(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년12월21일
(11) 등록번호 10-1575247
(24) 등록일자 2015년12월01일

(51) 국제특허분류(Int. Cl.)
G06F 7/72 (2006.01) *G06F 11/10* (2006.01)
(52) CPC특허분류
G06F 7/724 (2013.01)
G06F 11/10 (2013.01)
(21) 출원번호 10-2015-0027850
(22) 출원일자 2015년02월27일
심사청구일자 2015년02월27일
(56) 선행기술조사문헌
KR1020000056557 A
KR1020100110546 A

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
홍석희
서울특별시 노원구 덕릉로71길 30, 102동 1701호 (중계동, 양지대림아파트)
김창한
서울특별시 강남구 삼성로 212, 21동 507호 (대치동, 은마아파트)
(뒷면에 계속)
(74) 대리인
특허법인충현

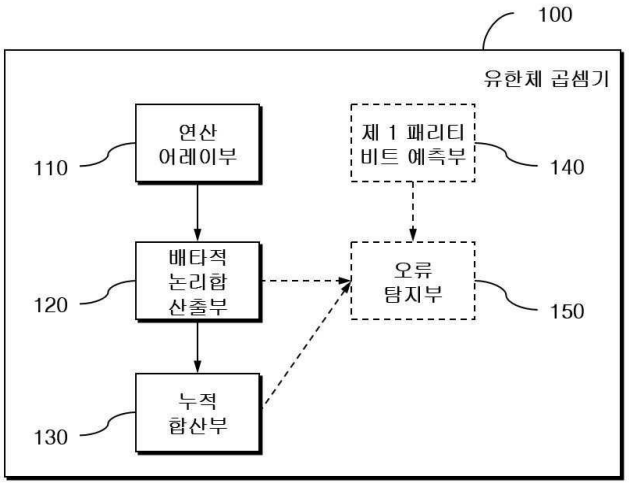
전체 청구항 수 : 총 5 항 심사관 : 지정훈

(54) 발명의 명칭 **타입Ⅱ최적 정규기저를 갖는 GF(2^n)의 곱셈기와 오류탐지**

(57) 요 약

본 발명은 유한체 곱셈기에 관한 것으로서, 제 1 입력과 제 2 입력에 대한 결과값의 각 자리수 값을 산출하는 연산 어레이부, 상기 각 자리수 값으로부터 선택적으로 배타적 논리합을 수행하여 단위별 상수값을 산출하는 배타적 논리합 산출부, 및 상기 배타적 논리합 산출부의 결과를 누적합산하여 유한체 곱셈의 결과값을 산출하는 누적합산부를 포함하고, 상기 단위별 상수값은 상기 유한체 곱셈의 결과값을 형성하는 각 단위별 상수값이고, 상기 유한체 곱셈기는 세미시스톨릭 구조로 형성되는 것을 특징으로 함으로써, 곱셈기에 대한 오류주입공격에 안전할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
G06F 7/726 (2013.01)
(72) 발명자
이승준
서울특별시 성북구 안암로9가길 35, 301호 (안암동5가)
장남수
서울특별시 마포구 월드컵북로27길 16 (성산동)
조성민
서울특별시 성북구 월곡로14길 26, 14동 902호

이 발명을 지원한 국가연구개발사업
과제고유번호 NRF-2014M3C4A7030648
부처명 미래창조과학부
연구관리전문기관 한국연구재단
연구사업명 차세대정보·컴퓨팅기술개발
연구과제명 암호 소프트웨어 안전성 연구
기 여 율 1/1
주관기관 고려대학교 산학협력단
연구기간 2014.07.01 ~ 2015.06.30

명 세 서

청구범위

청구항 1

유한체 곱셈기에 있어서,

제 1 입력과 제 2 입력에 대한 결과값의 각 자리수 값을 산출하는 연산 어레이부;

상기 각 자리수 값으로부터 선택적으로 배타적 논리합을 수행하여 단위별 상수값을 산출하는 배타적 논리합 산출부; 및

상기 배타적 논리합 산출부의 결과를 누적합산하여 유한체 곱셈의 결과값을 산출하는 누적합산부를 포함하고,

상기 단위별 상수값은 상기 유한체 곱셈의 결과값을 형성하는 각 단위별 상수값이고,

상기 유한체 곱셈기는 세미시스톨릭 구조로 형성되는 것을 특징으로 하는 유한체 곱셈기.

청구항 2

제 1 항에 있어서,

상기 연산 어레이부는,

논리곱 게이트와 배타적 논리합 게이트로 형성되는 것을 특징으로 하는 유한체 곱셈기.

청구항 3

제 1 항에 있어서,

상기 단위별 상수값은,

상기 배타적 논리합 산출부에서 각 단위별로 순차적으로 산출되는 것을 특징으로 하는 유한체 곱셈기.

청구항 4

제 1 항에 있어서,

상기 제 1 입력의 비트 중 하나의 비트와 상기 제 2 입력의 비트 중 하나의 비트를 이용하여 제 1 패리티 비트를 예측하는 제 1 패리티 비트 예측부; 및

상기 유한체 곱셈의 결과값의 각 비트 값과 상기 제 1 패리티 비트를 비교하여 오류가 있는지를 탐지하는 오류 탐지부를 더 포함하는 유한체 곱셈기.

청구항 5

제 4 항에 있어서,

상기 오류 탐지부는,

상기 연산 어레이부와 상기 누적합산부로부터 각 자리수의 비트 값을 수신하는 것을 특징으로 하는 유한체 곱셈기.

발명의 설명

기 술 분 야

[0001] 본 발명은 유한체 곱셈기에 대한 오류 탐지에 관한 것으로서, 더욱 상세하게는 타입Ⅱ최적 정규기저를 갖는 유한체 연산을 기반으로 하는 Semi-Systolic 구조의 곱셈기와 그것의 오류주입공격에 안전한 유한체 곱셈 연산의 오류 탐지에 관한 것이다.

