韓國컴퓨터情報學會 論文誌
第17卷 第3號, 2012. 3.

2012-17-3-1-1

GF(2^m) 상에서의 병렬 승산기 설계에 관한 연구

한 성 일*

A Study on the Construction of Parallel Multiplier over GF(2^m)

Sung-il Han*

요 약

본 논문에서는 계수순환과 기약 삼항식을 적용하여 시스템 복잡도를 개선한 GF(2^m)상의 승산기 구성방법과 구현회로를 제안하였다. 제안된 회로는 병렬 입출력 구조를 가지며, 승산항의 계수 순환과 기약 삼항식을 적용한 모듈로 연산을 하는 회로 구성의 특성상 기존의 타 논문에 비해 회로 복잡도가 감소함을 보였다. 본 논문에서 제안한 회로의 시스템 복잡도는 $2m^2$개의 2-입력 AND 게이트, $m \cdot (m+2)$개의 2-입력 XOR 게이트의 회로복잡도이며, 메모리나 스위치 등의 별도의 소자는 필요하지 않다. 연산에 소요되는 최대 지연시간은 $T_A + (2 + \lceil \log_2 m \rceil) T_X$ 이다. 본 논문에서 제안한 회로는 간단하고, 정규성을 보이며, 모듈구성이 가능하기 때문에 VLSI 회로 구성에 상대적으로 적합하다.

▸ Keyword : 유한체, 기약 삼항식, 승산기, 유한체 승산연산

Abstract

A low-complexity Multiplication over GF(2m) and multiplier circuit has been proposed by using cyclic-shift coefficients and the irreducible trinomial. The proposed circuit has the parallel input/output architecture and shows the lower-complexity than others with the characteristics of the cyclic-shift coefficients and the irreducible trinomial modular computation. The proposed multiplier is composed of $2m^2$ 2-input AND gates and $m \cdot (m+2)$ 2-input XOR gates without the memories and switches. And the minimum propagation delay is $T_A + (2 + \lceil \log_2 m \rceil) T_X$. The Proposed circuit architecture is well suited to VLSI implementation because it is simple, regular and modular.

▸ Keyword : Finite field, Irreducible trinomial, multiplier, Finite field multiplication

• 제1저자 : 한성일
• 투고일 : 2011. 09. 29, 심사일 : 2011. 12. 13, 게재확정일 : 2012. 01. 12.
* 인덕대학 정보통신과(Dept. of Information & Telecommunication, Induk College)
※본 연구는 인덕대학 교내연구비지원사업의 지원으로 진행되었음.

I. 서 론

유한체(Galois field)는 스위칭 이론, 오류 정정 부호, 디지털 신호 처리 및 화상 처리, 디지털 통신의 암호화 및 해독화를 요하는 보안, 통신등에 많이 응용되고 있다. 특히, $GF(2^m)$ 상에서의 연산은 신호 처리와 화상처리 응용 분야에서 특별한 계산을 요하거나 범용 컴퓨터 계산의 고속화를 보조하는 고성능 전용 컴퓨터의 설계에 효과적이며, VLSI 설계에 응용되고 있다.[1-2]

유한체를 구성하는 원소들은 표준, 정규, 쌍대기저 등에 의해 각 형식에 따른 다항식으로 표현되며, 각 기저의 특성에 따라 연산의 효율성과 회로구현의 용이성이 달라진다. 일반적으로 표준기저의 경우 타 기저에 비하여 기약다항식의 선택이 자유롭고, 이를 사용한 가산과 승산은 여타 연산의 기반이 되는 연산으로 활용되며 특히 오류정정 부호나 공개키 암호화 과정에서 사용되는 Product-sum(AB+C) 연산이나, Power-sum(AB^2+C) 연산은 유한체 승산 및 가산이 반복 적용되는 예이다.[3-4]

승산기의 구현 예로는 Yeh등[5]은 표준 기저 표현식을 사용하여 유한체상의 승산을 구현하는 직렬입력/직렬출력 시스토릭 배열 구조와 병렬입력/병렬출력 시스토릭 배열 구조의 승산기를 제안하였다. Scott등[6]은 표준 기저로 표현된 각 원소들의 유한체 승산을 실행하는 고속 승산기를 제시하였고, Wang등[7]은 Scott등이 제안한 유한체상의 승산 알고리즘을 이용하여 시스토릭 배열의 승산기를 제시하였다. 그러나 이들이 제시한 승산기는 레지스터를 이용하기 때문에 클럭시간을 필요로 한다.

한편, Mastrovito[8]에 의해서 기약 3항식 x^m+x+1에 대한 승산 알고리즘이 제안되었으며, 이를 바탕으로 Sunar[9]에 의하여 기약 3항식 x^m+x^n+1을 이용한 승산기가 제안되었으나 n을 정하여 행렬식으로 수식을 전개하는 과정을 필요로 한다. 일반적으로 표준기저를 사용하는 승산기 중에서 삼항식을 사용하는 경우는 적은 회로 복잡도를 갖는다.

유한체 승산연산의 설계를 위한 연구와 함께 AB^2+C 연산회로를 개발하기 위한 연구가 진행되었고, 최근 wei[10], wang[11], Lee[12], Byun[13], Ku[14] 등이 그 연산회로를 보였다. wei는 MUX와 DEMUX를 사용하여 8 가지의 계산을 수행하는 기본 셀을 제시하여 AB^2+C 연산 및 역원생성, 제산 등에 효율적인 회로를 제시하였으며, wang은 표준기저를 사용하여 하나의 단위 셀에 6개의 AND 게이트와 6개의 XOR 게이트와 17개의 1-bit latch가 사용되는 단방향 데이터 흐름에 대한 AB^2+C 연산기 회로를 제안하였다. Lee는 기약 AOP를 기반으로 하는 AB^2+C 연산기 회로를 하나의 단위 셀에 각각 1개씩의 AND와 XOR 게이트가 사용되고 3개의 1-bit latch로 구성하여 제시하였다. 그러나 이러한 회로들은 각각의 회로가 갖는 회로의 복잡도와 래치회로에 의한 상대적으로 긴 지연시간 때문에 고속 대용량의 연산처리가 필요한 오류정정 부호나 암호화 등의 분야에서 사용하기에는 제한이 있다고 할 수 있다. Byun은 래치회로가 없는 AOP 기반의 AB^2+C 연산기 회로를 제안하였으며, 고속 대용량의 연산에 적합하나, AOP를 사용하는 회로의 특성상 기약다항식이 고정이 되는 단점이 있다.

