

UNIVERSIDAD DE GRANADA

FACULTAD DE CIENCIAS



Departamento de Electrónica y Tecnología de Computadores

Simulación de circuitos basada en la implementación de
modelos avanzados de dispositivos y sensores
electrónicos en Verilog-A.

Tesis Doctoral

Andrés Roldán Aranda

Granada 2012

Editor: Editorial de la Universidad de Granada
Autor: Andrés María Roldán Aranda
D.L.: GR 2233-2012
ISBN: 978-84-9028-123-9

Simulación de circuitos basada en la implementación de modelos avanzados
de dispositivos y sensores electrónicos en Verilog-A.

Andrés Roldán Aranda

Tesis Doctoral

UNIVERSIDAD DE GRANADA
FACULTAD DE CIENCIAS

Departamento de Electrónica y Tecnología de Computadores

Granada 2012

D. Juan B. Roldán Aranda, Profesor Titular del Departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada y **D. Càndid Reig Escrivá**, Profesor Titular del Departamento de Ingeniería Electrónica de la Universidad de Valencia,

CERTIFICAN:

Que el trabajo de investigación recogido en la presente memoria titulada: “*Simulación de circuitos basada en la implementación de modelos avanzados de dispositivos y sensores electrónicos en Verilog-A*”, y presentada por **D. Andrés Roldán Aranda** para optar al grado de Doctor por la Universidad de Granada ha sido realizado bajo nuestra dirección en el Departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada.

Granada, 10 enero de 2012.

Dr. D. Juan B. Roldán Aranda
Profesor Titular de Universidad

Dr. D. Càndid Reig Escrivá
Profesor Titular de Universidad

AGRADECIMIENTOS

Me gustaría tener un recuerdo especial para aquellas personas que, de una forma o de otra, han contribuido a la realización de este trabajo de investigación. A mi director de tesis, D. Juan Bautista Roldán Aranda y por haber puesto a mi disposición todos los medios necesarios para que este trabajo se haya podido realizar, así como agradecer su inestimable ayuda para compaginar las diferentes actividades de modelado realizadas durante estos años.

A mi otro director de tesis, D. Cándid Reig Escrivà, Profesor de la Universidad de Valencia, al que agradezco su gran comprensión y esfuerzo por iniciar, establecer y consolidar la colaboración para el modelado de sensores magnetorresistivos, por su ayuda con las medidas experimentales de los sensores y circuitos de aplicación y tareas de revisión y supervisión de los trabajos realizados durante este tiempo.

A Susana Isabel Pinheiro Cardoso, responsable de la Sala Limpia de INESC-MN (Instituto de Engenharia de Sistemas e Computadores - Microsistemas e Nanotecnologias) de Lisboa, por permitirme hacer uso de sus instalaciones e inestimable ayuda en el proceso de fabricación de los sensores de corriente en las diferentes etapas. Por su ayuda para establecer el modelo de ruido de los dispositivos y por su cordial y afectuosa acogida durante mi estancia en Lisboa. Un recuerdo especial para el resto de personal del INESC-NM por su ayuda y colaboración.

A mi tutor de periodo de investigación tutelada, D. Juan E. Carceller Beltrán por su

inestimable ayuda y apoyo, no sólo desde el punto de vista científico, siempre dispuesto a orientarme y a asesorarme en mi trabajo, sino también del personal en el día a día de mi labor formativa como investigador.

A D. Francisco Gámiz Pérez, que como responsable del Grupo de Investigación ha hecho posible el desarrollo de este trabajo.

A D. Francisco Jiménez Molinos, compañero inseparable de muchas tardes en la Facultad y al que le agradezco su ayuda en las tareas de investigación.

A D. Carlos Sampedro Matarín, Diego Pedro Morales Santos por su buen hacer en la difícil labor de compartir un sitio de trabajo. A D. Salvador Rodríguez Bolívar con el que he mantenido productivas discusiones científicas. A D. Juan Antonio López Villanueva por su asesoramiento.

A D. Andrés Godoy Medina y D. Pedro Cartujo Cassinello por su compañerismo y al resto de miembros del Grupo de Investigación de Nanoelectrónica. A todos los miembros del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

A mis padres, a mi familia y amigos que han estado siempre conmigo durante este largo periodo.

A mi esposa, por su tesón durante el desarrollo del trabajo, por su ayuda y comprensión.

Todos son igualmente partícipes de este trabajo.

ÍNDICE

Portada	a
Autorización	e
Agradecimientos	1
Índice	3
Índice de figuras	7
Índice de listados	15
Índice de tablas	17
Prólogo	19
Estructura del documento	23
1 Herramientas para la implementación de modelos avanzados de dispositivos electrónicos en simuladores de circuitos.	25
1.1 Introducción	25
1.2 Evolución histórica	26
1.3 Simuladores de circuitos: generalidades	27
1.4 Los modelos compactos en los simuladores y en la industria	30
1.5 Descripción de modelos compactos desarrollados en código fuente	32
1.5.1 Estructuras de datos para sistemas de ecuaciones circuitales.	34

1.5.2	Estructuras de datos y funciones para modelos compactos.	35
1.5.3	Ventajas e inconvenientes de los modelos en código fuente.	38
1.6	Verilog-A: un lenguaje de descripción de hardware para aplicaciones analógicas y de señal mixta. Compiladores. Paquetes TCAD comerciales para la simulación circuital: módulos de Verilog-A.	39
1.7	Ventajas del uso de Verilog-A	39
1.8	Metodología para la implementación de un modelo compacto en Verilog-A. .	40
1.8.1	Nuevas funcionalidades de Verilog-A	43
1.8.2	Arquitectura de Verilog-A en los simuladores de circuitos.	49
2	ESPICE: Implementación de SPICE desarrollada en el Grupo de Nanoelectrónica de la UGR para la adaptación de modelos compactos.	51
2.1	Introducción	51
2.2	Características internas de la versión inicial de SPICE.	52
2.2.1	Diseño del Simulador: Librerías	52
2.2.2	Tipos de análisis posibles mediante SPICE	54
2.2.3	Arquitectura del Simulador SPICE3	55
2.2.4	Estructura de datos encapsulados en SPICE3	57
2.2.5	Compilación del Simulador SPICE3F4 para Windows	58
2.2.6	Aspecto final de SPICE3F5	60
2.3	ESPICE. Descripción del nuevo simulador	61
2.3.1	Introducción.	61
2.3.2	Interfaz de ventanas.	63
2.3.3	Otras mejoras.	68
2.3.4	Resultados del test de verificación.	71
2.3.5	Conclusiones.	71
2.4	Adaptación de modelos de dispositivos en código fuente.	72
2.4.1	Modificación del modelo compacto BSIMSOI en código fuente.	72
2.4.2	<i>Matching</i> de los transistores N-MOS y P-MOS.	75
2.4.3	Inclusión de los efectos de overshoot de la velocidad en el modelo BSIMSOI.	75
3	Implementación de modelos avanzados de dispositivos multipuerta nanométricos en Verilog-A.	83
3.1	Dispositivos multipuerta.	83
3.2	Modelo de transistores SGT cilíndricos.	91
3.2.1	Efectos cuánticos en los <i>Surrounding Gate Transistors</i>	92
3.2.2	Modelo de control de carga en inversión.	93
3.2.3	Modelo de corriente de drenador.	96
3.2.4	Modelado de las capacidades del SGT.	99

3.2.5	Simulaciones de circuitos con dispositivos SGT	105
3.3	Double-Gate MOSFETs.	111
3.3.1	(Single Gate) SG-MOSFET	113
3.3.2	Modelo I-V de un transistor DG-MOSFET	115
3.3.3	Simulación de circuitos con dispositivos DG-MOSFET.	120
4	Modelado de sensores de corriente magnetorresistivos.	125
4.1	Introducción	125
4.1.1	Principios magnetorresistivos.	127
4.1.2	Aplicaciones. Medida de corriente R(I).	140
4.2	Desarrollo del modelo cuasi-estático	143
4.2.1	Elementos magnetorresistivos GMR individuales.	146
4.2.2	Puentes magnetorresistivos.	160
4.2.3	Diseño de puentes magnetorresistivos.	163
4.2.4	Modelado de ruido.	167
5	Modelado de aplicaciones con sensores magnetorresistivos de corriente.	177
5.1	Convertidores resistencia a frecuencia.	177
5.1.1	Conversión R-F mediante magnetorresistencia simple.	178
5.1.2	Conversión R-F mediante puentes magnetorresistivos.	182
5.2	Convertidor Generalizado de Impedancias (GIC)	184
5.2.1	GIC polarizado en DC	186
5.2.2	Resultados	187
5.3	Medida de la potencia.Wattímetro.	190
5.3.1	Descripción del sensor de corriente.	191
5.3.2	Descripción del modelo	193
5.3.3	Aplicación para la medida de la potencia eléctrica.	195
6	Publicaciones	197
7	Apéndices	201
7.1	Fabricación de dispositivos MTJ	201
7.1.1	Deposición de las capas que forman la magnetorresistencia MTJ.	203
7.1.2	Definición del Electrodo Inferior.	204
7.1.3	Definición del área del dispositivo MTJ.	206
7.1.4	Deposición de capa de aislante.	209
7.1.5	Primera metalización.	210
7.1.6	Deposición de capa de aislante SiO ₂ sobre pistas de corriente.	211
7.1.7	Apertura de los <i>pads</i> enterrados bajo el óxido.	211

7.1.8	Segunda metalización: pistas de corriente.	213
7.1.9	Pasivación.	213
7.1.10	Apertura final de los <i>pads</i> enterrados.	214
7.1.11	Tratamiento térmico magnético	215
7.2	Run Sheet	216
7.3	Estructuras de datos de SPICE	227
7.3.1	Estructura de datos del dispositivo.	227
7.3.2	Estructura de datos de CIRCUITOS	228
7.4	Creación y modificación de modelos en código fuente.	239
7.5	Resultados del Test de verificación.	242
7.5.1	Test de Análisis de Punto de Operación (.OP)	242
7.5.2	Test de Análisis en Alterna (.AC)	243
7.5.3	Test de Análisis Transitorio (.TRAN)	244
7.5.4	Comprobación de la estimación de la función de transferencia (.FT) .	245
7.5.5	Test del análisis de Fourier posterior al análisis transitorio (.FOURIER)	246
7.5.6	Test del Análisis de Sensibilidad (.SENS)	247
7.5.7	Comprobación del análisis polo-cero (.PZ)	248
7.5.8	Test del Análisis de Ruido (.NOISE)	250
7.5.9	Test del Análisis de Distorsión (.DISTO)	252
7.6	Modelado de alta frecuencia de la parte intrínseca.	253
8	Conclusiones.	255
	Bibliografía	259
	Acrónimos	275
	Glosario	283

ÍNDICE DE FIGURAS

1	Fases en el proceso de Diseño de un producto electrónico	20
1.1	Fases de la simulación de circuitos.	28
1.2	Simuladores comerciales existentes en la actualidad.	30
1.3	Diagrama de flujo del proceso de diseño y soporte de la industria microelectrónica.	31
1.4	Oferta de modelos compactos de dispositivos MOSFET.	32
1.5	Evolución número de parámetros de los modelos de MOSFET.	33
1.6	Diagrama de bloques de un modelo en código fuente y su conexión con el núcleo del simulador.	33
1.7	Circuito ejemplo con amplificador operacional ideal (a) y sistema de ecuaciones correspondiente (b).	35
1.8	Lista ortogonal enlazada de nodos para representar las ecuaciones del circuito de la figura 1.7(b).	36
1.9	Descripción de las fases de creación/modificación de un modelo compacto en código fuente.	36
1.10	Modelo compacto de un diodo.	40
1.11	Diagrama de flujo de uso de los modelos en Verilog-A en el simulador de circuitos (<i>Smartspice</i> [®]) de <i>Silvaco</i> [™]	49
1.12	Diagrama de flujo propuesto por <i>Tiburón</i> [™] para la obtención de CML en el simuladores de circuitos.	50
2.1	Estructura general de SPICE	56
2.2	Diagrama de bloques del simulador.	56
2.3	Estructura de datos encapsulados de un circuito	57

2.4	Estructura del Simulador en <i>BorlandC++5.01</i> (a) Estructura de directorios del proyecto (b).	59
2.5	Aspecto Simulador Spice3F5 en versión original.	60
2.6	Aspecto Simulador Spice3F5 en versión final.	61
2.7	Gestores de ventanas existentes: GTK+ (a), QT (b), WXwidget (c) y Fox-Toolkit (d).	63
2.8	Estructura de bloques del gestor gráfico de Espice.	63
2.9	Librerías gráficas disponibles en Espice.	64
2.10	Arquitectura de comunicación multihebra.	64
2.11	Estructura de directorios en MSYS y resultado compilación.	66
2.12	Barra de menú e iconos principales.	66
2.13	Navegador de archivos de simulación, histórico de comandos y listado de gráficos.	67
2.14	Barra de menú e iconos principales.	67
2.15	Ventana principal.	68
2.16	Interfaz de idiomas (a), colores (b) y fuentes (c).	68
2.17	Ventana gestor de opciones globales.	69
2.18	Ventana gestor de componentes discretos o subcircuitos.	69
2.19	Detalle de los mensaje de depuración orientados a desarrollo de modelos compactos.	70
2.20	Detalle del editor integrado en el simulador.	70
2.21	Representación gráfica con Carta de Smith (a), gráfica en Polares (b).	70
2.22	Sistema de desarrollo utilizado en GNU/Linux®	71
2.23	Estructura de un dispositivo PDSOI MOSFET	72
2.24	Emparejamiento (<i>matching</i>) para dispositivos BSIMSOI tipo N y P con $\lambda = 0$ y $\lambda = 22 \text{ m}^3/(\text{Vs})$	76
2.25	Transconductancia de un NMOSFET SOI de puerta simple en función de la longitud de canal a temperatura ambiente.	79
2.26	Curvas de salida de dispositivos de puerta simple SOI PMOSFET y NMOSFET.	79
2.27	Esquema circuital del oscilador en anillo de 51 etapas inversoras usado para estudiar la influencia de los efectos del VO en la frecuencia de oscilación.	80
2.28	Tensión de salida del el circuito oscilador en función del tiempo para diferentes valores del parámetro VO a temperatura ambiente.	80
2.29	Frecuencia de oscilación vs. el parámetro de VO a temperatura ambiente.	81
2.30	Tensión de salida del anillo oscilador en función del tiempo para diferentes longitudes de canal a temperatura ambiente.	81
2.31	Frecuencia de oscilación vs. longitud del canal.	82
3.1	Evolución de los dispositivos electrónicos en el contexto de la Ley de Moore [Sko11].	84

3.2	Dispositivos multipuerta [LBS ⁺ 09].	85
3.3	Modelos compactos para dispositivos MOSFET.	88
3.4	Revisión histórica de la evolución de los modelos compactos para MOSFET.	89
3.5	Desarrollo de modelos de MOSFET multipuerta para simulación de circuitos [MCH ⁺ 08].	90
3.6	Estructura simulada del dispositivo SGT (a), sección del dispositivo (b).	91
3.7	Carga del canal por unidad de longitud para un transistor SGT.	92
3.8	Capacidad puerta-canal (C_{gc}) en función de la tensión de puerta aplicada.	93
3.9	$\Delta V_{TH,QM}$ calculado numéricamente y el ajuste proporcionado por la ecuación 3.2.6.	94
3.10	Curvas de salida (normalizadas al diámetro del cilindro) para un dispositivo NMOS SGT.	99
3.11	Polarización en DC+AC	100
3.12	Capacidades de puerta C_{gd} y C_{gs}	102
3.13	Capacidades de puerta C_{dg} y C_{ds}	103
3.14	Capacidades de puerta C_{sd} y C_{sg}	104
3.15	Hoja de simulación en ADS [®] para realizar el <i>matching</i> ¹⁸ de los dispositivos SGT N y P para $L = 50$ nm y $R = 3$ nm.	105
3.16	Corriente I_{DS} obtenida de la simulación de la figura 3.15.	106
3.17	Corriente I_{DS} obtenida de la simulación de la figura 3.15.	107
3.18	Corriente I_{DS} obtenida de la simulación de la figura 3.15.	107
3.19	Hoja de simulación en ADS [®] de un tren de pulsos de tensión de entrada a un inversor CMOS con dispositivos SGT N y P para $L = 50$ nm y $R = 3$ nm.	108
3.20	Simulación temporal y curva de transferencia del inversor CMOS.	108
3.21	Hoja de simulación en ADS [®] para un anillo oscilador CMOS de 7 etapas cargadas con C_L	109
3.22	Tensión de salida en función del tiempo del anillo para diferentes λ_{a_0}	110
3.23	Frecuencia del anillo oscilador frente al parámetro del <i>overshoot</i> de la velocidad.	110
3.24	Frecuencia de oscilación del anillo en función del radio, teniendo en cuenta el modelo cuántico y el clásico	111
3.25	Velocidad de los electrones en función de campo longitudinal para SG-MOSFET a temperatura ambiente.	113
3.26	Velocidad de los electrones en función del campo longitudinal en SG-MOSFET.	114
3.27	Velocidad de los electrones en función del campo longitudinal para SGMOS-FETs a temperatura ambiente.	114
3.28	Velocidad de los electrones en función del campo longitudinal para DG-MOSFETs a temperatura ambiente.	115
3.29	Estructura de transistor DG-MOSFET.	116
3.30	Hoja de simulación en Advanced Design System para obtener curvas de salida en un dispositivo DG-MOSFET (tipo N).	121

3.31	Corriente I_{DS} para un dispositivo DG-MOSFET N-MOS.	121
3.32	Hoja de simulación en ADS [®] de un inversor CMOS con dispositivos DG-MOSFETs N y P.	122
3.33	Corriente I_{DS} para un dispositivo DG-MOSFET N-MOS.	122
3.34	Hoja de simulación en ADS [®] para un anillo oscilador CMOS de 7 etapas cargadas con $C_L = 2$ fF.	123
3.35	Tensión de salida respecto al tiempo del anillo oscilador.	124
3.36	Tensión de salida y potencia consumida respecto al tiempo del anillo oscilador CMOS de 7 etapas cargadas con $C_L = 2$ fF.	124
4.1	Clasificación de los sensores de corriente [HBEK09].	126
4.2	Fotografía y detalle de una cabezal de lectura/escritura de un disco duro. . .	127
4.3	Efecto Hall sobre semiconductor tipo-n [Mei08].	128
4.4	Representación esquemática de la trayectoria que describen los electrones en vacío y en medio sólido en presencia de campos magnético y eléctrico.	130
4.5	Sensor de corriente AMR : estructura (a) y variación de la resistencia con el ángulo θ (b) [JSS09].	131
4.6	Caracterización de las resistividades para configuración de magnetizaciones: paralela al campo (a) y perpendicular al campo (b) [Tor10].	132
4.7	Sección del orbital $3d$ para diferentes orientaciones de la magnetización [Car10].	132
4.8	Efecto AMR mostrando las diferencias en la circulación de la corriente a través de un dispositivo.	133
4.9	Sensor de corriente AMR	134
4.10	Valores del coeficiente de magnetorresistivo para un dispositivo magnetorresistivo AMR fabricado en INESC [Car10].	134
4.11	Estructura GMR : sistema multicapa FM/NM/FM (a), acoplamiento FM/NM/FM (b)	135
4.12	Efecto GMR en estructura CIP	135
4.13	Acoplamiento entre dos capas ferromagnéticas espaciadas por una capa ultra delgada de material no magnético.	136
4.14	Corriente CPP atravesando una estructura tricapa GMR	136
4.15	Estructura GMR modificada con la introducción de una lámina antiferromagnética [RCBRM09].	137
4.16	Estructura de una magnetorresistencia MTJ [GdA09].	138
4.17	Funcionamiento básico de una magnetorresistencia MTJ [GdA09].	138
4.18	Estructura SAF de un dispositivo magnetorresistivo MTJ [GdA09].	139
4.19	Conducción por efecto túnel en el óxido aislante de una estructura MTJ [GdA09].	139
4.20	Lineas de campo magnético [RCBRM09] generadas por circulación de una corriente a través de una pista conductora rectangular.	141

4.21	Estructura y campo magnético generado por una pista de corriente de sección rectangular.	142
4.22	Estructuras básicas de las diferentes tecnologías de sensores magnetorresistivos [RCBRM09].	143
4.23	Sensor de corriente magnetorresistivo.	144
4.24	Esquema de compilado y simulación mediante CML [MOT09b].	146
4.25	Ángulos de los vectores de magnetización de las capas libre.	147
4.26	Sección vertical de un dispositivo magnetorresistivo SV (a) y respuesta del dispositivo (b) [RCBR09].	147
4.27	Diferentes capas que forman el dispositivo magnetorresistivo SV [RRR+11b].	148
4.28	Etapas del proceso de fabricación de los dispositivos magnetorresistivos SV realizado en INESC Lisboa [RCR+09].	149
4.29	Vista superior del dispositivo de $8,2 \times 8,2$ mm que contiene 4 magnetorresistencias SV.	150
4.30	Procedimiento de medida para la caracterización la magnetorresistencia de un dispositivo SV versus I_{SENSOR}	151
4.31	Caracterización de la magnetorresistencia de los dispositivos SV versus I_{SENSOR} .	151
4.32	Tensión de salida de los dispositivos magnetorresistivos SV.	152
4.33	Magnetorresistencia normalizada de los dispositivos SV.	153
4.34	Resistencia y magnetorresistencia normalizada de los dispositivos SV.	154
4.35	Parámetros térmicos extraídos en función de la temperatura [RRR+11e].	155
4.36	Evolución temporal de la magnetorresistencia para pulsos de corriente de diferentes amplitudes [RRR+11e].	156
4.37	Evolución temporal de la magnetorresistencia del sensor $SV_{50\mu\text{m}}$ para pulsos de diferentes amplitudes [RRR+11e].	157
4.38	Evolución histórica del coeficiente TMR.	159
4.39	Magnetorresistencia de efecto túnel (TMR) versus el campo magnético aplicado (H) para una MTJ con aislante MgO	160
4.40	Tipos de puentes de Wheatstone [RCBR09].	161
4.41	Tipos de sensores de corriente magnetorresistivos basados en el puente de Wheatstone [CBRR+09].	163
4.42	Sensor de corriente magnetorresistivo AA004 del fabricante NVE TM [NVE03].	164
4.43	Resistencias del puente Wheatstone en función de la corriente excitación (I_{SENSOR}) del sensor (ZMC20).	165
4.44	Hoja de simulación de ADS para el puente magnetorresistivo AMR (ZMC20) de Zetex.	166
4.45	Tensión de salida del puente magnetorresistivo AMR (ZMC20) de Zetex.	167
4.46	Espectro de ruido teórico para dispositivo MTJ.	168
4.47	Equivalente circuital de las fuentes de ruido existentes en una SV	170

4.48	Sistema de caracterización del ruido generado por una magnetorresistencia.	171
4.49	Densidad espectral de ruido para un sensor <i>SV</i>	172
4.50	Hoja de simulación de <i>ADS</i> [®] para el análisis de ruido en un dispositivo <i>SV</i>	172
4.51	Equivalente circuital de las fuentes de ruido existentes en una <i>MTJ</i>	173
4.52	Espectro de ruido para un sensor <i>MTJ</i> de barrera MgO.	174
4.53	Listado de contribuciones de ruido de las diferentes fuentes asociadas al dispositivo magnetorresistivo para 3 mA de corriente de polarización a 500 KHz obtenidas en <i>ADS</i> [®]	174
4.54	Hoja de simulación de <i>ADS</i> [®] para el análisis de ruido en un dispositivo <i>MTJ</i>	175
4.55	Configuración de un puente de Wheatstone formado por 4 elementos sensibles <i>MTJ</i> de barrera MgO con un área de $1 \times 2 \mu\text{m}^2$ con una resistencia de 9.2Ω	175
4.56	Hoja de simulación de <i>ADS</i> [®] para el análisis de ruido en un puente de Wheatstone configurado por cuatro sensores <i>MTJ</i> . Los terminales VS+ y VS- corresponden a los terminales V_out(+) y V_out(-) de los puentes representados anteriormente.	176
4.57	Comparativa del ruido generado por una única magnetorresistencia <i>MTJ</i> (datos en líneas) y el ruido obtenido a la salida del puente de Wheatstone (datos en símbolos) para las mismas condiciones de excitación.	176
5.1	Sensor inteligente: conexión (a) convencional, (b) usando conversores tensión-frecuencia V/F con conversión A/D implementada en el microcontrolador.	178
5.2	Esquemático del circuito conversor corriente-frecuencia.	179
5.3	Tensión de entrada (V_C) y salida (V_0) del comparador.	179
5.4	Valores experimentales y de simulación para la tensión de salida V_O en función del tiempo para el sensor de corriente <i>SV07_{R3}</i> [<i>RRR+11a</i>].	180
5.5	Hoja de simulación en <i>ADS</i> para el circuito conversor corriente-frecuencia y el modelo compacto del sensor magnetorresistivo <i>SV07_{R2}</i>	180
5.6	Tensión de entrada (V_C) (CH_1) y de salida (V_o) (CH_2) del comparador.	181
5.7	Frecuencia de oscilación normalizada de la tensión de salida versus corriente principal del sensor <i>SV07_{Ri}</i>	181
5.8	Esquema del montaje experimental para fabricación de la PCB.	182
5.9	Placa de circuito impreso para el conversor corriente a frecuencia para magnetorresistencia simple: cara superior (a), cara inferior (b).	182
5.10	Montaje experimental.	182
5.11	Hoja de simulación en <i>ADS</i> para el circuito conversor corriente-frecuencia y el modelo compacto del puente magnetorresistivo <i>SV07</i>	183
5.12	Esquema del montaje experimental para fabricación de la PCB.	184
5.13	Placa de circuito impreso para el conversor corriente a frecuencia para puente magnetorresistivo: cara superior (a), cara inferior (b).	184
5.14	Frecuencia de oscilación de la tensión de salida versus corriente principal de los puentes del sensor <i>SV07</i> : PN-SN (a) y PW-SW (b).	185

5.15	Convertidor generalizado de impedancias, GIC [MME08].	185
5.16	GIC en régimen DC.	186
5.17	Esquemático de polarización de un sensor SV07 mediante un GIC.	187
5.18	Simulaciones de la tensión de salida para excitaciones de corriente de entrada: sinusoidal (a), cuadrada (b).	188
5.19	Resultados experimentales para señal cuadrada y sinusoidal de 1 KHz y 10 mA de amplitud.	188
5.20	Esquema del montaje experimental para fabricación de la PCB.	189
5.21	Placa de circuito impreso para el conversor GIC: cara superior (a) y cara inferior (b).	189
5.22	Señales eléctricas de tensión, corriente y potencia [RCBR09].	190
5.23	Principio de medida de la potencia en circuitos integrados [RCBR09].	191
5.24	Sección vertical del dispositivo SV fabricado en INESC.	192
5.25	Vista superior al microscopio del sensor de corriente (a) SV07-SN y (b) SV07-PW [RRCB+10].	192
5.26	Estructura circuital del cuadripolo equivalente de los elementos SV [RRCB+10].	193
5.27	Valores experimentales de las resistencias del puente en función de I_{SENSOR} [RRCB+10].	194
5.28	Tensión de salida del puente en función de la corriente.	195
5.29	Datos experimentales y simulados de la tensión de salida en función de R_L para diferentes valores de V_S [RRCB+10].	196
5.30	Datos simulados (líneas) y experimentales (símbolos) de la tensión de salida en función de la potencia consumida en R_L [RRCB+10].	196
7.1	Representación de la variación del ratio TMR de una muestra de capa delgada y gruesa en función de la tensión de polarización.	202
7.2	Montaje de una muestra sobre una oblea soporte de 6" de diámetro.	203
7.3	Estructura de capas de materiales que forman la MTJ que son depositados sucesivamente en el paso 1 del <i>Run Sheet</i> en la máquina Nordiko 2000.	204
7.4	Sistema de fotosensibilización (<i>Track 2</i>) y revelado (<i>Track 1</i>) de Silicon Valley Group™ para 25 obleas de 6" de diámetro.	205
7.5	Sistema de litografía láser DWL 2.0.	205
7.6	Vistas de la definición del electrodo inferior de la muestra.	207
7.7	Diferentes etapas en el proceso de <i>lift-off</i>	207
7.8	Vistas de la definición del dispositivo MTJ.	208
7.9	Oblea soporte de 3" para introducción en la máquina Nordiko 3000.	208
7.10	Detalle del proceso de atacado conforme va aumentando el tiempo de procesado y consecuentemente la profundidad de material erosionado.	209
7.11	Deposición de capa de aislamiento sobre la muestra.	209
7.12	Deposición de la metalización de las pistas de polarización.	210

7.13	Revisión con el microscopio óptico (a) de las muestras (b).	211
7.14	Deposición de la capa de óxido de 5000 Å que aísla al dispositivo <i>MTJ</i> y sus contactos de la pista de corriente del sensor.	211
7.15	Vistas del proceso de apertura de los <i>pads</i> en el óxido de aislamiento.	212
7.16	Sistema de atacado por plasma de reacción química de Research Rainbow	213
7.17	Vistas de la pistas de corriente del sensor. Sobre el óxido de aislamiento de 5000 Å se deposita la metalización. (a), (b) y (c) muestran la típica cadena de procesos para la metalización.	214
7.18	Deposición de la capa de óxido de 2000 Å para aislar las pistas de corriente de la atmósfera externa del dispositivo <i>MTJ</i> .	214
7.19	Vistas de la apertura final de los <i>pads</i> de los dispositivos de test.	215
7.20	Imágenes del sistema de <i>annealing</i> .	216
7.21	Perfil de temperatura durante el calentamiento para el tratamiento térmico del <i>annealing</i> .	216
7.22	Relación modelo-elemento.	228
7.23	Circuito de test para punto de operación.	242
7.24	Circuito de test para el análisis en alterna.	243
7.25	Resultados del análisis en alterna.	244
7.26	Circuito de test para el análisis transitorio.	244
7.27	Resultados de la simulación temporal del circuito de la figura 7.26.	245
7.28	Circuito de test para de la función de transferencia	246
7.29	Circuito de test para el análisis de Fourier	247
7.30	Circuito de test para el análisis de sensibilidad.	248
7.31	Circuito de test para el análisis polo-cero.	249
7.32	Tensión en el Nodo 3 [Salida del Filtro en T].	249
7.33	Circuito de test para el análisis de ruido	250
7.34	Resultados del análisis de ruido	251
7.35	Circuito de test para el análisis de distorsión	252
7.36	Modelo alta frecuencia completo.	254

ÍNDICE DE LISTADOS

1.1	Estructura SPICEdev.	37
1.2	Modelo compacto de un diodo.	40
1.3	Otras posibilidades	43
1.4	Dispositivos de tipo N ó P, utilización del parámetro TYPE.	44
1.5	Funciones limitadoras.	46
1.6	Funciones de ayuda a la depuración.	47
1.7	Acceso a variables internas del modelo compacto.	47
1.8	Árbol de dependencias.	48
1.9	Eliminación de nodos internos.	49
2.1	Definición de librerías gráficas para representación de funciones.	63
2.2	Implementación de técnica antibloqueo en la comunicación SPICE \mapsto Interfaz gráfico.	64
2.3	Directivas de compilación condicional.	66
2.4	Listado del código de procesado numérico de los parámetros del modelo.	73
2.5	Listado del código de resolución de consultas de los parámetros del modelo.	73
2.6	Listado del código de inicialización por defecto de los valores de los parámetros.	74
2.7	Listado del código del cálculo de la corriente I_{DS}	74
2.8	SPICE Netlist para comprobación de MATCHING entre dispositivos NMOS y PMOS.	75
3.1	Listado de incorporación de capacidades en el modelo de SGT.	104
3.2	Listado de parámetros del dispositivo SGT N-MOS de $L=50$ nm.	105
4.1	Definición modelo compacto para sensor de corriente magnetorresistivo en Verilog-A.	144
4.2	Definición de los parámetros del modelo.	145
4.3	Tensión Primario.	145

4.4	Tensión Secundario.	145
4.5	Código necesario para incorporación del modelado térmico estático.	154
4.6	Código necesario para incorporación del modelado térmico dinámico.	157
4.7	Definición de contribuciones de ruido para dispositivo SV.	171
4.8	Definición de contribuciones de ruido para dispositivo MTJ	173
7.1	Estructura GENmodel.	227
7.2	Estructura GENinstance.	227
7.3	Estructura CKTcircuit.	228
7.4	Estructura CKTnode.	230
7.5	Estructura SPICEanalysis.	230
7.6	Estructura JOB.	231
7.7	Estructura TSKtask.	231
7.8	Estructura ACAN.	232
7.9	Estructura OP.	232
7.10	Estructura PZAN.	233
7.11	Estructura TFa.	233
7.12	Estructura TRANan.	233
7.13	Estructura TRCV.	234
7.14	Estructura SMPmatrix.	235
7.15	Estructura MatrixFrame.	235
7.16	Estructura INPtab.	236
7.17	Estructura INPntab.	237
7.18	Estructura INPtable.	237
7.19	Estructura CARD.	238
7.20	Estructura INPmodel.	238
7.21	Estructura INPparseTree.	238
7.22	Estructura INPparseNode.	239
7.23	Estructura PTelement.	239
7.24	Detección del modelo compacto utilizado.	240
7.25	Detección del nivel del modelo compacto utilizado.	240
7.26	Obtención del número máximo de nodos disponibles en un modelo compacto utilizado.	241
7.27	Interpretación de los datos asociados a un elemento.	241
7.28	Netlist del circuito de test para análisis en punto de operacion.	242
7.29	Netlist del circuito de test para análisis en frecuencia.	243
7.30	Netlist del circuito de test para análisis transitorio.	245
7.31	Netlist del circuito de test para análisis de Fourier.	247
7.32	Netlist del circuito de test para análisis de sensibilidad.	247
7.33	Netlist del circuito de test para análisis polo-cero.	248
7.34	Netlist del circuito de test para análisis de ruido.	250
7.35	Netlist del circuito de test para análisis de distorsión.	252

ÍNDICE DE TABLAS

1.1	Simuladores de Circuitos Electrónicos	30
1.2	Dimensiones de los modelos desarrollados en código fuente.	38
2.1	Tipos de análisis en corriente continua.	54
2.2	Tipos de análisis en corriente alterna.	55
2.3	Tipos de análisis transitorios.	55
2.4	Comparativa de características de los diferentes simuladores estudiados.	62
3.1	Futuros parámetros tecnológicos hasta el 2019 [ITR11] para las tecnologías <i>planar bulk</i> en color rojo y <i>multi-gate</i> en azul.	87
4.1	Comparativa de las prestaciones de los sensores magnetorresistivos de diferentes tecnologías [RCBR09].	143
4.2	Parámetros del modelo compacto para magnetorresistencias SV tanto para comportamiento lineal como para no lineal [RRR+11e].	152
4.3	Parámetros térmicos del modelo de magnetorresistencia para dispositivos SV [RRR+11e].	154
4.4	Parámetros del modelo térmico dependiente del tiempo para dispositivos SV [RRR+11e].	156

4.5	Comparativa del comportamiento térmico de los dispositivos magnetorresistivos SV [RRR+11e].	157
4.6	Métodos magnéticos no invasivos de monitorización corporal [SB06].	158
4.7	Características de Sensores magnetorresistivos AMR y GMR (SV) comerciales.	164
4.8	Parámetros del modelo de magnetorresistencia para el puente ZMC20	166
4.9	Parámetros del modelo de las magnetoresistencias del puente ZMC20.	166
5.1	Periodo de la señal para diferentes corrientes de entrada [RRR+11a].	179
5.2	Parámetros del modelo de resistencias del puente [RRCB+10].	194
7.1	Características de las cuatro magnetorresistencias formadas por la agrupación serie de dispositivos MTJ individuales.	203
7.2	Configuración de pasos y tiempos en el proceso de horneado.	204
7.3	Configuración de pasos y tiempos en el proceso de fotosensibilizado y revelado.	206

PRÓLOGO

El escalado de transistores **MOSFET** ha sido la estrategia seguida por la industria electrónica en los últimos 40 años para mejorar las prestaciones de los Circuitos Integrados (**IC**), tanto en sus aplicaciones de altas prestaciones (microprocesadores) como en las de almacenamiento de información (memorias). Las estrategias de escalado utilizadas han estado orientadas, en términos generales, hacia procesos de reducción de las dimensiones geométricas de los dispositivos electrónicos y de la tensión de alimentación de los **IC**. Las contrapartidas a esta tendencia son entre otras: el incremento de la complejidad de la tecnología de fabricación, aumento de la variabilidad de algunos parámetros, el aumento de la potencia disipada por los **IC**, en particular la potencia disipada en *standby*, etc.

En lo que respecta a las dimensiones de los transistores, la mejora de prestaciones de los **IC** se ha sustentado en una estrategia de disminución de la longitud del canal de los transistores **MOSFETs** (L_G), del grosor del óxido de puerta (T_{ox}), y del escalado de los contactos de fuente y drenador (S-source/D-drain). Esta reducción geométrica ha sido llevada a cabo en las distintas generaciones de transistores (tecnologías que se han ido poniendo a punto en fase de producción industrial y han sido caracterizadas por la longitud del canal de los transistores utilizados) estableciendo un equilibrio entre el consumo de potencia, la fiabilidad y las prestaciones de los **IC**.

Desde hace años, los sistemas electrónicos que antes ocupaban una o más placas han sido integrados en un único chip con igual o mayor funcionalidad. Ejemplos de estos dispositivos conocidos como **Sistemas en Chip (SoC)** son los receptores de televisión en un chip, las cámaras de vídeo en chip y las nuevas generaciones de sistemas integrados de telecomunicaciones que incluyen los bloques analógicos, digital e incluso de radiofrecuencia **RF** en un único chip. La tecnología de fabricación que ha permitido ésto es la **CMOS**, que es la piedra angular de la tecnología electrónica actual, y en función de las aplicaciones su combinación

con otras como la **BICMOS** para su uso en circuitos de radiofrecuencia (**RF**) . Aunque la mayoría de las funciones en esos sistemas integrados se implementan con bloques digitales o con procesadores digitales de señal (**DSP**), los circuitos analógicos se necesitan como interfaces entre el mundo real y el electrónico del interior del sistema. Los circuitos analógicos son indispensables en todas las aplicaciones electrónicas que interactúan con el mundo real y serán cada vez más importantes en los nuevos horizontes tecnológicos como los edificios inteligentes, las oficinas inalámbricas, productos multimedia, etc. Cuando los sistemas analógicos y digitales deben coexistir en un producto, la integración de éstos en un único chip es recomendable.

La clave para la gestión correcta del incremento de complejidad de los diseños manteniendo las condiciones óptimas del *time-to-market* es el uso de los sistemas de **Diseño Asistido por Computador (CAD)** y las herramientas de verificación.

Para facilitar el proceso del diseño del **IC** es necesario una metodología estructurada que esté soportada por herramientas **CAD** que permitan tratar de manera global el proceso de diseño en toda su complejidad, ver figura 1. El proceso de creación de un **IC** se divide en

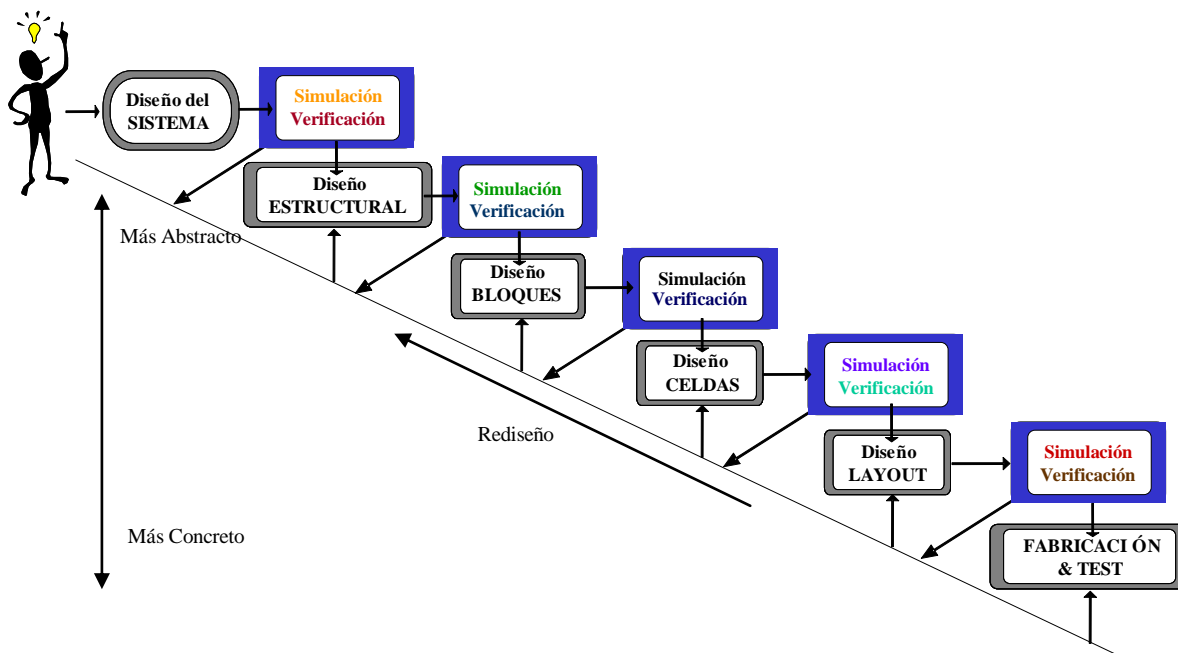


Figura 1. Proceso de diseño de un circuito electrónico.

siete etapas bien diferenciadas. En varias etapas del ciclo del diseño se realizan procesos de simulación y verificación que permiten detectar problemas que puedan limitar el alcance de los objetivos iniciales. Si aparecen errores se ha de abordar un proceso de re-diseño. Estas etapas son:

1. Diseño Conceptual: En la que se produce el diseño conceptual y donde se establecen las especificaciones de producto. La verificación profunda de las especificaciones es necesaria para el posterior éxito del producto en el contexto de su aplicación. Se suelen utilizar

- herramientas matemáticas como [MATLAB](#)/Simulink. Se establecen los parámetros de costo, tiempo de desarrollo, planificación de producción, etc.
2. Diseño del Sistema: Constituye la primera fase del diseño donde toda la arquitectura del sistema se diseña y particiona. Las partes software y hardware son definidas y se describen utilizando lenguajes específicos. En esta etapa se toman decisiones acerca de la implementación del sistema: encapsulado, tecnología a utilizar y estrategias de test. El particionamiento en diferentes niveles y sus especificaciones particulares son verificadas utilizando técnicas detalladas de co-simulación.
 3. Diseño de la Arquitectura: Durante esta etapa se realiza la descomposición de alto nivel de las funciones *hardware* y se crean los bloques funcionales que implementan las especificaciones de bloque obtenidas en la fase anterior. Se realiza la división entre lo analógico y digital. El cumplimiento de las especificaciones del diseño de la arquitectura a alto nivel se verifica mediante simulaciones analógico-digital.
 4. Diseño del Celdas: Se detalla el comportamiento de los bloques analógicos así como su implementación con la tecnología elegida, apareciendo por primera vez un esquemático global de producto. Los bloques analógicos se descomponen en sub-bloques. Las consideraciones relativas a la fabricación, tolerancias, etc., son tenidas en cuenta para garantizar la producción y viabilidad en fábrica. El circuito resultante de esta fase de diseño se verifica contra las especificaciones utilizando Sistemas de Diseño Asistido por Computador tales como SPICE. Este tipo de simuladores de circuitos y la creación de los modelos necesarios para su uso son el objeto principal de esta tesis doctoral.
 5. Diseño del *layout* de las Celdas: En esta fase se realiza la traducción de esquemas eléctricos de los diferentes bloques analógicos utilizados a su representación geométrica de multi-capas a nivel de silicio. Se realizan optimizaciones sobre el área ocupada por las celdas de manera que se obtenga un *layout* con el menor tamaño. Una vez concluida la optimización se procede con la extracción de los parásitos del *layout* y las simulaciones de los circuitos que incluyan estos parásitos de forma que se aseguren las especificaciones.
 6. Diseño completo del sistema: La generación del *layout* del IC no solo incluye la situación sobre el silicio de los diferentes bloques que conforman la aplicación sino también el enrutado de los sistemas de alimentación de estos. Los problemas clásicos de diafonía y los acoplos con el sustrato son muy importantes, por lo tanto nuevas técnicas de evaluación de aislamiento y acoplo deben ser empleadas para asegurar el correcto funcionamiento del chip. Deben incluirse estrategias de test que permitan verificar las características finales del dispositivo. El sistema es finalmente verificado mediante co-simulación del Hardware ([HW](#)) y el Firmware ([FW](#)).
 7. Fabricación y verificado: Se generan las máscaras que permiten la fabricación del Circuito Integrado. La comprobación y verificado se realiza durante y tras la fabricación de manera que se permita eliminar los dispositivos defectuosos.

Los sistemas [CAD](#) son muy necesarios para facilitar y/o automatizar muchas de las tareas rutinarias y repetitivas de diseño evitando los tediosos diseños manuales, permitiendo al diseñador centrarse en los aspectos creativos.

El simulador de circuitos es parte fundamental de los sistemas **CAD** utilizados en el proceso de diseño de productos electrónicos. La obtención e implementación de nuevos modelos compactos (**CM**) que permiten la descripción del comportamiento de los dispositivos electrónicos reales en los simuladores es igualmente necesaria. El avance de los procesos tecnológicos que posibilita la disminución del tamaño de los dispositivos electrónicos fabricados y el uso de técnicas de diseño innovadoras, obliga a depurar las técnicas de modelado y a desarrollar y actualizar modelos compactos que permitan a los simuladores de circuitos describir correctamente los dispositivos de una determinada tecnología.

Los avances tecnológicos en la fabricación de los dispositivos electrónicos deben ir acompañados del desarrollo de los modelos compactos que al incluirse en los simuladores permitan al diseñador verificar el correcto funcionamiento del producto electrónico en todos los niveles. En este trabajo de investigación se presentan los resultados obtenidos en las actividades de modelado e implementación de modelos compactos de dispositivos magnetorresistivos y semiconductores. Estos modelos se han implementado e incluido en un simulador de circuitos electrónicos que ha permitido la simulación de aplicaciones reales haciendo uso de ellos. La implementación final se puede distribuir en la comunidad de diseñadores de circuitos para ser utilizada en los simuladores de circuitos más famosos del mercado.

El hilo conductor de este trabajo está articulado en torno a la implementación de modelos compactos en **Verilog-A**. Este lenguaje es el estándar de facto en el desarrollo de modelos compactos en la actualidad por su flexibilidad, portabilidad y facilidad de uso. Los modelos desarrollados han sido compilados y utilizados en simuladores comerciales como **ELDO**[®] (Mentor Graphics) y **ADS**[®] (Agilent).

Se han estudiado modelados de dispositivos multipuerta nanométricos fabricados en sustratos **SOI**. Esta tecnología tiene todos los elementos necesarios para constituirse en la elegida para la fabricación de los transistores que serán las celdas básicas de las futuras tecnologías de **IC** en los próximos años. Además, se han simulado sensores de corriente basados en magnetorresistencias gigantes (**GMR**). El enfoque ha sido puesto en tecnologías convencionales de silicio para futuros nodos destinados a constituir la tecnología “*mainstream*” tanto en aplicaciones **HP** (*High-Performance*) como en aplicaciones **LP** (*Low-Power*), también en lo que actualmente despierta gran interés para dispositivos móviles como *smartphones* y *tabletPCs* las aplicaciones **LPHP** (*Low-Power High-Performance*). Este enfoque se encuadraría en lo que se conoce como “*more Moore*”. Por otro lado, haciendo uso de dispositivos ligados a una nueva línea de investigación asociada a la spintrónica abordamos el potencial de tecnologías emergentes (“*more than Moore*”). La compatibilidad de los sensores **GMR** con tecnologías **CMOS** y el potencial del conjunto en relación a sistemas de control de potencia en **IC** permite unificar el trabajo bajo el prisma del modelado compacto. Todo esto unido a una serie de aplicaciones circuitales que ponen en valor los modelos desarrollados e implementados constituyen el objeto esta tesis doctoral.

ESTRUCTURA

Esta tesis describe el proceso de modelado de diferentes dispositivos electrónicos, la elaboración de modelos compactos para simuladores de circuitos electrónicos y la simulación de numerosas aplicaciones circuitales usando estos modelos compactos.

En el **Capítulo 1** se introducen las herramientas para la implementación de modelos avanzados de dispositivos. Se recorre la evolución histórica y los diferentes tipos de simuladores de circuitos electrónicos existentes. Se presentan los modelos compactos diseñados tanto en código fuente como en [Verilog-A](#) y sus ventajas e inconvenientes.

En el **Capítulo 2** describe el simulador de circuitos desarrollado en el Grupo de Nanoelectrónica de la Universidad de Granada y el procedimiento para la modificación de modelos compactos ya existentes y la incorporación de nuevos efectos físicos en la simulación de circuitos.

Con el **Capítulo 3** se desarrollan modelos compactos para dispositivos multipuerta donde se incluyen diferentes efectos físicos esenciales en dispositivos nanométricos (efectos cuánticos, del *overshoot* de la velocidad, de la velocidad de saturación, de canal corto, ...). Se estudian las implicaciones de estos efectos en el comportamiento de diferentes circuitos. Se desarrollan los modelos de control de carga en inversión, corriente y capacidades para transistor [SGT](#) y doble puerta.

En el **Capítulo 4** se introducen los diferentes dispositivos magnetorresistivos y su utilización como sensores de corriente. Se realiza la caracterización experimental de los dispositivos incluyendo su comportamiento térmico (efectos de autocalentamiento y de ruido). Mediante el uso de varios dispositivos se construyen puentes de Wheatstone que son igualmente modelados.

En el **Capítulo 5** se presentan los resultados experimentales de un conjunto de aplica-

ciones con sensores magnetorresistivos de corriente (convertidores de resistencia a frecuencia, convertidores generalizados de impedancias y medidores de potencia). Se realizan simulaciones de los circuitos incluyendo los modelos compactos obtenidos en el capítulo 4. Finalmente, se extraen ordenadamente las conclusiones más importantes de este trabajo en la sección correspondiente.

En el apartado de **Publicaciones** se listan todos los resultados de comunicaciones a congresos y publicaciones en revistas incluidas en el índice [JCR](#) obtenidas durante el desarrollo de este trabajo.

En el **Apéndice** se han reunido todos los contenidos que por su nivel de detalle hemos extraído de los capítulos anteriores. Durante la lectura del presente documento se realizarán repetidas referencias a los contenidos incluidos en el apéndice.

CAPÍTULO

1

HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE MODELOS AVANZADOS DE DISPOSITIVOS ELECTRÓNICOS EN SIMULADORES DE CIRCUITOS.

1.1 Introducción

Un modelo compacto es una descripción matemática de un componente electrónico que es usado por diseñadores y tecnólogos para simular el comportamiento de ese componente en un circuito electrónico. Estos modelos permiten determinar numéricamente las características del dispositivo o componente (*corrientes, cargas y ruido* en función de las tensiones de los terminales) de manera precisa y rápida de modo que posibilitan la simulación de circuitos eléctricos completos [WTK07]. Los modelos compactos son usados en el diseño de IC para describir elementos activos y pasivos tanto desde un enfoque de parámetros concentrados como distribuidos.

La implementación de modelos compactos comenzó con el desarrollo de los primeros simuladores de circuitos. En la primera versión de SPICE ([Simulation Program with Integrated Circuit Emphasis](#)), a principios de los años 70, se implementó el modelo LEVEL1

de [SPICE](#). Posteriormente, en [FORTRAN](#), se fueron introduciendo mejoras y nuevos modelos (LEVEL2, LEVEL3, ...). En todos los casos estos modelos se compilaban dentro del simulador de circuitos; son lo que se llamaba modelos de código fuente.

En los últimos años ha cambiado el paradigma de implementación de modelos compactos. Se utilizan lenguajes externos al compilador ([Verilog-AMS](#) y, sobre todo, con un enfoque principalmente analógico, [Verilog-A](#)). Estos modelos se compilan por separado y el código compilado se utiliza en tiempos de ejecución por el simulador de circuitos.

Actualmente hay diferentes empresas dedicadas a mejorar los compiladores de [Verilog-A](#) y se están sucediendo las versiones de modelos compactos en este lenguaje. Además los modelos más importantes [BSIM](#) ([Berkeley Short-channel IGFET Model](#)), [PSP](#) ([Philips & SP model \(Penn State University\)](#)), [HiSIM](#) ([Hiroshima-university STARC IGFET Model](#)), [EKV](#) ([Enz-Krummenacher-Vittoz Model](#)), etc. están implementados en este lenguaje.

Sin entrar en demasiados detalles, las razones que han motivado que [Verilog-A](#) se convierta en un estándar para el desarrollo e implementación de modelos compactos han sido:

1. Los desarrolladores han descubierto la facilidad y potencia de uso como lenguaje de descripción.
2. Los simuladores comerciales han incluido las extensiones necesarias para permitir el uso de modelos compactos implementados en [Verilog-A](#) con la misma velocidad y fiabilidad que los tradicionales modelos descritos con código fuente en [lenguaje-C](#).
3. Los modelos implementados en [Verilog-A](#) se pueden compartir y usar en todos los simuladores, tanto gratuitos como de pago.
4. [Verilog-A](#) es muy sencillo de usar y puede ser incorporado también en paquetes software de extracción de parámetros de dispositivos.

1.2 Evolución histórica

El lenguaje estándar de modelado compacto de dispositivos electrónicos ha sido el [lenguaje-C](#) desde 1985, coincidiendo con la migración realizada del código fuente del simulador [SPICE](#) [[Simulation Program with Integrated Circuit Emphasis](#)] [[ND73](#)] desde el lenguaje original (versión: Spice2) [[Nag75](#)] escrito en [FORTRAN](#) a lenguaje C (versión:Spice3) [[JQN+92](#)]. Desde entonces se ha utilizado el [lenguaje-C](#) como herramienta para el desarrollo de modelos compactos. El Compact Model Council¹ [[CMC96](#)] recomendó el [lenguaje-C](#) en el pasado; sin embargo, en la actualidad recomienda la implementación de modelos en lenguaje [Verilog-A](#) para la implementación de modelos compactos de dispositivos electrónicos.

Las características que deben verificar todos los modelos compactos descritos tanto en [lenguaje-C](#) como en [Verilog-A](#) para simuladores de circuitos tanto analógicos como digitales son:

¹Compact Model Council: Es un grupo de trabajo de la industria [EDA](#) creado para elegir, mantener y promocionar el uso de modelos de dispositivos estándares.

- Las expresiones de corriente en función de la tensión $i(v)$ de los terminales deben ser continuas en todo el rango de utilización del dispositivo, y es recomendable que lo sean también en los extremos del intervalo de tensiones.
- Las expresiones de las derivadas de corriente en función de la tensión $\frac{\partial i(v_i)}{\partial v_i}$ deben ser continuas también para facilitar la convergencia del [Algoritmo de Newton-Raphson](#).
- Las expresiones de las derivadas n-ésimas de corriente en función de la tensión $\frac{\partial^n i(v_i)}{\partial^n v_i}$ de los terminales deben ser continuas también para poder realizar correctamente el análisis de distorsión armónica.
- Las expresiones de las cargas en función de la tensión $q_i(v_j)$ de los terminales deben ser continuas también para facilitar la convergencia del método de [Algoritmo de Newton-Raphson](#).

El lenguaje de descripción hardware [Verilog-AMS](#) [AI03] y más concretamente el sublenguaje para descripción analógica denominado [Verilog-A](#) han sufrido recientes mejoras en las capacidades de modelado compacto tras la publicación de la versión 2.2 del *Verilog-AMS Language Reference Manual (LRM)* [AI08]. Dichas mejoras se produjeron durante los años 2002-2004 gracias al trabajo conjunto de los desarrolladores de modelos y los fabricantes de los simuladores de circuitos. El estándar fue aprobado definitivamente por el [Accellera Board of Directors](#) en septiembre de 2004. Muchas de las mejoras introducidas en el sublenguaje de modelado analógico fueron propuestas por Lemaitre [LCMK03].

El sublenguaje [Verilog-A](#) era considerado ya un excelente lenguaje de modelado compacto con ventajas importantes sobre el [lenguaje-C](#) incluso antes de que las extensiones mencionadas anteriormente fueran introducidas en la nueva versión del [LRM](#). Las principales características que presenta serán detalladas en el apartado 1.7.

El lenguaje [Verilog-A](#) que implementa la versión 2.1 del [LRM](#) está disponible actualmente en casi todos los simuladores comerciales, y en herramientas de extracción de parámetros. La implementación de la versión 2.2 del estándar está disponible sólo en algunos simuladores y pronto sustituirá a la versión 2.1 en el resto de herramientas de simulación.

1.3 Simuladores de circuitos: generalidades

El análisis de circuitos se define como el cálculo de las tensiones y corrientes de un circuito excitado con señales de entrada definidas. Esta tarea puede ser desarrollada de manera automática por computadores permitiendo una mejora considerable en la precisión y tiempo de cálculo. Actualmente existen complejos programas de ordenador que realizan todo tipo de análisis sobre circuitos electrónicos.

En la figura 1.1 se presentan de forma gráfica las diferentes fases en la simulación de un circuito mediante un computador.

Las tareas de descripción del circuito y la presentación de resultados implican poco coste computacional. Una vez el circuito es procesado, se genera una lista enlazada que incluye a cada uno de los elementos del circuito. Mediante un análisis posterior de dicha lista

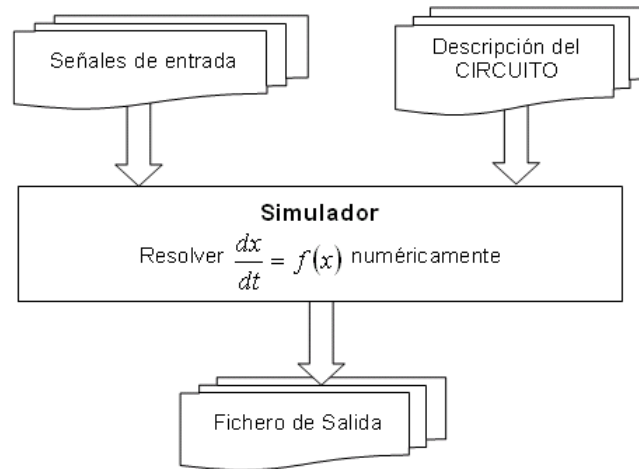


Figura 1.1 – Fases de la simulación de circuitos.

se formulan las ecuaciones constitutivas del circuito. Del mismo modo, una vez calculadas las tensiones y corrientes en cada nudo del circuito, se realiza la presentación de los datos.

En general, en el análisis de circuitos electrónicos se utilizan las Leyes de Kirchhoff de Corriente (KCL) en conjunción con las ecuaciones constitutivas de mallas del circuito. Las razones que motivan esta elección están relacionadas con la naturaleza de los componentes electrónicos, ya que la mayoría de los elementos activos se comportan como fuentes de corriente controladas.

La investigación en la simulación automática de circuitos comenzó en la década de los años 50 [Ped84]. Durante los años 1969 a 1970, *Ronald Rohrer* comenzó, con un grupo de unos diez alumnos postgraduados, un proyecto para estudiar el análisis de circuitos mediante el uso de computadoras digitales. Se estudiaron todas las facetas del problema computacional: formulación de ecuaciones, linealización de comportamientos de dispositivos no lineales, técnicas de integración numérica, de resolución de sistemas matriciales dispersos, de eliminación de Gauss de descomposición matricial LU y de pivotaje para aumento de estabilidad.

Como resultado de todo este estudio se obtuvo el programa **CANCER** (**C**omputer **A**nalysis of **N**onlinear **C**ircuits, **E**xcluding **R**adiation). Después de un uso intenso de los alumnos del *Electronics Research Laboratory* de la Universidad de Berkeley, el simulador **CANCER** evolucionó gracias al trabajo de *Larry Nagel*. La siguiente versión se denominó **SPICE1** bajo la dirección de su tutor de investigación *Donald Pederson* [Ped84].

Nagel profundizó en los aspectos propios de la elección de los componentes y en las técnicas numéricas necesarias en **SPICE1**. La versión posterior se denominó **SPICE2**.

A comienzos de 1975, **SPICE2** se convirtió en una herramienta **CAD** de difusión mundial. Este simulador, y los creados a partir de una evolución de éste denominados comúnmente *SPICE's like simulators*, compartían unas características comunes consistentes en:

- Una técnica de análisis nodal modificado (**MNA**) que permitía el uso de fuentes de tensión y fuentes de nodos flotantes.
- Incorporación de técnicas de integración *backward* de primer y segundo orden.

- Mejora de gestión de memoria del computador y del tiempo de cálculo.
- Incremento de la velocidad por el aprovechamiento de la dispersión de los valores de la matriz que representa el sistema de ecuaciones una vez linealizado.
- Uso de las técnicas de pivotaje para mantener la dispersión de la matriz del sistema.
- Linealización mediante el uso de [Newton-Raphson](#).
- Descomposición [LU](#) del sistema matricial para la obtención de la solución de las ecuaciones circuitales.

[SPICE](#) promovió y sirvió de base para otros programas de simulación en las universidades y la industria. La primera versión comercial del [SPICE](#) fue [ISPICE](#) ([Interactive Simulated Program Integrated Circuit Emphasis](#)) desarrollado en *National CSS, Inc.* Las versiones comerciales más destacadas de [SPICE](#) han sido [HSPICE](#) (comercializado originalmente por Shawn y Kim Hailey en *Meta Software*, que fue absovida posteriormente por *Synopsys*) y [PSPICE](#) (desarrollado por *MicroSim*, adquirida por *Orcad* y finalmente por *Cadence Design Systems*). Las versiones académicas de [SPICE](#) incluían [XSPICE](#), desarrollada en el Instituto Tecnológico de Georgia, versión en la que se agregaron códigos de análisis analógico y digital y [CIDER](#) (conocido con anterioridad como [CODECS](#) [[May88](#)], desarrollado por las universidades de *UC Berkeley/Oregon State Univ.*) que permitía simular dispositivos semiconductores e integrarlos en el simulador de circuitos.

[SPICE](#) fue adoptado como un estándar de facto por toda la industria electrónica, y hasta que se dispuso de versiones comerciales muchos fabricantes de [IC](#) disponían de versiones propietarias. En la actualidad existen todavía fabricantes de dispositivos y [IC](#) que disponen de departamentos para el desarrollo y mejora de simuladores de circuitos. Se pueden destacar los siguientes:

- Basados en [SPICE](#)
 - [ADICE](#) en [Analog Devices](#)
 - [LTspice](#) en [Linear Technology](#)
 - [Mica](#) en [Freescale Semiconductor](#)
 - [TISPIICE](#) en [Texas Instruments](#)
- No basados directamente en [SPICE](#)
 - [PowerSpice](#) en [IBM](#)
 - [Titan](#) en [Infineon/Qimonda](#)
 - [Lynx](#) en [Intel Corporation](#)
 - [PStar](#) en [NXP Semiconductor](#)

La simulación de circuitos es una de las tareas más críticas y que mayor tiempo consume durante el diseño de circuitos integrados. El diseño de circuitos [VLSI](#) demanda simulaciones precisas y muy tediosas. Los intentos realizados para aumentar la velocidad de simulación conllevan pérdida de precisión, y del mismo modo, el aumento de precisión conlleva el aumento del tiempos de simulación.

1



Figura 1.2 – Simuladores comerciales existentes en la actualidad.

En los últimos años se ha producido una concentración en el mercado de los simuladores comerciales de circuitos de donde han ido desapareciendo paulatinamente gran parte de los simuladores existentes hace una década. En la figura 1.2 se muestra un resumen de los simuladores existentes junto con el nombre de la empresa que lo comercializa. En la tabla 1.1 se muestra un listado de los simuladores de circuitos que han estado presentes en la historia reciente.

Logo	Nombre	Fabricante	Costo	Verilog-A
	SPICE	Univ. de Berkeley	Gratuito	No
	XSPICE,	Georgia Tech Research Institute	Bajo precio	No
	OPUS SPICE	Univ. Ljubljana, Slovenia	Gratuito	Si
	HSPICE	Synopsys	Pago	Si
	SPECTRE	Cadence	Pago	Si
	PSPICE	Cadence	Pago	No
	ADS	Agilent	Pago	Si
	APLAC	AWR	Pago	Si
	SmartSpice	SIMUCAD	Pago	Si
	Micro-Cap	Spectrum Software	Pago	No
	ICAP-4	Intusoft	Pago	No
	MultiSim	National Instruments	Pago	Si
	TopSPICE	PENZAR Development	Pago	No

Tabla 1.1 – Simuladores de Circuitos Electrónicos

1.4 Los modelos compactos en los simuladores y en la industria

El desarrollo de modelos compactos es una de las actividades principales en el desarrollo de productos electrónicos. En la figura 1.3 se presenta el diagrama de flujo del proceso de diseño y soporte de la industria microelectrónica. Cada vez que la industria de fabricación de

circuitos integrados (*foundry*) introduce una mejora en la tecnología es necesario un proceso de obtención de nuevos modelos que reproduzcan fielmente el comportamiento de dicha tecnología. Esta actividad produce unos modelos que deben acompañarse con un conjunto de parámetros que determinan las características concretas de la tecnología que está disponible en la industria. Para que los diseñadores puedan hacer uso de esos modelos y parámetros, se deben incluir dentro de los simuladores de circuitos convencionales de modo que se pueda reproducir el comportamiento de un circuito teniendo en cuenta las peculiaridades de la tecnología. Las empresas que desarrollan los simuladores deben incluir en sus nuevas versiones todos los modelos que son utilizados por las diferentes industrias de semiconductores en cada momento [MOT09b].

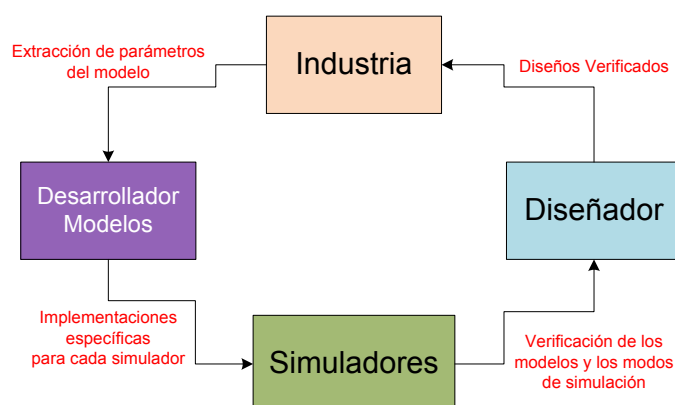


Figura 1.3 – Diagrama de flujo del proceso de diseño y soporte de la industria microelectrónica.

La elección del modelo compacto a utilizar en el diseño de un producto electrónico debe realizarse entre los modelos disponibles del simulador y los facilitados por la industria donde se va a fabricar (en este último caso, la empresa que comercializa el simulador debe incorporarlos) [LCMK03]. Existen multitud de modelos de dispositivos en el mercado que realizan la misma función. Sutiles variaciones en algunos casos, y grandes diferencias en otros, obligan a conocer tanto sus fundamentos físicos como tecnológicos. Dada la amplia oferta de modelos y la cantidad de simuladores comerciales disponibles (ver figura 1.4.a), a veces, es muy difícil incluir toda la oferta de modelos en todos los simuladores.

Desde el punto de vista de un desarrollador de modelos es necesario disponer de una plataforma que permita la interoperabilidad de un nuevo modelo compacto con todos los simuladores potencialmente disponibles por los diseñadores de circuitos. Esta solución genérica es alcanzada mediante la codificación del modelo compacto en *Verilog-A*, ver figura 1.4.b. Como actividad previa al inicio de un diseño se debe elegir el modelo concreto a utilizar. Es necesario realizar una serie de planteamientos antes de elegir el modelo:

- ¿Qué tipo de tecnología voy a utilizar en la fabricación?
- ¿Qué precisión necesito de mi modelo?
- ¿Entiendo las limitaciones que la tecnología tiene?
- ¿Dispongo de los ficheros de parámetros completos o parciales?
- ¿Tengo experiencia en el uso de modelos más complejos?

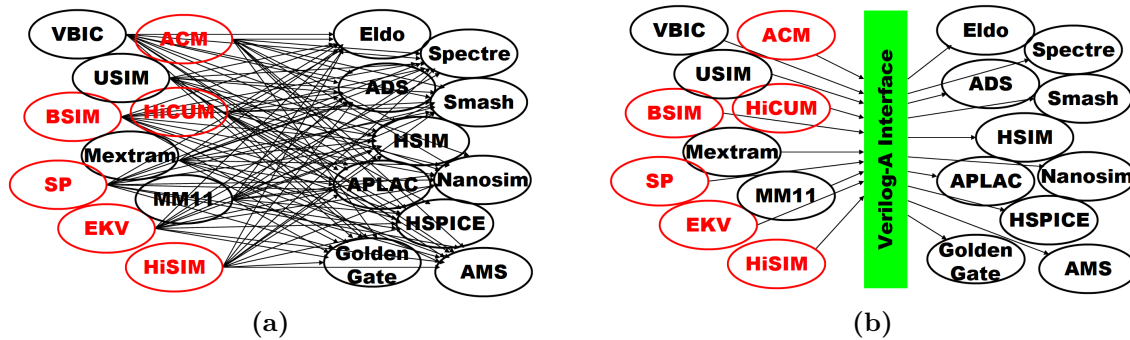


Figura 1.4 – Oferta de modelos compactos de dispositivos MOSFET y simuladores del mercado (a) y solución de interoperabilidad de modelos compactos con los simuladores de circuitos mediante el uso de *Verilog-A* (b).

- ¿De cuánto tiempo dispongo para hacer la caracterización del dispositivo?
- ¿Dispongo de algún extractor de parámetros comercial como Aurora, BSIM2EKV, IC-CAP, MBP, MOSTXX, UTMOS ... ?
- ¿Necesito utilizar el modelo en diferentes simuladores ? (ADS, AMI-Spice, Antrim-A/MS, APLAC, AVOSpice, ELDO, IntuSoft, HSIM, LTspice/SwitcherCAD, HSpice, MacSpice, Micro-CapV, MINIMOS-NT, MI-SUGAR, NanoSpice, Nexxim, NG-Spice, Pspice, Qucs, SABER, SANCAD, SIMetrix, SmartSpice, SMASH, Spectre, SpectreRF, SPICE3, Spice-Opus, TopSPICE, TRANZ-TRAN, TSpice, WinSpice (con soporte de Verilog-A))
- ¿Necesito realizar simulaciones a diferentes temperaturas?
- ¿En qué zona de operación del modelo voy a trabajar?

Una vez se resuleven las respuestas a las cuestiones anteriores podemos elegir entre los modelos disponibles que mejor se adapten a las necesidades del diseño. En la figura 1.5 se presenta la evolución del número de parámetros de los modelos compactos de dispositivos MOSFET frente al año de aparición del modelo [Gra10]. Se puede observar el aumento significativo del número de parámetros que incluyen y su progresión debida al continuo escalado de los transistores.

1.5 Descripción de modelos compactos desarrollados en código fuente

Los modelos compactos forman una parte esencial de los simuladores de circuitos. En la figura 1.6 se muestra un esquema que describe la interacción del simulador y el modelo. Para que el modelo pueda ser usado por el simulador, es necesario que la estructura del modelo sea compatible con la arquitectura del simulador. El diseñador de modelos debe conocer el interior del simulador para poder implementar un modelo de manera eficiente en el simulador. El lenguaje de programación y la distribución de ficheros de funciones y estructuras de datos determinan las características del modelo. Es necesario dotar de un control de versiones al modelo de manera que el diseñador pueda en cada momento ofrecer

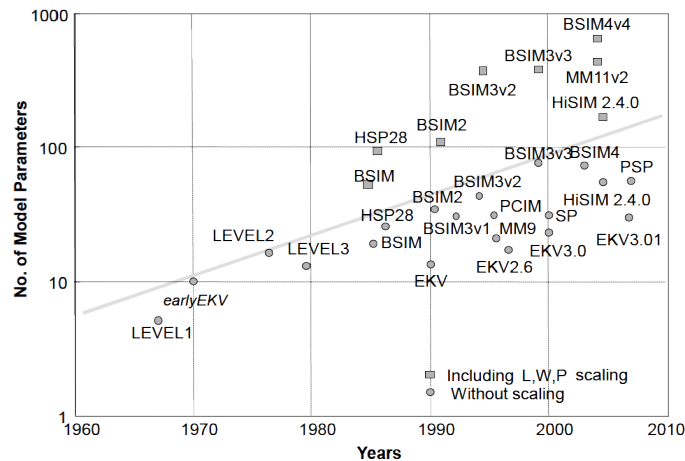


Figura 1.5 – Evolución número de parámetros de los modelos de *MOSFET*.

el listado de capacidades y limitaciones que el modelo presenta, y cómo evoluciona con el tiempo. Los modelos desarrollados en código fuente requieren una programación sistemática,

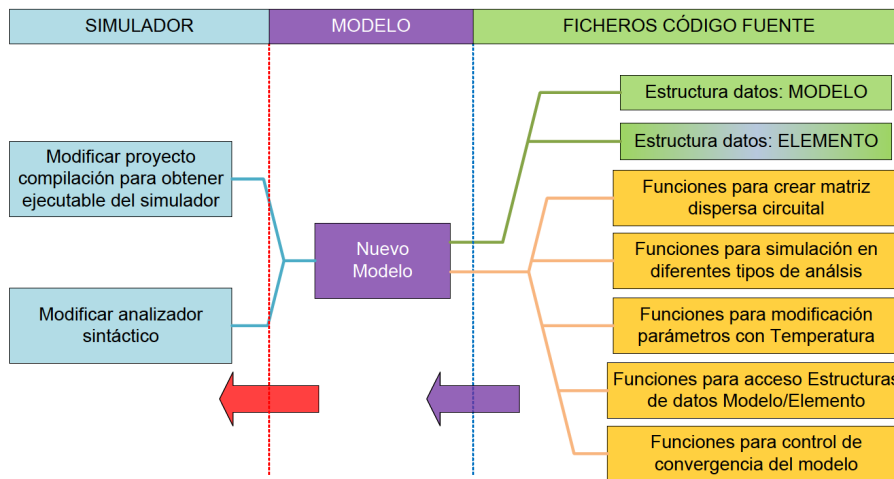


Figura 1.6 – Diagrama de bloques de un modelo en código fuente y su conexión con el núcleo del simulador.

revisión de las funciones que se implementan y grandes destrezas en el uso del lenguaje en el que el núcleo del simulador está codificado. Los modelos confeccionados para versiones de simuladores en *FORTRAN* no son compatibles con los codificados en *lenguaje-C* o *Verilog-A*, y la portabilidad queda limitada a la versión y lenguaje de programación usado en el núcleo del simulador. Los modelos compactos necesitan normalmente dos estructuras de datos para describir el *modelo* de dispositivo y los parámetros tecnológicos que comparten todos los *elementos* que se incluyen en el circuito. En el simulador, todos los dispositivos (*elementos*) que poseen parámetros tecnológicos idénticos se agrupan bajo una estructura de datos denominada *modelo*, ver figura 7.22. Tanto el *modelo* como el *elemento* necesitan un listado de parámetros por defecto con el formato necesario (valor entero, real, complejo, cadena de texto, etc.). Además, hace falta un conjunto de funciones específicas para añadir a la matriz dispersa que contiene los datos correspondientes a las ecuaciones circuitales



establecidas por la topología de conexión de los elementos del circuito, implementar las modificaciones de los parámetros con el cambio de temperatura, adaptar el comportamiento a los diferentes tipos de análisis, control de la convergencia de las variables de estado del modelo y realizar las tareas de sincronización con el núcleo del simulador. El modelo se integra mediante la inclusión en el proyecto de compilación del simulador de la información relativa a la ruta de los ficheros en código fuente del modelo y las opciones de enlazado necesarias para la obtención del nuevo ejecutable. Para que el simulador pueda atender e interpretar correctamente las instrucciones relativas al modelo compacto es necesario que sea reconocido por el analizador sintáctico. De esta manera, se enlaza el simulador con las funciones que implementan el modelo en código fuente que siguen el estándar definido por los creadores del simulador.

En los siguientes apartados se introducen brevemente las características más importantes de cada uno de los bloques del simulador. En el anexo correspondiente se describen los detalles técnicos de bajo nivel que presentan las características de las funciones y estructuras de datos de los modelos en código fuente.

1.5.1 Estructuras de datos para sistemas de ecuaciones circuitalas.

La formación automática de las ecuaciones de un circuito a partir de un fichero de descripción topológica queda fuera del alcance de esta tesis. Es necesario, sin embargo, introducir una breve referencia al proceso de establecimiento de la matriz dispersa que soporta esos datos dentro del simulador. La eficiencia de un simulador de circuitos queda condicionada por las características de esta estructura de datos, que suele implementarse mediante una lista ortogonal enlazada de nodos [Qua89b]. En la figura 1.8 se muestra la citada lista enlazada correspondiente al conjunto de ecuaciones 1.7(b) del circuito de la figura 1.7(a). Para disminuir el tamaño de la estructura sólo los elementos no nulos de la matriz son almacenados. Cada nodo de la matriz está enlazado mediante dos punteros a los elementos inferior y derecha, por lo tanto, cada elemento estará definido por un conjunto: (valor del elemento, número de la fila, número de la columna, puntero al elemento inferior, puntero al elemento contiguo situado a la derecha). A la hora de realizar las manipulaciones sobre los elementos de la matriz de manera sencilla se utilizan tres vectores normalizados:

- **FIC:** (**F**irst **e**lement **I**n **e**ach **C**olumn) - Primer elemento de cada columna.
- **FIR:** (**F**irst **e**lement **I**n **e**ach **R**ow) - Primer elemento de cada fila.
- **D:** elementos diagonales de la matriz.

En la figura 1.8 se presenta la lista ortogonal enlazada referente al sistema de ecuaciones 1.7(b). En el apartado 7.3.2.3 del Anexo se introducen todos los detalles de la implementación en lenguaje-C de la estructura de datos *SMPmatrix* original de SPICE2 que ha sido mejorada en SPICE3.

1.5.2 Estructuras de datos y funciones para modelos compactos.

El proceso de incorporación de un nuevo modelo compacto al núcleo del simulador [Qua89a] se divide en cuatro fases que se representan en la figura 1.9:

1. Crear la estructura datos dinámica para el modelo y los elementos modelados.
2. Crear funciones para el dispositivo y la matriz dispersa.
3. Modificar el analizador sintáctico para que procese el modelo y parámetros en el circuito.
4. Comprobar correcto funcionamiento sintáctico y numérico del modelo en circuitos.

En los próximos apartados, y en el Anexo 7.4, se introducen los detalles de las fases descritas en la figura 1.9 y el código asociado a ellas.

1.5.2.1 Creación de rutinas específicas del nuevo modelo

Cada dispositivo se describe por una estructura de datos (ver Anexo 7.3.1) que contiene punteros a funciones que realizan las operaciones específicas del dispositivo y tablas que almacenan los parámetros del dispositivo. También incluyen punteros a tablas y datos que son utilizados desde el interfaz de datos de usuario y las rutinas del interfaz del simulador. Esta estructura de datos es la única parte visible externamente del código específico del dispositivo, el resto es referenciado a través de los punteros a funciones contenido en la estructura.

Hay dos estructuras necesarias para cada dispositivo: una para el modelo y otra para el elemento (ver figura 7.22). La estructura de modelo contiene todos los parámetros que comparten varios elementos en común como resistencia por superficie, capacidad por área, etc. Estos parámetros son específicos al proceso usado para la fabricación del dispositivo. Hay una cabecera estándar que describe las cuatro primeras entradas en la estructura de modelo que deben aparecer en ese orden. El resto pueden organizarse según el orden que establezca el desarrollador del modelo. En la figura 7.22 del Anexo 7.3.1.1 se detalla la estructura modelo-elemento

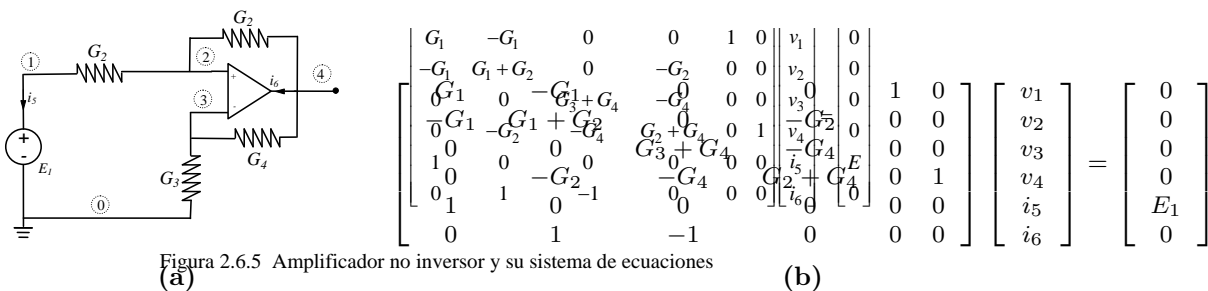
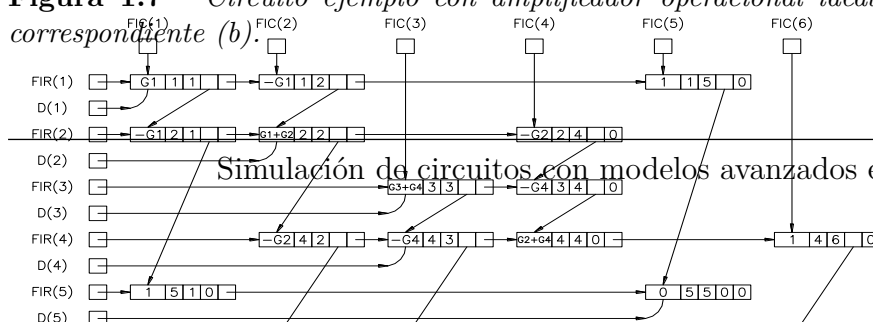


Figura 2.6.5 Amplificador no inversor y su sistema de ecuaciones

Figura 1.7 – Circuito ejemplo con amplificador operacional ideal (a) y sistema de ecuaciones correspondiente (b).



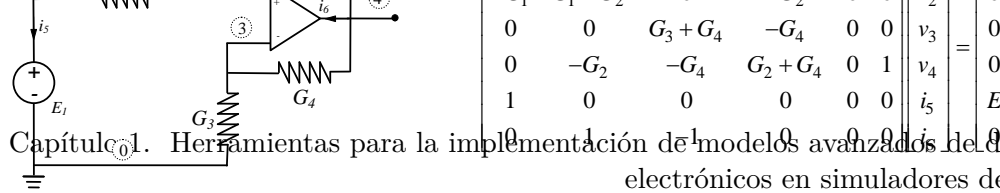


Figura 2.6.5 Amplificador no inversor y su sistema de ecuaciones

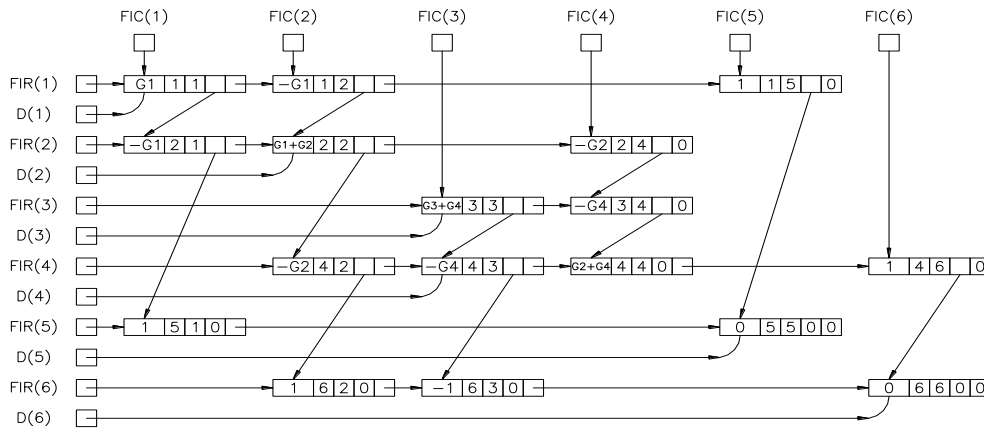


Figura 1.8 – Lista ortogonal enlazada de nodos para representar las ecuaciones del circuito de la figura 1.7(b).

Figura 2.6.6 Lista ortogonal enlazada

2.6.2 Métodos Indirectos

Los métodos indirectos se utilizan con grandes sistemas de ecuaciones de ventajas de los directos que se pueden resumir en:

1. Complejidad computacional. $O(n) = n^3$ prohibitiva en grandes sistemas aunque en circuitos electrónicos y sistemas dispersos se puede alcanzar $O(n) = n$.
2. Coste computacional de almacenamiento de datos altos al existir un gran número de elementos.
3. El error crece linealmente con el tamaño del sistema.

Las características generales de los métodos indirectos que los recomiendan para su uso con sistemas de ecuaciones de grandes dimensiones son:

1. Estiman una solución inicial y la mejoran sucesivamente hasta que se alcanza un criterio de error.
2. El coste computacional asociado a este tipo de métodos es siempre menor en un computador de precisión finita que en métodos de resolución directa.
3. La convergencia del método depende de las propiedades de la matriz de coeficientes.

Figura 1.9 – Descripción de las fases de creación/modificación de un modelo compacto en código fuente.

1.5.2.2 Descriptores de dispositivos

2.6.2.1 Métodos Estacionarios

Además de lo establecido en el apartado anterior, existen dos estructuras que albergan los descriptores de parámetros del modelo y del elemento. Cada listado contiene todos los parámetros disponibles además de la información necesaria por las rutinas que manejan dichos parámetros, su tipo, una bandera de existencia, etc. El tipo de los parámetros puede ser: entero, real, *flag*, nodo, complejo, cadena, instancia o *parsetree* combinados con uno o más modificadores. Estas constantes simbólicas indican que el parámetro requiere un valor entero, real, ningún valor, un nodo del circuito, un complejo, una cadena de caracteres, el identificador de otro dispositivo o un árbol de análisis. Estos tipos de datos son soportados por el analizador que interpreta los datos almacenados en el fichero de descripción topológica o en el interfaz de entrada/salida.

Se presenta el código fuente asociado a la descripción del dispositivo para establecer el conjunto de funciones necesarias a implementar para permitir los diferentes tipos de análisis del simulador, así como la manipulación de los datos internos del modelo. La sintaxis utilizada es de [lenguaje-C](#), y se han introducido los comentarios necesarios para la comprensión del significado de cada línea.

```

1 typedef struct SPICEdev {
2 struct{
3     char *name;           // Nombre del dispositivo
4     char *desccrptoq;    // Descripción del tipo de dispositivo
5     int terms;           // Número de terminales del dispositivo
6     int numNames;        // Número de nombres en el array termNames
7     char **termNames;    // Puntero al array de punteros de los nombres.
8
9     int numInstanceParms; // Número de parámetros existentes para los elementos
10    IFparm *instanceParms; // Array con las descripciones de los parámetros de los elementos
11
12    int numModelParms;    // Número de parámetros existentes para el modelo
13    IFparm *modelParms;   // Array con las descripciones de los parámetros del modelo
14
15 } DEVpublic;
16
17 int (*DEVparam)();       // Para introducir un parámetro en una instancia de un elemento
18 int (*DEVmodParam)();   // Para introducir un parámetro en una instancia de un modelo
19 int (*DEVload)();       // Para cargar un dispositivo en la matriz del sistema
20 int (*DEVsetup)();      // Para preprocesar los dispositivos antes del comienzo del proceso
21 iterativo de solución
22 int (*DEVpzSetup)();    // Para realizar el análisis polo-cero
23 int (*DEVtemperature)(); // Para realizar el procesado de los parámetros dependientes de la
24 temperatura
25 int (*DEVtrunc)();      // Para realizar el cálculo del error de truncamiento
26 int (*DEVfindBranch)(); // Para realizar la búsqueda de las ecuaciones de malla del
27 dispositivo.
28 int (*DEVvacLoad)();    // Para realizar la carga ante un análisis AC.
29 int (*DEVaccept)();     // Para confirmar la aceptación de time-point
30 void (*DEVdestroy)();   // Para liberar la memoria consumida por un modelo y elemento.
31 int (*DEVmodDelete)();  // Para borrar un modelo y todas sus instancias.
32 int (*DEVdelete)();     // Para borrar una instancia de un modelo.
33 int (*DEVsetic)();      // Para establecer las condiciones iniciales de elemento.
34 int (*DEVask)();        // Para obtener los detalles de un dispositivo.
35 int (*DEVmodAsk)();     // Para obtener los detalles de un modelo.
36 int (*DEVpzLoad)();     // Para realizar la carga ante un análisis PZ
37 int (*DEVconvTest)();   // Para realizar un test de convergencia en la iteración.
38 int (*DEVsenSetup)();   // Para establecer la información de sensibilidad de un dispositivo.
39 int (*DEVsenLoad)();    // Para realizar la carga ante un análisis de sensibilidad SENS.
40 int (*DEVsenUpdate)();  // Para actualizar la información de sensibilidad de un dispositivo.
41 int (*DEVsenAcLoad)();  // Para realizar la carga de los valores AC ante un análisis de
42 sensibilidad SENS.
43 void (*DEVsenPrint)();  // Para realizar impresión de los resultados de la información de
44 sensibilidad.
45 int (*DEVsenTrunc)();   // Para comprobar error de truncamiento durante el análisis de
46 sensibilidad.
47
48 int DEvinsSize;        // Tamaño de una instancia o elemento.
49 int DEVmodSize;        // Tamaño del modelo.
50 } SPICEdev;           // Estructura de datos necesaria para cada tipo de dispositivo.

```

Listado 1.1 – *Estructura SPICEdev.*

Como se puede observar en el listado 1.1, la estructura del dispositivo es de gran complejidad por el uso de punteros a estructuras y funciones, y por la gran cantidad de procedimientos independientes para el análisis y el acceso a valores internos del dispositivo. Es necesario asegurar una codificación eficiente y evitar los problemas típicos de desbordamiento al intentar acceder mediante punteros a funciones no implementadas.

1.5.2.3 Modificación de las rutinas existentes.

El procedimiento de creación y modificación de modelos en código fuente para incorporarlos a los simuladores se ha trasladado al apartado 7.4 del Anexo donde se detalla el conjunto mínimo de ficheros necesario a modificar. El nivel de detalle necesario para describir el procedimiento nos ha llevado a incorporarlo en el anexo por mantener la continuidad en

la redacción de esta explicación.

1.5.3 Ventajas e inconvenientes de los modelos en código fuente.

El desarrollo de modelos compactos en código fuente ha estado limitado durante mucho tiempo por la necesidad de compilar dicho modelo dentro de un simulador de circuitos. El acceso al código fuente del simulador, su estudio y conocimiento profundo han sido requisitos obligados para los desarrolladores de modelos compactos y programadores [LG06]. La sintaxis establecida en lenguaje-C por los creadores de SPICE y el uso de estructuras de datos y funciones de acceso a los parámetros internos de los modelos y elementos ha obligado a los desarrolladores de modelos a tener grandes conocimientos de programación. Además, los modelos desarrollados en código fuente han sido muy dependientes de la arquitectura del núcleo del simulador haciendo muy difícil, por no decir imposible, la portabilidad entre los diferentes simuladores existentes en el mercado. Los modelos compactos se implementaban de manera diferente según el simulador al que iban ser agregados, haciendo muy costosas las tareas de mantenimiento, actualización y distribución entre los diseñadores de circuitos y la industria de semiconductores. En la tabla 1.2 se presentan las características en términos de

Modelo Compacto	Versión	Código fuente		
		Nº Ficheros	Nº Líneas Funciones	Nº Líneas Estruc. Datos
BJT	1	25	6889	588
	2	25	7164	655
MOSFET	BSIM-1	22	7042	708
	BSIM-2	20	4789	800
	BSIM-3	19	11347	1867
	BSIM-3-V0	18	7871	1636
	BSIM-3-V1	19	9093	1688
	BSIM-3-V1a	18	7904	1637
	BSIM-3-V1s	19	8955	1684
	BSIM-3-SOI	19	14563	2521
	BSIM-3-SOI-DD	19	12315	2046
	BSIM-3-SOI-FD	19	11381	2046
	BSIM-3-SOI-PD	19	13213	2302
	EKV-2.0	20	4743	730
	MOS-1	25	7037	570
	MOS-2	25	8396	603
	MOS-3	25	7915	614
MOS-6	13	2931	516	
MOS-9	25	7955	616	
RES	RES	18	1343	206

Tabla 1.2 – Dimensiones de los modelos desarrollados en código fuente.

extensión de los ficheros de código de los principales modelos de MOSFET para SPICE.

1.6 Verilog-A: un lenguaje de descripción de hardware para aplicaciones analógicas y de señal mixta. Compiladores. Paquetes TCAD comerciales para la simulación circuital: módulos de Verilog-A.

Existen en la bibliografía guías de usuario, como por ejemplo la referencia [KZ04], que permiten a los diseñadores de circuitos el desarrollo de modelos de comportamiento de alto nivel pero no están orientadas al desarrollo de modelos compactos analógicos. Ésta es la razón que motiva la introducción de [Verilog-A](#), de modo que se puedan desarrollar modelos compactos que produzcan un código eficiente a partir de un conjunto de ecuaciones analíticas para describir el comportamiento de un dispositivo, sobre todo en un contexto de operación analógica.

1.7 Ventajas del uso de Verilog-A

La ventaja principal del uso de [Verilog-A](#) en modelado compacto sobre cualquier otro lenguaje de programación de propósito general es que libera al desarrollador de la carga de interactuar con el interfaz del simulador de circuitos. Dicho interfaz incluye cientos de primitivas para leer los parámetros de los modelos, chequear la topología, reservar memoria para la estructura de datos que soporta al modelo, inicializar las variables o recalculan éstas frente a cambios de temperatura, actualizar la matriz jacobiana del algoritmo de [Newton-Raphson](#) del analizador de circuitos, etc. Para complicar el proceso aun más, cada fabricante de simuladores ha definido unas primitivas diferentes para hacer el mismo proceso, lo que complica la creación y distribución del modelo compacto y lo hace dependiente del simulador. Existen simuladores con diferentes tipos de análisis disponibles lo que también incrementa el número de primitivas a implementar en un modelo compacto.

Aparte de hacer la programación de modelos compactos independientes del simulador, el compilador de [Verilog-A](#) calcula automáticamente las derivadas parciales simbólicas de las corrientes y cargas en el modelo compacto y determina de manera precisa la correcta introducción de esos valores en la matriz jacobiana del algoritmo del método de [Newton-Raphson](#) que se resuelve de manera iterativa por el simulador. En [lenguaje-C](#) este proceso debe ser realizado a mano por el programador. Sirva a modo de ejemplo que si existe en un simulador un modelo compacto descrito en [lenguaje-C](#) y se procede a modificar alguna ecuación que cree nuevas dependencias, el programador debe volver a recalculan las derivadas con la posibilidad de modificar la estructura de datos necesaria para soportar el modelo.

El desarrollador del modelo compacto está más centrado en obtener las ecuaciones correctas para la corriente y carga y compararlas con las medidas experimentales que en la obtención de las derivadas parciales de dichas expresiones que son difíciles de verificar. Como resultado de esta limitación se han producido en la historia reciente de los modelos compactos para dispositivos MOSFET errores en las expresiones de las derivadas en sus primeras versiones con sus consecuentes efectos.

Además **Verilog-A** proporciona un sistema robusto de definición de parámetros para modelos. La declaración de los parámetros establece un valor por defecto y un rango válido de valores que permite limitar la posibilidad de trabajar por error fuera de rango. El valor por defecto se puede establecer como función de otros parámetros definidos previamente. Esto es de gran utilidad si tenemos en cuenta que algunos de los parámetros por defecto de los dispositivos NMOS y PMOS son diferentes, pero gracias a las especificaciones del lenguaje, pueden ser incorporados directamente en la declaración de parámetros en vez de establecer un procedimiento especial de definición.

Resumiendo, las características de **Verilog-A** descritas anteriormente lo hacen muy eficiente en el desarrollo de modelos compactos. A modo de ejemplo cabe citar que la inclusión del **self-heating** en la versión de **Verilog-A** del modelo de **MOSFET BSIM-3**[BS105] se realizó en aproximadamente 2 días mientras que su inclusión en el modelo nativo en **C** se extendió durante dos semanas. También, es de destacar el hecho de que la implementación en **Verilog-A** de un modelo ha servido para detectar errores tipográficos en la implementación en **lenguaje-C** de las derivadas parciales en varios modelos.

1.8 Metodología para la implementación de un modelo compacto en Verilog-A.

Verilog-A es un lenguaje sencillo. Todos los conceptos necesarios pueden ser aprendidos directamente mediante la lectura detallada de un ejemplo como el que se presenta en la figura 1.10 y el Listado 1.2.

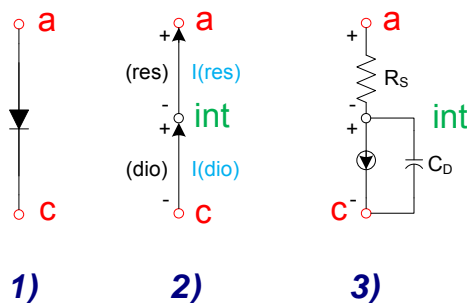


Figura 1.10 – Modelo compacto de un diodo.

```

1  `include "disciplines.h"
2  `include "constants.h"
3  module diode(a,c);
4      inout a,c;
5      electrical a,c,int;
6      branch (a,int) res;
7      branch (int,c) dio;
8
9      parameter real is = 10p from (0:inf);
10     parameter real rs = 0.0 from [0:inf];
11     parameter real cjo = 0.0 from [0:inf];
12     parameter real vj = 1.0 from (0:inf);
13     `ifdef __VAMS_COMPACT_MODELING__
14     aliasparam phi = vj;
15     (*desc="jct. voltage"*) real vd;
16     (*desc="current"*) real id;
17     (*desc="depl. charge"*) real qd;
18     (*desc="depl. cap."*) real cd;
19     (*desc="conductance"*) real gd;
20     `define GMIN ($simparam("gmin"))

```

```

21 `else
22     real vd, id, qd;
23     `define GMIN (1.0e-12)
24 `endif
25
26 analog begin
27     V(res) <+ I(res)*rs;
28     vd = V(dio);
29     id = is*(limexp(vd/\$vt) - 1.0);
30     if (vd < vj)
31     begin
32         qd = cjo*vj*(1.0-2.0*sqrt(1.0-vd/vj));
33     end
34     else
35     begin
36         qd=cjo*vd*(1.0+vd/(4.0*vj));
37     end
38 `ifdef VAMS COMPACT MODELING
39     gd = ddx(id, V(int));
40     cd = ddx(qd, V(int));
41 `endif
42
43     I(dio) <+ id+'GMIN*vd;
44     I(dio) <+ ddt(qd);
45     I(dio) <+ white noise(2.0*'P_Q*id, "shot");
46     V(res) <+ white noise(4.0*'P_K*$temperature*rs, "thermal");
47 end
48 endmodule

```

Listado 1.2 – Modelo compacto de un diodo.

Los módulos de Verilog-A como el que se presenta en el listado 1.2 reemplazan al conjunto de primitivas necesarias en los simuladores compatibles con SPICE también conocidos como (*Spice's Like Simulator*). Los módulos pueden ser utilizados directamente en las descripciones de circuitos o *netlist*, como si fueran primitivas o subcircuitos. Los módulos de Verilog-A pueden contener a su vez a otros módulos siendo un híbrido entre primitivas y subcircuitos.

En general, todos los módulos poseen un conjunto de puertos que representan los terminales de los dispositivos, en el listado anterior representados como a y c entre los paréntesis dentro de la cabecera del módulo, línea 3, ver figura 1.10.1. Los terminales del modelo compacto deben ser definidos como *inout*, línea 4 del listado 1.2. Esta sintaxis recuerda el origen del lenguaje Verilog como herramienta para el modelado lógico digital y la dirección indicando el flujo de señal. Del mismo modo, los terminales deben ser tipificados como *eléctricos*, línea 5 del listado 1.2. La definición *eléctrica* se importa mediante el uso de la primitiva de la línea 1, que contiene las definiciones de las *disciplinas* existentes. Estas disciplinas son esenciales al definir los diferentes tipos de datos que se usarán dentro del modelo compacto, las unidades y las tolerancias absolutas. En el modelado compacto, la disciplina más común es la eléctrica, que engloba a las variables de tensión y corriente. También se puede establecer un módulo de modelado de autocalentamiento para el que será necesario usar la disciplina térmica (*thermal discipline*). Del mismo modo, se pueden definir nodos internos (*int*, línea 5) declarando su disciplina, pero sin incluirlo en la lista que sigue al nombre del módulo, líneas 3 y 4 del listado 1.2.

En este ejemplo se declaran dos ramas internas (*res* y *dio*) figura 1.10(b), líneas 6 y 7, que representan nemónicos los pares de nodos que las forman: *res*=(a-*int*) y *dio*=(*int*-c). Tras las declaraciones de los puertos, nodos y mallas internas se introducen las definiciones de los parámetros que deben incluir un valor por defecto, líneas 9-12. Estos valores pueden hacer uso de números simples, con notación científica u operaciones con parámetros anteriormente definidos. Se pueden añadir rangos válidos de valores como por ejemplo *from*

(0:inf). Siguiendo la definición del estándar la notación matemática de corchetes [] indica que el rango incluye los extremos y en el caso de los paréntesis () no se incluyen. Existe la posibilidad de excluir valores específicos en los parámetros si el usuario hace uso de la primitiva `exclude` lo que permite al simulador generar un mensaje de error en el caso de que no se respete el rango del parámetro recibido por el módulo.

Se pueden usar extensiones del lenguaje para hacer *alias* de parámetros, línea 13, de modo que si el simulador la tiene implementada en su versión de lenguaje, permitirá que `vj` sea especificado como `phi` en el módulo. En el ejemplo se presenta una primitiva de compilación condicional que permite que el módulo sea utilizable incluso en simuladores que no dispongan de la extensión del lenguaje. El acento grave (`) sustituye a la directiva de compilación (#) del `lenguaje-C`. La etiqueta `__VAMS_COMPACT_MODELING__` está predefinida en el compilador de `Verilog-A` que soporta la extensión del lenguaje. A continuación se declaran las variables, que como en el `lenguaje-C`, debe realizarse al comienzo del bloque de instrucciones, líneas 15-19 y 22. Aquellas definiciones que van precedidas de la primitiva (`*desc=...*`), permiten gracias a la extensión del lenguaje, exportar los valores de la variable a través del interfaz del simulador de modo que están accesibles para su acceso al usuario, o para su impresión como parte de la información generada en el análisis del punto de operación `OP`. En caso de que el simulador no disponga de la extensión del lenguaje, la compilación condicional procesará la línea 22 donde se definen las variables reales `vd`, `id`, `qd`. Otra posibilidad que ofrece `Verilog-A` es la de poder acceder a valores internos del simulador, mediante la primitiva `$simparam(...)` de la línea 20, o `\$vt` o `\$temperature` de la línea 46 que proporcionan una mayor flexibilidad y donde el carácter \$ indica una función de sistema.

La definición del comportamiento del módulo se define dentro del bloque `analog`. Un módulo sólo puede contener un bloque `analog` que puede estar constituido por gran cantidad de sentencias. El bloque se define con las primitivas `begin` y `end` que sustituyen a las clásicas `{}` del `lenguaje-C`.

`Verilog-A` utiliza la primitiva de sintaxis `\textless +` para indicar una contribución en una tensión o corriente de una malla. Con la expresión de la línea 27, `V(res) <+I(res)*rs;` estamos estableciendo que la tensión de la malla `res` definida entre los nodos `a` e `int` será aumentada por el valor de la corriente que atraviesa esa misma malla por el valor de la resistencia serie del `rs`. Esta descripción de las relaciones entre tensiones y corrientes permite de manera cómoda aceptar cualquier valor de `rs` sin efectuar ninguna operación aritmética de división por cero que interrumpiría la ejecución de la simulación. El modelo de diodo estudiado posee diferentes contribuciones de corriente para la componente continua y otra para los efectos de alterna derivados del condensador interno.

Las funciones matemáticas disponibles en lenguaje C también están accesibles en `Verilog A`. Los operadores más frecuentes en el modelado compacto de dispositivos electrónicos son `+`, `-`, `*`, `/` y las funciones `abs`, `pow`, `sqrt`, `ln` y `exp`. Es importante destacar que existe una función que mejora la convergencia en las uniones semiconductoras `limexp` y se recomienda su uso en detrimento de `exp`.

Las corrientes provenientes de capacidades pueden ser definidas usando el operador derivada sobre la expresión de la carga mediante el uso del operador `ddt`. Existe una extensión

del lenguaje que además permite el cálculo de la derivada parcial de una expresión con respecto a una variable mediante el operador `ddx(expresión,variable)` como se muestra en las líneas 39,40. Este operador es muy interesante para el cálculo de los parámetros básicos de modelos compactos como son conductancias, transconductancias y capacidades.

Estructuras condicionales pueden ser utilizadas con las primitivas `if` y `else`, que pueden utilizar operadores lógicos como `&&`, `||`, `!`, los operadores relacionales `<`, `>`, `<=`, `>=` y los operadores de igualdad `==` y `!=`. Existen estructuras de selección `switch:case` y bucles `for` aunque no son muy usuales en modelos compactos.

Como complemento a las capacidades de Verilog-A se añade la posibilidad de incorporar fuentes de ruido de pequeña señal mediante las primitivas `flicker_noise` y `white_noise`. El ruido térmico y *shot*, ambos ruidos blancos, están presentes en el modelo de diodo en las líneas 45 y 46. La constante de Boltzmann y la carga del electrón se obtienen directamente del archivo de cabecera de constantes, la línea 2 que contiene constantes físicas y matemáticas con prefijos `P_` y `M_` respectivamente.

1.8.1 Nuevas funcionalidades de Verilog-A

Existen otros recursos de interés en el modelado compacto de dispositivos con Verilog-A [AI08] que no han sido introducidos en el ejemplo anterior y que hacen a este lenguaje mucho más sencillo que el lenguaje-C para la implementación de modelos compactos.

1.8.1.1 Parámetros de los modelos.

La definición de parámetros de modelo puede tener expresiones por defecto creadas a partir de parámetros declarados previamente, línea 1 del listado 1.3. En el caso de definir parámetros que describen diferentes modelos de un fenómeno físico, como el caso de la movilidad electrónica [HN06], línea 2, se debe hacer uso de una variable *tipo entero* con un rango de valores válidos.

```

1 | parameter real res= rho*length/width;
2 | parameter integer mobmod= 1 from [1:3]
3 | parameter real uc= (mobmod==3) ? -46.5E-3 : -46.5E-12
4 | parameter real p2= 0 from [0:1];
5 | parameter real p3= 0 from [0:1-p2];

```

Listado 1.3 – Otras posibilidades

Una vez establecido el valor de `mobmod = [1, 2, 3]` se podrá obtener el valor la movilidad, haciendo uso de una estructura condicional, mediante la evaluación de una expresión matemática como la que se presenta a modo de ejemplo para el modelo compacto de transistor MOSFET BSIM 4.5 [DXH+06].

$$\mu_{\text{eff}} = \begin{cases} \frac{U0 \cdot f(L_{\text{eff}})}{1 + (UA + UC \cdot V_{\text{bseff}}) \left(\frac{V_{\text{gsteff}} + 2V_{\text{th}}}{TOXE} \right) + UB \left(\frac{V_{\text{gsteff}} + 2V_{\text{th}}}{TOXE} \right)^2 + UD \left[\frac{V_{\text{th}} \cdot TOXE}{V_{\text{gsteff}} + 2V_{\text{th}}} \right]^2} & \text{para mobmod}=0 \\ 1 + \frac{U0 \cdot f(L_{\text{eff}})}{\left[UA \left(\frac{V_{\text{gsteff}} + 2V_{\text{th}}}{TOXE} \right) + UB \left(\frac{V_{\text{gsteff}} + 2V_{\text{th}}}{TOXE} \right)^2 \right] (1 + UC \cdot V_{\text{bseff}}) + UD \left[\frac{V_{\text{th}} \cdot TOXE}{V_{\text{gsteff}} + 2V_{\text{th}}} \right]^2} & \text{para mobmod}=1 \\ \frac{U0 \cdot f(L_{\text{eff}})}{1 + (UA + UC \cdot V_{\text{bseff}}) \left[\frac{V_{\text{gsteff}} + C_0(V_{\text{TH0}} - V_{\text{FB}} - \Phi_s)}{TOXE} \right]^{EU} + UD \left[\frac{V_{\text{th}} \cdot TOXE}{V_{\text{gsteff}} + 2V_{\text{th}}} \right]^2} & \text{para mobmod}=2 \end{cases}$$

También se puede incluir código condicional que defina el valor de un parámetro a partir de otros anteriores. En el listado, línea 3, se presenta el parámetro `uc` del modelo I-V que introduce el coeficiente de degradación de la movilidad por efecto cuerpo (*Body-effect of mobility degradation coefficient*) [Gro05] con valor -4.65E-11 si el parámetro `mobmod` vale 1 ó 2, y 0.0465 si `mobmod`=3. Además, se podría introducir el rango de valores recomendados para dicho parámetro [-1E-12, -1E-8] para `mobmod`=1,2 y [-0.1, -0] para `mobmod`=3.

