

**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA**

**ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA DE  
TELECOMUNICACIÓN**



**UNIVERSIDAD DE LAS PALMAS  
DE GRAN CANARIA**

**PROYECTO FIN DE CARRERA**

**SIMULACIÓN ELÉCTRICA DE DISPOSITIVOS DE  
HETEROESTRUCTURAS MEDIANTE EL SIMULADOR ELÉCTRICO  
APLAC**

**NAUZET QUINTANA RAMÍREZ**

Las Palmas de Gran Canaria, 2001

**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA**

**ESCUELA UNIVERSITARIA DE  
INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN**



**PROYECTO FIN DE CARRERA**

*Simulación Eléctrica de Dispositivos de Heteroestructuras  
Mediante el Simulador Eléctrico APLAC*

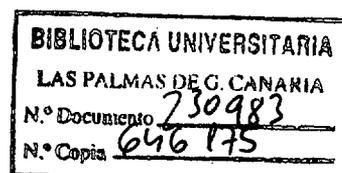
**ESPECIALIDAD : Telefonía y Transmisión de Datos**

**TUTOR/ES : Francisco Javier del Pino Suárez**

**Antonio Hernández Ballester**

**AUTOR : Nauzet Quintana Ramírez**

**FECHA : Octubre 2001**



**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA**

**ESCUELA UNIVERSITARIA DE  
INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN**



**PROYECTO FIN DE CARRERA**

*Simulación Eléctrica de Dispositivos de Heteroestructuras  
Mediante el Simulador Eléctrico APLAC*

Presidente:

Secretario:

Vocal:

Tutores:

Autor:

**NOTA : SOBRESALIENTE, 10 (MATERIAS DE HONOR)**

**ESPECIALIDAD :**

**TUTOR/ES :**

**AUTOR :**

**FECHA :**

<b>PETICIONARIO.....</b>	<b>I</b>
<b>ANTECEDENTES .....</b>	<b>III</b>
<b>OBJETO DEL TRABAJO .....</b>	<b>V</b>
<b>RESUMEN .....</b>	<b>VII</b>
<b>CAPITULO 1.....</b>	<b>7</b>
<b>INTRODUCCIÓN.....</b>	<b>7</b>
1.1 ¿QUÉ ES APLAC?.....	8
1.1.1 Una Pequeña Descripción de APLAC.....	10
1.1.1.1 Modos de Análisis.....	10
1.1.1.2 Análisis de Sistemas.....	10
1.1.1.3 Análisis Electrotérmicos .....	11
1.1.1.4 Modelos de APLAC.....	11
1.1.1.5 Sweep.....	11
1.1.1.6 El lenguaje APLAC .....	12
1.1.1.7 Salida.....	12
1.1.1.8 Optimización.....	12
1.1.2 Modelado.....	13
1.1.3 Opciones de APLAC.....	13
1.1.3.1 Opciones RF.....	14
1.1.3.2 Opciones de Sistema.....	14
1.1.3.3 Opciones Electromagnéticas.....	14
<b>CAPÍTULO 2.....</b>	<b>15</b>
<b>CARACTERÍSTICAS GENERALES DEL SIMULADOR ELÉCTRICO</b>	
<b>APLAC.....</b>	<b>15</b>
2.1 PRINCIPIOS BÁSICOS .....	15
2.2 PRIMER EJEMPLO.....	17
2.2.1 Realización de un Circuito RC.....	18
2.2.2 Realización de la Simulación.....	21
2.3 ARCHIVO DE ENTRADA DEL SIMULADOR APLAC.....	23
2.4 CONTROLES DEL SIMULADOR APLAC .....	25
2.4.1 Menú de fichero (Menú File).....	25
2.4.2 Menú de Instalación (Menú Setup).....	26
2.4.3 Menú de Herramientas (Menú Tools).....	27
2.5 CONTROLES DE LA VENTANA GRÁFICA DE APLAC.....	28
2.5.1 Menú de Opciones (Options Menu).....	29

## Contenido

---

2.5.2	Menú Editar (Edit menu)	29
2.5.3	Menú de ficheros (File menu)	30
2.6	LENGUAJE APLAC	30
2.6.1	Introducción con Dos Simples Ejemplos	32
2.6.2	Estructura del Archivo de Entrada	35
2.6.3	Propiedades del Lenguaje de Programación	36
2.6.3.1	Variables	37
2.6.3.1.1	Variables Constantes y Normales	37
2.6.3.1.2	Variables Funcionales	39
2.6.3.2	Sweep	40
2.6.3.3	Funciones Matemáticas	42
2.6.3.3.1	Funciones Matemáticas Definidas por APLAC	42
2.6.3.3.2	Realización de Nuevas Funciones	44
2.6.3.4	Declaraciones del Preprocesador	45
2.6.4	Modelado	46
2.6.4.1	Introducción	46
2.6.4.2	Creación de Modelos	47
2.6.4.3	Creación de Modelos Lineales	48
2.6.4.3.1	Ejemplo de un Transformador Híbrido	48
2.6.4.3.2	Fuentes Dependientes	49
2.6.4.3.3	Jerarquía de Modelos	49
2.6.4.4	Creación de Modelos No Lineales	50
2.7.5.4.1	Ejemplo de un Modelo de Diodo	50
2.7.5.4.2	Parámetros de Temperatura	52
2.7.5.4.3	Creación de Modelos de Carga No Lineal	52
2.7.5.4.3.1	Carga con una Tensión de Control	52
2.7.5.4.3.1	Carga con Varias Tensiones de Control	53
<b>CAPÍTULO 3</b>		<b>55</b>
<b>ESTRUCTURA Y PRINCIPIO DE FUNCIONAMIENTO DEL HFET</b>		<b>55</b>
3.1	PERFIL DE UNA HETEROESTRUCTURA	56
3.2	TRANSISTORES DE EFECTO DE CAMPO DE HETEROESTRUCTURA	58
3.2.1	Estructura de Bandas y Comportamiento del Dispositivo	59
3.2.1.1	Estructura convencional de los HFETs de $n^+$ -AlGaAs/GaAs	59
3.2.1.2	Principio de funcionamiento	60
3.2.1.3	Consideraciones tecnológicas	61
3.2.2	Familias de los dispositivos HFETs	63
3.2.3	Aplicaciones de los HFETs	64

<b>CAPÍTULO 4</b> .....	<b>67</b>
<b>MODELOS DEL HFET</b> .....	<b>67</b>
4.1 MODELO DE ANGELOV .....	68
4.1.1 Introducción.....	68
4.1.2 El Modelo .....	69
4.2 MODELO DEL FHGIAF.....	71
4.2.1 Adaptación del Modelo FhGIAF a APLAC .....	74
4.2.1.1 Modelo en Gran Señal.....	75
4.2.1.3 Modelo en Régimen Transitorio .....	77
4.2.1.4 Modelo en Pequeña Señal .....	81
4.2.1.5 Lista de Parámetros .....	82
<b>CAPITULO 5</b> .....	<b>85</b>
<b>IMPLEMENTACIÓN DEL MODELO DEL HFET EN APLAC</b> .....	<b>85</b>
5.1 ELEMENTOS BÁSICOS PARA MODELAR DISPOSITIVOS EN APLAC .....	86
5.1.1 Descripción de VCCS (Voltage-Controlled Current Source) .....	86
5.1.1.1 Declaración y Parámetros de la Fuente de Corriente Controlada por Tensión.....	86
5.1.1.2 Descripción del Modelo .....	89
5.1.1.3 Conductancia y Capacitancia Lineales .....	90
5.1.1.4 Valor de una Fuente Lineal.....	91
5.1.1.5 Valor de una Fuente No Lineal.....	91
5.1.2 Descripción del Estamento Function.....	92
5.1.3 Descripción de los Estamentos IF-THEN-ELSE y IFTE.....	93
5.1.4 Descripción del Bloque DefModel ... EndModel.....	93
5.2 MODELADO DEL HFET EN CONTINUA .....	96
5.2.1 Introducción de los Parámetros (HFET de enriquecimiento).....	96
5.2.2 Implementación de los Diodos.....	98
5.2.3 Implementación de la Fuente de Corriente.....	99
5.2.4 Creación del Modelo del HFET en Continua .....	101
5.3 CREACIÓN DEL HFET EN RÉGIMEN DINÁMICO .....	104
5.4 MODELO COMPLETO DEL HFET DE ENRIQUECIMIENTO .....	109
5.5 ERRORES TÍPICOS .....	113
5.6 CREACIÓN DEL MODELO DEL HFET CON EL EDITOR DE ESQUEMAS DE APLAC .....	114
5.6.1 Pasos Necesarios para Crear el Submodelo.....	114
<b>CAPÍTULO 6</b> .....	<b>121</b>

<b>VALIDACIÓN DE RESULTADOS Y CONCLUSIONES .....</b>	<b>121</b>
6.1 VALIDACIÓN DE LOS RESULTADOS OBTENIDOS .....	122
6.1.1 Análisis en DC .....	122
6.1.2 Análisis en Régimen Transitorio .....	125
6.1.3 Análisis en ac.....	132
6.2 CONCLUSIONES FINALES.....	134
<b>ANEXOS.....</b>	<b>137</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>115</b>
<b>PLIEGO DE CONDICIONES.....</b>	<b>119</b>
PROTECCIÓN LEGAL .....	119
Propiedad intelectual .....	119
Titularidad del proyecto .....	120
Invenciones universitarias y de organismos públicos de investigación. ....	121
<b>PRESUPUESTO .....</b>	<b>123</b>

## ***Agradecimientos***

A mi padre por su persistencia y paciencia, a mi madre por su comprensión y ambos por su apoyo incondicional. Por su puesto también a mis hermanos y a Chiqui, por soportarme.

El peticionario de este proyecto fin de carrera es el laboratorio de Dispositivos Optoelectrónicos adscrito al Departamento de Ingeniería Electrónica y Automática de la Universidad de las Palmas de Gran Canaria.

## Antecedentes

---

El modelado de nuevos dispositivos mediante simuladores siempre ha estado bajo un constante estudio debido a las ventajas que representa crear y simular nuevos componentes.

Por lo general el modelado de un nuevo componente requiere un simulador como SPICE3 de Berkeley y un tedioso trabajo generando códigos normalmente en lenguaje C.

Actualmente con la aparición de una nueva generación de simuladores como APLAC cuya meta principal ha sido dar al usuario la libertad de resolver problemas de diseño sin las restricciones de los simuladores convencionales, como SPICE, ha permitido que el modelado de nuevos componentes sea un trabajo menos tedioso. Aún así los usuarios deben familiarizarse con el lenguaje de programación del simulador que es quien permite una correcta implementación de un nuevo componente.

Por otro lado, el desarrollo de las técnicas MBE (*molecular beam epitaxial*) y MOCVD (*metal-organic chemical vapor deposition*) desde finales de los años 70, ha revolucionado el diseño de nuevos dispositivos electrónicos y optoelectrónicos. En nuestros días se pueden crecer capas muy finas de material semiconductor manteniendo un fino control sobre el dopaje y las conexiones. Este desarrollo tecnológico ha propiciado la consolidación de una disciplina científica: la "ingeniería de la anchura de la banda prohibida" (*band-gap engineering*), que estudia las heterouniones. Una heterounión es una estructura constituida por la unión de dos semiconductores distintos y, por ende, con diferente anchura de la banda prohibida. En este tipo de estructuras ofrecen un grado adicional de libertad en el diseño de dispositivos, ya que las fuerzas que actúan sobre los electrones y los huecos pueden ser controladas de forma independiente mediante el ajuste de los anchos de banda prohibida (es decir, de la composición) y del dopaje de los semiconductores.

La aparición de los Transistores de Efecto de Campo de Heteroestructura (HFET) ha supuesto casi una revolución en el campo de las microondas de ultra-alta velocidad y de los circuitos electrónicos digitales. Los HFETs muestran prestaciones extremadamente elevadas, encontrándose HFETs discretos de microondas con figuras de ruido de 1.3 dB a 60 Ghz y ganancias de 9.5 dB. En relación a los circuitos digitales, se han conseguido circuitos con retardos de propagación de 10 ps por puerta y memorias SRAM de tiempo de acceso de 0.5 ns.

El modelado de los HFETs haciendo uso del simulador APLAC es el objeto fundamental del presente proyecto fin de carrera.

## Objeto del Trabajo

---

El objetivo de este proyecto ha sido el estudio de la capacidad de modelado de dispositivos del simulador eléctrico APLAC. Para ello se implementará en dicho programa el modelo para los Transistores de Efecto de Campo de Heteroestructura (HFETs). Por tanto previamente a esta labor se debe hacer un estudio de su funcionamiento y definir el modelo más adecuado para su implementación. La elección de este dispositivo se basó en que en la actualidad la mayoría de los simuladores no los implementan o lo hacen pero no de forma óptima

Además, se realizará un estudio general del simulador eléctrico APLAC y más específicamente de su lenguaje de programación y del modelado de nuevos dispositivos.

## Resumen

---

La memoria de este trabajo fin de carrera está dividida en 6 capítulos. En el primer capítulo presentamos una introducción al simulador de nueva generación APLAC.

En el capítulo 2 se describen las características generales de APLAC. Así mismo se hará una descripción detallada del lenguaje APLAC y de su capacidad de modelado.

En los capítulos 3 y 4 se hace un estudio de la estructura básica y de los principios de funcionamiento de los dispositivos de heteroestructura a modelar. Como resultado de esta fase se tendrá un conocimiento completo del modelo de Angelov para el HFET.

En el capítulo 5 se analizan varias metodologías para la implementación de los dispositivos HFETs en el simulador APLAC además de un estudio de los estamentos y componentes básicos para la realización de la implementación. Se hace una descripción detallada de VCCS (Voltage-Controlled Current Source), que es el elemento básico de modelado en APLAC.

Finalmente, en el capítulo 6 validamos los resultados obtenidos realizando una comparación de las simulaciones obtenidas en APLAC y las obtenidas por IASPICE. En dicho capítulo se presentarán también las conclusiones que se extraen de este trabajo.

# Introducción

El análisis de circuitos electrónicos es una tarea común a la práctica totalidad de la “comunidad electrónica” a pesar de la enorme diversidad de tareas en que se ocupa.

Este problema está formalmente resuelto: basta tomar las leyes de Kirchoff y las ecuaciones constitutivas de los elementos del circuito y resolver la matemática asociada. Las ecuaciones constitutivas son las expresiones algebraicas con las que se describe el funcionamiento de cada uno de los componentes o dispositivos utilizados y son el objeto de la disciplina “modelado de dispositivos electrónicos”.

El problema de análisis más sencillo consiste en encontrar el punto de operación en dc de un circuito lineal. Para un circuito pequeño compuesto por elementos lineales, descritos por ecuaciones constitutivas lineales, la solución exacta en dc se puede calcular a mano. Sin embargo, para circuitos lineales mayores el análisis en dc y más aún los análisis en el dominio de la frecuencia o en el dominio del tiempo se convierten en tareas mucho más complejas. El análisis de circuitos que contienen elementos descritos por relaciones no lineales entre las corrientes y las tensiones, añaden un nivel más de complejidad al requerir la resolución de ecuaciones de rama no lineales junto con las ecuaciones derivadas de las leyes de Kirchoff. La conclusión es clara, sólo se podrán resolver a mano circuitos pequeños obteniéndose soluciones aproximadas.

El problema se complica todavía más cuando se quiere predecir el funcionamiento de un circuito eléctrico en el dominio del tiempo o el de la

frecuencia. Las ecuaciones no lineales se convierten en ecuaciones integro-diferenciales, las cuales sólo pueden ser resueltas bajo la aproximación de pequeña señal u otras restricciones.

Antes de la aparición de los simuladores eléctricos con los que los computadores resuelven las ecuaciones, los diseñadores que usaban elementos discretos tenían que utilizar placas tipo *protoboard* para analizar el funcionamiento de los circuitos eléctricos. Incluso en la actualidad se utiliza este método para construir circuitos analógicos. Sin embargo, este sistema es inadecuado para la fabricación de circuitos integrados. Esto se debe a que los transistores integrados dentro de un mismo chip funcionan de forma diferente a los transistores discretos situados en una placa de prueba. Además existe otro problema y es que los elementos integrados difieren de sus equivalentes discretos. La fabricación de un CI es un proceso caro tanto en coste como en tiempo lo cual hace que el diseño electrónico deba ser lo más preciso posible. Se hace por tanto necesario el empleo de un simulador circuital definido como aquel programa que permite dar solución a las ecuaciones que describen un circuito.

## 1.1 ¿Qué es APLAC?

APLAC es una herramienta de simulación y de diseño de circuitos y componentes electrónicos.

APLAC ha estado bajo el constante desarrollo en la Universidad de Helsinki desde 1972. La meta principal de APLAC ha sido dar al usuario la libertad de resolver problemas de diseño sin las restricciones de los simuladores convencionales, como SPICE.

Las primeras cuatro generaciones de APLAC fueron desarrolladas en lenguaje BASIC o HP-BASIC y en PASCAL. La necesidad creciente del desarrollo de una versión portátil de APLAC, promovida por la compañía Nokia, fomentó la codificación de la quinta generación de APLAC en lenguaje C.

El centro de investigación de Nokia unió su equipo de trabajo para el nuevo desarrollo de APLAC, y prestaron una especial atención al desarrollo de APLAC en el lenguaje C orientado a objetos y al modelado de componentes.

Desde 1989, el Centro de Investigación Técnico de Electrónica de Finlandia participó en el modelado de dispositivos semiconductores. En 1991, el laboratorio de microelectrónica de la universidad estatal de Ohio contribuyó al modelo estadístico del MOSFET. En 1998, se fundó la corporación de desarrollo de APLAC y se desarrolló la séptima generación de APLAC, que mejoró los algoritmos de convergencia para los análisis no lineales. Se mejoró el menú de acciones del simulador y se creó un lenguaje de alto nivel para el modelado y solución de problemas. Se añadió la técnica de la convolución para permitir el uso de frecuencias dependientes para los análisis transitorios de componentes no lineales. Además añadieron nuevos métodos de análisis para los mezcladores y condensadores conmutados y nuevos análisis de ruido para circuitos no lineales. Al día de hoy APLAC sigue siendo desarrollado.

La orientación a objetos fue realizada para crear macros en C las cuales tuvieran en cuenta la necesidad específica de la simulación de circuitos y esencialmente la simplificación en el modelado. La orientación a objetos ofreció numerosas ventajas como la actualización increíblemente fácil e inclusión de nuevos algoritmos y modelos. Como un ejemplo, puede mencionarse que un cambio en el algoritmo de integración numérica requiere la modificación de sólo dos líneas del programa, una que contiene la fórmula de integración y la segunda que contiene la fórmula de error de truncamiento. Esto abre una manera rápida para probar la eficiencia de varios algoritmos. Además la orientación a Objeto abre nuevas vistas en los métodos de la simulación, habilitando nuevos algoritmos que nunca podrían comprenderse en los simuladores procesales convencionales como SPICE. La orientación a objeto permite adoptar iteraciones individuales y estrategias de integración a nivel del modelo, no sólo para que, por ejemplo todos los MOSFET usen una estrategia y todos los BJTs otra sino también para que el componente decida qué estrategia escoger observando el ambiente en el circuito. Así en el análisis transitorio por ejemplo, algún MOSFET en una determinada iteración, en una parte latente del circuito, podría usar unos pasos de tiempo mayores y la integración simple mientras otro MOSFET pudiera utilizar un paso más corto y una integración más avanzada.

La inclusión de un nuevo modelo en APLAC requiere la introducción de los parámetros del modelo y escribir las ecuaciones del modelo en las macros en C.

## 1.1.1 Una Pequeña Descripción de APLAC

### 1.1.1.1 Modos de Análisis

APLAC es capaz de llevar a cabo análisis en DC, AC, de ruido, transitorio, oscilatorio, de equilibrio armónico multitono de gran-señal/pequeña-señal, capacidad de conmutación ideal y no ideal, análisis en el dominio del tiempo y medidas usando el bus IEEE-488. En análisis transitorio APLAC trata, a través de la convolución, componentes definidos por características de dependencia en frecuencia. APLAC permite análisis de Monte Carlo en la mayoría de los análisis básicos. Existen varios métodos de análisis de estabilidad. Las simulaciones eléctricas y electrotérmicas podemos realizarlas a través de los modelos electrotérmicos de los componentes. APLAC incluye una colección de bloques de sistemas para la simulación y diseño de circuitos analógicos y digitales de sistemas de comunicaciones.

### 1.1.1.2 Análisis de Sistemas

Existen tres formas básicas para lograr las simulaciones de sistemas en APLAC:

- Definir componentes o subsistemas usando la capacidad de modelado del lenguaje APLAC.
- En particular, si se está realizando el diseño de un receptor interesaría ver las simulaciones de sistemas basadas en expresiones analíticas, aplicadas para determinar cada figura de mérito como la ganancia, puntos de interceptación y puntos de compresión.
- Sistemas de simulación más generales en dominio del tiempo se pueden simular haciendo uso de sistemas discretos. Este simulador permite el análisis de las formas de onda y/o el espectro en algunos de los nodos del circuito.

### 1.1.1.3 Análisis Electrotérmicos

La simulación electromagnética tridimensional en APLAC se realiza usando FDTD (Diferencias Finita en el Dominio del Tiempo) método capaz de aproximar las ecuaciones de Maxwell en el dominio del tiempo. Debido al acercamiento versátil de FDTD puede resolver un gran número de problemas electromagnéticos como la radiación de varias antenas, y la caracterización de los dispositivos de microondas, etc.

### 1.1.1.4 Modelos de APLAC

APLAC incluye una rica colección de modelos básicos lineales y no lineales, modelos de dispositivos semiconductores tales como BJT, MOSFET, Diodos, DMOS, HBT, IGBT, JFET, MESFET, MEXTRAM, amplificadores operacionales, PLL, modelos de sistemas discretos en el tiempo y también modelos de simulaciones electromagnéticas.

Las fuentes de control VCCS (Voltage-Controlled Current Source), CCCS (Current-Controlled current Source), VCVS (Current-Controlled Voltage Source) y CCVS (Voltage-Controlled Voltage Source) son las llaves para la creación de nuevos modelos.

Los parámetros de los modelos de varios componentes podrían ser funcionalmente dependientes de la frecuencia, tiempo, temperatura o algún otro parámetro. Se pueden crear nuevos componentes definiendo las características no lineales en estática y en dinámica.

### 1.1.1.5 Sweep

La función Sweep permite llevar a cabo barridos lineales o logarítmicos de cualquier número de parámetros del circuito así como del tiempo, frecuencia o temperatura.

### 1.1.1.6 El lenguaje APLAC

El fichero de entrada del simulador es un archivo tipo texto que contiene los nodos, conexiones, componentes y la definición del circuito o sistema físico. El fichero de entrada puede ser escrito utilizando cualquier editor de texto o el editor de APLAC el cual genera automáticamente el fichero de entrada. El lenguaje APLAC entiende todas las funciones matemáticas básicas y además permite el uso de: vectores y matrices reales o complejas, operaciones aritméticas, operaciones con cadenas y las estructuras For-, While- y Repeat-Until-Loops y If-Then-Else.

El lenguaje APLAC traza al usuario una línea directa a las funciones en lenguaje C.

### 1.1.1.7 Salida

La salida es cualquier función definida por el usuario, parámetros del circuito, tiempo, frecuencia o temperatura, así como análisis o medidas. El resultado de los análisis puede consistir en tensiones y corrientes, parámetros Z-, Y-, H- y S-, análisis de estabilidad, ganancia, estabilidad de la carga, carta de Smith, distorsión armónica, formas de ondas armónicas, componentes espectrales, análisis de ruido, temperatura, temperatura del ruido, sensibilidades, energía, radiación, etc. Los resultados pueden trazarse en las coordenadas rectangulares o polares o en la carta de Smith. También pueden usarse gráficos 3D. Los gráficos pueden ser guardados en archivos HPGL-, CSDF o Meta-file o en gráficos internos. La ventana grafica incluye herramientas de creación automática de histogramas Monte-Carlo y transformada de Fourier.

### 1.1.1.8 Optimización

APLAC incluye diferentes métodos de optimización: gradientes, gradientes conjugadas, mínimo-máximo, búsqueda aleatoria, búsqueda exhaustiva, variable y centro de gravedad. Cualquier parámetro en el diseño del problema puede ser utilizado. Como una variable o una función. La optimización en el dominio de frecuencia y de tiempo también es posible.

### 1.1.2 Modelado

En APLAC, el elemento básico de modelado es VCCS o sus duales VCVS, CCVS y CCCS. Hay tres opciones de modelado diferentes:

- Elementos estáticos los cuales pueden ser lineales o no lineales. “Estático” significa sin memoria. Los elementos no lineales son descritos mediante su corriente y sus derivadas (conductancias).
- Elementos capacitivos. Son descritos mediante la carga y las capacidades.
- Los elementos dinámicos son descritos en el dominio de la frecuencia.

En APLAC existen limitaciones. Primero, APLAC no controla la razonabilidad de un modelo. Un modelo no lineal inapropiado provocará problemas de convergencia. Los parámetros funcionalmente dependientes del tiempo deben utilizarse sólo en análisis transitorios. Los elementos dependientes de frecuencias trabajan correctamente en AC y en análisis de equilibrio armónico pero no análisis transitorio, donde sólo se utilizan sus valores cuasi-estáticos.

Un modelo puede ser definido con el bloque DefModel ... EndModel y luego utilizar este nuevo componente como cualquier otro componente de APLAC.

### 1.1.3 Opciones de APLAC

Hay tres opciones principales:

- Opciones RF.
- Opciones de Sistemas.
- Opciones Electromagnéticas.

Se pueden realizar combinaciones entre las tres opciones.

### 1.1.3.1 Opciones RF

Las opciones RF activa las siguientes modalidades de análisis:

- Análisis de equilibrio armónico multitono de pequeña-señal/gran-señal.
- Análisis de ruido no lineales.
- Análisis en el dominio del tiempo del estado de equilibrio.
- Análisis de capacidad en conmutación.

### 1.1.3.2 Opciones de Sistema

Las opciones de simulación de sistema permiten simulaciones utilizando modelos de tiempo discreto.

### 1.1.3.3 Opciones Electromagnéticas

Las opciones electromagnéticas añaden varias simulaciones FDTD (Diferencia Finita en el Dominio del Tiempo).

# Características Generales del Simulador Eléctrico APLAC

En este capítulo se pretende realizar un estudio general del funcionamiento y de las características más relevantes del simulador eléctrico APLAC. Para un estudio más detallado el lector interesado puede remitirse a los manuales [1].

Para conseguir este objetivo realizaremos una sesión típica usando el simulador de APLAC y sus herramientas: el editor esquemático y el editor de APLAC.

En este capítulo se aprenderá que APLAC es una herramienta electrónica poderosa y versátil.

Debido a que el simulador está escrito en inglés, a la hora de presentar los comando se intentará hacer una traducción lo más exacta posible. Al lado de cada comando se pondrá su versión inglesa para facilitar al lector su entendimiento y poder identificarlo más fácilmente dentro del programa.

## 2.1 Principios Básicos

Como un principio muy básico, APLAC utiliza para realizar sus simulaciones dos herramientas de calculo:

- El simulador de APLAC.

- El editor de APLAC.

Se puede invocar las simulaciones directamente desde el editor de APLAC y no notarse el uso real del simulador por separado. Esto se debe a que el simulador realiza el análisis en segundo plano.

El editor de APLAC no es estrictamente necesario, ya que se puede realizar las simulaciones directamente utilizando el lenguaje de APLAC y un editor de texto.

Para ejecutar ambas herramientas se utilizan los siguientes ficheros de entrada:

- `aplac.exe` es el simulador de APLAC ejecutable, todas las simulaciones son realizadas por él. En UNIX se lanza ejecutando `aplac` en la consola del sistema junto al fichero a simular.
- `aplaced.exe` es el editor de APLAC. Su principal función es la de editar los esquemáticos de las simulaciones. Con este programa se pueden visualizar los componentes y sus interconexiones en modo gráfico. En UNIX se lanza ejecutando `aplac` en la consola del sistema.

En la figura 2.1 se ilustra la interacción entre el simulador de APLAC y el editor de APLAC:

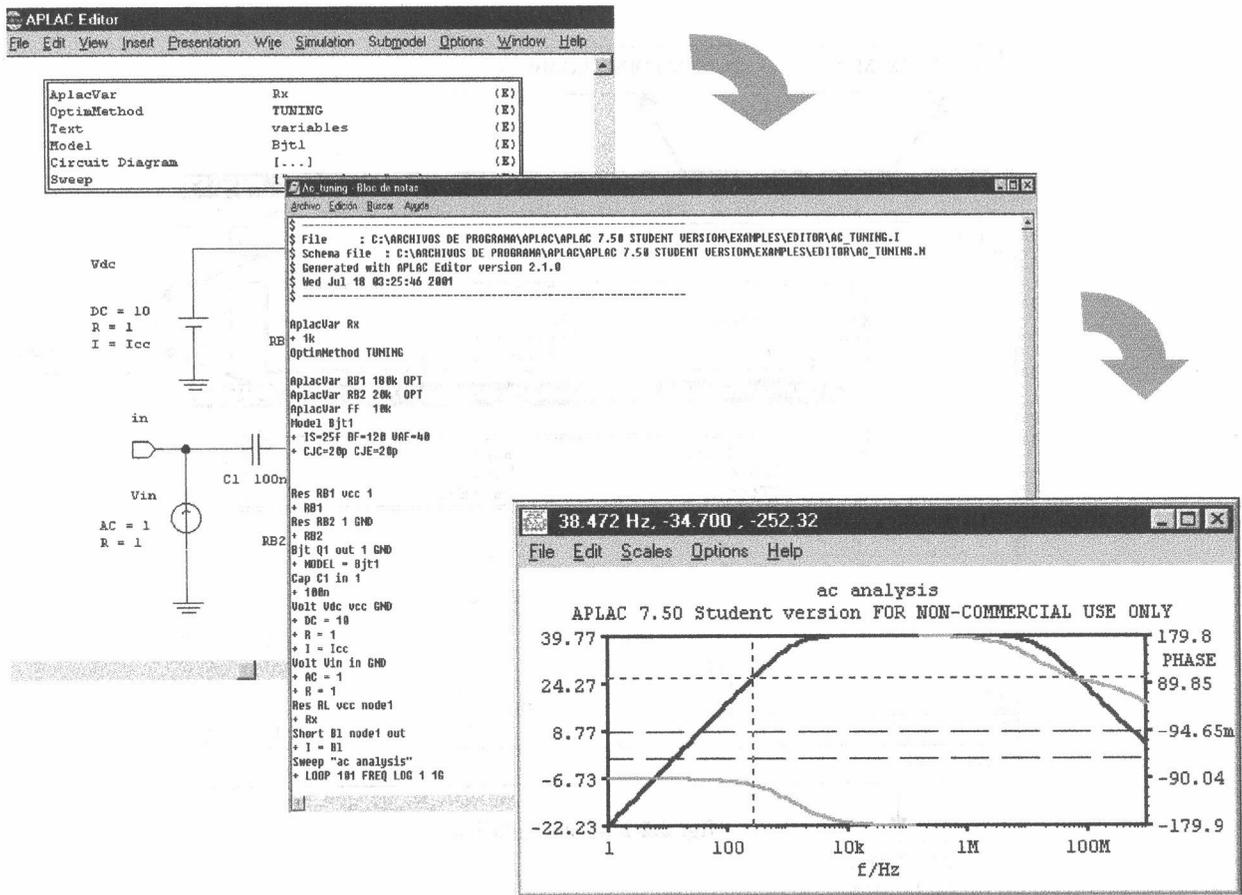


fig. 2.1 Interacción entre el simulador de APLAC y el editor de APLAC

## 2.2 Primer Ejemplo

Para observar los conceptos básicos del simulador APLAC, realizaremos una simulación muy simple. A continuación se detalla el entorno gráfico del editor de APLAC, figura 2.2.

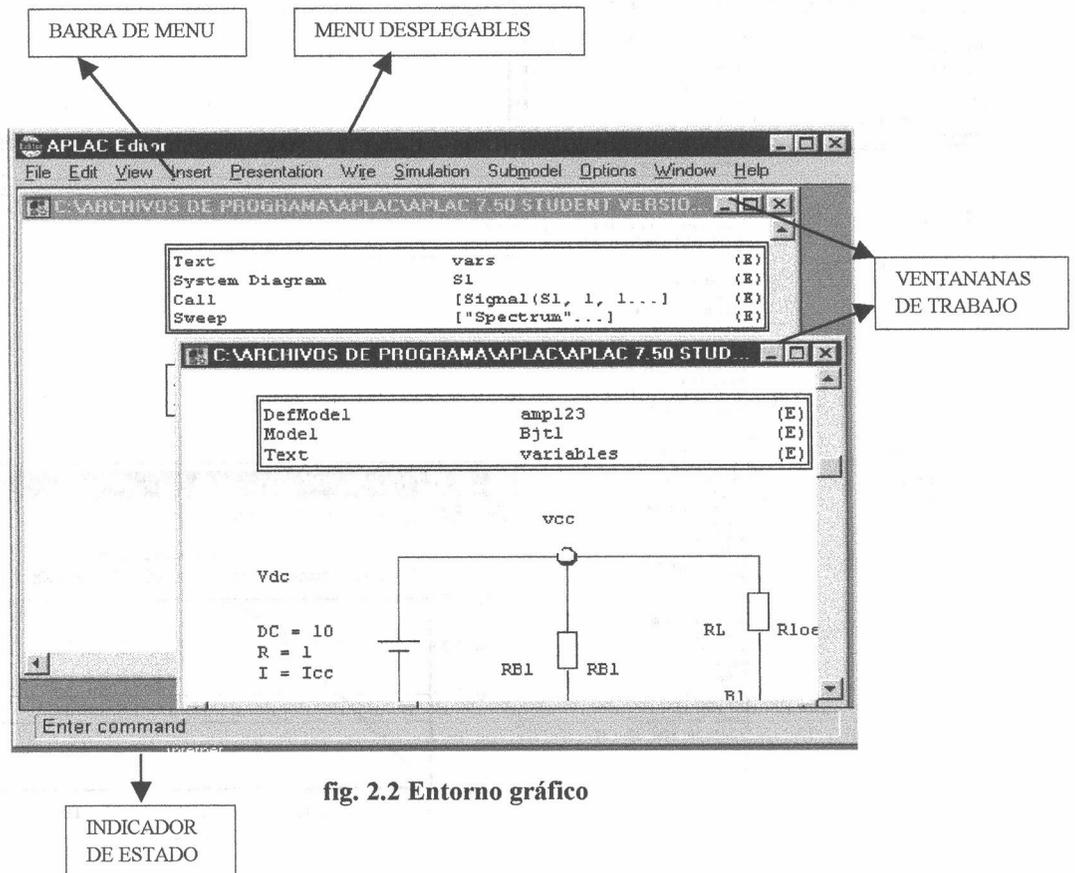


fig. 2.2 Entorno gráfico

El editor de APLAC esta compuesto de dos partes fundamentales:

- Un diagrama objeto el cual nos muestra los componentes de la simulación y las interconexiones entre los componentes.
- Uno o varios controles objetos (que define entre otras cosas los tipos de análisis que debe realizar el simulador).

### 2.2.1 Realización de un Circuito RC

Empezaremos con el circuito que se muestra en la figura 2.3. Este circuito es muy sencillo y nos servirá para mostrar los conceptos básicos de APLAC. Una vez asumido los conceptos básicos nos introduciremos en conceptos más complejos.

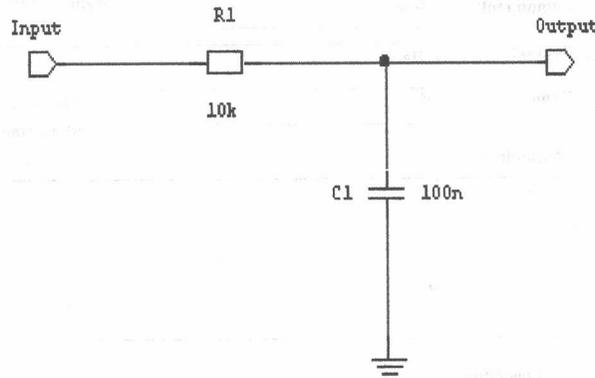


fig. 2.3 Circuito RC

El primer paso sería introducir los componentes. La inserción de los componentes lo realizamos en el menú insertar y luego en componentes (figura 2.4). Una vez realizado los dos pasos anteriores APLAC nos mostrará la librería de componentes (figura 2.5) en la cual seleccionamos los componentes a insertar.

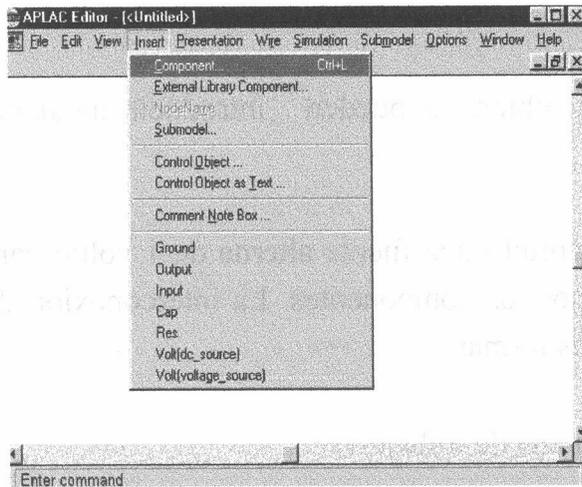


fig. 2.4 Menú insertar

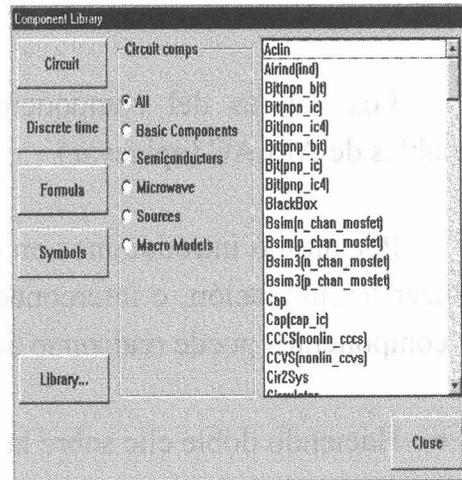


fig. 2.5 Librería de componentes

Una vez insertados todos los componentes, modificamos sus atributos. Haciendo doble clic sobre el componente accedemos a la ventana de sus atributos (figura 2.6), donde los modificamos. Los atributos serán los que gobiernen la conducta del componente.

