

## Taxonomía de Multiplicadores

E.I. Boemo<sup>1</sup>, E. Juárez y J. Meneses

Dto. de Ingeniería Electrónica. ETSI Telecomunicación  
Universidad Politécnica de Madrid.  
Ciudad Universitaria. (28040) Madrid, España  
Tf: +34 1 5495762 Fax: +34 1 5439652  
e-mail: ivan@die.upm.es

### 1. Resumen

En este artículo se analizan más de 50 trabajos sobre multiplicadores binarios y se realiza una clasificación de los mismos según sus ideas algorítmicas y consecuencias arquitecturales, señalándose además los artículos que aportan nuevos conceptos. También se resumen las principales características de algunos prototipos realizados con circuitos VLSI y FPGAs.

### 2. Introducción

La popularización de los sistemas de diseño VLSI y la aparición de circuitos de lógica programable posibilitan actualmente una migración rápida y económica de algoritmos hacia hardware. Esto ha traído como consecuencia un renovado interés en temas relacionados con aritmética binaria, y en especial en la multiplicación, una de las operaciones más comunes en procesamiento digital de señal. Aunque existe una extraordinaria cantidad de artículos que exponen diferentes realizaciones de multiplicadores, las aportaciones fundamentales en este tema tienen origen en la década del '60, cuando los objetivos eran la estandarización de componentes digitales y el desarrollo de potentes unidades aritméticas para ordenadores. La mayoría de estos trabajos tienen aún una validez universal, que excede el marco tecnológico para el cual fueron pensados.

En el célebre artículo "*A Suggestion for a Fast Multiplier*" [1], Wallace señala los tres aspectos en los que trabajar para hacer más rápida la multiplicación de dos números:

- 1º Acelerar la formación de productos parciales.
- 2º Reducir el número de productos parciales.
- 3º Acelerar la suma de productos parciales.

(En la multiplicación de dos números enteros de  $n$  bits se denominará *producto parcial* al producto del multiplicando por un bit del multiplicador, y *producto entre bits* al resultado de multiplicar un bit del multiplicando por un bit del multiplicador).

La primer opción conduce a un *array* de puertas AND que realiza los  $N$  productos parciales simultáneamente. Uno de los primeros circuitos que implementa esta idea es el multiplicador a válvulas realizado por J. Rajchman en 1943 [2].

La segunda opción fue abordada por Booth en 1951 [3], aunque en la actualidad se usa una modificación [4] de ese algoritmo, llamado usualmente MBA (*Modified Booth Algorithm*). La idea es recodificar uno de los operandos para reducir el número de productos parciales. El operando que se recodifica (multiplicador) se divide en grupos de 3, 4, o más bits, solapados entre sí en un bit y que se convierten, de acuerdo a una tabla de verdad, a dígitos con signo que indican las transformaciones a realizar sobre el otro operando (multiplicando). La suma de las sucesivas transformaciones sobre el segundo operando da el producto. El algoritmo anterior también es válido para operandos codificados en complemento a 2. Aunque se han realizado estudios teóricos sobre esquemas que utilizan grupos de un mayor número de bits [5], dos razones hacen que sea preferible recodificar el operando dividiéndolo en grupos de 3 bits [6]:

---

<sup>1</sup> Dirección Actual:

E.T.S. Ingeniería Informática, Universidad Autónoma de Madrid  
Ctra. de Colmenar Km. 15, 28049 Madrid - España  
e-mail: Eduardo.Boemo@ii.uam.es  
<http://www.ii.uam.es/~ivan/>

1° Los dígitos recodificados dependen solo de un pequeño grupo de bits del operando que se recodifica.

2° Las transformaciones a realizar sobre el otro operando son simples (desplazar, complementar, etc.)

La idea de Booth presenta como contrapartida un aumento en la longitud de los productos parciales a sumar y una pérdida de regularidad.

Para acelerar la suma de productos parciales, la tercera de las sugerencias anteriores, el esquema más habitual es el Arbol de Wallace, formado por un conjunto de *full-adders* (también llamados sumadores *carry-save*, CSA o sumadores 3:2), donde el retardo de la suma es proporcional al logaritmo del número de sumandos. Un inconveniente de los árboles es su poca regularidad. Este problema fue resuelto en [7].