1.8.1.2 Dispositivos NMOS y PMOS

Las expresiones utilizadas en el modelado compacto de dispositivos semiconductores se definen generalmente para dispositivos tipo-N, *i.e.*, NMOS o NPN. Las ecuaciones que se presentan en la bibliografía también describen dispositivos tipo-N, para evitar tener que implementar las ecuaciones dos veces dentro de un modelo compacto para dar respuesta a los dispositivos N ó P existe un método sencillo utilizando un parámetro `TYPE` definido de manera similar a la expresada en la línea 1 del listado 1.4, donde el parámetro puede tomar únicamente los valores 1 y -1, y permite cambiar la definición de las tensiones y corrientes para dispositivos tipo-P utilizando las mismas expresiones. Esto conlleva un código más eficiente y de mayor simplicidad.

```

1 parameter integer TYPE= 1 from [-1:1] exclude 0;
2 ...
3 Vds= TYPE*V(drain , source)
4 ...
5 I(drain , source) <+ TYPE*id_channel

```

Listado 1.4 – Dispositivos de tipo N ó P, utilización del parámetro `TYPE`.

El signo que aplica el parámetro `TYPE` sobre las expresiones únicamente es válido para las tensiones y corrientes; en el caso de la temperatura y la potencia eléctrica para la red de autocalentamiento no se introduce variación alguna.

1.8.1.3 Errores frecuentes en la codificación.

Una de las ventajas más importantes de `Verilog-A` es el cálculo automático de las derivadas parciales de las expresiones de corriente o carga para describir parámetros de pequeña señal, ruido, etc [MOT09a]. Pero existen otras partes del código que siguen siendo susceptibles de incorporar errores de usuario, que son fáciles de detectar en desarrollos previos realizados en `MATLAB` o programas similares. Uno de estos errores más frecuentes es el uso incorrecto de la llamada a la función logaritmo (`log` para el caso de base 10 y `ln` para base

natural).

1.8.1.4 Excepciones de punto flotante y derivadas ocultas.

Muchas de las funciones matemáticas utilizadas de forma usual en los modelos de los dispositivos semiconductores implican el uso de funciones exponenciales. Ésto puede generar valores desproporcionados de tensiones durante las iteraciones de [Newton-Raphson](#) [MOT⁺04]. Para evitar este problema frecuente, se recomienda el uso de funciones limitadoras como `limexp` que controlan el valor que puede tomar la función `exp`.

La función potencia (`pow(x, y)`) que devuelve el valor x^y y puede ser muy alto, no está definida para algunos valores del argumento, y su derivada no está definida para valores en los que sí lo está la función. En algunos casos se recomienda el cambio de la función `pow(a, b)` por su equivalente `exp(b*ln(a))` sobre todo cuando `a` está elevada a otras potencias.

El error de división por cero suele aparecer en muchos casos. Un ejemplo clásico se produce en el caso de que exista una relación `y=sqrt(x)`. Aunque la función no tiene problemas de división por cero, la derivada de ésta (`y'=1/sqrt(x)`) sí los presenta si `x` depende de una tensión nodal. Aunque a veces el simulador automáticamente sustituya la derivada por `y'=1/sqrt(x+epsilon)` puede producirse que `epsilon` sea mayor que `x` en ciertas circunstancias, lo que implicaría errores numéricos en la evaluación.

Otro caso similar al anterior, se presenta al hacer uso de la función valor absoluto (`abs(x)`) que no tiene derivada para `X=0`, y que por lo tanto su uso no está recomendado en expresiones que dependan de las tensiones. En el modelado térmico se pueden presentar expresiones como `Temp=$temperature+DTA+abs(V(dt))` que en la iteración inicial en la que las tensiones son asignadas a valores nulos, provocaría un error en la derivada.

1.8.1.5 Técnicas de la mejora de la convergencia de circuitos con la utilización de modelos en Verilog-A.

Las ecuaciones que describen los modelos compactos son usualmente no-lineales, y con frecuencia el método iterativo de [Newton-Raphson](#) utilizado por el núcleo del simulador no puede encontrar una solución, particularmente para los casos de análisis en punto de operación (DC analysis) o en la iteración inicial (`time=0`) de los análisis transitorios. Una estrategia utilizada por los simuladores para evitar este problema es el uso de un algoritmo de modificación. Este tipo de algoritmos se denomina *damping algorithms* y se utilizan en estrategias «*limiting*». En la actualidad, [Verilog-A](#) sólo soporta la utilización de estos algoritmos mediante el uso de la primitiva `limexp`. La primitiva `limexp` es de uso específico con la función exponencial, pero existen muchas otras funciones de alta no-linealidad que causan problemas de convergencia al algoritmo de [Newton-Raphson](#). Otra peculiaridad de la primitiva `limexp` es que se considera un operador analógico de lenguaje, lo que restringe su uso. Por ejemplo, en los modelos de transistores bipolares se hace uso frecuente de la ecuación de corriente del diodo con diferentes tensiones de alimentación y corrientes de saturación. Sería interesante unificar las diferentes ecuaciones en una única función analógica

que devolviera el valor numérico, pero por sus características de operador analógico, `limexp` no podría ser usado en este contexto. El modelo de SPICE de BJT también hace uso de la misma tensión de alimentación en el cálculo de las corrientes en dos diodos («ideal» y «nonideal»); en este sentido, sería más eficiente limitar la tensión de alimentación del diodo que hacer dos llamadas a la función `limexp`.

Las simulaciones de circuitos de gran tamaño no convergen con frecuencia incluso utilizando estrategias de limitación. En estos casos, es necesario utilizar métodos de homotopía [MKF95]. Este tipo de métodos pueden ser implementados si se hace uso de parámetros internos del simulador. Idealmente el simulador de Verilog-A debería reconocer las ecuaciones no-lineales y determinar automáticamente los algoritmos a utilizar por el simulador para mejorar la convergencia. Por ejemplo, el simulador podría automáticamente utilizar la función `limexp` en vez de `exp`. Sin embargo, las implementaciones actuales de Verilog-A no son lo suficientemente sofisticadas y los fabricantes de simuladores mantienen un alto nivel de secretismo sobre los algoritmos de limitación tratándolos como secretos industriales.

Los diseñadores de modelos compactos no deben preocuparse por los detalles de implementación de las funciones de limitación; sin embargo, deben conocer la existencia de las funciones tradicionales:

- «pnjlim» para limitar la tensión entre las uniones PN, incluso las de transistores bipolares.
- «fetlim» para limitar la tensión puerta-fuente de los transistores MOSFET y JFET.

Algunos simuladores convierten las llamadas a la función `limexp` en llamadas a `pnjlim`. Implementaciones de estas funciones están disponibles tanto en SPICE como en los simuladores de la familia. Se propone utilizar la siguiente sintaxis para hacer uso de estas funciones, líneas 1 y 2 del listado 1.5.

```

1 | vdio = $limit(V(a,c), "pnjlim", vcrit);
2 | vgs = $limit(V(g,s), "fetlim", vto);
3 |
4 | vbe=$limit(TYPE*V(base,emmitter)) //No válido
5 | vbe=TYPE*$limit(V(base,emmitter)) //Incorrecto
6 |
7 | if (type==1) //Correcto
8 |     begin
9 |         Vbe = $limit(V(b,e),"pnjlim",vcrit);
10 |        Vbc = $limit(V(b,c),"pnjlim",vcrit);
11 |     end
12 | else
13 |     begin
14 |         Vbe = $limit(V(e,b),"pnjlim",vcrit);
15 |        Vbc = $limit(V(c,b),"pnjlim",vcrit);
16 |     end

```

Listado 1.5 – Funciones limitadoras.

La expresión de la línea 4 presenta error porque la directiva `$limit` no se usa correctamente al no recibir como argumento una tensión de rama, y en la línea 5 por no ser utilizada como una función analógica. El uso correcto se presenta en el bloque condicional `if-end` de la línea 7 del listado 1.5.

Si el simulador no implementa el algoritmo de limitación, o el procedimiento de limitación no es apropiado para el tipo de análisis usado, la primitiva `$limit` devuelve simplemente el valor de su primer argumento. La sintaxis de la línea 1 puede ser usada también para aler-

tar al simulador de una no-linealidad importante. Una descripción en mayor detalle de estas características puede encontrarse [Acc04].

1.8.1.6 Procedimientos de depuración.

Frecuentemente durante el proceso de desarrollo de un modelo compacto en Verilog-A, el desarrollador necesita comprender el comportamiento que presenta el modelo y las posibles diferencias con los resultados esperados. Generalmente, este proceso requiere mostrar valores internos del modelo durante la simulación. Es muy útil disponer del valor de alguna variable en cada iteración, más que el valor una vez alcanzada la convergencia. Para ello se hace uso de la primitiva \$debug que muestra el valor en cada iteración.

```

1 gm = ddx(ids_channel, V(g));
2 $debug("%m: gm = ", gm);
3
4 @(final_step)
5   $strobe("Peak Voltage : %g V\n", vmax);
6
7 `ifdef CHECK_BACK_SURFACE
8   @(final_step)
9   case (back_surf)
10    1: $strobe("Back-surface went into inversion during analysis.\n");
11    2: $strobe("Back-surface went into accumulation during analysis.\n");
12   `endcase
13 `endif

```

Listado 1.6 – Funciones de ayuda a la depuración.

A diferencia de la primitiva \$debug (línea 2 del listado 1.6), que muestra el valor en cada iteración; \$strobe, en la línea 5, acepta los mismos argumentos, muestra los valores únicamente cuando se alcanza la convergencia. Mediante el uso de compilación condicionada se puede generar mensajes de interés para el desarrollador como se muestra en las líneas 7-13 del listado 1.6.

En algunas ocasiones puede ser necesario tener disponibles desde el Front-end del simulador los valores de variables internas del modelo compacto, como suele ocurrir en el caso de las capacidades internas de los dispositivos. Para lograr ésto existen dos alternativas:

- Definir una variable accesible desde el Front-End del simulador. Apartado 1.8, ver líneas 1-2 del listado 1.7.
- Crear un nodo de tensión externa con un valor similar al de la variable de interés, ver línea 4 del listado 1.7

```

1 (*desc="deplexion charge") real qd;
2 (*desc="deplexion capacitance") real cd;
3
4 module diode(a,c,test); //Nodo Auxiliar
5   inout a,c,text;
6   ...
7   I(test)<+ Cgs;

```

Listado 1.7 – Acceso a variables internas del modelo compacto.

1.8.1.7 Eliminación de riesgos en modelos mal formulados.

La codificación de modelos en Verilog-A evita la creación de dos tipos de modelos mal formulados. El primero representa los modelos sin comportamiento transitorio y AC

consistente. Estos problemas aparecen cuando el programador accede a la matriz jacobiana (disponible en [SPICE](#)) usada por el algoritmo de [Newton-Raphson](#) para introducir modificaciones para tener en cuenta efectos no cuasi-estáticos. [Verilog-A](#) no permite al programador acceder a la matriz jacobiana, de este modo, la matriz usada en el análisis transitorio es consistente con la usada para el análisis [AC](#). El segundo tipo es el basado en capacidades y no la carga como variable de estado. La determinación de que los modelos basados en capacidades no conservaban la carga, debido a los efectos del método de integración, fue realizada por Yang [[PEC83](#)]. Es necesario, por tanto, implementar los modelos basados en carga.

1.8.1.8 Optimizaciones automáticas.

La necesidad de crear modelos compactos en [Verilog-A](#) que permitan realizar simulaciones de circuitos muy rápidas, implica que el compilador debe realizar una serie de controles que permitan optimizar el código creado por el diseñador. Entre los controles más importantes se encuentran los siguientes:

- Evitar la repetición innecesaria de evaluaciones intermedias de funciones que se repiten a lo largo del código. Por ejemplo, si en una expresión de corriente se utiliza $id=is*(\exp(vd/\$vt)-1.0)$ de la que se calcula su derivada $gd=is/\$vt*(\exp(vd/\$vt))$ se evitará evaluar varias veces el valor de la función $\exp(vd/\$vt)$, esto ahorrará innumerables ciclos de máquina. El compilador de [Verilog-A](#) intenta buscar subexpresiones eficientes que en otros contextos harían muy poco legible el código escrito por el desarrollador. Un ejemplo de uso de subexpresiones, es el de calcular la inversa de un número y multiplicar por éste en vez de realizar diferentes cocientes. Los compiladores convencionales de [lenguaje-C](#) son incapaces de hacer este tipo de mejoras teniendo en cuenta las expresiones y sus derivadas al mismo tiempo.
- Determinar el árbol de dependencias de modo que el compilador conozca las variables necesarias a evaluar en cada tipo de análisis. Por ejemplo, en el listado 1.8 se presenta un código para la inclusión del ruido *flicker* en la unión de un diodo. La variable *flicker* solo se evaluará en el caso de un análisis de ruido, lo que implica que en el resto de tipos de análisis no se evaluará ni la variable ni la función `flicker_noise(fliker,ef,"1/f")`.

```

1| ...
2| flicker= kf * pow(id,af);
3| I(diode)<+ flicker_noise(fliker ,ef,"1/f");
4| ...

```

Listado 1.8 – *Árbol de dependencias.*

- Eliminación de nodos internos que aumentan el tamaño de la matriz a resolver en el algoritmo de [Newton-Raphson](#). Por cada nodo eliminado se ahorrán dos filas en la matriz que representa el circuito a resolver. La contribución de una tensión requiere una fila adicional para la corriente de malla que se introduce así como otra fila para la nueva tensión nodal. Se recomienda utilizar una estrategia de contribución en corriente a través de una conductancia para evitar la nueva fila de la matriz. Aprovechando, además, el valor de resistencia serie nula para eliminar el nodo interno, ver listado 1.9.

```

1 | ...
2 |   if (rs==0)
3 |     V(res)<+ 0
4 |   else
5 |     I(res)<+ V(res)/res;
6 |   ...

```

Listado 1.9 – Eliminación de nodos internos.

La formulación a través de resistencias o conductancias genera los mismos resultados pero en ocasiones implica diferentes comportamientos en convergencia del algoritmo de [Newton-Raphson](#). En caso de aparición del *problema de pequeñas resistencias*² es necesario evaluar si es más conveniente trabajar con formulación de conductancias o resistencias.

1.8.2 Arquitectura de Verilog-A en los simuladores de circuitos.

La implementación de modelos en [Verilog-A](#) en simuladores de circuitos permite su fácil distribución y uso por parte de la comunidad científica relacionada con el diseño de circuitos. Existen diferentes metodologías de distribución del modelo según el simulador de circuitos: a) distribución del código fuente del modelo para su compilación y uso en el simulador (ver figura 1.11) y b) distribución de módulos compilados cerrados (*CML-Compiled Module Library*) que pueden ser usados en diferentes simuladores (ver figura 1.12).

En la figura 1.11 se describe el diagrama de flujo del proceso de compilación-simulación de un modelo en el simulador de circuitos (*SmartSpice*[®]) de *Silvaco*[™]. La solución propietaria dispone de un editor y depurador que permite la ejecución paso a paso del código durante las simulaciones. Esta herramienta es de gran interés durante las primeras fases del desarrollo del modelo. Posteriormente el código fuente puede ser interpretado o compilado. El proceso se inicia con una fase de compilación donde se genera un fichero C a partir del código en [Verilog-A](#). Posteriormente, se finaliza con una fase de enlazado donde los ficheros C son analizados por el *CINTERPRETER* y transformados por un pseudo-código optimizado que será ejecutado durante la simulación. En el caso de que los ficheros C se compilen por un compilador C nativo o *gcc* se creará una librería enlazable dinámicamente (.so) que será enlazada con el ejecutable del simulador *SmartSpice*. Estas fases son transparentes al usuario. En la

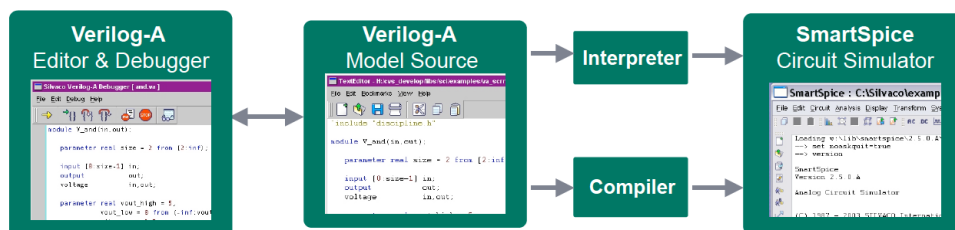


Figura 1.11 – Diagrama de flujo de uso de los modelos en [Verilog-A](#) en el simulador de circuitos (*SmartSpice*[®]) de *Silvaco*[™].

²Este problema aparece cuando en un mismo circuito aparecen resistencias con diferencias de valor de varios órdenes de magnitud, lo que implica que alguno de los valores puede perderse debido a la precisión numérica finita.

figura 1.12 se describe el diagrama de flujo del proceso de compilación del modelo compacto según la plataforma desarrollada por Tiburon Design Automation™. En primer lugar se codifica el modelo en Verilog-A. Posteriormente se procede con la compilación que genera un fichero CML sobre la que se genera un fichero de configuración de arquitectura de hardware y los modelos de componentes necesarios y necesario que será el fichero de configuración compacto en

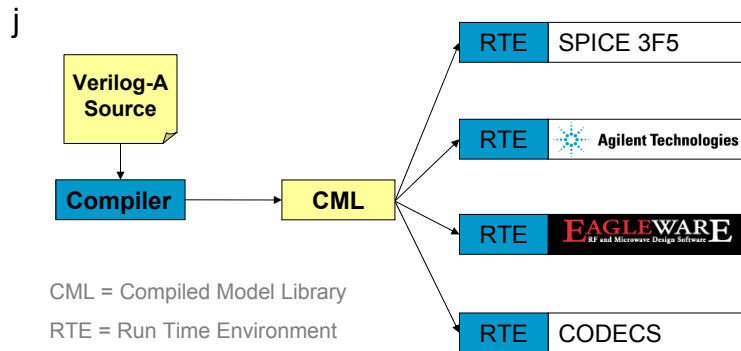


Figura 1.12 – Diagrama de flujo propuesto por Tiburon™ para la obtención de CML en el simuladores de circuitos.

lenguaje-C a partir del código en Verilog-A haciendo uso del conversor ADMS [LGM03]. Esta aplicación libre permite generar los ficheros fuente en lenguaje-C para un conjunto amplio de simuladores comerciales (SPICE3F5, Spectre, HSPICE, etc). Esto permite acelerar la migración del modelo a código nativo del simulador aumentando en ciertos casos la velocidad de ejecución de las simulaciones para los diseños de circuitos integrados que conlleven gran número de dispositivos.

CAPÍTULO

2

ESPICE: IMPLEMENTACIÓN DE SPICE DESARROLLADA EN EL GRUPO DE NANO ELECTRÓNICA DE LA UGR PARA LA ADAPTACIÓN DE MODELOS COMPACTOS.

2.1 Introducción

En este capítulo se describe la particularización y adaptación de la última versión libre de SPICE de la Universidad de Berkeley que hemos realizado con el fin de utilizarla para la modificación y mejora de los modelos compactos incluidos. Nuestra adaptación se denomina ESPICE. Con el objeto de facilitar la tarea principal se ha incorporado una interfaz gráfica, se han depurado errores y se ha sistematizado y simplificado el proceso de incorporación de modelos. Finalmente, se han realizado cambios en un modelo compacto BSIMSOI [HN05], para incorporar nuevos efectos físicos, como por ejemplo el *overshoot* de la velocidad, y se han estudiado circuitos en los que se ha analizado las repercusiones de los cambios incorporados.

Siguiendo la línea descriptiva del capítulo anterior, **SPICE**, es un simulador de circuitos de propósito general que acepta una descripción simple del circuito y genera información de simulación en diferentes modos de análisis: *AC*, *DC*, respuesta transitoria, etc. Diferentes versiones de este programa se han utilizado desde hace treinta años tanto en la industria como en el ámbito docente. **SPICE** es el programa de simulación de circuitos que más éxito ha obtenido hasta nuestros días.

Dado que la tecnología de circuitos integrados y las técnicas de diseño han evolucionado a gran velocidad en el último cuarto de siglo, la simulación de los circuitos mediante herramientas CAD ha incrementado su importancia en el proceso de desarrollo. La simulación de todas las secciones de un gran diseño a nivel de circuitos es hoy día esencial para la detección de fallos previa a la fabricación.

SPICE3 nació de SPICE2 utilizando los mismos algoritmos que han dado tan buenos resultados durante su vida útil, pero con una nueva implementación modular en un nuevo lenguaje de programación, incluyendo más modelos compactos.

2.2 Características internas de la versión inicial de SPICE.

SPICE es un simulador de propósito general y por lo tanto produce resultados aceptables en gran cantidad de problemas. Es posible desarrollar un simulador que utilice otras técnicas que mejoren a **SPICE** disminuyendo el tiempo de simulación sin disminuir la precisión de la solución, pero estas técnicas especiales no se han implementado en SPICE3 ya que sus desarrolladores originales optaron por la generalidad y fiabilidad de la versión anterior.

El simulador en cuestión ha sido diseñado de modo que puede formar parte de una topología de diseño multi-nivel y realice las funciones de simulador de subcircuitos.

2.2.1 Diseño del Simulador: Librerías

Después de la experiencia obtenida del desarrollo en **FORTRAN** de SPICE2, se planteó el desarrollo de una nueva versión orientada a Librerías, con este nuevo concepto, cada dispositivo implementado se diseña y almacena en un componente software denominado Librería que no comparte código alguno con otro dispositivo. De este modo, las rutinas del simulador son independientes unas de otras mejorando el mantenimiento y el desarrollo de la herramienta. Esta filosofía permite, en casos donde el espacio sea determinante, obtener una versión que incluya sólo los dispositivos necesarios. Así, el tamaño final del simulador es menor. Ésto se debe a que la carga de los modelos de los dispositivos no es dinámica sino que se incluyen sobre el fichero ejecutable en tiempo de enlazado.

En esta versión de SPICE, se han añadido también controles que facilitan la configuración del simulador. Mediante el cambio de un número no muy grande de parámetros y/o rutinas es posible la obtención de diferentes simuladores con comportamientos radicalmente diferentes. En este contexto, los modelos de dispositivos pueden ser creados a partir del código fuente de otros dispositivos anteriores.

A modo de resumen, los módulos o librerías que configuran SPICE3 son los siguientes:

- Analizador sintáctico de los comandos de entrada.
- Analizador sintáctico de los circuitos de entrada.
- Planificador de tareas de interacción/batch del usuario.
- Tratamiento de las matrices dispersas que almacenan los datos del circuito.
- Rutinas de coordinación y envío al Simulador y rutinas numéricas genéricas.
- Tipos de análisis disponibles:
 - OP - Punto de operación
 - AC - Corriente alterna
 - DC - Corriente continua.
 - TRAN - Análisis temporal
 - TF -Función de transferencia
 - PZ - Polos-Ceros
 - SENS - Sensibilidad
- Elementos Circuitales:
 - V-I Fuentes independientes de Tensión y Corriente
 - Fuentes dependientes de Tensión y Corriente
 - * E - Voltage controlled dependent voltage source
 - * G - Voltage controlled dependent current source
 - * H - Current controlled dependent voltage source
 - * F - Current controlled dependent current source
 - R - Resistencias
 - C - Condensadores
 - L - Inductancias e inductancias mutuas
 - Líneas de Transmisión
 - Líneas distribuidas R-C
 - Conmutadores conmutables por tensión y/o corriente.
- Dispositivos semiconductores:
 - Diodos
 - Transistores Bipolares
 - JFET
 - MESFET
 - MOSFET nivel-1
 - MOSFET nivel-2
 - MOSFET nivel-3
 - BSIM-MOSFET
 -
- Rutinas independientes del sistema Hardware

- Tratamiento Gráfico
- Interfaz de rutinas gráficas:
 - Sistema X windows versión 10
 - Sistema X windows versión 11
 - Paquete de Model Frame Buffer (MFB)

2.2.2 Tipos de análisis posibles mediante SPICE

Los diferentes tipos de análisis posibles sobre un circuito electrónico mediante el simulador **SPICE** se enumeran a continuación.

2.2.2.1 Análisis en corriente continua (DC)

El análisis en corriente continua permite obtener el punto de operación de un circuito electrónico y habilita el análisis de sensibilidad. En el análisis de punto de operación se determinan las tensiones y las corrientes en reposo en todos los nodos, para ello se deben cortocircuitar todos los inductores y suponer abiertos todos los condensadores del circuito. Este análisis se realiza de manera previa a los análisis temporales y de corriente alterna.

Existen diferentes modos de funcionamiento del análisis en continua dependiendo de los objetivos que se quieran obtener. Es necesario indicar al simulador las necesidades del análisis mediante un parámetro que se pasa al simulador en el fichero de datos de entrada.

Modos Funcionamiento:	Características
.DC	Curvas de transferencia.
.TF	Estimación de la función de transferencia en pequeña señal.
.OP	Cálculo del punto de operación.
.SENS	Estimación de la sensibilidad del circuito frente a variaciones de los componentes.

Tabla 2.1 – *Tipos de análisis en corriente continua.*

2.2.2.2 Análisis en corriente alterna (AC)

El análisis en corriente alterna no es un análisis directo e independiente y siempre supone que se utiliza una excitación sinusoidal a la entrada del circuito. SPICE realiza primero un análisis DC para calcular el punto de operación del circuito y determina un modelo linealizado y de pequeña señal de todos los dispositivos no-lineales del circuito bajo pruebas.

La generación de ruido blanco por parte de las resistencias y de los dispositivos semiconductores es simulado en este tipo de análisis. Se pueden realizar caracterizaciones en distorsión ya que se pueden utilizar dos señales de frecuencia diferente como excitación del circuito.

Existen diferentes modos de funcionamiento del análisis en alterna:

Modos Funcionamiento:	Características
.AC	Análisis en corriente alterna (AC) de pequeña señal.
.NOISE	Estimación del Ruido en el circuito.
.DISTO	Estimación de la Distorsión en el circuito.

Tabla 2.2 – *Tipos de análisis en corriente alterna.*

2.2.2.3 Análisis transitorio (TRAN)

El análisis transitorio permite estudiar la evolución en el tiempo de las variables del circuito independientemente de la excitación que se utilice. Como en el caso anterior, SPICE realiza primero un análisis DC para calcular el punto de operación del circuito y determina el modelo linealizado y de pequeña señal de todos los dispositivos no-lineales del circuito bajo pruebas. Finalmente, calcula todas las variables de salida del circuito en función del tiempo dentro del intervalo de simulación definido.

Existen diferentes modos de funcionamiento del análisis temporal:

Modos Funcionamiento:	Características
.TRAN	Análisis transitorio
.FOUR	Análisis de Fourier

Tabla 2.3 – *Tipos de análisis transitorios.*

2.2.3 Arquitectura del Simulador SPICE3

La arquitectura del programa define las características de las funciones implementadas por cada uno de los módulos que forman el simulador, así como los interfaces de comunicación entre ellos. En la figura 2.1 se detalla la arquitectura general del simulador. El módulo interfaz de usuario realiza todas las funciones de entrada/salida necesarias durante el análisis del circuito. Éste se comunica con el núcleo del simulador que implementa la rutinas matemáticas. El módulo de dispositivos permite aislar el código necesario para realizar las funciones de creación, manipulación de datos y cálculos internos de los diferentes modelos compactos accesibles por el núcleo del simulador. Como módulo intermedio, entre los dispositivos y las rutinas matemáticas, existe un bloque de librería de matrices que incluye todas las herramientas para el manejo de las matrices que representan al circuito bajo análisis. El tráfico de información entre los bloques ha de respetar obligatoriamente el flujo establecido en las flechas. De este modo, el usuario únicamente podrá acceder a la información de los dispositivos a través de consultas directas al núcleo del simulador ya que el interfaz de usuario sólo accede a través de los puntos de entrada generados por las rutinas de control del simulador. Los demás bloques no conectados directamente con el usuario no están accesibles para éste. La tarea de añadir nuevos modelos compactos para dispositivos no existentes, nuevos tipos de análisis y funciones de entrada y salida de datos obliga a conocer con detalle las

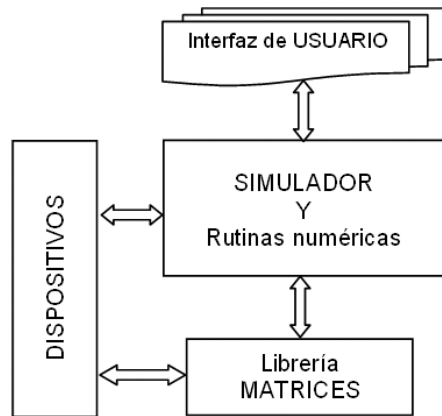


Figura 2.1 – Estructura general de SPICE

diferentes estructuras de datos. Éstas describen las características de la simulación a varios niveles de profundidad. También hace falta controlar las primitivas establecidas para realizar el traspaso de información entre las estructuras de datos de los diferentes bloques descritos anteriormente.

Para la modificación o creación de nuevos modelos compactos en código fuente es necesario conocer las partes del simulador descritas en el apartado 2.2.3.1. Como aplicación práctica del manejo de modelos basados en código fuente, se ha realizado una modificación del modelo compacto BSIMSOI (apartado 2.4.3) con el fin de poder estudiar el comportamiento en diferentes circuitos del modelo de código fuente original y caracterizar la influencia de nuevos efectos físicos incorporados en ellos [RGLVC97].

2.2.3.1 Diagrama de bloques del simulador

A continuación se presenta la descripción del diagrama de bloques del simulador: es importante resaltar que no hay conexión entre las estructuras que no están unidas por flechas, esto indica que están desacopladas. En esta descripción del diagrama de bloques, cada uno

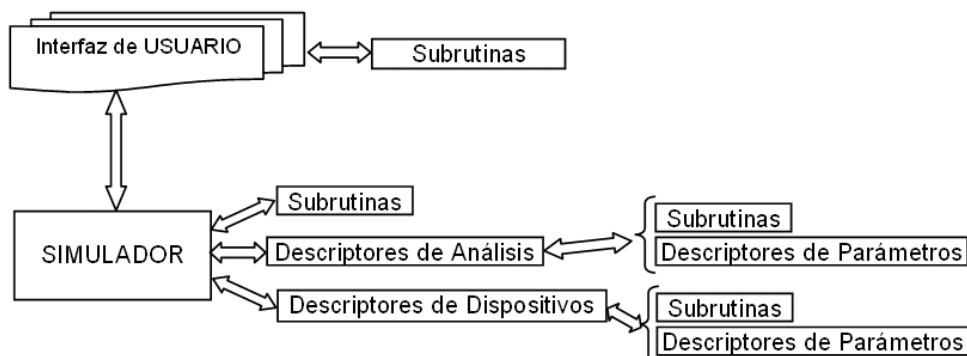


Figura 2.2 – Diagrama de bloques del simulador.

de los módulos menores exporta de manera estándar otra estructura de datos que contiene una descripción de las posibilidades del módulo; de esta manera todos los modelos de los dispositivos y los tipos de análisis pueden agruparse en un único vector o *array* en un nivel

superior de implementación.

En el nivel de descripción inferior, las estructuras de datos consisten en vectores de descriptores de parámetros y un conjunto de punteros a las funciones que implementan las capacidades de dicho módulo. En el siguiente nivel, la información de todos los paquetes que serán usados por la interfaz de usuario a través del simulador son agrupados y exportados en una sola estructura. Del mismo modo, el interfaz de usuario introduce en una estructura de datos todas las funciones que serán necesitadas por el simulador y la exporta al simulador durante la inicialización. Durante la ejecución, el simulador y el interfaz de usuario utilizan las estructuras de datos o descriptores exportadas por cada uno para determinar las capacidades disponibles, los parámetros utilizados en cada llamada, y los tipos de argumentos definidos por los parámetros.

2.2.4 Estructura de datos encapsulados en SPICE3

En la figura 2.3 se representa la estructura de datos encapsulados que contiene en SPICE3 toda la información referente a un circuito.

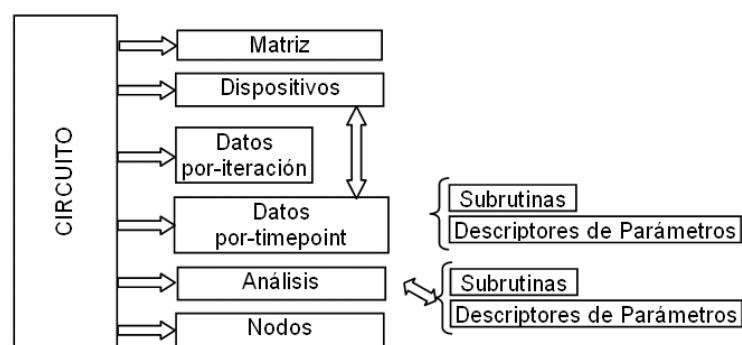


Figura 2.3 – Estructura de datos encapsulados de un circuito

Aunque esta estructura está pensada para representar un circuito completo, también puede hacer referencia a un subcircuito de un simulador de características avanzadas que utilice a SPICE3 como analizador de subcircuitos.

La estructura de datos encapsulados contiene gran cantidad de datos globales, así como punteros a otros módulos individuales y más especializados. Las otras estructuras almacenan los datos que son: o bien privados para un módulo individual, o pueden ser referenciados mediante un puntero. La estructura matriz (apartado 7.3.2.3) es completamente privada para la librería de matrices dispersas, aunque otros módulos pueden utilizar referencias mediante punteros a partes específicas de la matriz dispersa.

La estructura de datos asociada a los dispositivos es bastante compleja ya que puede albergar los numerosos tipos de modelos compactos usados en los circuitos electrónicos actuales. Los datos que son renombrados en cada iteración, el vector de excitaciones y el de soluciones son almacenados en un par de vectores asignados dinámicamente que son utilizados frecuentemente por el núcleo del simulador.

Los datos que describen cada análisis se almacenan en una lista enlazada de estructuras

definidas según el tipo de análisis, y son de acceso privado para el código que implementa cada tipo de análisis. La descripción de los nodos del circuito se almacena en una tabla de nodos y es utilizada por un número pequeño de rutinas incluidas en el módulo del circuito. Las descripciones de las estructuras de datos se han incorporado en el Anexo 7.3. En adelante se utilizarán referencias a las estructuras de datos en él descritas.

2.2.5 Compilación del Simulador SPICE3F4 para Windows

La versión original de SPICE3 se ha compilado en Windows de 32 bits. Para ello se ha estudiado la distribución de SPICE3F4 disponible para libre uso en la Universidad de Berkeley y un desarrollo primitivo para Windows 32 bits realizado por *Wolfgang Mues*. Se ha migrado el entorno de compilación a otro nuevo con interfaz gráfica de *Borland*. Se han introducido las mejoras pertinentes para aumentar la eficiencia del código añadiendo la información de depuración de código para la simulación paso a paso de los circuitos y el aumento de información de depuración en lo relativo a modelos compactos.

La estructura de la Aplicación de Windows 32 bits sobre Borland C++[®] 5.01 se presenta en la figura 2.4(a) donde se aprecia la estructura a nivel de librerías y módulos. La estructura de directorios del proyecto se muestra en la figura 2.4(b).

En la misma figura se pueden observar los diferentes niveles de programación en los que se ha subdividido la implementación completa del simulador. Las diferentes librerías que enlazadas estáticamente forman la versión inicial de SPICE3F4 son:

- DISPOSITIVOS:
 - ASRC:Fuente Arbitraria
 - BJT: Transistor bipolar
 - BSIM1: MOSFET
 - BSIM2: MOSFET
 - CAP: Condensador
 - CCCS: Fuente de corriente controlada por corriente
 - CCVS: Fuente de tensión controlada por corriente
 - CSW: Interruptor controlado por corriente
 - DIO: Diodo
 - IND: Inductor
 - ISRC: Fuente de corriente
 - JFET: JFET
 - LTRA: Líneas de Transmisión
 - MES: Transistor MESFET
 - MOS1: MOSFET - Nivel 1
 - MOS2: MOSFET - Nivel 2
 - MOS3: MOSFET - Nivel 3
 - MOS6: MOSFET - Nivel 6
 - RES: Resistencia
 - SW: Interruptor controlado por Tensión
 - TRA: Líneas de Transmisión
 - URC: Líneas RC uniformes
 - VCCS: Fuente de Corriente controlada por Tensión

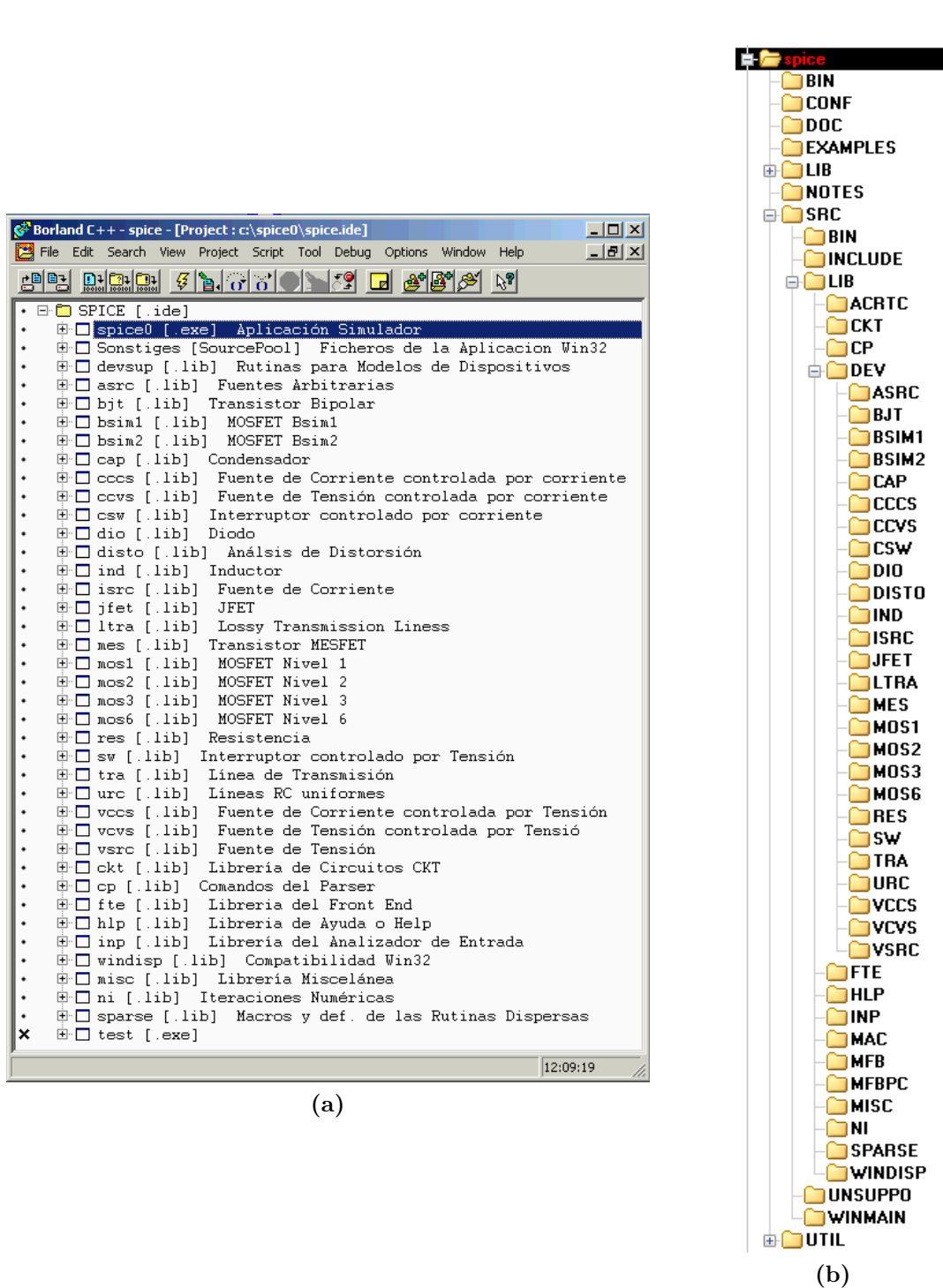


Figura 2.4 – Estructura del Simulador en BorlandC++5,01 (a) Estructura de directorios del proyecto (b).

- VCVS: Fuente de Tensión controlada por Tensión
- VSRC: Fuente de Tensión
- VARIOS:
 - DISTO: Análisis de Distorsión.
 - CKT: Librería de Circuitos CKT.

- CP: Comandos del Parser o Analizador.
- FTE: Interfaz de usuario o *Front End*.
- HLP: Ayuda.
- INP: Input Parser o analizador de entrada.
- WINDISP: Compatibilidad con la plataforma [Windows®](#) 32 bit.
- MISC: Rutinas misceláneas (time, dosdir, string, etc.).
- NI: Rutinas matemáticas de iteraciones numéricas.
- SPARSE: Rutinas de Matrices dispersas.
- DEVSUP: Rutinas para tratamiento de modelos de dispositivos.
- SONSTIGES: Ficheros auxiliares para la Aplicación [Windows®](#) 32 bit.
- SPICE: Núcleo de la Aplicación. Ficheros de arranque inicial en [Windows®](#). Configuración para el enlazador de librerías del Compilador.

2.2.6 Aspecto final de SPICE3F5

Una vez se han introducido todos los cambios pertinentes sobre el código fuente original, se obtiene la versión última existente del simulador de circuitos. La versión SPICE3F5 incluye todos los parches documentados en los portales de desarrollo de simuladores de circuitos y presenta un interfaz externo formado por una ventana de consola que funciona correctamente sobre la plataformas [Windows®](#) 32 bits. No es posible su migración directa a otra plataforma [Hardware](#) ni sistema operativo. Esta última limitación ha motivado la creación del simulador [ESPICE](#) dentro del Grupo de Nanoelectrónica de la Universidad de Granada, lo que permite la modificación y creación de nuevos modelos compactos en código fuente. La ventana

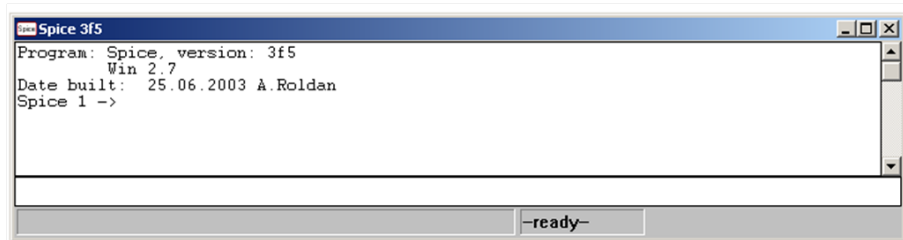


Figura 2.5 – *Aspecto Simulador Spice3F5 en versión original.*

principal de ejecución, figura 2.5, está formada por:

- Marco de Estado: Donde aparece el estado interno del simulador (preparado, realizando análisis, etc).
- Ventana de Salida de Datos del Simulador: Se muestran tanto los datos de salida como el resultado de ejecución de los comandos introducidos por teclado.
- Ventana de entrada de comandos: Interfaz de entrada mediante el que el usuario introduce las primitivas de operación al simulador.
- Marco de nombre del circuito en simulación: Muestra el nombre del fichero que contiene el circuito situado actualmente en la memoria de ejecución

La versión mejorada del interfaz gráfico del simulador se presenta en la figura 2.6 donde se puede apreciar la inclusión de una barra de iconos y menús para hacer más intuitivo el interfaz con el usuario.

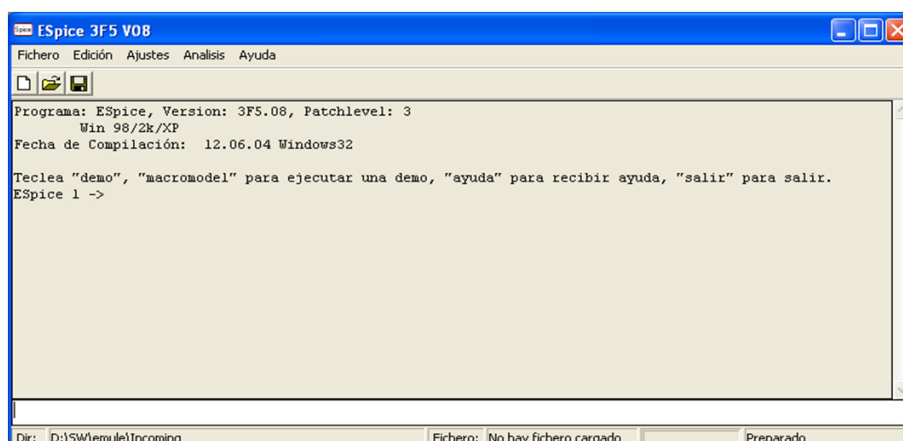


Figura 2.6 – Aspecto Simulador Spice3F5 en versión final.

Debido a las limitaciones que presenta la compilación del proyecto bajo *Borland C++ 5.01* por la imposibilidad de migración a otra plataforma y sistema operativo, la dificultad de aumentar el número de modelos compactos a incorporar en el ejecutable, la inexistencia de entornos gráficos no basados en las API de *Windows®* surgió la necesidad de crear una herramienta de simulación basada en una versión más evolucionada que la última versión de código libre de SPICE de la Universidad de Berkeley. En el apartado siguiente se detalla el proceso de desarrollo de la nueva herramienta que supera las limitaciones enumeradas anteriormente.

2.3 ESPICE. Descripción del nuevo simulador

Dado que el desarrollo de nuevos modelos compactos necesita de un simulador de circuitos donde poder realizar las pruebas de validación del modelo, en el contexto de esta tesis y encuadrada en la línea de modelado de dispositivos se planteó la necesidad al Grupo de Nanoelectrónica de la Universidad de Granada del desarrollo de un simulador de circuitos orientado a la investigación y estudio de modelos avanzados de dispositivos de última generación. El simulador permitiría incluir en los modelos ya existentes los efectos físicos caracterizados en las diferentes líneas del investigación del grupo y estudiar las implicaciones en diferentes circuitos electrónicos.

2.3.1 Introducción.

Los objetivos que se propusieron al inicio del proceso de desarrollo del simulador fueron los siguientes:

- Introducir nuevos modelos de dispositivos creados por los diferentes grupos de investigación (BSIM, MEXTRAM, BSIMSOI, EKV, HiSIM, HICUM, etc.)
- Modificar modelos de dispositivos existentes para introducir nuevos comportamientos.
- Mejorar los diferentes tipos de análisis disponibles e introducir nuevos.

- Introducir análisis de optimización bajo criterios.
- Acceso a información interna sobre:
 - Ecuaciones circuitales.
 - Proceso interno de linealización de dispositivos durante la fase de análisis.
 - Matrices dispersas.
- Mejora del interfaz gráfico añadiendo un interfaz de ventanas, intuitiva y reconfigurable.
- Obtención de un sistema multiplataforma ([Windows](#) y [GNU/Linux](#) al menos).
- Aplicación multilenguaje con opción de incorporación de nuevos idiomas.
- Plataforma libre de actualización y mejora.
- Difusión a las mejoras realizadas: <http://espice.ugr.es>

El proyecto del nuevo simulador tenía como punto de partida las experiencias realizadas con el simulador SPICE3F5, presentado en el apartado 2.2.5. Para incorporar las mejoras existentes en otros simuladores abiertos disponibles en Internet, se estudiaron los simuladores [SPICE+](#) y [NG-SPICE](#) y se compararon con la versión disponible para [Windows](#) obtenida en el Grupo de Nanoelectrónica.

Simulador	Ventajas	Inconvenientes
Versión Original de Berkeley (Spice3f5)	Amplia documentación. Fuente de partida del resto de alternativas a considerar.	Inestable para simulaciones complejas. Proyecto inactivo.
Spice+	Interfaz gráfica implementada	Basado en una versión antigua del Spice de Berkeley. Estructura de directorios confusa.
Spice3F5 compilado para Windows	Interfaz gráfica implementada.	Muy difícil de portar a GNU/Linux. Uso de librerías propietarias de Borland.
NG-Spice Next Generation Spice	Incluye: Spice, Cider y Xspice. Mejora ampliamente la estabilidad. Preparado para <i>autotools</i> GNU/Linux. Mejoras en el núcleo de Spice3f5. Posee soporte y desarrollo. Multiplataforma.	Errores en algunas simulaciones.

Tabla 2.4 – Comparativa de características de los diferentes simuladores estudiados.

En la tabla 2.4 se esquematizan las ventajas e inconvenientes que cada uno de los simuladores con código fuente accesible presentaban para ser elegidos como punto de partida. El proyecto de [NG-SPICE](#), que no disponía de interfaz gráfico basado en ventanas, fue elegido por incluir los simuladores ([SPICE](#), [CIDER](#) y [XSPICE](#)), y formar parte de una *suite* de diseño electrónico de distribución gratuita.

- CIDER Versión 1b1: está enlazado con ESPICE para realizar simulaciones de dispositivos electrónicos mediante el simulador DSIM. Esta herramienta permite un mayor grado de precisión en la simulación de dispositivos críticos y nuevos prototipos. Los dispositivos DSIM se describen en términos de su estructura geométrica y materiales que lo forman. Origen: Univ. de Berkeley.
- XSPICE es una extensión de SPICE que añade el soporte de modelado compacto y simulación de componentes digitales al incluir un algoritmo controlado por eventos digitales. Origen: Georgia Institute of Technology.

2.3.2 Interfaz de ventanas.

El interfaz gráfico de un simulador es esencial para determinar su utilidad. Por esta razón, hemos utilizado para **ESPICE** una gestión gráfica similar a la utilizada en programas como **MATLAB**. El gestor de ventanas elegido debía ser multiplataforma y permitir, siguiendo un procedimiento sencillo, el uso y compilación en diferentes plataformas. La versión del gestor de ventanas elegida entre las disponibles (ver figura 2.7) fue **GTK+**, que fue desarrollada inicialmente para crear el programa **GIMP**. Permite el uso de múltiples lenguajes de programación, es multiplataforma y se distribuye bajo licencia **Distribución de Software Berkeley**. Está basada en tres librerías: **Glib**: Hebras, llamadas al sistema, **Pango**: Diseño y renderizado de texto e internacionalización y **ATK**: Interfaces con características de una gran accesibilidad. La arquitectura del gestor gráfico de **ESPICE** permite que las funciones



Figura 2.7 – Gestores de ventanas existentes: *GTK+* (a), *QT* (b), *WXwidget* (c) y *FoxToolkit* (d).

se puedan representar con varias librerías gráficas, por situarse éstas en la capa superior. En la figura 2.8 se presentan las librerías disponibles: **GTK+** que es el entorno básico introducido en **ESPICE** para obtener el entorno gráfico multiplataforma, **X11** que es el sistema de ventanas original de los entornos UNIX y se conserva por compatibilidad original; y **OpenGL** que gestiona de manera transparente la generación de gráficos 2D/3D por *hardware* y posee aceleración por *hardware*, primitivas de dibujo, doble *buffer* y redibujado. El gestor

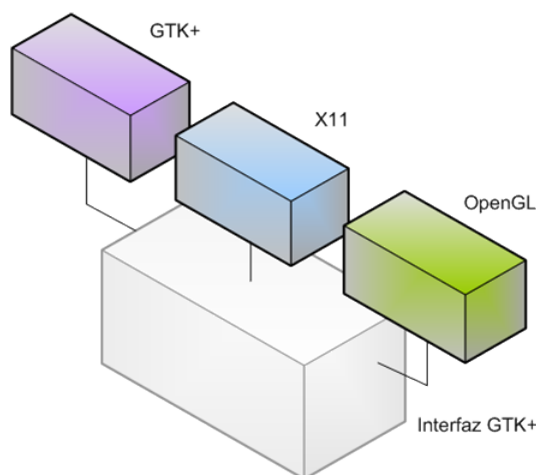


Figura 2.8 – Estructura de bloques del gestor gráfico de *Espice*.

de librerías gráficas (**X11** en este caso) utiliza una estructura de datos similar a la descrita en el listado 2.1. Las primitivas que la forman establecen funciones básicas necesarias para representar las gráficas disponibles en el simulador.

```

1| DISPDEVICE device [] = {
2|     . . . . .
3|     {"X11", 0, 0, 1024, 864, 0, 0, X11_Init,
4|      X11_NewViewport,
5|      X11_Close, X11_Clear,
6|      X11_DrawLine, X11_Arc, X11_Text, X11_DefineColor,
7|      X11_DefineLineStyle,

```

```

8 | X11_SetLineStyle, X11_SetColor, X11_Update,
9 | nodev, nodev, nodev, X11_Input,
10 | gen_DatatoScreen,}
11 | , ... }

```

Listado 2.1 – Definición de librerías gráficas para representación de funciones.

En la figura 2.9 se detallan las librerías gráficas disponibles en la última versión del simulador. Las librerías gráficas se implementan de manera separada en una estructura de directorios independiente y se interconectan con el núcleo del simulador gracias a las funciones existentes en el fichero de gestión gráfica `Display.c`. Una vez elegido el gestor de ventanas, es neces-

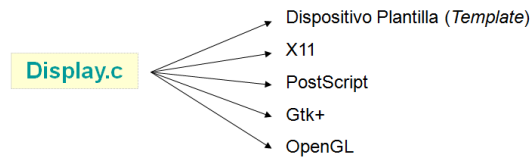


Figura 2.9 – Librerías gráficas disponibles en Espice.

rio determinar un procedimiento de comunicación entre el núcleo del simulador y el gestor mediante el intercambio de mensajes. Cada una de las partes (núcleo y gestor) están activas a la vez, son dos procesos o hebras que deben coordinarse. En la figura 2.10 se presenta la configuración multihebra que se ha establecido para la comunicación entre el núcleo original y el entorno gráfico. El programa principal arranca las hebras del simulador de circuitos, la

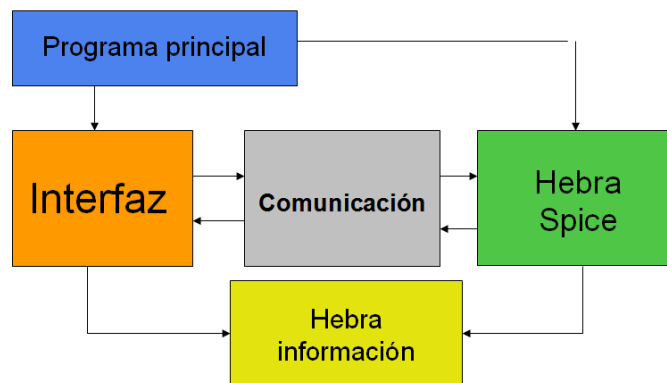


Figura 2.10 – Arquitectura de comunicación multihebra.

del entorno gráfico y la de información que permite refrescar el estado del núcleo y cambiar la configuración del núcleo durante la ejecución. La comunicación se ha implementado reescribiendo las funciones de la librería `stdio.h` (`getc`, `printf`, `scanf`, `putc`, `seek`, ...). Durante el proceso de comunicación se utilizan funciones antibloqueo para evitar que pueda existir un problema con las hebras. De este modo, durante el inicio de una comunicación hacia el interfaz gráfico se bloquea a éste haciendo una llamada a la primitiva `gdk_threads_enter()` para que no haga otra cosa, y se libera posteriormente llamando a `gdk_threads_leave()` (ver listado 2.2). Los elementos del interfaz gráfico que pueden ser actualizados por el núcleo del simulador son: barra de progreso de simulación, LEDs indicadores de estado, interfaz de navegación de directorios, gráficas de simulación, listado de circuitos activos, control de actualización de idiomas, etc.

```

1 | void AppendChar( char c, bool error)
2 | {
3 |     GtkTextIter end;
4 |     gdk_threads_enter();

```

```

5 // Obtiene el final del buffer buffer
6 gtk_text_buffer_get_end_iter (TBuffer, &end);
7 if ((c != TB) && &end) {
8     if (error)
9         gtk_text_buffer_insert_with_tags_by_name( TBuffer, &end, &c, 1, "red_foreground",
10            NULL);
11     else
12         gtk_text_buffer_insert( TBuffer, &end, &c, 1);
13 }
14 gdk_threads_leave();
15 }

```

Listado 2.2 – Implementación de técnica antibloqueo en la comunicación SPICE \mapsto Interfaz gráfico.

Para poder ejecutar el simulador de circuitos en el sistema operativo elegido ([GNU/Linux](#), [Windows](#)) es necesario realizar previamente la instalación de algunos paquetes software relacionados con el Gestor de Ventanas multiplataforma [GTK+](#). Del igual modo, y en el caso de que el usuario prefiera realizar la compilación del código fuente para introducir alguna modificación en los modelos compactos, o conocer el procedimiento de obtención del ejecutable, será necesaria la instalación adicional de algunos paquetes más orientados al desarrollo de aplicaciones basadas en [GTK+](#).

Sin entrar en los detalles de las librerías para ejecución y compilación, se presenta a continuación una única lista con los paquetes necesarios:

- WINDOWS XP-W2003-VISTA:
 - GTK-win32: gtk-2.10.11-win32-1.exe [o superior]
 - GTK-sourceview: libgtksourceview-1.6.0-devel-win32.zip
 - MinGW: <http://www.mingw.org/>
 - Msys: <http://www.mingw.org/msys.shtml>
- Distribuciones LINUX:
 - DEBIAN: automake, libtool, autoconf, libgtk2.0-dev, libgtkglext1-dev, libxaw7-dev, libgtksourceview1.0-0-dev
 - UBUNTU: automake, libtool, libgtk2.0-dev, libxaw7-dev, libgtksourceview-dev texinfo tex-common
 - SUSE: dbus-1-glib-devel, esound-devel, gconf2-devel, gnome-vfs2-devel, gtksourceview-devel, hal-devel, libbonobo-devel, libgnome-devel, libgnomeprint-devel, libidl-devel, orbit2, orbit2-devel, popt-devel

El proceso de compilación necesario para poder hacer uso del simulador en las plataformas más usuales a partir del código fuente es el siguiente:

- [GNU/Linux](#)[®]:

```

1 user> svn co http://espice.ugr.es/repos/espice
2 user> cd espice
3 user> ./compila
4 user> cd src
5 user> ./espice

```

- [Windows](#):

```

6 user> svn co http://espice.ugr.es/repos/espice
7 user> cd espice
8 user> ./compila_win
9 user> cd src
10 user> ./espice

```

Las peculiaridades asociadas a los diferentes sistemas operativos disponibles se articulan a través de directivas de compilación condicional (ver listado 2.3), que permiten seleccionar en tiempo de compilación un trozo de código u otro, dependiendo del sistema operativo para el que se está generando el simulador. En la línea 4 aparece la directiva de configuración del código para el sistema operativo **Windows**.

```

1 #ifdef WINDOWS
2 ...
3 #endif
4 ..
5 ./configure --with-gtk --with-windows

```

Listado 2.3 – Directivas de compilación condicional.

En la figura 2.11, se muestra la estructura de directorios creados por el gestor de versiones *subversion* [Sub11] sobre una ventana de consola *Msys* [MSY11] ejecutada sobre **Windows**. En la parte inferior de ésta, se ha realizado una actualización del código fuente del repositorio del que se ha descargado el fichero *espice_sp.wln* y el *script* de compilación automática para **Windows** *compila_win* donde se informa sobre la versión descargada del repositorio, que en este caso es la número 766. **ESPICE** se distribuye bajo **BSD** porque el simulador **SPICE**

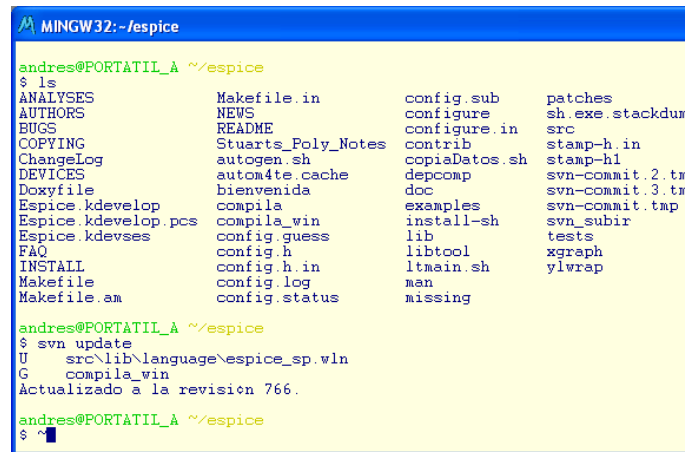


Figura 2.11 – Estructura de directorios en MSYS y resultado compilación.

original de la Univ. de Berkeley, en el cual está basado **NG-SPICE**, se distribuye bajo licencia **BSD** antigua, la cual es incompatible con la licencia **GPL** más genérica, por la característica vérica de esta última.

A continuación se presenta una colección de capturas de cada uno de las ventanas gráficas que conforman el simulador. En la figura 2.12 se muestra la barra de menús e iconos principales del simulador. El entorno gráfico es similar tanto en **Windows**® como en **GNU/Linux**®. En la figura 2.13(a) muestra el navegador de ficheros a través del que podemos



Figura 2.12 – Barra de menús e iconos principales.

desplazarnos por los directorios del sistema de ficheros para buscar y cargar el *netlist* con la descripción del circuito a simular. El histórico de comandos ejecutados durante la simulación (figura 2.13(b)) junto con el listado de representaciones gráficas o *plots* (figura 2.13(c)) y el listado de circuitos cargados y disponibles en memoria (figura 2.13(d)) forman el conjunto de ventanas de información al usuario. La barra de estado de la figura 2.14 muestra la informa-

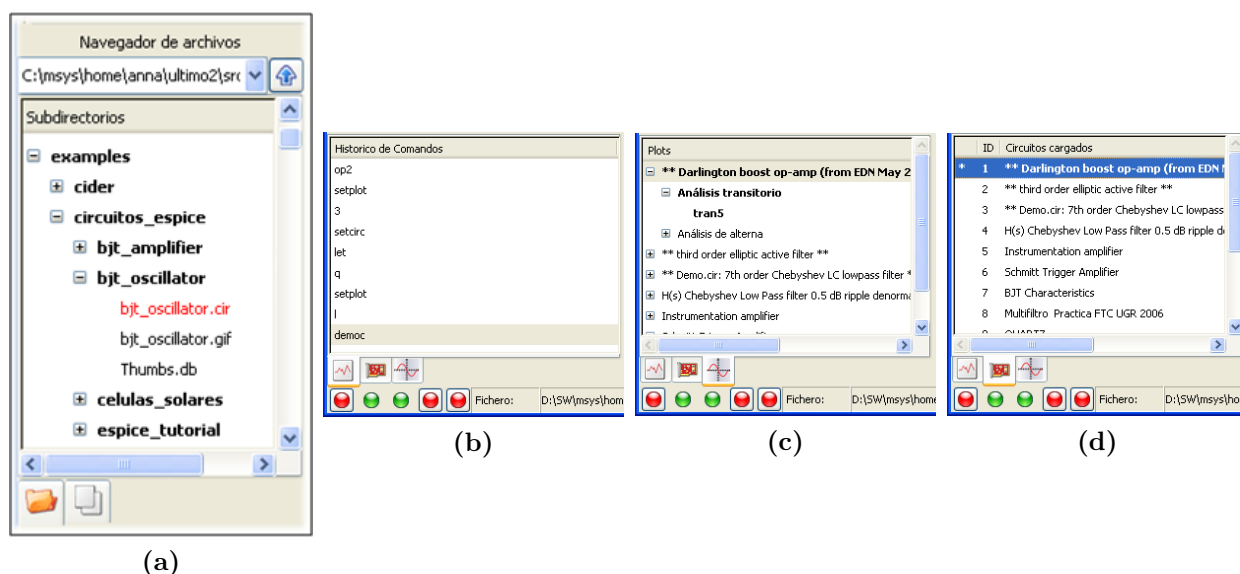


Figura 2.13 – Navegador de archivos de simulación (a), histórico de comandos (b), listado de gráficas (plots) disponibles (c) y listado de circuitos cargados en memoria (d).

ción disponible relativa al estado y progreso de la simulación, el nombre del fichero cargado, y el estado de circuitos y *plots* en memoria. La consola de entrada de datos e interfaz de

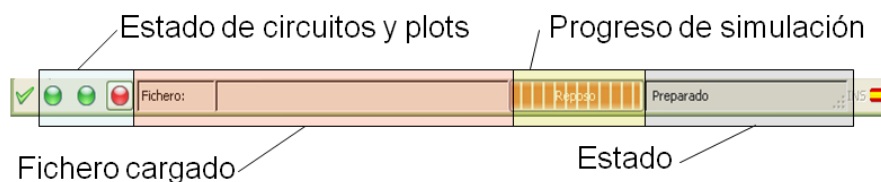


Figura 2.14 – Barra de menús e iconos principales.

usuario permite el uso de colores que facilitan la identificación de mensajes diversos relativos al estado de la simulación, resultados internos de los algoritmos de cálculo y estrategias utilizadas para mejorar la convergencia de las rutinas numéricas, entre otros. La ventana principal de la aplicación se representa en la figura 2.15. Además de estas facilidades de uso, el simulador ha sido dotado de un interfaz de selección de idiomas, cambio de tipos de letra y colores de mensajes establecidos, figura 2.16 que permiten la particularización de las características gráficas a demanda del usuario. Finalmente, cabe destacar las ventanas que permiten gestionar las opciones globales del simulador que ocasionalmente son necesarias variar para modificar el comportamiento del simulador (figura 2.17) y acceder al listado de componentes disponibles en la librería del simulador (figura 2.18).

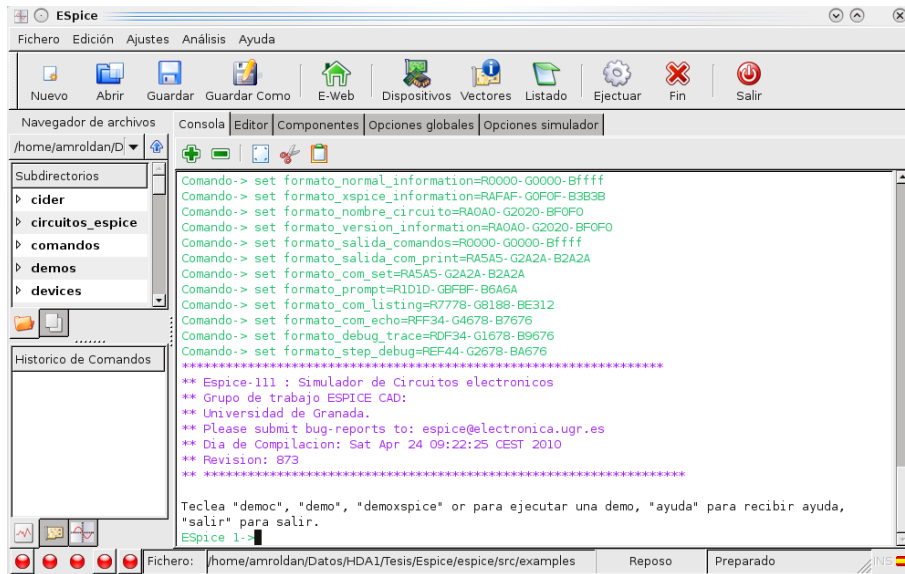
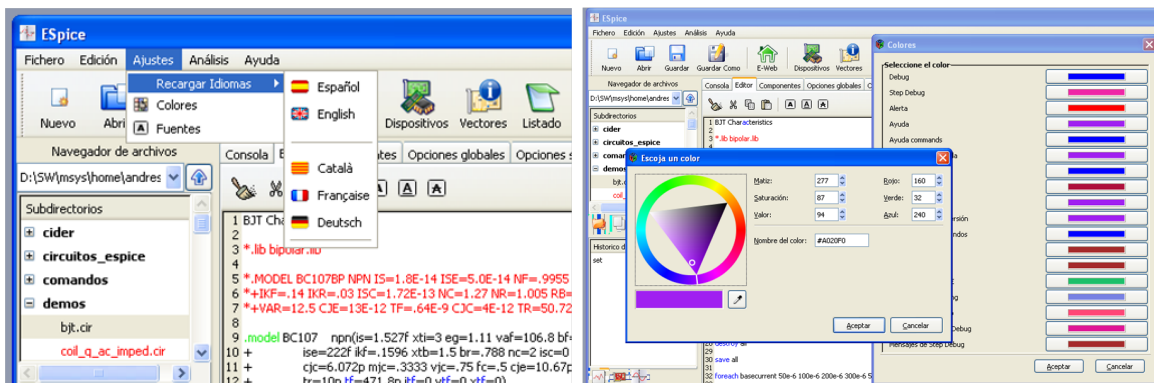
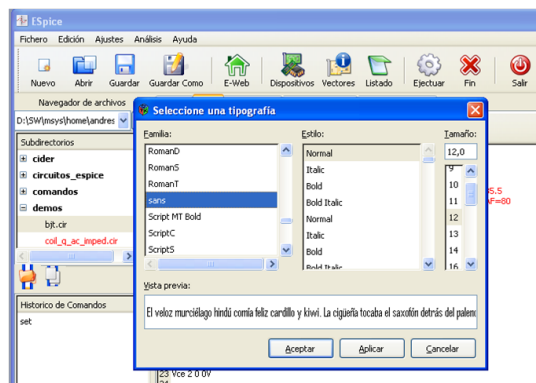


Figura 2.15 – Ventana principal.



(a)

(b)



(c)

Figura 2.16 – Interfaz de idiomas (a), colores (b) y fuentes (c).

2.3.3 Otras mejoras.

Además de las mejoras descritas en el apartado 2.3.2 hemos incorporado las siguientes:

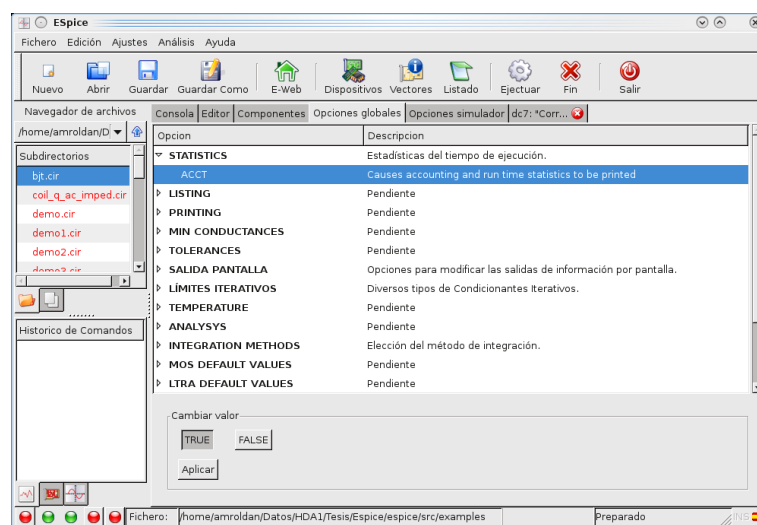


Figura 2.17 – Ventana gestor de opciones globales.

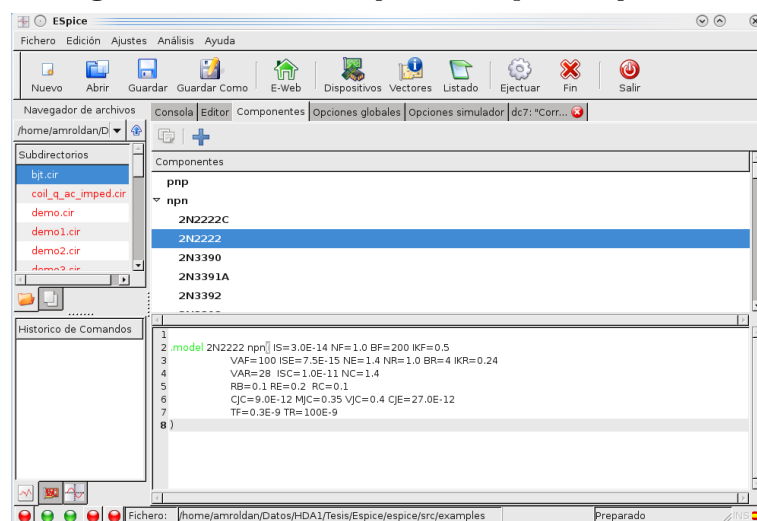


Figura 2.18 – Ventana gestor de componentes discretos o subcircuitos.

- Identificación y salida controlada de los mensajes de depuración para el caso de incorporación de nuevos modelos compactos en código fuente.
- Control de mensajes en algoritmos de convergencia paso a paso, ver figura 2.19.
- Identificación de los mensajes de salida provenientes del simulador XSPICE.
- Incorporación de modelos compactos existentes en la bibliografía aumentando el conjunto de elementos disponibles.
- Incorporación de un editor de textos para poder modificar la descripción topológica del circuito a simular, ver figura 2.20.
- Corrección de errores en la gestión gráfica de las representaciones polares y cartas de Smith, ver figura 2.21.
- Corrección del código que interpreta la lectura de los ficheros *netlist* que incluyen primitivas algorítmicas.
- Corrección del código para evitar problemas de gestión de memoria y punteros vacíos.
- Incorporación de un sistema de desarrollo gráfico para poder compilar y depurar el

2

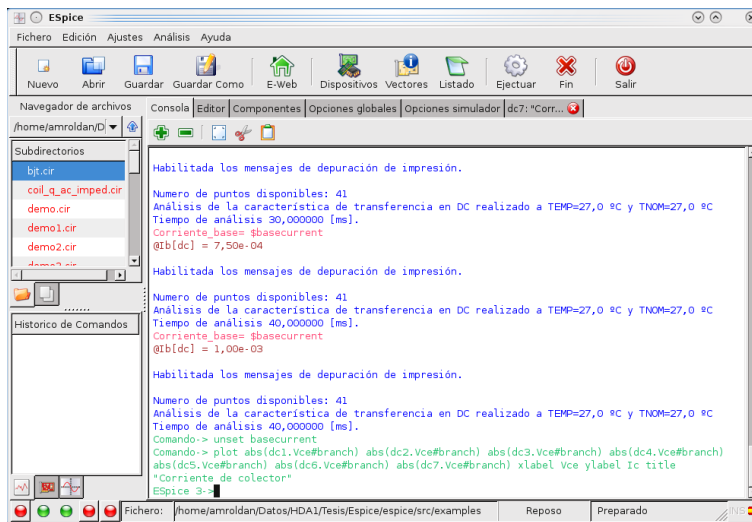


Figura 2.19 – Detalle de los mensaje de depuración orientados a desarrollo de modelos compactos.

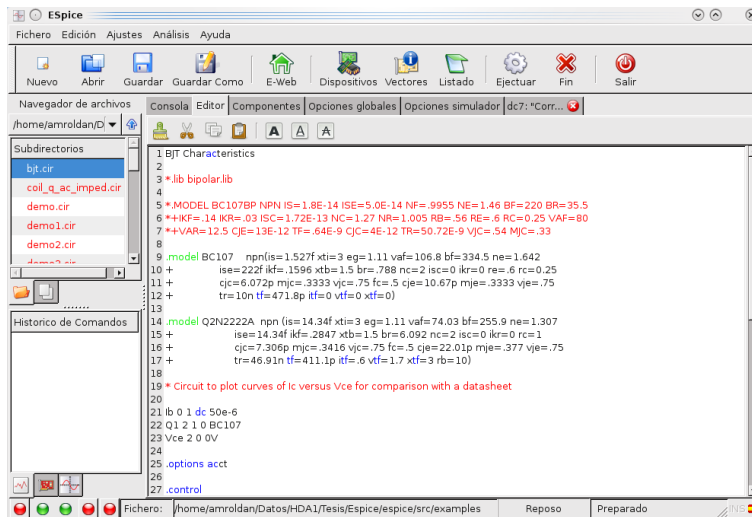
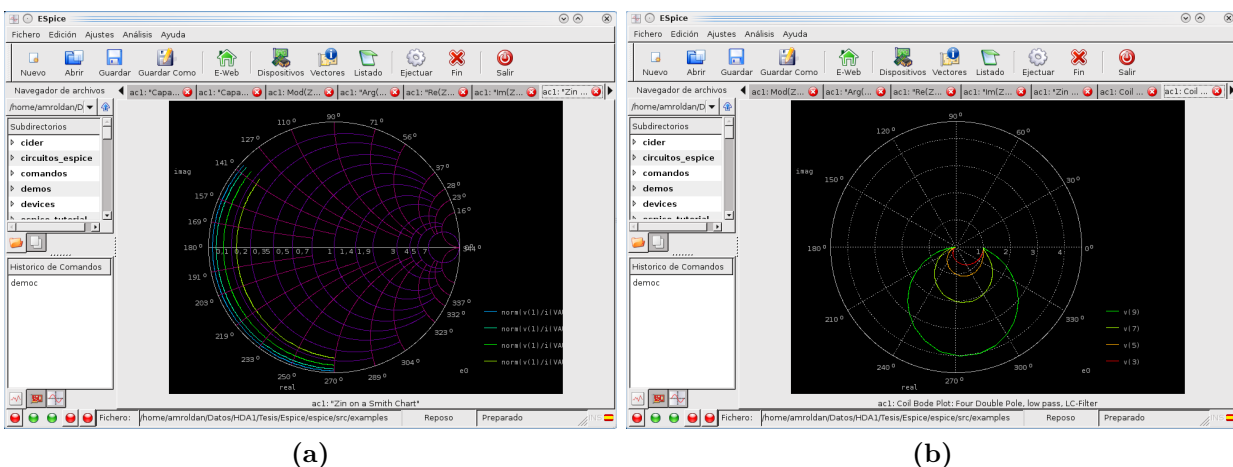


Figura 2.20 – Detalle del editor integrado en el simulador.



(a)

(b)

Figura 2.21 – Representación gráfica con Carta de Smith (a), gráfica en Polares (b).

código fuente de una manera más eficiente. La interfaz gráfico de desarrollo *IDE* elegido fue **Kdevelop**[®] que permite compilar y depurar todos los módulos del simulador, ver figura 2.22.

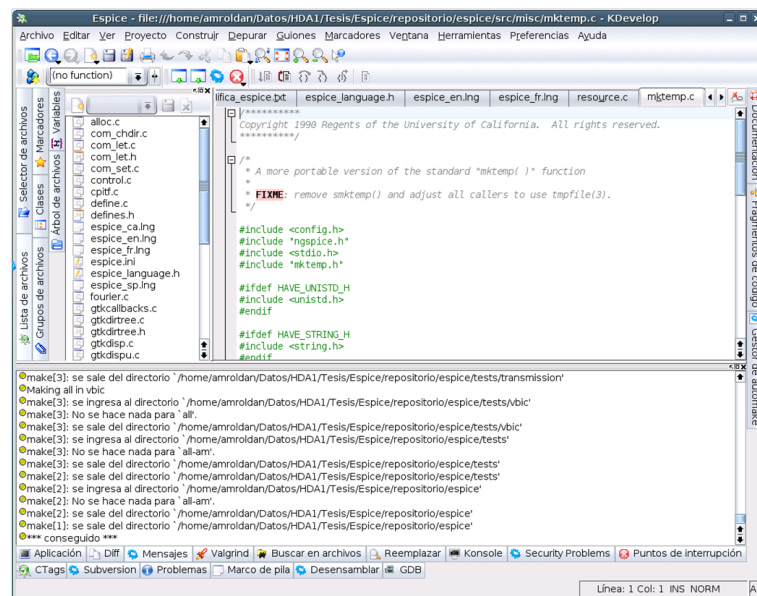


Figura 2.22 – Sistema de desarrollo utilizado en GNU/Linux[®]

2.3.4 Resultados del test de verificación.

Tras la compilación del Simulador **ESPICE** para Windows es necesario verificar el correcto funcionamiento de cada uno de los tipos de análisis posibles con diferentes circuitos. De este modo se pudieron detectar algunos errores que se resolvieron. Al ser un simulador de circuitos de propósito general es muy complicado establecer un conjunto de circuitos de test o verificación que permitan establecer la inexistencia de errores.

El procedimiento propuesto consiste en un conjunto de circuitos de prueba que serán simulados y cuyos resultados serán comprobados numéricamente al compararlos con los obtenidos con otros simuladores similares. El contenido de los Test de verificación se ha incorporado en el Anexo 7.5.

2.3.5 Conclusiones.

Los resultados de las simulaciones son totalmente satisfactorios. Se dispone de un simulador multiplataforma en código fuente que permite su uso en plataformas **Windows**[®] y **GNU/Linux**[®]. Los módulos correspondientes a los modelos compactos pueden ser actualizados y modificados permitiendo la incorporación de nuevos efectos físicos modelados haciendo uso de sus correspondientes expresiones analíticas y sus parámetros.

Aparte de poder realizar simulaciones de circuitos con dispositivos caracterizados por modelos compactos modificados, también se pueden realizar cambios sobre los tipos de aná-

lisis; así como acceder a los valores y estados de depuración internos de éstos. Los resultados numéricos obtenidos en diferentes tipos de simulaciones se ajustan a los caracterizados con otros simuladores comerciales (ADS, Agilent, ELDO, Mentor).

Disponemos, por tanto, de una herramienta de simulación que permite desarrollar y modificar los modelos compactos que se pueden obtener con simuladores de dispositivos desarrollados en el grupo de investigación de nanoelectrónica de la UGR, con simuladores comerciales como por ejemplo ATLAS de Silvaco y con medidas experimentales, y, finalmente, simular circuitos haciendo uso de los dispositivos modelados.

2.4 Adaptación de modelos de dispositivos en código fuente.

La puesta en valor del simulador multiplataforma ESPICE se llevó a cabo mediante la modificación de un modelo compacto basado en código fuente y la simulación de algunos circuitos haciendo uso de dicho modelo. En concreto, se ha complementado el modelo BSIM-SOI3p2 para profundizar en la descripción de los mecanismos de transporte no estacionarios del dispositivo, en particular, en relación al efecto de transporte no local correspondientes al *overshoot* de la velocidad (VO) [RGLVC97, SHWK+88, LR02, RVD+00]. Los dispositivos utilizados fueron MOSFET PD basados en tecnología SOI (Silicon On Insulator).

2.4.1 Modificación del modelo compacto BSIMSOI en código fuente.

La adaptación de un modelo compacto existente en código fuente implica varias operaciones que conllevan la modificación/creación de ficheros .c o .h y su incorporación a la librería del modelo, para su posterior enlazado con el ejecutable del simulador.

El modelo compacto sobre el que introducimos las modificaciones es BSIMSOI3p2, que implementa un modelo unificado para dispositivos SOI parcialmente (*PD*, *Partially depleted*) y completamente (*FD Fully depleted*) depleccionados basado en el concepto de *body-source built-in potential lowering* [PFW+03, Su02]. Este modelo se relaciona con sus predecesores para tecnología convencional bulk (BSIMs), con los cambios necesarios para adaptarlo a tecnología SOI. El dispositivo que se muestra en la figura 2.23 dispone de 5 terminales

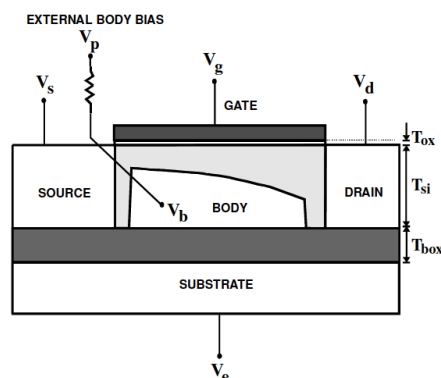


Figura 2.23 – Estructura de un dispositivo *PDSOI* MOSFET .

(fuente, puerta, drenador, sustrato, cuerpo). Consta de un canal fabricado sobre una lámina

de silicio de grosor T_{si} que se encuentra limitada (como un *sandwich*) por dos capas de óxido: el óxido de puerta de grosor T_{ox} y el óxido enterrado (*buried oxide*) de grosor T_{box} . En la configuración de cuerpo (*floating body*) flotante dispone de cuatro terminales de polarización con tensiones de puerta (V_g), drenador (V_d), fuente (V_s) y sustrato (V_e). El potencial de cuerpo (V_b) es iterado durante la simulación del circuito. Si existe un contacto, la tensión de cuerpo estará fijada por una fuente externa que definirá (V_p).

Como se indicó anteriormente, las modificaciones realizadas en *BSIMSOI* están orientadas a proporcionar una descripción más detallada de los mecanismos de transporte, en concreto, de los efectos no locales de transporte, relacionados con el *overshoot* de la velocidad. Para ello se hace uso del siguiente modelo de corriente [RGLVC97, RGLVCC00] (el significado de cada término de la ecuación se desglosa más adelante, dónde se explican los pormenores del modelo).

$$I_{\text{DS}} = \frac{W_{\text{eff}} \mu F(V_{\text{gs}}, V_{\text{ds}})}{L_{\text{eff}} \left[1 + \frac{\mu V_{\text{ds}}}{v_{\text{sat}} L_{\text{eff}}} \right]} + \overbrace{\lambda_{\text{OV}} \frac{W_{\text{eff}}}{L_{\text{eff}}^2} F(V_{\text{gs}}, V_{\text{ds}})}^{\text{Contribución por overshoot de la velocidad}} \quad (2.4.1)$$

El segundo término de la ecuación anterior da cuenta del nuevo efecto. Como se puede apreciar aparece un nuevo parámetro denominado `voshoot`, λ_{OV} en la ecuación anterior, que será explicado con más detalle en el apartado 2.4.3. Esta adición al modelo de corriente obliga a modificar los ficheros de descripción de modelo `IFparm B3SOImPTable[] = { model parameters }` en el fichero `b3soi.c` añadiendo con la directiva `IOP` el parámetro asociado a un número real y su texto de detalle asociado `IOP("voshoot", B3SOI_MOD_VOSHOOOT, IF_REAL, "Velocity Overshoot Effect") // VELOCITY OVERSHOOT`. El procesado del valor numérico asociado al parámetro para un dispositivo se realiza en el módulo de procesado de parámetros de modelo en el fichero `b3soimpar.c` (listado 2.4). El valor obtenido del fichero de descripción del dispositivo se almacena en la estructura de datos `(B3SOImodel*)inMod`. El número de parámetros existentes en el modelo es de 616 incluido éste último.

```

1 int B3SOImParam(param, value, inMod)
2 int param;
3 IFvalue *value;
4 GENmodel *inMod;
5 {
6     B3SOImodel *mod = (B3SOImodel*)inMod;
7     switch(param)
8     {
9     ...
10     case B3SOI_MOD_VOSHOOOT :
11         mod->B3SOIvoshoot = value->rValue;
12         mod->B3SOIvoshootGiven = TRUE;
13         break;
14     ...

```

Listado 2.4 – Listado del código de procesado numérico de los parámetros del modelo.

Para consultar y modificar el parámetro durante la simulación es necesario interrogar a la estructura de datos que almacena los valores de los parámetros, para ello se accede mediante un puntero a `(B3SOImodel *)inst`. Dicho proceso se realiza en el fichero `B3soiask.c`, ver listado 2.5.

```

1 int B3SOImAsk(ckt, inst, which, value)
2 CKTcircuit *ckt;
3 GENmodel *inst;
4 int which;

```

```

5 IFvalue *value;
6 {
7     B3SOImodel *model = (B3SOImodel *)inst;
8     switch(which)
9     {
10    ...
11        case B3SOL_MOD_VOSHOOT:
12            value->rValue = model->B3SOIvshoot;
13            return(OK); /* v3.2 */
14    ...

```

Listado 2.5 – Listado del código de resolución de consultas de los parámetros del modelo.

En caso de que el parámetro `vshoot` no esté disponible, se carga el valor por defecto ($0 \text{ m}^3/\text{V}/\text{s}$) en la estructura de datos del modelo para su uso posterior durante la evaluación de las ecuaciones de la corriente. Este procesado se realiza en el fichero `B3soiset.c`, ver listado 2.6.

```

1 int B3SOIsetup(matrix,inModel,ckt,states)
2 register SMPmatrix *matrix;
3 register GENmodel *inModel;
4 register CKTcircuit *ckt;
5 int *states;
6 {
7     register B3SOImodel *model = (B3SOImodel*)inModel;
8     register B3SOIinstance *here;
9     int error;
10    CKTnode *tmp;
11
12    /* v3.2 */
13    double Vbs0t, Qsi;
14
15    /* loop through all the B3SOI device models */
16    for( ; model != NULL; model = model->B3SOInextModel )
17    {
18        /* Default value Processing for B3SOI MOSFET Models */
19        ...
20        // VELOCITY OVERSHOOT
21        if (!model->B3SOIvshootGiven) //PATCH
22            model->B3SOIvshoot = 0; //Units: m3/V/s
23    ...

```

Listado 2.6 – Listado del código de inicialización por defecto de los valores de los parámetros.

La introducción del parámetro en las ecuaciones de la corriente I_{DS} según la ecuación 2.4.1 obliga a modificar las expresiones matemáticas existentes para la β del dispositivo. En el listado 2.7 que aparece en el fichero `B3soild.c` se muestran las expresiones modificadas que incluyen el uso del parámetro del overshoot de la velocidad. Es imprescindible asegurarse en caso de que no exista el parámetro en los ficheros de descripción del modelo de los dispositivos del circuito, que las expresiones numéricas existentes sean similares a las anteriores.

```

1 int B3SOIload(inModel,ckt)
2 GENmodel *inModel;
3 register CKTcircuit *ckt;
4 {
5     register B3SOImodel *model = (B3SOImodel*)inModel;
6     register B3SOIinstance *here;
7     register int selfheat;
8
9     /* Calculate Ids */
10    CoxWovL = model->B3SOIcox * Weff / Leff;
11    beta = ueff * CoxWovL; //Beta Original
12    beta_vshoot = (model->B3SOIvshoot) * CoxWovL / Leff; //Modificado
13    ...
14    //gche = beta * fgche1 / fgche2; // ORIGINAL
15    gche = fgche1 * (beta/fgche2 + beta_vshoot); // MODIFICADO
16    ...
17    if (here->B3SOIdebugMod > 0)
18    {
19        printf("landa=%g \n", model->B3SOIvshoot);
20        printf("Beta=%g\tBeta_vshoot=%g\t Leff=%g Ueff=%g\t CoxWovL=%g\n", beta,
21            beta_vshoot, Leff, ueff, CoxWovL);
22        printf("vo_shoot = %g \n", (model->B3SOIvshoot));
23        printf("gche_antigua = %g \n", fgche1 * beta/fgche2);

```

Listado 2.7 – Listado del código del cálculo de la corriente I_{DS}

Una vez realizadas las modificaciones indicadas es necesario volver a compilar el proyecto (es decir, todo el simulador circuitos completo) y esperar hasta obtener la nueva versión enlazada del simulador que contiene la versión modificada del modelo compacto BSIMSOI3p2.

2.4.2 *Matching* de los transistores N-MOS y P-MOS.

Es necesario equilibrar las corrientes de los transistores tipo N y P para montar un inversor CMOS (celda básica de los circuitos CMOS). Se ha partido de unos dispositivos descritos por un conjunto de parámetros pertenecientes a una tecnología convencional suministrados por la Universidad de Berkeley. Se ha realizado el *matching* teniendo en cuenta la movilidad de bajo campo y la contribución de los efectos de *overshoot* de la velocidad de los dispositivos N y P [RRG05b]. En el listado 2.8 se describe el circuito usado para obtener las curvas I_{DS} con la variación del parámetro *voshoot* haciendo uso de la primitiva `altermod @p1[voshoot] = $landa*1.5 [LF01]`, de este modo se calculan los efectos de *overshoot* de la velocidad tanto en dispositivos tipo P como N.

```

1 *SOI NMOSFET, FD Berkeley Spice Compatibility
2 vd1 d1 0 dc 1.5
3 vs1 s1 0 dc 0
4 ve1 e1 0 dc 0
5 vg1 g1 0 dc 3
6 vb1 b1 0 dc 0
7 ml d1 g1 s1 e1 b1 p1 w=10u l=0.15u debug = 1
8 .include bsimsoi3p2pmos.mod //Descripción de los parámetros del dispositivo
9 .option gmin=1e-25 itl1=500
10 .control
11 destroy all
12 foreach landa 0 22e-11
13   altermod @p1[voshoot] = $landa*1.5
14   print @p1[voshoot]
15   dc vd1 0 -2 -0.1 vg1 -1.5 -2.4 -0.3
16   *plot vd1#branch title "Landa=$landa \[m3/(Vs)\]"
17   echo "Simulando para Landa=$landa \[m3/(Vs)\]"
18 end
19 plot 1*dc1.vd1#branch 1*dc2.vd1#branch
20 let ids_landa0 =dc1.vd1#branch
21 let ids_landa22=dc2.vd1#branch
22 print ids_landa0[0 ,19] ids_landa22[0 ,19] > pmos_ids_vg1.dat
23 print ids_landa0[20 ,39] ids_landa22[20 ,39] > pmos_ids_vg2.dat
24 print ids_landa0[40 ,59] ids_landa22[40 ,59] > pmos_ids_vg3.dat
25 print ids_landa0[60 ,79] ids_landa22[60 ,79] > pmos_ids_vg4.dat
26 .endc
27 .end

```

Listado 2.8 – SPICE Netlist para comprobación de *MATCHING* entre dispositivos NMOS y PMOS.

A partir de éstas simulaciones, se puede obtener la relación de anchuras para que el dispositivo P esté equilibrado respecto al N [RRG05c]. En la figura 2.24 se muestra el resultado de las curvas I_{DS} vs. V_{DS} para ambos dispositivos. Se puede apreciar un buen ajuste.

2.4.3 Inclusión de los efectos de overshoot de la velocidad en el modelo BSIM-SOI.

Los efectos de VO se han incorporado siguiendo las directrices descritas en el apartado 2.4.1. Para la simulaciones estamos considerando MOSFETs SOI de simple puerta. Una vez se ha introducido el modelo analítico en el código fuente del simulador debemos desarrollar la estrategia para extraer el parámetro nuevo que se incorpora en el modelo (λ_{VO}). Para ello utilizamos las curvas de transconductancia en función de la longitud de canal de

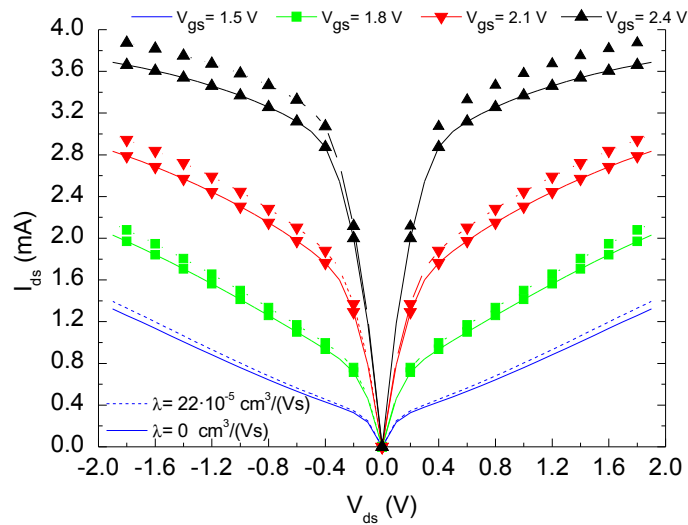


Figura 2.24 – Emparejamiento (matching) para dispositivos BSIMSOI tipo N y P con $\lambda = 0$ y $\lambda = 22 \text{ m}^3/(\text{Vs})$

los transistores, siguiendo el procedimiento descrito en [RGLVC97, RGLVCC00]. Las curvas han sido obtenidas con el simulador de Monte Carlo del Grupo de Nanoelectrónica de la Universidad de Granada. El ajuste de estas curvas con el modelo introducido en la ecuación 2.4.1 permite extraer el valor del parámetro.

2.4.3.1 Tecnología SOI

Los dispositivos de silicio sobre aislante (SOI) están considerados hoy día como la alternativa más seria que existe para sustituir a la tecnología bulk convencional para las futuras familias tecnológicas basadas en nodos de tamaño inferior a 22 nm [ITR11, NCB⁺11, Col04]. Entre otras ventajas presentan gran tolerancia a radiaciones externas, excelente inmunidad a efectos de *latch-up*, mejora de la pendiente subumbral y DIBL, reducción de capacidades parásitas y de los efectos de canal corto SCE. Además, el uso de un sustrato aislante (BOX) puede simplificar en algunos casos el proceso de fabricación. Para los nodos por debajo de 22 nm, los dispositivos multipuerta, muchos de ellos SOI, (*double-gate MOSFETs* (*DGMOSFETs*), *FinFETs*, *Trigate transistors*, *Surrounding Gate Transistor SGT*, etc.) son considerados las mejores opciones para ser considerados la base de la tecnología *mainframe*, debido a su gran inmunidad a los efectos de canal corto, reducción de capacidades parásitas, el uso de sustratos poco dopados y mejora de movilidad de bajo campo producida por la reducción de los mecanismos de dispersión coulombiana y la operación de inversión en volumen.

Las ventajas que presenta la tecnología SOI no sólo la hacen interesante para abordar la sustitución de la tecnología convencional en los nodos más pequeños sino que también le dan valor añadido en aplicaciones analógicas, de baja potencia, fotónica, militares y aeroespaciales [Con11].

2.4.3.2 Efectos del overshoot de la velocidad

El efecto del *overshoot* de la velocidad ha sido estudiado en profundidad tanto teórica como experimentalmente [PSB93, RGLVC97, SHWK+88]. Cuando la longitud de canal de los dispositivos se reduce por debajo de 200 nm pueden aparecer estos efectos de transporte no estacionario que suponen un aumento de la corriente de drenador y de transconductancia respecto al valor de referencia correspondiente al régimen de transporte estacionario. El transporte no estacionario de portadores inyectados desde la fuente al canal produce que la velocidad de éstos supere al valor de velocidad de saturación (máximo valor que se obtendría en régimen de transporte estacionario). Los efectos de *overshoot* de la velocidad aumentan, en general, al reducirse el tamaño de los transistores. Su incorporación en los modelos de los dispositivos es esencial para poder reproducir con fiabilidad las características más importantes a utilizar en simuladores de circuitos.

El modelado de los efectos de VO se realiza desde diferentes aproximaciones teóricas. Una de ellas, adapta las ecuaciones del transporte en su versión hidrodinámica para obtener la energía y temperatura de los portadores a lo largo del canal, de este modo se modifican las ecuaciones convencionales para obtener la corriente en el dispositivo y se puede tener en cuenta el aumento de velocidad de los portadores, que en esta aproximación, viene caracterizado por los tiempos de relajación del momento y de la energía [Lun00]. Otra aproximación, la que vamos a utilizar en nuestro estudio, se basa en una reformulación de la expresión de la velocidad de los portadores, añadiendo un nuevo término que depende del gradiente de campo longitudinal en el canal del transistor. La ecuación de la corriente se modifica para incluir este nuevo término. Con esta técnica se obtiene una expresión compacta y fácil de incluir en un desarrollo analítico. Además la estructura matemática de la expresión no añade inestabilidades numéricas, según nuestra experiencia en varios simuladores de dispositivos, en la resolución iterativa derivada del análisis de circuitos con dispositivos y modelos que incluyen efectos de VO [RGLVC97, RGJM+10]. Finalmente, también se puede abordar el modelado de los efectos de VO mediante la teoría de la dispersión, basada en la formulación matricial de los mecanismos de transporte típicamente aplicada en el contexto de los sistemas mesoscópicos [LR02, RVDL01, RRL02].

En general, la presencia de altos gradientes de campo eléctrico en zona de canal de transición fuente-canal provoca un aumento de los efectos de VO. Los portadores se aceleran rápidamente y alcanzan velocidades mayores a la velocidad de saturación. Si los canales son muy cortos, un porcentaje de ellos podría cruzar el canal sin sufrir dispersiones, lo que se suele denominar transporte quasi-balístico [Lun00].

Siguiendo el modelo de VO desarrollado en el grupo de nanoelectrónica de la UGR [RGLVC97, RGLVCC00] hemos modificado el modelo BSIMSOI y adaptado ESPICE para poder simular circuitos y caracterizar los efectos de VO en diferentes configuraciones circuitales.

2.4.3.3 Simulaciones y resultados.

Hemos utilizado un simulador ensemble Monte Carlo 2D de **MOSFET SOI** para caracterizar los dispositivos que estamos modelando. Las ecuaciones de Poisson y Schroedinger bidimensionales se resuelven autoconsistentemente con la ecuación de Boltzmann mediante el método de Monte Carlo [SMGGR06]. Los mecanismos de dispersión más importantes considerados son por interacción coulombiana, rugosidad superficial y fonones [GGS04]. Diferentes dispositivos **MOSFET SOI** de puerta simple han sido estudiados. Los dispositivos simulados poseen longitudes de canal entre 0.5 to 0,06 μm . En estos dispositivos los efectos del VO son importantes.

El modelo utilizado para incluir estos efectos es el siguiente:

$$I_{DS} = \frac{W_{eff}\mu F(V_{gs}, V_{ds})}{L_{eff} \left[1 + \frac{\mu V_{ds}}{v_{sat} L_{eff}}\right]} + \lambda_{OV} \frac{W_{eff}}{L_{eff}^2} F(V_{gs}, V_{ds}) \quad (2.4.2)$$

donde W_{eff} es la anchura efectiva del canal, L_{eff} la longitud efectiva del canal, v_{sat} la velocidad de saturación, μ la movilidad de bajo campo de los portadores en el canal y λ_{OV} el parámetro que caracteriza los efectos del *overshoot* de velocidad.

Se ha adaptado la ecuación de corriente de drenador del modelo BSIMSOI3p2 [HN05] para reflejar el segundo término de la ecuación 2.4.2. La función $F(V_{gs}, V_{ds})$ (ecuación 2.4.3) representa la integral de la carga en inversión en el canal [RGLVC97] y ha sido calculada siguiendo el desarrollo del modelo BSIMSOI3p2.

$$F(V_{gs}, V_{ds}) = \int_0^{V_{ds}} Q(V) dV \quad (2.4.3)$$

La extracción del parámetro λ_{OV} se ha realizado ajustando las curvas de transconductancia obtenidas mediante simulación y las predicciones del modelo expresado en la ecuación anterior, siguiendo la referencia [RGLVC97]. El ajuste se muestra en la figura 2.25. El parámetro de *overshoot* de velocidad obtenido para dos tensiones de puerta diferentes fue el mismo, $\lambda_{VO} = 22 \times 10^{-5} \text{ cm}^3/\text{Vs}$.

El valor obtenido para λ_{VO} es coherente con los valores utilizados previamente para reproducir valores experimentales como $\lambda_{VO} = 25 \times 10^{-5} \text{ cm}^3/\text{Vs}$ (**MOSFET** convencional) [RGLVC97] y $\lambda_{VO} = 10 \times 10^{-5} \text{ cm}^3/\text{Vs}$ (para **MOSFET SOI**) [RGLVCC00]. El valor máximo de la transconductancia que podría obtenerse con un régimen de transporte estacionario (limitando la velocidad de los portadores a la velocidad de saturación) sería ($g_m^{max} = C_{ox} \cdot v_{sat}$). Los valores representados por encima de este nivel muestran claramente la existencia de los efectos del *overshoot* de la velocidad.

El modelo BSIMSOI modificado y el valor obtenido para el nuevo parámetro se ha utilizado para simular dispositivos **MOSFET SOI** individuales. No se han simulado con Monte Carlo los dispositivos **MOSFET** tipo P; sin embargo, se han utilizado resultados publicados por Laux et al. [LF01] que muestran que los efectos del *overshoot* de la velocidad incrementan la corriente de drenador de dispositivos de 0.1 μm de longitud de canal en un

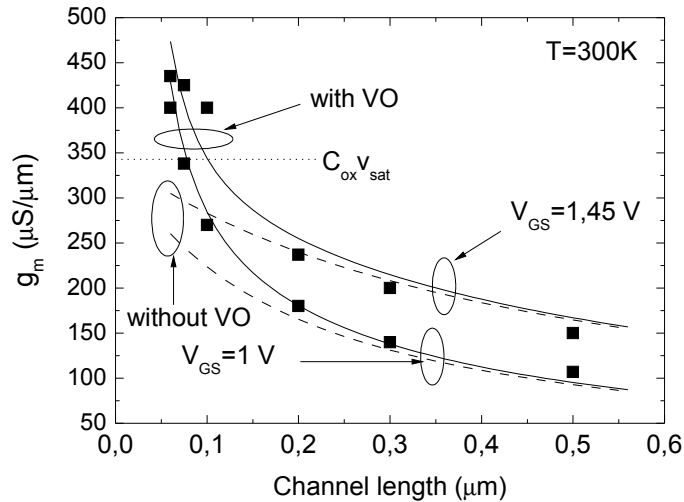


Figura 2.25 – Transconductancia de un NMOSFET SOI de puerta simple en función de la longitud de canal a temperatura ambiente. Los datos en símbolos representan las simulaciones de Monte Carlo. El modelo de transconductancia queda representado por líneas (discontinuas despreciando efecto VO $\lambda_{VO}=0$ y continuas incluyendo el efecto VO $\lambda_{VO} \neq 0$). La movilidad de bajo campo y λ_{VO} usados para ajustar las curvas g_m son: a) $V_{GS}=1$ V ($\lambda_{VO} = 22 \times 10^{-5}$ cm³/Vs y $\mu = 230$ cm²/Vs) b) $V_{GS} = 1,45$ V ($\mu = 570$ cm²/Vs y $\lambda_{VO} = 22 \times 10^{-5}$ cm³/Vs)

factor del 20 % (NMOS) y 30 % (PMOS). Hemos utilizando relaciones similares en nuestro caso para los efectos del VO en los dos elementos del inversor CMOS.

Las curvas de salida para un dispositivo MOSFET SOI de 170nm de longitud de canal se muestran en la figura 2.26. El resto de parámetros tecnológicos usados son: $T_{OX} = 5$ nm, $T_{BOX} = 500$ nm, $T_{Si} = 90$ nm, $N_A = N_D = 8 \times 10^{17}$ cm⁻³. Como se observa, las diferencias

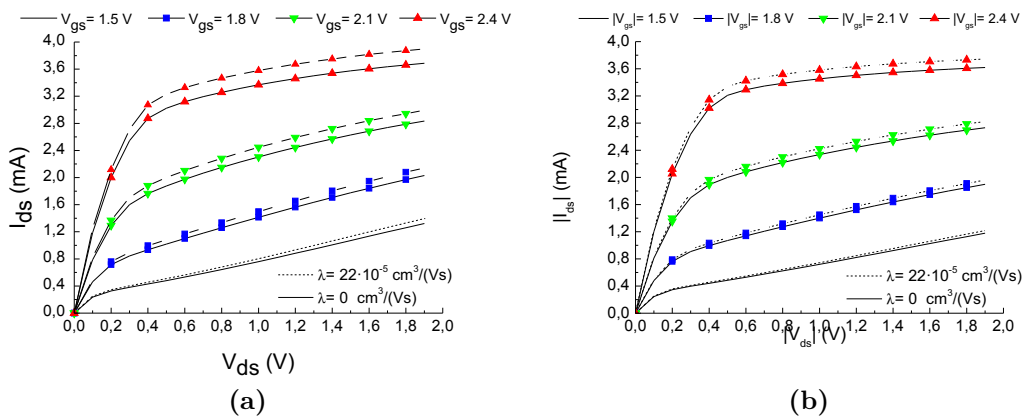


Figura 2.26 – Curvas de salida de dispositivos de puerta simple SOI PMOSFET (a) y NMOSFET (b) a temperatura ambiente, para el modelo BSIMSOI mejorado con valores diferentes de VO $\lambda_{VO}=0$ (líneas continuas - despreciando los efectos del VO) y $\lambda_{VO} = 22 \times 10^{-5}$ cm³/Vs (valor del parámetro VO extraído de la figura 2.25 representado en líneas discontinuas).

entre ambos conjuntos de curvas se incrementan a mayores tensiones drenador-fuente (V_{DS}), como se espera, ya que los efectos del VO están relacionados con el incremento del campo eléctrico longitudinal en el canal del dispositivo [RGLVC97, SL01].

Una vez puesto a punto el modelo compacto, hemos simulado un anillo oscilador CMOS de 51 etapas inversoras con ESPICE. Los dispositivos son similares a los utilizados anteriormente. La capacidad de carga establecida es de $C_L = 1$ pF y la tensión de alimentación es de $V_{DD} = 2$ V. El esquema completo del circuito simulado se muestra en la figura 2.27.

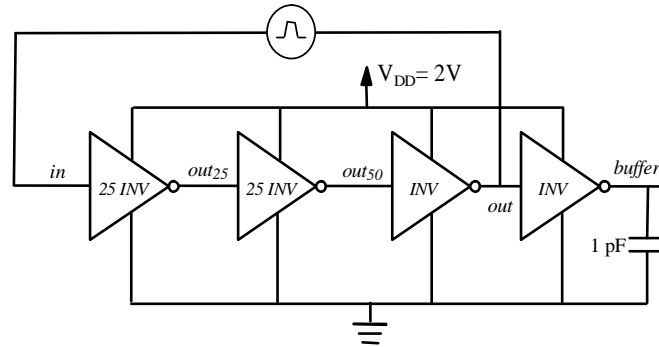


Figura 2.27 – Esquema circuital del oscilador en anillo de 51 etapas inversoras usado para estudiar la influencia de los efectos del VO en la frecuencia de oscilación.

Los resultados se muestran en la figura 2.28 donde se compara la tensión de salida obtenida para diferentes valores del parámetro λ_{VO} .