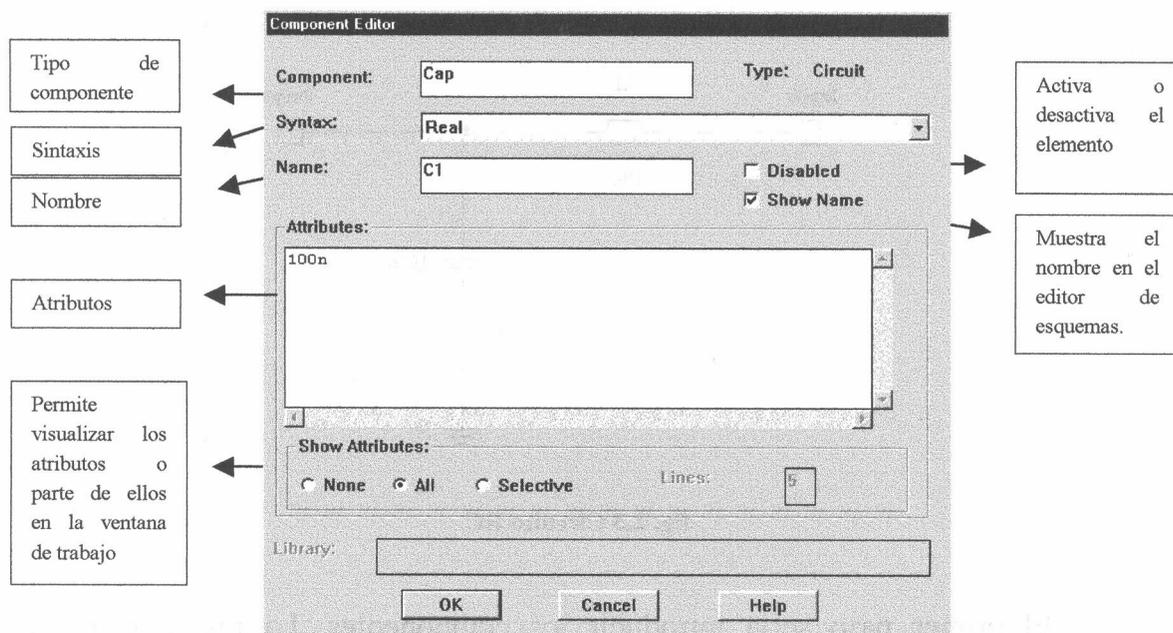


fig. 2.6 Editor de componentes

Los componentes pueden ser desplazados, rotados (Ctrl R), duplicados (Ctrl D), borrados (del) y desconectados (Ctrl + minúsculas), utilizando el menú editar.

Los valores del componente también se pueden introducir mediante variables de APLAC (aplacvar).

Por último introducimos en el circuito una fuente alterna de 1 voltio para realizar la simulación, e interconectamos los componentes. La interconexión de los componentes puede realizarse de tres formas:

- Haciendo doble clic sobre la ventana de trabajo vacía.
- Eligiendo en el menú Wire > Start Wiring.
- Apretando Ctrl +W.

Cuando se realiza las conexiones de los componentes, los componentes no conectados resaltarán del resto. Además, una vez conectado un componente puede ser desplazado, sin que se pierdan las conexiones.

### 2.2.2 Realización de la Simulación

Existen varias maneras de realizar la simulación en APLAC.

Hemos introducido una fuente de tensión de 1v en ac y ahora debemos indicar a APLAC a que frecuencias queremos realizar la simulación. Para ello debemos conocer algunos conceptos importantes como el *diagrama de objetos* y el *control de objetos*.

Se debe especificar un Diagrama Objeto como diagrama de circuito que contiene la información de los componentes, interconexiones, valores, etc. Para un circuito debe haber un sólo diagrama de circuito que normalmente se crea por defecto.

Los controles objetos son quienes gobiernan las tareas de los análisis. Puede haber un número indeterminado de controles objetos, los cuales pueden cumplir diferentes funciones como: barridos, vectores, variables, etc.

Las posiciones de los controles objetos pueden verse, modificarse o borrarse con la lista de controles objetos. APLAC ejecutará los controles objetos secuencialmente por lo que debe tenerse en cuenta su colocación. Por ejemplo no se puede colocar un barrido (sweep) antes que el diagrama del circuito puesto que APLAC determinaría un error.

Introduciremos el control objeto (figura 2.7) en el menú insertar.

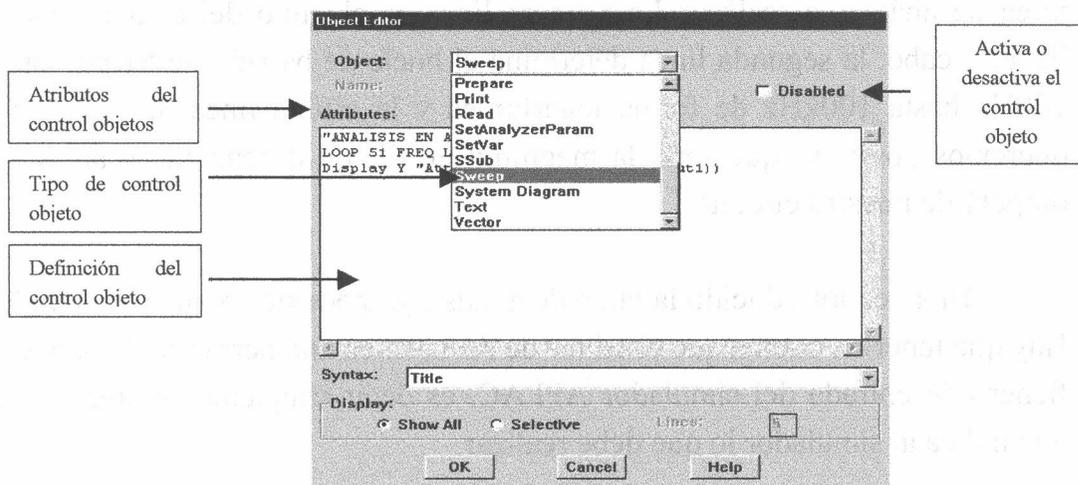


fig. 2.7 Control objeto

Para mostrar en el editor de APLAC un control objeto o la lista de los controles objetos ( figura 2.8), utilizamos el menú Show.

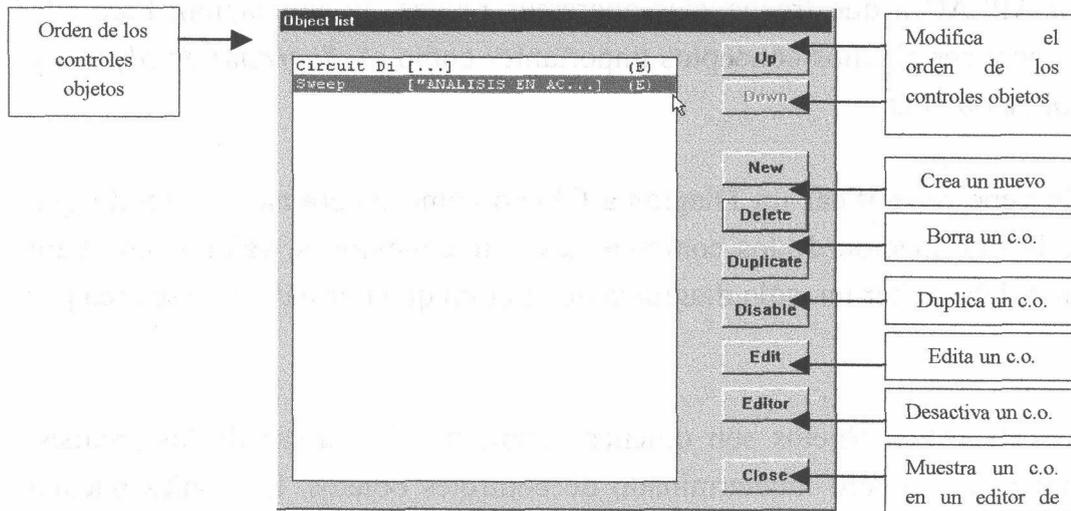


fig. 2.8 Lista de los controles objetos

Por último antes de la simulación definiremos los atributos del control objeto, especificando la tarea de análisis a realizar:

```
Sweep "ANALISIS EN AC"
LOOP 51 FREQ LOG 100Hz 100kHz
Display Y "Attenuation" MagdB(Vac(output1))
EndSweep
```

Ahora realizaremos una breve descripción de cómo hemos indicado a APLAC la tarea de análisis a realizar: La primera línea es el título del análisis que se va llevar a cabo, la segunda línea determina el bucle de barrido de 51 puntos desde 100Hz hasta 100kHz de forma logarítmica y la última línea determina lo que queremos observar, que sería la magnitud en dB de la tensión en ac del punto output1 de nuestro circuito.

Una vez introducido la tarea de análisis ya podemos realizar la simulación. Hay que tener en cuenta que el editor de APLAC es una herramienta para crear el fichero de entrada del simulador APLAC, es decir simplemente crea el archivo que indica al simulador lo que debe realizar.

Nota : en las versiones de APLAC de WIN95/NT hay por defecto en la lista de objeto, un control de objeto tipo Isweep “barrido interactivo” que sino vamos ha utilizarlo debemos eliminarlo.

Ahora nos vamos al menú simulación y le damos la orden de simular. Si lo hemos realizado correctamente nos saldrá el gráfico de la atenuación que se muestra en la figura 2.9.

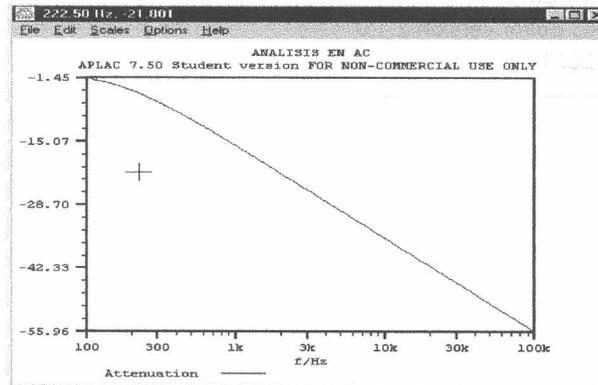


fig. 2.9 Atenuación

### 2.3 Archivo de Entrada del Simulador APLAC

Hay varias maneras de ver el archivo de entrada del simulador APLAC creado por el editor:

- 1) Dentro del editor en el menú de simulación le indicamos que genere el archivo de entrada del simulador y que nos lo muestre.
- 2) Una vez simulado el circuito en el menú file del simulador de APLAC hacemos Clic en edit y nos muestra el archivo que se va a ejecutar.
- 3) Utilizando un editor de texto como el notepad.exe en WINDOWS y en UNIX por ejemplo el editor vi.

De la primera forma sólo podemos observar el fichero (Los cambios que realicemos no se tendrán en cuenta), mientras con las otras dos formas podemos

modificarlos y ejecutarlos directamente en el simulador. En el caso anterior el fichero de entrada que se generó fue:

---

### Códigos

---

```

$ -----
$ File: C:\ARCHIVOS DE PROGRAMA\APLAC\APLAC 7.50 STUDENT
$VERSION\EXAMPLESMEMO.I
$ Schema file:C:\ARCHIVOS DE PROGRAMA\APLAC\APLAC 7.50 STUDENT
$VERSION\EXAMPLESMEMO.N
$ Generated with APLAC Editor version 2.1.0
$ Tue Apr 10 19:43:16 2001
$ -----

```

```

Volt V1 node1 GND
+ ac=1
Res R1 node1 Output1
+ 10k

Cap C1 Output1 GND
+ 100n
Sweep "ANALISIS EN AC"
+ LOOP 51 FREQ LOG 100Hz 100kHz
Display Y "Attenuation" MagdB(Vac(output1))

EndSweep

```

---

Las primeras líneas que van precedidas por el símbolo \$ son comentarios. La primera línea funcional es la que define volt:

```

Volt V1 node1 GND
+ ac=1

```

El signo “+” es el carácter de continuación de línea. La línea que comience con “+” es continuación de la línea anterior, de esta manera pueden añadirse varias líneas juntas. La definición de VOLT es una declaración para APLAC de una fuente cuyo nombre es V1 y que esta conectada entre los nodos NODO1 Y GND, siendo NODO1 el nodo positivo de la fuente. En la siguiente línea determina que es una fuente de alterna de un 1 voltio.

Conocer el lenguaje APLAC nos permitirá realizar las simulaciones directamente en el simulador, generando nosotros mismos el fichero de entrada.

## 2.4 Controles del Simulador APLAC

Se puede utilizar el simulador de APLAC como una única herramienta de trabajo. A continuación veremos los controles básicos (figura 2.10) para utilizar el simulador de APLAC como una única herramienta de trabajo:

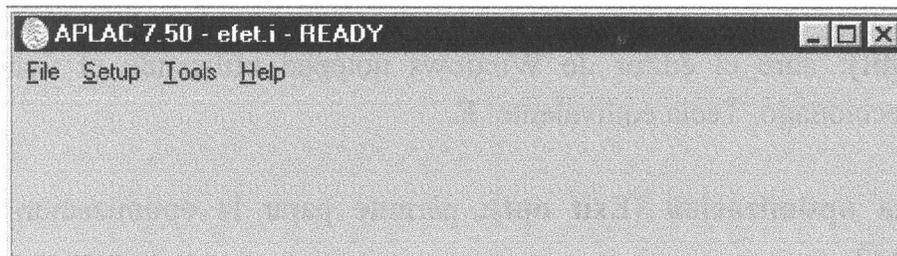


fig. 2.10 Ventana de trabajo del simulador

### 2.4.1 Menú de fichero (Menú File)

Este menú incluye los comandos básicos para controlar el simulador:

**Seleccionar fichero (Select file):** En este menú, el procedimiento de selección trabaja como cualquier herramienta de Windows, los ficheros que debemos seleccionar son del tipo:

- **.i:** define todos los ficheros de entrada del simulador. Cuando se ha seleccionado un fichero de entrada correcto puede comenzarse la simulación o invocar el editor de texto notepad para modificar el fichero de entrada
- **.cir:** los ficheros generados por el simulador SPICE. Se puede empezar la conversión de SPICE a APLAC apretando el botón CONVERT aunque este procedimiento suele dar muchos errores y se deben modificar a mano para que funcionen correctamente.
- **.grx:** se pueden cargar ficheros gráficos almacenados durante una simulación usando la ventana gráfica de APLAC .

- .iii: este tipo de fichero nos permite generar múltiples simulaciones una tras otra. Todos los ficheros de entrada irán generándose línea por línea.

**Simular (Run):** invoca el simulador de APLAC utilizando el fichero de entrada seleccionada. Tecla equivalente: **R**.

**Editar (Edit):** abre el editor de Windows notepad utilizando el fichero de entrada seleccionado. Tecla equivalente: **E**.

**Salir de la optimización (Exit opt):** permite parar la optimización. Tecla equivalente **O**.

**Salir del análisis de Monte Carlo (Exit Monte Carlo):** permite parar la simulación estadística.

**Abortar simulación (Abort simulation):** si no se está utilizando el análisis de montecarlo este menú permite salir de la simulación. Tecla equivalente: **ESC**.

**Continuar la simulación (Continue simulation):** se puede continuar la simulación previamente abortada.

**Salir de APLAC (Exit APLAC):** para y sale del simulador.

#### 2.4.2 Menú de Instalación (Menú Setup)

Este menú permite el usuario cambiar los parámetros en la interfase de Windows.

**Almacenamiento de los parámetros de instalación (Store setup):** graba los parámetros del simulador después de su uso. Los datos son almacenados en el fichero aplac32.ini, la siguiente vez que se ejecute el simulador los datos son restaurados.

**Instalación principal (Main setup):** este menú modifica los parámetros de control del simulador. Se puede posicionar y redimensionar el ventana del simulador. También se puede cambiar los colores predefinidos.

**Texto de salida (Text output):** permite mostrar todo el texto de salida definido por el fichero de entrada como alertas y errores producidos en la simulación. El texto de salida es creado junto con el texto de entrada, el texto de salida será reescrito cada vez que la simulación comience.

**Verboso (Verbose):** es similar a texto de salida, pero tiene un campo adicional llamado verbose time el cual controla la mayoría de la información cada instante en la simulación.

**Graficos (Graphics):** podemos determinar las dimensiones de las ventanas gráficas, tipo y color de letra a utilizar, color de fondo de la ventana, etc.

### 2.4.3 Menú de Herramientas (Menú Tools)

En este menú describiremos las principales herramientas:

**Sintaxis (Syntax):** muestra la sintaxis de un componente o una función. Primero se selecciona el componente o la función y en el texto de salida APLAC nos mostrará la sintaxis de lo que hallamos seleccionado (figura 2.11)

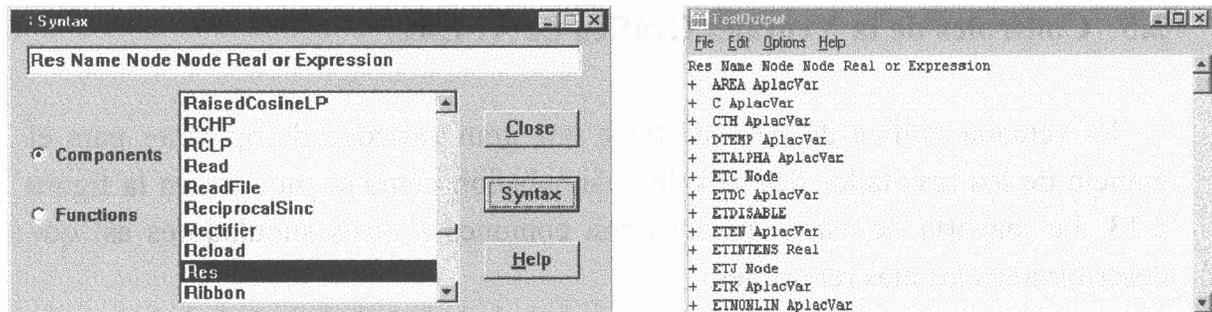


fig 2.11 Sintaxis

**Información de componentes (Component Info):** permite examinar los componentes incluidos en la última simulación de APLAC. El tipo y nombres de los componentes son listados. Eligiendo uno de ellos permitirá observar sus valores públicos (figura 2.12)

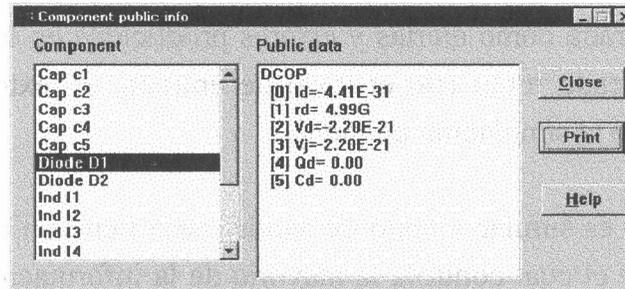


fig. 2.12 Información de componentes

**Información de nodo (Node info):** muestra los nodos del último circuito simulado y nos permite observar la tensión y la corriente que tienen dichos nodos.

**Información de variables (Aplacvar info):** lista todas las variables definidas en el fichero de entrada, y nos muestra información adicional de la variable.

**Optimización de Variables (Optimization variables):** es similar a **Aplacvar info**, permite observar información adicional sobre el estado de las variables optimizadas.

## 2.5 Controles de la Ventana Gráfica de APLAC

La ventana gráfica de APLAC tiene una gran variedad de opciones para el manejo de los resultados después de la simulación como se muestra en la figura 2.13. La mayoría de estas opciones son comunes a otros simuladores así que describiremos las más relevantes.

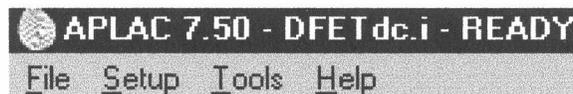


fig. 2.13 Menú de la venta gráfica de Windows

### 2.5.1 Menú de Opciones (Options Menu)

**Cursores (probe):** nos permite la medición de puntos específicos de las curvas así como los máximos y mínimos mediante los cursores, tal y como se muestra en la figura 2.14.

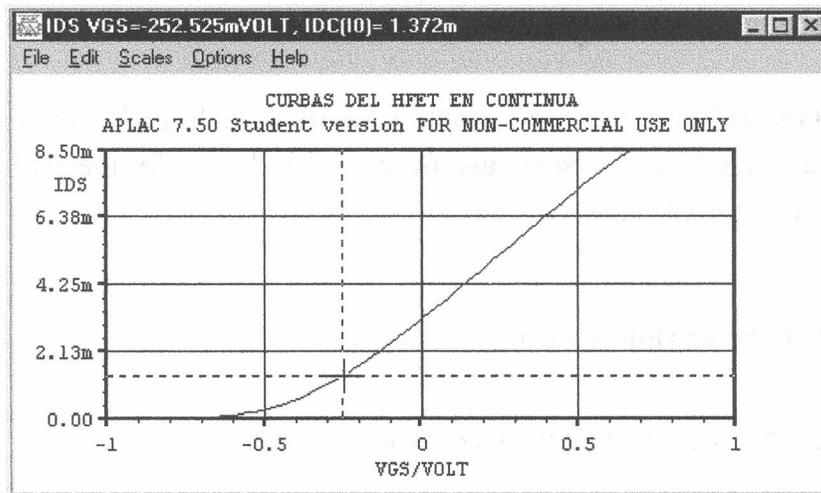


fig 2.14 Medición de un punto específico

**Pasar el eje X a logarítmico (Log X axis):** sólo podemos utilizar este comando si la curva representada no presenta valores negativos, convierte el eje X de lineal a logarítmico.

**Pasar el eje Y a logarítmico (Log Y axis):** igual que el comando anterior.

**Pantalla grande (Big Screen):** presenta la ventana gráfica a pantalla completa.

**Smith:** representa si es posible las curvas en la carta de Smith.

### 2.5.2 Menú Editar (Edit menu)

Podemos realizar un gran variedad de operaciones con las curvas en la ventana gráfica. APLAC permite copiar una curva y pegarla en otra ventana gráfica o en un bloc de notas con lo que obtendríamos todos los puntos de dicha curva como se muestra a continuación (resultado de copiar y pegar en bloc de notas la figura 2.14):

```

IDC(10)
CURVES 1 TYPE Y PEN 1 LINE 1 WIDTH 1 MARKER 0 STYLE NORMAL
POINTS 100
-1 8.5424869e-006      -1
-0.97979798 9.3478841e-006 -0.97979798
-0.95959596 1.0260772e-005 -0.95959596
-0.93939394 1.1306515e-005 -0.93939394
-0.91919192 1.2518092e-005 -0.91919192
-0.8989899 1.3938075e-005 -0.8989899
-0.87878788 1.5620912e-005 -0.87878788
-0.85858586 1.7635512e-005 -0.85858586
...

```

Las dos primeras columnas corresponden a los datos del eje X e Y y la tercera correspondería al eje Z en nuestro caso no es utilizado por lo que APLAC copia lo mismos valores que el Eje X.

### 2.5.3 Menú de ficheros (File menu)

En este menú destacaremos dos opciones:

**Transformada de Fourier (Fourier transform):** aplica la transformada discreta de Fourier a los resultados. También se puede aplicar FFT (La Transformada de Fourier Rápida).

**Histogramas (Histogram):** se utiliza para realizar análisis Monte Carlo a los resultados obtenidos, por ejemplo los diferentes tipos de distribución obtenidas desde la simulación estadística.

## 2.6 Lenguaje APLAC

Sirve para utilizar APLAC escribiendo directamente el fichero de entrada del simulador. El fichero de entrada podemos escribirlo con cualquier editor de texto o utilizando el editor esquemático de APLAC el cual genera automáticamente el fichero de entrada.

Descripción general del fichero de entrada:

- “\$” Empieza un comentario (solamente en mitad de una línea). “\*” toda la línea es un comentario (al principio de línea).

- Una declaración puede ser continuada en la próxima línea usando la marca de continuación "+", por ejemplo:

Res R1 1 2 1k es equivalente a

```
Res
+ R1
+ 1 2
+ 1K
```

- Todas las medidas son asumidas por APLAC en unidades internacionales.
- Las letras escalares son equivalentes:

Unidades	Por defecto	Multiplicador
E		$10^{18}$
P	PET o Pet	$10^{15}$
T	T o t	$10^{12}$
G	G o g	$10^9$
M	MEG o Meg	$10^6$
K	K o k	$10^3$
%	%	$10^{-2}$
m o mm	M o m	$10^{-3}$
mil	MIL o mil	$25.4^{-6}$
u	U o u	$10^{-6}$
n	N o n	$10^{-9}$
p	P o p	$10^{-12}$
f	F o f	$10^{-15}$
a	A o a	$10^{-18}$

Tabla 2.1

Por ejemplo se puede utilizar 1k, 1K y 1kohm. Hay que tener algunas precauciones, por ejemplo si queremos especificar una corriente de una fuente de un amperio debemos escribir 1 y no 1A (se interpretaría como  $10E-18$ ), otro error común es especificar  $1000\mu F$  como 0.01F (interpretado como  $10e-17$ ) la forma correcta sería 10000u o 10000uF.

- Directiva #ver en el fichero de entrada nos da los detalles de la versión de APLAC utilizada y las librerías que contiene.

### 2.6.1 Introducción con Dos Simples Ejemplos

Desarrollaremos el filtro paso bajo de la figura 2.15, utilizando así algunas de las ideas básicas sobre el lenguaje APLAC:

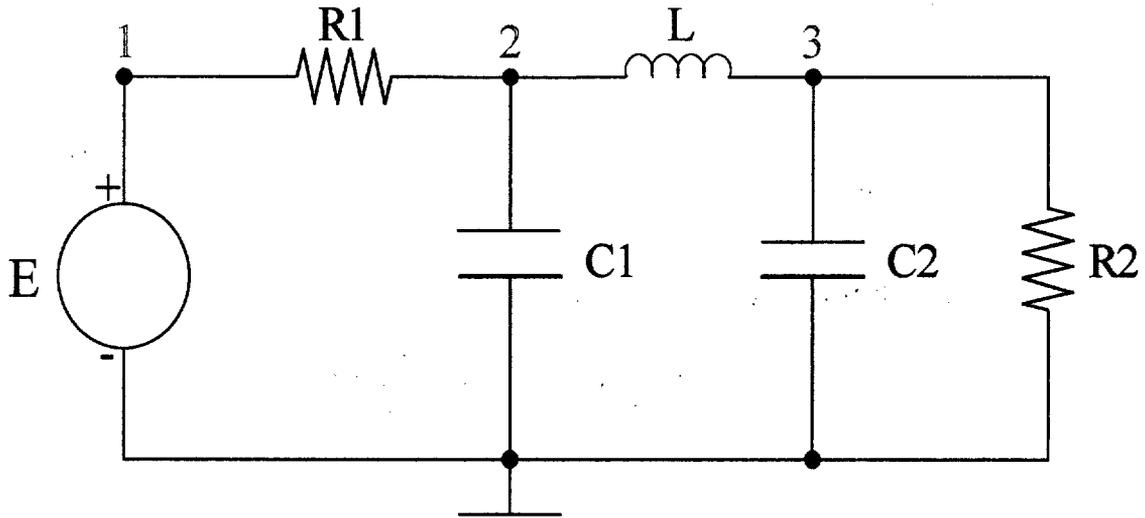


fig. 2.15 FILTRO PASO BAJO

Todas las declaraciones en el ejemplo está seguido por el carácter “\$”, el cual comienza un comentario, seguida de una descripción de la declaración en cuestión. El fichero de entrada del circuito figura 2.15 es como se muestra a continuación:

Códigos	Comentarios
Volt E 1 0 AC=1	\$Define una fuente AC ideal E. Su nodo positivo es 1 y el \$negativo 0. La amplitud es 1 V. El nodo tierra puede ser \$indicado con 0 o GROUND.
Res R1 1 2 50	\$Resistencia R1 definida entre los nodos 1 y 2 y su \$resistencia es de 50Ω.
Cap C1 0 2 0.1u	\$Condensador C1 definido entre los nodos 0 y 2 su \$capacidad es de 0.1uF

Continúa en la siguiente página

Códigos	Comentarios
Ind L 2 3 1m	\$Bobina L definida entre los nodos 2 y 3, su inductancia es \$de 1mH.
Cap C2 3 0 0.1u	\$Condensador C2 definido entre los nodos 3 y 0 de \$inductancia 0.1 uF.
Res R2 3 0 50	\$Resistencia R2 entre los nodos 3 y 0 de resistencia de \$50Ω.
Sweep "Análisis en AC"	\$ El barrido es utilizado para describir el trabajo a realizar. \$La ventana gráfica tendrá el título "Análisis en AC".
+ LOOP 100 FREQ LOG 0 10MEG	\$Los parámetros del eje x son definidos mediante el \$bucle. El cual es un barrido logarítmico en frecuencia \$utilizando 100 puntos desde 10 Hz hasta 10 MHz .
+ y "incremento" "DB" -200 0	\$Eje "y" de nombre incremento y unidad en DB, escalado \$desde -200 a 0.
Show Y Magdb(Vac(3))	\$Muestra los resultados especificados. Se mostrará la \$magnitud en DB de la tensión en AC del nodo3 utilizando \$la escala predeterminada del eje Y.
EndSweep	\$Un barrido siempre debe ser finalizado con EndSweep. \$Todas las líneas entre Sweep y el End Sweep son \$ejecutadas punto por punto.

La salida del análisis se muestra en la figura 2.16.

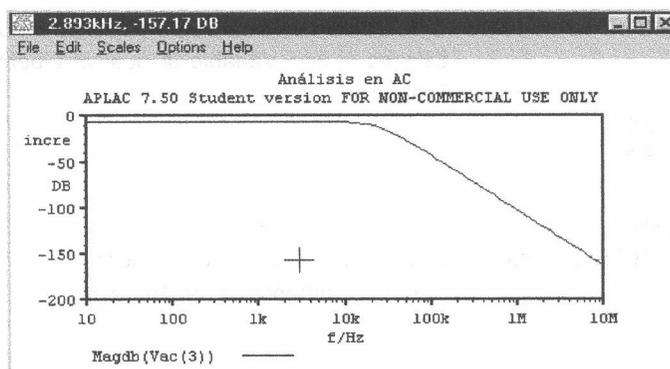


Fig. 2.16 Respuesta en amplitud del filtro paso bajo

Ahora consideremos que la bobina del circuito figura 2.15 tiene una desviación parasitaria de capacidad como se muestra en la figura 2.17.

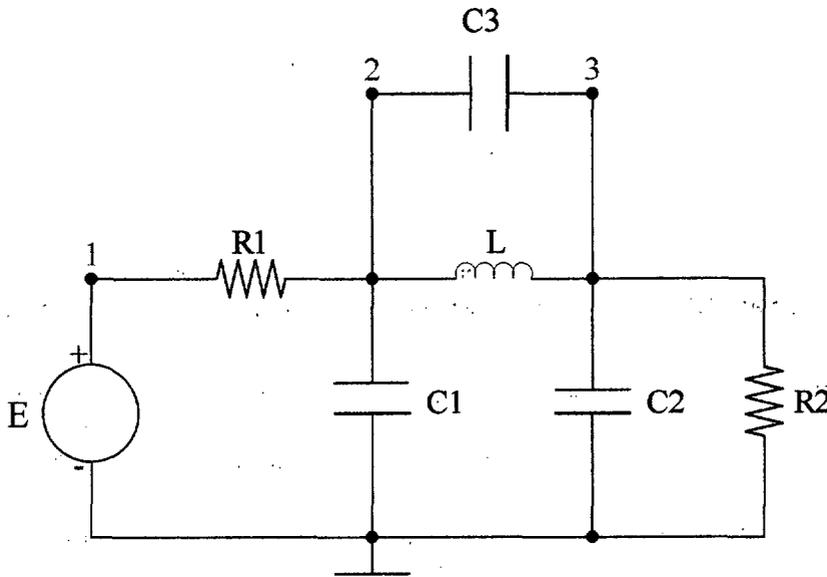


fig. 2.17 Filtro Paso Bajo con Desviación Parasitaria

El efecto de los diferentes valores de esta capacidad parásita en amplitud podemos simularlo. Básicamente se define una variable como **Aplacvar** al cual se le pueden asignar diferentes valores. El fichero quedaría como:

Códigos	Comentarios
AplacVar c	\$ Esta línea define la variable c.
Volt Eg 1 0 AC=1	
Res R1 1 2 50	
Cap C1 0 2 0.1u	
Ind L 2 3 1m C=c	\$Añadimos la capacidad parasitaria C con valor c, \$siendo c una Variable la cual nos permitirá cambiar su \$valor.
Cap C2 3 0 0.1u	
Res R2 3 0 50	
Sweep "AC Analysis"	
+ LOOP 5 APLACVAR c LOG 1p 1n	\$La variable c llega a 5 valores \$logarítmicamente desde 1pF a 1nF.
+ LOOP 1000 FREQ LOG 10 100MEG	
+ Y "Ganancia" "dB" -200 0	
Show Y MagdB(Vac(3))	
EndSweep	

La simulación es bastante diferente a la anterior, presentando los cambios producidos por la capacidad parásita como se muestra en la figura 2.18:

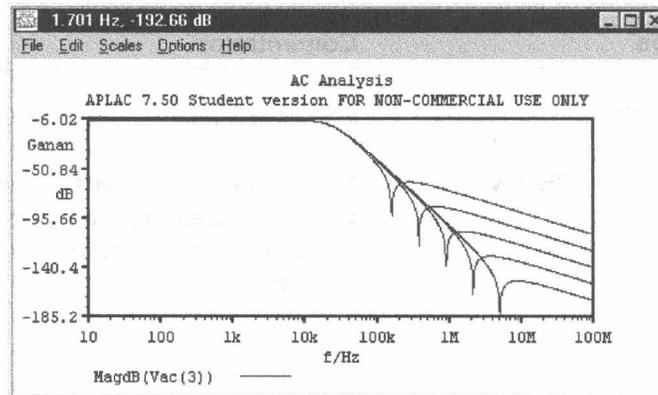


fig. 2.18 Respuesta en amplitud del filtro paso bajo considerando la capacidad parasitaria

## 2.6.2 Estructura del Archivo de Entrada

El fichero de entrada es un programa ejecutable que contiene directivas del preprocesador, descripción de circuito, variables y definiciones de funciones, expresiones matemáticas y declaraciones que controlan la simulación y optimización.

La estructura del archivo de entrada es bastante flexible aunque se debe cumplir algunas reglas, tales como:

- Todas las definiciones incluyendo **#define** deben ser escritas antes de utilizar **#undef** que termina el efecto del **#define**.
- Algunos estamentos deben definirse antes de su llamadas. Por ejemplo **Aplacvar**, **Model**, y **Defmodel** debemos definirlos al principio del archivo. La descripción del circuito debe estar definida antes de las declaraciones de los análisis. Cuando especificamos un componente la sintaxis es la siguiente: componente, nombre, nodos, argumentos obligatorios, argumentos opcionales.

- El camino más simple para poner en marcha un análisis (sin presentación gráfica) y mostrar los resultados es usando los estamentos **Analyze** y **Print**, por ejemplo:

Códigos	Comentarios
Analyze DC	\$ Se lleva a cabo el análisis en Dc.
Print S "vbe = " REAL Vdc(3) LF	\$ S comienza una cadena de texto, (S es un alias de \$Text), REAL es seguido por un número real, Vdc(3) \$da el voltaje en DC del nodo 3, y LF define final de \$la línea.

- A menudo lo más utilizado es llevar a cabo un análisis gráfico con lo que utilizamos el estamento **Sweep ... EndSweep**, por ejemplo:

Códigos	Comentarios
Sweep "Análisis transitorio"	\$ Definición de la primera ventana gráfica.
+ Loop 2500 TIME LIN 0 1u	
+ WINDOW 0	\$ Definición de la segunda ventana gráfica.
+ Y "Vout" "V" 0 5	
+ WINDOW 1	
+ X "t" "s" 0.9u 1u	\$ Utiliza un color predefinido para dibujar la curva
+ Y "Vout" "V" 0 5	
Show Y Vtran(2)	
PEN=2	
EndSweep	

### 2.6.3 Propiedades del Lenguaje de Programación

APLAC puede ser un lenguaje de programación o un simulador de circuitos. En este apartado pondremos mayor énfasis en su uso como lenguaje de programación. Esta característica es muy útil cuando se quiere definir circuitos complejos con análisis complejos.

### 2.6.3.1 Variables

La declaración **Aplacvar** define las variables básicas en APLAC. Existen diferentes tipos de variables: constantes, variables normales, variables estadísticas, variables optimizadas o variables funcionales.

#### 2.6.3.1.1 Variables Constantes y Normales

En este apartado haremos una descripción de diferentes formas de crear y modificar variables del tipo constantes y normales, que son las más utilizadas.

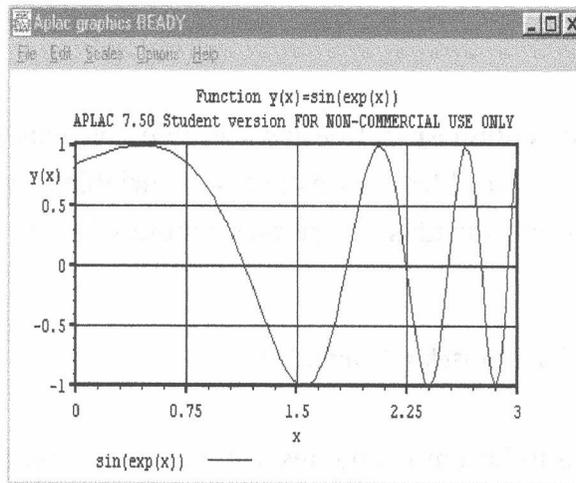
La siguiente declaración:

```
Aplacvar x=10
```

define una variable normal  $x$ , cuyo valor es 10. Si no definimos su valor en la declaración de una variable, APLAC utiliza por defecto  $x=1$ . El valor de una variable normal puede modificarse después de su declaración: con el estamento **Call**  $x=12*7$  o simplemente  $x=12*7$  dentro del bloque **Calc ... Endcalc**. Su valor también puede modificarse dentro de un bucle del Sweep. Por ejemplo vamos a representar una parte de la función  $f(x)=\sin(\exp(x))$ , utilizando el siguiente fichero de entrada:

Códigos	Comentarios
AplacVar x	
Sweep "Función $y(x)=\sin(\exp(x))$ "	
+ LOOP 101 APLACVAR x LIN 0 3	\$Cambiar la linealidad de $x$ desde 0 a 3 usando
+ Y "y(x)" "" -1 1	\$101puntos.
+ GRID	\$Usar grid en la ventana gráfica
Show Y $\sin(\exp(x))$	
EndSweep	

El resultado se muestra en la figura 2.19 :

fig. 2.19 Función  $\sin(\exp(x))$ 

Otra posibilidad para crear una variable es utilizando corchetes ({}), alrededor de la variable, donde aparece la primera vez. El fichero quedaría de la siguiente forma:

Códigos	Comentarios
<pre>Sweep "Función y(x)= sin (exp(x))" + Loop 101 APLACVAR {x} LIN 0 3 + Y "y(x)" "" -1 1 + GRID Show Y sin(exp(x)) Endsweep</pre>	<p>\$ Los corchetes alrededor x define la variable x.</p>

Si se quiere definir una **constante** por ejemplo 10, se haría como se muestra a continuación:

Códigos	Comentarios
<pre>Aplacvarx=10 CONST</pre>	<p>\$ CONST define una constante cuyo valor no puede ser \$modificado después.</p>

O utilizando la directiva **#define**:

```
#define x 10
```

Si se necesitan constantes, es recomendable utilizar las directivas **#define** o **CONST**, puesto que disminuyen el tiempo de computación del simulador.

### 2.6.3.1.2 Variables Funcionales

Una variable funcional se define de la siguiente manera:

```
Aplacvar y= Sqrt(sin(x))
```

Define la variable “Y” cuyo valor es de  $\sqrt{\sin(x)}$ , y admite nuevos valores cuando se modifica x. Las variables funcionales actualizan su valor automáticamente durante el análisis y no se puede cambiar su valor (excepto por modificación x). Es importante entender la diferencia entre una variable normal y una variable funcional. La variable normal, por ejemplo:

```
Aplacvar x=10
```

x puede redefinirse usando la declaración **Call** pero hay que tener en cuenta que:

```
Call x= 150*sin(z)
```

Redefine el valor de x usando el valor de z en el momento que la declaración **Call** es ejecutada. La declaración **Call** no cambia x desde una variable normal a una variable funcional. Además si z cambia de valor, el valor de x permanece inalterado y no sigue el nuevo valor de z.

