BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

"IMPLEMENTACIÓN DEL MODELO DE OTFTS EN UN SIMULADOR LIBRE SPICE PARA EL DISEÑO DE CIRCUITOS DIGITALES/ANALÓGICOS"

TESIS

QUE PARA TENER EL GRADO DE:
LICENCIADO EN ELECTRÓNICA

PRESENTA:
ELIENAI SIMÓN MIER

BAJO LA ASESORÍA DE:
DR. JOSÉ JOAQUÍN ALVARADO PULIDO

DR. VÍCTOR RODOLFO GONZÁLEZ DÍAZ

DRA. BLANCA SUSANA SOTO CRUZ

PUEBLA, PUE., JUNIO 2016

AGRADECIMIENTOS

Doy mis más sinceros agradecimientos a mis asesores de tesis el Dr. Joaquín Alvarado, la Dra. Susana Soto y al Dr. Víctor Rodolfo González por su gran apoyo durante este trabajo de tesis, gracias a la paciencia y a todos sus conocimientos que me han compartido y que me han servido para seguir desarrollándome en el ámbito profesional y personal.

También agradezco enormemente a mi familia, a mi papá y mi mamá, ya que sin su apoyo moral y económico esto no habría sido posible, no olvidando también agradecer a mis hermanos que me han servido de ejemplo para luchar cada día más y jamás darme por vencida y además ser un ejemplo para la sociedad, muchas gracias.

Sin duda alguna agradezco a mis amigos, compañeros y a todas aquellas personas que siempre me han apoyado y me han dado ánimos para seguir adelante, además de siempre ser solidarios y amistosos, muchas gracias por compartir su conocimiento y tiempo siempre que les fue posible, gracias a todos.

RESUMEN

Actualmente el poder fabricar dispositivos electrónicos a bajo costo es uno de los mayores retos que los investigadores se plantean. Una opción viable es la utilización de materiales alternos como los polímeros que generalmente se proponen (PMMA, P3HT y PCDTBT) para la construcción de un transistor de este tipo. Sin embargo la baja movilidad de los portadores limita su utilización en ciertas aplicaciones como son: el procesamiento y comunicaciones. Esta necesidad lleva al uso de modelos computacionales para la predicción del desempeño de estos dispositivos. De esta forma, se explora su uso en diversos tipos de aplicaciones, con las ventajas del el uso de software libre.

El presente trabajo tiene como objetivo la implementación del modelo analítico de los transistores orgánicos de capa delgada (OTFT por sus siglas en inglés) en un simulador de circuitos tipo SPICE de acceso libre y de esta forma realizar caracterizaciones a nivel simulación con transistores orgánicos.

Con esta implementación será posible brindar el acceso a toda aquella persona que quiera estudiar el comportamiento de este tipo novedoso de dispositivos así como desarrollar simulaciones de circuitos digitales y analógicos para su posterior fabricación.

Por lo tanto en este trabajo de tesis trabajaremos con simulación de circuitos a través de un modelo analítico el cual ya fue validado y existen resultados disponibles sobre la simulación y fabricación. Se procederá a implementar dicho modelo en el simulador de circuitos de acceso libre NGspice para posteriormente poder llevar a cabo la simulación de un oscilador de anillo y así compararlo con lo que se tiene en la literatura. Posteriormente se trabajará en la simulación de un amplificador con entrada diferencial con salida única (*single-ended*), el cual no se encuentra reportado en la literatura hasta el momento, ni en la simulación con el modelo que utilizaremos. Finalmente, se validarán los resultados de la simulación realizada comparando las simulaciones en SmartSpice (software comercial) y NGspice (software libre) con la finalidad de validar los resultados de la implementación.

En conjunto con la implementación del modelo de OTFTS en un software libre daremos a conocer los medios necesarios para poder llevar a cabo la simulación y así ofrecer la herramienta para desarrollar nuevos dispositivos con OTFTS a través de las celdas básicas que se desarrollen con el software libre. Por lo que realizaremos simulaciones en DC, AC y en el tiempo las cuales nos permitan

demostrar la confiabilidad haciendo la comparación con datos disponibles en artículos.

De acuerdo a lo mencionado anteriormente el presente trabajo sólo involucra el modelo de un transistor, y sobre el mismo podremos hacer variaciones de algunos parámetros con el fin de poder visualizar mejoras en los circuitos digitales y analógicos de tal manera que las simulaciones no presenten problemas de convergencia con el modelo implementado ya que es muy común tener estos problemas en el software libre.

Contenido

Introducción **7**

Material Orgánico como Semiconductor *8*

CAPÍTULO 1. Antecedentes **10**

1.1 Materiales *10*

1.2 Transistores Orgánicos *13*

1.2.1 Transistores de Efecto de Campo FET *13*

1.2.2 Clasificación de FETS *14*

1.2.2.1 MOSFET *15*

1.2.2.2 TFT *16*

1.2.3 Transistores Orgánicos de capa delgada OTFT *17*

CAPÍTULO 2. Modelo Analítico del OTFT **21**

2.1 Modelo de Corriente-Voltaje *21*

2.1.1 Descripción del Modelo UMEM *23*

2.2 Modelo de Capacitancia-Voltaje *27*

2.2.1 Modelo de Carga *28*

2.2.2 Modelo de Capacitancia *29*

CAPÍTULO 3. Implementación en NGSPICE **31**

3.1 Software comercial y libre *31*

3.1.1 SmartSpice de Silvaco ® *31*

3.1.2 NGspice *32*

3.2 Instalación de NGspice *32*

3.2.1 Requerimientos previos *33*

3.2.2 Instalación *34*

3.3 Implementación del modelo analítico de OTFTS *35*

3.3.1 Metodología de compilación y adaptación del modelo *36*

CAPÍTULO 4. Resultados de simulación en NGspice **44**

4.1 Oscilador de anillo *44*

4.1.1 Simulación del OTFT tipo P *45*

4.1.2 Simulación del oscilador de anillo *47*

4.2 Amplificador diferencial en NGspice *51*

4.2.1 Amplificador de una etapa *52*

4.2.1.1 Espejo de corriente *53*

4.2.2 Amplificador diferencial con espejo de corriente *54*

4.2.2.1 Simulación del Amplificador en NGspice *55*

CAPÍTULO 5. Conclusiones **59**

Anexo A: Modelo UMEM en Verilog-A **61**
Anexo B: Modelo de hicum0.va **65**
Anexo C: Metodologías para la compilación del modelo **66**
Anexo D: Modelo UMEM modificado **69**
Anexo E: Netlist de SmartSpice y NGspice para el OTFT tipo P **75**
Anexo F: Netlist del oscilador de anillo **76**
Anexo G: Respuesta transitoria del oscilador de anillo en NGspice **77**
Anexo H: Netlist del amplificador diferencial con salida única **78**
Referencias **80**

INTRODUCCIÓN

Desde que fue posible fabricar un transistor de efecto de campo (FET por sus siglas en inglés) a principios de 1960 y que gracias a Jack Kilby de Texas Instruments fue posible fabricar circuitos integrados (CIs) en el año de 1961, se hizo necesario una herramienta que permitiera a una persona hacer uso de estos transistores para diseñar un CI, así como el poder comprobar previo a la fabricación, las interconexiones de varios transistores para realizar una función específica. Como resultado de esta necesidad, fue en 1973 cuando se presentó al público la primera versión de un simulador de circuitos llamado SPICE [1].

Desde aquel momento y hasta nuestros días varias versiones de SPICE han sido lanzados al mercado, los cuales además de contener mejoras en su proceso, contienen también un mayor número de modelos eléctricos o analíticos de los diferentes tipos de dispositivos (ej. diodos, transistores de doble juntura, resistencias, etc.), incluyendo obviamente a los diferentes tipos de FETs que han venido surgiendo gracias a los avances de la tecnología, así como a los diferentes tipos de aplicaciones (alto rendimiento, bajo rendimiento, alta potencia, bajo voltaje-baja potencia, etc.).

Es debido a lo anterior que en el presente trabajo se aborda la implementación de un modelo analítico totalmente desarrollado para transistores de películas delgadas (TFT por sus siglas en inglés) [2] en un simulador libre tipo SPICE. En específico, el trabajo se concentra en TFTs desarrollados con materiales orgánicos, los cuales presentan características que los hacen atractivos para aplicaciones donde no es necesario una gran conductividad y que además cubran grandes áreas, que sean flexibles, con posibilidad de ser biocompatibles y que sean de bajo costo de fabricación.

Gracias a la naturaleza física del modelo desarrollado por [2], el cual fue posteriormente incluido en un simulador SPICE comercial [3] ha sido posible realizar diseño de circuitos tanto digitales como analógicos. Desafortunadamente, si una institución, profesor, estudiante o entusiasta diseñador que quisiera realizar circuitos con este tipo de transistores, se vería limitado si no contara con una licencia de este software comercial. Es entonces donde este trabajo propone contribuir al diseño de este tipo de transistores y dotar a los interesados de un modelo implementado y validado para la simulación eléctrica de circuitos digitales y analógicos libre de cualquier pago.

De esta forma será posible explotar el diseño de aplicaciones realizados con este tipo de transistores, la cual impulsará no sólo un mayor conocimiento del funcionamiento de estos dispositivos sino su optimización tanto en el proceso de fabricación como en la arquitectura de éste. En consecuencia, podrá ser posible la creación de nuevas metodologías para el diseño de circuitos integrados orgánicos, ya que su principio de funcionamiento no es el mismo que los circuitos comerciales que típicamente son basados en silicio.

Material Orgánico como Semiconductor

Recientemente los dispositivos flexibles han cobrado un creciente interés, debido a que éstos pueden cubrir diferentes tipos de aplicaciones a un bajo costo [4, 5]. En este sentido, los dispositivos realizados con materiales orgánicos, se han venido obteniendo mediante procesos de depósito sencillos y de menor costo en comparación con otros.

Regularmente estos tipos de materiales presentan ciertas características que son similares a los semiconductores inorgánicos (como es el caso del silicio), debido a los enlaces de sus moléculas las cuales se relacionan a uniones conjugadas π (pi), que permiten a los electrones moverse libremente en los recubrimientos de nubes de electrones π haciendo posible la conducción eléctrica [6, 7]. De tal forma que estos dispositivos orgánicos pueden ser ocupados en aplicaciones como LEDs [7], biosensores [6] así como en celdas solares y transistores de capas finas llamados OTFTS [8].

En el caso de los OTFTS los materiales orgánicos pueden ser ocupados como semiconductor en estructuras metal dieléctrico semiconductor (MDS), en conjunto con aislantes igualmente orgánicos. Con estas estructuras, actualmente, se han podido obtener transistores OTFTS tanto tipo P como N con muy buenas características eléctricas sin sufrir degradación en éstas después de un tiempo de haber sido fabricados, siempre y cuando se seleccionen materiales orgánicos que no pierdan sus cualidades, así como las estructuras y el proceso con el que se fabriquen estos transistores.

Esta tesis, está divida en 5 capítulos y 8 anexos. En el capítulo 1 se da a conocer conceptos generales acerca de los materiales que son utilizados en los diferentes transistores, de igual manera se explican los OTFTS.

Subsecuentemente en el capítulo 2, se describe el modelo UMEM para OTFTS a través de los modelo de corriente-voltaje, modelos de carga y capacitancia.

La implementación del modelo UMEM se hace en el capítulo 3 dando así a conocer el software que se utilizó y los requerimientos necesarios para poder hacer uso del mismo, por otra parte, también se da a conocer la metodología de compilación y adaptación para llevar a cabo la implementación.

Posteriormente en el capítulo 4, se muestran los resultados de simulación obtenidos para el circuito digital y analógico en el software libre, además de que se incorporan los resultados disponibles para hacer las respectivas comparaciones del modelo que se implementó, con el que ya está validado en el software comercial.

Finalmente en el capítulo 5, se presentan las conclusiones de este trabajo de tesis y también se menciona el trabajo a futuro que de ser posible estaría llevándose a cabo para tener la validación en tanto la parte digital como la parte analógica.

Para más detalle acerca del modelo UMEM en Verilog-A y de los Netlist de los circuitos simulados, se otorgan los anexos en la sección final de esta tesis.

CAPÍTULO 1

ANTECEDENTES

1.1 Materiales

Los polímeros han sido aplicados durante mucho tiempo como dieléctricos. En realidad, no hace mucho tiempo, toda la conducción eléctrica en polímeros – en su mayoría debido a iones débilmente ligados - fue considerada como un fenómeno indeseable [9]. Aunque la conductividad iónica de electrolitos poliméricos y poli-electrolitos ha sido ampliamente utilizadas en sistemas electroquímicos (por ejemplo en las fuentes de energía, sensores y el desarrollo de dispositivos de estado sólido electroquímico), la aparición de la electrónica de polímeros conductores ha abierto nuevas perspectivas en el uso de los mismos y su investigación desde el punto de vista químico y físico [10]. Esta aparición comienza en los años 70's, cuando una nueva clase de polímeros con alta conductividad electrónica, en su estado de oxidación parcial, fueron descubiertos.

Los científicos: Alan J. Heeger, Alan G. MacDiarmid y Hideki Shirakawa, desempeñaron un papel importante en este avance, recibiendo más tarde el Premio Nobel de química en el año 2000 "por el descubrimiento y desarrollo de polímeros electrónicamente conductores" [11, 12].

Como en muchos otros casos en la historia de la ciencia, hubo varios precursores de este descubrimiento, incluyendo teorías hechas por físicos y químicos, así como diferentes polímeros conductores que ya habían sido preparados. Por ejemplo en 1862, Henry Letheby preparó polianilina por el mecanismo de la oxidación anódica de la anilina, resultando ser conductivo y mostrar comportamiento electrocrómico [13]. Sin embargo, la preparación del poli-acetileno por parte de Shirakawa y colaboradores, así como el incremento de su conductividad después de cierto "dopaje" hecho por el grupo de MacDiarmid y Heeger, [14, 15] pusieron en marcha este nuevo campo de investigación.

Durante unos diez años después del descubrimiento del poli-acetileno conductor, la investigación en materiales conductores orgánicos se mantuvo en el límite entre

la química y la física del estado sólido. A mediados de los 80's el interés de los ingenieros y físicos aplicados en esta tecnología aumentó drásticamente.

En 1982, fue presentada una estructura capacitiva del tipo metal-óxido-semiconductor usando un semiconductor orgánico, por parte de Ebisawa, Kurokawa y Nara. El dispositivo se fabricó utilizando poli-acetileno como semiconductor sobre un polisiloxano como dieléctrico de compuerta y usando aluminio como electrodo de compuerta y oro como electrodos de fuente y drenador [16]. En 1987, C. W. Tang y Van Slyke realizaron un importante avance al mostrar un Diodo Emisor de Luz (LED) fabricado con dos moléculas orgánicas basadas en los compuestos: tri (8- hidroxiquinolina) (Alq3) y la diamina [17]. La fabricación de este dispositivo es considerado por muchos como el inicio real de la orgánica electrónica, ya que estos OLEDs, fueron los primeros dispositivos electrónicos basados en semiconductores orgánicos. A partir de 1987, los grupos de investigación comenzaron a concentrarse en la parte teórica, así como en aspectos más aplicados a la conductividad eléctrica en moléculas orgánicas. En ese mismo año H. Koezuka presentó el primer transistor OTFT, utilizando politiofeno polimerizado electroquímicamente como capa semiconductora [18]. Un año después, J. H. Burroughes publicó los primeros resultados de diodos Schottky y transistores orgánicos a base de poliacetileno como semiconductor [19].

A finales de los 80', los materiales orgánicos conductores de molécula pequeña (oligómeros) en vez de polímeros, fueron utilizados por algunos grupos de investigadores en dispositivos electrónicos. En realidad los oligómeros fueron utilizados inicialmente con el fin de entender mejor los dispositivos a base de polímeros [20]. Este tipo de moléculas como α-hexitienil y otras se presentaron como capas semiconductoras de OTFTS [21, 22]. En contraste con los materiales poliméricos, los oligómeros eran insolubles en disolventes, por lo que fueron introducidos los métodos de deposición a base de sublimación a alto vacío para obtener películas delgadas de semiconductor. Los valores más altos de movilidad de efecto de campo en este periodo, estaban en el intervalo de 10^{-4} – 10^{-3} $cm^2/V*s$. La molécula pequeña de pentaceno, que hoy en día se utiliza ampliamente como referencia de semiconductor orgánico, fue utilizada por primera vez en 1992 y en esa época mostró una movilidad de efecto de campo de $2*10^{-3}$ $cm^2/V*s$ [23]. Como se puede ver en la Figura 2.1, el 1,3 pentaceno es una molécula que consta de cinco anillos de benceno fusionados.

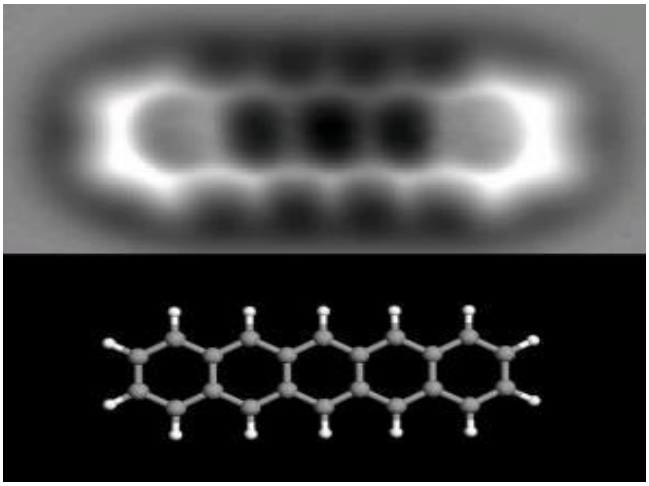


Figura 2.1. Molécula de Pentaceno [24].

Todos los semiconductores orgánicos mencionados anteriormente son semiconductores tipo P. En 1995, fue fabricado un OTFT usando un semiconductor tipo N, el C60 (Buckminsterfuller, conocido también como “bucky esferas”), dando como resultado una movilidad de efecto de campo de casi 10^{-1} $cm^2/V*s$ [25].

La electroquímica ha sido fundamental en la preparación y caracterización de estos novedosos materiales. Las técnicas electroquímicas han sido ajustadas para controlar la síntesis de estos compuestos así como el ajuste de estados de oxidación bien definidos de las moléculas. La preparación, caracterización y aplicación de actividad electroquímica en sistemas poliméricos son aún vigentes en la investigación electroquímica. Hay por lo menos dos razones principales de estas investigaciones: primero, la curiosidad de investigaciones interesados en comprender el comportamiento de estos sistemas, en particular los mecanismos de transferencia de carga y los procesos de transporte de carga que ocurren durante la reacción de óxido-reducción de materiales poliméricos conductores. La segunda razón de estudio, es la amplia gama de compuestos en el campo del almacenamiento de energía, electrocatálisis, electroquímica orgánica, bioelectroquímica, fotoelectroquímica, electroanálisis, sensores, pantallas electrocromáticas, microsistemas, dispositivos electrónicos, protección corrosiva, etcétera. Después de 40 años de investigación, los avances han sido numerosos e importantes, dentro de los cuales destaca el incremento en el desarrollo y fabricación de dispositivos electrónicos con una gran variedad de aplicaciones.

1.2 Transistores Orgánicos

Sin lugar a duda podemos decir que uno de los inventos más trascendentales del siglo XX ha sido el transistor. Desde que John Bardeen, William Shockley y Walter Brattain inventaron el primer transistor en 1947, comenzó el desarrollo revolucionario de la electrónica de semiconductores. El concepto de Transistores de Efecto Campo (FET) fue propuesto por primera vez por el físico austrohúngaro Julius Lilienfeld (J. E Lilienfeld, US Patent 1745175, 1930) y en la actualidad, este dispositivo, tienen el dominio de la industria de la microelectrónica convencional. Ellos son los pilares fundamentales para la base de circuitos digitales tales como sumadores, inversores y unidades de lógica aritmética (ALU), también se utilizan para construir los circuitos lógicos secuenciales como los flip-flops. Además, los FET son esenciales en los dispositivos de memoria modernos, circuitos integrados y los microprocesadores utilizados en computadoras personales y portátiles. En circuitos analógicos también tienen gran aplicación en amplificadores, comparadores, sensores, etcétera.

1.2.1 Transistores de Efecto de Campo FET

Los FETs son dispositivos de al menos tres terminales, en donde la corriente entre dos de estos es controlada a través de un campo eléctrico inducido por el voltaje aplicado en algunas de sus terminales.

Los FETs normalmente funcionan como un condensador; compuesto de dos placas, donde una de ellas funciona como un canal de conducción entre dos contactos óhmicos denominados Drenador (D) y una Fuente (S). La otra placa trabaja para controlar la carga inducida en el canal y se llama la compuerta (G) como se puede ver en la Figura 2.2. La dirección del movimiento de los portadores de carga en el canal, es de la fuente al drenador. Por lo tanto la relación entre estos tres componentes es que la compuerta controla el movimiento de portadores entre la fuente y oxido de la compuerta del drenador.

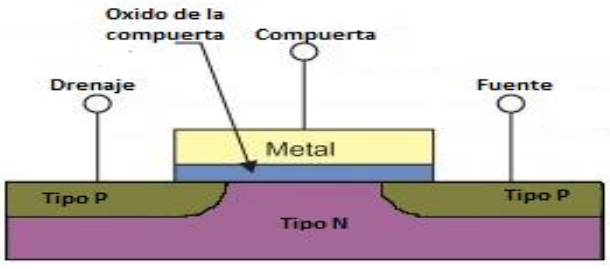


Figura 2.2. Estructura de un MOSFET [26].

Cuando este concepto de capacitor es aplicado al diseño de dispositivos, varias configuraciones pueden ser construidas en función de la diferencia en el control de la compuerta. Esto puede ser en el material de la compuerta, la ubicación de la compuerta con respecto al canal, cómo la compuerta está aislada del canal, y que tipo de portadores son inducidos por el voltaje de compuerta en el canal (ya sea electrones en un dispositivo de canal n, como huecos en un dispositivo de canal p y ambos, electrones y huecos en un dispositivo de doble inyección).

1.2.2 Clasificación de FETs

Clasificados por cómo se forma la compuerta del capacitor, encontramos tres tipos de FETs mostrados en la primera fila del diagrama 2.1:

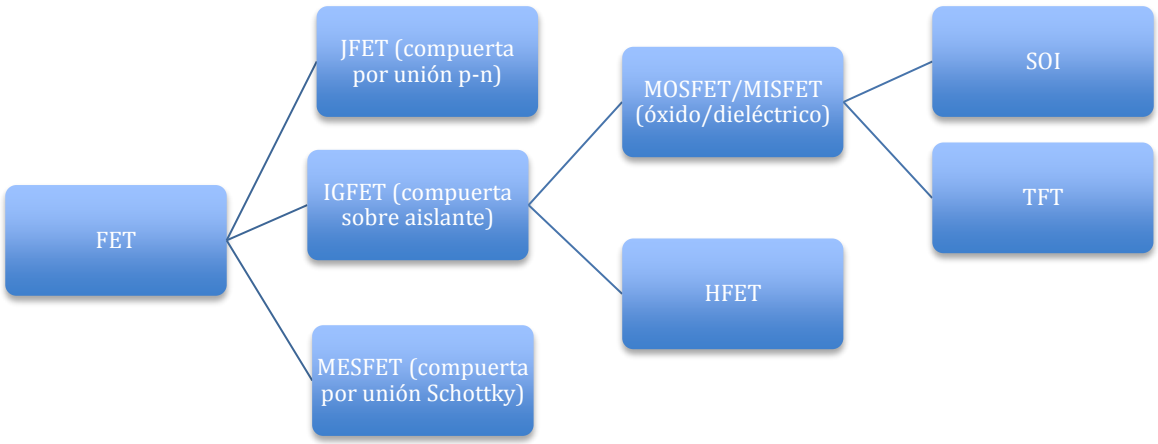


Diagrama 2.1 Clasificación de los FETs

Los FET de compuerta aislada (IGFET) que tienen una capa dieléctrica entre el contacto de compuerta y el canal; FET de unión (JFET) donde el capacitor es formado por la capa de agotamiento de una unión P-N y el FET de unión metal semiconductor (MESFET) donde la capa de agotamiento es debida a una barrera Schottky (ver Figura 2.3 b).

Dentro de la clasificación de IGFET tenemos los FET de Metal-Óxido-Semiconductor (MOSFET) donde específicamente tenemos una capa de óxido crecido, a comparación del Metal-Aislante-Semiconductor MISFET donde el dieléctrico es un material depositado sobre la estructura (ver Figura 2.3 a). En la rama del FET de Heterounión (HFET), el material de compuerta es una capa de

semiconductor con un Eg muy alto, crecido como una heterounión el cual actúa como dieléctrico.

Como sub clasificación de los MOSFET/MISFET, tenemos los FET de Silicio Sobre Dieléctrico (SOI). Es una estructura en la cual una capa superior de semiconductor, generalmente silicio cristalino de alta calidad, es formada encima de un dieléctrico que puede ser una “caja” enterrada en un substrato semiconductor. Un SOI es adecuado para el desarrollo de circuitos integrados de alta densidad.

