de Pareto generación tras generación.

- Preservación de diversidad
- *Niching/fitness sharing* y *crowding* en el frente de Pareto (variaciones)
- Uniforme / Diverso no dominada en el frente de Pareto generación tras generación.

Objetivo 4. Proveer al *DM* con un número limitado de puntos del frente de Pareto generación tras generación.

Algoritmo genérico *MOEA*

En general, basado en los objetivos de *MOEA*, un *MOEA* efectivo debería incorporar las siguientes operaciones genéricas en el algoritmo genérico *MOEA*, el cual se puede observar en el Algoritmo 2 (Coello et al., 2007).

Algoritmo 2 Algoritmo genérico *MOEA*.

Inicializar la población P y P^{iv}

Evaluar los valores objetivos $F(x)$ sobre la población

Asignación basada en rango en dominancia de Pareto

Calcular el número de *Niche*

Asignar *Fitness* compartido o *Crowding*

Mientras la condición no termine (número de generaciones u otra)

 Seleccionar “buenos” individuos de $P \rightarrow P^i$

 Cruzar, mutar los individuos en $P^i \rightarrow P^{ii}$

 Evaluar los valores objetivos de la descendencia P^{ii}

 Ranquear $(P^i \cup P^{ii}) \rightarrow P^{iii}$ basada en dominancia de Pareto

 Calcular el número de *Niche*

 Asignar *Fitness* compartido o *Crowding*

 Reducir $P^{iii} \rightarrow P$

 Copiar $P^{iii} \rightarrow P^{iv}$ basado en dominancia de Pareto

Fin

Algoritmo genético multi objetivo

El algoritmo genético multiobjetivo (*MOGA*) se basa en un método de asignación *fitness* basada en rango (Fonseca, 1993). El Algoritmo 3 muestra los diferentes operadores con que cuenta el *MOGA*. Donde, N' es el tamaño de población, g es la generación específica, $f_j(x_k)$ es la j -ésima función objetivo, x_k es el k -ésimo individuo, y P' la población (Coello et al., 2007).

Algoritmo 3 Algoritmo *MOGA*.

1: **Procedimiento** $MOGA(N', g, f_j(x_k)) \triangleright N'$ miembros evolucionaron en g generaciones para resolver $f_j(x_k)$

- 2: Inicializar la Población P'
- 3: Evaluar los valores objetivos
- 4: Asignar rango basado en dominancia de Pareto
- 5: Calcular el número de *Niche*
- 6: Asignar *fitness* escalada linealmente
- 7: Asignar *Fitness* compartido
- 8: **for** $i = 1$ hasta g **hacer**
- 9: Selección mediante muestreo universal estocástico
- 10: Cruzamiento de un solo punto
- 11: Mutación
- 12: Evaluar los valores objetivos
- 13: Asignar rango basado en dominancia de Pareto
- 14: Calcular el tamaño de nicho
- 15: Asignar *fitness* escalada linealmente
- 16: Asignar *Fitness* compartido
- 17: **Fin for**
- 18: **Fin procedimiento**

En el MOGA, el uso de rango ya obliga a la búsqueda a concentrarse únicamente en el óptimo global. Al implementar el *Fitness* compartido en el dominio del espacio objetivo más que en el dominio del espacio decisión, y únicamente entre pares de individuos no dominados, se puede esperar poder evolucionar a una representación distribuida uniformemente de la superficie de compensación global (Fonseca, 1993).

A continuación se da una pequeña descripción de los principales operadores del MOGA.

Fitness escalado linealmente

La asignación tradicional de *fitness* de acuerdo con el rango puede extenderse de la siguiente manera (Fonseca, 1993):

1. Ordenar la población según el rango.

2. Asignar *fitness* a cada individuo interpolando del mejor (rango 1) al peor (rango $n \leq N$, donde N es el tamaño de la población) en la forma habitual, de acuerdo con alguna función, generalmente lineal, pero no necesariamente.
3. Promediar los *fitness* de los individuos con el mismo rango, para que todos ellos sean muestreados a la misma razón. Téngase en cuenta que este procedimiento mantiene constante el *fitness* de la población global mientras mantiene una presión selectiva adecuada, como lo define la función utilizada.

Tamaño de nicho

Para prevenir la deriva genética se utiliza las técnicas convencionales de *fitness* compartido, en optimización de función multimodal. Sin embargo, introducen otro parámetro del algoritmo genético (GA), el tamaño de nicho σ_{share} , que debe establecerse con cuidado. Sobre la convergencia, los óptimos locales son ocupados por un número de individuos proporcional a sus valores *fitness*. Al contrario, la solución global de un MOP es plano en términos de *fitness* individual, y no hay manera de conocer el tamaño del conjunto solución de antemano, en términos de una métrica fenotípica (Fonseca, 1993).

Los tamaños de nicho pueden ser consecuentemente incorporados en el operador de asignación *fitness* escalada linealmente al usarlos para escalar *fitness* individuales dentro de cada rango. La proporción de *fitness* asignada al conjunto de individuos actualmente no dominados como un todo será independiente de sus coeficientes compartidos (Fonseca, 1993).

El parámetro compartido σ_{share} establece que tan separados dos individuos deben estar para que puedan disminuir el *fitness* del otro. Para calcular σ_{share} , Fonseca y Fleming utilizaron propiedades geométricas del Frente de Pareto (Superficie de compensación) y la definición de no-dominancia para solucionar el desconocimiento

previo del conjunto solución en el dominio de las variables de decisión. Por lo cual, obtuvieron la siguiente ecuación polinomial de orden-($q - 1$) (Fonseca, 1993).

$$N\sigma_{share}^{q-1} - \frac{\prod_{i=1}^q (M_i - m_i + \sigma_{share}) - \prod_{i=1}^q (M_i - m_i)}{\sigma_{share}} = 0$$

Para, $\sigma_{share} > 0$.

Donde, q representa el numero de funciones objetivo, cada M_i y m_i son los valores máximos y mínimos de cada i -ésimo objetivo, respectivamente.

Fitness compartido

El concepto compartido se usa para dividir la población en diferentes subpoblaciones de acuerdo con el parecido de los individuos en dos posibles espacios solución: el espacio objetivo y el espacio decisión. El parámetro compartido σ_{share} controla el alcance compartido, y se define una función compartida de ley de potencia $Sh(d)$ como una función de distancia métrica (d) entre dos individuos como sigue (Deb, K. and Goldberg, 1989):

$$Sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^\alpha, & \text{si } d < \sigma_{share}; \\ 0, & \text{si no} \end{cases}$$

Para implementar la idea de compartición (compartido), la recompensa de un individuo se degrada debido a la presencia de otros individuos en su vecindario. Cuando la proximidad de los individuos es definida en el espacio objetivo, es llamado compartición fenotípica (Deb, K. and Goldberg, 1989).

La distancia métrica (d_{ij}) considerada en la compartición fenotípica es la distancia entre cadenas en el espacio objetivo. En general, para una función de p -parámetro, la distancia d_{ij} se puede calcular usando cualquier norma de distancia adecuada en el espacio p -dimensional. Por simplicidad, aquí se adopta la distancia euclidiana en el espacio p -dimensional. Por lo tanto, para los individuos $x_i = [x_{1,i}, x_{2,i}, \dots, x_{p,i}]$ y $x_j = [x_{1,j}, x_{2,j}, \dots, x_{p,j}]$ la métrica d_{ij} se puede calcular como (Deb, K. and Goldberg, 1989).

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{k,i} - x_{k,j})^2},$$

donde $x_{1,i}, x_{2,i}, \dots, x_{p,i}$ son las funciones *fitness*.

Selección

En la selección de la rueda de la ruleta, los valores *fitness* se normalizan (por ejemplo, dividiendo cada *fitness* por el valor *fitness* máximo). La distribución de probabilidad se puede ver como una rueda de ruleta, donde el tamaño de cada segmento es proporcional a la probabilidad de selección normalizada de un individuo (Engelbrecht, 2007).

Cuando la selección de la rueda de la ruleta se usa para crear la descendencia para reemplazar a la población entera, se realizan llamadas independientes al algoritmo de la rueda de la ruleta. Fue encontrado que esto resulta en una alta varianza en el número de descendientes creados por cada individuo. Puede ocurrir que el mejor individuo no se seleccione para producir descendencia durante una generación dada. Para prevenir este problema, el muestreo universal estocástico se utiliza para determinar para cada individuo el número de descendientes, que producirá el individuo con únicamente una llamada al algoritmo del muestreo universal estocástico (Engelbrecht, 2007).

Cruzamiento y mutación

Al resolver problemas que tienen un espacio de búsqueda continua, usando GAs de codificación binaria, las cadenas binarias son normalmente elegidas para la codificación de las variables del problema. Cuando una codificación binaria (discreta) es usada para el espacio de búsqueda continuo, surgen una serie de dificultades: la incapacidad para lograr una precisión arbitraria en la solución obtenida, el mapeo fijo de las variables del problema, el problema inherente de

Hamming cliff asociado con la codificación binaria, y el procesamiento de los esquemas de *Holland* en el espacio de búsqueda continua (Deb & Agrawal, 1995).

Sin embargo, motivados por el éxito de *GAs* binarios en el espacio de búsqueda discreto, y la necesidad para un *GA* de código real eficiente para eliminar los problemas mencionados. Deb y Agrawal desarrollaron el cruzamiento binario simulado (*SBX*) de código real que simplemente simula al operador de cruza de punto único binario. Además, si el efecto del operador de cruza y de la subsecuente mutación se consideran, su potencia de búsqueda combinada se vuelve significativa (Deb & Agrawal, 1995).

Los resultados de simulación mostraron que los *GAs* de código real con el *SBX* pueden superar el problema de *Hamming cliff*, el problema de precisión y el problema de mapeo fijo (Deb & Agrawal, 1995).

Es de resaltar que el problema de optimización que se está tratando es discreto, ya que las reglas de diseño de *Layout* restringen el incremento de L , por ejemplo, para la tecnología de 180nm los incrementos de L serán igual a 90nm (para mayor referencia consultar el tema funciones objetivo del problema de optimización). Debido a que la mutación polinomial y el *SBX* están diseñados para espacios continuos, se utilizó la siguiente estrategia:

Sea c una variable de decisión cruzada o mutada; $\lambda = \frac{L_{min}}{2}$, donde L_{min} representa la longitud mínima del canal del *MOSFET*, y λ representa los incrementos permitidos de L .

Ahora, $k = f_{red}(\frac{c}{\lambda})$, donde f_{red} es una función de redondeo. Entonces, la variable de decisión c equivale a $c = (k)(\lambda)$.

Dominancia de Pareto

Considerando un individuo x_i en la generación t que es dominado por $p_i^{(t)}$ individuos en la población actual. Su posición actual en el rango de los individuos se puede dar por

$$\text{rank}(x_i, t) = 1 + p_i^{(t)}$$

Todos los individuos no dominados son asignados con rango 1 (Fonseca, 1993). Este proceso significa que un individuo con un rango mayor es peor que un individuo con un rango menor.

Elección de los parámetros del MOGA

Antes de realizar la aplicación del MOGA, se probó este algoritmo en una serie de funciones de prueba con restricciones para MOP, las cuales se pueden encontrar en el sitio web <http://www.cs.cinvestav.mx/~emoobook> creado para el libro “Algoritmos evolutivos para resolver problemas multi objetivos”, así mismo el código del MOGA se puede encontrar en este sitio web (Coello et al., 2007). Los únicos cambios que se realizaron al MOGA fueron la utilización del operador de mutación polinomial, el cruzamiento binario simulado (SBX) y el manejo de restricciones del apartado método utilizado en el manejo de restricciones. El MOGA se ejecutó en las funciones de prueba con restricciones para obtener sus parámetros que tengan una mejor distribución de convergencia al frente de Pareto, estos parámetros son la probabilidad de cruza (p_c) y mutación (p_m), el índice de distribución para SBX(n_c), el índice de distribución para la mutación polinomial (n_m), el número de generaciones ($gmax$), el tamaño de la población (m) y el tamaño de las soluciones del frente de Pareto ($tampareto$). En la Figura 5 se muestra el frente de Pareto (PF_{true}) y las soluciones de Pareto (P_{true}) (círculos de color rojo) para MOPs, y el frente de Pareto y las soluciones de Pareto obtenidos con el MOGA (círculos de color azul); y en la Tabla 1 se presenta las formas algebraicas de las funciones objetivo y restricciones de cada función de prueba. Además, el nombre correspondiente a cada función de prueba con restricciones se muestra en la Figura 5 y en su correspondiente ecuación algebraica en la Tabla 1. La arquitectura de

optimización para obtener cada PF_{true} y P_{true} en esta sección de resultados, se ejecutó en el sistema operativo Linux Ubuntu 16.04.5.

En la Figura 5, se observa que tanto las soluciones de Pareto como el frente de Pareto obtenidos con el *MOGA*, tiene una distribución de convergencia aceptable. Aunque en algunos casos no se obtuvo todo el frente de Pareto, por ejemplo, el frente de Pareto de Binh4 y el frente de Pareto de Jimenez; y tampoco se logró cubrir toda la gráfica de soluciones de Pareto en determinados casos, por ejemplo, el conjunto de soluciones de Pareto de Tanaka y el conjunto de soluciones de Osyczka.

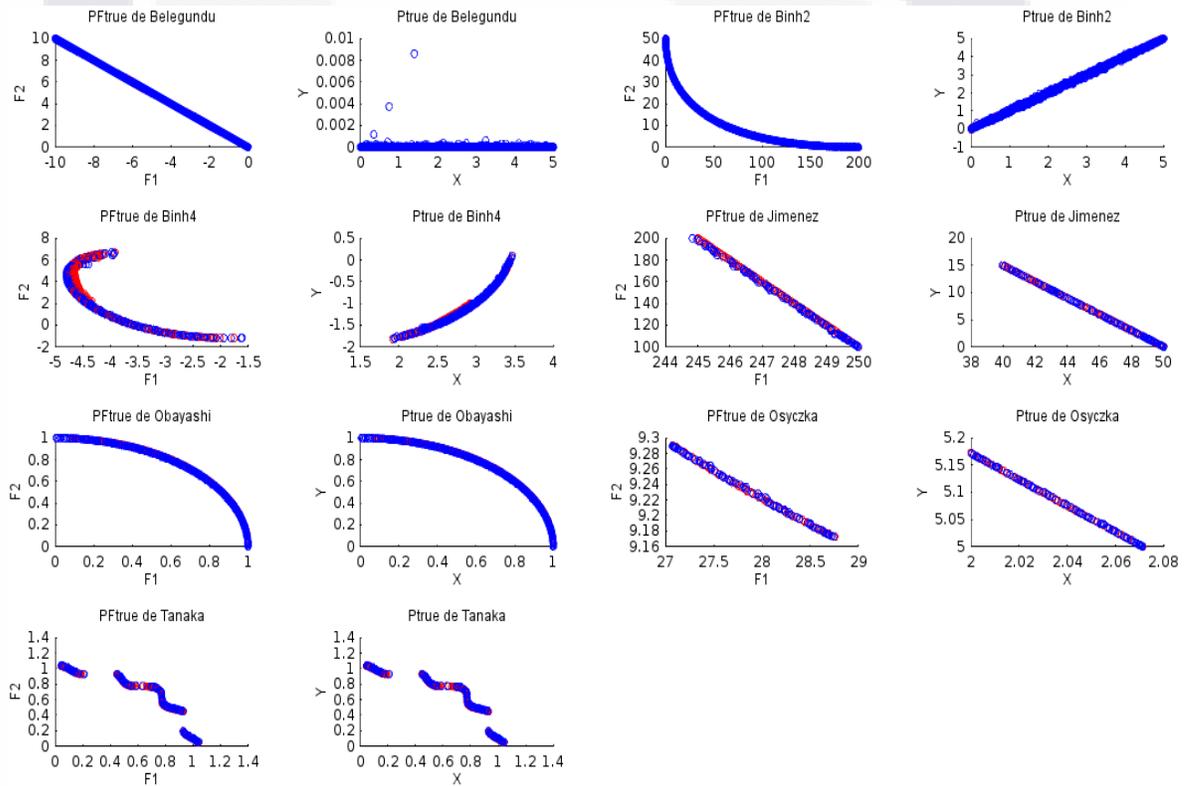


Figura 5 Frente de Pareto(PF_{true}) y soluciones de Pareto(P_{true}) de un conjunto de Funciones de prueba con restricciones para *MOP*.

Todas estas funciones de prueba numérica se optimizaron en el *MOGA* con los siguientes parámetros: $pc = 0.8$, $pm = 0.75$, $n_c = 5$, $n_m = 5$, $gmax = 1000$, $m = 200$ y $tampareto = 1000$.

Tabla 1 Funciones de prueba numérica para MOP (con restricciones laterales).

Nombre	Definición	Restricciones
Belegundu	$F = (f_1(x, y), f_2(x, y))$, donde $f_1(x, y) = -2x + y$, $f_2(x, y) = 2x + y$	$0 \leq x \leq 5, 0 \leq y \leq 3$, $0 \geq -x + y - 1$, $0 \geq x + y - 7$
Binh2	$F = (f_1(x, y), f_2(x, y))$, donde $f_1(x, y) = 4x^2 + 4y^2$, $f_2(x, y) = (x - 5)^2 + (y - 5)^2$	$0 \leq x \leq 5, 0 \leq y \leq 3$, $0 \geq (x - 5)^2 + y^2 - 25$, $0 \geq -(x - 8)^2 - (y + 3)^2 + 7.7$
Binh4	$F = (f_1(x, y), f_2(x, y), f_3(x, y))$, donde $f_1(x, y) = 1.5 - x(1 - y)$, $f_2(x, y) = 2.25 - x(1 - y^2)$, $f_3(x, y) = 2.625 - x(1 - y^3)$	$-10 \leq x, y \leq 10$, $0 \geq -x^2 - (y - .5)^2 + 9$, $0 \geq (x - 1)^2 + (y - 0.5)^2 - 6.25$
Jimenez	Maximizar $F =$ $(f_1(x, y), f_2(x, y))$, donde $f_1(x, y) = 5x + 3y$, $f_2(x, y) = 2x + 8y$	$x, y \geq 0$, $0 \geq x + 4y - 100$, $0 \geq 3x + 2y - 150$, $0 \geq -5x - 3y + 200$, $0 \geq -2x - 8y + 75$
Obayashi	Maximizar $F =$ $(f_1(x, y), f_2(x, y))$, donde $f_1(x, y) = x$, $f_2(x, y) = y$	$0 \leq x, y \leq 1$, $1 \geq x^2 + y^2$
Osyczka	$F = (f_1(x, y), f_2(x, y))$, donde $f_1(x, y) = x + y^2$, $f_2(x, y) = x^2 + y$	$2 \leq x \leq 7, 5 \leq y \leq 10$, $0 \leq 12 - x - y$, $0 \leq x^2 + 10x - y^2 + 16y - 80$
Tanaka	Maximizar $F =$ $(f_1(x, y), f_2(x, y))$, donde $f_1(x, y) = x$, $f_2(x, y) = y$	$0 < x, y \leq \pi$, 0 $\geq -(x^2) - (y^2) + 1$ $+ 0.1 \cos(16 \arctan \frac{x}{y})$ $\frac{1}{2} \geq (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2$

Manejo de restricciones

En la presencia de restricciones, al comparar dos soluciones (valores objetivos), cada solución puede ser factible o infactible. Por lo tanto, puede haber como máximo tres situaciones: 1) ambas soluciones son factibles; 2) una es factible y la otra no; y 3) ambas son infactibles (Deb et al., 2002).

En el ámbito de la optimización multiobjetivo, Se usa una regla simple para cada caso (Deb et al., 2002).

Case 1) Se utiliza asignación de rango basado en dominancia de Pareto.

Case 2) Elegir la solución factible.

Case 3) Elegir la solución con una violación de restricción general más pequeña.

Dado que en ningún caso las restricciones y los valores de la función objetivo se comparan entre sí, no es necesario tener ningún parámetro de penalización, lo que hace que el enfoque de manejo de restricciones propuesto sea útil y atractivo (Deb et al., 2002).

Para mantener la concordancia con el procedimiento del *MOGA* se utiliza la siguiente definición de dominación entre dos soluciones i y j , propuesta por Deb, et al. (Deb et al., 2002).

Definición 8: Se dice que una solución i domina-restricción a una solución j , si cualquiera de las siguientes condiciones es verdadera.

- 1) La solución i es factible y la solución j no lo es.
- 2) Las soluciones i y j son ambas infactibles, pero la solución i tiene una violación de restricción general más pequeña.
- 3) Las soluciones i y j son factibles y la solución i domina a la solución j .

El efecto de usar este principio de dominación-restricción es que cualquier solución factible tiene un mejor rango de no dominación que cualquier solución infactible (Deb et al., 2002).

Este método de manejo de restricciones propuesto por los autores es capaz de mantener soluciones uniformes en la región óptima de Pareto, en una serie de problemas de prueba con restricciones (Deb et al., 2002).



Metodología

Las librerías compartidas del software de simulación Ngspice se utilizaron para obtener los valores de las variables intrínsecas del transistor *MOSFET*, que son degradadas por los efectos de canal corto. Estos valores formaron parte de la metodología para obtener las funciones objetivo y restricciones, las cuales fueron parte de la arquitectura de optimización. Además, se realizó un análisis para reducir la complejidad computacional de la arquitectura de optimización, debido a que se conoce de antemano, que a los simuladores les toma más tiempo en proporcionar los datos de los modelos que los métodos analíticos.

Arquitectura de lectura de datos

Ngspice es un programa de simulación de circuitos electrónicos y eléctricos (*NGSPICE*, n.d.). Los circuitos pueden contener resistencias, capacitores, inductores, fuentes de voltaje y corriente independientes o dependientes, líneas de transmisión con pérdidas y sin pérdidas, interruptores y los cinco dispositivos semiconductores más comunes: diodos, *BJTs*, *JFETs*, *MESFETs* y *MOSFETs* (Holger Vogt, 2018).

Ngspice utiliza los modelos BSIM3, o BSIM4 para modelar los efectos físicos de los *MOSFETs* en las dimensiones submicrométricas. Para modelar I_{ds} , V_t , las corrientes de fuga, las resistencias, las capacitancias, etcétera; el modelo BSIM3, y BSIM4 incluyen como parte del modelo los efectos de canal corto, ya que son importantes en la obtención de la dinámica de los circuitos.

Ngspice puede ser compilado como librería compartida. Esto permite agregar a Ngspice una aplicación que tiene control sobre el simulador, es decir, se puede desarrollar una interfaz que exporte funciones que controlen el simulador y funciones de devolución de llamada para la obtención de datos para el análisis de la respuesta de los circuitos. Entonces, se puede enviar un archivo de entrada con las conexiones de un circuito a Ngspice, empezar la simulación en un proceso

separado, leer los datos de simulación en cada punto de tiempo, detener el simulador dependiendo de alguna condición, alterar los dispositivos o parámetros de modelo y luego reanudar la simulación (Holger Vogt, 2018).

Por lo cual, Ngspice es un software flexible ya que ofrece la posibilidad de desarrollar un algoritmo propio. Se puede resolver algún objetivo específico del desarrollo de un circuito electrónico, se puede ver el comportamiento de las funciones a resolver (voltaje, corriente, capacitancias del *MOSFET*, etcétera) alterando sus variables independientes, entre otros.

