(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0140220
(43) 공개일자 2023년10월06일

(51) 국제특허분류(Int. Cl.)
G06F 9/38 (2006.01) *G06F 9/30* (2018.01)
G06N 10/00 (2022.01) *B82Y 10/00* (2017.01)
(52) CPC특허분류
G06F 9/3877 (2013.01)
G06F 9/30141 (2013.01)
(21) 출원번호 10-2022-0039096
(22) 출원일자 2022년03월29일
심사청구일자 2022년03월29일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
김지훈
서울특별시 서초구 서초대로38길 12, 102동 603호 (서초동, 마제스타시티, 힐스테이트 서리풀)
이지혜
서울특별시 영등포구 영등포로 17, 101동 1501호 (양평동2가, 상록수아파트)
(74) 대리인
리앤목특허법인

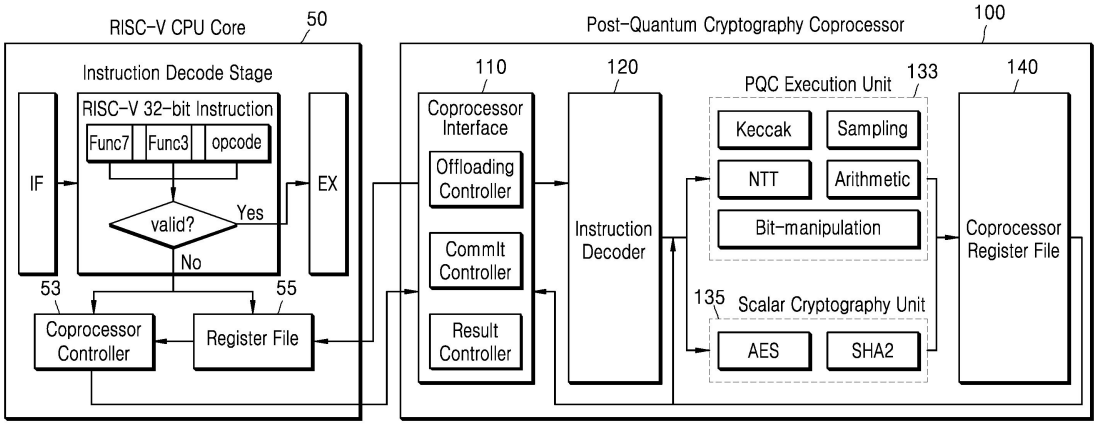
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **양자내성암호 보조 처리 장치 및 이를 이용한 연산 방법**

(57) 요 약

양자내성암호의 연산에 있어서 연산 성능을 향상시킬 수 있는 양자내성암호 보조 처리 장치 및 보조 처리 장치를 이용한 연산 방법을 위하여, 본 발명에 따른 보조 처리 장치는, 중앙 처리 장치에 연결되어 연산을 수행하는 보조 처리 장치에 있어서, 상기 중앙 처리 장치로부터 양자내성암호와 관련된 명령어를 수신하고, 상기 명령어와 관련하여 양자내성암호에 사용되는 연산에 대하여 미리 저장된 복수의 명령어 집합 구조를 이용하여 상기 명령어를 연산하는 연산 유닛을 포함한다.

대 표 도



(52) CPC특허분류
G06F 9/30196 (2013.01)
G06N 10/00 (2022.01)
B82Y 10/00 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711152738
과제번호	2021-0-00724-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보보호핵심원천기술개발(R&D,정보화)
연구과제명	임베디드 시스템 악성코드 탐지·복원을 위한 RISC-V 기반 보안 CPU 아키텍처 핵심 기술 개발
기 여 율	1/1
과제수행기관명	한국전자통신연구원
연구기간	2022.01.01 ~ 2022.12.31

명 세 서

청구범위

청구항 1

중앙 처리 장치에 연결되어 연산을 수행하는 보조 처리 장치에 있어서,

상기 중앙 처리 장치로부터 양자내성암호와 관련된 명령어를 수신하고, 상기 명령어와 관련하여 양자내성암호에 사용되는 연산에 대하여 미리 저장된 복수의 명령어 집합 구조를 이용하여 상기 명령어를 연산하는 연산 유닛;

을 포함하는, 보조 처리 장치.

청구항 2

제1항에 있어서,

상기 중앙 처리 장치와 상기 연산 유닛 사이에 구비되어 상기 중앙 처리 장치로부터 상기 명령어를 수신하고, 상기 명령어가 상기 연산 유닛에서 연산이 가능한지 확인하는 인터페이스 유닛;

을 더 포함하는, 보조 처리 장치.

청구항 3

제2항에 있어서,

상기 인터페이스 유닛과 상기 연산 유닛 사이에 구비되어 상기 인터페이스 유닛으로부터 상기 명령어를 수신하고, 상기 복수의 명령어 집합 구조 중에서 상기 명령어가 연산되는 명령어 집합 구조를 결정하는 명령어 디코더 유닛;

을 더 포함하는, 보조 처리 장치.

청구항 4

제3항에 있어서,

상기 연산 유닛과 연결되어 상기 복수의 명령어 집합 구조에서 연산된 연산 결과를 저장하는 레지스터 파일 유닛;

을 더 포함하는, 보조 처리 장치.

청구항 5

제4항에 있어서,

상기 인터페이스 유닛, 상기 명령어 디코더 유닛, 상기 연산 유닛, 및 상기 레지스터 파일 유닛은 파이프라인 구조를 갖는, 보조 처리 장치.

청구항 6

컴퓨터를 이용하여 제1항 내지 제5항 중 어느 한 항의 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램.

