IMPLEMENTACIÓN DE LA FUNCIÓN INVERSO MULTIPLICATIVO MODULAR EN HARDWARE REPROGRAMABLE

Carmen Rivera R..,Rubén Darío Nieto Londoño, Álvaro Bernal N

Escuela de Ingeniería Eléctrica y Electrónica
Grupo de Arquitecturas Digitales y Microelectrónica
Universidad del Valle
Cali - Colombia

cluciari@hotmail.com
alvaro@mafalda.univalle.edu.co
rnieto@eiee.univalle.edu.co

ABSTRACT

Digital communications are widely involved in several activities of the modern life. Exchange of the information requires secure systems. Cryptography concept applied to keep and transferring secure information is mandatory. Several cryptographic protocols like RSA are based on modular exponentiation function. Existence verification the of the modular multiplicative inverse is required to use this algorithm and consequently how to calculate it. In this paper hardware to calculate the modular inverse factor is presented. The hardware is based on Euclides and Binary algorithms. Simulation results for 8-bits operands using a programmable logical device (CPLD) were obtained.

RESUMEN

La penetración de las comunicaciones en diversos actividades cotidianas ha evolucionado conjuntamente con la criptografía a fin de proteger el intercambio de información electrónica cada vez mas elevado. Diversos protocolos criptográficos usan ampliamente la aritmética modular y fundamentan su aplicación en la existencia del inverso multiplicativo modular de un entero tanto para la obtención de claves, como para la selección de los módulos a ser utilizados en el algoritmo. En este artículo se presenta una concepción de hardware que permite calcular el inverso multiplicativo modular basándose en los algoritmos de Euclides y Binario para n bits. Se obtuvieron resultados experimentales para operandos de 8 bits, en un dispositivo lógico programable, específicamente un CPLD.

IMPLEMENTACIÓN DE LA FUNCIÓN INVERSO MULTIPLICATIVO MODULAR EN HARDWARE REPROGRAMABLE

Carmen Rivera, Rubén Nieto Londoño, Álvaro Bernal N.

Escuela de Ingeniería Eléctrica y Electrónica
Grupo de Arquitecturas Digitales y Microelectrónica
Universidad del Valle
Cali - Colombia

cluciari@hotmail.com , alvaro@mafalda.univalle.edu.co, rnieto@eiee.univalle.edu.co

RESUMEN

En este artículo se presentan dos arquitecturas que permiten calcular el inverso multiplicativo modular basándose en los algoritmos de Euclides y Binario para *n* bits. Se obtuvieron resultados para operandos de 8 bits, en un CPLD.

1. INTRODUCCIÓN

Las multiplicaciones modulares se pueden implementar usando diferentes tipos de algoritmos, uno de los mas usados es el algoritmo de Montgomery[1]. La aplicación de éste algoritmo requiere de la existencia del *inverso multiplicativo modular*, por esta razón se hace necesario no solamente verificar la existencia del inverso multiplicativo modular sino también realizar su cálculo.

El propósito del trabajo es concebir un hardware que permita el cálculo del inverso multiplicativo modular de un número entero, que puede ser utilizado como parte de la implementación del protocolo criptográfico RSA[2] para calcular la clave privada.

2. ALGORITMOS

Los algoritmos utilizados para calcular el inverso multiplicativo modular, son el algoritmo de Euclides [1] y el algoritmo Binario[3]. Los cuales se presentan en a continuación.

2.1. El algoritmo de Euclides modificado

no ← **m**
bo ← **b**
to ← 0
t ← 1
q ← no / bo
r ← no –q* bo
while r > 0 do
temp ← to – q * **t**
if temp > 0 do temp ← temp mod **m**
if temp < 0 do temp ← **m** – ((-temp) mod **m**
to ← t
t ← temp
no ← bo
bo ← r
q ← no / bo
r ← no – q*bo
if bo ≠ 1 then
el inverso multiplicativo de b no existe
if bo = 1 then

$$b^{-1} \bmod n = t$$

2.2. El algoritmo binario

g ← 1
while **m** y **b** are even do
m ← **n**/2, **b** ← **b**/2, g ←2g
while **m** > 0 do
while **m** is even do **m** ←**n**/2
while **b** is even do **b** ← **b**/2
t ← | **m** – **b** | / 2
if **m** ≥ **b** then
m ← t
else **b** ← t
p ← (g * **b**)
if p ≠ 1 el inverso multiplicativo de b no existe
if p = 1 el inverso multiplicativo es
t = **m** – **m** / **b**

Donde **m** es el módulo, **b** el número al cual se desea obtener el inverso multiplicativo modular y **t** el resultado.

3. IMPLEMENTACIÓN HARDWARE

Para la implementación y optimización del hardware se utilizaron técnicas como las de Wallace [4] y

Booth [5] para mejorar los procesos de multiplicación. Para la división fueron usados los métodos de multiplicación y sustracción [6][7].

El hardware que permite calcular el inverso multiplicativo modular mediante los algoritmos de Euclides y binario aparecen en las figuras 1 y 2 respectivamente.

