유한 필드 $GF(2^m)$상에서의 LSB 우선 디지트 시리얼 곱셈기 구현

김 창 훈† · 홍 춘 표†† · 우 종 정†††

요 약

본 논문에서는 유한 필드 $GF(2^m)$상에서 모듈러 곱셈 $A(x)B(x)\ mod\ G(x)$를 수행하는 LSB 우선 디지트 시리얼 시스톨릭 곱셈기를 구현하였다. 구현된 곱셈기는 디지트의 크기를 L로 설정했을 경우 연속적인 입력 데이터에 대해 $\lceil m/L \rceil$ 클럭 사이클 비율로 곱셈의 결과를 출력한다. 본 연구에서 구현된 곱셈기를 기존의 곱셈기와 비교 분석한 결과, 더 간단한 하드웨어 구조를 가지고, 데이터 처리 지연 시간이 감소되었다. 또한 본 연구에서 제안한 구조는 단방향의 신호 흐름 특성을 가지고 있으며, 매우 규칙적이기 때문에 m과 L에 대해 높은 확장성을 가진다.

Implementation of a LSB-First Digit-Serial Multiplier for Finite Fields $GF(2^m)$

Chang Hoon Kim† · Chun Pyo Hong†† · Jong Jung Woo†††

ABSTRACT

In this paper we, implement LSB-first digit-serial systolic multiplier for computing modular multiplication $A(x)B(x)$ mod $G(x)$ in finite fields $GF(2^m)$. If input data come in continuously, the implemented multiplier can produce multiplication results at a rate of one every $\lceil m/L \rceil$ clock cycles, where L is the selected digit size. The analysis results show that the proposed architecture leads to a reduction of computational delay time and it has more simple structure than existing digit-serial systolic multiplier. Furthermore, since the propose architecture has the features of regularity, modularity, and unidirectional data flow, it shows good extension characteristics with respect to m and L.

키워드 : 디지트 시리얼 구조(Digit-Serial Architecture), 유한 필드 곱셈기(Finite Field Multiplier), 암호문(Cryptography), 시스톨릭 배열(Systolic Array), 유한 필드 산술(Finite Field Arithmetic), VLSI

1. 서 론

유한 혹은 갈로이스 필드 ($GF(2^m)$) 상의 연산들은 오류 제어 코딩, 암호학 등 여러 분야에서 중요한 역할을 하고 있다[1, 2]. $GF(2^m)$ 상의 연산 중 덧셈은 비트별 배타적 논리합(XOR) 연산으로 비교적 적은 비용으로 빠른 속도 특성을 가지는 구현이 가능하다. 이와는 달리 곱셈은 유한 필드상의 중요한 연산으로서 복잡할 뿐만 아니라 구현에 따른 비용이 크다. 따라서 $GF(2^m)$상의 곱셈 연산에 대한 효율적인 하드웨어 구현에 대해서는 많은 연구가 이루어져 왔다[3, 4, 6, 7].

※ 본 연구는 한국과학재단 목적기초연구(R01-2000-00402) 지원으로 수행되었음.
† 준 회 원 : 대구대학교 대학원 컴퓨터정보공학과
†† 정 회 원 : 대구대학교 정보통신공학부 교수
††† 종신회원 : 성신여자대학교 컴퓨터정보학부 교수
논문접수 : 2002년 1월 25일, 심사완료 : 2002년 6월 17일

$GF(2^m)$의 원소를 표현할 때 표준 기저 표현법을 사용할 경우 곱셈 알고리즘은 승수의 처리 순서에 따라 LSB(least significant bit) 우선 과 MSB(most significant bit) 우선 방법으로 구분이 되는데, 일반적으로 LSB 우선 곱셈 알고리즘이 MSB 우선 곱셈 알고리즘에 비해 적은 계산 지연 시간을 갖는다[4]. 또한 $GF(2^m)$상의 곱셈기는 비트 패러럴(bit-parallel) 및 비트 시리얼(bit-serial)구조로 구분을 할 수 있는데, 일반적으로 비트 패러럴형은 비트 시리얼형에 비해 데이터 처리율은 높지만, 하드웨어가 복잡해진다는 단점이 있다. 이러한 시간-공간상의 상충관계를 개선하기 위하여 디지트 시리얼 구조의 곱셈기가 제안되었다[5-7].

