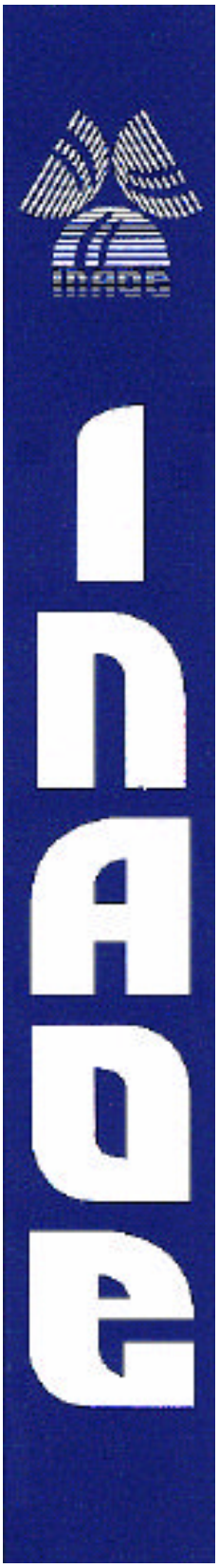

ANÁLISIS E IMPLEMENTACIÓN DE ALGORITMOS CIFRADORES PARA UNA UNIDAD RECONFIGURABLE DE PROCESAMIENTO CRIPTOGRÁFICO EN FPGA

Ignacio Algredo Badillo, René A. Cumplido Parra

REPORTE TÉCNICO NO. CCC-04-006
28 DE JUNIO DEL 2004

Luis Enrique Erro 1
Sta. Ma, Tonantzintla
72840, Puebla, México

ANÁLISIS E IMPLEMENTACIÓN DE ALGORITMOS CIFRADORES PARA UNA UNIDAD RECONFIGURABLE DE PROCESAMIENTO CRIPTOGRÁFICO EN FPGA

Ignacio Algredo Badillo[1], René A. Cumplido Parra[2]

[1]Coordinación de Ciencias Computacionales
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro 1. Sta. Ma. Tonantzintla
78240, Puebla, México
talion00z@ccc.inaoep.mx[1], rcumplido@inaoep.mx[2]

Resumen. En este trabajo se presenta la implementación de una unidad de procesamiento criptográfico, la cual tiene una arquitectura reconfigurable. La característica de reconfiguración fue motivada por seguridad, que da la posibilidad de intercambiar entre varios estándares criptográficos cuando se realiza una comunicación, sin tener varios dispositivos cifradores funcionando al mismo tiempo. Se realizó un análisis para la selección de la arquitectura basándose en las implementaciones de algunos algoritmos criptográficos mediante sus componentes básicos. Mientras que para la selección de algoritmos, se fundamentó en los estándares de IPSec (*Internet Protocol Security*). El desarrollo de este trabajo de investigación originó la implementación inicial de los algoritmos de cifrado DES-ECB, DES-CBC, AES-ECB, SHA-1 monobloque y MD5 monobloque.

Palabras clave. Cómputo reconfigurable, FPGA, DES, AES, SHA-1, MD5.

I. INTRODUCCIÓN

Actualmente, la criptografía tiene un papel importante en muchas aplicaciones de tecnologías de la información [1]. Con un número creciente de personas conectadas a Internet, la transmisión segura de información es un asunto de alta prioridad, donde aplicaciones como correos electrónicos, sucursales virtuales bancarias y empresariales, bases de datos médicas, comercio electrónico, entre otras requieren el intercambio de información privada.

La posibilidad de manejar distintos estándares criptográficos depende de que la mayoría de los protocolos de seguridad modernos y aplicaciones de seguridad sean definidas independientes de los algoritmos, es decir, permiten seleccionar de un conjunto de algoritmos criptográficos para la misma función de seguridad entre las diversas medidas de seguridad.

El desarrollo de este trabajo inició con la selección y el análisis de varios algoritmos criptográficos, así como los parámetros de sus llaves y los requerimientos de hardware. El análisis era necesario para la selección del tipo de arquitectura reconfigurable que iba a soportar los bloques cifradores. Este análisis se llevo a cabo al implementarse los estándares cifradores seleccionados en lenguaje VHDL, mediante componentes electrónicos básicos secuenciales y combinacionales (ORs, ANDs, XORs, registros, memorias, entre otros). Estos elementos indicaron los componentes comunes y por lo tanto, poder seleccionar un esquema de reconfiguración.

Por lo tanto, la arquitectura seleccionada no sólo cumple con los requerimientos establecidos sino también satisface factores económicos como costo de producción y consumo de potencia, teniendo en cuenta que sólo se mantiene un algoritmo criptográfico activo sin utilizar más hardware del necesario.

Los algoritmos seleccionados fueron simulados en Active-HDL e implementados en Xilinx ISE 6 para la medición de parámetros de hardware tales como uso de lógica y frecuencia de operación. Esta información obtenida prueba que la arquitectura reconfigurable propuesta puede tener una eficiencia similar a las soluciones de hardware existentes mientras mantienen requerimientos modestos de hardware.

