(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년07월17일
(11) 등록번호 10-1537018
(24) 등록일자 2015년07월09일

(51) 국제특허분류(Int. Cl.)
G06F 11/08 (2006.01) *G06F 12/14* (2006.01)
G06F 15/00 (2006.01) *G06F 21/00* (2006.01)
(21) 출원번호 10-2008-0096574
(22) 출원일자 2008년10월01일
심사청구일자 2013년09월27일
(65) 공개번호 10-2010-0037313
(43) 공개일자 2010년04월09일
(56) 선행기술조사문헌
JP06112937 A*
KR1020080023595 A*
KR1020070117172 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
리우 세바스찬
경기 성남시 분당구 성남대로 393, B동 826호 (정자동, 두산위브파빌리온)
(74) 대리인
박영우

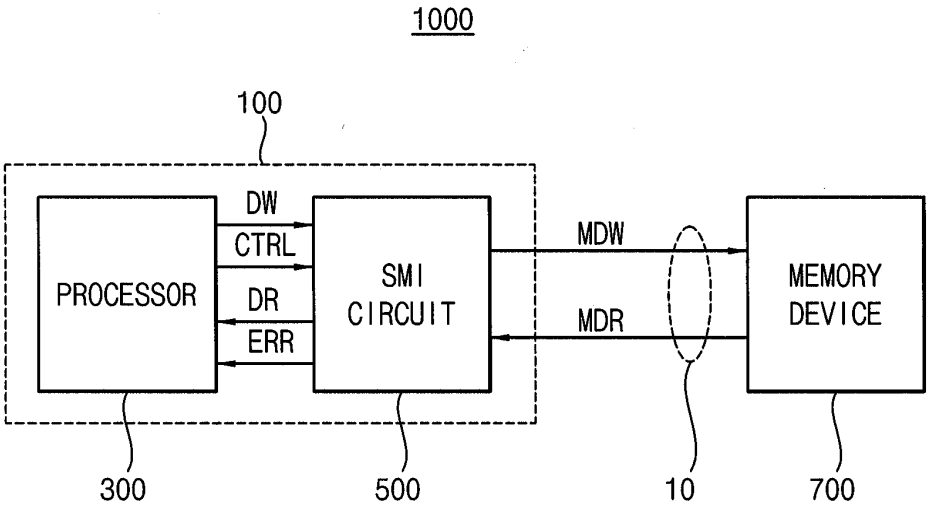
전체 청구항 수 : 총 10 항 심사관 : 정성용

(54) 발명의 명칭 **보안 메모리 인터페이스, 이를 포함하는 시스템 및 스마트카드**

(57) 요 약

보안 메모리 인터페이스는 기입 회로 및 독출 회로를 포함한다. 기입 회로는, 보안 인에이블 신호가 활성화된 경우 프로세서로부터 수신된 기입 데이터에 기초하여 암호화된 메모리 기입 데이터를 발생한다. 독출 회로는, 보안 인에이블 신호가 활성화된 경우, 메모리 장치로부터 수신된 암호화된 메모리 독출 데이터에 기초하여 독출 데이터 및 상기 독출 데이터의 에러 여부를 나타내는 에러 신호를 발생한다.

대 표 도 - 도1



명 세 서

청구범위

청구항 1

보안 인에이블 신호가 활성화된 경우, 프로세서로부터 수신된 기입 데이터에 기초하여 기입 에러 검출 코드를 계산하고, 상기 기입 데이터와 상기 기입 에러 검출 코드를 병합하여 암호화된 메모리 기입 데이터를 발생하는 기입 회로; 및

상기 보안 인에이블 신호가 활성화된 경우, 메모리 장치로부터 수신된 암호화된 메모리 독출 데이터에 기초하여 독출 데이터 및 상기 독출 데이터의 에러 여부를 나타내는 에러 신호를 발생하고, 상기 메모리 독출 데이터에 포함된 독출 에러 검출 코드 및 상기 메모리 독출 데이터에 포함된 상기 독출 데이터에 기초하여 계산된 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생하는 독출 회로를 포함하고,

상기 기입 데이터 및 상기 독출 데이터의 데이터 폭이 16비트인 경우에는 상기 기입 에러 검출 코드 및 상기 독출 에러 검출 코드는 16비트이며, 상기 기입 데이터 및 상기 독출 데이터의 데이터 폭이 24비트인 경우에는 상기 기입 에러 검출 코드 및 상기 독출 에러 검출 코드는 8비트인 보안 메모리 인터페이스.

청구항 2

제1 항에 있어서, 상기 보안 인에이블 신호가 비활성화된 경우,

상기 기입 회로는 상기 기입 데이터와 동일한 상기 메모리 기입 데이터를 발생하고,

상기 독출 회로는 상기 메모리 독출 데이터와 동일한 상기 독출 데이터를 발생하는 것을 특징으로 하는 보안 메모리 인터페이스.

청구항 3

삭제

청구항 4

제1 항에 있어서,

상기 기입 회로는 상기 기입 데이터에 기초하여 상기 기입 에러 검출 코드를 발생하는 제1 코드 계산기를 포함하고,

상기 독출 회로는 상기 독출 데이터에 기초하여 상기 기준 에러 검출 코드를 발생하는 제2 코드 계산기를 포함하고,

상기 제1 코드 계산기 및 상기 제2 코드 계산기는 동일한 구성의 논리 소자들로 구현된 것을 특징으로 하는 보안 메모리 인터페이스.

청구항 5

제1 항에 있어서,

논리 소자들로 구성된 공통의 코드 계산기를 더 포함하고,

상기 기입 회로와 상기 독출 회로는 상기 공통의 코드 계산기를 이용하여 상기 기입 데이터에 기초하여 상기 기입 에러 검출 코드를 발생하고 상기 독출 데이터에 기초하여 상기 기준 에러 검출 코드를 발생하는 것을 특징으로 하는 보안 메모리 인터페이스.

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

데이터를 저장하는 메모리 장치;

상기 메모리 장치의 액세스를 제어하는 프로세서;

상기 메모리 장치와 상기 프로세서 사이의 데이터 전송을 매개하는 보안 메모리 인터페이스; 및

상기 보안 메모리 인터페이스와 상기 메모리 장치를 연결하는 버스를 포함하고,

상기 보안 메모리 인터페이스는,

보안 인에이블 신호가 활성화된 경우, 상기 프로세서로부터 수신된 기입 데이터에 기초하여 기입 에러 검출 코드를 계산하고, 상기 기입 데이터와 상기 기입 에러 검출 코드를 병합하여 암호화된 메모리 기입 데이터를 발생하는 기입 회로; 및

상기 보안 인에이블 신호가 활성화된 경우, 상기 메모리 장치로부터 수신된 암호화된 메모리 독출 데이터에 기초하여 독출 데이터 및 상기 독출 데이터의 에러 여부를 나타내는 에러 신호를 발생하고, 상기 메모리 독출 데이터에 포함된 독출 에러 검출 코드 및 상기 메모리 독출 데이터에 포함된 상기 독출 데이터에 기초하여 계산된 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생하는 독출 회로를 포함하고,

상기 기입 데이터 및 상기 독출 데이터의 데이터 폭이 16비트인 경우에는 상기 기입 에러 검출 코드 및 상기 독출 에러 검출 코드는 16비트이며, 상기 기입 데이터 및 상기 독출 데이터의 데이터 폭이 24비트인 경우에는 상기 기입 에러 검출 코드 및 상기 독출 에러 검출 코드는 8비트인 시스템.

청구항 12

제11 항에 있어서, 상기 프로세서는,

통상의 액세스 또는 보안 액세스를 나타내는 명령어, 입출력 데이터가 저장되는 상기 메모리 장치의 주소, 및 상기 입출력 데이터가 저장되는 프로세서 레지스터의 주소 중 적어도 하나에 기초하여 상기 보안 인에이블 신호를 발생하는 것을 특징으로 하는 시스템.

청구항 13

삭제

청구항 14

제11 항에 있어서, 상기 기입 회로는,

상기 기입 데이터에 기초하여 상기 기입 에러 검출 코드를 발생하는 제1 코드 계산기; 및

상기 보안 인에이블 신호에 응답하여, 상기 기입 데이터와 상기 기입 에러 검출 코드를 병합하여 상기 메모리 기입 데이터를 발생하는 출력부를 포함하는 것을 특징으로 하는 시스템.

청구항 15

제14 항에 있어서, 상기 출력부는,

상기 보안 인에이블 신호 및 레지스터의 폭(register width)을 나타내는 제1 신호에 응답하여 메모리 액세스 폭

(memory access width)을 나타내는 제2 신호를 발생하는 선택 회로;

상기 보안 인에이블 신호 및 상기 제1 신호에 응답하여 상기 기입 데이터 및 상기 기입 에러 검출 코드를 병합하여 출력하는 믹서(mixer); 및

상기 믹서의 출력을 수신하고, 상기 제2 신호에 응답하여 상기 메모리 기입 데이터를 상기 버스로 출력하는 기입 유한 상태 머신을 포함하는 것을 특징으로 하는 시스템.

청구항 16

제14 항에 있어서, 상기 독출 회로는,

상기 보안 인에이블 신호에 응답하여, 상기 메모리 독출 데이터에서 상기 독출 에러 검출 코드 및 상기 독출 데이터를 분리하여 각각 출력하는 입력부;

상기 독출 데이터에 기초하여 상기 기준 에러 검출 코드를 발생하는 제2 코드 계산기; 및

상기 독출 에러 검출 코드 및 상기 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생하는 비교기를 포함하는 것을 특징으로 하는 시스템.

청구항 17

제16 항에 있어서, 상기 입력부는,

상기 보안 인에이블 신호 및 레지스터의 폭을 나타내는 제1 신호에 응답하여 메모리 액세스 폭을 나타내는 제2 신호를 발생하는 선택 회로;

상기 제2 신호에 응답하여 상기 버스로부터 상기 메모리 독출 데이터를 수신하여 출력하는 독출 유한 상태 머신; 및

상기 독출 유한 상태 머신의 출력을 수신하고, 상기 보안 인에이블 신호 및 상기 제1 신호에 응답하여 상기 독출 에러 검출 코드 및 상기 독출 데이터를 분리하여 각각 출력하는 디믹서(demixer)를 포함하는 것을 특징으로 하는 시스템.

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

발명의 설명

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 데이터의 보안에 관한 것으로서, 더욱 상세하게는 보안 메모리 인터페이스, 이를 포함하는 시스템 및 스마트카드에 관한 것이다.

배 경 기 술

[0002] 메모리 장치에 저장되는 보안이 필요한 데이터나 코드 등의 중요한 정보가 외부로부터의 불온한 공격에 의해 변질되거나, 메모리 장치의 자체적인 오류로 인하여 오류가 발생할 수 있다. 이러한 외부로부터의 공격이나 오류를 방지할 수 있는 수단 및 이미 발생한 데이터의 오류를 검출하여 후속 조치를 취할 수 있는 수단이 필요하다.

[0003] 특히 버스를 통해 데이터가 전송되는 경우에는, 전송 데이터는 감지 증폭기의 전력 소모와 관련이 있기 때문에

감지 증폭기의 전력 소모를 파악함으로써 쉽게 분석될 수 있고, 검침(probing) 등과 같은 수단을 이용하여 버스를 통한 공격이 용이하다.

[0004] 스마트카드(smart card) 내의 데이터 전송은 주로 버스를 경유하여 이루어진다. 즉 기입 동작시에는 중앙처리장치로부터 버스를 통하여 메모리 장치로 기입 데이터가 전송되고, 독출 동작시에는 메모리 장치로부터 중앙처리장치로 독출 데이터가 전송된다. 이와 같이 버스를 통해 데이터가 전송되는 시스템에서는 버스의 보안이 취약한 문제점이 있고 전송되는 데이터의 확실성(confidentiality)과 무결성(integrity)이 보장되어야 한다.

발명의 내용

해결 하고자하는 과제

[0005] 상기와 같은 문제점을 해결하기 위한 본 발명의 일 목적은 하드웨어의 구성을 통하여 효율적으로 메모리를 관리하고 데이터의 높은 보안성을 제공할 수 있는 보안 메모리 인터페이스를 제공하는 것이다.

[0006] 본 발명의 다른 목적은 하드웨어의 구성을 통하여 효율적으로 메모리를 관리하고 데이터의 높은 보안성을 제공할 수 있는 보안 메모리 인터페이스를 포함하는 시스템을 제공하는 것이다.

[0007] 본 발명의 또 다른 목적은 버스를 통해 전송되는 데이터의 보안성을 강화할 수 있도록, 하드웨어의 구성을 통하여 효율적으로 메모리를 관리하고 데이터의 높은 보안성을 제공할 수 있는 보안 메모리 인터페이스를 포함하는 스마트카드를 제공하는 것이다.

과제 해결수단

[0008] 상기 일 목적을 달성하기 위해, 본 발명의 일 실시예에 따른 보안 메모리 인터페이스는 기입 회로 및 독출 회로를 포함한다.

[0009] 기입 회로는, 보안 인에이블 신호가 활성화된 경우 프로세서로부터 수신된 기입 데이터에 기초하여 암호화된 메모리 기입 데이터를 발생한다. 독출 회로는, 상기 보안 인에이블 신호가 활성화된 경우 메모리 장치로부터 수신된 암호화된 메모리 독출 데이터에 기초하여 독출 데이터 및 상기 독출 데이터의 에러 여부를 나타내는 에러 신호를 발생한다.

[0010] 상기 보안 인에이블 신호가 비활성화된 경우, 상기 기입 회로는 상기 기입 데이터와 동일한 상기 메모리 기입 데이터를 발생하고, 상기 독출 회로는 상기 메모리 독출 데이터와 동일한 상기 독출 데이터를 발생할 수 있다.

[0011] 상기 독출 회로는, 상기 메모리 독출 데이터에 포함된 독출 에러 검출 코드 및 상기 메모리 독출 데이터에 포함된 상기 독출 데이터에 기초하여 계산된 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생할 수 있다.

[0012] 일 실시예에서, 상기 기입 회로는 상기 기입 데이터에 기초하여 기입 에러 검출 코드를 발생하는 제1 코드 계산기를 포함하고, 상기 독출 회로는 상기 독출 데이터에 기초하여 기준 에러 검출 코드를 발생하는 제2 코드 계산기를 포함할 수 있다. 상기 제1 코드 계산기 및 상기 제2 코드 계산기는 동일한 구성의 논리 소자들로 구현될 수 있다.

[0013] 다른 실시예에서, 상기 보안 메모리 인터페이스는, 논리 소자들로 구성된 공통의 코드 계산기를 더 포함할 수 있다. 상기 기입 회로와 상기 독출 회로는 상기 공통의 코드 계산기를 이용하여 상기 기입 데이터에 기초하여 기입 에러 검출 코드를 발생하고 상기 독출 데이터에 기초하여 기준 에러 검출 코드를 발생할 수 있다.