Otra forma de acelerar esta suma es el esquema de multiplicación propuesto por Dadda [8] en 1965. La idea es utilizar contadores paralelos para obtener la suma de los bits que forman las columnas de la matriz de productos parciales. Un contador paralelo es un circuito combinacional cuya salida es el número de 1's de su entrada. Aunque en principio se puedan utilizar contadores paralelos de un número grande de entradas es preferible, por razones de complejidad, limitarse al uso de contadores (3,2), aunque ello aumente el número de etapas necesarias [9]-[10] para conseguir la reducción.

Es interesante señalar la diferencia que existe entre los procedimientos que utilizan Wallace y Dadda para conseguir esta reducción: mientras que Wallace se fija en las filas de la matriz de productos parciales, Dadda lo hace en las columnas de la misma. Según esto, al esquema de Wallace lo podemos llamar "aproximación horizontal" al problema de la multiplicación y al de Dadda "aproximación vertical".

### 3. Clasificación según la estructura de palabra de los operandos.

Por estructura de palabra de los operandos se entiende la distribución espacio-temporal de los bits que forman el multiplicando y el multiplicador. Según se combine el paralelismo espacial con el paralelismo temporal (*pipelining*), se conseguirá diferentes distribuciones y por tanto, diferentes estructuras de palabra. Antes de continuar, es conveniente aclarar algunos términos. Se llamará *dígito* al grupo de bits que se transmiten a la vez en diferentes hilos (diferentes dígitos se transmiten uno a continuación del otro por el mismo grupo de hilos). Por *subpalabra* se entiende un grupo de dígitos. Diferentes subpalabras están separadas en el espacio

(ocupan diferentes grupos de hilos) y en el tiempo (aparecen en intervalos de tiempo no superpuestos).

De acuerdo con lo anterior, la estructura de una palabra de  $n$ -bits queda determinada por tres parámetros [11]: número de bits por dígito  $b$ , número de dígitos por subpalabra  $d$  y el número de subpalabras por palabra  $s$ . El producto de estos tres valores es igual al número total de bits de la palabra. Si ahora se agrupa los tres parámetros anteriores en el vector  $\langle b,d,s \rangle$  y se representa en un sistema de coordenadas logarítmico, la constelación de posibles estructuras estará sobre un triángulo equilátero. En la Fig.1 se representa el concepto anterior para una palabra de 8 bits.

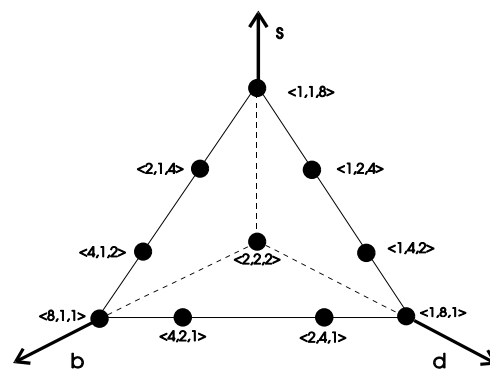


Fig.1: Constelación de estructuras para 8 bits

Observe que  $b$  y  $s$  hacen referencia a la dimensión espacial, y  $d$  a la dimensión temporal. El producto  $b \cdot s$  es igual al número total de hilos necesarios para la palabra.

Una palabra de 8 bits puede acomodarse en 8 hilos y ser procesada en un único ciclo de reloj; es decir su estructura es  $\langle 8,1,1 \rangle$ . Esta no es la única solución; también puede ocuparse un único hilo y procesarla en 8 ciclos de reloj, es decir  $\langle 1,8,1 \rangle$ . Otras soluciones intermedias entre ambos extremos son igualmente válidas. Más aun, si se utiliza un *pipeline* de 8 etapas, en un mismo instante estos 8 hilos estarán ocupados por 8 subpalabras correspondientes a distintos operandos. El vector será ahora  $\langle 1,1,8 \rangle$ . Los vértices del triángulo de la Fig.1 dan lugar a tres tipos muy frecuentes de multiplicadores:

**Bit-Serie:** su estructura genérica es la  $\langle 1,n,1 \rangle$ . El procesamiento es bit a bit. Minimiza el paralelismo espacial y en consecuencia presentan una gran economía en area pero velocidad limitada. Son ejemplos importantes de esta clase de multiplicadores los trabajos [12], [13], [14], [15], [16] y [17].

**Bit-Paralelo:** su estructura genérica es  $\langle n,1,1 \rangle$ . Corresponde a la mayoría de los array celulares combinacionales. Algunos ejemplos son [18] y [19].

