



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년07월02일
(11) 등록번호 10-1959002
(24) 등록일자 2019년03월11일

(51) 국제특허분류(Int. Cl.)
G06F 9/44 (2018.01)

(21) 출원번호 10-2014-7035472

(22) 출원일자(국제) 2013년05월29일
심사청구일자 2018년05월29일

(85) 번역문제출일자 2014년12월17일

(65) 공개번호 10-2015-0016331

(43) 공개일자 2015년02월11일

(86) 국제출원번호 PCT/US2013/043057

(87) 국제공개번호 WO 2013/181220
국제공개일자 2013년12월05일

(30) 우선권주장
1637/DEL/2012 2012년05월30일 인도(IN)
18/547-761 2013년07월10일 미국(US)

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485

(72) 발명자
루엑 앤드류 월리엄
미국 텍사스 78749 오스틴 오밸리 드라이브 9000
베르누초 크리쉬나 사이
미국 텍사스 78749 오스틴 도 매도우 드라이브
8015
(뒷면에 계속)

(74) 대리인
박자인

(56) 선행기술조사문항

US20110016300 A1
US20040117318 A1
WO2001033322 A1
CN100415085 A

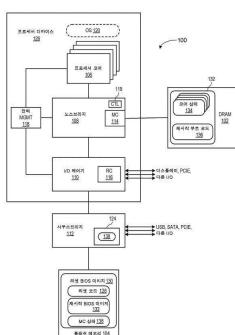
심사관 : 최정권

(54) 발명의 명칭 저전력 상태로부터 재시작시 회발성 메모리로부터 처리 시스템의 재초기화

(57) 요약

부트 구성 정보(130)는 저전력 상태 동안 처리 시스템(100)의 휘발성 메모리(102)에 저장된다. 저전력 상태로부터 재시작할 때, 프로세서 디바이스(126)는 비휘발성 메모리(104, 124)로부터 메모리 제어기(114)를 위한 구성 정보(138)에 액세스하고, 휘발성 메모리(102)에 액세스하도록 구성 정보(138)를 이용하여 메모리 제어기(114)를 복구한다. 프로세서 디바이스(126)는 저전력 상태 동안 휘발성 메모리(102)에 의해 유지되고 구성된 메모리 제어기(114)를 통해 액세스된 코어 상태 정보(134)를 이용하여 하나 이상의 프로세서 코어들(106)의 초기 정황들을 구성하고, 하나 이상의 프로세서 코어들(106)은, 비휘발성 메모리(104)로부터 부트 코드(132)에 액세스하기 보다는, 저전력 상태 동안 휘발성 메모리(102)에 의해 유지되고 구성된 메모리 제어기(114)를 통해 액세스된 재시작 부트 코드(136)를 실행함으로써 부트 프로세스를 완료한다.

대표 도



(72) 발명자

브랜노버 알렉산더 제이.

미국 매사추세츠 02467 체스트넛 힐 뉴튼 스트리트
783

킷친 파울 에드워드

미국 텍사스 78735 오스틴 트레일 크레스트 써클
4708

페레즈 로날드

미국 텍사스 78733 오스틴 리비아나 드라이브 2117

아로라 소누

인도 말레스와람 560003 제13 크로스 제4 메인
디-6/아디트야 네스트

명세서

청구범위

청구항 1

처리 시스템(processing system)을 동작시키는 방법으로서, 상기 방법은,

프로세서 디바이스(processor device)(126)의 전력-온 리셋(power-on reset)에 응답하여 비휘발성 메모리(non-volatile memory)(104)에 저장된 제 1 부트 구성 정보(boot configuration information)(130)를 이용하여 상기 프로세서 디바이스(126)를 초기화(initializing)하는 단계와, 여기서 상기 프로세서 디바이스(126)를 초기화하는 것은 상기 제 1 부트 구성 정보(130)의 부트 코드(boot code)를 실행시켜 제 2 부트 구성 정보(132)의 복사본(copy)을 상기 비휘발성 메모리(104)로부터 휘발성 메모리(102)로 전달하는 것을 포함하고;

상기 프로세서 디바이스(126)를 초기화하는 것에 후속하여 상기 프로세서 디바이스(126)를 저전력 상태(low-power state)에 있게 하는 단계; 그리고

상기 프로세서 디바이스(126)에 공급되는 전력이 감소되는 상기 저전력 상태로부터 재시작(resume)하도록 하는 표시에 응답하여, 상기 저전력 상태 동안 상기 휘발성 메모리(102)에 저장된 상기 제 2 부트 구성 정보(132)의 상기 복사본을 이용하여 상기 프로세서 디바이스(126)를 다시 초기화하는 단계를 포함하고, 여기서 상기 제 2 부트 구성 정보(132)의 상기 복사본을 이용하여 상기 프로세서 디바이스(126)를 다시 초기화하는 것은,

상기 제 2 부트 구성 정보(132)의 코어 상태 정보(core state information)에 기초하여 상기 프로세서 디바이스(126)의 프로세서 코어(processor core)(106)의 정황(context)을 구성하는 것과, 여기서 상기 프로세서 코어(106)의 상기 정황을 구성하는 것은 상기 프로세서 코어(106)의 지시 포인터(instruction pointer)를 상기 제 2 부트 구성 정보(132)의 제 2 부트 코드에 대한 리셋 벡터(reset vector)로 설정하는 것을 포함하고, 그리고

상기 프로세서 코어(106)를 이용하여 상기 제 2 부트 코드를 실행하는 것을 포함하는 것을 특징으로 하는 처리 시스템을 동작시키는 방법.

청구항 2

제1항에 있어서,

상기 프로세서 디바이스(126)를 다시 초기화하는 것은,

상기 비휘발성 메모리(104)에 저장된 메모리 제어기 구성 정보(memory controller configuration information)(138)를 이용하여 상기 프로세서 디바이스(126)의 메모리 제어기(114)를 구성하는 것과; 그리고

상기 구성된 메모리 제어기(114)를 통해 상기 휘발성 메모리(102)로부터 상기 제 2 부트 구성 정보(132)에 액세스하는 것을 포함하는 것을 특징으로 하는 처리 시스템을 동작시키는 방법.

청구항 3

제2항에 있어서,

상기 메모리 제어기(114)를 구성하는 것은, 타이밍 정보(timing information)와 어드레스 매핑 정보(address mapping information) 중 적어도 하나를 구성하는 것을 포함하는 것을 특징으로 하는 처리 시스템을 동작시키는 방법.

청구항 4

제1항에 있어서,

상기 프로세서 디바이스(126)를 다시 초기화하는 것은,

상기 제 2 부트 구성 정보(132)의 코어 상태 정보(134)에 기초하여 상기 프로세서 디바이스(126)의 프로세서 코어(106)의 정황을 구성하는 것과;

상기 프로세서 코어(106)를 이용하여 상기 제 2 부트 구성 정보(132)의 제 1 부트 코드(136)를 실행하는 것을 포함하는 것을 특징으로 하는 처리 시스템을 동작시키는 방법.

청구항 5

제4항에 있어서,

상기 프로세서 코어(106)의 상기 정황을 구성하는 것은 또한,

상기 코어 상태 정보(134)를 이용하여 상기 프로세서 코어(106)의 하나 이상의 레지스터 값(register value)들을 복구(restoring)하는 것;

상기 코어 상태 정보(134)를 이용하여 인터럽트 제어기 상태(interrupt controller state)를 복구하는 것; 및

하나 이상의 입출력(Input/Output, I/O) 서브시스템들을 복구하는 것

중 적어도 하나를 포함하는 것을 특징으로 하는 처리 시스템을 동작시키는 방법.

청구항 6

제1항에 있어서,

상기 프로세서 디바이스(126)를 초기화하는 것은 상기 제 1 부트 구성 정보(130)의 부트 코드(128)를 실행시켜 상기 제 1 부트 구성 정보의 복사본을 상기 비휘발성 메모리(104)로부터 상기 휘발성 메모리(102)로 전달하는 것을 포함하는 것을 특징으로 하는 처리 시스템을 동작시키는 방법.

