Full-pipelined CTR-AES를 이용한 Giga-bit 보안모듈 설계

T.Q. Vinh* · 박주현* · 김영철* · 김광옥**

A design of Giga-bit security module using Fully pipe-lined CTR-AES

T.Q. Vinh* · Ju Hyun Park* · Young chul Kim* · Kwang Ok Kim**

이 논문은 2007년도 한국전자통신연구원 연구비를 지원받았음

요 약

현재 가정과 소규모 사업장에서 재정적인 변화와 개인 커뮤니케이션 그리고 원격의료에 이르기까지 점점 GPON 사용이 일반화 되어가고 있다. 이러한 PON의 다중사용 때문에 개인정보 보호와 커뮤니케이션 보호를 위한 보안의 필요성이 더더욱 커지고 있다. 이를 위해 이 논문에서는 Virtex4 FPGA를 기반으로 AES의 카운터 모드를 구현하였다. 본 논문에서 구현된 구조는 pipeline 구조 구현을 위하여 크게 세 가지 특징을 가지고 있는데 1) composite filed 연산을 이용한 Subbyte, 2) efficient MixColumn transformation, 그리고 3)on-the-fly key scheduling이다. 구현된 S-box는 면적의 17% 감소와 on-the-fly key 스케줄링 기법으로 pipeline 구조에 특화된 key-expander 기능을 구현하였다.

ABSTRACT

Nowdays, homes and small businesses rely more and more PON(Passive Optical Networks) for financial transactions, private communications and even telemedicine. Thus, encryption for these data transactions is very essential due to the multicast nature of the PON In this paper , we presented our implementation of a counter mode AES based on Virtex4 FPGA. Our design exploits three advanced features; 1) Composite field arithmetic SubByte, 2) efficient MixColumn transformation, 3) and on-the-fly key-scheduling for fully pipelined architecture. By pipeling the composite field implementation of the S-box, the area cost is reduced to average 17 percent. By designing the on-the-fly key-scheduling, we implemented an efficient key-expander module which is specialized for a pipelined architecture.

키워드

GPON, CTR-AES, arithmetic SubByte, on-the-fly key expander

Ⅰ. 서론

현재 가정과 소규모 사업장에서 재정적인 변화와 개인 커뮤니케이션 그리고 원격의료에 이르기 까지 점점 GPON의 사용이 일반화 되어가고 있다. 이러한 GPON의 사용에 힘입어 개인정보 보호와 커뮤니케이션 보호를

* 전남대학교 컴퓨터정보통신학과
** 한국전자통신연구원

접수일자 2008. 05. 30

위한 보안의 필요성이 더더욱 커져가고 있다[1]. GPON의 보안을 위한 알고리즘은 5개의 AES(Advanced Encryption Standard)인 전자코드북(ECB), 암호블럭 chainin-g(CBC), 암호 피드백 (CFB), 출력 피드백(OFB), counter 모드 중 counter mode를 채택하여 사용하고 있다[2]. 이 논문에서 우리는 면적과 성능을 고려한 최적의 full pipelined 구조를 이용하여 CTR-AES를 구현한다.

현재까지 많은 논문에서 AES 알고리즘을 위한 하드웨어 구조를 제안하고 있다. 특히 대부분 high throughput을 위하여 pipeline 구조를 사용하고 있는데 그들의 구조는 outer, inner, 그리고 inner-outer 구조로 다양하게 구현되어 졌다[3][4][5]. 특히 [3],[4],[5]에서는 outer-round pipeline 구조를 사용하고 있는데, 그 구조는 각 round에 pipeline stage를 가지고 있으며 각 round 사이에 레지스터를 삽입하였다.

Outer-round pipelining 기술을 사용하여 설계한 하드웨어 throughput은 128-bit AES 경우 기존에 비해 11배 성능향상을 이루었고 3Gbit/s의 전송 속도를 보이고 있다. 그러나 optical network 경우, 10 Gbits/s 이상의 데이터 전송률을 요구하고 있다. 이 전송률을 위해서 [7],[8],[9]에서는 내부 pipeline 구조를 제안하였는데 내부 pipeline에서는 각 round 마다 sub-pipeline stage 로 나누고 각 sub-pipeline stage 사이에 레지스터를 삽입하게 된다. AES 알고리즘의 기본 구조는 SubByte, ShiftRow, MixColumn 그리고 AddRoundKey로 구성되어 있고 11라운로 동작하며 마지막 라운드에서는 MixColumn은 제외된다. Pipelining cipher round는 최대 동작 주파수를 위하여 각 stage의 delay 균형을 맞춰야 하는 어려움이 있다.