[0014] 상기 보안 메모리 인터페이스는, 상기 기입 데이터 또는 상기 독출 데이터를 상기 공통의 코드 계산기에 선택적으로 출력하는 제1 스위치, 및 상기 공통의 코드 계산기로부터 발생된 상기 기입 에러 검출 코드 또는 상기 기준 에러 검출 코드를 선택적으로 출력하는 제2 스위치를 더 포함할 수 있다.

[0015] 일 실시예에서, 상기 기입 회로는, 상기 기입 데이터에 기초하여 기입 에러 검출 코드를 발생하는 제1 코드 계산기, 및 상기 보안 인에이블 신호에 응답하여, 상기 기입 데이터와 상기 기입 에러 검출 코드를 병합하여 상기 메모리 기입 데이터를 발생하는 출력부를 포함할 수 있다.

[0016] 상기 독출 회로는, 상기 보안 인에이블 신호에 응답하여, 상기 메모리 독출 데이터에서 독출 에러 검출 코드 및 상기 독출 데이터를 분리하여 각각 출력하는 입력부, 상기 독출 데이터에 기초하여 기준 에러 검출 코드를 발생하는 제2 코드 계산기, 및 상기 독출 에러 검출 코드 및 상기 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생하는 비교기를 포함할 수 있다. 상기 제1 코드 계산기 및 상기 제2 코드 계산기는 동일한 구성의 논리

소자들로 구현될 수 있다.

[0017] 일 실시예에서, 상기 기입 회로 및 상기 독출 회로는 하드 웨어 로직게이트들로 구성될 수 있다.

[0018] 상기 다른 목적을 달성하기 위해, 본 발명의 일 실시예에 따른 시스템은, 데이터를 저장하는 메모리 장치, 상기 메모리 장치의 액세스를 제어하는 프로세서, 및 상기 메모리 장치와 상기 프로세서 사이의 데이터 전송을 매개하는 보안 메모리 인터페이스를 포함한다. 상기 보안 메모리 인터페이스는, 보안 인에이블 신호가 활성화된 경우, 상기 프로세서로부터 수신된 기입 데이터에 기초하여 암호화된 메모리 기입 데이터를 발생하는 기입 회로, 및 상기 보안 인에이블 신호가 활성화된 경우, 상기 메모리 장치로부터 수신된 암호화된 메모리 독출 데이터에 기초하여 독출 데이터 및 상기 독출 데이터의 에러 여부를 나타내는 에러 신호를 발생하는 독출 회로를 포함한다.

[0019] 상기 프로세서는, 통상의 액세스 또는 보안 액세스를 나타내는 명령어, 입출력 데이터가 저장되는 상기 메모리 장치의 주소, 및 상기 입출력 데이터가 저장되는 프로세서 레지스터의 주소 중 적어도 하나에 기초하여 상기 보안 인에이블 신호를 발생할 수 있다.

[0020] 일 실시예에서, 상기 시스템은 상기 보안 메모리 인터페이스와 상기 메모리 장치를 연결하는 버스를 더 포함할 수 있다.

[0021] 상기 기입 회로는, 상기 기입 데이터에 기초하여 기입 에러 검출 코드를 발생하는 제1 코드 계산기, 및 상기 보안 인에이블 신호에 응답하여, 상기 기입 데이터와 상기 기입 에러 검출 코드를 병합하여 상기 메모리 기입 데이터를 발생하는 출력부를 포함할 수 있다.

[0022] 상기 출력부는, 상기 보안 인에이블 신호 및 레지스터의 폭(register width)을 나타내는 제1 신호에 응답하여 메모리 액세스 폭(memory access width)을 나타내는 제2 신호를 발생하는 선택 회로, 상기 보안 인에이블 신호 및 상기 제1 신호에 응답하여 상기 기입 데이터 및 상기 기입 에러 검출 코드를 병합하여 출력하는 믹서(mixer), 및 상기 믹서의 출력을 수신하고, 상기 제2 신호에 응답하여 상기 메모리 기입 데이터를 상기 버스로 출력하는 기입 유한 상태 머신을 포함할 수 있다.

[0023] 상기 독출 회로는, 상기 보안 인에이블 신호에 응답하여, 상기 메모리 독출 데이터에서 독출 에러 검출 코드 및 상기 독출 데이터를 분리하여 각각 출력하는 입력부, 상기 독출 데이터에 기초하여 기준 에러 검출 코드를 발생하는 제2 코드 계산기, 및 상기 독출 에러 검출 코드 및 상기 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생하는 비교기를 포함할 수 있다.

[0024] 상기 입력부는, 상기 보안 인에이블 신호 및 레지스터의 폭을 나타내는 제1 신호에 응답하여 메모리 액세스 폭을 나타내는 제2 신호를 발생하는 선택 회로, 상기 제2 신호에 응답하여 상기 버스로부터 상기 메모리 독출 데이터를 수신하여 출력하는 독출 유한 상태 머신, 및 상기 독출 유한 상태 머신의 출력을 수신하고, 상기 보안 인에이블 신호 및 상기 제1 신호에 응답하여 상기 독출 에러 검출 코드 및 상기 독출 데이터를 분리하여 각각 출력하는 디믹서(demixer)를 포함할 수 있다.

[0025] 상기 제1 코드 계산기 및 상기 제2 코드 계산기의 각각은, 상기 기입 에러 검출 코드 또는 상기 기준 에러 검출 코드의 하위 비트들을 계산하는 하위 계산기, 및 상기 기입 에러 검출 코드 또는 상기 기준 에러 검출 코드의 상위 비트들을 계산하는 상위 계산기를 포함할 수 있다.

[0026] 상기 또 다른 목적을 달성하기 위해, 본 발명의 일 실시예에 따른 스마트카드는, 데이터를 저장하는 적어도 하나의 메모리 장치, 상기 메모리 장치의 액세스를 제어하는 메모리 제어 장치, 상기 메모리 장치와 상기 메모리 제어 장치를 연결하는 버스, 및 상기 버스에 연결되어 호스트와의 통신을 매개하는 호스트 인터페이스를 포함한다. 상기 메모리 제어 장치는, 보안 인에이블 신호가 활성화된 경우 프로세서로부터 수신된 기입 데이터에 기초하여 암호화된 메모리 기입 데이터를 발생하고, 상기 보안 인에이블 신호가 활성화된 경우 상기 메모리 장치로부터 수신된 암호화된 메모리 독출 데이터에 기초하여 독출 데이터 및 상기 독출 데이터의 에러 여부를 나타내는 에러 신호를 발생하는 보안 메모리 인터페이스를 포함한다.

[0027] 상기 보안 메모리 인터페이스는, 상기 메모리 독출 데이터에 포함된 독출 에러 검출 코드 및 상기 메모리 독출 데이터에 포함된 상기 독출 데이터에 기초하여 계산된 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생할 수 있다.

효 과

[0028] 상기와 같은 본 발명의 실시예들에 따른 보안 메모리 인터페이스, 이를 포함하는 시스템 및 스마트카드는 하드웨어적인 구성을 채택하여 높은 보안성을 제공하면서도 데이터의 보안 레벨에 따라 동작하여 메모리의 효율적인 이용을 가능하게 한다.

[0029] 하드웨어적으로 구성된 보안 메모리 인터페이스에 의해 데이터의 무결성 여부를 정확하게 검출할 수 있을 뿐만 아니라, 보안 메모리 인터페이스는 데이터의 보안 레벨에 따라 선택적으로 동작하므로 전력 절감 및 메모리의 효율적 이용이 가능하다.

[0030] 또한 본 발명의 실시예들에 따른 보안 메모리 인터페이스를 채택한 시스템에서는 암호화 코드를 저장하기 위한 특정한 메모리 블록이 불필요하고 메모리 용량의 증가에 따른 추가적인 비용의 발생을 감소할 수 있다.

[0031] 나아가, 본 발명의 실시예들에 따른 보안 메모리 인터페이스는, 데이터 버스와 프로세서 사이에 배치되고, 버스를 통한 외부의 공격에 의한 보안 데이터의 오류를 정확하게 검출할 수 있다.

발명의 실시를 위한 구체적인 내용

[0032] 본문에 개시되어 있는 본 발명의 실시예들에 대해서, 특정한 구조적 내지 기능적 설명들은 단지 본 발명의 실시예를 설명하기 위한 목적으로 예시된 것으로, 본 발명의 실시예들은 다양한 형태로 실시될 수 있으며 본문에 설명된 실시예들에 한정되는 것으로 해석되어서는 아니 된다.

[0033] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는바, 특정 실시예들을 도면에 예시하고 본문에 상세하게 설명하고자 한다. 그러나 이는 본 발명을 특정한 개시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0034] 제 1, 제 2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로 사용될 수 있다. 예를 들어, 본 발명의 권리 범위로부터 이탈되지 않은 채 제 1 구성요소는 제 2 구성요소로 명명될 수 있고, 유사하게 제 2 구성요소도 제 1 구성요소로 명명될 수 있다.

[0035] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0036] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 설시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0037] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미이다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미인 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0038] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

[0039] 도 1은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스를 포함하는 시스템을 나타내는 블록도이다.

[0040] 도 1을 참조하면, 시스템(1000)은 프로세서(300), 보안 메모리 인터페이스(Secure Memory Interface, 500) 및 메모리 장치(700)를 포함한다.

[0041] 메모리 장치(700)에는 데이터가 저장되고, 프로세서(300)는 메모리 장치(700)의 액세스를 제어한다. 보안 메모리 인터페이스(500)는 메모리 장치(700)와 프로세서(300) 사이의 데이터 전송을 매개한다.

[0042] 프로세서(300)는 제어 신호(CTRL) 및 기입 데이터(DW)를 발생하여 보안 메모리 인터페이스(500)에 제공한다. 보안 메모리 인터페이스(500)는 프로세서(300)로부터 제어 신호(CTRL) 및 기입 데이터(DW)를 수신하고, 메모리 기입 데이터(MDW)를 발생한다. 또한 보안 메모리 인터페이스(500)는 메모리 장치로부터 메모리 독출 데이터(MDR)를 수신하고 독출 데이터(MDW) 및 에러 신호(ERR)를 발생한다.

[0043] 제어 신호(CTRL)는, 통상의 액세스인지 보안 액세스인지를 나타내는 보안 인에이블 신호(EN_SMI), 기입 동작 또는 독출 동작을 나타내는 모드 신호, 기입 데이터(DW) 및 독출 데이터(DR)의 비트수를 나타내는 신호, 타이밍 제어 신호 등을 포함할 수 있다. 통상적인 어드레스 신호, 클록 신호 등은 설명의 편의상 그 도시를 생략한다.

[0044] 보안 메모리 인터페이스(500)는 보안 인에이블 신호(EN_SMI)가 활성화된 경우 프로세서(300)로부터 수신된 기입 데이터(DW)에 기초하여 암호화된 메모리 기입 데이터(MDW)를 발생하고, 보안 인에이블 신호(EM_SMI)가 활성화된 경우 메모리 장치(700)로부터 수신된 암호화된 메모리 독출 데이터(MDR)에 기초하여 독출 데이터(DR) 및 독출 데이터(DR)의 에러 여부를 나타내는 에러 신호(ERR)를 발생한다. 한편, 보안 인에이블 신호가 비활성화된 경우 보안 메모리 인터페이스(500)는, 기입 동작시 기입 데이터(DW)와 동일한 메모리 기입 데이터(MDW)를 발생하고, 독출 동작시 메모리 독출 데이터(MDR)와 동일한 독출 데이터(DR)를 발생한다.

[0045] 이와 같이, 보안 메모리 인터페이스(500)는 통상의 액세스인 경우, 즉 보안 인에이블 신호(EN_SMI)가 비활성화된 경우에는 통상의 메모리 인터페이스와 같은 동작을 수행하고, 보안 액세스인 경우, 즉 보안 인에이블 신호(EN_SMI)가 활성화된 경우에는 기입 데이터(DW)를 암호화하여 메모리 장치(700)에 기입하고 암호화된 메모리 독출 데이터(MDR)에 기초하여 독출 데이터(DR)를 추출하고 독출 데이터(DR)의 에러 여부를 판별한다.

[0046] 일 실시예에서, 보안 메모리 인터페이스(500)는 메모리 독출 데이터(MDR)에 포함된 독출 에러 검출 코드 및 메모리 독출 데이터(MDR)로부터 추출된 독출 데이터(DR)에 기초하여 계산한 기준 에러 검출 코드를 비교하여 에러 신호(ERR)를 발생한다. 보안 메모리 인터페이스(500)의 더욱 상세한 설명은 도 3 내지 도 13의 실시예들을 참조하여 후술한다.

[0047] 일 실시예에서, 프로세서(300)는 메모리 장치(700)의 액세스를 제어할 뿐만 아니라 다른 기능을 수행하는 프로세서일 수 있다. 예를 들어, 프로세서(300)는 스마트카드, 컴퓨터 등의 중앙처리장치(Central Processing unit)일 수 있으며, 프로세서(300)와 보안 메모리 인터페이스(500)는 메모리 제어 장치(100)로서 하나의 칩에 집적될 수 있다.

[0048] 에러 신호(ERR)가 활성화된 경우, 프로세서(300)는 외부의 공격 등에 의해 독출 데이터(DR)에 오류가 있는 것으로 판단하고, 프로세서(300)의 동작을 즉시 인터럽트하거나 메모리 장치(700)에 저장된 데이터를 삭제 및/또는 갱신할 수 있다.

[0049] 일 실시예에서, 보안 메모리 인터페이스(500)를 포함하는 칩(100)은 버스(10)를 통하여 메모리 장치(700)와 연결될 수 있다. 버스(10)는 어드레스 버스와 데이터 버스를 포함할 수 있으며, 직렬 전송 방식 또는 병렬 전송 방식의 버스일 수 있다. 전송 방식에 따라 메모리 장치(700)와 메모리 제어 장치(100)는 직렬화기(serializer) 및/또는 병렬화기(deserializer) 등을 포함할 수 있다.

[0050] 도 2는 도1 의 시스템에 의한 데이터 전송을 나타내는 도면이다.

[0051] 예를 들어, 보안 인에이블 신호(EN_SMI)가 논리 로우 레벨로 비활성화된 경우 통상의 액세스(regular access)를 나타내고 논리 하이 레벨로 활성화된 경우 보안 액세스(secured access)를 나타낼 수 있다.

