



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년04월11일
 (11) 등록번호 10-1725755
 (24) 등록일자 2017년04월05일

(51) 국제특허분류(Int. Cl.)
 H04L 12/707 (2013.01) H04L 12/729 (2013.01)
 H04L 12/751 (2013.01) H04L 12/801 (2013.01)
 H04L 12/935 (2013.01)

(52) CPC특허분류
 H04L 45/22 (2013.01)
 H04L 45/10 (2013.01)

(21) 출원번호 10-2015-7033700
 (22) 출원일자(국제) 2013년06월28일
 심사청구일자 2015년11월26일
 (85) 번역문제출일자 2015년11월26일
 (65) 공개번호 10-2016-0004348
 (43) 공개일자 2016년01월12일
 (86) 국제출원번호 PCT/US2013/048514
 (87) 국제공개번호 WO 2014/209347
 국제공개일자 2014년12월31일

(56) 선행기술조사문헌
 US20120063459 A1
 US20120170582 A1

(73) 특허권자
인텔 코포레이션
 미합중국 캘리포니아 95054 산타클라라 미션 칼리지 블러바드 2200

(72) 발명자
메지아 안드레스
 미국 캘리포니아주 95050 산타 클라라 아파트먼트 14비 사라토가 애비뉴 444
다이 동글라이
 미국 캘리포니아주 94588 플레젠튼 파크 플레이스 3459
모라 포르타 게스퍼
 미국 캘리포니아주 95054 산타 클라라 에스씨12 - 엠에스 322 미션 칼리지 2200

(74) 대리인
제일특허법인

전체 청구항 수 : 총 24 항

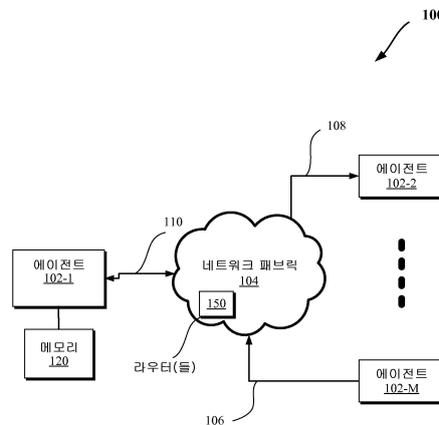
심사관 : 김상인

(54) 발명의 명칭 **적응형 라우팅을 이용하여 자원 활용도를 제어하기 위한 메커니즘**

(57) 요약

적응형 라우팅을 이용하여 자원 활용도를 제어하기 위한 기술들에 관한 방법들 및 장치가 설명된다. 일 실시예에서, 입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트가 라우팅 로직에서 결정된다. 라우팅 로직은, 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여 제 1 출력 포트 및 제 2 출력 포트로부터 출력 포트를 선택한다. 제 1 출력 포트는 착신 메시지에 대한 결정형 라우트를 제공하고, 제 2 출력 포트는 착신 메시지에 대한 적응형 라우트를 제공한다. 다른 실시예들이 또한 개시된다.

대표도



(52) CPC특허분류

H04L 45/125 (2013.01)

H04L 47/11 (2013.01)

H04L 47/12 (2013.01)

H04L 49/3018 (2013.01)

명세서

청구범위

청구항 1

입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트를 결정하기 위한 라우팅 로직을 포함하되,

상기 라우팅 로직은, 상기 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트와 제 2 출력 포트 중에서 상기 출력 포트를 선택하고,

상기 제 1 출력 포트는 상기 착신 메시지에 대한 결정형 라우트를 제공하고, 상기 제 2 출력 포트는 상기 착신 메시지에 대한 적응형 라우트를 제공하며,

상기 혼잡 정보는, 성능 가상 채널들의 수와 하위 성능 가상 채널 임계치와의 비교 또는 공유 크레딧들의 수와 하위 크레딧 임계치와의 비교 중 하나에 기초하여 결정되는

장치.

청구항 2

제 1 항에 있어서,

상기 결정형 라우트는, 소스 및 수신지 노드 쌍에 대해 동일한 경로를 선택할 수 있는

장치.

청구항 3

제 1 항에 있어서,

상기 적응형 라우트는, 트래픽 정보에 기초하여 소스 및 수신지 노드 쌍에 대해 상이한 경로를 선택할 수 있는

장치.

청구항 4

제 1 항에 있어서,

상기 성능 가상 채널들은 상기 착신 메시지를 통신할 수 있는

장치.

청구항 5

제 1 항에 있어서,

상기 공유 크레딧들은 상기 착신 메시지를 통신할 수 있는 하나 이상의 가상 채널에 대응하는

장치.

청구항 6

제 1 항에 있어서,

상기 라우팅 로직은, 상기 착신 메시지의 제 1 플릿(flit)의 검출에 응답하여 상기 출력 포트를 결정하는 장치.

청구항 7

제 1 항에 있어서,
상기 출력 포트의 이용가능성을 상기 착신 메시지와 매칭시키기 위한 스케줄러 로직을 포함하는 장치.

청구항 8

제 1 항에 있어서,
상기 입력 포트와 상기 출력 포트를 통신가능하게 커플링시키기 위한 크로스바 로직을 포함하는 장치.

청구항 9

제 1 항에 있어서,
상기 착신 메시지에 대응하는 데이터를 플릿 단위로 저장하기 위해, 하나 이상의 버퍼의 할당을 통해 데드락 (deadlock) 없는 라우팅을 지원하기 위한 로직을 포함하는 장치.

청구항 10

제 1 항에 있어서,
상기 착신 메시지에 대응하는 데이터를 패킷 레벨 단위로 통신하기 위해, 하나 이상의 가상 채널의 할당을 통해 데드락 없는 라우팅을 지원하기 위한 로직을 포함하는 장치.

청구항 11

제 10 항에 있어서,
상기 패킷은 복수의 플릿을 포함하는 장치.

청구항 12

제 1 항에 있어서,
상기 라우팅 로직은, 복수의 라우팅 가상 채널에 기초하여 데드락 없는 라우팅을 제공하는 장치.

청구항 13

제 1 항에 있어서,

상기 라우팅 로직은 제 1 에이전트를 링크를 통해 제 2 에이전트에 커플링시키고, 상기 링크는 포인트-투-포인트 상호접속을 포함하는

장치.

청구항 14

제 13 항에 있어서,

상기 제 1 에이전트와 상기 제 2 에이전트 중 하나 이상은 복수의 프로세서 코어를 포함하는

장치.

청구항 15

제 13 항에 있어서,

상기 제 1 에이전트와 상기 제 2 에이전트 중 하나 이상은 복수의 소켓을 포함하는

장치.

청구항 16

제 13 항에 있어서,

상기 제 1 에이전트, 상기 제 2 에이전트, 상기 라우팅 로직, 메모리 중 하나 이상은 동일한 집적 회로 다이 상에 있는

장치.

청구항 17

라우팅 로직에서, 입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트를 결정하는 단계를 포함하되,

상기 라우팅 로직은, 상기 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트와 제 2 출력 포트 중에서 상기 출력 포트를 선택하고,

상기 제 1 출력 포트는 상기 착신 메시지에 대한 결정형 라우트를 제공하고, 상기 제 2 출력 포트는 상기 착신 메시지에 대한 적응형 라우트를 제공하며,

상기 혼잡 정보는, 성능 가상 채널들의 수와 하위 성능 가상 채널 임계치와의 비교 또는 공유 크레딧들의 수와 하위 크레딧 임계치와의 비교 중 하나에 기초하여 결정되는

방법.

청구항 18

제 17 항에 있어서,

상기 출력 포트를 결정하는 단계는 상기 출력 포트의 이용가능성을 상기 착신 메시지와 매칭시키는 단계를 포함하는

방법.

청구항 19

제 17 항에 있어서,

상기 출력 포트를 결정하는 단계 후에 상기 입력 포트와 상기 출력 포트를 통신가능하게 커플링시키는 단계를 더 포함하는

방법.

청구항 20

제 17 항에 있어서,

상기 라우팅 로직은 상기 착신 메시지에 대응하는 데이터를 플릿 단위로 저장하기 위해, 하나 이상의 버퍼의 할당을 통해 데드락 없는 라우팅을 지원하도록 구성된

방법.

청구항 21

제 17 항에 있어서,

상기 라우팅 로직은 상기 착신 메시지에 대응하는 데이터를 패킷 레벨 단위로 통신하기 위해, 하나 이상의 가상 채널의 할당을 통해 데드락 없는 라우팅을 지원하도록 구성된

방법.

청구항 22

혼잡 정보를 저장하는 메모리와,

상기 메모리에 커플링되고, 입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트를 결정하는 라우팅 로직을 포함하되,

상기 라우팅 로직은, 상기 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트와 제 2 출력 포트 중에서 상기 출력 포트를 선택하고,

상기 제 1 출력 포트는 상기 착신 메시지에 대한 결정형 라우트를 제공하고, 상기 제 2 출력 포트는 상기 착신 메시지에 대한 적응형 라우트를 제공하며,

상기 혼잡 정보는, 성능 가상 채널들의 수와 하위 성능 가상 채널 임계치와의 비교 또는 공유 크레딧들의 수와 하위 크레딧 임계치와의 비교 중 하나에 기초하여 결정되는

시스템.

청구항 23

제 22 항에 있어서,

상기 결정형 라우트는 소스 및 수신지 노드 쌍에 대해 동일한 경로를 선택할 수 있는

시스템.

청구항 24

제 22 항에 있어서,

상기 적응형 라우트는, 트래픽 정보에 기초하여 소스 및 수신지 노드 쌍에 대해 상이한 경로를 선택할 수 있는 시스템.

청구항 25

삭제

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 일렉트로닉스 분야에 관한 것이다. 더 상세하게는, 본 발명의 실시예는 적응형 라우팅으로 자원 활용도를 제어하기 위한 기술들에 관한 것이다.

도면의 간단한 설명

[0002] 상세한 설명은 첨부된 도면들을 참조하여 제공된다. 도면들에서 참조 부호의 최좌측 숫자(들)은 그 참조 부호가 먼저 나타나는 도면을 식별시킨다. 상이한 도면들에서 동일한 참조 부호들의 이용은 유사하거나 동일한 항목들을 나타낸다.

도 1은 본 명세서에서 논의되는 다양한 실시예들을 구현하기 위해 활용될 수 있는 컴퓨팅 시스템들의 실시예에 대한 블록도를 예시한다.

도 2는 본 명세서에서 논의되는 하나 이상의 실시예들을 구현하기 위해 활용될 수 있는 컴퓨팅 시스템의 실시예에 대한 블록도를 예시한다.

도 3은 본 명세서에서 논의되는 하나 이상의 실시예들에서 이용될 수 있는 라우팅 로직의 블록도를 도시한다.

도 4는 실시예에 따라, 라우팅 기능에 대한 블록도를 예시한다.

도 5는 실시예에 따라, 라우팅 로직의 입력 포트의 가상 채널들의 로직 분할을 도시한다.

도 6은 실시예에 따라, 헤더 플릿들 상에서 수행되는 방법의 흐름도를 예시한다.

도 7은 실시예에 따라, 혼잡을 체크하기 위한 방법의 흐름도를 예시한다.

도 8은 실시예에 따라, 출력 포트를 선택하기 위한 방법의 흐름도를 예시한다.

도 9는 본 명세서에서 논의되는 하나 이상의 실시예들을 구현하기 위해 활용될 수 있는 컴퓨팅 시스템의 실시예에 대한 블록도를 예시한다.

