INTRODUCCION A LA TECNOLOGIA DEL ARSENIURO DE GA RESULTADOS EN EL DISEÑO DE SUMADORES Y MULTIPLICADORES DE ALTA VELOCIDAD.

V. de ARMAS, J. A. MONTIEL y J. Fco. LOPEZ
Centro de Microelectrónica Aplicada
Universidad de Las Palmas de Gran Canaria
Las Palmas de Gran Canaria
ESPAÑA

RESUMEN

Los avances en el desarrollo de circuitos integrados digitales en Arseniuros de Galio (GaAs) han progresado hasta tal punto que los diseñadores de procesadores de señales y datos pueden discernir el tipo de aplicación más conveniente para este tipo de tecnología. Dos de las primitivas básicas encontradas comunmente en los procesadores de datos y de señal son los sumadores y multiplicadores. En este artículo se enumeran las distintas familias lógicas empleadas con mayor frecuencia en GaAs, para posteriormente exponer el diseño *full-custom* en tecnología de 0.8 um. de distintos algoritmos y estilos de *layout* empleados a la hora de implementar estos dos tipos de primitivas, permitiéndole al diseñador un abanico de posibilidades a elegir según la aplicación a la que sean destinados.

ABSTRACT

Advances in the development of digital GaAs integrated circuits have progressed to the point that designers of signal and data processors can discern the system applications for which GaAs is best suited. Two of the most basic computation primitives used in DSP are adders and multipliers. In this paper different logic families frequently used in GaAs, are numbered, and full-custom designs in 0.8 um. GaAs technology usi different algorythms and layout styles wh implementing this two types of primitive allowing the designer a huge set of possibili ties when choosing an application.

1. INTRODUCCION.

El creciente aumento de la demanda d procesadores de señales e imágenes con baj consumo de potencia y altas prestaciones e velocidad, ha hecho que el diseño de est procesadores tengan una clara influenc dentro de la tecnología de *GaAs*. En es sentido, las limitaciones introducidas por tecnología, hace que el diseño de estas un dades, fuertemente fundamentadas en Silici tenga que ser reexaminado para sus impl mentaciones en *GaAs*. Aspectos tan estudi dos en Silicio, como pueden ser, la elecci de la arquitectura [GilPa87], o el algoritm utilizado, deben ser estudiados cuidados mente, al verse influenciados por restricci nes tales como, bajo *fan-in* y *fan-out*, limt ción del número de transistores por chip, aumento del coste de las puertas en *GaAs*, cual puede limitar su utilización en l diseños VLSI en esta tecnología.

Estas limitaciones favorecen unas impleme taciones de las unidades aritméticas, bi estructuradas, regulares y con unos flujos las líneas de datos y de control muy simpl

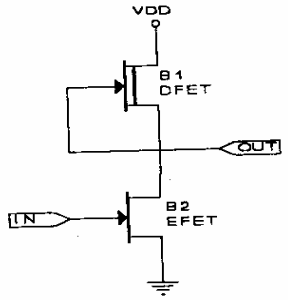
ste punto, podemos intentar realizar os algoritmos aritméticos, recursivos naturaleza, de forma interactiva, a base utilizar puertas. De esta forma, tanto la dología de diseño, como el tipo de *ut* utilizado, tienen una influencia funda- tal, que se manifiesta en problemas o *dispersión de las señales*, cuando te un retardo entre los tiempos de llegada una señal a dos puntos diferentes, y *sstalk*, que se manifiesta a través de ruido las líneas de señal.

este estudio se presentan diferentes im- mentaciones de sumadores y multiplicado- , al aparecer dichos bloque como primiti- s básicas en el procesado digital de señales SP), y en los sistemas de procesamiento imágenes. La implementación de estas nidades se ha realizado básicamente con la amilia lógica *SDCFL* (Lógica de Transisto- res Directamente Acoplados por Seguidor de Fuente), aportando los resultados de las simulaciones de estos módulos, realizados con diferentes metodologías de diseño.

2. DESCRIPCION DE LA TECNOLOGIA.

La familia lógica *DCFL* (Lógica de Transistores Directamente Acoplados), es la más compacta dentro de la tecnología de Arseniuro de Galio, así como la más adecuada para la realización de sistemas VLSI. Esta familia lógica se asemeja a la familia n-MOS en Silicio, utilizando un transistor de empobrecimiento (*D-MESFET*) como carga activa, y un transistor de enriquecimiento (*E-MESFET*) para implementar las funciones lógicas [LonBu90]. La puerta básica en esta familia, correspondiente a un inversor, es la que se refleja en la Ilustr. 1.

Como contrapartida, dentro de las limitaciones que introduce esta familia lógica, se destaca el bajo margen de ruido (< 100mV), debido a la barrera introducida por el diodo Schottky de la puerta del transistor de enriquecimiento. Otros problemas que plantea

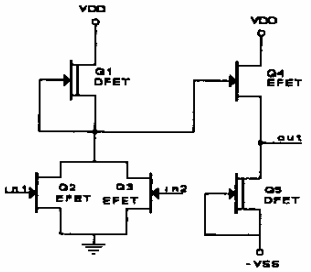


Ilustr. 1 Inversor DCFL.

son, la gran influencia del retardo de la puerta con el *fan-in* y *fan-out*, así como la dependencia del punto de conmutación del transistor *E-MESFET* con la temperatura, al tener un coeficiente negativo de variación con dicho factor.

