

**DISEÑO DE MULTIPLICADOR DE 64 BITS
UTILIZANDO LA TÉCNICA
DE BARRIDO SUPERPUESTO**

MONTALVO, LUIS ING.
ESCUELA POLITECNICA NACIONAL

RESUMEN

Se presenta el desarrollo del novel algoritmo de multiplicación por barrido superpuesto de 3-bits, así como también los aspectos importantes del diseño de la arquitectura, redes lógicas y layout de un multiplicador de 64-bits utilizado como caso de estudio.

INTRODUCCION

El objetivo de la presente investigación es desarrollar un prototipo de multiplicador de números de 64-bits, utilizando la novel técnica de barrido superpuesto de 3-bits, desarrollada en Ohio University. El diseño deberá cumplir los siguientes objetivos:

- 1.- Capacidad de ser implementado en VLSI en base a un proceso tecnológico fácilmente accesible, lo cual permitirá que el multiplicador pueda ser fabricado por un amplio número de laboratorios.
- 2.- Capacidad de ser encapsulado en un paquete normalizado bien establecido.
- 3.- Capacidad de ser conectado a un bus de sistema asíncronico normalizado, lo cual permitirá que la precisión alcanzable con el multiplicador pueda ser utilizada por un dispositivo más inteligente, tal como un maestro de bus de sistema (CPU, coprocesador, etc.)

ALGORITMOS BASICOS DE MULTIPLICACION

El algoritmo básico de multiplicación consiste en examinar, a la vez, un solo bit del multiplicador. Si el bit es "1", el multiplicando es sumado al totalizador y éste es luego desplazado un bit hacia la derecha; si el bit es "0", el totalizador es únicamente desplazado un bit hacia la derecha. Se procede entonces a examinar el siguiente bit y el proceso se repite hasta examinar los n-bits del multiplicador.

El proceso de multiplicación puede ser acelerado si se examinan más de un bit del multiplicador a la vez. Por ejemplo, si el número de bits examinado por cada vez es dos, la multiplicación se realiza de la siguiente manera:

Se examinan dos bits consecutivos del multiplicador (B_{i+1} , B_i) y se suma al totalizador un múltiplo del multiplicando de acuerdo con lo establecido en la Tabla #1; se desplaza luego el totalizador 2 bits hacia la derecha. Se procede entonces a examinar los dos bits siguientes y el proceso se repite hasta examinar los n-bits del multiplicador.

Tabla #1: Múltiplos de Multiplicando a Ser Sumados para Barrido de 2-bits no Superpuestos.

B_{i+1}	B_i	Múltiplos a Sumar
0	0	0
0	1	A
1	0	2A
1	1	3A

Si para sumar los múltiplos del multiplicando al totalizador se utiliza un sumador capaz de sumar tres números (SAC + SPC), la generación de 3A no es necesaria y puede substituirse por la suma al totalizador de los múltiplos A y 2A.

El método descrito se conoce con el nombre de "Multiplicación por Barrido de 2-bits no Superpuestos".

Para el caso de examinar 3-bits consecutivos del multiplicador cada vez, los múltiplos del multiplicando que deben sumarse al totalizador son los que se resumen en la Tabla #2.

Tabla #2: Múltiplos de Multiplicando a Ser Sumados para Barrido de 3-bits no Superpuestos.

B_{i+2}	B_{i+1}	B_i	Múltiplos a Sumar
0	0	0	0
0	0	1	A
0	1	0	2A
0	1	1	3A
1	0	0	4A
1	0	1	5A
1	1	0	6A
1	1	1	7A

En dicha tabla puede observarse que existen múltiplos del multiplicando (3A, 5A, 6A y 7A) que no pueden generarse por simples desplazamientos del multiplicando y que por lo tanto su generación requeriría de sumadores adicionales. El número de múltiplos del multiplicando puede reducirse a la mitad utilizando la técnica de barrido superpuesto de r-bits, donde "r" es el número de bits consecutivos que se examinan cada vez.

BARRIDO SUPERPUESTO DE r-BITS

El fundamento de la técnica consiste en reemplazar una cadena de k "1s" consecutivos del multiplicador por una suma al inicio y una resta al final de la cadena, en virtud de la siguiente propiedad:

$$2^{i+k} - 2^i = 2^{i+k-1} + 2^{i+k-2} + \dots + 2^i$$

De manera que una cadena de k "1s" consecutivos tal como la que se muestra en (1) puede reemplazarse por (2).

Columna: $i+k, i+k-1, i+k-2 \dots i, i-1$

Bit: 0 1 1 ... 1 0 (1)
 Bit: 1 0 0 ... -1 0 (2)

Se divide entonces el multiplicador en grupos de r -bits. Se examinan " $r+1$ " bits a la vez, r -bits del grupo presente y el bit menos significativo del grupo siguiente; de manera que, el bit menos significativo de cada grupo es examinado dos veces.

Los múltiplos del multiplicando que deben sumarse al totalizador dependen de " r ".