도 10은 본 명세서에서 논의되는 하나 이상의 실시예들을 구현하기 위해 활용될 수 있는 컴퓨팅 시스템의 실시예에 대한 블록도를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0003] 다음 설명에서, 다양한 실시예들의 철저한 이해를 제공하기 위해 다수의 특정 세부사항들이 기술된다. 그러나, 몇몇 실시예들은 특정 세부사항들 없이 실시될 수 있다. 다른 예들에서, 특정 실시예들을 모호하게 하지 않기 위해, 주지의 방법들, 절차들, 구성요소들 및 회로들은 상세히 설명되지 않았다. 본 발명의 실시예들의 다양한 양상들은 다양한 수단들, 예를 들어, 집적 반도체 회로들("하드웨어"), 하나 이상의 프로그램들로 조직화된 컴퓨터 판독가능 명령들("소프트웨어") 또는 하드웨어와 소프트웨어의 몇몇 조합을 이용하여 수행될 수 있다. 본 개시의 목적을 위해, "로직"에 대한 참조는 하드웨어, 소프트웨어 또는 이들의 몇몇 조합을 의미할 것이다.

[0004] 네트워크 시스템들을 설계하는 경우, 몇몇 주요 고려사항들은 면적 및 전력 손실을 최소화하는 것이다. 이러한 제한들은, 토폴로지, 스위칭 전략들, 라우팅 기능 및 아키텍처 구현들에 대한 특정한 선택들을 부과한다. 그 다음, 이러한 선택들을 밸런싱하는 것은, 네트워크 시스템들이 속도, 전력 소모 등의 관점에서 얼마나 양호하게 동작하는지에 대해 직접적인 영향을 미친다.

- [0005] 몇몇 실시예들은, 적응형 라우팅에 의한 자원 활용을 제어하기 위한 기술들과 관련된다(이는 또한 혼잡 인식 적응성 임계치 또는 CAAT로 지칭될 수 있다). 일 실시예에서, (입력 포트에서 수신될) 착신 메시지의 송신을 위한 출력 포트는 라우팅 로직(예를 들어, 도 1 내지 도 10을 참조하여 논의되는 로직(150))에서 결정된다. 라우팅 로직은, 그 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트 및 제 2 출력 포트로부터 출력 포트를 선택한다. 제 1 출력 포트는 착신 메시지에 대한 결정형(deterministic) 라우트를 제공하고, 제 2 출력 포트는 착신 메시지에 대한 적응형 라우트를 제공한다.
- [0006] 본 명세서에서 논의되는 실시예들, 예를 들어, 도 1 내지 도 2 및 도 9 내지 도 10을 참조하여 논의되는 시스템들을 구현하기 위해 다양한 컴퓨팅 시스템들이 이용될 수 있다. 더 상세하게는, 도 1은, 본 발명의 실시예에 따른 컴퓨팅 시스템(100)의 블록도를 예시한다. 시스템(100)은 하나 이상의 에이전트들(102-1 내지 102-M)(총괄적으로 본 명세서에서는 "에이전트들(102)" 또는 더 일반적으로 "에이전트(102)"로 지칭됨)을 포함할 수 있다. 실시예에서, 에이전트들(102) 중 하나 이상은 도 9 내지 도 10을 참조하여 논의되는 컴퓨팅 시스템들과 같은 컴퓨팅 시스템의 임의의 구성요소들일 수 있다.
- [0007] 도 1에 예시된 바와 같이, 에이전트들(102)은 네트워크 패브릭(104)을 통해 통신할 수 있다. 일 실시예에서, 네트워크 패브릭(104)은, 다양한 에이전트들(예를 들어, 컴퓨팅 디바이스들)이 데이터를 통신하도록 허용하는 컴퓨터 네트워크를 포함할 수 있다. 실시예에서, 네트워크 패브릭(104)은, 직렬(예를 들어, 포인트-투-포인트) 링크 및/또는 공유 통신 네트워크(실시예에서 링으로 구성될 수 있음)를 통해 통신하는 하나 이상의 상호접속들(또는 상호접속 네트워크들)을 포함할 수 있다. 예를 들어, 몇몇 실시예들은, FBD(Fully Buffered Dual) 인-라인 메모리 모듈들과의 통신을 허용하는 링크들 상에서, 예를 들어, FBD 링크가 메모리 모듈들을 호스트 제어기 디바이스(예를 들어, 프로세서 또는 메모리 허브)에 커플링하기 위한 직렬 링크인 경우, 구성요소 디버그 또는 검증용 용이하게 할 수 있다. 디버그 정보는 FBD 채널 호스트로부터 송신될 수 있어서, 디버그 정보는 채널 트래픽 트레이스 캡처 툴들(예를 들어, 하나 이상의 로직 분석기들)에 의해 채널을 따라 관측될 수 있다.
- [0008] 일 실시예에서, 시스템(100)은, 물리 계층, 링크 계층, 라우팅 계층, 전송 계층 및/또는 프로토콜 계층을 포함할 수 있는 계층화된 프로토콜 방식을 지원할 수 있다. 패브릭(104)은 포인트-투-포인트 또는 공유된 네트워크를 위해, 하나의 프로토콜(예를 들어, 캐싱 프로세서 또는 캐싱 인식 메모리 제어기)로부터 다른 프로토콜로(예를 들어, 패킷들의 형태로) 데이터의 송신을 더욱 용이하게 할 수 있다. 또한, 몇몇 실시예들에서, 네트워크 패브릭(104)은, 하나 이상의 캐시 코히어런트 프로토콜들을 고수하는 통신을 제공할 수 있다.
- [0009] 게다가, 도 1의 화살표들의 방향으로 도시된 바와 같이, 에이전트들(102)은 네트워크 패브릭(104)을 통해 데이터를 송신 및/또는 수신할 수 있다. 따라서, 몇몇 에이전트들은 단방향 링크를 활용할 수 있는 한편, 다른 에이전트들은 통신을 위해 양방향 링크를 활용할 수 있다. 예를 들어, 하나 이상의 에이전트들(예를 들어, 에이전트(102-M))은 데이터를 (예를 들어, 단방향 링크(106)를 통해) 송신할 수 있고, 다른 에이전트(들)(예를 들어, 에이전트(102-2))는 데이터를 (예를 들어, 단방향 링크(108)를 통해) 수신할 수 있는 한편, 몇몇 에이전트(들)(예를 들어, 에이전트(102-1))는 데이터를 (예를 들어, 양방향 링크(110)를 통해) 송신 및 수신하는 것 모두를 할 수 있다.
- [0010] 추가적으로, 에이전트들(102) 중 적어도 하나는 홈 에이전트일 수 있고, 에이전트들(102) 중 하나 이상은 요청 또는 캐싱 에이전트들일 수 있다. 일반적으로, 요청/캐싱 에이전트들은, 대응하는 "홈 에이전트"가 연관되는 메모리 어드레스에 대한 액세스를 위해 홈 노드/에이전트에 요청(들)을 전송한다. 추가로, 실시예에서, 에이전트들(102) 중 하나 이상(에이전트(102-1)에 대해 오직 하나만 도시됨)은 메모리(120)와 같은 메모리(그 에이전트에 전용되거나 다른 에이전트들과 공유될 수 있음)에 대한 액세스를 가질 수 있다. 몇몇 실시예들에서, 에이전트들(102) 각각(또는 적어도 하나)은, 에이전트와 동일한 다이 상에 있거나 그렇지 않으면 에이전트에 의해 액세스가능한 메모리(120)에 커플링될 수 있다. 또한, 도 1에 도시된 바와 같이, 네트워크 패브릭(104)은, 에이전트간 통신을 허용하기 위해 2개의 에이전트를 커플링시키기 위한 하나 이상의 라우팅 로직(150)을 포함할 수 있다. 아울러, 실시예에서, 매우 얇은 하드웨어 네트워크 인터페이스가 하나 이상의 라우팅 로직(150)에 모든 에이전트(102)를 커플링시킨다. 얇은 하드웨어 네트워크 인터페이스는, 아래에서 설명되는 Yunit 및 Shim 하드웨어 블록들과 유사하지만 상보적일 수 있다.
- [0011] 도 2는, 실시예에 따른 컴퓨팅 시스템(200)의 블록도이다. 시스템(200)은 복수의 소켓들(202-208)을 포함한다(4개가 도시되지만, 몇몇 실시예들은 더 많거나 더 적은 소켓을 가질 수 있다). 각각의 소켓은 프로세서를 포함한다. 또한, 시스템(200)의 다양한 에이전트들은 라우팅 로직(150)을 통해 통신할 수 있다. 라우팅 로직

(150)은 오직 202-208을 커플링하는 것으로 도시되지만, 라우팅 로직(150)은 시스템(200)의 다른 에이전트들을 커플링할 수 있다. 추가로, 구현에 따라 더 많거나 더 적은 라우팅 로직 블록들이 시스템에 존재할 수 있다. 추가적으로, 각각의 소켓은 포인트-투-포인트(PtP) 링크 또는 차동 상호접속, 예를 들어, QPI(Quick Path Interconnect), MIPI(Mobile Industry Processor Interface) 등을 통해 다른 소켓들에 커플링된다. 도 1의 네트워크 패브릭(104)에 대해 논의된 바와 같이, 각각의 소켓은, 예를 들어, 동적 랜덤 액세스 메모리(DRAM)를 포함하는 복수의 듀얼 인라인 메모리 모듈들(DIMM들)에 의해 형성된 시스템 메모리의 로컬 부분에 커플링된다.

[0012] 다른 실시예에서, 네트워크 패브릭은, 임의의 시스템 온칩(SoC 또는 SOC) 애플리케이션에 대해 활용될 수 있고, 관습적 또는 표준 인터페이스들, 예를 들어, AMBA(Advanced Microcontroller Bus Architecture), OCP(Open Core Protocol), MIPI(Mobile Industry Processor Interface), PCI(Peripheral Component Interconnect) 또는 PCIe(Peripheral Component Interconnect Express)를 위한 ARM 준수 인터페이스들을 활용할 수 있다.

[0013] 몇몇 실시예들은, IP 자원들 자체에 대한 어떠한 변경들도 없이, PCI-기반 시스템과 같은 PC(Personal Computer) 기반 시스템에서, AXI/OCP 기술들과 같은 이종 자원들의 이용을 가능하게 하는 기술을 이용한다. 실시예들은, 본 명세서에서 Yunit 및 shim으로 지칭되는 2개의 매우 얇은 하드웨어 블록들을 제공하고, 이들은 AXI/OCP IP를 자동-생성된 상호접속 패브릭에 플러그인하여 PCI-호환가능 시스템들을 생성하기 위해 이용될 수 있다. 일 실시예에서, Yunit의 제 1(예를 들어, 북쪽) 인터페이스는, 직접 미디어 인터페이스(DMI) 버스, PCI 버스 또는 PCIe(Peripheral Component Interconnect Express) 버스와 같은 PCI-호환가능 버스에 인터페이스하는 어댑터 블록에 접속한다. 제 2(예를 들어, 남쪽) 인터페이스는, AXI/OCP 상호접속과 같은 넌-PC 상호접속에 직접 접속한다. 다양한 구현들에서, 이러한 버스는 OCP 버스일 수 있다.

[0014] 몇몇 실시예들에서, Yunit은, PCI 구성 사이클들을, 타겟 IP가 이해할 수 있는 트랜잭션들로 전환함으로써 PCI 열거를 구현한다. 이러한 유닛은 또한, 재배치가 가능한 PCI 어드레스들로부터 고정된 AXI/OCP 어드레스들로의 어드레스 전환 또는 그 반대를 수행한다. Yunit은 생산자-고객 모델(예를 들어, PCI 생산자-고객 모델)을 충족시키기 위한 순서화 메커니즘을 추가로 구현할 수 있다. 그 다음, 개별적인 IP들이 전용 PCI shim들을 통해 상호 접속에 접속된다. 각각 shim은 대응하는 IP에 대한 전체 PCI 헤더를 구현할 수 있다. Yunit은 모든 액세스들을 PCI 헤더에 라우팅하고 디바이스 메모리 공간을 shim에 라우팅한다. shim은 모든 헤더 판독/기록 트랜잭션들을 소모하고 다른 트랜잭션들을 IP에 전달한다. 몇몇 실시예들에서, shim은 또한 IP에 대한 모든 전력 관리 관련 특징들을 구현한다.

