2000年度 大韓電子工學會 秋季綜合學術大會 論文集 第23卷 第2號 2000/11

단일 라운드 프로세스 방식의 IDEA 암호 알고리즘의 하드웨어 설계

* 최 영 민*, 권 용 진*
* 한국항공대학교 통신정보공학과

A VLSI Design of IDEA Cipher Algorithm Based On a Single Iterative Round Method

* Choi, Young-Min*, Kwon, Yong-Jin*
* Depatment of Telecommunication and Information Engineering, Hakuk Aviation University
E-mail : choiym@mail.hangkong.ac.kr

Abstract

Data security is an important issue in today's computer networks. In order to construct a safe infra in the open communication network, a cryptography is necessarily applied to several communication application fields like a high-speed networking system supporting real-time operation. A cryptography which has already realized by a software is designed by using a hardware to improve a throughput.

In this paper, we design hardware architecture of IDEA by using a single iterative round method to improve a encryption throughput. In addition, we intend to develop a hardware design methodology that a specific cryptography operate with high-speed. The hardware model is described in VHDL and synthesized by the Samsung KG 80 Library in the Synopsys development software tool. With a system clock frequency 20MHz, this hardware permits a data conversion rate of more than 116 Mbit/s.

I. 서론

전자상거래 및 홈뱅킹이 인터넷이나 전화망과 같은 공중망을 통하여 행해지고 있고, 인터넷을 통한 전자메일 및 파일의 교환이 빈번해짐에 따라 거래 정보, 전자메일, 파일을 변조 및 유출하여 악용하려는 시도들이 끊임없이 이어지고 있다. 따라서 이러한 시도로부터 데이터의 무결성을 유지하기 위해서 암호를 통한 정보보호의 필요성이 크게 부각되고 있다. 그러나 기존의 소프트웨어적인 방식으로 구현된 암호 알고리즘은 암호 처리 속도가 느린 단점으로 인해, 대량의 데이터를 실시간으로 처리하는 여러 응용분야에 실제적으로 적용하기란 어렵다. 이를 해결하기 위해 최근 암호 알고리즘을 하드웨어로 구현하여 처리 속도를 향상시키는 연구가 활발히 진행중이다[2][4].

본 논문에서는 이미 잘 알려진 DES보다 암호학적으로 더 안전한 IDEA를 ASIC 설계하여 처리 속도를 향상시키고자 한다. 뿐만 아니라 이 연구를 통해 특정 암호 알고리즘을 고속 하드웨어로 설계하는 방법론을 개발하고자 한다. 이를 위해 IDEA를 기능에 따라 암·복호 처리 부분, 키 생성 부분, 제어 부분, 입·출력 부분으로 나누어 설계하고 있다. 암·복호 처리 부분은 64비트의 평문을 암·복호화하는 기능을 수행하며, 이를 단일 라운드 반복 방식으로 설계하여 회로 크기를 줄이고 있고, 라운드 내에 정의된 복잡한 연산인 법 $(2^{16}+1)$ 대한 곱셈은 low-high algorithm[3]을 사용하여 효율적으로 설계하고 있다. 키 생성 부분은 평문의 암·복호 처리를 위해, 각 라운드에 필요한 키를 생성하며, 암호 키는 입력받은 키를 시프트하여 생성하고, 복호 키는 암호 키에 대한 법 (2^{16})에 대한 덧셈의 역원, 법 $(2^{16}+1)$에 대한 곱셈의 역원의 조합으로 생성된다. 법