El uso industrial de esta lógica, se basa en el desarrollo de unas técnicas de *bufferización* para mejorar las prestaciones de esta familia lógica. La solución adoptada en este estudio es la de utilizar un *buffer*, implementado por un seguidor de fuente conectado a la salida de la puerta lógica [Eshra90]. Al utilizar esta técnica de *bufferización* en todas las puertas, nos referiremos a esta lógica con la denominación *SDCFL* (Lógica de Transistores Directamente Acoplados por Seguidor de Fuente). La Ilustr. 2 muestra el diagrama de una puerta NOR *SDCFL* de 2 entradas.

La adición de un seguidor de fuente a la salida de la puerta *DCFL* mejora, entre otros aspectos, el margen de ruido, la capacidad de carga y el comportamiento frente a variaciones de temperatura. La conexión del transistor *E-MESFET* de la etapa de *buffer* a un tensión negativa $-V_{ss}$, aumenta el aislamiento entre la etapa lógica y la etapa de *buffer*. Al mismo tiempo, reestablece los niveles lógicos que atacan a la siguiente etapa, aumentando el margen de ruido. Cuando el transistor *E-MESFET* de la etapa de *buffer* está en *off*, el nivel lógico a la salida se hace más negativo, asegurando así, que la siguiente



Ilustr. 2 Puerta NOR SDCFL.

puerta lógica estará totalmente en corte.

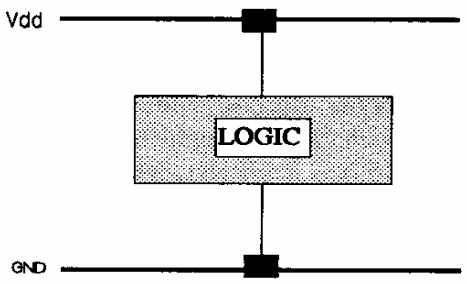
3. METODOLOGIA DE DISEÑO.

Uno de los aspectos más importantes para la realización del *layout* de las células estudiadas, es la elección de la metodología de diseño. En el presente trabajo se han utilizado dos tipos de estilos para el diseño físico. El primero corresponde a una metodología tradicional, y su implementación no tiene otro objetivo que el de comparación de resultados. La segunda metodolología es el *layout en anillo*, y ha sido elegida por dar los mejores resultados en el diseño de circuitos integrados VLSI en *GaAs*.

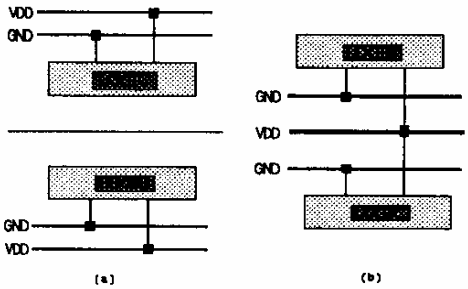
La metodología de diseño tradicional, que corresponde a la utilizada en Silicio con tecnología *n-MOS*, introduce la parte lógica entre los buses de alimentación V_{dd} y *GND*, tal como se observa en la Ilustr. 3.

El *layout en anillo*, consiste en introducir el bus *GND* entre la lógica y el bus de alimentación V_{dd}. Si la técnica de *bufferización* utiliza un bus de tensión negativa, como ocurre en nuestro caso, un bus V_{ss} se introduce entre la lógica y el bus *GND* para ambos tipos de metodologías.

La Ilustr. 4 muestra la representación del *layout en anillo*. La Ilustr. 4 (a) corresponde al *layout en anillo* con doble pareja de buses, la Ilustr. 4 (b) utiliza sólo una pareja de buses y presenta una mayor densidad de empaquetado.



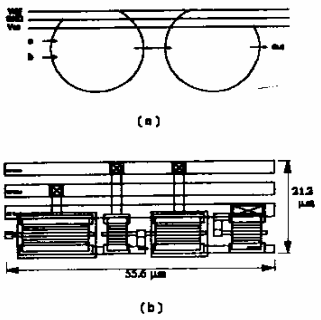
Ilustr. 3 *Layout* tradicional (tipo *n-MOS*).



Ilustr. 4 Estilos de layouts: Notación en anillo (a) con doble y (b) simple pareja de buses.

La utilización del *layout en anillo* hace necesaria la definición de una nueva representación simbólica, denominada *"Notación en Anillo"* [Eshra91], a partir de la cual, se suministra información topológica al diseñador, que permite entre otras cosas, la detección de asimetrías y descompensaciones en el diseño físico. El uso de la *"Notación en Anillo"*, ayuda a comprender como la colocación del bus *GND* entre V_{dd} y las líneas de propagación de las señales, reduce el fenómeno de acoplamiento y, además, facilita la interconexión entre los circuitos.

La Ilustr. 5 (a) refleja un inversor *SDCFL* con *notación en anillo*. Como se desprende de la figura, esta notación es una manera de representar los *layouts*, que facilita el estudio del problema de colocado de los transistores, así como la disposición de las conexiones locales, de una manera más exacta, tal y como establecen las restricciones de la tecnología. El resultado de esta notación, es el *layout en anillo* correspondiente, Ilustr. 5 (b).



...str. 5 Inversor SDCFL: (a) Notación en anillo, (b) Layout en anillo.

...observa como los transistores se disponen ...ngitudinalmente a lo largo de las líneas de ...imentación.

...ESTUDIO DE PRESTACIONES.

En este estudio, se han realizado un conjunto de sumadores y multiplicadores de 8 y 16 bits (en algunos casos, también de 4 y 5 bits), utilizando las notaciones o estilos descritos en el apartado anterior. Las formas de comparar los diseños realizados pueden ser muy diversas, basadas siempre en qué aspectos del circuito se van a evaluar: retardo, potencia o área. En el presente estudio se han utilizado dos índices de comparación para evaluar las prestaciones de los circuitos realizados.