BARRIDO SUPERPUESTO DE 3-BITS

Los múltiplos del multiplicando A que deben sumarse al totalizador para el caso de $r = 3$ se resumen en la Tabla #3.

El análisis de la Tabla #3 muestra que, con excepción de $6A$, todos los múltiplos del multiplicando son de la forma $2^i \cdot A$ y por lo tanto pueden obtenerse mediante simples desplazamientos del multiplicando. $6A$ puede obtenerse sumando $2A$ y $4A$.

Tabla #3: Múltiplos del Multiplicando para Barrido Superpuesto de 3-bits.					
B_{i+3}	B_{i+2}	B_{i+1}	B_i	Múltiplos a Sumar	Comentario
0	0	0	0	0	Cadena de "0s"
0	0	0	1	2A	Fin de cadena de 1s
0	0	1	0	2A	1 aislado
0	0	1	1	4A	Fin de cadena de 1s
0	1	0	0	4A	1 aislado
0	1	0	1	6A	Fin de cadena de 1s 1 aislado
0	1	1	0	6A	2 "1s" aislados
0	1	1	1	8A	Fin de cadena de 1s,
1	0	0	0	-8A	Inicio de cadena
1	0	0	1	-6A	Inicio, fin de cadena
1	0	1	0	-4A	Inicio de cadena, 1 aislado
1	0	1	1	-4A	Inicio, fin de cadena
1	1	0	0	-4A	Inicio de cadena
1	1	0	1	-2A	Inicio, fin de cadena
1	1	1	0	-2A	Inicio de cadena
1	1	1	1	0	Cadena de "1s"

ARBOLES SUMADORES

Mayor paralelismo aún, puede añadirse al proceso de multiplicación, si se examinan simultáneamente más de un grupo de r -bits; en tal caso es necesario disponer de estructuras capaces de sumar más de 3-bits a la vez.

Tales estructuras se conocen con el nombre de árboles sumadores y consisten de varios sumadores (SAC) arreglados en forma de árbol con un sumador (SPC) en la raíz para sumar los vectores suma y carry provenientes del último sumador SAC.

Por ejemplo, si el número de grupos examinado simultáneamente es dos, puede tenerse la estructura que se muestra en la Fig. #1.

Debe mencionarse que la estructura presentada no es única; sin embargo, es la óptima desde el punto de vista de mínimo producto tiempo * área para el caso de multiplicadores de 64-bits [1].

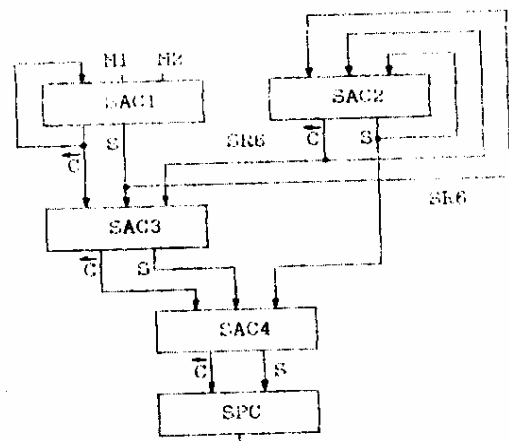


Fig. #1.- Árbol Sumador para Dos Grupos de r -bits.

GENERACION DE MULTIPLOS NEGATIVOS DEL MULTIPLICANDO

En el multiplicador desarrollado, no se ha analizado todavía el problema de la generación de los múltiplos negativos del multiplicando (-2A, -4A, -6A y -8A). Tales múltiplos pueden generarse obteniendo el complemento de 2 de los correspondientes múltiplos positivos del multiplicando, utilizando la misma circuitería del árbol sumador.

En los árboles sumadores, los vectores del carry y de la suma de cada SAC se hallan defasados en un bit, por lo cual el bit menos significativo del carry siempre es cero. Llámese a ésta posición un hueco. Obsérvese que un árbol sumador siempre tiene un número de huecos mayor o igual al número de entrada de datos, independientemente de la estructura del árbol. Por ejemplo en la Fig. #1 se tienen cuatro bits para el vector del carry y dos bits para entrada de datos al árbol. Puede asignársele entonces un hueco a cada entrada de datos. Cuando se requieran múltiplos negativos del multiplicando como entrada, se complementan los respectivos múltiplos positivos del multiplicando y se llena el hueco correspondiente con un "1". Es decir, se ha implementado la suma en complemento de 2.

ALGORITMO DE MULTIPLICACION CON BARRIDO SUPERPUESTO DE 3-BITS

El análisis teórico de los procesos de multiplicación presentado es el fundamento del siguiente algoritmo de multiplicación con barrido superpuesto de 3-bits.