본 논문에서는 이러한 연구동향을 바탕으로 기약삼항식을 적용하여 시스템 복잡도를 낮춘 새로운 $GF(2^m)$상의 AB^2+C 연산기법과 구현회로를 제안하였다. 본 논문에서 제안한 회로는 $2m^2$개의 2-입력 AND 게이트, $m \cdot (m+1)$개의 2-입력 XOR 게이트로 구성되며, 메모리나 스위치 등의 별도의 소자는 필요하지 않다. 입력신호에서 출력신호가 도출될 때까지의 전파지연시간은 Product-sum 연산과 power-sum 연산의 경우 $T_A+(2+\lceil \log_2 m \rceil)T_X$ 이다. 본 논문에서 제안한 회로는 모듈형태의 구성이 가능하고 시스템의 복잡도를 감소하였기 때문에 회로구성 시 보다 유리하다고 할 수 있으며, 결선구조 및 게이트의 배열 등이 규칙적인 배열 구조를 보임으로 규칙성 동일성이 있어 VLSI 회로 구성에 유리하다고 하겠다.

본 논문의 구성은 다음과 같다. I장의 서론에서는 기존 승산기의 연구에 대해서 정리하고 본 논문에서 적용한 이론에 대한 배경을 설명하였으며, II 장에서는 유한체의 가산 및 승산에 대한 이론적 배경 및 성질을 설명하였으며, III 장에서는 이를 바탕으로 기약 삼항식을 사용한 $GF(2^m)$상의 승산전개 기법 및 AB^2 연산기법을 보였다. IV 장에서는 전장에서 논의한 내용을 바탕으로 새로운 $GF(2^m)$상의 병렬 연산기를 설계하였다. V장에서는 본 논문의 구성과 장점을 타 논문과 비교하여 설명하였고, 이에 대한 결론을 서술하였다.

II. $GF(2^m)$상의 연산

1. 유한체상의 원소표현

유한체 $GF(2^m)$은 양의 정수 m에 대하여 2^m개의 원소들

로 구성된 체이며, 원소들 간의 연산이 사칙연산에 닫혀있으며, 교환 및 결합법칙이 성립하고, 항등원 및 역원이 존재한다. 유한체 $GF(2^m)$은 0과 1을 원소로 하는 기초체 $GF(2)$를 m차원으로 확장한 확장체이며, $GF(2^m)$상의 모든 연산은 모듈로(modulo) 2 연산을 기반으로 이루어진다. 0을 제외한 $GF(2^m)$상의 모든 원소들은 원시원소 α에 의해 표현되며, α는 기약다항식 식 (1)의 근이 된다.

$$F(x)=f_0+f_1x+\cdots+f_{m-1}x^{m-1}+x^m \tag{1}$$

따라서 $F(\alpha)=0$이 되며 다음 식 (2)가 성립한다.

$$\alpha^m=f_0+f_1\alpha+\cdots+f_{m-1}\alpha^{m-1} \tag{2}$$

이에 따라 $GF(2^m)$상의 모든 원소들은 m보다 낮은 차수를 갖는 α의 다항식으로 표현되며 다항식의 각 기저들을 표준기저라 한다.

표준기저를 적용하여 $GF(2^m)$상의 두 원소 A와 B를 정의하면 다음의 식 (3)과 같이 정의할 수 있다.

$$\begin{aligned}A&=a_0+a_1\alpha+\cdots+a_{m-1}\alpha^{m-1}\\ B&=b_0+b_1\alpha+\cdots+b_{m-1}\alpha^{m-1}\end{aligned} \tag{3}$$

여기서, 각 계수들은 모두 $GF(2)$에 속한다.

2. 유한체상의 가산 및 승산 연산

유한체 $GF(2^m)$상의 두 원소 A와 B의 가산연산을 정의하면 아래 식 (4)와 같다. 두 식에서 보는 바와 같이 $GF(2^m)$상의 가산은 가산 후 발생하는 자리올림을 고려하지 않아도 되므로 매우 쉽고 간단하게 이루어지는 특징이 있고, 승산을 비롯한 제산, 역산 등의 연산에서는 기약다항식을 적용한 모듈로 계산 과정이 적용되므로 연산과정이 가산에 비하여서는 복잡한 성질이 있다.

$$\begin{aligned}A+B=&(a_0\oplus b_0)+(a_1\oplus b_1)\alpha+\cdots\\&+(a_{m-2}\oplus b_{m-2})\alpha^{m-2}+(a_{m-1}\oplus b_{m-1})\alpha^{m-1}\end{aligned} \tag{4}$$

유한체 $GF(2^m)$상의 기약다항식 중 모든 항의 계수가 1인 다항식 $F(x)=x^m+\cdots+x+1$에 근 α를 대입하면 $F(\alpha)=\alpha^m+\cdots+\alpha+1=0$가 성립하며, 이를 α^m에 대하여 전개하면 다음의 식 (5)와 같다.

$$\alpha^m=\alpha^{m-1}+\cdots+\alpha+1 \tag{5}$$

식 (5)으로부터 m 이상의 차수를 갖는 항은 m-1 이하의 항으로 표현되며, $1\le i\le 2(m-1)$ 인 정수 i를 사용하여 다시 표현하면 식 (6) 과 같다.

$$\alpha^{m+i}=\alpha^{i-1} \tag{6}$$

표준기저를 사용하여 (m-1)차 이하의 다항식으로 표현된 $GF(2^m)$상의 두 원소 A 와 B를 식 (7), (8)과 같이 표현하면

$$\begin{aligned}A(\alpha)&=a_{m-1}\alpha^{m-1}+a_{m-2}\alpha^{m-2}+\cdots+a_1\alpha^1+a_0\\&=a_{m-1}\alpha^{m-1}+A_{m-2}(\alpha)\end{aligned} \tag{7}$$

$$\begin{aligned}B(\alpha)&=b_{m-1}\alpha^{m-1}+b_{m-2}\alpha^{m-2}+\cdots+b_1\alpha^1+b_0\\&=b_{m-1}\alpha^{m-1}+B_{m-2}(\alpha)\end{aligned} \tag{8}$$

