(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년01월13일
(11) 등록번호 10-2202578
(24) 등록일자 2021년01월07일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *G06F 21/55* (2013.01)
G06F 21/71 (2013.01) *H04L 12/26* (2006.01)
(52) CPC특허분류
H04L 63/1458 (2013.01)
G06F 21/554 (2013.01)
(21) 출원번호 10-2019-7005580
(22) 출원일자(국제) 2017년06월29일
심사청구일자 2020년06월24일
(85) 번역문제출일자 2019년02월25일
(65) 공개번호 10-2019-0038581
(43) 공개일자 2019년04월08일
(86) 국제출원번호 PCT/US2017/039999
(87) 국제공개번호 WO 2018/031140
국제공개일자 2018년02월15일
(30) 우선권주장
15/230,388 2016년08월06일 미국(US)
(56) 선행기술조사문헌
US20050018618 A1
US20140254388 A1

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 95054 산타 클라라 어거스틴 드라이브 2485
(72) 발명자
로호 가브리엘
미국 워싱턴 98007 벨레뷰 슈트 300 156번 에비뉴 엔이 - 2002
스테인만 마우라이스 비
미국 메사추세츠 01719 박스보로우 플로어스 1 2 앤드 3 센트럴 스트리트 90
(74) 대리인
박장원

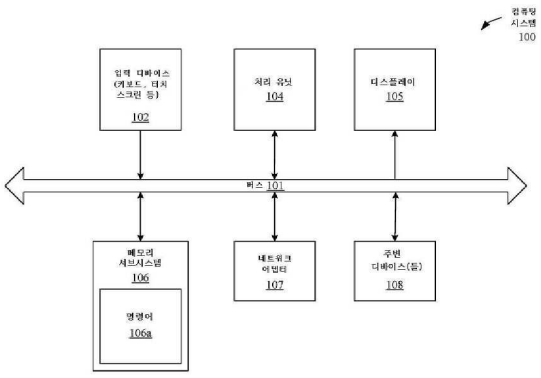
전체 청구항 수 : 총 19 항 심사관 : 문형섭

(54) 발명의 명칭 **비신뢰 상호접속 에이전트를 조절하기 위한 메커니즘**

(57) 요 약

호스트 SoC는, SoC의 내부 블록들 간에 로컬 트래픽을 송신하기 위한 NoC, 및 호스트 SoC에서 비신뢰 디바이스로부터의 메시지를 수신하기 위한 외부 프로세서 링크를 포함한다. 외부 프로세서 링크와 결합된 호스트 SoC의 트래픽 제어기는, 하나 이상의 시간 간격의 세트에 걸쳐 상기 비신뢰 디바이스로부터의 외부 트래픽의 양을 감시하고, 외부 트래픽의 양에 기초하여 트래픽 정책의 위반을 검출하고, 위반의 검출에 응답하여, 비신뢰 디바이스로부터의 메시지로 인해 발생하는 NoC의 트래픽을 감소시킨다.

대 표 도 - 도1



(52) CPC특허분류
G06F 21/71 (2013.01)
H04L 43/16 (2013.01)

명 세 서

청구범위

청구항 1

장치로서,

호스트 시스템-온-칩(systems on a chip: SoC)으로서, 상기 SoC의 내부 블록들 간에 로컬 트래픽을 송신하도록 구성된 네트워크 온 칩(network on chip: NoC)을 포함하는, 상기 호스트 SoC와;

상기 호스트 SoC에서 비신뢰(untrusted) 디바이스로부터의 메시지를 수신하도록 구성된 외부 프로세서 링크 - 상기 수신된 메시지들은 상기 호스트 SoC의 메모리에 대한 하나 이상의 메모리 요청을 포함하며 - 와; 그리고

상기 외부 프로세서 링크와 결합된 트래픽 제어기를 포함하되, 상기 트래픽 제어기는,

하나 이상의 시간 간격의 세트에 걸쳐 상기 NoC를 통해 전송된 상기 비신뢰 디바이스로부터의 외부 트래픽의 양을 감시하고,

상기 시간 간격의 세트 중 제1시간 간격 동안 상기 비신뢰 디바이스로부터 상기 호스트 SoC로 송신되는 메시지의 개수가 상기 제1시간 간격에 대한 최대 임계값을 초과하는지 여부를 결정함으로써 상기 외부 트래픽의 양에 기초하여 트래픽 정책의 위반을 검출하고,

상기 위반의 검출에 응답하여, 상기 비신뢰 디바이스로부터의 메시지로 인해 발생하는 상기 NoC의 트래픽을 감소시키도록 구성된, 장치.

청구항 2

제1항에 있어서, 상기 트래픽 제어기는,

상기 외부 프로세서 링크에서 수신되는 상기 비신뢰 디바이스로부터의 각 메시지에 응답하여 증분되고, 그리고

상기 제1시간 간격의 경과에 응답하여 리셋되도록 구성된 메시지 카운터를 포함하는, 장치.

청구항 3

제1항에 있어서, 상기 트래픽 제어기는,

상기 최대 임계값을 변경하기 위한 요청을 수신하고,

상기 요청의 인증에 응답하여 상기 요청에 따라 상기 최대 임계값을 변경하고, 그리고

상기 요청의 인증 실패에 응답하여 상기 요청을 무시하도록 구성된 구성 로직을 더 포함하는, 장치.

청구항 4

제1항에 있어서, 상기 트래픽 제어기는, 또한, 상기 비신뢰 디바이스로부터 상기 호스트 SoC로 송신되는 상기 메시지의 개수가 상기 제1시간 간격에 대한 상기 최대 임계값을 초과한다는 결정에 응답하여, 후속 시간 간격에 대한 최대 임계값을 상기 메시지의 개수와 상기 제1시간 간격에 대한 상기 최대 임계값 간의 차에 대응하는 양만큼 감소시키도록 구성된, 장치.

청구항 5

제1항에 있어서, 상기 트래픽 제어기는, 상기 시간 간격의 세트 중 N개의 가장 최근 시간 간격의 윈도우에 걸쳐 상기 비신뢰 디바이스로부터의 메시지의 평균 개수를 계산하고 상기 평균 개수가 최대 임계값을 초과한다고 결정함으로써 상기 트래픽 정책의 위반을 검출하도록 구성된, 장치.

청구항 6

제5항에 있어서, 상기 트래픽 제어기는,

시프트 버퍼 내의 복수의 레지스터로서, 상기 복수의 레지스터의 각 레지스터는 해당 레지스터에 연관된 상기 N개의 가장 최근 시간 간격 중 하나에 대한 메시지의 개수를 저장하도록 구성된, 복수의 레지스터; 및

상기 복수의 레지스터에 저장된 개수를 합산하도록 구성된 가산기를 포함하는, 장치.

청구항 7

제5항에 있어서, 상기 트래픽 제어기는,

상기 N개의 가장 최근 시간 간격 중 N-1개의 이전 시간 간격 각각에 대하여 검출되는 메시지의 개수의 시간-가중(time-weighted) 합을 저장하도록 구성된 제1레지스터;

상기 N개의 가장 최근 시간 간격 중 가장 최근 시간 간격에 대하여 검출되는 메시지의 개수를 저장하도록 구성된 제2레지스터; 및

상기 제2레지스터 내의 개수를 상기 제1레지스터 내의 시간-가중 합의 비트-시프트 버전(bit-shifted version)에 가산함으로써 새로운 시간-가중 합을 계산하도록 구성된 로직을 포함하는, 장치.

청구항 8

제1항에 있어서, 상기 트래픽 제어기는, 상기 비신뢰 디바이스로부터의 트래픽을 미리 정해진 기간 동안 제한함으로써 상기 NoC의 트래픽을 감소시키도록 구성된 조절 로직을 더 포함하는, 장치.

청구항 9

제1항에 있어서,

상기 트래픽 제어기는, 상기 로컬 트래픽이 로컬 트래픽 임계값을 초과하는지 여부를 결정하도록 구성된 로직 트래픽 모니터를 더 포함하고, 그리고

상기 트래픽 제어기는, 상기 로컬 트래픽이 상기 로컬 트래픽 임계값을 초과하는 경우 상기 위반의 검출에 응답하여 상기 NoC의 트래픽을 감소시키도록 더 구성된, 장치.

청구항 10

방법으로서,

호스트 SoC 내의 NoC를 통해 상기 호스트 SoC의 내부 블록들 간에 로컬 트래픽을 송신하는 단계와;

상기 호스트 SoC에서 외부 프로세서 링크를 통해 비신뢰 디바이스로부터의 메시지를 수신하는 단계 - 상기 수신된 메시지들은 상기 호스트 SoC의 메모리에 대한 하나 이상의 메모리 요청을 포함하며 - 와;

하나 이상의 시간 간격의 세트에 걸쳐 상기 NoC를 통해 전송된 상기 비신뢰 디바이스로부터의 외부 트래픽의 양을 감시하는 단계와;

상기 시간 간격의 세트 중 제1시간 간격 동안 상기 비신뢰 디바이스로부터 상기 호스트 SoC로 송신되는 메시지의 개수가 상기 제1시간 간격에 대한 최대 임계값을 초과하는지 여부를 결정함으로써 상기 외부 트래픽의 양에 기초하여 트래픽 정책의 위반을 검출하는 단계와; 그리고

상기 위반의 검출에 응답하여, 상기 비신뢰 디바이스로부터의 메시지로 인해 발생하는 상기 NoC의 트래픽을 감소시키는 단계를 포함하는, 방법.

청구항 11

제10항에 있어서,

상기 최대 임계값을 변경하기 위한 요청을 외부 디바이스로부터 수신하는 단계,

상기 요청의 인증에 응답하여 상기 요청에 따라 상기 최대 임계값을 변경하는 단계, 및

상기 요청의 인증 실패에 응답하여 상기 요청을 무시하는 단계를 더 포함하는, 방법.

청구항 12

제10항에 있어서, 상기 비신뢰 디바이스로부터 상기 호스트 SoC로 송신되는 상기 메시지의 개수가 상기 제1시간 간격에 대한 상기 최대 임계값을 초과한다는 결정에 응답하여, 후속 시간 간격에 대한 최대 임계값을 상기 메시지의 개수와 상기 제1시간 간격에 대한 상기 최대 임계값 간의 차에 대응하는 양만큼 감소시키는 단계를 더 포함하는, 방법.

청구항 13

제10항에 있어서, 상기 트래픽 정책의 위반을 검출하는 단계는,

상기 시간 간격의 세트 중 N개의 가장 최근 시간 간격의 윈도우에 걸쳐 상기 비신뢰 디바이스로부터의 메시지의 평균 개수를 계산하는 단계, 및

상기 평균 개수가 최대 임계값을 초과한다고 결정하는 단계를 포함하는, 방법.

청구항 14

제13항에 있어서, 상기 비신뢰 디바이스로부터의 트래픽을 미리 정해진 기간 동안 제한함으로써 상기 NoC의 트래픽을 감소시키는 단계를 더 포함하는, 방법.

청구항 15

제10항에 있어서,

상기 로컬 트래픽이 로컬 트래픽 임계값을 초과하는지 여부를 결정하는 단계; 및

상기 로컬 트래픽이 상기 로컬 트래픽 임계값을 초과하는 경우 상기 위반의 검출에 응답하여 상기 NoC의 트래픽을 감소시키는 단계를 더 포함하는, 방법.