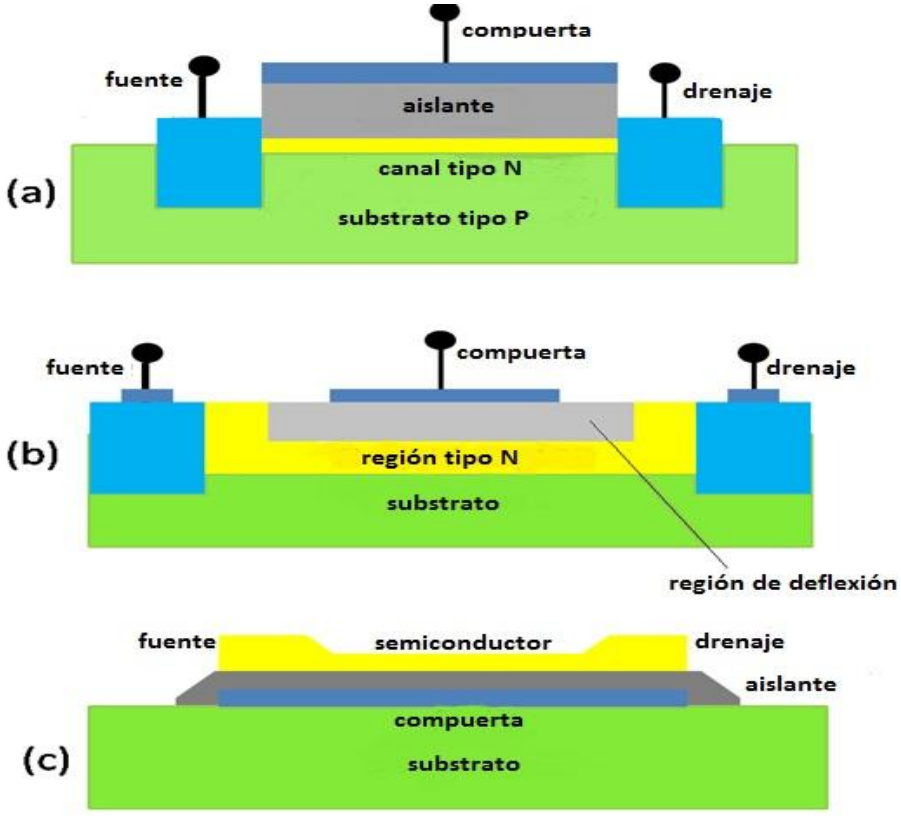


Figura 2.3. a) MISFET, b) MESFET, c) TFT; [27].

1.2.2.1 MOSFET

El FET más importante y ampliamente utilizado en la microelectrónica moderna es el MOSFET. El surtidor y el drenador están conectados por un semiconductor y la compuerta está separada del canal por una capa de óxido. En el caso de que no exista polarización (diferencia de potencial) aplicado entre la compuerta y el drenador o fuente, generalmente existe un doblamiento de bandas es inducido debido a la diferencia de energía entre la banda del material de la compuerta y el

nivel de Fermi del semiconductor, que puede producir acumulación o empobrecimiento de portadores debajo de la interfaz entre el semiconductor y el dieléctrico. Al variar el voltaje aplicado al contacto de compuerta en la magnitud y sentido adecuado, el doblamiento de bandas comienza a alinearse alcanzando la condición de banda plana. Si por ejemplo para un substrato de silicio tipo P se aplica un voltaje positivo de compuerta superior al necesario para alcanzar la condición de banda plana, los huecos se acumulan en la región cercana a la interfaz del dieléctrico-semiconductor. Si se aplica un voltaje en el sentido contrario, los portadores se alejan y se forma la región de deserción. A un mayor voltaje de polarización, el doblamiento de las bandas es mayor que el nivel de Fermi y la interfaz de dieléctrico-semiconductor queda más cercana a la banda de conducción que a la banda de valencia, por lo tanto, forma una capa de inversión, proporcionando así un canal entre drenador y surtidor por donde pasa corriente al aplicar un voltaje en drenador respecto a surtidor y el dispositivo se activa.

1.2.2.2 TFT

Tenemos también el FET de Capa Delgada o fina (TFT). El concepto de TFT fue propuesto en un principio por el Dr. Paul Weimer en 1962. La estructura es similar a la construida en MOSFET SOI con la excepción de que la película activa es una fina película depositada y que el sustrato puede ser vidrio como se muestra en la Figura 2.3c. Debido a que la capa de semiconductor se forma por deposición, el material amorfo tiene más defectos e imperfecciones que en un semiconductor cristalino, dando lugar a procesos de transporte más complicados en el TFT. En un TFT, la corriente es siempre muy limitada debido a la menor movilidad y la corriente de fuga es siempre mayor debido a los defectos. Las principales aplicaciones se encuentran donde se requieren sustratos de superficies grandes o flexibles y el uso de semiconductores convencionales no es factible. Un buen ejemplo son las pantallas de televisión con áreas grandes, donde se usa una matriz de transistores para controlar la matriz de elementos de iluminación. En tales aplicaciones el rendimiento en corrientes o velocidad, de los dispositivos, no son críticos.

Los TFTs pueden clasificarse en cuatro principales grupos de acuerdo a la posición de su compuerta y al ordenamiento de sus capas como se muestra a continuación en la Figura 2.4:

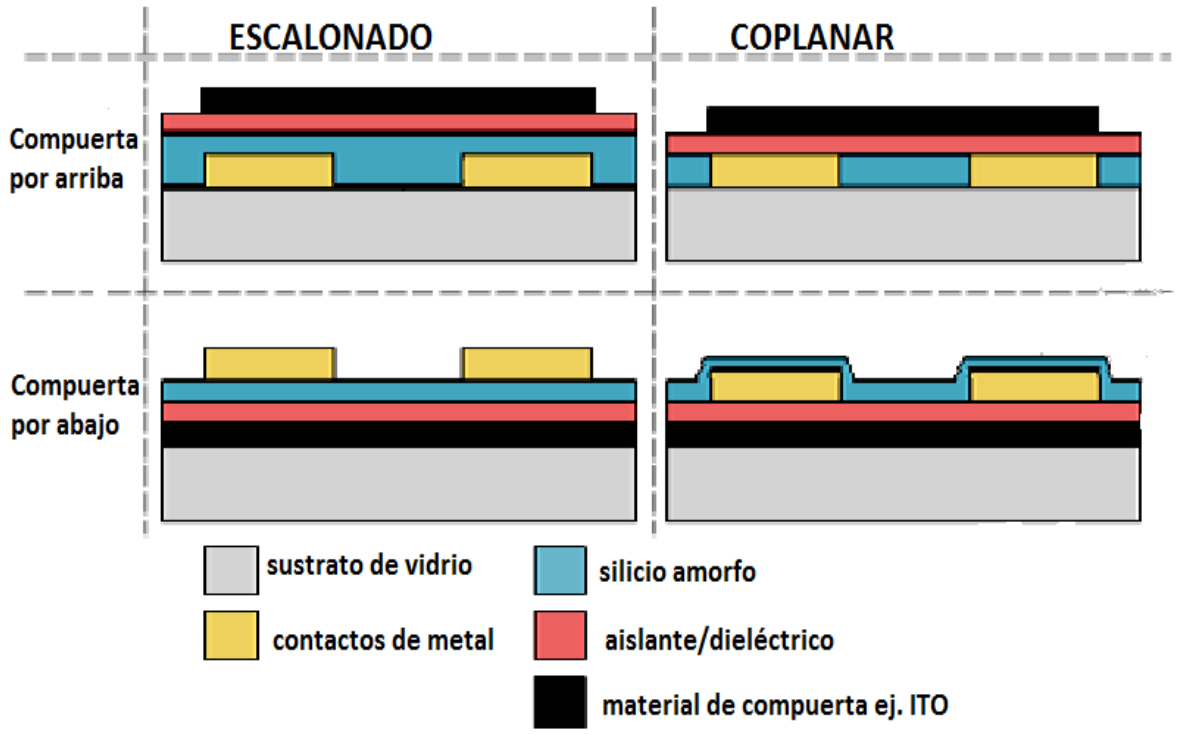
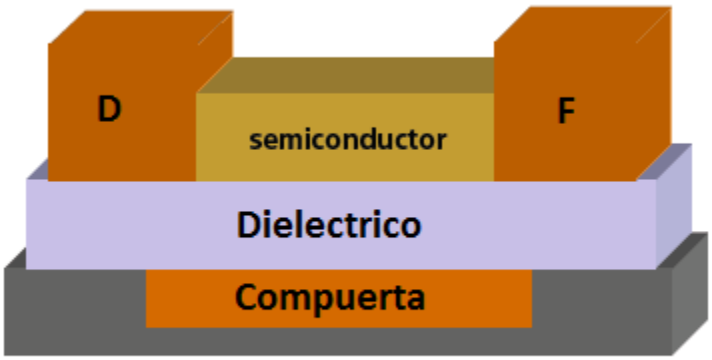


Figura 2.4. Clasificación de TFTs [28].

1.2.3 Transistores orgánicos de capa delgada OTFT

Un OTFT es un transistor de capa delgada que trabaja por el principio de efecto campo y que utiliza un semiconductor orgánico como capa activa. Los OTFTS pueden ser fabricados ya sea por una evaporación al vacío de moléculas pequeñas o por depósito de solución de polímeros (PTFT) o de moléculas pequeñas. También hay OTFTS cristalinos a veces hechos por transferencia mecánica de una capa monocristalina orgánica sobre un sustrato. El mayor impulso detrás de las investigaciones de estos dispositivos, provino de la idea de hacer circuitos integrados de bajo desempeño completamente de plástico, con los cuales poder hacer pantallas más baratas, livianas y flexibles.

Un OTFT es análogo a su contraparte inorgánica básicamente en diseño y función, constituido por las mismas terminales como en la Figura 2.5.



2.5. Estructura OTFT [28].

En un OTFT, que se ejemplifica en la Figura 2.5, la capa activa (semiconductor) se compone de una fina película de oligómeros o polímeros altamente conjugados para canal p tales como: α-hexitiofeno (α-6T) [25] pentaceno [29,30], P3HT [31], o de canal n: benzobisimidazo-benzofenan-trolina (BBL) [32] y perfluorado cobre-ftalocianina (F16- CuPc) [33].

En materiales inorgánicos bien ordenados, por ejemplo silicio cristalino (c-Si), la deslocalización de electrones sobre sitios equivalentes, conduce a un transporte a través de estados extensos en las bandas de conducción y valencia. Los portadores de carga se mueven a través de niveles de energía continuos en el sólido.

En materiales amorfos y policristalinos menos ordenados, el mecanismo de conducción propuesto es por saltos (hopping) entre estados discretos localizados. En el caso de materiales orgánicos los estados localizados se encuentran asociados a las moléculas entrelazadas o intramolecular y los saltos pueden ser intermoleculares como se describió en la sección anterior. La presencia de impurezas o defectos en la estructura también pueden dar lugar a trampas que alteran los niveles relativos de la energía y por tanto el flujo de portadores de carga.

Debido a su funcionamiento, los OTFTS han sido utilizados como dispositivos de conmutación en electrónica digital. Uno de los parámetros clave de su rendimiento es la movilidad (μ), que se describe como la facilidad con la que los portadores de carga pueden moverse dentro de la capa activa o canal bajo la influencia de un campo eléctrico y está directamente relacionado con la velocidad de conmutación del dispositivo. Este parámetro puede ser extraído de mediciones corriente-voltaje

y lo ideal sería que fuera tan grande como sea posible. Con los mejores materiales orgánicos se están logrando movilidades de 1-10 cm^2/ Vs [34,35]. La relación de Ion/Ioff, que se define como la relación de la corriente en los estados de "encendido" y "apagado", es indicativa también del rendimiento de conmutación de OTFTS. También es deseable una Ioff baja para eliminar las fugas mientras se encuentra en estado inactivo. Se han alcanzado relaciones de Ion/Ioff tan altas como 10^6 por la generación actual de OTFTS [36,37].

Otra característica de los semiconductores orgánicos es la energía de distribución de estados localizados (DOS) en el gap, los cuales son los estados de las moléculas individuales o segmentos de cadenas donde la conjugación es interrumpida por defectos estructurales o químicos. También debidos al desorden en la longitud de los enlaces de cadenas poliméricas, y puede ser directamente afectados por el tipo de alineación molecular dentro del material semiconductor. Estos introducen extensiones exponenciales de las bandas de conducción y valencia que penetran en la banda prohibida, estos son llamadas colas de las bandas y se representan en función volumétrica y de la energía.

También en las interfaces se tienen niveles de energía que pueden ser ocupados por electrones, estos se reportan por unidad de área y se representan igual que en los materiales inorgánicos (Nss).

Como un ejemplo, en la Sección de Electrónica del Estado Sólido (SEES) del Departamento de Ingeniería Eléctrica del CINVESTAV se comenzó a trabajar en dispositivos orgánicos desde 2005. Entre los principales resultados se pueden mencionar la fabricación y caracterización de diodos electroluminiscentes, celdas solares y transistores de capas finas con diferentes materiales poliméricos. Además en este último tipo de dispositivos se ha trabajado intensamente su modelación, lo que ha permitido obtener un modelo para predecir el comportamiento de éstos en circuitos más complejos.

En cuanto a la fabricación de OTFTS en la SEES, se ha desarrollado una técnica totalmente novedosa, tanto en la combinación de las capas poliméricas que utiliza, como en el proceso de fabricación, obteniéndose con esto, por primera vez una estructura con PMMA depositado sobre P3HT (podemos ver una estructura de un transistor orgánico en la Figura 2.6) [38], que logra la compatibilidad entre disolventes, es decir, la capa de PMMA se disuelve en anisol que no diluye el P3HT [38]. Este era uno de los problemas más difíciles de resolver en la fabricación de OTFTS, por lo que obtener esta selectividad en el grabado de PMMA fue uno de los principales logros. También se seleccionó este material como dieléctrico, debido a que permite una mayor estabilidad en las características de los dispositivos comparado con resultados anteriores. El PMMA

también permitió utilizar procesos de fotolitografía con solo una modificación con respecto al utilizado en la fabricación de CI de silicio. Lo cual no se había obtenido anteriormente y lo que permitiría un gran paso hacia la integración de este tipo de dispositivos. La combinación de estos resultados permitió obtener OTFTS con movilidad en el orden de 0.04 cm^2/Vs [37-40], los cuales fueron realizados con polímeros semiconductores como es el PCDTBT y el F8T2 [41-42].

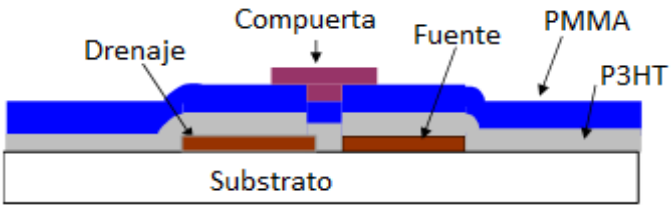


Figura 2.6. Estructura de un transistor orgánico [38].

CAPÍTULO 2

MODELO ANALÍTICO DEL OTFT

2.1 Modelo de Corriente-Voltaje

De acuerdo con [43] muchos de los modelos de corriente-voltaje desarrollados para OTFTS pueden reproducir con un cierto grado de precisión el comportamiento de la corriente del drenador para diferentes voltajes de compuerta en al menos una región de operación. De entre las principales diferencias que presentan estos modelos se pueden mencionar las siguientes:

- Las expresiones del modelo de movilidad de los portadores. Algunos se basan en las expresiones clásicas para el caso de movilidad en MOSFETs de Silicio, mientras que muy pocos se basan en las propiedades del material orgánico.
- El número de parámetros del modelo y de las regiones de operación que es capaz de modelar.
- Los procedimientos de extracción de parámetros.

La razón principal de que se ocupe el modelo clásico del MOSFET de silicio para modelar las características del OTFT es debida a que en los primeros trabajos mostrados por [44] sobre estos dispositivos, se observó un comportamiento muy cercano al MOSFET de silicio. Algunos trabajos se han enfocado en solamente hacer pequeñas correcciones al modelo del MOSFET de silicio para tratar de acercarlo lo mejor posible al caso orgánico. Sin embargo, se ha demostrado que estas aproximaciones no son correctas ya que los mecanismos de transporte de carga y de movilidad en los materiales amorfos son totalmente diferentes que al caso cristalino [45].

Los OTFTS pueden presentar una deformación en el origen de las características de salida debido a los contactos no-óhmicos que generalmente se tienen en las terminales de fuente y drenador. Además, estos tipos de dispositivos pueden presentar corrientes de fuga a través del dieléctrico de compuerta y/o por efectos de polarización del transistor. Así también, la naturaleza de alto desorden que presentan los materiales amorfos, principalmente los polímeros, dan lugar a preguntarse si los métodos de extracción de parámetros por medio de las

características corriente-voltaje medidos son correctos o no, ya que la movilidad, resistencia de contacto y el voltaje de umbral presentan una alta dependencia de los voltajes aplicados [45]. Debido a lo anterior, un mayor trabajo en el diseño de metodologías para la extracción de parámetros de los OTFTS debe de llevarse a cabo.

Sobre los modelos de movilidad y en específico del mecanismo físico que se presenta en los materiales amorfos, se puede comentar que el más importante es el de "hopping" como se mencionó anteriormente en la sección 2.2.3. Mientras que, la distribución de estados localizados de tipo donador o aceptor es representado por distribuciones Gaussianas o por una distribución exponencial con una temperatura característica, T_0, de la forma propuesta por [46]:

$$g(E) = g_{do} \cdot exp\left({}^{-E_r}/_{kT_0}\right) \tag{3.1}$$

Donde g_{do} es el máximo de la distribución de la densidad de estados, E_r es la energía relativa de una de las bandas de estados de trampas profundos o de cola (ver Figura 3.1), *k* es la constante de Boltzman.

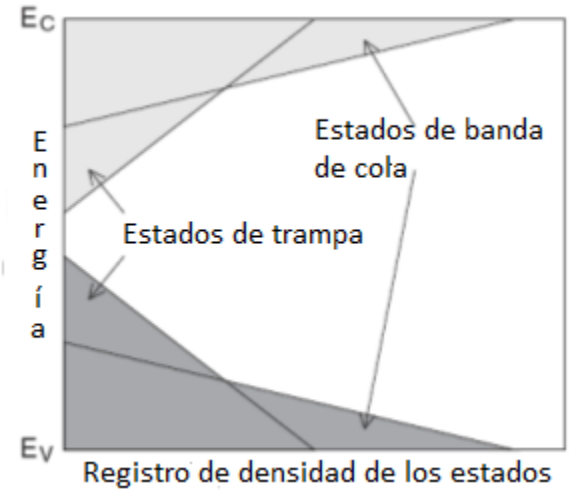


Figura 3.1 Diagrama de bandas de los semiconductores amorfos [47].

Tomando en consideración lo anteriormente mencionado, es posible realizar la selección del modelo analítico de entre todos los disponibles [47-56] que mayor precisión presenten en comparación con los datos medidos, además de que

considere los mecanismos que se presentan en los OTFTS. Por lo tanto, el presente trabajo se enfocará en el modelo UMEM (Unified Model and Extraction Method) desarrollado por [57] para el caso de TFTs de a-Si: H, después ampliado a TFTs de Polisilicio [58], actualizado para considerar materiales OTFTS en [59] y por ultimo ampliado para tomar en cuenta TFTs de óxidos metálicos [60-61].

El modelo UMEM toma en cuenta las regiones de operación antes y después del voltaje de umbral con transiciones continuas entre los régimen lineal y de saturación.

2.1.1 Descripción del Modelo UMEM.

El modelo de corriente-voltaje propuesto por UMEM se basa en la solución de la ecuación de Poisson (Potencial) en donde se asume que la carga localizada es mucho mayor que la concentración de carga libre en todas las regiones de operación y que la distribución de estados localizados es exponencial (ecuación 3.1). Solo bajo tales consideraciones es posible obtener una ecuación analítica para el campo eléctrico así como para la carga inducida en el canal. Gracias a la integral de este último a lo largo del canal, es posible calcular la corriente del drenador. Como resultado de tal operación se obtiene la siguiente expresión [38]:

$$I_{DS} = P(T,T_0)\frac{W}{L}C_i\frac{T}{2T_0}\left[(V_G - V_T)^{2T_0/T} - (V_{GS} - V_T - V_{DS})^{2T_0/T}\right] \quad (3.2)$$

Donde:

$$P(T,T_0) = P'(T,T_0)\frac{C_i^{\gamma}}{\varepsilon_s^{\gamma}}(V_{GS} - V_T)^{\gamma} \quad (3.3)$$

$$P'(T,T_0) = \frac{qkTN_V \cdot exp\left(-\frac{E_r}{kT}\right)}{\left[\pi qkTg_{d0} \cdot exp\left(-\frac{E_r}{kT}\right)\right]^{\frac{T_0}{T}}}\left[\frac{sen\left(\pi T/T_0\right)}{2kT_0}\right]^{\frac{T_0}{T}} \quad (3.4)$$

Y

$$\gamma = \frac{2T_0}{T} - 2 \quad (3.5)$$

Donde *W* es el ancho del canal, *L* es el largo del canal, C_i es la capacitancia del aislante por unidad de área, *T* es la temperatura absoluta, V_T es el voltaje de umbral, V_{GS} es la diferencia de potencial compuerta-fuente, V_{DS} es la diferencia de potencial drenador-fuente, ε_S es la permitividad del silicio y γ es el parámetro que describe el desorden que se tiene en el material orgánico semiconductor.

Por otra parte, la movilidad afectada por el efecto de campo en los OTFTS puede ser descrita de acuerdo con [62] como:

$$\mu_{FET} = \frac{\mu_0}{V_{aa}^{\gamma}}(V_{GS} - V_T)^{\gamma} \tag{3.6}$$

Donde V_{aa} es un parámetro adimensional y μ_0 puede ser tomado como un valor constante con valor de 1 $cm^2/V*s$. Vale la pena mencionar que la movilidad depende de V_{GS} y del desorden dado por γ, por lo que si el valor de este último parámetro es alto, esto significará que se tiene un material muy desordenado.

Entonces, para poder obtener una expresión completa de movilidad para la corriente del drenador, es posible conjuntar la ecuación 3.6 con 3.4, lo cual da como resultado la siguiente expresión:

$$\mu_{FET}(V_{GS}, T, T_0) = \mu_0 P(T, T_0) = \frac{\mu_0 q k_b T N_V \cdot exp\left(-\frac{E_r}{k_b T}\right)}{\left[\pi q k_b T g_{d0} \cdot exp\left(-\frac{E_r}{k_b T}\right)\right]^{\frac{T_0}{T}}} \left[\frac{sen\left(\pi T/T_0\right)}{2k_b T_0}\right]^{\frac{T_0}{T}} \frac{C_i^{\gamma}}{\varepsilon_s^{\gamma}}(V_{GS} - V_T)^{\gamma} \tag{3.7}$$

De acuerdo con la expresión anterior, el mecanismo de hopping que domina la movilidad en este tipo de dispositivos orgánicos está dado por los primeros dos términos, los cuales provienen de la expresión $P'(T,T_0)$.

Como se puede observar, la gran ventaja que presenta el modelo UMEM en comparación con los otros modelos, es que presenta ecuaciones analíticas sin parámetros de ajuste y que todos los parámetros que presenta tienen significado físico. Así también, que el modelo de movilidad depende de la densidad de estados localizados, g_{d0}, y del desorden del material dado por γ. Sin embargo, la expresión general de corriente del drenador (3.2) puede ser representado de forma más sencilla utilizando las aproximaciones propuestas en [38], lo cual permite incorporar los efectos de las resistencias de contacto, así como facilitar la metodología de extracción de los parámetros, pero con la contraparte de la aparición de parámetros de ajuste. Como resultado la expresión aproximada de corriente del drenador está dada por:

$$I_{DS} = \frac{1}{2} I_{DSsub}[1 - tanh(\delta)] + \frac{1}{2} I_{DSa}[1 + tanh(\delta)] \tag{3.8}$$

Donde $\delta = (V_{GT}-K)\,Q_1$, $V_{GT}=V_{GS} - V_T$, K y Q_1 son parámetros de ajuste y I_{DSsub} es la corriente del drenador en subumbral, I_{DSa} es la corriente del drenador después del voltaje de umbral, dada por:

$$I_{DSa}(V_{GS}, V_{DS}) = \frac{W}{L} C_i \mu_{FET0} \frac{V_{GT}^{1+\gamma} V_{DS}(1+\lambda|V_{DS}|)}{\left[1+R_T \frac{W}{L} C_i \mu_{FET} V_{GT}\right]\left[1+\left(\frac{V_{DS}}{V_{DSAT}}\right)^m\right]^{\frac{1}{m}}} \tag{3.9}$$

$$I_{DSsub} = I_{DSa}(V_{sub}, V_{DS}) \cdot exp\left(\frac{2.3\,V_{GT}}{S}\right) + I_0 \approx \frac{W}{L} C_i \mu_0 \frac{V_{GT}^{1+\gamma} V_{DSl}}{V_{aa}^{\gamma}} \quad (3.10)$$

Donde V_{sub} es un voltaje $V_{GS} < V_T$, V_{DSl} es un voltaje del drenador lo suficientemente pequeño que asegure estar en régimen lineal, R_T es la resistencia total dada por la suma de la resistencia de contacto y la resistencia del canal. Mientras que la movilidad a bajos campos, $\mu_{FET0} = \mu_0 / V_{aa}^{\gamma}$, λ representa la modulación del canal, m es un parámetro de ajuste de la curvatura de la característica de salida del transistor, $V_{DSAT} = \alpha V_{GT}$ es el voltaje de saturación del drenador, S es el parámetro medido del swing de subumbral e I_o corresponde al parámetro medido de la corriente mínima del transistor también conocido como I_{off}.

Para ejemplificar los resultados que pueden ser obtenidos al utilizar el modelo UMEM, se han tomado los resultados del trabajo realizado por Estrada, et. al [38]. En este trabajo se consideraron OTFTS realizados con diferentes materiales orgánicos, sin embargo solo presentaremos los resultados correspondientes a dos polímeros para ilustrar su efecto en la movilidad, en este caso el P3HT, P3HDT y el P3DDT (Figura 3.2). Mientras que para el caso de la característica transferencial (Figura 3.3) y de salida (Figura 3.4) del OTFT se tomó solamente el OTFT realizado con P3HT. La tabla 3.1 resume los materiales y parámetros obtenidos para estos OTFTS.

