LFSR 구조를 이용한 AB^2 곱셈기†

(AB^2 Multiplier based on LFSR Architecture)

전 일 수*, 김 현 성**
(Il-Soo Jeon, Hyun-Sung Kim)

요 약 Kim과 Fenn등은 LFSR 구조를 이용한 두 가지 구조의 효율적인 모듈러 AB 곱셈기를 구현하였다. 그들의 구조는 기약다항식으로 모든 계수가 1인 속성의 AOP를 이용함으로서 기존의 곱셈기들보다 효율적인 구조복잡도를 가졌다. 본 논문에서는 Kim의 곱셈기보다 효율적인 공간 복잡도를 가진 LFSR(Linear Feedback Shift Register) 구조 기반의 모듈러 AB^2 곱셈기와 모듈러 지수승기를 제안한다. 본 논문에서 제안한 구조도 Kim의 구조에서와 같이 기약다항식으로 AOP를 사용한다. 시뮬레이션 결과 본 논문에서 제안한 AB^2 곱셈기가 구조복잡도 면에서 Kim의 구조보다 XOR와 AND 게이트의 개수를 약 50% 정도 줄일 수 있었다. 제안한 구조는 공개키 암호화 시스템을 위한 기본구조로 사용될 수 있을 것이다.

핵심주제어 : LFSR 구조, 공개키 암호화 시스템, AOP 구조, 곱셈기

Abstract Kim and Fenn et al. proposed two modular AB multipliers based on LFSR(Linear Feedback Shift Register) architecture. These multipliers use AOP, which has all coefficients with '1', as an irreducible polynomial. Thereby, they have good hardware complexity compared to the previous architectures. This paper proposes a modular AB^2 multiplier based on LFSR architecture and a modular exponentiation architecture to improve the hardware complexity of the Kim's. Our multiplier also use the AOP as an irreducible polynomial as the Kim architecture. Simulation result shows that our multiplier reduces the hardware complexity about 50% in the perspective of XOR and AND gates compared to the Kim's. The architecture could be used as a basic block to implement public-key cryptosystems.

Key Words : LFSR Architecture, Public-Key Cryptosystem, AOP Architecture, Multiplier

I. 서 론

인터넷의 급속한 확산에 힘입어 전자상거래가 활발하게 이루어지고 있다. 안전한 전자상거래를 위해서는 보안기술이 필요하고, 이를 위한 암호화 시스템 구현의 필요성과 중요성이 크게 부각되고 있다. 지난 30여년간 암호화 시스템 등 여러 분야에서 유한필드에 대한 연구가 이루어졌다. 특히, 유한체 $GF(2^m)$은 2^m개의 원소를 가지고 각각의 원소들은 0과 1의 비트-스트링으로 구성된다. 이러한 속성 때문에 갈로아 필드 연산의 하드웨어 구현에 유한체 $GF(2^m)$이 적당하다[1-3].

공개키 암호화 시스템을 구현하기 위한 다양한 모듈러 연산기가 유한체 $GF(2^m)$상에서 제안되었다[1-8]. 많은 연구에서 공간 및 시간 복잡도 향상을 위한 여러 가지 모듈러 연산기가 제안되었다[5-8]. 1997년 Fenn등은 $GF(2^m)$상에서 LFSR (Linear Feedback Shift Register) 구조를 이용한 두 가지 형태의 모듈러 곱셈기를 설계하였다[7]. Fenn등의 구조는 효율적인 구조 복잡도를 가진 기약 다항식 AOP (All One Polynomial)에 기반한 모듈러 곱셈기이다. 2002년 Kim은 Fenn등의 구조를 향상시키기 위한 다양한 LFSR 구조를 설

† 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문
* 금오공과대학교 전자공학부
** 경일대학교 IT대학 컴퓨터공학부

계하였다[8].

