제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

고속의 몽고메리 모듈라 멱승기의 구현에 관한 연구

김인섭*, 김영철**
*전남대학교 전자공학과
**전남대학교 전자컴퓨터 정보통신공학부
e-mail:iskim@neuron.chonnam.ac.kr

Study on Implementation of a High-Speed Montgomery Modular Exponentiator

In-Seop Kim*, Young-Chul Kim**
*Dept of Electronics Engineering, Chonnam National Univ.
**Dept of Electronics & Computer Engineering and RRC-HECS, Chonnam National Univ.

요 약

정보의 암호화와 인증, 디지털 서명등에 효율적인 공개키 암호 시스템의 주 연산은 모듈라 멱승 연산이며 이는 모듈라 곱셈의 연속적인 반복 수행으로 표현될수 있다. 본 논문에서는 Montgomery 모듈라 곱셈 알고리즘을 사용하여 모듈라 곱셈을 효율적으로 수행하기 위한 모듈라 멱승 연산기를 구현하였으며 Montgomery 모듈라 곱셈시 발생하는 케리 전파 문제를 해결하기 위하여 CPA을 대신하는 CSA를 사용함으로써 멱승 연산시 발생하는 지연시간을 최소화시키는 결과가 얻어짐을 보였다. 본 논문에서는 Montgomery 모듈라 멱승 연산기 구현을 위하여 VHDL 구조적 모델링을 통하여 Synopsys사의 VSS와 Design analyzer를 이용한 논리 합성을 하였고 Mentor Graphics사 Model sim 및 Xilinx사 Design manager의 FPGA 시뮬레이션을 수행하여 성능을 검증 하였다.

1. 서론

최근 몇 년간 전자상거래 등에 필요한 정보 암호화(encryption), 전자서명 및 인증(electronic signature authentication)에 대한 관심과 수요가 급증하고 있다. 공개키 방식 암호 시스템은 이러한 요구를 충족시키는 기본적인 알고리즘을 갖추고 있는데, 이러한 알고리즘은 큰 정수를 기반으로 한 모듈라 연산에 의해 수행되며, 큰 정수 계수(modulus)의 소인수 분해가 매우 어려움에 그 안전성의 근거를 두고 있다. 모듈라 연산은 내부에 곱셈과 나눗셈이 복합되어 있어 계산 구조가 복잡하고 워드 사이즈가 크기 때문에 고속 실시간 처리를 위한 하드웨어 모듈의 구현이 어렵다. 그러나, 근래에 들어서는 VLSI 설계 기술의 발달과 함께 다양한 연산 알고리즘들이 연구되어 고속 모듈 구현에 대한 연구가 활발해지고 있으며, 하드웨어로 구현했을 경우 키의 안정성 면에서도 소프트웨어보다 월등하다는 장점을 가진다.[1][2] Montgomery 모듈라 멱승 연산기 구현에 대한 기존의 방법들을 살펴보면, Jeong과 Burleson은 몽고메리 알고리즘과는 달리 중간 결과값에 대한 모듈로 감소(modulo reduction)과정에서 계수의 뺄셈을 결정할 때 MSB(Most Signigicant Bit)우선 방식을 사용하였다[3].

본 논문에서는 케리 전파 문제를 해결하고 멱승시 발생하는 지연시간을 최소화하고 처리 속도를 높이기 위해 CPA(Carry Propagation Adder)를 대신하는 CSA(Carry Save Adder)를 사용하는 구조를 갖도록 멱승 연산기를 설계하였으며, VHDL(VHSIC Hardware Description Language)을 사용하여 Synopsys사 Design analyzer로 논리 합성을 수행하였으며, Mentor Graphics사 Model sim 및 Xilinx사 Design manager를 통한 FPGA 시뮬레이션 하여 그 성능을 검증하였다.

본 논문은 다음과 같이 구성되었다. 제 2장에서는 모듈라 곱셈과 Montgomery 알고리즘에 대해 기

※ 본 논문은 일부 "한국과학재단 지정 전남대학교 고품질 전기전자부품 및 시스템 연구센터의 연구비 지원"과 "IDEC의 CAD tool 지원" 에 의해 이루어졌음.

술하고, 제 3장에서는 멱승기의 고속 처리를 위한 개선된 구조를 기술하고, 제 4장에서는 VHDL을 이용한 개선된 구조의 하드웨어 설계의 기술 및 시뮬레이션 결과를 기술한다. 마지막 제 5장에서는 결론에 대해 기술하였다.