	T1	T2	T3
Configuración	Compuerta arriba/ contactos S y D abajo.	Compuerta abajo/ contactos S y D arriba	Compuerta abajo/ contactos S y D arriba
Semiconductor/espesor	P3HT/80nm	P3HDT/38nm	P3DDT/38nm
Aislante/espesor	PMMA/320nm	SiO_2/200nm	SiO_2/200nm
W/L (μm)	150/50	15000/10	15000/10
γ	1	0.3	1.8
T_0 (K)	445	353	567
g_{do} (cm^{-3}/eV)	$1.25x10^{-23}$	$1.25x10^{-23}$	$1.25x10^{-23}$
μ_{FET0} (cm^2/Vs); V_{GT}=-1V	$5x10^{-5}$	$3x10^{-4}$	$2x10^{-7}$
μ_{FET} (cm^2/Vs); V_{GT}=-30V	$1.5x10^{-3}$	$7.5x10^{-4}$	$8x10^{-5}$
RT (Ω)	$\approx 10^7$	$5x10^4$	$\approx 10^6$
α	0.43	0.64	0.5
M	1.7	1.7	1.5
λ (V^{-1})	$-1.4x10^{-4}$	$-3.5x10^{-3}$	$-2x10^{-3}$

Tabla 3.1. Configuración, materiales y parámetros extraídos en 3 OTFTS [38].

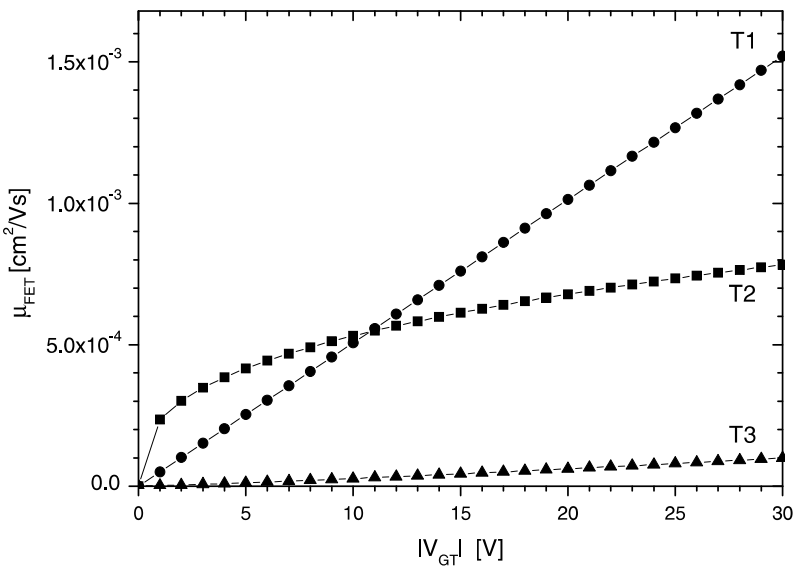


Figura 3.2. Comportamiento de la movilidad con respecto al potencial aplicado para 3 diferentes OTFTS tomado de [38].

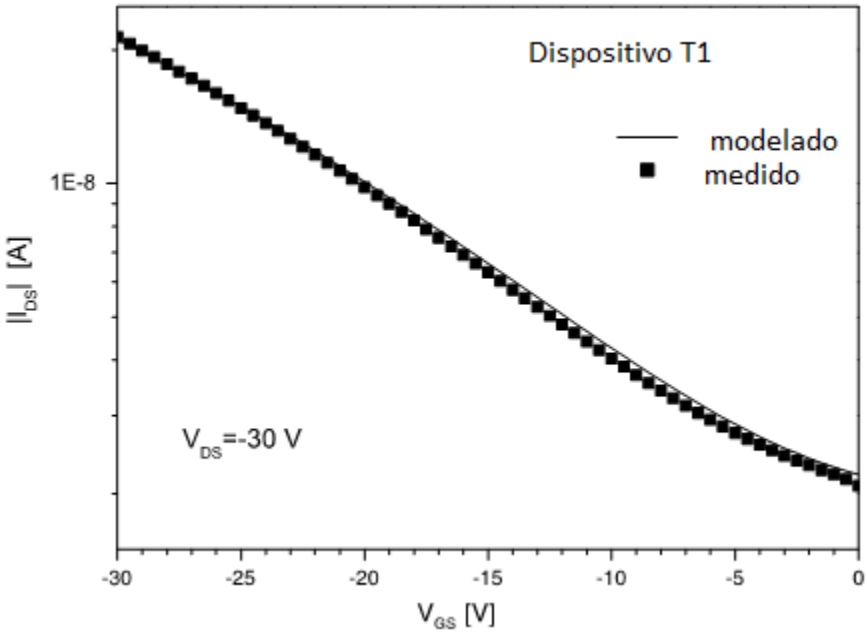


Figura 3.3. Comparación de la característica transferencial modelada en el transistor T1 con respecto a datos experimentales tomada de [38].

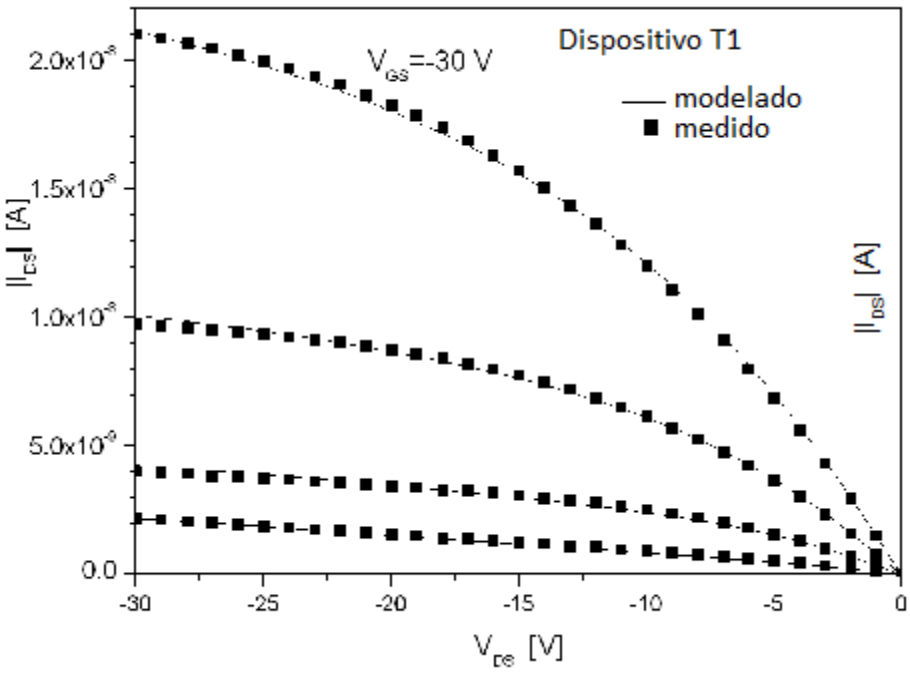


Figura 3.4. Comparación de la característica de salida modelada en el transistor T1 con respecto a datos experimentales tomada de [38].

Como se observa en las Figura 3.2, 3.3 y 3.4, se hace referencia a las respuestas obtenidas por medio de la simulación utilizando el modelo UMEM y los resultados experimentales, donde el margen de diferencia es mínimo en cuanto a las curvas de movilidad y la curva de salida para los 3 diferentes transistores, sólo para el caso de la curva de transferencia se observa una pequeña diferencia cuando vgs es aproximadamente de -15 V, pero este valor es muy mínimo respecto al resultado experimental.

2.2 Modelo de Capacitancia-Voltaje

Debido a que un modelo analítico de cualquier dispositivo semiconductor no solo necesita representar el comportamiento en condición estática –como lo visto en la sección anterior-, sino también su comportamiento dinámico (ej. cuando opera en frecuencia), es entonces necesario representar los elementos que intervienen en estas condiciones de operación, en este caso los más importantes son las capacitancias inherentes al tipo de estructura.

En específico, en esta subsección se mostrará el comportamiento de las capacitancias del OTFT en las tres condiciones de operación: 1) acumulación, 2) empobrecimiento parcial, 3) empobrecimiento total.

2.2.1 Modelo de Carga

Para desarrollar el modelo de carga es necesario considerar a la carga móvil en función del voltaje de compuerta como está dado por [62]:

$$Q_{mob} = \frac{C_i(V_{GT}-V)^{1+\gamma}}{V_{aa}^{\gamma}} \tag{3.11}$$

Así también, es necesario definir la carga que está presente a lo largo de todo el canal del transistor, lo cual está dado por:

$$Q_{CH} = W\int_0^L C_i(V_{GT}-V)dx \tag{3.12}$$

Como es posible observar, si se realiza la integración con respecto a dx, los demás términos dentro de la integral serían constantes y no se tendría una dependencia con respecto a los voltajes aplicados en las terminales. Por lo tanto, es necesario realizar un cambio de variable. Al respecto, se puede considerar que es posible realizar el cálculo de la corriente del drenador de la siguiente forma:

$$I_{DS} = W\mu_0 Q_{mob}\frac{dV}{dx} \tag{3.13}$$

Despejando dx de 3.13 para posteriormente reemplazarlo en 3.12, tenemos que:

$$Q_{CH} = W\int_0^{V_{DS}} C_i(V_{GT}-V)\frac{W\mu_0 Q_{mob}}{I_{DS}}dV \tag{3.14}$$

Resolviendo la integral realizando algunas consideraciones dadas en [61] para incorporar el efecto de la modulación del canal, se obtiene que:

$$Q_{CH} = WLC_i\frac{2+\gamma}{3+\gamma}\frac{\left[(V_{GT}-V_{DSe})^{3+\gamma}-V_{GT}^{3+\gamma}\right]}{\left[(V_{GT}-V_{DSe})^{2+\gamma}-V_{GT}^{2+\gamma}\right]} \tag{3.15}$$

Donde $V_{DSe} = V_{DS}/[1+(V_{DS}/\beta\, V_{GT})^{m*}]^{m*^{-1}}$, β y m* son parámetros de ajuste, los cuales son debidos al efecto de la modulación del canal.

Gracias a la anterior expresión es posible calcular la carga total en la terminal de compuerta, Q_G, ya que $Q_G = -Q_{CH}$. Por medio del uso del esquema de partición de carga dada de Ward-Dutton, es posible encontrar las expresiones de carga en las terminales fuente y drenador de la forma [60]:

$$Q_D = \pm\frac{LWC_i(2+\gamma)}{\left[(V_{GT}-V_{DSe})^{2+\gamma}-V_{GT}^{2+\gamma}\right]^2}\left[\frac{B}{5+2\gamma}-\frac{A_3V_{GT}^{2+\gamma}}{3+\gamma}\right] \tag{3.16}$$

$$Q_S = \mp(Q_{CH}-Q_D) \tag{3.17}$$

Donde $B = (V_{GT} - V_{DSe})^{5+2\gamma} - V_{GT}^{5+2\gamma}$ y $A_3 = (V_{GT} - V_{DSe})^{3+\gamma} - V_{GT}^{3+\gamma}$.

2.2.2 Modelo de Capacitancia

Como se mencionó al principio de esta sección, es necesario definir las capacitancias del OTFT para poder representar su comportamiento dinámico, por cual ayudándonos de las cargas encontradas en las tres terminales Q_G, Q_D y Q_S, podemos ahora definir las capacitancias no recíprocas como:

$$C_{xy} = -\frac{\partial Q_x}{\partial V_y} \quad x \neq y \tag{3.18}$$

$$C_{xy} = \frac{\partial Q_x}{\partial V_y} \quad x = y \tag{3.19}$$

Donde *x* e *y* hacen referencia a las terminales del OTFT. Por lo tanto, podemos encontrar la capacitancia entre dos terminales debido a la variación de carga en la terminal *x*, Q_x, la cual es provocada por la variación de voltaje en la terminal *y*, Q_y.

Es posible entonces, encontrar las nueve capacitancias intrínsecas de acuerdo a la siguiente relación, la cual se deriva de las leyes de conservación de carga:

$$C_{GG} = C_{GS} + C_{GD} = C_{SG} + C_{DG} \tag{3.20}$$

$$C_{DD} = C_{DS} + C_{DG} = C_{SD} + C_{GD} \tag{3.21}$$

$$C_{SS} = C_{SG} + C_{SD} = C_{GS} + C_{DS} \tag{3.22}$$

Una vez que se cuenta con las capacitancias intrínsecas, es necesario también considerar las capacitancias extrínsecas, la cual esta principalmente dada por el traslapamiento de la terminal de compuerta con las terminales de fuente y drenador, L_{OVR}. Por lo tanto, esta capacitancia, C_{OVR} está dada por $C_{OVR}=W*L_{OVR}*C_i$.

Es importante mencionar que a partir de 3.20 a 3.22, es posible determinar las capacitancias para las diferentes condiciones de operación, ya sea acumulación o empobrecimiento. En este caso solamente se ha considerado acumulación y de acuerdo con [63], solamente se añadirán algunos términos en Q_D y Q_S para poder representar la capacitancia en empobrecimiento.

De igual forma que para el caso de corriente-voltaje, para ejemplificar los resultados que pueden ser obtenidos al utilizar el modelo UMEM, se han tomado los resultados del trabajo realizado por [62]. En este trabajo se considera el dispositivo T1 dado en la tabla 3.1. La Figura 3.5 muestra la buena aproximación obtenida entre la característica C_{GG} medida y la modelada para diferentes frecuencias de operación.

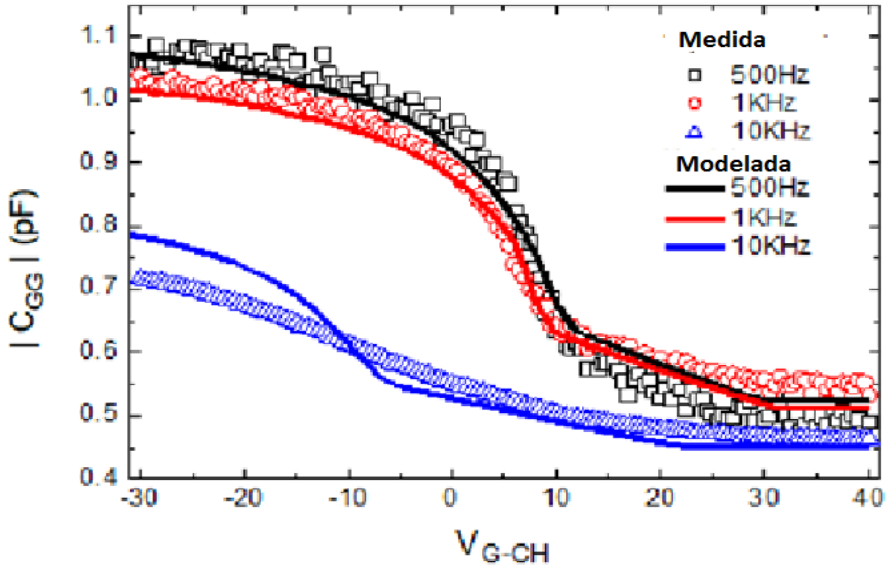


Figura 3.5. Comparación de la capacitancia CGG (capacitancia compuerta a compuerta) modelada con respecto a la medida a diferentes frecuencias de operación tomada de [63].

El modelo UMEM ha sido descrito mediante ecuaciones de corriente-voltaje, carga y capacitancia, por lo que de acuerdo a lo observado, en la comparación de las curvas características del OTFT podemos decir que muestra en la mayor parte el mismo resultado o muy semejante, lo que nos resulta conveniente hacer uso de este modelo de transistores orgánicos, además de ser único que se ha reportado en la literatura.

CAPÍTULO 3

IMPLEMENTACIÓN EN NGSPICE

3.1 Software comercial y libre

Para llevar a cabo la simulación de OTFTS es necesario hacer la implementación del modelo analítico UMEM en un software de simulación de circuitos, el cual es la herramienta principal del diseñador de circuitos. Al respecto, se hizo referencia de la implementación del modelo UMEM en el software comercial SmartSpice de Silvaco ®, por lo cual en el presente trabajo se mostrara a continuación la implementación de este modelo en el software libre NGspice.

3.1.1 SmartSpice de Silvaco ®

SmartSpice [64] cuenta con herramientas que manejan un alto rendimiento y exactitud en las cuales se requieren diseños complejos, análisis de señales analógicas y mixtas con alta precisión, así como un compilador potente para implementar fácilmente modelos analíticos desarrollados en Verilog-A. Además de que cuenta con un extenso número de modelos eléctricos y analíticos, incluyendo modelos analíticos de diferentes tipos de TFTs (silicio amorfo, polisilicio, orgánico universal y metal-óxido) de las cuales soporta análisis en DC, AC, transitorio. También incluye fuentes que van desde fuentes de voltaje hasta PWM y moduladores FM/ AM, y ciertamente también contiene elementos pasivos. La gran mayoría de los modelos en SmartSpice están escritos en Verilog-A que es el lenguaje de programación usada para sistemas analógicos.

Características principales:

- Es compatible con CADENCE, HSPICE y SPECTRE en cuanto a netlist, modelos, análisis y resultados.
- Una gran colección de modelos SPICE con tecnologías tradicionales (Bipolar, CMOS) y emergentes (TFT, UOTFT, SOI, HBT, FRAM, FinFET etc).
- Es soportado por Windows y Linux.
- Exactitud, velocidad, costo y fácil adopción.

3.1.2 NGspice

Es un simulador de circuitos de señales mixtas (creado por The University of California at Berkeley en USA), de código abierto basado en tres paquetes (SPICE3f5, Cider1b1 y XSPICE). Ha tenido muchas modificaciones por lo que hasta el momento se considera un simulador estable. NGspice es la suma del trabajo de mucha gente que contribuye al proyecto con su experiencia, por lo que queda abierta la opción de hacer contribuciones a este proyecto, lo cual ofrece la oportunidad de poder hacer uso de este software para llevar a cabo la implementación de nuevos modelos analíticos sin costo alguno.

Con NGspice podemos compilar en código C así como realizar la compilación de modelos compactos que están definidos en lenguaje Verilog-AMS. Además, contiene los siguientes modelos en Verilog-A para MOSFETS y transistores bipolares: ekv (nivel 44), hicum0 (nivel 7) y hicum2 (nivel 8), mextram (nivel 6) y psp102 (nivel 45).

Por otra parte, es importante mencionar que NGspice permite realizar simulaciones en AC, DC y transitoria solamente proporcionando el netlist, por lo cual no hay necesidad de utilizar programas esquemáticos para poder realizar el diseño y/o simulación del circuito. Como se puede apreciar, éste programa no es muy amigable, ya que incluso es estrictamente necesario ejecutar las simulaciones desde una terminal de Linux.

Cabe mencionar que este proyecto no está enfocado en desarrollar una interfaz gráfica aunque en la actualidad es posible contar con esta opción por medio de herramientas básicas de software desarrolladas por terceros. Por lo tanto, podemos encontrar la contribución de otros programas libres y comerciales con herramientas para simulación por medio de esquemáticos del circuito, así como crear el layout e inclusive crear simulaciones en línea con NGspice. Hasta la fecha también se tiene una herramienta para realizar un análisis propiedades electromagnéticas de interconexiones eléctricas [65].

3.2 Instalación de NGspice

A continuación se detallará el proceso de instalación del software libre NGspice, así como su habilitación para poder compilar el código Verilog-A. Cabe mencionar que este proceso es sólo para sistemas operativos de distribución Linux.

3.2.1 Requerimientos previos

Para poder instalar el programa NGspice, es necesario decidir con que sistema operativo se va a trabajar o cual es el más conveniente. Para nuestro caso fue más conveniente utilizar el sistema operativo Linux, en específico la distribución CentOs. Esta distribución fue elegida, ya que es la que utiliza actualmente el servidor de simulaciones de circuitos del Posgrado en Dispositivos Semiconductores (stark), así también porque ofrece la posibilidad de realizar la compilación por medio del programa ADMS, lo cual no es posible hasta el momento en Windows.

Una vez seleccionado el sistema operativo, es necesario instalar algunos programas con los cuales NGspice tiene dependencia siendo estos: bison, flex, cabeceras de X11 y ADMS.

De estos programas es posible mencionar que Bison es un programa generador de analizadores sintácticos (parte de un compilador que convierte las entradas en otras estructuras como árboles) de propósito general que pertenece al programa de GNU, por lo que GNU es un sistema operativo de software libre de tipo Unix (donde se puede estudiar, modificar, copiar, ejecutar, o mejorar si es posible). Para instalarlo se recomienda ejecutar los siguientes comandos:

```
[simon@stark] $ yum search bison
```

Este comando busca la existencia de bison en caso de que no se encuentre, es necesario ejecutar el siguiente comando:

```
[simon@stark] $ sudo yum  install bison
```

Gracias a éste, se descargará e instalará bison, para algunos casos es necesario ser superusuario de no ser así se le negara los permisos de instalación y modificación de aplicaciones en la cuenta de trabajo.

Para el caso de Flex se hace un procedimiento similar a Bison, el cual es una aplicación gratuita y es un generador de analizador léxico rápido.

Por otra parte, el programa X11 permite tener acceso a múltiples ventanas simultáneamente de modo grafico que son independientes del sistema operativo. A su vez, hace referencia al proyecto X de la versión 11 que es la actualmente usada.

Es necesario hacer la instalación de ADMS, debido a que es quien se encarga de convertir los modelos escritos en Verilog-A en código C utilizado en simuladores

SPICE incluyendo Spectre Circuit Simulator, NGspice y HSPICE, también puede consultar.

```
[simon@stark] $ sudo yum  install adms
```

En caso de ser necesario instalar X11 y flex esto se realizará como se hizo en los pasos anteriores:

```
[simon@stark] $ sudo yum  install flex

[simon@stark] $ sudo yum  install X11
```

Cuando se hayan instalado todos los paquetes mencionados anteriormente podremos proseguir a la instalación de NGspice.

3.2.2 Instalación

Ya que hemos instalado los paquetes anteriores iremos a instalar NGspice siguiendo los siguientes pasos (pueden variar dependiendo el sistema operativo).

1. Descargar NGspice de la página: *http://sourceforge.net/projects/NGspice/.* Se descargará como *ngspice-26.tar.gz,* llamado tar ball.

2. Guardar y descomprimir el tar ball desde un directorio local en nuestra máquina como sigue:

```
[simon@stark] $ tar –zxvf NGspice-26.tar.gz
```

3. Después se ejecutan las siguientes instrucciones para realizar la instalación, una vez haciendo esto el software libre quedará en */usr/local/bin/ngspice-26*

```
[simon@stark] $ mkdir release

[simon@stark] $ cd release

[simon@stark] $ ../configure --with-x --with-readline=yes --
disable-debug

[simon@stark] $ make

[simon@stark] $ sudo make install
```

Vale la pena mencionar que durante todo el proceso no se debió de haber generado ningún mensaje de error. En caso contrario, el proceso de instalación no puede ser completado o el programa no se ejecutará de forma adecuada, esto se debe a que lo más seguro, el sistema operativo a que se esté utilizando no sea compatible o bien se deba hacer uso de otras instrucciones para instalar un tar ball.

3.3 Implementación del modelo analítico de OTFTS

El modelo analítico UMEM de OTFTS se ha implementado directamente por el desarrollador del software comercial (SmartSpice en el nivel 37) por lo que se usa como referencia éste, así como la literatura disponible para poder llevar a cabo la validación de la nueva implementación en NGspice. A continuación se describe en forma general como se llevó a cabo las simulaciones circuitales partiendo del modelo analítico que ya fue implementado por [66], Figura 4.1.

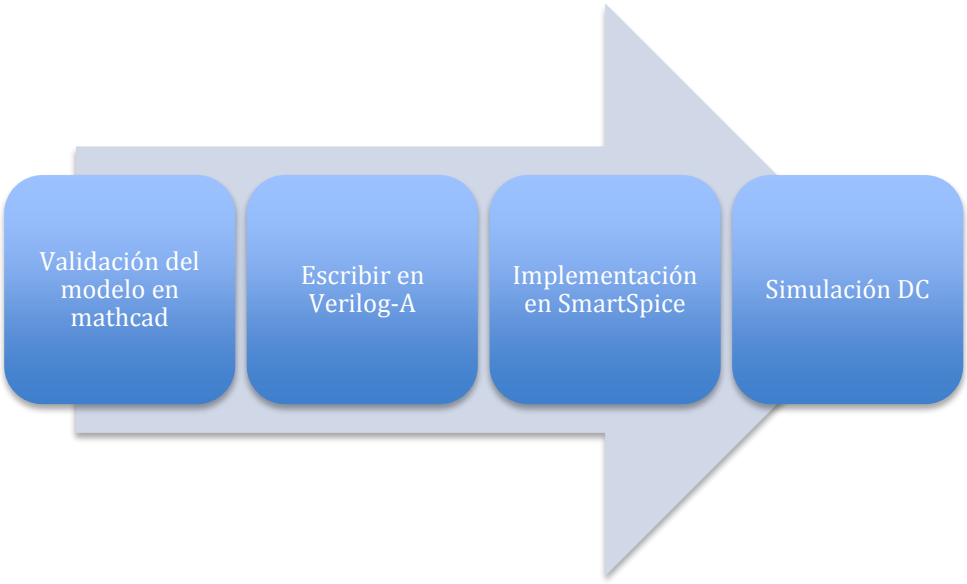


Figura 4.1. Esquema de validación primero en mathcad por [66].

En un principio se determinó que la implementación se llevaría a cabo por medio de la creación de un nuevo modelo dentro de la biblioteca de modelos del NGspice. Sin embargo, debido a que es necesario contar con conocimientos avanzados de programación orientada a objetos, este proceso no fue posible llevarlo a cabo satisfactoriamente. Por lo tanto, se determinó implementar el modelo de OTFTS directamente sobre algún otro modelo en Verilog-A ya previamente incluido dentro de la biblioteca de NGspice, es decir, se tendría que

editar el modelo preexistente de algún dispositivo, reemplazándolo entonces por el modelo analítico UMEM disponible.