디지트 시리얼 구조는 데이터를 일정한 크기의 디지트 크기로 나누고, 나누어진 데이터들은 디지트 단위로 처리되고 전송된다. 데이터 크기가 m 비트이고 디지트 크기가 L 비트이면 디지트 개수는 $N = \lceil m/L \rceil$ 이 된다. 비트 패러

럴 구조와 비트 시리얼 구조는 각각 1 클럭 사이클, m 클럭 사이클마다 결과를 출력하고, 디지트 시리얼 구조는 N 클럭 사이클마다 결과를 출력한다. 만약 디지트 크기를 적절히 선택한다면 공간-시간 상충관계를 개선할 수 있다.

본 논문의 저자들은 $GF(2^m)$ 상에서 디지트 시리얼 형태의 곱셈기 구조를 제안한 바 있다[7]. 제안된 곱셈기에 대한 분석 결과 Guo등[6]이 제안한 곱셈기에 비하여 약간의 하드웨어 증가로 계산 수행 지연 시간이 상당히 감소됨을 확인하였다. 또한 처리기 및 이들 사이의 신호 흐름은 단방향이기 때문에 높은 안전성을 가지며, 규칙적인 구조를 가지고 있기 때문에 m과 L에 대해 높은 확장성을 가진다. 만약 입력 데이터가 연속적으로 들어오면 초기 $3N$클럭 사이클 후에 N사이클마다 곱셈 결과 값이 출력된다. 본 연구에서는 이미 저자들이 제안한 곱셈기의[7] FPGA 구현에 관련된 연구 결과를 중심으로 기술하였다.

2. LSB 우선 디지트 시리얼 곱셈기 설계

2.1 LSB 우선 곱셈 알고리즘

$A(x)$와 $B(x)$는 $GF(2^m)$의 원소이고, $G(x)$는 차수 m인 원시 기약 다항식이다. 그리고 $P(x)$는 $A(x)B(x) \bmod G(x)$의 결과이다. 이때 다항식 $A(x), B(x), G(x)$ 및 $P(x)$는 다음과 같이 표현되고, 각각의 다항식들은 0과 1을 계수로 가진다.

$$\begin{aligned} A(x) &= a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1 x + a_0 \\ B(x) &= b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \cdots + b_1 x + b_0 \\ G(x) &= x_m + g^{m-1}x^{m-1} + g_{m-2}x^{m-2} + \cdots + g_1 x + g_0 \\ P(x) &= A(x)B(x) \bmod G(x) \end{aligned} \quad (1)$$

식 (1)에서 곱셈은 다음의 방법 중 하나로 계산된다.

$$\begin{aligned} P(x) &= A(x)B(x) \bmod G(x) \\ &= \{\cdots[A(x)b_{m-1}x \bmod G(x) + A(x)b_{m-2}]x \\ &\quad \bmod G(x) + \cdots + A(x)b_1\}x \\ &\quad \bmod G(x) + A(x)b_0 \end{aligned} \quad (2)$$

또는

$$\begin{aligned} P(x) &= A(x)B(x) \bmod G(x) \\ &= b_0 A(x) + b_1[A(x)x \bmod G(x)] \\ &\quad + b_2[A(x)x^2 \bmod G(x)] + \cdots \\ &\quad + b_{m-1}[A(x)x^{m-1} \bmod G(x)] \end{aligned} \quad (3)$$

어레이 형태의 곱셈기를 구현할 때, 곱수의 비트가 처리되는 순서에 따라 두 가지 방법이 사용된다. MSB 우선 방법은 식 (2)와 같이 곱수 B(x)의 MSB부터 곱셈을 시작하고, LSB 우선 방법은 식 (3)과 같이 곱수 B(x)의 LSB부터 곱셈을 시작한다. 다음은 MSB 우선 곱셈 알고리즘과 LSB 우선 곱셈 알고리즘이다[4].

