(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
G06F 7/52 (2006.01)

(45) 공고일자 2006년06월22일
(11) 등록번호 10-0591761
(24) 등록일자 2006년06월13일

(21) 출원번호 10-2004-0002393
(22) 출원일자 2004년01월13일

(65) 공개번호 10-2004-0092376
(43) 공개일자 2004년11월03일

(30) 우선권주장 1020030026482 2003년04월25일 대한민국(KR)

(73) 특허권자 삼성전자주식회사
경기도 수원시 영통구 매탄동 416

(72) 발명자 윤중철
서울특별시강북구수유2동372-9

손희관
경기도수원시팔달구망포동동수원엘지빌리지106동1101호

(74) 대리인 임창현
권혁수
오세준
송윤호

심사관 : 이정호

(54) 몽고메리 모듈러 곱셈기 및 캐리 저장 가산을 이용한몽고메리 모듈러 곱셈 방법

요약

몽고메리 모듈러스 곱셈 모듈의 모듈러스 곱셈 동작에서의 전력 소비를 감소시키고, 계산 속도를 증가시키는 방법이 제공된다. 코딩 스킴은 멀티플 모듈러스 값들을 얻기위한 가산기 및 메모리 소자의 필요성을 감소시키고, 캐리 저장 가산 및 캐리 전파 가산의 사용은 곱셈 모듈의 계산 속도를 향상시킨다. 특히, 본 발명은 정수 모듈러 연산 뿐만 아니라 다항식 모듈러 연산을 수행할 수 있다.

대표도

도 2

색인어

공개키, 모듈러 연산, 캐리 저장 가산, 캐리 전파 가산, 몽고메리 곱셈

명세서

도면의 간단한 설명

도 1은 모듈러스 선택기, 부스 레코더 및 누산기를 이용하여 구현된 종래의 몽고메리 모듈러 곱셈 알고리즘의 하드웨어 구성을 보여주는 도면;

도 2는 본 발명의 바람직한 실시예에 따른 모듈러 곱셈기의 전체적인 구성을 보여주는 도면;

도 3은 본 발명의 바람직한 실시예에 따른 모듈러스 레코더의 코딩 스킴을 보여주는 도면;

도 4는 모듈러스 레코더의 구체적인 구성 예를 보여주는 도면;

도 5는 본 발명의 바람직한 실시예에 따른 부스 레코더의 코딩 스킴을 보여주는 도면;

도 6은 부스 레코더의 구체적인 구성 예를 보여주는 도면;

도 7은 CSA를 사용한 본 발명의 실시예를 보여주는 도면;

도 8은 완전 컴프레서(201)의 일 예;

도 9는 축소된 컴프레서(203)의 일 예;

도 10은 도 9에 도시된 축소된 컴프레서들을 포함하는 누산기(171a)를 상세히 보여주는 도면;

도 11은 CPA를 사용한 본 발명의 실시예;

도 12는 본 발명의 실시예에 따른 k번째 비트 멀티플렉서 그룹;

도 13은 도 11에 도시된 멀티플렉서 그룹이 도 10에 도시된 누산기에 포함된 상세 회로도;

도 14는 본 발명의 다른 실시예에 따른 곱셈기;

도 15는 도 14에 도시된 부스 레코더의 코딩 스킴;

도 16은 도 15의 코딩 스킴에 따라서 부스 레코더를 구현한 회로의 실시예;

도 17은 정수 모듈러 연산과 다항식 모듈러 연산을 선택적으로 수행할 수 있는 본 발명의 바람직한 실시예에 따른 곱셈기;

도 18은 $b_{I}A$를 구하기 위한 본 발명의 바람직한 실시예에 따른 분해값 선택기의 코딩 스킴을 보여주는 도면;

도 19는 본 발명의 바람직한 실시예에 따른 모듈러스 선택기의 코딩 스킴을 보여주는 도면;

도 20은 도 17에 도시된 누산기의 상세한 회로 구성; 그리고

도 21은 본 발명의 실시예에 따른 k번째 비트 멀티플렉서 그룹들을 보여주고 있다

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 데이타 보안을 위한 암호화(cryptography) 시스템에 관한 것으로서, 특히 암호화시스템에서의 모듈러 곱셈연산을 위한 장치 및 방법에 관한 것이다.

타원 곡선 암호법(Elliptic Curves CryptoSystem, ECC)은 정수 모듈러 연산을 바탕으로 한 GF(p) 연산과 다항식 모듈러 연산을 바탕으로 한 GF(2^m)의 연산으로 구성된다.

정수 모듈러 연산인 몽고메리 모듈러 곱셈 알고리즘은 다음 수학식 1과 같다.

수학식 1

$$R=A*B*r^{-1} \bmod N, \text{ (radix } r=2^n)$$

모듈러 지수 알고리즘(modular exponential algorithm)에서, A, B 및 N은 각각 승수, 피승수 및 모듈러 수(modular number)이고, 각각은 n 비트이다.

도 1은 모듈러스 선택기(1), 부스 레코더(12) 및 누산기(2)를 이용하여 구현된 종래의 몽고메리 모듈러 곱셈 알고리즘의 하드웨어 구성을 보여주고 있다. 다중 모듈러스 선택기(1)는 다중 모듈러스(0, M, 2M 및 3M) 중 하나의 값을 선택하고, 선택된 값을 캐리 전파 가산기(carry propagation adder, CPA)(14)으로 출력한다. 값 3M을 얻기 위해서는 부가적인 가산기가 요구되고, 하드웨어 크기의 증가와 계산 속도의 저하를 초래한다. 누산기(2)는 2 개의 CPA들(14, 11)을 포함한다. 각 CPA는 누산기의 전달 지연 시간을 증가시키고 계산 속도를 저하시킨다. CPA(11)는 피승수 선택기(13)로부터의 부분곱(partial product value)과 누산기(2)의 이전 출력 P[i]를 받아들인다. 피승수 선택기(13)는 부분곱(-2A, -A, 0, A, 2A)을 얻기 위하여 승수와 부스 레코더(Booth Recoder)(12)의 출력을 받아들인다. CPA(11)는 부분곱 및 P[i]를 더한다. 누산값 P{i+ 1]을 얻기 위해 CPA(11)의 출력은 CPA(14)로 입력된다. 그 결과, 몽고메리 곱셈 P[i+ 1]=$A*B*R^{-1}$ mod M을 얻을 수 있다.

다항식 모듈러 연산은 수학식 2와 같다.

수학식 2

$$P(x)=A(x)B(x) \bmod G(x)$$

A(x) 및 B(x)는 GF(2^m)의 원소이고, G(x)는 차수 n인 원시 다항식이다. A(x), B(x) 및 G(x)는 수학식 3과 같이 표현된다.

수학식 3

$$A(x)=a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0$$

$$B(x)=b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_1x + b_0$$

$$G(x)=g_{n-1}x^{n-1} + g_{n-2}x^{n-2} + \cdots + g_1x + g_0$$

일반적으로 정수 모듈러 연산을 위한 곱셈기와 다항식 모듈러 연산을 위한 곱셈기는 각각 별도로 구성된다. 정수 모듈러 연산 및 다항식 모듈러 연산을 모두 할 수 있는 곱셈기가 구현된다면 하드웨어 크기는 크게 감소될 수 있을 것이다.

발명이 이루고자 하는 기술적 과제

따라서, 본 발명의 목적은 하드웨어의 크기를 늘리지 않으면서 모듈러 연산속도를 향상시키는 곱셈 연산장치를 제공함에 있다.

본 발명의 다른 목적은 래딕스-4 논리연산을 기반으로 하여 몽고메리 곱셈 알고리즘의 하드웨어 구성 및 동작을 효율적으로 실현하는 곱셈 연산장치를 제공함에 있다.

발명의 구성 및 작용

본 발명의 실시예에서는, 래딕스-4 논리연산을 기반으로 하여 몽고메리 곱셈알고리즘을 적용하며, 이를 래딕스-4 인터리브 몽고메리 곱셈 알고리즘(Radix-4 Interleaved Montgomery Multiplication Algorithm)이라 하고 R4IMM로 약칭한다.

본 발명의 실시예에서 보이는 곱셈장치의 논리연산체계는 공개키 방식의 암호화알고리즘을 적용하는 컴퓨터 시스템 또는 통신망에 적용될 수 있으며, 또한 휴대가능한 집적회로 카드(또는 스마트카드)에 내장되어 운용될 수 있다.

본 발명에 따른 실시예는 적어도 1224비트 이상의 큰 정수를 기반으로 하는 모듈로연산에 적용될 수 있다.

본 발명의 실시예를 설명하기에 앞서, 본 발명에 적용되는 모듈러 곱셈 알고리즘 R4IMM에서 참조되는 패래미터들(parameters)을 다음과 같이 정의한다.

M은 모듈러스(mudulus)로서 2보다 큰 양의 정수(integer)로서 홀수값(예컨대, 3, 7 등)을 가진다.

M'는 조건식 $(-M*M') \bmod 4 = 1$을 만족하는 정수이다.

n은 모듈러스 M의 비트길이(bit length)를 나타내는 값으로서 양의 정수이다. 본 발명의 실시예에서는 8로 할 것이다.

N은 모듈러스 M의 디지트길이(digit length)를 나타내는 값으로서 양의 정수이다.

여기서, n은 N의 2배값(n = 2N)으로서, n이 8이라면 N은 4가 된다.

R은 $R = 2^n = 2^{2N}$의 조건을 만족하는 양의 정수이다. n이 8이라면 R은 256이 된다.

R^{-1}은, R의 모듈로 역곱수로서, 조건식 $(R*R^{-1}) \bmod M = 1$을 만족하는 양의 정수이다.

A는, 피승수(multiplicand)로서, $0 \le A < M$의 조건을 만족하는 정수이다.

B는, 승수(multiplier)로서, $0 \le B < M$의 조건을 만족하는 정수이다. 여기서, $B = \sum_{I=0}^{N-1} b_I A^I$, $b_I \in \{0, 1, 2, 3\}$이다. b_I는 2비트이다.

본 발명에 적용되는 R4IMM 알고리즘은 다음과 같다.

$S_0 := 0$

for I := 0 to (N-1)

$q_I := (((S_I + b_I A) \bmod 4)*M') \bmod 4$

$S_{I+1} := (S_I + b_I A + q_I M) / 4$

endfor

if $(S_N \ge M)$ $S_N := S_N - M$

R4IMM 알고리즘에서 I는 디지트 인덱스 또는 연산의 반복회수를 나타내며, 알고리즘을 수행한 총 곱의 결과는 $S_N = A*B*R^{-1} \pmod M$으로 나타난다. 총 곱 S_N의 범위는 $0 \le S_N < M$이다. 상기 알고리즘에서 몫 q_I는 '$S_I + b_I A + q_I M$의 하위 2비트가 "00"이 될 수 있도록 더해 줄 M의 갯수'를 나타낸다. 리던던트수 체계(residue number system; RNS)에서

어떤 수에 계수 M의 정수배를 더한 수는 원래의 수와 같으므로, 계수 M의 정수배인 모듈러스곱 q_IM(이하, MM_I; multiple of modulus)을 더한 수는 원래의 수와 같다. 또한, $S_I + b_IA + q_IM$의 하위 2비트를 "00"으로 만든 후에 래딕스값 4로 나누면(즉, 2비트씩 오른쪽으로 쉬프트함) 유효자리의 숫자는 그대로 보존되기 때문에 정보가 유실되지 않는다.

이와 같은 R4IMM 알고리즘을 하드웨어적으로 구현하기 위해서는, 부분곱 PP_I와 모듈러스곱 MM_I를 구해야 한다. 단위승수 b_I와 몫 q_I가 2비트이므로, 부분곱 PP_I와 모듈러스곱 MM_I는 다음과 같이 4가지 경우의 가능값들을 각각 가지도록 설정될 수 있을 것이다(여기서, $b_I \in \{0, 1, 2, 3\}$이고 $q_I \in \{0, 1, 2, 3\}$으로 함).

$b_IA = PP_I \in \{0, A, 2A, 3A\}$

$q_IM = MM_I \in \{0, M, 2M, 3M\}$ 식 1

그러나,위 [식 1]과 같이 부분곱 PP_I와 모듈러스곱 MM_I를 설정하게 되면, 값 3A와 3M를 계산할 때 A 또는 M을 1비트 쉬프트한 값과 원래 값의 합을 구해야 한다(2A+ A, 2M+ M). 이를 하드웨어로 구현하기 위해서는 이러한 값을 계산할 독립된 가산기(adder)를 사용하거나 또는 이 값들을 미리 계산하여 메모리 등에 저장해 놓고 필요할 때마다 참조하는 방법을 사용하여야 한다. 그러한 방법은 하드웨어의 낭비를 초래하고, 또한 그 값들(3A, 3M)을 구하는 시간(미리 계산해 두거나 실시간으로 계산함)을 추가적으로 고려하여 하드웨어들 설계하여야 하므로 성능 저하의 요인이 될 수 있다.

그리하여, 본 발명에서는, 부분곱 PP_I와 모듈러스곱 MM_I를 구함에 따른 하드웨어 부담을 줄이고자 한다.

도 2는 본 발명의 바람직한 실시예에 따른 모듈러 곱셈기의 전체적인 구성을 보여 준다.

도 2의 모듈러 곱셈기(1000)는 모듈러스 레지스터(M_REG, 100)에 저장된 모듈러스(M), 피승수 레지스터(A_REG, 101)에 저장된 피승수(A), 승수 레지스터(B_REG, 102)에 저장된 승수(B), 모듈러스 레코더(Modulus recoder, 110), 부스 레코더(Booth recoder, 140), 멀티플 모듈러스(multiple modulus, MM_I)의 계산을 돕는 멀티플렉서(120), 부분곱(partial product, PPI)의 계산을 돕는 멀티플렉서(130), 그리고 모듈러 곱셈 연산을 돕는 누산기(170)를 포함한다. 누산기(170)는 부분곱(PP_I), 멀티플 모듈러스(MM_I) 및 보상 워드 신호(CW)를 입력받아 몽고메리 곱셈 결과를 출력한다. 본 발명의 실시예에서, 모듈러스 M은 양수이고, n비트(M[n-1:0])이다. 피승수 A는 양수 또는 음수이며, n+ 1비트(A[n:0])이고, 1 비트의 부호 비트를 가지며, 승수 B는 짝수 비트들을 갖는다. n이 짝수이면, B는 n+ 2 비트이고, 부호 비트는 2 비트이다. 또는 n이 홀수 이면, B는 n+ 1비트이고, 부호 비트는 1 비트이다. 승수 B는 매 클럭 사이클마다 2비트씩 우측으로 쉬프트된다. 승수 B의 하위 2비트 b1 및 b0와 비트 b_R(비트 b_R은 이전 사이클의 비트 b1)는 부스 레코더(140)로 제공된다.

본 발명의 바람직한 실시예에 따른 레지스터(100)는 모듈러스 M과 모듈러스 M의 1의 보수값 $\underline{M}$을 제공한다. 유사하게 레지스터(101)는 피승수 A와 피승수 A의 1의 보수값 $\underline{A}$을 제공하고, 레지스터(102)는 승수(B)를 제공한다.

