JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 20, No. 4 : 786~792 Apr. 2016

Camellia 블록 암호의 암 · 복호화기 코어 설계

손승일*

Design of Encryption/Decryption Core for Block Cipher Camellia

Seungil Sonh*

Department of Information and Telecommunications, Hanshin University, Osan 18101, Korea

요 약

Camellia 암호는 NTT사 및 미쓰비시 전자회사에서 공동으로 2000년도에 개발되었다. Camellia는 128비트 메시지 블록 크기와 128비트, 192비트 및 256비트 키(Key)에 대한 암호화 방식을 규정하고 있다. 본 논문은 키 스케줄용 레지스터 설정과 기존의 라운드 연산 블록을 통합한 수정된 라운드 연산 블록을 제안하였다. 키 생성과 라운드 연산에 필요한 총 16개의 ROM을 단지 4개의 이중포트 ROM만을 사용하여 구현하였다. 또한 메시지 버퍼를 제공하여 키 생성을 위한 KA와 KB 값이 도출되면 대기 시간없이 즉시 암호화나 복호화가 수행될 수 있도록 하였다. 제안한 Camellia 블록 암호 알고리즘을 Verilgo-HDL을 사용하고 설계하고, Virtex4 디바이스상에 구현하였으며, 최대 동작 주파수는 184.898MHz이다. 128비트 키 모드에서 최대 처리율은 1.183Gbps이며, 192비트 및 256비트 키 모드에서 최대 처리율은 876.5Mbps이다. 본 논문에서 설계된 암호 프로세서는 스마트 카드, 인터넷뱅킹, 전자상거래 및 위성방송 등과 같은 분야의 보안 모듈로 응용이 가능할 것으로 사료된다.

ABSTRACT

Camellia was jointly developed by Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation in 2000. Camellia specifies the 128-bit message block size and 128-, 192-, and 256-bit key sizes. In this paper, a modified round operation block which unifies a register setting for key schedule and a conventional round operation block is proposed. 16 ROMs needed for key generation and round operation are implemented using only 4 dual-port ROMs. Due to the use of a message buffer, encryption/decryption can be executed without a waiting time immediately after KA and KB are calculated. The suggested block cipher Camellia algorithm is designed using Verilog-HDL, implemented on Virtex4 device and operates at 184.898MHz. The designed cryptographic core has a maximum throughput of 1.183Gbps in 128-bit key mode and that of 876.5Mbps in 192 and 256-bit key modes. The cryptographic core of this paper is applicable to security module of the areas such as smart card, internet banking, e-commerce and satellite broadcasting.

키워드 : 대칭형 블록암호, 암호화, 복호화, Camellia, 암호 시스템

Key word : Symmetric block cipher, Encryption, Decryption, Camellia, Cryptosystem

Received 05 January 2016, **Revised** 18 March 2016, **Accepted** 01 April 2016

* **Corresponding Author** Seungil Sonh(E-mail:saisonh@hs.ac.kr, Tel:+82-31-379-0659)
Department of Information and Telecommunications, Hanshin University, Osan 18101, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2016.20.4.786 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

암호는 과거에는 군사적인 용도 등의 비밀 통신을 위해 주로 사용하였지만, 오늘날은 인터넷 기반의 사회, 경제 활동의 안정성, 신뢰성, 사생활 보호 등을 위한 핵심 기술로서 메일전송, 사용자 인증, 전자상거래, 인터넷뱅킹 등에서 다양하게 사용되고 있다[1,2].

대칭형 암호 시스템은 암호화를 위한 키와 복호화를 위한 키가 동일한 시스템으로 DES(Data Encryption Standard), IDEA(International Data Encryption Algorithm), SKIPJACK, MISTY, Camellia 및 AES (Advanced Encryption Standard) 알고리즘 등 다양하게 발표되었다[3].

우리나라에서 주관하여 개발된 대칭형 블록 암호는 SEED, ARIA 및 HIGHT가 있으며, 미국 NIST에서 주관한 AES 암호는 세계적으로 널리 알려진 암호 알고리즘이다[2,4,5].

