



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0035762
(43) 공개일자 2009년04월13일

(51) Int. Cl.

G06F 12/16 (2006.01) G06F 21/00 (2006.01)

(21) 출원번호 10-2007-0100704

(22) 출원일자 2007년10월08일

심사청구일자 2007년10월08일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가1 고려대학교 내

(72) 발명자

정성우

서울 용산구 이촌동 한가람아파트 217-1002

공준호

서울 광진구 광장동 현대아파트 310-1104

(74) 대리인

조영현, 나승택

전체 청구항 수 : 총 5 항

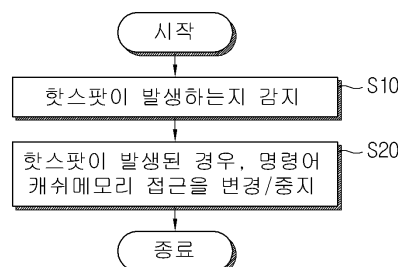
(54) 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법

(57) 요약

본 발명은 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 관한 것으로, 특히 명령어 캐쉬메모리에 핫스팟(hot spot)이 감지된 경우에는 명령어 반입 사이클 사이에 버블(bubble) 사이클을 삽입하거나, 복수 사이클 동안 명령어 반입을 중지함으로써, 명령어 캐쉬메모리의 온도 상승을 방지하는데 적당하도록 한 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 관한 것이다.

본 발명인 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법을 이루는 구성수단은, 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 있어서, 캐쉬 컨트롤러가 명령어 캐쉬 메모리를 구성하는 각각의 인스트럭션 비트셀 어레이에서 악의적인 코드에 의한 핫스팟이 발생하는지를 감지하는 단계, 상기 핫스팟이 감지된 경우에는 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬 메모리의 접근을 변경하거나 중지시키는 단계를 포함하여 이루어진 것을 특징으로 한다.

대표도 - 도2



특허청구의 범위

청구항 1

악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 있어서,

캐쉬 컨트롤러가 명령어 캐쉬 메모리를 구성하는 각각의 인스트럭션 비트셀 어레이에서 악의적인 코드에 의한 핫스팟이 발생하는지를 감지하는 단계;

상기 핫스팟이 감지된 경우에는 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬 메모리의 접근을 변경하거나 중지시키는 단계를 포함하여 이루어진 것을 특징으로 하는 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법.

청구항 2

청구항 1에 있어서, 상기 핫스팟이 발생하는지를 감지하는 단계는,

기준 사이클 동안 상기 각각의 인스트럭션 비트셀 어레이의 접근 횟수를 카운팅하는 과정, 상기 기준 사이클 동안 상기 접근 횟수가 임계값 이상인 인스트럭션 비트셀 어레이가 하나라도 존재하는가 판단하는 과정, 상기 접근 횟수가 임계값 이상인 인스트럭션 비트셀 어레이가 하나라도 존재하는 경우에 핫스팟이 발생하는 것으로 간주하는 과정을 포함하여 이루어진 것을 특징으로 하는 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법.

청구항 3

청구항 1에 있어서, 상기 핫스팟이 발생하는지를 감지하는 단계는,

상기 캐쉬 컨트롤러가 상기 명령어 캐쉬메모리에 접근하는 코드의 패턴이 사전에 저장되어 있는 악의적인 코드의 패턴과 동일한지 판단하는 과정, 상기 판단결과, 동일한 경우로 판단한 경우에는 악의적인 코드가 접근하여 핫스팟이 발생하는 것으로 간주하는 과정을 포함하여 이루어진 것을 특징으로 하는 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법.

청구항 4

청구항 2 또는 청구항 3에 있어서, 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬 메모리의 접근을 변경하는 단계는,

상기 기준 사이클 동안 상기 명령어 캐쉬메모리를 접근할 때마다 상기 캐쉬 컨트롤러가 버블(bubble) 사이클을 삽입하여 상기 명령어 캐쉬메모리의 접근을 제어하는 것을 특징으로 하는 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법.

청구항 5

청구항 2 또는 청구항 3에 있어서, 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬 메모리의 접근을 중지하는 단계는,

상기 기준 사이클 동안 상기 명령어 캐쉬메모리의 접근을 중지하도록 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬메모리의 접근을 제어하는 것을 특징으로 하는 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법.