- Paso 1: Generación de 2A, 4A y 8A mediante desplazamientos del multiplicando A y enrutamiento hacia sus posiciones correspondientes en las entradas del árbol sumador.
- Paso 2: Generación de 6A, mediante la suma de 2A y 4A.
- Paso 3: Si el bit menos significativo del multiplicador es igual a uno, alimentar el árbol sumador con el multiplicando complementado. Caso contrario alimentarlo con cero.
- Paso 4: Barrer cada grupo de 3 bits y el bit menos significativo del siguiente grupo. Seleccionar los múltiplos del multiplicando necesarios de acuerdo con la Tabla #3.
- Paso 5: Alimentar los múltiplos del multiplicando al árbol sumador. Repetir los pasos 4 y 5 hasta que se hayan barrido todos los bits del multiplicador.

ESTRATEGIA DE DISEÑO

Una vez desarrollado el algoritmo, es necesario encontrar la arquitectura más adecuada que permita implementarlo en silicio y debe definirse de manera clara la metodología de diseño a adoptarse.

La metodología jerárquica de diseño (Top-Down) de circuitos integrados permite al diseñador dividir el proceso de diseño en etapas organizadas jerárquicamente, de tal manera que el o ella puede concentrarse en

los aspectos importantes de cada etapa sin tener que preocuparse de los detalles de implementación de las etapas inferiores. Esta metodología de diseño de circuitos integrados ha mostrado ser muy eficiente para el diseño de sistemas muy complejos y ha sido adoptada en la presente investigación.

SELECCION DEL PROCESO TECNOLOGICO

Las dimensiones del circuito integrado a diseñarse deben ser tales que pueda ser encapsulado utilizando tecnología normalizada. La Tabla #4 resume las especificaciones para encapsulados DIP de 40, 48 y 64 pines [2].

Tabla #4: Area en Función de Número de Pines para Encapsulado DIP.

Pines (#)	40	48	64
Area (mm ²)	7	7	10.5

El ancho del multiplicador de 64-bits está determinado por el ancho del árbol sumador; en consecuencia, un estimado de éste parámetro será suficiente para definir las dimensiones del multiplicador. Diseños preliminares de celdas SAC condujeron a los resultados que se resumen en la Tabla #5.

Tabla #5: Estimado del Ancho del Multiplicador para Diferentes Tecnologías.

Tecnología (um)	2.5	1.0	0.3	0.1
Ancho del chip (mm)	14.2	5.6	1.7	0.11

Las tecnologías de 2.5 um y 1.0 um son ampliamente utilizadas en la implementación de prototipos en VLSI. La tecnología de 0.3 um se halla al momento en la etapa de investigación y 0.1 um constituye el límite impuesto por la física de semiconductores [3].

La contrastación de las Tablas #s 4 y 5 y la consideración que se acaba de exponer, muestran que el multiplicador de 64-bits es factible de implementarse utilizando tecnología de 1.0 um y que puede ser encapsulado en un chip DIP de 40, 48 o 64 pines. Este resultado permite al multiplicador cumplir con los objetivos 1 y 2 planteados.

ESTUDIO DEL AMBIENTE DE TRABAJO DEL MULTIPLICADOR

La característica 3 del objetivo de diseño demanda que el multiplicador de 64-bits sea diseñado de manera tal que pueda ser conectado a un bus de sistema normalizado. Las señales de interfaz que el multiplicador debe tener para poder conectarse al bus del sistema están determinadas por el protocolo de intercambio de datos a través del bus de tal sistema. El número de pines que el multiplicador requiere está determinado por el número total de líneas que ingresan al y salen del multiplicador. A su vez, el número de pines del multiplicador determina el tipo de encapsulado adecuado para el multiplicador.

Máximo Ancho de Bus de Sistema Comercial.-

El máximo ancho del bus de datos en los buses de sistemas disponibles comercialmente al momento es 32-bits. Los mas populares de tales buses de 32-bits son: VME bus (IEEE P1014), Futurebus (IEEE P896.1), Multibus II (IEEE Pxxx), Nubus (IEEE Pyyy) y Fastbus (ANSI/IEEE 690) [4].

Señales Necesarias de Entrada/Salida del Multiplicador.-

Las características del proceso de interacción entre el multiplicador y un maestro BTM son las características de un dispositivo de entrada/salida típico. A continuación se resumen las señales que el multiplicador requiere para poseer tales características.

- 1.- 32 líneas (Da1,, Do) para transferencia de datos.
- 2.- Señales de lectura (R), escritura (W) y reconocimiento de transferencia de datos (DTACK*).
- 3.- Señales de pedido de interrupción (IRQ) y reconocimiento de interrupción (IACK-IN*, INACKOUT*).
- 4.- Señal de indicación que el multiplicador no está disponible (MPRBSY).
- 5.- Señal de habilitación del multiplicador (CS).
- 6.- Señal de inicio de multiplicación (START).
- 7.- Líneas de alimentación (VDD, GND).
- 8.- Dos líneas de dirección.

El número total de pines que requiere el multiplicador es 45 y en consecuencia puede ser encapsulado en un paquete normalizado DIP de 48 pines.

Configuración de los Registros Internos del Multiplicador.-

Para un maestro BTM, el multiplicador no será mas que una colección de registros que pueden ser accedidos ya sea como localidades de memoria, o como pórticos de entrada-salida.

