IMPLEMENTACIÓN EN HARDWARE DEL ESTÁNDAR DE ENCRIPTACIÓN AVANZADO (AES), EN UNA PLATAFORMA FPGA, EMPLEANDO EL MICROCONTROLADOR PICOBLAZE™

HARDWARE IMPLEMENTATION OF THE ADVANCED ENCRYPTION STANDARD (AES), IN A FPGA PLATFORM, USING THE PICOBLAZE™ MICROCONTROLLER

J. Fernando Piñal M.[1], Ricardo Álvarez G. [1], Alba M. Sánchez G. [2]
ferpimo@hotmail.com / algor@ece.buap.mx / agalvez@solarium.cs.buap.mx

Recibido: marzo 19, 2009 / Aceptado: octubre 28, 2009 / Publicado: noviembre 23, 2009

RESUMEN. En este trabajo analizamos las características del estándar de encriptación avanzado AES y su implementación en una tarjeta de desarrollo FPGA Spartan-3E , utilizando una de las herramientas de procesamiento embebido del fabricante *Xilinx*®, el microcontrolador *PicoBlaze*™. Además se diseñó un bloque en VHDL, el cual es el encargado de realizar la interfaz entre el microcontrolador y los periféricos de entrada- salida de la tarjeta**.** El ingreso de los datos a cifrar puede realizarse de dos maneras: mediante un teclado conectado al puerto PS/2 de la tarjeta o transmitiéndolos por el puerto serie de una computadora personal; para esto se diseñó una interfaz programada en Matlab™. Los datos cifrados pueden observarse en el exhibidor LCD de la tarjeta de desarrollo, o bien se pueden transmitir en modo serial hacia una computadora personal. Estas opciones de funcionamiento del sistema se seleccionan mediante los interruptores deslizables de la tarjeta de desarrollo. La verificación del funcionamiento del sistema se realiza haciendo uso del documento oficial que describe a AES: FIPS-PUB 197. Aun cuando se implementó el algoritmo en un sistema basado en un procesador, se obtuvo un buen rendimiento. Se incluye la comparación del desempeño de nuestro diseño con otras arquitecturas que implementan también el mismo algoritmo.

PALABRAS CLAVE: AES, Criptografía, FPGA, Microcontrolador PicoBlaze™, Matlab™.

ABSTRACT. In this work we analyze the characteristics of the Advanced Encryption Standard AES and its implementation in a FPGA Spartan-3E starter board, using one of the embedded processing tools from the Xilinx® manufacturer, the *PicoBlaze*™ microcontroller. In addition a block in VHDL was designed for the interface between the microcontroller and the input-output elements of the board. The data to be encrypted can be supply from two ways: with a keyboard connected to the PS/2 port of the board, or transmitting it by the serial port of a personal computer. For this, a Matlab™ interface was designed. The encrypted data can be seen in the board LCD display or can be transmitted by serial port to a personal computer; this system options are chosen by the sliding switches of the starter board. The function system validation was made using the official paper that describes the AES: FIPS-PUB 197. Even the algorithm was developed in a processor based system, a good performance was obtained. We compare this performance with other architectures with the same algorithm.

KEYWORDS: AES, cryptography, FPGA, PicoBlaze™ Microcontroller, Matlab™.

Introducción

La criptografía, aunque establecida como una ciencia formal relativamente reciente, es tan antigua como la historia de la humanidad, ya que siempre ha existido la necesidad de poder enviar y recibir información sin que sea interceptada y/o alterada en el trayecto a su destinatario. La criptografía toma al mensaje a proteger

[1] Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla (FCE-BUAP), 18 Sur y Avenida San Claudio, Ciudad Universitaria, C.P 72590, Puebla, Pue. México. - www.ece.buap.mx

[2] Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla (FCC-BUAP), Av. 14 Sur Edificio 104, Ciudad Universitaria, C.P. 72570, Puebla, Pue. México. - www.cs.buap.mx

y lo modifica siguiendo una secuencia ordenada y establecida de pasos utilizando una llave secreta en el proceso, de manera que el mensaje obtenido pueda ser regresado a su forma original únicamente por el emisor y por el destinatario.

El objetivo de este trabajo es describir la implementación un sistema criptográfico en una plataforma electrónica. Para ello se eligió al estándar criptográfico *AES* como modelo a ser sintetizado. El sistema electrónico que lleva a cabo esta tarea es un FPGA elegido debido al desempeño y flexibilidad con la que un algoritmo de esta naturaleza puede ser implementado.