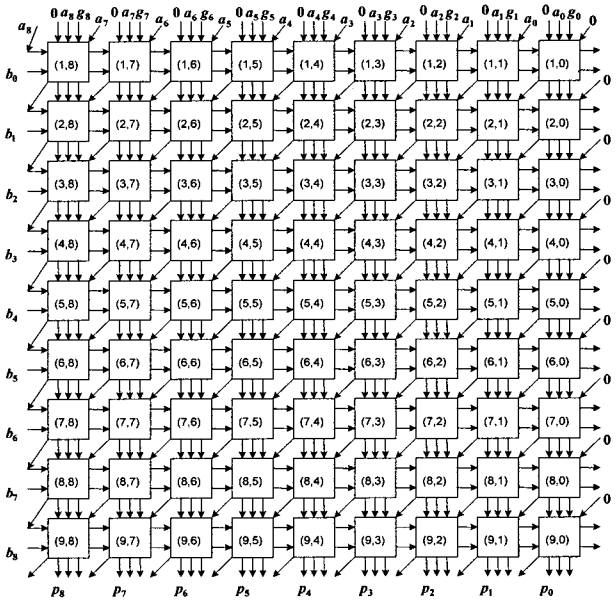
Input : $A(x)$, $B(x)$, $G(x)$
Output : $P(x) = A(x)B(x) \bmod G(x)$
1. $p_k^{(0)} = 0$, for $0 \le k \le m-1$
2. $p_{-1}^{(i)} = 0$, for $1 \le i \le m$
3. for $i = 1$ to m do
4. for $k = m-1$ to 0 do
5. $p_k^{(i)} = p_{m-1}^{(i-1)} g_k + b_{m-i} a_k + p_{k-1}^{(i-1)}$
6. end
7. end
8. $P(x) = p^m(x)$

(알고리즘 2.1) MSB 우선 곱셈 알고리즘

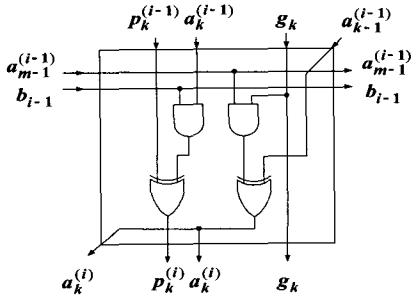
Input : $A(x)$, $B(x)$, $G(x)$
Output : $P(x) = A(x)B(x)\ mod\ G(x)$
1. $a_k^{(0)} = a_k$, for $0 \le k \le m-1$
2. $a_{-1}^{(i)} = 0$, for $1 \le i \le m$
3. $p_k^{(0)} = 0$, for $0 \le k \le m-1$
4. for $i = 1$ to m do
5. for $k = m-1$ down to 0 do
6. $a_k^{(i)} = a_{k-1}^{(i-1)} + a_{m-i}^{(i-1)} g_k$
7. $p_k^{(i)} = a_k^{(i-1)} b_{i-1} + p_k^{(i-1)}$
8. end
9. end
10. $P(x) = p^m(x)$

(알고리즘 2.2) LSB 우선 곱셈 알고리즘

위의 두 (알고리즘 2.1)과 (알고리즘 2.2) $a_k^{(i)}$, $p_k^{(i)}$는 각각 $A^{(i)}(x)$와 $P^{(i)}(x)$의 k번째 계수를 나타내고, a_i와 b_i는 $A(x)$와 $B(x)$의 i번째 계수를 나타내고, g_k는 $G(x)$의 k번째 계수를 나타낸다[4]. LSB 우선 방법으로 구현된 곱셈기는 MSB 우선 방법으로 구현된 곱셈기와 하드웨어 복잡도는 같으나, 계산 지연 시간은 더 적다. 위에 기술된 알



