



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년10월06일
(11) 등록번호 10-2309798
(24) 등록일자 2021년09월30일

(51) 국제특허분류(Int. Cl.)
G06F 13/16 (2006.01) G06F 12/02 (2018.01)
(52) CPC특허분류
G06F 13/1642 (2013.01)
G06F 12/0246 (2013.01)
(21) 출원번호 10-2015-0078506
(22) 출원일자 2015년06월03일
심사청구일자 2020년05월19일
(65) 공개번호 10-2016-0123930
(43) 공개일자 2016년10월26일
(30) 우선권주장
1966/CHE/2015 2015년04월16일 인도(IN)
(56) 선행기술조사문헌
US20140149985 A1
US20130254368 A1

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
나다쿠디티 스리니바사 라주
인도, 카르나타카, 방갈로르, 마라타할리, 문네코
라라, 바그데비 레이아웃, 7번 크로스, #7
삼파스 마드후수단
인도, 카르나타카, 방갈로르, 코디할리, 코디할리
메인 로드, 10, 사이언 알로에베라 아파트먼트,
#303
쿤눔푸라수 시반, 사라스 쿠마르
인도, 케랄라, 코타얌, 타자탄가디 피.오., 읍푸
델 파람빌(에이치)
(74) 대리인
특허법인가산

전체 청구항 수 : 총 10 항

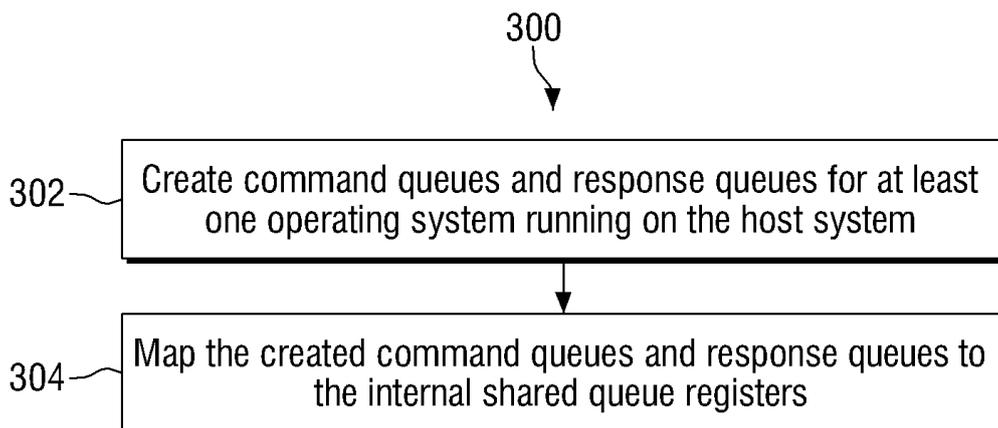
심사관 : 김세영

(54) 발명의 명칭 SR-I/OV 기반 비휘발성 메모리 컨트롤러 및 그 비휘발성 메모리 컨트롤러에 의해 큐에 리소스를 동적 할당하는 방법

(57) 요약

비휘발성 메모리(NVM) 컨트롤러에 의해 명령 큐 및 응답 큐에 리소스를 동적 할당하는 방법이 제공된다. 비휘발성 메모리(NVM) 컨트롤러에 의해 명령 큐 및 응답 큐에 리소스를 동적 할당하는 방법은 호스트 시스템(host system)에서 실행되는 복수의 운영 체제 중 적어도 하나의 운영 체제를 위해 명령 큐(command queue) 및 응답 큐(response queue)를 생성하되, 호스트 시스템에서 실행되는 복수의 운영 체제는 적어도 하나의 비휘발성 메모리 컨트롤러(controller)와 독립적으로 통신하고, 생성된 명령 큐 및 응답 큐를 복수의 내부 공유 큐 레지스터(internal shared queue registers)에 매핑(mapping)시키는 것을 포함한다.

대표도 - 도3



(52) CPC특허분류
G06F 13/1668 (2013.01)

명세서

청구범위

청구항 1

제1 명령 큐(command queue) 및 제1 응답 큐(response queue)를 생성하되,

비휘발성 메모리(NVM) 컨트롤러(controller)에 의해 호스트 시스템에서 실행되는 제1 운영 체제와 관련된 제1 입력 큐를 비휘발성 메모리 장치의 복수의 제1 레지스터에 매핑하고,

상기 비휘발성 메모리 컨트롤러에 의해 상기 호스트 시스템에서 실행되는 제2 운영 체제와 관련된 제2 입력 큐를 상기 복수의 제1 레지스터에 매핑하여 상기 제2 운영 체제가 상기 생성된 제1 명령 큐 및 제1 응답 큐를 공유하되,

상기 호스트 시스템에서 실행되는 복수의 상기 운영 체제는 적어도 하나의 비휘발성 메모리 컨트롤러와 독립적으로 통신하는, 비휘발성 메모리(NVM) 컨트롤러(controller)에 의해 명령 큐 및 응답 큐에 리소스(resource)를 동적 할당(dynamically allocating)하는 방법.

청구항 2

제 1항에 있어서,

상기 제1 명령 큐 및 제1 응답 큐는 SR-IOV를 지원하는 적어도 하나의 PF(물리 평선, physical function) 및 적어도 하나의 VF(버추얼 평선, virtual function)중 어느 하나로부터의 입력을 이용하여 생성되는, 비휘발성 메모리(NVM) 컨트롤러(controller)에 의해 명령 큐 및 응답 큐에 리소스(resource)를 동적 할당(dynamically allocating)하는 방법.

청구항 3

제 1항에 있어서,

상기 제2 입력 큐의 맵핑은 추가적인 명령을 생성할 때 발생하고,

상기 제2 입력 큐를 기반으로 하는 응답 큐는 상기 비휘발성 메모리 장치에 의해 지원되는 큐의 수를 초과하는, 비휘발성 메모리(NVM) 컨트롤러(controller)에 의해 명령 큐 및 응답 큐에 리소스(resource)를 동적 할당(dynamically allocating)하는 방법.

청구항 4

제 3항에 있어서,

상기 비휘발성 메모리 컨트롤러는 상기 비휘발성 메모리 장치 내에 배치되는, 비휘발성 메모리(NVM) 컨트롤러(controller)에 의해 명령 큐 및 응답 큐에 리소스(resource)를 동적 할당(dynamically allocating)하는 방법.

청구항 5

제 1항에 있어서,

각각의 운영 체제는 상기 호스트 시스템에서 실행되는 복수의 버추얼 머신에 의해 관리되는, 비휘발성 메모리(NVM) 컨트롤러(controller)에 의해 명령 큐 및 응답 큐에 리소스(resource)를 동적 할당(dynamically allocating)하는 방법.