청구항 7

중앙 처리 장치에 연결되어 연산을 수행하는 보조 처리 장치를 이용한 연산 방법에 있어서,

상기 중앙 처리 장치로부터 양자내성암호와 관련된 명령어를 수신하는 단계; 및

상기 명령어와 관련하여 양자내성암호에 사용되는 연산에 대하여 미리 구비된 복수의 명령어 집합 구조를 연산할 수 있는 연산 유닛을 이용하여 상기 명령어를 연산하는 단계;

를 포함하는, 연산 방법.

청구항 8

제7항에 있어서,

상기 명령어가 상기 복수의 명령어 집합 구조를 연산할 수 있는 연산 유닛을 이용하여 연산이 가능한지 확인하는 단계를 더 포함하는, 연산 방법.

청구항 9

제8항에 있어서,

상기 연산이 가능한지 확인하는 단계에서 확인된 명령어에 대하여, 상기 복수의 명령어 집합 구조 중에서 상기 명령어가 연산되는 명령어 집합 구조를 결정하는 단계를 더 포함하는, 연산 방법.

청구항 10

제9항에 있어서,

상기 복수의 명령어 집합 구조를 연산 가능한 연산 유닛에서 연산된 연산 결과를 저장하는 단계를 더 포함하는, 연산 방법.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시예들은, 양자내성암호의 연산을 수행하는 보조 처리 장치 및 보조 처리 장치를 이용한 연산 방법에 관한 것이다.

배 경 기 술

[0002] 오늘날 대부분의 컴퓨터 보안 분야에서는 RSA(Rivest-Shamir-Adleman) 암호시스템을 사용하고 있다. 이는 소인수분해에 대한 수학적 문제를 기반으로 한 시스템이며, 두 개의 수를 곱셈하기는 쉬우나 곱셈의 결과를 바탕으로 어떤 수가 곱해진 것인지 결정하는 것은 시간이 더 많이 드는 특성을 바탕으로 한다. 1994 년에 더 빠르고 효율적으로 소인수분해를 할 수 있는 쇼어 알고리즘(Shor's algorithm)이 제안되었고, 연산 효율이 높은 양자컴퓨터의 개발로 기존의 RSA 암호 시스템과 타원곡선암호(Elliptic curve cryptography)의 보안성이 위협받게 되었다.

[0003] 이러한 양자컴퓨팅 시대가 시작되며 컴퓨터 보안을 유지하기 위하여 양자내성암호가 대안으로 제안되었으며, 현재 NIST(National Institute of Standards and Technology)에서 양자내성암호 표준화 프로세스를 진행하고 있다.

[0004] 양자내성암호는 계산 복잡도가 높아 암호화 및 복호화, 검증에 필요한 연산 시간이 길며, 기존의 공개키 암호 알고리즘에 비해 키 사이즈가 크기 때문에 연산 기기의 메모리를 많이 사용한다는 한계점을 가지고 있다. 이러한 한계점을 보완하기 위하여 지금까지 많은 양자내성암호 구현에 관한 연구가 진행되었다. 양자내성암호를 위한 전담 하드웨어 가속기(dedicated hardware accelerator)를 통해 암호화 및 복호화 프로그램을 가속하는 경우, 타겟으로 하는 양자내성암호 알고리즘의 실행 시간을 단축할 수 있으나 하드웨어 구조가 타겟 알고리즘에 맞게 설계되기 때문에, 새롭게 개발되는 알고리즘은 지원할 수 없는 한계를 가지고 있다.

[0005] 반면 양자내성암호 시스템을 소프트웨어로 구현하는 경우, 다양한 알고리즘을 유연하게 지원할 수 있으나 실행 시간이 전담 하드웨어 가속기보다 오래 소요된다는 한계점이 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명이 해결하고자 하는 과제는 전술한 문제점을 극복하기 위한 것으로서, 양자내성암호의 연산에 있어서 연

산 성능을 향상시킬 수 있는 양자내성암호 보조 처리 장치 및 보조 처리 장치를 이용한 연산 방법을 제공하는 것이다. 그러나 이러한 과제는 예시적인 것으로, 이에 의해 본 발명의 범위가 한정되는 것은 아니다.

과제의 해결 수단

[0007] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 본 발명의 일 측면에 따른 보조 처리 장치는, 중앙 처리 장치에 연결되어 연산을 수행하는 보조 처리 장치에 있어서, 상기 중앙 처리 장치로부터 양자내성암호와 관련된 명령어를 수신하고, 상기 명령어와 관련하여 양자내성암호에 사용되는 연산에 대하여 미리 저장된 복수의 명령어 집합 구조를 이용하여 상기 명령어를 연산하는 연산 유닛을 포함하는 보조 처리 장치가 제공된다.

[0008] 일 예에 따르면, 본 발명의 일 측면에 따른 보조 처리 장치는, 상기 중앙 처리 장치와 상기 연산 유닛 사이에 구비되어 상기 중앙 처리 장치로부터 상기 명령어를 수신하고, 상기 명령어가 상기 연산 유닛에서 연산이 가능한지 확인하는 인터페이스 유닛을 더 포함할 수 있다.

[0009] 다른 예에 따르면, 본 발명의 일 측면에 따른 보조 처리 장치는, 상기 인터페이스 유닛과 상기 연산 유닛 사이에 구비되어 상기 인터페이스 유닛으로부터 상기 명령어를 수신하고, 상기 복수의 명령어 집합 구조 중에서 상기 명령어가 연산되는 명령어 집합 구조를 결정하는 명령어 디코더 유닛을 더 포함할 수 있다.

[0010] 또 다른 예에 따르면, 본 발명의 일 측면에 따른 보조 처리 장치는, 상기 연산 유닛과 연결되어 상기 복수의 명령어 집합 구조에서 연산된 연산 결과를 저장하는 레지스터 파일 유닛을 더 포함할 수 있다.

[0011] 또 다른 예에 따르면, 상기 인터페이스 유닛, 상기 명령어 디코더 유닛, 상기 연산 유닛, 및 상기 레지스터 파일 유닛은 파이프라인 구조를 가질 수 있다.