$(2^{16}+1)$에 대한 곱셈의 역원은 Fermat 정리[5]를 응용하여 최적화된 하드웨어를 얻고 있다. 제어 부분은 IDEA의 전체 흐름을 제어하고, 입·출력 부분은 PC와 데이터의 송·수신 가능하도록 설계하고 있다. 그리고 이러한 서브 블록은 표준 하드웨어 설계 언어인 VHDL(VHSIC Hardware Description Language)를 이용하여 top-down 설계 방식으로 논리 설계하고, 이를 Synopsys 설계 환경에서 삼성 SOG 설계 라이브러리 kg80으로 합성하고, VSS를 이용하여 pre-layout 타이밍 시뮬레이션을 수행하고, 실제로 Xilinx FPGA XC4062XL 디바이스에 블록별로 칩 테스트를 하여 올바르게 동작함 확인한다. 설계한 회로의 최대 동작 주파수는 20 MHz이고, 게이트 크기는 118,774 gate이며, 처리 속도는 116 Mbit/sec로 동작한다. 동일한 설계 방식을 사용하고 있는 [3]보다 처리속도가 2 배 빠른 것을 확인했으며, 지금은 IDEC 주관 제 9회 MPW 삼성 SOG 칩 제작 분야에 선정되어 ASIC 설계하고 있다.

2 장에서는 IDEA 암호 알고리즘을 소개하고, 3 장에서는 IDEA 암호 알고리즘의 블록별 설계 방법과 효율적인 내부 구조를 제안하고, 4 장에서는 회로 검증 및 기존회로와의 성능 비교를 하며, 5 장에서는 결론을 맺는다.

II. 이론적 배경

IDEA는 64 비트의 데이터 블록을 암·복호화하기 위해 128 비트의 키를 사용하는 블록 암호 방식으로, Xuejia Lai와 James Massey에 의해 제안되었다[1]. 암·복호화 블록은 8개의 라운드와 출력 변환으로 이루어져 있고, 각 라운드는 치환과 MA구조로 이루어져 있으며, 암호학적 강도를 증가시키기 위해 대수적 구조가 서로 다른 비트 대 비트 xor 연산, 법 2^{16}에 대한 덧셈, 법 $(2^{16}+1)$에 대한 곱셈 연산의 조합으로 실제 구성되어 있다. 또한 라운드는 평문과 서브키를 입력받아 비트 대 비트 xor 연산, 법 (2^{16})에 대한 덧셈, 법 $(2^{16}+1)$에 대한 곱셈 연산의 조합으로 출력문을 생성하고, 이 출력문은 다시 다음 라운드의 입력으로 사용된다. 이 동일한 라운드 구조를 8 회 수행하여 암·복호문이 생성되게 된다. IDEA의 암호 과정과 복호 과정에 사용되는 암·복호 구조는 그림 1로 유일하며, 키 데이터에 따라 암호문이 출력되기도 하고, 복호문이 출력되기도 한다.

IDEA는 128 비트의 키 데이터를 입력받아 52 개의 16 비트 서브키를 생성한다. 서브키에는 암호화에 사용되는 암호 서브키와 복호화에 사용되는 복호 서브키가 있다(그림 2). 암호 키 생성 방식은 128 비트 입력을 16 비트씩 분할하여 8 개의 서브키를 만들어내고, 이 과정이 끝나면 25 비트씩 왼쪽으로 시프트하여 나머지 서브키를 만들게 된다. 이러한 과정을 52 개의 서브키가 만들어질 때까지 반복 수행한다. 복호 키 생성 방식은 암호 키 생성 방식에 좀더 복잡한 연산이 첨가되어 있다. 즉, 복호 라운드 i에서 사용되는 처음 4 개의 복호 서브키는 (10-i) 번째 암호 라운드에서의 처음 4 개의 서브키로부터 유도되고, 마지막 두 개의 서브키는 (9-i) 번째 암호 키에서 유도된다. 여기서 처음과 4 번째 복호 서브키는 처음과 4 번째 암호 서브키의 법 $(2^{16}+1)$에 대한 곱셈의 역원이며, 두 번째와 세 번째의 복호 서브키는 두 번째와 세 번째의 암호 서브키의 법 (2^{16})에 대한 덧셈의 역원으로 변환되어야 한다.