II. TRABAJO PREVIO

La búsqueda de información relacionada se basó en trabajos que implementarán elementos criptográficos sobre plataformas reconfigurables o programables.
CryptoManiac [d], es un procesador que consiste básicamente de cuatro unidades funcionales operando en una memoria de datos común, con una arquitectura de 4 estados de pipeline. La siguiente tabla muestra el procesamiento de una configuración de CryptoManiac con una frecuencia de reloj de 360Mhz. En la tabla 1 se muestran los resultados de varias implementaciones de algoritmos criptográficos.

Algoritmo	**Total de ciclos**	**Bits/Ciclo**	**Procesamiento (Mbps)**
3DES	336 (7ciclos*48)	0.19	68
3DES corr.	392 (7ciclos*56)	0.16	59
AES-128/128	90 (9 ciclos*10)	1.42	511
AES-128/128 corr.	130 (9 ciclos*13)	0.98	353

Tabla 1. Resultados de procesamiento de CryptoManiac.

En [e] se presenta Cryptonite, un procesador que no ha sido construido en realidad, los resultados de este trabajo de investigación son basados en simulación de software y resultados de síntesis de partes de hardware dedicado. El autor esta consciente de que sus mediciones no han sido realizadas en hardware existente, y en la tabla 2 se tienen los resultados de simulaciones de algoritmos criptográficos.

Algoritmo	**Ciclos**	**Bits/Ciclo**	**Procesamiento (Mbps) a 400Mhz**
AES-128/128	80	1.6	640
AES-128/128	70	1.83	732
DES	35	1.83	732
3DES	105	0.61	244
MD5	504	1.02	406
SHA-1	488	1.05	420

Tabla 2. Resultados de procesamiento de Cryptonite.

III. REQUERIMIENTOS

Los requerimientos del sistema a implementar son:

Reconfiguración. La independencia de algoritmos de algunos protocolos de comunicación (por ejemplo IPSec o SSL) da la posibilidad de poder intercambiar algoritmos criptográficos durante una operación de comunicación. Realizar un intercambio de unidades cifradoras en hardware convencionales es costoso, tanto en recursos económicos como en recursos de hardware (cada elemento cifrador esta presente abarcando espacio de silicio y consumiendo energía), mientras que el uso de hardware reconfigurable es una posible solución.

Actualización. Al intentar aumentar la seguridad aparecen nuevos algoritmos criptográficos, o para los algoritmos existentes se encuentran diseños mejorados, modificados u optimizados. En cualquier caso, los productos hardware de seguridad no consideraron esas modificaciones en el tiempo de diseño y pueden quedar rezagados tecnológicamente. Este requerimiento motiva a ofrecer un producto donde códigos de configuración puedan actualizar al dispositivo criptográfico.

Eficiencia arquitectural. En ciertos casos, una arquitectura hardware puede ser más eficiente si ésta es implementada para un conjunto específico de parámetros (por ejemplo constantes de multiplicación en enteros o en campo de Galois puede ser más eficiente que la multiplicación general). Con dispositivos reconfigurables es posible diseñar y optimizar una arquitectura para un conjunto específico de parámetros [1].

Procesamiento. Se establece como un requerimiento de diseño alcanzar una velocidad de procesamiento ubicada en el estándar Ethernet de 100Mbps.

Cálculo de llaves. Además, en [m] indican que para reducir área, la selección fue generar las llaves al vuelo, o en el tiempo de procesamiento. Este requerimiento permite evitar consumir tiempo en el cálculo y almacenamiento de subllaves, así como su posterior direccionamiento.

IV. ANÁLISIS

Este análisis fue debido a que se necesitaba encontrar un máximo de elementos comunes entre los algoritmos a implementar, y tratar de obtener un mínimo de módulos reconfigurables, pero sin intentar complicar las conexiones entre tales elementos para obtener una distinta funcionalidad.

Las implementaciones se constituyeron de componentes electrónicos básicos tanto combinacionales como secuenciales. El análisis de los algoritmos cifradores se basó en que la mayoría de los cifradores pueden ser especificados como flujos de datos en un grafo constituido de pocos componentes [2]. En general, hay componentes comunes:

- Operaciones de aritmética simple. La mayoría de los algoritmos criptográficos hacen uso de operaciones simples como la suma y la resta. Estas operaciones se implementan fácilmente en hardware, pero debido a su naturaleza básica y simple, éstas no ofrecen ganancia al implementarse en sistemas reconfigurables.

- Multiplicación. Esta operación es una tarea difícil de realizar en hardware, consumen gran cantidad de recursos de hardware y calcula sus resultados lentamente. El algoritmo AES hace uso de estas operaciones. Hay varias formas de mejorarlas, dependiendo de sus operandos y su fin básico:

  * Propósito general.
    En estos multiplicadores los operandos pueden tomar cualquier valor, por lo que son los más costosos de implementar en hardware. Sin embargo, en los algoritmos criptográficos, los multiplicadores n x n, devuelven un resultado de n bits. En dispositivos reconfigurables,

el tamaño y número de los sumadores pueden ser reducidos, eliminando la necesidad de calcular bits que después son ignorados.

