IMPLEMENTACIÓN EN FPGA DE UN MULTIPLICADOR POR DIGITOS SOBRE CAMPOS FINITOS GF(2^m)

Víctor Carlos Tejeda-Calderón, Mario Alberto García-Martínez, Rubén Posada-Gómez
Instituto Tecnológico de Orizaba
División de Estudios de Postgrado
Orizaba, Veracruz
México
vctejeda@yahoo.com.mx, marioag1955@yahoo.com.mx

RESUMEN

En este artículo presentamos la implementación eficiente en FPGA (*Field Programmable Gate Array*) de un algoritmo para la multiplicación por dígitos sobre campos finitos *GF(2^m).* La multiplicación sobre campos finitos es una operación fundamental en áreas como la criptografía, los códigos de corrección de errores y el procesamiento digital de señales; así que se requiere de una multiplicación eficiente para mejorar el desempeño de diversos algoritmos utilizados en estas aplicaciones.
Las arquitecturas seriales para calcular la multiplicación presentan una buena eficiencia en espacio o área, pero su eficiencia en tiempo es pobre; en contraste, las arquitecturas paralelas realizan la operación en un solo ciclo de reloj, pero requieren de una gran cantidad de espacio. En la multiplicación por dígitos se procesan grupos de bits conocidos como *dígitos* en un solo ciclo de reloj. Esto nos permite operar en un rango de eficiencias de tiempo y de espacio según convenga, que dependerá del tamaño seleccionado del dígito. Hemos construido un multiplicador por dígitos que opera en el campo *GF(2^{239}),* que es un acampo recomendado por el NIST (*Nacional Institute of Standard Technology*) para criptosistemas de Curvas Elípticas (ECC).
Hacemos uso del paquete computacional ISE Versión 7.1i de Xilinx para el diseño. Se hace una descripción del circuito con VHDL (*Hardware Description Language*) y usamos la herramienta ModelSim para las simulaciones del multiplicador, el cual ha sido implementado en un FPGA Spartan 3 en una tarjeta prototipo de Digilent.

1. INTRODUCCIÓN

Los campos finitos o de Galois tienen numerosas aplicaciones en los sistemas de comunicación actuales, como la transmisión de datos o la grabación de discos compactos. Además, las operaciones aritméticas sobre estos campos se utilizan intensamente en aplicaciones como la criptografía, los códigos de corrección de errores y el procesamiento digital de señales. En tales aplicaciones se requiere de una alta velocidad de cómputo en el desarrollo de sus algoritmos.

Los campos finitos con *q* elementos se representan como *GF(q)* y los campos finitos de interés para aplicaciones computacionales son los campos de extensión de *GF(2)*, representados por *GF(2^m)*. La representación de los elementos del campo tiene una importancia fundamental en la eficiencia de las arquitecturas que realizan operaciones aritméticas sobre estos campos. Considerando *GF(2^m)* como un espacio vectorial sobre *GF(2),* se pueden seleccionar distintas bases de representación, y los elementos del campo se pueden ver como vectores de *m*-bits. En esta situación, la operación de suma es relativamente poco costosa, mientras que la multiplicación es la más importante y una de las más complejas. Existen distintas bases de representación de los elementos de *GF(2^m),* siendo las más populares la base polinomial o estándar [1], la base normal [2] y la base dual [3]. El uso de cierta base determina un tipo particular de algoritmo para la multiplicación, asociado con la complejidad en espacio y tiempo del circuito.
Una de las principales áreas de aplicación de los multiplicadores por dígitos sobre campos finitos *GF(2^m)* son los sistemas criptográficos, los que en la actualidad requieren cada vez más de un eficiente desempeño en velocidad, área, consumo de potencia y seguridad.
Generalmente las implementaciones realizadas en software de operadores aritméticos sobre campos finitos requieren de grandes recursos de cómputo y de grandes cantidades de memoria, lo que afecta el desempeño de un sistema de cómputo. Debido a esto recientemente encontramos en el estado del arte diversas propuestas de implementaciones en hardware de tales operadores [5], [6], [7], [8], [10].
Presentamos aquí la implementación de un multiplicador por dígitos que opera en el campo *GF(2^{239}),* que es un campo recomendado por el NIST (*Nacional Institute of Standard Technology*) para criptosistemas de Curvas Elípticas (ECC).
Usamos dispositivos reconfigurables como los FPGA´s, debido a su característica de reprogramación, lo que permite una mayor facilidad en el proceso de verificación y rediseño.