[0052] 통상의 액세스에 의한 기입 동작의 경우, 보안 메모리 인터페이스(500)는 프로세서(300)로부터 제공된 기입 데이터(DW)와 동일한 메모리 기입 데이터(MDW)를 메모리 장치(700)에 전송한다. 즉 n 비트 기입 데이터(D11) 및 2n 비트 기입 데이터(D12/D13)는 변동 없이 그대로 메모리 기입 데이터(MDW)로서 메모리 장치(700)에 전송된다.

[0053] 마찬가지로 통상의 액세스에 의한 독출 동작의 경우, 보안 메모리 인터페이스(500)는 메모리 장치(700)로부터 전송된 메모리 독출 데이터(MDR)와 동일한 독출 데이터(DR)를 프로세서(300)에 제공한다. 즉 보안 메모리 인터페이스(500)는 n 비트 독출 데이터(D21) 및 2n 비트 독출 데이터(D22/D23)를 수신하고 이를 그대로 독출 데이터(DR)로서 프로세서(300)에 제공한다.

[0054] 보안 액세스에 의한 기입 동작의 경우, 보안 메모리 인터페이스(500)는 기입 데이터(DW)를 암호화하고, 암호화된 메모리 기입 데이터(MDW)를 메모리 장치(700)에 전송한다. 즉 n 비트 기입 데이터(D14) 및 2n 비트 기입 데이터(D15/D16)는 암호화되어 비트수가 증가된 암호화된 기입 데이터 (SD11/SD12, SD13/SD14/SD15/SD16)가 메모

리 장치(700)로 전송된다.

[0055] 마찬가지로 보안 액세스에 의한 독출 동작의 경우, 보안 메모리 인터페이스(500)는 메모리 장치(700)로부터 전송된 암호화된 메모리 독출 데이터(MDR)로부터 독출 데이터(DR)를 추출하여 프로세서(300)에 제공한다. 즉 보안 메모리 인터페이스(500)는 암호화된 메모리 독출 데이터(SD21/SD22, SD23/SD24/SD25/SD26)로부터 n 비트 독출 데이터(D24) 및 2n 비트 독출 데이터(D25/D26)를 각각 분리하여 프로세서(300)에 제공한다.

[0056] 보안 액세스의 경우에는 기입 데이터(DW)와 메모리 기입 데이터(MDW) 사이에, 그리고 메모리 독출 데이터(MDR)와 독출 데이터(DR) 사이에 암호화 동작 및 에러 검출 동작을 위한 일정한 게이트 딜레이(Tgd)가 발생할 수 있다. 게이트 딜레이(Tgd)는 보안 메모리 인터페이스(700)의 하드웨어적인 구성에 따라 정해진다. 통상의 액세스인 경우에는 실질적으로 이러한 게이트 딜레이(Tgd) 없이 데이터의 전송이 이루어질 수 있다. 보안 메모리 인터페이스(700)의 하드웨어적인 구성에 의한 암호화 동작, 에러 검출 동작에 대해서는 도 3 이하의 도면을 참조하여 후술한다.

[0057] 일반적으로 메모리 장치에 저장된 데이터의 무결성(integrity)을 확인하는 방법으로서 소프트웨어적인 방법과 하드웨어적인 방법이 있다.

[0058] 소프트웨어적인 방법은 프로세서에 의해 수행되는 프로그램에 의해 코드를 발생하여 데이터에 이를 부가하는 방법이다. 이러한 소프트웨어적인 방법의 경우에는 프로세서에 의해 수행되는 프로그램 코드가 증가하고 프로그램이 복잡해지기 때문에 비교적 적은 수의 명령 집합을 신속하고 효율적으로 처리하는 것을 목적으로 하는 RISC(reduced instruction set computing) 프로세서 등에 적합하지 않다. 또한 소프트웨어적인 방법의 경우에는 암호화 알고리즘이 비교적 용이하게 파악될 수 있으므로 보안에 취약하다는 단점이 있다.

[0059] 하드웨어적인 방법은 메모리 장치에 암호화 코드를 저장하기 위한 여분의 메모리 블록을 확보하여 메모리 장치의 액세스가 수행될 때마다 암호화 코드를 상기 여분의 메모리 블록에 저장하고, 저장된 코드를 체크하는 방법이다. 하드웨어적인 방법의 경우에는 메모리 장치의 용량이 증가할수록 코드 저장을 위한 여분의 메모리 블록의 용량 또한 증가하기 때문에 비효율적이다. 또한 상기 여분의 메모리 블록을 이용한 하드웨어적인 방법의 경우에는 메모리 장치와 메모리 제어 회로 사이의 데이터 경로를 통한 외부의 공격에는 취약하다는 단점이 있다.

[0060] 본 발명의 일 실시예에 따른 보안 메모리 인터페이스(500)는 하드웨어적으로 구성되고 전송 데이터의 보안 레벨에 따라 통상의 액세스 동작 또는 보안 액세스 동작을 선택적으로 수행한다. 따라서 본 발명의 일 실시예에 따른 보안 메모리 인터페이스(500)는 데이터의 보안성을 향상시키고 데이터의 무결성 여부를 정확하게 검출할 수 있을 뿐만 아니라, 암호화 코드를 저장하기 위한 특정한 메모리 블록이 불필요하고 데이터의 보안 레벨에 따라 선택적으로 동작하므로 메모리를 효율적으로 이용할 수 있도록 한다.

[0061] 도 3은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스를 나타내는 블록도이다.

[0062] 도 3을 참조하면, 보안 메모리 인터페이스(500a)는 기입 회로(510) 및 독출 회로(520)를 포함한다.

[0063] 기입 회로(510)는, 보안 인에이블 신호(EN_SMI)가 활성화된 경우 프로세서(300)로부터 수신된 기입 데이터(DW)에 기초하여 암호화된 메모리 기입 데이터(MDW)를 발생한다. 전술한 바와 같이, 보안 인에이블 신호(EN_SMI)는 보안 메모리 인터페이스(500a)의 동작을 위한 제어 신호(CTRL)에 포함되는 신호이다. 독출 회로(520)는, 보안 인에이블 신호(EN_SMI)가 활성화된 경우 메모리 장치(700)로부터 수신된 암호화된 메모리 독출 데이터(MDR)에 기초하여 독출 데이터(DR) 및 독출 데이터(DR)의 에러 여부를 나타내는 에러 신호(ERR)를 발생한다.

[0064] 한편, 보안 인에이블 신호(EN_SMI)가 비활성화된 경우, 기입 회로(510)는 기입 데이터(DW)와 동일한 메모리 기입 데이터(MDW)를 발생하고, 독출 회로(520)는 메모리 독출 데이터(MDR)와 동일한 독출 데이터(DR)를 발생한다. 즉 통상의 액세스인 경우에는 보안 메모리 인터페이스(500a)는 수신된 데이터 신호를 그대로 통과시킨다는 점에서 일반적인 메모리 인터페이스와 같은 동작을 수행한다.

[0065] 기입 회로(510)는 제1 코드 계산기(512) 및 출력부(514)를 포함한다.

[0066] 제1 코드 계산기(512)는 기입 데이터(DW)에 기초하여 기입 에러 검출 코드(EDCW)를 발생한다. 출력부(514)는 보안 인에이블 신호(EN_SMI)에 응답하여, 기입 데이터(DW)와 기입 에러 검출 코드(EDCW)를 병합하여 메모리 기입 데이터(MDW)를 발생한다. 보안 인에이블 신호(EN_SMI)가 비활성화된 경우, 제1 코드 계산기(512) 및/또는 출력부(514)의 암호화 동작과 관련된 일부 구성은 디스에이블될 수 있으며, 이 경우 기입 회로(510)는 기입 데이터(DW)와 동일한 메모리 기입 데이터(MDW)를 출력한다.

[0067] 일 실시예에서, 출력부(514)는 수신된 기입 데이터(DW)를 메모리 기입 데이터(MDW)의 상위 비트들로 하고 수신된 기입 에러 검출 코드(EDCW)를 하위 비트들로 하여 메모리 기입 데이터(MDW)를 발생할 수도 있다. 반대로 수신된 기입 데이터(DW)를 메모리 기입 데이터(MDW)의 하위 비트들로 하고 수신된 기입 에러 검출 코드(EDCW)를 상위 비트들로 하여 메모리 기입 데이터(MDW)를 발생할 수도 있다. 다른 실시예에서, 출력부(514)는 수신된 기입 데이터(DW)와 기입 에러 검출 코드(EDCW)의 각 비트들을 인터리빙하여 메모리 기입 데이터(MDW)를 발생할 수도 있다.

[0068] 독출 회로(520)는 입력부(522), 제2 코드 계산기(524) 및 비교기(526)를 포함한다.

[0069] 입력부(522)는 보안 인에이블 신호(EN_SMI)에 응답하여 메모리 독출 데이터(MDR)에서 독출 에러 검출 코드(EDCR) 및 독출 데이터(DR)를 분리하여 각각 출력한다. 제2 코드 계산기(524)는 독출 데이터(DR)에 기초하여 기준 에러 검출 코드(EDCC)를 발생한다. 비교기(526)는 독출 에러 검출 코드(EDCR) 및 기준 에러 검출 코드(EDCC)를 비교하여 에러 신호(ERR)를 발생한다. 보안 인에이블 신호(EN_SMI)가 비활성된 경우, 입력부(522)의 암호화 동작과 관련된 일부 구성, 제2 코드 계산기(524) 및/또는 비교기(526)는 디스에이블될 수 있으며, 이 경우 독출 회로(520)는 메모리 독출 데이터(MDR)와 동일한 독출 데이터(DR)를 출력한다.

[0070] 입력부(522)는 기입 회로(512)에 포함된 출력부(514)의 구성과 상관되어 구현되고, 출력부(514)의 동작의 역순에 의해 독출 데이터(DR) 및 독출 에러 검출 코드(EDCR)를 분리하여 각각 출력한다.

[0071] 독출 회로(520)는 메모리 독출 데이터(MDR)에 포함된 독출 에러 검출 코드(EDCR) 및 메모리 독출 데이터(MDR)에 포함된 독출 데이터(DR)에 기초하여 계산된 기준 에러 검출 코드(EDCC)를 비교하여 에러 신호(ERR)를 발생한다. 데이터의 오류가 발생하지 않은 경우에는 수신된 독출 에러 검출 코드(EDCR)와 계산된 기준 에러 검출 코드(EDCC)가 일치하여 에러 신호(ERR)가 비활성화되고, 데이터의 오류가 발생한 경우에는 수신된 독출 에러 검출 코드(EDCR)와 계산된 기준 에러 검출 코드(EDCC)가 불일치하여 에러 신호(ERR)가 활성화된다.

[0072] 제어 신호(CTRL)에는 기입 동작 또는 독출 동작을 나타내는 모드 신호가 포함될 수 있다. 이러한 모드 신호에 응답하여 기입 회로(510)는 기입 동작시에만 인에이블되고 독출 회로(520)는 독출 동작시에만 활성화되도록 파워-다운 기능과 관련된 구성이 추가될 수 있다.

[0073] 도 3에는 기입 회로(510) 및 독출 회로(520)가 각각의 코드 계산기(512, 524)를 포함하는 실시예가 도시되어 있다. 이 경우 기입 회로(510)에 포함된 제1 코드 계산기(512) 및 독출 회로(520)에 포함된 제2 코드 계산기(524)는 동일한 구성의 논리 소자들로 구현된다.

[0074] 동일한 구성의 논리 소자들로 구현된다는 것은 동일한 입력에 대하여 동일한 코드가 출력된다는 것을 나타낸다. 따라서 데이터의 기입 과정에서, 메모리 장치(700)에 저장되어 있는 동안에, 그리고 독출 과정에서 아무런 오류가 발생되지 않은 경우에는, 기입 데이터(DW)와 독출 데이터(DR)가 동일하고, 기입 에러 검출 코드(EDCW), 독출 에러 검출 코드(EDCR) 및 기준 에러 검출 코드(EDCC)가 모두 동일하다. 반면에 오류가 발생된 경우에는, 메모리 장치(700)로부터 전송된 독출 에러 검출 코드(EDCR)와 메모리 장치(700)로부터 전송된 독출 데이터(DR)에 기초하여 계산된 기준 에러 검출 코드(EDCC)가 상이하다.

[0075] 이와 같이 비교기(526)는 독출 에러 검출 코드(EDCR)와 기준 에러 검출 코드(EDCC)의 동일 여부를 비교하여 에러 신호(ERR)를 발생한다.

[0076] 상기 설명한 바와 같이, 데이터와 코드의 병합 및 분리 방식에 따라서, 출력부(514)는 보안 인에이블 신호(EN_SMI)에 응답하여 기입 데이터(DW)와 기입 에러 검출 코드(EDCW)를 병합하기 위한 믹서(mixer), 인터리버(interleaver) 등을 포함할 수 있다. 입력부(522)는 보안 인에이블 신호(EN_SMI)에 응답하여 독출 데이터(DR)와 독출 에러 검출 코드(EDCR)를 분리하기 위한 디믹서(demixer), 디인터리버(deinterleaver) 등을 포함할 수 있다. 또한 출력부(514) 및/또는 입력부(522)는 채택된 데이터 전송 방식에 따라 요구되는 통상의 회로들, 예를 들어, 직렬화기/병렬화기(serializer/deserializer), 입출력 버퍼, 버스 구동 드라이버 등을 포함할 수 있다.

[0077] 도 4는 본 발명의 다른 실시예에 따른 보안 메모리 인터페이스를 나타내는 블록도이다.

[0078] 도 4를 참조하면, 보안 메모리 인터페이스(500b)는 기입 회로(530), 독출 회로(540) 및 코드 계산 회로(590)를 포함한다. 도 4에 도시된 보안 메모리 인터페이스(500b)의 기본적인 동작은 도 3에 도시된 보안 메모리 인터페이스(500a)의 동작과 유사하며 중복된 설명은 생략한다.

[0079] 도 3의 보안 메모리 인터페이스(500a)에서 기입 회로(510) 및 독출 회로(520)가 각각의 코드 계산기(512, 524)를 포함하는 것과 비교하여, 도 4의 보안 메모리 인터페이스(500b)는 논리 소자들로 구성된 하나의 공통 코드

계산기(592)를 포함한다.