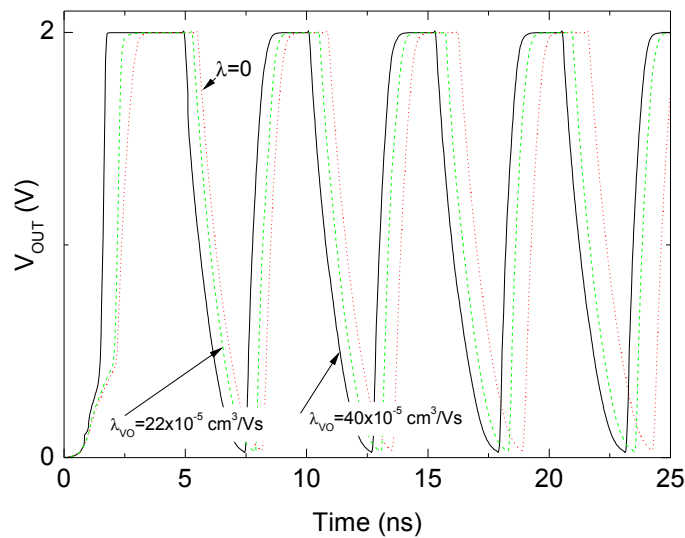


Figura 2.28 – Tensión de salida del el circuito oscilador en función del tiempo para diferentes valores del parámetro VO a temperatura ambiente.

En la siguiente figura se representa la frecuencia de oscilación del anillo de 51 etapas en función del valor del parámetro de VO. Como se observa, se produce un incremento importante de la frecuencia de oscilación con la variación del parámetro λ_{VO} . La simulación para $\lambda_{VO}=0$ representa el caso donde el régimen de transporte es estacionario y los efectos del VO son despreciables.

La relación entre la frecuencia de oscilación y el parámetro del VO es lineal. Este resultado es importante, ya que el control de los efectos de VO con una determinada tecnología que aumente los perfiles del gradiente del campo longitudinal en la zona de fuente (lo que implicaría un aumento de los efectos de VO) permitiría una mejora de la velocidad de

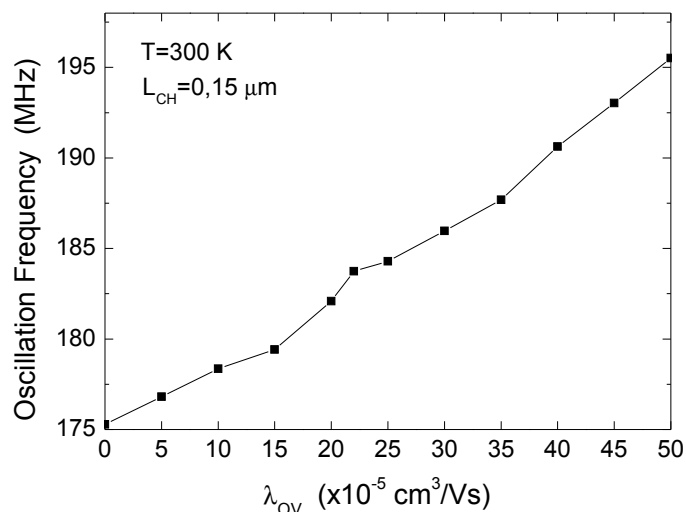


Figura 2.29 – Frecuencia de oscilación vs. el parámetro de VO a temperatura ambiente.

operación de algunos circuitos digitales basados en estos dispositivos.

En la siguiente figura se puede observar la variación de la frecuencia de oscilación en función de la longitud de canal de los dispositivos.

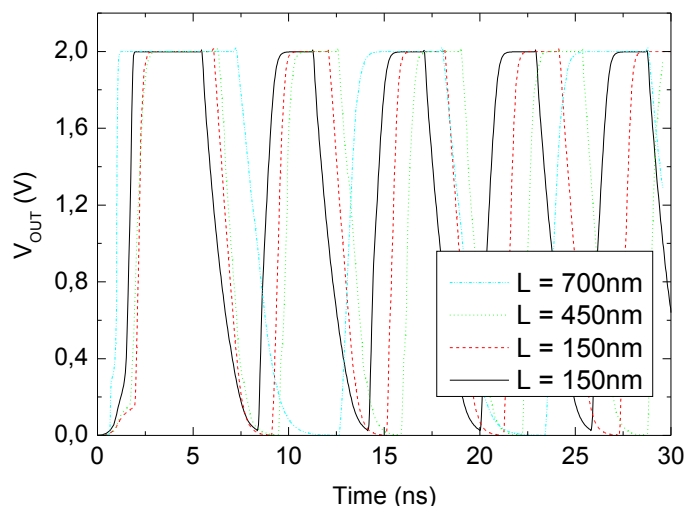


Figura 2.30 – Tensión de salida del anillo oscilador en función del tiempo para diferentes longitudes de canal a temperatura ambiente.

Por otro lado, en la figura 2.31 se muestra la frecuencia de oscilación en función de la longitud de canal incluyendo y despreciando los efectos del VO. Por debajo de los 250 nm la separación entre las curvas por la contribución de los efectos de VO es importante.

Por último hemos simulado la variación de la frecuencia de oscilación del anillo en función de la longitud de canal. Con el fin de ver la influencia de los efectos de VO hemos representado los resultados con y sin estos efectos. La contribución aumenta a medida que se reduce el canal, como es lógico, dada su dependencia con la derivada del campo longitudinal. Extrapolando los resultados para longitudes de 45 nm y menores se puede observar que la inclusión de estos efectos en los modelos compactos de los transistores es indispensable para describir adecuadamente su comportamiento y el de los circuitos basados en ellos.

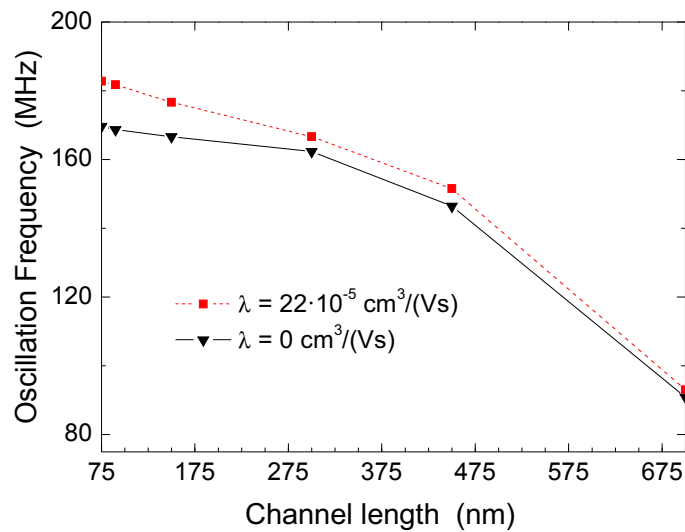


Figura 2.31 – Frecuencia de oscilación vs. longitud del canal a temperatura ambiente incluyendo los efectos de VO (líneas discontinuas) y despreciándolo (líneas continuas).

En el siguiente capítulo presentaremos la implementación de modelos compactos en Verilog-A. Constituye otro enfoque que presenta bastantes ventajas, lo que le ha hecho prevalecer y convertirse en un estándar actual. No obstante, en determinadas ocasiones, y dependiendo del contexto de la simulación de circuitos, puede ser interesante la posibilidad de modificar modelos existentes en código fuente como se ha descrito en este capítulo.

IMPLEMENTACIÓN DE MODELOS AVANZADOS DE DISPOSITIVOS MULTIPUERTA NANOMÉTRICOS EN VERILOG-A.

3.1 Dispositivos multipuerta.

Los dispositivos **MOSFET** multipuerta (*multi-gate*) son considerados como la alternativa más plausible a la tecnología *bulk* convencional para las próximas familias tecnológicas a utilizar en los circuitos integrados [ITR11]. Estos dispositivos incluyen características que permiten importantes disminuciones de la longitud de canal frente a sus homólogos *bulk* por su potencial en la reducción de los efectos de canal corto (**SCE**). Entre sus ventajas destacan la reducción del *threshold voltage roll-off*, *drain-induced barrier lowering (DIBL)*, *sub-threshold swing* y también disminución de las capacidades parásitas. En la figura 3.1 se muestra una evolución tecnológica y cronológica del tamaño de los transistores **MOSFET** en cuanto a la longitud de canal. Se presentan las fotografías de las estructuras de los dispositivos más probables para una determinada longitud de puerta además de los elementos diferenciadores ya sea por los materiales utilizados o la estructura de puertas establecida.

Actualmente la comunidad científica está realizando grandes esfuerzos en las tareas de fabricación, caracterización y simulación de estos dispositivos [CC03, NCB⁺11, Col08]. En

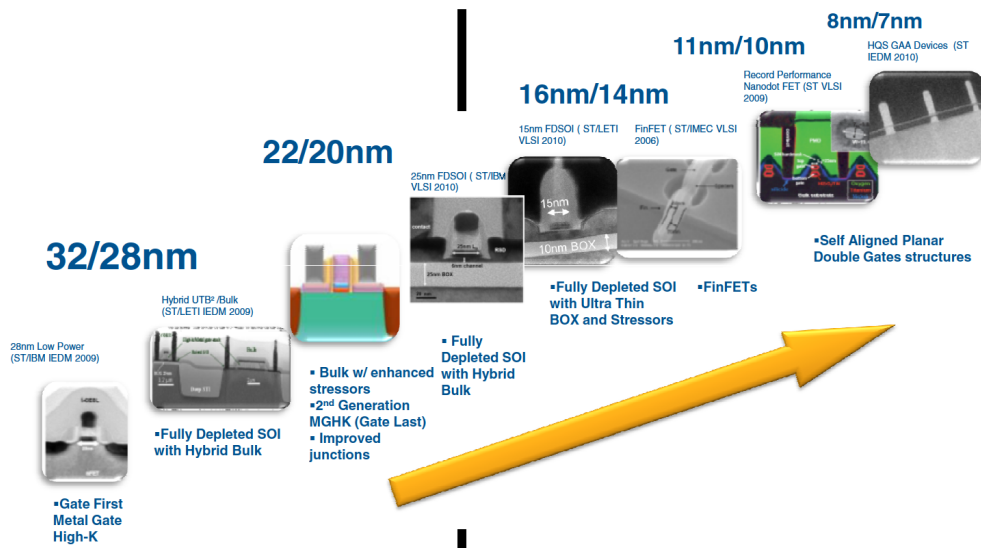


Figura 3.1 – Evolución de los dispositivos electrónicos en el contexto de la Ley de Moore [Sko11].

este contexto, es importante destacar los avances en la línea del modelado compacto de los dispositivos multipuerta [TDC⁺08, TF05, KHC08, TFG04, LVCCG⁺00, GRRB04, IJR⁺05, JIS⁺04, LI08, CP07, AP97, RGGB08, RGLVC97, RGLVCC00, RGJM⁺10]. Las características de carga, las capacidades más importantes de la estructura, movilidad de bajo campo, etc., están siendo descritas de manera analítica para incorporar los nuevos efectos físicos que aparecen cuando se reducen a escala nanométrica los dispositivos electrónicos. Entre estos efectos se encuentran el aumento del confinamiento estructural (*structural confinement* - SC) y el confinamiento eléctrico (*electrical confinement* - EC) también presente en los dispositivos *bulk*. Con la reducción del tamaño de los dispositivos, los efectos del confinamiento estructural crecen, lo que implica que la distribución de carga en estos dispositivos sea completamente diferente a la de los dispositivos *bulk* convencionales. Por esta razón, es necesario que los nuevos modelos describan con precisión la física de los dispositivos multipuerta. La complejidad física de estas estructuras complica el modelado y aumenta la carga algebraica de las expresiones analíticas. Sin embargo, existe una necesidad imperiosa de implementar los modelos en simuladores de circuitos para abordar las etapas de diseño circuital con éxito y poder así desarrollar plenamente estas tecnologías.

La reducción del tamaño de los MOSFET por debajo de los 22 nm está alcanzando el límite de la tecnología actual en cuanto a lo que se utiliza en producción actualmente. Los dispositivos MOSFET convencionales requerirán un alto dopado en el canal para mantener un control electrostático adecuado, lo que implicará un mayor *tunneling band-to-band*, *gate induced drain leakage* y gran variabilidad producida por las fluctuaciones estadísticas del dopaje del canal. Los dispositivos de simple puerta SOI (FDSOI) o multipuerta con bajos dopajes en el canal son los candidatos más recomendables para permitir el escalado más allá de longitudes de puerta de 22 nm. Los retos más importantes de los dispositivos SOI son el control del grosor de la lámina de silicio y su variabilidad y la optimización de las zonas de acceso (los contactos de los dispositivos) para aprovechar el comportamiento intrínseco.

Las tecnologías multipuerta, como ya se ha dicho anteriormente, ver figura 3.2, pre-

sentan importantes ventajas sobre la tecnología *bulk* convencional desde el punto de vista electrostático y del transporte. Una de estas estructuras multipuerta, los *MOSFET undoped surrounding-gate transistors* (SGT) (mostrado como *GAA gate-all-around* en la figura siguiente) muestran un excelente control de los efectos de canal corto y tienen un gran potencial de escalado [JIS⁺04, NCB⁺11]. Diferentes modelos de control de carga han sido publicados utilizando la aproximación de clásica sin tener en cuenta los efectos cuánticos [IJR⁺05, LI08, CP07]. Sin embargo, cuando el radio del cilindro y el grosor de óxido se reducen los fenómenos cuánticos no pueden ser ignorados.

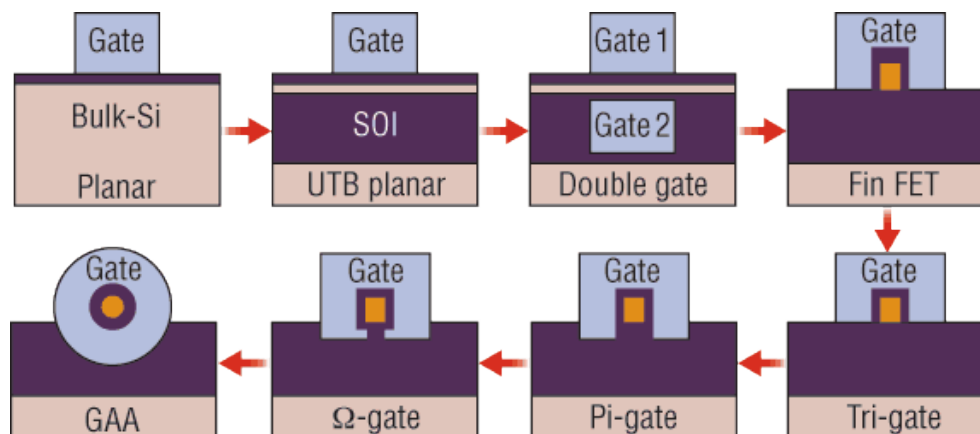


Figura 3.2 – Dispositivos multipuerta [LBS⁺09].

Para las aplicaciones digitales, se distinguen tres escenarios: alto rendimiento (*High Performance* - HP), baja potencia (*low operating power* LOP) y baja potencia en reposo (*low stand-by power* - LSTP). Las aplicaciones HP representan los circuitos integrados de gran complejidad que requieren altas frecuencias de reloj y que pueden manejar altos consumos como los ejemplos de microprocesadores para ordenadores de sobremesa. El objetivo del aumento de frecuencia requerido por la evolución de una tecnología de dispositivos a otra de menor longitud de canal implica que es necesario disminuir el tiempo intrínseco de conmutación en aproximadamente un 17% al año. Este tiempo de conmutación (τ , ecuación 3.1.1) es el tiempo necesario para que un transistor en conducción sea capaz de controlar al puerta de otro transistor a cambiar su salida desde masa a V_{DD} .

$$\tau = \frac{C \cdot V}{I} \quad (3.1.1)$$

donde C representa la capacidad de puerta, V la tensión de alimentación e I la corriente de conducción del dispositivo.

El procedimiento más eficiente de aumentar el rendimiento es hacer más pequeña la puerta del dispositivo y aumentar la corriente de conducción [AR09]. La tecnología LOP representa las aplicaciones de alto rendimiento portátil, como los ordenadores portátiles. Un objetivo fundamental es aumentar el rendimiento del circuito mientras que se reduzca al máximo el consumo mientras el circuito está activo. A nivel de dispositivo implica disminuir la potencia dinámica consumida, que puede modelarse mediante el producto potencia-retraso ($P \cdot \tau$, ecuación 3.1.2, representa la energía requerida para la conmutación de un único

transistor).

$$P \cdot \tau = C \cdot V^2 \quad (3.1.2)$$

donde C representa la capacidad de puerta, V la tensión de alimentación del dispositivo. El procedimiento más eficiente para disminuir la potencia dinámica consiste en disminuir la tensión de alimentación tanto como sea posible.

Finalmente, la opción de baja potencia en reposo (LSTP) se utiliza en aplicaciones de bajo coste y gran economía de escala como teléfonos móviles. En este caso se persigue incrementar el rendimiento manteniendo el consumo de energía tan baja como se pueda mientras el circuito integrado se encuentra desocupado. A nivel de transistor el consumo estático está dominado por las corrientes de fugas, por lo que la tecnología requerida debe proveer una baja corriente de corte (*off current*) [AR09, NCB⁺11].

En la tabla 3.1 se presentan los requerimientos de los dispositivos multipuerta MOSFET para las futuras tecnologías de 27 a 16 nm (*Node*¹) estimada por el equipo de “*Process Integration, Devices and Structures*” del ITRS 2011. Se detallan entre otros, los parámetros L_{gate} ², EOT ³, R_{DS} ⁴, CV/I ⁵, I_{off} ⁶, I_{on} ⁷.

Las implicaciones que la perspectiva de evolución tecnológica del ITRS sugieren que el control electrostático del transistor debe ser mejorado para mantener una baja corriente de corte (I_{off} ⁶) y a la vez una gran corriente de conducción (I_{on} ⁷). La variabilidad intrínseca de los dispositivos tiene que ser controlada de manera que se asegure un gran rendimiento de los circuitos integrados. Para reducir las corrientes de fugas parásitas e incrementar el acoplamiento capacitivo puerta-canal se tendrán que utilizar dieléctricos de alta permitividad y puertas de metal. Las resistencias series y las capacidades parásitas deberán ser también minimizadas. Se puede observar que no se espera la supervivencia de la tecnología *planar bulk* más allá del 2019 para las aplicaciones HP y del 2016 para las LOP y LSP.

Una vez han sido establecidas las ventajas tecnológicas y la proyección futura de los dispositivos multipuerta, es conveniente establecer una clasificación de estos dispositivos: a) los transistores planares en los que la puerta y el canal son horizontales, b) *FINFET* en los que el canal es vertical y la conducción es paralela a la superficie de la oblea, y c) los transistores verticales en los que la dirección de conducción es vertical. Aunque las estructuras se comportan de manera similar en cuanto al control electrostático (salvando las diferencias que las distintas orientaciones cristalográficas del semiconductor de canal producen), las secuencias de los procesos de integración son bastante diferentes unas de otras, cada una con sus ventajas e inconvenientes. Estas diferencias geométricas y operativas fuerzan la utilización

¹*Node*: Tamaño mínimo de la estructuras litográficas que se pueden usar de una manera matricial en un proceso de fabricación. Puede diferir ligeramente para memorias DRAM, ASIC o microprocesadores de una misma tecnología de fabricación.

² L_{gate} : longitud del canal del dispositivo.

³ EOT : *Equivalent Oxide Thickness* (nm).

⁴ R_{DS} : Resistencia parásita efectiva serie fuente-drenador ($\Omega - \mu\text{m}$).

⁵ CV/I : Retraso intrínseco del dispositivo NMOS (ps).

⁶ I_{off} : Corriente I_{DS} de corte (nA/ μm).

⁷ I_{on} : Corriente I_{DS} de saturación para dispositivos NMOS ($\mu\text{A}/\mu\text{m}$).

Option	Year	High Performance			Low Operating Power			Low Standby Power		
		2013	2016	2019	2013	2016	2019	2013	2016	2019
Node (nm)	Bulk	27	19	13	27	19	13	27	19	13
	MG									
L_{gate} (nm)	Bulk	20	15.3	11.7	20	15.3	11.7	20	15.3	11.7
	MG									
EOT (nm)	Bulk	0.79	0.61	-	0.8	-	-	1	-	-
	MG	0.84	0.76	0.65	0.9	0.86	0.74	1.2	1.1	0.9
V_{dd} (V)	Bulk	0.85	0.77	-	0.67	-	-	0.87	-	-
	MG			0.71		0.61	0.55		0.78	0.7
I_{off} (nA/ μ m)	Bulk	100	100	-	5	-	-	0.01	-	-
	MG			100		5	5		0.01	0.01
I_{on} (μ A/ μ m)	Bulk	1422	1670	-	555	-	-	378	-	-
	MG	1470	1684	1857	701	787	791	642	631	624
R_{sd} ($\Omega \cdot \mu$ m)	Bulk	206	149	-	222	-	-	255	-	-
	MG	285	257	202	350	270	208	390	300	218
CV/I (ps)	Bulk	0.51	0.37	-	1.04	-	-	1.92	-	-
	MG	0.41	0.29	0.21	0.6	0.47	0.37	0.9	0.71	0.57

Tabla 3.1 – *Futuros parámetros tecnológicos hasta el 2019 [ITR11] para las tecnologías planar bulk en color rojo y multi-gate en azul.*

de diferentes modelos compactos para caracterizar el comportamiento de los dispositivos en aplicaciones de distinta potencia y frecuencia.

Los modelos compactos para dispositivos MOSFET deben satisfacer los requerimientos específicos de las aplicaciones donde van a ser usados. Éstos para el diseño analógico necesitan de unas condiciones muy estrictas en cuanto a la continuidad numérica de la corriente de drenador así como en sus derivadas en todas las regiones de operación. Las aplicaciones de radiofrecuencia necesitan de la incorporación del modelado de ruido y de distorsión. Con el avance de la tecnología y la disminución del tamaño de los dispositivos, la descripción precisa del comportamiento del dispositivo a sido desplazada por nuevos requerimientos relacionados con cuestiones como la sobrecarga computacional y complejidad del proceso de extracción de parámetros. Los modelos más actuales incorporan un gran número de parámetros físicos además de un buen número de parámetros empíricos, aumentando la complejidad del modelo y su codificación.

Existen varios tipos de modelos compactos en la actualidad que permiten describir con bastante precisión las tecnologías existentes de dispositivos MOSFET. En la figura 3.3 se presentan las tres categorías de modelos compactos con más difusión, así como las relaciones entre ellos y algunos modelos de dominio público y propietarios más significativos de las citadas categorías.

Las categorías más relevantes son [WME+05]:

- **Modelos basados en la tensión umbral (V_T Threshold Voltage Model - TVM):** Pertenecen a esta categoría los primeros modelos compactos MOSFET desa-

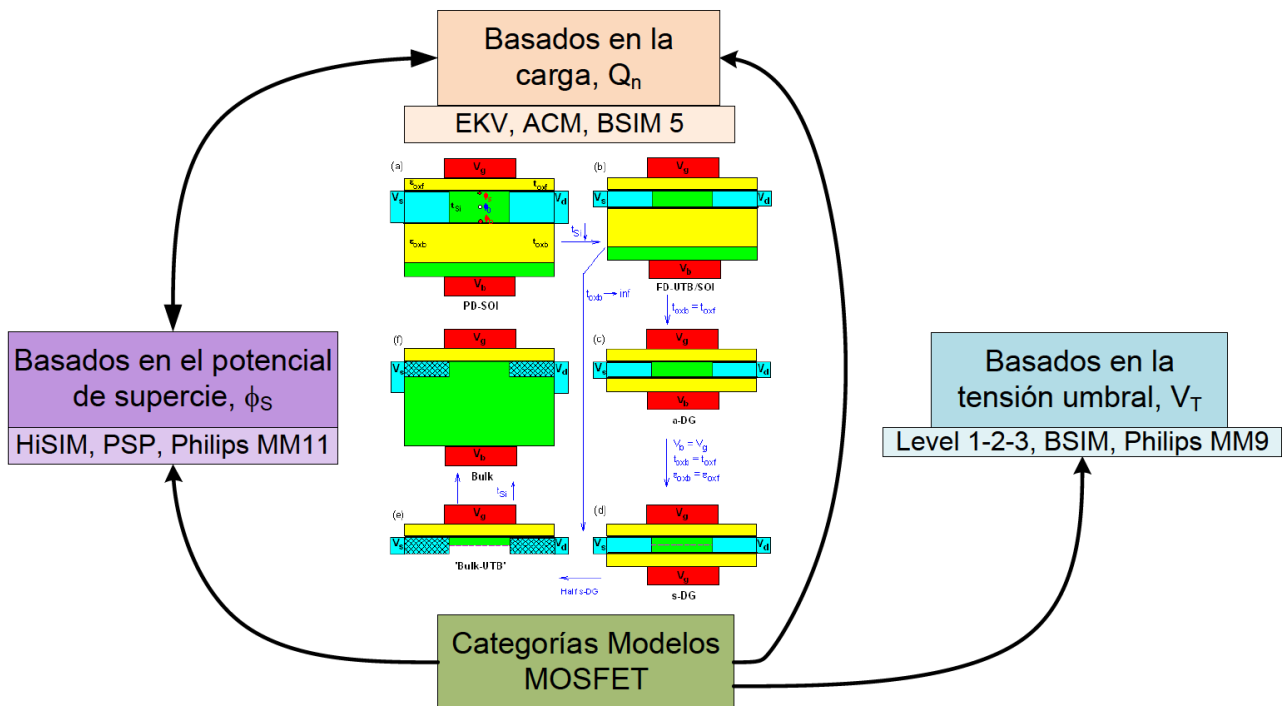


Figura 3.3 – Modelos compactos para dispositivos *MOSFET*, las relaciones existentes entre ellos y algunos de los modelos de dominio público y propietarios más significativos de las citadas categorías

rollados como dominio público (LEVEL1, LEVEL2, LEVEL3) y los posteriores BSIM⁸1, BSIM2, BSIM3 y BSIM4 desarrollados en la Univ. de Berkeley, y el modelo MM9 desarrollado por Philips con un enfoque más industrial. Los modelos basados en la tensión umbral son muy sencillos gracias a la aplicación de la aproximación de lámina de carga (*charge-sheet model* (CSM) [Bre78]) e implementan una descripción como función a trozos en la zona de inversión débil y fuerte, diferenciadas por la tensión V_T . Los modelos de Berkeley representan tres generaciones de modelado [BSI05]. A las características iniciales de la familia LEVEL1-3 se añadieron las incluidas en el BSIM1⁸ constituyen una segunda generación que introdujo las dimensiones del dispositivo en las ecuaciones del modelo y los parámetros, además de nuevos parámetros de ajuste. Estos modelos se han usado durante más de tres décadas porque han satisfecho las necesidades de los diseñadores.

- **Modelos basados en carga (Q_i ; *Inversion Charge Model-ICM*):** El formalismo aplicado en este tipo de modelos permite obtener la corriente de drenador en función de la densidad de carga en inversión en la fuente y el drenador. Este planteamiento evita algunos de los mayores problemas de los modelos basados en tensión umbral como la simetría y la continuidad. Este enfoque ha demostrado ser muy útil en aplicaciones analógicas de bajo consumo. El modelo se fundamenta en la descripción física del dispositivo y añade un conjunto reducido de parámetros de ajuste empírico. El proceso de extracción es relativamente sencillo [EKV11, ACM11]. Los modelos de mayor difusión

⁸BSIM: Berkeley Short Channel IGFET MOSFET.

entre los diseñadores son EKV⁹, ACM¹⁰ y BSIM⁵.

- **Modelos basados en el potencial de superficie (ϕ_s Surface Potencial Model-SPM):** Estos modelos describen las cargas y la corriente en función del potencial de superficie ϕ_s que es resuelto explícita o iterativamente. La carga computacional adicional derivada del cálculo de los potenciales se ha reducido por el uso de algoritmos analíticos (PSP¹¹) o numéricos (HiSIM¹²) eficientes. Estos modelos obtienen una expresión continua para la corriente a partir de las ecuaciones del transporte de difusión y deriva, y sus derivadas son continuas en todo el régimen de operación. Presentan, por tanto, ventajas para la simulación de circuitos de radiofrecuencia.

En la figura 3.4 se presenta la evolución histórica de los tres tipos de modelos compactos citados anteriormente y el uso en dispositivos MOSFET convencionales (*bulk CMOS*) y no convencionales (SOI y multipuerta, ver figura 3.2) donde se detallan los métodos numéricos usados.

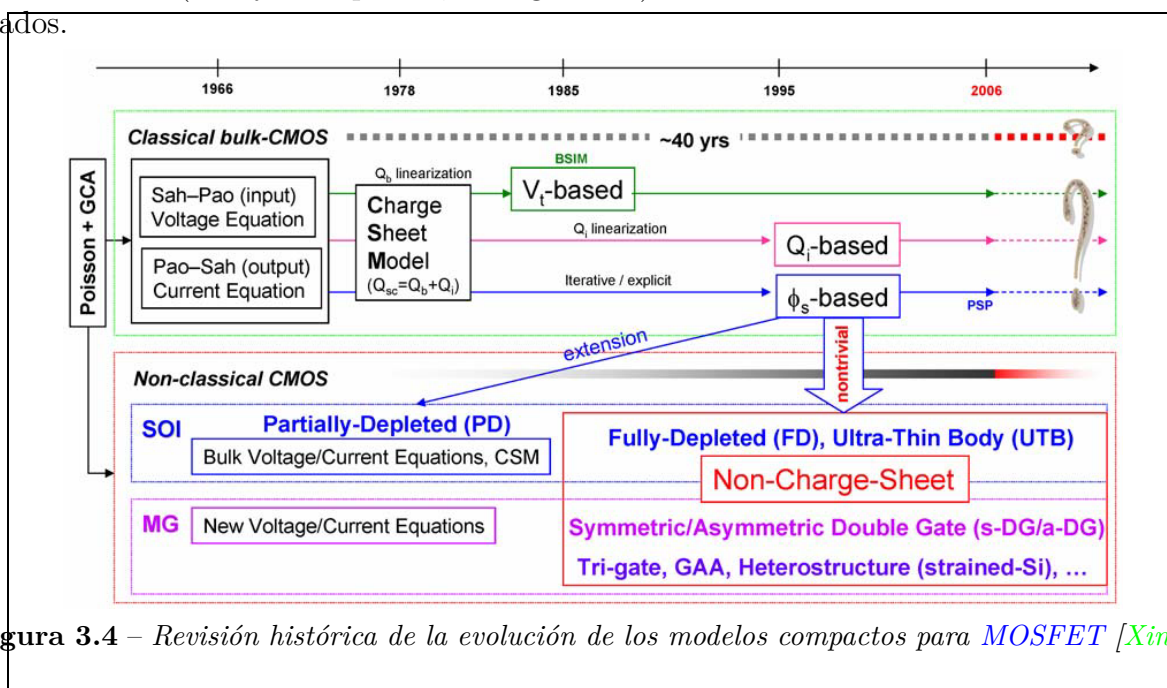


Figura 3.4 – Revisión histórica de la evolución de los modelos compactos para MOSFET [Xin04].

Aunque durante mucho tiempo han existido variados modelos compactos de libre distribución optimizados para aplicaciones concretas, con procedimientos de extracción de parámetros diferentes, se está produciendo una concentración de los equipos desarrolladores de modelos compactos para definir un estándar normalizado que permita unificar los procesos de caracterización, obtención de parámetros, diseño y simulación. Recientemente el grupo desarrollador del modelo BSIM⁸ de Berkeley ha propuesto el BSIM6⁸ para sustituir a los anteriores BSIM3⁸, BSIM4⁸ y PSP¹¹ y ser referencia en las nuevos procesos (*bulk CMOS*). Para aprovechar las ventajas del modelo basado en carga EKV⁹ el equipo de Berkeley lide-

⁹EKV: **E**nz **K**rummenacher **V**itoz en la EPFL

¹⁰ACM: **A**dvanced **C**ompact **M**odel

¹¹PSP: **P**hilips & **S**P model (Penn State University)

¹²HiSIM: **H**iroshima University **S**TARC **I**GFET **M**odel

rado por el profesor *Chenming Hu* y el grupo de modelado de la EPFL del EKV⁹, liderado por el profesor *Christian Enz* han decidido desarrollar conjuntamente el BSIM6. Dada la importancia de la decisión, suponemos que el BSIM6 se convertirá en el modelo de referencia en los próximos años. Previamente se había unido también el Semiconductor Technology Academic Research Center (STARC) y el laboratorio del profesor *Michiko Miura-Mattausch* (HiSIM) para lanzar el HiSIM3¹².

En la figura 3.5 se muestra la estructura genérica de un modelo MOSFET multipuerta [MCH⁺08]. Se aprecia el desarrollo de diferentes módulos que representan los efectos físicos que están presentes en un dispositivo según su tecnología de fabricación, tamaño, etc.

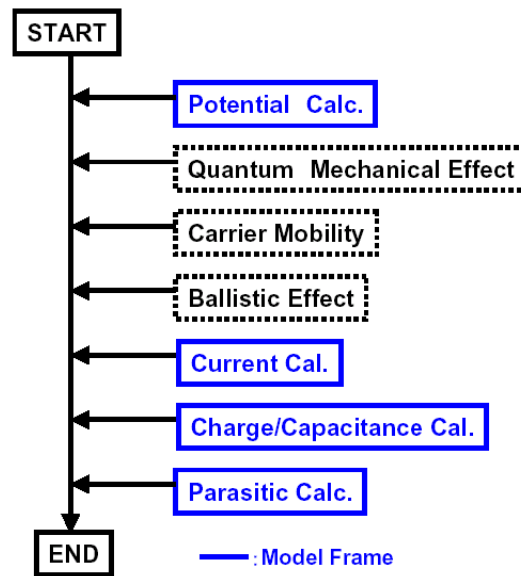


Figura 3.5 – Desarrollo de modelos de MOSFET multipuerta para simulación de circuitos [MCH⁺08].

Los diseñadores necesitan modelos en los que poder confiar y que describan correctamente numerosos fenómenos físicos. Deben permitir las simulaciones de circuitos integrados en tiempos razonables y facilitar datos que se ajusten a la realidad. Es recomendable que habiliten un análisis y diseño manual sencillo que permita al diseñador establecer las interrelaciones existentes entre los parámetros del circuito y los dispositivos para tomar las decisiones necesarias.

Las simulaciones digitales requieren además que el modelo describa el comportamiento dinámico que permita reproducir las transiciones de subida y bajada. Con la disminución de la tensión de alimentación deben introducirse más detalles, como la corriente residual de transistores en corte cuya importancia ha aumentado. El diseño de circuitos analógicos de altas prestaciones requiere además la exactitud en el modelo de pequeña señal, el ruido y el análisis armónico.

El modelado de dispositivos multipuerta tiene que encajar las necesidades generales que una aplicación electrónica requiere a un dispositivo general con las particularidades de estos dispositivos que por su pequeño tamaño presentan una alta influencia de los efectos

cuánticos, de los efectos de *overshoot* de la velocidad, etc. En este contexto de las actividades de modelado presentamos los modelos que se detallan a continuación enmarcadas en el contexto de la implementación en Verilog-A descrita en el primer capítulo de esta tesis.

3.2 Modelo de transistores SGT cilíndricos.

En este apartado se considera un dispositivo con una puerta que rodea completamente al canal de silicio cilíndrico, figura 3.6. Se denomina *Surrounding Gate Transistor*. Este dispositivo presenta un sustrato poco dopado ya que tiene, como se ha explicado anteriormente, un excelente control de los efectos de canal corto.

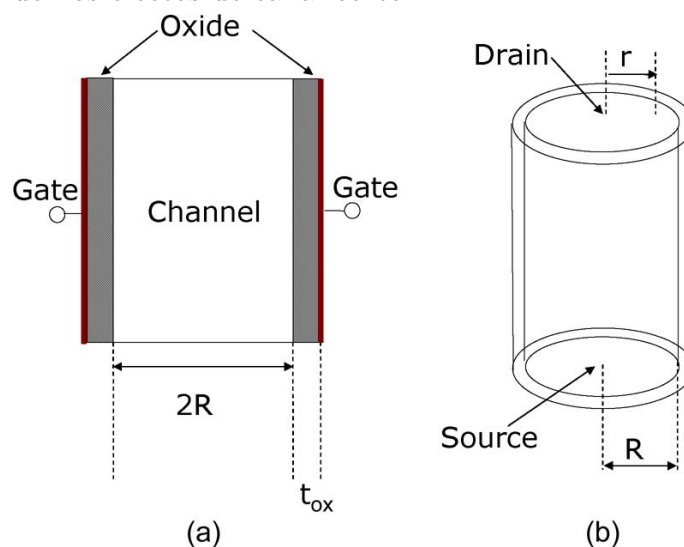


Figura 3.6 – Estructura simulada del dispositivo *SGT* (a), sección del dispositivo (b).

El modelo que vamos a utilizar es un modelo de control de carga que tiene en cuenta los efectos cuánticos debidos a: i) un grosor modificado de la capa de óxido que incluye la anchura finita de la carga de inversión dentro del sustrato, y ii) el aumento de la tensión umbral. El centroide de la distribución de carga en inversión se modela con una expresión empírica usada para modificar la expresión explícita clásica del modelo de control de carga [IJR⁺05, RGGB08], manteniendo sus ventajas iniciales. El modelo de control de carga ha sido validado con datos de simulación obtenidos por un simulador que resuelve auto-consistentemente las ecuaciones de Poisson and Schroedinger [GRGG⁺07] para un conjunto amplio de transistores *SGT* con distinta geometría en los que varía el grosor del óxido y el radio del cilindro de silicio.

El modelo de carga se ha usado para obtener la corriente de drenador que además de incluir los efectos cuánticos [TF05, RGGB08], incluyen los efectos de canal corto, el *overshoot* de la velocidad [RGLVC97], la modulación de la longitud del canal [LI08], la velocidad de saturación y un modelo detallado de movilidad de bajo campo [TFG04, RGJM⁺10]. Algunos datos experimentales como los de [SAB⁺06], han sido reproducidos correctamente con el modelo propuesto. Se han añadido los modelos de capacidad interna del dispositivo y se ha ultimado el modelo, junto con todo lo anterior, para abordar simulaciones circuitales tanto

DC como AC y de transitorios en el tiempo.

Finalmente, el modelo completo ha sido implementado en Verilog-A. Se han depurado los algoritmos iterativos para mejorar la convergencia y poder abordar la simulación de distintos tipos de circuitos basados en dispositivos SGT. Los resultados de simulación de circuitos basados en dispositivos SGT se presentan en la última sección del capítulo.

3.2.1 Efectos cuánticos en los Surrounding Gate Transistors.

Los mayoría de modelos de carga y corriente para transistores SGT no incluyen efectos cuánticos [IJR+05, JIS+04]. Sin embargo en el caso de los transistores SGT aun en el caso de poseer un radio mayor de 5 nm (límite a veces utilizado para delimitar la influencia de los efectos cuánticos estructurales), los efectos cuánticos son muy importantes si el grosor de la lámina de óxido de silicio T_{ox} es reducido a 1 nm o menos [RGJM+10].

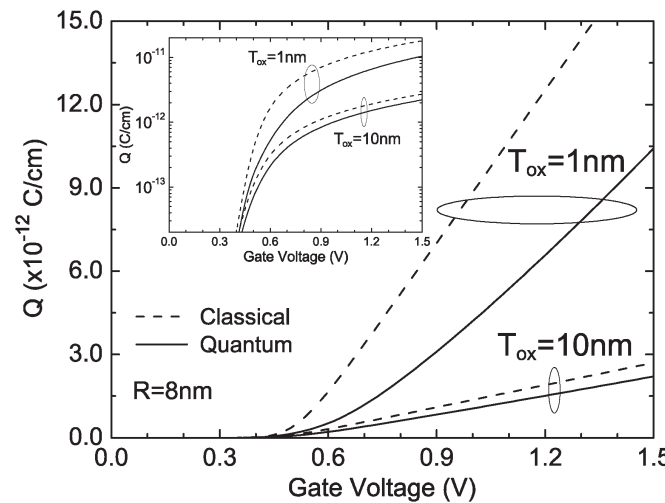
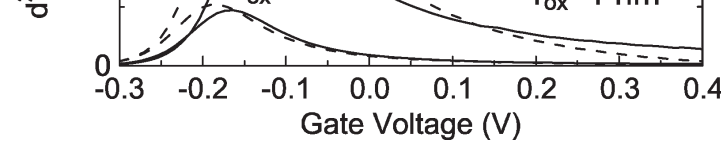
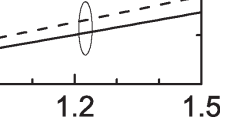


Figura 3.7 – Carga del canal por unidad de longitud para un transistor SGT con $R = 8$ nm a temperatura ambiente. Se han considerado dos grosores de óxido ($T_{ox} = 10$ nm y $T_{ox} = 1$ nm. El método clásico (cuántico) es representado por líneas discontinuas (sólidas).

La figura 3.7 compara la carga por unidad de longitud en el caso de un transistor SGT con $R = 8$ nm a temperatura ambiente, calculada usando el método clásico (línea discontinua) y la aproximación cuántica (línea sólida) para dos valores de grosor de óxido, $T_{ox} = 1$ nm y $T_{ox} = 10$ nm. Se ha utilizado un metal para el electrodo de puerta *midgap* con función trabajo similar al silicio intrínseco. La carga calculada se ha obtenido del simulador desarrollado en el Grupo de Investigación de Nanoelectrónica de la Universidad de Granada [GRGG+07].

Esto se pone de manifiesto en la figura 3.8, donde se muestra la capacidad puerta-canal (C_{gc}) en función de la tensión de puerta aplicada para un transistor de $R = 5$ nm con dos valores de grosor de óxido. Los resultados clásicos (cuánticos) se presentan en líneas discontinuas (sólidas). Para estos cálculos se ha supuesto una puerta metálica de aluminio. La capacidad de canal-puerta en el caso clásico y cuántico (C_{gc}), es muy similar para $T_{Ox} = 10$ nm, pero difiere cuando el grosor del óxido decrece. El valor del centroide de carga en



3.2. Modelo de transistores SGT cilíndricos.

with $R = 8$ nm at room
 $T_{ox} = 10$ nm and $T_{ox} =$
 in dashed (solid) line. A
 inversion es comparable a la inversión de los efectos cuánticos en este caso [RGJM+10]. De este modo, si queremos modificar el

Fig. 3. Second derivative of the charge as a function of the applied gate voltage for an SGT with $R = 5$ nm for two values of the oxide thickness. Classical (quantum) results are shown in dashed (solid) lines. The maximum inversion es comparable a la inversión de los efectos cuánticos en este caso [RGJM+10]. De este modo, si queremos modificar el

reproduced correctly
 main conclusions are

ROUNDING

y been published for
 mation, i.e., without
 ose models, quantum
 t that they were not
 nm. However, even
 ater than 5 nm, the
 f the oxide thickness

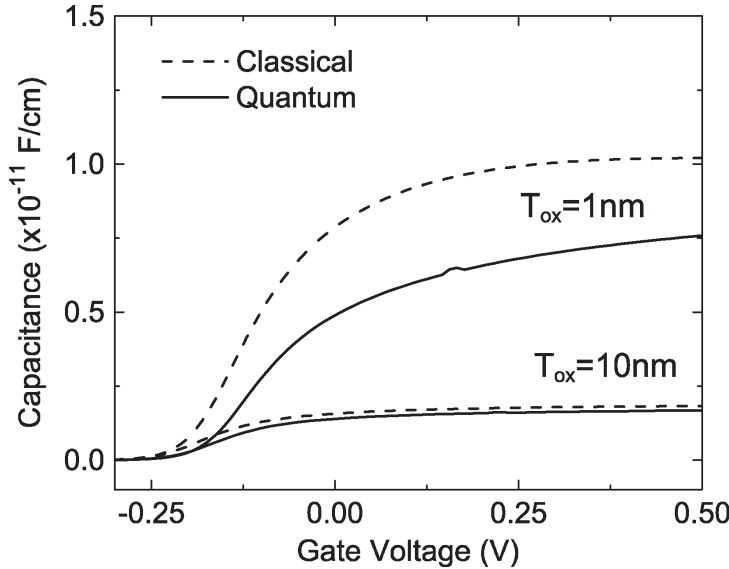


Figura 3.8 – Capacidad puerta-canal (C_{gc}) en función de la tensión de puerta aplicada para un transistor de $R = 5$ nm con dos valores de grosor de óxido. Los resultados clásicos (cuánticos) se presentan en líneas discontinuas (sólidas)

length in an SGT with
 l using (dashed line)
 quantum approach for
 m and $T_{ox} = 10$ nm
 s gate electrode for
 evaluated by using a
 n our research group
 es considered ($T_{ox} =$
 are above those for
 inner oxide are huge.
 er quantum effects in
 charge distribution in
 we can observe that
 ntum curves are due
 erence between quan-
 e almost constant in
 uantum and classical
 urve starting to rise
 ge. This means that
 ower than the quan-
 bserved in the case
 hreshold voltage is
 3 shows the second
 n of the applied gate

Fig. 4 Gate-to-channel capacitance C_{gc} as a function of the applied gate voltage for an SGT with $R = 5$ nm for two values of the oxide thickness. Classical (quantum) results are shown in dashed (solid) lines. For these calculations, an aluminum gate was assumed. The maximum of these curves gives the threshold voltage. As can be seen, the value of the quantum and classical threshold voltages has shifted slightly. We change here the threshold voltage difference by $\Delta V_{TH,QM}$, which is almost independent of the oxide thickness T_{ox} [LJR+09, LI08, CP07]. En nuestro caso, la carga del dispositivo SGT en su versión clásica viene dada por:

$$Q = C_{oxide} \left[\frac{2C_{gs} V_{TH}^2}{2C_{oxide} V_{TH}^2 + 4V_{TH}^2 \ln^2 \left(1 + \exp \left(\frac{V_{GS} - V_{TH}}{2V_{TH}} \right) \right)} \right] \quad (3.2.1)$$

3.2.2 Modelo de transistores SGT cilíndricos.
 En la bibliografía aparecen diferentes modelos de control de carga y de corriente relativos al dispositivo SGT [LJR+09, LI08, CP07]. En nuestro caso, la carga del dispositivo SGT en su versión clásica viene dada por:
 we can observe that
 ntum curves are due
 erence between quan-
 e almost constant in
 uantum and classical
 urve starting to rise
 ge. This means that
 ower than the quan-
 bserved in the case
 hreshold voltage is
 3 shows the second
 n of the applied gate

the quantum and classical threshold voltages has shifted slightly. We change here the threshold voltage difference by $\Delta V_{TH,QM}$, which is almost independent of the oxide thickness T_{ox} [LJR+09, LI08, CP07]. En nuestro caso, la carga del dispositivo SGT en su versión clásica viene dada por:
 parallel when $T_{ox} = 10$ nm, the slope of the two curves is very different when $T_{ox} = 1$ nm, which means that the gate-to-channel capacitance in the quantum and classical cases is also very different, with the quantum gate capacitance being lower than that in the classical case. This can be seen in Fig. 4, where gate-to-channel capacitance C_{gc} as a function of the applied gate voltage for an SGT with $R = 5$ nm is shown for two values of the oxide thickness. Classical (quantum) results are shown in dashed (solid) lines. For this calculation, an aluminum gate was assumed. The quantum and classical gate capacitances C_{gc} are very similar for $T_{ox} = 10$ nm but are quite different when the oxide thickness decreases. The reason for this



la carga Q_0 y de V_0 se detallan en la referencia [RGGB08, IJR+05]:

$$V_T = V_0 + 2V_{TH} \ln \left(1 + \frac{Q'}{Q_0} \right) \tag{3.2.2}$$

$$\Delta V_T = \frac{\left(\frac{2C_{oxide}V_{TH}^2}{Q_0} \right) Q'}{(Q_0 + Q')} \tag{3.2.3}$$

$$Q_0 = \left(\frac{4\epsilon_{si}}{R} \right) \left(\frac{kT}{q} \right) \tag{3.2.4}$$

$$V_0 = \phi_{MS} + \frac{kT}{q} \ln \left(\frac{8}{q\delta R^2} \right), \quad \delta = \frac{q^2 n_i}{kT \epsilon_{si}} \tag{3.2.5}$$

donde n_i representa la concentración intrínseca de portadores, ϕ_{MS} es la diferencia de la función trabajo is del metal-semiconductor y q es la carga del electrón. Q' es evaluada de la ecuación 3.2.1 asumiendo $V_T = V_0$ y $\Delta V_T = 0$.

3.2.2.1 Aumento de la tensión umbral.

La expresión utilizada para tener en cuenta la variación de la tensión umbral debida a la presencia de efectos cuánticos es la siguiente [RGGB08]:

$$\Delta V_{TH,QM} = 0.011V + \frac{1,3 \times 10^{-5}V \times cm^2}{R} \tag{3.2.6}$$

Existen otros modelos que calculan las correcciones cuánticas en la tensión umbral de los dis-

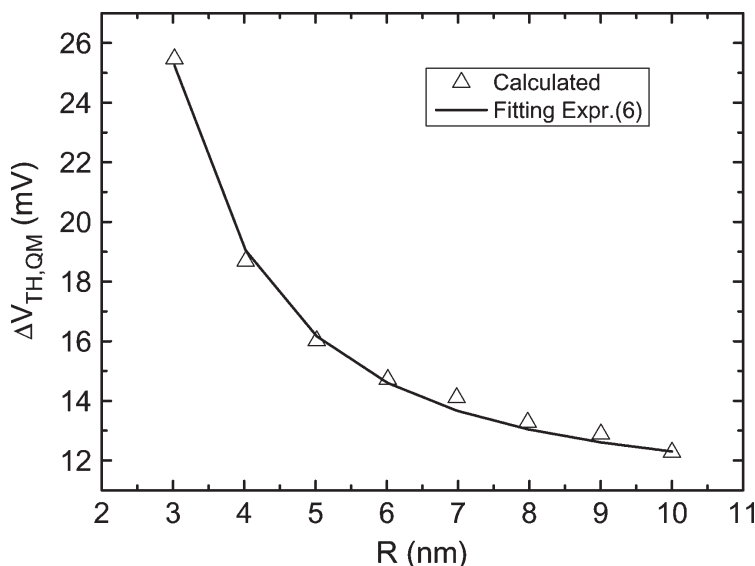


Figura 3.9 – $\Delta V_{TH,QM}$ calculado numéricamente y el ajuste proporcionado por la ecuación 3.2.6.

Fig. 7 shows the $\Delta V_{TH,QM}$ calculated numerically and the fitting provided by (6). There exist in the literature alternative analytical models

to compute the quantum corrections on the threshold voltage of SGT transistors obtained from physical basis [31]. Although these models have the advantage of catching the physics of the

la ventaja de incorporar la física del dispositivo, las expresiones son mucho más complejas que la mostrada en la ecuación 3.2.6.

3.2.2.2 Modelo cuántico de centroide.

En la aproximación clásica, aunque la carga está distribuida en el interior del silicio, el máximo de la carga en inversión está localizada justo en el interfaz del óxido y la capacidad de puerta-canal es muy similar a la capacidad del óxido C_{Oxide} . No obstante, en el caso cuántico, el gran desplazamiento de la carga dentro del cilindro de silicio hace que la capacidad puerta-canal total sea igual a la combinación serie de la capacidad asociada al óxido y la capacidad de la lámina de inversión. En nuestro caso se va a utilizar una capacidad de óxido modificada para dar cuenta de los efectos cuánticos en el cálculo de la carga en inversión y la diferente pendiente que aparece en las curvas (Q- V_G).

Para modelar el centroide de carga en inversión hemos utilizado la siguiente expresión empírica [RGGB08]:

$$\frac{1}{z_I} = \frac{1}{a + 2bR} + \frac{1}{z_{I0}} \left(\frac{N_I}{N_{I0}(R)} \right)^n \quad (3.2.7)$$

donde a , b , z_{I0} y n son constantes que no dependen de la polarización y N_I ($N_I = Q_I/q$) es la densidad de electrones por unidad de área. El modelo funciona bien para diferentes radios y rango de tensiones de puerta. Los parámetros utilizados para ajustar los datos experimentales son: $a = 0,55$ nm, $b = 0,198$, $z_{I0} = 5,1$ nm, $n = 0,75$, [RGGB08] y

$$N_{I0}(R) = 8,26 \times 10^{12} \text{ cm}^{-2} - 4,9 \times 10^{18} \text{ cm}^{-3} \times R(\text{ cm}) \quad (3.2.8)$$

La nueva capacidad de óxido (C_{oxide}^*) se obtiene del siguiente modelo:

$$\frac{1}{C_{oxide}^*} = \frac{1}{C_{oxide}} + \frac{1}{C_{centroid}} \quad (3.2.9)$$

donde C_{Oxide} es la capacidad de un condensador cilíndrico con radio externo igual a $(R + t_{ox})$ y radio interno R .

$$C_{oxide} = \frac{\varepsilon_{ox}}{R \ln \left(1 + \frac{t_{ox}}{R} \right)} \quad (3.2.10)$$

donde ε_{ox} es la permitividad del óxido. La capacidad del centroide $C_{centroid}$ es calculada como sigue:

$$C_{centroid} = \frac{\varepsilon_{si}}{(R - z_I) \ln \left(1 + \frac{z_I}{R - z_I} \right)} \quad (3.2.11)$$

donde ε_{si} es la permitividad del silicio.

3.2.2.3 Modelo de control de carga con efectos cuánticos.

En el cálculo de la carga en inversión incorporamos las expresiones 3.2.6 y 3.2.9 en la ecuación 3.2.1 que queda ahora como:

$$Q = C_{\text{oxide}}^* \left[-\frac{2C_{\text{oxide}}^* V_{\text{TH}}^2}{Q_0} + \sqrt{\left(\frac{2C_{\text{oxide}}^* V_{\text{TH}}^2}{Q_0}\right)^2 + 4V_{\text{TH}}^2 \ln^2 \left(1 + \exp\left(\frac{V_{\text{GS}} - V_{\text{TQM}} + \Delta V_{\text{TH,QM}} - V}{2V_{\text{TH}}}\right)\right)} \right] \quad (3.2.12)$$

donde

$$V_{\text{TQM}} = V_0 + 2V_{\text{TH}} \ln \left(1 + \frac{Q'}{Q_0}\right) + \Delta V_{\text{TH,QM}} \quad (3.2.13)$$

$$\Delta V_{\text{TH,QM}} = \frac{\left(\frac{2C_{\text{oxide}}^* V_{\text{TH}}^2}{Q_0}\right) Q'}{(Q_0 + Q')} \quad (3.2.14)$$

3.2.3 Modelo de corriente de drenador.

Una vez que se ha modelado la carga en el canal de un dispositivo SGT con un radio y grosor de óxido arbitrario, se puede obtener una expresión para la corriente de drenador. Siguiendo el trabajo desarrollado en [RGJM+10, RGLVC97] y partiendo de la siguiente ecuación obtenemos:

$$I_{\text{DS}} = \frac{2\pi R}{L} F(V_{\text{GS}}, V_{\text{DS}}) \left[\frac{\mu_{\text{eff}}}{\left(1 + \delta_0 \frac{\mu_{\text{eff}} V_{\text{DS}}}{v_{\text{sat}} L}\right)} + \frac{\lambda_a}{L} \right] \quad (3.2.15)$$

donde

$$F(V_{\text{GS}}, V_{\text{DS}}) = \int_0^{V_{\text{DS}}} Q(V) dV \quad (3.2.16)$$

L es la longitud de canal del transistor, μ_{eff} es la movilidad de bajo campo (detalles del modelo para este parámetro esencial en la referencia [TFG04]), v_{sat} es la velocidad de saturación (valor 10^7 cm/s), λ_a el parámetro del *overshoot* de la velocidad [RGLVC97] (el valor de λ_a usado en este trabajo es de 40×10^{-5} cm³/Vs). δ_0 es el parámetro introducido en la referencia [ARHR94] para mejorar la precisión del modelo de velocidad de los electrones en función del campo eléctrico longitudinal (E_{long} en la ecuación 3.2.17) [Tho80] cuando el parámetro β adopta el valor de 1 por simplicidad (β es un parámetro elegido normalmente entre 1 y 2. Asumimos el valor de $\beta = 1$, en línea con las aproximaciones realizadas por varios autores [ARHR94, GMCS07, RRGJM10]).

$$v(E_{\text{long}}) = \frac{\mu_{\text{eff}} E_{\text{long}}}{\left(1 + \left(\frac{\mu_{\text{eff}} E_{\text{long}}}{v_{\text{sat}}}\right)^\beta\right)^{\frac{1}{\beta}}} \quad (3.2.17)$$

La función $F(V_{GS}, V_{DS})$ representa la integral de la carga en inversión en el canal del dispositivo. Desarrollando se obtiene:

$$I_{DS} = \frac{2\pi R}{L} \left[\frac{\mu_{eff}}{\left(1 + \delta_0 \frac{\mu_{eff} V_{DS}}{v_{sat} L}\right)} + \frac{\lambda_a}{L} \right] \times \left[\frac{2kT}{q} (Q_s - Q_d) + \frac{Q_s^2 - Q_d^2}{2C_{Oxide}^*} + \frac{kT Q_0}{q} \ln \left(\frac{Q_d + Q_0}{Q_s + Q_0} \right) \right] \quad (3.2.18)$$

donde Q_s y Q_d son la carga en inversión calculada siguiendo la ecuación 3.2.12, en la fuente ($V = 0$) y en el drenador ($V = V_{DS}$) del dispositivo respectivamente.

Hemos introducido el modelo desarrollado por Trivedi et al. [TFG04] que permite tener en cuenta la contribución de mecanismos de dispersión por fonones y rugosidad superficial. Asumiendo $t_{Si} = 2R$ [RGJM+10], el modelo de movilidad usado presenta la siguiente expresión:

$$\mu_{eff} = \frac{U0}{1 + \frac{U0}{\mu_{ph(bulk)}} \left(\frac{\mu_{ph(bulk)}}{\mu_{ph}(t_{Si}(eff))} - 1 \right) + \theta \frac{U0}{\mu_{sr}}} \quad (3.2.19)$$

donde $U0$ y θ son parámetros de ajuste y el campo trasversal es calculado como en [LI08], usando el modelo de carga en inversión descrito anteriormente.

La modulación de la longitud de canal se introduce a través de la siguiente expresión [LI08, CCE06, Aro07]:

$$\Delta L = L_C \times \operatorname{arcsinh} \left(\frac{V_{DS} - V_{DSsat}}{E_{sat} L_C} \right) \quad (3.2.20)$$

donde ΔL es la longitud de la región de *pinch-off* cercana al drenador y L_C es calculada usando $L_C = a_{clh} \times \lambda_C$, con “ a_{clh} ” siendo el parámetro de ajuste con valor comprendido entre ($0 < a_{clh} < 1$) y λ_C la longitud característica¹³ del dispositivo SGT [LI08, Col04]. El valor escogido para a_{clh} fue 0,95.

El valor de la tensión de saturación, V_{DSsat} , se obtiene, como se ha detallado en [RGJM+10], estableciendo la continuidad de la corriente a lo largo de todo el canal, i.e., igualando la ecuación 3.2.18 con la corriente del canal saturado obtenida de:

$$I_{DS} = 2\pi R Q(V = V_{DSsat}) v_{sat} \quad (3.2.21)$$

Para incluir los efectos del *overshoot* de la velocidad en la ecuación 3.2.21, un término extra es necesario. Hemos seguido [RGLVC97], por lo tanto, un término proporcional al gradiente

¹³La longitud característica da información sobre la susceptibilidad de un dispositivo a sufrir efectos de canal corto. Representa la distancia de penetración de las líneas del campo eléctrico desde el drenador hacia el cuerpo del dispositivo o el control que la zona de drenador tiene en la zona de deplexión en el canal [Col04].

$$\lambda_C = \sqrt{\frac{2\epsilon_{si}(2R)^2 \ln\left(1 + \frac{2\epsilon_{ox}}{t_{si}}\right) + \epsilon_{ox} t_{si}^2}{16\epsilon_{ox}}} \text{ con } t_{si} = 2R.$$

del campo eléctrico longitudinal ($E_{long} = -dV(x)/dx$, con x siendo la coordenada que varía en la dirección fuente-drenador, $0 \leq x \leq L$, y $0 \leq V(x) \leq V_{DS}$) se añade a la velocidad de saturación. La inclusión de los efectos del *overshoot* de la velocidad nos conduce a la ecuación 3.2.22.

$$I_{DS} = 2\pi RQ(V = V_{DSsat}) \left(v_{sat} + \lambda_a \left\langle \frac{d^2V}{d^2x} \right\rangle \right) \quad (3.2.22)$$

donde el término de la expresión que aparece entre ángulos representa un valor medio del gradiente del campo eléctrico a lo largo del canal y puede ser aproximado por la siguiente expresión [RGLVC97]:

$$\left\langle \frac{d^2V}{d^2x} \right\rangle \approx b_{VO} \frac{V_{DS}}{L^2} \quad (3.2.23)$$

En este caso, donde b_{VO} es constante, hemos usado el valor $b_{VO} = 1$. El cálculo de V_{DSsat} determina la transición entre las regiones lineal y saturación. La obtención de V_{DSsat} requiere un algoritmo iterativo. Es interesante destacar que este algoritmo ha sido optimizado para minimizar el tiempo de cálculo, ya que en una simulación circuital son cientos de puntos de polarización los que se requieren para alcanzar la convergencia [RRGJM10]. En este último caso, la modulación de longitud de canal debe ser incluida también, i.e., la longitud de canal, L , debe ser sustituida por $(L - \Delta L)$. Para evitar tener que usar dos expresiones diferentes para la corriente de drenador (para la zona lineal y saturación) y poder hacer una transición suave entre ellas, hemos establecido una función de suavización (ecuación 3.2.24, con $A = 10$) según se describe en [Aro07].

$$V_{DSx} = F(V_{DS}, V_{DSsat}) \times V_{DSsat} \quad (3.2.24)$$

donde

$$F(V_{DS}, V_{DSsat}) = 1 - \frac{\ln \left[1 + e^{A \left(1 - \frac{V_{DS}}{V_{DSsat}} \right)} \right]}{\ln [1 + e^A]} \quad (3.2.25)$$

Para resumir, la expresión final de la corriente de drenador corresponde a:

$$I_{DS} = \frac{2\pi R}{(L - \Delta L)} \left[\frac{\mu_{eff}}{\left(1 + \delta_0 \frac{\mu_{eff} V_{DSx}}{v_{sat}(L - \Delta L)} \right)} + \frac{\lambda_a}{(L - \Delta L)} \right] \times \left[\frac{2kT}{q} (Q_s - Q_d) + \frac{Q_s^2 - Q_d^2}{2C_{Oxide}^*} + \frac{kTQ_0}{q} \ln \left(\frac{Q_d + Q_0}{Q_s + Q_0} \right) \right] \quad (3.2.26)$$

donde Q_d es obtenida mediante la ecuación 3.2.12 con $V = V_{DSx}$ y μ_{eff} es calculada a través de la ecuación 3.2.19 para los valores de tensión de puerta y drenador correspondientes.

Los resultados experimentales obtenidos por los dispositivos fabricados por Singh et al. [SAB+06] han sido reproducidos con este modelo.

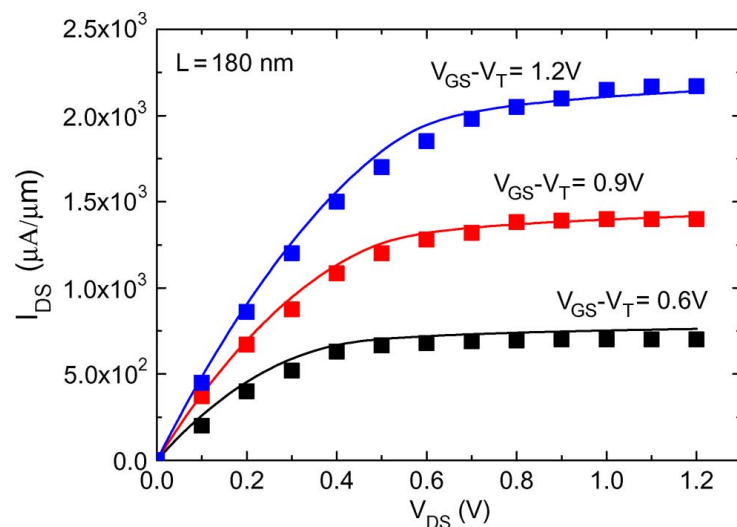


Figura 3.10 – Curvas de salida (normalizadas al diámetro del cilindro) para un dispositivo SGT con $R = 2.5$ nm, $T_{ox} = 9$ nm, $L = 180$ nm, y $V_T = 0.05$ V, y los datos experimentales presentados en la referencia [16]. Los símbolos representan los resultados experimentales y las líneas continuas los resultados obtenidos con nuestro modelo.

3.2.4 Modelado de las capacidades del SGT.

En una aplicación real, los dispositivos trabajan con tensiones en sus terminales que varían con el tiempo. Dependiendo de las características de la variación, el régimen de operación se clasifica en gran o pequeña señal. Si la variación de las tensiones es lo suficientemente pequeña, el dispositivo semiconductor puede modelarse mediante resistencias, condensadores, fuentes de corriente, etc. (modelo equivalente de pequeña señal). En este modelo se introducen los efectos capacitivos que son parte fundamental del modelo compacto del dispositivo.

Los modelos de capacidad a veces se plantean en base a la aproximación cuasi-estática que asume que todas las cargas del dispositivo pueden seguir las variaciones de las tensiones de los terminales inmediatamente y sin retraso [Bha09].

Sin pérdida de generalidad, las capacidades de los dispositivos semiconductores se dividen en dos tipos: intrínsecas y extrínsecas. Las capacidades intrínsecas caracterizan el comportamiento dinámico del dispositivo en la zona enmarcada por el oxido de puerta y drenador. Por otra parte, en las capacidades extrínsecas se asocian a las uniones PN en inversa fuente-sustrato y drenador-sustrato y en general las capacidades de *overlap* (o *underlap*, según el caso) y *fringe* entre puerta-fuente y puerta-drenador.

La hipótesis cuasiestática asume que si $v_D(t)$ son las variaciones temporales de la tensión de drenador, puerta y fuente; para cualquier instante (t'), las cargas en el dispositivo serían idénticas a las que se obtendría si las tensiones no dependieran del tiempo ($v_D(t')$, $v_{GS}(t')$, $v_{GS}(t')$). Algunos autores consideran que para aplicaciones de gate a la aproximación cuasiestática se puede utilizar en señales que cumplan que su frecuencia sea menor que el tiempo de tránsito de los portadores en el canal en condiciones de operación. El modelo aquí reproducido

simulated results obtained with a self-consistent 2-D Poisson and Schroedinger solver. Experimental results have also been reproduced correctly. The physically based approach permits a simulation of circuits with advanced models in Verilog-A. Simulación de circuitos con modelos avanzados en Verilog-A. coherent modeling of the main effects on the drain current of these transistors.

REP

[1] International Technology Roadmap for Semiconductors, Available: <http://public.itrs.net>

[2] G. Celler and S. Cristoloveanu, *Appl. Phys.*, vol. 93, no. 9, pp. 094301, 2003.

[3] A. Tsormpatzoglou, C. C. A. G. Ghibaudo, "Threshold voltage of double-gate MOSFETs," *IEEE Trans. Electron Devices*, no. 9, pp. 2512–2516, Sep. 2007.

[4] V. Trivedi and J. G. Fossum, "Threshold voltage of undoped double-gate MOSFETs," *IEEE Lett.*, vol. 26, no. 8, pp. 579–581, Aug. 2002.

[5] H. Kang, J. Han, and Y. Cho, "Threshold voltage of double-gate MOSFETs with undoped channel," *IEEE Lett.*, vol. 29, no. 8, pp. 927–929, Aug. 2005.

[6] V. Trivedi, J. G. Fossum, and S. Cristoloveanu, "Threshold voltage for nanoscale ultra-thin-body MOSFETs," *IEEE Lett.*, pp. 763–766, Aug. 2003.

[7] J. A. Lopez-Villanueva, M. R. Utrilla, and A. P. Nardelli, "Effects of channel doping on the threshold voltage of double-gate MOSFETs," *IEEE Lett.*, pp. 141–146, Jan. 2003.

[8] V. Trivedi, M. R. Reggiani, M. R. Utrilla, and S. Cristoloveanu, "Self-consistent solution for nanowire MOSFETs," in *Proc. IEEE Electron Device Meeting*, pp. 1–4, Dec. 2004.

[9] B. Iniguez, D. Jimenez, J. Roig, and S. Cristoloveanu, "Explicit continuous model for nanowire MOSFETs," *IEEE Trans. Electron Devices*, vol. 53, no. 8, pp. 3601–3607, Aug. 2006.

[10] B. Iniguez, B. Iniguez, J. S. Roig, and S. Cristoloveanu, "Continuous analytical model for double-gate MOSFETs," *IEEE Electron Lett.*, vol. 40, no. 16, pp. 1000–1001, Aug. 2004.

[11] B. Iniguez, D. Jimenez, and S. Cristoloveanu, "Explicit continuous model for nanowire MOSFETs," *Semicond. Sci. Technol.*, vol. 23, no. 5, pp. 054001, May 2008.

[12] H. Cho and J. D. Plummer, "Threshold voltage of MOSFETs with bulk trap states," *IEEE Electron Lett.*, vol. 33, no. 1, pp. 60–62, Jan. 2007.

[13] H. Cho and J. D. Plummer, "Threshold voltage of MOSFETs with depleted, surrounding-gate MOSFETs," *IEEE Electron Lett.*, vol. 18, no. 2, pp. 166–169, Feb. 2002.

[14] J. B. Roldan, A. Godoy, F. Gamiz, and J. A. Lopez-Villanueva, "Modeling of the inversion charge in nanowire MOSFETs including quantum effects," *IEEE Electron Lett.*, vol. 44, no. 1, pp. 41–43, Jan. 2008.

[15] J. B. Roldan, F. Gamiz, J. A. Lopez-Villanueva, and J. A. Lopez-Villanueva, "Modeling effects of electron transport in nanowire MOSFETs," *IEEE Trans. Electron Devices*, vol. 55, no. 1, pp. 1–8, Jan. 2008.

[16] J. B. Roldan, F. Gamiz, J. A. Lopez-Villanueva, and J. A. Lopez-Villanueva, "Deep submicrometer SOI MOSFETs: Modeling of self-heating and velocity overshoot effects," *IEEE Device Lett.*, vol. 21, no. 5, pp. 579–581, May 2000.

[17] F. G. Ruiz, A. Godoy, F. Gamiz, and J. A. Lopez-Villanueva, "Intrinsic velocity overshoot effects in the corner of nanowire MOSFETs," *IEEE Trans. Electron Devices*, vol. 55, no. 12, pp. 3377–3380, Dec. 2007.

[18] N. Singh, A. Agarwal, L. K. E. Lau, and G. Q. Lo, "Ballistic transport in fully depleted silicon nanowire MOSFETs," *IEEE Electron Lett.*, vol. 34, no. 1, pp. 383–386, May 2006.

[19] J. A. Lopez-Villanueva, P. C. P. and S. Rodriguez, "Effects of velocity overshoot on the behavior," *IEEE Trans. Electron Devices*, vol. 46, no. 11, pp. 99–101, Nov. 1997.

[20] N. Rodriguez, F. Gamiz, and J. A. Lopez-Villanueva, "Intrinsic layer centroid and polysilicon MOSFET behaviour: The influence of the channel doping," *IEEE Trans. Electron Devices*, vol. 55, no. 1, pp. 1–8, Jan. 2008.

[21] G. Baccarani and M. R. Utrilla, "Modeling of the inversion charge in nanowire MOSFETs," *IEEE Trans. Electron Devices*, vol. 55, no. 1, pp. 1–8, Jan. 2008.

Las intrínsecas son más complejas que las extrínsecas. En las primeras etapas del modelado compacto de los dispositivos MOSFET se utilizó el modelo Meyer [Mey71] que configuraba al modelo de capacidad del MOSFET como tres capacidades concentradas puerta-fuente C_{gs} , puerta-drenador C_{gd} y puerta-sustrato C_{gb} . Presentaba problemas de no conservación de la carga [PEC83]; sin embargo, este modelo ha sido usado ampliamente en los simuladores de circuitos a lo largo de muchas décadas. Posteriormente se han utilizado otros, como el que se presenta más adelante para los SGTs.

Un dispositivo intrínseco SGT es básicamente un elemento de tres terminales (no vamos a considerar el contacto del sustrato) en el que se puede aplicar un modelo capacitivo de alta frecuencia como el que se presenta en el apartado siguiente [JWY+07].

3.2.4.1 Modelado en alta frecuencia del dispositivo intrínseco.

El modelo de alta frecuencia que hemos planteado se fundamenta en una extensión del modelo cuasiestático (modelo cuasiestático completo) con lo que se aumenta el rango válido de frecuencias de uso, pero considerando únicamente la parte intrínseca del dispositivo y sin ionización de impacto. Este enfoque más generalista permite la obtención de modelos válidos incluso en el rango de RF y razonablemente adecuadas para aplicaciones digitales. La denominación de *modelo completo* parte de que estudia los efectos capacitivos de todos los terminales en todos los terminales restantes [REMB80, Tsi03].

Con el fin de mantener la generalidad en las deducciones consideramos el terminal de sustrato o *bulk*. De este modo parte de los resultados pueden ser extensibles a otros dispositivos multipuerta con contacto de sustrato (a pesar de que no vamos a utilizar el terminal de sustrato en nuestro modelado de capacidades del SGT, lo incluimos aquí para extender la generalidad del planteamiento). En la figura 3.11 se presenta el dispositivo MOSFET ge-

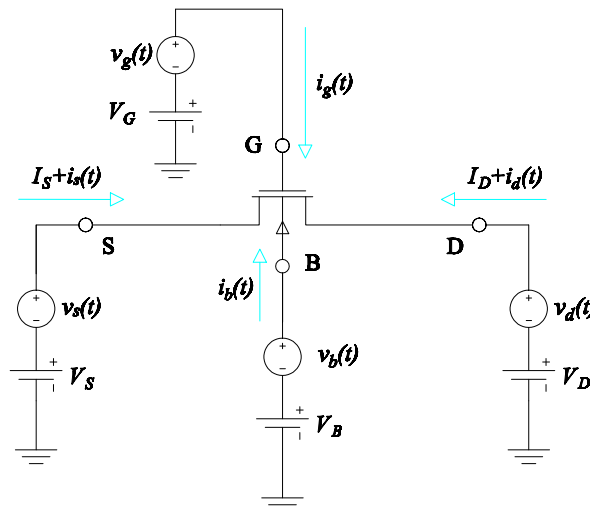


Figura 3.11 – Polarización en DC+AC

nérico polarizado por las tensiones V_S , V_G , V_B y V_D sobre las que se superpone una tensión de pequeña señal $v_s(t)$, $v_g(t)$, $v_b(t)$ y $v_d(t)$. Las corrientes definidas positivas entrantes al

dispositivo presentan un valor de polarización y otro de pequeña señal representados por $i_S(t) = I_S + i_s(t)$, $i_G(t) = I_G + i_g(t)$, $i_B(t) = I_B + i_b(t)$ y $i_D(t) = I_D + i_d(t)$. Las corrientes de carga asociadas a los terminales deben verificar (3.2.27) y sus valores vienen definidos por (3.2.28)¹⁵.

$$i_D(t) + i_G(t) + i_B(t) + i_S(t) = 0 \quad (3.2.27)$$

$$i_D(t) = \frac{\partial q_D}{\partial v_D} \frac{dv_D}{dt} + \frac{\partial q_D}{\partial v_G} \frac{dv_G}{dt} + \frac{\partial q_D}{\partial v_B} \frac{dv_B}{dt} + \frac{\partial q_D}{\partial v_S} \frac{dv_S}{dt} \quad (3.2.28a)$$

$$i_G(t) = \frac{\partial q_G}{\partial v_D} \frac{dv_D}{dt} + \frac{\partial q_G}{\partial v_G} \frac{dv_G}{dt} + \frac{\partial q_G}{\partial v_B} \frac{dv_B}{dt} + \frac{\partial q_G}{\partial v_S} \frac{dv_S}{dt} \quad (3.2.28b)$$

$$i_B(t) = \frac{\partial q_B}{\partial v_D} \frac{dv_D}{dt} + \frac{\partial q_B}{\partial v_G} \frac{dv_G}{dt} + \frac{\partial q_B}{\partial v_B} \frac{dv_B}{dt} + \frac{\partial q_B}{\partial v_S} \frac{dv_S}{dt} \quad (3.2.28c)$$

$$i_S(t) = \frac{\partial q_S}{\partial v_D} \frac{dv_D}{dt} + \frac{\partial q_S}{\partial v_G} \frac{dv_G}{dt} + \frac{\partial q_S}{\partial v_B} \frac{dv_B}{dt} + \frac{\partial q_S}{\partial v_S} \frac{dv_S}{dt} \quad (3.2.28d)$$

Si no existe señal variable aplicada en los terminales, $v_s(t) = v_g(t) = v_b(t) = v_d(t) = 0$, el valor de las derivadas será también nulo y consecuentemente las ecuaciones (3.2.28) también lo serán. Si las variaciones de pequeña señal verifican la hipótesis cuasiestática y son de pequeña magnitud, las corrientes de carga $i_d(t)$, $i_g(t)$, $i_b(t)$ y $i_s(t)$ también serán pequeñas. En estas circunstancias de variaciones de pequeña amplitud, se puede calcular $\frac{\partial q_k}{\partial v_j}$ donde q_k representa una de las cuatro cargas y v_j cualquiera de las tensiones de los terminales. Se definen las capacidades del siguiente modo [Tsi03]:

$$C_{kk} = \left. \frac{\partial q_k}{\partial v_k} \right|_{v_d=V_D, v_g=V_G, v_b=V_B, v_s=V_S} \quad (3.2.29)$$

$$C_{kj} = \left. -\frac{\partial q_k}{\partial v_j} \right|_{v_d=V_D, v_g=V_G, v_b=V_B, v_s=V_S}, \quad k \neq j \quad (3.2.30)$$

Usando las definiciones establecidas en (3.2.29) y (3.2.30) se podrían reescribir las ecuaciones (3.2.28) como (3.2.31), usando las capacidades equivalentes cuyo valor no es asociable a ningún condensador real.

$$i_d(t) = +C_{dd} \frac{dv_d}{dt} - C_{dg} \frac{dv_g}{dt} - C_{db} \frac{dv_b}{dt} - C_{ds} \frac{dv_s}{dt} \quad (3.2.31a)$$

$$i_g(t) = -C_{gd} \frac{dv_d}{dt} + C_{gg} \frac{dv_g}{dt} - C_{gb} \frac{dv_b}{dt} - C_{gs} \frac{dv_s}{dt} \quad (3.2.31b)$$

$$i_b(t) = -C_{bd} \frac{dv_d}{dt} - C_{bg} \frac{dv_g}{dt} + C_{bb} \frac{dv_b}{dt} - C_{bs} \frac{dv_s}{dt} \quad (3.2.31c)$$

¹⁵Se utiliza la misma notación para tensiones $v_D(t) = V_D + v_d(t)$, y también para resto de terminales.

$$i_s(t) = -C_{sd} \frac{dv_d}{dt} - C_{sg} \frac{dv_g}{dt} - C_{sb} \frac{dv_b}{dt} + C_{ss} \frac{dv_s}{dt} \quad (3.2.31d)$$

Se pueden obtener bastantes relaciones entre estas capacidades que simplifican el problema expresado en la ecuación 3.2.31. Una vez simplificado se puede particularizar para obtener las expresiones específicas del SGT. Todo esto se desglosa con detalle en el Anexo 7.6.

3.2.4.2 Obtención capacidades en el SGT

Con el planteamiento anterior se puede desarrollar el cálculo de las capacidades para el dispositivo que estamos considerando siguiendo el trabajo de [MIJR07]. En lo que respecta a las capacidades de puerta, las capacidades intrínsecas de puerta tendrán un valor positivo la capacidad C_{jj} y siempre tendrán valor negativo las capacidades C_{jk} .

$$C_{gd} = -\frac{dq_g}{dv_s} \quad (3.2.32a)$$

$$C_{gs} = -\frac{dq_g}{dv_s} \quad (3.2.32b)$$

donde q_g representa la carga en la puerta del dispositivo SGT [MIJR07].

En la gráfica 3.12 se muestran las cargas de capacidad de puerta normalizadas con el valor $2\pi RLC_{ox}$ para un transistor SGT con una puerta de metal *mid-gap* con los siguientes parámetros a temperatura ambiente: $N_{i0} = 1,45 \times 10^{10} \text{ cm}^{-3}$, $t_{ox} = 1,5 \text{ nm}$, longitud efectiva del canal $L=1 \text{ }\mu\text{m}$, radio $R = 6,25 \text{ nm}$, movilidad $\mu_0 = 290 \text{ cm}^2/\text{Vs}$.

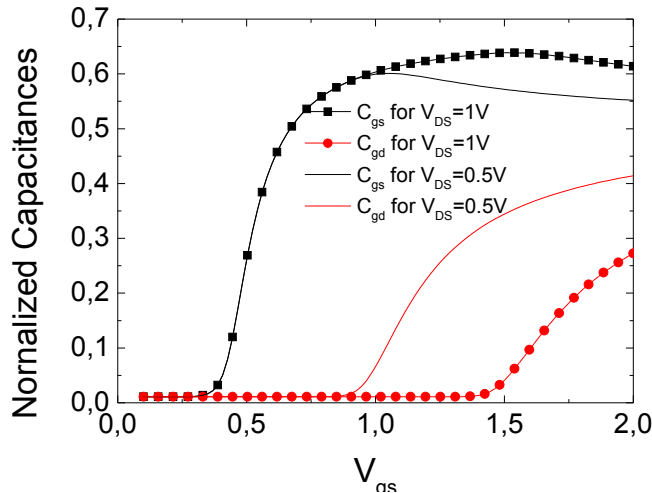


Figura 3.12 – Capacidades de puerta C_{gd} y C_{gs} .

En el cálculo de las capacidades de drenador y fuente se utilizan unas expresiones analíticas para la carga total de drenador q_D y fuente q_S que no representan cargas reales sino ficticias, obtenidas siguiendo el procedimiento *Ward's channel charge partitioning scheme*

[WD78], ecuación 3.2.33.

$$q_D = -2\pi R \int_0^L \frac{x}{L} Q dx \quad (3.2.33)$$

$$C_{dg} = -\frac{dq_d}{dv_g} \quad (3.2.34a)$$

$$C_{ds} = -\frac{dq_d}{dv_s} \quad (3.2.34b)$$

En la figura 3.13 se muestran los valores para las capacidades normalizadas con respecto al valor $(2\pi RLC_{ox})$ de drenador C_{dg} y C_{ds} para un dispositivo SGT similar al anterior.

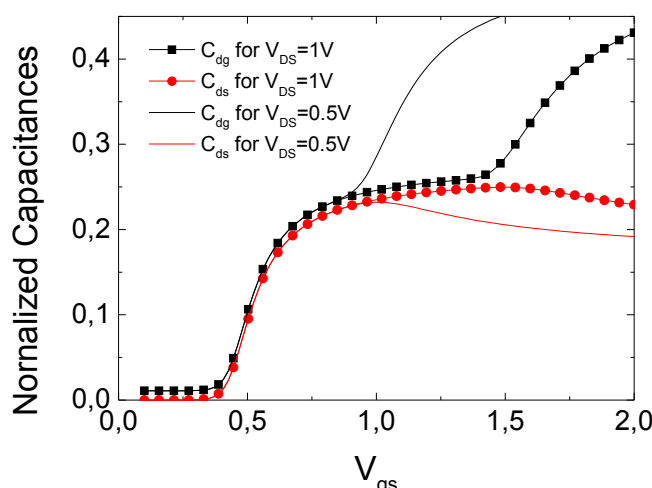


Figura 3.13 – Capacidades de puerta C_{dg} y C_{ds} .

Por otro lado, se verifica que la carga ficticia de fuente cumple la relación 3.2.35 a partir de lo expuesto en [SYWD80], ecuación 3.2.33.

$$q_s = q_{Tot} - q_d \quad (3.2.35)$$

Ahora se pueden calcular las capacidades restantes de fuente C_{sg} y C_{sd} :

$$C_{sd} = -\frac{dq_s}{dv_d} \quad (3.2.36a)$$

$$C_{sg} = -\frac{dq_s}{dv_g} \quad (3.2.36b)$$

En la gráfica 3.14 se muestran las cargas de capacidad de fuente normalizadas con el valor $2\pi RLC_{ox}$ para un transistor SGT con los mismos datos tecnológicos introducidos anteriormente.

El modelo de capacidad anterior está basado en una descripción dependiente de las cargas. Una vez determinadas las cargas en el drenador, puerta y fuente se procede al cálculo

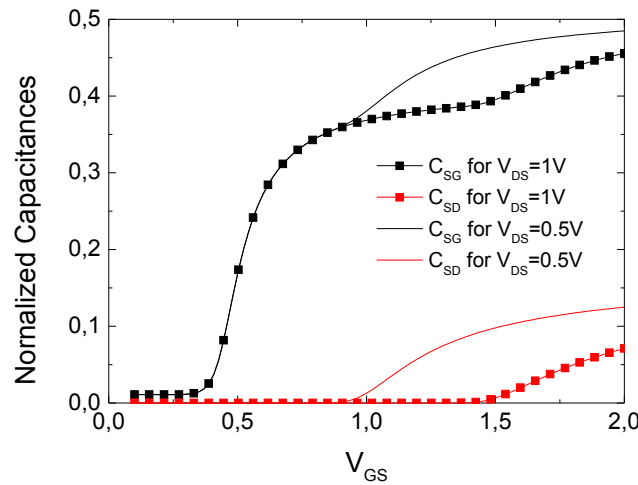


Figura 3.14 – Capacidades de puerta C_{sd} y C_{sg} .

de las capacidades mediante diferenciación matemática de la carga respecto a la tensión. Este modelo basado en carga cumple la conservación de carga al verificarse que $q_g + q_d + q_s = 0$. Se verifica así que los elementos de la matriz de capacidades en cada columna y en cada fila suman cero (ecuación 3.2.37) debido a las restricciones impuestas por la conservación de carga que son similares a las establecidas por la ley de Kirchhoff de corrientes.

$$C = \begin{bmatrix} C_{gg} & C_{gs} & C_{gd} \\ C_{sg} & C_{ss} & C_{sd} \\ C_{dg} & C_{ds} & C_{dd} \end{bmatrix} \implies \sum_i C_{ij} = \sum_j C_{ij} = 0 \quad (3.2.37)$$

3.2.4.2.1 Implementación de las capacidades.

Haciendo uso de la expresión 7.6.7, una vez conocidas todas las capacidades intrínsecas requeridas en la expresión (7.6.4), se procede a su incorporación en el modelo global del SGT con objeto de simular el comportamiento transitorio de circuitos digitales, entre ellos un anillo oscilador formado por transistores SGT de tipo N y P encuadrados en puertas CMOS¹⁶. En el listado 3.1 se presentan las contribuciones de corriente que cada capacidad introduce sobre las corrientes de los tres terminales. Estas contribuciones se incorporan tras la definición de la corriente estática (I_{DS}) establecida sobre los terminales drenador-fuente¹⁷.

```

1 | ...
2 | // Add charges
3 | I(d,s) <+ Cdd*ddt(V(d,s))-Cdg*ddt(V(g,s));
4 | I(g,s) <+ -Cgd*ddt(V(d,s))+Cgg*ddt(V(g,s));
5 | ...

```

Listado 3.1 – Listado de incorporación de capacidades en el modelo de SGT.

¹⁶En la capacidad externa añadida a cada puerta CMOS se ha incluido la contribución de las capacidades extrínsecas de la estructura.

¹⁷En este ejemplo puede observarse la potencia del lenguaje Verilog-A para introducir de una manera sencilla la contribución de señales dinámicas en la respuesta del transistor.

3.2.5 Simulaciones de circuitos con dispositivos SGT

En la figura 3.15 se muestra la hoja de simulación de ADS[®] utilizada para realizar el *matching*¹⁸ de los transistores SGT N-MOS y P-MOS que serán usados posteriormente para configurar los inversores CMOS. La tensión de drenador-fuente (V_{DS}) se ha barrido en el intervalo 0-1.2 V para tres tensiones de puerta (V_{GS}) (0.5, 0.85, 1.2 V). Los datos de corriente (I_{DS}) obtenidos siguiendo el procedimiento establecido en el apartado 3.2.3 se presentan en la figura 3.16 en color azul para el dispositivo N-MOS y en rojo para el P-MOS. El comportamiento diferente de la movilidad y de los efectos de *overshoot* de la velocidad en los dispositivos N y P motiva que las curvas en la zona lineal sean ligeramente diferentes.

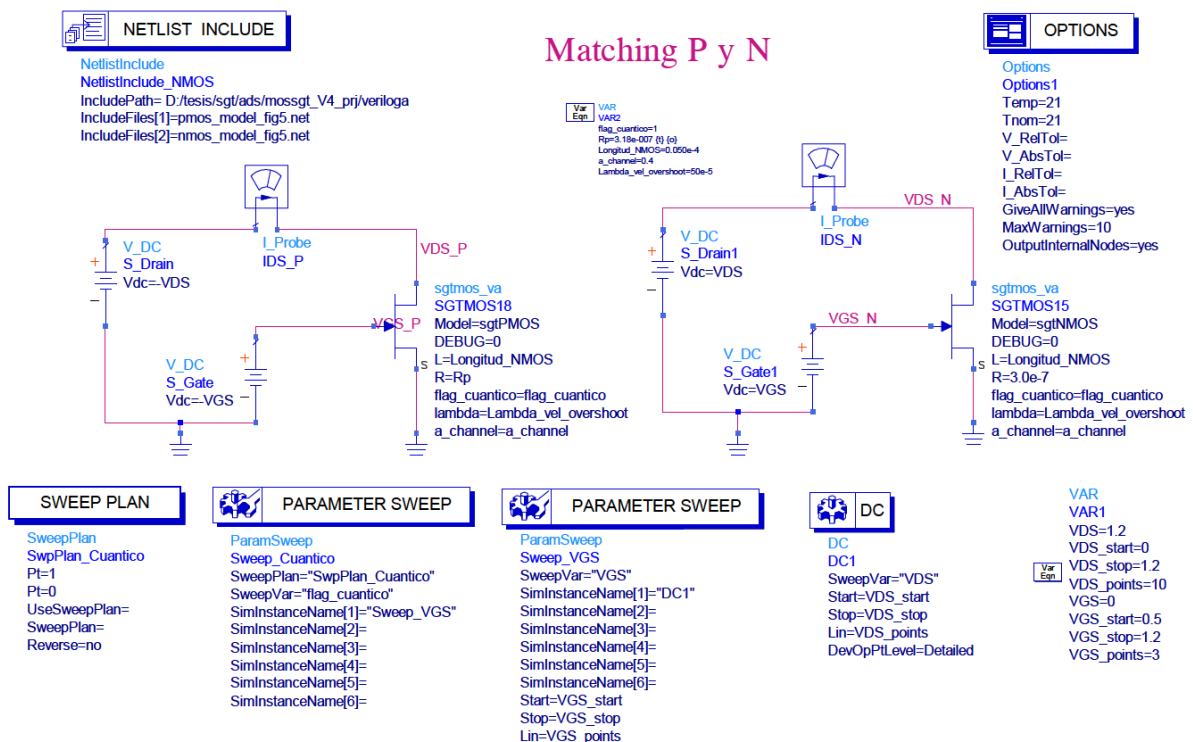


Figura 3.15 – Hoja de simulación en ADS[®] para realizar el *matching*¹⁸ de los dispositivos SGT N y P para $L = 50$ nm y $R = 3$ nm.

En el listado 3.2 se muestran los parámetros utilizados para realizar las simulaciones en el dispositivo N-MOS. Las unidades con las que se describe la dimensionalidad de cada parámetro acompañan al valor establecido en la tecnología que se simula. Además se incluyen *flags* que permiten activar o desactivar el modelo cuántico en el cálculo de la corriente I_{DS} , o permiten el volcado de la información interna de depuración del modelo durante las iteraciones del proceso de simulación.

```

1 | model sgtNMOS sgtmos_va \
2 | DEBUG=0 \ [0: sin información, 1: volcado de información de depuración]
3 | L= 50.0e-7 \ [cm] longitud del canal
4 | R = 3.0e-7 \ [cm] radio del cilindro de silicio
5 | tox = 1.5e-7 \ [cm] grosor de la lámina de silicio
6 | TYPE= 1 \ tipo [1: NMOS, -1: PMOS]
7 | na = 1E9 \ dopado del sustrato [cm^-3]
  
```

¹⁸También se utiliza la expresión “*sizing the transistors*” [SS98].

```

8 T0 = 300           \ temperatura de referencia [K]
9 t = 300           \ temperatura del dispositivo [K]
10 eg = 1.12         \ energía del band-gap [eV]
11 fim = 4.05        \ Función de trabajo del metal [eV]
12 afinidadsi = 4.05 \ Afinidad del silicio [eV]
13 ml = 8.9278E-31   \ Masa efectiva longitudinal dle electrón [kg]
14 nn = 0.75         \ Parámetro de modelado cuántico []
15 a = 5.5e-8        \ Parámetro de modelado cuántico [cm]
16 b = 0.198         \ Parámetro de modelado cuántico []
17 zi0 = 5.1e-7      \ Parámetro de modelado cuántico []
18 zi0_inc = 1       \ Para ajuste del centroide de P y N
19 flag_cuántico = 1 \ Alteración cuántica activa [0:NO, 1:SI]
20 eta = 1           \ Movilidad por Rugosidad y Fonones []
21 beta = 2.5e14     \ Movilidad por Rugosidad y Fonones [V/s]
22 tsi_ref = 14e-7   \ Movilidad por Rugosidad y Fonones [cm]
23 alfa = 4          \ Movilidad por Rugosidad y Fonones []
24 mu_ph_bulk = 1350 \ Movilidad por Rugosidad y Fonones [cm^2/Vs]
25 mu_min = 480      \ Movilidad por Rugosidad y Fonones [cm^2/Vs]
26 u0 = 900          \ Movilidad por Rugosidad y Fonones [cm^2/Vs]
27 theta = 0.83      \ Movilidad por Rugosidad y Fonones []
28 lambda = 50e-5    \ Movilidad por Rugosidad y Fonones [cm3/(Vs)]
29 reduccion = 0.37  \ Movilidad por Rugosidad y Fonones []
30 b_sat = 1         \ Corriente Saturación []
31 vsat = 1e7        \ Corriente Saturación [cm/s]
32 a_channel = 0.4   \ Modelo Modulación Longitud de Canal []
33 kmax = 800        \ Algoritmo Vds_sat [iteraciones]
34 a_vdsx = 5        \ Algoritmo Vds_sat []
    
```

Listado 3.2 – Listado de parámetros del dispositivo *SGT N-MOS* de $L=50$ nm.

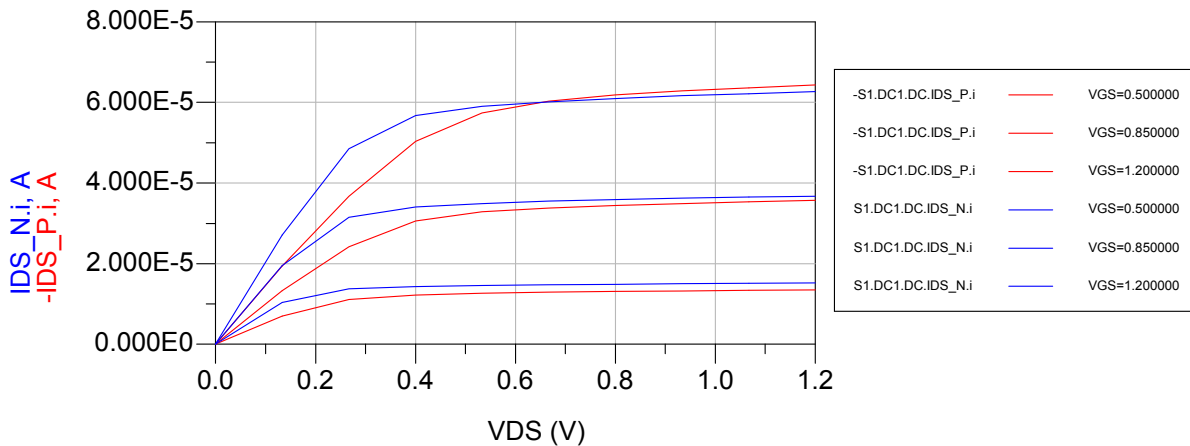


Figura 3.16 – Corriente I_{DS} obtenida de la simulación de la figura 3.15 mediante *ADS*[®] para los dispositivos *SGT N-MOS* (línea azul) y *P-MOS* (línea roja), descritos en el listado 3.2.

En la figura 3.17 se muestran los resultados de la simulación de la corriente I_{DS} para el caso cuántico y clásico de dispositivos *SGT N-MOS* (símbolo círculo) y *P-MOS* (símbolo triángulo) descritos en el listado 3.2 con el modelado clásico y cuántico.

En la figura 3.18 se muestran los resultados de dispositivos con $L = 30$ nm para la corriente I_{DS} para *SGT N-MOS* (línea azul) y *P-MOS* (línea roja) (los efectos cuánticos se incluyen). Los resultados obtenidos son similares a los anteriores y coinciden con los presentados en [IJR⁺05] si se desactivan los efectos cuánticos, de canal corto y de overshoot de la velocidad [RRG08a, RGJM⁺10].

Una vez se dispone de un par de transistores *SGT N-MOS* y *P-MOS* de baja concentración de dopado ($N_A = 10^{14}$ cm⁻³), de dimensiones $L = 50$ nm, $R_N = 3$ nm, $R_P = 3,18$ nm y $t_{ox} = 1,5$ nm y puerta de metal *midgap*, se puede confeccionar un inversor *CMOS* equilibrado. Para este circuitos hemos estudiado el comportamiento completo del modelo compacto simulando la evolución temporal de una conmutación digital. En la figura 3.19 se muestra la hoja de simulación de *ADS*[®] correspondiente para un inversor *CMOS* cargado con una ca-

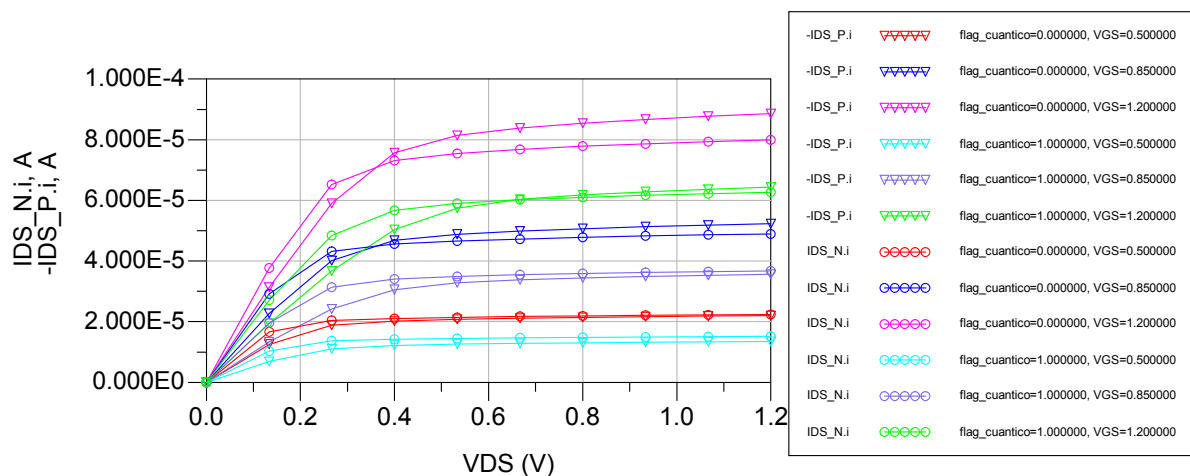


Figura 3.17 – Corriente I_{DS} obtenida de la simulación de la figura 3.15 mediante ADS[®] para los dispositivos SGT N-MOS (símbolo círculo) y P-MOS (símbolo triángulo) descritos en el listado 3.2 con el modelado clásico y cuántico.

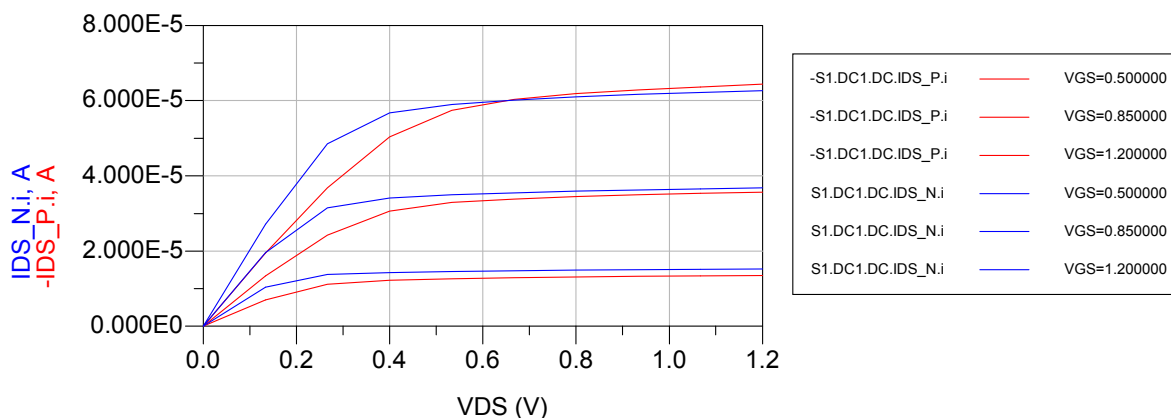


Figura 3.18 – Corriente I_{DS} obtenida de la simulación de la figura 3.15 mediante ADS[®] para los dispositivos SGT N-MOS (símbolo círculo) y P-MOS (símbolo triángulo) descritos en el listado 3.2 con el modelado clásico y cuántico para dispositivos de $L = 30$ nm.

pacidad de salida de 0.3 fF [RRGJM10, RRG08a, RRG09]. Hemos seleccionado los circuitos donde es posible estudiar la influencia de los efectos cuánticos y de *velocity overshoot*.

En la figura 3.20 se muestran los resultados de la tensión de salida del inversor CMOS frente a una tensión de entrada cuadrada incluyendo los efectos cuánticos (línea discontinua) y obviándolos (línea continua). Los efectos del *overshoot* de la velocidad se han tenido en cuenta en ambas simulaciones con un valor $\lambda_a = 40 \times 10^{-5} \text{ cm}^3/\text{Vs}$ [RRGJM10, RRG08a, RRG09].

Hemos estudiado también un anillo oscilador haciendo uso de siete etapas de inversores CMOS con capacidades de carga $C_L = 0,3$ fF.

La tensión de salida del anillo oscilador se muestra en la figura 3.22 donde se ha introducido la variación del parámetro del *overshoot* de la velocidad de $\lambda_a = 40 \times 10^{-5} \text{ cm}^3/\text{Vs}$.

Como se observa, existe una dependencia importante del valor de la tensión de salida

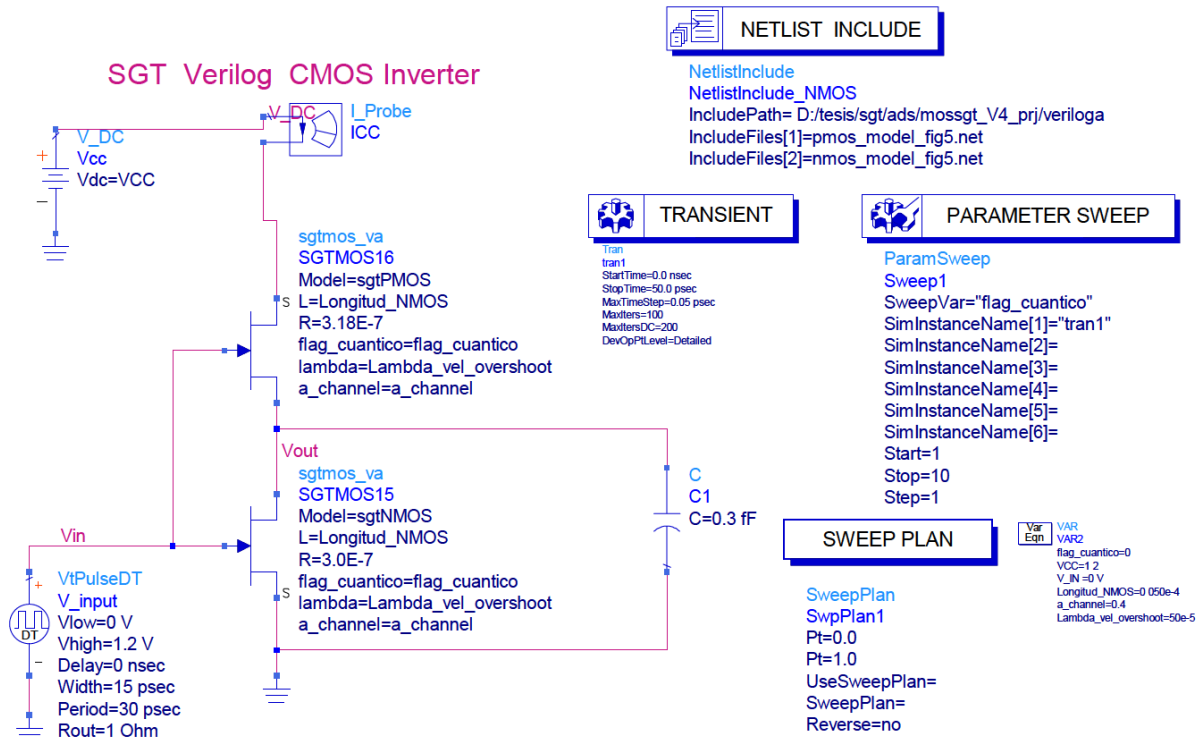


Figura 3.19 – Hoja de simulación en ADS® de un tren de pulsos de tensión de entrada a un inversor CMOS con dispositivos SGT N y P para $L = 50$ nm y $R = 3$ nm.

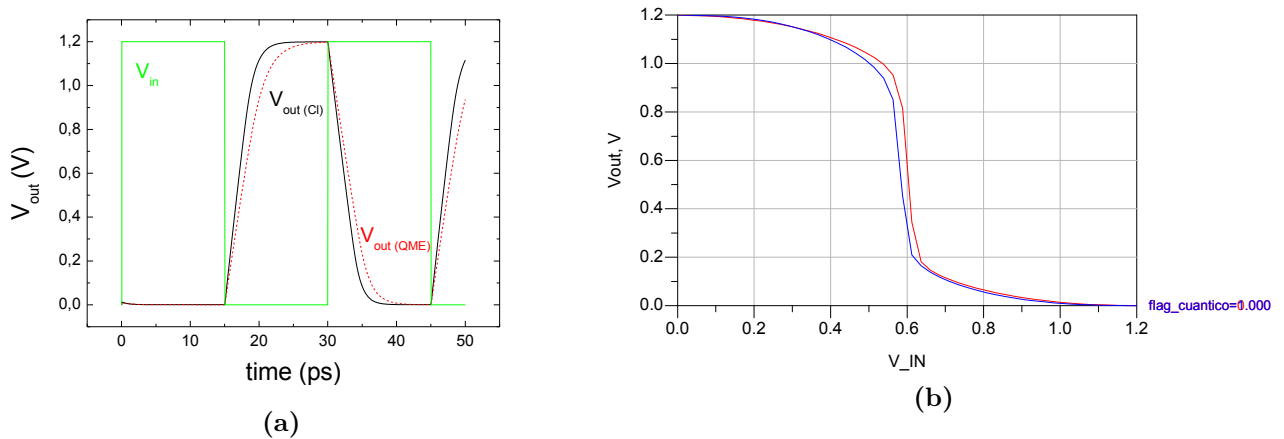


Figura 3.20 – Tensiones de entrada V_{IN} y salida V_{OUT} de un inversor CMOS para dispositivos SGT N y P con $L = 50$ nm, $R = 3$ nm y $t_{ox} = 1,5$ nm (a) y curva de transferencia IN/OUT del inversor CMOS con línea roja para el modelo clásico y azul para el cuántico (b).

con el parámetro λ_a . Este hecho es razonable para dispositivos con 50 nm de canal, para los que se sabe que estos efectos son importantes [RGLV+98].

