(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년03월08일
(11) 등록번호 10-0946256
(24) 등록일자 2010년03월02일

(51) Int. Cl.
G06F 7/52 (2006.01)
(21) 출원번호 10-2007-0137931
(22) 출원일자 2007년12월26일
심사청구일자 2007년12월26일
(65) 공개번호 10-2009-0070061
(43) 공개일자 2009년07월01일
(56) 선행기술조사문헌
논문1:한국정보과학회
KR1020050088506 A

(73) 특허권자
대구대학교 산학협력단
경북 경산시 진량읍 내리리 15 대구대학교 내
(72) 발명자
김태호
대구 달서구 진천동 청구아파트 101동 607호
김창훈
경남 통영시 평림동 851번지
홍춘표
대구 달서구 상인동 보성은하아파트 106동 506호
(74) 대리인
이덕록

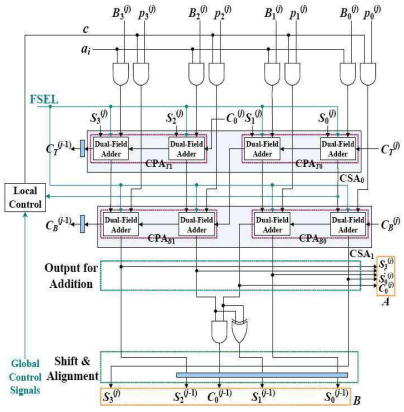
전체 청구항 수 : 총 1 항 심사관 : 강윤석

(54) 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의확장성있는 몽고매리 곱셈기

(57) 요 약

본 발명은 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성있는 몽고매리 곱셈기에 관한 것으로, 듀얼필드상의 확장성있는 몽고매리 곱셈기는 파이프라인 구조의 복수개의 처리기(PE:Processing Element); 및 상기 각 복수개의 처리기와 연결된 m-bit 우측 시프트 레지스터를 포함하여 이루어지고, 곱셈 또는 덧셈 연산은 컨트롤 신호(Add/Mult SEL)에 의해 수행되되, Add/Mult SEL이 0이면 덧셈 연산을 수행하고 n번째 PE에서 결과값을 얻을 수 있고, Add/Mult SEL이 1이면 큐로부터 곱셈결과를 얻을 수 있도록 하며, 각 PE는 m-비트의 k-비트 쉬프트 레지스터로부터 오퍼랜드 A의 각 비트($a_i,...,a_{i+k-1}$)를 가져오며(여기서 k는 처리기의 수를 나타냄), S와 C를 저장하기 위한 큐를 사용하고, 큐의 최대 길이는 메모리에 저장되는 워드의 최대 개수와 파이프라인 단계 수에 의존되며 수학식 5와 같이 계산되는 것을 특징으로 한다.

대 표 도 - 도9



이 발명을 지원한 국가연구개발사업

과제고유번호	07-기반-10
부처명	정보통신연구진흥원
연구사업명	IT특화연구소 : 유비쿼터스 신기술 연구센터 설립 및 운영
연구과제명	IT특화연구소 : 유비쿼터스 신기술 연구센터 설립 및 운영
주관기관	대구대학교
연구기간	2007년 5월 1일~2007년 12월 31일

특허청구의 범위

청구항 1

다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성 있는 몽고매리 곱셈기에 있어서,

파이프라인 구조의 복수개의 처리기(PE:Processing Element); 및 상기 각 복수개의 처리기와 연결된 m-bit 우측 시프트 레지스터를 포함하여 이루어지고,

곱셈 또는 덧셈 연산은 컨트롤 신호(Add/Mult SEL)에 의해 수행되되, 컨트롤 신호(Add/Mult SEL)가 0이면 덧셈 연산을 수행하여 n번째 PE에서 결과값을 얻을 수 있고, 컨트롤 신호(Add/Mult SEL)가 1이면 큐로부터 곱셈결과를 얻을 수 있도록 하며, 각 PE는 m-비트의 k-비트 쉬프트 레지스터로부터 오퍼랜드 A의 각 비트($a_i,..,a_{i+k-1}$)를 가져오며(여기서 k는 처리기의 수를 나타냄), S와 C를 저장하기 위한 큐를 사용하고, 큐의 최대 길이는 메모리에 저장되는 워드의 최대 개수와 파이프라인 단계 수에 의존되며 아래 수학식 5와 같이 계산되며,

[수학식 5]

$$Q_{\max} = \begin{cases} e-2 \cdot (k-1) & \text{if}\,(e+1) > 2k, \\ 0 & \text{otherwise.} \end{cases}$$

여기서 e는 워드의 개수를 나타낸다.

상기 캐리 세이브 가산기(Carry Save Adder:MP-CSA)는 워드 크기가 w-비트이면 n=[w/b]개의 CPA로 이루어지며,

b는 하나의 CPA를 구성하는 1-비트 듀얼 필드 가산기(Dual Field Adder:DFA)의 개수이고,

w=4이고 b=2인 데이터 패스(부)는,

각 4-비트 CSA는 두 개의 2-비트 CPA로 나누어지며 하나의 CPA는 두 개의 1-비트 DFA로 구성되고,

DFA는 캐리를 가지는 GF(p)상의 덧셈 연산과 캐리를 가지지 않는 GF(2^m)상의 덧셈 연산을 모두 수행하며,

FSEL은 유한체 GF(p)와 GF(2^m)을 선택하되, FSEL이 1이면 DFA는 GF(p)상의 곱셈 연산을, FSEL이 0이면 GF(2^m)상의 덧셈 연산을 수행하고,

데이터 패스는 $S^{(0)}+a_i \bullet B^{(0)}$의 최하위 비트를 로컬 컨트롤로 사용하며, 이 비트는 p의 가산여부를 결정하는 컨트롤 신호를 생성하는데 사용되고,

데이터 패스는 m-비트 덧셈값과 n-비트 캐리를 저장하기 위해 (m+n)-비트 레지스터를 가지며, 덧셈 연산을 수행하기 위한 CSA_1의 출력(A) 또는 곱셈 연산을 위한 결과값(B)을 출력하는 것을 특징으로 하는 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성있는 몽고매리 곱셈기.

청구항 2

삭제

청구항 3

삭제

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성 있는 몽고매리 곱셈기에 관한 것으로, 더욱 상세하게는 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성 있는 몽고매리 곱셈기에 관한 것이다.

[0002] 삭제

배 경 기 술

[0003] 알에스에이(RSA)(J.-J. Quisquater and C. Couvreur, "Fast Decipherment Algorithm for RSA Public-key Cryptosystem," *IEE Electronics Letters*, Vol. 18, No. 21, pp. 905-907, 1982), Diffie-Hellman 키교환 알고리즘(W. Diffie and M.E. Helman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. 22, pp. 644-654, 1976), 타원곡선 암호시스템(Elliptic Curve Cryptosystems: ECC)(N. Koblits, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, Vol. 48, No. 177, pp. 203-209, 1987)과 같은 암호 응용에서 모듈러 곱셈 및 지수승은 중요한 연산이다. 특히 모듈러 곱셈은 모듈러 지수 및 역원의 기본 연산으로 현재까지 많은 연구결과가 발표되었다. 모듈러 곱셈을 위해 고전적인 방법, 몽고매리(Montgomery) 알고리즘(P.L. Montgomery, "Modular Multiplication without Trial Division," *Math. Computation*, Vol. 44, pp. 519-521, 1985), Barret 알고리즘(P. Barrett, "Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor," *Lecture Notes in Computer Science*, Vol. 263, pp. 311-323, 1987) 등이 사용 되었다. 모듈러 곱셈 알고리즘 중, 몽고매리(Montgomery) 곱셈 방법은 내부 연산이 규칙적일 뿐만 아니라 나눗셈 연산은 쉬프트 연산으로 대체되기 때문에 하드웨어 구현에 매우 적합하다. 이러한 장점 때문에 다양한 형태의 변형된 몽고매리 알고리즘과 그에 따른 하드웨어 구현 방법이 연구되어 왔다.