[0080] 기입 회로(530)와 독출 회로(540)는 공통 코드 계산기(592)를 이용하여 기입 데이터(DW)에 기초하여 기입 에러 검출 코드(EDCW)를 발생하고 독출 데이터(DR)에 기초하여 기준 에러 검출 코드(EDCC)를 발생한다. 이를 위하여, 공통 코드 계산기(592)의 입력을 선택하기 위한 제1 스위치(594) 및 공통 코드 계산기(592)의 출력을 선택하기 위한 제2 스위치(596)를 더 포함할 수 있다. 제1 스위치(594)는 기입 데이터(DW) 또는 독출 데이터(DR)를 공통 코드 계산기(592)에 선택적으로 출력하고, 제2 스위치(596)는 공통의 코드 계산기(592)로부터 발생된 기입 에러 검출 코드(EDCW) 또는 기준 에러 검출 코드(EDCC)를 선택적으로 출력한다. 상기 설명한 바와 같이, 제어 신호(CTRL)에는 기입 동작 또는 독출 동작을 나타내는 모드 신호가 포함될 수 있고, 제1 스위치(594) 및 제2 스위치(596)는 상기 모드 신호에 응답하여 동작할 수 있다. 즉 상기 모드 신호에 따라, 기입 동작의 경우에는 스위치들(594, 596)의 입출력 단자들(NI, NO)은 기입 회로(530)와 연결하기 위한 단자들(N1, N3)과 연결되고, 독출 동작의 경우에는 스위치들(594, 596)의 입출력 단자들(NI, NO)은 독출 회로(540)와 연결되는 단자들(N2, N4)과 연결될 수 있다.

[0081] 기입 회로(530) 및 독출 회로(540)는 공통의 코드 계산기(592)를 이용하여 기입 에러 검출 코드(EDCW) 및 기준 에러 검출 코드(EDCC)를 발생한다. 따라서 데이터의 기입 과정에서, 메모리 장치(700)에 저장된 동안에, 그리고 독출 과정에서 아무런 오류가 발생되지 않은 경우에, 기입 데이터(DW)와 독출 데이터(DR)가 동일하고, 기입 에러 검출 코드(EDCW), 독출 에러 검출 코드(EDCR) 및 기준 에러 검출 코드(EDCC)가 모두 동일하다.

[0082] 반면에 오류가 발생된 경우에는, 메모리 장치(700)로부터 전송된 독출 에러 검출 코드(EDCR)와 메모리 장치(700)로부터 전송된 독출 데이터(DR)에 기초하여 계산된 기준 에러 검출 코드(EDCC)가 상이하다. 이와 같이 비교기(546)는 독출 에러 검출 코드(EDCR)와 기준 에러 검출 코드(EDCC)의 동일 여부를 비교하여 에러 신호(ERR)를 발생한다.

[0083] 이하에서는, 본 발명의 일 실시예에 따른 보안 메모리 인터페이스를 포함하는 시스템의 구성 및 동작을 더욱 상세히 설명한다. 후술하는 실시예들은 본 발명의 기술적 사상의 이해를 돕기 위한 예시적인 것이며, 이에 의해 본 발명의 기술적 범위가 한정되는 것으로 해석되어서는 안 될 것이다.

[0084] 도 5는 프로세서의 레지스터 구조의 일 예를 나타내는 도면이다.

[0085] 도 5를 참조하면, 도 1의 프로세서(300)는 복수의 일반 레지스터들(R0-R15), 복수의 확장 레지스터들(E8-E15), 그 밖의 특수한 목적의 레지스터들(PC, SPC_FIQ, SPC_IRQ, SR, SSR_FIQ, SSR_IRQ, SSR_SWI)을 포함할 수 있다. 프로세서(300)는 16 비트, 32 비트 그 밖의 임의의 비트 단위로 동작하는 프로세서일 수 있다. 이하에서는 설명의 편의를 위하여 16 비트 프로세서의 경우를 예로 들어 설명한다. 이 경우, 일반 레지스터들(R0-R15)은 16 비트, 확장 레지스터들(E8-E15)은 8 비트일 수 있다. 프로그램 카운터(PC)는 24 비트, 상태 레지스터(SR)는 8 비트일 수 있다.

[0086] 일반 레지스터들(R0-R15)은 프로세서(300)의 산술/논리 연산부(ALU; arithmatic and logic unit) 동작을 위한 소스 레지스터(source register) 또는 목적지 레지스터(destination register)가 될 수 있고, 메모리의 액세스 명령을 저장하는 인덱스 레지스터(index register)가 될 수 있다. 확장 레지스터들(E8-E15)은 도 5에 도시된 바와 같이 일반 레지스터들(R0-R15)과 결합되어 어드레스 레지스터들을 형성할 수 있다. 상기 결합에 의한 어드레스 레지스터들은 메모리 장치(700)의 액세스를 위한 어드레스를 저장하거나 확장된 비트수의 소스 레지스터 또는 목적지 레지스터가 될 수 있다.

[0087] 예를 들어, 16 비트 프로세서의 경우, 1 워드 명령을 위한 16 비트 명령어 또는 2 워드 명령을 위한 32 비트 명령어로 구성된 명령어 체계를 가질 수 있다. 또한, 프로세서(300)는 빅 엔디안 메모리 포맷(big endian memory format)을 채택할 수 있다. 이 경우 워드 데이터의 최상위 바이트는 메모리 장치(700)의 짝수 어드레스에 저장되고, 최하위 바이트는 홀수 어드레스에 저장된다. 예를 들어, 메모리 어드레스가 100h로 지정되고 메모리 기입 데이터가 1234h인 경우에 상위 바이트 12h는 메모리 어드레스 100h에 저장되고 하위 바이트 34h는 메모리 어드레스 101h에 저장된다.

[0088] 도 6은 통상의 액세스 또는 보안 액세스를 지정하는 명령어 맵의 일 예를 나타내는 도면이다.

[0089] 도 6에는 메모리 장치(700)의 액세스를 위한 명령어 세트 및 각 명령어에 대한 인덱스 레지스터의 각 비트(0-15)에 저장된 값을 나타내는 명령어 맵(instruction map)이 도시되어 있다. 도 6에 도시된 바와 같이, 본 발명의 일 실시예에 따른 보안 메모리 인터페이스 메모리를 포함하는 시스템에서, 프로세서의 명령어 세트는 통상의 액세스를 수행하기 위한 명령어(LDW, LDB)와 보안 액세스를 수행하기 위한 명령어(SLDW, SLDB)를 포함한다. LDW

및 SLDW는 워드 액세스임을 나타내고 LDB 및 SLDB는 바이트 액세스임을 나타낸다.

[0090] 예를 들어, 명령어 SLDW Dd, @[Ai]는 보안 액세스 모드에 의해 메모리 어드레스 Ai의 워드 데이터를 독출하여 목적지 레지스터 Dd에 저장할 것을 지시하고, 명령어 LDB @[Ai] Ds는 통상의 액세스 모드에 의해 소스 레지스터 Ds에 저장된 바이트 데이터를 메모리 어드레스 Ai에 기입할 것을 지시한다. 도 6에서 disp 및 edisp는 오프셋 상대 주소 지정을 위한 변위 및 확장 변위를 각각 나타낸다.

[0091] 이와 같이, 일 실시예에서, 명령어의 종류에 의해 통상의 액세스를 수행할 것인지 보안 액세스를 수행할 것인지가 결정될 수 있다. 도 6의 명령어 맵의 경우, 보안 액세스에 해당하는 명령어에 대하여 인덱스 레지스터의 최상위 3비트(즉, 제15, 14, 및 13 비트)의 값은 모두 1에 해당한다. 프로세서(300)는 1 비트의 플래그 레지스터를 구비하고 상기 인덱스 레지스터의 최상위 3비트의 값을 논리곱 연산(AND operation)하여 상기 플래그 레지스터에 저장하고, 상기 플래그 레지스터의 출력을 보안 인에이블 신호(EN_SMI)로서 발생할 수 있다. 즉 상기 인덱스 레지스터의 최상위 3비트의 값이 모두 1인 경우에만 상기 플래그 레지스터에 1의 값이 저장되고 이 경우 상기 플래그 레지스터의 출력인 보안 인에이블 신호(EN_SMI)가 활성화될 수 있다.

[0092] 도 6의 명령어 맵을 갖는 프로세서의 경우에는, 명령어의 종류 자체로서 통상의 액세스인지 보안 액세스인지를 결정할 수 있으나, 다른 방법에 의해 통상의 액세스 또는 보안 액세스가 결정될 수도 있다. 예를 들어, 보안 액세스의 여부는 어드레스 필터링에 의해 결정될 수 있다. 즉 특정한 레지스터가 소스 레지스터(source register) 또는 목적지 레지스터(destination register)로 지정되는 경우에만 보안 액세스가 수행되도록 시스템이 구현될 수도 있다. 또한 메모리 장치의 특정 영역에 해당되는 메모리 어드레스가 지정되는 경우에만 보안 액세스가 수행되도록 시스템이 구현될 수도 있다. 이 경우 프로세서는 소스 레지스터/목적지 레지스터로 지정된 레지스터의 어드레스 또는 메모리의 어드레스를 디코딩하여 보안 인에이블 신호(EN_SMI)로서 발생하기 위한 어드레스 인코더 또는 플래그 레지스터를 포함할 수 있다

[0093] 도 7은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스에 포함된 기입 회로의 일 예를 나타내는 도면이다.

[0094] 도 7을 참조하면, 기입 회로(550)는 제1 코드 계산기(552) 및 출력부(554)를 포함한다.

[0095] 제1 코드 계산기(552)는 기입 데이터(DW)에 기초하여 기입 에러 검출 코드(EDCW)를 발생한다. 출력부(554)는 보안 인에이블 신호(EN_SMI)에 응답하여, 기입 데이터(DW)와 기입 에러 검출 코드(EDCW)를 병합하여 메모리 기입 데이터(MDW)를 발생한다. 제1 코드 계산기(552) 및 출력부(554)의 기본적인 동작은 도 3에 도시된 제1 코드 계산기(512) 및 출력부(514)의 동작과 실질적으로 동일하다.

[0096] 출력부(554)는 선택 회로(557), 믹서(mixer, 558) 및 기입 유한 상태 머신(write finite state machine, 559)을 포함한다.

[0097] 선택 회로(557)는 보안 인에이블 신호(EN_SMI) 및 레지스터의 폭(register width)을 나타내는 제1 신호(RW24, RW16, RW8)에 응답하여 메모리 액세스 폭(memory access width)을 나타내는 제2 신호(AW32, AW16, AW8)를 발생한다. 믹서(558)는 보안 인에이블 신호(EN_SMI) 및 제1 신호(RW24, RW16, RW8)에 응답하여 기입 데이터(DW) 및 기입 에러 검출 코드(EDCW)를 병합하여 출력한다. 기입 유한 상태 머신(559)은 믹서(558)의 출력을 수신하고, 제2 신호(AW32, AW16, AW8)에 응답하여 메모리 기입 데이터(MDW)를 출력한다.

[0098] 도 5 및 도 6과 관련하여 설명한 바와 같이, 메모리 장치(700)의 액세스를 제어하는 프로세서(300)는 16 비트 프로세서일 수 있고, 이 경우 프로세서(300)는 16 비트의 일반 레지스터들(R0-R15) 및 8 비트의 확장 레지스터들(E8-E15)을 포함할 수 있다. 일반 레지스터들(R0-R15)과 확장 레지스터들(E8-E15)이 결합된 24 비트의 어드레스 레지스터들은 소스 레지스터(source register) 또는 목적지 레지스터(destination register)로 이용될 수 있고, 이 경우 프로세서(300)는 8 비트 메모리 액세스, 16 비트 메모리 액세스뿐만 아니라 24 비트 메모리 액세스를 지원할 수 있다. 제1 신호(RW, RW16, RW8)는 이러한 경우의 소스 레지스터의 폭을 나타내며 제1 신호의 세 개의 비트들(RW, RW16, RW8)은 선택적으로 하나의 비트만이 1의 값을 갖는다.

[0099] 도 10을 참조하여 후술하는 바와 같이, 제1 코드 계산기(552)는 소스 레지스터의 폭(즉 메모리 장치(700)에 저장하기 위한 기입 데이터(DW)의 비트수)을 나타내는 제1 신호(RW, RW16, RW8)에 응답하여 동작할 수 있다. 믹서(558)는 기입 데이터(DW)의 비트수를 나타내는 제1 신호(RW, RW16, RW8)에 응답하여 기입 에러 검출 코드(EDCW)의 비트들을 조합하고 이를 기입 데이터(DR)에 병합할 수 있다.

[0100] 기입 유한 상태 머신(559)은 선택 회로(557)의 출력인 메모리 액세스 폭을 나타내는 제2 신호(AW32, AW16, AW8)에 응답하여 실제 메모리 장치(700)로 전송되는 메모리 기입 데이터(MDW)를 출력한다. 제2 신호의 세 개의 비

트들(AW32, AW16, AW8)은 선택적으로 하나의 비트만이 1의 값을 갖는다. 선택 회로(557)에 대해서는 도 9를 참조하여 후술한다. 상기 설명한 바와 같이, 기입 유한 상태 머신(559)에서 출력되는 메모리 기입 데이터(MDW)는 보안 인에이블 신호(EN_SMI)가 비활성화된 경우에는 기입 데이터(DW)와 동일하고, 보안 인에이블 신호(EN_SMI)가 활성화된 경우에는 기입 데이터(DW)에 기입 에러 검출 코드(EDCW)가 병합된 암호화된 데이터이다.

[0101] 기입 유한 상태 머신(559)은, 도 7에 도시하지는 않았으나, 상기 설명한 바와 같이, 프로세서(300)로부터 제공된 제어 신호(CTRL)에 포함된 타이밍 제어 신호, 기입 동작 또는 독출 동작을 나타내는 모드 신호 등을 수신하여 동작할 수 있고, 기입 동작을 위한 메모리 어드레스 신호(MADD) 및 기입 인에이블 신호(WE)를 출력할 수 있다.

[0102] 만일 보안 메모리 인터페이스(500)와 메모리 장치(700)를 연결하는 버스의 폭이 24 비트로 한정된 시스템이라면, 32 비트 메모리 액세스 폭을 나타내는 제2 신호의 비트가 활성화된 경우 32 비트의 데이터가 한 번의 액세스 동작에 의해 전송될 수 없다. 이 경우, 기입 유한 상태 머신(559)은 32 비트의 메모리 기입 데이터(MDW)를 분할하여 이를 순차적으로 출력함으로써 두 번의 기입 동작을 수행할 수 있다.

[0103] 도 8은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스에 포함된 독출 회로의 일 예를 나타내는 도면이다.

[0104] 도 8을 참조하면, 독출 회로(560)는 입력부(564), 제2 코드 계산기(562) 및 비교기(566)를 포함한다.

[0105] 입력부(564)는 보안 인에이블 신호(EN_SMI)에 응답하여, 메모리 독출 데이터(MDR)에서 독출 에러 검출 코드(EDCR) 및 독출 데이터(DR)를 분리하여 각각 출력한다. 제2 코드 계산기(562)는 독출 데이터(DR)에 기초하여 기준 에러 검출 코드(EDCC)를 발생한다. 비교기(566)는 독출 에러 검출 코드(EDCR) 및 기준 에러 검출 코드(EDCC)를 비교하여 에러 신호(ERR)를 발생한다. 입력부(564), 제2 코드 계산기(562) 및 비교기(566)의 기본적인 동작은 도 3에 도시된 입력부(522), 제2 코드 계산기(524) 및 비교기(526)의 동작과 실질적으로 동일하다.