Camellia 암호는 NTT사 및 미쓰비시 전자회사에서 공동으로 2000년도 개발하였다. Camellia는 128비트 블록 크기와 128비트, 192비트 및 256비트 키(Key)에 대한 암호화 방식을 규정하고, AES(Advanced Encryption Standard) 암호화 방식과 같은 인터페이스로 동작하도록 정의하였다. 이 방식은 스마트 카드, 암호 하드웨어, 내장형 시스템 등에서 폭넓게 활용될 수 있도록 고안되었다[6].

최근 무선 인터넷을 기반으로 한 사물인터넷(IoT: Internet of Things)의 분야에서 암호의 중요성이 증대되고 있으며, 전자상거래, 인터넷 뱅킹 등 다양한 분야에서 암호 알고리즘을 사용하기 때문에 암호 알고리즘의 연구는 필요하다. 특히 Camellia 암호 알고리즘은 SSL/TLS용 차세대 인터넷 표준 암호 규격으로 채택되었고, IPSsec, S/MIME, XML과 같은 분야에서도 IETF에 의해 암호 규격으로 채택되었다. 그리고 EU는 전자정부를 위한 권고 암호로 채택한 바 있다. 본 논문에서는 이처럼 다양한 보안 분야에서 응용되는 Camellia 암호 알고리즘 코어를 설계한다. 이 때 자원 공유를 실현해 하드웨어 자원 사용을 최소화하고, 32비트 데이터 입출력을 인터페이스를 갖고 Camellia 암 · 복호화를 모두 지원한다.

본 논문의 2장에서는 Camellia 암호 알고리즘의 개념에 대해 소개한다. 3장에서는 Camellia 암호 알고리즘 기능 로직의 독창적 설계 방법을 제시하며, 4장에서는 설계된 Camellia 암호의 타당성 검증 및 성능 분석을 수행하였다. 마지막으로 5장에서 결론을 맺는다.

Ⅱ. Camellia 암호 알고리즘의 개념[2]

Camellia 암호 알고리즘은 키 스케줄링 부분(Key Scheduling Part)과 데이터 랜덤화 부분(Data Randomizing Part)인 라운드 연산 부분으로 나눌 수 있다. 먼저 키 스케줄링 부분의 구조는 그림 1과 같다. 사용하는 입력 키가 K라 할 때, K가 128비트이면 KL=K, KR=0을 사용해 부분 키(Subkey)를 생성하며, K가 192비트이면 KL=K>>64, KR=((K&MASK64) <<64 ‖ (~(K&MASK64))을 사용하여 부분 키를 생성하며, 마지막으로 K=256비트이면 KL=K>>128, KR= K&MASK128을 사용하여 부분 키를 생성한다. 여기서 MASK64는 64비트가 모두 1인 경우이며, MASK128은 128비트가 모두 1인 경우를 의미한다. 또한 부분 키 발생에 사용되는 128비트 KA와 KB 또한 KL과 KR로부터 생성되며, KB는 암 · 복호화 키가 192비트 혹은 256비트일 때만 사용된다.

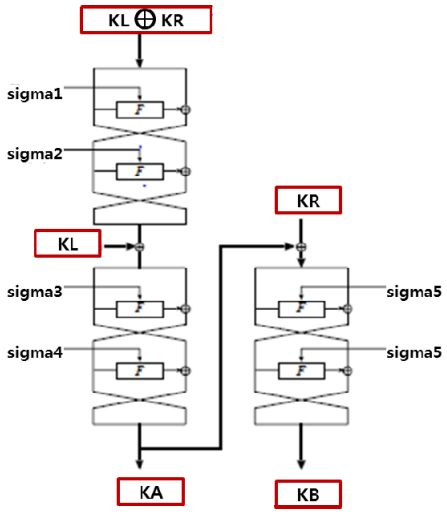

Fig. 1 Block diagram of a key scheduling part

그림 2의 F 함수는 2개의 64비트 입력을 받아 수행하게 되는 데, 하나는 입력 데이터 F_IN이고, 다른 하나는

부분 키인 KE 값이다. 그런데 KE 값은 실제적으로 이미 정의되어 있는 시그마(Sigma) 값을 사용한다. 두 입력 값을 비트 단위 XOR한 후 8비트 단위로 SBOX ROM을 액세스하고, 출력된 8비트 SBOX 출력들 중에서 5개나 6개의 모듬으로 XOR 연산을 수행하여 F 함수 출력인 F_OUT을 도출한다. 이러한 F 함수는 데이터 랜덤화 과정에서도 동일하게 사용된다.