청구항 16

시스템으로서,

비신뢰 디바이스;

메모리; 및

외부 프로세서 링크를 통해 상기 비신뢰 디바이스와 결합되고 상기 메모리와 결합된 호스트 SoC를 포함하되, 상기 외부 프로세서 링크는 상기 호스트 SoC에서 상기 비신뢰 디바이스로부터 메시지를 수신하도록 구성되고, 상기 수신된 메시지들은 상기 호스트 SoC의 메모리에 대한 하나 이상의 메모리 요청을 포함하며, 그리고 상기 호스트 SoC는, 상기 비신뢰 디바이스로부터의 메시지에 응답하여 상기 메모리로부터의 데이터를 상기 비신뢰 디바이스로 송신하도록 구성되고, 상기 호스트 SoC는,

상기 SoC의 내부 블록들 간에 로컬 트래픽을 송신하도록 구성된 NoC; 및

상기 외부 프로세서 링크와 결합된 트래픽 제어기를 포함하고, 상기 트래픽 제어기는,

하나 이상의 시간 간격의 세트에 걸쳐 상기 NoC를 통해 전송된 상기 비신뢰 디바이스로부터의 외부 트래픽의 양을 감시하고,

상기 시간 간격의 세트 중 제1시간 간격 동안 상기 비신뢰 디바이스로부터 상기 호스트 SoC로 송신되는 메시지의 개수가 상기 제1시간 간격에 대한 최대 임계값을 초과하는지 여부를 결정함으로써 상기 외부 트래픽의 양에 기초하여 트래픽 정책의 위반을 검출하고, 그리고

상기 위반의 검출에 응답하여, 상기 비신뢰 디바이스로부터의 메시지로 인해 발생하는 상기 NoC의 트래픽을 감소시키도록 구성된, 시스템.

청구항 17

제16항에 있어서, 상기 외부 프로세서 링크는, 상기 외부 프로세서 링크의 버퍼 용량에 기초하여 상기 비신뢰 디바이스로부터의 트래픽을 규제하도록 구성된 흐름 제어 로직을 더 포함하는, 시스템.

청구항 18

제16항에 있어서, 상기 트래픽 제어기는 상기 호스트 SoC와 함께 단일 집적 회로 상에 위치하는, 시스템.

청구항 19

제16항에 있어서, 상기 비신뢰 디바이스로부터의 메시지는, 상기 메모리에 대한, 판독 요청, 기입 요청, 및 어드레스 변환 요청을 포함하는, 시스템.

청구항 20

삭제

청구항 21

삭제

발명의 설명

기 술 분 야

[0001] 일부 근대 컴퓨터 시스템은, HyperTransport(HT)와 같은 통신 인터페이스를 통해 서로 통신할 수 있는, 중앙 처리 유닛(CPU), 그래픽 처리 유닛(GPU), 또는 기타 시스템 온 칩(systems on a chip: SoC)과 같은 다수의 처리 유닛을 이용한다. 일례로, HT는, 다수의 프로세서가 지점간 방식으로 서로 또는 다른 디바이스들과 통신할 수 있게 하는 고 대역폭 양방향 직렬/병렬 버스이다.

[0002] 그러나, 이러한 시스템에서는, 하나 이상의 SoC 또는 시스템의 다른 디바이스들 중 하나 이상의 (설계 불량, 버그, 또는 악의적 의도로 인해 발생하는지에 상관없는) 비협력적 거동이 SoC 또는 디바이스의 협력적 성능에 부정적인 영향을 미칠 수 있다. 예를 들어, 이러한 디바이스들은, 아무런 제약 없이 외부 인터페이스(예를 들어, HT)를 통해 과다한 메모리 요청 또는 다른 트래픽을 다른 SoC로 발행하도록 허용되면 서비스 거부(denial-of-service: DOS) 공격 및 기타 문제를 야기할 수 있다.

도면의 간단한 설명

[0003] 본 개시 내용은, 첨부 도면의 도면들을 한정하는 것이 아니라 예로서 예시되어 있다.

도 1은 컴퓨팅 시스템의 일 실시예를 도시한다.

도 2는 일 실시예에 따라 처리 유닛과 메모리를 포함하는 컴퓨팅 시스템의 일부분을 도시한다.

도 3은 트래픽 제어기 모듈의 일 실시예를 도시한다.

도 4는 일 실시예에 따라 트래픽 카운트 값들의 시간-가중 평균(time-weighted average)을 구현하기 위한 로직을 도시한다.

도 5는 일 실시예에 따라 비신뢰(untrusted) 외부 디바이스로부터의 트래픽을 조절(throttle)하기 위한 프로세스를 도시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0004] 다음에 따르는 설명은, 실시예들을 양호하게 이해하도록 특정 시스템, 구성요소, 방법 등의 예와 같은 다수의 특정 세부 사항을 설명하는 것이다. 그러나, 통상의 기술자에게는, 이들 특정 세부 사항 없이 적어도 일부 실시예를 실시할 수 있다는 점이 명백할 것이다. 다른 예에서, 공지된 구성요소 또는 방법은, 실시예들을 불필요하게 모호하게 하는 것을 피하도록 상세히 설명하지 않거나 간단한 블록도의 형태로 제시한다. 따라서, 특정 세부 사항은 단지 예시적인 것이다. 특정 구현예는, 이러한 예시적인 세부 사항과 다를 수 있으며, 여전히 실시예들의 범위 내에 있는 것으로 고려될 수 있다.

[0005] 일 실시예에서, 컴퓨팅 시스템의 호스트 프로세서 또는 기타 SoC는, 컴퓨팅 시스템의 다른 SoC, 디바이스, 및 모듈과의 통신을 가능하게 하도록 HyperTransport(HT) 등의 외부 인터페이스를 포함한다. 그러나, 이들 외부 디바이스의 거동은, 종종 보장될 수 없으며, 이들 외부 디바이스가 제삼자에 의해 제공되는 경우에 특히

그러하다. 따라서, 외부 디바이스들의 협력을 보장할 수 없는 경우, 호스트 SoC는, 성능 저하, DOS 공격, 및 비신뢰 디바이스 또는 비신뢰 디바이스에서 실행되는 소프트웨어가 어떠한 제약 없이도 소정의 액션을 수행할 수 있는 경우 발생할 수 있는 기타 문제점을 방지하도록, 이들 디바이스를 비신뢰 에이전트로 취급할 수 있다. 예를 들어, 외부 인터페이스를 통해 호스트 SoC에 과다한 개수의 메모리 요청을 발행하는 외부 디바이스는, 호스트 SoC의 NoC에 과부하를 걸 수 있으므로, 호스트 SoC의 내부 블록들 간에 로컬 트래픽이 서빙되는 것을 제한하는 DOS 상황이 발생할 수 있다.

[0006] 일 실시예에서, 외부 프로세서 링크 인터페이스는 흐름 제어 메커니즘(예를 들어, 크레딧 기반 흐름 제어)을 구현하지만, 이러한 메커니즘은, 로컬 하드웨어 구조 및 자원(예를 들어, 버퍼)이 과다가입(over-subscribed)되지 않거나 오버플로우하지 않음을 보장할 뿐이다. 크레딧 기반 흐름 제어 메커니즘의 경우, 일단 요청이 네트워크에서 앞으로 이동하였다면, 크레딧이 해제되고 더 많은 요청을 전송될 수 있다. 이러한 흐름 제어 메커니즘은, 트래픽의 단기 폭발이 하드웨어 자원을 압도하지 않음을 보장하지만, 반드시 전체 트래픽을 장기간 관리를 제공하지는 않는다.

[0007] 일 실시예에서, 호스트 SoC는, 비신뢰 외부 디바이스로부터의 트래픽을 감시하고 (메시지의 개수, 바이트 등에 의해 측정되는 바와 같은) 외부 디바이스로부터의 외부 트래픽의 양이 트래픽 정책을 위반함을 검출하는 경우 외부 디바이스로부터의 트래픽을 조절하는 트래픽 제어기 메커니즘을 구현한다. 일 실시예에서, 트래픽 제어기는, 하나 이상의 시간 간격의 세트에 걸쳐 비신뢰 디바이스로부터의 메시지 또는 바이트의 개수를 카운트하여 비신뢰 디바이스로부터의 트래픽 양이 정책을 위반하는지 여부를 결정한다. 정책을 위반한다면, 트래픽 제어기는 다수의 후속 시간 간격 동안 비신뢰 디바이스로부터 수용되는 트래픽의 양을 제한한다.

[0008] 도 1은, 전술한 바와 같이 비신뢰 디바이스에 대한 트래픽 제어 메커니즘을 구현하는 컴퓨팅 시스템(100)의 일 실시예를 도시한다. 일반적으로, 컴퓨팅 시스템(100)은, 랩톱 또는 데스크톱 컴퓨터, 이동 전화, 서버 등을 포함하지만 이에 한정되지 않는 다수의 상이한 유형의 디바이스 중 임의의 것으로서 구체화될 수 있다. 컴퓨팅 시스템(100)은 버스(101)를 통해 서로 통신하는 다수의 구성요소(102 내지 108)를 포함한다. 컴퓨팅 시스템(100)에서, 구성요소들(102 내지 108)의 각각은, 버스(101)를 통해 직접적으로 또는 나머지 구성요소들(102 내지 108) 중 하나 이상을 통해 그 나머지 구성요소들(102 내지 108) 중 임의의 것과 통신할 수 있다. 컴퓨팅 시스템(100)의 구성요소들(101 내지 108)은, 랩톱 또는 데스크톱 섀시 혹은 이동 전화 케이스와 같은 단일 물리적 케이스 내에 포함된다. 대체 실시예에서, 컴퓨팅 시스템(100)의 구성요소들 중 일부는, 전체 컴퓨팅 시스템(100)이 단일 물리적 케이스 내에 존재하지 않도록 주변 디바이스로서 구체화될 수 있다.

[0009] 컴퓨팅 시스템(100)은, 또한, 사용자로부터 정보를 수신하거나 사용자에게 정보를 제공하기 위한 사용자 인터페이스 디바이스를 포함한다. 특히, 컴퓨팅 시스템(100)은, 키보드, 마우스, 터치 스크린, 또는 사용자로부터 정보를 수신하기 위한 다른 디바이스 등의 입력 디바이스(102)를 포함한다. 컴퓨팅 시스템(100)은, 모니터, 발광 다이오드(LED) 디스플레이, 액정 디스플레이, 또는 다른 출력 디바이스와 같은 디스플레이(105)를 통해 정보를 사용자에게 표시한다.

[0010] 컴퓨팅 시스템(100)은, 또한, 유선 또는 무선 네트워크를 통해 데이터를 송신 및 수신하기 위한 네트워크 어댑터(107)를 포함한다. 컴퓨팅 시스템(100)은, 또한, 하나 이상의 주변 디바이스(108)를 포함한다. 주변 디바이스(108)는, 컴퓨팅 시스템(100)에 의해 사용되는, 대용량 저장 디바이스, 위치 검출 디바이스, 센서, 입력 디바이스, 또는 다른 유형의 디바이스를 포함할 수 있다.