[0106] 입력부(564)는 선택 회로(567), 독출 유한 상태 머신(read finite state machine, 569), 및 디믹서(demixer, 568)를 포함한다.

[0107] 선택 회로(567)는 보안 인에이블 신호(EN_SMI) 및 레지스터의 폭을 나타내는 제1 신호(RW24, RW16, RW8)에 응답하여 메모리 액세스 폭을 나타내는 제2 신호(AW32, AW16, AW8)를 발생한다. 독출 유한 상태 머신(569)은 제2 신호(AW32, AW16, AW8)에 응답하여 메모리 독출 데이터(MDR)를 수신하여 디믹서(568)에 출력한다.

[0108] 디믹서(568)는 독출 유한 상태 머신(569)의 출력을 수신하고, 보안 인에이블 신호(EN_SMI) 및 제1 신호(RW24, RW16, RW8)에 응답하여 독출 에러 검출 코드(EDCR) 및 독출 데이터(DR)를 분리하여 각각 출력한다.

[0109] 독출 유한 상태 머신(569)은 선택 회로(557)의 출력인 메모리 액세스 폭을 나타내는 제2 신호(AW32, AW16, AW8)에 응답하여 메모리 장치(700)로부터 전송되는 메모리 독출 데이터(MDR)를 출력한다. 상기 설명한 바와 같이, 독출 유한 상태 머신(569)은 보안 인에이블 신호(EN_SMI)가 비활성화된 경우에는 독출 데이터(DR)와 동일한 메모리 독출 데이터(MDR)를 수신하고, 보안 인에이블 신호(EN_SMI)가 활성화된 경우에는 독출 데이터(DR)와 독출 에러 검출 코드(EDCR)가 병합된 암호화된 메모리 독출 데이터(MDR)를 수신한다.

[0110] 독출 유한 상태 머신(569)은, 도 8에 도시하지는 않았으나, 상기 설명한 바와 같이, 프로세서(300)로부터 제공된 제어 신호(CTRL)에 포함된 타이밍 제어 신호, 기입 동작 또는 독출 동작을 나타내는 모드 신호 등을 수신하여 동작할 수 있고, 독출 동작을 위한 메모리 어드레스 신호(MADD) 및 독출 인에이블 신호(RE)를 출력할 수 있다.

[0111] 디믹서(568)는 도 7의 믹서(558)의 동작을 역순으로 수행하여 독출 에러 검출 코드(EDCR) 및 독출 데이터(DR)를 분리하여 각각 출력한다. 선택 회로(567) 및 제2 코드 계산기(562)의 구성 및 동작은 도 7의 선택 회로(557)와 제1 코드 계산기(552)의 구성 및 동작과 실질적으로 동일하다. 한편, 도 4의 실시예와 관련하여 설명한 바와 같이, 제1 코드 계산기(552) 및 제2 코드 계산기(562)는 하나의 코드 계산기로 대체될 수 있으며, 동일한 구성을 갖는 기입 회로(550)의 선택 회로(557)와 독출 회로(560)의 선택 회로(567)도 하나의 선택 회로로 대체될 수 있다.

[0112] 도 9는 도 7의 기입 회로 및 도 8의 독출 회로에 포함된 선택 회로의 일 예를 나타내는 회로도이다.

[0113] 도 9를 참조하면, 선택 회로(557, 567)는 인버터(21), 제1 내지 제4 논리곱 게이트(AND gate)들(22, 23, 24, 25), 제1 및 제2 논리합 게이트(OR gate)들(26, 27)을 포함한다.

[0114] 상기 설명한 바와 같이, 선택 회로(557, 567)는 보안 인에이블 신호(EN_SMI) 및 선택적으로 하나의 비트만이 1

의 값을 갖는 제1 신호(RW24, RW16, RW8)에 응답하여 제2 신호(AW32, AW16, RW8)를 발생한다, 제2 신호(AW32, AW16, RW8)는 선택적으로 하나의 비트만이 1의 값을 갖는다.

[0115] 보안 인에이블 신호(EN_SMI)가 논리 로우 레벨로 비활성화된 경우에, 제1 신호의 제1 비트(RW24)가 1의 값을 가지면 제2 신호의 제1 비트(AW32)만이 1의 값을 갖고, 제1 신호의 제2 비트(RW16)가 1의 값을 가지면 제2 신호의 제2 비트(AW16)만이 1의 값을 갖고 제1 신호의 제3 비트(RW8)가 1의 값을 가지면 제2 신호의 제3 비트(AW8)만이 1의 값을 갖는다.

[0116] 한편, 보안 인에이블 신호(EN_SMI)가 논리 하이 레벨로 활성화된 경우에, 제1 신호의 제1 비트(RW24)가 1의 값을 가지면 제2 신호의 제1 비트(AW32)만이 1의 값을 갖고, 제1 신호의 제2 비트(RW16)가 1의 값을 가지면 제2 신호의 제1 비트(AW32)만이 1의 값을 갖고, 제1 신호의 제3 비트(RW8)가 1의 값을 가지면 제2 신호의 제2 비트(AW16)만이 1의 값을 갖는다.

[0117] 즉, 통상의 액세스 또는 보안 액세스에 따른 기입 데이터(DW)와 메모리 기입 데이터(MDW)의 비트수의 관계를 요약하면 표 1과 같다. 독출 데이터(DR)와 메모리 독출 데이터(MDR)의 경우도 마찬가지이다.

[0118] [표 1]

	DW	MDW
EN_SMI = 0	8	8
	16	16
	24	32
EN_SMI = 1	8	16
	16	32
	24	32

[0119]

[0120] 도 10은 도 7의 기입 회로 및 도 8의 독출 회로에 포함된 코드 계산기의 일 예를 나타내는 블록도이다.

[0121] 도 10을 참조하면, 기입 회로(550)의 제1 코드 계산기(552)는 기입 에러 검출 코드(EDCR)의 하위 비트들(EDCl:8)을 계산하는 하위 계산기(32), 및 기입 에러 검출 코드(EDCR)의 상위 비트들(EDCh:8)을 계산하는 상위 계산기(36)를 포함하여 구성될 수 있다.

[0122] 도 10은, 16 비트 프로세서의 경우에 대한 것으로서, 기입 데이터(DW) 또는 독출 데이터(DR)의 비트수가 8비트인 경우 및 16 비트인 경우뿐만 아니라 어드레스 레지스터를 이용하여 24 비트의 데이터 전송을 수행할 수 있도록 구성된 코드 계산기의 일 실시예를 도시한 것이다.

[0123] 제1 바이트(Rl:8)는 도 5의 16 비트 일반 레지스터(Ri, i=0, 1, 2, ... ,15)의 하위 바이트(즉, 하위 8비트)의 값이고, 제2 바이트(Rh:8)는 상기 일반 레지스터(Ri)의 상위 바이트(즉, 상위 8비트)의 값이고, 제3 바이트(E:8)는 8 비트 확장 레지스터(Ei)의 바이트 값이다. 독출 회로(560)의 제2 코드 계산기(562)는 도 10의 구성과 동일하다. 다만 제1 코드 계산기(552)의 입력인 제1 바이트(Rl:8), 제2 바이트(Rh:8) 및 제3 바이트(E:8)는 상응하는 독출 데이터(DR)의 각 바이트로 대체되어 제2 코드 계산기(562)로 입력된다.

[0124] 하위 계산기(32)는 제1 바이트(Rl:8)에 기초하여 에러 검출 코드(EDC)의 하위 바이트(EDCl:8)를 발생하고, 상위 계산기(36)는 제2 바이트(Rh:8), 제3 바이트(E:8) 및 멀티플렉서(34)의 출력에 기초하여 에러 검출 코드(EDC)의 상위 바이트(EDCh:8)를 발생한다.

[0125] 멀티플렉서(34)는 제1 신호의 제2 비트(RW24)에 응답하여 디폴트 값인 0xFF 또는 하위 계산기(32)의 출력을 선택적으로 출력한다. 기입 데이터(DW) 또는 독출 데이터(DR)의 비트수가 24인 경우, 즉 소스 레지스터 또는 목적지 레지스터로 지정된 레지스터가 24 비트 어드레스 레지스터인 경우에는, 제1 신호의 제2 비트(RW24)의 값이 1이 되고 멀티플렉서(34)는 하위 계산기(32)의 출력을 선택하여 상위 계산기(36)에 출력한다. 기입 데이터(DW) 또는 독출 데이터(DR)의 비트수가 8 또는 16인 경우, 즉 소스 레지스터 또는 목적지 레지스터로 지정된 레지스터가 16 비트 일반 레지스터인 경우에는, 제1 신호의 제2 비트(RW24)의 값이 0이 되고 멀티플렉서(34)는 디폴트 값인 0xFF를 선택하여 상위 계산기(36)에 출력한다. 또한 제1 신호의 제2 비트(RW24)의 값이 0인 경우에는 확장 레지스터(Ei)의 값인 제3 바이트(Ei:8)는 0x00으로 셋팅된다.

[0126] 도 11은 도 10의 독출 회로에 포함된 하위 계산기의 일 예를 나타내는 도면이다.

[0127] 도 11을 참조하면, 하위 계산기(32)는 복수의 노어 게이트(NOR gate)와 복수의 인버터들로 구성된 제1 게이트 어레이(GA1), 제2 게이트 어레이(GA2), 및 제3 게이트 어레이(GA3)를 포함한다. 도11의 하위 계산기(32)는 2 노어 게이트 딜레이 및 1 인버터 딜레이를 갖는다. 도 2에 도시된 게이트 딜레이(Tgd)는 이러한 코드 계산기의 게이트 딜레이와 연관된다.

[0128] 도 11의 구성에 의한 하위 계산기(32)가 출력하는 에러 검출 코드(EDC)의 하위 바이트(EDCl:8), 즉 EDCl[0] 내지 EDCl[7]의 값은 다음과 같다.

[0129] EDCl[0] = !(D[7] ^ D[6] ^ D[0]);

[0130] EDCl[1] = !(D[6] ^ D[1] ^ D[0]);

[0131] EDCl[2] = !(D[6] ^ D[2] ^ D[1] ^ D[0]);

[0132] EDCl[3] = D[7] ^ D[3] ^ D[2] ^ D[1];

[0133] EDCl[4] = !(D[4] ^ D[3] ^ D[2]);

[0134] EDCl[5] = D[5] ^ D[4] ^ D[3] ;

[0135] EDCl[6] = !(D[6] ^ D[5] ^ D[4]);

[0136] EDCl[7] = !(D[7] ^ D[6] ^ D[5]);

[0137] 여기서, D[0] 내지 D[7]은 하위 계산기(32)의 입력인 제1 바이트(R1:8)의 각 비트 값을 나타낸다. (^)는 노어 연산(NOR operation)을 나타내고 (!)는 부정 연산(not operation)을 나타낸다.

[0138] 도 12는 도 10의 독출 회로에 포함된 상위 계산기의 일 예를 나타내는 도면이다.

[0139] 도 12를 참조하면, 상위 계산기(36)는 복수의 노어 게이트(NOR gate)와 복수의 인버터들로 구성된 제4 게이트 어레이(GA4), 제5 게이트 어레이(GA5), 제6 게이트 어레이(GA6) 및 제7 게이트 어레이(GA7)를 포함한다. 도12의 상위 계산기(36)는 4 노어 게이트 딜레이를 갖는다.

[0140] 도 12의 구성에 의한 상위 계산기(36)가 출력하는 에러 검출 코드(EDC)의 상위 바이트(EDCh:8), 즉 EDCh[0] 내지 EDCh[7]의 값은 다음과 같다.

[0141] EDCh[0] = D[14] ^ D[12] ^ D[8] ^ D[7] ^ D[6] ^ D[0] ^ C[0] ^ C[4] ^ C[6];

[0142] EDCh[1] = D[15] ^ D[14] ^ D[13] ^ D[12] ^ D[9] ^ D[6] ^ D[1] ^ D[0] ^ C[1] ^ C[4] ^ C[5] ^ C[6] ^ C[7];

[0143] EDCh[2] = D[15] ^ D[13] ^ D[12] ^ D[10] ^ D[8] ^ D[6] ^ D[2] ^ D[1] ^ D[0] ^ C[0] ^ C[2] ^ C[4] ^ C[5] ^ C[7];

[0144] EDCh[3] = D[14] ^ D[13] ^ D[11] ^ D[9] ^ D[7] ^ D[3] ^ D[2] ^ D[1] ^ C[1] ^ C[3] ^ C[5] ^ C[6];

[0145] EDCh[4] = D[15] ^ D[14] ^ D[12] ^ D[10] ^ D[8] ^ D[4] ^ D[3] ^ D[2] ^ C[0] ^ C[2] ^ C[4] ^ C[6] ^ C[7];

[0146] EDCh[5] = D[15] ^ D[13] ^ D[11] ^ D[9] ^ D[5] ^ D[4] ^ D[3] ^ C[1] ^ C[3] ^ C[5] ^ C[7];

[0147] EDCh[6] = D[14] ^ D[12] ^ D[10] ^ D[6] ^ D[5] ^ D[4] ^ C[2] ^ C[4] ^ C[6];

[0148] EDCh[7] = D[15] ^ D[13] ^ D[11] ^ D[7] ^ D[6] ^ D[5] ^ C[3] ^ C[5] ^ C[7];

[0149] 여기서, D[0] 내지 D[7]은 상위 계산기(36)의 입력인 제2 바이트(Rh:8)의 각 비트 값을 나타내고, D[8] 내지 D[15]는 상위 계산기(36)의 입력인 제3 바이트(E:8)의 각 비트 값을 나타내고, C[0] 내지 C[7]는 멀티플렉서(34)의 출력의 각 비트 값을 나타낸다. (^)는 노어 연산(NOR operation)을 나타내고 (!)는 부정 연산(not operation)을 나타낸다.

[0150] 이하 도 10 내지 도12의 구성에 의한 제1 코드 계산기(552)에서 출력되는 코드 값의 계산 결과를 예시한다.

[0151] 기입 데이터(DW)의 비트수가 8(즉, RW8=1)일 때에는 제1 바이트(R1:8)만이 코드 계산에 이용된다. 예를 들어, 제1 바이트(R1:8)가 0x01의 값인 경우 에러 검출 코드의 하위 바이트(EDC1:8)의 값은 0xD0의 값이 되고, 제1 바이트(R1:8)가 0x03의 값인 경우 에러 검출 코드의 하위 바이트(EDC1:8)의 값은 0xDE가 된다. 이 경우, 에러 검

출 코드의 상위 바이트(EDCh:8)는 무시되고 하위 바이트(EDCl:8)만이 기입 에러 검출 코드(EDCW)로서 이용될 수 있다.