En la figura 3.23 se muestra la frecuencia de oscilación del anillo frente al parámetro del overshoot de la velocidad, para diferentes radios del canal cilíndrico de silicio, teniendo en cuenta los efectos cuánticos (línea continua) y despreciándolos (línea discontinua). La diferencia es muy importante. Se aprecia una reducción aproximada del 30 % (o incluso

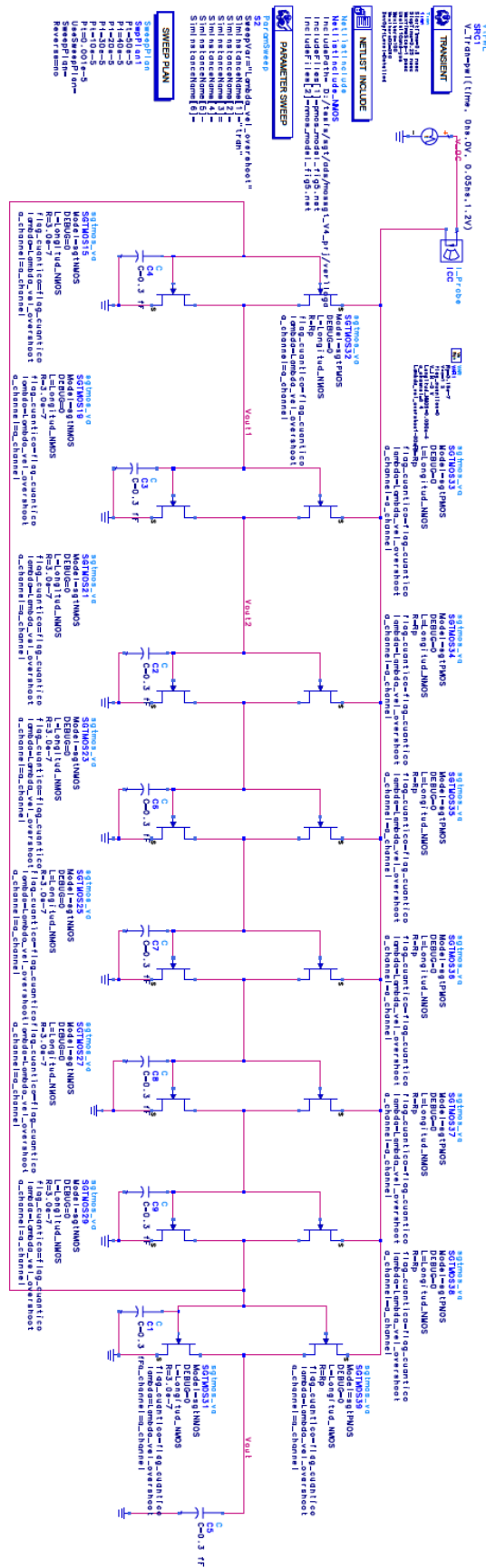


Figura 3.21 – Hoja de simulación en ADS® para un anillo oscilador CMOS para un anillo oscilador con C_L .

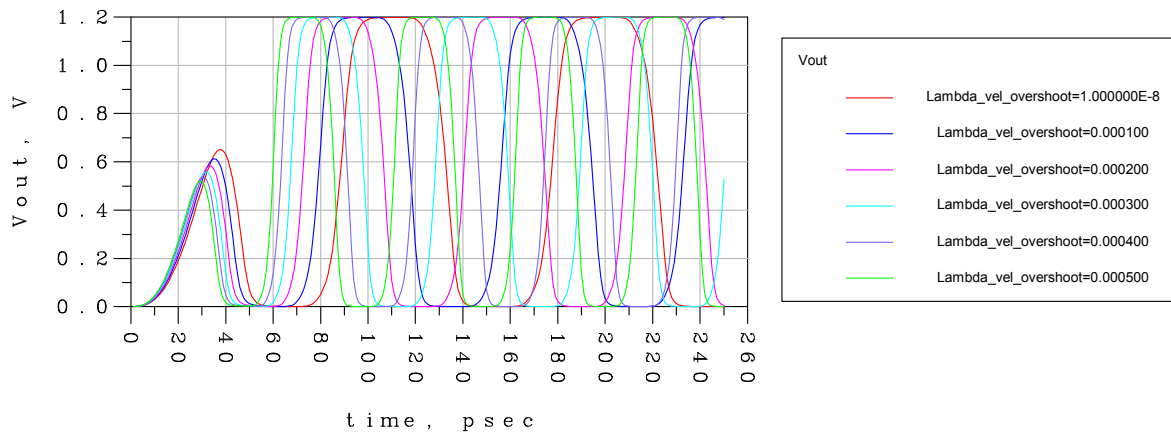


Figura 3.22 – Salida temporal del anillo frente al parámetro del overshoot de velocidad. Los parámetros tecnológicos de los dispositivos *SGT* usados en la simulación son $L = 50$ nm, $t_{ox} = 1,5$ nm.

mayor) si estos efectos se desactivan. Como se ha explicado en los apartados anteriores, esto es debido al gran confinamiento estructural de la carga en inversión producido por la geometría de estos dispositivos.

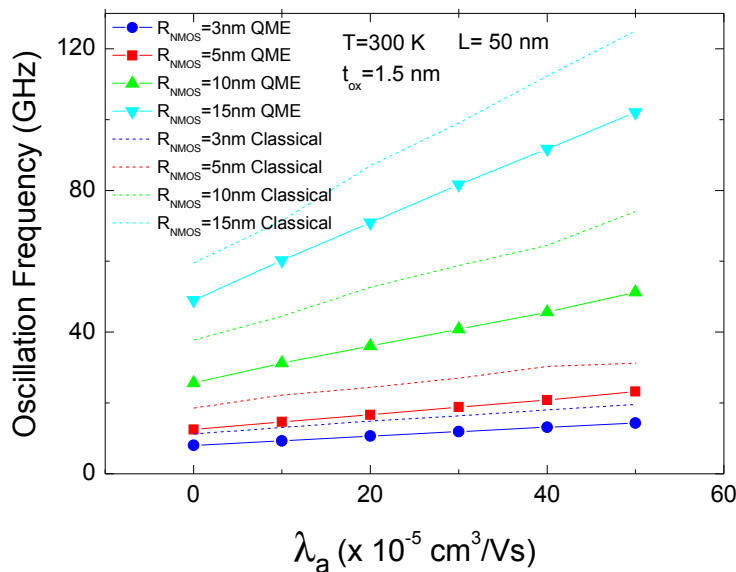


Figura 3.23 – Frecuencia del anillo oscilador frente al parámetro del overshoot de la velocidad, teniendo en cuenta los efectos cuánticos del modelo y el planteamiento clásico. Las características tecnológicas de los dispositivos *SGT* usadas en la simulación son $L = 50$ nm, $t_{ox} = 1,5$ nm.

La frecuencia de oscilación del anillo en función del radio teniendo en cuenta el modelo cuántico y el clásico, para diferentes valores del parámetro del *overshoot* de la velocidad, se presenta en la figura 3.24. Se observa cómo la variación de frecuencia de oscilación es una función compleja de las características geométricas del dispositivo y de efectos físicos como los modelados e incluidos en el cálculo de la figura. Se concluye que la potencia del modelo presentado aquí, desde el punto de vista del diseñador de circuitos, es muy alta y puede

permitir afinar diseños de circuitos basados en esta tecnología de transistores SGT.

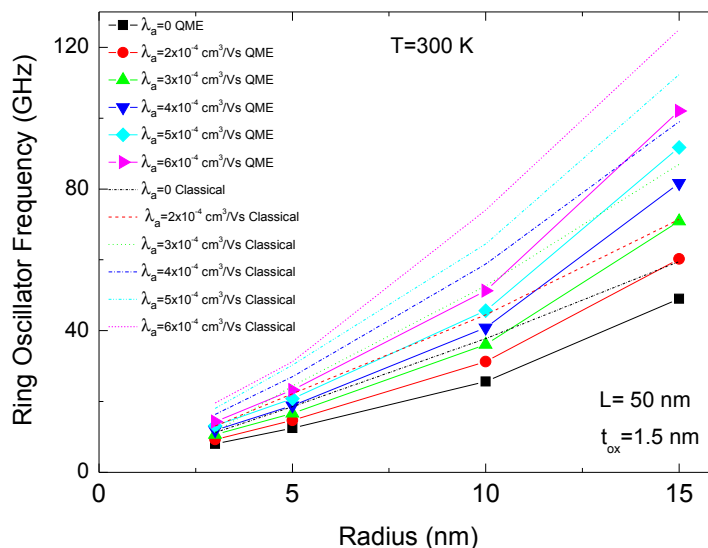


Figura 3.24 – Frecuencia de oscilación del anillo en función del radio, teniendo en cuenta el modelo cuántico y el clásico, para diferentes valores del parámetro del overshoot de la velocidad. Los parámetros tecnológicos de los dispositivos SGT son $L = 50$ nm, $t_{ox} = 1,5$ nm.

Los efectos del *overshoot* de la velocidad están conectados con la distribución de campo eléctrico en la fuente virtual [RGLVC97] que debe ser modelada con precisión (su importancia es esencial cuando la longitud de canal alcanza valores donde el transporte balístico de la carga es importante) para calcular correctamente las figuras de mérito de los circuitos [RRGJM10].

3.3 Double-Gate MOSFETs.

En el desarrollo de los modelos de SGT y también DG-MOSFET, que se presentan en este capítulo para ser implementados en Verilog-A estamos utilizando un parámetro fundamental: la movilidad de bajo campo de los portadores. Este parámetro, la movilidad, ha sido medido experimentalmente, estudiado mediante herramientas de simulación, y caracterizado en diferentes dispositivos SOI debido a su importancia [CC03, GF01, UiT03, CoI08, NCB⁺11].

La idoneidad y viabilidad de una tecnología está estrechamente ligada al valor de un conjunto básico de parámetros que describen las características principales del funcionamiento de los transistores, entre ellos está la movilidad de bajo campo. Si una nueva tecnología (basada, por ejemplo, en la utilización de sustratos tensos, en el uso de diferentes orientaciones cristalográficas, etc.) permite obtener corrientes más altas que en dispositivos de tamaño similar de silicio convencional, las frecuencias de operación de los circuitos integrados (centrando nuestro razonamiento en una aplicación digital) basados en ella podrán ser mayores¹⁹.

¹⁹Véanse las figuras anteriores para caracterizar la frecuencia de oscilación de un anillo oscilador.

Sin embargo, a parte de la movilidad de bajo campo hay otros parámetros importantes como la velocidad de saturación. Este último parámetro es también esencial, sobre todo en dispositivos de tamaño nanométrico donde los portadores alcanzan con facilidad la velocidad de saturación. No obstante, no ha sido sometido al exhaustivo escrutinio que ha sufrido la movilidad de bajo campo, ya que su obtención experimental es mucho más complicada. La velocidad de saturación es un parámetro importante en la mayoría de modelos compactos actuales (EKV, PSP, BSIM5s, HiSim, etc) [DXH⁺06, EKV95, MYK⁺11, XWJ⁺09].

La reducción de la movilidad de los portadores a altos campos longitudinales se incorpora en los modelos haciendo uso de expresiones similares a la introducida por Thornber [Tho80], donde se hace uso de la velocidad de saturación. En algunos modelos la velocidad de saturación se utiliza como parámetro de ajuste; sin embargo, esta práctica no es muy recomendable pues, como se muestra más adelante, este parámetro está perfectamente definido para una tecnología determinada. Solamente en dispositivos muy pequeños y donde los efectos balísticos, relacionados con el *overshoot* de la velocidad [Lun00], sean apreciables, el valor de la velocidad de saturación puede considerarse de menor importancia. En muchos modelos clásicos de MOSFET, la entrada en la región de operación de saturación se asocia a la saturación de la velocidad de los portadores en la región próxima al drenador [Tsi03].

Con el fin de caracterizar la velocidad de saturación en los dispositivos que estamos modelando, hemos hecho uso del simulador de Monte Carlo desarrollado en el grupo de nanoelectrónica de la Universidad de Granada. Para ello hemos caracterizado el transporte estacionario de alto campo longitudinal en una estructura SOI MOSFET de puerta simple, en un primer paso de aproximación al problema, y también de puerta doble. Hemos investigado las dependencias con la anchura de la capa de silicio y el papel de los diferentes mecanismos de dispersión, sobre todo la rugosidad superficial. La falta de resultados experimentales y de simulaciones para los SGT nos ha llevado a utilizar los datos obtenidos para los dispositivos DG-MOSFET. A pesar de ser estos últimos dispositivos unidimensionales, su estructura es la más parecida a los SGT, por tanto, es razonable utilizar los datos de velocidad de saturación en su caso. En este sentido, realizamos un paralelismo con la aproximación utilizada para la movilidad de bajo campo, ya que ésta es también una adaptación de un modelo correspondiente a un DG-MOSFET. No obstante, como se mostró anteriormente, el ajuste de los datos experimentales del SGT con el modelo que se propone es bueno, y teniendo en cuenta la ausencia absoluta de modelos analíticos y razonablemente compatibles con las necesidades del modelado compacto, encajan las aproximaciones realizadas.

Los dispositivos simulados están formados por una estructura SOI con el sustrato de silicio de orientación cristalográfica $\langle 100 \rangle$, y dopado de $N_A = 10^{16} \text{ cm}^{-3}$ para los SG-MOSFET e intrínseco para los DG-MOSFET. La anchura del aislante de puerta es de 1 nm y el óxido enterrado de 80 nm (para los dispositivos de doble puerta, DG-MOSFET simétricos, los dos óxidos tienen el mismo grosor). La puerta está formada por n^+ poly. La estructura unidimensional se caracteriza mediante la resolución autoconsistente de las ecuaciones de Schroedinger y Poisson, considerando un *grid* no uniforme adaptativo [RGLV⁺98]. El simulador de Monte Carlo utiliza las funciones de onda, potencial y otras magnitudes obtenidas resolviendo las ecuaciones anteriores [GF01, RGR06, DGR06, AFS82, GFW⁺85]. En la simulación

se utilizan mecanismos de dispersión por fonones y rugosidad superficial. La energía de los portadores se limita a 0.5 eV, por tanto no hace falta incluir mecanismos de ionización por impacto. La validez y exactitud del simulador ha sido comprobada reproduciendo resultados experimentales de dispositivos SOI [GLVR+98].

3.3.1 (Single Gate) SG-MOSFET

La dependencia de la velocidad de saturación con el campo efectivo ha sido estudiada en dispositivos con $T_{Si} = 2$ nm (anchura de la capa de silicio), como se muestra en la figura 3.25.

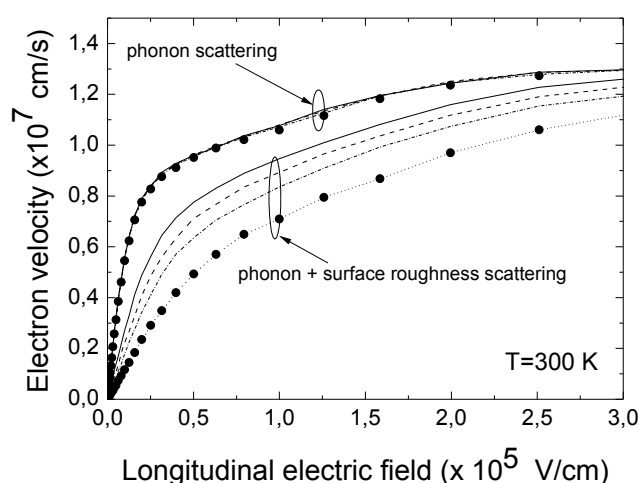


Figura 3.25 – Velocidad de los electrones en función de campo longitudinal para SG-MOSFET a temperatura ambiente, $T_{Si} = 2$ nm. Los campos efectivos utilizados para las diferentes curvas son los siguientes (línea continua – $E_{EFF} = 0,69 \times 10^5$ V/cm, línea discontinua – $E_{EFF} = 3,7 \times 10^5$ V/cm, línea discontinua y puntos – $E_{EFF} = 8,8 \times 10^5$ V/cm, línea discontinua y círculos – $E_{EFF} = 16,3 \times 10^5$ V/cm)

Como se puede observar, si se utiliza solo dispersión por fonones todas las curvas (para diferentes campos efectivos) coinciden, al igual que sucede en dispositivos convencionales *bulk*. Sin embargo, cuando se incorporan los mecanismos de dispersión por rugosidad superficial las curvas de velocidad dependen del campo efectivo correspondiente. En este caso, la velocidad de saturación disminuye si aumenta el campo. La dependencia de la velocidad de saturación con la anchura de la lámina de silicio es también de gran interés tecnológico ya que la inmunidad que presentan estos dispositivos a los SCE está relacionada con el grosor de la capa de silicio. Para profundizar en este aspecto, se han simulado dispositivos con T_{Si} variando en el intervalo de 2-8 nm, como se muestra en la figura 3.26 [RGRR06]. En estas simulaciones, para bajos campos longitudinales la población de electrones de las diferentes subbandas de energía coincide con la del equilibrio, pero si el régimen de transporte se desplaza hacia altos campos longitudinales la energía de los portadores crece y la evolución de la población en las diferentes subbandas (que como se sabe tienen asociada una masa efectiva para la conductividad distinta) cambia, lo que produce que las curvas de velocidad se crucen para campos longitudinales de aproximadamente 3×10^4 (V/cm). Hemos estudiado

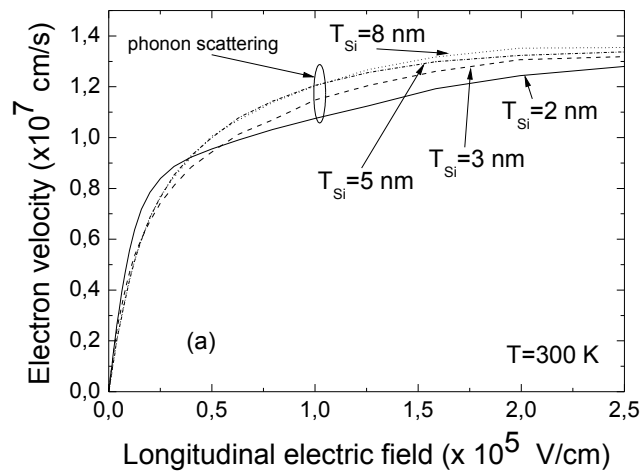


Figura 3.26 – Velocidad de los electrones en función del campo longitudinal en *SG-MOSFET* a temperatura ambiente. Sólo están incluidos mecanismos de dispersión por fonones. El campo efectivo correspondiente a las curvas mostradas es de $E_{EFF} = 7,4 \times 10^5$ (V/cm).

también la dependencia de la velocidad de saturación con la rugosidad de la superficie del óxido. Su caracterización se muestra en la figura 3.27.

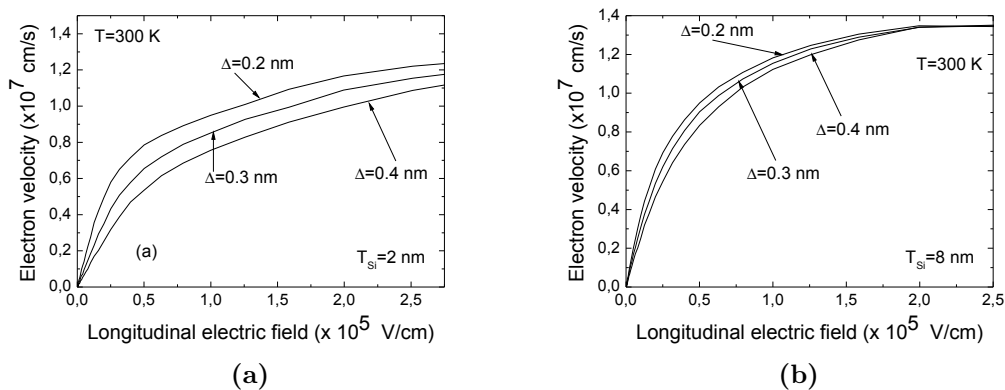


Figura 3.27 – Velocidad de los electrones en función del campo longitudinal para *SGMOSFETs* a temperatura ambiente (a) $T_{Si} = 2$ nm, (b) $T_{Si} = 8$ nm. Los mecanismos de dispersión considerados han sido fonones y rugosidad superficial (el parámetro Δ utilizado varía entre 0.2 y 0.4 nm. El parámetro Δ representa las variaciones abruptas de la interfase semiconductor-óxido relacionadas con el modelo desarrollado en [GRLV⁺99]). El campo efectivo correspondiente a las curvas es de $E_{EFF} = 7,4 \times 10^5$ (V/cm).

Como se puede observar, la velocidad de saturación depende de la rugosidad superficial y, en general, la velocidad baja si la rugosidad superficial aumenta. Para el dispositivo con la anchura de silicio menor, la carga está más cerca de las superficies del óxido, por eso la influencia de la rugosidad superficial es mayor que para el caso (b), de T_{Si} más ancho, y por consiguiente la velocidad de saturación es menor. Queda claro a la vista de estas figuras, que una mejora de la rugosidad superficial mediante la modificación de los procesos tecnológicos aumenta no solo la movilidad de bajo campo sino también la velocidad de

saturación. Finalmente hemos considerado una estructura SOI de doble puerta simétrica. Los dos óxidos (de grosor 1 nm) y el material de puerta (n^+ poly), así como la tensión de puerta es similar en todos los casos. El sustrato de silicio es intrínseco. Las curvas de velocidad obtenidas en este caso han sido las siguientes (figura 3.28). Las curvas de velocidad para las

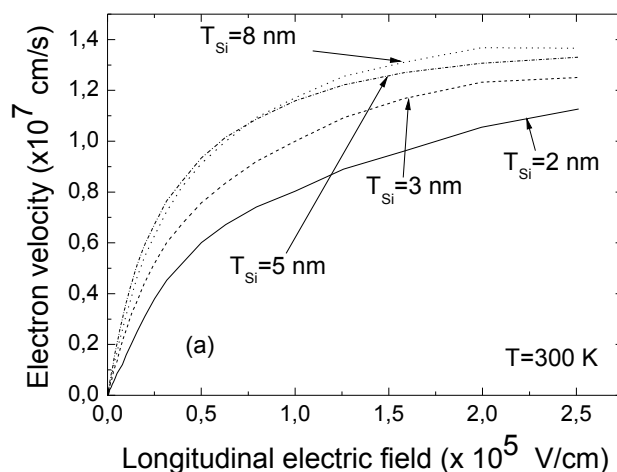


Figura 3.28 – Velocidad de los electrones en función del campo longitudinal para *DG-MOSFETs* a temperatura ambiente. Los mecanismos de dispersión considerados han sido fonones y rugosidad superficial. El campo efectivo correspondiente a las curvas es de $E_{EFF} = 5,7 \times 10^5$ V/cm.

diferentes anchuras de silicio no son iguales. El comportamiento de la velocidad de saturación es parecido al obtenido en *SG-MOSFET*, aunque se puede observar una separación mayor de las curvas (este resultado está relacionado con la diferente distribución de la población de electrones en las subbandas en los dispositivos *SG-MOSFET* y *DG-MOSFET*). La velocidad de saturación disminuye cuando se reduce la anchura de la lámina de silicio [RGRR06].

3.3.2 Modelo I-V de un transistor *DG-MOSFET*

En este apartado, y considerando los resultados expuestos anteriormente, se desarrolla un modelo de corriente para transistores de doble puerta simétricos con el sustrato intrínseco. En el desarrollo del modelo, como en el caso anterior, se tienen en cuenta los efectos cuánticos, los efectos de saturación de la velocidad, los efectos del *overshoot* de la velocidad y también los efectos de canal corto. En la figura 3.29 se esquematiza la geometría del dispositivo que vamos a modelar donde se consideran puertas metálicas. La longitud del canal se considera lo suficientemente larga para que el control electrostático sea descrito por la ecuación de Poisson 1D en la dirección y del canal.

El planteamiento de resolución de la ecuación de Poisson bajo la aproximación GCA es:

$$\frac{d^2\psi(x)}{dx^2} = \frac{d^2(\psi(x) - V)}{dx^2} = \frac{q}{\epsilon_{si}} n_i e^{\frac{q(\psi(x)-V)}{kT}} \quad (3.3.1)$$

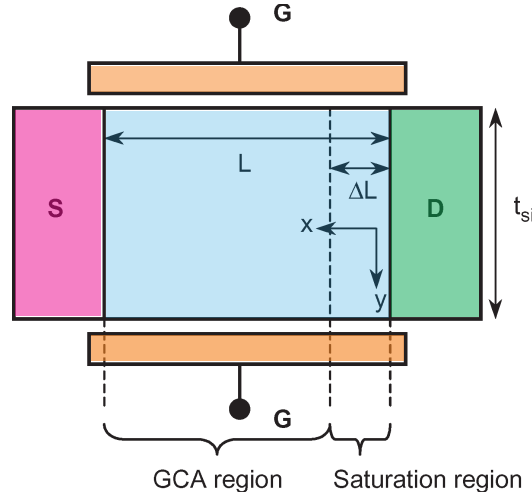


Figura 3.29 – Estructura de transistor *DG-MOSFET*. En la zona a la izquierda del canal tenemos la región *GCA* (gradual channel approximation), en la que se puede aplicar razonablemente la aproximación de canal gradual. A la derecha está la región de saturación o zona de pinch-off (zona de alto campo longitudinal).

siendo q la carga del electrón, n_i la concentración intrínseca de portadores, ϵ_{Si} la permitividad dieléctrica del silicio, $\psi(x)$ el potencial electrostático y $V = V(x)$ el nivel de Fermi para los electrones en el canal.

Las condiciones de contorno son las siguientes:

$$\psi(x)|_{(x=0)}^{\text{Fuente}} = \phi_{bi} \quad y \quad \psi(x)|_{(x=L)}^{\text{Drenador}} = \phi_{bi} + V_{DS} \quad (3.3.2)$$

donde ϕ_{bi} representa el potencial de *built-in* entre sustrato y contacto de fuente.

Resolviendo 3.3.1 con las condiciones de contorno 3.3.2 y aplicando la ley de Gauss con algunas aproximaciones se obtiene la relación implícita que relaciona la carga en el canal con la tensión de polarización donde Q representa la densidad de carga en inversión por unidad de área [MJG+07].

$$V_{GS} - \Delta\varphi - V + V_{TH} \ln \frac{qn_i T_{Si}}{8C_{oxide} V_{TH}} - V_{TH} \ln \frac{C_{oxide}}{C_{Si}} = \frac{Q}{2C_{oxide}} + V_{TH} \ln \frac{Q}{8C_{oxide} V_{TH}} + V_{TH} \ln \left[\frac{Q}{8C_{oxide} V_{TH}} + \frac{C_{Si}}{C_{oxide}} \right] \quad (3.3.3)$$

donde $V_{TH} = kT/q$, V_{GS} es la tensión puerta-fuente, ($V = 0$ en la fuente; todos los cálculos en este apartado serán realizados en este punto). $C_{oxide} = \epsilon_{oxide}/T_{ox}$ y $\Delta\varphi$ es la diferencia de función trabajo entre el silicio intrínseco y el electrodo de puerta.

3.3.2.1 Efectos cuánticos. Modelado del centroide de carga en inversión.

En los dispositivos *DG-MOSFET* existe aproximadamente el doble de carga en inversión en comparación con los transistores de una sola puerta. El centroide de la lámina de

inversión en este dispositivo se define como la penetración media de la distribución de la lámina de inversión en el silicio y puede ser calculada como 3.3.4 [LVCCG+00].

$$z_I = \frac{\int_0^{T_{Si}/2} zn(z)dz}{\int_0^{T_{Si}/2} n(z)dz} \quad (3.3.4)$$

donde $n(z)$ representa la concentración de electrones en la dirección perpendicular a la superficie del óxido y $T_{Si}/2$ se elige en el punto intermedio de la lámina de silicio, equidistante a las dos superficies del óxido.

Siguiendo el trabajo [LVCCG+00], se modela empíricamente el valor del centroide de carga en inversión y se define una nueva capacidad de óxido que tiene en cuenta los efectos cuánticos. Estos efectos se capturan mediante una capacidad asociada del centroide que se coloca en serie a la del óxido (de manera similar a la desarrollada en los dispositivos SGT anteriormente).

La nueva capacidad de óxido (C_{Oxide}^*) se obtiene siguiendo lo establecido en [LVCCG+00]:

$$\frac{1}{C_{oxide}^*} = \frac{1}{C_{oxide}} + \frac{1}{C_{centroid}} \quad (3.3.5)$$

$$C_{oxide}^* = \frac{C_{oxide}}{1 + C_{oxide} \frac{z_I}{\epsilon_{si}}} \quad (3.3.6)$$

3.3.2.2 Modelo de carga en inversión incluyendo efectos cuánticos.

En el cálculo de la carga en inversión utilizamos la expresión aproximada explícita presentada en [MJG+07]:

$$Q = 2C_{oxide}^* \left[-\frac{2C_{oxide}^* V_{TH}^2}{Q_0} + \sqrt{\left(\frac{2C_{oxide}^* V_{TH}^2}{Q_0} \right)^2 + 4V_{TH}^2 \ln^2 \left(1 + \exp \left(\frac{V_{GS} - V_{TQM} + \Delta V_{TQM} - V}{2V_{TH}} \right) \right)} \right] \quad (3.3.7)$$

donde:

$$Q_0 = 4V_{TH}C_{si} \quad \text{siendo} \quad C_{si} = \epsilon_{si}/T_{si} \quad (3.3.8)$$

$$V_0 = \Delta\varphi - V_{TH} \ln \left(\frac{qn_i T_{si}}{2Q_0} \right) \quad (3.3.9)$$

$$V_{TQM} = V_0 + 2V_{TH} \ln \left(1 + \frac{Q'}{2Q_0} \right) \quad (3.3.10)$$

$$\Delta V_{TQM} = \frac{\left(\frac{2C_{oxide}^* V_{TH}^2}{Q_0} \right) Q'}{\left(Q_0 + \frac{Q'}{2} \right)} \quad (3.3.11)$$

El valor de Q' se obtiene de una expresión parecida a la (3.3.7) pero simplificando el cálculo de la capacidad del óxido mejorada y el cálculo de ΔV_{TQM} y V_{TQM} , según se describe en [MJG+07].

3.3.2.3 Modelo de corriente de drenador.

Una vez que se ha modelado la carga en inversión en el canal de un dispositivo DG-MOSFET, se puede obtener la corriente de drenador integrando la carga. La corriente (3.3.12) se obtiene integrando la carga móvil en el canal asumiendo que el modelo de transporte de difusión y deriva puede ser utilizado (para canales suficientemente largos) [SKP+05].

$$I_{DS} = -\mu_{\text{eff}} \frac{W}{L} \int_{V_S}^{V_D} Q(V) \cdot dV \quad (3.3.12)$$

Para desarrollar la expresión (3.3.12) es necesario obtener dV derivando la expresión (3.3.3) donde se obtiene:

$$dV = \frac{dQ}{2C_{\text{ox}}} + V_{\text{TH}} \frac{dQ}{Q} + V_{\text{TH}} \frac{dQ}{8V_{\text{TH}}C_{\text{si}} + Q} \quad (3.3.13)$$

Introduciendo (3.3.13) en la ecuación de la corriente (3.3.12) se obtiene [MJG+07]:

$$\begin{aligned} I_{DS} &= -\mu_{\text{eff}} \frac{W}{L} \int_{Q_s}^{Q_d} Q \frac{dQ}{2C_{\text{ox}}} + V_{\text{TH}} dQ + V_{\text{TH}} \frac{Q dQ}{8V_{\text{TH}}C_{\text{si}} + Q} \\ &= \mu_{\text{eff}} \frac{W}{L} \left[-\frac{Q^2}{4C_{\text{ox}}} \Big|_{Q_s}^{Q_d} - 2V_{\text{TH}} Q \Big|_{Q_s}^{Q_d} + 8V_{\text{TH}}^2 C_{\text{si}} \ln(8V_{\text{TH}}C_{\text{si}} + Q) \Big|_{Q_s}^{Q_d} \right] \end{aligned} \quad (3.3.14)$$

$$= \mu \frac{W}{L} \left[2 \frac{kT}{q} (Q_s - Q_d) + \frac{Q_s^2 - Q_d^2}{4C_{\text{ox}}} + 8 \left(\frac{kT}{q} \right)^2 C_{\text{si}} \ln \left[\frac{Q_d + 2Q_0}{Q_s + 2Q_0} \right] \right] \quad (3.3.15)$$

La expresión de la corriente (3.3.15) necesita de los valores Q_d y Q_s que deben ser obtenidos mediante la expresión (3.3.7). Para evitar el coste computacional asociado al algoritmo de resolución numérica iterativo se puede hacer uso de una expresión explícita aproximada propuesta en [MJG+07].

Haciendo uso de un cálculo similar al realizado para el dispositivos SGT para incluir los efectos de *overshoot* de la velocidad y de la saturación de la velocidad de los portadores en el canal, introducimos la siguiente expresión:

$$I_{DS} = \frac{W}{L} F(V_{GS}, V_{DS}) \left[\frac{\mu_{\text{eff}}}{\left(1 + \delta_0 \frac{\mu_{\text{eff}} V_{DS}}{v_{\text{sat}} L}\right)} + \frac{\lambda_a}{L} \right] \quad (3.3.16)$$

donde

$$F(V_{GS}, V_{DS}) = \int_0^{V_{DS}} Q(V) dV \quad (3.3.17)$$

L es la longitud de canal del transistor, μ_{eff} es la movilidad de bajo campo (detalles del modelo para esta parámetro esencial en la referencia [TFG04]), v_{sat} es la velocidad de saturación (valor aproximado de 10^7 cm/s, aunque se han utilizado algunos de los resultados expuestos en la sección anterior para afinar el valor de la velocidad de saturación dependiendo de las características tecnológicas del dispositivo utilizado), λ_a el parámetro del *overshoot* de la velocidad [RGLVC97] (el valor de λ_a usado en este trabajo es de 40×10^{-5} cm³/Vs). δ_0 es el parámetro introducido en la referencia [ARHR94] para mejorar la precisión del modelo de velocidad de los electrones en función del campo eléctrico longitudinal (E_{long} en la ecuación (3.2.17)) [Tho80] cuando el parámetro β adopta el valor de 1 por simplicidad (β es un parámetro elegido normalmente entre 1 y 2. Asumimos el valor de $\beta = 1$, en línea con las aproximaciones realizadas por varios autores [ARHR94, GMCS07]).

La función $F(V_{GS}, V_{DS})$ representa la integral de la carga en inversión en el canal del dispositivo. Desarrollando se obtiene:

$$I_{\text{DS}} = \frac{W}{L} \left[\frac{\mu_{\text{eff}}}{\left(1 + \delta_0 \frac{\mu_{\text{eff}} V_{\text{DS}}}{v_{\text{sat}} L}\right)} + \frac{\lambda_a}{L} \right] \times \quad (3.3.18)$$

$$\times \left[2 \frac{kT}{q} (Q_s - Q_d) + \frac{Q_s^2 - Q_d^2}{4C_{\text{ox}}^*} + 8 \left(\frac{kT}{q} \right)^2 C_{\text{si}} \ln \left[\frac{Q_d + 2Q_0}{Q_s + 2Q_0} \right] \right]$$

donde Q_s y Q_d son la carga en inversión calculada siguiendo la ecuación 3.2.12, en la fuente ($V = 0$) y en el drenador ($V = V_{\text{DS}}$) del dispositivo respectivamente.

Al igual que en el caso del SGT hemos introducido el modelo desarrollado por Trivedi et al. [TFG04] que permite tener en cuenta la contribución de mecanismos de dispersión por fonones y rugosidad superficial. El modelo de movilidad usado presenta la misma expresión (3.2.19) que la usada para el SGT. La modulación de la longitud de canal se introduce a través de la siguiente expresión [LI08, CCE06, Aro07], ver ecuación (3.2.20).

El valor de la tensión de saturación, V_{DSsat} , se obtiene, como se ha detallado en [LI08, RGJM⁺10], estableciendo la continuidad a lo largo de todo el canal; como en el caso del SGT. Para incluir los efectos del *overshoot* de la velocidad en el DG-MOSFET hemos usado el mismo procedimiento que en el SGT (ecuación 3.2.21) [RGLVC97], por lo tanto, un término proporcional al gradiente del campo eléctrico longitudinal ($E_{\text{long}} = -dV(x)/dx$, con x siendo la coordenada que varía en la dirección fuente-drenador, $0 \leq x \leq L$, y $0 \leq V(x) \leq V_{\text{DS}}$) se añade a la velocidad de saturación. La inclusión de los efectos del *overshoot* de la velocidad nos conduce en el caso del DG-MOSFET a la expresión similar a (3.2.22) y (3.2.23). Para evitar tener que usar dos expresiones diferentes para la corriente de drenador (para la zona lineal y saturación) y poder hacer una transición suave entre ellas, hemos establecido una función de suavización (similar a la del caso SGT, ecuación (3.2.24)) según lo descrito en [Aro07, RGJM⁺10].

Para resumir, la expresión final de la corriente de drenador corresponde a:

$$I_{DS} = \frac{W}{(L - \Delta L)} \left[\frac{\mu_{\text{eff}}}{\left(1 + \delta_0 \frac{\mu_{\text{eff}} V_{DSx}}{v_{\text{sat}}(L - \Delta L)}\right)} + \frac{\lambda_a}{(L - \Delta L)} \right] \times \left[2 \frac{kT}{q} (Q_s - Q_d) + \frac{Q_s^2 - Q_d^2}{4C_{\text{ox}}^*} + 8 \left(\frac{kT}{q}\right)^2 C_{\text{si}} \ln \left[\frac{Q_d + 2Q_0}{Q_s + 2Q_0} \right] \right] \quad (3.3.19)$$

donde Q_d es obtenida mediante la ecuación (3.3.7) con $V = V_{DSx}$ y μ_{eff} es calculada a través de la ecuación similar a la del SGT (3.2.19) para los valores de tensión de puerta y drenador correspondientes.

Este modelo de transistores DG-MOSFET se ha implementado en Verilog-A de manera similar al de SGT. En cuanto a la implementación de las capacidades del dispositivo, hemos seguido un proceso similar al implementado para los SGTs. En este sentido, se ha seguido las líneas marcadas en [MJG+07]. Además, se han incluido los efectos cuánticos en el cálculo. También para esta implementación de las capacidades de la estructura se sigue el apéndice 7.6.

3.3.3 Simulación de circuitos con dispositivos DG-MOSFET.

Para el estudio de circuitos, similares a los introducidos en la sección de simulación correspondiente a los transistores SGT, es necesaria la implementación de las capacidades que caracterizan la estructura, además de los modelos de carga, corriente, etc., presentados en la subsecciones anteriores. Hemos seguido un procedimiento paralelo al utilizado para los SGTs para llevar a cabo esta tarea. El cálculo de las capacidades está basado en los resultados presentados por [MJG+07]; además, se han introducido las correcciones cuánticas en este cálculo siguiendo el modelo descrito en la sección 3.2.1. Una vez codificado el modelo desarrollado anteriormente para DG-MOSFETs en Verilog-A hemos realizado diferentes simulaciones. En este caso, el tiempo de simulación es bastante similar al de los transistores SGTs, y en lo que se refiere al cálculo de los algoritmos iterativos internos del modelo (e.g., continuidad de la corriente entre zonas de operación lineal y de saturación) no aparecen problemas de convergencia, ni siquiera en circuitos como el anillo oscilador que se presenta más adelante.

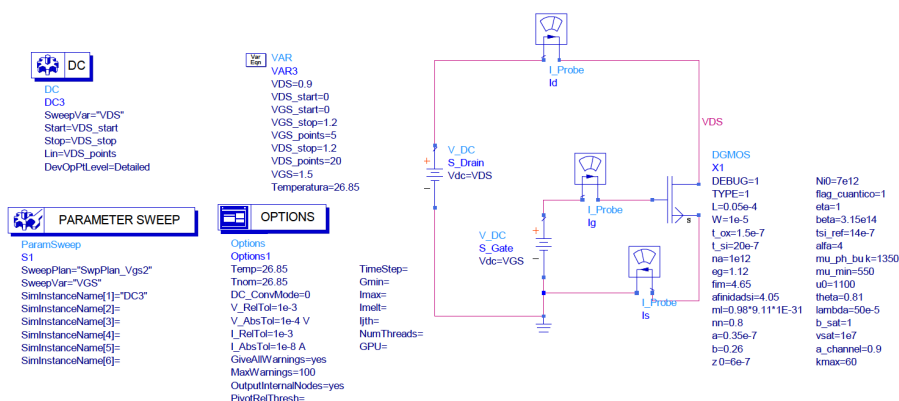


Figura 3.30 – Hoja de simulación en *Advanced Design System* para obtener curvas de salida en un dispositivo *DG-MOSFET* (tipo *N*) con los siguientes parámetros tecnológicos: $L=50$ nm, $W=100$ nm, $T_{\text{OX}}=1.5$ nm, $T_{\text{si}}=20$ nm.

Los resultados de simulación de curvas de salida en los dispositivos bajo estudio se muestran en la figura 3.31.

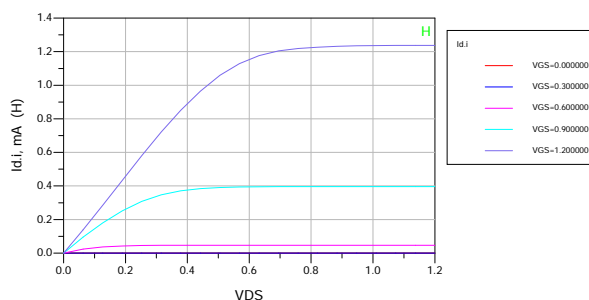


Figura 3.31 – Corriente I_{DS} obtenida de la simulación de la figura anterior mediante *ADS*[®] para un dispositivo *DG-MOSFET* tipo *N*.

Al igual que en la figura 3.17 (correspondiente al *SGT*) se observa continuidad en la transición de la zona lineal a la de saturación de las curvas de salida, asegurando el buen comportamiento de la conductancia de salida, parámetro esencial para la correcta simulación de circuitos analógicos. Hemos implementado también puertas lógicas, *CMOS* en este caso, utilizando los transistores *DG-MOSFET*. La hoja de simulación *ADS*[®] se muestra en la siguiente figura.

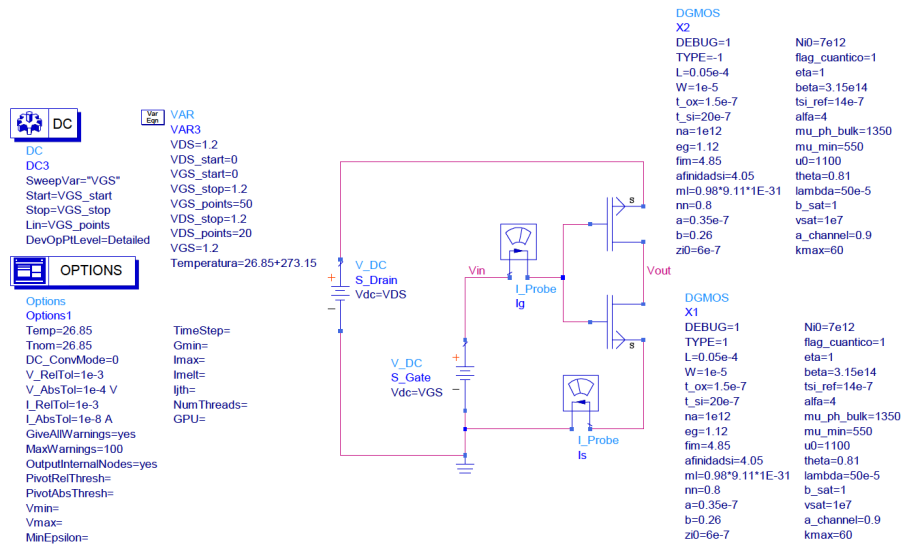


Figura 3.32 – Hoja de simulación en ADS® de un inversor CMOS con dispositivos DG-MOSFETs N y P para $L=50\text{ nm}$, $W=100\text{ nm}$, $T_{ox}=1.5\text{ nm}$, $T_{si}=20\text{ nm}$.

La salida de la simulación, utilizando $V_{DD}=1.5\text{ V}$ se muestra en la figura 3.33.

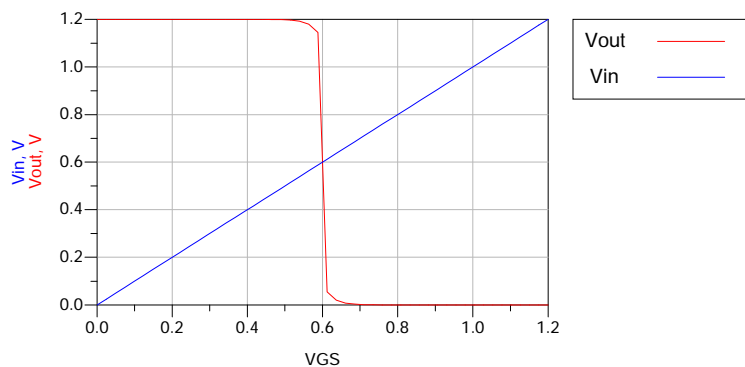


Figura 3.33 – Corriente I_{DS} obtenida de la simulación de la figura anterior mediante ADS® para una puerta CMOS.

Finalmente, hemos estudiado la salida de un anillo oscilador como el que se muestra en la hoja de simulación de ADS® de la figura 3.34. Como en el caso anterior, la capacidad de carga ($C_L = 2\text{ fF}$) recoge la contribución de las capacidades extrínsecas y de las líneas de metalización del circuito.

La tensión de salida del anillo se muestra en la figura 3.35. Se puede observar como se obtiene la oscilación de la tensión, y la capacidad del modelo desarrollado para abordar la simulación de circuitos digitales como éste.

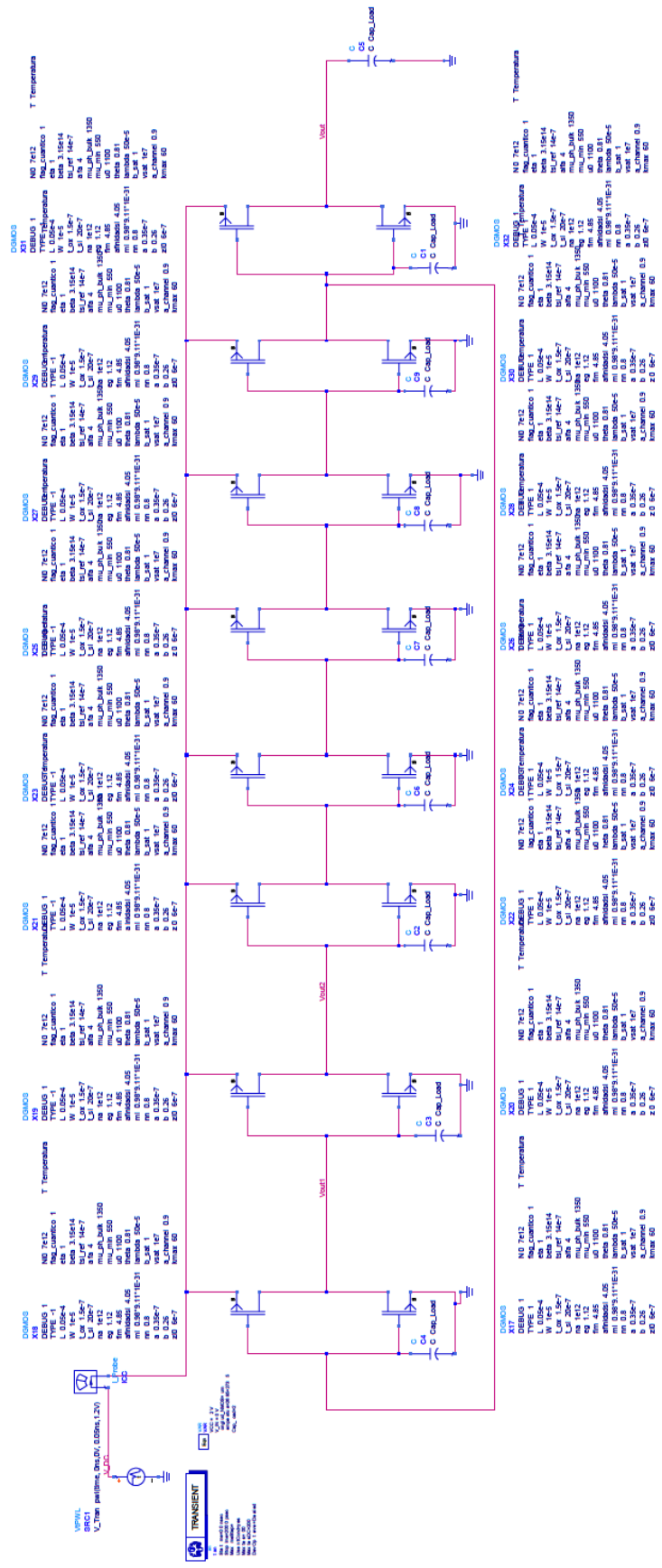


Figura 3.34 – Hoja de simulación en ADS® para un anillo oscilador CMOS de 7 etapas cargadas con $C_L = 2\text{ fF}$.

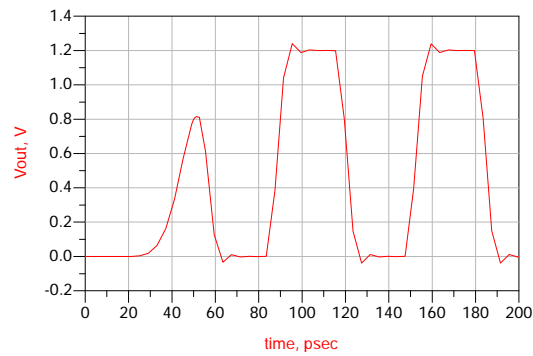


Figura 3.35 – Tensión de salida respecto al tiempo del anillo oscilador.

En la figura 3.36 se muestran en conjuntamente la tensión de salida y la potencia consumida por el oscilador. Se puede observar que una vez que se establece la forma de onda de la tensión de salida (80 ps) el número picos y valles en el consumo es igual al número de etapas del oscilador.

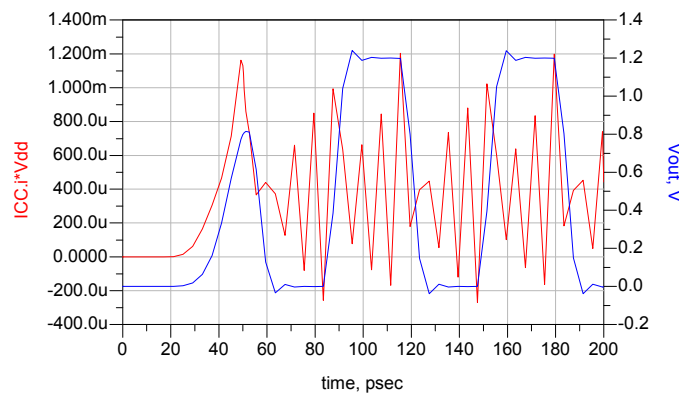


Figura 3.36 – Tensión de salida y potencia consumida respecto al tiempo del anillo oscilador CMOS de 7 etapas cargadas con $C_L = 2 \text{ fF}$.

CAPÍTULO

4

MODELADO DE SENSORES DE CORRIENTE MAGNETORRESISTIVOS.

4.1 Introducción

La medida de la corriente, la potencia y la energía eléctrica ha sido un tema de interés desde hace más de un siglo. Tradicionalmente, los transformadores y las resistencias *shunt* han sido los sensores más comunes en las aplicaciones de medida de las magnitudes anteriores. En concreto, para la medida de potencia los equipos electromecánicos han resuelto los requerimientos básicos durante años. Sin embargo, con el avance de la tecnología están apareciendo continuamente nuevos escenarios que demandan sistemas de medida integrables, precisos, versátiles y de bajo coste. Esta demanda está dirigiendo la investigación hacia nuevos sensores y técnicas de medida que satisfagan estos nuevos requerimientos.

La microelectrónica es uno de estos campos donde nuevos medidores son necesarios. Con el avance de la tecnología han aparecido una nueva serie de funciones y módulos integrados en los **IC**, **SoC** y **MEMS** que demandan monitorización de la distribución de potencia en diferentes partes del sistema. Los sistemas de medida convencionales no permitían su integración de manera monolítica estándar. Es necesario diseñar nuevos sistemas de medida y sensores que sean compatibles con la tecnología dominante **CMOS**. Los sensores de efecto Hall han sido utilizados con éxito durante mucho tiempo en la medida de corriente y potencia eléctrica. Sin embargo, con el descubrimiento de efecto magnetoresistivo gigante **GMR**

en 1988 por Fert [BBF+88] (a bajas temperaturas) y Grunberg [BGSZ89a] (a temperatura ambiente) se han desarrollado nuevos sensores de características muy interesantes que presentan ventajas con respecto al resto de sensores de corriente del mercado, ver figura 4.1.

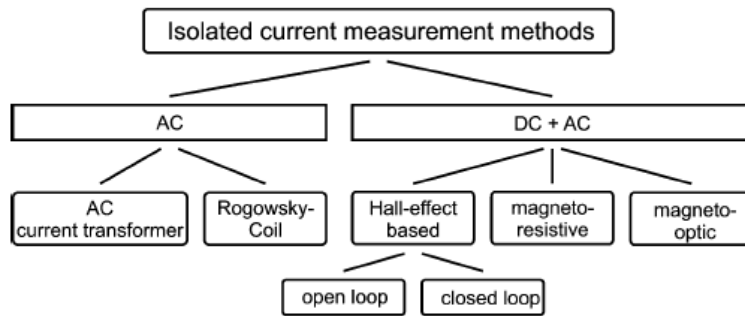


Figura 4.1 – Clasificación de los sensores de corriente [HBEK09].

Los sensores GMR presentan tres características intrínsecas que mejoran las de los sensores de efecto Hall. Primero, presentan una excelente sensibilidad a campos magnéticos muy bajos a temperatura ambiente. Segundo, son compatibles con la tecnología de fabricación CMOS y por lo tanto permite un alto nivel de integración y tercero, permite la medida campos paralelos a la superficie del IC, lo que es muy interesante en fabricación planar.

El efecto magnetorresistivo gigante GMR es un mecanismo de acoplamiento magnético que puede ser obtenido en estructuras multicapa con espesores de nanométricos [FFCC07]. En estos dispositivos, a temperatura ambiente, la resistencia varía con la aplicación de un campo magnético externo en unos niveles óptimos para su uso como sensores [BBF+88, BGSZ89b]. Así, en la actualidad, la utilidad de los dispositivos basados en GMR está fuera de toda duda, como se ha hecho fehaciente con la reciente concesión del Premio Nobel de Física en 2007 a los descubridores del mecanismo. Inicialmente usados en todas las cabezas lectoras de los actuales discos duros, figura 4.2, la constante evolución que está experimentando esta tecnología está abriendo nuevos campos de aplicación, principalmente relacionados con la medida de pequeños campos magnéticos mediante dispositivos miniaturizados, tanto la microelectrónica como en la biotecnología. En el caso de la microelectrónica, estos sensores pueden ser utilizados en aquellos escenarios en los que se requiere de una detección o medida de corriente eléctrica no intrusiva, mediante la medida indirecta del campo que ésta genera.

En lo que respecta a la biotecnología, numerosas bio-disciplinas se están beneficiando en la actualidad de estos avances. Así, por ejemplo, se ha descrito la utilización de sensores GMR en tratamientos de cáncer mediante hipertermia por inducción [MCGY07], en la detección de bacterias [MACn+09], en procesos parciales de secuenciación de ADN [XYH+08]. También se ha sugerido recientemente su aplicación en el diagnóstico de caries y en la medida de actividad neuronal.

Estos avances no están encontrando el acompañamiento comercial que merecen dada la marcada especificidad de las aplicaciones que de ellos se derivan. De hecho, este tipo de sensores tienen que ser, en la mayoría de casos, diseñados ex profeso para cada aplicación, por lo que las investigaciones siguen desarrollándose a nivel de laboratorio. Actualmente,

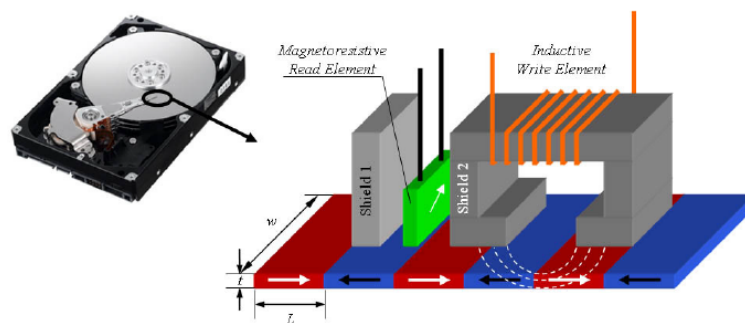


Figura 4.2 – Fotografía y detalle de una cabezal de lectura/escritura de un disco duro. El cabezal inductivo se usa durante la escritura para definir la magnetización de los bits mientras que el sensor magnetorresistivo detecta el campo magnético producido por los bits convirtiéndolo en una señal eléctrica [Fer07].

Non Volatile Electronics (NVE) e Infineon son las únicas compañías que tienen alguna línea de productos en este sentido, en aplicaciones de electrónica industrial principalmente.

Es importante recordar que todos los dispositivos anteriormente citados son habitualmente desarrollados directamente sobre substratos semiconductores, como silicio; aislantes, como vidrio, alúmina; o con tecnologías Silicon-On-Insulator SOI, siguiendo técnicas de microfabricación convencionales (litografía UV, *sputtering*, atacados físicos y químicos, etc.). La electrónica necesaria para la correcta funcionalidad de los mismos tiene que ser añadida posteriormente ya sea con tecnologías de integración híbrida, haciendo uso de técnicas de *wafer-bonding* (*System-on-Chip*, SoC y similares) o con planteamientos mucho más clásicos (utilización de *Printed Circuit Boards*, PCB, y cableado tradicional). Las ventajas que ofrece la posibilidad de generalizar los procesos necesarios para integrar monóticamente las tecnologías GMR y CMOS darían el espaldarazo definitivo a las actuales aplicaciones, y a la eclosión de muchas más.

El funcionamiento y la base de operación de los sensores que acabamos de describir en los párrafos anteriores es común a los sensores de corriente objeto de esta tesis. Por ello, los modelos que se proponen y se implementan en este trabajo tienen un gran potencial. Las diferentes adaptaciones de estos sensores a la biotecnología o a aplicaciones concretas de la electrónica, como las memorias no volátiles [ITR11], se pueden canalizar desde el punto de vista del modelado con ligeras modificaciones de los modelos que aquí se presentan. Por tanto, la proyección de los estos resultados de modelado, teniendo en cuenta que no existe actualmente casi ningún modelo compacto de dispositivos basados en GMR, es de muy largo alcance en diferentes tipos de aplicaciones.

4.1.1 Principios magnetorresistivos.

En este apartado se describen los fenómenos magnetorresistivos que constituyen la base física de los sensores que se estudiarán en este trabajo: el efecto magnetorresistivo en semiconductores (SMR), el efecto magnetorresistivo anisotrópico (AMR) y el gigante (GMR).

4.1.1.1 Efecto Hall y efecto magnetorresistivo (SMR) en semiconductores.

Los portadores de carga en semiconductores cuando están sometidos a un campo magnético externo sufren un efecto físico que aparece en su transporte debido a la fuerza de Lorentz.

La expresión de la fuerza que actúa sobre una partícula cargada en un campo electromagnético se presenta en la ecuación 4.1.1.

$$\vec{F} = e \cdot \vec{E}_e + e [\vec{v} \times \vec{B}] \quad (4.1.1)$$

donde e representa la carga de la partícula ($e = -q$ para electrones y $e = q$ para huecos), \vec{E}_e el campo eléctrico, v la velocidad de la partícula y finalmente \vec{B} la densidad de flujo magnético existente en la zona de desplazamiento de la carga. En la expresión 4.1.1 distinguiremos la parte primera como contribución debida al campo eléctrico y la segunda a la parte magnética. Tengamos ahora en cuenta el transporte de los portadores de carga a lo largo de una capa delgada de semiconductor de alto dopado tipo-n. Se desprecian los portadores minoritarios. La estructura se presenta en la figura 4.3. Sin tener en cuenta el movimiento térmico de

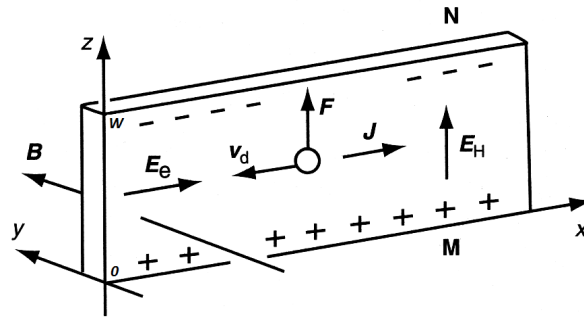


Figura 4.3 – Efecto Hall sobre semiconductor tipo-n [Mei08].

los portadores y dada la existencia de un campo eléctrico en la dirección \vec{x} con $\vec{E}_e = (E_x, 0, 0)$, los electrones sufren el efecto de deriva representado por el primer término de la expresión 4.1.1, que tiene la expresión:

$$\vec{v}_{dn} = \mu_n \cdot \vec{E}_e \quad (4.1.2)$$

donde μ_n representa la movilidad de los electrones que al moverse generan una densidad de corriente representada por la ecuación 4.1.3

$$\vec{J}_n = q\mu_n \cdot n \cdot \vec{E}_e \quad (4.1.3)$$

Al aplicar el campo magnético externo \vec{B} , paralelo al eje \vec{y} ver figura 4.3, aparece una fuerza que desplaza a las cargas en movimiento con una magnitud media de:

$$\vec{F}_n = e [\vec{v}_{dn} \times \vec{B}] \quad (4.1.4)$$

donde si se sustituye la ecuación 4.1.2 se obtiene:

$$\vec{F}_n = q\mu_n [\vec{E}_e \times \vec{B}] \quad (4.1.5)$$

De este modo el campo magnético \vec{B} empuja a los portadores hacia la parte superior del material semiconductor, (figura 4.3) donde la concentración de electrones empieza a crecer mientras que decrece en la parte inferior. Ésto provoca la aparición de un campo eléctrico entre los bordes superior e inferior del material, que se representará mediante \vec{E}_H y que interactúa a su vez con los electrones en movimiento empujándolos hacia abajo. Una vez se alcanza el equilibrio, existe un balance entre el campo vertical \vec{E}_H y la fuerza magnética representado por la ecuación 4.1.6.

$$e [\vec{E}_e \times \vec{B}] + e\vec{E}_H = 0 \quad (4.1.6)$$

A partir de este momento, los electrones comienzan a circular paralelos al eje del material (\vec{x}) como si el único campo aplicado fuese el externo. El campo eléctrico transversal (\vec{E}_H) que compensa la acción de la fuerza magnética de Lorentz se denomina *campo eléctrico Hall* y su valor es:

$$\vec{E}_H = - [\vec{v}_d \times \vec{B}] \quad (4.1.7)$$

o lo que es equivalente, teniendo en cuenta la ecuación (4.1.2) a:

$$\vec{E}_{Hn} = -\mu_n [\vec{E}_e \times \vec{B}] \quad (4.1.8)$$

Existe un efecto tangible asociado al campo eléctrico Hall consistente en la aparición de una diferencia de potencial entre los extremos superior e inferior del material. Esta diferencia de potencial se conoce como *tensión Hall* y cuyo valor se presenta en la ecuación (4.1.9)

$$V_H = \int_{z=0}^{z=w} E_H dz \quad (4.1.9)$$

Teniendo en cuenta (4.1.9), la geometría del material y el dopado se podría obtener una expresión equivalente:

$$V_{Hn} = \mu_n E_x B_y w \Rightarrow V_{Hn} = R_H \cdot B \quad (4.1.10)$$

donde R representa la resistencia de entrada de la muestra.

Las consideraciones anteriores muestran únicamente una descripción general del efecto Hall. Para comprender la influencia de la red cristalina en el transporte se deben tener en cuenta los efectos de las colisiones de los portadores con la red.

Una partícula cargada expuesta a campos eléctricos y magnéticos ortogonales se mueve describiendo una cicloide en el vacío [Kit86]. Por otra parte en un sólido, los portadores, que presentan una masa para el transporte caracterizada por la masa efectiva m^* , describen parcialmente la trayectoria del cicloide y pierden la mayor parte de su energía debido a las colisiones en el sólido para tiempos de vuelo mayores al tiempo de relajación de la

energía [Pop91], ver figura 4.4. La partícula inicia nuevamente una trayectoria cicloidea

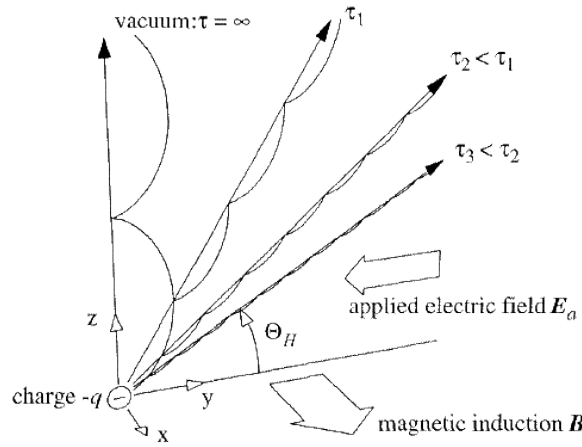


Figura 4.4 – Representación esquemática de la trayectoria que describen los electrones en vacío y en medio sólido en presencia de campos magnético y eléctrico. Tras el tiempo de relajación τ , el electrón pierde toda su energía cinética por las colisiones y comienza a describir una nueva trayectoria en la dirección del campo eléctrico aplicado [Kit86].

en la dirección del campo eléctrico. El ángulo medio de deflexión Θ_H está relacionado con el tiempo de relajación τ mediante la expresión (4.1.11) [Kit86].

$$\tan \Theta_H = -\omega_c \tau = \frac{-qB}{m^*} \tau = -\mu_n B \quad (4.1.11)$$

donde ω_c representa la frecuencia de la cicloide.

Sin embargo la magnitud del campo eléctrico Hall de la ecuación (4.1.8) puede ser también representada por el ángulo Hall $\tan \Theta_H = |E_H| / |E_e|$. Por lo tanto, las colisiones en la presencia de un campo de inducción magnética producen que el campo eléctrico resultante $E = E_e + E_H$ no sea paralelo al campo eléctrico aplicado E_e o a la densidad de corriente J_n .

La presencia del ángulo Hall Θ_H entre la densidad de corriente J_n y el campo eléctrico resultante E produce un incremento de la longitud de la trayectoria de las cargas en su movimiento en el sólido. Como resultado la resistividad de la muestra se incrementa. Este efecto es conocido como efecto magnetorresistivo geométrico en semiconductores [Pop91] donde la resistencia $R \propto B^2$.

4.1.1.2 Efecto magnetorresistivo anisotrópico (AMR).

Las variaciones de la resistividad que se producen en un material ferromagnético, cuando es sometido a un campo magnético externo que produce alteraciones de las propiedades de éste al interactuar con su vector de magnetización, se conoce como efecto magnetorresistivo anisotrópico AMR. Se genera un aumento de la resistencia cuando la corriente tiene la misma dirección del campo magnético externo y una disminución cuando son perpendiculares. El descubrimiento de este efecto se produjo en 1856 por William Thomson. Sus

experimentos con hierro y posteriormente con níquel desvelaron que las variaciones resistivas tenían el mismo comportamiento pero con mayor magnitud. En la figura 4.5.a se presenta la estructura de una magnetorresistencia AMR a través de la que circula una corriente de polarización I . Esta corriente produce una densidad de corriente \vec{J} paralela a la dirección del eje de avance de la corriente que interacciona con un campo magnético exterior que produce una magnetización \vec{M} en el material ferromagnético la cual está girada un ángulo θ con respecto a \vec{J} . Los materiales utilizados son generalmente metales de transición $3d$ y sus aleaciones. El material magnético que se suele utilizar para fabricar las magnetorresistencias es una aleación de níquel-hierro ($Ni_{80}Fe_{20}$) que presenta gran sensibilidad a la rotación magnética cuando actúan campos magnéticos externos y baja **coercitividad magnética**. Para la elección de materiales magnetorresistivos existe un criterio general: el material seleccionado debe tener baja anisotropía, un efecto magnetorresistivo elevado, elevada resistencia en una pequeña área y baja **magnetostricción**. Los materiales que cumplen estos requisitos son, entre otros, los combinados binarios de Ni, Fe y Co. El más típico es NiFe (81/19) que presenta una variación de resistencia $\frac{\Delta\rho}{\rho} = 2,2\%$ y también el NiCo (70/30) con $\frac{\Delta\rho}{\rho} = 3,7\%$. La expresión

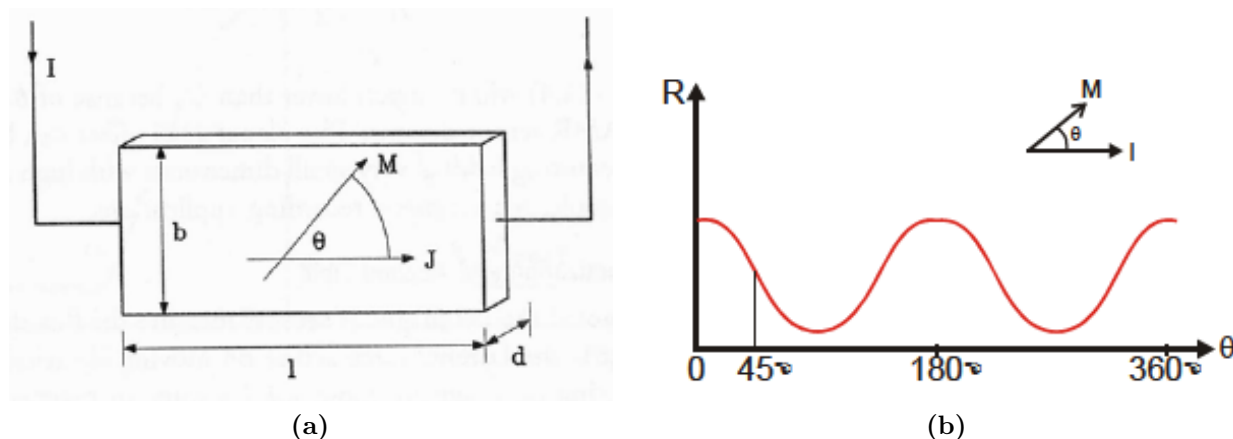


Figura 4.5 – Sensor de corriente AMR: estructura (a) y variación de la resistencia con el ángulo θ (b) [JSS09].

que relaciona la resistividad de la magnetorresistencia y el ángulo θ entre la dirección del vector de magnetización \vec{M} y la dirección de la densidad de corriente de polarización aplicada \vec{J} se da en la Ec.4.1.12.

$$\rho_{AMR}(\theta) = \rho_0 + \Delta\rho \cos^2(\vec{M}, \vec{J}) \quad (4.1.12)$$

donde ρ_0 es la resistividad mínima del material, que se obtiene cuando el vector de magnetización es perpendicular a la dirección de la corriente, ρ_{\perp} , y $\Delta\rho_m$ la variación máxima de la resistividad del material, $\Delta\rho = \rho_{\parallel} - \rho_{\perp}$ [RCBRM09], lo que permite reformular la ecuación 4.1.12 de la siguiente forma:

$$\rho_{AMR}(\theta) = \rho_{\perp} + (\rho_{\parallel} - \rho_{\perp}) \cos^2\theta \quad (4.1.13)$$

En la figura 4.5(b) se muestra la variación de la resistividad en función del ángulo θ . A partir de la definición establecida de la resistividad.

Se define el valor magnetorresistivo MR , ecuación 4.1.14, como la variación porcentual

con referencia al valor base de mínima resistencia.

$$MR = \frac{R_{MAX} - R_{MIN}}{R_{MIN}} \times 100 \tag{4.1.14}$$

Si en la magnetorresistencia de la figura 4.5.a se verifica que el espesor d es mucho menor que la anchura b y ésta a su vez menor que la longitud l , es decir $d \ll b < l$, circula una corriente constante I que establece una densidad de corriente \vec{J} en el eje \vec{x} , y teniendo en cuenta la ecuación 4.1.12, se obtiene el valor de la tensión entre los terminales de la magnetorresistencia, ecuación 4.1.15.

En la figura 4.6 se presenta configuración de equipos de medida para realizar la caracterización de las diferentes resistividades de las capas magnetorresistivas mediante el sistema de medida con cuatro puntas: dos para el generador de corriente (I) y dos para el medidor de tensión (V).

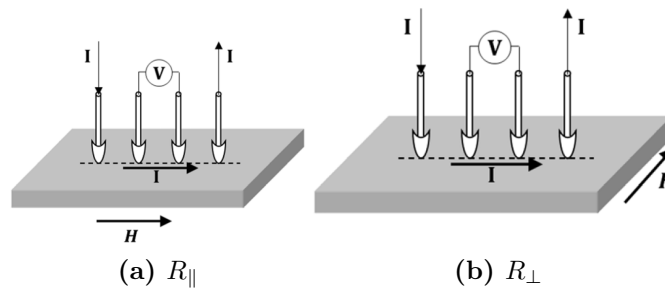
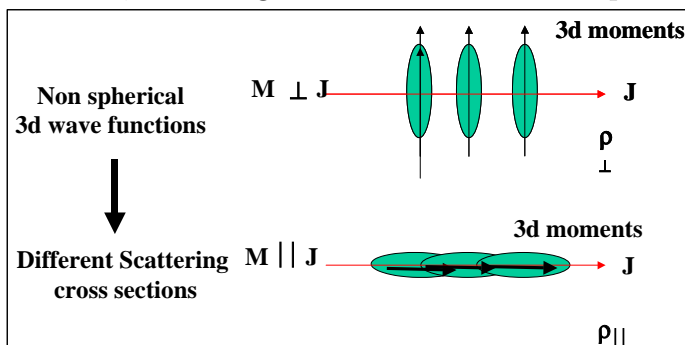


Figura 4.6 – Caracterización de las resistividades para configuración de magnetizaciones: paralela al campo (a) y perpendicular al campo (b) [Tor10].

$$V_x = \rho_{\perp} I \frac{l}{d \cdot b} \left(1 + \frac{\Delta\rho}{\rho} \cos^2\theta \right) \tag{4.1.15}$$

donde $\frac{\Delta\rho}{\rho} = \frac{\rho_{\perp} - \rho_{\parallel}}{\rho_{\perp}}$, [Phi00]. Así, el efecto magnetorresistivo se define como la dispersión (*scattering* de los electrones al interactuar con los orbitales atómicos de diferentes secciones orbitales distorsionados por el campo magnético. Esto es debido a que las funciones de onda e los orbitales $3d$ no son esféricos y por lo tanto presentan diferentes secciones de dispersión según su orientación, ver la figura 4.7. La resistencia producida por la interacción entre

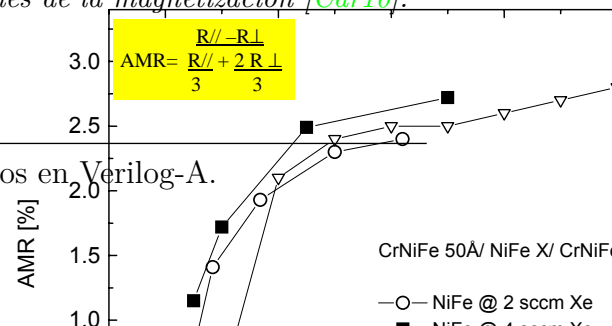
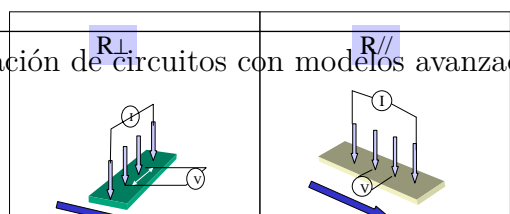
Anisotropía MR (AMR)



1st generation

Figura 4.7 – Sección del orbital 3d para diferentes orientaciones de la magnetización [Car10].

Simulación de circuitos con modelos avanzados en Verilog-A.



CrNiFe 50Å/ NiFe X/ CrNiFe

○— NiFe @ 2 sccm Xe
■— NiFe @ 1 sccm Xe

los electrones y los orbitales del material es máxima, figura 4.8.a, cuando la magnetización es paralela (i.e. 0° or 180°) a la dirección de la corriente (los orbitales tienen una mayor sección vertical en la dirección del flujo de la corriente) y mínima cuando la magnetización es perpendicular a la corriente, figura 4.8.b (los orbitales tienen una menor sección vertical en la dirección del flujo de la corriente y por lo tanto existe una menor dispersión del flujo de los electrones que implica una menor resistencia) [JSS09].

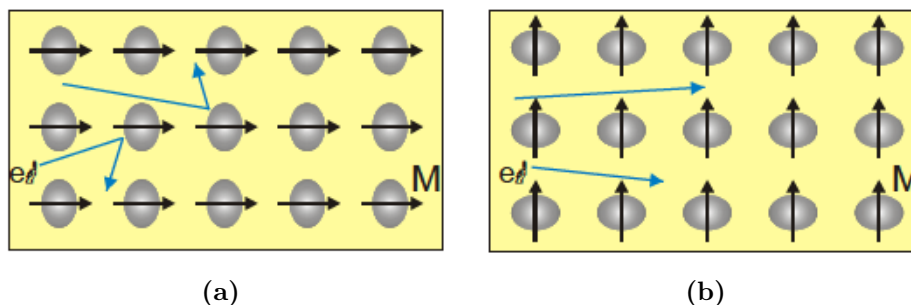


Figura 4.8 – Efecto **AMR** mostrando las diferencias en la circulación de la corriente a través de un dispositivo cuando la magnetización de las capas de hierro (amarillo) es: paralela (a) y perpendicular (b) [JSS09].

Para evitar los efectos no lineales existentes en la relación cuadrática R vs. H y conseguir linealizarla se pueden establecer dos alternativas:

- Añadir un campo magnético de desequilibrio, opción que complica de sobremanera la implementación práctica.
- Depositar sobre el material ferromagnético (*permalloy*) unas líneas conductoras de aluminio sobre la aleación con una inclinación de 45° (*barber poles*) que permiten la rotación de la corriente que atraviesa el elemento. La corriente toma el camino más corto y como puede observarse en la figura 4.9.a es perpendicular a las líneas de aluminio opuestas a las dirección de la corriente. Esto provoca un efecto de rotación de 45° en la dirección de la corriente, cambiando el ángulo del vector de magnetización relativo a la corriente de θ a $\theta - 45^\circ$ [RCBRM09]. En este caso la resistencia se puede expresar como [Phi00]:

$$R = R_0 \pm \Delta R \frac{H}{H_0} \sqrt{1 - \frac{H^2}{H_0^2}} \quad (4.1.16)$$

La microfabricación de estructuras **AMR** en la sala blanca de INESC en Lisboa descrita en [Car10] permite la obtención de dispositivos con valores de **AMR** entre el 1-3%. En la figura 4.10 se puede ver la relación existente entre el grosor nominal de la aleación NiFe y el factor magnetorresistivo del dispositivo **AMR** para dos condiciones diferentes en crecimiento de la aleación obtenidas con el sistema de deposición Nordico 3600.

4.1.1.3 Efecto magnetorresistivo gigante (GMR).

El efecto magnetorresistivo gigante fue descubierto en 1988 por Fert a bajas temperaturas y por Grünberg a temperatura ambiente [Fer08] cuando realizaban medidas de resistencia

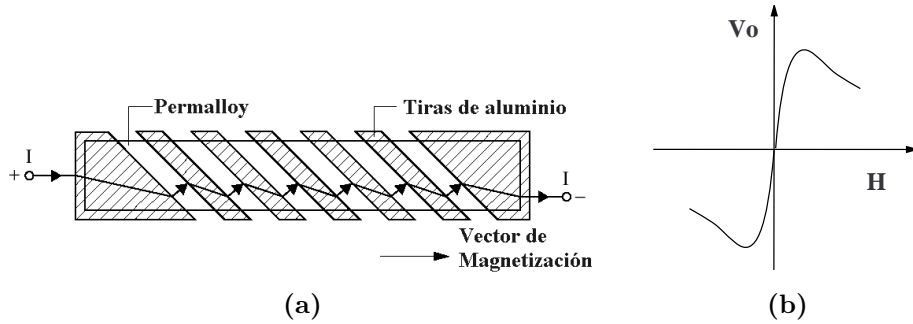


Figura 4.9 – Sensor de corriente AMR. a) Geometría de una magnetorresistencia AMR con tiras de aluminio. b) Característica V-H linealizada de un sensor AMR [PS01].

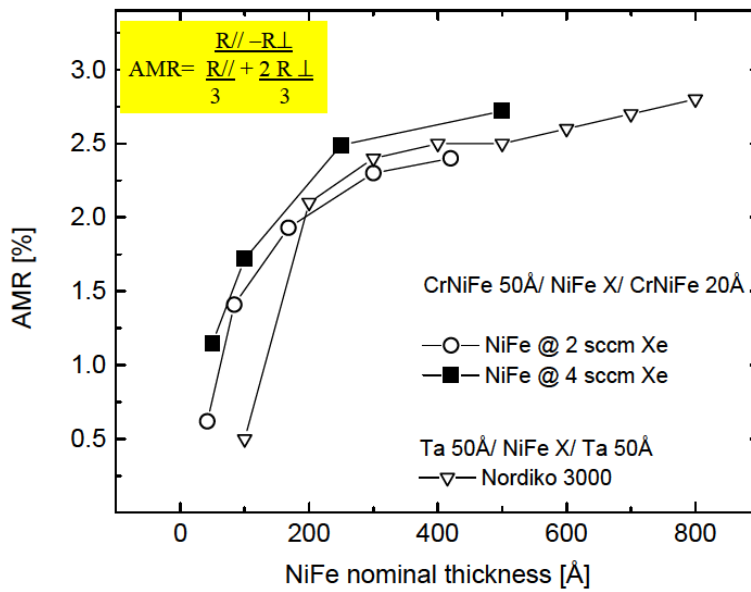


Figura 4.10 – Valores del coeficiente de magnetorresistivo para un dispositivo magnetorresistivo AMR fabricado en INESC [Car10].

eléctrica a estructuras metálicas multicapa. En 2007 recibieron el Premio Nobel de Física por la importancia del descubrimiento. La estructura de estos dispositivos está formada por multicapas de elementos metálicos de transición como por ejemplo Fe/Cr/Fe.

El sistema donde se manifiesta el efecto GMR está compuesto por capas delgadas de naturaleza ferromagnética (FM) típicamente de NiFe, CoFe o aleaciones de éstos) y capas ultra delgadas no magnéticas metálicas (NM) (Cu, Au o Ru). Su denominación de *gigante* se acuñó porque la variación de resistencia por efecto GMR era del 10-15 % que representaba mucho más del conocido hasta entonces por efecto AMR. El origen físico de este efecto es diferente al magnetorresistivo clásico, debido a que el mecanismo responsable del efecto GMR está relacionado con la dispersión de los electrones debida a la interacción de naturaleza magnética entre el *spin* de los electrones de conducción en la capa no magnética con los momentos magnéticos presentes en los materiales que constituyen las capas magnéticas adyacentes. Para obtener el efecto GMR en una estructura multicapa el espesor de la capa metálica espaciadora (NM) debe ser menor al recorrido libre medio de un electrón de

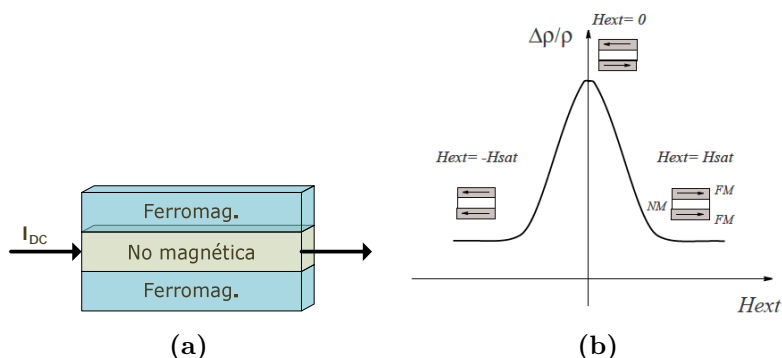


Figura 4.11 – Estructura *GMR*: sistema multicapa FM/NM/FM (a), acoplamiento FM/NM/FM (b)

manera que los electrones con un *spin* determinado en una capa (FM) puedan atravesar la capa (NM) antes de que su polarización sea variada por la dispersión. De modo esquemático las capas ferromagnéticas (FM) pueden entenderse como filtros polarizadores del *spin* de los electrones. La capa espaciadora (NM) permite que las capas ferromagnéticas (FM) posean direcciones magnéticas diferentes permitiendo el paso de electrones a su través. Cuando los momentos magnéticos de las capas ferromagnéticas (FM) están alineadas en la misma dirección, los electrones originados en una capa pueden atravesar con suficiente facilidad a través de las otras capas, ver figura 4.12(a). Si las magnetizaciones son antiparalelas los electrones generados por una capa serán bloqueados al intentar pasar a la otra adyacente aumentando el efecto resistivo. 4.12(b). El grosor de la lámina delgada no magnética (NM) que acopla

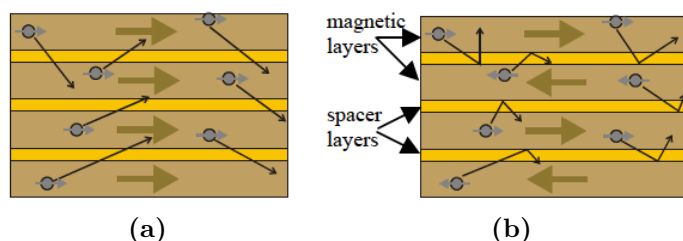


Figura 4.12 – Efecto *GMR* en estructura *CIP*, mostrando las diferencias de la dispersión de los electrones que circulan de izquierda a derecha a través de un dispositivo cuando la magnetización de las capas es: paralela (a) y antiparalela (b) [dMDM05].

las dos láminas ferromagnéticas es crítico a la hora de crear un acoplamiento antiparalelo entre las magnetizaciones de las láminas ferromagnéticas. La interacción oscila de manera periódica entre el acoplamiento ferromagnético a antiferromagnético dependiendo del grosor de la lámina intermedia no magnética, ver figura 4.13. Los dispositivos *GMR* son usados normalmente para medir corrientes que circulan en el mismo plano (*CIP*, *current-in-plane*) que las láminas magnetorresistivas. Aunque la impedancia de la magnetorresistencia es reducida debido a las corrientes paralelas (*shunt*) que circulan por las láminas, la configuración para medida de corrientes perpendiculares al plano de las láminas (*CPP*, *current-perpendicular-to-plane*) presenta una impedancia menor aun que no la hace recomendable para su uso en aplicaciones prácticas de sensorización [JSS05], ver figura 4.14.

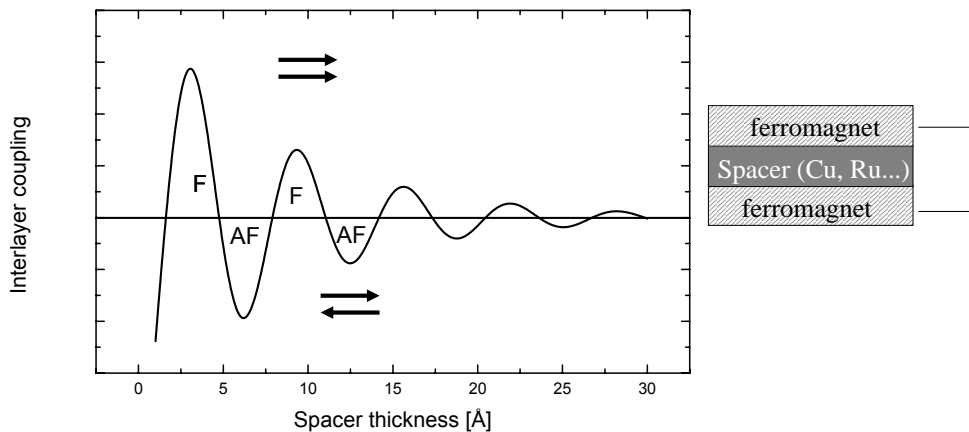
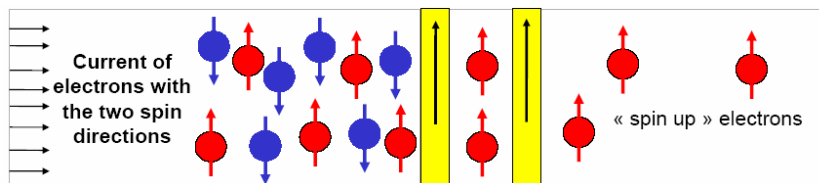
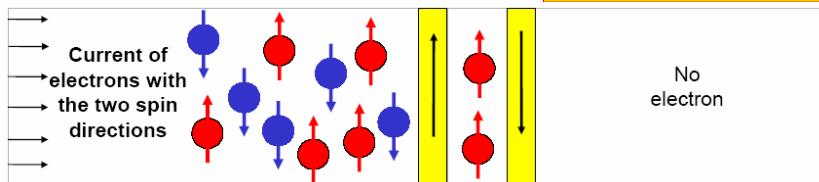


Figura 4.13 – Acoplamiento entre dos capas ferromagnéticas espaciadas por una capa ultra delgada de material no magnético. Los máximos corresponden con el acoplo ferromagnético y los mínimos con el acoplamiento antiferromagnético. Valores válidos para acoplamientos entre capas (Co/Cu), (Co/Ru) y (NiFe/Cu) [dMDM05].



(a) Magnetizaciones de las capas (FM) paralelas.



(b) Magnetizaciones de las capas (FM) antiparalelas.

Figura 4.14 – Corriente CPP atravesando una estructura tricapa GMR en con magnetizaciones: paralelas (a) y antiparalelas (b). Las flechas negras de las capas ferromagnéticas (FM) en amarillo indican la dirección de magnetización. La corriente total es debida únicamente a los electrones cuyo spin puede atravesar ambas interfaces, siendo muy pequeña en magnitud lo que desaconseja su uso en aplicaciones reales [JSS09].

A continuación se describen dos tipos de dispositivos magnetorresistivos GMR conocidos como *spin-valves* y dispositivos de efecto túnel magnetorresistivo.

4.1.1.3.1 Dispositivos *Spin-Valve*.

Las variaciones de resistencia que aparecen en una magnetorresistencia GMR, en estructuras multicapa ferromagnéticas (FM)/no-magnéticas (NM), son muy importantes en el caso CIP. Existe una versión modificada de la estructura multicapa GMR donde se utilizan únicamente dos láminas ferromagnéticas, figura 4.15.

La lámina inferior ferromagnética se deposita sobre una capa antiferromagnética (AF)

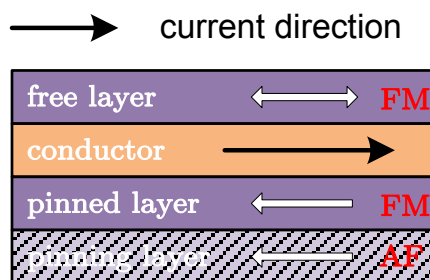


Figura 4.15 – Estructura *GMR* modificada con la introducción de una lámina antiferromagnética [RCBRM09].

denominada capa inmovilizadora (*pinning*). La capa antiferromagnética no tiene un vector de magnetización determinado debido a que su estructura está formada por celdas contiguas cuya magnetización es opuesta entre sí dando como resultante una magnetización nula. Sin embargo, tiende a mantener la magnetización de la capa ferromagnética adyacente (*pinned*) inmóvil. La otra capa ferromagnética (FM) está libre para rotar con la dirección del campo externo aplicado. Esta estructura se denomina *spin valve* (SV) que hace referencia a la forma de válvula que tendría la lámina ferromagnética superior (*free layer*) que se abriría y cerraría según el campo magnético externo aplicado, permitiendo el paso de los electrones con su *spin* polarizado a través del dispositivo. En un dispositivo SV correctamente polarizado la posición de reposo de la capa libre es perpendicular a la capa inmovilizada (*pinned*), obteniendo así el máximo de sensibilidad. La respuesta al campo magnético externo aplicado en la dirección de la capa inmovilizada es lineal dentro de un gran rango. Las estructuras SV serán tratadas con más detalle en el apartado 4.2.1.1.

El cambio que manifiesta la magnetorresistencia con la variación de la dirección del campo magnético externo se denomina MR [JSS09], ecuación 4.1.17 y se representa por la relación porcentual del cambio de la resistencia.

$$GMR (\%) = \frac{R_{\uparrow\downarrow} - R_{\parallel}}{R_{\parallel}} \times 100 = \frac{R_{max} - R_{min}}{R_{min}} \times 100 \quad (4.1.17)$$

4.1.1.3.2 Dispositivo de efecto túnel magnetorresistivo MTJ.

Las magnetorresistencias (*Magnetoresistance Tunnel Junction-MTJ*) están formadas por dos capas ferromagnéticas separadas por una capa de aislante (MgO o AlO_x), figura 4.16. En este tipo de estructuras el movimiento de los electrones de una capa ferromagnética a la otra se realiza a través del aislante por efecto túnel, lo que implica que el grosor de éste debe ser suficientemente pequeño (4-20 Å). Como la corriente que circula por el dispositivo lo hace de manera perpendicular al plano de la estructura se denomina geometría (CPP, *Current Perpendicular to Plane*). La geometría CPP de las magnetorresistencias MTJ necesita que el contacto eléctrico superior sea realizado en la parte superior de la estructura MTJ y el contacto inferior justo bajo el estructura. De modo similar a las estructuras magnetorresistivas de las SV, las MTJ disponen de una capa ferromagnética con magnetización inmovilizada (*pinned*) con el fin de establecer la dirección de referencia, mientras que la otra capa ferromagnética es libre (*free layer*) para rotar en la dirección del campo magnético ex-

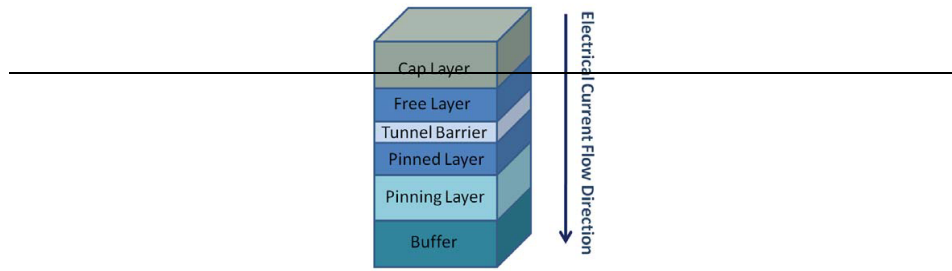


Figura 4.16 – Estructura de una magnetorresistencia *MTJ* [GdA09].

terno. Cuando las magnetizaciones son paralelas la resistencia es mínima R_{\parallel} , ver figura 4.17 y cuando lo son antiparalelas la resistencia es máxima $R_{\uparrow\downarrow}$ [GdA09]. En el caso de los dispo-

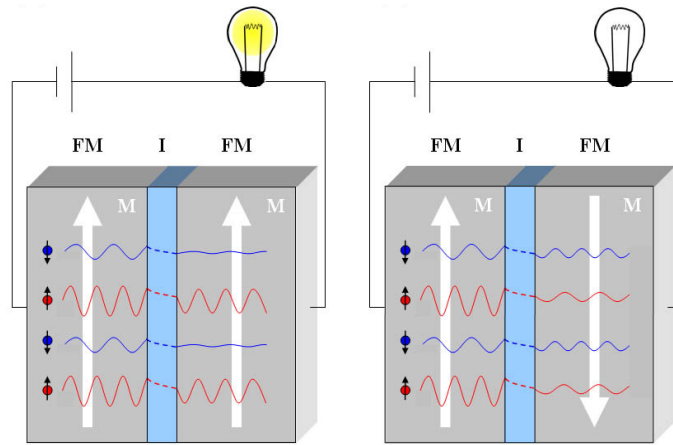


Figura 4.17 – Funcionamiento básico de una magnetorresistencia *MTJ* [GdA09].

sitivos de efecto túnel se define el parámetro TMR [GdA09], ecuación 4.1.18, que representa la relación porcentual del cambio de la resistencia en función del campo magnético externo se denomina

$$TMR (\%) = \frac{R_{\uparrow\downarrow} - R_{\parallel}}{R_{\parallel}} \times 100 = \frac{R_{\max} - R_{\min}}{R_{\min}} \times 100 \quad (4.1.18)$$

El efecto magnetorresistivo por efecto túnel fue observado por primera vez por Julliere [Jul75], que descubrió que las estructuras *MTJ* de materiales Fe/GeO/Co mostraban un coeficiente $TMR = 14\%$ a $4,2^{\circ}K$. Dado que el efecto no se manifestaba a temperatura ambiente, el descubrimiento no recibió la atención suficiente hasta que a finales de los años 80 el efecto *GMR* fue aplicado con éxito en la construcción de sensores magnéticos [BBF+88].

Comparando las magnetorresistencias *SV* y las *MTJ*, el coeficiente magnetorresistivo MR es considerablemente mayor para las *MTJ*, alcanzando valores a temperatura ambiente del 70% para aislantes AlO_x y de hasta 472% para aislante MgO. Esta diferencia se debe los mecanismos de efecto túnel de electrones con distinto *spin*.

La capa ferromagnética (FM) (*pinned*) inmovilizada por el momento magnético de la capa antiferromagnética (*pinning*) se puede sustituir por una estructura (*SAF*, *synthetic antiferromagnetic*, formada por (FM)+(NM)+(FM)) que transmite una mayor fuerza de inmovilización que una única capa (AF). La lámina *SAF* está formada por dos capas ferromagnéticas (FM) separadas por una delgada capa de material no magnético (NM). La capa

antiferromagnética (AF) inmoviliza la capa ferromagnética de la lámina SAF y ésta a su vez inmoviliza la otra ferromagnética (FM), en contacto con el aislante, mediante acoplamientos de mayor fuerza, figura 4.18. El efecto túnel dependiente del *spin* está relacionado

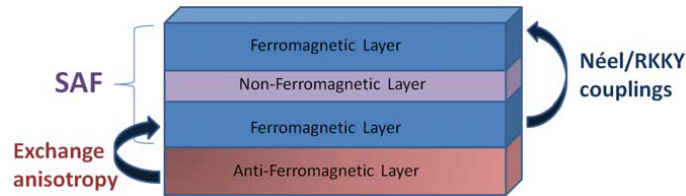


Figura 4.18 – Estructura SAF de un dispositivo magnetorresistivo MTJ [GdA09].

con la probabilidad de que los electrones puedan atravesar una barrera de material aislante por efecto túnel debido a la naturaleza ondulatoria de éstos. Aplicando una diferencia de potencial entre las dos capas ferromagnéticas separadas por una lámina aislante, se puede inducir que los electrones liberen un estado ocupado en una de las capas ferromagnéticas para ocupar un estado libre en la otra capa ferromagnética por efecto túnel. El primer modelo propuesto para explicar el efecto magnetorresistivo por efecto túnel fue Juliere en 1975 [Jul75]. Este modelo establece que el *spin* del electrón se conserva durante el proceso túnel, lo que implica que los electrones únicamente pueden pasar por efecto túnel entre estados que tengan el mismo *spin* en las dos capas ferromagnéticas. Los electrones con $spin = e_{FM1}^{\uparrow}$ (hacia arriba) solo pueden ocupar los estados vacíos con $spin = e_{FM2}^{\uparrow}$ de la otra capa y del mismo modo los electrones con $spin = e_{FM1}^{\downarrow}$ (hacia abajo) solo pueden ocupar los estados vacíos con e_{FM2}^{\downarrow} .

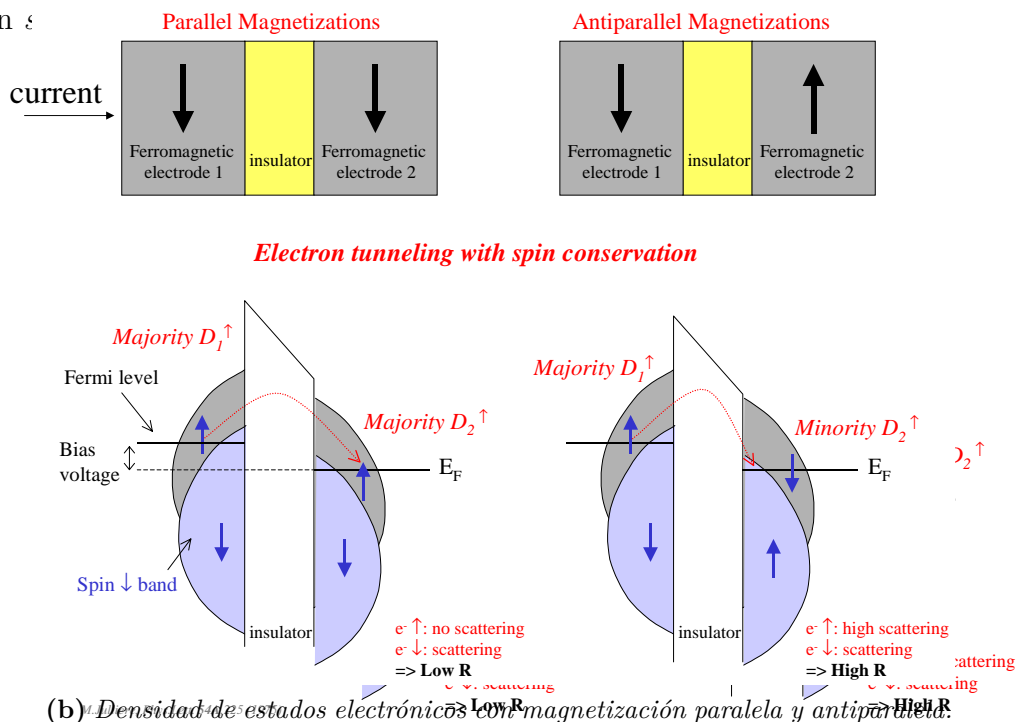


Figura 4.19 – Conducción por efecto túnel en el óxido aislante de una estructura MTJ [GdA09].

En los materiales ferromagnéticos existe una asimetría en la densidad de estados (D) para los electrones con $spin = e^{\uparrow}$, D_{\uparrow} y los de $spin = e^{\downarrow}$, D_{\downarrow} , cerca del nivel de Fermi que

orienta la magnetización de la capa en cierta dirección. Dicha asimetría se puede cuantificar con el parámetro P de cantidad de polarización por *spin* cuya expresión matemática es:

$$P = \frac{D_{\uparrow} - D_{\downarrow}}{D_{\uparrow} + D_{\downarrow}} \quad (4.1.19)$$

El efecto túnel a través del aislante es diferente según la orientación del *spin* de los electrones, ésto es lo que justifica su denominación de *spin dependent tunneling effect*. En la figura 4.19(b) se detalla el efecto túnel que aparece en el caso de magnetizaciones paralelas en las que se puede producir un tránsito por efecto túnel de electrones con $spin = e^{\uparrow}$ de una capa ferromagnética a otra con igual número de estados. Sin embargo para el caso de magnetizaciones antiparalelas los electrones con $spin = e^{\uparrow}$ son ahora minoritarios sufriendo una gran dispersión y presentando una gran resistencia eléctrica. El valor TMR representado por la ecuación (4.1.18) puede ser interpretado como:

$$TMR(\%) = \frac{J_{\uparrow\downarrow} - J_{\parallel}}{J_{\parallel}} \times 100 \quad (4.1.20)$$

donde J_{\parallel} y $J_{\uparrow\downarrow}$ representan la densidad de corriente cuando la magnetización de las capas (FM) es paralela y antiparalela respectivamente. Dado que la densidad de corriente J puede ser expresada como función de la densidad de estados de *spin* D_{\uparrow} y D_{\downarrow} en las dos capas ferromagnéticas (FM1) y (FM2), la ecuación 4.1.20 puede ser denotada como:

$$TMR(\%) = \frac{2P_1P_2}{1 - P_1P_2} \times 100 \quad (4.1.21)$$

con P_1 y P_2 representando el valor de la expresión 4.1.19 en las capas (FM1) y (FM2). La ecuación 4.1.21 se conoce como fórmula de Landée y define el TMR para un dispositivo MTJ para una polarización de los electrodos determinada sin dependencia de las características de la barrera túnel. En 1989 Slonczewski introdujo una nueva formulación para el TMR similar a la de Landée pero donde el desvanecimiento de la función de onda dentro de la barrera túnel es tenida en cuenta.

4.1.2 Aplicaciones. Medida de corriente R(I).

En este apartado se presenta el uso de magnetorresistencias como elemento sensible al campo magnético creado por una corriente que circula en una pista conductora. El uso de magnetorresistencias para esta aplicación se plantea como alternativa a métodos tradicionales de medida de corriente existentes: resistencias *shunt*, transformadores y sensores de efecto Hall. La primera convierte la corriente en tensión (ley de Ohm), los dos últimos hacen uso de relación de la corriente con el campo magnético creado por ella (ley de Ampère). Los sensores de corriente de estado sólido se basan en características intrínsecas de ciertos semiconductores para detectar el campo magnético y generar una salida proporcional. Los sensores magnetorresistivos detectan las variaciones de campo magnético creadas por la corriente que circula por la pista microfabricada por la deposición de una capa conductora,

figura 4.20. Para calcular el campo magnético creado por la corriente se puede aplicar la Ley de Biot-Savart Ec. 4.1.22

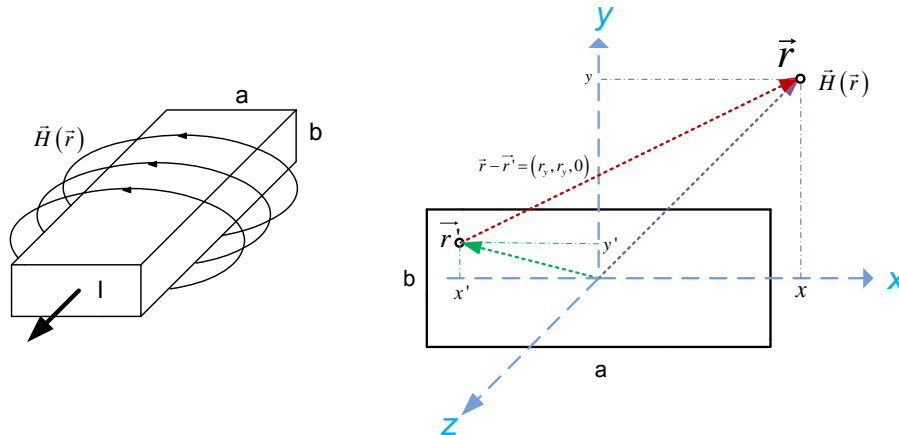


Figura 4.20 – Líneas de campo magnético [RCBRM09] generadas por circulación de una corriente a través de una pista conductora rectangular.

$$\vec{H}(\vec{r}) = \frac{1}{4\pi} \int_V \vec{J}(\vec{r}') \times \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^2} dr' \quad (4.1.22)$$

El campo magnético calculado en un punto arbitrario \vec{r} alrededor de la pista conductora de sección rectangular (anchura W y espesor h) y con una distribución de corriente uniforme de densidad $J = I/(W \cdot h)$ donde:

- $\vec{r} - \vec{r}'$ es el vector que une el punto externo a la pista donde se calcula el campo magnético \vec{r} y todos los puntos de la pista donde la densidad de corriente no es nula, \vec{r}' . Este vector está dentro del plano $(r_x, r_y, 0)$ donde $r_x = x - x'$ y $r_y = y - y'$.
- $|\vec{r} - \vec{r}'|$ es la distancia entre los puntos \vec{r} y \vec{r}' y puede sustituirse por $r = \sqrt{r_x^2 + r_y^2 + r_z^2}$.
- $\vec{J}(\vec{r}') \times (\vec{r} - \vec{r}')$ es equivalente a $(J \cdot r_y, -J \cdot r_x, 0)$

La corriente eléctrica que circula por la pista lo hace en dirección \vec{z} , figura 4.20.b. El campo magnético generado en la dirección transversal \vec{x} de un hilo infinito de sección rectangular se puede expresar haciendo uso de la ecuación 4.1.22 como:

$$\vec{H}_x = \frac{J}{2\pi} \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_{-\frac{a}{2}}^{\frac{a}{2}} \frac{y - Y}{(x - X)^2 + (y - Y)^2} dXdY \quad (4.1.23)$$

donde la densidad de corriente J puede obtenerse de la ecuación 4.1.24 siendo dA' el diferencial de superficie y $A = a \cdot b$ la sección del hilo rectangular.

$$I = \int \vec{J}(\vec{r}') \cdot \vec{n} dA' \quad (4.1.24)$$

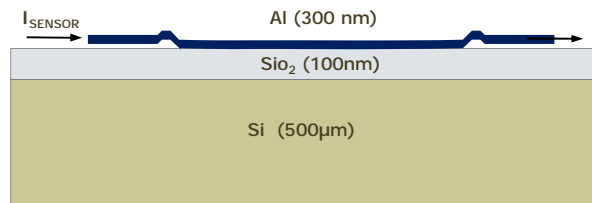
UTILIZADA EN CAP. 5

$$J = \frac{I}{A} = \frac{I}{ab} \tag{4.1.25}$$

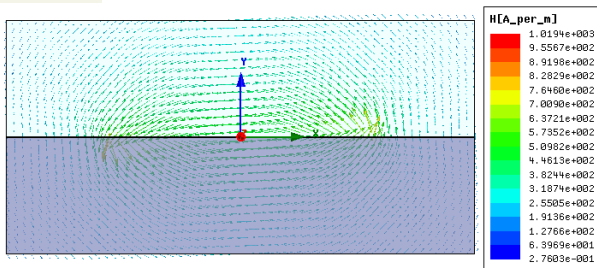
Sectional Front View



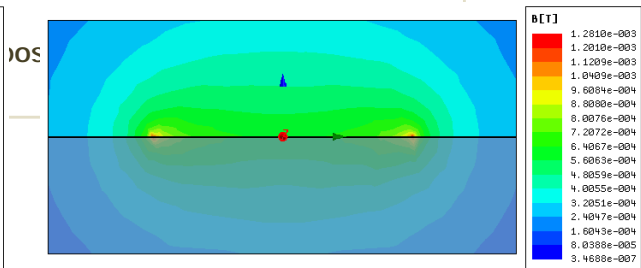
Figuras 4.21.a y 4.21.b se presenta una simulación magnetostática para una pista de aluminio de 100 μm de anchura y 300 nm de grosor depositada sobre una capa de 100 nm de silicio (SiO_2) sobre un sustrato de 500 μm de Si por la que circula una corriente de 10 mA. Las simulaciones han sido realizadas con *Maxwell 2D Designer* de Ansoft TM. La forma de campo es muy similar a la obtenida en caso de no tener en cuenta los sustratos de silicio y SiO_2 ya que las permitividades magnéticas de ambas sustancias son similares y la conductividad del SiO_2 se supone nula. En la figura 4.21(c) se puede apreciar que



(a) Sección vertical de deposición de materiales.