[0012] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 컴퓨팅 장치를 이용하여 상술한 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램이 제공된다.

[0013] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 본 발명의 일 측면에 따른 연산 방법은, 중앙 처리 장치에 연결되어 연산을 수행하는 보조 처리 장치를 이용한 연산 방법에 있어서, 상기 중앙 처리 장치로부터 양자내성암호와 관련된 명령어를 수신하는 단계, 및 상기 명령어와 관련하여 양자내성암호에 사용되는 연산에 대하여 미리 구비된 복수의 명령어 집합 구조를 연산할 수 있는 연산 유닛을 이용하여 상기 명령어를 연산하는 단계를 포함하는 연산 방법이 제공된다.

[0014] 일 예에 따르면, 본 발명의 일 측면에 따른 연산 방법은, 상기 명령어가 상기 복수의 명령어 집합 구조를 연산할 수 있는 연산 유닛을 이용하여 연산이 가능한지 확인하는 단계를 더 포함할 수 있다.

[0015] 다른 예에 따르면, 본 발명의 일 측면에 따른 연산 방법은, 상기 연산이 가능한지 확인하는 단계에서 확인된 명령어에 대하여, 상기 복수의 명령어 집합 구조 중에서 상기 명령어가 연산되는 명령어 집합 구조를 결정하는 단계를 더 포함할 수 있다.

[0016] 또 다른 예에 따르면, 본 발명의 일 측면에 따른 연산 방법은, 상기 복수의 명령어 집합 구조를 연산 가능한 연산 유닛에서 연산된 연산 결과를 저장하는 단계를 더 포함할 수 있다.

[0017] 전술한 것 외의 다른 측면, 특징, 이점은 이하의 발명을 실시하기 위한 구체적인 내용, 청구범위 및 도면으로부터 명확해질 것이다.

발명의 효과

[0018] 상기한 바와 같이 이루어진 본 발명의 일 실시예에 따르면, 양자내성암호의 연산에 있어서 연산 성능을 향상시킬 수 있는 양자내성암호 보조 처리 장치 및 보조 처리 장치를 이용한 연산 방법을 구현할 수 있다. 물론 이러한 효과에 의해 본 발명의 범위가 한정되는 것은 아니다.

도면의 간단한 설명

[0019] 도 1은 본 발명의 일 실시예에 따른 보조 처리 장치를 개략적으로 도시하는 도면이다.

도 2는 본 발명의 일 실시예에 따른 보조 처리 장치가 중앙 처리 장치에 액세스하는 회로를 개략적으로 도시하는 도면이다.

도 3은 본 발명의 일 실시예에 따른 연산 방법을 설명하기 위한 순서도이다.

도 4 내지 도 7은 본 발명의 일 실시예에 따른 복수의 명령어 집합 구조를 설명하기 위한 도면이다.

도 8 및 도 9는 본 발명의 일 실시예에 따른 보조 처리 장치의 성능을 실험한 실험 데이터 및 결과의 예시들이다.

발명을 실시하기 위한 구체적인 내용

[0020] 아래에서는 첨부한 도면을 참조하여 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 다양한 실시예들을 상세히 설명한다. 그러나 본 개시의 기술적 사상은 다양한 형태로 변형되어 구현될 수 있으므로 본 명세서에서 설명하는 실시예들로 제한되지 않는다. 본 명세서에 개시된 실시예들을 설명함에 있어서 관련된 공지 기술을 구체적으로 설명하는 것이 본 개시의 기술적 사상의 요지를 흐릴 수 있다고 판단되는 경우 그 공지 기술에 대한 구체적인 설명을 생략한다. 동일하거나 유사한 구성요소는 동일한 참조 번호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다.

[0021] 본 명세서에서 어떤 요소가 다른 요소와 "연결"되어 있다고 기술될 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라 그 중간에 다른 요소를 사이에 두고 "간접적으로 연결"되어 있는 경우도 포함한다. 어떤 요소가 다른 요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 요소 외에 또 다른 요소를 배제하는 것이 아니라 또 다른 요소를 더 포함할 수 있는 것을 의미한다.

[0022] 일부 실시예들은 기능적인 블록 구성들 및 다양한 처리 단계들로 설명될 수 있다. 이러한 기능 블록들의 일부 또는 전부는 특정 기능을 실행하는 다양한 개수의 하드웨어 및/또는 소프트웨어 구성들로 구현될 수 있다. 예를 들어, 본 개시의 기능 블록들은 하나 이상의 마이크로프로세서들에 의해 구현되거나, 소정의 기능을 위한 회로 구성들에 의해 구현될 수 있다. 본 개시의 기능 블록들은 다양한 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 본 개시의 기능 블록들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다. 본 개시의 기능 블록이 수행하는 기능은 복수의 기능 블록에 의해 수행되거나, 본 개시에서 복수의 기능 블록이 수행하는 기능들은 하나의 기능 블록에 의해 수행될 수도 있다. 또한, 본 개시는 전자적인 환경 설정, 신호 처리, 및/또는 데이터 처리 등을 위하여 종래 기술을 채용할 수 있다.

[0023] 도 1은 본 발명의 일 실시예에 따른 보조 처리 장치를 개략적으로 도시하는 도면이고, 도 2는 본 발명의 일 실시예에 따른 보조 처리 장치가 중앙 처리 장치에 액세스하는 회로를 개략적으로 도시하는 도면이다.

[0024] 도 1 및 도 2를 함께 참조하면, 본 발명의 일 실시예에 따른 보조 처리 장치(100)는 중앙 처리 장치(50)에 액세스할 수 있다. 예를 들어, 보조 처리 장치(100)는 중앙 처리 장치(50)에 연결되어 데이터를 송수신 할 수 있다. 예컨대, 중앙 처리 장치(50)는 RISC-V(Reduced Instruction Set Computer Principles) CPU 코어 장치 일 수 있다. 또한, 보조 처리 장치(100)는 중앙 처리 장치(50)에 부착할 수 있는 맞춤형 RISC-V PQC(Post-Quantum Cryptography) ISA(Instruction Set Architecture)를 포함하는 PQC 보조 프로세서 장치 일 수 있다.