[0011] 컴퓨팅 시스템(100)은, 메모리 서브시스템(106)에 저장된 명령어(106a)를 수신하고 실행하도록 구성된 처리 유닛(104)을 포함한다. 일 실시예에서, 처리 유닛(104)은 다수의 처리 요소를 포함하며, 각 처리 요소는, 그래픽 처리 유닛(GPU), 가속 처리 유닛(APU), 필드 프로그래머블 게이트 어레이(FPGA), 디지털 신호 프로세서(DSP), 또는 다른 임의의 주문형 집적 회로(ASIC) 또는 SoC일 수 있다.

[0012] 메모리 서브시스템(106)은, RAM 모듈, ROM 모듈, 하드 디스크, 및 다른 비일시적 컴퓨터 판독가능 매체와 같이 컴퓨팅 시스템(100)에 의해 사용되는 메모리 디바이스를 포함한다. 메모리 서브시스템(106)에 포함된 메모리는 컴퓨팅 시스템(100)의 메인 메모리로서 사용된다. 추가 유형의 메모리는, 메모리 서브시스템(106) 또는 컴퓨팅 시스템(100)의 다른 곳에 포함될 수 있다. 예를 들어, 캐시 메모리 및 레지스터도, 처리 유닛(104) 또는 컴퓨팅 시스템(100)의 다른 구성요소 상에 존재할 수 있다.

[0013] 도 2는, 메모리 서브시스템(106)에 포함된 메모리(230)와 함께 2-소켓 시스템으로서 구현된 처리 유닛(104)의 일 실시예를 도시한다. 처리 유닛(104)에서, 각각의 소켓(201, 202)은, SoC(예를 들어, 멀티코어 칩, APU 등)를

포함하고, 외부 프로세서 링크(240)를 통해 서로 접속된다. 일 실시예에서, 외부 프로세서 링크(240)를 구현하기 위한 인터페이스는, HyperTransport와 같은 개방형 표준, QuickPath Interconnect(QPI)와 같은 독점적 인터페이스, 또는 다른 일부 유사한 인터페이스일 수 있다.

[0014] 양측 소켓(201, 202)이 동일한 설계자로부터의 SoC를 포함하는 경우, SoC들은 링크(240)의 대역폭을 공평하게 공유하는 데 있어서 서로 협력하도록 설계될 수 있다. 그러나, 일 실시예에서, 2-소켓 시스템은 호스트 SoC(210)를 다른 설계자로부터의 비신뢰 외부 디바이스(220)와 접속한다. 이에 따라, 이용가능한 대역폭의 공정한 소비에 대한 디바이스(220)의 협력적 거동을 보장하지 못할 수 있다. 그러나, 비신뢰 디바이스(220)는, 여전히 링크(240)를 통해 메모리 요청(예를 들어, 메모리 판독 및 기입 요청) 및 다른 메시지 또는 트랜잭션(예를 들어, 메모리 변환 요청)을 발행할 수 있다.

[0015] 일 실시예에서, 호스트 SoC(210)는 메모리(230)에 접속된다. 메모리(230)는 메모리 서브시스템(106)의 일부이다. 일부 실시예에서, 메모리(230)는 입력 디바이스 또는 통신 디바이스와 같은 주변 디바이스의 메모리를 나타낼 수 있다. 메모리(230)는, 메모리(230)로 향하는, 호스트 SoC(210)를 통해 송신되는 메모리 요청 및 다른 메시지를 통해, 외부 디바이스(220)에 의해 액세스된다. 예를 들어, 외부 디바이스(220)는, 호스트 SoC(210)가 메모리(230)로부터의 판독 또는 메모리에 대한 기입을 행하게 하도록 호스트 SoC(210)에 메모리 요청을 발행한다. 외부 디바이스(220)로부터의 판독 요청에 대해, 호스트 SoC(210)는, 메모리(230)로부터 판독되는 데이터를 외부 디바이스(220)에 리턴함으로써 판독 요청에 응답한다. 외부 디바이스(220)는, 또한, 동일한 경로를 사용하여 메모리(230)에 대한 메모리 변환 요청을 수행하고, 여기서 호스트 SoC(210)는 변환된 메모리 어드레스를 외부 디바이스에 리턴함으로써 메모리 변환 요청에 응답한다.

[0016] 외부 디바이스(220)와 호스트 SoC(210) 간의 이러한 통신은, HT, QPI, 또는 다른 기술을 사용하여 구현되는 외부 프로세서 링크(240)를 통해 송신된다. 외부 프로세서 링크는, 또한, 외부 디바이스(220)로부터의 트래픽을 규제하기 위한 흐름 제어 로직(211)을 포함한다. 일 실시예에서, 호스트 SoC(210)는 외부 디바이스(220)로부터의 인입 메시지를 관리하기 위한 입력 버퍼 또는 다른 하드웨어 자원을 포함하고, 따라서, 흐름 제어 로직(211)은, 외부 디바이스(220)가 이들 하드웨어 자원의 용량을 초과하는 것을 방지하도록 다른 하드웨어 자원과 입력 버퍼의 용량에 기초하여 외부 디바이스(220)로부터의 메시지 트래픽을 규제한다. 일 실시예에서, 흐름 제어 로직(211)은, 호스트 SoC(210)에서 이용가능한 입력 버퍼의 개수를 나타내는 크레딧을 외부 디바이스(220)에 송신하는 크레딧 기반 흐름 제어 메커니즘을 구현한다. 이어서, 외부 디바이스(220)는 수신된 크레딧의 개수에 대응하는 메시지의 개수를 전송한다.

[0017] 호스트 SoC(210)는, 호스트 SoC(210)의 내부 블록들(예를 들어, 블록(214, 215)) 간에 로컬 트래픽을 송신하는 데 사용되는 온-칩 인터커넥트인 네트워크-온-칩(network on chip: NoC)(213)을 포함한다. 또한, NoC(213)는, 메모리(230)로 향하는 외부 디바이스(220)로부터의 메시지(예를 들어, 메모리 요청) 및 외부 디바이스(220)로 리턴되는 메모리(230)로부터의 데이터를 반송한다. 예를 들어, 메모리(230)로 향하는 외부 디바이스(220)로부터의 메시지는, 메모리(230)에서 수신되기 전에 외부 프로세서 링크(240), 흐름 제어 로직(211), 트래픽 제어기(212), 및 NoC(213)를 통해 송신된다. 메모리(230)로부터의 데이터(예를 들어, 판독 요청에 응답하여 판독된 데이터)는 동일한 경로를 통해 외부 디바이스(220)로 리턴된다.

[0018] 따라서, 외부 디바이스(220)가 너무 많은 메모리 요청을 발행하면, NoC(213)는 혼잡해질 수 있다. 그 결과, 호스트 SoC(210)의 내부 블록들(예를 들어, 블록들(214 내지 215))로부터의 로컬 트래픽은, NoC(213)에서의 혼잡 및/또는 메모리(230)에서의 큐잉 지연들로 인해 성능 저하를 겪을 수 있다.

[0019] 따라서, 일 실시예는 호스트 SoC(210)와 동일한 집적 회로 칩 상에 트래픽 제어기(212)를 포함할 수 있다. 트래픽 제어기(212)는, 외부 프로세서 링크(240)에 접속되고, 하나 이상의 시간 간격의 세트에 걸쳐 비신뢰 외부 디바이스(220)로부터 수신되는 메시지 또는 바이트의 개수를 감시하는 기능, 메시지의 개수에 기초하여 트래픽 정책의 위반을 검출하는 기능, 및 트래픽 정책이 위반된 경우 비신뢰 디바이스(220)로부터의 메시지로 인해 발생하는 NoC(213)의 트래픽을 감소시키는 기능을 포함하는 여러 기능을 수행한다.

[0020] 도 3은 트래픽 제어기(212)의 일 실시예를 도시한다. 트래픽 제어기(212)는, 외부 프로세서 링크(240)를 통해 외부 디바이스(220)로부터의 트래픽과 외부 디바이스로의 트래픽을 동적으로 감시하기 위한 감시 모듈(310), 관찰된 트래픽이 미리 정의된 트래픽 정책을 위반하는지 여부를 결정하기 위한 평가 모듈(320), 및 외부 디바이스(220)로부터 발생하는 과다한 트래픽에 의해 야기되는 바람직하지 않은 영향을 조절하거나 그 외에는 감소시키기 위한 조절 로직(330)을 포함한다.

[0021] 모듈들(310, 320, 330)의 동작에 의해, 트래픽 제어기(212)는, 비신뢰 외부 디바이스(220)로부터 호스트 SoC(210)로 송신되는 메시지(또는 바이트)의 개수가 다수의 연속 시간 간격의 각각에 대한 최대 임계값을 초과하는지 여부를 결정함으로써 소정의 트래픽 정책의 위반을 검출한다.

[0022] 감시 모듈(310)은 외부 프로세서 링크(240)를 통해 외부 디바이스(220)로부터의 트래픽과 외부 디바이스로의 트래픽을 감시한다. 일 실시예에서, 감시 모듈(310)은 외부 디바이스(220)에 의해 전송되는 메시지의 개수를 카운트하고, 대안으로, 감시 모듈(310)은 전송되는 바이트의 개수와 같은 다른 일부 메트릭을 추적할 수 있다. 이에 따라, 카운터(311)는, 연속적 시간 간격들의 각각에 대해 외부 프로세서 링크(240)를 통해 외부 디바이스(220)에 의해 전송되는 메시지 또는 바이트의 개수를 카운트한다.

[0023] 감시 모듈(310)은 시간 간격을 정의하기 위한 타이머(313)를 포함한다. 각 간격의 시작에서, 타이머(313)는 카운터(311)를 제로로 리셋한다. 시간 간격이 경과함에 따라, 카운터(311)는 외부 디바이스(220)로부터의 트래픽과 외부 디바이스로의 트래픽을 감시하고, 예를 들어, 카운터(311)는 외부 프로세서 링크(240)에서 외부 디바이스로부터 수신되는 각각의 메시지에 대해 증분될 수 있다. 대체 실시예에서, 카운터(311)는, 외부 디바이스로 송신되는 메시지와 외부 디바이스로부터 수신되는 메시지 모두에 대해 증분될 수 있다.

[0024] (레지스터들(312-1, 312-2,...312-N을 포함하는) 카운트 레지스터들(312)은 순환 시프트 버퍼로서 논리적으로 동작하는 시프트 레지스터들이며, 각각의 레지스터는 시간 간격들 중 하나의 시간 간격에 대한 카운트 값을 저장하는 데 사용된다. 따라서, 타이머(313)가 시간 간격이 경과되었음을 나타내는 경우, 현재 시간 간격에 대하여 관찰된 메시지 또는 바이트의 개수를 나타내는 카운터(311)의 카운트 값이 카운트 레지스터들(312) 중 현재 선택된 카운트 레지스터에 저장된다.

[0025] 현재 카운트 값이 현재 선택된 레지스터(예컨대, 레지스터(312-1))에 저장된 후, 타이머(313)는 카운트 레지스터들(312)의 세트에서 다음 레지스터(예컨대, 레지스터(312-2))를 선택한다. 마지막 레지스터(312-N)가 선택된 후, 시간 간격의 경과로 인해 제1레지스터(312-1)가 다시 선택되어 겹쳐쓰기된다. 따라서, 카운트 레지스터들(312)의 각각은 N개의 가장 최근 시간 간격의 세트 중 하나의 시간 간격에 대한 카운트 값을 저장한다.