Una de las principales motivaciones para este trabajo es investigar la manera en que se comporta el algoritmo *AES* corriendo bajo el microcontrolador de propósito general *Picoblaze*™, empotrado en una plataforma de aplicación reconfigurable como es un FPGA. El FPGA utilizado en este trabajo es un SPARTAN-3E™ *XC3S500-FG320-4* del fabricante *Xilinx*®, montado en una tarjeta de desarrollo *Digilent*®. Dicha tarjeta cuenta además con: memoria RAM, puertos RS232 y Ethernet, conectores PS/2 y VGA, LCD entre otros periféricos [1].

Esta arquitectura podría resultar especialmente útil en dispositivos portátiles de memoria, Smart cards o en sistemas electrónicos donde el espacio en el dispositivo sea muy limitado, ya sea por falta o limitación de recursos, pero se requiera añadir seguridad al flujo de datos.

Desarrollo

El algoritmo AES se originó en el algoritmo de cifrado Rijndael, en el cual la longitud de los bloques y la longitud de la llave se pueden especificar independientemente a 128, 192 o 256 bits. La especificación de AES usa las mismas opciones de longitud de llaves, pero limita la del bloque a 128 bits y se conforma de cuatro etapas que se repiten durante nueve rondas sobre una matriz de 4x4 llamada State, más una etapa adicional de expansión de la llave. Dichas etapas son: KeyExpansion: expansión de la llave secreta; SubBytes: usa una caja-S para hacer una substitución byte por byte del bloque; ShifRows: una simple permutación; AddRoundKey una simple operación XOR a nivel de bits del bloque actual con una porción de la llave expandida; y MixColumns: una substitución que hace uso de aritmética sobre GF(2^8) [3].
La etapa de cifrado MixColumns es la que más gasto computacional requiere al emplear más del 40% del tiempo necesario para cifrar 128-bits.

Diseño e Implementación del Sistema

El sistema, cuyo diseño se observa en la figura 1, se encarga de ejecutar cuatro tareas principales:

- Gestionar el menú de elección del usuario para el ingreso de datos.
- Ingresar los datos al sistema para su encriptación.
- Cifrar los datos utilizando el algoritmo *AES*-128.
- Regresar los datos encriptados vía serial y desplegarlos en la LCD.

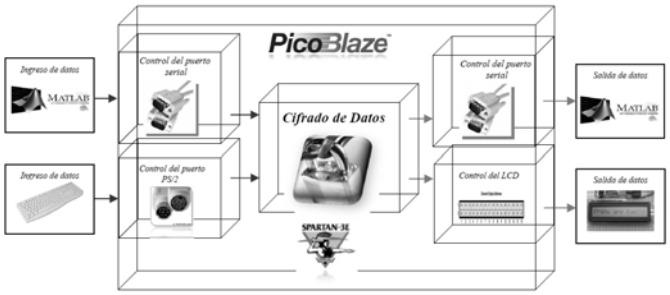

Figura 1. Diagrama conceptual del sistema.

El sistema es capaz de ingresar los datos de dos maneras distintas: por medio del puerto PS/2 conectando un teclado a la tarjeta, y mediante la PC usando una interfaz diseñada en Matlab™ y el puerto RS-232.

Recepción y Transmisión serial. Con el fin de ahorrar hardware, y como el sistema lo permite, se prescindió del uso de un *UART* para la comunicación serial, por medio del microcontrolador *PicoBlaze*™. El protocolo de comunicación se estableció a 115,200 baudios por segundo, utilizando 1-bit de inicio, 8-bits de datos y 1-bit de parada. Para la recepción se utilizó la técnica *bit-banging*, que consiste en detectar el bit de inicio de la comunicación para después muestrear el pin de recepción en un intervalo de tiempo determinado, y finalizar detectando el bit de parada de la recepción. Esta técnica resulta muy útil para transmisiones en dispositivos de procesamiento embebido. Al utilizar una velocidad de transmisión de 115,200 bps el intervalo de muestreo se estableció a 8.68µs, para satisfacer el *Tiempo de Símbolo* de la comunicación. Para la transmisión a la PC se envía primero un bit de inicio que satisfaga el *Tiempo de Símbolo*, después se toma el byte a enviar, bit por bit cada 8.68µs empezando por el *LSB*, para después poner la línea *Tx* en alto para indicar un bit de paro. De nueva cuenta esta tarea se realiza dentro del microcontrolador *PicoBlaze*™.