배 경 기 술

[0002] 유한체 GF(2^n)의 연산은 덧셈과 곱셈, 제곱으로 구성되어 있다. 덧셈과 제곱은 간단하게 구성되는 반면 곱셈은 유한체의 표현 방법에 따라 다양하게 구성된다. 유한체의 표현은 어떤 기저를 사용하여 표현하느냐에 따라 구분하는데, 대표적으로 다항식기저와 정규기저를 사용하고 있다. 즉, GF(2^n)을 구성하는 GF(2) 위의 기약다항식을 $f(x)=x^n+a_{n-1}x^{n-1}+\cdots+a_1x+1$ 이라 할 때, f(x)의 근을 α라 하면, GF(2^n)의 원소 $A=a_0+a_1\alpha+\cdots+a_{n-1}\alpha^{n-1},\ a_i\in GF(2)$ 와 같이 표현하는 것이 다항식기저를 이용한 표현이다. 이때 A를 벡터로 표시하면 $A=(a_0,a_1,\cdots,a_{n-1})$ 이다. 반면에 정규기저를 이용한 표현은 $\beta\in GF(2^n)$의 conjugate들이 GF(2)위에서 GF(2^n)의 기저가 될 때, 이 기저를 사용하는 것이다. 즉, $\{\beta,\beta^2,\cdots,\beta^{2^{n-1}}\}$ 이 기저가 되어 GF(2^n)의 원소 $A=a_0\beta+a_1\beta^2+\cdots+a_{n-1}\beta^{2^{n-1}},\ a_i\in GF(2)$ 와 같이 표현하는 것이다. 이때 β를 정규기저 생성자라 한다. 또한 A를 벡터로 표시하면 $A=(a_0,a_1,\cdots,a_{n-1})$ 이다.

[0003] 정규기저를 이용할 경우 제곱 연산은 계수들의 위치이동으로 표시되므로 하드웨어적으로는 연산이 없는 장점이 있다. 정규기저를 구성하는 방법은 여러 방법이 있으나 다음 [정리 1]에 제시된 Gaussian 정규기저 방법이 가장 널리 쓰인다.

[0004] [정리 1] n,k는 nk+1이 2 보다 큰 소수인 조건을 만족하는 양의 정수이고, α는 GF(2^{nk})에서 nk+1의 원시근이라 하자. Z_{nk+1} 에서 2의 위수(order)를 e라 할 때 $\gcd(nk/e,n)=1$ 라 하자. 그러면 Z_{nk+1}에서 k의 원시근 τ에 대하여 $\beta=\sum_{i=0}^{k-1}\alpha^{\tau^i}$ 는 GF(2) 위에서 GF(2^n)의 정규기저 생성자이다.

[0005] [참고] 양의 정수 n에 대하여 정리 1을 만족하는 k가 존재할 때 GF(2^n)은 GF(2) 위에서 타입 k인 가우시안 정규기저를 갖는다고 한다.

[0006] 정규기저를 이용한 곱셈의 경우

[0007] $$C=A\cdot B,\ A=\sum_{i=0}^{n-1}a_i\beta^{2^i},\ B=\sum_{i=0}^{n-1}b_i\beta^{2^i},\ C=\sum_{i=0}^{n-1}c_i\beta^{2^i}$$

[0008] 라 하면

[0009] $$c_0=(a_0,a_1,\cdots,a_{n-1})\cdot M\cdot(b_0,b_1,\cdots,b_{n-1})^T,\quad M=(m_{ij}):n\times n$$

[0010] 로 표시되고, 이때 행렬 M의 1의 개수(Weight)를 기저 $N=\{\beta,\beta^2,\cdots,\beta^{2^{n-1}}\}$ 의 곱셈에 대한 복잡도(Complexity) C_N이라 한다.

[0011] 모든 정규기저 N의 복잡도는 $C_N\geq 2n-1$ 이다. 특히 $C_N=2n-1$ 인 경우 N을 최적정규기저라 한다. 위 정리에서 k=1,2인 경우 최적정규기저가 되며 각각을 타입 I, II의 최적정규기저라 한다. 위 정리에서 k가 짝수인 경우 $C_N\leq kn-1$ 이다.

[0012] 본 발명과 관련된 선행기술로는 '타입 케이 가우시안 정규기저를 갖는 유한체의 직렬 곱셈연산기(한국공개특허 10-2007-0102912)' 등이 있다.

발명의 내용

해결하려는 과제

[0013] 본 발명이 해결하고자 하는 과제는 타입Ⅱ최적 정규기저를 갖는 유한체 연산을 기반으로 하는 Semi-Systolic 구조의 곱셈기를 제공하는 것이다.

과제의 해결 수단

[0014] 본 발명은 상기 첫 번째 과제를 달성하기 위하여, 유한체 곱셈기에 있어서, 제 1 입력과 제 2 입력에 대한 결과값의 각 자리수 값을 산출하는 연산 어레이부; 상기 각 자리수 값으로부터 선택적으로 배타적 논리합을 수행하여 단위별 상수값을 산출하는 배타적 논리합 산출부; 및 상기 배타적 논리합 산출부의 결과를 누적합산하여 유한체 곱셈의 결과값을 산출하는 누적합산부를 포함하고, 상기 단위별 상수값은 상기 유한체 곱셈의 결과값을 형성하는 각 단위별 상수값이고, 상기 유한체 곱셈기는 세미시스톨릭 구조로 형성되는 것을 특징으로 하는 유한체 곱셈기를 제공한다.

[0015] 본 발명의 실시예에 의하면, 상기 연산 어레이부는, 논리곱 게이트와 배타적 논리합 게이트로 형성되는 것을 특징으로 하는 유한체 곱셈기일 수 있다.

[0016] 본 발명의 실시예에 의하면, 상기 단위별 상수값은, 상기 배타적 논리합 산출부에서 각 단위별로 순차적으로 산출되는 것을 특징으로 하는 유한체 곰셈기일 수 있다.

[0017] 본 발명의 실시예에 의하면, 상기 제 1 입력의 비트 중 하나의 비트와 상기 제 2 입력의 비트 중 하나의 비트를 이용하여 제 1 패리티 비트를 예측하는 제 1 패리티 비트 예측부; 및 상기 유한체 곱셈의 결과값의 각 비트 값과 상기 제 1 패리티 비트를 비교하여 오류가 있는지를 탐지하는 오류 탐지부를 더 포함하는 유한체 곱셈기일 수 있다.

[0018] 본 발명의 실시예에 의하면, 상기 오류 탐지부는, 상기 연산 어레이부와 상기 누적합산부로부터 각 자리수의 비트 값을 수신하는 것을 특징으로 하는 유한체 곱셈기일 수 있다.