본 논문에서는 Kim이 제안한 구조의 구조 복잡도를 줄이기 위한 효율적인 LFSR AB^2 곱셈기를 제안하고, 이를 위한 응용으로 모듈러 지수승기를 제안한다. 본 논문에 제안한 구조는 Kim 및 Fenn등의 구조와 마찬가지로 기약 다항식 AOP의 속성을 이용한 LFSR 구조의 곱셈기이다. 제안한 구조는 Kim의 구조와 비교하여 추가적인 2개의 클럭사이클이 필요하지만 XOR와 AND 게이트의 수를 1/2 정도 줄일 수 있었다. 본 논문에서 제안한 구조는 제한적인 하드웨어 구조가 요구되는 응용을 위한 기본 구조로 사용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 유한필드 에 대한 기본적인 정의와 기약다항식으로서의 AOP 속성에 대하여 설명하고 지수연산에 있어서 모듈러 AB^2 곱셈의 필요성에 대해서 살펴본다. 3장에서는 Kim이 제안한 구조에 대해 살펴보고, 이 구조를 효율적으로 개선하기 위한 모듈러 AB^2 곱셈 알고리즘을 유도하고 이를 위한 LFSR 구조를 제안한다. 4장에서는 기존의 구조와 제안된 구조의 여러 가지 특성들을 비교분석한다. 끝으로 5장에서 결론을 맺는다.

II. 관련연구

유한필드는 Galois 필드(GF)라고도 불린다. 비록 유한필드가 많은 소수의 지수 차수에 대해서 존재하지만, 암호학에서 주로 사용되는 필드는 소수 q에 대한 소수 유한필드 GF(q)와 양수 m에 대한 이진유한필드 GF(2^m)이다. 유한필드 GF(2^m)은 길이가 m인 2^m개의 가능한 비트 스트링 GF(2^m)={(a_{m-1} a_{m-2} ⋯ a_1 a_0)|a_i∈GF(2), $0 \le i \le m-1$}으로 구성된다[3].

유한필드에서 원소들을 표현하기 위해서는 정규기저 (Normal Basis) 표기법, 이원기저(Dual Basis) 표기법, 다항식기저(Polynomial Basis) 표기법 등의 세 가지 표기법이 있다. 그러나 정규기저 표기법과 이원기저 표기법을 이용한 연산에서는 연산전후에 기저변환 단계를 거쳐야 하는 문제가 있다. 본 논문에서는 다항식기저 표기법으로 필드상의 원소들을 표현한다. 다항식기저 표기법에서의 GF(2^m)의 각 원소는 $a(x)=a_{m-1}x^{m-1}+a_{m-2}x^{m-2}+\cdots+a_1x+a_0$, a_i∈GF(2), $0 \le i \le m-1$ 와 같이 m차수 미만의 다항식으로 표현된다.

GF(2^m)상에서 연산 후 연산 결과를 필드의 원소로 만들기 위해서는 차수 m의 기약 다항식(Irreducible Polynomial)이 필요하고, 이 기약 다항식을 이용한 모듈러 연산이 필요하다. GF(2)의 원소를 계수로 갖는 m차의 기약 다항식을 $f(x)$ 할 때, 다항식의 계수가 모두 "1"인 다항식 $f(x)=x^m+x^{m-1}+x^{m-2}+\cdots+x+1$을 AOP(All One Polynomial)라 한다. 이 방정식의 근을 α라고 두면, AOP는 $\alpha^{m+1}+1=0$, (m+1은 소수) 의 속성을 가진다[6]. 100보다 작은 m에 대해서 m이 2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82 일 때 기약 다항식으로서의 AOP를 만족한다.