2. 모듈라 곱셈과 Montgomery 알고리즘

모듈라 곱셈은 두 수의 곱셈과 그 결과에 대한 모듈라 연산으로 이루어진다. 모듈라 곱셈연산을 수행하는 방법은 곱셈연산 과정후 모듈라 연산을 수행하는 방법과 곱셈연산과 모듈라 연산을 동시에 수행하는 방법이 있다. 모듈라 곱셈 알고리즘 중에서 가장 빠르다는 알고리즘은 Montgomery 알고리즘이며 이는 다시 Walter에 의해서 병렬처리에 적합하도록 수정되었다.[4]

공개키 암호 시스템에 사용하는 모듈라 멱승 A^E *mod* N은 AB *mod* N형태의 모듈라 곱셈의 반복으로 이루어지므로 모듈라 곱셈 AB mod N은 두 수 A, B를 곱한 결과를 모듈러스 N으로 나눈 나머지를 취하는 연산이다. 이를 효과적으로 수행하기 위해 Sedlak, Brickell, Barrett, 그리고 Montgomery 등의 알고리즘들이 제안되었으나, 이중 Montgomery 알고리즘이 간단하며 고속이기 때문에 널리 쓰이고 있다. Montgomery 모듈라 곱셈은 AB mod N가 아닌 ABR^{-1} mod N을 수행하는데 내부 연산이 규칙적이고 데이터 흐름이 일정한 구조를 가지고 있다. 여기서 R은 N과 서로소인 N보다 큰 정수이며 A, B 및 N은 모두 *n*비트라고 가정한다. 각 정수를 기저(base) γ로 표현하면 다음과 같다 $A=\sum_{i=0}^{k-1} A[i]\gamma^i$ $B=\sum_{i=0}^{k-1} B[i]\gamma^i$ 및 $M=\sum_{i=0}^{k-1} M[i]\gamma^i$과 같이 k자리로 나타낼 수 있는데 γ=2인 경우에는 k자리는 n비트와 같아진다.

Montgomery 알고리즘은 임의의 정수 χ의 R에 대한 N-잉여류(residues)를 χR mod N으로 정의하고, 이 군(group)내에서 정의된 연산을 사용함으로써 빠른 모듈라 곱셈을 수행할 수 있다. 먼저 N보다 크고 N과 서로소인 R을 선택하는데 이러한 조건을 만족하는 R은 항상 존재하며, Extended Euclidean 알고리즘을 이용하여 쉽게 구할 수 있다. 하지만 n이 N의 비트 수일 때 div R이나 mod R을 간단히 계산할 수 있도록 γ^i으로 하는 것이 일반적이다. 또한 Montgomery 알고리즘은 모듈라 감소(modular reduction)시 N -residue ({iR mod N | 0 ≤N-1}, N이 k비트라고 하면 R = γ^k)을 정의하고 이 집합 내에서 덧셈과 곱셈만으로 빠르게 모듈라 감소 연산을 수행한다.

Montgomery가 제안한 모듈라 곱셈 연산의 알고리즘은 그림 1과 같다.

```
ModMul (A, B, N, R)
step 1. N'= -N^-1 mod R
step 2. P = A · B
step 3. Q = P · N' mod R
step 4. P = (P + Q · N)/R
step 5. Return (P)
```

그림 1. Montgomery 모듈라 곱셈 알고리즘

그림 1의 알고리즘은 모듈라 배수 Q의 모든 자리수를 한번에 계산하므로 많은 자리수를 필요로 하고 비트 연산으로 처리하기가 어렵다.[4] 그래서 Q의 i번째 자리수를 Q[i]를 차례로 계산하여 $Q[i]Nr^i$를 P에 누적시키고, Extended Euclidean 알고리즘에 의하여 미리 계산된 값 n0 = $-N[0]^{-1}$ mod r을 사용하여 Montgomery 알고리즘을 비트 단위로 곱셈연산을 먼저 수행하고 난 다음에 모듈라 감소 연산을 수행하는 방법으로 나타내면 그림 2의 알고리즘과 같다.

```
ModMul '(A, B, N, r)
step 1. P = 0
step 2. for i=0 to k-1 {
step 3. P = P + A[i] · B
        }
step 4. for i=0 to k-1 {
step 5. Q[i] = P[0] · n0 mod r
step 6. P = (P + Q[i] · N) div r
        }
step 7. if (P ≥N) P-N
step 8. Return (P)
```

그림 2. 곱셈과 모듈라 감소가 분리된 알고리즘

이 알고리즘 에서는 두 개의 케리가 발생하게 되는데 곱셈 P = P + A[i] · B를 수행할 때 발생하는 케리와 모듈라 감소 P = (P + Q[i] · N) div r을 수

행할 때 발생하는 케리가 바로 그것이다. 본 논문에서는 기존논문의 CPA를 사용하여 케리 전파 문제가 발생했는데, 본논문에서 이를해결하고자 CSA를 사용하였고 최종단에 CLA(Carry Look-ahead Adder)를 사용해 최종 케리와 합(Sum)을 더하도록 멱승 연산기를 설계하였다.