[0026] 평가 모듈(320)은, 레지스터들(312)의 카운트 값들에 기초하여 합, 평균, 또는 다른 메트릭을 계산하는 가산기 로직(321)을 포함한다. 이어서, 가산기 로직(321)의 출력은 대응하는 시간 간격에 대한 임계값과 비교된다. 일 실시예에서, 가산기 로직(321)은, 레지스터들(312) 중 하나(예를 들어, 가장 최근에 경과된 시간 간격에 대응하는 레지스터)를 간단히 선택하여 그 간격에 대한 임계값과 비교할 수 있다.

[0027] 대안으로, 가산기 로직(321)은, N개의 가장 최근 시간 간격의 슬라이딩 윈도우에 걸쳐 비신뢰 외부 디바이스(220)로부터의 메시지의 평균 개수를 계산하도록 구성될 수 있다. 예를 들어, 가산기 로직(321)은, 타이머(313)에 의해 트리거되어 연속적 시간 간격들의 각각의 만료시 이러한 시간-기반 이동 평균을 계산할 수 있다. 따라서, 각 시간 간격에 대해, 가산기 로직(321)은 레지스터들(312)에 현재 저장되어 있는 N개의 카운트 값의 평균을 계산한다. N이 2의 거듭제곱인 구현예의 경우, 단순히 레지스터들의 합의 $\log_2(N)$ 최하위 비트를 드롭(drop)함으로써 N에 의한 나눗셈이 효율적으로 달성되므로, 평균을 연산하는 것이 간단하다.

[0028] 일부 상황에서는, 먼 과거가 더 최근의 행동보다 덜 중요하며, 이에 따라, 이전 시간 간격에 대해 측정된 카운트 값에 가중치를 적용하여 전체 합에 대한 입력 값을 감소시킨다. 일 구성에서, 가산기 로직(321)은 레지스터들(312)의 카운트 값의 시간-가중 평균을 계산한다. N개의 가장 최근 카운트 값이 x_0, $x_{1...}x_{N-1}$로 표현된다면, 비가중 이동 평균은 다음과 같이 표현될 수 있다.

수학식 1

[0029] $$\left(\sum_{i=0}^{N-1} x_i\right)/N$$

[0030] 지수 가중된 이동 평균은 다음에 따르는 식으로 표현되며, 여기서 α는 0보다 크고 1보다 작은 값이다.

수학식 2

[0031] $$\left(\sum_{i=0}^{N-1} \alpha^i \times x_i\right)/N$$

[0032] α가 2의 거듭제곱(예컨대, 1/2)인 경우, 하드웨어 오버헤드는 N개의 레지스터로부터 단지 2개의 레지스터로 감소될 수 있다. 도 4는, 일 실시예에 따라 α=1/2인 시간-가중 평균을 구현하는 가산기 로직(321)과 2개의 레지스터(312-1 및 312-2)를 도시한다. 레지스터(312-1)는 N-1개의 이전 간격의 시간-가중 합을 저장하고, 레지스터(312-2)는 카운터(311)에 의해 결정된 바와 같이 가장 최근 (N번째) 시간 간격에 대한 카운트 값을 저장한다. 현재 시간 간격의 종료시, 레지스터(312-1)의 값은 비트 시프터(401)에 의해 2로 나누어지고 (가산기(402)를 통해) 레지스터(312-2)로부터의 카운트 값에 가산된다. 가산기(402)로부터의 새로운 시간-가중 합은 레지스터(312-1)에 다시 저장된다. 이어서, 레지스터(312-2)는 제로로 리셋되어 다음 간격에 대한 카운팅을 시작한다. 이어서, 레지스터(312-1)의 새로운 시간-가중 합을 N으로 나눔으로써 새로운 시간-가중 평균이 계산된다.

[0033] 도 3을 다시 참조하면, 평가 모듈(320)은, N개의 시간 간격의 각각에 대한 최대 임계값을 저장하는 구성 레지스터(322)에 접속된 입력을 갖는 비교 로직(324)을 포함한다. 대체 실시예에서, 구성 레지스터(322)는 N개의 시간 간격 모두에 대해 하나의 임계값을 저장한다.

[0034] 주어진 시간 간격에 대한 최대 임계값은, 트래픽 정책을 위반하지 않으면서 그 시간 간격 동안 외부 프로세서 링크(240)를 통해 비신뢰 외부 디바이스(220)로부터 수신될 수 있는 메시지의 최대 개수 또는 바이트의 최대 개수를 나타낸다. 대체 실시예에서, 최대 임계값은, 또한, 호스트 SoC(210)로부터 외부 디바이스(220)로 송신되는 메시지 또는 바이트의 개수에 적용되며, 또는 외부 디바이스(220)로 송신되고 외부 디바이스로부터 수신되는 메시지 또는 바이트의 개수에 적용될 수 있다.

[0035] 트래픽 제어기(212)에서, 최대 임계값은, 소프트웨어에 의해 프로그래밍가능하다(예를 들어, 펌웨어에 의해 설정되거나 컴퓨팅 시스템(100) 상에서 실행되는 운영 체제에 의해 특정된다). 대체 실시예에서, 임계값은 제조시 프로그래밍될 수 있고 또는 하드웨어 내장된 값(hard-wired value)일 수 있다. 구성 레지스터(322)의 임계값은, 레지스터(322)에 저장된 값을 구성하기 위한 구성 로직을 포함하는 구성 모듈(326)을 통해 프로그래밍가능하다. 구성 모듈(326)은, 하나 이상의 새로운 최대 임계값을 나타내는 재구성 요청(325)의 수신에 응답하여 그 하나 이상의 새로운 임계값을 구성 레지스터(322)에 저장한다.

[0036] 재구성 요청(325)은, 호스트 SoC(210)에 의해 발행되거나, 대안으로 컴퓨팅 시스템(100) 내의 다른 SoC 또는 다른 SoC에서 실행되는 프로세스와 같은 외부 디바이스 또는 프로세스에 의해 발행될 수 있다. 일 실시예에서, 임계값은 신뢰받는 디바이스 또는 소프트웨어에 의해서만 프로그래밍가능하다. 이에 따라, 구성 모듈(326)은, 요청(325)이 신뢰받는 엔티티로부터 온 것임을 보장하도록 요청(325)을 인증하고, 성공적인 인증에만 응답하여 구성 레지스터(322)를 업데이트한다. 이에 따라, 구성 모듈(326)은 요청(325)의 인증 실패에 응답하여 요청(325)을 폐기하거나 무시한다.

[0037] 신뢰받는 소프트웨어의 예로는, 펌웨어, 운영 체제, 또는 다른 신뢰받는 시스템에 의해 암호로 서명되고 인증된 기타 소프트웨어가 있다. 일 실시예에서, 시간 간격의 지속 기간과 같은 트래픽 제어기(212)의 다른 파라미터들도, 유사한 요청 및 인증 메커니즘에 의해 구성된다.

[0038] 비교 로직(324)은, (합, 평균, 시간-가중 평균, 또는 레지스터(312)로부터 계산된 다른 메트릭을 나타내는) 가산기 로직(321)의 출력을, 구성 레지스터(322)에 의해 제공되는 바와 같이 가장 최근에 완료된 시간 간격에 대응하는 최대 임계값과 비교한다. 비교 로직(324)은, 가산기 로직(321) 출력이 최대 임계값을 초과하면 그 출력을 어서트(assert)한다.

[0039] 비교 로직(324)의 출력은 조절 로직(330)에 접속된다. 일 실시예에서, 비교 로직(324)의 출력은, 임계값을 초과하는 메시지 또는 바이트의 개수를 조절 로직(330)에 나타낸다. 따라서, 비교 로직(324)의 출력이 비신뢰 외부 디바이스(220)로의 과다한 트래픽 및/또는 비신뢰 외부 디바이스로부터의 과다한 트래픽에 의해 야기되는 트래픽 정책의 위반을 나타내는 경우, 조절 로직(330)은 외부 디바이스(220)로의 트래픽 및/또는 외부 디바이스로부터의 트래픽을 소정의 기간 동안 제한한다.

[0040] 외부 디바이스(220)의 트래픽을 제한하는 하나의 방법은, 외부 장치(220)로부터의 트래픽을 미리 정의된 저 레

벨로 조절하여, 외부 장치(220)로부터의 몇 개의 메시지만이 NoC(213)에 액세스할 수 있게 하는 것이다. 이는 외부 디바이스(220)가 계속해서 진행되는 것을 가능하게 하지만, 호스트 프로세서(SoC)로/로부터의 외부 프로세서 링크(240) 인터페이스를 사용하는 외부 디바이스의 능력이 감소된다. 조절의 정확한 레벨은, 신뢰받는 디바이스 또는 소프트웨어의 인증된 재구성 요청에 의해서도 구성될 수 있다.

[0041] 일부 실시예에서, 조절 레벨은, 현재 간격이 경과할 때까지 어떠한 메시지도 외부 디바이스(220)로 송신되지 않고 또는 외부 디바이스로부터의 수신되지 않도록 설정될 수 있다. 특히, 이 방법은, 외부 디바이스의 장기간 메시지 부재(starvation)를 피하도록 트래픽 제어기(212)가 비교적 짧은 간격 길이를 사용할 때 사용될 수 있다. 크레딧-기반 흐름-제어 메커니즘이 사용되는 일 실시예에서, 호스트 SoC(210)는, 현재 시간 간격이 경과할 때까지 외부 디바이스(220)에 리턴되는 크레딧의 개수를 제한함으로써 외부 디바이스(220)로부터의 트래픽을 조절한다.

[0042] 일 실시예에서, 외부 디바이스(220)는, 현재 간격의 나머지 동안 단순히 조절되는 것이 아니라 다음 M개의 간격 동안 조절된다. 일 실시예에서, 조절 레벨은, M개 간격에 걸쳐 점진적으로 완화되고 덜 제한적으로 되어, 외부 디바이스(220)로부터의 트래픽이 결국 정상적인 비제한 레벨로 리턴하게 할 수 있다. 다른 파라미터와 유사하게, M 값과 완화 속도는, 신뢰받는 디바이스 또는 프로세스로부터의 인증된 재구성 요청에 의해 구성될 수 있다.

[0043] 일 실시예에서, 조절 로직(330)은, 현재 시간 간격에 대한 최대 임계값과 카운트 값 사이의 차에 대응하는 양만큼 다음 후속 시간 간격에 대한 최대 임계값을 감소시킴으로써 외부 디바이스(220)로부터의 트래픽을 조절한다. 일 실시예에서, 조절 로직(330)은 다음 시간 간격에 대응하는 임계값을 계산된 차만큼 감소시킨다. 대안으로, 동일한 효과를 달성하기 위한 다른 방법은, 현재 시간 간격의 종료시, 카운터를 제로로 리셋하는 대신 임계값만큼 카운터(311)를 감분하는 것이다. 예를 들어 카운트 값 12가 임계값 10을 초과하여, 위반이 발생한다. 따라서, 카운트 값 12는 임계값 10만큼 감분되어, 다음 간격 동안 초기 카운트 값인 2를 남기게 된다. 다음 간격에서, 외부 디바이스(220)는 임계값이 초과되기 전에 8개의 추가 메시지를 전할 수 있다.