청구항 6

명령 큐 및 응답 큐에 리소스(resource)를 동적 할당(dynamically allocating)하는 비휘발성 메모리(NVM) 컨트롤러(controller)를 포함하되,

상기 비휘발성 메모리 컨트롤러는,

호스트 시스템에서 실행되는 제1 운영 체제와 관련된 제1 입력 큐를 비휘발성 메모리 장치의 복수의 제1 레지스터에 매핑하여 명령 큐(command queue) 및 응답 큐(response queue)를 생성하고, ,

상기 비휘발성 메모리 컨트롤러에 의해 상기 호스트 시스템에서 실행되는 제2 운영 체제와 관련된 제2 입력 큐를 상기 복수의 제1 레지스터에 매핑하여 상기 제2 운영 체제가 상기 생성된 제1 명령 큐 및 제1 응답 큐를 공유하되,

상기 호스트 시스템에서 실행되는 복수의 상기 운영 체제는 적어도 하나의 비휘발성 메모리 컨트롤러와 독립적으로 통신하는, 비휘발성 메모리.

청구항 7

제 6항에 있어서,

상기 제1 명령 큐 및 제1 응답 큐는 SR-IOV를 지원하는 적어도 하나의 PF(물리 펑션, physical function) 및 적어도 하나의 VF(버추얼 펑션, virtual function)중 어느 하나로부터의 입력을 이용하여 생성되는, 비휘발성 메모리.

청구항 8

제 6항에 있어서,

상기 제2 입력 큐의 맵핑은 추가적인 명령을 생성할 때 발생하고,

상기 제2 입력 큐를 기반으로 하는 응답 큐는 상기 비휘발성 메모리 장치에 의해 지원되는 큐의 수를 초과하는, 비휘발성 메모리.

청구항 9

제 8항에 있어서,

상기 비휘발성 메모리 컨트롤러는 상기 비휘발성 메모리 장치 내에 배치되는, 비휘발성 메모리.

청구항 10

제 6항에 있어서,

각각의 운영 체제는 상기 호스트 시스템에서 실행되는 복수의 버추얼 머신에 의해 관리되는, 비휘발성 메모리.

발명의 설명

기술 분야

[0001] 본 발명은 비휘발성 메모리(non volatile memory) 장치 및 단일경로입출력가상화(Single Root Input/output Virtualization; 이하 SR-IOV)에 관한 것으로, 특히 버추얼(virtual) 및/또는 물리(physical) 펑션(function)을 포함하여, 비휘발성 메모리 장치에 의해 지원되는 큐 리소스(queue resource)의 분배를 가능케 하는 비휘발성 메모리 장치에 관한 것이다.

배경 기술

[0002] 정보처리 상호운용(interoperability), 낮은 대기시간, 높은 처리량 및 낮은 컴퓨트 오버헤드(compute overhead)에 대한 요구가 증가하는 것을 만족시키기 위해, PCI-SIG(Peripheral Component Interconnect Special Interest Group)는 다수의 물리적 PCIe (PCI express)장치를 별도로 나타낼 수 있는 SR-IOV 사양을 도입하였다. SR-IOV는 PF(물리 펑션, physical function)와 VF(버추얼 펑션, virtual function)를 통해 작동한다. PF는 SR-IOV 확장 기능(extended capability)을 포함하는 풀(full) PCIe 펑션(function)이다. 그 기능(capability)은 SR-IOV 기능성(functionality) 관리 및 구성에 사용된다. PF는 PCIe 기반 사양에서 모든 확장 기능이 포함되어있다. VF는 데이터 이동에 필요한 리소스(resource)를 포함하는 "경량"(lightweight) 펑션이지만, 적절하게 최소화된 구성 리소스 세트(set of configuration resources)를 가지고 있다. SR-IOV는, 예컨대 NIC(Network Interface Controller), CAN(Converged Network Adapter), HBA(Host Bus Adapter)와 같은 하드웨어에 VF를 이용하여 직접 액세스(direct access)하기 위한 휘발성 메모리(VM)를 구현할 수 있다.

[0003] 비휘발성 메모리 컨트롤러(non volatile memory controller)는, 일반적으로 비휘발성 메모리 장치에 명령을 내리거나, 비휘발성 메모리 장치로부터 응답을 얻기 위해 다중 명령 큐(command queue) 및 응답 큐(response queue)를 사용한다. 각 큐(명령 및 응답)는 속성(attribute)(예를 들어, 기준 주소(base address), 큐 크기, 헤드 포인터(head pointer), 테일 포인터(tail pointer) 등)의 세트를 포함한다. 만약 이러한 속성들이 HW 로직(logic)(플립 플랍)으로서 구현된다면, 플립 플랍의 수는, 지원되는(supported) 큐의 수에 따라 확장된다. SR-IOV 기반 호스트 시스템(host system)에 있어서, 명령 큐 및 응답 큐는 각각의 비추얼 머신(virtual machine)과 관련된 비추얼 평선(virtual function)으로부터 수신된 입력에 기초하여 생성된다.

[0004] 몇몇 비휘발성 메모리 호스트 컨트롤러 사양에서, 호스트에 의해 액세스 된 테일 포인터(Tail pointer) 및 헤드 포인터(Head pointer)와 같은 속성들은, 해당 컨트롤러에 대한 인접한 주소(contiguous address)에 나타난다. 예를 들어, NVMe 호스트 컨트롤러 레지스터(register) 인터페이스(interface) 사양에 따라, 호스트는 각 큐의 테일 및 헤드 포인터에 대한 고정된 오프셋 주소(fixed offset address)를 가정한다. 다만, 큐의 크기, 큐의 기준 주소와 같은 다른 속성은, 컨트롤러 내에 어디든 저장될 수 있다. 호스트 시스템에 의해 지원되는 비추얼 머신의 수가 증가함에 따라, 장치에 의해 지원되는 비추얼 평선의 수도 또한 증가할 것으로 예상된다. 이러한 이유로, 장치에 의해 지원되는 큐의 총 개수 또한 증가할 것이다. 다만, 이는 명령 큐 및 응답 큐의 개수를 증가시키는데 요구되는 하드웨어 로직을 구현하기 위한 게이트 카운트(gate count) 면에서는 비효율적이 될 수 있기 때문에, 비휘발성 메모리 장치는 제한된 큐(limited queue)를 구현할 수 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명이 해결하고자 하는 과제는, SR-IOV를 지원하는 비휘발성 메모리 장치를 위한 컨트롤러를 제공하는 것이다.

[0006] 본 발명이 해결하고자 하는 다른 과제는, 다수의 운영 체제(multiple operating system) 사이에서 큐 프로세싱(queue processing)을 위해 복수의 내부 큐 레지스터(internal queue register)를 분배하기 위한 방법을 제공하는 것이다. 비휘발성 메모리 장치에 의해 지원되는 큐의 수는 운영 체제 사이에서 공유된다.