[0015] 따라서, Yunit을 구현하는 실시예들은, 단조적 호환가능성 블록이기 보다는 분산형 접근법을 취한다. 모든 IP들에 걸쳐 공통되는 기능, 예를 들어, 어드레스 전환 및 순서화는 Yunit에서 구현되는 한편, 전력 관리, 에러 핸들링 등과 같은 IP-특정 기능은, 그 IP에 맞춤형된 shim들에서 구현된다.

[0016] 이러한 방식으로, 새로운 IP는 최소한의 변경들로 Yunit에 추가될 수 있다. 예를 들어, 일 구현에서, 새로운 엔트리를 어드레스 재지향 테이블에 추가함으로써 변경들이 발생할 수 있다. shim들은 IP-특정적인 한편, 몇몇 구현들에서, 대부분의 기능(예를 들어, 90% 초과)은 모든 IP들에 걸쳐 공통이다. 이것은, 새로운 IP에 대해 기존 shim의 신속한 재구성을 가능하게 한다. 따라서, 몇몇 실시예들은 또한 자동-생성된 상호접속 패브릭들의 변형없는 이용을 가능하게 한다. 포인트-투-포인트 버스 아키텍처에서, 상호접속 패브릭들을 설계하는 것은 어려운 작업일 수 있다. 앞서 설명된 Yunit 접근법은, 최소의 노력으로 그리고 산업-표준 툴들에 대한 어떠한 변형들도 요구함이 없이 산업 생태계를 PCI 시스템으로 레버리지한다.

[0017] 도 2에 도시된 바와 같이, 각각의 소켓은 메모리 제어기(MC)/홈 에이전트(HA)(예를 들어, MCO/HA0 내지 MC3/HA3)에 커플링된다. 메모리 제어기들은 대응하는 로컬 메모리(MEMO 내지 MEM3로 라벨링됨)에 커플링되고, 로컬 메모리는 (도 9의 메모리(912)와 같은) 시스템 메모리의 일부일 수 있다. 몇몇 실시예들에서, 메모리 제어기(MC)/홈 에이전트(HA)(예를 들어, MCO/HA0 내지 MC3/HA3)는 도 1의 에이전트(102-1)와 동일하거나 유사할 수 있고, MEMO 내지 MEM3으로 라벨링된 메모리는 본 명세서의 임의의 도면들을 참조하여 논의되는 메모리 디바이스들과 동일하거나 유사할 수 있다. 또한, 일 실시예에서, MEMO 내지 MEM3은, 예를 들어, 마스터 및 슬레이브와 같이 데이터를 미러링하도록 구성될 수 있다. 또한, 시스템(200)의 하나 이상의 구성요소들은 몇몇 실시예들에서 동일한 집적 회로 다이 상에 포함될 수 있다.

[0018] 게다가, (예를 들어, 도 2에 도시된 바와 같은) 일 구현은 미러링에 의한 소켓 글루리스스(glueless) 구성에 관한 것이다. 예를 들어, 메모리 제어기(예를 들어, MCO/HA0)에 할당되는 데이터는 PtP 링크들을 통해 다른 메모리 제어기(예를 들어, MC3/HA3)에 미러링된다.

- [0019] 일반적으로, 네트워크 시스템들을 설계하는 경우, 몇몇 주요 고려사항들은 면적 및 전력 손실을 최소화하는 것이다. 이러한 제한들은, 토폴로지, 스위칭 전략들, 라우팅 기능 및 아키텍처 구현들에 대한 특정한 선택들을 부과한다. 아울러, 네트워크 설계는, 특정 시스템의 대역폭 요건들을 충족시키기 위해 함께 상호접속되는 몇몇 라우터들(예를 들어, 라우팅 로직(150))을 이용할 수 있다.
- [0020] 도 3은, 몇몇 실시예들에서 이용될 수 있는 라우팅 로직의 아키텍처를 도시한다. 하나 이상의 링크 제어기들("LC" 미도시)은, 2개의 인접한 스위치들의 물리 링크에 걸쳐 메시지(들)(또한 본 명세서에서는 패킷(들)과 상호교환가능하게 지칭됨)의 흐름을 조정하기 위해 이용된다. 버퍼들(예를 들어, 입력 버퍼들(302) 및 출력 버퍼들(304))은, 라우팅 로직(150)을 통해 송신되는 메시지들을 저장한다. 라우팅 유닛 로직(306)은, 입력 포트(310)에서의 착신 패킷에 대한 출력 포트(308)를 선택함으로써 라우팅 기능을 구현한다. 크로스바 유닛 로직(312)은 스위치 입력 버퍼들(302)을 스위치 출력 버퍼들(304)에 접속시킨다. 스케줄러 유닛 로직(314)은, 출력 포트 이용가능성들을, 입력 포트들에 위치한 메시지들로부터 발신되는 요청들에 매칭시킴으로써 모든 사이클마다 크로스바 유닛 로직(312)을 (예를 들어, 동기식으로) 구성한다. 라우팅 유닛 로직(306)은, 네트워크의 궁극적 성능을 도출하는 가장 중요한 파라미터들 중 하나이다. 라우팅 유닛을 설계하는 것은 다수의 가상 채널들을 갖는 무손실 네트워크들에 대해서는 어려운데, 이는, 완전히 접속되고 데드락(deadlock)들로부터 안전한 네트워크, 라이브-락(live-lock)들 등을 유지하면서 네트워크의 최대 잠재력을 추구(즉, 높은 대역폭 및 낮은 레이턴시를 달성하면서 낮은 전력을 소모)할 필요가 있기 때문이다. 메시지들은 일반적으로 헤더 플릿(flit)에서 수신지 어드레스를 운반한다. 특정 입력 포트(예를 들어, S)를 통해 라우팅 로직(150)에 도달하는 모든 헤더 플릿에 대해, 수신지 어드레스(예를 들어, D)가 검사되고, 모든 가능한 출력 포트들(예를 들어, P)이 선택된다. 몇몇 실시예들에서, 각각의 수신지 노드에 대해 이용될 수 있는 출력 포트(들)를 저장하는 테이블 기반 접근법이 이용된다. 다른 실시예들에서, 각각의 수신지 노드에 대해 이용될 수 있는 출력 포트(들)를 저장하는 최소 경로 분산형 라우팅 방식이 이용된다. 후자의 방법은, 최소 로직을 이용할 수 있고, 상이한 라우팅 알고리즘들의 이용을 허용하며, 스위치들에서 (에너지-소모 및 면적의 관점에서) 고가의 라우팅 테이블들을, 빠르고 효율적인 로직 블록들의 세트에 대체한다.
- [0021] 도 4는, 실시예에 따른 라우팅 기능의 주 구성요소들의 블록도를 예시한다. 실시예에서, 라우팅 기능은 도 3의 라우팅 유닛 로직(306)에 의해 구현된다. 도 4에 도시된 바와 같이, 라우팅 기능을 설계하는 경우 고려할 2개의 주 구성요소들, 즉, 효율적이고 효과적인 라우팅 프레임워크의 설계, 및 라우팅 알고리즘의 선택 및 그와 연관된 선택 기능이 존재한다.
- [0022] 일반적으로, 라우팅 알고리즘은 네트워크를 통한 패킷들에 의해 이용되는 경로들을 지정한다. 설계자들은, 가능한 최소의 통신 레이턴시를 전달하고, 가능한 한 효율적으로 내부 대역폭을 이용하고, 또한 혼잡이 발생하거나 병목현상이 검출되는 경우 트래픽을 적응시키는 라우팅 알고리즘들을 설계하는 문제에 직면한다. 라우팅 알고리즘들은, 3가지 상이한 카테고리들, 즉, 결정형, 적응형, 및 무의식형(oblivious)으로 분류될 수 있다.
- [0023] 결정형 라우팅 알고리즘은, 소스-수신지 노드 쌍에 대해 항상 동일한 경로를 선택한다. 결정형 접근법의 주 이점들 중 하나는, 단순성, 레이턴시 예측가능성 뿐만 아니라 패킷들의 순서대로의 도달의 보존이다. 그러나, 결정형 라우팅은, 모든 소스 및 수신지 쌍에 대해 항상 동일한 경로를 제공하기 때문에 네트워크 자원들의 비효율적 이용을 초래할 수 있다. 따라서, 결정형 라우팅 알고리즘들을 이용하는 네트워크들은, 처리율 악화로 초래되는 네트워크 혼잡 또는 병목현상들에 동적으로 대응하지 못한다.
- [0024] 한편, 무의식형 라우팅은, 동일한 소스-수신지 노드 쌍으로부터 상이한 경로들을 고려하지만, 경로는, 현재 네트워크 로드와 대한 어떠한 지식없이 소스 노드에서 선택된다.
- [0025] 다른 대안은, 패킷들이 상이한 경로들을 취하는 경우 적응형 라우팅을 이용하는 것이다. 그러나, 현재 트래픽 조건들에 따라 네트워크 내에서 차례들이 판단되고, 따라서, 대안적인 라우팅 경로들을 이용함으로써 네트워크 내의 경합을 회피한다. 이것은, 낮은 트래픽 조건 및 중간 트래픽 조건 하에서 더 높은 처리율 및 자원 활용도를 도출한다. 불행하게도, 적응형 라우팅은, 높은 로드 또는 병목현상들이 존재하는 경우 혼잡을 네트워크에 걸쳐 쉽게 확산시키는 경향이 있어서, 이러한 조건들이 존재하는 경우 처리율의 극단적인 손실을 초래한다.
- [0026] 아울러, 적응형 라우팅 메커니즘들은 2개의 상이한 기능들, 즉, 라우팅 및 선택으로 구성된다. 도 4에 도시된 바와 같이, 라우팅 기능은, 실시예에 따라 노드들 사이에 데드락 없음 및 완전한 접속도를 제공하는 출력 채널들의 세트를 제공한다. 선택 기능은, 트래픽 조건들에 대해 다른 노드들로부터 수집된 정보에 기초하여, 모든 가능한 옵션들 중 오직 하나의 옵션만을 선택하는 것을 담당한다.

