유한 필드 $GF(2^m)$상에서의 MSB 우선 디지트 시리얼 곱셈기 설계

정회원 김 창 훈*, 한 상 덕*, 홍 춘 표*

Design of MSB-First Digit-Serial Multiplier for Finite Fields $GF(2^m)$

Chang Hoon Kim*, Sang Duk Han*, Chun Pyo Hong* *Regular Members*

요 약

본 논문에서는 유한 필드 $GF(2^m)$상에서 모듈러 곱셈 $A(x)B(x)$ *mod* $G(x)$를 수행하는 MSB 우선 디지트 시리얼 곱셈기를 설계하였다. 이를 위하여 $GF(2^m)$상에서 MSB 우선 곱셈 알고리즘으로부터 자료 의존 그래프를 구하고, 이를 이용하여 효율적인 디지트 시리얼 시스톨릭 곱셈기를 설계한다. 설계된 곱셈기에 대한 VHDL 코드를 구하고 시뮬레이션을 거친 후 FPGA 로 구현한다. 구현된 곱셈기는 디지트의 크기를 L 로 설정했을 경우 연속적인 입력 데이터에 대해 $\lceil m/L \rceil$ 클럭 사이클 비율로 곱셈의 결과를 출력한다. 본 연구에서 구현된 곱셈기는 기존의 곱셈기와 비교 분석한 결과 시간 및 공간 복잡도가 감소되었으며, 간단한 구조로서 데이터 처리 지연시간을 줄일 수 있다. 또한 본 연구에서 제안한 구조는 단 방향의 신호 흐름 특성을 가지고 있으며, 매우 규칙적이기 때문에 m 과 L 에 대해 높은 확장성을 가진다.

ABSTRACT

This paper presents a MSB-first digit-serial systolic array for computing modular multiplication of $A(x)B(x)$ *mod* $G(x)$ in finite fields $GF(2^m)$. From the MSB-first multiplication algorithm in $GF(2^m)$, we obtain a new data dependence graph and design an efficient digit-serial systolic multiplier. For circuit synthesis, we obtain VHDL code for multiplier. 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 much more simple structure than existing digit-serial systolic multiplier. Furthermore, since the propose architecture has the features of unidirectional data flow and regularity, it shows good extension characteristics with respect to m and L.

Ⅰ. 서 론

유한 혹은 갈로이스 필드($GF(2^m)$)상의 연산들은 오류 제어 코딩, 암호학 등 여러 분야에서 중요한 역할을 한다[1][2]. $GF(2^m)$상의 연산 중 덧셈은 비트별 배타적 논리합(XOR) 연산으로 계산이 가능하기 때문에 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].

$GF(2^m)$의 원소를 표현할 때 표준 기저 (standard base) 표기법을 사용할 경우 곱셈 알고리즘은 승수의 처리 순서에 따라 LSB (least significant bit) 우선 과 MSB (most significant bit) 우선 방법으로 구분된다[4]. 또한 $GF(2^m)$상의 곱셈기는 비트 패러럴 (bit-parallel) 및 비트 시리얼 (bit-serial)구조로 구분

* 대구대학교 컴퓨터정보공학과 (chkim@dsp.taegu.ac.kr)
논문번호 : 010382-1208, 접수일자 : 2001년 12월 8일
※ 본 연구는 한국과학재단 목적기초연구(R01-2000-00402) 지원으로 수행되었음

을 할 수 있는데, 일반적으로 비트 패러럴형은 비트 시리얼형에 비해 데이터 처리율은 높지만, 하드웨어가 복잡해진다는 단점이 있다. 이러한 시간-공간상의 상충 관계를 개선하기 위하여 디지트 시리얼 (digit-serial) 구조의 곱셈기가 제안되었다[5][6][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 패러럴 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 상충 관계를 개선시킬 수 있다.

본 논문의 저자들은 $GF(2^m)$상에서 디지트 시리얼 형태의 곱셈기 구조를 제안한 바 있다[7]. 제안된 곱셈기에 대한 분석 결과, Guo 등 [6] 이 제안한 곱셈기에 비하여 더 간단한 구조를 가지며, 데이터 처리 지연시간이 감소됨을 확인하였다. 또한 처리기 및 이들 사이의 신호 흐름은 단 방향이기 때문에 높은 안정성을 가지고 있을 뿐만 아니라, 규칙적인 구조를 가지고 있기 때문에 m 과 L 에 대해 높은 확장성을 가진다. 만약 입력 데이터가 연속적으로 들어오면 초기 $3N$ 클럭 사이클 후에 N 사이클마다 곱셈 결과 값이 출력된다.