[0007] 본 발명이 해결하고자 하는 또 다른 과제는, 생성된 명령 큐 및 응답 큐에 기초하여 복수의 내부 공유 큐 레지스터(internal shared queue register)의 리포지터리(repository)를 구현함으로써 비휘발성 메모리 장치의 전력 소비와 셀 면적을 줄이는 것이다.

[0008] 본 발명의 기술적 과제들은 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0009] 상기 기술적 과제를 달성하기 위한 본 발명의 일 실시예에 따른 비휘발성 메모리 컨트롤러에 의해 명령 큐 및 응답 큐에 리소스를 동적 할당하는 방법은, 호스트 시스템(host system)에서 실행되는 복수의 운영 체제(operating system) 중 적어도 하나의 운영 체제를 위해 명령 큐(command queue) 및 응답 큐(response queue)를 생성하되, 호스트 시스템에서 실행되는 복수의 운영 체제는 적어도 하나의 비휘발성 메모리(NVM) 컨트롤러(controller)와 독립적으로 통신하고, 생성된 명령 큐 및 응답 큐를 복수의 내부 공유 큐 레지스터(internal shared queue registers)에 매핑(mapping)시키는 것을 포함한다.

[0010] 본 발명의 몇몇 실시예에서, 상기 명령 큐 및 응답 큐는 적어도 하나의 PF(물리 평선, physical function) 및 적어도 하나의 VF(비추얼 평선, virtual function)중 하나로부터의 입력을 이용하여 생성될 수 있다.

[0011] 본 발명의 몇몇 실시예에서, 상기 비휘발성 메모리 컨트롤러에 의해 생성된 명령 큐 및 응답 큐는 상기 복수의 내부 공유 큐 레지스터에 의해 지원되는(supported) 큐의 수에 기초하되, 상기 복수의 내부 공유 큐 레지스터에 의해 지원되는 상기 큐의 수는 비휘발성 메모리 장치에 의해 지원되는 큐의 수에 기초할 수 있다.

[0012] 상기 기술적 과제를 달성하기 위한 본 발명의 일 실시예에 따른 비휘발성 메모리는, 명령 큐 및 응답 큐에 리소스(resource)를 동적 할당(dynamically allocating)하기 위한 비휘발성 메모리(NVM) 컨트롤러(controller)를 포함하되, 상기 비휘발성 메모리 컨트롤러는, 호스트 시스템(host system)에서 실행되는 복수의 운영 체제 중 적어도 하나의 운영 체제를 위해 명령 큐(command queue) 및 응답 큐(response queue)를 생성하되, 상기 호스트 시스템에서 실행되는 복수의 운영 체제는 적어도 하나의 상기 비휘발성 메모리 컨트롤러와 독립적으로

통신하고, 상기 생성된 명령 큐 및 응답 큐를 복수의 내부 공유 큐 레지스터에 매핑(mapping)시킨다.

- [0013] 본 발명의 몇몇 실시예에서, 상기 명령 큐 및 응답 큐는 적어도 하나의 PF(물리 펑션, physical function) 및 적어도 하나의 VF(비취열 펑션, virtual function)중 하나로부터의 입력을 이용하여 생성될 수 있다.
- [0014] 본 발명의 몇몇 실시예에서, 상기 비휘발성 메모리 컨트롤러에 의해 생성된 상기 명령 큐 및 응답 큐는 상기 복수의 내부 공유 큐 레지스터에 의해 지원되는 큐의 수에 기초하되, 상기 복수의 내부 공유 큐 레지스터에 의해 지원되는 상기 큐의 수는 비휘발성 메모리(NVM) 장치에 의해 지원되는 큐의 수에 기초할 수 있다.
- [0015] 기타 실시예들의 구체적인 사항들은 상세한 설명 및 도면들에 포함되어 있다.

도면의 간단한 설명

- [0016] 도 1은 본 발명의 일 실시예에 따른 다수의 운영 체제가 비휘발성 메모리 컨트롤러를 이용하여 비휘발성 메모리 장치와 통신하는 시스템을 도시한 도면이다.
- 도 2는 본 발명의 일 실시예에 따른 SR-IOV를 지원하는 비휘발성 메모리 장치의 블록도이다.
- 도 3은 본 발명의 일 실시예에 따른 비휘발성 메모리 컨트롤러에 의해 리소스를 동적 할당(dynamically allocating) 하는 방법을 설명하기 위한 순서도이다.
- 도 4는 본 발명의 일 실시예에 따른 각 운영 체제에 의해 지원되는 비취열 펑션을 위해, 생성된 큐에 의해 내부 공유 큐 레지스터 리포지터리(repository)를 공유하는 것을 설명하기 위한 도면이다.
- 도 5는 본 발명의 일 실시예에 따른 생성된 명령 큐 및 응답 큐를 내부 공유 큐 레지스터에 매핑(mapping)시키는 것을 설명하기 위한 도면이다.
- 도 6A 및 6B는 명령 큐 및 응답 큐의 수에 대해 사용된 게이트의 수가 감소함을 보여주는 예시적인 그래프이다.

발명을 실시하기 위한 구체적인 내용

- [0017] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 것이며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 도면에서 표시된 구성요소의 크기 및 상대적인 크기는 설명의 명료성을 위해 과장된 것일 수 있다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭하며, "및/또는"은 언급된 아이템들의 각각 및 하나 이상의 모든 조합을 포함한다.
- [0018] 소자(elements) 또는 층이 다른 소자 또는 층의 "위(on)" 또는 "상(on)"으로 지칭되는 것은 다른 소자 또는 층의 바로 위뿐만 아니라 중간에 다른 층 또는 다른 소자를 개재한 경우를 모두 포함한다. 반면, 소자가 "직접 위(directly on)" 또는 "바로 위"로 지칭되는 것은 중간에 다른 소자 또는 층을 개재하지 않은 것을 나타낸다.
- [0019] 공간적으로 상대적인 용어인 "아래(below)", "아래(beneath)", "하부(lower)", "위(above)", "상부(upper)" 등은 도면에 도시되어 있는 바와 같이 하나의 소자 또는 구성 요소들과 다른 소자 또는 구성 요소들과의 상관관계를 용이하게 기술하기 위해 사용될 수 있다. 공간적으로 상대적인 용어는 도면에 도시되어 있는 방향에 더하여 사용시 또는 동작시 소자의 서로 다른 방향을 포함하는 용어로 이해되어야 한다. 예를 들면, 도면에 도시되어 있는 소자를 뒤집을 경우, 다른 소자의 "아래(below)" 또는 "아래(beneath)"로 기술된 소자는 다른 소자의 "위(above)"에 놓여질 수 있다. 따라서, 예시적인 용어인 "아래"는 아래와 위의 방향을 모두 포함할 수 있다. 소자는 다른 방향으로도 배향될 수 있고, 이에 따라 공간적으로 상대적인 용어들은 배향에 따라 해석될 수 있다.
- [0020] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소 외에 하나 이상의 다른 구성요소의 존재 또는 추가를 배제하지 않는다.
- [0021] 비록 제1, 제2 등이 다양한 소자나 구성요소들을 서술하기 위해서 사용되나, 이들 소자나 구성요소들은 이들 용어에 의해 제한되지 않음은 물론이다. 이들 용어들은 단지 하나의 소자나 구성요소를 다른 소자나 구성요소와 구별하기 위하여 사용하는 것이다. 따라서, 이하에서 언급되는 제1 소자나 구성요소는 본 발명의 기술적 사상

내에서 제2 소자나 구성요소 일 수도 있음은 물론이다.