승산 $C=AB$ 는 다음 식 (9)와 같이 정리되며

$$\begin{aligned}C(\alpha)=&\sum_{j=0}^{m-1}a_jb_j\alpha^{2j}\\&+\sum_{j=0}^{m-2}[a_{j+1}B_j(\alpha)+b_{j+1}A_j(\alpha)]\alpha^{j+1}\end{aligned} \tag{9}$$

승산연산은 다항식의 계수들을 곱하여 기약다항식 $F(\alpha)$에 의한 모듈로연산을 수행한 결과를 갖는다. $C(\alpha)$에 $\mathrm{mod}F(\alpha)$를 적용하여 (m-1)차의 유도된 다항식을 $P(\alpha)$라 하면 이는 다음의 식 (10)과 같다.

$$\begin{aligned}P(\alpha)&=A(\alpha)B(\alpha)\mathrm{mod}F(\alpha)=C(\alpha)\mathrm{mod}F(\alpha)\\&=p_0+p_1\alpha+\cdots+p_{m-1}\alpha^{m-1}\end{aligned} \tag{10}$$

3. 곱행렬 곱셈[8-9]과 Cyclic-Shift-Left[10] 성질

Mastrovito는 유한체 $GF(2^m)$상의 승산연산을 다항식과 기약다항식의 결합된 모양의 승산 매트릭스로 정의하여 다음과 같은 승산 방법을 제안하였으며, 그 후 기약다항식이 삼항식인 경우와 등간격 기약다항식 등에 대한 다양한 연구가 이루어졌다.

$f(x)$를 유한체 $GF(2^m)$상의 기약다항식이라 하고, $c(x)$는 다항식 $a(x)$와$b(x)$의 곱이라 하자. 이 때 다항식은 모두 $GF(2^m)$상의 원소이다. 유한체 상의 곱연산은

$c(x)=(a(x)\cdot b(x))\mathrm{mod}f(x)$로 표현되며, 곱연산과 모듈로연산을 통합하여 곱행열 M을 정의하면 $c=M\cdot b$가 되며 여기서, c와 b는 $c(x)$와 $b(x)$의 계수 행렬이다. Mastrovito는 주어진 기약다항식을 사용하여 곱행렬을 추출할 경우 다항식의 차수보다 커지게 되는 차수의 항들은 인접한 두 개의 계수 행렬 간의 XOR를 통해서 추출할 수 있음을 보였다.

유한체 상에서의 다항식의 곱과 모듈로 연산 과정을 행렬 및 계수 벡터를 통해서 표현하고자 한다.

유한체 $GF(2^m)$상의 기약다항식이 표준기저를 사용하는 경우 $f(x)=1+f_1x+\cdots+f_{m-1}x^{m-1}+x^m$ 이고 $c(x)$는 다항식 $a(x)$와$b(x)$의 곱이라 하자. 이 때 다항식은 모두 $GF(2^m)$상의 원소이다. 다항식의 곱을 $d(x)=a(x)\cdot b(x)$이라 하고 행렬로 표현하면 $d=A\cdot b$ 이고 d와 b는 계수행렬을 의미한다.

이 때 행렬 A는 다음 식 (11)과 같고,

$$A=\begin{bmatrix} a_0 & 0 & \cdots & 0 & 0 \\ a_1 & a_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m-1} & a_{m-2} & \cdots & a_1 & a_0 \\ 0 & a_{m-1} & \cdots & a_2 & a_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{m-1} & a_{m-2} \\ 0 & 0 & \cdots & 0 & a_{m-1} \end{bmatrix} \quad (11)$$

모듈로 연산을 수행하면, $c(x)=d(x) \bmod f(x)$ 이고 다음 식 (12)와 같다.

$$C(x)=\sum_{k=0}^{2m-2} d_k x^k \bmod f(x) \quad (12)$$
$$=\sum_{k=0}^{m-1} d_k x^k+\sum_{k=m}^{2m-2} d_k (x^k \bmod f(x))$$

Mastrovito는 기약다항식의 계수 및 비트이동에 의한 계수행렬 U와 아이덴티티 행렬을 사용하여 곱행렬 M을 $M=[I_{m\times m}, U]\bullet A$ 으로 제안하였으며, $c=[I_{m\times m}, U]\bullet d$ 에서 다음 식을 통해 승산연산을 완성하였다.

$c=[I_{m\times m}, U]\bullet A\bullet b$

여기서 기약다항식 계수와 계수의 비트이동에 의한 행렬 F는 식 (13)과 같고

$$F=\begin{bmatrix} f_0 & 0 & \cdots & 0 \\ f_1 & f_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & f_0 \\ f_{m-1} & f_{m-2} & \cdots & f_1 \end{bmatrix} \quad (13)$$

행렬 F의 각 계수간 모듈합을 통하여 행렬 U를 다음 식 (14)와 같이 구할 수 있다.

$$U=\begin{bmatrix} f_0 & f_0\oplus 0 & \cdots & 0 \\ f_1 & f_1\oplus f_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & f_1\oplus f_0 \\ f_{m-1} & f_{m-1}\oplus f_{m-2} & \cdots & f_2\oplus f_1 \end{bmatrix} \quad (14)$$

예제 1) 유한체 $GF(2^m)$상의 두 원소 $a(x)=x^4+x^2$ 과 $b(x)=x^3+x^2+1$ 의 곱셈을 기약다항식이 $f(x)=x^5+x^4+x^3+x^2+1$ 인 경우 수행하면,

$a=[00101]^T$ 이고 $b=[10110]^T$ 이며, $f=[10111]^T$ 이다. 따라서 비트이동 행렬 F는 식 (15)와 같고

$$F=[f, f[\downarrow 1], \cdots, f[\downarrow m-2]]=\begin{bmatrix} 1000 \\ 0100 \\ 1010 \\ 1101 \\ 1110 \end{bmatrix} \quad (15)$$

서로 이웃한 열끼리의 논리합에 의해 구해지는 행렬 U는 다음 식 (16)과 같다.

$$U=\begin{bmatrix} 1100 \\ 0110 \\ 1111 \\ 1011 \\ 1001 \end{bmatrix} \quad (16)$$