명세서

발명의 상세한 설명

기술 분야

<1> 본 발명은 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 관한 것으로, 특히 명령어 캐쉬메모리에 핫스팟(hot spot)이 감지된 경우에는 명령어 반입 사이클 사이에 버블(bubble) 사이클을 삽입하거나, 복수 사이클 동안 명령어 반입을 중지함으로써, 명령어 캐쉬메모리의 온도 상승을 방지하는데 적당하도록 한 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 관한 것이다.

배경 기술

<2> 캐쉬메모리(cache memory)는 컴퓨터 속에 장착해 속도를 빠르게 하는 임시메모리이다. 컴퓨터의 속도를 이야기

할 때에는 일반적으로 CPU(중앙처리장치)의 속도를 이야기한다. 이는 CPU의 속도에 의해 컴퓨터의 속도가 결정될 정도로 CPU에 비해 상대적으로 주변기기의 속도가 매우 느리다는 것을 나타낸다.

- <3> 이러한 문제점을 해결하기 위하여 주변기기의 속도를 빠르게 하는 대표적인 방법이 캐시메모리이다. 램캐시(외부캐시), 내부캐시, 디스크캐시 등 여러 종류의 캐시가 있다.
- <4> 하드디스크의 속도는 램보다 매우 느리다. 매번 프로그램을 실행시킬 때마다 디스크를 읽어야 하므로 속도가 느릴 수밖에 없다. 따라서 램과 디스크 사이에 일정량의 임시메모리를 만들고 처음 프로그램을 실행할 때 램으로 들어오는 내용을 그 임시메모리에도 보관한다. 그런 다음 프로그램을 실행시키면 하드디스크가 아닌 임시메모리에서 읽어오게 되기 때문에 읽어오는 시간이 매우 빨라지게 된다. 이 임시메모리를 캐시메모리라고 한다.
- <5> 요즘에는 PC(퍼스널컴퓨터)에서도 사용되며, 수 10메가바이트의 주기억장치에 대해 수 10킬로바이트의 캐시메모리를 갖추고 있다. 캐시메모리를 내부에 장착한 마이크로 프로세서도 나왔다. 펜티엄컴퓨터는 CPU 내부의 캐시를 두 개로 늘렸는데, 각각은 명령처리를 위한 캐시와 데이터처리를 위한 캐시이다. 이로 인해 CPU의 속도가 훨씬 빨라졌다.
- <6> 더욱 상세하게 캐시메모리는 처리속도가 엄청 빠른 CPU와 처리속도가 상대적으로 엄청 느린 HDD또는 주 메모리 등간의 속도완충장치의 역할을 하는 SRAM으로 구성된 메모리를 말한다.
- <7> 주 메모리에서 일단 캐시메모리로 정보를 전송받고 캐시메모리가 CPU와 정보를 주고받는 일이 동시에 이루어지면서 데이터 대기 시간을 최소로 줄여서 전체 시스템 속도를 향상시킨다. 따라서 CPU를 높이는 것보다 캐시메모리를 좀더 용량이 큰 것으로 업그레이드 하는게 속도 향상에 더 도움이 된다.
- <8> 캐시 메모리에는 1차(L1), 2차(L2)캐시로 나뉘는데, L1(1차)캐시는 CPU내부에 내장되어 있는 메모리로 CPU내부 클럭 주파수에 의해 빠르게 동작하고 L2(2차)캐시는 메인보드에서 지원되는 메모리로 외부 클럭 주파수에 의해 느리게 동작했었다. 하지만 펜티엄 프로부터는 L2캐시도 CPU에 내장이되어 빠른 속도로 동작하게 되었다. 펜티엄은 16[KB], 펜티엄 MMX 32[KB]의 L1 캐시를 갖고 있고, 펜티엄 프로는 256 ~ 512[KB]와 펜티엄 2 는 512[KB]의 L2 캐시를 갖고 있다. 셀러론은 펜티엄 2에서 바로 이 L2 캐시만을 없앤 것이다. 그리고 셀러론 A 는 이 L2캐시 메모리의 용량을 대폭 줄여 출시한 CPU를 말한다.
- <9> 그런데, 상기와 같은 캐시메모리는 너무 많은 접근이 발생하거나 컴퓨터 상에서 악의적인 코드가 실행되는 경우 온도가 높아져 타버리는 일이 발생할 수 있다. 따라서, 누군가가 일부러 악의적인 코드를 실행시킨다면 충분히 캐시메모리의 온도를 높일 수 있고, 결과적으로 캐시메모리가 타버릴 수 있게 된다.
- <10> 마이크로프로세서 안에 일반적으로 널리 알려져 있는 뜨거운 유닛이 악의적인 코드에 의하여 공격당할 경우에는, 마이크로프로세서 안에 설치되어 있는 온도 센서가 상기 뜨거운 유닛이 뜨거워지는 것을 감지하여 DTM(dynamic thermal management)을 실행시킨다. 그러면, 마이크로프로세서의 전체적인 온도가 내려가면서 악의적인 코드는 효과를 발휘하지 못하게 된다.
- <11> 그러나, 명령어 캐시 메모리와 같은 일반적으로 차가운 유닛에 대하여 악의적인 코드가 공격하는 경우에는, 상기 뜨거운 유닛에 설치되어 있는 온도 센서에 의하여 탐지되지 않기 때문에 물리적인 타격을 받을 수 있고, 심지어는 상기 명령어 캐시 메모리와 같은 일반적으로 차가운 유닛이 타버릴 수 있다.
- <12> 따라서, 상기 명령어 캐시 메모리와 같은 일반적으로 차가운 유닛이 악의적인 코드에 의하여 물리적인 손상을 받거나 타버리는 일이 발생하지 않도록 대처 방안을 제시할 필요가 있다. 그러나 현재까지 악의적인 코드에 의하여 캐시메모리와 같은 차가운 유닛이 공격당하는 것을 방지하는 대책을 마련하지 못하고 있는 실정이다.