본 논문은 이미 저자들이 제안한 곱셈기의 [7] FPGA 구현에 관련된 연구 결과를 중심으로 기술한다. 설계된 곱셈기를 VHDL을 이용하여 구현하며, 구현된 곱셈기는 디지트의 크기를 L 로 설정했을 경우 연속적인 입력 데이터에 대해 $\lceil m/L \rceil$ 클럭 사이클 비율로 곱셈의 결과를 출력한다. 구현된 곱셈기의 함수 시뮬레이션을 거쳐 알고리즘의 타당성을 검증하고, 합성된 회로에 대한 시간 해석 (timing simulation)을 거친 다음, ALTERA사의 FLEX10K 시리즈 EPF10K100ARC-3 칩을 이용하여 곱셈기를 제작한다. 본 연구에서 구현된 곱셈기를 기존의 곱셈기와 비교 분석한 결과 시간 및 공간 복잡도가 감소되었으며, 간단한 구조로서 데이터 처리 지연시간을 줄일 수 있다.

Ⅱ. MSB 우선 곱셈 알고리즘

본 절에서는 MSB 우선 곱셈 알고리즘을 분석한다[3]. $A(x)$와 $B(x)$는 $GF(2^m)$의 원소이고, $G(x)$는 차수 m 인 원시 기약 다항식이며, 그리고 $P(x)$는 $A(x)B(x)\ mod\ G(x)$의 결과로 나타낼 수 있다. 이때 다항식 $A(x)$, $B(x)$, $G(x)$ 및 $P(x)$는 식 (1) 과 같이 나타낼 수 있다.

$$\begin{aligned} A(x) &= a_{m-1}x^{m-1}+a_{m-2}x^{m-2}+\cdots+a_1x+a_0 \\ B(x) &= b_{m-1}x^{m-1}+b_{m-2}x^{m-2}+\cdots+b_1x+b_0 \\ G(x) &= x^m+g_{m-1}x^{m-1}+g_{m-2}x^{m-2}+\cdots+g_1x+g_0 \\ P(x) &= p_{m-1}x^{m-1}+p_{m-2}x^{m-2}+\cdots+p_1x+p_0 \end{aligned} \quad (1)$$

식 (1)에서 $P(x)$ 는 두 개의 다항식 $A(x)$, $B(x)$ 를 곱한 다음 $G(x)$로 모듈러 연산을 취하면 값을 구할 수가 있으며, 식 (2) 로 나타낼 수 있다.

$$P(x) = A(x)B(x)\ mod\ G(x) \quad (2)$$

표준 다항식 기저를 사용할 경우 곱셈 알고리즘은 LSB 우선 과 MSB 우선 방법으로 구분할 수 있다. LSB 우선 방법은 승수 $B(x)$의 LSB 부터 곱셈을 시작하고, MSB 우선 방법은 MSB 부터 곱셈을 시작하는데, 그림 1에 MSB 우선 곱셈 알고리즘이 주어져 있다[6].

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. $P(x) = p^m(x)$

그림 1. MSB 우선 곱셈 알고리즘

그림 1 의 알고리즘에서 $p_k^{(i)}$는 각각 $p^{(i)}(x)$의 k 번째 계수를 나타내고, a_i와 b_i는 $A(x)$와 $B(x)$의 i 번째 계수를 나타내고, g_k는 $G(x)$의 k 번째 계수를 나타낸다[4].

Ⅲ. 곱셈기 설계

그림 1에 기술된 알고리즘을 바탕으로 $GF(2^m)$ 상의 MSB 우선 곱셈 알고리즘의 데이터 의존 그래프 및 (i, k) 계산점에서의 회로도를 구하면 그림 2 및 그림 3과 같이 주어진다[6]. 이 경우 m=9 이고,

자료의존 그래프는 $m \times m$ 개로 구성되어 있다고 가정한다. 그림 2에 기술된 것처럼 결과 값 $P(x)$는 m 번 반복 후 자료 의존 그래프의 가장 아래 열에서 구할 수 있음을 알 수 있다.

