JOURNAL DE CIENCIA E INGENIERÍA

Vol. 01, No. 1, Agosto de 2009, Páginas 58-61

Investigación

Implementación de Funciones Aritméticas en el cuerpo finito binario GF(2233) usando Lenguajes de Descripción Hardware

Fernando A. Urbano Molano[1] y Jaime Velasco Medina[2]

[1,2] Grupo de Bionanoelectrónica Escuela IEEE, Universidad del Valle, Santiago de Cali, Colombia

Recibido: 18 de Septiembre de 2007; Revisado: 29 de Noviembre de 2007; Aceptado: 26 de Abril de 2008

Resumen—En este artículo se presenta la implementación de funciones aritméticas, adición y cuadrado, así como su simulación funcional y la de la multiplicación en Bases Normales Óptimas (BNO), en el cuerpo finito binario GF(2^{233}). La arquitectura del hardware es implementada usando lenguajes de descripción hardare (VHDL) y sintetizada con la herramienta Quartus II de Altera ® sobre la FPGA (Field Programmable Gate Array) EP2S180F150814. Además se muestran los resultados de la simulación y los tiempos de operación.

Palabras Clave: FPGA, criptografía, VHDL, campos finitos, Matlab, Bases NormalesÓptimas (BNO).

I. INTRODUCCIÓN

El uso generalizado de las redes informáticas así como el aumento constante del número de usuarios de estos sistemas, han motivado la necesidad de mejorar la seguridad en el almacenamiento y transmisión de la información. Son muchas las situaciones donde se debe garantizar la privacidad, la integridad o la autenticación de la información almacenada o transmitida. Tales necesidades se han podido satisfacer mediante el uso de distintos algoritmos criptográficos, los cuales emplean criptosistemas de clave privada o de clave pública.

La seguridad de un criptosistema de clave pública reside entonces en problemas matemáticos que se suponen computacionalmente difíciles de resolver, es decir, problemas para los que no se conocen algoritmos recientes para resolverlos. Los criptosistemas basados en curvas elípticas son la mejor alternativa para implementar sistemas criptográficos de clave pública, debido al menor tamaño de las claves que se requieren, manteniendo la misma seguridad computacional y la gran cantidad de grupos que se presentan sobre el mismo cuerpo base. Estos criptosistemas son actualmente usados en la telefonía celular y en el comercio electrónico donde cada día son más las aplicaciones debido a las ventajas que poseen.

La seguridad de los sistemas de clave pública actuales se basan en la complejidad del cálculo de un problema matemático como lo es la factorización de números grandes o el cálculo de logaritmos discretos para enteros grandes. Para implementaciones en hardware es generalmente mas conveniente usar campos finitos binarios de característica dos, $GF(2^m)$ que un campo de enteros modulo un número primo grande. Esto se debe en gran medida a que su aritmética esta libre de acarreos, lo cual implica reducción del área en la implementación y un mejor desempeño [1].

II. CONCEPTOS MATEMÁTICOS

A. Definición de Grupo y Cuerpo

Un cuerpo finito es un conjunto finito de elementos con propiedades específicas. Para entender estas propiedades, es conveniente introducir el concepto de grupo. Un grupo es un conjunto *(G, +)* de elementos con una operación binaria '+' que satisface las siguientes propiedades:

1. Un grupo es cerrado bajo la operación +, si $a \in G$ y $b \in G$ entonces $a + b \in G$.
2. La operación + es asociativa: $(a + b) + c = a + (b + c)$.
3. El grupo contiene un elemento identidad $e \in G$ tal que $a + e = e + a = a$ para $a \in G$.
4. Todo elemento $a \in G$ tiene un inverso $a^{-1} \in G$ tal que $a + a^{-1} = a^{-1} + a = e$.
5. Un grupo es abeliano o conmutativo $a + b = b + a$, para $a, b \in G$.

Un cuerpo es un conjunto *F* de elementos con dos operaciones binarias, representadas como, adición (+) y multiplicación (*), que presentan las siguientes propiedades:

1. Los elementos de *F* forman un grupo abeliano bajo la operación (+).
2. Los elementos del conjunto *F**, el cual es un conjunto que contiene todos los elementos del conjunto *F* pero no cumple la identidad aditiva, forman un grupo abeliano bajo la operación (*).
3. La propiedad distributiva aplica a dos operaciones $a * (b + c) = (a * b) + (a * c)$, para $a, b, c \in G$.