청구항 7

처리 시스템으로서,

상기 처리 시스템은 프로세서 디바이스(126)를 포함하고,

상기 프로세서 디바이스(126)는,

휘발성 메모리(102)에 결합가능한 메모리 제어기(114)와;

상기 메모리 제어기(114)에 결합된 리셋 제어기(reset controller)(116)와, 여기서 상기 리셋 제어기는 상기 프로세서 디바이스(126)에 공급되는 전력이 감소되는 저전력 상태로부터 재시작하도록 하는 표시에 응답하여 상기 메모리 제어기(114)를 구성하게 되고; 그리고

상기 메모리 제어기(114)에 결합된 프로세서 코어(106)를 포함하고,

상기 프로세서 코어(106)는,

상기 프로세서 디바이스(126)의 전력-온 리셋에 응답하여 비휘발성 메모리(104)에 저장된 제 1 부트 구성 정보(130)를 이용하여 상기 프로세서 디바이스(126)를 초기화하게 되고, 여기서 상기 프로세서 코어(106)는 상기 제 1 부트 구성 정보(130)의 부트 코드를 실행시켜 제 2 부트 구성 정보(132)의 복사본을 상기 비휘발성 메모리(104)로부터 상기 휘발성 메모리(102)로 전달함으로써 상기 프로세서 디바이스(126)를 초기화하게 되고, 그리고

상기 프로세서 코어(106)는,

상기 저전력 상태로부터 재시작하도록 하는 상기 표시에 응답하여 상기 메모리 제어기(114)를 통해 상기 휘발성 메모리(102)로부터 액세스된 상기 제 2 부트 구성 정보(132)를 이용하여 상기 프로세서 디바이스(126)를 다시 초기화하게 되는 것을 특징으로 하는 처리 시스템.

청구항 8

제7항에 있어서,

상기 처리 시스템은 또한 상기 비휘발성 메모리(104)를 포함하고,

상기 비휘발성 메모리(104)는 상기 제 1 부트 구성 정보(130)를 저장하게 되고,

상기 제 1 부트 구성 정보(130)는, 상기 프로세서 코어(106)에 의해 실행될 때 상기 제 1 부트 구성 정보(130)의 적어도 일부분을 상기 휘발성 메모리(102)에 저장하도록 상기 프로세서 코어(106)를 조작하는 부트 코드(128)를 포함하는 것을 특징으로 하는 처리 시스템.

청구항 9

제8항에 있어서,

상기 프로세서 코어(106)는 또한 상기 전력-온 리셋에 응답하여 상기 비휘발성 메모리(104)로부터 상기 부트 코드(128)를 실행하게 되는 것을 특징으로 하는 처리 시스템.

청구항 10

제7항에 있어서,

상기 제 2 부트 구성 정보(132)는 부트 코드(136)를 포함하고,

상기 프로세서 코어(106)는 상기 부트 코드(136)를 실행함으로써 상기 프로세서 디바이스(126)를 다시 초기화하게 되는 것을 특징으로 하는 처리 시스템.

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

발명의 설명

기술 분야

[0001] 본 개시는 일반적으로 처리 시스템에 관한 것으로, 더 구체적으로 처리 시스템들을 위한 전력 상태 전이들에 관한 것이다.

배경 기술

[0002] 처리 시스템은 종종 상대적인 비활성도의 기간 동안 전력 소비를 감소시키기 위해 하나 이상의 슬립(sleep) 상태들을 구현한다. 이들 슬립 상태들 중 몇몇은 전력으로부터 프로세서 디바이스의 하나 이상의 구성요소들, 또는 전체 프로세서 디바이스를 제거하는 것을 수반하며, 이것은 운영 체제가 제어를 재시작할 수 있기 전에 슬립 상태를 빠져나갈 때 프로세서 디바이스의 재초기화를 필요로 한다. 이러한 재초기화는 종래에 또한 "콜드 부트(cold boot)"라 불리는, 전력-온 리셋으로부터의 초기화와 동일한 방식으로 수행되며, 이것은 프로세서 디바이스 외부의 비휘발성 메모리로부터 액세스된(accessed) 기본 입/출력 시스템(BIOS)의 실행을 수반한다. 비휘발성 메모리로부터의 BIOS 코드의 액세스는 종종 비교적 느리고, 이에 따라 그러한 슬립 상태들을 빠져나갈 때 종래의 처리 시스템들에 의해 나타나는 비교적 긴 웨이크-업(wake-up) 대기 시간(latency)에 상당히 기여한다.

도면의 간단한 설명

[0003] 본 개시는 더 잘 이해될 수 있고, 첨부 도면들을 참조하여 당업자에게 다수의 특징들 및 장점들이 명백해진다.

도 1은 본 개시의 적어도 하나의 실시예에 따른 처리 시스템을 도시한 블록도이다.

도 2는 본 개시의 적어도 하나의 실시예에 따른 처리 시스템의 저전력 상태에 들어가는 방법을 도시한 흐름도이

다.

도 3은 본 개시의 적어도 하나의 실시예에 따른 처리 시스템을 초기화하고 재초기화하는 방법을 도시한 흐름도이다.

도 4는 본 개시의 적어도 하나의 실시예에 따라 집적 회로(IC) 디바이스를 설계하고 제조하는 방법을 도시한 흐름도이다.

상이한 도면들에서 동일한 도면 부호들의 이용은 유사하거나 동일한 항목들을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0004] 도 1 내지 도 4는, 전력이 휘발성 메모리에 유지되는 동안 처리 디바이스가 전력 공급 감소(powered down)되는 저전력 상태로부터 처리 시스템이 효과적으로 재시작하도록 하는 기술들의 예를 도시한다. 일실시예에서, 재시작 부트 구성 정보는 전력-온 리셋 또는 다른 콜드 부트 상태로부터 처리 시스템의 부트 프로세스의 부분으로서 휘발성 메모리에 저장된다. 저전력 상태에 들어가기 위한 표시자(indicator)가 주장될 때, 프로세서 디바이스는 적어도 최소의 전압이 휘발성 메모리에 제공되는 동안 전력 공급 감소되어, 휘발성 메모리가 저장된 정보를 유지하도록 한다. 저전력 상태로부터 빠져나가거나 재시작하기 위해 표시자를 수신하자마자, 프로세서 디바이스의 재시작 제어기는 비휘발성 메모리로부터 메모리 제어기를 위한 구성 정보에 액세스하고, 메모리 제어기를 통해 휘발성 메모리로의 액세스를 허용하도록 구성 정보를 이용하여 메모리 제어기를 구성하거나 그렇지 않으면 복구 한다. 프로세서 디바이스는, 저전력 상태 동안 휘발성 메모리에 의해 유지되고 구성된 메모리 제어기를 통해 액세스된 코어 상태 정보를 이용하여 하나 이상의 프로세서 코어들의 초기 정황들(contexts), 또는 상태들을 구성 할 수 있고, 하나 이상의 프로세서 코어들은, 저전력 상태 동안 휘발성 메모리에 의해 유지되고 구성된 메모리 제어기를 통해 액세스된 재시작 부트 코드를 실행함으로써 부트 프로세스를 완료할 수 있다. 저전력 상태로부터의 재시작의 프로세스 동안 휘발성 메모리로부터 코어 상태 정보 및 재시작 부트 코드의 액세스는 비휘발성 메모리로부터 액세스된 부트 코드 및 다른 부트 정보에 의존하는 종래의 재초기화 프로세스들에 비해 상당히 감소된 웨이크-업 대기 시간을 초래할 수 있으며, 이러한 개선된 대기 시간은 적어도 부분적으로 휘발성 메모리와 비휘발성 메모리 사이에서 일반적으로 발견된 액세스 시간 및 처리량에서의 차이로 인한 것이다. 더욱이, 이러한 프로세스는 비휘발성 메모리에서 부트 코드의 변형 또는 교환(swap)으로부터 초래된 인증되지 않은 액세스 또는 구성을 거부하기 때문에 개선된 동작 보안을 제공한다.