이 논문에서 우리는 면적과 성능 면에서 최적화 되어진 full pipelined 구조를 이용하여 CTR-AES를 구현하였다.

본 논문의 구성은 다음과 같다. 본론에서는 우리가 제안한 CTR-AES 구조 특징을 SubByte, MixColumn, 그리고 key expander 측면에서 설명하고 제안된 구조의 구현 및 다른 구조와 비교 설명할 것이다. 그리고 결론을 맺고자 한다.

Ⅱ. 본론

CTR AES는 Subbytes, ShiftRow, Mixcolumn, Add Roundkey 그리고 Key expander 블록으로 구성되어 있다. AES 총 11 round를 거쳐 ciphertext를 출력한다. 이 논문에서는 속도 향상을 위하여 loop를 사용하지 않고 inner and outer pipeline 구조, Subbyte에서는 composite field operation, 그리고 on-the-fly key expander 구조를 사용하여 속도를 향상시켰다.

1. GPON security 구조

GPON 암호화 모듈은 GPON의 Tx/Rx 링크에서 커뮤니케이션의 안정성을 보장하기 위하여 필요한 기능이다. 이 암호화 모듈을 이용하여 전송 데이터 프레임의 확실성, 무결성을 보장 받게 된다. 그림1은 GPON 암호화 모듈의 구성을 보여주고 있다.

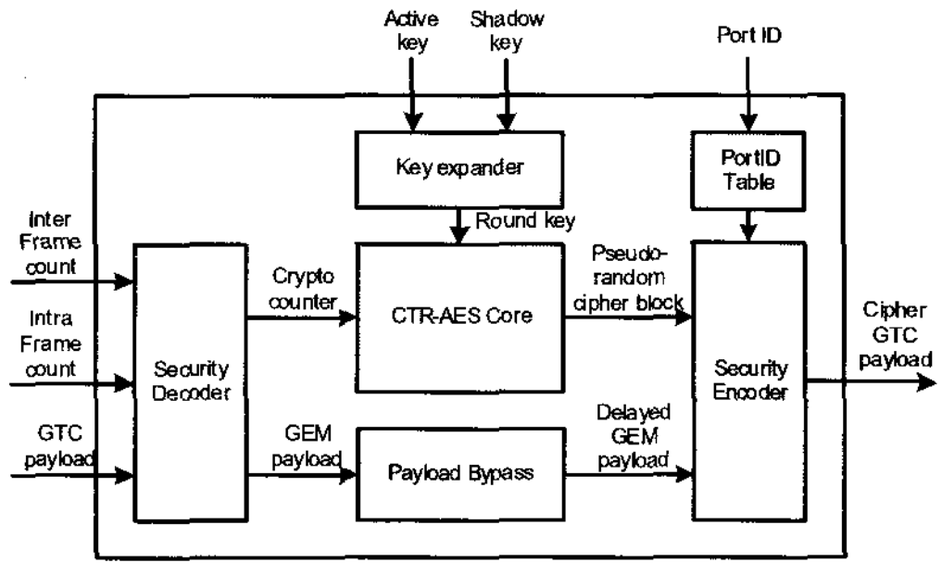

그림 1. GPON 암호화 모듈 구조
Fig. 1 The top structure of the GPON security module

a. Port-ID table : 이 블록은 4k 12bit 레지스터로 구현되어 지는데 기능은 port 식별자를 저장하는 기능이다. 적절한 Port-ID를 가지 프레임만 암호화 된다.
b. Security decoder : 다음과 같은 포맷을 가진 crypto 카운터를 만들어 낸다.
format : (Inter Frame Count[19:0] & Intra Frame Count[15:0]) & (Inter Frame Count[29:0] & Intra Frame Count[15:0]) & (Inter Frame Count[29:0] & Intra Frame Count[15:0]).
또한 payload 바이패스를 위하여 128 bit GTC payload를 저장한다.
c. Key expander : 이 블록은 128bit key 입력으로부터 라운드 킷값을 만들어 낸다. 라운드 킷값의 전체 bit 수는 1408 = 128*(10 + 1)이다.
d. CTR-AES core : 이 블록은 crypto 카운터 입력값 부분을 제외하고는 AES 알고리즘과 동일하다. Crypto

카운터는 매 128bit 데이터 블록마다 증가한다.

e. Payload bypass : 이 블록은 암호화 없는 payload 값을 전달한다. 주요 기능은 암호화 된 GEM payload 출력과 delay 시간을 동기화 하는데 있다.

f. Security encoder : 이 블록은 bypass 된 GEM payload로부터 cipher GEM payload와 암호화된 GEM payload를 다중화 하는 기능을 가지고 있다.

