

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIÓN



TESIS DOCTORAL

**OPTIMIZACIÓN MULTIOBJETIVO DE CIRCUITOS
DIGITALES DCFL/SDCFL EN GaAs**

LUIS GÓMEZ DÉNIZ

Las Palmas de Gran Canaria, 1992



09-1992/93

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
UNIDAD DE TERCER CICLO Y POSTGRADO

Reunido el día de la fecha, el Tribunal nombrado por el Excmo. Sr. Rector Magfco. de esta Universidad, el aspirante expuso esta TESIS DOCTORAL.

Terminada la lectura y contestadas por el Doctorando las objeciones formuladas por los señores jueces del Tribunal, éste calificó dicho trabajo con la nota de **APTO. CUM LAUDE POR UNANIMIDAD**

Las Palmas de G. C., a 18 de Diciembre de 1992

El Presidente: Dr. D. Francisco Rubio Royo, *Rubio Royo*

El Secretario: Dr. D. Roberto Sarmento Rguez., *Sarmento*

El Vocal: Dr. D. Salvador Bracho del Pino, *Bracho del Pino*

El Vocal: Dr. D. Octavio Nieto-Taladriz García, *Nieto-Taladriz*

Vocal: Dr. D. Blas-Pablo Dorta Naranjo, *Dorta Naranjo*

El Doctorando: D. Luis Gómez Depiz, *Gómez Depiz*



BIBLIOTECA UNIVERSITARIA
LAS PALMAS DE G. CANARIA
Nº Documento..... 342.016
Nº Copia..... 342.022

UNIVERSIDAD DE
LAS PALMAS DE GRAN CANARIA
E. T. S. I. TELECOMUNICACIÓN



TESIS DOCTORAL

Optimización multiobjetivo de circuitos
digitales DCFL/SDCFL en GaAs

Autor: D. Luis Gómez Déniz

Director: Dr. D. Antonio Núñez Ordóñez

Departamento de Electrónica y Telecomunicación

Diciembre 1992

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
DOCTORADO EN INGENIERÍA DE TELECOMUNICACIÓN
DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIÓN

Optimización Multiobjetivo de circuitos DCFL/SDCFL en GaAs

Tesis Doctoral presentada por D. Luis Gómez Déniz
Dirigida por el Dr. D. Antonio Núñez Ordóñez

El Director,

El Doctorando,



Las Palmas de Gran Canaria a 18 de Diciembre de 1992

RESUMEN

La tecnología de Arseniuro de Galio (GaAs) ha experimentado en los últimos años un gran avance. Dicha tecnología ofrece importantes ventajas para el diseño digital de alta velocidad. El diseño en GaAs -dadas las propiedades del material y su costo- es un diseño orientado a obtener altas prestaciones. De no ser así, no se entiende su aplicación. A diferencia del diseño tradicional en tecnología de Silicio, resulta imprescindible la estimación inicial de ciertas funciones de coste de cara a tener una visión preliminar del sistema a implementar.

El objetivo de todo proceso de diseño consiste en pasar de unas especificaciones iniciales, definidas en muy alto nivel, a su correspondiente realización física (*layout*). En el ámbito GaAs, el número de factores que afectan al diseño es mucho mayor y el estudio de viabilidad muestra mayor complejidad. De especial relevancia es la consideración del ruido (tanto dinámico como estático) durante el diseño de circuitos GaAs. Es ésta una circunstancia que, a diferencia de lo que acontece en circuitos CMOS, se ha tenido en cuenta ya desde los primeros diseños GaAs.

Tal como ocurrió hace años con los circuitos digitales de Silicio, los niveles de integración alcanzados en circuitos GaAs basados en MESFETs han superado actualmente la llamada frontera *VLSI*. Los diseñadores demandan cada vez más herramientas (o estrategias) capaces de optimizar diseños de forma eficiente y rápida. Un importante paso en el diseño de un circuito GaAs digital es la optimización del consumo de potencia mientras se mantiene el consumo de área y la velocidad del circuito dentro de unos límites razonables.

El objetivo del presente trabajo es desarrollar una estrategia de optimización para circuitos *full-custom* combinacionales GaAs con la intención de llenar el vacío existente actual. La optimización se encauza principalmente hacia la mejora del compromiso definido por el retardo de propagación, el consumo de potencia y la superficie ocupada.

La implementación de dicha estrategia requiere disponer de modelos analíticos seguros y de mínimo coste computacional. Para la estimación de retardos se procederá a la adaptación del modelo estimativo previamente desarrollado. En esta tesis se desarrolla además un modelo analítico completo para la estimación del consumo de potencia y del área-*chip* ocupada. Los tres modelos con los que se edifica la estrategia de optimización son tecnológicamente independientes, rápidos de computar y muestran una máxima desviación respecto a HSPICE del 10%.

La estrategia de optimización que se plantea se desarrolla en dos fases claramente diferenciadas. La primera de ellas explora las distintas posibilidades de "bufferización" para los circuitos bajo estudio. La segunda fase estudia las posibilidades que encierra el dimensionamiento de los transistores. El problema del dimensionado (*sizing*) se resuelve mediante técnicas de programación no lineal.

Se define una técnica de *buffer* sistemático que denominamos metodología robusta. Se demuestra que con ella pueden diseñarse circuitos DCFL de forma rápida, con absoluta fiabilidad y con buenas prestaciones. El empleo de esta metodología nace como una necesidad de cara a poder construir diseños complejos obviando el problema asociado al deterioro de los márgenes de ruido. De esta forma queda afianzado el diseño según la metodología robusta. Para esta metodología se definen las geometrías óptimas a emplear tanto para el seguidor

de fuente como para los transistores de las distintas etapas lógicas.

La interpretación de los resultados manifiesta que el diseño robusto tiene limitada la velocidad de conmutación. Esto no es consecuencia de que se haya impuesto una cota al consumo de potencia o al de área. Por ello se desarrolla la metodología de diseño adaptativo de cara a proporcionar una mayor flexibilidad a los diseños.

El interés de esta metodología adaptativa viene dado porque permite la implementación de diseños más rápidos (consumiendo más potencia). Asimismo el empleo de la técnica adaptativa permite reducir el consumo de potencia (ralentizando el circuito). El diseño adaptativo brinda al diseñador la posibilidad de *adaptar* las características del circuito a los requerimientos deseados (entiéndase como deseados posibles), moviéndose en un espacio de diseño más amplio que ofrece mayores márgenes a la optimización.

La diferencia radical entre ambas estrategias se traduce en comodidad y rapidez en el diseño. Así, un diseño de tipo adaptativo requerirá una realización más cuidadosa demandando un mayor tiempo de desarrollo. Dado que la celeridad en la terminación de un circuito GaAs constituye en gran número de casos un factor de primer orden, no extraña que el diseñador se decante por el diseño robusto, al menos para las primeras versiones o series de circuitos. Sin embargo, el diseño adaptativo goza siempre de una mejora sustancial en sus prestaciones y a él debe tenderse cuanto antes en el ciclo de vida de un producto. Por esta razón es necesario automatizar las técnicas de "buferrización" y dimensionado adaptativo proporcionando una ayuda directa al diseñador.

El algoritmo de optimización utiliza funciones de coste y curvas retardo-potencia-área que constituyen auténticas radiografías del circuito a implementar. El diseñador puede, en base a estos diagramas, tomar de forma interactiva o automática decisiones de diseño a nivel *pre-layout*. La herramienta se encarga de optimizar el circuito de forma muy rápida. Una vez decidido y editado un *layout*, pueden hacerse ajustes finos optimizando el *netlist* extraído.

La metodología desarrollada ofrece al diseñador información de primer orden, aportando el dimensionado óptimo y la correcta "buferrización" para un circuito determinado. Por último se realizan unos diseños experimentales que son empleados tanto para validar los resultados como para generar conclusiones sobre el diseño en GaAs.

El empleo de una optimización de tipo analítico hace posible obtener unas conclusiones a las que no es posible llegar mediante algoritmos de tipo heurístico. En base a esto, el diseñador dispone del conjunto completo de posibles soluciones ajustadas al criterio de optimización elegido. En esta tesis se demuestra que dado un diseño es posible reducir apreciablemente el consumo de potencia mediante la aplicación de la estrategia de optimización propuesta. El grado de optimización alcanzable en tiempo de propagación y en área es igualmente significativo, aunque menos notable. Los resultados revelan que la aplicación de la metodología de optimización, lleva la tecnología empleada (DCFL con E/D MESFET de $0.8\mu\text{m}$) *prácticamente* hasta sus límites en cuanto a velocidad se refiere (para las condiciones normales de operación).

Los programas de optimización que se desarrollan contienen aproximadamente 2000 líneas de código en lenguaje C, y hacen uso de numerosas tablas con coeficientes de caracterización y modelos. El código generado no tiene el nivel de herramienta comercial sino de prototipo, y deja espacio para su posterior mejora.

ORDENACIÓN DE ESTA MEMORIA

La memoria de esta tesis se compone de siete capítulos. El capítulo 1 se dedica al planteamiento del problema y a la revisión del estado del arte. El capítulo 2 versa sobre la descripción y análisis circuital de las puertas básicas constituyentes de la lógica bajo estudio (DCFL/SDCFL). El capítulo 3 expone los modelos estimativos soportes de la estrategia de optimización a plantear en capítulos 4 y 5. Los resultados obtenidos con el método se presentan en el capítulo 6. En el capítulo 7 se presentan las principales conclusiones de esta tesis y las líneas de investigación abiertas.

Se enmarcan dentro de la categoría de *capítulos de desarrollo de la tesis* los siguientes: 3, 4, 5 y 6. En el capítulo 3 se exponen los modelos expresamente elaborados para la consecución de esta tesis. Se aclara que para el modelo estimativo de retardos de propagación, a diferencia de los otros dos (potencia consumida y área ocupada), se ha procedido a la adecuada adaptación del modelo ya disponible [Herná92] en cuya implementación ha colaborado profusamente el autor de esta tesis.

En el capítulo 4 se presentan las técnicas de "bufeización" propuestas, técnicas que conjuntamente con el algoritmo de optimización que se expone en el capítulo 5, conforman el cuerpo de esta tesis. La aplicación conjunta de las técnicas de "bufeización" y el algoritmo de dimensionado de transistores se ha denominado *estrategia de optimización*.

En el capítulo 6 se aplica la estrategia de optimización desarrollada a varios diseños experimentales de referencia. Asimismo la discusión de los resultados obtenidos encuentra cabida en este capítulo.

Se ha seguido el criterio de escribir en cursiva los términos técnicos no pertenecientes al habla castellana.

AGRADECIMIENTOS

Esta es sin duda una de las páginas más complicadas de redactar. No resulta una tarea fácil el condensar en este espacio tan limitado el agradecimiento a las personas que han contribuido a que este trabajo vea la luz, pero con el ánimo de resaltar a las personas que han realizado una inestimable aportación a esta tesis, va mi agradecimiento:

A Antonio Núñez, director de la tesis. Por su apoyo y disposición constante y sus múltiples y acertadas $n+1$ contribuciones, tanto técnicas como literarias.

A Toni, compañero y colaborador. Por las veladas nocturnas y las innumerables discusiones científicas que hemos disfrutado juntos y que siempre recordaré con agrado.

Al grupo PATMOS cuna de este trabajo y especialmente a Salvador Perdomo. Por esa larga amistad y por las conversaciones técnicas mantenidas.

A mi entrañable amigo Palmés. Por esa visión transparente de las matemáticas y por ese buen humor. Espero ganarme un hueco en el apartado de agradecimientos de tu futura tesis.

No quisiera olvidar a las personas que han contribuido con su aportación en aspectos parciales. A Rafa Montenegro por esas tardes que me dedicó y por sus valiosos consejos matemáticos. A Roberto Esper-Chaín por su siempre eficaz colaboración y a Valentín, que me ayudó en la realización de algunas de las ilustraciones de esta tesis.

Quiero extender mi agradecimiento a mi familia, con mención especial a mi hermano Emilio y a las personas que me han rodeado durante los años de preparación de esta tesis.

Por último quiero expresar mi más emotivo agradecimiento a quien me ha brindado una compañía y un apoyo decisivo, *Barbara*.

*El dedo que se mueve escribe; y, después de escribir,
Sigue moviéndose: ni toda tu piedad ni talento
le impulsarán a cancelar media Línea,
Ni todas tus lágrimas borrarán una palabra.*

Omar Khayyam

a Alice

ÍNDICE

RESUMEN	v
ORDENACIÓN DE ESTA MEMORIA	vii
AGRADECIMIENTOS	ix
ÍNDICE	xv
CAPÍTULO 1	
INTRODUCCIÓN	1
1.1.- ESTADO DEL ARTE EN OPTIMIZACIÓN ELÉCTRICA.	3
1.1.1.- Técnicas de <i>sizing</i>	3
1.1.2.- <i>Sizing</i> según procedimientos de tipo heurístico	4
1.1.3.- <i>Sizing</i> según procedimientos analíticos	5
1.1.4.- Técnicas de "bufeización" (<i>buffering</i>)	7
1.2.- ESTADO DEL ARTE EN MODELOS ESTIMATIVOS DEL RETARDO	
DE PROPAGACIÓN	13
1.2.1.- Modelos de retraso unidad.	13
1.2.2.- Modelos basados en tablas empíricas.	14
1.2.3.- Modelos RC	15
1.2.4.- Modelos analíticos.	15
1.3.- ESTADO DEL ARTE EN MODELOS ESTIMATIVOS DE	
POTENCIA	16
1.4.- ESTADO DEL ARTE EN MODELOS ESTIMATIVOS DEL ÁREA	
OCUPADA.	17
1.5.- PLANTEAMIENTO DEL PROBLEMA	18
CAPÍTULO 2	
ESTUDIO DE LA FAMILIA LÓGICA DCFL	21
2.1.- INTRODUCCIÓN	22
2.1.1.- Familia lógica directamente acoplada (DCFL).	23
2.1.2.- Familia con seguidor de fuente (SDCFL).	26
2.2.- ANÁLISIS.	29
2.2.1.- Familia DCFL.	29
2.2.2.- Familia SDCFL.	33
2.3.- CONCLUSIONES	36

CAPÍTULO 3

MODELOS ESTIMATIVOS DEL RETARDO DE PROPAGACIÓN,

CONSUMO DE POTENCIA Y ÁREA OCUPADA	37
3.1.- FORMULACIÓN DE RETARDOS.	38
3.1.1.- Formulación e hipótesis de partida.	38
3.1.1.1.- Definiciones.	38
3.1.1.2.- Aproximación polinómica.	40
3.1.1.3.- Precaracterización.	42
3.1.1.4.- Puertas lógicas de múltiples entradas: inversor equivalente.	42
3.1.1.5.- Colisiones.	43
3.1.2.- Resultados.	45
3.1.3.- Conclusiones.	46
3.2.- MODELO PARA EL CÁLCULO DE LA POTENCIA DISIPADA.	47
3.2.1.- Formulación e hipótesis de partida.	47
3.2.2.- Definiciones.	53
3.2.3.- Metodología de cálculo de las funciones de estimación.	54
3.2.4.- Capacidades asociadas al cableado.	59
3.2.5.- Ajuste de P_{OL} y de P_{OH}	60
3.2.6.- Resultados.	65
3.2.7.- Conclusiones.	67
3.3.- MODELO PARA LA ESTIMACIÓN DEL ÁREA OCUPADA.	69
3.3.1.- Hipótesis de partida.	69
3.3.2.- Consideraciones generales.	70
3.3.2.1.- Celda y Estructura.	70
3.3.2.2.- Reglas de Diseño.	70
3.3.2.3.- Metal 1 y Metal 2.	70
3.3.2.4.- Estilos de Diseño.	71
3.3.2.5.- Área intrínseca y área extrínseca.	72
3.3.3.- Metodología.	72
3.3.3.1.- Modelo a nivel de puerta.	73
3.3.3.2.- Modelo a nivel de transistor.	78
3.3.4.- Resultados.	79
3.3.4.1.- Evaluación del error cometido.	80
3.3.4.2.- Definición del parámetro <i>delta</i>	81
3.3.5.- Conclusiones.	82
CAPÍTULO 4	
TÉCNICAS DE "BUFERIZACIÓN"	83
4.1.- INTRODUCCIÓN.	84
4.2.- ESTRATEGIA GENERAL A SEGUIR.	85
4.3.- TÉCNICAS DE "BUFERIZACIÓN".	85

4.3.1.- Necesidad de las técnicas de "bufeización".	86
4.3.2.- Diseño robusto y diseño adaptativo.	87
4.4.- ELECCIÓN DE UN CONJUNTO DE <i>BUFFERS</i>	88
4.4.1.- Circuito estudiado y condiciones de medida.	88
4.4.2.- Limitación del número de geometrías a estudiar.	90
4.4.3.- Diseño robusto: elección de la geometría del <i>buffer</i> a emplear.	92
4.4.4.- Diseño adaptativo: criterios de elección y conjunto de <i>buffers</i> a considerar.	96
4.4.4.1.- Elección de la geometría guiada por el consumo de potencia.	96
4.4.4.1.1.- Características potencia-retardo.	101
4.4.4.2.- Elección de la geometría guiada por el retardo de propagación.	101
4.4.4.2.1.- Conjunto de <i>buffers aceleradores</i>	102
4.4.4.3.- Aplicación de las técnicas desarrolladas a un circuito dado.	103
4.5.- ESTRATEGIA OPCIONAL PARA ACELERAR LA LÓGICA.	105
CAPÍTULO 5	
ALGORITMOS Y ESTRATEGIA GLOBAL DE OPTIMIZACIÓN	109
5.1.- INTRODUCCIÓN.	110
5.2.- PROPIEDADES ANALÍTICAS DE LOS MODELOS A EMPLEAR.	111
5.2.1.- Precisión en la estimación.	112
5.2.2.- Funciones analíticas.	112
5.2.3.- Limitaciones de los modelos (restricciones).	113
5.2.3.1.- Modelo temporal: limitaciones.	113
5.2.3.2.- Modelo de estimación de potencia: limitaciones.	115
5.2.3.3.- Modelo de estimación del área: limitaciones.	115
5.3.- Clasificación del problema.	116
5.3.1.- Técnicas de optimización de circuitos.	116
5.3.1.1.- Tipos de problemas de optimización.	116
5.3.1.2.- Clasificación de los problemas no lineales y técnicas de resolución.	118
5.4.- ESTUDIO DE LA FUNCIÓN OBJETIVO A OPTIMIZAR.	119
5.4.1.- Definición de la función objetivo.	120
5.4.1.1.- Estudio de la función de estimación del retraso. Aplicación a una cadena de n puertas lógicas.	120
5.4.1.2.- Estudio de la función de estimación de la potencia. Aplicación a una cadena de n puertas lógicas.	124
5.4.1.3.- Estudio de la función de estimación del área. Aplicación a una cadena de n puertas lógicas.	125

5.4.2.- Funciones de compromiso: relaciones.	126
5.4.3.- Coeficientes de peso.	127
5.4.4.- Cálculo de capacidades.	127
5.4.5.- Cálculo de los coeficientes $\beta_{fan-out}$	128
5.4.6.- Condiciones de contorno.	130
5.4.7.- Estudio de la convexidad de la función objetivo.	131
5.4.7.1.- Restricción al caso de dos variables.	132
5.4.7.1.1.- Función estimativa de la potencia: restricción al caso de dos variables.	132
5.4.7.1.2.- Función estimativa del retardo de propagación: restricción al caso de dos variables.	133
5.4.7.1.3.- Representación de la función objetivo para el caso restringido empleando dos variables.	134
5.4.7.2.- Programación no convexa.	135
5.5.- ELECCIÓN DEL ALGORITMO NUMÉRICO PARA LA RESOLUCIÓN DEL PROBLEMA.	136
5.5.1.- Conjunto de soluciones no inferiores.	136
5.5.2.- Algoritmo de minimización de la función objetivo.	138
5.5.3.- Aplicación del algoritmo al ejemplo de n puertas lógicas.	141
5.6.- IMPLEMENTACIÓN DEL ALGORITMO: PROGRAMA DESARROLLADO.	142
5.6.1.- Descripción del programa: diagrama de flujo.	143
5.6.2.- Ampliaciones del programa.	148
5.7.- ESTRATEGIA GLOBAL DE OPTIMIZACIÓN.	149
CAPÍTULO 6	
APLICACIÓN DE LA ESTRATEGIA DE OPTIMIZACIÓN A DISEÑOS EXPERIMENTALES DE REFERENCIA	
6.1.- INTRODUCCIÓN.	151
6.2.- DESCRIPCIÓN DEL CIRCUITO EJEMPLO N° 1.	153
6.2.1.- Estimación del retardo, potencia y área para la implementación según metodología de diseño robusta.	154
6.2.2.- Aplicación del algoritmo de dimensionado.	156
6.2.3.- Diagrama de soluciones no inferiores para el compromiso retardo- potencia.	158
6.3.- DESCRIPCIÓN DEL CIRCUITO EJEMPLO N° 2.	161
6.3.1.- Aplicación del algoritmo de dimensionado.	162
6.3.2.- Aplicación de las técnicas de "bufferización".	164
6.3.3.- Estudio de la dependencia de la solución óptima con la pendiente de entrada al circuito.	166
6.4.- DESCRIPCIÓN DEL CIRCUITO EJEMPLO N° 3.	168
6.4.1.- Descripción del circuito y aplicación de la estrategia de	

optimización.	168
6.4.2.- Aplicación del algoritmo de dimensionado.	168
6.5.- CONCLUSIONES.	172
CAPÍTULO 7	
CONCLUSIONES Y LÍNEAS ABIERTAS	173
7.1.- CONCLUSIONES.	173
7.2.- LÍNEAS FUTURAS.	175
REFERENCIAS	179

CAPÍTULO 1

INTRODUCCIÓN.

La tecnología de Arseniuro de Galio (GaAs) ha experimentado en los últimos años un gran avance. Dicha tecnología ofrece importantes ventajas para el diseño digital de alta velocidad. El diseño *full-custom* en GaAs -dadas las propiedades del material y su costo- es un diseño para prestaciones. De no ser así, no se entiende su aplicación. A diferencia del diseño tradicional en tecnología de Silicio, resulta imprescindible la estimación inicial de ciertas funciones de coste de cara a tener una visión preliminar del sistema a implementar.

En el diseño de circuitos digitales GaAs, el análisis de viabilidad toma un carácter prioritario, motivado principalmente por dos razones:

- el diseño para prestaciones apunta a objetivos muy concretos y,
- el escenario de diseño es más complejo (la toma de decisiones de compromiso no es tarea fácil).

El diseño de circuitos digitales de GaAs es un problema multidimensional. El diseñador dispone de mayor libertad debido entre otras razones a que no se encuentran estandarizados los valores de los niveles lógicos o de las fuentes de alimentación (situación distinta del caso CMOS). Además no puede afirmarse el predominio de una familia lógica o topología sobre otra hasta que no terminen de asentarse de forma definitiva.

El objetivo de todo proceso de diseño consiste en pasar de unas especificaciones iniciales, definidas en muy alto nivel, a su correspondiente realización física (*layout*). En el ámbito GaAs, el número de factores que afectan al diseño es mucho mayor y el estudio de viabilidad muestra mayor complejidad. De especial relevancia es la consideración del ruido (tanto dinámico como estático) durante el diseño de circuitos

GaAs. Esta es una circunstancia que, a diferencia de lo que acontece en circuitos CMOS, se ha tenido en cuenta ya desde los primeros diseños GaAs.

Tal como ocurrió hace años con los circuitos digitales de Silicio, los niveles de integración alcanzados en circuitos GaAs basados en MESFETs (*METal Semiconductor Field Effect Transistor*) han superado actualmente la llamada frontera *VLSI*. Los diseñadores demandan cada vez más herramientas (o estrategias) para optimizar los diseños de forma eficiente y rápida. La optimización se encauza principalmente hacia la mejora del compromiso definido por el retardo de propagación, el consumo de potencia y la superficie ocupada.

Estos tres aspectos confluyen hacia un serio conflicto cuando el diseño se aproxima a la frontera *VLSI*. Destaca entre ellos el factor retardo de propagación (*velocidad*), soporte más que justificado de la tecnología GaAs. La proximidad al nivel *VLSI* se ve obstaculizada por el consumo de potencia, de vital importancia al ser básicamente estático [Cates90] y por consiguiente siempre incómodamente presente. No de menor importancia resulta el acomodar de forma óptima el diseño sobre la costosa superficie de GaAs disponible.

Asimismo y debido a las particularidades que presenta la lógica bajo estudio en esta tesis (DCFL), el marco de variables a optimizar no se reduce a las tres mencionadas anteriormente, sino que hay que ampliarlo para introducir una nueva, el margen de ruido. Dicha variable no precisa de un trato particular en los actuales diseños realizados en CMOS.

El complejo problema de la optimización, merced al gran número de variables a manejar, precisa de una descomposición en niveles para poder ser acometido. Típicamente este estudio se hace a tres niveles:

- optimización lógica,
- optimización topológica y,
- optimización circuital (o eléctrica).

El primero de ellos está enfocado a la optimización a nivel funcional, es decir, de las expresiones booleanas. El segundo nivel se dirige al estudio de cómo aprovechar óptimamente el área disponible en el *chip*. Y el tercero de ellos aborda el problema del dimensionado de transistores y el manejo de cargas. Se considerará en este estado del arte, así como en esta tesis, la optimización eléctrica, sin olvidar que permanecerán siempre subyacentes aspectos topológicos.

Dado que no se conoce la existencia de ninguna herramienta (o incluso estrategia) desarrollada para circuitos GaAs en este tema, se pasará revista a los métodos disponibles para el caso del Silicio, comentándose en detalle los más relevantes. En esta línea, se presentará una clasificación acorde a los aspectos metodológicos. Donde sea pertinente se indicarán las restricciones de la aplicación de esos métodos a circuitos GaAs o sus posibles relaciones.

1.1.- ESTADO DEL ARTE EN OPTIMIZACIÓN ELÉCTRICA.

El complejo problema de la optimización eléctrica presupone un diseño óptimo desde el punto de vista del análisis lógico, y trata de mejorar sus prestaciones puramente eléctricas. Para el logro de tal fin se requieren modelos precisos para estimar el retardo de propagación, la potencia consumida y el área ocupada.

Las estrategias posibles de cara a optimizar eléctricamente un circuito son principalmente dos:

- técnicas de *sizing* (dimensionado) y,
- técnicas de *buffering* ("bufeización").

La metodología que se emplee para realizar el dimensionado (o la "bufeización") introduce una segunda clasificación del problema. Así, se distinguen los métodos de tipo heurístico de los analíticos.

Se pasa seguidamente a presentar los logros más importantes conseguidos mediante el empleo de dichas técnicas en los circuitos MOS. Al igual que sucede en el campo de la síntesis y optimización lógica, poco (o nada) hay desarrollado para su aplicación a la tecnología GaAs.

1.1.1.- Técnicas de *sizing*.

Se engloba en este apartado el conjunto de técnicas o algoritmos que permiten decidir un dimensionado óptimo de los transistores acorde a unas restricciones impuestas por el diseñador. Este dimensionado puede llevarse a cabo según dos tipos de estrategias:

- local (implicando, por lo general al camino crítico) o,
- global (implicando auténticas porciones del circuito).

La necesidad de esta división del problema surge, una vez más, como consecuencia de intentar reducir el espacio de trabajo. La evolución de las técnicas de optimización ha arrancado de la metodología local para, centrarse actualmente en procedimientos de ámbito global.

Asimismo, en dicha evolución se ha pasado de lo que se ha dado en llamar técnicas de tipo heurístico a las poderosas estrategias analíticas. La diferencia fundamental entre ambas radica en la definición del problema. Así, mientras que la metodología de tipo analítico parte de una formulación matemática del problema para su resolución, la metodología heurística resuelve el problema mediante una estrategia basada en realizar simulaciones y verificaciones con el ánimo de acercarse a la solución.

Las herramientas sustentadas sobre metodologías de tipo analítico son siempre más poderosas que las heurísticas y el conjunto de soluciones obtenido es más transparente de cara al usuario. Pero esto no constituye motivo alguno para que la optimización de tipo heurístico no siga siendo campo de investigación (HSPICE [HSPIC90] tiene incorporado una subrutina que permite realizar este tipo de optimización).

1.1.2.- Sizing según procedimientos de tipo heurístico.

Cuando no se dispone de sencillos modelos analíticos para el cálculo de retardos (como tiene lugar actualmente en tecnología GaAs y sucedió en las décadas pasadas en los diseños MOS), las estrategias de optimización han de formularse desde un punto de vista heurístico. Existen múltiples algoritmos para optimización heurística (ver por ejemplo [Joupp83], [Trimb83], [YuaSv89]). De entre los numerosos programas implementados destaca el programa denominado MTA [HofKi87] el cual se puede considerar como una mejora del programa que fuera pionero en este campo TILOS [FishDu85].

El programa MTA fue desarrollado en el año 1987, y se encuentra escrito en el lenguaje de programación C. Es una poderosa herramienta para dimensionar transistores de aplicación a circuitos combinacionales CMOS. Como característica adicional, que se sale de este apartado, MTA es también adecuado para realizar "buferezación". A diferencia de otros optimizadores (entre ellos TILOS), no se encuentra limitado sólo a circuitos implementados con PLAs o máquinas de estado finito, sino que MTA es capaz de manejar circuitos basados en células estándar.

Presenta otras innovaciones respecto a los optimizadores típicos de su

generación. Entre ellas cabe citar que incorpora una descripción jerárquica del circuito y estima los retardos mediante el empleo de una tabla de datos. Con los datos contenidos en las *look-up tables* se evalúa un polinomio que es función de la pendiente de entrada, del número de transistores, de las capacidades asociadas al interconexiónado y del tamaño de los transistores. Además incorpora una estrategia para tener en cuenta efectos de colisiones entre señales ([MeDaJ91], [Herná92]).

En el programa MTA se emplea la misma ecuación de *sensibilidad* que utiliza el optimizador TILOS para determinar los transistores cuyas dimensiones son susceptibles de ser alteradas. A través de la múltiple evaluación y cambio en el dimensionado de los transistores, el algoritmo se detiene una vez alcanzada la solución que satisface los objetivos planteados. Este es uno de los primeros programas que emplea una formulación posinómica. Este tipo de formulación es ampliamente empleada en la actualidad [HoNeS90].

La bondad de la optimización lograda se sitúa en torno al 10 % para el retardo de propagación. Los tiempos de cómputo son lo suficientemente bajos como para poder aplicar el programa a diseños que tienen un elevado número de transistores. Por lo general se precisan pocas iteraciones hasta alcanzar la estabilidad de la solución. Aún así, no siempre alcanzan la solución demandada lo cual puede llevar a tiempos de cómputo inútiles. Esta es una característica común a los optimizadores heurísticos [YuaSv88].

1.1.3.- *Sizing* según procedimientos analíticos.

Debido a lo tedioso de los procesos iterativos empleados en optimización heurística, su aplicación sólo es factible para llevar a cabo optimización local. Las técnicas analíticas pueden ser empleadas, con mayor éxito, para el problema de optimización global [AuAzD91].

El desarrollo de una optimización de tipo global no es posible sin la disponibilidad de algún procedimiento analítico. Mientras resulta relativamente sencillo formular modelos analíticos para la estimación de la potencia y del área, la situación es completamente distinta para el caso de los modelos temporales. Desde que se dispuso de los primeros modelos analíticos para la estimación de retardos (para circuitos MOS), una investigación importante se centró en el campo de la optimización analítica. A esta época pertenecen trabajos que se remontan a la década de los setenta [RuWoG77] y un conjunto más reciente está representado por ([Marpl86], [AuAzD91]).

Básicamente no hay grandes diferencias entre los distintos algoritmos de resolución del problema no lineal planteado. Todos ellos operan con multiplicadores de Lagrange y computan las derivadas de la función objetivo con respecto de la variable dimensión de puerta del transistor. El matiz diferenciador subyace en el modelo temporal que se emplee. Se puede establecer una clasificación según sea el modelo que se utilice. En esta línea se puede hablar de algoritmos de optimización que evalúan retardos mediante modelos temporales basados en aproximaciones RC (o de precisión similar) y los que hacen uso de modelos analíticos.

Los optimizadores analíticos basados en el cálculo de retardos mediante modelos RC pueden consultarse en [Hedlu87], [Cirit87]. Los optimizadores analíticos contruidos sobre modelos matemáticos se detallan en [RoDeP88], [HoNeS90]. Seguidamente se describe el programa AESOP, perteneciente al primer grupo, indicando sus características principales y restricciones más significativas.

AESOP (Automated Electrical Simulation and Optimization) es desarrollado en el año 1986 en la Universidad de Carolina del Norte. Se trata de una herramienta capaz de optimizar eléctricamente circuitos combinacionales, bien sean CMOS o NMOS. El espacio de diseño permite manejar restricciones de área (mínima-máxima), potencia (mínima-máxima) y retardos (mínimo-máximo). El margen de ruido no es considerado.

El área ocupada y la potencia consumida se modelan a través de expresiones lineales sencillas (ambas funciones proporcionales al ancho de puerta del transistor). La evaluación de retardos y la búsqueda de caminos críticos se hace utilizando el programa analizador temporal CRYSTAL [Ouste83] introduciendo unas mínimas modificaciones.

El empleo del modelo basado en la aproximación en árboles RC conduce a errores en la evaluación de retardos típicamente del 20%. Cuando se requiera mayor precisión puede utilizarse una estimación mediante SPICE [Nagel75] y seguidamente aplicar los algoritmos AESOP.

La formulación del problema se completa con las condiciones especificadas por el diseñador. El problema que se plantea es un problema de tipo no lineal, debido a la expresión para el cálculo de retardos. El algoritmo de optimización se resuelve mediante el método quasi-Newtoniano que emplea una aproximación de 2º orden para la expresión del retardo. De esta forma se linealiza el problema. Las condiciones de contorno no lineales se introducen a través de los multiplicadores de Lagrange.

El problema está bien condicionado (desde el punto de vista matemático) y la solución óptima se alcanza en pocas iteraciones (AESOP es capaz de dar un

dimensionado óptimo de un circuito que tiene más de dos mil transistores en un minuto). Como principales limitaciones pueden citarse:

- AESOP no optimiza a nivel de transistores, sino a nivel de puerta; asocia un factor de escala a cada puerta lógica para dimensionar los transistores. Puede adaptarse para que optimice a nivel de transistores a costa de un excesivo coste de tiempo de computación,
- los efectos de pendientes no están incluidos en AESOP.

El empleo de tablas de datos hacen a AESOP una herramienta independiente de la tecnología. AESOP brinda la posibilidad de seleccionar múltiples caminos para optimizar y no sólo el *camino crítico*. Porciones del circuito pueden ser también optimizadas (siempre partes combinacionales).

1.1.4.- Técnicas de "buferrización" (*buffering*).

A diferencia de lo que acontece al caso GaAs, para los circuitos CMOS existe abundante literatura tratando el problema de la "buferrización". Si bien no es objeto de debate en circuitos CMOS cuál es el tipo de *buffer* a emplear, la situación es bastante distinta cuando se trata de circuitos pertenecientes a la tecnología GaAs. Para circuitos CMOS el *buffer* es un inversor, mientras que el tipo de *buffer* para la lógica DCFL no se encuentra plenamente definido.

Esta revisión de técnicas de "buferrización" se centrará en los circuitos CMOS por ser la lógica más extensamente estudiada en la literatura, y con el ánimo de obtener inspiración para estudiar las lógicas de GaAs.

Una vez más, un planteamiento analítico será el deseado para la correcta resolución del problema. Dichos modelos están disponibles para el caso de los circuitos MOS, no siendo éste el estado actual referente a circuitos de GaAs. Del amplio trabajo desarrollado se pasa a comentar los logros que se consideran más importantes para nuestro trabajo.

En el año 1980 Mead y Conway [MeaCo80] deducen las primeras expresiones para el diseño óptimo de *buffers*. Son éstas unas expresiones obtenidas a partir de modelos muy simplificados que sentarían la base de posteriores estudios. Estaban orientadas a la lógica NMOS dominante en la época. Mediante las fórmulas matemáticas obtenidas, puede calcularse el número de etapas (inversores) a disponer para conseguir un retardo de propagación mínimo. Consideraciones respecto a potencia consumida o

área ocupada no son tenidas en cuenta.

En el año 1984 destaca otra importante contribución, esta vez de la mano de C. Lee y H. Soukup [LeeSo84]. Se presenta aquí un algoritmo capaz de calcular el número adecuado de etapas de inversores y el dimensionado óptimo de los transistores. La estrategia de optimización llevada a cabo permite reducir el retardo de un determinado camino crítico al mínimo posible para una tecnología dada, o minimizar el área de Silicio para un determinado retardo. El consumo de potencia no es considerado en la optimización. El procedimiento a seguir es de tipo analítico.

Se emplea un sencillo modelo físico (no se precisa de parámetros de ajuste) para la evaluación de retardos. Dicho modelo incluye asimismo capacidades asociadas al cableado. La resolución del algoritmo pasa por el uso de multiplicadores de Lagrange y un proceso iterativo. Pese a la sencillez del modelo temporal, los resultados obtenidos son excelentes. Radica además la importancia de este trabajo en el planteamiento y forma de resolución del problema, sentando líneas de trabajo seguidas en posteriores contribuciones.

En el año 1987 Hedenstierna y Jeppson [HedJe87] publican una importante investigación, planteando un modelo analítico para el cálculo de retardos que será ampliamente empleado posteriormente para resolución de problemas de optimización. En este trabajo se presenta un modelo (macromodelo) para la evaluación del retardo de propagación del inversor CMOS. Dicho modelo incluye efectos de pendiente (a diferencia de los modelos incorporados en AESOP), la cual es modelada como una rampa. No se conciben a partir de este momento, aquellos modelos temporales que no incluyan efectos de pendiente de señales.

El problema de la "buferrización" se plantea con la adición de un estudio de capacidades. Se obtiene de forma analítica el número de etapas idóneas así como el dimensionado óptimo de los transistores. En esta línea se mejora la aportación de Mead y Conway, al incluir efectos de capacidades intrínsecas no consideradas en la formulación de estos autores. Esta optimización, a diferencia de las anteriores presentadas, sí considera el área ocupada por los *buffers* además de la potencia consumida por los mismos.

Aparece un interesante artículo en el año 1989 con aplicación a compiladores de Silicio de la mano de A. Al-Khalili. Se presentan en esta contribución [AlZhA89] los modelos correspondientes para la evaluación de retardos, área estimada y consumo de potencia. El modelo para el retardo es analítico y trata con pendientes y con capacidades de cableado. Asimismo dispone de expresiones para el cálculo del consumo

de potencia de los *buffers*. Como aspecto a destacar es que estos modelos se encuentran formando parte de un generador de módulos de *buffers*. La herramienta dispone de los ficheros tecnológicos correspondientes, un optimizador y un generador automático de *layout*. Los *buffers* generados son optimizados tanto en prestaciones (entiéndase eléctricas) como en cuanto al *layout*.

En el año 1990 Hoppe y sus colaboradores publican un artículo [HoNeS90] donde se presenta la herramienta MOGLO para la optimización global de circuitos CMOS. La revisión de este trabajo se hará a través de la descripción de las características del programa MOGLO.

MOGLO (Multiple Objective Gate-Level Optimization), constituye una poderosa herramienta para la optimización de circuitos combinatoriales implementados en tecnología CMOS. Para el modelado de retardos, tiene incorporado un macromodelo (modelo completo para la puerta básica: el inversor). Dicho modelo es pseudo-analítico y está basado en los trabajos publicados por Hedenstierna y Jeppson [HedJe87]. El atributo "pseudo" viene a colación de un ajuste (vía SPICE) con la intención de incorporar los efectos de canal corto.

En MOGLO se utiliza un procedimiento similar al que se emplea en AESOP para escalar las puertas: un solo término por puerta que dé cuenta del dimensionado de todos los transistores. En concreto esto se hace diseñando la puerta con todos los transistores "tipo p" de la misma dimensión, W_p , (lo mismo para los "tipo n", W_n). Como factor de escala se toma el término W_n/W_p , elegido de tal forma que se modele un peor caso. De esta forma cada puerta queda identificada a través de un solo número. Las distintas puertas lógicas (NOR, NAND...) son reducidas a inversores equivalentes y mediante el macromodelo se estiman los retardos de propagación. El inversor equivalente es definido como aquel que retrasa tanto la señal de entrada como lo haría la puerta en cuestión. Se determinará ajustando el ancho de puerta del inversor.

Se incluye en la formulación del modelo los efectos asociados a la pendiente de ataque y a las capacidades de cableado. Se incorpora al modelo de Hedenstierna la inclusión de la capacidad asociada al efecto Miller, lo cual se hace mediante un factor de corrección extraído vía SPICE. La forma de onda es reproducida de forma analítica, y se conoce como señal de onda efectiva o equivalente. La extracción de todos los parámetros de ajuste se realiza de forma automática mediante la rutina de extracción (GAOP).

La precisión del modelo temporal presenta un error máximo respecto a SPICE del cinco por ciento (5%). Con esta precisión es de esperar que los resultados superen

a los suministrados por el programa AESOP, el cual basaba el cálculo de retardos en la aproximación RC con un error estimado mínimo del diez por ciento (10%). El área estimada se deduce de expresiones lineales sencillas, siguiendo las aproximaciones clásicas [LeeSo84], [ShSaF88]. El consumo de potencia se evalúa a través de una expresión lineal que tiene en cuenta la actividad de los nodos (probabilidad de carga o descarga que sufre durante un ciclo de reloj).

Estos modelos empleados gozan de las siguientes características: precisión, rapidez en la evaluación y son funciones continuas del ancho de puerta (variable a optimizar). Además, los modelos presentados son funcionalmente posinomios de los parámetros de diseño. Esto le atribuye una importante propiedad: las funciones posinómicas son siempre convergentes, por lo que el mínimo que resulte de la resolución del problema, será el mínimo absoluto. Esta propiedad no es exclusiva de MOGLO sino que ya había sido explotada por optimizadores anteriores [FisDu85].

El problema formulado se clasifica como un problema de optimización geométrica (no lineal) multivariable, que se resuelve aplicando multiplicadores de Lagrange. A su vez las funciones de Lagrange son minimizadas aplicando el algoritmo de Newton-Raphson. MOGLO permite estudiar el compromiso retardo-potencia-área para el diseño completo y obtener el conjunto de soluciones no inferiores para dicho compromiso. En el capítulo 5 se detallarán más aspectos acerca de la estrategia de optimización empleada por Hoppe. En cuanto al tiempo de cómputo se refiere, MOGLO no es especialmente costoso, pudiendo generar los gráficos típicos de evaluación de funciones en un tiempo de pocas horas (para circuitos de gran tamaño). Hay que indicar que generar esos gráficos vía SPICE es, sino imposible, sí prohibitivo.

En el año 1988 M. Robert, D. Deschacht y colaboradores publican un artículo donde presentan una herramienta de compilación (optimizador y generador de *layout*). Esta herramienta está disponible comercialmente bajo el nombre de PRINT [RoDeP88]. Existe literatura actualizada sobre PRINT [RoTrC91] (razón por la que hemos alterado la cronología de exposición). La revisión de este trabajo se hará a través de la descripción de las características del programa PRINT.

PRINT se define como una ambiciosa herramienta de compilación de aplicación a circuitos CMOS. Su versátil campo de aplicación le permite manejar tanto circuitos regulares (tipo células estándar) como circuitos *custom* (aunque MOGLO puede aplicarse a diseños *full-custom*, está orientado a diseños *semi-custom*). Asimismo no se encuentra restringido sólo a circuitos combinacionales sino que también los circuitos secuenciales tienen cabida en PRINT.

Su principal diferencia respecto de otros optimizadores radica en el modelo temporal empleado para el cálculo de retardos. PRINT incorpora un modelo físico, independiente de la tecnología, donde no existen parámetros de ajuste. Este modelo permite la definición de una formulación cerrada (explícita) para el cómputo de retardos. Los errores estimados frente a SPICE son inferiores al 5%. Además incorpora un estudio completo de capacidades (más completo que MOGLO), lo cual le confiere gran utilidad de cara a la evaluación de retardos tanto en la fase *pre-layout* como la comprometida *post-layout*. Este modelo está insertado en el analizador temporal PATHRUNNER [DePiR90], desarrollado también por este grupo de investigación.

Apoyado en la formulación explícita (directamente dependiente del ancho de puerta del transistor), el problema del *sizing* se reduce simplemente a la resolución del conjunto de ecuaciones resultante (función del número de transistores presentes). Las ecuaciones presentan un grado de complejidad tal que imposibilitan su resolución de forma rápida, por lo que en PRINT la estrategia de optimización se dirige al ámbito local. Pese a ello, los resultados obtenidos son excelentes y se computan de forma rápida. Pese a que la estrategia de optimización es local, se permite la variación simultánea de todos los transistores (en MOGLO los transistores se agrupan y se define una sola anchura para cada grupo de transistores). Los tiempos de cómputo son apreciablemente inferiores a los presentados por el optimizador MOGLO.

Tanto el analizador temporal PATHRUNNER como los algoritmos de optimización forman parte de PRINT. Además incorpora un compactador de *layout* y la captura del *layout* simbólico, que dotan a esta herramienta de un alto grado de automatismo. Ese conjunto de posibilidades hacen de PRINT un programa ideal para el desarrollo de diversos estilos de diseño de *layouts* optimizados.

Siguiendo la línea de los compiladores de *buffers*, D. Navarro y A. Roy [NavRo91] publican en el año 1991 un artículo donde se presenta un compilador de *buffers* para diseños optimizados. Este compilador realiza las siguientes funciones:

- evalúa capacidades de los inversores para determinar con precisión el tiempo de propagación, y calcular el área y la potencia del *buffer*,
- determina para cada una de las soluciones no inferiores el número de inversores óptimos,
- dimensiona todos los transistores,
- minimiza el *layout* particionando adecuadamente los transistores,
- genera un *layout full-custom* a partir de un fichero tecnológico dado.

Esta es una herramienta potente, la cual se encuentra insertada en un programa

de CAD, que permite el diseño optimizado completamente automatizado para circuitos CMOS *full-custom*. Se puede afirmar que presenta similitud con los trabajos desarrollados por el grupo de Hoppe, en el sentido que opera con funciones posinómicas e implementa una estrategia de optimización no lineal global y multiobjetivo. A diferencia del método usualmente utilizado por Hoppe [HoNeS90] ó AlKalili [AlZhA89], los cuales utilizan un cociente p/n constante para todos los transistores, el conjunto de soluciones no inferior produce soluciones más óptimas llegando incluso a igualar los resultados por el método tradicional con un consumo de potencia dos veces menor.

Esto pone de manifiesto que la solución óptima admitida como estándar no corresponde a una solución noinferior. Además en cuanto al generador de *layout*, al estar integrado en la determinación de las soluciones óptimas, se consigue unos resultados más fiables que haciéndolo por separado (a diferencia de la estrategia seguida por A. Al-Khalili que implica un gran número de iteraciones). Estos autores son los primeros en resolver de forma global la generación automática de *buffers*, teniendo en cuenta todos los parámetros de diseño, empleando una metodología completamente analítica.

Se ha comentado reiteradas veces a lo largo de esta revisión del estado del arte, que para el desarrollo de herramientas capaces de optimizar circuitos lo fundamental es disponer de un eficiente analizador temporal, pues los modelos de potencia y área muestran una dificultad no mayor que en el caso del Silicio. Además si es posible, se desea que el analizador evalúe los retardos mediante un modelo analítico. El hecho de que solo exista una reducida literatura concerniente con el problema de la optimización para el caso del GaAs [SaArC91], [SaMoC91], se debe a dos razones principalmente:

- dificultades a la hora de disponer de un modelo sencillo de retardos,
- dado los bajos niveles de integración no se requerían analizadores temporales (ni optimizadores). La *optimización* de los circuitos GaAs era completamente heurística o basada en la experiencia.

Si bien en cuanto al primer aspecto la situación permanece tal cual, no sucede lo mismo con el segundo; el nivel de integración para el caso GaAs ha sobrepasado ya la frontera marcada por el diseño *VLSI*, y por consiguiente, los diseñadores comienzan a demandar tales programas de ayuda. Es de esperar con ello el potenciamiento del desarrollo de estimadores temporales, así como su inclusión en optimizadores.

Las publicaciones relativas a la tecnología GaAs han apuntado siempre al hecho de conseguir que los circuitos operen a la máxima velocidad, manteniendo sólo un

interés escaso en cuanto al tema del *buffering* y *sizing* se refiere. Esta situación ha cambiado al ser ahora posible integrar sistemas en un *chip VLSI*, donde los compromisos de potencia-tiempo adquieren un papel prioritario.

Tras esta revisión del estado del arte en optimización se procede a presentar el estado actual del tema relativo a los modelos estimativos del retardo de propagación, consumo de potencia y del área ocupada. Una vez finalizada dicha revisión se estará en disposición de formular el problema abordado en esta tesis.

1.2.- ESTADO DEL ARTE EN MODELOS ESTIMATIVOS DEL RETARDO DE PROPAGACIÓN.

La utilidad de los simuladores digitales depende en gran medida de la consistencia y precisión con las que pueden modelar el funcionamiento lógico en un amplio rango de técnicas disponibles. Lo mismo es aplicable al caso en el que el propósito de la simulación es la verificación de las restricciones temporales impuestas o el análisis de las prestaciones temporales de grandes circuitos lógicos. En ese caso es preciso implementar en los simuladores lógicos algún modelo de cálculo del retraso que sufren las señales al propagarse. Los simuladores resultantes pueden denominarse analizadores o verificadores temporales.

En la literatura existente relativa a los modelos temporales que resuelven el problema del análisis temporal puede encontrarse formulaciones para distintas familias lógicas fundamentalmente para las TTL, ECL, EDMOS, NMOS, CMOS y BiCMOS. Se presentan diversas alternativas de cómputo de los retrasos.

Una posible clasificación de los modelos disponibles consiste en agruparlos atendiendo a la forma de las funciones de retraso como pertenecientes a una de las cuatro categorías siguientes: modelos de retraso unidad, modelos basados en tablas empíricas, modelos RC y modelos analíticos.

1.2.1.- Modelos de retraso unidad.

En los modelos de retraso unidad se asigna a todas las puertas lógicas el mismo valor de retraso de propagación. Algunos simuladores a nivel de puerta operan bajo este esquema con objeto de manipular circuitos secuenciales. Otros simuladores permiten que el usuario especifique los valores del retraso de algunas de las puertas del circuito.

En ambos casos los valores del tiempo de propagación obtenidos sólo pueden ser múltiplos enteros de cierto tiempo de propagación fundamental. Este tiempo fundamental se denomina con frecuencia "tiempo resoluble mínimo" (*minimum resolvable time, MRT*). La información se obtiene en forma de señales en dos estados introduciendo la dinámica del problema simplemente retrasando las transiciones el valor especificado del tiempo de propagación. En general, estos modelos de estimación del retraso introducen errores considerables en el cálculo pues se ignoran las dependencias con otros parámetros de circuito como las capacidades de carga a las puertas, "fuerza" de los dispositivos, pendientes de entrada (*input slew-rates*) y otros factores.

1.2.2.- Modelos basados en tablas empíricas.

En este grupo de modelos se han incluido aquellos en los que es parte fundamental de los mismos el proceso de precaracterización, entendido como la fase previa a la utilización del modelo propiamente dicho en la que se generan, por simulación o mediante medidas, tablas de datos que, posteriormente, serán el soporte del cálculo de resultados. El método basado en árboles RC podría considerarse como de este grupo, sin embargo, dada la gran variedad de modelos de esta clase se entiende que merecen un tratamiento aparte.

Este tipo de modelos son ampliamente utilizados en la construcción de modelos de dispositivos para simuladores eléctricos [Meije90]. Su ventaja principal es la rapidez de computación. Con ellos se obtiene una aceptable precisión al aplicarlos en la predicción de circuitos implementados con dispositivos previamente caracterizados.

En la estimación del retraso, las tablas son usualmente construidas a partir de simulaciones eléctricas (tipo SPICE). Así, Rao y sus colaboradores [RaTrH83] las utilizaron para definir operadores de retraso en el caso de circuitos CMOS.

Una alternativa en la que se requiere menos memoria a la hora de evaluar los tiempos de propagación fue propuesta por Jun [JunJu88], [JuJuP89]. En estos artículos las tablas generadas a partir de simulación eléctrica se ordenan y sufren un proceso de ajuste a una función polinómica. Los modelos obtenidos son de aplicación a circuitos NMOS y CMOS. Las variables independientes en la metodología son las pendientes de entrada a las puertas, las capacidades de carga y las dimensiones de los inversores. En su comunicación se asegura que el error medido con este método es del 5%, una figura que supera ampliamente a los métodos anteriores.

En el artículo se definen reglas para la inclusión de puertas de múltiples

entradas. Sin embargo, no se detallan expresiones con las que introducir las colisiones.

Este tipo de formulaciones tienen la ventaja de su flexibilidad, es decir, a priori, son aplicables los métodos a cualquier familia lógica si es posible definir con precisión los parámetros que se utilizan como variables independientes.

1.2.3.- Modelos RC.

En estos modelos un transistor (usualmente MOS) se considera como un resistor lineal y a cada nudo se asocia una capacidad, también lineal, a tierra. Como resultado se da cuenta del funcionamiento dinámico a través de modelos RC. De hecho se estima el tiempo de propagación en base a la constante de tiempo si se produce transición en los nudos.

Los modelos RC se hicieron populares en la segunda mitad de la década de los 70 en base a contribuciones como las de Bryant [Bryan80] y Hayes [Hayes82]. Penfield y Rubinstein [PenRu81], [RuPeH83] introducen los conceptos de evaluación de los retrasos por redes RC utilizando cotas máximas y mínimas de retraso, y proponen expresiones para el cálculo del retraso máximo y mínimo. Los modelos fueron ampliados con las contribuciones de Yu y sus colaboradores [YuWyZ85] a circuitos conteniendo lazos de resistores. Bajo este esquema se redujeron los límites superior e inferior del intervalo de valores de retraso.

