(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년07월15일
(11) 등록번호 10-2684511
(24) 등록일자 2024년07월09일

(51) 국제특허분류(Int. Cl.)
G06F 9/28 (2017.01) *G06F 9/50* (2018.01)
(52) CPC특허분류
G06F 9/28 (2013.01)
G06F 9/3846 (2013.01)
(21) 출원번호 10-2020-0087436
(22) 출원일자 2020년07월15일
심사청구일자 2022년06월15일
(65) 공개번호 10-2021-0021263
(43) 공개일자 2021년02월25일
(30) 우선권주장
16/542,012 2019년08월15일 미국(US)
(56) 선행기술조사문헌
KR1020190044572 A

(73) 특허권자
인텔 코포레이션
미합중국 캘리포니아 95054 산타클라라 미션 칼리지 블러바드 2200
(72) 발명자
베하르, 마이클
이스라엘 30900 에이치에이 지크론 야코브 벤-구리온 스트리트 40
마오르, 모쉬
이스라엘 26360 제트 키르얏 모즈킹 하라브-룩 스트리트 24
(뒷면에 계속)
(74) 대리인
양영준, 김연송, 백만기

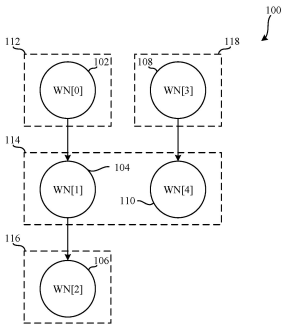
전체 청구항 수 : 총 13 항 심사관 : 지정훈

(54) 발명의 명칭 **작업부하의 정적 매핑의 비순차적 파이프라이닝된 실행을 가능하게 하기 위한 방법들 및 장치**

(57) 요 약

가속기의 하나 이상의 계산 빌딩 블록으로의 작업부하의 정적 매핑의 비순차적 파이프라이닝된 실행을 가능하게 하는 방법들, 장치, 시스템들 및 제조 물품들이 개시된다. 예시적인 장치는 제1 수의 크레디트들을 메모리에 로딩하기 위한 인터페이스; 크레디트들의 제1 수를 버퍼에서의 메모리 가용성과 연관된 크레디트들의 임계 수와 비교하기 위한 비교기; 및 크레디트들의 제1 수가 크레디트들의 임계 수를 충족시킬 때, 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 작업부하의 작업부하 노드를 선택하기 위한 디스패처를 포함한다.

대 표 도



(52) CPC특허분류
G06F 9/5066 (2013.01)
G06F 9/5077 (2013.01)
G06F 9/5083 (2013.01)
(72) 발명자
가바이, 로넨
이스라엘 1923800 라맛 하쇼펫 라맛 하쇼펫
로스너, 로니
이스라엘 3056922 아이에스 비냐미나 하-야인 25
왈터, 지기
이스라엘 2627335 에이치에이 하이파 하-티나 1 에이피피.2
아감, 오렌
이스라엘 30900 제트 지크론 야코브 바락 6/6

명 세 서

청구범위

청구항 1

장치로서,

제1 로컬 크레디트 매니저를 포함하는 제1 계산 유닛- 상기 제1 계산 유닛은 상기 제1 계산 유닛이 데이터를 기입하는 제1 버퍼와 연관됨 -;

제2 로컬 크레디트 매니저를 포함하는 제2 계산 유닛- 상기 제2 계산 유닛은 상기 제2 계산 유닛이 데이터를 판독하는 제2 버퍼와 연관됨 -;

상기 제1 계산 유닛 및 상기 제2 계산 유닛에 결합된 적어도 하나의 패브릭; 및

상기 적어도 하나의 패브릭에 결합된 중앙 크레디트 매니저를 포함하고, 상기 중앙 크레디트 매니저는,

상기 제1 로컬 크레디트 매니저로의 제1 크레디트의 송신을 야기하고- 상기 제1 크레디트는 상기 제1 버퍼에 저장될 제2 데이터를 생성하기 위해 상기 제1 계산 유닛에 의해 처리될 제1 데이터에 대응함 -;

상기 제1 계산 유닛의 상기 제1 로컬 크레디트 매니저로부터의 상기 제1 크레디트에 액세스하고;

상기 제2 계산 유닛에 대한 크레디트들의 수를 감소시키는, 장치.

청구항 2

제1항에 있어서, 상기 중앙 크레디트 매니저는 상기 제1 계산 유닛이 상기 제1 데이터를 처리하는 것에 응답하여 상기 제1 계산 유닛의 상기 제1 로컬 크레디트 매니저로부터의 상기 제1 크레디트에 액세스하는, 장치.

청구항 3

제1항에 있어서, 상기 중앙 크레디트 매니저는 상기 제2 버퍼에서의 상기 제2 데이터의 가용성에 응답하여 상기 제2 계산 유닛에 대한 상기 크레디트들의 수를 감소시키는, 장치.

청구항 4

제1항에 있어서, 상기 제2 계산 유닛에 대한 상기 크레디트들의 수는 크레디트들의 제1 수이고, 상기 중앙 크레디트 매니저는,

상기 제1 계산 유닛에 대한 크레디트들의 제2 수를 초기화하고;

상기 제2 계산 유닛에 대한 상기 크레디트들의 제1 수를 초기화하는, 장치.

청구항 5

제1항에 있어서, 상기 중앙 크레디트 매니저는 상기 제1 계산 유닛에 할당된 작업과 연관되는 상기 제1 데이터에 기초하여 상기 제1 로컬 크레디트 매니저로의 상기 제1 크레디트의 송신을 야기하는, 장치.

청구항 6

방법으로서,

프로세서 회로로 명령어를 실행함으로써, 제1 계산 유닛의 제1 로컬 크레디트 매니저에 제1 크레디트를 송신하는 단계- 상기 제1 크레디트는 상기 제1 계산 유닛과 연관된 제1 버퍼에 저장될 제2 데이터를 생성하기 위해 상기 제1 계산 유닛에 의해 처리될 제1 데이터에 대응하고, 상기 제1 계산 유닛은 상기 제1 버퍼에 데이터를 기입함 -;

상기 프로세서 회로로 명령어를 실행함으로써, 상기 제1 계산 유닛의 상기 제1 로컬 크레디트 매니저로부터의 상기 제1 크레디트에 액세스하는 단계; 및

상기 프로세서 회로로 명령어를 실행함으로써, 제2 로컬 크레디트 매니저를 포함하는 제2 계산 유닛에 대한 크

레디트들의 수를 감소시키는 단계를 포함하고, 상기 제2 계산 유닛은 상기 제2 계산이 데이터를 판독하는 제2 버퍼와 연관되는, 방법.

청구항 7

제6항에 있어서, 상기 제1 계산 유닛이 상기 제1 데이터를 처리하는 것에 응답하여 상기 제1 계산 유닛의 상기 제1 로컬 크레디트 매니저로부터의 상기 제1 크레디트에 액세스하는 단계를 더 포함하는, 방법.

청구항 8

제6항에 있어서, 상기 제2 버퍼에서의 상기 제2 데이터의 가용성에 응답하여 상기 제2 계산 유닛에 대한 상기 크레디트들의 수를 감소시키는 단계를 더 포함하는, 방법.

청구항 9

제6항에 있어서, 상기 제2 계산 유닛에 대한 상기 크레디트들의 수는 크레디트들의 제1 수이고, 상기 방법은 상기 제1 계산 유닛에 대한 크레디트들의 제2 수 및 상기 제2 계산 유닛에 대한 상기 크레디트들의 제1 수를 초기화하는 단계를 더 포함하는, 방법.

청구항 10

제6항에 있어서, 상기 제1 계산 유닛에 할당된 작업과 연관되는 상기 제1 데이터에 기초하여 상기 제1 로컬 크레디트 매니저에 상기 제1 크레디트를 송신하는 단계를 더 포함하는, 방법.

청구항 11

장치로서,

메모리;

명령어들; 및

상기 명령어들을 실행하여 제6항 내지 제10항 중 어느 한 항의 방법을 수행하는 프로세서 회로를 포함하는, 장치.

청구항 12

실행될 때 프로세서 회로로 하여금 제6항 내지 제10항 중 어느 한 항의 방법을 수행하게 하는 명령어를 포함하는 비일시적 컴퓨터 판독가능한 매체.

청구항 13

제6항 내지 제10항 중 어느 한 항의 방법을 수행하는 수단을 포함하는 장치.