Si el cuerpo tiene un conjunto de elementos finito, se llama un cuerpo finito o cuerpo de Galois y se representa por el símbolo *GF(q),* por sus siglas en inglés *Galois Field.* Los cuerpos finitos solo existen para $q = p^m$, donde *p* es un número primo y *m* es un entero positivo. El número de elementos de un cuerpo

finito es q. Cuando $q = 2^m$ para cualquier m, el cuerpo $GF(2^m)$ se llama cuerpo finito binario, si m es un entero positivo, el cuerpo finito binario $GF(2^m)$ consiste de los 2^m posibles cadenas de bits de tamaño m, donde el entero m se denomina grado del cuerpo. Los cuerpos finitos de característica 2 son atractivos debido a su aritmética libre de acarreos, y la disponibilidad de diferentes representaciones del cuerpo, los cuales pueden ser adaptados y optimizados para ambientes computacionales.

La representación de un elemento en el cuerpo finito $GF(2^m)$ usando bases normales presenta una ventaja computacional debido a que permite realizar el cálculo del cuadrado de un elemento de una manera eficiente. Sin embargo, multiplicar elementos distintos es un poco difícil [1,2].

B. Operaciones Aritméticas para Bases Normales Óptimas en GF(2^m)

El parámetro *tipo T* de una BNO es un entero positivo, el cual mide la complejidad de la multiplicación con respecto a la base. Generalmente, el parámetro *tipo T* de menor valor permite realizar una multiplicación más eficiente. Se puede demostrar que para un número entero que define el cuerpo finito m y un parámetro T dado, el cuerpo finito $GF(2^m)$ puede tener al menos una BNO. Sea m un entero positivo no divisible por 8 y el tipo T un entero positivo, entonces el *tipo T* de una BNO en el cuerpo finito $GF(2^m)$ existe si y solo si $p = Tm + 1$ es primo.

Una base normal para $GF(2^m)$ es de la forma:

$$\left\{b, b^2, b^{2^2}, ..., b^{2^{m-1}}\right\} \text{ donde } \beta \in GF(2^m),$$

en donde, cualquier elemento $\alpha \in GF(2^m)$ puede ser escrito de la forma:

$$a = \sum_{i=0}^{m-1} a_i b^{2^i} \text{ donde } a_i \in \{0,1\}$$

Si $\left\{b, b^2, b^{2^2}, ..., b^{2^{m-1}}\right\}$ es una base normal en el cuerpo finito $GF(2^m)$, entonces el elemento $a = \sum_{i=0}^{m-1} a_i b^{2^i}$ es representado por la cadena binaria $(a_0, a_1, a_2, ..., a_{m-1})$ donde $a_i \in \{0,1\}$.

En este caso, la identidad multiplicativa es representada por una cadena de unos, mientras que la identidad aditiva es representada por una cadena de ceros. Un concepto importante para la aritmética de las bases normales es el teorema de Fermat. Para todo $\beta \in GF(2^m)$ se tiene:

$$b^{2^m} = b$$

Este teorema es importante para realizar el cuadrado de un elemento en el cuerpo finito $GF(2^m)$. El cuadrado es una simple rotación.

Las siguientes operaciones aritméticas se definen sobre los elementos de un cuerpo finito $GF(2^{233})$ cuando se usa una BNO:

***Adición*:**

Si $a = (a_0 a_1 a_2 .... a_{m-1})$ y $b = (b_0 b_1 b_2 .... b_{m-1})$ son elementos de $GF(2^m)$, entonces

$a + b = c = (c_0 c_1 c_2 .... c_{m-1})$ donde $c_i = (a_i \approx b_i)$

***Cuadrado*:**

Sea $a = (a_0 a_1 a_2 .... a_{m-1}) \in GF(2^m)$, entonces

$$a^2 = \left(\sum_{i=0}^{m-1} a_i b^{2^i}\right)^2 = \sum_{i=0}^{m-1} a_{i-1} b^{2^i} = (a_{m-1} a_0 a_1 .... a_{m-2})$$

En este caso, el cuadrado es una simple rotación a la derecha de la representación del vector.

Multiplicación en Bases Normales Óptimas:

Sea A y B dos elementos de $GF(2^m)$ y C su producto. Entonces,

$$c = \begin{cases} \sum_{i=0}^{m-1} a_i b_i \beta^{2^i} + \sum_{j=1}^{v} \sum_{k=1}^{h_j} \left(\sum_{i=0}^{m-1} y\left((i - w_{j,k})\right), j^{\beta^{2^i}} \right) & \text{para } m \text{ impar} \\ \sum_{i=0}^{m-1} a_i b_i \beta^{2^i} + \sum_{j=1}^{v-1} \sum_{k=1}^{h_j} \left(\sum_{i=0}^{m-1} y\left((i - w_{j,k})\right), j^{\beta^{2^i}} \right) + F & \text{para } m \text{ par} \end{cases}$$

Donde

$$F = \sum_{k=1}^{\frac{h_v}{2}} \sum_{i=0}^{v-1} y\left((i - w_{v,k})\right), v\left(\beta^{2^i} + \beta^{2^{i+v}}\right) \text{ y } v = \frac{m}{2}.$$

Note que, para una base normal, la representación de δ_j se fija y también para $w_{j,k}$, $1 \le j \le v$, $1 \le k \le h_j$, lo anterior es válido para cualquier base normal de $GF(2^m)$ sobre GF(2) [3].