곱셈기(1000)는 반복적인 프로세스로 모듈러 곱셈의 해답을 찾는다. 모듈러스 레코더(110) 및 멀티플렉서(120)는 모듈러스(MM_I)를 선택하기 위해 사용된다. 모듈러스(MM_I)를 선택하기 위하여, 모듈러스 레코더(110)는 누산기(170)로부터 반복적인 데이터를 받아들인다. 본 발명의 실시예에서 반복적인 데이터 $SPP_I[1:0]$는, 누산기(170)에 저장된 합($S_I[1:0]$)과 캐리($C_I[1:0]$)의 최하위 2 비트, 부분곱($PP_I[1:0]$) 그리고 부분곱 반전 표시 신호(NEG_PP)에 근거한다. 합($S_I[1:0]$)과 캐리($C_I[1:0]$)는 2-비트 가산기(181)에서 결합되어 $S_I[1:0]$를 형성한다. 결합된 신호는 2-비트 가산기(182)에서 부분곱($PP_I[1:0]$)과 부분곱 반전 표시 신호(NEG_PP)와 결합되어 $SPP_I[1:0]$를 형성한다. 모듈러스 레코더(110)는 모듈러스의 두번째 최하위 비트 M[1]를 더 받아들인다. 모듈러스 레코더(110)는 $SPP_I[1:0]$ 및 모듈러스 M[1]를 이용하여 복수의 모듈러스 MM_I의 선택을 결정하기 위한 출력 신호를 발생한다. 본 발명의 실시예에 따른 상술한 설명에서 비트 크기는 한정되지 않는다. SPP_I는 2 비트 이상일 수 있으며 그에 따라 본 발명의 다른 구성들은 변경될 것이다.

모듈러스 레코더(110)는 복수의 신호들을 출력한다. 이 실시예에서, 모듈러스 레코더(110)는 2M, M, 0, $\underline{M}$ 가운데 하나를 선택하기 위해 멀티플렉서로 선택 신호(SEL_MM[1:0])를 출력한다. 멀티플렉서(120)는 모듈러스 M과 모듈러스 선택 신

호(SEL_MM[1:0])를 입력받고, MM_I를 출력한다. MM_I는 누산기(170)로 입력된다. 보상 워드 신호(CW)를 얻기 위해 모듈러스 반전 표시 신호(NEG_MM)는 반가산기(160)에서 부분곱 반전 표시 신호(NEG_PP)와 결합된다. 보상 워드 신호(CW)는 누산기(170)로 입력된다.

모듈러스 반전 표시 신호(NEG_MM)는 선택된 값$_I$이 비트 반전될 것인 지의 여부를 나타내기 위해 사용된다. 또한, 부분곱 반전 표시 신호(NEG_PP)는 선택된 부분곱(PP_I)이 비트 반전될 것인 지의 여부를 나타내기 위해 사용된다. 부분곱(PP_I)은 부스 레코더(140), 멀티플렉서(130) 및 AND 게이트(150)에 의해 수행되는 동작에 근거한다. 부분곱(PP_I)는 MM_I 및 CW와 함께 누산기(170)로 입력된다. 도 2에서 4:1 멀티플레서들이 도시되었으나 멀티플렉서의 특정비는 한정되지 않으며, 누산기는 5-2 컴프레서에 한정되지 않는다. 4:1 멀티플렉서는 2:1 멀티플렉서들 3 개로 대체될 수 있다.

도 3은 본 발명의 바람직한 실시예에 따른 모듈러스 레코더(110)의 코딩 스킴을 보여주고 있다. 도 3에는 모듈러스 레코더(110)로 입력되는 3 개의 입력 M[1] 및 SPP_I[1:0]을 보여주고 있으나 본 발명은 입력들 및 출력들을 다양하게 변경할 수 있다.

전형적으로 복수의 모듈러스 MM_I은 0, M, 2M 및 3M이다. 3M을 구하기 위해서는 1M에 2M을 더하기 위한 부가적인 가산기 또는 메모리 소자가 필요하다. 부가적인 가산기 및/또는 메모리 소자는 하드웨어 크기 및/또는 계산 지연 등을 초래하고 이는 계산 속도 및 전력 소모에 영향을 끼친다. 도 3의 코딩 스킴은 MM_I의 값을 얻기 위해 부가적인 가산기 또는 메모리 소자없이 비트 반전 및 비트 쉬프트를 이용한다. 모듈러스 레코더(110)는 모듈러스 M의 두번째 최하위 비트 M[1]과 SPP_I의 최하위 2비트를 입력받는다. 모듈러스 레코더(110)는 모듈러스 선택 신호(SEL_MM[1:0])를 출력한다. 선택된 모듈러스 값 MM_I는 누산기(250)로 전송된다. 본 발명의 상술한 설명에서 값들의 비트 크기는 한정되지 않는다. SPP_I는 2 비트 이상일 수 있으며, 그에 따라 본 발명의 다른 소자들은 변경될 수 있다.

도 4는 모듈러스 레코더(110)의 구체적인 구성 예를 보여주고 있다. 도 4에 도시된 예에서는, 모듈러스 레코더(110)가 인버터들(301-301)과 낸드 게이트들(311-317)로 구성되나, 모듈러스 레코더(110)의 회로 구현은 다양하게 변경될 수 있다.

감소된 하드웨어 크기 및 증가된 계산 속도 및 전력 감소를 위한 유사한 방법이 도 2 및 도 6에 도시된 바와 같은 부스 레코더(140)에 사용될 수 있다. 앞서 언급한 곱셈기(1000)는 모듈러스(MM_I) 및 부분곱(PP_I)을 누산기(170)로 입력해서 반복적인 프로세스로 모듈러 곱셈을 수행한다.

부스 레코더(140) 및 멀티플렉서(130)는 누산기(250)로 제공될 부분곱 PP_I의 값들 0, A, 2A, <u>A</u>, <u>2A</u>을 선택하기 위해 사용된다. 도 5는 본 발명의 바람직한 실시예에 따른 부스 레코더(140)의 코딩 스킴을 보여주고 있다. 부스 레코더(140)는 승수 B[1] 및 B[0]의 최하위 2비트와 반복값 B[1]의 이전 값인 B[R]을 입력받고, 부분곱 선택 신호(SEL_PP[1:0]), 부분곱 인에이블 신호(EN_PP) 및 부분곱 반전 표시 신호(NEG_PP)를 출력한다. 도 5에는 부스 레코더(140)로 입력되는 3 개의 입력 B[1], B[0] 및 B[R]을 보여주고 있으나 본 발명은 입력들 및 출력들을 다양하게 변경할 수 있다.

PP_I를 선택하기 위해, 부스 레코더(140)는 4 가지 값들 <u>2A</u>, <u>A</u>, A, 2A 중 하나를 선택하기 위한 부분곱 선택 신호(SEL_PP[1:0])를 멀티플렉서(130)로 출력한다. 도 6은 부스 레코더(140)의 구체적인 구성 예를 보여주고 있다. 도 6에 도시된 예에서는, 부스 레코더(140)가 인버터들(321-323)과 낸드 게이트들(331-340)로 구성되나, 부스 레코더(140)의 회로 구현은 다양하게 변경될 수 있다.

다시 도 2를 참조하면, 멀티플렉서(130)는 피승수 A의 값 및 신호(SEL_PP[1:0])를 받아들이고, 출력을 AND 게이트(150)로 출력한다. AND 게이트(150)는 멀티플렉서(130)로부터의 입력과 부스 레코더(140)로부터의 부분곱 인에이블 신호(EN_PP)를 받아들인다. AND 게이트(150)는 부분곱(PP_I)의 선택된 값을 누산기(170)로 출력한다. 부분곱 인에이블 신호(EN_PP)가 0일 때 AND 게이트(150)는 0인 PP_I를 누산기(250)로 출력한다. 부분곱 반전 신호(NEG_PP)는 반가산기(160)로 입력된다. 부분곱 반전 신호(NEG_PP)의 1의 값은 새로운 부분곱(PP_I)을 누산기(170)로 입력하기 위해서 <u>2A</u> 또는 <u>A</u> 중 하나를 구하기 위한 부분곱(PP_I)에 대한 비트 반전이 수행될 것임을 나타낸다.

부분곱(PP_I) 및 모듈러스(MM_I)의 덧셈에서, 보상 워드 신호(CW)는 반가산기(160)로부터 누산기(170)로 전달된다. 누산기(170)는 부분곱(PP_I), 모듈러스(MM_I) 및 보상 워드 신호(CW)를 컴프레스 네트워크(171)로 입력한다. 컴프레스 네트워크(171)는 CSA(Carry Save Adder) 및 CPA(Carry Propagate Adder)에서 사용된다. 앞서 설명한 도 1의 누산기는 CPA를 사용하여 전파 지연(propagation delay)으로 인해 동작 속도가 느렸다. CSA의 사용은 전파 지연을 감소시키고 계산 속도를 향상시키며 전력 소모를 감소시키므로써 동작 속도를 향상시킨다.

본 발명의 바람직한 실시예에서는 CSA 및 CPA를 혼합하여 계산 속도를 증가시키고, 전력 소모를 감소시킨다. 본 발명의 실시예에서 CPA는 마지막 반복에서 사용되고, CSA는 이전 반복에서 사용된다. 도 7 및 도 11은 CSA 및 CPA를 사용한 본 발명의 두 가지 실시예를 보여주고 있다.

본 발명의 바람직한 실시예에 따른 누산기(170a)가 도 7에 도시되어 있다. 누산기(170a)는 직렬로 연결된 n+ 2 개의 5-2 컴프레서들로 구성되며, 컴프레서들은 완전(full) 컴프레서들(예를 들면, 202)과 축소된(reduced) 컴프레서들(예를 들면, 203)로 나뉘어진다. 여기서, n은 모듈러스 값 M의 비트 길이이다. 누산기(170a)는 합(S)과 캐리(C)를 합 레지스터(S_REG, 173)와 캐리 레지스터(C_REG, 172)에 각각 저장한다. 합 레지스터(173) 및 캐리 레지스터(172)의 출력들은 캐리 전파 가산기(174)에 입력된다. 캐리 전파 가산기(174)는 리던던트 수(redundant number)를 정상 수(normal number)로 변환하고, 변환된 수를 최종 레지스터(175)에 저장한다.

이 실시예에서, 누산기(170a)로의 입력은 보상 워드 CW[1:0], 멀티플 모듈러스 값(MM_I) 및 부분곱(PP_I)이다. 처음 2 개의 완전 컴프레서들(201, 202)은 모듈러스(MM_I[1:0]) 및 부분곱(PP_I[1:0])과 함께 보상 워드(CW[1:0])를 입력받는다. 나머지 축소된 컴프레서들(203, 204, 205 등)은 멀티플 모듈러스의 나머지 비트들(MM_I[n+ 1:2]) 및 부분곱(PP_I[n+ 1:2])을 사용한다. 마지막 컴프레서(n+ 2 번째 컴프레서)(205)는 오버플로우를 방지하고, 첫번째 컴프레서는 세번째 전가산기가 없는 완전 컴프레서이다.

완전 컴프레서(201) 및 축소된 컴프레서(203)의 일 예가 도 8 및 도 9에 각각 도시되어 있다. 각 컴프레서는 현재 값(I) 및 다른 입력들을 이용하여 다음 값(I+ 1)을 얻기 위해 사용된다. 도 8은 본 발명의 바람직한 실시예에 따른 완전 컴프레서(201)를 보여주고 있다. 완전 컴프레서(201)는 복수의 입력들을 갖는다. 이 실시예에서, 완전 컴프레서(201)는 다섯 개의 입력들 즉, 1 비트 높은 컴프레서로부터의 다음 캐리 워드 비트값으로부터 얻어진 현재 캐리 워드 비트(C_I), 2 비트 높은 컴프레서로부터의 다음 합 워드 비트로부터 얻어진 합 워드 비트(S_I), 보상 워드(CW), 부분곱(PP_I) 및 멀티플 모듈러스(MM_I)를 입력받는다. 현재 컴프레서로 입력된 현재 캐리 워드 비트는 인덱스 "I"를 가지며, 나머지 고차 비트 컴프레서는 다음 캐리 워드 비트값(C_{I+1}[k+ 1])을 출력한다. k는 "k"번째 컴프레서를 나타낸다. 다음 캐리 워드 비트 값(C_{I+1}[k+ 1])은 캐리 레지스터(172)로 입력되고, 캐리 레지스터(172)는 현재 캐리 워드 비트값(C_I)을 k번째 컴프레서로 출력한다. 현재 합 워드 비트값(S_I[k])은 k+ 2 컴프레서로부터의 다음 합 워드 비트값(S_{I+1}[k+ 2])이 합 레지스터(173)로 입력되는 것에 의해서 구해진다. 상기 값들은 특정 비트 k, C_{I+1}[k] 및 S_{I+1}[k]에 대한 다음 캐리 워드 비트 및 다음 합 워드 비트 값을 구하기 위해 완전 컴프레서(201)에 의해서 사용된다. 이러한 값들은 각각 캐리 및 합 레지스터들(172, 173)을 통과한다. 앞서 설명한 바와 같이, 캐리 및 합 레지스터들(172, 173) 각각은 낮은 비트 컴프레서들로 입력들을 제공한다. 다음 캐리 워드 비트(C_{I+1}[k]) 및 다음 합 워드 비트(S_{I+1}[k])는 수학식 4와 같다.

수학식 4

$$(2C_{I+1}[k]+2CO1[k]+2CO2[k]+S_{I+1}[k])$$

$$=(C_I[k]+S_I[k])+PP_I[k]+MM_I[k]+CW[k]+CI1[k]+CI2[k]$$

단, k>1 이면, CW[k]는 입력되지 않는다.

본 발명의 실시예에서, 완전 컴프레서(201)는 3 개의 전가산기들을 포함한다. 제 1 전가산기(211)는 C_I, S_I 및 CW를 입력받고, 제 1 전가산기 캐리(FCO1) 및 제 1 전가산기 합(FSO1)을 출력한다. 제 1 전가산기 캐리(FCO1)는 제 1 출력 캐리(CO1)로서 출력되어서 다음 높은 비트 컴프레서(k+ 1)의 입력(CI1[k+ 1])이 된다. 제 2 전가산기(212)는 제 1 전가산기 합(FSO1), 부분곱의 비트값(PP_I[k]) 및 멀티플 모듈러스 비트값(MM_I[k])을 받아들이고, 제 2 전가산기 캐리(FCO2) 및

제 2 전가산기 합(FSO2)을 출력한다. 제 2 전가산기 캐리(FCO2)는 제 1 출력 캐리(CO2)로서 출력되어서 다음 높은 비트 컴프레서(k+ 1)의 입력(CI2[k+ 1])이 된다. 제 3 전가산기(213)는 제 2 전가산기 합(FSO2)과 낮은 비트 컴프레서(k-1)로부터의 CI1 및 CI2를 입력받고 제 3 전가산기 캐리(FCO3) 및 제 3 전가산기 합(FSO3)을 출력한다. 제 3 전가산기 캐리(FCO3)는 다음 캐리 워드 비트 값(C_{I+1})으로서 출력되어서 낮은 비트 컴프레서(k-1)로 입력되는 캐리 C_I로 사용된다. 제 3 전가산기 합(FS03)은 2 비트 낮은 컴프레서(k-2)로 입력되는 S_I로 사용된다. 비트 0에 대응하는 제 1 완전 컴프레서(201)는 다음 캐리 또는 합 워드들을 출력하지 않는다. 따라서 제 3 완전 컴프레서는 제 1 전가산기를 필요로하지 않는다. 반면, 비트 1에 대응하는 제 2 완전 컴프레서(202)는 다음 합 워드 비트를 출력하지 않는다.