(그림 1) $GF(2^9)$상의 데이터 의존 그래프

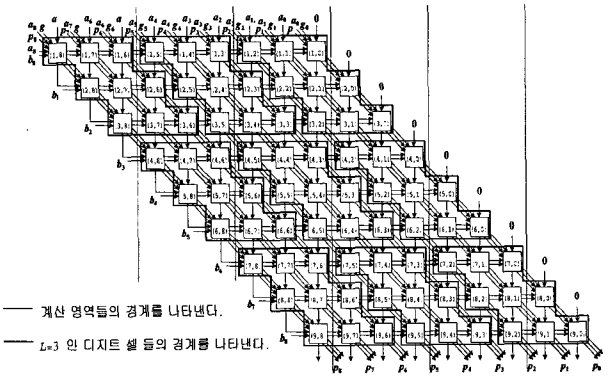


(그림 2) (그림 1)의 (i, k) 계산점 회로도

고리즘을 바탕으로 $GF(2^m)$상의 LSB 우선 곱셈 알고리즘의 데이터 의존 그래프 및 (i, k) 계산점에서의 회로도를 구하면 (그림 1) 및 (그림 2)와 같이 주어진다[7]. 이 경우 $m=9$이고, 자료의존 그래프는 $m \times m$ 개로 구성되어 있다고 가정한다. (그림 1)에 기술된 것처럼 결과 값 $P(x)$는 m번 반복 후 자료 의존 그래프의 가장 아래 열에서 구할 수 있음을 알 수 있다.

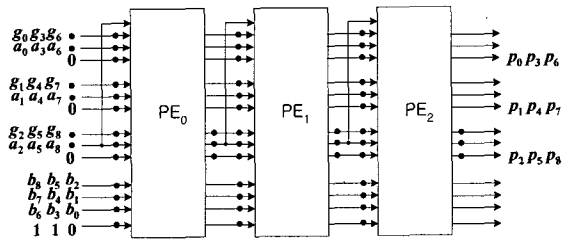
2.2 자료 의존 그래프의 변형

디지트 크기가 L인 디지트 시리얼 시스톨릭 어레이를 만들기 위해 (그림 1)을 $L \times L$로 묶으면, $L-1$개의 일시적인 결과는 인접한 오른쪽 디지트 셀에서 계산이 되어진다. 따라서 오른쪽으로 투영시킬 경우 일차원 신호 흐름 그래프(Signal Flow Graph : SFG)를 얻을 수 없다. 이러한 문제를 해결하기 위해 (그림 1)의 데이터 의존 그래프를 인덱스 변환시키면 (그림 3)의 데이터 의존 그래프를 얻을 수 있다[7].

