저비용 FPGA를 이용한 AES 암호프로세서 설계 및 구현

호정일° 이강 조윤석
hinansaint@hanmail.net°, {yk, yscho}@handong.edu

A Design and Implementation of AES Cryptography Processor using a Low Cost FPGA chip

Jung Il Ho°, Kang Yi, and Yun Seok Cho
School of Computer Science and Electronic Engineering, Handong Global University

요 약

본 논문의 목적은 AES(Advanced Encryption Standard)로 선정된 Rijndael 암호 및 복호 알고리즘을 하드웨어로 설계하고 이를 저비용의 FPGA로 구현하는 것이다. 설계된 AES 암호프로세서는 20만 게이트 급 이하의 FPGA로 구현한다는 비용의 제약 조건 하에서 대용량의 데이터를 암호화, 복호화 하기에 적합한 성능을 가지도록 하였다. 또한 구현 단계에서는 설계한 AES 암호프로세서와 UART 모듈을 동일 FPGA 상에서 통합하여 실용성 및 면적 효율성을 보였다. 구현된 Rijndael 암호 프로세서는 20만 게이트를 갖는 Xilinx 사의 Spartan-II 계열의 XC2S200 칩 사용시 53%의 면적을 차지 하였고, Static Timing Analyzer로 분석한 결과 최대 29.3MHz 클럭에서 동작할 수 있고 337Mbps의 최대 성능을 가진다. 구현된 회로는 실제 FPGA를 이용하여 검증을 수행하였다.

1. 서론

암호화 기술은 인터넷처럼 공개된 망에서 데이터 송수신시 발생할 수 있는 보안문제를 해결하는 방법으로 사용되며, 현재까지 여러 암호화 알고리즘이 개발되어 쓰여지고 있다. AES(Advanced Encryption Standard)는 대칭형 암호 알고리즘으로서 기존의 대칭형 표준 암호 알고리즘인 DES를 대체할 목적으로 개발되었다. Rijndael 은 2001년 11월에 미연방표준(FIPS 197)으로 제정 AES로 선정되었다[1].

본 논문에서는 표준 AES로 선정된 Rijndael 암호 알고리즘을 저가의 FPGA로 구현하였다. 기존의 하드웨어 설계 결과 [2][3] 등이 있으나 본 논문에서는 소규모 내장형 시스템에서 사용하기 위하여 저가의 FPGA로 구현하는 것을 목표로 하였다. 설계된 회로는 동일 칩에 UART와 통합되어 면적 효율성과 실용성을 보여주었다. 이 칩을 이용한 데이터 암호화, 복호화 하는 과정은 PC와 직렬 포트로 연결된 FPGA 보드를 이용하여 검증되었다.

본 논문의 구성은 다음과 같다. 2장에서는 AES로 선정된 Rijndael 알고리즘에 대해서 설명하고 3장에서는 AES 암호 프로세서의 내부 구조에 대하여 설명한 후 4장에서는 FPGA에서의 구현 결과와 검증 및 성능을 살펴보며 5장에서는 결론을 맺도록 한다.

2. AES - Rijndael 알고리즘

2.1 Rijndael 알고리즘의 특징

Rijndael 알고리즘은 128, 192, 256 비트 단위의 여러 가지 길이 텍스트와 암호/복호 키를 수용하는 유연성을 가지면서 기존의 공격(linear 및 differential 공격 등)에 안전하도록 설계되어 있다. 또한 일반 블록암호 알고리즘과는 달리 각 라운드(round) 변환이 Feistel 구조가 아니라 계층(layer)구조를 이루고 있어 여러 라운드 수행시 높은 확산 효과를 보장하고 있으며 각 계층은 역변환이 가능한 uniform 변환(*transformation*)을 지원하고 있다.

2.2 Rijndael 암호화 연산 과정

그림 1은 Rijndael 알고리즘을 이용하여 암호화 하는 전체 블록도이다. 암호화 과정은 Regular Round를 정해진 횟수만큼 반복한 뒤에 Final Round를 실행하여 수행된다. 각 라운드에 사용되는 라운드 키는 암호화 키를 이용하여 Key Generation 모듈에서 계산된다.