La Fig. #2 muestra la configuración de los registros internos del multiplicador, desde el punto de vista de un maestro BTM.

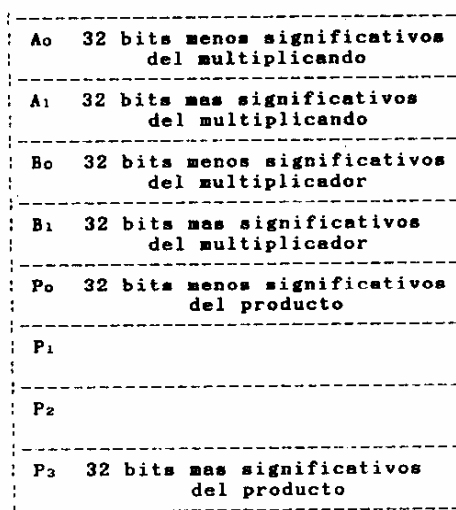


Fig. #2.- Configuración de los registros internos del multiplicador.

La Tabla #6 muestra la manera como el BTM selecciona cada uno de los registros del multiplicador para leer o escribir datos.

Tabla #6: Selección de Registros del Multiplicador.						
CS	R	W	C ₁	C ₀	BUS	COMENTARIO
1	X	X	X	X	Z	Multiplicador desconectado del bus
0	1	0	0	0	P ₀	Lea bloque menos significativo del producto P.
0	1	0	0	1	P ₁	
0	1	0	1	0	P ₂	
0	1	0	1	1	P ₃	Lea bloque mas significativo del producto P.
0	0	1	0	0	A ₀	Escriba bloque menos significativo de multiplicando A.
0	0	1	0	1	A ₁	Escriba bloque mas significativo del multiplicando A.
0	0	1	1	0	B ₀	Escriba bloque menos significativo de multiplicador B.
0	0	1	1	1	B ₁	Escriba bloque mas significativo de multiplicador B.

DISEÑO DE ARQUITECTURA

La Fig. #3 muestra la arquitectura que permite implementar el algoritmo de multiplicación por barrido superpuesto de 3-bits.

La circuitería de interfaz permite conectar el multiplicador a un bus asincrónico de 32-bits.

ROUTESEL es la red de enrutamiento y selección de los múltiplos del multiplicando A.

El Área dentro de la líneas de puntos constituye la sección aritmética y al momento se trabaja en su implementación.

SPC-AUX es un sumador auxiliar para la generación de 6A.

El bloque de control ha sido subdividido en dos secciones: la sección de control de interfaz y la sección de control aritmético. Las funciones de cada una de estas secciones se explica en detalle en los párrafos siguientes.