**Bit-Paralelo Pipeline:** su estructura es la  $\langle 1,1,n \rangle$ . El procesamiento se realiza en  $n$ -bit. Alcanzan la máxima velocidad pero ocupan un área significativa. Maximizan el paralelismo espacial y temporal. Los ejemplos más importantes en esta categoría son [20], [21] y [22].

Otras estructuras de palabra de los operandos llevarán a multiplicadores con diferentes compromisos entre las variables área y tiempo.

#### 4. Clasificación algorítmica

Existen dos clases de multiplicadores según el algoritmo utilizado:

**Booth-Arbol:** en una primera etapa utilizan el algoritmo de recodificación de Booth para reducir el número de productos parciales a sumar. Después, mediante un árbol de *full-adders* (Wallace) o de contadores paralelos (Dadda), realizan la suma de los productos parciales. Estos árboles permiten el procesamiento en paralelo de  $n$  productos parciales para obtener, en una sola etapa, dos números. Algunos ejemplos son [7], [41] y [47].

**Iterativos:** utilizan repetidamente el algoritmo de desplazar y sumar (*shift-and-add*) para calcular el producto. Se construyen a partir de un procesador elemental que se repite para formar un *array* regular. A su vez, los multiplicadores iterativos se clasifican atendiendo a la forma en que se comunican los procesadores elementales del *array*. Podemos encontrar dos tipos de comunicaciones:

**Local:** solo existe comunicación entre procesadores vecinos. La maximización del paralelismo temporal (*pipelining*) de un *array* combinacional conduce directamente a un *array* sistólico. La transformación se puede hacer de forma sistemática, utilizando *Retimming* [23], o bien usando técnicas *ad hoc*. Un ejemplo es [20]. Algunos análisis sobre multiplicadores *pipeline* son [24], [25] y [26].

**Global (broadcast):** Existen conexiones comunes entre más de dos procesadores elementales. En general, estos circuitos combinan comunicación global y local. Ejemplos principales: [19], [22], [27] y [28].

#### 5. Clasificación según el sistema de representación numérica

De acuerdo con el sistema numérico utilizado para representar los operandos y el resultado, podemos clasificar los multiplicadores en :

**S+M (signo y magnitud):** trabajan en valor absoluto y calculan de forma separada el signo del resultado. Aprovechan al máximo la regularidad de la multiplicación. Un ejemplo es [29].

**Complemento a dos:** mediante el algoritmo de Baugh-Wooley [30] o modificaciones del mismo, es posible transformar un multiplicador iterativo para números sin signo, en un multiplicador que opere en complemento a dos. La aplicación de este algoritmo disminuye la regularidad del circuito. Otros algoritmos para operar en este formato son [31], [32] y [33].

**RNS (Residue Numbering System):** en este sistema de numeración [34], se representa un número entero,  $X$ , mediante los restos de la división entera de éste con un conjunto de números primos entre sí. Usando RNS es posible realizar la multiplicación sin tener que propagar ninguna información de acarreo entre los distintos dígitos. Con frecuencia estos multiplicadores se realizan con tablas de *look-up* (LUTs) [35], aunque también se pueden encontrar otras realizaciones [36].

**Signed-Digit:** este sistema de numeración explicado en [37] y [38] permite que cada uno de los dígitos de un número tenga asociado un signo positivo o negativo. Al no existir en *Signed-Digit* cadenas de propagación de acarreo, es posible calcular los dígitos más significativos del producto eliminando el cálculo de los menos significativos. Un buen ejemplo es [39].

#### 6. Implementaciones de Multiplicadores

En la Tabla I se resumen los resultados de algunas realizaciones de multiplicadores VLSI. En la Tabla II se muestran resultados de simulación de una prospección de multiplicadores en FPGAs.

#### 7. Agradecimientos

Este trabajo ha sido financiado por la CICYT de España a través del proyecto TIC92-0083.