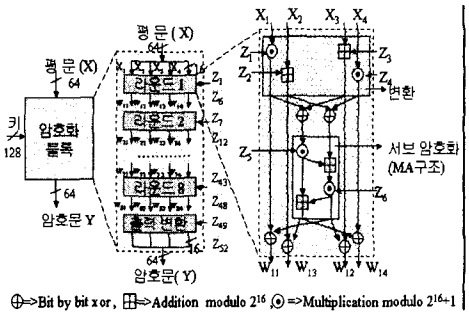

그림 1. IDEA 암·복호 방식의 전체 구조

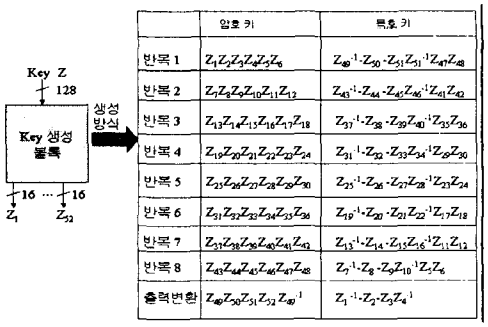

그림 3. IDEA 서브키 생성 규칙

III. 암호 알고리즘의 내부 구조 설계

3.1 암·복호 처리 부분

일반적인 블록 암호 알고리즘의 암·복호 처리 부분은 동일한 동작을 수행하는 여러 개의 라운드로 구성되어 있다. 이러한 암·복호 처리 부분을 하드웨어 설계할 때의 설계 방식으로는 라운드 전체를 직렬로 연결하는 전 라운드 구현형 방식과 하나의 단일 라운드로 여러 번

반복 수행하는 단일 라운드 구현형 방식이 있다[2]. 본 논문에서는 8 회의 라운드 동작으로 구성되는 IDEA의 암·복호 처리 부분을 하나의 단일 라운드 프로세서로 구현하고, 이를 8 회 반복 수행하도록 하는 단일 라운드 반복형 설계 방식을 채택하여 실제 회로의 크기를 줄이고자 한다.

IDEA의 단일 라운드는 6 개의 16 비트 대 비트 xor 연산, 4 개의 법 (2^{16})에 대한 덧셈, 4 개의 법 ($2^{16}+1$) 대한 곱셈으로 정의되어 있다. xor와 덧셈 연산은 비교적 쉬우나, 법 ($2^{16}+1$)에 대한 곱셈은 두 수의 곱셈한 결과를 ($2^{16}+1$)로 나누었을 때, 나머지를 구해야 하는 복잡하고 어려운 연산이다. 특히 ($2^{16}+1$)은 정수론에서 Fermat number중 소수로 알려져 있으며[5], 연산의 복잡성으로 인해 residue number system, 신호 처리, 암호 분야에 많이 적용되고 있다. 법 ($2^{16}+1$) 대한 곱셈의 하드웨어 설계는 복잡도가 높아 구현된 칩의 면적, 동작 속도에 큰 영향을 주므로, 고 성능으로 동작하도록 설계할 필요가 있다. 이 연산을 하드웨어로 설계할 때, 최적의 구조가 되도록, 아래의 low-high algorithm[3]을 사용한다.

$$Z = X \cdot Y \pmod{F_4} \quad [F_4 = 2^{2^4}+1]$$

1. $D = A \cdot B$
   D.lower = D(15 downto 0),
   D.higher = D(31 downto 16)
2. If (D.lower ≥ D.higher) than
   Z = D.lower − D.higher
   else
   Z = D.lower − D.higher + F_4

그림 3.low-high algorithm for $X \cdot Y \bmod F_4$

3.2 키 생성 부분

키 생성 부분은 128 비트의 키 데이터로부터 매 라운드마다 필요한 서브키를 생성한다. 서브키 생성 방식 그림 2에 나타나고 있다. 즉 암호 서브키는 128 비트의 암호 키로부터 차례로 대응함으로써 처음 암호키가 생성되고, 다음에 사용될 암호 키는 현재의 암호 키를 왼쪽으로 25 비트 순환 시프트하여 생성하면 된다. 그리고 복호 서브키는 암호 서브키 생성 규칙에 의해 만들어진 키에 법 (2^{16})에 대한 덧셈의 역원과 법 ($2^{16}+1$)에 대한 곱셈의 역원을 적용하여 생성한다. 그런데 법 ($2^{16}+1$)에 대한 곱셈의 역원은 IDEA에서 가장 복잡도가 높고, 실제 구현된 칩의 면적, 동작 속도에 큰 영향을 주므로, 고 성능으로 동작하도록 설계가 요구된다. 본 논문에서는 법 ($2^{16}+1$)에 대한 곱셈의 역원을 Fermat 정리[5]를 응용하여 효율적으로 설계한다(그림 4). 실제로 법 ($2^{16}+1$)에 대한 곱셈의 역원은 법 ($2^{16}+1$)에 대한 곱셈 Z^2와 법 ($2^{16}+1$)에 대한 Z^2 의 결과와 Z 와의 곱셈을 16 회만큼 반복 수행하면 해결된다.