[0004] 확장성 있는 몽고매리 곱셈기 설계 방법과 하드웨어 구현 방법은 다양하게 소개된 바 있다(E. Savas, A.F. Tenca, and .K. Ko, "A Scalable and Unified Multiplier Architecture for Finite Fields and ," *Lecture Notes in Computer Science*, Vol. 1965, pp. 277-292, 2000, A. Tenca and .K. Ko, "A Scalable Architecture for Modular Multiplication Based on Montgomery's Algorithm," *IEEE Trans. on Computers*, Vol. 52, No. 9, pp. 1215-1221, 2003, A.F. Tenca and .K. Ko, "A Scalable Architecture for Montgomery Multiplication," *Lecture Notes in Computer Science*, Vol. 1717, pp. 94-108, 1999). 확장성 있는 구조는 입력값의 크기에 제한을 받지 않고 고정된 크기의 회로를 이용하여 연산을 수행한다. 데이터를 일정한 크기의 워드 단위로 나눈 후 워드 단위로 처리 및 전송한다. 데이터 크기가 m-비트이고 워드의 크기가 w-비트이면 워드의 개수는 e=[m/w] 개 이다. 확장성 있는 구조는 워드의 크기가 커질수록 연산 시간을 단축할 수 있으나 하드웨어 복잡도가 증가한다. 그러나 면적 및 속도를 만족시키는 가장 적합한 워드 크기를 찾는다면 연산시간 및 하드웨어 복잡도에 있어 상충 관계를 개선할 수 있고, 확장성 있는 몽고매리 곱셈기의 연산시간 및 하드웨어 복잡도의 상충관계도 이미 분석된 바 있다.

[0005] 최초에 몽고매리(Montgomery)곱셈 알고리즘은 홀수 모듈러스와 함께 모듈러 곱셈 알고리즘을 수행하는 효율적인 방법으로 제안되었다. 만약 모듈러스가 소수이면 몽고매리 곱셈 알고리즘은 유한체 $GF(p)$상에서 매우 효율적인 곱셈 연산을 수행할 수 있으며, 다항식기저를 사용하고 기약다항식이 선택되면 유한체 $GF(2^m)$상에서도 곱셈 연산을 수행할 수 있다. 유한체 $GF(p)$상에서 몽고매리 곱셈 연산을 위해 캐리 전파 가산기(Carry Propagation Adder: CPA)를 이용하면 최대 처리 지연시간이 증가하고 비트수가 커지면 캐리 전파 문제가 발생한다. 이러한 문제를 해결하기 위해 기존에 제안된 몽고매리 곱셈기는 캐리 세이브 가산기(Carry Save Adder: CSA)를 이용한다. 그러나 CSA는 곱셈 결과가 합과 캐리로 구분되는 캐리 세이브(Carry Save: CS)형태이기 때문에 정확한 곱셈결과를 얻기 위해서는 추가적인 m-비트의 가산기 회로 또는 m 클럭 사이클이 필요하다(J.C. Ha and S.J. Moon, "A Design of Modular Multiplier Based on Multi-Precision Carry Save Adder," *Joint Workshop on Information Security and Cryptology (JWISC'2000)*, pp. 45-51, 2000). 최근 김 등(김대영, 이준용, "개선된 다정도 CSA에 기반한 모듈라 곱셈기 설계," 정보과학회논문지 : 시스템 및 이론, 제33권, 제34호, pp. 223-230, 2006)은 MP(Multi-Precision)-CSA에 기반한 $GF(p)$상의 효율적인 곱셈기를 제안하였다. MP-CSA는 CSA와 CPA를 결합한 형태로 캐리 전파 문제를 해결하는 동시에 결과값을 보정하기 위

한 클럭 사이클 수를 감소시킨다. 하지만 김 등의 구조는 m과 w에 대해 확장성을 제공하지 못하는 문제점이 있었다.

발명의 내용

해결 하고자하는 과제

[0006] 본 발명의 목적은 상기한 바와 같은 종래의 문제점을 개선하기 위하여 제안된 것으로, 기존에 제안된 구조에 비해 적은 플립플롭을 사용하며 추가회로를 필요로하지 않고 보정에 필요한 클럭 사이클 수를 감소시킬 수 있는 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성 있는 몽고매리 곱셈기를 제공함에 있다.

과제 해결수단

[0007] 상기한 바와 같은 목적을 달성하기 위한 본 발명의 바람직한 실시예에 따르면, 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성 있는 몽고매리 곱셈기는 파이프라인 구조의 복수개의 처리기(PE:Processing Element); 및 상기 각 복수개의 처리기와 연결된 m-bit 우측 시프트 레지스터를 포함하여 이루어지고, 곱셈 또는 덧셈 연산은 컨트롤 신호(Add/Mult SEL)에 의해 수행되되, Add/Mult SEL이 0이면 덧셈 연산을 수행하고 n번째 PE에서 결과값을 얻을 수 있고, Add/Mult SEL이 1이면 큐로부터 곱셈결과를 얻을 수 있도록 하며, 각 PE는 m-비트의 k-비트 쉬프트 레지스터로부터 오퍼랜드 A의 각 비트($a_i,...,a_{i+k-1}$)를 가져오며(여기서 k는 처리기의 수를 나타냄), S와 C를 저장하기 위한 큐를 사용하고, 큐의 최대 길이는 메모리에 저장되는 워드의 최대 개수와 파이프라인 단계 수에 의존되며 아래 수학식 5와 같이 계산되는 것을 특징으로 한다.

효 과

[0008] 이상 설명된 바와 같이, 본 발명에 따른 듀얼필드상의 확장성 있는 몽고매리 곱셈기에 의하면, 새로운 MP-CSA를 이용한 듀얼-필드(dual-field)상의 확장성 있는 몽고매리 곱셈기를 제안한다. 본 발명의 구조는 유한체 GF(p)와 GF(2^m)상의 곱셈 연산을 수행하며 기존에 제안된 유사한 구조에 비해 비교적 적은 플립플롭(Flip Flop: FF)을 사용한다. 또한 본 발명에 따른 회로는 Savas 등이 제안한 구조와 달리 결과값을 보정하기 위한 추가회로를 필요로 하지 않고 보정에 필요한 클럭 사이클 수를 감소시키는 효과가 있다. 더욱이 본 발명에 따른 곱셈기 회로는 덧셈을 위해 재사용될 수 있고 m과 w에 대해 높은 확장성을 가진다. 따라서 본 발명에서 제안한 구조는 암호응용을 위한 GF(p)와 GF(2^m)상의 곱셈기로서 매우 적합하게 되는 효과가 있다.

발명의 실시를 위한 구체적인 내용

[0009] 이하 본 발명에 따른 다정도 캐리 세이브 가산기를 이용한 듀얼필드상의 확장성 있는 몽고매리 곱셈기에 대하여 첨부도면을 참조하여 상세히 설명한다.

[0010] 도 1은 유한체 GF(p)상의 몽고매리 곱셈 알고리즘을 나타낸 것이고, 도 2는 유한체 GF(2^m)상의 몽고매리 곱셈 알고리즘을 나타낸 것이고, 도 3은 본 발명에 따른 다정도 캐리 세이브 가산기에 기반한 GF(p)상의 워드-레벨 몽고매리 곱셈 알고리즘을 나타낸 것이고, 도 4는 본 발명에 따른 다정도 캐리 세이브 가산기에 기반한 GF(2^m)상의 워드-레벨 몽고매리 곱셈 알고리즘을 나타낸 것이고, 도 5a는 확장성 있는 몽고매리 곱셈 알고리즘을 위한 자료의존 그래프를 나타낸 것이고, 도 5b는 파이프라인된 2개의 처리기(PE:Processing Element)를 이용한 4-비트 곱셈 연산(w=1,n=2) 과정을 도식적으로 나타낸 것이고,도 5c는 3개의 처리기를 이용한 도 5b와 동일한 계산과정을 도식적으로 나타낸 것이고, 도 6a는 워드-레벨 덧셈 연산을 위한 자료의존 그래프를 나타낸 것이고, 도 6b는 파이프 라인된 4개의 처리기를 이용한 6-비트 덧셈 연산과정(w=1,n=2)을 도식적으로 나타낸 것이고, 도 7은 본 발명에 따른 곱셈 및 덧셈 연산을 위한 파이프라인 구조의 다정도캐리 세이브 가산기를 이용한 듀얼필드상의 확장성 있는 몽고매리 곱셈기를 나타낸 것이고, 도 8은 도 7의 처리기의 블록 다이어그램을 나타낸 것이고, 도 9는 도 7의 처리기의 데이터 패스(부)를(w=4,b=2,n=2) 나타낸 것이다.

