

**MODELADO DE CIRCUITOS INTEGRADOS
ANALÓGICOS DESCRITOS BAJO EL ESTÁNDAR
VERILOG-A/MS.**

EDWIN DAVID VILLAMIZAR RIVERA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA

2014

**MODELADO DE CIRCUITOS INTEGRADOS
ANALÓGICOS DESCRITOS BAJO EL ESTÁNDAR
VERILOG-A/MS.**

EDWIN DAVID VILLAMIZAR RIVERA

Trabajo de grado para optar por el título de Ingeniero Electrónico

Director

LUIS CARLOS GUTIÉRREZ LÁZARO

Ingeniero Electrónico, MSc

Codirector

JAIME GUILLERMO BARRERO PÉREZ

Ingeniero Electrónico, MSc

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA**

2014

*“ A mis ejemplares padres Josefina y José David,
por su ilimitado apoyo
y ser un modelo positivo de vida.”*

Agradecimientos

Quiero agradecer a mis padres por su amoroso apoyo durante todo este proceso educativo y por brindarme la oportunidad de proyectarme académicamente. Gracias a ellos y sus inagotables consejos de vida soy lo que soy hoy en día. A *lili* por su compañía durante nuestra niñez y juventud.

A mi director de proyecto Luis Carlos *lucho*, por su total y desinteresado apoyo además de su motivación hacia el modelado de circuitos integrados analógicos.

Al profesor Jaime Barrero que junto con los *2050's*, fomentaron las condiciones para que el grupo CIDIC continuara.

Agradezco a *Vane*, por todo su amor.

RESUMEN

TÍTULO:

MODELADO DE CIRCUITOS INTEGRADOS ANALÓGICOS DESCRITOS BAJO EL ESTÁNDAR VERILOG-A/MS ¹

AUTOR: EDWIN DAVID VILLAMIZAR RIVERA²

PALABRAS CLAVE: Modelado, Verilog-A/MS, Op-amp, Fuente de Referencia.

DESCRIPCIÓN:

En los últimos años el modelado de circuitos integrados analógicos se ha perfilado como un área de crecimiento, debido a la necesidad de librerías que agilicen el proceso de diseño de *IC*, *SOCs* y en general de cualquier dispositivo electrónico que utilice circuitos analógicos. Por esta razón, es importante para la Universidad Industrial de Santander y específicamente para el grupo CIDIC incursionar en esta área.

Este proyecto utiliza el lenguaje de descripción de *hardware* analógico y de señal mezclada *Verilog-A/MS* para la creación de modelos fundamentales de circuitos integrados analógicos; un *Op – amp* y una fuente de referencia de voltaje. Con estos modelos se pretende influir en la selección de especificaciones y arquitecturas. Inicialmente se realiza un acercamiento a los *HDLs* analógicos y de señal mixta mas relevantes en la industria (*Matlab*, *SystemC-AMS*, *VHDL-AMS* y *Verilog-A/MS*), la descripción de sus beneficios y estructura básica. Posteriormente se presenta un análisis del comportamiento de los circuitos seleccionados, resaltando las ventajas de cada uno para la creación de circuitos integrados analógicos. Por ultimo se presenta el modelado detallado de cada uno de los circuitos y un análisis de resultados que justifica su selección para el desarrollo de este trabajo. Como parte de la verificación del funcionamiento, los modelos desarrollados se implementan en un circuito de administración de potencia.

Adicionalmente se crea un material de apoyo donde se presenta de manera mas amplia la explicación, creación y simulación de un modelo descrito mediante *Verilog-A/MS*.

¹Proyecto de Grado

²Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director MSc. Luis Carlos Gutiérrez Lázaro. Codirector MSc. Jaime Guillermo Barrero Pérez.

ABSTRACT

TITLE:

MODELING OF ANALOG INTEGRATED CIRCUITS WITH VERILOG-A/MS³

AUTHOR: EDWIN DAVID VILLAMIZAR RIVERA ⁴

KEYWORDS: Modeling, Verilog-A/MS, Op-amp, Voltage Reference.

DESCRIPTION:

In recent years the modeling of analog integrated circuits has emerged as a growth area, due to the need for libraries to speed the process of IC design, SOCs and in general, any electronic device that uses analog circuits. For this reason, it is important for the Industrial University of Santander and specifically for CIDIC group to deepen into this area.

This project uses the hardware description language for analog and mixed-signal Verilog-A/MS for creating basic models of analog integrated circuits; Op-amp and a voltage reference. These models are intended to influence the selection of specifications and architectures. First of all, this book presents an approach to analog HDLs and the most relevant mixed-signal languages (Matlab, SystemC-AMS, VHDL-AMS and Verilog-A/MS), second, it presents a description of its benefits and basic structure. Subsequently, an analysis of the behavior of selected circuits, highlighting the advantages of each to the creation of analog integrated circuits. Finally detailed modeling of each of the circuits and analysis results justified his selection for the development of this work is presented. As part of the verification of the operation, the models developed are implemented in a power management circuit.

Additionally, a support material is presented with more broadly explanation of building and simulating of a model described by Verilog-A/MS.

³Degree project

⁴Physical-Mechanical Engineering Faculty. Electrical, Electronics and Telecommunications School. Advisor MSc. Luis Carlos Gutiérrez Lázaro. Co Advisor MSc. Jaime Guillermo Barrero Pérez.

Contenido

INTRODUCCIÓN	14
1. MODELADO DE ICs	19
1.1. APLICACIONES DE <i>VERILOG-A/MS</i>	20
1.1.1. Modelado de Componentes	20
1.1.2. Bancos de Pruebas	20
1.1.3. Velocidad de Simulación	20
1.1.4. Diseño de Señal Mixta	21
1.1.5. <i>Top-Down Design</i>	21
1.2. SINTAXIS DE <i>VERILOG-A/MS</i>	21
1.2.1. <i>Lexical Tokens</i>	22
1.2.2. Tipos de Datos	24
1.2.3. Expresiones	25
1.2.4. Señales	25
1.2.5. Comportamiento analógico	25
1.2.6. Jerarquía	26
2. MODELOS	27
2.1. FUENTE DE VOLTAJE DE REFERENCIA	27
2.1.1. Parámetros	28
2.1.2. Regulación de Línea	28
2.1.3. Orden de Compensación	29
2.1.4. Ruido	30
2.2. AMPLIFICADOR OPERACIONAL	31
2.2.1. Parámetros	31
2.2.2. Modelado del Producto Ganancia-Ancho de Banda <i>GBW</i>	31
2.2.3. CMRR y PSRR	33
2.2.4. Modelado de <i>BIAS</i> y <i>offset</i>	33
2.2.5. Modelado del Ruido	35
2.2.6. <i>Slew Rate</i>	37
2.2.7. Clipping	37
3. ANÁLISIS DE RESULTADOS	39
3.1. FUENTE DE REFERENCIA DE TENSIÓN	39
3.1.1. Ruido	41
3.2. AMPLIFICADOR OPERACIONAL	43
3.2.1. Ruido	51

3.3. VALIDACIÓN MEDIANTE LA APLICACIÓN EN UN SISTEMA DE POTENCIA	52
4. OBSERVACIONES Y CONCLUSIONES	56
4.1. CONCLUSIONES	56
4.2. OSBERVACIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS	57
Bibliografía	59
A. Código de los Modelos	61

Lista de Figuras

1.	Metodología de Desarrollo Secuencial Estándar Utilizada en CI [3].	16
2.	Relación entre <i>Verilog-AMS</i> , <i>Verilog-A</i> y <i>Verilog-HDL</i> [1].	17
1.1.	Amplificador diferencial.	22
1.2.	Código Par Diferencial.	23
2.1.	Tipos de Compensación en el Modelo de la Fuente de Referencia.	30
2.2.	Ruido generado por el parámetro <i>npp</i>	31
2.3.	Modelado de <i>GBW</i>	33
2.4.	CMRR y PSRR.	34
2.5.	Modelo de Ruido [22].	35
2.6.	Comportamiento del Ruido.	36
2.7.	Comportamiento de la Función <i>slew()</i>	37
2.8.	Modelado de la Saturación.	38
3.1.	Alimentación VS Salida.	40
3.2.	Variación del Coeficiente de Temperatura. <i>comp</i> = 1.	41
3.3.	Variación del Voltaje en Relación con la Temperatura. <i>comp</i> = 2.	41
3.4.	Ruido en Fuente de Referencia.	42
3.5.	Variación de la Ganancia Para un <i>GBW</i> de 10 MHz.	44
3.6.	Variación de la Ganancia Para un <i>GBW</i> de 100 kHz.	44
3.7.	Valores de Atenuación Para un <i>GBW</i> de 10 MHz.	45
3.8.	Valores de Atenuación Para un <i>GBW</i> de 100 kHz.	45
3.9.	Valores de PSRR Para un <i>fpsr1</i> 1 kHz.	46
3.10.	Valores de PSRR Para un <i>fpsr1</i> 100 kHz.	46
3.11.	Ganancia de 20 dB en el tiempo, <i>offset</i> de 100 m[V] a frecuencia <i>in-band</i>	47
3.12.	Señal con Rechazo 60dB a 100 kHz. Frecuencia <i>in-band</i>	47
3.13.	Salida con Rechazo 60dB a 100 kHz. Frecuencia <i>in-band</i>	48
3.14.	PSRR 60dB, Señales a 100 kHz. Frecuencia <i>in-band</i>	48
3.15.	Modelado de la Distorsión por <i>Clipping</i>	49
3.16.	Modelado de la Distorsión por <i>Slew Rate</i>	49
3.17.	Configuración Ganancia Unitaria.	50
3.18.	Análisis de la Variación de Ganancias en Configuración <i>Buffer</i>	50
3.19.	Análisis de Estabilidad Por Variación de Margen de Fase en Configuración <i>Buffer</i>	51
3.20.	Variación del Parámetro <i>Vrms</i>	52
3.21.	Variación del Piso Térmico.	52
3.22.	Regulador LDO [23].	53
3.23.	Regulación de Línea.	54

3.24. Regulación de Carga.	55
3.25. Salida del Regulador.	55
3.26. Salida del Regulador.	55

Lista de Tablas

1.1. Operadores.	24
2.1. Parámetros y Rango de Valores.	28
2.2. Parámetros y Rango de Valores.	32
3.1. Valores de los Parámetros.	40
3.2. Valores de los Parámetros.	40
3.3. Valores de los Parámetros para Ruido.	42
3.4. Variaciones GBW para Ganancias de 40 dB y 120 dB.	43
3.5. Variaciones Margen de Fase para GBW de 100 kHz y 10MHz	43
3.6. Especificaciones de Diseño	54
3.7. Parámetros de Diseño para los Bloques Modelados.	54

INTRODUCCIÓN

El dinamismo de la industria de semiconductores, que produce anualmente millones de dólares, incide notablemente en las necesidades y el comportamiento de las personas; además, si se tiene en cuenta su evolución en la última década, es evidente su impacto en los hábitos de consumo. Consecuentemente, las empresas de diseño de circuitos integrados en su afán de mantenerse competitivas y a la vanguardia han tenido que adaptarse a una dinámica impulsada por ellas mismas, pero controlada por factores externos tales como mercado, entendido como la demanda de dispositivos de alto desempeño, portabilidad y bajo costo, y se han visto obligadas a desarrollar procesos que agilicen la fabricación de nuevos productos. Las tecnologías de administración de potencia e innovadores flujos de diseño son un claro ejemplo del progreso que exige un mercado globalizado que demanda alto desempeño.

El tiempo de diseño de nuevos sistemas analógicos y de señal mixta depende, además de su complejidad específica, de la metodología usada para su creación. Detenerse demasiado en alguna de las fases que lo componen, represa el desarrollo del producto y dilata su llegada al mercado (lo que va claramente en contra del concepto de *TTM*⁵). Por tanto, el compromiso entre tiempo y desempeño se convierte en el *trade-off* más importante para analizar.

Por consiguiente, los altos niveles de competitividad y la presión que genera la puntualidad de la entrega de un diseño, puede generar errores. A partir de esto surge la idea de realizar un proceso de diseño sistemático, incluyendo herramientas de alto nivel que permitan ahorros de tiempo durante el transcurso del mismo.

En la búsqueda de sistematización, el diseño de sistemas digitales integrados ha evolucionado con tal rapidez que ahora es posible encontrar dispositivos con millones de transistores gracias a herramientas de síntesis lógica, dando así solución a los problemas de automatización en el diseño. Por el contrario el diseño de circuitos integrados analógicos o de señal mixta no cuenta con una automatización como la de su contraparte digital, por lo que no es posible salvo algunos casos concretos realizar síntesis de circuitos analógicos satisfactoria [1]. En consecuencia, las casas de diseño y compañías de semiconductores requieren la creación de librerías con modelos parametrizables de circuitos integrados de señal mixta que permitan elegir arquitecturas y determinar las especificaciones del diseño.

Para abordar los problemas de diseño, empresas especializadas en el desarrollo de herramientas *CAD*⁶, proveen el software necesario para el modelado de alto nivel de circuitos integrados analógicos y digitales, mediante lenguajes de descripción de hardware (*HDL*⁷), los cuales han facilitado por años el diseño de sistemas complejos. Sin embargo, fueron necesarios años de desarrollo para lograr utilidad, divulgación y extensión desde la industria hasta la

⁵Siglas en inglés: Time To Market (Tiempo que toma un producto, desde su concepción hasta su disponibilidad en el mercado).

⁶Siglas en inglés: Computer Aided Design.

⁷Siglas en inglés: Hardware Description Language.

academia.

Debido a esto, ya en la década de los *90s* del siglo XX era posible realizar la descripción formal de circuitería o elementos electrónicos físicos de un circuito integrado. Finalmente surgieron dos lenguajes que encabezan en la actualidad la descripción de hardware *VHDL* y *Verilog*, ambos son estándares de la industria pero se diferencian por su sintaxis.

Por una parte, *VHDL* conserva una sintaxis estructurada, observándose claramente la inclusión de librerías, la definición de la entidad (entradas y salidas del módulo descrito), y la arquitectura. En consecuencia, la percepción del mismo es de descripción y no de programación. Por ende resulta pedagógico y didáctico que en la Universidad Industrial de Santander se trabaje con este tipo de lenguaje, lo cual se evidencia en la capacitación a sus estudiantes en las diferentes cátedras impartidas. Por otro lado, *Verilog* cuenta con una amplia aceptación y soporte mundial, así pues una motivación adicional es la de incursionar en este lenguaje de descripción y aportar a la academia regional y nacional documentación suficiente sobre el empleo de los lenguajes de descripción de hardware enfocados hacia la simulación de sistemas analógicos y no solo digitales. *Verilog-A* [2], hace parte de la familia de lenguajes utilizados para este tipo de casos, siendo *Verilog* utilizado en el dominio digital y *Verilog-AMS* en el campo analógico y de señal mixta.

Existen otras herramientas que no caben dentro de la definición de *HDLs*. Sin embargo, están facultadas para modelar circuitos integrados analógicos y de señal mixta a nivel sistema. Algunos como por ejemplo *SystemC-AMS* están diseñados específicamente para este propósito pero como desventaja tiene la falta de integración con las herramientas *CAD*. *MATLAB* por su lado, pese a ser un lenguaje de alto nivel y permitir modelado de *ICs*, no está diseñado con esta intención.

Respecto al ciclo de desarrollo de circuitos de señal mixta, es posible definir una metodología de diseño específica. Una visión general de cada uno de los pasos de esta metodología se muestra en la figura 1. En vista de la linealidad en su dependencia el diseñador es obligado a mirar atrás para corregir errores. Así pues, los miembros del grupo CIDIC egresados de la Universidad Industrial de Santander que se encuentran trabajando en casas de diseño cuentan con la experiencia en el área para concluir que sistemas bien modelados permiten determinar especificaciones y arquitecturas que claramente son los puntos neurálgicos y así evitar caer en los bucles de realimentación iterativos.

En este trabajo se propone aprovechar las ventajas de la descripción de hardware, para la creación de modelos de circuitos integrados analógicos que aporten velocidad de simulación y permitan definir de manera precisa especificaciones y seleccionar arquitecturas de sistemas más complejos, al igual que contribuir a la formación de una librería de alcances académicos e industriales de sistemas modelados en alto nivel, bajo el estándar *Verilog-A/MS*, siendo esta la primera vez en ser utilizado en la Universidad Industrial de Santander para un proyecto de grado, lo que promoverá su difusión y fomentará su uso en el entorno académico. Adicionalmente y como motivación, se procura seguir contribuyendo a la formación académica en Colombia, en el diseño de circuitos integrados demandados actualmente por el mercado.

Lenguajes de Descripción de *HARDWARE*

En ingeniería, los lenguajes de descripción de hardware se refieren al grupo ordenado de sintaxis, instrucciones, estructuras y órdenes textuales que sirven para realizar la descripción de un sistema. Estos lenguajes permiten detallar el funcionamiento de los sistemas con tal alto grado de abstracción, que por fortuna no resulta estrictamente necesario especificar el comportamiento de componentes de menor nivel. Fueron creados en la segunda mitad del

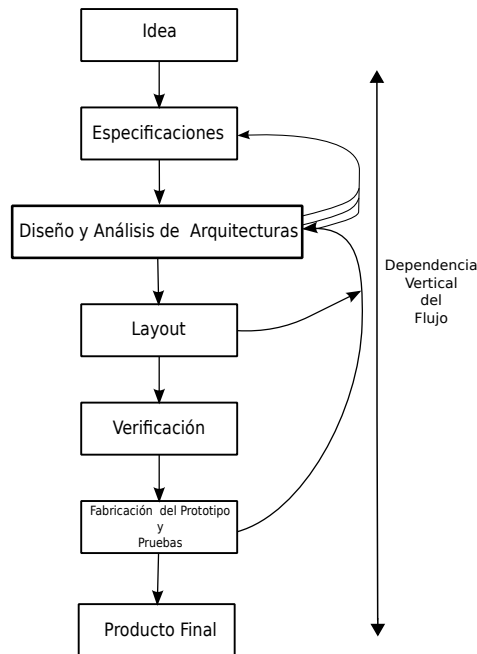


Figura 1: Metodología de Desarrollo Secuencial Estándar Utilizada en CI [3].

siglo XX con el propósito de evitar simulaciones a nivel transistor; época en que la escala de integración de ICs resultaría decisiva para su desarrollo como industria. Además, fueron adoptados por la academia ya que poseen similitudes con los lenguajes de programación pese a que son dos cosas distintas.