El primero de ellos corresponde al denominado *"índice de prestaciones"* (P_{ind}), y es un indicador influenciado por el nº de puertas, frecuencia máxima y área ocupada. Este índice está definido según la fórmula siguiente:

$$P_{ind} = \frac{N^{\circ}\ de\ bits * f_{máx}}{área} \quad \frac{GHz}{mm^2}$$

Este índice mide la densidad computacional del circuito en puertas x GHz/mm^2.

El segundo indicador está definido según la ecuación:

$$\beta = \frac{N^{\circ}\ de\ bits}{delay * área}$$

Este indicador da una medida del compromiso de retardo y área ocupada, en función del número de bits. Este copromiso viene determinado por el término *delay * área*.

5. ESTUDIO DE SUMADORES.

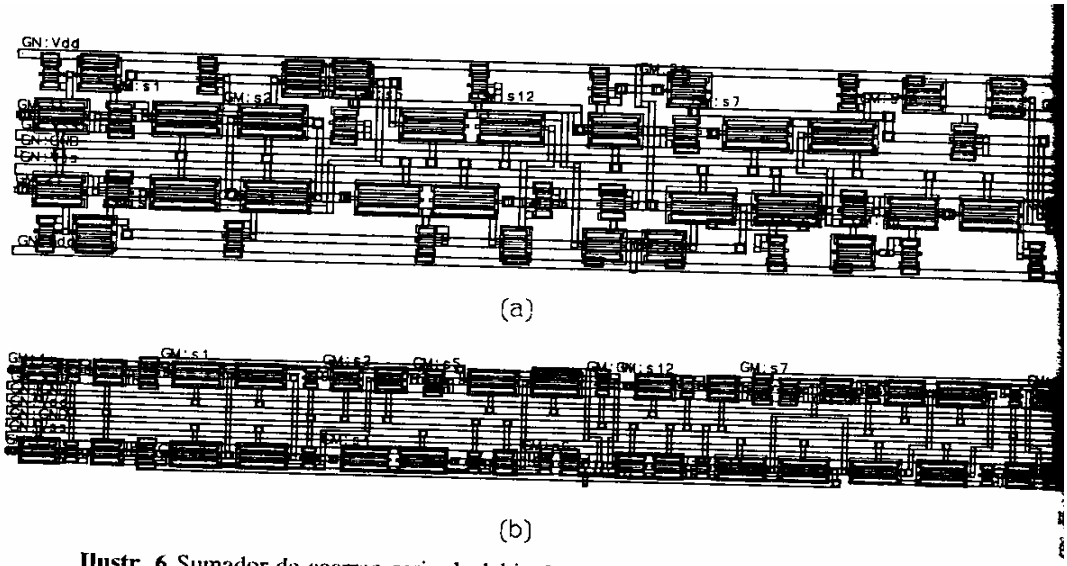
Los sumadores tienen una gran influencia en la realización de procesadores de señales y de datos. Para la tecnología *GaAs* se pueden disponer de varias arquitecturas de sumadores, que van desde el sumador de acarreo anticipado binario, con gran área ocupada y alta velocidad, hasta el sumador de acarreo serie, con bajo consumo de área, pero muy poca velocidad. Entre estos dos extremos se dispone de otros tipos de sumadores, tales como el sumador de acarreo anticipado, o el sumador selector de acarreo [SaArC91].

Las diferencias estre Silicio y *GaAs* (nº de transistores por chip, *fan-in* y *fan-out*) influyen de manera directa en la elección del sumador o multiplicador, hasta el extremo en que la arquitectura más rápida en Silicio no se corresponde con la más rápida en *GaAs*, ya que la primera puede basarse en el alto *fan-in* y *fan-out*, cosa que es totalmente inadmisible en *GaAs*.

5.1 Sumadores de acarreo serie.

Este tipo de sumadores (también denominado *ripple carry*) no es normalmente utilizado en los casos donde la velocidad juega un papel importante. Recordando que la principal ventaja de esta tecnología es la velocidad, el uso de este tipo de sumadores se ve relegado al caso donde la longitud de palabra es muy corta.

Las ecuaciones lógicas que definen el comportamiento de este tipo de sumadores son:



Ilustr. 6 Sumador de acarreo serie de 1 bit. *Layout* (a) tradicional y (b) *notación en anillo.*

$$S_i = a_i \oplus b_i \oplus c_i$$

$$C_i = a_i \cdot b_i + b_i \cdot ci\text{-}1 + a_i \cdot c_{i-1}$$

Estas ecuaciones se pueden implementar con estructuras OR-Exclusivas, realizadas a partir de dos inversores *SDCFL*, dos puertas NOR *DCFL* y una puerta OR actuando como seguidor de fuente de las anteriores. Estos sumadores han sido optimizados para bajo consumo de área y potencia.

La realización de este sumador según los diferentes estilos de *layout*, refleja que la implementación con *layout en anillo*, da sumadores más rápidos que los diseñados con la metodología tradicional. Esta rapidez se debe a que las interconexiones locales son más cortas. Esto se puede observar si se compara la Ilustr. 6 (a), que representa el *layout* de un sumador de acarreo serie realizado con una metodología tradicional, y la Ilustr. 6 (b), que muestra el *layout en anillo* del mismo sumador.

Dentro de los sumadores realizados con *notación en anillo*, aquellos que utilizan una pareja simple de buses ocupan menos área, aunque son un poco más lentos. Estos resultados se reflejan en la Tabla I. La herramienta de simulación utilizada en todas las simulaciones de este trabajo es *HSPICE*.