발명의 효과

[0019] 본 발명에 의하면, 타입 Ⅱ 곱셈기의 오류 탐지는 U Array에서 한번의 연산 오류의 경우 1비트에 영향을 미친다. 그러므로 U Arrary 오류 탐지에 의해 탐지된다. 또한 Network-XOR과 Accumulator의 한번의 연산 오류 역시 1비트에 영향을 줌으로 C의 오류 탐지에 의해 탐지된다. 전체적으로 탐지용 연산에서도 오류가 발생하지 않고 Latch나 선에서 오류가 발생하지 않는 조건하에서 연산에서의 한번의 오류는 100%탐지된다. 또한 Chiou의 곱셈기 대비 공간 복잡도는 47% 감소, 시간복잡도는 4번의 클럭 추가 곱셈기 성능을 보이며, 오류 탐지 방법에서도에서 chiou에 비하여 공간 복잡도는 47.1% 시간복잡도는 1.71% 증가하였다. 또한 오류 탐지를 위한 추가 부담(XOR+AND) 복잡도는 0.32% 시간복잡도는 2.53% 증가를 보였다.

도면의 간단한 설명

[0020] 도 1은 본 발명의 일 실시예에 따른 유한체 곱셈기의 블록도이다.

도 2는 본 발명의 실시예에 따른 유한체 곱셈기이다.

도 3은 본 발명의 실시예에 따른 유한체 곱셈기의 배타적 논리합 산출부의 동작을 도시한 것이다.

도 4는 본 발명의 실시예에 따른 유한체 곱셈기의 누적합산부를 도시한 것이다.

도 5은 본 발명의 실시예에 따른 유한체 곱셈기를 나타낸 것이다.

도 6 내지 7은 본 발명의 실시예에 따른 오류탐지부를 포함하는 유한체 곱셈기를 나타낸 것이다.

발명을 실시하기 위한 구체적인 내용

[0021] 본 발명에 관한 구체적인 내용의 설명에 앞서 이해의 편의를 위해 본 발명이 해결하고자 하는 과제의 해결 방안의 개요 혹은 기술적 사상의 핵심을 우선 제시한다.

[0022] 본 발명의 일 실시예에 따른 유한체 곱셈기는 제 1 입력과 제 2 입력에 대한 결과값의 각 자리수 값을 산출하는 연산 어레이부, 상기 각 자리수 값으로부터 선택적으로 배타적 논리합을 수행하여 단위별 상수값을 산출하는 배타적 논리합 산출부, 및 상기 배타적 논리합 산출부의 결과를 누적합산하여 유한체 곱셈의 결과값을 산출하는

누적합산부를 포함하고, 상기 단위별 상수값은 상기 유한체 곱셈의 결과값을 형성하는 각 단위별 상수값이고, 상기 유한체 곱셈기는 세미시스톨릭 구조로 형성되는 것을 특징으로 한다.

[0023] 이하 첨부된 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명을 용이하게 실시할 수 있는 실시 예를 상세히 설명한다. 그러나 이들 실시예는 본 발명을 보다 구체적으로 설명하기 위한 것으로, 본 발명의 범위가 이에 의하여 제한되지 않는다는 것은 당업계의 통상의 지식을 가진 자에게 자명할 것이다.

[0024] 본 발명이 해결하고자 하는 과제의 해결 방안을 명확하게 하기 위한 발명의 구성을 본 발명의 바람직한 실시예에 근거하여 첨부 도면을 참조하여 상세히 설명하되, 도면의 구성요소들에 참조번호를 부여함에 있어서 동일 구성요소에 대해서는 비록 다른 도면상에 있더라도 동일 참조번호를 부여하였으며 당해 도면에 대한 설명시 필요한 경우 다른 도면의 구성요소를 인용할 수 있음을 미리 밝혀둔다. 아울러 본 발명의 바람직한 실시 예에 대한 동작 원리를 상세하게 설명함에 있어 본 발명과 관련된 공지 기능 혹은 구성에 대한 구체적인 설명 그리고 그 이외의 제반 사항이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우, 그 상세한 설명을 생략한다.

[0025] 도 1은 본 발명의 일 실시예에 따른 유한체 곱셈기의 블록도이다.

[0026] 본 발명의 일 실시예에 따른 유한체 곱셈기(100)는 연산 어레이부(110), 배타적 논리합 산출부(120), 및 누적합산부(130)를 포함하고, 제 1패리티 비트 예측부(140) 및 오류 탐지부(150)를 더 포함할 수 있다.

[0027] 제안한 타입 II 최적 정규기저 곱셈기

[0028] 타입 II인 경우 Z_{2n+1}에서 $\tau=-1$이므로 $\beta=\alpha+\alpha^{-1}$ 이다. 그리고 보조정리 1에 의하여 $Z^{*}_{2n+1}=\{\pm 2^{i} \mid 0\le i\le n-1\}$ 이므로 타입 I의 경우와 같이 $GF(2^n)$의 정규기저는 집합적으로 다음과 같다.

[0029]
$$N=\{\beta,\ \beta^{2},\ \beta^{2^{2}},\ \cdots,\ \beta^{2^{n-1}}\}$$
$$=\{\alpha+\alpha^{-1},\ \alpha^{2}+\alpha^{-2},\ \cdots,\ \alpha^{n}+\alpha^{-n}\}$$

[0030] 따라서 $GF(2^n)$의 원소 A의 계수 위치를 다시 정리하면

[0031]
$$A=\sum_{i=1}^{n}a_i(\alpha^{i}+\alpha^{-i})$$

[0032] 이다. 또한, A를 $GF(2^n)$의 원소 $\overline{A}$로 표시하면 다음과 같다.

[0033]
$$\overline{A}=\sum_{k=1}^{2n}A_k\alpha^{k},\ 1\le i\le n,\quad A_i=A_{2n+1-i}=a_i$$

[0034] 즉

[0035]
$$\overline{A}=(a_1,a_2,\cdots,a_n,a_n,a_{n-1},\cdots,a_1)$$
$$=\sum_{i=1}^{n}a_i\alpha^{i}+(\sum_{i=1}^{n}a_{n+1-i}\alpha^{i})\alpha^{n}$$
$$=A1+A2\alpha^{n},\ A1=\sum_{i=1}^{n}a_i\alpha^{i},\ A2=\sum_{i=1}^{n}a_{n+1-i}\alpha^{i}$$

[0036] 가 된다.