[0044] 실제로, 트래픽 임계값을 초과하는 외부 디바이스(220)는, 자신의 카운터가 0보다 큰 값에서 시작하기 때문에 다음 간격에서 더욱 빨리 임계값에 도달할 것이다. 따라서, 현재 시간 간격에서 트래픽 정책을 위반하면, 다음 후속 시간 간격에 대한 임계값이 효과적으로 감소되어, 외부 디바이스가 다음 간격으로부터 트래픽 크레딧을 "차용"할 수 있다. 이에 따라, 부채는 다음 간격의 시작시 상환된다.

[0045] 일 실시예에서, 트래픽 제어기(212)는, 트래픽 정책을 위반한 후에도 외부 디바이스(220)가 호스트 SoC(210)와 계속 통신할 수 있게 하지만, 임계값에 도달한 후에 수용된 트래픽의 양을 (메시지, 바이트 등의 개수로) 카운팅한다. 임계값을 초과하는 이러한 양은, 후속 시간 간격들 동안 외부 디바이스가 상환하는 부채로서 기록된다.

[0046] 후속 시간 간격 동안 NoC(213) 또는 메모리(230)가 혼잡해지면, 트래픽 제어기(212)는 외부 디바이스(220)로부터의 트래픽을 조절한다. 외부 디바이스(220)로부터의 메시지가 조절로 인해 호스트 SoC(210)에 의해 지연되는 각각의 시간 간격에 대해, 외부 디바이스(220)의 부채는 구성가능한 양 R만큼 감소된다.

[0047] 트래픽 제어기(212)는, 또한, 외부 디바이스(220)가 회복불가능한 부채로 들어서는 것을 방지하도록 시간-기반 부채 면제를 구현한다. 일 실시예에서, 트래픽 제어기(212)는 매 C 시간 간격마다 현재 부채를 인수 β만큼 감소시킨다. 예를 들어, β가 0.5이고 C가 10일 때, 부채는 매 10번째 간격마다 반으로 줄어든다.

[0048] 외부 프로세서 링크(240)를 통한 외부 디바이스(220)로의 트래픽과 외부 디바이스로부터의 트래픽을 감시하는 것 이외에도, 트래픽 제어기(212)는, 또한, 로컬 트래픽으로 인해 NoC(213)의 트래픽 레벨을 감시하는 로컬 트래픽 모니터(340)를 포함한다. 감시된 로컬 트래픽은, 호스트 SoC(210)의 내부 블록들(예를 들어, 내부 블록들(214 및 215)) 간의 트래픽, 호스트 SoC(210)의 연산 유닛들(예를 들어, 중앙 처리 유닛들 또는 그래픽 처리 유닛들)로부터 NoC(213)로의 트래픽, 호스트 SoC(210)의 메모리 시스템으로의 트래픽 또는 호스트 SoC의 메모리 시스템으로부터의 트래픽, 또는 다른 로컬 트래픽 메트릭을 포함한다.

[0049] 로컬 트래픽 모니터(340)는, 로컬 트래픽 레벨에 관한 정보를 조절 로직(330)에 제공하여, NoC(213)가 외부 디바이스(220)로부터의 트래픽과 로컬 트래픽을 서빙하기 위한 충분한 대역폭을 갖는 시간 동안 조절 로직(330)이 외부 디바이스(220)로부터의 트래픽을 불필요하게 조절하는 것을 피할 수 있게 한다. 예를 들어, 호스트 SoC(210)가 대체로 유휴 상태에 있는 경우(예를 들어, 외부 디바이스(220)에 대한 작업 부하를 덜어준 경우), 이 메커니즘은, 외부 디바이스(220)가 가능한 최대 속도로 트래픽을 호스트 SoC(210)에 다시 주입하여 외부 디바이스(220)가 자신의 성능을 최대화할 수 있게 한다.

[0050] 다시 말하면, 호스트 SoC(210)의 로컬 트래픽 요구가 충분히 작으면, 트래픽 제어기(212)는, 외부 디바이스(220)가 다른 경우에 정상적인 간격당(또는 이동 평균) 임계값을 초과하였더라도 외부 디바이스(220)가 빠른 속도로 요청 발행을 계속할 수 있게 한다.

[0051] 일 실시예에서, 로컬 트래픽 모니터(340)는, 각 시간 간격 동안 호스트 SoC(210)로부터 시작되는, NoC(213)를 통해 송신되는 메시지 및/또는 바이트의 개수를 카운팅하기 위한 하나 이상의 카운터를 포함한다. 이러한 로컬 트래픽 카운트 값에 기초하여, 로컬 트래픽 모니터(340)는 이어서 NoC(213)의 로컬 트래픽이 로컬 트래픽 임계값을 초과하는지 여부를 결정한다. 로컬 트래픽이 로컬 트래픽 임계값을 초과하는 경우, 로컬 트래픽 모니터는, 외부 디바이스(220)가 트래픽 정책을 위반할 때 조절 로직(330)이 외부 디바이스(220)로 인한 트래픽을 감소시킬 수 있게 한다.

[0052] 전술한 바와 같이, 트래픽 제어기(212)는, 호스트 SoC(210)와는 별도의 집적 회로 패키지에 구현되는 외부 디바이스(220)로의 트래픽과 외부 디바이스로부터의 트래픽을 규제한다. 대체 실시예에서, 트래픽 제어기(212)에 구현된 메커니즘은, 호스트 시스템과 동일한 집적 회로 패키지 내의 비신뢰 블록으로의 트래픽과 이러한 비신뢰 블록으로부터의 트래픽을 규제하는 데 사용된다. 예를 들어, 일부 SoC 설계에서는, 제3자 설계자로부터의 비신뢰 블록들의 협력적 행동을 보장할 수 없는 이러한 비신뢰 블록들을 통합한다.

[0053] 이러한 경우에, 트래픽 제어기(212)에 의해 구현된 동일한 감시 및 조절 메커니즘은, 비신뢰 내부 블록을 NoC(213)에 접속하는 온-칩 상호접속부에 적용되어, 비신뢰 블록이 과다한 메시징으로 인한 (DOS형 행동과 같은) 성능 문제를 야기하지 않음을 보장한다.

[0054] 대체 실시예에서, 다수의 칩은 멀티-칩 모듈(MCM) 기술 및/또는 실리콘 인터포저-기반 통합(2.5D 스태킹)을 사용하여 함께 패키징되고, 여기서 하나 이상의 칩은 제3자 설계자로부터 공급될 수 있다. 이러한 패키징은, 트래픽 제어기(212)의 트래픽 제어 메커니즘이 적용되는 칩들(예를 들어, HT, AXI(Advanced eXtensible Interface) 등) 간에 소정 형태의 상호접속부를 제공할 것이다.

[0055] 도 5는 일 실시예에 따라 비신뢰 외부 디바이스로부터의 트래픽을 조절하기 위한 프로세스(500)를 나타내는 흐름도이다. 프로세스(500)의 동작은 트래픽 제어기(212) 및 컴퓨팅 시스템(100)의 다른 구성요소에 의해 수행된다.

[0056] 프로세스(500)는 블록(501)에서 시작한다. 블록(501)에서, 호스트 SoC(210)의 내부 블록들(예를 들어, 블록들(214 및 215))은 로컬 트래픽을 NoC(213)를 통해 서로에 송신함으로써 서로 통신한다. 또한, NoC(213)는, 외부 디바이스(220)와 메모리(230) 간에, 또는 외부 디바이스(220)와 호스트 SoC(210)의 내부 블록 간에 메시지를 반송하는 데 사용된다. 블록(501)으로부터, 프로세스(500)는 블록(511)에서 계속된다.

[0057] 블록(511)에서, 트래픽 제어기(212)는 재구성 요청(325)이 구성 모듈(326)에서 수신되었는지 여부를 결정한다. 재구성 요청(325)은, 호스트 SoC(210), 호스트 SoC(210)에 접속된 외부 디바이스, 또는 다른 일부 디바이스나 프로세스로부터 발생한다. 재구성 요청(325)은, 로컬 트래픽 임계값, 외부 디바이스(220)에 대한 최대 트래픽 임계값, 트래픽 정책 위반 후에 외부 디바이스(220)를 조절하기 위한 간격의 개수(M) 등과 같이 트래픽 제어기(212)의 하나 이상의 동작 파라미터에 대한 새로운 값을 특정한다. 재구성 요청(325)은, 또한 트래픽 제어기(212)에 의해 구현되는 조절 방법(예를 들어, M개 간격에 대한 조절, 부채 발생, 및 상환 등)을 선택하는 데 사용된다.

[0058] 블록(513)에서, 구성 모듈(326)은, 트래픽 제어기(212)를 재구성하도록 허용된 신뢰받는 디바이스로부터의 요청(325)인지 여부를 결정하도록 재구성 요청(325)을 인증한다. 인증이 성공적이면, 구성 모듈(326)은 요청(325)에 따라 구성을 변경한다(블록 515). 예를 들어, 구성 모듈(326)이 외부 디바이스(220)로부터의 트래픽에 대한 최대 임계값의 변경을 요청하는 재구성 요청(325)을 성공적으로 인증하면, 구성 모듈(326)은, 요청(325)을 성공적으로 인증한 후에 레지스터(322)에 저장된 임계값을 요청(325)에 표시된 새로운 값으로 업데이트한다. 블록(513)에서, 재구성 요청(325)이 성공적으로 인증되지 않으면, 구성 모듈(326)은 요청(517)을 무시하고 레지스터(322)를 업데이트하지 않는다. 블록(515) 또는 블록(517)으로부터, 프로세스(500)는 블록(521)에서 계속된다.

[0059] 블록(521)에서, 로컬 트래픽 모니터(340)는, NoC(213)의 로컬 트래픽(즉, 호스트 SoC(210)의 내부 블록들 간의 트래픽)이 로컬 트래픽 임계값을 초과하는지 여부를 결정한다. 로컬 트래픽 모니터는 로컬 메시지의 개수, 또는 대안으로 전송된 바이트의 개수를 카운팅하고, 이 카운트 값이 로컬 트래픽 임계값을 초과하지 않으면, 블록(523)에서 조절 로직이 디스에이블된다. 로컬 트래픽 요구가 작으므로, NoC(213)는, 외부 디바이스(220)로부터의 과다한 트래픽을 감시하고 잠재적으로 조절할 필요없이, 외부 디바이스(220)로부터의 트래픽을 서빙하기 위

한 충분한 대역폭을 갖는다. 따라서, NoC(213)를 통한 외부 디바이스(220)로부터의 메시지의 송신은, 조절이 디스에이블된 상태에서 블록(501)에서 계속된다.

[0060] 블록(521)에서, 로컬 트래픽 카운트 값이 로컬 트래픽 임계값을 초과하면, 로컬 트래픽 모니터(340)는 블록(531)에서 조절 로직(330)을 인에이블한다. 블록(533)에서, 조절 로직(330)이 인에이블된 상태에서 외부 디바이스(220)로부터의 트래픽과 로컬 트래픽의 전송이 계속된다. 조절 로직(330)이 인에이블되면, 트래픽 제어기(212)는, 외부 디바이스(220)로부터의 트래픽을 감시하고, 트래픽 정책의 위반을 검출하는 것에 응답하여 외부 디바이스(220)로 인한 NoC(213)의 트래픽 양을 감소시킨다. 블록(533)으로부터, 프로세스(500)는 블록(535)에서 계속된다.