Antes de desarrollar la arquitectura de lectura de datos, se realizó el siguiente análisis para determinar los conjuntos de puntos (L, W) que cumplan con la ecuación (1).

Si $W \geq L$, la corriente I_{ds} incrementa con respecto a W y L . Por lo cual, para obtener una corriente I_{ds} grande $W \gg L$ (W mucho mayor a L).

Sea

$$\frac{W}{L} = C \Leftrightarrow W = CL \text{ tal que } C \geq 1 \text{ ya que } W \geq L \tag{2}$$

Entonces, el conjunto de puntos (L, W) se encuentran sobre una recta con pendiente C y ordenada al origen cero. Por lo tanto, para cada valor de C se obtiene un conjunto de puntos diferentes. A la pendiente C se le llamará Factor de escala, por cuestiones de nomenclatura de los transistores *MOSFETs*.

De lo anterior si las longitudes W y L son pequeñas, se tendrá un transistor con un área pequeña y, la corriente I_{ds} no deberá mostrar cambio alguno de acuerdo con las ecuaciones (1) y (2). La suposición anterior no se cumple debido a que los transistores de tecnologías submicrométricas se encuentran fuertemente influenciados por los efectos de canal corto, obteniendo corrientes I_{ds} pequeñas para dimensiones W y L pequeñas.

Ya que la comparación entre los transistores debería mostrar las mismas características eléctricas (I_{ds} ideal, R_{ds} ideal, corrientes de fuga ideales, capacitancias ideales). Entonces, se optimizará sobre un único Factor de escala C , tal que C es la pendiente de la recta de la ecuación (2).

Tomando en cuenta las ventajas que ofrece Ngspice y el análisis de W y L , se desarrolló una arquitectura en el entorno de programación C++ ya que el código de librerías compartidas de Ngspice es nativo de C. Esta arquitectura se desarrolló para la lectura de datos de las variables dependientes (funciones) e independientes.

Las variables dependientes son:

- I_{ds} : corriente entre las terminales Drenaje y Fuente.
- R_{ds} : resistencia entre las terminales Drenaje y Fuente.
- g_m : la transconductancia de salida.
- V_i : el voltaje de disparo.
- I_{bd} : la corriente de fuga entre las terminales Substrato y Drenaje.
- I_{bs} : la corriente de fuga entre las terminales Substrato y Fuente.
- C_{gd} : la capacitancia entre las terminales Compuerta y Drenaje.
- C_{gs} : la capacitancia entre las terminales Compuerta y Fuente.
- C_{ds} : la capacitancia entre las terminales Drenaje y Fuente.
- C_{bg} : la capacitancia entre las terminales Substrato y Compuerta.
- C_{bs} : la capacitancia entre las terminales Substrato y Fuente.
- C_{bd} : la capacitancia entre las terminales Substrato y Drenaje.
- C_{gg} : la capacitancia entre las terminales Compuerta y Compuerta.
- C_{dg} : la capacitancia entre las terminales Drenaje y Compuerta.
- C_{dd} : la capacitancia entre las terminales Drenaje y Drenaje.
- $capbs$: la capacitancia de diodo entre las terminales Substrato y Fuente.
- $capbd$: la capacitancia de diodo entre las terminales Substrato y Drenaje.

Y, las variables independientes son:

- W : el ancho del canal del transistor *MOSFET*.
- L : la longitud del canal del transistor *MOSFET*.

- V_{ds} : el voltaje entre las terminales Drenaje y Fuente.
- V_{gs} : el voltaje entre las terminales Compuerta y Fuente.

Estas variables son los parámetros considerados por el modelo BSIM3 para *MOSFETs* de tecnología 180nm. Además, es importante establecer que el modelo BSIM4 considera parámetros adicionales para transistores con longitudes de canal más cortos, por ejemplo 130nm.

Los valores de las variables se obtuvieron del circuito mostrado en la Figura 6, en el cual V_{ds} varió de 0 V a 1.8 V con incrementos de .003V (ver justificación en el tema complejidad computacional de las funciones objetivo y restricciones), V_{gs} se mantuvo constante a .8V, y W y L variaron sobre varias rectas descritas por la ecuación (2); el rango de V_{ds} se eligió para cumplir con las condiciones del modo de saturación y por los parámetros de la tecnología de 180nm, e igualmente V_{gs} . En el tema arquitectura de software del problema de optimización existe un apartado llamado arquitectura del simulador ngspice, en el cual viene de una manera más detallada la comunicación que se debe establecer con las librerías compartidas de ngspice. Al obtener los datos de las variables dependientes e independientes, se guardaron en archivos de textos para graficarlos en GNU Octave, y poder ver su comportamiento. Debido a lo anterior, se tiene mayor posibilidad de desarrollar una metodología flexible, exacta, precisa y con una complejidad computacional mínima que encuentre los óptimos de estas variables que son degradadas por los efectos de canal corto.

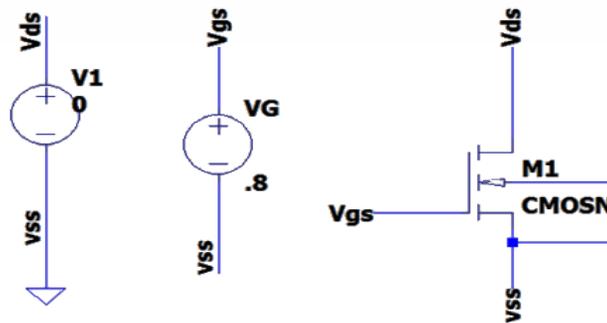


Figura 6 Circuito con un transistor *nMOS*.

Tanto Ngspice como GNU Octave se ejecutaron en el sistema operativo Linux Ubuntu 16.04.5. Aprovechando las librerías compartidas de Ngspice se realizó una comunicación entre la arquitectura descrita en C++ y estas librerías de Ngspice.

A continuación, se muestran las gráficas de algunas variables dependientes.

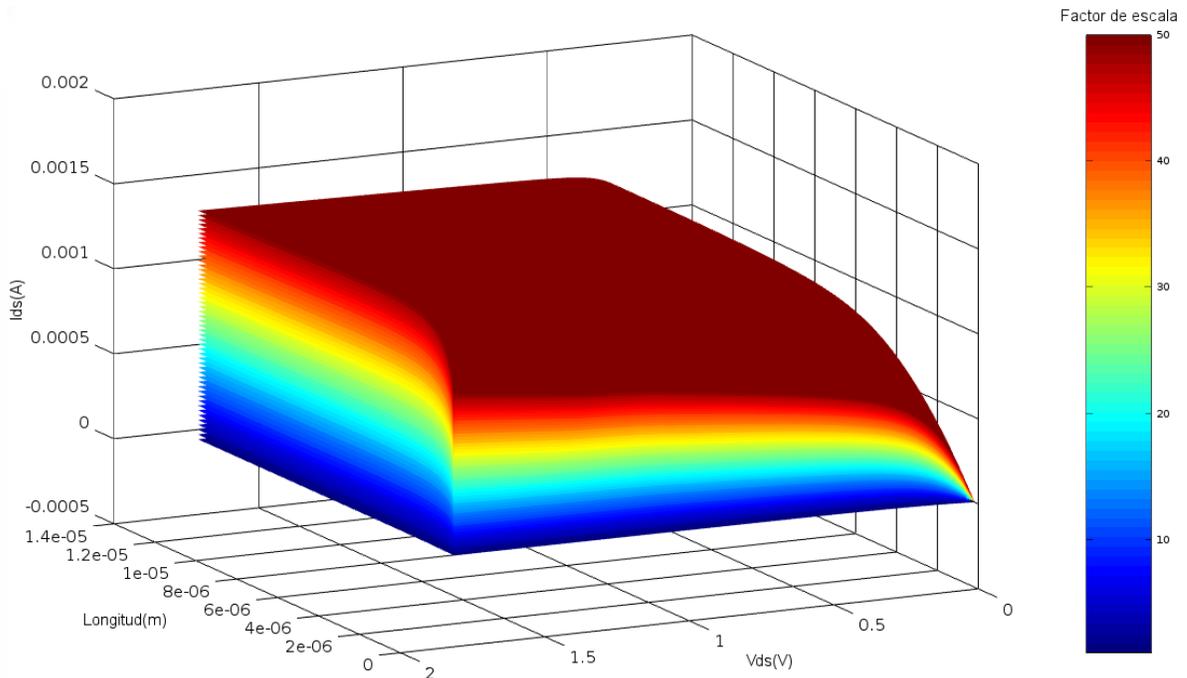


Figura 7 Variación de la corriente I_{ds} con respecto a L , V_{ds} y V_{gs} .

En la Figura 7 cada color (Factor de escala C) representa la variación de I_{ds} con respecto a V_{ds} , V_{gs} y el conjunto de puntos (W, L) sobre una recta descrita por la ecuación (2). Para cada color se puede observar que I_{ds} aumenta con respecto a V_{ds} , también con respecto a la longitud del canal L . Además, se puede observar que al aumentar el Factor de escala C ($W \gg L$) aumenta I_{ds} en cuanto a magnitud y a razón de cambio. Asimismo, en la Figura 8 la función de la transconductancia de salida g_m tiene el mismo comportamiento que I_{ds} .

En la Figura 9 cada color (Factor de escala C) representa la variación de C_{bg} con respecto a V_{ds} , V_{gs} y el conjunto de puntos (W, L) sobre una recta descrita por la ecuación (2). Para cada color se puede observar que C_{bg} aumenta con respecto a

V_{ds} , también con respecto a la longitud L a partir de aproximadamente $2 \mu\text{m}$. Además, se puede observar que al aumentar el Factor de escala C ($W \gg L$) aumenta C_{bg} en cuanto a magnitud y a razón de cambio.

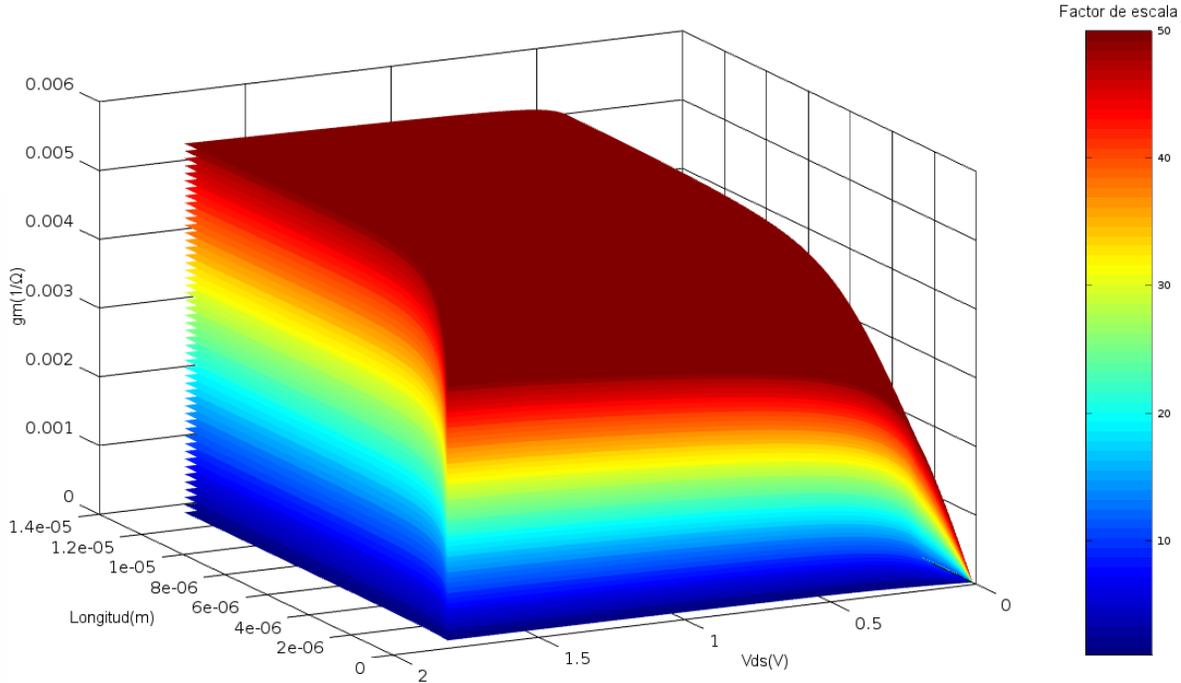


Figura 8 Variación de la transconductancia de salida g_m con respecto a L , V_{ds} y V_{gs} .

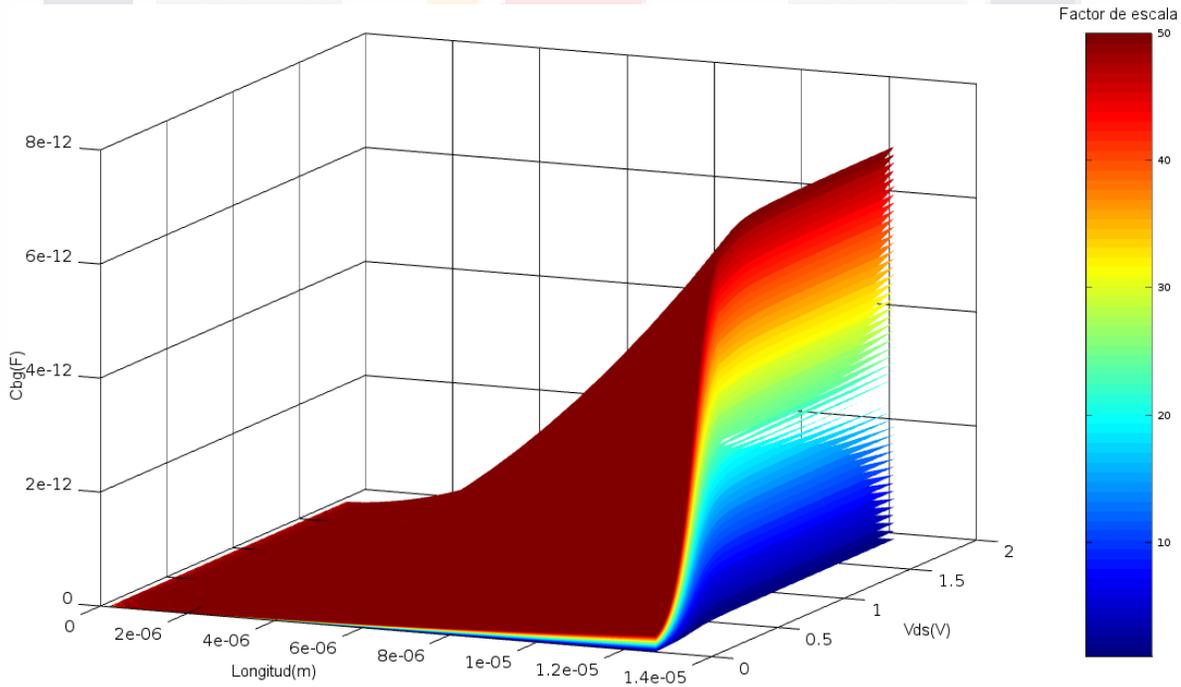


Figura 9 Variación de la capacitancia C_{bg} con respecto a L , V_{ds} y V_{gs} .

Es de resaltar, que en esta investigación durante el proceso de optimización del transistor *MOSFET* el Factor de escala C se mantendrá constante, lo cual ya se había mencionado anteriormente. Se optó por agregar las Figuras 7, 8 y 9, en las cuales se cuenta con diferentes valores de C para poder apreciar el comportamiento del *MOSFET* con estos valores, y tomarlo como una futura investigación donde el proceso de optimización contará con diferentes magnitudes de C .

Funciones objetivo del problema de optimización

Considerando a partir de saturación ($V_{ds} > V_{gs} - V_t$) que la corriente I_{ds} debe ser grande y su pendiente cero o constante (con una razón de cambio mínima), así como la transconductancia de salida g_m debe ser de magnitud grande y constante, la capacitancia C_{bg} debe ser pequeña; de acuerdo con las Figuras 7, 8 y 9 existen regiones en el plano R4 donde las funciones tienden a los valores óptimos. En estas Figuras, se puede observar que al incrementar la longitud (L), la corriente I_{ds} aumenta, la transconductancia de salida g_m aumenta y la capacitancia C_{bg} aumenta. Tomando en cuenta que I_{ds} es de maximización, g_m es de maximización, C_{bg} es de minimización y L es de minimización, una manera intuitiva sería encontrar en el proceso de optimización una longitud mínima (L_{min}) y una longitud máxima (L_{max}), tal que $L = [L_{min}, L_{max}]$. Esto es debido a que estas variables dependientes están en conflicto, por ejemplo el máximo de I_{ds} se encuentra en el máximo de L . Además, en este análisis no se tomaron en cuenta todas las variables, las cuales también son variables conflictivas.

Las variables dependientes causan la degradación del desempeño del transistor *MOSFET* en sus 4 principales características, las cuales son tamaño del circuito integrado, consumo de potencia, ganancia de V/I y ancho de banda; estas variables formarán parte de las funciones objetivo de nuestra metodología de optimización. Todas las funciones objetivo se evaluarán a partir de saturación ya que el enfoque de optimización se analizará y diseñará para circuitos analógicos.

Como se mencionó anteriormente en el apartado de arquitectura de lectura de datos, la variable $I_{ds}(L, W, V_{ds}, V_{gs})$ proviene de datos de simulación. La variable V_{ds} empieza en un voltaje inicial (V_{ini}) y va incrementando (Inc) hasta un voltaje final (V_{fin}). Por otro lado, la variable V_{gs} se mantiene constante; y, W y L se van incrementando de manera proporcional para que se mantenga constante C .

La manera más sencilla de cuantificar la ganancia de corriente es comparar los valores de I_{ds} en cada punto de V_{ds} para cada señal de corriente (Diferentes valores de W y L).

Sea I_{ds_1} una señal de corriente de un *MOSFET* con dimensiones W_1 y L_1 tal que el subíndice 1 indica que la señal de corriente 1 depende de estas dimensiones, e I_{ds_2} una señal de corriente de un *MOSFET* con dimensiones W_2 y L_2 , el subíndice 2 tiene el mismo significado. Ahora, supóngase que $Inc = .003V$, $V_{ini} = .6V$ (Punto de saturación), $V_{fin} = 1.8V$ e $I_{ds_2}(L_2, W_2, V_{ds_{200}}, V_{gs}) > I_{ds_1}(L_1, W_1, V_{ds_{200}}, V_{gs})$ en el punto $V_{ds_{200}} = .6V$.

Donde,

$$V_{ds_j} = V_{ds_{j-1}} + inc. \tag{3}$$

Con $j = 1, 2, 3, \dots, n$.

Tal que,

$$n = \frac{V_{fin}}{inc}. \tag{4}$$

Sustituyendo, $n = \frac{V_{fin}}{inc} = \frac{1.8V}{.003V} = 600$; lo cual implica que I_{ds_2} en el punto $V_{ds_{200}}$ tiene mayor valor de corriente que I_{ds_1} en el punto $V_{ds_{200}}$.

De aquí en adelante, se supondrá que $j = 200$ ($j = \frac{V_{ini}}{inc} = \frac{.6V}{.003V} = 200$) es el punto de saturación, pero teniendo en cuenta que en el proceso de optimización este podría cambiar.

Con lo cual, si se suman todos los valores de corriente en cada punto de V_{ds_j} a partir de saturación, se obtendría una cuantificación valida, ya que si

$$I_{ds_2}(V_{ds_{200}}) \geq I_{ds_1}(V_{ds_{200}}), I_{ds_2}(V_{ds_{201}}) \geq I_{ds_1}(V_{ds_{201}}), I_{ds_2}(V_{ds_{202}}) \geq I_{ds_1}(V_{ds_{202}}), \dots, I_{ds_2}(V_{ds_n}) \geq I_{ds_1}(V_{ds_n}).$$

Ahora sumando cada valor de corriente,

$$G_2 = I_{ds_2}(V_{ds_{200}}) + I_{ds_2}(V_{ds_{201}}) + I_{ds_2}(V_{ds_{202}}) + \dots + I_{ds_2}(V_{ds_n}),$$

$$G_1 = I_{ds_1}(V_{ds_{200}}) + I_{ds_1}(V_{ds_{201}}) + I_{ds_1}(V_{ds_{202}}) + \dots + I_{ds_1}(V_{ds_n}).$$

Donde, G_1 es la ganancia de corriente de la señal I_{ds_1} y G_2 es la ganancia de corriente de la señal I_{ds_2} . Por lo tanto, $G_2 \geq G_1$; pero, existe el caso donde no se cumple que $I_{ds_2}(V_{ds_j}) \geq I_{ds_1}(V_{ds_j})$, en este caso por ley de tricotomía se cumple una y solo una de las siguientes igualdades $G_2 = G_1, G_2 > G_1, G_2 < G_1$. Con lo cual, se pueden comparar las ganancias de corriente de las señales I_{ds} .

Se debe tomar en cuenta que únicamente se considerarán valores positivos de corriente, ya que el signo negativo en la corriente eléctrica únicamente indica que el flujo de cargas toma la dirección contraria que la corriente positiva. Por ejemplo, por ley de tricotomía es correcto $-2A > -5A$, pero por el flujo de cargas es correcto $-2A < -5A$, lo cual contradice la ley de tricotomía. Para solucionar esto, únicamente se tomarán valores de corriente positivo ($2A < 5A$).

Debido a que un objetivo de la tesis es reducir la complejidad computacional de la metodología de optimización, se realizaron pruebas para obtener la ganancia de corriente en un solo punto de V_{ds_j} . Tal que,

$$j = \left(\frac{V_{ini}}{inc}\right) + \left(\frac{V_{fin}}{inc} - \frac{V_{ini}}{inc}\right) * p \tag{5}$$

Donde, p es una variable para elegir en que voltaje (V_{ds_j}) a partir de saturación se comparan las ganancias de corriente. Por ejemplo, supóngase que $Inc = .003V$, $V_{ini} = .6V$ (Punto de saturación), $V_{fin} = 1.8V$ y $p = .5$. Con lo cual, $j = \left(\frac{.6v}{.003v}\right) + \left(\frac{1.8v}{.003v} - \frac{.6v}{.003v}\right) * .5 = 400$; donde las ganancias de corrientes se definen como: $G_2 = I_{ds_2}(V_{ds_{400}})$ y $G_1 = I_{ds_1}(V_{ds_{400}})$. Por lo tanto, con este método se reduce la complejidad computacional para obtener la ganancia de corriente. En el apartado de complejidad computacional de las funciones objetivo y restricciones, se analiza este tema con más detalle.

Otro aspecto importante que se debe tomar en cuenta son las reglas de diseño de *Layout*, las cuales describen cuan pequeñas pueden ser las características y cuan cerca pueden empaquetarse de manera confiable en un proceso de manufactura particular de un circuito. Mead and Conway (Mead, 1978) popularizaron las reglas de diseño escalable basadas en un único parámetro, λ , que caracteriza la resolución del proceso. λ generalmente es la mitad de la longitud mínima del canal del *MOSFET*, esta longitud es la distancia entre las terminales de Fuente y Drenaje del transistor (Neil H. E. Weste, 2011). Por lo cual, los incrementos de L serán igual a $L = L + \lambda$, por ejemplo, para la tecnología de 180nm los incrementos de L serán igual a 90nm.