[0011] 도 1 내지 도 9를 참조하면, 본 발명에 따른 듀얼-필드(dual-field)상의 확장성 있는 몽고매리(Montgomery)곱셈기에 의하면, 유한체 GF(p)와 GF(2^m)상의 곱셈 연산을 수행한다. 본 발명에 따른 다정도 캐리 세이브 가산기는 두 개의 캐리 세이브 가산기로 구성되며, w-비트의 워드를 처리하기 위한 하나의 캐리 세이브 가산기는 n =[w/b]개의 캐리 전파 가산기로 이루어진다. 여기서 b는 하나의 CPA가 포함하는 듀얼-필드 가산기의 개수이다.

본 발명에 따른 몽고매리 곱셈기는 기존의 연구결과에 비해 거의 동일한 시간 복잡도를 가지지만 낮은 하드웨어 복잡도를 가진다. 뿐만 아니라 본 발명에 따른 연산기는 기존의 연구와 달리 연산의 종료시 정확한 모듈러 곱셈의 결과를 출력한다. 더욱이 본 발명에 따른 회로는 m과 w에 대해 높은 확장성을 가진다. 따라서 본 발명에 따른 구조는 암호응용을 위한 GF(p)와 $GF(2^m)$상의 곱셈기로서 매우 적합하다 할 수 있다.

[0012] 몽고매리(Mongomery) 곱셈 알고리즘

[0013] 정수 A,B와 모듈러스 p가 주어졌을 때, 몽고매리 곱셈 알고리즘은 아래 수학식 1에서,

수학식 1

$$C = A \cdot B \cdot R^{-1} \bmod p$$

[0014]

[0015] 를 계산한다. 여기서 , $R=2^m$, A,B<p<R이고 p는 $m=[\log_2 p]$-비트이다. 본 발명에서는 p가 소수라고 가정한다. 상기 수학식 1을 바탕으로 도 1의 [알고리즘 1]과 같은 $GF(p)$상의 몽고매리 곱셈 알고리즘을 얻을 수 있다.

[0016] 도 1에서, 만약 최종 결과값이 p보다 크면 단계 6과 같이 뺄셈 연산이 수행되어야 한다.

[0017] 바이너리 확장 필드 $GF(2^m)$은 $GF(2)$상의 차수 (m-1)인 다항식으로 필드 원소가 표현된다. 두 다항식 A(x),B(x)가 주어졌을 때, 몽고매리 곱셈 연산은 아래 수학식 2와 같이 정의된다.

수학식 2

$$C(x) = A(x) \cdot B(x) \cdot x^{-m} \bmod p(x)$$

[0018]

[0019] 여기서 $C(x) \in GF(2^m)$이고, p(x)는 기약다항식이다. 도 1의 [알고리즘 1]의 $R=2^m$은 x^m으로 대체된다. 상기 수학식 2를 바탕으로 도2의 [알고리즘 2]와 같은 $GF(2^m)$상의 몽고매리 곱셈 알고리즘을 얻을 수 있다.

[0020] 도 2의 [알고리즘 2]에서 0^m은 m-비트가 모두 0인 상태를 나타낸다. 도 1의 [알고리즘 1]의 단계 6에서 수행하는 추가적인 뺄셈 연산은 $GF(2^m)$상의 ,몽고매리 곱셈 알고리즘에서는 생략한다. 또한 $GF(2^m)$상의 연산은 캐리가 발생하지 않기 때문에 덧셈 연산은 단순한 비트별 XOR 연산으로 대체할 수 있으며 기호 $\oplus$로 나타낸다.

[0021] **다정도 CSA에 기반한 워드-레벨 몽고매리(Montgomery) 곱셈 알고리즘**

[0022] 워드-레벨 구조는 데이터를 일정한 크기의 워드 단위로 나눈 후, 워드 단위로 처리 및 전송한다. 데이터 크기가 m-비트이고 워드 크기가 w-비트이면 워드-레벨 몽고매리 곱셈 연산은 e=[(m+1)/w]개의 워드로 나누어진다.

[0023] **다정도 CSA에 기반한 몽고매리(Montgomery) 곱셈기**

[0024] CPA에 기반한 몽고매리 곱셈기는 넌-리던던트(non-redundant) 형식의 결과값을 바로 얻어올 수 있다는 장점이 있지만 비트수가 커지면 캐리 전파 문제가 발생한다. 또한, 공개키 암호 응용은 2048-비트 이상의 키를 요구하기 때문에 캐리 전파 지연은 심각한 문제가 될 수 있다. 반면에 CSA는 1-비트 전가산기(Full Adder: FA)와 동일

한 캐리 전파 지연을 가지기 때문에 캐리 전파 문제가 발생하지 않고 고속의 연산이 가능하다. 그러나 내부 곱셈 결과가 CS 형태로 저장되기 때문에 non-redundant 형태의 결과값을 얻기 위한 추가적인 연산이 요구된다. 일반적으로 이러한 문제를 해결하기 위해 추가적인 회로 또는 클럭 사이클이 사용된다(J.C. Ha and S.J. Moon, "A Design of Modular Multiplier Based on Multi-Precision Carry Save Adder," *Joint Workshop on Information Security and Cryptology (JWISC'2000)*, pp. 45-51, 2000). 최근 CSA와 CPA를 결합한 방식을 사용하는 하이브리드 형태의 MP-CSA 구조가 김 등(김대영, 이준용, "개선된 다정도 CSA에 기반한 모듈라 곱셈기 설계," 정보과학회논문지 : 시스템 및 이론, 제33권, 제34호, pp. 223-230, 2006)에 의해 제안되었는데, 그 구조의 몽고매리 곱셈기는 두 개의 CSA로 구성되며 데이터 크기가 m-비트이면 하나의 CSA는 n=[m/b]개의 CPA로 나누어진다. 여기서 b는 하나의 CPA가 포함하는 FA의 개수이다. 김 등에 의해 제안된 몽고매리 곱셈기는 n 클럭 사이클의 추가로 결과값을 보정한다.

[0025] **GF(p)상의 워드-레벨 몽고매리(Montgomery) 곱셈 알고리즘**

[0026] 본 발명에서는 새로운 GF(p)상의 워드-레벨 몽고매리 곱셈 알고리즘을 제안한다. 본 발명에 따른 곱셈 알고리즘은 도 3에 도시된 [알고리즘 3]과 같다. 두 개의 오퍼랜드 B(피승수), A(승수)와 모듈러스 p가 주어졌을 때, 워드-레벨 몽고매리 곱셈 알고리즘은 $ABR^{-1} \bmod p$를 계산한다. 여기서 B는 워드단위로, A는 비트단위로 읽어온다. 도 3의 [알고리즘 3]에서 w-비트인 B와 p는 b-비트의 CPA로 나누어지며 곱셈 연산에서 사용되는 벡터 A, B, p는 다음과 수학식 3과 같이 나타낸다.