2. DESCRIPCIÓN DEL ALGORITMO

La multiplicación de campos finitos $GF(2^m)$ de dos elementos *A y B* para obtener un resultado *C= A*B mod p(α)* (donde *p(α)* es el polinomio irreducible) puede ser realizado en diferentes formas: serial, paralela o por dígitos. El algoritmo de multiplicación por dígitos introducido en [4] para campos binarios $GF(2^m)$, es eficiente en consumo de área, tiempo y potencia. Se procesan varios coeficientes del multiplicador *B* al mismo tiempo. El número de coeficientes que son procesados en paralelo es definido como el tamaño del *dígito*, y es definido como *D.* Denotamos *d* como el número total de dígitos con *d = [m/D].*

Sean:

$$A = \sum_{j=0}^{m-1} a_j \propto^j, \qquad B = \sum_{i=0}^{d-1} B_i \propto^{Di}, \qquad donde$$

$$B_i = \sum_{j=0}^{D-1} b_{Di+j} \propto^j \; 0 \le i \le d-1 \qquad (1)$$

$$C \equiv AB \bmod q(\propto) = A \sum_{i=0}^{d-1} B_i \propto^{Di} \bmod q(\propto) \qquad (2)$$

$$C \equiv AB \bmod q(\propto)$$

$$C \equiv [B_0 A + B_1 (A \propto^D \bmod q(\propto)) + B_2 (A \propto^D \propto^D \bmod q(\propto)) + \cdots + B_{d-1} (A \propto^{D(d-2)} \propto^D \bmod q(\propto))] \bmod q(\propto) \qquad (3)$$

Algoritmo de multiplicación por dígitos:

==========

Input: $A = \sum_{i=0}^{m-1} a_i \propto^i, donde \; a_i \in GF(2),$

$$B = \sum_{i=0}^{\lceil \frac{m}{D} \rceil - 1} B_i \propto^{Di},$$

$donde\ B_i\ es\ definido\ en\ la\ ecuación\ 1$

Output: $C \equiv A * B = \sum_{i=0}^{m-1} C_i \propto^i,$

donde $C_i \in GF(2)$

1: $C \leftarrow 0$

2: **for** $i = 0$ *to* $\lceil \frac{m}{D} \rceil - 1$ **do**

3: $C \leftarrow B_i A + C$

4: $A \leftarrow A \propto^D \bmod q(\propto)$

5: **end for**

6: Return $(C \bmod q(\propto))$

3. ARQUITECTURA DEL MULTIPLICADOR

La multiplicación por dígitos de *A (α) y B (α)* en el campo $GF(2^m)$ es una operación más compleja que la adición y requiere de tres pasos para su cálculo: una multiplicación polinomial, una operación de reducción principal módulo el polinomio irreducible y una operación de reducción final módulo el polinomio irreducible [4], [5].

La figura 1 muestra la arquitectura del multiplicador de dígitos serial/paralelo mapeada del algoritmo *LSD- First* [5]. A este tipo de arquitectura también se le llama multiplicador de acumulador simple (*SAM*) ya que utiliza una multiplicación polinomial que es el núcleo del multiplicador. Estas arquitecturas son comúnmente usadas en implementaciones de hardware para aplicaciones criptográficas.