* Multiplicación por una constante.
  Son componentes de hardware altamente especializados, por lo que multiplicadores por una constante son hechos considerablemente más pequeños y rápidos que los multiplicadores de propósito general.

* Multiplicación usando un esquema de codificación redundante.
  Estas implementaciones optimizan los recursos de hardware, realizando multiplicaciones constantes a través de un esquema redundante.

- Operaciones Lógicas Paralelas. El hardware permite que varias operaciones sean realizadas paralelamente, la cual es una ventaja fundamental de las implementaciones hardware sobre el software en cómputo. Dispositivos configurables pueden ser programados para realizar ciertas operaciones como operaciones lógicas complejas en paralelo. Además, dado que el número y tipo de unidades funcionales necesarios en algún punto del cómputo es configurado por la aplicación, el paralelismo nunca es restringido por la falta de unidades funcionales, como debe suceder en una arquitectura VLIW.

- Secuencias de operaciones lógicas. La mayoría de las arquitecturas reconfigurables, incluyendo los FPGAs Xilinx estándar comerciales, implementan unidades funcionales usando LUT's (*lookup tables*). Entonces, una secuencia de operadores puede ser combinado hacia un simple operador, usando una LUT apropiadamente.

- *Table LookUp.* La mayoría de los cifradores incluyen un elemento de substitución (cajas de substitución), las cuales no son fáciles de expresar como una transformación lineal y son implementadas como LUT's. Muchos dispositivos reconfigurables pueden implementar LUT's de este tipo, mientras otros deben necesitar memoria externa para almacenar LUT's apropiadamente.

- Rotación y Corrimiento. Por último, dos operaciones comunes en criptografía son las rotaciones y corrimientos a nivel de bits. Éstas son ineficientes al implementar en microprocesadores o software, mientras que en hardware no requieren recursos del todo, sólo simples reordenamientos de líneas.

Se pretende obtener una biblioteca completa de algoritmos criptográficos para diversas funcionalidades de la unidad procesadora criptográfica, pero debido a la gran cantidad de estándares se seleccionó un grupo inicial, basados en los utilizados en IPSec. Por lo que se necesitaban algoritmos con fundamentos distintos y obtener una idea general para la selección de la arquitectura.

Los estándares seleccionados fueron el FIPS-46-3 [3] para el DES, FIPS-197 [4] para el AES, FIPS-180-2[5] sólo para el SHA-1 y el RFC1321 [6] para el MD5.

Cabe señalar, que se hicieron varias implementaciones con diseños modulares para un algoritmo criptográfico, por ejemplo, partiendo de esquemas que son conformados con XOR's a nivel de bits hasta esquemas compuestos con XOR's a nivel de bus de datos. Tales consideraciones se realizaron para realizar un análisis a partir de componentes que conformaban a los diferentes algoritmos y poder seleccionar un esquema de reconfiguración que soportará las diferentes implementaciones criptográficas.

V. IMPLEMENTACIÓN Y RESULTADOS

Las implementaciones de los algoritmos cifradores fueron capturados y simulados usando Active-HDL. Estos diseños fueron sintetizados, mapeados, colocados y ruteados en un FPGA Xilinx XC2V1000-FG456 con la herramienta Xilinx ISE 6, además se creó un modelo de simulación "Post-Place & Route" que validó el funcionamiento de cada diseño mediante Active-HDL 5.1.

Las implementaciones de los algoritmos fueron diseñados optimizando recursos del FPGA (a excepción de la implementación del AES donde se utilizaron 20 bloques de memoria ROM), sin utilizar técnicas de pipeline, lazos desenrollados o la combinación de éstas dos técnicas. Diseños con estos métodos de mejoramiento deben aumentar la capacidad de procesamiento de las implementaciones de los algoritmos criptográficos que aquí se presentan.

1. DES

El esquema general de la implementación del DES-ECB se puede ver en la figura 1.

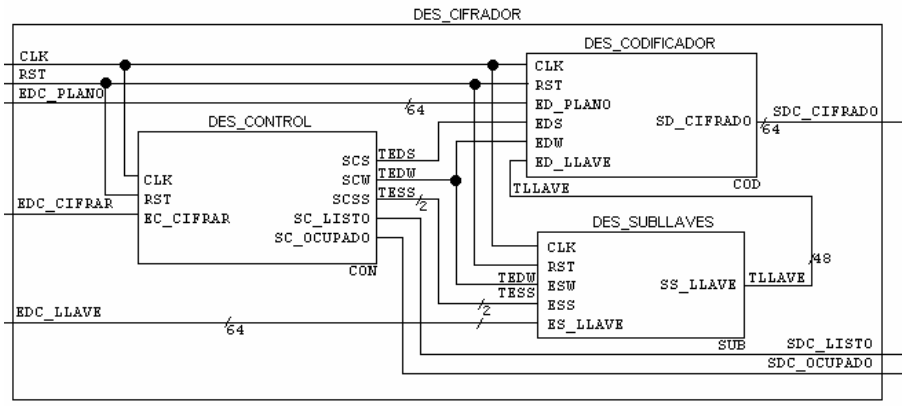

Figura 1. Diagrama a bloques de la implementación del DES-ECB.