즉, 본 논문에서는 AOP의 속성을 이용하여 GF(2^m)보다 하나 확장된 GF(2^{m+1})상에서 곱셈 연산이 수행된다 GF(2^{m+1})상의 한 원소 A는 다음 식과 같이 표현된다.

$$A=A_m\alpha^m+A_{m-1}\alpha^{m-1}+A_{m-2}\alpha^{m-2}+\cdots+A_1\alpha+A_0 \qquad (1)$$

여기서 (A_m=0)으로 표현되고, $A_i=a_i+A_m$, $0 \le i \le m-1$ 이다. 또한, 기저 { 1, α, α^2 ,⋯, α^{m-1}, α^m }은 GF(2^m)상의 표준기저에서 한 차원 확장된 기저이다. 이러한 속성은 AB^2곱셈연산을 수행하는데 있어서 효율적인 모듈러 감소를 제공할 수 있다.

유한필드 상에서 Diffie-Hellman 키 교환 방식, 디지털 서명 알고리즘과 ElGamal 암호화 방식과 같이 잘 알려진 알고리즘을 응용한 타원곡선 (Elliptic Curve) 기반의 공개키 암호화 시스템의 구현에 있어서 GF(p)나 GF(2^m) 상에서 지수 연산이 필요하다. 효율적인 지수 연산을 위해서 MSB (Most Significant Bit) 우선 방식의 알고리즘은 다음과 같다[8].

알고리즘의 Step 3에서 모듈러 AB^2 연산이 필요하다. 즉, 효율적인 모듈러 AB^2 연산은 공개키 암호 시스템의 기본적인 연산인 지수 연산을 위한 중요한 연산이다.

[알고리즘1] MSB-first Exponentiation

Input : A , E , $f(x)$
Output : $C = A^E \bmod f(x)$
Step 1 : if (e_{m-1} == 1) $C = A$ else $C = \alpha^0$

Step 2 : for $i = m - 2$ to 0
Step 3 : if (e_i == 1) $C = AC^2 \bmod f(x)$
else $C = \alpha^0 C^2 \bmod f(x)$

III. LFSR AB^2 곱셈기

본 장에서는 Kim이 제안한 LFSR 모듈러 AB^2 곱셈기를 살펴보고, 이 구조를 효율적으로 개선할 수 있는 모듈러 AB^2 곱셈 알고리즘과 이를 위한 효율적인 공간 복잡도의 LFSR 구조를 제안한다. 또한, 설계된 구조의 응용에 대해서 살펴본다.

3.1 Kim의 LFSR 구조

논문 [8]에서 Kim이 제안한 AOP의 속성을 적용한 AB^2 곱셈기 설계를 위한 GF(2^4)상의

a^3	a^2	a^1	a^0
P_4	P_4	P_4	P_4
A_3B_0	A_2B_0	A_1B_0	A_0B_0
A_2B_3	A_1B_3	A_0B_3	A_4B_3
A_1B_1	A_0B_1	A_4B_1	A_3B_1
A_0B_4	A_4B_4	A_3B_4	A_2B_4
A_4B_2	A_3B_2	A_2B_2	A_1B_2
p_3	p_2	p_1	p_0

그림 1. 모듈러 AB^2곱셈

여기서 $\alpha^5 = 1$이고, $\alpha^6 = \alpha$, $\alpha^7 = \alpha^2$, $\alpha^8 = \alpha^3$이다. 즉, AOP를 이용한 모듈러 제곱 연산은 계수의 재배치에 의해서 수행 될 수 있다. 그림 1의 3번째 행부터 7번째 행까지 각 행에 있는 B_i의 계수는 위의 GF(2^4)상의 모듈러 제곱 연산의 결과의 계수와 일치함을 확인할 수 있다. 그림 1의 두