발명의 내용

해결 하고자하는 과제

- <13> 본 발명은 상기와 같은 종래 기술의 문제점을 해결하기 위하여 창안된 것으로, 명령어 캐시메모리에 핫스팟(hot spot)이 감지된 경우에는 명령어 반입 사이클 사이에 버블(bubble) 사이클을 삽입하거나, 복수 사이클 동안 명령어 반입을 중지함으로써, 명령어 캐시메모리의 온도 상승을 방지하는데 적당하도록 한 악의적인 코드에 의한 명령어 캐시메모리의 손상 방지 방법을 제공하는 것을 그 목적으로 한다.

과제 해결수단

- <14> 상기와 같은 과제를 해결하기 위하여 제안된 본 발명인 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법을 이루는 구성수단은, 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 있어서, 캐쉬 컨트롤러가 명령어 캐쉬 메모리를 구성하는 각각의 인스트럭션 비트셀 어레이에서 악의적인 코드에 의한 핫스팟이 발생하는지를 감지하는 단계, 상기 핫스팟이 감지된 경우에는 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬 메모리의 접근을 변경하거나 중지시키는 단계를 포함하여 이루어진 것을 특징으로 한다.
- <15> 또한, 상기 핫스팟이 발생하는지를 감지하는 단계는, 기준 사이클 동안 상기 각각의 인스트럭션 비트셀 어레이의 접근 횟수를 카운팅하는 과정, 상기 기준 사이클 동안 상기 접근 횟수가 임계값 이상인 인스트럭션 비트셀 어레이가 하나라도 존재하는가 판단하는 과정, 상기 접근 횟수가 임계값 이상인 인스트럭션 비트셀 어레이가 하나라도 존재하는 경우에 핫스팟이 발생하는 것으로 간주하는 과정을 포함하여 이루어진 것을 특징으로 한다.
- <16> 또한, 상기 핫스팟이 발생하는지를 감지하는 단계는, 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬메모리에 접근하는 코드의 패턴이 사전에 저장되어 있는 악의적인 코드의 패턴과 동일한지 판단하는 과정, 상기 판단결과, 동일한 경우로 판단한 경우에는 악의적인 코드가 접근하여 핫스팟이 발생하는 것으로 간주하는 과정을 포함하여 이루어진 것을 특징으로 한다.
- <17> 또한, 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬 메모리의 접근을 변경하는 단계는, 상기 기준 사이클 동안 상기 명령어 캐쉬메모리를 접근할 때마다 상기 캐쉬 컨트롤러가 버블(bubble) 사이클을 삽입하여 상기 명령어 캐쉬메모리의 접근을 제어하는 것을 특징으로 한다.
- <18> 또한, 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬 메모리의 접근을 중지하는 단계는, 상기 기준 사이클 동안 상기 명령어 캐쉬메모리의 접근을 중지하도록 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬메모리의 접근을 제어하는 것을 특징으로 한다.