보상 워드(CW[1:0])는 2 비트를 가지므로 각각의 비트에 대한 2 개의 컴프레서들이 필요하다. 2 개의 컴프레서들(201, 202)은 복수의 값들을 입력받는 완전 컴프레서들이다. 본 발명의 실시예에서 컴프레서들(201, 202)은 5 가지 값들을 입력받는다. 높은 비트 컴프레서들[2:n+ 2]은 컴프레서들(201, 202)에 비해 적은 수의 값들을 입력받는 축소된 컴프레서들이다. 도 9에 도시된 축소된 컴프레서들은 제 1 전가산기가 반가산기로 대체된다. 따라서, 축소된 컴프레서 내의 반가산기(211)는 캐리(C_I)와 합(S_I)을 입력받고 제 1 반가산기 캐리(HCO1) 및 제 1 반가산기 합(HSO1)을 출력한다. 제 1 반가산기 캐리(HCO1)는 제 1 출력 캐리(CO1)로서, 다음 높은 비트 컴프레서(k+ 1)의 두번째 제 1 입력(CI1[k+ 1])으로 제공된다. 제 2 전가산기(212)는 제 1 반가산기 합(HSO1), 컴프레서의 비트와 관련있는 부분곱 비트값(PP_I[k]) 및 멀티플 모듈러스 비트값(MM_I[k])을 입력받는다. 제 2 전가산기(212)는 제 2 전가산기 캐리(FCO2) 및 제2 전가산기 합(FSO2)을 출력한다. 제 2 전가산기 캐리(FCO2)는 제 2 출력 캐리(CO2)로서, 다음 높은 비트 컴프레서(k+ 1)의 제 2 입력(CI2[k+ 1]으로 제공된다. 제 3 전가산기(213)는 제 3 전가산기 캐리(FCO3) 및 제 3 전가산기 합(FSO3)을 출력한다. 제 3 전가산기 캐리(FCO3)는 다음 캐리 워드 비트(C_{I+1})로서, 캐리 레지스터(172)를 통과한 후 낮은 비트 컴프레서(k-1)의 입력(C_I)으로 제공된다. 제 3 전가산기 합(FSO3)은 다음 합 워드 비트(S_{I+1})로서, 합 레지스터(173)를 통과하여 두 비트 낮은 컴프레서(k-2)의 입력(S_I)으로 제공된다. 도 10은 도 9에 도시된 바와 같은 축소된 컴프레서들을 포함하는 누산기(171a)를 상세히 보여주고 있다.

본 발명의 실시예에 따른 도 10에 도시된 누산기(171a)는, 완전 컴프레서들과 축소된 컴프레서들이 직렬로 연결되고, 컴프레서들의 수는 멀티플 모듈러스 값(MM_I) 및 부분곱 값(PP_I)의 비트 크기에 따라 결정된다. 최하위 2 비트 컴프레서들은 보상 워드(CW)를 입력으로서 사용하는 완전 컴프레서들이다. 첫번째 비트 컴프레서(201)는 캐리들(CO1[0], CO2[0])을 출력한다. 캐리들(CO1[0], CO2[0])은 각각 다음 높은 비트(두번째 비트) 컴프레서(202)의 입력(CI1[1], CI2[1])이 된다. 이러한 구조는 마지막 컴프레서(n+ 2)까지 계속되고, 가장 높은 비트 컴프레서는 캐리 출력들(CO1[n+ 2], CO2[n+ 2])을 출력하지 않는다. 가장 높은 비트 컴프레서는 오버플로우를 방지하고, 가장 높은 비트 컴프레서의 다음 캐리 워드 비트 및 다음 합 워드 비트 값들로부터 두번째 입력들이 구해진다.

각 컴프레서들의 다음 캐리 워드 비트값 및 다음 합 워드 비트값은 각각 캐리 및 합 레지스터들(172, 173)에 저장된다. 마지막 결과는 합 레지스터(173) 내에 저장된 일부분과 캐리 레지스터(172)에 저장된 다른 부분 형태로 분할되어서 발생된다. 최종 단일 워드 결과(S_N[n:0])를 얻기 위해서, 합 레지스터(172)와 캐리 레지스터(173)에 저장된 값들은 캐리 전파 가산기(174)에서 더해지고, 최종 단일 워드 결과(S_N[n:0])는 최종 레지스터(175)에 저장된다. 종래 시스템의 CPA 모드 대신에 본 발명의 실시예에 따른 시스템은 도 7에 도시된 바와 같이, CSA 모드를 사용한다. CSA 컴프레서는 각각의 가산기와 관련있는 3 개의 지연 경로들을 갖는다. 종래의 누산기에서, 지연 경로는 각 비트마다 존재하였다.

그러므로, 본 발명의 바람직한 실시예에 따른 도 7에서, 비트 크기 n과 무관하게 모든 컴프레서들에 대해 세 개의 지연 경로들이 존재한다. 종래의 기술에서는 "n"개의 지연 경로가 있었다. 따라서 본 발명은 모듈러 곱셈의 계산 속도를 현저하게 향상시킬 수 있다. 예를 들어, 종래의 1024 비트 곱셈기는 1024 지연(전가산기 경로)을 갖는 누산기를 포함하였으나, 본 발명의 바람직한 실시예에 따른 경로 지연은 단일 완전 컴프레서 또는 축소된 컴프레서에서 3이다. 이 예에서, 본 발명은 종래의 시스템에 비해 300배 빠르다.

본 발명의 다른 실시예는 누산기에서 CSA 및 CPA 사이를 스위칭할 수 있는 다양한 조합들을 포함한다. 도 11는 본 발명의 다른 실시예에 따른 누산기(170b)를 보여주고 있다. 누산기(170b)는 컴프레서들와 결합하여 원할 때마다 CPA 및 CSA 모드 사이를 스위칭하는데 사용되는 멀티플렉서들(MXG_{n+1}~MXG_0)을 포함한다. 이와 같은 구성에서 캐리 전파 가산기(174)는 리던던트 수(redundant number)를 노말 수(normal number)로 변환하지 않는다. 도 10의 누산기(170b)는 CSA 또는 CPA 모드에서 선택적으로 동작하고, 출력은 이미 노말 수 형태이다. 캐리 전파 가산기(174)를 제거함으로서 요구되는 하드웨어 크기가 감소된다.

멀티플렉서들(MXG_{n+1}~MXG_0)은 컴프레서들 내의 전가산기들 사이의 전기적 연결을 제어할 수 있다. 도 11에서, 처음 두 비트 컴프레서들(301, 302)은, 낮은 비트 컴프레서(301)에 의해서 사용되는 현재 캐리 워드 비트값(C_I[k-1])을 얻기 위해 다음 캐리 워드 비트값(C_{I+1}[k])이 캐리 레지스터(172)를 통과하지 않는다는 점을 제외하고는 컴프레서들(201, 202)과 각각 유사하게 동작한다. C_{I+1}[k]는 다음 높은 비트 컴프레서(k+ 1)를 통과하여 높은 비트 컴프레서의 멀티플렉서 그룹(400_{-2})으로 입력된다.

도 12는 본 발명의 실시예에 따른 k번째 비트 멀티플렉서 그룹(400)을 보여주고 있고, 도 13은 도 11에 도시된 멀티플렉서 그룹(400)이 도 10에 도시된 누산기(170b)에 포함된 상세 회로도이다. 계산 모드(CSA 사용 모드 또는 CPA 사용 모드)는 스위칭 신호(SW)에 의해서 제어될 수 있다. 본 발명의 실시예에서, k번째 비트의 멀티플렉서(400)는 도 9의 축소된 컴프레서(203)의 제 2 가산기(212)와 제 3 가산기(213) 사이에 위치한다. 따라서, 제 1 멀티플렉서(410)로 입력되는 제 1 입력(401)은 가산기(212)로부터의 FSO2이다. 제 1 멀티플렉서(410)의 제 2 입력(402)은 k-1번째 비트 컴프레서로부터의 현재 캐리 워드 비트값(C_I[k-1])이다. 현재 캐리 워드 비트값은 k-1번째 비트 컴프레서에 대한 다음 캐리 워드 비트값 C_{I+1}[k-1]으로부터 구해진다. 제 2 멀티플렉서(411)는 두 가지 값들을 입력받는데, 하나(403)는 k-1번째 비트 컴프레서로부터의 제 1 출력 캐리값(CO1[k-1])이고, 다른 하나(404)는 k번째 비트 컴프레서로부터의 현재 합 워드 비트값(S_I[k])이다. 현재 합 워드 비트값은 다음 합 워드 비트값(S_{I+1}[k])이 합 레지스터(173)를 통과하면서 구해진다. 제 3 멀티플렉서(412)의 두 입력들(405, 406)은 각각 k-1 비트 컴프레서로부터의 제 2 출력 캐리값(CO2[k-1]) 및 k-1 비트 컴프레서로부터의 다음 캐리 워드 비트값(C_{I+1}[k-1])이다.

스위칭 신호(SW)는 각 멀티플렉서들(410, 411, 412)의 두 입력들 중 어느 것을 제 3 전가산기(213)로 전달할 것인지를 결정한다. 어떤 값이 멀티플렉서들(410, 411, 412)을 통과하느냐에 따라서 캐리 저장 가산 또는 캐리 전파 가산의 동작 모드가 결정된다. 스위칭 신호(SW)의 값이 0이면, 스위칭 컴프레서들은 캐리 세이브 가산 모드로 동작된다. 만일 스위칭 신호(SW)의 값이 1이면, 컴프레서들의 아래 전가산기들은 직렬로 연결되고, 캐리 전파 가산 모드로 동작된다. 전가산기(213)는 다음 캐리 워드 비트값과 다음 합 워드 비트 값을 앞서 설명한 방법으로 출력한다.

캐리 및 합 워드들은 N번 반복해서 계산된다. N은 n이 짝수일 때 (n+ 2)/2이고, n이 홀수일 때 (n+ 1)/2이다. 현재 반복 사이클에서 출력되는 캐리 및 합 값들은 이전 반복 사이클의 캐리 및 합 값들과 더해지고 캐리 레지스터(172) 및 합 레지스터(173)에 각각 더해진다. 최종 결과(S_N[n:0])는 스위칭 신호(SW)의 변경에 따라서 레지스터들(172, 173)에 각각 저장된 캐리 및 합을 더하는 것에 의해서 구해진다.

CPA 가산기(174)와 레지스터(175)를 더한 것보다 멀티플렉서 그룹들(400)의 크기가 매우 작기 때문에 도 10에 도시된 실시예는 하드웨어 크기를 감소시킨다.

도 14는 본 발명의 다른 실시예에 따른 곱셈기(2000)를 보여주고 있다. 도 14에 도시된 곱셈기(2000)는, 도 2에 도시된 곱셈기(1000)의 하나의 멀티플렉서(130)를 세 개의 멀티플렉서들(131, 132, 133)로 대체하였다. 또한, 멀티플렉서가 분할됨에 따라, 도 2에 도시된 부스 레코더(140)가 멀티플렉서들(131, 132, 133)로 제공될 신호들(SFT_PP, NEG_PP)을 발생하기 위한 부스 레코더(190)로 대체된다.

도 15는 도 14에 도시된 부스 레코더(190)의 코딩 스킴을 보여주고 있고, 도 16은 도 15의 코딩 스킴에 따라서 부스 레코더(190)를 구현한 회로의 실시예를 보여준다. 부스 레코더(190)는 인버터들(401-403)과 낸드 게이트들(411-419)을 포함한다.

도 17은 정수 모듈러 연산과 다항식 모듈러 연산을 선택적으로 수행할 수 있는 본 발명의 바람직한 실시예에 따른 곱셈기를 보여주고 있다. 도 2에 도시된 곱셈기(1000) 및 도 14에 도시된 곱셈기(2000)는 정수 모듈러 연산만을 수행할 수 있었으나 도 17에 도시된 곱셈기(3000)는 주어진 선택 신호(SEL_FLD)에 따라서 정수 모듈러 연산과 다항식 모듈러 연산을 선택적으로 수행할 수 있다. 이 실시예에서, 곱셈기(3000)는 선택 신호(SEL_FLD)가 '0'일 때 정수 모듈러 연산 모드를 수행하고, 선택 신호(SEL_FLD)가 '1'일 때 다항식 모듈러 연산 모드를 수행한다. 곱셈기(3000)의 정수 모듈러 연산 모드는 도 2에 도시된 곱셈기(1000)의 정수 모듈러 연산과 동일하므로 이하 다항식 모듈러 연산 모드만을 설명한다. 또한, 도 17에 도시된 곱셈기(1000)에서 도 2의 곱셈기(1000)와 동일하게 동작하는 구성 요소들은 동일한 참조 번호를 병기한다.

몽고메리 모듈러 곱셈 알고리즘으로 다항식 연산을 수행할 때 곱의 결과는 $S(x)_{I+1} := (S(x)_I + b_IA(x) + q_IM(x))$이다. 앞서 설명한 정수 연산과 마찬가지로, q_I는 $(S(x)_I + b_IA(x) + q_IM(x))$의 최하위 2비트가 "00"이 되도록 설정되어야 한다. 그리고, $b_I \in \{0, 1, 2, 3\}$이고 $b_IA \in \{0, A, 2A, 3A\}$이다. 본 발명의 실시예에서는 3A를 구하기 위해서 2A+ A를 수행한다. 그러므로, $b_IA \in \{0, A, 2A, 2A+ A\}$이다.

레지스터(101)에 저장된 승수 B는 매 사이클마다 2비트씩 우측으로 쉬프트된다. 승수 B의 최하위 2비트 b1 및 b0는 분해값 선택기(530)로 제공된다. 분해값 선택기(530)는 승수 B의 최하위 2비트 b1 및 b0를 입력받고, $PP_I[n+ 1:0]$가 0, A 또는 2A가 되도록 그리고 AI[2+ 1:0]이 0 또는 A가 되도록 신호들(SEL_A1[1:0], SEL_A2, SEL_A3)을 출력한다. 도 18은 b_IA를 구하기 위한 본 발명의 바람직한 실시예에 따른 분해값 선택기(530)의 코딩 스킴을 보여주고 있다.

$PP_I[n+ 1:0]$을 선택하기 위해, 분해값 선택기(530)는 3 가지 값들 0, A 및 2A 중 하나를 선택하기 위한 선택 신호(SEL_A1[1:0])를 멀티플렉서(520)로 출력한다. 멀티플렉서(520)는 다항식 모듈러 연산 모드일 때(즉, SEL_FLD가 '1'일 때) 선택 신호(SEL_A1[1:0])를 멀티플렉서(130)로 출력한다. 멀티플렉서(130)는 피승수 A의 값들 및 선택 신호(SEL_A1[1:0])를 받아들이고, 출력을 AND 게이트(150)로 출력한다. 멀티플렉서(550)는 모드 신호(SEL_FLD)가 '1'일 때 분해값 선택기(530)로부터의 선택 신호(SEL_A2)를 AND 게이트(150)로 출력한다. AND 게이트(150)는 멀티플렉서(130)로부터의 입력과 멀티플렉서(550)로부터의 선택 신호(SEL_A2)를 받아들이고, 제 1 분해값($PP_I[n+ 1:0]$)을 누산기(580)로 출력한다.

$A_I[n+ 1:0]$를 선택하기 위해, 분해값 선택기(530)는 0 또는 A 중 하나를 선택하기 위한 선택 신호(SEL_A3)를 멀티플렉서(540)로 출력한다. 멀티플렉서(540)는 0, 피승수 A의 값 및 선택 신호(SEL_A3)를 받아들이고, 제 2 분해값($A_I[n+ 1:0]$)을 출력한다. 멀티플렉서(540)의 출력 중 최하위 2비트($A_I[1:0]$)는 멀티플렉서(560)와 가산기(591)로 입력되고, 나머지 비트($A_I[n+ 1:2]$)는 누산기(580)로 입력된다.

멀티플렉서(560)는 다항식 모듈러 연산 모드일 때 멀티플렉서(540)로부터의 출력중 최하위 2비트($A_I[1:0]$)를 CW[1:0]으로서 출력한다. 멀티플렉서(560)의 출력(CW[1:0])은 누산기(580)로 입력된다.