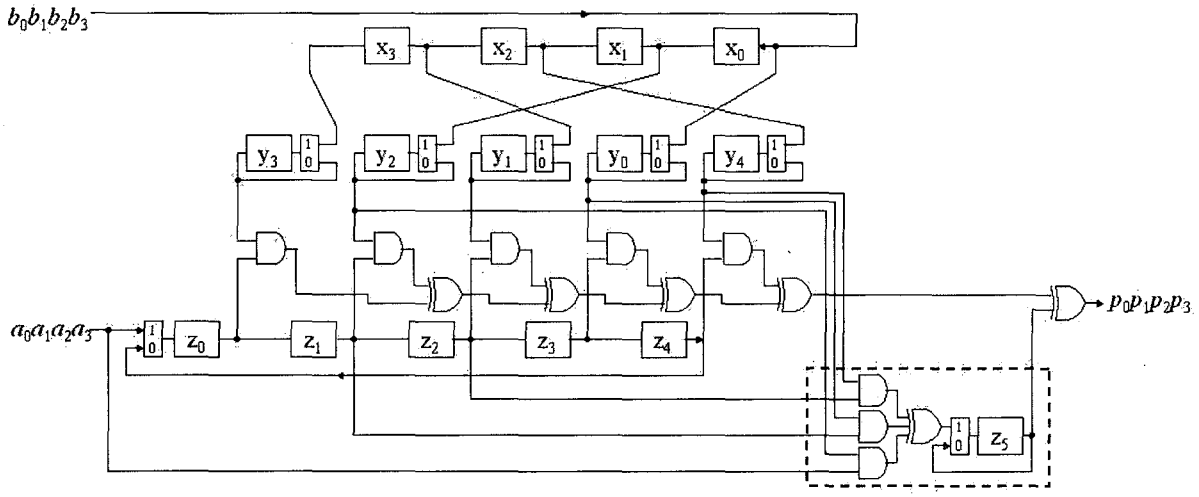

그림 2. Kim의 AB^2 곱셈기

알고리즘은 그림 1과 같다.

AOP의 속성을 이용한 모듈러 제곱연산, $B^2 \bmod \alpha^{m+1}+1$,은 $B^2=(B_m\alpha^m+B_{m-1}\alpha^{m-1}+\cdots+B_1\alpha+B_0)^2 \bmod \alpha^{m+1}+1$ 으로 계산된다 [8]. 이 식을 이용한 GF(2^4)상에서의 모듈러 제곱 연산은 다음과 같다.

$$B^2=(B_4\alpha^4+B_3\alpha^3+B_2\alpha^2+B_1\alpha+B_0)^2 \bmod \alpha^5+1$$
$$=B_2\alpha^4+B_4\alpha^3+B_1\alpha^2+B_3\alpha+B_0$$

번째 행에 강조된 부분은 AOP 연산의 한 차원 확장된 필드의 결과 값을 원래의 필드의 결과 값으로 계산하기 위한 추가적인 연산 부분이다. 그림 2는 그림 1의 알고리즘을 이용한 LFSR 구조에 기반 한 Kim의 AB^2 곱셈기를 보여준다. Kim의 구조는 전체 연산을 위해서 총 $2m$-1 클럭 사이클(clock cycle)이 필요하다. 그림 2의 오른쪽 점선 사각으로 표시된 부분이 그림 1의 2번째 행의 P_4 연산을 위한 부분이다. 여기서 P_4는 다음

과 같이 입력의 마지막 클럭인 m번째 클럭에 계산되어 z_5에 저장된다.

$$P_4=A_4B_0+A_3B_3+A_2B_1+A_1B_4+A_0B_2$$

하지만, 그림 2의 오른쪽 점선 사각 부분의 복잡도는 LFSR 구조의 속성을 이용하여 효율적으로 줄일 수 있다.

3.2 제안된 LFSR 구조

본 절에서는 Kim의 AB^2 곱셈기의 복잡도를 줄이기 위한 효율적인 곱셈기를 제안한다. 이를 위해 먼저 그림 1의 AB^2 곱셈 알고리즘을 분석하고, 이를 기반으로 새로운 LFSR AB^2 곱셈기를 제안한다.

그림 3은 모듈러 AB^2 곱셈 연산을 보여준다. 먼저 그림 3 (a)는 모듈러 연산이 적용되기 전의 AB^2 곱셈과정을 보여준다. 이 곱셈의 결과값은 GF(2^9)상의 결과이고, 이 결과를 원래의 확장된 필드인 GF(2^5)상의 결과 값으로 변환하기 위해서는 AOP의 속성이 적용된 기약다항식인 α^5+1을 이용한 모듈러 감소 연산을 이용한다. 그림 3 (b)는 그림 3 (a)의 연산결과에 모듈러 감소 연산을 적용한 연산 과정을 보여준다.