Dentro de los múltiples usos de los lenguajes de descripción de hardware, se destacan dos aplicaciones: simulación y síntesis. El primero se refiere a los modelos que predicen el comportamiento de un sistema y el segundo al proceso por el cual se realiza una descripción física con un nivel mayor de abstracción que resulta en una implementación en hardware. En la actualidad existen dos *HDLs* líderes a nivel mundial en la simulación y síntesis de sistemas, los dos son estándares libres y están disponibles para modelar hardware.

Ahora, es necesario aclarar que hasta el momento en esta sección se ha hecho referencia solo al ámbito digital, pero, ¿qué sucedería si se quisiera realizar una descripción, simulación, prueba o análisis de hardware que contenga dispositivos o circuitos analógicos?, o por ejemplo ¿la simulación de un sistema que contenga subsistemas analógicos y digitales?. En efecto, si recurriéramos a las herramientas que se han mencionado resultaría imposible hacerlo. En vista de esto se desarrolló un conjunto anexo de sintaxis para formar lenguajes de descripción de hardware de sistemas analógicos y de señal mixta. *Verilog-AMS* y *VHDL-AMS*.

El presente capítulo introduce los conceptos generales sobre los lenguajes de descripción de hardware analógicos y de señal mixta; de entrada se presenta *VHDL-A/MS* y en segundo lugar *Verilog-A/MS*.

VHDL-AMS

VHDL, aparece por la necesidad de simular sistemas digitales y verificar su funcionamiento antes de programarlos o fabricarlos en físico, su uso se extiende también como síntesis lógica

que permite optimizar los diseños creados por el ingeniero, en otras palabras reduce los errores humanos al simplificar bocetos de proyectos automatizando el diseño, ya que resulta humanamente imposible revisar, desarrollar y verificar sistemas complejos que involucren millones de transistores.

VHDL-AMS es una extensión del estándar *VHDL*, diseñado para el modelado de sistemas digitales, analógicos y de señal mixta. Fue creado para abarcar las diferentes necesidades en el proceso de diseño de sistemas [4]. Primero que todo posibilita la descripción completa de la estructura de un sistema, es decir, permite describir sistemas en función de subsistemas y la forma en la que estos están interconectados [5]. En segundo lugar permite la definición de un sistema en función de su comportamiento, ya sea describiéndolo vía ecuaciones o mediante lenguajes de programación.

Verilog-A/MS

Verilog ha sido por mucho tiempo el lenguaje para describir hardware digital complejo más popular. Inició como un lenguaje propietario de la compañía *Cadence Design Systems, Inc* luego de comprar *Gateway Design Automation* en 1990 pero fue donado por *Cadence* a la comunidad de diseñadores con la intención de que se convirtiera en un estándar. El estándar fue formalizado como *IEEE1364-1995*. A la par de este hecho aparece la necesidad de expandir el alcance del lenguaje, en otras palabras, era necesario desarrollar una extensión de *Verilog* que permitiera o soportara la simulación de sistemas analógicos y mezclados. Es por eso que en 1996 se lanza *Verilog-A*, que innovó al permitir la simulación analógica (pero, no solucionaba la integración de *Verilog-HDL* y *Verilog-A*, sin embargo esta no era su intención). A medida que el tiempo transcurría, las implementaciones de *Verilog-A* ampliaban su disponibilidad y el interés por integrarlo directamente con *Verilog-HDL*. Finalmente en el año 2000 se divulga la definición y sintaxis de *Verilog-AMS*.

Como se observa en la figura 2, *Verilog-AMS* concreta la integración funcional de dos lenguajes, *Verilog-HDL* y *Verilog-A*, además agrega la simulación de señal mixta.

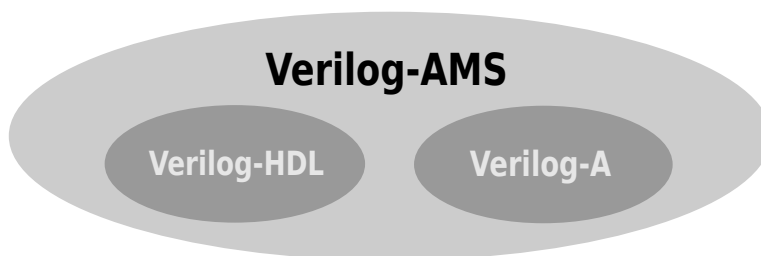


Figura 2: Relación entre *Verilog-AMS*, *Verilog-A* y *Verilog-HDL* [1].

Dentro de sus características, *Verilog-A/MS* proporciona la portabilidad que se necesita entre los diseños, su alto nivel de abstracción gracias al lenguaje de alto nivel logra la eficiencia requerida en la comunicación y la capacidad de componer un esquema totalmente funcional entre todos los niveles que integren el diseño; sistema, modelos, transistores etc.

Por consiguiente, el uso de los bloques modelados se extiende a la verificación [6].

SYSTEMC-AMS

SystemC-AMS es una extensión del estándar *IEEE1666-2011 SystemC*. Fue acogida como una iniciativa por el grupo *Accellera Systems Initiative*⁸ para expandir el lenguaje de verificación y diseño *SystemC* a un nivel analógico que permitiera profundizar en áreas de señal mixta y *RF*. Además de proporcionar una nueva alternativa a los estándares en simulación y verificación existentes.

Al estar basado en *SystemC* esta construido en C++ ordinario. Esto proporciona un lenguaje familiar al ingeniero que ha permitido una rápida adopción del lenguaje a nivel industrial [8]. *SystemC-AMS* permite el modelado a nivel de sistema, exploración de arquitecturas, verificación y análisis del rendimiento de sistemas.

MATLAB

MATLAB es un referente en cualquier rama de las ingenierías. Contiene librerías, *blocksets*⁹ y *toolboxes*¹⁰ que abarcan casi cualquier área de estudio. Por lo tanto, siempre será incluida como una alternativa viable en el diseño de circuitos integrados analógicos.

MATLAB tiene su propio lenguaje de programación, posee una versatilidad inigualable, es capaz de afrontar cualquier área sin ninguna dificultad o contratiempo. Este lenguaje es considerado como de alto nivel, permite la comunicación con *hardware* y la interacción entre diversos lenguajes de programación. Además a medida que las necesidades aumentan se adapta para presentar soluciones innovadoras, como es el caso de aceptar la creación de código *HDL* [9]. Esta versatilidad lo incluye dentro de las herramientas que acompañan las metodologías de diseño.

Cuando se medita sobre el papel que desempeña *MATLAB* en el diseño de circuitos integrados analógicos, se entiende su adaptación más como una herramienta de ayuda paralela ocupándose del tratamiento de datos, que como un mecanismo que cubra todo el proceso de diseño. Por otro lado, la oferta de *software* especializado en el área, aleja a *MATLAB* del propósito de construir una herramienta concreta que structure y desarrolle toda la metodología de *ICs*.

En este capítulo se revisaron las opciones más relevantes en el área de modelado de circuitos integrados analógicos, presentado los lenguajes de descripción de *hardware* analógicos y de señal mixta *VHDL-AMS* y *Verilog-A/MS*. Paralelamente se presentaron lenguajes alternativos considerados importantes como opciones para el desarrollo del proyecto. Además, la motivación del proyecto concluye, que el modelado de circuitos integrados analógicos es un área de crecimiento que permite la selección de arquitecturas y especificaciones de diseño.

Este proyecto será realizado modelando sistemas bajo el estándar *Verilog-A/MS* después de analizar la literatura [1, 3, 4, 6] disponible para el aprendizaje del lenguaje y las herramientas computacionales de diseño con las que dispone el grupo CIDIC. En el siguiente capítulo será presentada la teoría y léxico de *Verilog-A/MS*, así como las herramientas y el comportamiento de los circuitos y sistemas más relevantes.

⁸*Accellera* es un organización dedicada a la creación, soporte y promoción de estándares para la industria mundial de la electrónica [7].

⁹Término en inglés: Ampliación de *Simulink* mediante bloques que desarrollan un campo de estudio.

¹⁰Término en inglés: Conjunto de herramientas que ofrece *Matlab* para el desarrollo de un área.

Capítulo 1

MODELADO DE *ICs*

Los avances en la tecnología microelectrónica de las últimas décadas tienen hoy innumerales y variados dispositivos en el mercado que han presionado a la industria de fabricación a desarrollar métodos optimizados de diseño. Estos métodos deben tener el menor número de pasos y a su vez mejorar los resultados en velocidades de simulación, fabricación y producción. Es por esto que al adentrarse en la rutina del diseñador de circuitos integrados se manifiesta la necesidad de tener un mayor y mejor número de herramientas a su disposición. Una de las herramientas más importantes a la hora de diseñar un circuito integrado, son los modelos de dispositivos o circuitos, que sean lo mas cercanos a su comportamiento real, esto con el fin de reducir el error humano a la hora de implementar físicamente el diseño.

El modelado de circuitos integrados en sí mismo es una gran parte del diseño, ya que éste se encarga de proveer dispositivos virtuales que sigan el comportamiento de un sistema físico. Un modelo de alto nivel debe estar en la capacidad de reproducir la actuación de su correspondiente real.

Un modelo puede simular procesos físicos dentro de un transistor, por ejemplo empleando ecuaciones matemáticas que describan su respuesta a diferentes estímulos. Otros modelos permiten observar el comportamiento de un sistema específico como lo es un amplificador operacional. Ya que éste contiene varios transistores, es viable utilizar submodelos para conformar una descripción de tipo estructural del amplificador. En ocasiones en el diseño de *ICs* se derivan situaciones donde lo importante es conocer de antemano otro tipo de características de un dispositivo diferentes a su salida respecto a su entrada. El consumo de potencia, la ganancia por ancho de banda¹ y el ruido, son especificaciones de diseño cruciales para el desarrollo de la metodología de diseño. Estos detalles deben realizarse mediante cálculos manuales, seguidos de una tediosa serie de pruebas de verificación, que pueden concluir en errores, sacrificando horas de diseño.

Ahora bien, invertir tiempo en el modelado de un sistema o circuito, que tenga parámetros variables, y que sea lo suficientemente versátil para ajustarse tanto a diferentes proyectos como a especificaciones puntales de diseño resulta rentable y extremadamente útil para cualquier compañía diseñadora de circuitos. Asimismo, se ha convertido en objetivo de los diseñadores en la actualidad, producir sus propias librerías.

A lo largo de este capítulo se presenta el proceso de modelado de circuitos analógicos mediante *Verilog-A/MS*, al tiempo que se hace un recorrido por la sintaxis del lenguaje acompañado de un ejemplo sencillo de un amplificador diferencial.

¹Conocido como *Gain-bandwidth product GBW*.

1.1. APLICACIONES DE *VERILOG-A/MS*

Según *Kenneth Kundert* en [1], cinco son las razones por las cuales los ingenieros usan *Verilog-A/MS*.

- Para modelar componentes.
- Para crear bancos de pruebas.
- Para acelerar las simulaciones.
- Para verificar sistemas de señal mixta.
- Para apoyar y cimentar el flujo de diseño descendente “*top-down*”².

1.1.1. Modelado de Componentes

Al entrar a evaluar las opciones disponibles en simuladores de circuitos, se encuentra que las herramientas tradicionales como *SPICE* proporcionan variedad de modelos, pero la obtención de nuevos componentes es relativamente limitada. En contraste *Verilog-A/MS* provee gran variedad de características que pueden ser utilizadas para la descripción eficiente de una amplia gama de componentes o modelos [1]. Esta propiedad posibilita la creación de componentes básicos tales como resistores, capacitores e inductores, pero también propicia la concepción de elementos semiconductores como diodos, *BJTs* y *MOSFETs* [6]. De la misma manera permite la creación de bloques funcionales a nivel sistema, que abarcan amplificadores operacionales, conversores de datos y reguladores junto a componentes digitales o incluso multidisciplinarios como registros, compuertas lógicas, *latches*, sensores actuadores y transductores [10]. Además, introduce la mezcla de todos estos en un mismo contexto.

1.1.2. Bancos de Pruebas

Con el objetivo de tener un proceso efectivo de verificación del diseño, se puede escribir un fragmento de código en *Verilog-A/MS* que realice pruebas específicas y contextuales al circuito analizado. Este código llamado banco de pruebas³ reemplaza la circuitería esquemática que se agrega para verificar el diseño. Ya que no es necesario un circuito adicional, se gana tiempo de simulación y cómputo. Aparte de esto, tener una estructura organizada de verificación dentro de la metodología de diseño asegura una mejor depuración de errores y mayor confiabilidad en el diseño.

1.1.3. Velocidad de Simulación

La alta complejidad y tamaño de los circuitos electrónicos tiene como resultado tiempos de simulación extensos, tiempo que puede emplearse para el desarrollo de otras tareas si se cuenta con un proceso de verificación estructurado. Ya que en todo circuito es posible diferenciar bloques críticos, es conveniente separarlos y realizar simulaciones con las partes no críticas del circuito reemplazadas por modelos comportamentales. En este caso, el tiempo de simulación puede ser reducido teniendo en cuenta que un modelo comportamental de un bloque no crítico simulará más rápido que uno a nivel transistor del mismo bloque.

²Término en inglés que hace referencia al flujo de diseño descendente.

³Viene de la traducción del término original en inglés *Test Benches*, que consiste en proporcionarle un entorno adecuado no gráfico a la verificación.

1.1.4. Diseño de Señal Mixta

Al lograr la integración en un mismo diseño de circuitos analógicos y digitales, *Verilog-A/MS*, permite combinar de manera fácil y ágil las representaciones de ambos campos, abriendo la puerta a simulaciones entre compuertas lógicas y circuitos a nivel transistor. Ahora los diseñadores de ambas ramas pueden ejecutar lenguajes familiares y cómodos para ellos.

1.1.5. *Top-Down Design*

Esta metodología es de gran utilidad cuando se diseñan grandes y complejos sistemas. La idea básica es diseñar y verificar el sistema en un nivel abstracto o general antes de iniciar el diseño detallado de los bloques individuales. Esto quiere decir, que mucho antes de adentrarse en cálculos matemáticos complejos, se bosqueja una aproximación del sistema a nivel de bloques, permitiendo conocer así resultados de especificaciones y comportamiento de los bloques individuales y en conjunto.

En el diseño *top-down*, el desempeño necesario de un bloque individual para cumplir el rendimiento del sistema se evalúa detalladamente y garantiza el completo entendimiento antes de que sea desarrollado el bloque. Esto posibilita anticipar el comportamiento del sistema.

Claramente para lograr la reducción de tiempo es necesario que exista una interacción entre bloques y sistema. Acá es donde el lenguaje cumple su propósito y reclama importancia. *Verilog-A/MS* permite una comunicación fluida; una comunicación donde los modelos descritos pueden ser intercambiados.

Con un modelo, el diseñador puede correr diferentes escenarios para clarificar el significado y la intención del desempeño de un bloque en un sistema, al obtener un resultado no esperado, el ingeniero está en la capacidad de reestructurar la idea cambiando a su gusto un bloque por otro sin perjudicar el funcionamiento global de su idea, evadiendo de esta forma el costo en tiempo de cálculos manuales.

Esta versatilidad es explotada en este proyecto para avanzar en un campo algo olvidado en nuestra región, la creación de propiedad intelectual. Los modelos presentados en este proyecto dejarán sentada la base para librerías funcionales a nivel industrial que permiten específicamente la selección de parámetros de diseños y el ahorro de horas hombre en la elaboración de circuitos integrados.

1.2. SINTAXIS DE *VERILOG-A/MS*

En esta sección se muestran las generalidades del lenguaje *Verilog-A/MS* utilizando la estructura propuesta en [6] con la intención de familiarizar al lector con el léxico que se utiliza a lo largo del proyecto. Esto se hace a través del modelado básico de un amplificador diferencial que se observa en la figura 1.1, considerado por muchos autores como uno de los bloques fundamentales para la construcción de circuitos analógicos [11–13]. El amplificador diferencial de la figura 1.1 resulta una excelente elección para iniciar el estudio del modelado de *ICs* a la vez que la inducción al lenguaje *Verilog-A/MS*.

Desde un punto de vista didáctico, el par diferencial plantea un reto interesante, generar un bloque que se comporte como un amplificador diferencial en características de ganancia teniendo en cuenta una señal de entrada de *offset*, para encontrar las especificaciones de diseño óptimas del amplificador. En este punto no es necesario diseñar el amplificador diferencial a nivel transistor porque se desea comprobar previamente si este circuito cumple las características de desempeño dentro de un sistema.

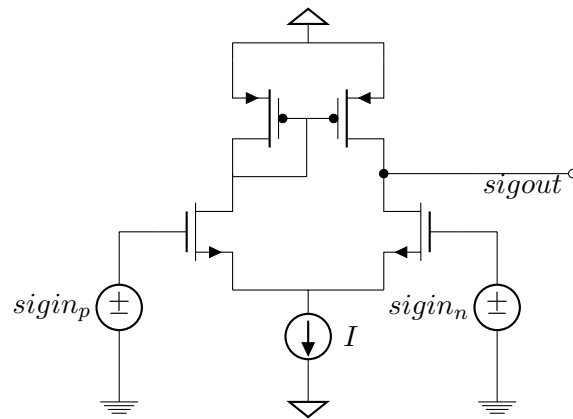


Figura 1.1: Amplificador diferencial.

La configuración por diferencial presenta en su estructura más simple una única salida; esta topología de amplificación sitúa en dicha salida, una señal amplificada de la diferencia de sus entradas. Ésta es representada por la señal *sigout*, que viene de:

$$V_{sigout} = A_V(V_{signin_p} - V_{signin_n}) \quad (1.1)$$

$$A_V = \frac{V_{sigout}}{(V_{signin_p} - V_{signin_n})} \quad (1.2)$$

Ahora es incluido el efecto que produce una señal de *offset*:

$$V_{sigout} = A_V((V_{signin_p} - V_{signin_n}) - V_{signin_offset}) \quad (1.3)$$

Con estas características en mente, el modelado toma forma, lo que se necesita es plasmar el concepto de amplificación de una forma descriptiva, mediante el lenguaje. A continuación se modela el amplificador diferencial solamente para las características indicadas anteriormente.

1.2.1. *Lexical Tokens*

En todos los lenguajes de programación existen las estructuras de lenguaje más simples, estas son llamadas sintaxis de bajo nivel o *lexical tokens*. Todas las declaraciones escritas con un lenguaje están construidas a partir de éstos, además un *token* es la unidad significativa más pequeña de una declaración que por definición está compuesta por uno o mas caracteres.