Tipo de Layout	No. de bits	Retardo (ns)	Potencia (mW)	Area (mm^2)	P_{tot}
F. 3	8	3.8	83	0.133	269
F. 3	16	7.7	169	0.264	133
F. 4(a)	8	3.6	83	0.175	216
F. 4(a)	16	7.4	169	0.389	95
F. 4(b)	8	3.7	83	0.150	245
F. 4(b)	16	7.5	169	0.299	121

Tabla I Resultados sumador de acarreo serie.

5.2 Sumador de acarreo anticipado.

Esta técnica se basa en los términos *propagador* y *generador*, cuyas fórmulas lógicas en función de los bits de entrada, a_i y b aparecen el la siguiente ecuación

$$G_i = a_i \cdot b_i$$

$$P_i = a_i + b_i$$

A partir de estas expresiones, la suma y e acarreo quedarían definidas como se expres en la primera ecuación de la página siguien te. La implementación de estas fórmula requiere el uso de puertas de alto *fan-in* *fan-out*, por lo que para su realización e

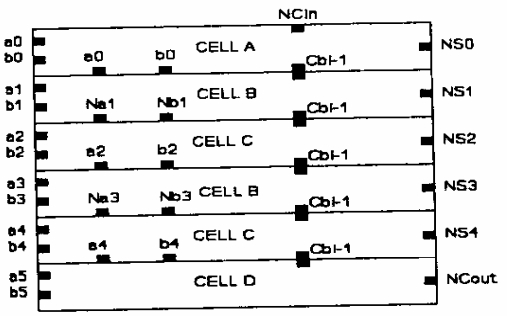
$$s_i = a_i \oplus b_i \oplus c_{i-1}$$

$$c_i = G_i + P_i\, G_{i-1} + P_i\, P_{i-1}\, G_{i-2} + \cdots + P_i\, P_{i-1}\, P_{i-2} \cdots P_0\, c0$$

aAs estas puertas deben descomponerse en tras con bajo *fan-in* y *fan-out.*

El sumador de acarreo anticipado (también denominado *Carry Look Ahead*), implementado en este estudio, consiste en $n+1$ células (operandos de n bits), conectadas serialmente. El acarreo se calcula a partir de dos términos, el primero de ellos (C_{ai}), es generado en la célula i, mientras que el segundo (C_{bi-1}), se genera en la célula anterior. Para compensar la carga de las entradas a_i y b_i, las células intermedias del sumador propagan dichas entradas y sus negadas de forma alternada.

La Ilustr. 7 muestra el diagrama de bloques para un sumador de 5 bits. Las células A y D, aparecen siempre como primera y última célula del sumador, mientras que las células B y C se repiten de forma alternada según el número de bits del sumador.



Ilustr. 7 Diagrama de bloques del sumador de acarreo anticipado de 5 bits.

Las ecuaciones lógicas de cada célula son diferentes, dependiendo de la localización de la célula dentro del sumador, o lo que es lo mismo, dependiendo de si dicha célula propaga las entradas directas o sus negadas. En concreto, las ecuaciones correspondientes a las células que se disponen en filas impares (célula *A* y célula *C*), se definen según las expresiones

$$s_i = (a_i \oplus b_i) \odot \overline{c_{i-1}} \;;$$

$$c_i = \overline{G_{i-1} + c_{bi-1}} \;;$$

$$c_{bi} = \overline{G_i + G_{i-1} + c_{bi-1}} \;;$$

caso de ser una fila par:

$$s_i = a_i \oplus b_i \oplus c_{i-1} \;;$$

$$c_i = P_{i-1} \cdot \overline{c_{bi-1}} \;;$$

$$c_{bi} = P_i\, P_{i-1}\, c_{bi-1} \;;$$

Los resultados de las simulaciones para las implementaciones de estas ecuaciones, con las diferentes metodologías, se reflejan en la Tabla II (en este caso sólo se han implementado con *layout en anillo*).

Tipo de Layout	No. de bits	Retardo (ns)	Potencia (mW)	Area (mm^2)	P_{int}
F. 4(a)	9	2.9	80	0.211	337
F. 4(a)	16	4.9	140	0.368	204
F. 4(b)	9	2.5	80	0.210	392
F. 4(b)	16	4.7	120	0.363	302

Tabla II Resultados sumador de acarreo anticipado.

Se observa como la realización con una pareja simple de buses es más rápido, rapidez que se obtiene debido a que las comunicaciones entre células son predominates en la ruta crítica, a diferencia que el sumador de acarreo serie, donde predominaban las interconexiones locales.

5.3 Sumador de acarreo anticipado binario.

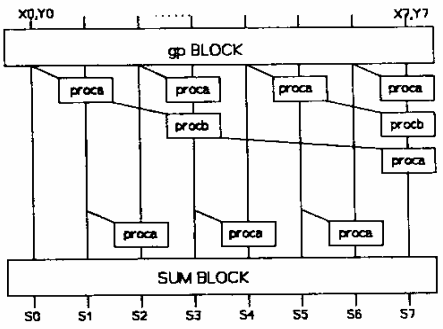
El sumador de acarreo anticipado binario, más conocido como *Brent and Kung* [Bre-Ku83], también está basado en los términos *propagador* (P_i) y *generador* (G_i) del acarreo, descritos en el apartado anterior. Las ecuaciones de la suma y el acarreo para este tipo de sumadores son:

$$c_i = G_i$$

$$s_i = P_i \oplus c_{i-1}$$

La implementación de estas ecuaciones, tiene como principal ventaja el utilizar puertas de dos entradas, muy factible para esta tecnología, teniendo además un *fan-out* muy limitado.