$c=M\bullet b$ 에서 $M=[I_{m\times m}, U]\bullet A$ 이므로

$c=M\bullet b=M=[I_{m\times m}, U]\bullet A\bullet b=$

$$\begin{bmatrix} 1000001100 \\ 0100001100 \\ 0010011111 \\ 0001010111 \\ 0000110011 \end{bmatrix}\bullet\begin{bmatrix} 00000 \\ 00000 \\ 10000 \\ 01000 \\ 10100 \\ 01010 \\ 00101 \\ 00010 \\ 00001 \end{bmatrix}\bullet\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$=\begin{bmatrix} 01111 \\ 00111 \\ 11100 \\ 00001 \\ 11111 \end{bmatrix}\bullet\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}=x^4 \text{ 이다.}$$

한편 유한체 상에서 AB^2을 구하기 위한 승산 알고리즘을 $GF(2^m)$상의 임의의 원소 $A(\alpha)$에 대하여 식 (7)과 (8)을 적용하면 다음과 같다.

$GF(2^m)$상의 임의의 원소 $A(\alpha)$에 대하여 α를 누승한 결과는 A의 각 계수들이 누승과 동일한 횟수만큼 상위차수로 이동하며, m+1 차수의 계수는 최저차수로 순환 이동한다. 이러한 누승에 대한 순환 이동을 식으로 표현하여 정리하면 다음 식 (17)과 같다.

$$\alpha^i=A_{<m-i+1>}+A_{<m-i+2>}+\cdots+A_{<m-i>}\alpha^m$$
$$=\sum_{j=0}^{m} A_{<j-i>}\alpha^j,\ i=0,1,2,\cdots,m \quad (17)$$
$$={}^{(i)}A$$

위와 같은 성질을 AB^2에 적용하면, $GF(2^m)$상의 임의의 원소 $B(\alpha)$의 거듭제곱은 유한체의 성질에 의하여 다음 식 (18)과 같이 전개되며

$$B^2 = (B_0 + B_1\alpha + \cdots + B_{m-1}\alpha^{m-1} + B_m\alpha^m)^2 \quad (18)$$
$$= B_0 + B_1\alpha^2 + \cdots + B_{m-1}\alpha^{2(m-1)} + B_m\alpha^{2m}$$

위의 순환 이동식과 유한체의 승산식을 적용하여 AB^2 연산을 전개하면 다음의 식 (19)와 같고 이를 통한 병렬 승산기가 제안되었다.

$$P = AB^2 = (\sum_{i=0}^{m} A_i\alpha^i)(\sum_{k=0}^{m} B_k\alpha^{2k})$$
$$= \sum_{k=0}^{m} B_k(\sum_{i=0}^{m} A_i\alpha^i)\alpha^{2k} \quad (19)$$
$$= \sum_{k=0}^{m} B_k(\sum_{i=0}^{m} A_{<i-2k>}\alpha^i) = \sum_{k=0}^{m} B_k(^{(2k)}A)$$

이는 유한체 상에서의 기약다항식이 AOP인 경우 거듭제곱과의 승산인 경우는 거듭제곱에 해당하는 원소의 계수들의 순환 이동에 대한 계수간 논리 곱으로 표현됨을 나타낸다.

4. 계수순환 모듈러 가산에 의한 승산 계수행렬의 정의

유한체 상에서의 승산을 동일차수의 계수들 간의 논리곱으로 간략화하기 위하여 기약다항식 연산이 적용된 승산계수행렬을 정의한다. 승산항의 계수를 기약다항식에 의해서 처리하기 위하여 승산항의 계수를 순환이동한 후 해당 기약다항식에 따라서 모듈러가산을 수행하는 블록을 구성하고 이를 수행하는 모듈을 계수순환 모듈러가산 연산부라 명한다. 해당 블록의 연산결과는 유한체 상의 승산에 있어서의 승산 계수 행렬이 도출된다.

유한체 상에서의 승산연산은 승산항의 계수순환과 기약다항식에 의한 모듈러 가산 및 피승산항과의 SOP 연산으로 정의될 수 있다. 계수순환은 승산항의 승산이 수행되는 횟수에 의해서 결정되고 모듈러가산은 기약다항식에 의해서 결정되며 SOP 블록은 승산계수행렬과 피승산항의 계수간의 논리곱에 의한 부분합으로 구성할 수 있다. 다음의 그림 1은 계수순환 모듈러가산에 의한 승산계수행렬을 사용한 승산과정의 블록도를 나타낸다.

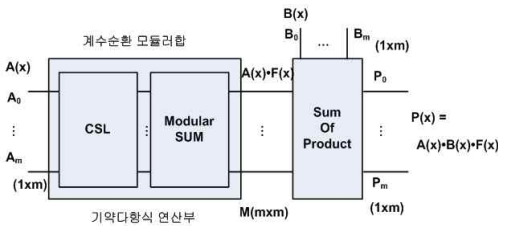

그림 1. 승산계수행렬을 사용한 승산연산의 구성도
Fig. 1. Architecture of Multiplication using coefficient Matrix

III. 기약 삼항식을 적용한 승산연산

본 장에서는 기약삼항식을 적용한 승산 연산의 장점을 소개하고 기존의 AOP를 통하여 정리된 승산 연산에 비하여 연산 비트가 감소된 새로운 승산 연산을 제안한다.

1. 기약 삼항식을 사용한 승산 연산

유한체 $GF(2^m)$상의 두 원소 A와 B의 승산 연산을 기약삼항식 $F(x) = x^m + x + 1$ 과 함께 정의하면 모든 m승항 이상의 항들은 유한체와 기약삼항식의 성질에 의해서 $x^m = x + 1$ 로 치환되어 간략하게 정의된다.