가산기(591)는 제 1 분해값($PP_I[n+ 1:0]$)의 최하위 2비트($PP_I[1:0]$)와 제 2 분해값($A_I[n+ 1:0]$)의 최하위 2비트($A_I[1:0]$)를 더하고, 결과를 가산기(592)로 출력한다. ($PP_I[1:0]+ A_I[1:0]$)는 $b_IA(x)[1:0]$이다.

가산기(592)는 가산기(591)의 출력과 합 레지스터(583)에 저장된 값의 하위 2 비트($S_I[1:0]$)를 더하고, 결과($SSPP_I[1:0]$)를 모듈러스 선택기(570)로 출력한다.

모듈러스 선택기(570)는 $(S(x)_I + b_IA(x) + q_IM(x))$의 최하위 2비트가 '00'이 될 수 있도록 모듈러스($MM_I[n+ 1:0]$)를 선택한다. 몽고메리 알고리즘에서, 모듈러스의 최하위 비트($MM_I[0]$)는 항상 1이다. 그러므로, $(S(x)_I + b_IA(x))$와 M[1]로부터 q_I가 결정된다.

도 19는 본 발명의 바람직한 실시예에 따른 모듈러스 선택기(570)의 코딩 스킴을 보여주고 있다.

$MM_I[n+ 1:0]$을 선택하기 위해, 모듈러스 선택기(570)는 3 가지 값들 0, M 및 2M 중 하나를 선택하기 위한 선택 신호(SEL_M1[1:0])를 멀티플렉서(510)로 출력한다. 멀티플렉서(510)는 다항식 모듈러 연산 모드일 때(즉, SEL_FLD가 '1'일 때) 선택 신호(SEL_M1[1:0])를 멀티플렉서(120)로 출력한다. 멀티플렉서(120)는 모듈러스 M의 값들 및 선택 신호(SEL_M1[1:0])를 받아들이고, 출력($MM_I[n+ 1:0]$)을 누산기(580)로 출력한다.

누산기(580)의 상세한 회로 구성이 도 20에 도시되어 있다. 누산기(580)는 도 13에 도시된 누산기(170b)와 동일하게, 직렬로 연결된 n+ 2 개의 5-2 컴프레서들로 구성되며, 컴프레서들은 완전 컴프레서들과 축소된 컴프레서들로 나뉘어진다.

여기서, n은 모듈러스 값 M의 비트 길이이다. 본 발명의 실시예에 따른 누산기(580)는 도 13에 도시된 누산기(170b)의 구성 요소들에 멀티플렉서 그룹(620)과 멀티플렉서들 (640, 650)을 더 포함한다. 멀티플렉서 그룹(620)은 멀티플렉서 그룹(610)과 전가산기(630) 사이에 위치한다.

정수 모듈러 연산 모드(SEL_FLD='0')에서, 멀티플렉서 그룹(610)으로부터의 출력은 멀티플렉서 그룹(620)을 통해 전가산기(630)로 입력되고, 전가산기(630)로부터 출력되는 캐리는 낮은 비트 컴프레서의 캐리 입력으로 전달된다. 누산기(580)의 정수 모듈러 연산 모드는 도 13에 도시된 누산기(170b)의 동작과 동일하다.

도 21은 본 발명의 실시예에 따른 k번째 비트 멀티플렉서 그룹들(610, 620)을 보여주고 있다. 멀티플렉서 그룹(610)은 도 12에 도시된 멀티플렉서 그룹(400)과 동일하게 제 1 내지 제 3 멀티플렉서들(611, 612, 613)을 가지며 제 1 내지 제 6 입력들(601-606)을 받아들인다. 도 12 및 도 21에 도시된 멀티플렉서 그룹들(400, 610)은 인출 번호만 다를 뿐 동일한 신호를 받아들여서 동일하게 동작하므로 상세한 설명은 생략한다.

멀티플렉서 그룹(620)은 다항식 모듈러 연산 모드동안 멀티플렉서(540)로부터의 제 2 분해값(A_I[n+ 1:2])을 제 1 분해값(PP_I[n+ 1:2])과 더하기 위해 제공된다. 제 2 분해값(A_I[n+ 1:2])의 하위 2비트는 멀티플렉서(560)를 통하여 CW[1:0]로서 출력되어서 누산기(581)로 입력된다. 그러므로, CW[1:0]은 누산기(580)에 의해서 제 1 분해값의 하위 2비트(PP_I[1:0])와 더해진다. 멀티플렉서 그룹(620)은 제 4 및 제 5 멀티플렉서들(621, 622)을 포함한다. 제 4 멀티플렉서(621)는 제 2 멀티플렉서(612)로부터의 출력과 멀티플렉서(540)로부터의 제 2 분해값(A_I[k])를 받아들인다. 제 5 멀티플렉서(622)는 제 3 멀티플렉서(613)로부터의 출력과 '0'을 받아들인다. 제 4 및 제 5 멀티플렉서들(2)621, 622)은 정수 모듈러 연산 모드(SEL_FLD='0')에서 제 2 및 제 3 멀티플렉서들(612, 613)로부터의 입력들을 각각 출력하고, 다항식 모듈러 연산 모드(SEL_FLD="1")에서 A_I[k] 및 0을 출력한다. 멀티플렉서들(621, 622)의 출력들(I12, I13)은 전가산기(630)로 제공된다.

다시 도 20을 참조하면, k번째 비트 멀티플렉서들(640, 650)은 다항식 모듈러 연산 모드(SEL_FLD='1') 동안 제 2 분해값(M_I[k])을 k번째 컴프레서의 캐리 입력으로 선택적으로 제공한다. 이는 제 1 분해값(MM_I[n+ 1:0])과 제 2 분해값(M_I[n+ 1:0])을 더해서 q_IM(x)를 구하기 위함이다.

멀티플렉서(650)는 모듈러스 선택기(570)로부터의 선택 신호(SEL_M2)에 응답해서 모듈러스(M[k]) 또는 '0'을 제 2 분해값(M_I[k])으로서 멀티플렉서(640)로 출력한다. 멀티플렉서(640)는 정수 모듈러 연산 모드(SEL_FLD='0')일 때 캐리 레지스터(582)의 k번째 캐리 비트를 그리고 다항식 모듈러 연산 모드(SEL_FLD='1')일 때 멀티플렉서(650)로부터의 제 2 분해값(M_I[k])을 낮은 비트 컴프레서의 캐리 비트로서 제공한다.

도 2에 도시된 정수 모듈러스 연산을 위한 곱셈기(1000)에 몇몇 구성 요소들을 부가한 도 17의 곱셈기(3000)는 정수 모듈러스 연산 뿐만 아니라 다항식 모듈러스 연산을 수행할 수 있다.

발명의 효과

전술한 본 발명의 실시예에 의하면, 본 발명은 모듈러 곱셈기에서 최소한의 구성요소만으로 캐리-세이브 모드와 캐리전파 모드로 선택적으로 동작할 수 있는 누산기의 구조를 제공함으로써, 하드웨어적인 부담을 줄이면서 모듈러 연산속도를 향상시키는 효과가 있다.

(57) 청구의 범위

청구항 1.

n-비트 모듈러스 M 및 이전 합 그리고 현재 부분곱을 받아들이고, 선택 신호를 발생하는 모듈러스 레코더; 그리고

입력들 -M, 0, M 및 2M을 받아들이고, 상기 선택 신호에 근거해서 상기 입력들 중 하나를 선택하는 멀티플렉서를 포함하는 멀티플 모듈러스 선택기.

청구항 2.

제 1 항에 있어서,

상기 입력 -M은 상기 모듈러스 M을 반전시키는 것에 의해서 구해지는 멀티플 모듈러스 선택기.

청구항 3.

제 1 항에 있어서,

상기 입력 2M은 상기 모듈러스 M을 쉬프트하는 것에 의해서 구해지는 멀티플 모듈러스 선택기.

청구항 4.

제 1 항에 있어서,

상기 모듈러스 M은 레지스터에 저장되는 멀티플 모듈러스 선택기.

청구항 5.

제 1 항에 있어서,

상기 모듈러스 레코더는 누산기로 입력되는 멀티플 모듈러스 반전 표시 신호(NEG_MM)를 더 발생하는 멀티플 모듈러스 선택기.

청구항 6.

제 1 항에 있어서,

상기 n-비트 모듈러스 M은 두번째 최하위 비트 M[1]를 포함하고, 그리고 상기 이전 합과 상기 현재 부분곱은 2 비트 SPP_I[1:0]인 멀티플 모듈러스 선택기.

청구항 7.

제 1 항에 있어서,

상기 선택 신호는 2비트 SEL_MM[1:0]을 포함하는 멀티플 모듈러스 선택기.

청구항 8.

캐리 가산 모드로 동작하고, 각각이 멀티플 모듈러스, 부분곱, 대응하는 현재 합 및 대응하는 현재 캐리를 받아들이고, 대응하는 다음 합 및 대응하는 다음 캐리를 발생하는 복수의 컴프레서들과;

상기 복수의 컴프레서들 각각으로부터 상기 대응하는 다음 합을 받아들이고, 대응하는 갱신된 현재 합을 출력하는 합 레지스터; 그리고

상기 복수의 컴프레서들로부터 상기 대응하는 다음 캐리를 받아들이고, 대응하는 갱신된 현재 캐리를 출력하는 캐리 레지스터를 포함하는 누산기.

청구항 9.

제 8 항에 있어서,

상기 합 레지스터 및 상기 캐리 레지스터는 분리된 레지스터인 누산기.

청구항 10.

제 8 항에 있어서,

상기 멀티플 모듈러스는 모듈러스에 근거해서 생성되는 누산기.

청구항 11.

제 10 항에 있어서,

상기 모듈러스의 비트 길이가 n일 때 상기 복수의 컴프레서들은 n+ 3 컴프레서들을 포함하는 누산기.

청구항 12.

제 11 항에 있어서,

상기 모듈러스는 n-비트 레지스터에 저장되는 누산기.

청구항 13.

제 8 항에 있어서,

상기 부분곱은 승수 및 피승수로부터 구해지는 누산기.

청구항 14.

제 13 항에 있어서,

상기 승수의 비트 길이는 n+ 1인 누산기.

청구항 15.

제 14 항에 있어서,

상기 승수는 (n+ 1)-비트 레지스터에 저장되는 누산기.

청구항 16.

제 13 항에 있어서,

n이 짝수이면, 상기 승수의 비트 길이는 n+ 2이고, n이 홀수이면, 상기 승수의 비트 길이는 n+ 1인 누산기.

청구항 17.

제 16 항에 있어서,

n이 짝수이면, 상기 승수는 (n+ 2)-비트 레지스터에 저장되고, n이 홀수이면, 상기 승수는 (n+ 1)-비트 레지스터에 저장되는 누산기.

청구항 18.

제 8 항에 있어서,

상기 복수의 컴프레서들은 5:2 컴프레서인 누산기.

청구항 19.

제 8 항에 있어서,

상기 복수의 컴프레서들의 제 1 그룹은 상기 대응하는 다음 합 및 상기 대응하는 다음 캐리를 발생하기 위한 보상 워드를 더 받아들이는 누산기.

청구항 20.

제 19 항에 있어서,

상기 복수의 컴프레서들의 제 1 그룹은 완전 컴프레서들인 누산기.

청구항 21.

제 19 항에 있어서,

상기 복수의 컴프레서들의 제 2 그룹은 상기 보상 워드를 받아들이지 않는 누산기.

청구항 22.

제 21 항에 있어서,

상기 복수의 컴프레서들의 상기 제 2 그룹은 축소된 컴프레서들인 누산기.

청구항 23.

제 8 항에 있어서,

상기 부분곱 및 상기 멀티플 모듈러스는 각각 n+ 2 비트인 누산기.

청구항 24.

제 19 항에 있어서,

상기 보상 워드는 2 비트인 누산기.

청구항 25.

제 19 항에 있어서,

상기 완전 컴프레서는 3 개의 전가산기들로 구성되는 누산기.

청구항 26.

제 22 항에 있어서,

상기 축소된 컴프레서는 1 개의 반가산기와 2 개의 전가산기들로 구성되는 누산기.

청구항 27.

제 8 항에 있어서,

최종 갱신된 현재 합 및 최종 갱신된 현재 캐리를 받아들이고, 최종 합을 출력하는 캐리 전파 가산기; 그리고

상기 최종 합을 저장하기 위한 최종 레지스터를 더 포함하는 누산기.

청구항 28.

제 8 항에 있어서,

상기 복수의 컴프레서들은 캐리 가산 모드 및 캐리 전파 모드로 동작하는 누산기.

청구항 29.

제 9 항에 있어서,

상기 캐리 가산 모드 및 상기 캐리 전파 모드는 제어 신호에 의해서 결정되는 누산기.

청구항 30.

제 28 항에 있어서,

복수의 컴프레서들의 제 1 그룹은 보상 워드를 더 수신하고 상기 대응하는 다음 합 및 상기 대응하는 다음 캐리를 발생하는 누산기.

청구항 31.

제 30 항에 있어서,

상기 복수의 컴프레서들의 제 1 그룹은 완전 컴프레서들인 누산기.

청구항 32.

제 30 항에 있어서,

상기 복수의 컴프레서들의 제 2 그룹은 상기 보상 워드를 수신하지 않는 누산기.

청구항 33.

제 32 항에 있어서,

상기 복수의 컴프레서들의 제 2 그룹은 축소된 컴프레서들인 누산기.

청구항 34.

제 33 항에 있어서,

상기 축소된 컴프레서들 각각은,

캐리 가산 모드 및 캐리 전파 모드로 동작할 수 있도록 상기 축소된 컴프레서들을 변형하는 멀티플렉서 그룹을 포함하는 누산기.

청구항 35.

제 34 항에 있어서,

각 멀티플렉서 그룹은 3 개의 2:1 멀티플렉서들을 포함하는 누산기.

청구항 36.

제 33 항에 있어서,

각각의 축소된 컴프레서들은,

상기 제어 신호에 따라서 상기 캐리 가산 모드 또는 상기 캐리 전파 모드 중 어느 하나로 동작하도록 상기 축소된 컴프레서들을 변형하는 멀티플렉서 그룹을 포함하는 누산기.

청구항 37.

제 36 항에 있어서,

상기 캐리 가산 모드는,

현재 컴프레서의 중간 전가산기로부터 현재 컴프레서의 하부 전가산기로 전달되는 제 1 신호, 낮은 비트 컴프레서의 상부 가산기로부터 상기 현재 컴프레서의 하부 전가산기로 전달되는 제 2 신호 그리고 낮은 비트 컴프레서의 중간 전가산기로부터 상기 현재 컴프레서의 하부 전가산기로 전달되는 제 3 신호를 포함하는 누산기.

청구항 38.

제 36 항에 있어서,

상기 캐리 전파 모드는 하위 비트 컴프레서의 하부 전가산기로부터 상위 비트 컴프레서의 멀티플렉서 그룹으로 전달되는 제 1 신호와 상기 상위 비트 컴프레서의 멀티플렉서 그룹으로부터 상기 상위 비트 컴플레서의 하위 전가산기로 전달되는 제 2 신호를 포함하는 누산기.

청구항 39.

제 33 항에 있어서,

상기 축소된 컴프레서들 각각은,

상기 현재 컴프레서의 중간 전가산기의 합, 낮은 비트 컴프레서의 대응하는 갱신된 현재 캐리, 상기 낮은 비트 컴프레서의 제 1 및 제 2 출력들, 상기 현재 컴프레서의 갱신된 현재 합 그리고 낮은 컴프레서의 대응하는 다음 캐리를 받아들이고 제 1 내지 제 3 출력들을 출력하는 멀티플렉서 그룹을 포함하는 누산기.

청구항 40.