효 과

- <19> 상기와 같은 과제 및 해결 수단을 가지는 본 발명인 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 의하면, 명령어 캐쉬메모리에 핫스팟(hot spot)이 감지된 경우에는 명령어 반입 사이클 사이에 버블(bubble) 사이클을 삽입하거나, 복수 사이클 동안 명령어 반입을 중지할 수 있기 때문에, 명령어 캐쉬메모리의 온도 상승을 방지할 수 있고, 결과적으로 상기 명령어 캐쉬메모리가 손상되는 것을 방지할 수 있는 효과가 있다.

발명의 실시를 위한 구체적인 내용

- <20> 이하, 도면을 참조하여 상기와 같은 구성수단 및 효과를 가지는 본 발명인 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 관한 바람직한 실시예를 상세하게 설명한다.
- <21> 도 1은 본 발명에 따른 명령어 캐쉬메모리 구조의 예시도이다.
- <22> 도 1에 도시된 바와 같이, 명령어 캐쉬메모리 구조는 복수개의 인스트럭션 비트셀 어레이(instruction bit-cell array)로 구성된다. 상기 각각의 인스트럭션 비트셀 어레이(instruction bit-cell array)들은 다른 유닛들에 의하여 접근되어진다. 도 1에서는 8개의 인스트럭션 비트셀 어레이(instruction bit-cell array)가 상기 명령어 캐쉬메모리를 구성하는 것으로 예시하였지만, 이는 당연히 변경될 수 있다.
- <23> 상기 인스트럭션 비트셀 어레이(instruction bit-cell array)들이 정상적인 코드에 의하여 접근되어지는 경우에는 문제가 없지만, 악의적인 코드에 의하여 접근되어지는 경우에는 상기 인스트럭션 비트셀 어레이(instruction bit-cell array)를 구성하는 명령어 캐쉬메모리의 온도가 높아지고, 결국 타버리는 상태까지 진전될 수 있다.
- <24> 따라서, 상기 명령어 캐쉬메모리는 악의적인 코드에 의하여 공격을 받고 있는지를 확인할 필요가 있고, 또한 공격을 받는 경우에 공격에 의한 손상을 방지할 수 있도록 하여야 한다.
- <25> 도 2는 본 발명에 따라, 악의적인 코드에 의한 명령어 캐쉬메모리의 손상을 방지하기 위한 방법의 절차도이다.
- <26> 도 2에 도시된 바와 같이, 악의적인 코드에 의하여 명령어 캐쉬메모리가 손상되는 것을 방지하기 위하여, 먼저 캐쉬 컨트롤러가 명령어 캐쉬메모리에 핫스팟(hot spot)이 발생하는 지를 감지한다(s10).
- <27> 상기 명령어 캐쉬메모리는 도 1에 도시된 바와 같이, 복수개의 인스트럭션 비트셀 어레이들로 구성된다. 따라서, 상기 캐쉬 컨트롤러는 상기 명령어 캐쉬 메모리를 구성하는 각각의 인스트럭션 비트셀 어레이들에서 악의적인 코드에 의해 핫스팟이 발생하는지를 감지한다.