La estructura de bloques del *Brent & Kung* (Ilustr. 8) está basada en en una primera etapa (*gp*), donde se calculan los términos p_i y g_i, una etapa final (*out*), donde se realiza la suma, y unas etapas intermedias. Estas etapas intermedias corresponden al árbol de generación del acarreo, y están formadas por células de retardo, y células procesadoras y propagadoras de acarreo (células *proca* y *procb*). La iplementación de éste árbol con lógica *SDCFL*, introduce el mismo retardo en las células procesadoras que en las células de retardo.



Ilustr. 8 Diagrama de bloques del sumador de acarreo anticipado binario..

Los resultados de la implementación de este algoritmo, con *layout en anillo*, se reflejan en la Tabla III.

La implementación de este sumador, en cualquiera de los estilos utilizados, tiene un aprovechamiento de área muy escaso, debido a que su estructura no es nada regular, y además, el área ocupada por las células de retardo es muy pequeña comparada con el área ocupada por las células procesadoras.

Tipo de Layout	No. de bits	Retardo (ns)	Potencia	Area (mm^2)	P_{ad}
F. 4(a)	8	1.3	92	0.466	510
F. 4(a)	16	1.7	218	1.142	382
F. 4(b)	8	1.4	92	0.434	507
F. 4(b)	16	1.8	218	1.078	381

Tabla III Resultados sumador de acarreo anticipado binario.

5.4 Sumador selector de acarreo.

El último tipo de sumador implementado en este trabajo es el sumador selector de acarreo, más conocido como *Carry Save Adder*.

En este tipo de sumadores, la suma de *n bits*, se divide en varios bloques. Cada uno de estos bloques suma *m bits*, siendo $m<n$. Cada bloque está compuesto por dos sumadores de *m bits*, cuyas estradas de acarreo se ponen a *uno lógico*, y a *cero lógico*, y sus salidas se llevan a un multiplexor, donde la entrada de selección corresponde al acarreo de salida de la etapa anterior.

La técnica para implementar los bloques parciales de sumas puede ser cualquiera de las vistas hasta ahora. Es recomendable, si la longitud de palabra de esos bloques es pequeña, utilizar sumadores de acarreo serie, ya que la realización con otro tipo de algoritmo puede incrementar el área del sumador de forma exagerada.

Con la finalidad de igualar el retardo de las señales de llegada al multiplexor (entrada de las sumas parciales y entrada de selección), los diferentes bloques se diseñan con diferentes longitudes de palabras.

Los acarreos de salida de los diferentes bloques, tienen un *fan-out* muy alto, debido a los multiplexores, hecho por el cual se han introducido *buffers* en dichas señales. Con ello se reduce el *fan-out* a un máximo de tres puertas.

Los resultados de las simulaciones para un sumador selector de acarreo de 8 y 16 bits, con notación en anillo y doble par de buses de alimentación se muestra en la Tabla IV.

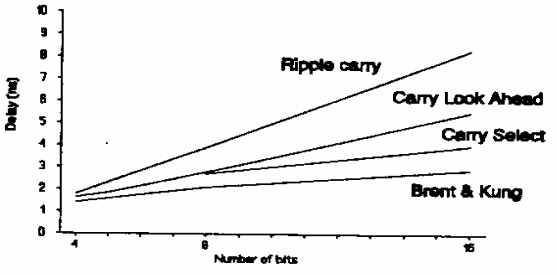
Tipo de Layout	No. de bits	Retardo (ns)	Potencia (mW)	Area (mm^2)	P_{ind}
F. 4(a)	8	2.3	107	0.370	302
F. 4(a)	16	3.5	270	0.930	199

Tabla IV Resultados sumador selector de acarreo.

5.5 Comparación de resultados.

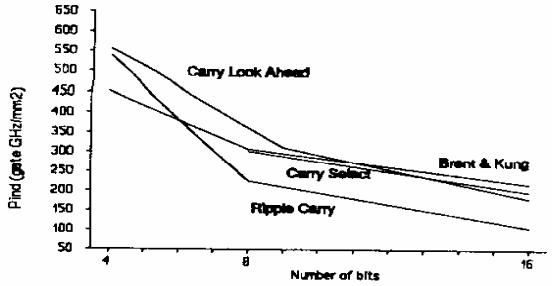
En este apartado se evalúan los diferentes sumadores presentados, según los índices definidos en el apartado 4. Debido a las restricciones de la tecnología *GaAs* se demuestra que no se pueden extrapolar los resultados obtenidos en Silicio, a la tecnología de *GaAs*.

La Ilustr. 9 representa el retardo frente al número de bits, donde se observa la predominancia del *Brent & Kung*, y la mayor lentitud del sumador de acarreo serie. Diferencia, que por otro lado no se acentúa mucho para longitudes de palabras pequeñas. Sin embargo, este análisis no es del todo completo, ya que limitaciones como el número de transistores por chip hace que la realización de un *Brent & Kung* de longitud de palabra muy grande, pueda verse imposibilitada en un diseño VLSI en *GaAs*.



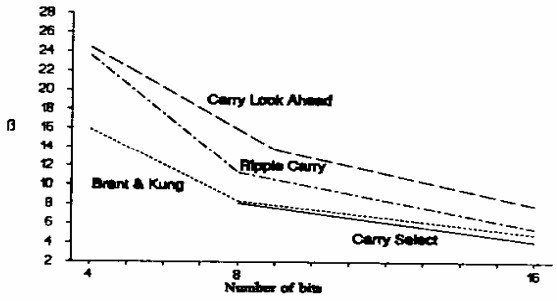
Ilustr. 9 Retardo frente Número de bits.