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

발명의 설명

기 술 분 야

[0001] 본 개시는 일반적으로 처리에 관한 것이며, 더 구체적으로 작업부하의 정적 매핑의 비순차적 파이프라이닝된 실행을 가능하게 하는 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 컴퓨터 하드웨어 제조자들은 컴퓨터 플랫폼의 다양한 컴포넌트들에서 사용하기 위한 하드웨어 컴포넌트들을 개발한다. 예를 들어, 컴퓨터 하드웨어 제조자들은 마더보드들, 마더보드들을 위한 칩셋들, 중앙 처리 유닛들(CPU들), 하드 디스크 드라이브들(HDD들), 솔리드 스테이트 드라이브들(SSD들), 및 다른 컴퓨터 컴포넌트들을 개발한다. 추가적으로, 컴퓨터 하드웨어 제조자들은 가속기들로서 알려진 처리 요소들을 개발하여 작업부하의 처리를 가속화시킨다. 예를 들어, 가속기는 CPU, 그래픽 처리 유닛들(GPU), 비전 처리 유닛들(VPU), 및/또는 필드 프로그래머블 게이트 어레이들(FPGA)일 수 있다.

도면의 간단한 설명

[0003] 도 1은 이종 시스템의 가속기 상에서 실행하는 작업부하를 나타내는 그래프의 그래픽 예시이다.

도 2는 버퍼들 및 파이프라이닝을 구현하는 이종 시스템의 가속기 상에서 실행하는 작업부하를 나타내는 그래프의 그래픽 예시이다.

도 3은 본 개시의 교시에 따라 구성된 예시적인 컴퓨팅 시스템을 나타내는 블록도이다.

도 4는 예시적인 하나 이상의 스케줄러를 포함하는 예시적인 컴퓨팅 시스템을 나타내는 블록도이다.

도 5는 도 3 및 도 4의 스케줄러들 중 하나 이상을 구현할 수 있는 예시적인 스케줄러의 블록도이다.

도 6은 도 5의 버퍼 크레디트 스토리지의 추가적인 상세를 도시하는 예시적인 스케줄러의 블록도이다.

도 7은 버퍼들 및 파이프라이닝을 구현하는 이종 시스템의 가속기 상에서 실행하는 작업부하를 나타내는 예시적인 그래프의 그래픽 예시이다.

도 8은 도 5의 스케줄러 및/또는 도 6의 스케줄러를 구현하기 위해 실행될 수 있는 머신 판독가능 명령어들에

의해 구현될 수 있는 프로세스를 나타내는 흐름도이다.

도 9는 도 5의 스케줄러 및/또는 도 6의 스케줄러의 인스턴스화들 중 하나 이상을 구현하기 위해 도 8의 명령어들을 실행하도록 구성된 예시적인 프로세서 플랫폼의 블록도이다.

도면들은 축척에 맞지 않는다. 일반적으로, 동일한 또는 유사한 부분들을 지칭하기 위해 도면(들) 및 첨부한 기입된 설명을 통해 동일한 참조 번호들이 사용될 것이다. 연결 참조들(예를 들어, 부착, 결합, 연결, 및 접합)은 광범위하게 해석되어야 하고, 달리 명시되지 않는 한 요소들의 집합 사이의 중간 부재들 및 요소들 사이의 상대 이동이 포함될 수 있다. 이와 같이, 연결 참조들은 2개의 요소들이 직접 연결되고 서로 고정 관계에 있는 것을 반드시 추론하는 것은 아니다.

서술어들 "제1", "제2", "제3" 등은 개별적으로 지칭될 수 있는 다수의 요소 또는 컴포넌트들을 식별할 때 본 명세서에서 사용된다. 그들의 사용 맥락에 기초하여 달리 특정되거나 이해되지 않는 한, 이러한 서술어들은 리스트 내의 우선순위, 물리적 순서 또는 배열의 임의의 의미, 또는 시간 상의 순서화를 암시하도록 의도되지 않고, 개시된 예들을 이해하기 위한 용이성을 위해 개별적으로 다수의 요소 또는 컴포넌트들을 참조하기 위한 라벨들로서 단지 사용된다. 일부 예들에서, 서술어 "제1"은 상세한 설명에서의 요소를 지칭하기 위해 사용될 수 있는 한편, 동일한 요소가 "제2" 또는 "제3"과 같은 상이한 서술어로 청구항에서 지칭될 수 있다. 이러한 경우들에서, 이러한 서술어들은 단지 다수의 요소 또는 컴포넌트들을 참조하기 위해 사용되는 것으로 이해되어야 한다.

발명을 실시하기 위한 구체적인 내용

[0004] 많은 컴퓨터 하드웨어 제조자들은 작업부하의 처리를 가속화하기 위해 가속기들로서 알려진 처리 요소들을 개발한다. 예를 들어, 가속기는 중앙 처리 유닛(CPU), 그래픽 처리 유닛(GPU), 비전 처리 유닛(VPU), 및/또는 필드 프로그래머블 게이트 어레이(FPGA)일 수 있다. 더욱이, 가속기들은 한편, 특정 타입들의 작업부하들을 최적화하도록 설계된 임의의 타입의 작업부하를 처리할 수 있다. 예를 들어, CPU들 및 FPGA들은 더 일반적인 처리를 처리하도록 설계될 수 있지만, GPU들은 비디오, 게임들, 및/또는 다른 물리학들 및 수학적 기반의 계산들의 처리를 개선하도록 설계될 수 있고, VPU들은 머신 비전 작업들의 처리를 개선하도록 설계될 수 있다.

[0005] 추가적으로, 일부 가속기들은 AI(artificial intelligence) 애플리케이션들의 처리를 개선하도록 특별히 설계된다. VPU가 특정 타입의 AI 가속기이지만, 많은 상이한 AI 가속기들이 사용될 수 있다. 사실 상, 많은 AI 가속기들은 ASIC(application specific integrated circuit)들에 의해 구현될 수 있다. 이러한 ASIC-기반 AI 가속기들은 머신 학습(ML), 딥 러닝(DL), 및/또는 지원 벡터 머신(SVM)들, 신경 네트워크(NN)들, 회귀 신경 네트워크(RNN)들, 콘볼루션 신경 네트워크(CNN)들, 롱 쇼트 텀 메모리(LSTM), 게이트 회귀 유닛(GRU)들 등을 포함하는 다른 인공 머신 구동 로직과 같은, 특정 타입의 AI에 관련된 작업들의 처리를 개선하도록 설계될 수 있다.

[0006] 컴퓨터 하드웨어 제조자들은 또한 하나보다 많은 타입의 처리 요소를 포함하는 이종 시스템들을 개발한다. 예를 들어, 컴퓨터 하드웨어 제조자들은 CPU들과 같은 범용 처리 요소들과 FPGA들과 같은 범용 가속기들, 및/또는 GPU들, VPU들, 및/또는 다른 AI 가속기들과 같은 더 맞춤화된 가속기들 양자 모두와 조합할 수 있다. 이러한 이종 시스템들은 SoC(systems on a chip)들로서 구현될 수 있다.

[0007] 개발자가 이종 시스템 상에서 함수, 알고리즘, 프로그램, 애플리케이션, 및/또는 다른 코드를 실행하기를 원할 때, 개발자 및/또는 소프트웨어는 컴파일 시간에 함수, 알고리즘, 프로그램, 애플리케이션, 및/또는 다른 코드에 대한 스케줄을 생성한다. 스케줄이 생성되면, 스케줄은 (어헤드 오브 타임(Ahead of Time) 패러다임 또는 저스트 인 타임(Just in Time) 패러다임을 위한) 실행가능 파일을 생성하기 위해 함수, 알고리즘, 프로그램, 애플리케이션, 및/또는 다른 코드 사양서와 조합된다. 더욱이, 함수, 알고리즘, 프로그램, 애플리케이션, 및/또는 다른 코드는 노드들을 포함하는 그래프로서 표현될 수 있으며, 그래프는 작업부하를 나타내고 각각의 노드는 그 작업부하의 특정 작업을 나타낸다. 또한, 그래프 내의 상이한 노드들 사이의 연결들은 특정 노드가 실행되도록 하기 위해 필요한 데이터 입력들 및/또는 출력들을 나타내고 그래프의 정점들은 그래프의 노드들 사이의 데이터 종속성들을 나타낸다.