- <28> 상기 캐쉬 컨트롤러는 다양한 방법에 의하여 상기 각각의 인스트럭션 비트셀 어레이들에서 핫스팟이 발생하는지를 감지하고, 다양한 방법에 의하여 상기 명령어 캐쉬메모리가 상기 핫스팟에 의하여 손상되는 것을 방지한다. 이에 대해서는 후술하겠다.
- <29> 상기 캐쉬 컨트롤러에 의하여 핫스팟이 감지된 경우에는, 상기 캐쉬 컨트롤러는 다른 유닛들에 의하여 상기 명령어 캐쉬메모리의 접근을 변경하거나 또는 명령어 캐쉬메모리의 접근을 중지할 수 있도록 제어한다(s20). 이와 같이, 핫스팟이 감지된 경우에 명령어 캐쉬메모리의 접근을 변경하거나 중지하는 것은 상기 명령어 캐쉬메모리의 온도가 높아지는 것을 방지하여 타버리는 것을 막기 위해서이다.
- <30> 도 3은 본 발명의 제1 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법의 절차도이다.
- <31> 본 발명의 제1 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법에 대하여 도 3을 참조하여 설명하면 다음과 같다.
- <32> 먼저, 상기 캐쉬 컨트롤러가 명령어 캐쉬메모리를 구성하는 각각의 인스트럭션 비트셀 어레이들에서 악의적인 코드에 의하여 핫스팟이 발생하는지를 감지한다. 이에 대하여 구체적으로 설명하면, 먼저 상기 캐쉬 컨트롤러가 정해진 기준 사이클 동안 다른 유닛에 의하여 상기 각각의 인스트럭션 비트셀 어레이들의 접근 횟수를 카운팅을 수행한다(s11).
- <33> 상기 각각의 인스트럭션 비트셀 어레이들의 접근 횟수를 카운팅하는 방법은 상기 캐쉬 컨트롤러에 구비되는 카운터에 의하여 이루어진다. 상기 카운터는 상기 각각의 인스트럭션 비트셀 어레이들에 일대일 대응되어, 다른 유닛에 의하여 상기 대응되는 각각의 인스트럭션 비트셀 어레이들의 접근 횟수를 카운팅한다.
- <34> 그런 다음, 상기 캐쉬 컨트롤러는 상기 기준 사이클 동안 상기 접근 횟수가 사전에 세팅된 임계값 이상인 인스트럭션 비트셀 어레이가 하나라도 존재하는지를 판단한다(s13). 예를 들면, 상기 기준 사이클을 100000 사이클 이라고 하고, 상기 임계값을 45000 이라고 할 때, 상기 캐쉬 컨트롤러는 상기 기준 사이클인 100000 사이클 동안 상기 임계값인 45000 이상으로 접근되는 인스트럭션 비트셀 어레이가 하나라도 존재하는지를 판단한다.
- <35> 상기 기준 사이클과 임계값은 사전에 세팅될 수 있는 값으로서, 칩 제조 전에 시뮬레이션을 통해서 결정될 수 있는 값이다. 즉, 시뮬레이션을 통하여 100000 사이클 동안 상기 인스트럭션 비트셀 어레이 중 적어도 하나에 45000 번 접근이 있는 경우에는 악의적인 코드에 의하여 접근이 되고 있다는 것으로 간주할 수 있는 것이다.
- <36> 따라서, 상기 기준 사이클(예를 들어 100000) 동안 상기 접근 횟수가 임계값(예를 들어 45000) 이상인 인스트럭션 비트셀 어레이가 하나라도 존재하는 경우에는 악의적인 코드에 의하여 접근이 되고 있다는 것으로 간주하여 상기 명령어 캐쉬메모리에 핫스팟이 발생하는 것으로 간주한다(s15). 즉, 상기 임계값은 시뮬레이션을 통하여 결정될 수 있는 값으로서, 상기 인스트럭션 비트셀 어레이에 악의적인 코드가 접근되고, 결과적으로 핫스팟이 발생하는 것으로 간주할 수 있는 분계값이라 할 수 있다.
- <37> 상기와 같이, 상기 기준 사이클 동안 상기 접근 횟수가 임계값 이상인 인스트럭션 비트셀 어레이가 하나라도 존재하는 것으로 판단되어 핫스팟이 발생하는 것으로 간주된 경우에는, 상기 캐쉬 컨트롤러는 다른 유닛에 의하여 상기 명령어 캐쉬메모리의 접근을 변경한다. 이와 같은 상기 명령어 캐쉬메모리의 접근 변경은 온도가 상승하는 것을 방지하고 결과적으로 타버리는 것을 막기 위함이다.
- <38> 상기 캐쉬 컨트롤러가 다른 유닛에 의하여 상기 명령어 캐쉬메모리의 접근을 변경하는 방법은 다양하다. 본 발명의 제1 실시예에서는 도 3에 도시된 바와 같이, 상기 기준 사이클 동안 다른 유닛이 상기 명령어 캐쉬메모리를 접근할 때마다, 상기 캐쉬 컨트롤러가 버블(bubble) 사이클을 삽입함으로써, 상기 명령어 캐쉬메모리의 접근이 변경될 수 있도록 제어한다(s21).
- <39> 상기와 같이, 명령어 캐쉬메모리의 접근이 있을 때마다, 버블 사이클을 삽입함으로써, 명령어 반입을 느리게 한다. 그러면, 결과적으로 상기 악의적인 코드의 접근은 감소되고, 상기 명령어 캐쉬메모리의 온도도 감소된다.
- <40> 도 4는 본 발명의 제2 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법의 절차도이다.
- <41> 도 4에 도시된 바와 같이, 상기 캐쉬 컨트롤러가 명령어 캐쉬 메모리를 구성하는 각각의 인스트럭션 비트셀 어레이들에서 악의적인 코드에 의한 핫스팟이 발생하는지를 감지하는 단계(s11, s13, s15)는 도 3에 도시된 본 발명의 제1 실시예와 동일하다. 따라서, 이에 대해서는 설명을 생략한다.
- <42> 다만, 상기 인스트럭션 비트셀 어레이들 중에 적어도 하나에 핫스팟이 발생한 것으로 간주된 경우에는 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬메모리의 접근을 중지한다(s21). 즉, 핫스팟이 발생된 것으로 간주된 경우에는