- [0022] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않은 한 이상적으로 또는 과도하게 해석되지 않는다.
- [0023] ‘큐(queue)’는 프로세싱을 위한 장치에 명령을 전송하기 위해, 호스트 시스템에 의해 공통적으로 사용되는 메커니즘을 의미한다. 일반적으로, 큐의 명령은 우선순위에 따라, 선입선출(first-in first-out)에 기초하여 또는 목적에 맞다면 순서 상관 없이 실행된다.
- [0024] ‘명령 큐(command queue)’와 ‘전송 큐(submission queue)’는 동일한 의미(interchangeably)로 사용될 수 있고, 명령은 호스트에 의해 생성되고 전송 큐에 배치되는 것을 의미한다.
- [0025] ‘응답 큐(response queue)’와 ‘완료 큐(completion queue)’는 동일한 의미로 사용될 수 있고, 이는 실행된 명령을 호스트에 신호를 보내기 위해 사용된다.
- [0026] 여기에 개시된 실시예들은, 비휘발성 메모리 컨트롤러에 의해 리소스들을 명령 큐 및 응답 큐에 동적 할당하는 방법을 제공한다. 비휘발성 메모리 컨트롤러에 의해 리소스들을 명령 큐 및 응답 큐에 동적 할당하는 방법은, 호스트 시스템에서 실행되는 다수의 운영 체제 중 적어도 하나의 운영 체제를 위한 명령 큐 및 응답 큐를 생성하는 것을 포함한다. 호스트 시스템에서 실행되는 다수의 운영 체제는, 적어도 하나의 비휘발성 메모리 컨트롤러와 독립적으로 통신할 수 있다. 비휘발성 메모리 컨트롤러에 의해 리소스들을 명령 큐 및 응답 큐에 동적 할당하는 방법은, 생성된 명령 큐 및 응답 큐를 내부의 단일 큐 레지스터 리포지터리(single internal queue register repository)에 있는 내부 공유 큐 레지스터(internal shared queue register)에 매핑(mapping)시키는 것을 포함한다. 명령 큐 및 응답 큐는 PF(물리 펄션, physical function)와 VF(버추얼 펄션, virtual function)로부터 수신된 입력을 이용하여 생성된다. 명령 큐 및 응답 큐는 비휘발성 메모리 컨트롤러의 내부 공유 큐 레지스터 리포지터리에 의해 지원(supported)되는 큐의 수에 기초하여 생성된다. 나아가, 명령 큐 및 응답 큐는 비휘발성 메모리 장치에 의해 구현된 내부 공유 큐 레지스터에 매핑된다.
- [0027] 제안된 방법 및 시스템은, 비휘발성 메모리 장치가 SR-IOV를 지원하는 경우, 영역을 최적화하는 방법으로 비휘발성 메모리 장치 리소스를 활용하기 위한 간단하고, 비용적으로 효율적인 방법이다. 비휘발성 메모리 장치에 의해 지원되는 큐의 수가, 전용 레지스터 및 전용 큐와 같은 전용 하드웨어 로직을 이용하여 구현되는 종래의 명령 큐 및 응답 큐의 구현과는 달리, 제안된 방법은 사용 시나리오(usage scenario)에 기초하여 명령 큐 및 응답 큐를 구현하는 비휘발성 메모리 장치를 가능하게 한다.
- [0028] 나아가, 제안된 방법은 펄션(function) 하나에 대한 큐의 수 면에서, 확장성(scalability)을 허용할 수 있다. 예를 들어, 생성된 명령 큐 및 응답 큐를 위해 구현된 레지스터는, 내부 공유 큐 레지스터에 매핑된 레지스터 세트에 기초하기 때문에, VF1- 4 큐나 VF2- 8 큐와 같은 여러 구성들이 사용될 수 있다.
- [0029] 도 1은 몇몇 실시예에 있어서, 다수 운영 체제가 적어도 하나의 비휘발성 메모리 컨트롤러(114)를 이용하여 SR-IOV를 지원하는 하나의 비휘발성 메모리 장치(102)와 통신하는 시스템(100)을 도시한다. 도 1은 호스트 컴퓨터(104)인 시스템(100)을 도시하고, 이 호스트 컴퓨터(104)는 하이퍼바이저 스위치(Hypervisor switch)(106)를 이용하는 복수의 버추얼 머신을 지원하는 운영 체제(108)를 가지고 있다. 몇몇 실시예에서, 시스템(100)은 (복수의 버추얼 머신을 이용하는) 다중 시스템을 제공하는 서버 시스템일 수 있다. 비휘발성 메모리 장치(102)의 활용은, 단일의 물리적 인터페이스(single physical interface)를 공유하는 다수의 OS(운영 체제)를 가능하게 함으로써 극대화될 수 있다. 도 1은 SR-IOV를 지원하는 다중 컨트롤러(114)를 이용하여, 비휘발성 메모리 장치(102)와 통신하는 다수의 OS(110₁, 110₂, ..., 110_N)를 갖는 복수의 버추얼 머신(VM₁, VM₂...VM_N)을 도시한다. 몇몇 실시예에서, 하이퍼바이저 스위치(106)는 비휘발성 메모리 장치(102)와 통신하는 복수의 OS(110₁, 110₂, ..., 110_N)를 관리할 수 있다. 또한, 하이퍼바이저 스위치(106)는 각 OS(110)에 하드웨어 리소스를 할당할 수 있다. 각 VM(VM₁, VM₂, ..., VM_N)은 VF 드라이버(112₁, 112₂, ..., 112_N)를 가질 수 있고, VF 드라이버(112₁, 112₂, ..., 112_N)는, 호스트 메모리로부터 데이터를 복사하고 호스트 메모리에 데이터를 복사하는 VF에 전달할, 전송 디스크립터(transmit descriptor) 및 수신 디스크립터(receive descriptor)를 로드 업(load up) 및 사용할 수 있다. 컨트롤러는, PCIe 루트 콤플렉스(PCIe root complex)를 통한 통신 인터페이스(communication interface)로서 주로 사용되고, 컨트롤러는 비휘발성 메모리 액세스, 제어 및 관리 기능을 제공할 수 있다. SR-

IOV를 지원하는 비휘발성 메모리 장치(102)는, 다중 컨트롤러를 포함할 수 있고, 다중 큐를 지원할 수 있다. 비휘발성 메모리 컨트롤러(202)에 관한 상세한 사항은 도 2와 함께 설명한다.