[0025] 예를 들어, 보조 처리 장치(100)는 메모리에 저장된 데이터(예컨대, 희소 행렬)의 검색, 추출, 연산, 읽기, 쓰기 등의 동작을 수행할 수 있다. 보조 처리 장치(100)는 마이크로프로세서(Microprocessor), 중앙처리장치(Central Processing Unit: CPU), 프로세서 코어(Processor Core), 멀티프로세서(Multiprocessor), ASIC(Application-Specific Integrated Circuit), FPGA(Field Programmable Gate Array) 등의 처리 장치를 포함할 수 있으나, 본 발명은 이에 한정되지 않는다.

[0026] 예컨대, 메모리는 보조 처리 장치(100)가 처리하는 데이터를 일시적 또는 영구적으로 저장할 수 있다. 메모리는 RAM(Random Access Memory), ROM(Read Only Memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(Permanent Mass Storage Device)를 포함할 수 있으나, 본 발명이 이에 한정되는 것은 아니다.

[0027] 도 1 및 도 2에 도시된 바와 같이, 보조 처리 장치(100)는 인터페이스 유닛(110), 명령어 디코더 유닛(120), 연산 유닛(130), 및 레지스터 파일 유닛(140)을 포함한다. 하지만 본 발명은 이에 한정되지 않으며, 보조 처리 장치(100)는 다른 구성요소를 더 포함할 수도 있고 일부 구성요소가 생략될 수도 있다. 보조 처리 장치(100)의 일부 구성요소는 복수의 장치로 분리될 수도 있고, 복수개의 구성요소들이 하나의 장치로 병합될 수도 있다. 도 2에 도시된 바와 같이, 보조 처리 장치(100)는 보조 처리 장치 제어 유닛(53) 및 레지스터 파일 유닛(55)과 서로 전기적으로 연결되어 데이터를 주고 받을 수 있다.

[0028] 보조 처리 장치(100)와 이에 포함된 모듈 또는 유닛들은, 전적으로 하드웨어일 수 있고, 또는 부분적으로 하드웨어이고 부분적으로 소프트웨어인 측면을 가질 수 있다. 예컨대, 보조 처리 장치(100)와 이에 포함된 각 유닛은 특정 형식 및 내용의 데이터를 처리하거나 전자통신 방식으로 주고받기 위한 하드웨어 및 이에 관련된 소프

트웨어를 통칭할 수 있다. 본 명세서에서 "모듈", "유닛", 또는 "장치" 등의 용어는 하드웨어 및 해당 하드웨어에 의해 구동되는 소프트웨어의 조합을 지칭하는 것으로 해석될 수 있다. 예컨대, 하드웨어는 CPU 또는 다른 프로세서(processor)를 포함하는 데이터 처리 기기일 수 있다. 또한, 하드웨어에 의해 구동되는 소프트웨어는 실행중인 프로세스, 객체(object), 실행파일(executable file), 실행 스레드(thread of execution), 프로그램(program) 등을 지칭할 수 있다.

[0029] 또한, 보조 처리 장치(100)를 구성하는 각각의 요소는 반드시 서로 물리적으로 구분되는 별개의 장치를 지칭하는 것으로 한정되는 것은 아니다. 즉, 도 2의 인터페이스 유닛(110), 명령어 디코더 유닛(120), 연산 유닛(130), 및 레지스터 파일 유닛(140) 등은 보조 처리 장치(100)에 의해 수행되는 동작에 따라 기능적으로 구분한 것일 뿐, 반드시 서로 독립적으로 구분되어야 하는 것이 아니다. 물론, 실시예에 따라서는 이들은 서로 물리적으로 구분되는 별개의 장치로 구현되는 것도 가능하다.

[0030] 이하에서는 상술한 것과 같은 구성을 취할 수 있는 보조 처리 장치(100)의 각 유닛들의 동작 등에 대해, 연산 방법과 함께 설명한다.

[0031] 도 3은 본 발명의 일 실시예에 따른 연산 방법을 설명하기 위한 순서도이다.

[0032] 도 3에 도시된 것과 같이, 본 실시예에 따른 연산 방법은 중앙 처리 장치에 연결되어 연산을 수행하는 보조 처리 장치를 이용한 연산 방법으로서, 중앙 처리 장치로부터 양자내성암호와 관련된 명령어를 수신하는 단계(S110)를 거친다. 이어서, 본 실시예에 따른 연산 방법은 상기 명령어와 관련하여 양자내성암호에 사용되는 연산에 대하여 미리 구비된 복수의 명령어 집합 구조를 연산 가능한 연산 유닛을 이용하여 명령어를 연산하는 단계(S120)를 거친다.

[0033] 도 1 및 도 2를 함께 참조하면, 본 발명의 일 실시예에 따른 연산 유닛(130)은 중앙 처리 장치(50)로부터 양자내성암호와 관련된 명령어를 수신할 수 있다. 예를 들어, 도 1 및 도 2에 도시된 바와 같이, 중앙 처리 장치(50)로부터 인터페이스 유닛(110) 및 명령어 디코더 유닛(120)을 통해 연산 유닛(130)으로 양자내성암호와 관련된 명령어가 수신될 수 있다.