[0005] 다음의 설명에서, 처리 디바이스에 대해 "초기화"와 "재초기화" 사이에서 구별이 이루어진다. 처리 디바이스를 초기화하는 것은 전력-온 리셋, 또는 "콜드 부트"에 응답하여 발생한다. 이와 대조적으로, 처리 디바이스를 재초기화하는 것은, 처리 디바이스가 전력 공급 감소되는 저전력 상태, 또는 "웜 부트(warm boot)"로부터의 재시작 또는 복귀에 응답하여 발생한다. "부트"라는 용어는 임의의 다양한 산업-표준 또는 독점 시스템 초기화 규격들 또는 프로세스들, 기본 입/출력 시스템(BIOS), 확장가능 펌웨어 인터페이스(EFI), 통합된 EFI(UEFI), 등을 언급할 수 있다. 논의의 용이함을 위해, 본 개시의 실시예들은 BIOS의 정황의 예에서 설명될 수 있다. 하지만, 본 개시는 BIOS에 기초한 구현들에 한정되지 않고, BIOS에 대한 인용은 달리 주지되지 않았다면 다른 시스템 초기화 규격들에 대한 인용을 포함한다.

[0006] 도 1은 본 개시의 적어도 하나의 실시예에 따라 유효 슬립으로부터의 재시작(resume-from-sleep) 프로세스를 구현하는 처리 시스템(100)의 예를 도시한다. 처리 시스템(100)은 예를 들어, 테스크탑 컴퓨터, 노트북 컴퓨터, 태블릿 컴퓨터, 서버, 계산-가능한 셀룰러 폰, 개인용 디지털 단말기(personal digital assistant), 등을 포함 할 수 있다. 도시된 예에서, 처리 시스템(100)은 휘발성 메모리(102), 비휘발성 메모리(104), 하나 이상의 프로세서 코어(106)의 세트, 노스브리지(northbridge)(108), 입/출력(I/O) 제어기(110), 사우스브리지(southbridge)(112), 메모리 제어기(114), 재시작 제어기(116), 및 전력 관리 제어기(118)를 포함한다.

[0007] 예시의 용이함을 위해, 본 개시의 예시적인 기술들은 정황의 한정되지 않은 예에서 기재되며, 이에 의해 휘발성 메모리(104)는 동적 랜덤 액세스 메모리(DRAM)를 포함하고, 비휘발성 메모리는 플래쉬(flash) 메모리를 포함한다. 따라서, 휘발성 메모리(102) 및 비휘발성 메모리(104)는 또한 각각 DRAM(102) 및 플래쉬 메모리(104)로 언급된다. 하지만, 휘발성 메모리(102)는 DRAM에 한정되지 않고, 그 대신 예를 들어, 정적 RAM(SRAM), 싸이리스터 RAM(T-RAM), 제로-커패시턴스 RAM 등을 포함할 수 있다. 마찬가지로, 비휘발성 메모리(104)는 플래쉬 메모리에 한정되지 않고, 그 대신 예를 들어, 판독 전용 메모리(ROM), 또는 강유전성 RAM(F-RAM), 자기 저항 RAM(M-RAM) 등과 같은 다른 유형들의 비휘발성 RAM을 포함할 수 있다.

[0008] 프로세서 코어들(106)은 예를 들어, 하나 이상의 처리 유닛(CPU) 코어들, 하나 이상의 그래픽 처리 유닛(GPU)

코어들, 하나 이상의 디지털 신호 프로세서(DSP) 코어들 뿐 아니라 이들의 조합들을 포함할 수 있다. 예시하기 위해, 처리 시스템(100)은 하나 이상의 CPU 코어들 및 하나 이상의 GPU 코어들을 포함하는 가속화된 처리 유닛(APU)을 포함할 수 있다. 노스브리지(108)는 프로세서 코어들(106)의 세트, 전력 관리 유닛(118), I/O 제어기(110)에 결합되고, DRAM(102)에 결합되거나 결합가능하다. 노스브리지(108)는 본 명세서에 기재된 바와 같이 리셋 및 재시작 동작들을 제어하는데 사용된 제어 레지스터들(119)의 세트를 포함하거나, 이와 연관된다.

[0009] I/O 제어기(110)는 사우스브리지(112)에 결합되거나 결합가능하고, 노스브리지(108)와 사우스브리지(112) 사이의 인터페이스로서 작용한다. I/O 제어기(110)는 디스플레이 인터페이스(예를 들어, 디스플레이에 연결하기 위한 고해상도 멀티미디어 인터페이스(HDMI)), 이산 GPU에 연결하기 위한 주변 구성요소 인터페이스-익스프레스(PCIe) 인터페이스, 확장 카드, 또는 다른 주변 기기, 하나 이상의 주변 디바이스들에 연결하기 위한 범용 직렬 버스(USB) 인터페이스 등과 같이 주변 구성요소들에 연결하기 위한 하나 이상의 집적된 I/O 인터페이스들을 포함할 수 있다. 사우스브리지(112)는 플래쉬 메모리(104)에 결합되고, I/O 제어기(110)와 플래쉬 메모리(104) 사이의 인터페이스로서 작용한다. 사우스브리지(112)는 하나 이상의 드라이브들에 연결하기 위한 직렬 개선된 기술 부착물(SATA) 인터페이스, 이산 GPU에 연결하기 위한 주변 구성요소 인터페이스-익스프레스(PCIe), 확장 카드, 또는 다른 주변 기기, 하나 이상의 주변 디바이스들에 연결하기 위한 범용 직렬 버스(USB) 인터페이스 등과 같이, 주변 구성요소들에 연결하기 위한 하나 이상의 I/O 인터페이스들을 더 포함할 수 있다. 메모리 제어기(114)는 DRMA(102)에 결합되거나 결합가능하고, DRMA(102)을 위한 인터페이스로서 작용한다. 메모리 제어기(114)는 도시된 예에서 노스브리지(108)의 부분으로서 도시된다.

[0010] 전력 관리 제어기(118)는 개선된 구성 및 전력 인터페이스(ACPI) 규격에 따른 것과 같이 하나 이상의 전력 상태 규격들에 따라 처리 시스템(100)의 특정한 구성요소들을 위한 전력 상태들을 구현하도록 구성된다. 운영 체제(OS)(120) 또는 다른 구성요소에 의해 주장된 전력 상태를 구현하도록 예시하기 위해, 전력 관리 제어기(118)는 하나 이상의 구성요소들을 위한 클록 주파수들을 변화시키거나, 전력 레일로부터 하나 이상의 구성요소들을 연결하거나 연결 해제하고, 하나 이상의 구성요소들에 공급된 전압을 변화시키거나, 이들의 조합을 이를 수 있다.

[0011] 재시작 제어기(116)는 메모리 제어기(114)에 결합되고, 메모리 제어기(114)를 위한 메모리 제어기(MC) 구성 정보가 저장되는 비휘발성 메모리에 결합된다. 비휘발성 메모리는 예를 들어, 사우스브리지(112)에서 구현된 예를 들어, 플래쉬 메모리(104) 또는 NVRAM(124)를 포함할 수 있다. 아래에 더 구체적으로 설명되는 바와 같이, 처리 시스템(102)이 슬립 상태로부터 재시작한다는 표시에 응답하여, 재시작 제어기(116)는 비휘발성 메모리로부터 MC 구성 정보{MC 상태 이미지(138)}에 액세스하고, 메모리 제어기(114)를 통해 DRAM(102)으로의 액세스를 가능하게 하도록 메모리 제어기(114)를 구성한다. 재시작 제어기(116)는 마이크로코드 또는 다른 펌웨어, 또는 이들의 조합을 실행하는 마이크로제어기로서, 본 명세서에 기재된 기능들을 수행하기 위해 고정 배선(hard-wired)로직 및 다른 회로로서 구현될 수 있다.

[0012] 적어도 하나의 실시예에서, 처리 시스템(100)은, 특정한 구성요소들의 회로가 형성되는, 하나 이상의 다이(die), 또는 "칩들(chips)"을 포함하는 프로세서 디바이스(126)를 포함한다. 예시된 예에서, 프로세서 디바이스(126)는 프로세서 코어들(106)의 세트, 노스브리지(108), I/O 제어기(110), 메모리 제어기(114), 재시작 제어기(116), 및 전력 관리 제어기(118)를 포함하는 집적 회로(IC) 디바이스이다. 사우스브리지(112), DRMA(102), 및 플래쉬 메모리(104)는 예를 들어, 마더보드를 통해 프로세서 디바이스(126)에 연결된 외부 디바이스들로서 구현된다. 다른 실시예들에서, 프로세서 디바이스(126)는 사우스브리지(112), DRAM(102), 및 플래쉬 메모리(104) 중 하나 이상을 포함할 수 있다.