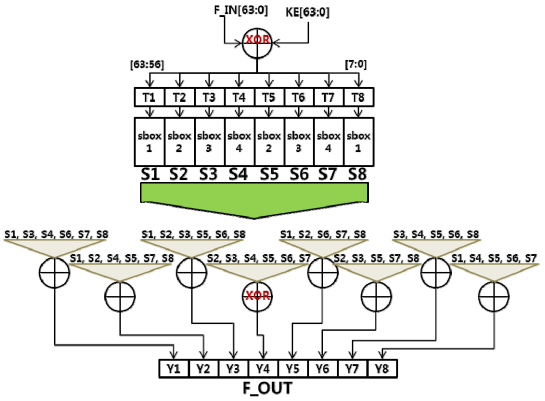

Fig. 2 Block diagram of F function

그림 3은 라운드 연산을 통해 128비트 입력 데이터 M에 대해 최종적인 암호화된 출력 C를 도출하는 흐름을 보여주는 블록도이다.

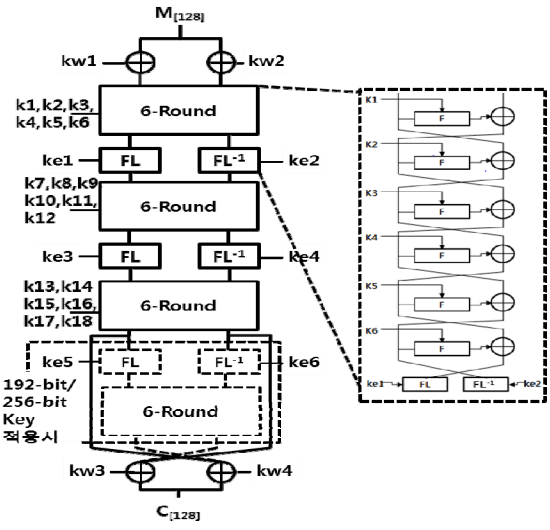

Fig. 3 Block diagram of a round operation block

라운딩을 시작하기 전 128비트 평문 M을 64비트씩 분할하여 미리 kw1과 kw2 화이트닝 키를 적용하여 사전 화이트닝(Prewhitening)을 수행하며, 마지막 6라운드를 제외하고는 매 6라운드 수행 후 FL과 FL^{-1} 함수를 적용한다. 각 6라운드의 수행에 대한 블록도는 그림 3에서 확인할 수 있다. 그리고, 192비트 키 혹은 256비트 키를 적용할 경우에는 추가적으로 FL 관련 함수와 6라운드 랜덤화를 수행한다. 최종적인 암호문을 출력하기 직전에 kw3과 kw4를 적용한 포스트 화이트닝(Postwhitening)을 수행한다.

FL 함수와 FL^{-1}함수는 Feistel 구조의 매 6 라운드 사이에 비규칙성을 더해주기 위해 삽입되었다. 이러한 함수를 사용하는 목적중의 하나는 미래의 알려지지 않은 공격을 방해하기 위함이다[7]. 이 함수는 AND, OR, XOR 및 회전 등의 논리 연산만을 사용하여 구현되기 때문에 하드웨어적으로 설계가 용이하다.

그림 4와 그림 5는 FL 함수와 FL^{-1}함수의 블록도를 보여준다.