- [0027] 설계자들은 통상적으로, DOR(Dimension Order Routing)과 같은 결정형 라우팅 알고리즘들을 지원하기 위해 유한 상태 머신들을 이용하여 간단한 라우팅 프레임워크들을 구현하며, 여기서 패킷들은, 수평 차원에서 먼저 라우팅되고 그 다음 수직 차원에서 라우팅된다. 분산형 라우팅을 구현하기 위한 다른 방법은, 각각의 수신지 노드에 대해 이용되어야 하는 출력 포트(들)를 저장하는 테이블을 이용하는 것이다. 이러한 접근법의 주 이점은, 임의의 네트워크 토폴로지 및 라우팅 알고리즘을 구현하기 위해 동일한 일반적인 설계가 재사용될 수 있다는 점이다. 주 난제는, 두 라우팅 알고리즘들 모두의 이점들, 즉, 높은 로드 조건들 하에서 결정형 라우팅의 레이턴시 예측가능성, 및 적응형 라우팅 방식들에 의해 제공되는 중간 및 낮은 로드 조건들 하에서의 높은 자원 활용도를 결합하는 (면적 및 에너지 관점에서) 효율적인 라우팅 프레임워크를 설계하는 것이다.
- [0028] 일반적으로, 상호접속 상의 트래픽은, 2개의 에이전트들(예를 들어, 도 1을 참조하여 논의된 에이전트들(102)) 사이에서 송신되고 있는 메시지들을 지칭한다. 이러한 메시지들은 다수의 플릿들로 송신될 수 있다. 플릿(플로우 제어 숫자에 대해서는 짧음)은 플로우 제어의 최소 단위이다. 메시지들은 일반적으로 다수의 플릿들로 이루어진다. 추가적으로, "플릿"은 실시예에서 80 비트를 포함할 수 있어서, 예를 들어, 여러 검출을 위한 8 비트, 링크 계층 헤더를 위한 8 비트, 및 데이터를 위한 64 비트를 포함할 수 있지만, 다른 구성들이 또한 이용될 수 있다. 라우터들(예를 들어, 로직(150))은, 모든 메시지 타입들이 동일한 물리 네트워크에 맵핑되도록 허용하는 다수의 가상 채널들을 지원하기 위한 이용가능성을 갖도록 설계될 수 있고, 따라서, 네트워크의 임의의 포인트에서 임의의 시간에 주어진 트래픽 혼합에 대해 이용가능한 대역폭의 최적의 이용을 제공할 수 있다.
- [0029] 하나의 네트워크 아키텍처 실시예는, 결정형, 오류-허용형 및 적응형 라우팅의 다양한 상태에서 데드락없는 라우팅에 대한 지원을 가능하게 하기 위해 가상 채널 플로우 제어를 이용한다. 또한, 버퍼들은 플릿 단위로 할당되는 한편, 가상 채널(VC)은 패킷 레벨로 할당된다. (도 5에 도시된 바와 같은) 실시예에서, 가상 채널들(VC)의 세트는 유연하게 2개의 로직 세트들, 즉, 라우팅 VC들(예를 들어, VC0-VC3) 및 성능 VC들(예를 들어, VC4-VC9)로 파티셔닝된다.
- [0030] 라우팅 VC들은, 이용되는 특정한 완전 라우팅 알고리즘의 데드락 없음 조건들을 충족하기 위해 이용될 수 있다. 이러한 상황에서, 라우팅 VC들은 데드락-없는 라우팅 알고리즘들을 지원하기 위해 요구되는 탈출 자원들을 제공한다. 또한, 라우팅 VC들은, 데드락-없는 프로토콜을 보장할 메시지 분류들 사이의 트래픽 분리를 제공하기 위해 이용된다. 이를 위해, 실시예는, 적어도 프로토콜에 존재하는 메시지 분류들만큼의 라우팅 VC들을 구현하여, 정확한 메시지 분류에 속하는 패킷들만이 그 메시지 분류에 할당된 라우팅 VC를 이용하게 한다. 아울러, 도 5는, 실시예에 따라, 라우터의 입력 포트의 10개의 가상 채널들의 로직 분할을 도시한다. VC들 중 4개(예를 들어, VC0-VC3)는 라우팅 VC들이고 나머지 6개(예를 들어, VC4-VC9)는 성능 VC 분류에 속한다.
- [0031] 몇몇 실시예들에서, 데드락-없는 라우팅 알고리즘들 및 프로토콜 트래픽 분리를 지원하는 것은, 패킷을 다른 라우팅 VC에 전송하기 위한 라우팅 VC에 대해, 다음의 규칙들, (a) 수신지 라우팅 VC가 "탈출" 포트인 것(데드락-없음이 요구되면, 동일한 메시지 분류에 대한 네트워크의 라우팅 VC들 사이의 의존도 그래프는 사이클들을 제시할 수 없음); 및/또는 (b) 수신지 라우팅 VC의 메시지 분류가 매칭하는 것(프로토콜 트래픽 분리)을 따름을 의미한다.
- [0032] 각각의 라우팅 VC는, 플릿을 수신하기 위해 자기 자신의 예비된 크레딧을 가질 수 있다(라우팅 VC가 이용되지 않으면, 적어도 하나의 플릿에 대한 공간을 제공하는 것을 보장한다). 성능 VC들은 (예를 들어, 적응형 또는 결정형 라우팅 방식들 둘 모두에 대한) 성능 개선을 위해 이용되는 VC들의 공통 공유 풀(pool)에 속할 수 있다. 성능 VC들은, 이들의 메시지 분류 제휴로 인해, 임의의 제한 없이 임의의 패킷에 의해 이용되도록 이용가능한 가상 채널들의 풀을 형성할 수 있다. 라우팅 VC들과는 반대로, 성능 VC들은 어떠한 예비된 크레딧도 갖지 않을 수 있어서, 임의의 잠재적인 착신 패킷은, 공간을 체크하기 위해 공유 크레딧 풀에 의존해야 한다.
- [0033] 요구된 라우팅 정보를 갖는 헤더 플릿이 입력 포트에 도달하자마자, 헤더 플릿은, 다음의 다운스트림 라우팅 로직에 커플링되는 대응하는 출력 포트로의 액세스를 획득하려 시도한다. 그러나, 이를 위해, (버퍼 공간 및 자유로운 VC 이용가능성을 포함하는) 자원들이 있는지를 체크할 필요가 있다.
- [0034] 몇몇 실시예들에서, 다음 라우터의 입력 포트에서 자유로운 VC를 요청하는 경우, 우선순위 순서화가 이용된다.
- [0035] (a) 먼저, 자유로운 성능 VC 및 크레딧 이용가능성을 체크한다 (자유로운 성능 VC가 크레딧 이용가능성을 보장하는 것은 아니다). 두 조건들 모두가 충족되면, 스위치 할당 파이프라인에 대한 이러한 요청을 준비한다.
- [0036] (b) 그렇지 않으면, 패킷의 메시지 분류에 속하는 자유로운 라우팅 VC에 대해 체크한다. 발견되면, 스위치 할당 파이프라인에 대한 이러한 요청을 준비한다.

- [0037] 실시예에서, 가상 채널들은 패킷 단위로 할당된다. 따라서, VC는, 이전 패킷에 대한 모든 플릿들이 로직(150)에 의해 포워딩된 경우에만 재사용될 수 있다. 이 때문에, 패킷의 테일(tail) 플릿이 입력 포트를 떠나면, VC는 다음 착신 패킷에 대해 이용가능하게 된다.
- [0038] 도 6은, 실시예에 따라, 입력 포트에서의 수신 시에, 헤더 플릿들(메시지의 제 1 플릿) 상에서 수행되는 방법의 흐름도를 예시한다. 이 방법은, 새로운 메시지 헤더가 스위치(예를 들어, 라우팅 로직(150))에 도달할 때마다 반복될 수 있다.
- [0039] 도 6을 참조하면, 동작(602)에서, 헤더 메시지(예를 들어, 메시지의 제 1 플릿)이 입력 포트에 도달하면, 라우팅 기능은 (예를 들어, 로직(150)을 통해) 모든 가능한 출력 포트들에 관한 정보를, 동작(604)의 기본적 라우팅 알고리즘에 기초하여 수집한다. 완전한 적응형 라우팅 및 부분적 적응형 라우팅의 경우, 레지스터 output_port_0이 탈출 라우트에 대한 출력 포트를 저장하는 한편, 레지스터 output_port_1은 적응성을 제공하는 출력 포트를 저장한다. 이와 동시에(또는 동작(604) 이후), 라우팅 로직(150)은 동작(606)에서, 이웃 라우터들(예를 들어, 다른 로직(150)에 근접한 로직(150), 여기서 근접은 하나 이상의 다른 로직(150)을 통한 통신 능력을 나타냄)에서 검출된 임의의 혼잡에 대한 정보를 (예를 들어, 일정하게) 업데이트할 수 있다. 이러한 정보에 기초하여, 라우팅 로직(150)은 2개의 가능한 출력 포트들에 대한 적절한 정보를 선택하고, 동작(608)에서 이들을 구성가능한 파라미터들(예를 들어, UPPER_PERFVC(상위 성능 VC), UPPER_CREDIT(상위 크레딧 값), LOWER_PERFVC(하위 성능 VC), 및 LOWER_CREDIT(하위 크레딧 값))에 비교한다. 이러한 절차는, 잠재적인 출력 포트들 각각에 대한 혼잡 플래그 레지스터를 (예를 들어, 동시에) 업데이트함으로써 요약될 수 있다. 아울러, 선택 기능은, 이전 단계 동안 업데이트된 혼잡 플래그 레지스터 및 출력 포트에서 이용가능한 VC의 타입에 기초하여 가장 편리한 출력 포트를 선택하는 것을 담당한다. 라우팅 기능이 종료되면, 패킷 콘텍스트는 스위치에서 업데이트되고, 패킷은, 스위치 파이프라인을 따라 적절한 출력 포트에 대해 경쟁하는 것으로 진행된다.
- [0040] 게다가, 라우팅 알고리즘은, 노드들 사이에서 데드락 없음 및 완전한 접속도를 제공하는 출력 포트들의 세트들(라우팅 알고리즘에 기초하여) 제공하는 것을 담당한다. 예를 들어, 라우팅 알고리즘에 의해 제공되는 2개의 가능한 출력 포트들 중, output_port_0은, 라우팅 기능에 결정권을 제공하는 포트로 특정된다. 완전한 적응형 라우팅을 다루는 경우, 이러한 포트는, 탈출 라우트를 제공하는 포트이다. 한편, output_port_1은 적응형 옵션으로 간주되고, 이러한 포트는, 적응형이 되는 능력을 라우팅 기능에 제공한다. 몇몇 경우들에서, output_port_0은, 존재하는 유일한 옵션일 수 있는 한편(즉, 오직 1차원으로서의 라우팅의 경우), 몇몇 다른 경우들에서는, output_port_0 및 output_port_1 둘 모두가 존재할 수 있다. 후자의 경우, 선택 기능은, 트래픽 조건들에 대해 다른 노드들로부터 수집된 정보에 기초하여, 모든 가능한 것 중 하나의 옵션만을 선택하는 것을 담당한다.
- [0041] 하나의 라우터 실시예는, 선택 기능을 수행하기 위해 CAAT(Content Aware Adaptive Throttle) 메커니즘을 구현하는 라우팅 프레임워크에 의존한다. 특히, 하나의 완전한 적응형 라우팅 알고리즘은, 적응형 출력 포트들의 이용을, 실시예에서 탈출 VC를 이미 이용한 그러한 메시지들로 제한한다. 또한, CAAT 메커니즘은, 현재 노드 상에 혼잡이 나타나는 경우 적응성의 이용을 제한한다. 이러한 방식으로, 혼잡 동안, 트래픽 대부분은 탈출 경로(결정형 라우트)를 따라서, 전체 네트워크에 걸친 혼잡의 신속한 확산 및 네트워크가 안정화되는 동안 유쾌하지 않은 성능 감소를 회피한다.
- [0042] 도 7은, 실시예에 따라 혼잡을 체크하기 위한 방법의 흐름도를 예시한다. 더 상세하게는, 도 7에 도시된 방법은, 도 6의 동작(606)을 수행하기 위해 이용될 수 있다. 이 방법은 사이클마다 라우팅 로직(150)의 각각의 출력 포트에 대해 반복될 수 있다. 예를 들어, 동시에, 이웃 라우터에서 이용가능한 성능 VC들의 넘버(#perfVCs) 및 공유 크레딧들의 넘버(#ShCred)에 대한 새로운 정보가 매 사이클마다 업데이트된다. 추가로, 혼잡을 검출하기 위해, CAAT는 다음 구성가능한 파라미터들 UPPER_PERFVC, UPPER_CREDIT, LOWER_PERFVC 및 LOWER_CREDIT에 의존한다.
- [0043] 동작(702)에서, 혼잡 플래그("congestion_flag")가 (예를 들어, 0 또는 구현에 따라 다른 값으로) 언셋(unset)된다. 몇몇 실시예들은 구성가능한 임계값들의 2개의 세트들을 이용한다. 각각의 세트는 자유로운 성능 VC들(UPPER_PERFVC 및 LOWER_PERFVC) 및 자유로운 공유 크레딧들(UPPER_CREDIT 및 LOWER_CREDIT)의 넘버로 구성된다. 예를 들어, 하위 임계치는 2개의 성능 VC들 및 2개의 자유로운 공유 크레딧들로 설정될 수 있는 한편, 상위 임계치는 5개의 성능 VC들 및 5개의 자유로운 공유 크레딧들로 설정될 수 있다. 이러한 접근법으로, 하위 파라미터들(LOWER_PERFVC, LOWER_CREDIT)의 값이 더 클수록, 라우팅 알고리즘의 동작은 더 결정적이 됨은 자명하다.