수학식 3

$$A = (a_{m-1}, \cdots, a_1, a_0),$$

$$B = (B^{(e-1)(b-1)}, \cdots, B^{(e-1)(1)}, B^{(e-1)(0)}, \cdots, B^{(0)(b-1)}, \cdots, B^{(0)(1)}, B^{(0)(0)}),$$

$$p = (p^{(e-1)(b-1)}, \cdots, p^{(e-1)(1)}, p^{(e-1)(0)}, \cdots, p^{(0)(b-1)}, \cdots, p^{(0)(1)}, p^{(0)(0)})$$

[0027]

[0028] 여기서 워드와 CPA 인덱스는 위첨자로 표시되고 비트는 아래 첨자로 표시한다. 예를 들어, 벡터 B에서 j워드의 k번째 CPA의 i번째 비트는 $B_i^{(j)(k)}$로 나타낼 수 있다. 벡터 B의 i에서 j까지의 비트는 $B_{j\cdots i}$로 나타낸다(j>i). 그리고 (x | y)는 두 비트 시퀀스의 결합을 나타낸다.

[0029] 본 발명에 따른 곱셈 알고리즘은 A의 각 비트에 대해 TS, B, p의 부분합을 계산한다. B가 완전히 읽혀졌을 때, A의 다음 비트를 읽어온 후 계산을 반복한다. 도 3의 [알고리즘 3]은 내부 결과를 저장하기 위해서 CS 형태를 사용한다. 덧셈 결과는 m-비트의 $TS^{(j)}$와 n-비트의 $TC^{(j)}$에 저장된다. m번의 반복을 수행한 후 CS 형태의 곱셈 결과를 얻을 수 있으며 non-redundant 형태의 결과를 얻기 위한 n번의 추가적인 덧셈 연산을 수행한다(단계 3). 여기서 입력값 a_i와 B는 모두 0으로 인가한다. 경우에 따라서 m번의 반복이 끝난 후 출력되는 결과값이 모듈러스 p보다 클 수 있으며 2번의 반복수행을 통해 최종결과가 p미만이 되도록 조정할 수 있다(T. Blum and C. Paar, "Montgomery modular exponentiation on reconfigurable hardware," *in Proc. 14th IEEE Symp. on Computer Arithmetic*, pp. 70-77, 1999). 본 발명에 따른 곱셈 알고리즘은 n번의 추가 클럭을 수행하기 때문에 n이 2보다 클 경우 뺄셈 연산을 제거할 수 있다.

[0030] **다정도 CSA에 기반한 상의 워드-레벨 몽고매리 곱셈 알고리즘**

[0031] 도 4의 [알고리즘 4]는 $GF(2^m)$상의 워드레벨 곱셈 알고리즘을 나타낸다. $GF(2^m)$상의 연산은 캐리가 발생하지 않기 때문에 내부 덧셈 연산은 비트별 XOR 연산으로 대체할 수 있다. $GF(2^m)$상의 기약다항식은 (m+1)-비트이기 때문에 인덱스 i는 0에서 m까지 수행한다.

[0032] 몽고매리 **곱셈 알고리즘의 병행성**

[0033] **다정도 CSA에 기반한 워드-레벨 몽고매리 곱셈기**

[0034] 워드-레벨 몽고매리 곱셈 알고리즘에서 i번째 반복의 j=0, j=1의 반복이 끝나면 (i+1)번째 반복이 즉시 수행된다. 이와 같은 계산을 나타내는 자료의존 그래프는 도 5a와 같다. 태스크 A, B는 기본적으로 다음과 같은 동작을 수행한다. 1) TS, $a_j \bullet B$,p 의 각 워드에 대한 덧셈 연산(p의 가산 여부는 TS의 최하위 비트에 의해 결정됨), 2) TS워드의 1-비트 우측 쉬프트 연산. 쉬프트된 $TS^{(j-1)}$의 생성은 $TS^{(j)}$의 최하위 비트가 계산된 후에 가능하다 (A. Tenca and .K. Ko, "A Scalable Architecture for Modular Multiplication Based on Montgomery's Algorithm," *IEEE Trans. on Computers*, Vol. 52, No. 9, pp. 1215-1221, 2003). 태스크 A는 2가지 동작에 추가적으로 $a_i \bullet B^{(0)}+TS^{(0)}$의 덧셈 결과로부터 TS의 최하위 비트를 저장한다(도 3의 [알고리즘 3]의 단계 7). 저장된 비트는 동일한 오퍼랜드의 다음 워드에 대해 p의 가산 여부를 결정하는데 사용된다. 자료의존 그래프는 한 개의 열에 대해 (e+1)개의 태스크를 가진다. 각 열의 태스크는 다른 처리기(Processing Element: PE)에서 계산되며 하나의 PE에서 생성된 데이터는 파이프라인 형태로 구성된 다음 PE로 전달한다. 각 태스크는 한 클럭 사이클에 계산된다. 본 발명에 따른 구조는 GF(p)상의 몽고매리 곱셈을 위해 t=[(m+n)/k]커널 사이클이 필요하다. 여기서 k는 파이프라인상의 PE 개수이다. 반면에 $GF(2^m)$상의 몽고매리 곱셈은 t=[m/k]커널 사이클이 필요하다.

[0035] 도 5b는 두 개의 PE를 사용하는 4-비트 곱셈 연산을 나타낸다. 여기서 w=1이고, n=2이다. 이 경우에 몽고매리 곱셈 연산은 3 커널 사이클을 요구한다. 회색 상자는 non-redundant 형태의 결과값을 출력하기 위한 추가적인 반복을 나타낸다. 도 5c는 도 5b와 동일한 계산에서 3개의 PE를 사용한 경우를 나타낸다. 이 경우, 추가적인 번의 반복에 관계없이 2 커널 사이클에 동작한다. 제안된 몽고매리 곱셈 연산의 전체 수행 시간은 아래 수학식 4와 같다.

수학식 4

$$\text{클럭사이클} = \begin{cases} t(2k+1)+e-2 & \text{if } (e+1) \le 2k, \\ t(e+1)+2(k-1) & \text{otherwise,} \end{cases}$$

[0036]

[0037] 첫 번째 계산식은 주어진 워드의 개수보다 PE의 개수가 더 많은 경우로 1 커널 사이클에 결과값을 출력하고, 두 번째 계산식은 파이프라인내의 PE 개수가 워드의 개수보다 적은 경우로 결과값을 출력하기 위해 2 커널 사이클 이상을 요구한다.

[0038] **다정도 CSA에 기반한 워드-레벨 가산기**

[0039] 본 발명에 따른 구조는 곱셈기 회로를 재사용해서 워드-레벨 덧셈 연산을 수행한다. GF(p)상의 덧셈 연산은 CS 형태의 결과를 출력하기 때문에 non-redundant 형태로 보정하기 위한 추가연산이 필요하다. 도 6a는 워드-레벨 덧셈 연산을 위한 자료의존 그래프를 나타낸다. 각 열은 e=[m/w]개의 태스크를 가지며 구분된 PE에 의해 계산을 수행한다. 첫 번째 PE는 오퍼랜드 A, B를 입력 받아서 CS 형태의 덧셈 결과를 출력한다. 다음 PE는 non-redundant 형태의 결과를 위해 추가적인 덧셈 연산을 수행한다. 여기서 a_i, B, p는 모두 0이 인가된다.

[0040] 도 6b는 4개의 PE를 사용한 6-비트 덧셈 연산을 나타낸다. 여기서 w=1, n=2이다. 회색 상자는 덧셈 연산을 위한 태스크를 나타내며, 두 번째 PE로부터 non-redundant 형태의 덧셈 연산 결과를 얻을 수 있다. 제안한 워드-레벨 가산기는 GF(p)와 $GF(2^m)$상의 덧셈 연산 결과를 (e+n-1)클럭 사이클 후에 출력한다.

[0041] **확장성 있는 몽고매리 곱셈기**

[0042] 도 7은 파이프라인으로 구성된 확장성 있는 구조를 나타낸다. 파이프라인 구조는 커널로 불리어지며 k개의 PE로 구성된다. 파이프라인의 각 PE는 전달 받은 워드를 다음 PE로 전달한다. 1-비트 입력 a_i와 n-비트 내부 캐리 $C^{(j)}$를 제외한 모든 경로는 w-비트로 구성된다. 또한 데이터 B, p,S 는 커널에 의해 워드단위로 처리한다. 회색 상자는 레지스터를 나타낸다.