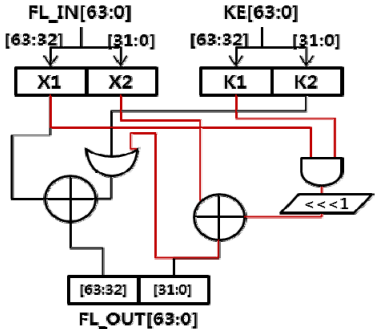

Fig. 4 FL function of Camellia

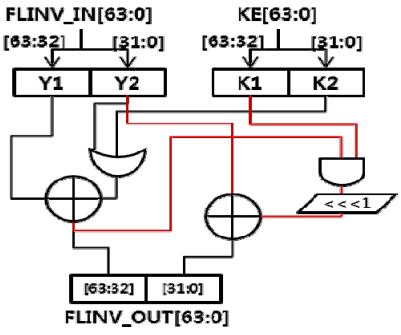

Fig. 5 FL^{-1} function of Camellia

Ⅲ. Camellia 암호 알고리즘 실행 블록 설계

일반적으로 블록 암호 설계에는 키 생성과 라운드 연산을 동시에 수행하기 위해 스케줄링 블록과 라운드 연산 블록을 분리하여 설계한다. 그러나, 본 논문에서는 키 생성을 주관하는 스케줄링 블록과 라운드 연산 블록의 구조가 매우 유사하다는 것과 32비트 외부 인터페이스를 사용하는 특징을 조합한다. 이를 통해 성능에 영향을 미치지 않으면서 라운드 키 생성을 위한 레지스터 설정과 라운드 연산을 수정된 라운드 연산 블록에서 통합하여 수행하는 기법을 제안한다. 아울러 F 함수에서 사용하는 8개의 256x8비트의 ROM을 이중포트(Dual port) ROM을 적용하여 4개의 이중포트 ROM만을 사용하여 키 생성과 라운드 연산을 수행하는 방안을 제안한다. 아울러 본 연구에서 설계한 Camellia 암호 프로세서는 128, 192, 256비트 키 모드를 모두 지원하며, 128비트 메시지에 대한 암호화 및 복호화를 모두 지원한다.

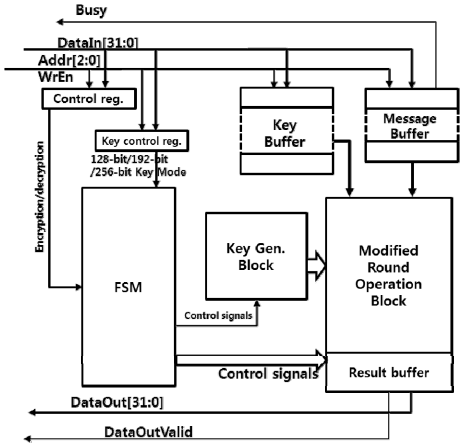

Fig. 6 Architecture of Camellia block encryption/ decryption core

그림 6은 본 논문에서 설계한 Camellia 블록 암호 알고리즘 코어의 아키텍쳐를 보여준다. 메시지 버퍼(Message Buffer)는 32비트로 된 4개의 레지스터로 구성되어 있다. 암호화시에는 총 128비트의 평문을 저장하며, 복호화시에는 총 128비트의 암호문을 저장한다. 메시지 버퍼는 키 값 및 제어 레지스터 등에 대한 설정을 완료한 후에 입력을 수신해야 한다. 4개의 완전한 입력 데이터가 수신되는 즉시 Camellia 암 · 복호화 모듈의 설정 값에 따라 암호화나 복호화를 수행하게 된다.

키 버퍼(Key Buffer)는 설정된 키 사이즈 정보에 따라 128비트, 192비트 및 256비트를 저장하며, 128비트 KL, KR, KA, KB 값을 사용해 키 설정 모드에 따라 서브키(Subkey)를 데이터에 대한 라운드 연산 혹은 키 생성을 위한 라운드 연산의 각 단계와 동기된 값을 제공한다. 일반 제어 레지스터(Control reg.)는 암호화 모드 및 복호화 모드를 설정하는 기능을 담당하고 있으며, 키 제어 레지스터(Key control register)는 키 크기 정보를 설정하는 기능을 담당한다.