[0037] $C = A \cdot B,\ A,B \in GF(2^n)$ 을 구하기 위하여

[0038]
$$A \equiv \overline{A} = A1 + A2\alpha^n$$
$$B \equiv \overline{B} = B1 + B2\alpha^n$$

[0039] 라 하자.

[0040]
$$C \equiv \overline{C} = (A1 + A2\alpha^n)(B1 + B2\alpha^n)$$

[0041] 위식을 다시 정리하면

[0042]
$$C \equiv \overline{C} = A1 \cdot B1 + A1 \cdot B2\alpha^n + A2 \cdot B1\alpha^n + A2 \cdot B2\alpha^{2n}$$

[0043] 이다.

[0044] 한편 $A1 \cdot B1 = \sum_{i=1}^{n} a_i B1\alpha^i = \sum_{i=1}^{n} a_i B1\alpha^{(i)}, B^{(i)} = B1\alpha^i$ 라 하면

[0045]
$$B^{(i)} = \sum_{j=1}^{n} b_j \alpha^{j+i}$$
$$B^{(i+1)} = B^{(i)}\alpha = \sum_{j=1}^{n} b_j \alpha^{j+i+1}$$

[0046] 와 같다.

[0047] $A1 \cdot B1$ 을 세미시스톨릭 곱셈기로 나타 낼 수 있다.

[0048] A1B2와 A2B1의 경우 AjBj계산후 α^n 을 곱하여 출력한다. 즉 $A1B2 = \sum_{i=2}^{2n} D_i$ 라 하면,

[0049]
$$A1B2\alpha^n = (D_{n+1} + D_{n+2})\alpha + (D_{n+1} + D_{n+3})\alpha^2 + \cdots + (D_{n+1} + D_{2n})\alpha^{n-1} + D_{n+1}\alpha^n + D_{n+1}\alpha^{n+1} + (D_{n+1}D_2)\alpha^{n+2} + \cdots + (D_{n+1} + D_n)\alpha^{2n}$$

[0050] 이므로, $(D_{n+1} + D_{n+2})\alpha + (D_{n+1} + D_{n+3})\alpha^2 + \cdots + (D_{n+1}D_{2n})\alpha^{n-1} + D_{n+1}\alpha^n$ 의 과정이 필요하다.

[0051] A2B2의 경우 $A2B2 = \sum_{i=2}^{2n} D_i\alpha^i$ 이고, A1B1의 경우 C_1부터 C_n까지 그대로이다.

[0052] 상기 과정을 정리하면 A와 B의 유한체 곱셈의 결과인 C는 A1B1, A1B2, A2B1, 및 A2B2 네개의 값으로 구성됨을 알 수 있다.

[0053] 연산 어레이부(110)는 제 1 입력과 제 2 입력에 대한 결과값의 각 자리수 값을 산출한다.

[0054] 보다 구체적으로, 연산 어레이부(110)는 제 1 입력과 제 2 입력에 대한 결과값의 각 자리수 값을 산출한다. 도 2와 같이 연산 어레이부는 C_1부터 C_{2n}까지의 각 자리수 값을 산출한다.

[0055] 연산 어레이부(110)는 논리곱 게이트와 배타적 논리합 게이트로 형성될 수 있다. 연산 어레이부를 구성하는 각 셀들은 제 1 입력의 자리수 값과 제 2 입력의 자리수의 값의 논리곱을 구하고, 상기 논리곱을 입력 C의 값과 배타적 논리합을 산출하여 C의 결과값으로 산출한다. 도 2와 같이, a_{in}, b_{in}, c_{in}을 입력받아 a_{in}과 b_{in}의 논리곱과 c_{in}의 배타적 논리합인 c_{out}을 출력한다. a_{in}과 b_{in}은 a_{out}과 b_{out}으로 출력된다.

[0056] 배타적 논리합 산출부(120)는 상기 각 자리수 값으로부터 선택적으로 배타적 논리합을 수행하여 단위별 상수값을 산출한다.

[0057] 보다 구체적으로, 연산 어레이부(110)가 산출한 각 자리수 값 중 결과값을 산출하기 위한 자리수 값을 선택하여 배타적 논리합을 수행하여 단위별 상수값을 산출한다. 상기 단위별 상수값은 상기 유한체 곱셈의 결과값을 형성하는 각 단위별 상수값으로, 상기 A1B1, A1B2, A2B1, 및 A2B2에 해당하는 값이다. 도 3과 같이, 배타적 논리합 산출부(120)는 연산 어레이부(110)의 결과들 중 필요한 값들만을 선택하여 배타적 논리합을 수행한다.

[0058] 상기 단위별 상수값은 배타적 논리합 산출부(120)에서 각 단위별로 순차적으로 산출된다. 효율성을 높이기 위하여, 상기 각 단위별 상수값은 파이프라인과 같이 순차적으로 산출되도록 구현된다.

[0059] 누적합산부(130)는 배타적 논리합 산출부(120)의 결과를 누적합산하여 유한체 곱셈의 결과값을 산출한다.

[0060] 보다 구체적으로, 입력되는 배타적 논리합 산출부(120)의 결과를 누적합산한다. 도 4와 같이, 그 결과를 이전 결과와 배타적 논리합을 레지스터에 저장하면서 누적합산한다.

[0061] 상기와 같이, 유한체 곱셈기(100)는 세미시스톨릭 구조로 구현된다.

[0062] 전체적인 연산 구성에 대해 도 2를 참조하여 구체적으로 설명하도록 한다. 연산 어레이부(110)는 U array이고, 배타적 논리합 산출부(120)는 Network-XOR이고, 누적합산부(130)는 n-Bit Accumulator이다. A1B2를 도 2의 U Array가 계산한 후, A2B1이 U Array 계산을 마치는 순간 A1B2는 Network-XOR을 계산 한 후 결과값을 도 4의 Accumulator의 Register C로 저장한다. 이와 동시에 A2B2가 U Array 계산을 마쳤을 경우, A1B2는 도 4의 Accumulator의 Register D로 저장한다. 그 후 A2B1은 Network-XOR 후 Accumulator 계산후 Register D로 이동, A2B1은 Network-XOR 후 Accumulator C로 이동, 그 후 A1B1이 도 2U Arrary 계산을 마친 경우는 A2B2는 Network-XOR 계산후, Accumulator C로 이동, A1B2와 A2B1은 XOR 후 D로 이동, 이후 XOR 한번으로 A2B2와 $(A1B2 \oplus A2B1)$의 XOR 후 D로 이동, A1B1은 Network-XOR 후 C로 이동, 마지막으로 한번의 XOR에 의하여 AB의 값을 출력한다.