제 31 항에 있어서,

상기 완전 컴프레서는 세 개의 전가산기들을 포함하는 누산기.

청구항 41.

제 33 항에 있어서,

상기 축소된 컴프레서는 1개의 반가산기와 2 개의 전가산기 그리고 3 개의 2:1 멀티플렉서들로 구성되는 누산기.

청구항 42.

-M, 0, M 및 2M(단, M은 n-비트 모듈러스 수) 중 하나를 멀티플 모듈러스로 선택하는 멀티플 모듈러스 선택기와;

부분곱 값을 구하기 위해 사용되는 제 1 값을 제공하는 부스 레코더; 그리고

몽고메리 곱셈 결과를 구하기 위한 제 2 값들을 누산하는 누산기를 포함하는 몽고메리 곱셈기.

청구항 43.

제 42 항에 있어서,

모듈러스 값을 저장하는 모듈러스 수 레지스터와;

피승수 값을 저장하는 피승수 레지스터와;

승수 값을 저장하는 승수 레지스터와;

상기 승수 값 및 피승수 값을 결합하는 AND 게이트; 그리고

상기 누산기 및 상기 AND 게이트로부터의 값들을 결합하고, 결합된 값을 출력하는 두 개의 가산기들을 더 포함하되;

상기 결합된 값은 상기 멀티플 모듈러스 선택기로 입력되는 몽고메리 곱셈기.

청구항 44.

모듈러스를 수신하는 단계; 그리고

이전 합과 현재 부분곱을 받아들이는 단계를 포함하되;

상기 모듈러스, 상기 이전 합 그리고 상기 현재 부분곱은 멀티플 모듈러스 값들 -M, 0, M 및 2M 중 하나를 선택하기 위해 사용되는 멀티플 모듈러스 발생 방법.

청구항 45.

제 44 항에 있어서,

상기 모듈러스의 두번째 최하위 비트인 상기 모듈러스의 일부를 수신하는 단계를 더 포함하는 멀티플 모듈러스 발생 방법.

청구항 46.

제 44 항에 있어서,

상기 이전 합 및 현재 부분곱의 최하위 2 비트인 이전 합 및 현재 부분곱의 일부를 받아들이는 단계를 더 포함하는 멀티플 모듈러스 발생 방법.

청구항 47.

제 44 항에 있어서,

상기 생성된 멀티플 모듈러스 값들 중 하나를 선택하기 위한 선택 신호를 발생하는 단계를 더 포함하는 멀티플 모듈러스 발생 방법.

청구항 48.

제 47 항에 있어서,

상기 선택된 값을 반전시키기 위한 모듈러스 반전 표시 신호를 발생하는 단계를 더 포함하는 멀티플 모듈러스 발생 방법.

청구항 49.

승수를 받아들이는 단계; 및

적어도 하나의 부분곱 값을 발생하기 위해 부분곱 선택 신호, 부분곱 인에이블 신호, 부분곱 반전 표시 신호를 발생하는 단계를 포함하는 부분곱 발생 방법.

청구항 50.

제 49 항에 있어서,

상기 승수를 2비트씩 쉬프트하는 단계를 더 포함하는 부분곱 발생 방법.

청구항 51.

대응하는 다음 합 및 다음 캐리를 발생하기 위해 복수의 멀티플 모듈러스, 부분곱들, 대응하는 현재 합들 및 대응하는 현재 캐리들을 수신하는 단계와;

갱신된 현재 합 및 갱신된 현재 캐리들을 발생하는 단계와;

리던던트 형식의 결과를 발생하기 위해 승수가 모두 사용될때까지 상기 수신 및 발생 단계들을 반복하는 단계; 그리고

노말 형식의 결과를 발생하기 위해서 캐리 전파 가산을 수행하는 단계를 포함하는 누산 방법.

청구항 52.

제 51 항에 있어서,

상기 반복 단계는 캐리 저장 가산기에 의해서 캐리 저장 가산을 수행하는 누산 방법.

청구항 53.

제 51 항에 있어서,

상기 캐리 전파 가산 수행 단계는 캐리 전파 가산기에 의해서 수행되는 누산 방법.

청구항 54.

제 53 항에 있어서,

스위칭 신호를 발생하는 단계를 더 포함하는 누산 방법.

청구항 55.

제 54 항에 있어서,

상기 스위칭 신호를 이용하여 캐리 저장 가산 및 상기 캐리 전파 가산 사이를 스위칭하는 단계를 더 포함하는 누산 방법.

청구항 56.

라딕스 2^N(N>1) 몽고메리 곱셈을 수행하는 방법에 있어서:

피승수, 모듈러스 및 승수를 받아들이는 단계와;

리던던트 형식의 결과를 발생하기 위해서 상기 피승수, 모듈러스 및 승수와 연관된 복수의 입력들을 캐리 저장 가산하는 단계; 그리고

노말 형식의 결과를 발생하기 위해서 캐리 전파 가산을 수행하는 단계를 포함하는 곱셈 방법.

청구항 57.

제 56 항에 있어서,

상기 캐리 저장 가산은 캐리 저장 가산기에 의해 수행되고, 상기 캐리 전파 가산은 캐리 전파 가산기에 의해서 수행되는 곱셈 방법.

청구항 58.

제 56 항에 있어서,

상기 캐리 저장 가산 및 상기 캐리 전파 가산은 누산기에 의해서 수행되는 곱셈 방법.

청구항 59.

제 56 항에 있어서,

스위칭 신호를 발생하는 단계를 더 포함하는 곱셈 방법.

청구항 60.

제 59 항에 있어서,

상기 스위칭 신호를 사용하여 캐리 저장 모드 및 캐리 전파 모드 사이를 스위칭하는 단계를 더 포함하는 곱셈 방법.

청구항 61.

라딕스 2^N(N>1) 몽고메리 곱셈을 수행하는 방법에 있어서:

피승수, 모듈러스 및 승수를 받아들이는 단계와;

리던던트 형식의 결과를 발생하기 위해서 상기 피승수, 모듈러스 및 승수와 연관된 복수의 입력들을 캐리 저장 가산하는 단계; 그리고

상기 리던던트 형식의 결과인 캐리 전파 모드를 노말 형식의 결과로의 변환을 수행하는 단계를 포함하는 곱셈 방법.

청구항 62.

n-비트 모듈러스 M 및 이전 합, 그리고 현재 부분곱을 받아들이고, 제 1 선택 신호를 발생하는 모듈러스 레코더와;

상기 n-비트 모듈러스 M 및 상기 이전 합, 상기 현재 부분곱 그리고 피승수를 받아들이고, 제 2 선택 신호를 발생하는 모듈러스 선택기와;

입력들 -M, 0, M 및 2M을 받아들이고, 정수 모듈러 연산 모드일 때 상기 제 1 선택 신호에 근거해서 상기 입력들 중 하나를 선택하고, 다항식 모듈러 연산 모드일 때 상기 제 2 선택 신호에 근거해서 상기 입력들 중 하나를 선택하는 멀티플렉서를 포함하는 멀티플 모듈러스 선택기.

청구항 63.

제 62 항에 있어서,

상기 입력 -M은 상기 모듈러스 M을 반전시키는 것에 의해서 구해지는 멀티플 모듈러스 선택기.

청구항 64.

제 62 항에 있어서,

상기 입력 2M은 상기 모듈러스 M을 쉬프트하는 것에 의해서 구해지는 멀티플 모듈러스 선택기.

청구항 65.

제 62 항에 있어서,

상기 모듈러스 M은 레지스터에 저장되는 멀티플 모듈러스 선택기.

청구항 66.

제 62 항에 있어서,

상기 모듈러스 레코더는 누산기로 입력되는 멀티플 모듈러스 반전 표시 신호(NEG_MM)를 더 발생하는 멀티플 모듈러스 선택기.

청구항 67.

제 62 항에 있어서,

상기 n-비트 모듈러스 M은 둘째 최하위 비트 M[1] 그리고 이전 합과 현재 부분곱의 합인 SPP_I[1:0]을 포함하는 멀티플 모듈러스 선택기.

청구항 68.

제 62 항에 있어서,

상기 제 1 선택 신호는 2비트 SEL_MM[1:0]을 포함하는 멀티플 모듈러스 선택기.

청구항 69.

제 62 항에 있어서,

상기 모듈러스 선택기는 누산기로 입력되는 멀티플 모듈러스 누산 표시 신호(SEL_M2)를 더 발생하는 멀티플 모듈러스 선택기.

청구항 70.

제 62 항에 있어서,

상기 이전 합, 상기 현재 부분곱 그리고 피승수인 $SSPP_I$[1:0]을 포함하는 멀티플 모듈러스 선택기.

청구항 71.

제 62 항에 있어서,

상기 제 2 선택 신호는 2비트 SEL_M1[1:0]을 포함하는 멀티플 모듈러스 선택기.

청구항 72.

정수 모듈러 연산 모드일 때 -M, 0, M 및 2M(단, M은 n-비트 모듈러스 수) 중 하나를 멀티플 모듈러스로 선택하고, 다항식 모듈러 연산 모드일 때 0, M 및 2M 중 하나를 멀티플 모듈러스로 선택하고, 멀티플 모듈러스 누산 표시 신호(SEL_M2)를 출력하는 멀티플 모듈러스 선택기와;

부분곱 값을 구하기 위해 사용되는 제 1 값을 제공하는 부스 레코더; 그리고

몽고메리 곱셈 결과를 구하기 위한 제 2 값들을 누산하는 누산기를 포함하되;

상기 누산기는 상기 다항식 모듈러 연산 모드일 때 상기 멀티플 모듈러스 누산 표시 신호(SEL_M2)에 근거해서 상기 제 2 값들에 상기 M을 더하는 몽고메리 곱셈기.

청구항 73.

제 72 항에 있어서,

모듈러스 값을 저장하는 모듈러스 수 레지스터와;

피승수 값을 저장하는 피승수 레지스터와;

승수 값을 저장하는 승수 레지스터와;

승수 값 및 피승수 값을 결합하는 AND 게이트; 그리고

상기 누산기 및 상기 AND 게이트로부터의 값들을 결합하고, 결합된 값을 출력하는 두 개의 가산기들을 더 포함하되;

상기 결합된 값은 상기 멀티플 모듈러스 선택기로 입력되는 몽고메리 곱셈기.

청구항 74.

제 72 항에 있어서,

상기 멀티플 모듈러스 선택기는,

n-비트 모듈러스 M 및 이전 합, 그리고 현재 부분곱을 받아들이고, 제 1 선택 신호를 발생하는 모듈러스 레코더와;

상기 n-비트 모듈러스 M 및 상기 이전 합, 상기 현재 부분곱 그리고 피승수를 받아들이고, 제 2 선택 신호를 발생하는 모듈러스 선택기; 그리고

입력들 -M, 0, M 및 2M을 받아들이고, 정수 모듈러 연산 모드일 때 상기 제 1 선택 신호에 근거해서 상기 입력들 중 하나를 선택하고, 다항식 모듈러 연산 모드일 때 상기 제 2 선택 신호에 근거해서 상기 입력들 0, M 및 2M 중 하나를 상기 멀티플 모듈러스로서 선택하는 멀티플렉서를 포함하는 몽고메리 곱셈기.

청구항 75.

제 72 항에 있어서,

상기 부스 레코더는,

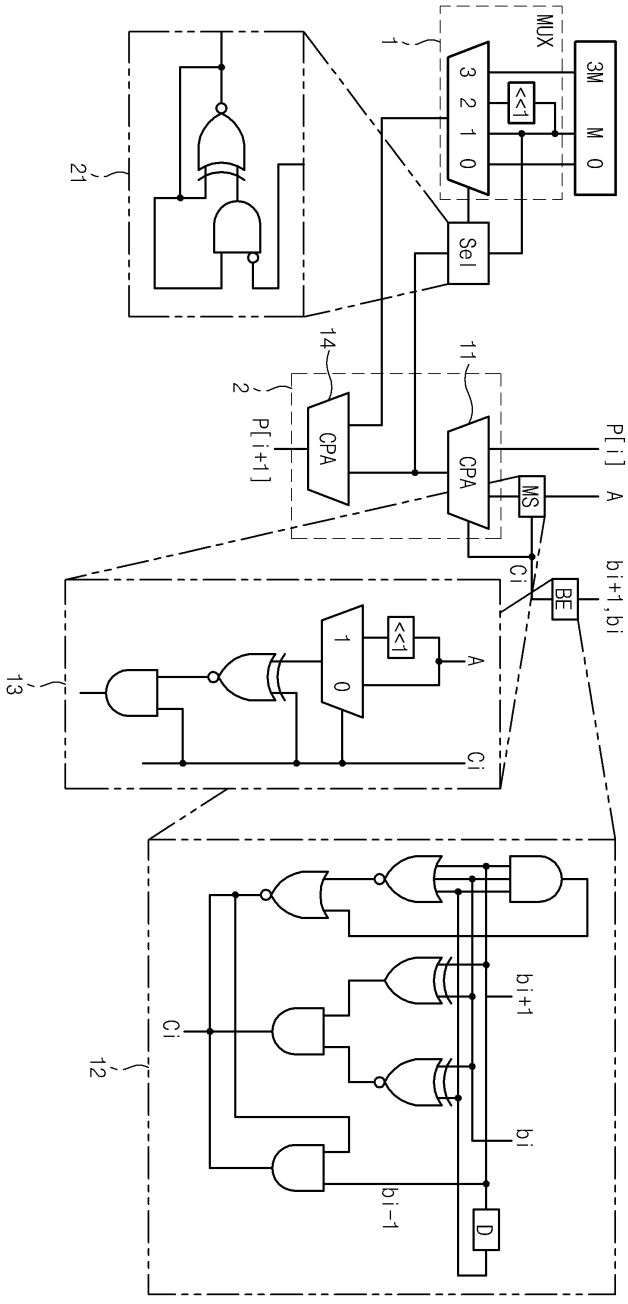
승수를 받아들이고, 제 3 선택 신호(SEL_PP[1:0])를 발생하는 제 1 선택기와;

상기 승수를 받아들이고, 제 4 선택 신호(SEL_A1[1:0])를 발생하는 제 2 선택기; 그리고

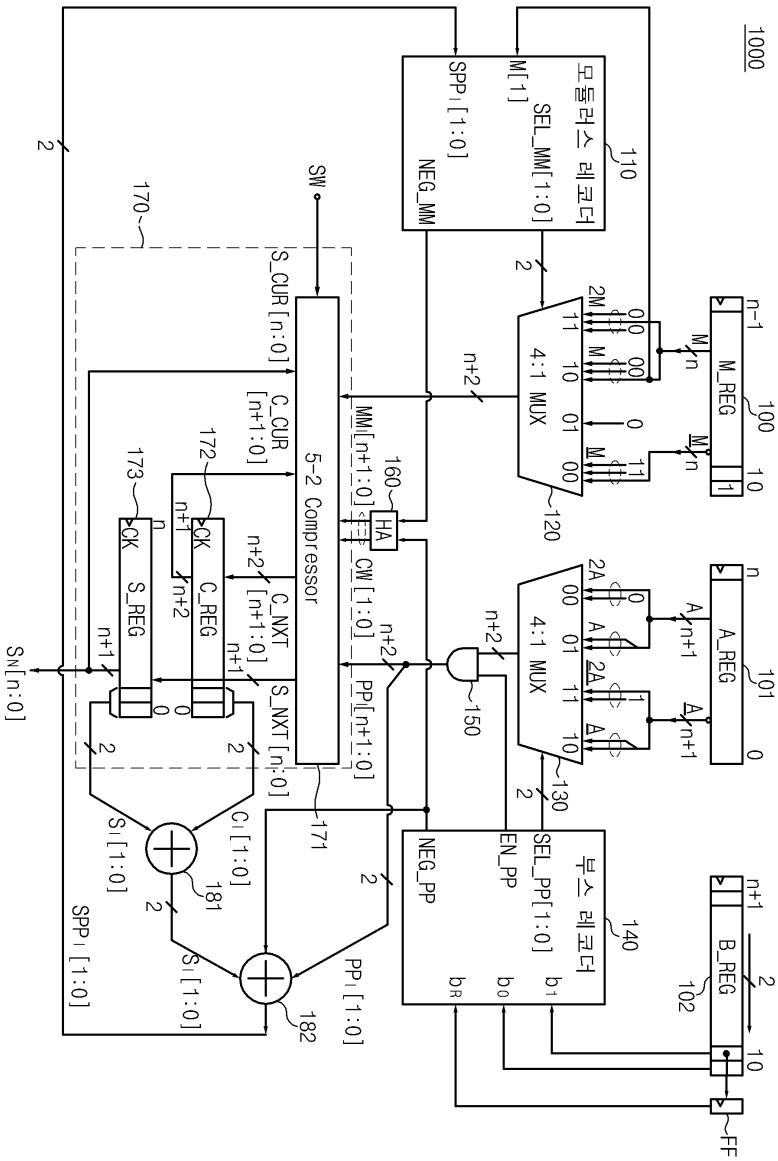
입력들 -A, 0, A 및 2A를 받아들이고, 정수 모듈러 연산 모드일 때 상기 제 3 선택 신호에 근거해서 상기 입력들 중 하나를 선택하고, 다항식 모듈러 연산 모드일 때 상기 제 4 선택 신호에 근거해서 상기 입력들 0, A 및 2A 중 하나를 선택하는 멀티플렉서를 포함하는 몽고메리 곱셈기.