Una **Aplacvar** definida por **Declare** es siempre una variable de tipo normal. Hay que tener en cuenta que:

```
Declare  
+ IAPLACVAR y=2*x
```

no es lo mismo que:

```
Aplacvar y=2*x
```

en el último ejemplo “y” es funcionalmente dependiente de x mientras que con la declaración **Declare** “y” es inicializada a 2x (en el momento de la definición) y no seguirá el valor de x automáticamente, si se le asigna un nuevo valor a x.

### 2.6.3.2 Sweep

El bloque **Sweep ... EndSweep** se define para una gran variedad de tareas gráficas o salidas estándar.

Aquí hay un ejemplo de un análisis básico que vimos anteriormente:

Códigos	Comentarios
Sweep "Análisis transitorio"	
+ Loop 2500 TIME LIN 0 1u	\$ Definición de la primera ventana gráfica.
+ WINDOW 0	
+ Y "Vout" "V" 0 5	
+ WINDOW 1	\$ Definición de la segunda ventana gráfica.
+ X "t" "s" 0.9u 1u	
+ Y "Vout" "V" 0 5	
Show Y Vtran(2) PEN=2	\$ Utiliza el color 2 para dibujar la curva.
EndSweep	

El título de ambas ventanas (WINDOW 0 y WINDOW 2) es "Análisis Transistorio". La escala del eje X en la primera ventana WINDOW 0 está definida por defecto por los parámetros del LOOP que van desde 0 a 1 $\mu$ s mientras el eje Y llamado "Vout" tiene una escala desde 0 a 5 y de unidad "V".

Se van a realizar dos gráficos, el primero (WINDOW 0) para producir resultados globales y la segunda (WINDDOW 1) para un análisis más específico, en él último 0.1 $\mu$ s del análisis transitorio. En la segunda ventana definimos la escala para el eje X.

La declaración **Show** (o **Display**) define que se muestra en las ventanas gráficas. En el ejemplo anterior se muestra la tensión transitoria del nodo 2. **Display** es similar a **Show** pero da libertad para dar nombre que queramos a la curva. Por ejemplo en el ejemplo anterior podríamos a ver puesto **Display "load" Y Vtran(2) ...** donde el nombre que le hemos dado a la curva sería "Load". Además de Vtran podíamos haber utilizado otros identificadores como **Y2, PHA, MA, DB, SPECTRUM, CIRCLE, WAVEFORM** que son propiedades versátiles de **Show** y **Display**.

**PEN** define el color de la curva que se va a representar. El argumento de **PEN** es un entero entre 0 y 15. Si no definimos **PEN** y vamos a representar varias curvas en la misma ventana, **PEN** se incrementa automáticamente por

uno en cada curva. Existen una gran variedad de atributos para modificar el modo de presentar las curvas como **LINE**, **MARKER**, **WIDTH**, **TEXTPEN**, **TEXCOLOR**, etc.

El ejemplo anterior podríamos haberlo escrito como se muestra a continuación:

---

#### Códigos

---

```
Sweep "Análisis transitorio"
+ LOOP 2500 TIME LIN 0 1u
      Show WINDOW 0 Y Vtran(2) WINDOW 1 Y Vtran(2)
EndSweep
```

---

La declaración Show garantiza la creación de dos ventanas y ambas auto escalada. Una vez creadas estas ventanas podemos escalar una de ellas para observar de manera específica el punto que queramos del análisis transitorio.

También podemos realizarlo de otra forma:

```
Show Y Vtran(2) window 1 Y Vtran(2)
```

Que muestra dos veces Vtran(2) en WINDOW 1.

Todavía hay otra forma de crear dos ventanas ambas mostrando Vtran (2):

---

#### Códigos

---

```
Sweep "Transient Analysis"
+ LOOP 2500 TIME LIN 0 1u
+ NW 2
      Show Y Vtran(2)
EndSweep
```

---

Donde **NW 2** especifica el número de ventanas a ser creadas en este caso 2.

### 2.6.3.3 Funciones Matemáticas

En este apartado veremos algunas funciones matemáticas ya definidas por APLAC y la forma de realizar nuevas funciones que nos faciliten la creación de circuitos y de sus variables.

#### 2.6.3.3.1 Funciones Matemáticas Definidas por APLAC

APLAC dispone de un gran número de funciones matemáticas ya definidas. Algunas de ella se muestran a continuación:

Función	Unidad de r	Intervalo	tipo	Descripción
Abs(r)			real	$ r $
Acos(r)	[rad]	$-1 \leq r \leq 1$	real	arccos(r)
Acosh(r)	[rad]	$r \geq 1$	real	Arcosh(r)
Asin(r)	[rad]	$-1 \leq r \leq 1$	real	Arcsin(r)
Asinh(r)	[rad]		real	Arsinh(r)
Atan(r)	[rad]	$[-\pi/2, \pi/2]$	real	Arctan(r)
Atanh(r)		$-1 \leq r \leq 1$	real	arctanh
CholeshyDecomp(mr)			entero	Calcula la descomposición de Cholesky de una matriz triangular.
CLinEqu(mrAre, mrAim, mrBre, Mrbin)			entero	Resuelve una o más puntos lineales de ecuaciones complejas.
Conj(c)			complejo	Conjugada de c
Cos(r)	[rad]		real	Cos(r)
Erf(r)			real	Calcula el error de la función definida como $erf(r) = \frac{2}{\sqrt{\pi}} \int_0^r e^{-t^2} dt.$
Exp(r)			real	$e^r$
Img(c)			real	Parte imaginaria de c
J0(r)			real	Funcion de Bessel del primer tipo y de orden 0
Ln(r)		$r \geq 0$	real	Ln(r)
Log10(r)		$r \geq 0$	real	Log(r)

Por ejemplo para representar el valor de la función cos(5):

```
Print REAL cos(5) LF
```

el identificador **REAL** indica que se espera un número real, mientras el **LF** indica el final de la línea.

Si queremos estructurar el resultado podemos definir el número de dígitos a representar usando el identificador **R** (o **FORMAT**) antes de **REAL**. Aunque en la mayoría de los casos el identificador **REAL** debería ser suficiente.

**TAB** permite situar el cursor donde queramos.

Los efectos de varios formatos para representar las funciones se dan en el fichero de entrada a continuación:

```
Print TAB=6 REAL cos(5) TAB=19 REAL sin(-5) LF
Print TAB=3 R " cos=%5.2f" cos(5) TAB=17 R " sin=%f" sin(-5) LF
Print DECIMALS=6 TAB=6 REAL cos(5) TAB=19 REAL sin(-5) LF
Print FORMAT="%4.1f" TAB=8 REAL cos(5) TAB=21 REAL sin(-5) LF
```

La salida sería:

```
283.662m    958.924m
cos= 0.28    sin=0.958924
283.662185m  958.924275m
0.3         1.0
```

Una expresión más compleja podemos usarla en lugar de una función simple como:

```
Print REAL (cos(5) + 5*3.1^4) LF
```

donde  $3.1^4$  significa  $3 \cdot 10^4$ .

Si se tienen espacios como en la expresión anterior se debe utilizar paréntesis puesto que el espacio es un separador.

Un número complejo  $z = a + bj$  puede ser representado en la forma  $(a, b)$ . También podría ser representado en magnitud y fase de la forma  $z \angle \emptyset$  escrito como  $(z \emptyset)$ . Los paréntesis indican que es un número complejo. Los números complejos también pueden usarse en la expresión:

```
Print RI (1, 2)*(5, 2) LF      * es equivalente a
```

Print RI (1, 2)\*(5.385' 21.8) LF

donde el identificador RI que indica que la salida debe ser representada en formato complejo (real part, imaginary part). Otras opciones son MA (magnitud y fase en grados) y DB (magnitud en decibelios y fase en grados).

### 2.6.3.3.2 Realización de Nuevas Funciones

Se pueden crear funciones propias utilizando la declaración **FUNCTION**. Supongamos que queremos utilizar varias veces la función  $f(x) = e^x \sin(x)$ , crearemos una nueva función que llamaremos mi función:

```
Function MiFuncion(x) exp(x) * sin(x)
```

pueden utilizarse espacios libremente en la definición de la función.

Se puede representar la nueva función junto con sus argumentos por ejemplo utilizando el bloque For ... EndFor, el fichero de entrada quedaría como:

```
Function Mifuncion(x) exp(x) * sin(x)
For i 0 10
If (i < 10) Then
Print BL
EndIf
Print REAL i BL REAL MiFunsion(i) LF
EndFor
```

donde **BL** significa espacio en blanco. Un espacio adicional es representado si i es menor de 10. El resultado sería:

```
0.000 0.000
1.000 2.287
2.000 6.719
3.000 2.834
4.000 -41.320
5.000 -142.317
6.000 -112.724
7.000 720.473
8.000 2.949k
9.000 3.339k
10.000 -11.983k
```

### 2.6.3.4 Declaraciones del Preprocesador

En este apartado veremos algunas directivas útiles para el desarrollo de un nuevo modelo:

Se puede definir alias para parámetros y otros conceptos con la declaración **#define**:

```
#define x 10k
#define MODE PHASE
```

En el caso anterior x es remplazado por 10k. Y en el fichero de entrada es cambiado exactamente por 10k. Igualmente, MODE es remplazado por PHASE. Hay que tener en cuenta no incluir ningún espacio en blanco demás en la declaración **#define** pues puede producir algún error y no utilizar “=” en la línea de definición, como x=10k porque el preprocesador reemplazaría x por =10k, ejemplo:

```
Show Y 2*x*Vtran(2)
```

Es remplazado por:

```
Show Y 2*=10k*Vtran
```

lo que produciría un error.

En conjunción con **#define** puedes usar la declaración **#ifdef ... #endif** y/o **#ifdef ... #else ... #endif**. Las líneas entre estas declaraciones son incluidas o no, dependiendo de los valores de **#define** y la prioridad del **#ifdef**, por ejemplo:

```
#define AC
#ifdef AC
.
.
.
#endif
```

Con la declaración **#include**, podemos insertar otros modelos o circuitos en la línea correspondiente. De esta forma es fácil reutilizar códigos generales, como nuevos componentes. Por ejemplo nosotros queremos utilizar un análisis PLL este fichero puede ser encontrado en plldefs.i y lo incluimos al principio de cada fichero donde simulamos los PLLs utilizando la declaración:

```
#include plldefs.i
```

Si el fichero de entrada y la definición del fichero están en diferentes directorios, el recorrido debe ser incluido como sigue:

```
#include /users/martti/aplac/definitions/plldefs.i
```

donde la barra (\) debería ser usado en Win95/NT y (/) en UNIX.

La declaración **#Libdir** define el directorio donde se realiza la primera búsqueda de los ficheros utilizando la directiva **#library**. Si el fichero que se especifica no lo encuentra APLAC en el recorrido definido entonces la búsqueda continua como explica la directiva **#include**.

La declaración **#library** es semejante a **#include** excepto que en **#library** puedes definir directivas **#define** e incluir la parte que queramos, por ejemplo:

```
#libdir /aplac/companymodels
#library bjt.lib MODEL1 MODEL2
```

El efecto de **#define** podría ser terminado por su correspondiente **#undef**, por ejemplo:

```
#define x 10k
* en esta parte cambia x por 10k
#undef x
* en esta parte x esta indefinido
#define x 100k
* en esta parte x es remplazado por 100k
#undef x
```

## 2.6.4 Modelado

### 2.6.4.1 Introducción

En general, los parámetros de un componente se pueden especificar directamente o a través de la referencia del modelo. Un condensador no lineal, por ejemplo, podría ser definido como se muestra a continuación:

```
Cap C1 1 2 1n G= 10Meg
```

o podría ser declarado definiendo el modelo correspondiente primero:

```
Model firmA1n R=1 L=1n G=10Meg
Cap C1 1 2 1n MODEL=FirmA1n
```

La directiva Model nos simplificará circuitos en los cuales tengamos un gran número de componentes de parámetros similares, solamente tendríamos que renombrar los parámetros distintos:

```
Model FirmA1n R=1 L=1n G=10Meg
Cap C1 1 2 1n MODEL=FirmA1n L=2n
Cap C2 2 3 1n MODEL=Fim1n
```

el valor de la inductancia parasitaria del condensador C1 sería 2nF en la simulación (a pesar de que en el modelo fuera de 1nF).

#### 2.6.4.2 Creación de Modelos

Un modelo puede ser creado con el bloque **Defmodel ... EndModel**. Por ejemplo vamos a crear una fuente sinusoidal usando nuestro propio modelo:

---

##### códigos

---

```
DefModel fuentesinoidal 2 Mas Menos PARAM 3 AMP FC PHA
Default AMP=1
Default FC=1
Default PHA=0
Volt Vin Plus Minus TRAN=AMP*cos(2*PI*FC*t + PHA)
EndModel
```

---

El nuevo modelo tiene dos nodos “Mas” y “Menos” y tres parámetros opcionales AMP, FC y PHA los parámetros pueden ser variables, nodos, derivaciones, o modelos. En este caso son variables. También pueden especificarse como **FLAG** y **FIXED**. El uso del nuevo componente es completamente similar al uso de componentes básicos. Si los parámetros no se definen en la llamada al submodelo entonces APLAC tomará el valor que tienen por defecto. La llamada al nuevo componente sería:

```
Fuentesinoidal Vin 1 0 FC=1k
```

se asume que la definición de “fuentesinoidal” esta definida antes de su primera referencia o incluida al principio del fichero con la declaración **#include**.

### 2.6.4.3 Creación de Modelos Lineales

En la creación de modelos lineales utilizaremos normalmente el bloque **DefModel ... EndModel**. Dentro de este bloque crearemos el circuito, la funciones matemáticas, dependencias funcionales, parámetros de entrada, frecuencia, temperatura, etc.

#### 2.6.4.3.1 Ejemplo de un Transformador Híbrido

Considerando un transformador híbrido, el cual consiste en tres inductancias, L1, L2 y L3 con acople mutuo entre todos ellos. Crearemos un macro modelo que llamaremos híbrido:

---

#### Códigos

---

```
DefModel Hibrido 5 n1 n2 n3 n0 n4
+ PARAM 6 L1 L2 L3 K12 K13 K23
Default L1= 0.15H
Default L2= 0.15H
Default L3= 0.15H
Default K12=0.99
Default K13=0.99
Default K23=0.99
Ind L1 n3 n0 L1
Ind L2 n2 n1 L2
Ind L3 n1 n4 L3
Muc M1 L1 L2 K=K12
Muc M2 L1 L3 K=K13
Muc M3 L2 L3 K=K23
EndModel
```

---

el modelo toma los valores de las inductancias y los coeficientes de acoplamiento como parámetros de entrada. Se especifican los valores por defecto para cuando se haga referencia al modelo y no especifiquen los parámetros, los tome por defecto. Como un suplemento podemos utilizar la función **Specified** para chequear si un parámetro se especifica o no.

Si los parámetros de acoplamiento son obligatorios en entonces la definición debería ser:

---

**Códigos**


---

```

DefModel Hibrido 5 n1 n2 n3 n0 n4
+ FIXED 3 K12 K13 K23
+ PARAM 3 L1 L2 L3
Default L1=0.15H
Default L2=0.15H
Default L3=0.15H
Ind L1 n3 n0 L1
Ind L2 n2 n1 L2
Ind L3 n1 n4 L3
Muc M1 L1 L2 K=K12
Muc M2 L1 L3 K=K13
Muc M3 L2 L3 K=K23
EndModel

```

---

en el último caso el modelo debe ser llamado como se muestra a continuación:

```
Hybrid H5 n1 n7 n2 n3 n4 0.95 0.93 0.97 L1=0.1 L3=0.2
```

después del nombre viene los cinco nodos y seguidamente los tres parámetros obligatorios, que se deben introducir en orden especificado en el modelo.

#### 2.6.4.3.2 Fuentes Dependientes

En muchos modelos, las fuentes dependientes se utilizan para simular comportamientos eléctricos. **Csource** define los cuatro tipos básicos de fuentes dependientes lineales. **Gm** es sinónimo de fuente de corriente dependiente de tensión. Para **Csource** y **Gm**, tienen opción especial **DELAY** que nos permite definir tiempo de retardo del modelo.

Podemos definir una dependencia de frecuencia de las fuentes haciendo referencia a la variable **f**. Las dependencias de frecuencia funciona bien en AC y en análisis armónico pero no transitorio.

#### 2.6.4.3.3 Jerarquía de Modelos

Debemos definir los modelos creados en paralelo, es decir, dentro del bloque **DefModel** no podemos definir otro bloque **DefModel**.

En un mismo fichero podemos almacenar todos los nuevos modelos que realicemos, poniéndolos en paralelo. Si queremos utilizar uno o varios de estos modelos podemos hacerlo utilizando la directiva **#include**. Por ejemplo, supongamos que el fichero donde tenemos todos nuestros modelos se llame “milibreria” la forma incluirla en otro fichero sería:

```
#include milibreria.i
```

En el caso de que nuestra librería fuera grande y sólo queremos utilizar parte de ella podemos utilizar la directiva **#library**.

#### 2.6.4.4 Creación de Modelos No Lineales

Todos los modelos no lineales son creados como macromodelos utilizando los elementos básicos de modelado: **VCCS**, **VCVS**, **CCCS**, **CCVS**.

##### 2.7.5.4.1 Ejemplo de un Modelo de Diodo

En la figura 2.20 se encuentra el modelo del diodo que queremos crear. La corriente  $i$  se define como  $i = I_S [\exp(\frac{u}{\eta V_T}) - 1]$ , donde  $I_S$  y  $\eta$  son parámetros constantes del modelo y  $V_T$  es la tensión equivalente de temperatura.

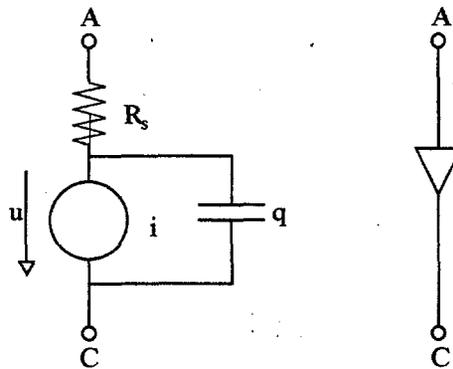


fig. 2.20 Circuito equivalente y símbolo de diodo

El modelado de la carga  $q$  lo veremos en la próxima sección. La parte estática que es la resistencia  $R_s$  y la fuente de corriente dependiente  $i$ , pueden ser implementadas como:

**Códigos**

```
DefModel NewDiode 2 A C
+ PARAM 3 RS IS N
Function ud=CV(0)
Function vt=Temp/11605
Function id=IS*(exp(ud/(N*vt))-1)
Res rs A int_node RS
VCCS DiodeVccs int_node C 1 int_node C
+ id G NONLINEAR NO_DERIV
EndModel
```

La función **CV** regresa el valor de la tensión de control: no pueden realizarse referencias directas a **Vdc**, **Vac** etc. en el interior del modelo.

Una **VCCS** puede tener un número indeterminado de tensiones de control. **CV** toma un índice como argumento, el índice comienza en 0 refiriéndose a la primera tensión de control.

Aunque APLAC tiene un algoritmo efectivo de derivación,

$$g = \frac{\partial i}{\partial u} \tag{2.1}$$

deberíamos incluir la derivada y así mejorar el tiempo de simulación. Una implementación mejorada del diodo quedaría de esta forma:

Códigos	Descripción
<pre>DefModel NewDiode 2 A C + PARAM 3 RS IS N Function ud=CV(0) Function vt=Temp/11605 Function i_and_g {nvt, expo} [ + nvt = N.* vt; + expo = exp(ud / nvt); + IS * (expo - 1), + IS * expo / nvt ] Res rs A int_node RS VCCS DiodeVccs int_node C 1 int_node C + i_and_g G NONLINEAR DERIV EndModel</pre>	<p>\$ Corriente, la coma se utiliza como separador...</p> <p>\$ Derivada = conductancia</p>

### 2.7.5.4.2 Parámetros de Temperatura

En el ejemplo del diodo utilizamos **Temp** aunque no la incluimos en la lista parámetros de entrada. Cada bloque DefModel automáticamente reconoce seis parámetros de temperatura, **TEMP** (alias **TEMPK**), **TEMPC**, **TNOM** (alias **TREF**) y **DTEMP** cuales son temperaturas en grados Kelvin y Celsius. Con estos parámetros podremos modelar la temperatura de nuestro circuito, por ejemplo:

```
NewDiode D1 n1 n2 IS=1E-13 N=1.5 RS=6 TEMPC=40
NewDiode D2 n1 n3 IS=1E-15 N=1.5 RS=3 DTEMP=30 TREF=35
```

Si no especificamos la temperatura se tomará por defecto la temperatura ambiente en la definición.

En el segundo diodo la temperatura es de 30 grados por encima de la temperatura ambiente y la temperatura de referencia es de 35 grados

### 2.7.5.4.3 Creación de Modelos de Carga No Lineal

Las fuentes tipo G, fuentes de corrientes controlada por tensión, deben ser estáticas: sus dependencias no lineales están basadas solamente en valores instantáneos de las cantidades controladas y no tienen memoria. Las fuentes dinámicas no lineales pueden definir una carga (capacidades no lineales, **VCCS**) o flujo ( inductancias no lineales, **CCVS**) o una combinación híbrida (**VCVS**, **CCCS**). A continuación describiremos cargas no lineales.

#### 2.7.5.4.3.1 Carga con una Tensión de Control

Una carga o capacidad no lineal se modelan usando **VCCS** con la opción C. La ecuación primaria entre la capacidad y la carga es:

$$C(u) = \frac{\partial q}{\partial u} \quad (2.2)$$

Se puede definir una ecuación para la capacidad en lugar de q.

Considerando como ejemplo el modelado de una capacidad no lineal:

$$C(u) = \frac{C_{j0}}{\sqrt{1-u/\emptyset}} \quad (2.3)$$

Donde  $C_{j0}$  y  $\emptyset$  son parámetros constantes. En orden a implementar este modelo en APLAC, la ecuación para la carga debe ser obtenido por integración analítica:

$$q(u) = 2C_{j0}\emptyset(1 - \sqrt{1-u/\emptyset}). \quad (2.4)$$

Las constantes de integración han sido elegidas como  $q=0$  para  $u=0$ . La implementación en APLAC que daría de la siguiente forma:

---

#### Códigos

---

```
DefModel JunctionCap 2 n1 n2
+ PARAM 2 CJO PHI
Function u=CV(0)
Function q_and_c {local} [
+ root = sqrt(1 - u / PHI);
+ 2 * CJO * PHI * (1 - local),
+ CJO / local ]
VCCS JunctionCapVccs n1 n2 1 n1 n2 q_and_c C
+ NONLINEAR DERIV
EndModel
```

---

La función **q\_and\_c** nos da dos valores, separada por una coma. Los corchetes indican a APLAC que la función devuelve más de un resultado. Dentro de la ecuación se pueden definir un número indefinido de expresiones separadas entre ellas por comas. Y {local} declara una variable local.

#### 2.7.5.4.3.1 Carga con Varias Tensiones de Control

Si hay varias tensiones de control, debemos conocer las cargas parciales. Como ejemplo podemos considerar un FET, cuya carga se almacena debido a la carga puerta-fuente y puerta-drenador,  $q_{gs}$  y  $q_{gd}$  como función de las tensiones  $u_{gs}$  y  $u_{gd}$ . Muchos modelos solo expresan la carga total de puerta,

$$q_g = q_{gs} + q_{ds} \quad (2.5)$$

y la capacidad no lineal

$$C_{gs} = \frac{\partial q_g}{\partial u_{gs}}, \quad C_{gd} = \frac{\partial q_g}{\partial u_{gd}} \quad (2.6)$$

pero esto no garantiza la conservación de la carga y no es compatible método de equilibrio armónico multitono.

Las ecuaciones primarias que se necesitan para el modelo son  $q_{gs}$  y  $q_{gd}$ . Se implementan usando dos elementos VCCS por separado con la opción C. Es recomendable calcular la derivadas y utilizar la opción NO\_DERIV. Ambos elementos VCCS necesitan dos derivadas,

$$C_{gs,gs} = \frac{\partial q_{gs}}{\partial u_{gs}}, \quad C_{gs,gd} = \frac{\partial q_{gs}}{\partial u_{gd}} \quad (2.5)$$

para la carga  $q_{gs}$  y

$$C_{gd,gs} = \frac{\partial q_{gd}}{\partial u_{gs}}, \quad C_{gd,gd} = \frac{\partial q_{gd}}{\partial u_{gd}} \quad (2.6)$$

para la otra carga.

# Estructura y principio de funcionamiento del HFET

El desarrollo de las técnicas MBE (*molecular beam epitaxial*) y MOCVD (*metal-organic chemical vapor deposition*) desde finales de los años 70, ha revolucionado el diseño de nuevos dispositivos electrónicos y optoelectrónicos. En nuestros días se pueden crecer capas muy finas de material semiconductor manteniendo un fino control sobre el dopaje y las conexiones. Este desarrollo tecnológico ha propiciado la consolidación de una disciplina científica: la "ingeniería de la anchura de la banda prohibida" (*band-gap engineering*), que estudia las heterouniones. Una heterounión es una estructura constituida por la unión de dos semiconductores distintos y, por ende, con diferente anchura de la banda prohibida.

En las estructuras convencionales (homouniones: el mismo semiconductor a ambos lados de la unión) los electrones y los huecos experimentan fuerzas proporcionales a los gradientes de concentración de impurezas (difusión) y al campo eléctrico aplicado (arrastre). Las fuerzas electrostáticas serán iguales y opuestas en electrones y huecos debido a que la anchura de la banda prohibida es constante. Las heterouniones ofrecen un grado adicional de libertad en el diseño de dispositivos, ya que las fuerzas que actúan sobre los electrones y los huecos pueden ser controladas de forma independiente mediante el ajuste de los anchos de banda prohibida (es decir, de la composición) y del dopaje de los semiconductores.

El desarrollo de uniones de buena calidad entre semiconductores binarios (III-V) y sus aleaciones (III-III-V o III-V-V) es posible si las constantes de red de



ambas funciones lineales de la fracción molar del Al ( $x_{Al}$ ). Obsérvese que los *offsets* no dependen del dopaje; sino sólo de la composición.

En la figura 3.2.b los semiconductores se han puesto en contacto. Como no hay ninguna tensión aplicada a la heterounión, los niveles de Fermi estarán alineados. En esta figura no se han tenido en cuenta los *offsets*. En las figuras 3.2.c y d se puede observar cómo la inclusión de los *offsets* en los diagramas de bandas produce que éstos se curven, apareciendo, en el caso presentado, una discontinuidad en la banda de conducción del sistema en equilibrio. A esta discontinuidad o “pico” se alude más adelante.

El *offset* de la banda de valencia  $\Delta E_V$  se encuentra justo entre las bandas de valencia de ambos semiconductores, de forma que, la pendiente en el GaAs y en el AlGaAs es la misma a ambos lados de la interfase. La ecuación de Poisson exige que la pendiente de las bandas de energía sean proporcionales al campo eléctrico

$$\nabla^2 \Phi = \frac{\rho}{\epsilon_0} \Rightarrow \nabla \Phi = \frac{\rho}{\epsilon_0} \Rightarrow -\nabla \Phi = E \Rightarrow pte \propto E$$

Como la permitividad es la misma y el campo eléctrico debe ser continuo en la unión de ambos semiconductores, se deberá cumplir que la pendiente de las bandas de energía sea la misma a ambos lados. En el AlGaAs, si la concentración de impurezas donadoras es uniforme, la curvatura es parabólica.

Las bandas de conducción deben seguir a sus respectivas bandas de valencia debido a que la anchura de la banda prohibida es constante para un mismo semiconductor. En la figura 3.2.d se observa cómo se unen las bandas de conducción, siendo el *offset*  $\Delta E_C$  la altura de la discontinuidad. El pico que se observa en la banda de conducción del AlGaAs es una región que está vacía de electrones. Es decir, esta zona será una zona de deplexión en la que habrá una carga fija positiva asociada a las impurezas donadoras ionizadas. La carga fija negativa correspondiente que se requiere para que se dé la condición de neutralidad en la interfase estará en el valle de la banda de conducción, el cual actúa como pozo de potencial para los electrones de conducción.

En el siguiente apartado se presenta una descripción de los transistores de efecto de campo de heteroestructura.

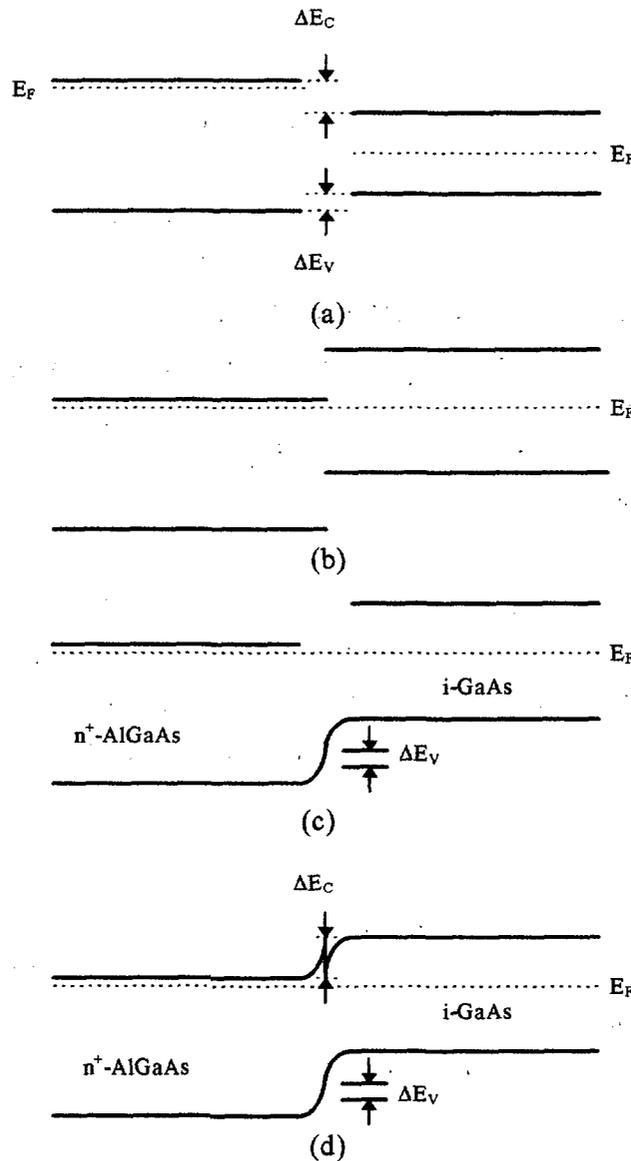


fig. 3.2 Construcción de un diagrama de bandas de energía de una heterounión: (a) antes del contacto, (b) después del contacto los niveles de Fermi quedan alineados, no se consideran los *offsets*, (c) incluye el *offset* de la BV, (d) incluye el *offset* de la BC.

### 3.2 Transistores de Efecto de Campo de Heteroestructura

La aparición de los Transistores de Efecto de Campo de Heteroestructura (HFET) ha supuesto casi una revolución en el campo de las microondas de ultra-alta velocidad y de los circuitos electrónicos digitales. Los HFETs muestran prestaciones extremadamente elevadas, encontrándose HFETs discretos de microondas con figuras de ruido de 1.3 dB a 60 GHz y ganancias de 9.5 dB. En relación a los circuitos digitales, se han conseguido circuitos con retardos de propagación de 10 ps por puerta y memorias SRAM de tiempo de acceso de 0.5 ns.

La mayor parte del trabajo realizado sobre los HFETs se basa en dispositivos de canal tipo n con heteroestructuras AlGaAs/GaAs. Como se vio en el apartado anterior, en este tipo de uniones, se hace crecer un material dopado de ancho de banda prohibida grande (AlGaAs) sobre otro no dopado de ancho de banda prohibida pequeño (GaAs). A este tipo de estructuras se las denomina “*modulation doped heterostructure*”. La estructura física de la heterounión es tal que el transporte de carga dentro del dispositivo electrónico se puede optimizar ajustando las anchuras de las capas y los niveles de dopaje.

Los HFETs tienen varias denominaciones, como son SDHTs (*Selectively Doped Heterostructure Transistors*), HEMTs (*High Electron Mobility Transistors*), TEGFETs (*Two-dimensional Electron Gas FETs*), y MODFETs (*Modulation Doped FETs*). Nosotros utilizaremos indistintamente las denominaciones HFET y HEMT. Si bien la estructura más utilizada es la  $n^+$ -AlGaAs/GaAs, también nos podemos encontrar con otro tipo de estructuras como pueden ser: AlGaAs-InGaAs-GaAs (dopado homogéneamente y simple y doble  $\delta$ -dopado), AlInAs-GaInAs-InP, etc..

En el apartado 3.2.1 presentamos la estructura básica de los HFETs y su principio de funcionamiento. En el apartado 3.2.2, haremos una descripción de las diferentes variantes a la estructura HFET convencional con las que nos podemos encontrar. Por último, acabaremos este capítulo haciendo un repaso de las principales aplicaciones que tienen los HFETs. Este será el objetivo del apartado 3.2.3.

### 3.2.1 Estructura de Bandas y Comportamiento del Dispositivo

En este apartado presentamos las características físicas de los HFETs. En primer lugar haremos una descripción de la estructura convencional de estos transistores, y seguidamente pasamos a explicar su principio de funcionamiento. Para terminar se comentarán algunas consideraciones tecnológicas de la fabricación de los HFETs.

#### 3.2.1.1 Estructura convencional de los HFETs de $n^+$ -AlGaAs/GaAs

En la figura 3.3 se muestra la sección transversal de un HFET convencional. Los contactos de fuente y de drenador son óhmicos, como los de los

transistores MOS de Si o los MESFETs de GaAs. La puerta se implementa mediante una barrera Schottky. La estructura de capas epitaxiales del dispositivo se muestra en la figura 3.4. El dispositivo se hace crecer a partir de un sustrato de GaAs. La concentración intrínseca de portadores en el sustrato de GaAs intrínseco es baja ( $n_i = 2.3 \cdot 10^6 \text{ cm}^{-3}$  a temperatura ambiente) como resultado de la amplia banda prohibida del GaAs. Es por ello por lo que el material intrínseco tiene una resistividad del orden de  $10^8 \Omega\text{cm}$  por lo que se le suele llamar material semi-aislante. Sobre el sustrato semi-aislante se hace crecer una capa de GaAs sin dopar o ligeramente dopada tipo p denominada *buffer* del orden de  $1 \mu\text{m}$  de profundidad. Sobre ésta se hace crecer otra capa sin dopar de  $10\text{nm}$  de AlGaAs denominada capa espaciadora, y a continuación otra de  $n^+$ -AlGaAs fuertemente dopada del orden de  $50$  a  $100 \text{ nm}$  llamada capa donadora. La capa de AlGaAs no dopada separa los electrones de las impurezas del canal. Por último, se hace crecer una capa de  $n^+$ -GaAs denominada *cap*. Las anchuras y dopajes de las diferentes capas influyen directamente sobre las propiedades del HFET. Las longitudes de puerta varían desde  $1.0$  a  $0.1 \mu\text{m}$  de acuerdo con las restricciones de velocidad, aplicación y *yield* que se requieran.

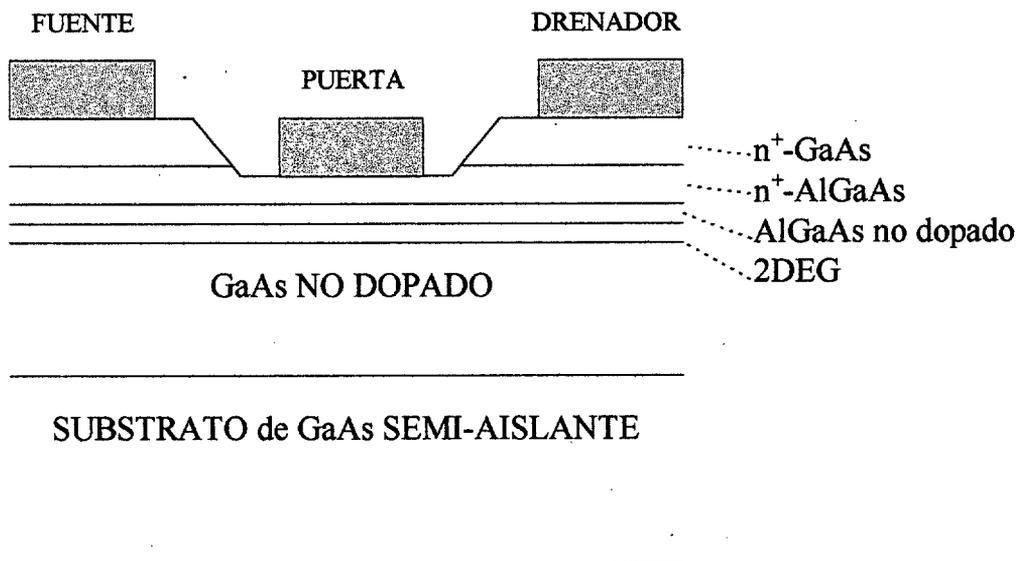


fig. 3.3 Sección transversal de un HFET convencional.

### 3.2.1.2 Principio de funcionamiento

Como vimos en el apartado 3.1, al formarse la heterounión se crea un valle en la banda de conducción. Los electrones se acumulan en este valle o pozo de

potencial formando una zona de carga superficial análoga al canal de inversión que se forma en la estructura metal-óxido semiconductor SiO<sub>2</sub>/Si de los MOS.

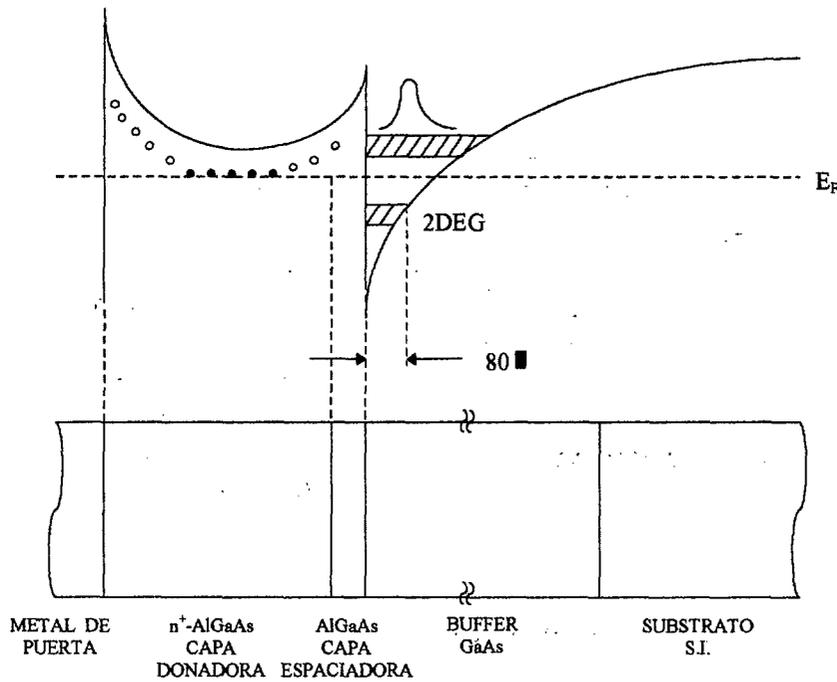


fig. 3.4 Estructura de capas epitaxiales y diagrama de bandas de energía para un HFET con polarización positiva.