Por tanto, para poder analizar las prestaciones de cada uno de los sumadores, así como la posibilidad de implementarlos en diseños VLSI en *GaAs*, se estudia el índice de prestaciones (P_{ind}) frente al número de bits. La Ilustr. 10 refleja como el *Brent & Kung* obtiene un mayor índice de prestaciones frente a los demás, cuando el número de bits es superior a 16. En otros casos, sumadores como el de acarreo anticipado, presenta un mejor índice de prestaciones debido a su bajo consumo de área frente al *Brent & Kung*.



Ilustr. 10 P_{ind} frente Número al número de bits.

Por último, el análisis del factor β en cada uno de los sumadores (Ilustr. 11), demuestra que el sumador de acarreo anticipado presenta el mejor compromiso entre área y velocidad, disminuyendo la diferencia con respecto al *Brent & Kung* y al sumador selector de acarreo, para longitudes de palabras mayores de 16 bits.



Ilustr. 11 β frente al número de bits.

6. MULTIPLICADORES.

Los mismos compromisos aplicados en el diseño de sumadores, aparecen en el diseño de multiplicadores y otros subsistemas arit-

méticos y de procesamiento digital de señales. En el lado de los multiplicadores, se presentan una gran variedad de arquitecturas con dos grandes vertientes: arquitecturas paralelas y arquitecturas seriales. Estas últimas presentan ventajas como disminución en el área ocupada y en el número de pines de entradas y salidas.

En el estudio realizado se han desarrollado dos arquitecturas paralelas que corresponden a los algoritmos de *Booth* por un lado, y *Baugh y Wooley's* por otro [Cavan84]; también se ha desarrollado una arquitectura serial y una serial/paralela.

6.1 Multiplicadores paralelos.

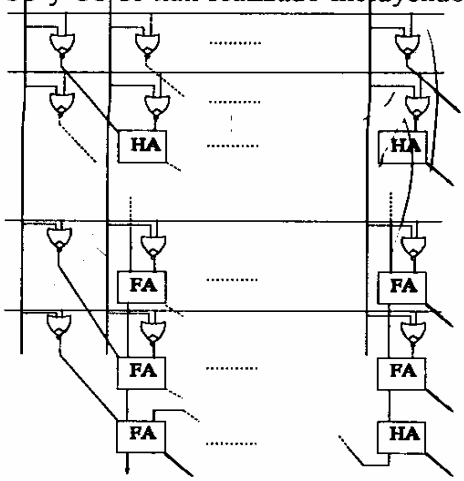
6.1.1 Algoritmo de Booth.

El algoritmo de multiplicación binaria, como es sabido, se implementa a base de operaciones de suma y desplazamiento. De esta forma, la implementación elemental de la multiplicación binaria se realiza mediante una *AND* lógica del multiplicando con el bit apropiado del multiplicador.

La Ilustr. 12 refleja el diagrama de bloques de un multiplicador con este algoritmo, donde las puertas *AND* se han sustituido por puertas *NOR*, para su adaptación en esta tecnología.

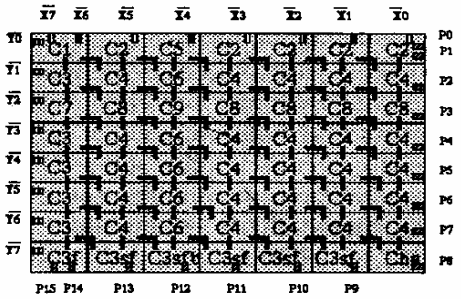
Para reducir el retraso del multiplicador, el *full-adder* de salida se ha implementado con notación en anillo y doble par de buses de alimentación.

Nótese que para multiplicadores de más de cuatro bits, el *fan-out* de las señales de entrada puede retrasar bastante la implementación. Por esto se deben incluir células amplificadoras de señal, de tal forma que el *fan-out* nunca supere las tres puertas. Con esta señalización, el diagrama de bloques para el caso de un multiplicador de *Booth* de 8 x 8 bits es la que se muestra en la Ilustr. 13. En esta implementación, las células *C6* y *C8* se han realizado incluyendo los



Ilustr. 12 Diagrama de bloques del multiplicador de *Booth*.

buffers adicionales para la limitación del *fan-out*. Estos *buffers* se han implementado con dos inversores *DCFL* y un seguidor de fuente.



Ilustr. 13 Plano de base del multiplicador de *Booth*.

La Tabla V muestra los resultados de la implementación de este multiplicador, en los caso de utilizar como sumador de salida, un *sumador de acarreo serie*, un *sumador de acarreo anticipado*, y un *sumador de acarreo anticipado binario*. La realización del multiplicador de *Booth* con *Brent & Kung* a la salida, aumenta la velocidad.

6.1.2 Algoritmo de Baugh y Wooley.

Esta estructura, que usa la multiplicación en

Tipo de Multiplicador	Retardo (ns)	Potencia (mW)	Area (mm^2)	P_{int}
Booth/RC	7.4	255	1.542	95
Booth/CLA	6.4	252	1.550	116
Booth/B&K	5.4	267	1.820	126

Tabla V Resultados multiplicador de Booth.

implemento a dos, fué propuesta por Baugh Wooley [Cavan84]. Su implementación es eramente diferente para el caso en que los meros a multiplicar *m x n* sean de igual o ferente longitud. La distribución de las ñales de entrada es la misma que para el ultiplicador de *Booth*.

a realización del diseño físico de este ultiplicador también se ha hecho imple- nentando la etapa final con diferentes estruc- uras de sumadores.

a Tabla VI refleja los resultados de las imulaciones para este tipo de multiplicador aralelo.