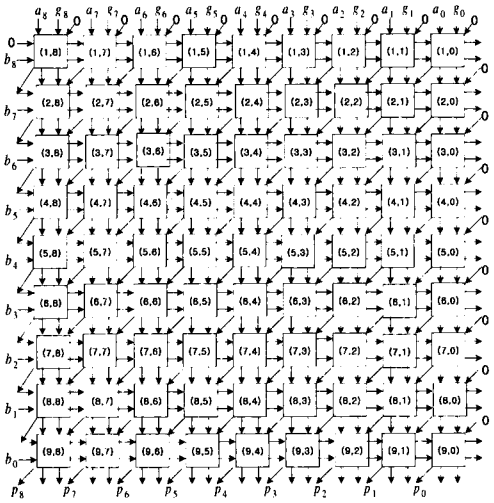


그림 2. $GF(2^9)$상의 데이터 의존 그래프

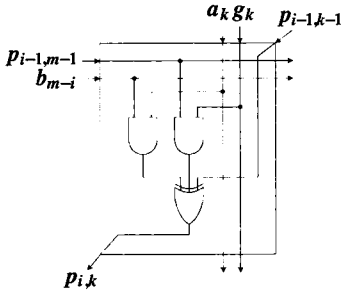


그림 3. 그림 2의 (i, k) 계산점의 회로도

1. 데이터 의존 그래프의 변형

디지트 크기가 L 인 디지트 시리얼 시스톨릭 어레이를 만들기 위해 그림 2를 $L \times L$ 로 묶으면, L-1 개의 일시적인 결과는 인접한 오른쪽 디지트 셀에서 계산이 되어진다. 그림 4는 (L=3, N=m/L=3) 의 경우 변형된 데이터 의존 그래프, 그림 5는 변형된 (i, k) 계산점에서의 회로도를 각각 나타낸다.

그림 4의 데이터 의존 그래프는 규칙적인 구조를 가지고 있지만 오른쪽으로 투영시킬 경우 일차원 신호 흐름 그래프(Signal Flow Graph: SFG)를 얻을 수 없다. 이러한 문제를 해결하기 위해 그림 4의 데이터 의존 그래프를 인덱스 변환시키면 그림 6의 변형된 데이터 의존 그래프를 얻을 수 있다[7].

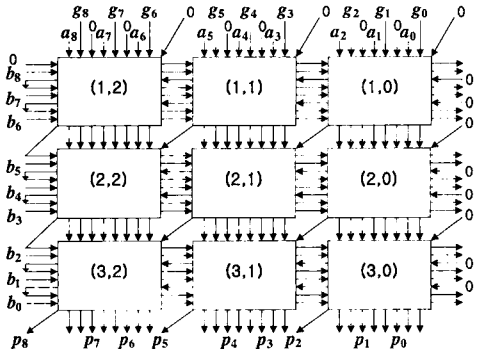


그림 4. 그림 2의 변형된 데이터 의존 그래프

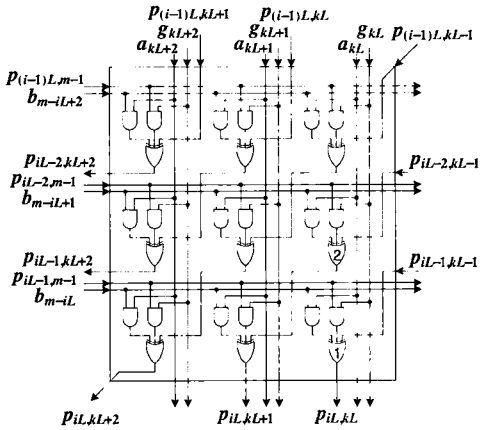


그림 5. 그림 4의 (i, k) 계산점 회로도

그림 6의 데이터 의존 그래프를 동쪽 방향으로 투영시키면 그림 7과 같은 일차원 SFG 를 구할 수 있다. 그림 7에 대응되는 각 PE 의 회로를 구한 다음, 간단한 변환 과정을 거치면 그림 8과 같은 PE 의 구조를 구할 수 있다. 여기서 '⊠' 은 그림 3에 주어진 (i, k) 계산점의 회로를 나타낸다.