REFERENCIA	TAMAÑO	TECNOLOGIA ( $\mu\text{m}$ )	AREA ( $\text{mm}^2$ )	CONSUMO (mW)	RETARDO (ns)	OBSERVACIONES
[40]	16x16	0.6 CMOS	2.4x3.2	400	7.4	MBA
[36]	8x8	2.5 CMOS	-	1130	50	MBA; Pipeline
[41]	16x16	0.5 CMOS	1.6x4.5	257	3.8	Wallace; CLA
[42]	4x4	1 AsGa	1.5x1.5	40	2.5	MBA
[43]	8x8	1.5 CMOS	-	21	30	MBA; CSA
[44]	16x16	2 CMOS	7.2	-	40	MBA
[22]	8x8	2.5 CMOS	-	250	14.3	CSA
[45]	4x4	2 NMOS	1.3x1.5	31.5	16	MBA
[46]	8x8	2 CMOS	-	40	18	Signed-Digit; MBA
[47]	8x8	3 CMOS	5.5x5.5	-	20	Dadda; Pipeline
[48]	16x16	1.5 CMOS	1.7x1.7	-	40	MBA; CSA
[49]	12x10	SERIE 74	-	-	520	CSA
[50]	32x32	2 CMOS	3.2x5.2	-	59	Signed-Digit; L. Multival.
[51]	8x8	1 NMOS	0.61x0.58	600	9.5	CSA
[39]	16x16	2.7 n-E/D MOS	5.8x6.3	-	120	Signed-Binary Digits
[28]	8x8	1 NMOS	2.1x2.1	1500	3	CSA; Pipelined
[7]	64x64	1.6 CMOS	3.8x6.5	-	47	Wallace modificado; MBA
[52]	16x16	1 CMOS	1.55x1.44	924	6.75	Parecido a Wallace; MBA

Tabla I: Resumen de características de multiplicadores VLSI.

REFERENCIA	TAMAÑO	MODELO	Nº CLBs	RETARDO (ns)	OBSERVACIONES
[53]	8x8	XC3090-100	235	26	[20]; CSA; Pipeline
[53]	8x8	XC4005-5	175	26	[19]; Pipeline
[53]	8x8	XC4005-5	162	26	Booth; Wallace; Pipeline
[53]	8x8	XC4005-5	158	24	CSA; CLA; Pipeline
[53]	8x8	XC4005-5	40	26ns/bit	Serie; [13]; Redondeo
[53]	8x8	XC4005-5	8	26ns/bit	Serie; [12]
[54]	16x16	XC4010-5	84	39	LUT; Pipeline
[53]	4x4	XC4005-5	42	28	[18]; Pipeline