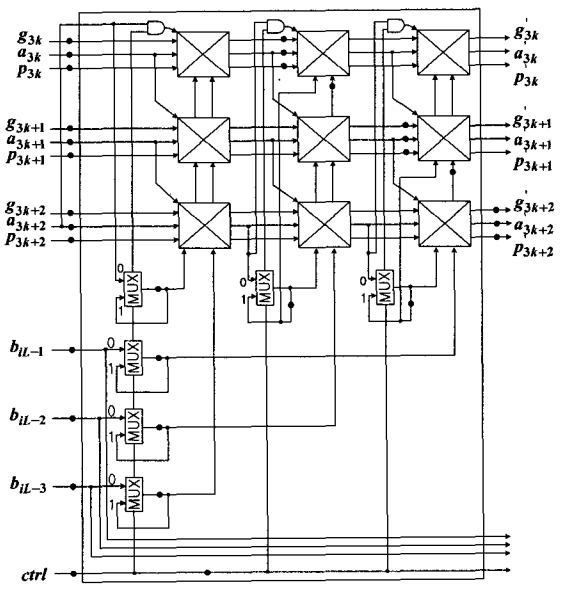


(그림 3) (그림 1)의 변형된 데이터 의존 그래프

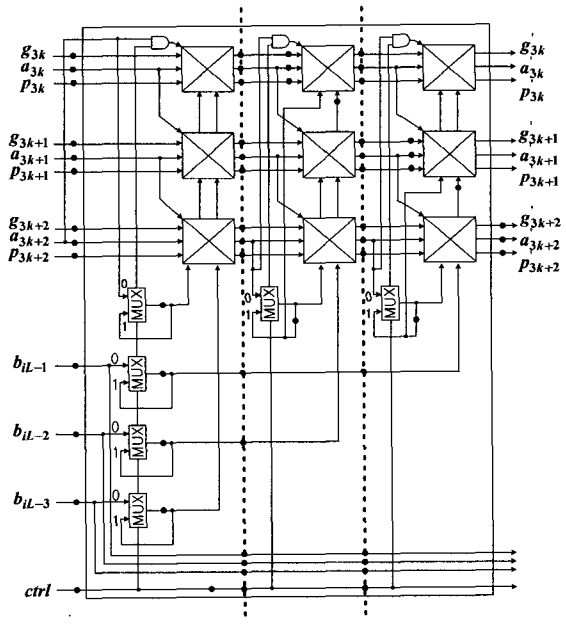
(그림 3)의 데이터 의존 그래프를 동쪽 방향으로 투영시킨 다음, 컷 셋 시스톨릭화 기법(cut-set systolization technique)[8]을 적용하면 (그림 4)와 같은 N개의 Processing Element(PE)로 구성된 디지트-시리얼 시스톨릭 곱셈기의 구조를 얻을 수 있다[7]. (그림 5)는 (그림 4)의 곱셈기를 구성하는 각각의 PE 구조를 나타내며, '•'은 1-비트 1-사이클 지연소자 이고, '⊠'는 (그림 2)의 (i, k) 계산점 회로를 나타낸다.



(그림 4) $L=3$일 때 $GF(2^9)$의 디지트 시리얼 곱셈기 구조



(그림 5) (그림 4)의 PE 구조 [7]



(그림 6) (그림 5)의 변형된 PE 구조

cut theorem[8]을 적용하여 (그림 5)의 각 PE는 파이프라인 될 수 있다. 각 PE의 변형된 구조는 (그림 6)과 같다. 점선을 따라 데이터 버스에 1비트 래치를 추가했다. 각 PE는 S+1 상태로 파이프라인 되고, L/(S+1)는 정수형이다. $4SL+\frac{SL}{2}+S$개의 1비트 래치를 추가함으로써, 최대 계산 전달 지연시간을 아래만큼 줄일 수 있다.

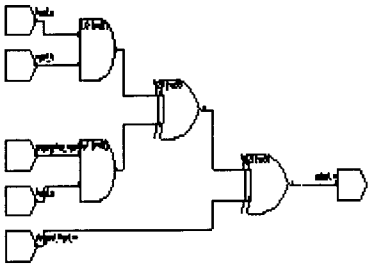
$$T'_{max}=\frac{L}{S+1}(T_{AND2}+T_{XOR2}+T_{MUX2}) \quad (4)$$

(그림 6)의 PE는 $L=3$, $S=2$ 이다. 최대 전달 지연시간은 $T_{max}=T_{AND2}+T_{XOR2}+T_{MUX2}$ 만큼 감소했고, 29개의 1비트 래치가 추가되었다. 식 (4)에 의해 디지트 크기와 파이프라인 상태를 결정할 수 있다.

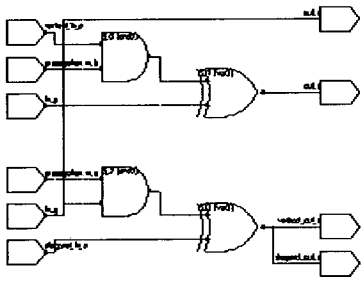
3. 곱셈기의 구현

본 절에서는 2절에서 설계된 곱셈기의 구현에 대하여 기술한다. 참고로 성능을 비교하기 위하여 Guo 등[6]이 제안한 곱셈기와 본 연구에서 제안한 곱셈기를 동일한 환경에서 구현 후 특성을 비교, 분석하였다.