Comunicación mediante un teclado PS/2. Mediante esta opción se pueden ingresar al sistema los datos a encriptar mediante un teclado para computadora. La gestión de los datos se hace utilizando el protocolo de comunicaciones PS/2. El sistema, de la misma manera que lo hace con la comunicación serial, gestiona la comunicación con el teclado por medio del microcontrolador *PicoBlaze*™. Con el fin de ahorrar recursos en el diseño, lo que se hace es esperar un flanco de bajada en la línea de reloj, comprobar un bit de inicio con un ‘0’ lógico en la línea de datos y comenzar a recibir los bits de los caracteres. Cada carácter recibido ingresa al sistema en un formato llamado *Scan-code*, diferente al código *ASCII*, por lo que se requirió la implementación de un decodificador por hardware para que el sistema pueda encriptar los datos.

Como se observa en la figura 2, después de haber desplegado los mensajes de bienvenida, el sistema entra en espera para que el usuario indique la manera en que desea ingresar los datos a encriptar; esto se hace por medio de los interruptores de la tarjeta de desarrollo.

Figura 2. Secuencia del menú de usuario para el sistema criptográfico

El sistema se encarga de gestionar el ingreso de los datos y la llave a encriptar los almacena linealmente en el *SPM* (*scratch pad memory*) del microcontrolador *PicoBlaze*™.

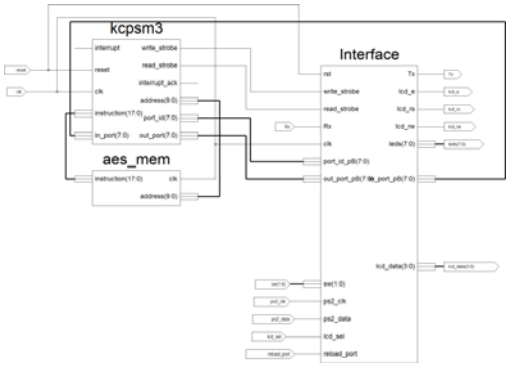

Figura 3. Diagrama esquemático de la arquitectura del sistema

Como se observa en la figura 3, a consecuencia del procesamiento embebido en el microcontrolador *PicoBlaze*™, la arquitectura del sistema ha resultado muy simple, conformada por tres bloques:

- *Microcontrolador PicoBlaze*™: nombrado en el diagrama esquemático como ***kcpsm3***, es el bloque principal del sistema ya que en él se desarrolla todo el procesamiento de datos, además de la gestión de periféricos externos.

- *Memoria de programa del microcontrolador*: designado en el programa como ***aes_mem***, es la memoria donde reside el programa ensamblado a ejecutar por el microcontrolador. Este programa fue compilado y ensamblado con la herramienta *pBlazeIDE*™ del desarrollador *Mediatronix*®.

- *Interface***:** es la entidad encargada de realizar la interfaz entre el microcontrolador y los periféricos *I/O* externos a él: *LCD, push-buttons, switches, RS-232* y *PS/2.* Además se encarga de decodificar los *Scan-codes* que envía el teclado en código *ASCII*, así como albergar la *S-Box* necesaria para la transformación *SubBytes* de *AES*.

Análisis de Resultados

Para verificar que el sistema codifica correctamente según AES-128, se utilizó el documento FIPS-PUB 197, el cual es el documento oficial que describe a AES, y se compararon los ejemplos que contiene dicho documento contra los datos que el sistema procesó, siendo éstos idénticos [2].

Tabla 1. Reporte del mapeo de recursos del diseño

Recurso	*Valor utilizado*	*Porcentaje XC3S500E*
IOBs Externos	25 de 232	10%
BUFGMUXs	2 de 24	8%
RAMB16s	1 de 20	5%
Slices	276 de 4656	5%
SLICEMs	78 de 2328	3%

La tabla 1 muestra el número de recursos utilizados en el diseño. Como se observa es mínimo, esto se debe a que el procesamiento y la gestión de datos se hace en el microcontrolador *PicoBlaze*™, dejando muchos recursos del FPGA disponibles para otras necesidades. Con esto se cumple una de las metas principales de este diseño, y aunque se haya ocupado más del 95% del espacio disponible en la memoria de programa del microcontrolador (974 / 1024 localidades), se ha obtenido una arquitectura compacta en función de los recursos del FPGA.