3. 멱승연산 고속처리를 위한 알고리즘 및 구조

본 논문에서 이용한 Montgomery 모듈라 감소(Modular Reduction) 알고리즘은 N-residue 상에서 $0 \leq T \leq RN$ 인 임의의 정수 T에 대해 $TR^{-1} \bmod N$을 계산하는 것과 같다. Montgomery 모듈라 감소 알고리즘은 T에 대해 N의 일정한 배수를 더하여 하위 n자리가 '0'인수를 R로 나누면 $TR^{-1} \bmod N$가 된다. *Montgomery* 모듈라 감소 알고리즘은 그림 3과 같다.

```
REDC( )
M := (T mod R)N' mod R
t := (T+MN) div R
if t ≥N then t:=t-N
Return t
```

그림 3. Montgomery Modular Reduction 알고리즘

모듈라 곱셈을 하기 위해 Montgomery 곱셈 알고리즘은 A를 f(A)로, B를 f(B)로 매핑 시킬 때 Montgomery 모듈라 감소 알고리즘 REDC( )를 각각 한 번씩 사용하고, F(C)을 C로 매핑 시킬 때 REDC( )를 한 번 사용한다.

Montgomery 곱셈 알고리즘은 단지 몇 개의 모듈라 곱셈이 필요한 경우에는 적합하지 않은 알고리즘이라 할수 있다. 그러나 암호 연산에서와 같이 큰 수가 입력되는 경우에는 큰 수에 대한 나눗셈의 반복 사용을 통한 모듈라 감소 방법과 Montgomery 모듈라 감소 방법을 비교하여 볼 때 전체 연산속도의 향상에 비해 수체계 변환의 연산부담은 거의 무시될 수 있다.[5]

본 논문에서 Montgomery 모듈라 감소 알고리즘을 이용한 개선된 구조는 그림 4와 같다.

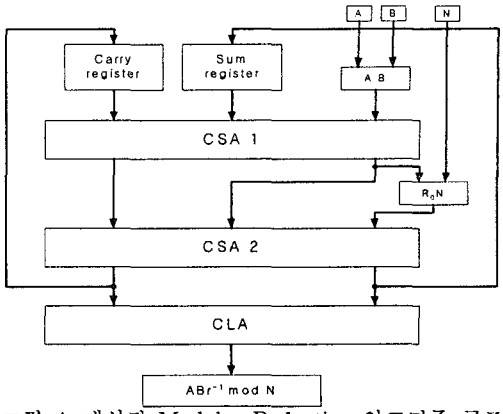

그림 4. 개선된 Modular Reduction 알고리즘 구조

4. Montgomery 모듈라 멱승 연산기 설계

이 장에서는 Montgomery 모듈라 감소 알고리즘을 기본으로 하여 디지털 서명을 위한 공개키 암호 시스템의 모듈라 멱승 연산기 설계를 기술한다. 본 논문은 Synopsys사의 Design analyzer로 논리 합성을 수행하였으며, Mentor Graphics사 Model sim 및 Xilinx사 Design manager를 통한 시뮬레이션 하여 그 성능을 검증 하였다. 알고리즘을 바탕으로 설계한 Montgomery 모듈라 멱승 연산기는 그림 5와 같다.

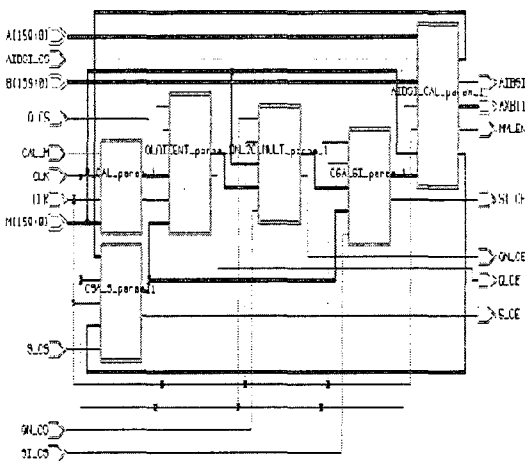

그림 5. Synopsys사 Design analyzer 합성 결과

본 논문에서는 모듈라 멱승을 빠르게 수행하기 위하여 모듈라 곱셈 연산과 모듈라 감소 연산으로 나누고 곱셈 계산 도중에 생성되는 중간 곱에 대하여 감소 연산을 수행하는 방식을 사용하였고, 다음 그

림 6에서는 VHDL에 의한 시뮬레이션 결과이다.