A =				A_4	A_3	A_2	A_1	A_0	
$\times B^2$ =				B_2	B_4	B_1	B_3	B_0	
				A_4B_0	A_3B_0	A_2B_0	A_1B_0	A_0B_0	
			A_4B_3	A_3B_3	A_2B_3	A_1B_3	A_0B_3		
		A_4B_1	A_3B_1	A_2B_1	A_1B_1	A_0B_1			
	A_4B_4	A_3B_4	A_2B_4	A_1B_4	A_0B_4				
A_4B_2	A_3B_2	A_2B_2	A_1B_2	A_0B_2					
P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	

(a) 모듈러 AB^2 곱셈

α^4	α^3	α^2	α^1	α^0
A_4B_0	A_3B_0	A_2B_0	A_1B_0	A_0B_0
A_3B_3	A_2B_3	A_1B_3	A_0B_3	A_4B_3
A_2B_1	A_1B_1	A_0B_1	A_4B_1	A_3B_1
A_1B_4	A_0B_4	A_4B_4	A_3B_4	A_2B_4
A_0B_2	A_4B_2	A_3B_2	A_2B_2	A_1B_2
P_4	P_3	P_2	P_1	P_0

(b) 모듈러 감소 연산이 적용된 AB^2 곱셈

그림 3. 확장된 기저상의 AB^2 곱셈 알고리즘

Kim은 논문 [8]에서 그림 3 (b)의 확장된 필드인 GF(2^5)상의 결과 값을 원래의 필드인 GF(2^4)상의 결과 값으로 변환하기 위한 알고리즘을 그림 1과 같이 제안하고 이를 위한 LFSR 곱셈기를 설계하였다. 그러나 그림 3 (b)의 속성을 최대한 활용하여 다음 그림 4와 같은 연산을 수행한다면 Kim의 구조보다 더 효율적인 곱셈기를 설계할 수 있다.

또한, AOP의 속성을 이용한 모듈러 연산을 위해서는 2장에서 기술한 것처럼 모듈러 AB^2 곱셈 연산을 위해서는 연산을 위한 필드의 원소 A와 B를 식 (1)을 이용하여 한 차원 확장하여야 한다. 그러나 식 (1)에서 필드의 원소를 한 차원 확장할 때 최고차항은 A_m=0로 초기화되고, 최고차항을 제외한 나머지 항들은 $A_i=a_i+A_m$, $0\le i\le m-1$ 으로 초기화 된다. 즉, 확장된 필드의 원소들은 최고차항의 계수가 항상 '0'임을알 수 있다. 즉, GF(2^4)상의 연산에서 A_4와 B_4는

α^4	α^3	α^2	α^1	α^0
A_4B_0	A_3B_0	A_2B_0	A_1B_0	A_0B_0
A_3B_3	A_2B_3	A_1B_3	A_0B_3	A_4B_3
A_2B_1	A_1B_1	A_0B_1	A_4B_1	A_3B_1
A_1B_4	A_0B_4	A_4B_4	A_3B_4	A_2B_4
A_0B_2	A_4B_2	A_3B_2	A_2B_2	A_1B_2
	P_4	P_4	P_4	P_4
P_4	p_3	p_2	p_1	p_0

그림 4. AB^2 곱셈 알고리즘

α^4	α^3	α^2	α^1	α^0
A_4B_0	A_3B_0	A_2B_0	A_1B_0	A_0B_0
A_3B_3	A_2B_3	A_1B_3	A_0B_3	A_4B_3
A_2B_1	A_1B_1	A_0B_1	A_4B_1	A_3B_1
A_0B_2	A_4B_2	A_3B_2	A_2B_2	A_1B_2
	P_4	P_4	P_4	P_4
	p_3	p_2	p_1	p_0

그림 5. 변환된 AB^2 곱셈 알고리즘

항상 '0'이다.