GPON에서의 AES 알고리즘은 데이터를 암호화하기 위하여 카운터 모드를 사용한다. 카운터 모드 암호화에서는 forward cipher 기능이 각 카운터 블록에 포함되어 있다. 그리고 결과는 평문 블록과 XOR 연산을 행하여 암호화 블록을 만들어 낸다. Forward cipher 함수는 CTR decryption 과 encryption 모두에서 사용된다. 그러므로, 하나의 하드웨어 구현을 encryption 과 decryption 모두에서 사용할 수 있다.

2. CTR-AES 구조

2.1 Subbyte transformation

Subbyte transformation 에서는 일반적으로 LUT를 사용하는 경우도 있으나 면적을 줄이고 메모리 요구를 없애기 위하여 우리는 Galios Field 연산을 사용하였다. 기본 연산은 $GF(2^8)$에서 multiplicative inverse를 계산하고 affine transformation을 적용한다. 그러나 $GF(2^8)$에서의 multiplicative inverse 계산은 많은 비용 때문에 $GF(2^4)$을 이용하여 $GF(2^8)$을 계산하였다.

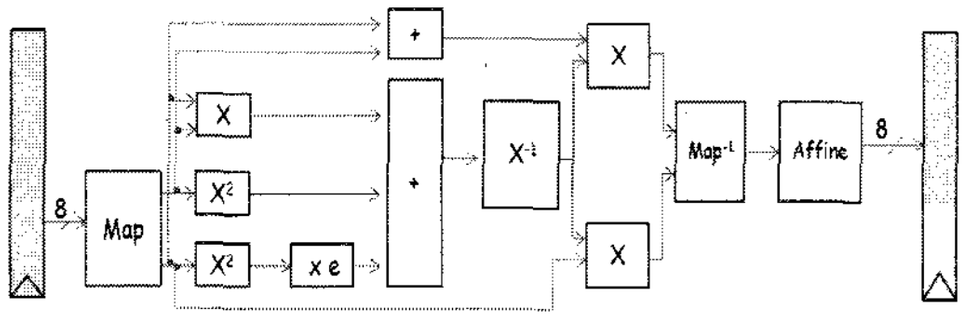

그림 2. GF 연산을 이용한 Non-pipelined S-box
Fig. 2 Non-pipelined S-box using GF operation

그림 2는 GF 연산을 이용한 non-pipelined S-box 구조를 보여주고 있는데 입력 값은 $GF(2^4)$의 두 연산으로 나누어지고 난 후 multiplicative inverse 값이 계산되어진다. 이 계산되어진 두개의 $GF(2^4)$ 값들은 다시 $GF(2^8)$으로 mapping 되어 지고 마지막 과정으로 affine transformation 이 수행되어진다.

비록 S-box를 위한 composite field 구현이 면적 면에서 효과적이긴 하지만 긴 critical path의 문제점이 있다. 이러한 단점을 해결하기 위하여 파이프라인 구조를 사용하는데 적절한 위치에 사용하지 못하는 파이프라인 구조는 레지스터의 사용을 증가시키므로 면적의 증가를 가져오는 문제점이 있다.