- [0044] 라우팅 로직(150) 상의 각각의 출력 포트에 대해, 동작(704)에서 포트 상의 자원들이 상위 임계치보다 크면(예를 들어, 성능 VC들의 넘버(#perfVCs)가 상위 성능 VC(UPPER_PERFVC)보다 크고, 공유 크레딧들의 넘버(#ShCred)가 상위 크레딧 값(UPPER_CREDIT)보다 큰 경우), 동작(706)에서 상태 비트(예를 들어, congestion_flag)가 클리어된다. 동작(708)에서, 포트 상의 자원이 하위 임계치와 동일하거나 그보다 작으면(예를 들어, 성능 VC들의 넘버(#perfVCs)가 하위 성능 VC(LOWER_PERFVC)보다 작거나 그와 동일하거나 또는 공유 크레딧들의 넘버(#ShCred)가 하위 크레딧 값(LOWER_CREDIT)보다 작거나 그와 동일한 경우), 동작(710)에서 상태 비트(예를 들어, congestion_flag)는, 출력 포트가 혼잡인 것을 나타내도록 설정된다. 동작(712)에서, 상태 비트(예를 들어, congestion_flag)가 리턴된다. 또한, 완전한 적응형 라우팅 알고리즘들은, 파라미터 LOWER_PERFVC가 0보다 큰 것을 보장함으로써 데드락이 없는 것을 보장한다. 그 다음, 이것은, 출력 포트 1이 탈출 라우트(라우팅 VC)를 차단하지 않는 것을 보장한다.
- [0045] 라우팅 로직(150)의 선택 기능(예를 들어, 도 6의 동작(608))은 도 8의 다음 흐름도에 도시된다. 도 8의 방법은, 출력 포트 1(예를 들어, 적응성을 제공하는 출력 포트를 지칭함)의 혼잡의 존재에 대해 체크하고, 검출되면, 결정형 포트를 승리자로 리턴하여, 라우트 상의 임의의 가능한 적응성을 무효화한다. 따라서, 도 8의 방법은, 혼잡 플래그의 값, 라우팅 알고리즘, 및 (예를 들어, 성능 VC들에 부여된 우선순위를 갖는) 출력 포트에서 이용가능한 VC의 타입에 기초하여 수행된다.
- [0046] 더 상세하게는, 동작(802)에서, 출력 포트 1이 유효로 설정되고, 출력 포트 1의 혼잡 플래그가 클리어된다. 동작(804)에서, 출력 포트 1의 혼잡 플래그에 대한 값을 검출한다. 동작(806)에서 혼잡이 존재하면, 동작(808)에서 출력 포트 1의 유효 비트/플래그가 클리어되고; 그렇지 않고 출력 포트 1이 동작(810)에서 유효이면, 출력 포트 1은 동작(812)에서 리턴된다. 출력 포트 1이 동작(810)에서 무효인 것으로 결정되면, 출력 포트 0이 동작(812)에서 출력 포트 1 대신에 리턴된다.
- [0047] 따라서, 본 명세서에서 설명되는 CAAT 메커니즘은, 완전한 적응형 라우팅에서 패킷 레이턴시 변화를 감소시키는 데 매우 효과적일 수 있다. 특히, 탈출 라우팅으로서 차원 순서 라우팅을 갖는 완전한 적응형 라우팅 알고리즘을 이용하는 경우, CAAT 메커니즘은, 네트워크가 혼잡한 경우 처리율을 허용가능한 레벨로 유지하면서 평균 패킷 레이턴시의 표준 편차의 변화를 효과적으로 감소시킬 수 있다. 따라서, CAAT는, 라우팅 기능이, 네트워크의 혼잡 조건들에 기초하여, 결정형 라우팅과 적응형 라우팅 사이에서 적절하게 스위칭하게 한다.
- [0048] 도 9는, 컴퓨팅 시스템(900)의 실시예에 대한 블록도를 예시한다. 도 1의 에이전트들(102) 중 하나 이상이 컴퓨팅 시스템(900)의 하나 이상의 구성요소들을 포함할 수 있다. 또한, 시스템(900)의 다양한 구성요소들은 도 9에 예시된 바와 같은 라우팅 로직(150)을 포함할 수 있다. 그러나, 로직(150)은, 예시된 것들을 포함하거나 배제하여, 시스템(900) 전반에 걸친 위치들에 제공될 수 있다. 컴퓨팅 시스템(900)은, 상호접속 네트워크(또는 버스)(904)에 커플링되는 하나 이상의 중앙 처리 장치(들)(CPU들)(902)(본 명세서에서 총괄적으로 "프로세서들(902)" 또는 더 일반적으로 "프로세서(902)로 지칭될 수 있음)를 포함할 수 있다. 프로세서들(902)은, 범용 프로세서, (컴퓨터 네트워크(905)를 통해 통신되는 데이터를 프로세싱할 수 있는) 네트워크 프로세서 등(RISC(reduced instruction set computer) 프로세서 또는 CISC(complex instruction set computer)를 포함함)과 같은 임의의 타입의 프로세서일 수 있다. 아울러, 프로세서들(902)은 단일 또는 다중 코어 설계를 가질 수 있다. 다중 코어 설계를 갖는 프로세서들(902)은 동일한 집적 회로(IC) 다이 상에 상이한 타입들의 프로세서 코어들을 집적할 수 있다. 또한, 다중 코어 설계를 갖는 프로세서들(902)은 대칭적 또는 비대칭적 멀티프로세서들로 구현될 수 있다.
- [0049] 프로세서(902)는, 다양한 실시예들에서 비밀 및/또는 공유될 수 있는 하나 이상의 캐시들을 포함할 수 있다. 일반적으로, 캐시는, 다른 곳에 저장되거나 더 먼저 컴퓨팅된 원래 데이터에 대응하는 데이터를 저장한다. 메모리 액세스 레이턴시를 감소시키기 위해, 데이터가 캐시에 저장되면, 장래의 이용은, 원래의 데이터를 프리페치(prefetch)하거나 재계산하기 보다는 캐시된 카피에 액세스함으로써 행해질 수 있다. 캐시(들)는, 시스템(900)의 하나 이상의 구성요소들에 의해 활용되는 전자 데이터(예를 들어, 명령들을 포함함)를 저장하기 위해, 임의의 타입의 캐시, 예를 들어, 레벨 1(L1) 캐시, 레벨 2(L2) 캐시, 레벨 3(L3), 중간-레벨 캐시, 최종 레벨 캐시(LLC) 등일 수 있다. 추가적으로, 이러한 캐시(들)는 다양한 위치들(예를 들어, 도 1, 도 2, 도 9 또는 도 10의 시스템들을 포함하는, 본 명세서에 논의된 컴퓨팅 시스템들에 대한 다른 구성요소들 내부)에 위치될 수 있다.
- [0050] 칩셋(906)이 추가적으로 상호접속 네트워크(904)에 커플링될 수 있다. 추가로, 칩셋(906)은 그래픽스 메모리 제어 허브(GMCH)(908)를 포함할 수 있다. GMCH(908)는, 메모리(912)에 커플링되는 메모리 제어기(910)를 포함

할 수 있다. 메모리(912)는, 예를 들어, 프로세서(902) 또는 컴퓨팅 시스템(900)의 구성요소들과 통신하는 임의의 다른 디바이스에 의해 실행되는 명령들의 시퀀스들을 포함하는 데이터를 저장할 수 있다. 또한, 본 발명의 일 실시예에서, 메모리(912)는 하나 이상의 휘발성 저장(또는 메모리) 디바이스, 예를 들어, 랜덤 액세스 메모리(RAM), 동적 RAM(DRAM), 동기식 DRAM(SDRAM), 정적 RAM(SRAM) 등을 포함할 수 있다. 하드 디스크와 같은 비휘발성 메모리가 또한 활용될 수 있다. 추가적인 디바이스들, 예를 들어, 다수의 프로세서들 및/또는 다수의 시스템 메모리들이 상호접속 네트워크(904)에 커플링될 수 있다.

[0051] GMCH(908)는, (예를 들어, 실시예에서 그래픽스 가속기를 통해) 디스플레이 디바이스(916)에 커플링되는 그래픽스 인터페이스(914)를 더 포함할 수 있다. 일 실시예에서, 그래픽스 인터페이스(914)는 가속된 그래픽스 포트(AGP)를 통해 디스플레이 디바이스(916)에 커플링될 수 있다. 본 발명의 실시예에서, 디스플레이 디바이스(916)(예를 들어, 평판 디스플레이)는, 예를 들어, 비디오 메모리 또는 시스템 메모리(예를 들어, 메모리(912))와 같은 저장 디바이스에 저장된 이미지의 디지털 표현을, 디스플레이(916)에 의해 해석 및 디스플레이되는 디스플레이 신호들로 전환하는 신호 변환기를 통해 그래픽스 인터페이스(914)에 커플링될 수 있다.

[0052] 도 9에 도시된 바와 같이, 허브 인터페이스(918)는 GMCH(908)를 입력/출력 제어 허브(ICH)(920)에 커플링할 수 있다. ICH(920)는 컴퓨팅 시스템(900)에 커플링된 입력/출력(I/O) 디바이스들에 대한 인터페이스를 제공할 수 있다. ICH(920)는 PCIe 규격을 준수할 수 있는 주변 구성요소 상호접속(PCI) 브릿지, 범용 직렬 버스(USB) 제어기 등과 같은 주변기기 브릿지(또는 제어기)(924)를 통해 버스(922)에 커플링될 수 있다. 브릿지(924)는 프로세서(902)와 주변 디바이스들 사이에 데이터 경로를 제공할 수 있다. 다른 타입들의 토폴로지들이 활용될 수 있다. 또한, 다수의 버스들이, 예를 들어, 다수의 브릿지들 또는 제어기들을 통해 ICH(920)에 커플링될 수 있다. 추가로, 버스(922)는 다른 타입들 및 구성들의 버스 시스템들을 포함할 수 있다. 아울러, ICH(920)에 커플링되는 다른 주변기기들은, 본 발명의 다양한 실시예들에서, 통합 드라이브 일렉트로닉스(IDE) 또는 소형 컴퓨터 시스템 인터페이스(SCSI) 하드 드라이브(들), USB 포트(들), 키보드, 마우스, 병렬 포트(들), 직렬 포트(들), 플로피 디스크 드라이브(들), 디지털 출력 지원부(예를 들어, 디지털 비디오 인터페이스(DVI)) 등을 포함할 수 있다.

[0053] 버스(922)는 오디오 디바이스(926), 하나 이상의 디스크 드라이브(들)(928) 및 네트워크 어댑터(930)(일 실시예에서는 NIC일 수 있음)에 커플링될 수 있다. 일 실시예에서, 네트워크 어댑터(930), 또는 버스(922)에 커플링되는 다른 디바이스들은 칩셋(906)과 통신할 수 있다. 또한, 다양한 구성요소들(예를 들어, 네트워크 어댑터(930))이 본 발명의 몇몇 실시예들에서 GMCH(908)에 커플링될 수 있다. 또한, 프로세서(902) 및 GMCH(908)는 단일 칩을 형성하도록 결합될 수 있다. 실시예에서, 메모리 제어기(910)는 CPU들(902) 중 하나 이상에 제공될 수 있다. 추가로, 실시예에서, GMCH(908) 및 ICH(920)는 주변기기 제어 허브(PCH)로 결합될 수 있다.