도면

도면1


MUX
1
3 2 1 0
<<1
3M M 0
Sel
21
14
11
2
P[i+1]
CPA
CPA
P[i] A
MS
Ci
BE
bi+1,bi
13
1 0
<<1
A
Ci
12
Ci
bi+1
bi
bi-1
D

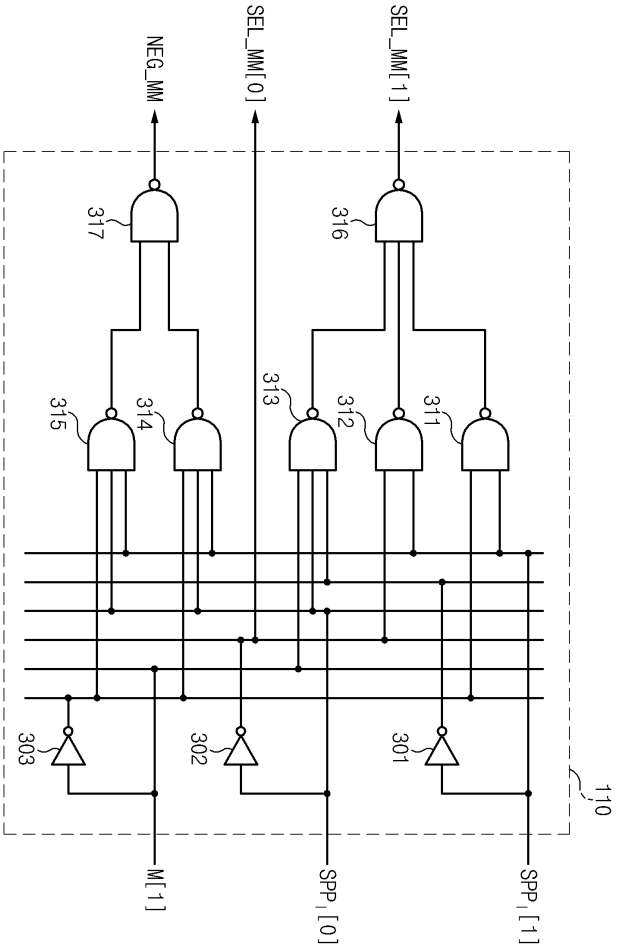
도면2


1000
모듈러스 레코더
SEL_MM[1:0]
NEG_MM
SPP[1:0]
M[1]
110
SW
170
S_CUR[n:0]
C_CUR[n+1:0]
5-2 Compressor
MM[n+1:0]
CW[1:0]
PP[n+1:0]
S_NXT[n:0]
C_NXT[n+1:0]
C_REG
S_REG
HA
160
172
173
171
M_REG
A_REG
B_REG
100
101
102
4:1 MUX
120
130
150
부스 레코더
SEL_PP[1:0]
EN_PP
NEG_PP
140
FF
SPP[1:0]
SP[1:0]
CP[1:0]
PP[1:0]
181
182
SN[n:0]

도면3

모듈러스 레코더(110)의 입력			모듈러스 레코더(110)의 출력		선택된 $MM_1[n+1:0]$
$SPP_1[1]$	$SPP_1[0]$	M[1]	SEL_MM[1:0]	NEG_MM	
0	0	0	11	0	0
0	0	1	11	0	0
0	1	0	01	1	$\underline{M}$
0	1	1	00	0	M
1	0	0	10	0	2M
1	0	1	10	0	2M
1	1	0	00	0	M
1	1	1	01	1	$\underline{M}$

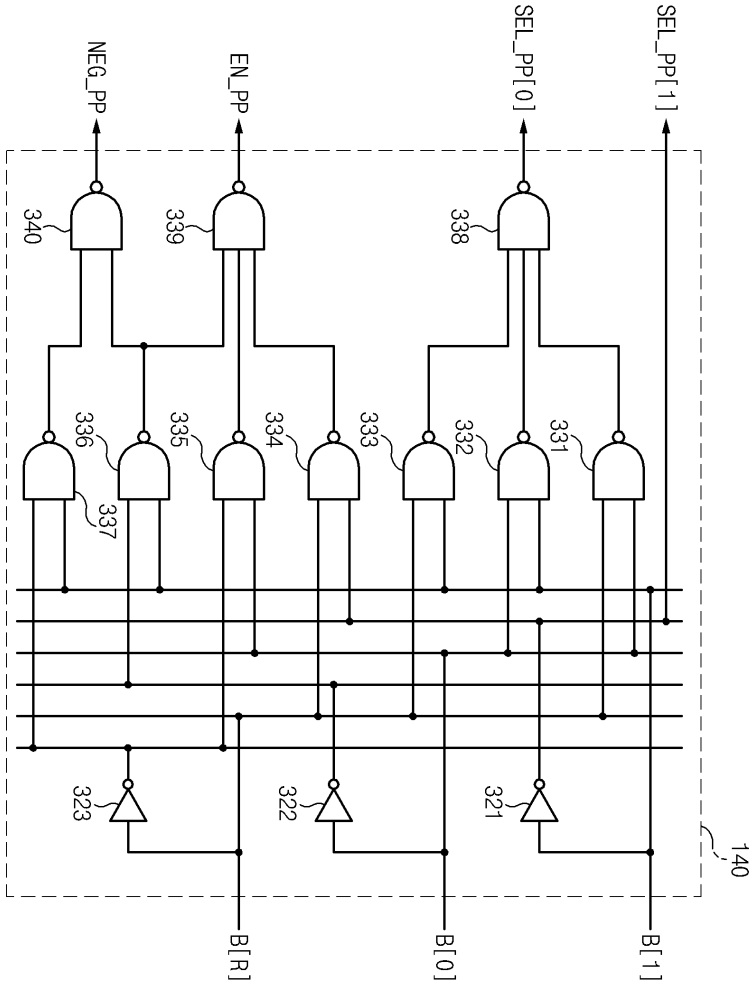
도면4


NEG_MM
SEL_MM[0]
SEL_MM[1]
317
316
315
314
313
312
311
303
302
301
110
M[1]
SPP_[0]
SPP_[1]

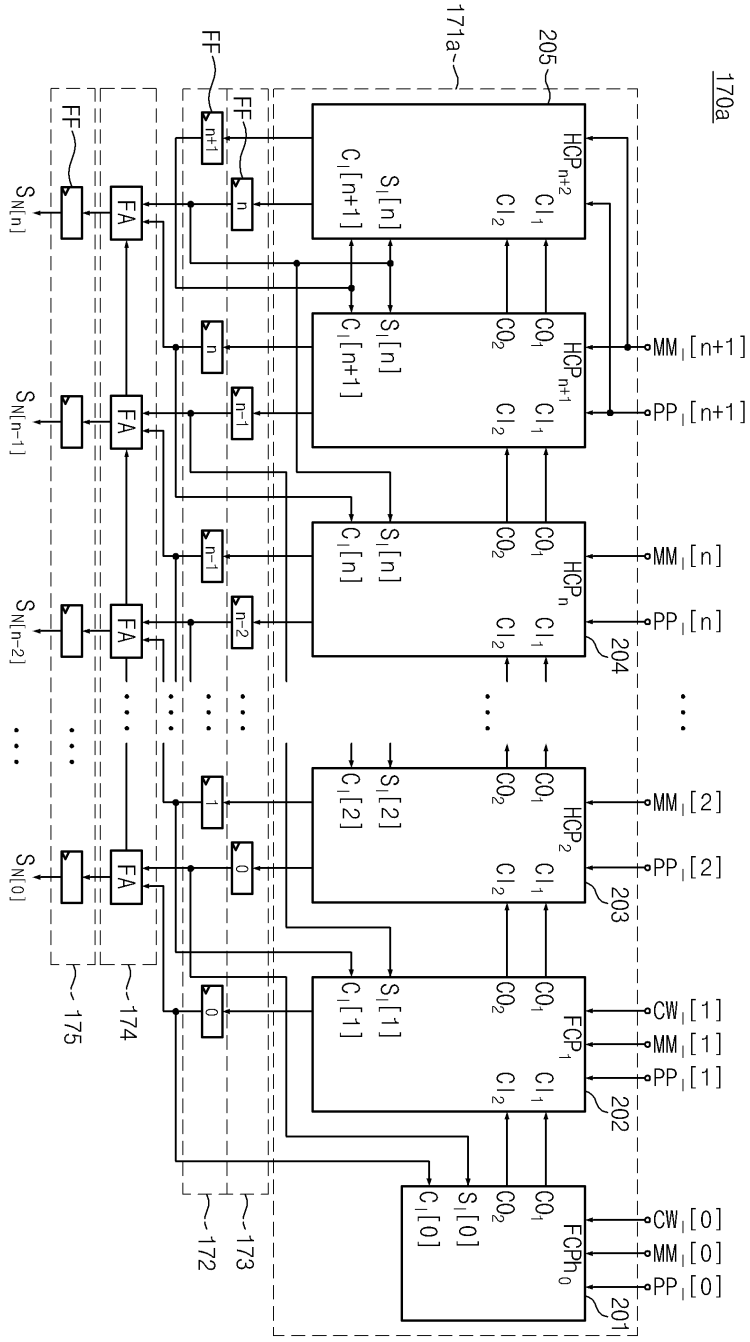
도면5

부스 레코더(140)의 입력			부스 레코더(140)의 출력			선택된 $PP_i[n+1:0]$
b_1	b_0	b_R	SEL_PP[1:0]	EN_PP	NEG_PP	
0	0	0	Don't care	0	0	0
0	0	1	00	1	0	A
0	1	0	00	1	0	A
0	1	1	10	1	0	2A
1	0	0	11	1	1	$\overline{2A}$
1	0	1	01	1	1	$\overline{A}$
1	1	0	01	1	1	$\overline{A}$
1	1	1	Don't care	0	0	0

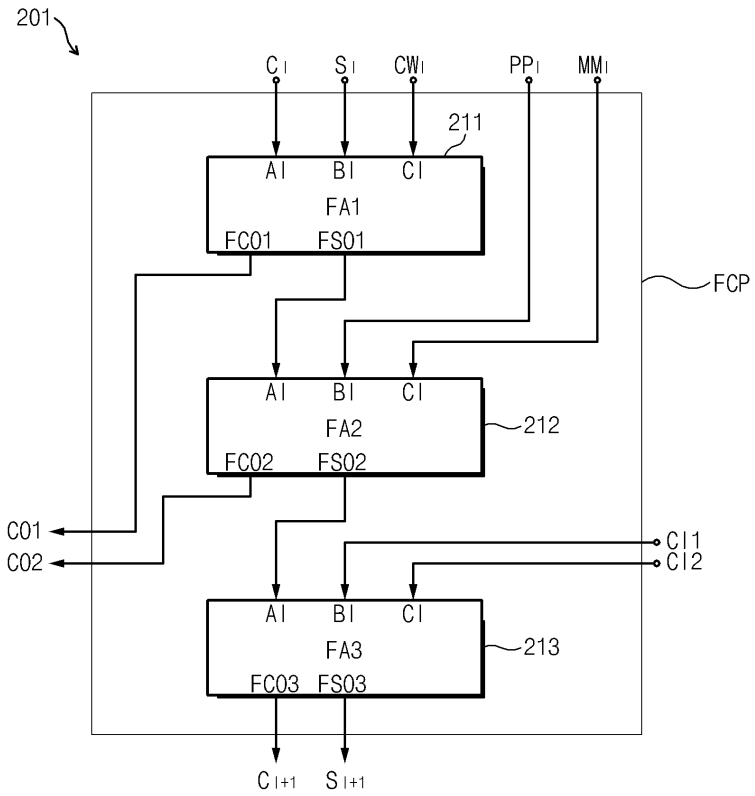
도면6


NEG_PP
EN_PP
SEL_PP[0]
SEL_PP[1]
340
339
338
336
335
334
333
332
331
337
323
322
321
140
B[R]
B[0]
B[1]

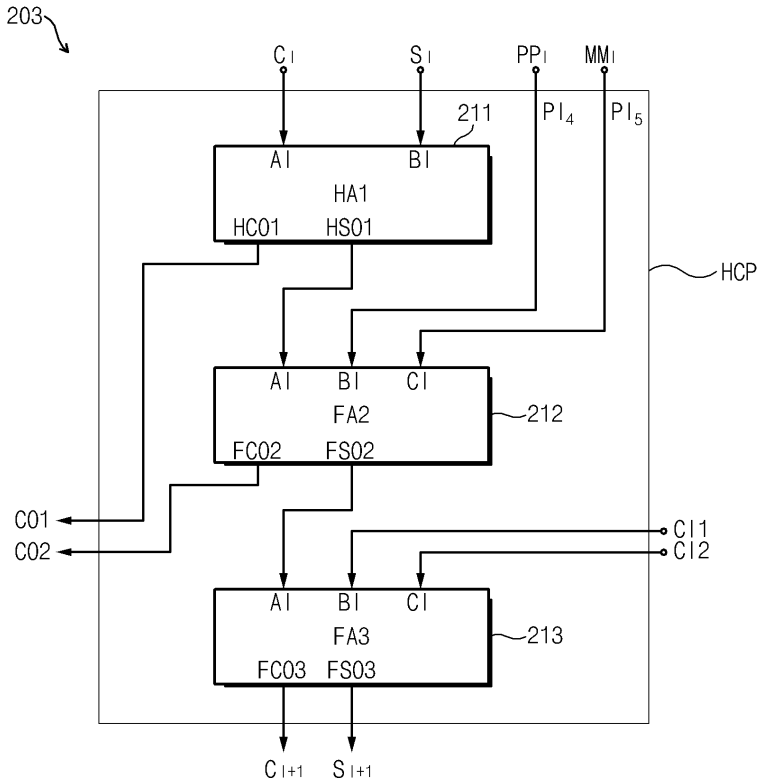
도면7


170a
171a
205
FF
FA
HCP
FCP
CW[1]
MM[n+1]
PP[n+1]
MM[n]
PP[n]
MM[2]
PP[2]
CW[1]
MM[1]
PP[1]
CW[0]
MM[0]
PP[0]
204
203
202
201
172
173
174
175