Se realizaron varios diseños e implementaciones del algoritmo DES en modo ECB. Los resultados de estas implementaciones se pueden ver en la tabla 3, tomando en cuenta que estos diseños utilizaron los mínimos recursos posibles. El diseño más rápido es el DESv2.1, el cual procesa a 561.72 Mbps. Cabe señalar que DESv2.1, con un proceso que tarda 17 ciclos de reloj, es más rápido que los diseños DESv3 o DESv3., los cuales su proceso tarda 16 ciclos de reloj.

Diseño	Período	Frecuencia	IOBs	Slices	LUTs 4-E	Ciclos	Procesamiento
DESv1	6.532ns	153.1MHz	189/324	217/5120	409/10240	18	544.32 Mbps.
DESv1.1	10.007ns	99.93MHz	189/324	202/5120	370/10240	18	355.30 Mbps.
DESv1.2	10.916ns	91.60MHz	189/324	874/5120	818/10240	18	325.71 Mbps.
DESv1.3	12.706ns	78.70MHz	189/324	875/5120	818/10240	18	279.83 Mbps.
DESv2	6.784ns	147.4MHz	189/324	219/5120	411/10240	17	554.93 Mbps.
DESv2.1	6.702ns	149.2MHz	189/324	223/5120	408/10240	17	561.72 Mbps.
DESv3	10.225ns	97.79MHz	189/324	214/5120	404/10240	16	391.19 Mbps.
DESv3.1	9.780ns	102.2MHz	189/324	215/5120	403/10240	16	409.00 Mbps.

Tabla 3. Resultados de la implementación del DES-ECB.

En la tabla 4 se muestran los resultados de la implementación del algoritmo DES en modo CBC, estándar requerido en el protocolo IPSec, con base al algoritmo DES-ECB (ver figura 1). El mejor resultado de procesamiento se obtuvo con el diseño DESv1.

DESv1	6.900ns	144.9MHz	253/324	250/5120	476/10240	17	545.60 Mbps.
DESv1.1	10.322ns	96.88MHz	253/324	1262/5120	1141/10240	17	364.73 Mbps.
DESv2	8.214ns	121.7MHz	253/324	245/5120	454/10240	16	486.97 Mbps.

Tabla 4. Resultados de la implementación del DES-CBC.

2. AES

El algoritmo AES a implementar procesará bloques de 128 bits con llaves de 128 bits, en modo ECB. El esquema general de la implementación se puede ver en la figura 2.

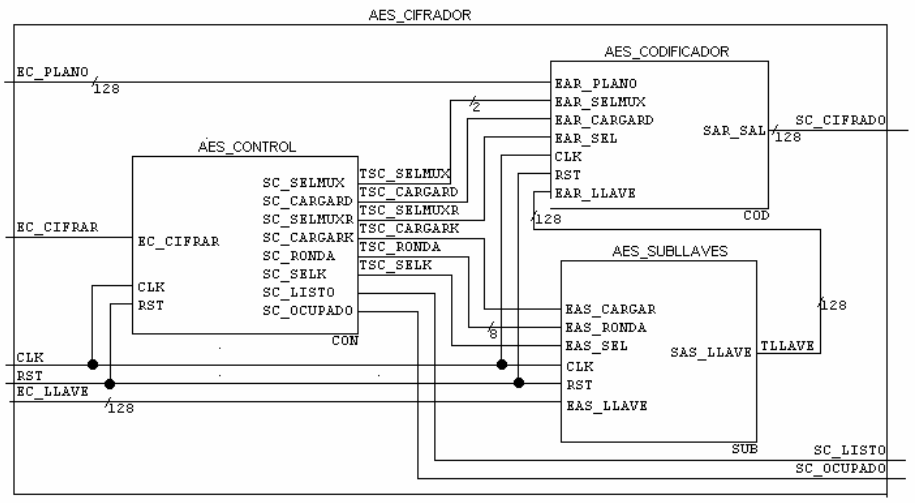

Figura 2. Diagrama a bloques de la implementación del AES-ECB.

En la tabla 5 se pueden ver los resultados de la implementación del algoritmo AES-ECB, donde se ve que se realizaron varios diseños. Una característica de estos diseños es que utilizan 20 bloques de memoria ROM para las Cajas-S (S-Box). El diseño AESv2 tiene el mejor procesamiento con 927.37 Mbps., con un proceso de 12 ciclos de reloj. Este diseño de 12 ciclos de reloj es mejor que los diseños AESv3 y AESv3.1, los cuales procesan en sólo 11 ciclos de reloj.