Rendimiento (Throughput). El número de ciclos de reloj necesarios para encriptar 128-bits a 50 MHz, según los datos obtenidos con la herramienta pBlazeIDE™, es de 5,247 ciclos. Para un algoritmo criptográfico, el Throughput (cantidad de datos procesados por unidad de tiempo) es:

$$Throughput = \frac{Frecuencia\ del\ reloj \times Número\ de\ bits}{Ciclos\ de\ procesamiento}$$

$$Throughput = \frac{50MHz \times 128\ bits}{5247\ ciclos} = 1.21Mbps$$

Comparación con otras Arquitecturas. Con el propósito de conocer el comportamiento del sistema en el mundo real, se comparó el desempeño del sistema con algunas otras implementaciones encontradas en literatura abierta. Estas implementaciones se dividen en tres categorías: Arquitecturas basadas en FPGA, Microprocesadores y Microcontroladores. En la tabla 2 se muestran dichos resultados para AES-128:

Tabla 2. Comparación Desempeño/Recursos entre diferentes dispositivos y arquitecturas

Autor	*Dispositivo*	*Slices*	*BRAM*	*Throughput*
Weaver[4]	FPGA/XVE600-8	460	10	690*Mbps*
Labbé[5]	FPGA/XCV1000-4	2151	4	390*Mbps*
Saggese[6]	FPGA/XCVE2000-8	446	10	1Gbps
Chodwiec[7]	FPGA/XC2530-5	222	3	139*Mbps*
Chodwiec[7]	FPGA/XC2530-6	222	3	166*Mbps*
Standaert[8]	FPGA/XC2300E	542	10	1.45*Gbps*
Gaj[9]	FPGA/XCV1000	2902	—	331.5*Mbps*
Saqib[10]	FPGA/XCV812E	2744	—	258.5*Mbps*
Amphion[11]	FPGA/XVE8	421	4	290*Mbps*
Amphion[11]	FPGA/XVE8	573	10	1.06*Gbps*
Segredo[12]	FPGA/XCV100-4	496	10	417*Mbps*
Segredo[12]	FPGA/XCV600E8	496	10	743*Mbps*
Calderón[13]	FPGA/*Altera*® PF10K	1584	—	637.24*Mbps*
Bernstein[14]	Microprocesador/*Intel*® *Pentium*™ (611 ciclos @ 66MHz)			13.8*Mbps* (*Teóricos*)
Bernstein[14]	Microprocesador/*AMD*® *Athlon*™*X2* 4600+ (213 ciclos @ 2.4GHz)			1.44*Tbps* (*Teóricos*)
Bernstein[14]	Microprocesador/*Intel*® *Core*™*2 Quad* Q6600 (201 ciclos @ 2.4GHz)			1.52*Tbps* (*Teóricos*)
Chung-Huang[15]	Microcontrolador/*Motorola*® 6805 (9000 ciclos @ 2.1MHz)			30*Kbps*
Permadi[16]	Microcontrolador/*Microchip*® PIC16F84 (12225 ciclos @ 4MHz)			41.8*Kbps*
Permadi[16]	Microcontrolador/*Microchip*® PIC16F877 (4559 ciclos @ 4MHz)			112.3*Kbps*
Chung-Huang[15]	Microcontrolador/*Motorola*® 68HC908 (7258 ciclos @ 8MHz)			141*Kbps*
Chung-Huang[15]	Microcontrolador/*Hitachi*® H8/300 (4180 ciclos @ 5MHz)			153.1*Kbps*
SIC-IAIK[17]	Microcontrolador/*Intel*® 8051 (3905 ciclos @ 8MHz)			262.2*Kbps*
Flowers-Schlunder[18]	Microcontrolador-DSP/*Microchip*® *PIC24/dsPIC*® (2808 ciclos @ 16MIPS)			729*Kbps*
Trabajo de Investigación	***PicoBlaze* embebido Spartan3E**	**276**	**1**	**1.21*Mbps***

Conclusiones y perspectivas

Se comprendió el funcionamiento del algoritmo AES para realizar una síntesis correcta y una implementación exitosa de dicho algoritmo en la tarjeta de desarrollo de FPGA.

Aunque la implementación obtenida del diseño del sistema demostró ser la menos competente contra arquitecturas dedicadas en FPGA, también demostró ser la que menos recursos necesita. Aunado a esto no solamente encripta, además se encarga del control de periféricos de entrada/salida.