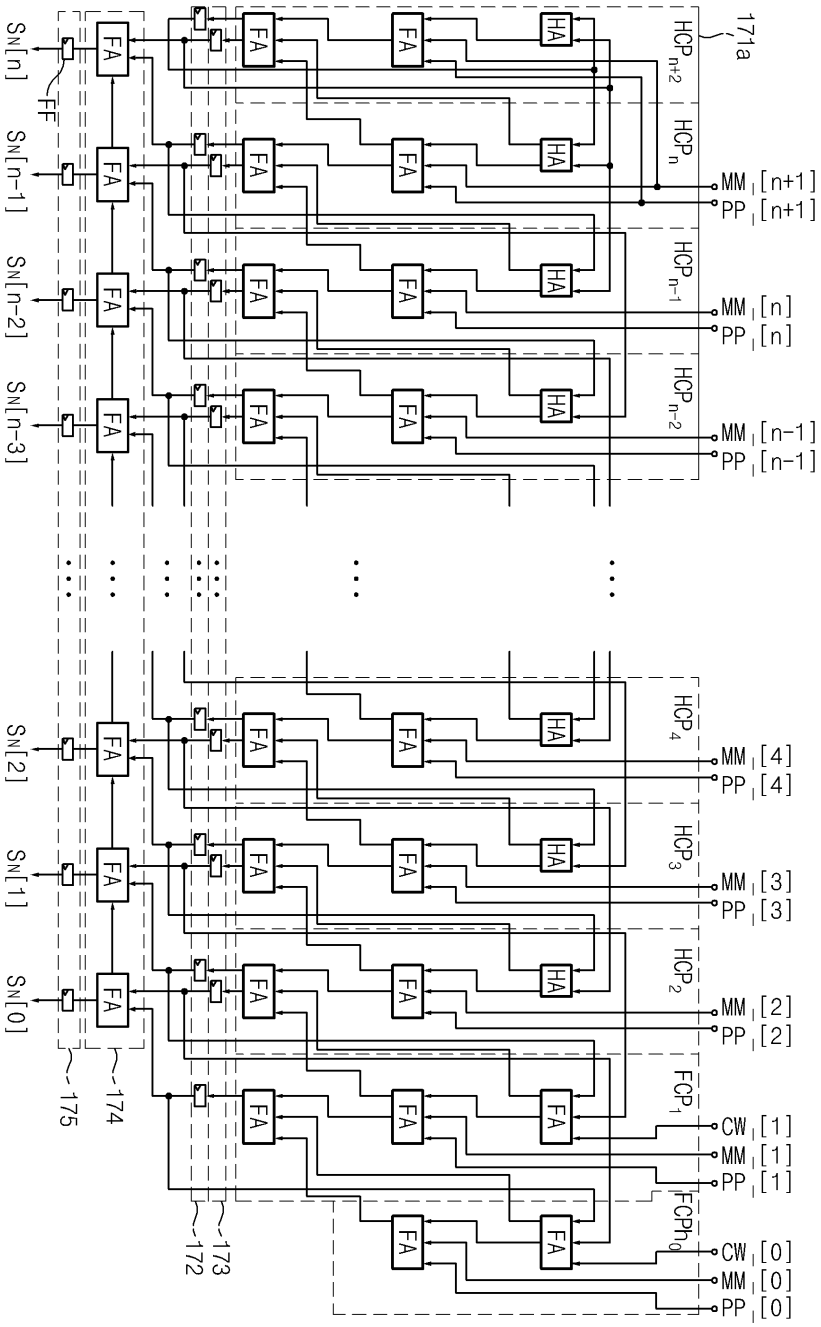
도면8


201
C I
S I
CW I
PP I
MM I
211
A I
B I
C I
FA1
FC01
FS01
FCP
A I
B I
C I
FA2
212
FC02
FS02
C01
C02
C I1
C I2
A I
B I
C I
FA3
213
FC03
FS03
C I+1
S I+1

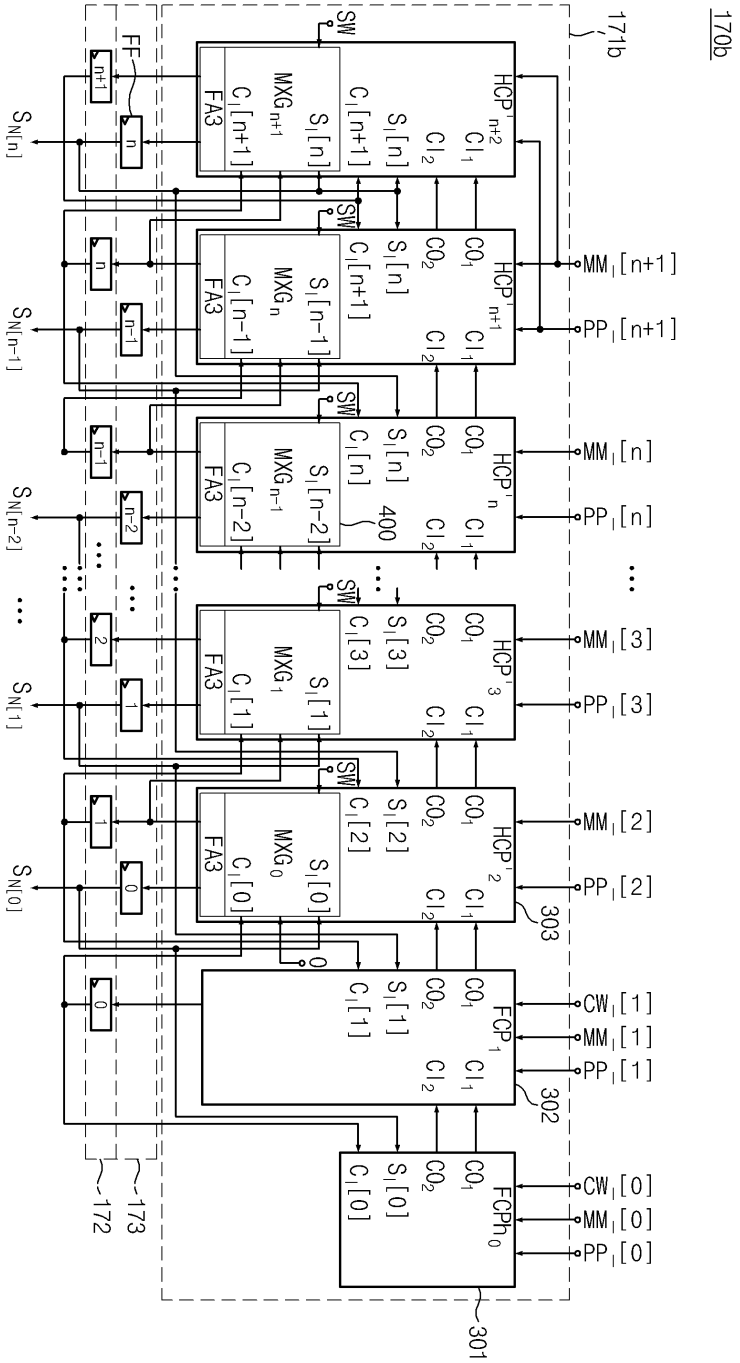
도면9


203
C I
S I
PP I
MM I
211
PI4
PI5
A I
B I
HA1
HC01
HS01
HCP
A I
B I
C I
FA2
212
FC02
FS02
C01
C02
C I1
C I2
A I
B I
C I
FA3
213
FC03
FS03
C I+1
S I+1

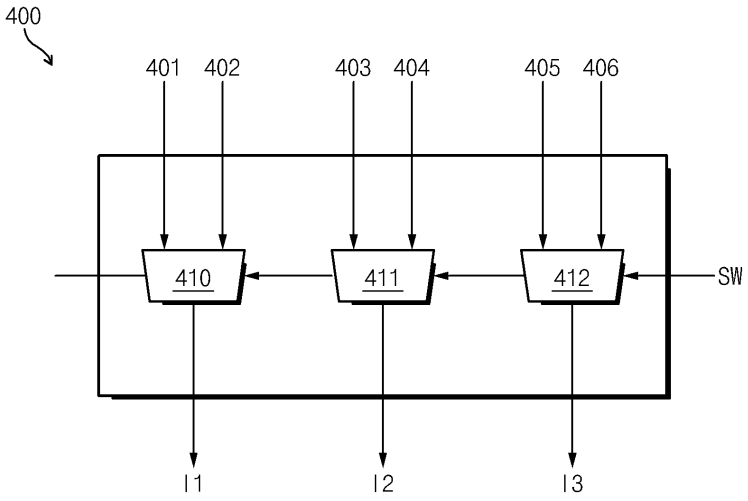
도면10


SN[n]
SN[n-1]
SN[n-2]
SN[n-3]
SN[2]
SN[1]
SN[0]
FF
FA
HA
HCP n+2
HCP n
HCP n-1
HCP n-2
HCP 4
HCP 3
HCP 2
FCP 1
FCP 0
171a
MM[n+1]
PP[n+1]
MM[n]
PP[n]
MM[n-1]
PP[n-1]
MM[4]
PP[4]
MM[3]
PP[3]
MM[2]
PP[2]
CW[1]
MM[1]
PP[1]
CW[0]
MM[0]
PP[0]
175
174
173
172

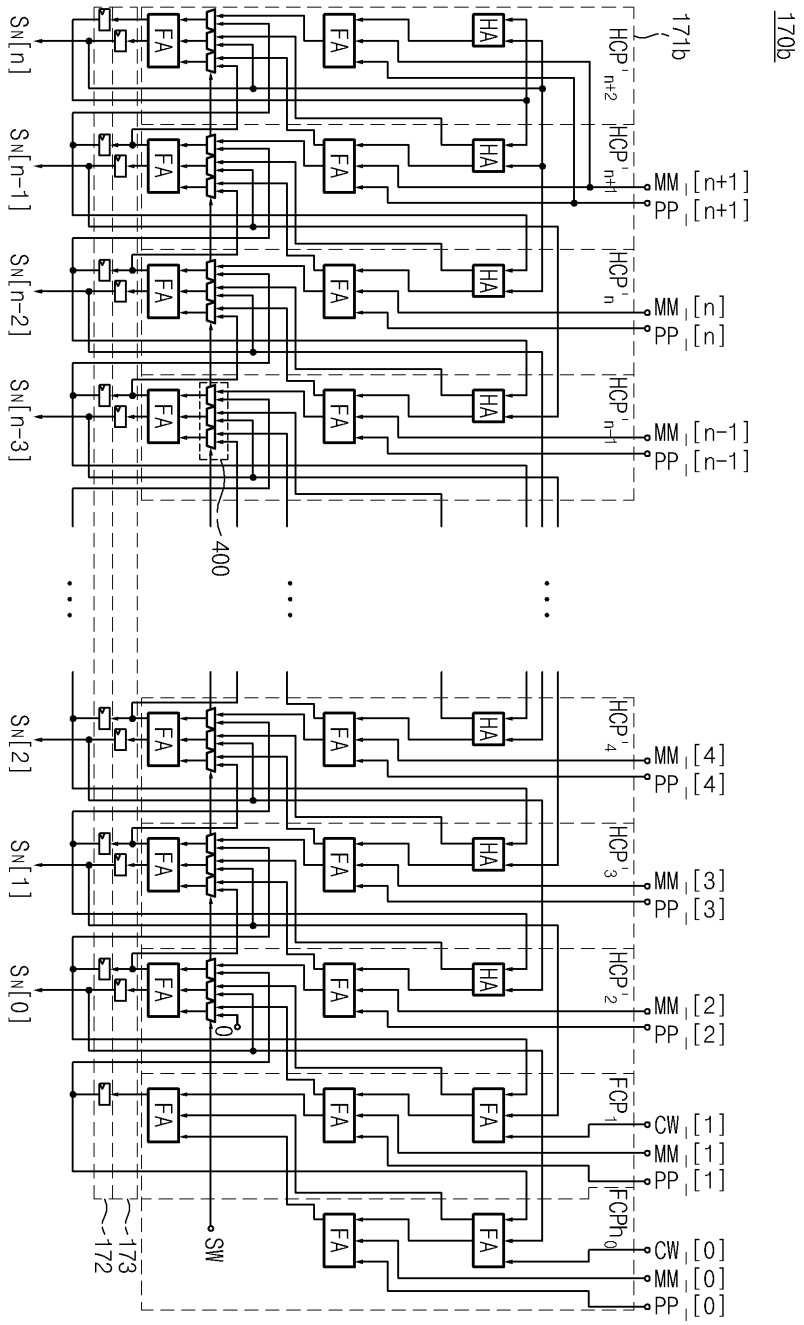
도면11


170b
171b
SW
FF
FA3
MXG
HCP'
FCP
FCPh
MM[n+1]
PP[n+1]
MM[n]
PP[n]
MM[3]
PP[3]
MM[2]
PP[2]
CW[1]
MM[1]
PP[1]
CW[0]
MM[0]
PP[0]
SN[n]
SN[n-1]
SN[n-2]
SN[1]
SN[0]
400
303
302
301
172
173

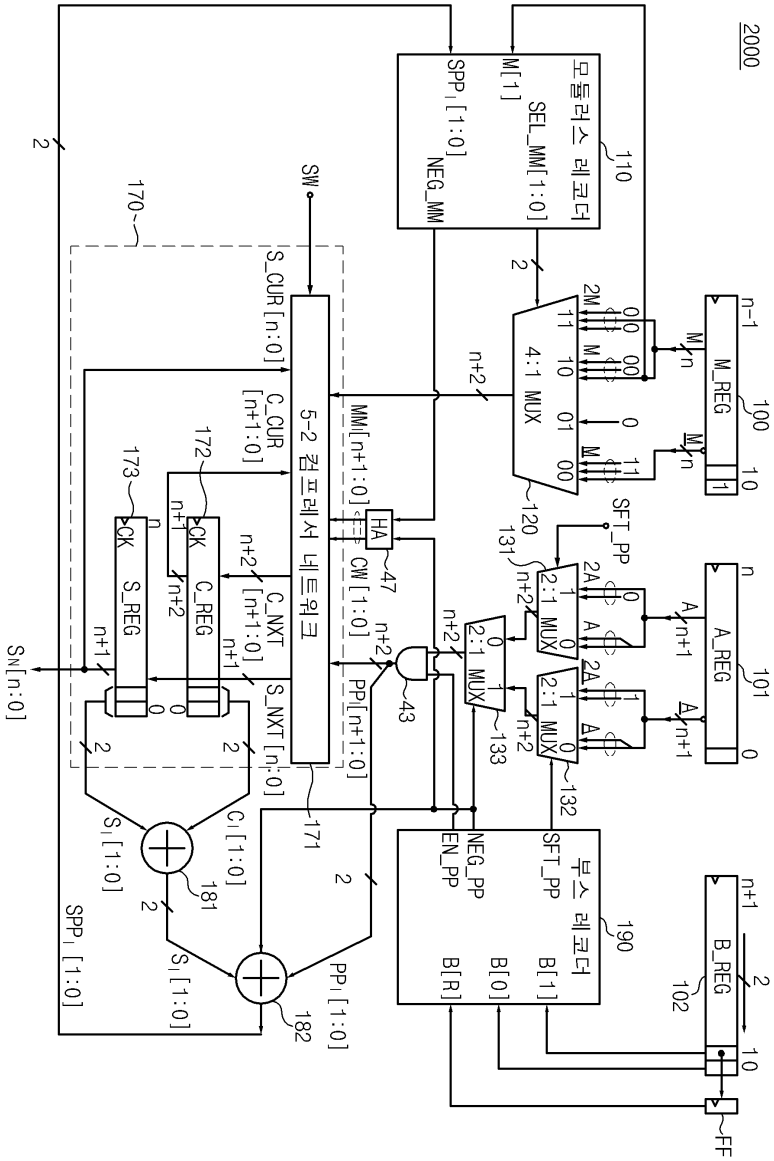
도면12



도면13


170b
171b
SN[n]
SN[n-1]
SN[n-2]
SN[n-3]
SN[2]
SN[1]
SN[0]
FA
HA
HCP'n+2
HCP'n+1
HCP'n
HCP'n-1
HCP'4
HCP'3
HCP'2
FCP1
FCP0
MM[n+1]
PP[n+1]
MM[n]
PP[n]
MM[n-1]
PP[n-1]
MM[4]
PP[4]
MM[3]
PP[3]
MM[2]
PP[2]
CW[1]
MM[1]
PP[1]
CW[0]
MM[0]
PP[0]
400
SW
172
173