Para obtener una única variable de L y W , se sabe que $W = CL$ y el área del transistor es $A = WL \Leftrightarrow A = CL^2$. Con la única condición de que el área será exclusiva de un L y W , ya que para diferentes L y W con la misma área se obtienen diferentes señales de corriente de acuerdo con los efectos de canal corto. Además, está garantizado que el área pertenece únicamente a un L y W ; ya que C permanece constante. Lo cual se demuestra a continuación.

Con lo establecido para W y A ; si crece L , entonces crece A y W , lo cual implica que para un L y W existe una única A .

Suponiendo que L crece, esto es $0 < L_1 < L_2 < L_3 \dots$, entonces sustituyendo las longitudes en la formula del área, se obtiene $0 < CL_1^2 < CL_2^2 < CL_3^2 \dots$, y en W , $0 < CL_1 < CL_2 < CL_3 \dots$. Por lo tanto, el área pertenece únicamente a un L y W .

Este análisis se realizó únicamente para simplificar la obtención de las funciones objetivo, puesto que en la arquitectura de lectura de datos de ngspice para el proceso de optimización en realidad se alteran tanto L como W , claro con la condición $W = CL$. También, para simplificar las funciones objetivo, el voltaje V_{gs} se omitió de las variables independientes ya que permanece constante durante la simulación de ngspice.

La función objetivo de I_{ds} es encontrar la máxima ganancia de corriente, y con esto obtener circuitos analógicos con ganancias aceptables. Ahora, obteniendo la función objetivo de I_{ds} de manera formal, sea $\Delta I_{ds}(V_{dsj}, A_i) = I_{ds}(V_{dsj}, A_i)$.

Donde,

$i = 1, 2, 3, \dots, m$ (cada i es un área que pertenece únicamente a un L y W), y j adquiere el valor de la ecuación (5).

Tal que, los ΔI_{ds} en un único punto de V_{ds} y diferentes áreas son:

$$\Delta I_{ds}(V_{dsj}, A_1)$$

$$\Delta I_{ds}(V_{dsj}, A_2)$$

.

.

.

$$\Delta I_{ds}(V_{dsj}, A_m)$$

Por lo tanto las ganancias de corriente de las señales I_{ds} son:

$$G_{I_{ds1}} = \Delta I_{ds}(V_{dsj}, A_1)$$

$$G_{I_{ds2}} = \Delta I_{ds}(V_{dsj}, A_2)$$

.

.

.

$$G_{I_{ds_m}} = \Delta I_{ds} (V_{ds_j}, A_m) \tag{6}$$

Tal que $G_{I_{ds_1}}, G_{I_{ds_2}}, \dots, G_{I_{ds_m}}$ corresponden a las ganancias de corriente de las diferentes señales de corriente I_{ds} . Además, como se había mencionado por ley de tricotomía estas ganancias se pueden comparar.

Los efectos de canal corto principalmente la degradación de la velocidad de los portadores, y la movilidad de los portadores afectan la corriente I_{ds} principalmente en cuanto a magnitud. Esto es más notorio en transistores con área pequeña. Con lo cual, la función objetivo con respecto a I_{ds} y al sistema de ecuaciones (6) deberá cumplir.

$$G_{I_{ds}} (V_{ds_j}, A_i) = \Delta I_{ds} (V_{ds_j}, A_i) \tag{7}$$

donde $G_{I_{ds}} (V_{ds_j}, A_i)$ es la función objetivo de ganancia de corriente la cual es de maximización. Es de resaltar que, con esta función objetivo se estaría obteniendo la longitud máxima (L_{max}).

Como se comentó anteriormente con valores grandes de R_{ds} se obtiene una ganancia de DC mejor en el circuito analógico. En consecuencia, la función objetivo con respecto a R_{ds} es la siguiente.

$$G_{R_{ds}} (V_{ds_j}, A_i) = \Delta R_{ds} (V_{ds_j}, A_i) \tag{8}$$

donde $G_{R_{ds}} (V_{ds_j}, A_i)$ es la función objetivo de ganancia de resistencia de drenaje a fuente la cual es de maximización. También, se estaría obteniendo la longitud máxima (L_{max}). El proceso para obtener esta función de ganancia es análogo al proceso de la función de ganancia de corriente (ecuación (7)).

Asimismo, los valores de la magnitud de g_m deben ser grandes para obtener una ganancia de DC mejor. La $g_m(A, V_{ds})$ tiene un comportamiento similar a $I_{ds}(A, V_{ds})$, como lo describe la ecuación (9) y este comportamiento se puede observar en las Figuras 7 y 8; tal que el V_{gs} permanece constante durante el proceso de optimización, y el V_t no varía demasiado. Entonces, se excluirá a la transconductancia de salida g_m del conjunto de funciones objetivo.

$$g_m = \frac{2I_{ds}}{V_{gs} - V_t} \tag{9}$$

Como se mencionó anteriormente, el voltaje de disparo V_t no varía demasiado, lo cual significa que no tiene cambios significativos con respecto a las longitudes W y L del transistor $MOSFET$. Por lo tanto, no se tomará en cuenta en la arquitectura de optimización.

Tanto, las capacitancias, como las corrientes de fuga se consideran como restricciones en el proceso de optimización. Ya que, se está tratando de obtener el menor número de funciones objetivo, para asegurar resultados admisibles en la metodología de optimización y un tiempo de ejecución aceptable. Por las mismas razones, la transconductancia de salida g_m y el voltaje de disparo V_t no se considerarán en el proceso de optimización. En el siguiente tema se establecen las restricciones de manera formal.

Restricciones del problema de optimización

Como ya se había mencionado las variables dependientes (modelo BSIM3) para las corrientes de fuga son I_{bd} e I_{bs} ; y para las capacitancias son C_{gd} , C_{gs} , C_{ds} , C_{bg} , C_{bs} , C_{bd} , C_{gg} , C_{dg} , C_{dd} , $capbs$ y $capbd$. Recuérdese, que las funciones objetivo son definidas a partir de estas variables dependientes. Se puede observar que es demasiado el número de variables dependientes para las corrientes de fuga y para las capacitancias, y este número crece en el modelo BSIM4. Con lo cual, serian demasiadas las funciones objetivo ya que para cada variable existe una función

objetivo. Por lo tanto, la metodología de optimización sería muy complicada, tanto en complejidad computacional como en el desarrollo analítico de la metodología (proceso de optimización), una manera de evitar esto es disminuir el número de variables y con esto el número de funciones objetivo.

Debido a que, los modelos BSIM3 y BSIM4 de las capacitancias son muy complejos, y además algunas capacitancias varían al cambiar sus longitudes, y otras al cambiar el voltaje en sus terminales. Por lo tanto, es muy complicado obtener una capacitancia equivalente (una sola capacitancia) de los modelos BSIM3 y BSIM4. Una manera intuitiva de resolver esto es sumar las magnitudes de las capacitancias a lo largo de V_{ds} , como si se estuviera obteniendo un área aproximada, y después comparar estas sumas para determinar que capacitancia es más dominante en cuanto a magnitud a lo largo de A . La razón por la cual se utilizó la suma de las magnitudes en lugar de la integral discreta se justifica en el apartado de complejidad computacional de las funciones objetivo y restricciones.

Con lo cual, obteniendo la restricción de las capacitancias de manera formal, sea $\Delta C_{gd}(V_{ds_j}, A_i) = C_{gd}(V_{ds_j}, A_i)$.

Donde, $i = 1, 2, 3, \dots, m$ (cada i es un área que pertenece únicamente a un L y W), $V_{ds_{200}}$ (punto de saturación), y V_{ds_j} toma los valores de la ecuación (3).

Por lo tanto las sumatorias de las magnitudes de la capacitancia C_{gd} son

$$G_{C_{gd_1}} = \Delta C_{gd}(V_{ds_{200}}, A_1) + \Delta C_{gd}(V_{ds_{201}}, A_1) + \dots + \Delta C_{gd}(V_{ds_n}, A_1)$$

$$G_{C_{gd_2}} = \Delta C_{gd}(V_{ds_{200}}, A_2) + \Delta C_{gd}(V_{ds_{201}}, A_2) + \dots + \Delta C_{gd}(V_{ds_n}, A_2)$$

·
·
·

$$G_{C_{gd_m}} = \Delta C_{gd}(V_{ds_{200}}, A_m) + \Delta C_{gd}(V_{ds_{201}}, A_m) + \dots + \Delta C_{gd}(V_{ds_n}, A_m) \tag{10}$$

Con lo cual, de acuerdo con el sistema de ecuaciones (10), las sumatorias de las magnitudes de la capacitancia C_{gd} deberán cumplir.

$$G_{C_{gd}}(V_{ds}, A_i) = \sum_{j=200}^n \Delta C_{gd}(V_{ds_j}, A_i) \tag{11}$$

La obtención de $G_{C_{gs}}(V_{ds}, A_i)$ y de las capacitancias restantes se obtienen de manera equivalente a la ecuación (11).

Las variables dependientes son funciones convergentes, en algunos casos en análisis *DC* no convergen debido generalmente a un error al especificar las conexiones del circuito (Holger Vogt, 2018). Por lo tanto, la sucesión de la sumatoria de la ecuación (11) es sumable ya que las funciones convergen (Spivak, 1996).

Ahora, las sumatorias de las magnitudes de las capacitancias de la ecuación (11) son una cuantificación de ganancia. Con lo cual, estas sumatorias son magnitudes discretas a lo largo de A_i , y al sumar estas sumatorias ($G_{C_{gd}}(V_{ds}, A_i)$), las sumas de estas sumatorias se comparan para determinar que capacitancia es de mayor magnitud a lo largo de A .

Con lo cual, para la capacitancia C_{gd} se obtiene la siguiente sumatoria

$$G_{C_{gd}}(A) = \sum_{i=1}^m G_{C_{gd}}(V_{ds}, A_i) \tag{12}$$

También, La obtención de $G_{C_{gs}}(A)$ y de las capacitancias restantes se obtienen de manera equivalente a la ecuación (12). Por lo tanto, al comparar $G_{C_{gd}}(A)$, $G_{C_{gs}}(A)$, y las demás capacitancias, se podrá determinar que capacitancia es más dominante en cuanto a magnitud a lo largo de A . Por ejemplo, suponiendo que $G_{C_{gd}}(A) < G_{C_{gs}}(A) < \dots < G_{capbd}(A)$, con lo cual $G_{capbd}(A)$ sería la capacitancia dominante.

Al obtener esta capacitancia dominante, se aplica un proceso análogo al de la ecuación (7). Esto es debido a la complejidad computacional. Esta capacitancia dominante pasaría a ser parte de la restricción de las capacitancias, como se muestra en la ecuación (13).

$$G_{capbd}(V_{dsj}, A_i) \leq Cap \quad (13)$$

Tal que Cap es una magnitud de capacitancia, la cual es elegida por el usuario dependiendo del comportamiento deseado del circuito analógico.

Esto es debido a que, las capacitancias varían debido a las longitudes del transistor y al voltaje en sus terminales. El principal efecto de las capacitancias es la variación de la frecuencia de operación en un circuito analógico, si las capacitancias son grandes disminuye la frecuencia. La ecuación (13) significa que tanto $capbd$, como las demás capacitancias deben ser menores o iguales a Cap , y con esto asegurar una frecuencia de operación aceptable. Con esta restricción se estaría obteniendo L_{max} .

Las dos corrientes de fuga fluyen de un punto en común (Substrato) hacia dos puntos diferentes, los cuales son la terminal de Drenaje y Fuente. Debido a esto, se puede aplicar la ley de corrientes de Kirchhoff para obtener una corriente de fuga total del substrato ($I_{fuga_tot_b} = I_{bd} + I_{bs}$), y con esto se tendría una única variable para las corrientes de fuga. En la referencia (Rjoub et al., 2014) también se utilizó la ley de corrientes de Kirchhoff para obtener la corriente de fuga de Compuerta total, para comparar los modelos de corriente de fuga de Compuerta propuestos por Rjoub, et al. con otros modelos, incluyendo el modelo BSIM4. Aunque, la corriente de fuga total se utilizó para comparación en una simulación, se puede aplicar esta misma analogía, pero en lugar de comparar, se optimizará. Con lo cual, la restricción para las corrientes de fuga sería.

$$G_{I_{fuga_tot_b}}(V_{dsj}, A_i) \leq I_{fug} \quad (14)$$

Asimismo, después de obtener la corriente de fuga total del substrato ($I_{fuga_tot_b} = I_{bd} + I_{bs}$), se aplica un proceso equivalente a la ecuación (7).

En el modelo BSIM4 existen otras corrientes de fuga, las cuales son:

La corriente de fuga entre la terminal de Compuerta y la región de traslape de la Fuente (I_{gs}).

La corriente de fuga entre la terminal de Compuerta y la región de traslape del Drenaje (I_{gd}).

La corriente de fuga entre las terminales de Compuerta y Substrato (I_{gb}).

La corriente de fuga entre la terminal de Compuerta y el canal invertido de Fuente (I_{gcs}).

La corriente de fuga entre la terminal de Compuerta y el canal invertido de Drenaje (I_{gcd}).

También, por la ley de corrientes de Kirchhoff se obtiene una corriente de fuga total de la Compuerta ($I_{fuga_tot_g} = I_{gs} + I_{gd} + I_{gb} + I_{gcs} + I_{gcd}$). Después, se aplica un desarrollo análogo a la ecuación (13), por ejemplo $G_{I_{fuga_tot_b}}(A) < G_{I_{fuga_tot_g}}(A)$. Con lo cual, la corriente de fuga total de la Compuerta sería parte de la restricción de las corrientes de fuga, como se muestra en la ecuación (15).

$$G_{I_{fuga_tot_g}}(V_{dsj}, A_i) \leq I_{fug} \tag{15}$$

Donde I_{fug} es una magnitud de corriente, la cual es también seleccionada por el usuario dependiendo de los criterios de diseño del circuito analógico. Esto es debido a que, uno de los mayores efectos de canal corto son las corrientes de fuga las cuales se mueven por diferentes partes del transistor. Esto afecta al consumo de potencia del circuito analógico, y también disminuye la ganancia de voltaje afectando el ancho de banda. La ecuación (15) significa que todas las corrientes de fuga deben ser menores o iguales a I_{fug} , y con esto asegurar un consumo de potencia y un ancho de banda aceptables. Con esta restricción se estaría adquiriendo L_{max} .

Como se había mencionado los efectos de canal corto, como la degradación de la velocidad de los portadores y la movilidad de los portadores afectan la corriente I_{ds} principalmente en cuanto a magnitud. Esto significa que al aumentar A estos efectos de canal corto van disminuyendo la degradación de I_{ds} , y al disminuir A aumenta la degradación de I_{ds} . Por lo tanto, una manera factible de obtener una restricción para este análisis es la siguiente.

$$limite_{I_{ds}} = por_{I_{ds}} \left(\frac{Max_{I_{ds}}}{100\%} \right) \quad (16)$$

Tal que $limite_{I_{ds}}$ es la restricción para disminuir estos efectos de canal corto, $Max_{I_{ds}}$ es el máximo de $G_{I_{ds}}(V_{dsj}, A_i)$ de la ecuación (7), $por_{I_{ds}}$ es ingresado por el usuario dependiendo de los criterios de diseño del circuito analógico. Por lo tanto, la restricción sería.

$$G_{I_{ds}}(V_{dsj}, A_i) \geq limite_{I_{ds}} \quad (17)$$

Es de resaltar que con esta restricción se estaría adquiriendo la longitud mínima (L_{min}).

Uno de los objetivos de la metodología de optimización es obtener una ganancia de corriente I_{ds} aceptable con respecto al área del transistor(A), lo cual significa que cuando existan razones de cambio pequeñas a medida que crece el área no será factible aumentar el área. La derivada numérica para $G_{I_{ds}}(V_{dsj}, A_i)$ de la ecuación (7) cumple con estas condiciones. Con la única observación, que en realidad en el proceso de optimización se estaría obteniendo la derivada parcial numérica, ya que se toman en cuenta todas las variables independientes. Por lo cual, se utiliza la definición de derivada parcial numérica aplicada directamente a I_{ds} . Además, recuérdese que I_{ds} es equivalente a $G_{I_{ds}}(V_{dsj}, A_i)$ de la ecuación (7).

$$\frac{\partial I_{ds}(V_{dsj}, A_{i+1})}{\partial A} \cong \frac{I_{ds}(V_{dsj}, A_{i+1}) - I_{ds}(V_{dsj}, A_i)}{A_{i+1} - A_i} \quad (18)$$

Tal que, j es igual a la ecuación (5) e $i = 1, 2, 3, \dots, m$.

Se utilizan únicamente las variables independientes A y V_{ds} por las razones que se dieron en el apartado de funciones objetivo.

La derivada parcial numérica únicamente es una aproximación de la derivada parcial por la derecha ($\lim_{\Delta A \rightarrow 0^+}$). Con lo cual, $\Delta A > 0$; esto se cumple para todo L y W . Ya que, para diferentes L se cumple que $0 < L_1 < L_2 < L_3 \dots < L_{m+1}$, sustituyendo las longitudes en la fórmula del área, se obtiene $0 < CL_1^2 < CL_2^2 < CL_3^2 \dots < CL_{m+1}^2$. En la ecuación (18), la i significa que incrementa la variable de área, en consecuencia $A_2 - A_1 = CL_2^2 - CL_1^2$, $A_3 - A_2 = CL_3^2 - CL_2^2$, ... , $A_{m+1} - A_m = CL_{m+1}^2 - CL_m^2$. Por lo tanto, $\Delta A > 0$ para todo L y W .

Puesto que, la razón de cambio de la ecuación (18) va decreciendo al aumentar el área del *MOSFET*. Esto es debido a la degradación de la velocidad de los portadores y a la movilidad de los portadores. Es decir, al incrementar A va disminuyendo la ganancia progresivamente, y con esto no tendría sentido aumentar el A si no existen ganancias significativas. Ahora, obteniendo la pendiente de la variable de la ecuación (15).

$$\theta = \tan^{-1} \frac{\partial I_{ds}(V_{dsj}, A_{i+1})}{\partial A} \quad (19)$$

Con lo cual, la restricción sería.

$$\theta \geq \text{ang} \quad (20)$$

Tal que *ang* es una magnitud de ángulo, la cual es elegida por el usuario dependiendo del comportamiento deseado del circuito analógico. Con esta restricción se obtendría L_{max} .

Las razones por la cual únicamente se establecieron las restricciones (17) y (20) con respecto a la función objetivo de ganancia de $I_{ds}(A, V_{ds})$, y no con respecto a las funciones de ganancia de $g_m(A, V_{ds})$ y $R_{ds}(A, V_{ds})$ son que, la función objetivo de $g_m(A, V_{ds})$ tiene un comportamiento similar a la función objetivo de $I_{ds}(A, V_{ds})$, como lo describe la ecuación (9); y también, la función objetivo de ganancia de $R_{ds}(A, V_{ds})$, como lo describe la ecuación (21). Además, este comportamiento similar con respecto a las restricciones (17) y (20), entre estas funciones objetivo se puede observar en la Figura 10. Estas funciones objetivo se obtuvieron de datos de simulación de un MOSFET de tipo Bulk de tecnología 180nm utilizando el software de simulación Ngspice.

$$R_{ds} = \frac{V_{ds_{j+1}} - V_{ds_j}}{I_{ds_{j+1}} - I_{ds_j}} \tag{21}$$

con $j = 1, 2, 3, \dots, n$ y para cada incremento de j , incrementa el voltaje V_{ds} y la corriente I_{ds} .

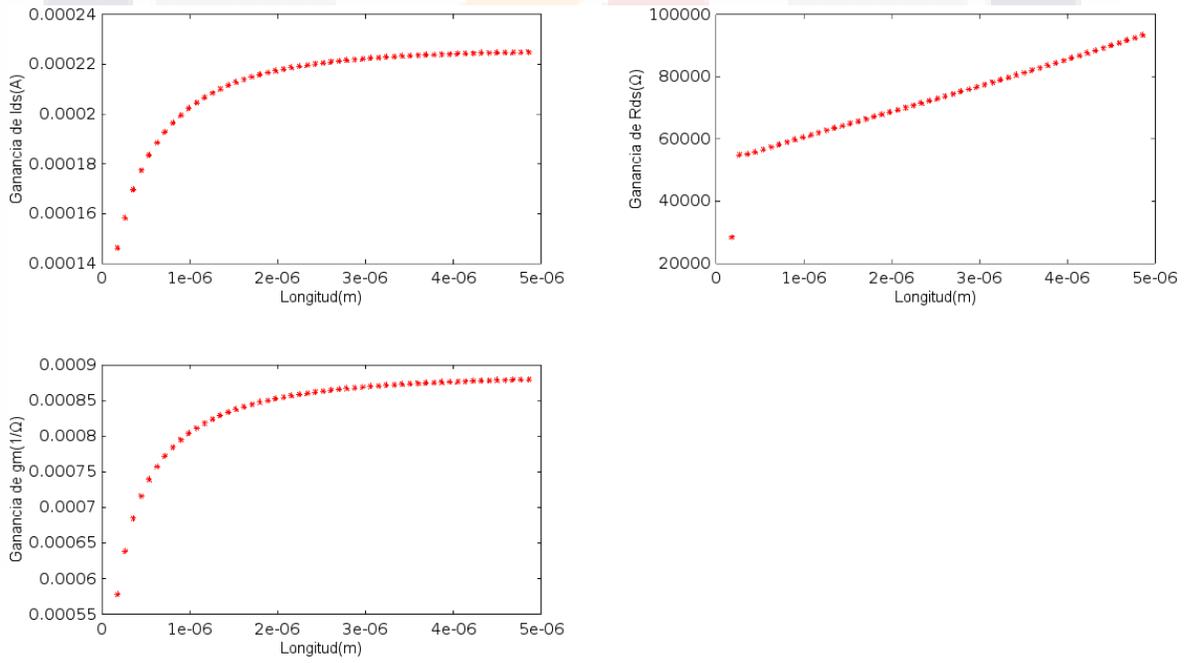


Figura 10 Funciones objetivo de ganancia de I_{ds} , g_m y R_{ds} .

Aunque, no se establecieron restricciones tales como las desigualdades (17) y (20) para $g_m(A, V_{ds})$ y $R_{ds}(A, V_{ds})$, existe una restricción la cual es necesaria para

$R_{ds}(A, V_{ds})$. Esto es debido a que las ecuaciones (9) y (21) no forman parte de los modelos BSIM3 y BSIM4. A continuación, se analiza este caso.

Para asegurar una ganancia uniforme la razón de cambio de I_{ds} con respecto a V_{ds} tiene que ser constante, y tender a cero. La función que cumple con estas características es la línea recta con pendiente cero, la cual se definirá como la corriente ideal de I_{ds} .

Definición provisional de corriente ideal I_{ds} .

Sea, $i = 1, 2, 3, \dots, m$ (cada i es un área que pertenece únicamente a un L y W), $V_{ds_{200}}$ (punto de saturación), y V_{ds_j} toma los valores de la ecuación (3); tal que $I_{ds_{ideal}}(V_{ds_j}, A_i) = I_{ds}(V_{ds_{200}}, A_i)$. Lo cual significa que $I_{ds_{ideal}}(V_{ds_j}, A_i)$ es una recta con pendiente cero a partir de saturación.