[0152] 기입 데이터(DW)의 비트수가 16(즉, RW16=1)일 때에는 제1 바이트(Rl:8) 및 제2 바이트(Rh:8)가 코드 계산에 이용되고 제3 바이트(E:8)는 0x00의 값으로 세팅된다. 멀티플렉서(34)는 디폴트 값인 0xFF를 출력으로 선택하여 상위 계산기(36)에 제공한다. 예를 들어, 제1 바이트(Rl:8)가 0x01의 값이고 제2 바이트(Rh:8)가 0x03의 값인 경우 에러 검출 코드의 하위 바이트(EDCl:8)의 값은 0xD0의 값이 되고 에러 검출 코드의 상위 바이트(EDCh:8)의 값은 0xDE의 값이 된다. 반대로 제1 바이트(Rl:8)가 0x03의 값이고 제2 바이트(Rh:8)가 0x01의 값인 경우 에러 검출 코드의 하위 바이트(EDCl:8)의 값은 0xDE의 값이 되고 에러 검출 코드의 상위 바이트(EDCh:8)의 값은 0xD0의 값이 된다.

[0153] 결과적으로 16 비트 보안 액세스의 경우에 하위 계산기(32)와 상위 계산기(36)는 제1 바이트(Rl:8) 및 제2 바이트(Rh:8)를 각각의 입력으로 하여 동일한 논리 연산을 수행함을 알 수 있다.

[0154] 이와는 달리, 기입 데이터(DW)의 비트수가 24(즉, RW24=1)인 경우에는 제1 바이트(Rl:8), 제2 바이트(Rh:8) 및 제3 바이트(E:8)가 모두 코드 계산에 이용된다. 멀티플렉서(34)는 하위 계산기(32)의 출력을 선택하여 상위 계산기(36)에 제공한다. 예를 들어, 제1 바이트(Rl:8), 제2 바이트(Rh:8) 및 제3 바이트(E:8)가 각각 0x01, 0x00, 0x00의 값인 경우 에러 검출 코드의 상위 바이트(EDCh:8)의 값은 0xBA의 값이 된다. 제1 바이트(Rl:8), 제2 바이트(Rh:8) 및 제3 바이트(E:8)가 각각 0x00, 0x01, 0x00의 값인 경우 에러 검출 코드의 상위 바이트(EDCh:8)의 값은 0xD6의 값이 된다. 제1 바이트(Rl:8), 제2 바이트(Rh:8) 및 제3 바이트(E:8)가 각각 0x00, 0x00, 0x01의 값인 경우 에러 검출 코드의 상위 바이트(EDCh:8)의 값은 0xC4의 값이 된다.

[0155] 결과적으로 24 비트 보안 액세스의 경우에는, 입력인 제1 바이트(Rl:8), 제2 바이트(Rh:8) 및 제3 바이트(E:8)가 모두 연관되어 에러 검출 코드의 상위 바이트(EDCh:8)가 결정됨을 알 수 있다. 이 경우, 에러 검출 코드의 하위 바이트(EDCl:8)는 무시되고 상위 바이트(EDCh:8)만이 기입 에러 검출 코드(EDCW)로서 이용될 수 있다.

[0156] 이하에서는, 도 7 내지 도 11을 참조하여, 보안 인에이블 신호(EN_SMI)가 활성화된 경우에 해당하는 보안 액세스 동작을 설명한다.

[0157] 도 10 및 도 11에 예시된 코드 계산기(552, 562)에 의해 출력되는 에러검출 코드의 하위 바이트(EDCl:8) 및 상위 바이트(EDCh:8)는 제1 신호(RW24, RW16, RW8)에 응답하여 선택적으로 출력될 수 있다.

[0158] 예를 들어, 기입 동작시 제1 신호의 제1 비트(RW24)가 1의 값인 경우에는 에러 검출 코드의 상위 바이트(EDCh:8)만이 기입 에러 검출 코드(EDCW)로서 도 7의 믹서(558)에 제공되어 24 비트의 기입 데이터(DW)와 병합되고, 메모리 기입 데이터(MDW)는 32 비트가 된다. 이 경우에는 24 비트의 기입 데이터(DW)에 대하여 8 비트의 기입 에러 검출 코드(EDCW)가 병합된다는 점에서 상대적으로 낮은 정도의 보안 레벨(relatively lower secure level)이라고 할 수 있다. 한편 독출 동작시에는 에러 검출 코드의 상위 바이트(EDCh:8)만이 도 8의 비교기(566)에 기준 에러 검출 코드(EDCC)로서 제공되어 상응하는 8 비트의 독출 에러 검출 코드(EDCR)와 비교된다.

[0159] 예를 들어, 기입 동작시 제2 신호의 제2 비트(RW16)가 1의 값인 경우에는 에러 검출 코드의 하위 바이트(EDCl:8) 및 상위 바이트(EDCh:8)가 모두 기입 에러 검출 코드(EDCW)로서 도 7의 믹서(558)에 제공되어 16 비트의 기입 데이터(DW)와 병합되고, 메모리 기입 데이터(MDW)는 32 비트가 된다. 이 경우에는 16 비트의 기입 데이터(DW)에 대하여 16 비트의 기입 에러 검출 코드(EDCW)가 병합된다는 점에서 상대적으로 높은 정도의 보안 레벨(relatively higher secure level)이라고 할 수 있다. 한편 독출 동작시에는 에러 검출 코드의 하위 바이트(EDCl:8) 및 상위 바이트(EDCh:8)가 모두 도 8의 비교기(566)에 기준 에러 검출 코드(EDCC)로서 제공되어 상응하는 16 비트의 독출 에러 검출 코드(EDCR)와 비교된다.

[0160] 예를 들어, 기입 동작시 제1 신호의 제3 비트(RW8)가 1의 값인 경우에는 에러 검출 코드의 하위 바이트(EDCl:8)만이 기입 에러 검출 코드(EDCW)로서 도 7의 믹서(558)에 제공되어 8 비트의 기입 데이터(DW)와 병합되고, 메모리 기입 데이터(MDW)는 16 비트가 된다. 마찬가지로. 8 비트의 기입 데이터(DW)에 대하여 8 비트의 기입 에러 검출 코드(EDCW)가 병합된다는 점에서 상대적으로 높은 정도의 보안 레벨(relatively higher secure level)이라고 할 수 있다. 한편 독출 동작시에는 에러 검출 코드의 하위 바이트(EDCl:8)만이 도 8의 비교기(566)에 기준 에러 검출 코드(EDCC)로서 제공되어 상응하는 8 비트의 독출 에러 검출 코드(EDCR)와 비교된다.

[0161] 도 7 및 도 8에는 레지스터의 폭을 나타내는 제1 신호(RW24, RW16, RW8)에 의해서 상대적으로 낮은 정도 또는 높은 정도의 보안 레벨이 결정될 수 있는 기입 회로(550) 및 독출 회로(560)의 실시예가 도시되어 있다. 이와는 다르게, 도 6의 명령어 맵을 변경함으로써 보안 레벨을 다양하게 결정할 수도 있다. 즉 도 6의 명령어 맵에서는

명령어의 종류 자체로서 통상의 액세스인지 보안 액세스인지가 결정될 수 있으나, 보안 액세스를 나타내는 명령어를 세분화하여 명령어 맵을 구성함으로써 상대적으로 낮은 정도의 보안 레벨 또는 상대적으로 높은 정도의 보안 레벨을 구현할 수 있다.

[0162] 도 11 및 도 12에는 16 비트 프로세서의 경우에 24 비트 메모리 액세스를 지원하면서도 적절한 게이트 딜레이(Tgd)를 갖도록 하위 계산기(32)와 상위 계산기(36)가 서로 다른 구성을 갖는 예가 도시되어 있다. 하위 계산기(32)와 상위 계산기(36)는 논리 소자들의 다양한 구성으로 구성될 수 있다. 예를 들어, 2n 비트 프로세서가 n 비트 및 2n 비트 메모리 액세스만을 지원하도록 구성된 시스템에서는, 하위 계산기(32)와 상위 계산기(36)는 레지스터의 상위 비트들과 하위 비트들을 각각 입력으로 하여 기입 에러 검출 코드(EDCR) 또는 기준 에러 검출 코드(EDCC)의 하위 비트들과 상위 비트들을 각각 출력하도록 동일한 구성으로 구현될 수도 있다.

[0163] 또한, 도 11 및 도 12에는 노어 게이트들과 인버터들을 이용한 구성 예를 도시하고 있으나, 코드 계산기는 다양한 하드웨어 방식으로 구현될 수 있다. 예를 들어, 코드 계산기는 순환 중복 체크(CRC; Cyclic Redundancy Check), 단순 체크섬(checksum) 코드 등을 발생하는 하드웨어로 구현될 수 있으며, 시프트 레지스터, 논리곱 게이트, 논리합 게이트, 배타적 논리합 게이트, 배타적 논리곱 게이트 등의 다양한 논리 소자들을 이용하여 구현될 수 있다.

[0164] 도 13은 본 발명의 일 실시예에 따른 시스템에서 명령어의 실행에 따른 메모리 장치의 데이터 저장 상태를 나타내는 도면이다.

[0165] 제1 블록(40)은 프로세서(300)에 의하여 순차적으로 실행되는 일련의 명령어를 나타내고, 제2 블록(50)은 상기 일련의 명령어가 수행된 후의 메모리 장치의 데이터 저장 상태를 나타내고, 제3 블록(60)은 기입 동작시 소스 레지스터로 지정되는 16 비트 일반 레지스터(R0)와 24 비트 어드레스 레지스터(A8)의 값을 예시하고 있다. 도 13에는 기입 에러 검출 코드(EDCW) 및 기입 데이터(DW)와 단순히 바이트 단위로 조합되어 메모리 기입 데이터(MDW)로서 저장된 것을 예시하고 있다. 상기 설명한 바와 같이, 메모리 기입 데이터(MDW)는 인터리빙 방식으로 병합되어 생성될 수 있으며, 도 3의 출력부(514)의 구성에 따라 암호화된 메모리 기입 데이터(MDW)는 다양한 방식으로 병합되어 생성될 수 있다.

[0166] 도 13에 도시된 바와 같이, 각 기입 데이터(DW)의 보안 레벨에 따라 통상의 액세스 명령(LDB, LDW)과 보안 액세스 명령(SLDB, SLDW)을 선택적으로 실행함으로써 메모리를 효율적으로 관리할 수 있다. 또한 본 발명의 실시예에 따른 보안 인터페이스 메모리를 포함하는 시스템에서는, 일반적인 하드웨어적인 보안 방식과는 다르게 메모리 장치의 용량이 증가하더라도 보안을 위한 추가적인 메모리 용량의 증가가 요구되지 않는다.

[0167] 도 14는 본 발명의 일 실시예에 따른 보안 메모리 인터페이스를 포함하는 스마트카드를 나타내는 블록도이다.

[0168] 도 14를 참조하면, 스마트카드(1200)는 적어도 하나의 메모리 장치(700a, 700b, 700c, 700d), 메모리 제어 장치(100), 버스(10a) 및 호스트 인터페이스(800)를 포함한다.

[0169] 각 메모리 장치(700a, 700b, 700c, 700d)는 데이터를 저장하고, 롬(ROM; read only memory), 램(RAM; random access memory), 불휘발성 메모리(NVM; non-volatile memory) 등 다양한 유형의 메모리를 포함할 수 있다.

[0170] 메모리 제어 장치(100)는 메모리 장치(700a, 700b, 700c, 700d)의 하나 이상에 대한 액세스를 제어하고, 메모리 제어 장치(100)와 메모리 장치(700a, 700b, 700c, 700d)는 버스(10a)를 통하여 연결된다. 호스트 인터페이스(800)는 버스(10a)에 호스트(3000)와의 통신을 매개한다. 메모리 제어 장치(100)는 메모리에 대한 액세스 제어뿐만 아니라, 스마트카드(2000)의 전반적인 동작을 제어하는 중앙처리장치의 코어(CPU Core)일 수 있다.

[0171] 메모리 제어 장치(100)는, 상기 설명한 바와 같은 보안 메모리 인터페이스(500)를 포함한다. 보안 메모리 인터페이스(500)는 보안 인에이블 신호가 활성화된 경우 프로세서로부터 수신된 기입 데이터에 기초하여 암호화된 메모리 기입 데이터를 발생하고, 상기 보안 인에이블 신호가 활성화된 경우 상기 메모리 장치로부터 수신된 암호화된 메모리 독출 데이터에 기초하여 독출 데이터 및 상기 독출 데이터의 에러 여부를 나타내는 에러 신호를 발생한다. 보안 메모리 인터페이스(500)는, 상기 메모리 독출 데이터에 포함된 독출 에러 검출 코드 및 상기 메모리 독출 데이터에 포함된 상기 독출 데이터에 기초하여 계산된 기준 에러 검출 코드를 비교하여 상기 에러 신호를 발생한다.

[0172] 스마트카드는 특정한 기능을 수행할 수 있도록 집적 회로 칩을 내장한 카드로서, 전자 화폐, 교통 카드, 전자 상거래, 출입 통제 등의 다양한 응용 분야에서 다목적으로 활용될 수 있다. 스마트카드는 그 특성상 보안 모듈 등의 수단이 포함될 것이 요구되고, 스마트카드의 데이터 전송은 주로 버스(10a)를 통하여 이루어진다. 버스를

통한 데이터 전송 방식은 버스 상의 전송 데이터가 쉽게 노출될 수 있고, 메모리의 감지 증폭기의 전력 소모를 파악함으로써 전송 데이터가 쉽게 노출될 수 있어 특히 보안에 취약하다.

[0173] 본 발명의 일 실시예에 따른 보안 메모리 인터페이스(500)는 메모리 제어 장치(100) 내에 집적되어, 메모리 장치의 직접적인 공격에 의해 오류가 발생한 경우뿐만 아니라 버스를 통한 검침(probing)과 같은 외부의 공격에 의해 오류가 발생한 경우에도 데이터의 오류 여부를 검출할 수 있다.

산업이용 가능성

[0174] 본 발명은 보안이 필요한 데이터의 오류 검출이 요구되는 장치 및 시스템에 유용하게 이용될 수 있다. 본 발명은 스마트카드 등과 같이 버스에 의한 데이터 전송을 수행하는 장치 및 시스템에 더욱 유용하게 이용될 수 있다.

[0175] 상기에서는 본 발명이 바람직한 실시예를 참조하여 설명하였지만, 해당 기술분야의 숙련된 당업자는 하기의 특허청구범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 것이다.