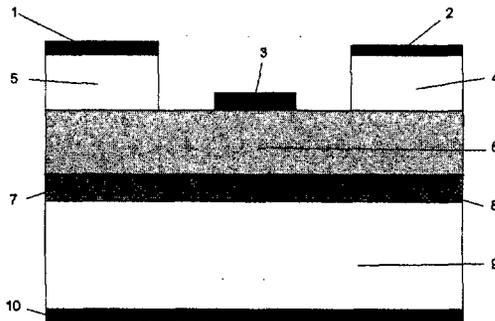
La anchura de este canal es muy pequeña con lo que los electrones quedan atrapados en un sistema bidimensional en la heterointerfase. Este es el motivo por el cual el canal de los HFETs se denomina gas de electrones bidimensional (2DEG: *two-dimensional electron gas*). La separación física de los electrones de las impurezas donadoras reduce la dispersión por impurezas y por tanto aumenta la movilidad así como la velocidad efectiva de los electrones bajo la influencia de un campo eléctrico. Esta superficie o lámina de electrones de elevada movilidad se puede utilizar como canal activo de un FET y se puede modular por el efecto de un campo desde un electrodo de puerta. Esta es en definitiva la base del principio de funcionamiento de los HFETs.

### 3.2.1.3 Consideraciones tecnológicas

En la figura 3.5 se muestra la estructura típica de un HFET en más detalle.

El primer paso en la fabricación del dispositivo es su aislamiento eléctrico por medio de ataque químico “MESA” bajo el canal no dopado o el sustrato semi-

aislante. Éstas regiones se aprecian en la figura a ambos lados de la fuente y el drenador.



1. Electrodo de fuente.
2. Electrodo de drenador.
3. Contacto metálico de la puerta Schottky.
4. Drenador. GaAs tipo n.  $N_D=2 \cdot 10^{20} \text{ cm}^{-3}$ .
5. Fuente. GaAs tipo n.  $N_D=2 \cdot 10^{20} \text{ cm}^{-3}$ .
6. AlGaAs tipo n.  $N_D=8 \cdot 10^{17} \text{ cm}^{-3}$ .
7. Dopaje delta tipo n.  $N_D=1.3 \cdot 10^{19} \text{ cm}^{-3}$ .
8. AlGaAs intrínseco.
9. GaAs intrínseco.
10. Electrodo del sustrato.

fig. 3.5 Estructura típica detallada de un HFET

Entonces se definen las áreas de fuente y drenador con fotorresistencia positiva y se evapora AuGe/Ni/Au como metal de contacto en fuente y drenador. El contacto metálico forma aleación durante 1 minuto a la temperatura de 44°C.

Como casi todas las dimensiones están por debajo de la micra, el tamaño y forma de los contactos óhmicos es crucial en el funcionamiento de estos dispositivos. Estas resistencias deben minimizarse todo lo posible, por ello, durante este proceso se difunde Ge para que haga contacto con el gas bidimensional.

A continuación se define la puerta generalmente mediante borrado químico o ataque de iones reactivos, con la ayuda de alguna capa que detenga el ataque. La profundidad del hueco se selecciona dependiendo de si se quiere que el dispositivo opere en modo de depleción o enriquecimiento. En modo de depleción, el grosor de la capa dopada debe ser tal que la barrera Schottky la vacíe de portadores, pero no vacíe el canal. En modo de enriquecimiento, la capa del AlGaAs dopado es más fina, de modo que el canal está también vacío.

El estrés mecánico causado por las interconexiones metálicas es un serio problema en dispositivos VLSI. La causa es el distinto coeficiente de expansión térmica de los metales y materiales dieléctricos de los del sustrato (GaAs, InP o Si). Cuando el estrés es muy elevado, puede llevar a la rotura de la capa o fallo en la interfase, con lo que se inutiliza el dispositivo.

Para reparar los daños por la exposición y activar las especies implantadas se requiere un proceso de recocido a temperatura elevada. Un método es colocar la muestra sobre una plataforma con poca masa térmica y calentarla con infrarrojos desde una lámpara de cuarzo a la temperatura de 850°C. Después de 15 segundos se apaga la lámpara y se deja enfriar por debajo de 400°C.

### 3.2.2 Familias de los dispositivos HFETs

Las tres aplicaciones más importantes de los HFETs surgen con la necesidad de obtener dispositivos FET con elevada capacidad de suministrar corriente para circuitos integrados digitales, transistores FET de potencia con corrientes y tensiones de ruptura elevadas y transistores FET de bajo ruido para microondas y ondas milimétricas. Para los circuitos digitales y para los FETs de potencia, las características más importantes exigibles a los dispositivos son una transconductancia extrínseca elevada y un amplio rango de puntos de operación posibles. Para el caso de los dispositivos de microondas, la transconductancia pico es importante, y en todos los casos, es deseable el tener una capacidad puerta-canal mínima para que la modulación en la carga del canal sea efectiva. El confinamiento de los portadores en un canal bien definido elimina dos efectos indeseados. Uno es la transferencia de portadores desde el canal hasta las capas de confinamiento de anchos de banda mayores, lo cual traería consigo la formación de un MESFET paralelo en el AlGaAs. El segundo es la inyección de portadores en el sustrato, la cual traería consigo un incremento en la conductancia de salida del transistor, especialmente en dispositivos con anchura del canal pequeña. Por último, la necesidad de tener un buen comportamiento en dispositivos con anchura del canal pequeña, donde se requiere la mayor velocidad posible, se hace crítica para una tecnología HFET.

Siguiendo estas sencillas pautas de trabajo, podemos encontrar dos variantes básicas de HFETs. Una consiste en un dispositivo cuya estructura se optimiza para conseguir el mejor confinamiento del canal posible. Este tipo de HFET requiere un ajuste de las anchuras y composiciones de las capas. La segunda variante consiste en un HFET cuyo dopaje es distribuido para permitir que la carga pueble el dispositivo. El dopaje se puede hacer a través de una capa dopada en volumen (*bulk-doped layer*), una capa dopada en  $\delta$  ( $\delta$ -*doped layer*), una super red (*superlattice*), dopando el propio canal de conducción, o creando un canal de inversión, sin dopaje. En la tabla 3-1 se muestra los distintos tipos de HFETs que

podemos encontrar. Los diagramas de bandas de los diferentes tipos de HFETs se muestran en la figura 3.6. A continuación se muestra la familias de los dispositivos HFETs:

**Tabla 3-1 Familia de los dispositivos HFETs**

Capa Donadora	◆ Dopaje Selectivo	<ul style="list-style-type: none"> <li>● Capa dopada en volumen (SDHT, MODFET, TEGFET, HEMT, etc)</li> <li>● Capa dopada en <math>\delta</math></li> <li>● Super red</li> </ul>
	◆ Puerta Aislada	<ul style="list-style-type: none"> <li>● MISFET</li> <li>● SISFET</li> </ul>
Confinamiento del Canal	<ul style="list-style-type: none"> <li>◆ Canal de pozo cuántico (SQW, etc.)</li> <li>◆ Estructura invertida (I-HEMT, I<sup>2</sup>-HEMT, etc.)</li> </ul>	
Dopaje del Canal	<ul style="list-style-type: none"> <li>◆ No Dopado</li> <li>◆ Dopado (DMT, etc.)</li> </ul>	

### 3.2.3 Aplicaciones de los HFETs

Los dispositivos y circuitos HFETs son adecuados para su uso en aplicaciones de alta velocidad, bajo ruido, y baja potencia. Otras aplicaciones para las que pueden ser útiles estos dispositivos son, por un lado las comunicaciones, y por el otro el procesamiento de datos. En cuanto a las aplicaciones en comunicaciones podemos encontrar enlaces radar, emisión directa de televisión por satélite, telefonía móvil, convertidores de televisión por cable, etc.. En cuanto a las aplicaciones de procesamiento de datos podemos encontrar supercomputadores, conmutadores para redes ATM , etc., aplicaciones todas ellas en las que las características de velocidad elevada y consumo de potencia bajo son críticas.

También existe un elevado número de aplicaciones militares posibles para los HFETs, en particular las antenas radar del tipo array en fase (*phased-array antenna radar*) . Estas consisten básicamente en un gran número de transistores individuales conectados en un circuito integrado que operan a frecuencias de microondas. El éxito de este tipo de antena está en la miniaturización, lo cual requiere un compromiso en la adaptación de los transistores con los otros elementos en el circuito de microondas. Los transistores normalmente no pueden operar en sus condiciones óptimas, y por lo tanto, para conseguir unas prestaciones aceptables, se necesitan dispositivos de muy buena calidad de forma que, incluso con sus prestaciones degradadas, estas todavía se adecuen a su función.

Por último, una aplicación que puede ser fundamental es la de la televisión de alta definición, en donde la complejidad de los circuitos electrónicos requieren potencias de disipación de los transistores individuales mínimas.

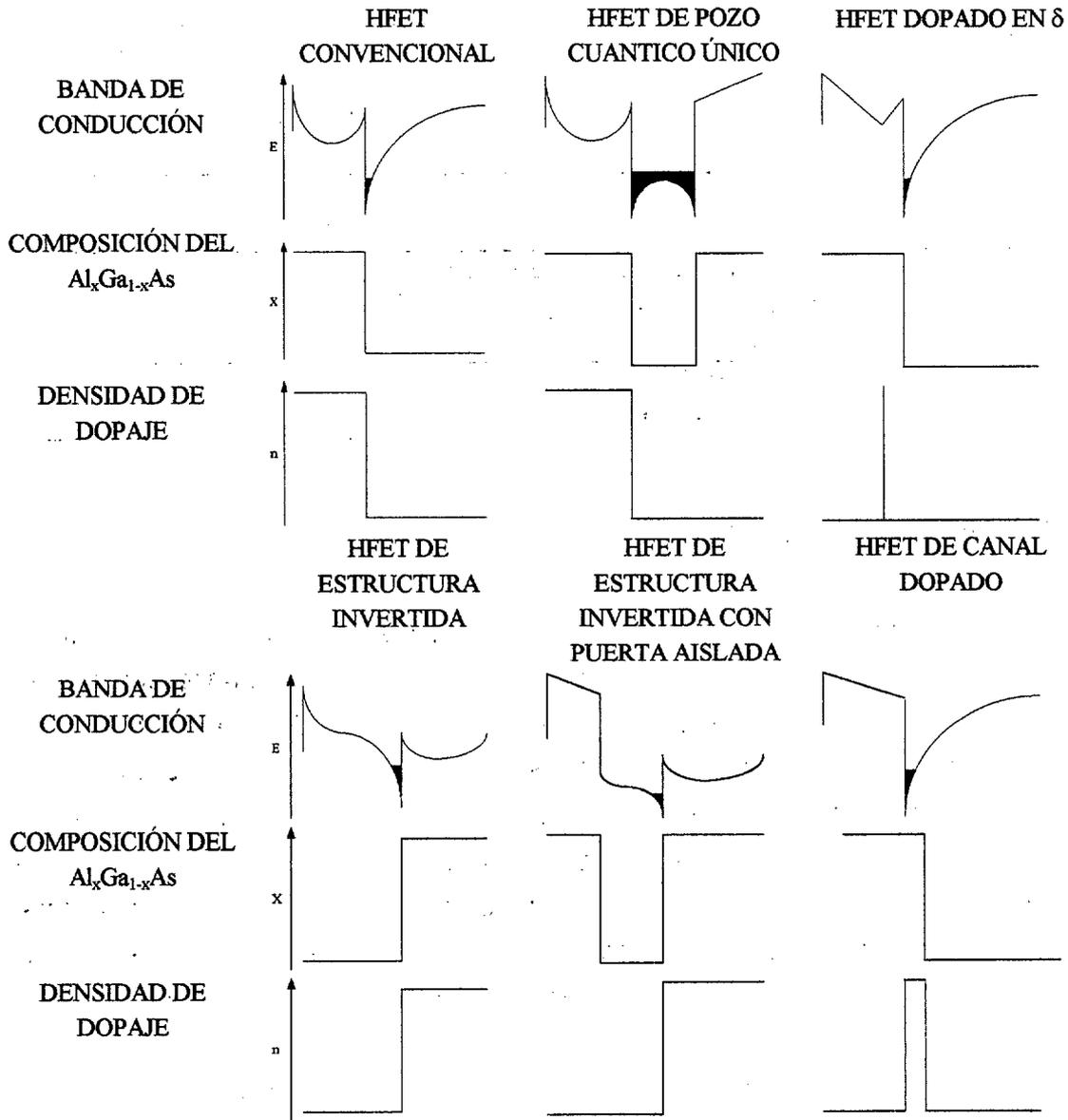


fig. 3.6 Diagramas de Bandas, Composición, y Dopaje de los diferentes HFETs

# Modelos del HFET

En la actualidad, los HFETs han demostrado ser los dispositivos óptimos para las aplicaciones de bajo ruido y ondas milimétricas. El uso de dichos dispositivos en aplicaciones no lineales, en la actualidad menos desarrollado, es un hecho de importancia, debido al incremento de la demanda de la integración MMIC de aplicaciones de ondas milimétricas no lineales. Para esas aplicaciones, los MESFETs no son aplicables debido a su limitado rango de frecuencias. Sin embargo, mientras que la caracterización del comportamiento no lineal de dichos componentes es ya bien conocido, la caracterización de los HFETs está todavía en su infancia.

En cuanto a las aplicaciones digitales, las tecnologías basadas en HFETs están todavía en proceso de maduración. No obstante, sus prestaciones en velocidad y consumo, así como las crecientes posibilidades de integración, las hace aparecer como una evolución probable de las futuras tecnologías digitales de GaAs. Las familias lógicas utilizadas y las metodologías de diseño utilizadas son normalmente desarrollo directo de las usadas con MESFETs sobre GaAs. Así, la familia lógica utilizada normalmente es la Direct Coupled FET Logic (DCFL) por su simplicidad y prestaciones.

Las herramientas de diseño para aplicaciones en gran señal de MESFETs y HFETs, requieren idealmente, un único modelo CAD que pueda proporcionar descripciones precisas del comportamiento no lineal de y ac sobre un amplio rango de puntos de polarización.

Al principio se hicieron intentos para utilizar el modelo convencional de los MESFETs para los HFETs, y así aparecieron los modelos de Curtice [10], Curtice-Ettemberg [11] y Materka [12]. Esos primeros esfuerzos fueron sólo parcialmente exitosos debido a las características peculiares de los HFETs. De hecho, aunque los MESFETs y los HFETs admiten el mismo circuito equivalente para pequeña señal y ambos exhiben un comportamiento en frecuencia similar hay diferencias en su funcionamiento. Quizá la discrepancia más importante es la compresión de la transconductancia,  $g_m$ , para bajas alimentaciones de puerta observadas en los HFETs y que no se detecta en los MESFETs.

Por lo tanto, empezaremos este capítulo viendo con detenimiento el modelo que más aceptación ha tenido para la simulación de los HFETs (apartado 4.1). Dicho modelo, es el de Angelov [13] a partir del cual se han derivado la mayoría de los modelos existentes para los HFETs. Un ejemplo de esto es el modelo del FhGIAF que es el modelo que hemos implementado nosotros y por ello su presentación será el objetivo del siguiente apartado (apartado 4.2).

## 4.1 Modelo de Angelov

### 4.1.1 Introducción

En la actualidad existen varios modelos empíricos para la simulación de MESFETs y HFETs en circuitos no lineales. Estos modelos se usan para predecir la ganancia, la distorsión por intermodulación, la generación de armónicos, etc., en función de la alimentación, para circuitos tales como amplificadores, mezcladores, y multiplicadores. En estos modelos no tienen en cuenta el hecho de que no sólo se tiene que modelar de forma correcta la característica corriente-tensión  $I_{ds}[V_{gs}, V_{ds}]$  sino que también hay que tener en cuenta sus derivadas, especialmente sí se quiere que el modelo sea utilizado para predecir la distorsión por intermodulación.

El modelo propuesto por Angelov [13] consiste en un modelo nuevo y simple, en donde la extracción de parámetros se puede llevar a cabo mediante simple inspección de las características de experimentales  $I_d[V_{gs}, V_{ds}]$  Y  $g_m(V_{gs})$ . Este modelo ha sido utilizado con buenos resultados en el modelado de la  $I_{ds}$  y sus derivadas.

### 4.1.2 El Modelo

Al igual que en los modelos anteriores, en el modelo de Angelov la corriente de drenador se expresa de la siguiente forma:

$$I_{ds}[V_{gs}, V_{ds}] = I_{dA}[V_{gs}] \cdot I_{dB}[V_{ds}] \quad (4.1)$$

donde el primer factor sólo depende de la tensión de puerta, y el segundo de la tensión de drenador.

$$I_{dB}[V_{ds}] = (1 + \lambda \cdot V_{ds}) \cdot \tanh(\alpha \cdot V_{ds}) \quad (4.1.1)$$

Como vemos, la corriente de drenador tendrá una variación con  $V_{ds}$  con forma de tangente hiperbólica viéndose afectada su amplitud por la modulación del canal. Sin embargo, el modelo de Angelov propone una nueva función para  $I_{dA}[V_{gs}]$ , cuya primera derivada tiene la misma forma en "campana" que la función de transconductancia medida  $g_m[V_{gs}]$ . La tangente hiperbólica describe bien la dependencia con la tensión de puerta y sus derivadas. La función  $I_{dA}[V_{gs}]$  propuesta es la siguiente:

$$I_{dA}[V_{gs}] = I_{pk} \cdot (1 + \tanh(\Psi)) \quad (4.1.2)$$

siendo la expresión total para  $I_{ds}$  la siguiente:

$$I_{ds}[V_{gs}, V_{ds}] = I_{pk} \cdot (1 + \tanh(\Psi)) \cdot (1 + \lambda \cdot V_{ds}) \cdot \tanh(\alpha \cdot V_{ds}) \quad (4.2)$$

Donde  $I_{pk}$  es la corriente de drenador para la cual tenemos una transconductancia máxima, sin tener en cuenta la contribución de la conductancia de salida (es decir, con la contribución de la conductancia de salida, restada).  $\lambda$  es el parámetro de modulación del canal y  $\alpha$  es el parámetro de la tensión de saturación. Los parámetros  $\alpha$  y  $\lambda$  son los mismos que en el modelo de Curtice.  $\Psi$  es en general una función potencial en serie centrada en  $V_{pk}$  y con  $V_{gs}$  como variable, es decir:

$$\Psi = P_1 \cdot (V_{gs} - V_{pk}) + P_2 \cdot (V_{gs} - V_{pk})^2 + P_3 \cdot (V_{gs} - V_{pk})^3 + \dots \quad (4.3)$$

$V_{pk}$  es la tensión de puerta para la transconductancia máxima  $g_{mpk}$ . La Función  $I_{ds}[V_{gs}, V_{ds}]$  seleccionada posee derivadas bien definidas. Una ventaja del modelo es, su simplicidad. Los diferentes parámetros pueden ser obtenidos como primera aproximación mediante la inspección de la curva  $I_{ds}[V_{gs}, V_{ds}]$  medidas en las siguientes condiciones de canal saturado: se suponen nulos todos los términos superiores de  $\psi$ ,  $\lambda$  se obtiene de la pendiente de la característica  $I_{ds}$ - $V_{ds}$ ,  $I_{pk}$  y  $V_{pk}$  se determinarían como los valores de la  $I_{ds}$  y la  $V_{gs}$  en la transconductancia pico:  $g_{mpk}$ . La transconductancia intrínseca: máxima  $g_{gmkm}$  se calcula de la transconductancia máxima medida  $g_{mpkm}$  teniendo en cuenta el efecto de realimentación debido a la resistencia del surtidor,  $R_s$  la cual se puede obtener de las medidas en dc:

$$g_{mpk} = \frac{g_{mpkm}}{(1 - R_s \cdot g_{mpkm})} \quad (4.4)$$

$P_1$  se obtiene ahora como:

$$P_1 = \frac{g_{mpkm}}{(1 - R_s \cdot g_{mpkm})} \quad (4.5)$$

En algunos HFETs,  $V_{pk}$  es ligeramente dependiente con la tensión de drenador  $V_{ds}$  en la región de saturación. Este efecto puede ser tenido en cuenta mediante:

$$V_{pk} = V_{pko} + \gamma \cdot V_{ds} \quad (4.6)$$

En la región no saturada y para una  $V_{ds}$  negativa,  $V_{pk}$  variará de forma considerable con  $V_{ds}$ . Si queremos que el modelo prediga el funcionamiento del transistor correctamente, debemos encontrar la dependencia de  $V_{pk}$  con  $V_{ds}$  (experimentalmente o modelada).

Para el modelado de las dependencias de las capacidades  $C_{gs}$  y  $C_{gd}$ , con la tensión de puerta y de drenador se utiliza el mismo tipo de funciones:

$$C[V_{gs}, V_{ds}] = C_A[\tanh(V_{gs})] \cdot C_B[\tanh(V_{ds})] \quad (4.7)$$

Debido a la similitud que existe entre  $I_{ds}(V_{gs}, V_{ds})$  y  $C_{gs}(V_{gs}, V_{ds})$  las funciones se pueden expresar como:

$$C_{gs} = C_{gso} \cdot [1 + \tanh(\Psi_1)] \cdot [1 + \tanh(\Psi_2)] \quad (4.8)$$

$$C_{gd} = C_{gdo} \cdot [1 + \tanh(\Psi_3)] \cdot [1 + \tanh(\Psi_4)] \quad (4.9)$$

donde

$$\Psi_1 = P_{0gsg} + P_{1gsg} V_{gs} + P_{2gsg} V_{gs}^2 + P_{3gsg} V_{gs}^3 + \dots \quad (4.10)$$

$$\Psi_2 = P_{0gsd} + P_{1gsd} V_{ds} + P_{2gsd} V_{ds}^2 + P_{3gsd} V_{ds}^3 + \dots \quad (4.11)$$

$$\Psi_3 = P_{0gdg} + P_{1gdg} V_{gs} + P_{2gdg} V_{gs}^2 + P_{3gdg} V_{gs}^3 + \dots \quad (4.11)$$

$$\Psi_4 = P_{0gdd} + (P_{1gdd} + P_{1cc} V_{gs}) V_{ds} + P_{2gdd} V_{ds}^2 + P_{3gdd} V_{ds}^3 + \dots \quad (4.13)$$

El término  $P_{1cc} V_{gs} V_{ds}$  refleja el acoplamiento cruzado de  $V_{gs}$  y  $V_{gd}$  sobre  $C_{gd}$ .

Cuando se considera suficiente una precisión del orden del 5% de  $C_{gs}$  Y  $C_{gd}$ , las ecuaciones (4.8)-(4.13) se pueden simplificar a:

$$C_{gs} = C_{gso} \cdot [1 + \tanh(P_{1gsg} V_{gs})] \cdot [1 + \tanh(P_{1gsd} V_{ds})] \quad (4.14)$$

$$C_{gd} = C_{gdo} \cdot [1 + \tanh(P_{1gdg} V_{gs})] \cdot [1 - \tanh(P_{1gdd} V_{ds} + P_{1cc} V_{gs} V_{ds})] \quad (4.15)$$

La ecuación (4.15) se puede simplificar aún más si se desprecia el acoplamiento cruzado para tensiones de drenador grandes ( $V_{ds} > 1V$ ):

$$C_{gd} = C_{gdo} \cdot [1 + \tanh(P_{1gdg} V_{gs})] \cdot [1 - \tanh(P_{1gdd} V_{ds})] \quad (4.16)$$

## 4.2 Modelo del FhGIAF

El modelo del FhGIAF fue diseñado en principio para modelar los transistores HFET fabricados en sus laboratorios. Se trata de HFETs de enriquecimiento y depleción con longitudes de puerta de 0.3 y 0.5  $\mu m$ . Este modelo se deriva del modelo de Angelov que hemos descrito en el apartado anterior y está implementado en IAFSPICE. La corriente de drenador tiene la misma forma que la ecuación (4.1) en este caso las ecuaciones que describen las componentes  $I_{dA}$  e  $I_{dB}$  son diferentes. El término  $I_{dB}$  ha sido modificado para dar cuenta de la variación de la modulación del canal. De esta forma tenemos que  $I_{dB}$  ya no depende solamente de  $V_{ds}$  sino que también depende de  $V_{gs}$ .

$$I_{dB}[V_{ds}] = \left(1 + \frac{\lambda \cdot V_{ds}}{1 + \Delta\lambda \cdot (V_{gs} - V_{to})^2}\right) \cdot \tanh(\alpha \cdot V_{ds}) \quad (4.17)$$

$\Delta\lambda$  es la variación del parámetro de modulación del canal. La tensión umbral,  $V_{to}$  se expresa a través de una relación empírica entre dos parámetros del modelo, pero conserva el mismo significado que en otros tipos de transistores definiendo las distintas regiones de trabajo del dispositivo.

$$V_{to} = V_c - \frac{2}{\beta} \quad (4.18)$$

Los parámetros de modelo  $V_c$  y  $\beta$  se describirán más adelante. De la ecuación (4.17) se desprende que al aumentar la tensión de puerta,  $V_{gs}$ , la variación de la corriente de drenador con  $V_{ds}$  se ve afectada menos por la modulación del canal.

Sin embargo, el cambio más significativo propuesto por el FhGIAF respecto al modelo de Angelov lo constituye el realizado con el término  $I_{dA}$ . La ecuación propuesta consiste en sumarle al término  $I_{dA}$  dado por Angelov otro término similar. Así tenemos:

$$I_{dA}[V_{gs}] = I_{dA1}[V_{gs}] + I_{dA2}[V_{gs}] = I_{dVC}(1 + \tanh(\Psi_1)) + I_{dVSB}(1 + \tanh(\Psi_2)) \quad (4.19)$$

$I_{dVC}$  y  $\Psi_1$  actúan de forma similar a  $I_{pk}$  y  $\Psi$  del modelo de Angelov, es decir, modelan el primer pico de la transconductancia (ver figura 4.1). En este caso se toman solamente los términos de orden 1 y 3 de la función potencial  $\Psi_1$ . Por tanto  $I_{dA1}$  queda:

$$I_{dA1}[V_{gs}] = I_{dVC}(1 + \tanh(\beta \cdot (V_{gs} - V_c) + \gamma \cdot (V_{gs} - V_c)^3)) \quad (4.20)$$

donde  $\beta$  y  $\gamma$  son los parámetros  $P_1$  y  $P_3$  del modelo de Angelov,  $I_{dVC}$  equivale a la  $I_{pk}$  y  $V_c$  es la  $V_{pk}$ . El término  $I_{dA2}$  corresponde al punto de transconductancia mínima, es decir, modela el segundo pico de la transconductancia (figura 4.1). Así,  $I_{dVSB}$  es la corriente de drenador para la cual tenemos una transconductancia mínima, y  $\Psi$  es en general una función potencial en serie centrada en  $V_{SB}$  ( $V_{gs}$  para transconductancia mínima) y con  $V_{gs}$  como variable. De esta forma y tomando sólo el primer término de la serie tenemos:

$$I_{dA2}[V_{gs}] = I_{dVSB}(1 + \tanh(\delta \cdot (V_{gs} + V_{SB}))) \quad (4.21)$$

$\delta$  es el parámetro  $P_1$  de la sucesión. En la mayoría de los casos y sobre todo en los HFETs de enriquecimiento, el pico negativo de la transconductancia está muy alejado del positivo, o es muy poco pronunciado. En estos casos se puede despreciar su efecto sobre la  $I_d$ , con lo que los parámetros  $I_{dVSB}$  y  $\delta$  toman valores nulos.

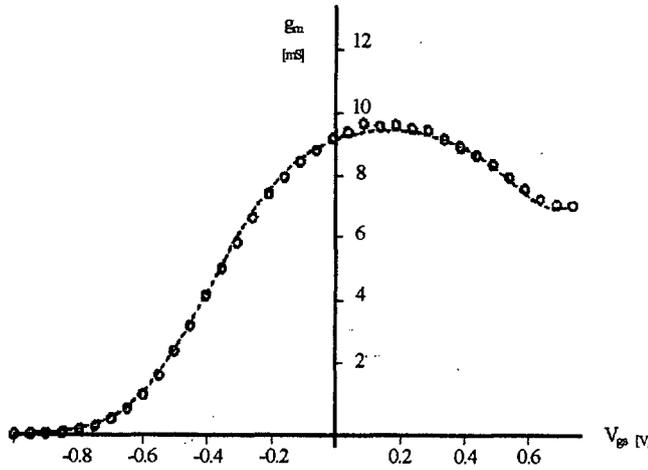


fig. 4.1 Transconductancia de un HFET de depleción de  $0.5 \mu\text{m} \times 25 \mu\text{m}$ , con  $V_{ds}=1.5 \text{ V}$ ,  $V_c= 0.13 \text{ V}$ , y  $\beta=2.15$ . Línea continua: medida; línea de puntos: IA-FSPICE.

La ecuación que modela la corriente de drenador queda finalmente como:

$$I_{ds} = f_1(V_{gs}) \cdot f_2(V_{ds}, V_{gs}) \cdot f_3(V_{ds}) \tag{4.22}$$

donde:

$$f_1[V_{gs}] = I_{dvc}(1 + \tanh(\beta \cdot (V_{gs} - V_c) + \gamma \cdot (V_{gs} - V_c)^3)) + I_{dVSB}(1 + \tanh(\delta \cdot (V_{gs} - V_c))) \tag{4.23}$$

$$f_2[V_{ds}, V_{gs}] = \left(1 + \frac{\lambda \cdot V_{ds}}{1 + \Delta \lambda \cdot (V_{gs} - V_{to})^2}\right) \tag{4.24}$$

$$f_3[V_{ds}] = \tanh(\alpha \cdot V_{ds}) \tag{4.25}$$

El modelado de las dependencias de las capacidades  $C_{gs}$ , y  $C_{gd}$  con la tensión de puerta y de drenador se hace utilizando el mismo esquema propuesto por Angelov. De esta manera, la capacidad  $C_{gs}$  se define para las diferentes regiones de trabajo del transistor como:

$$C_{gs} = C_{gso} + f_{c1}(V_{gs}) \cdot f_{c2}(V_{ds}) \quad (4.26)$$

donde  $C_{gso}$  es la capacidad para alimentación nula y  $f_{c1}$  y  $f_{c2}$  son las funciones descritas a continuación para tres regiones de trabajo: por debajo del *pinch-off* ( $V_{gs} < V_{at}$ ), el canal en conducción ( $V_{gs} \geq V_{bt}$ ) y una región intermedia entre ambas ( $V_{at} \leq V_{gs} < V_{bt}$ ).

$$\text{Para } V_{gs} < V_{at} \quad \begin{cases} f_{c1} = 0 \\ f_{c2} = 0 \end{cases} \quad (4.27)$$

$$\text{Para } V_{at} \leq V_{gs} < V_{bt} \quad \begin{cases} f_{c1} = C_{gs1} \cdot \frac{V_{gs} - V_{at}}{V_{bt} - V_{at}} \\ f_{c2} = 1 + V_{st} \cdot \tanh(\alpha \cdot V_s \cdot V_{ds}) \end{cases} \quad (4.28)$$

$$\text{Para } V_{gs} \geq V_{bt} \quad \begin{cases} f_{c1} = C_{gs1} \cdot (1 + C_f (V_{gs} - V_{bt})^m) \\ f_{c2} = 1 + V_{st} \cdot \tanh(\alpha \cdot V_s \cdot V_{ds}) \end{cases} \quad (4.29)$$

Para el caso de la  $C_{gd}$ , sólo hay que cambiar  $V_{gs}$ , por  $V_{gd}$  e invertir el signo de  $V_{ds}$ .

#### 4.2.1 Adaptación del Modelo FhGIAF a APLAC

Una vez definidos los elementos básicos del modelo, el siguiente paso es su adaptación para poder incluirlo en APLAC. Esta adaptación consiste en definir los circuitos equivalentes del transistor para su modelado en dc, ac, y régimen transitorio. Además se deberán definir los elementos parásitos que se han de incluir en dichos circuitos equivalentes para que el transistor quede modelado de forma completa.

### 4.2.1.1 Modelo en Gran Señal

El modelado en gran señal comprende dos aspectos, el comportamiento estático o dc, y el comportamiento dinámico o transitorio.

### 4.2.1.2 Modelo Equivalente en dc

El modelo de para los HFETs es el que se muestra en la figura 4.2. El valor de la corriente de drenador, representado como una fuente de corriente, es el dado por la ecuación (4.22). En las figuras 4.3 y 4.4 se muestran la transconductancia y las curvas  $I_{ds}$ - $V_{gs}$  e  $I_{ds}$ - $V_{ds}$  para un transistor de depleción de  $0.5 \mu\text{m}$ . Como se puede observar, para tensiones de puerta elevadas, aparece un mínimo en la transconductancia lo cual justifica la inclusión de  $I_{dA2}$  en la corriente de drenador. El modelo FhGIAF presenta una mayor exactitud para los transistores de depleción que para los de enriquecimiento.

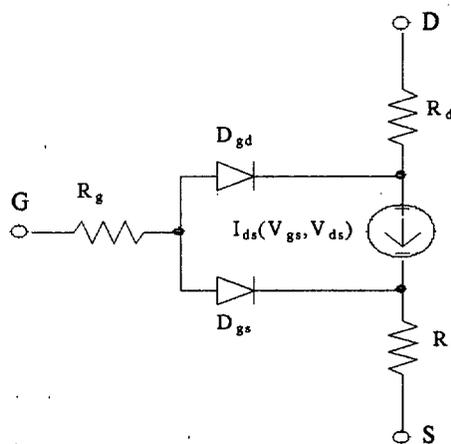


fig. 4.2 Circuito equivalente dc.

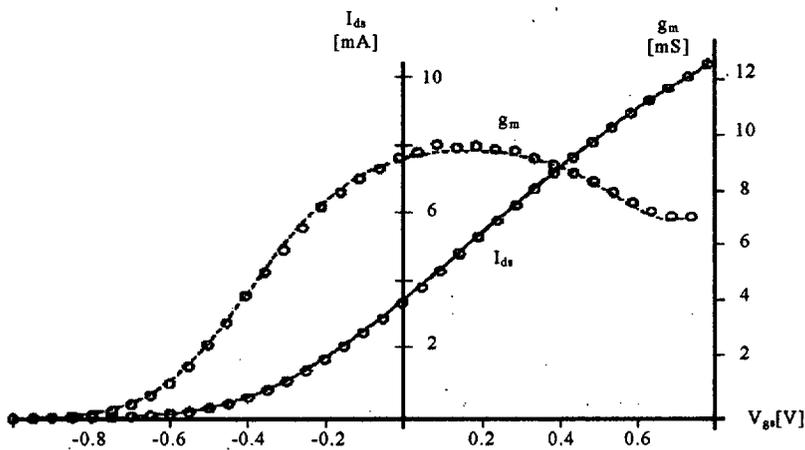


fig. 4.3 Transconductancia y Corriente de un HFET de depleción de  $0.5 \mu\text{m} \times 25 \mu\text{m}$ , con  $V_{ds} = 1.5 \text{ V}$ ,  $V_c = 0.13 \text{ V}$ , y  $\beta = 2.15$ . Línea continua: medida; línea de puntos: IAFSPICE.

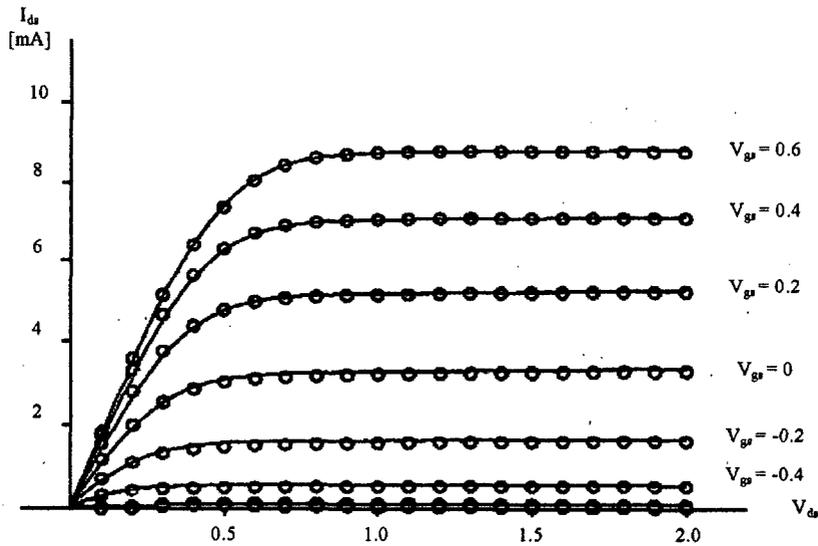


fig. 4.4 Curva I-V de un HFET de depleción de  $0.5 \mu\text{m} \times 25 \mu\text{m}$ , con  $V_c = 0.13 \text{ V}$ , y  $\beta = 2.15$ . Línea continua: medida; línea de puntos: IAFSPICE.

La corriente de puerta se modela a través de dos diodos conectados entre la puerta y la fuente y entre la puerta y el drenador. Estos diodos se modelan para tensiones de puerta positivas a través de la ecuación típica de los diodos, es decir:

$$I_g = I_{so} \cdot (e^{\frac{qV_g}{n \cdot k \cdot T}} - 1), \text{ para } V_g \geq 0 \quad (4.30)$$

donde  $I_{so}$  es la corriente de saturación del diodo que es igual tanto para  $I_{gs}$  como para  $I_{gd}$  al ser el transistor simétrico respecto a la puerta. Sin embargo, para tensiones de puerta negativas, especialmente con longitudes de puerta por debajo de la micra, aparecen corrientes de fuga debidas tanto a corrientes residuales desde el substrato, como al efecto túnel cerca de la superficie. En este caso, la corriente de puerta se modela como:

$$I_g = -I_{so} \cdot A \cdot (e^{\frac{q\sqrt{|B \cdot V_g|}}{n \cdot k \cdot T}} - 1), \text{ para } V_g < 0 \quad (4.31)$$

Los parámetros empíricos A y B dan cuenta de los fenómenos que hemos mencionado. Como se aprecia en la figura 4.5, el uso de este modelo en las dos regiones para los diodos permite obtener una buena aproximación para las corrientes de puerta para  $V_{ds} = 0$ . Para  $V_{ds} > 0$ , las variaciones en las corrientes de puerta pueden ser despreciadas.

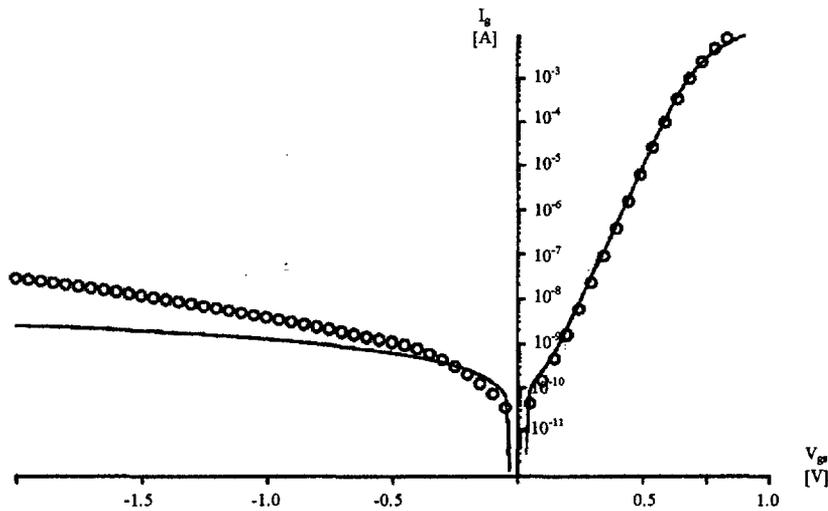


fig. 4.5 Corriente de Puerta frente a  $V_{gs}$  de un HFET de depleción de  $0.5 \mu\text{m} \times 25 \mu\text{m}$ , con  $V_{ds}=0$ ,  $V_c=0.13 \text{ V}$ , y  $\beta=2.15$ . Línea continua: medida; línea de puntos: IAFSPICE.