- [0030] 몇몇 실시예에서, 비휘발성 메모리 장치(102)는 플래시 메모리(예컨대, (NAND) 플래시, PCM(phase change memory), SSD(solid state drive) 또는 전원이 끊겨도 데이터를 유지할 수 있는 다른 메모리)일 수 있다.
- [0031] 전용 큐 및 레지스터가 VM 및 VF 각각 하나에 대해 할당되는 종래의 SR-IOV 기반 비휘발성 메모리 장치와는 달리, 제안된 시스템은 큐가 요구(requirement)에 기초하여 동적인 방식(dynamic manner)으로 공유될 수 있다. VF의 수 및 각 VF에 의해 지원되는 큐는, 시스템 요구사항에 기초하여 계속 변경할 수 있다. 제안된 시스템 및 방법은, 비휘발성 메모리 장치에 의해 지원되는 큐의 수에 기초하여, 변경된 시스템 요구사항을 이용하는 비휘발성 메모리 컨트롤러를 가능하게 할 수 있다. 변경된 시스템 요구사항은, VF 하나에 대해 지원되는 큐 및 VF에 관한 것으로, 생성된 명령 큐 및 응답 큐를 단일 내부 공유 레지스터에 동적으로 할당 및 재할당을 하기 위한 것이다.
- [0032] 도 2는 본 발명의 몇몇 실시예에 따른 SR-IOV를 지원하는 비휘발성 메모리 장치(102)의 블록도를 도시하고 있다. 명령 큐 및 응답 큐는 각 운영 체제(110)에 대해 VF(112)를 이용하여 생성된다. VF(112)의 개수 및 각 VF(112)를 위해 지원되는 큐는 설정 가능(configurable)하다.
- [0033] 몇몇 실시예에서, 각 OS(110)에 대한 VF(112)의 수는 VMM(116)에 의해 정해질 수 있다.
- [0034] 비휘발성 메모리 장치(102)는 호스트 시스템(100)과 비휘발성 메모리 장치(102) 사이의 통신을 위한 다중 비휘발성 메모리 컨트롤러(202_{0-N})를 포함할 수 있다.
- [0035] 몇몇 실시예에서, 비휘발성 메모리 컨트롤러(202)는 내부 공유 큐 레지스터 리포지터리(206)를 포함하도록 구성될 수 있다. 비휘발성 메모리 컨트롤러(202)는 생성된 명령 큐 및 응답 큐를 각 운영 체제(110)를 위해 내부 공유 큐 레지스터 리포지터리(206)에 매핑(mapping)시킬 수 있다. 내부 공유 큐 레지스터 리포지터리(206)의 레지스터 세트는, VF(112)로부터의 입력에 기초하여, 운영 체제(110)로부터 생성된 명령 큐 및 응답 큐에 분배된다. 비휘발성 메모리 컨트롤러(202)는 명령 큐 내에서 명령을 실행할 수 있다. 큐 관리자 모듈(queue manager module)은 비휘발성 메모리 장치(102)에 의해 지원되는 큐의 수에 따라, VF(112) 하나에 대해 지원되는 큐의 최대 개수를 확장(scale)시킬 수 있다.
- [0036] 도 3은 본 발명의 일 실시예에 따른 비휘발성 메모리 컨트롤러(202)에 의해 리소스를 동적 할당하는 방법(300)을 설명하기 위한 순서도이다. 비휘발성 메모리 컨트롤러(202)에 의해 리소스를 동적 할당하는 방법(300)의 다양한 단계들은, 각 블록으로 요약될 수 있고, 각 블록은 비휘발성 메모리 장치(102), 비휘발성 메모리 컨트롤러(202), 호스트 컴퓨터(104) 등과 같은 다양한 모듈에 의해 수행될 수 있는 몇몇 단계이다. 비휘발성 메모리 컨트롤러(202)에 의해 리소스를 동적 할당하는 방법(300) 및 여기에 묘사되어 있는 다른 설명은, 마이크로 컨트롤러(microcontroller), 마이크로프로세서(microprocessor) 또는 이에 상당하는 것을 이용하여 구현될 수 있는 제어 프로그램을 위한 기초를 제공할 수 있다. 비휘발성 메모리 컨트롤러(202)에 의해 리소스를 동적 할당하는 방법(300)은, 호스트 시스템(100)에서 실행되는 복수의 운영 체제 중 적어도 하나의 운영 체제(110)를 위한 명령 큐 및 응답 큐를 생성하는 것을 포함한다(302). 비휘발성 메모리 컨트롤러(202)는 VF(112)를 이용하여 운영 체제로부터 명령을 수신한다. 비휘발성 메모리 컨트롤러(202)는 보급된 비휘발성 메모리 장치(102)에 따라, 명령 큐 테일 포인터 및 응답 큐 헤드 포인터를 구현할 수 있다.
- [0037] 비휘발성 메모리 장치(102)가 다수의 호스트와 관련된 다수의 비휘발성 메모리 컨트롤러를 포함할 때, 각 호스트의 큐의 테일 및 헤드 포인터는, 특정 주소 또는 특정 명령/우선순위를 가지고 각각의 호스트에 액세스할 수 있다. 큐 파라미터(queue parameter)는 호스트 컴퓨터(104)에 의해 프로그램(program)된다. 호스트 컴퓨터(104)로부터 수신된 큐의 수는, 비휘발성 메모리 장치(102)에 의해 지원되는 큐의 수를 초과할 수 없다.
- [0038] 비휘발성 메모리 컨트롤러(202)에 의해 리소스를 동적 할당하는 방법(300)은, 생성된 명령 큐 및 응답 큐를 내부 공유 큐 레지스터에 매핑시키는 것을 포함한다(304). 생성된 명령 큐 및 응답 큐는 내부 공유 큐 레지스터 리포지터리(206)에 매핑된다. 호스트 시스템(100)에 의해 생성된 각 큐는, 내부 공유 큐 레지스터 리포지터리(206)의 내부 공유 큐 레지스터에 매핑될 수 있다.
- [0039] 비휘발성 메모리 컨트롤러(202)에 의해 리소스를 동적 할당하는 방법(300)의 다양한 행동(action), 행위(act), 블록, 단계 등은, 제시된 순서로, 또는 상이한 순서로 수행될 수 있거나, 동시에 수행될 수 있다. 또한, 몇몇 실시예에서, 몇몇 행동, 행위, 블록, 단계 등은, 본 발명의 범위를 벗어나지 않고 제외, 첨가, 수정, 생략

(skip) 등이 될 수 있다.