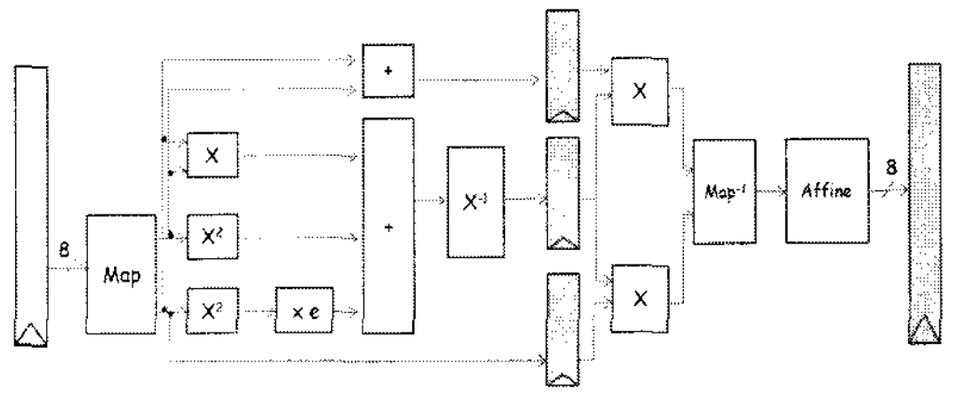

그림 3. 2-stage 파이프라인 S-box
Fig. 3 2-stage pipeline S-box

그림 3은 2-stage pipeline 구조를 보여주고 있는데 critical path는 반으로 줄어들었고 단지 3개의 4bit 레지스터만 사용되었다. 또한 그림 4는 3 stage pipeline 구조를 보여주고 있다[4].

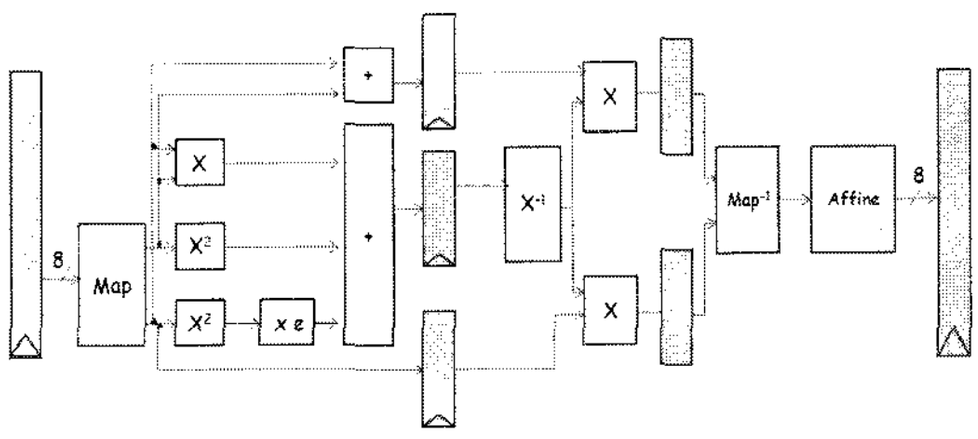

그림 4. 3-stage 파이프라인 S-box
Fig. 4 3-stage pipeline S-box

2.2 MixColumn

Mixcolumn transformation에서는 column의 영역은 $GF(2^8)$ 영역의 다항식으로 간주하고 고정된 다항식 c(x) = '03' x^3 + '01' x^2 + '01' x + '02 을 가진 modulo x^4+ 1를 곱하여 계산한다. 기본 MixColumn transformation은 식(1)과 같이 표현할 수 있다.

$$\begin{cases} s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c}) \\ s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}) \end{cases} \quad (1)$$

MixColumn transformation 구현을 위한 몇몇 구조들이 제안 되어졌으며 그 중 substructure-shared 구조들이 구현되어졌다[4][6][7][9]. 우리의 구조에서도 substructure sharing 기법을 적용하였으며 이 기법을 적용하기 위하여 (1)의 식은 다음의 형태로 바꾸어 져야 한다.

$$\begin{cases} s'_{0,c} = \{02\} \bullet (s_{0,c} \oplus s_{1,c}) \oplus s_{1,c} \oplus (s_{2,c} \oplus s_{3,c}) \\ s'_{1,c} = \{02\} \bullet (s_{1,c} \oplus s_{2,c}) \oplus s_{0,c} \oplus (s_{2,c} \oplus s_{3,c}) \\ s'_{2,c} = \{02\} \bullet (s_{2,c} \oplus s_{3,c}) \oplus s_{3,c} \oplus (s_{0,c} \oplus s_{1,c}) \\ s'_{3,c} = \{02\} \bullet (s_{3,c} \oplus s_{0,c}) \oplus s_{2,c} \oplus (s_{0,c} \oplus s_{1,c}) \end{cases} \quad (2)$$

식(2)의 MixColumn transformation 공식은 대칭적 구조를 가지고 있으며 하드웨어 구현의 면적 최적화를 위한 substructure sharing를 적용할 수 있게 되었다. {02} 상수 곱셈은 a= xtime(b) 함수에 의해 계산되어 지는데 함수 xtime()는 left shift 와 subsequent 조건적 bitwise XOR로서 바이트 level에서 구현 가능하다. 효과적인 xtime() 구조와 XOR-sharing을 적용하므로 최적의 MixColumn transformation을 구현할 수 있으며 구조는 그림 5와 같다.