[0034] 또한, 연산 유닛(130)은 명령어와 관련하여 양자내성암호에 사용되는 연산에 대하여 미리 저장된 복수의 명령어 집합 구조를 이용하여 명령어를 연산할 수 있다. 예를 들어, 도 2에 도시된 바와 같이, 연산 유닛(130)은 실행 유닛(133) 및 스칼라 암호화 유닛(135)을 포함할 수 있다. 예컨대, 실행 유닛(133)은 Keccak, NTT(Number Theoretic Transform), 샘플링, 및 조건부 덧셈 및 뺄셈을 포함한 산술 연산을 지원하는 복수의 명령어 집합 구조를 포함할 수 있다. 또한, 본 발명의 일 실시예에 따른 실행 유닛(133)은 비트 조작 확장의 연산을 지원하는 명령어 집합 구조를 포함할 수 있다. 또한, 스칼라 암호화 유닛(135)은 RISC-V 스칼라 암호화의 연산을 지원하는 명령어 집합 구조를 포함할 수 있다.

[0035] 연산 유닛(130)은 RISC-V 형식의 명령어 집합 구조(Instruction Set Architecture, ISA)를 포함하며, 명령어 집합 구조는 간단하고 직관적인 명령어 형식을 사용하며, 양자내성암호 연산에서 반복적으로 사용되는 연산을 명령어로 정의하여 사용하는 특성을 가진다.

[0036] 인터페이스 유닛(110)은 중앙 처리 장치(50)와 연산 유닛(130) 사이에 구비될 수 있다. 도 2를 참조하면, 인터페이스 유닛(110)은 중앙 처리 장치(50)와 연산 유닛(130) 사이에 구비되어 데이터를 송수신할 수 있다. 또한, 인터페이스 유닛(110)은 중앙 처리 장치(50)로부터 명령어를 수신할 수 있다. 또한, 인터페이스 유닛(110)은 명령어가 연산 유닛(130)에서 연산이 가능한지 확인할 수 있다. 예를 들어, 인터페이스 유닛(110)은 중앙 처리 장치(50)로부터 양자내성암호와 관련된 명령어를 수신하면 해당 명령어가 연산 유닛(130)에서 연산이 가능한지 확인한 후 연산이 가능한 명령어를 명령어 디코더 유닛(120)으로 전달할 수 있다.

[0037] 명령어 디코더 유닛(120)은 인터페이스 유닛(110)과 연산 유닛(130) 사이에 구비될 수 있다. 도 2를 참조하면, 명령어 디코더 유닛(120)은 인터페이스 유닛(110)과 연산 유닛(130) 사이에 구비되어 데이터를 송수신할 수 있다. 또한, 명령어 디코더 유닛(120)은 인터페이스 유닛(110)으로부터 명령어를 수신할 수 있다. 예를 들어, 명령어 디코더 유닛(120)은 인터페이스 유닛(110)으로부터 연산 유닛(130)에서 연산이 가능한 명령어를 수신할 수 있다. 또한, 명령어 디코더 유닛(120)은 복수의 명령어 집합 구조 중에서 명령어가 연산되는 명령어 집합 구조를 결정할 수 있다. 예를 들어, 명령어 디코더 유닛(120)은 인터페이스 유닛(110)으로부터 수신한 명령어와 대응되는 명령어 집합 구조를 연산 유닛(130)이 포함하고 있는 복수의 명령어 집합 구조 중에서 결정할 수 있다. 또한, 명령어 디코더 유닛(120)은 연산 유닛(130)으로 복수의 명령어 집합 구조 중에서 결정된 명령어 집합 구조에 해당하는 피연산자와 함께 연산과 관련된 제어 신호를 전달할 수 있다.

[0038] 레지스터 파일 유닛(140)은 연산 유닛(130)과 연결될 수 있다. 도 2를 참조하면, 레지스터 파일 유닛(140)은 연산 유닛(130)과 연결되어 데이터를 송수신할 수 있다. 또한, 레지스터 파일 유닛(140)은 연산 유닛(130)의 복수의 명령어 집합 구조에서 연산된 연산 결과를 저장할 수 있다. 또한, 도 2에 도시된 바와 같이, 레지스터 파일 유닛(140)은 복수의 명령어 집합 구조에서 연산된 연산 결과를 인터페이스 유닛(110)을 통해 중앙 처리 장치(50)로 전달할 수 있다.

[0039] 본 발명의 일 실시예에 따른 인터페이스 유닛(110), 명령어 디코더 유닛(120), 연산 유닛(130), 및 레지스터 파일 유닛(140)은 파이프라인 구조를 가질 수 있다. 도 2에 도시된 바와 같이, 본 발명의 일 실시예에 따른 중앙 처리 장치(50) 및 보조 처리 장치(100)는 파이프라인 구조를 가질 수 있다. 도 2에 도시된 바와 같이, 본 발명의 일 실시예에 따른 인터페이스 유닛(110), 명령어 디코더 유닛(120), 연산 유닛(130), 및 레지스터 파일 유닛(140)은 파이프라인 구조를 가질 수 있다. 예컨대, 파이프라인 구조는 폐회로 구조일 수 있다. 도 2에서 도시된 바와 같이, 인터페이스 유닛(110), 명령어 디코더 유닛(120), 연산 유닛(130), 및 레지스터 파일 유닛(140)은 인터페이스 유닛(110)으로부터 명령어 디코더 유닛(120), 연산 유닛(130), 및 레지스터 파일 유닛(140)을 거쳐서 다시 인터페이스 유닛(110)으로 연결되는 파이프라인 구조를 가질 수 있다.

[0040] 도 4 내지 도 7은 본 발명의 일 실시예에 따른 복수의 명령어 집합 구조를 설명하기 위한 도면이다. 도 4 내지 도 7은 본 발명의 일 실시예에 따른 RISC-V 형식의 명령어 집합 구조(Instruction Set Architecture, ISA)를 나타낸다.