- [0040] 도 4는 본 발명의 몇몇 실시예에 따른 각 OS(110)에 의해 지원되는 VF(112)를 위해 생성된 큐에 의한 내부 공유 큐 레지스터 리포지터리(206)의 분배를 예시적으로 도시하고 있다. 도 4는 시스템(100)에 의해 지원되는 여러 OS(110)의 VF(112)로부터 수신된 큐를 도시하고 있다. 내부 공유 큐 레지스터 리포지터리(206)에서는, VF(112)를 이용하여 OS(110)로부터 수신된 큐가 수신된 순서대로 레지스터 세트에 저장된다. 각 큐는 NVMe 사양에 따라 QID를 이용하여 참조될 수 있다. 여기에서, 비휘발성 메모리 컨트롤러(202)가 OS1의 VF1을 위해 생성된 큐를 내부 레지스터 세트1에 매핑할 때를 고려한다(402). 비휘발성 메모리 컨트롤러(202)가 OS2의 VF1으로부터 입력을 수신할 때, OS2의 VF1을 위해 생성된 큐를 내부 레지스터 세트2에 매핑시킨다(404). 비휘발성 메모리 장치(102)는, 생성된 큐의 수 및 내부 공유 큐 레지스터 리포지터리(206)에 이용된 레지스터 세트의 수에 기초하여, 내부 공유 큐 레지스터 리포지터리(206)의 명령 큐 및 응답 큐를 위한 하드웨어 로직을 구현할 수 있다. 따라서, 고정된 레지스터의 수가 각 VF(112)의 큐를 위해 구현되는 종래의 메카니즘과는 달리, 비휘발성 메모리 장치(102)에서 큐의 구현을 위해 이용되는 게이트 카운트 및 소비 전력에 관한 리소스들은 감소될 수 있다.
- [0041] 도 5에 도시된 바와 같이, 큐의 테일 및 헤드 포인터들은 호스트가 의도한 큐 아이디(ID)에 의해 인덱싱(indexing)된다. 다른 속성들은 컨트롤러에 의해 할당된 내부 큐 아이디에 의해 인덱싱된다. 이러한 방식으로, 제안된 방법은, 연속적으로 바뀌는 컨트롤러 하나에 대한 큐의 수를 지원하기 위해 필요한 하드웨어 리소스를 감소시킨다.
- [0042] 도 6A 및 6B는 본 발명의 실시예에 따른 명령 큐 및 응답 큐의 수에 대해 사용된 게이트의 개수의 감소를 나타내는 예시적인 그래프이다. 도 6A 및 6B는 종래의 방법 및 제안된 방법을 이용하여 비휘발성 메모리 장치(102) 내에서 명령 큐 및 응답 큐를 구현하기 위해 요구되는 게이트의 수를 나타낸다. 도 6A 및 6B에서, 명령 큐 및 응답 큐에 대한 비휘발성 메모리 장치(102) 구성 설정은 다음과 같다.
- [0043] 포트: 2개
- [0044] PF: 포트 당 1개
- [0045] VF: PF 당 16개의 VF
- [0046] IO 명령 큐: PF 당 128개 및 VF 당 16개의 큐
- [0047] 가정: 1비트 레지스터는 8개의 게이트를 요함
- [0048] 종래 방법을 이용한 명령 큐의 구현
- [0049] 큐의 총 개수: $128 * 2 + (16 * 16) * 2 = 768$
- [0050] 큐 당 레지스터 비트의 수= $192 = 6 \text{ DW}$
- [0051] 레지스터의 총 개수= $768 * 192 = 147456$
- [0052] 게이트의 총 개수= $147456 * 8 = \mathbf{1,179,648}$
- [0053] 제안된 방법을 이용한 명령 큐의 구현
- [0054] 큐의 총 개수: 256
- [0055] 큐 당 레지스터 비트의 수= $192 = 6 \text{ DW}$
- [0056] 레지스터의 총 개수= $256 * 192 = 49152$
- [0057] 게이트의 총 개수= $49152 * 8 = \mathbf{393,216}$
- [0058] 따라서, 도 5A에서 보여주는 바와 같이, 제안된 시스템 및 방법을 이용하면, 명령 큐를 구현하는데 요구되는 하드웨어 로직이, 종래 기술에 비해 약 66.6% 정도 절약된다.
- [0059] 종래 방법을 이용한 응답 큐의 구현
- [0060] 큐의 총 개수: $128 * 2 + (16 * 16) * 2 = 768$
- [0061] 큐 당 레지스터 비트의 수= 224
- [0062] 레지스터의 총 개수= $768 * 224 = 172032$

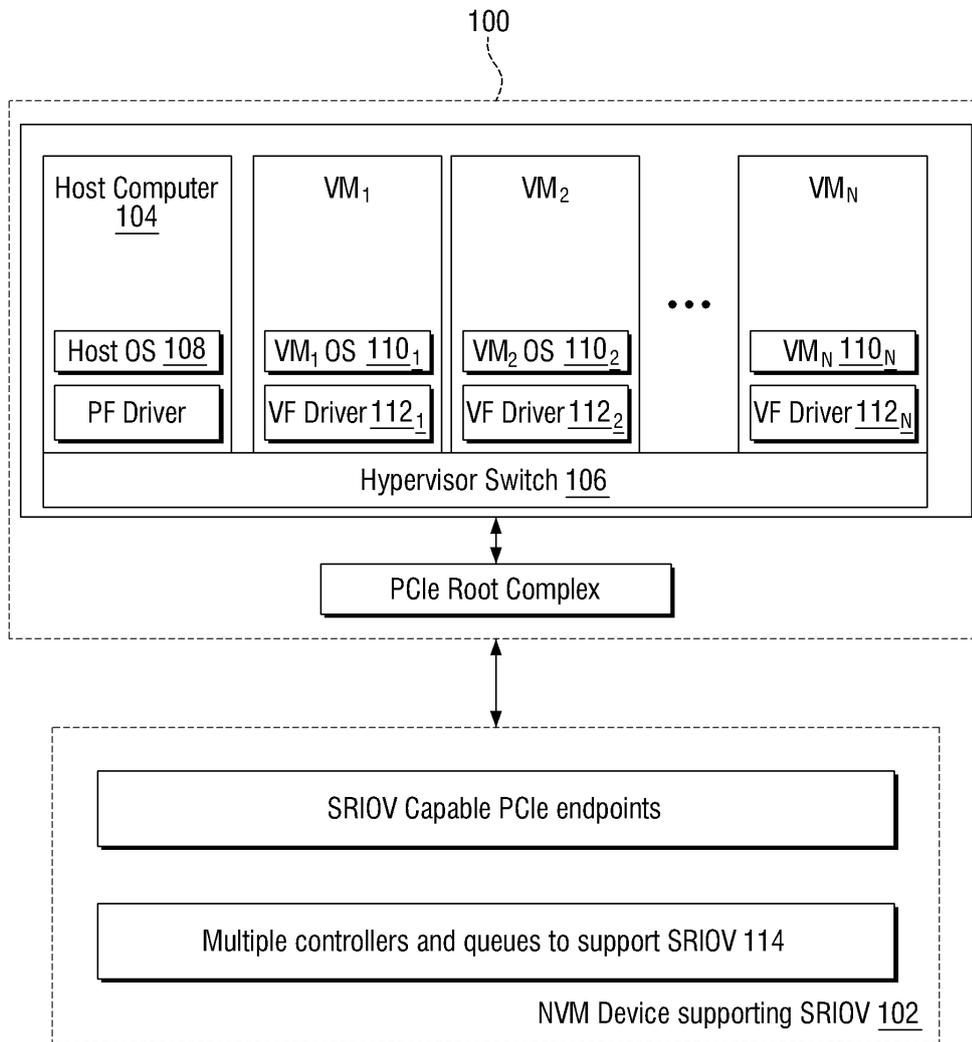
- [0063] 게이트의 총 개수= $172032 * 8 = 1,376,648$
- [0064] 제안된 방법을 이용한 응답 큐의 구현
- [0065] 큐의 총 개수: 256
- [0066] 큐 당 레지스터 비트의 수= 224
- [0067] 레지스터의 총 개수= $256 * 224 = 57344$
- [0068] 게이트의 총 개수= $57344 * 8 = 458,752$
- [0069] 따라서 제안된 시스템 및 방법을 이용하면, 응답 큐를 구현하는데 요구되는 하드웨어 로직이 종래 기술에 비해 약 66.6% 정도 절약된다.
- [0070] 개시된 실시예들은, 적어도 하나의 하드웨어 장치에서 실행되고 구성요소를 제어하기 위해 네트워크 관리 기능을 수행하는, 적어도 하나의 소프트웨어 프로그램을 통해 구현될 수 있다. 도 1 및 도 2에 나타난 구성요소들은, 하드웨어 장치 또는 하드웨어 장치와 소프트웨어 모듈의 조합 중 적어도 하나가 될 수 있는 블록을 포함한다.
- [0071] 이상 첨부된 도면을 참조하여 본 발명의 실시예들을 설명하였으나, 본 발명은 상기 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 제조될 수 있으며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다.