[0061] 블록(535)에서, 외부 프로세서 링크(240)를 통해 외부 디바이스(220)로부터 호스트 SoC(210)에서 메시지가 수신되지 않았다면, 프로세스(500)는 계속해서 블록(521)으로 되돌아간다. 블록(535)에서, 하나 이상의 메시지가 외부 디바이스(220)로부터 온 것이면, 카운터(311)는 수신되는 메시지(또는 바이트)의 개수만큼 증분된다(블록537). 블록(537)으로부터, 프로세스(500)는 블록(539)에서 계속된다.

[0062] 블록(539)에서, 현재 시간 간격의 종료에 도달하지 않았다면, 프로세스(500)는 계속해서 블록(533)으로 되돌아가고, 따라서, 현재 시간 간격의 종료까지 블록들(533 내지 539)이 반복된다. 블록들(533 내지 539)을 반복함으로써, 트래픽 제어기(212)는 현재 시간 간격 동안 비신뢰 외부 디바이스(220)로부터 수신되는 메시지의 개수를 감시한다. 타이머(313)에 의해 표시된 바와 같이, 시간 간격의 종료시, 프로세스(500)는 블록(539)으로부터 블록(541)으로 계속된다.

[0063] 블록(541)에서, 트래픽 제어기(212)의 다양한 구성요소는 최근 경과된 시간 간격에 대한 자신의 계산을 업데이트한다. 예를 들어, 가산기 로직(321)은, N개의 가장 최근의 시간 간격의 윈도우에 대해 비신뢰 외부 디바이스(220)로부터의 메시지들의 합 또는 평균 개수를 계산한다. 사용되는 조절 방법에 따라 경과된 각 시간 간격에 대하여 다른 계산이 수행되며, 예를 들어, 간격에 대한 최종 카운트 값은 레지스터들(312) 중 하나에 저장되고, N개의 시간 간격에 대한 카운트 값들의 합 또는 평균은 가산기 로직(321)에서 계산되고, 과거 위반에 대한 잔여 부채 및 잔여 조절 시간 등이 계산된다. 블록(541)으로부터, 프로세스(500)는 블록(543)에서 계속된다.

[0064] 블록(543)에서, 비교 로직(324)은, N개 간격에 대하여 카운트 값들의 계산된 합 또는 평균에 기초하여 트래픽 간격의 위반이 발생했는지 여부를 검출한다. 비교 로직(324)은, 구성 레지스터(322)에 의해 제공되는 바와 같이 경과된 시간 간격에 대응하는 최대 임계값에 대한 합 또는 평균을 비교한다. 최대 임계값이 초과되면, 트래픽 정책의 위반이 발생하였으며, 프로세스(500)는 블록(545)에서 계속된다.

[0065] 블록(545)에서, 트래픽 제어기(212)는 비신뢰 외부 디바이스(220)로부터의 트래픽의 조절을 시작함으로써 위반에 응답한다. 조절은, 소정의 기간(예를 들어, M개 간격) 동안 비신뢰 외부 디바이스(220)로부터의 메시지로 인한 NoC(213)의 트래픽을 감소시킨다. 조절 로직(330)에 의해 수행되는 특정 동작은 사용되는 조절 메커니즘에 의존하며, 일 실시예에서, 조절 메커니즘은 구성 레지스터(322)에 특정된다. 전술한 바와 같이, 하나의 조절 메커니즘은, 간격에 대해 카운팅된 메시지(또는 바이트)의 개수와 시간 간격에 대한 최대 임계값 사이의 차에 대응하는 양만큼 다음의 후속 시간 간격에 대한 최대 임계값을 감소시킨다. 이 방식에 따르면, 조절 로직(330)은 카운터(311)를 제로로 리셋하는 대신 카운터(311)를 임계값만큼 감분한다. 전술한 다른 조절 메커니즘은 부채 발생 및 상환 메커니즘이다. 이러한 방식에 따르면, 조절 로직(330)은, 블록(545)에서 트래픽 정책 위반 및 경과 시간에 각각 응답하여 부채를 발생시키고 감소시키기 시작한다. 대체 구성과 실시예에서는, 외부 디바이스(220)로부터의 메시지로 인한 NoC(213)의 트래픽을 감소시키기 위한 다른 동작들이 블록(545)에서 수행된다.

[0066] 블록(543)에서, 가장 최근에 경과된 시간 간격에 대한 최대 임계값이 초과되지 않았다면, 프로세스(500)는 블록(547)에서 계속된다. 블록(547)에서, 조절 로직은, 조절을 종료할지 여부를 결정하도록 (사용되는 조절 메커니즘에 의존하는) 다수의 조건 중 임의의 조건을 평가한다. 예를 들어, 트래픽 정책 위반이 발생한 후에 M개의 시간 간격에 대하여 외부 디바이스(220)로부터의 트래픽을 제한하는 조절 방법을 구현하는 경우, 조절 로직(330)은 M개의 간격이 경과하였다면 조절을 종료한다. 부채 발생 및 회수 메커니즘을 포함하는 조절 방법을 구현할 때, 조절 로직(330)은, 외부 디바이스(220)에 연관된 부채가 상환되었다면 조절을 종료한다. 이에 따라, 다른 조절 방법의 경우, 조절이 다른 기준에 기초하여 종료된다.

[0067] 조절을 종료하기 위한 기준이 충족되지 않으면, 조절은 블록(545)에서 계속된다. 블록(545)으로부터, 프로세스(500)는, 외부 디바이스(220)로부터의 트래픽과 로컬 트래픽의 송신이 사실상 조절과 함께 계속되는 블록(501)에서 더 계속된다. 블록(547)에서 기준이 충족되면, 블록(549)에서 조절이 종료되고, 프로세스(500)는, 외부 디

바이스(220)의 조절 없이 외부 디바이스(220)로부터의 트래픽과 로컬 트래픽의 송신이 계속되는 블록(501)에서 더 계속된다.

[0068] 따라서, 프로세스(500)는, 외부 트래픽의 감시 및 조절 없이 비신뢰 외부 디바이스(220)로부터의 외부 트래픽과 로컬 트래픽을 서빙하도록 로컬 트래픽이 작을 때(즉, 로커 트래픽 임계값 미만일 때) 블록들(501, 511, 521, 523)을 루핑(loop)한다. 로컬 트래픽이 클 때(즉, 로컬 트래픽 임계값을 초과할 때), 프로세스(500)는, 블록들(501, 511, 521, 및 531 내지 549)을 루핑하여, 하나 이상의 시간 간격의 세트에 걸쳐 외부 디바이스(220)로부터의 외부 트래픽을 감시하고 외부 트래픽에 의한 트래픽 정책의 위반 검출에 응답하여 NoC(213)의 트래픽을 감소시킨다.

[0069] 따라서, 전술한 바와 같은 처리 유닛은, SoC의 내부 블록들 간에 로컬 트래픽을 송신하도록 구성된 네트워크 온 칩(NoC)을 포함하는 호스트 시스템-온-칩(SoC), 호스트 SoC에서 비신뢰 디바이스로부터의 메시지를 수신하도록 구성된 외부 프로세서 링크, 및 외부 프로세서 링크에 접속된 트래픽 제어기를 포함한다. 트래픽 제어기는, 시간 간격들의 세트에 걸쳐 비신뢰 디바이스로부터의 외부 트래픽의 양을 감시하고, 외부 트래픽의 양에 기초하여 트래픽 정책의 위반을 검출하고, 위반 검출에 응답하여 비신뢰 디바이스로부터의 메시지로 인해 발생하는 NoC의 트래픽을 감소시키도록 구성된다.

[0070] 특히, 트래픽 제어기는, 주어진 시간 간격 동안 비신뢰 디바이스로부터 호스트 SoC로 송신되는 메시지의 개수가 그 시간 간격에 대한 최대 임계값을 초과하는지 여부를 결정함으로써 트래픽 정책의 위반을 검출한다. 트래픽 제어기는, 외부 프로세서 링크를 통해 비신뢰 디바이스로부터 수신되는 각각의 메시지에 응답하여 증분되고 주어진 시간 간격의 경과에 응답하여 리셋되는 메시지 카운터를 포함한다.

[0071] 트래픽 제어기는, 최대 임계값을 변경하기 위한 요청을 수신할 수 있는 구성 로직을 더 포함한다. 구성 로직은, 요청 인증에 응답하여 요청에 따라 최대 임계값을 변경하고, 또는 요청 인증 실패에 응답하여 요청을 무시한다.

[0072] 트래픽 제어기는, 또한, 비신뢰 디바이스로부터 호스트 SoC로 송신되는 메시지의 개수가 주어진 시간 간격에 대한 최대 임계값을 초과하는 것으로 결정한 것에 응답하여, 후속 시간 간격에 대한 최대 임계값을 감소시키도록 구성된다. 트래픽 제어기는, 주어진 시간 간격 동안 수신되는 메시지의 개수와 그 주어진 시간 간격에 대한 최대 임계값 사이의 차에 대응하는 양만큼 후속 시간 간격에 대한 최대 임계값을 감소시킨다.

[0073] 트래픽 제어기는, N개의 가장 최근의 시간 간격의 윈도우에 걸쳐 비신뢰 디바이스로부터의 메시지의 평균 개수를 계산하고 평균 수가 최대 임계값을 초과한다고 결정함으로써, 트래픽 정책의 위반을 검출한다.

[0074] 트래픽 제어기는 시프트 버퍼에 다수의 레지스터를 더 포함한다. 레지스터들의 각각은, 해당 레지스터에 연관된 N개의 가장 최근의 시간 간격 중 하나의 시간 간격에 대한 메시지의 개수를 저장한다. 트래픽 제어기는, 또한, 레지스터에 저장된 개수를 합산하는 가산기를 포함한다.

[0075] 트래픽 제어기는, N개의 가장 최근 시간 간격 중 N-1개의 이전 시간 간격의 각각에 대하여 검출된 메시지의 개수의 시간-가중 합을 저장하는 제1레지스터, 및 N개의 가장 최근 시간 간격 중 가장 최근 시간 간격에 대하여 검출된 메시지의 개수를 저장하는 제2레지스터를 포함한다. 트래픽 제어기는, 또한, 제2레지스터의 개수를 제1레지스터의 시간-가중 합의 비트-시프트된 버전에 가산함으로써 새로운 시간-가중 합을 계산하는 로직을 포함한다.

[0076] 트래픽 제어기는, 또한, 소정의 기간 동안 비신뢰 디바이스로부터의 트래픽을 제한함으로써 NoC의 트래픽을 감소시키는 조절 로직을 포함한다.

[0077] 트래픽 제어기는, 또한, 로컬 트래픽이 로컬 트래픽 임계값을 초과하는지 여부를 결정하는 로컬 트래픽 모니터를 포함한다. 트래픽 제어기는, 로컬 트래픽이 로컬 트래픽 임계값을 초과할 때 위반 검출에 대한 응답으로 NoC의 트래픽을 감소시킨다.