[0013] 본 개시에서 특정한 관계의 처리 시스템(100)의 메인 동작 프로세스들은 (1) 전력-온 리셋, 또는 "콜드 부트"프로세스; (2) 슬립 상태(S3)에 대한 중지 프로세스; (3) 슬립 상태(S3)로부터의 재시작 프로세스를 포함한다. 전력-온 리셋 프로세스 동안, 마이크로코드는 처리 시스템(100)의 예비 초기화를 수행하기 위해 부트스트랩 프로세서(BSP)로서 작용하는 프로세서 코어들(106) 중 하나에 의해 실행되고, 그 후에 하나 이상의 프로세서 코어들(106) 각각의 지시 포인터(IP)는 플래쉬 메모리(104)에 저장된 리셋 BIOS 이미지(130)의 리셋 부트 코드(128)와 연관된 리셋 벡터로 로딩된다. 리셋 BIOS 이미지(130)는 리셋 부트 코드(128) 및 다른 BIOS 정보와 같이, 처리 시스템(100)을 초기화하는데 사용된 리셋 부트 구성 정보를 포함한다. 프로세서 코어들(106)은 처리 시스템(100)의 초기화를 완료하기 위해 리셋 부트 코드(128)를 실행하고, 이 지점에서 제어는 OS(120)로 전달된다.

[0014] 일실시예에서, 리셋 부트 코드(128)는 재시작 BIOS 이미지(132)의 복사본을 플래쉬 메모리(104)로부터 DRAM(102)으로 전달하기 위해 프로세서 디바이스(126)를 조작하도록 구성된 코드를 포함하며, 이에 의해 재시작 BIOS 이미지(132)는 슬립 상태(S3)로부터 재시작할 때 처리 시스템(100)을 재초기화하는데 사용하기 위한 재시

작 부트 구성 정보를 나타낸다. 그러므로, 재시작 BIOS 이미지(132)는 하나 이상의 코어 상태 이미지들(134) 및 재시작 부트 코드(136)를 포함할 수 있다. 각 코어 상태 이미지(134)는 대응하는 프로세서 코어(106)의 프로세서 상태를 나타내는 정보를 포함하며, 이러한 정보는 예를 들어, 기계 상태 레지스터들(MSR), 일반적인 목적의 레지스터들(GPRs), 부유점(FP) 레지스터들과 같은 다양한 레지스터들에서의 값들 뿐 아니라 개선된 프로그래밍 가능 인터럽트 제어기(APIC) 상태, 다양한 I/O 서브시스템 상태 정보 등을 포함한다. 재시작 부트 코드(136)는 프로세서 디바이스(126)의 재초기화를 완료하기 위해 프로세서 디바이스(126)를 조작하도록 구성된 코드를 포함한다. 더욱이, 리셋 부트 코드(128)는 메모리 제어기(MC) 상태 이미지(138)를 NVRAM(124)과 같은 다른 비휘발성 메모리로 전달하기 위해 프로세서 디바이스(126)를 조작하도록 구성된 코드를 포함할 수 있다. 대안적으로, MC 상태 이미지(138)는 플래쉬 메모리(104)에 유지된다. MC 상태 이미지(138)는 메모리 제어기(114)를 구성하기 위해 사용된 MC 구성 정보를 포함하고, 이러한 정보는 예를 들어, 타이밍 정보, 어드레스 매핑 정보 등을 포함할 수 있다. 전력-온 리셋 프로세스는 도 3을 참조하여 더 구체적으로 설명된다.

[0015] 동작 동안 몇몇 지점에서, OS(120)는 슬립 상태(S3)로의 진입을 신호 발신(signal)할 수 있고, 이에 응답하여, 전력 관리 제어기(118)는 슬립 상태(S3)에 대한 중지 프로세스를 개시한다. 이러한 프로세스 동안, 프로세서 디바이스(126)는 전력 공급 감소되고, DRMA(102)은 자가-리프레쉬(self-refresh) 상태에 위치된다. 자가-리프레쉬 상태에서, DRAM(102)은 재시작 BIOS 이미지(132)와 연관된 재시작 부트 구성 정보를 유지한다. S3 슬립 상태에 대한 이러한 중지 프로세스는 도 2를 참조하여 더 구체적으로 설명된다.

[0016] 몇몇 후속 지점에서, 사우스브리지(112)는 웨이크 타이머의 만료와 같은 웨이크 이벤트, 주변 디바이스에 의한 인터럽트의 생성, 전력 버튼 또는 다른 "웨이크"-유형의 버튼의 사용자 조작 등에 응답하여 슬립 상태(S3)로부터의 나감(exit)을 신호 발신할 수 있다. 신호 발신된 나감에 응답하여, 재시작 제어기(116)는 NVRAM(124)(또는 대안적으로, 플래쉬 메모리(104))로부터 MC 상태 이미지(138)를 액세스하고, 프로세서 디바이스(126)가 DRAM(102)에 액세스하도록 메모리 제어기(114)를 구성하거나 그렇지 않으면 복구한다. 일단 메모리 제어기(114)가 구성되면, 각각의 또는 하나 이상의 프로세서 코어들(106)의 지시 포인터는 재시작 부트 코드(136)의 대응하는 재시작 벡터로 로딩되고, 프로세서 코어들(106)은 처리 시스템(100)의 재초기화를 완료하기 위해 재시작 부트 코드(136)의 대응하는 부분들을 실행하도록 해제된다. S3 슬립 상태 프로세스로부터의 이러한 재시작은 도 3을 참조하여 더 구체적으로 설명된다.

[0017] 도 2는 본 개시의 적어도 하나의 실시예에 따라 S3 슬립 상태로 중지하기 위해 처리 시스템을 준비하는 방법(200)을 도시한다. 논의의 용이함을 위해, 방법(200)은 도 1의 처리 시스템(100)의 정황의 예에서 설명된다. 블록(202)에서 마지막 사용자 입력 또는 다른 슬립-모드 상태 아래로 비교적 긴 휴식(lull)에 응답하여, OS(120)는 예를 들어, 처리 시스템(100)을 S3 슬립 상태에 위치시키는 인터럽트 핸들러를 불러오는(invokes), 시스템 관리 인터럽트(SMI)와 같은 인터럽트를 신호 발신함으로써, 또는 직접 OS 또는 더 높은 레벨의 소프트웨어 지시에 의해 S3 상태로의 중지를 신호 발신한다.

[0018] 프로세서 디바이스(126)는, 리셋으로부터의 복귀가 S3 슬립 상태로부터의 재시작으로 인해 또는 전력-온 리셋으로 인해 이루어지는 지의 여부를 나타내기 위해 플래그(S3_RESUME_FLAG)를 이용한다. 따라서, 블록(204)에서, 인터럽트 핸들러 또는 재시작 제어기(116)는, 처리 시스템(100)이 S3 상태에 있는 동안 전력 공급 상태로 남아 있을 하나 이상의 플롭들(flops) 또는 다른 저장 요소들에 지정된 값(예를 들어, "1")을 저장함으로써 플래그(S3_RESUME_FLAG)를 주장한다. 일실시예에서, 플래그(S3_RESUME_FLAG)는 초기에 사우스브리지(112)(S3 슬립 상태 동안 전력 공급 상태로 남아있는)의 NVRAM(124)에 저장되고, 재시작 제어기(116)는 후속하여 이러한 플래그의 복사본을 S3 슬립 상태로부터 재시작하는 프로세스의 부분으로서 노스브리지(108)의 제어 레지스터들(119)로 이동시킨다.