[0008] 실행가능 파일은 다수의 상이한 실행가능 섹션을 포함하고, 여기서 각각의 실행가능 섹션은 특정 처리 요소(예를 들어, CPU, GPU, VPU, 및/또는 FPGA)에 의해 실행가능하다. 실행가능 파일의 각각의 실행가능 섹션은 실행가능 서브-섹션들을 추가로 포함할 수 있고, 여기서 각각의 실행가능 서브-섹션은 특정한 처리 요소의 계산 빌딩 블록(CBB)들에 의해 실행가능하다. 추가적으로 또는 대안적으로, 본 명세서에 개시된 일부 예들에서, 개발자 및/또는 소프트웨어 개발 소프트웨어는 실행파일의 성공적인 실행을 결정하기 위한 기준(예를 들어, 성공 기

준(success criteria))을 정의할 수 있다. 예를 들어, 이러한 성공 기준은 이종 시스템 및/또는 특정 처리 요소의 활용의 임계값을 충족시키고 및/또는 다르게는 만족시키기 위해 실행파일을 실행하는 것에 대응할 수 있다. 다른 예들에서, 성공 기준은 실행파일을 시간의 임계량 내에 실행가능한 것에 대응할 수 있다. 그러나, 이종 시스템 및/또는 특정 처리 요소 상에서 실행파일을 어떻게 실행할지를 결정할 때 임의의 적절한 성공 함수가 이용될 수 있다. 이러한 방식으로, 성공 기준은 개발자, 소프트웨어, 및/또는 인공 지능 시스템이 성공 기준을 충족시키도록 최적화된 스케줄을 포함하는 실행파일을 생성하는 데 유익할 수 있다.

[0009] 도 1은 이종 시스템의 가속기 상에서 실행되는 작업부하를 나타내는 그래프(100)의 그래픽 예시이다. 그래프(100)는 제1 작업부하 노드(102)(WN[0]), 제2 작업부하 노드(104)(WN[1]), 제3 작업부하 노드(106)(WN[2]), 제4 작업부하 노드(108)(WN[3]), 및 제5 작업부하 노드(110)(WN[4])를 포함한다. 도 1에서, 가속기는 정적 소프트웨어 스케줄을 통해 그래프(100)에 의해 표현되는 작업부하를 실행하고 있다. 정적 소프트웨어 스케줄링은 가속기의 계산 빌딩 블록(CBB)들 상에서 그래프(100)의 상이한 작업부하 노드들을 실행하기 위한 미리 정의된 방식을 결정하는 것을 포함한다. 예를 들어, 정적 소프트웨어 스케줄은 제1 작업부하 노드(102)(WN[0])를 제1 CBB(112)에, 제2 작업부하 노드(104)(WN[1])를 제2 CBB(114)에, 제3 작업부하 노드(106)(WN[2])를 제3 CBB(116)에, 제4 작업부하 노드(108)(WN[3])를 제4 CBB(118)에, 그리고 제5 작업부하 노드(110)(WN[4])를 제2 CBB(114)에 할당한다.

[0010] 도 1에서, 정적 소프트웨어 스케줄은, 제1 작업부하 노드(102)(WN[0])가 제4 CBB(118) 상에서 실행하는 제4 작업부하 노드(108)(WN[3])와 병렬로 제1 CBB(112) 상에서 실행하는 것임을 약술한다. 도 1에서, 제4 CBB(118)가 제4 작업부하 노드(108)(WN[3])를 실행하는 것은, 제1 CBB(112)가 제1 작업부하 노드(102)(WN[0])를 실행하는 것보다 더 빠르다. 정적 소프트웨어 스케줄이 제2 CBB(114)가 제5 작업부하 노드(110)(WN[4])를 실행하는 것 전에 제2 CBB(114)가 제2 작업부하 노드(104)(WN[1])를 실행하는 것을 약술하기 때문에, 제1 CBB(112)가 제1 작업부하 노드(102)(WN[0])의 실행을 완료할 때까지 제2 CBB(114)는 유휴 상태이다. 또한, 작업부하 노드들이 후속 작업부하 노드들을 실행하기 전에 완전히 실행될 때까지 대기하는 것은 상당한 메모리 오버헤드를 요구하는데, 그 이유는 제1 작업부하 노드(예를 들어, 제1 작업부하 노드(102)(WN[0]))를 실행하는 CBB에 의해 생산되는 데이터가, CBB가 제2 작업부하 노드(예를 들어, 제2 작업부하 노드(104)(WN[1]))를 실행할 수 있기 전에 가속기 상에 저장될 필요가 있기 때문이다.

[0011] 도 2는 버퍼들 및 파이프라이닝을 구현하는 이종 시스템의 가속기 상에서 실행하는 작업부하를 나타내는 그래프(200)의 그래픽 예시이다. 그래프(200)는 제1 작업부하 노드(102)(WN[0]), 제2 작업부하 노드(104)(WN[1]), 제3 작업부하 노드(106)(WN[2]), 제4 작업부하 노드(108)(WN[3]), 및 제5 작업부하 노드(110)(WN[4])를 포함한다. 도 2에서, 가속기는 정적 소프트웨어 스케줄을 통해 그래프(200)에 의해 표현되는 작업부하를 실행하고 있다. 도 2의 정적 소프트웨어 스케줄은 파이프라이닝을 구현하고 제1 버퍼(202), 제2 버퍼(204), 및 제3 버퍼(206)를 포함하는 가속기의 CBB들 상의 그래프(200)의 상이한 작업부하 노드들에 대한 실행 스케줄을 약술한다. 추가적으로, 정적 소프트웨어 스케줄은 제1 작업부하 노드(102)(WN[0])를 제1 CBB(112)에, 제2 작업부하 노드(104)(WN[1])를 제2 CBB(114)에, 제3 작업부하 노드(106)(WN[2])를 제3 CBB(116)에, 제4 작업부하 노드(108)(WN[3])를 제4 CBB(118)에, 그리고 제5 작업부하 노드(110)(WN[4])를 제2 CBB(114)에 할당한다. 제1 버퍼(202)는 제1 CBB(112) 및 제2 CBB(114)에 결합되고, 제2 버퍼(204)는 제2 CBB(114) 및 제3 CBB(116)에 결합되고, 제3 버퍼(206)는 제4 CBB(118) 및 제2 CBB(114)에 결합된다.

[0012] 버퍼들(202, 204 및 206)은 정적 소프트웨어 스케줄이, 각각의 CBB가 시간 간격에서 전체 작업부하 노드를 실행하기 보다는 시간 간격에서 작업부하 노드의 일부(예를 들어, 타일)를 처리하는 것을 약술할 수 있게 한다. 유사하게, 정적 소프트웨어 스케줄은 다른 CBB들(예를 들어, 소비자들)에 의해 생산되는 데이터를 처리하는 CBB들이 작업부하 노드의 부분들(예를 들어, 타일)을, 실행할 수 있는 것- 작업부하의 그러한 부분들이 이용가능할 때 -을 약술할 수 있다. 그러나, 작업부하 노드들을 실행하는 CBB들은 이용가능한 데이터를 처리하고 메모리에 새로운 데이터를 기입하기 때문에, CBB 상에서 주어진 작업부하 노드를 실행하기 위해, 런타임 시 데이터의 임계량이 이용가능해야 하고, 런타임 시 결과들을 기입하기 위해 메모리 내에 공간의 임계량이 존재해야 한다. 버퍼들이 기본 정적 소프트웨어 스케줄링에 의해 메모리 오버헤드를 감소시키지만, 버퍼들이 있는 정적 소프트웨어 스케줄을 약술하는 것이 점점 더 어려운데, 그 이유는 그것이 런타임 시 데이터 가용성들 및/또는 종속성들에 크게 의존하기 때문이다. 더욱이, 전체 가속기의 부하는 가속기 상의 각각의 CBB의 처리 속도에 영향을 미칠 수 있기 때문에, 주어진 가속기의 CBB들을 효과적으로 이용하는 정적 소프트웨어 스케줄을 개발하는 것은 어렵다.

[0013] 본 명세서에 개시된 예들은 작업부하의 정적 매핑의 비순차적 파이프라이닝된 실행을 가능하게 하기 위한 방법

들 및 장치를 포함한다. 정적 소프트웨어 스케줄링과는 대조적으로, 본 명세서에 개시된 예들은 미리 결정된 정적 소프트웨어 스케줄에 의존하지 않는다. 오히려, 본 명세서에 개시된 예들은 가속기 및/또는 다른 처리 요소 상에서 이용가능한 데이터 및 이용가능한 메모리에 기초하여, 주어진 CBB에 할당되었던 어느 작업부하 노드들이 실행될지를 결정한다. 또한, 각각의 CBB는 크레디트들의 제1 수로서 표현되는, 제1 버퍼에서 이용가능한 주어진 작업부하 및 크레디트들의 제2 수로서 표현되는, 제2 버퍼에서 이용가능한 공간의 양과 연관된 데이터의 양을 추적한다. 이것은 주어진 CBB 상의 작업부하 노드들의 동적 런타임 스케줄링을 허용한다.