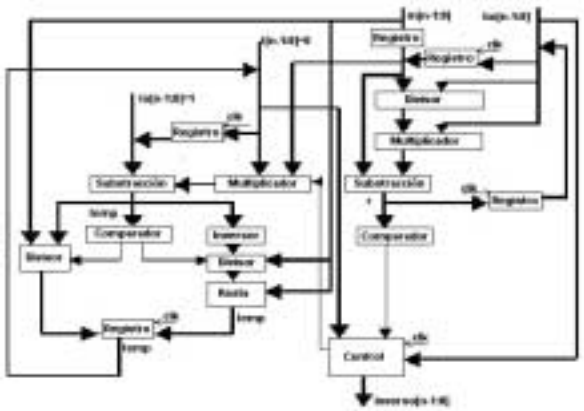

Figura 1. Diagrama de bloques

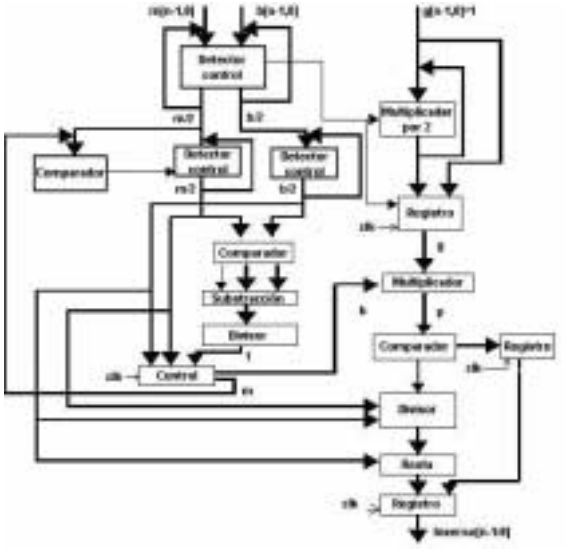

Figura 2. Diagrama de bloques

En ambos casos **m** y **b**, son el módulo y el número al que se quiere obtener el inverso multiplicativo modular. Al finalizar el proceso se obtendrá el inverso multiplicativo modular en caso que exista.

4. RESULTADOS

La tabla 1 ilustra los resultados obtenidos para los prototipos de 8 bits en el CPLD EP1K100FC484-2 de Altera que posee 4992 celdas lógicas. Para la simulación se usó el software MAX+plus II 10.1 de Altera [8]. La comparación se hizo considerando el tiempo de respuesta (**Tr**), el tiempo de máximo retardo (**Tm**) y el número de celdas lógicas (**Nc**) utilizados en cada implementación.

Tabla 1. Simulaciones para números de 8 bits

Algoritmo	**Tr**	**Tm[ns]**	**Nc**
Euclides	600ns	7.6	1262
Binario	2.5µs	7.5	818

En las figuras 3 y 4 se muestran lo resultados de simulación para los algoritmos Binario t de Euclides respectivamente; **w** y **l** corresponden a los datos de entrada, **w** es el modulo y **l** el número al que se le desea calcular el inverso multiplicativo modular, la salida es el inverso multiplicativo modular.

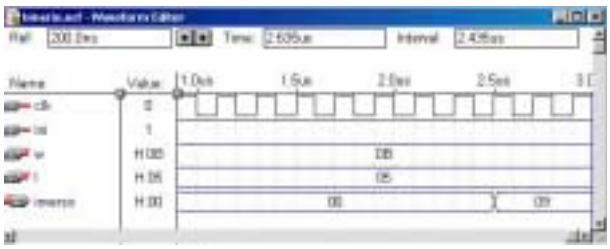

Figura 3. Simulación del algoritmo Binario

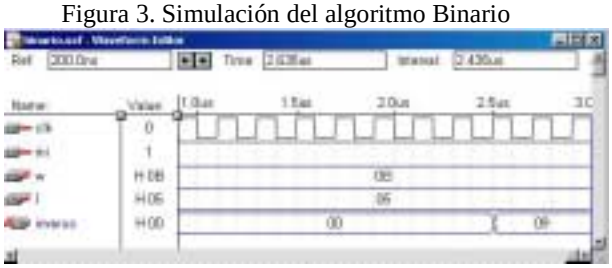

Figura 4. Simulación del algoritmo de Euclides

5. CONCLUSIONES

La codificación del algoritmo de Euclides en lenguaje de descripción de hardware requiere de mayor cuidado que la correspondiente al algoritmo binario. La arquitectura de los algoritmos es similar, ambos utilizan divisores, multiplicadores, registros, restadores, comparadores. La división es la operación aritmética que mayor hardware requiere, siendo éste un factor limitante en el desempeño de las arquitecturas.

Las simulaciones muestran que el algoritmo de Euclides es mas rápido, pero ocupa mayor área, mientras que el algoritmo Binario es lento pero ocupa menos área. Es difícil lograr en un diseño rapidez y poca área utilizada, estos parámetros son excluyentes entre si.

Dada la modularidad de la arquitectura, ésta se puede expandir fácilmente a operandos de mayor número de bits.

6. BIBLIOGRAFÍA

[1] A. Menezes, P. Van Oorschot and S. Vanstone. Hsndbook of Applied cryptography.

[2]http://www.terra.es/personal6/morenocerro2/seguridad/cripto/cripto_10.html

[3] http://www.cacr.math.uwaterloo.ca.2002

[4] Wallace C.S. "A Suggestion for a Fast Multiplier". *IEEE Trans. on Electronic Computers*, pp.14-17, Feb. 1964.

[5] Booth, A.D., "A signed Binary Multiplication Technique" in J. Mech. Appl. Math., Vol 4, Part 2, 1951.

[6] www.cse.iitd.ac.in

[7] Wilson, J. B., and Ledley, R. S., "An algorithm for Rapid Binary Division" in Ire Transactions o Electronic Computer, Ec-10, 1961. Pp662-670.

[8] Altera Digital Library 2001 Version 1, Altera Corporation. San José, CA.