곱셈기를 구현하기 위해 설계된 곱셈기를 VHDL로 기술하였으며, Synopsys사의 FPGA-Express(Version 2000, 11-FE3.5)를 이용하여 곱셈기 회로를 합성하였다. 또한 곱셈기의 기능 검증을 위해 Mentographics사의 VHDL-ChipSim을 이용하여 시뮬레이션을 수행하였으며, 대상 FPGA 디바이스로는 ALTERA사의 FLEX10K군인 EPF10K100ARC240-3을 사용했다.

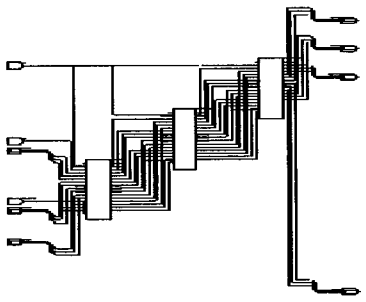


a. Guo 곱셈기

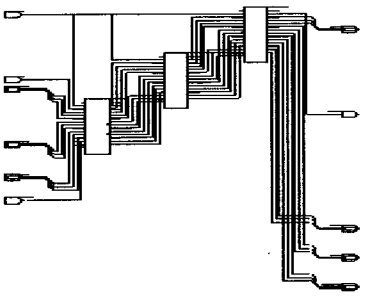


b. 본 연구에서 제안한 곱셈기

(그림 7) 각 곱셈기의 계산점 회로도

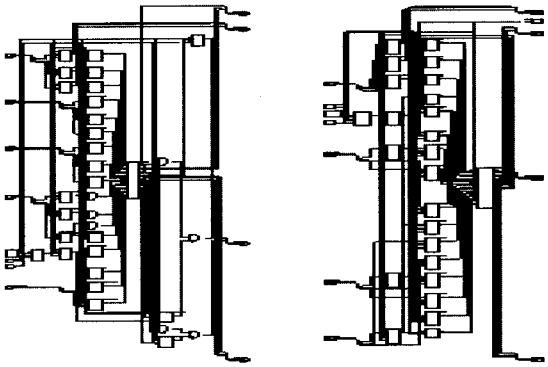


a. Guo 곱셈기



b. 본 연구에서 제안한 곱셈기

(그림 8) $m=9$, $L=3$인 디지트 시리얼 시스톨릭 곱셈기 회로 합성 결과



a. Guo 곱셈기 b. 본 연구에서 제안한 곱셈기

(그림 9) (그림 8)에 있는 각각의 PE 회로도

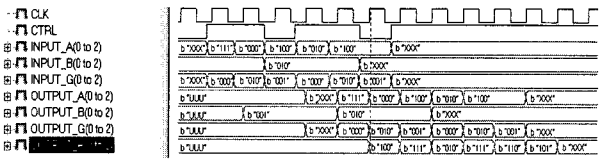
(그림 7)은 각 곱셈기의 계산점 회로의 합성 결과를 나타내며, (그림 8)은 $m=9$ 이고 $L=3$ 인 디지트 시리얼 시스톨릭 곱셈기의 회로 합성 결과를 나타내며, (그림 9)는 (그림 8)에 대한 각각의 PE 구조를 나타낸다.

합성된 회로의 기능 검증을 위하여, FPGA-Express로부터 Netlist 파일을 추출한 후, VHDL-ChipSim을 이용하여 시뮬레이션을 수행하였다. 시뮬레이션 과정에서 입력된 테스트 데이터 집합들은 아래와 같다.

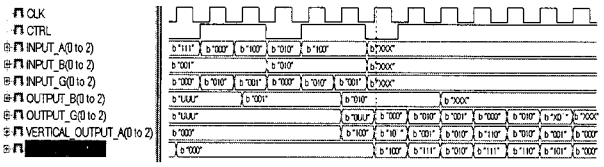
$$A(x)=(x^8+x^7+x^6+x^2, x^7+x^5+x^2)$$
$$B(x)=(x^6+x^3+1, x^7+x^4+x)$$
$$G(x)=(x^9+x^4+1, x^9+x^4+1)$$