기준 사이클 동안 상기 명령어 캐쉬메모리의 온도 상승을 방지하기 위하여 다른 유닛에 의하여 상기 명령어 캐쉬메모리의 접근을 상기 캐쉬 컨트롤러가 중지시킨다.

<43> 따라서, 상기 캐쉬 컨트롤러는 상기 핫스팟이 발생된 것으로 간주된 경우에는, 상기 기준 사이클 동안 상기 명령어 캐쉬메모리의 접근을 중지하도록, 다른 유닛에 의한 상기 명령어 캐쉬메모리의 접근을 제어한다.

<44> 도 5는 본 발명의 제3 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법의 절차도이다.

<45> 상기 제3 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법은 캐쉬 컨트롤러가 명령어 캐쉬 메모리를 구성하는 각각의 인스트럭션 비트셀 어레이에서 악의적인 코드에 의한 핫스팟이 발생하는지 감지하는 방법이 상술한 제1 및 제2 실시예와 다르다.

<46> 도 5에 도시된 바와 같이, 제3 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법은, 일단 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬메모리에 접근하는 코드의 패턴이 사전에 저장되어 있는 악의적인 코드의 패턴과 동일한지 비교판단한다(s11, s13).

<47> 상기 명령어 캐쉬메모리를 집중적으로 접근하는 악의적인 코드는 일정한 특이 패턴을 가지고 있으므로, 사전에 상기 악의적인 코드의 이미지를 저장해 놓고, 상기 명령어 캐쉬메모리에 접근하는 코드와 동일 여부를 판단할 수 있다. 상기 악의적인 코드는 다양하게 변경될 수 있으므로, 사전에 저장되는 악의적인 코드는 수시로 업데이트된다.

<48> 상기 명령어 캐쉬메모리에 접근하는 코드와 사전에 저장된 악의적인 코드의 패턴이 동일한지를 판단한 결과, 동일한 경우로 판단한 경우에는 상기 명령어 캐쉬메모리에 접근하는 코드는 악의적인 코드로 간주할 수 있기 때문에, 상기 명령어 캐쉬메모리에 핫스팟이 발생할 것으로 간주한다(s15). 이 경우는 상술한 제1 및 제2 실시예와 달리, 사전에 핫스팟이 발생할 것을 감지할 수 있는 장점이 있다.

<49> 상기와 같이 명령어 캐쉬메모리에 핫스팟이 발생할 것으로 간주된 경우에는 상기 명령어 캐쉬메모리가 고온화되어 타버릴 수 있기 때문에, 사전에 상기 명령어 캐쉬메모리의 온도가 상승하는 것을 방지해야 한다. 이를 위하여, 상기 캐쉬 컨트롤러가 다른 유닛에 의한 상기 명령어 캐쉬메모리의 접근을 변경하도록 제어한다.