Lin y Mead [LinMe84] extienden el método para incluir las pendientes en la evaluación de los tiempos de propagación. Chu y Horowitz [ChuHo87] desarrollaron un modelo con dos polos que incluye efectos de repartición de cargas en los nudos (este problema se encuentra desarrollado en una contribución de Tjörnström [Tjarn90]).

1.2.4.- Modelos analíticos.

En la estimación del retraso de circuitos digitales es deseable una formulación constituida por soluciones cerradas de las ecuaciones dinámicas que describen el funcionamiento de los bloques básicos del circuito. Las expresiones contendrán como parámetros valores con una significación física clara. Este tipo de modelos permite profundizar sobre la influencia de los distintos parámetros involucrados en el tiempo de propagación. A la vez se conoce de una manera explícita cuáles de los parámetros son críticos, con ello se posibilita la mejora de las prestaciones temporales del circuito. Otra ventaja de este tipo de formulación es que el modelo resultante es tecnológicamente independiente.

La obtención de modelos analíticos reposa sobre soluciones generales cerradas de las ecuaciones diferenciales que describen el funcionamiento transitorio de la puerta lógica que, sin embargo, no se pueden encontrar. De modo que, dado que las ecuaciones resueltas en la literatura disponible corresponden a problemas equivalentes simplificados, hipótesis simplificadoras distintas conducen a formulaciones también distintas del mismo problema que serán válidas en determinadas condiciones.

1.3.- ESTADO DEL ARTE EN MODELOS ESTIMATIVOS DE POTENCIA.

Los modelos estimativos de potencia no han experimentado el desarrollo que han tenido los modelos temporales, debido principalmente a que en los circuitos CMOS el consumo de potencia no es un problema tan serio como en los circuitos GaAs.

Existen actualmente pocas alternativas desarrolladas para estimar el consumo de potencia. Una de ellas consiste simplemente en contar el número de transistores presentes en el circuito. Este método es rápido pero altamente impreciso. La alternativa más eficaz, y también la más lenta, se basa en la simulación eléctrica (tipo SPICE).

La determinación correcta vía simulación, pasa por evaluar el consumo de potencia aplicando todas las combinaciones posibles de las señales de excitación a las entradas del circuito. Este método no es práctico para su aplicación en circuitos de grandes dimensiones.

Estas alternativas mencionadas son tan generales que encuentran aplicación en cualquier tecnología (CMOS, GaAs, BiCMOS...). La tendencia actual se dirige hacia formulaciones de tipo estadístico [RoMcC92].

La mayoría de los estimadores de potencia desarrollados para circuitos CMOS [ChoBa90], [DeKrW90], se dirigen al estudio del problema de la fiabilidad de los circuitos, fiabilidad que viene directamente asociada al problema de la electromigración. Los problemas asociados al consumo medio y al aumento de temperatura se revelan como de segundo orden (aunque éste último determina en última instancia la elección del tipo de encapsulado así como las características de funcionamiento del *chip*).

La principal contribución al consumo de potencia en los circuitos GaAs es debida al consumo estático [Cates90]. Dado que el consumo de potencia en los circuitos CMOS es un consumo dinámico la aplicación tanto de los modelos existentes como de las herramientas al caso GaAs es imposible.

De la revisión de la literatura y pese a la gran diferencia entre ambas tecnologías, se ha creído conveniente destacar la metodología propuesta por W. Röthig [RoMeC92]. El método consiste en una formulación estadística para la determinación del consumo de potencia en estructuras regulares en tecnología CMOS. El modelo emplea tablas de datos para almacenar el consumo de potencia en los estados lógicos alto y bajo. Dichos valores se miden con un simulador eléctrico (SPICE).

1.4.- ESTADO DEL ARTE EN MODELOS ESTIMATIVOS DEL ÁREA LOCAL OCUPADA.

Los modelos estimativos de la superficie ocupada por un circuito digital no han experimentado el desarrollo que presentan los modelos precedentes. Ello se debe principalmente a que el problema de la estimación del área está exenta de las dificultades asociadas al comportamiento no lineal que muestran los modelos estimativos para el consumo de potencia y el retardo de propagación.

El problema de estimar el área local de una porción de diseño determinada no es sumamente complejo, su tratamiento suele ser circunstancial dentro de artículos que tratan de temas relacionados con metodologías de diseño.

Así, es frecuente que se detallen modelos estimativos de área global en trabajos relacionados con síntesis de alto nivel ([HerFi89], [Marpl89], [CoBaA91]) y de área local en trabajos relacionados con optimización eléctrica ([HoNneS90], [HedJe87]). Los modelos que se presentan en estos trabajos son de naturaleza sencilla (completamente lineales) y ofrecen por lo general completa transportabilidad de una tecnología a otra. En esta línea desarrollaremos los modelos propuestos en esta tesis.

En otros casos las aportaciones en estimación de área local han sido muy específicas para estructuras importantes, como son las memorias (c.f.r. [MuQuF91]).

El problema de la estimación o modelado de área ocupada, especialmente en estructuras regulares, es distinto del problema de optimización topológica de esas estructuras, profusamente estudiado en la realización de herramientas de compactación, escalado de módulos y compilación de estructuras. En este segundo caso la validación de herramientas se ha hecho tradicionalmente midiendo el área generada.

Por otro lado el modelado que aquí nos interesa es un modelado local de área, frente al problema de área global ocupada, típicamente considerado en herramientas de colocación y ruteado, y en estudio de alternativas arquitecturales. En esta línea puede

verse por ejemplo el trabajo de Da Luz Reis [DaLuR83]. Ésta es una tesis doctoral dedicada al problema de la organización topológica de las diferentes partes funcionales de un microprocesador. Se realiza un estudio de medidas de área de estructuras como *PLAs*, memorias *ROM* así como un estudio comparativo, desde el punto de vista de superficie ocupada, de diferentes microprocesadores tales como (18085, 18748, MC6800, MC6809, Z80 y Z8000).

1.5.- PLANTEAMIENTO DEL PROBLEMA.

El objetivo de todo proceso de diseño consiste en pasar de unas especificaciones iniciales, definidas en muy alto nivel, a su correspondiente realización física (*layout*). En el ámbito GaAs, el número de factores que afectan al diseño es mucho mayor que lo que acontece para el caso CMOS y el estudio de viabilidad muestra mayor complejidad. De especial relevancia es la consideración del ruido (tanto dinámico como estático) durante el diseño de circuitos GaAs. Es ésta una circunstancia que, a diferencia de lo que acontece en circuitos CMOS, se ha tenido en cuenta ya desde los primeros diseños GaAs.

Una vez revisada la literatura disponible, y tras estudios de viabilidad se concluyó que una estrategia de optimización de tipo analítico es el método adecuado para llevar a cabo la optimización de diseños DCFL. Los optimizadores sustentados sobre metodologías de tipo analítico son siempre más poderosos que los heurísticos y el conjunto de soluciones obtenido es más transparente de cara al usuario.

Si bien las metodologías de tipo heurístico existentes para los circuitos MOS pueden ser aplicadas al caso GaAs (tras las modificaciones pertinentes), los problemas relativos a su posible no convergencia invitan a decantarse por estrategias de tipo analítico. Además, la literatura actual se centra en el estudio de las técnicas analíticas, habiendo quedadas relegadas a un segundo plano las metodologías de tipo heurístico.

Los optimizadores basados en procedimientos analíticos disponibles para los circuitos CMOS no pueden aplicarse directamente a los circuitos GaAs. Tanto los modelos estimativos del retardo de propagación como los modelos estimativos de potencia difieren de forma notable. Asimismo las técnicas de "bufeización" desarrolladas para circuitos MOS no encuentran aplicación al caso GaAs.

Habitualmente, los diseñadores contrastan los diseños frente a simulaciones eléctricas con SPICE dado que este simulador eléctrico ofrece precisión y versatilidad. Los resultados obtenidos (potencia consumida y retardo de propagación) tras la

optimización de un diseño suelen contrastarse con los resultados de SPICE. Éste es también el criterio adoptado durante el desarrollo de esta tesis.

Los modelos para la estimación del área ocupada sí pueden adaptarse al caso GaAs introduciendo factores de ajuste que den cuenta de los estilos de diseño -a nivel físico- mas usuales en los circuitos DCFL.

El objetivo del presente trabajo es desarrollar una estrategia de optimización para circuitos *full-custom* combinatoriales GaAs (lógica DCFL) con la intención de llenar el vacío existente actual. La optimización se encauza principalmente hacia la mejora del compromiso definido por el retardo de propagación, el consumo de potencia y la superficie ocupada.

La estrategia de optimización que se plantea se desarrolla en dos fases claramente diferenciadas, cada una de ellas objeto de automatización. La primera fase explora las distintas posibilidades de "buférización" para los circuitos bajo estudio. En este sentido, se pretende definir una técnica de "buférización" que permita el diseño con un alto grado de fiabilidad. La segunda fase estudia las posibilidades que encierra el dimensionamiento de los transistores. El problema del dimensionado (*sizing*) se resuelve mediante técnicas de programación no lineal. Como resultado de la aplicación de la estrategia de optimización se ofrece al diseñador ya sea la solución de compromiso entre las funciones de coste, óptima para la realización de su diseño (con el conocimiento de cada una de las dimensiones de los transistores), como un margen en torno a ese óptimo para su toma de decisiones.

CAPÍTULO 2

ESTUDIO DE LA FAMILIA LÓGICA DCFL.

En este capítulo se presenta la familia lógica directamente acoplada sobre la cual se implementan los circuitos que son objeto de estudio en esta tesis. Se presta un interés especial al estudio del margen de ruido. De este análisis se desprende que para mejorar la inmunidad al ruido es preciso añadir un seguidor de fuente o *buffer* a la salida de la puerta DCFL. La puerta lógica DCFL con el seguidor de fuente en su nodo de salida se la conoce con el nombre de puerta SDCFL. Las características de esta puerta son también estudiadas en este capítulo.

2.1.- INTRODUCCIÓN.

En los últimos años [Eshra90] se han desarrollado distintos tipos de dispositivos en GaAs. Hasta el momento puede hablarse de al menos dos generaciones. La primera incluye los siguientes:

- DMESFET, transistor de efecto campo metal semiconductor en modo de depleción,
- EMESFET, transistor de efecto campo metal semiconductor en modo de enriquecimiento,
- EJFET, transistor de efecto campo de unión en modo de enriquecimiento,
- CEJFET, transistor de efecto campo de unión complementario.

Con esta primera generación de dispositivos se alcanzan retrasos de conmutación de 70 a 80 ps con una potencia disipada del 0.5 a 1.5 mW.

La segunda generación está constituida por dispositivos más sofisticados del tipo de los siguientes:

- MODFET ó HEMT, transistor de alta movilidad de electrones,
- HBT, transistor bipolar de heterounión.

En éstos, la movilidad de los electrones es 5 veces superior a la movilidad de los dispositivos de la primera generación.

El problema de diseño de un circuito GaAs está menos restringido, en general, que el MOS de Si. La razón es la escasa tipificación de esta lógica, esto es, no hay niveles lógicos ni tensiones de alimentación estándar como en aquella. Las familias pueden pertenecer a una de las dos categorías básicas siguientes:

- lógica normalmente en conducción y
- lógica normalmente en corte.

La lógica normalmente en conducción se implementa con transistores de depleción ($V_T < 0$) exclusivamente mientras que la normalmente en corte utiliza ambos tipos de transistores.

Se considerará, en este capítulo, que son dos las familias lógicas objeto de análisis,

la DCFL (*Direct Coupled FET Logic*) y una "descendiente" de la misma, la SDCFL (*Source Follower Direct Coupled FET Logic*). Estas son lógicas normalmente en corte, esto es, se implementan con transistores de enriquecimiento y depleción. En particular, se utiliza el MESFET como dispositivo base, si bien esta familia se utiliza también con MODFETs [CirAb83] y ha sido incluso propuesta con JFETs, [Shur87]. Estas familias, que se pueden englobar bajo la denominación DCFL/SDCFL, son probablemente las más extendida en el campo del diseño digital sobre GaAs.

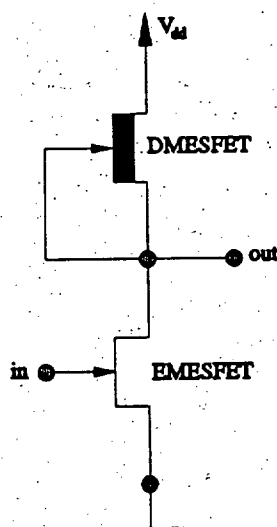


Fig. 2.1. Inversor DCFL.

2.1.1.- Familia lógica directamente acoplada (DCFL).

Esta familia es topológicamente muy similar a la E/D MOS de Si. Es la primera y la más simple de las utilizadas en GaAs. El inversor básico DCFL está constituido por un transistor de enriquecimiento actuando como dispositivo que conmuta y uno de depleción actuando como carga activa (figura 2.1).

Las ventajas de esta familia son la simplicidad del circuito y el escaso número de elementos por puerta. Esto se traduce en que los parásitos asociados a interconexiones son pequeños y en mayor densidad de integración, menor potencia disipada y mayor velocidad. La principal desventaja es el pequeño valor de la excursión lógica y por tanto de los márgenes de ruido.

Cuando se excita el terminal de entrada con una tensión correspondiente al nivel lógico

alto, el transistor que conmuta (EMESFET) se encuentra en conducción y el nudo de salida se halla en nivel lógico bajo. Si a la entrada el nivel de tensión es bajo, -inferior a la tensión de umbral-, el transistor se halla en estado de corte y la salida es alta.

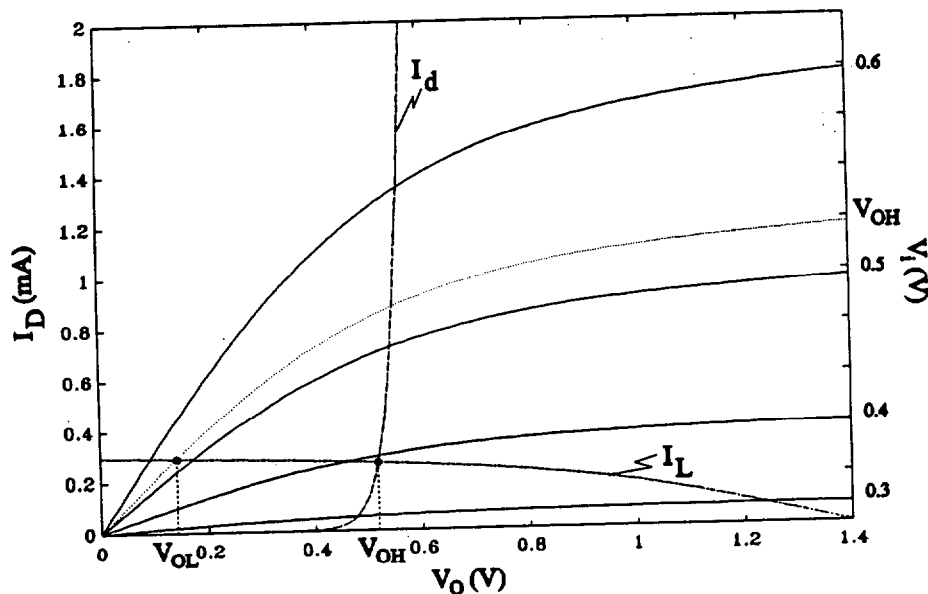


Fig. 2.2. Funcionamiento cualitativo del inversor DCFL.

La excursión lógica de esta familia está limitada por consideraciones cualitativas. De una parte, el nivel bajo de tensión debe ser inferior a la tensión de corte del transistor que conmuta $V_{OL} < V_T$ para que efectivamente esté cortado. Por otra parte, el nivel alto debe ser también inferior al potencial de contacto de la unión metal-semiconductor de puerta, $V_{OH} < V_{BI}$ pues está limitado por la tensión de conducción de esa unión metal-semiconductor en la puerta de control. Se concluye pues que la excursión lógica será aproximadamente la diferencia entre ambas magnitudes, $\Delta V \approx V_{BI} - V_T$. Se discute en lo que sigue los valores de tensión con mayor detalle.

En la figura 2.2 se muestra la característica de salida del transistor, $I_d = I_d(V_{GS}, V_{DS})$, particularizada a los intervalos de interés, $V_{GS} = V_i \in [V_T, 0.6]$ V y $V_{DS} = V_o \in [0, V_{DD}]$ V. Como se observa en la figura, el transistor es de enriquecimiento con V_T próxima a 0.2 V y V_{DD} es 1.4 V. La corriente que circula por el transistor que actúa como carga se denota por I_L .

En lo que sigue se considera el balance de corrientes en el nudo de salida. Con el inversor descargado (nudo de salida al aire), el nivel alto de tensión alcanzado a la salida será aproximadamente V_{DD} (en la gráfica se consideran nulas las resistencias serie). Esta situación corresponde a una tensión de entrada inferior a la de umbral, $V_i \leq V_T$. Al excitar el inversor

con un valor de tensión alto, el transistor que conmuta opera en zona lineal y el nudo de salida alcanzará un valor algo superior a 0 V (no se muestra la curva $I_d(1.4, V_o)$).

Esta no es la situación real. Se incluye en la figura 2.2 la curva característica del diodo Schottky (I_d) asociado a la puerta del transistor que conmuta. La presencia del diodo limita el valor de V_{OH} al de conducción de la unión rectificadora metal semiconductor. En el caso mostrado $V_{OH} \approx 0.53$ V. El valor del nivel de tensión bajo se obtiene de la intersección de la curva de carga (I_d) con la característica del transistor para $V_{GS} = V_i = V_{OH}$ (que aparece en línea punteada). Se obtiene $V_{OL} \approx 0.16$ V.

Se concluye que la unión tipo Schottky asociada a la puerta del transistor que conmuta limita el nivel lógico alto a unos 0.6 V correspondientes a la tensión de conducción del mismo. Esto es así con independencia del valor de la tensión de alimentación. El nivel de tensión bajo está limitado por la caída óhmica del canal cuando el MESFET se halla en conducción (típicamente 0.1 V).

De modo que la excursión lógica del inversor se reduce considerablemente, pasando a ser de aproximadamente 1.4 V al valor $0.53 - 0.16 = 0.37$ V. Dado el pequeño valor de la excursión lógica, se exige un gran control en el proceso de fabricación sobre la tensión de umbral del dispositivo, V_T . Lo mismo se puede decir respecto a los márgenes de ruido.

El margen de ruido es además muy sensible a las variaciones de la relación de aspecto del inversor y a las de la tensión de umbral. Por un lado, la intensidad disponible para descargar el nudo de salida será mayor cuanto mayor sea β (definida como el cociente W_E/W_D , donde W es el área del transistor y los subíndices hacen referencia a su modo de operación), por tanto, el valor de V_{OL} puede hacerse menor aumentando β . Con esto, se produce un aumento en la excursión lógica y por ende en el margen de ruido. También es posible conseguir un aumento en el nivel alto de tensión, V_{OH} , mediante el control de la tensión de umbral del transistor. La tensión de umbral del inversor, V_{TH} , puede aumentarse haciendo mayor la tensión de umbral del dispositivo, V_T .

El coste de estos aumentos de la excursión lógica es un mayor retraso de propagación. En efecto, si la relación de aspecto se hace mayor disminuyendo W_D , la corriente disponible para cargar el nudo de salida será menor. Esto conlleva un aumento del tiempo de carga del nudo de salida y el tiempo de propagación T_{PLH} mayor. Por otro lado, si V_T es grande, la diferencia $V_{GS} - V_T$ será menor que en caso contrario. Con ello se dispone de un menor nivel de corriente de descarga del nudo de salida (menor intensidad de saturación). El tiempo de propagación T_{PHL} aumenta.

El pequeño valor del margen de ruido hace inviable la implementación de puertas NAND conectando en serie transistores. Dado el pequeño valor, relativo al transistor MOS, de la resistencia en corte del MESFET, en el estado de corte puede circular una intensidad de corriente de fugas apreciable [Eshra90]. Así, al aumentar el número de transistores que conmutan, el '1' lógico se degrada. Así, las puertas de múltiples entradas disponibles son exclusivamente NOR. Se obtienen asociando en paralelo dos o más transistores de enriquecimiento.

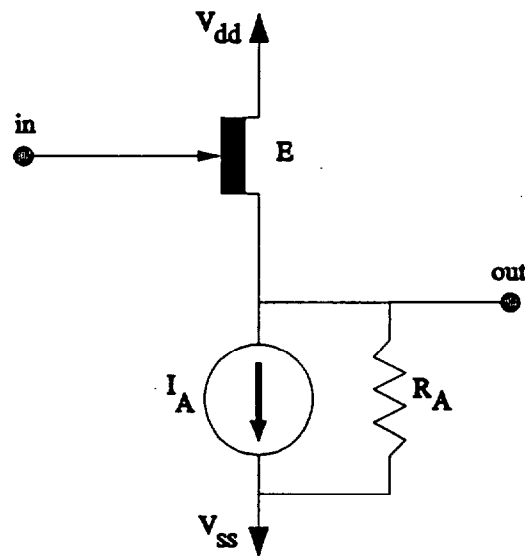


Fig. 2.3. Circuito seguidor de fuente.

2.1.2.- Familia con seguidor de fuente (SDCFL).

En los diseños de circuitos lógicos es habitual encontrar que la capacidad de carga a las puertas puede variar en un amplio rango dependiendo de los requerimientos del abanico de salida (*fan-out*) y la longitud de las líneas de interconexión entre los nudos de salida y la carga. En general, para disminuir el tiempo de propagación será preciso aumentar la corriente disponible para cargar y descargar la capacidad de carga.

Una posible solución circuital consiste en utilizar el seguidor de fuente de la figura 2.3. Este circuito presenta una gran ganancia en corriente y una pequeña ganancia en tensión ($A_v < 1$). El transistor de enriquecimiento (E), introduce un alto nivel de intensidad en el nudo de salida durante el proceso de carga. La descarga se realiza a través de la resistencia R_A y de la fuente I_A . Esta parte de descarga se implementa con un transistor de deplexión actuando como carga activa ($V_{GS}=0$ V).

La operación del seguidor de fuente se expone con la ayuda de la figura 2.4. En ella, la línea de carga (I_P) representa al transistor DMESFET.

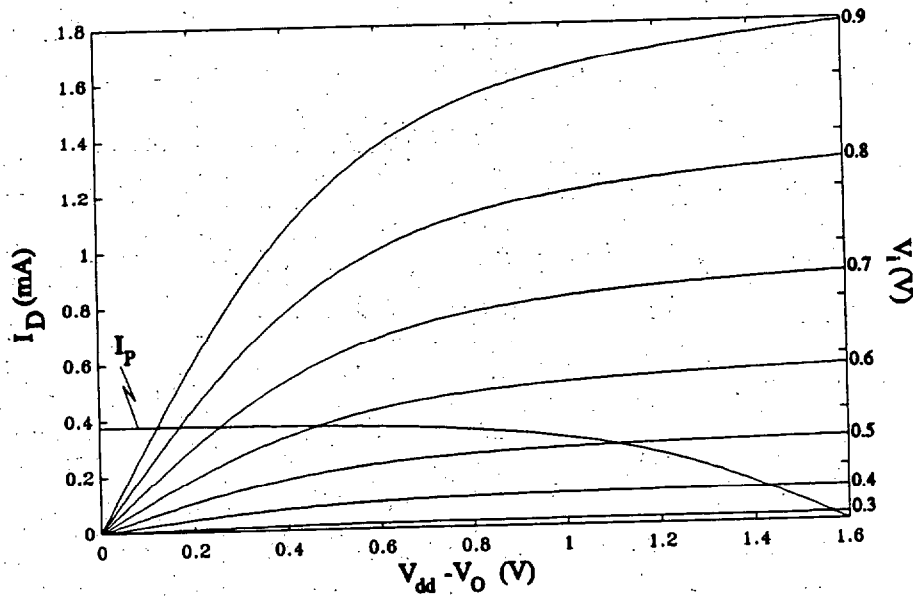


Fig. 2.4. Operación del seguidor de fuente.

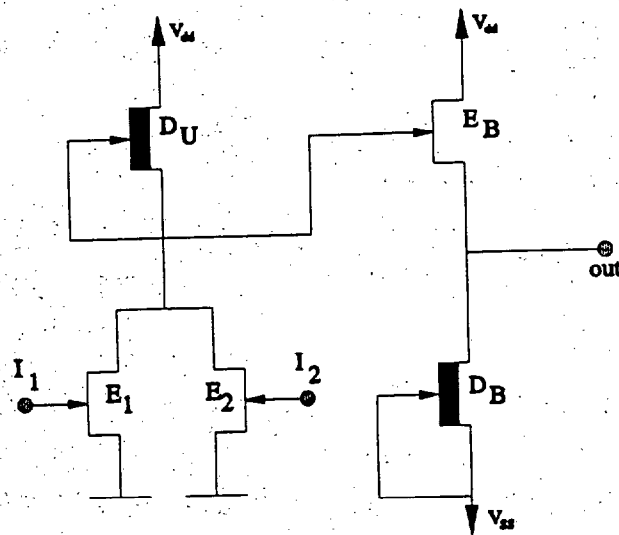


Fig. 2.5. NOR2 SDCFL.

Las intersecciones de la línea de carga con las curvas de salida del transistor de enriquecimiento dan los puntos de operación del circuito. Nótese cómo con E operando en saturación de las características, ante una variación relativamente pequeña de la tensión de entrada, V_i , la de salida varía en un rango amplio, $V_o = V_{DD} - V_{DS}$. De manera opuesta, fuera

de la zona de operación de saturación de las características, la tensión de salida cambiará poco ante grandes cambios de V_{DS} . En el diseño, por tanto, será tal que E opere en saturación. El modo de operación en zona óhmica puede evitarse con una elección adecuada de las tensiones de polarización.

Tabla 2.1. Conjunto de puertas típicas y sus dimensiones.

Tipo de puerta	Anchura del transistor D (μm)	Anchura del transistor E (μm)
Inversor	4	16
NOR 2	4	12
Buffer 1	4	6
Buffer 2	6	9.2

Con lo anterior, se deduce que la impedancia de entrada de un seguidor de fuente es grande ya que V_i permanece prácticamente constante. La capacidad C_{GD} del transistor de enriquecimiento es pequeña dada la operación en saturación. Por otro lado, la capacidad efectiva en pequeña señal es también pequeña, - está dada por $C_i = C_{GD} + C_{GS}(1-A_v)$ -, lo cual es beneficioso pues la carga del nudo interno a la puerta (figura 2.5) se hace mínima. La corriente de entrada será también pequeña si V_i es inferior a la tensión de conducción de la unión metal semiconductor asociada a la puerta del transistor. Con esto, la ganancia en corriente es muy grande. Por último, se pueden soportar cargas mayores que con una configuración en fuente común. Esto es así porque la conductancia de salida de esta etapa es mucho menor que la de la etapa en fuente común, de hecho, tiende al valor $\frac{1}{g_m}$.

La familia SDCFL (*Source follower Direct Coupled FET Logic*) se implementa a partir de la anterior (figura 2.1), añadiendo un seguidor de fuente como etapa de salida (figura 2.5).

El terminal de fuente se conecta a un raíl de tensión ligeramente negativa ($V_{SS} \approx -0.2\text{V}$). La lógica resulta ser más rápida. Por otro lado, se pueden usar transistores de

Tabla 2.2. Variación del nivel de tensión de salida con la carga en DCFL.

<i>fan-out</i>	1	2	3	4	5
V_{OH} (V)	0.5287	0.5060	4.4940	0.4858	0.4796

conmutación más estrechos que en la familia DCFL gracias a que la parte que genera la función lógica es bien seguida desde la capacidad de carga por el seguidor de fuente. La contrapartida es que disipa más que la anterior y los niveles de integración alcanzables son inferiores.

Para esta familia es aplicable lo expuesto para el caso DCFL respecto a la degradación de los niveles lógicos. De modo que también en este caso existe una limitación respecto del máximo *fan-out* admisible. Su valor típico es 3. En relación con las dimensiones, se puede establecer un cierto conjunto más o menos restringido de puertas. En la tabla 2.1 se muestra un conjunto de dimensiones típicas de puertas SDCFL. En el caso de puertas NOR, se asocian en paralelo un máximo de 3 de las mismas dimensiones. Esto es, el máximo *fan-in* es 3 para preservar el margen de ruido cuando todos los transistores que conmutan se encuentren en estado de conducción.

Con esto, aunque el diseño del circuito sea completamente a medida (*full-custom*) en la práctica se utiliza un conjunto restringido de puertas. Aún así, se pueden efectuar pequeñas variaciones en la dimensión de las puertas para optimizar los diseños. En relación con las dimensiones de los seguidores de fuente (*buffers*), suelen utilizarse dos clases cuyas dimensiones se presentan en las filas cuarta y quinta de la tabla. De entre ambas, la que mejores prestaciones ofrece en cuanto a la velocidad es la de la cuarta fila, esto es, $W_{BE}=9.2 \mu\text{m}$, $W_{ED}=6 \mu\text{m}$. Ésta se implementa fundamentalmente en los inversores. El otro dimensionamiento es típico de las puertas de múltiples entradas. En general, los diseños en DCFL/SDCFL son muy uniformes en el sentido de la variedad de puertas y las cargas.

2.2.- ANÁLISIS.

Se atenderá en este punto a la descripción cuantitativa de las dos familias lógicas presentadas. En una primera etapa, se presentan las características en DC para después pasar al funcionamiento en régimen dinámico. Como conclusiones, se imponen algunas restricciones al conjunto de puertas de las familias.

2.2.1.- Familia DCFL.

Al igual que ocurre en la lógica NMOS, a la que se parece, la familia DCFL es también una lógica con "ratios" (*ratioed logic*) cuyas transiciones y niveles dependen de proporciones de la geometría de los dispositivos. En la lógica DCFL el nivel bajo de tensión a la salida de las

puertas depende de las dimensiones de la mismas. Por otro lado, el nivel alto depende significativamente de la carga (*fan-out*). Esta es una diferencia importante con respecto al caso MOS. En las tablas 2.2 y 2.3 se muestran resultados de simulación SPICE. En la primera se dan los niveles altos de tensión alcanzados al incrementar la carga a un inversor. A medida que aumenta, se obtienen valores más pequeños de esta tensión.

La justificación cualitativa está en que la curva asociada al diodo de la figura 2.2 es más abrupta cuanto mayor sea la carga a la puerta. La máxima desviación calculada relativa a la situación de *fan-out* 1 es del -9.3%. Este hecho obliga a imponer limitaciones en la carga. Típicamente se utiliza 3 como *fan-out* máximo.

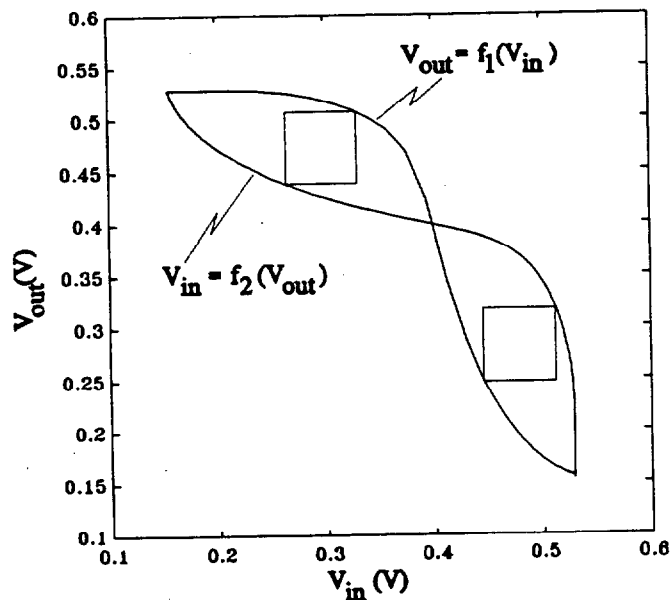


Fig. 2.6. Característica de transferencia en DC del inversor DCFL.

La tabla 2.3 presenta la variación del nivel bajo con la anchura del transistor que conmuta. Puede observarse cómo disminuye la tensión de salida al aumentar las dimensiones del transistor. Este hecho ya ha sido comentado en el párrafo precedente. La mayor variación relativa a una anchura de $22 \mu\text{m}$ de entre las presentadas es del -36.6%. Con objeto de asegurar un margen de ruido aceptable, es típica la dimensión $W=30\mu\text{m}$.

En la figura 2.6 puede verse la curva característica de transferencia del inversor DCFL de relación de aspecto $\beta = W_E/W_D = 30/4$ ($f_1(V_{in})$). En la misma, se incluye la curva vista desde la salida por un inversor idéntico ($f_2(V_{out})$) para estimar los márgenes de ruido por el método del cuadrado máximo. El valor del margen de ruido alto es: $NM_H \approx 0.17 \text{ V}$, el bajo resulta ser ligeramente inferior. La excursión lógica tiene un valor de 0.3727 V .

Tabla 2.3. Variación del nivel bajo de tensión de salida con la anchura en DCFL.

W (μm)	22	26	30	34	38
V _{OL} (V)	0.2028	0.1759	0.1560	0.1407	0.1286

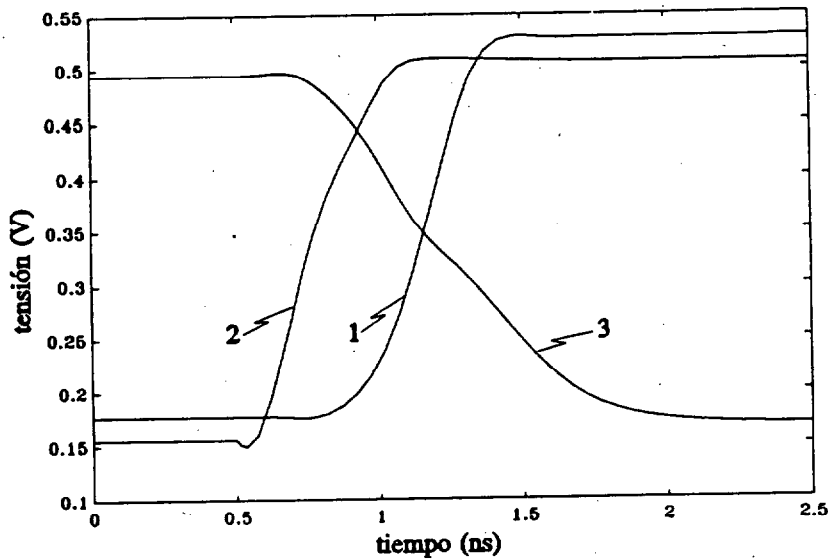


Fig. 2.7. Variación de los niveles de tensión a la salida.

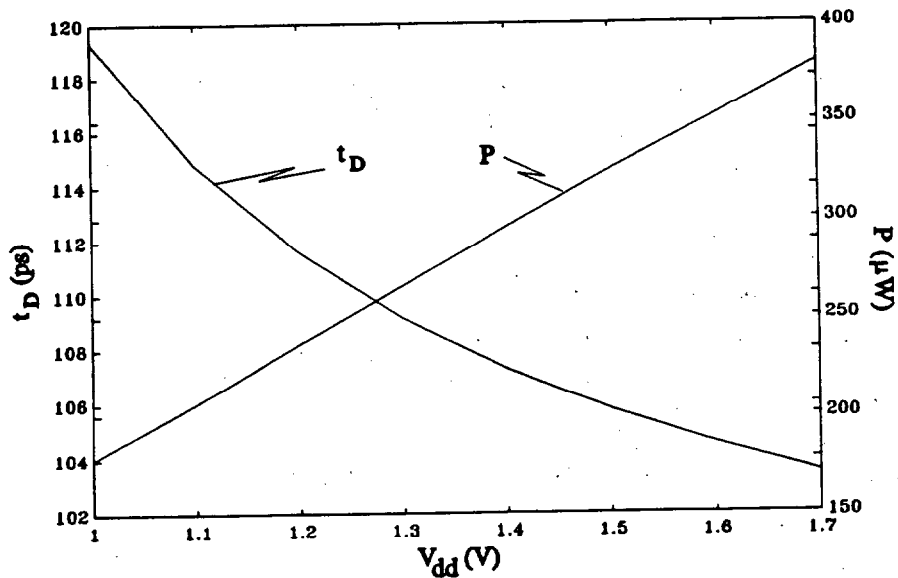


Fig. 2.8. Retraso y potencia disipada por puerta frente a la tensión de alimentación.

Para mostrar la forma de las señales de salida de esta familia se incluye la figura 2.7. En ella pueden apreciarse distintas señales conmutando correspondientes a una cadena de

inversores en distintas situaciones de carga. El *fan-out* es 1, 2 y 3 (como indican los números) para cada caso. En general, para las relaciones de aspecto típicas, la transición de carga del nudo de salida resulta ser más lenta que la contraria. Esto podría corregirse disminuyendo la relación de aspecto. El coste es un aumento de la potencia disipada y la ya comentada disminución del margen de ruido. Por otra parte, se aprecia la influencia de la carga sobre la respuesta transitoria del inversor. Para *fan-out* 3, la forma de onda de salida es mucho más lenta que la correspondiente a una carga inferior. La carga degrada significativamente la respuesta transitoria de la familia, la implicación es una acotación de la carga que puede ser tolerada.

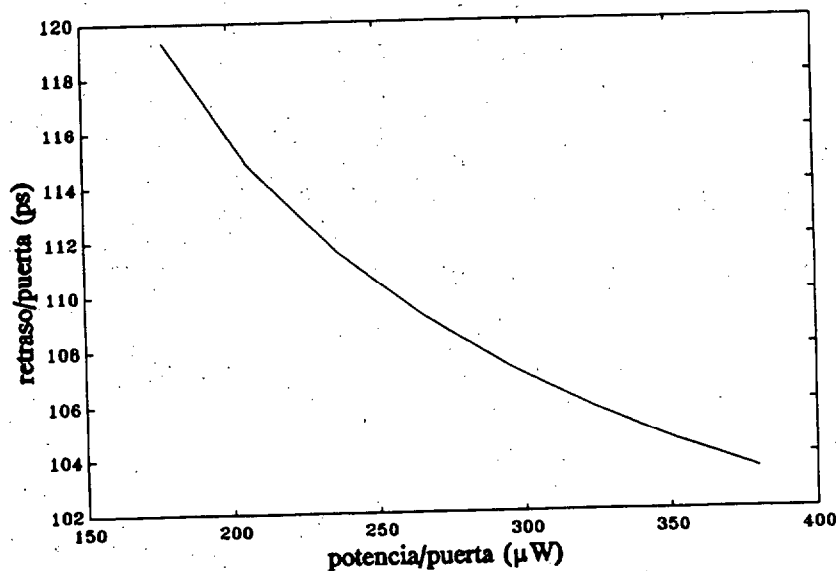


Fig. 2.9. Retraso de propagación frente a la potencia disipada por el inversor DCFL.

La figura 2.8 ha sido obtenida mediante simulación de una cadena de 5 inversores en anillo. En ella se da idea de las prestaciones de la familia en el compromiso retardo-potencia disipada. V_{DD} denota el valor de la tensión de alimentación, P es la potencia disipada por puerta y t_D representa el retardo de propagación medido.

La tensión de alimentación (V_{DD}) de los inversores se ha hecho variar en el intervalo (1,1.8) V. El retardo medido varía en el intervalo (103.47,119.33) ps y la potencia disipada

Tabla 2.4. Variación del nivel alto de tensión de salida con la carga en SDCFL.

<i>fan-out</i>	1	2	3	4	5	6
V_{OH} (V)	0.5382	0.5187	0.5080	0.5006	0.4949	0.4903

por puerta en (178.02,380.01) μW .

Como era de esperar, la potencia crece con la tensión de polarización mientras que, por su parte, el retraso obedece a una relación inversa. A mayor valor de la tensión de alimentación, el retraso de cada puerta se hace menor. Esto es así ya que al aumentar V_{DD} , las corrientes de drenador de los transistores aumentan, con ello, la carga asociada a los nudos de salida puede ser alojada y desalojada en menos tiempo pues los niveles lógicos no varían significativamente. A su vez, el producto tensión-intensidad se hace también mayor. La figura 2.9 muestra la dependencia directa entre ambas magnitudes.

El valor de la tensión de alimentación usual es de 1.4 V aproximadamente. La tipificación de los valores de estas magnitudes es muy escasa, si se compara con el caso MOS, lo cual dificulta en gran medida la tarea de realizar análisis globales para todos los grupos de diseño. Este valor de compromiso arroja un retraso de propagación por puerta de 107.27 ps con una potencia disipada de 295.93 μW .

2.2.2.- Familia SDCFL.

El hecho de que esta familia sea descendiente de la anterior hace que arrastre algunas de sus propiedades. En particular, el valor de los márgenes de ruido sigue siendo pequeño. Respecto a las dependencias de los niveles de tensión se pueden seguir los razonamientos expuestos para la familia DCFL ya que el seguidor de emisor, idealmente tan solo los desplazará. Con esto, esta familia es también *ratioed-logic*. La tabla 2.4 (análoga a la tabla 2.2) muestra cómo varía el nivel lógico alto con la carga del nudo de salida de la puerta. Las cargas recorren el intervalo [1, 6] a efectos exclusivamente ilustrativos ya que, como se ha dicho, el *fan-out* máximo típico es 3. Se produce una degradación del nivel lógico alto relativa al mejor caso con carga mínima del -8.9% (la disminución absoluta es de 47.9 mV, que teniendo presente los márgenes de ruido resulta ser muy importante, el 43.5% del margen de ruido).

La influencia de la dimensión del transistor que conmuta sobre la tensión correspondiente al nivel bajo se refleja en la tabla 2.5. Cuanto mayor es la anchura del referido transistor, mejor es el nivel de tensión estacionario, esto es, más pequeño resulta. Los valores negativos se alcanzan gracias a la polarización negativa aplicada al terminal de fuente del transistor que actúa como carga activa del seguidor de fuente, -0.2 V. Cualitativamente, la dependencia es la misma a la obtenida en el caso DCFL, la disminución relativa al peor de los casos presentados es del -114.6%. Como queda reflejado en la tabla 2.1, las dimensiones típicas para la anchura del transistor responsable de la conmutación es de 12 ó 16 μm . De nuevo la elección se realiza atendiendo al compromiso márgenes de ruido-

Tabla 2.5. Variación del nivel bajo de tensión de salida con la anchura en SDCFL.

W (μm)	8	12	16	20	24	28
V_{OL} (V)	0.098	-0.024	-0.080	-0.111	-0.130	-0.143

retraso de propagación.

La figura 2.10 muestra la curva de transferencia del inversor de la familia lógica. Como en la figura 2.6, se representa en la misma gráfica la característica vista por un inversor idéntico a la salida.

Si se comparan ambas curvas, se nota cómo la excursión lógica en este caso ha aumentado hasta un valor aproximado de 0.6 V frente al medido para la familia DCFL de valor 0.3727 V. Esto es, se duplica el valor de la excursión lógica. Sin embargo, los márgenes de ruido no mejoran. El margen de ruido en nivel alto, obtenido de la figura es $NM_H=0.11$ V; para el nivel bajo se tiene, $NM_L=0.10$ V.

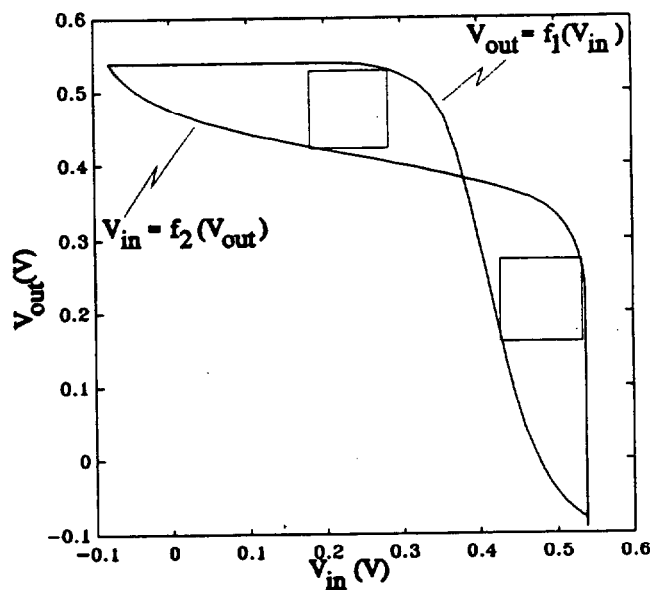


Fig. 2.10. Característica de transferencia en DC de un inversor SDCFL.

En las figuras 2.11 y 2.12 (análogas a la figura 2.8) se muestran los resultados obtenidos por simulación de una cadena de inversores SDCFL haciendo variar las tensiones de alimentación V_{DD} y V_{SS} respectivamente. La dependencia funcional es distinta en ambos casos. El intervalo de variación con V_{DD} de ambas magnitudes es $t_D \in [95.69, 97.97]$ ps y $P \in [224.77, 627.62]$ μW .

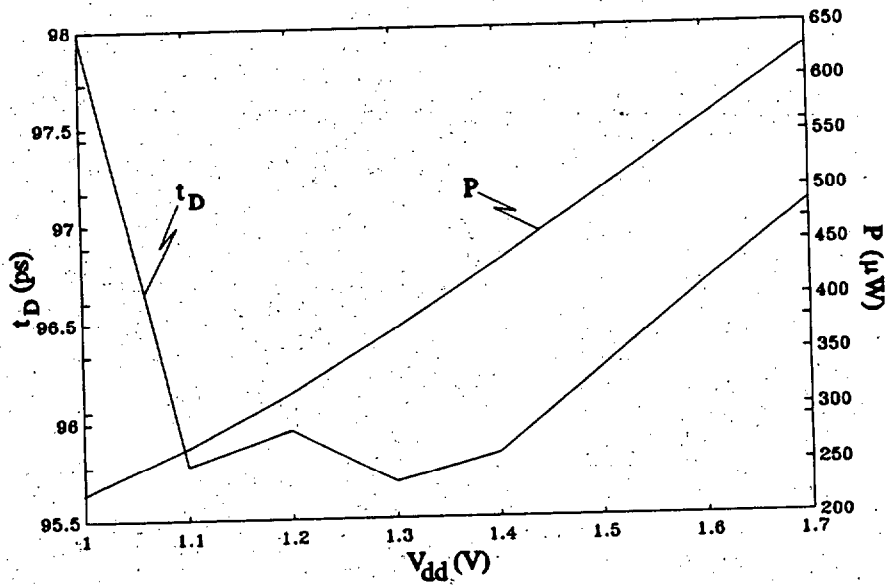


Fig. 2.11. Retraso y potencia disipada por puerta frente a la tensión V_{dr}

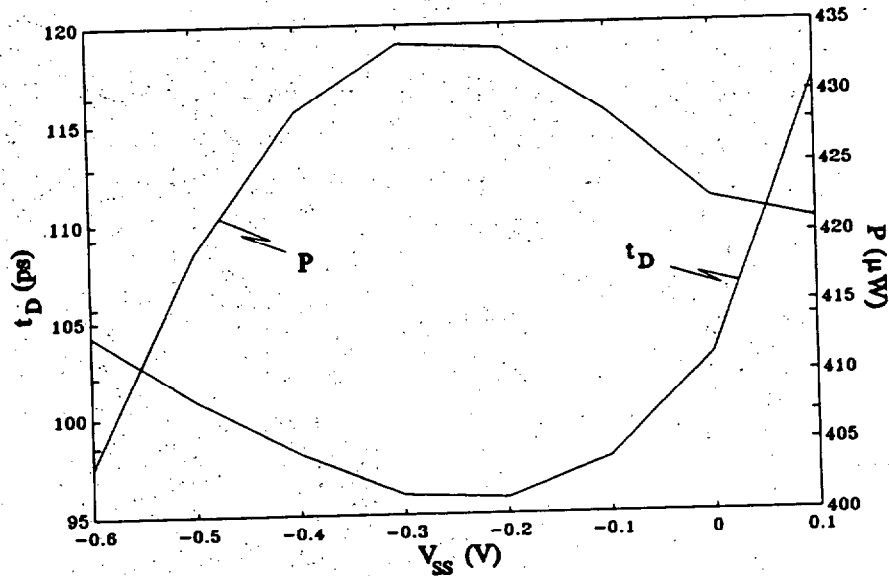


Fig. 2.12. Retraso y potencia disipada por puerta frente a la tensión de alimentación V_{dr}

Si es la tensión de polarización negativa, V_{SS} , la que varía, $t_D \in [95.83, 117.11]$ ps y $P \in [403.48, 433.65]$ μW . La elección de los valores de las tensiones es de nuevo un compromiso entre la terna retraso-potencia-área. Típicamente sus valores son los ya indicados, $V_{DD}=1.4$ V y $V_{SS}=-0.2$ V.

Comparando los valores de la potencia disipada de la familia objeto de análisis con los obtenidos para la DCFL se observa un mayor consumo de la lógica con seguidor de fuente. En términos relativos, la potencia disipada aumenta en un 31.7 %.

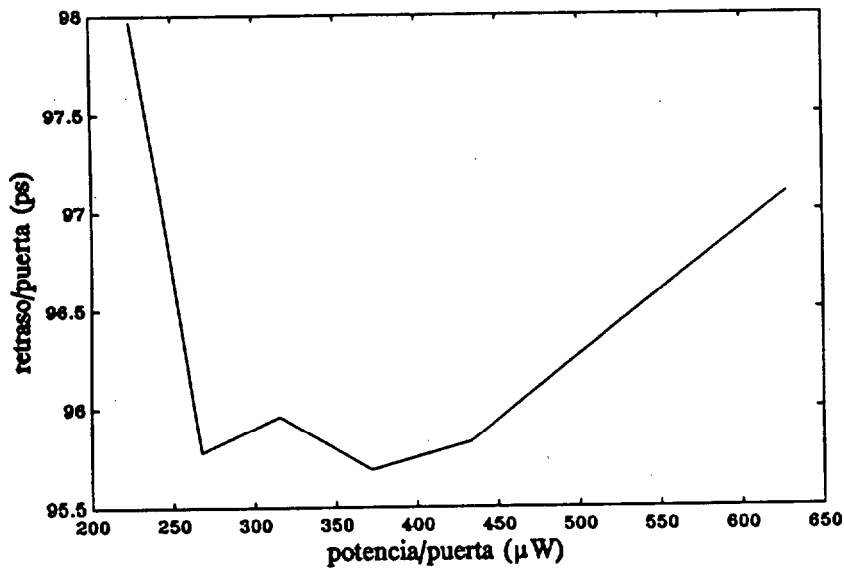


Fig. 2.13. Retraso de propagación frente a la potencia disipada del inversor SDCFL.

Este hecho se mencionó anteriormente, se debe al aumento del número de dispositivos por puerta. Al tiempo, se observan retrasos de propagación menores. La mejora en el retraso es del 12% respecto a la familia DCFL. De modo que por simulación se justifica la presencia del seguidor de fuente en la lógica SDCFL.

La figura 2.13 se incluye a modo de resumen de las prestaciones de la familia en el compromiso retraso de propagación-potencia disipada.

2.3.- CONCLUSIONES.

Se concluye este capítulo dedicado al estudio de la familia lógica DCFL. Hemos dedicado una atención especial al análisis del ruido. Los resultados presentados revelan la necesidad de poner un *buffer* a la salida de la puerta DCFL. A su vez, las características del *buffer* de geometría 9.2/6 dotan a la puerta DCFL "bufferizada" -para la tecnología que se estudia- de alta inmunidad al ruido. En el capítulo 4 se hará uso de esta geometría para el seguidor de fuente de cara a desarrollar la metodología de diseño robusto.

CAPÍTULO 3

MODELOS ESTIMATIVOS DE RETARDO DE PROPAGACIÓN, CONSUMO DE POTENCIA Y ÁREA OCUPADA.

Se dedica este capítulo a presentar los modelos estimativos del retardo de propagación, consumo de potencia y área estimada para el caso de circuitos combinatoriales DCFL/SDCFL. Se comienza exponiendo la metodología a seguir para la estimación del retardo. Se sigue con el modelo de potencia y de área; estos dos elaborados expresamente para abordar el problema de la optimización analítica. Dado que el desarrollo del modelo temporal ha sido un paso previo a esta tesis, se procede a resumir los logros y deficiencias de éste así como a su adaptación para incorporarlo a la estrategia de optimización. Los modelos estimativos de potencia consumida y área ocupada son expuestos detalladamente.

3.1.- FORMULACIÓN DE RETARDOS.

Se presenta a continuación el modelo empleado para la estimación de los retardos de propagación en circuitos combinatoriales DCFL/SDCFL. La formulación permite la rápida evaluación de los tiempos de propagación, suministrando además información acerca de la forma de onda de las señales que se propagan. La comparación de los resultados obtenidos frente a los dados por HSPICE se obtiene con un error relativo inferior al diez por ciento. Merced a este modelo se determinan los caminos críticos que posteriormente se optimizarán.

3.1.1.- Formulación e hipótesis de partida.

Se detalla a continuación, de forma resumida, los aspectos más relevantes del modelo temporal.

3.1.1.1.- Definiciones.

La familia lógica considerada se implementa con MESFET's de enriquecimiento y de depleción. Supuesta una longitud de canal fija (que viene dada por la tecnología), el diseñador tiene control sobre la anchura del transistor. Esta anchura constituye la variable de diseño; el valor que toma es función de las prestaciones requeridas.

Se define el parámetro β , relación de aspecto para una puerta lógica de única entrada, como la razón entre las anchuras de los transistores de enriquecimiento y de depleción. Éste es un parámetro geométrico que puede obtenerse directamente del *layout*. Se definirá posteriormente la variable relación de aspecto equivalente para modelar las puertas lógicas de múltiples entradas.