2. 데이터 의존 그래프의 컷-셋 시스톨릭화

그림 7의 SFG 에 대하여 컷 셋 시스톨릭화 기법(cut-set systolization techniques) [8]을 적용하면 그림 9와 같은 N개의 Processing Element(PE)로 구성된 디지트-시리얼 시스톨릭 곱셈기의 구조를 얻을 수 있다[7]. 또한 그림 10은 그림 9의 곱셈기를 구성하는 각각의 PE 구조를 나타내며, '•'은 1-비트 1-사이클 지연소자 이고, '⊠'는 그림 3의 (i, k) 계산점 회로를 나타낸다. 그림 9의 곱셈기에 기술된 바와 같이 만약 입력 데이터가 연속적으로 들어오면 초기 $3N$ 클럭 사이클 후에 N 사이클마다 곱셈 결과 값이 출력된다.

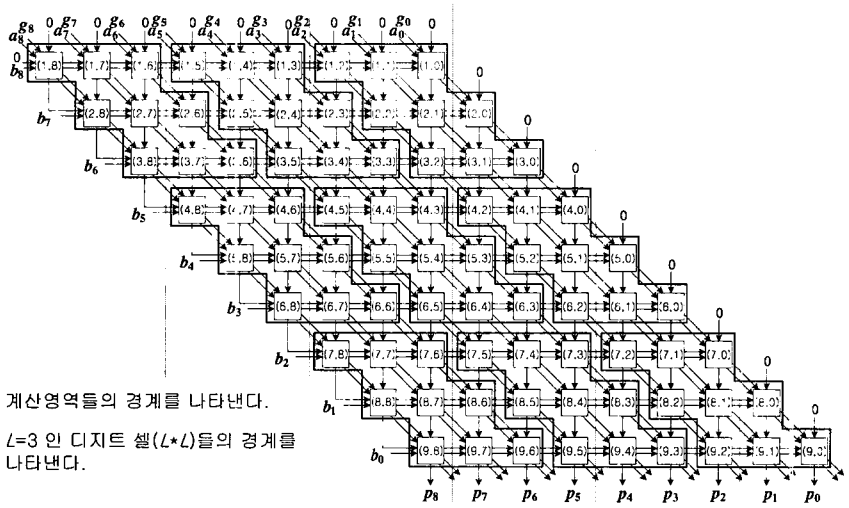


그림 6. 그림 2의 변형된 데이터 의존 그래프

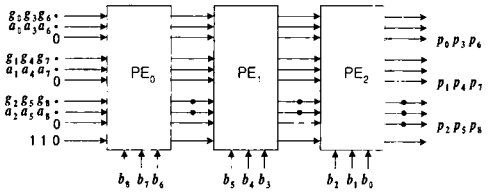


그림 7. L=3일 때 $GF(2^9)$의 디지트 시리얼 곱셈기 구조

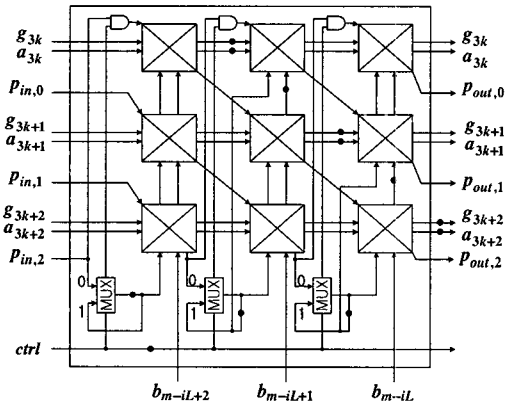


그림 8. 변형된 각 PE의 구조

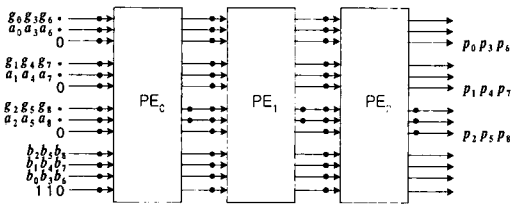


그림 9. L=3일 때 $GF(2^9)$의 디지트 시리얼 곱셈기 구조

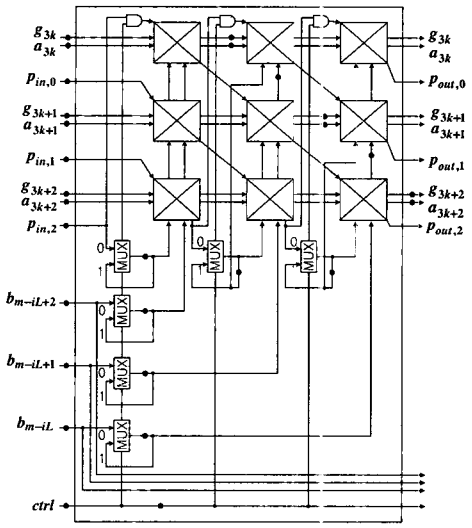


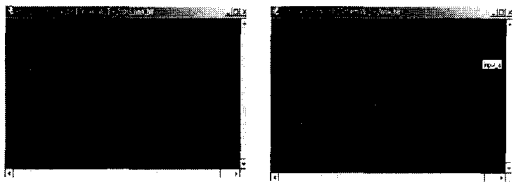
그림 10. 그림 9의 PE 구조 [7]

Ⅳ. FPGA 구현

본 절에서는 3절에서 설계된 곱셈기의 성능을 분석한 다음, 곱셈기의 FPGA의 구현에 대하여 기술한다. 성능을 비교하기 위하여 Guo 등[6] 이 제안한 곱셈기와 본 연구에서 제안한 곱셈기를 동일한 환경에서 구현 후 특성을 분석하였다.

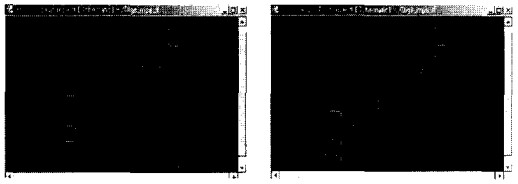
설계된 곱셈기를 VHDL로 기술하였으며, Synopsis사의 FPGA-Express (Version 2000,11-FE3.5)를 이용하여 곱셈기 회로를 합성하였다. 또한 곱셈기의 기능 검증을 위해 Mentographics사의