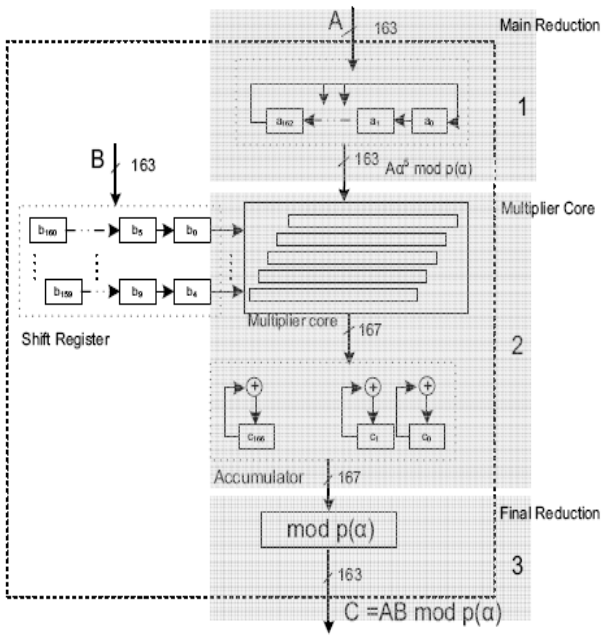

Figura 1. Arquitectura del multiplicador por dígitos usando un tamaño de dígito (D=5) para GF (2^{163}).

Como se puede observar, los polinomios de entrada *A y B* son de 163 bits, entran al núcleo del multiplicador donde se realizan los productos y sumas parciales, es decir, *C= Bi*A + C;* después se realiza la reducción principal *A= A*α^D mod p(α)* y finalmente una reducción final *C mod p (α)* [5].

El circuito de multiplicación polinomial (*núcleo del multiplicador*) calcula los resultados intermedios (sumas y productos parciales) y los almacena en el acumulador *C*. En esta operación se obtienen *m* columnas y *D* filas en cada ciclo de reloj. El circuito de *reducción principal* recorre *A* hacia la izquierda *D* posiciones para obtener la reducción módulo el polinomio irreducible (paso 4 del algoritmo). El circuito de *reducción final* reduce lo contenido en el acumulador para conseguir el resultado final en *C* (paso 6 del algoritmo).

4. IMPLEMENTACIÓN EN FPGA DEL MULTIPLICADOR POR DIGITOS

Todas las pruebas y mediciones se realizaron en una tarjeta de desarrollo de Digilent que contiene un FPGA SPARTAN3: XC3S200-FT256. El FPGA contiene 200,000 compuertas, 960 bloques de celdas lógicas (CLBs) y 1,920 slices.

4.1. Cálculos previos de complejidades de espacio y tiempo.

La complejidad de espacio en función del número de compuertas lógicas se muestra en la Tabla 1.

Tabla1. Complejidad de espacio el multiplicador por dígitos.

Polinomio Irreducible	Complejidad en área (Compuertas)
General	*(m+k)D* XORs + *(m+k+1)D* AND*s*

En la Tabla 2 se muestran los requerimientos de espacio para un campo finito $GF(2^{239})$ que utiliza como polinomio irreducible a $P(x)= x^{239} + x^5 +1$.

Tabla 2. Complejidad de espacio del multiplicador por dígitos que opera en un campo $GF(2^{239})$ para diferentes valores de *D*.

m	***D***	Complejidad de área	
		Compuertas	Slices
239	5	2445	1222
239	10	4890	2445
239	30	14670	7335
239	60	29340	14670

La complejidad en tiempo la hemos determinado previamente haciendo las siguientes consideraciones:

La frecuencia del FPGA que se maneja en la Tarjeta SPARTAN 3 es de *50 Mhz*.

Así entonces:

f = *50 Mhz*

T= 1 / 50 Mhz = *0.02 µseg en un ciclo de reloj*

T_{LSDE} = *0.02 µseg * Numero de ciclos de reloj (d= Total de dígitos ó ciclos de reloj)*

Donde: *d = m/D ciclos de reloj*

Ejemplo de tiempo usando el multiplicador por dígitos con un campo finito GF (2^{239}).