본 논문에서 설계된 Camellia 암 · 복호화 코어는 라운드 연산 블록을 키 생성과 메시지 처리에 대해 공유하기 위해 수정된 라운드 연산 블록을 설계하였는데, 수정된 라운드 연산 블록은 그림 7에 제시하였다. 그림 1과 그림 3에서 알 수 있듯이 키 스케줄링 블록과 라운드 연산 블록은 공통적으로 F 함수를 사용하며, 키 스케줄링 블록은 2번째와 4번째 라운드 연산이 종료될 때 128비트 KL 혹은 KR과 XOR 연산을 추가적으로 사용한다. 그리고 메시지에 대한 라운드 연산의 경우에는 매 6라운드 연산 후 FL과 FL^{-1} 함수를 수행한다. 따라서 이를 모두 반영하도록 라운드 연산 블록을 수정하였다.

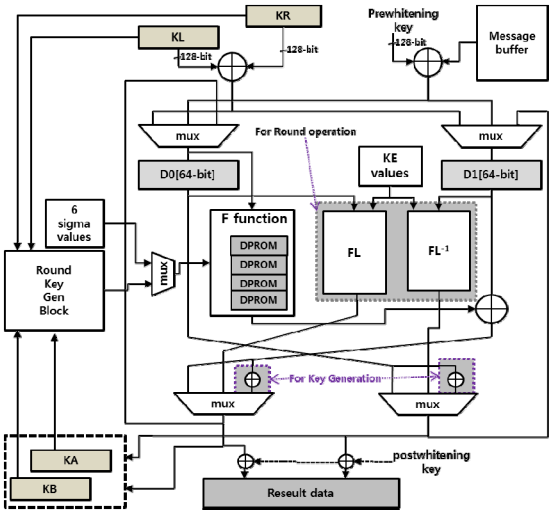

Fig. 7 The modified round operation block

이러한 수정이 가능한 이유는 Camellia 암호 알고리즘은 KA와 KB 레지스터에 대한 설정이 완료되면 더 이상 라운드 연산 블록을 사용할 필요가 없기 때문이다.

라운드 키 생성은 KL, KR, KA 및 KB만을 사용하여 구현된다. KL, KR, KA 값을 사용하는 128비트 키 모드는 KB 값 생성에 4 클럭이 소요되며, 192, 256비트 키 모드는 KA와 KB 값을 생성하는데 6 클럭이 소요된다. 그림 7에서 볼 수 있듯이 라운드 연산 중간 값을 저장하는 각 128비트 D0와 D1 레지스터와 F 함수를 공유하기 때문에 4개의 이중포트 ROM만을 사용하여 메모리의 사용을 최소화하고, 레지스터의 사용을 줄 일 수 있도록 하였다. 설계시 한 가지 주의할 점은 본 설계에서 사용한 이중포트 ROM은 1클럭의 레이튼시(Latency)를 가지므로 데이터 패스상에서 한 클럭 앞선 데이터를 사용해야 한다. 본래는 F 함수의 입력으로 D0를 사용해야 하는데, ROM의 레이튼시를 고려해 D0에 저장되기 직전의 값을 F 함수 입력으로 사용한다. 또한 라운드 키에 대해서도 동일한 방식을 적용하였다.