VHDL- ChipSim을 이용하여 시뮬레이션을 수행하였으며, 대상 FPGA 디바이스로는 ALTERA사의 FLEX10K군인 EPF10K100ARC240-3을 사용했다. 그림 11은 각 곱셈기의 계산점 회로의 합성 결과를 나타내며, 그림 12는 m=9 이고 L=3 인 디지트 시리얼 시스톨릭 곱셈기의 회로 합성 결과를 나타내며, 그림 13은 그림 12를 구성하는 PE 의 합성 결과를 나타낸다.



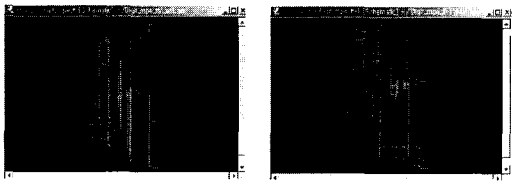
a. Guo 곱셈기 b. 본 연구에서 제안한 곱셈기

그림 11. 각 곱셈기의 계산점 회로



a. Guo 곱셈기 b. 본 연구에서 제안한 곱셈기

그림 12. m=9, L=3 인 디지트 시리얼 곱셈기 회로 합성 결과



a. Guo 곱셈기 b. 본 연구에서 제안한 곱셈기

그림 13. 그림 12에 있는 각각의 PE 회로도

합성된 회로의 기능 검증을 위하여 Synopsis사의 FPGA-Express를 이용하여 회로를 구현했으며, 이로부터 Netlist 파일을 추출한 후, VHDL-ChipSim을 이용하여 시뮬레이션을 수행하였다. 시뮬레이션 과정에서 입력된 테스트 데이터 집합은 식 (3) 과 같다.

$$\begin{aligned} A(x) &= (x^8+x^5+x^4+x,\ x^8+x^7+x^6+x^5+x^3+1) \\ B(x) &= (x^8+x^6+x^5+x^2+1, x^8+x^7+x^3+x^2+x) \\ G(x) &= (x^9+x^4+1,\ x^9+x^4+1) \end{aligned} \quad (3)$$

시뮬레이션 결과는 각각 그림 14과 그림 15에 기술되어 있다. 여기서, 그림 14는 Guo 등[6] 이 제안한 곱셈기에 대한 시뮬레이션 결과이고, 그림 15는 본 연구에서 제안한 MSB 우선 방식의 곱셈기에 대한 시뮬레이션 결과이다. 그림에 기술된 바와 같이 두 개 곱셈기 모두가 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 $P(x)=(x^8+x^5+1,\ x^7+x^6+1)$로 주어져야 하며, 시뮬레이션 결과 그림 14 및 15 에 기술된 것처럼 동일한 결과 식을 얻었다. 또한 본 연구에서는 m=256 일 때, L 값을 2부터 32 까지 변화시키면서 동일한 시험을 수행했으며 시험 결과 모두 정확한 결과를 얻었다.