Tabla II: Resumen de características de multiplicadores en FPGAs

## 8. Bibliografía

- [1] Wallace C.S. "A Suggestion for a Fast Multiplier". *IEEE Trans. on Electronic Computers*, pp.14-17, Feb. 1964.
- [2] R. Oberman, *Digital Circuits for Binary Arithmetic*, pp. 125, The Macmillan Press LTD, 1979.
- [3] A. Booth. "Assigned Multiplication Technique". *Quarterly J. of Mech. Appl. Math.* Vol.4, Part 2. 1951.
- [4] L. Rubinfeld, "A Proof of the Modified Booth's Algorithm for Multiplication", *IEEE Trans. on Computers*, pp. 1014-1015, Oct. 1975.
- [5] H. Sam y A. Gupta, "A Generalized Multibit Recording of Two's Complement Binary Numbers and its Proof with Application in Multiplier Implementations". *IEEE Trans. on Computers*, VOL. 39, pp. 1006-1014, Ago. 1989.
- [6] A. Cooper. "Parallel Architecture Modified Booth Multiplier". *IEE Proceedings*, Vol.135, Pt.G, N°3. Jun. 1988.
- [7] M. Santoro y M. Horowitz, "A Pipelined 64x64-bit Iterative Multiplier", *IEEE J.of Solid-State Circuits*, VOL. 24, n°2, pp.487-493, Abr. 1989.
- [8] L. Dadda. "Some Schemes for Parallel Multipliers". *Alta Frecuencia*, Vol.XXXIV, N°5. May 65.
- [9] P. Capello y K. Steiglitz. "A VLSI Layout for Pipelined Dadda Multiplier". *ACM Trans. on Computer Systems*, Vol.1, N°2, Mayo 1983.
- [10] A. Dhurkadas. "Fast Parallel Multiplier". *Proc. of the IEEE*, Vol.72, N°1, pp.134-136. Ene. 1984.
- [11] S. Smith y R. Morgan. "Generic ASIC Architecture and Synthesis Scheme for DSP". *Proc. ICASSP 89*, Glasgow, May. 1989.
- [12] S. Smith, J. Payne y R. Morgan. "Digit-Serial VLSI Microarchitecture". *Int. Conf. on Applic. Specific Array Processors*, 1990.
- [13] R. Lyon. "Two's Complement Pipeline Multiplier". *IEEE Transaction on Communications*, pp.418-425. Sep. 1989.
- [14] L. Dadda. "On Serial-Input Multipliers for Two's Complement Numbers". *IEEE Transaction on Computer*, Vol. 38, N°9, pp.1341-1345. Sep. 1989.
- [15] T. Rhyne y R. N. Strader, "A Signed Bit-Sequential Multiplier", *IEEE Trans. on Computers*, VOL. C-35, pp. 896-901, Oct. 1986.
- [16] I. Chen y R. Willoner. "An O(n) Parallel Multiplier with Bit-Sequential Input and Output". *IEEE Trans. on Computer*, Vol.C-28, N°10, pp.712-727. Oct. 1979.
- [17] D. Ait-Boudaoud, M. Ibrahim y B. Hayes-Gill. "Novel Pipelined Serial/Parallel Multiplier". *Electronic Letters*. Vol.26, N°9, pp.583-584. Abril 1992.
- [18] De Mori R. "Suggestion for an I.C. Fast Parallel Multiplier". *Electronic Letters*. Vol.5, N°3, pp.50-51. Feb. 1969.
- [19] H. Guild. "Fully Iterative Fast Array for Binary Multiplication and Addition". *Electronics Letters*. Vol.5, N°12. pp.263. Jun. 1969.
- [20] J. McCanny y J. McWhinter. "Completely iterative, pipelined multiplier array suitable for VLSI". *IEE Proc.* Vol.129, Pt. G, N°2. Abril 1982.
- [21] R. Roy y M. A. Bayoumi, "An Efficient Two's Complement Systolic Multiplier for Real-Time Digital Signal Processing", *IEEE Trans. on Circuit and Systems*, VOL. 36, pp. 1488-1493, Nov. 1989.
- [22] Hatamian M. and G.L.Cash. "A 70-MHz 8-bit x 8 bit Parallel Pipelined Multiplier in 2.5-um CMOS". *IEEE Journal of Solid-State Circuits*. August 1986.
- [23] C. Leiserson, F. Rose y J. Saxe. "Optimizing Synchronous Circuits by Retiming". Third Caltech Conference on VLSI. Pasadena, Ene. 1982.
- [24] Deverell J. "Pipeline Iterative Arithmetic Arrays". *IEEE Trans. on Computers*. Mar. 1975.
- [25] T. Hallin y M. Flynn. "Pipeline of Arithmetic Functions". *IEEE Trans. on Computer*, pp.880-886. Ago. 1972.
- [26] R. Jump y S. Ahuja, "Effective Pipeline of Digital Systems", *IEEE Trans. on Computers*, VOL. C-27, pp. 855-865, Sep. 1978.
- [27] I. Deegan. "Concise Cellular Array for Multiplication and Division". *Electronics Letters*. Vol.7, N°23. pp.702-703. Nov. 1971.
- [28] T. Noll, D. Schmitt-Landsiedel, H. Klar y G. Enders, "A Pipeline 330-MHz Multiplier", *IEEE Journal of Solid-State Circuits*, VOL. SC-21, pp. 411-416, Jun. 1986.
- [29] J. Arechabala, E.I. Boemo, J. Meneses, F. Moreno y C. Lopez-Barrio. "Full Systolic Binary Multiplier". *Proceeding of IEE*. Part G, Vol.139, N°2. Abril 1992.
- [30] C. Baugh y A. Wooley. "A Two's Complement Parallel Array Multiplication Algorithm". *IEEE Trans. on Computer*, Vol.C-22, N°12. Dic. 1973.