La corriente I_{ds} debe ser igual a la corriente ideal I_{ds} , esto es $I_{ds}(V_{ds_j}, A_i) = I_{ds_{ideal}}(V_{ds_j}, A_i)$. Con lo cual, un procedimiento es encontrar un error para determinar la similitud entre estas dos funciones (I_{ds} e $I_{ds_{ideal}}$).

Dado que, se va a determinar el error entre dos funciones (para cada valor de las variables independientes existe un único valor de la variable dependiente), el método que mejor se adecua para encontrar el error en cada punto es el error absoluto. Como lo describe la siguiente ecuación.

$$E_a = \sum_{j=200}^n |I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_j}, A_i)| \quad (22)$$

Para normalizar el error se utiliza la siguiente estrategia. Suponiendo que para todo j , $I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_j}, A_i) \leq 0$; pero, como se están utilizando datos de simulación, y por lo cual no existe una comprobación analítica que asegure esto. Se

utilizará el Lema 1 en la arquitectura de optimización para cuando $I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_j}, A_i) > 0$.

Se sabe que,

$$|a| = |-a| \tag{23}$$

Lema 1.

Si $I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_j}, A_i) > 0$, entonces por la ecuación (23)

$$\begin{aligned} |I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_j}, A_i)| &= |I_{ds}(V_{ds_j}, A_i) - I_{ds_{ideal}}(V_{ds_j}, A_i)| = \\ |I_{ds_{ideal}}(V_{ds_j}, A_i) - [-I_{ds}(V_{ds_j}, A_i) + 2 * I_{ds_{ideal}}(V_{ds_j}, A_i)]| \end{aligned}$$

Como se puede asegurar que $I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_j}, A_i) \leq 0$ o $I_{ds_{ideal}}(V_{ds_j}, A_i) - [-I_{ds}(V_{ds_j}, A_i) + 2 * I_{ds_{ideal}}(V_{ds_j}, A_i)] \leq 0$ para todo V_{ds_j} a partir de saturación. Por simplificación, únicamente se utiliza $I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_j}, A_i) \leq 0$, entonces

$$I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_{200}}, A_i) + \dots + I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_n}, A_i) \leq 0.$$

Con lo cual,

$$\begin{aligned} |(n - 201) * I_{ds_{ideal}}(V_{ds_j}, A_i) - [I_{ds}(V_{ds_{200}}, A_i) + \dots + I_{ds}(V_{ds_n}, A_i)]| &= \\ -[(n - 201) * I_{ds_{ideal}}(V_{ds_j}, A_i) - [I_{ds}(V_{ds_{200}}, A_i) + \dots + I_{ds}(V_{ds_n}, A_i)]] &= \\ -(I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_{200}}, A_i)) + \dots + -(I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_n}, A_i)) &= \\ |I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_{200}}, A_i)| + \dots + |I_{ds_{ideal}}(V_{ds_j}, A_i) - I_{ds}(V_{ds_n}, A_i)| \end{aligned} \tag{24}$$

La ecuación (24) y (22) son equivalentes. Ahora, la diferencia de la ecuación (24) se normaliza a $(n - 201) * I_{ds_{ideal}}(V_{ds_j}, A_i)$, para obtener el error relativo de I_{ds} .

$$E_{r_{I_{ds}}} = \frac{|(n - 201) * I_{ds_{ideal}}(V_{ds_j}, A_i) - [I_{ds}(V_{ds_{200}}, A_i) + \dots + I_{ds}(V_{ds_n}, A_i)]|}{(n - 201) * I_{ds_{ideal}}(V_{ds_j}, A_i)} \tag{25}$$

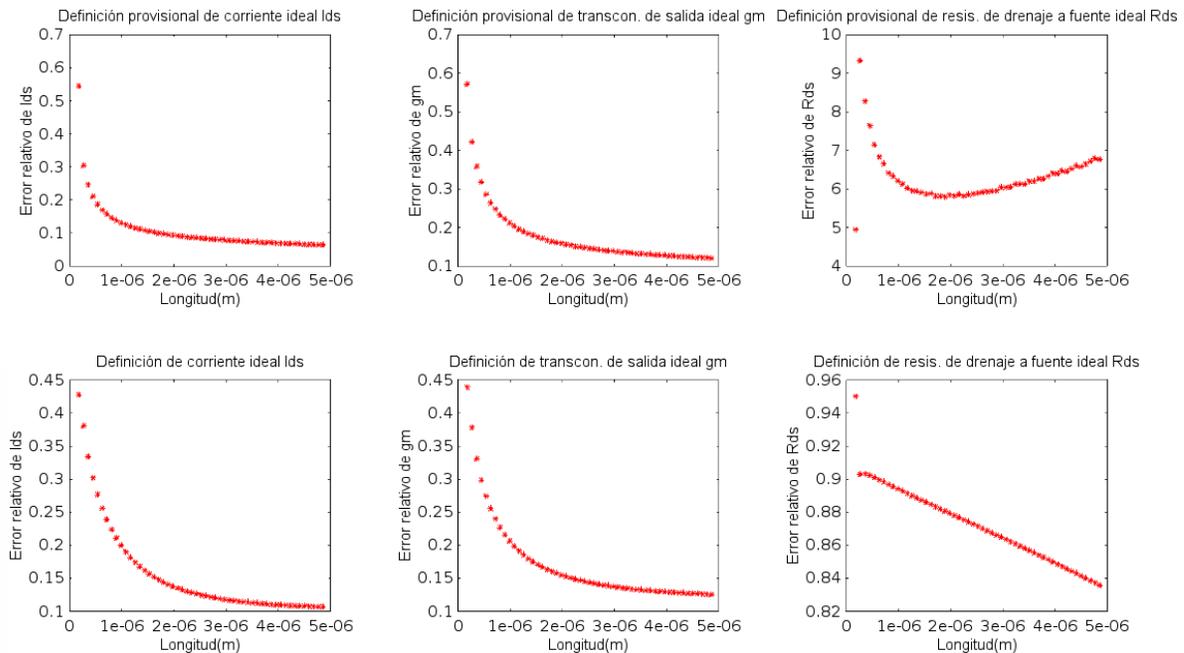


Figura 11 Error relativo de I_{ds} , g_m y R_{ds} .

Por el comportamiento natural del transistor *MOSFET* de tipo *Bulk*, a medida que crece el A las funciones de $I_{ds}(A, V_{ds})$, $g_m(A, V_{ds})$ y $R_{ds}(A, V_{ds})$ son más idénticas a la función ideal, por ejemplo $I_{ds}(V_{ds_j}, A_i)$ sería aproximadamente igual $I_{ds_{ideal}}(V_{ds_j}, A_i)$. El problema de esta normalización (ecuación (25)) es que no se cumple para $R_{ds}(A, V_{ds})$, como lo muestra la Figura 11; con título, definición provisional de resistencia de drenaje a fuente ideal R_{ds} . También, Estos errores relativos se obtuvieron de datos de simulación de un *MOSFET* de tipo *Bulk* de tecnología 180nm con el simulador ngspice.

Para solucionar este problema la corriente ideal I_{ds} es igual a una cota máxima, lo cual significa que ninguna señal de corriente I_{ds} sobrepasa esta cota. Dando una definición.

Definición de corriente ideal I_{ds} .

Sea, $i = 1, 2, 3, \dots, m$ (cada i es un área que pertenece únicamente a un L y W), $V_{ds_{200}}$ (punto de saturación), y V_{ds_j} toma los valores de la ecuación (3) ; tal que

$I_{ds_{ideal}}(V_{ds_j}, A_i) = L$, donde L es una cota máxima. Lo cual significa que $I_{ds_{ideal}}(V_{ds_j}, A_i)$ es una recta con pendiente cero a partir de saturación.

Con lo cual, aplicando esta nueva definición de corriente ideal a la ecuación (25), se soluciona el problema, como lo muestra la Figura 11; con título, definición de resistencia de drenaje a fuente ideal R_{ds} . Por lo tanto, la restricción sería.

$$E_{r_{I_{ds}}} \leq E_{I_{ds}} \tag{26}$$

Tal que $E_{I_{ds}}$ es ingresado por el usuario dependiendo del comportamiento deseado del circuito analógico. Con esta restricción se adquiriría L_{max} .

Debido al BIBL, a la modulación de la longitud del canal y a los impactos de ionización; la resistencia R_{ds} varía a lo largo de V_{ds} . Con lo cual, la ganancia de DC de un circuito analógico no es constante. Por lo tanto, la restricción para R_{ds} sería.

$$E_{r_{R_{ds}}} \leq E_{R_{ds}} \tag{27}$$

Donde, la desigualdad (27) se obtiene de manera análoga a la desigualdad (26). El $E_{R_{ds}}$ es también ingresado por el usuario dependiendo de los criterios de diseño del circuito analógico. Con esta restricción se adquiriría L_{max} .

A continuación, se muestra el conjunto de funciones objetivo y restricciones que se obtuvieron para el proceso de optimización, para obtener el rango $L = [L_{min}, L_{max}]$ del transistor.

Funciones objetivo	Ecuación	L
$G_{I_{ds}}(V_{ds_j}, A_i) = \Delta I_{ds}(V_{ds_j}, A_i)$	(7)	L_{max}
$G_{R_{ds}}(V_{ds_j}, A_i) = \Delta R_{ds}(V_{ds_j}, A_i)$	(8)	L_{max}
Restricciones	Ecuación	L
$G_{capbd}(V_{ds_j}, A_i) \leq Cap$	(13)	L_{max}

$$G_{I_{fuga_tot_g}}(V_{dsj}, A_i) \leq I_{fug} \quad (15) \quad L_{max}$$

$$G_{I_{ds}}(V_{dsj}, A_i) \geq \limite_{I_{ds}} \quad (17) \quad L_{min}$$

$$\theta \geq \text{ang} \quad (20) \quad L_{max}$$

$$E_{r_I_{ds}} \leq E_{I_{ds}} \quad (26) \quad L_{max}$$

$$E_{r_R_{ds}} \leq E_{R_{ds}} \quad (27) \quad L_{max}$$

Error de truncamiento para los puntos discretos

El tipo de simulación que se está utilizando es llamada simulación transitoria, en la cual el voltaje V_{ds} tiene un voltaje inicial (V_{ini}) y va incrementando (Inc) hasta un voltaje final (V_{fin}). Por lo cual, se cuenta con puntos discretos (puntos asociados con datos), lo cual significa que para cada incremento de V_{ds} las variables dependientes tendrán un valor. Recuérdese que la cantidad de puntos discretos es $n = \frac{V_{fin}}{inc}$ (ecuación (4)), tal que los incrementos de V_{ds} (Inc) son igualmente espaciados. Ya que V_{fin} se mantiene constante durante la simulación, entonces, cuando Inc es pequeño, n tiende a infinito, y al contrario si Inc es grande, n es pequeño. El problema es elegir el tamaño de Inc , ya que, si Inc es muy pequeño la complejidad computacional aumentara; pero, si Inc es muy grande no se tomarían en cuenta las suficientes magnitudes de las variables dependientes para obtener las restricciones (13), (15), (26) y (27). Esto significa que, cuanto más pequeño sea Inc mayor será la exactitud al obtener las restricciones, pero aumentara la complejidad computacional. Antes de obtener el error de truncamiento para poder elegir el tamaño de Inc , se efectuó el siguiente análisis.

Cuando se tiene que estimar valores intermedios entre puntos asociados con datos, el método más común que se usa para este propósito es la interpolación polinomial. Dados $n + 1$ puntos asociados con datos, hay uno y sólo un polinomio de grado n que pasa a través de todos los puntos. La interpolación polinomial consiste en determinar el polinomio único de n -ésimo grado que se ajuste a $n + 1$ puntos

asociados con datos. Con lo cual, este polinomio proporciona una fórmula para calcular valores intermedios (Canale, 2015).

Durante la simulación la variable *Inc* es constante. Por lo tanto, únicamente se utilizará la interpolación con datos igualmente espaciados y en orden ascendente.

La siguiente ecuación se conoce como *fórmula hacia delante de Newton-Gregory*, la cual es un método que proporciona el polinomio exclusivo para datos igualmente espaciados (Canale, 2015).

$$f_n(x) = f(x_0) + \Delta f(x_0)\alpha + \frac{\Delta^2 f(x_0)}{2!}\alpha(\alpha - 1) + \dots + \frac{\Delta^n f(x_0)}{n!}\alpha(\alpha - 1) \dots (\alpha - n + 1) + R_n \quad (28)$$

donde

$$R_n = \frac{f^{(n+1)}(\xi)}{(n + 1)!} h^{n+1} \alpha(\alpha - 1)(\alpha - 2) \dots (\alpha - n) \quad (29)$$

Tal que R_n es el error de truncamiento de la ecuación (28), h es equivalente a *inc*, $f^{(n+1)}$ es la $(n + 1)$ – ésima derivada, ξ esta en alguna parte del intervalo que contiene la incógnita y los datos, n es el grado del polinomio, $\Delta^n f(x_0)$ es la n – ésima diferencia hacia adelante y $\alpha = \frac{x-x_0}{h}$; donde, x es la incógnita y x_0 es el primer punto discreto.

La ecuación (29) es la única que se va a utilizar. Puesto que, únicamente se desea determinar el error de truncamiento cuando h incrementa o decrece.

Las restricciones (13), (15), (26) y (27) únicamente utilizan los puntos discretos de sus respectivas variables dependientes. En las restricciones (13) y (15) se realiza una sumatoria de los puntos discretos, la cual es una aproximación de un área bajo la curva; y en las (26) y (27) se realiza una resta de los puntos discretos entre un variable ideal y una variable dependiente del *MOSFET*, también, es una

aproximación de un área entre dos curvas. Con lo cual, existe una estrecha analogía entre las restricciones (13), (15), (26) y (27) y la integral numérica.

La regla del trapecio es la primera de las fórmulas cerradas de integración numérica de *Newton-Cotes*. Una manera de obtener la regla del trapecio consiste integrar el polinomio de interpolación de primer grado hacia adelante de *Newton-Gregory*. Este polinomio se obtiene de la ecuación (28), y es el siguiente (Canale, 2015).

$$f_1(x) = f(x_0) + \Delta f(x_0)\alpha + \frac{f^{(2)}(\xi)}{2}h^2\alpha(\alpha - 1) \quad (30)$$

donde, el error de truncamiento equivale a

$$R_1 = \frac{f^{(2)}(\xi)}{2}h^2\alpha(\alpha - 1) \quad (31)$$

Al integrar la ecuación (30) se obtiene la regla del trapecio con su respectivo error de truncamiento para una sola aplicación (El área entre dos puntos discretos). La regla del trapecio de aplicación múltiple utiliza $n + 1$ puntos discretos, lo cual equivale a sumar el área bajo la curva de n segmentos. Para obtener el error de truncamiento de la regla del trapecio de aplicación múltiple, se suman los errores individuales de cada segmento, es decir se suman los errores de la regla del trapecio para una sola aplicación (Canale, 2015).

Dado que existe una rigurosa analogía entre las restricciones (13), (15), (26) y (27) y la integral numérica. Entonces, se utiliza el método para obtener el error de truncamiento de la regla del trapecio de aplicación múltiple.

Ahora, el error de truncamiento para las restricciones en un solo segmento es igual a la ecuación (31). Ya que, no se conoce la variable dependiente porque proviene de datos de simulación, entonces, no se puede obtener la segunda derivada de la ecuación (31). Por fortuna, se puede utilizar una diferencia dividida finita para aproximar la segunda derivada (Canale, 2015). Por lo tanto la ecuación (31) es aproximadamente igual a

$$R_1 \cong \frac{f(x_2) - 2f(x_1) + f(x_0)}{2h^2} h^2 \alpha (\alpha - 1) \tag{32}$$

Por lo tanto, el error de truncamiento para las restricciones en varios segmentos es igual a la sumatoria de los errores individuales de cada segmento.

$$E_a = \sum_{i=1}^n \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{2h^2} h^2 \alpha_i (\alpha_i - 1) \tag{33}$$

Donde i corresponde a cada segmento, y E_a significa que es un error aproximado. Para tener una mejor visualización de la ecuación (33), observe la Figura 12. Por ejemplo, para obtener el error de truncamiento para dos segmentos (de x_0 a x_2) la ecuación (33) es igual a

$$E_a = \frac{f(x_2) - 2f(x_1) + f(x_0)}{2h^2} h^2 \alpha_1 (\alpha_1 - 1) + \frac{f(x_3) - 2f(x_2) + f(x_1)}{2h^2} h^2 \alpha_2 (\alpha_2 - 1) \tag{34}$$

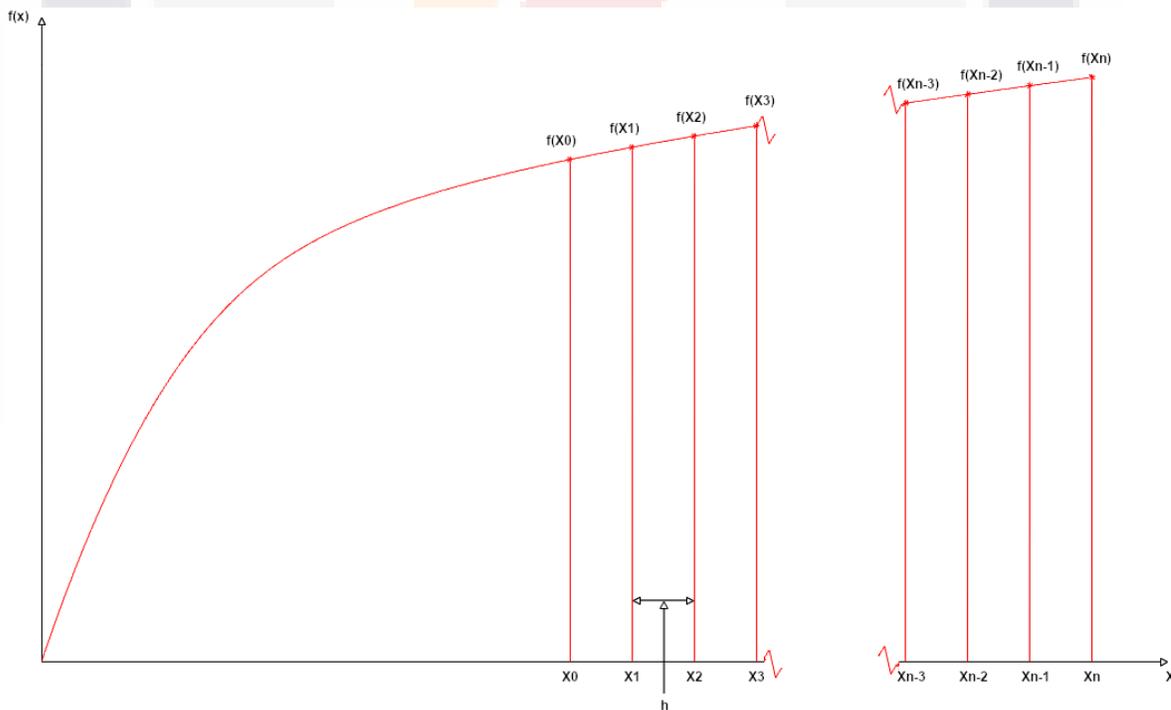


Figura 12 Formato general y nomenclatura para el error de truncamiento en varios segmentos.

Para reducir el error aproximado mucho más rápidamente, los puntos discretos deberían estar centrados alrededor, y tan cerca como sea posible de las incógnitas (Canale, 2015). Esta es la razón por lo cual se utilizó el error de truncamiento de un polinomio de interpolación de primer grado hacia adelante de *Newton-Gregory*. Por lo tanto, la incógnita es igual a

$$x^i = x_{i-1} + \frac{x_i - x_{i-1}}{2} = \frac{x_{i-1} + x_i}{2} \quad (35)$$

Donde x^i es la incógnita i , mas no un punto discreto; la incógnita siempre está a la mitad de cada segmento i , por ejemplo a la mitad de x_0 a x_1 . Entonces, α_i es igual a

$$\alpha_i = \frac{x^i - x_{i-1}}{h} = \frac{\frac{x_{i-1} + x_i}{2} - x_{i-1}}{h} = \frac{x_i - x_{i-1}}{2h} = \frac{h}{2h} = \frac{1}{2} \quad (36)$$

Para probar que $\alpha_i = \frac{1}{2}$ para todos los números naturales se demuestra con el principio de inducción matemática.

A) $P(1)$.

$$\alpha_1 = \frac{x_1 - x_0}{2h} = \frac{h}{2h} = \frac{1}{2}$$

$P(1)$ es verdadera.

B) Si $\alpha_k = \frac{x_k - x_{k-1}}{2h} = \frac{h}{2h} = \frac{1}{2}$, entonces $\alpha_{k+1} = \frac{x_{k+1} - x_k}{2h} = \frac{h}{2h} = \frac{1}{2}$.

Por el principio de inducción matemática $\alpha_i = \frac{1}{2}$ para todos los números naturales.

Por lo tanto, la ecuación (33) equivale a

$$E_a = -\frac{1}{8} \sum_{i=1}^n |f(x_{i+1}) - 2f(x_i) + f(x_{i-1})| \quad (37)$$

Dado que $f(x_{i+1}) - 2f(x_i) + f(x_{i-1})$ es la segunda diferencia hacia adelante, la cual corresponde al numerador de la estimación numérica de la segunda derivada (Canale, 2015). Este numerador es el que designa el signo de la segunda derivada. Con lo cual los $f(x_{i+1}) - 2f(x_i) + f(x_{i-1})$ de cada segmento podrían ser positivos

o negativos, anulándose entre sí. Para resolver este problema, se aplica el valor absoluto a la segunda diferencia hacia adelante, como se observa en la ecuación (37).

Para cada área (A) del transistor *MOSFET* existe una señal diferente de las variables dependientes, y la ecuación (37) únicamente está definida para una única señal. Una manera intuitivamente lógica de resolverlo es realizar la sumatoria del error de truncamiento de cada señal, como se muestra en la ecuación (38).

$$E_a = \sum_{k=1}^m -\frac{1}{8} \sum_{i=1}^n |f(x_{i+1}, A_k) - 2f(x_i, A_k) + f(x_{i-1}, A_k)| \quad (38)$$

Complejidad computacional de las funciones objetivo y restricciones.

Ngspice se utiliza en lugar de un método analítico (Modelos de pequeña y gran señal) por la exactitud que ofrece el simulador, pero le lleva más tiempo en proporcionar los datos de los modelos que el método analítico (Aggarwal & O’Reilly, 2007) (Sallem, Pereira, Fakhfakh, & Fino, 2013). Sin embargo, la metodología de optimización está enfocada en obtener el rango $L = [L_{min}, L_{max}]$ del transistor. Con lo cual, el tiempo utilizado en el proceso de Ngspice, se compensaría o mejoraría al implementar el rango L del *MOSFET* en una gran cantidad de circuitos analógicos.

Para que la metodología de optimización sea más eficiente, se optó por reducir la complejidad computacional del conjunto de funciones objetivo y restricciones. Para las funciones objetivo de ganancia de I_{ds} y R_{ds} , se determinó la siguiente complejidad computacional.