$GF(2^m)$상의 두 원소 A(x)와 B(x)의 승산연산을 전개한다. A(x)와 B(x)를 각각 다음 식 (20)과 (21)과 같이 표현한다.

$$A(x) = a_0 + a_1x + \cdots + a_{m-1}x^{m-1} = \sum_{i=0}^{m-1} a_i\,x^i \quad (20)$$

$$B(x) = b_0 + b_1x + \cdots + b_{m-1}x^{m-1} = \sum_{i=0}^{m-1} b_i\,x^i \quad (21)$$

두 다항식의 곱 P(x)는 다음 식 (22)와 같으며,

$$P(x) = A(x) \bullet B(x) = \quad (22)$$
$$= A(x) \bullet \sum_{i=0}^{m-1} b_ix^i = \sum_{i=0}^{m-1} b_i(A(x) \bullet x^i)$$

다항식의 계수를 사용하여 행렬식으로 표현하면 다음 식 (23)과 같고, $F(x) = x^m + x + 1$ 의 기약 삼항식으로 모듈러 연산을 수행한 결과는 다음의 식 (24)와 같다.

$$[p_0\ p_1\ \cdots\ p_{m-1}] = [b_0\ b_1\ \cdots\ b_{m-1}] \bullet \begin{bmatrix} A(x) \\ A(x)x \\ \vdots \\ A(x)x^{m-1} \end{bmatrix} \quad (23)$$

$$[p_0\ p_1\ \cdots\ p_{m-1}] = [b_0\ b_1\ \cdots\ b_{m-1}]$$
$$\bullet \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots & a_{m-2} & a_{m-1} \\ a_{m-1} & (a_0+a_{m-1}) & a_1 & a_2 & \cdots & a_{m-3} & a_{m-2} \\ a_{m-2} & (a_{m-1}+a_{m-2}) & (a_{m-1}+a_0) & a_1 & \cdots & \vdots & \vdots \\ a_{m-3} & (a_{m-2}+a_{m-3}) & (a_{m-1}+a_{m-2}) & (a_{m-1}+a_0) & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_2 & (a_2+a_3) & (a_3+a_4) & \cdots & \cdots & \ddots & a_1 \\ a_1 & (a_1+a_2) & (a_2+a_3) & \cdots & \cdots & (a_{m-1}+a_{m-2}) & (a_{m-1}+a_0) \end{bmatrix} \quad (24)$$

다음의 예제 2는 기약 삼항식을 사용한 승산의 예 중에서 m이 4인 경우에 대한 승산연산이며, 특정 m 값이 아닌 일반적인 경우에는 m 값의 증가에 따른 회로의 구성만 고려하면 동일한 방법으로 회로를 구성할 수 있다.

예제 2) 위의 승산에 대하여 $F(x) = x^4 + x + 1$을 적용하

여 승산 연산을 다시 전개하면 $x^4 = x+1$과 같고 m이 4인 경우와 같으므로 연산결과는 다음 식 (25)와 같다.

$$\begin{aligned} A(x) \bullet 1 &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 \\ A(x) \bullet x &= a_0 x + a_1 x^2 + a_2 x^3 + a_3 (x+1) \\ &= a_3 + (a_0 + a_3) x + a_1 x^2 + a_2 x^3 \\ A(x) \bullet x^2 &= a_2 + (a_2 + a_3) x + (a_0 + a_3) x^2 + a_1 x^3 \\ A(x) \bullet x^3 &= a_1 + (a_1 + a_2) x + (a_2 + a_3) x^2 + (a_3 + a_0) x^3 \end{aligned} \quad (25)$$

이를 행렬로 표현하면 다음과 같고 따라서 승산결과 P(x)의 계수는 다음 식 (26)과 같이 구할 수 있다.

$$[p_0\ p_1 \cdots p_3] = [b_0\ b_1 \cdots b_3] \bullet \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_3 & (a_0 + a_3) & a_1 & a_2 \\ a_2 & (a_2 + a_3) & (a_3 + a_0) & a_1 \\ a_1 & (a_1 + a_2) & (a_2 + a_3) & (a_3 + a_0) \end{bmatrix} \quad (26)$$

이들 계수는 최고차 항의 계수가 최저차 항의 계수가 되는 계수의 순환 이동과 함께 기약다항식의 계수가 있는 차수의 항의 계수간의 논리곱에 의해서 결정되며, 이는 승산 연산과 기약 다항식에 의한 모듈로 연산이 동시에 진행됨을 의미한다. 이에 대한 연산과정을 승산회로로 구성하여 그림 2에 도시하였다.

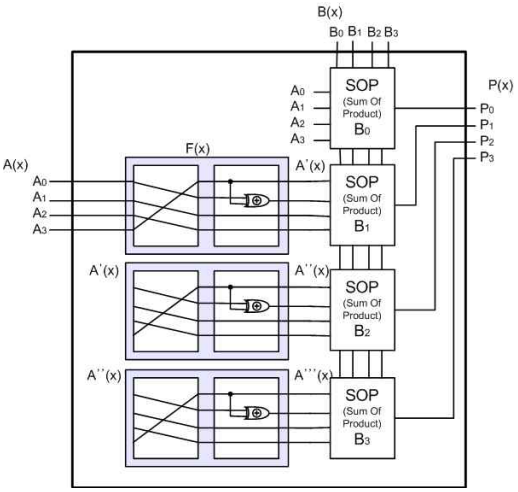

그림 2. 예제의 기약 삼항식을 사용한 승산기의 구성도
Fig. 2. Architecture of Multiplier using Irreducible Trinomial

위의 그림 2에서 F(x) 블록은 승산연산과 모듈로연산이 동시에 이루어지는 기약삼항식 연산부를 의미하며, SOP(Sum Of Product) 블록은 계수들 간의 곱연산과 합연산을 의미한다.

2. 기약 삼항식을 사용한 $A \bullet B^2$ 연산

유한체 $GF(2^m)$상의 두 원소 A와 B^2의 승산연산을 기약 삼항식을 사용하여 전개하여 그 결과가 승산연산과 같이 계수간의 순환이동과 기약 다항식의 계수가 있는 차수의 항의 계수간 논리곱에 의해서 결정됨을 증명하고, 이를 통해 승산연산 및 모듈로 연산의 일반화를 도출하여 새로운 $A \bullet B^2$승산기의 구성을 제안하였다.

$GF(2^m)$상의 두 원소 A(x)와 $B(x)^2$의 승산연산을 전개한다.