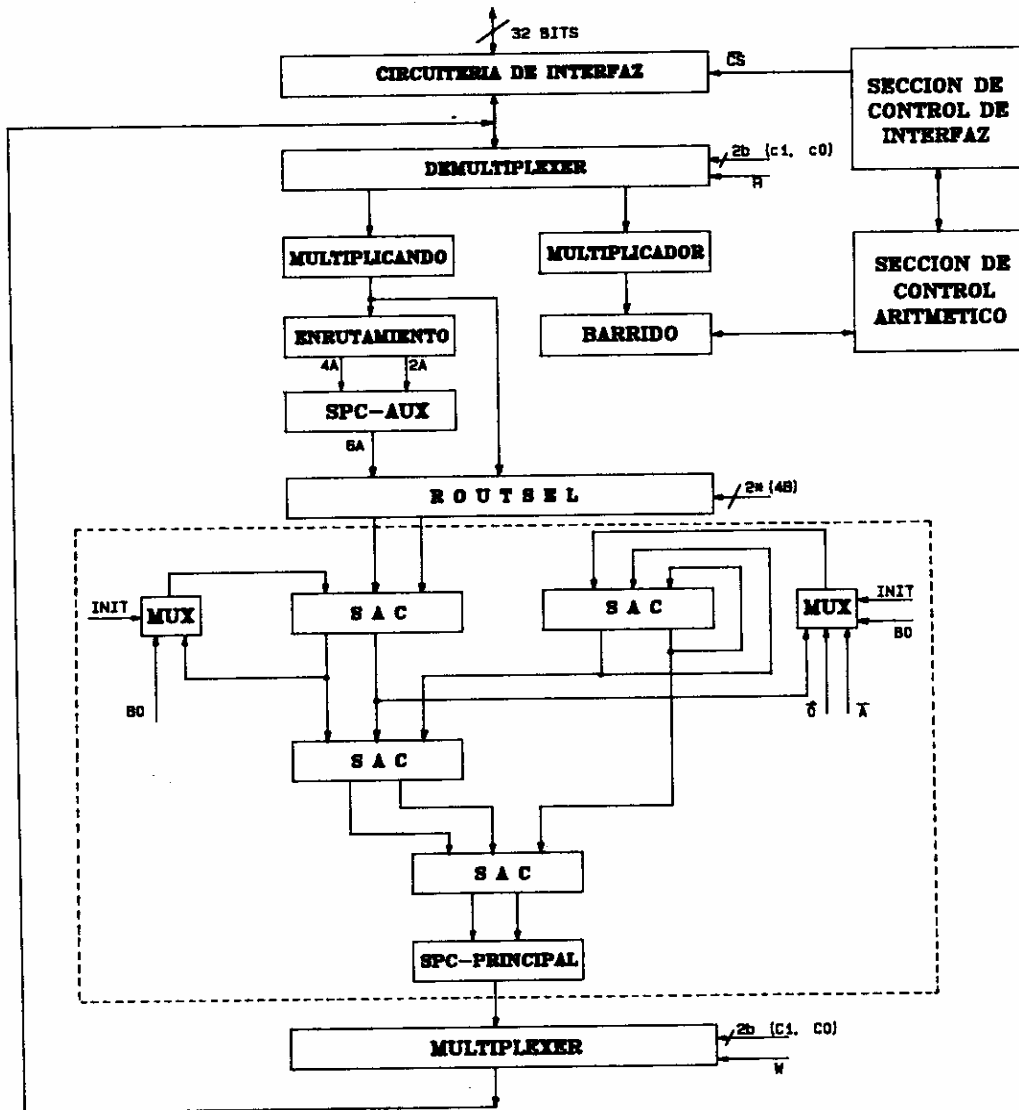


Fig. #3.- Arquitectura para Multiplicador de 64-bits.

Sección de Control de Interfaz.-

La sección de control de interfaz es responsable por las siguientes funciones:

- 1.- Determinar el momento en que el multiplicador ha sido seleccionado de acuerdo con las líneas de habilitación (CS) y dirección.
- 2.- Poner el dato en el bus del sistema (operación de lectura) y activar la señal de reconocimiento de transferencia de datos (DTACK*).
- 3.- Retener el dato proveniente del bus del sistema (operación de escritura) y activar la señal de reconocimiento de transferencia de datos (DTACK*).
- 4.- Una vez que recibe la indicación, proveniente de la sección de control aritmético, de que el producto se halla listo, debe activar la señal de pedido de interrupción (IRQ).
- 5.- Retener la señal de reconocimiento de la interrupción (IACKIN*) y desactivar (IACKOUT*).
- 6.- Activar la señal de que el multiplicador no se halla disponible (MPRBSY) de acuerdo con la señal de inicio de multiplicación (START) y la indicación proveniente de la sección de control aritmético.
- 7.- Indicar a la sección de control aritmético el momento de iniciar la multiplicación.

La sección de control aritmético es responsable por las siguientes funciones:

- 1.- Iniciar el proceso de multiplicación, una vez que recibe la indicación para hacerlo desde la sección de control de interfaz.

- 2.- Activar las señales de control para llevar a cabo la multiplicación.
- 3.- Indicar a la sección de control de interfaz el momento en que el producto se halla listo.

DISEÑO DE REDES LOGICAS

La etapa siguiente en el proceso de diseño es la implementación detallada de cada uno de los bloques definidos en la arquitectura del multiplicador de 64-bits mostrados en la Fig. #3.

A continuación se presenta, tan solo, el diseño de la sección de control aritmético, pues ésta constituye la parte medular de la arquitectura diseñada.

Diseño de la sección de control aritmético

El multiplicador utiliza los conceptos de paralelismo superpuesto de pipelining. El pipeline diseñado es del tipo unifuncional, estático y escalar.

El pipelining se lleva a cabo subdividiendo la tarea de multiplicación en una secuencia de subtareas, cada una de las cuales es ejecutada por una etapa de hardware especializada que opera concurrentemente con las otras etapas en el pipeline.

El diseño de la sección de control aritmético requiere de una definición clara de cada una de las subetapas del pipeline, pues su función es generar las señales de control para cada una de ellas.

Definición de las etapas de hardware del pipeline.-

La Tabla #7 muestra las etapas de hardware del pipeline, las subetapas que las conforman y las funciones que las etapas realizan.

Tabla #7: Etapas de Hardware del Pipeline		
Etapas de Hardware	Subetapas de Hardware	Función
S ₁	- Red de enrutamiento - SPC-AUX	Generación de 6A
S ₂	- Multiplexer 2 a 1 - Multiplexer 3 a 1	Realimentación de árbol sumador.
S ₃	- Multiplexer 11 a 1	Barrido del multiplicador B.
S ₄	- Par inversor - Red de enrutamiento - Multiplexer 16 a 1	Selección de múltiplos del multiplicando A.
S ₅	- SAC	Ejecución del primer nivel del árbol sumador.
S ₆	- SAC	Ejecución de segundo nivel del árbol sumador y llenado de huecos.
S ₇	- SAC - Red de enrutamiento (6 bits menos signif.)	Ejecución del tercer nivel del árbol sumador y llenado de huecos.
S ₈	- SPC-PRINCIPAL	Suma de vectores carry y suma finales.