[0043] **워드-레벨 곱셈 및 덧셈 연산**

[0044] 본 발명에 따른 구조는 워드-레벨 몽고매리 곱셈 연산을 수행하며 추가적인 가산기 없이 워드-레벨 덧셈 연산을 수행한다. 이와 같은 연산을 위해 멀티플렉서와 컨트롤 신호를 추가한다. 곱셈 또는 덧셈 연산은 컨트롤 신호(Add/Mult SEL)에 의해 수행된다. Add/Mult SEL이 0이면 덧셈 연산을 수행하고 n번째 PE에서 결과값을 얻을 수 있다. 반면에 Add/Mult SEL이 1이면 큐로부터 곱셈결과를 얻을 수 있다. 각 PE는 m-비트의 k-비트 쉬프트 레지스터로부터 오퍼랜드 A의 각 비트($a_i,...,a_{i+k-1}$)를 가져온다. S와 C를 저장하기 위한 큐를 사용하고, 큐의 최대 길이는 메모리에 저장되는 워드의 최대 개수와 파이프라인 단계 수에 의존되며 아래 수학식 5와 같이 계산한다.

수학식 5

$$Q_{\max} = \begin{cases} e-2\cdot(k-1) & \text{if}\,(e+1) > 2k, \\ 0 & \text{otherwise.} \end{cases}$$

[0045]

여기서 e는 워드의 개수를 나타낸다.

[0046] **처리기 구조**

[0047] 도 8은 PE의 블록 다이어그램을 나타낸다. 데이터 패스는 파이프라인의 이전단계로부터 워드 $S^{(j)}$, $C^{(j)}$, $B^{(j)}$, $p^{(j)}$를 전달 받아서 새로운 워드 $S^{(j-1)}$, $C^{(j-1)}$을 계산한다. 입력 B와 p를 지연시키는 것을 출력으로 $B^{(j-1)}$, $p^{(j-1)}$,$S^{(j-1)}$, $C^{(j-1)}$을 출력하기 위해서이다. 즉, 하나의 PE가 j번째 워드로 계산하면 다음 PE는 (j-2)번째 워드로 계산한다. 데이터 패스는 덧셈 또는 곱셈 연산 결과를 출력한다(덧셈: A, 곱셈: B). 본 발명의 구조는 곱셈 및 덧셈 연산을 모두 수행하기 위해 n개의 PE에 (w+n)-비트 멀티플렉서, w-비트 멀티플렉서, 컨트롤 신호를 추가한다.

[0048] 데이터 패스는 두 개의 CSA로 구성되며 워드 크기가 w-비트 이면 하나의 CSA는 n=[w/b]개의 CPA로 이루어진다. 여기서 b는 하나의 CPA를 구성하는 1-비트 DFA의 개수이다.

[0049] 도 9는 w=4이고 b=2인 데이터 패스를 나타낸다. 각 4-비트 CSA는 두 개의 2-비트 CPA로 나누어지며 하나의 CPA는 두 개의 1-비트 DFA로 구성된다. DFA는 캐리를 가지는 GF(p)상의 덧셈 연산과 캐리를 가지지 않는 $GF(2^m)$상의 덧셈 연산을 모두 수행한다. FSEL은 유한체 GF(p)와 $GF(2^m)$을 선택한다. FSEL이 1이면 DFA는 GF(p)상의 덧셈 연산을 FSEL이 0이면 $GF(2^m)$상의 덧셈 연산을 수행한다. 데이터 패스는 $S^{(0)}+a_i \bullet B^{(0)}$의 최하위 비트를 로컬 컨트롤로 사용하며, 이 비트는 p의 가산여부를 결정하는 컨트롤 신호를 생성하는데 사용된다.

[0050] 데이터 패스는 m-비트 덧셈값과 n-비트 캐리를 저장하기 위해 (m+n)-비트 레지스터를 가진다. 또한 덧셈 연산을 수행하기 위한 CSA_i의 출력(A) 또는 곱셈 연산을 위한 결과값(B)을 출력한다.

[0051] 본 발명에서는 새로운 다정도 MP-CSA를 이용한 듀얼-필드 상의 확장성 있는 몽고매리 곱셈기를 제안하였다. 본 발명에 따른 구조는 몽고매리 곱셈 연산을 위해 t 커널 사이클 후에 완전한 결과를 출력한다. 표 1에 본 발명에서 제안된 구조와 기존에 제안된 구조의 성능을 비교하였다. 종래 기술의 데이터 패스는 내부 결과를 저장하기 위해 2w-비트 FF을 가지는 반면에 본 발명에서 제안된 구조는 (w+n)-비트 FF을 가진다. 일반적으로 n은 w보다 작은 값을 사용하기 때문에 기존에 제안된 구조에 비해 비교적 적은 FF 개수를 가진다. 또한 본 발명의 구조는 CS 형태의 결과를 non-redundant 형태로 변환하기 위한 추가적인 회로를 요구하지 않는다. 또한 곱셈 및 덧셈 연산을 모두 수행한다. 표 2에 종래기술에 의해 제안된 구조에 대해 w, k, m 의 다양한 선택에 따른 커널 사이클, 클럭 사이클, FF 개수를 비교하였다. 표 2에서 n=4로 가정하며 ECC를 위해 NIST(NIST, Recommended elliptic curves for federal government use, May 1999. http://csrc.nist.gov/encryption)에서 권고하는 다섯 가지 GF(p), $m \in \{192,224,266,384,521\}$, 표준 필드 크기를 선택하였다. 표 2에 나타나듯이 두 개의 구조가 동일한 커널 사이클을 가지면 각 곱셈 연산은 동일한 클럭 사이클에 수행된다. 또한 w 또는 k가 증가할 수록 종래기술과 제안된 곱셈기의 FF개수 차이는 커진다. GF(p)상의 곱셈 연산과 달리 $GF(2^m)$상의 곱셈 연산은 두 개

의 구조가 동일한 시간 및 하드웨어 복잡도를 가진다.

표 1

표 1. Dual-field상의 확장성 있는 Montgomery 곱셈기 성능 비교

	Savas	도 7
Latency (kernel cycles)	$\lceil m/k \rceil$	$\lceil (m+n)/k \rceil$
Circuit Requirement (datapath)	DFA : $2w$ AND_2 : $2w$ XOR_2 : 0 FF : $2w$	DFA : $2w$ AND_2 : $2w+(n-1)$ XOR_2 : $n-1$ FF : $w+n$
Circuit Requirement (kernel)	FF : $8wk-4w$	FF : $6wk+2nk-3w-n$
Circuit Requirement (adder)	N/A	MUX_2 : $2n+2$
Results Form	Carry Save	non-redundant
Operation	Multiplication	Multiplication and Addition

AND_2 : 2-to-1 AND gate
XOR_2 : 2-to-1 XOR gate
MUX_2 : 2-to-1 MUX

[0052]

표 2

표 2. GF(p)상의 곱셈기 성능 비교

w	k	m	e	커널 사이클		클럭 사이클		FF 개수	
				Savas	도7	Savas	도7	Savas	도7
16	16	192	12	12	13	406	439	1984	1612
		224	14	14	15	474	507	1984	1612
		266	17	17	17	576	576	1984	1612
		384	24	24	25	814	847	1984	1612
		521	33	33	33	1152	1152	1984	1612
16	24	192	12	8	9	402	451	3008	2444
		224	14	10	10	502	502	3008	2444
		266	17	12	12	603	603	3008	2444
		384	24	16	17	806	855	3008	2444
		521	33	22	22	1109	1109	3008	2444
32	16	192	6	12	13	400	433	3968	3100
		224	7	14	15	467	500	3968	3100
		266	9	17	17	568	568	3968	3100
		384	12	24	25	802	835	3968	3100
		521	17	33	33	1104	1104	3968	3100

[0053]

[0054] 본 발명에서 제안된 구조는 다음과 같은 세 가지 장점을 가진다. 1) 기존에 제안된 구조에 비해 비교적 적은 FF 개수를 가진다. 2) non-redundant 형태의 결과값을 출력하기 위한 추가 회로가 필요하지 않고, 보정에 필요한 클럭 사이클 수를 감소시켰다. 3) 제안된 곱셈기 회로를 재사용해서 덧셈 연산을 수행한다. 따라서 제안된 다정도 CSA에 기반한 듀얼-필드상의 확장성 있는 듀얼-필드 몽고매리 곱셈기는 암호 디바이스의 곱셈 및 덧셈 연산을 위해 적합하며, 특히 스마트카드나 휴대용 장치와 같은 저면적 응용에 매우 적합하다.