Si en la ecuación (7) se tomaran en cuenta todos los puntos discretos de V_{ds} , entonces se sumarían $n - 200 + 1 = n - 201$ magnitudes discretas de la corriente I_{ds} ; tal que, n es el punto del voltaje final de V_{ds} y el punto 200 corresponde al punto de saturación. Ahora, a nivel de programación esta sumatoria de magnitudes de I_{ds} se resolvería en un ciclo con $n - 201$ sumas. Suponiendo que cada suma se tarda

5ms, con lo cual el tiempo total del ciclo sería $(n - 201)5ms$, al contrario de la aplicación de la ecuación (7) a nivel de programación que tardaría únicamente 5ms. Para la ecuación (8), se realiza el mismo análisis de complejidad computacional que el de la ecuación (7).

Para las restricciones, se determinó la siguiente complejidad computacional. Las restricciones (17) y (20) no se obtienen a base de una sumatoria, con lo cual su complejidad computacional está directamente relacionada a las operaciones aritméticas y trigonométricas que utilizan.

Las restricciones (13), (15), (26) y (27) utilizan una sumatoria. Con lo cual, a nivel código se utiliza un ciclo. Antes de, establecer su complejidad computacional, se utiliza el error de truncamiento (ecuación (38)) que se determinó para estas restricciones. En la Figura 13, $h = 10^{-1}, 10^{-2}, \dots, 10^{-5}$, se puede observar que al disminuir h decrece el error de truncamiento para cada variable dependiente. Además, a partir de $h = 10^{-3}$ (asterisco azul) no existen cambios significativos en el error de truncamiento. Por lo tanto, se eligió $h = inc = 10^{-3}$.

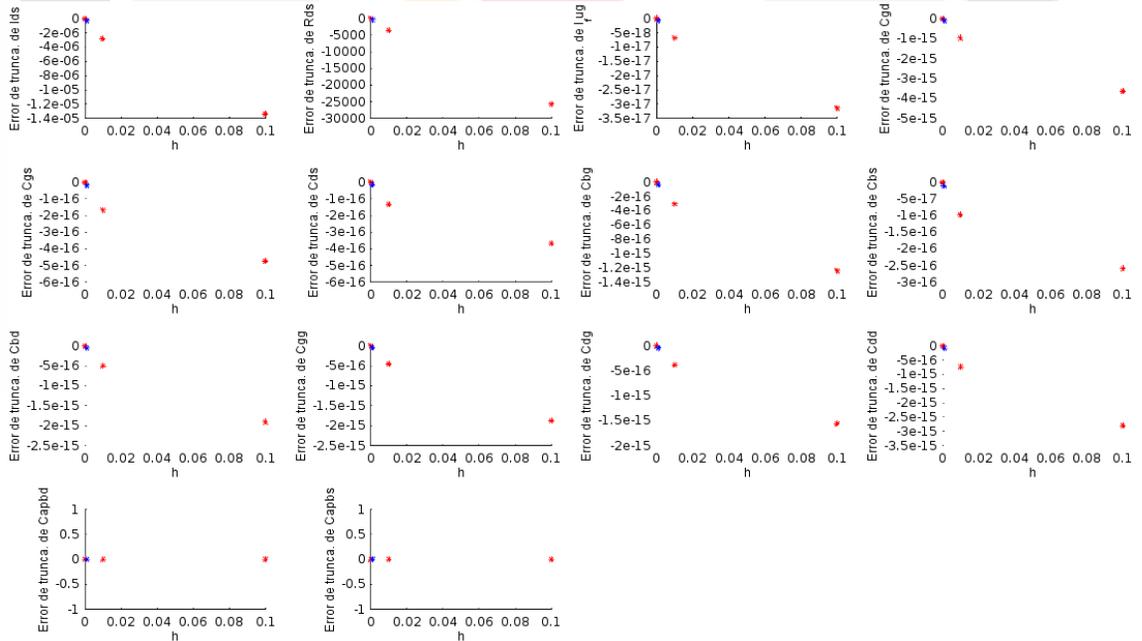


Figura 13 Error de truncamiento de las variables dependientes.

La restricción (13) se obtiene de la ecuación (11), la ecuación (11) a nivel código realizaría un ciclo de $n - 201$ iteraciones, donde n equivale a la ecuación (4). Ahora, recuérdese que $V_{fin} = 1.8$, y suponiendo que cada suma se tarda $5ms$. Entonces, la complejidad computacional de la ecuación (11) es $(n - 201)5ms = 1599x5ms$.

La ventaja de usar la ecuación (38) se nota cuando $inc = 10^{-3}$ decrece, con esto n tiende a infinito, y la complejidad computacional aumenta; pero el error de truncamiento a partir de $inc = 10^{-3}$ no tiene cambios significativos, lo cual significa que no es adecuado que inc decrezca después de 10^{-3} .

El análisis para las restricciones (15), (26) y (27) es equivalente al de la restricción (13).

La razón por la cual se utilizó la sumatoria de valores discretos en lugar de la integral discreta, se muestra a continuación. Un método para resolver la integral discreta es la regla del trapecio de aplicación múltiple, la fórmula para esta regla es la siguiente (Canale, 2015).

$$I = (x_n - x_0) \frac{f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n)}{2n} \quad (39)$$

Donde, f es la variable dependiente, x_0, x_1, \dots, x_n son puntos discretos de la variable independiente.

Se puede observar que la ecuación (39) es más compleja que una simple sumatoria de valores discretos. Por lo tanto a nivel código, la complejidad computacional de la ecuación (39) es menos eficiente que el de las ecuaciones (13), (15), (26) y (27). A pesar de que se está siendo demasiado riguroso en este análisis de complejidad computacional, la razón es porque ya se había hecho hincapié sobre el simulador ngspice, que le toma más tiempo en proporcionar los datos que los métodos analíticos.

Justificación de la metodología de optimización a utilizar

Las funciones objetivo y restricciones se desarrollaron de acuerdo con la naturaleza del problema que se trata de optimizar; el problema de optimización es obtener la mínima área, el mínimo consumo de potencia, la máxima ganancia de voltaje y el máximo ancho de banda de un *MOSFET*, el cual se define como *MOSFET* ideal. El *MOSFET* ideal se implementará en algunos circuitos analógicos, y con esto poder comparar sus desempeños con las arquitecturas implementadas con tecnología *CMOS* moderna. En la metodología de optimización se debe tomar en cuenta que las funciones objetivo y restricciones se obtuvieron de datos de simulación de Ngspice; y no de métodos analíticos, como obtención de cotas, de continuidad, de linealidad, de derivadas, etcétera. Además, para obtener el *MOSFET* ideal, se tienen que optimizar dos funciones objetivo, lo cual se debe tratar como un problema multiobjetivo en la metodología de optimización. Teniendo en cuenta lo anterior, el siguiente paso es decidir el tipo de metodología de optimización que obtendrá de manera eficiente el *MOSFET* ideal.

Si nada es conocido acerca del sistema a optimizar aparte de los tipos y el número de variables de entrada (variables independientes) y salida (funciones objetivo), el escenario es considerado como optimización de caja negra. Suponiendo que las funciones objetivo existen, entonces, los valores de las funciones se generarían por simulación o incluso ejecutando un experimento. Al abordar un problema de optimización del mundo real, generalmente se puede seguir una de las siguientes estrategias o ambas:

- Se toma el problema como se da y se adapta un algoritmo a él.
- Se intenta encontrar un “buen” modelo/formulación de la función objetivo original que sea fácil de optimizar mediante algoritmos disponibles.

El último a menudo se elige en optimización matemática “tradicional”, por ejemplo, para optimización convexa: “si se puede formular un problema como un problema de optimización convexa, entonces se puede resolver de manera eficiente.” Pero

entonces, se debe tener cuidado que la formulación de la función objetivo sea la adecuada para el problema dado, ya que los ex óptimos locales y globales serían inalcanzables, podrían aparecer nuevos óptimos. En optimización de caja negra, la relación de entrada-salida no puede ser expresada en una forma algebraica cerrada, y no hay derivadas de ningún orden disponibles, ni pruebas de continuidad. Los algoritmos para optimización de caja negra típicamente siguen la primera estrategia. Los algoritmos que se han desarrollado para optimización de caja negra son la búsqueda aleatoria pura (el método de Monte Carlo), el cual es universalmente aplicable, también es demasiado ineficiente para ser tomado en serio; Sin embargo, el procedimiento básico de Monte Carlo llevó a métodos más exitosos, por ejemplo, el algoritmo de Metrópolis, búsqueda aleatoria controlada, recocido simulado, algoritmos evolutivos, inteligencia de enjambre y sistemas inmune artificial (Preuss, 2015). A continuación, se da una pequeña descripción de algunos problemas de optimización del mundo real, tal que el sistema a optimizar se definió, o no, como un problema de caja negra.

- Los autores optimizaron el layout del *MOSFET* de potencia a través del algoritmo evolutivo basado en la población (*OPT-IA*) inspirado en el principio de selección clonal del ser humano. Se utilizaron dos simuladores, PAN (Analizador de MOS de potencia) y Spice, los cuales junto con el *OPT-IA* fueron integrados en un marco de optimización general llamado *EDA*. En el sistema *EDA*, la expresión analítica no se pudo conocer ya que está asociada a resultados que provienen del simulador Spice. Por lo cual, se pudo abordar como un problema de optimización de caja negra. La función objetivo optimizada del pico de corriente fue reducida cerca del 30% con respecto a la función sin optimizar (Greco & Rinaudo, 2012).
- Debido a la falta de información sobre el funcionamiento interno de los problemas del mundo real y su complejidad, ellos se pueden clasificar como problemas de caja negra. Los algoritmos evolutivos y los algoritmos de enjambre han demostrado resolver muchos problemas complejos del mundo real. Los autores desarrollaron la infraestructura de búsqueda paralela

adaptativa del enjambre (*PASS*) para resolver un gran rango de problemas de optimización de caja negra, el núcleo de optimización del *PASS* es la optimización estándar del enjambre de partículas 2011 (*SPSO2011*). El *PASS* es totalmente auto adaptativo debido a los bloques de selección efectiva para encontrar los mejores parámetros para el *SPSO2011*; de mapeo del algoritmo al hardware y el uso de los beneficios de la computación paralela que puede llevar a una aceleración notable. Además, el modelo de isla se utiliza para evitar el estancamiento temprano en subóptimos. De acuerdo con los resultados de los 7 problemas de caja negra, el *PASS* supero en desempeño a el *SPSO2011*, el algoritmo de evolución diferencial y el algoritmo de colonia artificial de abejas (Shuka & Brehm, 2019).

- En esta investigación, los autores utilizaron un método analítico (programación geométrica) basado en los modelos de pequeña y gran señal de los circuitos analógicos. Las ecuaciones obtenidas de los modelos de pequeña y gran señal aunque no del todo exactas, toma mucho menos tiempo que SPICE para proporcionar g_m , g_{ds} , V_{th} , V_{eff} , etcétera. El proceso de optimización para el dimensionamiento de los circuitos analógicos es acelerado a costa de la exactitud. Además, resaltaron la importancia que han tenido los algoritmos de optimización estocástica de caja negra para resolver el dimensionamiento, tales como algoritmos genéticos y recocido simulado (Aggarwal & O'Reilly, 2007).

En la metodología de optimización, otro aspecto importante que se debe tomar en cuenta es que se está tratando con un problema de optimización multiobjetivo (*MOP*). Ya que, se tienen que optimizar simultáneamente dos funciones objetivo para obtener el *MOSFET* ideal. Existen dos maneras importantes para resolver optimización multiobjetivo: la agregación ponderada (la función objetivo es igual a la suma ponderada de los subobjetivos), y el frente de Pareto (Conjunto de soluciones).

La agregación ponderada es uno de los enfoques más simples para tratar con *MOPs*, el cual es definir una función objetivo agregativa como una suma ponderada de los subobjetivos (Engelbrecht, 2007):

$$f(x) = \sum_{k=1}^{n_k} w_k f_k(x)$$

donde $n_k \geq 2$ es el número total de subobjetivos, y $w_k \in [0,1]$, $k = 1, \dots, n_k$ con $\sum_{k=1}^{n_k} w_k = 1$. Mientras el enfoque de agregación es muy simple de implementar y computacionalmente eficiente: este sufre de los siguientes problemas:

- Es difícil obtener los mejores valores para los pesos, w_k , ya que estos dependen del problema.
- Estos métodos deben ser reaplicados para encontrar más de una solución, ya que solo una solución puede ser obtenida con una sola ejecución de un algoritmo de agregación. Sin embargo, incluso para aplicaciones repetidas, no hay garantía que diferentes soluciones sean encontradas.
- En el método anterior, la agregación ponderada convencional no puede resolver los *MOPs* con un frente de Pareto cóncavo (Das & Dennis, 1997).

Para abordar estos problemas, se propuso el método de agregación con un cambio dinámico de los pesos, y el método de variación de curvatura dinámico; con estos métodos se genera todos los puntos del frente de Pareto (Dellino, Fedele, & Meloni, 2011).

Los algoritmos evolutivos parecen particularmente adecuados para resolver *MOPs*, porque tratan simultáneamente con un conjunto de posibles soluciones (la llamada población) que permiten encontrar varios miembros del conjunto óptimo de Pareto en una sola ejecución del algoritmo (Deb, 2008), en lugar de tener que realizar una serie de ejecuciones separadas como en el caso de las técnicas tradicionales de programación matemática. Adicionalmente, los algoritmos evolutivos son

menos susceptibles a la forma o continuidad del frente de Pareto (por ejemplo, pueden lidiar fácilmente con frentes de Pareto discontinuos o cóncavos), mientras estos dos problemas son una preocupación real para las técnicas de programación matemática (Coello Coello, 1999) (Coello, Lamont, & Veldhuizen, 2007).

En la metodología de optimización se incorpora el simulador Ngspice para obtener las funciones objetivo a optimizar y las restricciones, y se está lidiando con un *MOP*. Recuérdese, que el simulador Ngspice se utiliza para lidiar con los diferentes modelos de las nuevas tecnologías *MOSFETs*. Por la investigación sobre los problemas de caja negra y los *MOPs* que se realizó anteriormente; y tomando en cuenta que en la metodología de optimización está anexado el simulador Ngspice y es un *MOP*. Por lo tanto, los algoritmos evolutivos son la opción más adecuada ya que son capaces de lidiar eficientemente con problemas de caja negra (Simulaciones), y con *MOP* con respecto a las técnicas de optimización Matemáticas.

Han existido algunas investigaciones que realizan la optimización del transistor *MOSFET* con aplicaciones particulares (potencia alta, ópticas, compuerta doble, circuitos en general), la optimización se realizó por medio del algoritmo genético multiobjetivo (*MOGA*), teniendo excelentes resultados de optimización (Melik-Adamyán, 2009) (Bendib & Djeflal, 2013) (Djeflal & Ferhati, 2016) (Bencherif et al., 2019). Debido a la gran variante de algoritmos evolutivos que existen, al existir de *MOGA* en optimización de tecnologías basadas en *MOSFETs* y que *MOGA* pertenece a los algoritmos evolutivos; parece razonable, utilizar la metodología de optimización *MOGA* para obtener el *MOSFET* ideal. Antes de realizar la aplicación de *MOGA*, se darán algunas definiciones de los elementos básicos de los algoritmos evolutivos multi objetivos.

Arquitectura de software del problema de optimización

Las arquitecturas de software estándares facilitan construir un sistema que se utiliza para resolver un problema determinado, ya que son arquitecturas probadas en

sistemas similares al sistema a desarrollar. El sistema que se va a desarrollar es un sistema de lectura y optimización de datos, los datos se van a leer del simulador Ngspice y se van a optimizar a través del *MOGA*. Por lo tanto, se requieren 3 arquitecturas: una arquitectura para el simulador, una arquitectura para la comunicación entre el simulador y el *MOGA*, y una arquitectura para la etapa de optimización (*MOGA*). A continuación, se describen las arquitecturas a utilizar en cada etapa y se presentan sus correspondientes diagramas.

Arquitectura del simulador ngspice

Ngspice soporta simulación de nivel mixto y proporciona un enlace directo entre parámetros de tecnología y desempeño de circuito. Un circuito de nivel mixto y un simulador de dispositivo pueden proveer mayor precisión de simulación que un circuito autónomo o simulador de dispositivo mediante el modelado numérico de los dispositivos críticos en un circuito. Las extensiones de nivel mixto para ngspice están en CIDER, un circuito de nivel mixto y un simulador de dispositivo integrado en código ngspice. Además, ngspice soporta simulación de señal mixta y nivel de tablero a través de la integración de código XSPICE (Holger Vogt, 2018).

Históricamente, los simuladores de circuito han soportado un algoritmo de simulación analógica o un algoritmo de simulación digital. Ngspice hereda la estructura de XSPICE y soporta ambos algoritmos analógicos y digitales, por lo cual es un simulador de modo mixto (Holger Vogt, 2018). En la Figura 14, se presenta la arquitectura de software de ngspice/XSPICE.

En la arquitectura de software de ngspice/XSPICE, se puede observar que la información que incluye cada módulo (generador de modelo de código, librerías compartidas, etc.) proporciona los datos necesarios, para intuir que realiza cada módulo. Sin embargo, puesto que a esta arquitectura no se le pueden realizar modificaciones, no es necesario comprenderla detalladamente.

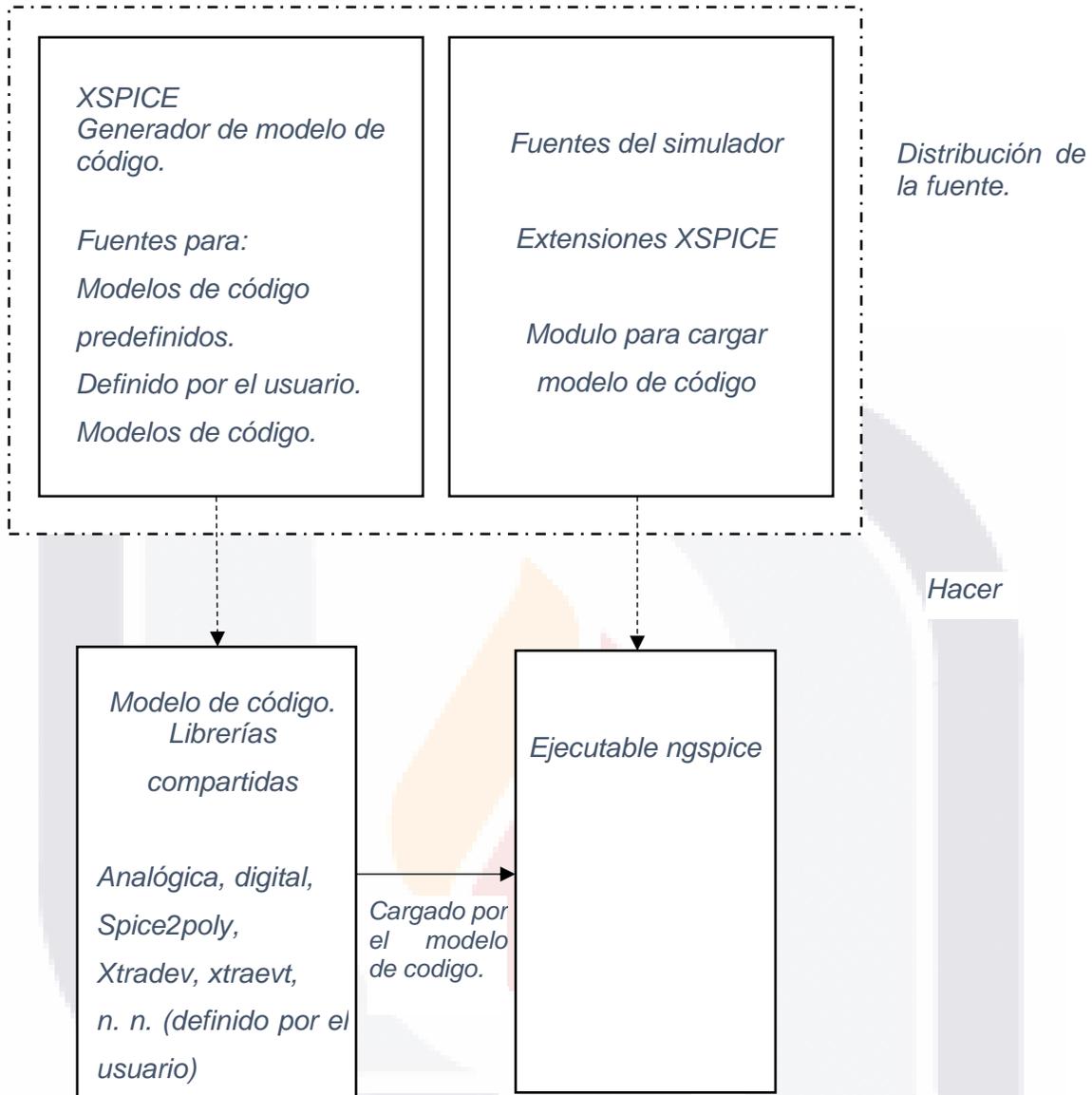


Figura 14 Diagrama de ngspice/XSPICE (Holger Vogt, 2018).

Ngspice permite diseñar una interfaz de usuario, ya que puede ser compilado como librería compartida. Se puede desarrollar y optimizar esta interfaz de usuario sin necesidad de alterar el propio código fuente de ngspice. Por lo tanto, se pueden llevar a cabo una serie de rutinas, como, desarrollar un código que envíe un archivo con las conexiones de un circuito a ngspice, realice la lectura de datos de las variables del circuito, altere parámetros del circuito, detener el simulador de acuerdo con una condición, etcétera. Para realizar estas rutinas en el código de

usuario, se requieren una serie de funciones y comandos propios de ngspice, a continuación se describen algunas funciones y comandos (Holger Vogt, 2018).

- `pvector_info ngGet_Vec_Info(char*)`: `pvector_info` es tipo estructura del código de ngspice, el argumento es el nombre de un vector.
- `int ngSpice_Init(SendChar*, SendStat*, ControlledExit*, SendData*, SendInitData*, BGThreadRunning*, void)`: El simulador es inicializado, los argumentos son direcciones de puntero de algunas funciones de ngspice.
- `ngSpice_Command((char *)"bg_run")`: Empieza ngspice dentro de la tarea principal, tal que `bg_run` es un comando de ngspice. Otros comandos son `bg_halt`, `alter @dev[param] = <expression>`, etc.

Arquitectura multiobjetivo para el MOGA

La optimización multiobjetivo ha sido aplicada a problemas de arquitectura de software, y se ha demostrado ser un paradigma efectivo particular en problemas que pueden ser nativamente formulados a través de métricas cuantificables. La arquitectura de software que se aplicó para la optimización está basada en la optimización multiobjetivo evolutiva, esta arquitectura es nombrada Enfoque Evolutivo para la refactorización de arquitectura de software multiobjetivo (*EASIER*). Los primeros resultados experimentales del enfoque de los autores de este estudio son prometedores en términos de aplicaciones prácticas (Arcelli et al., 2018).