Multiplicador por dígitos con *m=239* bits y *D=20*

d=m/D = 12 dígitos o ciclos de reloj

T= 0.02 µseg * 12 = ***0.24 µseg***

La figura 2 muestra los tiempos de operación en microsegundos del multiplicador por dígitos en el campo $GF(2^{239})$ para diferentes valores de *D*. La frecuencia usada es de *50 Mhz*, es decir, un periodo de *0.02 µseg*.

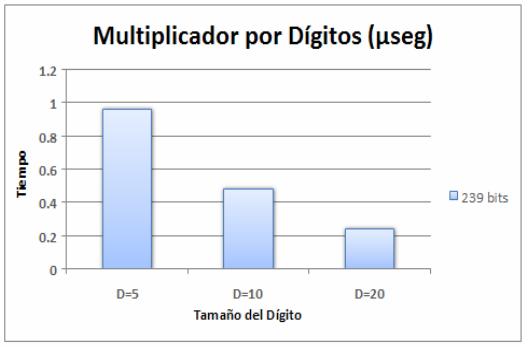

Figura 2. Complejidad de tiempo en microsegundos del multiplicador por dígitos para diferentes valores de *D*.

4.2. Resultados del proceso de síntesis del circuito.

La figura 3 muestra la complejidad en área (slices) reportada por el sintetizador, para diferentes valores de *D*. Se puede observar que para *m=239* bits se aprovecha casi en su totalidad el espacio del FPGA al usar un valor de *D=5,* que requiere de 1,918 slices de un total de 1,920 slices con que cuenta el FPGA.

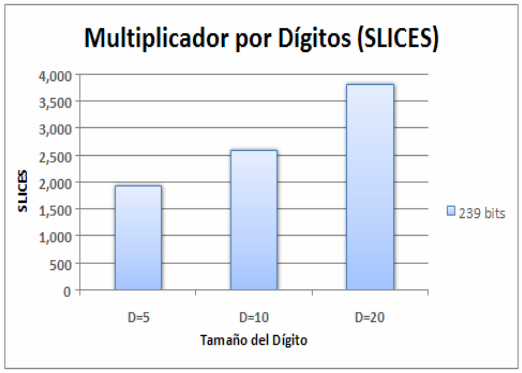

Figura 3. Complejidad de espacio (slices) que ocupa el multiplicador por dígitos en el FPGA

En la figura 4, se presenta la complejidad de tiempo en función del número de ciclos de reloj.

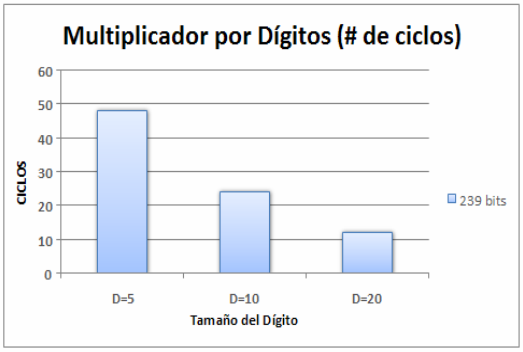

Figura 4. Complejidad de tiempo del multiplicador en ciclos de reloj para varios valores de *D*.

5. COMPARACIÓN CON OTRAS ARQUITECTURAS

Comparamos nuestro multiplicador con algunos multiplicadores propuestos en el estado del arte reciente [7], [8], [9].

Tabla 3. Tiempos de multiplicación en Software/Hardware reportadas en [7].

Implementación	**sw/hw**	***m***	**Mult.**	**Plataformas**
López (1999)	SW	162	10.5mS	UltraSparc a 300 Mhz
Savas (2000)	SW	160	18.3µS	Micro ARM a 80 Mhz
Rodríguez (2000)	SW	163	5.4µS	Pentium II a 450 Mhz
Rosner (1998)	HW	168	4.5mS	FPGA XC4052 a 16 Mhz
Orlando (1999)	HW	167	0.21mS	FPGA XCV400E a 76 Mhz
Lee (2000)	HW	192	2.88µS	*No implementado*
García (2004)	HW	239	3.1 µS	Virtex-300 (75 Mhz)

Tabla 4. Tiempos de multiplicación en arquitecturas por dígitos reportadas en [9].

m= 167	Digit Size	Clock (Mhz)	Montgomery (msec)
	4	85.7	0.55
	8	75.5	0.35
	16	76.7	0.21

Tabla 5. Complejidad de tiempo de multiplicadores por dígitos reportados en [8].