Inversión:

Si $a \neq 0$ y $a \in GF(2^m)$, el inverso multiplicativo del elemento a denotado como a^{-1} es el único elemento $c \in GF(2^m)$ para lo cual $a \bullet c = 1$ [1].

$$a^{-1} = a^{2^n - 2} = \left(a^{2^{n-1} - 1}\right)^2$$

III. SIMULACIONES FUNCIONALES

A partir de análisis de las funciones aritméticas, se efectuó la simulación a nivel funcional de los algoritmos mencionados en el punto anterior, con el objetivo de ayudar a visualizar el comportamiento de las variables que intervienen en ellas. Las simulaciones de los algoritmos se realizaron mediante el programa de simulación Matlab.

A. Adición

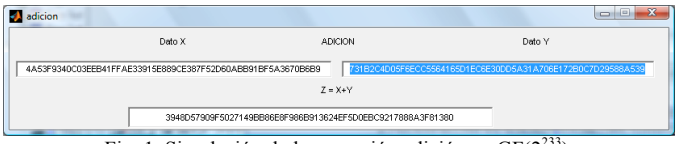

Fig. 1. Simulación de la operación adición en GF(2^{233}).

B. Cuadrado

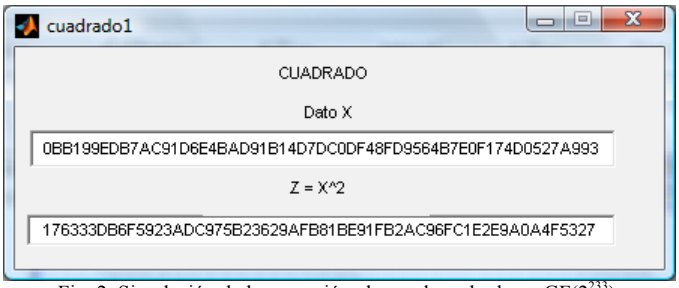

Fig. 2. Simulación de la operación elevar al cuadrado en GF(2^{233}).

C. Multiplicación en Bases Normales Óptimas

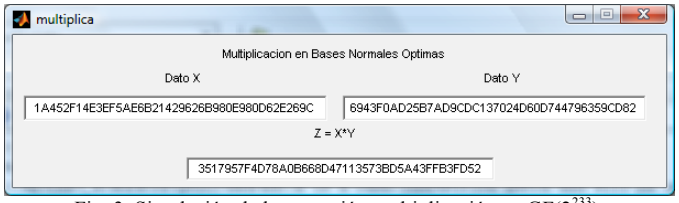

Fig. 3. Simulación de la operación multiplicación en GF(2^{233}).

D. Inversión

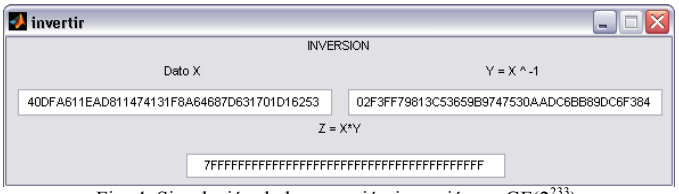

Fig. 4. Simulación de la operación inversión en GF(2^{233}).

A partir de las simulaciones realizadas, se comprobó el funcionamiento de las operaciones de adición y elevar al cuadrado, además se visualizó de una manera más adecuada el algoritmo de multiplicación que actualmente se encuentra en etapa de implementación en hardware.

IV. IMPLEMENTACIÓN DE LA ADICIÓN Y EL CUADRADO SOBRE GF(2^{233})

En esta sección se presenta la implementación en hardware de las funciones aritméticas en adición y elevar al cuadrado. Para el caso de la adición la arquitectura presenta un bloque funcional implementado con compuertas XOR con dos entradas sobre GF(2^{233}), funcionando de manera paralela, el diseño se basó en el bloque sumador descrito en [4].

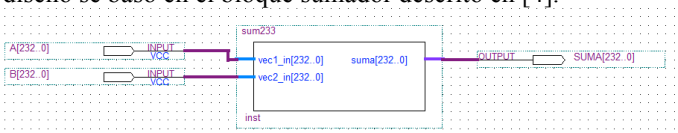

Fig. 5. Bloque funcional para la operación adición en GF(2^{233}).

De la misma manera se diseñó el bloque funcional para la operación elevar al cuadrado con el fin de realizar la operación en un solo ciclo de reloj, la arquitectura simplemente consiste en rotar a la izquierda el bit más significativo y concatenarlo con el bit de menor peso.