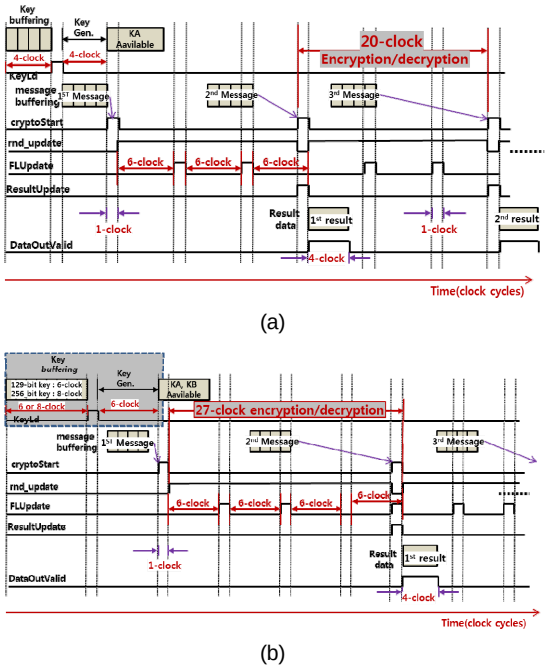

Fig. 8 Timing diagram of Camellia cipher (a) in 128-bit key mode (b) in 192 or 256-bit key mode

그림 8은 본 논문에서 설계한 Camellia 암호 프로세서의 타이밍도를 보여주고 있다. 먼저 128비트의 키를 사용하는 경우의 동작 타이밍을 설명하면 32비트 외부 버스를 통해서 4회에 걸쳐 키 값을 키 버퍼에 저장한다. 128비트의 키가 키 버퍼에 저장되면 즉시 키 내용을 D0와 D1와 전달(KeyLd 신호)하여 수정된 라운드 연산 블록을 사용해 키 생성을 4 클럭에 걸쳐 수행한다. 이 때 메시지 버퍼는 키 버퍼와 독립적으로 동작하기 때문에 외부 버퍼는 프로세서가 키 생성을 수행하고 있는 중에 암호화나 복호화를 위한 메시지를 메시지 버퍼에 저장할 수 있다.

키 생성을 완료하였을 때 메시지 버퍼에 128비트의 메시지가 존재하면 즉시 메시지 버퍼의 내용을 D0와 D1으로 전달(CryptoStart 신호)하여 라운드 연산(rnd_update 신호)을 수행한다. 매 6 라운드 마다 1회의 FL, FL-1 함수 연산(FLUpdate 신호)을 수행한다. FLUupdate 신호가 활성화되면 그림 7의 하단에 있는 mux는 FL이나 FL^{-1} 함수의 결과 값을 선택한다. 그리고 라운드 연산의 마지막 클럭에서는 rnd_update 신호가 활성화되지 않고 새로운 메시지가 메시지 버퍼에 존재하면 즉시 CryptoStart 신호를 활성화하여 다음 클럭 에지에서 D0와 D1에 전달하여 다음 메시지에 대한 연속적인 라운드 연산을 수행할 수 있도록 하였다. 아울러 이 때 결과 값을 저장하기 위한 신호(ResultUpdate) 신호를 활성화하여 포스트 화이트닝(Post whitening) 처리된 결과 값을 결과 레지스터에 저장하고 이후 4클럭 연속으로 결과 값을 외부로 전송한다. 본 논문에서 설계한 192, 256비트 키 모드에 대한 타이밍은 128비트 키 모드와 비교하여 키 버퍼에 저장하는 키 사이즈가 다르며, 192, 256비트 키 모드는 동일하게 6클럭의 키 생성 사이클이 소요되고, 24회의 라운드 수행과 3회의 FL, FL^{-1} 함수연산을 수행해야 한다는 점이 다르다. 이에 대한 내용은 그림 8의 (b)에 나타나 있다.

본 논문에서 설계한 Camellia 암호 프로세서는 연속적으로 입력되는 메시지에 대해 128비트 키 모드에 대해서는 매 20클럭마다, 192, 256비트 키 모드에 대해서는 매 27클럭마다 128비트 결과 값을 도출할 수 있다.

Ⅳ. 설계된 암호 코어의 검증 및 성능분석

설계한 Camellia 암호 알고리즘의 검증을 위해 참고문헌 [6]에서 제시한 테스트 케이스 등을 활용하였다. Verilog HDL을 사용하여 Camellia 코어를 설계하였으며 Xilinx사의 Virtex4 디바이스 상에서 구현하였다. 테

스트 케이스의 입력과 결과 값을 설계된 코어에서 입력에 대한 출력 값이 서로 일치하는지를 비교하여 검증을 완료하였다.