[0078] 비신뢰 디바이스로부터의 트래픽을 조절하는 방법은, 호스트 SoC의 NoC를 통해 호스트 SoC의 내부 블록들 간에 로컬 트래픽을 송신하는 단계, 호스트 SoC에서 외부 프로세서 링크를 통해 비신뢰 디바이스로부터의 메시지를 수신하는 단계, 하나 이상의 시간 간격의 세트에 걸쳐 비신뢰 디바이스로부터의 외부 트래픽의 양을 감시하는 단계, 외부 트래픽의 양에 기초하여 트래픽 정책의 위반을 검출하는 단계, 및 위반 검출에 응답하여, 비신뢰 디바이스로부터의 메시지로 인해 발생하는 NoC의 트래픽을 감소시키는 단계를 포함한다.

[0079] 트래픽 정책의 위반을 검출하는 단계는, 시간 간격들의 세트 중 주어진 시간 간격 동안 비신뢰 디바이스로부터 호스트 SoC로 송신되는 메시지의 개수가 해당 시간 간격에 대한 최대 임계값을 초과하는지 여부를 결정하는 단

계를 더 포함한다.

[0080] 비신뢰 디바이스로부터의 트래픽을 조절하는 방법은, 최대 임계값을 변경하기 위한 요청을 외부 디바이스로부터 수신하는 단계, 요청 인증에 응답하여 그 요청에 따라 최대 임계값을 변경하는 단계, 및 요청 인증 실패에 응답하여 그 요청을 무시하는 단계를 더 포함한다.

[0081] 방법은, 비신뢰 디바이스로부터 호스트 SoC로 송신되는 메시지의 개수가 주어진 시간 간격에 대한 최대 임계값을 초과한다는 결정에 응답하여, 메시지의 개수와 그 시간 간격에 대한 최대 임계값 사이의 차에 대응하는 양만큼 후속 시간 간격에 대한 최대 임계값을 감소시키는 단계를 더 포함한다.

[0082] 트래픽 정책의 위반을 검출하는 단계는, 시간 간격들의 세트 중 N개의 가장 최근 시간 간격의 윈도우에 걸쳐 비신뢰 디바이스로부터의 메시지의 평균 개수를 계산하는 단계, 및 평균 개수가 최대 임계값을 초과하는 것으로 결정하는 단계를 포함한다.

[0083] 비신뢰 디바이스로부터의 트래픽을 조절하는 방법은, 소정 기간 동안 비신뢰 디바이스로부터의 트래픽을 제한함으로써 NoC의 트래픽을 감소시키는 단계를 더 포함한다.

[0084] 방법은, 로컬 트래픽이 로컬 트래픽 임계값을 초과하는지 여부를 결정하는 단계, 및 로컬 트래픽이 로컬 트래픽 임계값을 초과할 때 위반 검출에 응답하여 NoC의 트래픽을 감소시키는 단계를 더 포함한다.

[0085] 비신뢰 디바이스를 포함하는 컴퓨터 시스템은, 메모리, 메모리에 접속되고 외부 프로세서 링크를 통해 비신뢰 장치에 접속된 호스트 SoC를 더 포함하며, 이러한 외부 프로세서 링크는 호스트 SoC에서 비신뢰 디바이스로부터의 메시지를 수신한다. 호스트 SoC는, 비신뢰 디바이스로부터의 메시지에 대한 응답으로 데이터를 메모리로부터 비신뢰 장치로 송신한다. 호스트 SoC는, SoC의 내부 블록들 간에 로컬 트래픽을 송신하도록 구성된 NoC, 및 외부 프로세서 링크와 결합된 트래픽 제어기를 더 포함한다.

[0086] 트래픽 제어기는, 하나 이상의 시간 간격들의 세트에 걸쳐 비신뢰 디바이스로부터의 외부 트래픽의 양을 감시하고, 외부 트래픽의 양에 기초하여 트래픽 정책의 위반을 검출하고, 위반 검출에 응답하여 비신뢰 디바이스로부터의 메시지로 인해 발생하는 NoC의 트래픽을 감소시킨다. 트래픽 제어기는, 호스트 SoC와 함께 단일 집적 회로 칩 상에 위치한다.

[0087] 외부 프로세서 링크는, 또한, 외부 프로세서 링크의 버퍼 용량에 기초하여 비신뢰 디바이스로부터의 트래픽을 규제하는 흐름 제어 로직을 포함한다.

[0088] 비신뢰 디바이스로부터의 메시지는, 판독 요청, 기입 요청, 및 메모리로 향하는 어드레스 변환 요청을 포함한다.

[0089] 본원에서 사용되는 바와 같이, "결합된"이라는 용어는, 직접적으로 또는 하나 이상의 중간 구성요소를 통해 간접적으로 결합된 것을 의미할 수 있다. 본원에서 설명하는 다양한 버스를 통해 제공되는 신호들 중 임의의 신호는, 다른 신호들과 함께 시간 멀티플렉싱될 수 있고 하나 이상의 공통 버스를 통해 제공될 수 있다. 또한, 회로 구성요소들 또는 블록들 간의 상호접속부는 버스 또는 단일 신호 라인으로서 표시될 수 있다. 대안으로, 각각의 버스는 하나 이상의 단일 신호 라인일 수 있고, 단일 신호 라인의 각각은 대안으로 버스일 수 있다.

[0090] 소정의 실시예는, 비일시적 컴퓨터 판독가능 매체에 저장된 명령어를 포함할 수 있는 컴퓨터 프로그램 제품으로서 구현될 수 있다. 이들 명령어는, 설명한 동작들을 수행하도록 범용 또는 전용 프로세서를 프로그래밍하는 데 사용될 수 있다. 컴퓨터 판독가능 매체는, 기계(예를 들어, 컴퓨터)에 의해 판독가능한 형태(예를 들어, 소프트웨어, 처리 애플리케이션)로 정보를 저장하거나 송신하기 위한 임의의 메커니즘을 포함한다. 비일시적 컴퓨터 판독가능 저장 매체는, 자기 저장 매체(예를 들어, 플로피 디스켓); 광 저장 매체(예를 들어, CD-ROM); 광 자기 저장 매체; ROM; RAM; 소거가능 프로그래밍가능 메모리(예를 들어, EPROM 및 EEPROM); 플래시 메모리, 또는 전자 명령어를 저장하기에 적합한 다른 유형의 매체를 포함할 수 있지만, 이에 한정되지 않는다.

[0091] 또한, 일부 실시예는, 컴퓨터 판독가능 매체가 한 개보다 많은 컴퓨터 시스템에 저장되고 및/또는 이러한 컴퓨터 시스템에 의해 실행되는 분산형 연산 환경에서 실시될 수 있다. 또한, 컴퓨터 시스템들 간에 전송되는 정보는, 컴퓨터 시스템들을 접속하는 송신 매체를 거쳐 풀링 또는 푸싱될 수 있다.

[0092] 일반적으로, 컴퓨터 판독가능 저장 매체 상에 탑재된 호스트 SoC(210) 및/또는 그 일부를 나타내는 데이터 구조는, 프로그램에 의해 판독될 수 있고 호스트 SoC(210)를 포함하는 하드웨어를 제조하도록 직접적으로 또는 간접적으로 사용될 수 있는 데이터베이스 또는 다른 데이터 구조일 수 있다. 예를 들어, 데이터 구조는, Verilog 또

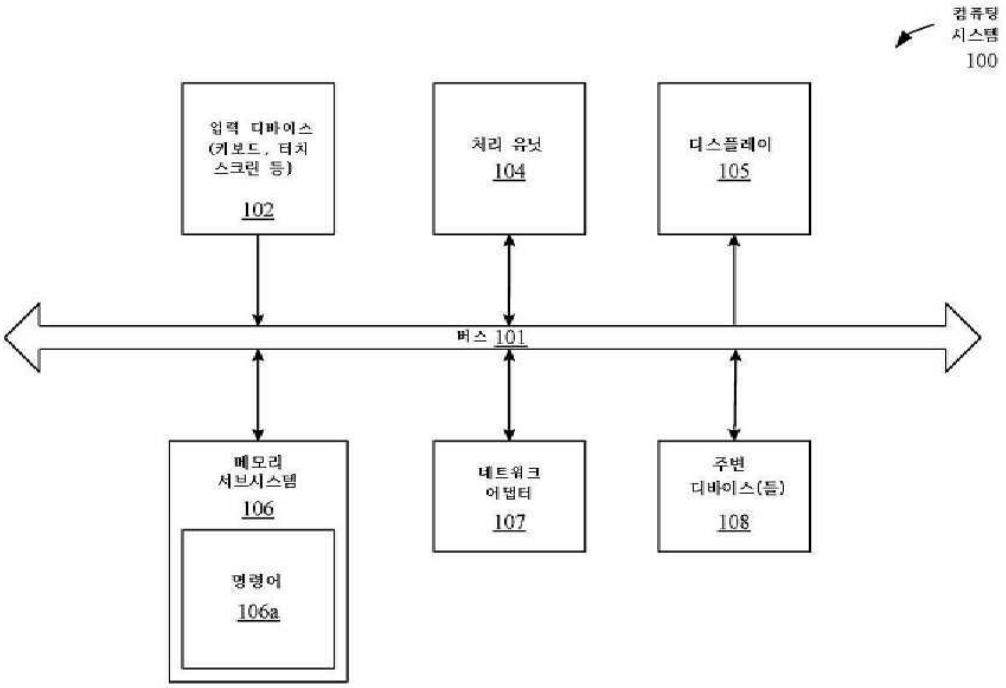
는 VHDL과 같은 고 레벨 설계 언어(HDL)의 하드웨어 기능에 대한 행동-레벨 디스크립션 또는 레지스터-전송 레벨(RTL) 디스크립션일 수 있다. 디스크립션은, 합성 라이브러리로부터의 게이트들의 리스트를 포함하는 넷리스트(netlist)를 생성하도록 디스크립션을 합성할 수 있는 합성 툴에 의해 판독될 수 있다. 넷리스트는, 호스트 SoC(210)를 포함하는 하드웨어의 기능을 또한 나타내는 게이트들의 세트를 포함한다. 이어서, 넷리스트는, 마스크에 적용할 기하학적 형상을 설명하는 데이터 세트를 생성하도록 배치 및 라우팅될 수 있다. 이어서, 마스크는, 호스트 SoC(210)에 대응하는 반도체 회로 또는 회로들을 제조하도록 다양한 반도체 제조 단계에서 사용될 수 있다. 대안으로, 컴퓨터 판독가능 저장 매체 상의 데이터베이스는, 넷리스트(합성 라이브러리를 갖거나 갖지 않음) 또는 원하는 바와 같은 데이터 세트, 또는 그래픽 데이터 시스템(GDS) II 데이터일 수 있다.

[0093] 본원의 방법(들)의 동작들이 특정한 순서로 도시되고 설명되었지만, 각 방법의 동작들의 순서는 변경될 수 있어서, 소정의 동작들이 역순으로 수행될 수 있고 또는 소정의 동작이 다른 동작들과 함께 적어도 부분적으로 동시에 수행될 수 있다. 다른 일 실시예에서, 별개의 동작의 서브 동작 또는 명령어는 간헐적 방식 및/또는 교번 방식일 수 있다.