Para llevar a cabo esta labor es necesario conocer los modelos que están disponibles en NGspice por lo que retomando lo dicho anteriormente en la sección 4.1.2 tenemos a: EKV, HICUM0, HICUM2 y PSP. Tomando en cuenta esto se escogió, el modelo HICUM0, el cual tiene asignado el nivel 7 (case 7) y el cual está dedicado a simular transistores bjt, es muy importante conocer el nivel en el cual trabajaremos para que se pueda proseguir con la compilación, ya que esto determina como le llamaremos al módulo del OTFT (en este caso debe ser igual al de hicum0), para conocer el nivel se debe acceder al archivo inpdomod.c (Figura 4.1), dentro del directorio:

```
[simon@stark] parser $
/../ngspice26/src/spicelib/parser/inpdomod.c
```

Dentro de este directorio, es donde encontramos el nivel que le corresponde al código de Verilog-A para hicum0. Esto es importante para poder llevar a cabo la ejecución del modelo, y así finalmente obtener una simulación del dispositivo. En las siguientes líneas veremos una sección de código que encontraremos en el archivo de inpdomod.c.

```
break;
case 7:
     type = INPtypelook("hicum0");
if(type < 0) {
     err= INPmkTemp(
          "Device type HICUM0 not available in this
binary\n");
}
break;
```

Este nivel de hicum0 debe ser el mismo para la implementación del modelo de OTFT, ya que de otra forma se generarán errores debido a que no corresponderá el nivel asignado para que el modelo pueda ser ejecutado.

3.3.1 Metodología de compilación y adaptación del modelo

Teniendo en mente el directorio de instalación de NGspice, y utilizando un editor de textos mediante terminal (ej. puede ser *emacs* o *gedit*) se procedió a realizar los siguientes pasos:

1. Como se trabajara en HICUM0 se debe acceder al directorio de Hicum0 localizado en NGspice, desde la terminal sería:

```
[simon@stark] $ cd
/usr/local/bin/ngspice26/src/spicelib/devices/adms/hicum0/
```

2. Dentro de hicum0 se eliminan los archivos generados durante la implementación del mismo, debido a que necesitamos el espacio para el nuevo modelo, por lo cual se ejecutan los comandos:

```
[simon@stark] $ make clean

[simon@stark] $ rm Makefile Makefile.in Makefile.am
```

3. Verificando que se hayan eliminado todos los archivos (sólo queda la carpeta admst), preferentemente se puede realizar en otra ventana de terminal, en el cual este en el mismo directorio de hicum0, se ejecuta el comando:

```
[simon@stark] $ ls
```

4. En caso de que no permita eliminar u hacer modificaciones, se verifica si se cuenta con los permisos, por medio de:

```
[simon@stark] $ su

Contraseña:    ********
```

5. Ahora se prosigue a la implementación del modelo, recordando que está escrito en lenguaje Verilog-A, se debe entonces localizar el archivo .va de hicum0, que se encuentra en:

```
[simon@stark] $ cd usr/local/bin/ngspice-
26/src/spicelib/devices/adms/hicum0/admst/
```

Utilizando el editor de texto con:

```
[simon@stark] $ gedit &
```

Se abre el archivo *hicum0.va* para entonces proceder a eliminar el modelo hicum0 original (tenerlo a la mano en otra parte visualizado para el futuro) y se procede a sustituirlo con el modelo de OTFTS (ver anexo A). Después de esto se deben de guardar los cambios pero no se debe cambiar el nombre de *hicum0.va*.

6. Se verifica si el modelo puede compilarse tal y como es originalmente con el siguiente comando para crear el archivo *Makefile.am*:

```
[simon@stark] $ admsXml admsva/hic0_full.va -Iadmsva -e
../admst/ngspiceMakefile.am.xml
```

Una vez ejecutado se observa que se han tenido errores, referente a que no coincide con *hicum0.va* original, notemos que el modelo de *hicum0.va* (ver anexo B) tiene lo siguiente:

- Modulo llamado *hic0_full (d, g, s, e, tnode)*
- Entradas y salidas declaradas como: *inout(d, g, s, e, tnode)*
- Nodos eléctricos: *electrical (d, g, s, e, tnode)*

Considerando lo anterior, debemos adaptar el modelo de OTFTS con el módulo y nodos de *hicum0.va,* como para TFTs sólo se tienen 3 terminales (drenaje, compuerta y fuente) se usan d, g, s respectivamente, y los restantes (e, tnode) se conectarán siempre a 0 Volts o a tierra.

Se ejecuta entonces una vez más el último comando.

Debido a que NGspice no dice específicamente cuál es el error, lo que queda es verificar cada sección de código hasta dar con el problema, una vez hecho todos los pasos necesarios se captó lo siguiente como posible error:

- Declaración de constantes, estás deben agregarse directamente en la sección del código donde se utilizan, no hay que declararlas.
- Existe la posibilidad de que el uso de ecuaciones extensas a NGspice le dé problemas al realizar las operaciones, es mejor ponerlas por partes.
- El resultado de operaciones que den como resultado valores cercanos a cero, puede producir que se indefinan otras ecuaciones, se recomienda sumarle un número muy pequeño tal que no se vea afectado el resultado de simulación. Este proceso puede llevarle más tiempo del esperado debido a la búsqueda del problema, así como la verificación de cada una de las ecuaciones y el valor que esté resultando.

7. Verificar si se ha creado el archivo Makefile.am, con la siguiente instrucción, (se observara un resultado similar al de la Figura 4.2):

```
[simon@stark] $ ls
```

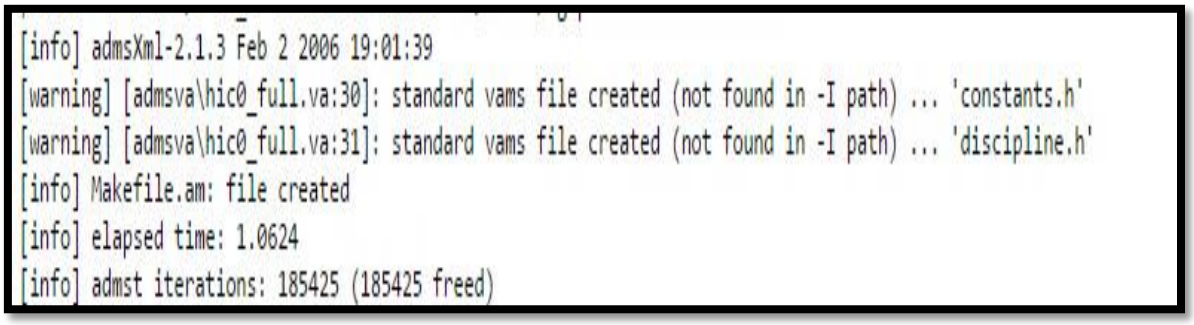


Figura 4.2. Archivo Makefile.am creado correctamente.

8. Una vez solucionado el paso anterior se continua con la compilación, para lo cual se accede con la otra ventana de terminal a NGspice solamente, con el siguiente comando:

```
[simon@stark] $ cd /usr/local/bin/ngspice-26/
```

En la dirección indicada se ejecuta:

```
[simon@stark] $ ./autogen.sh –adms

[simon@stark] $ ./configure –enable-adms
```

En donde la primera instrucción (*autogen*) es utilizada para simplificar la creación y compilación de programas, mientras que la segunda (*configure)* se utiliza para verificar y configurar la maquina en que haremos la compilación.

9. En la carpeta de *hicum0* abierta anteriormente en la terminal se ejecuta lo siguiente para crear los módulos que se requieren para continuar la compilación:

```
[simon@stark] $ admsXml –Iadmsva admsva/hicum0.va -e
../admst/ngspiceVersion.xml       -e
../admst/ngspiceMakefile.am.xml       -e
../admst/ngspiceMODULEitf.h.xml       -e
../admst/ngspiceMODULEinit.c.xml       -e
../admst/ngspiceMODULEinit.h.xml       -e
../admst/ngspiceMODULEext.h.xml       -e
../admst/ngspiceMODULEdefs.h.xml       -e
../admst/ngspiceMODULEask.c.xml       -e
../admst/ngspiceMODULEmask.c.xml       -e
../admst/ngspiceMODULEpar.c.xml       -e
../admst/ngspiceMODULEmpar.c.xml       -e
../admst/ngspiceMODULEload.c.xml       -e
../admst/ngspiceMODULEacld.c.xml       -e
```

```
../admst/ngspiceMODULEtemp.c.xml        -e
../admst/ngspiceMODULEsetup.c.xml        -e
../admst/ngspiceMODULEguesstopology.c.xml        -e
../admst/ngspiceMODULE.c.xml
```

10. Se verifica si se crearon todos los archivos en el directorio de hicum0 (aparecerá un similar a la Figura 4.3), con el siguiente:

```
[simon@stark] $ ls
```

```
$ ls
Makefile     constants.h     hic0_fullacld.c  hic0_fullguesstopology.c  hic0_fullload.c  hic0_fullsetup.c
Makefile.am  discipline.h    hic0_fullask.c   hic0_fullinit.c           hic0_fullmask.c  hic0_fulltemp.c
Makefile.in  hic0_full.c     hic0_fulldefs.h  hic0_fullinit.h           hic0_fullmpar.c
admsva       hic0_fullext.h  hic0_fullitf.h           hic0_fullpar.c
```

Figura 4.3. Archivos generados.

11. En el mismo directorio de hicum0 se utiliza para crear archivos de objeto con extensión * .o:

```
[simon@stark] $ make CFLAGS=”-g”
```

12. Se verifica la creación de dichos archivos con (Ver Figura 4.4):

```
[simon@stark] $ ls
```

```
$ ls
Makefile     hic0_full.c     hic0_fullask.o           hic0_fullinit.h  hic0_fullmask.o  hic0_fullsetup.o
Makefile.am  hic0_fulldefs.h hic0_fullinit.o          hic0_fullmpar.c  hic0_fulltemp.c
Makefile.in  hic0_full.o     hic0_fullext.h           hic0_fullitf.h   hic0_fullmpar.o  hic0_fulltemp.o
admsva       hic0_fullacld.c hic0_fullguesstopology.c hic0_fullload.c  hic0_fullpar.c   libhic0_full.a
constants.h  hic0_fullacld.o hic0_fullguesstopology.o hic0_fullload.o  hic0_fullpar.o
discipline.h hic0_fullask.c  hic0_fullinit.c          hic0_fullmask.c  hic0_fullsetup.c
```

Figura 4.4. Archivos *.o generados.

13. Ubicándonos en el directorio raíz en donde se encuentra el programa NGspice se procede a finalizar la implementación, en esta sección se analizan los posibles errores que hayan sido saltados, aquí se llevan a cabo las ecuaciones jacobianas por ejemplo, utilizadas para la simulación de modelos. Así entonces se ejecuta el último comando de compilación (este proceso puede tardar de entre 5 a 20 minutos dependiendo el rendimiento de la máquina utilizada):

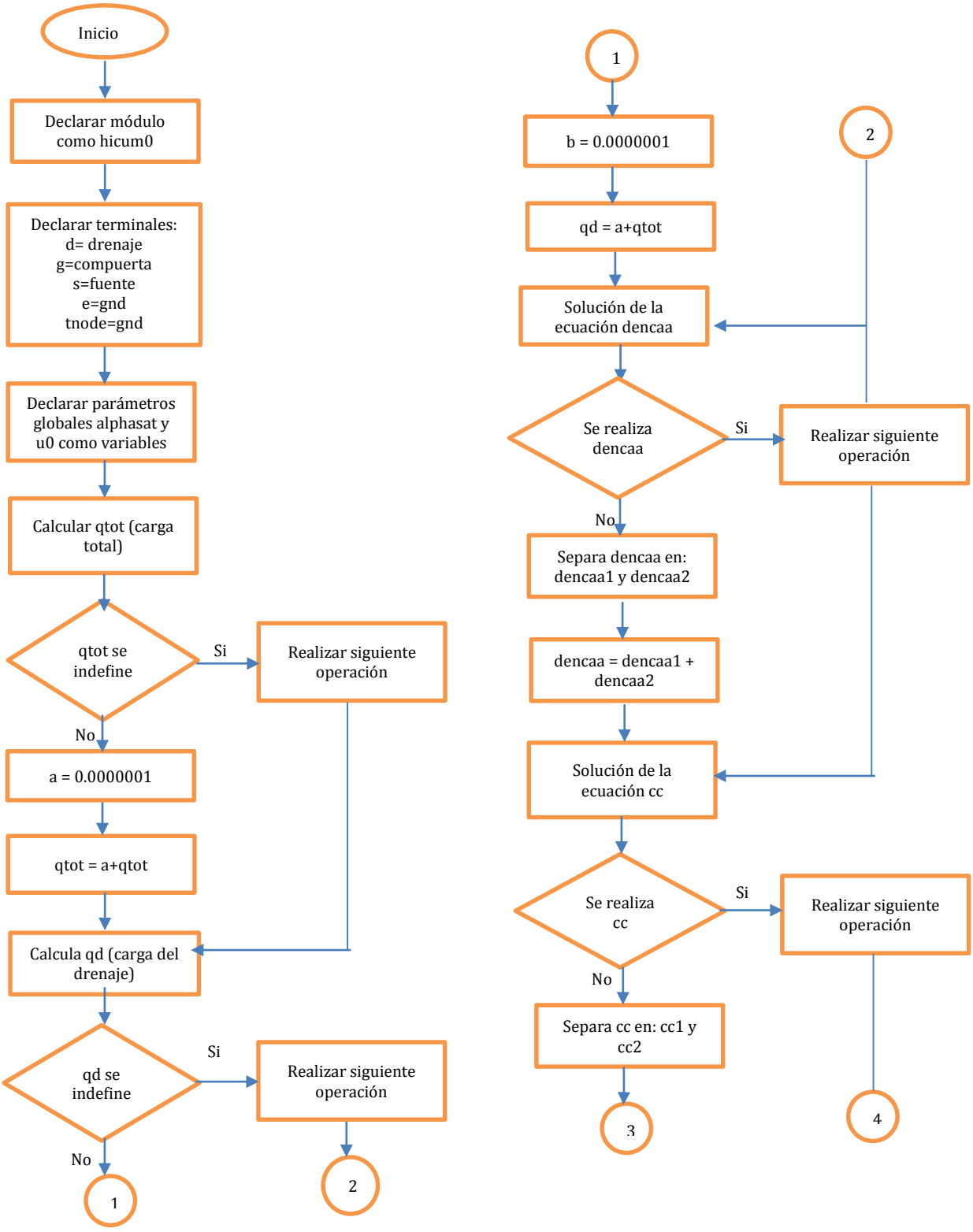
```
[simon@stark] $ make CFLAGS=”-g”
```

14. Se finaliza entonces con la compilación, haciendo la instalación en NGspice del modelo.

```
[simon@stark] $ make install
```

Hasta el momento se ha presentado la metodología que se debe seguir para llevar a cabo la compilación. Por lo cual es importante mencionar que esto es posible sí y solo sí el modelo que se implementa es totalmente compatible en sus expresiones analíticas (ej. funciones matemáticas, sintaxis, implementación de subfunciones, rango de validez de las funciones, etc.) con NGspice. Por lo tanto, debido a que el modelo original de los OTFTs no cumplía con estas condiciones, se realizaron varias modificaciones durante todo el proceso de compilación de acuerdo a los problemas que se iban generando, además de que se tuvo que verificar incluso renglón por renglón para verificar donde se encontraban los posibles errores. Como resultado en la Figura 4.5 se dan a conocer las modificaciones que se realizaron para poder implementar el modelo sin problemas de convergencia.

Una vez hecho esto, si se desea cambiar o modificar el modelo es necesario realizar todos y cada uno de los pasos ya mencionados, o bien se pueden seguir la información otorgada en el anexo C, el modelo con y sin modificaciones se encuentran en el anexo A y 4, respectivamente.


Inicio
Declarar módulo como hicum0
Declarar terminales: d= drenaje g=compuerta s=fuente e=gnd tnode=gnd
Declarar parámetros globales alphasat y u0 como variables
Calcular qtot (carga total)
qtot se indefine
Si
Realizar siguiente operación
No
a = 0.0000001
qtot = a+qtot
Calcula qd (carga del drenaje)
qd se indefine
Si
Realizar siguiente operación
No
1
2
1
b = 0.0000001
qd = a+qtot
Solución de la ecuación dencaa
Se realiza dencaa
Si
Realizar siguiente operación
No
Separa dencaa en: dencaa1 y dencaa2
dencaa = dencaa1 + dencaa2
Solución de la ecuación cc
Se realiza cc
Si
Realizar siguiente operación
No
Separa cc en: cc1 y cc2
3
2
4

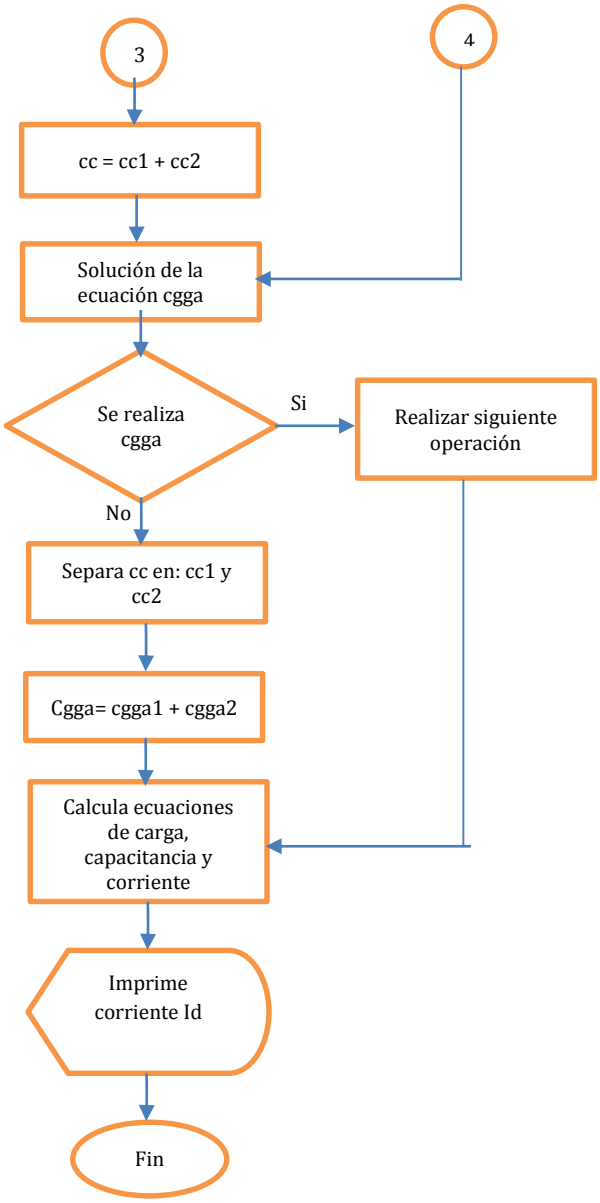


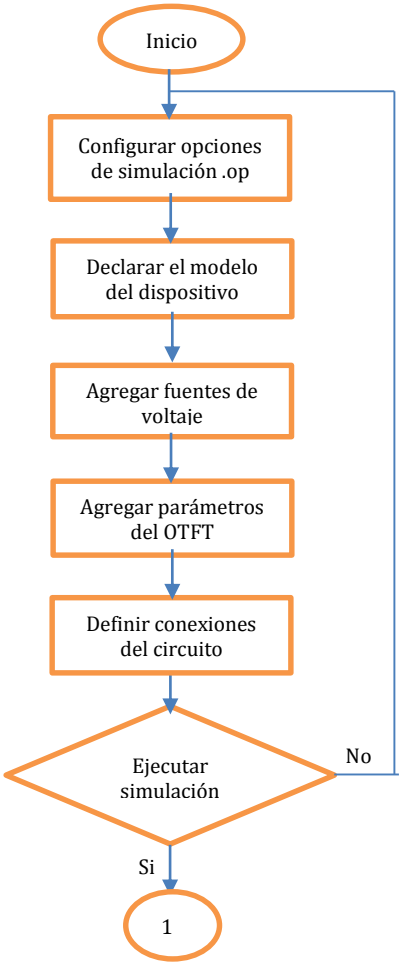
Figura 4.5. Diagrama de flujo de las modificaciones realizadas en el modelo.

CAPÍTULO 4

RESULTADOS DE SIMULACIÓN EN NGSPICE

4.1 Oscilador de anillo

El propósito de este trabajo es hacer la simulación de circuitos a través del modelo analítico implementado en NGspice por lo que se simuló un circuito digital (oscilador de anillo) y un circuito analógico (amplificador) no sin antes primero simular las curvas características del OTFT tipo P. Esto permitirá verificar que la descripción del modelo analítico en Verilog-A e implementado en NGspice coincide con los resultados que se tengan disponibles en artículos científicos con el fin de validar la implementación. Para llevar a cabo la simulación de circuitos en NGspice se debe seguir el siguiente método mostrado en la Figura 5.1.



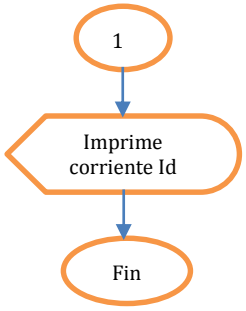


Figura 5.1. Diagrama de flujo para hacer un netlist en NGspice.

4.1.1 Simulación del OTFT tipo P

El modelo implementado en NGspice ha sido validado por medio de resultados de simulación en SmartSpice de Silvaco ® además de que se tienen también resultados experimentales de circuitos digitales reportados en [67], donde se hace el uso de parámetros mostrados en la Tabla 5.1.

Los resultados experimentales que se han obtenido corresponde a el material orgánico dinaphtho[2,3-b:2',3'-f] thieno[3,2-b]thiophene DNTT como semiconductor orgánico con similar movilidad a la del pentaceno y tri (propyleneglycol) diacrylate TPGDA como aislante de la compuerta, y con este mismo material se ha fabricado y simulado un oscilador de anillo como se reporta en [67]. Cabe mencionar que se hizo uso de los mismos parámetros del dispositivo para llevar a cabo la simulación de un oscilador de anillo y poder comprobar la implementación del modelo en el software libre. A continuación mostramos el resultado de simulación en la Figura 5.2, del OTFT tipo P.

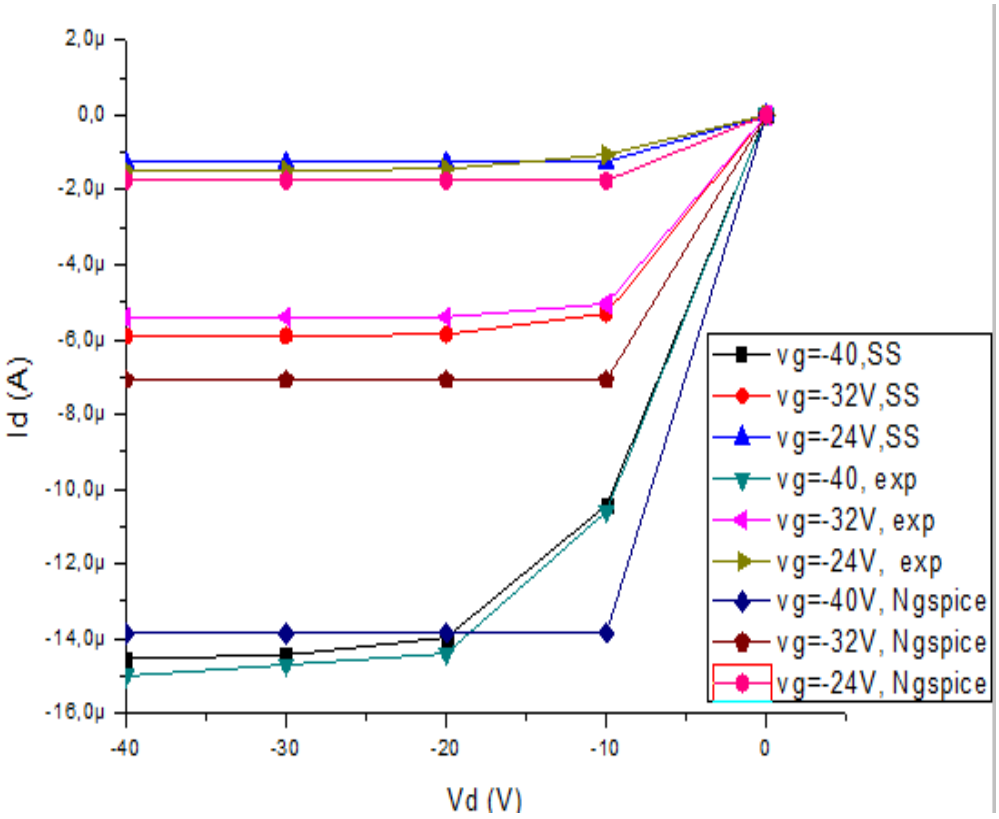


Figura 5.2. Curvas de salida para el OTFT tipo P con Vd de 0 a -40 V, SS (SmartSpice nivel 37), art exp [67] y NGspice, los netlist correspondientes a SmartSpice y NGspice se encuentran el anexo E.

Parámetro	Valor	Parámetro	Valor
$\mathbf{W(\mu m)}$	2000	Γ	0.031
$\mathbf{L(\mu m)}$	200	$\lambda(S)$	0
$\mathbf{C_i(F/cm^2)}$	$4.83x10^{-9}$	M_{SAT}	3.90
$\mathbf{V_T\ (V)}$	-17.86	A_{SAT}	1.19
$\mathbf{V_O(V)}$	0.948	$\sigma_0(S)$	$8.66x10^{-15}$
$\mathbf{V_{ACC}}$	1	$R_S(\Omega)$	0
$\mathbf{\mu_{ACC}\left(cm^2/_{Vs}\right)}$	1.07	$R_D(\Omega)$	0

Tabla 5.1. Parámetros usados en la simulación de NGspice y SmartSpice tomados [67].