[0019] 블록(206)에서, 인터럽트 핸들러 또는 재시작 제어기(116)는, 처리 시스템(100)이 S3 상태에 있는 동안 DRAM(102)에 저장된 정보를 유지하도록 DRAM(102)을 자가-리프레쉬 모드에 위치시키고, 블록(208)에서, 전력 관리 제어기(118)는 리셋 신호를 주장하고, 프로세서 디바이스(126)는, 리셋 신호가 주장된 상태로 남아있는 동안에 전력 공급 감소된다. 본 명세서에 사용된 "전력 공급 감소"라는 용어는, 레지스터들, 캐쉬들, 플롭들, 및 다른 저장 요소들의 상태들이 신뢰성있게 유지되지 않는 상태로 위치되는 것을 언급한다. 이것은 프로세서 디바이스(126)의 일부 또는 모든 구성요소들로부터 전력을 완전히 제거함으로써, 프로세서 디바이스의 일부 또는 모든 구성요소들에 비-동작(non-operational) 전압을 제공(즉, 저장 요소들의 유지 전압보다 낮은 전압을 제공)함으로써, 프로세서 디바이스의 일부 또는 모든 구성요소들을 클록-게이팅(clock-gating)함으로써 등에 의해 달성될 수 있다. 이 지점에서, 처리 시스템(100)은 S3 슬립 상태에 있도록 고려된다.

[0020] 도 3은 본 개시의 적어도 하나의 실시예에 따라 전력-온 리셋으로부터 또는 S3 슬립 상태로부터의 재시작으로부터 프로세서 디바이스(126)를 개시하기 위한 방법(300)의 예를 도시한다. 방법(300)은, 프로세서 디바이스(126)를 위한 리셋 신호가 주장 해제(deasserted)되는 블록(302)에서 개시한다. 리셋 신호의 주장 해제는 전력-온 리셋으로 인해 발생할 수 있어서, 콜드 부트를 신호 발신하거나, S3 슬립 상태로부터의 동작 상태로의 재시작을 나타내는 신호 발신으로 인해 발생할 수 있다. 후자는, 처리 시스템(100)이 S3 슬립 상태 등에 있는 동안 전력 버튼 또는 사용자 입력 디바이스의 사용자 조작에 응답하여, 예를 들어 타이머의 만료에 의해 신호 발신될 수 있다. 프로세서 디바이스(126)는, 예비 초기화가 수행될 때까지 프로세서 코어들(106)이 부트 코드를 실행시키는 것을 멎게(stall) 하기 위해 플래그(WAIT_BOOT_INIT)를 이용하고, 이에 따라 이 플래그는 WAIT_BOOT_INIT=1을 설정함으로써 블록(302)에서 멎음 상태로 초기화될 수 있다.

[0021] 블록(304)에서, 리셋의 주장 해제는 재시작 제어기(116)에 전파되고, 이러한 재시작 제어기(116)는 블록(306)에서 플래그(S3_RESUME_FLAG)의 상태를 체크함으로써 재시작 프로세스를 개시한다. 재시작 제어기(116)가 S3_RESUME_FLAG가 주장됨(예를 들어, 1과 동일함)을 결정하여, 리셋의 주장 해제가 S3 상태로부터의 재시작의 결과였음을 나타내면, 블록(308)에서, 재시작 제어기(116)는 DRAM(102)을 자가-리프레쉬 모드로부터 벗어나게 한다. 재시작 제어기(116)는 플래쉬 메모리(104), 또는 S3 슬립 상태 동안 저장되고, MC 상태 이미지(138)에 나타난 정황 정보를 이용하여 메모리 제어기(114)의 이전 상태를 구성하거나 그렇지 않으면 복구하는 NVRAM(124)으로부터 MC 상태 이미지(138)를 액세스한다. 이러한 구성은 메모리 제어기(114)를 위한 타이밍 정보를 복구하는 것과, 이전의 어드레스 매펑 정보(예를 들어, 페이지 테이블들을 구성하는) 등을 복구하는 것을 포함할 수 있다.

[0022] 프로세서 디바이스(126)는, 리셋으로부터의 복귀가 S3 상태로부터의 재시작인지의 여부에 관계없이 BSP로 신호 발신하기 위해 플래그(S3_EXIT_RST)를 이용한다. 메모리 제어기(114)를 구성한 후에, 재시작 제어기(116)는 블록(310)에서 플래그(S3_EXIT_RST)를 주장한다. 전술한 바와 같이, 프로세서 디바이스(126)는, 예비 초기화가 수행될 때까지 프로세서 코어들(106)이 부트 코드를 실행하는 것으로부터 멈추기 위해 플래그(WAIT_BOOT_INIT)를 이용한다. 따라서, 메모리 제어기(114)를 복구하고 플래그(S3_EXIT_RST)를 주장한 후에, 블록(312)에서, 재시작 제어기(116)는 플래그(WAIT_BOOT_INIT)를 주장 해제하여, 아래에 설명된 실행을 시작하기 위해 BSP 및 다른 프로세서 코어들(106)을 해제한다. 플래그들(S3_EXIT_RST 및 WAIT_BOOT_INIT)은 예를 들어, 노스브리지(108)의 하나 이상의 제어 레지스터들(119)에 위치될 수 있다.

[0023] 블록(306)으로 되돌아가면, 재시작 제어기(116)가 S3_RESUME_FLAG가 주장되지 않음을 결정하여, S3 슬립 상태와 다른 상태로부터 전력-온 리셋 또는 복귀를 나타내면, 플래그(S3_EXIT_RST)는 주장되지 않은 상태(예를 들어, "0")로 남아있고, 방법(300)의 흐름은 블록(306)으로부터 블록(312)으로 도약하고, 이에 의해 플래그(WAIT_BOOT_INIT)는 전술한 바와 같이 주장 해제된다.

[0024] 블록들(304-312)의 프로세스와 병렬로, 블록(302)에서의 리셋의 주장 해제는 블록(314)에서 BSP 및 다른 프로세서 코어들(106)로 전파된다. 이에 응답하여, 블록(316)에서, BSP는 전력-온 자가 테스트(POST)를 수행하는 것, 코어 빌트-인 자가-테스트(BIST)를 작용하는 것, 퓨즈들을 코어들에 분배하는 것, 및 초기화 시퀀스의 나머지에 대해 준비하도록 요구된 내부 정황을 초기화하는 것과 같이 프로세서 디바이스(126)를 예비로 초기화하도록 리셋 마이크로코드의 실행을 초기화한다. 예비 초기화가 완료된 후에, 블록(318)에서, 프로세서 코어들(106)은, 플래그가 주장 해제될 때까지 제어 레지스터들(119)에서의 플래그(WAIT_BOOT_INIT)를 반복적으로 폴링(poll)하여, 실행을 시작하기 위해 프로세서 코어들(106)을 해제한다. 일단 이렇게 해제되면, 블록(320)에서, 프로세서 코어들(106)은, 블록(302)에서의 리셋의 주장 해제가 전력-온 리셋(S3_EXIT_RST = 0)에 의해 야기되거나 S3 슬립 상태(S3_EXIT_RST = 1)로부터의 재시작에 의해 야기되었는지의 여부를 결정하기 위해 플래그(S3_EXIT_RST)에 액세스한다.

[0025] 플래그(S3_EXIT_RST)가 주장 해제(예를 들어, "0")되는 경우에, 방법(300)의 흐름은 플래쉬 메모리(104)를 이용하여 수행된 BIOS 초기화 프로세스로 향한다. 따라서, 블록(322)에서, BSP는 리셋 마이크로코드의 실행을 완료하고, 이에 따라 프로세서 디바이스(126)의 예비 초기화를 완료한다. 이에 응답하여, 블록(324)에서, BSP의 지시 포인터는 콜드 부트 BIOS 초기화에 사용된 디폴트 리셋 벡터로 설정되고, 블록(326)에서, BSP는 시스템 초기화 프로세스를 완료하기 위해 플래쉬 메모리(104)로부터 리셋 BIOS 이미지(130)의 리셋 부트 코드(128)의 실행을 시작한다. 일실시예에서, 리셋 부트 코드(128)는 DRAM(102)에서 재시작 BIOS 이미지(132)의 복사본을 저장하도록 BSP를 조작하기 위한 지시들을 포함하고, 이에 따라 리셋 부트 코드(128)의 실행은 블록(328)에서 플래쉬 메모리(104)로부터 DRAM(102)으로의 재시작 BIOS 이미지(132)의 복사본의 전달과, 블록(329)에서 플래쉬 메모리(104)로부터 NVRAM(124)으로의 MC 상태 이미지(138)의 복사본의 전달을 포함할 수 있다. 리셋 부트 코드(128)의

실행이 완료될 때, 하나 이상의 프로세서 코어들(106)의 IP는 OS(120)의 초기 벡터에 맞추어 설정되고 처리 시스템(100)의 제어는 블록(330)에서 OS(120)로 전달되어, 처리 시스템(100)의 초기화를 완료한다.