[0014] 각각의 작업부하 노드에 대해, 크레디트들의 제1 수가 제1 임계값을 충족시키고 크레디트들의 제2 수가 제2 임계값을 충족시킬 때, CBB는 작업부하 노드를 실행할 수 있다. 이것은 전체 작업부하의 주어진 그래프와는 독립적인 비순차적 파이프라이닝된 실행을 허용한다. 본 명세서에 개시된 예들은 가속기의 하나 이상의 계산 빌딩 블록으로의 작업부하의 정적 매핑의 비순차적 파이프라이닝된 실행을 가능하게 하는 장치를 제공한다. 예시적인 장치는 제1 수의 크레디트들을 메모리에 로딩하기 위한 인터페이스; 크레디트들의 제1 수를 버퍼에서의 메모리 가용성과 연관된 크레디트들의 임계 수와 비교하기 위한 비교기; 및 크레디트들의 제1 수가 크레디트들의 임계 수를 충족시킬 때, 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 작업부하의 작업부하 노드를 선택하기 위한 디스패처를 포함한다.

[0015] 도 3은 본 개시의 교시에 따라 구성된 예시적인 컴퓨팅 시스템(300)을 나타내는 블록도이다. 도 3의 예에서, 컴퓨팅 시스템(300)은 예시적인 시스템 메모리(302) 및 예시적인 이종 시스템(304)을 포함한다. 예시적인 이종 시스템(304)은 예시적인 호스트 프로세서(306), 예시적인 제1 통신 버스(308), 예시적인 제1 가속기(310a), 예시적인 제2 가속기(310b), 및 예시적인 제3 가속기(310c)를 포함한다. 예시적인 제1 가속기(310a), 예시적인 제2 가속기(310b), 및 예시적인 제3 가속기(310c) 각각은 가속기의 동작에 대해 일부 일반적인 각종 CBB들 및 각각의 가속기들의 동작에 일부 특정한 각종 CBB들을 포함한다.

[0016] 도 3의 예에서, 시스템 메모리(302)는 이종 시스템(304)에 결합된다. 시스템 메모리(302)는 메모리이다. 도 3에서, 시스템 메모리(302)는 호스트 프로세서(306), 제1 가속기(310a), 제2 가속기(310b) 및 제3 가속기(310c) 중 적어도 하나 사이의 공유 스토리지이다. 도 3의 예에서, 시스템 메모리(302)는 컴퓨팅 시스템(300)에 로컬인 물리적 스토리지이지만; 다른 예들에서, 시스템 메모리(302)는 컴퓨팅 시스템(300) 외부에 있을 수 있고 및/또는 다르게는 컴퓨팅 시스템(300)에 대해 원격일 수 있다. 추가 예들에서, 시스템 메모리(302)는 가상 스토리지일 수 있다. 도 3의 예에서, 시스템 메모리(302)는 지속적 스토리지(예를 들어, ROM(read only memory), PROM(programmable ROM), EPROM(erasable PROM), EEPROM(electrically erasable PROM) 등)이다. 다른 예들에서, 시스템 메모리(302)는 지속적 기본 입력/출력 시스템(BIOS) 또는 플래시 스토리지일 수 있다. 추가 예들에서, 시스템 메모리(302)는 휘발성 메모리일 수 있다.

[0017] 도 3에서, 이종 시스템(304)은 시스템 메모리(302)에 결합된다. 도 3의 예에서, 이종 시스템(304)은 호스트 프로세서(306) 및/또는 제1 가속기(310a), 제2 가속기(310b), 또는 제3 가속기(310c) 중 하나 이상에서 작업부하를 실행함으로써 작업부하를 처리한다. 도 3에서, 이종 시스템(304)은 SoC이다. 대안적으로, 이종 시스템(304)은 임의의 다른 타입의 컴퓨팅 또는 하드웨어 시스템일 수 있다.

[0018] 도 3의 예에서, 호스트 프로세서(306)는 컴퓨터 또는 컴퓨팅 디바이스(예를 들어, 컴퓨팅 시스템(300))와 연관된 동작들의 완료를 실행하고, 수행하고, 및/또는 용이하게 하는 명령어들(예를 들어, 머신 판독가능 명령어들)을 실행하는 처리 요소이다. 도 3의 예에서, 호스트 프로세서(306)는 이종 시스템(304)을 위한 1차 처리 요소이고 적어도 하나의 코어를 포함한다. 대안적으로, 호스트 프로세서(306)는 (예를 들어, 하나보다 많은 CPU가 이용되는 예에서) 공동-1차 처리 요소일 수 있는 반면, 다른 예들에서, 호스트 프로세서(306)는 2차 처리 요소일 수 있다.

[0019] 도 3의 예시된 예에서, 제1 가속기(310a), 제2 가속기(310b), 및/또는 제3 가속기(310c) 중 하나 이상은 하드웨어 가속과 같은 컴퓨팅 작업들을 위해 이종 시스템(304) 상에서 실행하는 프로그램에 의해 이용될 수 있는 처리 요소들이다. 예를 들어, 제1 가속기(310a)는 AI를 위한 머신 비전 작업들을 처리하는 처리 속도 및 전체 성능을 개선하도록 설계 및/또는 다르게는 구성되거나 구조화되는 처리 자원들을 포함하는 처리 요소이다(예를 들어, VPU).

[0020] 본 명세서에 개시된 예들에서, 호스트 프로세서(306), 제1 가속기(310a), 제2 가속기(310b), 및 제3 가속기(310c) 각각은 컴퓨팅 시스템(300) 및/또는 시스템 메모리(302)의 다른 요소들과 통신한다. 예를 들어, 호스트 프로세서(306), 제1 가속기(310a), 제2 가속기(310b), 제3 가속기(310c), 및/또는 시스템 메모리(302)는 제1 통신 버스(308)를 통해 통신한다. 본 명세서에 개시된 일부 예들에서, 호스트 프로세서(306), 제1 가속기

(310a), 제2 가속기(310b), 제3 가속기(310c), 및/또는 시스템 메모리(302)는 임의의 적절한 유선 및/또는 무선 통신 시스템을 통해 통신할 수 있다. 추가로, 본 명세서에 개시된 일부 예들에서, 호스트 프로세서(306), 제1 가속기(310a), 제2 가속기(310b), 제3 가속기(310c), 및/또는 시스템 메모리(302) 각각은 임의의 적절한 유선 및/또는 무선 통신 시스템을 통해 컴퓨팅 시스템(300) 외부의 임의의 컴포넌트와 통신할 수 있다.

[0021] 도 3의 예에서, 제1 가속기(310a)는 예시적인 콘볼루션 엔진(312), 예시적인 RNN 엔진(314), 예시적인 메모리(316), 예시적인 메모리 관리 유닛(MMU)(318), 예시적인 DSP(320), 예시적인 컨트롤러(322), 및 예시적인 직접 메모리 액세스(DMA) 유닛(324)을 포함한다. 추가적으로, 예시적인 콘볼루션 엔진(312), 예시적인 RNN 엔진(314), 예시적인 DMA 유닛(324), 예시적인 DSP(320), 및 예시적인 컨트롤러(322) 각각은 예시적인 제1 스케줄러(326), 예시적인 제2 스케줄러(328), 예시적인 제3 스케줄러(330), 예시적인 제4 스케줄러(332), 및 예시적인 제5 스케줄러(334)를 각각 포함한다. 예시적인 DSP(320) 및 예시적인 컨트롤러(322) 각각은 예시적인 제1 커널 라이브러리(336) 및 예시적인 제2 커널 라이브러리(338)를 추가적으로 포함한다.

[0022] 도 3의 예시된 예에서, 콘볼루션 엔진(312)은 콘볼루션과 연관된 작업들의 처리를 개선하도록 구성되는 디바이스이다. 또한, 콘볼루션 엔진(312)은 시각적 이미지의 분석과 연관된 작업들 및/또는 CNN들과 연관된 다른 작업들의 처리를 개선한다. 도 3에서, RNN 엔진(314)은 RNN들과 연관된 작업들의 처리를 개선하도록 구성되는 디바이스이다. 추가적으로, RNN 엔진(314)은 세그먼트화되지 않은, 연결된 필기 인식, 음성 인식의 분석과 연관된 작업들, 및/또는 RNN들과 연관된 다른 작업들의 처리를 개선한다.

[0023] 도 3의 예에서, 메모리(316)는 콘볼루션 엔진(312), RNN 엔진(314), MMU(318), DSP(320), 컨트롤러(322) 및 DMA 유닛(324) 중 적어도 하나 사이의 공유 스토리지이다. 도 3의 예에서, 메모리(316)는 제1 가속기(310a)에 로컬인 물리적 스토리지이지만; 다른 예들에서, 메모리(316)는 제1 가속기(310a) 외부에 있을 수 있고 및/또는 다르게는 제1 가속기(310a)에 대해 원격일 수 있다. 추가 예들에서, 메모리(316)는 가상 스토리지일 수 있다. 도 3의 예에서, 메모리(316)는 지속적 스토리지(예를 들어, ROM, PROM, EPROM, EEPROM 등)이다. 다른 예들에서, 메모리(316)는 지속적 BIOS 또는 플래시 스토리지일 수 있다. 추가 예들에서, 메모리(316)는 휘발성 메모리일 수 있다.