La familia lógica en cuestión se enmarca dentro de las llamadas *ratioed logic*. Quiere esto significar que los valores estacionarios de tensión a la salida de las puertas lógicas varían con las dimensiones de la propia puerta así como con el fanout. Este aspecto se debe a que la puerta del transistor MESFET conduce apreciables valores de corriente. Consecuencia de esto es que la carga asociada al fanout no puede ser modelada con un condensador lineal. Para modelar dicha carga [GoHeN92] se requieren:

- un condensador lineal,
- una fuente de tensión y una resistencia adicionales.

Estos elementos correctivos son necesarios para modelar el fenómeno altamente no

lineal asociado a la puerta del transistor MESFET (diodo Schottky).

Con objeto de fijar los valores de tensión para los que se calcula tanto el retardo de propagación como las propiedades de la señal a propagar, se define la llamada excursión lógica virtual. Se entiende como tal a la diferencia entre el menor valor de la tensión correspondiente al uno lógico (V_{OH}) y el mayor valor de tensión correspondiente al cero lógico (V_{OL}) que pueden ser encontrados en un nudo de salida de una puerta para las posibles combinaciones tanto en las dimensiones de las puertas lógicas como en la carga de las mismas.

Según esta definición, se pueden definir los niveles de tensión V_d , V_c y V_n , como los valores de la excursión lógica virtual medidos al diez, cincuenta y noventa respectivamente (ver figura 3.1).

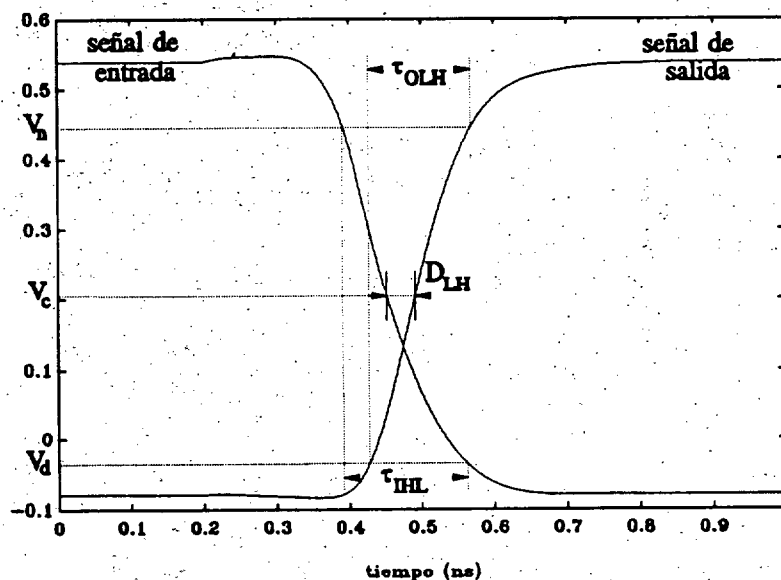


Fig. 3.1. Definición de los parámetros de señal.

Se define tiempo de propagación de una puerta lógica como el intervalo temporal transcurrido desde que la señal de tensión de entrada toma el valor V_c hasta que lo toma la de la salida. Se referencia como retardo D_{LH} si la transición a la salida de la puerta es de subida (nivel lógico bajo a nivel lógico alto) y por D_{HL} en caso contrario. Debido a como se han definido estos tiempos, pueden tomar valor positivo nulo o incluso negativo (ver figura 3.1).

Los parámetros relativos a la forma de onda se definen a continuación. Así, se denota por tiempo de bajada (subida) de la señal de entrada, al intervalo temporal transcurrido desde que la señal de entrada toma el valor V_n (V_d) hasta que toma el valor V_d (V_n). Se referenciará

este valor como τ_{IHL} (τ_{ILH}). Este parámetro guarda relación directa con la forma de onda de entrada por lo que se indica como *pendiente de entrada*.

De forma análoga se define el tiempo de subida (bajada) de la señal de tensión a la salida de la puerta lógica. Dichos valores se referenciarán como τ_{OHL} (τ_{OLH}) y se denominarán *pendientes de salida*. Estos valores deben ser calculados para propagar las pendientes por el circuito. De esta forma, la pendiente de entrada a la puerta $i+1$ se corresponderá con la pendiente de salida de la puerta i . Con este conjunto de definiciones se está en disposición de presentar el polinomio estimativo de retardos y de pendientes (nótese que la pendiente de entrada a la puerta inicial de una cadena dada es siempre de atributos conocidos).

3.1.1.2.- Aproximación polinómica.

El retardo de propagación y la información necesaria acerca de la forma de onda que se propaga por el circuito se obtienen evaluando un polinomio [JuJuP89] en tres variables:

- la relación de aspecto de la puerta lógica (β),
- la carga de la puerta (básicamente capacidades asociadas al cableado y al *fan-out*) y,
- la pendiente de la señal de excitación a la puerta en cuestión.

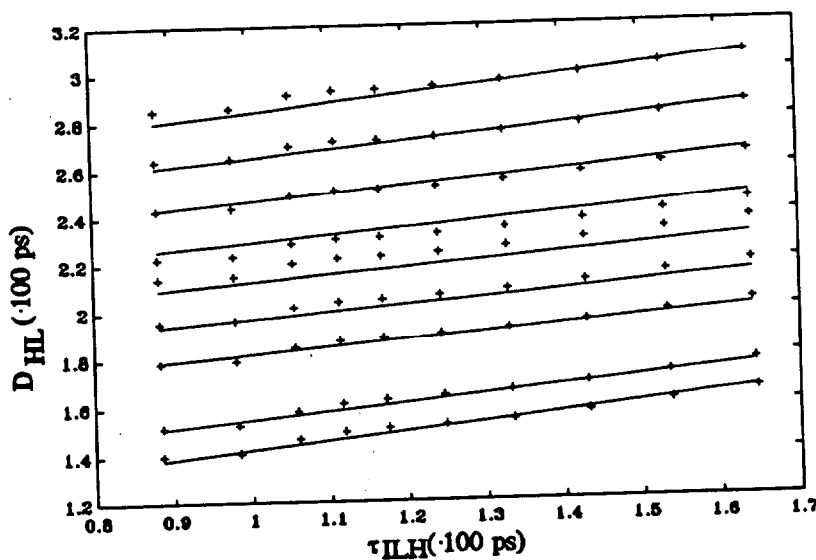


Fig. 3.2. Dependencia del retardo del inversor SDCFL para varias pendientes de excitación y diferentes cargas.

La metodología propuesta [JuJuP89] ha sido adaptada para el caso particular de las familias lógicas DCFL/SDCFL. Consiste en basar el cálculo del retardo de propagación y

de los atributos de las señales en una expresión que da cuenta del hecho experimental que se observa en la figura 3.2.

Dicha figura se ha obtenido mediante simulación con HSPICE. En ella se muestra la variación marcadamente lineal del retardo de propagación (medido en el punto anteriormente indicado) del inversor SDCFL para una relación de aspectos dada al variar la capacidad de carga. Similar resultado se obtiene para la pendiente de salida.

Las condiciones de medida tanto de los tiempos de propagación como de los atributos de las pendientes se eligen de la siguiente forma:

- se dispone una puerta lógica para *filtrar* la señal suministrada por el generador de señal. Típicamente se dispone de una simple puerta inversora,
- se dispone una carga en el nodo terminal en el cual se mide el retardo. Por lo general se ubica en esa posición una puerta inversora.

Cuando se evalúa la expresión polinómica se reproducen las mismas condiciones de medida. Estas condiciones son conocidas como *de filtrado* y se hará uso de esta expresión a lo largo de esta tesis.

Se propone la siguiente expresión para modelar los retardos:

$$D_{HL}(\beta, C_L^{HL}, \tau_{LH}) = \sum_{i=0}^{i=2} \{d_i^{HL} \beta^i + g_i^{HL} \beta^i C_L^{HL} + [h_i^{HL} \beta^i + j_i^{HL} \beta^i C_L^{HL} + k_i^{HL} \beta^i C_L^{HL} C_L^{HL}] \cdot \tau_{LH}\} \quad \{3.1\}$$

$$\beta = \frac{W_{pull-down}}{W_{pull-up}}$$

donde,

- τ_{LH} representa la pendiente de la señal de entrada,
- β es la relación de aspecto del inversor particular y,
- C_L^{HL} es la capacidad de carga (ver definición detallada en 5.4.1.1).

Resulta una expresión similar para el retardo D_{LH} y para las pendientes de las señales que se propagan.

El polinomio a evaluar presenta grado uno en la variable pendiente de excitación y

segundo grado en la relación de aspecto y la carga. Los coeficientes del polinomio se calculan mediante un procedimiento de ajuste por mínimos cuadrados de forma que aproxime la expresión del mismo a los resultados obtenidos por HSPICE en cierto número de simulaciones variando los parámetros τ_{HL} , β y C_L^{HL} . Esta fase se conoce como precaracterización de la lógica en cuestión.

Mediante esta expresión polinómica, seleccionando adecuadamente los coeficientes requeridos, se evalúan los dos retardos (D_{HL} y D_{LH}) y las dos pendientes (τ_{HL} y τ_{LH}).

3.1.1.3.- Precaracterización.

Consiste ésta en la realización de múltiples simulaciones con HSPICE con objeto de generar grandes tablas de datos que, posteriormente son procesadas para obtener los coeficientes del polinomio.

La actual tendencia en la uniformidad en los diseños *VLSI* confiere sustento a la metodología, reduciendo el número de posibilidades y haciendo viable el uso de las tablas de coeficientes. En el caso particular que concierne, se emplea un estilo de diseño altamente regular, con un conjunto reducido de dimensionado de transistores. El *fan-out* nunca es superior a tres. Ello permite acotar eficientemente los valores de las variables de la formulación.

Así, se precisa tan solo del orden de 1200 simulaciones HSPICE para caracterizar completamente la familia SDCFL. El proceso de caracterización se encuentra completamente automatizado y emplea menos de cuatro horas (en una estación de trabajo SUN). Dicho proceso requiere el fichero tecnológico así como los valores de tensión de las fuentes de alimentación. Los valores de capacidad linealizada y de la resistencia necesarios para el modelado del *fan-out* se obtienen mediante el algoritmo de optimización presente en HSPICE.

Nótese que este proceso de precaracterización se realiza sólo una vez para cada tecnología.

3.1.1.4.-Puertas lógicas de múltiples entradas: inversor equivalente.

La metodología expuesta es capaz de evaluar el retardo de propagación para el caso particular de la puerta inversora. Las puertas de múltiples entradas (NOR y OR) requieren un tratamiento especial.

La aproximación clásicamente seguida cuando se emplea un macromodelo temporal (como el que se presenta) para operar con puertas de múltiples entradas consiste en reducir éstas a inversores equivalentes [JuJuP89].

Para realizar tal operación, se definen inversores que emulan el comportamiento temporal de la puerta a modelar, operando siempre bajo condición de peor caso. Consecuencia de este proceder es el error que se consigue, y el no poder determinar con el rigor necesario el camino crítico.

En el modelo que se expone, se han definido lo que se ha dado en llamar como *colisión de señales*. Merced al tratamiento especial al que se somete la estimación del retardo para el caso de puertas de múltiples entradas, se reducen significativamente los errores.

3.1.1.5.- Colisiones.

De forma resumida se exponen los aspectos más relevantes de la metodología tratando con colisiones de señales.

Dada una puerta de múltiples entradas, se dice que existe colisión entre las que señales que excitan la puerta lógica cuando se producen transiciones simultáneas, o muy próximas en el tiempo de dichas señales. Esta proximidad hace variar la evolución de los transistores que conmutan en la puerta lógica. Por supuesto, se exige que el estado lógico a la salida de la puerta excitada cambie.

La aproximación habitual para tratar este problema consiste en considerar que la relación de aspecto del inversor equivalente es la suma de las correspondientes a n inversores (uno por transistor que conmuta). Cuando no existe simultaneidad en las señales excitadoras, se considera que la relación de aspecto equivalente es la correspondiente al inversor formado por el primer transistor en el que tiene lugar la transición. Se detecta un intervalo en el que no queda correctamente definida la relación de aspecto, alcanzándose errores superiores al 200%.

La alternativa desarrollada [GoHeN92] consiste en definir no un inversor equivalente para los dos casos comentados anteriormente, sino una relación de aspecto equivalente función de los atributos de las señales (donde va incluido implícitamente la función lógica). De esta forma, el intervalo de incertidumbre es incluido a través de una beta (β) equivalente que puede tomar infinitos valores en un intervalo definido.

Se han obtenido expresiones (vía simulación) que determinan la relación de aspecto

equivalente. Se observa en la gráfica 3.3 como se consigue el solapamiento (o ajuste) completo de las señales en el intervalo de medida que interesa para el caso de una puerta NOR de dos entradas. La diferencia entre los niveles de tensión (nivel bajo), es consecuencia del hecho de ser ésta una lógica del tipo *ratioed*. Precisamente se definió el retardo de propagación de forma tal que el efecto de variación de los niveles de tensión quedara incluido durante la fase de precaracterización. Se obtienen resultados similares para el caso de otras puertas lógicas.

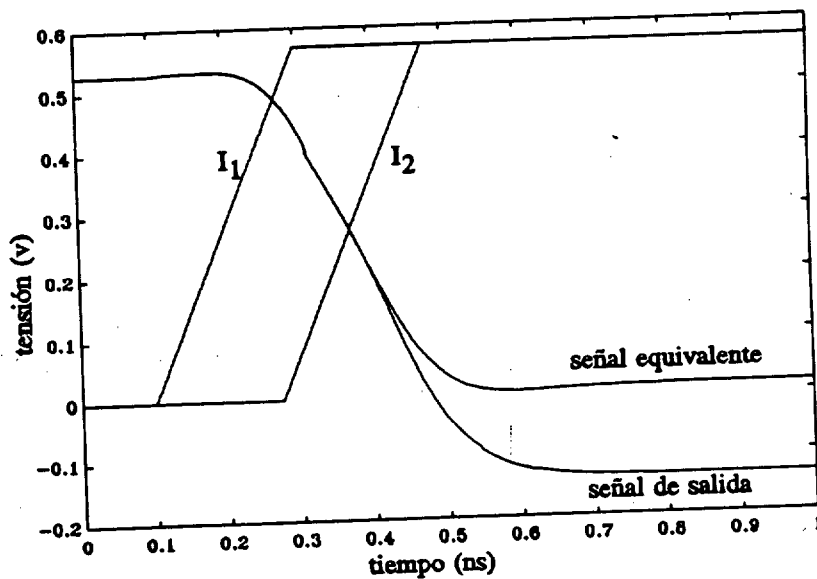


Fig. 3.3. Salida de la puerta NOR-2 y del inversor equivalente.

La forma en que se obtiene la relación de aspecto equivalente es mediante un ajuste por mínimos cuadrados a la nube de puntos calculados vía simulación. En este caso se dispone de dos polinomios de grado tres. Los atributos de la pendiente de salida de la puerta de múltiples entradas resultan del polinomio {3.1}, empleando la nueva relación de aspecto.

Dos son los atributos a identificar por cada señal [HeGoN92]:

- pendiente propiamente dicha y,
- desfase relativo entre señales confluyendo en una misma puerta lógica.

Tal como se definió la pendiente de la señal, el sentido de la transición (nivel lógico bajo a alto y viceversa), se deduce sin más que observar el signo asociado a la misma (positivo para transición de subida y negativo para la de bajada).

Durante la fase de caracterización de la lógica se tienen en cuenta las colisiones

ampliando el rango de variación de los valores de β así como de los posibles valores de la capacidad de carga.

A continuación se presentan algunos resultados así como conclusiones y limitaciones de la metodología descrita.

3.1.2.- Resultados.

La aplicación de la metodología ha dado lugar a la implementación de un primer prototipo de analizador temporal, capaz de operar con colisiones y de suministrar el camino crítico dado un circuito determinado [Herná92]. Este prototipo se ha escrito en el lenguaje de programación C y corre sobre estación de trabajo SUN.

La estructura modular del programa desarrollado permite la incorporación de nuevas subrutinas. Además, al estar contenidos todos los coeficientes en ficheros, se incorporan nuevos coeficientes sin necesidad de compilación. Esto confiere el carácter de no dependencia tecnológica a la metodología expuesta.

Han sido testeados multitud de circuitos (tanto DCFL como SDCFL), desde cadenas de inversores hasta multiplicadores y sumadores. En todos los casos, los tiempos de cómputo empleados por el prototipo han sido al menos tres órdenes de magnitud inferiores a los requeridos por HSPICE.

El error estimado no sobrepasó en ningún caso el diez por ciento (comparado con HSPICE), siendo en muchas de las ocasiones inferior al cinco por ciento. La tabla 3.1 resume los resultados más significativos.

Tabla 3.1. Resultados y error relativo estimado.

Cicuito analizado	HSPICE (ns)	Metodología (ns)	Error (%)
Cadena de 54 inversores	5.078	5.201	+ 2.4
Cadena de 200 inversores	19.216	19.058	- 0.8
Multiplicador de 2 bits	423.59	452.95	+ 6.9

Los errores frente a HSPICE en las distintas configuraciones analizadas son lo suficientemente pequeños como para implicar la formulación en una estrategia de optimización. Se recuerda que los modelos basados en árboles RC para el caso de circuitos

MOS arrojan errores superiores al 25%.

Se destaca la circunstancia de que los errores no se acumulan por etapas, sino que tienden a cancelarse unos con otros al aumentar el tamaño del circuito. Esto es debido en parte a que las pendientes se estabilizan.

3.1.3.- Conclusiones.

Para concluir la presentación en esta sección del modelo temporal desarrollado para circuitos combinatoriales DCFL/SDCFL se resaltan sus características principales.

El modelo evalúa los retardos de propagación incluyendo los efectos de las pendientes y de la capacidad asociada al interconexión. Se destaca el bajo error que se comete en la estimación (sabida la dificultad que el problema encierra).

El prototipo de programa desarrollado computa los tiempos de propagación así como las pendientes, todo ello bajo la consideración de los efectos de las colisiones en puertas de múltiples entradas. Esta operación consume poco tiempo de cómputo y además el programa aporta el camino crítico acorde con el vector de entrada.

La simplicidad que enmarca a la metodología le confiere un potencial apto para su aplicación en circuitos *VLSI*.

Debido a que el modelo emplea ficheros de coeficientes, en la medida en que se mejore la extracción de estos coeficientes, mejora el modelo. Cabe pues plantear la posibilidad de extraer los coeficientes directamente vía medida eléctrica, y no a través de simulaciones.

Como ampliación no recogida en la metodología, se cita el hecho de la inclusión de circuitos secuenciales. La optimización de dichos circuitos no constituye materia de análisis de esta tesis. Su interés en diseño de sistemas digitales síncronos es considerablemente menor que la optimización de las redes lógicas combinatoriales que pueden encontrarse separadas por registros o *latches*.

Dado el carácter analítico y la simplicidad de las expresiones a manejar, la metodología expuesta es apta para la implementación de una estrategia de optimización. Esto viene reforzado por la circunstancia de que se maneja una expresión polinómica (continua y derivable en la variable a optimizar).

3.2.- MODELO PARA EL CÁLCULO DE LA POTENCIA DISIPADA.

Se presenta a continuación el modelo desarrollado para la estimación de la potencia disipada para circuitos combinatoriales DCFL/SDCFL. La formulación es completamente analítica y permite la evaluación del consumo de potencia dado el fichero tecnológico y características topológicas de la familia lógica bajo estudio. La metodología expuesta es lo suficientemente rápida como para poderse utilizar de forma práctica. La comparación de los resultados obtenidos frente a los arrojados por HSPICE es excelente.

3.2.1.- Formulación e hipótesis de partida.

La potencia disipada por los circuitos lógicos en consideración es baja, relativa a otras familias lógicas más difundidas (NMOS, CMOS, ECL), debido fundamentalmente a que las tensiones de alimentación empleadas son de menor valor (1.4 voltios frente a los 5 voltios típicamente utilizados en CMOS). Así, una puerta SDCFL típica tiene un consumo de potencia del orden de 0.25 a 0.5 mW y una puerta DCFL arroja un consumo del orden de 80-100 μ W. Si se desea hacer una comparación entre tecnologías de Si y GaAs es más correcto hacerla entre lógicas correspondientes como ECL-SCFL, CMOS-DPTL, y NMOS-DCFL, pero no es éste nuestro interés aquí.

En general, la potencia disipada en un circuito digital implementado en cualquier tecnología, puede expresarse como la suma de tres términos {3.2}:

$$P_{dis} = P_{est} + P_{din} + P_{fug} \quad \{3.2\}$$

donde:

P_{dis} : potencia disipada.

P_{est} : potencia estática.

P_{din} : potencia dinámica.

P_{fug} : potencia de fugas.

El término estático corresponde a aquel valor de la potencia disipada cuando el circuito se encuentra operando en estado estacionario. Por lo tanto, es la potencia suministrada por las fuentes de alimentación (V_{DD} y V_{SS}) en DC.

El término dinámico da cuenta de la potencia requerida para cargar y descargar todos

los condensadores durante las transiciones de las señales. Como se desprende de la definición, la potencia dinámica es función de la frecuencia de las señales presentes en el circuito.

El último término (fugas), se asocia a la potencia disipada debido a las corrientes de fugas. En general su valor es mucho menor que los otros dos y constante en el tiempo.

Además, dado que los modelos de HSPICE para los MESFETs no incluyen fenómenos de pérdidas, no ha sido posible su evaluación. Su adición al modelo de potencia que se presenta no supone complejidad adicional.

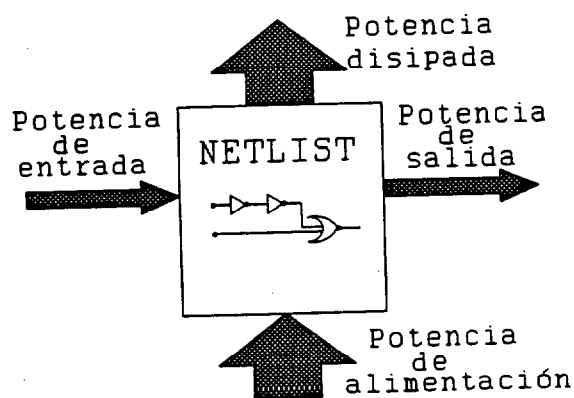


Fig. 3.4. Términos de Potencia.

La fórmula general para el cálculo de la potencia disipada por un circuito lógico viene dada por la siguiente expresión {3.3}:

$$P_{dis} = P_{ali} + P_i - P_o \quad \{3.3\}$$

donde:

P_{ali} : potencia entregada por las fuentes a las puertas lógicas a través de los railes de alimentación.

P_i : potencia de entrada al circuito lógico.

P_o : potencia de salida del circuito lógico.

Los distintos términos se calculan a partir de las expresiones:

$$P_{ali} = \frac{1}{\Delta t} \cdot \sum V_{ali} \cdot \int_0^{\Delta t} i_{ali}(t) dt \quad \{3.4\}$$

el sumatorio se extiende a todos los raíles de alimentación.

$$P_i = \frac{1}{\Delta t} \cdot \sum \int_0^{\Delta t} V_i(t) \cdot i_i(t) dt \quad \{3.5\}$$

el sumatorio se extiende a todos los terminales de entrada.

$$P_o = \frac{1}{\Delta t} \cdot \sum \int_0^{\Delta t} V_o(t) \cdot i_o(t) dt \quad \{3.6\}$$

el sumatorio se extiende a todos los terminales de salida.

En la expresión {3.4} se puede poner fuera de la integral el valor de V_{ali} siempre que se le considere constante. En una situación real no es ni mucho menos constante, dado que su valor se ve alterado por los efectos asociados a los raíles de alimentación, que a altas frecuencias han de ser considerados como auténticas líneas de transmisión [Perdo92]. La inclusión de estos efectos queda fuera del alcance de este modelo de potencia.

Gracias a la regularidad que presentan las puertas lógicas pertenecientes a la familia DCFL/SDCFL respecto a sus dimensiones y a las configuraciones de entrada/salida, los términos de potencia de entrada y salida son similares. Por otro lado, el caso típico es aquel en que mejore el término de potencia entregada por las fuentes frente a la diferencia de las potencias de entrada y salida. Esto es así ya que los niveles de corriente $i_i(t)$ e $i_o(t)$ son corrientes que atraviesan uniones metal-semiconductor y serán mucho menores que $i_{ali}(t)$. Además, los valores de tensión que las multiplican son, en promedio, al menos un orden de magnitud menor en valor que la mayor tensión de alimentación (0.3 voltios frente a 1.4 V).

En la figura 3.5 se representan la potencia de entrada y de salida consumida por el

inversor SDCFL medida con HSPICE. La frecuencia de la señal de excitación de la puerta es superior a 700 MHz, dando lugar a unos picos de potencia (de un valor relativo aproximado al 30%), que como se observa pueden despreciarse comparados con los valores de la potencia en el estado estacionario (la amplitud del pico no es el término a considerar, sino el área encerrada bajo el pico, que sí es mucho menor que la encerrada bajo la parte de la curva correspondiente al estado estacionario).

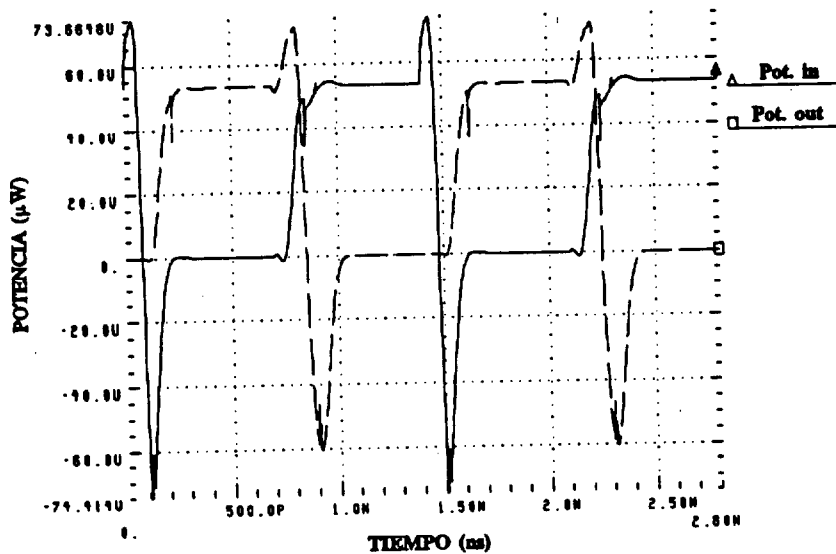


Fig. 3.5. Potencia total de entrada y Potencia total de salida del inversor SDCFL.

Tanto la amplitud del *glitch* como su duración decrecen con la frecuencia de excitación del circuito, siendo el representado lo que puede considerarse un peor caso para los circuitos bajo estudio. La simetría que presentan las curvas se debe a que las geometrías empleadas son iguales, tanto la puerta en cuestión como la que actúa de carga.

El empleo de otras topologías no altera de forma significativa esta simetría. Asimismo se observa como el consumo de potencia por período puede ser despreciado, dado que según reza la fórmula {3.3} se expresa como $P_i - P_o$ (de aquí se deduce la importancia de las simetrías de las curvas). Además, respecto al valor de la potencia entregada por las fuentes de alimentación, tanto P_i como P_o , son aproximadamente inferiores en 1 orden de magnitud, por lo que queda justificada su exclusión {3.7}.

$$P_i - P_o \ll P_{ali} \Rightarrow P_{dis} \approx P_{ali} \quad \{3.7\}$$

El pico de potencia de valor negativo se asocia a la potencia entregada por las capacidades del circuito y no representan consumo de potencia asociado a las fuentes de alimentación a la puerta.

Similar comportamiento muestran las curvas de potencia de entrada y salida para el caso del inversor DFCL. Como diferencia a resaltar es el hecho de que dicha lógica presenta unos consumos de entrada y salida inferiores al de la lógica SDCFL en las mismas condiciones de carga. Ello es consecuencia de que sólo existe una fuente de alimentación (V_{DD}), y de la simplicidad de las puertas (por ejemplo, dos transistores frente a cuatro para el inversor). Los mismos comentarios y conclusiones obtenidos para la lógica SDCFL se aplican directamente al caso de la lógica DCFL.

En la figura 3.6 se representa la potencia total disipada por el inversor SDCFL operando a una frecuencia superior a 700 MHz, medida con HSPICE. Esta frecuencia, superior a la de trabajo de los circuitos bajo estudio, se ha escogido para poner de manifiesto los transitorios de potencia, que se deben a fenómenos capacitivos y que aumentan con la frecuencia. Por consiguiente, en la gráfica se representa lo que se puede considerar como un peor caso.

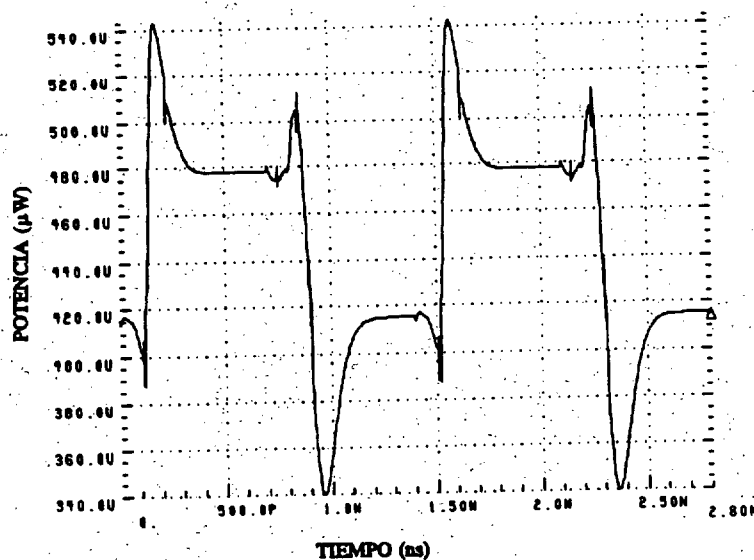


Fig. 3.6. Potencia total consumida por el inversor SDCFL.

Estos picos de potencia (*glitches*) tienen lugar en los intervalos temporales en los que se produce la transición en la señal de excitación de la puerta. En particular, en esta situación, el pico mayor representa aproximadamente tan sólo el 12% del valor de la función

en estado estacionario. El valor relativo del pico de potencia ha sido medido para todas las configuraciones usualmente empleadas por los diseñadores, no excediendo del 15% y, por lo tanto puede ser despreciado.

Las condiciones de medida han sido las mismas empleadas para el caso del retardo de propagación (ver sección 3.1), es decir, se dispone de una puerta llamada de *filtrado*, cuya finalidad es producir una forma de onda similar a las que se propagan por el circuito (y no la rampa proveniente del generador de señal).

Como carga se ha empleado un inversor de misma geometría que el que se testeaba, dando lugar a la simetría que se observa en la gráfica. El empleo de otras cargas no modifica las conclusiones a las que se llega, sólo hace menos simétrica a la curva.

Por tanto, si se desprecian los términos de potencia asociados a las transiciones, el error cometido será del orden del área bajo los picos frente al área bajo la función de potencia en estado estacionario, relativas a la longitud del intervalo en consideración. Esta segunda hipótesis es la básica del modelo de estimación de potencia, que en lo que sigue se propone:

$$P_{dis} \approx P_{ali} \approx P_{est} \quad \{3.8\}$$

Tal como acontece en el caso de los consumos de potencia de entrada y salida, se llega a las mismas conclusiones para el caso de la lógica DCFL. De esta forma, para circuitos DCFL/SDCFL, el modelo de potencia que se presenta estima el consumo de potencia evaluando sólo la potencia entregada por las fuentes de alimentación en el régimen estacionario.

Comentario aparte merece el análisis de los transitorios de las señales (del nivel alto al bajo o viceversa). Pese a que se ha afirmado que se evalúan sólo los valores de potencia en el estado estacionario, la potencia consumida durante las transiciones, que como se observa en las figuras 3.5 y 3.6 puede llegar a ser comparable con la potencia del estacionario, es incluida de la siguiente forma:

- se suponen transiciones ideales (señales tipo escalón) y,
- mediante el parámetro q (ver apartado 3.2.3) se pondera temporalmente la duración del transitorio.

Esta es una aproximación que tenderá en la mayoría de los casos a sobreestimar el

consumo de potencia, que puede verse compensado con el consumo de potencia asociada de los *glitches*, que ha sido despreciada. En los casos en que el diseño haya sido cuidadoso, y el estacionario de las señales no sea comparable con la duración de los transitorios (lo mismo ocurrirá para la potencia consumida) tendrá lugar una sobrestimación del consumo de potencia.

3.2.2.- Definiciones.

En el cálculo de la potencia disipada en un circuito intervienen los conceptos de partición en puertas lógicas y de inversor equivalente.

Dado un circuito lógico arbitrario, se entiende por partición en puertas la partición natural del mismo en conjuntos de transistores constituyendo cada agrupación una puerta lógica. De esta manera, se considera cualquier circuito como una interconexión de puertas lógicas.

Se entiende por inversor equivalente a una puerta lógica (NOR) de las resultantes de la partición en puertas lógicas del circuito, aquel inversor que se obtiene de las puertas NOR, tal que la dimensión del ancho de puerta del transistor que conmuta en el inversor es el resultado de la suma de las anchuras de los canales de aquellos transistores de entrada a la puerta de múltiples entradas que tengan su entrada excitada con una tensión correspondiente al nivel lógico alto (1 , para lógica positiva). Mientras no se afirme lo contrario se considera lógica positiva. La definición de inversor equivalente se aclara en las figuras 3.7.a y 3.7.b. La reducción de otro tipo de puertas lógicas a inversores equivalentes se acomete de forma similar.

En la definición queda implícita la hipótesis de potencia estática ya establecida, pues aquel transistor excitado con un cero lógico (0) se desprecia por estar cortado y no contribuir al consumo de potencia. El inversor equivalente de una puerta NOR de múltiples entradas con todas sus entradas excitadas por un cero (0) lógico tendrá un transistor que conmuta de dimensiones nulas. Es claro de la definición, que el inversor equivalente depende del vector de entrada a la puerta NOR de múltiples entradas.

La reducción de puertas lógicas a inversores equivalentes no añade error alguno a la metodología (salvo el propio introducido durante las parametrizaciones). Asimismo, dado que los valores de carga de los nodos de salida de las puertas lógicas se corresponden con valores puramente topológicos, no se requiere tener en cuenta efectos asociados a colisiones.

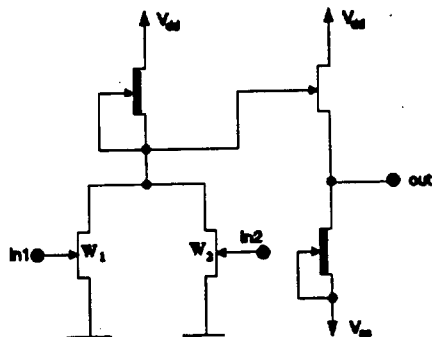


Fig. 3.7.a. Puerta NOR2.

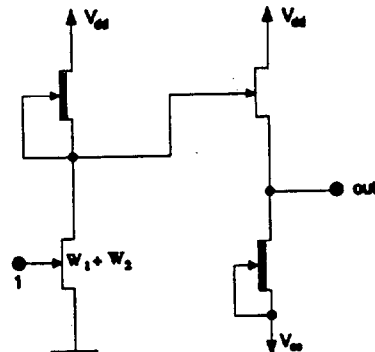


Fig. 3.7.b. Inversor equivalente.

En lo que sigue, P_{OH} denota la potencia disipada por una puerta *inversor equivalente* en una situación estacionaria, cuando la salida en tensión del mismo se encuentra en nivel alto (nivel lógico 1). P_{OH} no dependerá de las dimensiones del transistor que conmuta en el inversor (de la definición, sus dimensiones son nulas), ya que en esa situación estará cortado.

Análogamente P_{OL} denota la potencia disipada cuando la salida está en nivel bajo. P_{OL} no será función de la carga de la puerta si se considera que está cargada por más inversores, (o en general puertas) lo suficientemente próximas como para despreciar los efectos de las pérdidas en las interconexiones, ya que todas tendrán su correspondiente transistor que conmuta en estado de corte.

Finalmente se define el parámetro q , como la frecuencia relativa de que un nodo de salida de una puerta lógica se encuentre en el estado lógico alto. Considerando todas las situaciones posibles de los vectores de entrada (que aumenta en número de forma exponencial con el número de entradas al circuito), el parámetro q está próximo al concepto de probabilidad de que el nudo en cuestión esté en estado lógico alto. En lo que sigue se referenciará a q con ese nombre.

3.2.3.- Metodología de cálculo de las funciones de estimación.

De las hipótesis anteriormente formuladas se tiene, para inversores DCFL/SDCFL:

$$\begin{aligned}
P_{dis} = P_{ali} &= \frac{1}{\Delta t} \cdot [V_{DD} \cdot \int_0^{\Delta t} I_{DD}(t) dt + V_{SS} \cdot \int_0^{\Delta t} I_{SS}(t) dt] = \\
&= \frac{1}{\Delta t} \cdot [V_{DD} \cdot (q \cdot I_{DD}^H + (1-q) \cdot I_{DD}^L) \cdot \Delta t + V_{SS} \cdot (q \cdot I_{SS}^H + (1-q) \cdot I_{SS}^L) \cdot \Delta t] = \\
&= q \cdot (V_{DD} \cdot I_{DD}^H + V_{SS} \cdot I_{SS}^H) + (1-q) \cdot (V_{DD} \cdot I_{DD}^L + V_{SS} \cdot I_{SS}^L) = \\
&= q \cdot P_{OH} + (1-q) \cdot P_{OL} \tag{3.9}
\end{aligned}$$

donde, $I_{DD}(t)$ es la corriente entregada al inversor por la fuente V_{DD} , I_{DD}^H es el valor de esa corriente cuando la salida del inversor se encuentra en estado alto, I_{DD}^L es la corriente cuando la salida está a nivel bajo (ver figuras 3.8.a y 3.8.b). Los subíndices SS se refieren a la fuente V_{SS} con el mismo significado. Los términos:

$$P_{OH} = V_{DD} \cdot I_{DD}^H + V_{SS} \cdot I_{SS}^H, \quad P_{OL} = V_{DD} \cdot I_{DD}^L + V_{SS} \cdot I_{SS}^L,$$

se corresponden con las potencias anteriormente definidas. Obviamente el término $(1-q)$ representa la probabilidad de que la salida se encuentre en el nivel bajo de tensión.

De la expresión {3.9} se sigue que para el cómputo de la potencia disipada por un inversor es preciso el previo conocimiento de los valores de q , P_{OH} y P_{OL} . Se necesita por tanto encontrar un método capaz de suministrar dichos datos y a ser posible de forma analítica y automática.

La probabilidad de que la salida del inversor se encuentre en nivel alto puede calcularse con precisión utilizando algún simulador lógico [HILO88], sin más que evaluar el cociente entre el número de veces que el nodo está en el nivel alto y el número de intervalos totales de simulación para todos los vectores de entrada posible. Sin embargo, una aproximación habitualmente usada, consiste en tomar como valor $q = 1/2$.

Esta aproximación no necesita justificación para el caso del inversor. La situación es distinta para el caso de puertas NOR de múltiples entradas. La justificación en este caso viene dada por el hecho de que frecuentemente los diseñadores evitan que a las entradas de la puerta NOR se active simultáneamente más de un transistor de *pull-down*. Bajo esta restricción, la puerta NOR se comporta dinámicamente como un simple inversor para el que no resulta sólo aproximado el valor $q = 1/2$, sino que es exactamente el valor

correspondiente. Los excelentes resultados obtenidos en los múltiples casos estudiados, confirman el hecho de poder ignorar otros casos.

$$I_{DD}(t) = \begin{cases} I_{DD}^H & 0 \leq t < t_c \\ I_{DD}^L & t_c \leq t \leq \Delta t \end{cases} \quad q = \frac{t}{\Delta t}$$

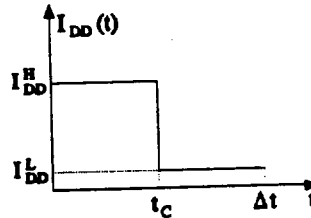


Fig. 3.8.a. Definición de \$I_{DD}\$

Fig. 3.8.b. Dependencia de \$I_{DD}\$ con el tiempo.

Para el cálculo de \$P_{OH}\$ y \$P_{OL}\$ existen al menos dos alternativas, deducir expresiones analíticas que den los valores de las corrientes de salida de las fuentes en función de la carga del inversor y sus dimensiones o bien, obtener sus valores a través de simulación eléctrica y almacenar dichos valores en tablas. Es claro que una formulación analítica es en general más deseable por múltiples razones. De entre ellas, por ejemplo, porque facilita un conocimiento más profundo (físico) del funcionamiento de la puerta, así como que hace a la metodología independiente tecnológicamente.

Sin embargo, al plantear las ecuaciones de circuito de forma general, se obtienen sistemas de ecuaciones trascendentes que sólo ofrecen solución numérica (típicamente mediante algoritmos iterativos costosos en tiempo de cómputo). Para su resolución mediante técnicas analíticas se requieren aproximaciones significativas tan restrictivas que las soluciones obtenidas se muestran poco precisas (fuera de los límites necesarios). Por lo tanto, y en aras de conseguir una mayor precisión así como un menor tiempo de cómputo, se ha optado por la alternativa consistente en utilizar simulación eléctrica para el cálculo de \$P_{OH}\$ y \$P_{OL}\$ y su almacenamiento en tablas de datos.

Se han realizado múltiples simulaciones mediante el programa HSPICE de inversores de dimensión variable del transistor que conmuta para distintas configuraciones de carga y para los dos valores del vector de entrada (alto y bajo). Como variable de simulación se toma la dimensión del transistor de *pull-down*, para efectuar el barrido de todas las configuraciones posibles, para de esta forma poder dar cuenta de las puertas NOR de múltiples entradas presentes en los circuitos reales. Este proceso se encuentra totalmente automatizado a través de diferentes subrutinas escritas en el lenguaje de programación C.

La alternativa de caracterizar mediante simulación los valores de P_{OH} y de P_{OL} es viable merced a que el número de situaciones distintas usualmente empleadas por los diseñadores es relativamente pequeño. En efecto, el estilo de diseño se muestra cada vez más y más regular en los actuales circuitos lógicos. Esta tendencia, que ya es norma en los diseños *VLSI* implementados en tecnología CMOS, se ha adoptado también en los diseños GaAs.

En la familia lógica considerada (DCFL/SDCFL) este hecho se ve reforzado en el sentido de que sólo se emplean unos pocos tipos de puertas lógicas (generalmente inversores y puertas NOR/OR de hasta tres entradas). Así, en las simulaciones se permitió la variación del factor de área del inversor en el intervalo [8,60], y se tomaron nueve configuraciones de cargas posibles (hasta *fan-out* tres).

Dada la dificultad de expresar el *fan-out* mediante capacidades lineales (tal como se hace, por ejemplo en CMOS), éste se referencia según la geometría de la puerta *colgada* del nudo de salida de la puerta en cuestión. Así, si los valores del ancho de puerta del transistor de *pull-down* de la etapa lógica son $A = 12$ y $B = 16$ micras respectivamente, se pueden representar todas las combinaciones posibles de forma sencilla: ($AA \Rightarrow$ carga 12 ... $BBB \Rightarrow$ carga 16...etc).

A su vez cada uno de los valores se encuentra asociado al *buffer* existente. Otras configuraciones futuras se incorporan de igual forma. Como se comenta en el apartado 3.2.5, se empleará un procedimiento de ajuste para tratar de forma más cómoda la carga asociada al *fan-out*.

Los resultados obtenidos para P_{OH} y P_{OL} utilizando uno de los *buffers* típicamente empleados por los diseñadores se presentan en las figuras 3.9.a y 3.9.b. La inclusión de otros tipos de *buffers* no presenta dificultad adicional.

En la figura 3.9.a se ha representado el consumo de potencia P_{OH} del inversor SDCFL en función del ancho de puerta del transistor de *pull-down* de la etapa lógica. Como parámetro se ha dispuesto la carga (expresada según se ha comentado anteriormente).

Queda patente en esta gráfica el hecho anteriormente comentado de la independencia de P_{OH} con la dimensión del ancho de puerta del transistor de *pull-down*. El hecho de que P_{OH} aumente con la carga es debido a que la lógica presenta variaciones de los niveles de tensión con el ancho de los transistores, es decir lo que se conoce como *ratioed logic* [LonBu90]. Esta circunstancia no tiene lugar en los circuitos CMOS. Obsérvese cómo al aumentar la

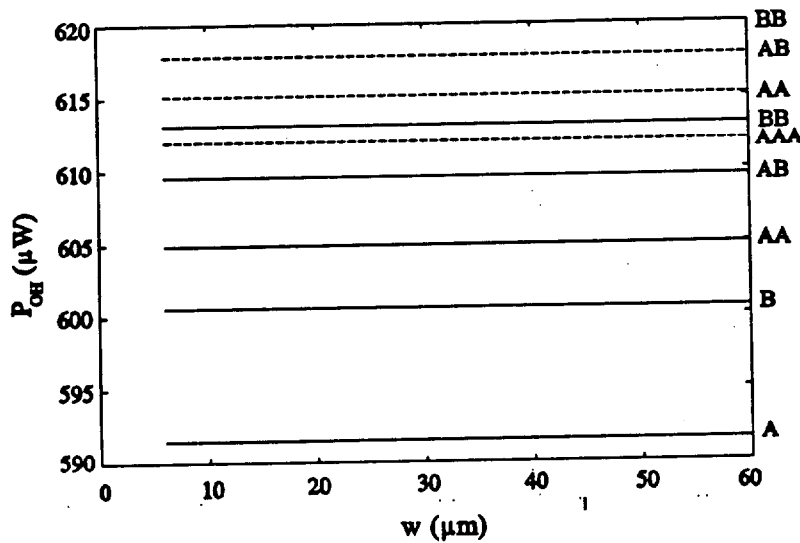


Fig. 3.9.a. Dependencia de P_{OH} con w (dimensión del transistor del "driver").

dimensión (ancho de la puerta) del transistor de *pull-down* del inversor, la potencia disipada disminuye. La situación es bien distinta al caso CMOS, pues aquí sólo se permite la variación de la dimensión de un transistor, mientras que en los inversores CMOS varían las dimensiones de ambos.

El hecho de que el consumo de potencia disminuya con el ancho del transistor *pull-down* de la etapa inversora se debe a que al aumentar el ancho de dicho transistor, éste demanda un flujo mayor de corriente, haciendo que la tensión en el nudo de drenador decrezca; como consecuencia de esto, disminuye el flujo de corriente a través de los transistores de *pull-up* y de *pull-down* de la etapa de *buffer*.

Un estudio comparativo de los valores P_{OH} y P_{OL} , tal como se observa en las figuras 3.9.a y 3.9.b, muestra que el valor de P_{OH} es siempre superior al de P_{OL} . Así, el valor mínimo para P_{OH} es aproximadamente de $590 \mu W$, mientras que el valor mínimo de P_{OL} (para la configuración $w = 12 \mu m$), es inferior a $500 \mu W$.

Esto se debe a que cuando la salida del inversor se encuentra en el estado alto, fluye más corriente por el transistor de *pull-up* de *buffer* (típicamente $300 \mu A$). Esta corriente es demandada por los transistores de carga.

Para la situación opuesta, el flujo de corriente se reduce a valores típicos del orden de $100 \mu A$; valor que disminuye dramáticamente con la dimensión del transistor de *pull-down*

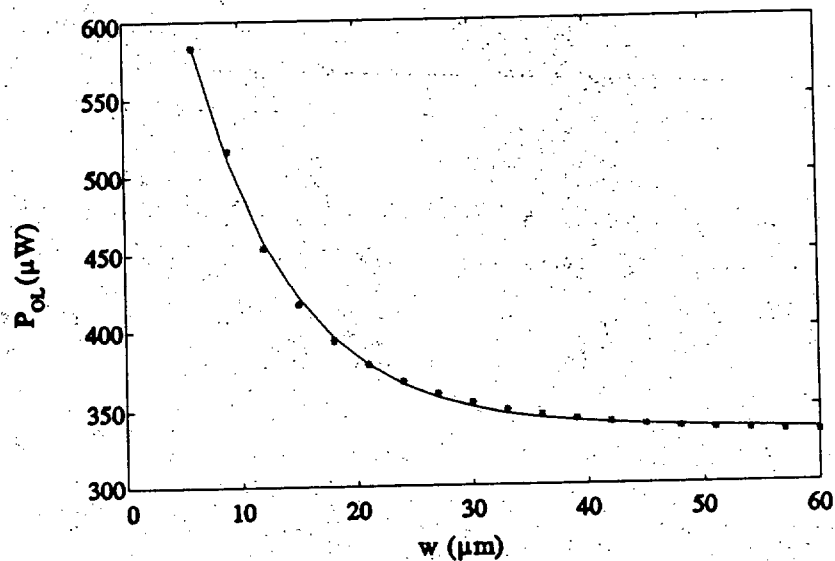


Fig. 3.9.b. Dependencia de P_{OL} con w (dimensión del transistor de carga).

de la etapa inversora. Precisamente dada la disminución de la corriente a través de los transistores del *buffer*, la mayor contribución al consumo de potencia viene dada por el transistor de *pull-up* de la etapa inversora cuando la salida se encuentra en estado bajo. Asimismo es la etapa del seguidor de fuente la que contribuye con un mayor consumo de potencia cuando la salida de la puerta está a nivel alto.

3.2.4.- Capacidades asociadas al cableado.

Los efectos asociados a las capacidades debidas al cableado (capacidades de interconexión) de corta longitud que pueden ser modeladas a través de simples condensadores, no han sido incluidos.

La definición de capacidad equivalente de entrada en gran señal es posible, pero el hecho de ser un valor promediado introduce errores que no se cometen con la formulación presentada.

En cualquier caso, la no inclusión de las capacidades parásitas asociadas a las interconexiones, puede ser completamente despreciada siempre que los valores de los condensadores implicados no sean excesivos. Esta situación resulta ser cierta dentro de los límites que delimitan el problema bajo estudio.

Así, una simple estimación asocia un valor del orden de 10^{-16} vatios de potencia consumida por condensador. Este valor se debe a que la potencia es proporcional a $C \cdot V^2$, y los valores de capacidad parásita son del orden de decenas de femptofaradios (10^{-14} faradios), mientras que la tensión se expresa en centenas de milivoltios (10^{-1} voltios).

Por consiguiente, en primera aproximación estas capacidades pueden ser despreciadas en lo que afecta al consumo de potencia, y las estimaciones a nivel *post-layout* se realizarán en el marco de esta simplificación.

3.2.5.- Ajuste de P_{OL} y de P_{OH} .

Atendiendo a la dependencia de P_{OL} con el ancho de puerta (w), la dependencia funcional representada en la figura 3.9.b, ha sido ajustada mediante una expresión polinómica. Se observa en la gráfica el excelente acuerdo entre los datos experimentales y los obtenidos evaluando el polinomio.

Según este proceder, se expresa P_{OL} mediante un polinomio de grado cuatro en la variable considerada. El ajuste polinómico se ha efectuado mediante mínimos cuadrados.

En la tabla 3.2 se pueden ver los coeficientes del polinomio para un caso particular empleando las dimensiones del *buffer* que en la tabla se indican. Éstos se encuentran ordenados según el criterio de potencias decrecientes. Se justificará en el siguiente capítulo de esta tesis (capítulo 4) el empleo de la geometría 9.2/6 para el *buffer* (razón de su presencia en las tablas de datos que aparecen en este capítulo).

Tabla 3.2. Valores del polinomio de ajuste para la función P_{OL} .

Buffer	9.2/6				
Coef. (μW)	c_1	c_2	c_3	c_4	c_5
	170.71E-6	-27.883E-3	1.6794	-44.974	802.99

Inicialmente se intentó el ajuste de la función empleando una expresión exponencial {3.10},

$$P_{OL}(w) = a_1 \cdot \exp(-\lambda_1 \cdot w) + a_2 \cdot \exp(-\lambda_2 \cdot w) \quad \{3.10\}$$

que contiene dos parámetros lineales (a_1 y a_2), y dos no lineales (λ_1 y λ_2). Si bien el ajuste logrado empleando la función exponencial era ligeramente mejor (en algunos casos) que el que se alcanza mediante el polinomio, se hubo que cambiar de dependencia funcional debido a problemas de convergencia en el algoritmo de optimización. Esto es consecuencia de que las funciones exponenciales muestran enormes problemas cuando de convergencia numérica se trata. Dado que el algoritmo de optimización (ver capítulo 5) emplea dos métodos numéricos para resolver el problema planteado (gradiente de descenso y Newton-Raphson), ha sido imprescindible la sustitución de dicha expresión exponencial por el polinomio anteriormente expuesto.

Se indican en la tabla 3.3 los valores de los coeficientes requeridos por la función exponencial (a utilizar cuando se precise un mejor acercamiento a los datos experimentales). Para el propósito perseguido, basta la precisión alcanzada por la expresión polinómica.

Tabla 3.3. Valores del parámetro P_{OL} .

Buffer	a_1 (mW)	a_2 (mW)	λ_1	λ_2
6/4	0.279	0.33	0.1135	0.213 E-03
9.2/6	0.506	0.34	0.1212	0.246 E-03

El hecho de reemplazar los valores de P_{OL} mediante una expresión matemática permite la rápida evaluación de un número mayor de posibilidades para las dimensiones de los transistores, dado que se barre un continuo de valores. De esta forma se prescinde de las tablas de datos.

La tabla 3.4 contiene los parámetros necesarios para el cómputo de P_{OH} . Los datos contenidos en dicha tabla se han ordenado según el criterio de carga efectiva creciente (se entiende por carga efectiva creciente a aquella cantidad obtenida sumando los valores de los anchos de los transistores de carga). Se observa cómo la potencia disipada P_{OH} crece con la carga efectiva en todos los casos excepto para la pareja *BB*, *AAA*. Este par se diferencia de los demás en que:

- se pasa de *fan-out* igual a 2 a tener un *fan-out* igual a 3 y,
- el valor de la carga efectiva se incrementa sólo un valor de cuatro unidades.