여러 가지 모드에 대한 시뮬레이션 중에서 그림 9는 본 논문에서 설계한 Camellia 코어의 128비트 키를 적용한 암호화 과정을 검증한 파형이다. 제공한 입력 “01234567”, “89abcdef”, “fedcba98”, “76543210"에 대해 최종적인 암호 출력은 ”67673138“, ”54966973“, ”08570656“, ”48eabe43“으로 테스트 케이스에서 제시 값과 동일함을 확인하였다. 복호화 과정은 암호화 과정의 반대로 입력으로 암호문을 제공하고, 동일한 키를 적용하면 원래의 평문을 도출할 수 있다.

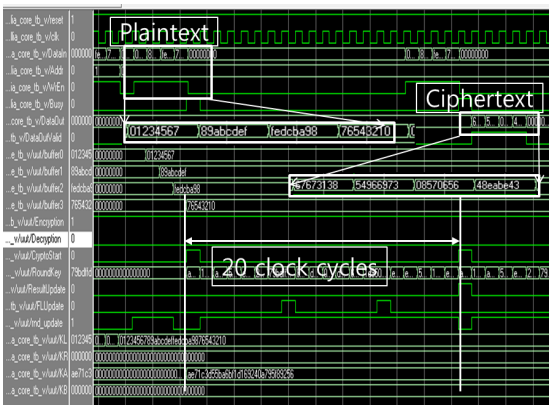

Fig. 9 Simulation of encryption for Camellia cipher (key size : 128-bit key)

본 논문에서 설계한 Camellia 암호 프로세서는 약 184.898MHz에서 동작하는 것을 확인하였다. 최대 처리율의 공식은 [(메시지블록 크기/블록처리 클럭수) x 주파수)] 이고, 128비트의 메시지 블록 단위의 연속적인 처리가 가능하도록 설계하였기 때문에, 128비트 키 모드에서 최대 처리율은 1183Mbps(1.183Gbps)였으며, 192, 256비트 키 모드에서는 최대 876.5Mbps였다. 사용된 슬라이스 수는 3464개이고 4개의 이중포트 ROM을 사용하고, 최대 지연 경로는 라운드 키를 선택하여 F 함수 내에서 XOR연산을 수행하는 경로로 약 5.408ns였다.

Camellia 암호 알고리즘의 구현에 대한 비교 분석은 설계 방법이 상이하여 비교가 쉽지 않았다. 참고적으로 기존의 발표 결과를 살펴보면, 논문[8]는 128비트 키 모드만 지원하면서 4개의 S-box만을 제공하여 1052 슬라이스의 적은 면적으로 VirtexE 상에 구현하여 135Mbps의 성능을 가지며, 논문[9]은 스키메틱(Schematic)을 프로그램으로 번역하는 SPT를 사용하여 암호화만을 FPGA 상에 구현한 것으로 3403개의 슬라이스와 이중포트 ROM을 사용하고 670Mbps의 처리율을 가졌다. 여기서 사용된 이중포트 ROM의 수는 제시되지 않았다. 그리고 논문[10]은 128비트 키 모드의 암호화만을 고속으로 구현하는데 11287 슬라이스를 사용하고 88개의 블록 ROM을 사용하여 33.25Gbps 성능을 보이지만, 너무 많은 디바이스 자원을 사용하는 문제가 있다. 앞의 내용을 종합해 볼 때, 128비트 키 모드의 암호화만을 지원하는 비슷한 하드웨어 자원을 사용하는 논문[9]와 비교해 보면 , 본 논문은 128, 192, 256비트 키 모드에 대한 암호화 및 복호화를 지원하고 훨씬 높은 성능을 보임을 알 수 있었다.