[0054] 추가적으로, 컴퓨팅 시스템(900)은 휘발성 및/또는 비휘발성 메모리(또는 스토리지)를 포함할 수 있다. 예를 들어, 비휘발성 메모리는, 하기의 것들, 즉, 판독 전용 메모리(ROM), 프로그래머블 ROM(PROM), 소거가능 PROM(EPROM), 전기적 EPROM(EEPROM), 디스크 드라이브(예를 들어, 928), 플로피 디스크, 콤팩트 디스크 ROM(CD-ROM), 디지털 다기능 디스크(DVD), 플래쉬 메모리, 자기-광학 디스크, 또는 전자 데이터(예를 들어, 명령들을 포함함)를 저장할 수 있는 다른 타입들의 비휘발성 머신-판독가능 매체 중 하나 이상을 포함할 수 있다.

[0055] 메모리(912)는 실시예에서, 하기의 것들, 즉, 운영 시스템(O/S)(932), 애플리케이션(934), 디렉토리(901), 및/또는 디바이스 드라이버(936) 중 하나 이상을 포함할 수 있다. 메모리(912)는 또한 메모리 맵핑된 I/O(MMIO) 동작들에 전용되는 영역들을 포함할 수 있다. 메모리(912)에 저장된 프로그램들 및/또는 데이터는 메모리 관리 동작들의 일부로서 디스크 드라이브(928)와 교환될 수 있다. 애플리케이션(들)(934)은, 네트워크(905)에 커플링된 하나 이상의 컴퓨팅 디바이스들과 하나 이상의 패킷들을 통신하도록 (예를 들어, 프로세서(들)(902) 상에서) 실행할 수 있다. 실시예에서, 패킷은, 적어도 하나의 전송기로부터 적어도 하나의 수신기로 (예를 들어, 네트워크(905)와 같은 네트워크를 통해) 송신되는 하나 이상의 전기 신호들에 의해 인코딩될 수 있는 하나 이상의 심볼들 및/또는 값들의 시퀀스일 수 있다. 예를 들어, 각각의 패킷은, 소스 어드레스, 수신지 어드레스, 패킷 타입 등과 같이, 패킷을 프로세싱 및/또는 라우팅하는데 활용될 수 있는 다양한 정보를 포함하는 헤더를 가질 수 있다. 각각의 패킷은 또한, 컴퓨터 네트워크(예를 들어, 네트워크(905))를 통해 다양한 컴퓨팅 디바이스들 사이에서 패킷이 전송하고 있는 미처리 데이터(또는 콘텐츠)를 포함하는 페이로드를 가질 수 있다.

[0056] 실시예에서, 애플리케이션(934)은, 예를 들어, 디바이스 드라이버(936)를 통해, 시스템(900)의 다양한 구성요소들과 통신하기 위해 O/S(932)를 활용할 수 있다. 따라서, 디바이스 드라이버(936)는, O/S(932)와 네트워크 어댑터(930) 또는 시스템(900)에 커플링된 다른 I/O 디바이스들 사이에서, 예를 들어, 칩셋(906)을 통해 통신 인

터페이스를 제공하기 위한 네트워크 어댑터(930) 특정 커맨드들을 포함할 수 있다.

- [0057] 실시예에서, O/S(932)는 네트워크 프로토콜 스택을 포함할 수 있다. 프로토콜 스택은 일반적으로, 네트워크(905)를 통해 전송되는 패킷들을 프로세싱하기 위해 실행될 수 있는 절차들 또는 프로그램들의 세트를 지칭하고, 여기서 패킷들은 특정 프로토콜에 부합할 수 있다. 예를 들어, TCP/IP(Transport Control Protocol/Internet Protocol) 패킷들은 TCP/IP 스택을 이용하여 프로세싱될 수 있다. 디바이스 드라이버(936)는, 예를 들어, 프로토콜 스택을 통해 프로세싱되는 메모리(912)의 버퍼들을 나타낼 수 있다.
- [0058] 네트워크(905)는 임의의 타입의 컴퓨터 네트워크를 포함할 수 있다. 네트워크 어댑터(930)는, 네트워크(905)를 통해 데이터를 송신 및/또는 수신하기 위해 이용가능한 (예를 들어, 메모리(912)에 저장된) 디스크립터들에 할당되는 (예를 들어, 메모리(912)에 저장된) 버퍼들에 패킷들을 기록하는 직접 메모리 액세스(DMA) 엔진을 더 포함할 수 있다. 추가적으로, 네트워크 어댑터(930)는, 어댑터 관련 동작들을 수행하기 위한 로직(예를 들어, 하나 이상의 프로그래머블 프로세서들)을 포함할 수 있는 네트워크 어댑터 제어를 포함할 수 있다. 실시예에서, 어댑터 제어기는 MAC(매체 액세스 제어) 구성요소일 수 있다. 네트워크 어댑터(930)는 임의의 타입의 휘발성/비휘발성 메모리(예를 들어, 메모리(912))를 참조하여 논의된 하나 이상의 캐시(들) 및/또는 다른 메모리 타입들을 포함함)와 같은 메모리를 더 포함할 수 있다.
- [0059] 도 10은, 본 발명의 실시예에 따라 포인트-투-포인트(PtP) 구성으로 배열되는 컴퓨팅 시스템(1000)을 예시한다. 특히, 도 10은, 프로세서들, 메모리 및 입력/출력 디바이스들이 다수의 포인트-투-포인트 인터페이스들에 의해 상호접속되는 시스템을 도시한다. 도 1 내지 도 9를 참조하여 논의된 동작들은 시스템(1000)의 하나 이상의 구성요소들에 의해 수행될 수 있다.
- [0060] 도 10에 예시된 바와 같이, 시스템(1000)은 몇몇 프로세서들을 포함할 수 있고, 명확화를 위해 이들 중 오직 2개의 프로세서들(1002 및 1004)만이 도시된다. 프로세서들(1002 및 1004) 각각은, 메모리들(1010 및 1012)과의 통신을 가능하게 하기 위한 로컬 메모리 제어기 허브(GMCH)(1006 및 1008)를 포함할 수 있다. 메모리들(1010 및/또는 1012)은 도 10의 메모리(1012)를 참조하여 논의되는 것들과 같은 다양한 데이터를 저장할 수 있다. 도 10에 도시된 바와 같이, 프로세서들(1002 및 1004)(또는 칩셋(1020), I/O 디바이스들(1043) 등과 같은 시스템(1000)의 다른 구성요소들)은 또한 도 1 내지 도 10을 참조하여 논의되는 바와 같은 하나 이상의 캐시(들)를 포함할 수 있다.
- [0061] 실시예에서, 프로세서들(1002 및 1004)은, 도 10을 참조하여 논의되는 프로세서들(1002) 중 하나일 수 있다. 프로세서들(1002 및 1004)은 PtP 인터페이스 회로들(1016 및 1018)을 각각 이용하여 포인트-투-포인트(PtP) 인터페이스(1014)를 통해 데이터를 교환할 수 있다. 또한, 프로세서들(1002 및 1004)은, 포인트-투-포인트 인터페이스 회로들(1026, 1028, 1030 및 1032)을 이용하여 개별적인 PtP 인터페이스들(1022 및 1024)을 통해 칩셋(1020)과 데이터를 각각 교환할 수 있다. 칩셋(1020)은, 예를 들어, PtP 인터페이스 회로(1037)를 이용하여, 고성능 그래픽스 인터페이스(1036)를 통해 고성능 그래픽스 회로(1034)와 데이터를 추가로 교환할 수 있다.
- [0062] 적어도 일 실시예에서, 디렉토리 캐시 및/또는 로직이 프로세서들(1002, 1004) 및/또는 칩셋(1020) 중 하나 이상에 제공될 수 있다. 그러나, 본 발명의 다른 실시예들은 도 10의 시스템(1000) 내의 다른 회로들, 로직 유닛들, 또는 디바이스들에 존재할 수 있다. 게다가, 본 발명의 다른 실시예들은, 도 10에 예시된 몇몇 회로들, 로직 유닛들 또는 디바이스들 전반에 걸쳐 분산될 수 있다. 예를 들어, 시스템(1000)의 다양한 구성요소들은 도 1의 로직(150)을 포함할 수 있다. 그러나, 로직(150)은, 예시된 것들을 포함하거나 배제하여 시스템(1000) 전반에 걸친 위치들에 제공될 수 있다.
- [0063] 칩셋(1020)은 PtP 인터페이스 회로(1041)를 이용하여 버스(1040)와 통신할 수 있다. 버스(1040)는, 버스 브릿지(1042) 및 I/O 디바이스들(1043)과 같이, 자신과 통신하는 하나 이상의 디바이스들을 가질 수 있다. 버스(1044)를 통해, 버스 브릿지(1042)는, 키보드/마우스(1045), 통신 디바이스들(1046)(예를 들어, 모뎀들, 네트워크 인터페이스 디바이스들, 또는 컴퓨터 네트워크(1005)와 통신할 수 있는 다른 통신 디바이스들), 오디오 I/O 디바이스, 및/또는 데이터 저장 디바이스(1048)와 같은 다른 디바이스들과 통신할 수 있다. 데이터 저장 디바이스(1048)는, 프로세서들(1002 및/또는 1004)에 의해 실행될 수 있는 코드(1049)를 저장할 수 있다.
- [0064] 하기 예들은 추가적인 실시예들과 관련된다. 예 1은, 입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트를 결정하기 위한 라우팅 로직을 포함하는 장치를 포함하고, 라우팅 로직은, 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트 및 제 2 출력 포트로부터 출력 포트를 선택하고, 제 1 출력 포트는, 착신 메시지에 대한 결정형 라우트를 제공하고, 제 2 출력

포트는 착신 메시지에 대한 적응형 라우트를 제공한다. 예 2는, 예 1의 장치를 포함하고, 결정형 라우트는, 소스 및 수신지 노드 쌍에 대해 동일한 경로를 선택할 수 있다. 예 3은, 예 1의 장치를 포함하고, 적응형 라우트는, 트래픽 정보에 기초하여, 소스 및 수신지 노드 쌍에 대해 상이한 경로를 선택할 수 있다. 예 4는, 예 1의 장치를 포함하고, 혼잡 정보는, 착신 메시지를 통신할 수 있는 성능 가상 채널들의 수와 하위 성능 가상 채널 임계치와의 비교에 기초하여 결정된다. 예 5는, 예 1의 장치를 포함하고, 혼잡 정보는, 착신 메시지를 통신할 수 있는 하나 이상의 가상 채널들에 대응하는 공유 크레딧들의 수와 하위 크레딧 임계치와의 비교에 기초하여 결정된다. 예 6은, 예 1의 장치를 포함하고, 라우팅 로직은, 착신 메시지의 제 1 플릿(flit)의 검출에 대한 응답으로 출력 포트를 결정한다. 예 7은, 예 1의 장치를 포함하고, 출력 포트의 이용가능성을 착신 메시지와 매칭시키기 위한 스케줄러 로직을 포함한다. 예 8은, 예 1의 장치를 포함하고, 입력 포트 및 출력 포트를 통신가능하게 커플링시키기 위한 크로스바 로직을 포함한다. 예 9는, 예 1의 장치를 포함하고, 착신 메시지에 대응하는 데이터를 플릿 단위로 저장하기 위해, 하나 이상의 버퍼들의 할당을 통해 데드락(deadlock) 없는 라우팅을 지원하기 위한 로직을 포함한다. 예 10은, 예 1의 장치를 포함하고, 착신 메시지에 대응하는 데이터를 패킷 레벨 단위로 통신하기 위해, 하나 이상의 가상 채널들의 할당을 통해 데드락 없는 라우팅을 지원하기 위한 로직을 포함한다. 예 11은, 예 10의 장치를 포함하고, 패킷은 복수의 플릿들을 포함한다. 예 12는, 예 1의 장치를 포함하고, 라우팅 로직은, 복수의 라우팅 가상 채널들에 기초하여 데드락 없는 라우팅을 제공한다. 예 13은, 예 1의 장치를 포함하고, 라우팅 로직은 제 1 에이전트를 링크를 통해 제 2 에이전트에 커플링시키고, 링크는 포인트-투-포인트 상호접속을 포함한다. 예 14는, 예 13의 장치를 포함하고, 제 1 에이전트 및 제 2 에이전트 중 하나 이상은 복수의 프로세서 코어들을 포함한다. 예 15는, 예 13의 장치를 포함하고, 제 1 에이전트 및 제 2 에이전트 중 하나 이상은 복수의 소켓들을 포함한다. 예 16은, 예 13의 장치를 포함하고, 제 1 에이전트, 제 2 에이전트, 라우팅 로직 및 메모리 중 하나 이상은 동일한 집적 회로 다이 상에 있다.