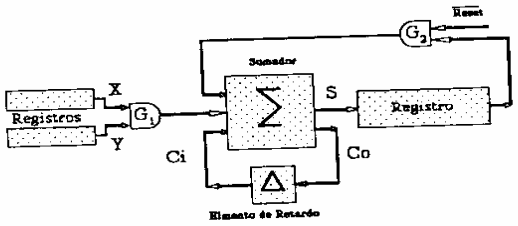
Tipo de Multiplicador	Retardo (ns)	Potencia (mW)	Area (mm^2)	P_{int}
Baugh/RC	7.5	265	1.190	82
Baugh/CLA	6.5	272	1.920	97
Baugh/B&K	5.5	284	2.190	109

Tabla VI Resultados multiplicador de Baugh/Wooley.

6.2 Multiplicadores seriales y seriales/paralelos.

La técnica de multiplicación serial en *GaAs* es bastante apropiada para aplicaciones de procesado digital de señales, ya que se cuenta con una frecuencia de reloj muy rápida, y por otro lado la limitación del número de transistores por chip queda superada, al necesitar éste tipo de multiplicador mucho menos hardware.

La ruta de datos para un multiplicador serial está compuesta de una puerta NOR, de un registro de 8 bits con entradas en paralelos y salida serie y un sumador con acarreo de entrada, tal como se refleja en Ilustr. 14. Sin embargo, la implementación del multiplicador serial tiene un aspecto crítico, como es el diseño de los registros de desplazamientos.



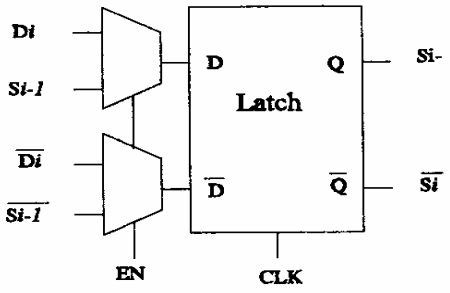
Ilustr. 14 Diagrama de bloques de un multiplicador serial.

Estos deben ser implementados con lógica estática, con un consumo de área por bit, similar a un sumador con acarreo de entrada.

La realización del registro de desplazamiento con entradas en paralelo y salida serie, se ha implementado anexionando un multiplexor a la entrada del registro de desplazamiento. No obstante, para evitar el problema que pueda aparecer por el *skew* de la señal a la llegada al registro de desplazamiento, la solución que se presenta es la de disponer de dos multiplexores idénticos, uno para la entrada directa, y otro para la entrada negada a la célula de almacenamiento, tal como aparece en la Ilustr. 15.

La Tabla VII presenta los resultados de área, potencia y retardo, de los bloques que componen el multiplicador serial.

La simulación de un multiplicador serial de 8 x 8 bits hace uso de una señal de reloj, que trabajando a una frecuencia de *1 GHz* da un retardo de 64 ns, con un área de 0.536 mm^2 y una potencia consumida de 92mW. Estos resultados demuestran que la utilización de un multiplicador paralelo, siempre prevalecerá cuando el área utilizada no sea un parámetro crítico.

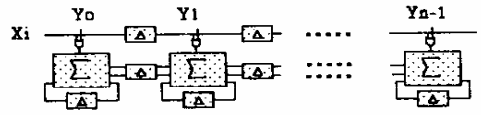


Ilustr. 15 Registro de desplazamiento de un bit.

Célula	Retardo (ns)	Potencia (mW)	Area (mm^2)
Latch+MUX	0.53	3.41	0.023
Latch	0.53	1.91	0.012
Sumador	0.65	6.94	0.023

Tabla VII Resultado de los bloques del multiplicador serial.

Otra alternativa a la multiplicación serial, que reduce el tiempo de computación, es la multiplicación serial/paralela. La implementación básica es la que se describe en Ilustr. 16. En esta estructura, la multiplicación se lleva a cabo mediante sucesivas adiciones dentro de una misma columna de la matriz de productos parciales, convenientemente desplazadas.



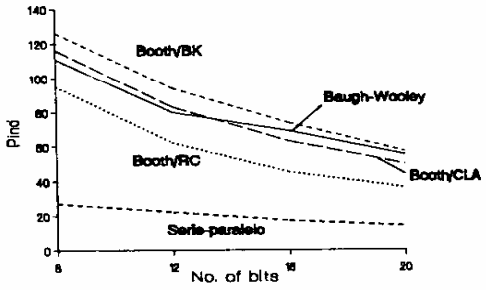
Ilustr. 16 Diagrama de bloques de un multiplicador serie paralelo.

La ruta de datos para el multiplicador serie-paralelo de 8 x 8 bits, consta de ocho sumadores con acarreo, almacenamiento temporal de las sumas parciales, ocho puertas NOR, y un registro de desplazamiento de 8 bits, con entradas en paralelo y salida serie (implementado de igual forma que para el multiplicador serie).

De esta forma, la multiplicación de 8 x bits, precisa de 8 + 8 pulsos de reloj. Es permite, que a igual frecuencia de trabajo GHz) que para el multiplicador serie, dich multiplicación se pueda realizar en 16 n con un coste de área de 0.92 mm^2.

6.3 Comparación de resultados.

La Ilustr. 17 presenta el índice de prestacio nes (P_{ind}) para los multiplicadores expuesto (excepto multiplicador serial).



Ilustr. 17 Indice de prestaciones para los multiplicadores paralelos.

El índice más bajo corresponde a los multi plicadores seriales paralelos, debido a que e retardo de éstos, tiene mayor influenci frente al área ocupada. Dentro de los multi plicadores paralelos, el que mejor resultado da es el de Booth con sumador *Brent* y *Kung* a la salida. El algoritmo de *Baugh Wooley* obtiene un mayor índice de presta ciones para un número de bits superior a 2(

Para poder comparar esta gráfica con lo resultados de la multiplicación serial, l Tabla VIII presenta los resultados de área retardo, para una multiplicación serial de 8 8 bits.