[0063] 표 1은 다른 곱셈기와 본 곱셈기 발명과의 복잡도 비교이다.

표 1

Multipliers	Kwon[1]	Chiou [9]	Proposed Multiplier
Array Type	Systolic	Semi-Systolic	Semi-Systolic
Function	AB+C	AB+C	AB+C
Number of Cells	n^2	$n(2n+1)$	n^2
Space Complexity			
2-Input AND	$2n^2+n$	$2n^2+n$	n^2
2-Input XOR		$2n^2+5n+1$	n^2+2n
3-Input XOR	n^2+n		
1-bit Latch or Flip-Flop	$5n^2+2n-2$	$9n(n+1)/2+2$	$5n(n+1)/2$
Total transistor counts	$64n^2+34n-16$	$60n^2+56n+22$	$32n(n+1)$
Time Complexity			
Cell Delay	$T_A+T_{XB}+T_L$	$T_A+T_X+T_L$	$T_A+T_X+T_L$
Latency	$n+1$	$n+1$	$n+5$
Total Delay	$(n+1)(T_A+T_X+T_L)$	$(n+1)(T_A+T_X+T_L)$	$(n+5)(T_A+T_X+T_L)$

[0064]

[0065] 도 5는 $GF(2^3)$의 예시이다.

[0066] $GF(2^3)$의 경우,

[0067] $A1=a_1\alpha+a_2\alpha^2+a_3\alpha^3,\quad B1=b_1\alpha+b_2\alpha^2+b_3\alpha^3$ 이므로,

[0068] $$A1\cdot B1=a_1(b_1\alpha^2+b_2\alpha^3+b_3\alpha^4)+a_2(b_1\alpha^3+b_2\alpha^4+b_3\alpha^5)+a_3(b_1\alpha^4+b_2\alpha^5+b_3\alpha^6)$$

[0069] $= d_2\alpha^2 + d_3\alpha^3 + d_4\alpha^4 + d_5\alpha^5 + d_6\alpha^6$ 라 하면,

[0070] $A1B2\alpha^3 = (d_4 + d_5)\alpha + (d_4 + d_6)\alpha^2 + d_4\alpha^3 + (d_4 + d_2)\alpha^5 + (d_4 + d_3)\alpha^6$

[0071] $A1B2\alpha^3 = (d_4 + d_5)\alpha + (d_4 + d_6)\alpha^2 + d_4\alpha^3 + (d_4 + d_2)\alpha^5 + (d_4 + d_3)\alpha^6$

[0072] $A2B2 = e_2\alpha^2 + e_3\alpha^3 + e_4\alpha^4 + e_5\alpha^5 + e_6\alpha^6$ 라 하면,

[0073] $A2B2\alpha^6 = e_2\alpha + e_3\alpha^2 + e_4\alpha^3 + e_5\alpha^4 + e_6\alpha^5$ 이다.

[0074] 곱셈기에 오류가 입력되는지를 판단하기 위하여, 제 1 패리티 비트 예측부(140)와 오류 탐지부(150)를 더 포함할 수 있다.

[0075] 제 1 패리티 비트 예측부(140)는 상기 제 1 입력의 비트 중 하나의 비트와 상기 제 2 입력의 비트 중 하나의 비트를 이용하여 제 1 패리티 비트를 예측한다.

[0076] 오류 탐지부(150)는 상기 유한체 곱셈의 결과값의 각 비트 값과 상기 제 1 패리티 비트를 비교하여 오류가 있는지를 탐지한다. 연산 어레이부(110)와 누적합산부(130)로부터 각 자리수의 비트 값을 수신하여 오류 탐지에 이용할 수 있다.

[0077] 이하, 도 6과 7을 참조하여 오류 탐지하는 과정을 자세히 설명하도록 한다.

[0078] $A = (a_1, a_2, \cdots a_n), B = (b_1, b_2, \cdots b_n), C = (c_1, c_2, \cdots c_n) \in GF(2^n)$ 에 대하여 $C = A \cdot B,\ A, B, C \in GF(2^n)$ 의 각각 parity bit를 $A_p = \sum_{i=1}^{n} a_i,\ B_P = \sum_{i=1}^{n} b_i,\ C_P = \sum_{i=1}^{n} c_i, \in GF(2)$ 라 하고, A와 B의 곱셈 연산에서 얻은 C의 parity bit를 C_p(Act), C_p(Pred)라 한다면 오류 없이 정상적인 곱셈이 이루어진다면 C_p(Act) = C_p(Pred)이다.

[0079] 2n+1은 소수 α는 2n+1의 원시근((2n+1)th primitive root of unity) 라 하자 그러면

[0080] $\widetilde{A} = \sum_{i=0}^{2n} \widetilde{A_i}\alpha^j, \widetilde{B} = \sum_{j=0}^{2n} \widetilde{B_j}\alpha^j, \quad \widetilde{A_i}, \widetilde{B_j} \in GF(2)$ 라 하면,

[0081] $\widetilde{A}, \widetilde{B} = GF\left(2^{2n}\right)$ 이다.

[0082] 그리고 $\widetilde{C} = \widetilde{A} \cdot \widetilde{B} = \sum_{k=0}^{2n} \widetilde{C_k}\alpha^k$ 라 하면

[0083] $$(\widetilde{C_0}\ \widetilde{C_1}\ \cdots\ \widetilde{C_{2n}})^T = \begin{pmatrix} \widetilde{A_0} & \widetilde{A_{2n}} & \cdots & \widetilde{A_2} & \widetilde{A_1} \\ \widetilde{A_1} & \widetilde{A_0} & \cdots & \widetilde{A_3} & \widetilde{A_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{A_{2n}} & \widetilde{A_{2n-1}} & \cdots & \widetilde{A_1} & \widetilde{A_0} \end{pmatrix} \begin{pmatrix} \widetilde{B_0} \\ \widetilde{B_1} \\ \vdots \\ \widetilde{B_{2n}} \end{pmatrix}$$ 이다

[0084] 유한체의 원소 A 기저로 표현되었을 경우와 같이 parity bit 를 $\widetilde{A}$ 에서도 다음과 같이 정의하자

[0085] $\widetilde{A} \in GF(2^{2n})$ 인 경우, $\widetilde{A_P} = \sum_{i=0}^{2n} \widetilde{A_i} \in GF(2)$ 라 하자, 그러면

[0086] $\widetilde{C_P} = \widetilde{A_P} \cdot \widetilde{B_P}$ 이다.