Estas circunstancias, que no se dan en el resto de los casos, evidencian la no linealidad

del problema bajo estudio. Esta particular configuración ha sido analizada en detalle. Se han considerado todas las corrientes y tensiones involucradas. Así se obtiene que el par *BB*, pese a tener un carga efectiva menor, requiere unos flujos de corriente en las puertas de los transistores de carga, ligeramente superior (tan sólo de $0.4 \mu\text{A}$) para el ajuste de corrientes.

Esto se debe al estado de polarización del diodo Schottky presente en la puerta de los MESFETs (afectado directamente por la resistencia parásita de fuente, R_s). Esta ínfima diferencia basta para que el consumo de potencia sea, asimismo, ligeramente superior.

Tabla 3.4. Valores del parámetro P_{OH}

Buffer	9.2/6								
Carga	A	B	AA	AB	BB	AAA	AAB	ABB	BBB
$P_{OH}(\mu\text{W})$	591	601	605	610	613	612	615	618	620
Buffer	6/4								
Carga	A	B	AA	AB	BB	AAA	AAB	ABB	BBB
$P_{OH}(\mu\text{W})$	440	448	452	456	460	459	461	464	466

Consecuencia directa de la presencia de no linealidad en los valores de P_{OH} , es el hecho de que no pueden ser ajustados a través de una sencilla expresión. Puede observarse en la figura 3.10 la bondad del ajuste alcanzado con un polinomio de grado quinto (5°) para el inversor SDCFL cuyo *buffer* tiene el dimensionado típico 9.2/6.

En dicha gráfica, P_{OH} se expresa en μW y w en micras (aquí w se refiere a la anchura del transistor de carga). El ajuste para otras configuraciones conduce a similares resultados.

Dado que la potencia P_{OH} es independiente del transistor *pull-down*, puede obtenerse el consumo de potencia P_{OH} para cualquier configuración de carga contenida entre los límites representados.

Nótese que, al igual que sucede para el caso de la estimación de P_{OL} , la función de ajuste barre un continuo de valores de w . De esta forma, se incluyen nuevos dimensionados

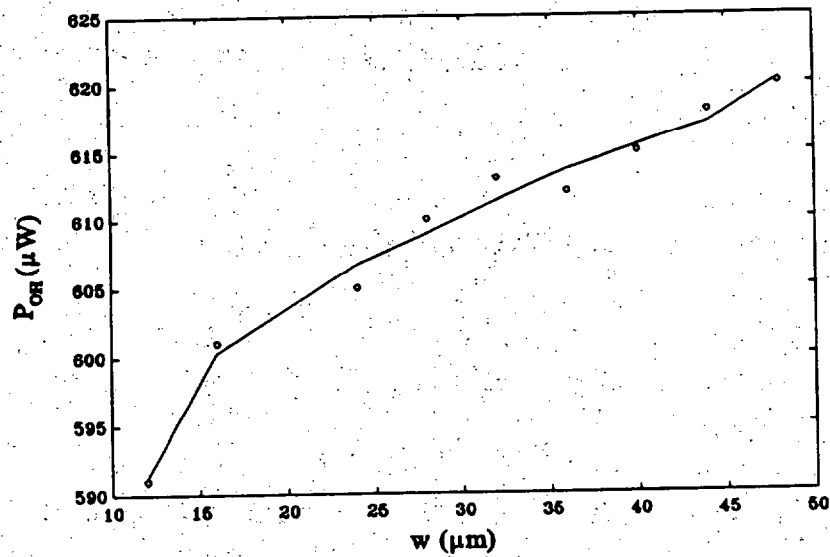


Fig. 3.10. Ajuste de P_{OH} mediante polinomio de 5º grado (lógica SDCFL).

de transistores sin necesidad de realizar múltiples medidas de P_{OH} y P_{OL} .

La tabla 3.5 contiene los coeficientes del polinomio de ajuste para ambos casos (9.2/6 y 6/4). Estos coeficientes se encuentran ordenados según potencias decrecientes. Dicho ajuste se ha llevado a cabo mediante el programa MATLAB [MATLA85] y se encuentra completamente automatizado.

Tabla 3.5. Valores de los coeficientes del polinomio de ajuste para la función P_{OH} .

Buffer	9.2/6					
Coef.(μW)	c_1	c_2	c_3	c_4	c_5	c_6
	5.216E-6	-8.301E-4	5.158E-2	-1.566	23.841	457.412
Buffer	6/4					
Coef.(μW)	c_1	c_2	c_3	c_4	c_5	c_6
	5.405E-6	-8.313E-4	4.962E-2	-1.441	21.027	325.572

El valor del parámetro P_{OH} concerniente a la lógica DCFL ha sido estimado para una de las geometrías más empleadas por los diseñadores, es decir, el transistor de *pull-up* con

una anchura de 4 micras y el de *pull-down* con un ancho de puerta de 30 micras. La incorporación de nuevas geometrías vuelve a ser, como ocurre con la lógica SDCFL, sencilla. Los valores P_{OH} medidos se extienden a todo el conjunto de combinaciones admitidas. Así, se abarca desde la configuración:

- inversor DCFL-inversor DCFL (*fan-out* = 1, carga $w_g = 30$ micras),
- inversor DCFL-inversor DCFL (*fan-out* = 3, carga $w_g = 90$ micras),

donde w_g representa la suma de los anchos de puerta de los transistores de carga. Dado que w_g se extiende hasta el valor de 90 micras, quedan incluidas todas las posibles combinaciones DCFL - SDCFL admitidas.

Los valores P_{OH} para el caso de la lógica DCFL muestran una distribución (ver figura 3.11) que permiten un ajuste numérico mediante un polinomio de tercer grado, lo cual redonda en simplicidad y menor tiempo de cómputo. Los valores de este polinomio se presentan en la tabla 3.6.

Tabla 3.6. Valores de los coeficientes del polinomio de ajuste (DCFL).

Dim.	4/30			
Coef. (μW)	c_1	c_2	c_3	c_4
	3.58E-5	-7.00E-3	5.06E-1	271.0

En lo que al valor del parámetro P_{ol} se refiere para la lógica DCFL, dado que el conjunto de geometrías posibles se reduce al de 4/30, 4/60 y 4/90 (este último raramente empleado), se precisa un sólo valor para cada geometría posible de *pull-down*:

$$P_{ol}(4/30) = 308.84 \mu W,$$

$$P_{ol}(4/60) = 310.66 \mu W,$$

$$P_{ol}(4/90) = 311.08 \mu W,$$

Tal como se presentará posteriormente, los resultados obtenidos empleando las expresiones de ajuste (polinomios) justifican la no necesidad del manejo de tablas de datos.

El empleo de las expresiones de ajuste hace asimismo más fácil la programación de los algoritmos pertinentes.

Siempre que se requiera mayor precisión pueden utilizarse los datos contenidos en las tablas (sólo para el cálculo de P_{OH} , donde el ajuste no es tan bueno como para el caso de P_{OL}).

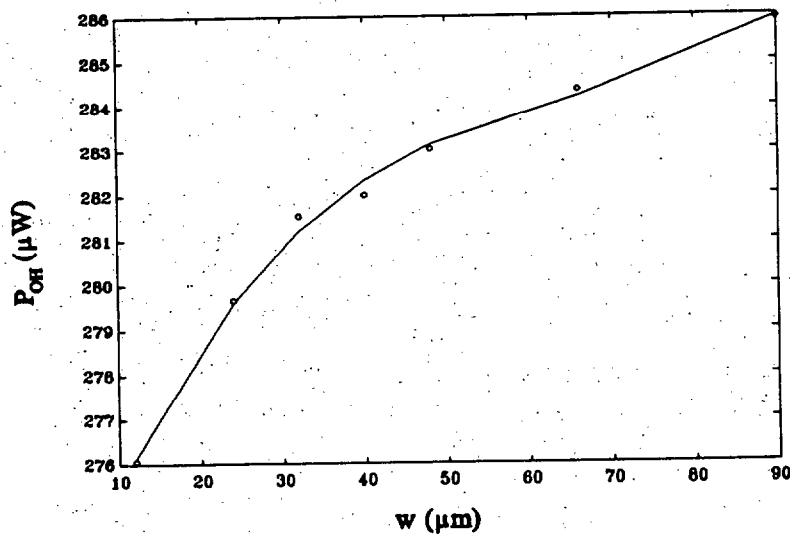


Fig. 3.11. Ajuste de P_{OH} mediante polinomio de 3º grado (lógica DCFL).

Además de la ventaja que supone el manejo de un menor número de coeficientes (con la consiguiente reducción del tiempo de cómputo), se dispone de esta forma de una expresión analítica del consumo de potencia, útil para optimización.

3.2.6.- Resultados.

Ante un circuito lógico descrito por partición en puertas lógicas, la metodología de cálculo de la potencia disipada consiste en reducirlo a inversores equivalentes a partir del conocimiento del vector de entrada o de un conjunto representativo de sus vectores de entrada.

Si el dato que se maneja es q , en vez de un vector de entrada particular, la reducción se hace suponiendo igual probabilidad para cada combinación presente a la entrada de las diferentes puertas lógicas. Seguidamente se define el inversor equivalente para el cálculo de la P_{OL} correspondiente; dicho inversor posee una dimensión ponderada por el número de

combinaciones posibles.

Así, si a la salida de una puerta NOR de dos entradas (cuyos transistores de *pull-down* son de dimensión w_1 y w_2) se tiene una frecuencia de estado alto $q = q_0$, para el cálculo de P_{ol} del inversor equivalente se calcula el valor de w equivalente ponderando las tres posibilidades (01, 11, 10) de obtener un cero lógico a la salida.

Con ésto la potencia disipada por la puerta se computa por:

$$P_{dis} = q_0 \cdot P_{OH} + (1 - q_0) \cdot \left[\frac{1}{3} \cdot P_{ol}(w_1) + \frac{1}{3} \cdot P_{ol}(w_2) + \frac{1}{3} \cdot P_{ol}(w_1 + w_2) \right] \quad \{3.11\}$$

seleccionando los parámetros de las tablas a partir de la correspondiente topología.

Con objeto de validar la metodología, han sido testeadas múltiples estructuras, presentándose alguno de los resultados más significativos en la tabla 3.7. En dicha tabla puede verse el consumo de potencia por ciclo de reloj así como el error relativo frente a HSPICE.

Se observa el buen acuerdo entre los datos estimados según la formulación presentada y los obtenidos vía HSPICE. Las simulaciones se efectuaron para vectores de entrada tales que $q = 1/2$. Las condiciones de medida fueron las mismas para todos los circuitos, excepto para los osciladores en anillo. Así, se dispuso siempre de puertas lógicas de *filtrado* como de puertas lógicas actuando de carga. La frecuencia de trabajo era de 500 MHz, con señales de 200 picosegundos de rampa.

Los osciladores en anillo consistían en 13 puertas NOR de dos entradas, las cuales estaban cortocircuitadas, de tal forma que cada puerta NOR actuaba como un simple inversor. La frecuencia de oscilación resultó ser de aproximadamente 300 MHz para la geometría $w = 12$ micras, y de 220 MHz para la geometría $w = 16$ micras.

El aumento de error que se obtiene en la estimación de potencia para el caso de los sumadores está asociado al valor del parámetro q . Pese a que el vector de entrada se seleccionó de tal forma que $q = 1/2$, sólo la detallada simulación lógica puede garantizar dicho valor.

Destacan estos ejemplos por la presencia de puertas OR, las cuales también se reemplazaron por los respectivos inversores equivalentes (ver esquemático en el capítulo 6).

Tabla 3.7. Resultados y error estimado.

Circuitos testeados	Formulación (mW)	HSPICE (mW)	Error (%)	n° trans.
Cadena de 10 inversores DCFL (w=30)	2.951	2.954	+ 0.1	20
Cadena de 10 inversores SDCFL (w=12)	5.240	5.276	+ 0.6	40
Cadena de 10 inversores SDCFL (w=16)	5.062	5.130	+ 1.3	40
Oscilador en anillo SDCFL (w=12, 13 etapas)	5.191	5.317	+ 2.3	65
Oscilador en anillo SDCFL (w=16, 13 etapas)	5.168	5.301	+ 2.5	65
Sumador CLA 4 bits	21.05	20.59	+ 2.2	217
Sumador CLA 5 bits	24.31	25.85	-6	272

Como comentario adicional se añade el hecho de que para estructuras regulares (repetitivas) es útil estimar el consumo de potencia por bit, cantidad que sirve para estimar el consumo de potencia de estructuras más complejas.

Para el caso en cuestión se obtiene un consumo por bit, para el sumador de acarreo anticipado (*Carry Look-Ahead*) del orden de 5.15 mW. Una estimación arrojaría una potencia consumida para el sumador de 10 bits del orden de 52 mW.

3.2.7.- Conclusiones.

Se concluye la presentación del modelo para la estimación del consumo de potencia para los circuitos implementados en las familias DCFL y SDCFL. Constituye ésta una metodología robusta y de uso sencillo. Es tecnológicamente independiente, y requiere tan sólo la precaracterización del conjunto de geometrías empleadas. La comparación con las estimaciones

HSPICE se desvían menos del 10% en los casos evaluados.

El modelo presentado se define como *estático*. Para incluir efectos asociados a las capacidades sería preciso disponer además de los niveles de tensión V_{OL} y V_{OH} en los nudos capacitivos.

Al evaluarse los términos de potencia mediante expresiones polinómicas (contínuas y derivables en la variable w), se espera buen comportamiento numérico de las funciones, lo cual redundará en bajos tiempos computacionales para el algoritmo de optimización.

La formulación permite la rápida evaluación a nivel de *pre-layout* de circuitos lógicos combinacionales DCFL/SDCFL. Además, la formulación es adecuada para incorporarla en estrategias de optimización, bien sea acometiendo el problema potencia-retardo de propagación o el interesante problema potencia-área-retardo.

Como ampliaciones no recogidas en la metodología cabe citar tres:

- inclusión de corrientes parásitas y efectos de capacidades parásitas,
- dependencias del consumo de potencia con el estilo de diseño y,
- aplicación a circuitos secuenciales.

Cualquiera de los dos primeros aspectos se incorporan a la metodología presentada sin añadir excesiva complejidad. La inclusión de los efectos asociados a las capacidades parásitas serviría para realizar evaluación a nivel de *post-layout*.

Mención especial requiere la adaptación de la metodología al caso de circuitos secuenciales. En este caso, la asignación del valor del parámetro q no es sencilla, dado que las tensiones en los nodos del circuito (entiéndase niveles alto y bajo), son no sólo función del vector de entrada sino que, también lo son del estado anterior del circuito lógico.

La aplicación directa de la metodología aquí presentada pasa por una minuciosa simulación lógica con el fin de asignar el correcto valor al parámetro q .

Esta simulación queda notablemente simplificada en el caso de diseños con técnicas síncronas en las que las distintas redes combinacionales se hayan separadas por registros (*flip-flops*) ó básculas (*latches*) activadas por la señal de reloj. Afortunadamente esta técnica es dominante en los diseños *VLSI*, que son precisamente los que requieren estimaciones y evaluación de la potencia consumida.

3.3.- MODELO PARA LA ESTIMACIÓN DEL ÁREA OCUPADA.

Se presenta a continuación el modelo desarrollado para la estimación del área ocupada. A diferencia de los modelos ya expuestos (potencia y retardos de propagación), veremos como unas sencillas expresiones matemáticas son suficientes para modelar la superficie ocupada en el *chip*. La metodología expuesta es analítica y de bajo coste computacional, asegurando su empleo de forma práctica como herramienta de estimación a nivel de *pre-layout*. Los resultados obtenidos se comparan con los medidos directamente de los *layouts*, suministrando errores inferiores al 5%.

3.3.1.- Hipótesis de partida.

Dentro del marco de desarrollo de esta tesis, se pretende la formulación de un modelo capaz de estimar, con error controlado, el área ocupada por los circuitos bajo estudio, es decir los edificados sobre lógicas DCFL o/y SDCFL (siempre circuitos diseñados *a medida*, es decir *full-custom*). Constituye ésta la primera restricción.

A su vez, se desea que la formulación sea tecnológicamente independiente. La naturaleza del problema no plantea ninguna dificultad especial de cara a conseguir dicha formulación. Con el afán de implementar una estrategia de optimización, el hecho de disponer de expresiones analíticas dota al problema de mayores facilidades de resolución. Una vez más, la simplicidad del tema que se aborda lleva implícita esta condición.

Otro importante pilar sobre el que se sustenta el modelo de área se debe a la exigencia de regularidad geométrica en los diseños. Tal como se ha comentado en anteriores secciones de este capítulo (3.1 y 3.2), los diseñadores, cada vez más, emplean estrategias de replicación de estructuras de cara a acelerar el tedioso proceso de diseño. Estos hábitos de diseño permiten asimismo cometer menos errores.

Como última restricción se muestra el hecho de que el modelo a exponer va enfocado a la estimación del área de celdas y estructuras (ver definición en apartado 3.3.2) todas ellas implementadas según técnicas a medida (*full-custom*), en especial técnicas de diseño con orientación matricial o en bandas. A partir de estas estimaciones es fácil obtener estimaciones de bloques básicos de mayor complejidad.

Sin embargo no se pretende realizar estimaciones a nivel del *chip* completo. Así tampoco entra en el seno de este modelo el complejo tema del ruteado de pistas de

interconexión (*routing*) ni los aspectos referentes al problema del posicionamiento relativo de las diferentes estructuras y bloques básicos (*placement*).

3.3.2.- Consideraciones generales.

Se presentan a continuación algunas consideraciones útiles para el desarrollo del modelo que resultan de observar las líneas dominantes en diseños implementados en forma *full-custom* en GaAs.

3.3.2.1.- Celda y Estructura.

Se define como celda a la unidad funcional básica que se compone de un número no excesivo de puertas lógicas mediante la cual se construyen estructuras. Dado que las dimensiones de las interconexiones implicadas en el conexionado interno de la celda (transistor a transistor) son pequeñas (típicamente 100 μm), la estimación del área que ocupan no requiere de tratamiento especial.

Se define como estructura al bloque funcional compuesto de celdas agrupadas de forma regular. Por ejemplo la celda básica del multiplicador de *n-bits* la constituye el sumador de 1 *bit*. Aquí, el papel de las interconexiones ya requiere tratamiento especial, precisando de parámetros de ajuste.

3.3.2.2.- Reglas de Diseño.

Las reglas de diseño establecen el conjunto de distancias y geometrías (anchos, largos, separaciones...) que deben verificar los elementos que se definen en cada capa del proceso, entre sí mismos y con elementos de otras capas, para garantizar una correcta fabricación de las estructuras definidas en el trazado (*layout*). En los diseños incluidos en esta tesis se han utilizado las reglas definidas en el proceso EGAS de TriQuint (GigaBit logic) a las que se ha tenido acceso a través de la Universidad de Adelaida.

3.3.2.3.- Metal 1 y Metal 2.

El conjunto de reglas de diseño disponibles para las lógicas DCFL/SDCFL permite la

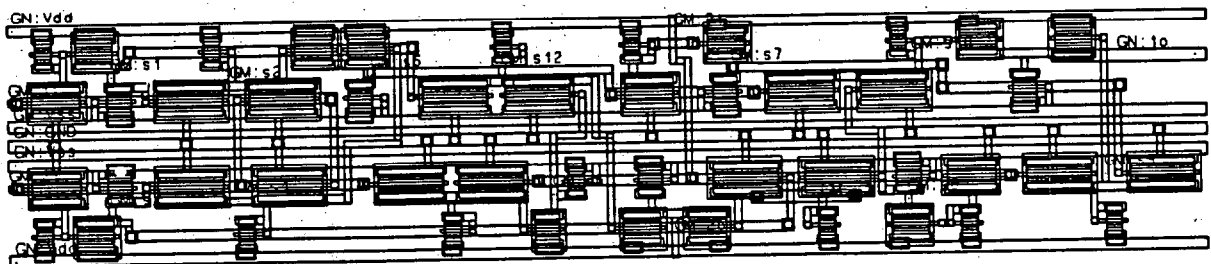
existencia de dos niveles de metalización en la fabricación de los circuitos bajo estudio. Estos niveles se conocen con el nombre de *metal 1* y *metal 2*.

El primero de ellos se emplea para la conexión entre puntos cercanos (por ejemplo, transistor a transistor, como puede observarse en la figura 3.12) así como para metalizaciones de puerta o de los contactos. El metal 2, de menor resistividad que el anterior, se emplea para cubrir grandes distancias, así como para el transporte de señales (bus de datos, reloj...) o conexión a las fuentes de alimentación. Se corresponde con las largas tiras visibles en la figura 3.12.

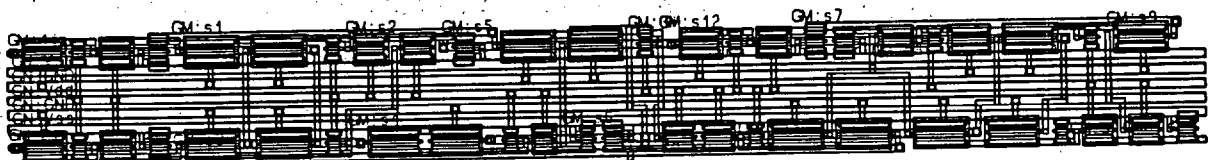
3.3.2.4.- Estilos de Diseño.

El estilo de diseño es aquella especial disposición física (a nivel de *layout*) que de dispositivos (transistores, diodos...) o de puertas o de celdas, que resulta de aplicar algún criterio principal de diseño o restricción tecnológica y que la cual se respeta durante el diseño completo del circuito.

El estilo de diseño adquirido, suele responder a los hábitos personales del diseñador, aunque por lo general, se encuentra estandarizado según sea el circuito a implementar y el tipo de la lógica disponible. Por ejemplo, es clásico disponer los elementos en forma matricial o en tiras de bit adosables en sentido ortogonal a los buses (figura 3.12).



(a)



(b)

Figs. 3.12.a, 3.12.b. Layout del sumador serie de 1 bit utilizando: a) diseño tradicional, b) estructura en anillo.

El modelo de área que se presenta se ha desarrollado para admitir dos estilos básicos de diseño (típicamente empleados por los diseñadores de circuitos GaAs), los llamados *clase A* (ver figura 3.12.a) y *clase B* (ver figura 3.12.b) con sus respectivas variantes.

El diseño denominado *clase A* se corresponde con el estilo tradicionalmente empleado en diseño NMOS.

El estilo perteneciente a la *clase B* es conocido con el nombre de *notación en anillo* [SaArC91]. La inclusión de nuevos estilos no presenta dificultad adicional, siempre que manifiesten regularidad

3.3.2.5.- Área intrínseca y área extrínseca.

Se define como área activa a la ocupada por el transistor. Se recuerda que los circuitos a desarrollar sólo contienen transistores y metalizaciones (no se modelan, por ejemplo diodos o condensadores).

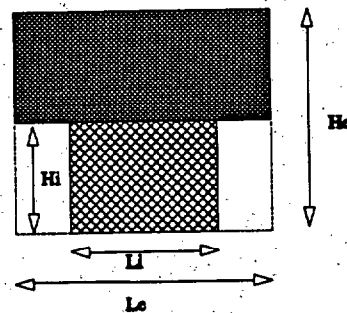
Se conoce como área intrínseca (ver figura 3.13) a la superficie ocupada por el transistor cuyo ancho total incluye el espacio reservado por las reglas de diseño. Se hace notar que el diseñador sólo puede alterar la anchura de los transistores.

Se define como área extrínseca (ver figura 3.13) a la del transistor cuya altura incluye la porción de conexionado correspondiente.

3.3.3.- Metodología.

Con el ánimo de desarrollar una estrategia de optimización, capaz de manejar relaciones área-potencia-retraso, se precisan expresiones analíticas, a ser posible lineales y funciones de la variable a optimizar: el ancho de los transistores.

Merced a la regularidad de los diseños, es sencillo desarrollar un modelo de estimación de área donde se necesitan pocos parámetros de ajuste; la mayoría de ellos pueden extraerse directamente de las reglas de diseño. A medida que el circuito a diseñar crece en tamaño, es por lo general inevitable que paralelamente vaya aumentando la falta de regularidad. En estos casos será preciso introducir parámetros de ajuste.



He: Altura extrínseca.
 Hi: Altura intrínseca.
 Le: Longitud extrínseca.
 Li: Longitud intrínseca.

Fig. 3.13. Estimación del área del transistor.

Básicamente son dos las posibilidades a la hora de elaborar un modelo para la estimación del área ocupada: desarrollarlo a nivel de transistor o a nivel de puerta. A continuación se detallan cada una de las posibilidades y se establece una comparación entre ambas estrategias.

3.3.3.1.- Modelo a nivel de puerta.

Tal como observa Hoppe [HoNeS90] el área activa del transistor crece de forma lineal con el ancho de puerta. A su vez, el área ocupada por n transistores se computa simplemente sumando las áreas individuales de cada transistor. El área ocupada por las líneas de conexión se promedia a través de ajustes. La estimación del área total en estos términos presupone un diseño regular. Este tipo de diseño siempre se encuentra al menos en los bloques más significativos de un circuito.

Así, y siguiendo los esquemas típicamente empleados en CMOS, se estima el área ocupada a través de la expresión {3.12},

$$A(w) = a_1 \cdot w + a_2 \quad \{3.12\}$$

donde w es el ancho del transistor (siempre medido en μm), y los coeficientes a_1 y a_2 son

dependientes de las reglas y estilos de diseño. A través de estos parámetros se incluye además la superficie ocupada por las interconexiones. De cara a la evaluación del área ocupada por las puertas, existirán valores de los coeficientes a_1 y a_2 según sea la puerta en cuestión (NOR, inversor...).

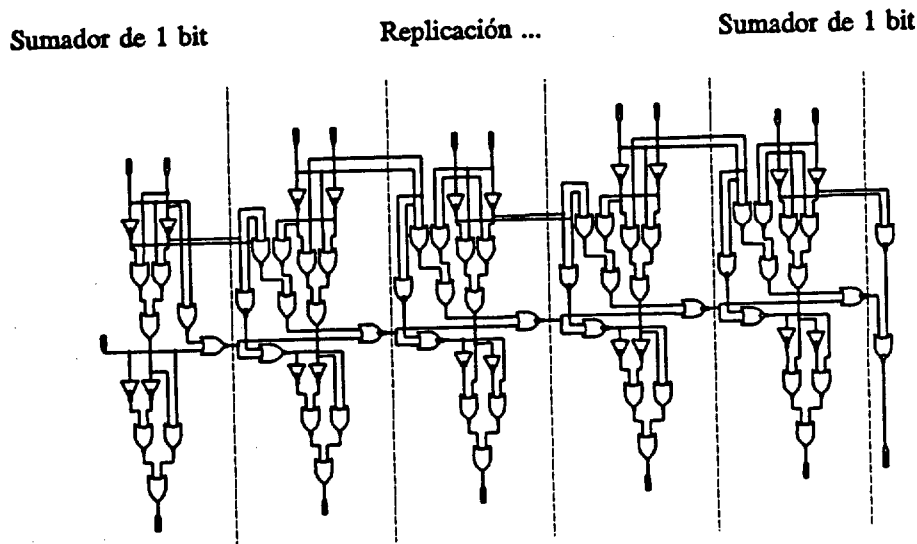


Fig. 3.14. Esquema lógico de un sumador de acarreo anticipado de 5 bits.

La metodología consiste pues en estudiar los diferentes estilos y determinar los valores de los coeficientes de ajuste. En el caso que concierne se estudiarían los dos estilos de diseño admitidos, es decir los llamados *clase A* y *clase B*.

Se enfocará la metodología hacia el estilo de diseño más empleado por los diseñadores de circuitos DCFL/SDCFL, el cual resulta ser el estilo de notación en anillo (o *clase B*). Destaca este estilo por su gran regularidad y compacidad (diseños más fácilmente estructurables), no alcanzadas por el estilo tradicional.

Debido a esto, permite la extrapolación de estimaciones para estructuras de n bits conocida la superficie ocupada por el módulo de 1 bit. La notación en anillo (ver figura 3.12.b) consiste básicamente en situar los buses de alimentación en el centro de la célula, de manera que la línea de alimentación más negativa se sitúe al lado de la lógica.

Con ello se logra un apantallamiento eléctrico, disminuyendo el efecto asociado al ruido de señales debido a las oscilaciones en el bus de alimentación (V_{DD}). Esto es importante en tecnologías de alta velocidad como la que nos ocupa.

A su vez, dicha regularidad facilita el diseño de estructuras mediante simple replicación de celdas. Puede verse en la figura 3.14 el esquemático correspondiente a un sumador de acarreo anticipado (*Carry Look Ahead*) de 5 bits. De la misma forma que se expande el esquemático desde 1 bit hasta n bits, lo hace el *layout* correspondiente. Igualmente ocurre para otros diseños.

Así se observa en la figura 3.15 el plano de base correspondiente al multiplicador paralelo implementado según algoritmo de Booth [SaMoC91]. Cada celda perteneciente a la primera fila es un semisumador, mientras que el conjunto restante se compone de sumadores completos (se añaden mediante simple replicación). La traducción de este plano de base al *layout* es directa.

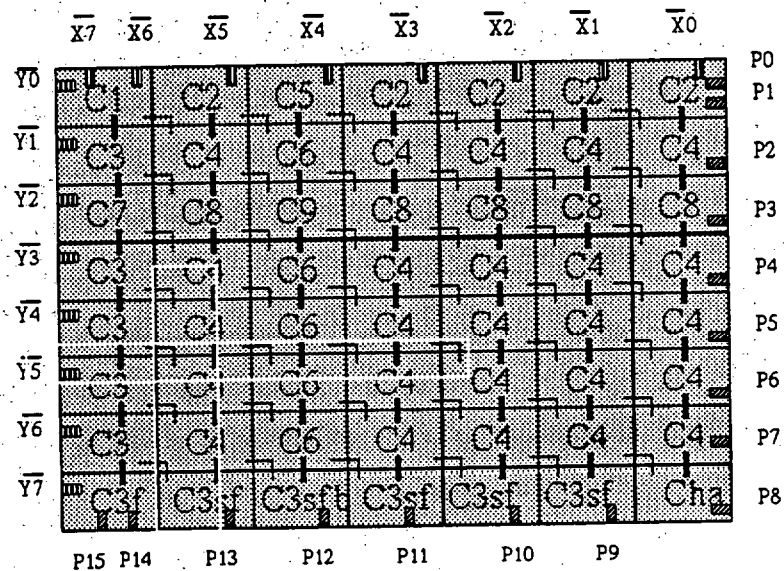


Fig. 3.15. Plano de base del multiplicador de 8x8 bits.

Como expresión para el cómputo del área para los circuitos bajo estudio, se propone la siguiente:

$$A(w) = a_1 \cdot \sigma \cdot (w - a_2) + a_3 \cdot \chi \quad \{3.13\}$$

donde a_1 , a_2 y a_3 , son coeficientes de ajuste dependientes de las reglas de diseño (se calculan a partir de estudio de diferentes *layouts* significativos). Los valores de σ y χ son función del particular estilo de diseño adoptado. A diferencia de la expresión {3.12}, se incluyen varios estilos de diseño (en este ocasión son dos) en una misma expresión.

Por lo general, y apuntando a objetivos de optimización, suele permitirse una sola variable por puerta lógica a efectos de dimensionar la puerta. Ésta puede ser el ancho de puerta de cualquiera de los transistores presentes.

En los circuitos bajo estudio, dicha variable se corresponderá con el ancho de puerta del transistor de *pull-down* de la etapa inversora. En base a esto se han determinado los valores de los coeficientes que aparecen en la expresión {3.13}.

Pueden verse en la tabla 3.8 dichos valores para el caso de la puerta inversor. Las unidades de los coeficientes están puestas en la tabla; los parámetros σ y χ son adimensionales. De esa forma, de la expresión {3.13} se obtiene directamente el área expresada en micras cuadradas (μm^2).

Tabla 3.8. Coeficientes para la puerta inversora SDCFL.

Coeficientes	a_1 (μm)	a_2 (μm)	a_3 (μm^2)	σ	χ
Estilo A	28.8	12	887.04	1	1
Estilo B	28.8	12	887.04	0.7361	1.281

La misma fórmula se aplica al caso de otras puertas lógicas. Así, la puerta NOR, considerada al igual que el inversor, como elemento aislado lleva asociada los valores de los coeficientes que se muestran en la tabla 3.9. En esta ocasión, la variable w viene dada por {3.14},

$$w = w_1 + w_2 + \dots + w_{n-1} + w_n \quad \{3.14\}$$

donde w_i es el ancho de puerta de cada uno de los transistores de *pull-down* de la etapa inversora.

Tabla 3.9. Coeficientes para la puerta NOR SDCFL.

Coeficientes	a_1 (μm)	a_2 (μm)	a_3 (μm^2)	σ	χ
Estilo A	28.8	12	1071.36	1	1
Estilo B	28.8	12	1071.36	0.736	1.187

Dada la regularidad del diseño de estas puertas básicas, los coeficientes de ajuste (obtenidos directamente de las reglas de diseño) permiten el cálculo exacto del área ocupada por cada una de las puertas. Algunas estimaciones del área ocupada por los transistores se presentan en la tabla 3.10.

Tabla 3.10. Área estimada según modelo a nivel de puerta para diferentes puertas lógicas.

Puerta lógica	Área estimada (= = Área medida)
Inversor clase A	887.04 μm^2
Inversor clase B	1136.32 μm^2
Puerta 2-Nor clase A	1416.96 μm^2
Puerta 2-Nor clase B	1526.06 μm^2
Puerta 3-Nor clase A	1762.56 μm^2
Puerta 3-Nor clase B	1780.42 μm^2

Se recuerda el hecho de que se supone regularidad en el diseño, con lo cual resulta razonable estimar el área del circuito completo mediante la simple adición de la superficie ocupada por cada puerta o celda elemental.

Como principal restricción que supone esta forma de evaluar el área se manifiesta el hecho de que en la formulación se ha permitido la existencia de una sola variable: el ancho de puerta del transistor de *pull-down* de la etapa inversora.

Ampliar la metodología con la intención de incluir otras variables (transistores de la etapa *buffer*...), supone caracterizar topológicamente los circuitos con un número mayor de tablas de coeficientes.

Si bien es apropiado este modelo por su exactitud en la estimación del área ocupada, no resulta práctico de cara a la implementación de una estrategia de optimización (donde el aspecto más interesante reposa precisamente en variar las dimensiones de más de un sólo transistor por puerta lógica).

Con el propósito de disminuir el número de datos a manipular, se presenta a continuación una variante de la metodología.

3.3.3.2.- Modelo a nivel de transistor.

El modelo para la estimación de área a nivel de puerta se presenta tremendamente dependiente del estilo de diseño y no tiene en cuenta las anchuras de todos los transistores.

Aunque es cierto que los diseñadores tienden a manejar sólo el ancho de algunos transistores por puerta (dejando los demás fijados a los valores preestablecidos), siempre es más deseable disponer de una metodología que permita tantas variables como transistores existan en el *netlist*. Este argumento adquiere mayor peso cuando se pretende el desarrollo de una estrategia de optimización a nivel de transistor.

El modelo a nivel de transistor que se propone carece de las limitaciones del anterior modelo, permitiendo considerar:

- cambios en todos los tamaños de los transistores (anchos de puerta, dado que la altura está fijada por la tecnología) y,
- es adaptable fácilmente a cualquier estilo de diseño (la regularidad exigida anteriormente pasa a desempeñar un papel secundario).

El modelo que se expone está en la línea de otros semejantes desarrollados para el caso CMOS, pero contempla las particularidades de los estilos de diseño empleados en GaAs. Además, comparte con aquellos el hecho de ser un modelo unidimensional, aunque este vocablo carece del mismo significado que para el caso CMOS.

Se entiende en este contexto de diseño GaAs, que se modela el tamaño del transistor a través de una sola variable: su anchura de puerta (a diferencia de las puertas CMOS, para el caso GaAs la dimensión vertical de los transistores que constituyen una puerta lógica es siempre fija). La forma general para la estimación del área viene dada por la expresión {3.15},

$$\text{Área} = \{ \sum (\text{área cada transistor}) \} \cdot q \quad \{3.15\}$$

donde q es un coeficiente de ajuste determinado a partir del estudio del conjunto de *layouts* representativos [SaArC91], [SaMoC91]. De esta forma basta un coeficiente para introducir diferentes estilos de diseño (no a nivel de puerta, sino a nivel de todo el circuito).

El área de cada transistor se obtiene directamente a partir de la anchura que se haya dado y de las reglas de diseño asociadas a la tecnología particular. Tal como se puede ver en la figura 3.13, se calcula el área por transistor sumando los siguientes términos:

- área ocupada por el transistor de lados L_e y H_i (ver figura 3.13). Este área incluye el espacio destinado a respetar las reglas de diseño de separación entre transistores (distancias mínimas entre el metal 1 de puerta y los respectivos cajones de difusión entre los transistores adyacentes según la dirección horizontal),
- área ocupada por el interconexión asociado. Este área incluye el espacio destinado a respetar las reglas de diseño de separación de líneas de conexión. Asimismo, dependiendo del estilo de diseño, se amplía esta superficie para dar cuenta del espaciado entre transistores en la dirección vertical (ver figura 3.13).

A diferencia de modelos estimativos a nivel de transistor (ver por ejemplo [Marpl89]), se incluye directamente el conexionado asociado a cada transistor.

Siendo w el ancho del transistor (cualquiera de los presentes en el circuito), se estima el área ocupada por cada transistor a través de la expresión {3.16},

$$\text{Área} = \zeta \cdot w + \varphi \quad \{3.16\}$$

donde los parámetros ζ y φ se determinan de las reglas y estilo de diseño. Para el caso que concierne, los valores se corresponden con $22 \mu\text{m}$ y $246.4 \mu\text{m}^2$ respectivamente.

Dado que cada transistor contribuye al área final del circuito con una porción de área debida a las interconexiones (que es proporcional a la anchura del transistor implicado), esta metodología no es fuertemente dependiente de cómo se trazan los buses. Esta característica permite la aplicación de la metodología a diferentes estilos de diseño.

3.3.4.- Resultados.

Se presentan en el apartado 3.3.4.1 algunos resultados obtenidos, indicando en cada caso el error relativo que se comete. Como corolario se define, en el apartado 3.3.4.2, el coeficiente *delta*, que da cuenta de la superficie ocupada por *bit*.

3.3.4.1.- Evaluación del error cometido.

Se ha estimado el área de un número significativo de circuitos. En la tabla 3.11 se presentan algunos resultados correspondientes a *layouts* en estilo *ring notation*. La estimación tuvo lugar empleando el modelo a nivel de transistor, dado que se pretendía que el error cometido se correspondiera con lo que sería un peor caso *típico*.

Puesto que la estimación a nivel de puerta es más precisa (y requiere un número mayor de parámetros), puede emplearse allí donde sea necesario un refinamiento en la estimación. Para los objetivos planteados en esta tesis, el error que se obtiene usando el modelo a nivel de transistor es totalmente aceptable.

El error relativo que se obtiene, se sitúa en torno al 5% sobre los casos estimados (sin superar en ninguna ocasión el valor del 10%).

Tabla 3.11. Resultados y error estimado.

Circuito	n° transistores	Área estimada (mm ²)	Área postlayout (mm ²)	Error (%)
Flip-Flop D	31	0.00987	0.0100	+ 1.30
Flip-Flop T	32	0.0103	0.0104	+ 0.96
Sumador 4 bits	217	0.1036	0.101	- 2.50
Sumador 5 bits	272	0.130	0.123	- 5.69

La presencia de una cota de error tan pequeña en la estimación para el caso de los *flip-flops*, es producto de la regularidad con que fueron diseñados. Esta regularidad es también manifiesta en el diseño de las celdas que conforman los sumadores. El aumento de error que se obtiene para el caso de los sumadores (que sigue siendo admisible), debe ser atribuido a la inevitable pérdida de regularidad consecuencia del agrupamiento de celdas.

Como comentario adicional se añade el hecho de que, la propia medida del área de *layouts* de talla considerable, es asimismo fuente de error, puesto que no se encuentra de ningún modo establecido cuál es exactamente el área a medir.

Se aclara esto indicando que, se comparan los resultados de la formulación presentada

con los que se miden del *layout*, los cuales contienen, en algunas ocasiones, espacios vacíos, o líneas de interconexión replicadas o prolongadas. Todos ellos son factores a aumentar el área (que se calcula sin más que multiplicar la base por la altura del *layout*, sin restar las porciones vacías).

Con esto no se quiere decir que no se tiene en cuenta la posible existencia de estos huecos; su presencia se ha incluido en la metodología a través de los diferentes parámetros de ajuste, que en definitiva, modelan la existencia de estas regularidades (los resultados obtenidos confirman, una vez más, la validez de estos coeficientes).

3.3.4.2.- Definición del parámetro *delta*.

En aquellos casos en los que se trata con estructuras de diseño altamente regular, como atañe por ejemplo a sumadores y multiplicadores, puede definirse un parámetro *delta*, según expresa {3.17}.

$$\delta = \frac{\text{Área ocupada}}{\text{bit}} \quad \{3.17\}$$

que para el caso de los sumadores presentados, se corresponde con el valor de:

$$\delta = 0.024 \text{ mm}^2.$$

Haciendo uso de este coeficiente, se puede realizar una estimación grosera del área ocupada por sumadores de mayor número de *bits*. Así, según este proceder, se obtienen los resultados que se muestran en la tabla 3.12. La comparación con el área ocupada por el *layout* físico manifiesta la utilidad del parámetro *área / bit*.

Tabla 3.12. Resultados y error relativo.

Circuito	Área estimada (mm ²)	Área medida (mm ²)	Error relativo (%)
Sumador 9 bits	0.216	0.205	≈ 6
Sumador 16 bits	0.384	0.330	≈ 16

Como era de esperar, el error cometido ha aumentado respecto a los presentados en la tabla 3.11. Merced al carácter meramente estimativo, puede decirse que los resultados siguen siendo buenos, pudiendo servir como primera referencia a la hora de tomar las primeras decisiones de diseño.

Este parámetro aquí definido encuentra además aplicación a la hora de evaluar el índice de prestaciones de un diseño completo [SaArC91]. Este índice relaciona el número de puertas lógicas, la frecuencia máxima de funcionamiento y el área utilizada, proporcionando una medida de la densidad de computación dada en puertas·GHz/mm².

3.3.5.- Conclusiones.

Se concluye la presentación de este modelo para la estimación del área para circuitos implementados en las familias DCFL/SDCFL. Se califica como metodología robusta y de fácil aplicación.

A su vez, es tecnológicamente independiente, y requiere de un conjunto pequeño de parámetros que se obtienen directamente a partir de las reglas de diseño y de los principales estilos de diseño a emplear.

La formulación permite la rápida estimación a nivel de *pre-layout* de circuitos lógicos (no importando si son combinacionales o no). Dado el carácter sencillo y el manejo de expresiones matemáticas exentas de complicación, es adecuada para incorporarla en estrategias de optimización (o en estrategias de compactación topológica del *layout*).

Conjuntando el modelo de área con el modelo de potencia presentado en el capítulo 3.2, resulta sencillo explorar prestaciones área-potencia, en fase *pre-layout*, para los circuitos bajo estudio. Los compromisos área-potencia-retardo, siempre más complicados de establecer, serán analizados en el capítulo 5.

CAPÍTULO 4

TÉCNICAS DE "BUFERIZACIÓN".

Se dedica este capítulo al desarrollo de técnicas de "bufeización" de ámbito de aplicación local, para circuitos combinatoriales DCFL/SDCFL, que permitan la optimización de los diseños según criterios de consumo de potencia, retraso y área ocupada, todo ello circunscrito a las condiciones impuestas por el margen de ruido. El primer paso que daremos será estudiar la optimización local del circuito (o circuitos) básicos de esta familia lógica. De los estudios realizados, se derivará un conjunto de reglas de diseño que encontrarán su aplicación una vez integradas en el algoritmo de optimización global (a presentar en el capítulo 5). No se deben esperar logros espectaculares de las prestaciones de los circuitos sometidos a estas técnicas, como quedará ampliamente justificado. Por último se plantean posibles mejoras de la familia lógica en cuestión mediante la elección de otros valores para las fuentes de alimentación.

4.1.- INTRODUCCIÓN.

Ciertamente la optimización de un circuito debe irse haciendo a todos los niveles: algoritmo, arquitectura, estructura de bloques básicos, estructura lógica, estructura circuital, estructura geométrica y física. Las decisiones tomadas a niveles altos del proceso de diseño tienen mayores consecuencias en las prestaciones resultantes. Como se indicó en el capítulo 1, nuestro estudio se ha centrado en el nivel circuital, muy variado en estas tecnologías.

El diseño de circuitos digitales GaAs constituye un problema multidimensional en el que, del conjunto de variables a considerar, son pocas las que poseen un valor aceptado de forma estándar.

Así y a diferencia de lo que acontece para las familias lógicas de silicio, algunas variables, como los valores de las fuentes de alimentación o el tipo de *buffer* a emplear, no se encuentran todavía establecidas de forma estándar (muy diferente del caso CMOS, donde los cinco voltios de las fuentes se emplean desde hace más de una década). Esto da lugar a que el diseñador navegue entre un sinnúmero de posibilidades (tanto a nivel eléctrico y circuital como lógico) a la hora de tomar una decisión de cara a mejorar el diseño a realizar.

Uno de los objetivos de esta tesis radica en constreñir este amplio conjunto de posibilidades, presentando al diseñador los criterios adecuados a cada situación de compromiso.

De acuerdo con este objetivo, y una vez definida la implementación lógica, el siguiente paso lo constituye el dimensionado de transistores. Para el caso CMOS, el dimensionado de transistores lleva parejo el dimensionado de los *buffers*, dado que éstos son asimismo inversores.

La elección del tipo de *buffer* a conectar a la salida de la puerta DCFL ha sido discutida en el capítulo 2 de esta tesis. La configuración circuital de este tipo de *buffer* no se corresponde con la del simple inversor DCFL, sino que resulta ser, en términos eléctricos, un seguidor de fuente. La geometría de los transistores que lo integran no se deduce directamente de la de los transistores de la etapa lógica de la puerta. El caso bajo estudio presenta una complejidad superior al que se resuelve para circuitos CMOS. Su resolución requiere dos fases claramente diferenciadas:

- elección de la geometría del *buffer* y,
- dimensionado de los transistores que componen la etapa

lógica de la puerta, es decir, los transistores de *pull-down* de dicha etapa (se recuerda que el ancho del transistor de *pull-up* de la etapa inversora se considera fijo).

Este capítulo se dedica a la primera de las fases, y aporta técnicas de *buffering* para lógicas DCFL. la segunda fase del proceso de optimización se trata de forma analítica en el siguiente capítulo. En él se aportan técnicas de *sizing* en DCFL.

4.2.- ESTRATEGIA GENERAL A SEGUIR.

La mejor solución que se puede ofertar al diseñador es siempre una solución analítica, donde se tenga un completo entendimiento de cada uno de los parámetros que modelan el comportamiento de la puerta lógica. Sin embargo, los modelos que se ofrecen en esta tesis, son de naturaleza pseudoempírica, donde se manejan expresiones polinómicas cuyos coeficientes no poseen un claro significado físico. En contrapartida las expresiones que damos modelan las puertas cuantitativamente con un alto grado de precisión (errores menores del 10 por ciento).

La estrategia a seguir consistirá en el estudio de las estructuras o redes lógicas del diseño con estos modelos para decidir en cuanto a criterios de consumo de potencia, área ocupada y retraso se refiera. Dado que el cómputo de estas expresiones es inmediato, pueden rastrearse múltiples combinaciones interesantes. Del conjunto de datos a obtener, se construirán tablas de información donde se ofrece al diseñador la solución local al problema que plantea. Para el delicado análisis del ruido, se empleará simulación numérica, HSPICE, [HSPIC90] sobre el conjunto de *buffers* preseleccionados.

Pese a que los niveles de tensión de las fuentes de alimentación se han definido en el capítulo 2, dado que se está en disposición de realizar un análisis de posibilidades, se presentará un breve estudio implicando dichas variables. Para ello será preciso realizar simulación vía HSPICE, puesto que no se dispone del conjunto de coeficientes necesarios.

4.3.- TÉCNICAS DE "BUFERIZACIÓN".

Se designa bajo este nombre, el conjunto de reglas de optimización local, dirigidas a determinar según criterios conocidos, el conjunto de *buffers* óptimos y su ubicación física en el circuito según sea la carga a soportar por los nodos de salida de las puertas. Es objeto de

este estudio pues,

- determinar el conjunto de *buffers*¹ adecuado a los requerimientos impuestos por el margen de ruido,
- deducir bajo qué condiciones de carga (*fan-out*), resulta más apropiado el empleo de un determinado buffer.

Todo ello se estudia a nivel de células primitivas o de estructuras completas y aisladas (sumadores, multiplicadores, decodificadores...). Se incluye en este estudio el problema asociado a la cargabilidad de los buses, la distribución de la señal de reloj y el reparto de su carga. Sin embargo se excluye el estudio de los interfases de entrada-salida (*pads*).

4.3.1.- Necesidad de las técnicas de "bufeización".

Todo circuito complejo requiere una señal de reloj que ha de ser distribuida por todo el *chip*. Este hecho da lugar a que, en determinadas partes del circuito se pueden encontrar nodos sometidos a una gran capacidad de carga, lo cual puede generar problemas de *clock-races* y *skews*. La presencia de estos problemas puede evitarse a través de una correcta "bufeización" de los nodos sometidos a esas condiciones de carga.

Para el caso de los circuitos CMOS, como regla general, se puede decir que se requiere un *buffer* cada vez que algún nodo del circuito ve una capacidad muy superior a la del bloque que lo ataca. Esta situación tiene lugar para elementos básicos de un circuito digital, como buses, reloj y *pads* de salida. Para los circuitos que se estudian, se precisa colocar un *buffer* en todos los nudos si lo que se pretende es un circuito robusto que muestre buen comportamiento bajo posibles condiciones extremas (entiéndase altas temperaturas).

En el capítulo 2 (apartado 2.1.2) se propuso una topología para el *buffer* así como una geometría. Pese a que en dicha sección se justificó el excelente comportamiento eléctrico mostrado por este particular *buffer*, no se realizó un análisis comparativo que determine el margen de optimización posible. Es objetivo de este capítulo poner de manifiesto las características ofrecidas por este *buffer* en comparación con otros posibles candidatos. Pese a ello, y a sabiendas de la dificultad que el problema entraña, se pretende ampliar el conjunto de geometrías posibles.

¹ Se entiende por conjunto de *buffers* al conjunto finito de posibles geometrías para los transistores que integran el seguidor de fuente.

4.3.2.- Diseño robusto y diseño adaptativo.

De las familias lógicas más versátiles disponibles comercialmente destaca sobremanera la DCFL. Razones fundamentales para ello son, entre otras, la sencillez de la topología de las puertas, su bajo consumo de potencia y su alto nivel de integración (actualmente niveles VLSI).

Como principal desventaja destaca el estrecho *logic-swing* (o excursión de señal) a que da lugar, que se traduce directamente en muy poca inmunidad al ruido y alta sensibilidad a la carga. Aunque estáticamente la cargabilidad de salida es buena, su comportamiento dinámico (velocidad de conmutación) se reduce en un factor de $1/n$, donde n indica el número de puertas conectadas al nodo de salida [Sarmi91]. Esto implica que el retraso asociado a la puerta, crece cuando aumenta la carga del nodo de salida, es decir, con el *fan-out* y con la longitud de las líneas de interconexión.

Estas circunstancias mencionadas anteriormente quedan paliadas con poca penalización en área mediante el empleo del buffer. Según sea la estrategia de empleo del *buffer* clasificaremos los diseños en:

- robusto,
- adaptativo.

Denominamos diseño robusto a aquella metodología de diseño consistente en colocar *buffers* a todas las puertas del circuito. Además, se concibe una geometría fija para el *buffer* por tipo de puerta (o una sola geometría para todo el circuito). Según este proceder, un diseño basado en n puertas NOR y m puertas inversoras, contendría n *buffers* tipo 'A', y m *buffers* tipo 'B' (donde 'A' se corresponde con el *buffer* asociado a la puerta NOR y 'B' a la puerta inversora).

Denominamos diseño adaptativo a aquella metodología de diseño que dispone *buffers* según sean las condiciones de carga de los nodos. Queda incluida en la definición la posibilidad de nodos de salida sin *buffer*.

Se da por supuesto el hecho de que en ambos casos, los *buffers* empleados (o la no presencia de *buffer* alguno), garantizan el correcto funcionamiento del circuito bajo condiciones normales y extremas admitidas. La diferencia radical entre ambas estrategias, se traduce en comodidad y rapidez en el diseño. Así, un diseño según una estrategia adaptativa requerirá una implementación más cuidadosa demandando un mayor tiempo de desarrollo. Dado que la celeridad en la terminación de un diseño constituye en gran número

de casos un factor prioritario, no extraña que el diseñador se decante por el diseño robusto, al menos para las primeras versiones o series de circuitos. Sin embargo, el diseño adaptativo goza siempre de una mejora sustancial en sus prestaciones, y a él debe tenderse cuanto antes en el ciclo de vida de un producto.

La mejora en dichas prestaciones ha de entenderse en cuanto al consumo de potencia y área se refiere (el término velocidad del circuito no es fácil de aumentar mediante las técnicas aquí expuestas), así como a su estabilidad en temperatura y ruido.

Puesto que el consumo de potencia (estática en el caso GaAs) constituye un factor sobremanera crítico, todo esfuerzo en disminuir su valor está permitido (incluso a pesar de que ello conlleve una pérdida moderada de velocidad del circuito).

4.4.- ELECCIÓN DE UN CONJUNTO DE *BUFFERS*.