Diseño	Período	Frecuencia	IOBs	Slices	LUTs 4-E	Ciclos	Procesamiento
AESv1	11.604ns	86.17MHz	262/324	2210/5120	4090/10240	13	848.51 Mbps.
AESv1.1	13.763ns	72.65MHz	262/324	1872/5120	3457/10240	13	715.40 Mbps
AESv1.2	16.169ns	61.84MHz	262/324	4211/5120	5090/10240	13	608.95 Mbps.
AESv1.3	16.673ns	59.97MHz	262/324	4213/5120	5090/10240	13	590.54 Mbps.
AESv2	11.502ns	86.94MHz	263/324	2335/5120	4327/10240	12	927.37 Mbps.
AESv2.1	12.167ns	82.18MHz	263/324	2210/5210	4061/10240	12	876.69 Mbps.
AESv3	15.384ns	65.00MHz	263/324	2737/5120	5165/10240	11	756.39 Mbps.
AESv3.1	14.105ns	70.89MHz	263/324	2778/5120	5108/10240	11	824.98 Mbps.

Tabla 5. Resultados de la implementación del AES-ECB.

3. SHA-1

El algoritmo SHA-1 fue acorde al FIPS-180-2, el diagrama general de esta implementación se puede ver en la figura 3.

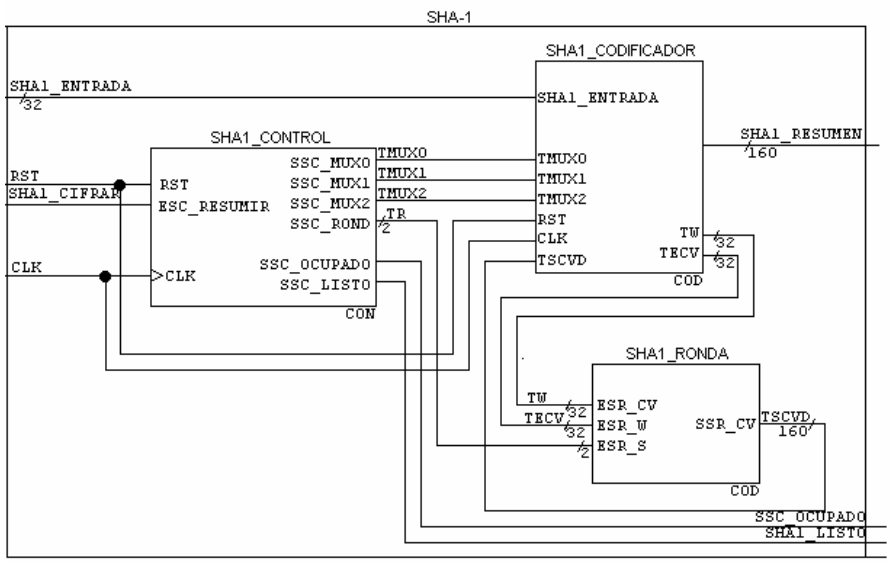

Figura 3. Diagrama a bloques de la implementación del SHA-1.

Los resultados de la implementación del algoritmo SHA-1 se pueden ver en la tabla 6. El diseño SHAv2 es el que mayor procesamiento presenta de 636.43 Mbps con 80 ciclos de reloj.

Diseño	**Período**	**Frecuencia**	**IOBs**	**Slices**	**LUTs 4-E**	**Ciclos**	**Procesamiento**
SHAv1	10.113ns	98.88MHz	197/324	830/5120	666/10240	82	617.41 Mbps
SHAv1.1	28.572ns	34.99MHz	197/324	897/5120	710/10240	82	218.53 Mbps
SHAv1.2	20.310ns	49.23MHz	197/324	3875/5120	3404/10240	82	307.42 Mbps
SHAv1.3	17.178ns	58.21MHz	197/324	3873/5120	3396/10240	82	363.48 Mbps
SHAv2	10.056ns	99.44MHz	197/324	579/5120	647/10240	80	636.43 Mbps.

Tabla 6. Resultados de la implementación del SHA-1.

4. MD5

El diagrama a bloques de la implementación del algoritmo MD5 conforme al RFC-1321que se utilizó en el diseño se puede ver en la figura 4.