[0024] 도 3의 예시된 예에서, 예시적인 MMU(318)는 메모리(316) 및/또는 시스템 메모리(302)의 어드레스들에 대한 참조들을 포함하는 디바이스이다. MMU(318)는 콘볼루션 엔진(312), RNN 엔진(314), DSP(320) 및/또는 컨트롤러(322) 중 하나 이상에 의해 이용되는 가상 메모리 어드레스들을 메모리(316) 및/또는 시스템 메모리(302) 내의 물리 어드레스들로 추가로 변환한다.

[0025] 도 3의 예에서, DSP(320)는 디지털 신호들의 처리를 개선하는 디바이스이다. 예를 들어, DSP(320)는, 카메라들, 및/또는 컴퓨터 비전에 관련된 다른 센서들로부터의 데이터와 같은 연속적인 실세계 신호들을 측정, 필터링 및/또는 압축하기 위한 처리를 용이하게 한다. 도 3에서, 컨트롤러(322)는 제1 가속기(310a)의 제어 유닛으로서 구현된다. 예를 들어, 컨트롤러(322)는 제1 가속기(310a)의 동작을 지시한다. 일부 예들에서, 컨트롤러(322)는 크레디트 매니저를 구현한다. 또한, 컨트롤러(322)는 호스트 프로세서(306)로부터 수신된 머신 판독가능 명령어들에 어떻게 응답하는지를 콘볼루션 엔진(312), RNN 엔진(314), 메모리(316), MMU(318), 및/또는 DSP(320) 중 하나 이상에게 지시할 수 있다.

[0026] 도 3의 예시된 예에서, DMA 유닛(324)은, 콘볼루션 엔진(312), RNN 엔진(314), DSP(320) 및 컨트롤러(322) 중 적어도 하나가 호스트 프로세서(306)와는 독립적인 시스템 메모리(302)에 액세스하는 것을 허용하는 디바이스이다. 예를 들어, DMA 유닛(324)은 하나 이상의 아날로그 또는 디지털 회로(들), 로직 회로들, 프로그래머블 프로세서(들), 프로그래머블 컨트롤러(들), 그래픽 처리 유닛(들)(GPU(들)), 디지털 신호 프로세서(들)(DSP(들)), 애플리케이션 특정 통합 회로(들)(ASIC(들)), 프로그래머블 로직 디바이스(들)(PLD(들)) 및/또는 필드 프로그래머블 로직 디바이스(들)(FPLD(들))에 의해 구현될 수 있다.

[0027] 도 3의 예에서, 제1 스케줄러(326), 제2 스케줄러(328), 제3 스케줄러(330), 제4 스케줄러(332), 및 제5 스케줄러(334) 각각은, 콘볼루션 엔진(312), RNN 엔진(314), DMA 유닛(324), DSP(320), 및 컨트롤러(322)가 각각, 제1 가속기(310a)에 오프로딩되고 및/또는 다르게는 제1 가속기(310a)에 전송되었던 작업부하의 일부를 언제 실행할지를 결정하는 디바이스이다. 추가적으로, 제1 커널 라이브러리(336) 및 제2 커널 라이브러리(338) 각각은 하나 이상의 커널을 포함하는 데이터 구조이다. 제1 커널 라이브러리(336) 및 제2 커널 라이브러리(338)의 커널들은, 예를 들어, 각각 DSP(320) 및 컨트롤러(322) 상에서 높은 처리량을 위해 컴파일된 루틴들이다. 커널들은, 예를 들어, 컴퓨팅 시스템(300) 상에서 실행될 실행파일의 실행가능 서브-섹션들에 대응한다.

[0028] 본 명세서에 개시된 예에서, 콘볼루션 엔진(312), RNN 엔진(314), 메모리(316), MMU(318), DSP(320), 컨트롤러(322) 및 DMA 유닛(324) 각각은 제1 가속기(310a)의 다른 요소들과 통신한다. 예를 들어, 콘볼루션 엔진(312), RNN 엔진(314), 메모리(316), MMU(318), DSP(320), 컨트롤러(322), 및 DMA 유닛(324)은 예시적인 제2 통신 버스(340)를 통해 통신한다. 일부 예들에서, 제2 통신 버스(340)는 구성 및 제어(CnC) 패브릭 및 데이터 패브릭에 의해 구현될 수 있다. 본 명세서에 개시된 일부 예들에서, 콘볼루션 엔진(312), RNN 엔진(314), 메모리(316), MMU(318), DSP(320), 컨트롤러(322) 및 DMA 유닛(324)은 임의의 적절한 유선 및/또는 무선 통신 시스템을 통해 통신할 수 있다. 추가적으로, 본 명세서에 개시된 일부 예들에서, 콘볼루션 엔진(312), RNN 엔진(314), 메모리(316), MMU(318), DSP(320), 컨트롤러(322) 및 DMA 유닛(324) 각각은 임의의 적절한 유선 및/또는 무선 통신 시스템을 통해 제1 가속기(310a) 외부의 임의의 컴포넌트와 통신할 수 있다.

[0029] 앞서 언급한 바와 같이, 예시적인 제1 가속기(310a), 예시적인 제2 가속기(310b), 및 예시적인 제3 가속기(310c) 각각은 가속기의 동작에 대한 일부 일반적인 각종 CBB들 및 각각의 가속기들의 동작에 일부 특정한 각종 CBB들을 포함한다. 예를 들어, 제1 가속기(310a), 제2 가속기(310b), 및 제3 가속기(310c) 각각은 메모리, MMU, 컨트롤러, 및 CBB들 각각에 대한 각자의 스케줄러들과 같은 일반적인 CBB들을 포함한다.

[0030] 도 3의 예에서, 제1 가속기(310a)는 VPU를 구현하고 콘볼루션 엔진(312), RNN 엔진(314), 및 DSP(320)(예를 들어, 제1 가속기(310a)의 동작에 특정한 동작에 특정한 CBB들)를 포함하지만, 제2 가속기(310b) 및 제3 가속기(310c)는 제2 가속기(310b) 및/또는 제3 가속기(310c)의 동작에 특정한 추가적인 또는 대안적인 CBB들을 포함할 수 있다. 예를 들어, 제2 가속기(310b)가 GPU를 구현하는 경우, 제2 가속기(310b)의 동작에 특정한 CBB들은 스레드 디스패처, 그래픽 기술 인터페이스, 및/또는 컴퓨터 그래픽 및/또는 이미지 처리를 처리하는 처리 속도 및 전체 성능을 개선하는 데 바람직한 임의의 다른 CBB를 포함할 수 있다. 또한, 제3 가속기(310c)가 FPGA를 구현하는 경우, 제3 가속기(310c)의 동작에 특정한 CBB들은 하나 이상의 산술 로직 유닛(ALU), 및/또는 일반적인 계산들을 처리하는 처리 속도 및 전체 성능을 개선하는 데 바람직한 임의의 다른 CBB를 포함할 수 있다.

[0031] 도 3의 이종 시스템(304)은 호스트 프로세서(306), 제1 가속기(310a), 제2 가속기(310b), 및 제3 가속기(310c)를 포함하지만, 일부 예들에서, 이종 시스템(304)은 ASIP(application-specific instruction set processor)들, PPU(physic processing unit)들, 지정된 DSP들, 이미지 프로세서들, 코프로세서들, 부동 소수점 유닛들, 네트워크 프로세서들, 멀티 코어 프로세서들, 및 프론트엔드 프로세서들을 포함하는 임의의 수의 처리 요소(예를 들어, 호스트 프로세서들 및/또는 가속기들)를 포함할 수 있다.

[0032] 또한, 도 3의 예에서, 콘볼루션 엔진(312), RNN 엔진(314), 메모리(316), MMU(318), DSP(320), 컨트롤러(322), DMA 유닛(324), 제1 스케줄러(326), 제2 스케줄러(328), 제3 스케줄러(330), 제4 스케줄러(332), 제5 스케줄러(334), 제1 커널 라이브러리(336), 및 제2 커널 라이브러리(338)는 제1 가속기(310a) 상에서 구현되고, 콘볼루션 엔진(312), RNN 엔진(314), 메모리(316), MMU(318), DSP(320), 컨트롤러(322), DMA 유닛(324), 제1 스케줄러(326), 제2 스케줄러(328), 제3 스케줄러(330), 제4 스케줄러(332), 제5 스케줄러(334), 제1 커널 라이브러리(336), 및 제2 커널 라이브러리(338) 중 하나 이상은 호스트 프로세서(306), 제2 가속기(310b), 및/또는 제3 가속기(310c) 상에서 구현될 수 있다.