Por último, en el circuito equivalente en dc encontramos las resistencias parásitas que conectan el dispositivo intrínseco con los terminales externos. Nótese la inclusión de la resistencia de puerta,  $R_g$ , la cual no aparece en los MESFETs ni en los MOSFETs debido al material de puerta altamente conductor que poseen. En el caso de los HFETs, no se puede despreciar su resistencia aunque sea mucho menor que las resistencias de drenador y de surtidor.

#### 4.2.1.3 Modelo en Régimen Transitorio

El comportamiento dinámico está regido por los cambios que se producen en la carga almacenada en las distintas regiones del dispositivo al variar la tensión. El caso cuasi estático puede modelarse considerando por separado el comportamiento intrínseco y el extrínseco. El circuito equivalente del HFET para régimen dinámico es el que se muestra en la figura 4.6.

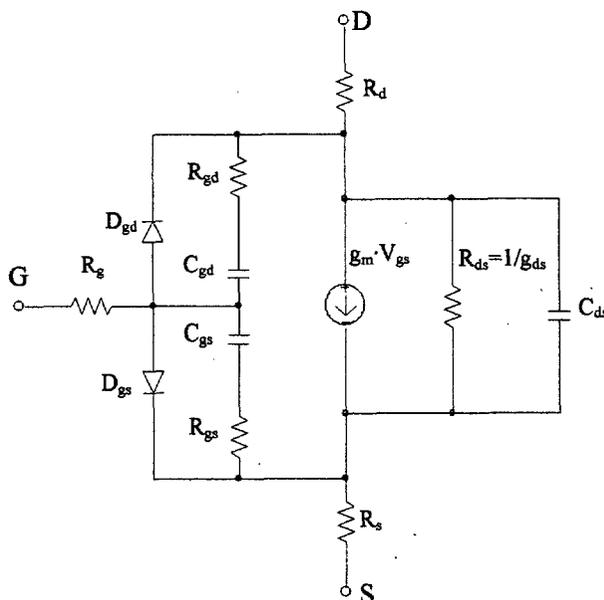


fig. 4.6 Circuito equivalente para régimen dinámico.

El comportamiento extrínseco se modela mediante condensadores no lineales. En este caso, esos condensadores son los descritos en el apartado 4.2 como  $C_{gs}$  y  $C_{gd}$ . En serie con estas capacidades nos encontramos con las resistencias  $R_{gs}$  (también llamada  $R_i$ ) y  $R_{gd}$  las cuales, aunque aparecen en el modelo, no están implementadas en IAFSPICE. Las conductancias de puerta a drenador,  $g_{gd}$ , y de puerta a surtidor,  $g_{gs}$  se definen como las derivadas de las corrientes de puerta respectivas. Así, para el caso de la conductancia de puerta a surtidor tenemos:

$$g_{gs} = \frac{\partial I_{gs}}{\partial V_{gs}} = I_{so} \cdot \frac{q \cdot e^{\frac{q \cdot V_{gs}}{n \cdot k \cdot T}}}{n \cdot k \cdot T}, \text{ para } V_{gs} \geq 0 \quad (4.32)$$

$$g_{gs} = \frac{\partial I_{gs}}{\partial V_{gs}} = I_{so} \cdot A \cdot \frac{q \cdot B \cdot e^{\frac{q \cdot \sqrt{|B \cdot V_{gs}|}}{n \cdot k \cdot T}}}{2 \cdot n \cdot k \cdot T \cdot \sqrt{|B \cdot V_{gs}|}}, \text{ para } V_{gs} < 0 \quad (4.33)$$

Para el caso de la conductancia de puerta a drenador las ecuaciones son las mismas cambiando  $V_{gs}$  por  $V_{gd}$ .

En el circuito equivalente para régimen dinámico aparece también la capacidad de drenador a surtidor,  $C_{ds}$  conectada en paralelo a la resistencia  $R_{ds}$ . El valor de  $C_{ds}$  es normalmente pequeño comparado con los valores de  $C_{gs}$  Y  $C_{gd}$ . Además, su valor se considera que no varía con la tensión. La resistencia de drenador a surtidor,  $R_{ds}$  se define como la inversa de la conductancia  $g_{ds}$ . Tanto la transconductancia,  $g_m$  como la conductancia de drenador a surtidor,  $g_{ds}$ , vienen definidas como las derivadas parciales de la corriente de drenador en un punto de polarización Q respecto a las tensiones  $V_{gs}$  Y  $V_{ds}$  respectivamente, es decir:

$$I_{ds}[V_{gs}, V_{ds}] = I_Q + \left. \frac{\partial I_q}{\partial V_{gs}} \right|_Q + \left. \frac{\partial I_q}{\partial V_{ds}} \right|_Q = I_Q + g_m + g_{ds} \quad (4.34)$$

Aplicando estas definiciones obtenemos las siguientes expresiones para  $g_m$  y para  $g_{ds}$ :

$$g_m = f_3 \cdot \left\{ f_2 \cdot \left[ I_{drc} \left( \frac{\beta + \gamma \cdot 3 \cdot (V_{gs} - V_C)^2}{\cosh^2(\beta \cdot (V_{gs} - V_C) + \gamma \cdot (V_{gs} - V_C)^3)} \right) + I_{d/sb} \left( \frac{\delta}{\cosh^2(\delta \cdot (V_{gs} - V_{sb}))} \right) \right] - f_1 \cdot \left[ \frac{V_{ds} \cdot \lambda \cdot \Delta\lambda \cdot 2 \cdot (V_{gs} - V_{to})}{(1 + \Delta\lambda \cdot (V_{gs} - V_{to})^2)^2} \right] \right\} \quad (4.35)$$

$$g_{ds} = f_1 \cdot \left[ \left( \frac{\lambda}{1 + \Delta\lambda \cdot (V_{gs} - V_{to})^2} \right) \cdot \tanh(\alpha \cdot V_{ds}) + \left( 1 + \frac{\lambda \cdot V_{ds}}{1 + \Delta\lambda \cdot (V_{gs} - V_{to})^2} \right) \cdot \frac{\alpha}{\cosh^2(\alpha \cdot V_{ds})} \right] \quad (4.36)$$

Para  $V_{ds} < 0$ , debemos cambiar el signo de dicha tensión en la ec.(4.22) y volver a calcular las expresiones de  $g_m$  y  $g_{ds}$  tal y como lo acabamos de hacer.

El comportamiento intrínseco se modela teniendo en cuenta la carga almacenada en la estructura del HFET, es decir, la carga almacenada en las capacidades  $C_{gs}$  y  $C_{gd}$ . Por tanto, la carga almacenada en las capacidades  $C_{gs}$  y  $C_{gd}$  quedarán definidas como su integral respecto a  $V_{gs}$  y  $V_{gd}$  respectivamente. Así, para tres regiones para las que están definidas las capacidades de puerta tenemos las siguientes expresiones:

$$\text{Para } V_{gs} < V_{at} \quad Q_{gs} = C_{gs0} \cdot V_{gs} + Q_{c1} \quad (4.38)$$

$$\text{Para } V_{at} \leq V_{gs} < V_{bt} \quad Q_{gs} = C_{gs0} \cdot V_{gs0} + C_{gs1} \cdot \frac{V_{gs} - V_{at}}{2 \cdot (V_{bt} - V_{at})} \cdot f_{c2} + Q_{c2} \quad (4.39)$$

$$\text{Para } V_{gs} \geq V_{bt} \quad Q_{gs} = Q_{gs0} \cdot V_{gs} + C_{gs1} \left[ V_{gs} + C_f \cdot \frac{(V_{gs} - V_{bt})^{m+1}}{m+1} \right] \cdot f_{c2} + Q_{c3} \quad (4.40)$$

Para  $Q_{gd}$  las expresiones son las mismas cambiando  $V_{gs}$  por  $V_{gd}$  e invirtiendo el signo de  $V_{ds}$ . Los valores de  $Q_{c1}$ ,  $Q_{c2}$  y  $Q_{c3}$  deben ser definidos para respetar las condiciones de continuidad de la carga y de sus derivadas en las interfases entre las tres regiones. De no ser así, podríamos tener problemas de convergencia a la hora de simular con APLAC. Por tanto, para  $V_{gs} = V_{at}$  se deberá cumplir:

$$Q_1 = Q_2, C_1 = C_2, \frac{\partial C_1}{\partial V_{gs}} = \frac{\partial C_2}{\partial V_{gs}} \quad (4.41)$$

donde los subíndices 1 y 2 indican las regiones anterior y posterior de la interfase. Las condiciones a cumplir en la otra interfase,  $V_{gs} = V_{bt}$ , son:

$$Q_2 = Q_3, C_2 = C_3, \frac{\partial C_2}{\partial V_{gs}} = \frac{\partial C_3}{\partial V_{gs}} \quad (4.42)$$

donde los subíndices 2 y 3 indican aquí igualmente las regiones anterior y posterior de la interfase. Aplicando estas condiciones obtenemos los valores de  $Q_{c1}$ ,  $Q_{c2}$  y  $Q_{c3}$ :

$$Q_{c2} = Q_{c3} \quad (4.43)$$

$$Q_{c3} = Q_{c2} - C_{gs1} \cdot f_{c2} \cdot \frac{(V_{bt} + V_{at})}{2} \quad (4.44)$$

Como vemos, las cargas  $Q_{c2}$  y  $Q_{c3}$ , dependen de  $Q_{c1}$ . Desde un punto de vista físico, esta carga da cuenta de la carga residual que tiene el dispositivo que se supone nula.

### 4.2.1.4 Modelo en Pequeña Señal

El modelo en pequeña señal se divide en una parte intrínseca y en otra extrínseca. En la figura 4.7.a se muestra el circuito equivalente en pequeña señal completo. El modelo intrínseco corresponde al encerrado por la línea discontinua. El elemento principal lo constituye la fuente de corriente controlada por tensión de valor  $I_{ds} = \hat{g}_m \cdot V_{gs}$ . La expresión para la transconductancia modificada es la siguiente:

$$\hat{g}_m = g_m \cdot e^{-j\Omega\tau} = g_m \cdot (\cos\Omega\tau - j \text{sen}\Omega\tau) , \text{ con } \Omega = 2 \cdot \pi \cdot \text{frec} \quad (4.44)$$

donde  $g_m$  es la transconductancia y  $\tau$  es un exceso de fase similar al parámetro ptf de los transistores bipolares.

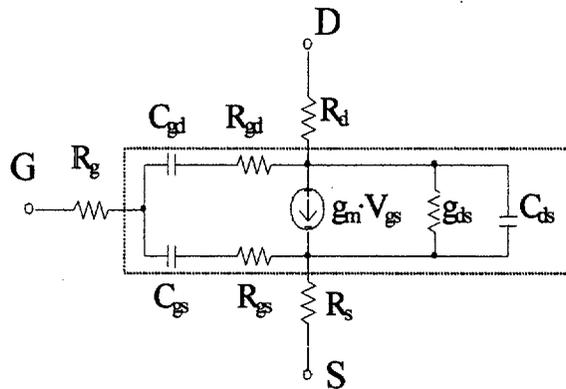


fig. 4.7a Circuito equivalente en pequeña señal simplificado

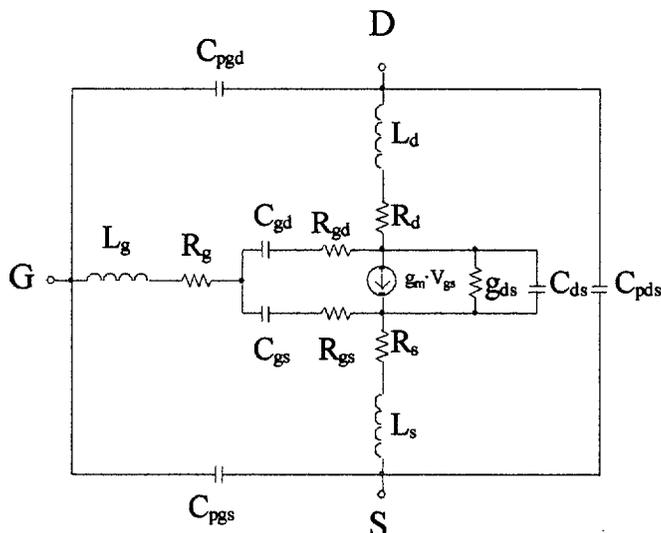


fig. 4.7b Circuito equivalente en pequeña señal completo

La parte extrínseca del modelo corresponde a las resistencias de los terminales. Para poder hacer pruebas con transistores reales y poder comparar los resultados con los del modelo, debemos añadir el efecto de los PADs. Esto se hace incluyendo tres bobinas y tres condensadores tal y como se muestra en la figura 4.7.b. Estos elementos no forman parte del modelo del transistor y por tanto se deben incluir como elementos externos dentro del fichero de entrada de APLAC.

#### **4.2.1.5 Lista de Parámetros**

Como conclusión, en este capítulo se han presentado los elementos adecuados para simular las características más relevantes de los HFETs. En la tabla 4-1 se recoge una lista de los parámetros del modelo implementado. En ella aparecen los nombres de los parámetros dados tanto en el modelo, como en APLAC e internamente en el código fuente, junto con las unidades respectivas.

Tabla 4-1 Parámetros APLAC de los transistores HFETs

Nombre en el Modelo	Nombre en APLAC	Significado	Unid.
$I_{GSS}$	IS	Corriente de saturación de los diodos de puerta	A/ $\mu\text{m}$
N	NP	Coefficiente de emisión o de no idealidad	-
A	AFACT	Parámetro de ajuste A	-
B	GEXP	Parámetro de ajuste B	V <sup>-1</sup>
$CD_{VC}$	CDVC	Corriente de drenador para transconductancia máxima	A/ $\mu\text{m}$
$\beta$	BETA	Parámetro de ajuste $\beta$	V <sup>-1</sup>
$V_C$	VC	Tensión de puerta para transconductancia máxima	V
	GAMMA	Parámetro de ajuste $\gamma$	V <sup>-3</sup>
$CD_{VSB}$	CDVSB	Corriente de drenador para transconductancia mínima	A/ $\mu\text{m}$
$\delta$	DELTA	Parámetro de ajuste $\delta$	V <sup>-1</sup>
$V_{SB}$	VSB	Tensión de puerta para transconductancia máxima	V
$\lambda$	LAMBDA	Parámetro de modulación del canal	V <sup>-1</sup>
$\Delta\lambda$	DXL	Variación del parámetro de modulación del canal	V <sup>-2</sup>
$\alpha$	ALPHA	Parámetro de tensión de saturación	V <sup>-1</sup>
$V_{at}$	VAT	Tensión de pinch-off	V
$V_{bt}$	VBT	Tensión de conducción	V
$C_{gs0}$	CGS0	Capacidad para alimentación nula	F/ $\mu\text{m}$
$C_{gs1}$	CGS1	Factor de capacidad para alimentación nula	F/ $\mu\text{m}$
$V_{st}$	VST	Parámetro de ajuste $V_{st}$	-
$V_s$	VS	Parámetro de ajuste $V_s$	-
$C_f$	FC	Parámetro de ajuste $C_f$	V <sup>-m</sup>
m	MFACT	Parámetro de ajuste m	-
$R_d$	RD	Resistencia de drenador	$\Omega \mu\text{m}$
$R_s$	RS	Resistencia de surtidor	$\Omega \mu\text{m}$
$R_g$	RG	Resistencia de puerta	$\Omega/\mu\text{m}$
$R_{gs}$	RGS	Resistencia puerta-surtidor	$\Omega \mu\text{m}$
$R_{gd}$	RGD	Resistencia puerta-drenador	$\Omega \mu\text{m}$
$Q_{c1}$	QC1	Carga residual	-
$\tau$	TD	Exceso de fase	rad
$C_{ds}$	CDS	Capacidad drenador-surtidor	F/ $\mu\text{m}$

# Implementación del Modelo del HFET en APLAC

En este capítulo explicaremos como se ha realizado la implementación del modelo de HFET utilizando los circuitos equivalentes en ac, dc y régimen transitorio comentados en el capítulo 4.

La implementación de un nuevo componente en APLAC puede realizarse de dos formas diferentes: utilizando el editor de esquemas de APLAC o en modo de programación APLAC. En nuestro caso utilizaremos ambos métodos comentando las limitaciones y ventajas de ambos.

Haremos hincapié en los errores más frecuentes que podemos encontrarnos y como detectarlos. Así mismo, se estudiará las limitaciones de APLAC.

La principal herramienta que utiliza APLAC en sus modelos y también la que se utiliza para realizar nuevos modelos son las fuentes controladas.

- VCCS (Voltage-Controlled Current Source)
- CCCS (Current-Controlled current Source)
- CCVS (Current-Controlled Voltage Source)
- VCVS (Voltage-Controlled Voltage Source)

En nuestro caso utilizaremos la VCCS de la cual haremos un análisis detallado de sus funciones y opciones en el siguiente apartado 5.1.1.

## 5.1 Elementos Básicos para Modelar Dispositivos en APLAC

En este apartado veremos los principales bloques y estamentos para poder realizar el modelado de cualquier dispositivo en APLAC, viendo sus principales características y un detallado análisis de sus parámetros obligatorios y parámetros opcionales así como su correcta utilización. Haremos especial hincapié en la fuente de corriente controlada por tensión (VCCS) que es la que utilizaremos en el modelado del HFET. También se describirá el estamento Function útil para definir funciones de modelado dentro de APLAC y el Bloque DefModel ... EndModel el cual sirve para definir submodelos que pueden ser nuevos elementos o circuitos.

### 5.1.1 Descripción de VCCS (Voltage-Controlled Current Source)

Como principio fundamental de APLAC podemos decir que todos los componentes están representados mediante fuentes de corriente. Estas fuentes pueden ser, o bien independientes (Curr) o bien dependientes de tensión (VCCS). Todos los circuitos y modelos son finalmente mapeados a fuentes de corriente de estos dos tipos, las cuales son los únicos elementos directamente asociadas al analizador APLAC.

En este apartado definiremos las características principales de la VCCS (Voltage-Controlled Current Source), así como una descripción del modelo y sus valores.

#### 5.1.1.1 Declaración y Parámetros de la Fuente de Corriente Controlada por Tensión

La definición general de una fuente de corriente dependiente de tensión es la siguiente:

##### DECLARACIÓN

VCCS nombre n+ n- m n1+ n1- ... nm+ nm- valor  
+ [ Parámetros opcionales ]

**+ [ Parámetros electro térmicos ]**

Forma general: VCCS "nombre" n+ n- m n1+ n1-... nm+ nm-valor C DC=dc DERIV ...

Ejemplo: VCCS "VC1" 1 2 1 3 4 2.5

A continuación pasamos a describir los parámetros obligatorios y opcionales necesarios para describir de forma completa una fuente de corriente dependiente de tensión. Dicha descripción se hará en forma de tabla en la que se especifica el nombre del parámetro, sus argumentos y una breve descripción del mismo.

**PARÁMETROS OBLIGATORIOS**

Descripción de los parámetros mínimos que debemos introducir en la declaración de la fuente: -

Par	Arg	Descripción
	nombre	Nombre del elemento.
	n+	Nodo positivo de salida
	n-	Nodo negativo de salida.
	m	Número m de tensiones de control.
	ni+	Nodo positivo de control i (i variará dependiendo de m)
	ni-	Nodo negativo de control
	valor	Variable que define la corriente o la carga del elemento (y sus derivadas con respecto a las tensiones de control, si se le da la derivada). El primer valor define el valor de la corriente o la carga y la siguiente secuencia de valores es igual a sus derivadas (conductancia o capacidad) con respecto a la tensión de control.

**PARÁMETROS OPCIONALES**

Descripción de los parámetros que nos permiten especificar de modo más detallado el funcionamiento de la fuente:

Par	Arg	Descripción
C / G / Y		
C		VCCS dinámica o capacitiva. La corriente i es igual a la derivada respecto al tiempo de la función estática q (carga almacenada por el elemento). La carga es una función explícita de la tensión de control, $q = q(u),$ y la corriente del elemento es dada por $i = \frac{\partial q}{\partial t}.$
G		VCCS estática o conductiva. La corriente i es estática (instantánea), viene definida por la función de las tensiones de control, $i = i(u)$ . Esta opción viene definida por defecto.

Continúa en la siguiente página

Par	Arg	Descripción
Y		Parámetro lineal (dependiente de la frecuencia), que permite transadmitancias complejas en ac y en análisis armónicos. En dc y análisis transitorio la parte imaginaria es omitida.
DC		Valores iniciales para las tensiones de control en dc. DC debe definir m valores, uno por cada tensión de control.
DERIV/ NO_DERIV/ DEBUG_DERIV		
DERIV		Las derivadas son definidas por el argumento valor.
NO_DERIV		Las derivadas son calculadas por el simulador. Esta opción viene definida por defecto.
DEBUG_DERIV	r	Diferencia relativa máxima permitida entre la derivada analítica y la numérica. Esta opción se utiliza para chequear la validez de las formulas de la derivada dada. Si la diferencia relativa entre la derivada analítica y la numérica excede este valor, una tabla de aviso nos da el resultado de los valores de la derivada analítica y la numérica.
EULER / TRAPEZ / GEAR		
EULER		Se utiliza la regla de Euler en la integración numérica. Utilizado sólo con componente tipo C.
TRAPEZ		Se utiliza la regla trapezoidal en la integración numérica. Utilizado sólo con componente tipo C.
GEAR		Se utilizà la formula Gear-Shichman en la integración numérica. Utilizado sólo para componente tipo C.
LINEAR / NONLINEAR		
LINEAR		Elemento lineal. Opción por defecto.
NONLINEAR		Elemento no lineal.
POLY	i	Dependencia no lineal polinómica de orden p. El valor de APLACVAR debería definir los coeficientes polinomicos ( $p_0 + p_1 + \dots + p_{m(p+1)-1}$ ). Para una tensión de control, m=1, los coeficientes son definidos como (p=3): $i = p_0 + p_1 u_1 + p_2 u_1^2 + p_3 u_1^3$ En el caso de que m =2 y p =2 : $i = (p_0 + p_1 u_1 + p_2 u_1^2)(p_3 + p_4 + p_4 u_2 + p_5 u_2^2)$ Nótese que el orden p es el mismo para todas las tensiones de control.
ACTIVE / PASSIVE		
ACTIVE		Elemento activo, se utiliza para el análisis de estabilidad.
PASSIVE		Elemento pasivo. Opción por defecto.
U0	vx	Vector de valores iniciales de corriente para un elemento capacitivo

### 5.1.1.2 Descripción del Modelo

Una vez descrita la definición general de una VCCS y sus parámetros, pasamos a estudiar como es su modelo dentro de APLAC.

Una VCCS puede ser estática o dinámica. La corriente de un elemento estático (conductor) es una función estática de las tensiones de control,

$$i = i(u). \tag{5.1}$$

Por estático se entiende que la función  $i(u)$  depende sólo de valores instantáneos de las tensiones. También se supone que la corriente  $i$  puede expresarse como una función de las tensiones de control. Las fuentes dependientes en APLAC son estáticas por defecto.

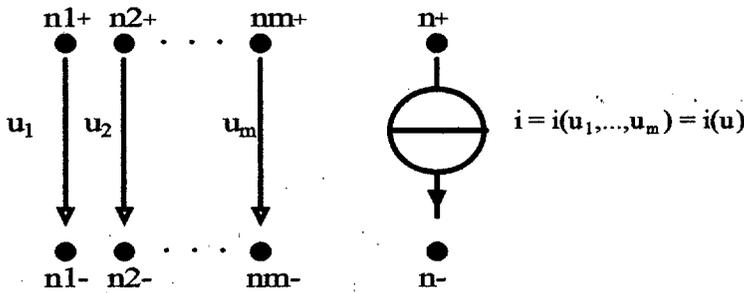


fig. 5.1 VCCS estática.

La corriente de un elemento dinámico (capacitivo) depende de la derivada de una función estática,  $q$  (carga almacenada en el elemento). La carga es una función de las tensiones de control,

$$q = q(u), \tag{5.2}$$

y la corriente viene dada por

$$i = \frac{\partial q}{\partial t}. \tag{5.3}$$

VCCS se vuelve capacitivo, si el identificador C es especificado. Un elemento dinámico sólo puede definirse en el dominio de la frecuencia dando el identificador Y.

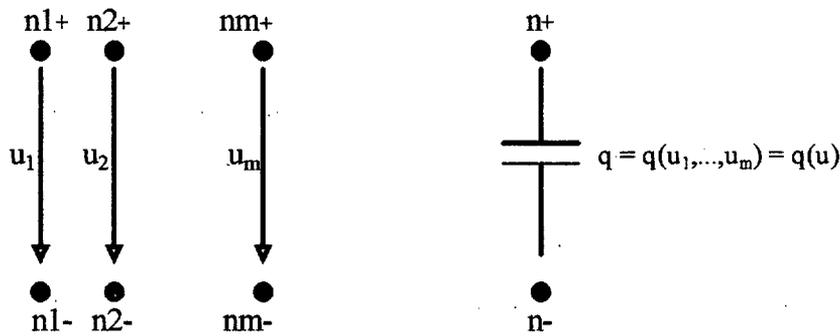


fig. 5.2 VCCS dinámica.

Un VCCS siempre define una fuente de corriente: la corriente  $i$  fluye del  $n+$  al  $n-$  y el argumento  $m$  define el número de tensiones de control. Debe haber una tensión de control como mínimo, y no hay ningún límite fijo al número de tensiones de control.

### 5.1.1.3 Conductancia y Capacitancia Lineales

Las propiedades de un elemento lineal pueden darse en términos de una tensión de control. Generalizar las propiedades de muchas tensiones de control es fácil porque su contribución puede tratarse separadamente. La corriente de una conductancia lineal es proporcional a la tensión:

$$i = gu. \quad (5.4)$$

Donde  $g$  es o una conductancia o una transconductancia. En el dominio de la frecuencia,  $g$  puede ser compleja, y entonces se llama transadmitancia

La carga de una capacidad lineal es proporcional a la tensión

$$q = Cu, \quad (5.5)$$

lo cual significa que (suponiendo que  $C$  es invariante en el tiempo) la corriente es

$$i = C \frac{\partial u}{\partial t}. \quad (5.6)$$

#### 5.1.1.4 Valor de una Fuente Lineal

Por defecto, una VCCS es lineal. En APLAC, el primer valor no se utiliza en fuentes lineales con lo que se le asigna automáticamente un 0. Para un elemento tipo G el segundo valor define la transconductancia,  $g$ , con respecto a la primera tensión de control. El tercer valor definiría la conductancia con respecto a la segunda tensión de control y así sucesivamente. Se deben definir  $m+1$  valores.

Para los elementos tipo C el segundo valor define la capacidad,  $C$ , con respecto a la primera tensión de control. El tercer valor define la capacidad con respecto a la segunda tensión de control y así sucesivamente. Se definen  $m+1$  valores.

Un elemento tipo Y sólo puede ser lineal. El segundo valor define la transconductancia  $g$  (o la parte real de la transconductancia) con respecto a la primera tensión de control. El tercer valor define la conductancia con respecto a la segunda tensión de control y así sucesivamente. Después de las transconductancias, se declararán la parte imaginaria de las transadmitancias en orden. Los elementos tipo Y son sólo utilizados en análisis ac y en análisis de armónicos.

#### 5.1.1.5 Valor de una Fuente No Lineal

El valor de un elemento estático no lineal (G, NONLINEAR) debe definir la corriente,  $i$ , como su primer valor. Para un elemento capacitivo (C, NONLINEAR), el primer valor debe ser igual a la carga,  $q$ .

Por defecto o si se ha especificado el identificador NO\_DERIV, no se necesita especificar más valores. En este caso, las derivadas (conductancia o capacitancia) son calculadas numéricamente. Sin embargo para mejorar el tiempo de computación del simulador es recomendable que se especifiquen dichas derivadas de forma explícita. Se puede verificar la validez de las derivadas analíticas utilizando el identificador DEBUG\_DERIV e introduciéndole un nivel relativo de error entre ambas derivadas. Si las diferencias relativa entre la derivada analítica y numérica se excede de dicho valor, aparecerá una tabla de aviso con los resultados de ambas.

La función que determina el valor de la VCCS hace uso de las tensiones de control mediante el comando CV. La primera tensión de control sería CV(0), la segunda sería CV(1) y así sucesivamente. La función debería calcular y definir la corriente o carga y (en caso de que proceda) la transconductancia o capacitancia, en función de dichas tensiones.

Los elementos no lineales tipo Y no son permitidos.

### 5.1.2 Descripción del Estamento Function

Una función se define con el estamento function. Una función puede ser de dos tipos:

- Simple, nos devuelve uno o varios valores.
- De acceso, nos permite que las funciones sean dependientes de variables por ejemplo: Function RDep(x) [ 1, sqrt(x), 2\*x ], Rdep es una función de acceso de tres dimensiones que depende de la variable x, cada vez que realicemos una llamada a la función tenemos que especificar el valor de la variable x, por ejemplo: AplacVar TempDep RDep(Temp), el valor de la variable TempDep sería la función Rdep asignándole la temperatura a la variable x.

A continuación se muestra la descripción del estamento function.

ESTAMENTO	DESCRIPCIÓN
FUNCTION	Define una función. FUNCTION nombre expresión FUNCTION nombre (par1, par2, ...) expresión FUNCTION nombre {lvar1, lvar2, ... } expresión FUNCTION nombre (par1, par2, ...) {lvar1,lvar2, ...} expresión Donde par1, par2, ... son parámetros y lvar1, lvar2, ... variables locales aunque par1, par2 ... podrían utilizarse también como variables locales. Si la expresión consiste en múltiples subexpresiones separadas por comas, el valor de la función es el valor de la última subexpresión. Los parámetros pueden ser del tipo real, complejo, vector, vectores complejos, matrices o matrices complejas.

### 5.1.3 Descripción de los Estamentos IF-THEN-ELSE y IFTE

Ambos estamentos son iguales, la diferencia está en su utilización. Mientras el IFTE sólo puede ser utilizado dentro de funciones, el bloque IFTE-THEN-ELSE también puede utilizarse como rutina de interrupción del simulador o para condicionar el uso de un componente u otro según la condición que le demos.

A continuación se muestra la descripción del estamento function:

ESTAMENTO	DESCRIPCIÓN
IF	Estructura típica IF-THEN-ELSE-ENDIF: IF e THEN Definición/Estamentos ELSE Definición/Estamentos ENDIF Siendo e una expresión real
IFTE	Estructura típica IF-THEN-ELSE-ENDIF IFTE (e1,e2,e3) real es igual a e2 si e1 es verdadero sino e3, también puede utilizarse ifte (e1,e2,0) es igual que poner ifte (e1,e2).

### 5.1.4 Descripción del Bloque DefModel ... EndModel

DefModel comienza la descripción del bloque de un nuevo circuito el cual finaliza con EndModel. Nos permite crear nuevos componentes, modelos o macro modelos en APLAC.

Todos los componentes, variables APLAC, funciones, nodos, vectores, etc. son locales dentro del nuevo componente. Las funciones, vectores numéricos, matrices y modelos definidos fuera del bloque DefModel .. EndModel son visibles dentro del bloque. El nodo tierra (0 o GND) y los nodos definidos como globales también son visibles dentro del bloque. Todo lo demás es invisible dentro del modelo. Los Bloques DefModel ... EndModel no se pueden anidar.

Un modelo creado por el usuario puede consistir en cualquier componente APLAC y otros modelos creados por el usuario. Todos los estamentos de programación y funciones pueden definirse dentro del bloque DefModel ... EndModel.

Un modelo creado con DefModel acepta todos los parámetros electro térmicos. Si se especifica un parámetro electro térmico, todos los componentes dentro del DefModel tomarán este parámetro excepto VCCS, VCVS, CCCS y CCVS.

A continuación se muestra la descripción detallada del bloque DefModel ... EndModel.

## DECLARACIÓN

```
DEFMODEL      name      m      n1      n2      ...      nm
+ [ Parámetros opcionales ]
+ [ Parámetros electro térmicos ]
EndModel
```

Forma general: Defmodel "nombre" m n1 n2 ... nm FIXED=fixed1 fixed2 Param ...

EndModel

Ejemplo: DefModel "DM1" 1 1

EndModel

## PARÁMETROS OBLIGATORIOS

Descripción de los parámetros mínimos que debemos introducir en la declaración de un nuevo submodelo:

Par	Arg	Descripción
	Name	Nombre del elemento.
	m	Número de nodos externos.
	ni	Nodos externos del modelo.

## PARÁMETROS OPCIONALES

Descripción de los parámetros que nos permiten especificar de modo más detallado la declaración del submodelo:

Par	Arg	Descripción
FIXED	i	Números de parámetros fijos del modelo.
	x	Parámetros fijos del modelo que pueden ser variables APLAC.
FLAG	i	Número de flag del modelo.
	s	Flag del modelo.
PARAM	i	Número de parámetros opcionales del modelo.
	s	Parámetros opcionales del modelo los cuales pueden ser una variable APLAC, un nodo o un modelo.
GLOBAL_PARAM	i	Parámetros globales opcionales del modelo.
	s	Parámetros globales del modelo. Si los parámetros s no son especificados cuando se realiza la llamada al DefModel el valor por defecto de s sería el valor de una Variable APLAC llamado desde el nivel superior del circuito.
CIRCUIT		Sólo componentes reales (sistemas y bloques electromagnéticos no son permitidos dentro del DefModel).
EM		Dentro de este bloque de DefModel sólo se permiten componentes electromagnéticos. La combinación de sistemas y bloques electromagnéticos y componentes no se permiten.
SYSTEM		Sólo los bloques de sistemas son permitidos en este DefModel.
VECTOR	i v	Número de argumentos de vectores reales.
	SVi	Vectores reales.
MATRIX	im	Número de argumentos del tipo matriz real.
	SMi	Matrices reales.
CVECTOR	lcv	Número de argumentos del tipo vector complejo.
	SCVi	Vectores complejos.
CMATRIX	lcm	Número de argumentos del tipo matriz compleja.
	SCMi	Matrices complejas.

## 5.2 Modelado del HFET en CONTINUA

Una vez descrito los elementos básicos para el modelado de dispositivos en APLAC pasaremos a explicar el uso de dichos elementos para modelar el HFET.

Como vimos en el capítulo 3, apartado 3.2.1.2 el circuito equivalente en dc de un HFET es el que se muestra en la figura 5.3.

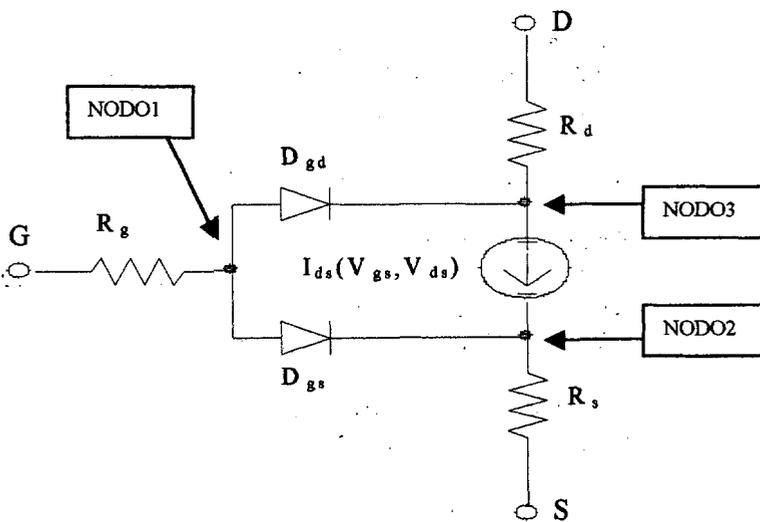


fig. 5.3 Circuito del HFET en dc

Como se puede observar, el modelo consta de una serie de componentes los cuales pueden tener asociados una serie de ecuaciones que describen su funcionamiento.

Primero realizaremos una descripción de cada componente en lenguaje APLAC para luego realizar el submodelo completo en el apartado (5.2.4).

### 5.2.1 Introducción de los Parámetros (HFET de enriquecimiento)

Los parámetros pueden ser introducidos de formas diferentes según la utilización que queremos para los parámetros:

- Podemos introducir los parámetros del submodelo con el estamento APLACVAR con lo que los parámetros no podrían ser modificados en la llamada al submodelo (con lo que crearíamos un submodelo fijo).

- O podemos introducirlos con los parámetros opcionales del DefModel FIXED y podrían ser modificados por el usuario en la llamada al submodelo.

A continuación se muestra la forma de introducir los parámetros para un HFEET de enrequecimiento en el submodelo, incluyendo los valores por defecto para los mismos.

Códigos	Parámetros
DEFAULT	BETA=2.8
DEFAULT	ALPHA=5
DEFAULT	VC=0.65
DEFAULT	VS=5
DEFAULT	LAMBDA=0.05
DEFAULT	DELTA=0
DEFAULT	DXL=10
DEFAULT	RG=0.2
DEFAULT	RD=600
DEFAULT	RS=600
DEFAULT	GAMMA=4
DEFAULT	CDVSB=0
DEFAULT	CDVC=0.23E-3
DEFAULT	Q=1.60022E-22
DEFAULT	K=1.381E-23
DEFAULT	AFACT=4
DEFAULT	GEXP=0.075
DEFAULT	N=1.5
DEFAULT	ISO=1E-14
DEFAULT	CGS0=2.5E-16
DEFAULT	CGS1=9E-16
DEFAULT	VST=1
DEFAULT	VAT=-0.3
DEFAULT	VBT=0.3
DEFAULT	FC=1.5
DEFAULT	VS=0.4
DEFAULT	MFACT=1
DEFAULT	QC1=0
DEFAULT	CDS=2E-16
DEFAULT	TD=3E-12

Algunos de los parámetros se ven afectados por la anchura del canal y por tanto debemos tenerla en cuenta en la definición de los mismos.

A continuación se muestra la forma de introducir este efecto en los parámetros del submodelo.

Códigos	Descripción
APLÁCVAR ISOW CALL ISOW=ISO*WW APLACVAR CDVSBW CALL CDVSBW=CDVSB*WW APLACVAR CDVCW CALLCDVCW=CDVC*WW APLACVAR CGS0W APLACVAR CGS1W CALL CGS1W=CGS1*WW APLACVAR RSW CALL RSW=RS/WW APLACVAR RGW CALL RGW=RG*WW APLACVAR RDW CALL RDW=RD/WW APLACVAR CDSW	Declaramos nuevas variables y luego le asignamos el efecto que produce el ancho del canal sobre ellas.

## 5.2.2 Implementación de los Diodos

Lo primero que debemos definir son las ecuaciones de los diodos mediante funciones APLAC:

$$I_g = I_{so} \cdot (e^{\frac{qV_g}{n \cdot k \cdot T}} - 1), \text{ para } V_g \geq 0 \quad (3.30)$$

$$I_g = -I_{so} \cdot A \cdot (e^{\frac{q \sqrt{|B \cdot V_g|}}{n \cdot k \cdot T}} - 1), \text{ para } V_g < 0 \quad (3.31)$$

Para implementar estas ecuaciones utilizaremos variables locales dentro de la función para facilitar su entendimiento:

Códigos	Descripción
*Para $V_g \geq 0$ FUNCTION VDD=CV(0)	Para tensiones de control $\geq 0$  Define la tensión de control a la función VDD
FUNCTION IGP {EXPO} +EXPO=EXP(Q*VDD/(N*K*TEMP));	EXPO define una variable local. + determina que es continuación de la línea anterior. Esta línea asigna un valor a la variable local y el ";" determina que no es el resultado de la función principal.
+ ISOW*(EXPO-1)	Resultado de la función IGP.