[0065] 예 17은, 라우팅 로직에서, 입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트를 결정하는 단계를 포함하는 방법을 포함하고, 라우팅 로직은, 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트 및 제 2 출력 포트로부터 출력 포트를 선택하고, 제 1 출력 포트는, 착신 메시지에 대한 결정형 라우트를 제공하고, 제 2 출력 포트는 착신 메시지에 대한 적응형 라우트를 제공한다. 예 18은, 예 17의 방법을 포함하고, 출력 포트의 이용가능성을 착신 메시지와 매칭시키는 단계를 더 포함한다. 예 19는, 예 17의 방법을 포함하고, 입력 포트 및 출력 포트를 통신가능하게 커플링시키는 단계를 더 포함한다. 예 20은, 예 17의 방법을 포함하고, 착신 메시지에 대응하는 데이터를 플릿 단위로 저장하기 위해, 하나 이상의 버퍼들의 할당을 통해 데드락 없는 라우팅을 지원하는 단계를 더 포함한다. 예 21은 예 17의 방법을 포함하고, 착신 메시지에 대응하는 데이터를 패킷 레벨 단위로 통신하기 위해, 하나 이상의 가상 채널들의 할당을 통해 데드락 없는 라우팅을 지원하는 단계를 더 포함한다.

[0066] 예 22는, 혼잡 정보를 저장하는 메모리; 및 메모리에 커플링되고, 입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트를 결정하는 라우팅 로직을 포함하는 시스템을 포함하고, 라우팅 로직은, 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트 및 제 2 출력 포트로부터 출력 포트를 선택하고, 제 1 출력 포트는, 착신 메시지에 대한 결정형 라우트를 제공하고, 제 2 출력 포트는 착신 메시지에 대한 적응형 라우트를 제공한다. 예 23은, 예 22의 시스템을 포함하고, 결정형 라우트는 소스 및 수신지 노드 쌍에 대해 동일한 경로를 선택할 수 있다. 예 24는, 예 22의 시스템을 포함하고, 적응형 라우트는, 트래픽 정보에 기초하여 소스 및 수신지 노드 쌍에 대해 상이한 경로를 선택할 수 있다.

[0067] 예 25는, 라우팅 로직에서, 입력 포트에서 수신되는 착신 메시지의 송신을 위한 출력 포트를 결정하기 위한 수단을 포함하는 장치를 포함하고, 라우팅 로직은, 라우팅 로직에 통신가능하게 커플링된 하나 이상의 다른 라우팅 로직에서 검출되는 혼잡 정보에 기초하여, 제 1 출력 포트 및 제 2 출력 포트로부터 출력 포트를 선택하고, 제 1 출력 포트는, 착신 메시지에 대한 결정형 라우트를 제공하고, 제 2 출력 포트는 착신 메시지에 대한 적응형 라우트를 제공한다. 예 26은, 예 25의 장치를 포함하고, 출력 포트의 이용가능성을 착신 메시지와 매칭시키기 위한 수단을 더 포함한다. 예 27은, 예 25의 장치를 포함하고, 입력 포트 및 출력 포트를 통신가능하게 커플링시키기 위한 수단을 더 포함한다. 예 28은, 예 25의 장치를 포함하고, 착신 메시지에 대응하는 데이터를 플릿 단위로 저장하기 위해, 하나 이상의 버퍼들의 할당을 통해 데드락 없는 라우팅을 지원하기 위한 수단을 더 포함한다. 예 29는, 예 25의 장치를 포함하고, 착신 메시지에 대응하는 데이터를 패킷 레벨 단위로 통신하기 위해, 하나 이상의 가상 채널들의 할당을 통해 데드락 없는 라우팅을 지원하기 위한 수단을 더 포함한다.

[0068] 예 30은, 프로세서 상에서 실행되는 경우, 예 17 내지 예 21 중 어느 하나의 하나 이상의 동작들을 수행하도록

프로세서를 구성하는 하나 이상의 명령들을 포함하는 컴퓨터 판독가능 매체를 포함한다. 예 31은, 예 17 및 예 18 중 어느 하나의 방법을 포함하고, 입력 포트 및 출력 포트를 통신가능하게 커플링하는 단계를 더 포함한다. 예 32는, 예 17 내지 예 19 중 어느 하나의 방법을 포함하고, 착신 메시지에 대응하는 데이터를 플릿 단위로 저장하기 위해, 하나 이상의 버퍼들의 할당을 통해 데드락 없는 라우팅을 지원하는 단계를 더 포함한다. 예 33은, 예 17 내지 예 20 중 어느 하나의 방법을 포함하고, 착신 메시지에 대응하는 데이터를 패킷 레벨 단위로 통신하기 위해, 하나 이상의 가상 채널들의 할당을 통해 데드락 없는 라우팅을 지원하는 단계를 더 포함한다.

[0069] 본 발명의 다양한 실시예들에서, 예를 들어, 도 1 내지 도 10을 참조하여 본 명세서에서 논의된 동작들은 하드웨어(예를 들어, 회로), 소프트웨어, 펌웨어, 마이크로코드 또는 이들의 조합으로 구현될 수 있고, 이들은, 예를 들어, 본 명세서에 논의된 프로세스를 수행하도록 컴퓨터를 프로그래밍하기 위해 이용되는 명령들(또는 소프트웨어 절차들)이 저장된 (예를 들어, 비일시적) 머신 판독가능 또는 (예를 들어, 비일시적) 컴퓨터 판독가능 매체를 포함하는 컴퓨터 프로그램 물건으로 제공될 수 있다. 또한, 용어 "로직은, 예를 들어, 소프트웨어, 하드웨어, 또는 소프트웨어와 하드웨어의 결합들을 포함할 수 있다. 머신 판독가능 매체는 도 1 내지 도 10에 대해 논의된 것들과 같은 저장 디바이스를 포함할 수 있다. 추가적으로, 이러한 컴퓨터 판독가능 매체는 컴퓨터 프로그램 물건으로서 다운로드될 수 있고, 프로그램은, 통신 링크(예를 들어, 버스, 모뎀, 또는 네트워크 접속)를 통해, 반송파 또는 다른 전파 매체의 데이터 신호들을 통해 원격 컴퓨터(예를 들어, 서버)로부터 요청 컴퓨터(예를 들어, 클라이언트)에 전송될 수 있다.

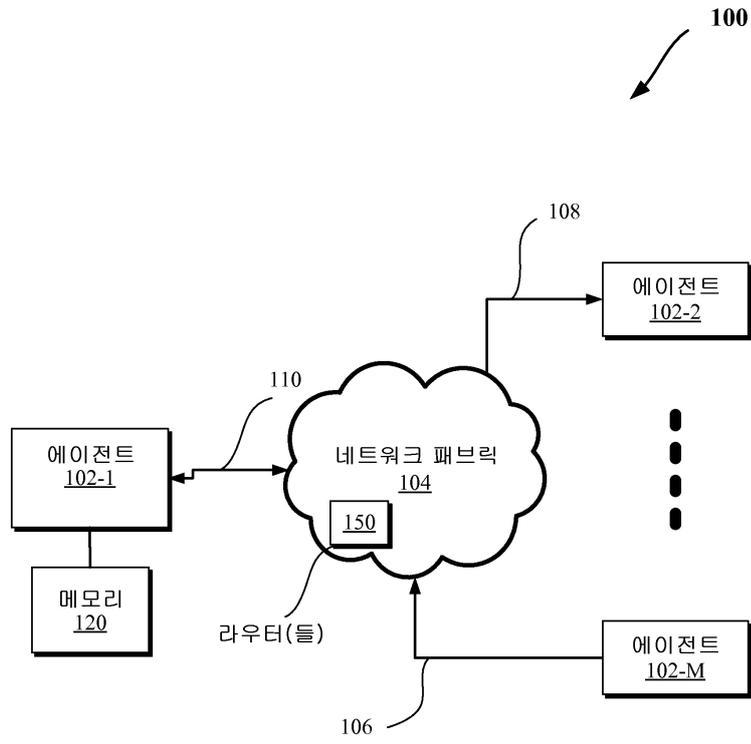
[0070] 본 명세서에서 "일 실시예" 또는 "실시예"의 참조는, 실시예와 관련되어 설명되는 특정한 특징들, 구조 또는 특성이 적어도 구현에 포함될 수 있음을 의미한다. 본 명세서의 다양한 위치들에서 "일 실시예에서"라는 구문의 등장은 모두 동일한 실시예를 지칭할 수도 있고 그렇지 않을 수도 있다.

[0071] 또한, 설명 및 청구항들에서, 용어들 "커플링된" 및 "접속된"은, 이들의 파생어들과 함께 이용될 수 있다. 본 발명의 몇몇 실시예들에서, "접속된"은, 둘 이상의 요소들이 서로 직접 물리적으로 또는 전기적으로 접촉하는 것을 나타내기 위해 이용될 수 있다. "커플링된"은, 또한 둘 이상의 요소들이 직접 물리적으로 또는 전기적으로 접촉하는 것을 의미할 수 있다. 그러나, "커플링된"은, 둘 이상의 요소들이 서로 직접 접촉하지 않을 수 있지만 여전히 서로 협력 또는 상호작용할 수 있는 것을 의미할 수 있다.

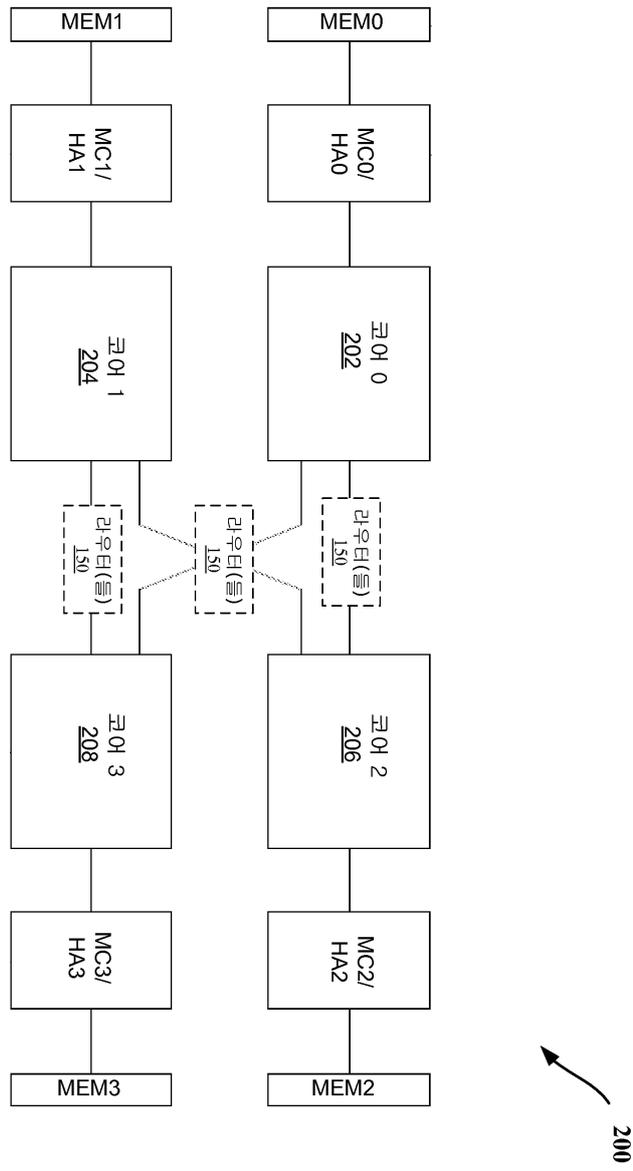
[0072] 따라서, 본 발명의 실시예들은 구조적 특징들 및/또는 방법론적 동작들에 특정된 언어로 설명되었지만, 청구된 요지는, 특정한 특징들 또는 설명된 동작들에 제한되지 않을 수 있음을 이해해야 한다. 오히려, 특정한 특징들 및 동작들은, 청구된 요지를 구현하는 예시적인 형태들로 개시된다.