V. RESULTADOS DE SIMULACIÓN

Con el propósito de verificar el funcionamiento de la adición y el cuadrado se realizaron las siguientes simulaciones en el cuerpo finito GF(2^{233}) y se compararon con las funcionales de la sección anterior.

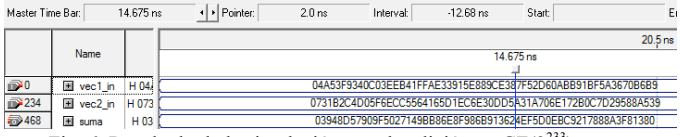

Fig. 6. Resultado de la simulación para la adición en GF(2^{233}).

El costo del hardware es muy pequeño en comparación al tamaño de la FPGA, 233 ALUTs con un porcentaje inferior al 1% y un tiempo de operación de 10.113 ns.

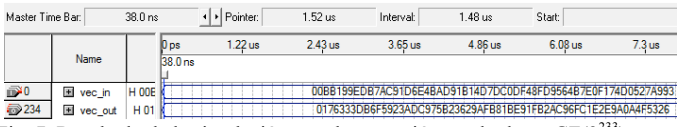

Fig. 7. Resultado de la simulación para la operación cuadrado en GF(2^{233}).

El costo del hardware para la operación cuadrado, también es muy pequeño si se tiene en cuenta que únicamente se trabajó con los bits y su ubicación, sin involucrar al hardware. Se obtuvo un tiempo de operación de 8.278 ns.

VI. CONCLUSIONES

En éste artículo se realizó la simulación a nivel funcional utilizando Matlab, de las funciones aritméticas en el campo finito GF(2^{233}) usando Bases Normales Óptimas. En este trabajo se presentan las implementaciones de la adición y el cuadrado, optimizados en velocidad y área y con el objetivo de utilizar la menor cantidad de recursos hardware.

Las simulaciones funcionales además de facilitar la visualización del comportamiento de las variables que intervienen en los algoritmos, permiten la comprobación de los resultados de las implementaciones hardware.

Como trabajo futuro se pretende implementar la multiplicación en bases Normales Óptimas propuesta en [3] y [5].

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a Paulo Realpe y Vladimir Trujillo, del Grupo Bionanoelectrónica, EIEE, Universidad del Valle, por su constante apoyo y orientación.

REFERENCIAS

[1] V. Trujillo-Olaya, J. Velasco-Medina y J. López. *Design of an Elliptic Curve Cryptoprocessor over GF(2^{163})*. Iberchip. 2005. Disponible en Internet: http://www.iberchip.org/.../articles/78/78--jvelasco-DESIGN%20OF%20AN%20ELLIPTIC%20CURVE%20**CRYPTOPROCESSOR**.pdf.

[2] J. M. Cruz y F. Rodríguez. *Multiplicación Escalar en Curvas de Koblitz. Arquitectura en Hardware Reconfigurable*. Sección de Computación, Departamento de Ingeniera Eléctrica. Instituto Politécnico Nacional. México D.F.

[3] A. Reyhani-Masoleh y A. Hasan. *Efficient Multiplicacion Beyond Optimal Normal Bases*. En: IEEE Transactions on Computers. Vol. 52, No. 4. Abril 2003.

[4] P. Realpe-Muñoz, V. Trujillo-Olaya y J. Velasco-Medina. Implementación de un Multiplicador Paralelo a Nivel de Digito sobre GF(2^{163}) usando Bases Normales Gaussianas. Grupo de Bionanoelectrónica, EIEE, Universidad del Valle, Iberchip 2007, Lima, Perú. Disponible en: http://www.iberchip.org/iberchip2007/articulos/4/c/paper/3--Implementacion_Multiplicador.pdf

[5] A. Reyhani-Masoleh y A. Hasan. *A New Construction of Massey-Omura Parallel Multiplier over GF(2^m)*. En: IEEE Transactions on Computers. Vol. 51, No. 5. Mayo 2002.

Fernando Aparicio Urbano Molano: Ingeniero Físico y Diplomado em Docencia Universitaria, Universidad del Cauca, candidato a Magister en Ingeniería con énfasis en Electrónica, Universidad del Valle. Area de investigación: Diseño Digital, criptoprocesadores, procesadores DSP, Robótica Móvil y control inteligente. Actualmente se desempeña como profesor catedrático de la Corporación Universitaria Autónoma del Cauca. E-mail: faurbano@gmail.com

Jaime Velasco Medina: Profesor titular Universidad del Valle, director del grupo de Bionanoelectrónica. Líneas de investigación: Criptografia, Bionanoelectronica. Diseño de Sistemas Digitales Avanzados Basados en FPGA. Diseño de Equipos Biomédicos, Diseño de Equipos para Instrumentación Industrial, Sistemas Tolerantes a Fallas.