A(x)와 B2(x)를 각각 다음 식 (27)과 (28)과 같이 표현한다.

$$A(x) = a_0 + a_1 x + \cdots + a_{m-1} x^{m-1} = \sum_{i=0}^{m-1} a_i x^i \quad (27)$$

$$B(x)^2 = (b_0 + b_1 x + \cdots + b_{m-1} x^{m-1})^2 = \sum_{i=0}^{m-1} (b_i x^i)^2 \quad (28)$$

여기서 유한체의 성질에 의하여 거듭제곱은 각 차수의 변수의 거듭제곱과 같으므로 식 (28)은 다음의 식 (29)와 같고

$$\begin{aligned} B(x)^2 &= B(x)^2 \bmod F(x) \\ &= b_0 + b_1 x^2 + \cdots + b_{m-1} x^{2m-2} = \sum_{i=0}^{m-1} b_i x^{2i} \end{aligned} \quad (29)$$

이를 승산에 적용하면, 두 다항식의 곱 P(x)는 다음 식 (30)과 같다.

$$\begin{aligned} P(x) &= A(x) \bullet B(x)^2 \bmod F(x) \\ &= A(x) \bullet \sum_{i=0}^{m-1} b_i x^{2i} = \sum_{i=0}^{m-1} b_i (A(x) \bullet x^{2i}) \end{aligned} \quad (30)$$

연산의 결과를 위에서 언급한 식 (12)의 순환이동을 적용한 전개식으로 치환하면 다음 식 (31)과 같다.

$$\begin{aligned} P(x) &= A(x) \bullet B(x)^2 \bmod F(x) \\ &= \sum_{i=0}^{m} B_k (^{2k} A(x)) \bullet x^{2i} \bmod F(x) \end{aligned} \quad (31)$$

이는 기약 삼항식을 기반으로 하여 정규기저로 표현된 유한체 $GF(2^m)$상의 두 원소 A와 B에 대하여 AB^2의 연산을 피승산항(A)의 순환이동과 승산항(B2)의 각 계수를 순차적이고 반복적으로 승산한 후 동일 차수의 계수들을 기약 삼항식에 의해서 모듈로 가산함으로써 구할 수 있음을 나타낸다. 이러한 승산의 전개를 순환이동, 모듈로연산, 각 계수의 부분곱 등의 회로구성으로 다음의 예제 3)의 연산을 도시하면 그림 3과 같다.

예제 3) 위의 AB^2승산에 대하여 $F(x) = x^4 + x + 1$을 적용하여 승산 연산을 다시 전개하면 $x^4 = x+1$과 같으므로 연산결과는 다음 식 (32)와 같다.

$$A(x) \cdot 1 = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$
$$A(x) \cdot x^2 = a_2 + (a_2 + a_3)x + (a_0 + a_3)x^2 + a_1 x^3$$
$$A(x) \cdot x^4 = (a_3 + a_0) + (a_3 + a_0 + a_1)x + (a_1 + a_2)x^2 + (a_2 + a_3)x^3$$
$$A(x) \cdot x^8 = (a_0 + a_2) + (a_1 + a_2 + a_3)x + (a_2 + a_3 + a_0)x^2 + (a_1 + a_3)x^3 \quad (32)$$

이를 행렬로 표현하면 다음과 같고 따라서 승산결과 P(x)의 계수는 다음 식 (33)과 같이 구할 수 있다.

$$[p_0\, p_1 \cdots\ p_3] = [b_0\, b_1 \cdots\ b_3] \cdot \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_2 & (a_2+a_3) & (a_3+a_0) & a_1 \\ (a_3+a_0) & (a_3+a_0+a_1) & (a_1+a_2) & (a_2+a_3) \\ (a_0+a_2) & (a_1+a_2+a_3) & (a_2+a_3+a_0) & (a_1+a_3) \end{bmatrix} \quad (33)$$

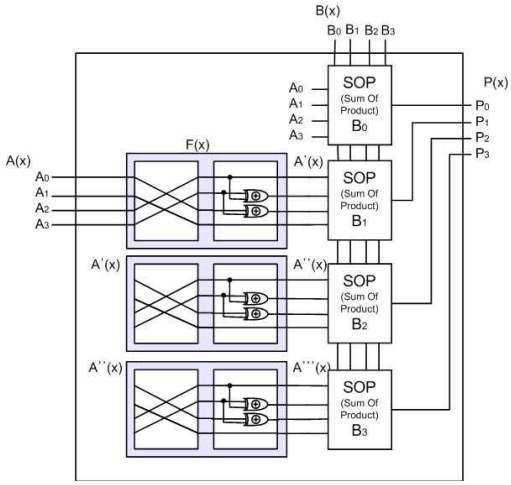

그림 3. 예제의 기약 삼항식을 사용한 AB^2 연산기의 구성도
Fig. 3. Architecture of AB^2 Multiplier using Irreducible Trinomial

IV. 기약 삼항식을 적용한 $GF(2^m)$상의 승산기구성

본 장에서는 기약 삼항식을 기반으로 하여 표준기저로 표현된 $GF(2^m)$상의 두 원소 A와 B에 대하여 승산 및 AB^2+C 연산을 피 승산항 계수의 순환이동을 수행하는 순환이동 블록, 순환 이동된 피 승산항 계수와 승산항 계수간의 부분 곱을 수행하는 부분곱 연산블록 및 동일 차수의 계수들 간의 모듈러 계산을 수행하는 기약삼항식 연산블록으로 구성한다.

특히 AB^2+C 연산의 경우 동일한 기약삼항식을 사용하는 조건에서는 승산연산에서 사용된 블록들을 반복 사용함으로써 구현이 되는 정규성을 보였고 기약다항식 연산블록의 경우에는 다양한 기약 삼항식을 사용하기 위하여 멀티플렉서 형태의 회로 구성으로 제안하였다. 그리고 가산연산을 위한 MSB(Modulo sum Block)블록은 SOP 연산 블록에 추가로 구성이 가능하다.

각각의 블록은 동작에 대한 특징을 설명하기 위하여 계수 순환이동 블록은 CSB(Cyclic Shift Block), SOP 연산 블록은 SOP(Sum Of Product), 기약다항식 연산블록은 ITB(Irreducible Trinomial Block)로 명명하였으며, 이들을 사용한 승산 및 AB^2+C 연산에 대한 병렬 승산기의 구조는 다음 그림 4와 같다. 상황에 따라서 가산연산의 경우는 입력 신호를 가산하지 않고 설계하면 AB^2 연산기로 구성할 수 있다. 또한 아래의 그림 4에서 CSB0 와 ITB0의 경우는 결선만으로 구성된 단순한 패스회로로 구성된다.