부호의 설명

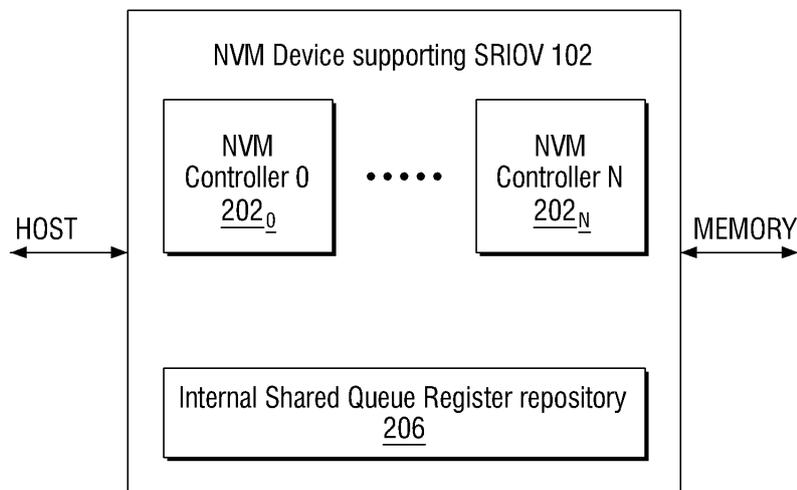
- [0072] 102: SR-IOV를 지원하는 비휘발성 메모리 장치
- 104: 호스트 컴퓨터
- 106: 하이퍼바이저 스위치
- 108: 호스트 운영 체제(OS)
- 110_{1-N}: 운영 체제(OS)
- 112_{1-N}: VF 드라이버
- 114: SR-IOV를 지원하는 다중 컨트롤러
- 202_{0-N}: 비휘발성 메모리 컨트롤러
- 206: 내부 공유 큐 레지스터 리포지터리