[0087] 유한체 GF(2^n)가 타입 II 최적 정규기저를 갖는 경우 앞에서 보았듯이

[0088] $N = \{\beta, \beta^2, \beta^{2^2}, \cdots, \beta^{2^{n-1}}\} = \{\alpha+\alpha^{-1}, \alpha^2+\alpha^{-2}, \cdots, \alpha^n+\alpha^{-n}\}$ 이다

[0089] 따라서, GF(2^n)의 원소 A의 계수 위치를 다시 정리하면

[0090] $$A = \sum_{i=1}^{n} a_i(\alpha^i + \alpha^{-i})$$ 이다.

[0091] 또한 A를 GF(2^{2n}) 의 원소 $\bar{A}, \tilde{A}$ 로 다시 표시하면 다음과 같다.

[0092] $\bar{A} = \sum_{k=1}^{2n} A_k\alpha^k$, $\tilde{A} = \sum_{k=0}^{2n} \tilde{A}_k\alpha^k$, $1 \le i \le n$ 에 대해 $A_i = A_{2n+1-i} = a_i$ 이고, $\tilde{A}_i = \widetilde{A_{2n+1-i}} = a_i$, $\tilde{A}_0 = 0$ 이다. 그러므로 정수로 보면, $\bar{A}_P = \tilde{A}_P = 2A_P$ 이다. 앞으로 $A, \bar{A}, \tilde{A} \in GF(2^n)$ 를 같은 원소로 보자.

[0093] $1 = \alpha+\alpha^2+\cdots+\alpha^{2n}$ 이므로

[0094] $\tilde{C} = \sum_{i=0}^{2n} \tilde{C}_i\alpha^i = \sum_{i=1}^{2n} (\tilde{C}_0+\tilde{C}_i)\alpha^i \in GF(2^n)$ 인 경우, $\tilde{C} \equiv \bar{C} \equiv C$ 로 보면

[0095] $\sum_{i=1}^{2n} (\tilde{C}_0+\tilde{C}_i) = 2C_P$ 이다.

[0096] $A, B \in GF(2^n)$ 를 $\tilde{A}, \tilde{B}$ 로 표시하고 A와 B의 곱을 표시한 결과를 $\tilde{C} = \sum_{i=0}^{2n} \tilde{C}_i\alpha^i$ 라 하면

[0097]

[0098] $$(\tilde{C}_0\ \tilde{C}_1\ \cdots\ \tilde{C}_{2n})^T = \begin{pmatrix} 0 & \widetilde{A_{2n}} & \cdots & \tilde{A}_2 & \tilde{A}_1 \\ \tilde{A}_1 & 0 & \cdots & \tilde{A}_3 & \tilde{A}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{A_{2n}} & \widetilde{A_{2n-1}} & \cdots & \tilde{A}_1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \tilde{B}_1 \\ \vdots \\ \widetilde{B_{2n}} \end{pmatrix}$$ 이다. 그리고 $\bar{C} = \bar{A} \cdot \bar{B} = \sum_{i=1}^{2n} (\tilde{C}_0+\tilde{C}_i)\alpha^i$ 이다.

[0099] Parity값을 정수로 보면

[0100] $\bar{A}_P = 2A_P$, $\bar{B}_P = 2B_P$, $\tilde{C}_0 = \sum_{i=1}^{2n} \widetilde{A_{2n+1-i}}\bar{B}_i = 2\sum_{i=1}^{n} a_ib_i$ 이므로

[0101] $$2C_P = \sum_{i=1}^{2n} (\tilde{C}_0+\tilde{C}_i) = \sum_{i=0}^{n} \tilde{C}_i + \sum_{i=2}^{n} \tilde{C}_0 = \tilde{C}_P + (2n-1)\tilde{C}_0$$
$$= \bar{A}_P\bar{B}_P + (2n-1)\tilde{C}_0 = 2A_P2B_P + 2(2n-1)\sum_{i=1}^{n} a_ib_i.$$

[0102] 그러므로 $A, B \in GF(2^n)$ 에 대하여 $C = A \cdot B$ 라 하면,

[0103] $$C_P(pred) = \sum_{i=1}^{n} a_ib_i$$ 이다.

[0104] U Array에서 계산하는 C1=A1B1은 $\sum_{1 \le i,j \le n} a_i b_j \alpha^{i+1}$ 이므로 $C1 = A1_p B1_p = A_p B_p$ 이다. 이때 오류탐지는 U Array가 끝난 후 $C1 = A1B1,\ C2 = A1B2,\ C3 = A2B1,\ C4 = A2B2$ 를 계산하여 확인하고 AB의 계산이 끝난 후 $C_P(pred) = \sum_{i=1}^{n} a_i b_i$ 와 C_p(Act)를 비교하여 수행한다. 탐지 과정에서 $Ck_p = Ai_p Bj_p$ 확인에 1개의 AND 게이트와 2n-1개의 Xor 게이트가 필요하다. $C_P(pred) = \sum_{i=1}^{n} a_i b_i$ 의 계산은 U Array n클럭 타임에 Serial 방법으로 계산하면 1개의 AND 게이트, 1개의 XOR게이트, 1개의 1-bit 레지스터가 필요하다.

[0105] C_p(Act), C_p(Pred)의 비교에는 n개의 XOR 게이트가 필요하다. 그리고 Time Delay는 의 U Array 연산 오류 탐지를 위한 과정은 $Ai_p Bj_p + C_2 + \cdots C_{2n}$ 을 계산하는 과정에서 UArray 계산중에 $A_p B_p + C_2 + \cdots + C_n$ 을 계산하고, 그결과와 $C_{n+1}, \cdots, C_{2n}$ 과의 XOR는 BTX에서 계산한다. 즉, BTX 의 Delay는 $\lceil \lg(n+1) \rceil T_X$ 이다. 그러나 n이 홀수 이므로 $\lceil \lg n \rceil T_X$ 이다.