도면의 간단한 설명

[0055] 도 1은 유한체 GF(p)상의 몽고매리 곱셈 알고리즘을 나타낸 것이다.

[0056] 도 2는 유한체 $GF(2^m)$상의 몽고매리 곱셈 알고리즘을 나타낸 것이다.

[0057] 도 3은 본 발명에 따른 다정도 캐리 세이브 가산기에 기반한 GF(p)상의 워드-레벨 몽고매리 곱셈 알고리즘을 나타낸 것이다.

[0058] 도 4는 본 발명에 따른 다정도 캐리 세이브 가산기에 기반한 $GF(2^m)$상의 워드-레벨 몽고매리 곱셈 알고리즘을 나타낸 것이다.

[0059] 도 5a는 확장성 있는 몽고매리 곱셈 알고리즘을 위한 자료의존 그래프를 나타낸 것이다.

[0060] 도 5b는 파이프라인된 2개의 처리기(PE:Processing Element)를 이용한 4-비트 곱셈 연산(w=1,n=2) 과정을 도식적으로 나타낸 것이다.

[0061] 도 5c는 3개의 처리기를 이용한 도 5b와 동일한 계산과정을 도식적으로 나타낸 것이다.

[0062] 도 6a는 워드-레벨 덧셈 연산을 위한 자료의존 그래프를 나타낸 것이다.

[0063] 도 6b는 파이프 라인된 4개의 처리기를 이용한 6-비트 덧셈 연산과정(w=1,n=2)을 도식적으로 나타낸 것이다.

[0064] 도 7은 본 발명에 따른 곱셈 및 덧셈 연산을 위한 파이프라인 구조의 다정도캐리 세이브 가산기를 이용한 듀얼 필드상의 확장성 있는 몽고매리 곱셈기를 나타낸 것이다.

[0065] 도 8은 도 7의 처리기의 블록 다이어그램을 나타낸 것이다.

[0066] 도 9는 도 7의 처리기의 데이터 패스(부)를(w=4,b=2,n=2) 나타낸 것이다.

도면

도면1

[알고리즘 1] : $GF(p)$상의 Montgomery 곱셈 알고리즘

Input : $A,\ B \in GF(p)$ and p

Output : $C \in GF(p)$

1. $C=0$
2. for $i=0$ to $m-1$
3. $\quad C=C+a_i \cdot B$
4. $\quad C=C+c_0 \cdot p$
5. $\quad C=C/2$
6. if $C \geq p$ then $C=C-p$
7. return C

도면2

[알고리즘 2] : $GF(2^m)$상의 Montgomery 곱셈 알고리즘

Input : $A(x)$, $B(x) \in GF(2^m)$ and $p(x)$

Output : $C(x) \in GF(2^m)$

1. $C(x) = 0^{m+1}$
2. for $i = 0$ to $m-1$
3. $\quad C(x) = C(x) \oplus a_i \cdot B(x)$
4. $\quad C(x) = C(x) \oplus c_0 \cdot p(x)$
5. $\quad C(x) = C(x)/2$
6. return $C(x)$

도면3

[알고리즘 3] : MP-CSA에 기반한 $GF(p)$상의 워드-레벨 Montgomery 곱셈 알고리즘

Input : A, $B \in GF(p)$ and p

Output : $C \in GF(p)$

1. $(TS, TC) = (0^m, 0^{n-1})$
2. $(CT, CB) = (0^m, 0^n)$
3. for $i = 0$ to $(m-1)+n$
4. if $i > m-1$ then
5. $(a_i, B) = (0, 0^m)$
6. $(CT_0, TS^{(0)(0)}) = TS^{(0)(0)} + a_i \cdot B^{(0)(0)} + CT_{n-1}$
7. $parity = TS_0^{(0)(0)}$
8. $(CB_0, TS^{(0)(0)}) = TS^{(0)(0)} + parity \cdot p^{(0)(0)} + CB_{n-1}$
9. for $k = 1$ to $n-1$
10. $(CT_k, TS^{(0)(k)}) = TS^{(0)(k)} + a_i \cdot B^{(0)(k)} + TC_{k-1}^{(0)}$
11. $(CB_k, TS^{(0)(k)}) = TS^{(0)(k)} + parity \cdot p^{(0)(k)} + CT_{k-1}$
12. $TC_{k-1}^{(0)} = TS_0^{(0)(k)} \cdot CB_{k-1}$
13. $TS_0^{(0)(k)} = TS_0^{(0)(k)} \oplus CB_{k-1}$
14. for $j = 1$ to $e-1$
15. $(CT_0, TS^{(j)(0)}) = TS^{(j)(0)} + a_i \cdot B^{(j)(0)} + CT_{n-1}$
16. $(CB_0, TS^{(j)(0)}) = TS^{(j)(0)} + parity \cdot p^{(j)(0)} + CB_{n-1}$
17. for $k = 1$ to $n-1$
18. $(CT_k, TS^{(j)(k)}) = TS^{(j)(k)} + a_i \cdot B^{(j)(k)} + TC_{k-1}^{(j)}$
19. $(CB_k, TS^{(j)(k)}) = TS^{(j)(k)} + parity \cdot p^{(j)(k)} + CT_{k-1}$
20. $TC_{k-1}^{(j)} = TS_0^{(j)(k)} \cdot CB_{k-1}$
21. $TS_0^{(j)(k)} = TS_0^{(j)(k)} \oplus CB_{k-1}$
22. $TS^{(j-1)} = (TS_0^{(j)} | TS_{w-1\cdots 1}^{(j-1)})$
23. $TS_{w-1}^{(e-1)} = 0$
24. $C = TS$
25. return C

도면4

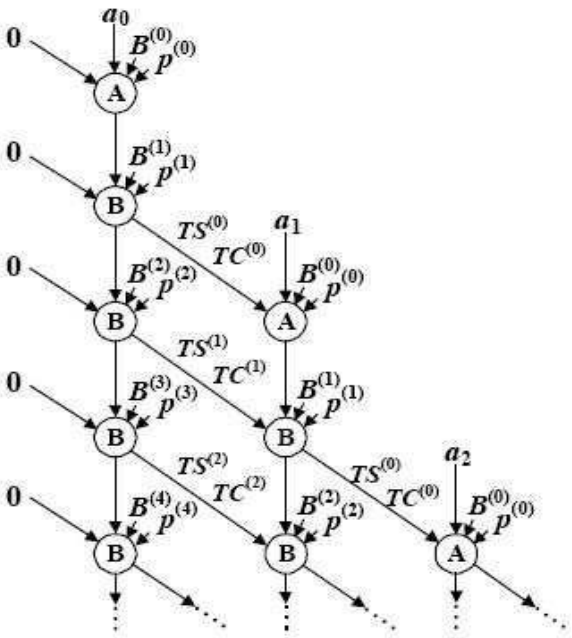
[알고리즘 4] : MP-CSA에 기반한 $GF(2^m)$상의 워드-레벨 Montgomery 곱셈 알고리즘