그러므로 그림 4의 알고리즘은 모든 행에 B_4를 이용한 연산이 필요한 행을 삭제한 구조인 그림 5의 알고리즘으로 변환될 수 있다.

그림 6은 그림 5에 기반 한 LFSR AB^2 곱셈기를 보여준다. 본 논문에서 제안한 곱셈기는 에서 제안한 그림 6의 구조를 사용한다. 또한, product term1과 product term2는 그림 5의 승수와 피승수인 A와 B를 각각 입력으로 한다. 제어 신호인 Control signal을 통하여 AB^2 곱셈기의 첫 값으로 피승수 B를 입력하고, 나머지 값들은 이전 AB^2 곱셈기의 결과 값을 피승수로 입력한

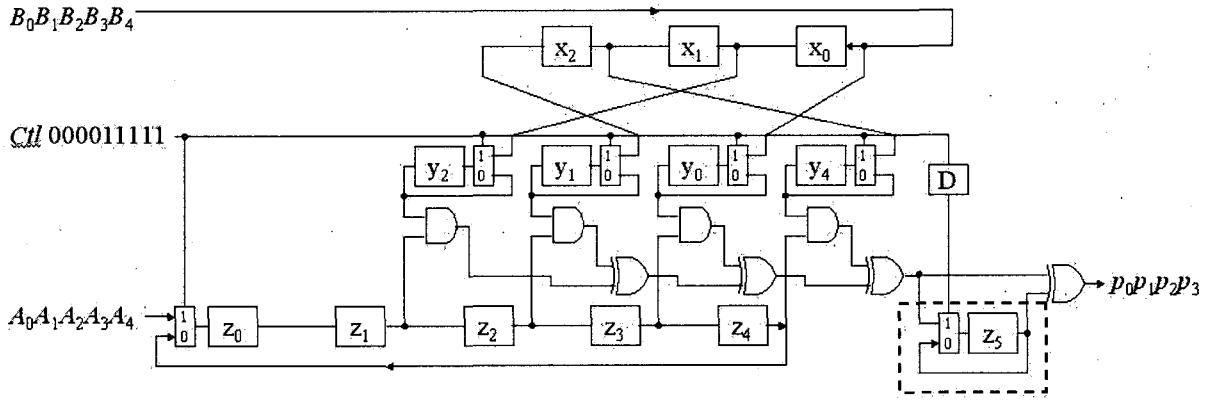

그림 6. 제안한 AB^2 곱셈기

그림 2의 점선 사각 부분의 복잡도를 아주 효율적으로 줄일 수 있다. 또한, 그림 2에서의 x_3과 y_3의 구조와 이와 연관된 추가적인 1개의 Mux, AND, XOR 게이트를 줄일 수 있었다. 그러나 점선 사각 부분의 Mux를 컨트롤하기 위한 하나의 Delay가 추가적으로 요구되고, 전체적인 수행시간이 Kim의 구조보다 2클럭 사이클을 추가적으로 요구된다.

3.3 제안된 구조의 응용

본 논문에서 제안된 AB^2 곱셈기는 모듈러 지수기를 설계하기 위한 기본구조로 활용될 수 있다. 지수기는 하드웨어 복잡도를 최소화하는데 초점을 맞춘 순차 처리방식과 계산시간을 최소화하는데 초점을 맞춘 병렬 처리방식이 있다. 본 논문에서는 하드웨어 복잡도를 최소화하는 순차 처리방식을 채택하여 본 논문에서 제안한 AB^2 곱셈기를 사용하여 지수기를 그림 7과 같이 설계한다.

그림 7은 알고리즘 1을 위한 GF(2^m)상의 지수기를 보여준다. 여기서 AB^2 곱셈기는 본 논문다. 그림 7의 구조는 다양한 입력의 조합을 통하여 나눗셈과 역원기로도 활용이 가능하다.