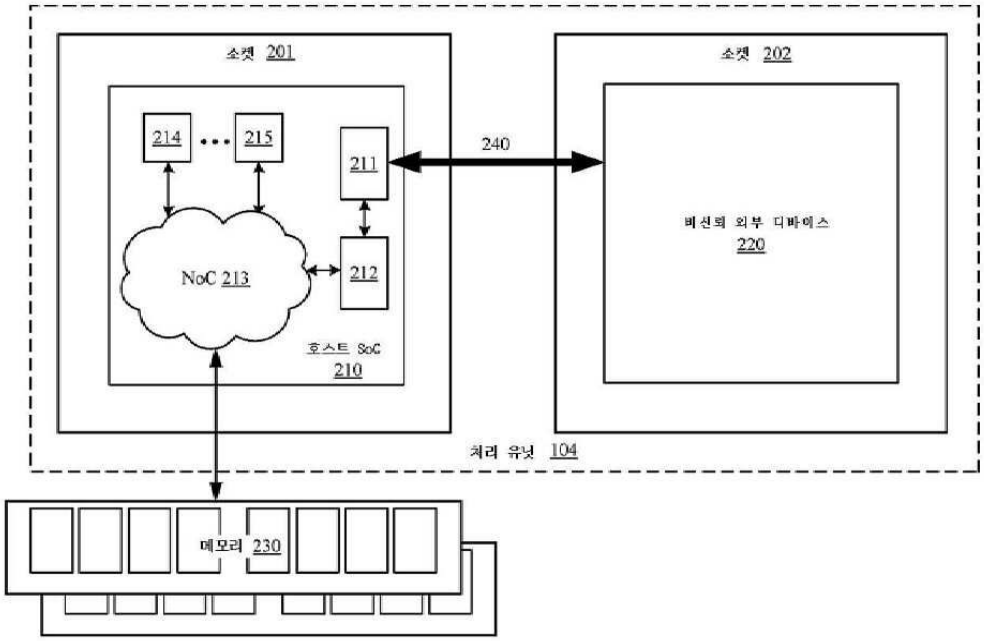
[0094] 전술한 명세서에서, 실시예들은 예시적인 실시예들을 참조하여 설명되었다. 그러나, 첨부된 청구범위에 설명된 바와 같이 실시예들의 더욱 넓은 범위를 벗어나지 않으면서 다양한 변형과 변경이 이루어질 수 있음은 명백할 것이다. 이에 따라, 명세서와 도면은 제한적인 의미라기보다는 예시적인 의미로 간주되어야 한다.

도면

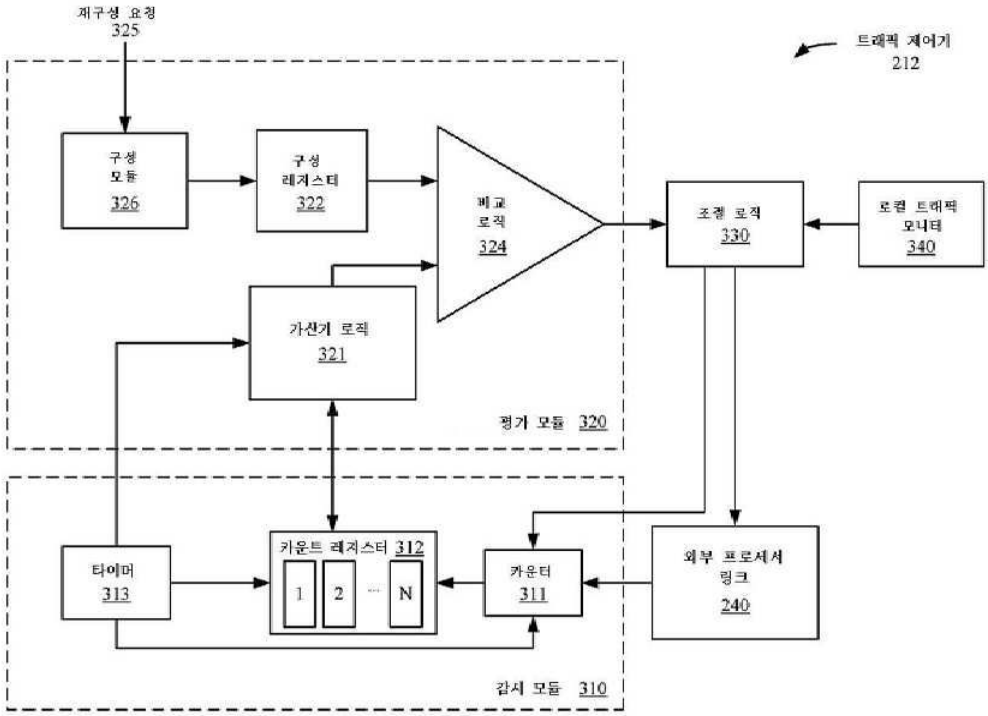
도면1



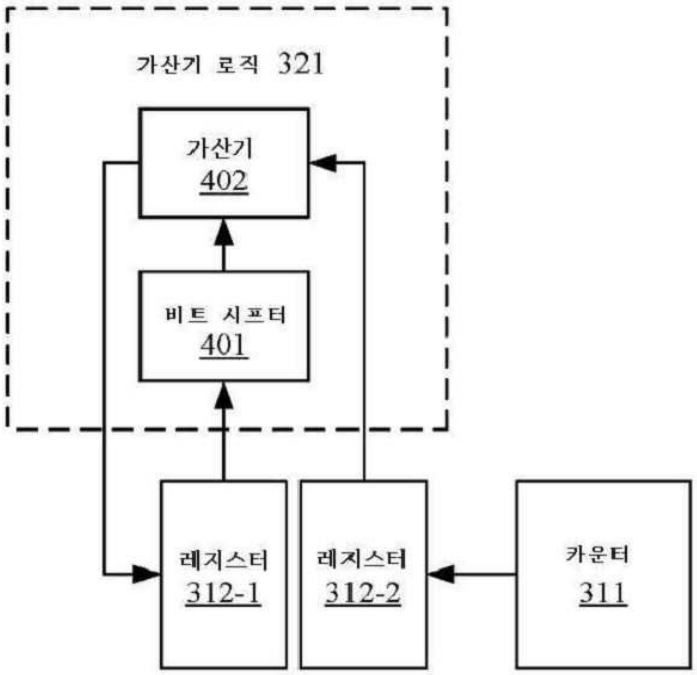
도면2


소켓 201
214
215
211
240
소켓 202
NoC 213
212
비선의 외부 디바이스
220
호스트 SoC
210
처리 유닛 104
메모리 230

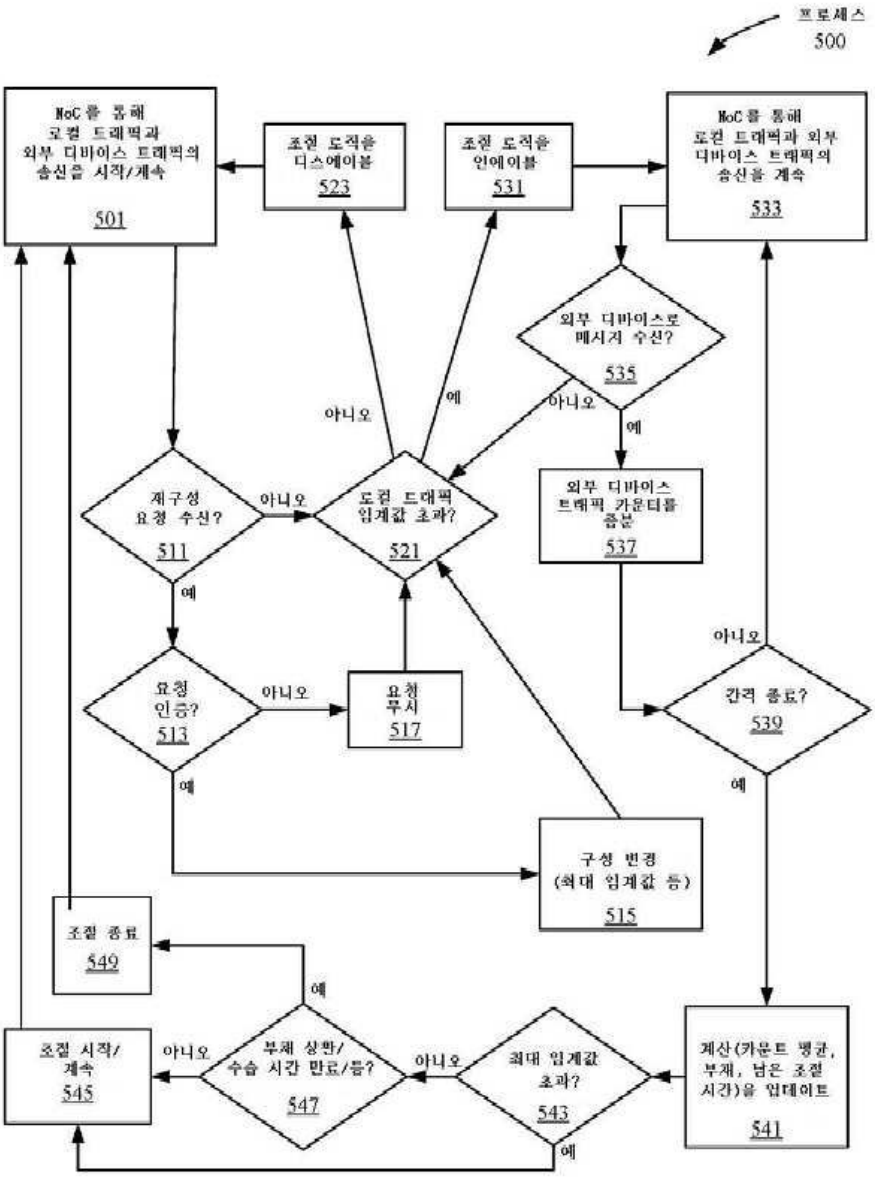
도면3


재구성 요청
325
트래픽 제어기
212
구성
모듈
326
구성
레지스터
322
비교
로직
324
조절 로직
330
로컬 트래픽
모니터
340
가산기 로직
321
평가 모듈 320
타이머
313
카운트 레지스터 312
1
2
N
카운터
311
외부 프로세서
링크
240
감시 모듈 310

도면4



도면5


프로세스
500
MoC를 통해 로컬 트래픽과 외부 디바이스 트래픽의 송신을 시작/계속
501
조절 로직을 디스에이블
523
조절 로직을 인에이블
531
MoC를 통해 로컬 트래픽과 외부 디바이스 트래픽의 송신을 계속
533
외부 디바이스로 메시지 수신?
535
아니오
예
제구성 요청 수신?
511
아니오
로컬 트래픽 임계값 초과?
521
예
외부 디바이스 트래픽 카운터를 증분
537
요청 인증?
513
아니오
요청 무시
517
예
아니오
간격 종료?
539
예
구성 변경 (최대 임계값 등)
515
조절 종료
549
예
조절 시작/계속
545
아니오
부채 상환/수습 시간 만료/등?
547
아니오
최대 임계값 초과?
543
예
계산(카운트 평균, 부채, 남은 조절 시간)을 업데이트
541