Input : $A(x)$, $B(x) \in GF(2^m)$ and $p(x)$

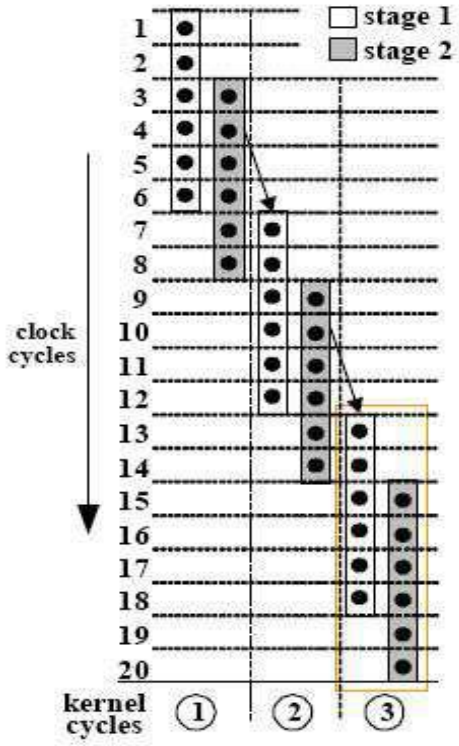
Output : $C(x) \in GF(2^m)$

1. $TS = 0^{m+1}$
2. for $i = 0$ to m
3. $\quad TS^{(0)} = a_i \bullet B^{(0)} \oplus TS^{(0)}$
4. $\quad parity = TS_0^{(0)}$
5. $\quad TS^{(0)} = parity \bullet p^{(0)} \oplus TS^{(0)}$
6. $\quad$ for $j = 1$ to $e-1$
7. $\quad\quad TS^{(j)} = a_i \bullet B^{(j)} \oplus TS^{(j)}$
8. $\quad\quad TS^{(j)} = parity \bullet p^{(j)} \oplus TS^{(j)}$
9. $\quad\quad TS^{(j)} = (TS_0^{(j)} | TS_{w-1\cdots 1}^{(j-1)})$
10. $\quad TS_{w-1}^{(e-1)} = 0$
11. $C = TS$
12. return $C(x)$

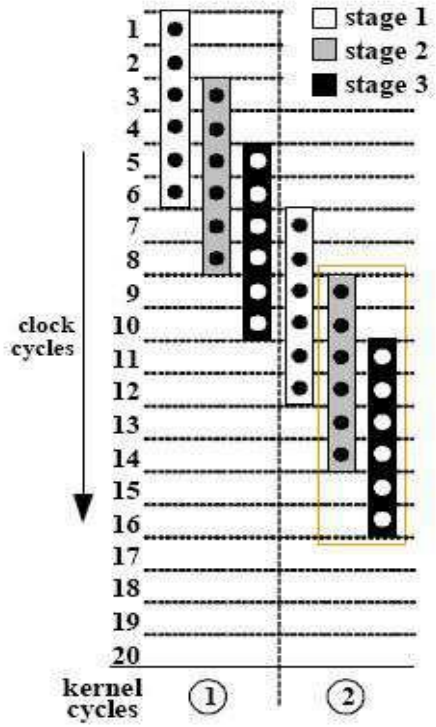
도면5a


0
a_0
$B^{(0)}$ $p^{(0)}$
A
$B^{(1)}$ $p^{(1)}$
B
$TS^{(0)}$
$TC^{(0)}$
a_1
$B^{(2)}$ $p^{(2)}$
$B^{(0)}$ $p^{(0)}$
$TS^{(1)}$
$TC^{(1)}$
$B^{(3)}$ $p^{(3)}$
$B^{(1)}$ $p^{(1)}$
$TS^{(2)}$
$TC^{(2)}$
$TS^{(0)}$
$TC^{(0)}$
a_2
$B^{(4)}$ $p^{(4)}$
$B^{(2)}$ $p^{(2)}$
$B^{(0)}$ $p^{(0)}$

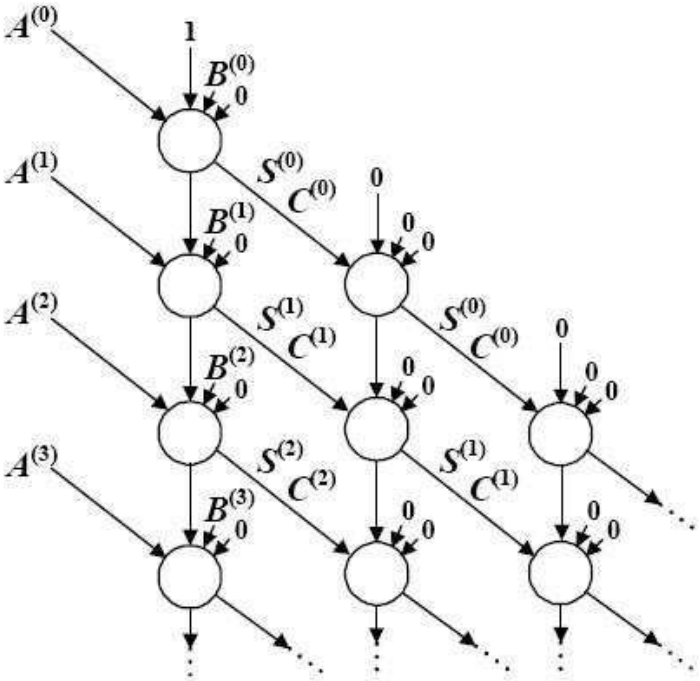
도면5b


stage 1
stage 2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
clock cycles
kernel cycles
1
2
3

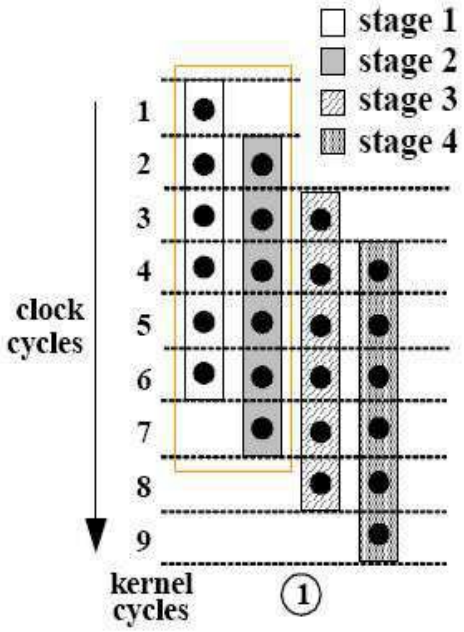
도면5c


stage 1
stage 2
stage 3
clock cycles
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
kernel cycles
1
2

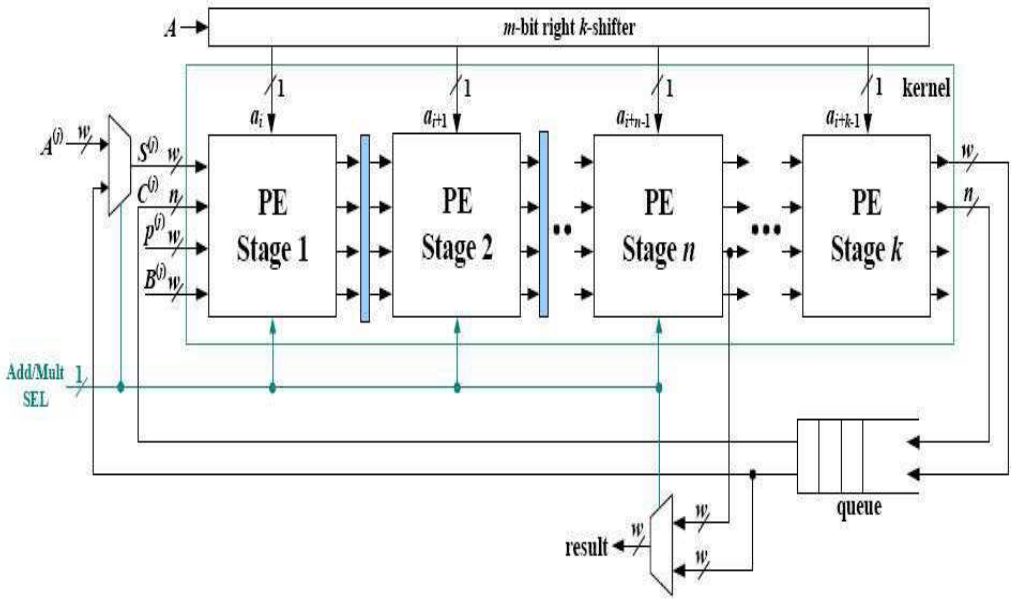
도면6a


$A^{(0)}$
1
$B^{(0)}$
0
$A^{(1)}$
$S^{(0)}$
$C^{(0)}$
0
$B^{(1)}$
0
$A^{(2)}$
$S^{(1)}$
$C^{(1)}$
$B^{(2)}$
$A^{(3)}$
$S^{(2)}$
$C^{(2)}$
$B^{(3)}$