El núcleo de *EASIER* está representado por el algoritmo NSGA-II el cual es un algoritmo genético multiobjetivo elitista, que toma como entrada una arquitectura de software inicial y busca el espacio de arquitecturas por acciones de recombinación de refactorización extraídas de un repositorio, por ejemplo, de una librería de acciones de refactorización. El proceso de búsqueda está dirigido por tres funciones principales objetivo de aptitud, las cuales son indicador de calidad de desempeño, la distancia arquitectónica (que será expresada como una medida de la intensidad de los cambios inducidos por las acciones de refactorización) y el número de anti-

patrones de desempeño. Además, un analizador de desempeño fue diseñado para lograr la calidad adecuada en los requerimientos (Arcelli et al., 2018).

El objetivo es adaptar esta arquitectura de software en el *MOGA*, el algoritmo será capaz de optimizar el tamaño del transistor *MOSFET* de acuerdo con sus variables intrínsecas que degradan su funcionamiento. La adaptación se llevará a cabo siguiendo la misma metodología de desarrollo de la arquitectura *EASIER*. En la Figura 15 se muestra el diagrama de arquitectura de alto nivel adaptado al *MOGA*.

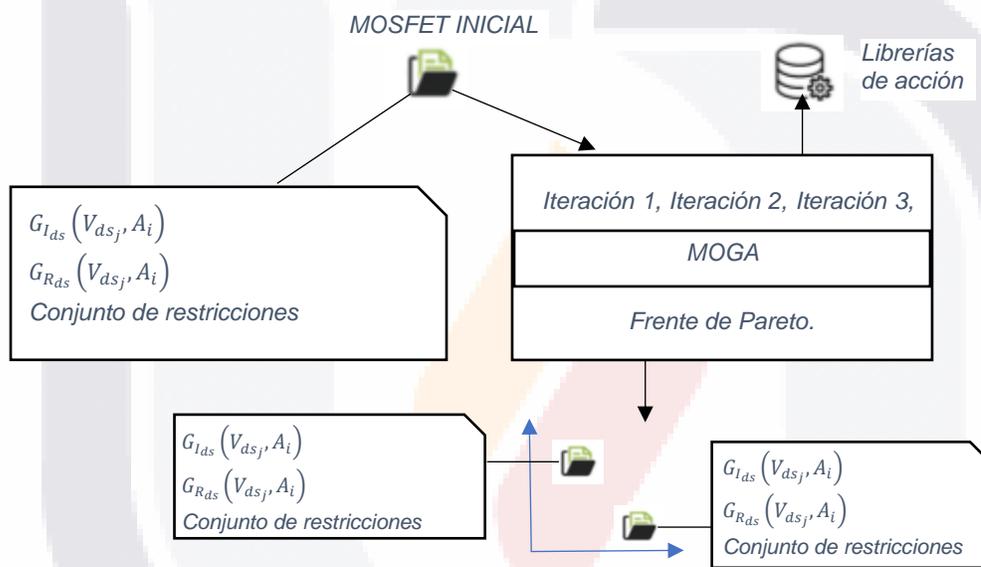


Figura 15 Arquitectura de alto nivel (Arcelli, Cortellessa, Emidio, & Pompeo, 2018).

Arquitectura de comunicación

La gran ventaja que tiene el simulador NGSPICE es que permite el acceso a funciones y comandos, ya que estas funciones se declaran globalmente en el código del usuario para ser accesibles por el lenguaje C de ngspice y los comandos son argumentos de la función `int ngSpice_Command(char*)`. El modelo de comunicación que se establecerá entre el código del usuario y el código de NGPICE es el mostrado en la Figura 16, cabe recalcar que el siguiente modelo está basado en el código intrínseco de NGSPICE, pero en el manual de NGSPICE no contiene ningún modelo de comunicación basado en su propio código.

Unión de las arquitecturas para el flujo de datos entre ngspice y MOGA

Para obtener la arquitectura final deberá existir una correcta comunicación entre las tres arquitecturas. Al conocer el proceso con el que cada arquitectura resuelve sus tareas se propondrá una arquitectura general de manera intuitiva, la cual estará integrada por la arquitectura del simulador NGSPICE, la arquitectura del MOGA y la arquitectura de la etapa de comunicación.

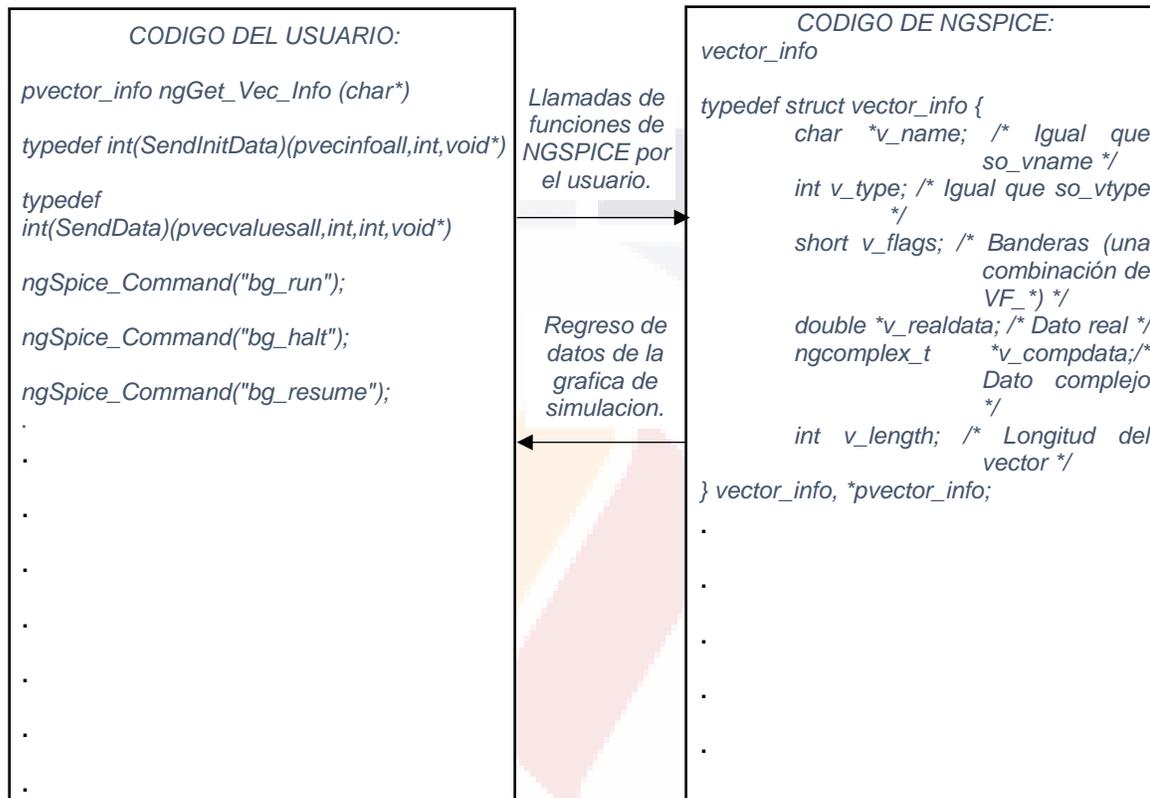


Figura 16 Modelo de comunicación basado en el código de comunicación de NGSPICE (Holger Vogt, 2018)

La clave para la comunicación entre las tres arquitecturas está en la arquitectura de comunicación. Por lo tanto, las arquitecturas se conectarán de la siguiente forma: el módulo de código de NGSPICE de la arquitectura mostrada en la Figura 16 será parte del módulo de librerías compartidas de la arquitectura NGSPICE (Figura 14). Por último, el módulo de código del usuario de la arquitectura mostrada en la Figura 16 formará parte del módulo de funciones objetivo y restricciones, así como del módulo de iteraciones de la arquitectura del MOGA (Figura 15). Tomando en cuenta que NGSPICE permite declarar sus funciones de manera global en el diseño de código para ser accesibles por el lenguaje C, y leer los datos de la gráfica de simulación. Entonces, la comunicación entre las arquitecturas quedo completada ya

que con la llamada de funciones en el código del usuario y el regreso de datos del código de NGSPICE mostrados en la Figura 16, serán los encargados de comunicar a las arquitecturas.



Resultados

Los siguientes parámetros del *MOGA* fueron los establecidos para la optimización del transistor *MOSFET*: $p_c = 0.8$, $p_m = 0.75$, $n_c = 5$, $n_m = 5$, $g_{max} = 1000$, $m = 200$ y $t_{ampareto} = 1000$. A pesar de los inconvenientes mencionados en la sección elección de los parámetros del *MOGA*, estos parámetros fueron elegidos para la optimización de las funciones objetivo con restricciones (ecuaciones (7), (8), (13), (15), (17), (20), (26) y (27)). Debido a que el *MOGA* ha tenido excelentes resultados de optimización en el transistor *MOSFET* con aplicaciones particulares (potencia alta, ópticas, compuerta doble, circuitos en general).

Debido a que, los efectos de canal corto degradan el desempeño del transistor *MOSFET* con aplicaciones en circuitos integrados analógicos, se aplicó el *MOGA* a la optimización de las funciones objetivo y restricciones (ecuaciones (7), (8), (13), (15), (17), (20), (26) y (27)) para obtener la mínima área, el mínimo consumo de potencia, la máxima ganancia de voltaje y el máximo ancho de banda del *MOSFET* implementado en circuitos analógicos. El *MOGA* se ejecutó en un *MOSFET Bulk* de tecnología 180nm, la cual pertenece al BSIM3; y en un *MOSFET Bulk* de tecnología 90nm (Cao, n.d.), la cual pertenece al BSIM4. Las funciones objetivo de la ganancia de I_{ds} (Ecuación (7)) y R_{ds} (Ecuación (8)) para todo ciclo de optimización fueron de maximización. Así como, cada PF_{true} y P_{true} de ambas tecnologías del *MOSFET*, se adquirieron de la arquitectura de optimización en una PC con 1.60 GHz y 8.00 GB de RAM.

Puesto que, el simulador Ngspice se utilizó en la arquitectura de optimización por la exactitud de los datos obtenidos, esto causa que el proceso de optimización sea más tardado que si se implementara un método analítico (Modelos de pequeña y gran señal). Por lo tanto se realizaron pruebas de convergencia en las tecnologías de 180nm y 90nm para determinar el número de generaciones (g_{max}) y el tamaño de la población (m) que ejecuten el ciclo de optimización en un tiempo aceptable. Los restantes parámetros del *MOGA* permanecieron como los obtenidos en las

funciones de prueba con restricciones. En la tabla 2 se muestra el tiempo de ejecución del *MOGA* para determinados valores de $gmax$ y m . Tal que Cap , I_{fug} , $limite_{I_{ds}}$, ang , $E_{I_{ds}}$ y $E_{R_{ds}}$ son variables de las restricciones (13), (15), (17), (20), (26) y (27) respectivamente; λ es el incremento de L , C es el factor de escala, p es una variable para elegir en que voltaje (V_{ds}) a partir de saturación se comparan las ganancias, lim_inf es el longitud mínima del canal del *MOSFET* y lim_sup es el longitud máxima del canal del *MOSFET*. Se puede observar en la tabla 2, que para la tecnología 180nm con $gmax = 30$ y $m = 200$ el tiempo de ejecución del *MOGA* es de 63.9824 minutos; y con $gmax = 3$ y $m = 100$ el tiempo de ejecución es de 1.57662 minutos. Además, para la tecnología 90nm con $gmax = 30$ y $m = 200$ el tiempo de ejecución del *MOGA* es de 67.3228 minutos; y con $gmax = 3$ y $m = 100$ el tiempo de ejecución es de 1.81773 minutos. Los valores de las variables de las restricciones para la tecnología 180nm en ambas ejecuciones del *MOGA* fueron iguales, también se establecieron iguales en la tecnología 90nm.

Tabla 2 Pruebas de convergencia con respecto al tiempo de las tecnologías 180nm y 90nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	$E_{R_{ds}}$	I_{fug}
Tecnología 180nm	30	200	89.3	0	1	1	1.1×10^{-12}
Tecnología 180nm	3	100	89.3	0	1	1	1.1×10^{-12}
Tecnología 90nm	30	200	-85.9	0	1	1	1.683×10^{-8}
Tecnología 90nm	3	100	-85.9	0	1	1	1.683×10^{-8}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.7×10^{-13}	180nm	1800nm	90nm	8	.5	63.9824 minutos
	1.7×10^{-13}	180nm	1800nm	90nm	8	.5	1.57662 minutos
	1.575×10^{-13}	90nm	900nm	45nm	10	.5	67.3228 minutos
	1.575×10^{-13}	90nm	900nm	45nm	10	.5	1.81773 minutos

Ahora, en la Figura 17 se observa el PF_{true} y P_{true} (con los valores de la Tabla 2) de la tecnología 180nm con $gmax = 30$ y $m = 200$ (Círculos de color rojo); y con $gmax = 3$ y $m = 100$ (Círculos de color azul). también, en la Figura 18 se observa el PF_{true} y P_{true} (con los valores de la Tabla 2) de la tecnología 90nm con $gmax = 30$ y $m = 200$ (Círculos de color rojo); y con $gmax = 3$ y $m = 100$ (Círculos de color azul).

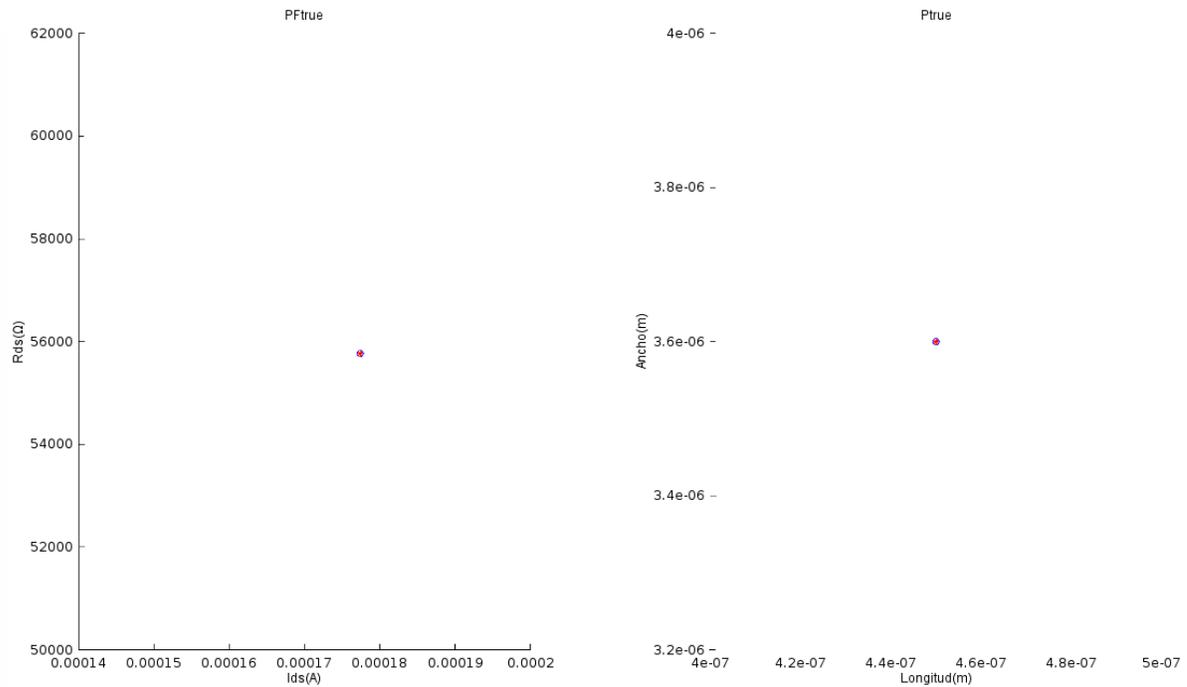


Figura 17 PF_{true} y P_{true} de la tecnología 180nm para determinar $gmax$ y m .

Es de resaltar que la tecnología de 90nm es predictiva, lo que significa que todavía sus valores no han sido probados. Con lo cual, el comportamiento de la variable dependiente I_{ds} es anormal, pero aun así se utilizó esta tecnología de 90nm en el proceso de optimización para determinar el comportamiento de las funciones objetivo y restricciones del *MOSFET* en el modelo BSIM4. El tiempo de ejecución del *MOGA* para ambas tecnologías, cuando $gmax = 30$ y $m = 200$ es demasiado grande en comparación con $gmax = 3$ y $m = 100$, y en las Figuras 17 y 18 se muestra que el PF_{true} y P_{true} es el mismo para las dos tecnologías. Además, basándose en que la tecnología de 180nm está probada, lo que significa que los valores que arroje el simulador son más confiables que la tecnología de 90nm. Y,

como se observa en la Figura 17 (tecnología 180nm) únicamente se obtuvo una solución de Pareto y un punto del frente de Pareto, debido a esto se optó por ejecutar el MOGA con diferentes valores de ang para obtener un conjunto de soluciones del frente de Pareto. Con lo cual, el tiempo de ejecución del MOGA con $gmax = 30$ y $m = 200$ para cada ejecución de ang sería demasiado grande. Entonces, se eligió $gmax = 3$ y $m = 100$ en ambas tecnologías para obtener el conjunto de soluciones de Pareto al variar ang .

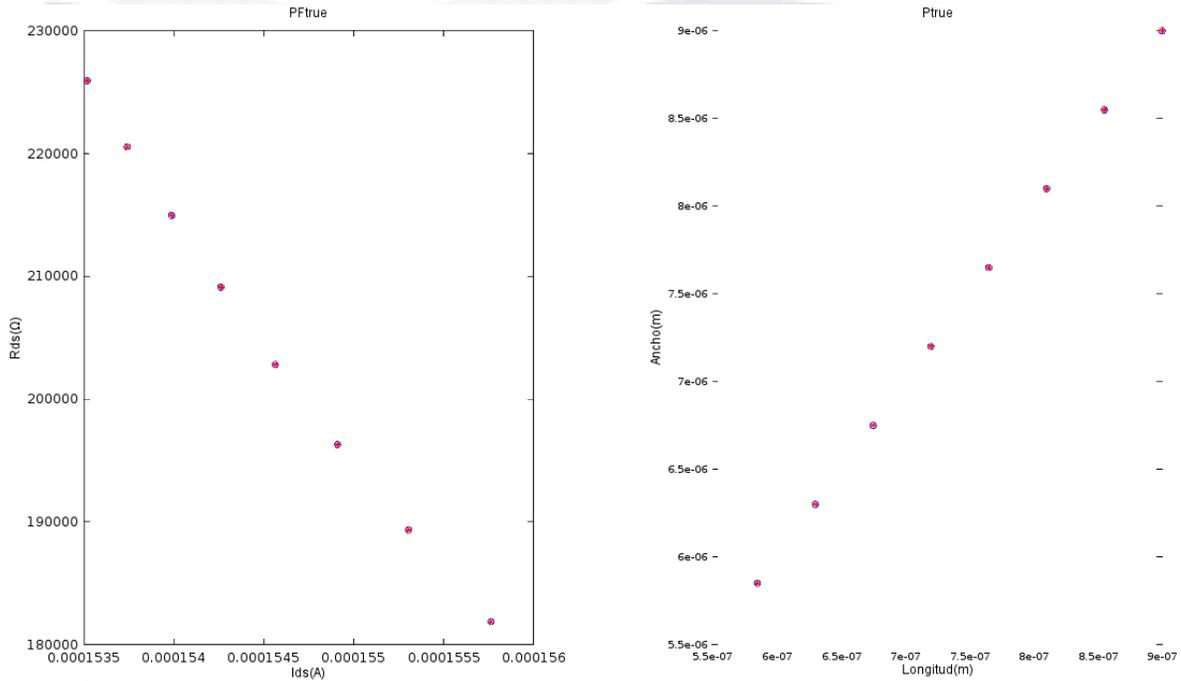


Figura 18 PF_{true} y P_{true} de la tecnología 90nm para determinar $gmax$ y m .

La variable ang cambió sus valores para obtener el conjunto de soluciones del frente de Pareto de acuerdo con la siguiente ecuación.

$$ang = ang_1, ang_2, ang_3, \dots, ang_i \tag{40}$$

Para cada valor de ang ($ang_1, ang_2, ang_3, \dots, ang_i$) se realizó un ciclo de optimización con el MOGA.

A continuación, se muestran las Tablas 3 y 4; y Figuras 19 y 20, para las tecnologías 180nm y 90nm respectivamente. Donde, se obtuvo el PF_{true} y P_{true} para cada tecnología, tal que Cap , I_{fug} , $limite_{Ids}$, E_{Ids} y E_{Rds} se establecieron para cubrir el

espacio entero de las variables de decisión; y ang varió de acuerdo con la ecuación (51).

Tabla 3 Conjunto de restricciones y parámetros del MOGA para obtener el PF_{true} y P_{true} de la tecnología 180nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	$E_{R_{ds}}$	I_{fug}
Tecnología 180nm	3	100	85.3, 85.5, ... , 89.5	0	1	1	1.1×10^{-12}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.7×10^{-13}	180nm	1800nm	90nm	8	.5	165.055 minutos

Tabla 4 Conjunto de restricciones y parámetros del MOGA para obtener el PF_{true} y P_{true} de la tecnología 90nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	$E_{R_{ds}}$	I_{fug}
Tecnología 90nm	3	100	-79, -80, ..., -90	0	1	1	1.683×10^{-8}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.575×10^{-13}	90nm	900nm	45nm	10	.5	52.1292 minutos

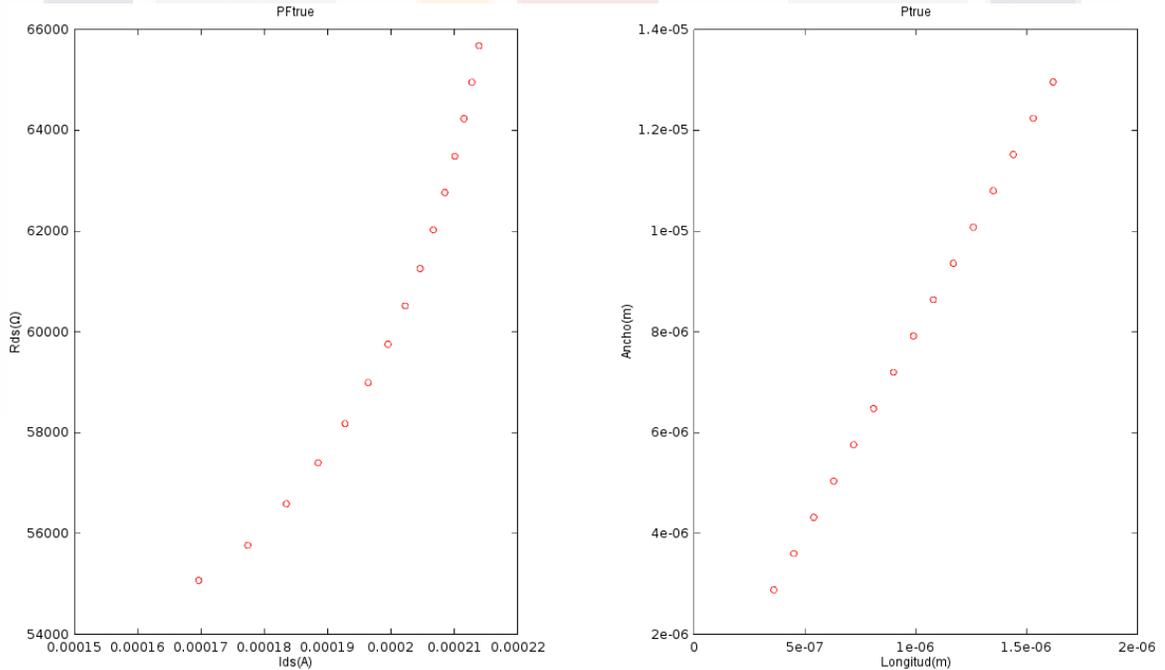


Figura 19 PF_{true} y P_{true} de la tecnología 180nm con los valores de la Tabla 3.