시뮬레이션 결과는 각각 (그림 10)과 (그림 11)에 기술되

어 있다. 여기서, (그림 10)은 Guo 등[6] 이 제안한 MSB 우선 방식에 대한 시뮬레이션 결과이고, (그림 11)은 본 연구에서 제안한 LSB 우선 방식에 대한 시뮬레이션 결과이다. (그림 10)과 (그림 11)에 기술된 바와 같이 초기 9($3N$) 클럭 사이클 후에 첫 번째 곱셈 연산 결과가 나타나며, 두 번째 결과는 3(N)클럭 사이클 후에 나타난다. 정확한 곱셈 결과 값은 $P(x)=(x^8+x^5+x^4+x^3+x,\ x^8+x^7+x^6+x^5+x^4+x^2+1)$ 로 주어지며, 시뮬레이션의 결과 동일한 식을 얻었다. 또한 본 연구에서는 $m=256$ 일때, L 값을 변화시키면서(2~32) 동일한 시험을 수행했으며 시험결과 모두 정확한 결과를 얻었다



(그림 10) Guo 곱셈기에 대한 시뮬레이션 결과



(그림 11) 본 연구에서 제안된 곱셈기에 대한 시뮬레이션 결과

4. 성능 분석

본 연구에서 구현된 디지트 시리얼 시스톨릭 곱셈기와 Guo 등[6]이 제안한 디지트-시리얼 시스톨릭 곱셈기를 비교하였다. <표 1>은 FLEX10K100ARC240-3 디바이스에서 $m=9$, $L=3$ 일때 FPGA 구현을 위한 특성을 비교한 결과이다. <표 2>는 두 개 곱셈기의 하드웨어 복잡도 및 데이터 지연시간 측면에서 일반적인 특성을 비교한 결과이며, <표 1> 과 <표 2>로부터 본 연구에서 구현된 곱셈기는 Guo 등[6] 이 제안한 곱셈기에 비해 각 PE당 한 개의 플립플롭(전체적으로는 N개)이 증가하지만 처리 시간 측면에서는 상당히 개선됨을 알 수 있다.

<표 1> m = 9, L = 3일때 FPGA 구현을 위한 특성 비교

구 분	PE				전 체		
	시 간		면 적		시 간	면 적	
	클럭 (MHz)	지연시간 (ns)	D-ff	LUT	지연시간 (ns)	D-ff	LUT
구현된 곱셈기	27.25	36.70	31	23	330.3	93	69
Guo의 곱셈기	18.98	52.70	30	23	474.3	90	69

<표 2> 디지트 시리얼 시스톨릭 곱셈기의 특성 비교

구 분	Guo 곱셈기	구현된 곱셈기
처리기 개수	처리기 : N XOR 2 : L-1 AND 2 : L-1	처리기 : N
처리기 복잡도	AND 2 : $2L^2+L$ XOR 2 : L-1 XOR 3 : L^2-$2L$+2 XOR 4 : L-1 1비트 래치 : $10L+5SL$ MUX : $2L$	AND 2 : $2L^2+L$ XOR 2 : $2L^2$ 1비트 래치 : $(10L+1)+4SL+(SL/2)+S$ MUX : $2L$
지연 시간 (cycles)	$3N$	$3N$
최 대 처리기 지연 시간	With pipelining inside PE	
	$(L)(T_{AND2}+T_{XOR3}+T_{MUX2})/(S+1)$	$(L)(T_{AND2}+T_{XOR2}+T_{MUX2})/(S+1)$
	Without pipelining inside PE	
	$T_{AND2}+T_{XOR4}+(L-1)$ $(T_{AND2}+T_{XOR3}+T_{MUX2})$	$T_{AND2}+T_{XOR2}(L-1)$ $(T_{AND2}+T_{XOR2}+T_{MUX2})$

$N=m/L$
AND 2 : 2-입력 AND gate
XOR 2 : 2-입력 XOR gate
XOR 3 : 2(2-입력 XOR) gates
XOR 4 : 3(2-입력 XOR) gates
MUX 2 : 2-to-1 multiplexer
T_{AND2} : AND 2 전달지연시간
T_{XORi} : XOR i 전달지연시간
T_{MUX2} : MUX 2 전달지연시간