Después de simular las curvas de salida para los OTFTS se obtuvo el porcentaje de error, para las curvas mostradas en la Figura 5.2 para las simulaciones de SmartSpice y NGspice.

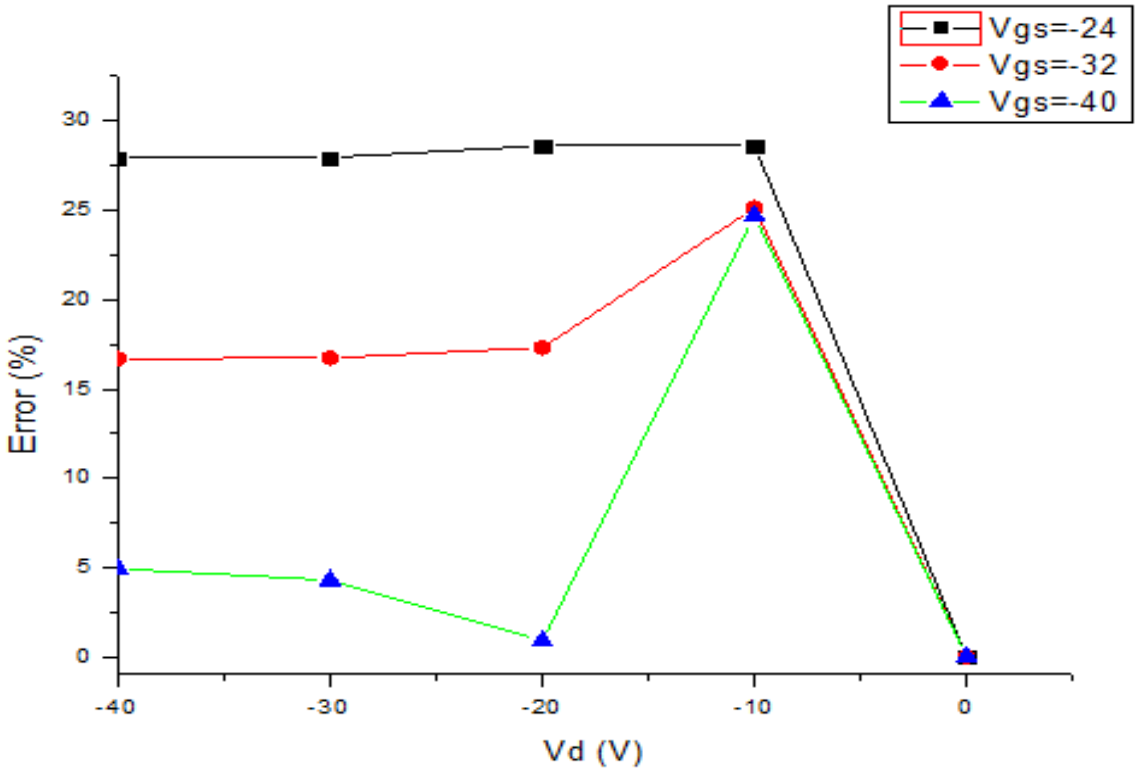


Figura 5.3. Gráfica de error porcentual para diferentes niveles de voltaje, Vgs = -24, -32 y -40 V.

Como se observa en la Figura 5.3 se calculó el error promedio para Vgs = -24, -32 y -40 V con Vd = -10, -20, -30 y -40 V. Existen muchas razones por las que error sea notable, de las cuales uno de ellos es la falta de información más completa acerca del dispositivo que se está simulando o bien llevar a la fabricación el dispositivo para ver la manera de poder incorporar nuevas ecuaciones a este modelo UMEM de OTFTS. Así también, es necesario el contar con mayor cantidad de datos medidos para poder realizar la extracción de parámetros de acuerdo a la implementación del modelo.

4.1.2 Simulación de oscilador de anillo

Actualmente la electrónica ha revolucionado el estilo de vida actual en gran medida gracias a los avances que ha tenido la electrónica digital en las últimas décadas, algo muy importante de la electrónica digital es el desarrollo de circuitos tales como inversores, compuertas lógicas, flip-flops, memorias, microprocesadores y otros más, sin duda mencionamos también a los osciladores de anillo que se diseña a través de inversores lógicos, donde el funcionamiento digital parte del concepto de tener respuestas de 0 y 1, bajo y alta, o bien apagado

o encendido para tal caso que se requiera. Un oscilador de anillo es comúnmente usado en microcontroladores o en donde se requiera sustituir un cristal de cuarzo o un condensador por ejemplo, por lo que dependiendo de la frecuencia de oscilación este va dar o no paso a una señal requerida.

Un oscilador de anillo es un circuito digital compuesto por un número impar de inversores, con el que se tiene una retroalimentación como se observa en la Figura 5.4, el cual dependiendo el número de inversores será la frecuencia de oscilación provocada por cada una de las capacitancias de los transistores y también de aquellas que se usen como cargas del oscilador.

El oscilador de anillo que se toma en cuenta en esta tesis es el mismo que se simuló y fabricó en el trabajo ya mencionado anteriormente [64], el cual se compone de 5 etapas como se muestra en la Figura 5.4, donde cada etapa es un inversor lógico.

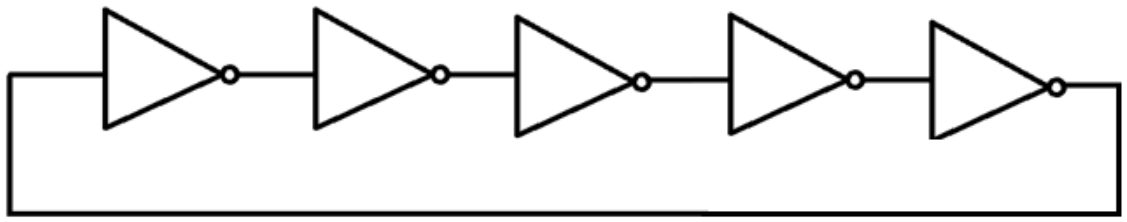


Figura 5.4. Oscilador de anillo de 5 etapas.

El oscilador de anillo a simular se muestra en la Figura 5.5 b) (el netlist del circuito se encuentra en el anexo F). En nuestro caso se utilizó un Vdd = -60 V, por lo que se espera que el circuito oscile aproximadamente en 1.3 kHz como se indica en la Figura 5.5 a) que es cuando el dispositivo está nuevo y aun no sufre degradación, causada por el oxígeno presente en el aire.

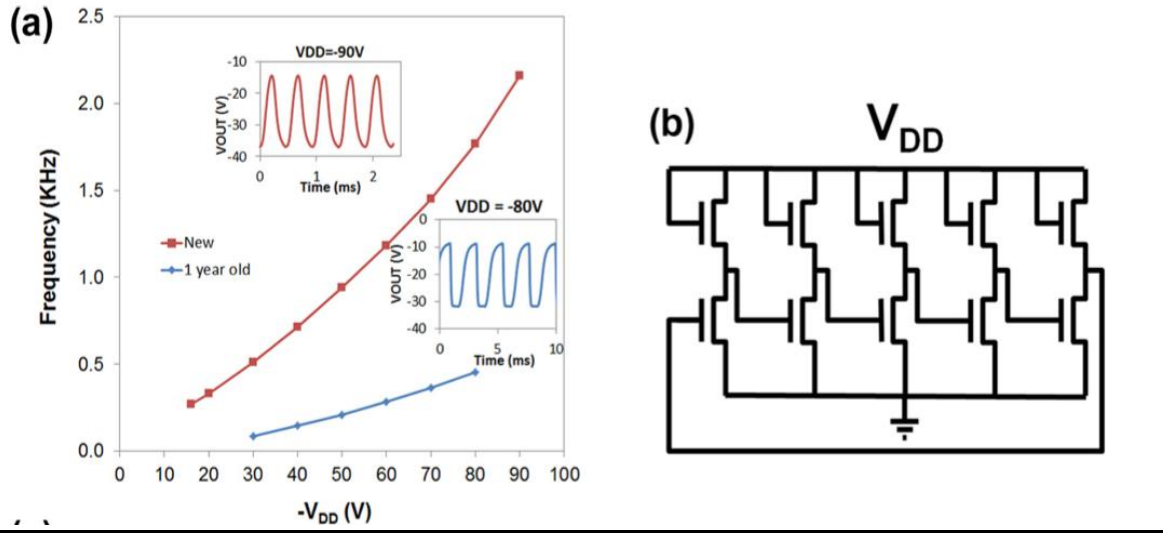


Figura 5.5. a). Frecuencia de salida en funcion de Vdd en un rango de 0 a -90 V cuando el dispositivo esta nuevo y cuando tiene un año de que se fabricó, b) circuito del oscilador de anillo con transistores tipo P [67].

Una vez que se tiene el modelo del OTFT implementado y que se realizó el netlist del circuito se procede a ejecutar la simulación en NGspice y se hace de la siguiente manera:

Siguiendo los siguientes pasos en la terminal se obtiene una respuesta en modo grafico (ver anexo G):

```
simon@stark] $ ngspice

   -> source oscilador_anillo.sp

   -> tran 500us 10ms

   -> plot V(1)
```

El resultado transitorio del oscilador de anillo se muestra en la Figura 5.6, con frecuencia de oscilación $\approx 2.3\ kHz$. Sin duda es posible observar que la frecuencia no es la misma debido a que los resultados mostrados son tomados experimentalmente y estos pueden estar alterados por efectos parásitos del dispositivo como lo son las capacitancias que se forman entre los electrodos de la fuente y el drenaje, al igual que también las resistencias RD y RS puede tener efectos que no se consideran en el modelo, sabiendo esto se puede mejorar el circuito o bien con más estudio de la estructura del OTFT se puede contribuir al

modelo de Verilog-A por medio de la fabricación y un profundo estudio de su comportamiento.

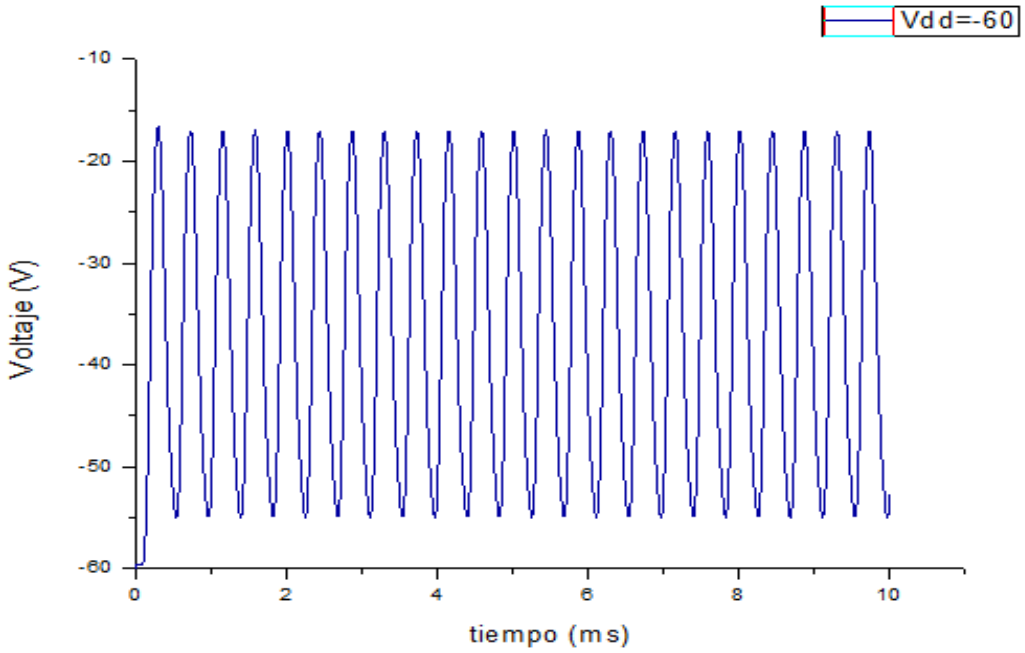


Figura 5.6. Respuesta transitoria del oscilador de anillo con OTFTS, obtenida en Ngspice, usando Origin para graficar los datos.

Debido a que el oscilador de anillo está simulado a partir de un nuevo modelo para transistores orgánicos no es posible calcular o tener ecuaciones matemáticas para calcular los tiempos de oscilación al igual que la frecuencia como se hace con los transistores comunes de Silicio, es por eso que sólo se aproxima tomando en cuenta el periodo de la onda de oscilación, ver Figura 5.7.

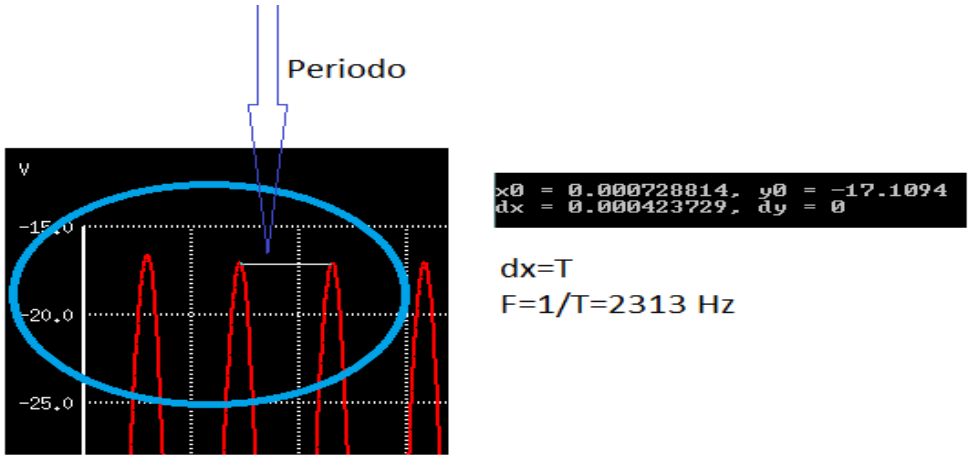


Figura 5.7. Frecuencia de oscilación.

4.2 Amplificador diferencial en NGspice

Un amplificador diferencial (o par diferencial) es llamado a aquel amplificador cuya salida es proporcional a las entradas, ejemplo clásico de un par diferencial es el de la Figura 5.8.

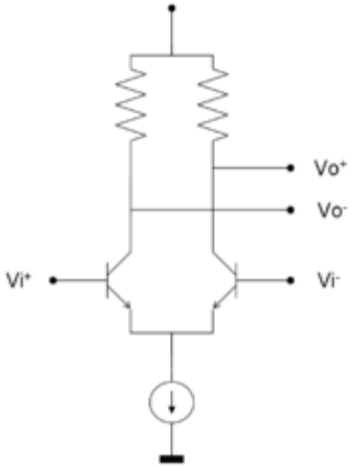


Figura 5.8. Par diferencial con transistores bjt.

Los amplificadores diferenciales están constituidos por transistores, ya sea bjt, mosfet, TFT, etc., que por ende éstos dispositivos han tenido un papel muy importante en la revolución de la electrónica analógica. Debido a que es posible el hecho de poder medir señales muy pequeñas por medio de un sistema de

amplificación. Esto ha resaltado más el uso de amplificadores diferenciales ya que es posible obtener señales de salida en tensión o en corriente.

Cuando hablamos de los amplificadores es necesario tomar en cuenta la ganancia y el ancho de banda del circuito.

Se puede hablar entonces de la ganancia en lazo abierto de un amplificador para determinar la precisión de un sistema retroalimentado empleando un amplificador operacional. La ganancia puede ser determinada mediante parámetros como la transconductancia (gm) y la resistencia R_D.

4.2.1 Amplificador de una etapa

Para la validación del modelo para circuitos analógicos se ha usado una configuración tipo single-ended, por lo que esta configuración permite medir una señal de salida respecto a un potencial de entrada fijo que está dado como un potencial medio respecto a Vdd y Vss.

En la Figura 5.9 se muestra una configuración de un amplificador diferencial con salida single-ended, donde la ganancia está determinada por $g_{mN}(r_{ON} \parallel r_{OP})$, donde g_{mN} es la transconductancia del transistor tipo N (M1=M2), r_{ON} y r_{OP} son las resistencias de salida para el transistor tipo N y tipo P respectivamente. El ancho de banda para este circuito está determinado por la capacitancia de carga C_L, además notemos que esta configuración usa un espejo tipo P y una fuente de corriente Iss.

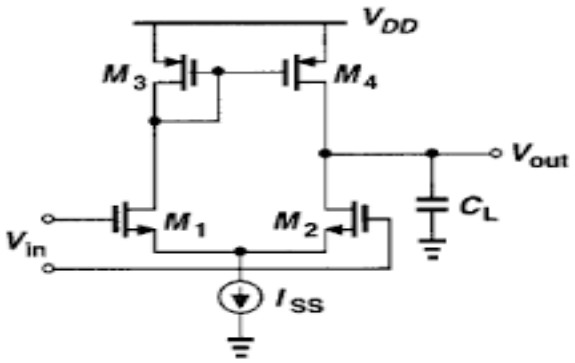


Figura 5.9. Amplificador con salida única (single-ended) [68].

Es importante tener en cuenta la fuente de corriente Iss con la que se trabajara, debido a que en muchos de los casos se puede sustituir por un espejo de corriente, usando transistores en la región de saturación y pueden actuar como fuente de corriente, el cual para la validación se hará uso de la configuración de la Figura 5.9 pero con un espejo de corriente como el que se muestra a continuación.

4.2.1.1 Espejo de corriente

Un espejo de corriente como el de la Figura 5.10 usando M1 y M2 y con tamaños iguales se espera que $V_{GS1} = V_{DS1} = V_{GS2}$, porque se tiene los mismos voltajes de compuerta a fuente, entonces se espera una misma corriente de drenaje y el mismo potencial cuando las resistencias son iguales.

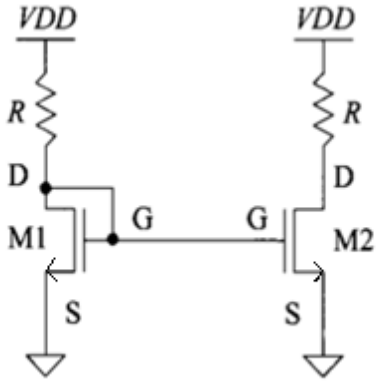


Figura 5.10. Espejo de corriente [68].

Operación en DC

Cuando se tiene un amplificador diferencial se considera el diseño en base a la corriente Iss que se tiene, por ejemplo, consideremos la Figura 5.11 basándonos en que:

$$I_{SS} = i_{D1} + i_{D2}$$

Se asume que cuando los potenciales de compuerta son iguales ambos transistores están en la región de operación lo cual es necesario considerarlo para poder seleccionar el tipo señal que podemos amplificar.

Para poder determinar la corriente i_D se considera cuando el transistor opera en saturación lo que para un transistor de Silicio

$$i_D = \frac{\beta_n}{2} (V_{GS} - V_{THN})^2$$

Y para los OTFTS es:

$$I_{Ds} = \frac{W}{L} C_i \mu_{FET} \frac{(V_{GS} - V_T)^{1+\gamma} V_{Ds}(1 + \lambda |V_{Ds}|)}{\left[1 + R\frac{W}{L} C_i \mu_{FET}(V_{GS} - V_T)\right]\left[1 + \left(\frac{V_{Ds}}{VDsat}\right)^m\right]^{\frac{1}{m}}}$$

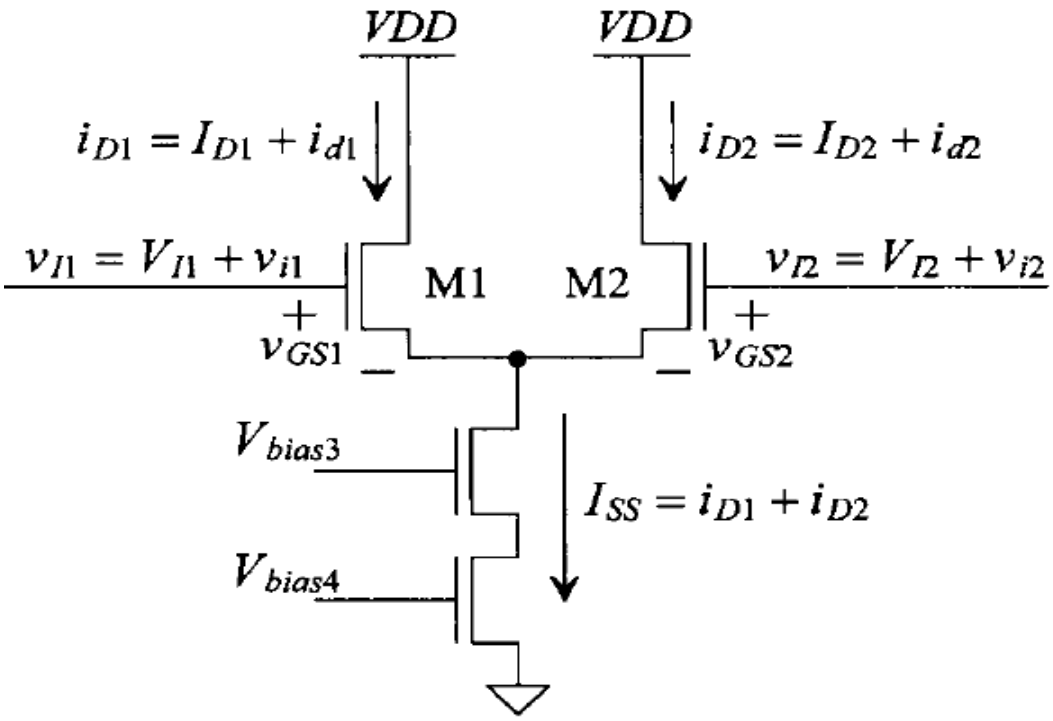


Figura 5.11. Amplificador diferencial [68].

Donde Iss= $I_{D1} + I_{D2}$

4.2.2 Amplificador diferencia con espejo de corriente

De la Figura 5.9, se determina que el voltaje máximo de salida está dado por el transistor M4 en saturación:

$$V_{OMzx} = VDD - V_{SD,sat}$$

Si se realiza un análisis en modo diferencial la respuesta de salida en DC para un amplificador diferencial debería ser idealmente como el de la Figura 5.12, esto sucede cuando las señales de entrada están en modo diferencial, por ejemplo para este caso se tiene el punto de operación en 2.5 aproximadamente (cruce de las curvas), que es donde se encuentran los transistores en saturación y se puede

amplificar una señal para ese punto de Vgs. El espejo de corriente es utilizado para que el amplificador no se encuentre muy limitado en cuanto al rango de operación.

La explicación general que se ha dado es necesaria para poder validar el circuito analógico, es decir, para se obtenga de la mejor manera posible un comportamiento que se parezca a la teoría explicada.

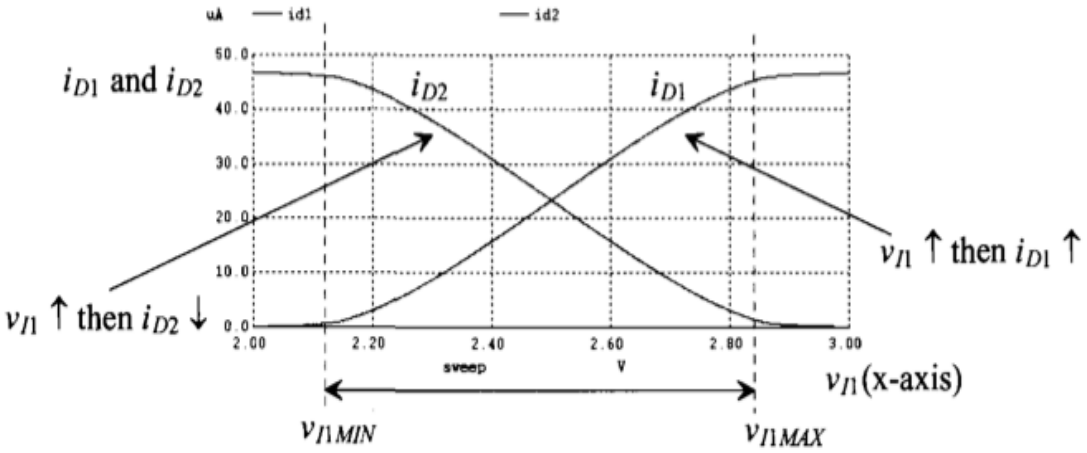


Figura 5.12. Operación en DC del amplificador [68].

4.2.2.1 Simulación del Amplificador en NGspice

Para la simulación del amplificador con OTFTS se hace uso de la configuración similar al de la Figura 5.13 en modo single-ended, con un espejo de corriente que es tipo P, es decir, la ganancia va estar determinada por la gm(rp||rn), donde es gm del transistor tipo P, y rm y rp son las resistencias del transistor tipo N y P respectivamente.

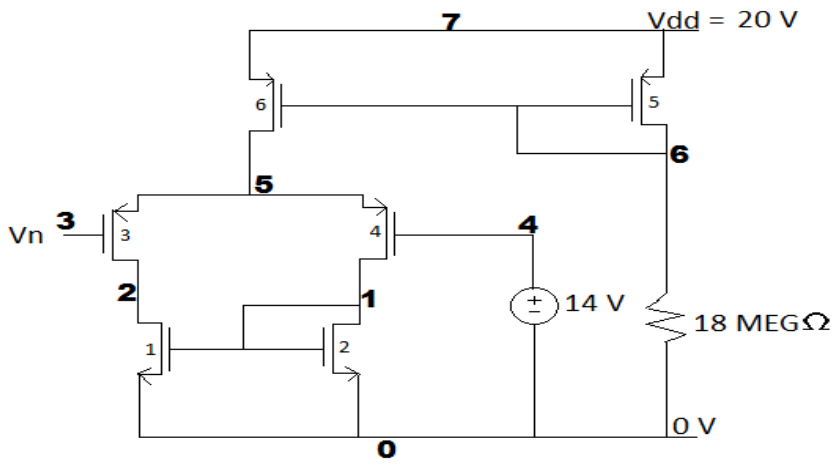


Figura 5.13. Amplificador con salida única (single-ended).