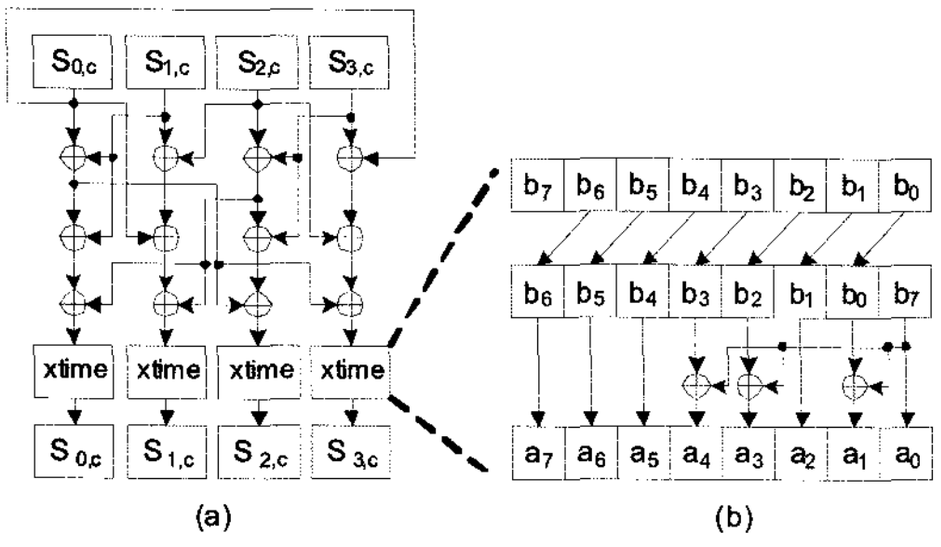

그림 5. (a) MixColumn의 효과적인 구조
(b) xtime() 함수 구현 구조
Fig. 5 (a) an efficient architecture of MixColumn
(b) implementation architecture of xtime() function

2.3 Key Expander 블록

128 bit AES 알고리즘에서 Key expansion 함수는 초기 키 값으로부터 11 라운드 키 값을 생성한다. 파이프라인 AES 구조의 경우, 모든 라운드 키 값은 동시에 가능해야만 한다. 그러므로 몇몇 연구자들은 첫 번째 라운드 키 값을 계산한 후 10라운드를 위하여 하드웨어를 10번 반복 복사하여 구현하였다[4][5]. 이러한 구조는 동시에 모든 라운드 키 값을 계산할 수 있는 장점이 있지만, 많은 면적을 소모하게 된다. Xinmiao Zhang는 on-the-fly 방법에서 동작할 수 있는 key expander를 제안하였다[9].

이 논문에서 우리는 면적 효과적이며 on-the-fly 방법으로 라운드 키를 계산할 수 있는 key expander를 구현하였다. Sub-pipelined 라운드 프로세서 대칭적 동작을 위하여 r개의 sub-stage로 key expander를 나누었다.

우리는 11 라운드 키 값을 저장하기 위하여 11 레지스터를 사용하는데 이 구조는 [9]에서 사용하는 구조와는 다르다. [9]에서 사용된 구조는 모든 라운드 키 값과 내부 pipeline stage를 위한 임시 값 저장을 위하여 r sets의 레지스터를 사용하였는데 이 방법을 이용하여 우리는 면적을 줄일 수 있었다.