<50> 상기 캐쉬 컨트롤러가 다른 유닛에 의하여 상기 명령어 캐쉬메모리의 접근을 변경하는 방법은 다양하다. 본 발명의 제3 실시예에서는 도 5에 도시된 바와 같이, 상기 기준 사이클 동안 다른 유닛이 상기 명령어 캐쉬메모리를 접근할 때마다, 상기 캐쉬 컨트롤러가 버블(bubble) 사이클을 삽입함으로써, 상기 명령어 캐쉬메모리의 접근이 변경될 수 있도록 제어한다(s21).

<51> 상기와 같이, 명령어 캐쉬메모리의 접근이 있을 때마다, 버블 사이클을 삽입함으로써, 명령어 반입을 느리게 한다. 그러면, 결과적으로 상기 악의적인 코드의 접근은 감소되고, 상기 명령어 캐쉬메모리의 온도도 감소된다.

<52> 도 6은 본 발명의 제4 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬메모리의 손상 방지 방법의 절차도이다.

<53> 도 6에 도시된 바와 같이, 상기 캐쉬 컨트롤러가 명령어 캐쉬 메모리를 구성하는 각각의 인스트럭션 비트셀 어레이들에서 악의적인 코드에 의한 핫스팟이 발생하는지를 감지하는 단계(s11, s13, s15)는 도 5에 도시된 본 발명의 제3 실시예와 동일하다. 따라서, 이에 대해서는 설명을 생략한다.

<54> 다만, 상기 인스트럭션 비트셀 어레이들 중에 적어도 하나에 핫스팟이 발생한 것으로 간주된 경우에는 상기 캐쉬 컨트롤러가 상기 명령어 캐쉬메모리의 접근을 중지한다(s21). 즉, 핫스팟이 발생된 것으로 간주된 경우에는 기준 사이클 동안 상기 명령어 캐쉬메모리의 온도 상승을 방지하기 위하여 다른 유닛에 의하여 상기 명령어 캐쉬메모리의 접근을 상기 캐쉬 컨트롤러가 중지시킨다.

<55> 따라서, 상기 캐쉬 컨트롤러는 상기 핫스팟이 발생된 것으로 간주된 경우에는, 상기 기준 사이클 동안 상기 명령어 캐쉬메모리의 접근을 중지하도록, 다른 유닛에 의한 상기 명령어 캐쉬메모리의 접근을 제어한다.

도면의 간단한 설명

<56> 도 1은 본 발명에 따른 명령어 캐쉬메모리 구조의 예시도이다.

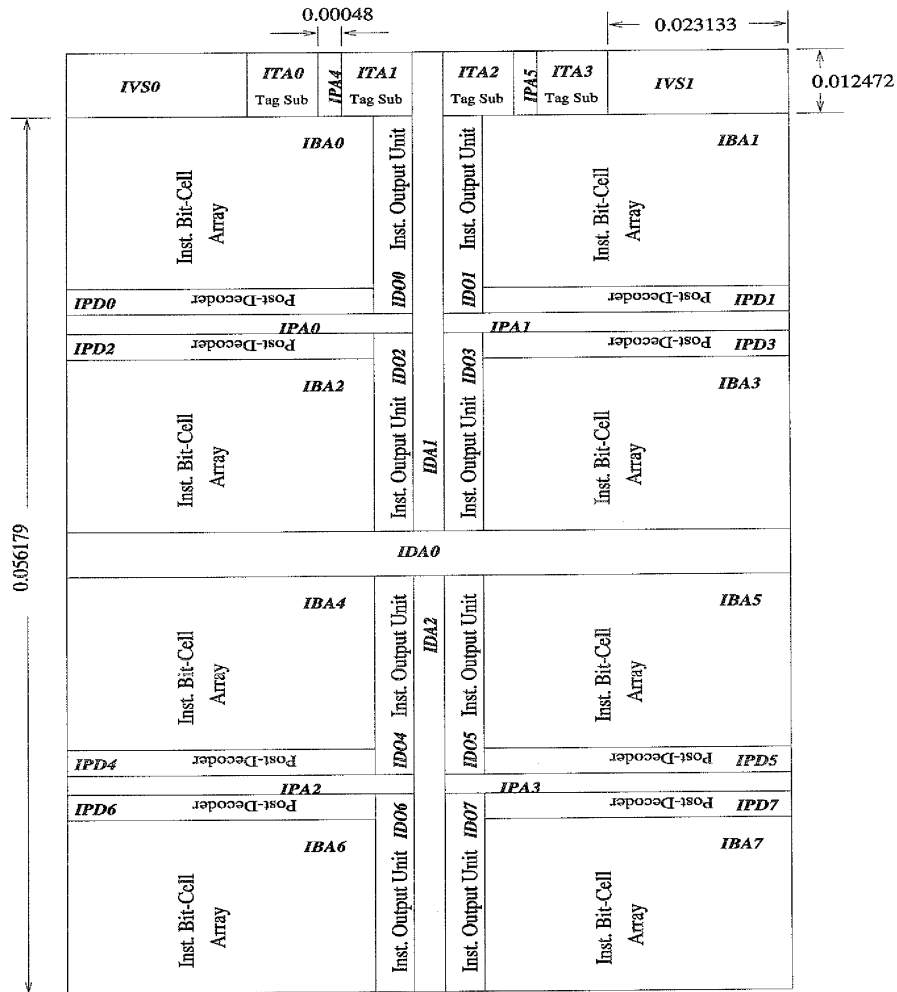
<57> 도 2는 본 발명에 따른 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법의 절차도이다.