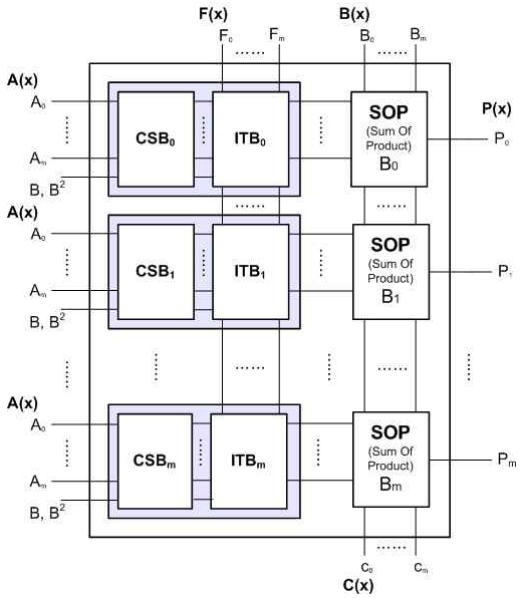

그림 4. 제안된 병렬 승산기의 구성도
Fig. 4. Architecture of proposed Multiplier

그림 4에서 도시된 CSB는 회로 결선만으로 수행되어 별도의 게이트 및 지연시간은 없으며 SOP 블록은 승산항과 피승산항의 계수들 간의 부분곱과 XOR 게이트로 구성이 되며 연산에 필요한 회로구성은 2-입력 AND 게이트 m개와 XOR 게이트 m개로 구성된다. m개의 SOP 블록으로 구성되면 회로 복잡도는 $2m^2$과 같고, 2-입력 AND 게이트와 2입력 XOR 게이트에서 발생하는 지연시간은 $T_A + T_X$ 이다. ITB 블록은

2-입력 XOR 게이트로 구성되며 승산항의 계수마다 하나의 블록이 소요되어 전체 회로 복잡도는 m이며 발생하는 지연시간은 T_X이다. 가산 블록은 각 SOP 블록내에 ITB 블록으로부터 연산된 AB^2의 결과들 중 동일 차수의 계수들과 C에 대한 모듈로 가산으로 구성된다. 따라서 m개의 승산항 계수마다 m+1개의 2-입력 XOR 게이트가 필요하여 회로 복잡도는 $m(m+1)$이고 지연시간은 $(1+\lceil \log_2 m \rceil)T_x$이다.

위의 회로 구성에 대한 내용을 종합하면, 본 논문에서 제안한 $GF(2^m)$상의 병렬 승산기 및 AB^2+C 연산회로는 $2m^2$개의 2-입력 AND 게이트와 $m(m+2)$개의 2-입력 XOR 게이트가 사용되며, 입력 신호에서부터 출력으로 신호가 전달될 때까지의 최종 전파 지연시간은 $T_A+(2+\lceil \log_2 m \rceil)T_x$이다. 이는 기약삼항식에 의한 전파지연을 고려한 결과이다.

V. 비교 및 검토

본 논문에서는 $GF(2^m)$상의 새로운 승산 및 AB^2+C 연산기법 및 그에 대한 연산회로를 제안하였고, 본 논문과 기존의 타 논문과의 회로구성에 필요한 구성요소를 기준으로 각 항목별 비교와 고찰을 하였다. 그 결과를 다음의 표 1에 정리하였다.

1) 연산회로구성 방법

파이프라인 연산으로 고속 대용량연산에 적합하나 메모리 소자와의 동기신호의 필요에 의해서 회로가 복잡해지는 시스톨릭 구조의 단점에 비해서 복잡도가 개선된 비 시스톨릭 구조를 갖는다. 따라서 회로구성방식에서 시스템복잡도 개선이 가능한 장점이 있다.

2) 기약다항식

유한체 연산에 적용되는 기약다항식의 경우 wang[11]의 논문은 일반 기약다항식에 대하여 다루었으며, Lee[12]와 Byun[13]의 논문에서는 AOP(All-One-Polynomial)로 제한하였고, 본 논문에서는 기약 삼항식으로 제한하였다.

3) 적용기저

비교를 위해서 정리한 표 1에서 와 같이 Lee[12]와 Byun[13]의 논문에서는 각 계수에 ⊕1을 취한 확장기저를 사용하였고, wang[11]의 논문과 본논문은 별도의 연산이 필요하지 않는 표준기저를 사용하였다.

4) 회로복잡도

wang은 표준기저를 사용하여 하나의 단위 셀에 6개의 AND 게이트와 6개의 XOR 게이트와 17개의 1-bit latch가 사용되는 단방향 데이터 흐름에 대한 AB^2+C 연산기 회로를 제안하였고 Lee는 기약 AOP를 기반으로 하는 AB^2+C 연산기 회로를 하나의 단위 셀에 각각 1개씩의 AND와 XOR 게이트가 사용되고 3개의 1-bit latch로 구성하여 제시하였다. Byun은 래치회로가 없는 AOP 기반의 AB^2+C 연산기 회로를 제안하였으며, 고속 대용량의 연산에 적합하나, AOP를 사용하는 회로의 특성상 기약다항식이 고정이 되는 단점이 있다.

본 논문에서는 타 논문과의 비교를 위해서 회로구성을 2입력 게이트로 구성하였고, 표 1에서와 같이 $2m^2$개의 2-입력 AND 게이트와 $m(m+2)$개의 2-입력 XOR 게이트를 사용하여 회로를 구성하였다.

5) 연산지연시간

본 논문에서는 Byun의 논문에서와 같이 메모리 소자를 사용하지 않으므로 전파지연 시간과 동기시간 제어를 위한 Latency가 필요하지 않다. 따라서 단순히 게이트 소자에 의한 지연으로 최종 전파 지연시간은 $T_A+(2+\lceil \log_2 m \rceil)T_x$이고 이는 고속의 연산에 적합한 회로구성임을 보인다.