[0026] 블록(320)으로 되돌아가면, 플래그(S3_EXIT_RST)가 주장(예를 들어, "1")되는 경우에, 방법(300)의 흐름은 플래쉬 메모리(104)가 아니라 DRAM(102)에 저장된 재시작 부트 구성 정보를 이용하여 수행된 BIOS 재초기화 프로세스로 향한다. 따라서, 블록(332)에서, 프로세서 코어들(106)의 일부 또는 모두는 복구된 메모리 제어기(114)를 통해 DRAM(102)으로부터 대응하는 코어 상태 이미지(134)에 액세스하기 위해 마이크로코드 또는 다른 펌웨어를 실행한다. 각 프로세서 코어(106)는 대응하는 코어 상태 이미지(134)에 저장된 정황 정보에 기초하여 그 정황을 로딩한다. 각 코어 상태 이미지(134)는 재시작 BIOS 이미지(132)의 재시작 부트 코드(136)의 대응하는 재시작 벡터를 포함하고, 블록(334)에서, 이러한 재시작 벡터는 대응하는 프로세서 코어(106)의 지시 포인터로 로딩된다. 블록(336)에서, 프로세서 코어들(106)은 처리 시스템(100)의 초기화를 완료하기 위해 각 재시작 벡터들에서 시작하여 DRAM(102)으로부터 재시작 부트 코드(136)의 실행을 개시한다. 재시작 부트 코드(136)의 실행이 완료될 때, 프로세서 코어들(106) 중 하나의 프로세서 코어의 IP는 OS(120)의 초기 벡터로 설정되고, 처리 시스템(120)의 제어는 블록(330)에서 OS(120)로 전달되어, 처리 시스템(100)의 재초기화를 완료한다.

[0027] 적어도 하나의 실시예에서, 전술한 장치 및 기술들은 도 1의 프로세서 디바이스(126)와 같이, 하나 이상의 집적 회로(IC) 디바이스들(또한 집적 회로 패키지들 또는 마이크로칩들로 언급됨)을 포함하는 시스템에서 구현된다. 전자 설계 자동화(EDA) 및 컴퓨터 보조 설계(CAD) 소프트웨어 툴들은 이들 IC 디바이스들의 설계 및 제조에 사용될 수 있다. 이들 설계 툴들은 일반적으로 하나 이상의 소프트웨어 프로그램들로서 나타난다. 하나 이상의 소프트웨어 프로그램들은, 회로를 제조하기 위해 제작 시스템을 설계하거나 적응하기 위해 프로세스의 적어도 일부분을 수행하도록 하나 이상의 IC 디바이스들의 회로를 나타내는 코드 상에서 작용하도록 컴퓨터 시스템을 조작하기 위해 컴퓨터 시스템에 의해 실행가능한 코드를 포함한다. 이러한 코드는 지시들, 데이터, 또는 지시들과 데이터의 조합을 포함할 수 있다. 설계 툴 또는 제조 툴을 나타내는 소프트웨어 지시들은 일반적으로 계산 시스템에 액세스가능한 컴퓨터 판독가능 저장 매체에 저장된다. 마찬가지로, IC 디바이스의 설계 또는 제조의 하나 이상의 단계들을 나타내는 코드는 동일한 컴퓨터 판독가능 저장 매체 또는 상이한 컴퓨터 판독가능 저장 매체에 저장되고 이로부터 액세스될 수 있다.

[0028] 컴퓨터 판독가능 저장 매체는 임의의 저장 매체, 또는 지시들 및/또는 데이터를 컴퓨터 시스템에 제공하기 위해 사용할 동안 컴퓨터 시스템에 의해 액세스가능한 저장 매체의 조합을 포함할 수 있다. 그러한 저장 매체는 광학 매체{예를 들어, 컴팩트 디스크(CD), 디지털 다용도 디스크(DVD), 블루-레이 디스크}, 자기 매체(예를 들어, 플로피 디스크, 자기 테이프, 또는 자기 하드 드라이브), 휘발성 메모리{예를 들어, 랜덤 액세스 메모리(RAM), 또는 캐쉬}, 비휘발성 메모리{예를 들어, 판독 전용 메모리(ROM) 또는 플래쉬 메모리}, 또는 마이크로 전기 기계 시스템들(MEMS)-기반의 저장 매체를 포함할 수 있지만, 여기에 한정되지 않는다. 컴퓨터 판독가능 저장 매체는 계산 시스템{예를 들어, 시스템 RAM 또는 ROM}에 내장될 수 있고, 계산 시스템{예를 들어, 자기 하드 드라이브}에 고정되게 부착될 수 있고, 계산 시스템{예를 들어, 광학 디스크 또는 범용 직렬 버스(USB)-기반의 플래쉬 메모리}에 제거가능하게 부착될 수 있거나, 유선 또는 무선 네트워크{예를 들어, 네트워크 액세스가능 저장부(NAS)}를 통해 컴퓨터 시스템에 결합될 수 있다.

[0029] 도 4는 본 개시의 적어도 하나의 실시예에 따라 본 발명의 하나 이상의 양상들을 구현하는 IC 디바이스의 설계 및 제조를 위한 방법(400)의 예를 도시하는 흐름도이다. 전술한 바와 같이, 다음의 프로세스들 각각에 대해 생성된 코드는 대응하는 설계 툴 또는 제조 툴에 의한 액세스 및 사용을 위해 컴퓨터 판독가능 저장 매체에 저장되거나 그렇지 않으면 구현된다.

[0030] 블록(402)에서, IC 디바이스를 위한 기능적 명세서가 생성된다. 기능적 명세서{종종 마이크로 아키텍처 명세서(MAS)라 언급됨}은 C, C++, 시스템C, 시뮬링크(Simulink), 또는 MATLAB을 포함하는 임의의 다양한 프로그래밍 언어들 또는 모델링 언어들에 의해 나타날 수 있다.

[0031] 블록(404)에서, 기능적 명세서는 IC 디바이스의 하드웨어를 나타내는 하드웨어 설명 코드를 생성하는데 사용된다. 적어도 하나의 실시예에서, 하드웨어 설명 코드는 적어도 하나의 하드웨어 설명 언어(HDL)를 이용하여 나타나는데, 이것은 IC 디바이스의 회로들의 형식적인 설명 및 설계에 대한 임의의 다양한 컴퓨터 언어들, 명세서 언어들, 모델링 언어들을 포함한다. 생성된 HDL 코드는 일반적으로 IC 디바이스의 회로들의 동작, 회로들의 설계 및 조직화를 나타내고, 시뮬레이션을 통해 IC 디바이스의 정확한 동작을 증명하도록 테스트한다. HDL의 예들은 아날로그 HDL(AHDL), 베리로그(Verilog) HDL, 시스템 베리로그 HDL, 및 VHDL을 포함한다. 동기화된 디지털 회로들을 구현하는 IC 디바이스들에 대해, 하드웨어 설명자(descriptor) 코드는 동기 디지털 회로들의 동작의

요약 표현(Abstract representation)을 제공하기 위해 레지스터 전달 레벨(RTL) 코드를 포함할 수 있다. 다른 유형들의 회로에 대해, 하드웨어 설명자 코드는 회로의 동작의 요약 표현을 제공하기 위해 행위-레벨(behavior-level) 코드를 포함할 수 있다. 하드웨어 설명 코드에 의해 나타난 HDL 모델은 일반적으로 설계 증명을 통과하기 위해 시뮬레이션 및 디버깅의 하나 이상의 순환들(rounds)을 받는다.

[0032] 하드웨어 설명 코드에 의해 나타난 설명을 증명한 후에, 블록(406)에서, 합성 툴은 IC 디바이스의 회로의 초기 물리적 구현을 나타내거나 한정하는 코드를 생성하기 위해 하드웨어 설명 코드를 합성하도록 사용된다. 일실시 예에서, 합성 툴은 회로 디바이스 순간들(예를 들어, 게이트들, 트랜지스터들, 저항들, 커패시터들, 인덕터들, 다이오드들 등) 및 회로 디바이스 순간들 사이의 네트들(nets) 또는 연결들을 포함하는 하나 이상의 넷리스트들(netlists)을 생성한다. 대안적으로, 넷리스트의 전부 또는 일부는 합성 툴의 이용 없이도 수동으로 생성될 수 있다. 하드웨어 설명 코드에서와 같이, 넷리스트들은 하나 이상의 넷리스트들의 최종 세트가 생성되기 전에, 하나 이상의 테스트 및 증명 프로세스들을 받을 수 있다.