Continúa en la siguiente página

Códigos	Descripción
*Para $V_g < 0$ FUNCTION IGN {EXPO} +EXPO=EXP +((Q*SQRT(ABS(GEXP*VDD) +)))/(N*K*TEMP)); + (-ISOW*AFACT*(EXPO-1))	Define la corriente del diodo para tensiones de control $< 0$ .
FUNCTION IG IF (VDD>=0) THEN IGP + ELSE IGN	Controla que función del diodo se utiliza dependiendo del valor de la tensión de control utilizando el estamento IF-THEN-ELSE.

Una vez que tenemos las funciones del diodo procederemos a su modelado mediante una fuente de corriente controlada por tensión (VCCS) que mediante estas funciones se comportara como el diodo que deseamos:

Implementación:

Códigos	Descripción
	Definición del diodo puerta-drenador:
VCCS DGD nodo1 nodo2 1 nodo1 nodo2	Nombre del diodo DGD colocada entre el nodo1 y nodo2 dependiente de 1 tensión entre el nodo1 y el nodo2 ( $V_{gd}$ ).
+ [IG, 0]	El valor de la corriente viene determinada por la función IG, la conductancia es 0 (la introduciremos cuando implementemos el régimen dinámico).
+ G	Es una fuente estática.
+ NONLINEAR	Es no lineal
+ DERIV	La derivada son introducidas.
	Definición del diodo puerta-surtidor
VCCS DGS nodo1 nodo3 1 nodo1 nodo3	Nombre de la fuente DGS colocada entre el nodo1 y nodo2 dependiente de 1 tensión entre el nodo1 y nodo3 ( $V_{gs}$ ).
+ [IG,0]	
+ G	
+ NONLINEAR	
+ DERIV	

### 5.2.3 Implementación de la Fuente de Corriente

Al igual que en el apartado anterior definiremos las funciones de la fuente de corriente principal del transistor:

$$I_{ds} = f_1(V_{gs}) \cdot f_2(V_{ds}, V_{gs}) \cdot f_3(V_{ds}) \quad (4.22)$$

$$f_1[V_{gs}] = I_{dvc}(1 + \tanh(\beta \cdot (V_{gs} - V_c) + \gamma \cdot (V_{gs} - V_c)^3)) + I_{dvsb}(1 + \tanh(\delta \cdot (V_{gs} - V_c))) \quad (4.23)$$

$$f_2[V_{ds}, V_{gs}] = \left(1 + \frac{\lambda \cdot V_{ds}}{1 + \Delta \lambda \cdot (V_{gs} - V_{to})^2}\right) \quad (4.24)$$

$$f_3[V_{ds}] = \tanh(\alpha \cdot V_{ds}) \quad (4.25)$$

La implementación de las ecuaciones en APLAC se hace de la siguiente forma:

Códigos	Descripción
FUNCTION VGS=CV(0)	Asignamos la primera tensión de control a $V_{gs}$ .
FUNCTION VDS=CV(1)	Asignamos la primera tensión de control a $V_{ds}$ .
FUNCTION F1 { F11, F12 } + F11= CDVCW*(1+TANH(BETA* + (VGS-VC)+GAMMA*(VGS-VC)^3)); + F12= CDVSBW*(1+TANH + (DELTA*(VGS-VSB))); + (F11+F12)	Definimos la función F1 con dos variables locales F11, F22.
FUNCTION VTO=(VC-2/BETA)	Definimos la variable de la tensión umbral $V_{TO}$ .
FUNCTION F2 { F21 } + F21= (1+DXL*((VGS-VTO)^2)); + (1+LAMBDA*VDS/F21)	
FUNCTION F3 TANH(ALPHA*VDS)	
FUNCTION IDS=F1*F2*F3	Ecuación que modela la corriente de drenador.

Ahora implementaremos la fuente de corriente mediante VCCS modelando su comportamiento con la funciones anteriores:

Códigos	Descripción
VCCS IDS nodo2 nodo3 2 nodo1 nodo3 nodo2 nodo3	Fuente de corriente IDS colocado entre nodo3 y nodo2 dependiente de 2 tensiones: nodo1-nodo3 ( $V_{gs}$ ) y nodo2-nodo3 ( $V_{ds}$ ).
+ [IDS, 0,0]	La corriente de la fuente viene definida por IDS, las conductancias serán introducidas en el régimen dinámico.
+ G	
+ NONLINEAR	
+ DERIV	

### 5.2.4 Creación del Modelo del HFET en Continua

Una vez que ya hemos desarrollado los principales componentes del modelo pasamos al desarrollo del modelo en sí. El fichero generado tendrá extensión .i. Antes de la creación del modelo debemos simular el circuito para determinar los problemas que puedan surgir. Esto se debe a que, una vez creado el modelo no podremos acceder a sus funciones o parámetros, y sólo podríamos acceder a las variables que declaremos publicas.

El desarrollo del modelo en APLAC no tiene una forma única pero el desarrollo más optimizado es la expuesta a continuación.

Circuito completo del HFET en continua es:

Códigos	Descripción
DEFMODEL EFET 3 D G S	Nuevo modelo EFET con tres nodos de salida D (drenador) G(puerta) S(surtidor).
+ PARAM 20 BETA ALPHA VC + VSB LAMBDA DELTA DXL + RG RD RS GAMMA CDVSB + CDVC Q AFACT K N ISO + GEXP WW	El modelo en continua dispone de 20 variables opcionales las cuales pueden ser modificadas en la llamada al modelo.
DEFAULT BETA=2.8 DEFAULT ALPHA=5 DEFAULT VC=0.65 DEFAULT VSB=5	Listado de los valores por defecto de los parámetros opcionales para un HFET de enriquecimiento, estos valores serán tomados por APLAC si en la llamada al submodelo no se especifica el valor del parametro.

Continúa en la siguiente página

Códigos	Descripción
<pre> DEFAULT LAMBDA=0.05 DEFAULT DELTA=0 DEFAULT DXL=10 DEFAULT RG=0.2 DEFAULT RD=600 DEFAULT RS=600 DEFAULT GAMMA=4 DEFAULT CDVSB=0 DEFAULT CDVC=0.23E-3 DEFAULT Q=1.60022E-22 DEFAULT K=1.381E-23 DEFAULT AFACT=4 DEFAULT GEXP=0.075 DEFAULT ISO=1E-14 DEFAULT N=1.5 </pre>	
<pre> IF NOT(PARAMGIVEN(WW)) THEN CALL EXIT ("SE DEBE ESPECIFICAR W ") ENDIF </pre>	<p>Procedimiento que interrumpe la ejecución de la simulación si el parámetro ww (ancho del canal) no es especificado, en la ventana de salida se indica que el parámetro debe ser introducido.</p>
<pre> APLACVAR ISOW CALL ISOW=ISO*WW APLACVAR CDVSBW CALL CDVSBW=CDVSB*WW APLACVAR CDVCW CALL CDVCW=CDVC*WW APLACVAR RSW CALL RSW=RS/WW APLACVAR RGW CALL RGW=RG*WW APLACVAR RDW CALL RDW=RD/WW </pre>	<p>Variables afectadas por el ancho del canal.</p>
<pre> RES  RG G NODO1  RGW RES  RD D NODO2  RDW RES  RS S NODO3  RSW </pre>	<p>Resistencias del modelo:  Resistencia de puerta  Resistencia de drenador  Resistencia surtidor</p>
<pre> FUNCTION VDD=CV(0) </pre>	<p>Funciones de los diodo del HFET</p>
<pre> FUNCTION  IGP  {EXPO} + EXPO=EXP(Q*VDD)/(N*K*TEMP)); + ISOW*(EXPO-1) </pre>	
<pre> FUNCTION  IGN  {EXPO} + EXPO=EXP ((Q*SQRT(ABS + (GEXP*VDD)))/(N*K*TEMP)); + (-ISOW*AFACT*(EXPO-1)) </pre>	

Continúa en la siguiente página

Códigos	Descripción
VCCS DGD NODO1 NODO2 1 NODO1 NODO2 + [ IG, 0 ] + NONLINEAR + DERIV	Modelado del diodo puerta-drenador.
VCCS DGS NODO1 NODO3 1 NODO1 NODO3 + [ IG, 0 ] + NONLINEAR + DERIV	Modelado del diodo puerta surtidor.
FUNCTION VGS=CV(0)  FUNCTION VDS=CV(1)	Funciones de la fuente dependiente del HFET.
FUNCTION F1 { F11, F12 } + F11= CDVCW*(1+TANH(BETA*(VGS-VC)+ + GAMMA*(VGS-VC)^3)); + F12= CDVSBW*(1+TANH(DELTA*(VGS-VSB))); + (F11+F12)	
FUNCTION VTO=(VC-2/BETA)	
FUNCTION F2 { F21 } + F21= (1+DXL*(sqr(VGS-VTO))); + (1+LAMBDA*VDS/F21)	
FUNCTION F3 TANH(ALPHA*VDS)	
FUNCTION IDS=F1*F2*F3	
VCCS IDC NODO2 NODO3 2 NODO1 NODO3 NODO2 NODO3 + [ IDS, 0, 0 ] + G + NONLINEAR + DERIV	Modelado de la fuente dependiente del HFET
ENDMODEL	Final del modelo.

Y la llama al modelo se haría de la siguiente forma:

```
EFET          NOMBRE          NODO1 NODO2 NODO3 WW=
```

Para utilizar nuestro modelo en otro fichero debemos utilizar la directiva INCLUDE más la ruta ,(la forma de introducir la ruta difiere en sistema operativo WINDOWS y UNIX), donde se encuentra el submodelo por ejemplo:

#include "C:\MILIBRERIA\EFET.i" en el sistema operativo WINDOWS.

#include "/MILIBRERIA/EFET.i" en el sistema operativo UNIX.

El parámetro  $ww$ , aunque lo hemos designado como un parámetro opcional para APLAC, en el modelo es obligatorio. En APLAC podríamos introducirlo como parámetro obligatorio de la siguiente forma:

```
DEFMODEL EFET 3 D G S
+ FIXED WW
+ PARAM ...
```

con lo que la llamada al modelo sería:

```
EFET NOMBRE NODO1 NODO2 NODO3 valor
```

Sin embargo si no introdujéramos el valor, APLAC nos daría el error

APLAC versión ERROR: items missing

y no sabríamos cual es el parámetro que debemos introducir. En cambio de la forma en que nosotros lo hemos introducido, el error que nos produciría sería:

APLAC versión ERROR: exit message: SE DEBE ESPECIFICAR WW.

### 5.3 Creación del HFET en Régimen Dinámico

El circuito equivalente en régimen dinámico es el que se muestra en figura 5.4.

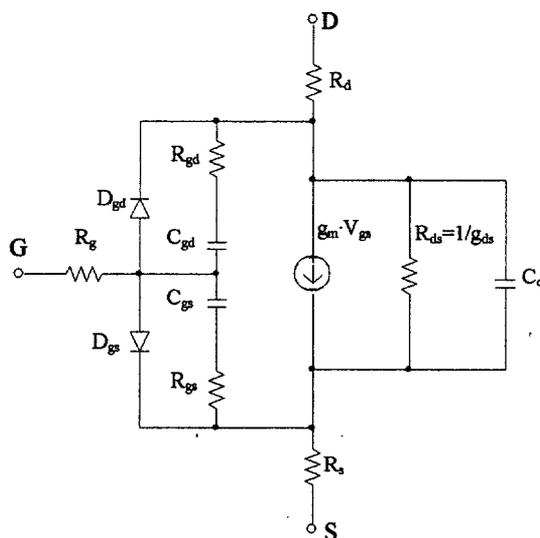


fig. 5.4 Circuito equivalente para régimen dinámico.

Debemos incluir en el modelo anterior las conductancias y capacitancias del modelo.

Primero modelamos las capacidades y cargas de los condensadores puerta-drenador y puerta surtidor:

$$C_{gs} = C_{gs0} + f_{c1}(V_{gs}) \cdot f_{c2}(V_{ds}) \quad (4.26)$$

$$\text{Para } V_{gs} < V_{at} \quad \begin{cases} f_{c1} = 0 \\ f_{c2} = 0 \end{cases} \quad (4.27)$$

$$\text{Para } V_{at} \leq V_{gs} < V_{bt} \quad \begin{cases} f_{c1} = C_{gs1} \cdot \frac{V_{gs} - V_{at}}{V_{bt} - V_{at}} \\ f_{c2} = 1 + V_{st} \cdot \tanh(\alpha \cdot V_s \cdot V_{ds}) \end{cases} \quad (4.28)$$

$$\text{Para } V_{gs} \geq V_{bt} \quad \begin{cases} f_{c1} = C_{gs1} \cdot (1 + C_f (V_{gs} - V_{bt})^m) \\ f_{c2} = 1 + V_{st} \cdot \tanh(\alpha \cdot V_s \cdot V_{ds}) \end{cases} \quad (4.29)$$

$$\text{Para } V_{gs} < V_{at} \quad Q_{gs} = C_{gs0} \cdot V_{gs} + Q_{c1} \quad (4.38)$$

$$\text{Para } V_{at} \leq V_{gs} < V_{bt} \quad Q_{gs} = C_{gs0} \cdot V_{gs0} + C_{gs1} \cdot \frac{V_{gs} - V_{at}}{2 \cdot (V_{bt} - V_{at})} \cdot f_{c2} + Q_{c2} \quad (4.39)$$

$$\text{Para } V_{gs} \geq V_{bt} \quad Q_{gs} = Q_{gs0} \cdot V_{gs} + C_{gs1} V_{gs} + C_f \cdot \frac{(V_{gs} - V_{bt})^{m+1}}{m+1} \cdot f_{c2} + Q_{c3} \quad (4.40)$$

$$Q_1 = Q_2, C_1 = C_2, \frac{\partial C_1}{\partial V_{gs}} = \frac{\partial C_2}{\partial V_{gs}} \quad (4.41)$$

$$Q_{c2} = Q_{c3} \quad (4.43)$$

$$Q_{c3} = Q_{c2} - C_{gs1} \cdot f_{c2} \cdot \frac{(V_{bt} + V_{at})}{2} \quad (4.44)$$

La implementación de las ecuaciones en APLAC sería de la siguiente forma:

Códigos	Descripción
FUNCTION FC1 CGS1W*(CV(0)-VAT) +/(VBT-VAT)	Función FC1 para la región de trabajo: $V_{at} \leq V_{gs} < V_{bt}$ .
FUNCTION FC2IM 1+VST* +TANH(ALPHA*VS*CV(1))	Función FC2 para la región de trabajo: $V_{at} \leq V_{gs} < V_{bt}$ y $V_{gs} \geq V_{bt}$ .
FUNCTION FC1M {LOCAL} + LOCAL=((CV(0)-VBT)*MFACT); + CGS1W*(1+FC*LOCAL)	Función FC1 para la región de trabajo: $V_{gs} \geq V_{bt}$ .
FUNCTION FC1= IF (CV(0)<VAT) THEN 0 + ELSE IF (CV(0)>=VAT AND CV(0)<VBT) + THEN FC1I + ELSE FC1M	Expresión general de fc1, para cualquier región de trabajo.
FUNCTION FC2=IF (CV(0)<VAT) THEN 0 +ELSE FC2IM	Expresión general de fc1, para cualquier región de trabajo.
FUNCTION CG=CGS0W+FC1*FC2	Capacidad para las diferentes regiones de trabajo.
FUNCTION QGME CGS0W*CV(0)+QC1	Función de la carga para la región de trabajo: $V_{gs} < V_{at}$ .
FUNCTION QGI {QGI1,QGI2} + QGI1= SQR(CV(0)-VAT); + QGI2= QGI1/(2*(VBT-VAT)); + CGS0W*CV(0)+CGS1W*QGI2*FC2+QC2	Función de la carga para la región de trabajo: $V_{at} \leq V_{gs} < V_{bt}$ .
FUNCTION QGM {QGM1,QGM2} + QGM1= (CV(0)-VBT)^(MFACT+1); + QGM2= CV(0)+FC*QGM1/(MFACT+1); + CGS0W*CV(0)+CGS1W*QGM2*FC2+QC3	Función de la carga para la región de trabajo: $V_{gs} \geq V_{bt}$ .
FUNCTION FUNCTIONQG=IF CV(0)<VAT + THEN QGME + ELSE IF (CV(0)>=VAT AND CV(0)<VBT) + THEN QGI + ELSE QGM	Función general de la carga.
APLACVAR QC2 CALL QC2=QC1	Condición de la interfase que evita problemas de convergencia.
FUNCTION QC3=QC2-CGS1W*FC2* + (VBT+VAT)/2	Condición de la interfase que evita problemas de convergencia.

Ahora modelamos las conductancias:

$$g_{gs} = \frac{\partial I_{gs}}{\partial V_{gs}} = I_{so} \cdot \frac{q \cdot e^{\frac{q \cdot V_{gs}}{n \cdot k \cdot T}}}{n \cdot k \cdot T}, \text{ para } V_{gs} \geq 0 \quad (4.32)$$

$$g_{gs} = \frac{\partial I_{gs}}{\partial V_{gs}} = I_{so} \cdot A \cdot \frac{q \cdot B \cdot e^{\frac{q \cdot \sqrt{|B \cdot V_{gs}|}}{n \cdot k \cdot T}}}{2 \cdot n \cdot k \cdot T \cdot \sqrt{|B \cdot V_{gs}|}}, \text{ para } V_{gs} < 0 \quad (4.33)$$

$$g_m = f_3 \cdot \left\{ f_2 \cdot \left[ I_{dVC} \left( \frac{\beta + \gamma \cdot 3 \cdot (V_{gs} - V_C)^2}{\cosh^2(\beta \cdot (V_{gs} - V_C) + \gamma \cdot (V_{gs} - V_C)^3)} \right) + I_{dVSB} \left( \frac{\delta}{\cosh^2(\delta \cdot (V_{gs} - V_{SB}))} \right) \right] \right. \quad (4.35)$$

$$\left. - f_1 \cdot \left[ \frac{V_{ds} \cdot \lambda \cdot \Delta \lambda \cdot 2 \cdot (V_{gs} - V_{to})}{(1 + \Delta \lambda \cdot (V_{gs} - V_{to})^2)^2} \right] \right\}$$

$$g_{ds} = f_1 \cdot \left[ \left( \frac{\lambda}{1 + \Delta \lambda \cdot (V_{gs} - V_{to})^2} \right) \cdot \tanh(\alpha \cdot V_{ds}) + \left( 1 + \frac{\lambda \cdot V_{ds}}{1 + \Delta \lambda \cdot (V_{gs} - V_{to})^2} \right) \cdot \frac{\alpha}{\cosh^2(\alpha \cdot V_{ds})} \right] \quad (4.36)$$

Las funciones que modelan las ecuaciones de la conductancias son:

Códigos	Descripción
<pre> FUNCTION GDP {EXPO} + EXPO= EXP(Q*VDD/(N*K*TEMP)); + (ISOW*Q*EXPO)/(N*K*TEMP)                     </pre>	Función de conductancia $g_{gs}$ , $g_{gd}$ para $V_{gs} \geq 0$
<pre> FUNCTION GDN {EXPO} + EXPO=EXP((Q*SQRT(ABS(GEXP*VDD)) +/(N*K*TEMP)); +(ISOW*AFAC*Q*GEXP*EXPO) +/(2*N*K*TEMP*SQRT(ABS(GEXP*VDD)))                     </pre>	Función de conductancia $g_{gs}$ , $g_{gd}$ para $V_{gs} < 0$
<pre> FUNCTION GD IF (VDD&gt;=0) THEN GDP + ELSE GDN                     </pre>	Función de conductancia $g_{gs}$ , $g_{gd}$
<pre> FUNCTION GM {GM1, GM2, GM3} + GM1=CDVCW*((BETA+GAMMA*3*(sqr(VGS- +VC))) +/(sqr(COSH(BETA*(VGS-VC)+GAMMA +*((VGS-VC)^3)))); + GM2= CDVSBW*(DELTA +/(sqr(COSH(DELTA*(VGS-VSB)))); + GM3=(VDS*LAMBDA*DXL*2*(VGS-VTO)) +/(sqr(1+DXL*(sqr(VGS-VTO)))); + F3*( F2*( GM1+GM2)-F1*GM3 )                     </pre>	Función de conductancia de la fuente dependiente respecto a la primera tensión de control ( $V_{gs}$ ).
<pre> FUNCTION GDS {GDS1,GDS2,GDS3} + GDS1= LAMBDA/(1+DXL*(sqr(CV(0)-VTO))); + GDS2= 1+LAMBDA*CV(1)/(1+DXL +*(sqr(CV(0)-VTO))); + GDS3= ALPHA/(sqr(COSH(ALPHA*CV(1)))); + F1*(gds1*f3+gds2*gds3)                     </pre>	Función de conductancia de la fuente dependiente respecto a la segunda tensión de control ( $V_{ds}$ ).

Ahora implementaremos los condensadores puerta-surtidor y puerta-drenador mediante una VCCS, modelando su comportamiento con las funciones anteriores:

Códigos	Descripción
VCCS CGS nodo1 nodo3 2 nodo1 nodo3 nodo2 nodo3 + [QG,CG,0]  + C + NONLINEAR + DERIV	Condensador $C_{GS}$ colocado entre nodo1 y nodo2, depende de dos tensiones $V_{gs}$ y $V_{ds}$ . La carga viene determinada por la función QG y la capacidad por la función CG.  Elemento dinámico.
VCCS CGD nodo1.nodo2 2 nodo1 nodo2 nodo3 nodo2 + [QG,CG,0] + C + NONLINEAR + DERIV	Condensador $C_{GD}$ depende de $V_{gd}$ y $-V_{ds}$ y para poder utilizar las mismas funciones que el condensador $C_{GS}$ invertimos el signo de $V_{ds}$ invirtiendo los nodos de la segunda tensión de control (nodo3 nodo2) obteniendo $V_{SD}$ .

Al implementar los condensadores con VCCS estos adquieren polaridad, lo que producirá un error en las simulaciones AC o TRANSITORIO según en que posición los coloquemos, debido a que en el modelo los condensadores no tienen polaridad. APLAC no dispone de ninguna directiva que nos compruebe que tipo de análisis se está llevando a cabo. Por este motivo hemos tenido que introducir un parámetro, que llamamos análisis, y unas funciones externas al modelo.

En la llamada al modelo se debe introducir un "0" en el valor del parámetro análisis si la simulación que vamos a realizar es en continua o transitorio y "1" si es en alterna.

La funciones serían:

Códigos	Descripción
FUNCTION QGP=IF ANALISIS==1 THEN QG + ELSE QG	Si el valor del parámetro análisis es igual a uno se invertirá el signo de la carga del condensador.
FUNCTION CGP=IF ANALISIS==1 THEN CG + ELSE CG	Si el valor del parámetro análisis es igual a uno se invertirá el signo de la capacidad del condensador.

## 5.4 Modelo Completo del HFET de Enriquecimiento

Una vez desarrollado el modelo en DC, TRANSITORIO y AC ya podemos realizar el modelo completo del HFET :

Códigos	Descripción
DEFMODEL EFET 3 D G S	Defenición del modelo
+ PARAM 32 BETA ALPHA VC + VSB LAMBDA DELTA DXL + RG RD RS GAMMA CDVSB + CDVC WW Q AFACT K + N ISO GEXP CGS0 CGS1 + VST VAT VBT FC VS MFACT + QC1 CDS TD ANALISIS	El modelo completo dispone de 30 parámetros opcionales y' dos (ww, análisis) forzados a obligatorios.
DEFAULT BETA=2.8 DEFAULT ALPHA=5 DEFAULT VC=0.65 DEFAULT VSB=5 DEFAULT LAMBDA=0.05 DEFAULT DELTA=0 DEFAULT DXL=10 DEFAULT RG=0.2 DEFAULT RD=600 DEFAULT RS=600 DEFAULT GAMMA=4 DEFAULT CDVSB=0 DEFAULT CDVC=0.23E-3 DEFAULT Q=1.60022E-22 DEFAULT K=1.381E-23 DEFAULT AFACT=4 DEFAULT GEXP=0.075 DEFAULT N=1.5 DEFAULT ISO=1E-14 DEFAULT CGS0=2.5E-16 DEFAULT CGS1=9E-16 DEFAULT VST=1 DEFAULT VAT=-0.3 DEFAULT VBT=0.3 DEFAULT FC=1.5 DEFAULT VS=0.4 DEFAULT MFACT=1 DEFAULT QC1=0 DEFAULT CDS=2E-16 DEFAULT TD=3E-12	Listado de los valores por defecto de los parámetros opcionales para un HFET de enriquecimiento, estos valores serán tomados por APLAC si en la llamada al submodelo no se especifica el valor del parametro.

Continúa en la siguiente página

Códigos	Descripción
<pre> APLACVAR ISOW CALL ISOW=ISO*WW APLACVAR CDVSBW CALL CDVSBW=CDVSB*WW APLACVAR CDVCW CALL CDVCW=CDVC*WW APLACVAR CGS0W CALL CGS0W=CGS0*WW APLACVAR CGS1W CALL CGS1W=CGS1*WW APLACVAR RSW CALL RSW=RS/WW APLACVAR RGW CALL RGW=RG*WW APLACVAR RDW CALL RDW=RD/WW APLACVAR CDSW=CDS*WW </pre>	Variables afectadas por el ancho del canal.
<pre> IF NOT(PARAMGIVEN(ANALISIS)) THEN CALL EXIT ("SE DEBE ESPECIFICAR EL TIPO + DE ANALISIS QUE SE VA A + REALIZAR.;DC,TRAN=0, AC=1") ENDIF </pre>	Procedimiento que interrumpe la ejecución de la simulación si el parámetro análisis no es especificado, en la ventana de salida se indica que el parámetro debe ser introducido.
<pre> IF (ANALISIS !=0 AND ANALISIS !=1) THEN CALL EXIT (" ANALISIS TIENE QUE SER + PARA DC,TRAN=0 O AC=1 NO SE PERMITEN + VALORES DIFERENTES") ENDIF </pre>	Procedimiento que interrumpe la ejecución de la simulación si el parámetro análisis es distinto de 0 o 1.
<pre> IF NOT(PARAMGIVEN(WW)) THEN CALL EXIT ("SE DEBE ESPECIFICAR W ") ENDIF </pre>	Procedimiento que interrumpe la ejecución de la simulación si el parámetro ww (ancho del canal) no es especificado, en la ventana de salida se indica que el parámetro debe ser introducido.
<pre> RES   RG   G   NODO1 RGW RES   RD   D   NODO2 RDW RES   RS   S   NODO3 RSW </pre>	Resistencias del modelo.
<pre> FUNCTION VDD=CV(0)  FUNCTION   IGP   {EXPO} + EXPO=EXP(Q*VDD/(N*K*TEMP)); + ISOW*(EXPO-1)  FUNCTION   IGN   {EXPO} + EXPO=EXP ((Q*SQRT(ABS(GEXP*VDD))) + /(N*K*TEMP)); + (-ISOW*AFACT*(EXPO-1))  FUNCTION   GDP   { EXPO } + EXPO= EXP(Q*VDD/(N*K*TEMP)); + (ISOW*Q*EXPO)/(N*K*TEMP) </pre>	Funciones que modelan el comportamiento de los díodos.

Continúa en la siguiente página

Códigos	Descripción
<pre> FUNCTION  GDN  {EXPO} + EXPO=EXP((Q*SQRT(ABS(GEXP*VDD))) + /(N*K*TEMP)); + (ISOW*AFACT*Q*GEXP*EXPO) + /(2*N*K*TEMP*SQRT(ABS(GEXP*VDD)))  FUNCTION  GD  IF (VDD&gt;=0) THEN GDP + ELSE GDN  FUNCTION  IG  IF (VDD&gt;=0) THEN IGP + ELSE IGN </pre>	
<pre> VCCS  DGD NODO1 NODO2 1 NODO1 NODO2 + [ IG, GD ] + NONLINEAR + DERIV </pre>	Diodos del modelo.
<pre> VCCS  DGS NODO1 NODO3 1 NODO1 NODO3 + [ IG, GD ] + NONLINEAR + DERIV </pre>	
<pre> FUNCTION  FC1I CGS1W*(CV(0)-VAT)/(VBT-VAT) FUNCTION  FC2IM 1+VST*TANH(ALPHA*VS*CV(1)) </pre>	Funciones que modelan el comportamiento de los condensadores
<pre> FUNCTION FC1M {LOCAL} + LOCAL=FC*((CV(0)-VBT)^MFACT); + CGS1W*(1+LOCAL)  FUNCTION FC1= IF (CV(0)&lt;VAT) THEN 0 + ELSE IFCV(0)&gt;=VAT AND CV(0)&lt;VBT) THEN FC1I + ELSE FC1M  FUNCTION FC2=IF (CV(0)&lt;VAT) THEN 0 + ELSE FC2IM FUNCTION CG=CGS0W+FC1*FC2 </pre>	
<pre> APLACVAR QC2 CALL QC2=QC1 FUNCTION QC3=QC2-CGS1W*FC2*(VBT+VAT)/2 </pre>	
<pre> FUNCTION QGME CGS0W*CV(0)+QC1 FUNCTION QGI {QGI1,QGI2} + QGI1= SQR(CV(0)-VAT); + QGI2= QGI1/(2*(VBT-VAT)); + CGS0W*CV(0)+CGS1W*QGI2*FC2+QC2 </pre>	
<pre> FUNCTION QGM {QGM1,QGM2} + QGM1= (CV(0)-VBT)^(MFACT+1); + QGM2= CV(0)+FC*QGM1/(MFACT+1); + CGS0W*CV(0)+CGS1W*QGM2*FC2+QC3 </pre>	
<pre> FUNCTION QG=IF CV(0)&lt;VAT THEN QGME + ELSE IF (CV(0)&gt;=VAT AND CV(0)&lt;VBT) THEN QGI + ELSE QGM </pre>	
<pre> FUNCTION QGP=IF ANALISIS==1 THEN -QG + ELSE QG </pre>	
<pre> FUNCTION CGP=IF ANALISIS==1 THEN -CG + ELSE CG </pre>	

Continúa en la siguiente página

Códigos	Descripción
VCCS CGS NODO1 NODO3 2 NODO1 NODO3 NODO2 NODO3 + [QGP,CGP,0] + C + NONLINEAR + DERIV	Condensadores del modelo.
VCCS CGD NODO1 NODO2 2 NODO1 NODO2 NODO3 NODO2 + [QGP,CGP,0] + C + NONLINEAR + DERIV	
CAP CDS NODO2 NODO3 CDSW	
FUNCTION VGS=CV(0)	Funciones que modelan el comportamiento de la fuente dependiente.
FUNCTION VDS=CV(1)	
FUNCTION F1 { F11, F12 } + F11= CDVCW*(1+TANH(BETA*(VGS-VC)+GAMMA + *(VGS-VC)^3)); + F12= CDVSBW*(1+TANH(DELTA*(VGS-VSB))); + (F11+F12)	
FUNCTION VTO=(VC-2/BETA)	
FUNCTION F2 { F21 } + F21= (1+DXL*(sqr(VGS-VTO))); + (1+LAMBDA*VDS/F21)	
FUNCTION F3 TANH(ALPHA*VDS)	
FUNCTION GM { GM1, GM2, GM3 } +GM1=CDVCW*((BETA+GAMMA*3*(sqr(VGS-VC))) + /{(sqr(COSH(BETA*(VGS-VC)+GAMMA*(VGS-VC)^3)))); + GM2= CDVSBW*(DELTA/(sqr(COSH(DELTA*(VGS- VSB)))); +GM3=(VDS*LAMBDA*DXL*2*(VGS-VTO)) + /{(sqr(1+DXL*(sqr(VGS+VTO)))); + F3*( F2*( GM1+GM2)-F1*GM3 )*COS(2*PI*TD*F)	
FUNCTION GDS {GDS1,GDS2,GDS3} + GDS1= LAMBDA/(1+DXL*(sqr(CV(0)-VTO))); + GDS2= 1+LAMBDA*CV(1)/(1+DXL*(sqr(CV(0)-VTO))); + GDS3= ALPHA/(sqr(COSH(ALPHA*CV(1)))); + F1*(GDS1*F3+GDS2*GDS3)	
FUNCTION IDS=F1*F2*F3	

Continúa en la siguiente página

Códigos	Descripción
VCCS IDC NODO2 NODO3 2 NODO1 NODO3 NODO2 NODO3 + [ IDS, GM, GDS] + NONLINEAR + DERIV ENDMODEL	Fuente dependiente del modelo.      Final del modelo.

## 5.5 Errores Típicos

Algunos errores que podemos cometer y que son difíciles de detectar porque APLAC realiza la simulación sin encontrar error son los que se enumeran a continuación:

- Cuando queremos determinar una región para realizar una comparación no podemos introducirlo de esta forma ( $A \leq C < B$ ) puesto que APLAC no comprobaría que  $C < B$ . La forma correcta de realizarlo sería: ( $C \geq A$  AND  $C < B$ ).
- La comparación de dos elementos no puede realizarse de esta forma ( $A = B$ ) puesto que APLAC asignaría el valor de B a A y siempre serían iguales, la forma correcta es ( $A == B$ ).
- Realizar una llamada a un parámetro o función antes de su declaración.
- Anidar varios bloques de DefModel.
- Un error producido por una función que hallamos creado es difícil de detectar, pero podemos analizar las funciones que creamos y analizar los valores que nos devuelve APLAC. Por ejemplo vamos a comprobar los valores de salida que nos da las funciones de la fuente. Para ello creamos un fichero con extensión .i con las funciones y los parámetros.  $V_{ds}$  y  $V_{gs}$  los introducimos como parámetros y  $V_{ds}$  fijo a 2.5. Realizamos este barrido el siguiente barrido:

```

SWEEP "COMPROBACIÓN"
+ LOOP 8 APLACVAR VGS LIN 0.1 0.8
PRINT S "IDS=" BL REAL IDS LF
ENDSWEEP

```

La salida que nos da APLAC es la que se muestra en la figura 5.5.

```

TextOutput
File Edit Options Help

APLAC 7.50 Wed May 30 2001 at 06:13:56
Copyright (c) APLAC Solutions Corporation, Finland, 1998-1999

IDS= 151.550u
IDS= 461.295u
IDS= 1.101m
IDS= 2.137m
IDS= 3.504m
IDS= 5.062m
IDS= 6.672m
IDS= 8.217m

End of APLAC 7.50 Wed May 30 2001 at 06:13:56 (CPU-time = 31 ms)

```

fig. 5.5 Salida de la función

Como se puede observar APLAC nos devuelve el valor IDS para cada VGS del barrido con lo que podemos comprobar el funcionamiento de las funciones. También podemos almacenar los datos de salida con la directiva PRINT APPENDFILE, esta directiva también nos permite almacenar los datos de una curva en un fichero. La forma de utilizar esta directiva sería por ejemplo:

```
PRINT APPENDFILE=/libreria/result real vtran(9)
```

guardará los datos de la curva en transitorio del nodo9 en el fichero resul.

## 5.6 Creación del Modelo del HFET con el Editor de Esquemas de APLAC

Desarrollaremos en este apartado el modelo del HFET haciendo uso del editor de esquemas y lo convertiremos en un submodelo para poder ser utilizado en APLAC como un componente normal o en cualquier otro circuito.

### 5.6.1 Pasos Necesarios para Crear el Submodelo

Lo primero que debemos realizar es el circuito. Para ello insertaremos los componentes y haremos las conexiones. Para insertar las VCCS debemos indicar

a APLAC de cuantas tensiones de control dependen. Se debe tener en cuenta la simbología que utiliza APLAC para representar una VCCS (ver figura 5.6).

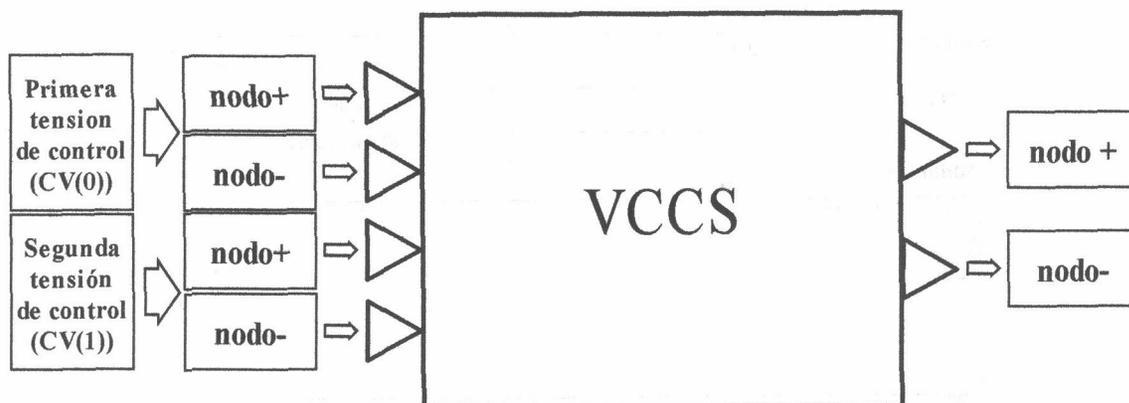


fig. 5.6. Simbología de una VCCS dependiente de dos tensiones de control

Una vez insertados los componentes y realizadas las conexiones pasaremos a introducir los códigos que modelan las VCCS y los procedimientos de interrupción. Para ello utilizaremos los controles objetos como texto e introducimos las funciones en lenguaje APLAC (figura 5.7).

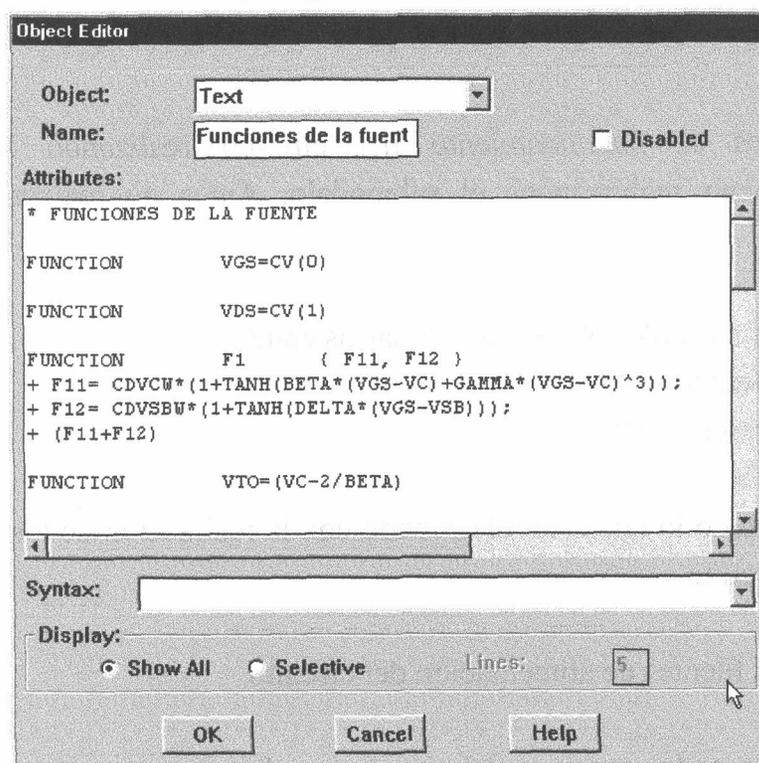


fig 5.7 Inserción de la funciones de la fuente principal de transistor mediante el editor de objeto

El siguiente paso es definir los atributos de los componentes. Esto hace tal y como se indica en la figura 5.8.