Para realizar la simulación del amplificador se utilizaron los parámetros dados por [42], los cuales corresponden a transistores orgánicos, donde se utilizó pentaceno y N,N'-ditridecylperyne-3,4,9,10-tetracarboxylic diimide (PTCDI-CC13h27) como semiconductor orgánico y PMMA como aislante. Se hace uso de la información [43] ya que tenemos la posibilidad de usar transistores tipo p y tipo n del cual nos ayudan en el diseño analógico aunque por el momento no hay reporte de circuitos analógicos con este modelo de OTFTS, se realizó un diseño de un amplificador para después en caso de ser posible, pueda ser fabricado y de esta manera se pueda comprobar la respuesta de simulación que ha obtenido hasta el momento con la implementación el modelo UMEM de OTFT.

Parámetro	**Tipo P**	**Tipo N**
L	$120x10^{-6}$	$120x10^{-6}$
W	$500x10^{-6}$	$500x10^{-6}$
$\boldsymbol{\lambda}$	0.00095852	0
$\boldsymbol{R_0}$	10^{10}	10^{10}
m	1.5422	3.4656
$\boldsymbol{T_{ox}}$	$165x10^{-9}$	$165x10^{-9}$
$\boldsymbol{V_t}$	0.70869	5.4272
$\boldsymbol{\gamma}$	0.133237	0.1
$\boldsymbol{\alpha}$	0.88734	0.61682
R	837.6009	0
Vaa	0.014047	20100
$\boldsymbol{\mu_0}$	$4.2x10^{-6}$	$6.9x10^{-6}$

Tabla 5.2. Parámetros para el amplificador para OTFT tipo P y tipo N [43].

El netlist se encuentra en el anexo H correspondiente al circuito de la Figura 5.12

En los resultados de simulación del amplificador en señal DC, el nivel de dc está en 14 V, por lo que la señal que amplifiquemos deberá estar en el mismo nivel de voltaje de entrada. Considerando lo anterior, se procede entonces a realizar la simulación en NGspice. Como es posible observar en el netlist del amplificador, se realizó la simulación en modo DC, transitorio y AC, de los cuales se puede ver el resultado de las simulaciones en las Figuras 5.14, 5.15, 5.16 respectivamente.

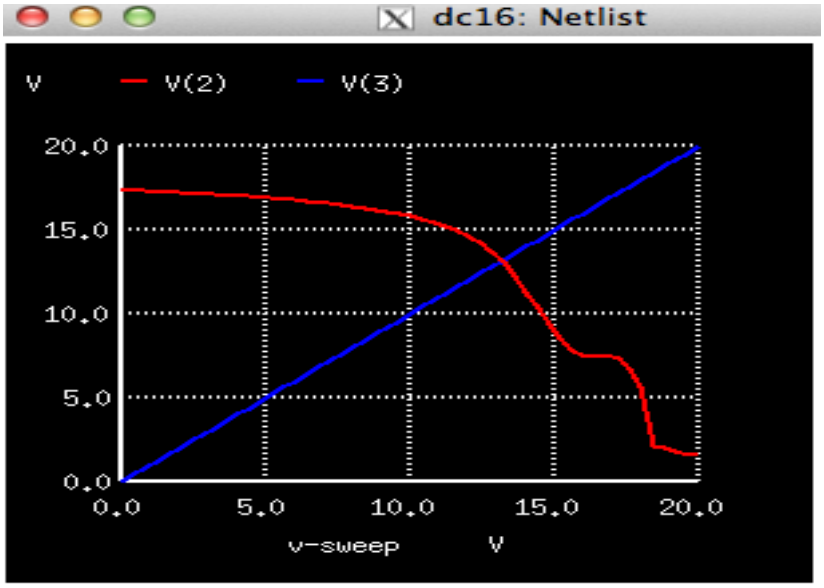


Figura 5.14. Operación en DC del amplificador, V (2) es la salida y V (3) es la entrada.

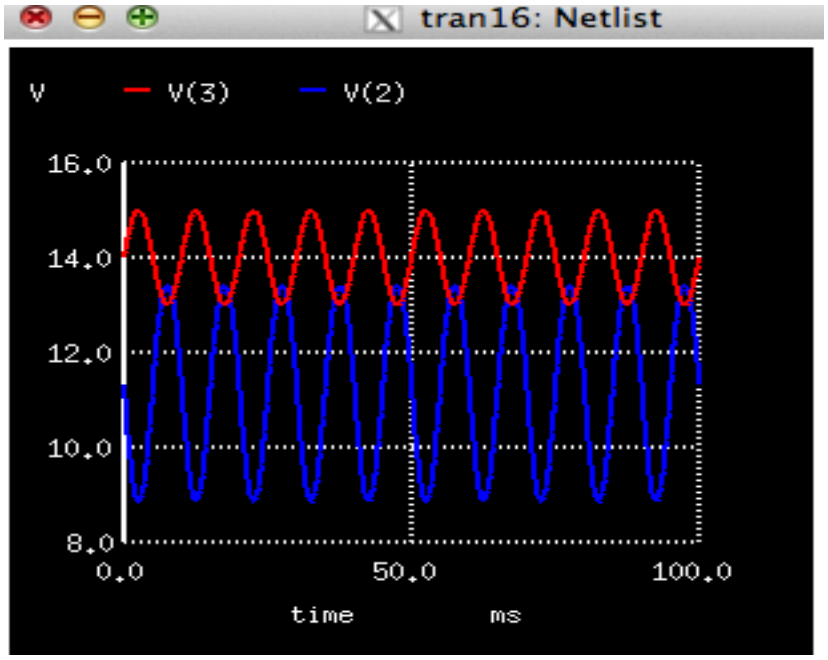


Figura 5.15. Respuesta transitoria del amplificador, V(3) entrada y V(4) salida.

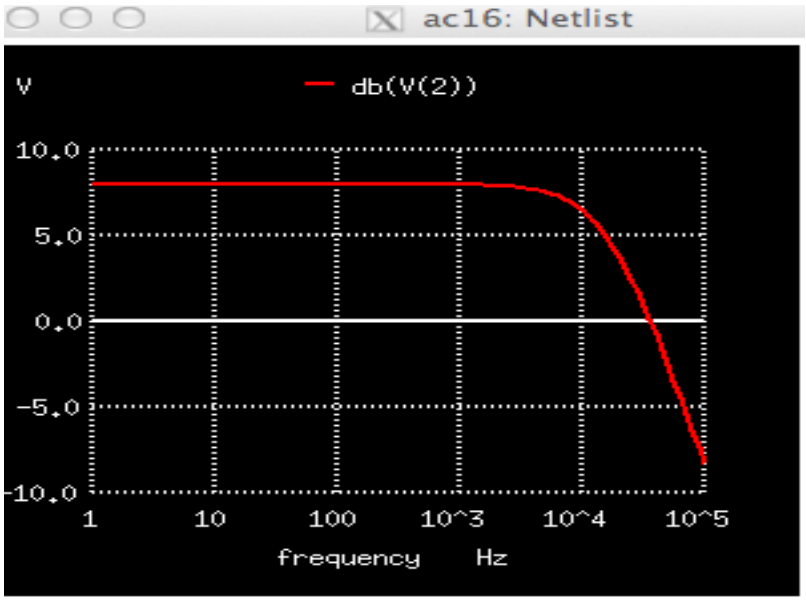


Figura 5.16. Respuesta AC en dB.

El amplificador que se ha simulado obtiene una ganancia $Av \approx 3.4 \approx 10dB$. Respecto a los resultados se menciona que fueron obtenidos de forma empírica debido a que el hecho de realizar un diseño a través de ecuaciones matemáticas conlleva a la solución de nuevas ecuaciones que por el momento no están definidas, por lo que nos enfocamos a conocer el modelo de OTFTS y observar las respuestas que obtenemos al cambiar los parámetros.

CAPÍTULO 5

CONCLUSIONES

- En la presente tesis se trabajó con el modelo analítico UMEM (modelo unificado y método de extracción, por sus siglas en inglés) para transistores orgánicos de película delgada (OTFT). Es importante mencionar que este modelo muestra varias ventajas. La más importante es que éste es válido para las diferentes regiones de operación del transistor la cual comprende desde subumbral hasta el sobre umbral, así como en lineal y saturación.

- Se concluye además, que el modelo utilizado es uno más de los grandes avances que se tienen en este ámbito de la investigación. Una de las razones por lo cual se dice esto, es porque el modelo es la unificación de conceptos, en donde, se incorporan efectos de carga y capacitancia basados en el efecto de campo del transistor, haciendo efectivo el uso de ecuaciones que describen el comportamiento del dispositivo, ya que este incluye parámetros que tienen significado en cuanto a la física del material semiconductor orgánico, que como se ha dicho anteriormente no se tenía un modelo para este dispositivo OTFT, el cual usara parámetros reales y no aproximaciones que siguen el comportamiento de un transistor convencional.

- Así también se ha abordado la implementación de este modelo analítico UMEM en un simulador de acceso libre, en el cual fue posible llevar a cabo la simulación de circuitos de manera eficaz, haciendo uso de topologías existentes para transistores de Silicio, es decir, utilizando configuraciones básicas de diseño tanto digital como analógicos.

- Es importante mencionar que la implementación de este modelo así como su metodología estarán al alcance de los estudiantes, investigadores e interesados en diseñar circuitos con estos OTFTS, dando así la posibilidad de obtener mejoras en cuanto a diseño o inclusive aportar mejoras al modelo UMEM, por lo que queda abierta la opción de trabajar con esta tesis.

-

- Vale la pena recalcar, que es necesario considerar que los OTFTS no tienen un comportamiento igual al de los transistores de Silicio. Lo cual provoca que no se pueda contar hasta el momento con una metodología de diseño de circuitos confiable como en el caso de los transistores de silicio. Por lo tanto en este trabajo se trató de considerar el comportamiento del OTFT en la medida de lo posible en el diseño de los circuitos propuestos. Como resultado se pudo tener un buen acercamiento en los resultados entre el diseño y la simulación circuital con respecto a la literatura disponible.

- Cabe mencionar que estos circuitos simulados se proponen para llevarlos a la fabricación para que podamos verificar de manera experimental el funcionamiento obtenido en NGspice, del que se propone también incorporar efectos parásitos que aún no han sido considerado en el modelo. Además de que este trabajo sólo se ha validado para diseños digitales, debido a que no se encuentran reportados resultados e información acerca de topologías para circuitos analógicos, y esto es importante, para que así se tenga la certeza acerca del modelo y de la implementación en ambos tipos de diseño.

- Como trabajo a futuro, queda por hacer más simulaciones de circuitos analógicos y digitales, esto para observar su comportamiento, y así poder trabajar en mejoras del modelo UMEM que puedan determinar el resultado más aproximado a una respuesta experimental. Así también en proponer una metodología de diseño con este tipo de dispositivos. Por lo cual será importante trabajar en la fabricación de los circuitos que se simularon y de ser posible llevarlos a alguna aplicación donde podamos comprobar su funcionalidad.

Anexo A: Modelo UMEM en Verilog-A

En este anexo se muestra el código en Verilog-A, que como se ha dicho es el lenguaje que se utiliza para describir modelos analógicos. El código de Verilog-A corresponde al modelo analítico UMEM [69]. Este está compuesto primeramente por la inclusión de las bibliotecas, posteriormente se agrega el módulo de hic0_full, este incluye las terminales que tiene el dispositivo (5 terminales), cabe mencionar que sólo son 3 terminales las que se utilizan (d, g, s), las restantes (e y tnode) se configuran a 0 volts en el netlist del circuito. Esto se hace porque el modelo original perteneciente a hicum0 es de un dispositivo bjt, y este tiene 5 terminales y se hizo la adaptación al modelo de bjt para OTFTS. Consecuentemente se hace la declaración de parámetros globales y locales del modelo, posteriormente se inicializa el bloque análogo del transistor, donde se inicializan los nodos de voltaje, y se hace la descripción en la región lineal y de saturación, además de incluir las ecuaciones de carga y capacitancia para que finalmente se haga el cálculo de las corrientes.

```
// UMEM MODEL FOR OTFTS//
//DEPARTMENT OF ELECTRICA ENGINEERING, SEES DIE CINVESTAV-IPN, MEXICO   2012 //

`include "constants.h"
`include "discipline.h"

module hic0_full (d, g, s, e, tnode);
inout d, g, s, e, tnode;
electrical d, g, s, e, tnode;
branch (d, s) canal;

parameter real TYPE=1 from[-1:1] exclude 0;
parameter real L = 30e-6 from[0.0:inf];
parameter real W = 20e-6 from[0.0:inf];

parameter real ALPHASAT=0.847;
parameter real EPS=3.2;
parameter real EPSI=3;
parameter real GAMMA=0.117;
parameter real LAMBDA=-6.388e-4;
parameter real M=3.105;
parameter real RD =3.144e6;
parameter real RS =3.144e6;
parameter real TNOM=27;
parameter real TOX=310e-9;
parameter real VAA=3.073e26;
parameter real VTO=1.002;
parameter real SL=6;
parameter real DVL=2;
parameter real QL=0.45;
parameter real IOL=4.924e-10;
parameter real SS=6;
parameter real DVS=2;
parameter real QS=0.5;
parameter real IOS=5.1e-9;

parameter real TS=80e-9;
parameter real OFFSET=3e-14;
parameter real LOVD=15e-6;
parameter real LOVS=15e-6;
parameter real EG=1.83;
parameter real NCS=2e27;
```

```
parameter real NVS=2e27;
parameter real NB=2e23;
parameter real M1=4;
parameter real SHIFTQ=7;
parameter real DELTAQ=50;
parameter real VMINIQ=0.1;
parameter real VDMIN=0.7;
parameter real MU0=1e-4 from[1e-6:10];
parameter real CIF=1e-12 from[1e-15:10e-11];
parameter real f=10000 from[10:1000000];

real VTH, NIEG, EPSIF;//, EPSI;
real ACF, VTDF, FACTF;
real covfd; // Frequency dependent gate-drain overlap capacitance
real covfs; // Frequency dependent sorce-drain overlap capacitance
real covf; // Total frequency dependent overlap capacitance
real fisf; // Frequency and Vg dependent surface potencial
real qa,qb,qd,gamma1,gamma2,gammat,a,b,a1,b1,c,ab,a1b1,c1;
real vgc,k, k1, k2, k2D, k3, k4, expo, vdseD,vdsq, vdseq, vgtQ,vdQ;
real dencaa,dencaa1,dencaa2, caa, ca, cb, cc, cc1, cc2, cd, ce, cf, cg, ch, ck,
cd1,cd2;//WMAXF,
real qtotitot;
real qtot;
real qgate;
real qdrain;
real qsource;
real cgga1;
real cgga2;
real cgga;
real cgd;
real cgs;
real CggNum;
//#################################################################
real EPSI0, EPS0;
real MOS25cg, MUFET;
real TNOM0;
real VT;
real Vds, Vgs, Vgd;
real vgs, vgd, mode, vds;
real vgt, vgte, vdse, vsate;
real vgtdL, vgtedL, vsatedL, vdsedL;
real nsadL, gchidL, gchdL, idcdL;
real idscL;
real vgtdS, vgtedS, vsatedS, vdsedS;
real nsadS, gchidS, gchdS, idcdS;
real idscS;
real idc, ids, ifinal;
real nsa, gchi, gch, VMINI, DELTA;

analog begin

EPSIF = 2.6 + ((3.54 - 2.6)/ (1+pow(f/207,0.39)));
VMINI = 0.1;
DELTA = 1;
TNOM0 =TNOM; //+`P_CELSIUS0;
EPSI0 = 8.854214871e-12 * EPSI;
EPS0 = 8.854214871e-12 * EPS;
VT= -VTO;
MOS25cg = EPSI0 / TOX;
MUFET = (1/pow(VAA,GAMMA))*MU0; // UNITS=> (m^2)/Vs
Vds = TYPE * V(d, s);
Vgs = TYPE * V(g, s);
Vgd = TYPE * V(g, d);

//Initialize node voltages
       if (Vds >= 0.0)
              begin
              mode = 1;
              vds = Vds;
              vgs = Vgs;
              vgd = Vgd;
```

```
        end
        else
                begin //reverse mode
                mode = -1;
                vds = -Vds;
                vgs = Vgd;
                vgd = Vgs;
        end

//Description
vgt = (VMINI/2) * (1+((vgs-VT)/VMINI) + sqrt((DELTA*DELTA)+pow((((vgs-VT)/VMINI)-1),2)));

// Abobe Threshold Characteristic
vsate = ALPHASAT * vgt;
vdse = vds / (pow((1 + (pow((vds/vsate),M))), (1 / M)));
nsa = (pow(vgt ,GAMMA)) * vgt * (MOS25cg / 1.6021918e-19);
gchi = 1.6021918e-19 * nsa * MUFET * (W / L);
gch = gchi / (1 + (gchi * (RS + RD)));
idc = gch * vdse * (1 + (LAMBDA * (abs(vds)-ALPHASAT*(vgt))));

//Current
//Subthreshold Lineal Characteristic

        if (vgs < VTO) begin
                vgtdL = VT + DVL;
                vsatedL = ALPHASAT * vgtedL;
                vdsedL = vds / (pow((1 + (pow((vds/vsatedL),M))), (1 / M)));
                nsadL = (pow(vgtedL ,GAMMA)) * vgtedL * (MOS25cg / 1.6021918e-19);
                gchidL = 1.6021918e-19 * nsadL * MUFET * (W / L);
                gchdL = gchidL / (1 + (gchidL * (RS + RD)));
                idcdL = gchdL * vdsedL * (1 + (LAMBDA * vds));
                idscL = idcdL * exp((vgt / SL) * 2.3);
                ids = abs(idscL) * (1 - tanh((vgs - vgtdL) * QL)) / 2 + abs(idc) * (1
+tanh((vgs - vgtdL) * QL)) / 2 + IOL;
//              ids = 0.5 * abs(idscL) * (1 - tanh((vgs - vgtdL)* QL)) + 0.5 * abs(idc) * (1
+ tanh((vgs - vgtdL)* QL)) + IOL;
        end

//Subthreshold Saturation Characteristics
        else    begin
                vgtdS = VT + DVS;
                vgtedS = vgtdS - VT;
                vsatedS = ALPHASAT * vgtedS;
                vdsedS = vds / (pow((1 + (pow((vds/vsatedS),M))), (1 / M)));
                nsadS = (pow(vgtedS ,GAMMA)) * vgtedS * (MOS25cg / 1.6021918e-19);
                gchidS = 1.6021918e-19 * nsadS * MUFET * (W / L);
                gchdS = gchidS / (1 + (gchidS * (RS + RD)));
                idcdS = gchdS * vdsedS * (1 + (LAMBDA * vds));
                idscS = idcdS * exp((vgt / SS) * 2.3);

                ids = 0.5 * abs(idscS) * (1 - tanh((vgs - vgtdS)* QS)) + 0.5 * abs(idc) * (1
+ tanh((vgs - vgtdS) * QS)) + IOL;
        end
        ifinal = ids;
        if (mode > 0)
                I(canal) <+ TYPE * ifinal;
        else
                I(canal) <+ -TYPE * ifinal;

                covfd = CIF * W * (LOVD);
                covfs = CIF * W * (LOVS);
                covf = covfd + covfs;
                qa=( pow((vgt - vdse),(3+GAMMA))- pow((vgt),(3+GAMMA)));
                qb=(((3 + GAMMA)*(-1)* ids )/ (1 + (LAMBDA * vds)));
                qtotitot = (W*CIF)* (W*CIF) * MUFET * qa/qb ;
                vgtQ = (VMINIQ/2) * (1+((vgs-VT-SHIFTQ)/VMINIQ) +
sqrt((DELTAQ*DELTAQ)+pow((((vgs-VT-SHIFTQ)/VMINIQ)-1),2)));
                vdQ= vds / (pow((1 + (pow((vds/(ALPHASAT * vgtQ)),M1))), (1 / M1)));
                gamma1=GAMMA+2;
                gamma2=GAMMA+3;
                gammat=gamma1/gamma2;
```

```
                a=pow((vgtQ-vdQ),(GAMMA+3));
                b=pow(vgtQ,(GAMMA+3));
                ab=a-b;
                a1=pow((vgtQ-vdQ),(GAMMA+2));
                b1=pow(vgtQ,(GAMMA+2));
                a1b1=(a1-b1)+0.0000001;
                c1=(1/a1b1)*ab; //
                qtot=-W*L*CIF*gammat*c1;
qgate=1-qtot;

                qd=(W*L*CIF*(GAMMA+2))/((pow((vgtQ-vdQ),(GAMMA+2))-
pow(vgtQ,(GAMMA+2)))*(pow((vgtQ-vdQ),(GAMMA+2))-pow(vgtQ,(GAMMA+2)))+0.0000001);
                qdrain= qd*(((pow((vgtQ-vdQ),(2*GAMMA+5))-
pow(vgtQ,(2*GAMMA+5)))/(2*GAMMA+5))-((pow((vgtQ-vdQ),(GAMMA+3))-
                pow(vgtQ,(GAMMA+3)))*pow(vgtQ,(GAMMA+2))/(GAMMA+3)));
                qsource= -qgate - qdrain;
                CggNum=ddx(qgate,V(g)) + (covf);
                vgc=vgt;
                vdseD=vds / (pow((1 + (pow((vds/(ALPHASAT * (vgc+DVL))),M))), (1 / M)));
                k=W*CIF*MUFET/L;
                k1=pow((W*CIF),2.0)*MUFET*1e-4;
                k2=pow((vds/vsate),M);
                k2D=pow((vds/(ALPHASAT * (vgc+DVL))),M);
                k3=(1-tanh((vgc-DVL)*QL))/2.0;
expo=exp((2.3*vgc)/SL);
dencaa1=(k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*k3/(1+RD*k*pow((vgc+DVL),(GA
MMA+1))));
dencaa2=(k*pow(vgc,(GAMMA+1))*(1+(LAMBDA*vds))*vdse*k4/(1+RD*k*pow(vgc,(GAMMA+1))))+IOL;
dencaa= dencaa1 + dencaa2;
caa=(-1e4*k1/(GAMMA+3))*((pow((vgc-vdse),(GAMMA+2))*(GAMMA+3)*(1-vdse*k2/(vgc*(1+k2))))-
(pow(vgc,(GAMMA+2))*(GAMMA+3)));
ca=k*pow((vgc+DVL),GAMMA)*(GAMMA+1)*(1+(LAMBDA*vds))*vdseD*expo*k3/(1+RD*k*pow((vgc+DVL),(GA
MMA+1)));
cb=k*pow((vgc+DVL),GAMMA)*(1+(LAMBDA*vds))*vdseD*expo*k3*k2D/((1+RD*k*pow((vgc+DVL),(GAMMA+1
)))*(1+k2D));

cc1=k*k*pow((vgc+DVL),((2*GAMMA)+1))*(1+(LAMBDA*vds))*vdseD*expo*k3*RD*(GAMMA+1);
cc2=((1+RD*k*pow((vgc+DVL),(GAMMA+1)))*(1+RD*k*pow((vgc+DVL),(GAMMA+1)));
cc=cc1+cc2;
cd=2.3*k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*k3/(SL*(1+RD*k*pow((vgc+DVL),(
GAMMA+1))));
ce=k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*QL*(1-tanh(pow(((vgc-
DVL)*QL),2)))/(2*(1+RD*k*pow((vgc+DVL),(GAMMA+1))));
cf=k*pow(vgc,GAMMA)*(GAMMA+1)*(1+(LAMBDA*vds))*vdse*k4/(1+RD*k*pow(vgc,(GAMMA+1)));
cg=k*pow(vgc,GAMMA)*(1+(LAMBDA*vds))*vdse*k2*k4/((1+k2)*(1+RD*k*pow(vgc,(GAMMA+1))));
ch=k*k*pow(vgc,((2*GAMMA)+1))*(1+(LAMBDA*vds))*vdse*k4*RD*(GAMMA+1)/((1+RD*k*pow(vgc,(GAMMA+
1)))*(1+RD*k*pow(vgc,(GAMMA+1))));
ck=k*pow(vgc,(GAMMA+1))*(1+(LAMBDA*vds))*vdse*QL*(1-tanh(pow(((vgc-
DVL)*QL),2)))/(2*(1+RD*k*pow(vgc,(GAMMA+1))));
cd1=k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*k3/(1+RD*k*pow((vgc+DVL),(GAMMA+1
)));
cd2=k*pow(vgc,(GAMMA+1))*(1+(LAMBDA*vds))*vdse*k4/(1+RD*k*pow(vgc,(GAMMA+1)));
cgga1=((1e4*k1/(GAMMA+3))*(pow((vgc-vdse),(GAMMA+3))-pow(vgc,(GAMMA+3)))*(ca+cb-cc+cd-
ce+cf+cg+ch+ck));
cgga2=((caa/dencaa)+cgga1/pow((cd1+cd2+IOL),2.0));
cgga=cgga2+covf+OFFSET;
cgd=(2.0*(cgga-covf)*(1-pow((vgc/((2*vgc)-vdse)),2))/3.0)+covfd;
cgs=(2.0*(cgga-covf)*(1-pow(((vgc-vdse)/(2*vgc-vdse)),2))/3.0)+covfs;

                I(s) <+ TYPE * ddt(qsource);
                I(g) <+ TYPE * ddt(qgate);
                I(d) <+ TYPE * ddt(qdrain);
        end
endmodule
```

Anexo B: Modelo de hicum0.va

```
//  HICUM Level_0 Version_1.12: A Verilog-A description
//  (A simplified version of HICUM Level2 model for BJT)
//  ## It is modified after the first version of HICUM/L0 code    ##

module hic0_full (c,b,e,s,tnode);


//Node definitions

       inout          c,b,e,s,tnode;
       electrical     c     `P(info="external collector node");
       electrical     b     `P(info="external base node");
       electrical     e     `P(info="external emitter node");
       electrical     s     `P(info="external substrate node");
       electrical     ci    `P(info="internal collector node");
       electrical     bi    `P(info="internal base node");
       electrical     ei    `P(info="internal emitter node");
       electrical     tnode `P(info="local temperature rise node");


       //Branch definitions
       branch         (ci,c)         br_cic_i;
       branch         (ci,c)         br_cic_v;
       branch         (ei,e)         br_eie_i;
       branch         (ei,e)         br_eie_v;
       branch         (bi,ei)        br_biei;
       branch         (bi,ci)        br_bici;
       branch         (ci,ei)        br_ciei;
       branch         (b,bi)         br_bbi_i;
       branch         (b,bi)         br_bbi_v;
       branch         (b,e)          br_be;
       branch         (b,ci)         br_bci;
       branch         (b,s)          br_bs;
       branch         (s,ci)         br_sci;
       branch         (tnode )       br_sht;
```

Nota: sólo se anexa una parte del código de hicum0, para futuras referencias se puede localizar en el directorio: */usr/local/bin/src/SPICEli/devices/adms/admst/hicum0.va,* antes de ser modificado.