Comparada contra soluciones en microcontrolador, esta arquitectura es la más rápida, inclusive posee un rendimiento casi por el doble de la solución cercana más rápida, el microcontrolador *PIC24/dsPIC*® del fabricante *Microchip*® optimizado para *DSP* y con instrucciones de 16-*bits*, frente a el microcontrolador *PicoBlaze*™ que ejecuta instrucciones de solamente 8-*bits*.

Siguiendo la misma línea de procesamiento embebido, sería muy interesante considerar el desempeño que podría obtenerse al utilizar algún microprocesador empotrable como *MicroBlaze*™ ó *PowerPC*™, ya que con ellos se podría simplificar el algoritmo de encriptación al emplear instrucciones de 32-*bits* en lugar de las de 8, por lo que fortuitamente se prescindiría de iteraciones, alcanzando así un mucho mayor *Throughput* que el alcanzado para este diseño.

También sería muy útil considerar la herramienta de Codiseño Hw/Sw desarrollada conjuntamente entre los fabricantes *Xilinx*® y *Mathworks*®: *SystemGenerator*™, la cual puede implementar este tipo de algoritmos de manera optimizada.

Referencias

1. Xilinx®. "Spartan-3E Starter Kit Board User Guide (UG230 v1.0)" [en línea]. USA 2006. <http://www.xilinx.com>
2. Federal Information Processing Standards Publication 197. "*Announcing the Advanced Encryption Standard (AES)*" [en línea]. USA Noviembre 26 2001. Disponible en Internet: http://www.csrc.nist.gov
3. William Stallings. USA, 2003. "*Cryptography and Network Security. Principles and Practices*". 3º Edición. Pearson/Prentice Hall.
4. Techinical report, U.C. Berkeley BRASS group. "*High performance, compact AES implementations in Xilinx® FPGAs"* USA 2002. <http://www.cs.berkey.edu/~nnweaver/sfra/rijndael.pdf>
5. Labbé A. Pérez A. "*AES implementations on FPGA: Time flexibility tradeoff*".
6. "*An FPGA-based performance analysis of the unrolling, tiling and pipelining of the AES algorithm in field programmable logic and applications*". Saggese G.P., Mazzeo A., Mazzocca N. and Strollo A.G.M. Ches 2003 Páginas: 292-302, Köln, 2003
7. "*Very compact FPGA implementation of the AES algorithm in cryptographic hardware and embedded systems*". Chodowiec, Pawel and Gaj Kris., Ches 2003, Páginas: 319-333, Köln, 2003
8. "*Efficient implementation of Rijndael encryption in reconfigurable hardware: Improvments and Design tradeoffs*". Standaert François Xavier, Rouvroy Gael, Quisquart Jean Jacques and Legat Jean Didier. Ches 2003, Páginas: 334-350, Köln, 2003
9. "*Comparison of the hardware performance of the AES candidates using reconfigurable hardware*".Gaj Kris and Chodowiec Pawel., In the third AES candidate conference.,New York, 2000
10. "*4.2 Gbit/s single chip FPGA implementation of AES algorithm*".Rodriguez Henriquez F., Saqib N.A. and Diaz Perez A. Electronics letters volumen 39. Páginas: 1115-1116, USA, 2003
11. Amphion. "*CS5210-10: High performance AES encryption cores*". *2003*
12. "*Diseño de un procesador criptográfico Rijndael en FPGA*". Segredoas Alejandro, Zabala Enrique y Bello Gustavo. In X Workshop IBERCHIP, Página: 64, Cartagena, 2004
13. "*Implementación en Hardware del algoritmo Rijndael*". Jácome Calderón Germán, Velazco Medina Jaime, López Hernández Julio. In X Workshop IBERCHIP,Página: 113, Cartagena, 2004
14. Daniel Julius Bernstein. "*AES Speed*". USA Marzo 2008. : http://cr.yp.to/aes-speed.html
15. Chung-Huang Yang. "*Performance of AES on microcontrollers*". USA Marzo 2008. http://www.crypto.idv.tw/AES/index.htm
16. Edi Permadi. "*Implementing AES using PIC16F84*", USA Marzo 2008. http://edipermadi.wordpress.com/2008/01/21/implementing-aes-using-pic16f84
17. Stiftung Secure Information and Communication Technologies. "*Performance-optimized AES implementation for 8051-based microcontrollers*". Austria Marzo 2008. <http://jce.iaik.tugraz.at
18. David Flowers and Howard Henry Schlunder; *Microchip® Technology Inc.* "*Data Encryption Routines for PIC24 and dsPIC® devices*". <http://ww1.microchip.com/downloads/en/AppNotes/0144a.pdf>