Ⅴ. 결 론

본 논문에서는 128비트 입력에 대해 128비트, 192비트 및 256비트 키를 적용할 수 있는 Camellia 암 · 복호화 코어를 설계하였다. 설계된 Camellia 암호 알고리즘은 Xilinx Vertex4에서 184.898MHz에서 동작하며, 초당 최대 암 · 복화율은 128비트 키 모드에서는 최대 1183Mbps, 192비트 및 256비트 키 모드에서는 최대 876.5Mbps의 데이터 처리율을 가진다. 특히 본 논문은 수정된 라운드 연산 블록을 제안하여 키 생성을 위한 KA, KB 값을 도출하는 것과 기존의 라운드 연산을 공유하는 구조를 제안하였다. 키 생성과 라운드 연산에 필요한 총 16개의 ROM을 단지 4개의 이중포트 ROM만을 사용하여 설계가 가능하도록 하였다. 또한 메시지 버퍼를 제공하여 키 생성을 위한 KA와 KB 값이 도출되면 대기 시간없이 즉시 암호화나 복호화가 가능하도록 구현하였다. 본 논문에서 설계된 Camellia 암호 알고리즘 코어는 보안 관련 분야의 핵심 코어로서 사용이 가능할 것으로 사료된다.

향후 본 연구 결과를 활용하면 AES, SEED, ARIA 등과 같은 알고리즘을 통합한 통합 블록 암호 알고리즘 코어의 설계에도 많은 도움이 될 것으로 판단된다.

ACKNOWLEDGMENTS

This paper was supported by Hanshin University in 2016.

REFERENCES

[1] Sungjoo Ha and Jongho Lee, “Design of fast encryption/ decryption for block cipher ARIA,” *Institute of korean electrical and electronics engineers*, vol. 57 no. 9, pp.1652-1659, Sep. 2008.

[2] Seungil Sonh, “Design of Encryption/Decryption Core for Block Cipher HIGHT,” *KIEE*, vol.16 no. 4, pp.778-784, April 2012.

[3] Seungil Sonh, Byeongyoon Choi and Mingoo Kang, “Technology Trend of Cipher Chips,” *KSII*, vol.1 no.2, pp.1491-1500, Oct. 2001.

[4] Byeongyoon Choi and Jinil Kim, “CPLD Implementation OF SEED Cryptographic Coprocessor,” *KISPS*, vol. 1 no.1-2, pp.177-185, Oct. 2000.

[5] Ashwini M. Deshpande et al., "FPGA Implementation of AES Encryption and Decryption," *International Conf. on control, automation, communications and energy conservation*, pp.1-6, June 2009.

[6] M. Matsui and S. Moriai, "A Description of the Camellia Encryption Algorithm," *Network Working Group Request for Comments: 3713*, Apr. 2004.

[7] Kazumaro Aoki, Tetsuya Ichikawa et al., "Camellia: A 128-bit Block Cipher Suitable for Multiple Platforms," *Proceedings of the 7th Annual International Workshop on Selected Areas in Cryptography*, pp.39-56, 2000.

[8] Huiju Cheng and Howard Heys, "Compact Hardware Implementation of the Block Cipher Camellia with Concurrent Error Detection," *Canadian Conference on Electrical and Computer Engineering*, pp.1129-1132, Apr. 2007.

[9] Masashi Watanabe, Keisuke Iwai, Hidema Tanaka, and Takakazu Kurokwwa, "FPGA implementation of Ciphers using Schematic to Program Translator(SPT),“ *Bulletin of Networking, Computing, Systems, and Software*, vol.4, no. 1, pp.1-8, Jan. 2015.

[10] Daniel Denning, James Irvine and Malachy Devlin, "A High Throughput Camellia Implementation," *Research in Microelectronics and Electronics*, vol. 1, pp137-140, July 2005.

손승일(Seungil Sonh)

1989년 연세대학교 전자공학과(학사)
1991년 연세대학교 대학원 전자공학과(석사)
1998년 연세대학교 대학원 전자공학과(박사)
1998 ~ 2002년 호남대학교 컴퓨터공학과 조교수
2008 ~ 2009년 미국 미시간공과대학 방문교수
2014 ~ 2015년 미국 조지아공과대학 방문교수
2002년 ~ 현재 한신대학교 정보통신학과 교수
※관심분야 : Security, ASIC Design