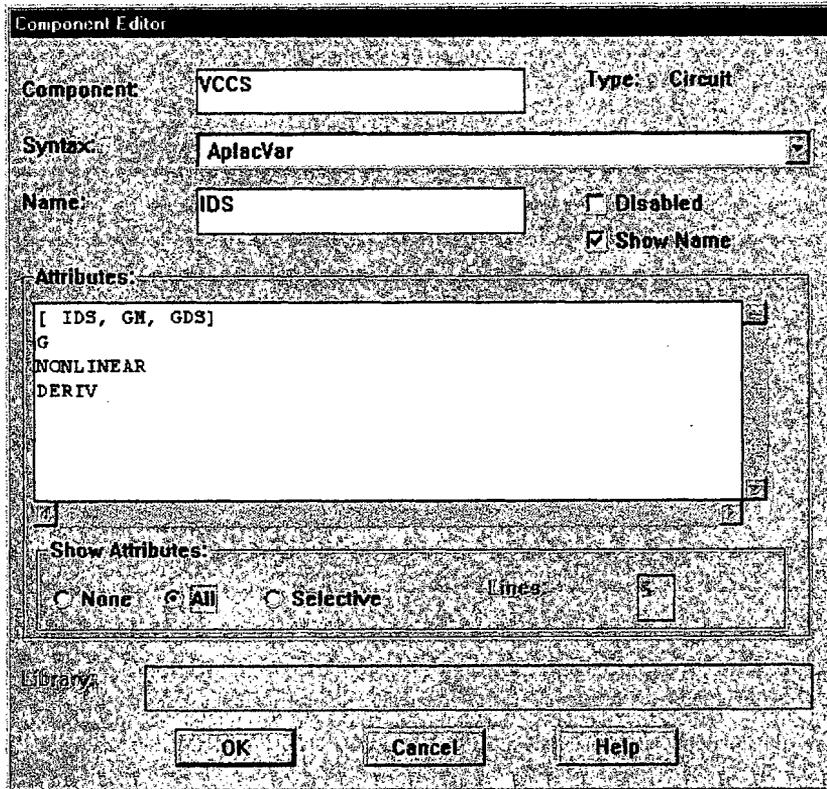


fig. 5.8 Atributos de los elementos

Comprobamos el funcionamiento del circuito realizando los análisis necesarios y luego realizaremos el submodelo. Antes de realizar este paso debemos:

1. Quitar los controles objetos innecesarios como:
  - Comandos de análisis,
  - Optimizaciones,
  - Salidas,
  - Y todo lo que esté relacionado con la realización de los análisis del circuito.
2. Quitar las fuentes de alimentación del circuito.
3. Definir los nodos externos del circuito con los elementos de nodo Input y Output.

- Convertir el diagrama objeto del circuito en una definición de modelo. Borraremos en la lista de objeto el diagrama de circuito y añadiremos la definición del modelo (figura 5.9).

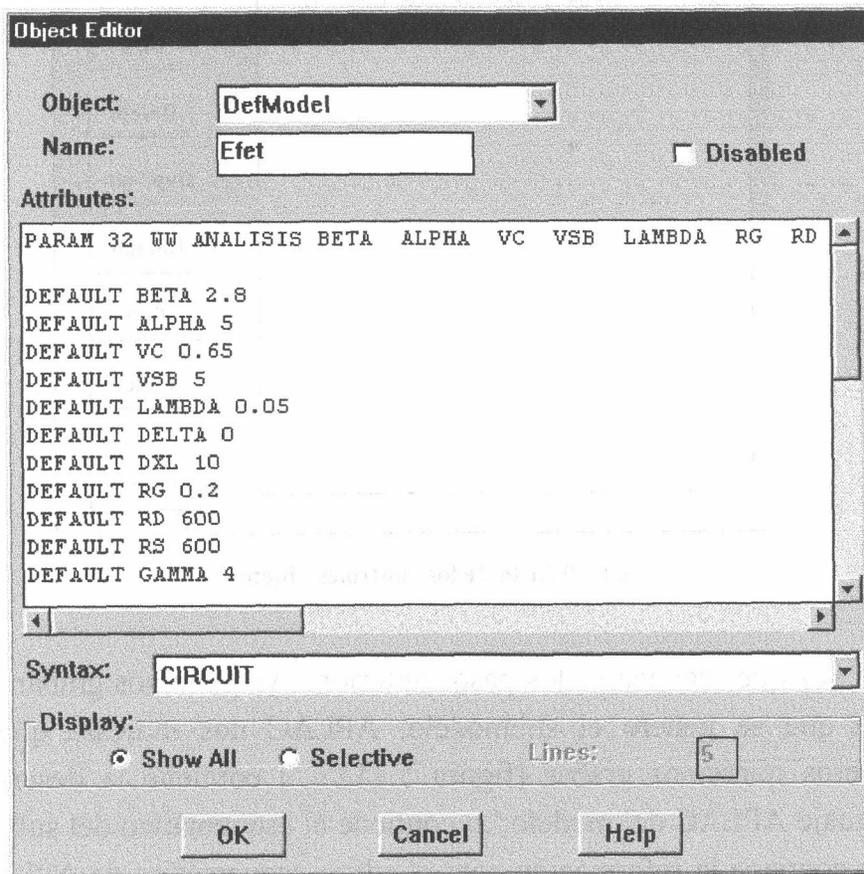


fig. 5.9 Definición del modelo.

- El orden de los controles objetos es muy importante por que son ejecutados por APLAC secuencialmente, en nuestro submodelo el orden es que se muestra en la figura 5.10.

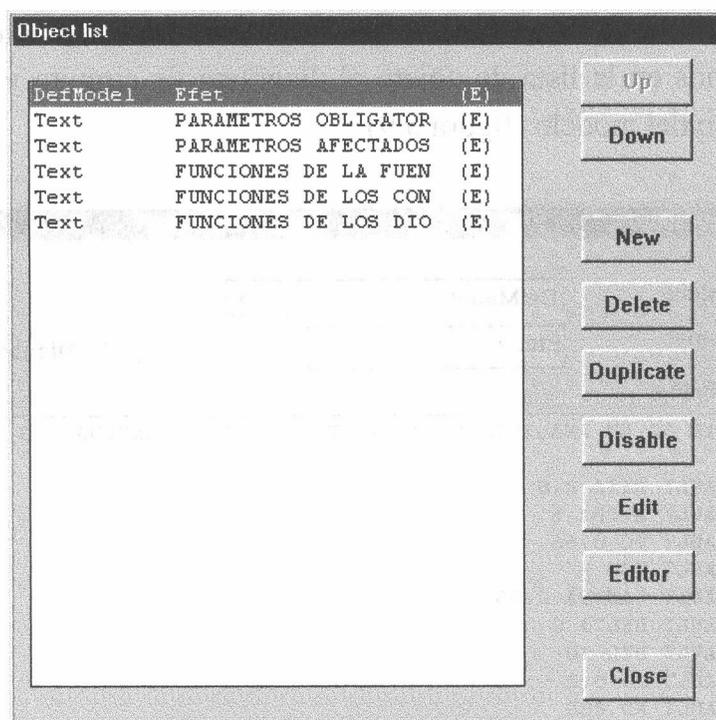


fig. 5.10 Lista de los controles objetos

6. Una vez que realicemos los pasos anteriores ya podemos grabar el fichero para que se genere el submodelo. APLAC nos indicará que tipo de ficheros queremos grabar (figura 5.11) : .i contiene la descripción en lenguaje APLAC del modelo; .n contiene el esquemático del submodelo y .sub contiene la información para el editor esquemático de APLAC sobre el símbolo usado para el modelo, número de nodos, y números de posibles parámetros y sus valores por defecto.

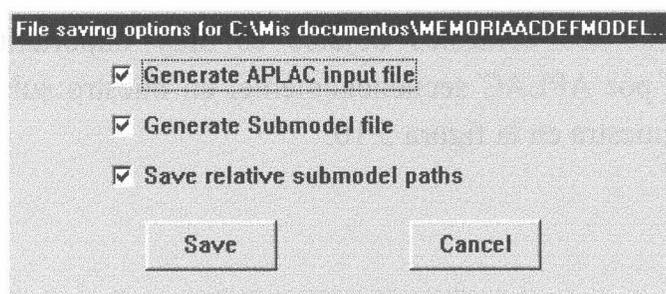


fig. 5.11 Petición de grabación.

Finalmente el circuito quedaría como se muestra en el anexo B.

El submodelo aparecerá con una forma determinada por APLAC (figura 5.13) que puede ser modificada utilizando la forma de un componente que disponga APLAC. Para ello en el menú Submodel seleccionamos Change Symbol and Pin Order y elegimos una forma semejante que disponga APLAC para nuestro componente (figura 5.14). Seguidamente ponemos de forma correcta los pines del componente (figura 5.15).



fig. 5.13 Presentación por defecto



fig. 5.14 Presentación por elección

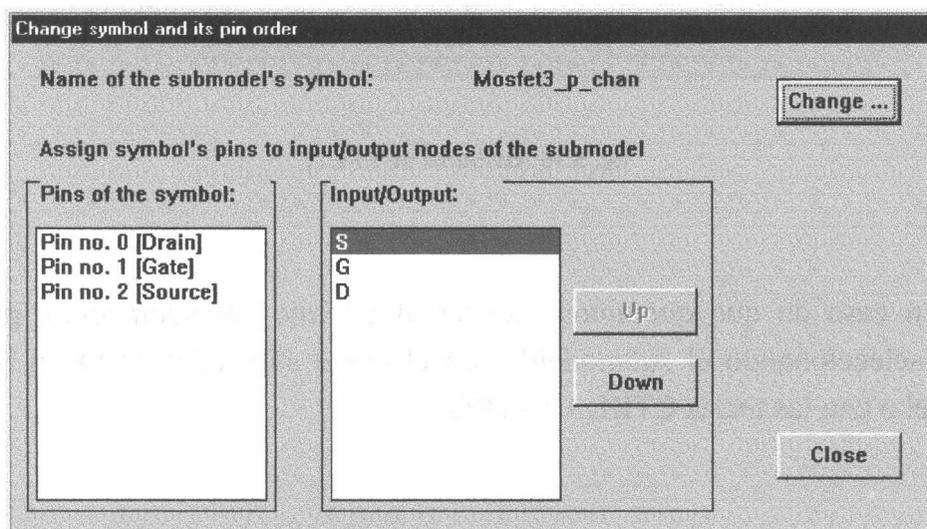


fig 5.15 Modificación del símbolo y el orden de los pin

Una vez modificado el símbolo y el orden de los pines adecuándolo al nuevo símbolo, lo grabamos y en la llama a nuestro submodelo aparecerá con el nuevo símbolo.

En los atributos de nuestro submodelo podremos modificar los parámetros opcionales e introducir los parámetros obligatorios del modelo (figura 5.16)

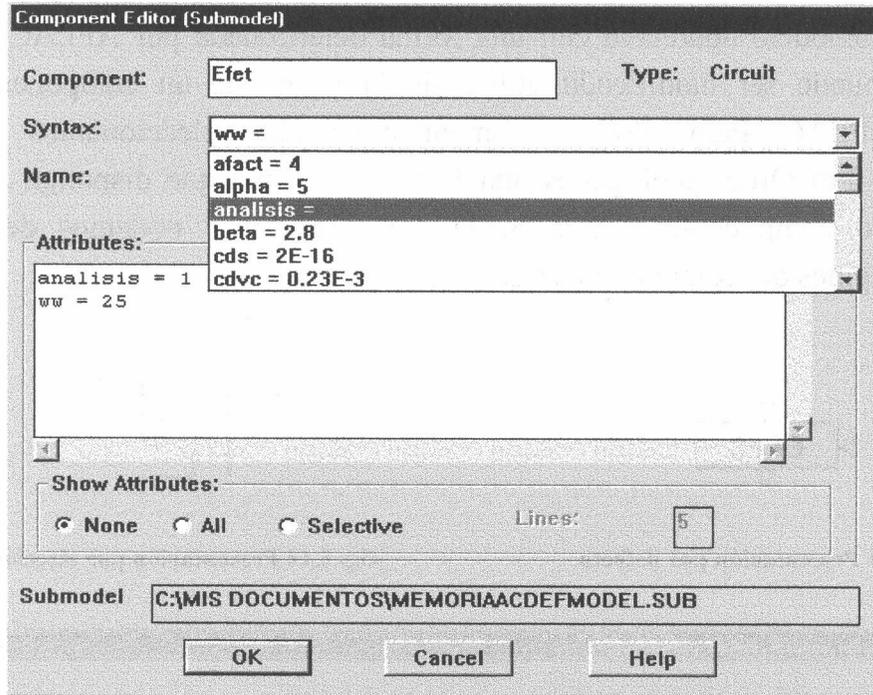


fig. 5.16 Atributos de EFET.

En caso de que queramos acceder al circuito del submodelo, podemos hacerlo seleccionando el submodelo y en el menú submodel en Open Selected submodel o con las teclas CTRL + INTRO.

# Validación de Resultados y Conclusiones

En el presente trabajo se ha estudiado, por un lado, el simulador eléctrico APLAC y, por otro, los transistores HFETs así como sus modelos eléctricos más importantes. Comenzamos en el primer capítulo haciendo una introducción a APLAC. Luego en el capítulo 2 se vió la creación de circuitos en lenguaje APLAC y en el editor de esquemas de APLAC y la realización de las simulaciones. Se hizo hincapié en el hecho de que el editor de esquemáticos es un programa interactivo aunque se puede trabajar con el simulador utilizando el lenguaje APLAC.

En el tercer capítulo se estudiaron las heteroestructuras y los transistores de efecto de campo de heteroestructura (HFETs). El estudio de este tipo de transistores se organizó señalando cual es la estructura básica así como el principio de funcionamiento de los mismos. También se presentaron algunas consideraciones tecnológicas en cuanto a la fabricación de dichos transistores.

Siguiendo con el estudio de los transistores HFET, en el cuarto capítulo revisamos el modelo de Angelov [6] el cual, a pesar de ser un modelo semiempírico, es hoy por hoy el modelo más aceptado y sobre el que se basan la mayoría de modelos que hay en la actualidad para los HFET. Un ejemplo de esto es el modelo del FhGIAF que es el que nosotros hemos implementado, y que, por tanto, también fue objeto de estudio en este capítulo.

En el capítulo 5 se hace una descripción detallada de cómo se introduce un modelo determinado dentro de APLAC. Esta tarea no puede ser llevada a cabo sin conocer, por un lado, al dispositivo y su modelo y, por el otro, el simulador y

su forma de implementar los diferentes dispositivos. Por tanto, los estudios realizados en los capítulos anteriores se convierten en imprescindibles. El modelo implementado es el modelo del HFET del FhGIAF que se presentó en el capítulo 4.

Por otro lado, es importante notar que a pesar de que la forma en como se implementa un determinado modelo dentro de APLAC depende directamente del modelo y del simulador, los procedimientos explicados en este capítulo pueden ser aplicados a la implementación de otros dispositivos.

Para finalizar, en este capítulo comenzaremos llevando a cabo la validación de los resultados obtenidos comparándolos con los resultados que da IAFSPICE, para acabar presentamos una serie de conclusiones.

## 6.1 Validación de los Resultados Obtenidos

La validación de los resultados obtenidos se realiza comparando las simulaciones hechas con el modelo que implementamos en APLAC y las hechas con IAFSPICE.

### 6.1.1 Análisis en DC

El circuito utilizado para la validación de los resultados en dc es el que se muestra en la figura 6.1. Se trata del circuito típico utilizado para realizar la medida de las características de salida y entrada en dc de un transistor. De esta forma, haciendo un barrido de la tensión  $V_{DS}$  para diferentes valores de la tensión  $V_{GS}$  tenemos la característica  $I_{DS}$  vs.  $V_{DS}$ . De la misma forma, haciendo un barrido de  $V_{GS}$  para una  $V_{DS}$  fija, obtenemos la característica  $I_{DS}$  vs.  $V_{GS}$ .

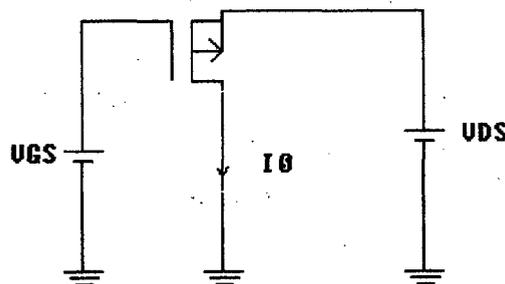


fig. 6.1 Circuito para evaluar las curvas características en dc de un transistor HFET.

Implementaremos el circuito de la figura 6.1 y realizaremos el análisis en APLAC, tal y como se muestra en el siguiente archivo:

Códigos	Descripción
#INCLUDE C:\PROYECTO\EFET.I	Incluimos el modelo del Hfet que desarrollamos en el capítulo5 .
APLACVAR VD APLACVAR VG	Variables que utilizaremos en el barrido.
EFET EFET1 nodo1 nodo2 GND + ww = 25 + analisis = 0	Llamada al submodelo Efet con ancho de canal de 25 $\mu$ para realizar un análisis en dc.
Volt VGS nodo2 GND + DC=VG Volt VDS nodo3 GND + DC=VD	Fuentes de alimentación cuyo valor en dc depende de las variables $V_d$ y $V_g$ .
Short VMEAS nodo3 nodo1 +I=I0	Para poder medir una corriente en APLAC debemos utilizar un Amperímetro, el elemento Short define un amperímetro y la corriente que pasa por él la hemos llamado I0.
SWEEP " Curvas Características en dc " + DC + LOOP 8 APLACVAR VG + LIN 0.1 0.8 + LOOP 100 APLACVAR VD + LIN 0 2.5	Inicio del barrido. Análisis en continua. Bucle lineal de 8 puntos de la variable $V_g$ desde 0.1 a 0.8. Bucle lineal de 100 puntos de la variable $V_d$ desde 0 a 2.5.
+ X "VDS" "VOLT" 0 2.5 + Y "IDS" "AMP" 0 6.0E-3	Determina el nombre del eje , la unidad en que se representa y se determina la escala del eje.
+ GRID + BIG_SCREEN	Se utilizará rejillas en la ventana gráfica. La ventana gráfica será a pantalla completa.
SHOW XY VD IDC(I0)	Se representará en el eje X la variable $V_d$ y en el eje Y la corriente en dc I0 y obtendremos la curva característica $I_{DS}$ vs. $V_{DS}$ .
ENDSWEEP	Final del barrido.

Continúa en la siguiente página

Códigos	Descripción
CALL VD=2.5	Dejamos $V_{ds}$ fija.
SWEEP " Curvas Características en dc " + DC	
+ LOOP 1000 APLACVAR VG	Hacemos un Barrido de $V_{gs}$ .
+ LIN -11 + X "VGS" "VOLT" -1 1 + Y "IDS" "AMP" 0 8.2E-3 + GRID + BIG_SCREEN	
SHOW XY VG IDC(I0)	Obtenemos la curva característica $I_{DS}$ vs $V_{GS}$ .
ENDSWEEP	

En las figuras 6.2.a y 6.2.b se pueden observar los resultados obtenidos para un transistor de enriquecimiento de  $0.5 \mu\text{m}$ . Las figuras 6.3.a y 6.3.b representan las mismas medidas para un transistor de depleción. En línea continua se muestran los resultados de IAFSPICE y con círculo la de nuestra implementación. Se observa que tanto para un caso como para el otro la coincidencia de los resultados es total.

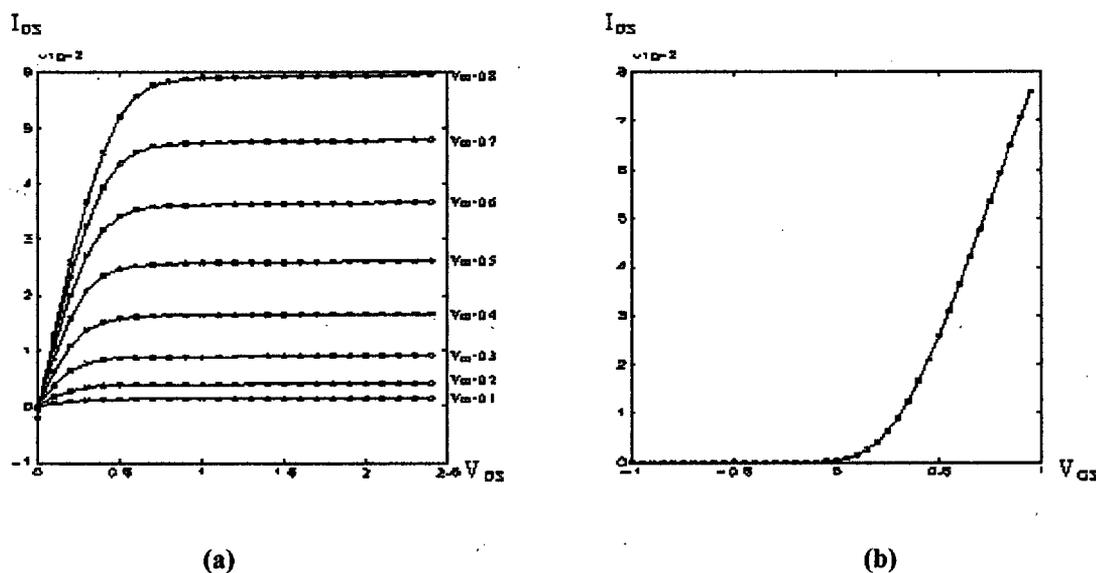


fig. 6.2 Características (a)  $I_{DS}$  vs.  $V_{DS}$  y (b)  $I_{DS}$  vs.  $V_{GS}$  de un HFET de enriquecimiento de  $0.5 \mu\text{m}$ .

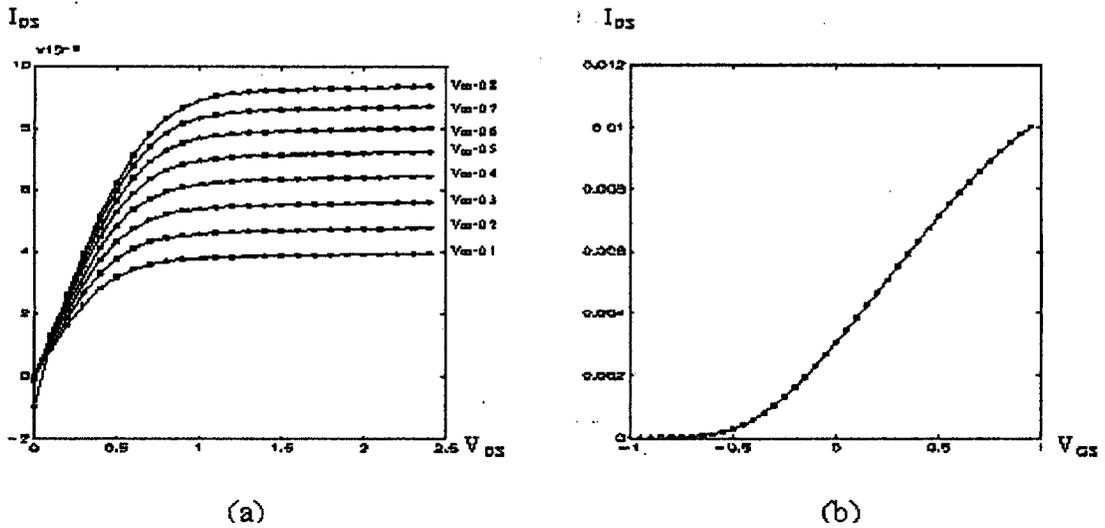


fig. 6.3 Características (a)  $I_{DS}$  vs.  $V_{DS}$  y (b)  $I_{DS}$  vs.  $V_{GS}$  de un HFET de enriquecimiento de  $0.5 \mu\text{m}$ .

### 6.1.2 Análisis en Régimen Transitorio

El circuito utilizado para el análisis en régimen transitorio se muestra en la figura 6.4. Esta figura consiste en tres inversores DCFL conectados en una cadena de a los que se le aplica un pulso a la entrada. Los inversores DCFL se construyen con un transistor de carga activa de tipo depleción de anchura aproximadamente la mitad del transistor activo que es de enriquecimiento.

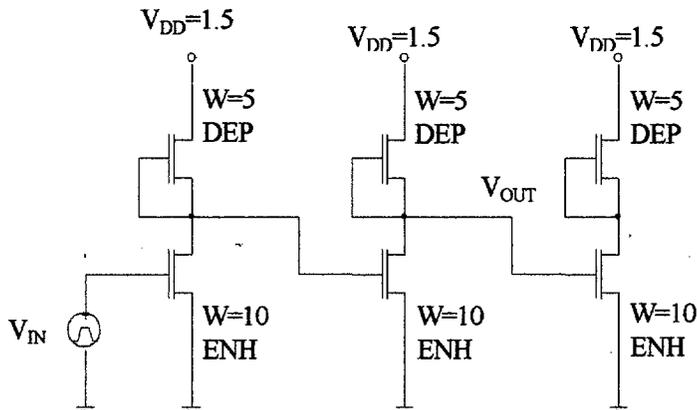


fig. 6.4 Cadena de tres inversores DCFL conectados en serie para comprobación del análisis en régimen transitorio.

Para facilitar la realización de este circuito y del próximo en el que utilizaremos 16 inversores, crearemos un submodelo en APLAC del inversor DCFL (figura 6.5) de la forma que se describe en el siguiente archivo:

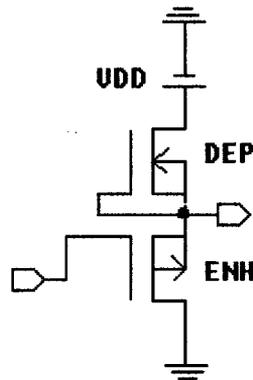


fig. 6.5 Submodelo del inversor creado en el editor de APLAC.

Códigos	Descripción
#include "C:\\PROYECTO\\EFET.I" #include "C:\\PROYECTO\\DFET.I"	Incluimos los submodelo del hfet de enriquecimiento y empobrecimiento.
DefModel        inversor + 2 In Out	Definimos el nuevo submodelo.
Efet ENH GND In Out + ww = 10 + analisis = 0	
Dfet DEP node1 Out Out + ww = 5 + analisis = 0	
Volt VDD node1 GND + DC = 1.5	
EndModel	

Una vez definido el submodelo del inversor creamos el circuito que vamos a analizar:

```

Códigos

#include "C:\\PROYECTO\\INVERSOR.I"

inversor inversor3 vout Output1

inversor inversor2 node1 vout

```

Continúa en la siguiente página

## Códigos

```

inversor inversor1 node2 node1

Volt Vin node2 GND
+PULSE = 0 1 100P 100P 100P 400P 1N

Sweep "ANALISIS TRANSITORIO"

+ LOOP 100 TIME LIN 0 1n
DISPLAY
+ WINDOW 1 Y "VOUT" VTRAN(VOUT)

EndSweep

```

Los resultados obtenidos se muestran en la figura 6.6

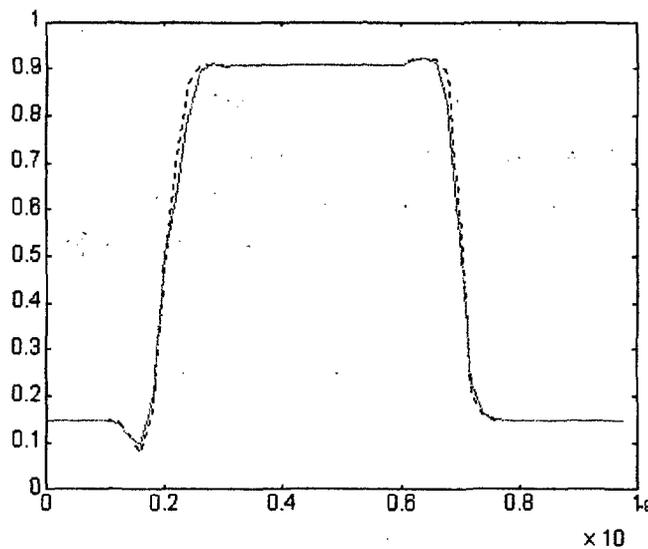


fig. 6.6 Resultados del análisis en régimen transitorio para tres inversores.  $V_{OUT}(IAFSPICE)$ : línea continua;  $V_{OUT}(APLAC)$ : línea de puntos.

El mismo análisis lo hemos hecho sobre una cadena de 16 inversores para comprobar el funcionamiento del programa con un mayor número de transistores. El fichero de entrada para esta simulación se muestra a continuación:

---

**Códigos**


---

```
#include "C:\\PROYECTO\\INVERSOR.I"

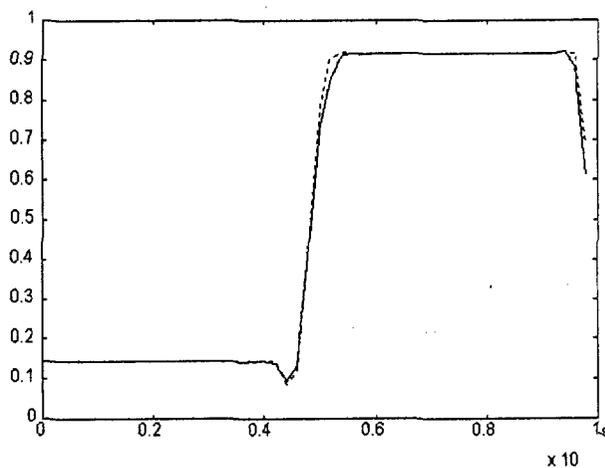
inversor inversor1 nodo1 nodo1
inversor inversor2 nodo1 nodo2
inversor inversor3 nodo2 nodo2
inversor inversor4 nodo2 nodo3
inversor inversor5 nodo3 nodo4
inversor inversor6 nodo4 nodo5
inversor inversor7 nodo5 nodo6
inversor inversor8 nodo6 nodo7
inversor inversor9 nodo7 nodo8
inversor inversor10 nodo8 nodo9
inversor inversor11 nodo9 nodo10
inversor inversor12 nodo10 nodo11
inversor inversor13 nodo11 nodo12
inversor inversor14 nodo12 output
inversor inversor15 output node13
inversor inversor16 node13 output2

Volt V1 NODO1 GND
+ PULSE = 0 1 100p 100p 100p 400p 1n

Sweep "ANALISIS TRANSITORIO"
+ LOOP 50 TIME LIN 0 9.8000000000E-10
DISPLAY
+ WINDOW 0 Y "PULSO DE ENTRADA" VTRAN(NODO1)
+ WINDOW 1 Y "SALIDA DECIMOCUARTO INVERSOR" VTRAN(OUTPUT)
EndSweep
```

---

Las curvas obtenidas a la salida del decimocuarto inversor tanto en IAFSPICE como en APLAC se muestran en la figura 6.7. Se puede observar como se ha retrasado la salida respecto a la señal que salía del segundo inversor (ver figura 6.6) debido al efecto de las capacidades internas del dispositivo.



**fig. 6.7** Resultados del análisis en régimen transitorio para 16 inversores.  $V_{OUT}$ (IAFSPIICE): línea continua;  $V_{OUT}$ (APLAC): línea de puntos.

Se ha realizado un análisis más para comprobar el funcionamiento del simulador para circuitos aún mayores. Se trata de un sumador tipo *full-adder*

cuyo esquema de bloques es el que se presenta en la figura 6.8. El fichero de entrada de APLAC de dicho circuito es el que se muestra a continuación:

**Códigos**

```
#include "C:\PROYECTO\EFET.I"
#include "C:\PROYECTO\DFET.I"

VOLT VDD 1 0 PULSE 0 1.5 10P 100P 1 1 1
EFET Z2 3 6 0 WW=10 analisis=0
DFET Z3 1 3 3 WW=5 analisis=0
EFET Z4 6 13 0 WW=10 analisis=0
DFET Z5 1 6 6 WW=5 analisis=0
EFET Z6 4 15 0 WW=10 analisis=0
DFET Z7 1 4 4 WW=5 analisis=0
EFET Z8 15 9 0 WW=10 analisis=0
DFET Z9 1 15 15 WW=5 analisis=0
EFET Z10 8 7 0 WW=10 analisis=0
DFET Z11 1 8 8 WW=5 analisis=0
EFET Z12 2 5 0 WW=10 analisis=0
DFET Z13 1 2 2 WW=5 analisis=0
EFET Z14 7 13 0 WW=10 analisis=0
DFET Z15 1 7 7 WW=5 analisis=0
EFET Z16 5 9 0 WW=10 analisis=0
DFET Z17 1 5 5 WW=5 analisis=0

VOLT VIN18 33 0 PULSE= 0 0.8 150P 25P 25P 0.475N 1N
EFET Z19 10 34 0 WW=12 analisis=0

DFET Z20 1 10 10 WW=6 analisis=0
EFET Z21 34 33 0 WW=12 analisis=0
DFET Z22 1 34 34 WW=6 analisis=0

VOLT VIN23 40 0 PULSE= 0 0.8 150P 25P 25P 0.975N 2N

EFET Z24 12 41 0 WW=12 analisis=0
DFET Z25 1 12 12 WW=6 analisis=0
EFET Z26 41 40 0 WW=12 analisis=0
DFET Z27 1 41 41 WW=6 analisis=0

VOLT VIN28 47 0 PULSE= 0 0.8 150P 25P 25P 1.975N 4N

EFET Z29 11 48 0 WW=12 analisis=0
DFET Z30 1 11 11 WW=6 analisis=0
EFET Z31 48 47 0 WW=12 analisis=0
DFET Z32 1 48 48 WW=6 analisis=0
EFET Z33 55 53 0 WW=12 analisis=0
DFET Z34 1 55 55 WW=6 analisis=0
EFET Z35 58 12 0 WW=10 analisis=0
DFET Z36 1 58 58 WW=5 analisis=0
EFET Z37 62 10 0 WW=10 analisis=0
DFET Z38 1 62 62 WW=5 analisis=0
EFET Z39 61 11 0 WW=12 analisis=0
DFET Z40 1 61 61 WW=6 analisis=0
EFET Z41 1 73 13 WW=10 analisis=0
EFET Z42 1 72 13 WW=10 analisis=0
EFET Z43 73 62 0 WW=11 analisis=0
EFET Z44 72 55 0 WW=11 analisis=0
EFET Z45 73 58 0 WW=11 analisis=0
EFET Z46 72 61 0 WW=11 analisis=0
DFET Z47 13 0 0 WW=7 analisis=0
DFET Z48 1 72 72 WW=8 analisis=0
DFET Z49 1 73 73 WW=8 analisis=0
```

Continúa en la siguiente página

**Códigos**

EFET	Z50	1	80	9	WW=10	analisis=0
EFET	Z51	1	79	9	WW=10	analisis=0
EFET	Z52	80	11	0	WW=11	analisis=0
EFET	Z53	79	53	0	WW=11	analisis=0
EFET	Z54	80	55	0	WW=11	analisis=0
EFET	Z55	79	61	0	WW=11	analisis=0
DFET	Z56	9	0	0	WW=7	analisis=0
DFET	Z57	1	79	79	WW=8	analisis=0
DFET	Z58	1	80	80	WW=8	analisis=0
EFET	Z59	1	87	53	WW=10	analisis=0
EFET	Z60	1	86	53	WW=10	analisis=0
EFET	Z61	87	12	0	WW=11	analisis=0
EFET	Z62	86	10	0	WW=11	analisis=0
EFET	Z63	87	62	0	WW=11	analisis=0
EFET	Z64	86	58	0	WW=11	analisis=0
DFET	Z65	53	0	0	WW=7	analisis=0
DFET	Z66	1	86	86	WW=8	analisis=0
DFET	Z67	1	87	87	WW=8	analisis=0

SWEEP "ANALISIS TRANSITORIO"  
+ LOOP 100 TIME LIN 0.000000e+00 1.000000e-08

DISPLAY  
+ WINDOW 2 Y "SOUT" VTRAN(9)  
+ WINDOW 1 Y "SI" VTRAN(10)  
+ WINDOW 1 Y "CI-1" VTRAN(11)  
+ WINDOW 1 Y "CI" VTRAN(12)  
+ WINDOW 0 Y "COUT" VTRAN(13)

ENDSWEEP

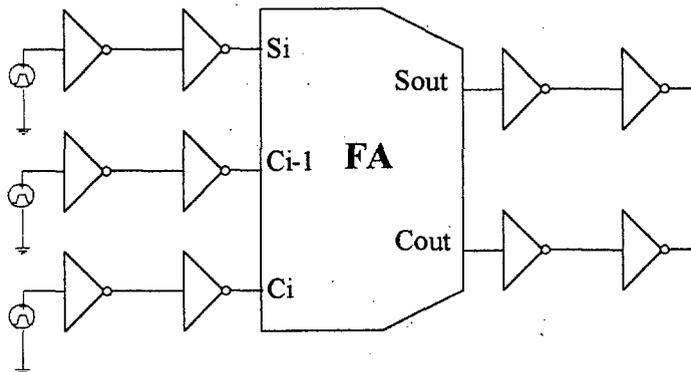


fig. 6.8 Diagrama de bloques de un sumador FA.

Este circuito está realizado con puertas DCFL y consta de 67 transistores HFET. Las señales aplicadas a las entradas se hacen pasar primero por dos inversores conectados en serie siendo sus formas de onda a la salida de los inversores las que se muestran en la figura 6.9. Las formas de onda a las salidas Sout y Cout se muestran en las figuras 6.10.a y 6.10.b respectivamente. En ellas se observa de nuevo cómo se superponen las predicciones.

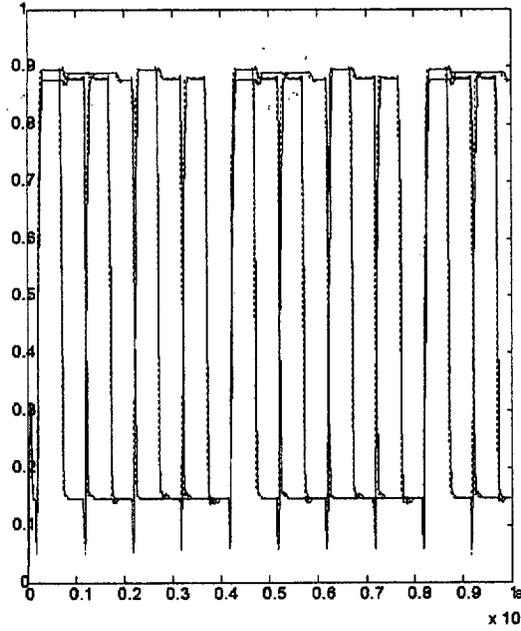
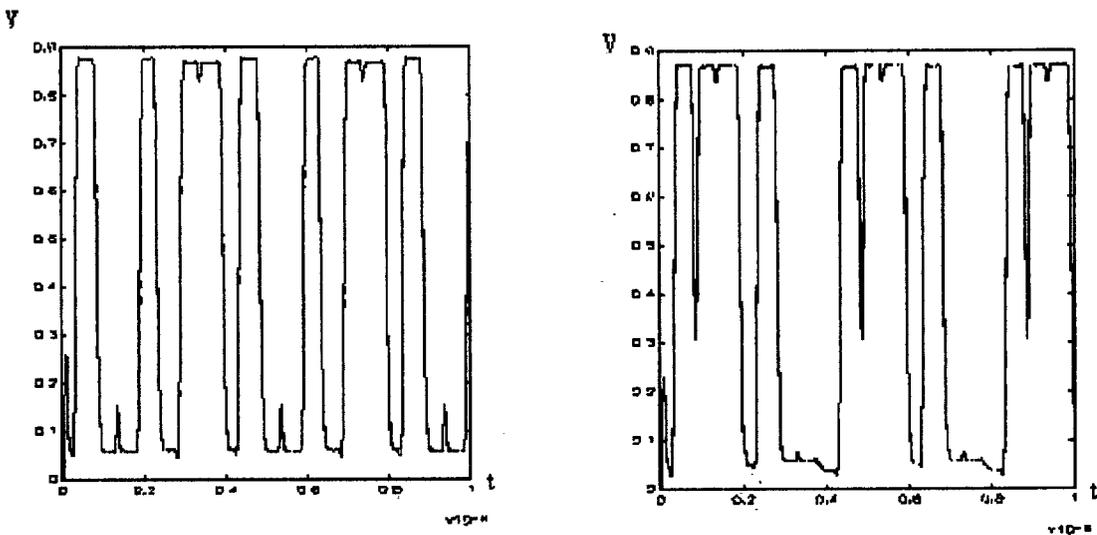


fig. 6.9 Señales de entrada Si (línea continua), Ci (línea de puntos) y Ci-1 (línea discontinua).