- [31] S. D. Pezaris, "A 40-ns 17-Bit by 17-Bit Array Multiplier", *IEEE Trans. on Computers*, pp. 442-447, Abr. 1971.
- [32] J. Gibson y R. Gibbardet. "Synthesis and Comparison of Two's Complement Parallel Multiplier". *IEEE Trans. on Computer*, Oct. 1975.
- [33] J. Majthia y R. Katai. "An Iterative Array for Multiplication of Signed Binary Numbers". *IEEE Trans. on Computer*, pp. 214-216. Feb. 1971.
- [34] F.J. Taylor, "Residue Arithmetic: A Tutorial with Examples", *Computer*, May 1984.
- [35] F.J. Taylor, "A VLSI Residue Arithmetic Multiplier", *IEEE Trans. on Computers*, VOL. C-31, pp. 540-546, Jun. 1982.
- [36] H. Razavi y J. Battelini. "Design of Residue Arithmetic Multiplier". *IEE Proceedings*, Vol.139, Pt.G, N°5. Oct 1992.
- [37] A. Avizienis. "Signet-Digit Number Representation for Fast Parallel Arithmetic". *IRE Trans. on Electronic Computers*. pp.389-400. Sep. 1961.
- [38] J. Morán. "Un Tutorial de Aritmética con Representación de Dígitos con Signo. Implementaciones Paralelas y Serie En-Linea". *Informe Técnico, Dto. Ingeniería Electrónica, ETSIT-UPM*. Madrid. 1993.
- [39] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa y N. Takagi. "A High-Speed Multiplier Using a Redundant Binary adder Tree". *IEEE Journal of Solid-State Circuits*, Vol.SC-22, N°1, pp.28-34, Abr. 1987.
- [40] Y. Oowaki et al. "A Sub-10-ns 16x16 Multiplier Using 0.6- $\mu$ m CMOS Technology". *IEEE J.of Solid-State Curcuits*, Vol.SC-22, N°5, pp. 762-767, Oct. 1987.
- [41] K. Yano, T.Yamanaka, T. Nishida, M. Saito, K. Shimohisagui y A. Shimizu. "A 3.8-ns CMOS 16x16 Multiplier using Complementary Pass-Transistor Logic". *IEEE J.of Solid-State Curcuits*, Vol.25, n°2, pp. 388-395, Abr. 1990.
- [42] E. Delhayé, C. Rocher, J. Baelde, J. Gibereau y M. Rocchi. "A 2.5-ns, 40-mW, 4x4 GaAs Multiplier in Two's Complement Mode". *IEEE Journal of Solid-State Circuits*, Vol.SC-22, N°3, pp.409-414. Jun. 1987.
- [43] S. Aziz, W. Waller y P. Lee. "A Compilable High-Speed Multiplier in Differential Split-Level CMOS". *Proceeding Second Eurochip Workshop on VLSI Design Training*. pp.122-127. Grenoble 1991.
- [44] J. Kernhof, M. Beunder, B. Hoefflinger y W. Hass. "High-Speed CMOS Adder and Multiplier Modules for DSP in a Semicustom Environment". *IEEE J. of Solid-State Circuits*, Vol.24, N°3, pp.570-575. Jun. 1989.
- [45] N. Shanbhag y P. Juneja, "Parallel Implementation of a 4x4-bit Multiplier Using Modified Booth's Algorithm", *IEEE J. of Solid-State Circuits*, Vol. 23, pp. 1010-1013, Ago. 1988.
- [46] W. Balakrishnan y N. Burgess. "Very-high-speed VLSI 2s-complement multiplier using signed binary digits". *IEE Proceedings-E*, Vol.139, Pt.E, N°1. Ene. 1992.
- [47] D. Crawley y G. Amaratunga. "8x8 Bit Pipelined Dadda Multiplier in CMOS". *IEE Proceedings*, Vol.135, Pt.G, N°6. Dic. 1988.
- [48] D. Henlin, M. Fertsch, M. Mazin y E. Lewis. "A 16 bit x 16 bit Pipelined Multiplier Macrocell". *IEEE Journal of Solid-State Circuits*, Vol.SC-20, N°2, pp.542-547. Abr. 1985.
- [49] A. Habibi y P. Wintz. "Fast Multipliers". *IEEE Trans. on Computers*, pp.153-157. Feb. 1970.
- [50] S. Kawahito, M. Kameyama y T. Higuchi. "Multiple-Valued Radix-2 Signed-Digit Arithmetic Circuits for High-Performance VLSI Systems". *IEEE J. of Solid-State Circuits*, Vol.25, N°1, pp.125-131. Feb. 1990.
- [51] J. Lee, H. Garvin y C. Slayman. "A High-Speed High-Density Silicon 8x8-bit Parallel Multiplier". *IEEE Journal of Solid-State Circuits*, Vol.SC-22, N°1. pp.35-40. Abr. 1985.
- [52] R. Sharma, A. López, J. Michejda, S. Hillenius, J. Andrews, A. Studwell, "A 6.75-ns 16x16-bit Multiplier in Single-Level-Metal CMOS technology", *IEEE J.of Solid-State Circuits*, Vol.24, n°4, pp. 922-927, Ago. 1988.
- [53] E. Juárez. "Análisis de Multiplicadores binarios Segmentados". Informe Interno LSI, Dto. Ingeniería Electrónica. ETSIT-UPM, 1993
- [54] K. Chapman. "Fast Integer Multipliers Using Xilinx FPGAs". *Comunicación Privada*. 1993.