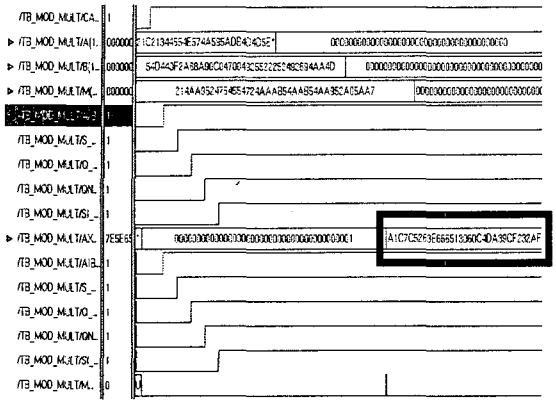

그림 6. VHDL에 의한 시뮬레이션 결과

본 논문에서는 Generic문을 사용하여 공개키 암호 시스템에 사용되어지는 암호 프로세서의 Key size를 임의로 적용해 호환성을 부과 시켰으며, 구현한 Key size는 KCDSA(Korean Certificate-based Digital Signature Algorithm)의 Key size 160 비트를 참조해 구현하였고, 다음 그림 7은 Mentor Graphics사의 Model sim을 이용한 시뮬레이션 결과이다.

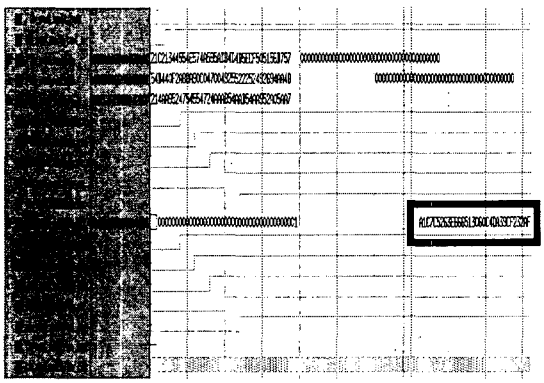

그림 7. Mentor Graphics사의 Model sim 결과

위 그림 6, 그림 7 의 시뮬레이션 결과를 살펴보면 데이터 A와 B, 그리고 모듈러스(N)의 각각 입력이 들어와 모듈라 멱승 연산을 수행하고 결과값은 MM_END 신호가 발생하면 출력되는 형식으로 설계되었고 각각 동일한 결과값이 출력되었음을 알수 있었고, 다음 그림 8은 Xilinx사의 Design manager를 통한 FPGA 검증 결과를 보여주는 그림이다.

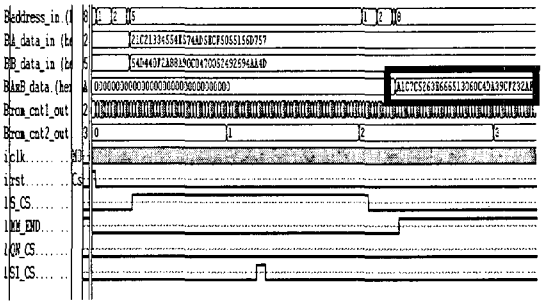

그림 8. Xilinx를 이용한 FPGA 시뮬레이션 결과

5. 결론

본 논문에서는 Montgomery 모듈라 알고리즘을 이용해 모듈라 멱승 연산기를 구현하였으며 개선된 구조를 통하여 케리 전파 문제를 해결하고 멱승시 발생하는 지연시간을 최소화하고 처리 속도를 높이기 위해 CPA(Carry Propagation Adder)를 대신하는 CSA(Carry Save Adder)를 사용하는 구조를 갖도록 멱승 연산기를 설계하였다.

앞으로 수행되어야 할 과제중의 하나는 구현된 Montgomery 모듈라 멱승 연산기를 이용해 공개키 암호 알고리즘에 적용하여 암호 프로세서 구현에 관한 효율성을 증명할 수 있는 연구가 필요할 것이다.

참고문헌

[1] ETRI 정보화기술연구소, "정보보호산업 시장동향 및 전망", ITFIND 주간기술동향 1055권, 2002. 7

[2] IEEE Std 1363-2000. IEEE Standard Specification for Pubic-Key Cryptography. IEEE Computer society, August 29, 2000.

[3] Y.Jeong, W.Burleson, "VLSI Array Algorithms and Architectures for Modular Multiplication," IEEE Tran. On VLSI systems, vol. 5, pp.211-217, June 1997.

[4] 이석용, 김성두, 정용진, "파이프라인 구조의 고속 RSA 암호화 칩 설계" 정보과학회논문 28권6호. 2001. 6

[5] Thomas Blum, Christof Paar, "Montgomery Modular Exponentiation On Reconfigurable Hardware", IEEE Symposium On Computer Arithmetic, April 14-16, 1999, Australia.