A continuación se presentan los tipos de *tokens*:

- Espacio en blanco.

```
1 // VerilogA for Amplif, Ampl_Dif, veriloga
2
3 'include "constants.vams"
4 'include "disciplines.vams"
5
6 module Ampl_Dif(sigin_p, sigin_n, sigout);
7
8 input sigin_p, sigin_n;
9 output sigout;
10 electrical sigin_p, sigin_n, sigout;
11
12
13 parameter real gain = 1;
14 parameter real sigin_offset = 0;
15
16
17     analog
18         V(sigout) <+ (V(sigin_p, sigin_n) - sigin_offset) * gain;
19
20
21
22
23
24
25 endmodule
```

Figura 1.2: Código Par Diferencial.

- Comentario.
- Operadores.
- Números.
- Identificador.
- Palabras clave.
- Cadenas.

Los espacios en blanco están en toda la descripción de la figura 1.2, corresponden al símbolo espacio en blanco, tabuladores y líneas nuevas de código y no tienen un significado distinto al de separar.

Los comentarios se pueden presentar de dos formas. Una de ellas es iniciar con los caracteres `//`, el comentario terminará cuando finalice esa línea, esto se ve claramente en la figura 1.2 en la línea número 1. La otra es comentar un bloque iniciando con `/*` y cerrar con `*/`.

Los operadores aceptados por *Verilog-A/MS* se muestran en la tabla 1.1. Dentro del código de la figura 1.1 se pueden observar los operadores aritméticos en la línea 18.

+ - * /	aritmética
%	módulo
> >= < <=	relación
!	negación lógica
&&	and lógica
	or lógica
==	igual lógico
!=	desigualdad lógica
~	bit-wise negación
&	bit-wise and
	bit-wise or inclusiva
^	bit-wise or exclusiva
^~ or ~^	bit-wise equivalencia
<<	left shift
>>	right shift
? :	condicional
or	evento or

Cuadro 1.1: Operadores.

Los identificadores establecen a un objeto un único nombre, de manera que pueda referirse a él más adelante durante la descripción. En el código del par diferencial de la figura 1.2 se observan varios identificadores como: *gain*, *sigout*, *sigin_offset*.

Las palabras claves son reservadas por el lenguaje, están predefinidas para la construcción del mismo. Se deben utilizar todas en letras minúsculas. En el código de la figura 1.2 las palabras claves son: *include*, *module*, *input*, *output*, *electrical*, *parameter*, *real* y *endmodule*.

1.2.2. Tipos de Datos

Verilog-A/MS soporta datos del tipo real, entero y parámetros. Asimismo arreglos de números reales como una extensión de los tipos de datos reales.

Además, *Verilog-A/MS* introduce un nuevo tipo de dato, llamado nodo, este es utilizado para representar señales analógicas. Éstos nodos tienen disciplinas que definen la naturaleza

del potencial y flujo entre otras características.

En las líneas de código 13 y 14 de la figura 1.2, se presenta la sintaxis para la definición de parámetros de tipo real.

1.2.3. Expresiones

Una expresión en *Verilog-A/MS* se define como una construcción, que combina operadores y operandos. Un operador, como se vio en la sección *Lexical Tokens*, es un arreglo de caracteres que relacionan los valores de uno o más operandos para producir un resultado.

La línea 18 del código del amplificador diferencial, es un ejemplo de como utilizar los operadores y operandos para formar una expresión.

1.2.4. Señales

Para diferenciar las señales analógicas de las digitales, *Verilog-A/MS* identifica si la señal tiene asociada una disciplina. Una disciplina es un conjunto de propiedades que describe las señales analógicas en el lenguaje. Estas propiedades permiten reconocer por ejemplo si se está accediendo a un potencial o a un flujo.

Las señales tienen además de las disciplinas, una naturaleza asociada que permite proveerle singularidad a la disciplina. Por ejemplo una disciplina de tipo *electrical*, como el caso de la figura 1.2 consiste de dos naturalezas, voltaje y corriente. Estas definen las cantidades, unidades y características físicas de las señales tan pronto se accede a una de ellas.

Todos los nodos, puertos y ramas son leídas mediante *access functions*. El nombre de la *access function* es tomado de la disciplina del nodo, puerto o rama asociada. Por ejemplo, considere la línea 18 de la figura 1.2.

```
V(sigout) <+ (V(sigin_p, sigin_n) - sigin_offset) * gain;
```

En este caso *electrical* es la disciplina. El nombre de la *access function* es *V*, que se utiliza para acceder al voltaje del puerto *sigout*.

1.2.5. Comportamiento analógico

La descripción de un modelo de forma comportamental se hace mediante la construcción de varios operadores para nodos bajo ciertos procedimientos o control de tiempo.

Cuando se describe un modelo de tipo analógico, como es el caso que compete a este proyecto, la descripción comportamental de un bloque debe iniciar con declaración *analog*.

Es decir, que todas las descripciones, características, interacciones, fórmulas y demás elementos que conformen la explicación detallada del proceder del modelo y comportamiento en el dominio del tiempo deben ir encapsuladas entre un *analog*.

Como se observa en el ejemplo del amplificador operacional de la figura 1.2, la descripción está contenida en el *analog*. Todo lo que esté por fuera de este bloque de código no interfiere para nada en el comportamiento del modelo.

1.2.6. Jerarquía

Cuando se necesita tener una estructura ordenada de modelos y submodelos, ya sea por la complejidad de la descripción o por incluir partes digitales y analógicas al mismo tiempo, *Verilog-A/MS* ofrece la estructura jerárquica indicada para conseguirlo. Esto permite que modelos contengan fracciones de bloques con una comunicación fluida entre ellos.

Para esto necesita definir donde inicia y termina el módulo. Esto se hace con *module* y *endmodule*. La jerarquía de modelos podría apreciarse como una relación padre-hijo, el modelo padre esta en la cima de la jerarquía y el hijo en la inferior.

Capítulo 2

MODELOS

Después del análisis de diversos lenguajes para el modelado de sistemas y *hardware* analógico y de señal mixta más relevantes en la industria, de los circuitos interesantes para el desarrollo de este proyecto y de las especificaciones fundamentales a tener en cuenta al modelar circuitos integrados analógicos a través capítulo 1, se seleccionó *Verilog-A/MS* como el lenguaje más conveniente para el modelado de los circuitos destacados.

A lo largo del capítulo 2 se presentará de forma detallada el modelado de los circuitos seleccionados, resaltando las características y especificaciones de respuesta y comportamiento que contribuirán a la selección de arquitecturas y/o especificaciones de diseño.

2.1. FUENTE DE VOLTAJE DE REFERENCIA

Una fuente de referencia proporciona una tensión estable, idealmente esta debe ser independiente a cualquier variación interna o externa de temperatura. De este modo la fuente de referencia puede mantener polarizados los demás bloques funcionales del circuito o sistema [14]. En la vida real este rango de temperatura depende del contexto y la aplicación para el cual sea diseñada la fuente de referencia. Para el caso que compete a este proyecto, la fuente de referencia operará en un rango de temperatura entre -50°C y 150°C teniendo en cuenta los máximos rangos encontrados en la literatura [14] [15] [16].

Por ser un modulo ampliamente utilizado en el diseño y desarrollo de circuitos integrados, la fuente de tensión de referencia combina las características necesarias para ser objeto de modelado en este proyecto.

Una referencia debe proveer un voltaje que cumpla con las siguientes condiciones [17]:

- Poca dependencia de la alimentación.
- Poca dependencia con los parámetros del proceso.

- Dependencia bien definida con la temperatura.

2.1.1. Parámetros

La tabla 2.1 resume todos los parámetros del modelo, indicando los valores mínimos y máximos admitidos.

Parámetros	Valor mínimo	Valor Máximo
vref [V]	0.5	10
vov [V]	0.15	1
reglin [V/V]	0	1
tc [ppm]	0	1000
comp [-]	1	2
np [uV]	0	10k

Tabla 2.1: Parámetros y Rango de Valores.

Los 6 parámetros de la fuente de referencia de voltaje ofrecen al usuario la posibilidad de manipular el comportamiento de la misma a su antojo, dándole la maniobrabilidad que necesita en un modelo parametrizable, prediciendo el comportamiento del circuito antes de adentrarse en el proceso de diseño a nivel transistor, ahorrando tiempo y costos. A continuación se realiza una breve descripción de cada uno de ellos para familiarizar al lector con la terminología. El comportamiento de cada uno de ellos dentro del modelo puede observarse en el apéndice A.

La fuente de referencia puede dividirse en dos grupos de parámetros, los relacionados a la compensación y uno relacionado con el ruido. Con los parámetros *vref* y *vov* el usuario indica cuanto quiere que sea la tensión de salida de la fuente y cuando debe ser el mínimo de valor de sobretensión necesario para que la fuente pueda operar. El parámetro *reglin* será el asociado a la regulación de línea, que corresponde a la capacidad del modelo de mantener el voltaje a la salida cuando varíe la señal de alimentación. Por otro lado, los parámetros *tc* y *comp* están asociados a la compensación a variaciones de temperatura, siendo *tc* el coeficiente de variación con la temperatura y *comp* el tipo de compensación que el usuario desea; que por simplicidad puede ser de primer o segundo orden.

Finalmente con el parámetro *npp*, el diseñador puede adicionar una cantidad de ruido a la salida del modelo.

2.1.2. Regulación de Línea

El parámetro *reglin* permite cuantificar la regulación de línea en [V/V] que se desea a la salida de la fuente. El modelado matemático de la regulación de línea se hizo teniendo

en cuenta dos casos, el primero, en el cual la tensión de alimentación es menor que la suma $vref + vov$, la fuente asignará a la salida una tensión proporcional a la señal de alimentación dada por $slope * V_{dd}$, donde $slope$ corresponde al factor proporcionalidad que se explica en la ecuación 2.1.

$$slope = \frac{vref}{vref + vov} \quad (2.1)$$

El segundo caso ocurre cuando la señal de alimentación sobrepasa $vref + vov$, en este caso el modelo de la fuente de referencia asigna a la salida la tensión de $vref + vrl$, donde vrl viene dada por la ecuación 2.2.

$$vrl = vdiff * reglin \quad (2.2)$$

Y $vdiff$ por:

$$vdiff = V_{dd} - vref \quad (2.3)$$

Donde V_{dd} es la tensión de alimentación.

2.1.3. Orden de Compensación

El modelo de la fuente de referencia permite dos tipos de compensación de temperatura. El tipo de compensación se puede escoger mediante el parámetro $comp$, donde 1 y 2 corresponden a compensaciones de primer y segundo orden respectivamente. El modelado se realiza mediante un arreglo lineal por partes dependiendo de la variación de la temperatura.

El coeficiente de temperatura tc es el parámetro que define como la salida de una fuente de referencia de tensión varía sobre una temperatura dada. Este parámetro da al usuario una idea del comportamiento de la compensación, las unidades son partes por millón [ppm] [18]. Las ecuaciones que definen este parámetro están dadas en la literatura [19] [20] y son las que se observan en las ecuaciones 2.4, 2.5 y 2.6.

$$tc(V_0) = \frac{\Delta V_0}{\Delta T} \quad \frac{mV}{^\circ C} \quad (2.4)$$

$$tc(V_0) = 100 * \frac{\frac{\Delta V_0}{V_0}}{\Delta T} \quad \frac{\%}{^\circ C} \quad (2.5)$$

$$tc(V_0) = 10e6 * \frac{\frac{\Delta V_0}{V_0}}{\Delta T} \quad \frac{ppm}{^\circ C} \quad (2.6)$$

Con base en las ecuaciones anteriormente descritas y las unidades usuales en el diseño de circuitos, se tomó como referencia para el modelado del tc la ecuación 2.6.

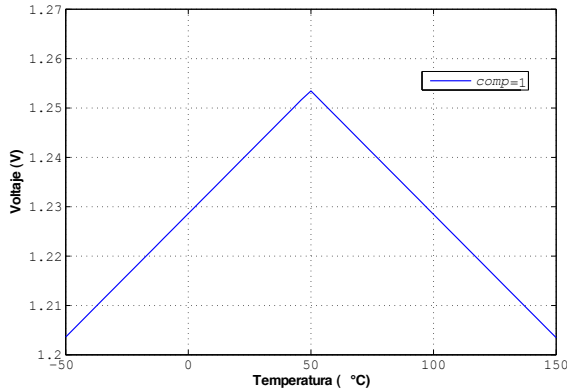
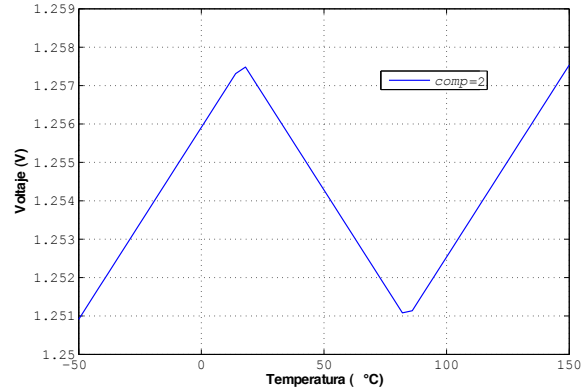
(a) Compensación de Primer Orden. $comp = 1$.(b) Compensación de Segundo Orden. $comp = 2$.

Figura 2.1: Tipos de Compensación en el Modelo de la Fuente de Referencia.

La compensación de primer orden plantea un punto de inflexión en 50°C , como se observa en la figura 2.1(a). Así, cada vez que el diferencial de tensión alcanza su valor máximo, es decir el punto de inflexión en temperatura, el modelo fuerza su descenso proporcionalmente al coeficiente de temperatura tc .

Por su parte la compensación de segundo orden tiene dos puntos de inflexión definidos en 17°C y 84°C (figura 2.1(b)). Aun así, usa el mismo concepto de la compensación de primer orden para neutralizar los efectos de la temperatura en el modelo.

2.1.4. Ruido

El ruido incluido en la fuente de referencia se modeló con la intención de aportar una solución al problema de la no visualización de los efectos de ruido en simulaciones transitorias, ya que el ruido generado mediante las funciones del lenguaje solo es observable mediante análisis *.noise*. La figura 2.2 ilustra el concepto de ruido utilizado en la fuente de regulación de voltaje. Esta figura muestra valores aleatorios de tensión sobre la señal de salida del regulador. Su comportamiento está dado por el parámetro npp , que asigna una señal de voltaje del valor fijado, multiplicada por números aleatorios generados mediante el mecanismo *\$random* del lenguaje.

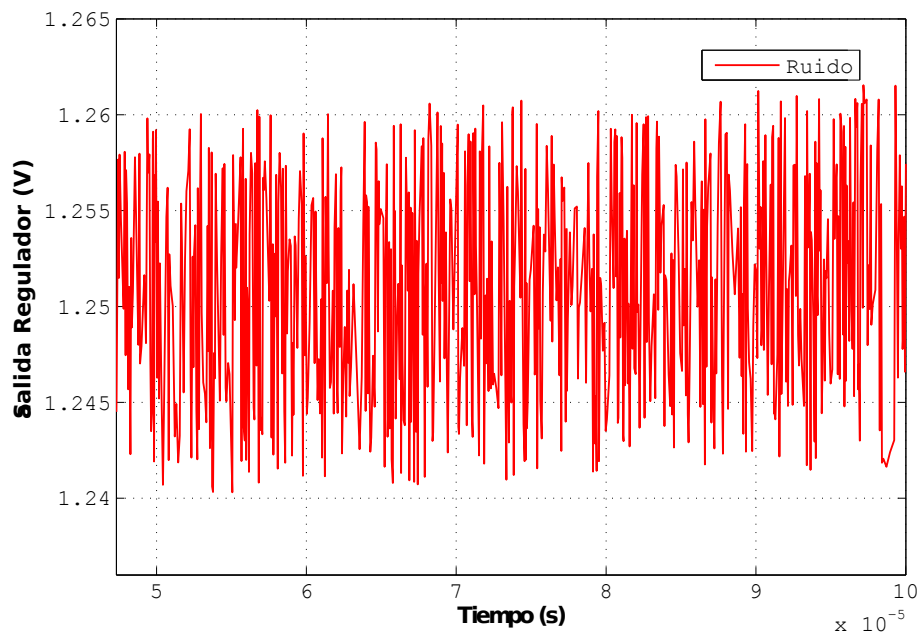


Figura 2.2: Ruido generado por el parámetro npp .

2.2. AMPLIFICADOR OPERACIONAL

2.2.1. Parámetros

En la tabla 2.2, se listan todos los parámetros del modelo del amplificador indicando sus límites superiores e inferiores, como guía para el lector durante el recorrido de la sección y lo que resta del libro.

El primer paso de un diseño es encontrar el conjunto completo de especificaciones para el sistema electrónico. El diseñador tiene que especificar el comportamiento deseado para luego poder seleccionar topologías, circuitos integrados y sistemas. En la práctica existen muchas maneras de especificar el desempeño de un circuito electrónico, esta sección presentará los aspectos que a juicio del autor son fundamentales en el rendimiento del amplificador operacional.

2.2.2. Modelado del Producto Ganancia-Ancho de Banda GBW

Debido a la importancia del GBW como parámetro a la hora de escoger topologías y como especificación en el desarrollo de circuitos integrados, resulta primordial incluirlo como una de las especificaciones esenciales. El análisis del ancho de banda se realiza observando el comportamiento que tienen los circuitos en el dominio de la frecuencia, donde se puede

Parámetros	Valor mínimo	Valor Máximo
gain [dB]	0	120
GBW [Hz]	5.5 k	4G
Margen de Fase [°]	0	90
outbias [V]	V_{ss}	V_{dd}
offset [V]	1μ	30m
CMRR [dB]	38	139
PSRR [dB]	38	139
f _{psr} [Hz]	-	-
n _{pb} [Hz]	-	-
v _{rms} [V _{rms}]	1n	inf
en [V \sqrt{Hz}]	1n	25 μ
f _{min} [Hz]	1	100
Slew Rate [V/ μS]	4n	10.5k
v _{clipp} [V]	0	500m
v _{clipn} [V]	0	500m
r _{out} [Ω]	0	inf
c _{out} [F]	0	inf

Tabla 2.2: Parámetros y Rango de Valores.

observar características de un sistema como estabilidad y margen de fase.