(a)

(b)

fig. 6.10 Señales de salida (a) Sout y (b) Cout. IAFSPICE: línea continua, APLAC: línea de puntos.

### 6.1.3 Análisis en ac

Por último, el circuito utilizado para comprobar el análisis en ac es el que se muestra en la figura 6.11. El fichero de entrada correspondiente a este circuito se presenta a continuación:

---

#### Códigos

---

```

INCLUDE "C:\PROYECTO\EFET.I"

EFET H1 2 3 0 WW=25

VOLT VDD 1 0 DC=3
RES RD 1 2 1K
SHORT VMEAS 2 4 I=I0
CAP CSUNT 4 0 0.1E-6
VOLT VGS 0 3 AC=1

SWEEP "ANALISIS DEL HFET EN ALTERNA"
+ AC
+ LOOP 20 FREQ LIN 0.05G 22.5G
+ GRID
  DISPLAY
+ WINDOW 0 RI "I0" IAC(I0)
+ WINDOW 1 RI "I0" IAC(I0)
+ WINDOW 2 DB "I0" IAC(I0)
+ WINDOW 3 MA "I0" IAC(I0)

ENDSWEEP

```

---

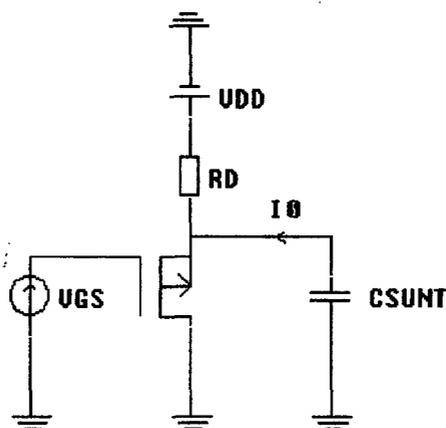


fig. 6.11 Circuito para comprobar el análisis en ac.

Consiste básicamente en un transistor polarizado en el que se realiza un análisis en ac. La señal de salida es la corriente que circula por el amperímetro I0, la cual va a ser un número complejo por la propia definición de los análisis en

ac. Por tanto, las posibles medidas que podemos obtener son la parte real y la parte imaginaria de la corriente (figura 6.12), su magnitud y su fase (figura 6.13) y su magnitud en decibelios (figura 6.14).

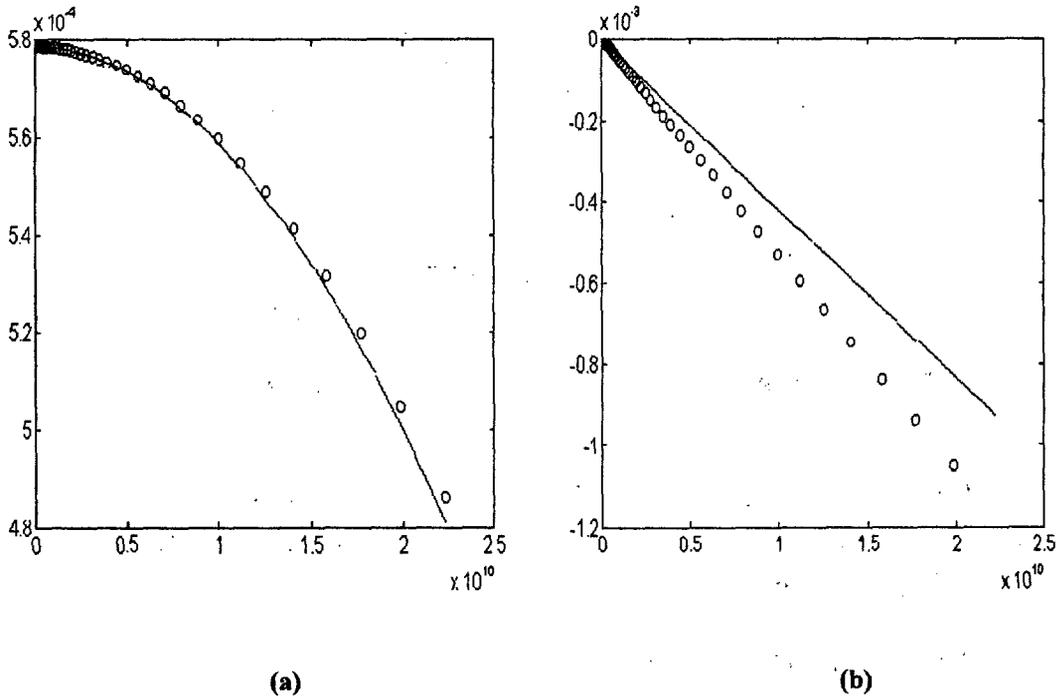


fig. 6.12 Parte real (a) y parte imaginaria (b) de la corriente  $I(I_0)$  del circuito de la figura 6.11.

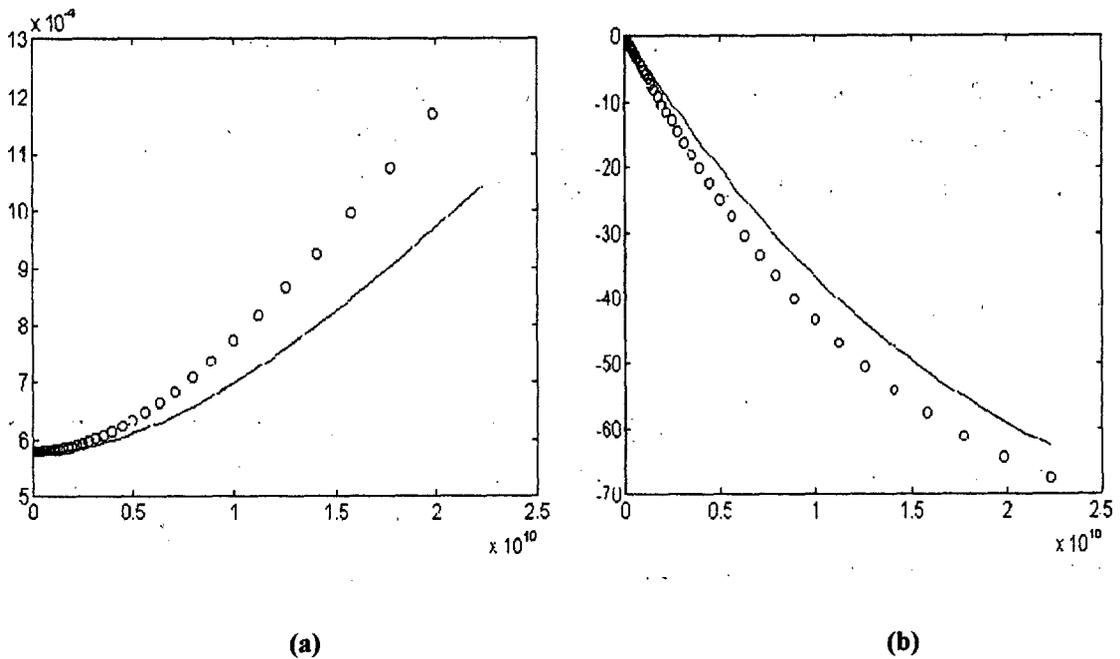


fig. 6.13 Magnitud (a) y fase (b) de la corriente  $I(I_0)$  del circuito de la figura 6.11.

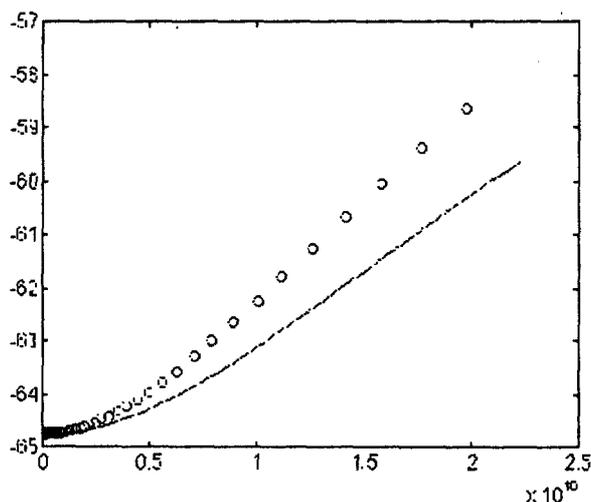


fig. 6.14 Magnitud de la corriente  $I(I_0)$  del circuito de la figura 6.11.

Se observa que hay una diferencia entre los resultados de uno y otro simulador, esto es debido a que no hemos podido introducir la transconductancia modificada  $\hat{g}_m$ , la cual es:

$$\hat{g}_m = g_m \cdot e^{-j\Omega\tau} = g_m \cdot (\cos\Omega\tau - j\text{sen}\Omega\tau), \text{ con } \Omega = 2 \cdot \pi \cdot \text{frec} \quad (4.44)$$

donde  $g_m$  es la transconductancia y  $\tau$  es un exceso de fase similar al parámetro  $ptf$  de los transistores bipolares. En APLAC, en las fuentes VCCS no lineales no se permite introducir transconductancias complejas ni utilizar el parámetro opcional de la fuente "Y". Este parámetro que permite transconductancias complejas en ac y en análisis armónicos que solo se permiten en fuentes lineales. Se hace referencia a este hecho en el capítulo 4 apartado (4.2.1.4).

## 6.2 Conclusiones finales

Además de lo ya reseñado y probado respecto a la validez del trabajo realizado queremos indicar que con este trabajo el HFET no queda completamente modelado. Falta incorporar los análisis de temperatura, polos y ceros, ruido, distorsión, sensibilidad, etc.

En cuanto al análisis de temperatura, decir que actualmente no existe un modelo fiable de su influencia y, por tanto, no se puede implementar. Para llevar a cabo esta tarea, lo primero que deberíamos hacer es fabricar un transistor HFET y obtener sus parámetros de modelo. Posiblemente tendríamos que modificar el

modelo de IAFSPICE puesto que dicho modelo, al ser semiempírico, depende de la tecnología utilizada y, por tanto, puede no servirnos para nuestro transistor. Sin embargo, el modelo resultante sería siempre muy parecido al de Angelov. Una vez hecho esto se estudiaría el efecto de la temperatura sobre el comportamiento eléctrico de nuestro HFET haciéndose un modelo e implementándolo dentro de nuestro simulador.

En cuanto al simulador eléctrico APLAC, es una herramienta muy versátil y potente en la creación de nuevos modelos aunque el tratamiento de las conductancias complejas se tendría que cambiar para poder ser introducidas en fuentes no lineales que es el principal punto débil que nos hemos encontrado del simulador

## Anexos

Lista de los parámetros de los HFETs de enriquecimiento y depleción de 0.5 y 0.3 $\mu$ m.

*Transistores de 0.5 $\mu$ m:*

### EFET

GEXP=0.075	AFACT=4	VC=0.65	VSB=5	LAMBDA=0.05	RD=600	RS=600	RG=0.2	MFACT=1
IS=1E-14	DXL=10	VS=0.4	DELTA=0	CDVC=0.23E-3	NP=1.5	BETA=2.8	FC=1.5	CDVSB=0
TD=3E-12	VAT=-0.3	RGS=800	RGD=800	CGS0=2.5E-16	VST=1	VBT=0.3	CDS=2E-16	CGS1=9E-16
ALFA=5	GAMMA=4							

### DFET

GEXP=0.075	AFACT=5	VC=0.05	VSB=0.78	LAMBDA=0.09	RD=500	RS=500	RG=0.2	MFACT=1
IS=1E-14	DXL=3	VS=0.4	DELTA=1.9	CDVC=0.145E-3	NP=1.35	BETA=2.3	FC=0.42	CDVSB=0.12E-3
TD=3E-12	VAT=-0.9	RGS=900	RGD=900	CGS0=2.3E-16	VST=1	VBT=-0.4	CDS=2E-16	CGS1=6.5E-16
ALFA=3.1	GAMMA=2.5							

*Transistores de 0.3 $\mu$ m:*

### EFET

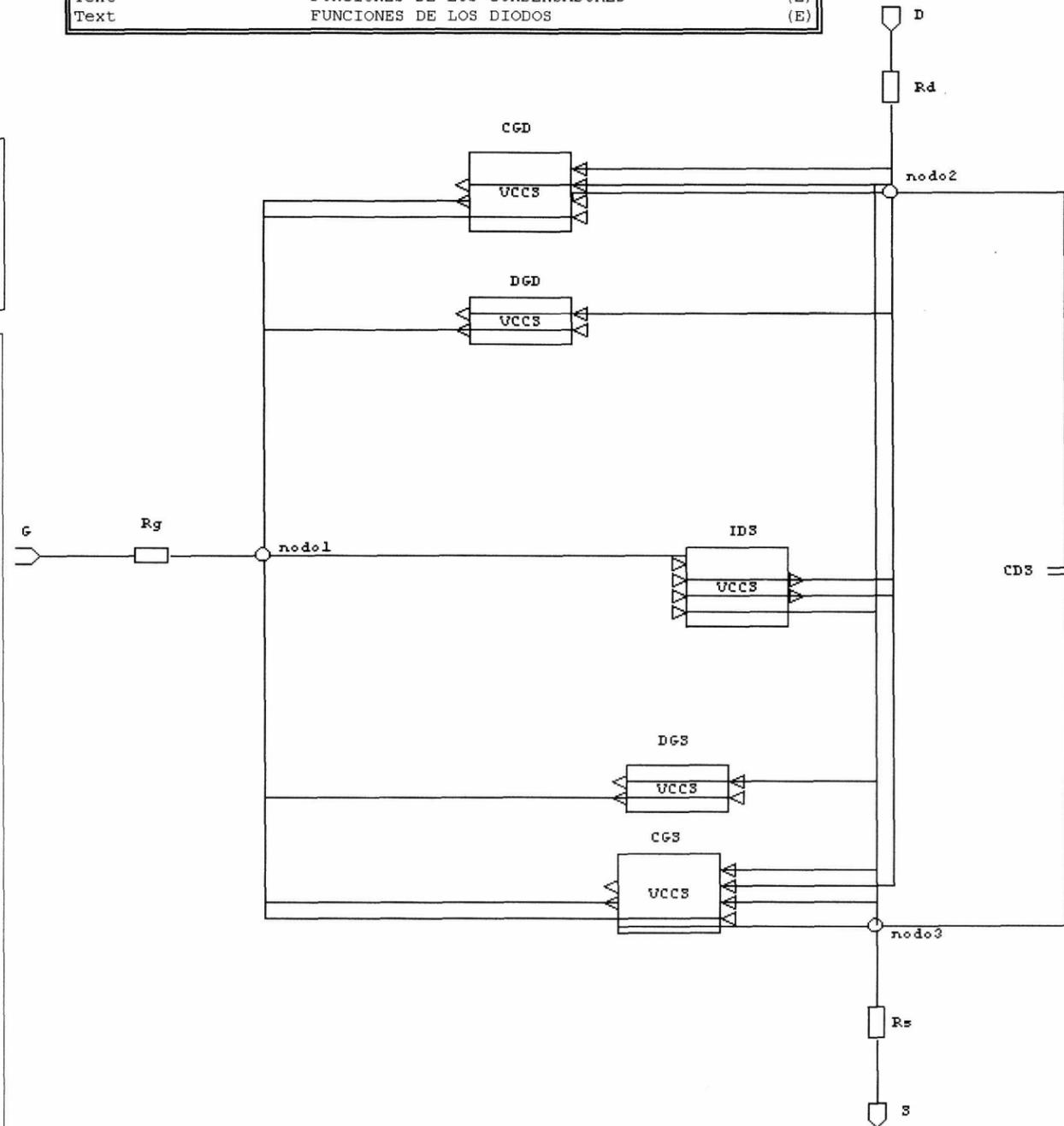
GEXP=0.075	AFACT=4	VC=0.55	VSB=5	LAMBDA=0.05	RD=600	RS=600	RG=0.3	MFACT=1
IS=1E-13	DXL=10	VS=0.4	DELTA=0	CDVC=0.25E-3	NP=1.5	BETA=2.8	FC=1	CDVSB=0
TD=2E-12	VAT=-0.4	RGS=800	RGD=800	CGS0=2E-16	VST=1	VBT=0.15	CDS=2E-16	CGS1=6E-16
ALFA=5	GAMMA=4							

**DEFET**

GEXP=0.075	AFACT=5	VC=-0.1	VSB=0.78	LAMBDA=0.15	RD=500	RS=500	RG=0.3	MFACT=2
IS=1E-14	DXL=3	VS=0.4	DELTA=1.9	CDVC=0.16E-3	NP=1.35	BETA=2.35	FC=0.7	CDVSB=0.12E-3
TD=2E-12	VAT=-1.1	RGS=900	RGD=900	CGS0=1.8E-16	VST=1	VBT=-0.4	CDS=2E-16	CGS1=4E-16
ALFA=3.5	GAMMA=2							

Circuito del Efet en el editor de APLAC :

DefModel	Efet	(E)
Text	PARAMETROS OBLIGATORIOS	(E)
Text	PARAMETROS AFECTADOS POR W	(E)
Text	FUNCIONES DE LA FUENTE (E)	(E)
Text	FUNCIONES DE LOS CONDENSADORES	(E)
Text	FUNCIONES DE LOS DIODOS	(E)



```

IF NOT(PARAMGIVEN(ANALISIS)) THEN
    CALL EXIT ("SE DEBE ESPECIFICAR EL TIPO DE ANALISIS QUE SE VA A REALIZAR::DC,TRAN=0, AC=1")
ENDIF
IF (ANALISIS !=0 AND ANALISIS !=1) THEN
    CALL EXIT ("ANALISIS TIENE QUE SER PARA DC,TRAN=0 O AC=1 NO SE PERMITE OTRO VALOR")
ENDIF
IF NOT(PARAMGIVEN(W)) THEN
    CALL EXIT ("SE DEBE ESPECIFICAR W ")
ENDIF
    
```

```

*FUNCIONES DE LOS CONDENSADORES
*EN ESTE CASO TOMAREMOS CV(0) Y CV(1) PARA EVITAR LA CONFUSIÓN ENTRE VGS Y VGD
*PARA VAT<CV(0)<=VBT
FUNCTION FC1I CGS1W*(CV(0)-VAT)/(VBT-VAT)
FUNCTION FC2IM 1+VST*TANH(ALPHA*VS*CV(1))
*PARA VGS>=VBT
FUNCTION FC1M (LOCAL)
+ LOCAL=FC*((CV(0)-VBT)^MFACT);
+ CGS1W*(1+LOCAL)
*EXPRESION GENERAL DE FC1
FUNCTION FC1= IF (CV(0)<VAT) THEN 0 ELSE IF (CV(0)>=VAT AND CV(0)<VBT) THEN FC1I ELSE FC1M
+
*EXPRESION GENERAL DE FC2
FUNCTION FC2=IF (CV(0)<VAT) THEN 0 ELSE FC2IM
*EXPRESION GENERAL DE FC2
FUNCTION CG=CGSOW+FC1*FC2
*FUNCIONES DE LAS CARGAS
APLACVAR QC2
CALL QC2=QC1
FUNCTION QC3=QC2-CGS1W*FC2*(VBT+VAT)/2
*PARA VGS<VAT
FUNCTION QGHE CGSOW*CV(0)+QC1
*PARA VAT<=VGS<VBT
FUNCTION QGI (QGI1,QGI2)
+ QGI1= SQR(CV(0)-VAT);
+ QGI2= QGI1/(2*(VBT-VAT));
+ CGSOW*CV(0)+CGS1W*QGI2*FC2+QC2
*PARA VGS>=VAT
FUNCTION QGM (QGM1,QGM2)
+ QGM1= (CV(0)-VBT)^(MFACT+1);
+ QGM2= CV(0)+FC*QGM1/(MFACT+1);
+ CGSOW*CV(0)+CGS1W*QGM2*FC2+QC3
FUNCTION QG=IF CV(0)<VAT THEN QGHE ELSE IF (CV(0)>=VAT AND CV(0)<VBT) THEN QGI ELSE QGM
+
FUNCTION QGP=IF (ANALISIS==1) THEN -QG
+ ELSE QG
FUNCTION QCP=IF (ANALISIS==1) THEN -CG
+ ELSE CG
    
```

```

* FUNCIONES DE LA FUENTE
FUNCTIONVGS=CV(0)
FUNCTION VDS=CV(1)
FUNCTIONF1 ( F11, F12 )
+ F11= CDVCW*(1+TANH(BETA*(VGS-VC)+GAMMA*(VGS-VC)^3));
+ F12= CDVSEW*(1+TANH(DELTA*(VGS-VSB)));
+ (F11+F12)
FUNCTIONVTO=(VC-2/BETA)
FUNCTION F2 ( F21 )
+ F21= (1+DEL*(SQR(VGS-VTO)));
+ (1+LAMBDA*VDS/F21)
FUNCTION F3 TANH(ALPHA*VDS)
*CONDUCTANCIA DE LA FUENTE DEPENDIENTE
FUNCTION GM ( GM1, GM2, GM3 )
+ GM1= CDVCW*((BETA+GAMMA^3*(SQR(VGS-VC)))/(SQR(COSH(BETA*(VGS-VC)+GAMMA*(VGS-VC)^3))));
+ GM2= CDVSEW*(DELTA/(SQR(COSH(DELTA*(VGS-VSB))));
+ GM3= ( VDS*LAMBDA*DEL^2*(VGS-VTO)/(SQR(1+DEL*(SQR(VGS-VTO))));
+ F3*( F2*( GM1+GM2)-F1*GM3 )
FUNCTIONGDS (GD81,GD82,GD83)
+ GD81= LAMBDA/(1+DEL*(SQR(CV(0)-VTO)));
+ GD82= 1+LAMBDA*CV(1)/(1+DEL*(SQR(CV(0)-VTO)));
+ GD83= ALPHA/(SQR(COSH(ALPHA*CV(1))));
+ F1*(gd81*f3+gd82*gd83)
*PUENTE DEPENDIENTE DE EFET
FUNCTION IDS=F1*F2*F3
    
```

```

*FUNCIONES DEL DIODO
FUNCTION VDD=CV(0)
FUNCTION IGP ( EXPO )
+ EXPO=EXP(Q*VDD/(N*K*TEMP));
+ ISOW*(EXPO-1)
FUNCTION IGM ( EXPO )
+ EXPO=EXP((Q*SQR(ABS(GEXP*VDD)))/(N*K*TEMP));
+ (-ISOW*AFACT*(EXPO-1))
FUNCTION GDP ( EXPO )
+ EXPO= EXP(Q*VDD/(N*K*TEMP));
+ (ISOW*Q*EXPO)/(N*K*TEMP)
FUNCTION GDN ( EXPO )
+ EXPO=EXP((Q*SQR(ABS(GEXP*VDD)))/(N*K*TEMP));
+ (ISOW*AFACT*Q*GEXP*EXPO)/(2*N*K*TEMP*SQR(ABS(GEXP*VDD)))
FUNCTION GD
+ELSE GDV IF (VDD>=0) THEN GDP
FUNCTION IG
+ELSE IGM IF (VDD>=0) THEN IGP
    
```

## Referencias Bibliográficas

---

---

[1] APLAC™ 7.5. User's Guide Vol. 1. Working with APLAC. Contains a quick tour of APLAC and complete information on using all its graphical interface commands and features. (1999).

[2] APLAC™ 7.5. User's Guide Vol. 2. APLAC Language. It contains a tutorial for the beginner, and many complete simulation examples, giving valuable information to the new as well as the experienced user. Also included here is a limited collection of frequently asked questions with answers related to APLAC in general. (1999).

[3] APLAC™ 7.5. APLAC Reference Manual Vol. 1. Programming, Analysis, and Optimization. Contains the syntax for analysis and programming statements, and for all the functions available in the APLAC Language. (1999).

[4] APLAC™ 7.5. APLAC Reference Manual Vol. 2. Analog components. Documents the component models available in the core version of APLAC. This includes passive lumped and active components, sources and the general transmission line. (1999).

[5] APLAC™ 7.5. Reference Manual Vol. 3. Passive RF components. Describes models for passive transmission line structures and the switched-capacitor switch available to the user with the RF option. (1999).

[6] APLAC™ 7.5. System Simulation Manual Vol. 1. Formula-based simulations. Tells how formula-based simulation is performed, how system blocks must be defined for simulating at a system level, and gives the syntax for the available system component model statements. (1999).

- [7] APLAC™ 7.5. System Simulation Manual Vol. 2 . Discrete-time simulations. Describes the discrete-time system analyzer, the component models available and their use. (1999).
- [8] Electromagnetic Simulation Manual FDTD simulation. Discusses the EM option, the building blocks (component models) and analysis. (1999).
- [9] Lombu90. Stephen L. Long, Steven E. Butner, "Galium Arsenide Digital Circuit Design", McGraw-Hill. 1990.
- [10] Curti80. W. Curtice, "A MESFET model for use in the design of GaAs integrated circuits", IEEE Tran. Microwave Theory Tech., MTT-28: 448-455, 1980.
- [11] Curet85. W. Curtice y M. Ettenberg, "A nonlinear GaAs FET model for use in the design of output circuits for power amplifiers", IEEE Trans. Microwave Theory Tech., MTT-33: 1383, 1985.
- [12] Matka85. A. Materka y T.Kacprzak,"Computer calculation of large-signal GaAs FET amplifier characteristics", IEEE Trans. Microwave Theory Tech., MTT-33: 129-134, 1985.
- [13] Angel92. I. Angelov et al., "A new empirical nonlinear model for HEMT and MESFET devices", IEEE Trans. Microwave Theory Tech., MTT-40: 2258-2266, 1992.
- [14] Paul C. R. "Analysis of Linear Circuits", New York: McGraw-Hill, 1989.
- [15] Chen T. Wang. Devices Research Institute. "Introduction To Semiconductor Technology: GaAs and Related Compounds". New York. Chichester. Brisbane. Toronto. Singapore: A Wiley-Interscience Publication John Wiley&Sons. 1989
- [16] J. M. Ruiz, G. L. Araujo, G. Sala. "Física de los Dispositivos Electrónicos Volumen I" Departamento de Publicaciones E.T.S Ingenieros de Telecomunicación.1982.

[17] J. M. Ruiz, G. L. Araujo, G. Sala. "Física de los Dispositivos Electrónicos Volumen II" Departamento de Publicaciones E.T.S Ingenieros de Telecomunicación.1982.

[18] L. W Ángel, y R. Rohrer, " Computer Análisis of nonlinear circuits", IEEE Journal of Solid-State Circuits, SC-6:166-182, Agosto 1971.

# Pliego de condiciones

---

---

Los requisitos necesarios para el correcto funcionamiento del software diseñado es el siguiente:

- Estación de trabajo con sistema operativo SOLARIS Ver. 2.6
- Programa APLAC Ver. 2.3.0
- Programa MATLAB.
- Licencia del APLAC y del entorno de desarrollo.

## Protección legal

### Propiedad intelectual

La propiedad intelectual, que comprende las creaciones literarias, científicas, artísticas, programas de ordenador, etc., se diferencia de la industrial en que aquella el derecho surge por la creación, sin ser necesario acudir a ningún registro para el nacimiento del derecho. A pesar de que no es preciso el registro para poseer el derecho, la Ley regula la existencia de un Registro Intelectual en el que inscribir este tipo de creaciones.

La propiedad intelectual se regula en España por el Real decreto Legislativo 1/1996, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia. La Ley de Propiedad Intelectual de 11 de

Noviembre de 1987 (BOE de 7/11/1987), y en el artículo 10 recoge las materias de que es objeto:

1. Son objeto de propiedad intelectual todas las creaciones originales literarias, artísticas o científicas expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro, comprendiéndose entre ellas:
  - a) Los libros, folletos, impresos, epistolarios, escritos, discursos y alocuciones, conferencias, informes forenses, explicaciones de cátedra y cualesquiera otras obras de la misma naturaleza.
  - b) Las composiciones musicales con o sin letra.
  - c) Las obras dramáticas y dramático musicales, las coreografías, las pantomimas y, en general, las obras teatrales.
  - d) Las obras cinematográficas y cualesquiera otras obras audiovisuales.
  - e) Las esculturas y las obras de pintura, dibujo, grabado, litografía, y las historietas gráficas, tebeos o cómics, así como sus ensayos o bocetos y las demás obras plásticas, sean o no aplicadas.
  - f) Los proyectos, planos, maquetas y diseños de obras arquitectónicas y de ingeniería.
  - g) Los gráficos, mapas y diseños relativos a la topografía, la geografía y, en general, a la ciencia.
  - h) Las obras fotográficas y las expresadas por procedimiento análogo a la fotografía.
  - i) Los programas de ordenador.
2. El título de una obra, cuando sea original, quedará protegido como parte de ella.

### **Titularidad del proyecto**

Sobre este tema no existe hasta el momento una normativa a la que atenerse, por lo tanto, tenemos que ajustarnos a alguna publicación específica sobre el tema, cuyo contenido indique, o del que pueda inferirse la titularidad del proyecto. En concreto, en el capítulo 7 de “La política y la gestión de la propiedad industrial en un centro público de investigación” de Ignacio Fernández de Lucio y Domingo Represa Sánchez, aparecen una serie de documentaciones, que sin referirse concretamente a la situación que nos ocupa (proyecto fin de

carrera), si podemos extrapolar de alguna manera el estado legal del mismo, en cuanto a su titularidad, y participación de su explotación, si la hubiera.

En este sentido, podemos destacar de dicha obra lo siguiente:

### **“Titularidad de los resultados”**

En la actualidad, teniendo en cuenta la complejidad y los costes de las investigaciones, la mayor parte de las invenciones se llevan a cabo en el seno de las empresas, universidades u OPIS; ya casi no existe el inventor solitario que, a través de su ingenio y por su cuenta, tiene una idea genial y la plasma en una invención. Esto hace que se tenga que regular a quién y en que condiciones pertenecen los resultados de la investigación realizada por los trabajadores en las empresas. En los casos en que la titularidad pertenezca a la empresa, será sin perjuicio del derecho que tiene el trabajador o trabajadores a aparecer como inventores de la misma (Véase el artículo 14 de la Ley de Patentes y el artículo 4º. Ter. del Acta de Estocolmo de 14 de julio de 1967 modificativa del Convenio de París para la Protección de la Propiedad Industrial, publicado en el BOE de 1 de febrero de 1974) y los artículos 14 a 20 de la Ley española de patentes, Ley 11, de 20 de Marzo de 1986, de patentes, J.M. Oter Lastres et al (1987).

### **Invencciones universitarias y de organismos públicos de investigación.**

El artículo 20 de la Ley de Patentes regula esta materia y remite a los Estatutos de las Universidades el desarrollo de la misma (más información en M. Pérez de 1984).

El artículo 20.1 dice:

*“Las normas del presente Título serán aplicables a los funcionarios y trabajadores del Estado, Comunidades Autónomas, Provincias, Municipios y demás Entes Públicos sin perjuicio de los previstos en los párrafos siguientes”.*

Los párrafos siguientes regulan esta materia para las invenciones realizadas por profesores e investigadores de la Universidad, así como su posible aplicación a los investigadores de OPIS.

La titularidad de las invenciones realizadas por los profesores pertenece a la Universidad, si esta invención es producto de la investigación que realiza dentro de su función o actividad de profesor, permitiendo, sin embargo, el Art. 11

de LRU sobre contratación con terceros, que se determine en el contrato la titularidad de los resultados, A. Bercovitz (1986).

El artículo 20, en su apartado 2, atribuye a la Universidad la titularidad de las patentes creadas como consecuencia de la función de investigación, sin perjuicio del artículo 14 de la Propia Ley en el que se dice:

*El inventor tiene frente al titular de la solicitud de la patente el derecho a ser mencionado como tal inventor en la patente.*

La titularidad económica corresponde a la Universidad aunque, como se verá más adelante, el profesor tenga derecho a una participación en los beneficios. También, pertenece al profesor el derecho moral a aparecer como creador, considerándose éste un derecho personalísimo al que no le pueden obligar a renunciar.

La participación de los profesores en los beneficios de las invenciones se recoge en el apartado 4 del mismo artículo, que regula el derecho de los inventores a participar en los beneficios que obtenga la universidad con estas invenciones.

La regulación de esta materia se tiene que realizar en los distintos estatutos de las universidades, como se señala en el mencionado artículo 20 de la ley de Patentes. En la práctica pocos son los Estatutos de universidad que recogen esta regulación y si lo hacen es de forma incompleta ya que, al ser en muchos casos posteriores a la Ley de Patentes, no contemplan todas las situaciones previstas en ella, a lo que cabe añadir la dificultad que supone la modificación de los Estatutos....

## *Presupuesto*

---

---

El cálculo del presupuesto de este proyecto se ha seguido según la “Propuesta de baremos orientativos para el cálculo de honorarios “ establecida por el Colegio Oficial de ingenieros Técnicos de Telecomunicación a partir del 1-01-2001.

Esta propuesta establece que para “Trabajos tarifados por tiempo empleado” se aplique la siguiente ecuación:

$$H = Hn * 9700 + He * 10500$$

Siendo:

H= Honorarios a percibir.

Hn= Horas contabilizadas en jornada normal.

He= Horas contabilizadas fuera de la jornada normal de trabajo.

Los honorarios que se obtengan por aplicación de la clave “H” se reducirán a medida que aumente el número de horas, a cuyo efecto serán multiplicados por los coeficientes reductores con arreglo a la siguiente escala

			COEFICIENTE
Hasta	36 horas		C=1
Exceso de	36 horas	Hasta 72 horas	C=0,9
Exceso de	72 horas	Hasta 108 horas	C=0,8
Exceso de	108 horas	Hasta 144 horas	C=0,7
Exceso de	144 horas	Hasta 180 horas	C=0,65
Exceso de	180 horas	Hasta 360 horas	C=0,60
Exceso de	360 horas	Hasta 510 horas	C=0,55
Exceso de	510 horas	Hasta 720 horas	C=0,50
Exceso de	720 horas	Hasta 1080 horas	C=0,45
Exceso de	1080 horas		C=0,40

## CÁLCULO DEL PRESUPUESTO

### • COSTES DEBIDOS A LOS RECURSOS HUMANOS

En este apartado se incluyen los honorarios a percibir por el ingeniero técnico en el desarrollo del proyecto en función de las horas de trabajo que se ha empleado en la realización del mismo.

Particularizando para el proyecto que aquí se dispone, establecemos una tabla indicativa acerca del tiempo parcial empleado en cada fase del mismo.

DESCRIPCIÓN	TIEMPO PARCIAL
Búsqueda y estudio de la documentación	390 horas
Estudio del sistema de simulación APLAC	490 horas
Diseño y simulaciones del cabezal	600 horas
Realización de la memoria	530 horas

En definitiva, se necesitaron un total de 2010 horas para la realización de este proyecto. Considerando estas horas como jornada laboral normal para su tarificación se obtienen los siguientes honorarios:

$$H = 2010 * 9700 = 19.497.000 \text{ ptas. } (117.451,80\text{€})$$

Aplicando el coeficiente de corrección correspondiente tenemos:

$$H = 19.497.000 * 0.4 = 7.798.800 \text{ ptas. (46.980,72€)}$$

- COSTES DE AMORTIZACIÓN DE LOS EQUIPOS INFORMÁTICOS Y HERRAMIENTAS SOFTWARE***

A continuación, se detallan los costes relacionados a la utilización de los equipos y herramientas software empleados en la elaboración del presente proyecto. Los costes están divididos entre el número de usuarios que acceden a ellos los cuales se han estimado en un número de 50.

Costes debidos a la utilización de herramientas software					
Descripción	Tiempo de uso	Coste anual (ptas)		Total	
		Total	Usuario	Pesetas	Euros
Sistema operativo SunOs Release 4.1.3, Openwindows y aplicaciones X11	7 meses	150.300	3.006	1.753	10,53
Entorno de diseño y simulación APLAC	7 meses	166.386	3.327	1.940	11,66
Entorno Windows NT	7 meses	50.950	1.019	594	3,57
Microsoft Office 97	7 meses	74.700	1.494	871	5,23
<b>COSTES DE HERRAMIENTAS SOFTWARE</b>		<b>TOTAL</b>		<b>5.158</b>	<b>31</b>

Costes debidos a la utilización de equipos informáticos					
Descripción	Tiempo de uso	Coste anual (ptas)		Total	
		Total	Usuario	Pesetas	Euros
Estación de trabajo SUN Sparc modelo Sparc Station 10					
Amortización 3 años	7 meses	870.000	17.400	10.150	61
Mantenimiento	7 meses	262.000	5.240	3.056	18,36
Servidor para simulación SUN Sparc Station 10					
Amortización 3 años	7 meses	843.333	16.867	9839	59,13
Mantenimiento	7 meses	262.000	5.240	3.056	18,36
Impresora Hewlett Packard Laserjet 4L					
Amortización 3 años	7 meses	25.000	500	291	1,74
Mantenimiento	7 meses	24.212	484	282	1,69
Ordenador Personal Pentium II 266 Mhz					
Amortización 3 años	7 meses	53.333	1.067	622	3,73
Mantenimiento	7 meses	15.000	300	175	1,05
<b>COSTES DE EQUIPOS INFORMÁTICOS</b>		<b>TOTAL</b>		<b>27.471</b>	<b>165,06</b>

- *OTROS COSTES*

En este apartado se incluyen los costes debidos al uso de Internet, material fungible y a la elaboración del documento final.

Otros costes				
Descripción	Nº de unidades	Coste unidad (ptas)	Total	
			Pesetas	Euros
Horas de uso de internet	200 horas	189 ptas/hora	37.800	226,8
Paquetes papel DIN-A4 80 gr/m <sup>2</sup>	3	750	2.250	13,50
Fotocopias	1.000	5	5.000	30,01
Encuadernación	3	4.500	13.500	81,03
<b>OTROS COSTES</b>	<b>TOTAL</b>		<b>58.550</b>	<b>351,34</b>

● **PRESUPUESTO TOTAL**

Costes	Total	
	Pesetas	Euros
Costes de herramientas software	5.158	31
Costes de equipos informáticos	27.471	165,06
Costes de recursos humanos	7.798.800	46.980,72
Otros costes	58.550	351,34
<b>PRESUPUESTO FINAL</b>	<b>7.889.979</b>	<b>47.529,99</b>

Por lo tanto, el precio de coste global del proyecto asciende a SIETE MILLONES OCHOCIENTAS OCHENTA Y NUEVE MIL NOVECIENTAS SETENTA Y NUEVE PESETAS (7.889.979 ptas.) o CUARENTA Y SIETE MIL QUINIENTAS VEINTE Y NUEVE EUROS CON NOVENTA Y NUEVE CÉNTIMOS (26.664,98 euros.).