La Tabla #8 es la tabla de reservación para el pipeline del multiplicador. Las filas: S₁, S₂, S₈ corresponden a las 8 etapas del pipeline y las columnas: t₁, t₂, t₃, t₁₈ a 18 unidades de tiempo. El número total de unidades de tiempo constituye el "tiempo de multiplicación".

Una X marcada en la celda (i,j) de la tabla indica que la etapa S_i será utilizada "j"

unidades de tiempo después del inicio de la multiplicación.

Marcas múltiples, en una columna, corresponden a la utilización simultánea de múltiples etapas del pipeline; mientras que, marcas múltiples en una fila corresponden a la utilización repetitiva o prolongada de una etapa determinada.

Tabla #8: Tabla de Reservación del Pipeline.

FUNCION	ETAPA	UNIDAD DE TIEMPO																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
GENERACION 6A	S ₁	X	X																
REALIMENTACION	S ₂			X	X	X	X	X	X	X	X	X	X	X					
BARRIDO DE B	S ₃		X	X	X	X	X	X	X	X	X	X	X						
SELECCION DE A	S ₄			X	X	X	X	X	X	X	X	X	X	X					
1ER NIVEL SAC	S ₅				X	X	X	X	X	X	X	X	X	X	X				
2DO NIVEL SAC	S ₆					X	X	X	X	X	X	X	X	X	X	X			
3ER NIVEL SAC	S ₇						X	X	X	X	X	X	X	X	X	X	X		
SPC-PRINCIPAL	S ₈																	X	X

Del análisis de la tabla de reservación pueden obtenerse las siguientes conclusiones:

- Una vez que se llena el pipeline, con cada unidad de tiempo se obtienen seis bits del vector del carry y seis bits del vector suma.
- El barrido del multiplicador B se realiza un ciclo de reloj en adelanto a la selección de los múltiplos del multiplicando A correspondientes, de tal manera que la información pertinente se halla disponible al inicio de ésta función. La información obtenida debe preservarse para la ejecución del primero, segundo y tercer niveles del árbol sumador. En consecuencia, se requieren cuatro retenedores para almacenar la misma información en instantes de tiempo distintos.
- La sección de control aritmético no es mas que un contador especial módulo 19. Cada una de las unidades de tiempo de la tabla de reservación corresponde a un estado distinto del contador.

PLAN DE PISO DEL CHIP

El multiplicador de 64-bits diseñado utiliza los principios de pipelining; por lo tanto, hay un flujo de datos que atraviesa una serie de etapas desde la entrada hacia la salida del chip. Consecuencia de esto es que los bloques funcionales de cada una de las etapas del pipeline pueden ser situados uno a continuación de otro, manteniendo una relación espacio-tiempo unidimensional.

La Fig. #4 muestra el arreglo de los subsistemas principales. Los buses que llevan el flujo de datos corren horizontalmente,

en el nivel de difusión, a través de éstos bloques funcionales.

Las principales líneas de control corren verticalmente a través de estos bloques, en el nivel de metal. Las líneas de alimentación, tierra y reloj corren en paralelo a las líneas de control.

Los drivers tres-estados y los pads constituyen el "marco" del multiplicador de 64-bits y se hallan situados en la periferia del chip.

La circuitería del "marco" es responsable del interfaz al bus de 32-bits del sistema.

DISEÑO DEL LAYOUT

La mayoría de las celdas básicas requeridas para la implementación de los circuitos de la sección de control mostrada en la Fig.#4 se encuentran en la "Biblioteca Stanford" instalada en SCALDSsystem y pueden utilizarse tal como están diseñadas o pueden ser ajustadas a las necesidades del diseño.

Una estructura clave que no se halla en la "Biblioteca Stanford" es el "multiplexer" y debió ser investigada y desarrollada; sin embargo, por motivos de extensión no se incluyen en el presente trabajo los resultados obtenidos [5].

La sección de control aritmética fue implementada en base a un PLA, cuyo layout fue generado, utilizando software desarrollado localmente en "Ohio University" [6] y software desarrollado por UW/NW VLSI Consortium [7]. El resultado se muestra en la Fig. #5.