El módulo del amplificador contiene un componente que propone la ubicación del polo dominante a partir del GBW y el margen de fase. Estos argumentos de estimación se observan en la figura 2.3 donde se asume que la ubicación del polo 2 es dentro de la zona demarcada (± 1 década del GBW definido por el usuario). Así mismo se sabe que el GBW y UGF son aproximadamente iguales para sistemas que se comportan como unipolares. A partir de dichas suposiciones se obtiene que la frecuencia del polo 2 y su ganancia están descritas mediante las ecuaciones 2.7 y 2.8 respectivamente.

$$f_{P2} = \frac{GBW}{10^{(1-\frac{PM}{45})}} \quad (2.7)$$

$$A_{P2} = 20 \log_{10} \left(\frac{GBW}{f_{P2}} \right) \quad (2.8)$$

La ecuación 2.9, es la cantidad de décadas que separan al polo 1 y al polo 2 es decir, la diferencia en frecuencia de la ubicación de los polos. Finalmente la ecuación 2.10 estima la ubicación del polo 1.

$$DECS = \frac{A_0 - A_{P2}}{20} \quad (2.9)$$

$$f_{P1} = \frac{f_{P2}}{10^{DECS}} \quad (2.10)$$

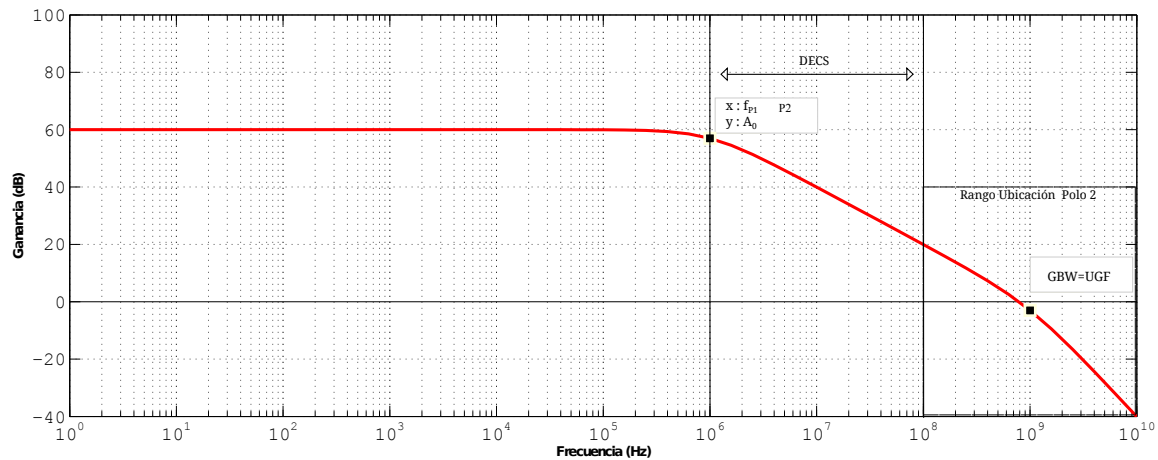


Figura 2.3: Modelado de GBW.

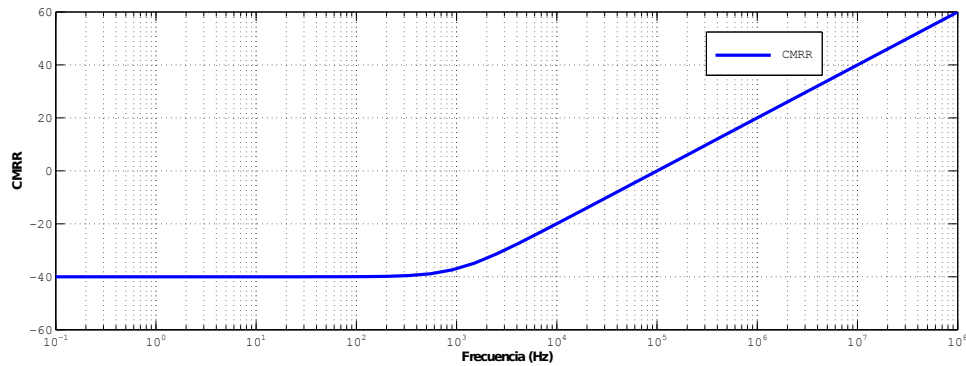
2.2.3. CMRR y PSRR

El rechazo a señales en modo común o CMRR tiene un parámetro dentro del modelo que indica que tanta atenuación a señales en modo común desea el usuario, permitiéndole así formarse una idea de que posibilidades tiene a la hora de escoger una topología para su diseño. El CMRR fue modelado asumiendo que el valor ingresado por el usuario sea el rechazo en dB. El modelo atenuará la señal en esa cantidad indicada para valores que estén dentro del rango de frecuencias del primer polo del sistema. Este comportamiento se observa en la figura 2.4(a).

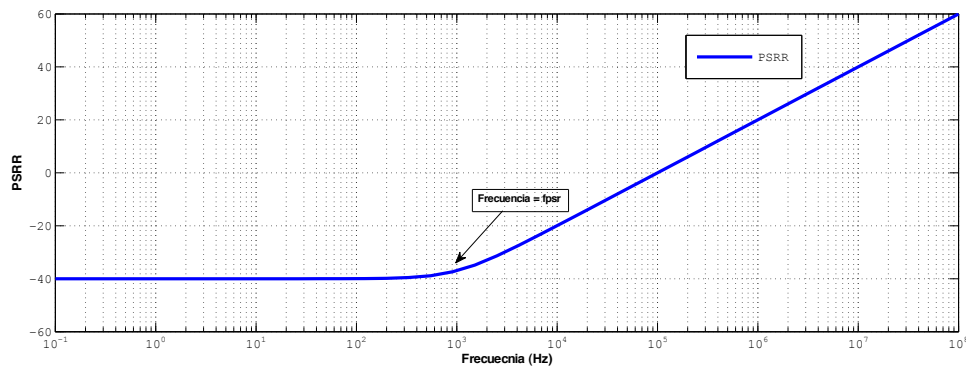
Para el PSRR se le da la flexibilidad al usuario de escoger la frecuencia de PSRR junto con el nivel de rechazo que desea. La actuación del modelado del PSRR se percibe en la figura 2.4(b). El parámetro f_{psrr} indica la frecuencia donde el rechazo al ruido de la fuente de alimentación empieza a disminuir.

2.2.4. Modelado de *BIAS* y *offset*

Con el fin de darle una aproximación más cercana a la realidad, se incluyen dos parámetros relacionados al comportamiento en DC: *outbias* y *offset*. En el modelo del amplificador



(a) Comportamiento del CMRR.



(b) Comportamiento del PSRR

Figura 2.4: CMRR y PSRR.

operacional el usuario tiene la posibilidad de establecer una tensión de polarización (*outbias*) que por defecto es 0.9 [V], correspondiente a $\frac{V_{dd}-V_{ss}}{2}$ para $V_{dd} = 1.8$ y $V_{ss} = 0$. Esta tensión ubica la señal de salida en un nivel DC definido por el usuario.

Por su parte el parámetro *offset* asigna un valor en DC, modelando de esta manera perturbaciones añadidas por la movilidad, efectos del óxido y *edge effects*¹ [21].

El aporte de estos parámetros se hace mediante la línea de código que se muestra a continuación, donde al nodo "nodesr" se le atribuye un valor proporcional a la ganancia en dB sumado al valor de *BIAS*.

```
V(nodesr) <+ ( (pow(10,(gain/20)))*offset*signumber+ outbias );
```

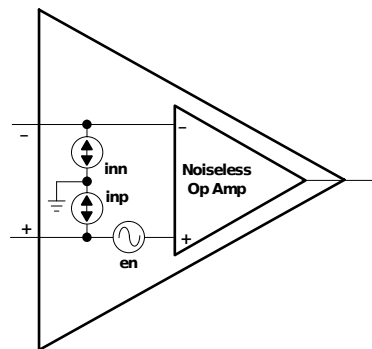
¹Término en inglés: Imperfecciones producidas por el óxido en los bordes de los elementos.

2.2.5. Modelado del Ruido

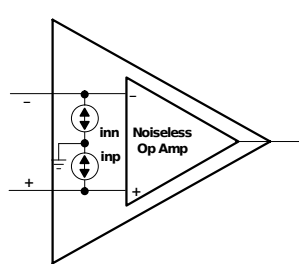
Los amplificadores operacionales tienen diferentes tipos de ruido asociado (shot, térmico, flicker, burst y avalancha). Teniendo en cuenta que en la mayoría de los casos no es posible separar los efectos generados por estos tipos de ruido al interior de un *Op Amp* [22], y analizando cuales de estos son los mas significativos, se decidió modelar dos clases de ruido: el ruido *flicker* que ha sido objeto de estudio en las clases de electrónica por estar siempre presente en la naturaleza y en muchos dispositivos activos y pasivos [22], y el ruido térmico por su aproximación a ruido blanco cubriendo de esta manera la mayor parte del espectro. El modelo del presente proyecto posee una configuración que permite al usuario incluir estos dos tipos de ruido, para después tomar decisiones específicas sobre arquitecturas y especificaciones.

De esta manera, si el diseñador conoce las causas del ruido está en la capacidad de controlar el modelo minimizando o aumentando el ruido en un ancho de banda en particular y controlar la proporción en la que los diversos tipos de ruido afectan el desempeño del circuito.

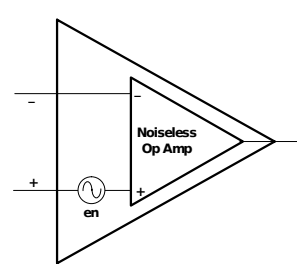
El modelo de ruido en un *Op Amp*, puede representarse como la interconexión de fuentes de tensión de ruido y fuentes de corriente de ruido tal y como se observa en la figura 2.5(a). La figura 2.5(b) simboliza el ruido únicamente como la conexión de fuentes de corriente. Para el caso que compete a este proyecto el modelo del ruido es basado en la figura 2.5(c), que representa el ruido como una fuente de tensión.



(a) Modelo General de Ruido.



(b) Modelo Mediante Fuente de Corriente.



(c) Modelo Mediante Fuente de Tensión.

Figura 2.5: Modelo de Ruido [22].

Verilog-A/MS provee un par de funciones que permiten incluir los dos tipos de ruido mencionados al modelo. Estas son:

```
white_noise( pwr [ , name ] )
```

Y

```
flicker_noise( pwr , exp [ , name ] )
```

Al hacer una suma de las dos es posible obtener una fuente de voltaje de ruido, que modela el comportamiento del mismo referenciado a la salida del *Op Amp*.

El análisis previo a la extracción de las ecuaciones de ruido se realizó mediante el estudio de la figura 2.6. Donde el comportamiento en bajas frecuencias corresponde al ruido *flicker* y el comportamiento horizontal representa un piso de ruido térmico. Partiendo de esto el equivalente del ruido blanco a modelar estaría dada por la ecuación 2.11, donde E_N corresponde al piso térmico de ruido, F_{3dB} es la frecuencia de esquina (donde los ruido térmico y *flicker* son iguales), F_2 representa al parámetro *npb* (ancho de banda de la potencia de ruido) y $F_1 = 1Hz$.

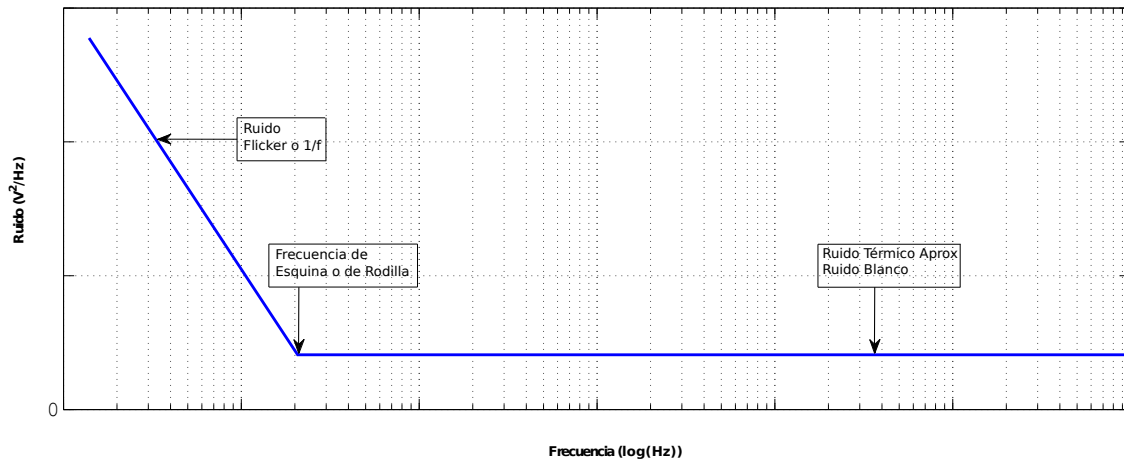


Figura 2.6: Comportamiento del Ruido.

$$E_Q = E_N * \sqrt{1 + \frac{F_{3dB}}{F_2 - F_1}} \quad (2.11)$$

Y el ruido integrado en banda está dado por:

$$V_{rms} = E_Q * \sqrt{NPB} \quad (2.12)$$

Asumiendo como voltaje pico de ruido:

$$V_{pp} = 6 * V_{rms} \quad (2.13)$$

La frecuencia de esquina despejando de 2.11 será:

$$F_{3dB} = \frac{\left(\frac{E_Q^2}{E_N^2} - 1\right) (F_2 - F_1)}{\ln\left(\frac{F_2}{F_1}\right)} \quad (2.14)$$

Reemplazando todo lo anterior en 2.14, se obtiene la frecuencia de esquina:

$$F_{3dB} = \frac{\left(\left(\frac{V_{rms}}{\sqrt{NPB}}\right)^2 - 1\right) (NPB - 1)}{\ln(NPB)} \quad (2.15)$$

De donde finalmente se obtiene el termino *pwr* de la función *flicker* de *verilog-A/MS*, asumiendo como conocida la densidad de potencia del ruido en 1 Hz:

$$pwr = F_{3dB} * E_N^2 \quad (2.16)$$

2.2.6. Slew Rate

En el modelado de gran señal, el parámetro *slewrates* identifica la no linealidad del amplificador al responder con una rapidez limitada a las variaciones en la señal de entrada. Este parámetro es modelado mediante la función *slew()* que posee *Verilog-A/MS*. La expresión de salida será limitada al valor impuesto en el parámetro *slewrates* una vez la señal sobrepase dicho valor, como se observa en la figura 2.7 [10].

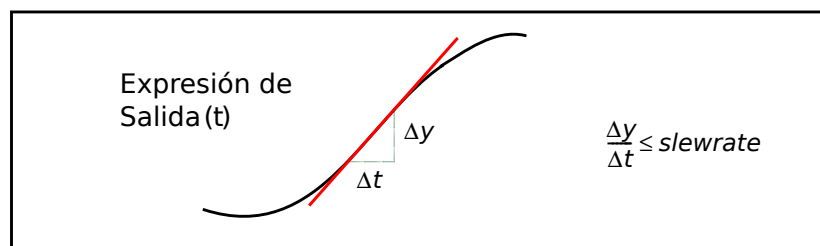


Figura 2.7: Comportamiento de la Función *slew()*.

2.2.7. Clipping

La saturación o *clipping*, limita la señal de salida a los rieles de alimentación o valores inferiores que el diseñador requiera. Esto faculta al usuario para incluir este tipo de distorsión

en gran señal proveniente de diferentes situaciones.

El modelo esta diseñado para que cuando la señal de salida supere los rangos impuestos por la alimentación(o por el usuario), la señal se corte y en su lugar aparezcan los valores de V_{dd} o V_{ss} (o los valores de los parámetros $vclipp$ y $vclipn$).

La figura 2.8 muestra la capacidad del modelo de saturar una señal por diferentes motivos. La línea continua identifica la señal saturada por los valores puestos en los parámetros $vclipp$ y $vclipn$, mientras que la línea a trazos representa un *clipping* por los rieles de alimentación. Finalmente la línea punteada seria la señal sin ningún tipo de distorsión.

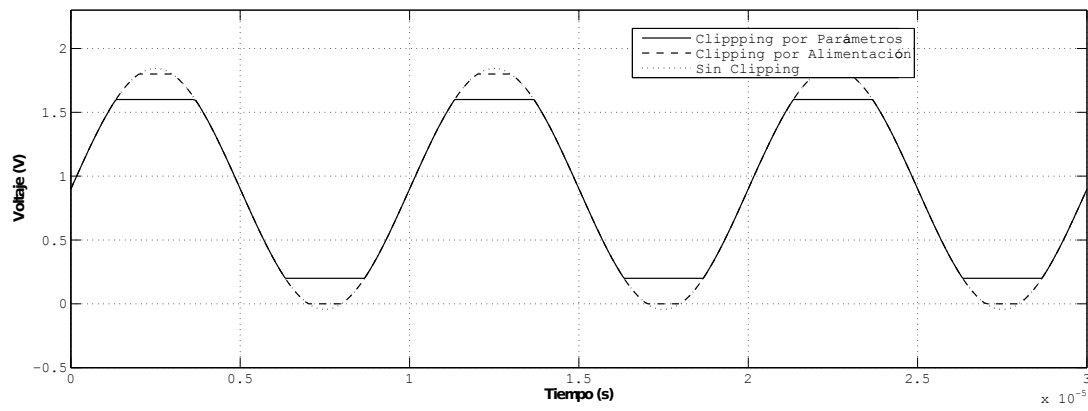


Figura 2.8: Modelado de la Saturación.

Capítulo 3

ANÁLISIS DE RESULTADOS

En el capítulo anterior, se presentaron cada una de las especificaciones de los circuitos modelados, y se analizaron las principales características que cada una de ellas aportaba a la creación de los modelos.

Este capítulo esta orientado a la recopilación y análisis de resultados de los circuitos modelados en el capítulo 2, además de su validación en un sistema de potencia (regulador *LDO*). Durante el transcurso del mismo se desarrollan simulaciones en AC, DC, temperatura, transitorias, análisis de ruido y distorsión de los modelos según sea el caso. Esto permite ver el comportamiento del amplificador operacional y la fuente de referencia de voltaje descritos en *Verilog-A/MS* observando detalladamente mediante gráficas y tablas el funcionamiento de los mismos.

3.1. FUENTE DE REFERENCIA DE TENSIÓN

Para ver con claridad que tan dependiente es el modelo de la fuente de referencia a una señal de alimentación se presenta la figura 3.1, donde se observa la señal de salida para tres valores diferentes de regulación(0.01, 0.1 y 0.2) de línea y la señal de alimentación. Cada línea tiene al final un marcador para observar el valor final regulado y por lo tanto el modelado de la dependencia de la salida con respecto a ala alimentación. Las condiciones para esta simulación están dadas por los parámetros del modelo, los cuales se presentan en la tabla 3.1.