[0041] 도 4를 참조하면, 본 발명의 일 실시예에 따른 Keccak 명령어 집합이 도시되어 있다. 또한, 도 5를 참조하면, 본 발명의 일 실시예에 따른 NTT(Number Theoretic Transform) 명령어 집합이 도시되어 있다. 또한, 도 6을 참조하면, 본 발명의 일 실시예에 따른 중심이항분포(Centered binomial distribution) 샘플링 명령어 집합이 도시되어 있다. 또한, 도 7을 참조하면, 본 발명의 일 실시예에 따른 피연산자의 부호에 따른 조건부 덧셈과 조건부 뺄셈 명령어 집합이 도시되어 있다.

[0042] 예를 들어, 본 발명의 일 실시예에 따른 복수의 명령어 집합 구조는 NIST(National Institute of Standards and Technology)의 양자내성암호 표준화 프로세스에서 반복적으로 사용되는 RISC-V 32비트 형식 명령어에 매핑될 수 있다. 또한, 본 발명의 일 실시예에 따른 복수의 명령어 집합 구조는 NTRU, FALCON, CRYSTALS-KYBER, CRYSTALS-DILITHIUM, FrodoKEM, LWR(Learning-With-Rounding) 기반 알고리즘과 같은 LWE(Learning-With-Errors) 기반 알고리즘을 포함한 PQC 알고리즘에도 적용될 수 있다.

[0043] 예를 들어, 도 4를 참조하면, Keccak 명령어 집합은 SHA3-256, SHA3-512, SHAKE-256 및 SHAKE-512의 기본 연산이다. XOR5, ROLX, ANDX 및 XROL 명령어는 Keccak 연산의 계산을 가속화할 수 있다. 예컨대, Keccak 명령어 집합은 25개의 8바이트 Keccak 상태로 계산되며 보조 처리 장치 레지스터 파일은 상태 데이터를 저장할 수 있으므로 베이스라인 코어의 레지스터 파일만 활용하는 것보다 데이터 이동 수가 적다. 또한, ROL, ANDN과 같은 비트 조작 명령어는 Keccak 명령어 집합에 포함될 수 있다.

[0044] 예를 들어, 도 5를 참조하면, NTT 및 INTT(Inverse NTT)용 MR4, MR4U 및 MR8 명령어 집합은 버터플라이 연산을 위한 반복적인 몽고메리 감소를 지원할 수 있다.

[0045] 예를 들어, 도 6을 참조하면, 샘플링을 위한 SND 명령어 집합은 중심 이항 분포 샘플링에서 반복적인 비트 덧셈과 뺄셈을 가속화할 수 있다.

[0046] 예를 들어, 도 7을 참조하면, 산술 명령어 CON4 및 CON8 명령어 집합은 소스 피연산자의 부호에 따라 조건부 덧셈과 뺄셈을 지원할 수 있다. 예컨대, 본 발명의 명령어 집합 구조의 유연성으로 인해 일반 산술 연산과 개발 중인 향후 PQC 알고리즘을 가속화할 수 있다.

[0047] 도 8 및 도 9는 본 발명의 일 실시예에 따른 보조 처리 장치의 성능을 실험한 실험 데이터 및 결과의 예시들이다.

[0048] 본 발명의 일 실시예에 따른 보조 처리 장치를 이용한 양자내성암호 연산은 기존의 RISC-V 명령어만으로 실행할 때보다 코드 사이즈가 감소하여 메모리 사용량을 줄일 수 있으며, 한 사이클 당 하나의 커스텀 RISC-V 명령어를 수행하였을 때 수행 시간도 감소시킬 수 있다.

[0049] 도 8을 참조하면, 본 발명의 일 실시예에 따른 복수의 명령어 집합 구조를 이용하여 양자내성암호를 연산하는 수행 시간이 감소하였음을 나타내는 그래프가 도시되어 있다.

[0050] 도 9에 도시된 바와 같이, 복수의 명령어 집합 구조를 이용함으로써 양자내성암호의 연산에 대하여 수행 시간이

최대 74.6% 로 감소할 수 있다.

[0051] 본 발명에 따르면, 양자내성암호에 반복적으로 사용되는 연산을 RISC-V 형식의 커스텀 명령어 집합 구조를 정의하고 이를 연산하는 보조프로세서의 하드웨어 구조를 제안하여, 기존의 RISC-V CPU core 에 보조프로세서 인터페이스를 통해 연결하여 사용할 수 있다. 본 발명에 따르면, 양자내성암호 프로그램을 기존의 RISC-V 명령어만으로 실행할 때보다 코드 사이즈가 감소하여 메모리 사용량을 줄일 수 있으며, 한 사이클 당 하나의 커스텀 RISC-V 명령어를 수행하였을 때 수행 시간도 감소시킬 수 있다. 제안하는 아키텍처는 NIST 양자내성암호 표준화 알고리즘들을 대상으로 하기 때문에, 기존의 전담 하드웨어 가속기와 비교하였을 때 더 다양한 알고리즘 프로그램을 가속할 수 있으며 앞으로 표준으로 사용될 양자내성암호 시스템 가속이 가능하다. 또한 기존의 RISC-V CPU 코어의 하드웨어 구조를 크게 수정하지 않고 보조프로세서 인터페이스를 통해 보조프로세서를 연결하여 사용할 수 있는 장점을 가지고 있다.

[0053] 이상 설명된 다양한 실시예들은 예시적이며, 서로 구별되어 독립적으로 실시되어야 하는 것은 아니다. 본 명세서에서 설명된 실시예들은 서로 조합된 형태로 실시될 수 있다.

[0054] 이상 설명된 다양한 실시예들은 컴퓨터 상에서 다양한 구성요소를 통하여 실행될 수 있는 컴퓨터 프로그램의 형태로 구현될 수 있으며, 이와 같은 컴퓨터 프로그램은 컴퓨터로 판독 가능한 매체에 기록될 수 있다. 이때, 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수개 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 애플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다.