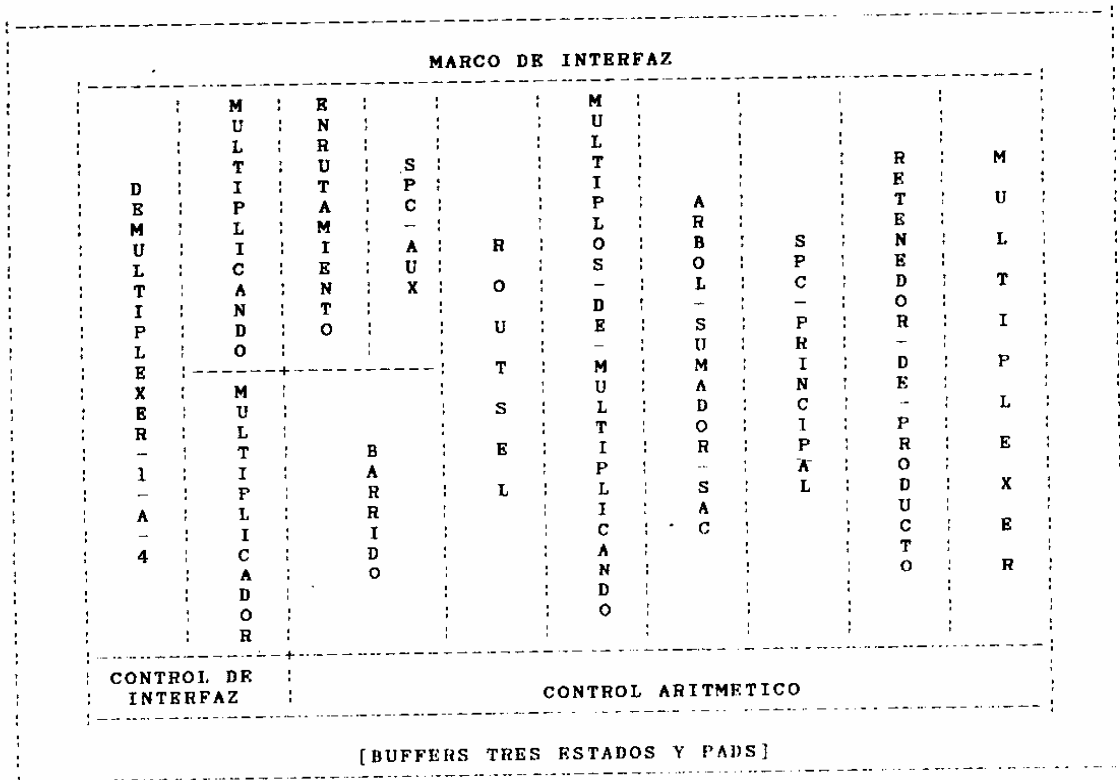


Fig. #4.- Plan de Piso del Multiplicador de 64-bits.

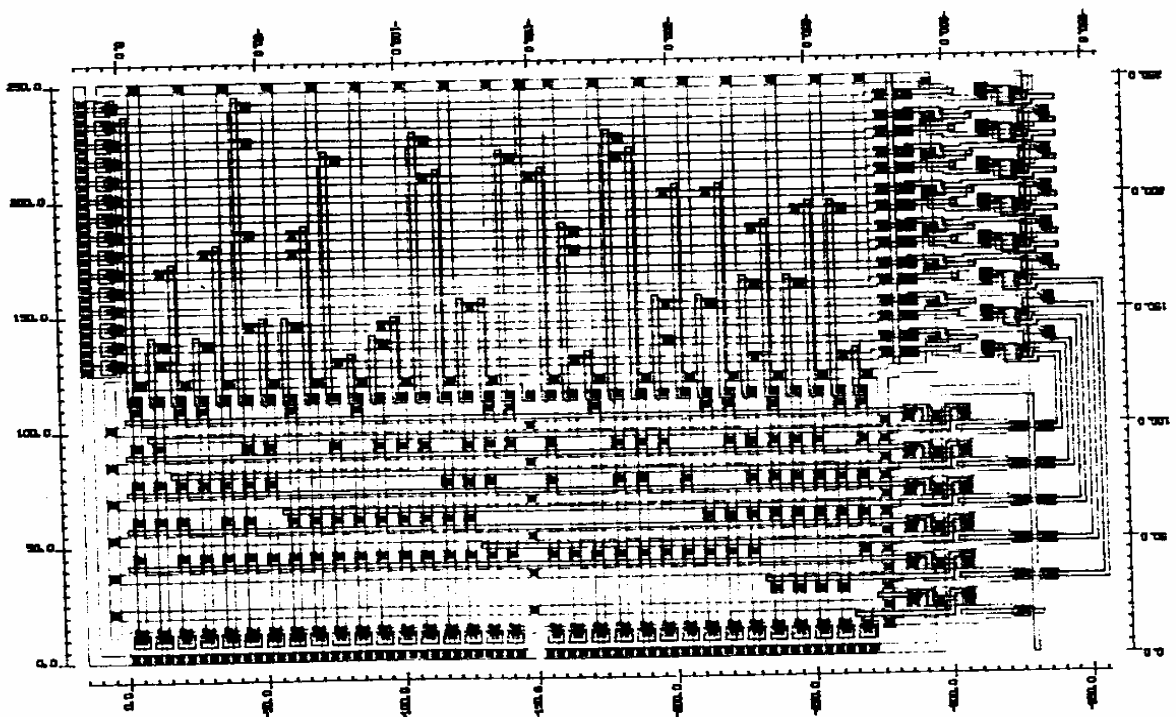


Fig. #5.- Layout del PLA de la Sección de Control Aritmético.

CONCLUSIONES

El presente trabajo de investigación ha abarcado todos los campos desde el desarrollo del algoritmo hasta el diseño del layout de un multiplicador de 64-bits, utilizando la técnica de barrido superpuesto de 3-bits.