(b) Campo magnético.



(c) Magnitud campo magnético.

Figura 4.21 – Estructura y campo magnético generado por una pista de corriente de sección rectangular.

el campo magnético que induce la corriente que circula por la pista depositada de aluminio en la superficie del dispositivo es aproximadamente constante, por lo que se puede depositar la magnetorresistencia AMR, GMR o MTJ directamente sobre su superficie o bajo ella sin que se aprecie variación del campo magnético. Los sustratos de silicio y óxido son transparentes al campo magnético.

4.1.2.1 Comparativa de prestaciones de las diferentes tecnologías.

En la tabla 4.1, [RCBR09], se presenta un resumen de las características de los diferentes tecnologías de los sensores magnetorresistivos, figura 4.22. Cabe destacar que las diferencias más importantes se encuentran en los niveles de sensibilidad de campo magnético, que son mayores para los dispositivos de efecto túnel MTJ y que presentan como inconveniente una menor impedancia.

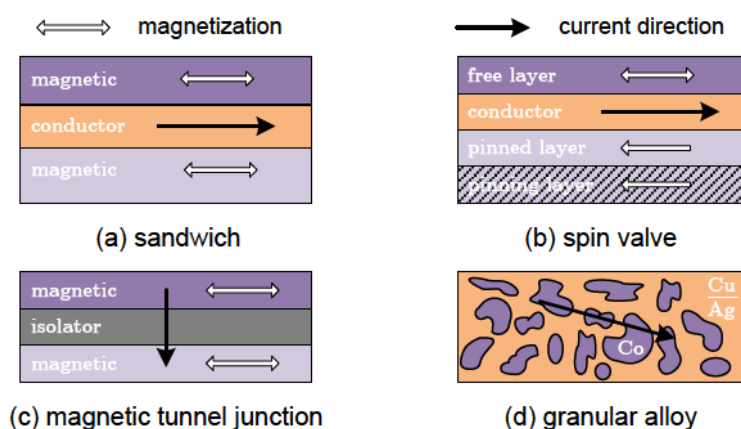


Figura 4.22 – Estructuras básicas de las diferentes tecnologías de sensores magnetorresistivos [RCBRM09].

	AMR (d)	Sandwich (a)	SV (b)	MTJ (c)
Coeficiente MR (%)	2	5-8	5-20	10-200
Campo de Saturación (Oe)	5-20	10-40	5-50	5-25
Histéresis	baja	baja	baja	baja
Sentibilidad (%/Oe)	0.4	0.5	1.0	10
Resistencia	-	2-10 Ω/\square	2-10 Ω/\square	10 – 10 ⁶ $\Omega \mu m^2$
Ancho de Banda	> 1 MHz	> 1 MHz	> 1 MHz	> 1 MHz
Deriva Térmica	0.2	0.1-0.2	0.1-0.2	< 0.1

Tabla 4.1 – Comparativa de las prestaciones de los sensores magnetorresistivos de diferentes tecnologías [RCBR09].

4.2 Desarrollo del modelo cuasi-estático

En este apartado se presenta el modelo cuasi-estático realizado para modelar los sensores magnetorresistivos de corriente tanto AMR, GMR como MTJ. En la figura 4.23 se puede observar la estructura de un sensor de corriente magnetorresistivo individual. La corriente bajo medida I_{SENSOR} se hace circular por una pista conductora que presenta baja resistencia de inserción lo que permite que el sensor en su conjunto sea no invasivo ya que no introduce elementos parásitos significativos al contrario de lo que ocurre con los sensores de corriente *shunt* y transformadores. Esta corriente, al circular por la pista, induce un campo magnético asociado, estudiado en el apartado 4.1.2 cuyas variaciones son detectadas por la magnetorresistencia MR acoplada. La configuración más habitual en la polarización de las MR es la de excitación por una corriente continua o I_{FEED} , cuyo valor ha de ser lo menor posible para evitar efectos de autocalentamiento e interferir con el campo magnético generado B objeto de la medida. En la figura 4.23(b) se establece a modo de convenio un modelo equivalente formado por un bipuerto. Como entrada o primario dispone de dos terminales (1)-(1') los asociados a la corriente de entrada I_{SENSOR} y como salida (2)-(2') para la corriente de polarización I_{FEED} . Inicialmente se establece aislamiento galvánico entre ambos puertos. La diferencia de potencial que se establece en el primario cuando es excitado por una corriente

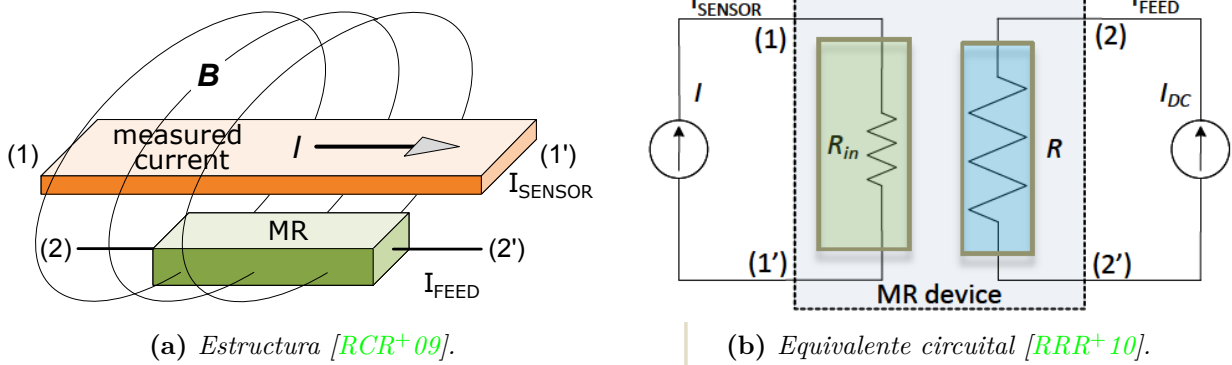


Figura 4.23 – Sensor de corriente magnetorresistivo.

se modela inicialmente por una resistencia R_{in} . Dado que la pista conductora se suele realizar mediante deposición de aluminio y el tamaño de ésta es muy corto cabe esperar que los parásitos inductivos sean despreciables. El efecto magnetorresistivo que se pone de manifiesto en las variaciones de la tensión del puerto de salida, terminales (2)-(2'), en caso de polarizarse a corriente constante $I_{FEED} = cte$ se modela en una resistencia concentrada R que depende del campo magnético inducido B por la corriente del sensor I_{SENSOR} . Aunque el efecto magnetorresistivo se presenta distribuido a lo largo de la estructura de la magnetorresistencia se puede concentrar en una resistencia equivalente de valor:

$$R = R_0 + f(B) \xrightarrow{B=g(I_{SENSOR})} R = R_0 + h(I_{SENSOR}) \quad (4.2.1)$$

Una de las ventajas que presentan los sensores de corriente magnetorresistivos es que permiten reutilizar todos los circuitos conocidos para linealización, polarización y acondicionamiento de la señal generada por éstos. La tensión que se establece en el primario verifica la expresión 4.2.2.

$$V_{(1)} - V_{(1')} = R_{in} \cdot I_{SENSOR} \quad (4.2.2)$$

del mismo modo, la relación de tensiones en el puerto de salida verifica la expresión 4.2.3.

$$V_{(2)} - V_{(2')} = R = R_0 + MR_1 I_{SENSOR} + MR_2 I_{SENSOR}^2 + \dots = R_0 + \sum_{i=1}^n MR_i I_{SENSOR}^i \quad (4.2.3)$$

Para establecer las relaciones circuitales establecidas por las ecuaciones 4.2.2 y 4.2.3 en un modelo compacto que represente al bipuerto definido por la figura 4.23.b implementado en Verilog-A es necesario utilizar las siguientes líneas de código fuente:

```

1 module magneto( np, nn, ncp, ncn );
2   inout np, nn, ncp, ncn ;
3   electrical np, nn, ncp, ncn ;
4
5   branch( ncp, ncn ) input_resistance ;

```

Listado 4.1 – Definición modelo compacto para sensor de corriente magnetorresistivo en Verilog-A.

Como se introdujo anteriormente, se definen las etiquetas necesarias para identificar los terminales del puerto de salida $(np) \rightarrow (2)$, $(nn) \rightarrow (2')$ y del puerto de control o entrada

$(ncp) \leftarrow (1)$, $(ncn) \rightarrow (1')$. Una vez establecidos los puertos del módulo, es necesario definir sus características. En la línea 2 se asignan propiedades de entrada-salida necesarias para poder realizar las simulaciones en los escenarios de uso del sensor de corriente y en la línea 3 se definen como señales eléctricas por lo que deberán verificar las leyes de Kirchhoff en los terminales. Más adelante en el modelado térmico, será necesario hacer uso de señales de naturaleza no-eléctrica para modelar los nodos térmicos internos del modelo. Para implementar de manera cómoda la ecuación 4.2.2 se puede establecer una rama interna que denominaremos *input_resistance* entre los nodos (ncp) y (ncn) del primario. A continuación, listado 4.2, se establecen los parámetros que al inicio de la simulación serán entregados por el simulador al modelo compacto.

```

1 parameter real Rin = 7.0 from [0:inf]; // Resistencia de la pista de entrada de corriente
  de control [Ohm]
2
3 parameter real R0 = 1000.0 from (0:inf); // Magnetorresistencia intrínseca [Ohm]
4
5 // Coeficiente de Magnetorresistencia
6 parameter real MR1 = 0.0 from [-inf:inf]; // Coeficiente de Magnetorresistencia [Ohm/Amp]
7 parameter real MR2 = 0.0 from [-inf:inf]; // Coeficiente de Magnetorresistencia [Ohm/Amp^2]
8 parameter real MR3 = 0.0 from [-inf:inf]; // Coeficiente de Magnetorresistencia [Ohm/Amp^3]

```

Listado 4.2 – Definición de los parámetros del modelo.

El valor de la resistencia R_{in} de la línea 2 se establece por defecto a 7.0Ω y el intervalo permisible de su valor se define desde $[0, +\infty]$. Para evitar los problemas en el caso de que $R_{in} = 0 \Omega$ se establecerán controles adicionales. La resistencia intrínseca de la magnetorresistencia, a temperatura T_0 , para un valor de corriente $I_{SENSOR} = 0 A$ se define en el parámetro $R0$ en la línea 3. Los coeficientes de la magnetorresistencia: lineal $MR1$, y no lineales $MR2$ y $MR3$ aceptan cualquier valor real.

```

1 if (Rin == 0) begin
2   V(input_resistance) <+ 0;
3 end
4 else begin
5   I(input_resistance) <+ V(input_resistance) / Rin;
6 end

```

Listado 4.3 – Tensión Primario.

En el listado 4.3 se establece la relación entre la tensión en el puerto de entrada y la corriente de excitación I_{SENSOR} . Haciendo uso de una cláusula *if/else* se puede definir el comportamiento en el caso de que $R_{in} = 0 \Omega$. Con esta solución se evita evaluar un cociente con denominador nulo que generaría una excepción de división por cero.

```

1 // Resistencia_Total [Ohms] *****
2 analog function real Resistencia_Total;
3 input II ,VV, R0,MR1,MR2,MR3;
4 real II ,VV, R0,MR1,MR2,MR3;
5 begin
6   Resistencia_Total= R0+MR1*II+MR2*II*II+MR3*II*II*II;
7   if (Debug) $strobe("Resistencia_total=%3.2e",Resistencia_Total,I_V);
8 end
9 endfunction
10 //FIN: Resistencia_Total
11 Magnetorresistencia=Resistencia_Total(I(input_resistance),V(np, nn),R0,MR1,MR2,MR3);
12 I(np, nn) <+ V(np, nn)/Magnetorresistencia;

```

Listado 4.4 – Tensión Secundario.

Los efectos de la magnetorresistencia en el puerto de salida se modelan haciendo uso de la expresión 4.2.3. La implementación en Verilog-A se presenta en el listado 4.4 donde se utiliza de una función externa que calcula el valor de la resistencia a partir de la corriente de primario II , la resistencia intrínseca $R0$ y los coeficientes magnetorresistivos $MR1$, $MR2$ y

MR3. Se determina la contribución de la corriente entre los terminales (np) y (nn) aplicando la Ley de Ohm.

Una vez desarrollados los elementos necesarios para representar el sensor de corriente de la figura 4.23(a) mediante el modelo circuital de la figura 4.23(b), codificado en Verilog-A, se procede a la generación del módulo compilado para su uso en diferentes simuladores de circuitos. En la figura 4.24 se describe el diagrama de flujo del proceso de compilación del

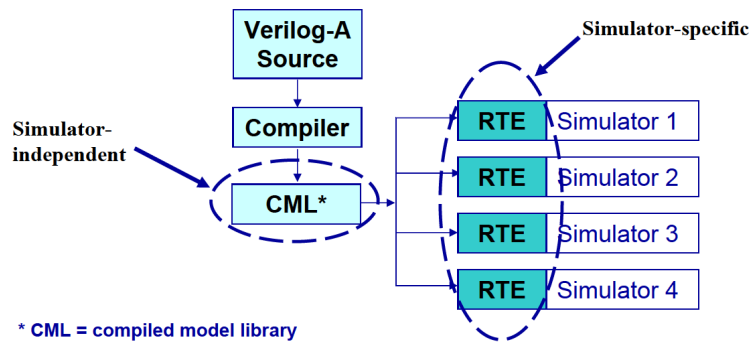


Figura 4.24 – Esquema de compilado y simulación mediante CML [MOT09b].

modelo compacto según la plataforma desarrollada por Tiburon Design Automation que se ha descrito en el apartado 1.8.2.

4.2.1 Elementos magnetorresistivos GMR individuales.

La existencia de sensores de corriente magnetorresistivos GMR de diferentes tecnologías motiva la particularización del modelado de dichos sensores para cada caso. Se va introducir cada tecnología y se va a proceder a la caracterización de los sensores en un banco de medidas y cámara térmica para posteriormente modelar los resultados experimentales y contrastarlos con los resultados de simulación.

4.2.1.1 Dispositivos Spin-Valve.

Una estructura spin-valve está basada en una región activa GMR, que fue introducida en el apartado 4.1.1.3 y está constituida por dos capas ferromagnéticas (FM1) y (FM2) separadas por una capa no magnética (NM). El vector de magnetización de una de las capas ferromagnéticas se fija en una dirección particular (capa fija o inmóvil - *pinned* - P), a través de un acoplamiento de intercambio directo con una capa antiferromagnética (AF) que se añade contigua, figura 4.25. El vector de magnetización de la otra capa ferromagnética queda libre (capa libre - *free* - F) para poder girar si se aplica un campo magnético externo. La magnetización de la capa FM2 está inmóvil (*pinned*) por el acoplamiento existente con la capa antiferromagnética, figura 4.15. En la capa FM1 (*free*) no existe este acoplamiento y el vector de magnetización puede girar libremente. Se utiliza normalmente una capa separadora conductora de cobre cuyo grosor se debe mantener por debajo de los 20 Å, figura 4.13.

La variación de la resistencia de la estructura SV depende del ángulo entre el vector de magnetización de la capa libre (θ_{libre}) y el de la capa inmóvil ($\theta_{inmóvil}$) siguiendo la

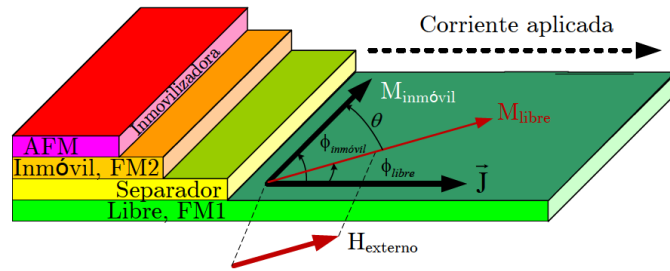


Figura 4.25 – Ángulos de los vectores de magnetización de las capas libre (free-FM1) e inmóvil (pinned-FM2) referidos a la dirección de la corriente aplicada \vec{J} .

relación 4.2.4 en la que se han eliminado los términos $\frac{\Delta R_m}{R}$ de segundo orden y superiores; esta aproximación es aceptable para valores magnetorresistivos que están por debajo del 10% [Hei93].

$$\delta R \approx -\frac{\Delta R_m}{R} \left(R_s \frac{w}{h} \right) \frac{\cos(\theta_{\text{libre}} - \theta_{\text{inmóvil}})}{2} \quad (4.2.4)$$

donde $\frac{\Delta R_m}{R} = (R_{\uparrow\downarrow} - R_{\uparrow\uparrow}) / R_{\uparrow\uparrow}$ es el valor magnetorresistivo conocido como MR del dispositivo SV presentado con anterioridad en la ecuación 4.1.17, R_s es la resistencia superficial definida por $R_s = \frac{\rho}{t}$, siendo ρ la resistividad y t la altura de la estructura SV y se mide en Ω/\square , La ecuación 4.2.4 puede ser expresada también como 4.2.5 haciendo uso de las equivalencias existentes entre $R = \rho \frac{w}{ht}$ y $R_{SV} = R_0 + \Delta R$:

$$\delta \rho_{SV} = \rho \left(1 - \frac{MR}{2} \cos(\theta_{\text{libre}} - \theta_{\text{inmóvil}}) \right) \quad (4.2.5)$$

La relación establecida entre el campo magnético externo aplicado en la dirección θ_{free} y la variación de la resistencia expresada en términos de MR se presenta gráficamente en la figura 4.26 La estructura interna de un dispositivo (SV) como los utilizados en el presente

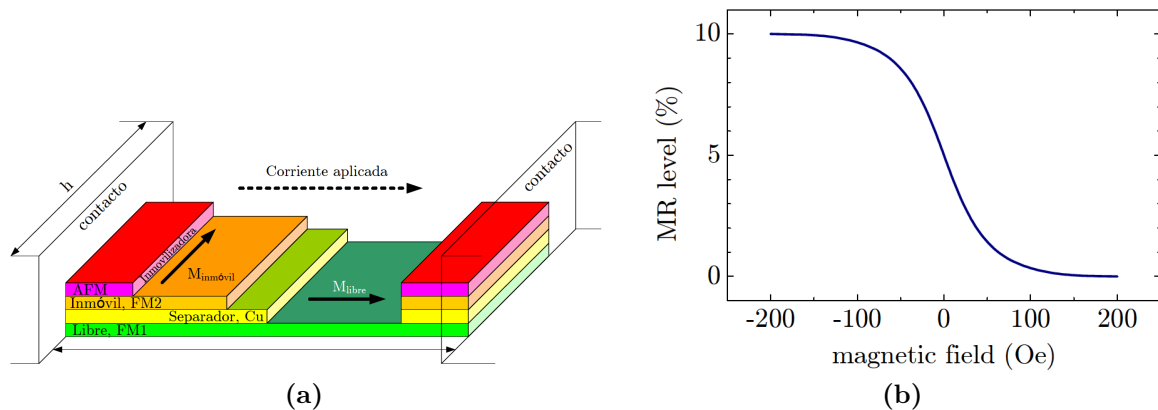


Figura 4.26 – Sección vertical de un dispositivo magnetorresistivo SV (a) y respuesta del dispositivo (b) [RCBR09].

trabajo se muestra en la figura 4.27. En este caso, la capa inmovilizadora (pinning) de MnIr ha sido depositada sobre las láminas GMR, pero también hubiera sido posible situarla en la zona inferior como en la figura 4.15. Las condiciones de crecimiento de las capas

SENSORS LAYOUT

SPIN VALVE

ION BEAM DEPOSITION

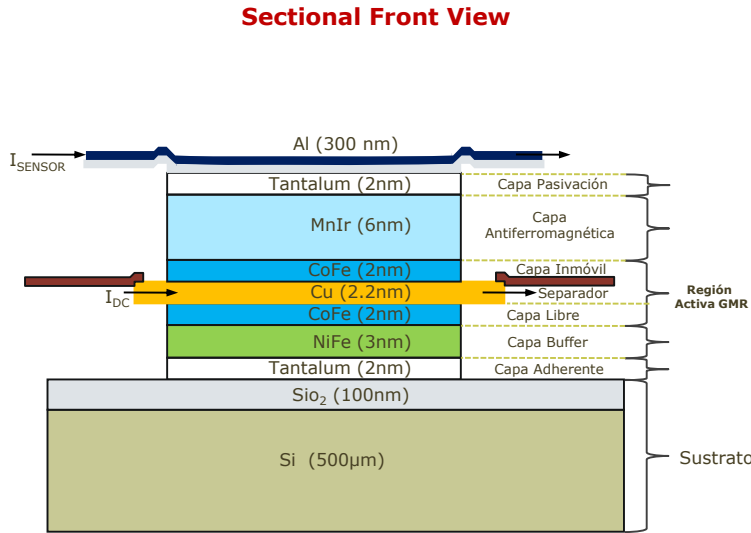
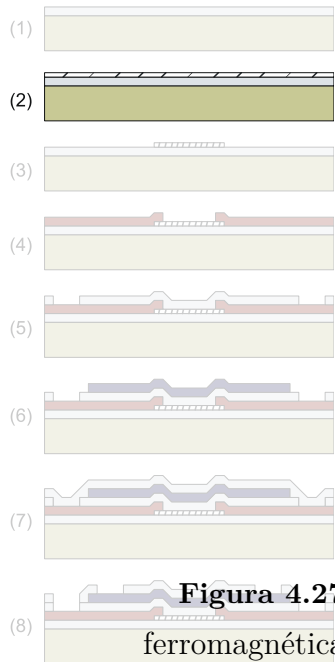


Figura 4.27 – Diferentes capas que forman el dispositivo magnetorresistivo SV [RRR+ 11b].

ferromagnéticas de naturaleza ligeramente diferentes:

Spin valve structure deposition

- silicon
- silicon oxide
- spin valve
- aluminium

de Ta
o SiO₂
ma se
capas

1. Una capa (*free*) donde el vector de magnetización puede girar libremente con el campo externo que está formada por una capa CoFe (20 Å) y otra capa de NiFe (30 Å). Esta composición permite aumentar el valor MR, además con la introducción de la capa de CoFe se evita que haya una definición incorrecta de las interfaces debido a que el Cu y el NiFe son miscibles.
2. La otra capa ferromagnética de CoFe (20 Å) cuyo vector de magnetización está inmovilizado (*pinned*) por un campo de acoplamiento con la capa superior de material antiferromagnético MnIr (60 Å). De esta forma se consigue la inmovilización del vector de magnetización, manteniendo el principio de funcionamiento de la estructura *spin-valve*, siempre y cuando no se superen campos magnéticos externos del orden del campo de acoplamiento, aproximadamente unos 300 Oe, valor a partir del cual la capa dejaría de estar inmóvil.

Existe un acoplamiento de intercambio indirecto entre las dos capas ferromagnéticas mencionadas que se pretende que sea débil para que el vector de magnetización de la capa libre pueda girar sin ningún tipo de campo que lo impida; de esta forma la fuerza de este acoplamiento está controlada por el espesor de la capa separadora no magnética de Cu (25 Å), ver figura 4.13. La influencia de este espesor no desaparece con la intensidad del acoplamiento, ya que también es responsable de la variación del valor MR del dispositivo. En la parte inferior, justo encima del óxido de silicio y en el techo de la estructura SV, se depositan sendas capas de Ta (20 Å), capa de color blanco en la figura 4.27, actuando como capas de pasivación, las cuales previenen la oxidación, minimizan la rugosidad y mejoran la adherencia.

SENSORS LAYOUT

PASSIVATION LAYER DEPOSITION AND PADS OPENING

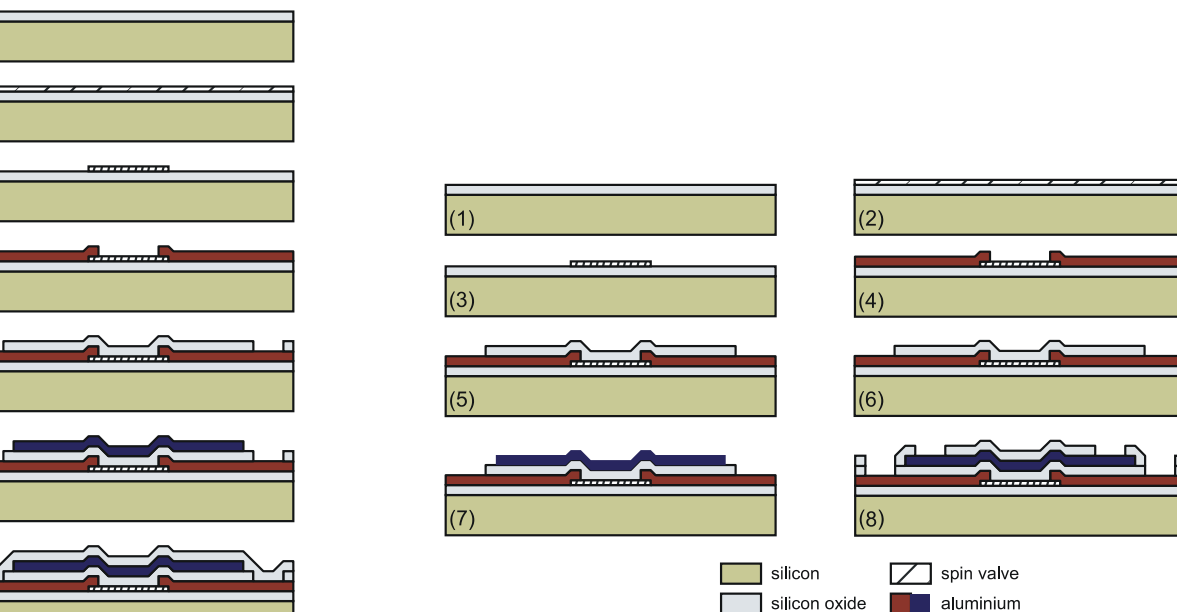


Figura 4.28 – Etapas del proceso de fabricación de los dispositivos magnetorresistivos SV realizado en INESC Lisboa [RCR⁺09].

Las capas de Ta y NiFe sirven para mejorar la textura del cristal. La estructura magnetorresistiva simple se forma con las dos capas de material ferromagnético de CoFe y la espaciadora de cobre. El vector de magnetización de la capa ferromagnética superior se inmoviliza en una dirección fija usando la capa (*pinning*) de material antiferromagnético (MnIr). La velocidad de deposición mediante *sputtering* varía entre 0.3-0.6 Å/s.

Durante el proceso de deposición se aplica al sustrato un campo de 40 Oe para establecer los vectores de magnetización de las capas libre (*free*) e inmóvil (*pinned*). La oblea se rota 90° entre la deposición de las capas libre e inmóvil para establecer la configuración cruzada del dispositivo *spin-valve*. De esta manera, se conseguirá un efecto de histéresis bajo [CBRR⁺09] ya que la capa libre nunca se invertirá, simplemente rotará 90°.

El proceso de microfabricación del dispositivo consiste en cinco procesos litográficos consecutivos. El primero de ellos, figura 4.28(3), forma el dispositivo SV sobre un área activa de 200 μm × 3 μm mediante litografía láser sobre el material fotosensible. A continuación se realiza un proceso de aislado de las zonas donde se retiran las capas inferiores mediante *Ion Milling*. Los contactos del dispositivo SV son formados mediante la técnica de *lift-off* usando una capa conductora de 150 nm de Al_{98,6}Si_{1,0}Cu_{0,4}, figura 4.28(4), y una de pasivación de

150 nm TiW(N₂) depositada por *sputtering*. A continuación se deposita una capa de aislante SiO₂ de espesor 1500 Å, figura 4.28(5), como aislante eléctrico entre el dispositivo y las pistas de corriente que son depositadas posteriormente por *sputtering* con un grosor de 250 nm Al_{98,6}Si_{1,0}Cu_{0,4} y una nueva capa de pasivación para evitar problemas de corrosión y oxidación de 15 nm TiW en atmósfera de (N₂) y definidas mediante *lift-off*, figura 4.28(7). La capa final consiste en un aislante final de Al₂O₃ de grosor 1500 Å, figura 4.28(8), donde se definen y abren los *pads* mediante un atacado húmedo (*wet-etch*).

En la figura 4.29 se muestran los cuatro tipos distintos de elementos magnetorresistivos que se van a utilizar, con una estructura como figura 4.27. En la figura 4.29 se muestran las distintas anchuras de pista. Estos dispositivos fueron integrados en un *chip* de 40 pines utilizando la técnica de *wire-bonding*.

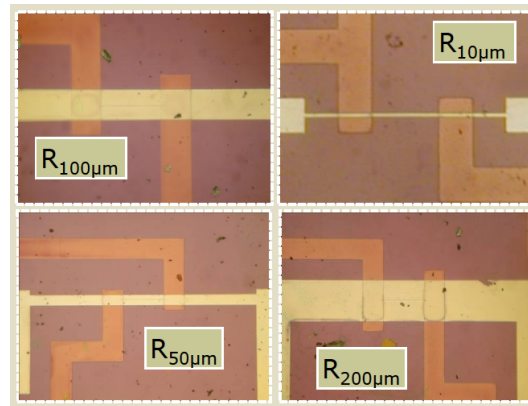


Figura 4.29 – Vista superior del dispositivo de $8,2 \times 8,2$ mm que contiene 4 magnetorresistencias SV[RCR+09].

4.2.1.1.2 Caracterización en corriente continua.

El proceso de caracterización se realizó una vez el chip había sido encapsulado. El procedimiento de medida se describe en la figura 4.30 donde mediante la fuente de corriente se hace circular una corriente triangular I_{SENSOR} de ± 10 mA por los terminales (1)-(1') y con un multímetro se mide la magnetorresistencia promediada SV entre los terminales (2)-(2'), obteniendo los valores de R_{in} y $R(I_{\text{SENSOR}})$ del modelo presentado con anterioridad en la figura 4.23. El procedimiento de caracterización de los parámetros necesarios para el modelo de corriente cuasiestática¹⁴ del listado 4.4 a partir de los datos experimentales consta de dos fases. La primera consiste en la obtención de una aproximación lineal de la relación $R = f(I_{\text{SENSOR}})$ que será suficiente en caso de que el ajuste tenga un error menor de 4%. En caso de superarse el error se aborda la segunda fase en la que se procede con un ajuste cúbico que garantiza el éxito en todos los casos estudiados. En la figura 4.31 se muestran los resultados experimentales (símbolos) y en línea continua los obtenidos del modelo lineal aplicado (curvas simuladas con ADS® de Agilent®) y cuyos coeficientes se presentan en la tabla 4.2. Los valores experimentales corresponden a las cuatro magnetorresistencias individuales existentes en las esquinas exteriores del chip de la figura 4.29. Los dispositivos magnetorresistivos son idénticos y varía únicamente la anchura de la pista de corriente por la que circula

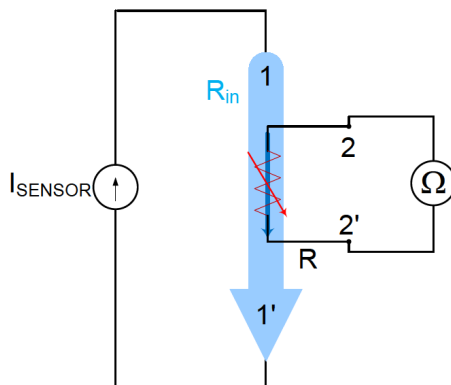


Figura 4.30 – Procedimiento de medida para la caracterización la magnetorresistencia de un dispositivo *SV* versus I_{SENSOR} .

I_{SENSOR} , figura 4.23(a). Todas las gráficas tienen pendiente positiva salvo la relativa a la magnetorresistencia R_4 de $100 \mu\text{m}$ en la que el generador de corriente se conectó con la polaridad contraria.

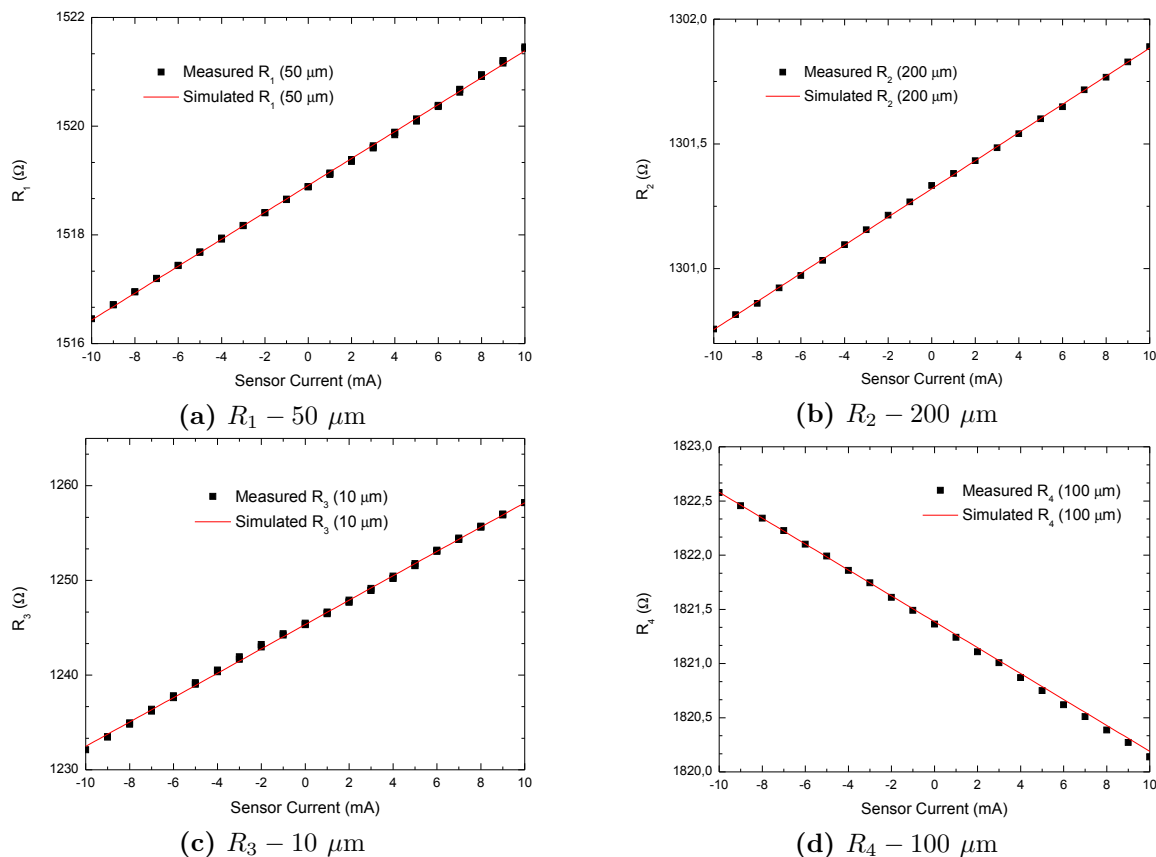


Figura 4.31 – Caracterización de la magnetorresistencia de los dispositivos *SV* versus I_{SENSOR} .

La linealidad que presentan los sensores magnetorresistivos es bastante alta para el régimen de corrientes para los que han sido diseñados ($I_{\text{SENSOR}} = \pm 10 \text{ mA}$) como se puede observar en la gráfica 4.31. En la tabla 4.2 se han incluido los parámetros del modelo para simulación tanto en el caso lineal como cúbico [RRR⁺10], pero debido a la linealidad del

sensor ($MR_2 = MR_3 = 0$), no es necesario su uso en el simulador para obtener un buen ajuste de los datos experimentales y simulados [RCR⁺09].

		$R_0(\Omega)$	$MR_1(\Omega/A)$	$R_0(\Omega/A)$	$MR_1(\Omega/A)$	$MR_2(\Omega/A^2)$	$MR_3(\Omega/A^3)$
10 μm	R_3	1245.34	1286.57	1245.42	1237.39	-2353.05	818609.51
50 μm	R_1	1518.90	247.68	1518.88	241.98	758.39	83187.94
100 μm	R_4	1821.38	119.54	1821.38	-118.94	-82.53	-8875.37
200 μm	R_2	1301.36	-60.97	1301.36	63.57	-171.80	-39942.54

Tabla 4.2 – Parámetros del modelo compacto para magnetorresistencias SV tanto para comportamiento lineal como para no lineal [RRR⁺11e].

En la figura 4.32 se presentan los resultados de simulación (línea continua) y experimentales (símbolos) de un caso en que el sensor magnetorresistivo se ha excitado con una corriente I_{SENSOR} de forma triangular de amplitud ± 10 mA y una corriente de polarización de la SV de 1 mA, donde se adquieren los datos de tensión de salida sobre la SV. El correcto ajuste entre los datos experimentales y las simulaciones en este análisis transitorio demuestran la capacidad del modelo diseñado para uso en aplicaciones circuitales reales.

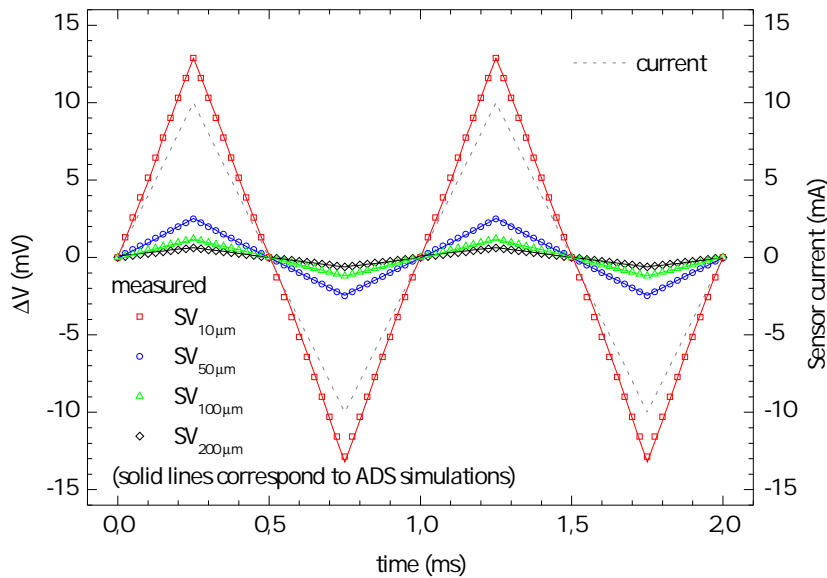


Figura 4.32 – Tensión de salida de los dispositivos magnetorresistivos SV con una corriente de polarización $I_{\text{FEED}} = 1$ mA excitados por una corriente $I_{\text{SENSOR}} = \pm 10$ mA triangular [RRR⁺11e].

4.2.1.1.3 Caracterización térmica estática.

La caracterización térmica de los sensores de corriente es muy importante sobre todo cuando la magnitud de la corriente es elevada porque implica variaciones importantes de las condiciones de trabajo del sensor debido al efecto Joule. Los efectos de calentamiento térmico inherentes dentro del chip cuando la corriente a medir, I_{SENSOR} , circula por las pistas superiores, deben tenerse en cuenta en el proceso de modelado. Para proceder correctamente es necesario realizar la caracterización térmica estática del sensor, en este sentido es necesario

el uso de una fuente de señal , una fuente de alimentación , un voltímetro , un termómetro resistivo y un sistema de adquisición multiplexado . Las medidas han sido realizadas en la cámara climática (ACS-CH600) por el Dpto. de Ingeniería Electrónica de la Universidad de Valencia. La cámara permite rangos de temperatura desde -20°C a 60°C . Con el uso del sistema de adquisición multiplexado se puede realizar la medida de varios dispositivos al mismo tiempo una vez que la temperatura se ha estabilizado. La corriente del sensor I_{SENSOR} se hace variar realizando una adquisición promediada. Una aplicación de instrumentación virtual realizada en *LabView* existente en el laboratorio del Dpto. de Ingeniería Electrónica de la Univ. de Valencia registra todas las medidas obtenidas.

Para poder incluir los efectos térmicos en el modelo compacto es necesario establecer las relaciones entre los parámetros del modelo de interés y la temperatura. La ecuación 4.2.6 establece el comportamiento térmico del parámetro magnetorresistivo y su efecto descrito por las contribuciones ΔR_T y ΔMR_T .

$$R = R_{RT,0} + MR_{RT}I_{\text{SENSOR}} + (\Delta R_T + \Delta MR_T I_{\text{SENSOR}})T \quad (4.2.6)$$

donde se han introducido estos dos parámetros para la caracterización térmica estática asumiendo una dependencia lineal con la temperatura [RRR+11e]. En la figura 4.33 se presentan

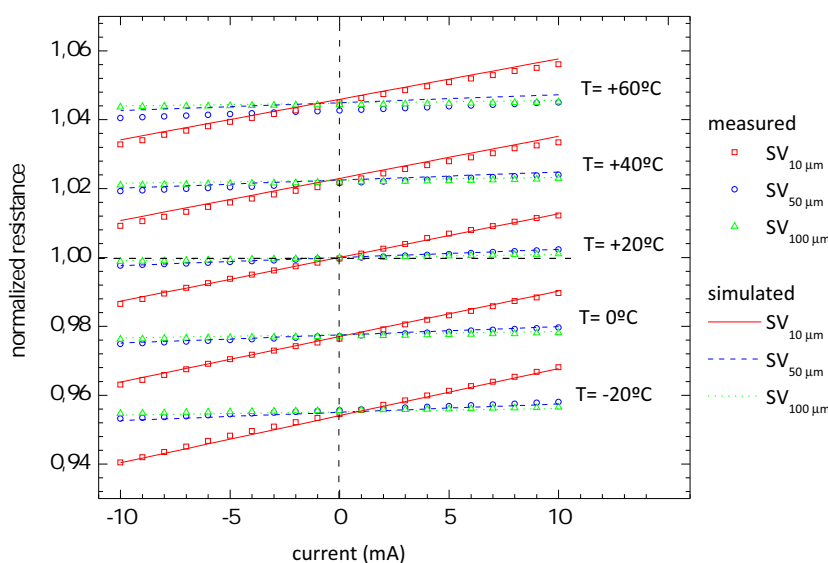


Figura 4.33 – Magnetorresistencia normalizada de los dispositivos *SV* para temperaturas ($-20, 0, 20, 40, 60^{\circ}\text{C}$) versus corriente $I_{\text{SENSOR}} = (-10, 10 \text{ mA})$. Datos experimentales en símbolos y simulación en líneas [RRR+11e].

los resultados experimentales obtenidos durante la medida realizada en la cámara térmica. Los parámetros del modelo térmico estático fueron extraídos mediante un proceso similar al utilizado en el apartado anterior y se muestran en la tabla 4.3. Estos datos permiten reproducir mediante simulación los datos experimentales de la figura 4.33. Como puede observarse, existe una linealidad razonable dentro del rango de temperaturas considerado. Además, se obtiene un incremento de la resistencia con la temperatura pero la variación del parámetro MR es muy pequeña. Para clarificar este punto se presentan los valores normalizados a $T = 20^{\circ}\text{C}$ y $I_{\text{SENSOR}} = 0 \text{ mA}$ de la magnetorresistencia R y el factor MR en función de

Device	$R_{RT,0}$ (Ω)	ΔR_T ($\Omega/^\circ\text{C}$)	MR_{RT} (Ω/A)	ΔMR_{RT} ($\Omega/\text{A}/^\circ\text{C}$)
$SV_{10\ \mu\text{m}}$	1105.13	1.268	1400	-2.68
$SV_{50\ \mu\text{m}}$	1153.50	1.295	271.77	-0.148
$SV_{100\ \mu\text{m}}$	1081.43	1.210	97.0	-0.196

Tabla 4.3 – Parámetros térmicos del modelo de magnetorresistencia para dispositivos SV [RRR⁺11e].

la temperatura, figura 4.34. La figura anterior pone de manifiesto que la resistencia varía

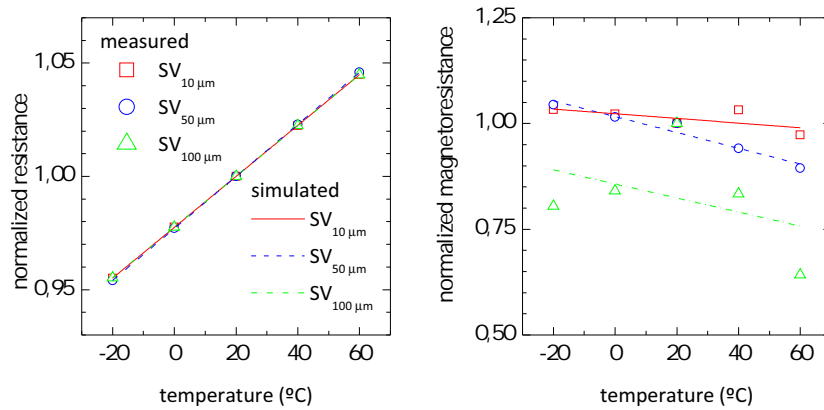


Figura 4.34 – Resistencia y magnetorresistencia normalizada de los dispositivos $SV_{10\ \mu\text{m}}$, $SV_{50\ \mu\text{m}}$, $SV_{100\ \mu\text{m}}$ versus temperatura. Datos experimentales en símbolos y simulación en líneas [RRR⁺11e].

linealmente con la temperatura mientras que la magnetorresistencia presenta un comportamiento aparentemente lineal pero que difiere conforme la anchura de la pista de corriente aumenta.

En el listado 4.5 se muestra el código introducido en el modelo de Verilog-A para incluir los efectos térmicos estáticos. En caso de que exista una diferencia entre la temperatura de simulación y la temperatura de medida de los parámetros se realiza una corrección de los parámetros R_{0T} , MR_1 , MR_2 , MR_3 , P_{1T} , P_{2T} y P_{3T}

```

1  `define SCALE_T_LINEAR_REL(x, s) (x+s*delta_T)
2
3  // Temperature effects
4  T = $temperature; //Temperatura [Kelvin] del dispositivo
5  T_nom = Tnominal + `P_CELSIUS0; //Temperatura [K] de medida de parametros
6
7  //acoplamiento térmico
8  delta_T = T - T_nom;
9  @(initial_step) begin
10 if (DEBUG) $strobe("T=%g   temperature=%g   T_nom=%g",T,$temperature,T_nom);
11 end
12
13 //Actualizamos el valor de los parámetros desde la temperatura de medida a la temp. del
   dispositivo
14 if (delta_T > 0) begin
15   R0_T =SCALE_T_LINEAR_REL(R0, TC_R0);
16   MR1_T =SCALE_T_LINEAR_REL(MR1, TC_MR1);
17   MR2_T =SCALE_T_LINEAR_REL(MR2, TC_MR2);
18   MR3_T =SCALE_T_LINEAR_REL(MR3, TC_MR3);
19   P1_T =SCALE_T_LINEAR_REL(P1, TC_P1);
20   P2_T =SCALE_T_LINEAR_REL(P2, TC_P2);
21   P3_T =SCALE_T_LINEAR_REL(P3, TC_P3);
22 end
23 else begin
24   R0_T =R0;
25   MR1_T =MR1;
26   MR2_T =MR2;
27   MR3_T =MR3;
28   P1_T =P1;
29   P2_T =P2;

```

of the MR is appreciated, and order of magnitude of the MR as a function of the temperature are plotted in Fig. 7. In this figure it is demonstrated that while the resistance is a linear function of the temperature, the magnetoresistance slightly moves off from this behavior. With respect to this, a deeper study should be required to shed light on this issue.

For reproducing the heating produced by the Joule effect in the current strips, we have made use of the so-called Foster normal form of the thermal one-port [11,12], sketched in Fig. 8. In this circuit topology, the time constants take the form of $R_{th}C_{th}$. The static heat transfer from the active area of the sensor to the ambient is subdivided in two processes in series: from device to package, modelled by the thermal resistance, R_{th_dp} , given in K/W, whose value is connected with a technological characteristic specific to fabrication and encapsulation procedures; and from package to ambient, modelled by the thermal resistance R_{th_pa} , the thermal package–ambient air resistance depends not only on the size, form

of the device but also on its orientation, and the air flow next to the package. Each resistance has been linked to a thermal capacitance (C_{th_dp} and C_{th_pa} for our model) to account for the two main time constants. The instantaneous power dissipated in the sensor Al strip (P_{diss}) is represented in Fig. 8 by a current generator, and the temperature is represented by node voltages (a battery sets the constant ambient temperature T_a).

From the thermal circuit described in Fig. 8, the static magnetoresistance temperature can be calculated as follows:

$$\begin{aligned} \Delta T &= R_{th_da} P_{diss} + T_a \\ P_{diss} &= I_{in}^2 R_{in} \\ R_{th_da} &= R_{th_dp} + R_{th_pa} \end{aligned} \quad (4)$$

4.2.1.1.4. Caracterización y modelado de los efectos térmicos dependientes del tiempo:

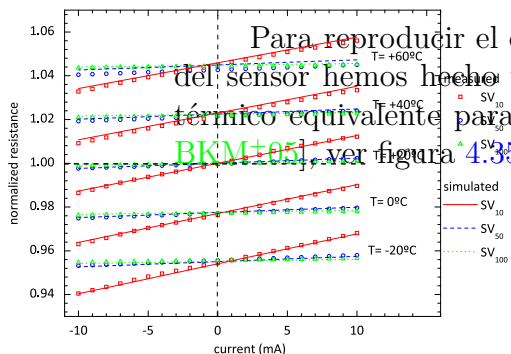


Fig. 6. Normalized magnetoresistance versus driven current for SV devices at different temperatures. Simulated data are shown in lines and experimental measurements in symbols.

Table 2
Extracted parameters for the static steady-state thermal model.

Device	$R_{RT,0}$ (Ω)	ΔR_T ($\Omega/^\circ\text{C}$)	MR_{RT} (Ω/A)	ΔMR_T ($\Omega/\text{A}/^\circ\text{C}$)
SV ₁₀	1105.13	1.268	1400	-2.68
SV ₅₀	1153.50	1.295	271.77	-0.148
SV ₁₀₀	1081.43	1.210	97.0	-0.195

Figura 4.35 – Parámetros térmicos extraídos en función de la temperatura [RRR⁺11e].

po de la forma $R_{th}C_{th}$. La transferencia estática de calor desde la área activa del sensor al ambiente exterior es subdividida en dos procesos serie: uno del dispositivo al encapsulado, modelado por la resistencia térmica R_{th_dp} , expresada en K/W, cuya contribución está justificada por las características tecnológicas del procedimiento de encapsulado del dispositivo; y del encapsulado al ambiente, modelado por la resistencia térmica R_{th_pa} [PM02]. La resistencia térmica del encapsulado–aire depende no solo del tamaño, la forma y la estructura del encapsulado sino también de la orientación, el flujo de aire cercanos al empaquetado. Cada una de las resistencias se ha unido a una capacidad térmica (C_{th_dp} y C_{th_pa}) para representar las constantes de tiempo más representativas. La potencia instantánea disipada en la pista de aluminio del sensor (P_{diss}) se representa en la figura 4.35 por un generador equivalente y la temperatura se muestra como la tensión en los diferentes nodos. La temperatura ambiente T_a se modela como un generador de tensión o batería.

La temperatura estática de la magnetorresistencia de la figura 4.35 puede ser calculada como:

$$\begin{aligned} \Delta T &= R_{th_da} P_{diss} + T_a \\ P_{diss} &= I_{SENSOR}^2 R_{in} \\ R_{da} &= R_{th_dp} + R_{th_pa} \end{aligned} \quad (4.2.7)$$



Hemos seguido un procedimiento similar al publicado por [BMBF00] para la extracción de los parámetros térmicos del modelo. Los resultados obtenidos se presentan en la tabla 4.4. El

Device	R_{th_dp} (K/W)	C_{th_dp} (J/K)	R_{th_pa} (K/W)	C_{th_pa} (J/W)	MR_{RT} (Ω/A)	R_{in} (Ω)
SV _{10μm}	56.0	2.6	31.5	0.03	1267.1	18.1
SV _{50μm}	20.6	3.7	14.2	0.15	265.2	8.04
SV _{100μm}	17.6	4.6	10.1	0.34	113.6	7.49

Tabla 4.4 – Parámetros del modelo térmico dependiente de la temperatura de los dispositivos magnetorresistivos SV. Para validar el modelo propuesto se han realizado medidas experimentales para caracterizar el comportamiento térmico dinámico consistente en la alimentación de pulsos de corriente de diferentes amplitudes a través de las pistas de corriente y la medida de la resistencia. Los resultados se muestran en las figuras 4.36 y 4.37.

modelo térmico completo ha sido implementado en Verilog-A. Se han realizado simulaciones transitorias para determinar el grado de ajuste con las medidas experimentales de los dispositivos magnetorresistivos SV. Para validar el modelo propuesto se han realizado medidas experimentales para caracterizar el comportamiento térmico dinámico consistente en la alimentación de pulsos de corriente de diferentes amplitudes a través de las pistas de corriente y la medida de la resistencia. Los resultados se muestran en las figuras 4.36 y 4.37.

Table 3

Complete parameters of the time-dependent thermal model.

Table 4

Thermal parameters comparison

Device R_{th_dp} (K/W) C_{th_dp} (J/K) R_{th_pa} (K/W) C_{th_pa} (J/K) MR_{RT} (Ω/A) R_{in} (Ω)

Device I (mA) $\Delta R_{max(B)}$ (Ω)

SV_{10 μ m} 56.0 2.6 31.5 0.03 1267.1 18.1

SV₁₀ 10 12.9

SV_{50 μ m} 20.6 3.7 14.2 0.15 265.2 8.04

SV₅₀ 20 25.8

SV_{100 μ m} 17.6 4.6 10.1 0.34 113.6 7.49

SV₁₀₀ 100 11.9

SV₅₀ 50 12.4

SV₁₀₀ 100 24.8

SV₁₀₀ 50 5.96

SV₁₀₀ 100 11.9

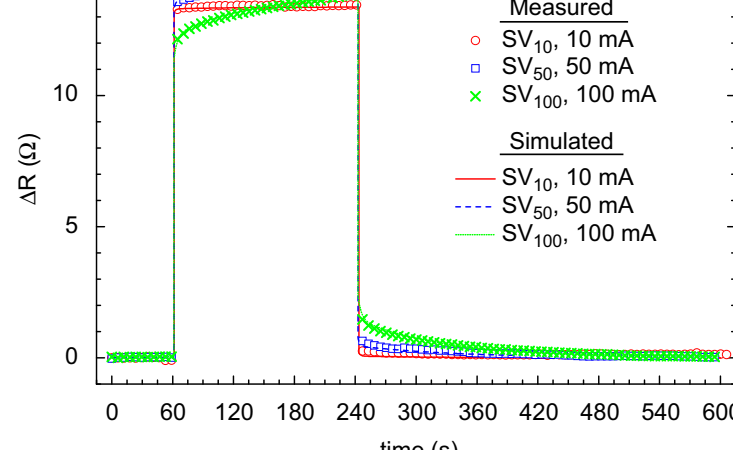


Figura 4.36 – Evolución temporal de la magnetorresistencia para pulsos de corriente de diferentes amplitudes, dando similares campos excitantes.

figuras se pueden observar dos tipos de respuestas. Una relativa al campo magnético (respuesta abrupta) y otra relativa al autocalentamiento o *self-heating* (incremento de la curva). En la figura 4.36 comparamos la respuesta de diferentes dispositivos a pulsos de corriente que crean campos magnéticos efectivos similares. Una pequeña desviación del valor de ΔR en la fase de enfriamiento es observada siendo generada por una pequeña histéresis producida por la gran corriente de excitación utilizada. La respuesta comparativa del dispositivo a diferentes pulsos de corriente se presenta en la figura 4.37.

De los resultados anteriores y de su análisis comparativo se pueden extraer algunas conclusiones cualitativas que de manera resumida se presentan en la tabla 4.5. $\Delta R_{max(B)}$ representa el incremento de la resistencia debido al efecto magnetorresistivo relacionado con el campo magnético efectivo aplicado que a su vez está asociado con la sensibilidad

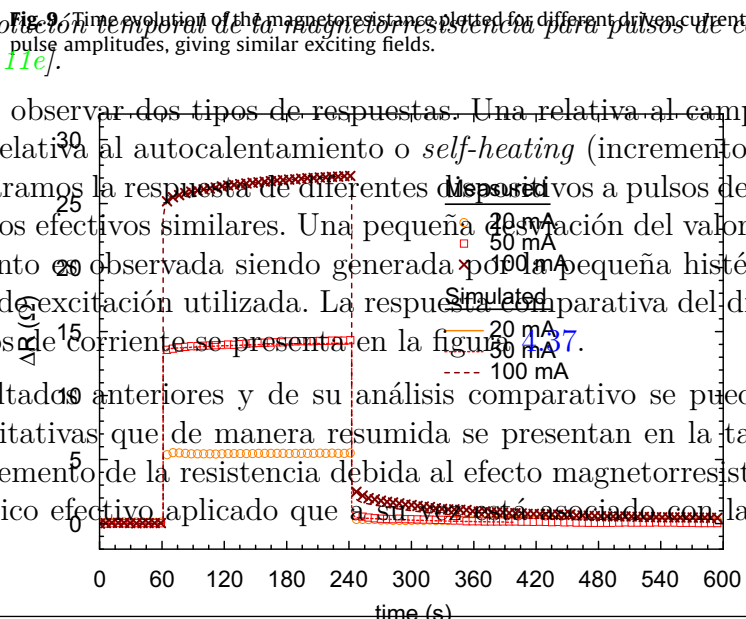


Fig. 10. Time evolution of SV₅₀ sensor magnetoresistance plotted for different driven current pulse amplitudes.

pulse amplitude, making use of the Table 3), and taking into account maximum temperature increasing data be obtained. As observed, higher voltages and narrow current strips, due to the heating ($\Delta R_{max}(T)$) can be obtained temperature coefficients maximum relative value is for a 100

4. Conclusions

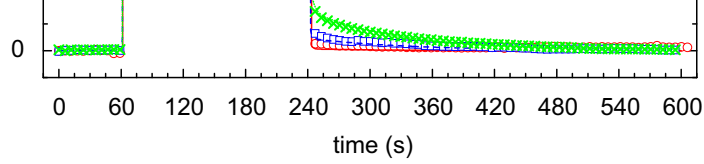
An advanced and complete model of magnetoresistance sensors is presented. Both thermal effects (including steady-state and transient regimes) are taken into account in a coherent modeling approach. The making use of experimental measurements of different devices following a physically based model has been implemented in Verilog-A. Applications based on simulation of the sensor data obtained in the lab have been achieved. The results of the model for different current pulse amplitudes are compared with the results obtained by the proposed model, showing that the inclusion of fabrication frequency effects or statistical dispersion in the model requires further studies.

Acknowledgments

This work was supported in part by the Spanish Government under Projects GV05/2008 and GV06/2008 and by the Spanish Ministry of Education under projects ENE2008-06588-C04-04 and ENE2008-06588-C04-04

References

[1] P.P. Freitas, R. Ferreira, S. Cardoso, F. Silva, J. Phys.: Condens. Matter 19 (2007) 165501
 [2] P.P. Freitas, F. Silva, N.J. Oliveira, L.V. Costa, J. Phys.: Condens. Matter 19 (2007) 165502



4.2. Desarrollo del modelo cuasi-estático

Fig. 9. Time evolution of the magnetoresistance plotted for different driven current pulse amplitudes, giving similar exciting fields.

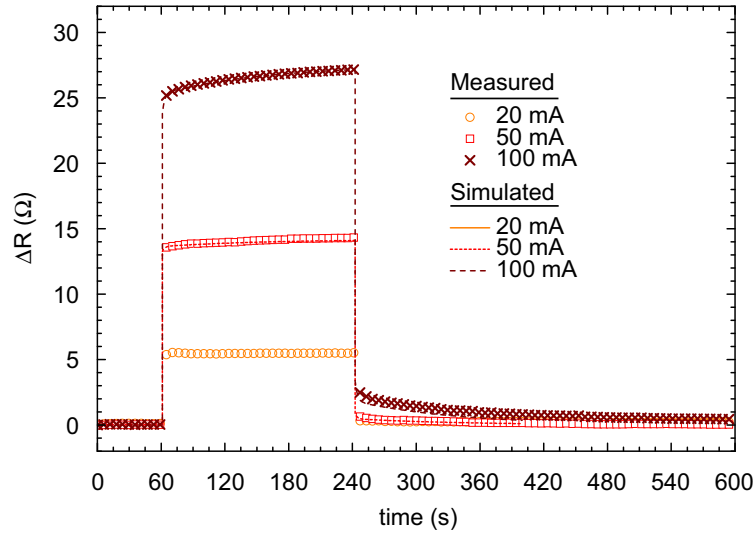


Figura 4.37 – Evolución temporal de la magnetorresistencia del sensor $SV_{50\mu m}$ para pulsos de diferentes amplitudes de corriente.

dispositivos mostrada en la tabla 4.2

Se obtienen anchura de pista pueden obtenerse máximo relativo dispositivo SV_{100} .

self-heating (increasing curve). In Fig. 9, we compare the response of different devices to current pulses giving similar effective magnetic fields. As observed, the effect of the electrical current on the heating is higher compared with the effect of the R_{in} . A small deviation from the R_0 value at the end of the cooling tie is observed, which is due to the small ΔR_{RRR} produced by the steady pulse exciting current. The comparative of the response of a particular device (SV_{50}) to different current pulses is shown in Fig. 10.

Device	Quantitative Conclusions (ΔR_{RRR})	ΔR_{RRR} (Ω)	ΔR_{RRR} (Ω)	ΔR_{RRR} (Ω)
$SV_{10\mu m}$	10	12.9	0.16	0.2
$SV_{10\mu m}$	20	25.8	0.32	0.4
$SV_{50\mu m}$	50	12.4	0.70	0.9
$SV_{50\mu m}$	100	24.8	2.8	3.6
$SV_{100\mu m}$	50	5.96	0.52	0.6
$SV_{100\mu m}$	100	11.9	2.1	2.5

Tabla 4.5 – Comparativa del comportamiento térmico de los dispositivos magnetorresistivos SV [RRR+11e].

En el listado 4.6 se muestra el código introducido en el modelo compacto de Verilog-A para incluir los efectos térmicos en régimen dinámico. Ha sido necesario incluir los nuevos parámetros del modelado térmico de la ecuación 4.2.7 y el código necesario para elevar la temperatura interna del dispositivo según la potencia disipada en la resistencia de entrada R_{in} .

```

1 // Parameter for self-heating
2 parameter integer Selfheating = 0 from [0:1];
3 parameter real Rth_in = 40.0 from [0: inf]; // Thermal resistance [Ohm]
4 parameter real Rth_cp = 16.6 from [0: inf]; // Thermal resistance [Ohm]
5 parameter real Cth_cp = 5.8 from [0: inf]; // Thermal capacitance [F]
6 parameter real Rth_pa = 4.16 from [0: inf]; // Thermal resistance [Ohm]
7 parameter real Cth_pa = 1.4 from [0: inf]; // Thermal capacitance [F]
8 parameter real Tnominal = 20 from (-P_CELSIUS0: inf); // measeasuring T [C]
9

```

4. Conclusions

An advanced and complete distance sensors is presented. thermal effects (including st regimes) are taken into account coherent modeling approach making use of experimental devices following a physically has been implemented in Ve electronics applications based data obtained in the lab have b of the model for different cu results obtained by the propos improving it by including f frequency effects or statistical further studies.

Acknowledgments

This work was supported Government under Projects O the Andalusian Regional Gove and by the Spanish Ministry projects ENE2008-06588-C04-

References

- [1] P.P. Freitas, R. Ferreira, S. Carro, J. Phys.: Condens. Matter 19 (2007) 157.
- [2] P.P. Freitas, F. Silva, N. Schöpfer, Sensors and Actuators A: Phys. 157 (2009) 157.
- [3] Proceedings of the IEEE (2006) 23–28.
- [4] M. Schröter, S. Lehmann, Phys. Rev. Lett. 97 (2006) 279–286.
- [5] A. Balijepalli, J. Ervin, W. Lepkowski, a PD SOI MESFET for wide temperature range, IEEE Trans. Electron Devices 50 (2003) 1264–1273.
- [6] F. Parrain, S. Megherbi, G. Raynaud, G. Schöpfer, N. Faure, P. Cusin, modeling of an electrostatically actuated microvalve, IEEE Trans. Electron Devices 53 (2006) 23–28.
- [7] B. Das, W.C. Black, A.V. Pohm, magnetoresistance memory bits, IEEE Trans. Electron Devices 53 (2006) 23–28.
- [8] C. Reig, D. Ramírez, H.H. Li, P.P. Freitas, valve structures, IEE Proc.—Circuits, Devices, Syst. 154 (2007) 157.


```

10
11   if (Selfheating)           //En caso de activar la opcion de Self-Heating
12       T = V(t_Chip);        //aumentamos la tempratura del dispositivo con la temperatura
13                               //del nodo interno de tensión del acoplamiento térmico
14   delta_T = T - T_nom;
15
16 //Efectos del autocalentamiento
17 V(t_Ambient) <+ $temperature;
18   if (Rth_pa == 0) begin
19       V(temp_PackageAmbient) <+ 0.0;
20   end
21   else begin
22       I(temp_PackageAmbient) <+ V(temp_PackageAmbient)/Rth_pa;
23       I(temp_PackageAmbient) <+ ddt(Cth_pa*V(temp_PackageAmbient));
24       Potencia_Joule = pow(I(input_resistence), 2.0)*Rth_in; //+V(np, nn)*V(np, nn)/
25                               Magnetorresistencia;
26       I(temp_PackageAmbient) <+ -1.0*Potencia_Joule;
27       if (DEBUG)
28           $strobe("Pot_Joule=%g  V(t)=%g", Potencia_Joule, V(temp_PackageAmbient));
29   end
30   if (Rth_cp == 0) begin
31       V(temp_ChipPackage) <+ 0.0;
32   end
33   else begin
34       I(temp_ChipPackage) <+ V(temp_ChipPackage)/Rth_cp;
35       I(temp_ChipPackage) <+ ddt(Cth_cp*V(temp_ChipPackage));
36       Potencia_Joule = pow(I(input_resistence), 2.0)*Rth_in; //+V(np, nn)*V(np, nn)/
37                               Magnetorresistencia;
38       I(temp_ChipPackage) <+ -1.0*Potencia_Joule;
39       if (DEBUG)
40           $strobe("Pot_Joule=%g  V(t)=%g", Potencia_Joule, V(temp_ChipPackage));
41   end

```

Listado 4.6 – Código necesario para incorporación del modelado térmico dinámico.

4

4.2.1.2 Dispositivos MTJ.

Las estructuras de efecto túnel magnetorresistivo **MTJ** son actualmente los dispositivos magnetorresistivos más avanzados en cuanto a su potencial en aplicaciones de medida de campos magnéticos. Sus ventajas en relación a sensibilidad (**TMR**) los han hecho desmarcarse del resto de tecnologías. En la figura 4.38 se representa la evolución histórica de la sensibilidad de estos dispositivos a lo largo del tiempo. La capa aislante de MgO destaca por su capacidad de producir coeficientes **TMR** cercanos a 600% a temperatura ambiente y 1144% a 5 K [IHA+08].

El uso de estos sensores se encuadra en dos tipos de aplicaciones diferenciadas según su frecuencia de trabajo: alta frecuencia, en unidades de disco duro (HDD) donde los datos son leídos a frecuencias entre 50 MHz–1 GHz y baja frecuencia (< 100 KHz) en sensores magnéticos y de corriente. Ejemplos como sensores magnéticos a baja frecuencia son las aplicaciones de magnetocardiografía (**MCG**) y magnetoencefalografía (**MEG**). Los requerimientos en cuanto a nivel de señal disponible para estas aplicaciones se dan en la tabla 4.6 [SB06].

	Método Magnético	Nivel de Señal (pT)	Rango de Frecuencias (Hz)	Método Eléctrico
Cerebro	Magnetoencefalografía (MEG)	$10^{-2} - 1$	0.1-100	Electroencefalografía (EEG)
Corazón	Magnetocardiografía (MCG)	1-100	1.01-100	Electrocardiografía (ECG)

Tabla 4.6 – Métodos magnéticos no invasivos de monitorización corporal [SB06].

La estructura de los dispositivos de efecto túnel magnetorresistivos está basada en una

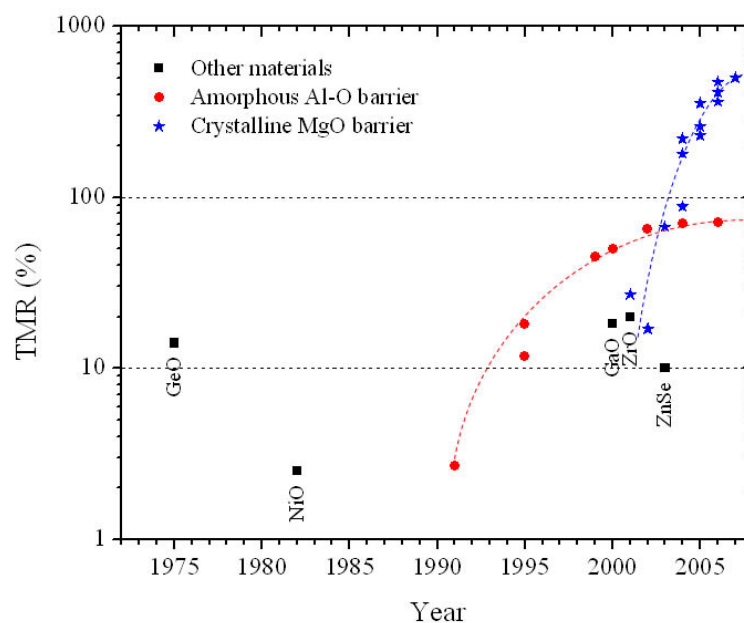


Figura 4.38 – Evolución histórica del coeficiente TMR desde el descubrimiento de las MTJ en 1975. Todos los puntos están medidos a temperatura ambiente excepto los dos primeros a 4.2 K [GdA09]

región activa GMR que se introdujo en el apartado 4.1.1.3.2 y está constituida por dos capas ferromagnéticas (FM1) y (FM2) separadas por una capa aislante. El cambio de resistencia en un dispositivo MTJ se produce cuando el ángulo (θ) entre los vectores de magnetización de las capas ferromagnéticas se modifica por la presencia de un campo magnético externo. En estos dispositivos se verifica que el número de electrones es directamente proporcional al área del elemento y por lo tanto la resistencia del elemento es inversamente proporcional al área. El producto resistencia-área (RA), expresado usualmente en ($\Omega\mu\text{m}^2$), se utiliza para comparar los elementos con diferentes áreas. La resistencia total (R) para un campo magnético aplicado varía entre un valor mínimo (R_P) para $\theta = 0^\circ$ y un máximo ($R_{AP} = R_P + \Delta R$). Así R puede expresarse como (4.2.8).

$$R = R_P + \frac{\Delta R}{2} (1 - \cos \theta) \quad (4.2.8)$$

Cuando se polariza el dispositivo MTJ mediante una tensión externa (V_{BIAS}) se establece una diferencia de potencial entre las dos capas ferromagnéticas separadas por el aislante. Éste forma una barrera para el flujo de la corriente. Existen diferentes modelos en la bibliografía que establecen la corriente que atraviesa la barrera: a) modelo de Simmons para láminas (FM1) y (FM2) iguales [Sim63] y b) Brinkman para láminas no iguales [BDR70]

Los dispositivos MTJ presentan variaciones de la resistencia túnel (TMR) con la tensión de polarización (V_{BIAS}). Normalmente con el incremento de la tensión de polarización se produce una disminución del TMR, ver figura 7.1. La tensión de polarización a la cual se obtiene la mitad de TMR se denomina $V_{1/2}$.

Si la tensión aplicada entre el aislante supera cierto valor, el elemento (unión) se destruye. La tensión a la que se produce la ruptura se denomina tensión de ruptura (V_{break}). En

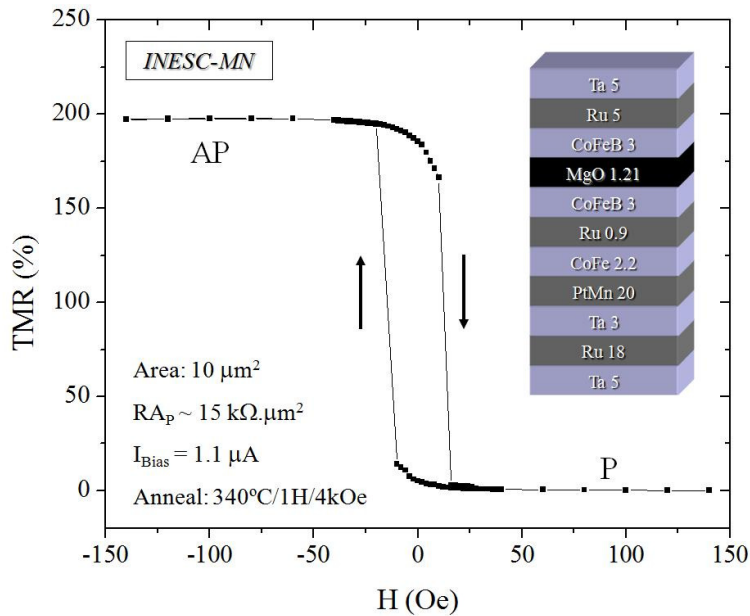


Figura 4.39 – Magnetorresistencia de efecto túnel (TMR) versus el campo magnético aplicado (H) para una *MTJ* con aislante *MgO*. *P* y *AP* representan los estados paralelo y antiparalelo respectivamente. Las flechas indican la dirección de la medida. Los grosores de las diferentes capas de material se expresan en nm [GdA09].

una aproximación se puede relacionar (V_{break}) con la rigidez dieléctrica (E) de la barrera y el grosor (t) del aislante mediante (4.2.9)

$$V_{\text{break}} \propto E \cdot t \quad (4.2.9)$$

4.2.1.2.1 Fabricación de sensores de corriente MTJ.

La microfabricación de los dispositivos magnetorresistivos *MTJ*, estudiados y caracterizados en esta tesis, ha sido realizada en una colaboración del Dpto. de Ingeniería Electrónica de la Universidad de Valencia e INESC-MN, en las instalaciones de éste último. La estructura básica de las *MTJ* fue presentada en el apartado 4.1.1.3.2. Las diferentes fases de fabricación se detallan en el apéndice 7.1 por el alto nivel de detalle del procedimiento.

A lo largo del proceso de fabricación se ha referenciado al documento denominado *Run Sheet*¹ que está disponible también en el apéndice.

4.2.2 Puentes magnetorresistivos.

En las aplicaciones de medida de corriente es deseable una relación lineal entre la corriente de entrada y la magnitud de salida del sensor. El puente de Wheatstone formado

¹Es un documento en el que se detalla todo lo relativo a las diferentes fases de la fabricación del dispositivo en dentro de la sala blanca. La configuración de las diferentes máquinas, los grosores de las capas depositadas, los tiempos de revelado, atacado y *lift-off*

por cuatro elementos magnetorresistivos simples ofrece buena linealidad y genera una tensión de salida nula en caso de no existir campo magnético externo aplicado. Actualmente existen gran cantidad de sensores magnetorresistivos tanto **GMR** como **MTJ** integrados en puente de Wheatstone para aplicaciones en campos muy diversos como son los de sensores de control de posición, geomagnetismo, biosensores, corriente y medidores de energía [FFCC07]. El

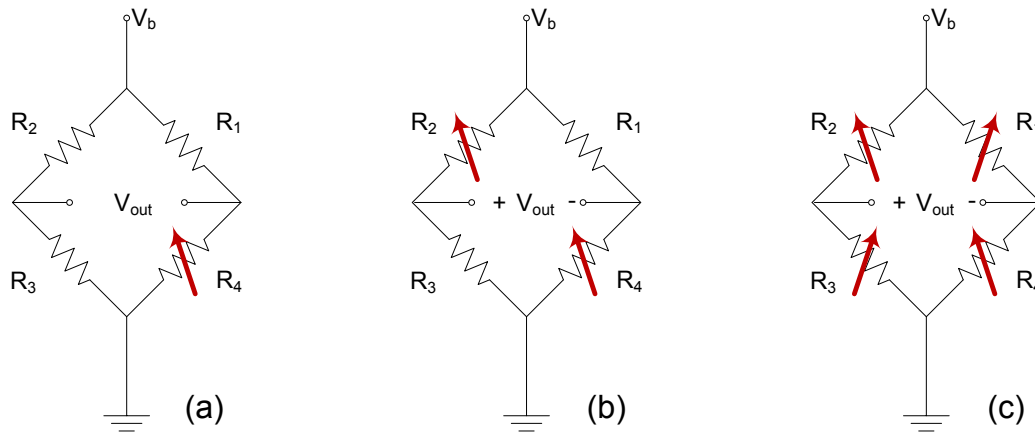


Figura 4.40 – Tipos de puentes de Wheatstone [RCBR09].

puente de Wheatstone es un circuito formado por dos ramas paralelas de dos elementos en serie en cada una de ellas. Fue descrito inicialmente por Samuel Hunter Christie in 1833, aunque su uso fue difundido por Sir Charles Wheatstone, que descubrió gran cantidad de aplicaciones para esa topología en 1843. Se usa con frecuencia en la medida de la resistencia de un componente desconocido. Está formado por dos divisores resistivos. La tensión de salida diferencial del puente, figura 4.40, tiene la expresión siguiente:

$$V_{\text{out}} = \left(\frac{R_3}{R_2 + R_3} - \frac{R_4}{R_1 + R_4} \right) V_b \quad (4.2.10)$$

Tradicionalmente la incorporación de un galvanómetro o amperímetro de alta sensibilidad entre los terminales de salida del puente ha permitido la monitorización de la corriente que fluye de un divisor a otro, I_g . Si los divisores de tensión tienen la misma relación $R_3/R_2 = R_4/R_1$, se dice que el puente está balanceado generando una tensión nula a la salida y una corriente $I_g = 0$. Si una de las resistencias del puente cambia su valor comienza a circular corriente $I_g \neq 0$. La tensión de salida del puente es diferencial, en los diseños de sensores de corriente magnetorresistivos se utilizan puentes de Wheatstone para obtener una tensión de salida linealmente proporcional a la corriente I_{SENSOR} , y con tensión de salida nula en ausencia de corriente. Dependiendo de los requerimientos industriales existen diferentes configuraciones de puentes [RRR+10, RCBR09, RCBR08].

4.2.2.1 Puentes de un elemento variable.

La configuración en puente de Wheatstone basada en único elemento variable está formada por cuatro resistencias del mismo valor nominal (R), con una de ellas variable con

un incremento (ΔR) proporcional a la corriente I_{SENSOR} . Su estructura viene dada por la figura 4.40(a) donde $R_1 = R_2 = R_3 = R$ y $R_4 = R + \Delta R(I_{\text{SENSOR}})$.

$$V_{\text{out}}^{\text{Single-R}} = \frac{\frac{\Delta R}{R}}{2(2 + \frac{\Delta R}{R})} V_b \quad (4.2.11)$$

La relación entre la tensión de salida $V_{\text{out}}^{\text{Single-R}}$ y la variación de resistencia ΔR no es lineal. La utilidad de esta configuración está limitada al caso en el que $\frac{\Delta R}{R} \ll 1$.

4.2.2.2 Puentes de dos elementos variables.

La configuración en puente de Wheatstone basada en dos elementos variables está formada por cuatro resistencias del mismo valor nominal (R) con dos de ellas variables con un incremento (ΔR) proporcional a la corriente I_{SENSOR} . Su estructura viene dada por la figura 4.40(b) donde $R_1 = R_3 = R$ y $R_2 = R_4 = R + \Delta R(I_{\text{SENSOR}})$.

$$V_{\text{out}}^{\text{Dual-R}} = \left(\frac{\frac{\Delta R}{R}}{2 + \frac{\Delta R}{R}} \right) V_b \quad (4.2.12)$$

La relación entre la tensión de salida $V_{\text{out}}^{\text{Dual-R}}$ y la variación de resistencia ΔR no es lineal aunque tiene una magnitud del doble de la anterior. La expresión 4.2.12 puede aproximarse para el caso de pequeñas variaciones de resistencia, $\frac{\Delta R}{R} \ll 1$, por esta otra:

$$V_{\text{out}}^{\text{Dual-R}} \approx \frac{\Delta R}{R} \frac{V_b}{2} \quad (4.2.13)$$

4.2.2.3 Puentes de cuatro elementos variables.

La configuración en puente de Wheatstone basada en cuatro elementos variables está formada por cuatro resistencias variables con un incremento (ΔR) proporcional a la corriente I_{SENSOR} . Su estructura viene dada por la figura 4.40(c) donde $R_2 = R_4 = R + \Delta R(I_{\text{SENSOR}})$ y $R_1 = R_3 = R - \Delta R(I_{\text{SENSOR}})$ ².

$$V_{\text{out}}^{\text{Quad-R}} = \frac{\Delta R}{R} V_b \quad (4.2.14)$$

La relación entre la tensión de salida $V_{\text{out}}^{\text{Quad-R}}$ y la variación de resistencia ΔR es lineal, y la amplitud es la mayor de los casos anteriores; estas dos razones motivan que sea la configuración elegida para implementar los puentes magnetorresistivos que serán usados como base de los sensores de corriente estudiados en este trabajo.

²La diferente dirección de circulación de la corriente a medir por la pista respecto a las diferentes resistencias, hace que el campo magnético asociado cambie de orientación respecto al *easy axis* lo que hace variar su valor con signo contrario

4.2.3 Diseño de puentes magnetorresistivos.

Los diferentes tipos de diseños realizados [CBRR⁺09] se presentan en la figura 4.41. Están formados por cuatro elementos magnetorresistivos variables $R_1 - R_4$ que aumentan o disminuyen su valor dependiendo del sentido en el que la corriente I_{SENSOR} atraviesa el dispositivo. El campo magnético que ésta genera es paralelo o antiparalelo a la capa inmobilizada (*pinned*) mientras que la capa libre (*free*) (inicialmente perpendicular al campo magnético) rotará dependiendo de la orientación del campo magnético externo. Las resistencias R_1 y R_3

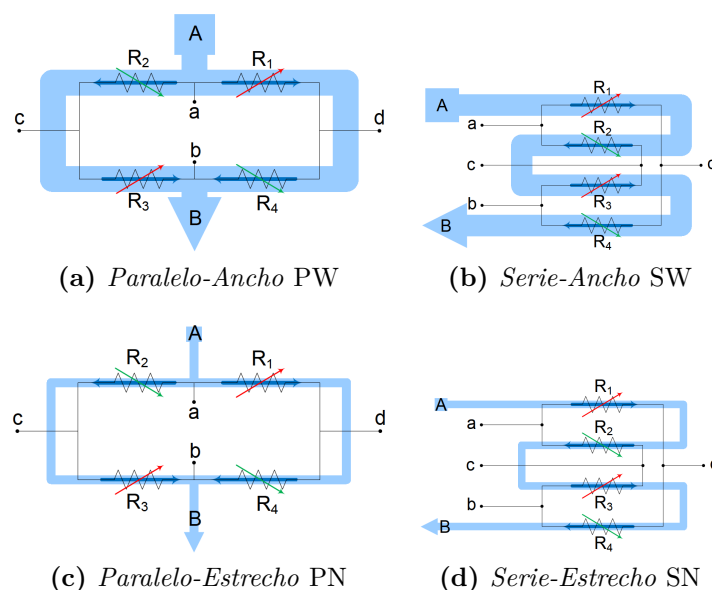


Figura 4.41 – Tipos de sensores de corriente magnetorresistivos basados en el puente de Wheatstone [CBRR⁺09].

disminuyen cuando la corriente I_{SENSOR} circula por la pista que une los terminales (A) y (B). Por el contrario, R_2 y R_4 aumentan con la corriente I_{SENSOR} . La polarización del puente se realiza a través de los terminales (a) y (b) obteniéndose la salida de tensión diferencial en los terminales (c) y (d). Dependiendo de la sensibilidad que se persigue se elige la configuración: serie para alta sensibilidad y paralelo para el caso de baja sensibilidad.

Para conseguir minimizar las diferencias físicas y eléctricas de las cuatro resistencias, éstas se fabrican a la vez, lo que mejora las características del puente como la deriva térmica y la inmunidad a campos magnéticos externos. Hemos fabricado sensores con pistas de corriente de 10 (*narrow-N*) y 100 (*wide-W*) μm para estudiar su comportamiento tanto en serie (SN y SW) como en paralelo (PN y PW).

Es muy importante respetar una regla de diseño en los puentes de Wheatstone consistente en asegurar que la impedancia que carga a cada una de las dos ramas en los terminales de salida (c) y (d) igual, por lo que se evitará una asimetría en la corriente de polarización I_{FEED} que circula por cada sensor.



4.2.3.1 Puentes AMR.

En este apartado se han caracterizado y modelado diferentes circuitos comerciales basados en dispositivos AMR.

Un resumen de las características más importantes de estos sensores magnéticos comerciales se presenta en la tabla 4.7 [RCBR09]. Se añade también para contrastar las prestaciones un circuito basado en GMR. El dispositivo AA002-02 es un sensor AMR y el AA004-02 de

Fabricante	Magnetorresistencia	Encapsulado	Resistencia Nom. (Ω)	Rango Temperatura (°C)	Rango de Trabajo ⁶	Sensibilidad	Tensión Offset (mV/V)	Coef. Temperatura Resistivo (%/K)
NVE	ZMC20	DIL-14 (modif.)	1200 - 2200	-40 a +125	-40 a +125	0.25 mV/V/A	±2	-0.3
NVE	HMC1015	SOIC	800 - 1300	-40 a +85	-40 a +85	1.0 mV/V/G	±2	0.025
NVE	AA004-02 ⁵	SOIC	5000	-40 a +85	-40 a +85	0.9-1.3 mV/V/Oe	±4	0.14

Features:

- Wheatstone Bridge Analog Output
- Operating Temperature to 125°C Continuous
- Wide Linear Range of Operation
- Near Zero Voltage Operation
- DC to >1MHz Frequency Response
- Small, Low-Profile Surface Mount Packages

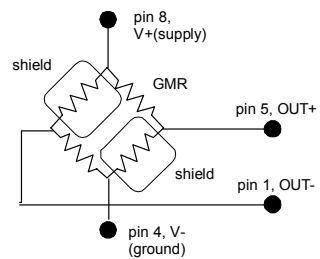
Applications:

- General Motion, Speed and Position Sensing
- Low Power, Low Voltage Applications
- Low Field Sensing for Magnetic Media Detection
- Current Sensing

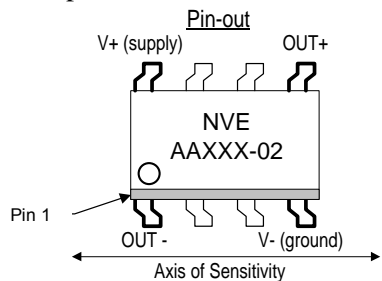
Description:

The AA-Series GMR sensors are general-purpose magnetometers for applications. They exhibit excellent linearity, a large output signal with applied magnetic field, and a purely ratiometric output.

Functional Block Diagram



(a) Esquema del puente de medida.



(b) Pin-out

Figura 4.42 – Sensor de corriente magnético de fabricante NVE™[NVE03].

Part Number	Saturation Field (Oe)	Linear Range (Oe)	Sensitivity (mV/V-Oe)	Resistance (Ohms)	Package
AA002-02	15	1.5 - 10.5	3.0 - 4.2	5K ±20%	SOIC8
AA003-02	20	2.0 - 14	2 - 3.2	5K ±20%	SOIC8
AA004-00	50	5 - 35	0.9 - 1.3	5K ±20%	MSOP8
AA004-02	50	5 - 35	0.9 - 1.3	5K ±20%	SOIC8
AA005-02	100	10 - 70	0.45 - 0.65	5K ±20%	SOIC8
AA006-00	50	5 - 35	0.9 - 1.3	30K ±20%	MSOP8
AA006-02	50	5 - 35	0.9 - 1.3	30K ±20%	SOIC8

⁶ Unidades: 1 Gauss (G) = 1 Oersted (en aire), 1G=79.58 A/m, 1G=10⁻⁴ Tesla

⁷ 0.4 Gauss (Campo magnético terrestre = 0.5 Gauss)

⁸ Detección del signo del campo magnético.

Los sensores HMC1021 y AA004-2 poseen encapsulado SOIC, figura 4.42(b), y deben ser soldados sobre una pista de corriente en una placa de circuito impreso (*PCB*) para funcionar como sensores de corriente. El sensor ZMC20, sin embargo, dispone de un encapsulado *DIL-14* modificado que integra la pista de corriente dentro del dispositivo. Este último ha sido caracterizado con detalle en las secciones siguientes.

Al igual que en el apartado 4.2.1.1.2, se utilizó un conjunto de equipos de instrumentación conectados al PC para la adquisición de los datos. Para la obtención del modelo se caracterizaron los valores de las resistencias independientes ($R_1 - R_4$) medidas en función de la corriente de excitación del sensor (I_{SENSOR}). Con objeto de extraer los valores exactos, se midió la resistencia del puente entre nodos consecutivos. La resistencia R_n es obtenida a partir de consideraciones serie/paralelo. Los resultados se muestran en la figura 4.43. Del

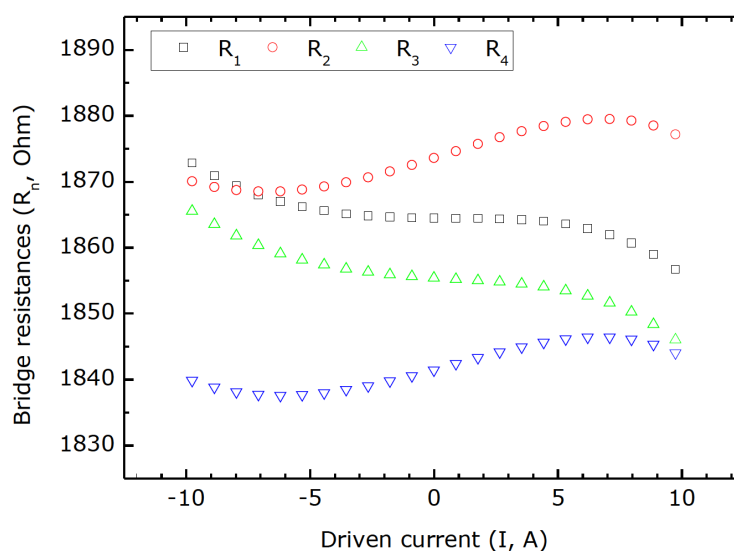


Figura 4.43 – Resistencias del puente Wheatstone en función de la corriente excitación (I_{SENSOR}) del sensor (ZMC20).

análisis de los datos se puede extraer el comportamiento casi lineal de las resistencias para bajas corrientes. Sin embargo para corrientes mayores a 1 A, se observa comportamiento cúbico [RRR+10]. El modelo de la resistencia se puede plantear de esta forma:

$$R = R_0 + MR_1 \cdot I_{\text{SENSOR}} + MR_2 \cdot I_{\text{SENSOR}}^2 + MR_3 \cdot I_{\text{SENSOR}}^3 \quad (4.2.15)$$

donde I_{SENSOR} es la corriente a determinar en el sensor, R_0 es el valor de una resistencia del puente con corriente nula, y MR_i los coeficientes polinómicos. El valor de estos coeficientes ha sido obtenido con métodos de aproximación numérica y se presentan en la tabla 4.8. Como se puede apreciar en la tabla 4.8, aunque los valores $R_1 - R_4$ son parecidos, no son exactamente iguales. Concretamente, MR_1 es mayor para R_1 y R_3 que para R_2 y R_4 . Esto se debe a la configuración del puente de Wheatstone de 2 elementos sensibles, figura 4.40(b). Se realizaron las simulaciones del puente en *Advanced Design System* con el modelo im-

	$R_0(\Omega)$	$MR_1(\Omega/A)$	$MR_2(\Omega/A^2)$	$MR_3(\Omega/A^3)$
R_1	1870.41	1.06	0.0055	-0.0088
R_2	1879.55	0.1666	0.0031	0.0084
R_3	1861.31	1.24	0.0065	-0.0088
R_4	1847.32	0.0051	0.00719	0.0083

Tabla 4.8 – Parámetros del modelo de magnetorresistencia para el puente ZMC20

plementado en Verilog-A, figura 4.44. Cada resistencia es considerada como un cuadripolo resistivo dependiente de la corriente y es modelada individualmente. La conexión de éstas se ha realizado para asegurar que la corriente (I_{SENSOR}) produce las variaciones necesarias en las resistencias sensibles, ver apartado 4.2.2.2. Haciendo uso de los datos descritos en la

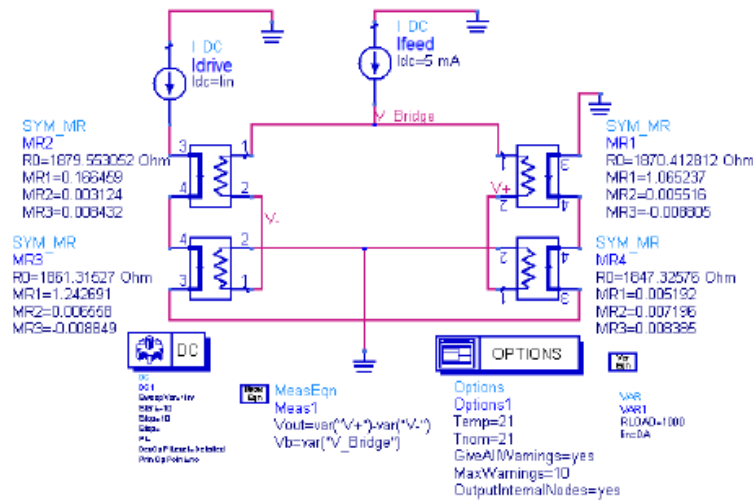


Figura 4.44 – Hoja de simulación de ADS para el puente magnetorresistivo AMR (ZMC20) de Zetex.

tabla anterior [RRR+10] hemos obtenido la respuesta en tensión en los terminales de salida una vez que el sensor ha sido polarizado con una corriente DC (I_{FEED} en la figura 4.44). Los resultados experimentales y los simulados se presentan en la figura 4.45 en función de la corriente del sensor (I_{drive} , figura 4.44). Las simulaciones reproducen las medidas con exactitud. Es importante destacar las propiedades de linealización que ofrece el puente, a pesar de la no linealidad de los elementos que lo componen. El modelo reproduce también las características de *offset* de tensión de salida.

	SV07-SN		SV07-PW	
	$R_0(\Omega)$	$MR_1(\Omega/A)$	$R_0(\Omega)$	$MR_1(\Omega/A)$
R_1	1131.5	1441.1	1118.6	122.6
R_2	1096.0	1384.1	1153.7	118.1
R_3	1062.5	1342.4	1154.2	124.9
R_4	1128.0	1410.1	1130.8	124.7

Tabla 4.9 – Parámetros del modelo de las magnetoresistencias del puente ZMC20.

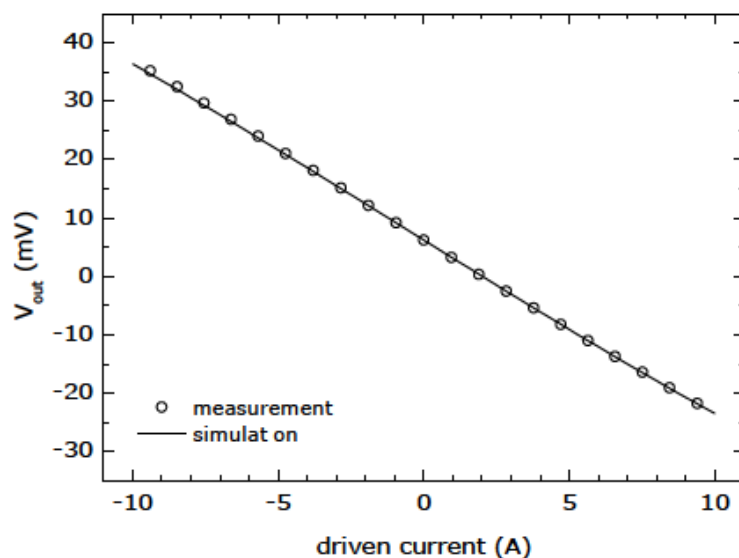


Figura 4.45 – Tensión de salida del puente magnetorresistivo AMR (ZMC20) de Zetex.

4.2.3.2 Puentes GMR.

4.2.4 Modelado de ruido.

Desde un punto de vista de práctico, la caracterización del ruido de un dispositivo es esencial para su utilización en aplicaciones circuitales. Para asegurar que la relación señal-ruido (SNR) es aceptable en aplicaciones de sensores de corriente, se necesita caracterizar el ruido. Se han realizado estudios en estos dispositivos electrónicos que relacionan los parámetros físicos de éstos, la contribución global del ruido, la calidad y fiabilidad de los dispositivos [Van94].

Aunque el factor TMR de las MTJ es significativamente más importante que su equivalente MR en las SV, el ruido intrínseco de los sensores MTJ es también mayor. Las MTJ únicamente podrán presentar mejores prestaciones que las SV si su relación señal-ruido (SNR) mejora.

En los dispositivos GMR, y MTJ, en particular, existen diferentes mecanismos responsables del ruido. En la figura 4.46 se muestran las curvas típicas de ruido obtenidas cuando circula una corriente a través de la barrera MTJ causando fluctuaciones en la tensión con una densidad espectral de ruido representada por (S_V) [KvPX03]. Las diferentes aportaciones del ruido se pueden clasificar en:

- Ruido independiente de la frecuencia o blanco (WNB), cuya densidad espectral de ruido es constante con la frecuencia, domina a altas frecuencias, y está compuesta por ruido térmico y *shot*.
- Ruido dependiente de la frecuencia, con una densidad espectral de ruido inversamente proporcional a alguna potencia de la frecuencia y domina a bajas frecuencias con un valor superior al ruido blanco.

In a magnetic tunnelling system, there are many mechanisms that can cause noise.

Figure 2-7 represents the typical noise curves that will be presented in this work, obtained

when a current flows through a MTJ causing voltage fluctuations with a power spectral

density represented by S_V [25].

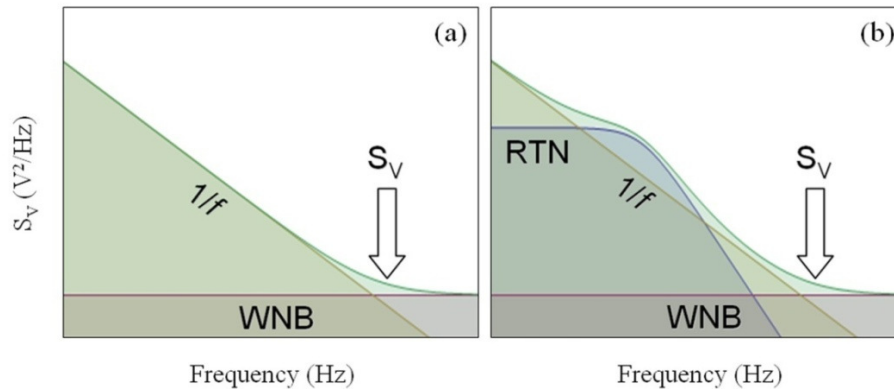


Figura 4.46 – Espectro de ruido teórico para dispositivo MTJ [KvPX03] sin (a) y con (b) ruido aleatorio telegráfico (RTN). El nivel de ruido blanco se representa con (WNB). S_V denota la densidad espectral de ruido y se representa en ejes logarítmicos tanto ordenada como abscisa.

4.2.4.1 Ruido independiente de la frecuencia.

Existen tres efectos que producen ruido independiente de la frecuencia en dispositivos magnetorresistivos. El ruido térmico, el ruido *shot* y el ruido termomagnético. El ruido termomagnético presenta un pico de resonancia en el intervalo de 2-10 GHz, como las aplicaciones en sensores de corriente a esta frecuencia quedan fuera del ámbito este trabajo, no lo consideraremos.

4.2.4.1.1 Ruido térmico - Johnson.

El ruido térmico, también llamado *Johnson* o *Nyquist*, asociado al comportamiento aleatorio del movimiento térmico de los electrones es descrito por la ecuación 4.2.16.

$$S_V^{\text{Thermal}}\left(\frac{V^2}{\text{Hz}}\right) = 4k_B T R \quad (4.2.16)$$

donde k_B es la constante de Boltzmann, T representa la temperatura y R la resistencia del dispositivo. Está presente tanto en los dispositivos MTJ como en el resto de las estructuras.

4.2.4.1.2 Ruido *shot*.

El ruido *shot* que aparece en las discontinuidades en los medios de conducción (p.e. presencia de barreras de potencial) como consecuencia de la naturaleza discreta de la carga eléctrica queda descrita por la ecuación 4.2.17.

$$S_V^{\text{Shot}}\left(\frac{V^2}{\text{Hz}}\right) = 2e I_{BIAS} R^2 \quad (4.2.17)$$

donde I_{BIAS} es la corriente media que fluye por el dispositivo, T representa la temperatura y R la resistencia del dispositivo. La barrera túnel del dispositivo MTJ es un ejemplo de discontinuidad en el medio de conducción; esto motiva que el ruido *shot* esté presente en las MTJ y no otras estructuras como las GMR o AMR que están fabricadas con capas metálicas

continuas. Normalmente presenta un valor menor que el ruido térmico.

4.2.4.2 Ruido dependiente de la frecuencia.

El límite de detección de campo magnético para los sensores magnetorresistivos queda establecido por el nivel de ruido blanco total. Sin embargo existen otros mecanismos que pueden limitar la capacidad de detección dependiendo de la frecuencia de trabajo. Para aplicaciones de baja frecuencia el mecanismo de ruido más importante es el ruido eléctrico $1/f$ que está presente en todas las medidas con polarización no nula del dispositivo magnetorresistivo, por lo que se recomienda trabajar a frecuencias superiores al pliegue⁹ $1/f$ para disminuir el ruido.

En el caso de los dispositivos **MTJ** aparece un ruido magnético de la forma $1/f$ de varios órdenes de magnitud superior al ruido eléctrico $1/f$ en la región lineal del sensor, siendo la aportación más importante en las aplicaciones de estos sensores [XLN+06].

4.2.4.2.1 Ruido aleatorio telegráfico - RTN.

El ruido aleatorio telegráfico (**RTN**) es la tercera fuente de ruido para los dispositivos **MTJ** que ha de ser tenido en cuenta en el caso de las aplicaciones de baja frecuencia, figura 4.46(b). Se debe a cambios abruptos de resistencia provocados por variaciones en las estructuras ferromagnéticas por activación térmica de dominios magnéticos inestables en la capa libre (*free*), desplazamientos de los átomos de oxígeno en la barrera, etc. [XLN+06]. La densidad espectral de ruido asociada posee una expresión típica Lorentziana 4.2.18:

$$S_V^{\text{RTN}}\left(\frac{V^2}{\text{Hz}}\right) = \frac{S_0}{1 + \left(\frac{f}{f_0}\right)^2} \quad (4.2.18)$$

donde S_0 es la densidad de ruido a 0 Hz y f_0 es la frecuencia de corte.

4.2.4.2.2 Ruido eléctrico $1/f$ *Rosa - Flicker*

Casi todos los dispositivos electrónicos por los que fluye una corriente presentan fluctuaciones en tensión con una densidad espectral de ruido inversamente proporcional a la frecuencia, figura 4.46. A pesar de que existen diferentes modelos para describir los efectos de este ruido $1/f$, es todavía el ruido menos entendido. El modelo que usaremos será el modelo paramétrico propuesto por Hooge que presenta la siguiente forma: [HH69, PJA99]

$$S_V^{\text{Flicker}}\left(\frac{V^2}{\text{Hz}}\right) = \alpha_H \frac{V^2}{N_c f^{AF}} \quad (4.2.19)$$

donde α_H es el parámetro *Hooge*, N_c es el número de portadores de carga en la muestra, f es la frecuencia y AF es el coeficiente de dependencia de la frecuencia.

⁹Frecuencia a la que el ruido $1/f$ alcanza el valor del **WNB**. Ver flecha vertical en figura 4.46.

Para los dispositivos magnetorresistivos **MTJ**, N_c está relacionado con el número de trampas para las cargas que es proporcional al área de la barrera. La ecuación 4.2.19 se formula como 4.2.20, siendo $AF \approx 1$.

$$S_V^{1/f-MTJ} \left(\frac{V^2}{Hz} \right) = \gamma \frac{V^2}{Af} = \gamma \frac{I^2 R^2}{Af} \quad (4.2.20)$$

En el caso de las **SV** el valor del ruido eléctrico *flicker* tiene la expresión 4.2.21:

$$S_V^{1/f-SV} \left(\frac{V^2}{Hz} \right) = \frac{\gamma I^2 R^2}{N_c f} \quad (4.2.21)$$

donde γ es el parámetro *Hooge*, N_c es el número de portadores que participan en la corriente I y f es la frecuencia. El procedimiento para establecer el valor de N_c es considerar un portador por cada átomo en las capas de baja resistencia de la **SV**. Es una buena aproximación establecer que únicamente la capa libre (*free*), espaciadora (NM) e inmovilizada (*pinned*) contribuyen con sus electrones [FFCC07].

4.2.4.3 Ruido en los dispositivos **SV**

Los dispositivos **SV** incorporan los siguientes mecanismos de ruido:

- Ruido independiente de la frecuencia:
 1. Ruido Térmico
- Ruido dependiente de la frecuencia:
 1. Ruido *Flicker*

Las fuentes de ruido se suponen incorreladas debido a la naturaleza de los efectos físicos que producen el ruido. Así pues, la densidad espectral de ruido total está formada por la suma aditiva de las densidades espectrales de ruido térmico y *flicker*. En la figura 4.47 se muestra el equivalente circuitual de las fuentes de ruido de los dispositivos **SV**. De todas las fuentes

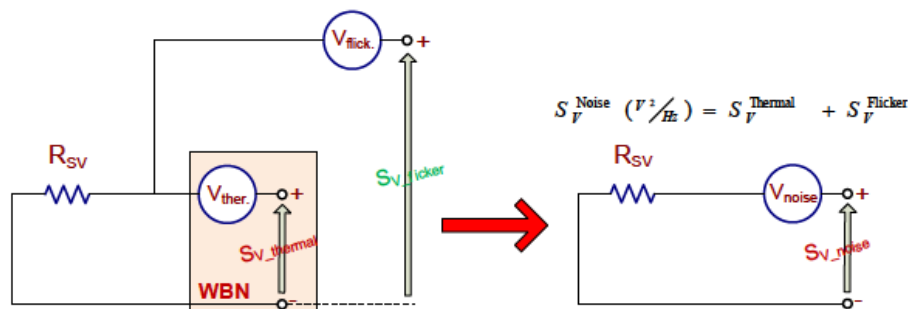


Figura 4.47 – Equivalente circuitual de las fuentes de ruido existentes en una **SV**

de ruido presentadas en los apartados anteriores, únicamente son de interés en el rango de

frecuencias de las aplicaciones de sensorización de corriente las siguientes:

$$S_V^{\text{Total}}\left(\frac{V^2}{\text{Hz}}\right) = S_V^{\text{Thermal}} + S_V^{1/f-SV} \quad (4.2.22)$$

La obtención de los datos experimentales se ha realizado usando la instrumentación descrita en el diagrama de la figura 4.48 [GdA09] donde la magnetorresistencia bajo prueba se polariza con una fuente DC de bajo ruido y mediante un amplificador de tensión de bajo ruido se conecta a un analizador de espectros.

El listado de código necesario para incluir la definición de las contribuciones de ruido en el modelo compacto de Verilog-A se muestra en el listado 4.7.

```

1 module magnetorresistencia( np, nn, ncp, ncn);
2
3   inout np,nn,ncp,ncn;
4   electrical np,nn,ncp,ncn;
5
6   branch(np,nn)    output_resistance;
7   ...
8   Iout=V(np, nn)/Magnetorres;
9   ...
10  if(Noise) begin
11    V(output_resistance) <+ white_noise(4.0 * `P_K * T * Magnetorres , "thermal")+
12    flicker_noise(gamma_hooge*pow(Iout*Magnetorres,2)/Nc, Ffe, "flicker-SV");
13  end

```

Listado 4.7 – Definición de contribuciones de ruido para dispositivo SV.

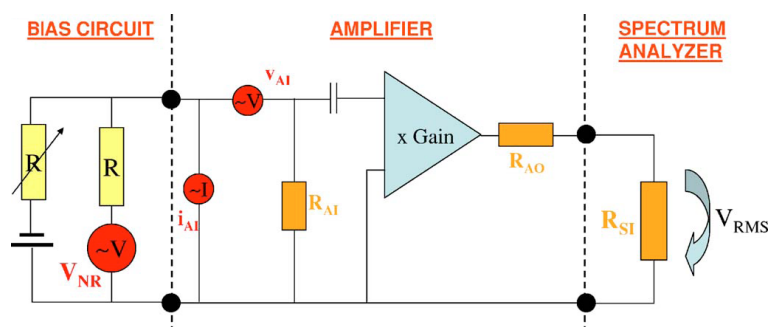


Figura 4.48 – Sistema de caracterización del ruido generado por una magnetorresistencia que es polarizada con una batería (para minimizar el ruido). El ruido es amplificado por un amplificador de tensión de bajo ruido Femto DLPVA-100-BLN-S y medido con un analizador de espectros de tiempo real Tektronix RSA3308A [GdA09].