[0055] 본 명세서에서, "부", "모듈", "유닛" 등은 프로세서 또는 회로와 같은 하드웨어 구성(hardware component), 및/또는 프로세서와 같은 하드웨어 구성에 의해 실행되는 소프트웨어 구성(software component)일 수 있다. 예를 들면, "부", "모듈", "유닛" 등은 소프트웨어 구성 요소들, 객체 지향 소프트웨어 구성 요소들, 클래스 구성 요소들 및 태스크 구성 요소들과 같은 구성 요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들에 의해 구현될 수 있다.

[0056] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0057] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0058] 50: 중앙 처리 장치

100: 보조 처리 장치

110: 인터페이스 유닛

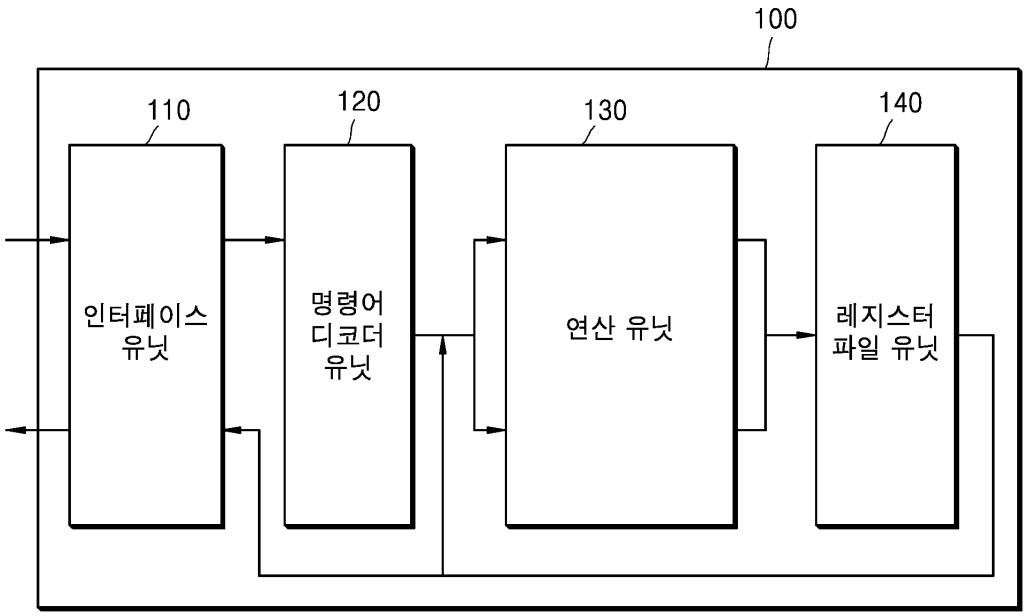
120: 명령어 디코더 유닛

130: 연산 유닛

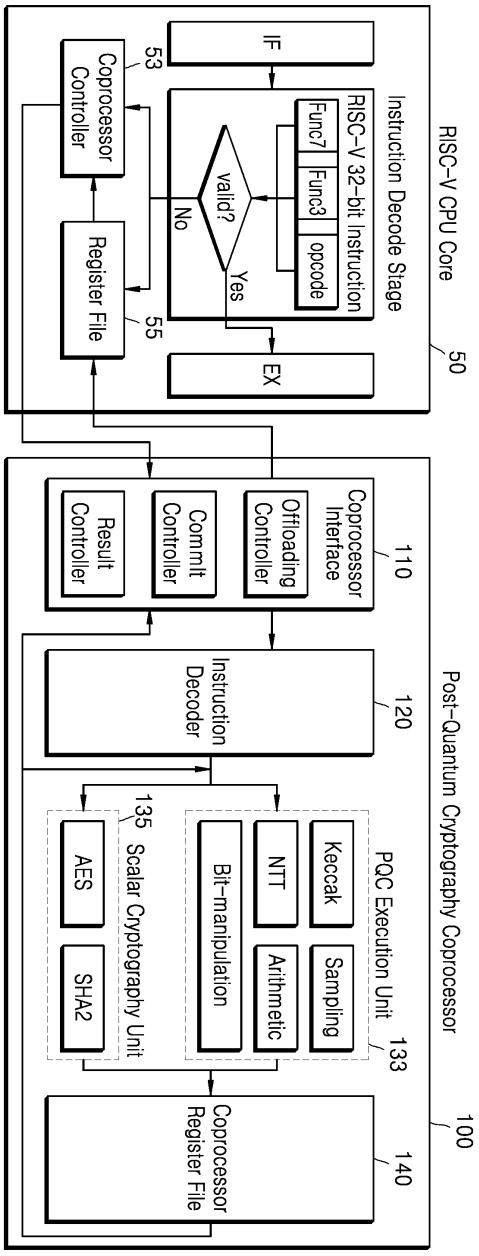
140: 레지스터 파일 유닛

도면

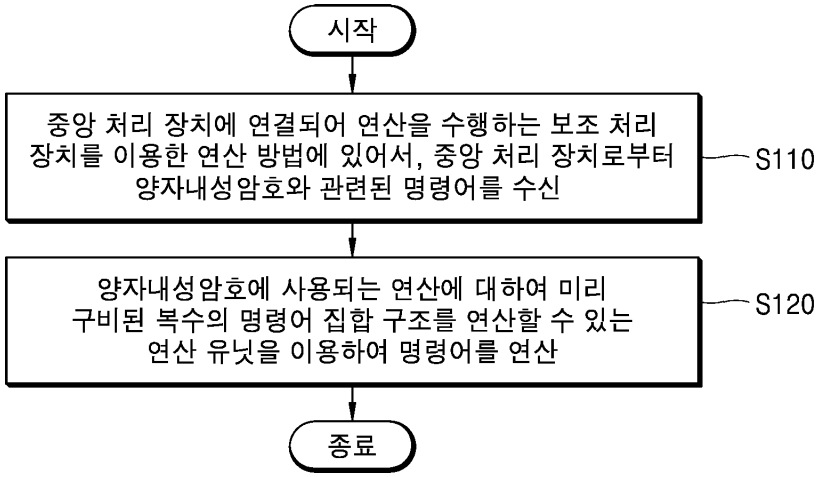
도면1



도면2


RISC-V CPU Core
50
Instruction Decode Stage
IF
RISC-V 32-bit Instruction
Func7
Func3
opcode
valid?
No
Yes
EX
Coprocessor Controller
53
Register File
55
Post-Quantum Cryptography Coprocessor
100
Coprocessor Interface
110
Offloading Controller
Commit Controller
Result Controller
Instruction Decoder
120
PQC Execution Unit
133
Keccak
NTT
Bit-manipulation
Sampling
Arithmetic
Scalar Cryptography Unit
135
AES
SHA2
Coprocessor Register File
140

도면3



도면4

Name	Operation
XOR5	CR[rs] ^ CR[rs+1] ^ CR[rs+2] ^ CR[rs+3] ^ CR[rs+4] → CR[rd]
