Implementación en un FPGA de Sincronización de Reloj Utilizando Interpolación B-Spline

Cyntia E. Enríquez Ortiz 1, Raúl Fernández Zavala 2

Unidad Profesional Interdisciplinaria de
Ingenierías y Tecnologías Avanzadas (UPIITA) 1,2
Instituto Politécnico Nacional

Resumen

Para proporcionar seguridad a los sistemas embebidos se requieren algoritmos de cifrado que cumplan ciertas restricciones tales como bajo consumo de energía y uso mínimo de recursos. Para poder satisfacer estos requerimientos se han desarrollado varios algoritmos de cifrado ligeros (*Lightweight Cryptography*). El algoritmo de cifrado simétrico TEA (*Tiny Encryption Algorithm*) es uno de ellos y fue diseñado para minimizar el uso de recursos a la vez que incrementa la velocidad de cifrado. En este trabajo se describe la arquitectura en un FPGA para la implementación del cifrado TEA.

Palabras Clave: TEA, Cifrado, FPGA.

I. Introducción

Los protocolos y algoritmos de cifrado constituyen el principal elemento para proteger la información en las redes de computadoras y los sistemas de almacenamiento de datos. Hasta hace algunos años estos sistemas consistían de grandes equipos; sin embargo, recientemente se ha incrementado la necesidad de transmitir información entre pequeños dispositivos móviles tales como asistentes personales, redes de sensores, dispositivos RFID, etc. Los algoritmos de seguridad en estos sistemas deben de cumplir restricciones tales como bajo consumo de energía y uso mínimo de recursos. Para lograr satisfacer estos requerimientos se han desarrollado distintos algoritmos de cifrado ligeros (*Lightweight Cryptography*) [1]; entre estos algoritmos se encuentra el cifrado TEA (*Tiny Encryption Algorithm*) [2] introducido por David Wheeler y Roger Needham y presentado por vez primera en 1994 en el *Fast Software Encryption Workshop*. Este algoritmo se ha considerado como un mecanismo de seguridad en sistemas RFID en algunos trabajos [3], [4]. Otros algoritmos de cifrado ligero son SEA [5] y el DESL [6].

II. Algoritmo TEA

El cifrado TEA presenta uno de los algoritmos más rápidos, compactos y eficientes para cifrado simétrico. El tamaño del bloque de datos es de 64 bits, mientras que la longitud de la llave es de 128 bits. El algoritmo es iterativo y se repite 32 veces. El algoritmo proporciona las propiedades de difusión y confusión sin la necesidad de utilizar cajas de sustitución o de permutación y puede ser fácilmente codificado en ensamblador. Una de las debilidades del TEA es la existencia de llaves equivalentes; para cada llave TEA se pueden generar tres llaves diferentes que producen el mismo texto cifrado para el mismo texto de entrada.

En la figura 1 se muestra el diagrama a bloques de un ciclo de cifrado TEA y como se puede observar presenta una estructura Feistel. Un ciclo del algoritmo TEA está formado por dos rondas de una red de Feistel y la operación XOR es sustituida por sumas en el algoritmo de cifrado. Mediante sumas y desplazamientos se mezclan los datos con la llave. La estructura en hardware es muy simple y solo requiere de las operaciones de suma y OR-exclusiva para proporcionar no linealidad.

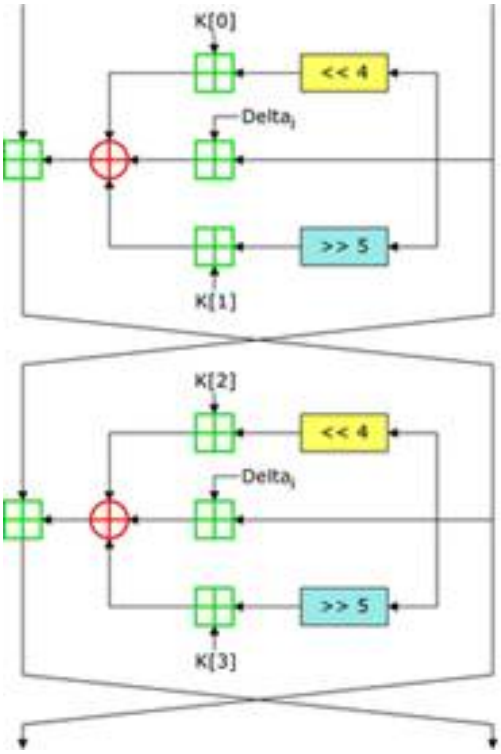

Figura 1: Diagrama a bloques de TEA.

III. Implementación de la Arquitectura en Hardware

Aunque TEA fue propuesto originalmente para ser implementado en software, su diseño simple lo hace adecuado para una implementación en hardware. En la figura 2 se muestra el bloque procesador diseñado para implementar el cifrado TEA. Este módulo contiene la lógica necesaria para realizar un ciclo de cifrado y al ser utilizada en forma iterativa permite implementar en 32 ciclos de reloj el algoritmo TEA.

Figura 2: Procesador para cifrado TEA.