En la figura 4.49(a) se muestra el espectro de ruido obtenido experimentalmente en INESC y descrito en [FFCC07] para el caso de un dispositivo SV de $2,5 \times 80 \mu\text{m}^2$ con una estructura laminar de Ta 20 Å / Ni₈₀Fe₂₀ 30 Å / Co₈₁Fe₁₉ 25 Å / Cu 26 Å / Co₈₁Fe₁₉ 25 Å / Mn₇₆Ir₂₄ 60 Å / Ta 30 Å / Ti₁₀W₉₀N 150 Å un número estimado de portadores $N_C \approx 2,48 \times 10^{11}$, un parámetro Hooge $\gamma \approx 1$, una resistencia $\approx 650 \Omega$ para una corriente de polarización $I_{BIAS} \approx 0,1 \text{ mA}$ [FFCC07]. En la figura 4.49(b) se presentan los resultados obtenidos en el simulador ADS® para un dispositivo de similares características. Para la obtención de los valores es necesario realizar un análisis AC, activar la generación de ruido en todos los dispositivos y elegir los nodos del circuito donde se pretende obtener la contribución de ruido. En la figura 4.50 se muestra la hoja de simulación de ADS® para la simulación de ruido. Los resultados son muy buenos y ajustan con precisión los datos experimentales.

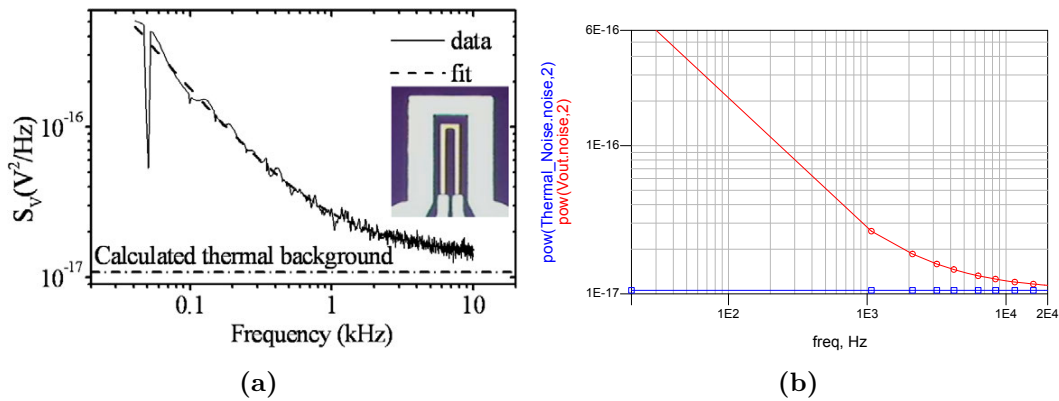


Figura 4.49 – Densidad espectral de ruido para un sensor *SV* de $2,5 \times 80 \mu\text{m}^2$ en régimen lineal con una resistencia de 650Ω y una corriente de polarización de 0.1 mA . Se muestran las contribuciones del ruido térmico en azul y $1/f$ en rojo [FFCC07] (a) y datos de simulación obtenidos del modelo compacto en Verilog-A para el mismo dispositivo (b).

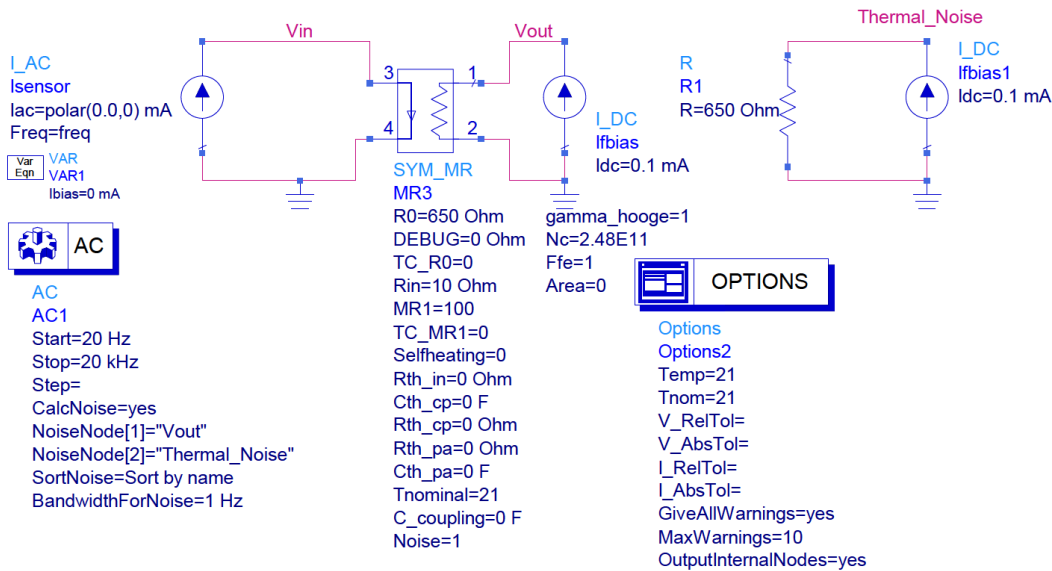


Figura 4.50 – Hoja de simulación de ADS® para el análisis de ruido en un dispositivo *SV*.

4.2.4.4 Ruido en los dispositivos MTJ

De modo similar a los dispositivos *SV*, las magnetorresistencias *MTJ* incorporan los siguientes mecanismos de ruido al incluir una barrera túnel en la estructura de capas:

- Ruido independiente de la frecuencia:
 1. Ruido Térmico
 2. Ruido *Shot*
- Ruido dependiente de la frecuencia:
 1. Ruido *Flicker*

Las fuentes de ruido se suponen también incorreladas debido a la naturaleza de los efectos físicos que producen el ruido. La densidad espectral de ruido total está formada por la suma

aditiva de las densidades espectrales de ruido blanco, tanto térmico como *shot*, y *flicker*. En la figura 4.51 se muestra el equivalente circuitual de las fuentes de ruido de los dispositivos **MTJ** que difieren de la **SV** en la adición del ruido *shot* generado en la barrera aislante. De

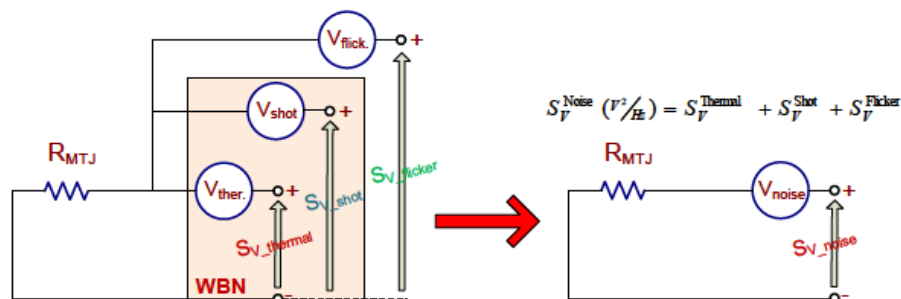


Figura 4.51 – *Equivalente circuitual de las fuentes de ruido existentes en una MTJ*

todas las fuentes de ruido presentadas en los apartados anteriores, únicamente son de interés en el rango de frecuencias de las aplicaciones de sensorización de corriente las siguientes:

$$S_V^{\text{Total}}\left(\frac{V^2}{\text{Hz}}\right) = S_V^{\text{Thermal}} + S_V^{\text{Shot}} + S_V^{1/f-SV} \quad (4.2.23)$$

La obtención de los datos experimentales se ha realizado con la instrumentación descrita en el diagrama de la figura 4.48 donde la magnetorresistencia bajo prueba se polariza con una fuente **DC** de bajo ruido y mediante un amplificador de tensión de bajo ruido se conecta a un analizador de espectros.

El listado de código necesario para incluir la definición de las contribuciones de ruido en el modelo compacto de **Verilog-A** se muestra en el listado 4.8. En este caso se han incluido las estructuras condicionales establecidas para modelar el ruido de los dispositivos **MTJ** y **SV**.

```

1 module magnetorresistencia( np, nn, ncp, ncn);
2
3   inout np, nn, ncp, ncn;
4   electrical np, nn, ncp, ncn;
5
6   branch(np, nn)    output_resistance;
7
8   ...
9   Iout=V(np, nn)/Magnetorres;
10  ...
11  if(Noise) begin
12    if(Area != 0) begin // Caso válido para MTJ
13      V(output_resistance) <+ white_noise(4.0 * `P_K * T * Magnetorres , "thermal")+
14        white_noise(2.0*`P_Q*abs(Iout)*pow(Magnetorres,2) , "Shot-MTJ")+
15        flicker_noise(gamma_hooge*pow(Iout*Magnetorres,2)/Area, Ffe, "flicker-MTJ");
16    end
17  else begin // Caso válido para SV
18    V(output_resistance) <+ white_noise(4.0 * `P_K * T * Magnetorres , "thermal")+
19      flicker_noise(gamma_hooge*pow(Iout*Magnetorres,2)/Nc, Ffe, "flicker-SV");
20  end
21 end

```

Listado 4.8 – *Definición de contribuciones de ruido para dispositivo MTJ.*

En la figura 4.52 se muestra el espectro de ruido obtenido experimentalmente en INESC para el caso de un dispositivo **MTJ** de barrera *MgO* con $\text{TMR} \approx 120\%$ y un producto $RA \approx 30 \Omega \mu\text{m}^2$, un parámetro *Hooge* $\alpha_H \approx 1,24 \times 10^{-9} \mu\text{m}^2$, una resistencia $\approx 9,5 \Omega$, un área de $1 \times 2 \mu\text{m}^2$, para corrientes de polarización I_{BIAS} de 3-0 mA [FFCC07]. En la

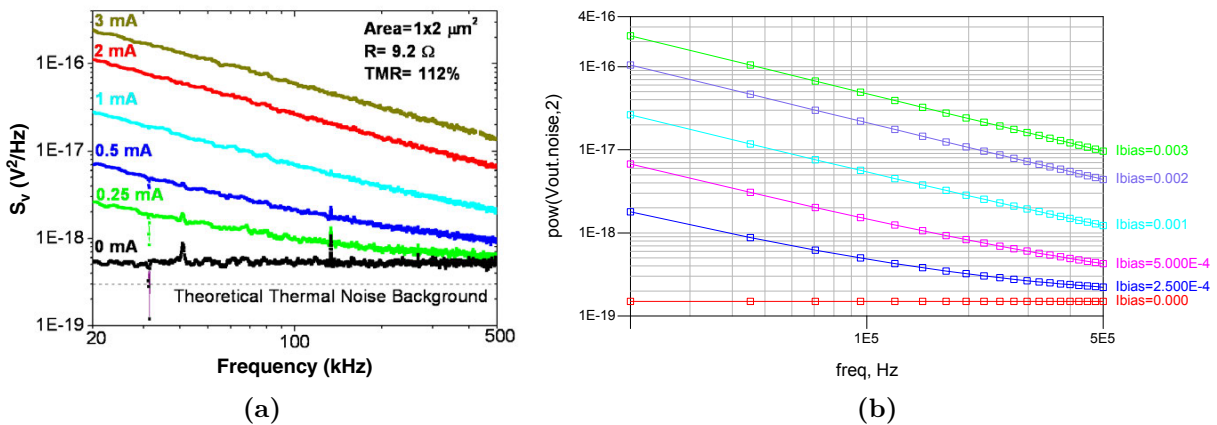


Figura 4.52 – Espectro de ruido para un sensor *MTJ* de barrera *MgO* con un área de $1 \times 2 \mu\text{m}^2$ con una resistencia de 9.2Ω en función de la corriente de polarización de 3-0 mA [FFCC07]. La medida se realizó sin aplicación de campo magnético en la zona de mínima resistencia para evitar la contribución del ruido $1/f$ magnético (a) y datos de simulación obtenidos del modelo compacto en *Verilog-A* para el mismo dispositivo (b).

figura 4.53 se muestra el listado de contribuciones de las diferentes fuentes de ruido que implementan el modelo compacto de un dispositivo magnetorresistivo, los datos mostrados corresponden a una corriente de polarización de 3 mA a 500 KHz. La figura 4.52(b) presenta los resultados obtenidos en el simulador *ADS*® para un dispositivo de similares características. En este caso se ha añadido el ruido *shot*. En la figura 4.54 se muestra la hoja de simulación de *ADS*® para la simulación de ruido. Como en el caso anterior, los resultados de simulación ajustan perfectamente con los datos experimentales. Se comprueba que el modelado de

4

index	Vout.NC.vnc	Vout.NC.type	Vout.NC.name
Ibias=0.003, freq=500.0 kHz			
0	3.111 nV	_total	_total
1	0.0000 V	I_Source	I_Sensor
2	3.111 nV	magnetorresistencia	MR1
3	285.2 pV	magnetorresistencia	MR1.Shot-MTJ
4	3.073 nV	magnetorresistencia	MR1.flicker-MTJ
5	0.0000 V	magnetorresistencia	MR1.flicker-SV
6	386.6 pV	magnetorresistencia	MR1.thermal

Figura 4.53 – Listado de contribuciones de ruido de las diferentes fuentes asociadas al dispositivo magnetorresistivo para 3 mA de corriente de polarización a 500 KHz obtenidas en *ADS*®.

ruido de los dispositivos magnetorresistivos *SV* y *MTJ* realizado en *Verilog-A* proporciona unos resultados muy fiables y permite añadir estos efectos a los circuitos de las aplicaciones de sensores de corriente.

4.2.4.5 Ruido en puentes de Wheatstone.

Una vez se ha comprobado que el modelado de los dispositivos individuales es correcto y que los datos obtenidos del simulador se ajustan a los experimentales, se pueden realizar simulaciones de ruido para puentes de Wheatstone de 4 elementos variables en los que cada uno de ellos añade su ruido, de modo individual, al ruido total del circuito que constituye el puente de Wheatstone. Para ello se creará un elemento formado por cuatro magnetorresistencias *MTJ* similares a las descritas en la figura 4.52, que se polarizarán con una corriente

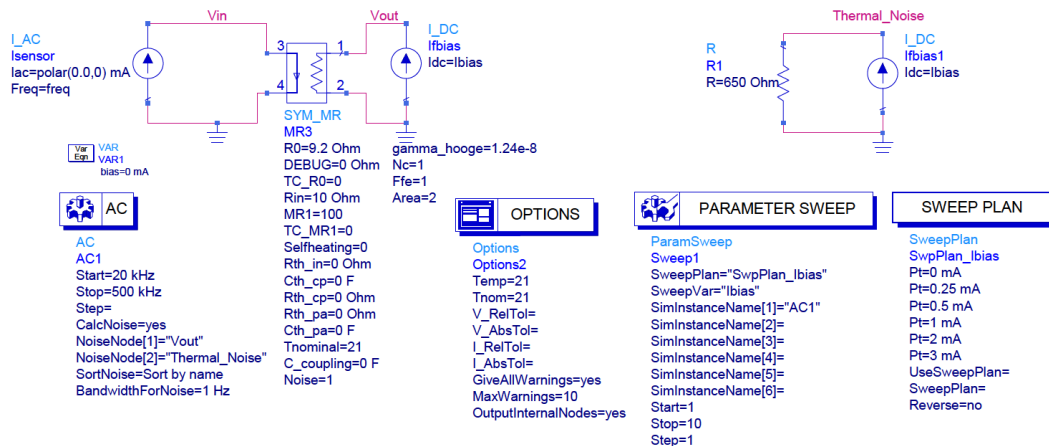


Figura 4.54 – Hoja de simulación de ADS® para el análisis de ruido en un dispositivo MTJ.

DC de 0.1 mA y donde la corriente I_{SENSOR} se establecerá con circulación serie a través de los cuatro elementos, ver figura 4.55

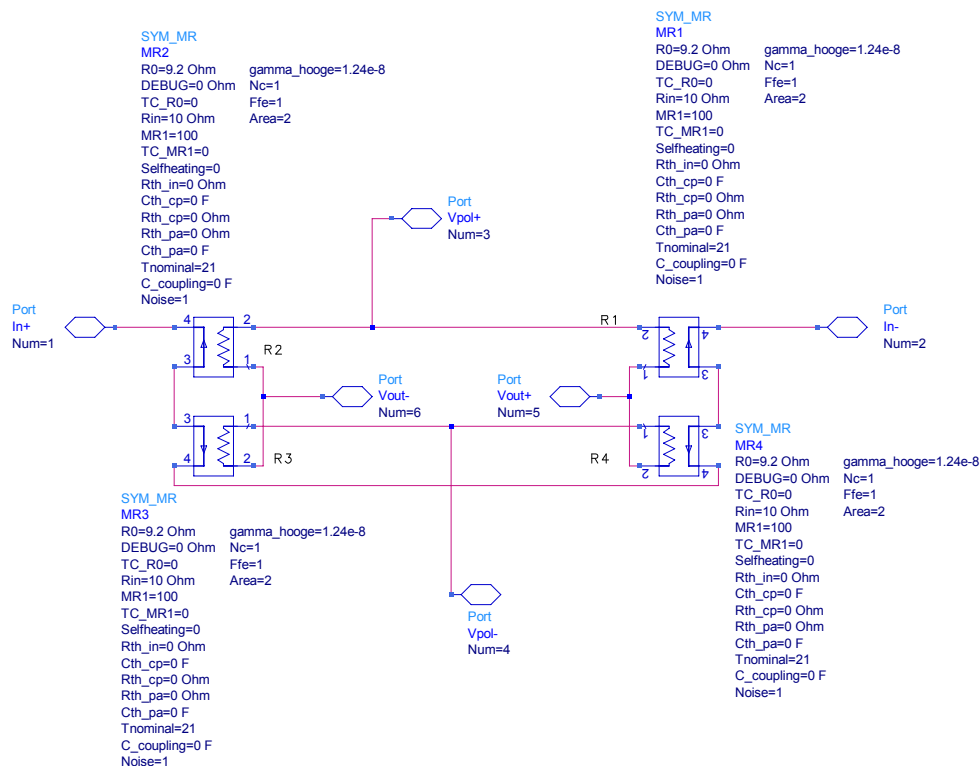


Figura 4.55 – Configuración de un puente de Wheatstone formado por 4 elementos sensibles MTJ de barrera MgO con un área de $1 \times 2 \mu\text{m}^2$ con una resistencia de 9.2Ω .

En la figura 4.57 se muestran una comparativa sobre el ruido obtenido en la salida del puente, terminal $VS+$ de la figura 4.56, de dispositivos MTJ y una única magnetorresistencia MTJ similar a la de la figura 4.52. El ruido a la salida del puente es menor que en el caso de la magnetorresistencia simple y esto se debe a la conexión paralela de las fuente de ruido.

En este sentido, en cuanto al ruido, la configuración para instrumentación del puente de

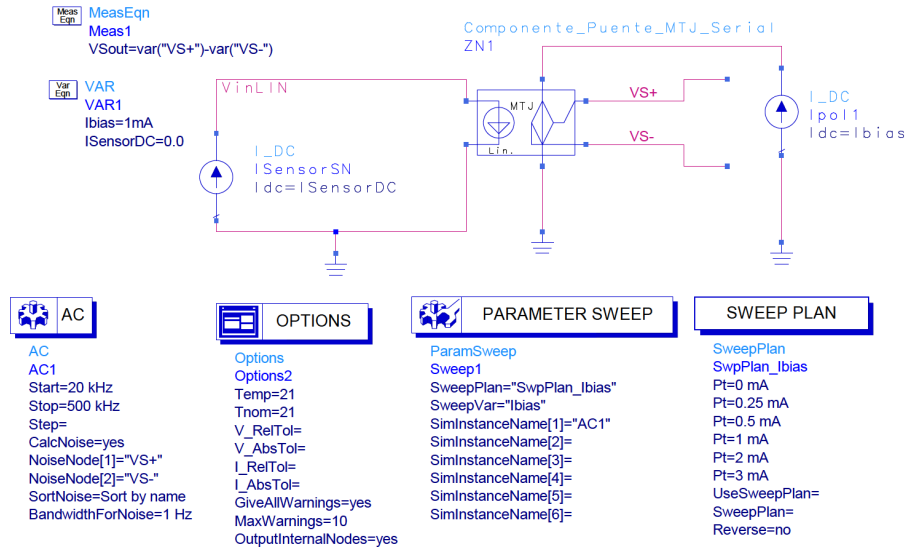


Figura 4.56 – Hoja de simulación de ADS® para el análisis de ruido en un puente de Wheatstone configurado por cuatro sensores MTJ. Los terminales VS+ y VS- corresponden a los terminales V_out(+) y V_out(-) de los puentes representados anteriormente.

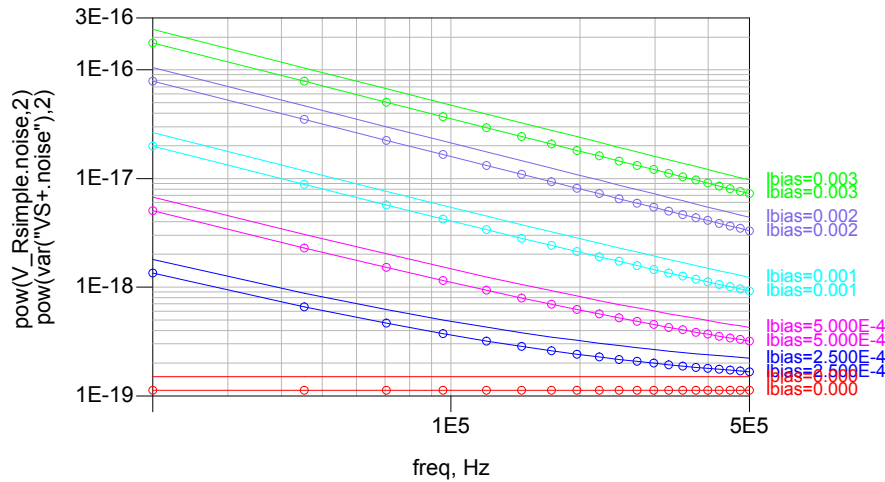


Figura 4.57 – Comparativa del ruido generado por una única magnetorresistencia MTJ (datos en líneas) y el ruido obtenido a la salida del puente de Wheatstone (datos en símbolos) para las mismas condiciones de excitación.

Wheatstone es más ventajosa. En cualquier caso, para cada tipo de configuración de medida se debe estudiar mediante simulación, las condiciones que permiten mejorar la relación señal-ruido.

CAPÍTULO

5

MODELADO DE APLICACIONES CON SENSORES MAGNETORRESISTIVOS DE CORRIENTE.

5.1 Conversores resistencia a frecuencia.

El uso generalizado de sistemas de medida basados en microcontroladores ha hecho necesaria la proliferación de circuitos conversores de tensión a frecuencia como alternativa al uso clásico de conversores analógico-digital tanto externos como internos. En la figura 5.1(a) se presenta el esquema clásico de digitalización de la señal proveniente de un sensor, por el contrario en la figura 5.1(b) se muestra el sistema de medida basado en la conversión realizada mediante los temporizadores internos del microcontrolador una vez la señal del sensor ha sido transformada en señal binaria. Este procedimiento presenta algunas ventajas, entre las que cabe destacar: alta inmunidad al ruido, facilidad de transmisión de la señal binaria, gran rango dinámico disponible y alta precisión disponible en la conversión frecuencia-código. La relación de compromiso entre velocidad y precisión se puede superar mediante el uso de técnicas eficientes de detección de frecuencias [Mei08]. Debido al auge en el uso de microprocesadores y microcontroladores para procesamiento digital de señal, se han diseñado, simulado y fabricado diferentes circuitos basados en conversores resistencia-frecuencia para su uso con sensores magnetorresistivos de corriente tanto simples como en puente [MW95,

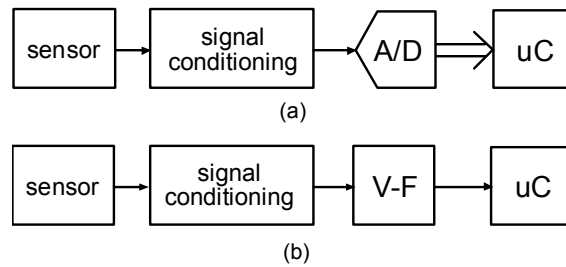


Figura 5.1 – Sensor inteligente: conexión (a) convencional, (b) usando convertidores tensión-frecuencia V/F con conversión A/D implementada en el microcontrolador.

[Kal00], de los que se obtiene fácilmente una señal digital con frecuencia proporcional a la corriente que atraviesa el sensor.

5.1.1 Conversión R-F mediante magnetorresistencia simple.

En este apartado se van a presentar los resultados obtenidos en el diseño de un conversor de corriente a frecuencia para un sensor de corriente basado en una magnetorresistencia simple, figura 5.2, que emplea dos amplificadores operacionales.

El uso del sensor magnetorresistivo nos ha permitido mejorar el diseño inicial del conversor resistencia-frecuencia de alta resolución [Kal00] al convertirlo en un conversor corriente-frecuencia.

El circuito consiste en un oscilador basado en un amplificador de puente y un comparador. Las resistencias R_1 , R_2 , MR y la combinación serie de los elementos R_3 y C forman las cuatro ramas del puente que con el amplificador operacional (OAmpl) OA_1 forma el amplificador de puente. MR representa el circuito equivalente del sensor de corriente SV07 usado. Asumiendo que los amplificadores operacionales son ideales se puede establecer que el circuito se comporta como un oscilador si se verifica que MR tiene un valor mayor que $R_1 \cdot R_3 / R_2$.

Las formas de onda de la tensión de entrada V_C del comparador OA_2 y la de salida (output) V_o se muestran en la figura 5.3, donde el intervalo de tiempo T_1 y T_2 corresponden al tiempo en el que la tensión de salida V_o alcanza los valores de tensión V_1 y $-V_2$. La tensión de entrada del comparador V_C está comprendida entre los valores $\pm \alpha \cdot (V_1 + V_2)$, donde α es igual a $(R_2 \cdot MR - R_1 \cdot R_3) / [R_2 \cdot (R_1 + MR)]$. La expresión teórica del periodo de oscilación se muestra en la ecuación 5.1.1 y ha sido calculada para los elementos $R_1 = 1 \text{ K}\Omega$, $R_2 = 22 \text{ K}\Omega$, $R_3 = 12 \text{ K}\Omega$, $R_4 = 680 \Omega$, $C = 0,33 \mu\text{F}$ y la magnetorresistencia $SV07_{R3}$.

$$T = \frac{(1 + \mu)^2 (R_2 MR - R_1 R_3) RC}{\mu R_1 (R_2 + R_3)} \text{ where } \mu = \frac{V_{\text{out}}^{\text{max}}}{V_{\text{out}}^{\text{min}}} \quad (5.1.1)$$

Se ha conseguido un ajuste muy bueno entre los datos de simulación, tabla 5.1 y los datos experimentales, figura 5.4 donde se presenta el caso de un sensor de corriente excitado por una corriente DC de -10, 0 y 10 mA respecto al tiempo.

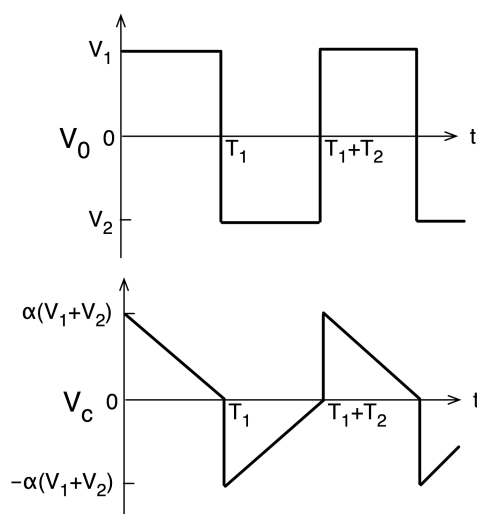
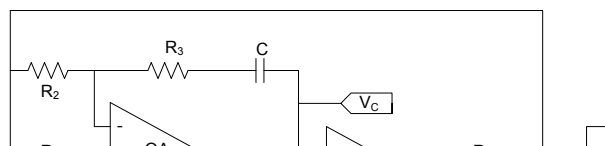


Figura 5.3 – Tensión de entrada (V_C) y salida (V_0) del comparador.

La principal ventaja que aporta este circuito es que la variación de resistencia producida en la magnetorresistencia MR dada por δMR , es convertida proporcionalmente a una variación del periodo de la señal, δT :

$$\delta T = \frac{(1 + \mu)^2 R_2 C \delta MR}{\mu R_1} \quad (5.1.2)$$

De la ecuación 5.1.2, se puede deducir que la sensibilidad del circuito puede ser fácilmente reajustada mediante la variación de R_1 y R_2 , teniendo en cuenta que R_3 no afecta a la sensibilidad y puede ser usada para ajustar el periodo de la señal.

Tabla 5.1 – Periodo de la señal para diferentes corrientes de entrada [RRR⁺ 11a].

(ms)	I=-10 mA	I=0 mA	I=10 mA
Teórica	15.4	15.8	16.2
Experimental	15.7	16.1	16.5
Simulación	15.8	16.1	16.6

En la figura 5.5 se muestra la hoja de simulación de ADS[®] donde se presenta el circuito conversor junto con el modelo compacto de magnetorresistencia simple SV07_{R2} desarrollado

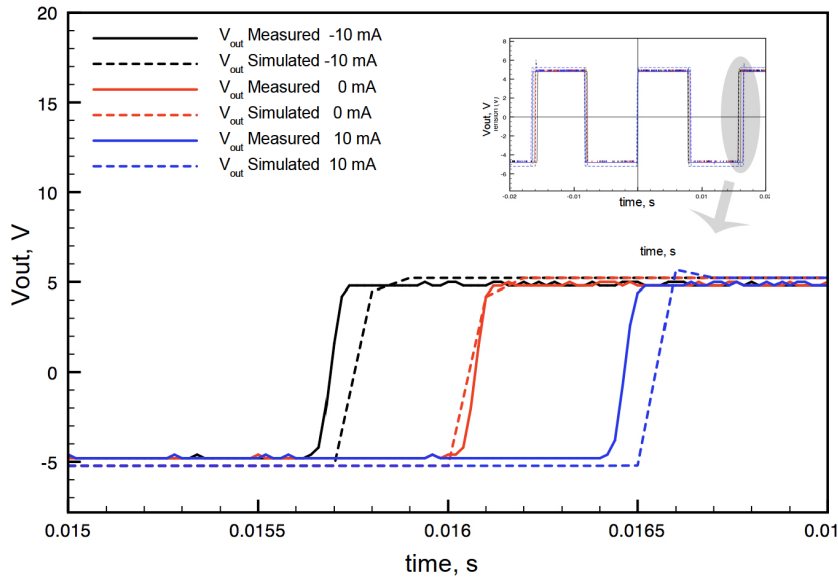


Figura 5.4 – Valores experimentales y de simulación para la tensión de salida V_O en función del tiempo para el sensor de corriente $SV07_{R3}$ [RRR⁺11a].

en . Las medidas experimentales obtenidas para los valores de corriente del sensor se pre-

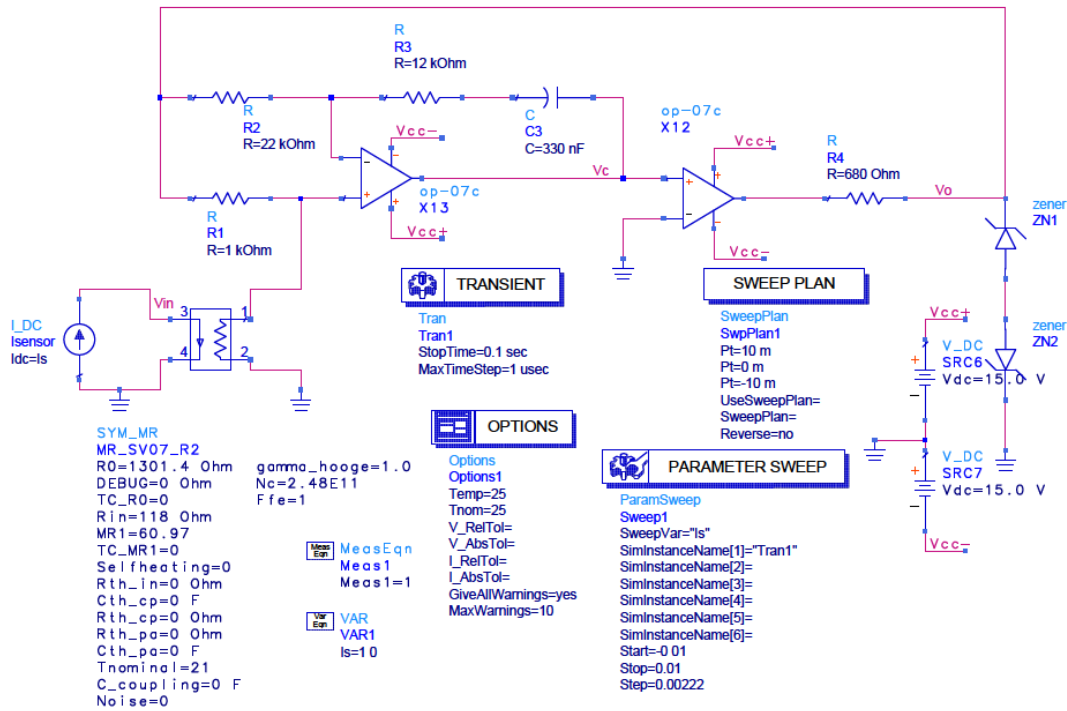
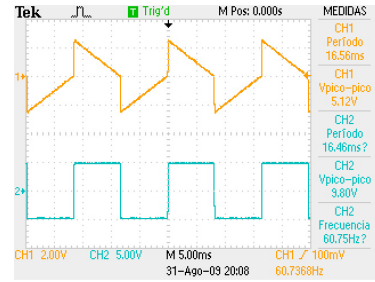


Figura 5.5 – Hoja de simulación en ADS para el circuito convertidor corriente-frecuencia y el modelo compacto del sensor magnetorresistivo $SV07_{R2}$.

sentan en la figura 5.6 donde se detallan los casos (a) -10 mA, (b) 0 mA y (c) 10 mA. Los resultados obtenidos son muy similares a los esperados teóricamente. La frecuencia normalizada de la tensión de salida del convertidor para el caso del sensor $SV07_{Ri}$ se presenta en la figura 5.7 donde se aprecia la relación lineal existente entre la frecuencia de oscilación y la



(c)

Figura 5.6 – Tensión de entrada (V_C) (CH_1) y de salida (V_o) (CH_2) del comparador para corrientes del sensor de: (a) -10 mA , (b) 0 mA y (c) 10 mA [RRR⁺ 11a].

corriente de excitación del sensor, lo que demuestra la coherencia de la aplicación. Del mismo modo, se han realizado simulaciones para las diferentes magnetorresistencias integrantes del dispositivo $SV07$ atendiendo a sus diferentes parámetros MR_i . La implementación del circui-

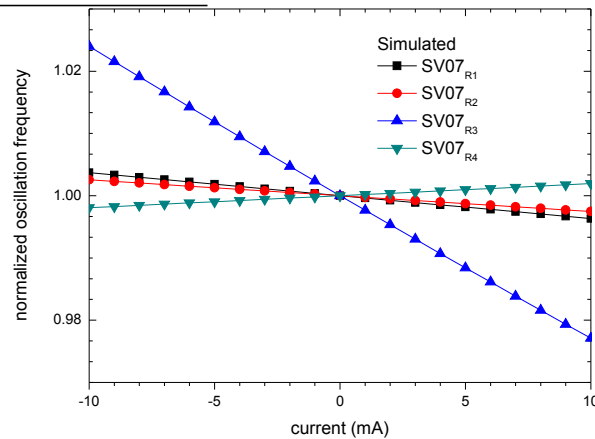


Figura 5.7 – Frecuencia de oscilación normalizada de la tensión de salida versus corriente principal del sensor $SV07_{Ri}$.

to se ha realizado con [Altium®](#) usando mayoritariamente elementos [SMD 0805](#) en una [PCB](#) a doble cara. El circuito fabricado cuyo esquema se presenta en la figura 5.8 permite elegir mediante la selección, a través de los jumpers J15 y J19, de las cuatro magnetorresistencias $SV07_{Ri}$ existentes en el puente $SV07$, posibilitando realizar todas las medidas experimentales con la misma placa. Esta solución de diseño aporta gran flexibilidad en las medidas y disminuye en gran medida el tiempo necesario para realizar el proceso de caracterización del conversor corriente-frecuencia para todas las magnetorresistencias del chip.

En la figura 5.9(a) se muestra la cara superior y la inferior en 5.9(b) de la [PCB](#) utilizada. El proceso de fabricación se ha realizado por aislado mediante el uso de la máquina de prototipado LPKF S62 del Dpto. de Electrónica de la Univ. de Granada. La vista superior de la [PCB](#) una vez montada y cableada se muestra en la figura 5.10.

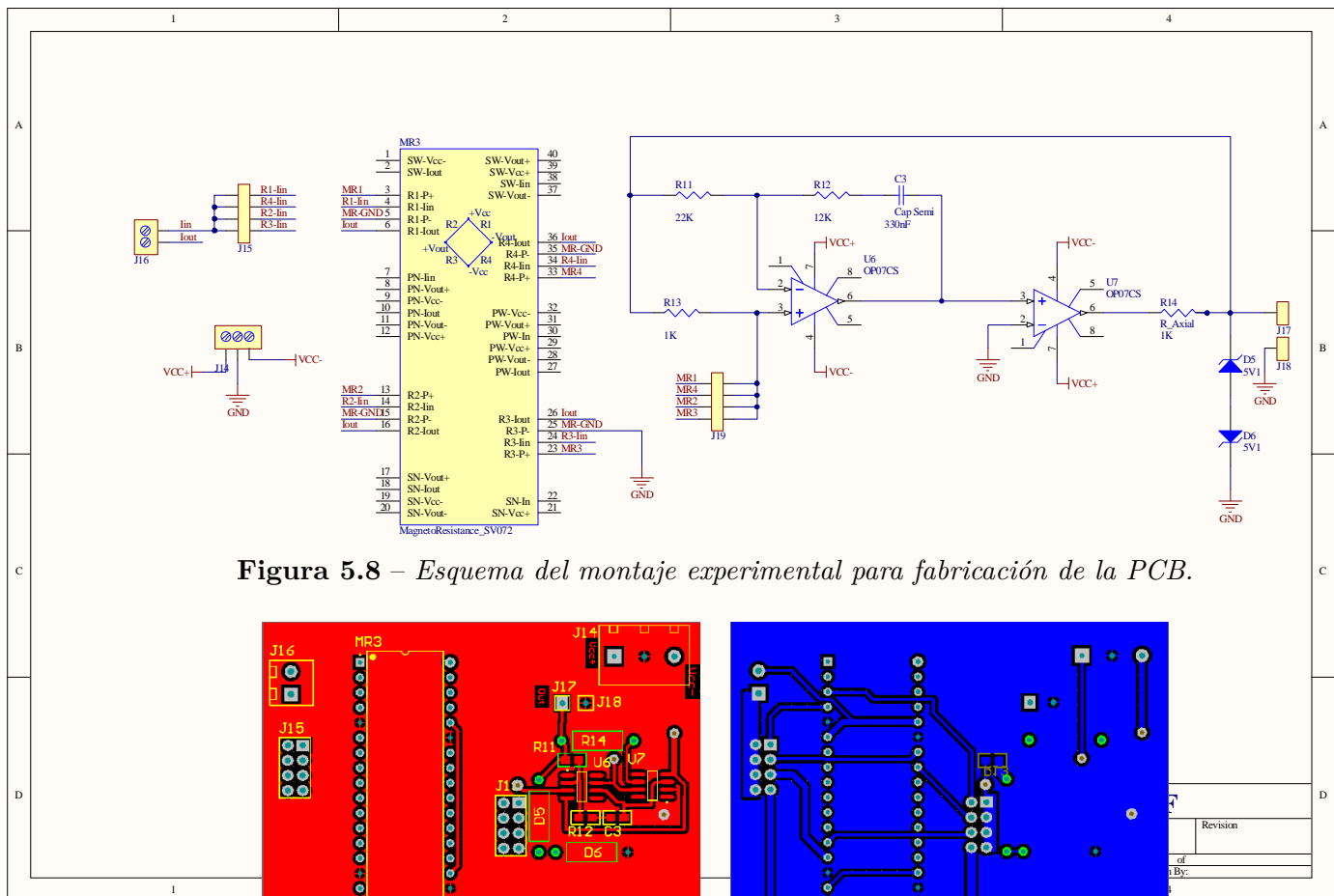
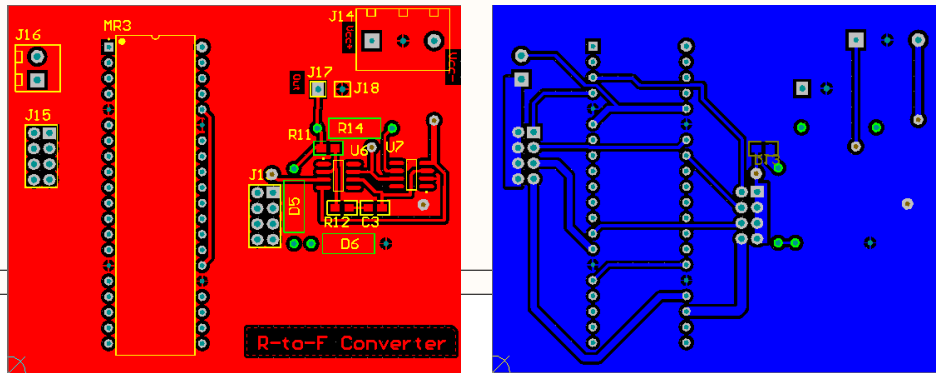


Figura 5.8 – Esquema del montaje experimental para fabricación de la PCB.



(a)

(b)

Figura 5.9 – Placa de circuito impreso para el convertor corriente a frecuencia para magnetorresistencia simple: cara superior (a), cara inferior (b).

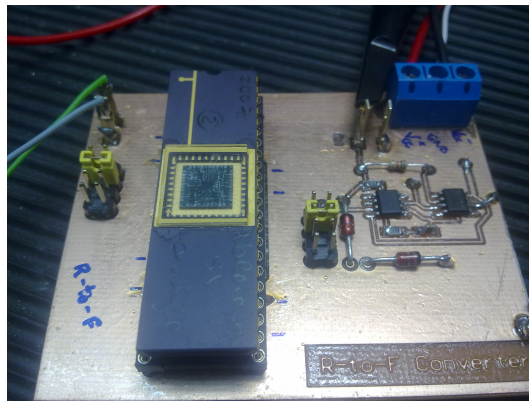


Figura 5.10 – Montaje experimental.

5.1.2 Conversión R-F mediante puentes magnetorresistivos.

En este caso hemos utilizado un circuito oscilador similar al anterior pero con la diferencia de permitir el uso de puentes magnetorresistivos. El diseño final implementado está basado en el trabajo publicado por [FGMT97] donde se ha eliminado la parte de modulación en ancho de pulso.

En la figura 5.11 se muestra la hoja de simulación de ADS® donde se presenta el circuito conversor junto con el modelo compacto de puente magnetorresistivo *SV07* desarrollado en Verilog-A. El funcionamiento del circuito de la figura 5.11 se basa en el oscilador de rela-

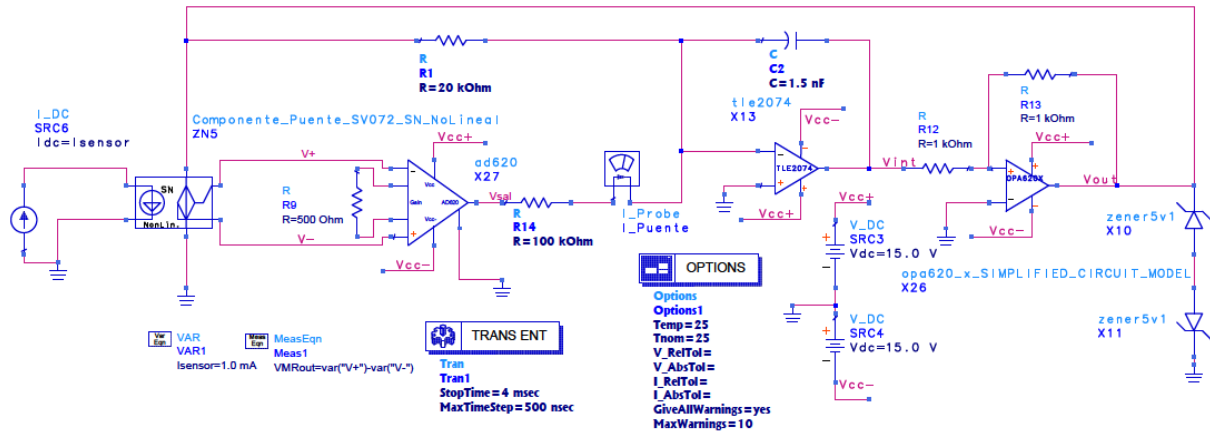


Figura 5.11 – Hoja de simulación en ADS para el circuito conversor corriente-frecuencia y el modelo compacto del puente magnetorresistivo *SV07*.

jación implementado con el amplificador operacional X_{13} y el comparador X_{26} . El puente magnetorresistivo ZN_5 está excitado por la tensión cuadrada generada a la salida del comparador X_{26} . La tensión diferencial del puente alimenta el amplificador diferencial X_{27} que con su ganancia G , la convierte a una corriente cuadrada de valor I_x que se añade algebraicamente en el nodo sumador del integrador, formado por R_1 , C_2 y X_{13} , a la corriente I_0 que circula por R_1 . Teniendo en cuenta que los valores de saturación para $V_{out} = V_{out}^{\max} = V_{out}^{\min}$ son iguales por la acción de los diodos *zener* (X_{10} y X_{11}), la tensión de salida del integrador tiene forma triangular con una amplitud pico a pico de $2V_{out}R_{12}/R_{13}$. Independientemente del valor de V_{out} , la tensión a la salida del integrador V_{int} tiene un ciclo de trabajo del 50% y una frecuencia representada por la ecuación 5.1.3.

$$f = \frac{1}{4R_1C_2} \frac{R_{13}}{R_{14}} \left[1 + \left(\frac{R_4}{R_4 + R_3} - \frac{R_2}{R_2 + R_1} \right) G \frac{R_1}{R_9} \right] = f_0 \left(1 + kG \frac{R_1}{R_9} \right) \quad (5.1.3)$$

La implementación del circuito se ha realizado con Altium® usando mayoritariamente elementos SMD 0805 en una PCB a doble cara. El circuito fabricado cuyo esquema se presenta en la figura 5.12 permite elegir mediante la selección de los *jumpers* J1, J9, J11 y J12 los cuatro puentes magnetorresistivos *SV07 – SW*, *SV07 – SN*, *SV07 – PW* y *SV07 – PN* existentes en el chip, lo que permite realizar todas las medidas experimentales con la misma placa. Esta solución de diseño aporta gran flexibilidad en las medidas y disminuye el tiempo necesario para realizar el proceso de caracterización del conversor corriente-frecuencia para todos los puentes del chip.

En la figura 5.13(a) se muestra la cara superior y la inferior en 5.13(b) de la PCB fabricada. El proceso de fabricación se ha realizado por aislado mediante el uso de la máquina de prototipado LPKF S62 del Dpto. de Electrónica de la Universidad de Granada.

Los resultados experimentales se muestran en la figura 5.14(b) para los puentes mag-

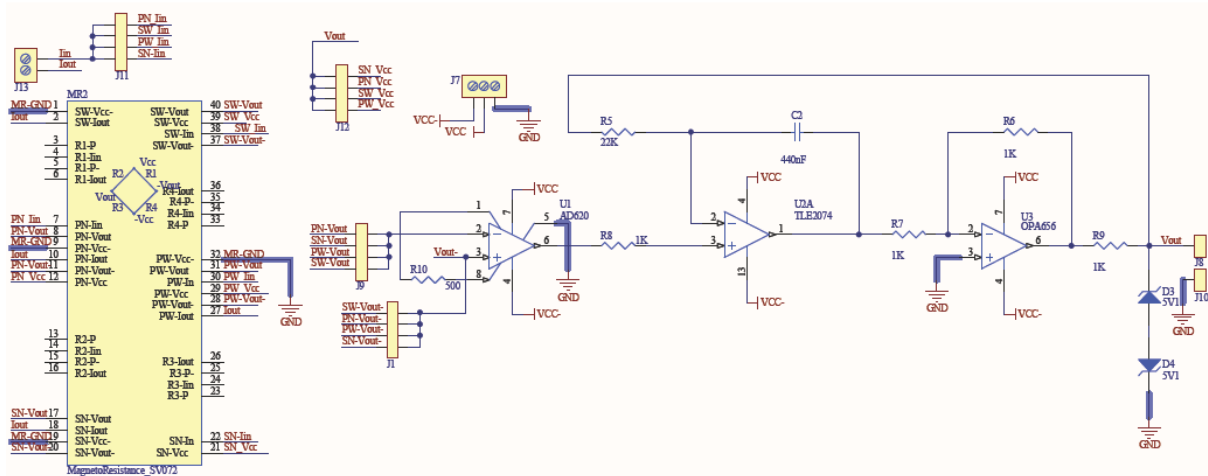
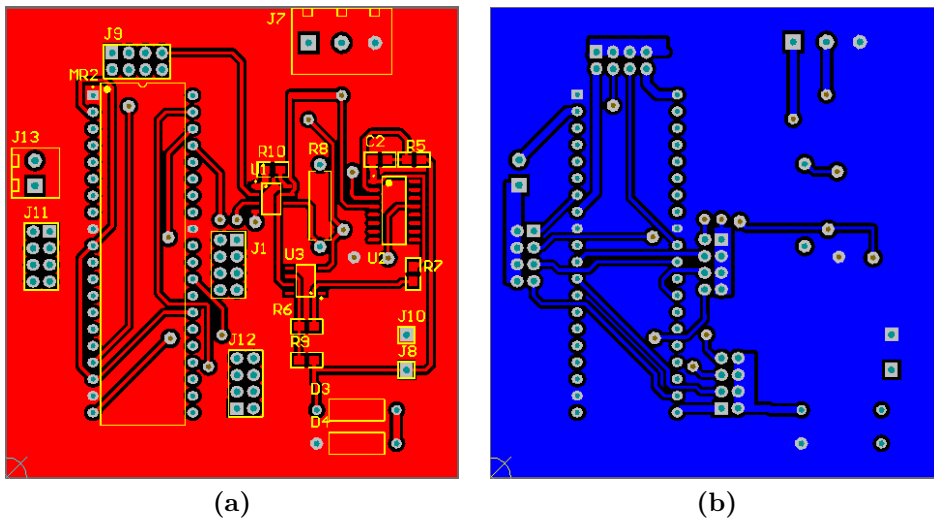


Figura 5.12 – Esquema del montaje experimental para fabricación de la PCB.



(a)

(b)

Figura 5.13 – Placa de circuito impreso para el convertor corriente a frecuencia para puente magnetorresistivo: cara superior (a), cara inferior (b).

netorresistivos de pista conductora estrecha (*narrow*) tanto paralelos como serie y en 5.14(b) para pista conductora ancha (*wide*). Para ambos casos, la respuesta corriente-frecuencia es lineal. El comportamiento real es similar al obtenido mediante simulación pero debido a las tolerancias de los elementos pasivos y de los circuitos integrados los resultados numéricos varían levemente.

5.2 Convertidor Generalizado de Impedancias (GIC)

Los convertidores de impedancia son circuitos activos RC diseñados para simular elementos dependientes de la frecuencia, como por ejemplo bobinas, para ser usados en filtros de frecuencia. El convertidor generalizado de impedancia, **GIC**, es una configuración de convertidor de impedancias que puede ser utilizada para simular inductancias y para sintetizar impedancias. Es usado desde 1969 como un método de sintetización de inductores utilizando

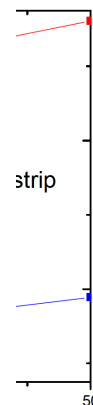
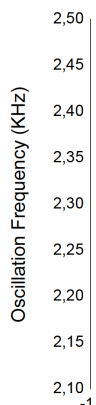


Figura
puentes
resister
cia. Su
muestr

l de los
luctan-
que se

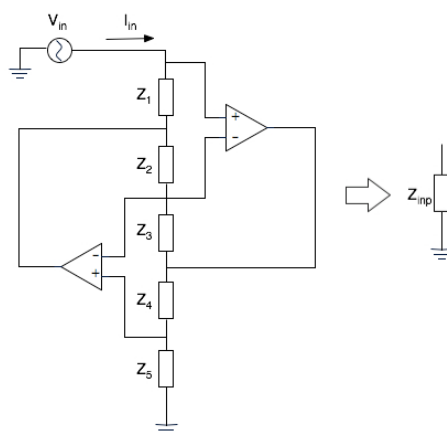


Figura 5.15 – *Convertidor generalizado de impedancias, GIC [MME08].*

Considerando los amplificadores operacionales como ideales, se obtiene que la impedancia de entrada del circuito viene dada por:

$$Z_{inp} = \frac{Z_1 \cdot Z_3 \cdot Z_5}{Z_2 \cdot Z_4} \tag{5.2.1}$$

Por lo que la impedancia total depende de qué valores de resistencias o condensadores asignemos a las impedancias $Z_1 - Z_5$. La flexibilidad ofrecida por estas cinco impedancias permite obtener una impedancia de entrada determinada para cada aplicación específica, por ejemplo, podemos obtener una inductancia sustituyendo Z_2 por un condensador, del tal forma



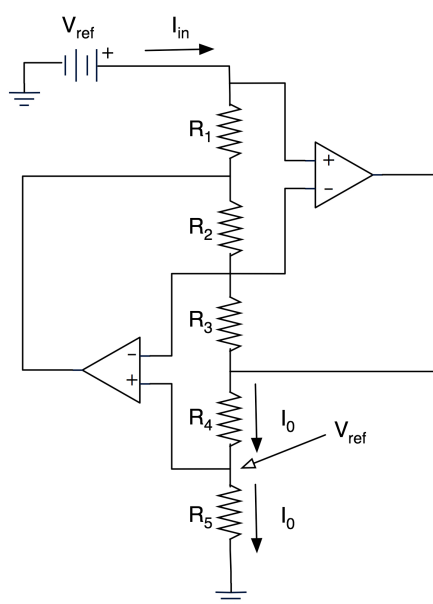


Figura 5.16 – GIC en régimen DC.

nales ideales, tenemos una tensión igual a V_{ref} en todas las entradas de los amplificadores operacionales. Podemos fijar entonces la corriente I_0 fijando a su vez V_{ref} y R_5 , con lo que la corriente que circula por R_4 será la misma, I_0 :

$$I_{R4} = I_0 = \frac{V_{\text{ref}}}{R_5} \quad (5.2.3)$$

La resistencia R_4 está alimentada por una corriente constante, I_0 [MBE05], [RCR+05]. R_4 puede ser reemplazada por un sensor o un grupo de sensores conectados en serie, de tal forma que todos los sensores compartirán la misma corriente. También podemos sustituir R_4 por un sensor en puente de Wheatstone, con cuatro resistencias activas, $R = R_0 \cdot (1 \pm x(I_{\text{SENSOR}}))$. La tensión de salida del sensor será una señal diferencial con componentes en modo común y diferencial. Se puede deducir que la tensión en modo común puede minimizarse en este caso usando una baja tensión de referencia, V_{ref} , pudiéndola conseguir con un ajuste apropiado de R_5 para una I_0 dada.

Para prevenir derivas térmicas, la corriente de entrada, I_{in} , debe ser lo más baja posible. Una forma de conseguirlo es seleccionando altos valores de resistencia para R_1 y R_3 , y bajos valores para R_2 y R_4 , pero teniendo en cuenta que las resistencias vistas desde las entradas de los dos amplificadores operacionales deben de estar emparejadas como se explicó en el apartado 4.2.3.

Una última posibilidad es la de utilizar el GIC como amplificador de corriente, en la cual podemos establecer una relación entre la corriente de entrada, I_{in} , y la corriente por R_5 , I_0 , tal que:

$$I_0 = I_{in} \frac{R_1 \cdot R_3}{R_2 \cdot R_5} \quad (5.2.4)$$

5.2.2 Resultados

Una vez vistos los aspectos teóricos de los convertidores generalizados de impedancias, hacemos uso de uno de ellos, en estructura de polarización DC, para proporcionar una corriente constante a un sensor de corriente magnetorresistivo. En nuestro caso, utilizaremos un sensor SV07 alimentado con una corriente de 1 mA mediante un GIC polarizado en continua, fijando $V_{ref} = 2,5 V$ y $R_5 = 2,5 K\Omega$.

El sensor producirá una salida en tensión diferencial que será proporcional a una corriente medida (I). El circuito usado para la simulación y que incluye el modelo compacto en Verilog-A se muestra en la figura 5.17.

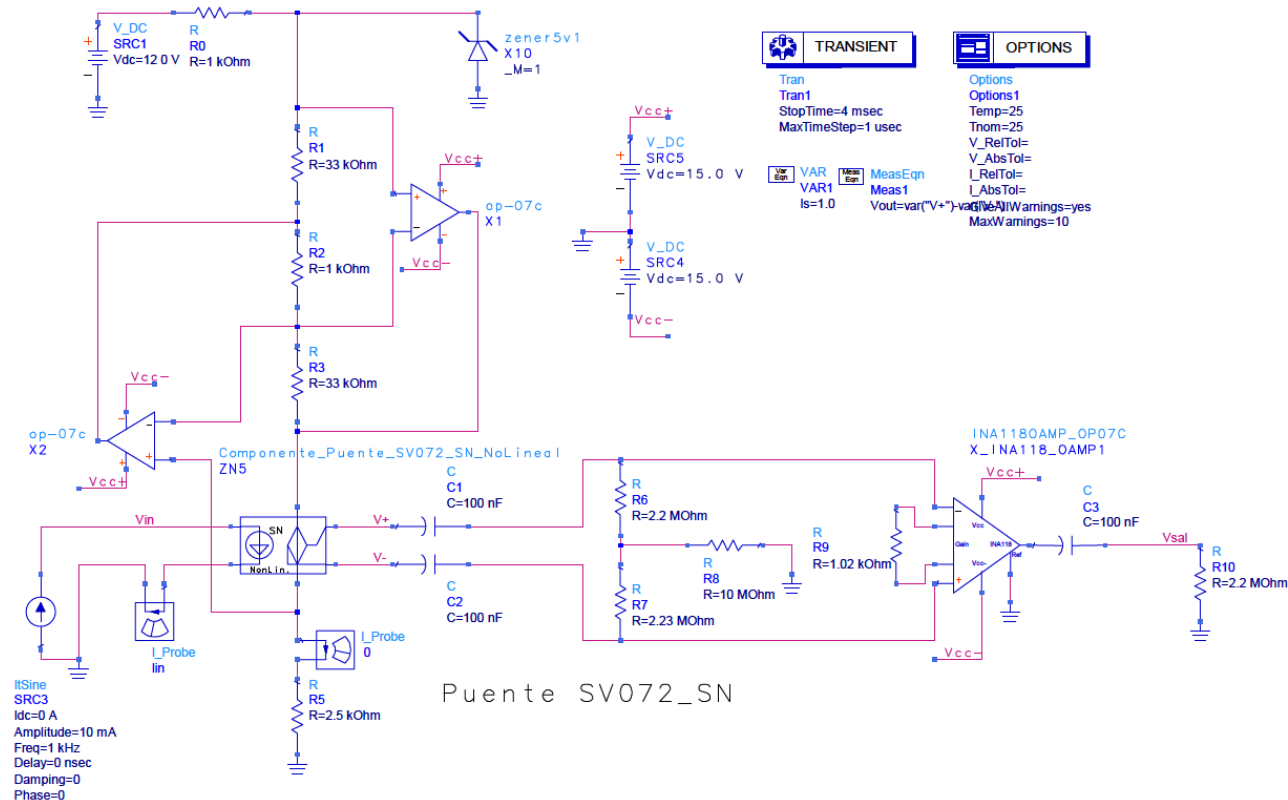


Figura 5.17 – Esquemático de polarización de un sensor SV07 mediante un GIC.

La salida del sensor es filtrada paso alto por un filtro diferencial analógico [CPA96], asegurándonos un gran CMRR. La ganancia del amplificador de instrumentación es fijada en 50 para obtener un nivel de tensión significativo. Otro filtro es colocado a la salida del amplificador de instrumentación para eliminar la tensión de *offset* introducida en la sonda de medida por el amplificador. Las simulaciones se han realizado en ADS[®] usando el modelo compacto para los sensores magnetoresistivos implementado en Verilog-A. En la figura 5.18 se muestran los resultados obtenidos al introducir una excitación de corriente de baja frecuencia.

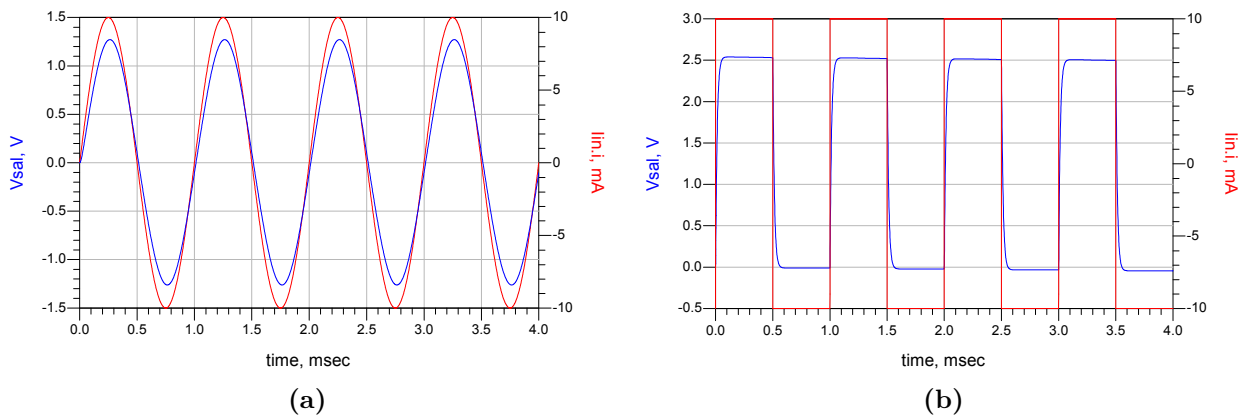


Figura 5.18 – Simulaciones de la tensión de salida para excitaciones de corriente de entrada: sinusoidal (a), cuadrada (b).

Los resultados experimentales obtenidos del circuito fabricado, figura 5.21, usando como sensores de corriente el SV07_PN se presentan en la figura 5.19 donde se observa la tensión de salida normalizada en función del tiempo para señales de corriente de entrada sinusoidales y cuadradas de 1 KHz y 10 mA de amplitud. La implementación del circuito se ha realizado

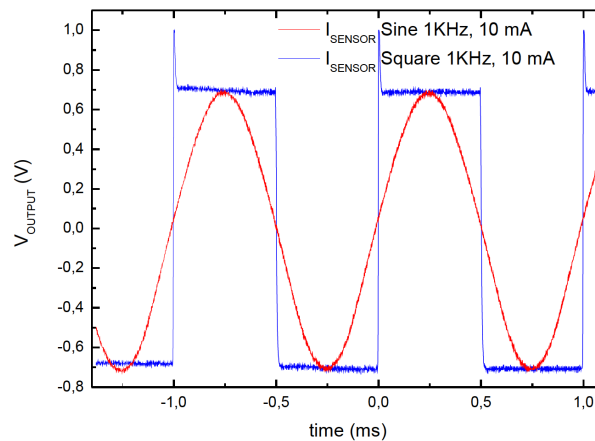


Figura 5.19 – Resultados experimentales para señal cuadrada y sinusoidal de 1 KHz y 10 mA de amplitud.

con Altium[®] usando también mayoritariamente elementos SMD 0805 en una PCB a doble cara como en el apartado anterior. El circuito fabricado cuyo esquema se presenta en la figura 5.20 permite elegir mediante la selección de los *jumpers* J2, J4, J7 y J8, los cuatro

puentes magnetorresistivos SV07-SW, SV07-SN, SV07-PW y SV07-PN existentes en el chip, lo que permite realizar todas las medidas experimentales con la misma placa. Esta solución de diseño aporta gran flexibilidad en las medidas y disminuye en gran medida el tiempo necesario para realizar el proceso de caracterización del conversor corriente-frecuencia para todos los puentes del chip. En la figura 5.21(a) se muestra la cara superior y la inferior

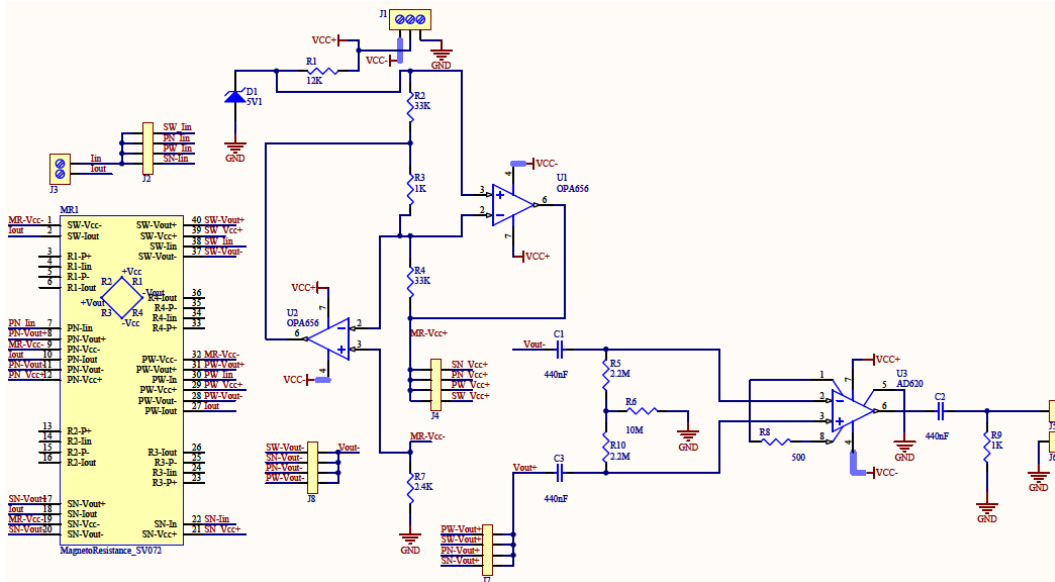


Figura 5.20 – Esquema del montaje experimental para fabricación de la PCB.

en 5.21(b). El proceso de fabricación se ha realizado por aislado mediante el uso de la máquina de prototipado LPKF S62 del Dpto. de Electrónica de la Univ. de Granada.

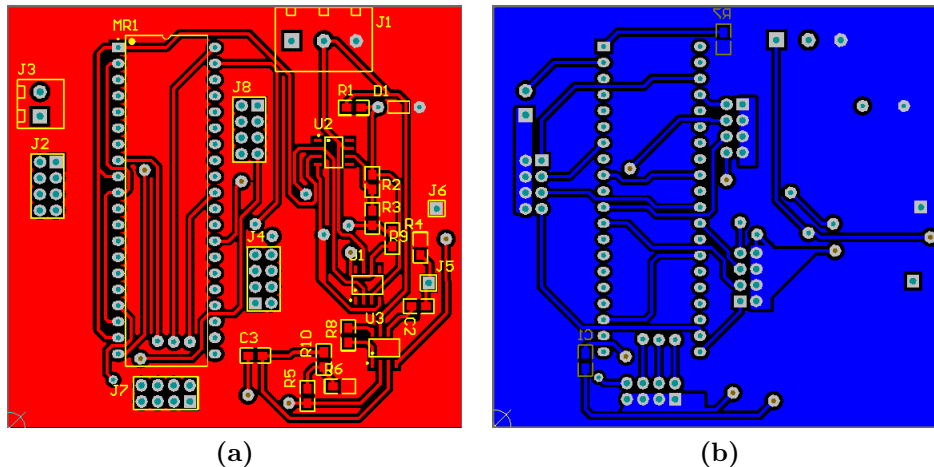


Figura 5.21 – Placa de circuito impreso para el conversor GIC: cara superior (a) y cara inferior (b).

Con estos resultados se comprueba el funcionamiento satisfactorio del modelo a frecuencias de hasta 10 kHz, frecuencia límite que nos permite nuestro sistema de medida, fijada por el amplificador de transconductancia.

5.3 Medida de la potencia. Wattímetro.

La necesidad de medir la potencia eléctrica ha motivado el desarrollo de varios tipos de medidores de potencia. Los básicos y más clásicos son medidores electromecánicos de aguja que todavía están en uso aún cuando presentan inconvenientes en la respuesta dinámica por el uso de bobinas de inductancia alta. Actualmente el ancho de banda del medidor de potencia es un parámetro de gran importancia debido a las necesidades de medida de señales no armónicas con alto contenido de armónicos. Existen dos tipos de estrategias para la medida en estas circunstancias: (a) la media separada de la tensión $v(t)$ y la corriente $i(t)$ en tiempo real y la conversión A/D para un posterior procesamiento digital en el que se multiplicarán las señales y (b) la multiplicación analógica de dos señales proporcionales a la tensión y corriente.

Siguiendo la estrategia analógica, la salida del transductor es la potencia instantánea de la señal definida por la ecuación 5.3.1 [RCBR09]:

$$P(t) = i(t) \cdot v(t) \quad (5.3.1)$$

en el caso de señales armónicas la potencia se define V_m e I_m que corresponden con la amplitud de la tensión y la corriente respectivamente y P , la potencia activa en un periodo:

$$v(t) = V_m \cdot \sin(\omega t + \phi) \quad (5.3.2a)$$

$$i(t) = I_m \cdot \sin(\omega t) \quad (5.3.2b)$$

$$P = \frac{1}{T} \int_0^T v(t) \cdot i(t) = VI \cdot \cos(\phi) \quad (5.3.2c)$$

donde $V = V_m/\sqrt{2}$ e $I = I_m/\sqrt{2}$ representan los valores RMS de la tensión y la corriente, figura 5.22.

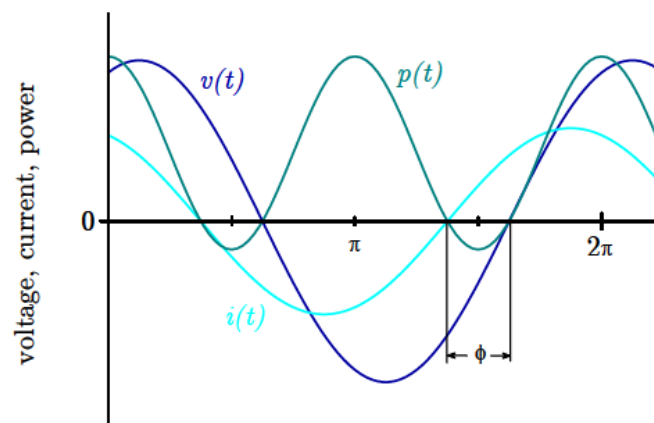


Figura 5.22 – Señales eléctricas de tensión, corriente y potencia [RCBR09].

Los transductores de potencia basados en sensores de efecto Hall como elemento multiplicativo han sido usados como elementos de medida. El problema básico que presentan es la limitada sensibilidad que obliga al uso de núcleos ferromagnéticos para concentrar el flujo

magnético en la zona del sensor. Dicho esto, los sensores magnetorresistivos surgen como potenciales sustitutos de los sensores Hall en esta aplicación. La idea básica es el empleo de los elementos magnetorresistivos como un multiplicador analógico muy simple usando la configuración del puente de Wheatstone.

La configuración se basa en un puente magnetorresistivo alimentado a una tensión proporcional a la tensión objeto de medida (V_{in}). Al mismo tiempo, se mide una corriente proporcional a la corriente de la señal que genera un campo magnético que hace variar la magnetorresistancia de los elementos del puente, ver figura 5.23. La tensión de salida ($+V_{out}$ y $-V_{out}$) del puente es linealmente proporcional al campo magnético generado y así mismo, a la tensión de salida. Como consecuencia directa de estos dos efectos, la tensión de salida es dependiente del producto de las dos señales. Esta idea ha sido aplicada usando un sensor de corriente comercial AMR (KMZ51) [VPK05]. Una configuración similar ha sido implementada con éxito para sensores de efecto Hall para la medida de la energía eléctrica [BKGDV06].

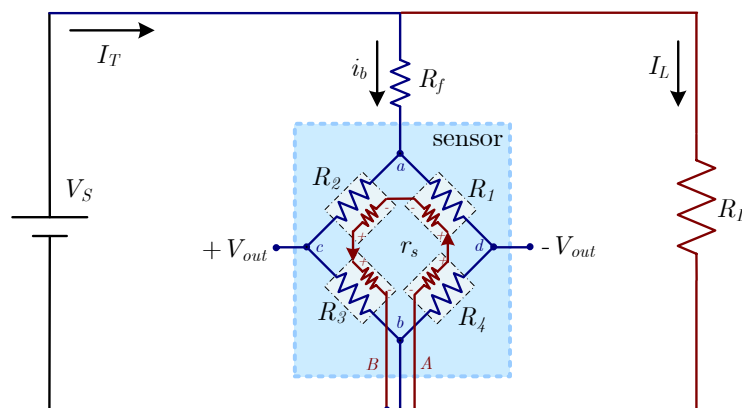


Figura 5.23 – Principio de medida de la potencia en circuitos integrados [RCBR09].

La sustitución del sensor AMR por uno GMR ha sido objeto de este apartado, donde se describe el uso de sensores magnetorresistivos GMR en una aplicación de medida de consumo de potencia a nivel de circuito integrado haciendo uso de las propiedades multiplicativas del los puentes de Wheatstone [RRCB+10].

5.3.1 Descripción del sensor de corriente.

Los dispositivos SV se depositaron (siguiendo el procedimiento del apartado 4.2.1.1.1) en formato rectangular ($200 \mu\text{m} \times 3 \mu\text{m}$) para formar el puente de Wheatstone, figura 5.24(b). El detalle del proceso de fabricación se puede encontrar en [CBRR+09].

En la figura 5.25 se muestran dos fotografías al microscopio correspondientes a la vista superior de los sensores de corriente fabricados. Con objeto de obtener una comparativa hemos comprobado dos configuraciones diferentes: la serie (SV07-SN, figura 5.25(a)) y la paralela (SV07-PW, figura 5.25(b)). En ambas configuraciones la corriente I_{SENSOR} fluye por las líneas de color claro sobre las resistencias R_1 y R_3 y en dirección opuesta sobre las R_2 y R_4 . Consecuentemente cuando se introduce una corriente entre los terminales A y B, dependiendo del signo R_1 y R_3 incrementa o decremantan su valor mientras que R_2 y R_4 lo

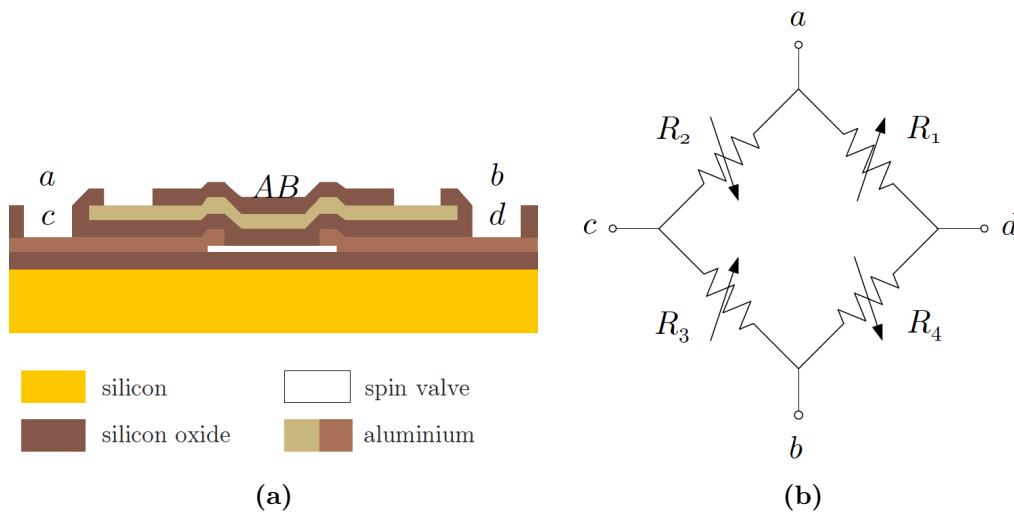


Figura 5.24 – (a) Sección vertical de los dispositivos *SV* utilizados en este trabajo. (b) Estructura del puente de Wheatstone resultante [RRS04b].

hacen de manera opuesta. Se obtiene así un comportamiento de puente de cuatro elementos sensibles, figura 5.24(b). Una configuración similar a la del sensor SV07-SN fue realizada y usada con éxito en una aplicación mixta IC/PCB para un sensor de corriente [RRS+04a]. Es necesario destacar que en este caso los cuatro dispositivos magnetorresistivos han sido depositados y patroneados a la vez con idéntico *easy axis* para minimizar la variabilidad asociada a las diferencias físicas y eléctricas entre ellos. Este procedimiento ha mejorado las características en cuanto a deriva térmica e inmunidad a campos magnéticos externos. El puente se polariza a través de los terminales *a* y *b*, mientras que la tensión de salida diferencial se mide en los terminales *c* y *d*.

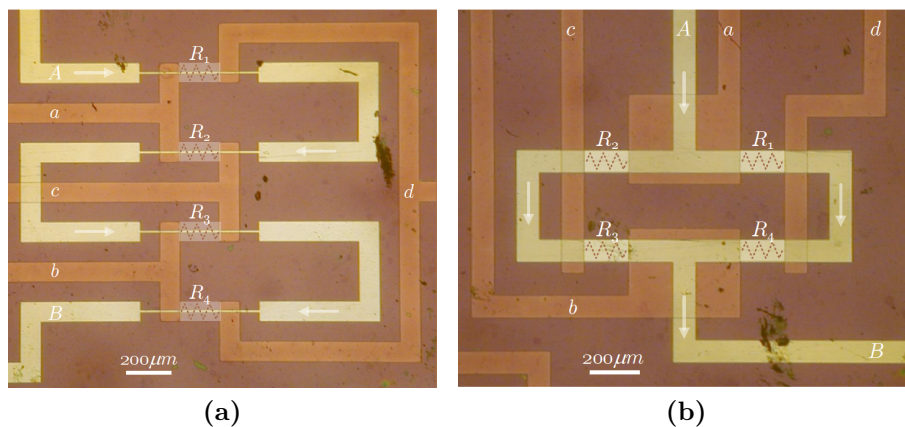


Figura 5.25 – Vista superior al microscopio del sensor de corriente (a) SV07-SN y (b) SV07-PW [RRCB+10].

5.3.2 Descripción del modelo

Desde el punto de vista eléctrico cada elemento magnetorresistivo ha sido modelado mediante un cuadripolo, figura 5.26, de manera similar a la introducida en el apartado 4.2.1.1 del tema anterior. Dadas las características de las SV fabricadas, dentro del intervalo de

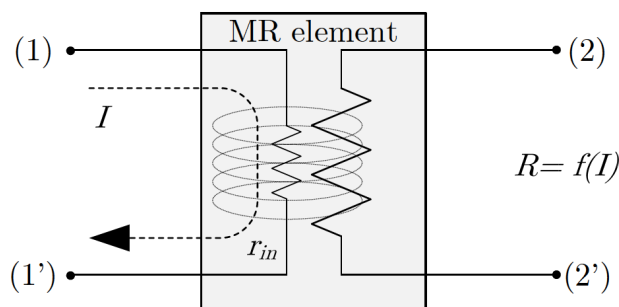


Figura 5.26 – Estructura circuital del cuadripolo equivalente de los elementos SV [RRCB⁺ 10].

comportamiento lineal, la función ($f(I)$) que relaciona la corriente que circula por el sensor (I_{SENSOR}) con la magnetorresistencia (R) puede representarse con bastante precisión por la expresión polinómica:

$$R = R_0 + \sum_{i=1}^n MR_i \cdot I_{SENSOR}^i \quad (5.3.3)$$

donde I_{SENSOR} es la corriente de control del sensor aplicada entre los terminales (1) y (1'), R es la resistencia de salida entre los terminales (2) y (2'), R_0 es la resistencia de reposo para cuando $I_{SENSOR} = 0$ y MR_i son los coeficientes del polinomio. Esta forma general puede tener en cuenta efectos moderados de segundo orden como histéresis y derivas térmicas. Las no linealidades presentes en los dispositivos simples quedan atenuadas cuando se configura el puente de Wheatstone gracias a sus propiedades linealizadoras.

5.3.2.1 Procedimiento de extracción de parámetros

Los parámetros del modelo descrito en la ecuación (5.3.3) se obtienen mediante ajuste de las curvas experimentales en línea con lo expuesto ya en el capítulo anterior. El sistema de instrumentación es similar al descrito anteriormente; es decir, está basado en GPIB y permite obtener en un PC las medidas realizadas con la fuente de alimentación (PS2521G, Tektronix), fuente de corriente (220, Keithley), unidad de adquisición (34970A, Agilent) y un multímetro (34401A, Agilent). Las resistencias independientes (R_1 - R_4) son medidas para cada dispositivo en función de la corriente de control I_{SENSOR} dentro del intervalo de operación. Para obtener los valores se miden las resistencias del puente entre nodos consecutivos ($a-d$, $d-b$, $b-c$, $c-a$, ver figura 5.24(b)) y posteriormente las resistencias R_n eran obtenidas de consideraciones en las asociaciones serie-paralelo. En la figura 5.27 se muestran los valores normalizados teniendo en cuenta que el valor típico de R_n es (1.1 ± 0.1) k Ω . Como cabía esperar, ambos sensores de corriente presentan un comportamiento de puente completo con las resistencias actuando en pares R_1/R_3 y R_2/R_4 . Las resistencias son lineales y dependientes del signo de la corriente de control. Para este caso se puede utilizar una versión linealizada de la

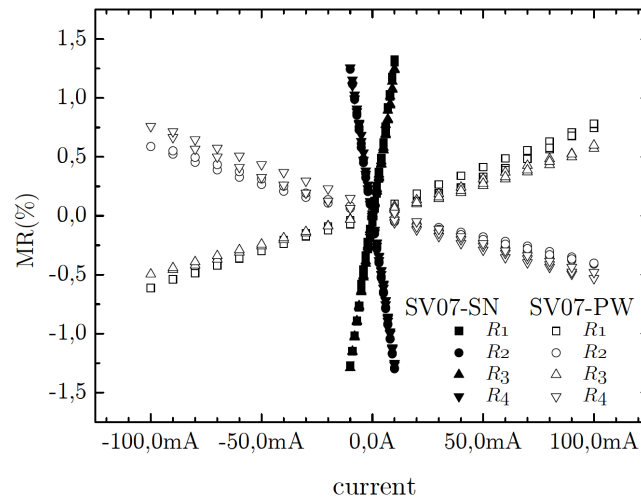


Figura 5.27 – Valores experimentales de las resistencias del puente en función de I_{SENSOR} [RRCB⁺10].

ecuación (5.3.3) en la que $MR_i = 0, \forall i > 1$. Es necesario mencionar que las resistencias medidas son independientes de la corriente de polarización. Los valores se presentan en la tabla 5.2. Como se puede observar, existe una considerable tolerancia en los valores de R_0 , lo que origina una tensión de *offset* a la salida del puente. Además MR_1 es aproximadamente diez veces mayor para los dispositivos serie SV07-SN. Esto es debido a la anchura de la pista de corriente, que es diez veces menor generando una mayor densidad de corriente y un mayor campo magnético asociado.

Tabla 5.2 – Parámetros del modelo de resistencias del puente [RRCB⁺10].

	SV07-SN		SV07-PW	
	R_0 (Ω)	MR_1 (Ω/A)	R_0 (Ω)	MR_1 (Ω/A)
R_1	1131.5	1441.1	1118.6	122.6
R_2	1096.0	1384.1	1153.7	118.1
R_3	1062.5	1342.4	1154.2	124.9
R_4	1128.0	1410.1	1130.8	124.7

5.3.2.2 Verificación del modelo.

La verificación del modelo analítico se hizo mediante el simulador de circuitos. Para ello, se introdujo una corriente DC de polarización entre los terminales (a) a (b) de 0.2, 1.0 y 5.0 mA y la tensión de salida entre los terminales (c) y (d) fue medida para una corriente de control entre los terminales (A) y (B), ver figura 5.25. En la figura 5.28 se presenta la comparativa entre los datos experimentales y los simulados en ADS[®]. Se observa un comportamiento altamente lineal en el intervalo de operación del sensor de corriente. El ajuste entre los datos medidos y simulados es muy bueno. No se han observado efectos de autocalentamiento e histéresis en el proceso de modelado. Además los dispositivos presentan una respuesta lineal con la tensión de polarización. Esta última observación es de vital

importancia para el uso posterior del puente como elemento multiplicativo.

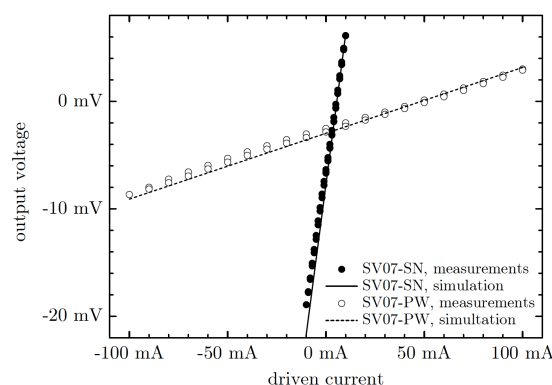


Figura 5.28 – Tensión de salida del puente en función de la corriente I_{SENSOR} . Datos experimentales (símbolos) y simulación (líneas) para los dispositivos SV07-SN y SV07-PW [RRCB⁺10].

5.3.3 Aplicación para la medida de la potencia eléctrica.

La medida de la potencia eléctrica DC es una de las aplicaciones emergentes de los sensores de corriente GMR en el contexto de los circuitos integrados (IC).

5.3.3.1 Resultados

Hemos realizado barridos para diferentes valores de R_L , tanto experimentalmente como en simulación para un conjunto de valores de tensión V_S . La potencia entregada a R_L (P_L) ha sido variada para estudiar el comportamiento de la tensión de salida del puente. Los valores resultantes se muestran en la figura 5.29 para tres valores diferentes de V_S (3.3, 5 y 9 V DC). Las simulaciones se realizaron con el simulador de circuitos ADS[®] y ELDO[®] usando el modelo compacto de magnetorresistencia desarrollada en Verilog-A. Para las medidas experimentales se ha utilizado el sistema de instrumentación basado en GPIB descrito anteriormente.

Las simulaciones se ajustan con precisión a las medidas experimentales. Las desviaciones para R_L bajas (altas I_L) se deben a los efectos de calentamiento. Las corrientes a las que estos efectos aparecen se muestran en línea discontinua en la gráfica. Debido a la menor anchura de pistas de corriente los efectos de autocalentamiento son más apreciables en los dispositivos serie. Combinando ambos tipos de sensores se puede cubrir un amplio rango de valores de R_L para una tensión de entrada moderada. Para mayores valores de R_L , P_L disminuye y también V_{out} . Un valor ligeramente mayor de tensión de *offset* ha sido medido en esta región. Este *offset* es más importante para altas tensiones de entrada y es reproducido perfectamente por el modelo.

Finalmente con el objeto de demostrar la utilidad del circuito para la medida de la potencia DC a nivel de IC, se presentan en la figura 5.30 la tensión de salida con el *offset* corregido en función de la potencia entregada a la carga (R_L).

Nuevamente las simulaciones se ajustan con precisión a los datos experimentales. De-

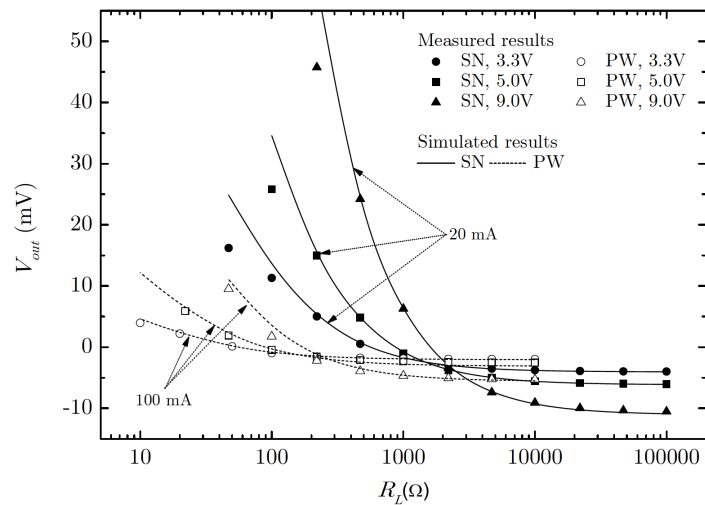


Figura 5.29 – Datos experimentales y simulados de la tensión de salida en función de R_L para diferentes valores de V_S [RRCB⁺10].

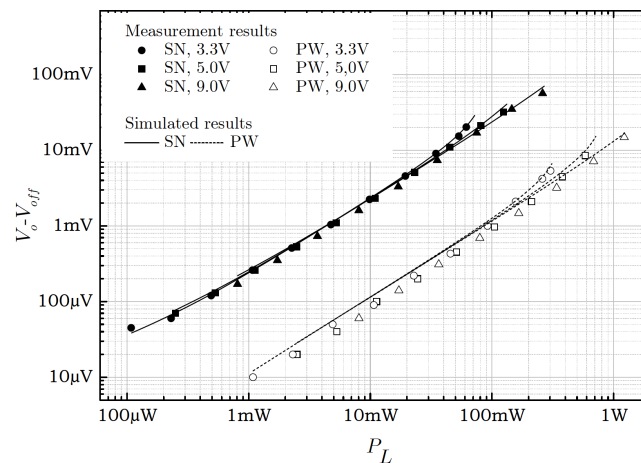


Figura 5.30 – Datos simulados (líneas) y experimentales (símbolos) de la tensión de salida en función de la potencia consumida en R_L [RRCB⁺10].

bido a su alta sensibilidad, los sensores serie generan una tensión de salida mayor. Un rango de medida de 100 μ W hasta 1 W puede ser abarcado mediante la combinación de ambos dispositivos serie y paralelo. El rango la salida de los dispositivos es muy lineal. Las desviaciones de linealidad que se producen a altas potencias se deben al efecto de r_s , ver figura 5.23. Las corrientes de control del sensor han de mantenerse dentro de los márgenes en los que los efectos del autocalentamiento son despreciables. Las pequeñas desviaciones a bajas potencias se deben a los efectos de la tensión de *offset* que perturban la linealidad.

CAPÍTULO

6

PUBLICACIONES

Este trabajo ha sido financiado por los siguientes proyectos:

- **Desarrollo de modelos para SPICE y simulación de circuitos fabricados con dispositivos SOI submicra** (TEC2004-03926).
- **Desarrollo de modelos para SPICE y simulación de circuitos y dispositivos basados en tecnologías emergentes (silicio tenso, SIGE, SOI, GEOI)** (TEC2005-01948/MIC).
- **Modelado compacto para la caracterización termoeléctrica de dispositivos electrónicos nanométricos de última generación orientado al diseño de circuitos de baja potencia y RF** (P08-TIC-3580).
- **European platform for low-power applications on Silicon-on-Insulator Technology EUROSOI+.**(FP7-ICT-216373).
- **Desarrollo de aplicaciones de medida industriales basadas en la tecnología de sensado magnetorresistiva** (ENE2005-08721-C04-03, Ministerio de Educación y Ciencia)
- **Diseño de matrices de sensores GMR para el mapeo de nano-partículas magnéticas** (AE-UVEG-2011, Universitat de València)

Publicaciones y Contribuciones Derivadas

- A. Roldán, C. Reig, J.B. Roldán, A. Cano-Abellán, S. Cardoso y P.P. Freitas, *A Current-Time conversor circuit based on giant magnetoresistance sensors. A Verilog-A implementation*, MOS-AK 2011, Paris, Francia, 2011 [RRR+11a].
- A. Roldán, C. Reig, J.B. Roldán, M.D. Cubells-Beltrán, S. Cardoso y P.P. Freitas, *Simulation study of IC power measurement applications by means of Verilog-A models for GMR current sensors*, MOS-AK 2011, Paris, Francia, 2011 [RRR+11c].
- A. Roldán, C. Reig, J.B. Roldán, A. Cano-Abellán, S. Cardoso y P.P. Freitas, *Quasi-static magnetoresistive sensor modeling for current-time conversion circuit applications*, 8th Spanish Conference on electron Devices, Palma de Mallorca, España, 2011 [RRR+11b].
- A. Roldán, J.B. Roldán, C. Reig, M.D. Cubells-Beltrán, D. Ramírez, S. Cardoso and P.P. Freitas, *A DC behavioral electrical model for quasi-linear spin-valve devices including thermal effects for circuit simulation*, Microelectronic Journal, Vol. 42, N. 2, pags. 365-370, 2011, [RRR+11e].
- A. Roldán, C. Reig, M.D. Cubells-Beltrán, J.B. Roldán, D. Ramirez, S. Cardoso y P.P. Freitas, *Analytical compact modeling of GMR based current sensors: application to power measurement at the IC level*, Solid State Electronic, Vol. 54, páginas 1606-1612, 2010, [RRCB+10]
- A.M. Roldán, J.B. Roldán, F. Gámiz and F. Jiménez-Molinos, *Advanced modeling of Cylindrical Surrounding Gate Transistors for circuit simulation*, XXV Conference on Design of Circuits and Integrated Systems DCIS, Lanzarote, Spain, November 17-19, 2010 [RRGJM10].
- A. Roldán, C. Reig, M.D. Cubells-Beltrán, J.B. Roldán, D. Ramírez, S. Cardoso, P.P. Freitas, *A Verilog-A implementation of a model for spin-valve based current sensors* Aceptado en la sesión de Posters de MOS-AK 2010, Sevilla, España [RRR+11d].
- A. Roldán, J.B. Roldán, *Análisis simbólico de circuitos mediante técnicas de análisis modal modificado. Tecnologías aplicadas a la enseñanza de la electrónica, IX Congreso TAAE 2010* realizado en Madrid, 2010 [RR10].
- A. Roldán, C. Reig, J.B. Roldán, A. Cano-Abellán, D. Ramírez, *Quasi-Static Electrical Model for Magnetoresistive Current Sensors. International Conference on Renewable Energies and power quality* celebrado en Granada, 2010 [RRR+10].
- C. Reig, M.D. Cubells, D. Ramírez, A. Roldán, J.B. Roldán, S. Cardoso, P.P. Freitas, *GMR Current Sensors for IC applications., IBERNAM 2009*, Sevilla, España [RCR+09].
- A. Roldán, J.B. Roldán, F. Gámiz, *Simulation study of digital circuits based on nanometric Surrounding Gate Transistors: the role of quantum and velocity overshoot effects*, MOS-AK 2009/ESSDERC/ESSCIRC, Atenas, Grecia [RRG09].
- A. Roldán, J.B. Roldán, F. Gámiz, *Simulation of CMOS inverters based on the novel surrounding gate transistor. A Verilog-A implementation*, MOS-AK 2008//ESSDERC/ESSCIRC, Edinburgo, Gran Bretaña [RRG08b].
- A. Roldán, J.B. Roldán, F. Gámiz, *A in-depth simulation study of CMOS inverters ba-*

- sed on the novel Surrounding Gate Transistors, International Conference on Advances in Electronics and Micro-electronics ENICS-2008, Valencia, España [RRG08a].*
- J.B. Roldán, F. Gámiz, A. Roldán, N. Rodríguez, *Characterization of electron transport at high fields in silicon-on-insulator devices: a Monte Carlo study*, Semiconductor Science and Technology, Vol. 21, 2006 [RGRR06].
 - A. Roldán, J.B. Roldán, F. Gámiz, *Characterization and modelling of SOI MOSFET velocity overshoot: enhancement of SPICE BSIMSOI model*, Eurosoi, Granada, 2005 [RRG05a].



CAPÍTULO

7

APÉNDICES

7.1 Fabricación de dispositivos MTJ

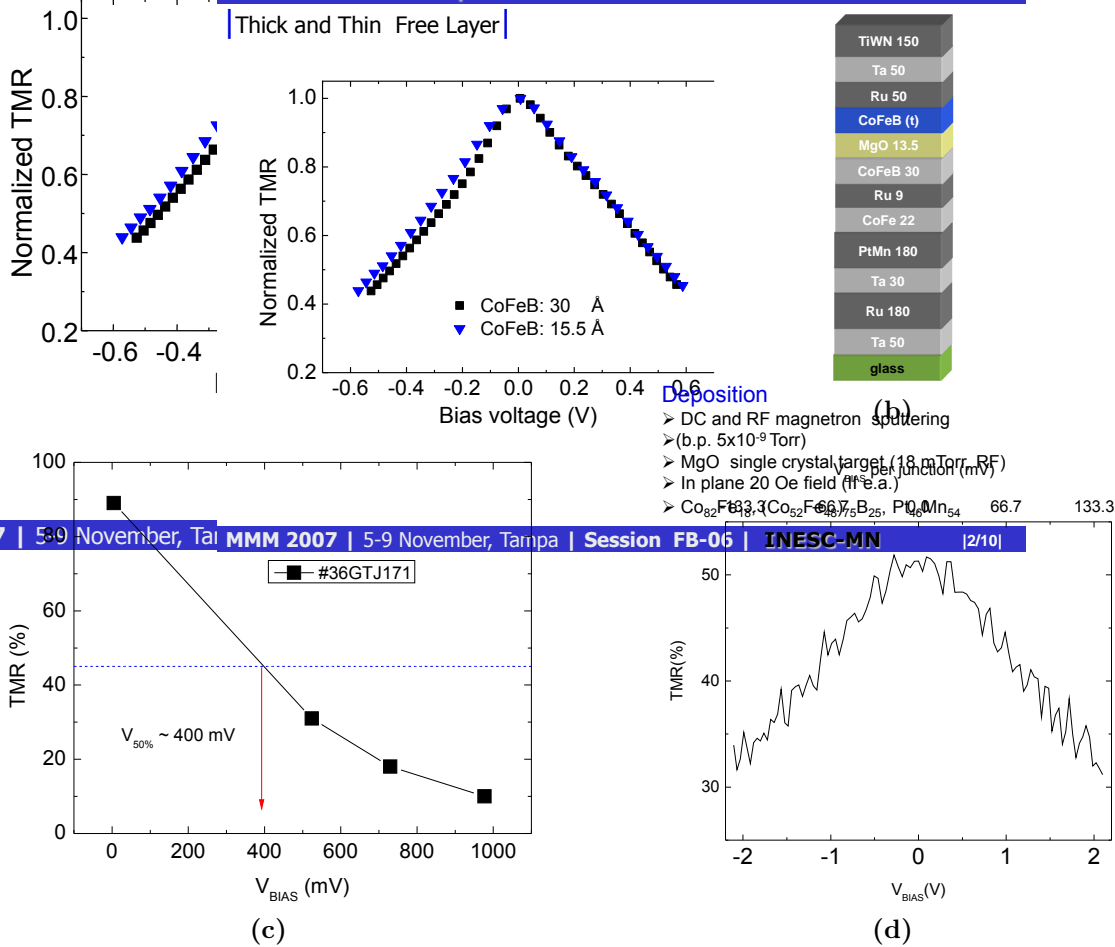
En este apartado se detalla el proceso de fabricación que se ha seguido para la obtención de las magnetorresistencias **MTJ** que se describen en este trabajo. Para la fabricación de estos sensores se han utilizado tres pasos de litografía estándar para definir: el electrodo inferior (*bottom electrode*), la zona del dispositivo (*junction area*) y los contactos metálicos sobre el electrodo superior (*top electrode*). Posteriormente se depositan mediante un último paso de litografía las pistas de corriente que crearán el campo magnético a medir.

En los siguientes subapartados se describe el proceso de fabricación de una magnetorresistencia de cuatro contactos similar a la usada para test y verificación del proceso de fabricación. Unas estructuras similares se han introducido en la zona central de la muestra de $8,2 \times 8,2 \text{ mm}^2$.

La novedad de los sensores de corriente **MTJ** consiste en la configuración en serie de múltiples dispositivos **MTJ** de manera que puedan ser polarizados con una tensión de decenas de voltios (como es usual en las aplicaciones circuitales de instrumentación) sin superar la tensión (V_{break}) de los elementos. Una de las limitaciones inherentes de las **MTJ** es que el ratio **TMR** depende de la tensión de polarización de la resistencia. En la figura 7.1(a) se presenta la variación normalizada del **TMR** con respecto a la tensión de polarización de los electrodos del dispositivo **MTJ** de dimensiones $2 \times 30 \text{ }\mu\text{m}^2$ cuya estructura de láminas se muestra en la figura 7.1(b). Para obtener un $\text{TMR} \approx 50 \%$ es necesario establecer una tensión de polarización máxima por dispositivo **MTJ** de $V_{1/2}=400 \text{ mV}$ [WACF07]. En la figura 7.1(d) se muestra el ratio **TMR** para el caso de conexión serie 15 dispositivos magnetorresistivos.

Thick and Thin Free Layer

TMR Bias Dependence



MMM 2007 | 5-9 November, Taipei | MMM 2007 | 5-9 November, Tampa | Session FB-06 | INESC-MN | [2/10]

Figura 7.1 – Representación de la variación del ratio TMR de una muestra de capa delgada (15.5 Å) y gruesa (30 Å) en función de la tensión de polarización (a). Estructura laminar de la MTJ (b). Variación del ratio TMR en función de V_{BIAS} para un dispositivo de tamaño $2 \times 30 \mu\text{m}^2$ (c) idem. para un conjunto de 15 dispositivos MTJ conectados en serie [CMF+ 07].

Para la fabricación de este dispositivo se ha partido del diseño presentado en [RCBR08] y revisado por los mismos autores. En nuestro diseño se ha utilizado un dispositivo MTJ básico de tamaño $2 \times 6 \mu\text{m}^2$. Las magnetorresistencias fabricadas en las esquinas exteriores del chip son asociaciones en serie de la resistencia unitaria; en la tabla 7.1 se presentan las diferentes configuraciones fabricadas detallando el número de resistencias unitarias, la impedancia equivalente y el tamaño real.



				Estructura Serie
Nº Dispos. en serie	MR (KΩ)	Anchura (μm)		
1	0.666	6		
R1	16·6 = 96	64	40	
R2	16·5 = 80	54	34	
R3	16·2 = 32	21	14	
R4	20·1 = 20	13	8	

Tabla 7.1 – Características de las cuatros estructuras de dispositivos *MTJ* individuales. Se detalla el número de resistencias unitarias en serie, la impedancia equivalente (KΩ), la anchura (μm) y la definición de máscaras utilizadas. *Figure II.20* Picture of the SVG in-line track. Up to 25 wafers (Si, Ø6 inch) in a cassette can be baked, coated, cooled down and spined automatically.

7.1.1 Deposición de las capas que forman la magnetorresistencia MTJ.

Se monta la muestra de *Si/SiO₂* *Coating and developing* sobre¹ de 6" ha *Exposure* de cintas adhesivas, figura 7.2. Se deposita sobre toda la muestra el conjunto de capas de

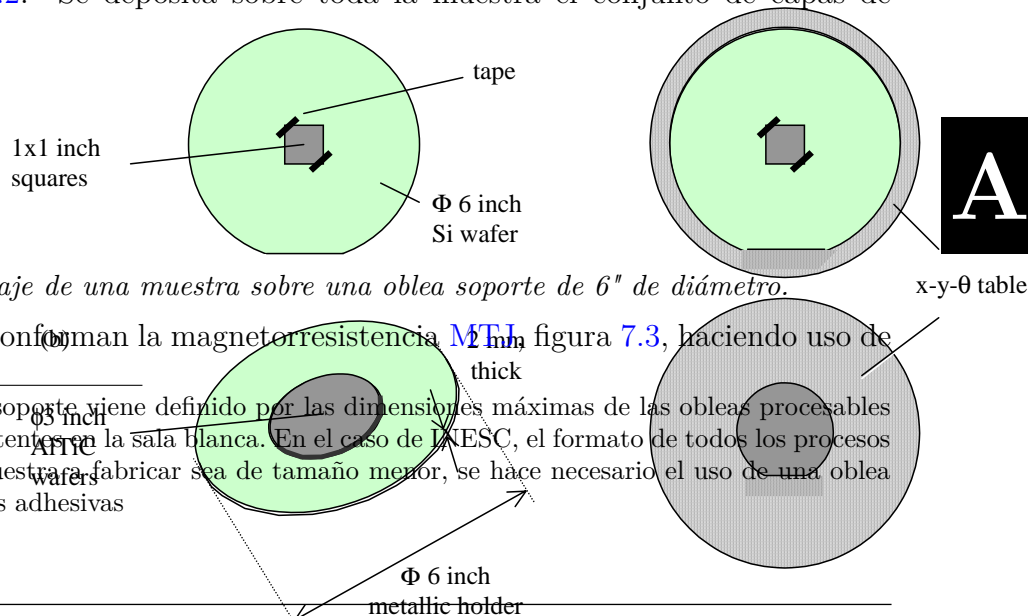


Figura 7.2 – Montaje de una muestra sobre una oblea soporte de 6" de diámetro.

diferentes materiales que conforman la magnetorresistencia *MTJ*, figura 7.3, haciendo uso de

¹El tamaño de la oblea de soporte viene definido por las dimensiones máximas de las obleas procesables en las diferentes máquinas existentes en la sala blanca. En el caso de INESC, el formato de todos los procesos es de 6". En caso de que la muestra a fabricar sea de tamaño menor, se hace necesario el uso de una oblea soporte y el montaje con cintas adhesivas

Figure II.21 – Sample mounting for the track and for lithography: a) Smaller than 6 inch samples are mounted onto a Si wafer both for coating and developing and for the exposure, b) Ø3 inch AlTiC wafers (< 2.2 mm thick) are mounted on a metallic holder and loaded manually into the tracks, and are directly placed over the x-y-θ table for the exposure.

Nanofabrication Process Example

Sub 50nm MTJ process : CMP based via opening

INESC-MN

resultado una estructura

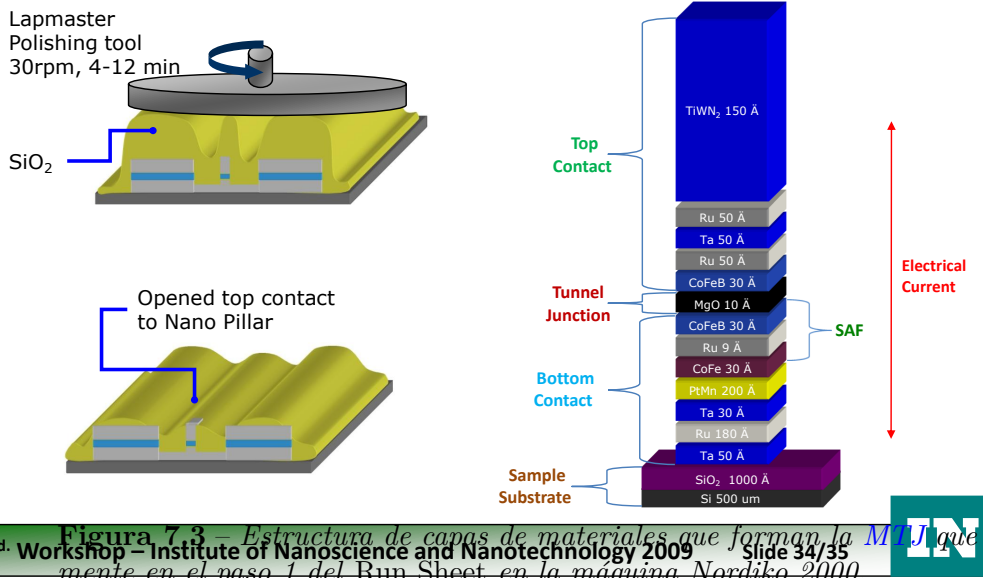


Figura 7.3 – Estructura de capas de materiales que forman la MTJ que son depositados sucesivamente en el paso 1 del Run Sheet en la máquina Nordiko 2000. Slide 34/35

7.1.2 Definición del Electrodo Inferior.

Antes de iniciar la primera fase del proceso litográfico es necesario realizar un horneado de la muestra en una atmósfera de vacío y nitrógeno para mejorar, mediante la nebulización de un spray, la adherencia del polímero fotosensible (*photoresist*) sobre la superficie de la muestra [CdF01]. En la tabla 7.2 se detallan los pasos y tiempos de calentamiento usados en el proceso de horneado del sustrato realizado horno de Yield Engineering Systems™ existente. La muestra se fotosensibiliza siguiendo el proceso detallado en la tabla 7.3 en la

Paso	Objetivo	Procedimiento
1	Deshidratación de la muestra y eliminación de oxígeno de la cámara	Realización de Vacío a 10 Torr durante 2 minutos
		Entrada de N2 a 760 Torr durante 3 minutos (3 veces)
2	Nebulización de HDMS	Realización de Vacío a 1 Torr durante 3 minutos
		Hexamethyldisilazane (HDMS) a 6 Torr durante 5 minutos
3	Eliminación gases de la cámara	Realización de Vacío a 4 Torr durante 1 minuto
		Entrada de N2 a 500 Torr durante 2 minutos
		Realización de Vacío a 4 Torr durante 2 minutos

Tabla 7.2 – Configuración de pasos y tiempos en el proceso de horneado.

línea 2 (*TRK-2*) del sistema (SVG™) mediante una cobertura de *photoresist*².