도면의 간단한 설명

[0176] 도 1은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스를 포함하는 시스템을 나타내는 블록도이다.

[0177] 도 2는 도1 의 시스템에 의한 데이터 전송을 나타내는 도면이다.

[0178] 도 3은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스를 나타내는 블록도이다.

[0179] 도 4는 본 발명의 다른 실시예에 따른 보안 메모리 인터페이스를 나타내는 블록도이다.

[0180] 도 5는 프로세서의 레지스터 구조의 일 예를 나타내는 도면이다.

[0181] 도 6은 통상의 액세스 또는 보안 액세스를 지정하는 명령어 맵의 일 예를 나타내는 도면이다.

[0182] 도 7은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스에 포함된 기입 회로의 일 예를 나타내는 도면이다.

[0183] 도 8은 본 발명의 일 실시예에 따른 보안 메모리 인터페이스에 포함된 독출 회로의 일 예를 나타내는 도면이다.

[0184] 도 9는 도 7의 기입 회로 및 도 8의 독출 회로에 포함된 선택 회로의 일 예를 나타내는 회로도이다.

[0185] 도 10은 도 7의 기입 회로 및 도 8의 독출 회로에 포함된 코드 계산기의 일 예를 나타내는 블록도이다.

[0186] 도 11은 도 10의 독출 회로에 포함된 하위 계산기의 일 예를 나타내는 도면이다.

[0187] 도 12는 도 10의 독출 회로에 포함된 상위 계산기의 일 예를 나타내는 도면이다.

[0188] 도 13은 본 발명의 일 실시예에 따른 시스템에서 명령어의 실행에 따른 메모리 장치의 데이터 저장 상태를 나타내는 도면이다.

[0189] 도 14는 본 발명의 일 실시예에 따른 보안 메모리 인터페이스를 포함하는 스마트카드를 나타내는 블록도이다.

[0190] <도면의 주요부분에 대한 부호의 설명>

[0191] 300: 프로세서 500: 보안 메모리 인터페이스

[0192] 700: 메모리 장치 510, 530, 550: 기입 회로

[0193] 520, 540, 560: 독출 회로 514, 534, 554: 출력부

[0194] 512, 524, 552, 552, 562: 코드 계산기

[0195] 526, 546: 비교기 522, 542, 564: 입력부

[0196] 559, 569: 유한 상태 머신 32: 하위 계산기

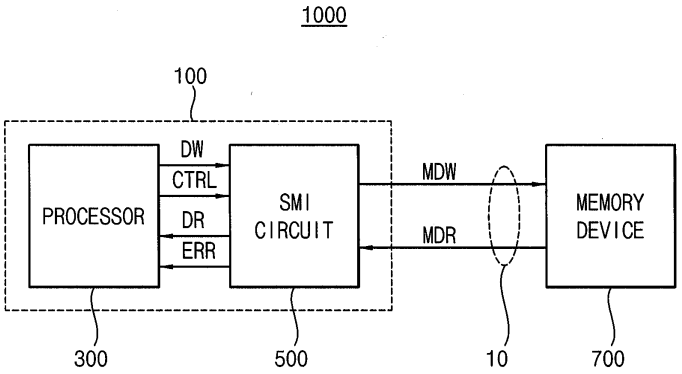
[0197] 36: 상위 계산기 10, 10a: 버스

[0198] DW: 기입 데이터 MDW: 메모리 기입 데이터

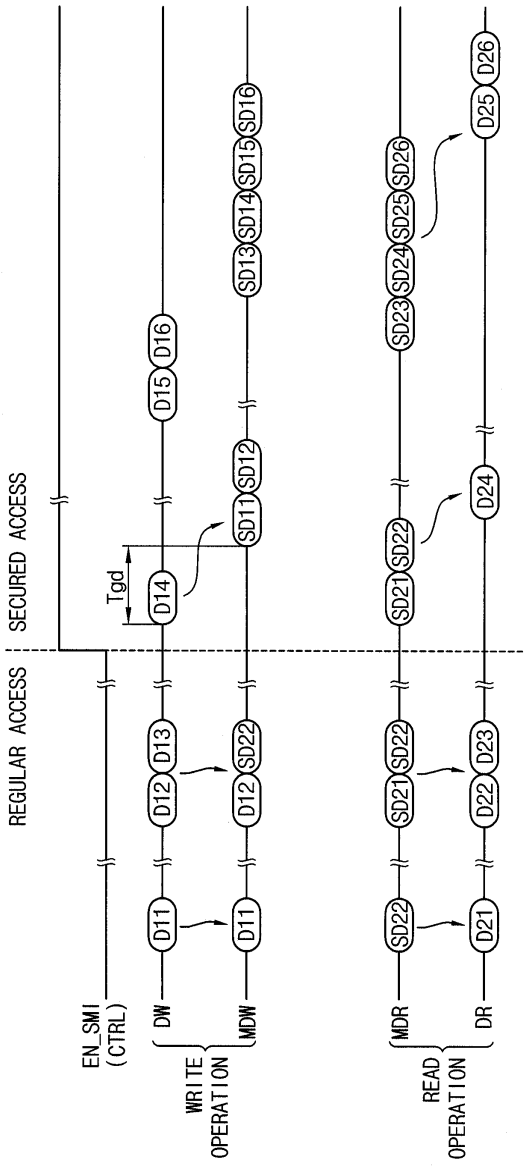
[0199] DR: 독출 데이터 MDR: 메모리 독출 데이터

도면

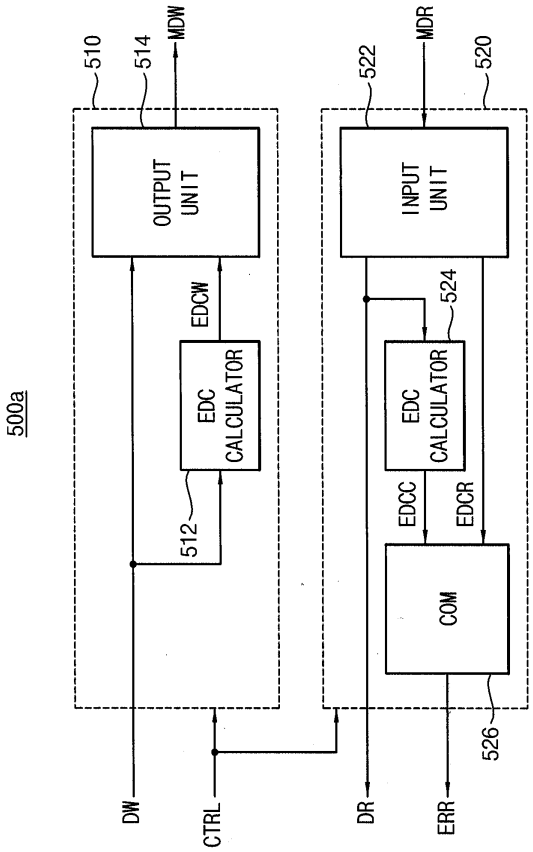
도면1


1000
100
PROCESSOR
DW
CTRL
DR
ERR
SMI
CIRCUIT
MDW
MDR
MEMORY
DEVICE
300
500
10
700

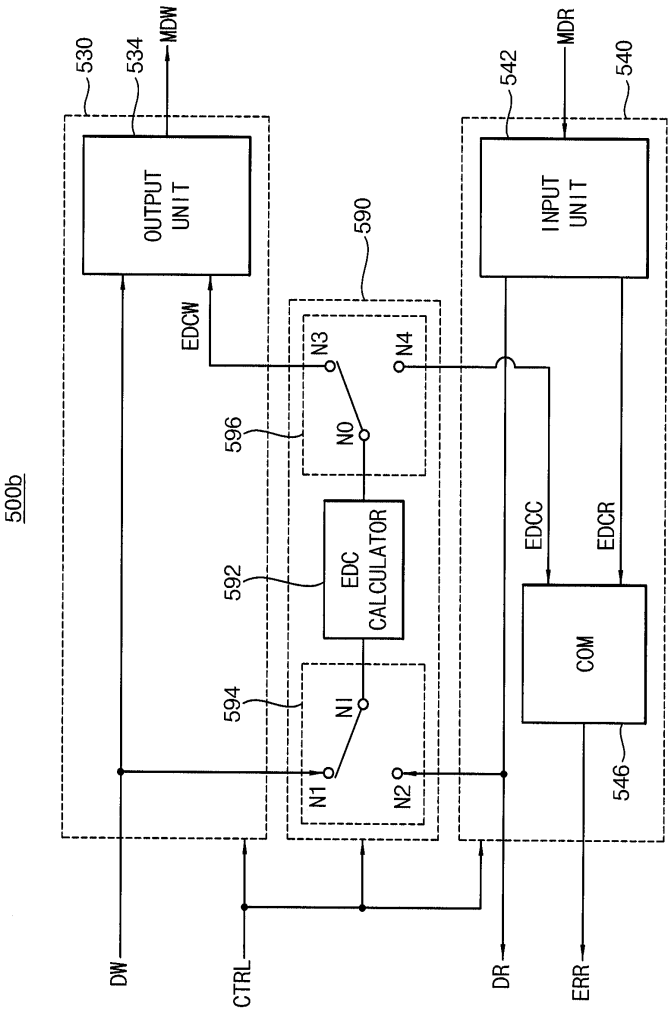
도면2


REGULAR ACCESS
SECURED ACCESS
EN_SMI
(CTRL)
DW
MDW
WRITE
OPERATION
MDR
DR
READ
OPERATION
Tgd
D11
D12
D13
D14
D15
D16
SD11
SD12
SD13
SD14
SD15
SD16
SD21
SD22
SD23
SD24
SD25
SD26
D21
D22
D23
D24
D25
D26

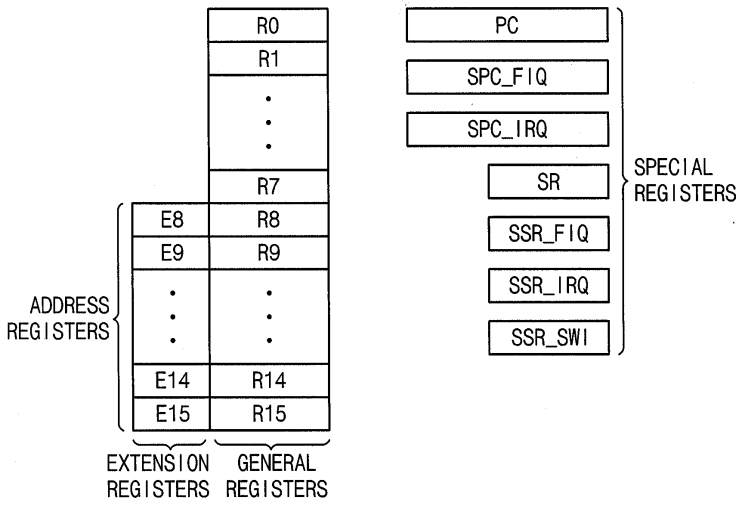
도면3


500a
510
514
MDW
OUTPUT UNIT
EDCW
EDC CALCULATOR
512
DW
CTRL
522
MDR
520
INPUT UNIT
524
EDC CALCULATOR
EDCC
EDCR
COM
526
DR
ERR

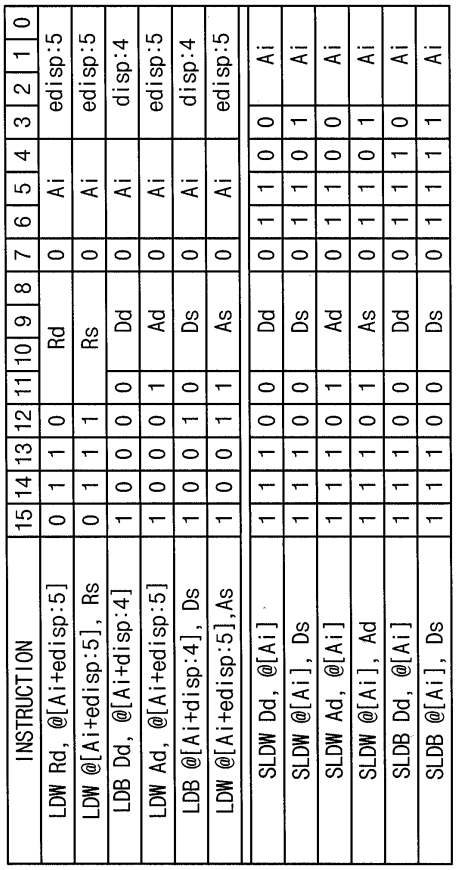
도면4


500b
530
534
MDW
OUTPUT UNIT
542
MDR
540
INPUT UNIT
590
EDCW
N3
N4
596
N0
EDC CALCULATOR
592
EDCC
EDCR
COM
546
594
N1
N1
N2
DW
CTRL
DR
ERR

도면5

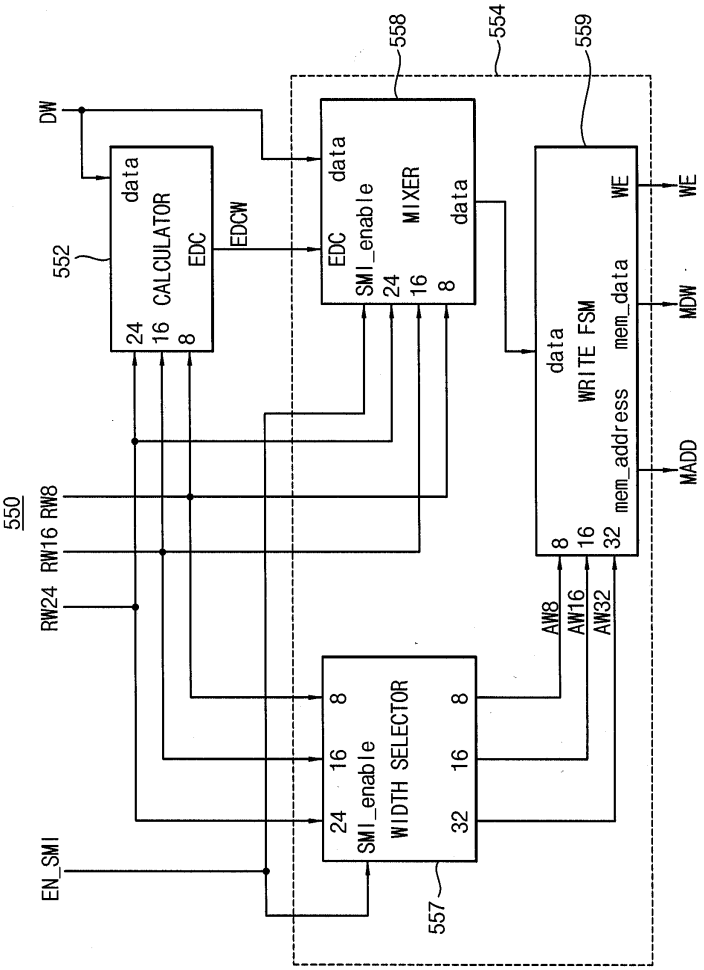

R0
R1
R7
E8
R8
E9
R9
E14
R14
E15
R15
ADDRESS REGISTERS
EXTENSION REGISTERS
GENERAL REGISTERS
PC
SPC_FIQ
SPC_IRQ
SR
SSR_FIQ
SSR_IRQ
SSR_SWI
SPECIAL REGISTERS