Como paso previo al estudio comparativo de los diversos *buffers* posibles (entiéndase geometrías posibles), se impone la preselección de éstos; preselección que se establece en función de los requerimientos funcionales. Como criterio prioritario destaca el margen de ruido. En base a este importante parámetro serán excluidos del conjunto de *buffers* diferentes candidatos.

4.4.1.- Circuito estudiado y condiciones de medida.

El circuito bajo estudio (inversor SDCFL) es el que se muestra en la figura 4.1. Como objetivo se plantea seleccionar el *buffer* óptimo para el inversor DCFL. Se entiende óptimo en el sentido de:

- respetar el margen de ruido de seguridad,
- ocupar menos área,
- presentar un bajo consumo de potencia,

todo ello condicionado a un retardo por puerta moderado.

La metodología a seguir se sustenta en la estimación de las variables arriba mencionadas. Esta estimación se ha llevado a cabo a través de los modelos que para tal fin se han desarrollado (capítulo 3). Las condiciones de medida (ver figura 4.2), se corresponden

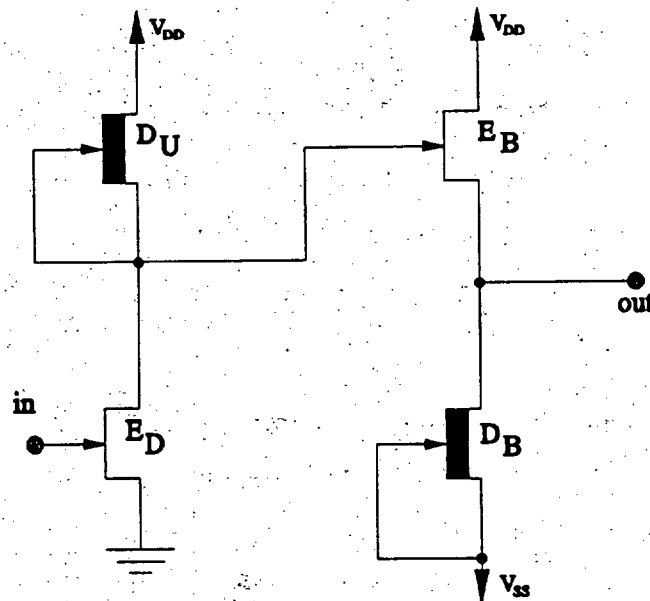


Fig. 4.1. Inversor SDCFL.

con las encontradas en los circuitos en condiciones típicas de funcionamiento, rampas de excitación del orden de 100 picosegundos (frecuencias de aproximadamente 1 GHz). Los valores de tensión de las fuentes de alimentación se han fijado (capítulo 2) a los valores de 1.4 V para la fuente V_{DD} y de 0.2 V para la fuente V_{SS} . En la sección 4.5 se estudia el margen de variación de estos valores.

Para la geometría de los transistores que conforman la etapa lógica de la puerta, se han respetado los mismos que se detallan en el capítulo 2, es decir, cuatro micras (valor del ancho de puerta) para el transistor de *pull-up* y doce micras para el transistor de *pull-down*. El empleo de otras dimensiones para el transistor de *pull-down* no modifica cualitativamente los resultados.

Condición especial presenta la cargabilidad de las puertas a estudiar. Se ha elegido de forma que sea máxima. Como *fan-out* máximo se admite hasta cuatro puertas cargando un nodo de salida. Esto no constituye una limitación para el caso de alguno de los *buffers* dado que, por ejemplo, el *buffer* 9.2/6 (entre otros) soporta hasta un *fan-out* de seis (según condiciones de carga).

Dado el fuerte carácter asimétrico mostrado en las transiciones de las señales (entre transición de estado alto a bajo y la opuesta correspondiente), no resulta sencillo dimensionar de forma óptima la geometría del buffer. Se puede realizar un sencillo estudio comparativo entre la mejora a obtener en el retardo según la variable anchura de puerta, sin más que

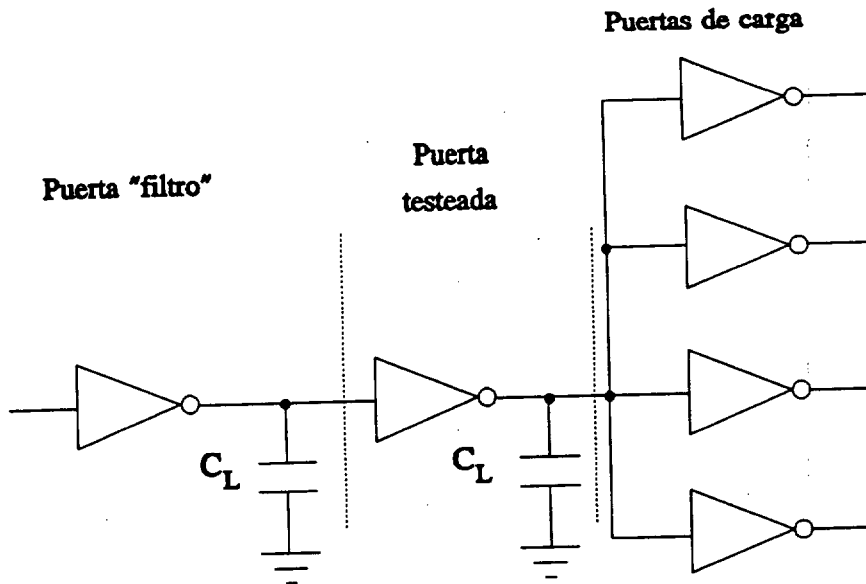


Fig. 4.2. Condiciones de medida.

derivar la expresión del retardo {3.1} respecto a dicha variable (la derivada de D_{HL} tiene un valor negativo pequeño mientras que la derivada de D_{LH} es positiva y tiene un valor superior a la anterior).

Consecuencia de ello es que una determinada geometría resulte mejor (por ejemplo, más rápida) para un tipo de transición que para la otra. La forma de solventar esta cuestión viene guiada por la consideración del retardo de propagación promedio (suma de los retardos por transición dividido por dos).

Mención especial requiere el tratamiento del *fan-out*. Dado que los niveles de tensión (y por consiguiente el margen de ruido) a la salida de la puerta lógica son función de la carga vista por el nudo de salida, una particular geometría puede resultar óptima para unas condiciones de carga asimismo particulares.

Se concluye por tanto que es necesario un análisis de la puerta inversora bajo condiciones de carga diferentes. Se sacrificarán los casos en que no se respeten los niveles lógicos adecuados para el margen de ruido requerido.

4.4.2.- Limitación del número de geometrías a estudiar.

El número de posibles geometrías (infinitas) se encuentra afortunadamente restringido.

Tabla 4.1. Restricciones debidas al margen de ruido y/o potencia y/o retardo.

Buffer	V_{OH} (mV)	V_{OL} (mV)	Potencia (μ W)	Retardo (ps)
9.2/6	500.6	-24.40	540.27	241.12
9.2/4	517.1	≈ 0	507.26	[311.10]
9.2/11	[380.8]	-64.37	628.77	173.45
8/6	[472.2]	33.37	513.55	242.88
20/6	535.1	21.82	[747.58]	240.55

En la tabla 4.1 se muestran resultados de simulaciones realizadas fijando las dimensiones de uno de los transistores del seguidor de fuente. En las filas primera, segunda y tercera se fija la dimensión del transistor de *pull-up* al valor $9.2 \mu\text{m}$. En las restantes se fija el transistor de *pull-down* al valor $6 \mu\text{m}$. Las conclusiones que se derivan de los valores estacionarios de tensión y de los de la potencia disipada son los siguientes:

- con el *pull-up* a $9.2 \mu\text{m}$:
 - el valor mínimo de la anchura del canal del transistor de carga del seguidor de fuente es $4 \mu\text{m}$,
 - el valor máximo para el transistor de *pull-down* resulta ser de $10 \mu\text{m}$ (con $w = 11 \mu\text{m}$ el nivel alto es inferior al umbral permitido),
- con el *pull-down* a $6 \mu\text{m}$:
 - no se debe alcanzar el valor $w = 20 \mu\text{m}$ para el transistor de *pull-up*, ya que la potencia disipada se dispara,
 - la dimensión del *pull-up* debe ser superior a $8 \mu\text{m}$ ya que el valor de tensión alto es inferior al umbral.

Este procedimiento en la definición de las cotas es el que se emplea en el resto de los casos. Las restricciones podrán variar según sea la combinación *pull-up*, *pull-down* presente. Se ilustran estos aspectos restrictivos en la tabla 4.1. Se aprecia en dicha tabla cómo afecta al nivel alto de tensión la geometría del *buffer*. A diferencia, el nivel bajo de tensión muestra ligeras fluctuaciones nada significativas (se garantiza siempre el corte del transistor de la siguiente etapa). Tal como se expone arriba, se descarta la geometría de 4 micras para el transistor de *pull-down* debido al retardo a que da lugar ($\approx 30\%$ mayor que empleando el tipo 9.2/6). Asimismo se descarta la geometría implementada con un *pull-up* de 20 micras debido al alto consumo de potencia parejo a su uso ($\approx 38\%$ mayor que empleando el tipo 9.2/6). Los valores de V_{OH} y/o potencia y/o retardo entre corchetes hacen referencia a

aquellas geometrías que violan el margen de ruido y/o dan lugar a consumos de potencia o retardos elevados.

4.4.3.- Diseño robusto: elección de la geometría del *buffer* a emplear.

Del análisis de las posibles geometrías se obtiene la figura 4.3. En dicha ilustración se ha tomado como parámetro el ancho de transistor de *pull-down*. Se ha representado además varios casos que violan las restricciones expuestas anteriormente (con el fin de aclarar su exclusión).

El punto de corte de la curva con el valor de 500 mV indica que no se viola el nivel alto de tensión permisible. Justamente coincide este valor para el caso del *buffer* 9.2/6 (se advierte la no linealidad de la gráfica). Para la topología que tiene como anchura de *pull-down* 9 micras, dicho punto de corte se alcanza para un transistor de *pull-up* de aproximadamente 15 micras.

Resulta de particular interés observar el estrechísimo margen de maniobrabilidad que ofrece el tipo de seguidor de fuente bajo estudio. Tal es así que se espera que la optimización mediante técnicas de "buferrización" conlleve unos resultados para nada espectaculares (ver apartado 4.4.4.3).

Puesto que la gráfica 4.3 ha sido obtenida bajo condiciones de peor caso, la elección del *buffer* adecuado a partir de la misma (a emplear en la estrategia de diseño robusto) garantiza el correcto funcionamiento eléctrico de los diseños a implementar. Dado que el consumo de potencia constituye un factor altamente crítico, se requiere realizar el correspondiente análisis a título comparativo.

Se muestra en la figura 4.4 la dependencia funcional del consumo de potencia por puerta en función del ancho del transistor de *pull-up* del *buffer* (las condiciones de medida coinciden con las anteriores). El comportamiento es prácticamente lineal con un consumo creciente según el ancho del transistor (y creciente con la dimensión del *pull-down* si se considera a éste como parámetro). Se ha fijado la dimensión del transistor de *pull-down* al valor de 6 micras.

Se han diferenciado dos zonas, la primera denominada *de consumo moderado alto*, y la segunda *consumo alto*. La demarcación se estableció en el valor de 600 μW (se recuerda que se representa un peor caso, que por otra parte, no es el típicamente encontrado en los

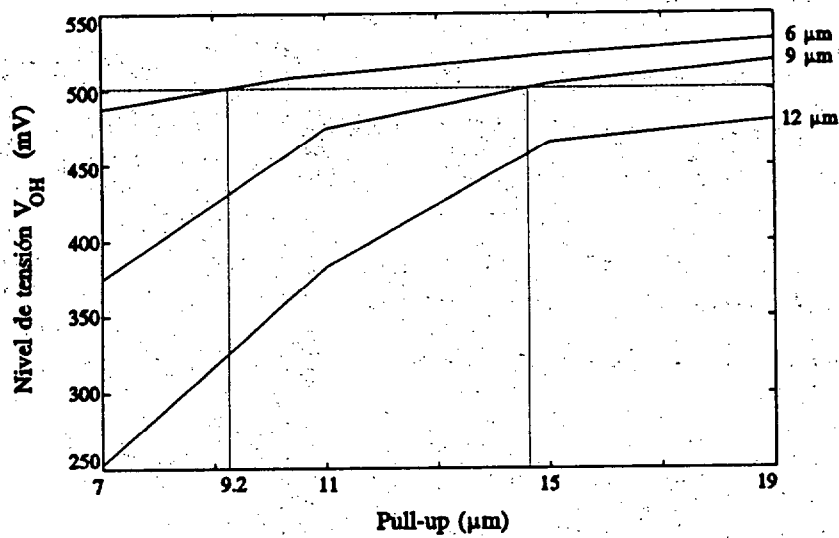


Fig. 4.3. Dependencia de V_{OH} con la geometría del buffer.

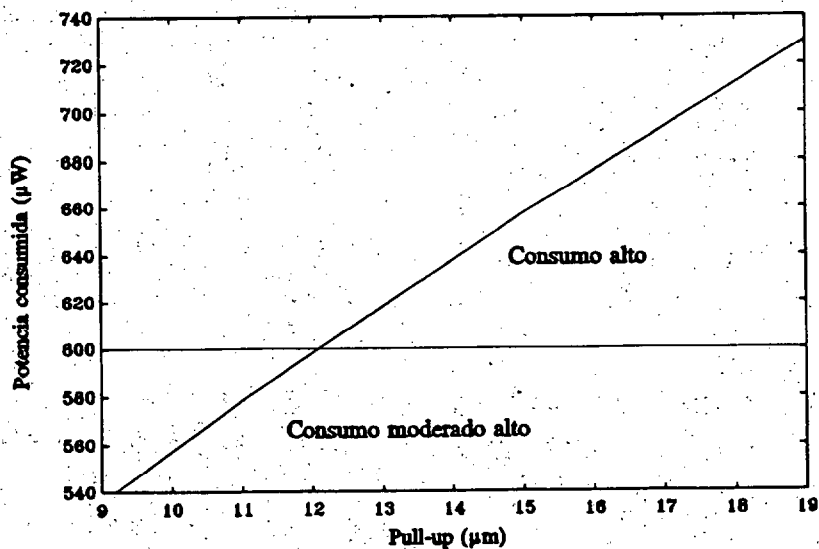


Fig. 4.4. Consumo de potencia en función de la dimensión del pull-up.

diseños). La geometría que dispone un *pull-down* de 9 micras (no contemplada en dicha figura) manifiesta un excesivo consumo de potencia (su representación se ubica, casi por completo, dentro de la zona de *consumo alto*). Otras configuraciones muestran la misma linealidad.

La conjunción de esta gráfica con la anterior (figura 4.3), invita a centrar la atención en la geometría 9.2/6. Tal como se aprecia en la gráfica, el consumo para dicha

configuración es marcadamente bajo, siendo su valor de $540 \mu\text{W}$ aproximadamente.

Como último y decisivo criterio de selección, se realiza el análisis comparativo del ya reducido número de geometrías en base a los retardos de propagación. Una vez más el estudio tiene lugar en las mismas condiciones de medida. Los resultados se ilustran en la figura 4.5.

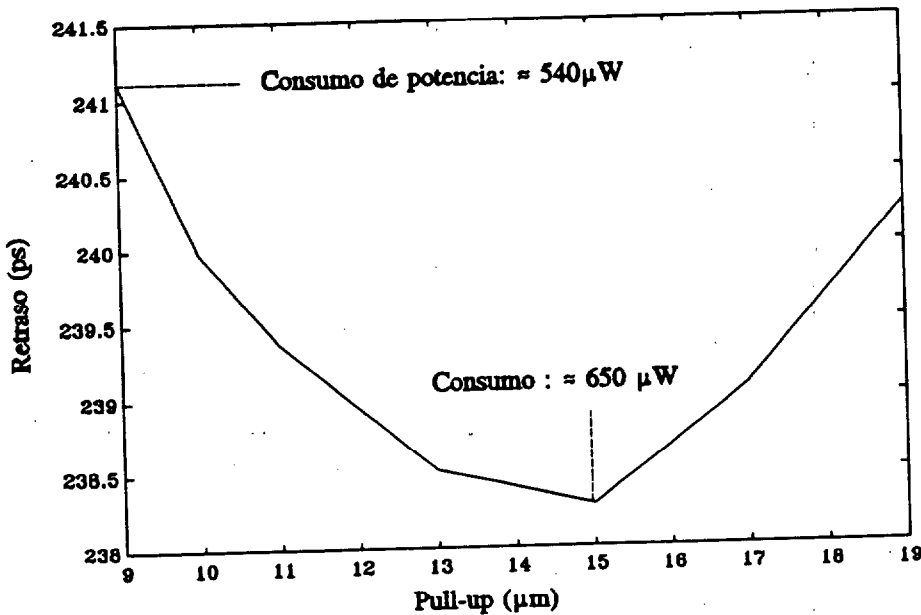


Fig. 4.5. Dependencia del retardo de propagación con la dimensión del pull-up.

Tal como se observa en la figura 4.5, el retardo de propagación alcanza un valor mínimo (para una dimensión de 15 micras). Para geometrías mayores la puerta vuelve a hacerse más lenta. En principio no debiera de suceder esto, dado que al disponer de mayor corriente la transición de carga ha de ser más rápida (se ha fijado el ancho de puerta del *pull-down*). La razón radica en que, si bien dicha transición se acelera, no sucede lo mismo para la transición de descarga, que se vuelve más lenta (se ha de desalojar asimismo más corriente a través del transistor de *pull-down*). Una vez más aparece la particularidad de las transiciones fuertemente asimétricas de las puertas SDCFL/DCFL.

Además a esta explicación se ha de adosar otro factor que contribuye a que ambas transiciones (carga y descarga) se ralenticen. Consultando la figura 4.3, se aprecia cómo se incrementa el valor de tensión V_{OH} con la dimensión del transistor de *pull-up*. Por consiguiente, al ser mayor el *logic-swing*, también es mayor el tiempo necesario para que se alcance la tensión del punto de medida del retardo (0.2311 voltios).

Se aprecia cómo a pesar del importante rango de geometrías consideradas (de 9.2 micras hasta 19 micras) apenas se logra una mejora en el retardo por puerta. Se consigue sólo una mejora máxima en dicho retardo de hasta 2.8 picosegundos. Esto supone escasamente una mejora del 0.5% (respecto del caso 9.2/6) con una importante incremento en el consumo de potencia del orden de 21.70%.

Tras los análisis efectuados, claramente el tipo de dimensiones a emplear en el marco que denominamos *diseño robusto* resulta ser el tipo 9.2/6.

En una estrategia de diseño robusto basado en fijar un *buffer source follower* (seguidor de fuente) optimizado a cada puerta DCFL, es imprescindible que la sensibilidad al *fan-out* sea baja. En la figura 4.6 puede apreciarse que el *buffer* 9.2/6 es poco sensible al *fan-out* y resulta idóneo para el diseño.

Ello quiere significar que, por una parte las conclusiones (entiéndase potencia-retardo) a las que se ha llegado pueden ser extrapoladas al caso en el que se ataca un *fan-out* más bajo. La otra razón (directamente relacionada con la anterior) viene del hecho de que el nivel de tensión V_{OH} es, prácticamente fijo con independencia de la carga (a diferencia de otras configuraciones que muestran una mayor fluctuación en el valor de dicha tensión).

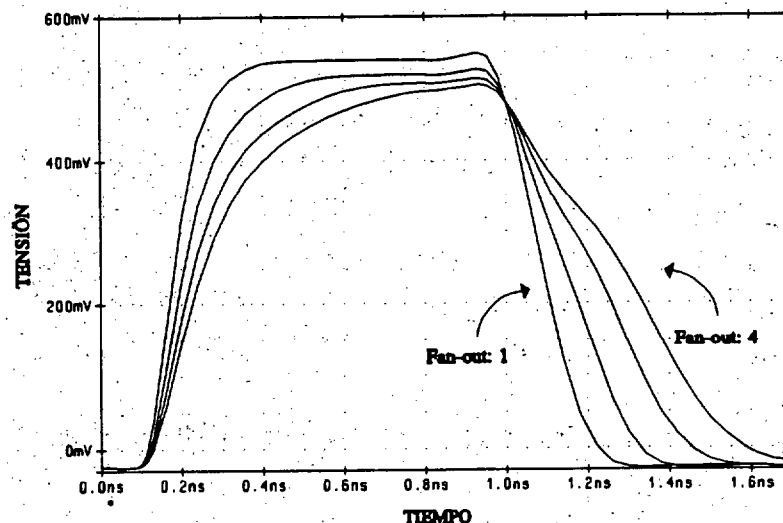


Fig. 4.6. Dependencia de la respuesta transitoria con respecto del fan-out (buffer 9.2/6).

4.4.4.- Diseño adaptativo: criterios de elección y conjunto de *buffers* a considerar.

Se procede en este apartado a desarrollar unos criterios de elección del tipo de dimensionado del *buffer* a disponer en el nodo de salida de las diversas puertas lógicas con la finalidad de mejorar cualquiera de los siguientes aspectos: área ocupada, consumo de potencia y retardos de propagación (sabida la dificultad de lograrlo).

A diferencia del apartado anterior, no es objetivo el seleccionar un único *buffer* capaz de operar en condiciones de peor caso para todo tipo de carga en el nodo de salida (y todo tipo de *fan-in* en el nodo de entrada). Lo que se pretende es ofertar un abanico de posibilidades a elegir entre ellas de acuerdo con los intereses del diseñador. Se da por sentado que los elementos integrantes de ese abanico garantizan el correcto funcionamiento del circuito en las apropiadas condiciones (por ejemplo, si un determinado tipo de seguidor de fuente soporta hasta un *fan-out* de tres, en estas condiciones se ha de utilizar tal seguidor). El diseño de cada puerta se adapta a las condiciones de carga de cada nodo del circuito.

Se distinguen básicamente dos criterios que presentan conflicto: consumo de potencia (a él va parejo el consumo de área) y retardo de propagación. Se comienza esta discusión con el primero de ellos.

4.4.4.1.- Elección de la geometría guiada por el consumo de potencia.

En este caso la estrategia consiste en elegir determinada geometría en aras de disminuir el consumo de potencia (paralelamente disminuirá la superficie ocupada).

Tal como se ha expuesto en el apartado 4.4.3, el consumo de potencia disminuye con las dimensiones de los transistores que integran el seguidor de fuente. Puesto que al disminuir el consumo se ralentiza el diseño, se limitará el consumo mínimo de forma que las prestaciones en cuanto a velocidad del circuito no se vean degradadas en exceso.

Ello obliga a encauzar la búsqueda dentro de la zona denominada *consumo moderado bajo* (no representada en la figura 4.4), la cual comprende aquellos consumos contenidos en el rango de entre 350 μW hasta 540 μW .

Se presenta en la figura 4.7 el consumo de potencia de la puerta lógica inversora SDCFL en función del ancho de puerta del transistor de *pull-up* para un *fan-out* de 1 a la salida de la puerta. Como parámetro se ha tomado el ancho de puerta del transistor de *pull-*

down (expresado igualmente en micras).

Los puntos de ruptura que aparecen en la representación indican las geometrías que satisfacen los márgenes de ruido (por lo tanto aptas). Por ejemplo, se observa que si se selecciona un *pull-down* de 6 micras, no se permite la elección de un *pull-up* menor de 6 micras. Se ha indicado en dicha figura (a título de referencia) la configuración para el seguidor de fuente correspondiente al diseño robusto.

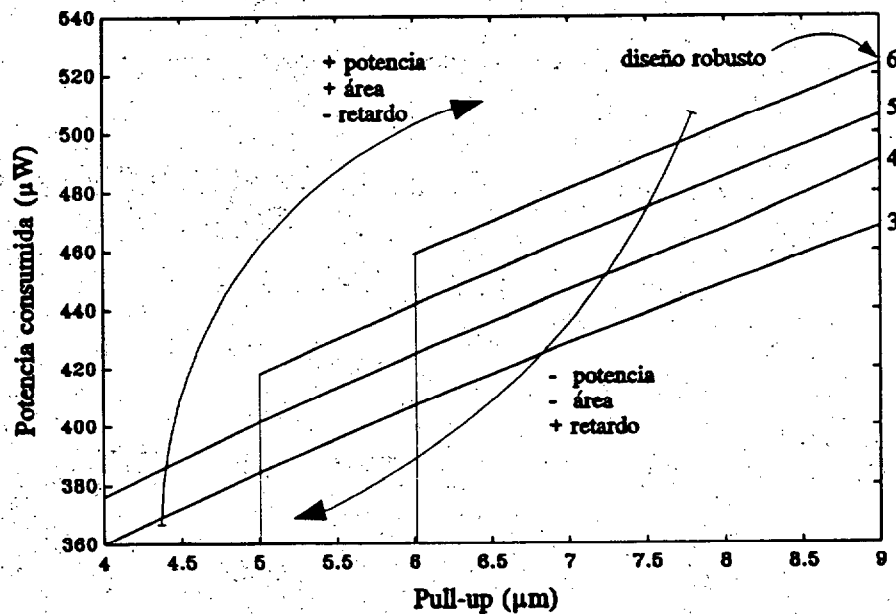


Fig. 4.7. Consumo de potencia en función de la geometría del buffer, fan-out = 1.

El sentido de las flechas da cuenta de cómo evolucionan las características de la puerta en cuanto área ocupada, potencia consumida y retardo se refiere. Mediante esta gráfica, el diseñador puede optar por la elección del dimensionado (dentro de un conjunto moderadamente amplio de posibilidades) adecuado según requerimientos de potencia, área y retardo de propagación. A modo de ejemplo se han dispuesto en la tabla 4.2 los valores del retardo y de la potencia para tres casos significativos.

Tomando como referencia la geometría 9/6 (correspondiente al diseño robusto), se calculan los siguientes porcentajes:

- para el dimensionado 4/3 (mínimo consumo) se obtiene un aumento del retardo de propagación del 30 % (aproximadamente), con una disminución en el consumo del 32 % (aproximadamente),
- para el dimensionado 6.5/5 (consumo intermedio), se obtiene un aumento

del retardo del orden del 4 %, con una disminución en el consumo del orden del 15 %.

Tabla 4.2. Potencia consumida y retardo de propagación (*fan-out* = 1).

Buffer	Potencia (μW)	Retardo (ps)
4/3	360.07	133.25
6.5/5	441.56	105.78
9/6	523.82	102.01

Se presenta en la figura 4.8 el consumo de potencia de la puerta inversora SDCFL en función del ancho de puerta del transistor de *pull-up* para un *fan-out* de 2 a la salida de la puerta lógica. Comparando con la figura anterior se observa como se ha reducido el número de dimensiones posibles.

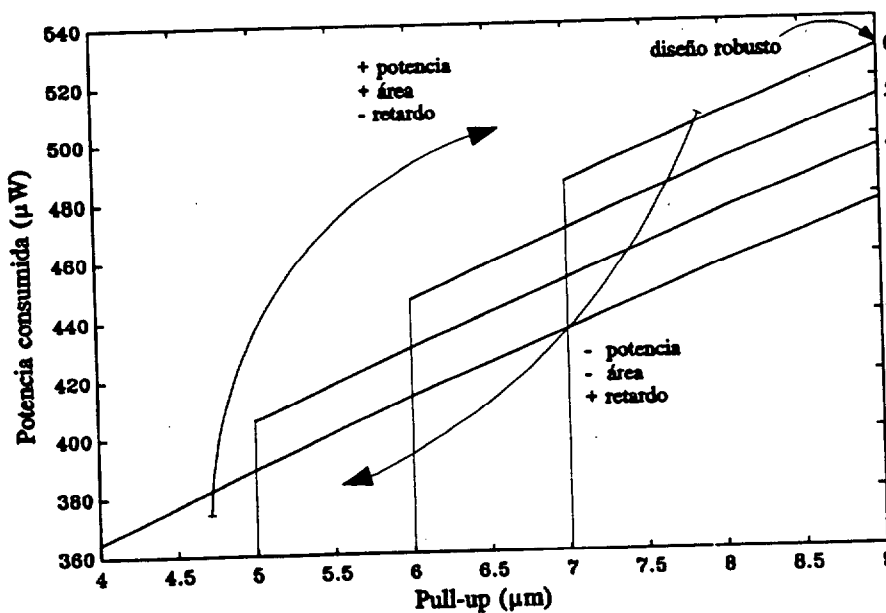


Fig. 4.8. Dependencia del consumo de potencia con la geometría del buffer, *fan-out* = 2.

Pueden consultarse en la tabla 4.3 las características retardo-potencia correspondientes a este caso. Tomando como referencia la geometría 9/6 (correspondiente al diseño robusto), se calculan los siguientes porcentajes:

- para el dimensionado 4/3 (mínimo consumo) se obtiene un aumento del

retardo de propagación del 57 % (aproximadamente), con una disminución en el consumo del 31 % (aproximadamente),

- para el dimensionado 6.5/5 (consumo intermedio), se obtiene un aumento del retardo del orden del 11 %, con una disminución en el consumo del orden del 15 %.

Se aprecia cómo se deterioran las prestaciones de la puerta relativas a la velocidad. Sin embargo, el consumo de potencia muestra el mismo grado de mejoría. El importante aumento de los retardos de propagación lleva a no aplicar "bufferización" adaptativa en aquellos nodos cargados con *fan-out* de cuatro (en estos casos se implantará siempre la geometría 9.2/6, dadas las excelentes características ofrecidas).

Tabla 4.3. Características potencia-retardo (*fan-out* = 2).

Buffer	Potencia (μW)	Retardo (ps)
4/3	364.86	227.15
6.5/5	446.34	161.35
9/6	530.14	144.50

Por último se presenta en la figura 4.9 el consumo de potencia de la puerta inversora SDCFL en función del ancho de puerta del transistor de *pull-up* para un *fan-out* de 3 a la salida de la puerta lógica. Tal como acontece para el caso anterior, el número de posibles geometrías vuelve a reducirse.

En este sentido no puede seleccionarse la geometría 6.5/5, que pierde su operatividad. A efectos de realizar la comparación se sustituye ésta por otro dimensionado (se ha elegido la geometría 7/4).

Las características potencia-retardo se expresan en la tabla 4.4. Tomando como referencia el seguidor de fuente de dimensiones 9/6 (correspondiente al diseño robusto), se calculan los siguientes porcentajes:

- para el dimensionado 4/3 (mínimo consumo) se obtiene un aumento del retardo de propagación del 60 % (aproximadamente) con una disminución en el consumo del 31 % (aproximadamente),
- para el dimensionado 6.5/5 (consumo intermedio) se obtiene un aumento del

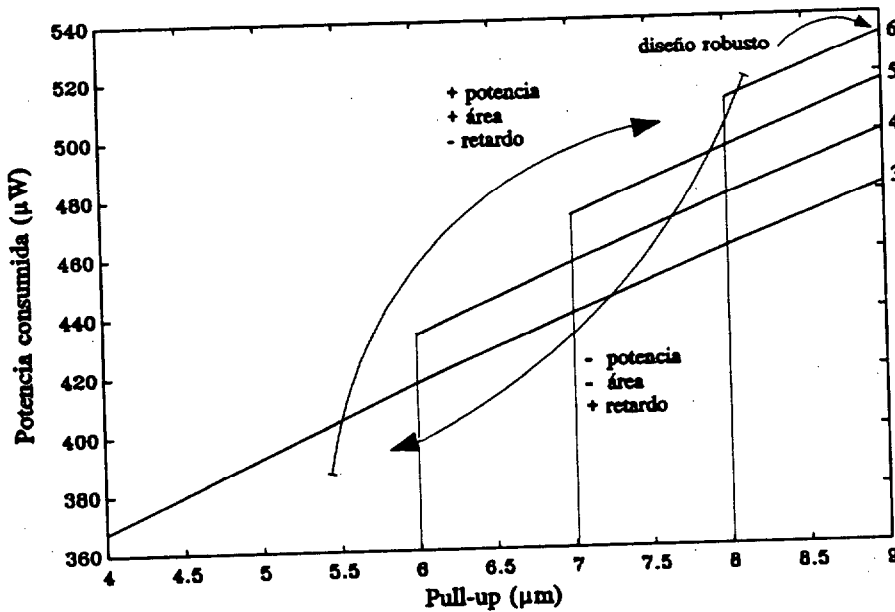


Fig. 4.9. Dependencia del consumo de potencia con la geometría del seguidor de fuente, fan-out = 3. retardo del orden del 30 % con una disminución en el consumo del orden del 14 %.

Tabla 4.4. Características potencia-retardo (fan-out = 3).

Buffer	Potencia (μW)	Retardo (ps)
4/3	367.50	308.18
7/4	456.70	250.55
9/6	533.60	193.65

Un deterioro importante de las prestaciones en velocidad se manifiesta en esta ocasión. La mejora lograda en cuanto al consumo de potencia sigue la línea mostrada en los casos precedentes. Se aclara que de entre estos valores, pueden trasladarse (de forma cuantitativa) al diseño completo el consumo de área y potencia. Es decir, que si el diseñador implementa un diseño con *buffers*, por ejemplo, 7/4 y 4/3, conoce con absoluta seguridad que el área ocupada y el consumo de potencia será menor que si el diseño emplea *buffers* 9.2/6.

El retardo de propagación en cambio, ha de tomarse con cautela no pudiendo trasladarse a cualquier parte del circuito directamente, dado que es función de la pendiente de entrada, del fan-in de las puertas, y además se encuentra sujeto a las posibles fluctuaciones mostradas por los niveles de tensión (especialmente V_{OH}).

Asimismo se apunta el hecho de que para todas las configuraciones implicadas en la

elaboración de estas gráficas (figuras 4.7 a 4.9), el nivel alto de tensión inferior fue de 500 milivoltios. El nivel de tensión correspondiente a V_{OL} se mantuvo en todos los casos inferior al valor de 0 voltios.

4.4.4.1.1.- Características potencia-retardo.

A modo de resumen se muestran en la tabla 4.5 los valores del producto potencia-retardo para las tres situaciones de cargabilidad ($fan-out = 1, 2, 3$). Las casillas que contienen líneas de puntos indican que el *buffer* en cuestión no es operativo bajo esas condiciones de funcionamiento.

Tabla 4.5. Valores del producto potencia-retardo.

Buffer	fan-out = 1 (mW·ps)	fan-out = 2 (mW·ps)	fan-out = 3 (mW·ps)
4/3	47.80	82.88	113.25
6.5/5	46.71	72.01	---
7/4	---	---	114.42
9/6	53.43	76.60	103.32

Bajo las restricciones anteriormente expuestas (cautela en la interpretación de los resultados), estas relaciones dan una muestra del alcance de la aplicación de estas técnicas de cara a mejorar el consumo de potencia (y consecuentemente la superficie ocupada).

En el siguiente apartado se pretende la aplicación de estas técnicas encauzadas a la mejora del circuito en cuanto al parámetro velocidad se refiere. Ello obligará a la búsqueda de dimensiones en la zona de consumo de potencia alto de la figura 4.4.

4.4.4.2.- Elección de la geometría guiada por el retardo de propagación.

En esta ocasión la estrategia consiste en elegir qué determinada geometría en aras de disminuir los retardos de propagación. Evidentemente, tanto el área ocupada como el consumo de potencia aumentarán sus respectivos valores.

El estudio a nivel de puerta (como se realizó en el anterior apartado) no es válido cuando se implica la estimación del retardo de propagación. La importante influencia del *fan-in*, así como de las fluctuaciones en los niveles de tensión (de menor orden es la influencia

debida a las pendientes de excitación), dificultan enormemente la tarea a llevar a cabo. Además, si no se impone un consumo de potencia (y de área) como límite, el conjunto de posibles geometrías se vuelve aún más intratable. Dado que es inviable analizar todos los diferentes casos a que conduce considerar:

- el *fan-in* a las puertas,
- el *fan-out* de las puertas y,
- las pendientes de excitación,

se procederá al estudio no a nivel de puerta, sino a nivel de circuito. La comparación de resultados no se hará por tanto con el formato de gráficas y tablas anteriores sino con la aplicación a un circuito de referencia.

En este trabajo de tesis hemos analizado exhaustivamente múltiples geometrías para el seguidor de fuente. Del conjunto de soluciones más interesantes obtenidas se pasa a presentar las más significativas, a utilizar en diseño adaptativo en nodos circuitales o caminos de propagación críticos en velocidad.

4.4.4.2.1.- Conjunto de *buffers* aceleradores.

Denominamos con este nombre (*aceleradores*) aquellos tipos de *buffers* cuyo empleo en un determinado nodo de salida de una puerta conducen a una mejora sustancial en la velocidad del mismo. Ha de entenderse como sustancial a la mejora de al menos un 10 % en el retardo de propagación. Por contrapartida, el consumo de potencia y de área puede aumentar un porcentaje significativamente mayor. Aún así, se pretende que a través de estas técnicas de "buferrización", combinadas con el dimensionado óptimo de los transistores (ver capítulo 5), se logren importantes mejoras. Por consiguiente, dentro de este marco de diseño, una pequeña modificación puede adquirir una importante dimensión.

Se ha realizado un análisis de una gran variedad de estructuras lógicas típicamente encontradas en los circuitos digitales (con diverso *fan-in*, *fan-out* y excitadas con las rampas usualmente existentes).

Las geometrías aceleradoras óptimas obtenidas son las siguientes:

- dimensionado 12/7 (*fan-out* = 1,2,3,4),
- dimensionado 12/8 (*fan-out* = 1,2,3),

velocidad conseguida merced al empleo de los *buffers* aceleradores mejora según aumenta el valor de estas capacidades (capacidades debidas al interconexión, y por tanto siempre presentes). Pueden consultarse en la tabla 4.6 los resultados obtenidos según los dimensionados.

El primer ejemplo (a título comparativo) se ha implementado según metodología de diseño robusto. En el segundo diseño hemos empleado una estrategia robusta pero basada en el *buffer* acelerador 17/7.

En el tercer diseño hemos empleado una estrategia adaptativa haciendo uso de los *buffers* aceleradores. En el cuarto diseño (que también contiene *buffers* aceleradores) se ha reemplazado uno de los *buffers* aceleradores por uno de bajo consumo (8/4) en los nodos que tienen *fan-out* igual a uno.

Tabla 4.6. Comparación de resultados para las diversas implementaciones.

Buffer	Retardo (ps)	Potencia (mW)	Área (μm^2)
9.2/6	925.88	3.286	10320.0
17/7	801.18	4.235	11193.6
Adaptativo 1	868.48	3.885	10775.6
Adaptativo 2	888.55	3.586	10313.6

Con respecto al diseño robusto basado en el *buffer* 9.2/6 (ver tabla 4.7) se consigue una mejora sustancial en velocidad y con respecto al tercer diseño, se consigue una disminución en el consumo de potencia. Los porcentajes que en dicha tabla aparecen están referenciados al seguidor de fuente de geometría 9.2/6.

Tabla 4.7. Comparación de resultados (referenciados al *buffer* 9.2/6).

Diferencia relativa	Adaptativo 1	Adaptativo 2	Robusto (17/7)
Retardo	- 6.20%	- 4.04%	- 13.46%
Potencia	+ 18.2%	+ 9.2%	+ 28.7%
Área	+ 4.41%	- 0.06%	+ 8.46%

De especial interés se manifiesta el incremento en el consumo de potencia, mientras que el aumento de la superficie no resulta dramático. El decremento en el retardo de propagación no es espectacular, pero resulta altamente significativo si se considera que el diseño robusto constituye de por sí, un diseño prácticamente optimizado. Se presentan en el capítulo 6 más ejemplos (combinados con la estrategia de dimensionado de transistores).

4.5.- ESTRATEGIA OPCIONAL PARA ACELERAR LA LÓGICA.

Aunque esta tesis se desarrolla en torno a la topología para el seguidor de fuente se pretende mostrar el juego de posibilidades que la lógica DCFL/SDCFL ofrece cuando se actúa sobre otros parámetros eléctricos. En particular analizamos a continuación el margen de optimización que ofrece la polarización del circuito.

Conviene observar que aunque los valores de las tensiones en las familias lógicas en GaAs no se pueden considerar fijos o estándar, no deben alterarse en los diseños, por el confucionismo o incompatibilidad en su uso que se crea. De hecho tal optimización no corresponde a una fase de optimización del diseño, sino de puesta a punto de las familias lógicas soportadas por cada proceso.

La razón por la que se eligieron unos valores tan pequeños para las fuentes de alimentación (+1.4 y -0.2 voltios) está relacionada con el hecho de que se persigue la integración a muy alta escala (VLSI).

Dado que el consumo de potencia aumenta de forma dramática con el valor creciente de dichas tensiones, constituye razón prioritaria la selección de tensiones de alimentación de voltaje reducido. Además estas tensiones provienen de un exhaustivo análisis de la familia lógica DCFL, que confirma su empleo (ver capítulo 2).

Posibles parejas de voltajes alrededor de este par son factibles, aunque por lo general dan lugar a transiciones fuertemente asimétricas y no significativamente más rápidas.

Ya se ha demostrado en este capítulo lo difícil que resulta acelerar un diseño dado mediante el dimensionado del seguidor de fuente. Similar dificultad ofrece el pretender hacerlo mediante el dimensionado de los transistores que integran la etapa lógica (ver capítulo 6). Situación análoga también acontece cuando se pretende acelerar la lógica mediante la alteración de los voltajes de alimentación.

La alteración del voltaje de las fuentes de alimentación redundará de forma positiva en la mejora de los retardos de propagación (con resultados como se verá no espectaculares). Lo hará de forma negativa en el consumo de potencia (vuelve a notarse una vez más el molesto incremento del consumo).

Para nada afectará, en principio, al consumo de área (incluso pudiera llegar a disminuir, debido a que el dimensionado de transistores pudiera ser inferior).

Puesto que no se dispone del conjunto de coeficientes requeridos por nuestros modelos para estimar tanto la potencia como el retardo para estas polarizaciones no estándar en DCFL/SDCFL, se obtendrán estos valores vía simulación HSPICE.

De las múltiples posibilidades analizadas, se presenta aquella que se alimenta mediante una tensión extrema V_{DD} de + 1.8 voltios y con una fuente de - 0.3 voltios para V_{SS} . Otra variación intermedia estudiada es aquella que emplea una tensión de 1.7 voltios para V_{DD} y - 0.25 voltios para la otra fuente. Estos valores límites para las fuentes de alimentación surgen debido a que otros valores dan lugar a que se violen los márgenes de ruido y/o a la degradación de las características de conmutación de la lógica.

Se ha simulado el circuito contenido en la figura 4.10 para ambas parejas de tensiones. Se puede ver en la tabla 4.8 la comparación de los resultados obtenidos. Del estudio comparativo se desprende que con los nuevos pares de tensiones de alimentación se mejora el retardo de propagación, pero en una proporción menor que si se aplican las técnicas de "bufeización" propuestas. Además, el consumo de potencia aumenta, esta vez, de tal forma que hacen inviable la aplicación de dichos voltajes.

Tabla 4.8. Comparación de resultados para las distintas parejas de tensiones de alimentación.

Tensiones de alimentación	Retardo (ps)	Consumo (mW)
$V_{DD} = 1.8V, V_{SS} = 0.30V$	877.93	5.318
Diferencia relativa	- 5.17%	+ 61.83%
$V_{DD} = 1.7V, V_{SS} = 0.25V$	869.017	4.719
Diferencia relativa	- 6.14%	+ 43.60%

Se concluye este capítulo sin más que resaltando el potencial de optimización que ofrece la técnica de "bufferización" que para diseño adaptativo ha sido desarrollada. A su vez, queda afianzado el diseño según la metodología robusta (con la justificación completa del tipo de dimensionado óptimo 9.2/6 para el seguidor de fuente), para aquellas aplicaciones en las que acortar el tiempo de diseño sea prioritario.

En muchas aplicaciones bastará hacer una optimización con *buffer* fijo excepto en los caminos críticos y nodos muy cargados por *fan-out* e interconexiones para los que se debe hacer una optimización con *buffer* adaptativo. Los márgenes de optimización alcanzables han sido también evaluados.

CAPÍTULO 5

ALGORITMOS Y ESTRATEGIA GLOBAL DE OPTIMIZACIÓN.

Se presentan a continuación el algoritmo para realizar el dimensionado de las puertas lógicas y la estrategia global de optimización desarrollados en base a los modelos expuestos en el capítulo 3, para la mejora de prestaciones de los circuitos combinatoriales DCFL/SDCFL. La estrategia de optimización global se llevará a cabo en dos fases:

- fase de dimensionado de los *buffers* (*buffering*) aplicando los métodos del capítulo anterior y,
- fase de dimensionado de los transistores (*sizing*), aplicando métodos que se desarrollan en éste.

La metodología para dimensionar los transistores es completamente analítica y suministra el conjunto de dimensiones óptimas de los transistores presentes en la ruta crítica del circuito. La aplicación de dicha metodología es lo suficientemente rápida como para poder utilizarse de forma práctica. La comparación de los resultados frente a los que se obtienen a partir de simulación numérica (HSPICE) resulta excelente.

5.1.- INTRODUCCIÓN.

El diseño en GaAs -dadas las propiedades del material y su costo- es un diseño para prestaciones. De no ser así, no se entiende su aplicación. A diferencia del diseño tradicional en tecnología de Silicio, resulta imprescindible la estimación inicial de ciertas funciones de coste (a definir), de cara a tener una visión preliminar del sistema a implementar.

En el ámbito GaAs, el número de factores que afectan al diseño es mucho mayor y el estudio de viabilidad muestra mayor complejidad. Si bien no es objetivo de esta tesis el plantear análisis de viabilidad o decidir desde niveles de jerarquía elevados alternativas de diseño a seguir [Sarmi91], conviene tener en cuenta que estos temas no se pueden abordar si no se dispone de una metodología capaz de acometer el problema desde la raíz, es decir, a nivel de transistores.

Tal como se ha comentado en el capítulo 1 (Introducción), toda herramienta de síntesis precisa del conocimiento de funciones estimativas que partan desde los niveles de transistor o, desde una jerarquía inmediatamente superior como puede ser la puerta lógica.

Se han presentado en el capítulo 3 los modelos estimativos del retraso a lo largo de un camino (3.1), del consumo de potencia (3.2) y del área ocupada por el circuito (3.3). Dichos modelos, desarrollados expresamente para abordar el problema de la optimización, han sido extensamente evaluados, siendo conocido el error que se comete en su aplicación.

Tal como se ha detallado en el capítulo 2 (Estudio de la familia lógica DCFL/SDCFL), de especial relevancia es la consideración del ruido (tanto dinámico como estático) durante el diseño de circuitos GaAs. Es ésta una circunstancia que, a diferencia de lo que acontece en circuitos CMOS, se ha tenido en cuenta ya desde los primeros diseños GaAs. Para circuitos CMOS, es ahora cuando comienzan los primeros estudios profundos tratando el aspecto del ruido [CheKa91]. Por lo tanto, ha de ser incluida como variable más de cara a encontrar soluciones óptimas en el circuito.

A diferencia de lo que sucede para el caso de estimación de retrasos, área y potencia, no se dispone de modelo analítico para la estimación del ruido. Dada la dificultad de desarrollar un modelo analítico capaz de tratar con el importante problema de la estimación del ruido, la estrategia a seguir consiste en seleccionar el conjunto de puertas (o de soluciones posibles) de tal forma que se respete el margen de ruido. Por consiguiente, las soluciones posibles entrarán a formar parte de las restricciones al algoritmo (ver sección 5.5.3), y se sabe que las soluciones respetan, en todo momento, los márgenes de ruido (ver

apartado 2.2.2).

En este capítulo, se procede a la presentación de un algoritmo y de una estrategia de optimización capaz de operar con los modelos anteriormente citados. Este algoritmo pone de manifiesto la sensibilidad de las prestaciones del circuito (entiéndase como prestaciones las soluciones en el espacio de diseño área-retardo-potencia), según sean las dimensiones de los transistores y según sea el tipo de *buffer* presente a la salida de las puertas lógicas.

Se resalta el hecho de que las soluciones aportadas por el algoritmo (así como por la metodología en general), son de carácter cuantitativo, dependiendo su grado de precisión de los ficheros tecnológicos y de coeficientes requeridos. En la medida en que estos ficheros estén validados, los resultados pueden integrarse en otras herramientas de síntesis, diseño o verificación.

Como soluciones, dicha estrategia de optimización suministra dos tipos de resultados:

- por una parte, aporta al diseñador unas soluciones de compromiso en la elección del dimensionado óptimo de cada una de las puertas a constituir el circuito,
- por otra parte, proporciona criterios para la elección del *buffer* adecuado, de entre el conjunto finito definido en el capítulo anterior, según compromisos. Ésta, a diferencia de la anterior, no se encuentra automatizada, salvo que sólo se utilice la técnica de diseño robusto en todo el circuito.

Cada uno de estos aspectos serán convenientemente detallados en las secciones siguientes.

5.2.- PROPIEDADES ANALÍTICAS DE LOS MODELOS A EMPLEAR.

Antes de tomar cualquier decisión acerca de cual elegir de entre las diferentes técnicas de optimización de circuitos propuestas, conviene llevar a cabo un estudio de las propiedades analíticas que poseen los modelos presentados en el capítulo 3.

Un estudio en cuanto a precisión y propiedades de las funciones de estimación se presenta a continuación.

5.2.1.- Precisión en la estimación.

Ante todo, hay que indicar el grado de precisión mostrado por el modelo temporal, el modelo de potencia y el de área.

El error observado se sitúa, para los tres modelos, por debajo del 10% para todas las estructuras evaluadas. De entre los tres modelos, destaca el modelo estimativo de retrasos con errores, en la mayoría de las ocasiones inferiores al 5% (hecho éste de especial relevancia dada la complejidad que presenta el problema).

Los modelos para la estimación del área ocupada y para la estimación del consumo de potencia, muestran a su vez, una gran precisión en la evaluación de estructuras regulares (las cuales pueden constituir un alto porcentaje del circuito).

5.2.2.- Funciones analíticas.

De cara a implementar una herramienta para optimización analítica de circuitos, se requiere disponer, a su vez, de modelos analíticos para las funciones estimativas.

Orientados por este propósito se han construido los correspondientes modelos de área, potencia y estimación de retardos.

Si bien desde un principio tanto el modelo del retraso como de área eran analíticos, la situación era algo diferente para el modelo de potencia. Era éste, fundamentalmente, un modelo basado en tablas de datos (*look-up tables*). Se pudo sortear esta dificultad (dificultad que impedía la implementación de una estrategia de optimización completamente analítica), mediante el empleo de funciones polinómicas de ajuste. Merced a estas expresiones analíticas, eran ajustados los datos contenidos en las tablas. Se recuerda que el ajuste se conseguía empleando polinomios de hasta 5º grado.

De esta forma, se dispone de tres funciones estimativas analíticas, las cuales contienen como variable, a la variable de diseño por excelencia, es decir, el ancho de puerta del transistor.

Por el hecho de venir dadas las tres funciones estimativas como expresiones polinómicas (para el caso del área, en realidad es una función lineal, o sea polinomio de grado uno) gozan de las siguientes propiedades:

- son funciones continuas (en la variable w , ancho de puerta del transistor),
- poseen derivadas y éstas son funciones continuas (en la variable w) y,
- se computan de forma muy rápida.

Por consiguiente, cabe esperar buenos resultados y se prevé que baste un sencillo algoritmo numérico para la resolución del problema.

Resalta el hecho de que las funciones estimativas sean simples expresiones polinómicas, a diferencia de las expresiones posinómicas presentes en los algoritmos de optimización para circuitos CMOS (consultar referencias [FisDu85], [Hedlu87], [HoNeS90], [NavRo91]).

5.2.3.- Limitaciones de los modelos (restricciones).

Además del grado de precisión de los modelos y, de las propiedades de las funciones estimativas, es necesario explicitar las limitaciones y restricciones empleadas.

Respetando el orden de presentación, se inicia este pequeño sumario por el modelo temporal. Se indicarán restricciones y, asimismo, cómo abordarlas de cara a una futura ampliación que quedará para trabajos posteriores a esta tesis.

5.2.3.1.- Modelo temporal: limitaciones.

Dada la complejidad que muestra la puerta lógica base de la familia SDCFL, no es posible la resolución analítica de la respuesta temporal. Esta dificultad viene dada por dos razones:

- las expresiones que modelan al comportamiento del transistor MESFET incluyen funciones trascendentes,
- la topología de la puerta SDCFL incluye dos fuentes de alimentación (V_{DD} y V_{SS}) y se compone de cuatro transistores, a diferencia de la puerta inversora CMOS, que emplea una sola fuente de alimentación y se compone de sólo dos transistores.

Además, la respuesta transitoria analítica para el caso de la puerta inversora DCFL [Herná92] se muestra de una enorme complejidad, de cara ya no sólo a incluirla en un algoritmo de optimización, sino simplemente a su evaluación numérica.

Los modelos analíticos simplificados arrojan errores en la estimación del retraso de un orden superior al 100% para el caso de una simple puerta inversora [LonBu90]. Queda patente su inutilidad de cara a implementar una estrategia de optimización sustentada sobre modelos de esta índole. Es esta la principal razón que ha llevado a la búsqueda de un modelo preciso y rápido de evaluar. El modelo es tecnológicamente independiente y, puesto que se calculan los coeficientes vía simulación HSPICE, incluye fenómenos de canal corto y demás efectos incorporados en los modelos de HSPICE.

Tal como se comentará posteriormente (sección 5.4.1.1), se toma el modelo completo para el retraso con vistas a determinar el camino crítico, que es la ruta que se pretende optimizar. Durante la búsqueda de dicho camino, se consideran tanto efectos de pendiente como de colisiones. Esto implica la máxima precisión en la determinación del camino crítico alcanzable con el modelo temporal.

Como contrapartida (de cara a optimizar) se limita el modelo en base a las siguientes restricciones:

- se permite sólo una variable por puerta. Dicha variable es el ancho de puerta del transistor de *pull-down* de la etapa inversora. Esta es una limitación que viene dada por el modo cómo se ha construido el modelo. Futuras implementaciones pueden salvar esta restricción sin más que definir otro polinomio conteniendo una dimensión más de otro transistor como variable adicional,
- los efectos asociados a colisiones no son tenidos en cuenta (durante la optimización, se opera sobre el camino crítico reducido a inversores equivalentes). Por consiguiente, la estimación del retraso se hará según el criterio del peor caso.