FPGA 구현을 위하여 m=9, L=3일 때 Max+plusII상에서의 평면도와 프로그래밍 된 핀 배치도를 구했으며, 입출력 핀 배치는 pin파일을 참고하여 설정하였다. 마지막으로 컴파일 된 파일을 프린트 포트를 통해 ALTERA사의 FLEX10K군인 EPF10K100ARC- 240-3 칩으로 다운로드 한 다음, 로직 분석기를 이용하여 곱셈기의 특성을 분석하였다. 특성 분석 결과 합성 회로에 대한 시뮬레이션 결과와 동일한 결과를 얻었으며, 최고 20MHz 클럭에 대하여 정상 작동함을 확인하였다.

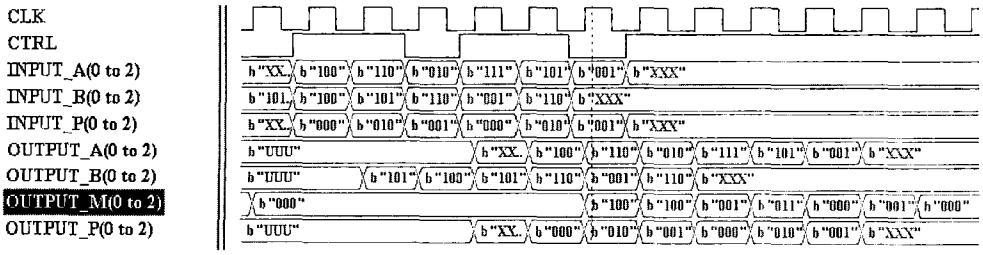


그림 14. Guo 곱셈기에 대한 시뮬레이션 결과

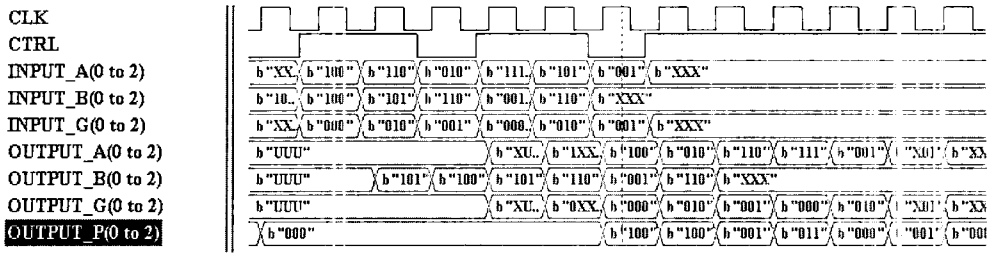


그림 15. 본 연구에서 제안된 곱셈기에 대한 시뮬레이션 결과

V. 성능 분석

본 연구에서 구현된 디지트 시리얼 시스톨릭 곱셈기와 Guo 등[6] 이 제안한 디지트-시리얼 시스톨릭 곱셈기를 비교하였다. 표 1은 두 개 곱셈기의 하드웨어 복잡도 및 데이터 지연시간 측면에서 일반적인 특성을 비교한 결과이다. 여기서 3 입력 XOR 게이트는 2 입력 XOR 게이트 두 개로 구성되고, 4 입력 XOR 게이트는 2 입력 XOR 게이트 세 개로 구성된다고 가정하였다[8]. 표 2는 m=9, L=3일 때 FPGA 구현과 관련하여 지연 시간 및 칩 면적 특성을 비교한 결과이다.

성능 분석 결과 제안한 시스톨릭 곱셈기는 데이터 처리 지연 시간이 $3N(T_{XOR2})$ 만큼 감소하였고, 게이트의 수는 $[\{N(L\text{-}1)\text{-}1\}$Latchs+$(L\text{-}1)$(XOR2+AND2)] 만큼 감소하였으며, 데이터 버스는 Guo 등[6] 이 제안한 곱셈기에 비해 $(L\text{-}1)$ 만큼 들어 줄었다. 따라서 본 논문에서 제안한 디지트 시리얼 곱셈기는 Guo 등[6] 이 제안한 곱셈기보다 더 간단한 하드웨어 회로로 구성이 가능하고, 데이터 처리 지연 시간이 감소됨을 알 수 있다.