En la figura 3 se muestra el diagrama esquemático del procesador TEA. Para implementar en hardware el algoritmo se definió en verilog HDL la estructura de la figura 1 y mediante una máquina de estados se controlan dos multiplexores para canalizar los datos de entrada o realimentar los datos de salidas de un ciclo básico almacenados en dos registros. Esta operación se repite 32 veces para realizar las 64 rondas de la estructura Feistel correspondiente.

Figura 3: Diagrama esquemático del procesador TEA.

IV. Validación y Resultados

La implementación del sistema se realizó utilizando un sistema de desarrollo DE2 basado en un FPGA *Cyclone II EP2C35672C6* de Altera. Como herramienta de síntesis se utilizó el ambiente de desarrollo *Quartus II Web Edition 8.1*. La descripción en hardware del procesador TEA se realizó en verilog HDL. En la figura 4 se muestra el resultado obtenido de la simulación del procesador para cifrado TEA.

En la figura 5 se muestra el resumen del reporte de compilación generado por el entorno de desarrollo *Quartus II*. Como se puede observar, el número de elementos lógicos utilizados representa aproxima-

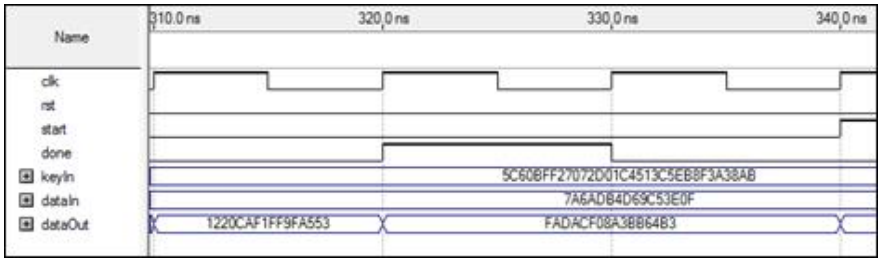

Figura 4: Resultados de la simulación del cifrado TEA.

damente el 1 % (457) del dispositivo utilizado, en tanto que los registros utilizados son menos del 1 % (103). La frecuencia de operación máxima a la que puede trabajar el procesador TEA es de 64.74 MHz con la cual se obtiene una velocidad de cifrado de 129.48 Mbps.

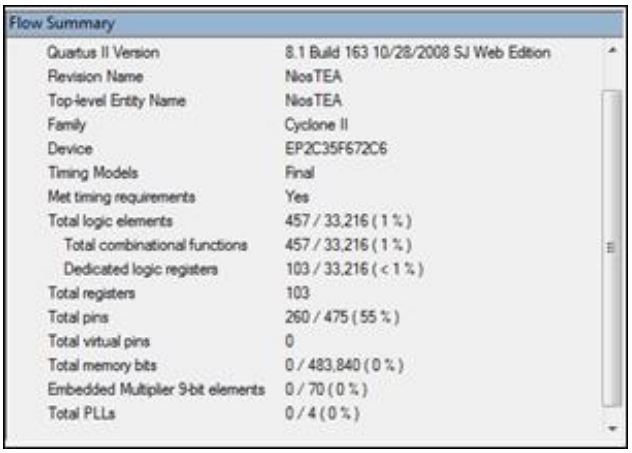

Figura 5: Resumen del reporte de compilación.

V. Conclusiones

En este trabajo se llevó a cabo el diseño en hardware para implementar el algoritmo de cifrado TEA. La descripción correspondiente a una ronda básica de este algoritmo se desarrolló en verilog HDL. El procesador diseñado puede ser utilizado como coprocesador en un sistema basado en microprocesador o como unidad funcional complementaria a la unidad aritmético-lógica para extender el conjunto básico de instrucciones de un procesador definido en software (*softprocesor*). La arquitectura diseñada se puede optimizar en varias formas; por ejemplo, la frecuencia máxima de operación de 64.74 MHz puede ser incrementada segmentando el procesador; de igual manera, el número de elementos lógicos utilizados (457) se puede disminuir si solo se implementa una ronda de la estructura Feistel.

Referencias

[1] David J. Wheeler and Roger M. Needham, “TEA, a tiny encryption algorithm”, *Proc. Fast Software Encryption: Second International Workshop*, Lecture Notes in Computer Science, vol. 1008, December 1994.

[2] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, L. Uhsadel, “A Survey of Lightweight-Cryptography Implementations”, *IEEE Design & Test of Computers*, Vol. 24, No. 6, November 2007.

[3] Y. Yu, Y. Yang, N. Yan, H. Min, “A novel design of secure RFID tag baseband”, *EU RFID Forum 2007*, March 2007.

[4] Israsena, P, “Securing ubiquitous and low-cost RFID using tiny encryption algorithm”, *International Symposium on Wireless Pervasive Computing*, January 2006.

[5] F. Standaert, G. Piret, N. Gershenfeld, J. Quisquater, “SEA a Scalable Encryption Algorithm for Small Embedded Applications”, *Workshop on Lightweight Crypto*, July 2005.

[6] A. Poschmann, G. Leander, K. Schramm, C. Paar, “New Light-Weight Crypto Algorithms for RFID”, *IEEE International Symposium on Circuits and Systems 2007*, May 2007.