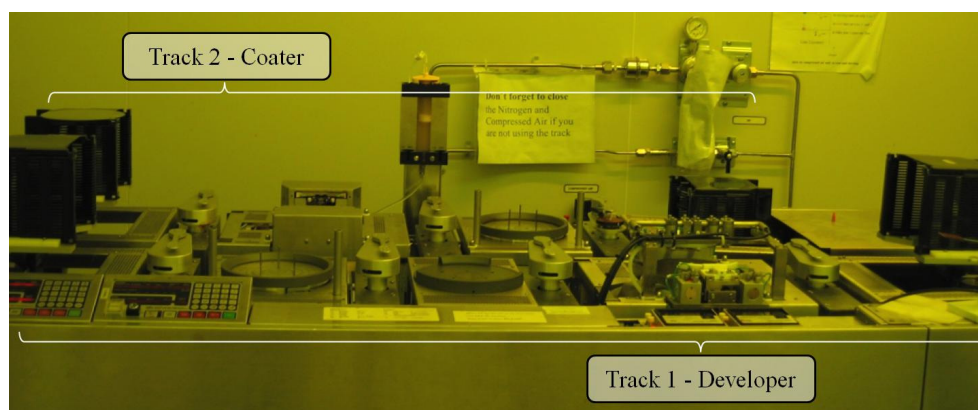


Figura 7.4 – Sistema de fotosensibilización (*Track 2*) y revelado (*Track 1*) de Silicon Valley Group™ para 25 obleas de 6" de diámetro.

La litografía láser se realiza con un equipo DWL 2.0, figura 7.5, fabricado por Heidelberg™ que usa un láser de 440 nm de Neón-Argón para definir los patrones en el *photoresist* (PFR7790G27cP, JSR Electronics) de 1,5 μm de grosor con una resolución de 0,8 μm y una precisión de alineamiento de 0,1 μm .

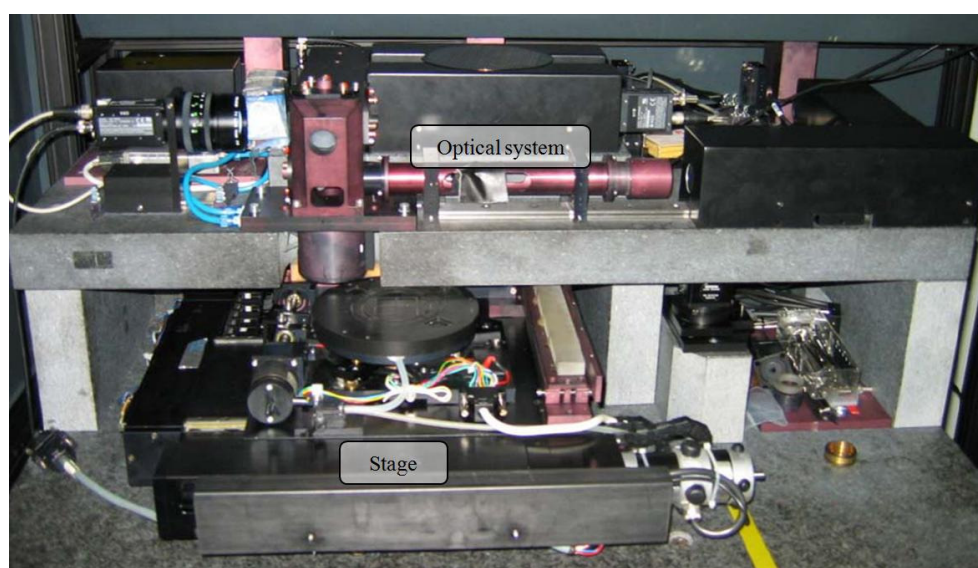


Figura 7.5 – Sistema de litografía láser DWL 2.0.

La máscara utilizada es la L1BOT invertida³, mediante la DWL 2.0. y se procede con el revelado (*develop*) de la muestra en la pista 1 (*Track 1*) de la máquina SVG™ mediante el líquido *Ethyl lactate* (Pth70eg) mediante el programa 6/2, ver tabla 7.3.

Finalmente la muestra es atacada por *ion milling* eliminando el material desprotegido, paso 4 del *Run Sheet*, pasando al baño de disolvente orgánico denominado *microstrip*

²Se utiliza una iluminación de color amarillo en la sala para evitar el velado involuntario de las muestras.

³Una máscara invertida es generada vía *software* con el propósito de obtener el negativo de la máscara original.

<u>Objetivo</u>	<u>Línea / Receta</u>	<u>Procedimiento</u>
Fotosensibilización de la muestra (1.5 μm photoresist)	Línea - TRK 2	Cocido a 110°C durante 60 s
		Enfriado durante 30 s
		Añadido del photoresist a 800 rpm durante 5 s
	Receta (6/2)	Giro a 2.8 Krpm durante 40 s y posteriormente a 1.6 Krpm durante 5 s
		Limpiado del photoresist de los bordes de la oblea a 1 Krpm durante 2 s
		Giro a 1.5 Krpm durante 15 s
		Cocido a 100°C durante 60 s
Revelado	Línea - TRK 1	Cocer at 110°C durante 60 s
		Enfriado durante 30 s
		Nebulizar agua y girar a 500 rpm durante 1s
	(Receta 6/2)	Añadido del revelador a 500 rpm durante 5 s.
		Actuación del revelador durante 60 s con la oblea en reposo.
		Giro con agua destilada a 1 Krpm durante 20 s
		Secado con la oblea girando a 3.5 Krpm durante 30 s.

Tabla 7.3 – Configuración de pasos y tiempos en el proceso de fotosensibilizado y revelado.

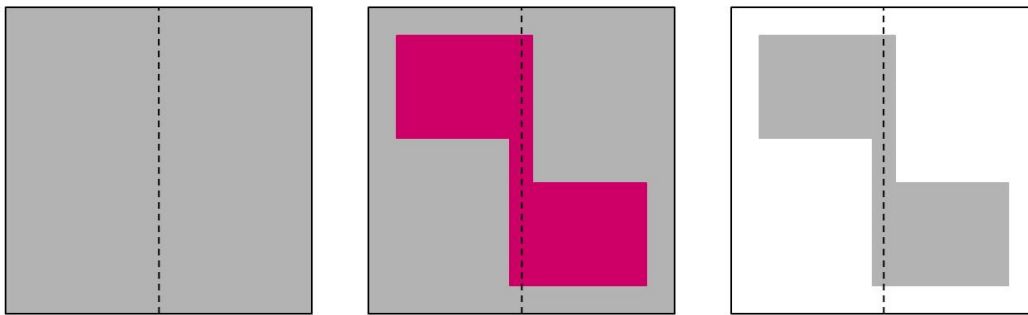
calentado a unos 65°, [paso 5 del Run Sheet](#), figura 7.6(c). Este último paso puede incluir un baño en ultrasonidos para facilitar el desprendimiento de todos los residuos de *photoresist* revelado disminuyendo la duración procesado.

7.1.3 Definición del área del dispositivo MTJ.

Se deposita *photoresist* en la línea 2 (*TRK-2*) del sistema (SVG™) siguiendo el proceso detallado en la tabla 7.3. Se aplica la máscara L2TOP invertida mediante el láser DWL 2.0, sobre los *pads* de $420 \times 420 \mu\text{m}^2$ y el área del dispositivo y se revela en la pista 1 del sistema (SVG™), [paso 6 del Run Sheet](#). El material no protegido por el *PR* es eliminado por atacado mediante *ion beam milling*, [paso 7 del Run Sheet](#). Para detener el proceso de atacado una vez superada la capa aislante de MgO se utiliza una muestra de calibración sobre sustrato de vidrio que tiene depositada la misma estructura de capas. Una vez que ha transcurrido el tiempo estimado de eliminación de materiales por atacado, se realiza un control visual para

solvent called *microstrip*. The latter wet etching process is called resist strip; it can also include ultrasounds in order to facilitate the removal of all residues. The end of this first step is determined by optical inspection with a microscope.

Top View



Cut View

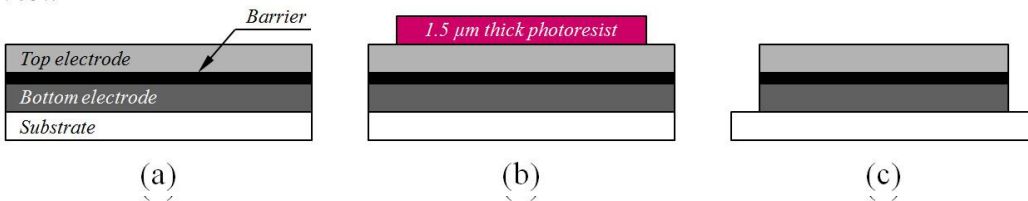
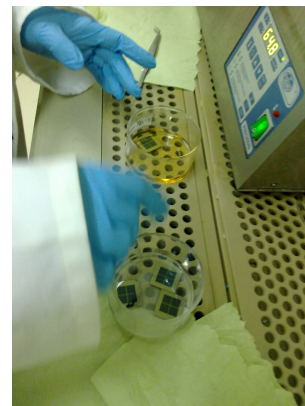


Figure 3-17: Schematic top and cut view of the bottom electrode definition. The bulk sample deposited in a substrate (a) is patterned defining the final bottom electrode area (b); the sample is etched by ion beam milling and the photoresist leftovers are removed by wet etch (c).



(a) *Disolvente μstrip limpio.*



(b) *Introducción muestras en disolvente.*



(c) *Actuación en ultrasónicos.*



(d) *Eliminación de restos líquidos mediante aire comprimido.*

Figura 7.7 – *Diferentes etapas en el proceso de lift-off.*

detectar si la muestra de vidrio está completamente limpia; en caso afirmativo el proceso de



avoid material redeposition in the junction sidewalls, which can have serious repercussions in the electrical properties of the device.
the electrical properties of the device.

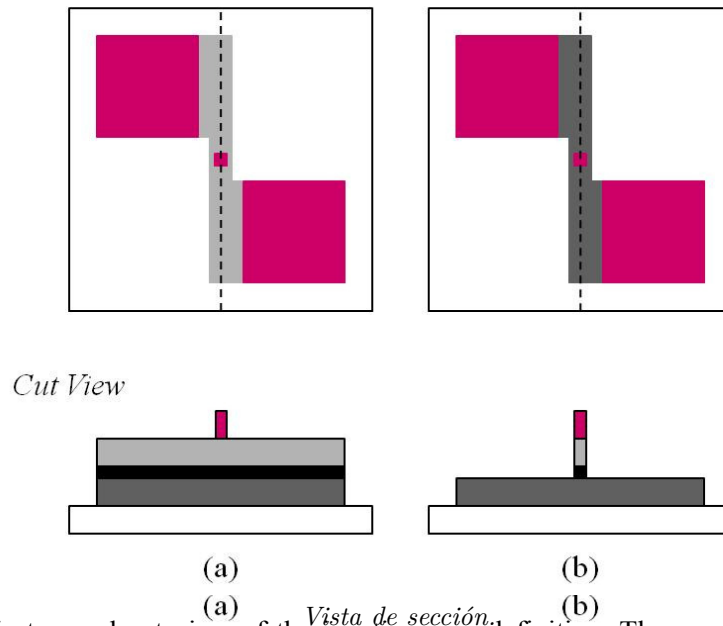
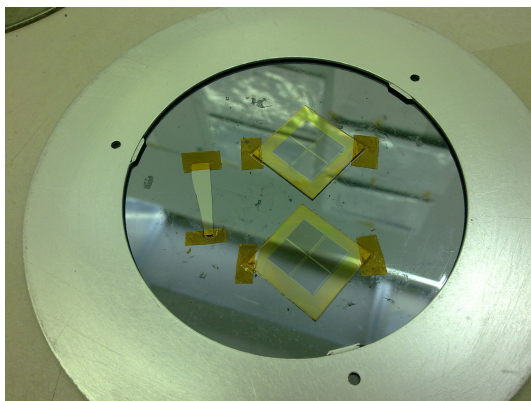
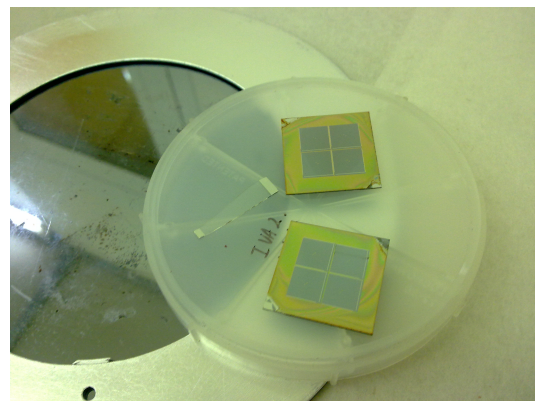


Figure 3-18: Schematic top and cut view of the junction area definition. The sample is patterned defining the junction area and the contact pads (a), and then it is etched by ion beam milling until the bottom electrode is reached (b).
 Figura 3-18: Vistas de la definición del dispositivo MTT. Sobre la pila de capas que forman la magnetorresistencia MTT (a) se deposita el fotoresist para posterior litografía con máscara y se define el área de la unión y los pads (a), y luego se etcha por ion beam milling hasta que el electrodo inferior es alcanzado (b).
 atacado habrá concluido y en caso negativo será necesario prolongar el tiempo de atacado hasta dejar el vidrio completamente limpio, figura 7.9(b).



(a) a) Muestras antes de iniciar el 2º atacado.



(b) b) Muestra de calibración limpia.

Figura 7.9 – Oblea soporte de 3" para introducción en la máquina Nordiko 3000.

El segundo atacado es el paso más crítico del proceso de fabricación. Si el atacado se detiene antes de que la profundidad obtenida llegue a la barrera, figura 7.10 el dispositivo estará en cortocircuito y el coeficiente magnetorresistivo TMR será nulo. Del mismo modo, si el atacado finaliza antes de llegar a las capas metálicas existentes bajo la barrera aislante no habrá contacto eléctrico con el electrodo inferior y también el TMR será nulo. Es muy importante tener en cuenta que se pueden producir deposiciones de material en las paredes laterales de la barrera que tendrían serias repercusiones en el comportamiento eléctrico del dispositivo al limitar el efecto túnel deseado. Para evitar esta disfunción no deseada se realiza una variación de la inclinación del haz iónico durante la fase final de aislado que pasa de

3.3.2.3. Insulating layer deposition

At this point the junction pillar is already defined and, without removing the PR leftovers, a thick layer of insulator is deposited (typically 40 nm of Al₂O₃) with the sputtering system described in section 3.1.1.4. This insulating layer prevents the electrical current from flowing between bottom electrodes through the barrier, but the junction area. After the deposition, the remaining PR from the previous step works as a lift-off mask for the insulator (self-aligned process), as illustrated in Figure 3-19 (b).

Figura 7.10 – Detalle del proceso de ataque conforme va aumentando el tiempo de procesamiento y consecuentemente la profundidad de material erosionada.

Assist Gun: 160W 105mA +735V/350V ; 12sccm Ar ; Assist Neut: 30% subst.rot. 60° subst.pan.
unos 70 durante los 140 segundos iniciales hasta 30 durante los 60 segundos finales..

The insulator lift-off duration increases with the junction area decrease and increasing insulator thickness. For junctions with few μm² and 40 nm thick Al₂O₃, this lift-off can take several days (~ three days), mostly due to a non-uniform PR profile. In order to increase the lift-off success and decrease its duration, the surface of the PR can be hardened with a 30 s pre-development step before the lithography.

Una vez que el dispositivo MTJ ha sido definido y sin haber eliminado el photoresist, se deposita una capa de aislante de Al₂O₃ de 40 nm de espesor durante unos 60 minutos en el equipo de deposición UHV2 mediante la técnica de deposición catódica o sputtering. Esta capa de aislamiento evita que la corriente pueda circular desde el electrodo superior al inferior sin atravesar la zona de barrera de MgO. Tras la deposición del óxido, el photoresist existente del paso anterior es utilizado como máscara para el proceso de lift-off, figura 7.11(b). *Top View*

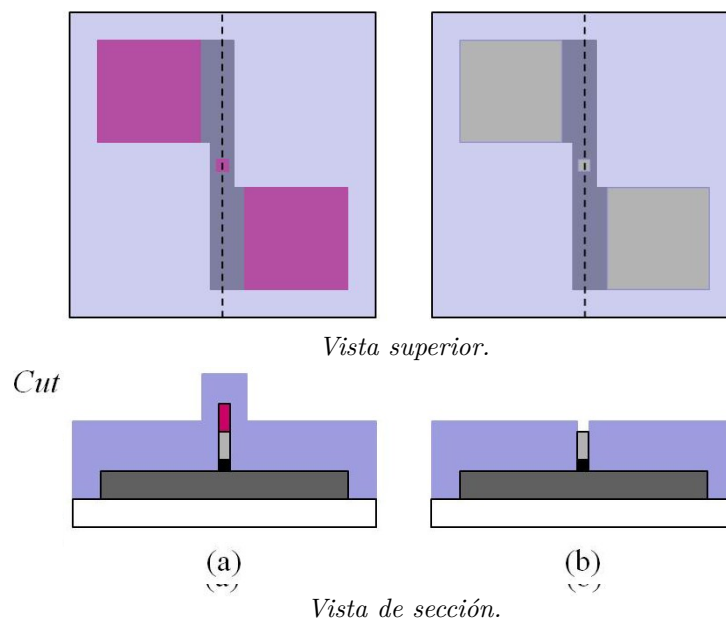


Figura 7.11 – Deposición de capa de aislamiento sobre la muestra. La capa de aislante es depositada (a) y posteriormente mediante lift-off se abre una acceso hacia los electrodos superiores de la estructura MTJ (b).

and then a via is opened, through lift-off, to both the junction area and the bottom contact pads (b). La duración del proceso de lift-off, paso 9 del Run Sheet, aumenta con el decremento del área del dispositivo MTJ y el incremento de grosor del óxido. Para dispositivos MTJ de algunas μm² y 40 nm de grosor de alúmina Al₂O₃ puede extenderse durante varios días

3.3.2.4. Metallization

After the insulator lift-off is complete, the via to the junction area and bottom contact pads are open. In this process step, both the top lead and contact pads are defined by lift-off as

Simulación de circuitos con modelos avanzados en Verilog-A. 209
illustrated in Figure 3-20. After the patterning is complete, the metallization sequence is performed in the Nordiko 7000 system (see section 3.1.1.2 and Table 3-2).

(aproximadamente tres días), debido mayormente al perfil no uniforme del *photoresist*. Con la intención de aumentar el porcentaje de éxito y disminuir el tiempo del *lift-off* la superficie del *photoresist* puede ser endurecida con unos 30 segundos de pre-revelado antes de la realización de la litografía.

7.1.5 Primera metalización.

Una vez que ha finalizado el proceso de *lift-off* del aislante, [paso 9 del Run Sheet](#), los contactos del electrodo superior del dispositivo **MTJ** quedan accesibles. En este proceso de metalización se definen tanto el contacto superior del dispositivo como los 40 *pads* de conexión al encapsulado ($420 \times 420 \mu\text{m}^2$). Es necesario un nuevo proceso de fotolitografía con la máscara L3MET no invertida mediante el láser DWL 2.0, [paso 10 del Run Sheet](#), ver figura 7.12. Tras el revelado del *photoresist* se realiza la deposición del $\text{Al}_{98,5}\text{Si}_{0,5}\text{Cu}_1$ (3000 Å)/ TiWN_2 (150 Å) en la máquina Nordiko 7000. El material depositado ~~de *Top View*~~ ~~co *Top View*~~

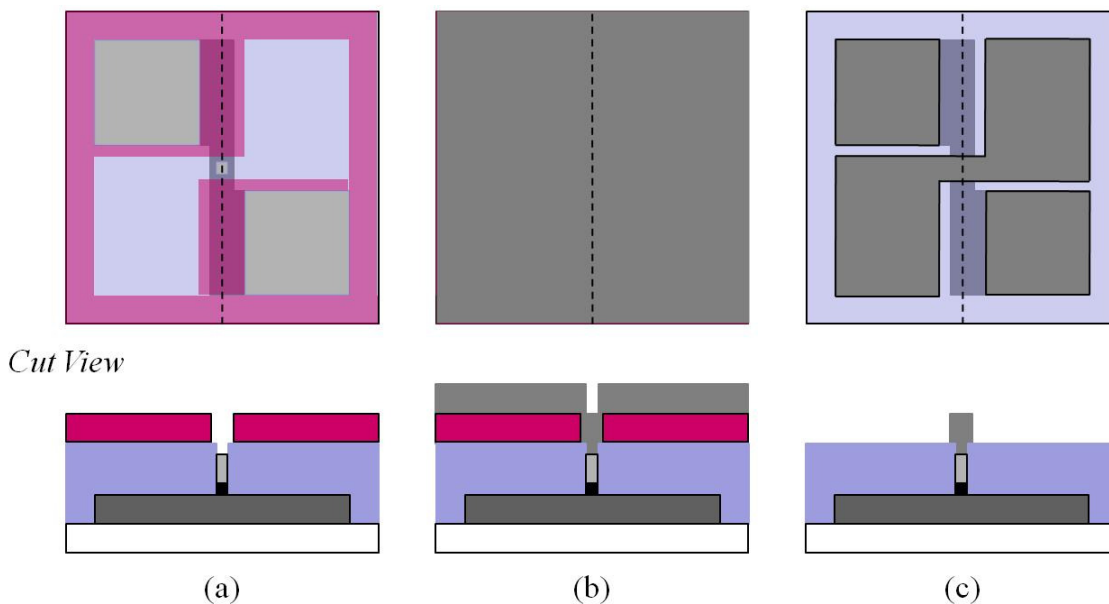


Figure 3-20: Schematic top and cut view of the metalization process that defines the final contact leads. The sample is patterned defining the top lead and contacts, and also the bottom pads (a). Through a sputter-etch step and two metal layers deposition (described in Table 3-2) the entire sample is covered with metal (b). After the final lift-off step, the metal remains only on the contact pads and the top lead (c).

El proceso de *lift-off*, [paso 12 del Run Sheet](#), posterior a la metalización, finaliza con la revisión en el microscopio óptico, figura 7.13 y la comprobación de que todo el metal sobrante ha sido retirado. The final lift-off completion is determined by optical inspection with a microscope (see Figure 3-21). The standard micro-fabrication process is complete when all the metal leftovers are removed.

The final lift-off completion is determined by optical inspection with a microscope (see Figure 3-21). The standard micro-fabrication process is complete when all the metal leftovers are removed.



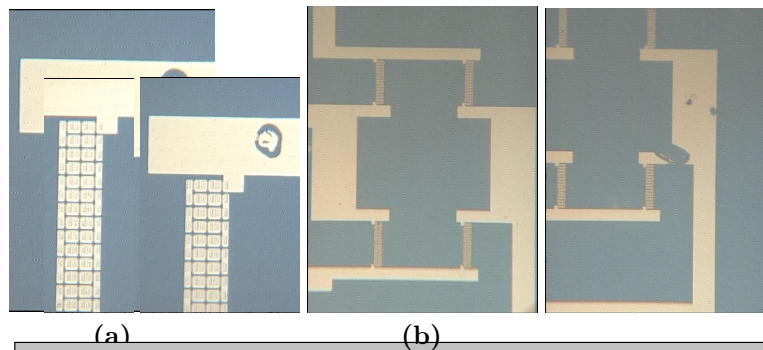


Figura 7.13 – Revisión con el microscopio óptico (a) de las muestras (b).

7.1.6 Deposición de capa de aislante SiO₂ sobre pistas de corriente.

Se deposita una capa gruesa de óxido para evitar problemas de aislamiento entre el dispositivo y sus contactos y la pista de corriente del sensor que será depositada sobre este óxido en el apartado 7.1.8. En la figura 7.14 se representa la deposición de la capa de óxido de silicio que se corresponde con el paso 13 del Run Sheet que se realiza en la máquina Alcatel SCM 450 mediante pulverización catódica o sputtering.

Deposition Time	SiO ₂ thickness	Ar gas flow	Pressure	Power Source	Rotation
3hrs	5000Å	20 sccm	3.0 mtorr	140W	4 rpm

Comments: Sample

STEP 14
Machine: V
Run Sheet fo

Date: __/__/__
Pág. 10

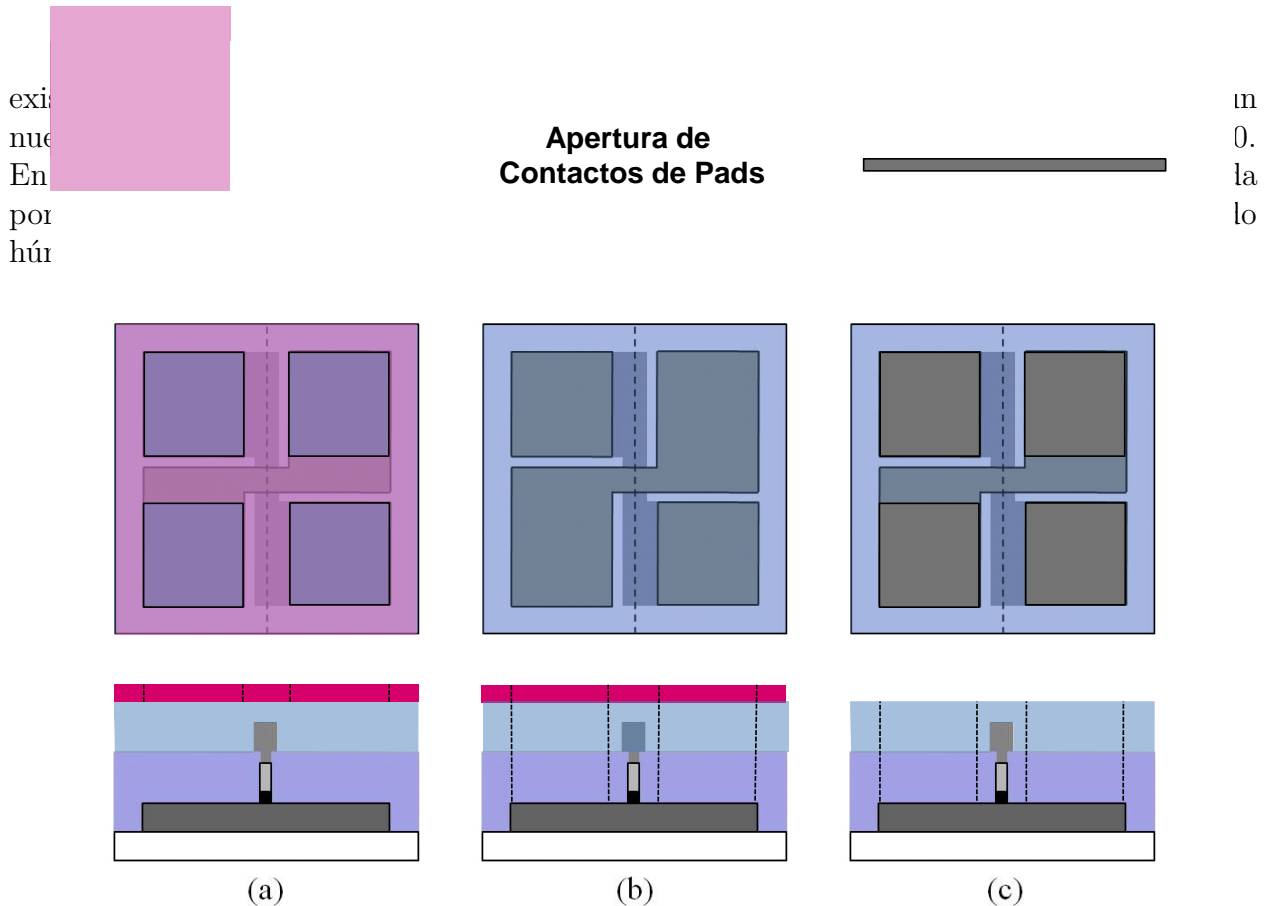
Figura 7.14 – Deposición de la capa de óxido de 5000 Å que aísla al dispositivo MTJ y sus contactos de la pista de corriente del sensor.

La duración del proceso de deposición de la capa gruesa de óxido es de unas 4 horas y durante este tiempo se introduce una rotación en la muestra de 4 rpm para asegurar una deposición homogénea.

7.1.7 Apertura de los pads enterrados bajo el óxido.

Para permitir el acceso a la zona metálica de los 40 pads del perímetro exterior y los dispositivos de test y verificación del centro de la máscara, es necesario retirar el óxido





Vista de sección.

Figura 7.15 – *Vistas del proceso de apertura de los pads en el óxido de aislamiento de 5000 Å. Se deposita el photoresist (a), mediante la iluminación con máscara y el posterior revelado (b) y a través del ataque húmedo RIE se alcanzan las metalizaciones (c).*

El interior de los polígonos de la máscara L4PAD definen la zona de *pads* donde existirá contacto eléctrico. La máscara real utilizada en el proceso litográfico está invertida, de manera que sólo se ilumina el interior de los cuadrados amarillos. Y las zonas que quedan expuestas a la luz pierden la capa gruesa de aislante durante el proceso de ataque húmedo del SiO₂.

En este proceso no se utiliza la técnica de *lift-off* porque el disolvente utilizado durante el revelado del *photoresist* oxida y corroe la capa metálica de aluminio. Aunque la alúmina Al₂O₃ presenta mejores características como lámina aislante por ser menos porosa y más compacta que el SiO₂, ésta no puede ser eliminada por ataque químico en la máquina LAM™ disponible en la Sala Limpia. Sin embargo, la alúmina puede ser atacada en los sistemas IBD, pero al ser un óxido muy compacto el ritmo de ataque es muy lento (menor de 1 Å/min). Esto conlleva decenas de horas de procesado para eliminar los 5000 Å existentes. El ritmo de ataque del SiO₂ en la máquina LAM es de 8 Å/s aproximadamente, lo que implica una duración del proceso de unos 6 minutos.



Figura 7.16 – Sistema de ataque por plasma de reacción química de Research Rainbow .

Tras el proceso fotolitográfico, se inicia una fase de ataque por plasma húmedo RIE- (*Reactive Ion Etching*) en la máquina LAM de Research Rainbow™, imagen 7.16. Este proceso conforma el [paso 15 del Run Sheet](#) que tiene que ir seguido de la eliminación del *photoresist* mediante *lift-off* que se realiza en el [paso 16 del Run Sheet](#) en el baño atemperado de ultrasonidos del disolvente orgánico *microstrip*.

7.1.8 Segunda metalización: pistas de corriente.

Una vez finalizado el proceso de *lift-off* de la capa gruesa de aislante, [paso 16 del Run Sheet](#), se procede con la metalización de las líneas de corriente del sensor. En esta fase se definen tanto las pistas de corriente como nuevamente los 40 *pads* del perímetro exterior y también de los dispositivos de test (máscara L5MET no invertida) mediante el láser DWL 2.0, [paso 17 del Run Sheet](#), ver figura 7.17. Tras el revelado del *photoresist* se realiza la deposición del $\text{Al}_{98,5}\text{Si}_{0,5}\text{Cu}_1$ (3000 Å)/ TiWN_2 (150 Å) en la máquina Nordiko 7000.

En máscara L5CUR se define la zona donde se deposita el metal y los *pads* perimetrales donde existe contacto eléctrico y donde se realiza el *wire-bonding* de conexión al zócalo de 40 *pines*. La máscara es no invertida, de manera que se iluminará el interior de los objetos grises. Las zonas que no quedan expuestas a la luz del láser serán las que pierdan la capa metálica tras el proceso de *lift-off*.

7.1.9 Pasivación.

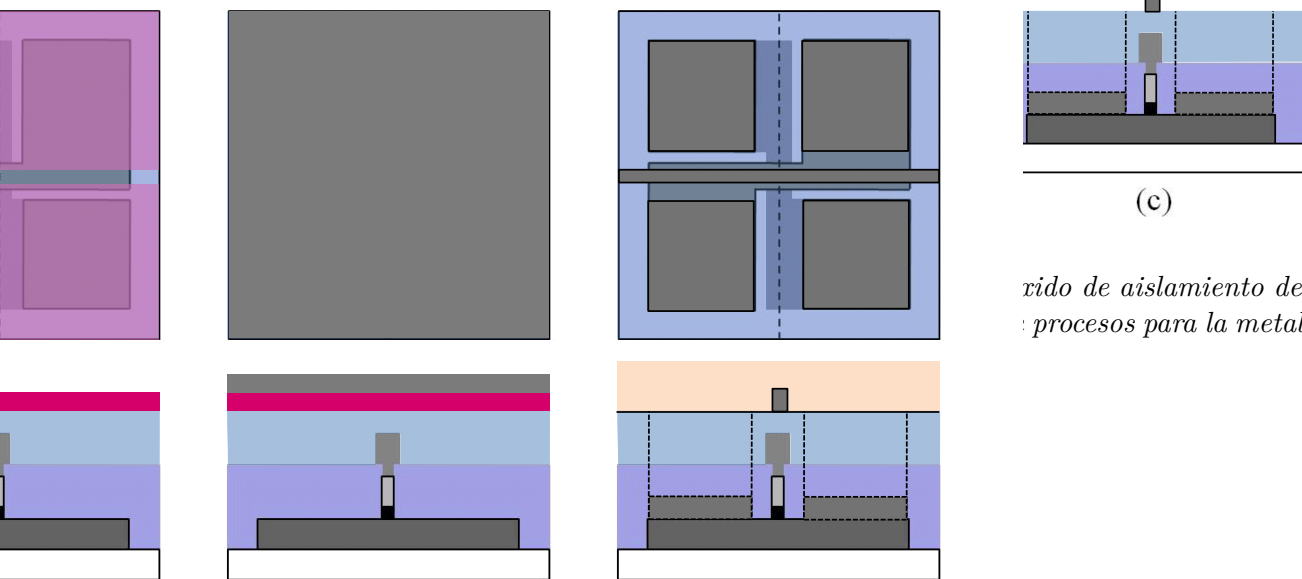
Se deposita una capa gruesa de SiO_2 de unos 2000 Å para evitar problemas corrosión de las pistas de corriente del sensor. En la figura 7.18 se representa la deposición de la capa aislante de SiO_2 que corresponde con el [paso 20 del Run Sheet](#), realizada en la máquina Alcatel SCM 450 mediante pulverización catódica o *sputtering*.

A

Deposición Metalización Pistas Corriente



Deposición de SiO2 sobre Metalización Pistas Corriente



(c)

Capa de aislamiento de 5000 Å
deposicionada en los procesos para la metalización.

Figura 7.18 – Deposición de la capa de óxido de 2000 Å para aislar las pistas de corriente de la atmósfera externa del dispositivo MTJ.

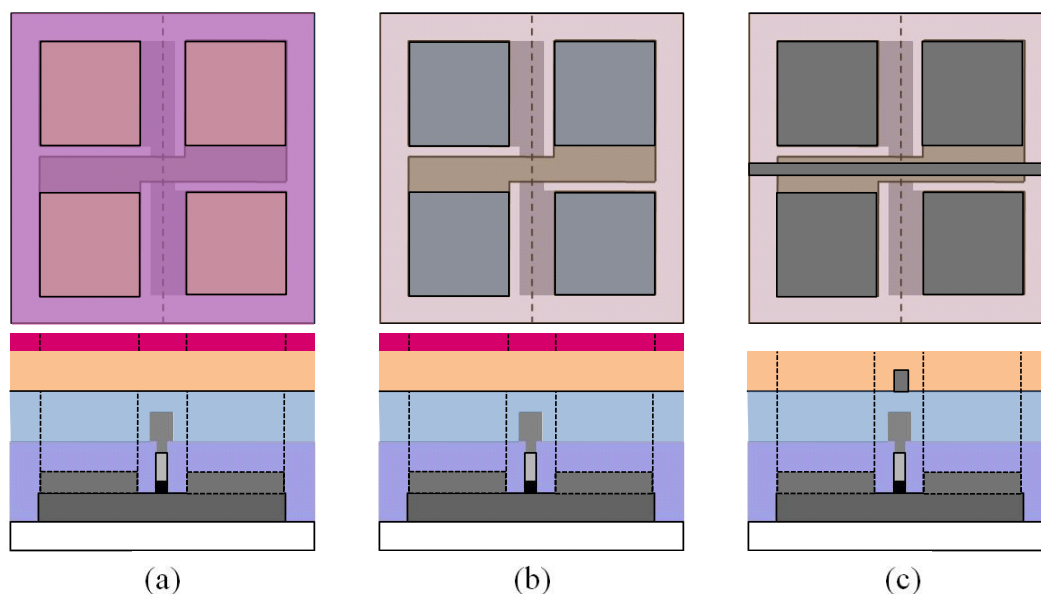
La duración del proceso de deposición de la capa gruesa de SiO₂ es de unas 2 horas. Durante este tiempo se introduce una rotación de homogeneización en la muestra de 4 rpm.

7.1.10 Apertura final de los *pads* enterrados.

Para permitir el acceso a los *pads* metálicos es necesario retirar el óxido crecido sobre ellos. A tal efecto se realiza el [paso 21 del Run Sheet](#) donde se inicia un proceso fotolitográfico con la máscara L4PAD no invertida utilizada también en el [paso 11 del Run Sheet](#) mediante el láser DWL 2.0. En la figura 7.19 se detalla el proceso de deposición de la capa *photoresist* que es iluminado por el láser con la máscara invertida para únicamente eliminar tras el proceso de revelado las zonas de los *pads* de contacto.

Tras el proceso fotolitográfico, se inicia una fase de atacado por plasma RIE. Este proceso conforma el [paso 22 del Run Sheet](#) que sigue con la eliminación del *photoresist* mediante *lift-off*, [paso 23 del Run Sheet](#).

Apertura de PAD enterrados por RIE



Vista de sección.

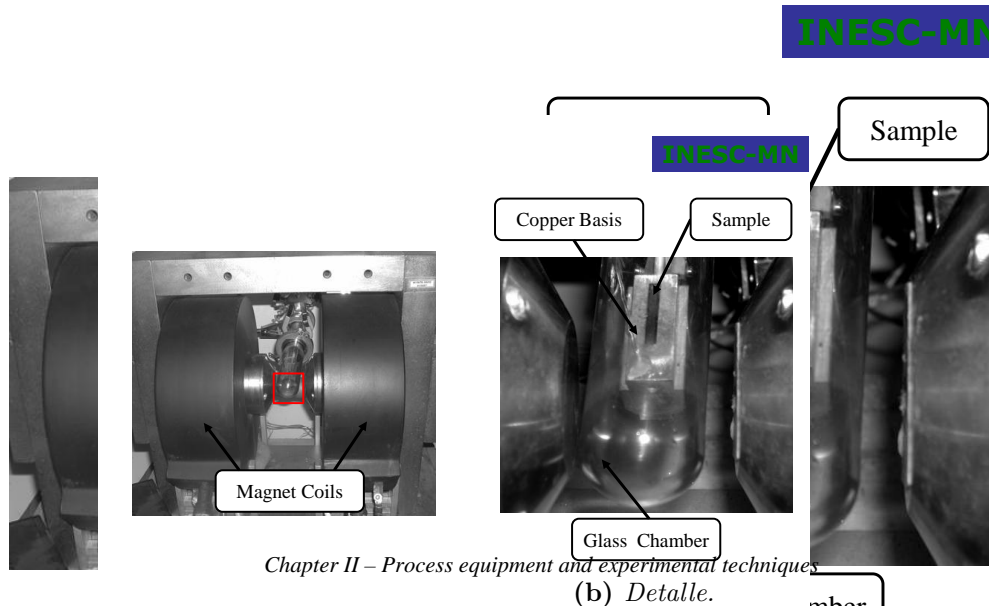
Figura 7.19 – Vistas de la apertura final de los pads de los dispositivos de test. Sobre el óxido de aislamiento de 2000 Å se deposita el photoresist (a), para la iluminación con máscara y el posterior revelado (b). Mediante atacado húmedo RIE se elimina el óxido existente inicialmente sobre los pads metálicos (c).

Una vez finalizado todo el proceso de fabricación del sensor es necesario realizar un recocido para obtener las mejores prestaciones.

7.1.11 Tratamiento térmico magnético

El *Magnetic Thermal Annealing* es un tratamiento térmico magnético que permite obtener las características óptimas del dispositivo magnetorresistivo. El proceso se realiza con el equipo de recocido térmico de la figura 7.20 y se divide en los siguientes pasos:

1. Arranque de la bomba de vacío mecánica para obtener una presión menor de 10^{-3} Torr.
2. Obtención de una presión menor de 10^{-6} Torr mediante el uso de una bomba especial *turbo molecular pump* de altas prestaciones.
3. Inicio de una fase de calentamiento con una pendiente en ($^{\circ}\text{C}/\text{min}$) establecida hasta que se alcance una temperatura que será mantenida durante un determinado intervalo de tiempo y un posterior enfriado natural. El ciclo de temperatura se muestra en la figura 7.21.
4. Presurizado atmosférico de la cámara de vacío para extracción de la muestra.



Chapter II – Process equipment and experimental techniques

(b) Detalle.

Figura 7.20 – Imágenes del sistema de annealing (a). La cubierta de cristal extraíble (b) rodea al soporte de cobre que es calentado por una lámpara halógena de 100 W (12V). La separación de las bobinas se ajusta para obtener el campo magnético deseado en la zona de la muestra.

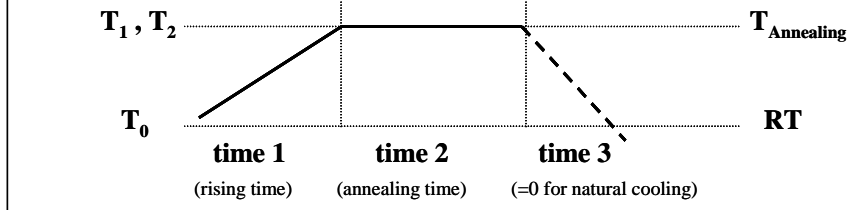
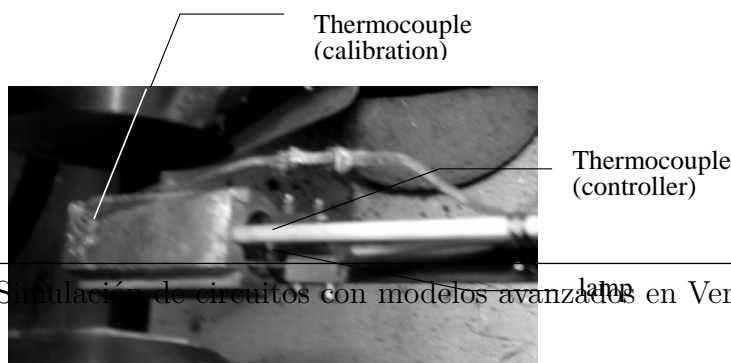
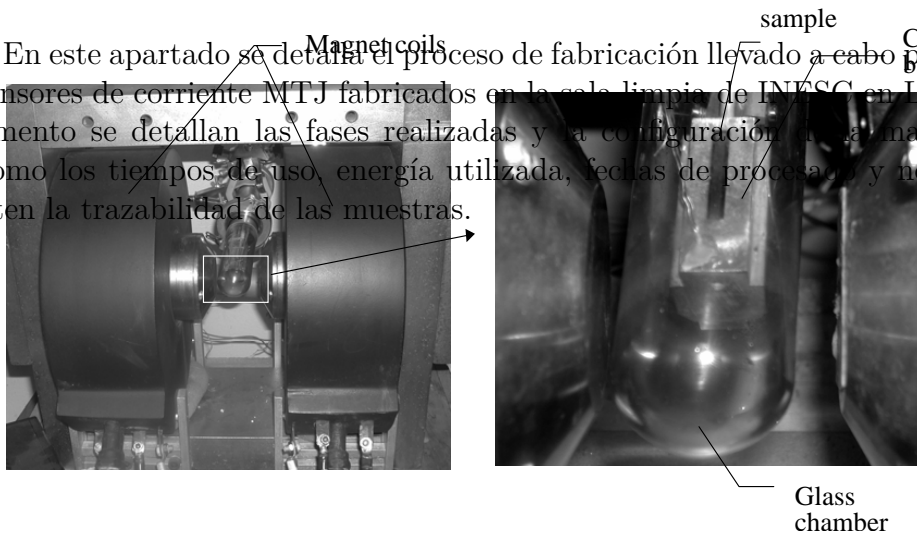


Figura 7.21 – Perfil de temperatura durante el calentamiento para el tratamiento térmico del annealing. Figure II.12 – Heating-cooling profile used for the thermal treatments.

7.2 Run Sheet

En este apartado se detalla el proceso de fabricación llevado a cabo para la obtención de los sensores de corriente MTJ fabricados en la sala limpia de INESC en Lisboa. En el citado documento se detallan las fases realizadas y la configuración de la maquinaria requerida así como los tiempos de uso, energía utilizada, fechas de procesamiento y notas de interés que faciliten la trazabilidad de las muestras.



Run Sheet - MTJ

RUN: 4 different

Process Start : 05/04/2011

Process Finish : 06/06/2011

4 dies of Si/SiO₂ (1000 Å) substrate 1x1 inch²

STEP 1 Tunnel Junction Deposition

Date: ___/___/___

Machine: Nordiko 2000

Comments:

- **Bottom electrode:** Ta 50 Å/ Ru 180 Å/ Ta 30 Å/ PtMn 200 Å/ CoFe 20 Å/ Ru 9 Å/ CoFeB 30 Å
- **Tunnel junction** structure: MgO 10 Å
- **Top electrode:** CoFeB y Å/ Ru 100 Å
- Total height: ≈ 660 Å

SAF

Samples:

- 428 : CoFeB 15 Å
- 429 : CoFeB 30 Å
- 430 : CoFeB 30 Å
- 431 : CoFeB 15.5 Å

Calibration samples: VSM, Top electrode for 2nd etch (glass substrate)

STEP 2 TMR passivation – 150 Å TiWN₂

Date: ___/___/___

Machine: Nordiko 7000

Seq.17 – mod.2 – f.9 (1' soft sputter etch) P=60W/40W, p=3mTorr, 50 sccm Ar
 mod 3 – f.19 (150 Å TiW, 27'') P=0.5 kW, 3mTorr, 50sccm Ar + 10 sccm N₂

Readings – Module 2

Run#	Power1	Power2	Gas flux	Pressure
	F59R8B1	F40R1	50.2 sccm Ar	3.1mTorr

Readings – Module 3

Run#	Power	Voltage	Current	Gas flux	Pressure
	0.5kW	419V	1.2A	(50.5 Ar +10.6 N ₂)sccm	3.0 mTorr

Calibration samples: Top electrode for 2nd etch

Step 1: Read Values

B.P: 8.6×10^{-8} Torr

Seq. 3: pre sputtering of all targets x2

B.P: 8.4×10^{-8} Torr

Sequence	Function	Read Values:
38	F18, Ta 50	40mA/ 331V/ 10W/ 9.7sccm/ 4.6mT/ S4T3/ 100%
	F7, Ru 180	40mA/ 302V/ 10W/ 7.8sccm/ 5.1mT/ S4T3/ 100%
	F6, Ta 30	40mA/ 329V/ 10W/ 9.8sccm/ 4.6mT/ S4T3/ 100%
	F5, PtMn 200	30mA/ 292V/ 10W/ 8.9sccm/ 5.1mT/ S4T3/ 100%
	F9, CoFe 20	F34R3B271 / 7.7sccm/ 5.2mT/ S4T6/ 100%
	F3, Ru 9	40mA/ 292V/ 10W/ 7.8sccm/ 7.8mT/ S4T3/ 100%
	F47, CoFeB 30	40mA/ 417V/ 10W/ 8.7sccm/ 5.2mT/ S4T3/ 100%
	F94, MgO Cleaning, 2'30''	F149R0B291 / 9.6sccm/ 5.1mT/ S4T2/ 100%
39	F51, MgO x	F129R0B272 / 9.5sccm/ 18.2mT/ S4T5/ 50%
	F50, CoFeB y	40mA/ 418V/ 10W/ 8.8sccm/ 5.1mT/ S4T3/ 100%
	F4, Ru 50	40mA/ 302V/ 10W/ 7.8sccm/ 5.1mT/ S4T3/ 100%
	F18, Ta 50	40mA/ 330V/ 10W/ 9.9sccm/ 4.6mT/ S4T3/ 100%

STEP 3 1st Lithography – Stack definition

Date: 08 / 04 / 2011

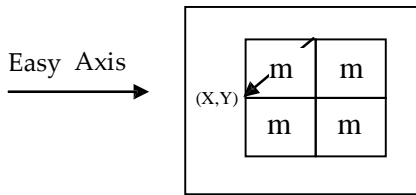
Machine: Vacuum bake/vapour prime; photoresist tracks; DWL

Die size: 8.2 x 8.2 mm²

Coating: Recipe 6/2

Mask: ValSV_L1BOT_inverted

Map: Raquel2 (The same for the 4 dies) J. Bernardo



Energy : 80
Power : 120mW

Focus : -50

Alignment marks: none

Develop : Recipe 6/2

Development time : 1 min

Optical Inspection:

STEP 4 1st Ion Milling – Stack definition Date: 11 / 04 / 2011

Machine: N3600

Total thickness to etch: 660 Å (etch rate: ~1 Å /s → time: 660 s)

Base Pressure (Torr): 2.04x10⁻⁷Torr

Batc : etchjunction

Recipe etch junction stack all : etch pan 60 deg
cool_down_200s

Assist Gun: 160W 105mA +750V/-350V 12sccm Ar; Assist Neut: 30% subst.rot 60° subst.pan

Assist Gun	Power (W)	V+ (V)	I+ (mA)	V- (V)	I- (mA)	Ar Flux (sccm)
Read Values	195	723.5	104.5	344.3	2.6	10.6

STEP 5 Resist Strip Date: 11 / 04 / 2011

Machine: Chemical Workbench: Susana Cardoso

Started: _____

Stoped: _____

Total Time in Hot Micro-Strip : 5hours

Ultrasonic Time : 4 x 10min

Optical Inspection:

Sample	Comments
Ok!	No photo resist

STEP 6 2nd Lithography – Top Electrode and Junction Definition Date: ___/ ___/

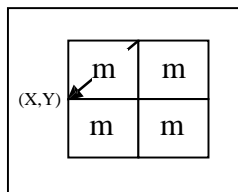
Machine: Vacuum bake/vapour prime; photoresist tracks; DWL

Coating: Recipe 6/2

Mask: ValSV_L2TOP_inverted

Map: Raquel2

Easy Axis
→



Energy : 80
Power : 120mW
Focus : -50
Alignment marks: (168, 54) μm
OFFSET: (-168, -452) μm

Develop : Recipe 6/2

Development time : 1 min

Optical Inspection:

STEP 7 2nd Ion Milling – Top Electrode and Junction Definition

Date: 28 / 04 / 2011

Machine: N3000

Total thickness to etch: Ru 100 Å + CoFeB 30 Å + MgO 10 Å + SAF 59 Å \approx 200 Å (etch rate: \sim 1Å/s → time: 200 s)

Base Pressure (Torr): 2.12×10^{-7} Torr T Cryo (K): 107K

Standard Etching Recipe (function_etch_angle_70°and40°):
etch pan 70° (140 seg)
etch pan 30° (60 seg)

Assist Gun: 160W 105mA +735V/-350V 12sccm Ar; Assist Neut: 30% subst.rot 60° subst.pan

Calibration Sample	Structure
	CoFe20 Å/Ru 9 Å/CoFeB 30 Å/MgO x Å/CoFeB y Å/Ru 50 Å/Ta 50 Å/ TiW(N) 150 Å

Assist Gun	Power (W)	V+ (V)	I+ (mA)	V- (V)	I- (mA)	Ar Flux (sccm)
Read Values	58	500	29.0	200.0	-	8.0

Wafer	samples	Etching Turn	Time	Effect
		Etch pan 70°	140''	----
		Etch pan 40°	60''	Calibration sample still has material
		Etch pan 40°	20''	Finished: Calibration sample cleared!

Composition of the calibration sample that must be put in the same holder together with the sample :
CoFe 20 Å/ Ru 9 Å/ CoFeB 30 Å/ MgO x Å/ CoFeB y Å/ Ru 50 Å/ Ta 50 Å/ TiW(N) 150 Å
(The etch should be stopped when the calibration sample is completely transparent)

STEP 8 Insulating Layer Deposition

Date End: 4 / 05 / 2011.

Responsible: **Fernando**

Machine: **UHV2**

Dep. Rate \sim 11.43 Å/min

Deposition Time	Al2O3 thickness	Ar gas flow	Pressure	Power Source
-----------------	-----------------	-------------	----------	--------------

62min	>1000 Å	45 sccm	3.0 mTorr	200W
-------	---------	---------	-----------	------

STEP 9 Oxide Lift-Off Date: 4 / 05 /2011

Machine: Chemical Workbench

Started: _____
 Stopped: _____

Total Time in Hot Micro-Strip : 4days

Ultrasasonic Time : 3x10min

Optical Inspection:

STEP 10 3rd Exposure – Sensor Contacts Date: 4 / 05 /2011

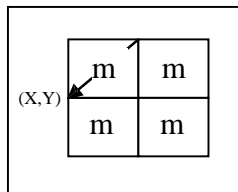
Machine: Vacuum bake/vapour prime; photoresist tracks; DWL

Coating: Recipe 6/2

Mask: ValSV_L3MET_noninverted

Map: Raquel2

Easy Axis
 →



Power : 120mW
 Focus : -50
 Alignment marks: (168, 54) µm
 OFFSET: (-168, -452) µm
 (-168, -452) µm

Energy : 80

Develop : Recipe 6/2

Development time : 1 min

Optical Inspection:

STEP 11 Contact Leads Deposition $Al_{98.5}Si_{0.5}Cu_1$ 3000 Å / TiWN₂ 150 Å Date: 4 / 05 / 2011

Machine: Nordiko 7000

Seq.48 (svpad) –
 mod.2 – f.9 (1' soft sputter etch) P=60W/40W, p=3mTorr, 50 sccm Ar
 mod.4 – f.1 (3000 Å Al, 80'') P=2 kW, 3mTorr, 50 sccm Ar
 mod 3 – f.19 (150 Å TiWN₂, 27'') P=0.5 kW, 3mTorr, 50sccm Ar + 10 sccm N₂

Readings – Module 2					
Run#	Power1	Power2	Gas flux	Pressure	
	F60R8B122	F40R2	50.2 sccm Ar	3.2mT	

Readings – Module 4					
Run#	Power	Voltage	Current	Gas flux	Pressure
	2kW	404V	5.0A	50.2 sccm Ar	3.1 mT

Readings – Module 3					
Run#	Power	Voltage	Current	Gas flux	Pressure
	0.5kW	417V	1.2A	(50.2Ar + 10.6 N ₂) sccm	2.9 mT

STEP 12 **Metal lift-off**

Date: 5 / 5 / 2011

Machine: Chemical Workbench

Started: _____

Stoped: _____

Total Time in Hot Micro-Strip : 18hours

Ultrasonic Time : 40 min

Optical Inspection:

STEP 13 **Insulating Layer Deposition**

Date: 5 / 5 / 2011

Responsible: **Fernando**

Machine: **Alcatel**

Deposition Time	SiO ₂ thickness	Ar gas flow	Pressure	Power Source	Rotation
3h45	5000 Å	20 sccm	3.0 mTorr	140W	4 rpm

Comments: Sample

STEP 14 **4th Lithography – Sensor pads protection (before passivation)**

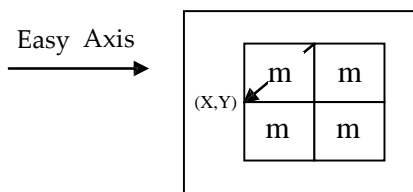
Date: __/ __/ __

Machine: **Vacuum bake/vapour prime; photoresist tracks; DWL**

Coating: Recipe 6/2

Mask: ValSV_L4PAD_non_inverted

Map: Raquel2



Energy : 80

Power : 120mW

Focus : -50

Alignment marks: (168, 54) µm

OFFSET: (-168, -452) µm

(-168, -452) µm

(-168, -452) µm

Develop : Recipe 6/2

Development time : 1 min

Optical Inspection:

STEP 15 **RIE etch**

Date: __/ __/ __

Machine: LAM
Responsible: Virgínia Soares

Recipe: Low_power_noO2

Set Points: 100W RF, Chamber Pressure 140 mT, Gap 1.3cm / 200 sccm Ar + 100 sccm CF₄/
He Clamp 140 Torr/ maximum time_step = 150s

(A calibration sample is required)

STEP 16 **Photo-resist Strip**

Date: __/ __/ __

Machine: Chemical Workbench

Started: _____
Stoped: _____

Total Time in Hot Micro-Strip : 1day Ultrasonic Time : 20min

Optical Inspection:

STEP 17 **5th Lithography – Current Lines definition**

Date: __/ __/ __

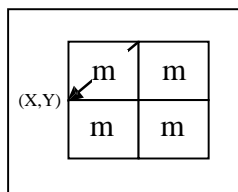
Machine: Vacuum bake/vapour prime; photoresist tracks; DWL

Coating: Recipe 6/2

Mask: ValSV_L5CUR_non_inverted

Map: Raquel2

Easy Axis
→



Energy : 80

Power : 120mW Focus : -50

Alignment marks: (168, 54) μ m

OFFSET: (-168, -452) μ m

(-168, -452) μ m

(-168, -452) μ m

(-168, -452) μ m

Develop : Recipe 6/2

Development time : 1 min

Optical Inspection:

STEP 18 Current Lines Deposition Al_{98.5}Si_{0.5}Cu_{0.5} 3000 Å /TiWN₂ 150 Å Date: ___/___/___

Machine: Nordiko 7000

Seq.48 (svpad) - mod.2 - f.9 (1' soft sputter etch) P=60W/40W, p=3mTorr, 50 sccm Ar
 mod.4 - f.1 (3000A Al, 1'20'') P=2 kW, 3mTorr, 50 sccm Ar
 mod 3 - f.19 (150A TiW, 27'') P=0.5 kW, 3mTorr, 50sccm Ar + 10 sccm N₂

Readings – Module 2					
Run#	Power1	Power2	Gas flux	Pressure	
	F59R2B124	F40R2	50.2 sccm Ar	3.2mT	

Readings – Module 4					
Run#	Power	Voltage	Current	Gas flux	Pressure
	2kW	399V	5.1A	50.1Ar sccm	3.0 mT

Readings – Module 3					
Run#	Power	Voltage	Current	Gas flux	Pressure
	0.5kW	416V	1.2A	(50.2Ar + 10.7 N ₂) sccm	3.0 mT

STEP 19 Metal lift-off Date: ___/___/___

Machine: Chemical Workbench

Started: _____
 Stopped: _____

Total Time in Hot Micro-Strip : 3 days Ultrasonic Time: 3 hours

Optical Inspection:

STEP 20 Passivation (2000 Å SiO₂) Date: ___/___/___

Responsible: **Fernando**
 Machine: **Alcatel**

Deposition Time	Al ₂ O ₃ thickness	Ar gas flow	Pressure	Power Source
	2000 Å	45 sccm		200W

STEP 21 **6th Exposure** – Pads protection

Date: ___/___/___

Machine: **Vacuum bake/vapour prime; photoresist tracks; DWL**

Coating: Recipe 6/2

Mask: ValSV_L4PAD_non_inverted

Map: Raquel

Develop : Recipe 6/2

Development time : 1 min

Optical Inspection:

STEP 22 **RIE Etch** (2000 Å SiO₂)

Date: 02 / 06 / 2011

Machine: LAM

Responsible: Virgínia Soares

Recipe: Low_power_noO2

Set Points: 100W RF, Chamber Pressure 140 mT, Gap 1.3cm / 200 sccm Ar + 100 sccm CF₄/
He Clamp 140 Torr/ maximum time_step = 150s

(A calibration sample is required)

STEP 23 **Resist Strip**

Date: 06 / 06 / 2011

Acetone + ultrasonic

Rinse with IPA + DI water + dry compressed air

Started: 1h

Stoped:

Total Time in acetone: 1 hour

Ultrasonic Time : 20 min

Optical inspection:

STEP 24 **Anneal**

Date: 06 / 06 / 2011

Setup: Old annealing setup

Temperature: 320°C real (corresponds to 300 °C as setpoint)

Annealing Time: 1h

H Field: 4kOe

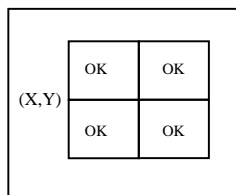
Setup: Old annealing setup

Temperature: 320°C real (corresponds to 300 °C as set-point)

Annealing Time: 1h

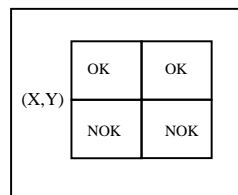
H Field: 4kOe

Sample 428



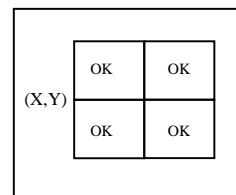
3: encapsulated
1: probe station

Sample 429



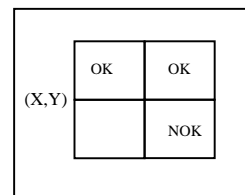
2: encapsulated

Sample 430



3: encapsulated
1: probe station

Sample 431



2: encapsulated

7.3 Estructuras de datos de SPICE

7.3.1 Estructura de datos del dispositivo.

Las estructuras de datos que soportan a todos los dispositivos (resistencias, fuentes, diodos, etc.) en SPICE3 se pueden dividir en tres categorías diferentes:

1. Aquellas que describen los dispositivos al nivel requerido por el interfaz de usuario.
2. Las que definen los diferentes dispositivos al nivel requerido por el simulador.
3. Las estructuras encapsuladas dentro de las estructuras de los mismos dispositivos.

Las estructuras de los dispositivos son bastante variables porque soportan los datos de modelos muy complejos y diferentes, sin embargo, contienen unas partes comunes que serán descritas en los próximos apartados.

7.3.1.1 Estructura GENmodel

La estructura GENmodel define la parte estándar que debe aparecer al comienzo de todos los modelos de dispositivos. Esta estructura permite a SPICE3 la búsqueda entre los modelos de dispositivos sin conocimiento alguno de los dispositivos en sí mismos. Los campos que forman esta estructura son variados.

```

1 typedef struct sGENmodel{ //Modelo para la estructura de un dispositivo
2   int GENmodType; //Identificador de este tipo de modelo. Permite que el código acceda a las
   rutinas correctas para procesar el modelo referenciado por la estructura SPICEdev.
3   struct sGENmodel *GENnextModel; //Puntero al siguiente modelo de la lista enlazada. Ese
   modelo debe tener el mismo valor de GENmodType y un nombre distinto, pero no se conoce
   otra información de éste.
4   GENinstance * GENinstances; //Puntero a una lista enlazada de los elementos del circuito
   del mismo modelo. Los elementos tienen una estructura que permite recorrer la lista sin
   tener conocimiento de los dispositivos.
5   IFuid GENmodelName; //Puntero a una cadena de caracteres con el nombre del modelo.
6 } GENmodel;
```

Listado 7.1 – Estructura GENmodel.

La estructura GENmodel se puede representar gráficamente, ver figura 7.22 para aclarar la relación modelo-elemento.

7.3.1.2 Estructura GENinstance.

La estructura GENinstance sirve para atravesar la estructura de datos de los dispositivos en busca de un elemento concreto sin conocimiento de las características de los dispositivos. Todos los dispositivos pueden ser inspeccionados con esta estructura si se tiene la precaución de no referenciar un nodo con un número mayor que el número de nodos definidos para un dispositivo de este tipo en la estructura SPICEdev correspondiente.

```

1 typedef struct sGENinstance{
2   struct sGENmodel *GENmodPtr; //Puntero que referencia el modelo el modelo de este elemento
   (backpointer). Permite el acceso a los parámetros del modelo cuando solo se dispone de
   un puntero a un elemento.
3   struct sGENinstance *GENnextInstance; //Puntero al próximo elemento de la lista enlazada
   de elementos de modelo.
4   IFuid GENname; //Puntero a una cadena de caracteres con el nombre del elemento
5   modelo int GENstate; //Número del índice del estado
6   int GENnode1; // Número del nodo para el elemento. No todos los nodos son
7   int GENnode2; // válidos en todos los elementos. Cuando se referencia a
```



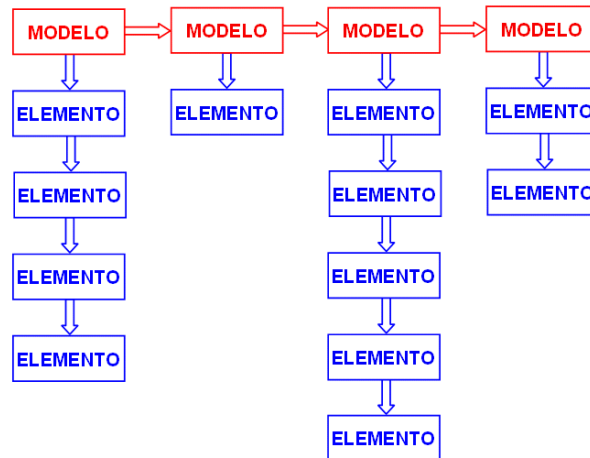


Figura 7.22 – Relación modelo-elemento.

```

8  int GENnode3; // un nodo o terminal (GENnodei) con i mayor que
9  int GENnode4; // (DEVices[inst->GENmodPTR->GENmodType])->DEVpublic.terms
10 int GENnode5; // es un error y producirá efectos impredecibles
11 int GENnode6; // Añadido para crear el Nodo Body en MOS
12 int GENnode7; // Añadido para crear el Nodo Temp en MOS
13 } GENinstance ;

```

Listado 7.2 – Estructura GENinstance.

7.3.2 Estructura de datos de CIRCUITOS

El módulo de circuitos contiene todas estructuras abstractas de datos que describen completamente un circuito electrónico. En esta descripción no existe referencia alguna al tipo de análisis a realizar ni el conjunto de dispositivos que lo forman. A continuación se presentan las estructuras *CKTcircuit* y *CKTnode*.

7.3.2.1 Estructura CKTcircuit

Para detallar lo explicado en el apartado EstructuraDatosEncapsuladosSPICE3, se presenta la descripción del tipo abstracto de datos denominado *CKTcircuit*, definido en la implementación en lenguaje C utilizada en el simulador para contener los datos relativos a un circuito electrónico.

```

1 typedef struct {
2     GENmodel *CKThead[MAXNUMDEVS]; //Array que apunta al 1er modelo
3     STATistics *CKTstat; //Array que almacena las estadísticas
4     double *(CKTstates [8] ); //Array que almacenan los datos per-timepoint
5     double CKTtime; //Tiempo de simulación durante TRAN Analysis
6     double CKTdelta; //Time-step actual
7     double CKTdeltaOld [7]
8     double CKTtemp; //Temperatura por defecto para los dispositivos
9     double CKTnomTemp; //Temperatura por defecto para los modelos
10    double CKTvt;
11    double CKTag [7];
12    int CKTorder; //Orden el método de integración
13    int CKTmaxOrder; //Máximo orden del método de integración
14    int CKTintegrateMethod; //Método de integración a utilizar
15    SMPmatrix *CKTmatrix; //Puntero a la Matriz dispersa del MNA
16    int CKTniState; //Estado interno
17    double *CKTrhs; //Re(Valor actual de la excitación RHS4 a cargar)5
18    double *CKTrhsOld; //Re(Valor previo del RHS para test de convergencia)

```

⁴RHS Right hand side o vector excitación del Análisis nodal Modificado

⁵RE(x) = Parte real del valor de x

```

19 | double *CKTrhsSpare; //Re(Valor RHS de reserva para reordenar)
20 | double *CKTirhs; //Im(Valor actual de la excitación RHS a cargar) 6
21 | double *CKTirhsOld; //IM(Valor previo del RHS para test de convergencia)
22 | double *CKTirhsSpare; //IM(Valor RHS de reserva para reordenar)
23 | double *CKTrhsOp; //Valores del Punto de Operación}
24 | double *CKTsenRhs; //RE(Valores actuales de la sensibilidad del RHS)
25 | double *CKTseniRhs; //IM(Valores actuales de la sensibilidad del RHS)
26 | int CKTmaxEqNum; //Número de la mayor ecuación definida
27 | int CKTcurrentAnalysis; //Análisis en ejecución (si es que hay alguno)
28 | CKTnode *CKTnodes; //1er Elemento de lista enlazada de nodos que describen las ecuaciones del
    | circuito
29 | CKTnode *CKTlastNode; //Último nodo de la lista enlazada anterior
30 | int CKTnumStates; //Tamaño del vector CKTstates que almacena datos pertimepoint
31 | long CKTmode; //Bit Vector que establece estado máquina para el control del análisis
32 | int CKTbypass; //Núm. Máx. de iteraciones para diferentes análisis
33 | int CKTdcMaxIter; // iteration limit for DC OP. (itl1)
34 | int CKTdcTrcvMaxIter; // Iteration limit for DC TRAN. curv (itl2)
35 | int CKTtranMaxIter; // Iteration limit for each timepoint for TRAN(itl4)
36 | int CKTbreakSize; //Tabla de Breakpoints y auxiliares
37 | int CKTbreak;
38 | double CKTsaveDelta;
39 | double CKTminBreak;
40 | double *CKTbreaks;
41 | double CKTabstol; //Diferentes tolerancias del circuito
42 | double CKTpivotAbsTol;
43 | double CKTpivotRelTol;
44 | double CKTreltol;
45 | double CKTchgtol;
46 | double CKTvoltTol;}
47 | double CKTgmin; //Mínima conductancia para dispositivos que crean conductancias nulas
48 | double CKTdelmin; //Mínimo Time-step permitido
49 | double CKTtrtol; //Para controlar el error de truncamiento
50 | double CKTfinalTime; //Parámetros del análisis DC TRAN
51 | double CKTstep;
52 | double CKTmaxStep;
53 | double CKTinitTime;
54 | double CKTomega; //Frecuencia del Análisis en AC
55 | double CKTsrcFact; //Necesario para las fuentes de tensión
56 | double CKTdiagGmin; //Parámetros de convergencia
57 | int CKTnumSrcSteps;
58 | int CKTnumGminSteps;
59 | int CKTnoncon;
60 | double CKTdefaultMosL; //Parámetros del MOSFET
61 | double CKTdefaultMosW;
62 | double CKTdefaultMosAD;
63 | double CKTdefaultMosAS;
64 | unsigned int CKThadNodeset:1;
65 | unsigned int CKTfixLimit:1; //Información MOSFET SPICE2
66 | unsigned int CKTnoOpIter:1;
67 | unsigned int CKTisSetup:1;
68 | JOB *CKTcurJob; //Puntero a la estructura JOB del análisis en realización
69 | SENstruct *CKTsenInfo; //Puntero a estructura que almacena datos del análisis de sensibilidad
    | en realización concurrente con el análisis actual.
70 | double *CKTtimePoints; //Lista de todos los timepoints aceptados en la simulación
    | transitoria actual
71 | double *CKTdeltaList; //Listado de todos los timepoints en la simulación transitoria actual
72 | int CKTtimeListSize; //Tamaños de las listas anteriores
73 | int CKTtimeIndex; //Posición actual en las listas anteriores
74 | int CKTsizeIncr; //Tamaño a incrementar los vectores anteriores si no hay espacio
75 | unsigned int CKTtryToCompact:1;
76 | unsigned int CKTbadMos3:1; //Utilizar las ecuaciones antigua del MOS3
77 | unsigned int CKTkeepOpInfo:1; //Flag para el análisis de pequeña señal
78 | int CKTtroubleNode; //Número del nodo con problemas de no convergencia
79 | GENinstance *CKTtroubleElt; //Instancia del dispositivo no convergente
80 | }
81 | CKTcircuit;

```

Listado 7.3 – Estructura *CKTcircuit*.

Como se puede observar, está compuesta por una agrupación de estructuras de datos y parámetros relativos a:

- Listado de nodos que almacenan las ecuaciones.
- Listado de dispositivos
- Características de los análisis a realizar
- Definiciones de parámetros de convergencia del análisis.
- Datos resultantes de la simulación

⁶IM(x) = Parte imaginaria del valor de x

La estructura de datos que contiene al circuito es de una complejidad importante. Ésta incluye a su vez a otras estructuras de datos dedicadas y optimizadas para realizar funciones específicas.

7.3.2.1.1 Estructura CKTnode

Se utiliza para describir una ecuación en el circuito. De manera general, un nodo corresponde a un nodo del circuito, pero las ecuaciones añadidas al incorporar las fuentes de tensión conllevan la creación de nuevas ecuaciones y por lo tanto estructuras *CKTnode*. El formalismo aplicado para la obtención de las ecuaciones se basa en el procedimiento de análisis nodal modificado *MNA* [CWRB75].

La estructura de datos *CKTnode* se representa por,

```

1  typedef struct sCKTnode {
2      IFuid name; //Nombre identificador del nodo dado por el int. usuario
3      int type; //Tipo de dato representado por la ecuación: Nodo V o I
4      int number; //Número de ecuación asignado a la ecuación en concreto
5      double ic; //Condición inicial para ese nodo dada por el int. usuario
6      double nodeset; //Tensión inicial proporcionada por la directive .NODESET
7      double *ptr; //Si las condiciones .NODESET y .IC se aplican a este nodo es un puntero a
           los campos numéricos de la matriz dispersa correspondiente a elemento diagonal (número,
           número)
8      struct sCKTnode *next; //Puntero al siguiente nodo de mayor número
9      unsigned int icGiven:1; //Flag para indicar si existe parámetro .IC7
10     unsigned int nsGiven:1; //Flag para indicar si existe parámetro .NODESET8
11 } CKTnode;

```

Listado 7.4 – Estructura *CKTnode*.

7.3.2.2 Estructura de datos de Análisis

Estas estructuras son utilizadas por SPICE3 para almacenar los datos de los análisis que van a ser realizados. Cada análisis específico se denomina trabajo (*JOB*) y tiene una estructura de datos propia. Todos los trabajos tienen un prefijo en su estructura de datos que las rutinas de más alto nivel pueden examinar. Ésto permite organizarlos en grupos que pueden ser ejecutados a la vez y que son llamadas tareas (*tasks*). Además, cada tipo de análisis dispone de una estructura de datos propia que la describe e identifica para las rutinas de más alto nivel.

7.3.2.2.1 Estructura SPICEanalysis

Proporciona información a las rutinas de más alto nivel sobre los análisis que *SPICE3* es capaz de realizar y su definición en lenguaje C es:

```

1  typedef struct {
2      IFanalysis public; //Interfaz de descripción de los parámetros del análisis del Int. Usuar.
           con el simulador
3      int size; //Tamaño en bytes de la estructura privada para almacenar los datos del análisis
4      int domain;
5      int do_ic;
6      int (*(setParm))(); //Puntero a la función que acepta valores de los parámetros y los
           introduce en la estructura de datos específica de dispositivos
7      int (*(askQuest))(); //Puntero a la función que accede a los valores en la estructura de
           datos específica de dispositivos y la exporta a aquellas rutinas que no tienen

```

⁷.IC Define condiciones iniciales V(NODNUM)=VAL V(NODNUM)=VAL ...

⁸.NODESET solo ayuda en el proceso de convergencia del análisis en DC, no afecta a la solución final del punto de operación

```

conocimiento de esta estructura de datos
8  int (*an_init)();
9  int (*an_func)();
10 } SPICEanalysis;

```

Listado 7.5 – Estructura *SPICEanalysis*.

7.3.2.2 Estructura JOB

Esta estructura de trabajo almacena los datos necesarios en un único análisis. Es una estructura que se incluye como primer elemento en cada estructura de datos de análisis específica. Esto permite a las rutinas normales manipular las estructuras de datos sin necesidad de conocer los detalles de cada tipo de estructura de datos de análisis concreta.

```

1 typedef struct sJOB {
2   int JOBtype; //Tipo del Trabajo
3   struct sJOB *JOBnextJob; // Próximo trabajo en la lista
4   IFuid JOBname; // Nombre del trabajo
5 } JOB;

```

Listado 7.6 – Estructura *JOB*.

7.3.2.3 Estructura TSKtask

Esta estructura define a un conjunto de análisis que pueden ser realizados a la vez. Dichos análisis podrán compartir elementos como los valores de las opciones introducidas mediante las directivas *.OPTIONS*⁹ permite modificar los parámetros del simulador para variar la precisión, velocidad, los parámetros por defecto de algunos dispositivos. La estructura de datos está formada por un conjunto de valores globales para todos los análisis a realizar y una lista enlazada a todos los trabajos o *JOBS* que forman parte de la tarea.

Los valores introducidos mediante la directiva *.OPTIONS* se almacenan como si de un análisis se tratara para de este modo presentar un interfaz más uniforme y conveniente al interfaz de usuario. Se enlaza con la lista enlazada de trabajos a realizar pero se trata como un caso especial. Solo puede existir un trabajo *.OPTIONS* en cada tarea y todos los valores que acompañan a esta directiva en el fichero fuente del circuito serán introducidos en un único trabajo. Estas opciones pueden ser tratadas fácilmente al comienzo de una tarea y por ésto, el análisis relativo *.OPTIONS* ha sido destacado e incluido directamente en la estructura *TSKtask*.

La estructura tiene la siguiente sintaxis en lenguaje C:

```

1 typedef struct \{
2   JOB taskOptions; //Estructura JOB para almacenar los datos .OPTIONS
3   JOB *jobs; //Lista enlazada de JOBS de todos los análisis a realizar como parte de esta
  tarea
4   char *TSKname; //Nombre de la tarea asignada por el Interfaz de Usuario
5   double TSKtemp; //Temperatura por defecto a utilizar por las instancias de los dispositivos
  de la tarea
6   double TSKnomTemp; //Temperatura por defecto a la caul se consideran medidos los pará
  metros de los modelos.
7   int TSKmaxOrder; //Máximo orden de integración permitido durante las simulaciones
  transitorias de la tarea
8   int TSKintegrateMethod; //Método de integración utilizado durante las simulaciones
  transitorias de la tarea
9   int TSKcurrentAnalysis; //Tipo actual de análisis en realización en la tarea.
10  #define DOING_DCOP 1
11  #define DOING_TRCV 2
12  #define DOING_AC 4

```

⁹.OPTIONS OPT1 OPT2 ... (or OPT=OPTVAL ...)



```

13 #define DOING_TRAN 8
14 int TSKbypass;
15 int TSKdcMaxIter; //Número máximo de iteaciones de Newton-Rapshon a realizar antes de
    abandonar un análisis de punto de operación (.OP) (it11)
16 int TSKdcTrcvMaxIter; //Número máximo de iteraciones de Newton-Rapshon a realizar antes de
    abandonar un análisis de función de transferencia en DC (.TF) (it12)
17 int TSKtranMaxIter; //Número máximo de iteraciones de Newton-Rapshon a realizar antes de
    abandonar un análisis de un timepoint en el análisis transitorio (.TRAN) (it14)
18 int TSKnumSrcSteps; //Número de puntos en el Source stepping
19 int TSKnumGminSteps; //Número de pasos en el Gmin stepping a tomar ambos en un intento de
    obtener el punto de operación (.OP)
20 double TSKminBreak;
21 double TSKabstol;
22 double TSKpivotAbsTol;
23 double TSKpivotRelTol;
24 double TSKreltol;
25 double TSKchgtol; //Tolerancias por tarea para se usadas en los campos del
26 double TSKvoltTol; //mismo nombre en la estructura de datos CKTcircuit durante el análisis
27 double TSKgmin; //Mínima conductancia configurable por el usuario para aquellos casos en
    que algunos modelos que presentan conductancia nula.
28 double TSKdelmin; //Mínimo timestep o incremento de tiempo permitido a utilizar antes de
    abandonar el análisis transitorio
29 double TSKtrtol; //Tolerancias por tarea para se usadas en los campos del mismo nombre en la
    estructura de datos CKTcircuit durante el análisis
30 double TSKdefaultMosL; //Valores por defecto de la Longitud, anchura, área de
31 double TSKdefaultMosW; // drenador y área de fuente para este circuito.
32 double TSKdefaultMosAD;
33 double TSKdefaultMosAS;
34 unsigned int TSKfixLimit:1; //Flag para elegir activación del algoritmo no simétrico de
    limitación de SPICE2 o el de SPICE3 de correcta simetría.
35 unsigned int TSKnoOpIter:1; //Flag para indicar la no realización de iteraciones en el aná
    lisis .OP e ir directamente al gmin step
36 unsigned int TSKtryToCompact:1; //Flag para las líneas LTRA
37 unsigned int TSKbadMos3:1; //Flag para los modelos MOS3
38 unsigned int TSKkeepOpInfo:1; //Flag para los análisis de pequeña señal
39 } TSKtask;

```

Listado 7.7 – Estructura TSKtask.

7.3.2.2.4 Estructura ACAN (.AC)

La estructura *ACAN* contiene los datos necesarios para controlar un análisis *AC*. Los tres primeros campos de la estructura son necesarios para combinarse con la estructura *JOB* estudiada en el apartado 7.3.2.2.2. Los campos restantes son específicos para el análisis *AC*.

```

1 typedef struct {
2     int JOBtype;
3     JOB *JOBnextJob; //Puntero hacia el siguiente trabajo a realizar
4     char *JOBname; //Nombre de este trabajo
5     double ACstartFreq; //Frecuencia inferior del análisis en AC
6     double ACstopFreq; //Frecuencia superior del análisis en AC
7     double ACfreqDelta; //Factor de incremento de frecuencia Hz/década/octava
8     double ACsaveFreq; //Si el análisis se interrumpió, esta es la frecuencia final completada,
    permite continuar desde donde se quedó
9     int ACstepType; //Tipo del factor de incremento:LINEAR/OCTAVE/DECADE
10    int ACnumberSteps; //Numero de pasos a calcular según los datos anteriores
11 } ACAN;

```

Listado 7.8 – Estructura ACAN.

7.3.2.2.5 Estructura OP (.OP)

La estructura *OP* soporta el cálculo del punto de operación y asigna un nombre para este análisis. Los tres primeros campos permiten su combinación con la lista enlazada de trabajos.

```

1 typedef struct {\
2     int JOBtype;
3     JOB *JOBnextJob;
4     char *JOBname;
5 } OP;

```

Listado 7.9 – Estructura OP.

7.3.2.2.6 Estructura PZAN (.PZ)

La estructura *PZAN* describe el análisis polo-cero a realizar. Como en casos anteriores, los tres primeros campos permiten su combinación con la lista enlazada de trabajos.

```

1 typedef struct {
2     int JOBtype;
3     JOB *JOBnextJob;
4     IFuid JOBname;
5     int PZin_pos; //Nodo de entrada positivo
6     int PZin_neg; //Nodo de entrada negativo
7     int PZout_pos; //Nodo de salida positivo
8     int PZout_neg; //Nodo de salida negativo
9     int PZinput_type;
10    int PZwhich;
11    int PZnumswaps; //Numero de intercambios de filas y columnas realizados en la descomposi
    ción de la matriz
12    int PZbalance_col;
13    int PZsolution_col;
14    PZtrial *PZpoleList; //Puntero al principio de la lista enlazada de los polos encontrados
    en el circuito
15    PZtrial *PZzeroList; //Puntero al principio de la lista enlazada de los ceros encontrados
    en el circuito
16    int PZnPoles; //Número de Polos
17    int PZnZeros; //Número de Ceros
18    double *PZdrive_pptr;
19    double *PZdrive_nptr;
20 } PZAN;
```

Listado 7.10 – Estructura PZAN.

7.3.2.2.7 Estructura TFan (.TF)

La estructura *TFan* describe el análisis de función de transferencia a realizar. Como en casos anteriores, los tres primeros campos permiten su combinación con la lista enlazada de trabajos.

```

1 typedef struct \{
2     int JOBtype;
3     JOB *JOBnextJob;
4     IFuid JOBname;
5     CKTnode *TFoutPos; //Nodo positivo de la tensión de salida
6     CKTnode *TFoutNeg; //Nodo negativo de la tensión de salida
7     IFuid TFoutSrc; //Nombre identificador de la fuente de salida si la variable de salida es
    la corriente que circula por una fuente
8     IFuid TFinSrc; //Nombre identificador de la fuente de entrada tanto si es una fuente de
    tensión o corriente.
9     char *TFoutName; //Nombre imprimible para una salida del formato v(x,y)
10    unsigned int TFoutIsV :1; //Flags para indicar si la señal de salida es una tensión
11    unsigned int TFoutIsI :1; //entre dos nodos o una corriente a través de una fuente de
    tensión.Si ninguno de los flags están activados, no se ha definido salida alguna lo que
    genera error.
12    unsigned int TFinIsV :1; //Flags que indican si la señal de entrada es una fuente
13    unsigned int TFinIsI :1; //de tensión o de corriente. Son mutuamente exclusivos.
14 } TFan;
```

Listado 7.11 – Estructura TFa.

7.3.2.2.8 Estructura TRANan (.TRAN)

La estructura *TRANan* contiene todos los datos para controlar al análisis transitorio que describe. Como en casos anteriores, también, los tres primeros campos permiten su combinación con la lista enlazada de trabajos (*JOBS*).

```

1 typedef struct \{
2     int JOBtype;
3     JOB *JOBnextJob;
4     char *JOBname;
5     double TRANfinalTime; //Tiempo final hasta el que deberá realizarse el análisis
6     double TRANstep; //Incremento de tiempo sugerido por el usuario
7     double TRANmaxStep; //Máximo incremento de tiempo interno definido por el usuario
8     double TRANinitTime; //Tiempo inicial desde el que deben almacenarse los datos de salida
9     long TRANmode; //Variable LONG que debe tener el flag UIC activo para indicar que las
```

```

condiciones iniciales suministradas en la especificación deben sustituir las obtenidas
del análisis del punto de operación, antes de comenzar el análisis transitorio
10 #define TRAN_TSTART 1
11 #define TRAN_TSTOP 2
12 #define TRAN_TSTEP 3
13 #define TRAN_TMAX 4
14 #define TRAN_UIC 5
15 GENERIC * TRANplot; //Las estructura de representación devuelta por la función
    OUTpBeginPlot necesita ser almacenada, en caso de evento, el análisis puede ser parado o
    abortado.
16 } TRANan;

```

Listado 7.12 – Estructura *TRANan*.

7.3.2.2.9 Estructura TRCV (.SENS)

La estructura *TRCV* almacena todos los datos para controlar al análisis de sensibilidad. Este tipo de análisis permite caracterizar la variación de la tensión o corriente del circuito en función de las variaciones de los parámetros de éste. Como en casos anteriores, también, los tres primeros campos permiten su combinación con la lista enlazada de trabajos (*JOBS*).

```

1 | typedef struct \{
2 |     int JOBtype;
3 |     JOB *JOBnextJob;
4 |     char *JOBname;
5 |     double TRCVvStart[TRCVNESTLEVEL]; //Almacena la tensión-corriente inicial para cada una de
6 |     double TRCVvStop[TRCVNESTLEVEL]; //Almacena la tensión-corriente final para cada una de las
7 |     double TRCVvStep[TRCVNESTLEVEL]; //Almacena el incremento a añadir al valor de las fuentes
8 |     double TRCVvSave[TRCVNESTLEVEL]; //Este vector almacena los valores originales del punto de
9 |     int TRCVgSave[TRCVNESTLEVEL]; //Flag que indica que se ha almacenado el valor en DC en el
10 |     IFuid TRCVvName[TRCVNESTLEVEL]; //Vector que contiene los nombres de las fuentes a barrer
11 |     GENinstance *TRCVvElt[TRCVNESTLEVEL]; //Puntero a la estructura GENinstance de las fuentes
12 |     int TRCVvType[TRCVNESTLEVEL]; //Tipo de cada elemento
13 |     int TRCVset[TRCVNESTLEVEL]; //Vector de flags para indicar al código del análisis de funció
14 |     int TRCVnestLevel; //Mayor nivel de anidamiento que se ha usado en el análisis
15 |     int TRCVnestState; //Nivel de anidamiento de la tensión o corriente que ha cambiado más
16 | } TRCV;

```

Listado 7.13 – Estructura *TRCV*.

7.3.2.3 Estructuras de datos de la Matriz Dispersa.

Estas estructuras son usadas por la librería de matrices dispersas para manejar en cualquier parte del código la matriz del circuito objeto de la simulación, y son de acceso privado para la librería de matrices dispersas. Otras partes del simulador SPICE3 acceden mediante punteros a estas estructuras, pero el uso global de la estructura está restringido a la librería, por lo tanto los cambios de la estructura solo tendrán efecto para la librería.

7.3.2.3.1 Estructura SMPmatrix - MatrixFrame

La estructura *SMPmatrix* original de SPICE2 ha sido mejorada en SPICE3 recibiendo la nueva denominación de *MatrixFrame*. La versión actual que se presenta a continuación es la definitiva. El formato exterior de ésta se ha mantenido sin cambios para poder asegurar

la compatibilidad con las estructuras que hacen referencia a ésta.

```

1 typedef char SMPmatrix; //La estructura está soportada por un puntero a la Matriz dispersa
  representada por MatrixFrame representado por una variable carácter prototipada.
2 struct MatrixElement
3 {
4     RealNumber Real; //Valor real del elemento
5     RealNumber Imag; //Valor imaginario del elemento
6     int Row; //Número de Fila del elemento
7     int Col; //Número de Columna del elemento
8     struct MatrixElement *NextInRow; //Puntero al siguiente elemento en la fila
9     struct MatrixElement *NextInCol; //Puntero al siguiente elemento en la columna
10    char *pInitInfo;
11 };
12 typedef struct MatrixElement *SMPelement;
13 typedef struct MatrixFrame *MatrixPtr; //Puntero a la Matriz dispersa

```

Listado 7.14 – Estructura *SMPmatrix*.

La estructura *SMPmatrix* se ha reconvertido en esta nueva definición en la estructura de datos *MatrixFrame* que incluye gran cantidad de datos auxiliares relativos al estado de la matriz dispersa. Debido a la importancia de esta nueva estructura abstracta de datos se va a presentar su definición completa. La estructura contiene todos los punteros a las listas enlazadas que contienen los elementos de la matriz. Además se incluyen otros parámetros y punteros que son de uso global por las rutinas de tratamiento de la matriz dispersa.

```

1 struct MatrixFrame {
2     RealNumber AbsThreshold; //Valor absoluto que un elemento debe tener para ser considerado
  un candidato a pivote.
3     int AllocatedSize; //Tamaño asignado de la matriz actual. El tamaño de la matriz puede
  crecer cuando el parámetro EXPANDABLE está activo (TRUE) y AllocatedSize es el mayor tamaño
  que la matriz puede tomar sin que la matriz necesite ser reasignada.
4     int AllocatedExtSize; //Tamaño asignado del vector usado para traducir un fila externa y
  los números de columnas en sus valores internos
5     BOOLEAN Complex; //Flag que indica cuando la matriz es compleja (TRUE) o real
6     int CurrentSize; //Este valor es usado durante la construcción de la matriz cuando el pará
  metro TRANSLATE está activo (TRUE) e indica el número de filas y columnas internas que
  tienen los elementos
7     ArrayOfElementPtrs Diag; //Vector de punteros que direccionan a los elementos de la
  diagonal
8     BOOLEAN *DoCmplxDirect; //Vector de flags, uno por cada columna de la matriz. Si un
  flag está activado (TRUE) entonces la columna correspondiente en una matriz compleja ser
  á eliminada en la función spFactor() utilizando direccionamiento directo en lugar del
  indirecto.
9     BOOLEAN *DoRealDirect; //Vector de flags, uno por cada columna de la matriz. Si un flag
  está activado (TRUE) entonces la columna correspondiente en una matriz real será
  eliminada en la función spFactor() utilizando direccionamiento directo en lugar del
  indirecto.
10    int Elements; //Número original de elementos (número total de elementos menos fill-in de
  la matriz.
11    int Error; //Estado de error de la librería de matrices dispersas.
12    int ExtSize; //Máximo valor del número de columnas o filas externas
13    int *ExtToIntColMap; //Vector que es usado para convertir números de columnas externas
  en internas. Está presente sólo si la opción TRANSLATE está activada (TRUE)
14    int *ExtToIntRowMap; //Vector que es usado para convertir números de filas externas en
  internas. Está presente sólo si la opción TRANSLATE está activada (TRUE)
15    BOOLEAN Factored; //Indica si la matriz ha sido factorizada. Está activo (TRUE) durante
  spFactor() y spOrderAndFactor() y (FALSE) en spCreate() y spClear()
16    int Fillins; //Número de fill-in creados durante la factorización de la matriz
17    ArrayOfElementPtrs FirstInCol; //Array de punteros que direccionan al primer elemento no
  nulo de la columna correspondiente en el índice
18    ArrayOfElementPtrs FirstInRow; //Array de punteros que direccionan al primer elemento no
  nulo de la fila correspondiente en el índice
19    unsigned long ID; //Identificador que distingue a la estructura de datos de la matriz
  dispersa. Cuando el parámetro de depuración DEBUG está activo (TRUE), todas las funciones
  externas disponibles de matrices dispersas chequean este identificador para asegurar que
  están trabajando con una matriz válida
20    RealVector Intermediate; //Variable de almacenamiento temporal utilizada por la rutina
  spSolve(). Se utiliza durante la sustitución ascendente y descendente Ax=b > Ly=b y Ux=y}
21    BOOLEAN InternalVectorsAllocated; //Flag que indica si los vectores de Markowitz e
  intermedios han sido creados. Se crean en la rutina spCreateInternalVectors()
22    int *IntToExtColMap; //Vector usado para convertir los números de columna internos en
  los números de columna externos
23    int *IntToExtRowMap; //Vector usado para convertir los números de columna internos en los
  números de columna externos
24    int *MarkowitzRow; //Vector que contiene la lista de los elementos no nulos sin contar
  los pivotes de cada fila. Se usa para generara y actualizar el vector MarkowitzProd
25    int *MarkowitzCol; //Vector que contiene la lista de los elementos no nulos sin contar
  los pivotes de cada columna. Se usa para generara y actualizar el vector MarkowitzProd
26    long *MarkowitzProd; //Vector que contiene los productos de las filas y columnas
  Markowitz. El elemento con el menor producto es el mejor pivote para mantener la dispersió
  n

```

A


```

27  int MaxRowCountInLowerTri; //Máximo número de elementos no diagonales en la fila de la
    matriz triangular inferior L. Este valor es utilizado en el cálculo de una estimación del
    error de redondeo de la matriz
28  BOOLEAN NeedsOrdering; //Flag que indica que la matriz necesita ser ordenada o reordenada
    . Es activo (TRUE) e spCreate() y spGetElement() o spGetAdmittance() si se añaden nuevos
    elementos a la matriz después de haber sido factorizada. Toma valor (FALSE) en
    spOrderAndFactor()
29  BOOLEAN NumberOfInterchangesIsOdd; //Flag que indica si el valor resultante de la suma
    de los intercambios de filas y columna es par. Usado para determinar el signo del
    determinante
30  BOOLEAN Partitioned; //Flag que indica que las columnas de la matriz han sido divididas
    en dos grupos. Aquellas que serán accedidas directamente e indirectamente en spFactor()
31  int PivotsOriginalCol; //Columna original de la cual se eligió el pivote de columna
32  int PivotsOriginalRow; //Fila original de la cual se eligió el pivote de fila
33  char PivotSelectionMethod; //Variable tipo carácter que indica el método de éxito para
    buscar al pivote
34  BOOLEAN PreviousMatrixWasComplex; //Flag necesario para conocer el procedimiento de
    borrado de la matriz. Cuando se trata con matrices reales es importante que los términos
    imaginarios de los elementos de la matriz sean nulos. Por lo tanto, si la matriz previa
    era compleja, entonces la matriz actual será borrada como si fuera compleja aunque fuera
    real.
35  RealNumber RelThreshold; //El valor que un elemento debe tener con respecto de otro en su
    fila para ser considerado un candidato a pivote
36  BOOLEAN Reordered; //Flag que indica que la matriz ha sido reordenada. Es desactiva en
    spCreate() y se activa en spMNA_Preorder() y spOrderAndFactor() y es usada en spPrint()
37  BOOLEAN RowsLinked; //Flag que indica la existencia de los punteros a filas. Las rutinas
    AddByIndex() no generan dichos punteros que son necesarios para otras rutinas como
    spOrderAndFactor() y spScale(). Los punteros de filas son generados por spLinkRows()
38  int SingularCol; //Valor normalmente a cero, pero si la matriz es singular, contiene el n
    úmero de columna exterior del pivote que es nulo
39  int SingularRow; //Valor normalmente a cero, pero si la matriz es singular, contiene el nú
    mero de fila exterior del pivote que es nulo
40  int Singletons; //Número de singletons disponibles para realizar el pivoteo. Si la fila I
    y la columna I contienen singletons solo se cuenta uno de ellos.
41  int Size; //Número de filas y columnas en la matriz. No cambia si la matriz es reordenada
42  struct MatrixElement TrashCan; //Es un MatrixElement vacío que es usado por el usuario
    para introducir elementos relativos a las filas o columnas cero. Cuando el usuario añade
    un elemento en la fila o columna cero, la matriz devuelve un puntero a TrashCan
43
44 // Campos relativos a la Asignación de Memoria
45
46  AllocationListPtr TopOfAllocationList; //Puntero que direcciona a la entrada superior de
    una lista. La lista contiene todos los punteros a los segmentos de memoria que han sido //
    asignados a esta matriz. Esto es usado cuando la memoria debe ser liberada en la eliminaci
    ón de la matriz
47  int RecordsRemaining; //Número de huecos libres en la lista de asignaciones
48  ElementPtr NextAvailElement; //Puntero al próximo elemento disponible que ha sido
    asignado en memoria pero que no ha sido aún usado. Los elementos de la matriz son
    asignados en grupos de ELEMENTS_PER_ALLOCATION de manera que permitan acelerar dicha
    asignación y la //eliminación.
49  int ElementsRemaining; //Número de elementos sin utilizar en el último bloque de
    elementos asignados.
50  ElementPtr NextAvailFillin; //Puntero al siguiente fill-in que ha sido asignado pero
    permanece todavía sin usar. Fills-in son asignados en grupos para mantenerlos cercanos fí
    sicamente en la memoria al margen de la matriz.
51  int FillinsRemaining; //Número de fills-in sin usar en el último grupo de fills-in
    asignados.
52  struct FillinListNodeStruct *FirstFillinListNode; //Puntero que direcciona al comienzo de
    la lista enlazada que soporta la lista de fill-ins
53  struct FillinListNodeStruct *LastFillinListNode; //Puntero que direcciona al final de la
    lista enlazada que soporta la lista de fill-ins
54 };

```

Listado 7.15 – Estructura *MatrixFrame*.

7.3.2.4 Estructuras de datos del Analizador de Entrada (Input parser)

Estas estructuras son usadas por el analizador de entrada para mantener almacenar los datos del circuito actual bajo análisis.

7.3.2.4.1 Estructura *INPtab*

La estructura *INPtab* contiene todos los nombres definidos en el interfaz de usuario excluyendo los nombres de ecuaciones. Es equivalente a una tabla *Hash* hecha de listas enlazadas a estructuras similares.

```
1 | struct INPtab {
```

```

2 char *t_ent; //Cadena de caracteres que almacena el nombre del este elemento de la
  estructura INPtab
3 struct INPtab *t_next; //Puntero al siguiente elemento de loa lista enlazada
4 \};

```

Listado 7.16 – Estructura *INPtab*.

7.3.2.4.2 Estructura *INPntab*

La estructura *INPntab* es casi idéntica a *INPtab* descrita anteriormente, pero es usada para los nombres de las ecuaciones del circuito como nombres de nodo o nombres de ecuaciones de corriente. Esta estructura posee un campo adicional.

```

1 struct INPntab {
2 char *t_ent; //Cadena de caracteres que almacena el nombre del este elemento de la
  estructura INPntab
3 GENERIC* t_node; //Puntero a la estructura interna del simulador correspondiente a este
  nombre
4 struct INPntab *t_next; //Puntero al siguiente elemento de loa lista enlazada
5 };

```

Listado 7.17 – Estructura *INPntab*.

7.3.2.4.3 Estructura *INPtable*

La estructura *INPtable* es la estructura global del analizador de entrada. Un elemento de este tipo de estructura de datos es asignado para cada circuito a analizar y es utilizado por las rutinas del analizador de entrada.

Muchos de los campos de la estructura son de la forma *defXmod*. Cada uno de estos campos apunta a la estructura interna de un modelo de [SPICE3](#) que representa el modelo por defecto a usar para el dispositivo *X*.

```

1 typedef struct sINPtables{
2 struct INPtab **INPsymtab; //Tabla hash para los nombres excluyendo a los de ecuaciones.
  Es un vector reservado de punteros al comienzo de las cadenas.
3 struct INPnTab **INPtermsymtab; //Tabla hash para los nombres de ecuaciones y terminales
  . Es un vector reservado de punteros al comienzo de las cadenas
4 int INPsize; //Tamaño del vector de punteros INPsymtab
5 int INPtermsize; //Tamaño del vector de punteros INPtermsymtab
6 GENERIC *defAmod; //Referencia sin usar
7 GENERIC *defBmod; //Referencia a modelo de una fuente cualquiera
8 GENERIC *defCmod; //Referencia a modelo de condensador
9 GENERIC *defDmod; //Referencia a modelo de diodo
10 GENERIC *defEmod; //Referencia a modelo de Fuente de tensión controlada por tensión. VCVS
11 GENERIC *defFmod; //Referencia a modelo de Fuente de corriente controlada por corriente.
  CCCS
12 GENERIC *defGmod; //Referencia a modelo de Fuente de corriente controlada por tensión.
  VCCS
13 GENERIC *defHmod; //Referencia a modelo de Fuente de tensión controlada por corriente
  CCVS
14 GENERIC *defImod; //Referencia a modelo de Fuente independiente de corriente
15 GENERIC *defJmod; //Referencia a modelo de JFET
16 GENERIC *defKmod; //Referencia a modelo de inductor mutuo
17 GENERIC *defLmod; //Referencia a modelo de inductor
18 GENERIC *defMmod; //Referencia a modelo de MOSFET
19 GENERIC *defNmod; //Referencia sin utilizar
20 GENERIC *defOmod; //Referencia sin utilizar
21 GENERIC *defPmod; //Referencia sin utilizar
22 GENERIC *defQmod; //Referencia a modelo de BJT
23 GENERIC *defRmod; //Referencia a modelo de resistencia
24 GENERIC *defSmod; //Referencia a modelo de Interruptor controlado por //tensión
25 GENERIC *defTmod; //Referencia a modelo de línea de transmisión
26 GENERIC *defUmod; //Referencia a modelo de línea RC
27 GENERIC *defVmod; //Referencia a modelo de Fuente independiente de tensión
28 GENERIC *defWmod; //Referencia a modelo de Interruptor controlado por //corriente
29 GENERIC *defYmod; //Referencia a sin usar
30 GENERIC *defZmod; //Referencia a modelo de MESFET}
31 } INPtables;

```

Listado 7.18 – Estructura *INPtable*.

7.3.2.4.4 Estructura CARD

La estructura *CARD* que sirve para almacenar una copia de una línea del fichero de entrada y todos los datos asociados con ella durante el análisis. Las líneas del fichero de entrada son leídas y separadas en líneas físicas y lógicas. Las líneas físicas son delimitadas por los caracteres de línea nueva y su final. Las lógicas son el resultado de concatenar las líneas físicas para formar una de contenido completo.

```

1| typedef struct card{
2|     int linenum; //Número de línea asociado a la línea física que poseía el primer carácter de
   |     la línea lógica.
3|     char *line; //Línea lógica. Es la línea exacta si es una única línea lógica que fue leída
   |     como una única línea física. Si la línea lógica contiene varias líneas físicas marcadas con
   |     el carácter de continuación en la primera columna, representa la línea lógica formada de
   |     la concatenación de las líneas físicas sustituyendo el carácter de continuación por el
   |     espacio
4|     char *error; //Mensaje de error asociado con la línea lógica.
5|     struct card *nextcard; //Puntero a la siguiente estructura que contiene la próxima línea
   |     lógica
6|     struct card *actualLine; //Si este puntero no es NULL, la línea lógica está formada por
   |     varias líneas físicas. Este puntero direccionará a una lista enlazada de las líneas fí-
   |     sicas que forman la lógica. Cada elemento de esta lista tiene estructura CARD pero en el
   |     campo LINE siempre será una línea física nunca una lógica y NEXTCARD apuntará a la línea
   |     física siguiente de la lógica. Y ACTUALLINE siempre será NULL
7| } card;

```

Listado 7.19 – Estructura *CARD*.

7.3.2.4.5 Estructura INPmodel

La estructura *INPmodel* es utilizada por el analizador de entrada para almacenar toda la información relativa a los modelos que aparecen en el fichero fuente del circuito. Durante el primer recorrido del análisis del fichero las líneas conteniendo el parámetro “.MODEL” son almacenadas en una lista enlazada de estructuras *CARD* y durante el segundo recorrido del fichero, estas líneas son analizadas y los punteros a los modelos creados son almacenados en la estructura para un más rápido acceso. El principio de la listas de estructuras *INPmodel* es almacenada en una variable global cosa que limita al analizador a un circuito por análisis.

```

1| typedef struct sINPmodel{ //Estructura para almacenar los modelos después de ser leídos en la
   |     primera pasada del analizador de entrada
2|     IFuid INPmodName; //Identificador del Modelo
3|     int INPmodType; //Número del Tipo dispositivo
4|     struct sINPmodel *INPnextModel; //Puntero al próximo modelo de la lista
5|     int INPmodUsed; //Flag que indica que ya ha sido usado
6|     card *INPmodLine; //Puntero a la línea que describe el modelo
7|     GENERIC *INPmodfast; //Puntero de acceso rápido al modelo
8| } INPmodel;

```

Listado 7.20 – Estructura *INPmodel*.

7.3.2.4.6 Estructura INPparseTree

La estructura *INPparseTree* contiene los datos del árbol de análisis que el interfaz de usuario ha leído y que será posteriormente evaluado por el simulador. El árbol de análisis encapsula la estructura estándar definida por el interfaz de usuario añadiendo algunos campos para uso interno del analizador.

```

1| typedef struct INPparseTree {
2|     IFparseTree p; //Estructura del árbol de análisis requerida por el interfaz de usuario
3|     struct INPparseNode *tree; //Estructura de árbol para la ecuación proporcionada por el
   |     usuario
4|     struct INPparseNode **derivs; //Vector de punteros asignado hacia las estructuras
   |     internas de árbol de análisis que representa las derivadas parciales del árbol de aná-
   |     lisis original respecto a todas las variables usadas en la expresión.
5| } INPparseTree;

```

Listado 7.21 – Estructura *INPparseTree*.

7.3.2.4.7 Estructura INPparseNode

La estructura *INPparseNode* se utiliza para construir los árboles de análisis. Estas estructuras son utilizadas para construir el árbol en que cada nodo puede representar una variable o constante, o un operador con uno o dos operandos que son a su vez estructuras *INPparseNode*.

```

1 typedef struct INPparseNode {
2     int type; //Tipo de nodo representado por el elemento +,-,x, constante, llamada a funció
3     n.
4     struct INPparseNode *left; //Si es un operador de dos nodos, representa el operador de la
5     izquierda
6     del operador del nodo. Si es un operador único representa a éste mismo y NULL si no es un
7     operador
8     struct INPparseNode *right; //Si es un operador de dos operandos representa al operando
9     de la derecha y en caso contrario NULL
10    double constant; //Si es una constante, representa su valor numérico
11    int valueIndex; //Índice de la variable situada en este punto del árbol. Este índice
12    representa la posición en el vector de los nombres de variables en la estructura
13    IFparseTree dentro de INPparseTree
14    char *funcname; //Nombre de la función llamada desde este nodo
15    int funcnum; //Número de la función en una tabla interna. Es usado para procesar más rá
16    pidamente la función utilizando las cláusulas CASE en vez de comparación de cadenas.
17    double (*function)(); //Puntero a la función
18 } INPparseNode;

```

Listado 7.22 – Estructura *INPparseNode*.

7.3.2.4.8 Estructura PTelement

La estructura *PTelement* es utilizada para almacenar temporalmente los datos introducidos durante el análisis sintáctico en un árbol de análisis. Son introducidas en una pila de análisis durante el proceso de análisis sintáctico y son sacadas de la pila conforme las operaciones van consumiéndolas.

```

1 typedef struct PTelement {
2     int token; //Tipo de representación o token léxico que el analizador ha encontrado. Puede
3     ser un operador, el final de la entrada, un paréntesis o un valor
4     int type; //Si el token es un valor entonces indica el tipo de valor, podrá ser: TYP_NUM
5     para indicar un número, TYP_STRING para indicar una cadena de caracteres, TYP_NODE para
6     indicar un nodo del árbol de análisis. El analizador léxico no generará TYP_PNODE, pero
7     el analizador lo generará internamente y lo meterá en la pila.
8     union{
9         char *string; //Cadena de caracteres reconocida como token
10        double real; //Número real encontrado por el analizador léxico
11        INPparseNode *pnode; //Estructura de nodo interno del árbol de análisis resultante de
12        la combinación de los elementos analizados previamente
13    } value; //Unión usada para mantener el valor del token. Dependiendo del tipo de
14    valor se rellenará uno de los tres campos
15 } PTelement ;

```

Listado 7.23 – Estructura *PTelement*.

7.4 Creación y modificación de modelos en código fuente.

En el apartado 7.3.1 se establece el procedimiento para crear un nuevo dispositivo. Para que el simulador pueda procesar las referencias a modelos y a elementos pertenecientes al nuevo modelo compacto codificado, es necesario modificar las rutinas existentes que realizan las tareas de análisis sintáctico de la entrada de datos tanto por fichero como por consola.