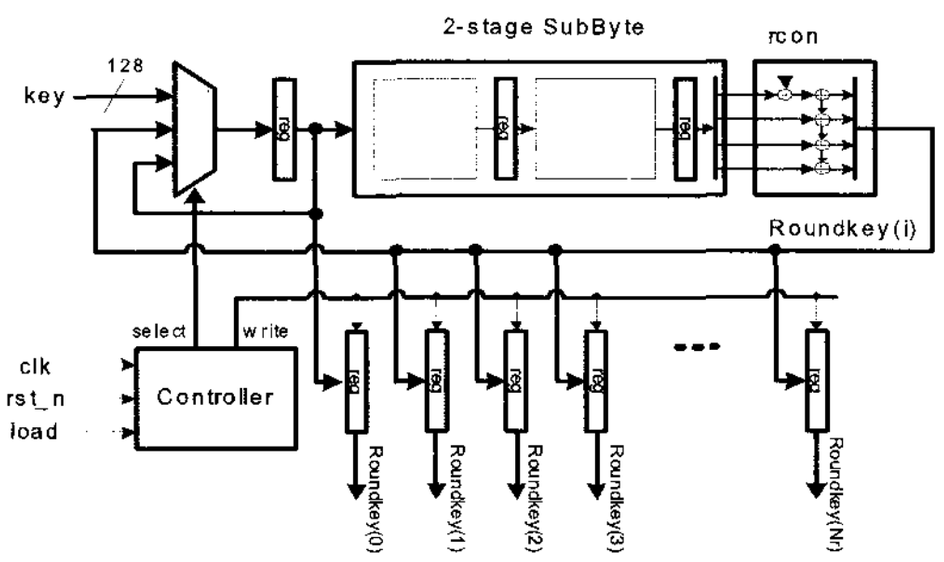

그림 6. On-the-fly key expander 구조
Fig. 6 Architecture of on-the-fly key expander

3-stage를 갖는 on-the-fly key expander 용 sub-pipelined 구조는 그림 6과 같다.

III. 구현 및 성능 비교

우리는 Virtex-4 VLX100-12 에서 128 bit CTR-AES의 full pipelined 구조를 구현하였다. Xilinix ISE 8.2i를 이용하여 합성하였고 post placement timing를 확인하였다. 암호화 및 복호화 동작의 시뮬레이션을 위하여 ModelSim 5.8c을 사용하였다.

이 논문에서는 3개의 sub-pipeline과 5개 sub pipe line 구조를 가지고 있는 두 개의 full pipeline 구조의 AES core를 구현하였다. 3개 sub-pipeline을 가진 AES는 31stage를 필요로 한다. 그러므로 31 클럭 후에 암호화 된 블록값을 얻을 수 있다. 이 구조를 사용하여 26.7Gbits/s

의 데이터 전송율을 얻을 수 있었다. 또한 5개의 sub-pipeline 성능은 3개의 sub- pipeline 보다 더 좋은 성능을 보였다. 그러나 5 sub-pipeline은 더 많은 면적과 전력을 소모하였다. 표 1은 기존의 AES 구현과 우리의 AES 구현을 비교하였다. LUT's, 레지스터, slides, 그리고 최대 주파수에 관점에서 우리의 설계를 평가하였다.

표 1. FPGA 기반 AES 알고리즘 구현 비교
Table. 1 Comparison of FPGA implementation of the AES algorithm

Design	Device	Frequency (MHz)	Throughput (Mbps)	slices	BRAMs	Mbps/slice
Shuenn-Shyang [3]	XCV1000e-8	125.38	1604	1857	0	0.867
Jae-Gon Lee [4]	XCV3200e-8	40	5120	8009	104	0.639
Saqib, N.A. [5]	XCV812e-8	20.192	2584	2744	0	0.942
Jarvinen [7]	XCV1000e-8	129.2	16500	11719	0	1.408
Xinmiao Zhang (r=3) [9]	XCV812e-8	93.5	11965	9406	0	1.272
Xinmiao Zhang (r=7) [9]	XCV1000e-8	168.4	21556	11022	0	1.956
Our AES core design (r=3)	XCV1000e-8	91.1	11661	8914	0	1.308
Our AES core design (r=5)	XCV1000e-8	150.25	19232	9820	0	1.958
Our AES core design (r=3)	XC4VLX100-12	208.49	26686	9478	0	2.816
Our AES core design (r=5)	XC4VLX100-12	247.19	31640	9904	0	3.195

이 논문에서는 3 sub-pipelined stage를 가진 full-pipelined 구조를 사용하였는데 전체 31 stage가 소모되었다. 그러므로 31 clock 사이클 후에 해당되는 입력 값의 암호화 된 블록이 처음 출력되며 매 clock 마다 다음 암호화된 블록을 출력한다. 구현된 구조를 통하여 우리는 26.7 Gbits/s의 throughput을 얻을 수 있었다.

기존의 AES 는 VirtexE device에서 구현하였으므로 정확한 비교를 위하여 우리의 결과도 VirtexE family에서 다시 합성 및 검증하였다.