표 1. 모듈러 AB^2 곱셈기 비교

구조 / 항목	Kim	제안한 곱셈기
기약다항식	AOP	AOP
연산 후 결과	GF(2^m)	GF(2^m)
레지스터	$3m+3$	$3m+2$
AND	$2m$	m
XOR	$2m-1$	m
MUX	$m+3$	$m+2$
Latency	$2m-1$	$2m+1$

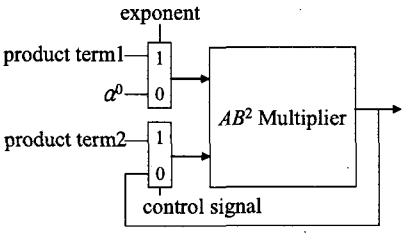

그림 7. 모듈러 지수기

Ⅳ. 분석 및 시뮬레이션

본 장에서는 본 논문에서 제안한 AB^2 곱셈기 구조와 Kim이 제안한 비트순차 LFSR 구조와 여러 가지 특성을 비교 및 분석하고 시뮬레이션 결과를 제시한다.

4.1 분석

본 절에서는 3.1절에서 상세히 살펴본 논문 [8]에서 Kim이 제안한 모듈러 곱셈기와 본 논문에서 제안한 구조의 특성을 비교 및 분석한다. 표 1은 여러 가지 특성에 기반 한 이들 두 구조간의 비교를 보여준다.

표 1에서 보여준 것처럼 본 논문에서 제안한 구조는 동일한 크기 m에 대하여 Kim의 AND와 XOR 게이트의 개수를 약 50% 줄일 수 있었고, 레지스터와 Mux를 각각 1개씩 줄일 수 있었다. 즉, 본 논문에서 제안한 구조는 Kim의 구조와 비교하여 효율적인 구조 복잡도를 갖는다. Kim의 구조에서 XOR 게이트의 수가 2m-2가 아니라 2m-1이 된 이유는 그림 2에서 점선 안의 3 입력 XOR 게이트가 2 개의 2 입력 XOR 게이트로 고려되었기 때문이다.

시간 복잡도(latency) 측면에서는 Kim의 구조보다 항상 추가적인 2개의 클럭 사이클이 필요하다. 그러나 이러한 시간 복잡도의 증가는 m의 크기가 임의적으로 증가하더라도 상수 2의 추가적인 클럭 사이클이 요구된다. 하지만, 그림 2의 Kim의 구조 복잡도는 m의 크기에 의존적으로 복잡도가 증가함을 확인할 수 있다. 따라서 본 논문에서 제안한 모듈러 AB^2 곱셈기가 시간 공간의 곱 복잡도(Time/Area product) 면에서 기존의 Kim의 구조보다 효과적임을 알 수 있다.

4.2 시뮬레이션

제안한 구조의 논리적인 검증을 위하여 먼저 C 언어로 알고리즘의 검증을 수행하였고, Altera사의 MAX+PLUSII를 이용하여 구조 시뮬레이션 하였다. 그림 8은 본 논문에서 제안한 구조의 $GF(2^4)$상에서 승수가 $A=x^3+x$이고 피승수가 $B=x^2+x+1$일 때의 시뮬레이션 결과를 보여준다. 그림 8에서 중앙의 1.1us, 즉, m+2클럭 사이클인 6클럭 사이클부터 AB^2 곱셈의 출력 값인 $P=x^2+x$가 출력된다.

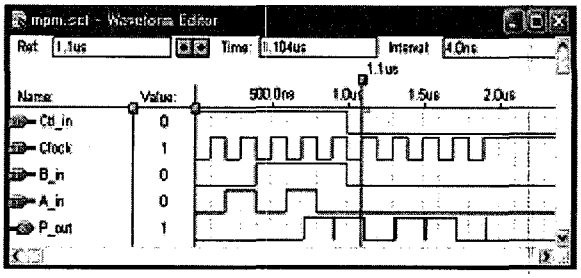

그림 8. 시뮬레이션 결과

그림 8의 시뮬레이션을 위한 전체적인 모듈러 AB^2 곱셈 연산의 처리과정은 다음과 같다.