La dependencia de las señal de salida del modelo con respecto a la temperatura debe verificarse en las gráficas 3.2 y 3.3. Estas representan la compensación de voltaje de la fuente de referencia para un barrido de temperatura entre -50°C y 150°C . La figura 3.2 ilustra una compensación de primer orden para tres valores de coeficiente de temperatura mediante un arreglo lineal a trazos con un punto de inflexión en 50°C . La configuración de los parámetros

Parámetros	Valor	Valor	Valor
vref [V]	1.25	1.25	1.25
vov [V]	0.15	0.15	0.15
reglin [V/V]	0.01	0.1	0.2
tc [ppm]	100	100	100
comp [-]	1	1	1
npp [μV]	3000	3000	3000

Tabla 3.1: Valores de los Parámetros.

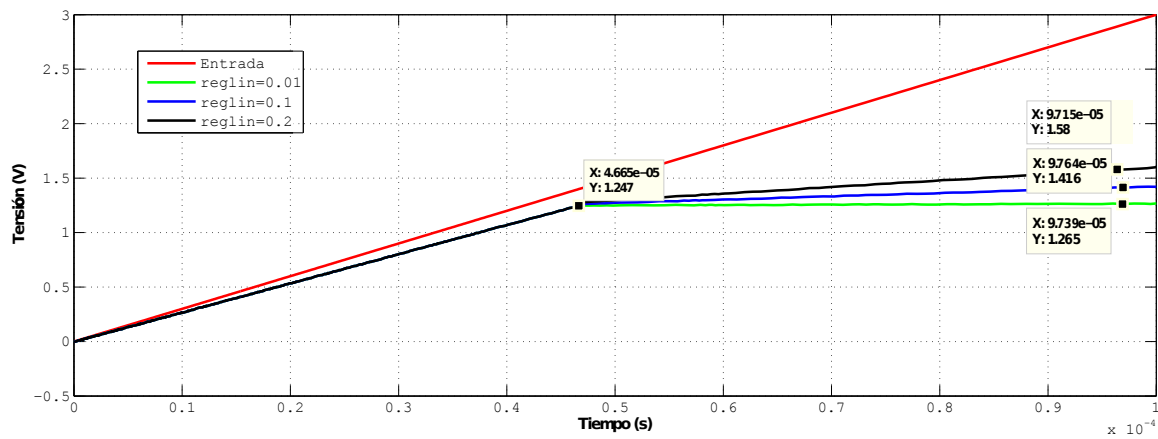


Figura 3.1: Alimentación VS Salida.

del modelo se muestran en la tabla 3.2.

Parámetros	Valor	Valor	Valor
vref [V]	1.25	1.25	1.25
vov [V]	0.15	0.15	0.15
reglin [V/V]	0.01	0.01	0.01
tc [ppm]	100	200	500
comp [-]	1 y 2	1 y 2	1 y 2
npp [μV]	3000	3000	3000

Tabla 3.2: Valores de los Parámetros.

De la misma forma la figura 3.3, ilustra el modelado de la compensación de tensión de segundo orden por medio del mismo arreglo lineal a trazos, pero esta vez el modelo tiene dos puntos de inflexión reduciendo la variación de la tensión con respecto a la temperatura.

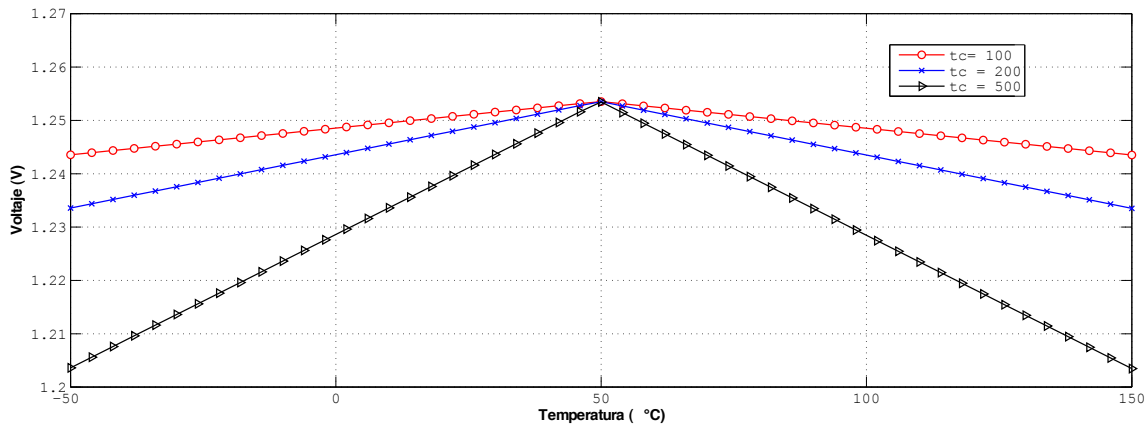


Figura 3.2: Variación del Coeficiente de Temperatura. $comp = 1$.

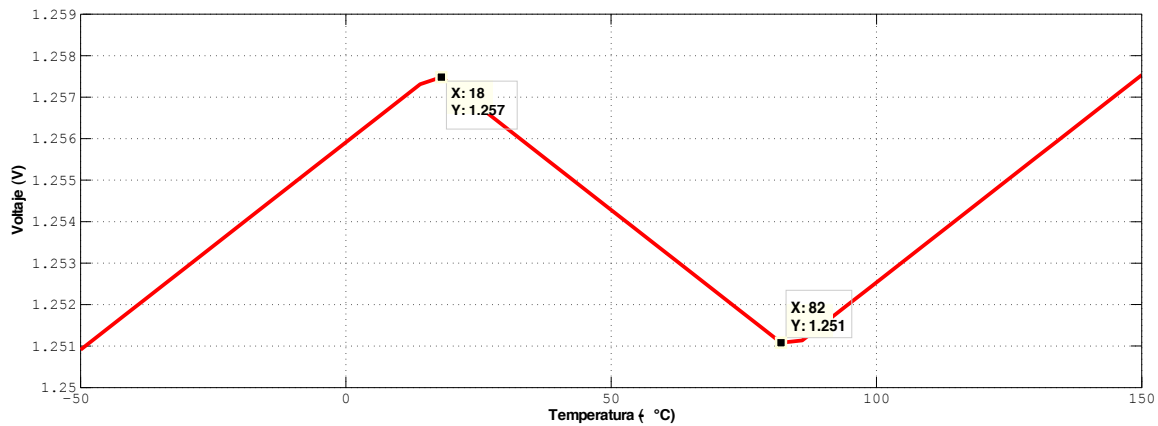


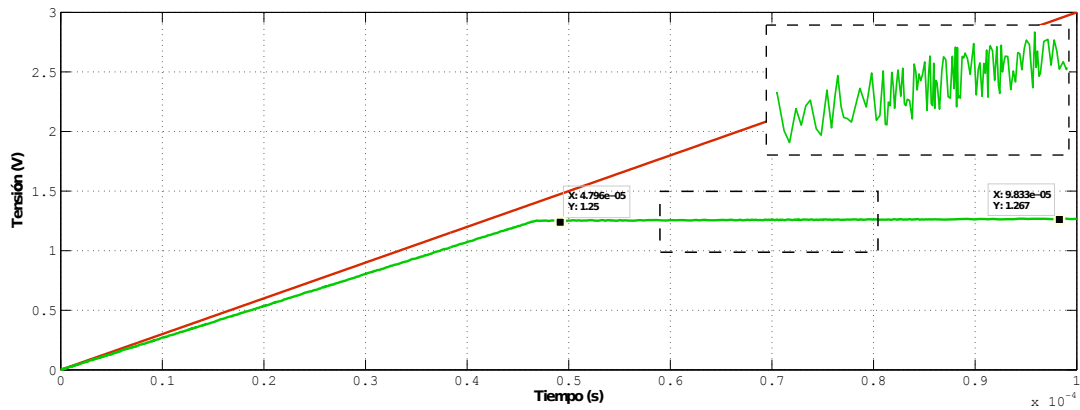
Figura 3.3: Variación del Voltaje en Relación con la Temperatura. $comp = 2$.

3.1.1. Ruido

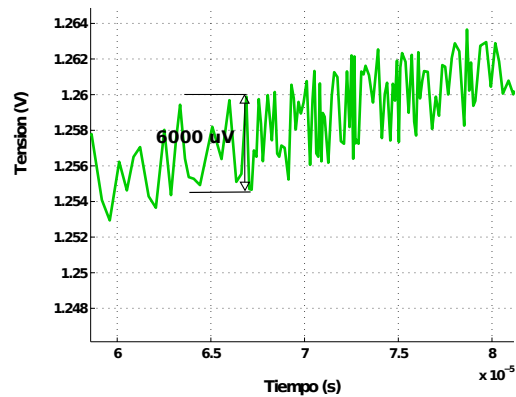
El modelado del ruido en este bloque corresponde a un aporte aleatorio de voltaje en el orden de los micro-Volts. Por ésta razón el análisis del comportamiento debe hacerse en el dominio del tiempo (figura 3.4). Los parámetros se listan en la tabla 3.3. La figura 3.4(a) es la salida de la fuente de referencia, en esta se hace un acercamiento que luego se presenta en la figura 3.4(b). Esta ultima deja ver que para los parámetros listados, el valor del ruido corresponde al esperado, habiendo puesto $3000 \mu V$ en el parámetro npp , teniendo como voltaje de ruido $6 mV_{pp}$.

Parámetros	Valor
vref [V]	1.25
vov [V]	0.15
reglin [V/V]	0.01
tc [ppm]	100
comp [-]	1
npp [μV]	3000

Tabla 3.3: Valores de los Parámetros para Ruido.



(a) Extracción de Ruido de la Fuente de Referencia.



(b) Ruido en el Dominio de el Tiempo.

Figura 3.4: Ruido en Fuente de Referencia.

3.2. AMPLIFICADOR OPERACIONAL

Análisis AC

Con los análisis de tipo AC, se pretende caracterizar el comportamiento del modelo en frecuencia, variando parámetros que tengan influencia en esta, tales como: la ganancia *gain*, el *CMRR*, *PSRR*, margen de fase (*PM*) y *GBW*.

Con la intención de comparar los datos de simulación con los datos esperados por el usuario, se hacen variaciones de el producto ganancia-ancho de banda *GBW* a un valor de ganancia específico y dejando los demás parámetros del modelo sin cambios. Los resultados de las simulaciones se muestran en las tablas 3.4 y 3.5.

Ganancia 40 dB							
GBW [Hz] Ingresado	GBW [Hz] Medido	Margen Fase Ingresado [°]	Margen Fase Medido [°]	UGF Medido [Hz]	% Error GBW	% Error Margen Fase	
1k	1.009k	60	88.51	1.044k	0.9	47.51	
100k	101.4k	60	69.67	96.5k	1.4	16.11	
10M	10.17M	60	67.9	9.623M	1.7	13.16	
100M	100.6M	60	67.85	95.84M	0.6	13.08	
1G	1.018G	60	67.96	972M	1.8	13.26	
Ganancia 120 dB							
GBW [Hz] Ingresado	GBW [Hz] Medido	Margen Fase Ingresado [°]	Margen Fase Medido [°]	UGF Medido [Hz]	% Error GBW	% Error Margen Fase	
1k	1.486k	60	87.92	1.044k	48.6	46.53	
100k	124.5k	60	69.67	96.5k	24.5	16.11	
10M	10.09M	60	67.24	9.623M	0.9	12.06	
100M	101.3M	60	67.19	95.85M	1.3	11.98	
1G	1.014G	60	67.31	972.1G	1.4	12.18	

Tabla 3.4: Variaciones *GBW* para Ganancias de 40 dB y 120 dB.

GBW 100k Hz						
Margen Fase Ingresado [°]	Margen Fase Medido [°]	GBW [Hz] Medido	UGF Medido [Hz]	% Error Margen Fase	% Error GBW	
20	40.81	101.5k	66.32k	104.05	1.5	
40	52.98	101.5k	81.9k	32.45	1.5	
60	69.06	101.5k	96.79k	15.1	1.5	
80	81.04	101.5k	101.6k	1.3	1.5	
90	84.46	101.5k	102.3k	-6.15	1.5	
GBW 10 MHz						
Margen Fase Ingresado [°]	Margen Fase Medido [°]	GBW [Hz] Medido	UGF Medido [Hz]	% Error Margen Fase	% Error GBW	
20	29.86	10.14M	4.973M	49.3	1.4	
40	47.29	10.14M	7.625M	18.22	1.4	
60	67.25	10.14M	9.504M	12.08	1.4	
80	80.67	10.14M	10.48M	0.83	1.4	
90	84.33	10.14M	10.58M	-6.3	1.4	

Tabla 3.5: Variaciones Margen de Fase para *GBW* de 100 kHz y 10MHz .

Diferentes valores de ganancia son mostrados en la figura 3.5, todos para un *GBW* de 10 MHz. De esta se observa que el modelo predice diferentes valores para los polos dominantes de cada curva, que todas las curvas tienen la misma frecuencia de corte y que a mayor ganancia

menor ancho de banda, lo que confirma que la aproximación matemática para el modelado funciona.

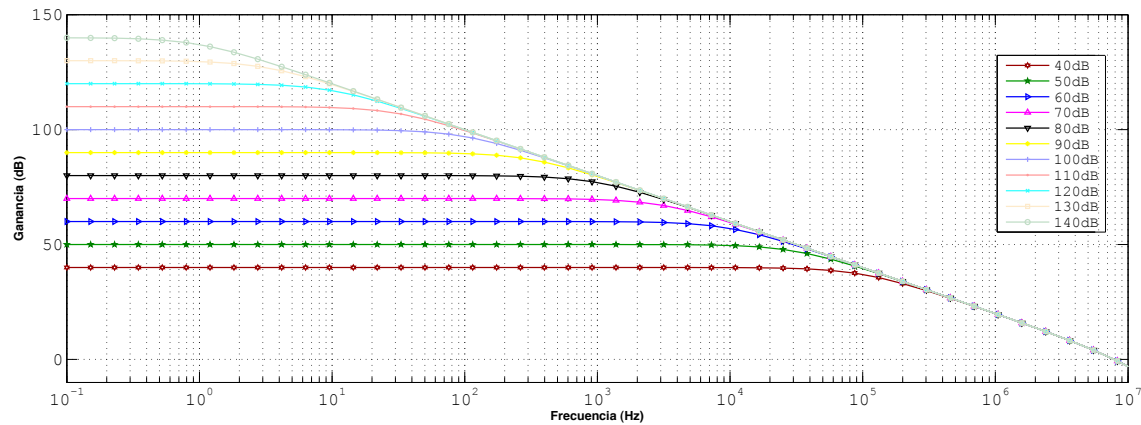


Figura 3.5: Variación de la Ganancia Para un GBW de 10 MHz.

La figura 3.6 muestra la salida del amplificador variando los mismos valores de ganancia pero esta vez con un GBW de 100 kHz. De esta configuración se observa que la curva azul con círculo pareciese no alcanzar su correspondiente valor de 140 dB, esto se explica porque el polo estimado está a frecuencias menores que las de la escala de la gráfica.

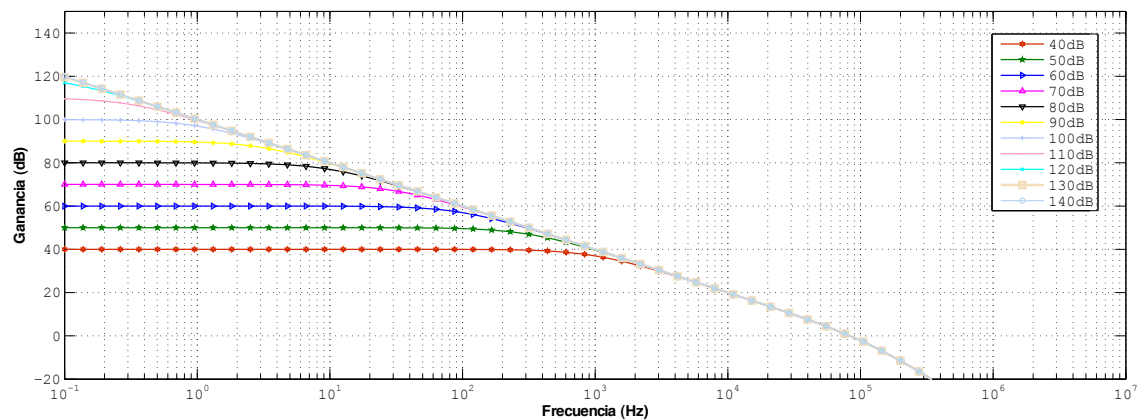


Figura 3.6: Variación de la Ganancia Para un GBW de 100 kHz.

Como se mencionó en el capítulo 2, tanto el CMRR como el PSRR son modelados mediante un filtro en frecuencia, para el caso del CMRR el cero dominante está directamente relacionado y ubicado en la frecuencia del polo 1 del sistema, lo que le da coherencia al modelo. Las figura 3.7 y 3.8 ilustran lo mencionado anteriormente para diferentes valores de atenuación a señales

en modo común, la primera para un GBW de 10 MHz y la segunda para un GBW de 100 kHz. Se ve que todas las señales que entren en modo común serán atenuadas en el valor especificado.

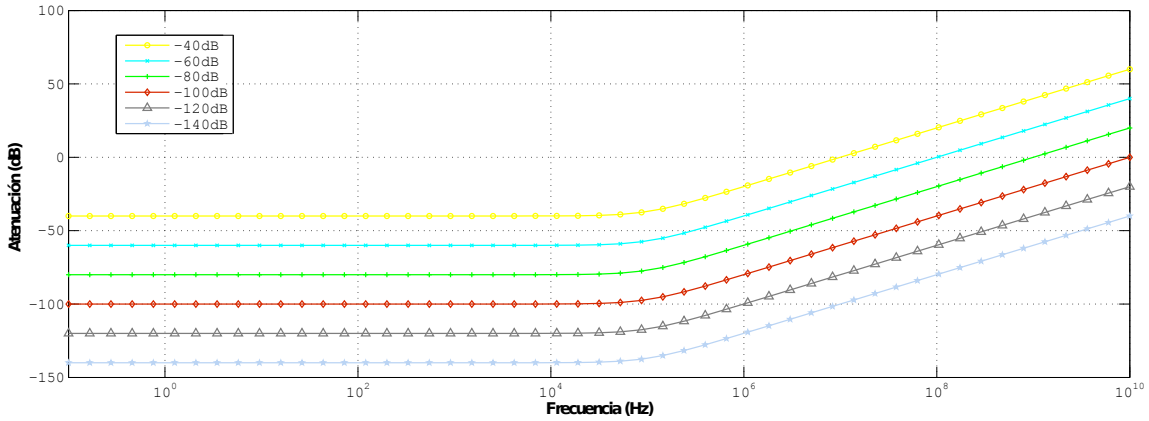


Figura 3.7: Valores de Atenuación Para un GBW de 10 MHz.