$$Z \cdot Z^{-1} = 1 \bmod F_4 \quad [F_4 = 2^{2^4}+1]$$

$$Z^{-1} \pmod{F_4} = Z^{F_4-2} \bmod F_4$$
$$= Z^{(1+2+2^2+\cdots+2^{15})} \bmod F_4$$
$$= Z(Z\cdots(Z(Z^2))^{2}\cdots)^2 \bmod F_4$$

그림 4. 법 ($2^{16}+1$)에 대한 곱셈의 역원 식

3.4 제어 부분

제어부분은 IDEA의 암·복호화 수행 시간이 최소가 되도록 암·복호화 처리 부분과 키 생성 부분에 필요한 제어신호를 발생하도록 설계한다. 그림 5에 제어 부분의 내부 블록도가 나타나고, 실제로 클럭 입력에 따라 암호·복호화 알고리즘의 동작을 마이크로 오퍼레이션 레벨으로 기술하여 내부 신호를 발생하도록 설계하고 있다.

3.4 입·출력 부분

설계한 암호 칩이 PC나 16비트 마이크로컨트롤러와 데이터 송·수신이 가능하도록 16 비트 단위로 입력받도록 설계한다(그림 5).

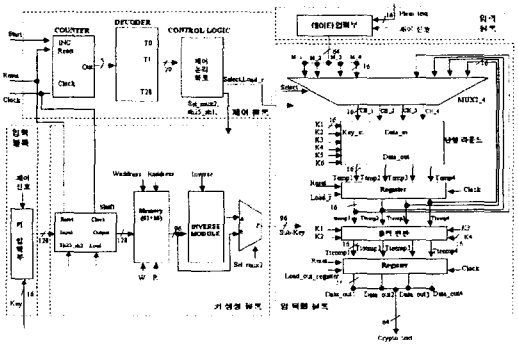

그림 5. 설계한 IDEA의 전체 하드웨어 구조

Ⅳ. 설계 검증 및 성능 비교

IDEA의 암호 알고리즘을 삼성 SOG KG 80 라이브러리로 합성한 후, SADAS3을 동작하여 게이트 딜레이를 추출하여, Synopsys VSS에서 pre-layout 타이밍 시뮬레이션을 수행하고, 이를 소프트웨어로 구현된 IDEA 암호

프로그램과 비교함으로써 설계의 정확성을 검증하고 있다. 여기에 사용된 소프트웨어[6]는 PGP에 구현된 IDEA로서, C언어로 작성되었으며 SUNSPARC20으로 컴파일 및 실행하였다. 또한 IDEA의 서브 블록들을 Xilinx FPGA XC4062XL 디바이스에 프로그램하여 블록별 칩 테스트를 수행하고, 올바르게 동작함을 검증했다.

그림 7은 평문을 $(3736353433323130)_{16}$로, 암호 키를 $(707E6E486C6B6A6968676665646362616 0)_{16}$로 하였을 때, 암호문 $(509A8E0AE6E5EC9C)_{16}$가 출력됨을 나타내는 pre-layout 타이밍 시뮬레이션을 나타내고 있다.