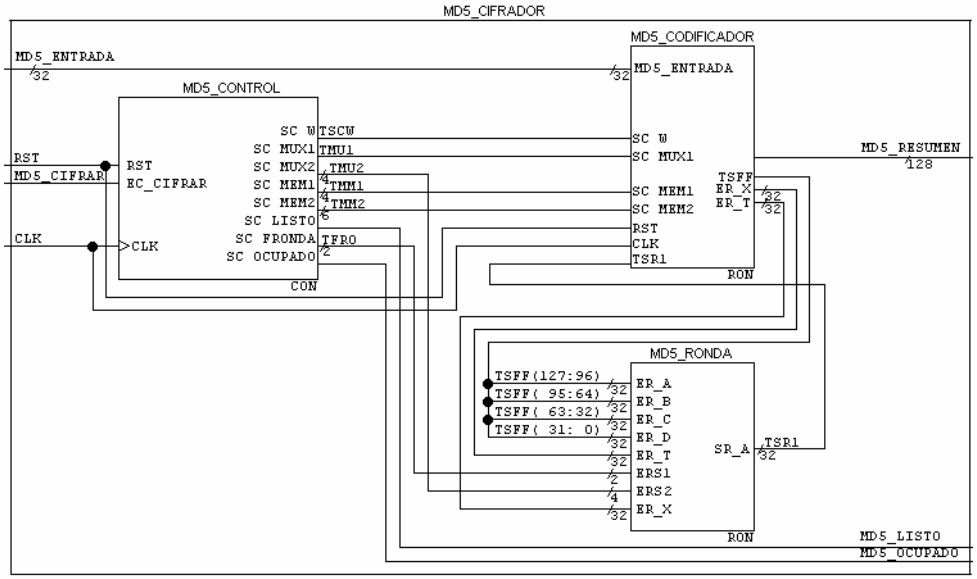

Figura 4. Diagrama a bloques de la implementación del MD5.

Se realizaron varias implementaciones del algoritmo MD5, cuyos resultados de implementación se pueden ver en la tabla 7. El diseño MD5v2 realiza el mejor procesamiento de 406.48 Mbps., en 65 ciclos de reloj,

Diseño	**Período**	**Frecuencia**	**IOBs**	**Slices**	**LUTs 4-E**	**Ciclos**	**Procesamiento**
MD5v1	27.430ns	36.45MHz	165/324	833/5120	1081/10240	66	282.81 Mbps.
MD5v1.1	27.734ns	36.05MHz	165/324	770/5120	926/10240	66	279.71 Mbps.
MD5v1.2	30.756ns	32.51MHz	165/324	1474/5120	1493/10240	66	252.22 Mbps.
MD5v1.3	28.845ns	34.66MHz	165/324	1474/5120	1494/10240	66	268.94 Mbps.
MD5v3	19.378ns	51.60MHz	165/324	899/5120	1229/10240	65	406.48 Mbps.
MD5v3.1	21.166ns	47.24MHz	165/324	895/5120	1211/10240	65	372.15 Mbps.

Tabla 7. Diagrama a bloques de la implementación del MD5.

VI. PROPUESTA DE RECONFIGURACIÓN

Los resultados de las implementaciones de la sección V presentan varias opciones de procesamiento, que pueden ser utilizados en diferentes estándares de velocidad en la transferencia de datos. Pero, la finalidad de realizar las implementaciones funcionales de los algoritmos criptográficos en base a diseños modulares era para seleccionar un esquema de reconfiguración, entre los cuales consideramos los siguientes:

1. Unidades de Procesamiento Reconfigurables y Unidades de Procesamiento Básicas.

   Se considera que se tienen unidades de procesamiento comunes a las implementaciones de los algoritmos criptográficos seleccionados, y se tienen unidades especializadas para realizar el trabajo criptográfico requerido. Estas unidades especiales son los elementos reconfigurables del sistema, ya que son útiles y únicas para una implementación dada, ver figura 5.

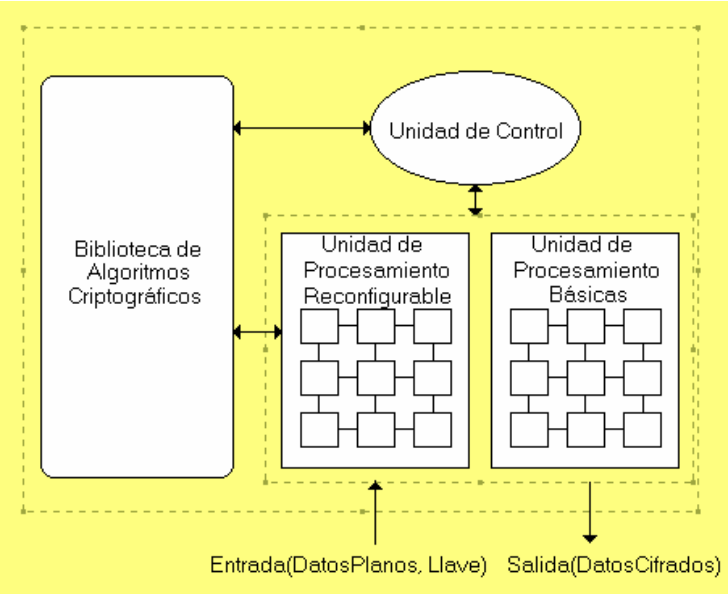

Figura 5. Esquema de unidades de procesamiento reconfigurables y básicas.

2. Esquemas de Bloques Cifradores.

Esta plataforma requiere que para obtener una diversidad en su funcionalidad criptográfica es necesario reemplazar totalmente un módulo por otro módulo con una operación criptográfica distinta o con un diferente diseño. Ver figura 6.