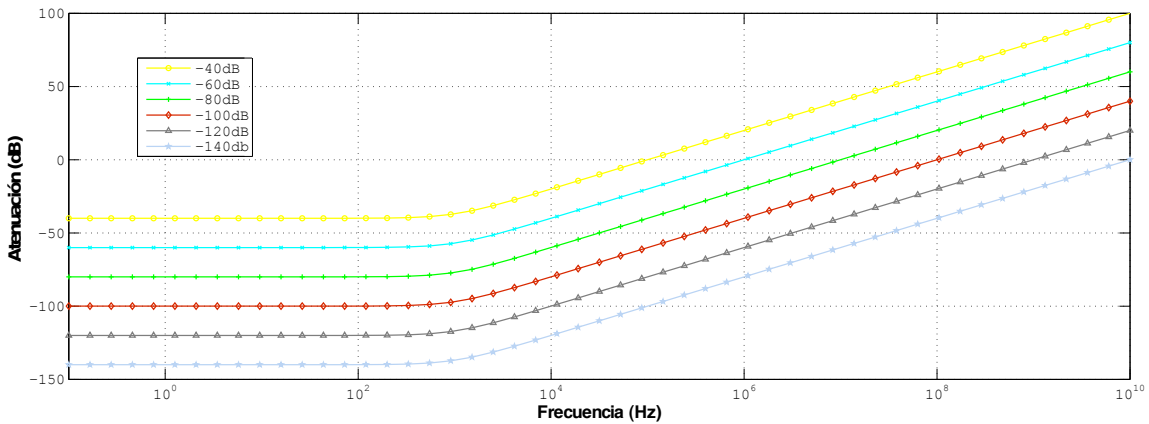


Figura 3.8: Valores de Atenuación Para un GBW de 100 kHz.

El modelado matemático del PSRR tiene un fundamento similar al del CMRR, pero el PSRR le da la libertad al usuario de asignar la frecuencia a la que empieza a disminuir el rechazo, las figuras 3.9 y 3.10 son tomadas para diferentes valores de rechazo empezando en -40 dB hasta llegar a -140 dB, con pasos de 20 dB. La figura 3.9 tiene una frecuencia f_{psr} de 1 kHz y por su parte la figura 3.10 tiene un valor de f_{psr} de 100 Hz. En ambas figuras se observa como el parámetro f_{psr} actúa como frecuencia límite al rechazo de las señales de la fuente, después de la frecuencia impuesta por este parámetro el rechazo disminuye.

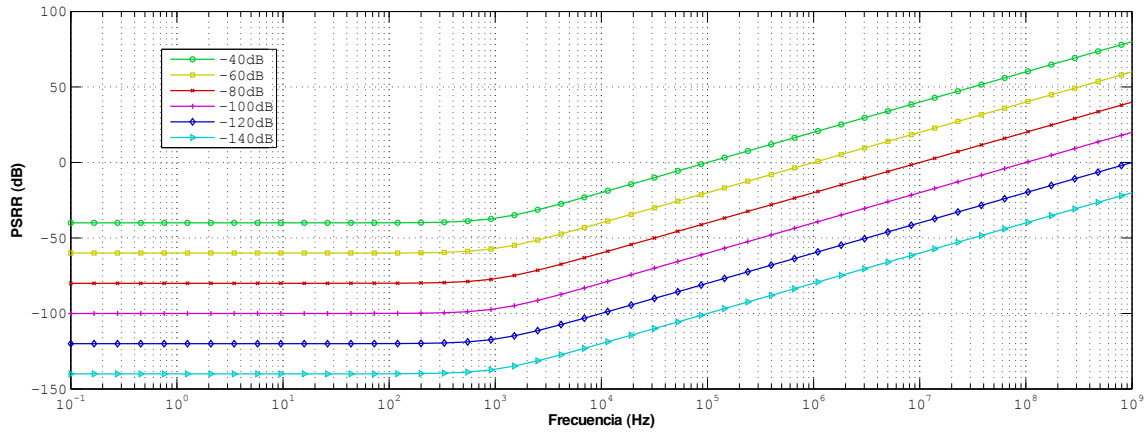


Figura 3.9: Valores de PSRR Para un fpr1 1 kHz.

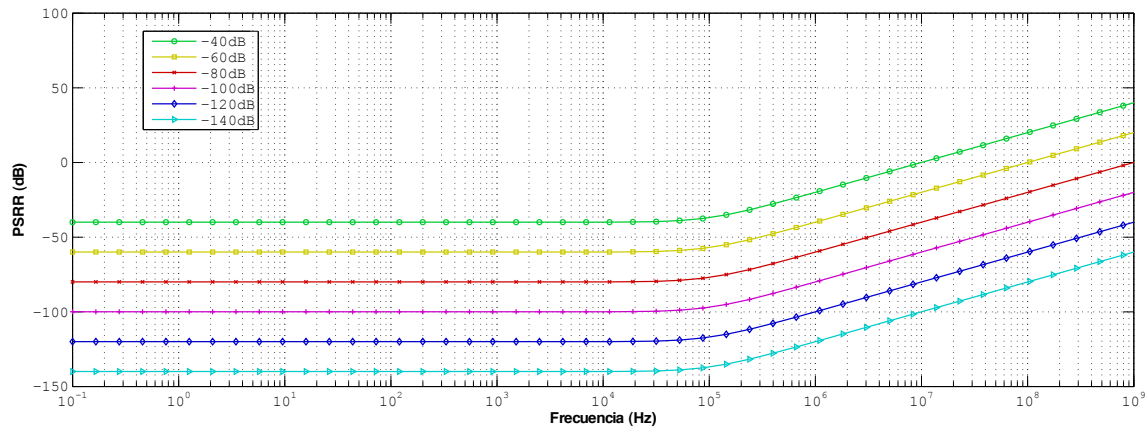


Figura 3.10: Valores de PSRR Para un fpr1 100 kHz.

Análisis Transitorio

El análisis transitorio permite ver las señales de interés en el amplificador en el dominio del tiempo. Así pues, ésta subsección presenta de forma detallada los resultados de simulaciones tras la variación de diferentes parámetros del modelo del amplificador operacional.

La figura 3.11 enfrenta una señal de entrada con amplitud de 10 mV a una frecuencia de 100 kHz que representa una señal dentro de la banda de amplificación del *Op Amp*, el parámetro *gain* se fijó en 20 dB, para observar una ganancia de 10 V/V. El amplificador se comporta tal cual lo esperado, observando que la señal azul de salida tiene una amplitud de 100 mV. Adicionalmente el parámetro *outbias* se fijó en 100 mV para darle un nivel de DC a

la señal de salida, observando de esta forma lo transversal de los parámetros en *verilog-a/MS*.

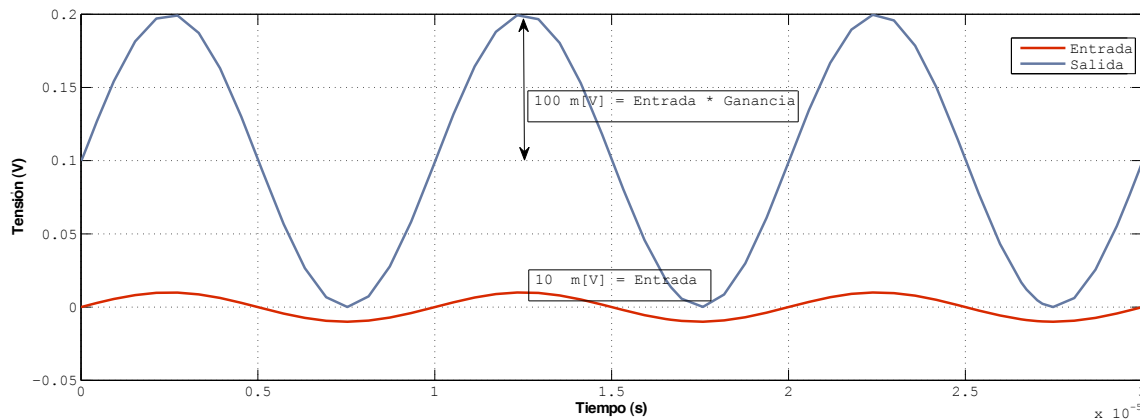


Figura 3.11: Ganancia de 20 dB en el tiempo, *offset* de 100 m[V] a frecuencia *in-band*.

El estudio del comportamiento del CMRR en el tiempo se realiza aplicando una señal en la terminal de modo común del modelo, de amplitud 1 V a 100 kHz (claramente una frecuencia al interior de la banda de rechazo). El parámetro CMRR está fijado en 60 dB. La figura 3.12 confronta la señal de entrada y la señal de salida. Ya que la señal de salida está atenuada por un factor de 1000, se separa la señal azul en la gráfica 3.13. De ésta gráfica se entiende que la señal de entrada en modo común sale del amplificador a una amplitud de 1 mV.

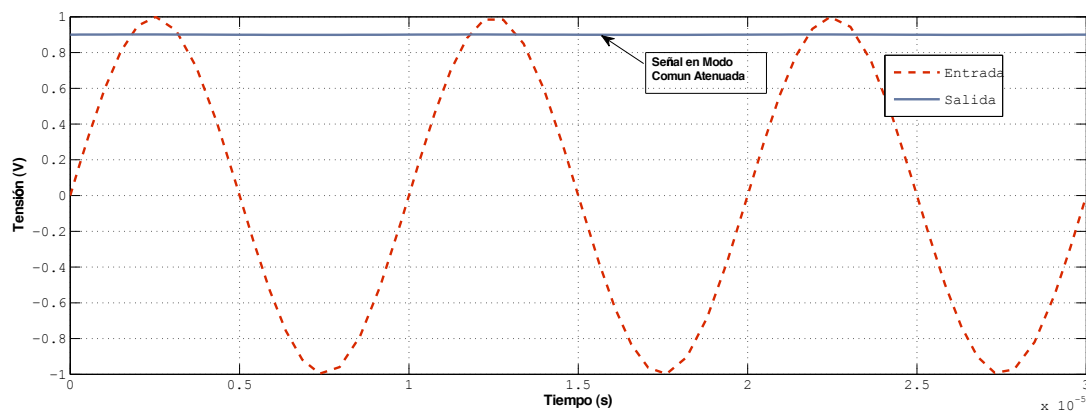


Figura 3.12: Señal con Rechazo 60dB a 100 kHz. Frecuencia *in-band*.

Para el caso del PSRR se realiza un procedimiento semejante al del CMRR. La señal de entrada está sobre la tensión de alimentación en DC, es por esto que la figura 3.14 muestra la señal de entrada con amplitud de 1 V sobre una tensión de 1.8 V. La señal de salida

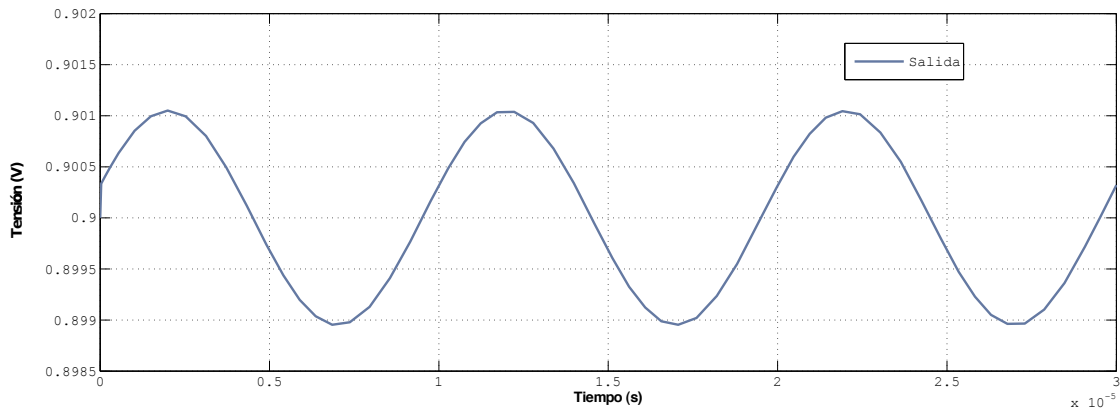


Figura 3.13: Salida con Rechazo 60dB a 100 kHz. Frecuencia *in-band*.

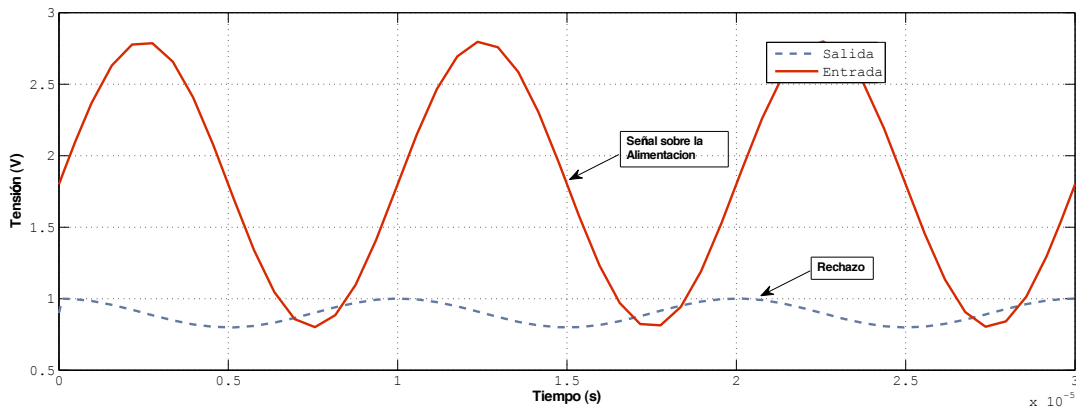


Figura 3.14: PSRR 60dB, Señales a 100 kHz. Frecuencia *in-band*.

está atenuada en 1000 V/V.

Para evaluar el resultado del modelado de la saturación (*clipping*), se presenta la figura 3.15. La señal de alimentación es de $V_{dd} = 1.8$ V, $V_{ss} = 0$ y los parámetros v_{clipp} y v_{clipn} establecidos en 0.05 V. Según esto, la señal de salida del *Op Amp* debería truncarse en 1.75 V. La figura muestra señales de salida a diferentes valores de ganancia que acercan esta señal a los límites establecidos por el usuario, y se observa que para estas condiciones si una señal de 1 mV es amplificada 60 dB o más, será truncada, pues supera los límites establecidos. Se concluye entonces, que el modelo de saturación cumple con lo esperado, ya que las señales que superen por exceso los límites se ven truncadas en el valor deseado.

Una de las no linealidades de los amplificadores operacionales que es modelada en este proyecto es el *slew rate*, los resultados de las simulaciones que comprueban su funcionamiento

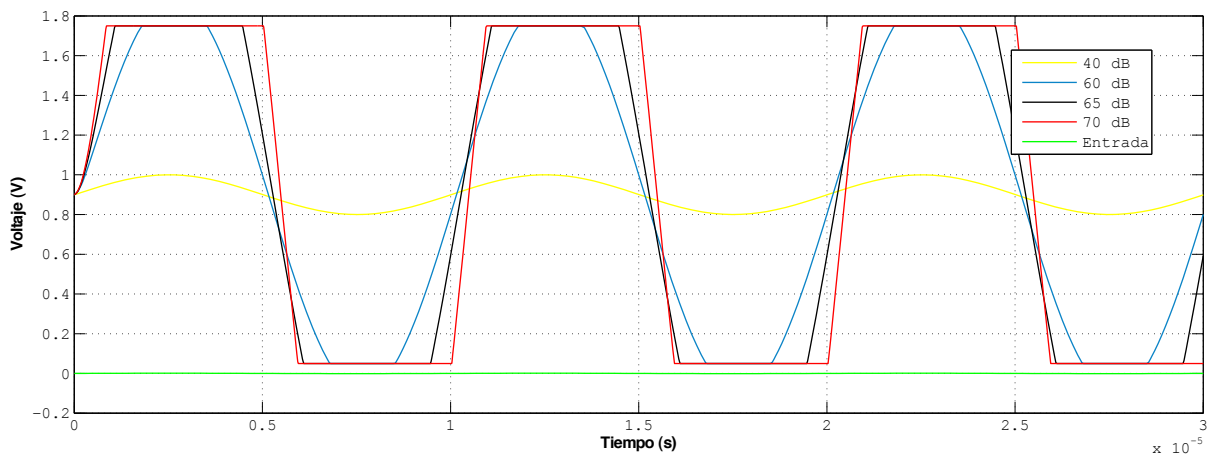


Figura 3.15: Modelado de la Distorsión por *Clipping*.

se muestran en la figura 3.16. En esta, se presenta una señal de salida variando los valores del parámetro *slewrates*. Se observa como la señal de salida es distorsionada a medida que los valores de *slewrates* varían. Así pues, la figura muestra una señal de 100 kHz, que se ve distorsionada para valores de *slew rate* menores a $0.0628 \text{ V}/\mu\text{s}$.

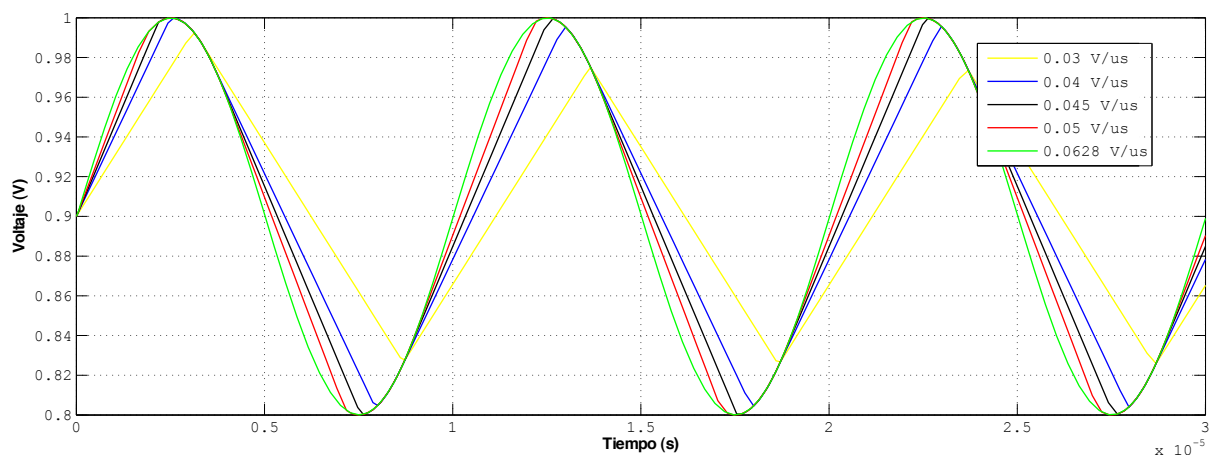


Figura 3.16: Modelado de la Distorsión por *Slew Rate*.