[0033] 도 4는 예시적인 하나 이상의 스케줄러를 포함하는 예시적인 컴퓨팅 시스템(400)을 나타내는 블록도이다. 일부 예들에서, 컴퓨팅 시스템(400)은 도 3의 컴퓨팅 시스템(300)에 대응할 수 있다. 도 4의 예에서, 컴퓨팅 시스템(400)은 예시적인 입력(402), 예시적인 컴파일러(404), 및 예시적인 가속기(406)를 포함한다. 일부 예들에서, 가속기(406)는 도 3의 제1 가속기(310a)에 대응할 수 있다. 도 4에서, 입력(402)은 컴파일러(404)에 결합된다. 입력(402)은 가속기(406)에 의해 실행되는 작업부하이다. 일부 예들에서, 컴파일러(404)는 도 3의 호스트 프로세서(306) 및/또는 외부 디바이스에 대응할 수 있다.

[0034] 도 4의 예에서, 입력(402)은, 예를 들어, 가속기(406)에 의해 실행될 함수, 알고리즘, 프로그램, 애플리케이션, 및/또는 다른 코드이다. 일부 예들에서, 입력(402)은 함수, 알고리즘, 프로그램, 애플리케이션 및/또는 다른 코드의 그래프 설명이다. 추가적인 또는 대안적인 예들에서, 입력(402)은 딥 러닝 및/또는 컴퓨터 비전과 같은 AI 처리와 관련된 작업부하이다.

[0035] 도 4의 예시된 예에서, 컴파일러(404)는 입력(402) 및 가속기(406)에 결합된다. 컴파일러(404)는 입력(402)을 수신하고, 가속기(406)에 의해 실행될 하나 이상의 실행파일로 입력(402)을 컴파일한다. 예를 들어, 컴파일러(404)는 입력(402)을 수신하고 작업부하(예를 들어, 입력(402))의 다양한 작업부하 노드들을 가속기(406)의 다양한 CBB들에 할당하는 그래프 컴파일러이다. 추가적으로, 컴파일러(404)는 가속기(406)의 메모리 내의 하나

이상의 버퍼에 대한 메모리를 할당한다.

[0036] 도 4의 예에서, 가속기(406)는 컴파일러(404)에 결합되고, 예시적인 크레디트 매니저(408), 예시적인 CnC 패브릭(410), 예시적인 데이터 패브릭(411), 예시적인 콘볼루션 엔진(412), 예시적인 DMA 유닛(414), 예시적인 RNN 엔진(416), 예시적인 DSP(418), 예시적인 메모리(420) 및 예시적인 MMU(422)를 포함한다. 추가적으로, 예시적인 콘볼루션 엔진(412), 예시적인 DMA 유닛(414), 예시적인 RNN 엔진(416), 및 예시적인 DSP(418) 각각은 예시적인 제1 스케줄러(424), 예시적인 제2 스케줄러(426), 예시적인 제3 스케줄러(428), 및 예시적인 제4 스케줄러(430)를 각각 포함한다. 더욱이, 예시적인 DSP(418)는 예시적인 커널 라이브러리(432)를 포함한다. 일부 예들에서, 제1 스케줄러(424)는 도 3의 제1 스케줄러(326)에 대응할 수 있다. 추가적인 또는 대안적인 예들에서, 제2 스케줄러(426)는 도 3의 제3 스케줄러(330)에 대응할 수 있다. 추가 예들에서, 제3 스케줄러(428)는 도 3의 제2 스케줄러(328)에 대응할 수 있다. 일부 예들에서, 제4 스케줄러(430)는 도 4의 제4 스케줄러(332)에 대응할 수 있다.

[0037] 도 4의 예시된 예에서, 크레디트 매니저(408)는 컴파일러(404) 및 CnC 패브릭(410)에 결합된다. 크레디트 매니저(408)는 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및/또는 DSP(418) 중 하나 이상과 연관된 크레디트들을 관리하는 디바이스이다. 일부 예들에서, 크레디트 매니저(408)는 컨트롤러에 의해 크레디트 매니저 컨트롤러로서 구현될 수 있다. 크레디트들은 작업부하 노드의 출력을 위해 메모리(420)에서 이용가능한 공간의 양 및/또는 메모리(420)에서 이용가능한 작업부하 노드들과 연관된 데이터를 나타낸다. 예를 들어, 크레디트 매니저(408)는 컴파일러(404)로부터 수신된 하나 이상의 실행파일에 기초하여, 주어진 작업부하의 각각의 작업부하 노드와 연관된 하나 이상의 버퍼로 메모리(420)를 파티셔닝할 수 있다. 작업부하 노드가 버퍼에 데이터를 기입하도록 구성되는 경우, 작업부하 노드는 생산자이고, 작업부하 노드가 버퍼로부터 데이터를 판독하도록 구성되는 경우, 작업부하 노드는 소비자이다.

[0038] 도 4의 예에서, 크레디트 매니저(408)는 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및/또는 DSP(418) 중 하나 이상에 크레디트들을 전송하고/하거나 그로부터 크레디트들을 수신하도록 추가로 구성된다. 일부 예들에서, 크레디트 매니저(408)는 가속기(406)의 제어 유닛으로서 구현된다. 예를 들어, 크레디트 매니저(408)는 가속기(406)의 동작을 지시할 수 있다. 또한, 크레디트 매니저(408)는 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및/또는 DSP(418) 중 하나 이상에게, 컴파일러(404)로부터 수신된 실행파일들 및/또는 다른 머신 판독가능 명령어들에 어떻게 응답하는지를 지시할 수 있다.

[0039] 도 4의 예에서, CnC 패브릭(410)은 크레디트 매니저(408), 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및 DSP(418)에 결합된다. CnC 패브릭(410)은, 크레디트 매니저(408), 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및/또는 DSP(418) 중 하나 이상이 크레디트 매니저(408), 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및/또는 DSP(418) 중 하나 이상에 크레디트들을 송신하고/하거나 그로부터 크레디트들을 수신하는 것을 허용하는 적어도 하나의 로직 회로 및 전자 상호결합들의 네트워크이다. 일부 예들에서, CnC 패브릭(410)은 도 3의 제2 통신 버스(340)에 대응할 수 있다.

[0040] 도 4의 예에서, 데이터 패브릭(411)은 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), DSP(418), 메모리(420), 및 MMU(422)에 결합된다. 데이터 패브릭(411)은, 크레디트 매니저(408), 콘볼루션 엔진(412), RNN 엔진(416), DSP(418), 메모리(420), 및/또는 MMU(422) 중 하나 이상이 크레디트 매니저(408), 콘볼루션 엔진(412), RNN 엔진(416), DSP(418), 메모리(420), 및/또는 MMU(422) 중 하나 이상에 데이터를 송신하고/하거나 그로부터 데이터를 수신하는 것을 허용하는 적어도 하나의 로직 회로 및 전자 상호결합들의 네트워크이다. 일부 예들에서, 데이터 패브릭(411)은 도 3의 제2 통신 버스(340)에 대응할 수 있다.

[0041] 도 4의 예시된 예에서, 콘볼루션 엔진(412)은 CnC 패브릭(410) 및 데이터 패브릭(411)에 결합된다. 콘볼루션 엔진(412)은 콘볼루션과 연관된 작업들의 처리를 개선하도록 구성되는 디바이스이다. 또한, 콘볼루션 엔진(412)은 시각적 이미지의 분석과 연관된 작업들 및/또는 CNN들과 연관된 다른 작업들의 처리를 개선한다. 일부 예들에서, 콘볼루션 엔진(412)은 도 3의 콘볼루션 엔진(312)에 대응할 수 있다.

[0042] 도 4의 예시된 예에서, DMA 유닛(414)은 CnC 패브릭(410) 및 데이터 패브릭(411)에 결합된다. DMA 유닛(414)은, 콘볼루션 엔진(412), RNN 엔진(416), 또는 DSP(418) 중 적어도 하나가 각각의 프로세서(예를 들어, 호스트 프로세서(306))와는 독립적인 가속기(406)에 원격인 메모리(예를 들어, 시스템 메모리(302))에 액세스하는 것을 허용하는 디바이스이다. 일부 예들에서, DMA 유닛(414)은 도 3의 DMA 유닛(324)에 대응할 수 있다. 예를 들어, DMA 유닛(414)은 하나 이상의 아날로그 또는 디지털 회로(들), 로직 회로들, 프로그래머블 프로세서(들), 프로그래머블 컨트롤러(들), GPU(들), DSP(들), ASIC(들), PLD(들) 및/또는 FPLD(들)에 의해 구현될 수 있다.