ROLX	CR[rs] ^ ROL(CR[rs+2], 1) → CR[rd]
ANDX	CR[rs] ^ ANDN(CR[rs+1], CR[rs+2]) → CR[rd]
XROL	ROL(CR[rs] ^ CR[rs+1], shamt) → CR[rd]

도면5

Name	Operation
MR4	(rs1 - (rs1 * rs2 * 0xFFFF) * rs3) >> 16 → rd
MR4U	(rs1 + (rs1 * rs2 * 0xFFFF) * rs3) >> 16 → rd
MR8	(rs1 - (rs1 * rs2 * 0xFFFFFFFF) * rs3) >> 32 → rd

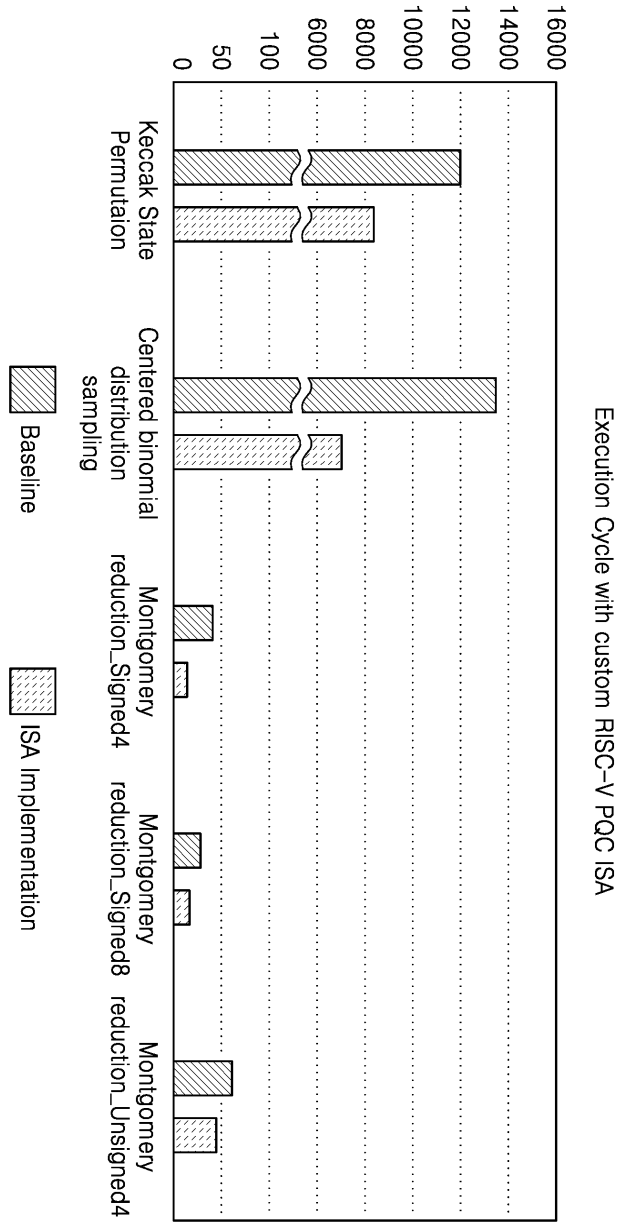
도면6

Name	Operation
SND2	(rs1[rs2] + rs1[rs2+1]) − (rs1[rs2+2] + rs1[rs2+3]) → rd
SND3	(rs1[rs2] + rs1[rs2+1] + rs1[rs2+2] − (rs1[rs2+3] + rs1[rs2+4] + rs1[rs2+5]) → rd
SND4	(rs1[rs2] + rs1[rs2+1] + rs1[rs2+2] + rs1[rs2+3]) − (rs1[rs2+4] + rs1[rs2+5] + rs1[rs2+6] + rs1[rs2+7]) → rd
SND5	(rs1[rs2] + rs1[rs2+1] + rs1[rs2+2] + rs1[rs2+3] + rs1[rs2+4]) − (rs1[rs2+5] + rs1[rs2+6] + rs1[rs2+7] + rs1[rs2+8] + rs1[rs2+9]) → rd

도면7

Name	Operation
CON4	rs1[15] & rs2 → rd
CON8	rs1[31] & rs2 → rd

도면8



도면9

Target Operation	Execution Time (cycle)		Ratio(%)
	Baseline	ISA Implementation	
Keccak State Permutaion	12032	8384	69.68
Centered binomial distribution sampling	13463	7063	52.46
Montgomery reduction_Signed 4Bytes	35	14	40.00
Montgomery reduction_Signed 8Bytes	25	15	60.00
Montgomery reduction_Unsigned 4Bytes	63	47	74.60