$B^2=(x^2+x+1)^2 \bmod \alpha^5+1=x^4+x^2+1$

$AB^2=(x^3+x)(x^4+x^2+1) \bmod \alpha^5+1=x^2+x$

Ⅴ. 결 론

본 논문에서는 Kim이 제안한 AB^2 곱셈기의 구조 복잡도를 효율적으로 개선할 수 있는 새로운 LFSR 모듈러 곱셈기를 제안하고 이를 기반으로 하는 모듈러 지수승기를 제안하였다. 제안한 모듈러 곱셈기는 기약 다항식으로서 AOP를 이용한 효율적인 하드웨어 복잡도를 가진 구조이다. 표 1에서 보여 준 것처럼 제안한 AB^2 곱셈기는 기존의 곱셈기보다 효율적인 구조 복잡도를 가짐을 확인할 수 있다. 제안한 AB^2 곱셈기는 공개키 기반의 암호화 프로세서 설계에 효율적으로 이용할 수 있을 것으로 기대된다.

참 고 문 헌

[1] W.Diffie and M.E.Hellman, "New directions in cryptography," *IEEE Trans. on Info. Theory*, Vol. 22, pp. 644-654, Nov. 1976

[2] T.ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. on Info. Theory*,

Vol. 31(4). pp. 469-472, July 1985
[3] R. Lidl, H.Niederreiter, and P.M.Cohn, *Finite Fields* (*Encyclopedia of Mathematics and Its Applications*), Cambridge University Press, 1997.
[4] I.S.Reed and T.K.Truong, "The use of finite fields to compute convolutions," *IEEE Trans. Inform. Theory*, Vol. IT-21, pp. 208-213, Mar. 1975
[5] 김현성, 유기영, "유한필드상에서의 곱셈기 설계 방법," 한국정보과학회 정보보호 연구회지, Vol. 1, No. 1, pp. 12-19, 2001년 4월
[6] 김현성, 유기영, "유한필드상에서의 AOP 곱셈기 설계," 한국정보과학회 정보보호 연구회지, Vol. 3, No. 1, pp. 12-20, 2003년 4월
[7] S.T.J.Fenn, M.G.Parker, M.Benaissa, and D.Taylor, "Bit-serial multiplication in GF(2^m) using irreducible all-one polynomials," *IEE Proc.-Comput. Digit. Tech.*, Vol. 144, No. 6, pp. 391-393, Nov. 1997
[8] H.S.Kim, *Bit-Serial AOP Arithmetic Architecture for Modular Exponentiation*, Ph.D. Thesis, Kyungpook Nat'l Univ., 2002.

전 일 수 (Il-Soo Jeon)

- 1984년 경북대학교 전자공학과(공학사)
- 1988년 경북대학교 대학원 전자공학과(공학석사)
- 1995년 경북대학교 대학원 전자공학과(공학박사)
- 1984년~1985년 삼성전자(주)
- 1989년~2004년 경일대학교 컴퓨터공학과 교수
- 2004년~현재 금오공과대학교 전자공학부 조교수

<관심분야> : 정보보호, 패턴인식

김 현 성 (Hyun-Sung Kim)

- 1996년 경일대학교 컴퓨터공학과 졸업(공학사)
- 1998년 경북대학교 대학원 컴퓨터공학과(공학석사)
- 2002년 경북대학교 대학원 컴퓨터공학과(공학박사)
- 2000년-2002년 (주)디토정보기술 선임연구원
- 2002년-현재 경일대학교 컴퓨터공학부 교수

<관심분야> : 정보보호, 보안 프로토콜, 공개키 암호화 시스템 설계