Configuración Ganancia Unitaria

Adicionalmente, se decidió escoger la configuración de la figura 3.2 conocida como ganancia unitaria o *buffer* para culminar la caracterización del modelo del amplificador operacional.

Esta configuración permite observar características de un amplificador mediante la respuesta al impulso.

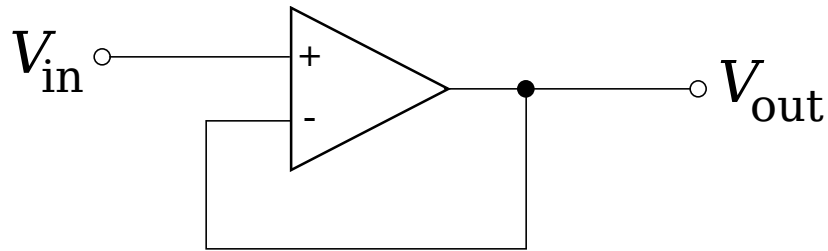


Figura 3.17: Configuración Ganancia Unitaria.

La figura 3.18, muestra el efecto de la variación del parámetro *gain* en la configuración *buffer*. Se observa que para ganancias pequeñas el amplificador no iguala el valor final ya que a ganancias pequeñas el amplificador pierde su característica "ideal" de ganancia infinita. Por otro lado la figura 3.19, permite ver el comportamiento del modelo del *Op Amp* para variaciones del margen de fase.

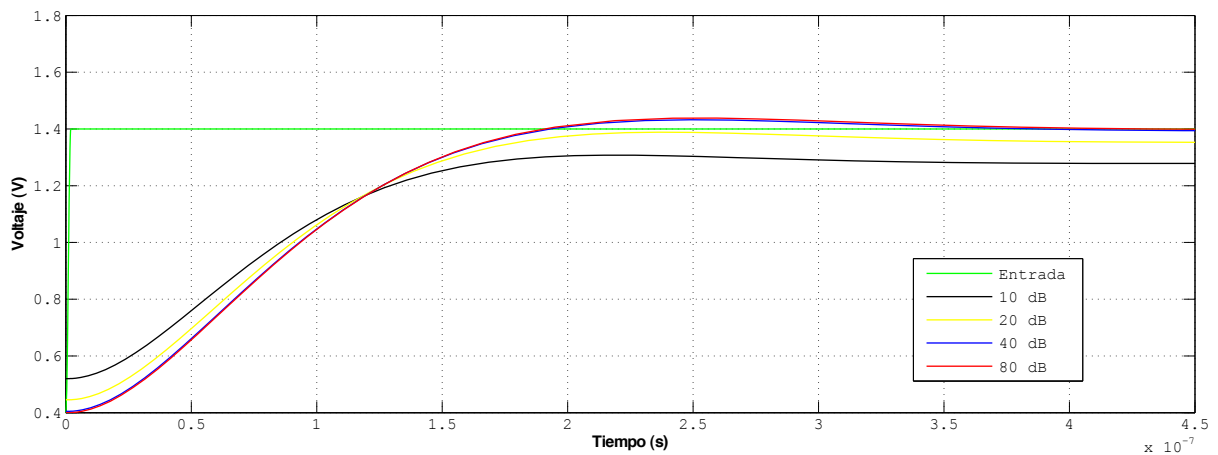


Figura 3.18: Análisis de la Variación de Ganancias en Configuración *Buffer*.

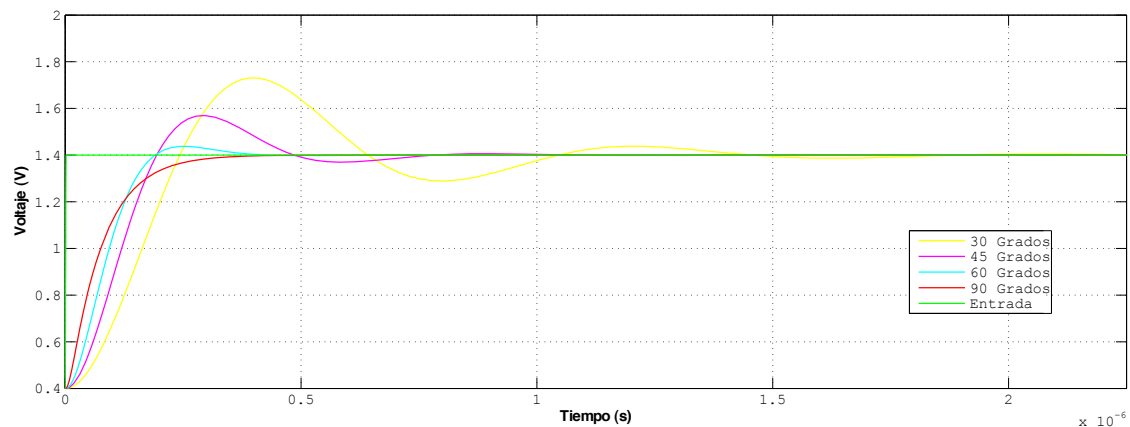


Figura 3.19: Análisis de Estabilidad Por Variación de Margen de Fase en Configuración *Buffer*.

3.2.1. Ruido

Con la intención de adquirir experiencia en el modelado de las características de diversos sistemas y de poder dar solución a la necesidades de los usuarios entregándoles cada vez más herramientas en los bloques creados y alcanzar más experiencia en el manejo del lenguaje, la creación del ruido en este proyecto se hizo de dos formas. Para la fuente de referencia de voltaje, el ruido fue modelado haciendo una contribución de tensión a la salida, utilizando la función *\$random* del lenguaje para crear pequeños valores aleatorios de tensión. Por su parte, el bloque *Op Amp* modela esta perturbación haciendo uso de funciones exclusivas de ruido con las que cuenta el lenguaje.

Para tener una idea del comportamiento del ruido del amplificador se necesita ver el cambio del ruido integrado en banda definido por el usuario mediante el parámetro *vrms*. Diferentes valores dentro del rango de 10 nV hasta 10 mV son presentados en la figura 3.20, se observa que a medida que aumenta *vrms* el aporte del ruido *flicker* al circuito se vuelve más significativo llegando a estar en el orden de 15 uV. Por otro lado, el ruido térmico se analiza variando el parámetro *en* desde 10 nV hasta 1 uV. Ya que el ruido térmico es relevante a frecuencias altas comparado con el *flicker*, la figura 3.21 corresponde a un *zoom* al rango de frecuencias después de los 100 Hz hasta 100 kHz.

Según la figura 3.21, variaciones en el parámetro *en* si corresponde a una variación en el piso de ruido del aporte total de esta señal, confirmando que el modelado realizado se aproxima al comportamiento real de una fuente de ruido.

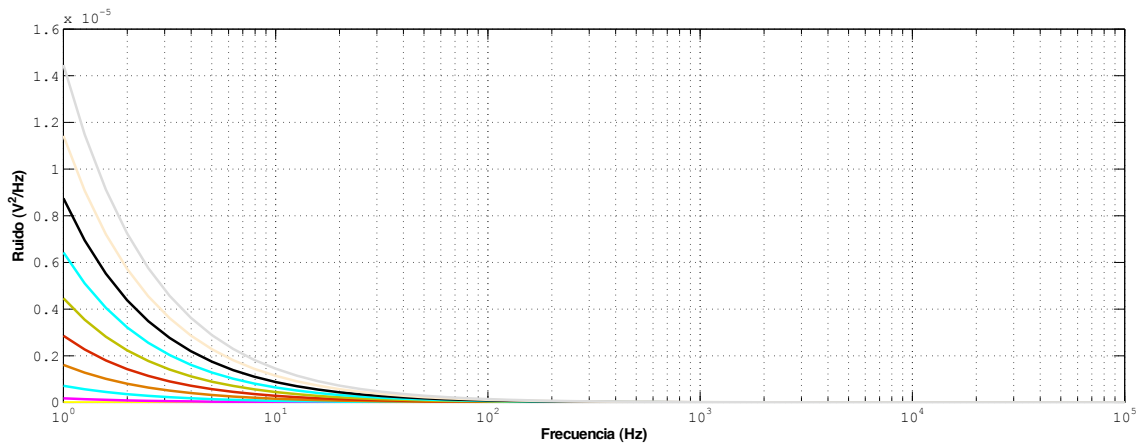


Figura 3.20: Variación del Parámetro Vrms.

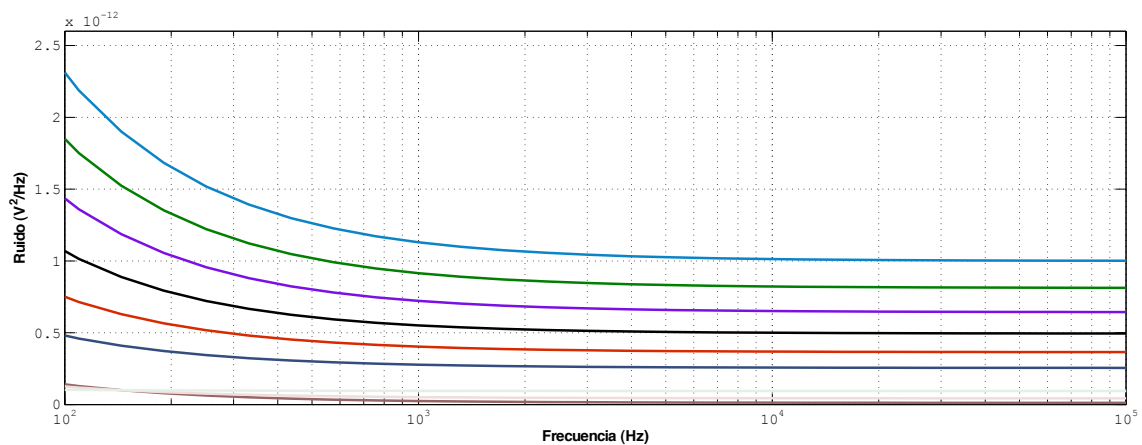


Figura 3.21: Variación del Piso Térmico.

3.3. VALIDACIÓN MEDIANTE LA APLICACIÓN EN UN SISTEMA DE POTENCIA

Un *LDO* es un circuito regulador de voltaje lineal, la ubicación típica de estos circuitos está entre el suministro de energía y un circuito encargado de desarrollar una tarea específica. Idealmente, se quiere que provea un voltaje de salida invariable a cambios en la tensión de entrada y en la carga [23]. Sus características principales son: la baja disipación de calor y una alta eficiencia. Su estructura tiene como principales elementos un amplificador diferencial, que es el encargado de generar una tensión de control que mantenga al circuito operando en regulación; una fuente de referencia cuyo objetivo es situar una señal de voltaje lo mas estable

posible como patrón de comparación en el amplificador y una etapa de potencia que suministra la tensión y corriente requeridas por la carga. Esto sin duda, lo convierte en una elección ideal para corroborar el funcionamiento de los modelos desarrollados en este proyecto en un sistema que incluya diferentes elementos de circuito.

La estructura general de un *LDO* se muestra en la figura 3.22, en esta se observan los bloques catalogados como cruciales; V_{REF} , E_A y EP . El objetivo planteado es diseñar un regulador lineal tipo *LDO* dadas las especificaciones de la tabla 3.6, variando de manera controlada los parámetros de los bloques modelados en *verilog-A/MS*. El bloque del amplificador operacional será utilizado en la etapa de control, la fuente proveerá una señal de referencia que será comparada por el amplificador. Por su parte el transistor de potencia y los elementos integrados (C_{in} , R_1 , R_2 y C_b) utilizados son de tecnología AMI¹ 0.6 μm del *kit* de diseño de la NCSU².

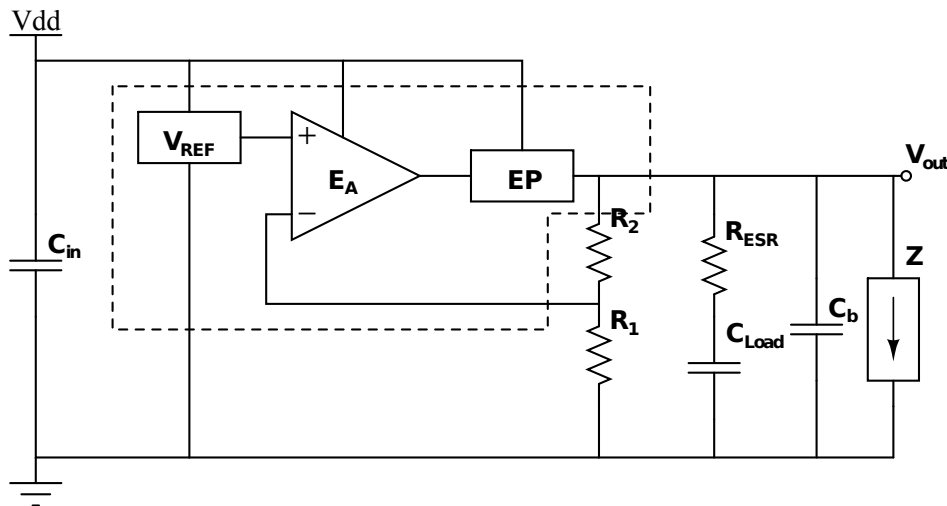


Figura 3.22: Regulador LDO [23].

Los valores para los parámetros obtenidos en el diseño se resumen en la tabla 3.7.

Las figuras 3.23 y 3.24 son las regulaciones de línea y de carga respectivamente, se observa que para variaciones en la tensión de alimentación, la salida apenas fluctúa en un rango de 2.5 mV, mientras que en la regulación de carga se observa una variación de 100 μV , obteniendo los mínimos valores posibles de acuerdo a lo establecido como especificaciones.

En las figuras 3.25 y 3.26 se observa la salida del *LDO*, en la primera se presenta junto con la salida del modelo de la fuente de referencia, y en la segunda el voltaje a una escala más representativa. De este par de gráficas se ve que la regulación se mantiene un voltaje de

¹Siglas en inglés: American Microsystems, Inc (ahora ON Semiconductor).

²Siglas en inglés: North Carolina State University.

Especificaciones	Valor
Vout	2.5 V
Iout	0-100mA
Dropout	200 mV
Regulación Línea	La menor Posible
Regulación de Carga	La menor Posible

Tabla 3.6: Especificaciones de Diseño

Parámetros Fuente de Referencia	Valor	Parámetros Amplificador Operacional	Valor
vref [V]	1.25	gain [dB]	80
vov [V]	0.15	GBW [Hz]	10M
reglin [V/V]	0.001	PM [°]	60
tc [ppm]	10	slew rate [V/us]	10
Comp [-]	1		
npp [uV]	0.001		

Tabla 3.7: Parámetros de Diseño para los Bloques Modelados.

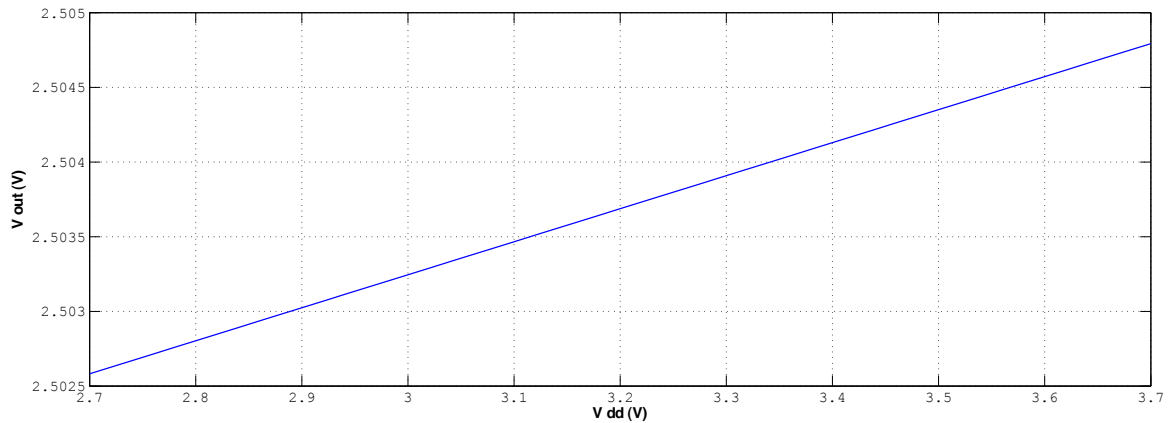


Figura 3.23: Regulación de Línea.

2.5 V, logrando así lo propuesto en las especificaciones de diseño.

También se concluye que los modelos creados en este proyecto son de alta flexibilidad, parametrizables y lo suficientemente robustos para su aplicación en diferentes circuitos analógicos.

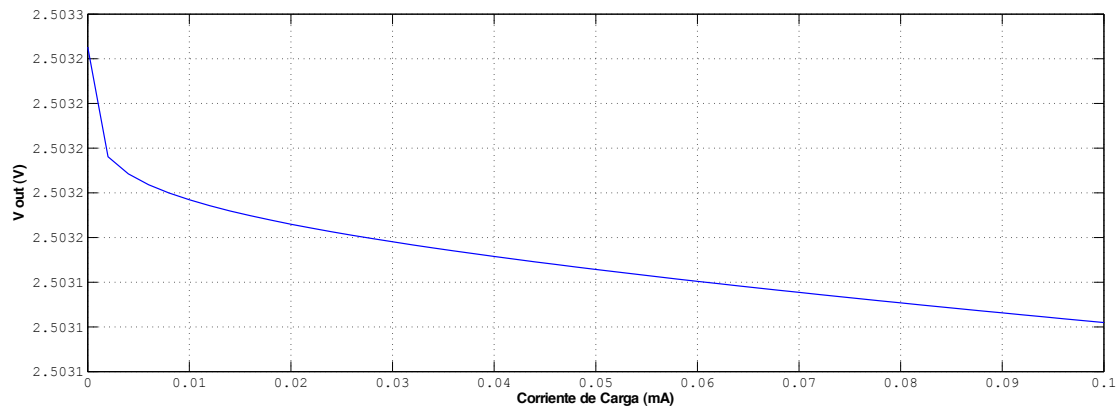


Figura 3.24: Regulación de Carga.