Este trabajo ha demostrado que para realizar diseños eficientes en VLSI, todavía es necesario un conocimiento profundo en todos los niveles de la jerarquía de diseño. Es necesaria mayor investigación en el campo de "compiladores de silicio" de tal manera que el diseñador de sistemas pueda concentrarse en su campo específico sin tener que preocuparse de los detalles fastidiosos de la implementación del layout [8].

El autor ha hecho uso intensivo de paquetes de software para diseño VLSI disponibles en el Departamento de Ingeniería Eléctrica y Computación de Ohio University, tales como utilitarios para generación automática de layout, scripts desarrollados localmente en Ohio University, Biblioteca Stanford y SCALDSYSTEM [9], [10].

Evaluación del Diseño.

Para determinar el grado de éxito alcanzado por el diseño es necesario revisar los objetivos de diseño.

El objetivo 1 ha sido alcanzado plenamente, pues el proceso de fabricación nMOS depletion-load con contactos butting y $\lambda = 1 \mu\text{m}$ es ampliamente utilizado.

El objetivo 2 ha sido plenamente alcanzado ya que el número de pines que el multiplicador de 64-bits necesita es 45. Por lo tanto, el chip puede ser encapsulado en un DIP de 48 pines.

Finalmente, el objetivo 3 ha sido también plenamente alcanzado pues el multiplicador de 64-bits incluye la circuitería necesaria para conectarlo a un bus de sistema normalizado de 32-bits.

Trabajo Futuro.

En el futuro, tendrá que concluirse el diseño del layout de la sección aritmética y luego las secciones de control y aritmética tendrán que ser integradas en una unidad uniforme coherente.


Finalmente, el archivo CIF del multiplicador de 64-bits deberá enviarse para su fabricación y entonces deberá ingresarse en la fase de pruebas y correcciones del multiplicador de 64-bits diseñado.

ANEXOS

GLOSARIO DE TERMINOS

B	= Multiplicando.
n	= Multiplicador.
	= Número de bits del multiplicador.
SAC	= Sumador con acumulación de carry.
SPC	= Sumador con propagación de carry.
k	= Número de "1s" consecutivos en una cadena.
r	= Número de bits a barrer en cada grupo.
BTD	= Bus para transferencia de datos.
DTACK*	= Data Transfer Acknowledge.
IRQ	= Interrupt Request.
IACKIN*	= Interrupt Acknowledge In.
IACKOUT*	= Interrupt Acknowledge Out.
MPRBSY	= Multiplier Busy.
CS	= Chip Select.
V _{DD} , GND	= Alimentación eléctrica.
C _i	= Líneas de direccionamiento.
ROUTSEL	= Red de enrutamiento y selección de los múltiplos del multiplicando A.

BIOGRAFIA

 MONTALVO, LUIS.- Nació en Quito el 3 de Agosto de 1955. Obtuvo el título de Ingeniero en Electrónica y Telecomunicaciones en la Escuela Politécnica Nacional (1982). Becado por la Comisión Fulbright y auspiciado por la Escuela Politécnica Nacional obtuvo el título de MSc. in Computer Engineering en Ohio University, Athens, Ohio (1986). Fue asistente de cátedra en el laboratorio de Robótica en el departamento de Ingeniería Eléctrica y Computación en Ohio University (1984 - 1986). Desde 1981 hasta la fecha trabaja en el área de Sistemas Digitales en el Departamento de Electrónica y Telecomunicaciones de la E.P.N. Sus principales áreas de interés son: Microprocesadores, Diseño VLSI y Procesamiento de Imágenes.

REFERENCIAS

- [1] Raju, V. S., "Parallel Processing and VLSI Design: A High Speed Multiplier", Master Thesis, ECE Dept., Ohio University, November 1985.
- [2] MOS Integrated Circuits Group, Motorola Inc., "Motorola Microprocessors Data Manual", Austin, Texas, 1981.
- [3] Mead, C and Conway, L., "Introduction to VLSI Systems", Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1983.
- [4] Borril, P. L., "Micro Standards Special Feature: A comparison on 32-Bit Buses", IEEE Micro, December 1985.
- [5] Montalvo, L., "VLSI Implementation of Control Section of Overlapped 3-Bit Scanning 64-Bit Multiplier", Master Thesis, ECE Dept., Ohio University, September 1986.
- [6] Nobles, R., "Development of a Distributed Design System for Integrated Circuit Design Using VAX 11/750 and SCALDS-System Computers", Master Thesis, ECE Dept., Ohio University, January 1986.
- [7] UW/NW VLSI Consortium, "VLSI Design Tools Reference Manual, Release 1.0", Department of Computer Science, University of Washington, Seattle, Washington, 1983.
- [8] Denyer, P. and Renshaw, D., "VLSI Signal Processing: A Bit-Serial Approach", Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1985.
- [9] Newkirk, J. A. and Mathews, R., "The VLSI Designer's Library", Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1983.
- [10] Valid Logic Systems Inc., "SCALDS-System UNIX programmer's Manual", Mountain View, California, 1984.