5. 결 론

본 논문에서는 $GF(2^m)$ 상에서 LSB 우선 방식의 곱셈기를 구현하였다. 구현된 곱셈기를 Guo 등[6]이 제안한 MSB 우선 방식의 디지트 시리얼 곱셈기와 비교 분석한 결과, 하드웨어 복잡도는 약간 증가하였지만 계산 지연 시간을 많이 줄일 수 있었다. 또한 제안한 곱셈기는 단 방향의 신호 흐름을 가지기 때문에 안정적이며, 매우 규칙적인 구조를 가지고 있기 때문에 m과 L에 대해 높은 확장성을 가진다.

참 고 문 헌

[1] R. E. Blahut, "Theory and Practice of Error Control Codes," MA : Addison-Wesley, 1983
[2] B. Schneier, "Applied Cryptography," Wiley, 1996
[3] C. L. Wang and J. L. Lin, "Systolic Array Implementation of Multipliers for Finite Field $GF(2^m)$," IEEE Trans. on Circuits and Syst, Vol.38, No.7, pp.796-800, July, 1991.
[4] S. K. Jain, L. Song, and K. K. Parhi, "Efficient Semi-Systolic Architectures for Finite Field Arithmetic," IEEE Trans. on VLSI System, Vol.6, No.1, pp.101-113, March, 1998.
[5] R. Hartley and P. F. Corbett, "Digit-Serial Processing Techniques," IEEE Trans. CAS-37, Vol.37, No.6, pp.707-719, June, 1990.

[6] J. H. Guo and C. L. Wang, "Digit-Serial Systolic Multiplier for Finite Field $GF(2^m)$," IEE Proc.-Comput. Digit. Tech., Vol.145, No.2, pp.143-148, March, 1998.

[7] C. H. Kim, S. D. Han and C. P. Hong, "An Efficient Digit-Serial Systolic Multiplier for Finite Fields $GF(2^m)$," To be appeared on Proc. on 14*th* Annual IEEE International ASIC/SOC Conference, pp.12-15, Sept, 2001.

[8] N. Weste, and K. Eshraghian, "Principles of CMOS VLSI design : A System Perspective," Addison Wesley, Reading, MA, 1985.

[9] S. Y. Kung, "VLSI Array Processors," Englewood Cliffs, NJ : Prentice Hall, 1988.

[10] W. F. Lee, "VHDL Coding and Logic Synthesis with Synopsis," Academic Press, 2000.

김 창 훈

e-mail : chkim@dsp.taegu.ac.kr

2000년 대구대학교 컴퓨터정보공학부 졸업(학사)

2000년~현재 대구대학교 컴퓨터정보공학과 석사과정

관심분야 : 암호 시스템, 내장형 시스템, 재구성형 컴퓨팅

홍 춘 표

e-mail : cphong@daegu.ac.kr

1978년 경북대학교 전자공학과 졸업(학사)

1978년~1985년 국방과학연구소 연구원

1985년~1986년 Georgia Institute of Technology, School of Electrical and Computer Engineering 석사

1986년~1991년 Georgia Institute of Technology, School of Electrical and Computer Engineering 박사

1992년~현재 대구대학교 정보통신공학부 교수

관심분야 : DSP 하드웨어 및 소프트웨어, 컴퓨터 구조, VLSI 신호처리, 내장형 시스템

우 종 정

e-mail : jwoo@cs.sungshin.ac.kr

1982년 경북대학교 전자공학과 학사

1982년~1988년 산업연구원 책임연구원.

1988년~1993년 Univ. of Texas at Austin 전기 컴퓨터 공학과 석사 및 박사

1998년~1999년 Univ. of Texas at Austin 전기 컴퓨터 공학과 객원교수

1993년~현재 성신여자대학교 교수

관심분야 : 컴퓨터구조, 임베디드시스템, 웹캐싱, CAI