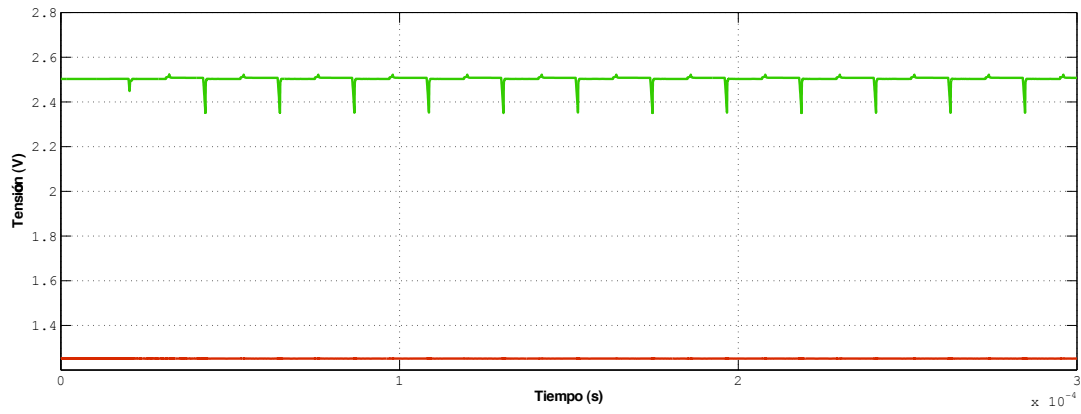


Figura 3.25: Salida del Regulador.

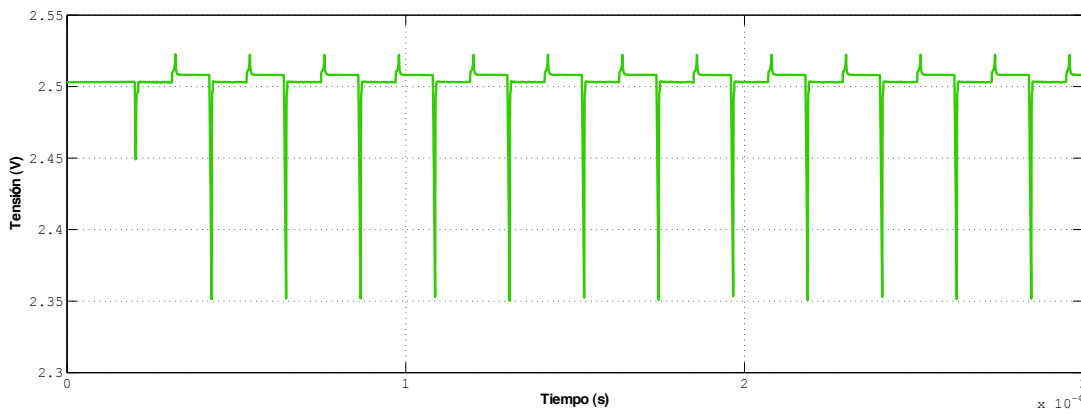


Figura 3.26: Salida del Regulador.

Capítulo 4

OBSERVACIONES Y CONCLUSIONES

4.1. CONCLUSIONES

Este trabajo afronta el reto de promover el uso de lenguajes de descripción de *hardware* analógico y de señal mixta en la Universidad Industrial de Santander, para poder tener diferentes herramientas a la hora de solucionar problemas en el área de diseño de circuitos integrados analógicos. Con esto se quiere innovar dentro de la región al aprovechar la versatilidad que ofrecen los estándares mundiales en el área de modelado de circuitos. Se modelaron dos circuitos trascendentales; el amplificador operacional de amplio uso académico e industrial por décadas y una fuente de referencia que es indispensable para casi cualquier circuito electrónico actual.

Luego de analizar durante el capítulo 1 la documentación disponible sobre el modelado de circuitos analógicos y de señal mixta, se decidió adoptar *verilog-A/MS* como el lenguaje óptimo para la realización de este proyecto. Este, cumple con las características buscadas al inicio del planteamiento del proyecto; el lenguaje debía ser un estándar de la industria mundial en el modelado de circuitos integrados analógicos y de señal mixta; debía tener documentación abierta que permitiera el auto-aprendizaje del mismo; debía compilar tanto en *suites* de diseño top de la industria como en herramientas libres o que tuvieran versión de prueba; debía aportar el factor de innovación a la región.

El modelo del amplificador operacional cuenta con más de 15 parámetros totalmente modificables al gusto del diseñador, con lo que se le garantiza flexibilidad para que pueda tomar decisiones basadas en resultados verídicos comprobados a lo largo de este proyecto. Por otro lado *Verilog-A/MS* ofrece portabilidad en los modelos permitiendo ser analizados en diferentes simuladores además de la versatilidad de poder ser integrado en simulaciones de circuitos

digitales. En un aspecto más técnico, el amplificador operacional pone a orden del usuario un modelado serio del comportamiento de especificaciones importantes a tener en cuenta en el diseño de circuitos integrados, como los son el *GBW*, margen de fase, ruido térmico, ruido *flicker*, *PSRR*, *CMRR*, *offset* y *slew rate*.

La fuente de referencia de voltaje tiene 6 parámetros igualmente configurables, que permitirán manipular variables de diseño importantes en sistemas más complejos. Este proyecto ofrece al usuario un modelo que esta en la capacidad de ofrecer tensiones de referencia de comportamiento predecible y manipulable de regulación de linea, ruido y compensación de primer y segundo orden con respecto a la temperatura.

La inclusión de ruido resulta fundamental en el modelado de cualquier sistema o circuito electrónico en la actualidad. Es por esto que durante el proceso de este proyecto se modelaron dos diferentes formas de incluir ruido en los bloques. Para el caso del amplificador operacional, el ruido se basó en las funciones propias de *verilog-A/MS*, obteniendo resultados como los vistos en el capítulo 3 que son producto del desarrollo del lenguaje. Para el modelo de la fuente, el usuario decide que cantidad de ruido quiere adicionar a su señal de salida mediante la contribución de una señal de tension. Estos se hicieron con el fin de aportar nuevas formas de modelado de ruido y así contribuir con generar un impacto de desarrollo en el área.

Finalmente los modelos son validados en un sistema de potencia. De esto se concluye que los modelos pueden ser utilizados sin ninguna discriminación en esquemas que contengan circuitos a nivel transistor, afirmando una vez mas las ventajas del lenguaje.

Se anexa un tutorial donde se explica paso a paso la creación de un modelo, simulación del mismo y explicación de cada una de sus partes, permitiendo así difundir al interior de la Universidad Industrial de Santander los conocimientos adquiridos a lo largo de la realización de este proyecto.

4.2. OSBERVACIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS

Una de las ventajas mencionadas de *Verilog-A/MS* es su portabilidad, permitiendo a los modelos la formación de librerías de diseño muy completas. Por otro lado es posible migrar modelos en lenguaje *verilog-A/MS* a modelos tipo *SPICE*. Se recomienda analizar las ventajas de esta migración con la intención de crear librerías para la Universidad Industrial de Santander y su implementación en entornos de diseño como *HSPICE* y *OrCAD*.

El tutorial podría hacer parte de una clase de laboratorio de los cursos de dispositivos electrónicos, fundamentos de circuitos integrados o diseño de circuitos analógicos del pènsum de la carrera de ingeniería electrónica de la Universidad Industrial de Santander .

La interacción entre los bloques modelados y los demás elementos de circuitos, resultó uno de los problemas más relevantes en el desarrollo del presente proyecto. Analizar diferentes tipos de interacción a los propuestos en este documentos sería útil para la continuación del área de modelado de circuitos de la universidad.

Bibliografía

- [1] Kenneth S. Kundert & Olaf. Zinke, *The Designer's Guide to Verilog AMS*. Kluwer Academic Publishers, 2004.
- [2] I. Miller and T. Cassagnes, "VERILOG-A AND VERILOG-AMS PROVIDES A NEW DIMENSSION IN MODELING AND SIMULATION," 2000.
- [3] Dan. Fitzpatrick & Ira. Miller, *Analog Behavioral Modeling With The Verilog-A Language*. Kluwer Academic Publishers, 2003.
- [4] Gregory D. Peterson & Darrell A. Teegarden, *The System Designer's Guide to VHDL-AMS*. Morgan Kaufmann Publishers, 2006.
- [5] E. Christen, K. Bakalar, A. Dewey, and E. Moser, "Analog and Mixed-Signal Modeling Using the VHDL-AMS Language," in *36th Design Automation Conference*, 1999, pp. 3–11.
- [6] *Verilog-A Language Reference Manual*. Open Verilog International, Los Gatos, CA, Version 1.0, 1996.
- [7] Accellera Systems Initiative. (2014) About us. [Internet]. Visite: <http://www.accellera.org/about/>
- [8] ——. (2014) Systemc. [Internet]. Visite: <http://www.accellera.org/community/systemc/>
- [9] MathWorks. (2014) Hdl coder, generate verilog and vhdl code for fpga and asic designs. [Internet]. Visite: <http://www.mathworks.com/products/hdl-coder/>
- [10] *Verilog-AMS Language Reference Manual*. Accellera Organization, Inc. Napa, CA, Version 2.3, 2008.
- [11] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw Hill, 2001.
- [12] Rasoul. Dehghani, *Design of CMOS Operational Amplifiers*. Artech House, 2013.

-
- [13] Santiram. Kal, *BASIC ELECTRONICS: Devices, Circuits and IT Fundamentals*. Prentice Hall, 2002.
- [14] J. C. M. Ardila, “Diseño de una fuente de voltaje de referencia de bandgap integrada en tecnología CMOS,” Tesis de pregrado de la Universidad Industrial de Santander, 2007.
- [15] J. J. S. Noval, “Aplicación de la Programación Geométrica en la Optimización del Diseño de una Fuente de Voltaje de Referencia,” Tesis de pregrado de la Universidad Industrial de Santander, 2009.
- [16] J. J. C. Castellanos, “Diseño de una Fuente de Voltaje de Referencia de Bandgap Menor a un Voltio Integrada en Tecnología CMOS,” Tesis de pregrado de la Universidad Industrial de Santander, 2006.
- [17] Inaoe. (2014) Acondicionamiento de sensores resistivos y capacitivos. [Internet]. Visite: http://www-elec.inaoep.mx/~materesa/pdf/acon_12/tema4_aco_12.pdf
- [18] Maxim Integrated. (2012) Understanding voltage-reference temperature drift. [Internet]. Visite: <http://www.maximintegrated.com/app-notes/index.mvp/id/4419>
- [19] Chi-Wah Kok & Wing-Shan Tam, *CMOS Voltage References An Analytical and Practical Perspective*. John Wiley and Sons, 2012.
- [20] José María Drake Moyano. (2014) Referencias de tensión y reguladores de tensión. [Internet]. Visite: http://www.ctr.unican.es/asignaturas/instrumentacion_5_IT/IEC.6.pdf
- [21] Art Zirger. (2014) Random offset in cmos ic design. [Internet]. Visite: http://ecee.colorado.edu/~ecen4827/lectures/Random.Offset_CMOS_IC_Design_CU_Lecture_Art_Zirger.pdf
- [22] T. Instruments, “Chapter 10 Op Amp Noise Theory and Applications, Tech. Rep. Literature number: SLOA082, 2008.
- [23] L. C. G. Lázaro, “Diseño de un Regulador LDO Integrable en Tecnología CMOS,” Tesis de pregrado de la Universidad Industrial de Santander, 2008.

Apéndice A

Código de los Modelos

Fuente de Referencia de Voltaje

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

module voltage_reference(dd,ref);

inout dd,ref;
electrical dd,ref;

parameter real vref = 1.25 from [0.5:10];
parameter real vov = 0.15 from [0.15:1];
parameter real reglin = 1e-3 from [0:1];
parameter real tc = 10 from [0:1000];
parameter integer comp = 1 from [1:2];
parameter npp = 30 from [0:1e4];

real vdiff;
real vrl;
real slope;
real noisepp;
real vref2nd;
```

```

analog begin

vdiff = V(dd)-vref;
vrl = vdiff*reglin;
slope = vref/(vref+vov);
noisepp = 1e-6*npp;
vref2nd = vref-1e-6*tc*34;

if (comp == 1) // 1st order compensation
begin
if (V(dd) < (vref + vov) )
begin
V(ref) <+ slope*V(dd)+1e-3*noisepp*($random % 1000);
V(ref) <+ -1e-6*abs(($temperature-323))*tc;
end
else
begin
V(ref) <+ vref+vrl+1e-3*noisepp*($random % 1000);
V(ref) <+ -1e-6*abs(($temperature-323))*tc;
end
end
else
begin
if ($temperature < 357)
begin
if (V(dd) < (vref + vov))
begin
V(ref) <+ slope*V(dd)+1e-3*noisepp*($random % 1000);
V(ref) <+ -1e-6*abs(($temperature-290))*tc;
end
else
begin
V(ref) <+ vref+vrl+1e-3*noisepp*($random % 1000);
V(ref) <+ -1e-6*abs(($temperature-290))*tc;
end
end
end

```

```
end
else
begin
if (V(dd) < (vref + vov))
begin
          V(ref) <+ slope*V(dd)+1e-3*noisepp*($random % 1000);
          V(ref) <+ 1e-6*abs((357-$temperature))*tc;
        end
        begin
          V(ref) <+ vref2nd+vrl+1e-3*noisepp*($random % 1000);
          V(ref) <+ 1e-6*abs((357-$temperature))*tc;
        end
      end
end
end
endmodule
```

Amplificador Operacional

```
'include "constants.vams"
'include "disciplines.vams"
'define PI 3.14159265358979323846
module opamp(pos, neg, out, ss, dd, cm, psr);

inout pos, neg, out, ss, dd, cm, psr;
electrical pos, neg, out, ss, dd, cm, psr, nodesr, nodeclip;

//MAIN AC PARAMETERS

parameter real gain=80;
parameter real GBW=10M;
parameter real PM=60;

real p1;
real Ap1;
real p2;
real Ap2;
real decs;

// BIAS AND OFFSET PARAMETERS
parameter real outbias=0.9;
parameter real offset=50u;
integer randnumber;
integer signumber;

// CMRR PARAMETERS
parameter real CMRR=100;
real acm;

// PSRR PARAMETERS
parameter real PSRR=60;
parameter real fpsr=1k;
```

```
real apsr;

// NOISE PARAMETERS
parameter real npb=1k;
parameter real vrms=10u;
parameter real en=10n;
parameter real fmin=1;
real eq;
real f3dB;
real flicker_power;

// SLEW-RATE PARAMETERS
parameter real slewrate=1;
real sr;

// CLIPPING PARAMETERS
parameter real vclipp=50m from [0:500m];
parameter real vclipn=50m from [0:500m];
real clipp;
real clipn;

// CURRENT LIMITING PARAMETERS
real imax;

// OUTPUT IMPEDANCE AND LOAD CAPACITANCE PARAMETERS
parameter real rout=100k;
parameter real cout=10p;

analog begin

// POLE PARAMETERS

p2=GBW/(pow(10,(1-(PM/45)))); //en Hz
Ap2=20*log(GBW/p2); // en dB
decs=(gain-Ap2)/20;
p1=p2/(pow(10,decs));
```

```

Ap1= ((GBW/(pow(10,(1-(PM/45))))))
/(pow(10,(((20*log(gain)) -(20*log((GBW)
/(GBW/(pow(10,(1-(PM/45))))))))/(20)))));
// CMRR CALCULATIONS
acm = (pow(10,gain/20))/(pow(10,CMRR/20));
// PSRR CALCULATIONS
apsr = 1/(pow(10,PSRR/20));
// NOISE CALCULATIONS
eq = vrms/sqrt(npb);
f3dB = (((pow(eq,2)/pow(en,2))-1)*(npb-ffmin))/(ln(npb/ffmin));
flicker_power = f3dB*pow(en,2);

// SIGNAL DEFINITION

// Offset insertion
signumber = 1; // Sign of the offset signal
V(nodesr) <+ ( pow(10,(gain/20))*offset*signumber+ outbias );

// Transfer function for differential signals (includes transient response)
V(nodesr) <+ (pow(10,(gain/20)))
*laplace_np(V(pos, neg),{1},
{ -2*PI*((GBW/(pow(10,(1-(PM/45)))))/(pow(10,(((gain)
-(20*log((GBW)/(GBW/(pow(10,(1-(PM/45))))))))/(20))))),
0, (-2*PI*(GBW/(pow(10,(1-(PM/45))))),0 , -1/(2*PI*rout*cout),0});

// Transfer function for common mode signals
V(nodesr) <+ ( acm/(pow(10,gain/20) ))
*laplace_zd(V(cm),{ -2*PI*
((GBW/(pow(10,(1-(PM/45)))))/(pow(10,(((gain)
-(20*log((GBW)/(GBW/(pow(10,(1-(PM/45))))))))/(20))))), 0 },{1});

// Transfer function for supply-coupled signals
V(nodesr) <+ apsr*laplace_zd(V(psr,dd),{ -2*PI*fpsr, 0 },{1});

// Thermal and flicker noise contributions
V(nodesr) <+ flicker_noise(flicker_power,1,"FlickerNoise")

```

```
+white_noise(pow(en,2),"WhiteNoise");

// Slew-rate insertion
sr = 1e6*slewrates;
V(nodeclip) <+ slew(V(nodesr),sr,-1*sr);

// Clipping insertion
clipp = V(dd)-vclipp;
clipn = V(ss)+vclipn;

if (V(nodeclip) > clipp || outbias > clipp)
    V(out) <+ clipp;
else if (V(nodeclip) < clipn || outbias < clipn)
    V(out) <+ clipn;
else
    V(out) <+ V(nodeclip);

@(initial_step)

$strobe

("Polo 1\t\t",p1,"Hz",
"\n", "Polo 2\t\t",p2,"Hz",
"\n", "GBW\t\t", GBW,"Hz",
"\n", "Common Mode Gain\t\t",acm,"V/V",
"\n", "Random Number\t\t",randnumber,"N/A",
"\n", "Offset Sign\t\t",signnumber,"N/A",
"\n\n\n\n\n") ;

@(final_step)
end
endmodule
```