도면6

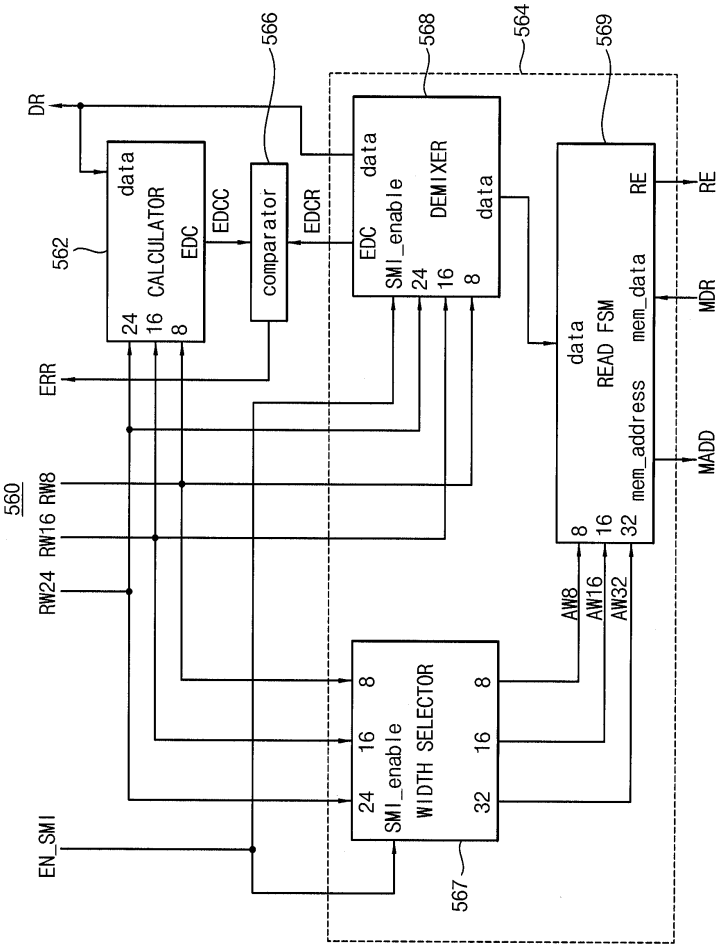


INSTRUCTION	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDW Rd, @[Ai+edisp:5]	0	1	1	0	Rd				0	Ai			edisp:5			
LDW @[Ai+edisp:5], Rs	0	1	1	1	Rs				0	Ai			edisp:5			
LDB Dd, @[Ai+disp:4]	1	0	0	0	0	0	Dd		0	Ai			disp:4			
LDW Ad, @[Ai+edisp:5]	1	0	0	0	1	Ad			0	Ai			edisp:5			
LDB @[Ai+disp:4], Ds	1	0	0	1	0	Ds			0	Ai			disp:4			
LDW @[Ai+edisp:5],As	1	0	0	1	1	As			0	Ai			edisp:5			
SLDW Dd, @[Ai]	1	1	1	0	0	Dd			0	1	1	0	0	Ai		
SLDW @[Ai], Ds	1	1	1	0	0	Ds			0	1	1	0	1	Ai		
SLDW Ad, @[Ai]	1	1	1	0	1	Ad			0	1	1	0	0	Ai		
SLDW @[Ai], Ad	1	1	1	0	1	As			0	1	1	0	1	Ai		
SLDB Dd, @[Ai]	1	1	1	0	0	Dd			0	1	1	1	0	Ai		
SLDB @[Ai], Ds	1	1	1	0	0	Ds			0	1	1	1	1	Ai		

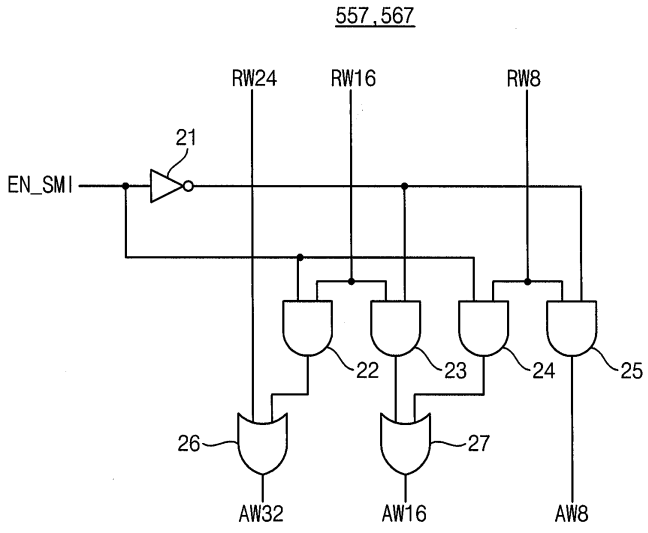
도면7


550
552
554
557
558
559
DW
RW8
RW16
RW24
EN_SMI
EDCW
MDW
MADD
WE
AW8
AW16
AW32
data
CALCULATOR
EDC
24
16
8
MIXER
SMI_enable
WIDTH SELECTOR
WRITE FSM
mem_data
mem_address
32

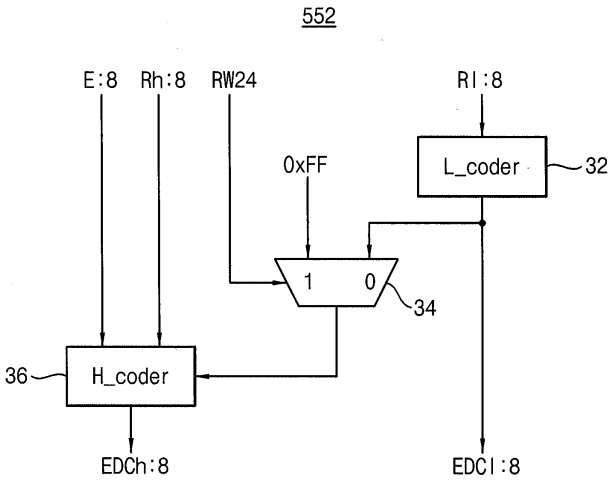
도면8


566
568
564
569
DR
562
data
CALCULATOR
EDC
24
16
8
EDCC
comparator
EDCR
EDC
data
SMI_enable
DEMIXER
24
16
8
data
RE
RE
MDR
MADD
data
READ FSM
mem_data
mem_address
8
16
32
ERR
560
RW8
RW16
RW24
AW8
AW16
AW32
EN_SMI
SMI_enable
WIDTH SELECTOR
24
16
8
8
16
32
567

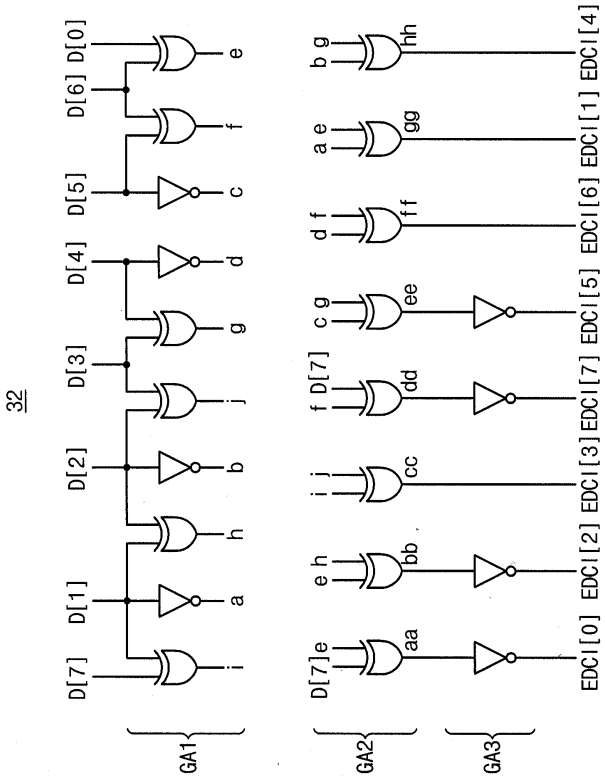
도면9


557,567
RW24
RW16
RW8
21
EN_SMI
22
23
24
25
26
27
AW32
AW16
AW8

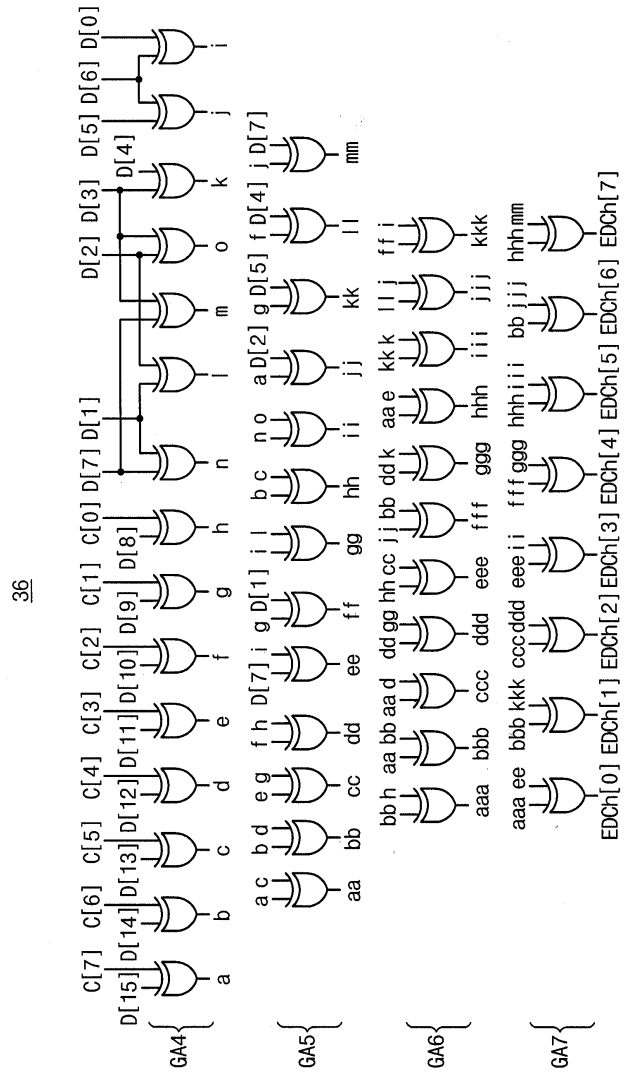
도면10


552
E:8
Rh:8
RW24
RI:8
0xFF
L_coder
32
1
0
34
36
H_coder
EDCh:8
EDCI:8

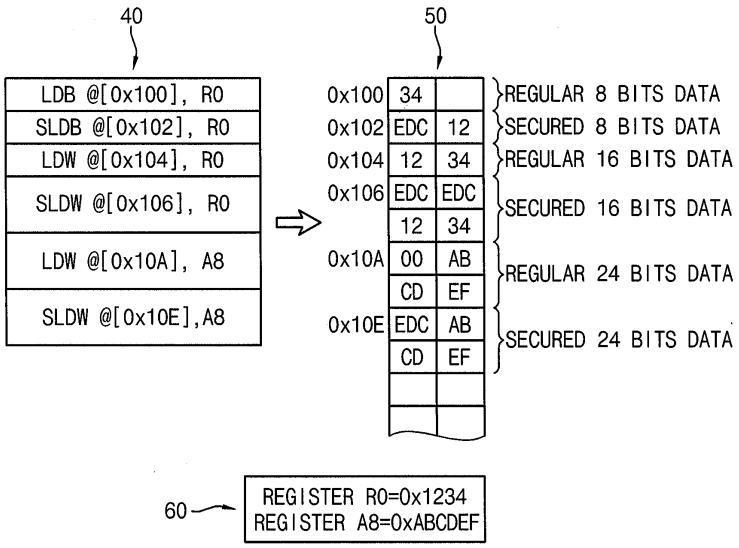
도면11


32
D[0]
D[6]
D[5]
D[4]
D[3]
D[2]
D[1]
D[7]
GA1
GA2
GA3
EDCI[4]
EDCI[1]
EDCI[6]
EDCI[5]
EDCI[7]
EDCI[3]
EDCI[2]
EDCI[0]

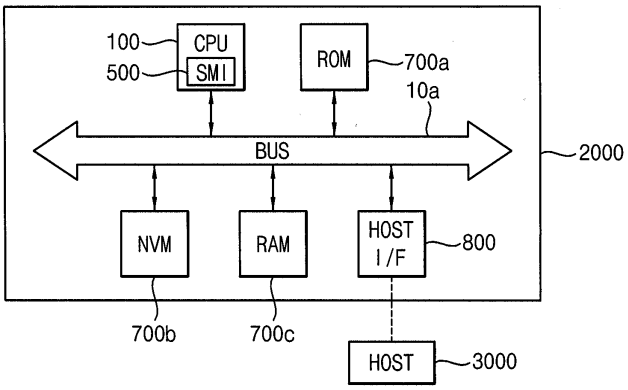
도면12


36
GA4
GA5
GA6
GA7
C[7] C[6] C[5] C[4] C[3] C[2] C[1] C[0] D[7] D[1]
D[15] D[14] D[13] D[12] D[11] D[10] D[9] D[8]
D[2] D[3] D[4] D[5] D[6] D[0]
EDCh[0] EDCh[1] EDCh[2] EDCh[3] EDCh[4] EDCh[5] EDCh[6] EDCh[7]

도면13


40
50
LDB @[0x100], R0
SLDB @[0x102], R0
LDW @[0x104], R0
SLDW @[0x106], R0
LDW @[0x10A], A8
SLDW @[0x10E],A8
0x100 34
0x102 EDC 12
0x104 12 34
0x106 EDC EDC
12 34
0x10A 00 AB
CD EF
0x10E EDC AB
CD EF
REGULAR 8 BITS DATA
SECURED 8 BITS DATA
REGULAR 16 BITS DATA
SECURED 16 BITS DATA
REGULAR 24 BITS DATA
SECURED 24 BITS DATA
60
REGISTER R0=0x1234
REGISTER A8=0xABCDEF

도면14


100
CPU
500
SMI
ROM
700a
10a
BUS
2000
NVM
RAM
HOST I/F
800
700b
700c
HOST
3000