[0106] BTX에서 $2T_x$를 1클럭으로 고려하면 n+1개의 XOR은 $2^{2k-2} < n+1 \le 2^{2k}$ 인 경우 $k = \lceil \log_4 n \rceil$ 클럭 만에 계산 할 수 있다. 그러면 A1B2, A2B1, A2B2, A1B1의 결과를 차례대로 검증하면 $\lceil \log_4 n \rceil + 3$ 클럭이 필요하나 C의 계산에 5클럭이 추가로 필요한바 최종적으로는 $\lceil \log_4 n \rceil - 2$ 클럭이 필요하고 C_p(Act)계산과 C_p(Pred)와의 비교에 $\lceil \log_4 n \rceil$ 클럭이 추가로 필요하다.

[0107] 전체적으로 $2\lceil \log_4 n \rceil - 2$ 클럭이 추가로 소요된다. n=233인 경우 k=4 이므로 6클럭이 더 필요하며, 오류 탐지를 위하여 $A_p B_p + C_2 + \cdots + C_n$ 계산에 연산당 Latch 1개가 필요함으로 n개의 Latch가 필요하고 BTX 또한 $l = \lceil (n+1)/2^2 \rceil + \lceil (n+1)/2^4 \rceil + \cdots + \lceil (n+1)/2^{2k-2} \rceil + 1$ 개의 Latch가 필요하다. 결과적으로 전체적으로 l+1개의 Latch가 필요하다.

[0108] 오류 탐지는 U Array에서 한번의 연산 오류의 경우 1비트에 영향을 미친다. 그러므로 U Arrary 오류 탐지에 의해 탐지된다. 또한 Network-XOR 과 Accumulator의 한번의 연산 오류 역시 1비트에 영향을 줌으로 C의 오류 탐지에 의해 탐지된다. 전체적으로 탐지용 연산에서도 오류가 발생하지 않고 Latch나 선에서 오류가 발생하지 않는 조건하에서 연산에서의 한번의 오류는 100%탐지된다.

[0109] 시스톨릭 오류 탐지의 결과는 다음 표 2와 같다.

표 2

	Complexities=original+[overhead]	Overhead^*(%)	Coverage
AND XOR 1-bit Latch Total	$n^2+[2]$	0.004	100%(**) ??(***)
	$n^2+n+[2n]$	0.85	
	$5n(n+1)/2+[n+l]$	0.23	
	$9n(n+1)/2+[3n+l+2]$	0.32	
Delay	$(n+5)(T_A+T_X+T_L)+$ $[(2\lceil \log_4(n+1) \rceil -2)(T_A+T_X+T_L)$	2.53	

* : n=233, (overhead/original)×100%, ** 한번의 연산오류, *** 모든오류

[0110] $l=\lceil (n+1)/2^2 \rceil+\lceil (n+1)/2^4 \rceil+\cdots+\lceil (n+1)/2^{2k-2} \rceil+1,\ 2^{2k-2}<n+1\le 2^{2k}$

[0111] 본 발명의 실시예들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0112] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