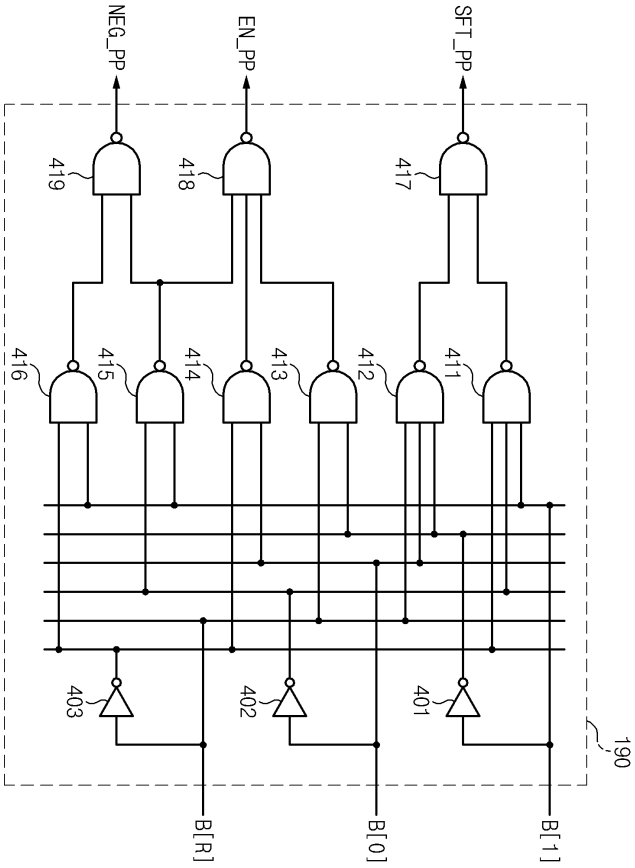
도면14


2000
모듈러스 레코더
110
SEL_MM[1:0]
M[1]
NEG_MM
SPP1[1:0]
4:1 MUX
120
M_REG
100
A_REG
101
B_REG
102
SFT_PP
131
2:1 MUX
132
133
부스 레코더
190
NEG_PP
EN_PP
B[0]
B[1]
B[R]
FF
SW
5-2 컴프레서 네트워크
171
S_CUR[n:0]
C_CUR[n+1:0]
S_NXT[n:0]
C_NXT[n+1:0]
S_REG
C_REG
172
173
170
MM[n+1:0]
CW[1:0]
PP[n+1:0]
HA
47
43
SN[n:0]
S1[1:0]
C1[1:0]
181
182
PP1[1:0]
SPP1[1:0]

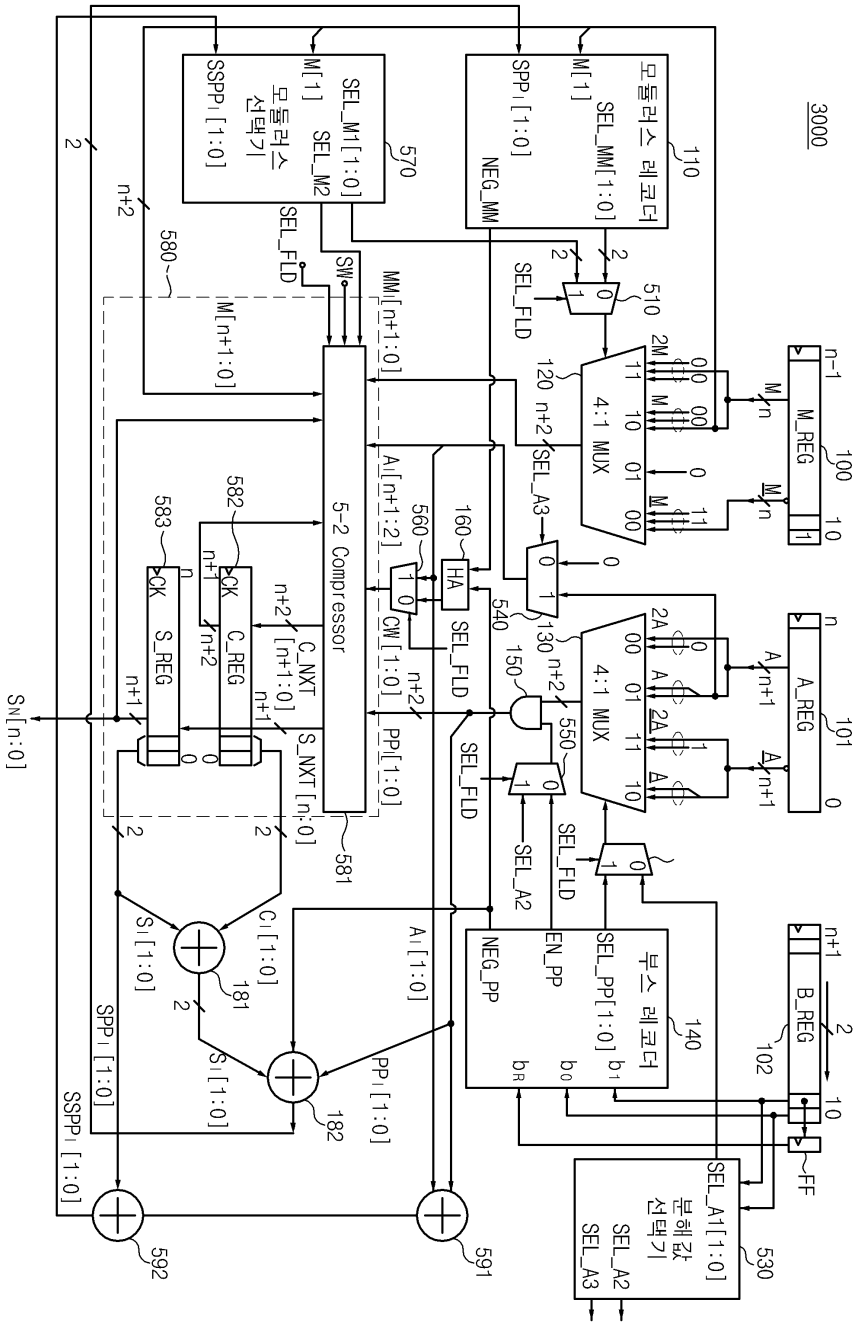
도면15

부스 레코더(190)의 입력			부스 레코더(190)의 출력			선택된 $PP_i[n+1:0]$
B[1]	B[0]	B[R]	SFT_PP	EN_PP	NEG_PP	
0	0	0	0	0	0	0
0	0	1	0	1	0	A
0	1	0	0	1	0	A
0	1	1	1	1	0	2A
1	0	0	1	1	1	$\overline{2A}$
1	0	1	0	1	1	$\overline{A}$
1	1	0	0	1	1	$\overline{A}$
1	1	1	0	0	0	0

도면16


NEG_PP
EN_PP
SFT_PP
419
418
417
416
415
414
413
412
411
403
402
401
190
B[R]
B[0]
B[1]

도면17


3000
모듈러스 레코더
SEL_MM[1:0]
M[1]
SPP[1:0]
NEG_MM
110
모듈러스 선택기
SEL_M1
SEL_M2
M[1]
SSPP[1:0]
570
SEL_FLD
SW
580
M[n+1:0]
MM[n+1:0]
510
120
4:1 MUX
M_REG
100
A_REG
101
B_REG
102
FF
부스 레코더
SEL_PP[1:0]
NEG_PP
EN_PP
140
SEL_A3
SEL_A2
SEL_A1[1:0]
분해값 선택기
530
160
HA
560
CW[1:0]
540
130
150
550
5-2 Compressor
581
582
583
C_REG
S_REG
C_NXT
S_NXT
PP[1:0]
A[n+1:2]
181
182
591
592
SN[n:0]

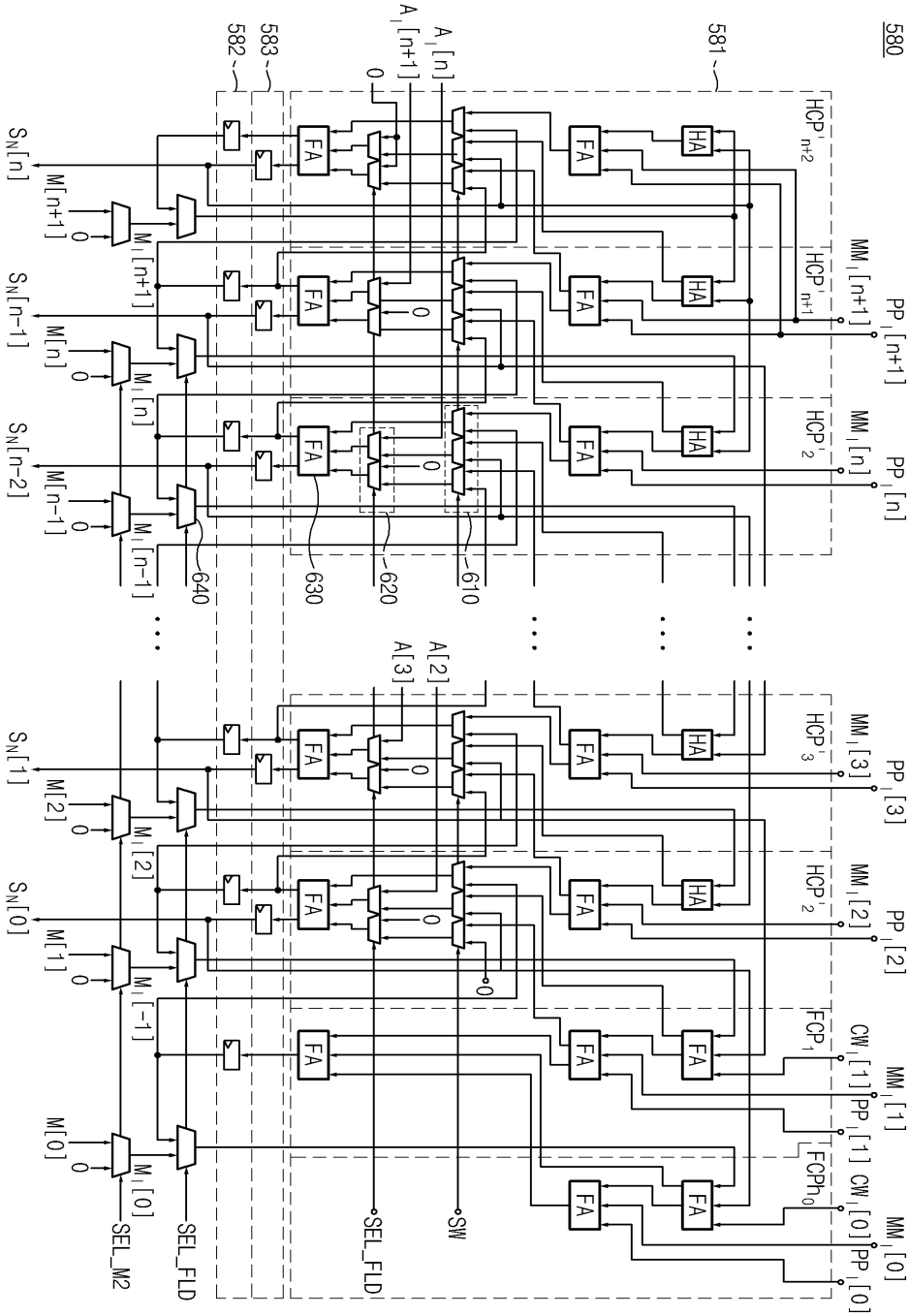
도면18

분해값 선택기(530)의 입력		분해값 선택기(530)의 출력			선택된 제1 분해값 $PP_i[n+1:0]$	선택된 제2 분해값 $A_i[n+1:0]$
B[0]	B[1]	SEL_A1[1:0]	SEL_A2	SEL_A3		
0	0	Don't care	0	0	0	0
0	1	01	1	0	A	0
1	0	00	1	0	2A	0
1	1	00	1	1	2A	A

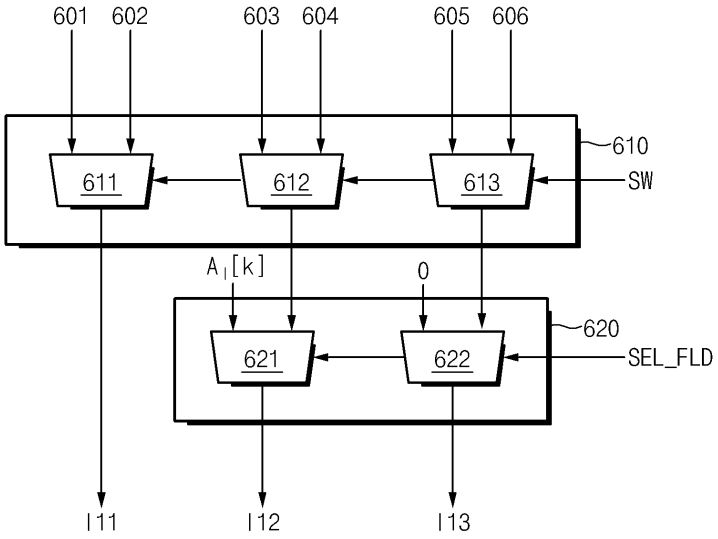
도면19

모듈러스 선택기(570)의 입력		q_i	모듈러스 선택기(570)의 출력		선택된 제1 분해값 $MM_i[n+1:0]$	선택된 제2 분해값 $M_i[n+1:0]$
$SSPP_i[1:0]$	M[1]		SEL_M1[1:0]	SEL_M2		
00	0	00	01	0	0	0
01	0	01	10	0	M	0
10	0	10	11	0	2M	0
11	0	11	11	1	2M	M
00	1	00	01	0	0	0
01	1	11	11	1	2M	M
10	1	10	11	0	2M	0
11	1	01	10	0	M	0

도면20


580
581
582
583
HCP'n+2
HCP'n+1
HCP'2
HCP'3
FCP1
FCPh0
MM1[n+1]
MM1[n]
MM1[3]
MM1[2]
MM1[1]
MM1[0]
PP1[n+1]
PP1[n]
PP1[3]
PP1[2]
PP1[1]
PP1[0]
CW1[1]
CW1[0]
A1[n+1]
A1[n]
A[3]
A[2]
SN[n]
SN[n-1]
SN[n-2]
SN[1]
SN[0]
M[n+1]
M[n]
M[n-1]
M[2]
M[1]
M[0]
M1[n+1]
M1[n]
M1[n-1]
M1[2]
M1[1]
M1[-1]
M1[0]
SEL_M2
SEL_FLD
SW
610
620
630
640
FA
HA

도면21


601
602
603
604
605
606
610
611
612
613
SW
A_I[k]
0
620
621
622
SEL_FLD
I11
I12
I13