Digit Size D=16	Field *m*	Platform	Time (msec)
	155	VLSI 40 Mhz	3.9
	155	Xilinx FPGA XC4020XL, 15 Mhz	18.4
	113	Xilinx FPGA XCV300, 45 Mhz	3.7
	155	VLSI, 66 Mhz	5.7
	167	Xilinx FPGA XCV400E, 76.7 Mhz	0.21

Atendiendo a las tablas presentadas, podemos observar una mayor eficiencia en el tiempo de operación de nuestro multiplicador comparada con los resultados reportados en el estado del arte. El multiplicador por dígitos que hemos presentado calcula una multiplicación en un tiempo de *0.24 µseg* usando un dígito de tamaño *D=20* para un campo finito $GF(2^{239})$.

6. CONCLUSIONES

Hemos presentado la implementación en un FPGA Spartan3 de Xilinx, de un multiplicador por dígitos que opera en el campo $GF(2^{239})$ y que usa un polinomio irreducible $P(x)= x^{239} + x^5 + 1$, los cuales son valores sugeridos por el NIST para aplicaciones criptográficas de curvas elípticas ECC. Se ha mostrado que con la selección del tamaño del dígito *D*, se puede obtener una implementación eficiente en el FPGA considerando las complejidades de tiempo y espacio que se requieran para aplicaciones específicas. Una aplicación directa de nuestro multiplicador será la construcción de un criptoprocesador para curvas elípticas, los que requieren cada vez más de un eficiente desempeño en velocidad, área, consumo de potencia y seguridad.

7. REFERENCIAS

[1] E.D. Mastrovito, VLSI Architectures for Multiplication Over Finite Fields GF(2^m), Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, *Proc. Sixth Int Conf., AAECC-6*, New York: Springer-Verlag, Roma, pp. 297-309, Julio 1988.

[2] J. Omura y J. Massey, Computational Method and Apparatus for Finite Field Arithmetic, U.S. Patent Number 4,587,627, Mayo 1986.

[3] S.T.J. Fenn, M. Benaissa y D. Taylor, GF(2m) Multiplication and Division Over the Dual Basis, *IEEE Trans. Computers*, vol.45, no.3, pp. 319-327, Marzo 1996.

[4] L. Song, K. Parhi. Low_Energy Digit_Serial_Parallel Finite Field Multipliers. *Department of Electrical and Computer Engineering University of Minnesota,* Minneapolis. 1997.

[5] K. Kumar, T. Wollinger. Optimized Digit Multipliers for Elliptic Curve Cryptography. *Communication Security Group (COSY).* Ruhr-Universitaet Bochum, Germany, 2005.

[6] C. Paar. Reconfigurable Hardware in Modern Cryptography. Electrical and Computer Engineering Department and Computer Science Department. *Cryptography and Information Security Group*. Germany, 2006.

[7] M.A. García-Martínez. Construcción de operadores básicos sobre Campos Finitos *GF*(2^m). *Tesis doctoral*. Cinvestav, IPN. México D.F. Diciembre, 2004.

[8] G. Orlando. Efficient Elliptic Curve Processor Architectures for Field Programmable Logic. Electrical. *Engineering. Worcester Polytechnic Institute.* 2002

[9] C. Paar. Reconfigurable Hardware in Modern Cryptography. Cryptography and Information Security Group Electrical & Computer Engineering Dept. and Computer Science Dept. *Worcester Polytechnic Institute.* Worcester, MA, USA. Essen, Germany. 2006.

[10] E. Baz. Implementación en Hardware Reconfigurable de Multiplicadores sobre Campos Finitos GF(2^m). *Tesis de Maestría*. División de Estudios de Postgrado. Instituto Tecnológico de Orizaba. Diciembre. 2006.