Ⅳ. 결론 및 향후 연구 방향

이 논문에서 우리는 AES 알고리즘을 위한 full pipelind 구조를 구현하였다. 우리의 설계에서는 3가지 주요한 특징이 있는데 (1) composite field arithmetic SubByte, (2) 면적 효과적인 MixColumn, (3) on-the-fly sub-pipeline 된 key expander 이다. 전체 31 stage가 소모되었으며 Virtex4 VLX 100 device에서 26.7 Gbits/s throughput을 구현하였다. 그러므로 우리 구조는 GPON system의 암호화에 적당한 구조이다.

감사의 글

본 연구는 2007년도 한국전자통신연구원의 지원에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다.

참고문헌

[1] "Gigabit-capable Passive Optical Networks (G-POPN) : Transmission convergence layer specification", ITU-T G.984.3 Amendnment 1, July. 2005

[2] Morris Dworkin, "Recommendation for Block Cipher Modes of Operation", NIST Special Publication, http://csrc.nist.gov/CryptoToolkit/modes/, 2001

[3] Shuenn-Shyang Wang; Wan-Sheng Ni, "Aneffici-ent FPGA implementation of advanced encryption standard algorithm", Proceedings of the 2004 International Symposium on Circuits and Systems, Vol 2, 23-26 May 2004 Page(s):II 597-600 Vol 2.

[4] Jae-Gon Lee, Woong Hwangbo, Seonpil Kim, Chong-Min Kyung, "Top-down implementation of pipelined AES cipher and its verification with FPGA-based simulation accelerator", 6th International Conference On ASIC Proceedings, page(s): 68- 72, 24-27 Oct. 2005.

[5] Saqib, N.A; Rodriguez-Henriquez, F ; Diaz-Pere, A. " ES algorithm imlementation-an efficient approach for sequential and pipeline architectures", Proceedings of the Fourth Mexican International Conference on Computer Science, page(s): 126 ~ 130, 8-12 Sept. 2003

[6] Nedjah, N.; de Macedo Mourelle, L.; Cardoso, M.P., "A Compact Pipelined Hardware Implementation of the AES-128 Cipher", ITNG 2006. Third International Conference on Information Technology: New Generations, page(s):216 - 221, 10-12 April 2006.

[7] Yongzhi Fu; Lin Hao; Xuejie Zhang; Rujin Yang "Design of an extremely high performance counter mode AES reconfigurable processor", Second International Conference on Embedded Software and Systems , 16-18 Dec. 2005 Page(s):7 pp.

[8] Hodjat, A.; Verbauwhede, I., "Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors", IEEE Transactions on Computers, Volume 55, Issue 4, page(s):366 - 372, April 2006

[9] Xinmiao Zhang; Parhi, K.K., "High-speed VLSI architectures for the AES algorithm", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol 12, Issue 9, page(s): 957 - 967, Sept. 2004

[10] V. Rijmen, "Efficient Implementation of the Rijndael SBox", http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

[11] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC Implementation of the AES Sboxes", Proc. RSA Conf. 2002, Feb. 2002

저자소개

박주현(Park,Ju Hyun)

1999년 조선대학교 전자공학과 (공학사)
2002년 전남대학교 전자공학과 (공학석사)
2006년~현재 전남대학교 컴퓨터공학과 박사과정
※관심분야 : 디지털 시스템 설계, 임베디드SoC 설계, 저 전력 설계, DSP

Truong Quang Vinh

1999년 Ho Chi Minh university 전자공학과(공학사)
2003년 AIT, Bangkok, Thailand 전산학과 (공학석사)
2006년 ~ 현재 전남대학교 컴퓨터공학과 박사과정
※관심분야 : 디지털 시스템 설계, 임베디드 SoC 설계, 저 전력 설계, DSP

김광옥(Kim, KwangOk)

2001년 전남대학교 전자공학과 (공학석사)
2001년~현재 : 한국전자통신연구원 광통신연구센터선임연구원
※관심분야 : 광가입자 망 기술, FTTH 서비스기술, 암호화 기술

김영철(Kim, Young Chul)

1981년 한양대학교 전자공학과 (공학사)
1987년 University of Detroit, M.S
1993년 Michigan State University Ph.D
1993년 ~ 현재 전남대학교 전자컴퓨터공학부 교수
2004년 ~ 현재 전남대학교 LG이노텍 연구개발 지원센터 소장
※관심분야 : 임베디드 SoC 설계, 저 전력 설계