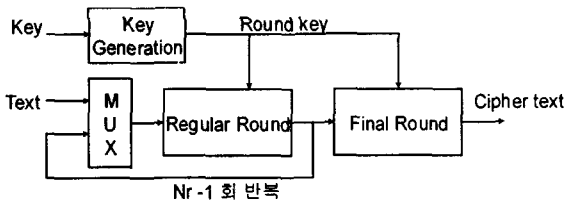

그림 1. 전체 암호화 과정

그림 2는 그림 1의 regular 라운드 블록의 내부 구조를 보여준다. Regular 라운드 과정은 4개의 연산을 반복하

는 구조로 되어 있으며 반복회수 Nr는 키값의 크기에 의해 정해진다 (128비트 키의 경우 Nr=10). Final Round는 Regular Round에서 MixColumn을 제외하면 된다.

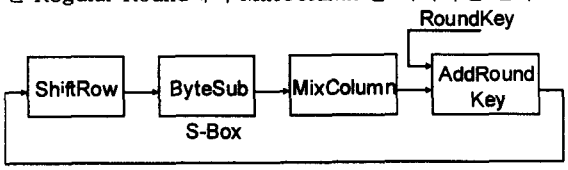

그림 2. Regular Round 과정

2.3 각 계층(layer)의 연산정의

입력된 데이터의 크기를 128비트로 가정하자. 128비트 데이터 블록을 각 원소가 바이트인 4*4 행렬로 간주될 수 있다. ByteSub은 이 데이터 블록을 나타내는 행렬의 각각의 원소를 그림 3처럼 주어진 일대일 변환표에 의해서 다른 8비트 값으로 치환하는 연산이다.

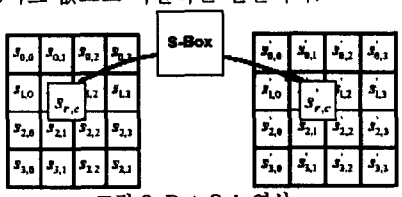

그림 3. ByteSub 연산

MixColumn은 행렬의 32 비트 열(column) 데이터를 그림 5의 행렬식에 의해서 새로운 32비트 값을 얻는 연산이다. 각 열에 대한 치환은 그림 4와 같이 이루어진다.

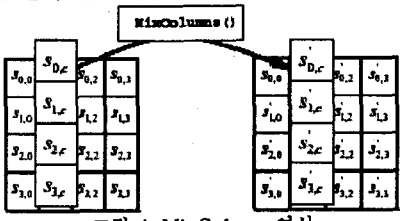

그림 4. MixColumn 연산

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

그림 5. 각 Column 에 대한 Mix Column 연산식

ShiftRow연산은 그림 6과 같이 32 비트 행(row) 데이터에 가해지는 회전변환(rotate)이다.

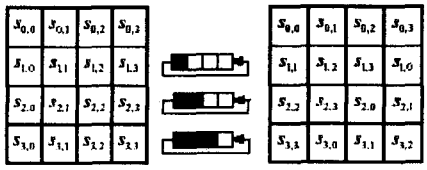

그림 6. ShiftRow 연산

3. Rijndael 의 하드웨어 설계

3.1 Rijndael 암호프로세서의 전체 블록도

그림 7은 설계된 AES 암호프로세서 내부 데이터 경로의 구조를 보여준다. 설계 단순화를 위해 여기서는 키 사이즈와 텍스트 블록 사이즈를 128 비트로 고정하였다. 면적 요구량을 최소화하기 위해서 공유 기능 블록을 최대한 추출하여 재활용하는 루프(loop) 구조를 채택하였다.

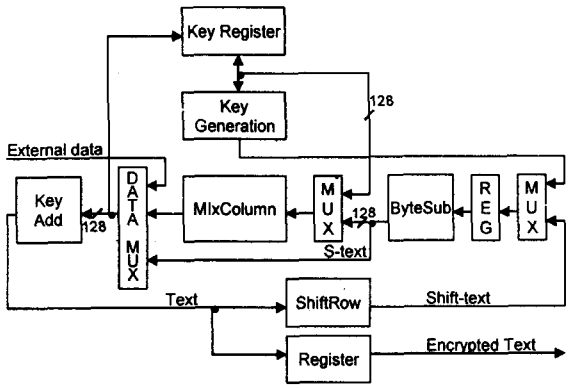

그림 7. Rijndael 암호프로세서 블록도

3.2 최소면적을 얻기 위한 모듈별 설계 방법

3.2.1 FPGA특성을 이용한 설계

ByteSub 연산은 S-Box라는 lookup 테이블을 이용하여 구현되며, 한 바이트의 ByteSub 연산과 InvByteSub 연산을 위해서는 512바이트(=256*8*2비트)의 ROM이 필요하다. 따라서 128 비트 전체를 위해서 16x512바이트가 필요하지만, 본 설계에서는 FPGA에 자체 내장되어 있는 이중 포트 방식의 블록 ROM을 이용하여 ROM의 필요량을 8x512 바이트로 줄였다.