그림 6은 설계한 회로와 기존회로와의 성능 비교를 나타내고 있다. [3]은 단일 라운드 구현형 방식으로 설계되고, [4]는 파이프라인 방식이 적용하여 설계되고 있다. 본 논문의 처리 속도는 동일한 설계방식을 사용한 [3]보다 약 2 배 빠르며, [4]보다 4.5 배 느리고, 게이트 크기는 [3]보다 1.25 배 많으며, 3.3 배 적음을 확인한다.

	area(gate)	throuthtput (Mbit/sec)
본 논문	118,774	116 Mbit/s
[3]	95,000	55 Mbit/s
[4]	400,000	533 Mbit/s

그림 6. 기존 논문과의 성능 비교

V. 결론

개인의 정보가 중요시되는 정보화 시대에 암호를 이용한 정보 보호는 필수적이다. 그러나 현재까지 대부분의 제안된 알고리즘들은 소프트웨어 방식으로 구현되어 왔고, 그 결과 느린 처리 속도로 인해, 실시간으로 데이터를 암·복호해야 하는 여러 응용시스템에 적용하기는 어렵다. 이러한 처리 속도의 문제점을 극복하기 위해 암호 알고리즘의 하드웨어 구현이 요구된다.

본 논문에서는 IDEA 암호 알고리즘을 단일 라운드 반복형 방식으로 하드웨어 설계하고 있다. 이를 위해 IDEA내에 정의된 복잡하고 어려운 연산에 대한 효율적인 방법을 제안하고, PC에 장착하여 데이터 송·수신이 가능하도록 입·출력 인터페이스를 재 정의하고 있다. 그 결과 [3]보다 동작속도가 2 배 높음을 확인했으며, 현재 IDEC 주관 제 9 회 MPW 삼성 SOG 칩 제작 분야에 선정되어 칩 제작중이고 있다.

향후 연구방향으로는 회로의 지연과 면적을 좀더 개선하기 위해, 법 $(2^{16}+1)$ 에 대한 곱셈과 법 $(2^{16}+1)$ 에 대한 곱셈의 역원에 대한 게이트 레벨에서의 최적 알고리즘 연구와 ATM과 FDDI와 같은 고속 네트웍 통신망 등에 적용할 수 있도록, 전 라운드 방식을 사용하여 파이프라인 처리가 가능하도록 하여 암호 처리 속도를 높이는 연구 등이 있다.

참고문헌

[1] X.lai and J.L. Messay, "A proposal for a new block encryption standard", In Advance in Cryptology-EUROCRYPT'90, page 389-404, Berlin,1990,Springer Verlag

[2] H.Bonnenberg, A.Curiger, N.Felber, H.Kaeslin, X.Lai, "VLSI Implementation of a New Block Cipher", 1991 IEEE Int Conf. Computer Design:VLSI in computer and Processor. 1991, pp 510-513

[3] Zhongde Wan, Jullien G.A, Miller W.C. "An algorithm for multiplication modulo $(2^{16}+1)$", Signals, Systems and Computers, 1995. Conference Record of the Twenty-Ninth Asilomar Conference, page 956-960

[4] Stefan Wolter,Holger Matz,Andres Schubert and Rainer Laur, "On the VLSI Implementation of the International Data Encryption Algorithm IDEA",Circuit and Systems, 1995. ISCAS '95,1995 IEEE International Symposium, Page(s): 397 -400 vol.1, 1995

[5] 김용태, 박승안, "정수론", 경문사,1997년 4월

[6] Philip Zimmermann, "PGP Source Code and Internals", MIT Press,1994

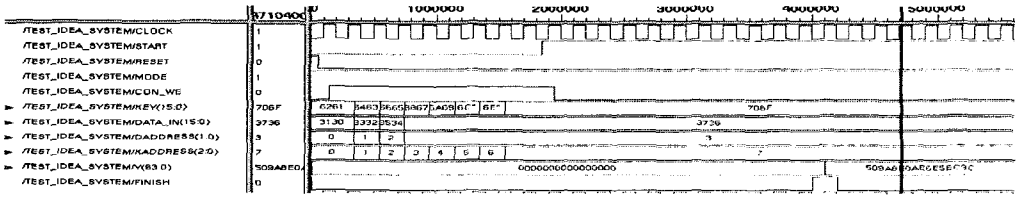

그림 7. Pre-layout timing simulation