Se observa como el área ocupada disminuy mucho, con respecto a los multiplicadore paralelos, no obstante, al tener un retard muy alto (casi 10 veces superior) su índic de prestaciones es muy bajo. Dentro de tod

Tipo de Multiplicador	Retardo (ns)	Potencia (mW)	Area (mm^2)	P_{ind}
Serie	64.0	255	0.540	9
Serie/Paralelo	16.0	252	0.940	27.1

Tabla VIII Resultados de la multiplicación serie y serie paralelo (8 x 8 bits).

llos, el multiplicador serie/paralelo se pre-enta como buena alternativa cuando el ctor área es crítico.

. Bibliografía.

GilPa87] B.K. Gilbert and G-W. Pan, "The aplication of Gallium Arsenide Integrated Circuit Technology to the Design and Fabrication of Future Generation Digital Signal Processors: Problems and Promises", *Proceedings of the IEEE*, Vol. 76, Nº 7, Julio 1987, pp. 816-834.

hra91] K. Eshraghian, "Fundamentals of High Speed Systems: Gallium Arsenide VLSI Technology", Prentice Hall, 1991.

hra90] K. Eshragian, "Fundamentals of High speed systems: Gallium Arsenide VLSI Technology". Notas de curso doctoral en la Universidad de Las Palmas de G. C. (España), 1990.

[Cavan84] Cavanagh, J.J., "Digital Computer Arithmetic Design and Implementation", McGraw-Hill, 1984.

[LonBu90] Long, S.I., and Butner, S.E. "Gallium arsenide digital integrated circuit design", McGraw-Hill, New York, 1990.

[SaArC91] R. Sarmiento, V. Armas, P.P. Carballo, J. López y A. Núñez, "Diseño y optimización de sumadores en tecnología GaAs", *VI jornadas de diseño de circuitos integrados*, Santander, 1991.

[BreKu83] R. P. Brent and H. T. Kung, "A Regular Layout for Parallel Adders", IEEE *Transactions on Computers*, Vol. 16, Nº 12, December 1983, pp. 41-58.

BIOGRAFIA DE LOS AUTORES.

Valentín de Armas nació en Las Palmas de G.C. en Noviembre de 1966. Recibió el título de Ingeniero Industrial, rama Eléctrica, en la Universidad de Las Palmas de G.C. en 1991. Posteriormente entró en el *Centro de Microelectrónica Aplicada (CMA)* como investigador dentro de la división ASIC, engargándose del estudio y diseño de sistemas y subsistemas aceleradores hardware en GaAs. En 1991 comenzó a impartir clases en la ETSIT en la asignatura "Ordenadores" así como en la ETSII en la asignatura "Calculadoras". Inició sus estudios de tercer ciclo en 1991 en la Universidad de Las Palmas, donde ha completado el programa de doctorado en Ingeniería Electrónica. Ha participado en el proyecto europeo PATMOS (ESPRIT 3237). Actualmente sus investigaciones van dirigidas al campo del diseño de rutas de datos para tecnología GaAs. Ha publicado 9 artículos, la mayor parte de ellos sobre diseño de circuitos integrados en Arseniuro de Galio.

Juan A. Montiel nació en Las Palmas de G.C. en 1964. Recibió el título de Ingeniero Superior Industrial, rama Eléctrica, en 1991 por La Universidad de Las Palmas de Gran Canaria. Posteriormente se incorporó en la división ASIC del *Centro de Microelectrónica Aplicada (CMA)* donde actualmente realiza tareas de investigación el el diseño de estructuras de alta velocidad. Durante los años 1987 a 1990 trabajó en el área de Ciencias de la Computación e Inteligencia Artificial dentro del departamento de Informática y Sistemas, donde era investigador encargado del desarrollo de sistemas de información geográfica. Desde 1991 hasta 1992 trabajó en la implementación *full-custom* de multiplicadores hardware de alta velocidad, con tecnología GaAs de Gigabit Logic y Vitesse. Ha participado en el proyecto europeo PATMOS. Actualmente imparte docencia en la asignatura "Instrumentación Electrónica Avanzada" en la ETSIT de la Universidad de Las Palmas de G.C. actual línea de investigación va encamin al estudio y desarrollo de sistemas de velocidad, entornos de diseño y compilad de subsistemas hardware de alta velocid Ha publicado 10 artículos, varios de ellos carácter internacional.

José Fco. López nació en La Palma, Is Canarias, en Abril de 1966. Recibió el tit de Licenciado en Ciencias Físicas, especi dad de Electrónica, por la Universidad Sevilla (España) en 1989. Posteriormente unió al *Centro de Microelectrónica Aplic (CMA)*, como investigador encargado diseño de sumadores y multiplicadores alta velocidad, dentro de la división ASIC. Actualmente es profesor de la asig tura "Integración de Sistemas Analógico Sensores" en la ETSIT de la Universidad Las Palmas de G.C., donde ha completad programa doctoral en Ingeniería Electróni Ha participado en PATMOS (ESPRIT 32 y en otros proyectos de investigación. Dur te el año 1992 fue contratado por la empr *Thomson Composants Microondes* en Or (Francia), como investigador encargado diseño de memorias ROM en Arseniuro Galio. Su actual línea de investigación encaminada al diseño de memorias de a velocidad y a la creación de compiladores estructuras. Ha publicado ocho artículos, mayoría de ellos sobre diseño GaAs.