<58> 도 3은 본 발명의 제1 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법의 절차도이다.

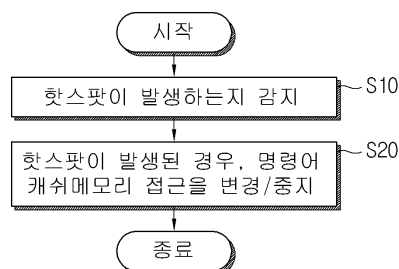
- <59> 도 4는 본 발명의 제2 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법의 절차도이다.
- <60> 도 5는 본 발명의 제3 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법의 절차도이다.
- <61> 도 6은 본 발명의 제4 실시예에 따른 악의적인 코드에 의한 명령어 캐쉬 메모리의 손상 방지 방법의 절차도이다.

도면

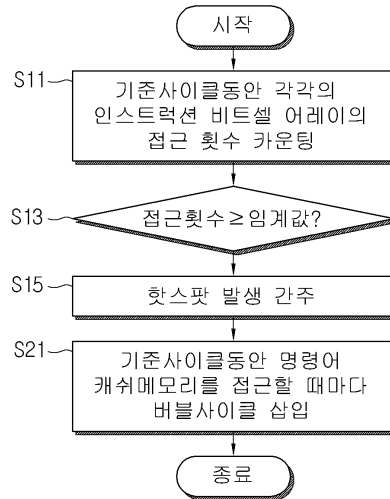
도면1



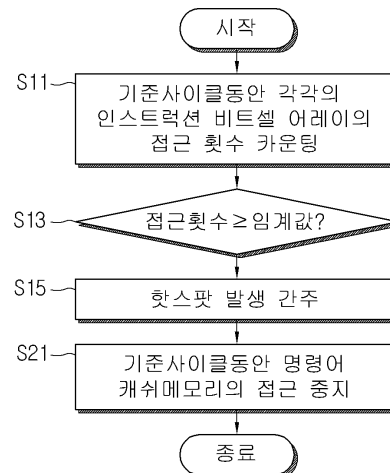
도면2



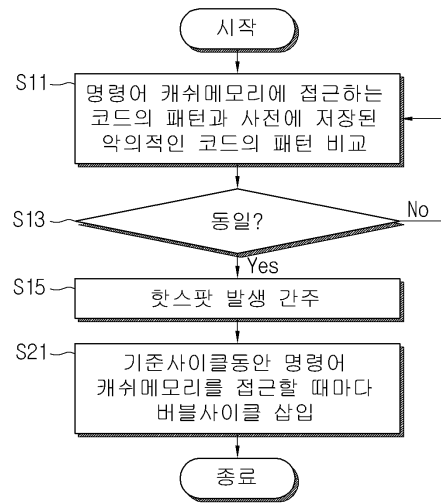
도면3



도면4



도면5



도면6