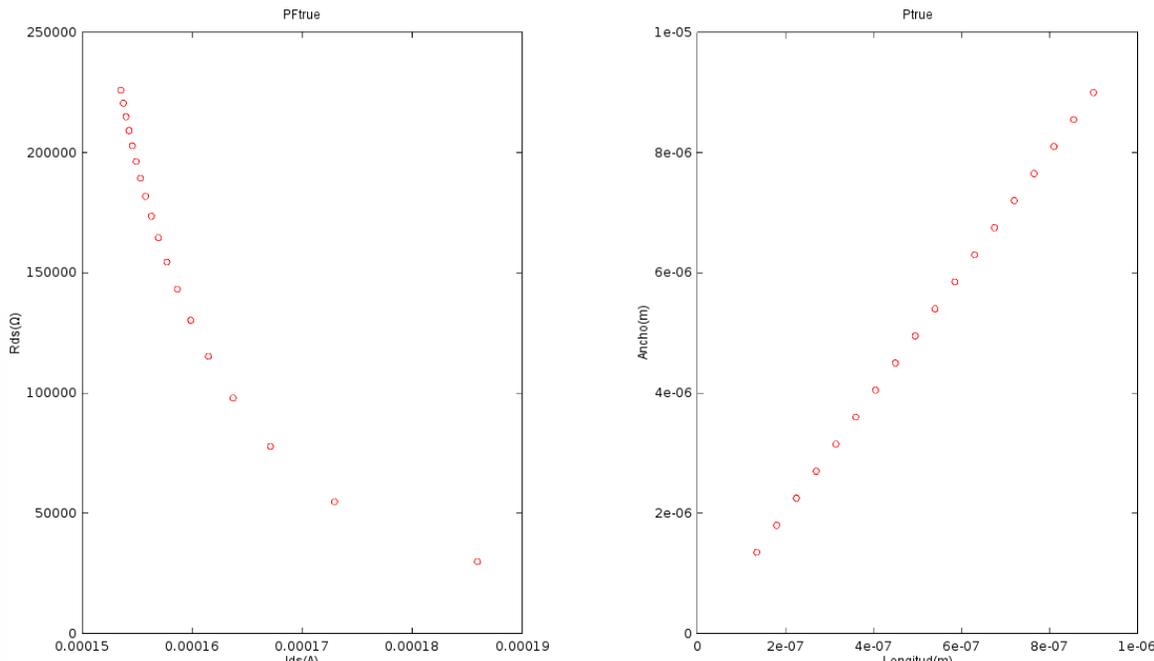


Figura 20 PF_{true} y P_{true} de la tecnología 90nm con los valores de la Tabla 4.

En la Figura 19, para cada valor de ang se obtiene una solución del frente de Pareto, al contrario de la Figura 20. Al observar la Figura 19, cuando el ancho del canal del transistor *MOSFET* (W) y la longitud del canal del transistor *MOSFET* (L) incrementan; también, las funciones objetivo de ganancia I_{ds} y R_{ds} del PF_{true} incrementan. En la Figura 20, cuando W y L incrementan, la función objetivo R_{ds} incrementa, pero la función objetivo I_{ds} decrece. Esto es debido a que la tecnología de 90nm es predictiva, como ya se había mencionado.

Se presentan las Tablas 5, 6, 7, 8 y 9 para la tecnología 180nm. Donde, los valores de Cap , E_{Ids} , E_{Rds} , $limite_{Ids}$, e I_{fug} se establecieron respectivamente en cada Tabla para obtener la longitud mínima (L_{min}) y la longitud máxima (L_{max}) del *MOSFET*, tal que $L = [L_{min}, L_{max}]$. Además, ang varió de acuerdo con la ecuación (51), y recuérdese que estas variables se establecieron en las ecuaciones (13), (15), (17), (20), (26) y (27).

Tabla 5 Definición de Cap para obtener PF_{true} y P_{true} de la tecnología 180nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	E_{Rds}	I_{fug}
Tecnología 180nm	3	100	87.5, 87.7, ... , 89.5	0	1	1	1.1×10^{-12}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	4.652×10^{-14}	180nm	1800nm	90nm	8	.5	37.7587 minutos

Tabla 6 Definición de $E_{I_{ds}}$ para obtener PF_{true} y P_{true} de la tecnología 180nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	E_{Rds}	I_{fug}
Tecnología 180nm	3	100	86.7, 86.9, ..., 89.3	0	.256	1	1.1×10^{-12}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.7×10^{-13}	180nm	1800nm	90nm	8	.5	55.795 minutos

Tabla 7 Definición de E_{Rds} para obtener PF_{true} y P_{true} de la tecnología 180nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	E_{Rds}	I_{fug}
Tecnología 180nm	3	100	86.7, 86.9, ... , 89.3	0	1	.898	1.1×10^{-12}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.7×10^{-13}	180nm	1800nm	90nm	8	.5	61.413 minutos

Tabla 8 Definición de $limite_{I_{ds}}$ para obtener PF_{true} y P_{true} de la tecnología 180nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	E_{Rds}	I_{fug}
Tecnología 180nm	3	100	88.75, 88.8, ... , 89.55	70	1	1	1.1×10^{-12}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.7×10^{-13}	180nm	1800nm	90nm	8	.5	84.7756 minutos

Tabla 9 Definición de I_{fug} para obtener PF_{true} y P_{true} de la tecnología 180nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	$E_{I_{ds}}$	$E_{R_{ds}}$	I_{fug}
Tecnología 180nm	3	100	87.5, 87.7, ... , 89.5	0	1	1	1.0814×10^{-12}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.7×10^{-13}	180nm	1800nm	90nm	8	.5	36.7935 minutos

En la Figura 21, se muestra el PF_{true} y P_{true} para cada variable Cap , $E_{I_{ds}}$, $E_{R_{ds}}$, $limite_{I_{ds}}$, e I_{fug} , tal que sus valores se establecieron en las Tablas 5, 6, 7, 8 y 9 respectivamente. Para identificar, en la Figura 21 el PF_{true} y P_{true} para cada variable, a cada PF_{true} y P_{true} se le asigno el título de la ecuación ((13), (15), (17), (26) y (27)) a la que pertenece cada variable.

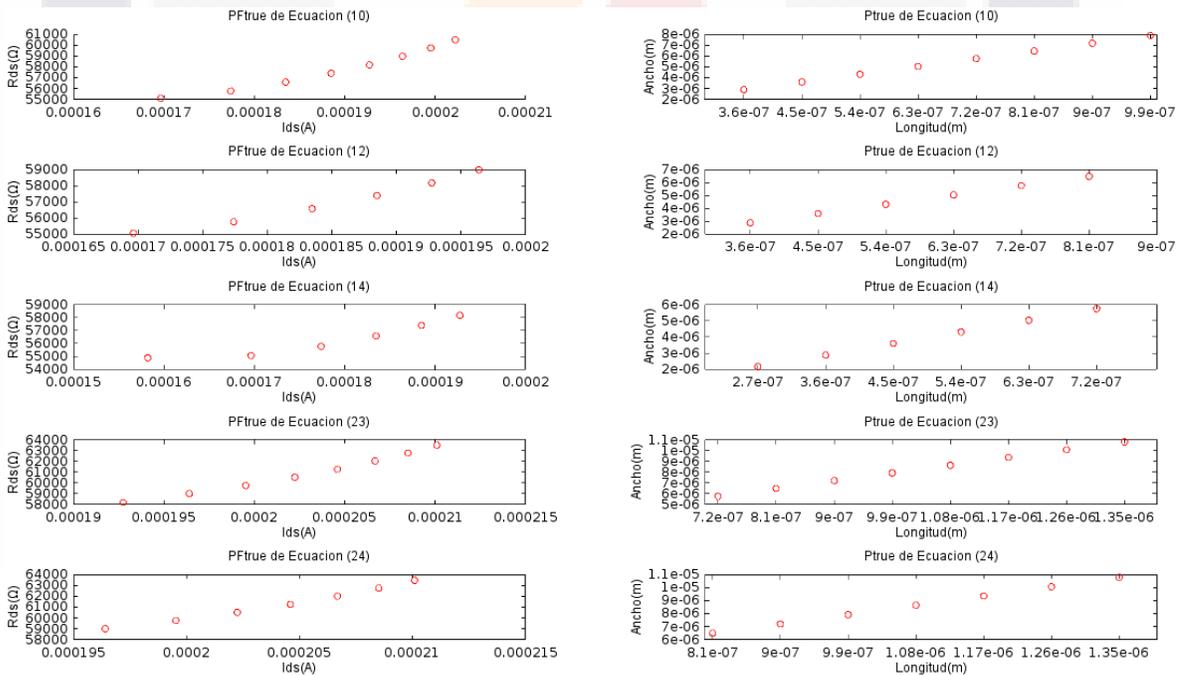


Figura 21 PF_{true} y P_{true} de la tecnología 180nm para cada ecuación (13), (15), (17), (26) y (27).

En la Figura 21, se observa que al incrementar L y W , asimismo, la curva del frente de Pareto incrementa. Además, se muestra el rango de $L = [L_{min}, L_{max}]$ en las soluciones de Pareto para cada ecuación (13), (15), (17), (26) y (27), por ejemplo,

para la ecuación (13) cuyo valor cap se encuentra en la Tabla 5, y las demás restricciones se establecieron para optimizar en el espacio entero de las variables de decisión, se obtuvo $L = [3.6x10^{-7}, 9.9x10^{-7}]$.

A continuación, se exponen las Tablas 10, 11, 12 y 13 para la tecnología 90nm. Tal que, el proceso de optimización para obtener el PF_{true} y P_{true} con respecto a los valores de Cap , E_{Ids} , E_{Rds} e I_{fug} es análogo a la optimización de la tecnología 180nm. La variable $limite_{I_{ds}}$ no se tomó en cuenta para la tecnología 90nm, ya que la curva de I_{ds} decrece cuando incrementa L .

Tabla 10 Definición de Cap para obtener PF_{true} y P_{true} de la tecnología 90nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	E_{Ids}	E_{Rds}	I_{fug}
Tecnología 90nm	3	100	-82, -83, ... , -90	0	1	1	$1.683x10^{-8}$
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	$4.507x10^{-14}$	90nm	900nm	45nm	10	.5	31.0357 minutos

Tabla 11 Definición de E_{Ids} para obtener PF_{true} y P_{true} de la tecnología 90nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	E_{Ids}	E_{Rds}	I_{fug}
Tecnología 90nm	3	100	-79, -80, ... , -90	0	.609	1	$1.683x10^{-8}$
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	$1.575x10^{-13}$	90nm	900nm	45nm	10	.5	50.1528 minutos

Tabla 12 Definición de E_{Rds} para obtener PF_{true} y P_{true} de la tecnología 90nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	E_{Ids}	E_{Rds}	I_{fug}
Tecnología 90nm	3	100	-79, -80, ... , -90	0	1	.928	$1.683x10^{-8}$
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	$1.575x10^{-13}$	90nm	900nm	45nm	10	.5	50.1528 minutos

Tabla 13 Definición de I_{fug} para obtener PF_{true} y P_{true} de la tecnología 90nm.

	$gmax$	m	ang	$limite_{I_{ds}}$	E_{Ids}	E_{Rds}	I_{fug}
Tecnología 90nm	3	100	-82, -83, ... , -90	0	1	1	3.739×10^{-9}
	Cap	lim_inf	lim_sup	λ	C	p	$tiempo$
	1.575×10^{-13}	90nm	900nm	45nm	10	.5	31.9349 minutos

En la Figura 22, se muestra el PF_{true} y P_{true} para cada variable Cap , E_{Ids} , E_{Rds} e I_{fug} .

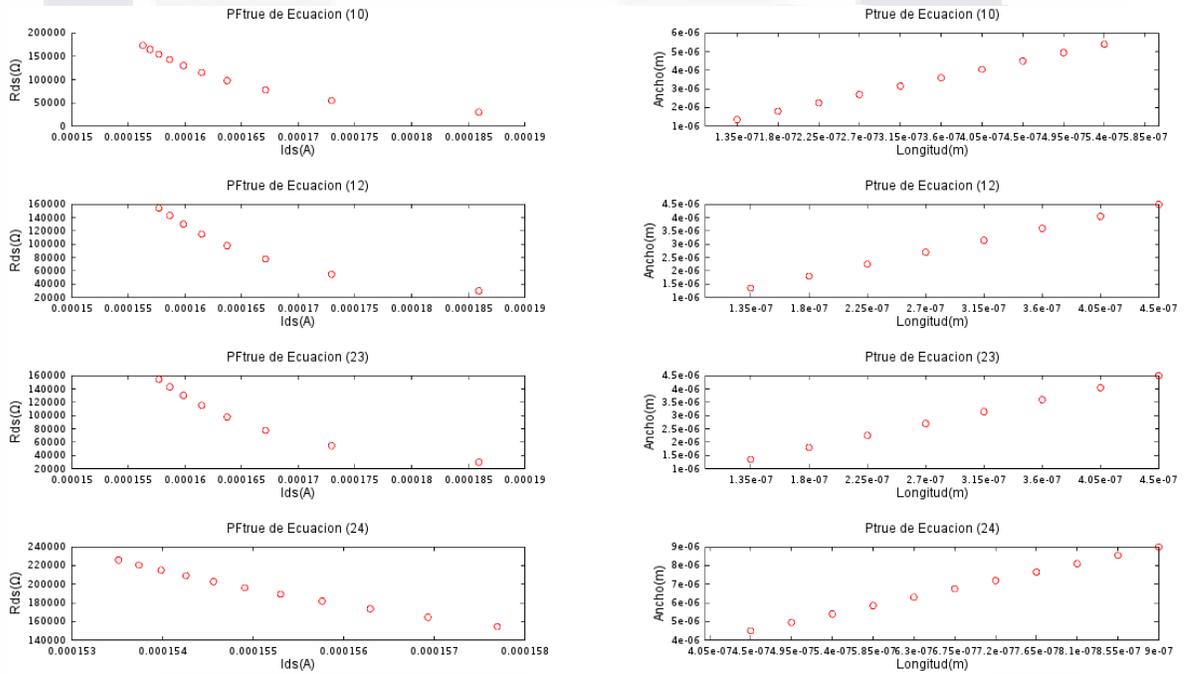


Figura 22 PF_{true} y P_{true} de la tecnología 90nm para cada ecuación (13), (15), (26) y (27).

En la Figura 22, al incrementar L y W , asimismo, la función objetivo de ganancia R_{ds} incrementa, pero la función objetivo de ganancia I_{ds} decrece.

Después de obtener el PF_{true} y P_{true} con el MOGA, para el conjunto de funciones objetivo y restricciones (ecuaciones (7), (8), (13), (15), (17), (20), (26) y (27)), se optó por probar estos conjuntos de soluciones de Pareto en dos amplificadores operacionales (*opamps*), los cuales son el amplificador de dos etapas (Figura 23) y

el amplificador *Rail to Rail* (Figura 24). Las pruebas que se realizaron a los dos amplificadores en el simulador LTspice XVII fueron las siguientes:

- 1) La distorsión armónica total (*THD*) la cual describe la no linealidad de un amplificador. En general, cuanto más pequeño sea el *THD* el amplificador operacional tendrá una respuesta más lineal. El *THD* se utilizó para comprobar el P_{true} de las ecuaciones (26) y (27), y se usó para establecer la mínima área del *opamp* con respecto a estas restricciones.
- 2) El margen de fase y ganancia que definen el margen de estabilidad de un amplificador analógico. A mayor margen de fase y a menor margen de ganancia, mayor margen de estabilidad. El margen de fase y ganancia se utilizaron para probar el P_{true} de las ecuaciones (13) y (15). También, estas pruebas se utilizaron para establecer la mínima área (con respecto a las ecuaciones (13) y (15)), el mínimo consumo de potencia y el máximo ancho de banda del *opamp*.

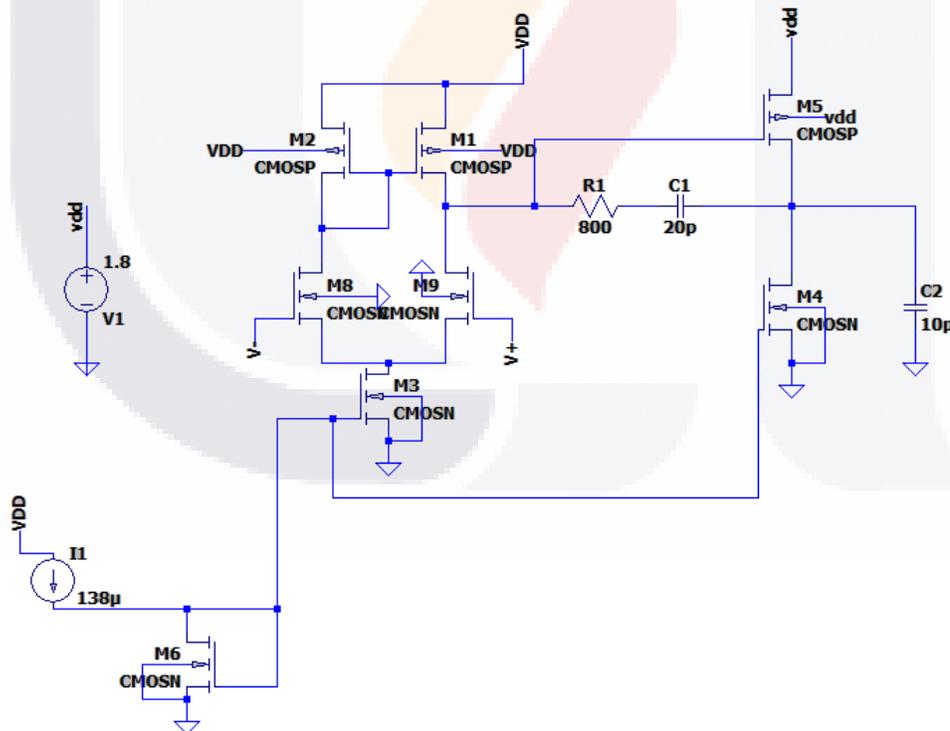


Figura 23 Amplificador de dos etapas.

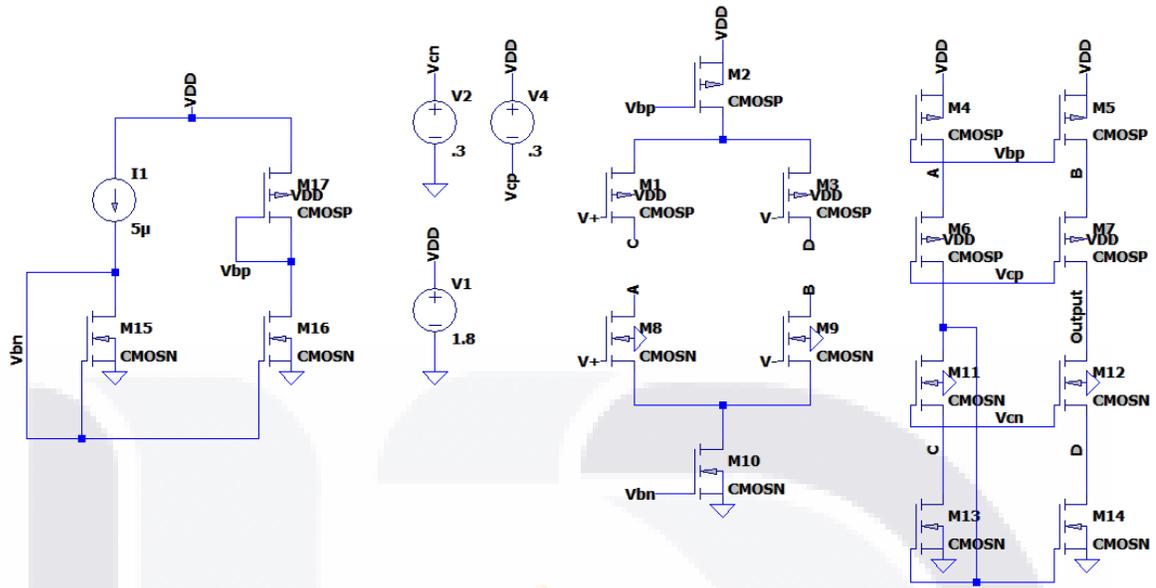


Figura 24 Amplificador Rail to Rail.

Antes de continuar con los resultados de las pruebas antes mencionadas, se aplicó una normalización para comparar la ganancia de un *opamp* (ecuación 52) con el incremento de su área (ecuación 53).

$$por_{area} = (area - area_base) \frac{100}{area_base} \quad (41)$$

Tal que, la variable *area* es el área del transistor con $L \neq L_{min}$, y la variable *area_base* es el área del transistor con L_{min} .

$$por_{gan} = (ganancia - ganancia_base) \frac{100}{ganancia_base} \quad (42)$$

El establecimiento de las variables de la ecuación (53) es análogo a la ecuación (52).

Esta normalización se enfocó al PF_{true} y P_{true} , tal que las variables de las restricciones se establecieron para cubrir el espacio entero de las variables de decisión, así mismo, la normalización se definió para la ecuación (17) y *ang* varió de acuerdo con la ecuación (51). Además, se estableció para encontrar la mínima área y máxima ganancia del *opamp*.

En la siguiente Tabla se presentan los resultados de simulación para obtener por_{gan} y por_{area} del Frente de Pareto y soluciones de Pareto con respecto a las Figuras 19 (Tecnología 180nm) y 20 (Tecnología 90nm), para el amplificador de dos etapas y el amplificador Rail to Rail.

Tabla 14 Resultados de simulación con LTspice XVII para obtener por_{area} y por_{gan} .