Aún así, el error cometido por puerta en la estimación del retraso sigue siendo inferior al 10% en todos los casos testados.

Se resalta el hecho de que, aunque los efectos asociados a las colisiones de señales no son tenidos en cuenta, sí lo son los efectos asociados a la forma de onda (pendiente).

Operando de esta forma se consigue evaluar el retardo de propagación de la cadena de inversores equivalentes con la máxima precisión que es capaz de dar el modelo temporal.

Las razones por las que se toman estas restricciones vienen dadas por la búsqueda de simplicidad y ahorro computacional. Su inclusión en la metodología a desarrollar no precisa complejidad adicional.

5.2.3.2.- Modelo de estimación de potencia: limitaciones.

Es éste también un modelo tecnológicamente independiente que obtiene los valores de los coeficientes de los polinomios vía simulación HSPICE. Por consiguiente, las mejoras en los modelos de HSPICE, significan mejoras en el modelo estimativo de potencia.

El comentario que se cita en la primera de las restricciones para el modelo del retraso, se aplica de igual forma para el caso de la potencia. Esto es consecuencia, una vez más, de la dificultad de obtener un modelo analítico para la estimación del consumo de potencia.

Dada la naturaleza estática del modelo, las restricciones tomadas en el caso anterior, no tienen ningún efecto sobre la estimación de la potencia.

5.2.3.3.- Modelo de estimación del área: limitaciones.

Al igual que sucede en los circuitos CMOS, el modelo de área para el caso GaAs muestra similar simplicidad y las mismas restricciones.

El modelo es fuertemente dependiente del estilo de diseño con el que se realiza el circuito. En esta línea y, con el afán de introducir (si ese fuera el caso) nuevos estilos, se exige simplemente regularidad en el trazado del circuito. Es ésta una exigencia tan razonable que, dado las tendencias actuales de diseño, podría considerarse trivial.

Al igual que para los modelos anteriores, nuevas ampliaciones encuentran cabida en el sencillo modelo de estimación del área. Así la incorporación de nuevos estilos de diseño pasa por la obtención de los parámetros necesarios (que son sólo dos, los llamados ζ y φ). Con el nuevo conjunto de parámetros, se aplica la metodología sin modificación alguna.

A diferencia de los modelos de potencia y retraso, el modelo estimación de área opera a nivel de transistor, admitiéndose tantas variables como transistores existan en el circuito. Por motivos de compatibilidad con los anteriores modelos, se restringe el conjunto de variables a la misma que se ha definido para los casos anteriores (la anchura del transistor, w).

5.3.- Clasificación del problema.

La optimización del circuito se desarrollará en dos fases claramente diferenciadas. La primera de ellas, versará sobre la optimización local o técnicas de "bufeización". La segunda de ellas, analítica, trata el problema del dimensionado de los transistores.

A lo largo de los capítulos que conforman esta tesis, se han hecho con frecuencia comparaciones entre los circuitos GaAs y los circuitos CMOS. Éstos se han tomado como punto de referencia en todo momento dado que, al ser la tecnología *VLSI* de mayor difusión, es también en la que más se ha trabajado y de la que se conocen más referencias.

En estas comparaciones tanto las similitudes (pocas ellas) como sobre todo las diferencias entre distintos aspectos, han sido profusamente remarcadas. A continuación se presenta una somera clasificación de las técnicas más comunes para la optimización de circuitos CMOS. El objetivo de esta presentación radica en poner de manifiesto la complejidad del problema bajo estudio.

5.3.1.- Técnicas de optimización de circuitos.

Ante todo, se presenta un conjunto de definiciones y una clasificación de las distintas técnicas de optimización.

5.3.1.1.- Tipos de problemas de optimización.

De acuerdo con la forma en que esté planteado el problema, existen dos grandes categorías de técnicas de optimización:

- analíticas,
- iterativas.

Por otra parte, las propiedades relativas a las condiciones restrictivas y a la linealidad introducen una segunda clasificación dentro de estas dos categorías.

Se entiende como método de resolución analítico aquel que dada una función, lineal o no lineal, con condiciones restrictivas o no, admite solución analítica. El procedimiento para llegar a esta solución es a través del cálculo de las derivadas de la función a optimizar. El principio de las técnicas analíticas se reduce al de los máximos y mínimos ordinarios de una función. Para el caso de condiciones restrictivas de igualdad, es de uso frecuente la resolución mediante los llamados multiplicadores de Lagrange.

Por el contrario, las técnicas iterativas conducen a soluciones numéricas. Estas técnicas son las que más se emplean en la optimización de circuitos, dado que las funciones a optimizar son típicamente de marcado carácter no lineal y sujetas a condiciones restrictivas de desigualdad.

Tiene especial relevancia el hecho de que la función a optimizar sea cóncava o convexa.

Se define una función de una sola variable $f(x)$ como convexa, si para cada par de valores de x , por ejemplo, x' y x'' ,

$$f[\lambda x'' + (1-\lambda)x'] \leq \lambda f(x'') + (1-\lambda)f(x'), \quad \forall \lambda: 0 \leq \lambda \leq 1 \quad \{5.1\}$$

Se dice que $f(x)$ es función estrictamente convexa si " \leq " puede sustituirse por " $<$ ".

Se dirá que $f(x)$ es función cóncava si la expresión {5.1} se cumple cuando se reemplaza " \leq " por " \geq ". De similar forma se define la función estrictamente cóncava.

Las funciones de múltiples variables (este es el caso en cuestión), admiten también la clasificación de cóncavas o convexas. Así, dada una función multivariable expresada como suma de términos, si todos los términos son cóncavos (se puede verificar que lo sea con su segunda derivada cuando el término incluye nada más que una variable), entonces la función es cóncava. De manera similar, la función es convexa si todos los términos son convexas.

Si un problema de programación no lineal no tiene restricciones, el hecho de que la función objetivo sea cóncava garantiza que un máximo local es un máximo global. De igual

forma, una función objetivo convexa asegura que un mínimo local es un mínimo global.

Si existen restricciones (caso que nos concierne), se necesita una condición más para dar esta garantía. Esta es que la región factible (conjunto de soluciones posibles) sea un conjunto convexo. Se define como conjunto convexo a un conjunto de puntos tales que, para cada punto de la colección, el segmento de recta que los une está totalmente contenido en la colección. Esta definición se extrapola para el caso de funciones multivariantes, categoría a la que pertenece nuestro problema de optimización.

5.3.1.2.- Clasificación de los problemas no lineales y técnicas de resolución.

Los problemas de optimización no lineal (conocidos también con el nombre de problemas de programación no lineal), se presentan de muchas formas distintas. Al contrario de lo que sucede para los problemas de programación lineal, no se dispone de un algoritmo que resuelva todos estos tipos especiales de problemas.

Los problemas lineales no muestran, por lo general, ninguna dificultad en su resolución, existiendo diversos métodos (ya programados en versión comercial) para su resolución. De entre ellos, el más conocido es el llamado método del simplex, desarrollado en el año 1947 [GasSa75].

Los problemas no lineales se manifiestan, por lo general de extrema dificultad a la hora de su resolución, debido principalmente a lo poco desarrollado que se encuentran las técnicas matemáticas relativas al tema.

Así, se han desarrollado algoritmos para algunas clases (tipos especiales) de problemas de programación no lineal. Se comentarán los más importantes.

Destacan los siguientes tipos de problema:

- programación cuadrática,
- programación geométrica.

El problema cuadrático es aquel que tiene restricciones lineales pero la función objetivo incluye el cuadrado de una variable o el producto de dos variables. Se han desarrollado multitud de algoritmos para este caso, con la suposición de que la función a optimizar es cóncava.

La programación cuadrática es muy importante porque las formulaciones de este tipo surgen de manera natural en muchas aplicaciones.

Cuando se aplican las técnicas de programación no lineal a problemas de ingeniería, en especial circuitos electrónicos, muchas veces la función objetivo y las funciones de restricción toman la forma {5.2}:

$$g(x) = \sum_{i=1}^N c_i \cdot P_i(x), \quad \text{donde} \quad \{5.2\}$$

$$P_i(x) = x_1^{a_{i1}} \cdot \dots \cdot x_n^{a_{in}} \quad \text{para } i=1,2,\dots,N$$

Los problemas que contienen una función objetivo de este tipo pertenecen a la segunda categoría. En tal caso, las c_i y las a_{ij} representan las variables físicas, y las x_i son las correspondientes de diseño.

Estas funciones por lo general no son ni cóncavas ni convexas, con lo que no se pueden aplicar las técnicas de programación no convexa. Sin embargo, existe un caso importante en el que el problema se puede transformar en un problema de programación convexa equivalente. Este caso es aquel en el que todos los coeficientes c_i en cada función son estrictamente positivos, es decir, las funciones son polinomios positivos generalizados (también conocidos como posinomios).

Dado que las funciones posinómicas son convexas, se garantiza la existencia de un solo mínimo que resulta ser el mínimo global.

La mayoría de algoritmos para optimización de circuitos CMOS ([HoNeS90], [NavRo91]) operan sobre funciones posinómicas (en realidad se formula el problema inicial mediante posinomios realizando adecuadamente un cambio de variable), sabida la enorme ventaja que supone conocer la convexidad de dichas funciones. Además, existe una extensa literatura abordando dicho tema, véase por ejemplo [WilBe67].

5.4.- ESTUDIO DE LA FUNCIÓN OBJETIVO A OPTIMIZAR.

El motivo de presentación de esta clasificación viene justificado por la conveniencia de enmarcar el problema bajo estudio dentro del conjunto de procedimientos que se encuentran

en una fase avanzada de desarrollo. Siguiendo esta línea, han sido muchos los intentos para transformar el problema a uno de tipo posinómico o cuadrático. Aparte del estudio en sí, ha sido necesario desarrollar programas informáticos para llegar a conclusiones que reafirmen o descarten un método determinado (siendo éste un proceso altamente laborioso).

La naturaleza del problema que se aborda al tratar con circuitos DCFL/SDCFL presenta una complejidad superior a la ya clásica aproximación para circuitos de CMOS. Tras fallidos intentos de reformular el problema para encajarlo en una de las categorías anteriormente expuestas, se ha optado por simplificar los modelos (sólo ha sido necesario imponer restricciones en el modelo estimativo de retrasos). Estas restricciones han sido ya comentadas en el apartado 5.2.3, y permiten la resolución del problema mediante el método generalizado de gradiente reducido [LasWa78], también conocido como GRG (*Generalized Reduced Gradient*). Seguidamente se procede al estudio de la función objetivo a manejar.

5.4.1.- Definición de la función objetivo.

La función objetivo a optimizar es aquella que relaciona las expresiones correspondientes a:

- D , estimación del retraso o tiempo de propagación,
- P , estimación de la potencia y,
- A , estimación del área.

Se procede a expandir cada una de las funciones estimativas para abordar el problema de la optimización de una red lógica que implica n puertas lógicas (con las correspondientes cargas, es decir, *fan-out* y capacidades asociadas al cableado).

5.4.1.1.- Estudio de la función de estimación del retraso. Aplicación a una cadena de n puertas lógicas.

Se expresa el retraso de una puerta lógica reducida a inversor equivalente según la expresión {5.3}. A su vez, se calcula la pendiente a propagar mediante un polinomio similar, con los coeficientes de pendiente pertinentes.

En dicha expresión D_{HL} se refiere al tiempo de propagación de la puerta lógica cuando su salida pasa de un estado lógico alto al estado lógico bajo. Similar expresión se deduce para la otra transición sin más que elegir convenientemente el juego de coeficientes $\{d, g, h, j, k\}$.

$$\begin{aligned}
D_{HL}(\beta, C_L^{HL}, \tau_{LH}) &= (d_0^{HL} + d_1^{HL} \cdot \beta + d_2^{HL} \cdot \beta^2) + (g_0^{HL} + g_1^{HL} \cdot \beta + g_2^{HL} \cdot \beta^2) \cdot C_L^{HL} + \\
&+ \{h_0^{HL} + h_1^{HL} \cdot \beta + h_2^{HL} \cdot \beta^2 + (j_0^{HL} + j_1^{HL} \cdot \beta + j_2^{HL} \cdot \beta^2) \cdot C_L^{HL} + (k_0^{HL} + k_1^{HL} \cdot \beta + k_2^{HL} \cdot \beta^2) \cdot C_L^{HL} \cdot C_L^{HL}\} \cdot \tau_{LH} = \\
&= \sum_{i=0}^2 d_i^{HL} \beta^i + \sum_{i=0}^2 g_i^{HL} \beta^i C_L^{HL} + \sum_{i=0}^2 \{h_i^{HL} \beta^i + j_i^{HL} \beta^i C_L^{HL} + k_i^{HL} \beta^i C_L^{HL} C_L^{HL}\} \cdot \tau_{LH} = \\
&= \sum_{i=0}^2 \{d_i^{HL} \beta^i + g_i^{HL} \beta^i C_L^{HL} + [h_i^{HL} \beta^i + j_i^{HL} \beta^i C_L^{HL} + k_i^{HL} \beta^i C_L^{HL} C_L^{HL}] \cdot \tau_{LH}\}
\end{aligned} \quad \{5.3\}$$

$$\beta = \frac{W_{pull-down}}{W_{pull-up}}$$

Dado que la transición es del estado alto al estado bajo, la pendiente de excitación ha de ser τ_{LH} (para la otra transición se tomará el valor τ_{HL}).

En esta expresión, el supraíndice que acompaña a la variable β , es el operador que indica potencia en el sentido matemático. Mientras no se afirme lo contrario, los supraíndices se entenderán como tales, excepto los que acompañen a dicha variable.

El valor de la capacidad C_L^{HL} se obtiene a partir de {5.4}:

$$C_L^{HL} = C_{\pi} + C_{out}^{HL} + \xi^{HL} \cdot \beta_{fan-out} \quad \{5.4\}$$

donde:

- C_{π} = capacidad asociada al cableado,
- C_{out}^{HL} = capacidad de salida de la puerta,
- ξ^{HL} = capacidad equivalente del inversor unidad,
- $\beta_{fan-out}$ = beta equivalente asociada al *fan-out*.

Los valores numéricos asociados a cada uno de estos términos pueden consultarse en [Herná92]. Expresiones similares se deducen para la otra transición.

En base a estas definiciones, se expresa el retraso a lo largo de una cadena de n puertas como sigue:

$$D = D(\beta_1, \beta_{fan-out_1}, \beta_2, \beta_{fan-out_2}, \dots, \beta_{n-1}, \beta_{fan-out_{n-1}}, \beta_n, \beta_{fan-out_n}) \quad \{5.5\}$$

Según las restricciones aceptadas, se toma como variable a dimensionar el ancho de puerta del transistor de *pull-down* de la etapa inversora (quedando las dimensiones del resto de transistores fijas). En estos términos el factor β , relación de aspecto, se expresa para cada una de las n puertas a considerar según {5.6}.

$$\begin{aligned} \beta_{fan-out_1} &= \beta_1(w_1) \\ \beta_{an-out_1} &= \beta_2(w_2) \\ &\vdots \\ &\vdots \\ \beta_{fan-out_{n-1}} &= \beta_{n-1}(w_{n-1}) \\ \beta_{fan-out_n} &= \beta_n(w_n) \end{aligned} \quad \{5.6\}$$

El hecho de que $\beta_{fan-out}$ sea sólo función de la dimensión del siguiente transistor de *pull-down* es consecuencia de que se optimiza el camino crítico, permaneciendo fijas las dimensiones de las puertas conectadas a los nudos. La influencia de estas puertas se encuentra contenida en el término C_L como elemento capacitivo. Es precisamente éste término el que acopla el retardo de cada puerta con el de la siguiente puerta lógica.

A partir de este momento, se expresará la función estimativa del retraso como función de β , entendiéndose ésta en los términos anteriormente mencionados.

El efecto capacitivo asociado al cableado se encuentra contenido en el término C_r . Este valor se considera fijo, lo cual supone que al variar el dimensionado de los transistores, para nada se alteran las dimensiones de la pista de conexión. Dado que se trata de una metodología a nivel de *pre-layout*, los valores para la capacidad asociada al cableado, son meramente estimativos. Para tener en cuenta variaciones en este valor capacitivo, sería preciso definir {5.7},

$$C_r = C_r(w_i) \quad 1 \leq i \leq n \quad \{5.7\}$$

En esta expresión el subíndice i referencia a la puerta cuyo transistor de *pull-down* posee la dimensión de puerta de w micras.

Esta relación podría deducirse a partir de las reglas de diseño. En la metodología que se presenta no se considera a C_L como variable. Disponiendo de la adecuada expresión que, presumiblemente es lineal en la variable w , su inclusión no supondría cambios significativos.

Se observa que el polinomio para la estimación del retraso es de grado 2 en la variable w . A consecuencia de las expresiones {5.4} y {5.6}, aparecen términos del tipo w_i , w_{i+1} , y términos tipo $w_i^2 \cdot w_{i+1}$. La presencia de términos como este último presentado, impiden la reformulación del problema a uno de tipo cuadrático.

Con la concreción de dependencias funcionales expuestas, el retraso a lo largo de una cadena se evalúa merced a una expresión como la siguiente:

$$D = D(\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) \quad \{5.8\}$$

Esta expresión evalúa el retraso a lo largo de una cadena de n puertas reducidas a inversores equivalentes. La técnica de reducción a inversores equivalentes fue expuesta en la sección 3.1.

En esta expresión se computa el retraso sin más que sumar los retrasos de propagación de cada puerta según sea la transición de alto a bajo (a la salida de la puerta) o de bajo a alto (a la salida de la puerta).

Dado que no se suponen inversores simétricos (aproximación típicamente admitida para circuitos CMOS), es preciso modificar {5.8} y expresarla como:

$$D(\beta) = \vartheta_{HL} \cdot D(\beta)_{HL} + \vartheta_{LH} \cdot D(\beta)_{LH} \quad \{5.9\}$$

$$\vartheta_{HL} + \vartheta_{LH} = 1$$

donde los coeficientes ϑ_{HL} y ϑ_{LH} son pesos -adimensionales- definidos para dar mayor relevancia a una transición que a otra.

El hecho de definir estos coeficientes de peso se debe a que, para los circuitos bajo estudio, se muestran más rápidas las transiciones de nivel lógico alto a bajo (siempre referenciadas a las salida de la puerta). De esta forma queda incluida esta peculiar característica de la lógica en cuestión.

Para la cadena de n puertas lógicas, la función retraso toma la forma:

$$\begin{aligned}
 D &= [\vartheta_{HL} \cdot D_{HL}(\beta_1) + \vartheta_{LH} \cdot D_{LH}(\beta_1)] + [\vartheta_{HL} \cdot D_{HL}(\beta_2) + \vartheta_{LH} \cdot D_{LH}(\beta_2)] + \dots \\
 &\dots + [\vartheta_{HL} \cdot D_{HL}(\beta_{n-1}) + \vartheta_{LH} \cdot D_{LH}(\beta_{n-1})] + [\vartheta_{HL} \cdot D_{HL}(\beta_n) + \vartheta_{LH} \cdot D_{LH}(\beta_n)] = \quad \{5.10\} \\
 &= \sum_{i=1}^n [\vartheta_{HL} \cdot D_{HL}(\beta_i) + \vartheta_{LH} \cdot D_{LH}(\beta_i)]
 \end{aligned}$$

donde se expresa cada término D_{HL} y D_{LH} a partir de {5.3}.

Los coeficientes son seleccionados según el sentido de las transiciones (alto a bajo o viceversa) a lo largo de la cadena de inversores (tanto los d_i , g_i , h_i , j_i , k_i , como los parámetros asociados a los condensadores).

5.4.1.2.- Estudio de la función de estimación de la potencia. Aplicación a una cadena de n puertas lógicas.

Dada una cadena de n puertas (reducidas a inversores equivalentes según se expone en 3.2), se computa la potencia consumida según la expresión siguiente {5.11}:

$$P = (1/2) \cdot \sum_{i=1}^n [P_{OH}^i + P_{OL}^i] \quad \{5.11\}$$

donde se computa el consumo de potencia promediando el consumo de cada puerta lógica cuando la salida en estado alto (P_{OH}) y cuando está en estado bajo (P_{OL}). Las expresiones para P_{OH}^i y P_{OL}^i , vienen dadas por {5.12}.

$$\begin{aligned}
 P_{OH} &= p_{H5} \beta^5 + p_{H4} \beta^4 + p_{H3} \beta^3 + p_{H2} \beta^2 + p_{H1} \beta^1 + p_{H0} = \\
 &= \sum_{i=0}^5 p_{Hi} \beta^i \quad \{5.12\}
 \end{aligned}$$

$$\begin{aligned}
 P_{OL} &= p_{L4} \beta^4 + p_{L3} \beta^3 + p_{L2} \beta^2 + p_{L1} \beta^1 + p_{L0} = \\
 &= \sum_{j=0}^4 p_{Lj} \beta^j
 \end{aligned}$$

La potencia total consumida por las puertas que integran la cadena será {5.13}:

$$P = (1/2) \cdot \sum_{i=1}^n [P_{OH}^i + P_{OL}^i] = P(\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) \quad \{5.13\}$$

Por lo tanto la potencia se expresa como suma de polinomios, de grado 4° para el cómputo de P_{OL} y de grado 5° para P_{OH} . A diferencia de las expresiones presentadas en la sección 3.2., se han escrito las fórmulas para el cómputo de la potencia en función de la variable β , en vez de términos de la variable w . Ello implica el ajuste numérico de los coeficientes polinómicos a la nueva situación (basta simplemente realizar la multiplicación oportuna). Se ha obrado de esta forma para lograr una formulación compacta.

Para la evaluación de la expresión anterior, se ha de tener en cuenta que en el cálculo de la potencia P_{OL} para la puerta β_i , la variable a introducir en la expresión de P_{OL} , es β_i . Sin embargo, para el caso de P_{OH} , la variable resulta ser $\beta_{fan-out}$. Esto es debido a que, P_{OL} , es función sólo del ancho de puerta del *pull-down* de la etapa inversora, mientras que P_{OH} es independiente de este valor, dependiendo sólo de las puertas que cargan la salida de la puerta.

Desde el punto de vista de la programación esto supone un tratamiento especial durante el cómputo de la función estimativa de la potencia.

5.4.1.3.- Estudio de la función de estimación del área. Aplicación a una cadena de n puertas lógicas.

Para cada transistor, la superficie ocupada se estima a través de la expresión:

$$A = \zeta \cdot \beta + \varphi \quad \{5.14\}$$

Los coeficientes que aparecen han sido definidos en la sección 3.3. Tal como ocurre en el caso de la potencia, precisan ser ajustados a la nueva situación en la que el área se evalúa en términos de β (basta realizar la multiplicación correspondiente).

Bajo la hipótesis de admitir sólo como variable el ancho del *pull-down* de la etapa inversora, el área ocupada por la cadena de n puertas se expresa como sigue:

$$\begin{aligned} A = (\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) &= A_{\beta_1} + A_{\beta_2} + \dots + A_{\beta_{n-1}} + A_{\beta_n} = \\ &= \sum_{i=1}^n [\zeta \beta_i + \varphi] \end{aligned} \quad \{5.15\}$$

Esta es la expresión más sencilla, desde el punto de vista matemático, siendo función

lineal de la variable de diseño β .

5.4.2.- Funciones de compromiso: relaciones.

Las soluciones que ofrecen un compromiso entre retraso D , P consumo de potencia y A , área ocupada, se determinan utilizando técnicas de programación no lineal.

A diferencia de lo que ocurre en circuitos CMOS, en el ámbito GaAs se consigue el mínimo retraso con las dimensiones mínimas posibles para los transistores (estando fijados los *buffers* de cada puerta según criterios definidos en el capítulo 4).

Este constituye un resultado que puede parecer incluso inverosímil en comparación con lo que acontece en circuitos CMOS. Este fenómeno ha sido mostrado con riguroso detalle en el capítulo 4.

Por lo tanto, carece de sentido el optimizar respecto a la función retraso y a la función área como compromiso, ya que tal compromiso no existe. Así, y como un avance de las conclusiones de esta tesis, se aconseja que si lo que pretende el diseñador es el diseño del circuito que opere a la máxima frecuencia posible, la solución resulta bien sencilla, basta diseñar el circuito empleando los transistores mínimos ofrecidos por la tecnología.

Como contrapartida, tal circuito ofrece el consumo de potencia máximo. Es este un aspecto a considerar, dado que, al ser el consumo estático, pudiera ser prohibitivo el diseño de un circuito GaAs *VLSI* con estas características.

El hecho de que la tecnología se escale hacia abajo en cuanto a la longitud del canal se refiere no implica un consumo de potencia infinitamente creciente (la dependencia estudiada en esta tesis aborda sólo la variable anchura de puerta, no la variable longitud de puerta).

Por consiguiente, las funciones que sí ofrecen compromiso para las soluciones son:

- Área y Potencia.
- Potencia y Retraso.
- Área, Potencia y Retraso.

Se desarrollará la metodología para este último problema. Resuelto éste, quedan

solucionados los dos anteriores.

5.4.3.- Coeficientes de peso.

Para llevar a cabo esta tarea, se construye la función suma ponderada de las tres funciones estimativas:

$$\begin{aligned}
 H(\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) &= \delta_D D^*(\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) + \delta_P P^*(\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) + \\
 &\quad + \delta_A A^*(\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) \\
 \delta_D &\in [0, 1] \\
 \delta_P &\in [0, 1] \\
 \delta_A &\in [0, 1] \\
 \delta_D + \delta_P + \delta_A &= 1
 \end{aligned}
 \tag{5.16}$$

Los coeficientes delta son los llamados coeficientes de peso. Según el valor de cada delta, se consigue que prime un criterio sobre el otro. Las funciones D^* , P^* y A^* , se corresponden con funciones adimensionales normalizadas a valores unidad (en el apartado 5.5.1 se comentan propiedades de esta función $H(\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n)$).

Antes de dar paso a la explicación del algoritmo empleado para la resolución del problema, se procede a construir la función objetivo para el caso de una cadena de n puertas lógicas. Se parte de la suposición de que la cadena a optimizar es el camino crítico correspondiente a un circuito y que se dispone de todos los coeficientes necesarios (retraso, potencia y área) para la correcta evaluación de las funciones estimativas.

Sea el esquemático (ver figura 5.1), donde la puerta lógica $n+1$ se ha añadido al final de la cadena y actúa como carga (un simple inversor en este caso).

Aplicando el concepto de inversor equivalente y calculando la capacidad equivalente por *fan-out*, el esquemático anterior se reduce al que se muestra en la figura 5.2.

5.4.4.- Cálculo de capacidades.

Cada condensador C_{L_i} , tiene una carga que se obtiene de la siguiente forma {5.17}, para

transición de nivel lógico alto a nivel lógico bajo a la salida de la puerta,

$$\begin{aligned}
 C_{L_1} &= C_{\pi_1} + C_{out1}^{HL} + \xi^{HL} \cdot \beta_{fan-out_1} \\
 C_{L_2} &= C_{\pi_2} + C_{out2}^{HL} + \xi^{HL} \cdot \beta_{fan-out_2} \\
 &\vdots \\
 &\vdots \\
 C_{L_{n-1}} &= C_{\pi_{n-1}} + C_{out(n-1)}^{HL} + \xi^{HL} \cdot \beta_{fan-out_{n-1}} \\
 C_{L_n} &= C_{\pi_n} + C_{outn}^{HL} + \xi^{HL} \cdot \beta_{fan-out_n}
 \end{aligned}
 \tag{5.17}$$

donde cada uno de los términos ha sido definido anteriormente.

Para el caso de transición de nivel lógico bajo a nivel lógico alto a la salida de la puerta, el valor de cada condensador se obtiene a través de las expresiones {5.18}.

$$\begin{aligned}
 C_{L_1}^* &= C_{\pi_1} + C_{out1}^{LH} + \xi^{LH} \cdot \beta_{fan-out_1} \\
 C_{L_2}^* &= C_{\pi_2} + C_{out2}^{LH} + \xi^{LH} \cdot \beta_{fan-out_2} \\
 &\vdots \\
 &\vdots \\
 C_{L_{n-1}}^* &= C_{\pi_{n-1}} + C_{out(n-1)}^{LH} + \xi^{LH} \cdot \beta_{fan-out_{n-1}} \\
 C_{L_n}^* &= C_{\pi_n} + C_{outn}^{LH} + \xi^{LH} \cdot \beta_{fan-out_n}
 \end{aligned}
 \tag{5.18}$$

5.4.5.- Cálculo de los coeficientes $\beta_{fan-out}$.

Cada elemento $\beta_{fan-out_i}$, es calculado a partir de las fórmulas {5.19}. Tal como expresan estas relaciones, en el cálculo de la capacidad de salida asociada al nodo i no se toma el valor de la w_{i+1} asociada al inversor equivalente.

En vez de eso, se evalúa la capacidad debida a la siguiente puerta, a partir del ancho del transistor al que realmente estaba conectado en el esquemático inicial (y no en el reducido a inversores equivalentes). Para el cómputo de la potencia se procede de igual forma.

Los términos tipo w_a , referencian a las puertas de carga (ver figura 5.1), y el subíndice que le acompaña da cuenta de cuál de los transistores de *pull-down* está conectado

al nodo de salida de la puerta anterior.

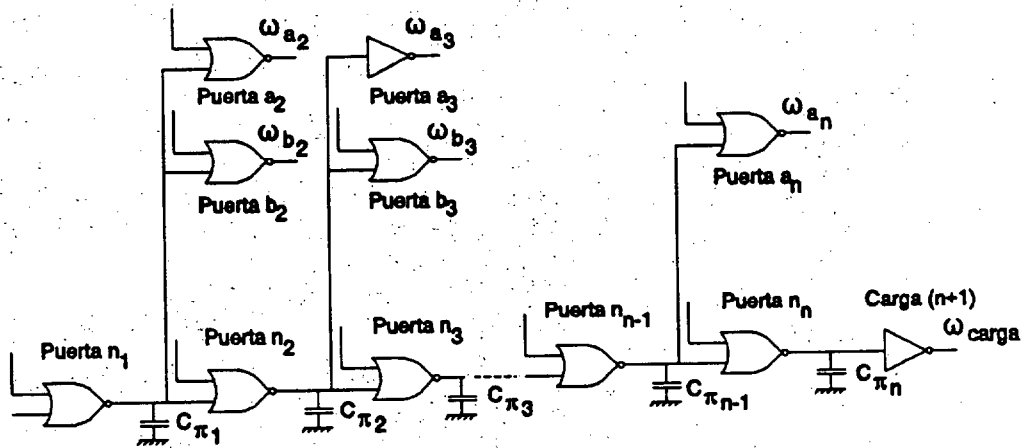


Fig. 5.1. Ejemplo de circuito a optimizar.

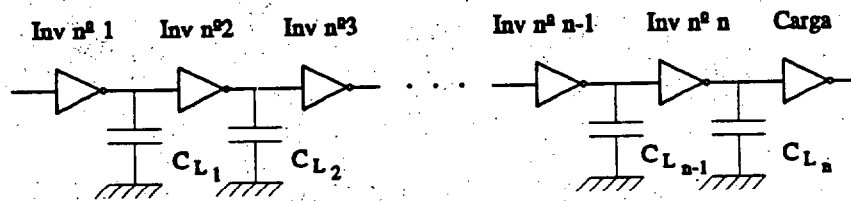


Fig. 5.2. Reducción del circuito a inversores equivalentes.

Sin embargo, a efectos de computar el retraso al atravesar cada puerta lógica se toma para la w del polinomio estimativo del retraso, la w del inversor equivalente. Esto se debe a que, al despreciar los efectos de colisiones de señales, se calcula el ancho efectivo por simple adición de los anchos de los transistores que conforman la puerta.

$$\begin{aligned}
 \beta_{fan-out_1} &= \left[\frac{W_{2_1}}{W_{pull-up_1}} + \frac{W_{a_2}}{W_{pull-up_2}} + \frac{W_{b_2}}{W_{pull-up_2}} \right] \\
 \beta_{fan-out_2} &= \left[\frac{W_{3_1}}{W_{pull-up_3}} + \frac{W_{a_2}}{W_{pull-up_2}} + \frac{W_{b_2}}{W_{pull-up_2}} \right] \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 \beta_{fan-out_{n-1}} &= \left[\frac{W_{n_1}}{W_{pull-up_n}} + \frac{W_{a_n}}{W_{pull-up_n}} \right] \\
 \beta_{fan-out_n} &= \left[\frac{W_{carga}}{W_{pull-up_{carga}}} \right]
 \end{aligned} \tag{5.19}$$

Una vez realizada la reducción del circuito a su equivalente en inversores equivalentes y conocidos los valores de cada una de las capacidades C_L (cableado y carga asociada al *fan-out*), se construyen las funciones estimativas tal como se ha indicado en los correspondientes apartados. De las expresiones anteriores, la función $H(\beta)$ a optimizar que relaciona las funciones estimativas viene dada por {5.20}.

$$\begin{aligned}
 \beta &= (\beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n) \\
 H(\beta) &= \delta_D \cdot D^*(\beta) + \delta_P \cdot P^*(\beta) + \delta_A \cdot A^*(\beta) = \\
 &= \delta_D \cdot \sum_{i=1}^n [\vartheta_{HL} \cdot D_{HL}(\beta_i) + \vartheta_{LH} \cdot D_{LH}(\beta_i)] + \delta_P \cdot (1/2) \cdot \sum_{i=1}^n [P_{OH}^i + P_{OL}^i] + \delta_A \cdot \sum_{i=1}^n [\zeta \cdot \beta_i + \varphi] = \tag{5.20} \\
 &= \sum_{i=1}^n [\delta_D \cdot [\vartheta_{HL} \cdot D_{HL}(\beta_i) + \vartheta_{LH} \cdot D_{LH}(\beta_i)] + \delta_P \cdot (1/2) \cdot [P_{OH}^i + P_{OL}^i] + \delta_A \cdot [\zeta \cdot \beta_i + \varphi]]
 \end{aligned}$$

En esta expresión, cada función estimativa es adimensional y se encuentra correspondientemente normalizada a su valor unidad (se han suprimido los asteriscos para hacer la notación más clara). Se indica en **negrita** que β es un vector n -dimensional. Esta función $H(\beta)$ es la que relaciona todas las dimensiones de los transistores presentes en el camino crítico a optimizar.

5.4.6.- Condiciones de contorno.

Para la resolución del problema, se precisan las condiciones de contorno (restricciones)

pertinentes. En este caso, vienen dadas directamente por las reglas de diseño de la tecnología.

Así, existe un tamaño (β) mínimo para cada tipo de puerta, bien sean inversores, puertas NOR o puertas OR. Además, dicho valor queda a su vez restringido por el *fan-out* a la salida de la puerta. Esta es una situación bien diferente al caso CMOS, la cual se detalló en el capítulo anterior.

Esta condición se expresa matemáticamente como:

$$\{\forall\{\text{puerta lógica}\}\} \cap \{\forall\beta \in [\beta_{\text{permitidas}}]\} \text{ ha de ser: } \beta \geq \beta_{\text{min}} \quad \{5.21\}$$

A su vez, el consumo de potencia y el área ocupada limitan el tamaño máximo de los transistores a cierto valor. Existirá un valor máximo para cada tipo de puerta. Suponer esta condición como condición de contorno servirá para evitar puntos calientes en el circuito. Se entiende por punto caliente a aquella zona del circuito donde se disipa un valor alto de potencia, que pudiera ser peligroso para el correcto funcionamiento del circuito.

Realizando varias pasadas del algoritmo y fijando convenientemente los valores topes para la dimensión de los transistores, puede distribuirse adecuadamente la disipación de la potencia, evitando los peligrosos puntos calientes. Esta condición se define matemáticamente como {5.22}.

$$\{\forall\{\text{puerta lógica}\}\} \cap \{\forall\beta \in [\beta_{\text{permitidas}}]\} \text{ ha de ser: } \beta \leq \beta_{\text{max}} \quad \{5.22\}$$

La definición del problema se da por concluida, ahora, sólo resta elegir el algoritmo numérico adecuado.

5.4.7.- Estudio de la convexidad de la función objetivo.

Como paso previo a la exposición del algoritmo de optimización, resulta imprescindible acometer el estudio de la convexidad de la función a optimizar.

A diferencia de lo que acontece para las funciones objetivos usuales en el caso CMOS, no resulta tarea fácil asegurar la convexidad de la función objetivo que se ha

presentado en este capítulo. Ello es consecuencia directa de la naturaleza de las funciones estimativas manejadas (en especial la función estimativa de retardos de propagación). Se asegura la convexidad de una función de n variables cuando la matriz jacobiana asociada a dicha función es definida positiva para todos los valores posibles de las n variables [VlaSi83].

Para el caso en cuestión, resulta prácticamente imposible (matemáticamente) el demostrar que la matriz jacobiana asociado a la función {5.20} es definida positiva por dos razones:

- la función objetivo posee n -variables,
- los coeficientes asociados a los polinomios provienen de un procedimiento de ajuste, pudiendo variar el signo asociado a cada uno de ellos.

Estas dos razones imposibilitan el demostrar matemáticamente la convexidad de la función objetivo {5.20}. Sin embargo esto no constituye un problema insalvable; se proponen dos alternativas para abordarlo.

5.4.7.1.- Restricción al caso de dos variables.

Como primera de ellas, se opta por restringir el problema al caso de dos variables (2 puertas lógicas) y representar gráficamente la función objetivo en el dominio de valores apropiado. De esta forma, se dispondrá de un procedimiento *intuitivo* para observar la posible convergencia de dicha función. Dando por hecho el que este proceder no constituye ninguna demostración, se hace notar que lo que se pretende es estudiar el comportamiento cualitativo de la función (número de posibles mínimos locales) y especialmente observar posibles problemas en la frontera del dominio.

5.4.7.1.1.- Función estimativa de la potencia: restricción al caso de dos variables.

Se comienza estudiando la función estimativa potencia para el caso de dos variables, es decir, el caso de una cadena de tres puertas lógicas (donde la última actúa de carga, no siendo considerada su dimensión como variable).

El conjunto de valores que se representa en la figura 5.3 se extiende al dominio de valores típicamente encontrados en los circuitos estudiados.

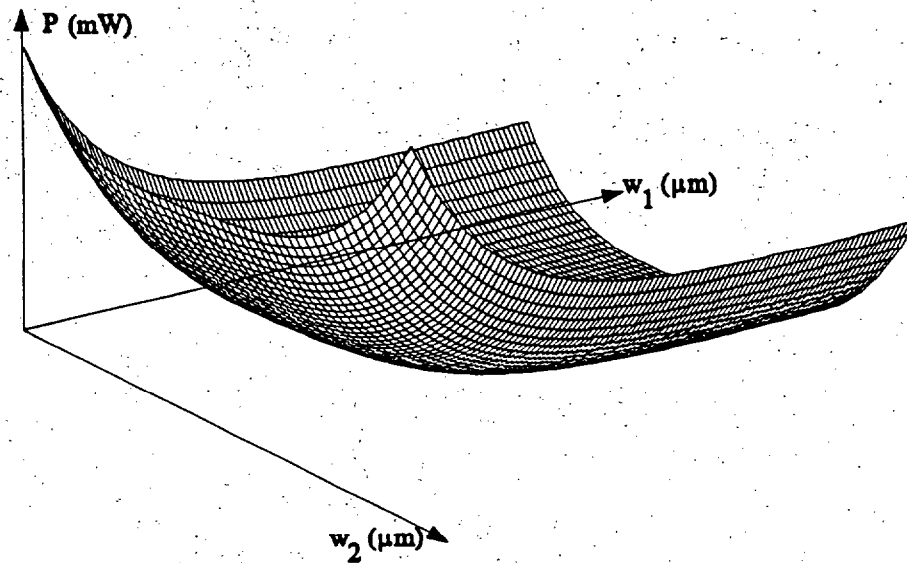


Fig. 5.3. Función estimativa de la potencia (caso particular de dos variables).

Se observa claramente que la función para el caso de dos variables es convexa, con los máximos localizados en el borde (lo cual era de esperar). Además, sólo se aprecia la existencia de un valle de mínimos (esto se interpreta desde el punto de vista de la programación como excelente). La existencia de este valle posibilita el que al conjuntar dicha función con la estimativa de retardos y área, el conjunto de soluciones óptimas se acrecienta. Dado que la función estimativa de la potencia para el caso de una cadena de n puertas se obtiene mediante simple adición de las funciones estimativas parciales, nada parece indicar que la función en n -variables para la potencia presente un comportamiento no convexo.

El algoritmo que se presenta en este capítulo convergió en todos los casos a los valores mínimos que en esta gráfica se representan sin importar el valor inicial dado a las variables.

5.4.7.1.2.- Función estimativa del retardo de propagación: restricción al caso de dos variables.

Para estudiar la convexidad en el caso de la función estimativa del retardo de propagación, se hace imprescindible el fijar la variable pendiente de la señal a propagar. Se eligió un valor típico de doscientos (200) picosegundos. Asimismo el circuito contiene las correspondientes capacidades asociadas el interconexiónado (valores usuales de valor aproximado a diez

femtofaradios).

Bajo este marco, se obtiene la figura 5.4, cuyo comportamiento matemático resulta similar al ofrecido por la función potencia. Se indica que la escala para los valores de las variables es la misma en ambos casos. Tal como sucede para el caso anterior, el algoritmo converge rápidamente a los valores mínimos observados desde cualquier valor inicial para las variables.

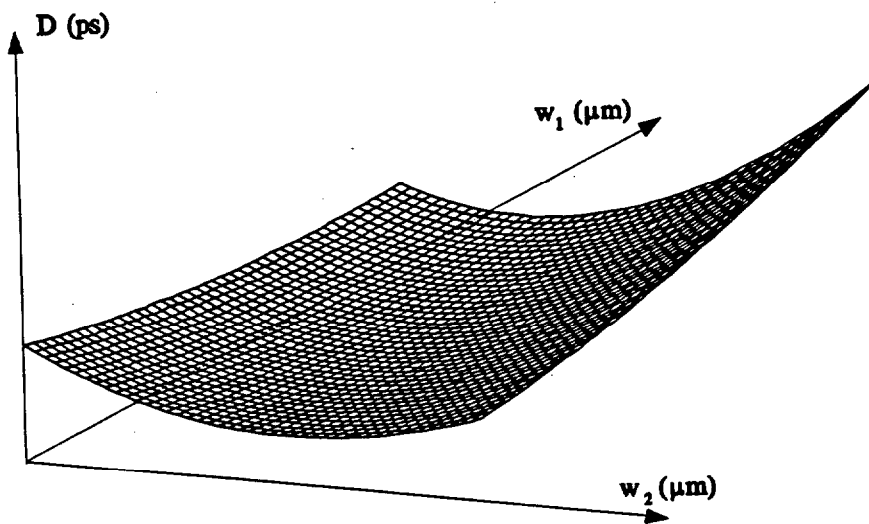


Fig. 5.4. Función estimativa del retardo (caso particular de dos variables).

5.4.7.1.3.- Representación de la función objetivo para el caso restringido empleando dos variables.

Se presenta en la figura 5.5 la función objetivo comprendiendo las tres funciones estimativas (normalizadas a valores unidad), restringida al caso particular de dos variables y para un conjunto de coeficientes de peso dados.

La función estimativa del área ocupada no se ha representado anteriormente por separado dado que se corresponde con un plano.

Según el monótono comportamiento de la función objetivo parece adecuado un método de búsqueda de mínimos según un criterio descendente.

Tal como se expone en el apartado 5.5, se optó por el método del gradiente

descendente (que posee paso de cálculo adaptativo).

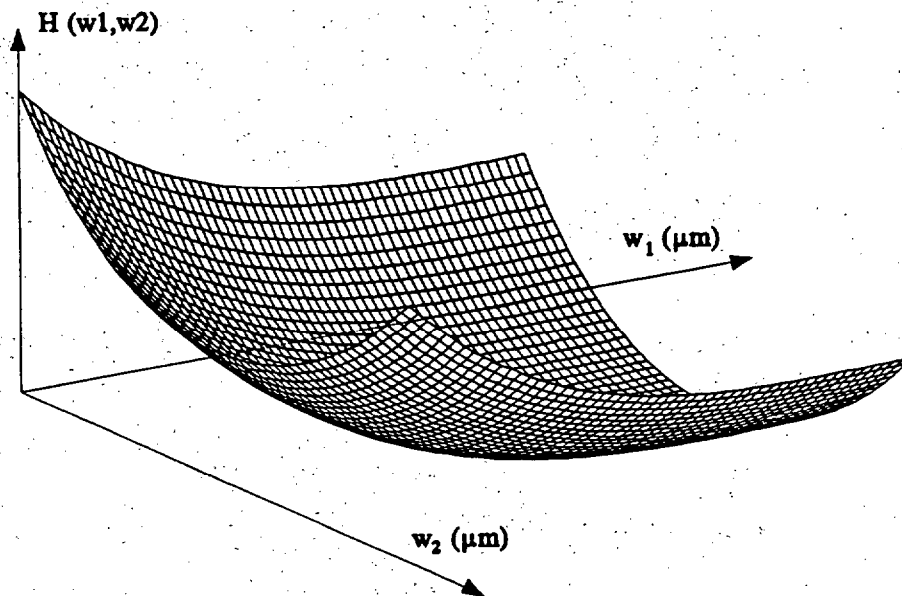


Fig. 5.5: Función objetivo (caso de dos variables).

Otros muchos casos han sido estudiados según la restricción a dos variables (variando cargas, pendientes y factores de peso), dando como resultado diagramas cuyo comportamiento cualitativo no difiere en exceso del que se ha expuesto. Los datos numéricos más significativos (valor y posición de los mínimos obtenidos) se detallan en el capítulo de resultados (capítulo 6).

5.4.7.2.- Programación no convexa.

Como segunda alternativa se propone el admitir la posibilidad de que la función objetivo sea no convexa y aplicar un procedimiento de búsqueda algorítmico que se detenga cuando encuentre un mínimo local. Seguidamente se reinicializa la búsqueda a partir de distintas soluciones pruebas iniciales con el fin de encontrar cuantos mínimos locales distintos sean posibles. De encontrarse más de un mínimo, se elige el mejor de éstos (es decir, *el más mínimo*).

Estas técnicas de búsqueda se enmarcan dentro de la categoría de las llamadas técnicas de minimización restringida secuencial [FiaGa68]. Su aplicación al caso que concierne, tal como se ha comentado en la sección anterior ha conducido en todas las ocasiones a una única

solución óptima, lo cual proporciona fundadas razones para afirmar la conjetura de la convexidad de la función objetivo manejada en este capítulo.

5.5.- ELECCIÓN DEL ALGORITMO NUMÉRICO PARA LA RESOLUCIÓN DEL PROBLEMA.

Se trata de obtener el mínimo de la función {5.20} sometida a las restricciones de contorno {5.21} y {5.22}.

Se clasifica como un problema de programación multivariable no lineal, que ha de ser solucionado vía algoritmo numérico.

5.5.1.- Conjunto de soluciones no inferiores.

Se pretende obtener el conjunto de soluciones no inferiores para la función {5.20}. Un conjunto óptimo de dimensiones de transistores $\beta = [\beta_n]$ se denomina *no inferior*. Se dice que β^* es no inferior si no existe otra alternativa de diseño β que mejore al menos una de las funciones estimativas sin degradar ninguna de las otras.

Para el problema reducido de una función objetivo unidimensional (es decir, con una sola variable), la forma de determinar este conjunto resulta sencilla: bastaría una representación gráfica de los valores de ambas funciones (gráfico x-y) que hagan mínima la función objetivo correspondiente. Todos los puntos pertenecientes a las gráficas serían soluciones no inferiores y el punto de corte de dichas funciones coincide con la mejor solución posible.

Para el problema multidimensional no es posible la representación gráfica para determinar la intersección (las funciones estimativas no se cortarían en un punto sino en una curva, superficie...).

La minimización de la función {5.20} nos proporciona las soluciones no inferiores de una forma sencilla. Se puede demostrar que si las funciones estimativas son convexas, toda solución que minimiza la función objetivo es no inferior, y que se pueden encontrar todas las soluciones no inferiores con una variación sistemática de los coeficientes de peso (ver demostración en [ChaHa83]).

Para ilustrar esto tomamos el problema ejemplo de obtener las soluciones no inferiores para el compromiso retardo-potencia. Se construye la función objetivo correspondiente (basta un sólo coeficiente de peso):

$$H(\beta) = \alpha \cdot D(\beta) + (1-\alpha) \cdot P(\beta) \quad \{5.23\}$$

Con respecto a esta función podemos destacar dos importantes características:

- resulta ser una combinación lineal de las funciones estimativas $D(\beta)$ y $P(\beta)$ y,
- la derivada de esta función respecto de la variable alfa igualada a cero (condición de extremo) expresa la condición de corte de las funciones estimativas.

Esta última propiedad nos garantiza la existencia de la intersección de ambas funciones, es decir la condición óptima para aquellos valores de β que anulen dicha derivada.

La estrategia a seguir para encontrar las soluciones no inferiores consiste en minimizar la función objetivo {5.23} que se obtiene para cada alfa -desde el valor 0 hasta el valor 1- con respecto de las dimensiones de los transistores. Dicha operación nos proporcionará las *betas* que minimizan a cada función estimativa. Esto es cierto siempre que las funciones estimativas sean definidas positivas en el conjunto de soluciones posibles (condición que se cumple para las tres funciones estimativas retardo, potencia y área).

Una vez conocido el conjunto de soluciones no inferiores, basta representar en un gráfico los valores de las funciones estimativas en función de alfa. Es decir, dado un valor de alfa,

- se minimiza la función {5.23} (se obtiene el conjunto de dimensiones no inferiores) y,
- se evalúan las funciones retardo-potencia para esas dimensiones.

Repitiendo el proceso (barriendo todos los valores de alfa), se obtiene el gráfico que permite obtener el punto de corte de ambas funciones (ver ejemplos en el capítulo 6). Nótese que este sencillo procedimiento permite hallar la solución de compromiso de forma gráfica para el problema n -dimensional.

De la discusión anterior queda claro que la minimización de la función objetivo nos proporciona el conjunto de soluciones no inferiores. Para obtener la solución óptima (el valor mínimo para las funciones estimativas) es necesario calcular todo el conjunto de soluciones no inferiores.

Se presenta en el siguiente apartado el algoritmo que se emplea para la minimización de la función objetivo {5.20}

5.5.2.- Algoritmo de minimización de la función objetivo.

Ya se han comentado los diferentes intentos que hemos hecho de reformular el problema para su más sencilla resolución (bien empleando programación numérica o métodos estándar primal-dual).

Tras la búsqueda de un método sencillo y eficaz, se ha decidido aplicar el método del gradiente descendente sometido a condiciones de contorno. Hacemos notar que hemos descartado otros métodos debido también a problemas de convergencia (se recuerda que, dado el orden de magnitud de algunos de los coeficientes, es preciso operar con números en coma flotante de doble precisión).

Antes de aplicarlo al problema en cuestión se procederá a detallar como opera en esencia este algoritmo (para consulta más detallada, ver por ejemplo [LasWa78]).

El método del gradiente descendente se basa en buscar el máximo (el mínimo se calcula de igual forma sin más que cambiar el signo del parámetro λ que aparece en el desarrollo) de la función $f(\mathbf{x})$ (donde \mathbf{x} es un vector de n variables) según un procedimiento iterativo de búsqueda según la dirección del máximo gradiente, hasta que (en esencia) se alcance una solución óptima \mathbf{x}^* en la que $\nabla f(\mathbf{x}) = 0$.

Sin embargo, no resultaría práctico cambiar \mathbf{x} continuamente en la dirección de $\nabla f(\mathbf{x})$, ya que esta serie de cambios requeriría una *reevaluación continua* de $\partial f / \partial x_j$ y el cambio de dirección de la trayectoria.

Por consiguiente, una mejor forma de proceder es continuar el movimiento en una dirección *fija* a partir de la solución prueba actual, sin detenerse hasta que $f(\mathbf{x})$ deje de aumentar. Este punto de detención sería la siguiente solución prueba, por lo que se debe volver a calcular el gradiente para determinar la nueva dirección de movimiento. Con este

enfoque, cada iteración supone cambiar la solución prueba *actual* x' de la siguiente forma:

$$\text{Modificar } x' = x' + \lambda \cdot \nabla f(x) \quad \{5.24\}$$

donde λ^* , es el valor positivo de λ que *maximiza* $f(x' + \lambda \cdot \nabla f(x'))$; es decir:

$$f(x' + \lambda \cdot \nabla f(x')) = \max_{\lambda > 0} f(x' + \lambda \cdot \nabla f(x')) \quad \{5.25\}$$

[Nótese que $f(x' + \lambda \cdot \nabla f(x'))$ es sencillamente $f(x)$ donde:

$$x_j = x_j + \lambda \cdot \left[\frac{\partial f}{\partial x_j} \right]_{x=x'} , \quad j = 1, 2, \dots, n. \quad \{5.26\}$$

y estas expresiones para la x_j incluyen sólo constantes y λ , de manera que $f(x)$ se convierte en una función de una sola variable].

Las iteraciones de este procedimiento de búsqueda del gradiente continúan hasta que $\nabla f(x) = 0$, dentro de una pequeña tolerancia ϵ , o sea, hasta que:

$$\left| \frac{\partial f}{\partial x_j} \right| \leq \epsilon, \quad \text{para toda } j = 1, 2, \dots, n. \quad \{5.27\}$$

Desde el punto de vista de la programación, se puede desglosar el algoritmo en los siguientes pasos:

- Paso inicial: Elegir el valor de ϵ y valor inicial para el vector x' . Se verifica la regla de detención.
- Paso iterativo n° I :

Se expresa $f(x' + \lambda \cdot \nabla f(x'))$ como función de λ estableciendo {5.28}:

$$x_j = x_j' + \lambda \cdot \left(\frac{\partial f}{\partial x_j} \right)_{x=x'} , \quad \text{para } j = 1, 2, \dots, n. \quad \{5.28\}$$

Posteriormente se sustituyen estas expresiones y λ en $f(x)$.