표 1. $GF(2^m)$상의 AB^2+C 연산기의 구성 비교표
Table 1. Comparisions of the multipliers for AB^2+C over $GF(2^m)$

Multipliers		wang[11]	Lee[12]	Byun[13]	Proposed
Architecture		systolic	systolic	non-systolic	non-systolic
Polynomials		General Polynomial	Irreducible AOP	Irreducible AOP	Irreducible Trinomial
Basis		Standard	Extended Standard	Extended Standard	Standard
circuit complexity	No. of 2-input AND	3m2	(m+1)2	(m+1)2	2m2
	No. of 2-input XOR	2m2	(m+1)2	(m+1)(m+2)	m(m+2)
	1-bit-latch	(17/2)m2	3(m+1)2	-	-
Latency		(5/2)m	m+1	-	-
Propagation delay		TA+3TX+TL	TA+TX+TL	$T_A+(1+\lceil \log_2 m \rceil)T_x$	$T_A+(2+\lceil \log_2 m \rceil)T_x$
Note		TA, TX, TL are the propagation delay of a 2-input AND gate, 2-input XOR gate and 1-bit latch, respectively.			

VI. 결 론

본 논문에서는 유한체 승산연산에 필요한 유한체 상의 다항식들을 정규기저로 표현하였고 모듈로 연산을 위한 기약다항식은 삼항식이 갖는 특징을 활용하여 승산연산의 순환이동 성질과 모듈로 연산에 필요한 논리합 게이트를 단위셀로 하는 순환 및 모듈로 연산을 위한 블록을 제안하였으며, 제안된 셀

을 중복 사용하여 거듭제곱을 위한 회로구성에 적용이 가능함을 보였다.

본 논문에서 제안한 연산기를 회로의 구성 소자에 따른 복잡도와 지연시간 등으로 기존의 제안된 연산기와 비교하였으며, 그 결과 시스템 복잡도의 개선에 적합하며 기약다항식이 고정되는 구조가 아닌 기약삼항식을 사용하는 구조로 구성이 됨을 확인하였다. 또한 게이트만으로 구성되는 간단한 구조와 게이트를 연결하는 동일한 배선구조, 승산항의 차수가 증가함에 따른 동일한 셀의 연속배열 및 m의 증가에 따른 각 연산모듈의 규칙적인 증가가 갖는 정규성은 대규모 직접회로 구현에 유리함을 알 수 있다.

향후 연구로는 기약 삼항식의 일반적 형태인 $x^m + x^n + 1$을 이용한 승산기설계가 있으며, n을 정하여 행렬식으로 수식을 전개하는 과정을 고려한 승산기의 설계에 대한 연구가 진행될 경우 적은 회로 복잡도를 갖는 효율적인 승산기의 설계가 가능할 것으로 사료된다.

참 고 문 헌

[1] H.M. Shao, T.K.Truong, L.J. Deutsch, J.H. Yaeh and I.S. Reed, " A VLSI design of a pipeling reed-solomon decoder," IEEE Trans. Comput., vol. C-34, pp. 393-403, May 1985.

[2] K.C Smith. "The prospect for multivalued logic : A technology and applications view." IEEE Trans. Comput., vol. C-30, pp. 619-634, Sept. 1981.

[3] I.S. Hsu, T.K. Troun, "A Comparison of VLSI Architecture of Multipliers using Dual, Normal or Standard Bases," IEEE Trans. Comput., vol. C-37, pp. 735-739, 1988.

[4] H. Okano and H. Imai, "A Construction method of high-speed decoders using ROM's for Bose Chaudhuri Hocquenghem and Reed Solomon codes," IEEE Trans. Comput. vol. C-36, pp. 1165-1171, 1987.

[5] C.S. Yeh, I.S Reed and T.K. Truong, "Systolic multipliers for finite field GF(2m)," IEEE Trans. Comput., vol. C-33, pp. 357-360, Apr. 1984.

[6] P.A. Scott, S.E. Tarvares and L.E. Peppard, "A fast multiplier for GF(2m)," IEEE J.Select. Areas Commun., vol. SAC-4, Jan. 1986.

[7] C.C. Wang, T.K. Truong, H.M. Shao, L.J. Deutsch, J.K. Omura and I.S. Reed, "VLSI architecture for computing multiplications and inverses in GF(2m,)," IEEE Trans. Comput.,. vol. C-34, pp. 709-717, Aug. 1985.

[8] E.D. Mastrovito, "VLSI Architectures for Computation in Galois Fields," PhD thesis, Linkoping Univ., Dept. of Electrical Eng., Linkoping, Sweden, 1991.

[9] B. Sunar and C. K. Koc, "Mastrovito Multiplier for All Trinomials," IEEE Trans. Computers, vol. 48, no. 5, pp.522-527, May 1999.

[10] S.W. Wei, "A Systolic power-sum circuit for GF(2m)," IEEE Trans. Comput., vol. 43, pp. 226-229, Feb. 1994.

[11] C. L. Wang and J. H. Guo, "New systolic arrays for C+AB2, inversion, and division in GF(2m)," IEEE Trans. Comput., vol. 49, pp. 1120-1125, Oct. 2000.

[12] C. Y. Lee, E. H. Lu, and L. F. Sun, "Low-Complexity Bit-Parallel Systolic Architecture for Computing AB2+C in a class of Finite Field GF(2m)," IEEE Trans. Circuit & Systems-II : Analog and Digital Signal Processing, vol. 48, no. 5, pp. 519-523, May 2001.

[13] Gi-Young Byun and Heung-Soo Kim, "Low System Complexity Bit-Parallel Architecture for Computing AB2+C in a Class of Finite Fields GF(2m)," Journal of The Institute of Electronics Engineers, Vol. 40, No. 6, pp.378-384, Nov. 2003.

[14] K.M. Ku, K.J. Ha and K.Y.Yoo, "Design of new AB2 multiplier over GF(2m) using cellular automata," IEE Proc.-Circuits Devices Syst., Vol. 151, No. 2, April 2004.

저 자 소 개

한성일

1996: 인하대학교 공학사.
1998: 인하대학교 공학석사.
2004: 인하대학교 공학박사.
1998 – 2000: 대우통신 광통신연구실 선임연구원
2004 – 현재: 인덕대학 정보통신과 교수
관심분야 : 컴퓨터 구조, 네트워크 이론, 디지털 통신, 회로 설계
E-mail : hansi@induk.ac.kr