Longitudes del MOSFET (um)		.18	.27	.36	.45	.54	.63	.72	.81	.9	.99	1.08	1.17
Tecnología 180nm aplicada al OPAMP 2 etapas	por_{area}	0%	125%	300%	525%	800%	1125%	1500%	1925%	2400%	2925%	3500%	4125%
	por_{gan}	0%	195.202%	272.241%	331.313%	381.986%	427.979%	470.864%	511.454%	550.254%	587.625%	623.837%	659.101%
Tecnología 180nm aplicada al OPAMP rail to rail	por_{area}	0%	125%	300%	525%	800%	1125%	1500%	1925%	2400%	2925%	3500%	4125%
	por_{gan}	0%	217.93%	559.022%	1066.3%	1475.07%	1826.57%	2149.48%	2454.22%	2744.87%	3023.22%	3290.21%	3546.47%
Longitudes del MOSFET (m)		9x10 ⁻⁰⁸	1.35x10 ⁻⁰⁷	1.8x10 ⁻⁰⁷	2.25x10 ⁻⁰⁷	2.7x10 ⁻⁰⁷	3.15x10 ⁻⁰⁷	3.6x10 ⁻⁰⁷	4.05x10 ⁻⁰⁷	4.5x10 ⁻⁰⁷	4.95x10 ⁻⁰⁷	5.4x10 ⁻⁰⁷	5.85x10 ⁻⁰⁷
Tecnología 90nm aplicada al OPAMP 2 etapas	por_{area}	0%	125%	300%	525%	800%	1125%	1500%	1925%	2400%	2925%	3500%	4125%
	por_{gan}	0%	2078.07%	9357.03%	22652.2%	40451.9%	60564.6%	81121.5%	100851%	119013%	135228%	149302%	161068%
Tecnología 90nm aplicada al OPAMP rail to rail	por_{area}	0%	125%	300%	525%	800%	1125%	1500%	1925%	2400%	2925%	3500%	4125%
	por_{gan}	0%	878.996%	1951.96%	2997.91%	3928.44%	4724.52%	5401.21%	5982.98%	6494.24%	6958.97%	7397.55%	7815.45%

Tomando en cuenta que la variable dependiente I_{ds} de la tecnología 90nm decrece cuando crece L , no se tomara en cuenta para obtener el rango de $L = [L_{min}, L_{max}]$. Ahora, de acuerdo con los valores de por_{area} y por_{gan} de la Tabla 14 el $L = [1.5xlim_inf, 2.5xlim_inf]$ para la tecnología 180nm aplicada al *opamp* dos etapas es el rango ideal. Ya que cuando $L_{min} = 1.5xlim_inf = 1.5x180nm = 2.7x10^{-7}m$, los $por_{area} = 125\%$ y $por_{gan} = 195.202\%$, entonces el por_{gan} es mayor por_{area} ; y, cuando $L_{max} = 2.5xlim_inf = 2.5x180nm = 4.5x10^{-7}m$, el por_{area} es algo mayor por_{gan} , esto significa que no conviene aumentar más la longitud (L) del *MOSFET*, ya que esto es peor cuando incrementa L . Estos resultados de simulación concuerdan con el PF_{true} de la Figura 19. Debido a que, la función de ganancia I_{ds} va disminuyendo progresivamente al aumentar L , pero el decrecimiento de R_{ds} es menos notorio que I_{ds} . Para la tecnología 180nm aplicada al *opamp rail to rail* se realizó un proceso análogo al *opamp* dos etapas, obteniendo $L = [1.5xlim_inf, 6xlim_inf]$. Aunque el por_{gan} es aceptable con respecto a por_{area} , se debe tomar en consideración que $L_{max} = 6xlim_inf = 6x180nm = 1.08x10^{-6}m$ es una longitud demasiado grande. Además, con $limite_{I_{ds}} = 70$ de la Figura 21 se obtuvo una $L_{min} = 2.7x10^{-7}m$, lo que concuerda con la L_{min} obtenida con los datos de simulación.

Los valores que se obtuvieron del *THD*, el margen de fase y ganancia con LTspice XVII, a continuación se presentan para ambos *opamps*:

Caso 1) Tecnología 180n aplicada al *opamp* dos etapas:

- A. P_{true} de la ecuación (13) (Figura 21): Con $L = 4.5x10^{-07}$, *margen de ganancia* = $-49.6009db$ y *margen de fase* = 84.5262° ; con $L = 8.1x10^{-07}$, *margen de ganancia*: $-39.5424db$ y *margen de fase* = 82.6317° ; y con $L = 1.44x10^{-06}$, *margen de ganancia*: $-28.2897db$ y *margen de fase* = 79.4856° .
- B. P_{true} de la ecuación (15) (Figura 21): Con $L = 4.5x10^{-07}$, *margen de ganancia* = $-49.6009db$ y *margen de fase* = 84.5262° ; con $L = 7.2x10^{-07}$, *margen de ganancia*: $-41.7175db$ y

margen de fase = 83.1195° ; y con $L = 1.17 \times 10^{-06}$,
margen de ganancia: $-32.4748db$ y *margen de fase* = 80.967° .

- C. P_{true} de la ecuación (26) y (27) (Figura 21): Con $L = 2.7 \times 10^{-07}$, $THD = 3.22537\%$; con $L = 8.1 \times 10^{-07}$, $THD = 3.10088\%$; y con $L = 1.35 \times 10^{-06}$, $THD = 3.05961\%$.

Caso 2) Tecnología 180n aplicada al *opamp rail to rail*:

- A. P_{true} de la ecuación (13) (Figura 21): Con $L = 4.5 \times 10^{-07}$,
margen de ganancia = $-47.1267db$ y *margen de fase* = 84.2365° ;
 con $L = 8.1 \times 10^{-07}$, *margen de ganancia*: $-34.5233db$ y
margen de fase = 67.8689° ; y con $L = 1.44 \times 10^{-06}$,
margen de ganancia: $-24.4505db$ y *margen de fase* = 61.1881° .

- B. P_{true} de la ecuación (15) (Figura 21): Con $L = 4.5 \times 10^{-07}$,
margen de ganancia = $-47.1267db$ y *margen de fase* = 84.2365° ;
 con $L = 7.2 \times 10^{-07}$, *margen de ganancia*: $-36.8745db$ y
margen de fase = 70.023° ; y con $L = 1.17 \times 10^{-06}$,
margen de ganancia: $-27.8938db$ y *margen de fase* = 63.3067° .

- C. P_{true} de la ecuación (26) y (27) (Figura 21): Con $L = 2.7 \times 10^{-07}$, $THD = 4.43532\%$; con $L = 8.1 \times 10^{-07}$, $THD = 3.24515\%$; y con $L = 1.35 \times 10^{-06}$, $THD = 3.24903\%$.

Caso 3) Tecnología 90n aplicada al *opamp dos etapas*:

- A. P_{true} de la ecuación (13) (Figura 21): Con $L = 1.8 \times 10^{-07}$,
margen de ganancia = $-11.1192db$ y *margen de fase* = 62.587° ;
 con $L = 4.05 \times 10^{-07}$, *margen de ganancia*: $-9.89848db$ y
margen de fase = 58.6545° ; y con $L = 6.3 \times 10^{-07}$,
margen de ganancia: $-9.55054db$ y *margen de fase* = 57.5047° .

- B. P_{true} de la ecuación (15) (Figura 21): Con $L = 1.8 \times 10^{-07}$,
margen de ganancia = $-11.0271db$ y *margen de fase* = 62.5432° ;
 con $L = 4.05 \times 10^{-07}$, *margen de ganancia*: $-9.89848db$ y
margen de fase = 58.6545° ; y con $L = 7.2 \times 10^{-07}$,
margen de ganancia: $-9.46119db$ y *margen de fase* = 57.0447° .

C. P_{true} de la ecuación (26) y (27) (Figura 21): Con $L = 1.35 \times 10^{-07}$, $THD = 3.26737\%$; con $L = 4.5 \times 10^{-07}$, $THD = 2.68758\%$; y con $L = 8.1 \times 10^{-07}$, $THD = 2.6341\%$.

Caso 4) Tecnología 90n aplicada al *opamp rail to rail*:

A. P_{true} de la ecuación (13) (Figura 21): Con $L = 1.8 \times 10^{-07}$, *margen de ganancia* = $-60.1947db$ y *margen de fase* = 123.582° ; con $L = 4.05 \times 10^{-07}$, *margen de ganancia*: $-50.6847db$ y *margen de fase* = 100.088° ; y con $L = 6.3 \times 10^{-07}$, *margen de ganancia*: $-46.0254db$ y *margen de fase* = 96.6154° .

B. P_{true} de la ecuación (15) (Figura 21): Con $L = 1.8 \times 10^{-07}$, *margen de ganancia* = $-60.1947db$ y *margen de fase* = 123.582° ; con $L = 4.05 \times 10^{-07}$, *margen de ganancia*: $-50.6847db$ y *margen de fase* = 100.088° ; y con $L = 7.2 \times 10^{-07}$, *margen de ganancia*: $-44.6265db$ y *margen de fase* = 95.8379° .

C. P_{true} de la ecuación (26) y (27) (Figura 21): Con $L = 1.35 \times 10^{-07}$, $THD = 6.75669\%$; con $L = 4.5 \times 10^{-07}$, $THD = 2.86395\%$; y con $L = 8.1 \times 10^{-07}$, $THD = 2.83716\%$.

Como se observa en el caso 1, caso 2, caso 3 y caso 4 en las ecuaciones (13) y (15), a mayor L menor margen de estabilidad. Debido a esto, el área crece, el consumo de potencia incrementa y el ancho de banda decrece. Estos resultados concuerdan con el rango de las restricciones (13) y (15) (P_{true}) de las Figuras 21 y 22. Además, en el caso 1, caso 2, caso 3 y caso 4 en las ecuaciones (26) y (27), cuando L incrementa menor THD , excepto en el inciso C del caso 2, pero la diferencia es muy pequeña en las dos L de mayor magnitud. Los valores del THD concuerdan con la normalización del error relativo de las ecuaciones (26) y (27), como se observa en las Figuras 21 y 22.

Conclusiones

La tecnología *CMOS* ha estado en constante evolución tratando de mejorar el desempeño de los circuitos integrados. Estos circuitos están incorporados en el hardware de diversos dispositivos electrónicos (médicos, móviles, telecomunicaciones, etcétera); con lo cual, se verá beneficiado el usuario final en sus actividades laborales, escolares, sociales, etcétera, si se logra obtener el mayor desempeño de los procesos *CMOS*.

Debido al conjunto de funciones objetivo y restricciones que se obtuvieron por la degradación de los efectos de canal corto al correcto funcionamiento del *MOSFET*, se pudieron obtener diferentes soluciones de Pareto y frentes de Pareto. Estos conjuntos de soluciones de Pareto y frentes de Pareto se utilizaron para realizar diferentes pruebas, tales como determinar la ganancia de un *opamp* dependiendo del incremento de su área, el *THD* del *opamp*, y el margen de fase y ganancia del *opamp*. Estas pruebas se efectuaron en los *opamps* dos etapas y *rail to rail* con la tecnología 180nm (BSIM3) y con la tecnología 90nm (BSIM4), y se percató que cuando incrementa el área del *MOSFET* la ganancia del *opamp* incrementa de manera aceptable, hasta cierto rango de *L*. Además, cuando incrementa el área la respuesta del *opamp* es más lineal (*THD*) y el margen de estabilidad del *opamp* es menor.

También, se comprobó que los diferentes PF_{true} y P_{true} obtenidos con la arquitectura de optimización, la cual incluye el *MOGA* concuerdan con estas pruebas. Por lo tanto, el conjunto de funciones objetivo y restricciones se definieron de manera correcta a partir de los principales efectos de canal corto del *MOSFET*, para poder optimizar ambas tecnologías del *MOSFET* con diferentes modelos de los efectos de canal corto (BSIM3 y BSIM4), de acuerdo con los dos *opamps* en los que se aplicaron estas tecnologías del *MOSFET*.

La complejidad computacional de la arquitectura de optimización fue aceptable tomando en cuenta el gran consumo recursos computacionales que efectúa el simulador Ngspice. Esta complejidad computacional admisible se debió, a la obtención del error de truncamiento para los incrementos igualmente espaciados de V_{ds} , y por las pruebas de convergencia que se aplicaron a la arquitectura de optimización para la obtención de los parámetros del *MOGA* que consumieran menos recursos computacionales.

El PF_{true} y P_{true} obtenidos del *MOSFET* de tecnología 180nm y 90nm concuerdan con las pruebas que se le realizaron a los dos *opamps*. Debido a esto se puede tener un resumen de soluciones de Pareto y frentes de Pareto aceptables, con estas soluciones se puede elegir un rango de L de acuerdo con los criterios de diseño del *opamp*. Por lo tanto, este rango de L se puede aplicar a una gran cantidad de circuitos analógicos con diseños específicos, que se incorporarían al hardware de aparatos electrónicos; beneficiando al usuario final en sus actividades cotidianas.

El factor de escala C se mantuvo constante durante el proceso de optimización del transistor *MOSFET*, puesto que los transistores deberían mostrar las mismas características eléctricas. Sin embargo, falta comprobar que resultados se obtendrían al variar C en el proceso de optimización. Debido a esto, una futura investigación sería obtener el rango de L al variar C y con esto comparar los resultados de ambas arquitecturas de optimización (C constante y C variable).

Referencias

- Abdel-Bary, Y. M., & Khalil, A. I. (2003, 27-30 Dec. 2003). *Optimization technique for low phase-noise differential LC VCO design*. Paper presented at the 2003 46th Midwest Symposium on Circuits and Systems.
- Aggarwal, V., & O'Reilly, U.-M. (2007). Design of Posynomial Models for Mosfets: Symbolic Regression Using Genetic Algorithms. In R. Riolo, T. Soule & B. Worzel (Eds.), *Genetic Programming Theory and Practice IV* (pp. 219-236). Boston, MA: Springer US.
- Arcelli, D., Cortellessa, V., Emidio, M. D., & Pompeo, D. D. (2018, 30 April-4 May 2018). *EASIER: An Evolutionary Approach for Multi-objective Software Architecture Refactoring*. Paper presented at the 2018 IEEE International Conference on Software Architecture (ICSA).
- Baek, K., Gim, J., Kim, H., Na, K., Kim, N., & Kim, Y. (2013). Analogue circuit design methodology using self-cascode structures. *Electronics Letters*, 49(9), 591-592.
- Ben G. Streetman, S. K. B. (2015). SOLID STATE ELECTRONIC DEVICES. In P. Education (Ed.), (Septima ed., pp. 596). United States of America: Pearson Education.
- Bencherif, H., Dehimi, L., Pezzimenti, F., De Martino, G., & Della Corte, F. G. (2019). Multiobjective Optimization of Design of 4H-SiC Power MOSFETs for Specific Applications. *Journal of Electronic Materials*, 48(6), 3871-3880.
- Bendib, T., & Djefal, F. (2013). Multi-Objective-Based Approach to Optimize the Analog Electrical Behavior of GSDG MOSFET: Application to Nanoscale Circuit Design. In G.-C. Yang, S.-I. Ao & L. Gelman (Eds.), *IAENG Transactions on Engineering Technologies: Special Volume of the World Congress on Engineering 2012* (pp. 315-325). Dordrecht: Springer Netherlands.
- Bera, S., Mondal, C., & Biswas, A. (2017). Development of a methodology for the extraction of BSIM3v3.2.2 parameters of Ge-channel MOSFETs and estimation of analog circuit performance. *Microsystem Technologies*, 23(9), 4123-4131.
- Canale, S. C. C. y. R. P. (2015). *Métodos numéricos para ingenieros* (Séptima ed.).
- Cantürk, I., & Kahraman, N. (2015, 9-11 July 2015). *Comparative analog circuit design automation based on multi-objective evolutionary algorithms: An application on CMOS opamp*. Paper presented at

- the 2015 38th International Conference on Telecommunications and Signal Processing (TSP).
- Cao, Y. (Kevin). (n.d.). *Nanoscale Integration and Modeling (NIMO) Group, ASU*.
- Chen, Y., Mohamed, M., Jo, M., Ravaioli, U., & Xu, R. (2013). Junctionless MOSFETs with laterally graded-doping channel for analog/RF applications. *Journal of Computational Electronics*, 12(4), 757-764.
- Cheung, W., & Wong, N. (2006, 12-15 Dec. 2006). *Optimized RF CMOS Low Noise Amplifier Design Via Geometric Programming*. Paper presented at the 2006 International Symposium on Intelligent Signal Processing and Communications.
- Coello, C. A. C., Lamont, G. B., & Veldhuizen, D. A. V. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second Edition ed.). USA: Springer, Boston, MA.
- Coello Coello, C. A. (1999). A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3), 269-308.
- Cui, J., Zheng, Q., Ning, B., Yu, X., Zhao, K., Wei, Y., et al. (2018, 16-20 July 2018). *Hot-Carrier Effect on TID Irradiated Short-Channel UTTB FD-SOI n-MOSFETs*. Paper presented at the 2018 IEEE Nuclear & Space Radiation Effects Conference (NSREC 2018).
- Das, I., & Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1), 63-69.
- Deb, K. (2008). Introduction to Evolutionary Multiobjective Optimization. In J. Branke, K. Deb, K. Miettinen & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (pp. 59-96). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Deb, K. and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In Schaffer, J. D., Editor, *Proc. Third Int. Conf. on Genetic Algorithms*, Pp. 42-50. Morgan Kaufmann.
- Deb, K., & Agrawal, R. B. (1995). Simulated Binary Crossover for Continuous Search Space. *Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, UP 208 016, India*.

- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*.
- Dellino, G., Fedele, M., & Meloni, C. (2011). Dynamic Objectives Aggregation Methods in Multi-objective Evolutionary Optimization. In N. Nedjah, L. dos Santos Coelho, V. C. Mariani & L. de Macedo Mourelle (Eds.), *Innovative Computing Methods and Their Applications to Engineering Problems* (pp. 85-103). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Djeffal, F., & Ferhati, H. (2016). A new high-performance phototransistor design based on both surface texturization and graded gate doping engineering. *Journal of Computational Electronics*, 15(1), 301-310.
- Ehrgott, M. (2005). *Multicriteria optimization* (Second edi). Springer Berlin.
- Engelbrecht, A. P. (2007). *Computational Intelligence An Introduction* (Second Edition ed.).
- Fonseca, C. M. and P. J. F. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*(S.Forrest, Ed.), San Mateo, CA: Morgan Kaufmann, July 1993.
- Frank, Laux, & Fischetti. (1992, 13-16 Dec. 1992). *Monte Carlo simulation of a 30 nm dual-gate MOSFET: how short can Si go?* Paper presented at the 1992 International Technical Digest on Electron Devices Meeting.
- Greco, G., & Rinaudo, S. (2012, 2012//). *Automatic Layout Optimization of Power Discrete Devices Using Innovative Distributed Model Techniques*. Paper presented at the Progress in Industrial Mathematics at ECMI 2010, Berlin, Heidelberg.
- Hinterding, R. (2000, 16-19 July 2000). *Representation, mutation and crossover issues in evolutionary computation*. Paper presented at the Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512).
- Holger Vogt, M. H., Paolo Nenzi. (2018). Ngspice Users Manual Version 28plus (Describes ngspice master branch version).
- Huang, C., & Li, Y. (2015, 9-11 Sept. 2015). *Electrical characteristic of InGaAs multiple-gate MOSFET devices*. Paper presented at the 2015 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD).

- Jaafar, H., Aouaj, A., Bouziane, A., & Iñiguez, B. (2018, 26-27 April 2018). *Analytical study of drain current and transconductance for a new cylindrical gate MOSFET structure*. Paper presented at the 2018 4th International Conference on Optimization and Applications (ICOA).
- Jafari, A., Zekri, M., Sadri, S., & Mallahzade, A. (2010, 19-21 March 2010). *Design of Analog Integrated Circuits by Using Genetic Algorithm*. Paper presented at the 2010 Second International Conference on Computer Engineering and Applications.
- Jin, X., & Hoe, D. H. K. (2012, 5-8 Aug. 2012). *Optimization of short channel CMOS LNAs by geometric programming*. Paper presented at the 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS).
- Kotti, M., Benhala, B., Fakhfakh, M., Ahaitouf, A., Benlahbib, B., Loulou, M., et al. (2011, 19-22 Dec. 2011). *Comparison between PSO and ACO techniques for analog circuit performance optimization*. Paper presented at the ICM 2011 Proceeding.
- Lam, S., & Chan, M. (2013, 5-8 Aug. 2013). *Effect of parasitic capacitances and resistances on the RF performance of nanoscale MOSFETs*. Paper presented at the 2013 13th IEEE International Conference on Nanotechnology (IEEE-NANO 2013).
- Liu, M., & He, J. (2009, 23-24 May 2009). *A Hybrid Genetic Algorithm with Hyper-Mutation and Elitist Strategies for Automated Analog Circuit Design*. Paper presented at the 2009 International Workshop on Intelligent Systems and Applications.
- Maji, K. B., Jaiswal, H., Kar, R., Mandal, D., & Ghoshal, S. P. (2015, 4-6 Nov. 2015). *Opposition Harmony Search algorithm based optimal sizing of CMOS analog amplifier circuit*. Paper presented at the 2015 International Conference on Science and Technology (TICST).
- Mattausch, H. J., Sadachika, N., Yumisaki, A., Kaya, A., Imafuku, W., Johguchi, K., et al. (2009). *Correlating Microscopic and Macroscopic Variation With Surface-Potential Compact Model*. *IEEE Electron Device Letters*, 30(8), 873-875.
- Mead, C., & Conway, L. (1978). *Introduction to VLSI Systems*.
- Melik-Adamyanyan, A. F. (2009). *Application of genetic algorithms in problems of optimization of the physical design in microelectronics*. *Automatic Documentation and Mathematical Linguistics*, 43(4), 244-250.

- Mukhopadhyay, A., Bhattacharya, S., Iizuka, T., Maiti, T. K., Miura-Mattausch, M., Gau, A., et al. (2018, 29-31 March 2018). *MOSFET optimization toward power efficient circuit design*. Paper presented at the 2018 International Symposium on Devices, Circuits and Systems (ISDCS).
- Neamen, D. A. (2012). *Semiconductor physics and devices*. New York, NY: McGRAW-HILL.
- Neil H. E. Weste, D. M. H. (2011). *CMOS VLSI Design a circuits and systems perspective* (FOURTH ed.). United States of America: ADDISON WESLEY.
- NGSPICE. (n.d.). <http://ngspice.sourceforge.net/>
- Preuss, M. (2015). Experimentation in Evolutionary Computation. In M. Preuss (Ed.), *Multimodal Optimization by Means of Evolutionary Algorithms* (pp. 27-54). Cham: Springer International Publishing.
- Razavi, B. (2001). *Design of Analog CMOS Integrated Circuits* (International Edition ed.). United States: McGraw-Hill.
- Rjoub, A., Taradeh, N. A., & Al-Mistarihi, M. F. (2014, 29 Sept.-1 Oct. 2014). *Gate leakage current accurate models for nanoscale MOSFET transistors*. Paper presented at the 2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS).
- Sallem, A., Pereira, P., Fakhfakh, M., & Fino, H. (2013, 2013//). *A Multi-objective Simulation Based Tool: Application to the Design of High Performance LC-VCOs*. Paper presented at the Technological Innovation for the Internet of Things, Berlin, Heidelberg.
- Shuka, R., & Brehm, J. (2019, 2019//). *A Parallel Adaptive Swarm Search Framework for Solving Black-Box Optimization Problems*. Paper presented at the Architecture of Computing Systems – ARCS 2019, Cham.
- Spivak, M. (1996). *Cálculo Infinitesimal* (Segunda ed.). México, D.F.
- Srivastava, V. M. (2015, 17-20 Dec. 2015). *Design optimization of high-k dielectric based double-gate MOSFET and its performance*. Paper presented at the 2015 Annual IEEE India Conference (INDICON).
- Veldhuizen, D. A. Van. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*.
- Weixun, Y., Kolm, R., & Zimmermann, H. (2008, 18-21 May 2008). *A low-voltage low-power fully differential rail-to-rail input/output*

opamp in 65-nm CMOS. Paper presented at the 2008 IEEE International Symposium on Circuits and Systems.