[0043] 도 4에서, RNN 엔진(416)은 CnC 패브릭(410) 및 데이터 패브릭(411)에 결합된다. RNN 엔진(416)은 RNN과 연관된 작업들의 처리를 개선하도록 구성되는 디바이스이다. 추가적으로, RNN 엔진(416)은 세그먼트화되지 않은, 연결된 필기 인식, 음성 인식의 분석과 연관된 작업들, 및/또는 RNN들과 연관된 다른 작업들의 처리를 개선한다. 일부 예들에서, RNN 엔진(416)은 도 3의 RNN 엔진(314)에 대응할 수 있다.

[0044] 도 4의 예에서, DSP(418)는 CnC 패브릭(410) 및 데이터 패브릭(411)에 결합된다. DSP(418)는 디지털 신호들의 처리를 개선하는 디바이스이다. 예를 들어, DSP(418)는, 카메라들, 및/또는 컴퓨터 비전에 관련된 다른 센서들로부터의 데이터와 같은 연속적인 실세계 신호들을 측정, 필터링 및/또는 압축하기 위한 처리를 용이하게 한다. 일부 예들에서 DSP(418)는 도 3의 DSP(320)에 대응할 수 있다.

[0045] 도 4의 예에서, 메모리(420)는 데이터 패브릭(411)에 결합된다. 메모리(420)는 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416) 및 DSP(418) 중 적어도 하나 사이의 공유 스토리지이다. 일부 예들에서, 메모리(420)는 도 3의 메모리(316)에 대응할 수 있다. 메모리(420)는 크레디트 매니저(408)에 의해 수신된 실행파일과 연관된 작업부하의 하나 이상의 작업부하 노드와 연관된 하나 이상의 버퍼로 파티셔닝될 수 있다. 도 4의 예에서, 메모리(420)는 가속기(406)에 로컬인 물리적 스토리지이다. 그러나, 다른 예들에서, 메모리(420)는 가속기(406) 외부에 있을 수 있고 및/또는 다르게는 가속기(406)에 대해 원격일 수 있다. 추가 예들에서, 메모리(420)는 가상 스토리지일 수 있다. 도 4의 예에서, 메모리(420)는 지속적 스토리지(예를 들어, ROM, PROM, EPROM, EEPROM 등)이다. 다른 예들에서, 메모리(420)는 지속적 BIOS 또는 플래시 스토리지일 수 있다. 추가 예들에서, 메모리(420)는 휘발성 메모리일 수 있다.

[0046] 도 4의 예시된 예에서, 예시적인 MMU(422)는 데이터 패브릭(411)에 결합된다. MMU(422)는 가속기(406)에 대해 원격인 메모리 및/또는 메모리(420)의 어드레스들에 대한 참조들을 포함하는 디바이스이다. MMU(422)는 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및/또는 DSP(418) 중 하나 이상에 의해 이용되는 가상 메모리 어드레스들을, 가속기(406)에 대해 원격인 메모리 및/또는 메모리(420) 내의 물리 어드레스들로 추가로 변환한다. 일부 예들에서, MMU(422)는 도 3의 MMU(318)에 대응할 수 있다.

[0047] 도 4의 예에서, 제1 스케줄러(424), 제2 스케줄러(426), 제3 스케줄러(428), 및 제4 스케줄러(430) 각각은, 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및 DSP(418)가 각각, 가속기(406)의 추가적인 CBB 및/또는 크레디트 매니저(408)에 의해, 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), 및 DSP(418)에 각각 할당되었던 작업부하의 일부(예를 들어, 작업부하 노드)를 언제 실행할지를 결정하는 디바이스이다. 주어진 작업부하 노드의 작업들 및/또는 다른 동작들에 따라, 작업부하 노드는 생산자 또는 소비자일 수 있다. 생산자 작업부하 노드는 다른 작업부하 노드에 의해 이용되는 데이터를 생산하는 반면, 소비자 작업부하 노드는 다른 작업부하 노드에 의해 생산된 데이터를 소비하고/하거나 다르게는 처리한다.

[0048] 도 4의 예시된 예에서, 커널 라이브러리(432)는 하나 이상의 커널을 포함하는 데이터 구조이다. 일부 예들에서, 커널 라이브러리(432)는 도 3의 제1 커널 라이브러리(336)에 대응할 수 있다. 커널 라이브러리(432)의 커널들은, 예를 들어, DSP(418) 상에서 높은 처리량을 위해 컴파일된 루틴들이다. 커널들은, 예를 들어, 가속기(406) 상에서 실행될 실행파일의 실행가능 서브-섹션들에 대응한다. 도 4의 예에서, 가속기(406)는 VPU를 구현하고 크레디트 매니저(408), CnC 패브릭(410), 데이터 패브릭(411), 콘볼루션 엔진(412), DMA 유닛(414), RNN 엔진(416), DSP(418), 메모리(420) 및 MMU(422)를 포함하는 한편, 가속기(406)는 도 4에 도시된 것들에 추가적인 또는 대안적인 CBB들을 포함할 수 있다.

[0049] 도 4의 예에서, 동작 시, 제1 스케줄러(424)는 콘볼루션 엔진(412)에 할당된 작업부하 노드들에 대해 작업부하 노드로의 입력 버퍼들 및 작업부하 노드로부터의 출력 버퍼들에 대응하는 크레디트들을 로딩한다. 예를 들어, 입력 버퍼는 작업부하 노드가 데이터를 판독하도록 구성되는 버퍼이고, 반면에 출력 버퍼는 작업부하 노드가 데이터를 기입하도록 구성되는 버퍼이다. 일부 예들에서, 제1 작업부하 노드의 입력 버퍼는 제2 작업부하 노드의 출력 버퍼일 수 있다. 더욱이, 제1 스케줄러(424)는 크레디트 매니저(408)로부터 크레디트들을 수신하고/하거나 다르게는 획득한다.

[0050] 도 4의 예에서, 동작 시, 제1 스케줄러(424)는 콘볼루션 엔진(412)에 할당된 작업부하 노드를 선택하고, 제1 스케줄러(424)가 선택된 작업부하 노드로의 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했는지를 결정한다. 예를 들어, 제1 스케줄러(424)는 입력 버퍼에 대한 생산자 작업부하 노드로부터 수신된 크레디트들의 수를, 입력 버퍼에 대한 크레디트들의 임계 수와 비교한다. 제1 스케줄러(424)가 크레디트들의 임계량을 수신하지 않았을 경우, 제1 스케줄러(424)는 콘볼루션 엔진(412)에 할당된 다른 작업부하 노

드 상에서 프로세스를 반복한다.

[0051] 도 4에 예시된 예에서, 동작 시에, 제1 스케줄러(424)가 선택된 작업부하 노드로의 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했다면, 제1 스케줄러(424)는 선택된 작업부하 노드에 대한 출력 버퍼에 데이터를 기입하기 위해 제1 스케줄러(424)가 크레디트들의 임계량을 수신했는지를 결정한다. 예를 들어, 제1 스케줄러(424)는 출력 버퍼에 대한 소비자 작업부하 노드로부터 수신된 크레디트들의 수를, 선택된 작업부하 노드에 대한 출력 버퍼에 대한 크레디트들의 임계 수와 비교한다. 제1 스케줄러(424)가 크레디트들의 임계량을 수신하지 않았을 경우, 제1 스케줄러(424)는 콘볼루션 엔진(412)에 할당된 다른 작업부하 노드 상에서 프로세스를 반복한다. 출력 버퍼에 데이터를 기입하기 위해 제1 스케줄러(424)가 크레디트들의 임계량을 수신했다면, 제1 스케줄러(424)는 선택된 작업부하 노드가 실행할 준비가 된 것을 표시한다. 후속하여, 제1 스케줄러(424)는 콘볼루션 엔진(412)에 할당된 추가적인 작업부하 노드들에 대해 이 프로세스를 반복한다.

[0052] 도 4의 예에서, 동작 시에, 콘볼루션 엔진(412)에 할당된 작업부하 노드들이 분석된 후에, 제1 스케줄러(424)는 실행할 준비가 된 작업부하 노드들을 스케줄링한다. 제1 스케줄러(424)는 후속하여, 스케줄에 따라 작업부하 노드를 디스패치한다. 디스패치된 작업부하 노드가 콘볼루션 엔진(412)에 의해 실행된 후, 제1 스케줄러(424)는 입력 버퍼 및/또는 출력 버퍼에 대응하는 크레디트들을 크레디트 매니저(408)에 전송한다. 제1 스케줄러(424)는 실행될, 스케줄 내의 추가적인 작업부하 노드들이 존재하는지를 결정한다. 스케줄 내의 추가적인 작업부하 노드들이 존재하는 경우, 제1 스케줄러(424)는 스케줄 내의 다음 작업부하 노드가 콘볼루션 엔진(412) 상에서 실행되게 한다.