[0113] 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

부호의 설명

[0114] 100: 유한체 곱셈기

110: 연산 어레이부

120: 배타적 논리합 산출부

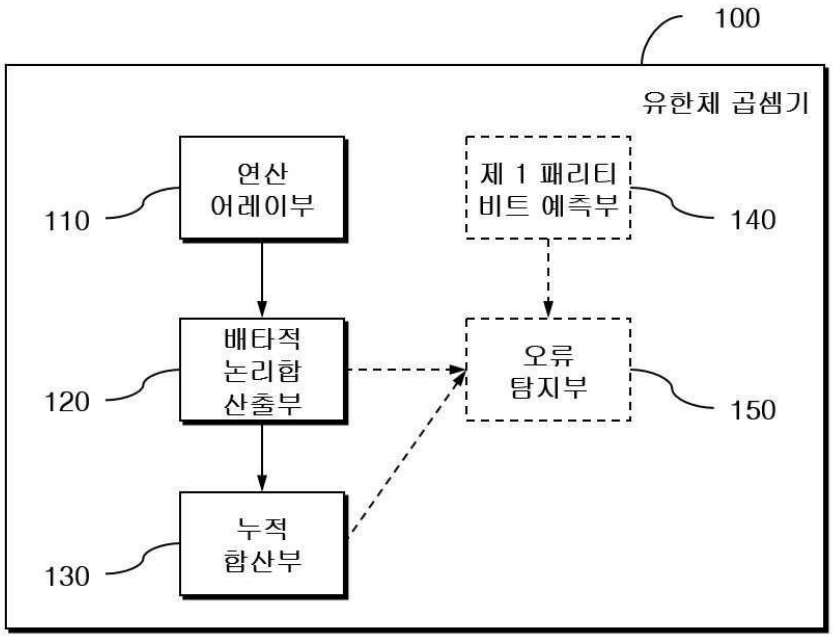
130: 누적 합산부

140: 제 1패리티 예측부

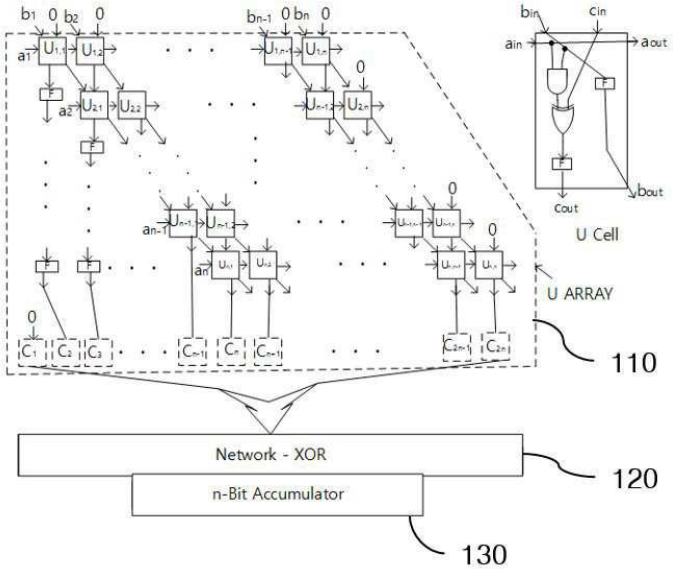
150: 오류 탐지부

도면

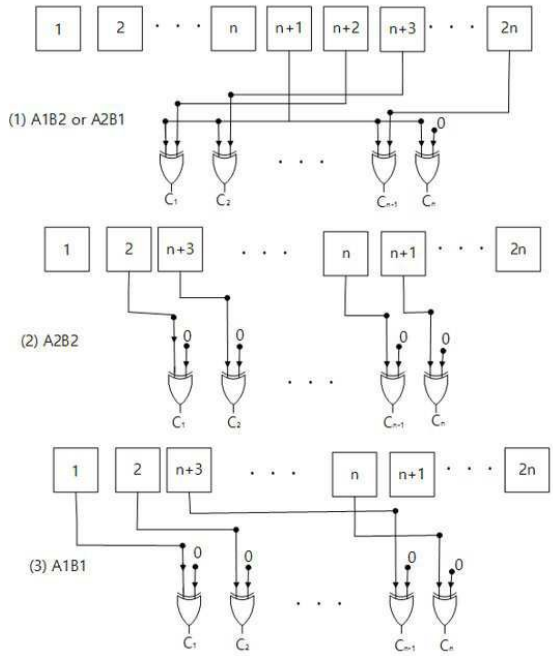
도면1


100
유한체 곱셈기
연산
어레이부
110
제 1 패리티
비트 예측부
140
배타적
논리합
산출부
120
오류
탐지부
150
누적
합산부
130

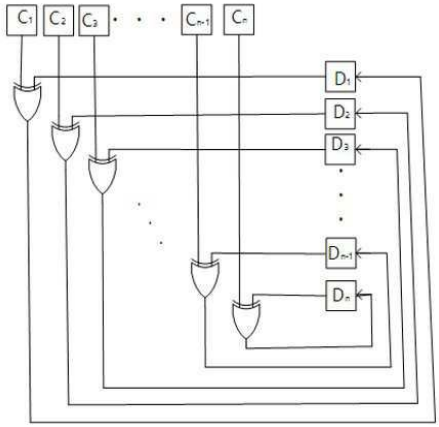
도면2


U Cell
U ARRAY
110
Network - XOR
120
n-Bit Accumulator
130

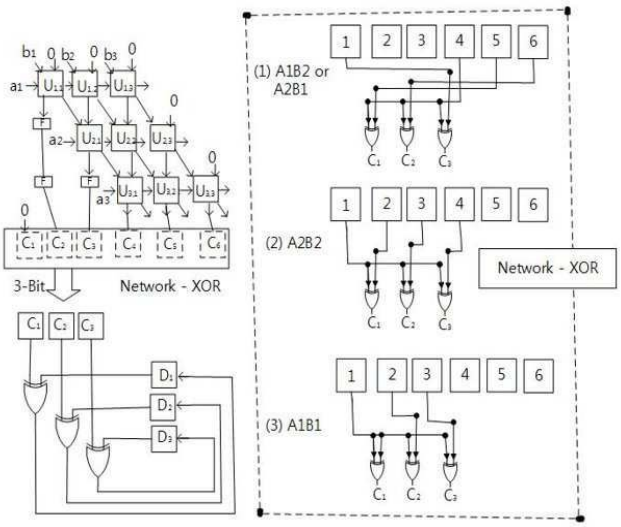
도면3


1
2
n
n+1
n+2
n+3
2n
(1) A1B2 or A2B1
(2) A2B2
(3) A1B1
0

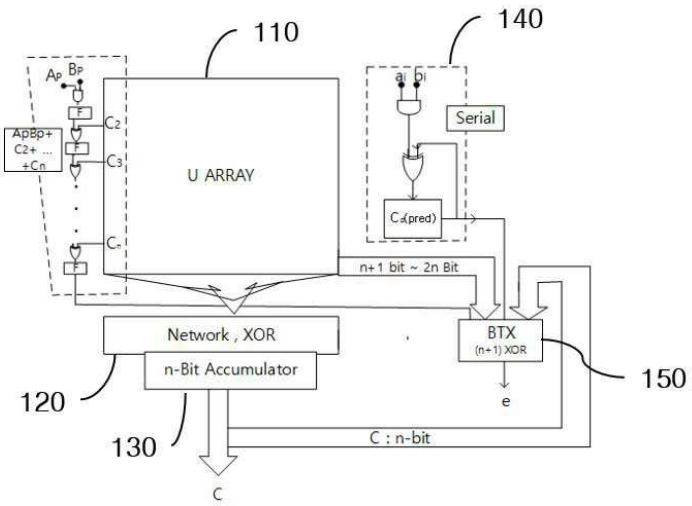
도면4


C1
C2
C3
Cn-1
Cn
D1
D2
D3
Dn-1
Dn

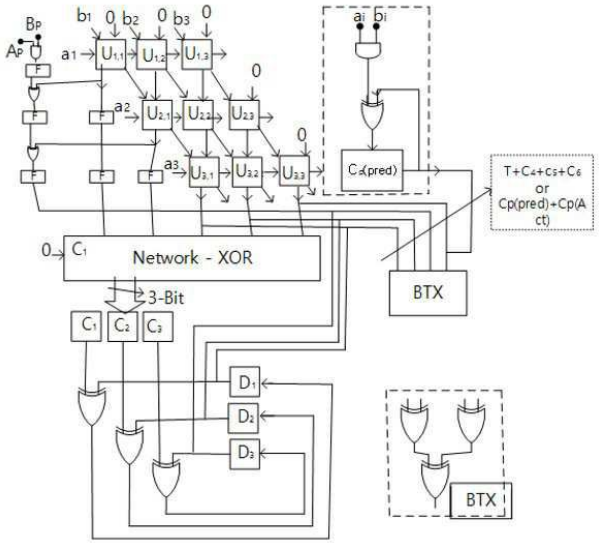
도면5


(1) A1B2 or A2B1
(2) A2B2
(3) A1B1
Network - XOR
3-Bit
Network - XOR

도면6


110
140
U ARRAY
Serial
n+1 bit ~ 2n Bit
Network , XOR
n-Bit Accumulator
BTX
(n+1) XOR
120
130
150
C : n-bit
C
e

도면7



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 제3항

【변경전】

유한체 곱셈기

【변경후】

유한체 곱셈기