도면

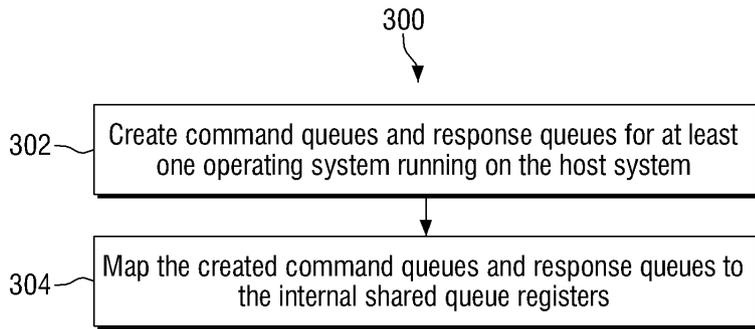
도면1



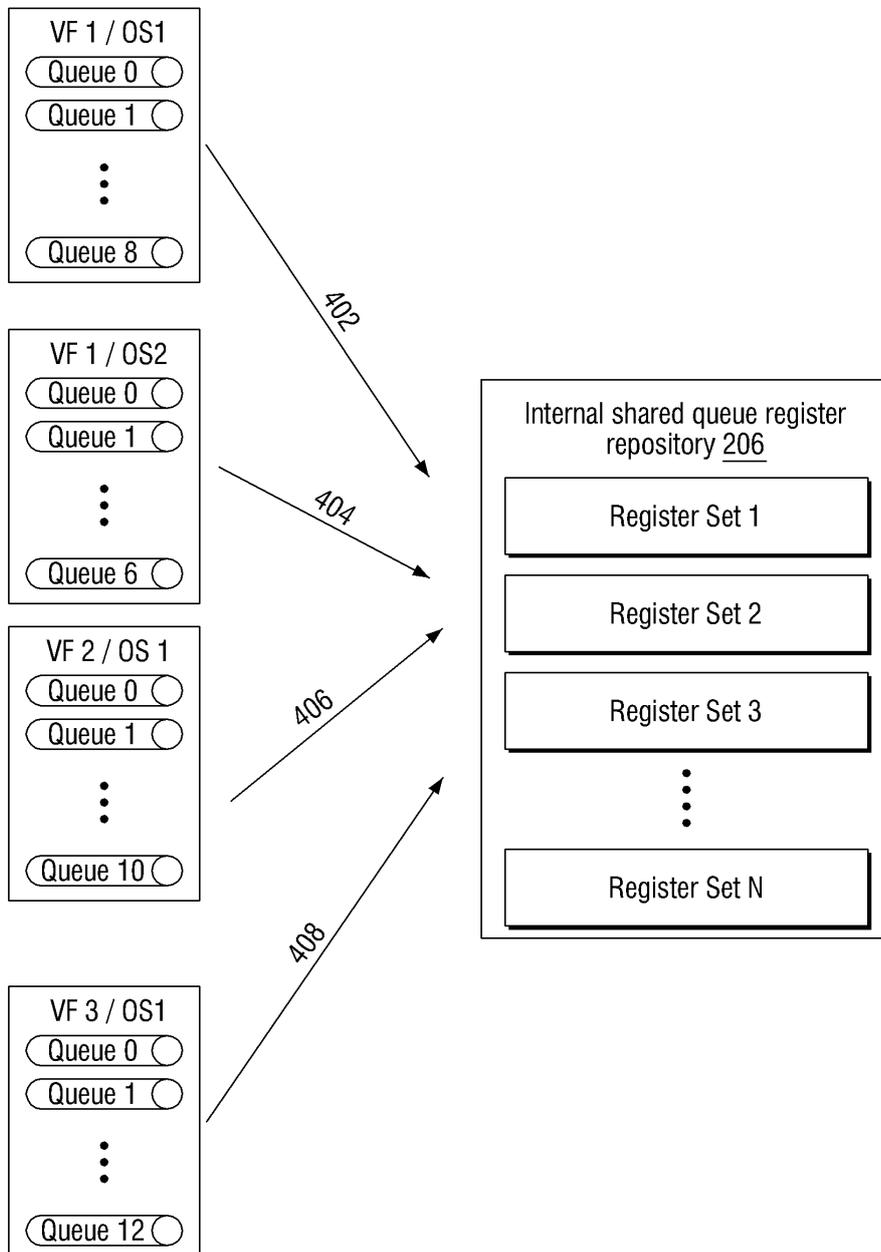
도면2



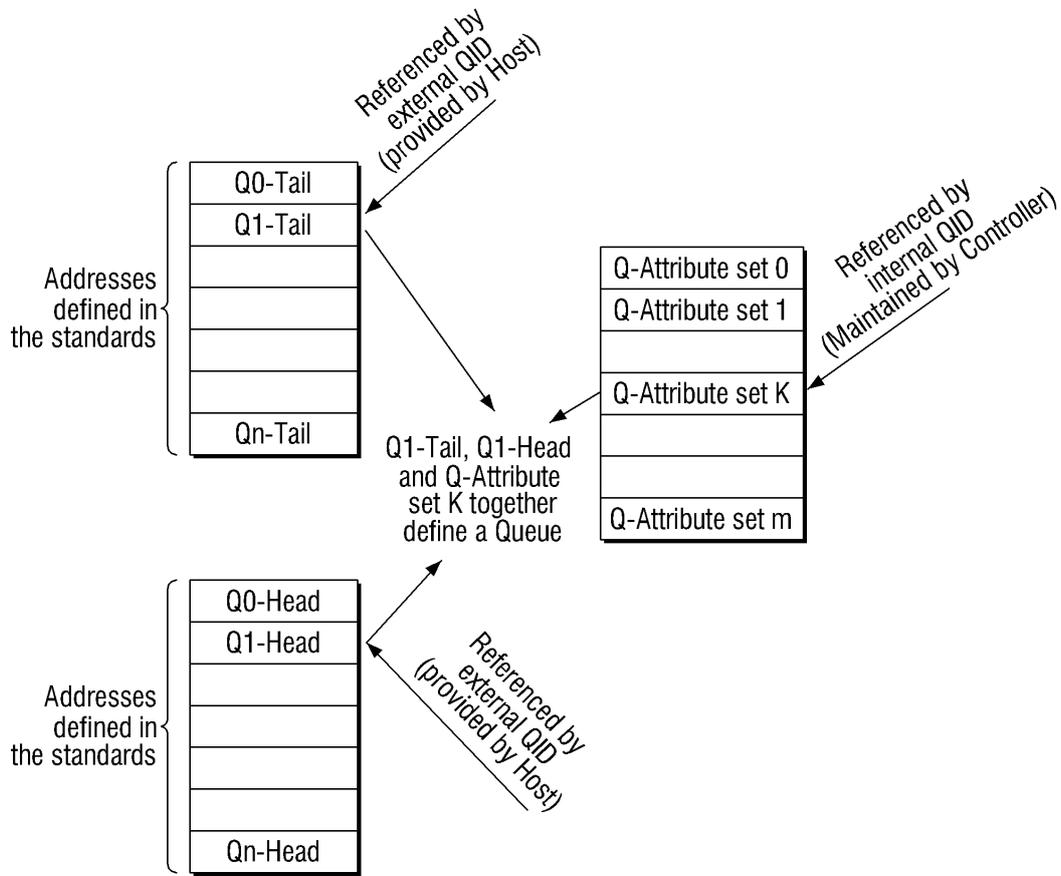
도면3



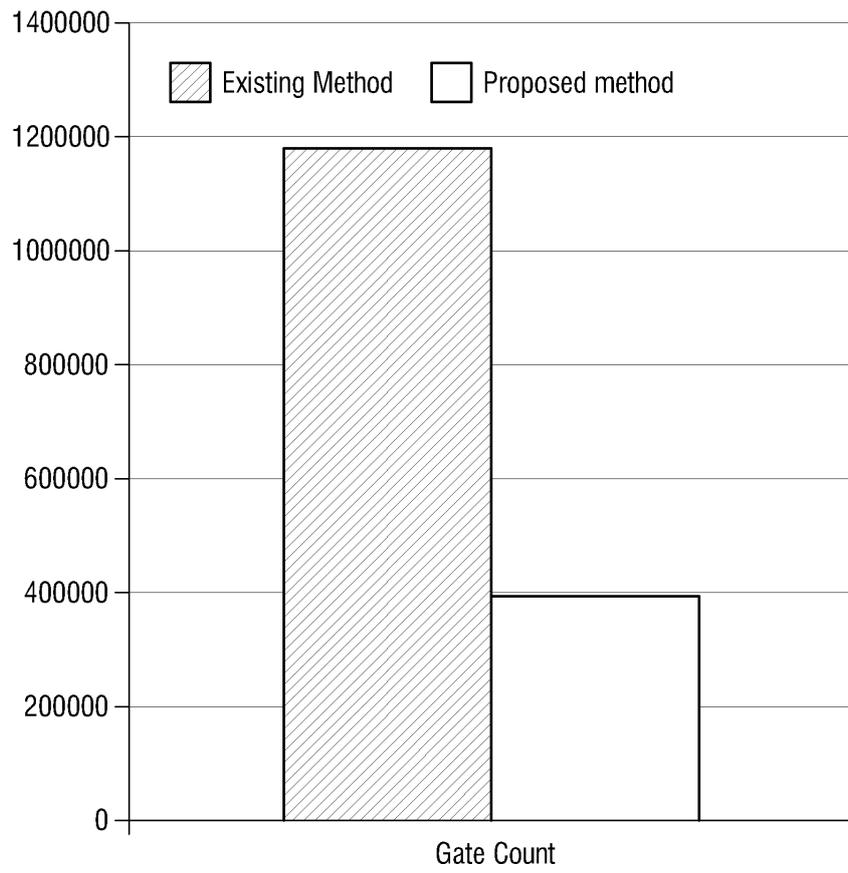
도면4



도면5



도면6a



도면6b