표 1. 디지트 시리얼 시스톨릭 곱셈기의 특성 비교

구분	Guo 곱셈기	구현된 곱셈기
처리기 개수	처리기 : N XOR2 : L-1 AND2 : L-1	처리기 : N 1비트 래치 : 1
처리기 복잡도	AND2 : $2L^2+L$ XOR2 : L-1 XOR3 : L^2-$2L$+2 XOR4 : L-1 1비트 래치 : $10L$ MUX : $2L$	AND2 : $2L^2+L$ XOR3 : L^2 1비트 래치 : $9L$+1 MUX : $2L$
지연 시간 (cycles)	$3N$	$3N$
최대 처리기 지연 시간	$T_{AND2}+T_{XOR4}+$ $(L\text{-}1)(T_{AND2}+$ $T_{XOR3}+T_{MUX2})$	$(L\text{-}1)T_{MUX2}+$ $L(T_{AND2}+T_{XOR3})$

$N=m/L$
AND2 : 2-입력 AND gate
XOR2 : 2-입력 XOR gate
XOR3 : 2(2-입력 XOR) gates
XOR4 : 3(2-입력 XOR) gates
MUX : 2-to-1 multiplexer
T_{AND2} : AND2 전달지연시간
T_{XORi} : XORi 전달지연시간
T_{MUX2} : MUX 전달지연시간

표 2. m=9, L=3 때 FPGA 구현을 위한 특성 비교

구분	PE				전체		
	시간		면적		시간	면적	
	클럭 (MHz)	지연 시간 (ns)	D-ff	LUT	지연 시간 (ns)	D-ff	LUT
제안한 곱셈기	20.04	49.90	28	23	449.1	84	69
Guo의 곱셈기	18.98	52.70	30	23	474.3	90	69

D-ff : D flip-flop
LUT : Look-Up Table

VI. 결 론

본 논문에서는 $GF(2^m)$상에서 MSB 우선 방식의 곱셈기를 구현하였다. 구현된 곱셈기를 Guo 등[6] 이 제안한 MSB 우선 방식의 디지트 시리얼 곱셈기와 비교 분석한 결과, 시간 및 공간 복잡도가 감소되었다. 따라서 본 논문에서 제안한 디지트 시리얼 곱셈기는 Guo 등[6] 이 제안한 디지트 시리얼 곱셈기보다 더 간단한 하드웨어 회로로 구성이 가능할 뿐만 아니라, 데이터 처리 지연 시간이 감소됨을 알 수 있다. 또한 제안한 곱셈기는 단 방향의 신호 흐름을 가지기 때문에 안정적이며, 매우 규칙적인 구조를 가지고 있기 때문에 m 과 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.* on 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, "A Digit-Serial Systolic Multiplier for Finite Fields $GF(2^m)$", *Proc. on 5th WSES/IEEE World Multiconference On Circuits, Systems, Communications & Computers*. July 9-12, 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(Chang Hoon Kim) 정회원

2000년 2월 : 대구대학교 컴퓨터정보공학부 학사

2000년 3월~현재 : 대구대학교 컴퓨터정보공학과 석사과정

<주관심 분야> 암호 시스템, 내장형 시스템, 재구성형 컴퓨팅

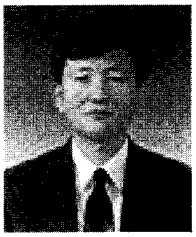
한 상 덕(Sang Duk Han) 정회원

2000년 2월 : 대구대학교 컴퓨터정보공학과 학사

2000년 3월~현재 : 대구대학교 컴퓨터정보공학과 석사과정

<주관심 분야> 마이크로 프로세서 설계, 암호 시스템

홍 춘 표(Chun Pyo Hong) 정회원



1978년 2월 : 경북대학교 전자공학과 학사

1986년 12월 : Georgia Institute of Technology, Electrical and Computer Engineering 석사

1991년 12월 : Georgia Institute of Technology, Electrical and Computer Engineering 박사

1994년 9월~현재 : 대구대학교 정보통신공학부 교수

<주관심 분야> DSP 하드웨어 및 소프트웨어, 컴퓨터 구조, VLSI 신호처리, 내장형 시스템