- Paso iterativo n° 2:

Se utiliza el procedimiento de búsqueda en una dimensión para encontrar $\lambda = \lambda^*$, valor que maximiza:

$$f(x' + \lambda \cdot \nabla f(x')) \text{ para } \lambda \geq 0 \quad \{5.29\}$$

- Paso iterativo n° 3:

Se hace $x' = x' + \lambda^* \cdot \nabla f(x')$. Seguidamente se pasa a chequear la regla de detención.

- Regla de detención.

Se evalúa $\nabla f(x')$ en $x = x'$. Se verifica si:

$$\left| \frac{\partial f}{\partial x_j} \right| \leq \epsilon, \text{ para toda } j = 1, 2, \dots, n. \quad \{5.30\}$$

Si se verifica dicha condición, el proceso se detiene con la x' actual como la aproximación a una solución óptima deseada x^* . De no ser así, se regresa al paso iterativo.

Un aspecto importante de este método es que, convierte el problema multivariable en uno de una sola variable, λ .

En los casos en que sea posible, se resuelve la ecuación en la variable λ de forma analítica. En el problema bajo estudio esto no es realizable, y dicha ecuación se resuelve mediante el método numérico de Newton-Raphson.

Dada la circunstancia de que se parte de un conjunto de puertas lógicas ya de por sí optimizadas, la solución del problema no se encuentra en ninguna ocasión numéricamente distante de la solución óptima a hallar.

Con esto se pretende afirmar que, tanto el proceso iterativo de búsqueda del gradiente, como el proceso de resolución mediante el algoritmo de Newton-Raphson, no requieren gran número de iteraciones.

En todos los casos estudiados, bastaron menos de 20 iteraciones para resolver el problema. Este bajísimo número de iteraciones se traduce directamente en un bajísimo coste

computacional, lo cual posibilita la aplicación de la metodología que se presenta.

5.5.3.- Aplicación del algoritmo al ejemplo de n puertas lógicas.

Para la minimización de la función objetivo {5.20}, el algoritmo anterior se aplica de la siguiente forma (supuestos conocidos todos los parámetros necesarios):

- Como conjunto de valores iniciales se tomaría el conjunto finito:

$$\Omega_0 = \{\beta_0, \lambda_0, \delta, \tau\}, \text{ siendo:}$$

$$\beta_0 = \{\beta_1, \beta_2, \dots, \beta_{(n-1)}\}$$

$$\delta = \{\delta_D, \delta_P, \delta_A\}$$

$$\tau = \{\tau_{HL}, \tau_{LH}\}$$

{5.31}

- Se define cada variable β_i , en función de λ a partir de:

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_{n-2} \\ \beta_{n-1} \end{pmatrix}^{k+1} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_{n-2}^k \\ \beta_{n-1} \end{pmatrix}^k + \lambda \cdot \begin{pmatrix} -\partial H / \partial \beta_1 \\ -\partial H / \partial \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ -\partial H / \partial \beta_{n-2} \\ -\partial H / \partial \beta_{n-1} \end{pmatrix}^k \quad \{5.32\}$$

El supraíndice indica la iteración correspondiente. Para $k = 0$, la matriz que contiene las dimensiones (*betas*) se llena con los valores correspondientes al predimensionado inicial de los transistores. Se recuerda que la puerta n -ésima no es una variable.

El signo negativo que acompaña a las derivadas parciales se debe a que para minimizar la función se ha de cambiar el signo del parámetro λ que aparece en la ecuación {5.24}. Como paso posterior, se sustituyen los valores de $\beta_i^{(k+1)}$ en la función $H(\beta)$. Con ello se logra expresar dicha función en términos del parámetro λ .

Para hallar el nuevo vector de soluciones, se calcula la dirección del máximo gradiente de la función $H = H(\lambda)$, es decir:

$$\beta^{k+1} = (\beta_1^{k+1}, \beta_2^{k+1}, \dots, \beta_{n-2}^{k+1}, \beta_{n-1}^{k+1})$$

$$H = H(\beta^{k+1}) = H(\lambda) \quad \{5.33\}$$

$$\text{cálculo de } \lambda \Rightarrow dH(\lambda)/d\lambda = 0$$

Para la resolución de esta ecuación se emplea el algoritmo de Newton-Raphson, que resuelve haciendo:

$$\lambda^{n+1} = \lambda^n - \frac{dH(\lambda^n)/d\lambda}{d^2H(\lambda^n)/d\lambda^2} \quad \{5.34\}$$

En el programa implementado, se desarrollaron las derivadas de forma analítica. Estas expresiones se integran en el programa y mediante bucles, se logra el cálculo de todas las derivadas que se precisan. De esta forma se evita el tener que realizar las derivadas de forma numérica (con el coste computacional elevado que ello requiere). Mediante este proceder, se consigue:

- precisión en el cómputo de las funciones y,
- acelerar la evaluación de las funciones derivadas.

Una vez que se conoce el nuevo valor del parámetro λ , se dispone del nuevo vector $\beta^{(k+1)}$.

Se precisa comprobar que se satisfacen las condiciones de contorno. Para los valores que satisfagan dichas condiciones, el algoritmo se ejecuta sin alteración alguna. Para los valores que se salgan del dominio de definición, se ha seguido la estrategia de elección que se detalla en la figura 5.6. Tal como se desprende de esta figura, si un nuevo valor de β no está contenido en el dominio de definición, se elige el valor permitido más próximo (bien $\beta_{máx}$ o $\beta_{mín}$). Con los nuevos valores permitidos para las nuevas dimensiones de los transistores, se prosigue con el algoritmo, hasta llegar a la condición de parada.

5.6.- IMPLEMENTACIÓN DEL ALGORITMO: PROGRAMA DESARROLLADO.

Este algoritmo expuesto ha dado paso a la implementación del programa que a continuación

se presenta.

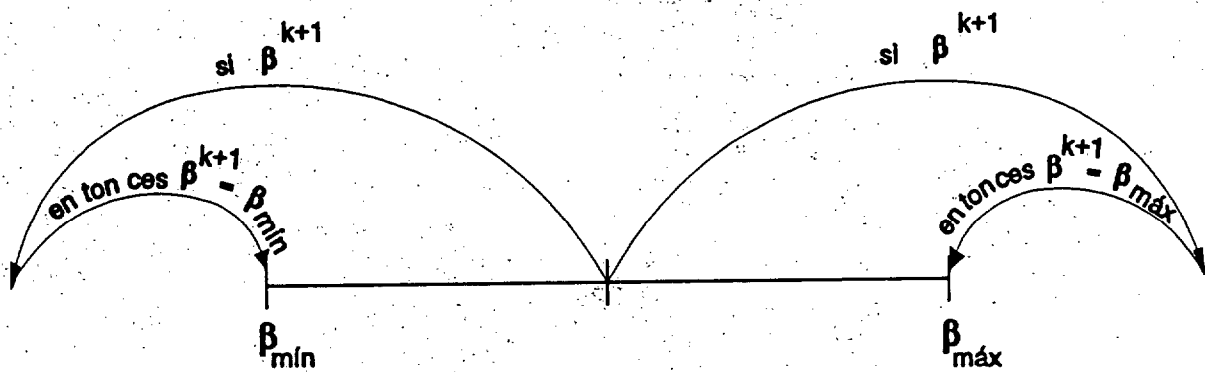


Fig. 5.6. Criterio de ajuste al dominio de definición.

5.6.1.- Descripción del programa: diagrama de flujo.

Se describe el programa mediante el comentario de cada una de las cajas que componen el diagrama de flujo (ver figura 5.7). Como datos iniciales, el programa requiere:

- fichero conteniendo el conjunto de parámetros necesarios (coeficientes para el cálculo del retraso, potencia, área y capacidades equivalentes). Estos valores se encuentran contenidos en ficheros lo cual permite cambios sin necesidad de nueva compilación (proporcionando transportabilidad y fácil adaptación a nuevos ficheros tecnológicos),
- conexionado de puertas y valores de las capacidades asociadas al cableado. Esta etapa presupone haber aplicado las técnicas de "buferrización" desarrolladas en el capítulo 4,
- valores iniciales para el predimensionado de los transistores y valor inicial para la variable λ (necesaria para la resolución de la ecuación {5.33}),
- valores de los coeficientes de peso (tanto para la función objetivo como para las transiciones de las señales).

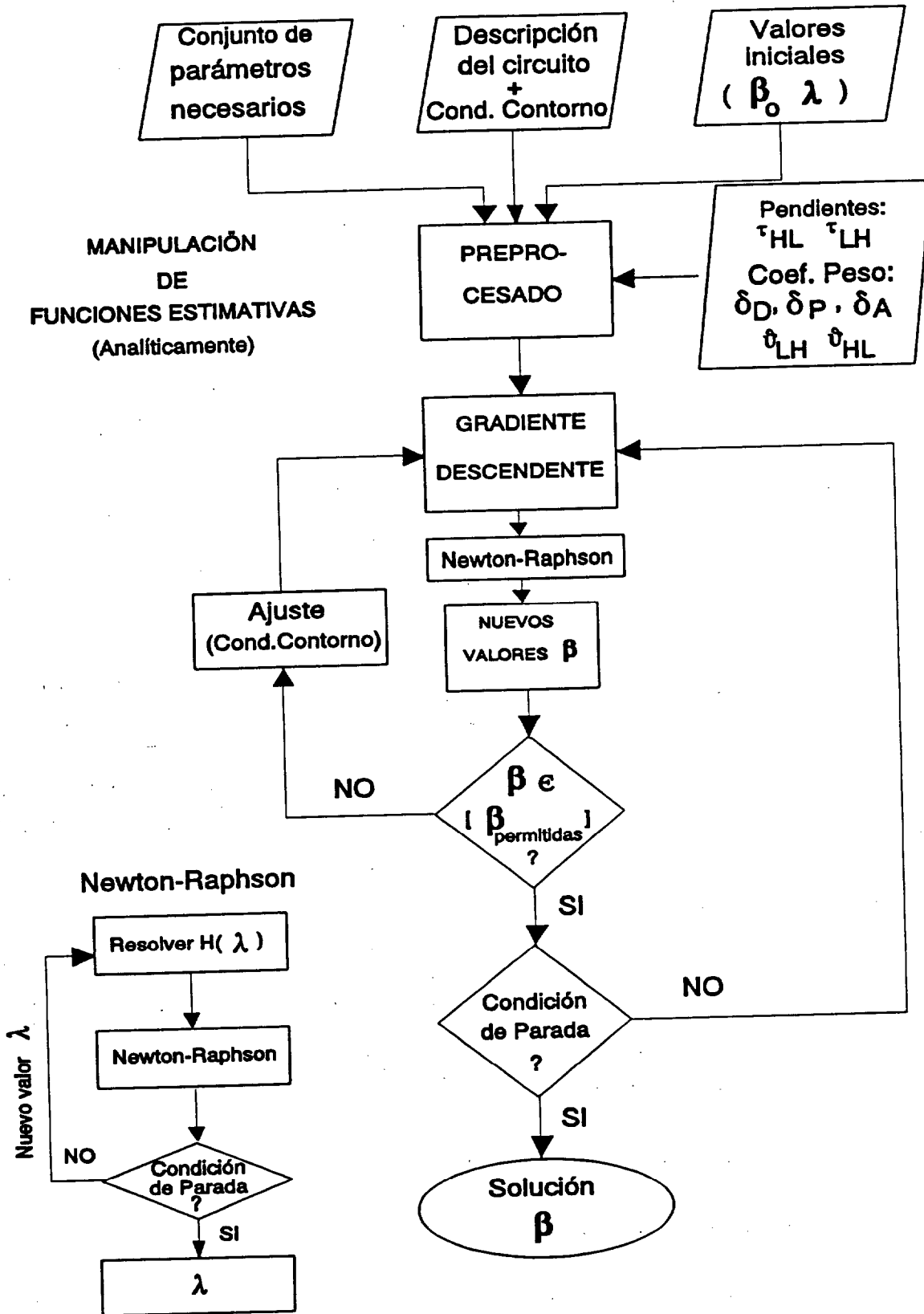


Fig. 5.7. Diagrama de flujo del programa.

A continuación, se comenta el desarrollo del algoritmo:

- evaluación del gradiente,
- cómputo de las nuevas soluciones para el dimensionado de transistores,
- verificación de las condiciones de contorno:
 - de no verificarse, se desvía el flujo del programa hacia la subrutina de ajuste,
 - de verificarse, se continúa el flujo del programa,
- chequeo de la condición de parada y/o parada/vuelta a la iteración siguiente.

La subrutina encargada del proceso iterativo (gradiente descendente), contiene un conjunto de subrutinas necesarias para la resolución de la ecuación {5.33}, vía método de Newton-Rahpson {5.34}.

El método del gradiente es siempre convergente al menos a una solución mínima local [Polak71]. Puesto que el dominio posible de soluciones es conocido, un número reducido de iteraciones es suficiente para asegurar la convergencia. En particular, sobre el conjunto de ejemplos estudiados (algunos de ellos partiendo de valores del dimensionado de transistores para nada próximos a la solución), se afirma que el programa ha convergido en no más de 20 iteraciones.

La subrutina de cálculo de la ecuación {5.33} resulta ser la más importante desde el punto de vista de la convergencia. Si bien el algoritmo Newton-Raphson tiene asegurada la convergencia para valores cercanos a la solución [ChuLi75], no es así para valores de la variable a obtener situados a una distancia considerable de la solución.

Además se ha de tener en cuenta que el cociente que aparece en la ecuación {5.34}, no puede ser igual a cero (condición a comprobar en el programa). A estos aspectos hemos de añadir el hecho de que se computan derivadas de segundo orden (para el cálculo del gradiente sólo se evalúan derivadas de primer orden).

Queda claro que la convergencia o no convergencia del programa se encuentra fuertemente condicionada al valor inicial que se adopte para la variable λ y a su evolución durante el desarrollo del algoritmo.

Dada la dificultad que presenta el demostrar matemáticamente las condiciones de convergencia del algoritmo de Newton-Raphson para todo valor de λ , la alternativa que se

ha seguido consiste en detectar la no convergencia del algoritmo (limitando el número de iteraciones) y en relanzar el mismo a partir de un valor diferente de la solución inicial adoptada (según cierto criterio determinado). Procediendo de esta forma, afirmamos que sobre el amplio conjunto de ejemplos estudiados el algoritmo ha convergido siempre.

Queda de esta forma abierta una línea futura de trabajo centrada en el refinamiento de los algoritmos existentes, por ejemplo, incorporación del método de Newton-Raphson modificado para aquellos casos en los que la pendiente de la función se hace nula. Es posible la inclusión de rutinas de seguridad que garanticen la convergencia incluso bajo situaciones extremas (cargas no típicamente usadas por los diseñadores, atacadas con pendientes de excitación fuera del rango normal de trabajo...).

Como salida el programa proporciona el dimensionado de transistores adecuado a los factores de peso preseleccionados. Además se incorpora una subrutina que permite hacer múltiples análisis variando los valores de los coeficientes de peso (cada coeficiente desde el valor inicial cero hasta el valor final), para construir las gráficas potencia-área-tiempo para el circuito en estudio. Pueden consultarse ejemplos completamente desarrollados en el capítulo 6 (resultados).

El programa desarrollado no sólo calcula el conjunto de soluciones no inferiores para el circuito a optimizar, sino que permite asimismo la exploración de otros importantes aspectos. Por ejemplo, resulta sencillo explorar qué rango de pendientes de excitación son las idóneas.

Esta posibilidad no se encuentra automatizada actualmente, pero bastaría incorporar un bucle global que realizara el oportuno barrido de pendientes y seleccionara de entre ese conjunto la pendiente óptima. Se indica que no conocemos ningún programa existente que incorpore esta posibilidad.

La distribución de puntos calientes en el circuito constituye otra singular aportación del programa. En el menú de entrada de datos, el usuario puede optar entre dos posibilidades:

- dimensionado mínimo-máximo global (que afecta a cada una de las puertas existentes) o,
- dimensionado mínimo-máximo local (particular para cada puerta en cuestión).

De esta forma puede limitarse el tamaño mínimo de un transistor determinado

consiguiendo que en ese punto del camino crítico el consumo de potencia quede acotado al valor que se desee. Estas condiciones afectan directamente al conjunto de betas permitidas, existiendo en el caso de que se opte por la elección segunda, un rango de valores (ver figura 5.6) para cada puerta lógica.

Pueden hacerse los mismos comentarios en el caso de la función área ocupada. De nuevo señalamos que no conocemos ningún programa ó referencia existente que incorpore esta posibilidad.

El tiempo de CPU requerido para resolver un problema determinado (ver más detalles en el capítulo 6), es función principalmente de:

- número de puertas lógicas consideradas y,
- valores del predimensionado inicial elegido.

De estos dos aspectos, el más crítico es el primero de ellos, porque el segundo desaparece en las primeras iteraciones.

El tiempo necesario para optimizar una cadena cualquiera es independiente del fan-out y carga capacitiva que presenten los nodos (siempre que se encuentren contenidas en el rango adecuado de trabajo). La única diferencia radica en el tiempo que se emplea en llenar las matrices fan_out[i] (correspondiente a la puerta i de la cadena) y las matrices capacidad_hl[i], capacidad_lh[i]. Para circuitos con un número de puertas incluso grande, este tiempo es completamente despreciable.

A título comparativo se cita que la obtención del conjunto completo de soluciones no inferiores (compromiso retardo-potencia) para una cadena de 13 puertas lógicas consume un tiempo de CPU en una estación de trabajo SUN SPARC2 inferior a 1 segundo. El problema similar para tecnología CMOS resuelto por Hoppe y colaboradores consume 50 segundos aproximadamente [HoNeS90] en una máquina de características similares.

El programa se ha escrito en lenguaje C y corre sobre estación SUN bajo entorno UNIX (también corre en PC bajo el sistema operativo DOS). El número de líneas de código es aproximadamente 2000, de las cuales la mayoría se emplean en la obtención de la derivada primera (respecto de la variable β y respecto de la variable λ) y de la derivada segunda (respecto de la variable λ).

En este programa también se incorporan distintas rutinas de seguridad y de chequeo de condiciones. Así, se chequea la existencia de los ficheros de datos (coeficientes) indicando un error si éstos ficheros no están disponibles.

Los valores mínimo-máximo que se demandan se someten al chequeo de pertenencia o no al rango de valores permitidos. También se computan los signos de pendientes así como de los distintos coeficientes de peso.

Dada las magnitudes de los números con los que el programa opera, se requiere trabajar con números en coma flotante de doble precisión (especialmente en la aplicación de Newton-Raphson).

Para los casos prácticos estudiados no se ha empleado asignación dinámica de memoria (funciones tipo `alloc()` o `malloc()`) así como no se han explotado las posibilidades de la programación en paralelo. Dada la estructura modular del programa, toda modificación es sencilla de realizar.

5.6.2.- Ampliaciones del programa.

Como ampliación del programa se recogen las siguientes:

- desarrollar subrutinas capaces de extraer el conjunto de coeficientes necesarios a partir de:
 - los ficheros tecnológicos,
 - las características de la lógica (tensiones alimentación, cargabilidad...),
 - las reglas de diseño (caso particular del modelo de área),
 - integrarlo en un entorno de diseño estándar (tipo CADENCE, por ejemplo).

Al mismo tiempo que se integraría el programa, se integraría el analizador temporal, permitiendo realizar tanto la captura de esquemáticos como la búsqueda de los caminos críticos.

Esta última ampliación permitiría asimismo incorporar un compilador y compactador de layout, cerrando de esta forma el amplio bucle de la síntesis lógica.

5.7.- ESTRATEGIA GLOBAL DE OPTIMIZACIÓN.

El objetivo de este apartado es unificar y presentar como una estrategia global las metodologías expuestas en este capítulo y en el anterior (capítulo 4). Expresando la estrategia de optimización en forma de pseudocódigo, se escribe para un esquemático dado, de coeficientes conocidos:

Metodología()

```
{
  int i = 0;
  Camino_Crítico = VERDAD;

  Mientras (Camino_Crítico){

    Camino_Crítico[i+1] = Identificar_Camino_Crítico();           (Capítulo 3)

    Técnicas_de_Buferización(Camino_Crítico[i+1]);             (Capítulo 4)
    O_ptimizar(Camino_Crítico[i+1]){                             (Capítulo 5)
      Optimizar(Camino_Crítico[i+1]);
      Si (Cambiar_Buffer) Técnicas_de_buferización(Camino_Crítico[i+1]);
      Sino Continuar_Metodología();

      Si (Buscar_Nuevo_Camino){
        Camino_Crítico = VERDAD;
        i++;
        Sino Salir();
      }
    }
  }
}
```

Se desprende de este algoritmo que la optimización sobre el camino crítico prosigue hasta no lograr mejoras sustanciales en el circuito. Se ha indicado entre paréntesis el capítulo donde se ha desarrollado la *función* que realiza la operación en cuestión.

CAPÍTULO 6

APLICACIÓN DE LA ESTRATEGIA DE OPTIMIZACIÓN A DISEÑOS EXPERIMENTALES DE REFERENCIA.

Se procede en este capítulo a la presentación de resultados. Aplicaremos la estrategia de optimización (técnicas de *buffering* y algoritmo de dimensionado) a tres circuitos seleccionados. En el primer circuito intentamos establecer una comparación entre las prestaciones del diseño robusto y el diseño optimizado. En el segundo estudiaremos la dependencia de la solución óptima con la señal de entrada. Por último en el tercer ejemplo optimizamos la ruta crítica de un sumador de 4 *bits*. El algoritmo de dimensionado nos permitirá obtener los diagramas de soluciones no inferiores. El análisis de estos diagramas nos permitirá rastrear todas las soluciones óptimas del espacio de diseño. La comparación de los resultados se hará con los valores retardo-potencia-área de la metodología de diseño robusto. Asimismo se confrontarán los resultados con los obtenidos vía simulación eléctrica (tiempos de computación serán igualmente comparados).

6.1.- INTRODUCCIÓN.

En los capítulos precedentes se han presentado los modelos desarrollados, las técnicas de "bufeización" y el algoritmo de dimensionado de transistores. Las técnicas de *buffering* junto al algoritmo de *sizing* conforman lo que hemos definido como estrategia de optimización.

En los circuitos MOS se entiende siempre por *camino crítico* de un circuito el que presenta un retardo mayor. El alto consumo de potencia que muestran los circuitos GaAs requiere considerar no sólo *caminos críticos* en cuanto al retardo de propagación, sino que han de considerarse también los caminos que contribuyen a la potencia (o/y al área ocupada) total del circuito con un consumo alto de potencia (o/y de área). En general, un circuito no puede optimizarse mediante la optimización de un sólo camino crítico. Para optimizar un circuito dado es necesario encontrar el camino crítico (bien sea en consumo de potencia, de retardo o de área) y una vez mejorado éste, se procede a la búsqueda del nuevo camino crítico y se mejora; este proceso continua hasta que ya no pueda mejorarse ningún camino sin degradar otro [HoNeS90].

Para los propósitos que perseguimos supondremos que conocemos de antemano el camino crítico del circuito a optimizar. Esta tarea puede realizarse aplicando un analizador temporal [Herná92]. Una vez optimizado este camino detendremos el proceso y centraremos la discusión en la interpretación de resultados.

La estrategia de optimización puede ser aplicada a cualquier diseño DCFL/SDCFL cuyos coeficientes retardo-potencia-área conozcamos. Tras el análisis de múltiples diseños y con el ánimo de mostrar las posibilidades que esta estrategia encierra hemos creído conveniente elegir tres circuitos y sobre éstos realizar los múltiples análisis que la estrategia brinda al diseñador.

El primer circuito (apartado 6.2) es simplemente un cadena de 12 inversores. El interés de este circuito estriba en que nos permitirá apreciar la sensibilidad del algoritmo de dimensionado ante la cargabilidad de las puertas. Para las capacidades debidas al interconexión se tomarán valores típicamente encontrados en los ficheros HSPICE. Los resultados se compararán con los del diseño implementado según estrategia robusta.

En el segundo ejemplo (apartado 6.3) aplicaremos la estrategia de optimización (*buffering* y *sizing*) a un circuito sencillo y obtendremos la curva retardo-potencia para el mismo. Además rastreamos la dependencia de la solución óptima con la pendiente de excitación.

Como tercer y último ejemplo hemos seleccionado el circuito correspondiente a la ruta crítica de un sumador CLA (*Carry Look-Ahead*) 4 bits. Este diseño CLA está disponible en versión *post-layout* lo cual ha posibilitado el conocimiento de las capacidades del interconexiónado (asignaremos a cada nodo de salida de las distintas puertas lógicas el valor medio del conjunto de capacidades). Una vez que conozcamos las soluciones no inferiores realizaremos la oportuna simulación con HSPICE para el nuevo dimensionado.

6.2.- DESCRIPCIÓN DEL CIRCUITO EJEMPLO N° 1.

Se presenta en la figura 6.1 el esquemático del circuito a optimizar. Estamos interesados en obtener el conjunto de soluciones no inferiores para el compromiso retardo-potencia.

La ruta crítica de este circuito tiene 11 puertas lógicas (bien inversores o puertas NOR equivalentes). Se ha añadido una puerta adicional que actúa de carga. Los nodos de salida de las puertas están cargadas con puertas inversoras y puertas NOR de dos entradas dando lugar a diferentes valores para el *fan-out*. La descripción del circuito se completa con los siguientes datos:

- C_L capacidad asociada al interconexiónado de valor 10 femtofaradios para todas los condensadores,
- predimensionado de transistores (pertenecientes al camino crítico):
 - 16 micras para los inversores,
 - 14 micras para las puertas NOR,
- dimensionado de transistores (transistores que no pertenecen al camino crítico):
 - 16 micras para los inversores,
 - 14 micras para las puertas NOR,
 - 12 micras para la puerta de carga,
- geometría de los buffers 9.2/6 para todas las puertas lógicas.

En las puertas NOR pertenecientes al camino crítico sólo uno de los transistores de la etapa lógica conmuta. Esta condición especial supondrá la sustitución de cada puerta NOR por el correspondiente inversor equivalente (que es justamente el inversor que tiene un ancho de puerta de 14 micras).

Los valores de las capacidades, como se indicó anteriormente (apartado 6.1), se corresponden con los típicamente encontrados en los ficheros HSPICE generados por el

extractor de capacidades.

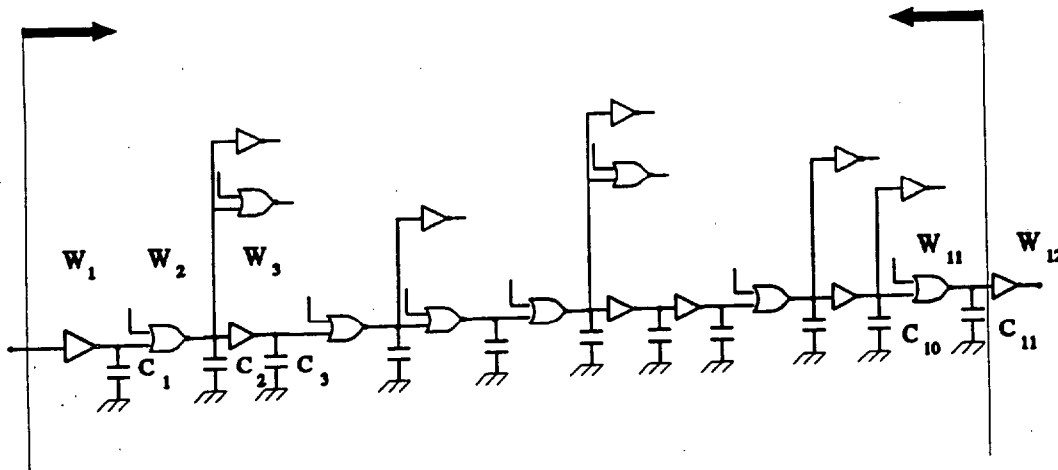


Fig. 6.1. Ejemplo de circuito a optimizar.

6.2.1.- Estimación del retardo, potencia y área para la implementación según metodología de diseño robusta.

Se menciona en el capítulo 4 (apartado 4.3.2) que la metodología de diseño robusta se caracteriza por el empleo de un sólo tipo de buffer (9.2/6) y unas pocas geometrías para los transistores que componen la etapa lógica de las puertas.

Dado el circuito (figura 6.1), se procede a la estimación del retardo de propagación, de la potencia consumida y del área ocupada mediante las funciones estimativas desarrolladas en el capítulo 3. La descripción del circuito se corresponde con la definida anteriormente salvo que:

- se dimensionan todos los transistores que pertenecen al camino crítico a 16 micras para los inversores y a 14 micras para las puertas NOR (estos valores son usualmente empleados por los diseñadores).

Se apunta el hecho de que los diseñadores hacen uso de estas dimensiones de forma casi *intuitiva*. La estrategia de optimización permite demostrar que el dimensionado mencionado, sin ser el óptimo, da lugar a un compromiso retardo-potencia-área que permite

el diseño robusto. Empleando este conjunto binario de dimensiones pueden implementarse diseños seguros y en poco tiempo.

Se realizarán dos evaluaciones para el retardo de propagación. La primera de ellas para una pendiente de entrada de 200 picosegundos. En la segunda estimación atacaremos el circuito con una pendiente de 400 picosegundos. Estos valores no se han escogido arbitrariamente sino que pueden encontrarse en un circuito DCFL/SDCFL en condiciones normales de funcionamiento. Se muestran en la tabla 6.1 los resultados de la estimación para cada valor de pendiente de entrada.

Tabla 6.1. Estimación retardo-potencia-área (diseño robusto).

Pendiente (ps)	Retardo (ns)	Potencia (mW)	Área (mm ²)
$\tau_i = 200$	1.241	5.602	0.019096
$\tau_i = 400$	1.208	5.602	0.019096

Se aprecia cómo influye la pendiente en el retardo del circuito. Se produce una mejora en el retardo de aproximadamente 33 picosegundos si la pendiente de excitación ó ataque al circuito está *adaptada*.

El análisis de los resultados nos lleva a afirmar que el hecho de que el circuito aparentemente retarde menos para la pendiente de 400 ps que para la de 200 ps se debe a que el punto de medida del retardo en la pendiente ascendente se *desplaza acercándose* al punto de medida de la señal de salida, la cual experimenta un desplazamiento menor hacia la derecha en el tiempo.

Para circuitos que no *siguen* las pendientes de excitación, es decir aquellos que absorben la pendiente inicial en pocas etapas esto pasará siempre dado que el retardo (tal como usualmente se mide) es una magnitud relativa entre las señales entrada-salida. Sin embargo este escaso margen de mejora en el retardo de propagación conduce a la disminución de la frecuencia de operación (reloj) del circuito.

Por lo tanto siempre ha de emplearse la pendiente menor posible (aquí 200 picosegundos o una menor). No obstante el límite inferior para la pendiente de entrada será aquel que su empleo causa una disminución de la frecuencia de operación del circuito. Esta disminución viene dada principalmente por los acoples capacitivos que generan tanto más *overshooting* cuanto más rápida es la pendiente.

Aunque la fiabilidad de los modelos estimativos empleados fue demostrada en el capítulo 3 (apartados 3.1, 3.2 y 3.3), se cree conveniente realizar una comparación frente a HSPICE para este caso, dado que se ha elegido como referencia. Para el circuito excitado con la señal de 200 picosegundos, los datos medidos con HSPICE han sido los siguientes:

- retardo de propagación: 1.350 ns y,
- consumo de potencia: 5.6991 mW,

lo cual supone un error del 8.07 % para el retardo y del 1.68 % para el consumo de potencia. Ambos valores se encuentra por debajo de la cota de fiabilidad, que hemos fijado en una desviación máxima permisible del 10%. El tiempo de cómputo empleado por HSPICE fue de 52.90 segundos en una máquina SUN SPARC2.

6.2.2.- Aplicación del algoritmo de dimensionado.

A continuación aplicamos el algoritmo de dimensionado (*sizing*) al circuito de la figura 6.1. El *buffer* que se emplea es el 9.2/6. Fijaremos los límites para el dimensionado de los transistores en un valor mínimo de 12 micras y un valor máximo de 30 micras (los cuales garantizan el correcto funcionamiento del circuito).

El conjunto de dimensiones posibles es pequeño a diferencia del caso CMOS (donde el margen de ruido -para la mayoría de los circuitos- no requiere un tratamiento especial) Dichas cotas limitan el tamaño de los transistores que pertenecen a los inversores así como a los que integran la puerta NOR. El hecho de que el conjunto de dimensiones posibles sea reducido redundará sobremanera en los porcentajes de mejora esperados.

La dimensión de cada transistor proporcionada por el programa de dimensionado para el retardo mínimo (consumo de potencia máximo) se corresponden con los siguientes valores:

$$\begin{aligned}
 w[1] &= 12.000000 \text{ (micras)} & w[2] &= 12.000000 \text{ (micras)} \\
 w[3] &= 12.000000 \text{ (micras)} & w[4] &= 12.000000 \text{ (micras)} \\
 w[5] &= 12.000000 \text{ (micras)} & w[6] &= 12.000000 \text{ (micras)} \\
 w[7] &= 12.000000 \text{ (micras)} & w[8] &= 12.000000 \text{ (micras)} \\
 w[9] &= 12.000000 \text{ (micras)} & w[10] &= 12.000000 \text{ (micras)} \\
 w[11] &= 12.000000 \text{ (micras)}
 \end{aligned}$$

$$\text{retardo} = 1.015373 \text{ (ns)} \quad \text{potencia} = 5.774030 \text{ (mW)} \quad \text{área} = 0.018392 \text{ (mm}^2\text{)}.$$

El truncamiento de las anteriores dimensiones se discute más adelante.

La dimensión de cada transistor proporcionada por el programa de dimensionado para el retardo máximo (consumo de potencia mínimo) se corresponden con los siguientes valores:

$$\begin{aligned} w[1] &= 30.000000 \text{ (micras)} & w[2] &= 27.481753 \text{ (micras)} \\ w[3] &= 30.000000 \text{ (micras)} & w[4] &= 27.481753 \text{ (micras)} \\ w[5] &= 30.000000 \text{ (micras)} & w[6] &= 27.481753 \text{ (micras)} \\ w[7] &= 30.000000 \text{ (micras)} & w[8] &= 27.481753 \text{ (micras)} \\ w[9] &= 27.481753 \text{ (micras)} & w[10] &= 30.000000 \text{ (micras)} \\ w[11] &= 30.000000 \text{ (micras)} \end{aligned}$$

$$\text{retardo} = 2.049358 \text{ (ns)} \quad \text{potencia} = 5.230024 \text{ (mW)} \quad \text{área} = 0.022471 \text{ (mm}^2\text{)}.$$

El número de análisis efectuados fue de 20. El tiempo de cómputo no superó los dos segundos (en una máquina SUN SPARC2). A partir de estos resultados podemos observar que:

- existen grandes diferencias en las dimensiones de los transistores según sean los valores de los coeficientes de peso,
- el retardo mínimo tiene lugar para el dimensionado mínimo de los transistores y,
- el consumo de potencia es mínimo para la configuración que tiene las dimensiones máximas en casi todas las puertas.

El primer aspecto resulta de especial interés dado que pone de manifiesto el amplio *juego* que ofrece la optimización. Para este circuito analizado -empleando sólo la técnica de dimensionado- se consigue una mejora en el retardo de hasta 1.03 nanosegundos y de aproximadamente 550 μW para el consumo de potencia.

El segundo aspecto tiene una sencilla explicación: el incremento en la dimensión del transistor se traduce directamente en una aumento de la carga soportada por el nodo anterior, por lo que al disminuir su tamaño el circuito se acelera. Así y a diferencia del caso CMOS, la mejora en el retardo de la puerta i lleva pareja una ralentización de la puerta $i-1$ (y anteriores).

El consumo de potencia ha sido ampliamente discutido en el capítulo 3 (apartado 3.2)

Destaca el hecho de que el consumo mínimo no tenga lugar para el dimensionado máximo de todas las puertas. Esta situación aparece en aquellos circuitos que tienen algunos nodos que soportan cargas muy diferentes (valor en femtofaradios) de otros nodos. La razón viene dada por el balance entre los consumos P_{OL} y P_{OH} (éste último fuertemente dependiente de la carga asociada al nodo de salida).

Puesto que la variación permisible en las dimensiones de los transistores es pequeña, el cambio en el área resulta ser despreciable. Podremos optimizar con respecto al área ocupada una vez se combinen las técnicas de *sizing* con las de *buffering*.

Se observa que las dimensiones de los transistores vienen expresadas en números en como flotante. Para su empleo en el diseño tienen que ser convenientemente truncadas al número más cercano posible que sea compatible con la resolución del proceso y las herramientas de edición y chequeo de reglas de diseño del entorno que se utilice. Por ejemplo si consideramos, por sencillez, que la resolución es una micra, truncaremos a valores enteros.

Definimos truncado según criterio *cercano-mayor* a aquella estrategia que consiste en tomar para cada dimensionado real (flotante), el entero inmediatamente superior. De forma similar definimos truncado según criterio *cercano-menor* a la estrategia que consiste en tomar para cada dimensionado real el entero inmediatamente inferior.

Se aconseja truncar según el criterio de dimensionado cercano-mayor si lo que se pretende es que el diseño real presente un consumo de potencia menor (y consecuentemente más lento). El criterio de dimensionado cercano-menor se aconseja en aquellos diseños en los que ha de primar la velocidad de conmutación frente al consumo de potencia.

6.2.3.- Diagrama de soluciones no inferiores para el compromiso retardo-potencia.

Se presenta en la figura 6.2 el diagrama de soluciones no inferiores para el circuito que se está estudiando. La pendiente de excitación es de 200 ps y el coeficiente $\delta_A = 0$. Esto significa que no se ha impuesto ninguna restricción al área. Se recuerda que los coeficientes δ_D , δ_P y δ_A son los que *ponderan* las diferentes funciones estimativas retardo, potencia y área respectivamente.

Para un mejor manejo de estos coeficientes los renombramos como sigue:

- δ_D *alpha*,

- δ_p beta y δ_A gamma.

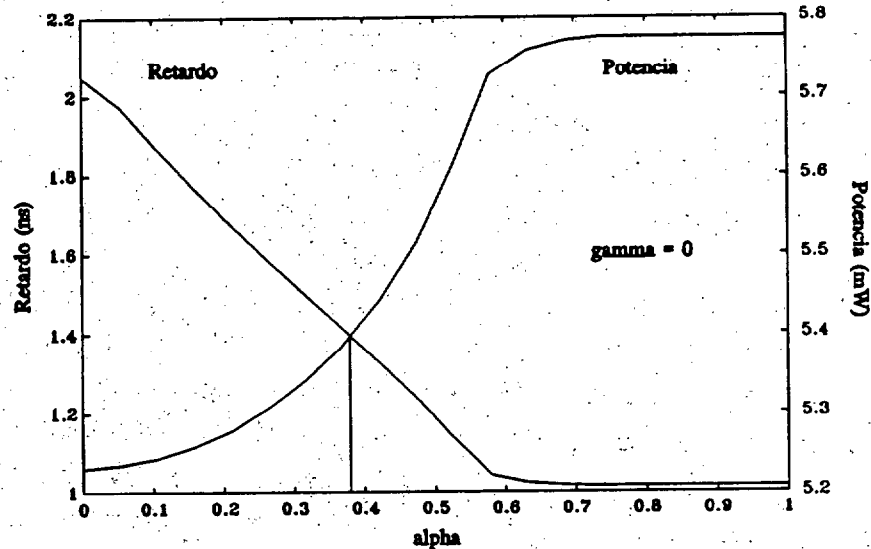


Fig. 6.2. Soluciones no inferiores para el compromiso retardo-potencia ($\tau_i = 200$ ps).

La solución óptima para el compromiso retardo-potencia viene dada por el punto de corte de ambas gráficas. Una vez conocido el valor de δ_D (*alpha*) para el que se cortan las gráficas se obtiene el conjunto de dimensiones óptimas. Dicha solución óptima representa el mejor compromiso posible entre ambas funciones estimativas.

A partir de este gráfico el diseñador dispone del conjunto completo de soluciones de diseño no inferiores para el circuito en estudio. De entre todas esas soluciones el diseñador puede optar por la solución óptima o bien por cualquier otra que satisfaga sus necesidades.

Es evidente que intentar encontrar la solución óptima mediante simulación HSPICE (variando arbitrariamente las dimensiones de los transistores) resulta prácticamente imposible.

Así, dado un diseño podemos optar por disminuir el consumo de potencia (a costa de ralentizar el circuito) sin más que elegir un valor de δ_D situado a la izquierda del punto de corte, o por acelerar el circuito (aumentando el consumo de potencia) eligiendo valores a la derecha de dicho punto de corte.

En este caso el valor del coeficiente δ_D para el que se logra el mejor compromiso retardo-potencia es de 0.37. El conjunto de dimensiones óptimas para la pendiente de excitación de 200 picosegundos se corresponde con el siguiente:

$$\alpha = 0.368421 \quad \beta = 0.631579 \quad \gamma = 0.000000$$

$$w[1] = 22.0450 \text{ (micras)} \quad w[2] = 18.3929 \text{ (micras)}$$

$$w[3] = 18.6886 \text{ (micras)} \quad w[4] = 18.7157 \text{ (micras)}$$

$$w[5] = 19.6601 \text{ (micras)} \quad w[6] = 18.7247 \text{ (micras)}$$

$$w[7] = 18.7452 \text{ (micras)} \quad w[8] = 18.5229 \text{ (micras)}$$

$$w[9] = 18.3314 \text{ (micras)} \quad w[10] = 19.7842 \text{ (micras)}$$

$$w[11] = 19.5722 \text{ (micras)}$$

$$\text{retardo} = 1.411864 \text{ (ns)} \quad \text{potencia} = 5.385542 \text{ (mW)} \quad \text{área} = 0.020134 \text{ (mm}^2\text{)}$$

La comparación de estos datos frente a los valores mínimos y máximos puede consultarse en la tabla 6.2. La importancia de la comparación con los valores máximos-mínimos radica en que nos suministra información acerca de lo *lejos* o *cerca* que se encuentra nuestra solución óptima de las soluciones que ofrecen los valores extremos. Dicha gráfica aporta indirectamente información acerca de lo amplio que es el conjunto de soluciones no inferiores.

Tabla 6.2. Comparación frente a los valores mínimos y máximos.

Retardo de propagación	Potencia consumida	Área ocupada
+ 39.04 % (retardo mín.)	+ 3.54 % (potencia mín.)	+ 9.47 % (área mín.)
- 31.10 % (retardo máx.)	- 9.72 % (potencia máx.)	- 10.40 % (área máx.)

La comparación de resultados frente a los obtenidos según la estrategia de diseño robusto se presenta en la tabla 6.3.

Tabla 6.3. Comparación de resultados con los obtenidos según diseño robusto.

Retardo de propagación	Consumo de potencia	Área ocupada
+ 4.58 %	- 3.87 %	+ 5.43 %

Nótese que el retardo mínimo empeora con respecto al obtenido mediante estrategia de diseño robusta en un 4.58 %. Este valor se alcanza con una disminución en el consumo de potencia del 3.87 %. El consumo de área ha aumentado como consecuencia de que no se impuso ninguna cota a su valor ($\gamma = 0$).

- C_L (C_1, C_2, C_3, C_4) capacidad asociada al interconexión de valor 5 femtofaradios para todos los condensadores,
- predimensionado de transistores (pertenecientes al camino crítico):
 - 16 micras para los inversores,
 - 14 micras para las puertas NOR,
- dimensionado de transistores (transistores que no pertenecen al camino crítico):
 - 16 micras para los inversores,
 - 14 micras para las puertas NOR,
 - 12 micras para la puerta de carga.

Aplicaremos la estrategia de optimización (*sizing + buffering*) a este circuito y rastreamos la dependencia de las soluciones no inferiores con la pendiente de excitación.

6.3.1.- Aplicación del algoritmo de dimensionado.

La formulación del problema se completa con los siguientes datos:

- pendiente de entrada de 100 picosegundos,
- dimensión mínima del ancho de puerta de 12 micras,
- dimensión máxima del ancho de puerta de 30 micras y,
- una única geometría para el *buffer* (9.2/6 en todas las puertas).

Se ha acordado un valor de 5 femtofaradios para todas las capacidades excepto para la de la puerta que tiene un *fan-out* de 1 (a la que se le asignó un valor de 10 femtofaradios). Se observa en la figura 6.4 el conjunto de soluciones no inferiores para el compromiso retardo-potencia.

Es importante observar el gran número de soluciones no inferiores distintas que se aprecia en esta figura. Veremos más adelante (figuras 6.6 y 6.8) que no sucede lo mismo para otros diseños. En dicha ilustración se indica el valor adoptado para el coeficiente *gamma*. El mejor compromiso entre ambas funciones se alcanza para un valor de $\delta_D \approx 0.43$. El conjunto de dimensiones óptimas para la pendiente de excitación de 100 picosegundos se corresponde con el siguiente:

$$\alpha = 0.421053 \quad \beta = 0.578947 \quad \gamma = 0.000000$$

$$w[1] = 20.734563 \text{ (micras)} \quad w[2] = 16.978731 \text{ (micras)}$$

$$w[3] = 16.644212 \text{ (micras)} \quad w[4] = 18.170918 \text{ (micras)}$$

$$\text{retardo} = 0.507599 \text{ (ns)} \quad \text{potencia} = 1.982808 \text{ (mW)} \quad \text{área} = 0.007228 \text{ (mm}^2\text{)}$$

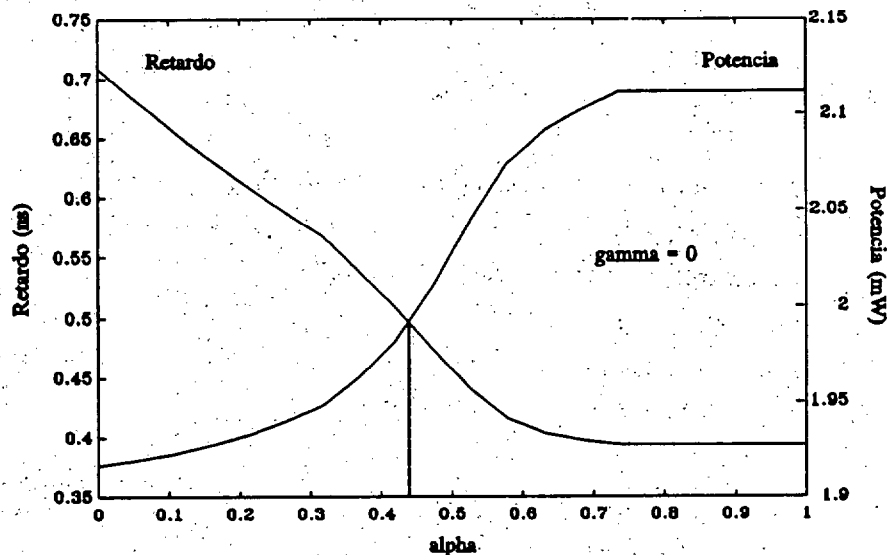


Fig. 6.4. Soluciones no inferiores para el compromiso retardo-potencia ($\tau_i = 100 \text{ ps}$).

Tabla 6.4. Comparación respecto de los valores mínimos y máximos.

Retardo de propagación	Potencia consumida	Área ocupada
+ 10.96 % (retardo mín.)	+ 4.75 % (potencia mín.)	+ 4.35 % (área mín.)
- 19.61 % (retardo máx.)	- 4.11 % (potencia máx.)	- 6.55 % (área máx.)

Nótese el cambio apreciable entre las dimensiones de uno y de otro transistor. Estos resultados obtenidos hasta el momento nos permiten afirmar que el ancho de puerta más grande lo tiene siempre el primer transistor (para los diseños implementados con el buffer 9.2/6). La explicación de esto es sencilla: el aumento de la dimensión de puerta de dicho transistor se traduce en una disminución en el retardo sin aumento apreciable del consumo de potencia.

Esta afirmación no se cumple para los transistores que no sean principio de una cadena puesto que al aumentar su ancho de puerta aumentan la carga del nodo precedente

(ver apartado 6.2.2).

Resulta de alto interés el hecho de que a partir de unos pocos resultados puedan derivarse reglas de diseño. Se muestra en la tabla 6.4 la comparación de resultados con los valores máximos-mínimos de las funciones.

6.3.2.- Aplicación de las técnicas de "bufferización".

Seguidamente pasamos a reemplazar alguno de los *buffers* por otro de geometría distinta. Dado que estamos interesado en reducir el consumo de potencia seleccionamos una geometría para el nuevo *buffer* de forma que reduzca dicho consumo.

De entre los posibles seguidores de fuente hemos optado por el de geometría 6/4. Este *buffer* permite reducir el consumo de potencia y el área ocupada de forma apreciable.

La formulación del problema es idéntica a la anterior con la salvedad de que los *buffers* se han dispuesto como sigue:

- *buffer* 9.2/6 en el nodo que tiene un *fan-out* de 3,
- *buffer* 6/4 en el resto de los nodos.

Se observa en la figura 6.5 el conjunto de soluciones no inferiores para el compromiso retardo-potencia.

La comparación entre este diagrama y el anterior (figura 6.4) nos indica que el número de soluciones no inferiores diferentes se ha visto reducido. Se observan asimismo ligeras fluctuaciones en las funciones estimativas las cuales son consecuencia del aumento de la no linealidad del problema planteado. Esta no linealidad se manifiesta en forma de *ruido* numérico.

El mejor compromiso entre ambas funciones se alcanza para un valor de $\delta_D \approx 0.18$. El conjunto de dimensiones óptimas para la pendiente de excitación de 100 picosegundos se corresponde con el siguiente:

$$\alpha = 0.187895 \quad \beta = 0.812105 \quad \gamma = 0.000000$$

$$w[1] = 21.318786 \text{ (micras)} \quad w[2] = 26.794931 \text{ (micras)}$$

$$w[3] = 17.287438 \text{ (micras)} \quad w[4] = 17.155370 \text{ (micras)}$$

$$\text{retardo} = 0.645022 \text{ (ns)} \quad \text{potencia} = 1.700093 \text{ (mW)} \quad \text{área} = 0.007105 \text{ (mm}^2\text{)}.$$

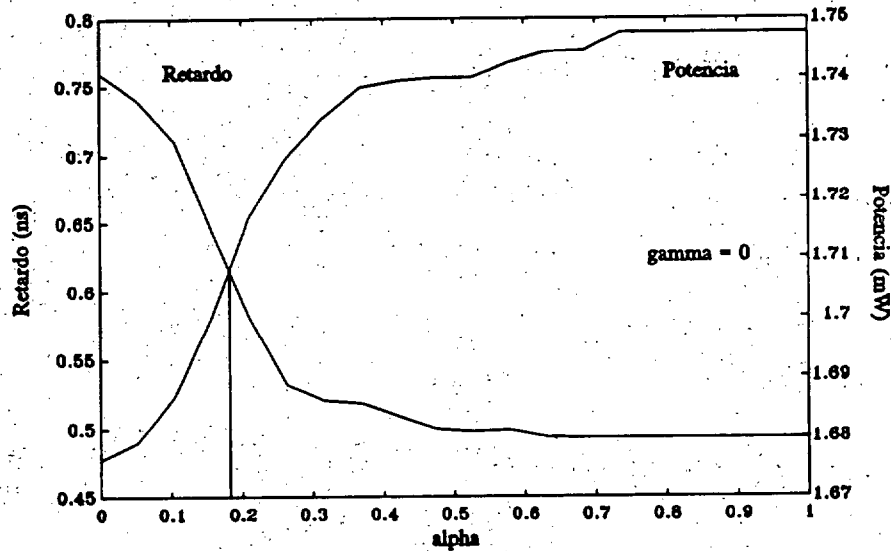


Fig. 6.5. Soluciones no inferiores para el compromiso retardo-potencia ($\tau_i = 100 \text{ ps}$).

A diferencia del caso anterior, el transistor con una dimensión de puerta mayor no aparece al principio de la cadena. Esto significa que el balance entre las funciones P_{OH} (potencia en estado lógico alto a la salida de la puerta) y P_{OL} (potencia en el estado lógico bajo a la salida de la puerta) se suaviza con la presencia de otro *buffer*.

La comparación de resultados con los valores máximos-mínimos de las funciones se muestra en la tabla 6.5

Tabla 6.5. Comparación respecto de los valores mínimos y máximos.

Retardo de propagación	Potencia consumida	Área ocupada
+ 30.8 % (retardo mín.)	+ 1.41 % (potencia mín.)	+ 6.84 % (área mín.)
- 15.09 % (retardo máx.)	- 2.72 % (potencia máx.)	- 6.31 % (área máx.)

En la tabla 6.6 presentamos la comparación entre los resultados con la intención de

apreciar el alcance de la estrategia de optimización sobre este circuito.

Tabla 6.6. Comparación de resultados tras la aplicación de la estrategia de optimización.

Optimización aplicada	Retardo de propagación	Potencia consumida	Área ocupada
<i>sizing</i>	0.507599 ns	1.982808 mW	0.072 (mm ²)
<i>sizing + buffering</i>	0.645022 ns	1.700093 mW	0.0071 (mm ²)
diferencia relativa	+ 27 %	- 14.25 %	- 1.7 %

El consumo de potencia ha disminuido de forma considerable. Obsérvese la dificultad que encierra el *arañar* unos pocos microwatios (o picosegundos o micrómetros). Como se indicó anteriormente esto es consecuencia del reducido número de dimensiones posibles que se manejan.

6.3.3.- Estudio de la dependencia de la solución óptima con la pendiente de entrada al circuito.

Para hacer este estudio haremos uso del circuito (y de los resultados) anterior (anteriores). Emplearemos una pendiente de excitación de 400 picosegundos.