[0033] 대안적으로, 개략적인 에디터 툴은 IC 디바이스의 회로의 개략도를 드래프트(draft)하는데 사용될 수 있고, 개략적인 캡처 툴은 결과적인 회로 도면을 캡처하고, 회로 도면의 구성요소들 및 연결성을 나타내는 하나 이상의 넷리스트들(컴퓨터 판독 가능 매체 상에 저장된)을 생성하는데 사용될 수 있다. 캡처된 회로 도면은 테스트 및 증명을 위해 시뮬레이션의 하나 이상의 순환들을 받을 수 있다.

[0034] 블록(408)에서, 하나 이상의 EDA 툴들은 IC 디바이스의 회로의 물리적 레이아웃을 나타내는 코드를 생성하기 위해 블록(406)에서 생성된 넷리스트들을 이용한다. 이러한 프로세스는, 예를 들어 IC 디바이스의 회로의 각 요소의 장소를 결정하거나 고정하기 위해 넷리스트들을 이용하는 배치(placement) 툴을 포함할 수 있다. 더욱이, 라우팅(routing) 툴은 넷리스트(들)에 따라 회로 요소들을 연결하는데 필요한 와이어들을 추가하고 라우팅하기 위해 배치 프로세스 상에서 구축된다. 결과적인 코드는 IC 디바이스의 3차원 모델을 나타낸다. 코드는 예를 들어, 그래픽 데이터베이스 시스템 II(GDSII) 포맷과 같은 데이터베이스 파일 포맷으로 나타날 수 있다. 이러한 포맷의 데이터는 일반적으로 기하학적(geometric) 형태들, 텍스트 라벨들, 및 계층형 형태로 회로 레이아웃에 관한 다른 정보를 나타낸다.

[0035] 블록(410)에서, 물리적 레이아웃 코드(예를 들어, GDSII 코드)는 제조 설비에 제공되는데, 이러한 제조 설비는 IC 디바이스를 제조하기 위해 제조 설비(예를 들어, 마스크 작업들을 통해)의 제조 툴들을 구성하거나 그렇지 않으면 적응하도록 물리적 레이아웃 코드를 이용한다. 즉, 물리적 레이아웃 코드는 하나 이상의 컴퓨터 시스템들로 프로그래밍될 수 있고, 이것은 제조 설비의 툴들의 동작 또는 그 안에서 수행된 제조 동작들을 전체적으로 또는 부분적으로 제어할 수 있다.

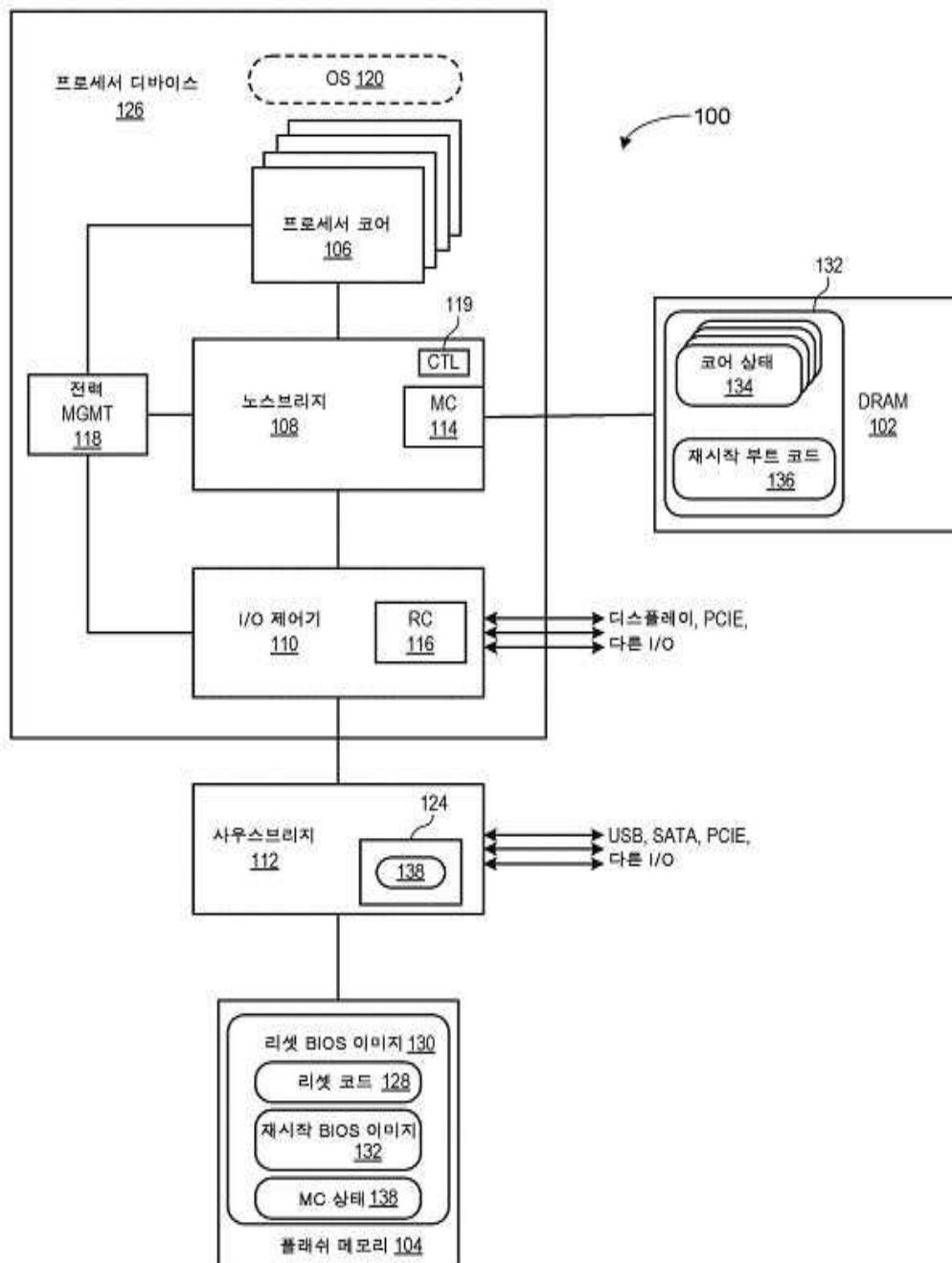
[0036] 일반적인 설명에서 전술한 작용들 또는 요소들이 모두 필요한 것은 아니며, 특정한 작용 또는 디바이스의 부분이 요구되고, 하나 이상의 추가 작용들이 수행되거나, 설명된 것들 외에도, 요소들이 포함된다는 것이 주지된다. 또한, 작용들이 기술되는 순서는 수행되는 순서일 필요는 없다.

[0037] 또한, 개념들은 특정 실시예들을 참조하여 설명되었다. 하지만, 아래의 청구항들에 설명된 본 개시의 범주에서 벗어나지 않고도 다양한 변형들 및 변경들이 이루어질 수 있다는 것을 당업자는 인식한다. 따라서, 명세서 및 도면들은 제한적인 관점이 아니라 예시적으로 간주되고, 모든 그러한 변형들은 본 개시의 범주 내에 포함되도록 의도된다.