Anexo C: Metodologías para la compilación del modelo

Este anexo incluye las metodologías de compilación del modelo, así como los comandos que se utilizan para llevar a cabo el proceso de compilación e implementación.

Metodología de compilación	
Manual NGSPICE	Nueva metodología Eli NGSPICE
Identificar el modelo sobre el que se trabajará (hicum0 o hicum2, p.e)	Identificar el modelo sobre el que se trabajará (hicum0 o hicum2, p.e)
El proceso no es automático, se debe editar a mano.	El proceso no es automático, se debe editar a mano.
Guardar el código Verilog-A del nuevo modelo	Guardar el código Verilog-A del nuevo modelo
Crear archivos 'Makefile.am' , en la dirección de 'hicum0'(a)	Eliminar todos los archivos existentes de hicum0, en la dirección de 'hicum0' (1)
Actualizar/crear archivos auto/tools, en la dirección de 'NGspice' (b)	Crear archivos Makefile.am, en la dirección de 'hicum0' (2)
Editar archivo 'Makefile.am' y agregar el nuevo dispositivo 'hicum0', en la dirección de 'NGspice' (c)	Preparación automática del sistema a construir, en la dirección de 'NGspice' (3)
Crear archivo 'Makefile.in', en la dirección de 'NGspice' (d)	Configuración y actualización específicamente de 'configure.ac', en la dirección de 'NGspice' (4)
Actualizar archivo 'configure', en la dirección de 'NGspice' (e)	Crear el código C autogenerado para la compilación de 'hicum0', en la dirección de 'hicum0' (5)
Crear el código C generado automáticamente listo para la compilación de hicum0 dispositivo, en la dirección de 'hicum0' (f)	Creación de archivos objeto y librerías, en la dirección de 'hicum0' (6)
Compilar el código C autogenerado (g)	Creación de archivos objeto y librerías, en la dirección de 'NGspice' (7)
Actualizar ejecutable 'NGspice', en la dirección de 'NGspice' (h)	Instalación del modelo en 'NGspice', en la dirección de 'NGspice' (8)
Instalar NGspice, en la dirección de 'NGspice' (i)	

a. `$ admsXml admsva/hic0_full.va -Iadmsva -e ../admst/ngspiceMakefile.am.xml`
b. `$ grep hicum0 configure.ac`
c. `$ grep hicum0 Makefile.am`
```
      adms/hicum0     \
       adms/hicum0     \
```
d. `$ automake`
e. `$ configure`
f. `$ make adms`
```
admsXml.exe -Iadmsva admsva/hic0_full.va \
-e ../admst/ngspiceVersion.xml \
-e ../admst/ngspiceMakefile.am.xml \
-e ../admst/ngspiceMODULEitf.h.xml \
-e ../admst/ngspiceMODULEinit.c.xml \
-e ../admst/ngspiceMODULEinit.h.xml \
-e ../admst/ngspiceMODULEext.h.xml \
-e ../admst/ngspiceMODULEdefs.h.xml \
-e ../admst/ngspiceMODULEask.c.xml \
-e ../admst/ngspiceMODULEmask.c.xml \
-e ../admst/ngspiceMODULEpar.c.xml \
-e ../admst/ngspiceMODULEmpar.c.xml \
-e ../admst/ngspiceMODULEload.c.xml \
-e ../admst/ngspiceMODULEacld.c.xml \
-e ../admst/ngspiceMODULEtemp.c.xml \
-e ../admst/ngspiceMODULEsetup.c.xml \
-e ../admst/ngspiceMODULEguesstopology.c.xml \
-e ../admst/ngspiceMODULE.c.xml
```
g. `$ make CFLAGS="-g"`
h. `$ make`
i. `$ make install`
j. `$ cat hicum0.sp`

1. `$ make clean`
   `$ rm Makefile Makefile.in Makefile.am`
2. `$ admsXml admsva/hic0_full.va -Iadmsva -e ../admst/ngspiceMakefile.am.xml`
3. `$ ./autogen.sh –adms`
4. `$ ./configure –enable-adms`
5. `[simon@stark] $ admsXml -Iadmsva admsva/hicum0.va -e ../admst/ngspiceVersion.xml -e ../admst/ngspiceMakefile.am.xml -e ../admst/ngspiceMODULEitf.h.xml -`

```
e ../admst/ngspiceMODULEinit.c.xml                    -
e ../admst/ngspiceMODULEinit.h.xml                    -
e ../admst/ngspiceMODULEext.h.xml                     -
e ../admst/ngspiceMODULEdefs.h.xml                    -
e ../admst/ngspiceMODULEask.c.xml                     -
e ../admst/ngspiceMODULEmask.c.xml                    -
e ../admst/ngspiceMODULEpar.c.xml                     -
e ../admst/ngspiceMODULEmpar.c.xml                    -
e ../admst/ngspiceMODULEload.c.xml                    -
e ../admst/ngspiceMODULEacld.c.xml                    -
e ../admst/ngspiceMODULEtemp.c.xml                    -
e ../admst/ngspiceMODULEsetup.c.xml                   -
e ../admst/ngspiceMODULEguesstopology.c.xml           -
e ../admst/ngspiceMODULE.c.xml
```

6. `make CFLAGS="-g"`
7. `make CFLAGS="-g"`
8. `make install`

Anexo D: Modelo UMEM modificado

Con este anexo, damos a conocer el código de Verilog-A del modelo UMEM, pero ya con las modificaciones que se hicieron mención en el diagrama de flujo que corresponde.

```
// UMEM MODEL FOR OTFTs//
//  DEPARTMENT OF ELECTRICA ENGINEERING, SEES DIE CINVESTAV-IPN, MEXICO   2012 //

`include "constants.h"
`include "discipline.h"
//`define P_K         1.3806226e-23
             // Boltzmann constant in  eV/
//`define P_Q         1.6021918e-19
            //Electron charge
//`define M_PI        3.14159265358979323846
//`define P_E0        8.854214871e-12
                     //VACCUM PERMITTIVITY IN [F/m]

//Define module with 3 terminals
//module hic0_full (c,b,e,s,tnode);
module hic0_full (d, g, s, e, tnode);

//Declare input/output nodes ("inout")
inout d, g, s, e, tnode;

// type of nodes "electrical"
electrical d, g, s, e, tnode;

// Internal nodes
branch (d, s) canal;

// model parameters Definitions//////////7

//Geometry
//(*info=”1.0 N-TYPE, -1.0 P-TYPE”*)
parameter real TYPE=1 from[-1:1] exclude 0;
//(*info=”Channel length UNITS=>m”*)
 parameter real L = 30e-6 from[0.0:inf];
//(*info=”Channel Width UNITS=>m”*)
parameter real W = 20e-6 from[0.0:inf];

// MODEL PARAMETERS
//(*info=”Saturation modulation parameter UNITS=>eV”*)
 parameter real ALPHASAT=0.847 from[-5:inf];        ///PARAMETRO EDITADO
//(*info=”Relative dielectric constant of substrate”*)
 parameter real EPS=3.2;
//(*info=”Relative dielectric constant of gate insulator”*)
 parameter real EPSI=3;
//(*info=”Power law mobility parameter”*)
parameter real GAMMA=0.117;
//(*info=”Output conductance parameter UNITS=>1/V”*)
 parameter real LAMBDA=-6.388e-4;
//(*info=”Knee shape parameter “*)
 parameter real M=3.105;
//(*info=”Drain Resistance UNITS=>ohm”*)
 parameter real RD =3.144e6;
//(*info=”Source Resistance UNITS=>ohm “*)
 parameter real RS =3.144e6;
//(*info=”Parameter measurement temperature UNITS=>°C”*)
 parameter real TNOM=27;
//(*info=”Thin oxide thickness UNITS=>m”*)
parameter real TOX=310e-9;
//(*info=”Characteristic voltage for field effect mobility UNITS=>V”*)
parameter real VAA=3.073e26;
//(*info=”Zero bias threshold voltage UNITS=>V”*)
 parameter real VTO=1.002;
```

```
//(*info="Slope of the linear transfer curve in subthreshold"*)
parameter real SL=6;
//(*info="A value above VT to sew the curves below/above threshold"*)
parameter real DVL=2;
//(*info="A value to avoid imaginary numbers when sewing the 2 regions"*)
parameter real QL=0.45;
//(*info="IOFF for the linear transfer curve UNITS=>A"*)
parameter real IOL=4.924e-10;
//(*info="Slope of the saturation transfer curve in subthreshold"*)
parameter real SS=6;
//(*info="Value similar to DVL but obtained in saturation"*)
parameter real DVS=2;
//(*info="A value to avoid imaginary numbers when sewing the 2 regions"*)
parameter real QS=0.5;
//(*info="IOFF for the saturation transfer curve"*)
 parameter real IOS=5.1e-9;

// ################# UBCM parameters ######################

//(*info="Semiconductor thickness (m)"*)
parameter real TS=80e-9;
//(*info="Offset capacitance from measurements (F)"*)
 parameter real OFFSET=3e-14;
//(*info=""*)parameter real Q1C=4.0;
//(*info="Overlap length between gate and drain (m)"*)
parameter real LOVD=15e-6;
//(*info="Overlap length between gate and source (m)"*)
 parameter real LOVS=15e-6;
//(*info="Band gap of the semiconductor (eV)"*)
 parameter real EG=1.83;
// taken from P3HT
//(*info="Carrier concentration in valence band m^-3"*)
 parameter real NCS=2e27;
// = 2e21 cm^-3
//(*info="Carrier concentration in valence band m^-3 "*)
parameter real NVS=2e27;
// = 2e21 cm^-3
//(*info="Semiconductor non intentional doping m^-3"*)
 parameter real NB=2e23;
// = 2e17cm^-3
parameter real M1=4;
parameter real SHIFTQ=7;
parameter real DELTAQ=50;
parameter real VMINIQ=0.1;

parameter real VDMIN=0.7;
parameter real MU0=1e-4 from[1e-6:10];
parameter real CIF=1e-12 from[1e-15:inf];
// For calculation of basic variables
real VTH, NIEG, EPSIF;//, EPSI;

//Extrapolated 108dditiona of tanh fitting parameters
real ACF, VTDF, FACTF;

// Capacitance equations
parameter real f=10000 from[10:1000000];
//real CIF; // Frequency dependent insultor capacitance
real covfd; // Frequency dependent gate-drain overlap capacitance
real covfs; // Frequency dependent sorce-drain overlap capacitance
real covf; // Total frequency dependent overlap capacitance
real fisf; // Frequency and Vg dependent surface potencial
// real wof, wtd1, wtdf, cdf, cgg;

real qa,qb,qd,gamma1,gamma2,gammat,a,b,a1,b1,c,ab,a1b1,c1;
real vgc,k, k1, k2, k2D, k3, k4, expo, vdseD,vdsq, vdseq, vgtQ,vdQ;

real dencaa,dencaa1,dencaa2, caa, ca, cb, cc, cc1, cc2, cd, ce, cf, cg, ch, ck,
cd1,cd2;//WMAXF,
//(* desc="qtotItot" *)
real qtotitot;
//(* desc="qtot1" *)
```

```
real qtot;
//(* desc="qgate" *)
real qgate;
//(* desc="qdrain1" *)

real qdrain;
//(* desc="qsource" *)
real qsource;
real cgga1;
//(* desc="cgga2" *)
real cgga2;
//(* desc="cgga" *)
real cgga;
//(* desc="cgd" *)
real cgd;
//(* desc="cgs" *)
real cgs;
//(* desc="CggNum" *)
real CggNum;
//#################################################################

//Internal Variables
real EPSI0, EPS0;

//Variables 109dditional
real MOS25cg, MUFET;

//Temperature Variables
real TNOM0;
real VT;

//Variables for bias voltages
real Vds, Vgs, Vgd;
real vgs, vgd, mode, vds;
real vgt, vgte, vdse, vsate;

//Variables for subthreshold
real vgtdL, vgtedL, vsatedL, vdsedL;
real m1x, m2x, m3x, m4x;
real nsadL, gchidL, gchdL, idcdL;
real idscL;
real vgtdS, vgtedS, vsatedS, vdsedS;
real nsadS, gchidS, gchdS, idcdS;
real idscS;

//Variables for final results
real idc, ids, ifinal;

// Variables for description
real nsa, gchi, gch, VMINI, DELTA;
//real debug1;
analog begin

// Initial Values
EPSIF = 2.6 + ((3.54 - 2.6)/ (1+pow(f/207,0.39)));

//EPSI=EPSIF*0.67;
VMINI = 0.1;
DELTA = 1;
TNOM0 =TNOM; //+`P_CELSIUS0;
EPSI0 = 8.854214871e-12 * EPSI;
EPS0 = 8.854214871e-12 * EPS;
VT= -VTO;
MOS25cg = EPSI0 / TOX;
MUFET = (1/pow(VAA,GAMMA))*MU0; // UNITS=> (m^2)/Vs
Vds = TYPE * V(d, s);
Vgs = TYPE * V(g, s);
Vgd = TYPE * V(g, d);

//Initialize node voltages
```

```
        if (Vds >= 0.0)
                begin
                mode = 1;
                vds = Vds;
                vgs = Vgs;
                vgd = Vgd;
        end
        else
                begin //reverse mode
                mode = -1;
                vds = -Vds;
                vgs = Vgd;
                vgd = Vgs;
        end

//Description
vgt = (VMINI/2) * (1+((vgs-VT)/VMINI) + sqrt((DELTA*DELTA)+pow((((vgs-VT)/VMINI)-1),2)));

// Abobe Threshold Characteristic
vsate = ALPHASAT * vgt;
//vdse =-vsate;
if (ALPHASAT > 1) begin
                vdse =-vsate;
end
                else begin
                vdse=vds / (pow((1 + (pow((vds/vsate),M))), (1 / M)));
                //vdse=vds / (pow((1 + (pow((vds/vsate),3.9))), (1 / 3.9)));
end
//vdse=vds / (pow((1 + (pow((vds/vsate),M))), (1 / M)));
nsa = (pow(vgt ,GAMMA)) * vgt * (MOS25cg / 1.6021918e-19);
gchi = 1.6021918e-19 * nsa * MUFET * (W / L);
gch = gchi / (1 + (gchi * (RS + RD)));
idc = gch * vdse * (1 + (LAMBDA * (abs(vds)-ALPHASAT*(vgt))));

//Current
//Subthreshold Lineal Characteristic

        if (vgs < VTO) begin
                vgtdL = VT + DVL;
                vgtedL = vgtdL - VT;
                vsatedL = ALPHASAT * vgtedL;
                vdsedL = vds / (pow((1 + (pow((vds/vsatedL),M))), (1 / M)));
                //vdsedL = vds / (pow((1 + (pow((vds/vsatedL),3.9))), (1 / 3.9)));
                nsadL = (pow(vgtedL ,GAMMA)) * vgtedL * (MOS25cg / 1.6021918e-19);
                gchidL = 1.6021918e-19 * nsadL * MUFET * (W / L);
                gchdL = gchidL / (1 + (gchidL * (RS + RD)));
                idcdL = gchdL * vdsedL * (1 + (LAMBDA * vds));
                idscL = idcdL * exp((vgt / SL) * 2.3);
                ids = abs(idscL) * (1 - tanh((vgs - vgtdL) * QL)) / 2 + abs(idc) * (1
+tanh((vgs - vgtdL) * QL)) / 2 + IOL;
//              ids = 0.5 * abs(idscL) * (1 - tanh((vgs - vgtdL)* QL)) + 0.5 * abs(idc) * (1
+ tanh((vgs - vgtdL)* QL)) + IOL;
        end

//Subthreshold Saturation Characteristics
        else    begin
                vgtdS = VT + DVS;
                vgtedS = vgtdS - VT;
                vsatedS = ALPHASAT * vgtedS;
                ///vdsedS = vds / (pow((1 + (pow((vds/vsatedS),M))), (1 / M)));
                m3x=(pow((vds/vsatedS),M));
                m4x=(pow((1 + m3x), (1 / M)));
                vdsedS = vds/m4x;
                nsadS = (pow(vgtedS ,GAMMA)) * vgtedS * (MOS25cg / 1.6021918e-19);
                gchidS = 1.6021918e-19 * nsadS * MUFET * (W / L);
                gchdS = gchidS / (1 + (gchidS * (RS + RD)));
                idcdS = gchdS * vdsedS * (1 + (LAMBDA * vds));
                idscS = idcdS * exp((vgt / SS) * 2.3);

                ids = 0.5 * abs(idscS) * (1 - tanh((vgs - vgtdS)* QS)) + 0.5 * abs(idc) * (1
+ tanh((vgs - vgtdS) * QS)) + IOL;
```

```
        end
        ifinal =ids;
        if (mode > 0)
                I(canal) <+ TYPE * ifinal;

        else
                I(canal) <+ -TYPE * ifinal;

                covfd = CIF * W * (LOVD);
                covfs = CIF * W * (LOVS);
                covf = covfd + covfs;
                qa=( pow((vgt - vdse),(3+GAMMA))- pow((vgt),(3+GAMMA)));
                qb=(((3 + GAMMA)*(-1)* ids )/ (1 + (LAMBDA * vds)));
                qtotitot = (W*CIF)* (W*CIF) * MUFET * qa/qb ;

                vgtQ = (VMINIQ/2) * (1+((vgs-VT-SHIFTQ)/VMINIQ) +
sqrt((DELTAQ*DELTAQ)+pow((((vgs-VT-SHIFTQ)/VMINIQ)-1),2)));
                //vdQ= vds / (pow((1 + (pow((vds/(ALPHASAT * vgtQ)),M1))), (1 / M1)));
                if (ALPHASAT > 1)  begin
                vdQ=ALPHASAT*-1*vgtQ;
                end
                else  begin
                vdQ= vds / (pow((1 + (pow((vds/(ALPHASAT * vgtQ)),M1))), (1 / M1)));
                end
                gamma1=GAMMA+2;
                gamma2=GAMMA+3;
                gammat=gamma1/gamma2;


                a=pow((vgtQ-vdQ),(GAMMA+3));
                b=pow(vgtQ,(GAMMA+3));
                ab=a-b;
                a1=pow((vgtQ-vdQ),(GAMMA+2));
                b1=pow(vgtQ,(GAMMA+2));
                a1b1=(a1-b1)+0.0000001;
                //c=pow(a1b1,-1);//error
                //c1=log(-1*exp(a1b1));//error
                ///c1=1/a1;  salio
                c1=(1/a1b1)*ab; //
                qtot=-W*L*CIF*gammat*c1;
                qgate=-qtot;

                qd=(W*L*CIF*(GAMMA+2))/((pow((vgtQ-vdQ),(GAMMA+2))-
pow(vgtQ,(GAMMA+2)))*(pow((vgtQ-vdQ),(GAMMA+2))-pow(vgtQ,(GAMMA+2)))+0.0000001);
                qdrain= qd*(((pow((vgtQ-vdQ),(2*GAMMA+5))-
pow(vgtQ,(2*GAMMA+5)))/(2*GAMMA+5))-((pow((vgtQ-vdQ),(GAMMA+3))-
pow(vgtQ,(GAMMA+3)))*pow(vgtQ,(GAMMA+2))/(GAMMA+3)));
                qsource= -qgate - qdrain;
                CggNum=ddx(qgate,V(g)) + (covf);
                vgc=vgt;
                vdseD=vds / (pow((1 + (pow((vds/(ALPHASAT * (vgc+DVL))),M))), (1 / M)));
                //vdseD=vds / (pow((1 + (pow((vds/(ALPHASAT * (vgc+DVL))),3.9))), (1 /
3.9)));
                //m1x=(pow((vds/(ALPHASAT * (vgc+DVL))),4));
                //m2x=(pow((1 + m1x), (1 / 4)));
                //vdseD=vds / (pow((1 + m1x), (1 / 4)));
                k=W*CIF*MUFET/L;
                k1=pow((W*CIF),2.0)*MUFET*1e-4;
                k2=pow((vds/vsate),M);
                //k2=pow((vds/vsate),3.9);
                k2D=pow((vds/(ALPHASAT * (vgc+DVL))),M);
                //k2D=pow((vds/(ALPHASAT * (vgc+DVL))),3.9);
                k3=(1-tanh((vgc-DVL)*QL))/2.0;
                k4=(1+tanh((vgc-DVL)*QL))/2.0;
                expo=exp((2.3*vgc)/SL);

        dencaa1=(k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*k3/(1+RD*k*pow((vgc+
DVL),(GAMMA+1))));
                dencaa2=(k*pow(vgc,(GAMMA+1))*(1
+(LAMBDA*vds))*vdse*k4/(1+RD*k*pow(vgc,(GAMMA+1))))+IOL;
                dencaa= dencaa1 + dencaa2;
```

```
        //dencaa=(k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*k3/(1+RD*k*pow((vgc
+DVL),(GAMMA+1))))+(k*pow(vgc,(GAMMA+1))*(1
        +(LAMBDA*vds))*vdse*k4/(1+RD*k*pow(vgc,(GAMMA+1))))+IOL;
                caa=(-1e4*k1/(GAMMA+3))*((pow((vgc-vdse),(GAMMA+2))*(GAMMA+3)*(1-
                vdse*k2/(vgc*(1+k2))))-(pow(vgc,(GAMMA+2))*(GAMMA+3)));

        ca=k*pow((vgc+DVL),GAMMA)*(GAMMA+1)*(1+(LAMBDA*vds))*vdseD*expo*k3/(1+RD*k*pow((vgc+
DVL),(GAMMA+1)));

        cb=k*pow((vgc+DVL),GAMMA)*(1+(LAMBDA*vds))*vdseD*expo*k3*k2D/((1+RD*k*pow((vgc+DVL),
(GAMMA+1)))*(1+k2D));

        cc1=k*k*pow((vgc+DVL),((2*GAMMA)+1))*(1+(LAMBDA*vds))*vdseD*expo*k3*RD*(GAMMA+1);
                cc2=((1+RD*k*pow((vgc+DVL),(GAMMA+1)))*(1+RD*k*pow((vgc+DVL),(GAMMA+1))));
                cc=cc1/cc2;

        //cc=k*k*pow((vgc+DVL),((2*GAMMA)+1))*(1+(LAMBDA*vds))*vdseD*expo*k3*RD*(GAMMA+1)/((
1+RD*k*pow((vgc+DVL),(GAMMA+1)))*(1+RD*k*pow((vgc                    +DVL),(GAMMA+1))));

        cd=2.3*k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*k3/(SL*(1+RD*k*pow((vg
c+DVL),(GAMMA+1))));
                ce=k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*QL*(1-
tanh(pow(((vgc-
                DVL)*QL),2)))/(2*(1+RD*k*pow((vgc+DVL),(GAMMA+1))));

        cf=k*pow(vgc,GAMMA)*(GAMMA+1)*(1+(LAMBDA*vds))*vdse*k4/(1+RD*k*pow(vgc,(GAMMA+1)));

        cg=k*pow(vgc,GAMMA)*(1+(LAMBDA*vds))*vdse*k2*k4/((1+k2)*(1+RD*k*pow(vgc,(GAMMA+1))))
;

        ch=k*k*pow(vgc,((2*GAMMA)+1))*(1+(LAMBDA*vds))*vdse*k4*RD*(GAMMA+1)/((1+RD*k*pow(vgc
,(GAMMA+1)))*(1+RD*k*pow(vgc,(GAMMA+1))));
                ck=k*pow(vgc,(GAMMA+1))*(1+(LAMBDA*vds))*vdse*QL*(1-tanh(pow(((vgc-
DVL)*QL),2)))/(2*(1+RD*k*pow(vgc,(GAMMA+1))));

        cd1=k*pow((vgc+DVL),(GAMMA+1))*(1+(LAMBDA*vds))*vdseD*expo*k3/(1+RD*k*pow((vgc+DVL),
(GAMMA+1)));

        cd2=k*pow(vgc,(GAMMA+1))*(1+(LAMBDA*vds))*vdse*k4/(1+RD*k*pow(vgc,(GAMMA+1)));
                cgga1=((1e4*k1/(GAMMA+3))*(pow((vgc-vdse),(GAMMA+3))-
pow(vgc,(GAMMA+3)))*(ca+cb-cc+cd-ce+cf+cg+ch+ck));
                cgga2=((caa/dencaa)+cgga1/pow((cd1+cd2+IOL),2.0));
                cgga=cgga2+covf+OFFSET;
//              cgga=((caa/dencaa)+((1e4*k1/(GAMMA+3))*(pow((vgc-vdse),(GAMMA+3))-
pow(vgc,(GAMMA+3)))*(ca+cb-cc+cd-ce+cf+cg+ch+ck))/pow((cd1+cd2+IOL),2.0))+covf+OFFSET;

                cgd=(2.0*(cgga-covf)*(1-pow((vgc/((2*vgc)-vdse)),2))/3.0)+covfd;

                cgs=(2.0*(cgga-covf)*(1-pow(((vgc-vdse)/(2*vgc-vdse)),2))/3.0)+covfs;

                I(s) <+ TYPE * ddt(qsource);
                I(g) <+ TYPE * ddt(qgate);
                I(d) <+ TYPE * ddt(qdrain);


        end

endmodule
```