3.2.2 간단한 조합 논리회로를 이용한 상수 곱셈기

이 모듈은 MixColumn() 연산과 Inv_MixColumn() 연산을 수행하는데, Rijndael에서의 모든 산술 연산은 바이트를 기본 피연산자로 하여 수행되는 $GF(2^8)$ 에서의 연산이다. Rijndael 알고리즘에서 사용되는 덧셈(⊕)은 비트 단위의 EXOR 연산이며 곱셈(⊗)은 degree가 8인 다항식 M(x) 와의 MODULO 연산이다. M(x)는 다음 식과 같다.

$$M(x) = X^8 + X^4 + X^3 + X + 1$$

M(x)가 1바이트 보다 크기 때문에 곱셈 결과는 바이트의 크기를 넘을 수 없다. 한편, 2를 곱하는 연산은 좌측 1비트 쉬프트 연산으로 대체 가능하며 3을 곱하는 연산은 2배 연산과 자신을 더하는 연산으로써 구현이 가능하다. 예를 들어, M ⊗ 7을 계산하는 과정은 다음과 같이 배분법칙을 이용하여 다시 쓸 수 있다.

M ⊗ 7 = M ⊗ (4 ⊕ 2 ⊕ 1)
= (M ⊗ 4) ⊕ (M ⊗ 2) ⊕ (M ⊗ 1)
= (M ⊗ 2 ⊗ 2) ⊕ (M ⊗ 2) ⊕ (M ⊗ 1)

이러한 GF(2^8) 에서의 곱셈 연산 성질을 이용하여 상수와의 곱셈기 구현은 2배를 해 주는 곱셈 연산기 (쉬프트기)와 덧셈기 (EXOR)만 있으면 되므로 회로의 구성이 간단해진다. 그림 8은 조합논리로 구현된 ⊗ 2를 수행하는 모듈의 내부를 보여준다. EXOR의 입력 1B는 MODULO 연산을 위한 M(x)의 X^4+X^3+X+1부분 값으로 2:1 MUX와 더불어 MODULO M(x) 연산을 구현한다.

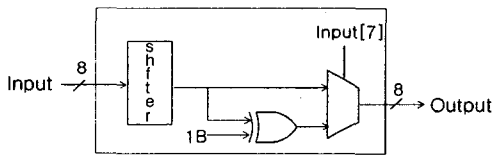

그림 8. ⊗ 2 모듈의 내부 모습

그림 4의 배열에는 01,02,03의 바이트 상수만 존재하므로 첫 열을 계산하기 위해 그림 9와 같이 1배, 2배, 3배를 하는 조합논리회로를 구현하면 MixColumn 연산을 구현하게 된다. 그림 9의 X2 모듈은 그림 8을 사용한다.

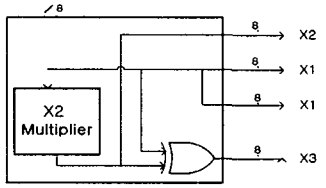

그림 9. T_0 블록의 내부 모습

그림 10에 있는 T_i블록은 그림 9의 회로를 이용하여 구현된다. 그림의 윗부분은 암호화를 위한 부분이고 아래는 복호화를 위한 역변환식 계산 (InvMixColumn) 부분이다.