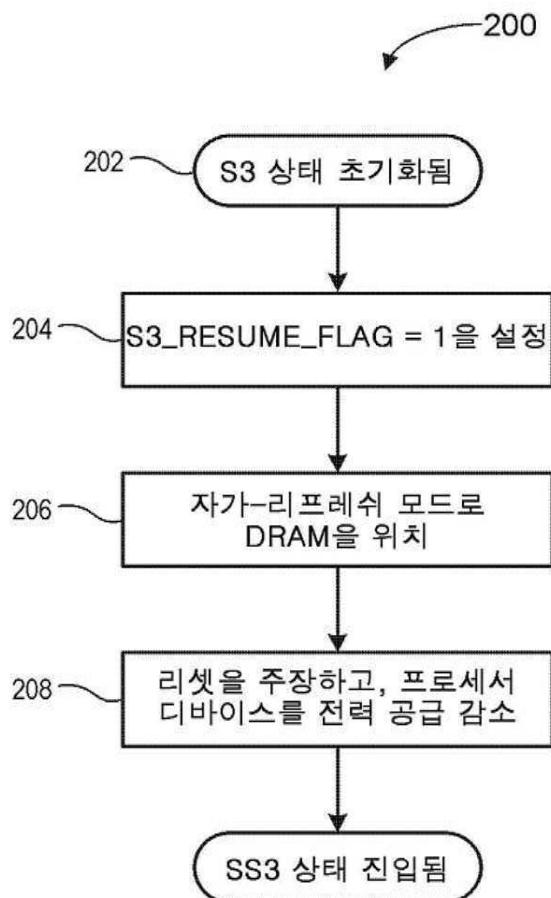
[0038] 문제점들에 대한 이익들, 다른 장점들, 및 해법들은 특정 실시예들에 관해 위에서 설명되었다. 하지만, 문제들에 대한 이점들, 장점들, 해법들, 및 임의의 이점, 장점 또는 해법이 더 많이 발생하거나 나타나도록 할 수 있는 임의의 특징(들)은 임의의 또는 모든 청구항들의 중요하고, 필요하거나 본질적인 특징으로서 해석되지 않아야 한다.

도면

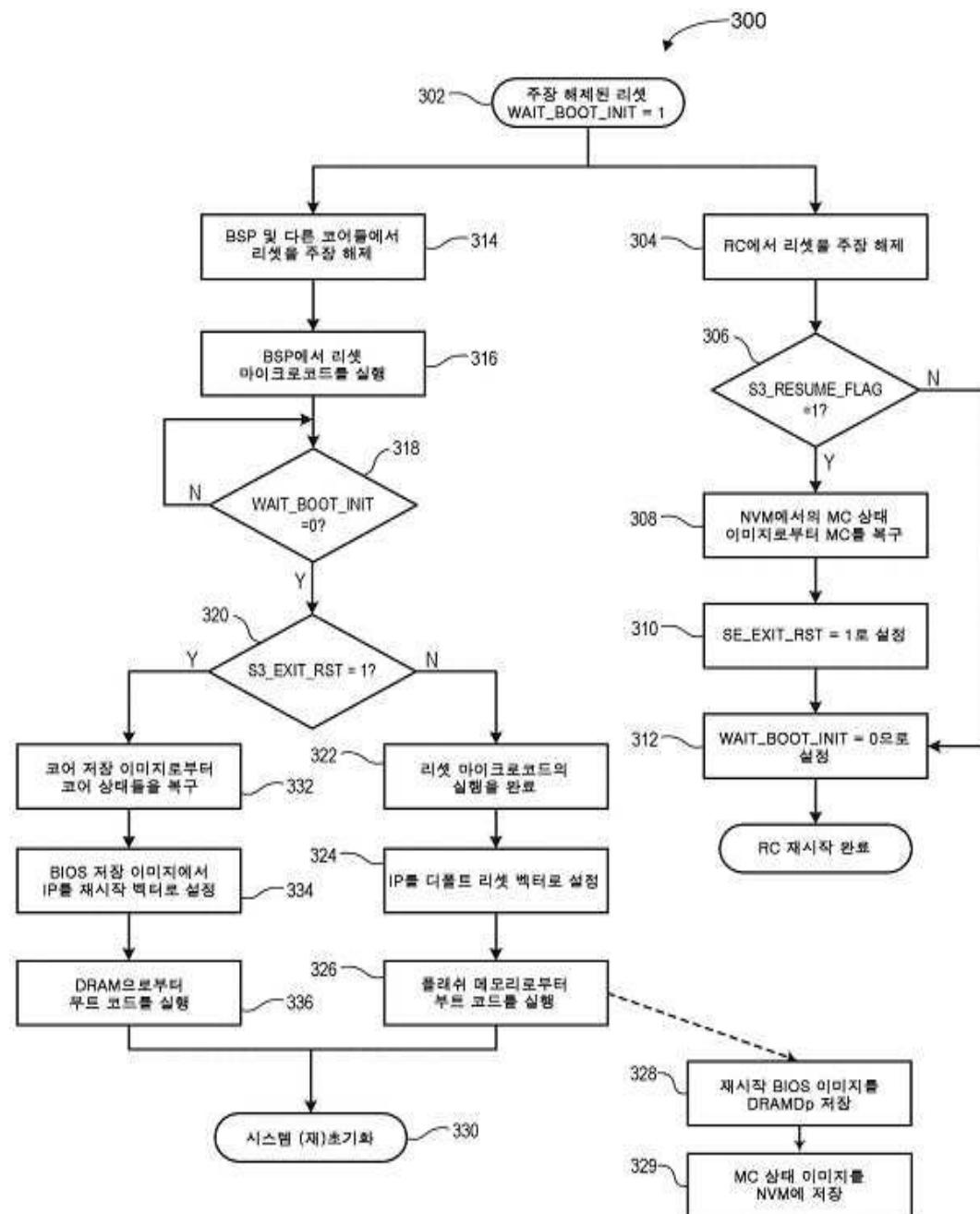
도면1



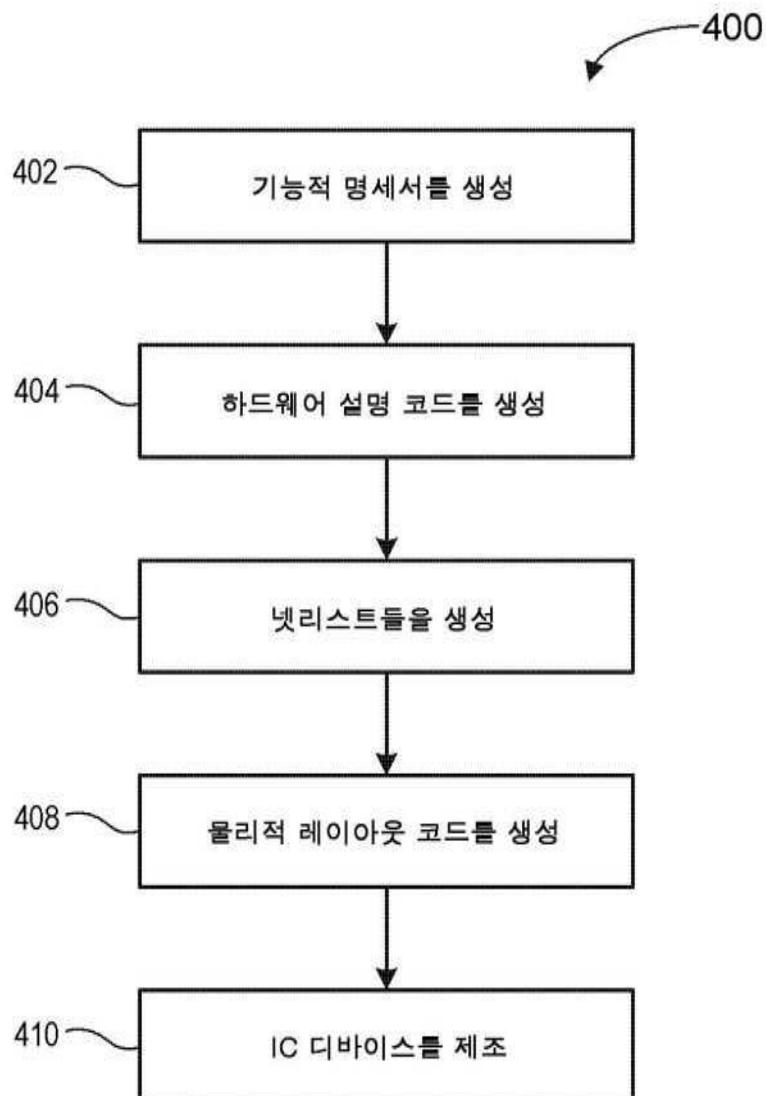
도면2



도면3



도면4



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 2

【변경전】

상기 비휘발성 메모리(104, 124)에 저장된

【변경후】

상기 비휘발성 메모리(104)에 저장된