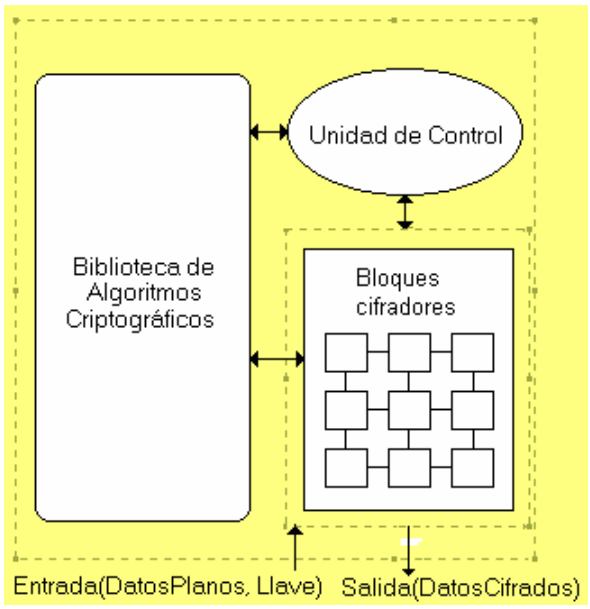

Figura 6. Esquema de bloques cifradores.

3. Sistema con ALU especializada.

Este esquema presenta un modo de procesamiento similar a un microprocesador, pero con una unidad lógica – aritmética (ALU) especial para realizar el procesamiento de distintas implementaciones de algoritmos criptográficos, ver figura 7.

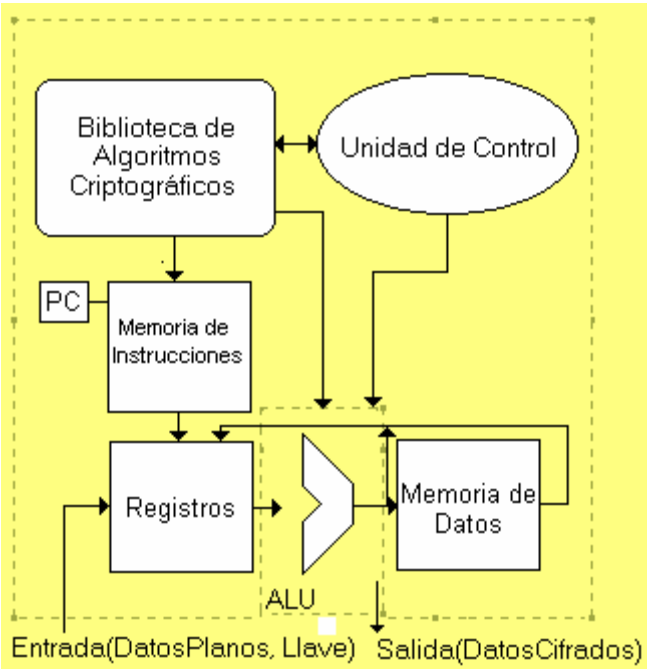

Figura 7. Sistema con ALU especializada.

Por lo que para la selección de alguno de estos esquemas, se realizó un análisis basado en las implementaciones de la sección V, al comparar los elementos modulares que componen cada implementación de los diferentes algoritmos criptográgicos. Los resultados de este análisis se pueden ver en la tabla 8.

Componentes	**DES ECB**	**DES CBC**	**AES ECB**	**SHA-1 1bloque**	**MD5 1bloque**
AND2TO1_32BIT				4	4
NOT_32BIT				1	2
OR2TO1_32BIT					3
XOR2TO1_8BIT			33		
XOR4TO1_8BIT			16		
XOR2TO1_9BIT			16		
XOR2TO1_32BIT	1	1	1	5	3
XOR3TO1_32BIT			1	3	
XOR4TO1_32BIT			1		
XOR5TO1_32BIT			1		
XOR2TO1_48BIT	1	1			
XOR2TO1_64BIT		1			
XOR2TO1_128BIT			2		
REGISTRO_28BIT	2	2			
REGISTRO_32BIT	2	2			16
REGISTRO_128BIT			2		1
REGISTRO_160BIT				1	
REGISTRO_512BIT				1	
SUMADOR2TO1_32BIT				9	8
MUX2TO1_8BIT			32		
MUX3TO1_28BIT	2	2			

Tabla 8. Resultados del análisis para la selección de la plataforma (continúa).

MUX2TO1_32BIT	2	2		1	
MUX3TO1_32BIT				1	
MUX4TO1_32BIT				1	1
MUX2TO1_64BIT		1			
MUX2TO1_128BIT			2		1
MUX3TO1_128BIT			1		
MUX2TO1_160BIT				1	
MUX2TO1_512BIT				1	
BLOQUES RAM 64X32					1
BLOQUES RAM 64X4	8	8			
BLOQUES RAM 256X8			10		
FSM 7 ESTADOS					1
FSM 8 ESTADOS				1	
FSM 11 ESTADOS			11		
FSM 17 ESTADOS	17	17			
CONTADOR_4BCD					1
CONTADOR_64BCD					1
CONTADOR_80BCD				1	

Tabla 8. Resultados del análisis para la selección de la plataforma (continuación).