A modo de resumen se presentan algunas modificaciones necesarias para la integración de nuevos modelos.

- **INPdomodel.c:** fichero existente dentro del directorio SRC\SPICELIB\PARSER \que vincula el tipo de modelo que puede aparecer en una línea de texto del fichero de descripción topológica (.MODEL) al nombre interno del dispositivo y realiza una posterior conversión usando la función CKTtypelook al número asociado al tipo de dispositivo. Si el dispositivo no presenta información asociada del tipo (.MODEL) esta rutina no necesita actualización. El contenido de esta rutina es simplemente una comparación secuencial de la cadena asociada al tipo de modelo existente en la etiqueta .MODEL con las cadenas de texto de los diferentes modelos disponibles en el simulador.

```

1 // ----- Check if model is a BJT -----
2 if ((strcmp(typename, "npn") == 0) || (strcmp(typename, "pnp") == 0)) {
3   err = INPfindLev(line, &lev);
4   switch(lev) {
5     case 0:
6     case 1:
7       type = INPtypelook("BJT");
8       if (type < 0) {
9         err = INPmkTemp(
10          "Device type BJT not available in this binary\n");
11       }
12       break;
13     case 2:
14       type = INPtypelook("BJT2");
15       if (type < 0) {
16         err = INPmkTemp(
17          "Device type BJT2 not available in this binary\n");
18       }
19       break;
20     case 4:
21       type = INPtypelook("VBIC");
22       if (type < 0) {
23         err = INPmkTemp(
24          "Device type VBIC not available in this binary\n");
25       }
26       break;
27     default: /* placeholder; use level 4 for the next model */
28       err = INPmkTemp(
29        "Only BJT levels 1-2, 4 are supported in this binary\n");
30       break;
31   }
32   INPmakeMod(modname, type, image);
33 } /* end if ((strcmp(typename, "npn") == 0) || (strcmp(typename, "pnp") == 0)) */
34 /* ----- Check if model is a diode ----- */
35 else if (strcmp(typename, "d") == 0) {
36   type = INPtypelook("Diode");
37   if (type < 0) {
38     err =
39     INPmkTemp
40     ("Device type Diode not available in this binary\n");
41   }
42   INPmakeMod(modname, type, image);
43 } /* else if (strcmp(typename, "d") == 0) { */
44

```

Listado 7.24 – Detección del modelo compacto utilizado.

- **INPfindLev.c:** se utiliza para distinguir las diferentes posibilidades existentes en los modelos compactos de los dispositivos MOSFET. Los modelos MOSFET se diferencian por el nivel (level) que acompaña a .MODEL en la definición.

```

1 .model nmos_sgt nmos level=10 // Model: SGT MOSFET
2 .model N1 NMOS Level=8 Tnom=27.0 Nch= 2.498E+17 Tox=9E-09 Xj=1.00000E-07 // Model:
BSIM3v3
3 .model P1 PMOS level=14 version=4.4.0 biunit=1 paramchk=1 mobmod=0 // Model: BSIM4.4.0
4 .model MODJ NJF level=1 vto=-3.5 beta=4.1E-4 lambda=0.002 rd=200

```

Listado 7.25 – Detección del nivel del modelo compacto utilizado.

- **INP_numnodes.c:** Esta rutina es utilizada para obtener a partir del nombre del elemento el número máximo de nodos que pudiera tener. Permite el uso de dispositivos con un número diferente de nodos.

```

1 int inp_numnodes(char c)
2 {
3     if (isupper(c))
4         c = tolower(c);
5     ...
6     case 'E':
7     case 'e': return (2);
8     case 'F':
9     case 'f': return (2);
10    case 'G':
11    case 'g': return (2);
12    case 'H':
13    case 'h': return (2);
14    ...
15    case 'M': // En caso de comenzar por M ó m el nombre del elemento
16    case 'm': return (7); // devuelve 7 como número máximo de nodos para un MOSFET
17    case 'N':
18    case 'n': return (3); // Variable Resistor
19    ...

```

Listado 7.26 – Obtención del número máximo de nodos disponibles en un modelo compacto utilizado.

- **INPpas2.c:** La detección del carácter inicial de cada elemento en el fichero de *netlist* permite procesar sus nodos y parámetros asociados. Cada elemento circuital es procesado por una función diferente de modo que se facilita el proceso de añadido de un nuevo modelo compacto.

```

1 void INPpas2(void *ckt, card * data, INPtables * tab, void *task)
2 {
3     ...
4     for (current = data; current != NULL; current = current->nextcard){
5         c = *(current->line);
6         c = islower(c) ? toupper(c) : c;
7
8         switch (c) {
9             ...
10            case 'r':
11            case 'R':
12                /* Rname <node> <node> [<val>][<mname>][w=<val>][l=<val>] */
13                INP2R(ckt, tab, current);
14                break;
15            case 'c':
16            case 'C':
17                /* Cname <node> <node> <val> [IC=<val>] */
18                INP2C(ckt, tab, current);
19                break;
20            ...
21            case 'g':
22            case 'G':
23                /* Gname <node> <node> <node> <node> <val> */
24                INP2G(ckt, tab, current);
25                break;
26            case 'e':
27            case 'E':
28                /* Ename <node> <node> <node> <node> <val> */
29                INP2E(ckt, tab, current);
30                break;
31            case 'f':
32            case 'F':
33                /* Fname <node> <node> <vname> <val> */
34                INP2F(ckt, tab, current);
35                break;
36            case 'h':
37            case 'H':
38                /* Hname <node> <node> <vname> <val> */
39                INP2H(ckt, tab, current);
40                break;
41            case 'd':
42            case 'D':
43                /* Dname <node> <node> <model> [<val>] [OFF] [IC=<val>] */
44                INP2D(ckt, tab, current);
45                break;
46            ...
47            case 'm':
48            case 'M':
49                /* Mname <+_node> <center_node> <-_node> <node> <model> [resistance=<val>]
50                [cursor=<val>] */
51                INP2M(ckt, tab, current);
52                break;

```

Listado 7.27 – Interpretación de los datos asociados a un elemento.

7.5 Resultados del Test de verificación.

A continuación se presentan los diferentes circuitos de prueba utilizados para verificar el correcto funcionamiento de cada uno de los tipos de análisis con diferentes circuitos. El procedimiento de verificación comienza con un circuito que utiliza un tipo de análisis, la simulación y la salida de los resultados obtenidos. Los resultados han sido comparados exitosamente con otros simuladores comerciales.

7.5.1 Test de Análisis de Punto de Operación (.OP)

El circuito elegido para comprobar el funcionamiento del análisis de punto de operación utiliza un transistor bipolar en configuración de emisor común, ver en la figura 7.23. La

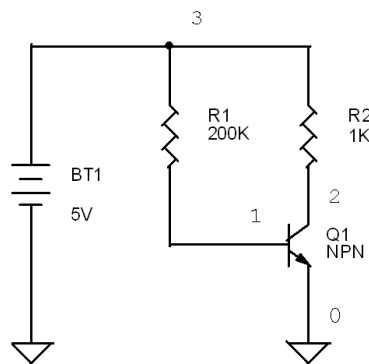


Figura 7.23 – Circuito de test para punto de operación.

descripción topológica dicho circuito y el listado de comandos a ejecutar se muestran en el listado 7.28.

```

1 | Title: One-Transistor Circuit
2 | Q1 2 1 0 QMOD
3 | R1 2 3 1K
4 | R2 1 3 200K
5 | VCC 3 0 5
6 | .MODEL QMOD NPN
7 | .control
8 | *VOLCADO FICHERO.CIR ANALIZADO SINTACTICAMENTE
9 | listing > bjt.lst
10 | *ANALISIS PUNTO DE OPERACION}
11 | op
12 | rusage everything > op.ope
13 | print V(2) V(1) > op.txt
14 | display > op.var
15 | .endc
16 | .end

```

Listado 7.28 – Netlist del circuito de test para análisis en punto de operación.

Las variables disponibles en el circuito se han almacenado en el fichero *op.var*, cuyo contenido se muestra a continuación:

```

Title:One-Transistor Circuit
Name: op1 (Operating Point)
Date: Wed Sep 03 13:37:35 2010
V(1) : voltage, real, 1 long
V(2) : voltage, real, 1 long [default scale]
V(3) : voltage, real, 1 long
vcc#branch : current, real, 1 long

```

El resultado de la simulación se ha volcado al fichero de texto *op.txt* donde se almacena el valor de cada una de las variables del circuito una vez alcanzado el punto de operación. Los

valores obtenidos tras la simulación son:

```
v(2) = 2.896719e+00
v(1) = 7.934391e-01
```

Se puede realizar un volcado de la información interna de la aplicación sobre el desarrollo temporal de la simulación para ver la distribución de las diferentes fases del análisis:

```
Total elapsed time: 0.000 seconds.
Elapsed time since last call: 0.000 seconds.
Current dynamic memory usage = 596 KB,
Dynamic memory limit = 709828 KB.
Nominal temperature = 27
Operating temperature = 27
Total iterations = 6
Transient iterations = 0
Circuit Equations = 5
Transient timepoints = 0
Accepted timepoints = 0
Rejected timepoints = 0
Total Analysis Time = 0.01
Transient time = 0
Matrix reordering time = 3.43425E-08
L-U decomposition time = 9.73232E-08
Matrix solve time = 1.31666E-07
transient L-U decomp time = 0
Transient solve time = 0
Transient iters per point = 6
Load time = 1.58383E-07
```

7.5.2 Test de Análisis en Alterna (.AC)

El circuito elegido en este caso es un RLC paralelo que presenta una resonancia alrededor de la frecuencia $f_{resonancia} = \frac{1}{2\pi\sqrt{LC}}$. Se utilizará un generador de corriente para excitar la impedancia equivalente del circuito de la figura 7.24. La representación topológica de dicho

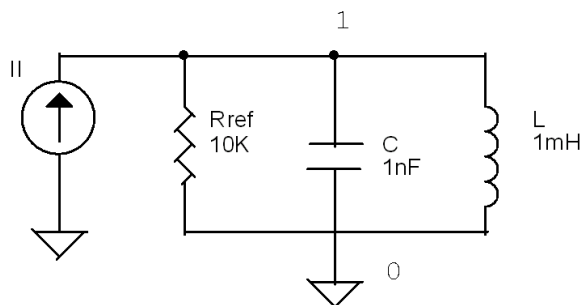


Figura 7.24 – Circuito de test para el análisis en alterna.

circuito en el formato de texto se muestra en el listado 7.29.

```
1 Title: Parallel RLC Circuit
2 II 0 1 ac 0.1
3 L 1 0 1m
4 C 1 0 1n
5 Rref 1 0 10K
6 .control
7 *ANALISIS FRECUENCIA [ac ( DEC | OCT | LIN ) N Fstart Fstop]}
8 ac dec 25 1.0E5 1E6
9 plot v(1) xlimit 100k 300k ylog
10 print V(1) > ac.txt
11 display > ac.var
12 asciiplot V(1) > ac.plt
13 .endc
14 .end
```

Listado 7.29 – Netlist del circuito de test para análisis en frecuencia.

El valor de la tensión en los extremos de la impedancia resonante equivalente se presenta en la figura 7.25. La resonancia está cerca de la frecuencia 160 KHz. Los tiempos de simulación

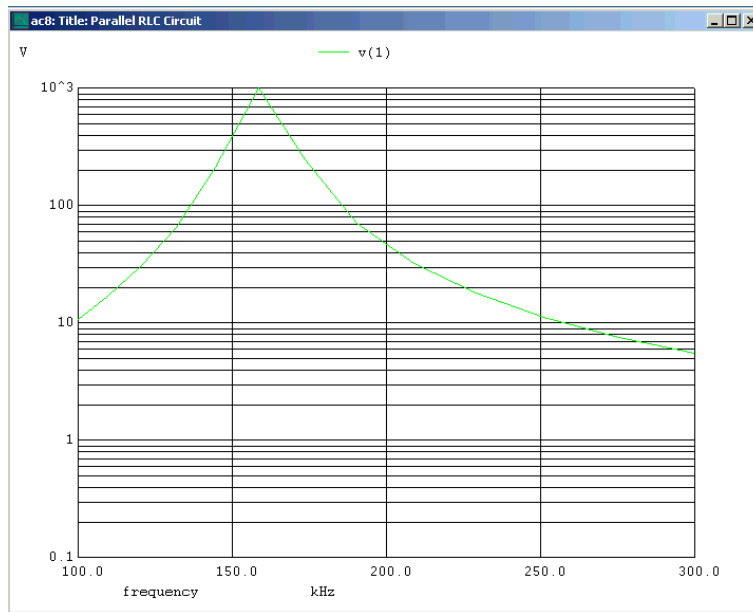


Figura 7.25 – Resultados del análisis en alterna.

de este análisis de alterna se muestran en este resumen obtenido mediante el comando `rusage`:

```
Total elapsed time: 0.000 seconds.
elapsed time since last call: 0.000 seconds. Current dynamic memory usage = 676 KB,
Dynamic memory limit = 938192 KB.
Nominal temperature = 27
Operating temperature = 27
Total iterations = 6
Transient iterations = 0
Circuit Equations = 3
Transient timepoints = 0
Accepted timepoints = 0
Rejected timepoints = 0
Total Analysis Time = 6.10598E-08
Transient time = 0
matrix reordering time = 1.2212E-07
L-U decomposition time = 6.10598E-08
Matrix solve time = 1.8318E-07
transient L-U decomp time = 0
Transient solve time = 0
Transient iters per point = 6
Load time = 2.44239E-07
```

7.5.3 Test de Análisis Transitorio (.TRAN)

El circuito bajo prueba es un resonante RLC serie. La fuente de señal genera una escalón de tensión que se aplica al circuito entre los nodos 1 y 0, figura 7.26. La representación de



Figura 7.26 – Circuito de test para el análisis transitorio.

dicho circuito en el formato de texto utilizado como fuente corresponde al listado 7.30. Se

han realizado de manera consecutiva los análisis de punto de operación y transitorio.

```

1 | Title: Series RLC circuit
2 | Vin 1 0 PWL 0 0 10N 5 25M 5 25.01M 0
3 | L1 2 3 0.125
4 | C1 3 0 1u
5 | R1 1 2 50
6 | .control
7 | *ANALISIS TRANSITORIO
8 | tran 0.2M 50M
9 | print V(3) V(1) > tran.txt
10 | display > tran.var
11 | asciiplot V(1) V(3) > tran.plt
12 | .endc
13 | .end

```

Listado 7.30 – Netlist del circuito de test para análisis transitorio.

Las variables internas disponibles en el circuito se han almacenado en el fichero *tran.var*, cuyo contenido se muestra a continuación:

```

Title: Title: series RLC circuit
Name: tran1 (Transient Analysis)
Date: Thu Sep 04 15:19:56 2010
V(1)      : voltage, real, 262 long
V(2)      : voltage, real, 262 long
V(3)      : voltage, real, 262 long
l1\#branch : current, real, 262 long
time      : time, real, 262 long [default scale]
vin\#branch : current, real, 262 long

```

El valor de las variables de tensión correspondiente a los nodos 1 (entrada) y 3 (salida) se han volcado al fichero de texto *tran.txt* donde se almacena el valor de cada una de las variables en cada instante de tiempo simulado (*time-step*). Los valores obtenidos tras la simulación se representan gráficamente

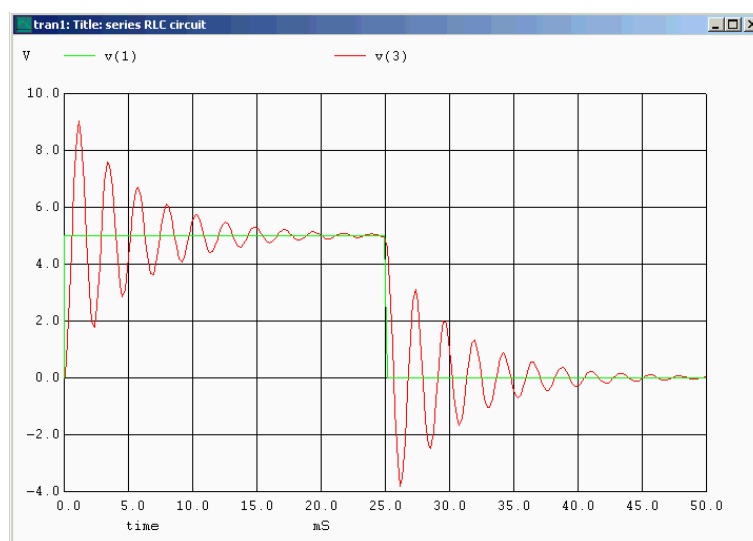


Figura 7.27 – Resultados de la simulación temporal del circuito de la figura 7.26.

7.5.4 Comprobación de la estimación de la función de transferencia (.FT)

El circuito elegido para el cálculo de la función de transferencia entre dos puertos y sus impedancias de entrada y salida correspondientes es un amplificador de pequeña señal con un dispositivo BJT. La fuente de corriente *II* se utiliza como fuente excitadora en el análisis (.FT), ver figura 7.26. La descripción del circuito y los análisis a realizar se incluyen en el

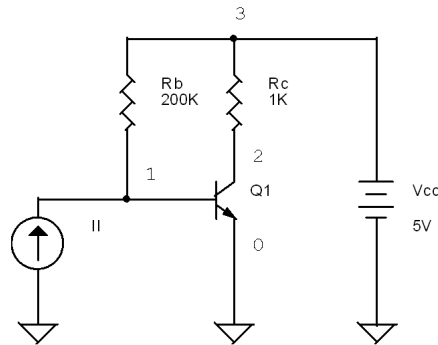


Figura 7.28 – Circuito de test para de la función de transferencia

fichero de texto correspondiente al listado siguiente:

```
Title: One-Transistor Circuit
Q1 2 1 0 QMOD
RC 2 3 1K
RB 1 3 200K
VCC 3 0 5
II 0 1
.MODEL QMOD NPN}
.control
*Análisis de Función de Transferencia
tf v(2) ii}
display > ft.var
print all > ft.txt
rusage everything > tf.ope
.endc
.end
```

El resultado del análisis de función de transferencia entre el puerto de entrada definido por los nodos (1,0) y el de salida (2,0) es:

```
transfer_function = -9.93889e+04 //Equivalente a la ganancia del circuito}
ii#input_impedance = 1.222154e+03 //Impedancia de entrada
output_impedance_at_v(2) = 9.999999e+02 //Impedancia de salida}
```

Los tiempos de simulación obtenidos para este caso se muestran a continuación:

```
Total elapsed time: 0.000 seconds.
Elapsed time since last call: 0.000 seconds.
Current dynamic memory usage = 672 KB,
Dynamic memory limit = 709676 KB.
Nominal temperature = 27
Operating temperature = 27
Total iterations = 12
Transient iterations = 0
Circuit Equations = 5
Transient timepoints = 0
Accepted timepoints = 0
Rejected timepoints = 0
Total Analysis Time = -4.57512E-08
Transient time = 0
Matrix reordering time = -9.15024E-08
L-U decomposition time = -1.83005E-07
Matrix solve time = -2.74507E-07
Transient L-U decomp time = 0
Transient solve time = 0
Transient iters per point = 12
Load time = -3.06929E-07
```

7.5.5 Test del análisis de Fourier posterior al análisis transitorio (.FOURIER)

El circuito elegido para el análisis de Fourier es muy sencillo y consta únicamente de una resistencia alimentada por un generador de señal triangular. Una vez realizado el análisis transitorio de al menos 5 periodos de la señal de excitación se realiza un análisis espectral a frecuencia 1KHz obteniéndose los valores de cada componente armónico, ver la figura 7.29. La descripción del circuito y los parámetros del análisis a realizar se incluyen en el fichero

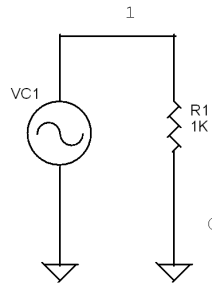


Figura 7.29 – Circuito de test para el análisis de Fourier

de texto siguiente.

```

1 | Title: Análisis de Fourier
2 | R1 1 0 1k
3 | V 1 0 PULSE(-1 1 0NS 0.5ms 0.5ms 0.1ns 1ms)
4 | .control
5 | *ANALISIS TRANSITORIO
6 |   tran 5u 10ms 0 10us
7 |   rusage everything > tran.ope
8 |   print V(1) > tran.txt
9 |   plot v(1)
10 |  display > tran.var
11 |  asciiplot V(1) > tran.plt
12 | *ANALISIS DE FOURIER
13 |   set nfreqs=5 *Armonicos para el analisis de Fourier
14 |   fourier 1000 V(1) > tran.fou *Periodo de Vin => Freq=1/1E-3s=1000Hz
15 | .endc
16 | .end

```

Listado 7.31 – Netlist del circuito de test para análisis de Fourier.

El resultado del análisis espectral de la tensión entre los nodos (1,0) se muestra en diferentes formatos y para las frecuencias armónicas nF_c en el siguiente listado.

```

Fourier analysis for v(1):
No. Harmonics: 10, THD: 11.99 %, Gridsize: 200, Interpolation Degree: 1

```

Harmonic	Frequency	Magnitude	Phase	Norm. Mag	Norm. Phase
0	0	2e-07	0	0	0
1	1000	0.810236	-90	1	0
2	2000	9.22435e-15	-87.905	1.13848e-14	2.09471
3	3000	0.08973	-90	0.110745	-3.6155e-05
4	4000	8.7488e-15	-84.066	1.07978e-14	5.93362
5	5000	0.0320895	-90	0.0396052	-7.2742e-05
6	6000	8.89099e-15	-77.921	1.09733e-14	12.0785
7	7000	0.0162091	-90	0.0200054	-0.00011007
8	8000	8.11568e-15	-70.626	1.00164e-14	19.3744
9	9000	0.00967396	-90	0.0119397	-0.00014845

7.5.6 Test del Análisis de Sensibilidad (.SENS)

El circuito elegido para realizar el análisis de sensibilidad de la variable de salida correspondiente a la corriente que circula por la fuente VMEAS se muestra en la figura 7.30 Este análisis se desarrolla en dos fases: durante la primera, se calcula la sensibilidad en DC, posteriormente en AC. El resultado genera los valores de variación de la salida frente a cambios de la entrada. El fichero de texto con la descripción del circuito es el siguiente:

```

1 | Title: Current mirror
2 | REF 3 2 4.3K
3 | Q1 2 2 0 QMOD
4 | Q2 1 2 0 QMOD
5 | VMEAS 3 1
6 | VCC 3 0 5
7 | .MODEL QMOD NPN BF=100 VA=50}
8 | .control
9 | *ANALISIS SENSIBILIDAD}
10 |  sens i(vmeas)
11 |  print all > sens.txt
12 |  rusage everything > sens.ope
13 | .endc
14 | .end

```

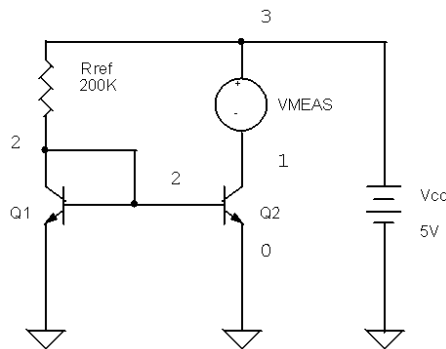


Figura 7.30 – Circuito de test para el análisis de sensibilidad.

Listado 7.32 – Netlist del circuito de test para análisis de sensibilidad.

El resultado del análisis de sensibilidad genera este listado que indica cómo afectan a la tensión VMEAS las diferentes variaciones de los parámetros del circuito:

```

q1: bf = 1.017945e-07
q1: br = 2.756382e-12
q1: eg = -1.14605e-12
q1: fc = -2.54423e-12
q1: is = -1.02812e+13
q1: nc = -6.36058e-13
q1: ne = -8.48078e-13
q1: nf = 3.073697e-02
q1: nr = -1.27804e-12
q1: tnom = 1.572806e-04
q1: vaf = 8.790935e-14
q1: xti = -4.24039e-13
q1: area = -1.02812e-03
q1: temp = -5.48740e-05
q2: bf = 1.017945e-07
q2: br = -6.51058e-12
q2: eg = 1.965291e-12
q2: fc = 4.362946e-12
q2: is = 1.034977e+13
q2: nc = 1.090736e-12
q2: ne = 1.454315e-12
q2: nf = -3.09418e-02
q2: nr = 2.181473e-12
q2: tnom = -1.58329e-04
q2: vaf = -1.62931e-06
q2: xti = 7.271576e-13
q2: area = 1.034977e-03
q2: temp = 5.523972e-05
ref = -2.41466e-07
vcc = 2.649248e-04
vmeas = -1.92740e-05

```

7.5.7 Comprobación del análisis polo-cero (.PZ)

En este caso se ha seleccionado un filtro en T pasivo que se muestra en la figura 7.31. El análisis polo-cero permite obtener una función de transferencia $V_{SALIDA}/V_{ENTRADA}$ [VOL] o $V_{SALIDA}/I_{ENTRADA}$ [CUR] y la obtención de los polos [POL], ceros [ZER] o ambos [PZ] entre un puerto de entrada definido por (NODE1, NODE2) y otro de salida (NODE3, NODE4). La descripción topológica y listado de comandos de la simulación del filtro-T es el siguiente:

```

1 Title Bridge T Filter
2 V1 1 0 12 AC 1
3 C1 1 2 1u
4 C2 2 3 1u
5 R3 2 0 1k
6 R4 1 3 1k
7 .control

```

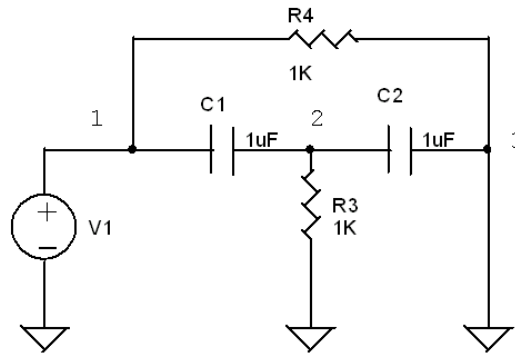


Figura 7.31 – Circuito de test para el análisis polo-cero.

```

8 *ANALISIS POLO-CERO
9 *.PZ NODE1 NODE2 NODE3 NODE4 [CUR VOL] [POL ZER PZ]
10 pz 1 0 3 0 vol pz
11 rusage everything > pz.ope
12 print all > pz.txt
13 display > pz.var
14 .endc
15 .end
    
```

Listado 7.33 – Netlist del circuito de test para análisis polo-cero.

Los polos-ceros encontrados por el simulador para este circuito entre el puerto de entrada (1, 0) y el de salida (3, 0) son:

```

pole (1) = -2.61803e+03,0.000000e+00
pole (2) = -3.81966e+02,0.000000e+00
zero (1) = -1.00000e+03,0.000000e+00
zero (2) = -1.00000e+03,0.000000e+00
    
```

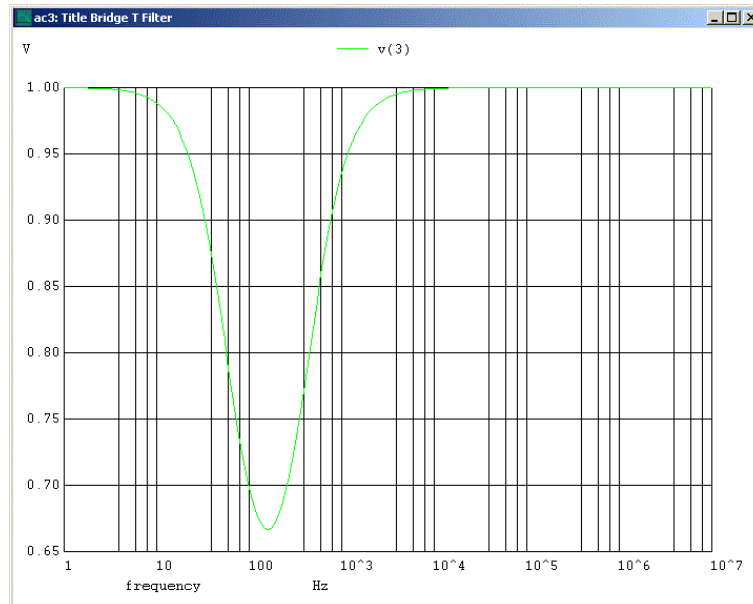


Figura 7.32 – Tensión en el Nodo 3 [Salida del Filtro en T].

Los tiempos de simulación utilizados para este análisis de función de transferencia se muestran a continuación:

```

Total elapsed time: 4.000 seconds.
elapsed time since last call: 0.000 seconds.
Current dynamic memory usage = 1200 KB
Dynamic memory limit = 816280 KB.
Nominal temperature = 27
    
```



```

Operating temperature = 27
Total iterations = 6
Transient iterations = 0
Circuit Equations = 5
Transient timepoints = 0
Accepted timepoints = 0
Rejected timepoints = 0
Total Analysis Time = 0.0199999
Transient time = 0
Matrix reordering time = 0
L-U decomposition time = 0
Matrix solve time = 0
transient L-U decomp time = 0
Transient solve time = 0
Transient iters per point = 6
Load time = 0

```

7.5.8 Test del Análisis de Ruido (.NOISE)

Para comprobar el análisis de ruido se ha seleccionado es una etapa amplificadora que utiliza como elemento activo un transistor NPN y se presenta en la figura 7.33. El análisis de

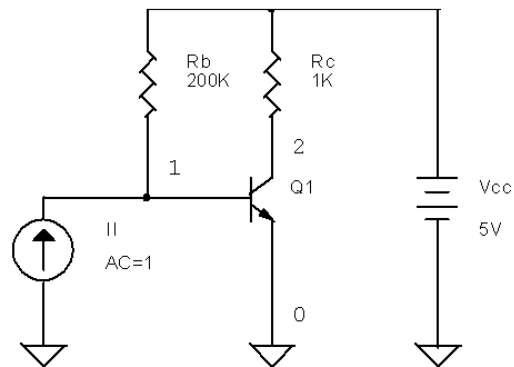


Figura 7.33 – Circuito de test para el análisis de ruido

ruido calcula el ruido total en un puerto de salida. Las magnitudes obtenidas corresponden a:

1. Densidad espectral de ruido [V/I]. Unidades $\left[\frac{V^2}{Hz}, \frac{A^2}{Hz}\right]$
2. Ruido de integración entre los límites de frecuencia dados. Unidades $[V^2, A^2]$

El fichero de texto con la descripción del amplificador se muestra en el listado 7.34:

```

1 | Title: Transistor noise analysis
2 | Q1 2 1 0 QMOD
3 | RC 2 3 1K
4 | RB 1 3 200K
5 | VCC 3 0 5
6 | II 0 1 AC 1
7 | .model qmod npn cje=1p cjc=2p
8 | .control
9 | *ANALISIS DE RUIDO [.noise v(OUT) SRC { DEC OCT LIN } NP FSTART FSTOP <PTSPRSUM>]
10 | * noise2 Noise Test, Integrated Noise - V2 or A2
11 | * noise1 Noise Test, Noise Spectral Density Curves - (V2 or A2)/Hz
12 | noise v(2) ii dec 10 0.1 10MEG
13 | rusage everything > noise.ope
14 | setplot noise1
15 | display > n_spec.var
16 | print onoise_spectrum inoise_spectrum > n_spec.txt
17 | setplot noise2
18 | display > n_tot.var
19 | print onoise_total > inoise_total > n_tot.txt
20 | .endc
21 | .end

```

Listado 7.34 – *Netlist del circuito de test para análisis de ruido.*

Los valores del ruido dentro del intervalo de frecuencias elegidas para la simulación [0.1, 10MHz] ascienden a:

```
onoisetotal = 1.181317e-07 V2
inoisetotal = 8.344801e-1 A2
```

Los tiempos de simulación utilizados para este análisis de función de transferencia se muestran a continuación:

```
Total elapsed time: 139.000 seconds.
elapsed time since last call: 0.000 seconds.
Current dynamic memory usage = 18496 KB
Dynamic memory limit = 682364 KB.
Nominal temperature = 27
Operating temperature = 27
Total iterations = 18
Transient iterations = 0
Circuit Equations = 5
Transient timepoints = 0
Accepted timepoints = 0
Rejected timepoints = 0
Total Analysis Time = 0.0200001
Transient time = 0
matrix reordering time = 1.69734E-07
L-U decomposition time = 3.39467E-07
Matrix solve time = 5.09201E-07
transient L-U decomp time = 0
Transient solve time = 0
Transient iters per point = 18
Load time = 5.94067E-07
```

Los valores de la densidad espectral de ruido tanto de entrada como de salida se muestran en la figura 7.34.

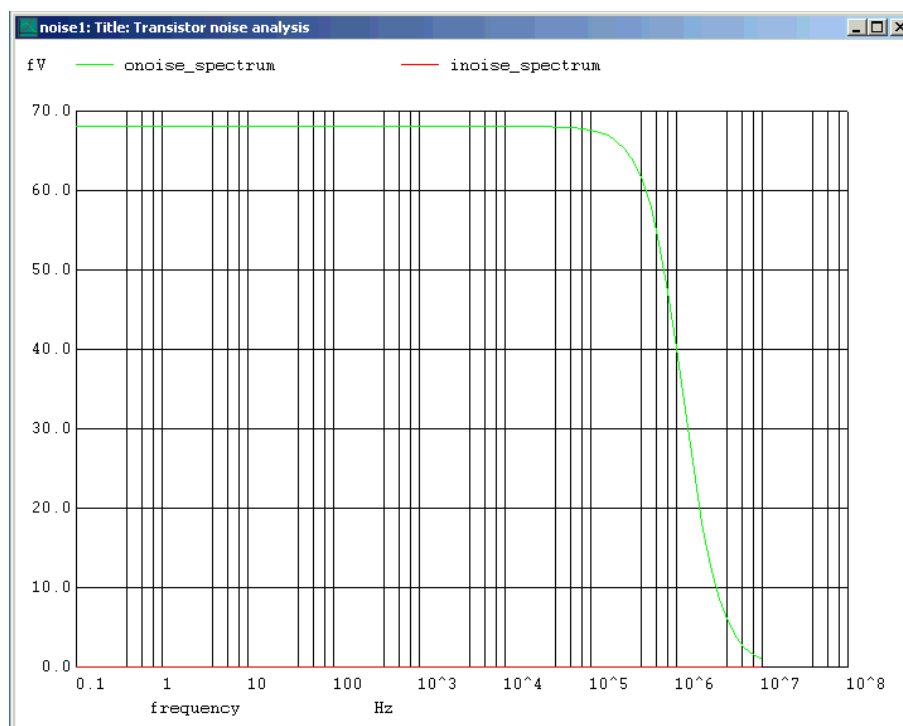


Figura 7.34 – *Resultados del análisis de ruido*

7.5.9 Test del Análisis de Distorsión (.DISTO)

El circuito elegido para realizar el análisis de distorsión utiliza un dispositivo pasivo no lineal como generador de productos armónicos y de intermodulación. El diodo *D1* se polariza mediante la alimentación que introduce en continua la fuente VCC. El circuito a analizar se presenta en la figura 7.35. Este análisis tiene algunas particularidades especiales, parte de

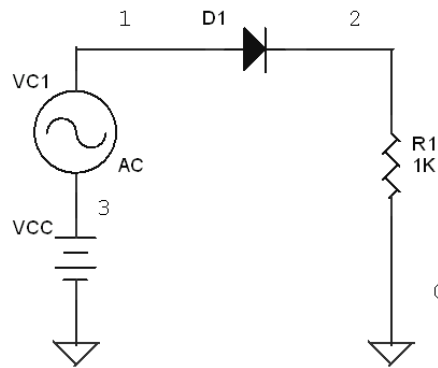


Figura 7.35 – Circuito de test para el análisis de distorsión

las características de éste se introducen en la fuente de excitación y el resto en la directiva de análisis (.DISTO) [Roy89]. La descripción del circuito se presenta en el listado 7.35:

```

1 | Title:Diode-Resistor circuit
2 | r1 2 0 1k
3 | d1 1 2 diode
4 | vcc 1 3 5v ac 0.001 sin(5 0.01 1000) distof1 0.01 distof2 0.01
5 | vcc2 3 0 0v
6 | *Amplitud Distorsión
7 | .model diode D is=1.0e-14 tt=0.1n cjo=2p
8 | .control
9 | *ANALISIS DE DISTORSION
10 | *.DISTO [DEC/OCT/LIN] ND FSTART FSTOP <F2OVERF1>
11 | disto dec 1 1000.0 1000.0
12 | rusage everything > disto.ope
13 | display > disto.var
14 | .endc
15 | .end

```

Listado 7.35 – Netlist del circuito de test para análisis de distorsión.

Como resultado del análisis se dispone de diferentes representaciones que deben ser seleccionados mediante la primitiva `.setplot NUM_PLOT`. Se puede elegir entre el estudio del segundo y tercer armónico de la señal excitadora de 1KHz de la fuente de señal VCC. Se muestra un volcado de la ventana de salida de datos de la consola del simulador:

```

Spice 2 -> setplot
Type the name of the desired plot:
Current disto2 Title:Diode-Resistor circuit (DISTORTION - 3rd harmonic)
      disto1 Title:Diode-Resistor circuit (DISTORTION - 2nd harmonic)
      op1 Title:Diode-Resistor circuit (Operating Point)
      const Constant values (constants)
Spice 3 -> setplot disto2
Spice 4 -> print all
v(1) = 0.000000e+00,0.000000e+00
v(2) = -2.60903e-11,7.749508e-17 // DISTORTION 3rd harmonic
v(3) = 0.000000e+00,0.000000e+00
frequency = 1.000000e+03,0.000000e+00
vcc\#branch = 2.609026e-14,-7.74951e-20
vcc2\#branch = 2.609026e-14,-7.74951e-20
Spice 5 -> setplot disto1
Spice 6 -> print all
v(1) = 0.000000e+00,0.000000e+00
v(2) = 3.422550e-08,-5.96281e-14 // DISTORTION - 2nd harmonic
v(3) = 0.000000e+00,0.000000e+00
frequency = 1.000000e+03,0.000000e+00
vcc\#branch = -3.42255e-11,5.962811e-17
vcc2\#branch = -3.42255e-11,5.962811e-17

```

7.6 Modelado de alta frecuencia de la parte intrínseca.

Como se explica en la sección 3.2.4.1, es importante resaltar algunas relaciones y desigualdades entre las capacidades. Algunas de las capacidades recíprocas no son simétricas, es decir, $C_{kj} \neq C_{jk}$, ya que por ejemplo C_{gd} representa el efecto de drenador sobre la puerta y C_{dg} el efecto de la puerta sobre el drenador que es mucho mayor.

En el caso de que todas las tensiones variables sean iguales $v_s(t) = v_g(t) = v_b(t) = v_d(t) = v(t)$, se verifica (7.6.1):

$$i_{da}(t) = (+C_{dd} - C_{dg} - C_{db} - C_{ds}) \frac{dv}{dt} \quad (7.6.1)$$

y dado que no existe tensión entre los terminales, las corrientes son nulas también, por lo tanto,

$$C_{dd} - C_{dg} - C_{db} - C_{ds} = 0 \Rightarrow C_{dd} = C_{dg} + C_{db} + C_{ds} \quad (7.6.2)$$

Se puede llegar a una expresión similar, (7.6.3), en el caso de suponer que $\frac{dv_g}{dt} = \frac{dv_b}{dt} = \frac{dv_s}{dt} = 0$ y mediante (3.2.31).

$$(C_{dd} - C_{gd} - C_{bd} - C_{sd}) \frac{dv_d}{dt} = 0 \Rightarrow C_{dd} = C_{gd} + C_{bd} + C_{sd} \quad (7.6.3)$$

Procediendo de un modo similar, se deriva (7.6.4):

$$C_{dd} = C_{dg} + C_{db} + C_{ds} = C_{gd} + C_{bd} + C_{sd} \quad (7.6.4a)$$

$$C_{gg} = C_{gd} + C_{gb} + C_{gs} = C_{dg} + C_{bg} + C_{sg} \quad (7.6.4b)$$

$$C_{bb} = C_{bg} + C_{bd} + C_{bs} = C_{gb} + C_{db} + C_{sb} \quad (7.6.4c)$$

$$C_{ss} = C_{sg} + C_{sd} + C_{sb} = C_{gs} + C_{ds} + C_{bs} \quad (7.6.4d)$$

Para simplificar la complejidad numérica y el tiempo de evaluación de las expresiones presentadas en (3.2.31), se pueden referenciar todas las tensiones de los terminales con respecto a la tensión de fuente ¹⁰.

$$v_D = v_{DS} + v_S \quad (7.6.5a)$$

$$v_G = v_{GS} + v_S \quad (7.6.5b)$$

$$v_B = v_{BS} + v_S \quad (7.6.5c)$$

y aplicando expresiones análogas para el caso de (3.2.31) se puede llegar a (7.6.6), donde el valor del paréntesis es nulo como puede observarse en (7.6.2).

$$i_d(t) = C_{dd} \frac{dv_{ds}}{dt} - C_{dg} \frac{dv_{gs}}{dt} - C_{db} \frac{dv_{bs}}{dt} + (C_{dd} - C_{dg} - C_{db} - C_{ds}) \frac{dv_s}{dt} \quad (7.6.6)$$

Así se obtienen unas expresiones computacionalmente más simples (7.6.7) que serán usadas

¹⁰La tensión en minúscula y subíndice en mayúscula representa la suma de la tensión DC más la variable.

en el modelado que presentamos a continuación:

$$i_d(t) = +C_{dd} \frac{dv_{ds}}{dt} - C_{dg} \frac{dv_{gs}}{dt} - C_{db} \frac{dv_{bs}}{dt} \quad (7.6.7a)$$

$$i_g(t) = -C_{gd} \frac{dv_{ds}}{dt} + C_{gg} \frac{dv_{gs}}{dt} - C_{gb} \frac{dv_{bs}}{dt} \quad (7.6.7b)$$

$$i_b(t) = -C_{bd} \frac{dv_{ds}}{dt} - C_{bg} \frac{dv_{gs}}{dt} + C_{bb} \frac{dv_{bs}}{dt} \quad (7.6.7c)$$

$$i_s(t) = -i_d(t) - i_g(t) - i_b(t) \quad (7.6.7d)$$

Con los datos aportados por (7.6.7) se puede proceder a complementar el circuito equivalente de pequeña señal para las corrientes de transporte, con las nuevas capacidades y fuentes controladas auxiliares. En la figura 7.36 se muestra el circuito equivalente completo de alta frecuencia para las corrientes de transporte y carga [Tsi03, RRGJM10].

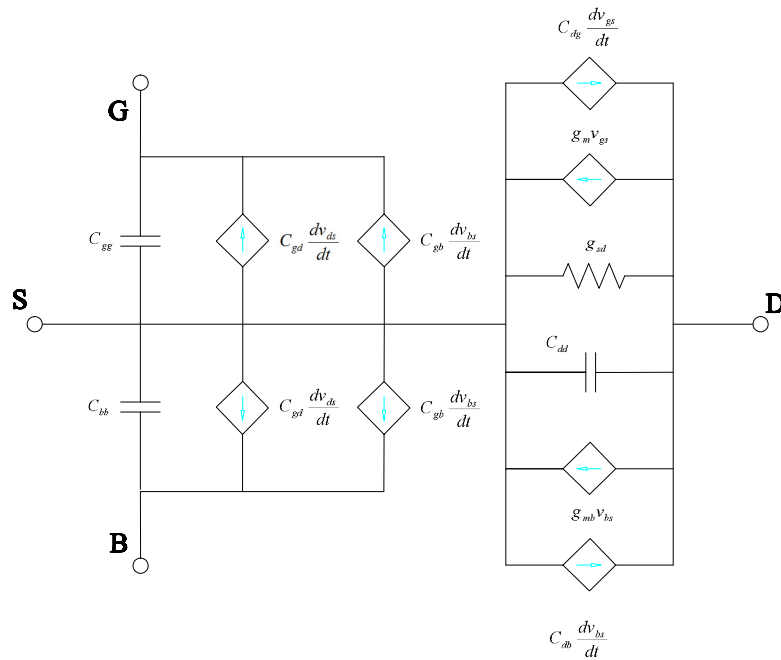


Figura 7.36 – Modelo alta frecuencia completo.

CAPÍTULO

8

CONCLUSIONES.

Uno de los resultados más importantes presentados en esta tesis está relacionado con la demostración de que los modelos compactos codificados en [Verilog-A](#) son una alternativa eficaz para sustituir los modelos basados en código fuente. La implementación en [Verilog-A](#) presenta innumerables ventajas como la facilidad de codificación, su naturaleza multiplataforma, etc. A lo largo de los cinco capítulos que componen este trabajo se desarrollan diferentes modelos compactos para un conjunto variado de dispositivos electrónicos, tanto transistores como sensores, en particular, sensores de corriente. Todos ellos han sido implementados en diferentes simuladores de circuitos, principalmente en [Verilog-A](#), para estudiar el comportamiento de circuitos electrónicos basados en los dispositivos modelados. Los modelos introducidos incorporan las relaciones entre tensiones y corrientes, los efectos térmicos, las contribuciones de los efectos capacitivos, y también las fuentes de ruido.

Entre las deducciones más destacables cabe señalar las que se enumeran a continuación:

- Se han investigado en profundidad las diferentes alternativas que existen en la actualidad para la implementación de modelos compactos para simulación circuital. Se ha profundizado en las dos opciones principales: implementación de modelos en código fuente y en [Verilog-A](#).
- Se ha desarrollado un simulador basado en SPICE multiplataforma, ESPICE, con todas las prestaciones de un simulador de circuitos convencional y se le ha dotado de una interfaz gráfica y funciones que permiten incorporar fácilmente modelos en código fuente. Este simulador, en su nueva versión, posibilita también la modificación de los algoritmos de simulación, y la depuración durante el proceso iterativo de desarrollo de los modelos incluidos.

- Se ha modificado el modelo BSIMSOI en código fuente para incluir los efectos del *overshoot* de la velocidad en transistores de canal corto de puerta simple fabricados sobre tecnología SOI. Tras la compilación completa del simulador hemos llevado a cabo la simulación de varios circuitos digitales, entre ellos un anillo oscilador, y se ha podido estudiar la dependencia de la frecuencia de oscilación del anillo respecto al parámetro que controla la incidencia de los efectos del *overshoot* de la velocidad. En particular, se ha observado un aumento de la frecuencia de oscilación de un 20 % para dispositivos de longitudes de canal de 75 nm respecto a los de canal largo. Para un dispositivo de 150 nm, existe un aumento de un 10 % aproximadamente cuando el parámetro que controla los efectos de *overshoot* se multiplica por un factor 4 (de 10 a 40 cm³/Vs).
- Se ha implementado en Verilog-A un modelo completo para transistores SGT, basado en una descripción de modelo de control de carga. Se han incluido efectos cuánticos, de saturación de la velocidad, de canal corto y de *overshoot* de la velocidad, entre otros. También se ha implementado la matriz de capacidades que describe el comportamiento del dispositivo en régimen dinámico. Posteriormente, se han estudiado diferentes circuitos digitales: inversores CMOS y anillos osciladores. La tensión de salida de los inversores y anillos osciladores, así como la frecuencia de oscilación, ha sido analizada, evaluando la contribución de los efectos físicos más representativos incluidos en el modelo.
- En dispositivos de longitud de canal de 50 nm, la frecuencia de oscilación depende ostensiblemente del radio del canal de silicio cilíndrico. Se ha observado un comportamiento casi lineal de la frecuencia de oscilación con la variación del parámetro que caracteriza los efectos de *overshoot* de la velocidad.
- Con el fin de caracterizar en profundidad la velocidad de saturación de los portadores en los transistores a modelar, se ha estudiado este parámetro en profundidad por medio de un simulador de Monte Carlo desarrollado en nuestro grupo de investigación. Se ha demostrado que tanto en dispositivos de puerta simple como de puerta doble la velocidad de saturación depende de los mecanismos de dispersión por rugosidad superficial en la superficie semiconductor-óxido. Para un mismo dispositivo (de simple puerta) la velocidad de saturación depende del campo efectivo si se incluyen los efectos de dispersión por rugosidad superficial. También depende del grosor de la lámina de silicio que constituye el canal, disminuyendo la velocidad de saturación a medida que decrece el grosor de la lámina de silicio. La velocidad de saturación disminuye a medida que aumenta la rugosidad superficial de la interfase semiconductor-óxido, este efecto es más importante para láminas de silicio delgadas.
- En dispositivos de doble puerta, el estudio de la velocidad de saturación muestra efectos similares a los dispositivos de puerta simple, aunque la sensibilidad del valor que se obtiene para la velocidad de saturación ante variaciones del grosor de la lámina de silicio y de la rugosidad superficial es mayor para estos dispositivos.
- Hemos desarrollado también un modelo (implementado en Verilog-A) para dispositivos de doble-puerta DG-MOSFETs. Las expresiones para el cálculo de la corriente de drenador incluyen, al igual que en SGTs, los efectos del *overshoot* de la velocidad, la saturación de la velocidad, cuánticos y de canal corto. Tras implementar la contribu-

ción de las capacidades de la estructura, se han simulado también diferentes circuitos digitales basados en este dispositivo.

- En el caso de los dispositivos magnetorresistivos, hemos desarrollado un modelo compacto (implementado en [Verilog-A](#)) de cuatro terminales que puede ser usado con dispositivos [AMR](#) y [GMR](#) ([SV](#) y [MTJ](#)). Dispone de un conjunto de parámetros que, una vez extraídos, permiten reproducir resultados experimentales de estos sensores de corriente para diferentes tecnologías de fabricación.
- El modelo permite tratar sensores de corriente comerciales basados en dispositivos magnetorresistivos [AMR](#) de altas corrientes bajo condiciones cuasiestáticas para comportamientos lineales y no-lineales de la relación resistencia-corriente.
- Para los dispositivos [SV](#) hemos realizado ajustes paramétricos para obtener las relaciones resistencia-corriente tanto en elementos individuales como en puentes de Wheatstone, y se han caracterizado las variaciones de los parámetros del modelo con la temperatura. Hemos realizado simulaciones de la tensión de salida de puentes magnetorresistivos en régimen transitorio, reproduciendo correctamente los datos experimentales.
- Se han incluido en el modelo las dependencias térmicas y se ha implementado el código necesario para reproducir este comportamiento en el simulador de circuitos. Los datos obtenidos en simulación concuerdan con los experimentales, dejando de manifiesto que el enfoque Foster mediante dos celdas RC para modelar el comportamiento térmico es satisfactorio.
- Mediante la colaboración establecida por el Dpto. de Ingeniería Electrónica de la Universidad de Valencia y el INESC-MN de Portugal, hemos podido acceder al laboratorio de caracterización de la sala blanca de INESC-MN para medir y modelar el ruido eléctrico de dispositivos [SV](#) y [MTJ](#) simples, incorporando en el modelo compacto los efectos del ruido. El modelo se ha validado mediante los correspondientes datos experimentales.
- Se han diseñado, fabricado, caracterizado y simulado diferentes circuitos de instrumentación basados en sensores [SV](#). En particular, algunas de las aplicaciones implementadas son convertidores de resistencia a frecuencia, de corriente a frecuencia y un convertidor generalizado de impedancias. En estos casos, las medidas experimentales obtenidas en los circuitos coinciden con los resultados de simulación basados en el nuevo modelo implementado para los sensores. Confirmando así su validación experimental en su uso a nivel de aplicaciones.
- En la estancia realizada en Lisboa, se han podido fabricar con éxito un conjunto de dispositivos magnetorresistivos [MTJ](#) siguiendo las directrices del Prof. Càndid Reig, y utilizando la tecnología disponible en el INESC-MN. Se han considerado en los que los puentes de Wheatstone están formados por la conexión en serie de elementos unitarios [MTJ](#), lo que permite la utilización de estos sensores de corriente a las tensiones usuales de las aplicaciones de instrumentación convencionales.

Como líneas futuras de investigación proponemos:

- Modelado e implementación en [Verilog-A](#) de FinFETs y transistores *Trigate*.



- Introducción en los modelos de transistores ya existentes de los elementos extrínsecos como resistencias serie, capacidades de *overlap*, de unión y *fringing* entre los terminales puerta-fuente y puerta-drenador.
- Mejora de los modelos de movilidad para transistores con diferentes geometrías y orientaciones cristalográficas, y definición de estrategias de extracción de parámetros para los nuevos parámetros introducidos.
- Introducción de los efectos de histéresis en los sensores **GMR** estudiados: tanto **SV** como **MTJs**. Estudio de su influencia en los circuitos que se han diseñado en base a ellos.
- Estudio y modelado del comportamiento en frecuencia de los sensores de corriente **GMR**. Implementación en **Verilog-A**.
- Desarrollo de un simulador de dispositivos **GMR** para estudiar en profundidad la física del transporte para mejorar los modelos compactos y proponer nuevos diseños para incrementar la sensibilidad de los sensores.



BIBLIOGRAFÍA

- [Acc04] Accellera. *Proposed Verilog-A Language Extensions for Compact Modeling.*, August 2004.
- [ACM11] ACM. <http://www.eletr.ufpr.br/ogouveia/msm.pdf>, 2011.
- [AFS82] Tsuneya Ando, Alan B. Fowler, and Frank Stern. Electronic properties of two-dimensional systems. *Rev. Mod. Phys.*, 54:437–672, Apr 1982.
- [AI03] Inc. Accellera International. *Verilog-AMS language reference manual, Version 2.1.* Accellera International, Inc., Jan 2003.
- [AI08] Inc. Accellera International. *Verilog-AMS language reference manual, Version 2.3.* Accellera International, Inc., August 2008.
- [Ant69] A. Antoniou. Realization of gyrator using operational amplifiers and their use in rc active network synthesis. *IEEE Proceedings*, 116:1838–1842, 1969.
- [AP97] C.P. Auth and J.D. Plummer. Scaling theory for cylindrical, fully-depleted, surrounding-gate mosfet's. *Electron Device Letters, IEEE*, 18(2):74–76, feb 1997.
- [AR09] A. Amara and O. Rozeau. *Planar Double-Gate Transistor: From technology to circuit.* Springer, 2009.
- [ARHR94] N.D. Arora, R. Rios, Cheng-Liang Huang, and K. Raol. Pcim: a physically based continuous short-channel igfet model for circuit simulation. *Electron Devices, IEEE Transactions on*, 41(6):988–997, jun 1994.

- [Aro07] N. Arora. *MOSFET modelling for VLSI simulation. Theory and practice*. World Scientific, 2007.
- [BBF⁺88] M. N. Baibich, J. M. Broto, A. Fert, F. Nguyen Van Dau, F. Petroff, P. Etienne, G. Creuzet, A. Friederich, and J. Chazelas. Giant magnetoresistance of (001)fe/(001)cr magnetic superlattices. *Phys. Rev. Lett.*, 61(21):2472–2475, Nov 1988.
- [BDR70] W.F. Brinkman, R.C. Dynes, and J.M. Rowell. Tunneling conductance of asymmetrical barriers. *Journal of Applied Physics*, 41(5):1915–1921, 1970.
- [BGSZ89a] G. Binasch, P. Grünberg, F. Saurenbach, and W. Zinn. Enhanced magnetoresistance in layered magnetic structures with antiferromagnetic interlayer exchange. *Phys. Rev. B*, 39:4828–4830, Mar 1989.
- [BGSZ89b] G. Binasch, P. Grunberg, F. Saurenbach, and W. Zinn. Enhanced magnetoresistance in layered magnetic-structures with antiferromagnetic interlayer exchange. *Physical Review B*, 39(7):4828–4830, 1989.
- [Bha09] A. Bhattacharyya. *Compact MOSFET models for VLSI design*. John Wiley and Sons, 2009.
- [BKGDV06] Marija Blagojevic, Maher Kayal, Michel Gervais, and Daniela De Venuto. SOI Hall-sensor front end for energy measurement. *IEEE Sensors Journal*, 6(4):1016–1021, 2006.
- [BKM⁺05] M.A. Belaid, K. Ketata, H. Maanane, M. Gares, K. Mourgues, and J. Marcon. Analysis and simulation of self-heating effects on rf ldmos devices. In *Simulation of Semiconductor Processes and Devices, 2005. SISPAD 2005. International Conference on*, pages 231 – 234, sept. 2005.
- [BMBF00] M. Busani, R. Menozzi, M. Borgarino, and F. Fantini. Dynamic thermal characterization and modeling of packaged algaas/gaas hbt. *Components and Packaging Technologies, IEEE Transactions on*, 23(2):352–359, jun 2000.
- [Bre78] J.R. Brews. A charge-sheet model of the mosfet. *Solid-State Electronics*, 21(2):345 – 355, 1978.
- [BSI05] BSIM3. *BSIM Research Group*. University of California, Berkeley., 2005.
- [Car10] S. Cardoso. *Materiais magnéticos em nanotecnologias*, 2010.
- [CBRR⁺09] M.D. Cubells-Beltrán, C. Reig, D. Ramírez, S. Cardoso, and P.P. Freitas. Full wheatstone bridge spin-valve based sensors for ic currents monitoring. *IEEE Sensors Journal*, 2009. In press.
- [CC03] G. K. Celler and S. Cristoloveanu. Frontiers of silicon-on-insulator. *Applied Physics Review*, 93(9):4955–4978, 2003.

-
- [CCE06] Eric A. Vittoz Christian C. Enz. *Charge-Based MOS Transistor Modeling: The EKV Model for Low-Power and RF IC Design*. Wiley, 2006.
- [CdF01] Susana Cardoso de Freitas. *Dual-Stripe GMR and Tunnel Junction Read Heads and Ion Beam Deposition and Oxidation of Tunnel Junctions*. PhD thesis, Instituto Superior Técnico, Universidad Técnica de Lisboa, 2001.
- [CMC96] CMC. *CMC - Compact Model Council - URL*. CMC - Compact Model Council, Agosto 1996.
- [CMF⁺07] S. Cardoso, R.J. Macedo, R.A. Ferreira, A. Augusto, and P.P. Freitas. Ion beam assisted deposition of mgo barriers for magnetic tunnel junctions. In *52nd Conference on Magnetism and Magnetic Materials*, Nov 2007.
- [Col04] Jean-Pierre Colinge. Multiple-gate soi mosfets. *Solid-State Electronics*, 48(6):897 – 905, 2004. Silicon On Insulator Technology and Devices.
- [Col08] J. Colinge. *FinFETs and Other Multi-Gate Transistors*. Springer Publishing Company, Incorporated, 1st edition, 2008.
- [Con11] SOI Industry Consortium. *Soi industry consortium - promoting the benefits of soi technology*, 2011.
- [CP07] H.-J. Cho and J. D. Plummer. Modeling of surrounding gate mosfets with bulk trap states. *Electron Devices, IEEE Transactions on*, 54(1):166 –169, jan. 2007.
- [CPA96] O. Casas and R. Pallas-Areny. Basics of analog differential filters. *Instrumentation and Measurement, IEEE Transactions on*, 45(1):275 –279, feb 1996.
- [CWRB75] Ho Chung-Wen, A. Ruehli, and P. Brennan. The modified nodal approach to network analysis. *Circuits and Systems, IEEE Transactions on*, 22(6):504 – 509, June 1975.
- [DGRG06] L. Donetti, F. Gámiz, J. B. Roldán, and A. Godoy. Acoustic phonon confinement in silicon nanolayers: Effect on electron mobility. *AIP*, 100(1):013701, 2006.
- [dMDM05] M.J. de Moura Dias Mendes. *Micromagnetic Simulations of Spin Valve devices*. PhD thesis, Instituto Superior Técnico, Universidad Técnica de Lisboa, 2005.
- [DXH⁺06] Mohan V. Dunga, Xuemei (Jane) Xi, Jin He, Weidong Liu, Kanyu M. Cao, Xiaodong Jin, Jeff J. Ou, Mansun Chan, Ali M. Niknejad, and Chenming Hu. *BSIM4.6.0 MOSFET Model*. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 2006.

- [EKV95] C. Enz, F. Krummenacher, and E. Vittoz. An analytical mos transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications. In *Analog Integrated Circuits and Signal Processing Journal on Low-Voltage and Low-Power Design*, volume 8, pages 83–114, 1995.
- [EKV11] EKV. <http://legwww.epfl.ch/ekv/>, 2011.
- [Fer07] A. Fert. The origin, development and future of spintronics. *Nobel Lecture*, Dic 2007.
- [Fer08] Albert Fert. Nobel lecture: Origin, development, and future of spintronics. *Rev. Mod. Phys.*, 80(4):1517–1530, Dec 2008.
- [FFCC07] P. P. Freitas, R. Ferreira, S. Cardoso, and F. Cardoso. Magnetoresistive sensors. *Journal of Physics-Condensed Matter*, 19(16):21, 2007.
- [FGMT97] V Ferrari, C Ghidini, D Marioli, and A Taroni. A conditioning circuit for resistive sensors combining frequency and duty-cycle modulation of the same output signal. *Measurement Science and Technology*, 8(7):827, 1997.
- [GdA09] José M. Gamelas de Almeida. *Tunnel Junction Sensors*. PhD thesis, Instituto Superior Técnico, 2009.
- [GF01] F Gamiz and M V Fischetti. Monte carlo simulation of double-gate silicon-on-insulator inversion layers: The role of volume inversion. *Journal of Applied Physics*, 89(10):5478, 2001.
- [GFW⁺85] S. M. Goodnick, D. K. Ferry, C. W. Wilmsen, Z. Liliental, D. Fathy, and O. L. Krivanek. Surface roughness at the si(100)-sio₂ interface. *Phys. Rev. B*, 32:8171–8186, Dec 1985.
- [GGS04] F. Gámiz, A. Godoy, and C. Sampedro. Monte carlo simulation of electron velocity overshoot in dgsoi mosfets. In *International Workshop of Computational Electronics*, 2004.
- [GLVR⁺98] F. Gamiz, J.A. Lopez-Villanueva, J.B. Roldan, J.E. Carceller, and P. Cartujo. Monte carlo simulation of electron transport properties in extremely thin soi mosfet's. *Electron Devices, IEEE Transactions on*, 45(5):1122 –1126, may 1998.
- [GMCS07] C. Galup-Montoro and M. Cherem-Schneirer. *Mosfet modelling for circuit analysis and design*. World Scientific, 2007.
- [Gra10] W. Grabinski. Gnu/open source cad tools for verilog-a compact model standardization. In *CTCCM Modeling Course*, Tarragona, June 2010. GMC Suisse.
- [GRGG⁺07] F.J. Garcia Ruiz, A. Godoy, F. Gamiz, C. Sampedro, and L. Donetti. A comprehensive study of the corner effects in pi-gate mosfets including quantum effects. *Electron Devices, IEEE Transactions on*, 54(12):3369 –3377, dec. 2007.

-
- [GRLV⁺99] F. J. Gamiz, J. B. Roldán, J. A. Lopez-Villanueva, P. Cartujo-Cassinello, and J. E. Carceller. Surface Roughness at the Si-SiO₂ Interfaces in Fully Depleted Silicon-On-Insulator Inversion Layers. *J. Appl. Phys.*, 86(12):6854–6863, December 1999.
- [Gro05] BSIM3/BSIM4 Device Group. *BSIM3v3.0*. U. Berkeley, California, 2005.
- [GRRB04] E. Gnani, S. Reggiani, M. Rudan, and G. Baccarani. A new approach to the self-consistent solution of the schrodinger-poisson equations in nanowire mosfets. In *Solid-State Device Research conference, 2004. ESSDERC 2004. Proceeding of the 34th European*, pages 177 – 180, sept. 2004.
- [HBEK09] M. Hartmann, J. Biela, H. Ertl, and J.W. Kolar. Wideband current transducer for measuring ac signals with limited dc offset. *Power Electronics, IEEE Transactions on*, 24(7):1776 –1787, july 2009.
- [Hei93] D.E. Hein. Design and operation of spin valve sensors. In *The Magnetic Recording Conference (TMRC)*, 1993.
- [HH69] F. N. Hooge and A. M. H. Hoppenbrouwers. 1/[finite part integral] noise in continuous thin gold films. *Physica*, 45(3):386 – 392, 1969.
- [HN05] Prof. Chenming Hu and Prof. Ali M Niknejad. *BSIMSOIv4.0*. University of California, Berkeley, compact modeling council edition, Nov 2005.
- [HN06] Prof. Chenming Hu and Prof. Ali M Niknejad. *BSIM 4.5.0*. University of California, Berkeley, compact modeling council edition, May 2006.
- [Hon07] Honeywell. HMC1001/1002-HMC1021/1022 (Datasheet), Aug. 2007.
- [IHA⁺08] S. Ikeda, J. Hayakawa, Y. Ashizawa, Y. M. Lee, K. Miura, H. Hasegawa, M. Tsunoda, F. Matsukura, and H. Ohno. Tunnel magnetoresistance of 604diffusion in cofeb/mgo/cofeb pseudo-spin-valves annealed at high temperature. *Applied Physics Letters*, 93(8):082508, 2008.
- [IJR⁺05] B. Iniguez, D. Jimenez, J. Roig, H.A. Hamid, L.F. Marsal, and J. Pallares. Explicit continuous model for long-channel undoped surrounding gate mosfets. *Electron Devices, IEEE Transactions on*, 52(8):1868 – 1873, aug. 2005.
- [ITR11] ITRS. International technology roadmap for semiconductors, 2011.
- [JIS⁺04] D. Jimenez, B. Iniguez, J. Sune, L.F. Marsal, J. Pallares, J. Roig, and D. Flores. Continuous analytic i-v model for surrounding-gate mosfets. *Electron Device Letters, IEEE*, 25(8):571 – 573, aug. 2004.
- [JQN⁺92] B. Jonhson, T. Quarles, A. Newton, D.O. Pederson, and A.Sangiovanni-Vicentelli. *SPICE3: Version 3f User's Manual*. Electronics Research Laboratory, 1992.

- [JSS05] A. Jander, C. Smith, and R. Schneider. Magneto-resistive sensors for nondestructive evaluation. In *12th SPIE International Symposium*, march 2005.
- [JSS09] Albrecht Jander, Carl Smith, and Robert Schneiderb. 10th spie international symposium, nondestructive evaluation for health monitoring and diagnostics. In *Magneto-resistive Sensors for Nondestructive Evaluation*, 2009.
- [Jul75] M. Julliere. Tunneling between ferromagnetic films. *Physics Letters A*, 54(3):225 – 226, 1975.
- [JWY⁺07] He Jin, Bian Wei, Tao Yadong, Yang Shengqi, and Tang Xu. Analytic carrier-based charge and capacitance model for long-channel undoped surrounding-gate mosfets. *Electron Devices, IEEE Transactions on*, 54(6):1478 –1485, june 2007.
- [Kal00] S. Kaliyugavaradan. A linear resistance-to-time converter with high resolution. *Instrumentation and Measurement, IEEE Transactions on*, 49(1):151 –153, feb. 2000.
- [KHC08] Hongki Kang, Jin-Woo Han, and Yang-Kyu Choi. Analytical threshold voltage model for double-gate mosfets with localized charges. *Electron Device Letters, IEEE*, 29(8):927 –930, aug. 2008.
- [Kit86] C. Kittel. *Introduction to Solid State Physics*. Elsevier, 1986.
- [KvPX03] K. B. Klaassen, J. C. L. van Peppen, and X. Xing. Noise in magnetic tunnel junction devices. *Journal of Applied Physics*, 93(10):8573 –8575, may 2003.
- [KZ04] K.S. Kundert and O. Zinke. *The Designers Guide to Verilog-AMS*. Kluwer, 2004.
- [LBS⁺09] D. Lin, G. Brammertz, S. Sioncke, C. Fleischmann, A. Delabie, K. Martens, H. Bender, T. Conard, W.H. Tseng, J.C. Lin, W.E. Wang, K. Temst, A. Vatomme, J. Mitard, M. Caymax, M. Meuris, M. Heyns, and T. Hoffmann. Enabling the high-performance ingaas/ge cmos: a common gate stack solution. In *Electron Devices Meeting (IEDM), 2009 IEEE International*, pages 1 –4, dec. 2009.
- [LCMK03] Laurent Lemaitre, Geoffrey Coram, Colin McAndrew, and Ken Kundert. Extensions to verilog-a to support compact device modeling. In *Extensions to Verilog-A to Support Compact Device Modeling*. Behavioral Modeling and Simulation Conference, 2003.
- [LF01] S.E. Laux and M.V. Fischetti. Monte carlo study of velocity overshoot in switching a 0.1-microncmos inverter. In *Electron Devices Meeting, 1997 IEDM Technical Digest. IEEE International*, pages 877–880, 2001.

-
- [LG06] Laurent Lemaitre and Ben Gu. Adms: Compact model synthesis. In *ADMS: Compact Model Synthesis*. MOS-AK: Compact Modeling for Emerging Technologies, 2006.
- [LGM03] L. Lemaître, W. Grabinski, and C. McAndrew. Compact device modeling using verilog-ams and adms. In *Electron Technol. Internet Journal*, pages 1–5, June 2003.
- [LI08] A Lázaro and B Iñíguez. Rf and noise model of gate-all-around mosfets. *Semiconductor Science and Technology*, 23(7):075022, 2008.
- [LR02] M. Lundstrom and Z. Ren. Essential physics of carrier transport in nanoscale mosfets. *Electron Devices, IEEE Transactions on*, 49(1):133–141, jan 2002.
- [Lun00] M. Lundstrom. *Fundamentals of Carrier Transport*. Cambridge University Press, 2000.
- [LVCCG⁺00] J.A. Lopez-Villanueva, P. Cartujo-Cassinello, F. Gamiz, J. Banqueri, and A.J. Palma. Effects of the inversion-layer centroid on the performance of double-gate mosfets. *Electron Devices, IEEE Transactions on*, 47(1):141–146, jan 2000.
- [MACn⁺09] M. Mujika, S. Arana, E. Castaño, M. Tijero, R. Vilares, J. M. Ruano-Lopez, A. Cruz, L. Sainz, and J. Berganza. Magnetoresistive immunosensor for the detection of escherichia coli o157:h7 including a microfluidic network. *Biosensors & Bioelectronics*, 24(5):1253–1258, 2009.
- [May88] Kartikeya Mayaram. *CODECS: A Mixed-Level Circuit and Device Simulator*. PhD thesis, EECS Department, University of California, Berkeley, December 1988.
- [MBE05] Diego Ramirez Munoz, Silvia Casans Berga, and Candid Reig Escriva. Current loop generated from a generalized impedance converter: A new sensor signal conditioning circuit. *Review of Scientific Instruments*, 76(6):066103, 2005.
- [MCGY07] S. C. Mukhopadhyay, K. Chomsuwan, C. P. Gooneratne, and S. Yamada. A novel needle-type SV-GMR sensor for biomedical applications. *IEEE Sensors Journal*, 7(3-4):401–408, 2007.
- [MCH⁺08] H.J. Mattausch, M. Chan, J. He, H. Koike, M. Miura-Mattausch, T. Nakagawa, Y.J. Park, T. Tsutsumi, and Z. Yu. Development of multigate mosfet models for circuit simulation with a compact modelling partform. In *The 15th International Conference MIXDES 2008*, Pozna, POLAND, June 2008. MIXDES.
- [Mei08] Gerard Meijer. *Smart sensor Systems*. Wiley-Interscience, 2008.

- [Mey71] John E. Meyer. Mos models and circuit simulation. *RCA Review*, 32(1):42–63, Mar 1971.
- [MIJR07] O. Moldovan, B. Iniguez, D. Jimenez, and J. Roig. Analytical charge and capacitance models of undoped cylindrical surrounding-gate mosfets. *Electron Devices, IEEE Transactions on*, 54(1):162–165, Jan. 2007.
- [MJG⁺07] O. Moldovan, D. Jimenez, J.R. Guitart, F.A. Chaves, and B. Iniguez. Explicit analytical charge and capacitance models of undoped double-gate mosfets. *Electron Devices, IEEE Transactions on*, 54(7):1718–1724, July 2007.
- [MKF95] Wolfgang Mathis, Martin Koch, and Uwe Feldmann. Parameter embedding methods for finding dc operating points of transistor circuits. *Proc. NOLTA '95*, pages 1159 – 1164, 1995.
- [MME08] D. Muñoz, J. Moreno, and C. Escrivá. Constant current drive for resistive sensors based on generalized impedance converter. *IEEE Transactions on instrumentation and measurements*, 57(10):2290–2296, 2008.
- [MOT⁺04] Marek Mierzwinski, Patrick O'Halloran, Boris Troyanovsky, Karti Mayaram, and Robert Dutton. New capabilities for verilog-a implementations of compact device models. In *Nano Science and Technology Institute*. Nanotech, 2004.
- [MOT09a] Marek Mierzwinski, Patrick O'Halloran, and Boris Troyanovsky. Practical considerations for developing, debugging, and releasing verilog-a models. In *MOS-AK Workshop*, Baltimore, 2009. Tiburon Design Automation Santa Rosa, CA.
- [MOT09b] Marek Mierzwinski, Patrick O'Halloran, and Boris Troyanovsky. Practical considerations for verilog-a models. In *CMOS Emerging Technologies Conference*, Vancouver, September 2009. Tiburon Design Automation Santa Rosa, CA.
- [MSY11] MSYS. Msys, 2011. A contraction of "Minimal SYStem", is a Bourne Shell command line interpreter system.
- [MW95] K. Mochizuki and K. Watanabe. A linear resistance-to-frequency converter. In *Instrumentation and Measurement Technology Conference, 1995. IMTC/95. Proceedings. Integrating Intelligent Instrumentation and Control'.*, IEEE, page 339, April 1995.
- [MYK⁺11] H.J. Mattaush, A. Yumisaki, A. Kaya, T. Koide, and M. Miura-Mattausch. Application of surface-potential morel hisim2 to variation analysis of cmos technologies. In *The 8th International Workshop on Compact Modeling*, pages 21–26, 2011.
- [Nag75] L. Nagel. *SPICE2: A Computer Program to Simulate Semiconductors Circuits*. Electronics Research Laboratory, May 1975.

-
- [NCB⁺11] A. Nazarov, J.-P. Colinge, F. Balestra, J.-P. Raskin, F. Gamiz, and V.S. Lysenko. *Semiconductor-On-Insulator Materials for Nanoelectronics Applications*. Springer, 2011.
- [ND73] L. W. Nagel and Pederson D.O. *SPICE: A Computer Program to Simulate Semiconductors Circuits*. Electronics Research Laboratory, Apr 1973.
- [New69] Isaac Newton. De analysi per aequationes numero terminorum infinitas. 1, 1669.
- [NVE03] Corp. NVE. AA and AB-Series Analog Sensors (Datasheet), April 2003.
- [PEC83] Y. Ping, B.D. Epler, and P.K. Chatterjee. An investigation of the charge conservation problem for mosfet circuit simulation. *Solid-State Circuits, IEEE Journal of*, 18(1):128 – 138, feb 1983.
- [Ped84] D. O. Pederson. A historical review of circuit simulation. *IEEE Transactions on Circuits and Systems*, 31(1):103 – 111, January 1984.
- [PFW⁺03] Su Pin, S.K.H. Fung, P.W. Wyatt, Wan Hui, A.M. Niknejad, M. Chan, and Hu Chenming. On the body-source built-in potential lowering of soi mosfets. *Electron Device Letters, IEEE*, 24(2):90 – 92, feb. 2003.
- [Phi00] Philips. *Magnetoresistive sensors for magnetic field measurement*. Philips, 2000.
- [PJA99] Darrell A. Teegarden Peter J. Ashenden, Gregory D. Peterson. *The System Designer's Guide to VHDL-AMS*. 1999.
- [PM02] K.L. Pandya and W. McDaniel. A simplified method of generating thermal models for power mosfets. In *Semiconductor Thermal Measurement and Management, 2002. Eighteenth Annual IEEE Symposium*, pages 83 –87, 2002.
- [Pop91] R.S. Popovic. *Hall effect devices*. The Adam Hilger Series on Sensors. Jones, B.E., 1991.
- [PS01] J. Pelegrí-Sebastiá. *Magnetoresistive sensors with spin-valve technology: fabrication, electrical characterization and study of applicability*. Phd. thesis, Universitat de Valencia, 2001.
- [PSB93] M.R. Pinto, E. Sangiorgi, and J. Bude. Silicon mos transconductance scaling into the overshoot regime. *Electron Device Letters, IEEE*, 14(8):375 –378, aug 1993.
- [Qua89a] T. Quarles. Adding devices to spice3. Technical Report UCB/ERL M89/45, EECS Department, University of California, Berkeley, 1989.
- [Qua89b] T. Quarles. The spice3 implementation guide. Technical Report UCB/ERL M89/44, EECS Department, University of California, Berkeley, 1989.
-

- [RCBR08] C. Reig, M.D. Cubells-Beltrán, and D. Ramírez. Electrical isolators based on tunneling magnetoresistance technology. *IEEE Transactions on Magnetism*, 44(11):4011–4014, 2008.
- [RCBR09] C. Reig, M. D. Cubells-Beltrán, and Ramírez. Gmr based electrical current sensors. In A.D. Torres and D.A. Pérez, editors, *Giant Magnetoresistance: New Research*, pages 1–40. Nova Science Publishers, Inc., 2009.
- [RCBRM09] Cándid Reig, María-Dolores Cubells-Beltrán, and Diego Ramírez Muñoz. Magnetic field sensors based on giant magnetoresistance (gmr) technology: Applications in electrical current sensing. *Sensors*, 9(10):7919–7942, 2009.
- [RCR⁺05] D. Ramírez, S. Casans, C. Reig, A. E. Navarro, and J. Sánchez. Build a precise dc floating-current source. *EDN*, 18:83–84, August 2005.
- [RCR⁺09] C. Reig, M.D. Cubells, D. Ramírez, A. Roldán, J.B. Roldán, S. Cardoso, and P.P. Freitas. Gmr current sensors for ic applications. In *Ibernam 2009*, 2009.
- [REMB80] Jennifer A. Robinson, Youssef A. El-Mansy, and A.Roy Boothroyd. A general four-terminal charging-current model for the insulated-gate field-effect transistor–i. *Solid-State Electronics*, 23(5):405 – 410, 1980.
- [RGG08] J.B. Roldan, A. Godoy, F. Gamiz, and M. Balaguer. Modeling the centroid and the inversion charge in cylindrical surrounding gate mosfets, including quantum effects. *Electron Devices, IEEE Transactions on*, 55(1):411 –416, jan. 2008.
- [RGJM⁺10] J.B. Roldán, F. Gámiz, F. Jiménez-Molinos, C. Sampedro, A. Godoy, F.J. García-Ruiz, and N. Rodríguez. An analytical i-v model for surrounding-gate transistors that includes quantum and velocity overshoot effects. *Electron Devices, IEEE Transactions on*, 57(11):2925 –2933, nov. 2010.
- [RGLV⁺98] J.B. Roldan, F. Gamiz, J.A. Lopez-Villanueva, P. Cartujo, and J.E. Carceller. A model for the drain current of deep submicrometer mosfets including electron-velocity overshoot. *Electron Devices, IEEE Transactions on*, 45(10):2249 –2251, oct 1998.
- [RGLVC97] J.B. Roldan, F. Gamiz, J.A. Lopez-Villanueva, and J.E. Carceller. Modeling effects of electron-velocity overshoot in a mosfet. *Electron Devices, IEEE Transactions on*, 44(5):841 –846, may 1997.
- [RGLVCC00] J.B. Roldán, F. Gámiz, J.A. López-Villanueva, and P. Cartujo-Cassinello. Deep submicrometer soi mosfet drain current model including series resistance, self-heating and velocity overshoot effects. *Electron Device Letters, IEEE*, 21(5):239 –241, may 2000.

-
- [RGRR06] J.B. Roldán, F Gámiz, A Roldán, and N Rodríguez. Characterization of electron transport at high fields in silicon-on-insulator devices: a monte carlo study. *Semiconductor Science and Technology*, 21(1):81, 2006.
- [Roy89] J.S. Roychowdhury. Spice3 distortion analysis. Technical Report UCB/ERL M89/48, EECS Department, University of California, Berkeley, 1989.
- [RR10] A. Roldán and J.B. Roldán. Análisis simbólico de circuitos mediante técnicas de análisis modal modificado. In *Tecnologías aplicadas a la enseñanza de la electrónica, IX Congreso TAAE*, Jul 2010.
- [RRCB⁺10] A Roldán, C Reig, M.D. Cubells-Beltrán, J.B. Roldán, D. Ramírez, S Cardoso, and P.P. Freitas. Analytical compact modeling of gmr based current sensors: application to power measurement at the ic level. *Solid State Electronics*, vol 54(8):1606–1612, JUL 2010.
- [RRG05a] A Roldán, J.B. Roldán, and F Gámiz. Characterization and modeling of soi mosfet velocity overshoot: enhancement of spice bsimsoi model. In *First Workshop of the Thematic Network on Silicon on Insulator technology, devices and circuits*, 2005.
- [RRG05b] A. Roldán, J.B. Roldán, and F. Gámiz. Single gate soi mosfet velocity overshoot characterization, modelling and enhancement of spice bsimsoi model. In *EUROSOI meeting 2005*, pages 109–110, Granada, (España), 2005.
- [RRG05c] A. Roldán, J.B. Roldán, and F. Gámiz. Spice bsimsoi enhancement to account for velocity overshoot effects. In *V Conferencia de Dispositivos Electrónicos (CDE)*, page 118, Tarragona, (España), 2005.
- [RRG08a] A. Roldán, J.B. Roldán, and F. Gámiz. A in-depth simulation study of cmos inverters based on the novel surrounding gate transistors. In *International Conference on Advances in Electronics and Micro-electronics ENICS-2008*, 2008.
- [RRG08b] A. Roldán, J.B. Roldán, and F. Gámiz. Simulation of cmos iverters based on the novel sourrounding gate transistor. a verilog-a implementation. In *MOS-AK Edinburgo 2008/ESSDERC/ESSCIRC*, 2008.
- [RRG09] A. Roldán, J.B. Roldán, and F. Gámiz. Simulation study of digital circuits based on nanometric surrounding gate transistors: the role of quantum and velocity overshoot effects. In *MOS-AK Atenas 2009/ESSDERC/ESSCIRC*, 2009.
- [RRGJM10] A.M. Roldán, J.B. Roldán, F. Gámiz, and F. Jiménez-Molinos. Advanced modeling of cylindrical surrounding gate transistors for circuit simulation. In *XXV Conference on Design of Circuits and Integrated Systems DCIS-2010*, pages 402–425, November 2010.
-

- [RRL02] Jung-Hoon Rhew, Zhibin Ren, and Mark S. Lundstrom. A numerical study of ballistic transport in a nanoscale mosfet. *Solid-State Electronics*, 46(11):1899 – 1906, 2002.
- [RRR⁺10] A. Roldán, C. Reig, J.B. Roldán, A. Cano-Abellán, and D. Ramírez. Quasi-static electrical model for magnetoresistive current sensors. In *International Conference on Renewable Energies and Power Quality 2010*, 2010.
- [RRR⁺11a] A. Roldán, C. Reig, J.B. Roldán, A. Cano-Abellán, S. Cardoso, and P.P. Freitas. A current-time conversor circuit based on giant magnetoresistance sensors. a verilog-a implementation. In *MOS-AK Paris 2011*, April 2011.
- [RRR⁺11b] A. Roldán, C. Reig, J.B. Roldán, A. Cano-Abellán, S. Cardoso, and P.P. Freitas. Quasi-static magnetoresistive sensor modeling for current-time conversion circuit applications. In *8th Spanish Conference on electron Devices*, Feb 2011.
- [RRR⁺11c] A. Roldán, C. Reig, J.B. Roldán, M.D. Cubells-Beltrán, S. Cardoso, and P.P. Freitas. Simulation study of ic power measurement applications by means of verilog-a models for gmr current sensors. In *MOS-AK Paris 2011*, April 2011.
- [RRR⁺11d] A. Roldán, C. Reig, J.B. Roldán, M.D. Cubells-Beltrán, S. Cardoso, and P.P. Freitas. A verilog-a implementation of a model for spin-valve based current sensors. In *MOS-AK Sevilla 2010*, Sep 2011.
- [RRR⁺11e] Andrés M. Roldán, Juan B. Roldán, Cándid Reig, M.-D. Cubells-Beltrán, Diego Ramírez, Susana Cardoso, and Paulo P. Freitas. A dc behavioral electrical model for quasi-linear spin-valve devices including thermal effects for circuit simulation. *Microelectronics Journal*, 42(2):365 – 370, 2011.
- [RRS⁺04a] C. Reig, D. Ramírez, F. Silva, J. Bernardo, and P. Freitas. Design, fabrication, and analysis of a spin-valve based current sensor. *Sensors and Actuators A-Physical*, 115(2-3):259–266, 2004.
- [RRS04b] C. Reig, D. Ramírez, and F. Silva. Design, fabrication, and analysis of a spin-valve based current sensor. *Sensors and Actuators A-Physical*, 115(2-3):259–266, 2004.
- [RVD⁺00] Zhibin Ren, R. Venugopal, S. Datta, M. Lundstrom, D. Jovanovic, and J. Fossum. The ballistic nanotransistor: a simulation study. In *Electron Devices Meeting, 2000. IEDM Technical Digest. International*, pages 715 –718, 2000.
- [RVDL01] Z. Ren, R. Venugopal, S. Datta, and M. Lundstrom. Examination of design and manufacturing issues in a 10 nm double gate mosfet using nonequilibrium green’s function simulation. In *Electron Devices Meeting, 2001. IEDM Technical Digest. International*, pages 5.4.1 –5.4.4, 2001.

-
- [SAB⁺06] N. Singh, A. Agarwal, L.K. Bera, T.Y. Liow, R. Yang, S.C. Rustagi, C.H. Tung, R. Kumar, G.Q. Lo, N. Balasubramanian, and D.-L. Kwong. High-performance fully depleted silicon nanowire (diameter le; 5 nm) gate-all-around cmos devices. *Electron Device Letters, IEEE*, 27(5):383 – 386, may 2006.
- [SB06] Karsten Sternickel and Alex I Braginski. Biomagnetism using squids: status and perspectives. *Superconductor Science and Technology*, 19(3):S160, 2006.
- [SHWK⁺88] G.A. Sai-Halasz, M.R. Wordeman, D.P. Kern, S. Rishton, and E. Ganin. High transconductance and velocity overshoot in nmos devices at the 0.1- μ m gate-length level. *Electron Device Letters, IEEE*, 9(9):464 –466, sep 1988.
- [Sim63] John G. Simmons. Generalized formula for the electric tunnel effect between similar electrodes separated by a thin insulating film. *Journal of Applied Physics*, 34(6):1793–1803, 1963.
- [Sko11] T. Skonicki. Tutorial at the soi conference. In *IEEE International SOI Conference*, 2011.
- [SKP⁺05] Jean-Michel Sallese, François Krummenacher, Fabien Prégaldiny, Christophe Lallement, A. Roy, and C. Enz. A design oriented charge-based current model for symmetric dg mosfet and its correlation with the ekv formalism. *Solid-State Electronics*, 49(3):485 – 489, 2005.
- [SL01] M. Solomon and S.E. Laux. The ballistic fet: design, capacitance and speed limit. In *Electron Devices Meeting, 2001 IEDM Technical Digest. IEEE International*, pages 95–98, 2001.
- [SMGGR06] C. Sampedro-Matarin, F. Gamiz, A. Godoy, and F.J.G. Ruiz. The multivalley effective conduction band-edge method for monte carlo simulation of nanoscale structures. *Electron Devices, IEEE Transactions on*, 53(11):2703 –2710, nov. 2006.
- [SS98] A. Sedra and K. Smith. *Microelectronics Circuits 4th. Ed.* Oxford University Press, 1998.
- [Su02] P. Su. *An International Standard Model for SOI Circuit Design*. PhD thesis, EECS Department, University of California, Berkeley, 2002.
- [Sub11] Team Subversion. Subversion, 2011. Now subversion is called subversion.apache.org.
- [SYWD80] Oh Soo-Young, D.E. Ward, and R.W. Dutton. Transient analysis of mos transistors. *Electron Devices, IEEE Transactions on*, 27(8):1571 – 1578, aug 1980.

- [TDC⁺08] A. Tsormpatzoglou, C.A. Dimitriadis, R. Clerc, G. Pananakakis, and G. Ghi-
baudo. Threshold voltage model for short-channel undoped symmetrical
double-gate mosfets. *Electron Devices, IEEE Transactions on*, 55(9):2512 –
2516, sept. 2008.
- [TF05] V.P. Trivedi and J.G. Fossum. Quantum-mechanical effects on the thresh-
old voltage of undoped double-gate mosfets. *Electron Device Letters, IEEE*,
26(8):579 – 582, aug. 2005.
- [TFG04] V.P. Trivedi, J.G. Fossum, and F. Gamiz. A compact qm-based mobility model
for nanoscale ultra-thin-body cmos devices. In *Electron Devices Meeting, 2004. IEDM Technical Digest. IEEE International*, pages 763 – 766, dec. 2004.
- [Tho80] K. K. Thornber. Relation of drift velocity to low field mobility and high field
saturation velocity. *Journal of Applied Physics*, 51(4):2127–2136, 1980.
- [Tor10] José Manuel García Torres. Pfc: Implementación de películas magnetorresis-
tivas co-ag electrodepositadas en dispositivos microelectrónicos, Oct 2010.
- [Tsi03] Yannis Tsvividis. *Operation and Modeling of the MOS Transistor. 2nd Ed.*
Oxford University PRes, Columbia University, 2003.
- [UiT03] Ken Uchida and Shin ichi Takagi. Carrier scattering induced by thickness
fluctuation of silicon-on-insulator film in ultrathin-body metal oxide semicon-
ductor field-effect transistors. *Applied Physics Letters*, 82(17):2916–2918, 2003.
- [Van94] L.K.J. Vandamme. Noise as a diagnostic tool for quality and reliability of
electronic devices. *Electron Devices, IEEE Transactions on*, 41(11):2176 –
2187, nov 1994.
- [VPK05] M. Vopalensky, A. Platil, and P. Kaspar. Wattmeter with AMR sensor. *Sensors
and Actuators A-Physical*, 123(24):303–307, 2005.
- [WACF07] P. Wisniowski, J. Almeida, S. Cardoso, and P.P. Freitas. Effect of cofeb thick-
ness and shape anisotropy on magnetic response of free layer in mgo magnetic
tunnel junctions. In *52nd Conference on Magnetism and Magnetic Materials*,
page 339, November 2007.
- [WD78] D.E. Ward and R.W. Dutton. A charge-oriented model for mos transistor
capacitances. *Solid-State Circuits, IEEE Journal of*, 13(5):703 – 708, oct 1978.
- [WME⁺05] J Watts, C. McAndrew, C. Enz, C. Galup-Montoro, G. Gildenblat, C. Hu,
Langenvelde R., M. Miura-Mattausch, R. Rios, and C. Sah. Advanced com-
pact models for mosfets. In *Eighth International Conference on Modeling and
Simulation of Microsystems*, 2005.

-
- [WTK07] Reinout Woltjer, Luuk Tiemeijer, and Dick Klaassen. An industrial view on compact modeling. *Solid-State Electronics*, 51(11-12):1572 – 1580, 2007. Special Issue: Papers Selected from the 36th European Solid-State Device Research Conference - ESSDERC'06.
- [Xin04] Zhou Xing. Unified approach to mos transistor compact modeling. Technical report, Nanyang Technological University, Singapore, 2004.
- [XLN⁺06] Haiwen Xi, Jay Loven, Robert Netzer, Javier I Guzman, Scott Franzen, and Sining Mao. Thermal fluctuation of magnetization and random telegraph noise in magnetoresistive nanostructures. *Journal of Physics D: Applied Physics*, 39(10):2024, 2006.
- [XWJ⁺09] Li Xin, Wu Weimin, A. Jha, G. Gildenblat, R. Langevelde, G.D.J. Smit, A.J. Scholten, D.B.M. Klaassen, C.C. McAndrew, J. Watts, C.M. Olsen, G.J. Coram, S. Chaudhry, and J. Victory. Benchmark tests for mosfet compact models with application to the psp model. *Electron Devices, IEEE Transactions on*, 56(2):243 –251, feb. 2009.
- [XYH⁺08] Liang Xu, Heng Yu, Shu-Jen Han, S. Osterfeld, R.L. White, N. Pourmand, and S.X. Wang. Giant magnetoresistive sensors for dna microarray. *Magnetics, IEEE Transactions on*, 44(11):3989 –3991, Nov 2008.
- [YWY⁺08] Bo Yu, Lingquan Wang, Yu Yuan, P.M. Asbeck, and Yuan Taur. Scaling of nanowire transistors. *Electron Devices, IEEE Transactions on*, 55(11):2846 –2858, nov. 2008.
- [Zet06] Semiconductors Zetex. ZMC10 Current Sensor (Datasheet), July 2006 2006.

ACRÓNIMOS

AC	Altern Current. Corriente Alterna.	47, 48, 92, 171, 186, 247
AMR	Anisotropic Magnetoresistance	10, 11, 18, 127, 130, 131, 133, 134, 142, 143, 164, 166– 168, 191, 257
BICMOS	Bipolar and Complementary MOS. Tecnología MOS Complementaria y Bipolar	20
BJT	Bipolar Junction Transistor	245
BSIM	Berkeley Short-channel IGFET Model	26, 40

CANCER	Computer Analysis of Nonlinear Circuits, Excluding Radiation	28
CIP	Current in magnetoresistance layer 's plane	10, 135, 136
CM	Compact Model. Modelo Compacto.	22
CML	Compiled module library	7, 11, 49, 50, 146
CMOS	Complementary MOS. Tecnología MOS Complementaria	9, 10, 19, 22, 80, 89, 104–109, 121–127, 256
CPP	Current perpendicular to magnetoresistance layer 's plane	10, 135–137
DC	Direct Current. Corriente Continua.	55, 92, 99, 166, 171, 173, 175, 178, 186, 187, 194, 195, 247, 253
DG-MOSFET	Double Gate Metal Oxide Semiconductor Field Effect Transistor.	5, 9, 10, 111, 112, 115, 116, 118–121
DSP	Digital Signal Processor. Procesador digital de señales	20

ESPICE	SPICE en Español	60, 63, 66, 71
FDSOI	Fully Depleted SOI Transistor. Transistor SOI competamente Deplexionado	84
FIC	First element In each Column	34
FIR	First element In each Row	34
FORTTRAN	Formula Translating System.	26, 52
GAA	Gate All Around transistor.	85
GCA	Gradual channel approximation	115, 116
GIC	Generalized Impedance Converter	13, 184, 187, 189
GMR	Gigant Magnetoresistance	10, 18, 22, 125– 127, 134– 138, 142, 143, 146, 147, 159, 161, 164, 167, 168, 191, 195, 257, 258
GPIB	General Purpose Interface Bus	193, 195
IBD	Ion Beam Deposition	149, 212
IC	Integrated Circuit.	195
ISPICE	Interactive Simulated Program Integrated Circuit Emphasis	29

ITRS	International Technology Roadmap for Semiconductors	86
JCR	Journal of Citation Reports	24
LED	Light-Emitting Diode	64
LRM	Language Reference Manual.	27, 283
MATLAB	MATrix LABoratory.	21, 44, 63
MCG	Magnetocardiography	158
MEG	Magnetoencephalography	158
MEMS	Micro Electro-Mechanical Systems. Microsistemas Electro-Mecánicos.	125
MOSFET	Metal Oxide Semiconductor Field Effect Transistor.	7, 9, 19, 32, 33, 38, 40, 78, 79, 83–85, 87–90, 100, 112
MTJ	Magnetic Tunnel Junction	10–14, 16, 18, 137–140, 142, 143, 158–161, 167–170, 172–176, 201–204, 207–211, 214, 257, 258

N-MOS	Dispositivo MOS tipo N	15, 105– 107
netlist	descripción topológica de un circuito.	41
NVE	Non Volatile Electronics	127
OP	Operation Point Analysis.	42, 283
P-MOS	Dispositivo MOS tipo P	105– 107
PCB	Printed Circuit Boards	127, 165, 181, 183, 188
PDSOI	Partially Depleted SOI transistor. Transistor SOI parcialmente Deplexionado	8, 72
RF	Radio Frequency. Radiofrecuencia	19, 20, 100
RIE	Reactive Ion Etching	212– 215
RTE	Run-time environment	50
RTN	Random Telegraph Noise	168, 169
SAF	Synthetic antiferromagnetic	138, 139
SCE	Short Channel Effects. Efectos de Canal Corto	76, 83, 113
SG-MOSFET	Single Gate Metal Oxide Semiconductor Field Effect Transistor.	5, 9, 112– 115

SGT	Surrounding Gate Transistor	5, 9, 15, 23, 76, 85, 91–94, 96, 97, 99, 100, 102– 108, 110– 112, 117– 121, 256
SMD	Surface Mount Devices	181, 183, 188
SMR	Semiconductor Magnetoresistance	127
SNR	Signal to noise ratio	167
SoC	System on a Chip	125
SOI	Silicon On Insulator. Tecnología de Silicio sobre Aislante.	8, 22, 76, 78, 79, 89, 112, 113, 115, 127, 256
SPICE	Simulation Program with Integrated Circuit Emphasis	25, 26, 29, 38, 41, 46, 48, 52, 54, 62, 66, 287, 288

SV	Spin Valve	11–13, 17, 18, 137, 138, 146– 154, 156, 157, 167, 170– 174, 191– 193, 257, 258
TMR	Tunnel Magnetoresistance Resistance	158, 167, 201, 208
VLSI	Very Large Scale of Integration	29
WNB	White Noise Background	167– 169

GLOSARIO

(*desc=...*) extensión del [LRM](#) que exporta el valor de la variable que se define con esa primitiva en el módulo.. [42](#)

\$simparam extensión del [LRM](#) que permite acceder a los valores internos del simulador.. [42](#)

\$temperature extensión del [LRM](#) que devuelve el valor de la temperatura de simulación .. [42](#)

\$vt extensión del [LRM que devuelve el valor \$v_t = \frac{K*T}{ELECTRONCHARGE}\$.. \[42\]\(#\)](#)

___VAMS_COMPACT_MODELING___ primitiva del compilador de Verilog-AMS.. [42](#)

OP tipo de análisis que devuelve el valor de las tensiones, corrientes y transconductancias y capacidades de los dispositivos en régimen estacionario.. [42](#)

<+ primitiva de Verilog-A para definir las contribuciones a las corrientes internas del modelo.. [42](#)

0805 Tamaño de dispositivo 0805 mils (2013 métrico): 0.08" × 0.05" (2.0 mm × 1.25 mm). Tamaño para resistencias de potencia típica de 1/10 o 1/8 W. [181](#), [183](#), [188](#)

Advanced Design System [EDA](#) de simulación de circuitos de Agilent™. [9](#), [10](#), [12](#), [22](#), [105–109](#), [121–123](#), [150](#), [165](#), [171](#), [172](#), [174–176](#), [179](#), [183](#), [188](#), [194](#), [195](#)

Altium [EDA](#) de diseño de placas de circuito impreso de Altium Limited™. [181](#), [183](#), [188](#)

analog primitiva de Verilog-A para definir las funciones en el interior de un módulo.. [42](#)

API Una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas comúnmente *librerías*).. [61](#)

begin primitiva de Verilog-A para definir el comienzo de las funciones en el interior de un módulo analógico.. [42](#)

BSD La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.. [63](#), [66](#)

CAD Uso de computadores en el proceso de diseño y documentación de un bien o servicio. [20–22](#), [28](#)

CIDER es un simulador de dispositivos integrado con un simulador de circuitos que permite enlazar directamente los parámetros tecnológicos de fabricación y el circuito donde se usa dicho dispositivo. La precisión de los resultados obtenidos es considerablemente mejor que los proporcionados por un modelo compacto del dispositivo. [29](#), [62](#)

CMRR La razón de rechazo al modo común o CMRR es uno de los parámetros de un amplificador operacional u OAMP. En un amplificador operacional cuando las tensión $V_{in} = V^+ - V^- = 0$ existe una pequeña tensión de salida cuando lo ideal sería que esta fuera cero. La CMRR es una medida del rechazo que ofrece la configuración a la entrada de voltaje común. Se suele expresar en decibelios.. [188](#)

coercitividad magnética La coercitividad magnética representa la resistencia de un material a ser magnetizado. Se mide en Oersteds.. [131](#)

ddt primitiva de Verilog-A para calcular la derivada simbólica de una expresión con respecto al tiempo.. [42](#)

ddx primitiva de Verilog-A para calcular la derivada simbólica de una expresión con respecto a una variable.. [43](#)

EDA Plataforma de herramientas software para el diseño electrónico. [26](#), [283](#), [285](#)

EKV El modelo para EKV es un modelo matemático para transistores MOSFET orientado a la simulación de circuitos y el diseño analógico. Fue desarrollado en 1995 por los profesores C. C. Enz, F. Krummenacher y E. A. Vittoz. El modelo es muy preciso en la zona sub-umbral (e.g. $V_{bulk} = V_{source}$ y $V_{gate} - V_{source} < V_{Threshold}$). Además incorpora los efectos presentes en los dispositivos submicra. [26](#), [89](#), [90](#)

ELDO [EDA](#) de simulación de circuitos de Mentor Graphics™. [22](#), [195](#)

end primitiva de Verilog-A para definir el final de las funciones en el interior de un módulo analógico.. [42](#)

flicker_noise primitiva de Verilog-A para calcular la contribución de ruido flicker o de contribución espectral $\frac{1}{f}$ en variable circuital de un módulo.. [43](#)

FORTRAN es un lenguaje de programación de alto nivel y procedural, desarrollado para propósitos generales por IBM en 1957 para el equipo IBM 704. Fue el primero desarrollado con estas características. Está fuertemente orientado al cálculo y por ende es uno de los de mayor eficiencia en la ejecución.. [26](#), [33](#), [287](#)

FW Aplicación lógica que permite el control de los componentes físicos de un producto electrónico. [21](#)

GNU/Linux GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o *kernel* libre similar a *Unix* denominado *Linux*, que es usado con herramientas de sistema *GNU*. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU, en inglés: General Public License) y otra serie de licencias libres.. [8](#), [62](#), [65](#), [66](#), [71](#), [286](#)

GPL La Licencia Pública General de *GNU* o más conocida por su nombre en inglés *GNU General Public License* o simplemente sus siglas del inglés *GNU GPL*, es una licencia creada por la *Free Software Foundation* en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.. [66](#), [286](#)

GTK+ Librería multiplataforma para creación de interfaces gráficos de usuarios. Dispone de un conjunto completo de *widgets* <http://www.gtk.org/>. [63](#), [65](#)

HiSIM Modelo basado en potencial de superficie para transistores ultra-small MOS resultado de la colaboración entre el Semiconductor Technology Academic Research Center (STARC) y el laboratorio del Prof. Michiko Miura-Mattausch. [26](#), [89](#)

HW Conjunto formado por todas los componentes físicos de un producto electrónico. [21](#), [60](#)

IC Circuito compuesto básicamente de elementos semiconductores y pasivos que se fabrica en la superficie de una oblea semiconductor y posteriormente es encapsulado. Componen la mayoría de los productos electrónicos del mercado. [19–22](#), [25](#), [29](#), [125](#), [126](#)

inout primitiva de definición de las características de un puerto.. [41](#)

KDevelop KDevelop es un entorno de desarrollo integrado para sistemas [GNU/Linux®](#) y otros sistemas Unix, publicado bajo licencia [Licencia Pública General](#), orientado al uso bajo el entorno gráfico KDE, aunque también funciona con otros entornos. A diferencia de muchas otras interfaces de desarrollo, KDevelop no cuenta con un compilador propio, por lo que depende de *gcc* para producir código binario.. [71](#)

layout Representación de un circuito integrado en forma de geometrías planares correspondientes a las deposiciones de metal, óxidos o capas semiconductoras que forman los elementos internos del IC. [21](#)

lenguaje-C Es un lenguaje de programación de propósito general desarrollado por Dennis Ritchie en 1972 en los Bell Telephone Laboratories para usarlo en el sistema operativo UNIX®.. [26](#), [27](#), [33](#), [34](#), [37–40](#), [42](#), [43](#), [48](#), [50](#)

limexp función auxiliar de Verilog-A utilizada para evitar los desbordamientos en la evaluación de funciones exponenciales.. [42](#)

LU La factorización o descomposición LU (del inglés Lower-Upper) es una forma de factorización de una matriz como el producto de una matriz triangular inferior y una superior. Debido a la inestabilidad de este método, por ejemplo si un elemento de la diagonal es cero, es necesario premultiplicar la matriz por una matriz de permutación. Método llamado factorización $PA = LU$ o LU con pivote. Esta descomposición se usa en el análisis numérico para resolver sistemas de ecuaciones (más eficientemente) o encontrar las matrices inversas. [29](#)

magnetostricción Se denomina magnetostricción a la propiedad de los materiales magnéticos que hace que éstos cambien de forma al encontrarse en presencia de un campo magnético.. [131](#)

MNA Técnica de simulación de circuitos que permite el establecimiento del sistema de ecuaciones que representa al circuito usando como incógnitas del problema las tensiones en los nodos del circuito. [28](#), [230](#)

Newton-Raphson Método numérico denominado de Newton [[New69](#)]. (conocido también como el método de Newton-Raphson o el método de Newton-Fourier) que implementa un algoritmo eficiente para encontrar aproximaciones de los ceros o raíces de una función real. También puede ser usado para encontrar el máximo o mínimo de una función, encontrando los ceros de su primera derivada.. [27](#), [29](#), [39](#), [45](#), [48](#), [49](#)

NG-SPICE Versión de [Simulation Program with Integrated Circuit Emphasis \(SPICE\)](#) (Next Generation Spice) disponible en <http://ngspice.sourceforge.net/>. [62](#), [66](#)

OAmp Un amplificador operacional es un bloque de amplio uso en el diseño analógico formado por un amplificador de tensión de alta ganancia acoplado en DC a la entrada diferencial y con salida simple. La tensión que genera a la salida es cientos de miles de veces mayor que la diferencia de tensión entre los terminales de la entrada.. [178](#)

OpenGL OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en CAD, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compete con Direct3D en plataformas Microsoft Windows.. [63](#)

PSP El modelo PSO ha sido desarrollado conjuntamente por NXP Research (anteriormente Philips Research) y el grupo de investigación del Prof. Gildenblat de la Arizona State University (anteriormente Penn State University). Adecuado para simulaciones digitales, analógicas y RF; para tecnologías CMOS bulk actuales y futuras, basado en características físicas, conjuga las mejores propiedades del modelo SP (Penn State University) y MOS Model 11 (NXP), un tiempo de simulación y número de parámetros similar a los del MOS Model 11 y dispone de unos procedimientos simples de extracción de parámetros. [26](#), [89](#)

self-heating Proceso de calentamiento debido a la disipación interna del dispositivo. [40](#)

SoC Circuito que integra todos los componentes dentro de un chip. Todas las partes analógicas, digitales, mixtas e incluso las de radio frecuencia se fabrican en el sustrato de una oblea semiconductor con la posibilidad de usar tecnologías de fabricación diferentes en cada bloque. [19](#), [127](#)

SPICE+ Versión de [SPICE](#) disponible en <http://spiceplus.sourceforge.net/> que introduce un interfaz de usuario gráfico, editor de esquemáticos, editor de modelos, ficheros de descripción de esquemas, modelos y símbolos codificados en XML, etc.. [62](#)

SPICE2 Versión de [SPICE](#) codificada en lenguaje [FORTRAN](#). [34](#), [234](#)

SPICE3 Versión de [SPICE](#) codificada en lenguaje C. [34](#), [58](#), [230](#), [234](#), [237](#)

time-to-market Espacio de tiempo que transcurre desde que se generan las especificaciones de un producto hasta que éste esté disponible en las tiendas. [20](#)

Verilog-A Lenguaje de modelado compacto analógico. [7](#), [22](#), [23](#), [26](#), [27](#), [31–33](#), [39–50](#), [82](#), [91](#), [92](#), [104](#), [111](#), [120](#), [146](#), [154](#), [156](#), [157](#), [166](#), [171–174](#), [180](#), [187](#), [188](#), [195](#), [255–258](#)

Verilog-AMS Lenguaje de modelado derivado del lenguaje de descripción hardware Verilog que incluye extensiones analógicas y mixtas (AMS) para permitir el modelado de estos sistemas. Extiende los simuladores basados en eventos de VHDL en simuladores de tiempo continuo que deben resolver ecuaciones diferenciales en el dominio analógico del tiempo. La simulación analógica y digital quedan acoplados en un mismo simulador.. [26](#), [27](#)

white_noise primitiva de Verilog-A para calcular la contribución de ruido blanco en variable circuital de un módulo.. [43](#)

Windows Sistema operativo realizado por Microsoft™. [4](#), [58](#), [60–62](#), [65](#), [66](#), [71](#)

XSPICE es una extensión del simulador [SPICE](#) que permite la inclusión de nuevos modelos y técnicas de modelado con código. La librería de modelos codificados de XSPICE contiene unos 40 bloques que incluyen sumadores, multimplicadores, integradores, modelos magnéticos limitadores, funciones de transferencia en el dominio de Laplace, puertas digitales, elementos de almacenamiento digital y una máquina de estados generalizada. XSPICE fue desarrollado por Fred Cox, Bill Kuhn, y un grupo de colaboradores del *Georgia Tech Research Institute*, dentro del *Georgia Institute of Technology*. [29](#), [62](#), [69](#)