El conjunto de soluciones no inferiores para la relación retardo-potencia se presenta en la figura 6.6. La solución óptima se alcanza para el valor $\delta_D = 0.24$. La estimación de retardo, potencia y área para ese valor de *alpha* es la siguiente:

$$\text{retardo} = 0.491311 \text{ (ns)} \quad \text{potencia} = 1.733439 \text{ (mW)} \quad \text{área} = 0.006743 \text{ (mm}^2\text{)}.$$

En la tabla 6.7 se puede apreciar la influencia que la pendiente tiene sobre las prestaciones del circuito. La comparación se hace con los datos del ejemplo anterior.

A tenor de estos porcentajes parecería indicado excitar al circuito con una señal lenta en vez de con una señal rápida. Se comentó anteriormente (apartado 6.2.1) que de obrar de

esa forma sólo lograríamos disminuir la frecuencia de operación (aspecto para nada deseado).

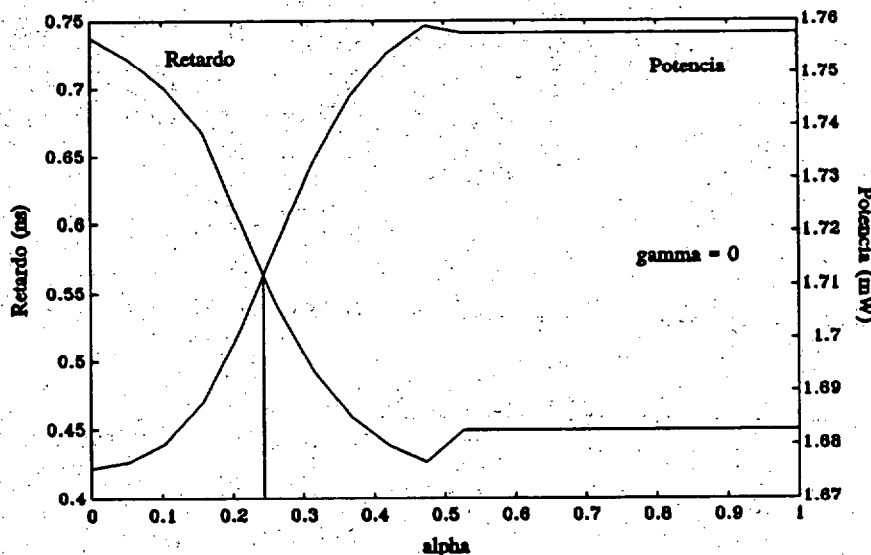


Fig. 6.6. Soluciones no inferiores para el compromiso retardo-potencia ($\tau_r = 400$ ps).

Tabla 6.7. Comparación respecto al caso anterior.

Retardo de propagación	Potencia consumida	Área ocupada
- 28.20 %	+ 1.96 %	- 5.09 %

El interés de este resultado viene dado por la razón de que en un diseño dado, las señales que recorren un camino tienen un determinado valor -pendiente estable- y en base a éste se puede implementar la optimización con respecto a la señal (el diseñador no tiene prácticamente posibilidad de cambiar el valor de dicha pendiente).

Se ha explorado la dependencia de la solución óptima con la pendiente y tras los análisis efectuados se concluye que el efecto de la pendiente sobre el dimensionado desaparece para rutas con más de 10 puertas (número fuertemente dependiente de la carga presente en dicho camino).

Por consiguiente se aconseja realizar este tipo de estudio sólo en caminos cortos (obsérvese que el tiempo de cómputo para un camino de estas características es inferior a 1 segundo).

6.4.- DESCRIPCIÓN DEL CIRCUITO EJEMPLO N° 3.

Se ha elegido como tercer y último ejemplo el diseño correspondiente a un sumador CLA (*Carry Look-Ahead*) de cuatro *bits*. El esquemático del circuito puede verse en la figura 6.7. La ruta crítica para este circuito *atraviesa* 6 puertas NOR y ha sido determinada mediante nuestro analizador temporal [Herná92]. Dicha ruta es el camino del *carry* que comienza en el *bit* b_0 y finaliza en el nodo que tiene la etiqueta C_{out} .

6.4.1.- Descripción del circuito y aplicación de la estrategia de optimización.

En este esquemático pueden apreciarse algunas de las tendencias de diseño en lógica DCFL/SDCFL. Así se observa que el *fan-out* nunca es superior a 2. Las puertas que se emplean se reducen a puertas NOR y OR de 2 entradas e inversores.

Una vez reducido el circuito a los correspondientes inversores equivalentes, completamos la formulación del problema como sigue:

- pendiente de entrada de 200 picosegundos,
- dimensión mínima del ancho de puerta de 12 micras y,
- dimensión máxima del ancho de puerta de 30 micras.

Se dispondrá una puerta de carga (de dimensión 16 micras) en el nodo de salida del *carry*. Para optimización *pre-layout* daremos inicialmente un valor de 15 femtofaradios a todas las capacidades de interconexiones locales. Para optimización *post-layout* utilizaremos las capacidades extraídas.

Formulado el problema pasamos a optimizar el camino crítico. Tal como se hizo anteriormente, primero aplicaremos el algoritmo de dimensionado (manteniendo los *buffers* fijos) para posteriormente aplicar las técnicas de "bufferización".

6.4.2.- Aplicación del algoritmo de dimensionado.

El programa implementado nos permite obtener el conjunto de soluciones no inferiores para el compromiso de las tres funciones estimativas (ver figura 6.8). Para la realización de dicha figura hemos fijado el valor del coeficiente *gamma* (asociado al área estimada) a un valor $\delta_A = 0.12$.

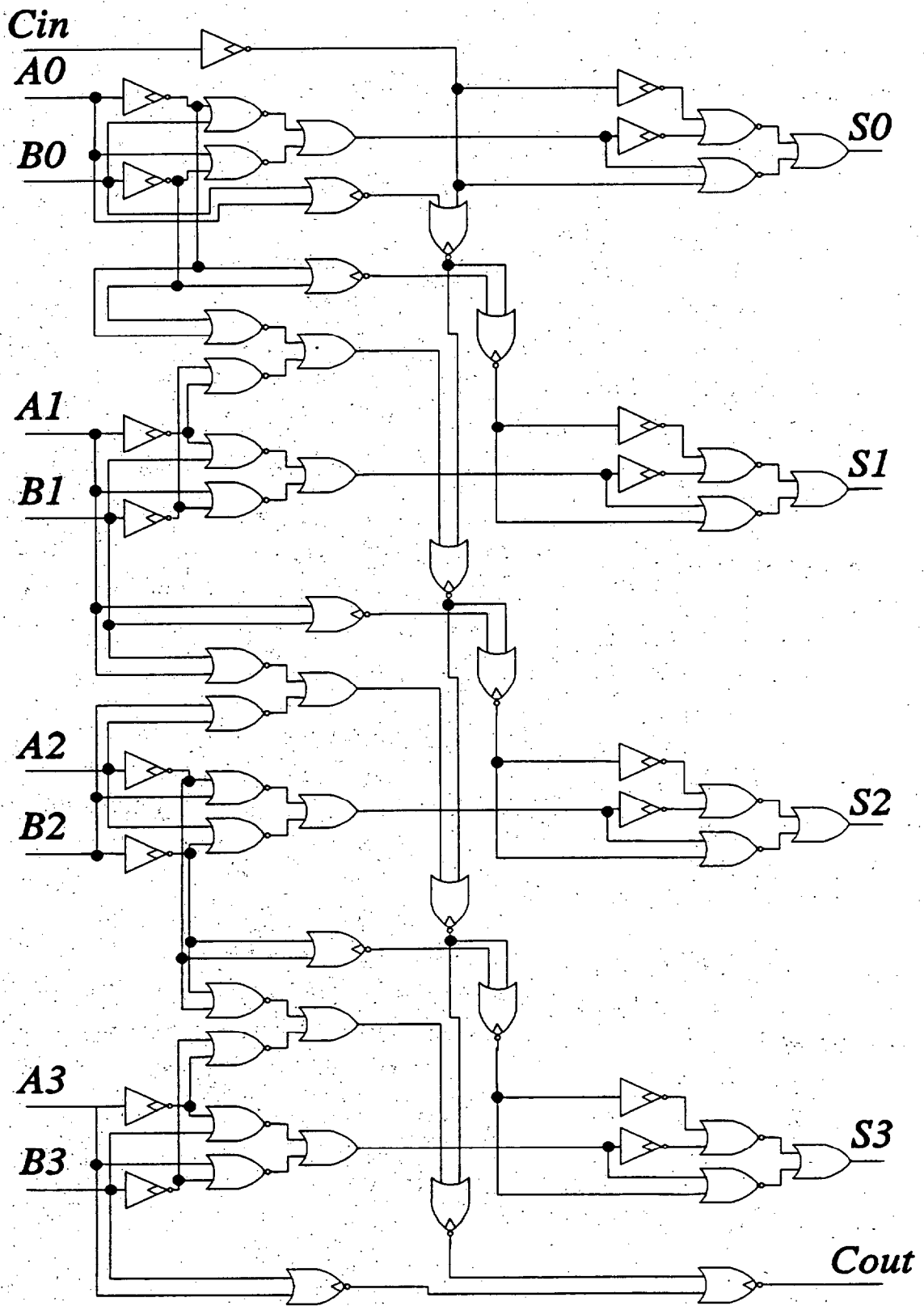


Fig. 6.7. Esquema lógico correspondiente al sumador Carry Look-Ahead de 4 bits.

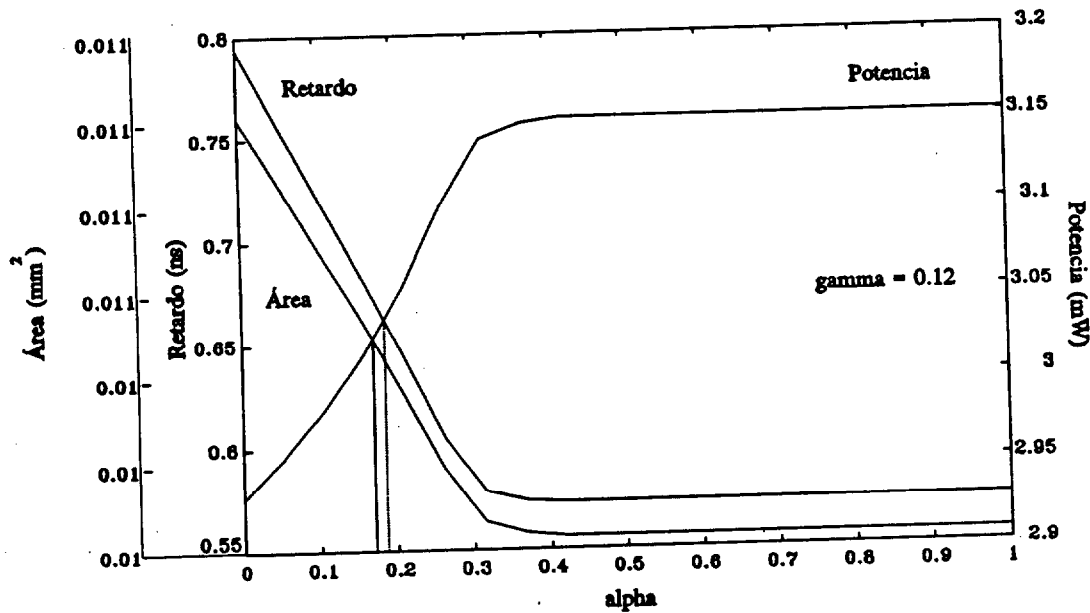


Fig. 6.8. Soluciones no inferiores para el compromiso retardo-potencia-área ($\tau_i = 200$ ps).

Se observa que el conjunto de soluciones no inferiores para la función retardo presenta el mismo perfil que el correspondiente al área ocupada (salvo factor de escala). Nótese que para el valor de δ_A seleccionado no existe una solución óptima para el compromiso de las tres funciones.

Este hecho no resta importancia a los datos que puedan derivarse a partir de esta figura. La información contenida en dicha gráfica permite al diseñador seleccionar el mejor compromiso para la pareja retardo-potencia o para la pareja potencia-área. Debido a la proximidad de ambos puntos, podemos considerar que las gráficas se cortan en un sólo punto; elegiremos el valor de $\delta_D = 0.18$.

Finalizada la tarea de dimensionar los transistores procedemos a cambiar los *buffers* presentes por uno de geometría 7/4 en todos los nodos excepto en las dos últimas puertas. El *buffer* de dimensiones 9.2/6 es más apto para regenerar una señal determinada que el seguidor de fuente 7/4; es por esto por lo que se emplea en las últimas puertas (con una importante mejora en el retardo de propagación).

La aplicación de las técnicas de *buffering* nos permite comparar los resultados para las dos implementaciones (ver tabla 6.8).

El consumo de potencia ha vuelto a disminuir de forma apreciable (como era de esperar). Además hemos logrado disminuir el consumo de área (en un valor también

importante) es decir, hemos conseguido disminuir el área a la vez que reducimos el consumo de potencia. Destacamos este último aspecto porque es sabido que en lógica SDCFL al disminuir el área el diseño experimenta un incremento en el consumo de potencia, por el balance de corriente que ocurre en el *buffer* (ver apartado 3.2.3).

Tabla 6.8. Comparación de resultados tras la aplicación de la estrategia de optimización.

Optimización aplicada	Retardo de propagación	Potencia consumida	Área ocupada
<i>sizing</i>	0.68266 ns	3.013946 mW	0.0105 (mm ²)
<i>sizing + buffering</i>	0.69469 ns	2.663691 mW	0.0099 (mm ²)
diferencia relativa	+ 1.76 %	- 11.62 %	- 5.68 %

Hemos creído oportuno realizar la simulación mediante HSPICE para el camino optimizado. Para el circuito que tiene una sola geometría para el *buffer* (9.2/6), hemos medido:

- retardo de propagación: 753.29 ps (error \approx - 9.37 %)
- potencia consumida: 3.0698 mW (error \approx - 1.85 %).

Para el diseño que tiene dos geometrías para el seguidor de fuente se obtiene:

- retardo de propagación: 768.12 ps (error \approx - 9.55 %)
- potencia consumida: 2.70159 mW (error \approx - 1.40 %).

Como se aprecia, el error cometido es menor del 10 % para ambos casos, como era de esperar dado que esta cota de error viene impuesta por el analizador temporal usado en la estimación [Herná92].

De los análisis efectuados se deduce que la estrategia de optimización está indicada para aquellos circuitos en los que se desee reducir el consumo de potencia (y/o de área). Para hacer esto se deben elegir los *buffers* clasificados dentro del grupo consumo moderado-bajo (apartado 4.4.3).

También es posible *acelerar* el circuito sin más que seleccionar los seguidores de fuente enmarcados en la categoría de consumo alto.

6.5.- CONCLUSIONES.

Se concluye este capítulo dedicado a la presentación de una selección demostrativa de los resultados de evaluación obtenidos tras la aplicación de la estrategia de optimización a diseños experimentales de referencia. Se califica esta estrategia como una herramienta capaz de suministrar al diseñador información de primera magnitud.

El método presentado minimiza la relación retardo-potencia-área en el camino crítico, permitiendo la variación simultánea de las dimensiones de todos los transistores de la etapa lógica pertenecientes a dicho camino.

Pueden imponerse distintas restricciones tanto al consumo de área como al consumo de potencia con lo cual se impide la acumulación excesiva de carga sobre puertas pertenecientes a caminos que no están siendo optimizados (o evitar excesos de potencia en determinadas zonas).

Los diagramas retardo-potencia-área constituyen auténticas radiografías del circuito a implementar. El diseñador puede, en base a estos diagramas, tomar decisiones a nivel *pre-layout* de diseño de forma rápida y con absoluta fiabilidad. Por ejemplo, el cálculo de todas las soluciones no inferiores para el camino crítico de un sumador de 4 bits tarda 2 segundos de CPU en una workstation SUN SPARC2. Una vez decidido y editado el *layout*, pueden hacerse ajustes finos optimizando el *netlist* extraído. De forma adicional, la estrategia puede ser utilizada de cara a poner a punto una familia lógica y una metodología de diseño que de ella se derive, puesto que permite explorar múltiples aspectos.

CONCLUSIONES Y LÍNEAS ABIERTAS.

7.1.- CONCLUSIONES.

Este trabajo ha puesto de manifiesto que es posible mejorar las prestaciones de un diseño DCFL mediante el empleo de una técnica de "buferrización" eficaz. Dicha técnica se ha de complementar con un dimensionado óptimo para los transistores que integran el diseño. Se demuestra que la combinación de ambas técnicas en la metodología propuesta es viable y es eficaz, proporcionando márgenes de optimización significativos.

Un importante paso en el diseño de un circuito GaAs digital es la optimización del consumo de potencia mientras se mantiene el consumo de área y la velocidad del circuito dentro de unos límites razonables. La metodología expuesta en esta tesis no sólo es idónea para la consecución de dicho objetivo sino que además, permite descartar soluciones de poco interés de cara a la implementación de un diseño DCFL con nuevas opciones de cargabilidad a la entrada o salida de las puertas.

Hemos definido una técnica de *buffer* sistemático que denominamos metodología robusta. Se demuestra que con ella pueden diseñarse circuitos DCFL de forma rápida, con absoluta fiabilidad y con buenas prestaciones.

El empleo de esta metodología nace como una necesidad de cara a poder construir diseños complejos obviando el problema asociado al deterioro de los márgenes de ruido. De esta forma queda afianzado el diseño según la metodología robusta. Para esta metodología se han definido las geometrías óptimas a emplear tanto para el seguidor de fuente como para los transistores de las distintas etapas lógicas.

Para la tecnología utilizada (GaAs E/D MESFET de $0.8\mu\text{m}$) se ha justificado el empleo del *buffer* de geometría 9.2/6 para aquellas aplicaciones en las que acortar el tiempo de diseño sea prioritario. En aquellos diseños robustos en los que se desee sólo una optimización inicial, bastará hacer una optimización con *buffer* fijo excepto en los caminos críticos y nodos muy cargados por *fan-out* e interconexiones para los que se debe hacer una optimización con *buffer* adaptativo.

Se ha desarrollado un estudio de las posibilidades de mejorar las prestaciones que ofrece la lógica DCFL. En esta línea, se exploraron los tipos de geometrías, pendientes de excitación, cargabilidad de las puertas y los niveles de tensión a emplear. Los análisis se realizaron en algunas situaciones empleando los modelos desarrollados para tal fin, y en otras haciendo uso del simulador HSPICE.

La interpretación de los resultados manifiesta que el diseño robusto tiene limitada la velocidad de conmutación. Esto no es consecuencia de que se haya impuesto una cota al consumo de potencia o al de área. Por ello se ha desarrollado la metodología de diseño adaptativo de cara a proporcionar una mayor flexibilidad a los diseños.

El interés de esta metodología adaptativa viene dado porque permite la implementación de diseños más rápidos (consumiendo más potencia). Asimismo el empleo de la técnica adaptativa permite reducir el consumo de potencia (ralentizando el circuito). El diseño adaptativo brinda al diseñador la posibilidad de *adaptar* las características del circuito a los requerimientos deseados (entiéndase como deseados posibles), moviéndose en un espacio de diseño más amplio que ofrece mayores márgenes a la optimización.

La diferencia radical entre ambas estrategias se traduce en comodidad y rapidez en el diseño. Así, un diseño de tipo adaptativo requerirá una realización más cuidadosa demandando un mayor tiempo de desarrollo. Dado que la celeridad en la terminación de un circuito GaAs constituye en gran número de casos un factor de primer orden, no extraña que el diseñador se decante por el diseño robusto, al menos para las primeras versiones o series de circuitos. Sin embargo, el diseño adaptativo goza siempre de una mejora sustancial en sus prestaciones y a él debe tenderse cuanto antes en el ciclo de vida de un producto. Por esta razón es necesario automatizar las técnicas de "bufferización" y dimensionado adaptativo proporcionando una ayuda directa al diseñador.

Para ello, una vez definido el conjunto de *buffers*, se ha procedido a desarrollar un programa de optimización. Ésta herramienta suministra al diseñador información de primera magnitud. El método expuesto lleva a cabo la minimización (de forma analítica) de la relación retardo- potencia-área en el camino crítico, permitiendo la variación simultánea de las dimensiones de todos los transistores de la etapa lógica pertenecientes a dicho camino.

El algoritmo de optimización está guiado por funciones de coste y curvas retardo- potencia-área que constituyen auténticas radiografías del circuito a implementar. El diseñador puede, en base a estos diagramas, tomar de forma interactiva o automática decisiones de diseño a nivel *pre-layout*. La herramienta se encarga de optimizar el circuito de forma muy

rápida. Una vez decidido y editado un *layout*, pueden hacerse ajustes finos optimizando el *netlist* extraído.

Los programas de optimización desarrollados contienen aproximadamente 2000 líneas de código en lenguaje C, y hacen uso de numerosas tablas con coeficientes de precaracterización y modelos. El código generado no tiene el nivel de herramienta comercial sino de prototipo, y deja espacio para su posterior mejora.

El empleo de una optimización de tipo analítico hace posible obtener unas conclusiones a las que no es posible llegar mediante algoritmos de tipo heurístico. En base a esto, el diseñador dispone del conjunto completo de posibles soluciones ajustadas al criterio de optimización elegido. En esta tesis se ha demostrado que dado un diseño es posible reducir apreciablemente el consumo de potencia mediante la aplicación de la estrategia de optimización propuesta. El grado de optimización alcanzable en tiempo de propagación y en área es igualmente significativo, aunque menos notable. Los resultados revelan que la aplicación de la metodología de optimización, lleva la tecnología empleada (DCFL con E/D MESFET de $0.8\mu\text{m}$) *prácticamente* hasta sus límites en cuanto a velocidad se refiere (para las condiciones normales de operación).

Puesto que la metodología presentada es tecnológicamente independiente, es de esperar que los resultados a obtener para una tecnología de distintas características presenten una similar o mayor bondad en cuanto a la mejora de las prestaciones se refiere.

7.2.- LÍNEAS FUTURAS.

El tema abordado en esta tesis deja abiertos vario temas de investigación en varios aspectos diferentes. La labor desarrollada hasta el momento por nuestro grupo de investigación en sistemas digitales en Arseniuro de Galio nos ha llevado a concluir que el esfuerzo investigador debe hacerse con mayor urgencia en varios aspectos.

En este sentido el trabajo de investigación presentado en esta tesis puede continuarse en las siguientes líneas:

a) Mejora de los modelos estimativos y extensión a circuitos secuenciales. Aplicación a otras familias lógicas en GaAs.

Este proyecto de investigación se encuentra en fase de desarrollo. Los nuevos modelos han

de incorporar la consideración de los siguientes aspectos:

- la inclusión de corrientes parásitas y efectos de capacidades parásitas,
- el manejo de interconexiones de mayor longitud (efectos de *cross-talk*, degradación de señales...)
- la dependencia del consumo de potencia y del área-*chip* con el estilo de diseño,
- la dependencia de la potencia consumida con la temperatura,
- adaptación a circuitos secuenciales.

La metodología presentada es capaz de integrar estos nuevos aspectos. El estudio de estos aspectos hará posible explorar nuevas soluciones de diseño.

Mención especial merece la aplicación de la metodología a otras familias lógicas en GaAs. Esta es una línea que suscita un interés especial en la comunidad científica actual. Dado que dichas familias lógicas probablemente empleen dispositivos cuánticos (HEMTs ó HBTs), será preciso abordar su estudio y desarrollar las correspondientes estrategias de optimización.

Dado que estos dispositivos operan a frecuencias muy altas, el tratamiento especial del ruido así como de las interconexiones se vuelve prioritario. Existe en el grupo una línea de investigación centrada en dicha problemática.

b) Integración de la estrategia de optimización en un entorno de diseño estándar.

Esta línea de investigación se revela como el aspecto inmediato a desarrollar en el grupo para explotar la herramienta en la producción de diseños. La incorporación de la estrategia en un entorno de diseño estándar (CADENCE) está siendo actualmente acometida. De hecho ya se ha modificado el extractor de CADENCE para reconocer dispositivos de la lógica DCFL. La sustitución del extractor PHASE2 por PD extract de CADENCE, facilita el acceso al conjunto de herramientas de CADENCE, como el compilador de estructuras ó el compactador de *layout*.

La integración de los programas de optimización en este entorno facilitará grandemente el desarrollo de macrocélulas, módulos y bloques *full-custom* optimizados, para su uso en la síntesis de circuitos complejos.

c) Ayuda para los niveles altos de concepción y síntesis de arquitecturas.

A más largo plazo, pueden aplicarse los modelos de estimación y la estrategia de optimización propuestos para estudiar la incidencia que una determinada arquitectura tiene sobre las prestaciones de un sistema complejo (procesadores RISCs, sistemas de procesamiento digital de señales...). Las alternativas arquitecturales y lógicas a emplear, según sea la tecnología para la implementación de los distintos elementos (unidad aritmética-lógica, registros...) podrán ser analizadas, y podrán realizarse de forma asistida tomas de decisiones delicadas antes de acometer con todo detalle (esfuerzo y costo) el diseño físico ó *layout* definitivo.

REFERENCIAS

- [AgLeR77] B. J. Agule, J. D. Lesser, A. E. Ruehli y P. K. Wolff, "An Experimental System for Power/Timing Optimization of LSI Chips," *Proc. of the 14th Design Automation Conf.*, 1977.
- [Akers82] S. B. Akers, "Clustering Techniques for VLSI", in *Proc. IEEE International Symposium on Circuits and Systems*, pág. 472-476, 1982.
- [AlZha89] A. Al-Khalili, Y. Zhu y D. Al-Khalili, "A Module Generator for Optimized CMOS Buffers," *Proc. 66th ACM/IEEE Design Automation Conference*, pág. 245-249, 1989.
- [Annar86] M. Annaratone, *Digital CMOS Circuit Design*. Capítulos 2 y 3. Kluwer Academic, Boston, 1986.
- [AuAzD91] D. Auvergne, N. Azemard, D. Deschacht y M. Robert, "An Accurate and Efficient Delay Time Modelling and its Application to CMOS Data Path Evaluation and Transistor Sizing," *Proc. of the 13th IMACS, World Congress on Computation and Applied Mathematics*, Julio 1991.
- [BalSá91] J. Ballesteros y J. Sánchez, "Técnicas de Optimización topológica para reducir el área de Arrays Lógicos Programables," *Actas del VI Congreso de Diseño de Circuitos Integrados*, Santander, pág.485-490, Noviembre. 1991.
- [Boyer88] D. Boyer, "Symbolic Layout Compaction Review," *Proc. of the 25th Design Automation Conference*, pág. 383-389, Junio 1988.
- [Brand87] D. Brand, "Logic Synthesis," *Design Systems for VLSI Circuits: Logic Synthesis and Silicon Compilation*. Martinus Nijhoff, pág. 301.-326, 1987.
- [BrDH79] R. K. Brayton, S. W. Director, G. H. Hachtel y L. M. Vidigal, "A New Algorithm for Statistical Circuit Design Based on Quasi-Newton Methods and Function Splitting," *IEEE Trans. on Circuits and Systems*, vol. CAS-26, pág. 784-794, Septiembre. 1979.
- [KerPi87] B. Kernighan y R. Pike, El Entorno de Programación UNIX. Capítulos 5, 6, 7, 8. Prentice-Hall, Inc. Englewood Cliffs, México 1987.
- [BrHaM84] R. Brayton, G. Hachtel, C. McMullen y A. Sangiovanni-Vicentelli, Logic Minimization Algorithms for VLSI Synthesis. Kluwer Academic Publishers, 1984.
- [Bryan80] R. Bryant, "An algorithm for MOS Logic simulation," *LAMBDA magazine*, Vol. 1, núm. 3, pág. 46-53, 1980.
- [Bryan86] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. on Computer*, Agosto 1986.

- [Burst85] M. Burstein y M. Youssef, "Timing Influenced Layout Design," *Proc. of the 22nd Design Automation Conference*, pág. 124-130, 1985.
- [Cates90] Ron Cates, "Gallium Arsenide finds a new niche," *IEEE Spectrum*, pág. 25-28, Abril 1990.
- [Ciri87] M. A. Cirit, "Transistor Sizing in CMOS Circuit," *Proc. of the 24th ACM/IEEE Design Automation Conference*, pág. 121-124, 1987.
- [CoBaA91] J. Cortadella, R. Badía y E. Ayguadé, "Scheduling in a Continuous Area-Time Design Space," in *Proc. Microprocessing and Microprogramming*, nº 32, pág. 199-206, 1991.
- [Curti80] W. R. Curtice, "A MESFET Model for Use in the design of GaAs Integrated Circuits," *IEEE Trans. Microwave Theory and Tech.*, vol. MTT-28, no. 5, pág. 448-456, Mayo 1980.
- [Curti88] W. R. Curtice, "GaAs MESFET Modeling and Nonlinear CAD," *IEEE Trans. Microwave Theory and Tech.*, vol. 36, no. 2, pág. 220-230, Febrero 1988.
- [ChaCh89] M. Chan y C. F. Chen, "PROMPT3: A Cell Based Transistor Sizing Program Using Heuristic and Simulated Annealing," *Proceedings of the IEEE 1989 Custom Integrated Circuits Conference*. 1989.
- [ChaHa83] V. Chankong y. Y. Haimes, Multiobjective Decision Making: Theory and Methodology. North-Holland Series in System Science and Engineering, Elsevier Science, 1983, vol. 8. New York.
- [CheKa91] H. Y. Chen y S. M. Yang, "A New Circuit Optimization Technique for High Performances CMOS Circuits," *IEEE Trans. Computer-Aided Design*, vol. 10, no. 5, pág. 670-676, Mayo. 1991.
- [CHoBa90] S. Chowdhury y J. S. Barkatullah, "Estimation of Maximum Currents in MOS IC Logic Circuits," *IEEE Trans. Microwave theory and Tech.* MTTT-28, no. 5, pág. 448-456, Mayo 1980.
- [ChuHo87] C. Chu y M. Horowitz, "Charge-sharing models for switch-level Simulation," *IEEE Trans. Computer-Aided Design*, Vol. 6, pág. 1053-1060, 1987.
- [ChuLi75] L. O. Chua y N. N. Wang, "A New Approach to Overcome the Overflow Problem in Computer-Aided Analysis of Nonlinear Resistive Circuits," *International Journal of Circuit Theory and Applications*, vol. 3, pág. 261-282, 1975.
- [DahBj74] G. Dahlquist y A. Björck, Numerical Methods, Prentice Hall, 1974.
- [DaLuR83] R. Da Luz Reis, "Evalueur Topologique Predictif pour la Generation Automatique des Plans de Masse de Circuits VLSI", tesis, L'Institut National

- Polytechnique de Grenoble, 1983.
- [Dawbr89] J. Dawbrowski, "Mixed-Mode Timing Verification for VLSI Designs," *Proc. of Int. AMSE Conf.*, MSE Press vol. 4, Brighton 1989.
- [DeKrW90] S. Devadas, K. Kreutzer y J. White, "Estimation of Power Dissipation in CMOS Combinatorial Circuits," *IEEE Custom Integrated Conference*, pág. 1971-1976, 1990.
- [DenSh90] A. C. Deng y C. Shiau, "Generic linear RC delay modeling for digital CMOS circuits," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 4, Abril 1990.
- [DePiR90] D. Deschacht, P. Pinede, M. Robert y D. Auvergne, "PATH RUNNER: An accurate and fast timing analyser," *Proc of EDAC90*, Glasgow, 12-15 Marzo 1990.
- [DeRoA88] D. Deschacht, M. Robert y D. Auvergne, "Explicit formulation of delays on CMOS data path," *IEEE Jour. of Solid State Circuits*, vol. 23, pág. 1257-1264, Octubre 1988.
- [DiKaL91] Carlos H. Diaz, Sung-Mo Kang y Yusuf Leblebici, "An Accurate Analytical Delay Model for BiCMOS Driver Circuits," *IEEE Trans. on Computer-Aided Design*, vol. 10, no. 5, pág. 577-588, Mayo 1991.
- [DiLoP77] M. Díaz, E. Lora-Tamayo y L. Pun, Introducción al Diseño Asistido por Ordenador de Circuitos Electrónicos. Cap. 3, 4. Alhambra, Madrid, 1977.
- [DunAg84] A. E. Dunlop y V. D. Agrawal, "Chip Layout Optimization Using Critical Path Weighting," *Proc. of the 21st Design Automation Conference*, pág. 133-136, 1984.
- [EisSc91] V. Eisele y D. Schmitt-Landsiedel, "Optimal Delay-Power Tradeoffs for Replicative Logic Circuitry," *Proc. Int. Symposium on Circuits and Systems (ISCAS)*, pág. 2264-2267, 1991.
- [EisSC91] V. Eisele y D. Schmitt-Landsiedel, "Optimization and Architectural Evaluation of Regular Combinatoric Logic Structures," *Proc. of the Euromicro Conf.91*, 1991.
- [Eshra90] K. Eshraghiam, Gallium Arsenide Integrated Circuit Design. *Notas de curso*. Capítulo 2. ETSIT, Las Palmas de Gran Canaria, 1990.
- [FisDu85] J. P. Fishburn y A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proc. 1985 IEEE Conf. Computer-Aided Design*, pág. 326-328, 1985.
- [FujKu72] T. Fujisawa y E. S. Kuh, "Piecewise Linear Theory of Nonlinear Networks,

- * *SIAM J. Appl. Math.*, vol. 22, pág. 307-328, Marzo 1972.
- [Gasc91] D. M. Gasca González, Cálculo Numérico I. Temas 20 y 23. Unidades didácticas, UNED, 6ª ed. Madrid 1991.
- [GasSa75] Gass y Saul, Linear Programming, 4th ed., McGraw-Hill, New York, 1975.
- [GeuCo85] A. J. de Geus y W. Cohen, "A rule-based System for Optimising combinational Logic," *IEEE Design and Test of Computers*, 2(4), Agosto 1985.
- [GilMu72] P. E. Gill y W. Murray, "Quasi-Newton Methods for Unconstrained Optimization," *J. Inst. Math. Appl*, vol. 9, pág. 91-108, 1972.
- [GoHeN92] L. Gómez, A. Hernández y A. Núñez, "Timing Model for SDCFL Digital Circuits," *Microprocessing and Microprogramming*, núm. 34, pág. 193-196, Febrero 1992.
- [Hayes82] J. P. Hayes, "A Unified Switching Theory for VLSI," *Proc. IEEE*, pág. 1140-1141, Oct. 1982.
- [HedJe87] N. Hedenstierna y K. O. Jeppson, "CMOS circuit speed and buffer optimization," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pág. 270-281, Marzo 1987.
- [Hedlu87] K. S. Hedlund, "AESOP: A tool for automated transistor sizing," in *Proc. 24th Design Automation Conf.*, pág. 114-120, 1987.
- [HeGoN90] A. Hernández, L. Gómez y A. Núñez, *PATMOS First Periodic Progress Report*, CAML-3, ESPRIT/BRA 3237, Appendix 1, 1990.
- [HerFi89] A. Herrigel y W. Fichtner, "An Analytic Optimization Technique for Placement of Macro Cells," in *Proc. 26th Design Automation Conf.*, pág. 376-381, 1989.
- [Herná92] A. Hernández, "Modelado del tiempo de propagación para análisis y verificación temporal de circuitos DCFL en GaAs," Tesis Doctoral en preparación. ETSIT, Universidad de Las Palmas de Gran Canaria, 1992.
- [HILO88] System HILO User's Manual. GenRad Incorporated, 1988.
- [Himme72] D. M. Himmelblau, Applied Non Linear Programming. Capítulo 4, McGraw-Hill, 1972.
- [Hitch82] R. B. Hitchcock, "Timing Verification and The Timing Analysis Program," *Proc. IEEE/ACM 19th Design Automation Conf.*, pág. 594-604, 1982.
- [HoCa074] S. J. Hong, R. G. Cain y D.L. Ostapko, "MINI: A Heuristic Approach for

- Logic Minimization," *IBM J. of Res. and Dev.*, vol. 18, pág.443-458, Septiembre 1974.
- [HofKi87] M. Hoffmann y J. K. Kim: "Delay optimization of combinational static CMOS logic", *Proceedings of 24th ACM/IEEE DAC87*, pág. 125-131, 1987.
- [Hofma86] M. Hofmann, "A Table-Driven Approach to Timing Analysis," *VLSI Technical Bulletin*, vol. 1, no. 2, pág. 16-18, Septiembre 1986.
- [HoNeS90] B. Hoppe, G. Neuendorf, D. Schmitt-Landsiedel y W. Specks, "Optimization of high-speed CMOS logic circuits with analytical models for signal delay, chip area, and dynamic power dissipation," *IEEE Trans. Computer-Aided Design*, vol.9,no. 3, pág.236-247, Marzo. 1990.
- [Hoppe88] B. Hoppe, "Circuit Optimization: Gate Level Modeling and Multiobjective programming," *Microprocessing and Microprogramming*, vol. 25, pág. 171-176, 1988.
- [Horow83] M. A. Horowitz, "Timing Models for MOS Pass Networks," *Proc. of IEEE ISCAS'83*, pág. 198-201, Newport Beach, Mayo 1983.
- [Horow84] M. A. Horowitz, "Timing models for MOS circuits," Ph.D. dissertation, Stanford Univ., Enero. 1984.
- [Hoyte82] L. P. J. Hoyte, "Automated Calculation of Devices Sizes for Digital IC Designs," Tesis doctoral, MIT, 1982.
- [HSPIC90] HSPICE User's Manual. H9001. Meta-Software, Inc. California, 1990.
- [HwaWu91] J. S. Hwang y. Y. Wu, "Efficient Techniques in the Sizing and Constrained Optimization of CMOS Combinational Logic Circuits, *IEEE Proc. on Computers and Digital Techniques*, vol. 138, no. 3, Mayo 1991.
- [HyShP86] C. H. Hyun, M. S. Shur, y A. Peczalski, "Analysis of Noise Margin and Speed of GaAs MESFET DCFL Using UM-SPICE," *IEEE Tran. on Electron Dev.*, vol. ED-33, no. 10, pág. 1421-1426, Octubre 1986.
- [IsaKe66] E. Issacson y H. B. Keller, Analysis of Numerical Methods. John Wiley, 1966.
- [JacKu89] M. Jackson y E. Kuh, "Performance-Driven Placement of Cells Based IC's," in Proc. 26th Design Automation Conf., pág. 370-375, 1989.
- [Joupp83] N. P. Jouppi, "TV: An NMOS Timing Analyzer," *Proceedings of the Third Caltech Conference on VLSI*, pág. 71-85, 1983.
- [JuJuP89] Y. H. Jun, K. Jun y S. B. Park, "An accurate and efficient delay time modeling for MOS logic circuits using polynomial approximation," *IEEE*

- Trans. Computer-Aided Design*, vol. 8, no. 9, pág. 1027-1032, Sept. 1989.
- [JunJu88] Y. H. Jun y K. Jun, "An accurate and efficient delay simulator for MOS logic circuits using polynomial approximation," *Proc. IEEE ISCAS'88*, vol. 3, 1988.
- [Kanum81] A. Kanuma, "CMOS Circuit Optimization", *Solid-State Electron*, vol.26, pág. 47-58, 1981.
- [KerRi87] B. W. Kernighan y D. M. Ritchie, El Lenguaje de Programación C. Prentice-Hall, Inc. Englewood Cliffs, México 1987.
- [LasWa78] L. S. Lasdon y A.D. Warren, Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems, in H.G. Green berg (ed.), Design and Implementation of Optimization Software, Sijthoff and Noordhoff, Alphen and den Rijn, The Hetherlands, 1978.
- [LeeSo84] C. M. Lee y H. Soukup, "An algorithm for CMOS timing and area optimization," *Jour. of Solid-State Circuits*, vol. sc-19, núm. 5, Octubre 1984.
- [Lewi84] E. T. Lewis, "Optimization of Device Area and Overall Delay for CMOS VLSI designs," *Proc. IEEE on* , vol. 72, pág. 670-689, Junio 1984.
- [LiaWo83] Y. Z. Liao y C. K. Wong, "An Algorithm to Compact a VLSI Symbolic Layout with Mixed Constraints," *Proc. of the 20th Design Automation Conf.*, pág. 107-112, Junio 1983.
- [LinLi75] H. C. Lin y L. W. Linholm, "Optimized Output Stage for MOS Integrated Circuits," *IEEE Journal. Solid-State Circuits*, Abril 1975.
- [LinMe84] T.-M. Lin y C. A. Mead, "Signal delay in general RC networks," *IEEE Trans. Computer-Aided Design*, Vol. CAD-3, pág. 331-349, 1984.
- [LonBu90] S. I. Long y S. E. Butner, Gallium Arsenide Digital Integrated Circuits Design, McGraw-Hill, New York, 1990.
- [LonSun88] S. I. Long y M. Sundaram, "Noise-Margin limitations on Gallium-Arsenide VLSI," *IEEE J. Solid-State Circuits*, vol. 23, no. 4, pág. 893-900, Agosto 1988.
- [Luenb72] D. G. Luenberger, "Introduction to Linear and Nonlinear Programming," Addison-Wesley, Agosto 1972.
- [MaBrN88] A. Malik, R. Brayton, A. Newton y A. Sangiovanni-Vicentelli, "A Modified Approach To Two-Level Minimization," in *Proc. of the International Conference on Computer-Aided Design*, 1988.
- [MaFoB84] L. Markov, J. Fox y J. Blar, "Optimization Techniques for Two-Dimensional

- Placement," en *Design Automation Conference*, pág. 652-654, 1984.
- [MAPLE88] Reference Manual, Fifth Edition. Bruce W. Char, Keith O. Geddes and Gaston H. Gonnet. Symbolic Computation Group. Department of Computer Science, University of Waterloo, Waterloo, Ontario.
- [MarGa86] D. Marple y A. El Gamal, "Optimal Selection of Transistors Sizes in Digital VLSI Circuits," *Actas de la Stanford Conference on VLSI*, pág. 151-172, Marzo 1987.
- [MarGa86] Marple, D. y A. El Gamal, "Area-Delay Optimization of Programmable Logic Arrays", *4th MIT conf. on VLSI* (1986), 171-194.
- [Marl78] W. H. Marlow, Mathematics for Operations Research. John Wiley & Sons, New York, pág. 281-284, 1978.
- [Marpl86] D. P. Marple, "Performance Optimization of Digital VLSI Circuits," Ph. D. dissertation, Stanford Univ., Octubre 1986.
- [Marpl89] D. Marple, "Transistor Size Optimization in the Tailor Layout System," *Proc. 26th Design Automation Conf.*, pág. 43-48, 1989.
- [Mata90] A. Mata, Turbo C. Iniciación y Programación Avanzada. Paraninfo, Madrid 1990.
- [MatGl86] M. D. Matson y L. A. Glasser, "Macromodeling and optimization of digital MOS VLSI circuits," *IEEE Trans. on Computer-Aided Design*, vol. CAD-5, núm. 4, pág. 659-677, Octubre 1986.
- [MATLA85] Manual del usuario. *The MathWorks, Inc*, 1985.
- [MeaCo80] C. A. Mead, L. A. Conway, Introduction to VLSI Systems, Addison-Wesley Pub. Comp. 1980.
- [MeDaJ91] E. Melcher, M. Dana y F. Jutand, "ISTIANA: Input Slope dependent Timing Analysis by Abstraction," *PATMOS Second Periodic Progress Report*, CAML-1, ESPRIT/BRA 3237, Appendix 1, 1991.
- [Meije90] P. B. L. Meijer, "Fast and Smooth Highly Nonlinear Multidimensional Tables Models for Device Modeling," *IEEE Trans. on Circuits and Systems*, Vol. 37, núm. 3, Marzo 1990.
- [Meyer82] G. J. Meyers, Advances in Computer Architecture. John Wiley & Sons, 1982.
- [MiBrS85] G. De Michelli, R. K. Brayton y A. Sangiovanni-Vicentelli, "Optimal State Assignment of Finite State Machines," *IEEE Trans. on CAD*, pág. 269-285, Julio 1985.

- [MosMe79] A. Moshen y C. Mead, "Delay-Time Optimization for Driving and Sensing of Signals on High-Capacitance Paths of VLSI Systems," *Journal of Solid State Circuits*, pág. 540-548, 1979.
- [MuQuF91] J. Mulder, N. Quach y M. Flynn, "An Area Model for On-Chip Memories and its Application," *IEEE J. Solid-State Circuits*, vol. 26, SC-26, pág.98-106, Feb. 1991.
- [Nagel75] L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," *Electronics Research Laboratory*, University of California, Berkeley, artículo ERL-M520, 1975.
- [NavRo91] D. Navarro y A. Roy, "Un Compilador de Buffers para Diseños Optimizados," *Actas del VI Congreso de Diseño de Circuitos Integrados*, pág.367-374, Noviembre. 1991.
- [Newto79] A. R. Newton, "The Simulation of Large-Scale Integrated Circuits," *IEEE Trans. on Circuits and Systems*, vol. CAS-26, pág. 741-749, Septiembre 1979.
- [NhaBo80] H. N. Nham y A. K. Bose, "A Multiple Delay Simulator for MOS LSI Circuits," *Proceedings of the 17th Design Automation Conference*, pág. 610-617, Junio 1980.
- [Nye81] W. Nye, "DELIGHT: An Optimization-Based Computer-Aided Design System," *Proc. IEEE International Symp. on Circuits and Systems*, pág. 851-855, Abril, 1981.
- [ObeKa88] F. Obermeier y R. Katz, "An Electrical Optimizer that Considers Physical Layout," *Proc. IEEE of the 25th Design Automation Conference*, pág. 453-459, Junio 1988.
- [Ouste83] J. K. Ousterhout, "CRYSTAL: A Timing Analyzer for NMOS VLSI Circuits," *Proc. 3rd Caltech VLSI Conf.*, R. Bryant Ed., pág. 57-70, 1983.
- [Ouste85] J. K. Ousterhout, "A switch level timing verifier for digital MOS VLSI," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pág. 336-349, Julio 1985.
- [ParOs91] L. París y T. Osés, "Posicionado de Celdas Estándar en un Entorno de Floorplanning," *actas del VI Congreso de Diseño de Circuitos Integrados*, Santander, pág.321-326, Noviembre. 1991.
- [PenRu81] P. Penfield y J. Rubenstein, "Signal delay in RC tree networks," *IEEE Proc. 19th DAC*, pág. 613-617, 1981.
- [Perdo92] S. Perdomo, "Caracterización de Estructuras VLSI para Circuitos Integrados de Muy Alta Velocidad Mediante una Metodología Basada en la Mezcla de los Dominos de Frecuencia-Tiempo," Tesis doctoral en preparación. ETSIT,

- Universidad de Las Palmas de Gran Canaria, 1992.
- [PilRo90] L. T. Pillage y R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. on Computer-Aided Design*, vol. 9, núm. 4, pág. 352-366, Abril 1990.
- [Polak71] E. Polak, Computational Methods in Optimization: A Unified Approach. Academic Press, Inc. New York, 1971.
- [RaSaH79] N. B. G. Rabbat, A. L. Sangiovanni-Vicentelli y H. Y. Hsieh, "A Multi-level Newton Algorithm With Macromodelling and Latency for the Analysis of Large-Scale Nonlinear Circuits in the Time Domain," *IEEE Trans. on Circuits and Systems*, vol. CAS-26, pág. 733-741, Septiembre 1979.
- [RaTrH83] V. B. Rao, T. N. Trick y I. N. Hajj, "A Table-Driven Delay-Operator Approach to Timing Simulation of MOS VLSI Circuits," *IEEE Proc. International Conference on Computer-Aided Design*, pág. 445-448, New York, Noviembre 1983.
- [Rocch90] M. Rocchi, High-Speed Digital IC Technologies. Capítulos 3, 5, 7 Artech House Inc. Norwood 1990
- [RoDeP88] M. Robert, D. Deschächt, S. Pravossoudovtch, G. Cathebras, D. Auvergne, B. Dumotier y D. Tranvaux, "PRINT METHODOLOGY: A Compilation Approach for Cell Library Generation," *Proc. ISCAS*, pág. 965-968, 1988.
- [RoMeC92] W. Röthig, E. Melcher, S. Chakroun y M. Dana, "Power Consumption Estimation Using Statistical Signal Properties," *Microprocessing and Microprogramming*, no. 35, pág. 691-696, 1992.
- [RoTrCa91] M. Robert, J. Trauchessec, G. Cathebras, V. Bonzom, N. Azemard, D. Deschacht y D. Auvergne, "Evaluation of VLSI Layout Style Implementation for Efficiency," *Proc. Euro ASIC 91*, pág. 1-4, Mayo, 1991.
- [RuPeH83] J. Rubinstein, P. Pendfield, Jr. y M. A. Horowitz, "Signal Delay in RC Tree Networks," *IEEE Trans. on Computer Aided Design*, Vol. CAD-2, núm. 3, pág. 202-211, Julio 1983.
- [RuWoG77] A. E. Ruehli, P. K. Wolff y G. Goertzel, "Analytical Power/Timing Optimization Technique for Digital Systems," *Proc. of the 14th Design Automation Conf.*, pág. 142-146, 1977.
- [SaArC91] R. Sarmiento, V. Armas, P. P. Carballo, J. López y A. Núñez, "Diseño y Optimización de Sumadores en Tecnología GaAs," *VI Congreso de Diseño de Circuitos Integrados*, pág. 407-412, Noviembre. 1991.
- [SaMoC91] R. Sarmiento, J. A. Montiel, P. P. Carballo, J. López y A. Núñez, "Diseño y Optimización de Multiplicadores para Procesado Digital de Señales en

- Tecnología GaAs," *Actas del VI Congreso de Diseño de Circuitos Integrados*, pág. 413-418, Noviembre. 1991.
- [Sarmi91] R. Sarmiento, "Aportaciones al Diseño de Procesadores GaAs. Resultados de las Técnicas de Partición e Implementación en Función de los Parámetros Tecnológicos," Tesis Doctoral. ETSIT, Universidad de Las Palmas de Gran Canaria, 1992.
- [Sasao81] T. Sasao, "Multiple-Valued Decomposition of Generalized Boolean Functions and the Complexity of Programmable Logic Arrays," *IEEE Trans. on Comp.*, vol: C-30, No. 9, pág. 635-643, Septiembre, 1981.
- [SauTh85] G. Saucier y G. Thuau, "Systematic and Optimized Layout of MOS Cells," *Proc. 22nd Design Automation Conf.*, pág. 53-61, 1985.
- [Sclat88] N. Sclater, Gallium Arsenide IC Technology: Principles and Practice, Ed. Tab Books Inc., PA, 1988.
- [ShSaF88] J. Shyu, A. Sangiovanni-Vicentelli, J. P. Fishburn y A. E. Dunlop, "Optimization-Based Transistor Sizing," *IEEE J. Solid-State Circuits*, vol. 23, pág. 400-409, Abril 1988.
- [Shur87] M. S. Shur, GaAs Devices and Circuits. Capítulo 7. Plenum Press, New York, 1987.
- [StoBur80] J. Stoer t R. Bulirsch, Introduction to Numerical Analysis, Springer Verlag, 1980.
- [Subra84] P. Subramaniam, "Table Methods for Timing Simulation," *Conf. on Custom Integrated Circuits*, pág. 310-314, 1984.
- [Sze81] S. M. Sze, Physics of Semiconductor Devices. 2ª edición. Capítulos 1, 6 y 12. John Wiley & Sons. New York, 1981.
- [TaYoI82] T. Takada, K. Yokoyama, M. Ida, y T. Sudo, "A MESFET Variable-Capacitance Model for GaAs Integrated Circuit Simulation," *IEEE Trans. Microwave Theory and Tech.*, vol. MTT-28, no. 5, pág. 719-724, Mayo 1982.
- [Temes69] G. C. Temes, Optimization Methods in Circuit Design. Capítulo 5. Prentice-Hall, Inc., 1969.
- [Tjarn90] R. Tjörnström, "Switch-Level Simulation based on Local Decisions," *Integration the VLSI Journal*, Febrero/Marzo 1990.
- [ToOkS83] T. Tokuda, K. Okazaki, K. Sakashita, I. Ohkura y T. Enomoto, "Delay-Time Modeling for ED MOS Logic LSI," *IEEE Trans. on Computer-Aided Design*, vol. CAD-2, n. 3, pág. 129-134, Julio 1983.

- [Trim83] S. Trimberger, "Automated Performance Optimization of Custom Integrated Circuits," *Proc. Int. Symp. on Circuits and Systems*, pág. 194-197, 1983.
- [VlaSi83] J. Vlach y K. Singhal, Computer Methods for Circuit Analysis and Design, ed. Van Nostrand Reinhold, New York, 1983.
- [WesEsh85] N. Weste y K. Eshraghian, Principles of CMOS VLSI Design- A System Perspective. Addison Wesley, USA, 1985.
- [WilBe67] D. J. Wilde y C. S. Beightler, Foundations of Optimization. Englewood Cliffs, NJ, Prentice-Hall, 1967.
- [Willi84] R. E. Williams, Gallium Arsenide Processing Techniques. Artech House, Inc., Dedham, MA, 1984.
- [YuaSv88] J. Yuan y C. Svensson, "CMOS Circuit Speed Optimization Based on Switch Level Simulation," *Proc. of ISCAS'88*, pág. 2109-2112, 1988.
- [YuaSv89] J. Yuan y C. Svensson, "A Simulation-Based Fast Algorithm for CMOS Speed Optimization," *Proc. of ISCAS'89*, pág. 868-871, 1989.
- [YuWyZ85] Q. Yu, J. L. Wyatt, Jr., C. Zukowski, H. N. Tan y P. O'Brien, "Improved bounds on signal delay in linear RC models for MOS interconnect," in *Proc. IEEE ISCAS85*, pág. 903-906, Kyoto, Junio 1985.
- [Ziel63] A. van der Ziel, "Gate Noise in Field Effect Transistors at Moderately High Frequencies," *Proc. IEEE*, vol. 51, pág. 461-467, 1963.