Anexo E: Netlist de SmartSpice y NGspice para el OTFT tipo P

Con este netlist anexado se hizo la simulación de OTFT tipo P, en SmartSpice y en NGspice, dando seguimiento al diagrama de flujo de cómo se construye un netlist para simulación, cabe mencionar que la diferencia entre SmartSpice y NGspice es la forma de escribir el nombre del dispositivo (Mx y qNx) además del número de nodos que se utilizan.

```
***********TIPO P LEVEL 37 en SMARTSPICE
.title Netlist
.option temp=25.0 reltol=1e-5 abstol=1e-15
.model OTFT PTFT (LEVEL=37
+VACC=1
+LAMBDA=0
+VT=-17.86
+V0=0.948
+GAMMA=0.031
+ASAT=1.19
+MUACC=1.07e-4
+MSAT=3.90
+SIGMA0=8.66e-15
+RS=0
+RD=0
+EPS=2
+EPSI=0.54)
**++CIF=4.83e-9 no lo tiene L37
vd 2 0 DC -60
vg 1 0 DC -40
M1      2     1       0      OTFT    L=200e-6      W=2000e-6
.DC vd 0 -40 -1
* vg -10 -40 -10
.iplot dc I(Vd)
.end
***********Netlist para NGspice
.title Netlist
.option temp=25.0 reltol=1e-5 abstol=1e-15
.model  mybjtmodelp pnp  level=7
+TYPE=-1
+EPS=2
+EPSI=0.54
+TOX=1e-7
+W=2000e-6
+L=200e-6
+VTO=-17.86
+GAMMA=0.031
+VAA=1
+ALPHASAT=1.19
+MU0=1.07e-4
+M=3.90
+LAMBDA=0
+RD=700e3
+RS=400e3

vd 2 0 DC -60
vg 1 0 DC -40

qN1     2     1       0     0      0       mybjtmodelp
.DC vd 0 -40 -1
* vg -10 -40 -10
.plot dc I(Vd)
.op
.end
```

Anexo F: Netlist del oscilador de anillo

Este netlist es para la descripción del oscilador de anillo, solo para NGspice.

```
.title Netlist
.option temp=25.0 reltol=1e-5 abstol=1e-15
.model mybjtmodel1 pnp  level=7
+TYPE=-1
+TOX=1e-7
+L=200e-6
+W=2000e-6
+VAA=1
+LAMBDA=0
+VTO=-17.86
+GAMMA=0.031
+ALPHASAT=1.19
+MU0=1.07e-4
+M=3.90
+RS=0
+RD=0
+EPS=2
+EPSI=0.54
+CIF=4.83e-9

vd 2 0  DC  -60
*       D       G       S       T
qN1      1      2       2       0       0       mybjtmodel1
qN2      1      3       0       0       0       mybjtmodel1
qN3      4      2       2       0       0       mybjtmodel1
qN4      4      1       0       0       0       mybjtmodel1
qN5      5      2       2       0       0       mybjtmodel1
qN6      5      4       0       0       0       mybjtmodel1
qN7      6      2       2       0       0       mybjtmodel1
qN8      6      5       0       0       0       mybjtmodel1
qN9      3      2       2       0       0       mybjtmodel1
qN0      3      6       0       0       0       mybjtmodel1
*R1     2       1       0.9Meg
*R2     2       4       0.9Meg
*R3     2       5       0.9Meg
*R4     2       6       0.9Meg
*R5     2       3       0.9Meg
C1      2       1       45p
C2      1       0       45p
C3      2       4       45p
C4      4       0       45p
C5      2       5       45p
C6      5       0       45p
C7      2       6       45p
C8      6       0       45p
C9      2       3       45p
C10     3       0       45p

.ic V(1)=-60 V(4)=-30 V(5) =-60 V(6) =-30 V(3) =-60
.print tran  V(3)
* V(1) V(4) V(5) V(6)
.TRAN 500us 10ms
.end
```

Anexo G: Respuesta transitoria del oscilador en NGspice

Con este anexo será posible observar la salida gráfica que se obtiene directamente en NGspice, al ejecutar los comandos mediante la terminal.

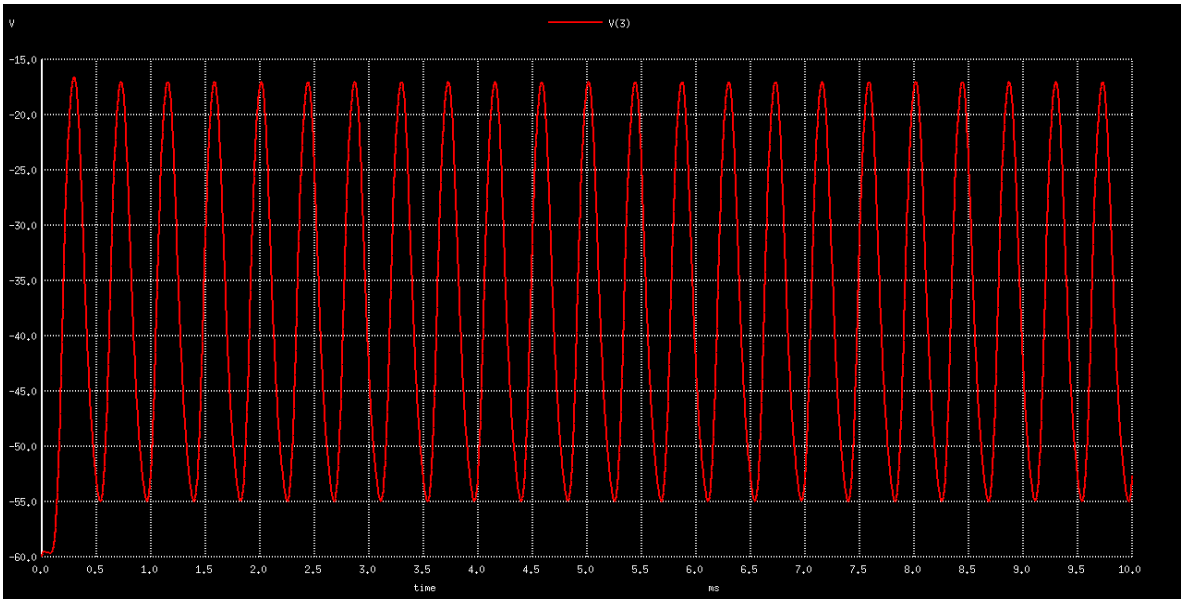


Figura G. Respuesta transitoria del oscilador en un rango de 500us a 10ms

Anexo H: Netlist del amplificador diferencial con salida única

Este netlist da a conocer la forma en que se simuló el circuito analógico, amplificador de entrada diferencial con salida única.

```
.title Netlist

.option temp=25.0 reltol=1e-5 abstol=1e-15

.model mybjtmodel  npn level=7
+TYPE=1
+EPS=6.5
+EPSI=2.6
+TOX=165e-9
+W=100e-6
+L=100e-6
**MODEL PARAMETERS PREVIOUSLY EXTRACTED  **
+CIF=1e-12
+VTO=5.4272
+GAMMA=0.1
+VAA=20100
+ALPHASAT=0.61682
+MU0=6.9e-6
+M=3.4656
+LAMBDA=0
+RD=0
+RS=0
+IOL=1.39e-10
+IOS=5.76e-9
+SS=2.2
+QS=1
+QL=1

.model mybjtmodelp  pnp level=7
+TYPE=-1
+EPS=6.5
+EPSI=2.6
+TOX=165e-9
+W=200e
+L=100e-6

**MODEL PARAMETERS PREVIOUSLY EXTRACTED  **
+CIF=1e-12
+VTO=0.70869
+GAMMA=0.133237
+VAA=0.014047
+ALPHASAT=0.88734
+MU0=4.2e-6
+M=1.5422
+LAMBDA=0.00095852
+RD=0
+RS=0
+IOL=1.39e-10
+IOS=5.76e-9
+SS=2.2
+QS=1
+QL=1
.model mybjtmodelp1  pnp level=7
+TYPE=-1
+EPS=6.5
+EPSI=2.6
+TOX=165e-9
+W=400e-6
+L=100e-6
```

```
**MODEL PARAMETERS PREVIOUSLY EXTRACTED  **
+CIF=1e-12
+VTO=0.70869
+GAMMA=0.133237
+VAA=0.014047
+ALPHASAT=0.88734
+MU0=4.2e-6
+M=1.5422
+LAMBDA=0.00095852
+RD=0
+RS=0
+IOL=1.39e-10
+IOS=5.76e-9
+SS=2.2
+QS=1
+QL=1
.model mybjtmodelp2  pnp level=7
+TYPE=-1
+EPS=6.5
+EPSI=2.6
+TOX=165e-9
+W=800e-6
+L=100e-6
**MODEL PARAMETERS PREVIOUSLY EXTRACTED  **
+CIF=1e-12
+VTO=0.70869
+GAMMA=0.133237
+VAA=0.014047
+ALPHASAT=0.88734
+MU0=4.2e-6
+M=1.5422
+LAMBDA=0.00095852
+RD=0
+RS=0
+IOL=1.39e-10
+IOS=5.76e-9
+SS=2.2
+QS=1
+QL=1

**Fuentes de Voltaje**
vd   7       0   DC   20
vp      4       0       dc      10
vn      3       0       DC   14  AC  1  sin(141      100)
*vn      3   0          DC       10

**Definicion del circuito**
qN1             2               1               0               0       0       mybjtmodel
qN2             1               1               0               0       0       mybjtmodel
qN3             2               3               5               0       0       mybjtmodelp
qN4             1               4               5               0       0       mybjtmodelp
qN5             6               6               7               0       0       mybjtmodelp1
qN6             5               6               7               0       0       mybjtmodelp2

R3              6       0       18MEG
*Rp             7       0       10G
C2              2       0       1p

*.dc    vn      0       20      0.4
*.plot  dc   I(R3)
*.plot dc V(1) V(5) V(6) V(2) V(3)
**.tran   100us       100ms
**.plot tran V(2) V(3)
.ac dec  20    1   10k
.plot ac V(2) V(3)
.end
```

REFERENCIAS

[1] Quarles, T., Pederson, D., Newton, R., Sangiovanni-Vincentelli, A. and Wayne, C. *The Spice Home Page*. [online] Bwrcs.eecs.berkeley.edu. Available at: http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/.

[2] Estrada, M., Mejía, I., Cerdeira, A., Pallares, J., Marsal, L. and Iñiguez, B. (2008). Mobility model for compact device modeling of OTFTS made with different materials. *Solid-State Electronics*, 52(5), pp.787-794.

[3] Castro-Carranza, [. (2012). *International Conference on Organic Electronics*.

[4] Zschieschang, U., Kang, M., Takimiya, K., Sekitani, T., Someya, T., Canzler, T., Werner, A., Blochwitz-Nimoth, J. and Klauk, H. (2012). Flexible low-voltage organic thin-film transistors and circuits based on C 10 -DNTT. *J. Mater. Chem.*, 22(10), pp.4273-4277.

[5] Wong, W. and Salleo, A. (2008). *Flexible electronics*. New York: Springer.

[6] Göllner, M. (2011). *Double-Gate pentacene thin film transistor for biosensing*.

[7] Zafar Q, Z Ahmad, Sulaiman K. PFO-DBT: MEH-PPV: PC 71 BM mezcla ternaria Assisted Plataforma como un fotodetector. *Sensores* . 2015; 15 (1): 965-978.

[8] Estrada, M., Mejia, I., Cerdeira, A. and Iñiguez, B. (2008). MIS polymeric structures and OTFTS using PMMA on P3HT layers. *Solid-State Electronics*, 52(1), pp.53-59.

[9] O'Dwyer, J. (1973). *The theory of electrical conduction and breakdown in solid dielectrics*. Oxford: Clarendon Press.

[10] Inzelt, G., Pineri, M., Schultze, J. and Vorotyntsev, M. (2000). Electron and proton conducting polymers: recent developments and prospects. *Electrochimica Acta*, 45(15-16), pp.2403-2421.

[11] Chiang, C., Fincher, C., Park, Y., Heeger, A., Shirakawa, H., Louis, E., Gau, S. and MacDiarmid, A. (1977). Electrical Conductivity in Doped Polyacetylene. *Phys. Rev. Lett.*, 39(17), pp.1098-1101.

[12] Shirakawa, H. (1996). Commentary: Reflections on “Simultaneous polymerization and formation of polyacetylene film on the surface of concentrated soluble ziegler-type catalyst solution,” by Takeo Ito, Hideki Shirakawa, and Sakuji

Ikeda,J. Polym. Sci.: Polym. Chem. Ed., 12, 11 (1974). *J. Polym. Sci. A Polym. Chem.*, 34(13), pp.2529-2530.

[13] Letheby, H. (1862). XXIX.—On the production of a blue substance by the electrolysis of sulphate of aniline. *J. Chem. Soc.*, 15(0), pp.161-163.

[14] Chiang, C., Fincher, C., Park, Y., Heeger, A., Shirakawa, H., Louis, E., Gau, S. and MacDiarmid, A. (1977). Electrical Conductivity in Doped Polyacetylene. *Phys. Rev. Lett.*, 39(17), pp.1098-1101.

[15] Chiang, C., Druy, M., Gau, S., Heeger, A., Louis, E., MacDiarmid, A., Park, Y. and Shirakawa, H. (1978). Synthesis of highly conducting films of derivatives of polyacetylene, (CH)x. *J. Am. Chem. Soc.*, 100(3), pp.1013-1015.

[16] Ebisawa, F., Kurokawa, T. and Nara, S. (1983). Electrical properties of polyacetylene/polysiloxane interface. *J. Appl. Phys.*, 54(6), p.3255.

[17] Tang, C. and VanSlyke, S. (1987). Organic electroluminescent diodes. *Appl. Phys. Lett.*, 51(12), p.913.

[18] Koezuka, H., Tsumura, A. and Ando, T. (1987). Field-effect transistor with polythiophene thin film. *Synthetic Metals*, 18(1-3), pp.699-704.

[19] Burroughes, J., Jones, C. and Friend, R. (1988). New semiconductor device physics in polymer diodes and transistors. *Nature*, 335(6186), pp.137-141.

[20] Horowitz, G., Fichou, D., Peng, X., Xu, Z. and Garnier, F. (1989). A field-effect transistor based on conjugated alpha-sexithienyl. *Solid State Communications*, 72(4), pp.381-384.

[21] Madru, R., Guillaud, G., Al Sadoun, M., Maitrot, M., André, J., Simon, J. and Even, R. (1988). A well-behaved field effect transistor based on an intrinsic molecular semiconductor. *Chemical Physics Letters*, 145(4), pp.343-346.

[22] Clarisse, C., Riou, M., Gauneau, M. and le Contellec, M. (1988). Field-effect transistor with diphthalocyanine thin film. *Electronics Letters*, 24(11), pp.674-675.

[23] Horowitz, G., Peng, X., Fichou, D. and Garnier, F. (1992). Role of the semiconductor/insulator interface in the characteristics of π-conjugated-oligomer-based thin-film transistors. *Synthetic Metals*, 51(1-3), pp.419-424.

[24] Gross, L., Mohn, F., Moll, N., Liljeroth, P. and Meyer, G. (2009). The Chemical Structure of a Molecule Resolved by Atomic Force Microscopy. *Science*, 325(5944), pp.1110-1114.

[25] Panzer, M., Newman, C. and Frisbie, C. (2005). Low-voltage operation of a pentacene field-effect transistor with a polymer electrolyte gate dielectric. *Appl. Phys. Lett.*, 86(10), p.103503.

[26] Clemens J. Först, C., R. Ashman, C., Schwarz, K. and E. Blöchl, P. (2003). *The interface between silicon and a high-k oxide*. [online] Info.tuwien.ac.at. Available at: http://info.tuwien.ac.at/theochem/si-srtio3_interface/si-srtio3.html [Accessed 30 May 2016].

[27]Wikipedia. (2015). *Organic field-effect transistor*. [online] Available at: https://en.wikipedia.org/wiki/Organic_field-effect_transistor.

[28] Brody, T. (1984). The thin film transistor—A late flowering bloom. *IEEE Trans. Electron Devices*, 31(11), pp.1614-1628.

[29] Haddon, R., Perel, A., Morris, R., Palstra, T., Hebard, A. and Fleming, R. (1995). C60 thin film transistors. *Appl. Phys. Lett.*, 67(1), p.121.

[30] Katz, H. (1997). Organic molecular solids as thin film transistor semiconductors. *Journal of Materials Chemistry*, 7(3), pp.369-376.

[31] Kelley TW, Muyres DV, Baude PF, Smith TP, Jones TD (2003) High performance organic thin film transistors. In: Organic and Polymeric Materials and Devices. Symposium (Mater Res Soc Symposium Proceedings vol 771),169.Lin, Y.-Y., et al., IEEE Trans. Electron Dev. (1997) 44 (8), 1325.

[32] Sung Kyu Park, Yong Hoon Kim, Jeong In Han, Dae Gyu Moon, and Won Keun Kim, (2002). High-performance polymer tfts printed on a plastic substrate. *IEEE Trans. Electron Devices*, 49(11), pp.2008-2015.

[33] Babel, A. and Jenekhe, S. (2002). Electron Transport in Thin-Film Transistors from an n-Type Conjugated Polymer. *Adv. Mater.*, 14(5), p.371.

[34] Bao, Z., Lovinger, A. and Brown, J. (1998). New Air-Stable n -Channel Organic Thin Film Transistors. *J. Am. Chem. Soc.*, 120(1), p.207.

[35] Kelley TW, Muyres DV, Baude PF, Smith TP, Jones TD (2003) High performance organic thin film transistors, Mater Res Soc Proc 771,169.

[36] Sundar, V. (2004). Elastomeric Transistor Stamps: Reversible Probing of Charge Transport in Organic Crystals. *Science*, 303, pp.1644.

[37] Kagan, C. and Andry, P. (2003). *Thin-film transistors*. New York: Marcel Dekker.

[38] Bao, Z., Dodabalapur, A. and Lovinger, A. (1996). Soluble and processable regioregular poly(3-hexylthiophene) for thin film field-effect transistor applications with high mobility. *Appl. Phys. Lett.*, 69, pp.4108-10.

[39] Estrada, M., Ulloa, F., Avila, M., Sanchez, J., Cerdeira, A., Castro-Carranza, A., Iniguez, B., Marsal, L. and Pallares, J. (2013). Frequency and Voltage Dependence of the Capacitance of MIS Structures Fabricated With Polymeric Materials. *IEEE Trans. Electron Devices*, 60(6), pp.2057-2063.

[40] Estrada, M., Cerdeira, A., Mejia, I., Avila, M., Picos, R., Marsal, L., Pallares, J. and Iñiguez, B. (2010). Modeling the behavior of charge carrier mobility with temperature in thin-film polymeric transistors. *Microelectronic Engineering*, 87(12), pp.2565-2570.

[41] Mejia, I., Estrada, M. and Avila, M. (2008). Improved upper contacts PMMA on P3HT PTFTS using photolithographic processes. *Microelectronics Reliability*, 48(11-12), pp.1795-1799.

[42] Estrada, M., Ulloa, F., Avila, M., Sanchez, J., Cerdeira, A., Castro-Carranza, A., Iniguez, B., Marsal, L. and Pallares, J. (2013). Frequency and Voltage Dependence of the Capacitance of MIS Structures Fabricated With Polymeric Materials. *IEEE Trans. Electron Devices*, 60(6), pp.2057-2063.

[43] Papadopoulos, N., Marsal, A., Picos, R., Puigdollers, J. and Hatzopoulos, A. (2012). Simulation of organic inverter. *Solid-State Electronics*, 68, pp.18-21.

[44] Castro-Carranza, A., Estrada, M., Cerdeira, A., Nolasco, J., Sanchez, J., Marsal, L., Iniguez, B. and Pallares, J. (2014). Compact Capacitance Model for OTFTS at Low and Medium Frequencies. *IEEE Trans. Electron Devices*, 61(2), pp.638-642.

[45] Tsumura, A., Koezuka, H. and Ando, T. (1986). Macromolecular electronic device: Field-effect transistor with a polythiophene thin film. *Appl. Phys. Lett.*, 49(18), p.1210.

[46] Brotherton, S. (2013). *Introduction to thin film transistors*. Cham: Springer.

[47] Shur, M. and Hack, M. (1984). Physics of amorphous silicon based alloy field-effect transistors. *J. Appl. Phys.*, 55(10), p.3831.

[48] Ortiz-Conde, A., García Sánchez, F., Liou, J., Cerdeira, A., Estrada, M. and Yue, Y. (2002). A review of recent MOSFET threshold voltage extraction methods. *Microelectronics Reliability*, 42(4-5), pp.583-596.

[49] Boudinet, D., Le Blevennec, G., Serbutoviez, C., Verilhac, J., Yan, H. and Horowitz, G. (2009). Contact resistance and threshold voltage extraction in n-

channel organic thin film transistors on plastic substrates. *J. Appl. Phys.*, 105(8), p.084510.

[50] Ryu, K., Kymissis, I., Bulovic, V. and Sodini, C. (2005). Direct extraction of mobility in pentacene OFETs using C-V and I-V measurements. *IEEE Electron Device Lett.*, 26(10), pp.716-718.

[51] Natali, D., Fumagalli, L. and Sampietro, M. (2007). Modeling of organic thin film transistors: Effect of contact resistances. *J. Appl. Phys.*, 101(1), p.014501.

[52] Altazin, S., Clerc, R., Gwoziecki, R., Boudinet, D., Ghibaudo, G., Pananakakis, G., Chartier, I. and Coppard, R. (2011). Analytical modeling of the

[53] contact resistance in top gate/bottom contacts organic thin film transistors. *Organic Electronics*, 12(6), pp.897-902.

[54] Deen, M., Kazemeini, M. and Holdcroft, S. (2008). Contact effects and extraction of intrinsic parameters in poly(3-alkylthiophene) thin film field-effect transistors. *J. Appl. Phys.*, 103(12), p.124509.

[55] Marinov, O., Deen, M., Zschieschang, U. and Klauk, H. (2009). Organic Thin-Film Transistors: Part I-Compact DC Modeling. *IEEE Trans. Electron Devices*, 56(12), pp.2952-2968.

[56] Lara Bullejos, P., Jiménez Tejada, J., Rodríguez-Bolívar, S., Deen, M. and Marinov, O. (2009). Model for the injection of charge through the contacts of organic transistors. *J. Appl. Phys.*, 105, p.084516.

[57] Jiménez Tejada, J., Awawdeh, K., Villanueva, J., Carceller, J., Deen, M., Chaure, N., Basova, T. and Ray, A. (2011). Contact effects in compact models of organic thin film transistors: Application to zinc phthalocyanine-based transistors. *Organic Electronics*, 12(5), pp.832-842.

[58] Cerdeira, A., Estrada, M., Iniguez, B. and Soto, S. (2011). Modeling the subthreshold region of OTFTS. *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control*.

[59] Cerdeira, A., Estrada, M.,García, R., Ortiz-Conde, A. and García Sánchez, F. (2001). New procedure for the extraction of basic a-Si:H TFT model parameters in the linear and saturation regions. *Solid-State Electronics*, 45(7), pp.1077-1080.

[60] Cerdeira, A., Estrada, M., Iñiguez, B., Pallares, J. and Marsal, L. (2004). Modeling and parameter extraction procedure for nanocrystalline TFTs. *Solid-State Electronics*, 48(1), pp.103-109.

[61] Estrada, M., Cerdeira, A., Resendiz, L., García, R., Iñiguez, B., Marsal, L. and Pallares, J. (2006). Amorphous silicon carbide TFTs. *Solid-State Electronics*, 50(3), pp.460-467.

[62] Cerdeira, A., Estrada, M., Soto-Cruz, B. and Iñiguez, B. (2012). Modeling the behavior of amorphous oxide thin film transistors before and after bias stress. *Microelectronics Reliability*, 52(11), pp.2532-2536.

[63] Mejia, I., Salas-Villasenor, A., Avendano-Bolivar, A., Gnade, B. and Quevedo-Lopez, M. (2012). Modeling and SPICE simulation of CdS-pentacene hybrid CMOS TFTs. *2012 8th International Caribbean Conference on Devices, Circuits and Systems (ICCDCS)*.

[64] Fjeldly, T., Ytterdal, T. and Shur, M. (1998). *Introduction to device modeling and circuit simulation*. New York: Wiley.

[65] Castro-Carranza, A., Estrada, M., Nolasco, J., Cerdeira, A., Marsal, L., Iñíguez, B. and Pallarès, J. (2012). Organic thin-film transistor bias-dependent capacitance compact model in accumulation regime. *IET Circuits, Devices & Systems*, 6(2), p.130-135.

[66] Simulación del señales mixtas
SPICE Models Manual. (2015). Santa Clara, CA 95054: Silvaco, Inc.

[67] Taylor, D., Patchett, E., Williams, A., Ding, Z., Assender, H., Morrison, J. and Yeates, S. (2015). Fabrication and simulation of organic transistors and functional circuits. *Chemical Physics*, 456, pp.85-92.

[68] Baker, R., Li, H. and Boyce, D. (1998). *CMOS circuit design, layout, and simulation*. New York: IEEE Press.

[69] Castro Carranza, A. (2013). *UMEM-BASED CAPACITANCE MODEL FOR ORGANIC FIELD EFFECT TRANSISTORS: DEVELOPMENT AND IMPLEMENTATION*. Doctorado. Universitat Rovira I Virgili.