La información que muestra la tabla 8, indica la base del análisis para la selección de la plataforma que soportará las implementaciones de los algoritmos criptográficos. Estos resultados son los módulos necesarios para la implementación de cada estándar criptográfico seleccionado. La tabulación resalta que los componentes no son comunes entre las diferentes implementaciones criptográficas, a excepción de la XOR de 32 bits de dos entradas – una salida. Además, se mantiene una pequeña similitud entre el uso de componentes iguales y la cantidad usada, por lo que el mantener un mínimo número de componentes reconfigurables y básicos no sería un esquema ideal, aunado a la conexión entre los diferentes módulos que se tengan presentes para una función criptográfica, es decir el esquema de la figura 5 no es apropiado.

En contraste con el esquema de la figura 7, se tendría una ALU especial que cambiaría su funcionalidad conforme sea requerida. Esta opción reconfiguraría únicamente a la ALU para poder realizar las operaciones criptográficas de manera secuencial manejando microcódigos. La mayor desventaja radica en la naturaleza secuencial, porque gran parte de los algoritmos criptográficos pueden tener esquemas paralelos en el procesamiento de la información, además que ciertos modos de funcionamiento de cada estándar cifrador permiten el uso de pipeline.

Finalmente, el esquema restante (ver figura 6) consiste en manejar bloques o módulos completos que contienen la lógica necesaria para realizar el procesamiento criptográfico, ya sea de autenticación o de cifrado.

Es decir, el esquema base de la plataforma criptográfica tendrá un diagrama similar al de la figura 6, donde el módulo principal "bloques cifradores" será el único elemento reconfigurable de la plataforma. Este esquema da las ventajas de poder utilizar el paralelismo o manejar etapas de pipeline a diferentes niveles, además de manejar mínimos recursos de hardware y evitar un complicado esquema de conexión si se utilizarán elementos básicos o comunes, debido a la forma funcional de cada estándar criptográfico. Además, nuevas implementaciones podrán ser agregadas sin gran dificultad por la reconfiguración total del módulo principal.

VII. CONCLUSIONES

En este trabajo se presentaron implementaciones en hardware de algoritmos criptográficos. Se presentan resultados de algoritmos de cifrado como el AES en modo ECB y del DES en modo CBC

y ECB. Mientras tenemos implementaciones de algoritmos de autenticación como el SHA-1 y el MD5, los dos en modo monobloque.

Las mejores velocidades de procesamiento alcanzado por las implementaciones de los algoritmos cifradores son para el DES-ECB de 554.9 Mbps., para el DES-CBC de 545.6 Mbps., para el AES-ECB de 927.3 Mbps., para el SHA-1 de 636.4 Mbps y para el MD5 de 406.48 Mbps. Estas implementaciones fueron realizadas sin técnicas de pipeline, desenrollamiento de lazos o alguna combinación de éstas. Los resultados de procesamiento presentados en las tablas son obtenidos de los archivos generados por el Place & Route del ISE 6.

Además, las implementaciones presentadas fueron base para la realización del análisis y la selección de la plataforma reconfigurable. El análisis generó resultados como un mínimo número de elementos comunes y una idea de recursos necesarios para cada implementación de un estándar criptográfico. Este último dato (ver tablas 3-7) indica que manejar los cuatro algoritmos criptográficos al mismo tiempo en un FPGA, saturan el uso de sus recursos y aumenta el consumo de energía, así como también se dificulta la reconfiguración de las implementaciones, ya sea para agregar, modificar, cambiar o actualizar la funcionalidad del sistema.

Trabajo por concluir es el diseño e implementación de la arquitectura reconfigurable, es decir, de la unidad de procesamiento criptográfico que soportará los diversos estándares de cifrado y de autenticación. También se aumentará la capacidad de procesamiento debido a las velocidades alcanzadas (ver tablas 3-7), ya que el requerimiento inicial era de 100 Mbps y se intentará manejar velocidades de 1Gbps, tomando en cuenta que en IPSec se maneja el DES en modo CBC y los algoritmos SHA-1 y MD5 no pueden manejar etapas de pipeline en un proceso multibloque de información.

Trabajo a futuro implica tener una biblioteca de estándares criptográficos que incluyan algoritmos de llave pública y de llave privada, así como de autenticación y manejar diversos estándares de velocidad de comunicación o tener diferentes implementaciones dependientes del uso de recursos de hardware.

REFERENCIAS

[1] Adam J Elbirt, “Reconfigurable Computing for Symmetric-key Algorithms”. Tesis. 2002.

[2] Alex Panato, Marcelo Barcelos, Ricardo Reis, “A Low Device Occupation IP to Implement Rijndael Algorithm”.

[3] Federal Information Processing Standards (FIPS) Publication 46-3, “Data Encryption Standard (DES)”, US DoC/NIST, Octubre 1999.

[4] Federal Information Processing Standards (FIPS) Publication 197, “Announcing the Advanced Encryption Standard (AES)”, US DoC/NIST, Noviembre 2001.

[5] Federal Information Processing Standards (FIPS) Publication 180-2, “Announcing the Secure Hash Standard”, US DoC/NIST, Agosto 2002.

[6] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, MIT and RSA Data Security, Inc., Abril 1992.