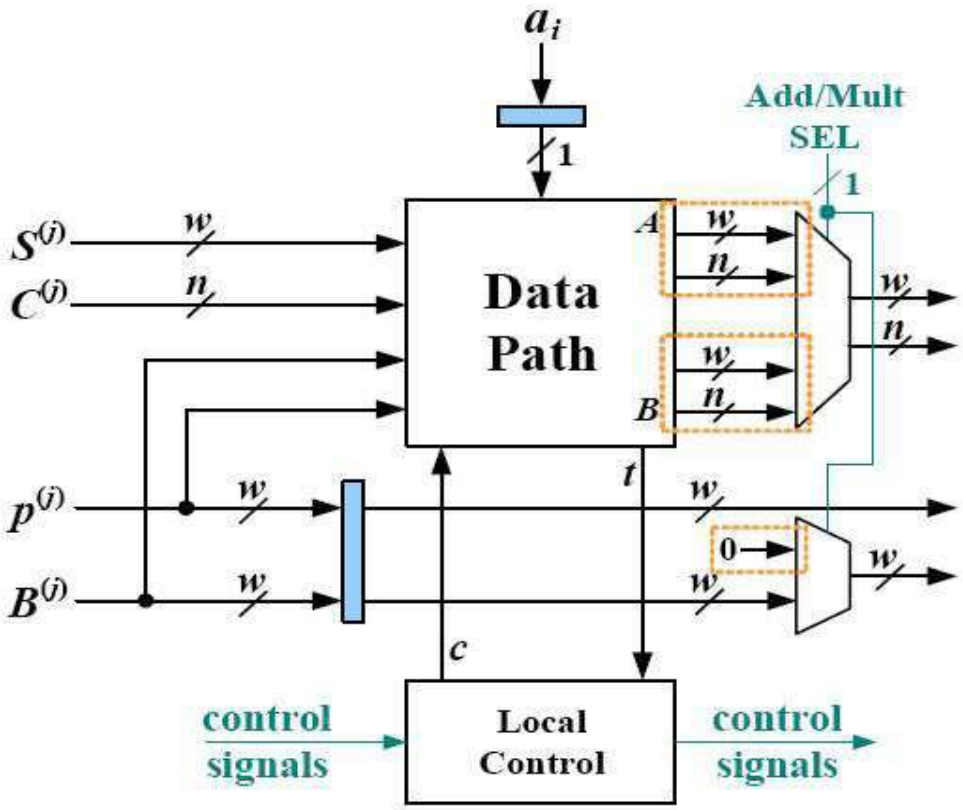
도면6b


stage 1
stage 2
stage 3
stage 4
clock cycles
1
2
3
4
5
6
7
8
9
kernel cycles
1

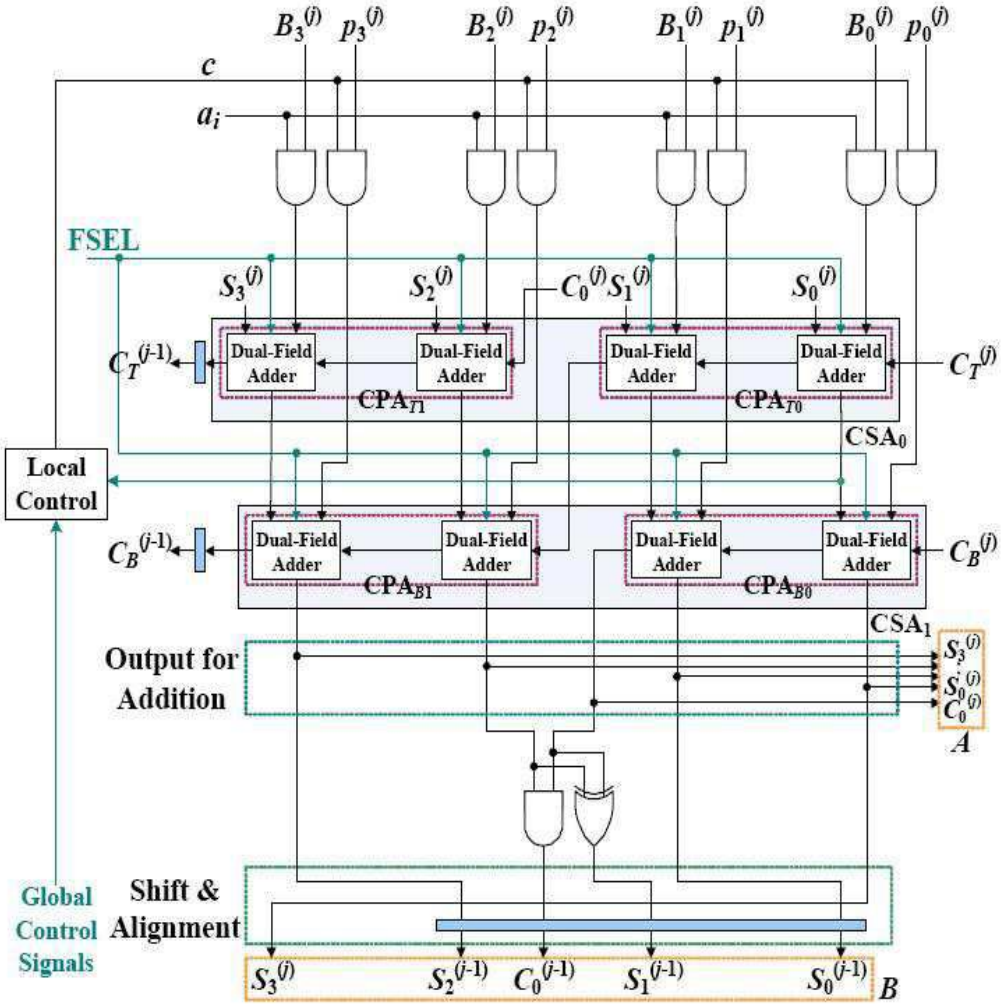
도면7


A
m-bit right k-shifter
kernel
PE Stage 1
PE Stage 2
PE Stage n
PE Stage k
Add/Mult SEL
queue
result

도면8


a_i
1
Add/Mult
SEL
1
$S^{(j)}$
w
$C^{(j)}$
n
Data
Path
A
w
n
B
w
n
w
n
t
w
$p^{(j)}$
w
0
$B^{(j)}$
w
w
w
c
control
signals
Local
Control
control
signals

도면9


$B_3^{(j)}$ $p_3^{(j)}$ $B_2^{(j)}$ $p_2^{(j)}$ $B_1^{(j)}$ $p_1^{(j)}$ $B_0^{(j)}$ $p_0^{(j)}$
c
a_i
FSEL
$S_3^{(j)}$ $S_2^{(j)}$ $C_0^{(j)}$ $S_1^{(j)}$ $S_0^{(j)}$
$C_T^{(j-1)}$
Dual-Field Adder
CPA_{T1}
CPA_{T0}
$C_T^{(j)}$
CSA_0
Local Control
$C_B^{(j-1)}$
CPA_{B1}
CPA_{B0}
$C_B^{(j)}$
CSA_1
Output for Addition
A
Global Control Signals
Shift & Alignment
$S_3^{(j)}$ $S_2^{(j-1)}$ $C_0^{(j-1)}$ $S_1^{(j-1)}$ $S_0^{(j-1)}$
B