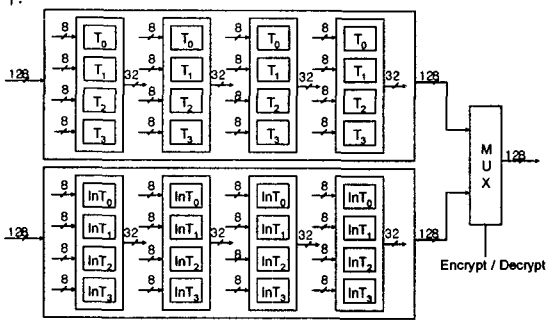

그림 10. MixColumn 모듈 내부

3.2.3 공유 면적의 최대화와 처리량 최대화를 위한 고려

공유 블록을 많이 늘리기 위해서 Regular Round와 Final Round를 공유하도록 설계하였고, 라운드별 키값 생성을 위한 Key Generation 모듈에 사용되는 ByteSub과 MixColumn은 따로 두지 않고, Regular Round를 위한 부분과 공유하였다. 먼저 생성된 키는 KeyRegister에 저장하여 두고 라운드 계산에서 차례로 읽어내어 연산에 사용한다. 키값은 데이터에 비해서 자주 바뀌지 않고 반복하여 사용되기 때문에 키값 계산과 데이터 계산을 순차적으로 하여도 성능에는 지장이 없다. 데이터 경로의 폭을 칩의 면적이 허용하는 한 크게 설정하여 128 비트로 설계함으로써 각 라운드를 수행하는 데 필요한 클럭 사이클 수를 최소화하여 면적 제약조건 하에서 가능한 높은 데이터 처리량을 얻고자 하였다. 즉, 그림 7에서 Regular Round의 루프 부분에 레지스터를 하나만 두어서 1 클럭 사이클에 한 라운드의 수행이 끝날 수 있도록 하였다.

4. FPGA 에서의 구현 및 검증

4.1 FPGA로의 구현

Rijndael 프로세서를 FPGA 보드 상에 구현하였고, PC와 데이터를 주고 받을 수 있도록 UART도 통합하여 구현하였다. FPGA는 20만 게이트 크기를 갖는 Xilinx 사의 XC2S200을 목표로 하였다. 본 설계는 VHDL로 기술되어 ModelSim XE II로 시뮬레이션하여 검증하였으며, Xilinx ISE 5.2i를 사용하여 합성 및 구현하였다. Rijndael 암호 프로세서를 합성한 결과 전체 칩의 53% 면적을 차지하였고 최대 동작주파수는 29.3MHz였다 이로부터 최대 처리량을 계산해 보면 약 337Mbps 이다., UART까지 통합한 설계 결과는 58% 정도의 면적을 차지하였다.

4.2 Rijndael 암호프로세서의 하드웨어 검증

검증을 위해서 UART 와 통합된 Rijndael 프로세서를 실제 FPGA 보드 상에서 구현하였다. Rijndael 프로세서에 사용된 외부클럭 속도는 25MHz이었으며 (처리량은 약 290Mbps) 직렬 포트를 이용하여 데이터를 전송받았다. 암호프로세서의 동작을 검증하기 위해 별도의 직렬통신 프로토콜과 구동소프트웨어를 제작하여 이용하였다. 이 소프트웨어는 PC 내부에 있는 파일들을 직렬 포트로 Rijndael 암호프로세서에 전달하고 암호/복호화된 데이터를 다시 파일로 저장하는 역할을 한다. 파일의 내용을 비교하여 암호/복호 연산의 성공을 확인할 수 있었다.

5 결론

본 논문에서는 차세대 암호화 알고리즘인 Rijndael 알고리즘을 개당 삼만원 이하의 비용이 드는 저비용 FPGA에서 구현하면서 매우 높은 처리 능력을 얻도록 하였다. 동일 블록의 추출과 상수 행렬 곱셈 연산을 간단한 논리연산으로 대체하고 FPGA의 특성을 고려한 설계로써 저가 FPGA의 면적 요구량과 성능을 동시에 만족시킬 수 있었다. 구현된 Rijndael 프로세서는Xilinx XC2S200 칩의 53% 면적만을 사용하여 구현되었고 337Mbps의 성능을 보였다. 설계결과는 FPGA보드를 통하여 검증되었다.

참고문헌

[1]Joan Daemen, Vincent Rijmen "The Rijndael Block Cipher," AES Proposal pp. 7-8, March 1999.

[2]김방현, "파이프라이닝을 이용한 AES 암호화 알고리즘의 FPGA 구현," 한국정보과학회 논문지: 컴퓨팅의 실제, 8권 6호, pp. 717-726, 2002. 12.

[3] N. Sklavos, O. koufopaviou "Architecture and VLSI Implementation of the AES-Proposal Rijndael," IEEE TRANS ON COMPUTERS, 2002. 12.