도면

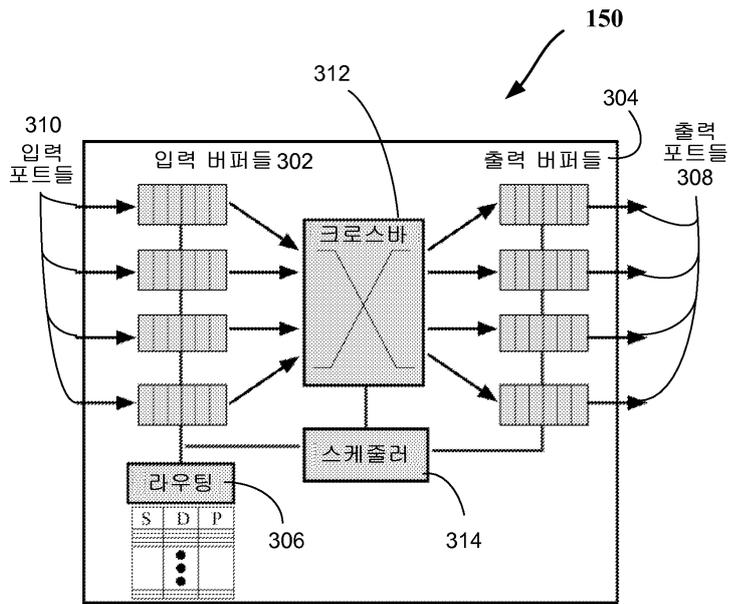
도면1



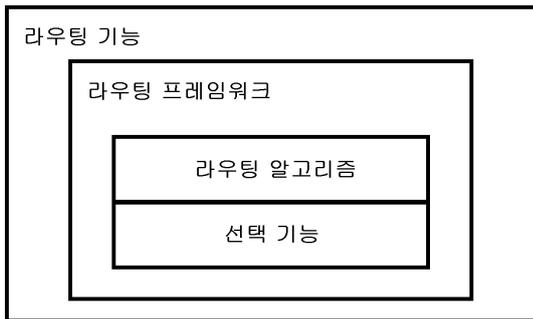
도면2



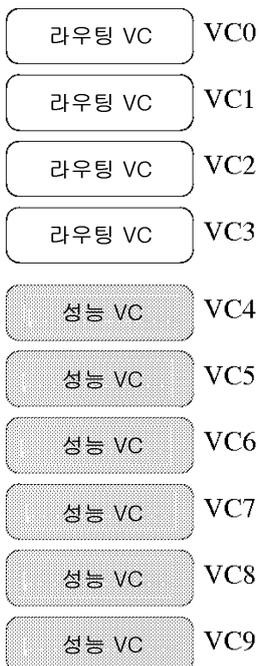
도면3



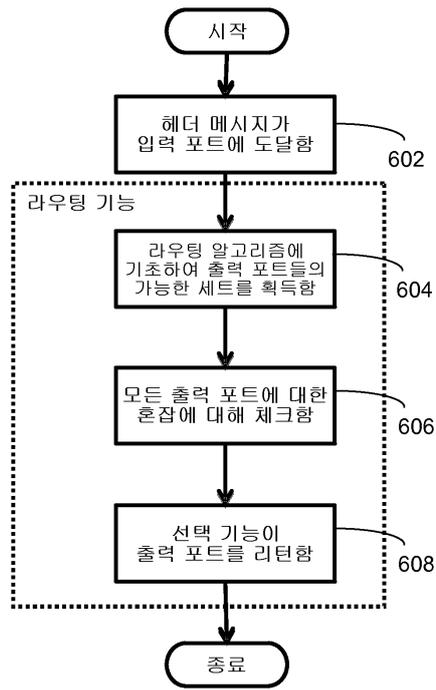
도면4



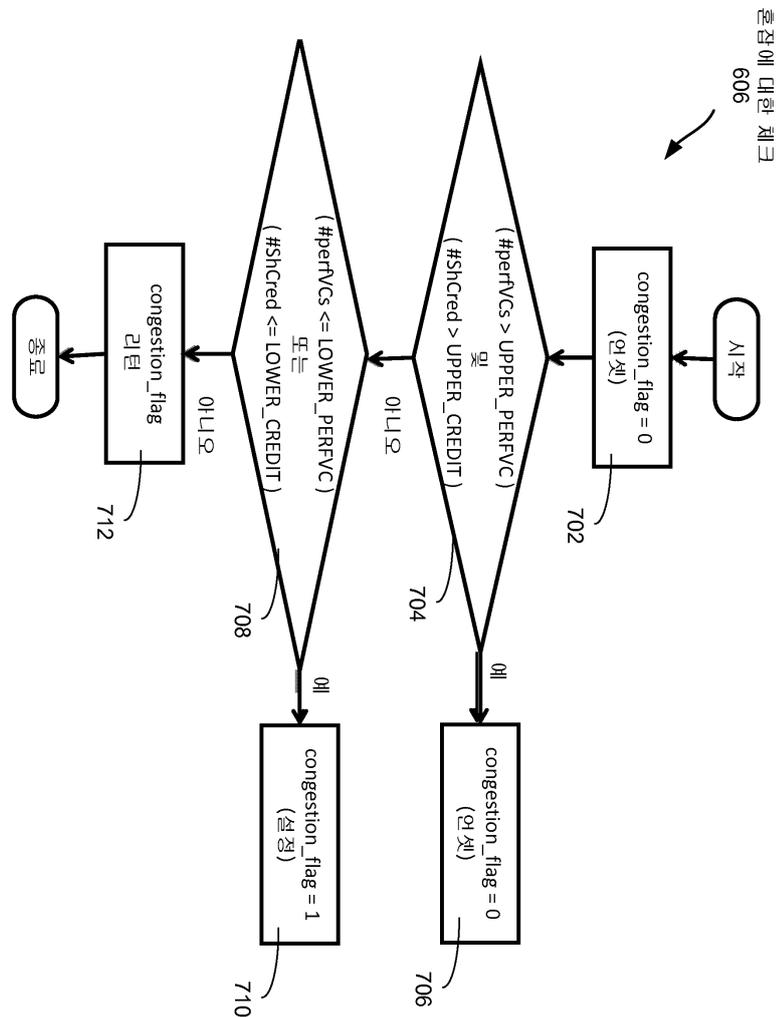
도면5



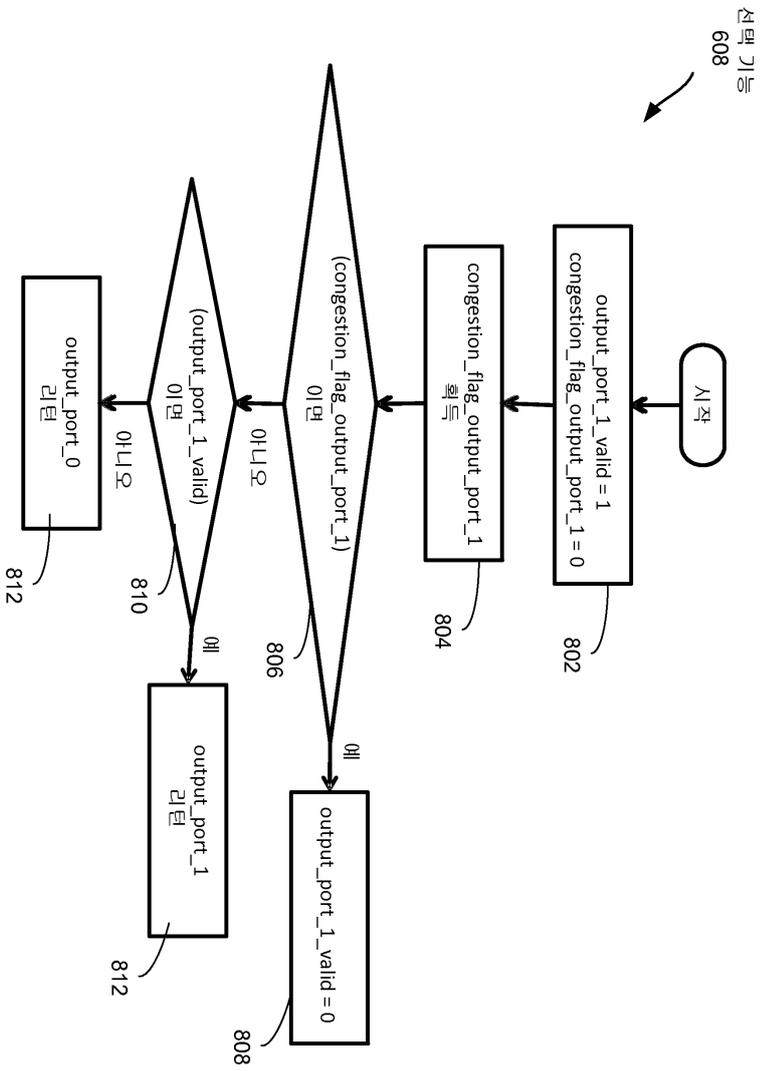
도면6



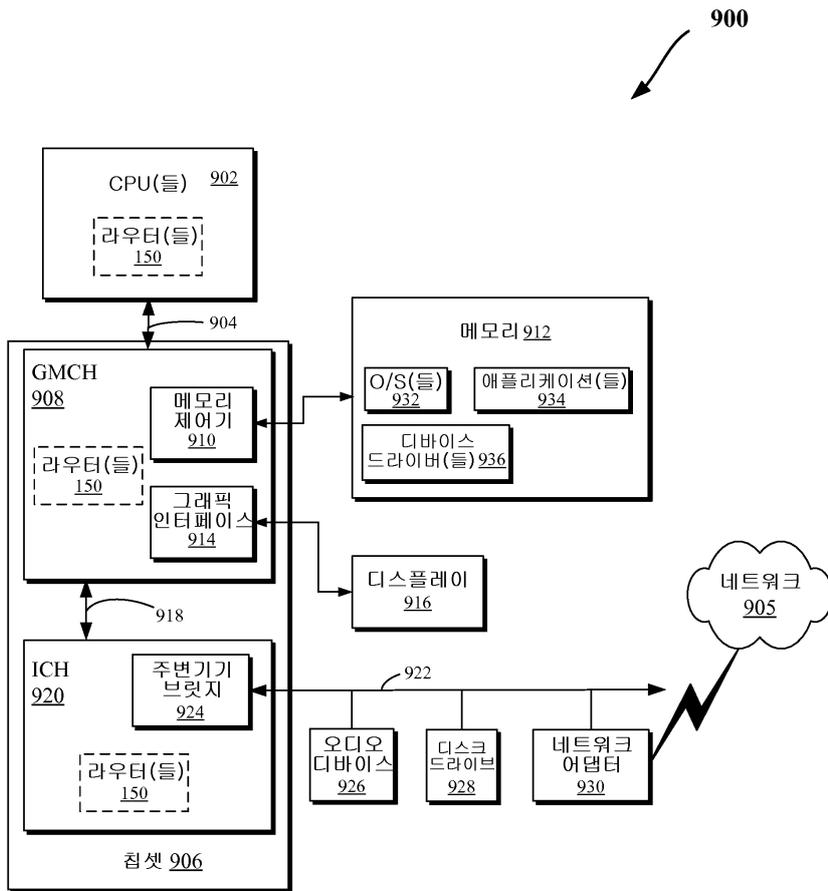
도면7



도면8



도면9



도면10