[0053] 도 5는 도 3 및 도 4의 스케줄러들 중 하나 이상을 구현할 수 있는 예시적인 스케줄러(500)의 블록도이다. 예를 들어, 스케줄러(500)는 도 3의 제1 스케줄러(326), 제2 스케줄러(328), 제3 스케줄러(330), 제4 스케줄러(332), 및/또는 제5 스케줄러(334), 및/또는 도 4의 제1 스케줄러(424), 제2 스케줄러(426), 제3 스케줄러(428) 및/또는 제4 스케줄러(430), 및/또는 도 6의 스케줄러(600), 및/또는 도 7의 제1 스케줄러(722), 제2 스케줄러(724), 제3 스케줄러(726), 및/또는 제4 스케줄러(728)의 예시적인 구현이다.

[0054] 도 5의 예에서, 스케줄러(500)는 예시적인 작업부하 인터페이스(502), 예시적인 버퍼 크레디트 스토리지(504), 예시적인 크레디트 비교기(506), 예시적인 작업부하 노드 디스패처(508), 및 예시적인 통신 버스(510)를 포함한다. 스케줄러(500)는, 스케줄러(500)가 연관되는 CBB가, 스케줄러(500)가 연관되는 CBB에 할당되었던 작업부하의 일부(예를 들어, 작업부하 노드)를 언제 실행할지를 결정하는 디바이스이다.

[0055] 도 5의 예시된 예에서, 작업부하 인터페이스(502)는 스케줄러(500), 버퍼 크레디트 스토리지(504), 크레디트 비교기(506), 및/또는 작업부하 노드 디스패처(508) 외부의 다른 디바이스들과 통신하도록 구성되는 디바이스이다. 예를 들어, 작업부하 인터페이스(502)는 스케줄러(500)가 연관되는 CBB에 의해 실행될 작업부하 노드들을 수신하고/하거나 다르게는 획득할 수 있다. 추가적으로 또는 대안적으로, 작업부하 인터페이스(502)는 다른 스케줄러들, 다른 CBB들, 및/또는 다른 디바이스들로부터 크레디트들을 송신 및/또는 수신할 수 있다. 또한, 작업부하 인터페이스(502)는 작업부하 노드로의 입력 버퍼들 및/또는 작업부하 노드로부터의 출력 버퍼들에 대응하는 크레디트들을 버퍼 크레디트 스토리지(504) 내로 및/또는 밖으로 로딩할 수 있다.

[0056] 일부 예들에서, 예시적인 작업부하 인터페이스(502)는 인터페이싱을 위한 예시적인 수단을 구현한다. 인터페이싱 수단은 도 8의 적어도 블록들(802, 818, 및 822)에 의해 구현되는 것과 같은 실행가능 명령어들에 의해 구현된다. 예를 들어, 도 8의 블록들(802, 818, 및 822)의 실행가능 명령어들은 도 9의 예에 예시된 예시적인 가속기(912) 및/또는 예시적인 프로세서(910)와 같은 적어도 하나의 프로세서 상에서 실행될 수 있다. 다른 예들에서, 인터페이싱 수단은 하드웨어 로직, 하드웨어로 구현되는 상태 머신들, 로직 회로, 및/또는 하드웨어, 소프트웨어, 및/또는 펌웨어의 임의의 다른 조합에 의해 구현된다.

[0057] 도 5에 예시된 예에서, 버퍼 크레디트 스토리지(504)는 작업부하 인터페이스(502), 크레디트 비교기(506), 및/또는 작업부하 노드 디스패처(508) 중 적어도 하나 사이의 공유 스토리지이다. 버퍼 크레디트 스토리지(504)는 스케줄러(500)에 로컬인 물리적 스토리지이지만; 다른 예들에서, 버퍼 크레디트 스토리지(504)는 스케줄러(500) 외부에 있을 수 있고 및/또는 다르게는 그에 대해 원격일 수 있다. 추가 예들에서, 버퍼 크레디트 스토리지(504)는 가상 스토리지일 수 있다. 도 5의 예에서, 버퍼 크레디트 스토리지(504)는 지속적 스토리지(예를 들어, ROM, PROM, EPROM, EEPROM 등)이다. 다른 예들에서, 버퍼 크레디트 스토리지(504)는 지속적 BIOS 또는 플래시 스토리지일 수 있다. 추가 예들에서, 버퍼 크레디트 스토리지(504)는 휘발성 메모리일 수 있다.

[0058] 도 5의 예에서, 버퍼 크레디트 스토리지(504)는 스케줄러(500)가 연관되는 CBB에 할당된 작업부하 노드들과 연

관된 작업부하 노드들로의 입력 버퍼들 및/또는 작업부하 노드들로부터의 출력 버퍼들에 대응하는 크레디트들을 저장하는 것과 연관되는 메모리이다. 예를 들어, 버퍼 크레디트 스토리지(504)는 스케줄러(500)가 연관되는 CBB에 할당되는 각각의 작업부하 노드에 대한 필드들, 및 스케줄러(500)가 연관되는 CBB에 할당된 작업부하 노드들과 연관된 작업부하 노드들로의 각각의 입력 버퍼들 및/또는 작업부하 노드들로부터의 각각의 출력 버퍼들에 대한 필드들을 포함하는 데이터 구조로서 구현될 수 있다.

[0059] 도 5의 예시된 예에서, 버퍼 크레디트 스토리지(504)는 스케줄러(500)가 연관되는 CBB에 할당되었던 작업부하 노드들 및/또는 작업부하 노드들로의 입력 버퍼들 및/또는 작업부하 노드들로부터의 출력 버퍼들에 대응하는 크레디트들의 임계량을 추가적으로 또는 대안적으로 저장할 수 있다. 또한, 버퍼 크레디트 스토리지(504)는 각각의 작업부하 노드로의 입력 버퍼들 및/또는 각각의 작업부하 노드로부터의 출력 버퍼들에 대한 크레디트들의 임계 수와 연관된 필드를 포함한다.

[0060] 도 5의 예에서, 작업부하 노드가 생산자일 때(예를 들어, 작업부하 노드가 다른 작업부하 노드에 의해 이용될 데이터를 생성할 때), 크레디트들의 임계 수는 스케줄러(500)가 연관되는 CBB가 생산자 작업부하 노드를 실행할 수 있기 전에 충족되어야 하는 출력 버퍼(예를 들어, 메모리(420) 내의 파티셔닝된 공간) 내의 공간의 임계량에 대응한다. 추가적으로, 작업부하 노드가 소비자일 때(예를 들어, 작업부하 노드가 다른 작업부하 노드에 의해 생성된 데이터를 처리할 때), 크레디트들의 임계 수는 스케줄러(500)가 연관되는 CBB가 소비자 작업부하 노드를 실행할 수 있기 전에 충족되어야 하는 입력 버퍼(예를 들어, 메모리(420) 내의 파티셔닝된 공간) 내의 데이터의 임계량에 대응한다.

[0061] 일부 예들에서, 예시적인 버퍼 크레디트 스토리지(504)는 저장하기 위한 예시적인 수단을 구현한다. 저장 수단은 도 8에서 구현되는 것과 같은 실행가능 명령어들에 의해 구현될 수 있다. 예를 들어, 실행가능 명령어들은 도 9의 예에 예시된 예시적인 가속기(912) 및/또는 예시적인 프로세서(910)와 같은 적어도 하나의 프로세서 상에서 실행될 수 있다. 다른 예들에서, 저장 수단은 하드웨어 로직, 하드웨어로 구현되는 상태 머신들, 로직 회로, 및/또는 하드웨어, 소프트웨어, 및/또는 펌웨어의 임의의 다른 조합에 의해 구현된다.

[0062] 도 5에 예시된 예에서, 크레디트 비교기(506)는, 스케줄러(500)가 연관되는 CBB에 할당된 작업부하 노드들로의 입력 버퍼들 및/또는 작업부하 노드들로부터의 출력 버퍼들에 대응하는 크레디트들의 임계 수가 수신되었는지를 결정하도록 구성되는 디바이스이다. 크레디트 비교기(506)는 스케줄러(500)가 연관되는 CBB에 할당된 작업부하 노드를 선택하도록 구성된다.

[0063] 도 5의 예에서, 크레디트 비교기(506)는 스케줄러(500)가 선택된 작업부하 노드에 대한 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했는지를 결정하도록 추가로 구성된다. 예를 들어, 크레디트 비교기(506)는 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)로부터 수신된 크레디트들의 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드를, 선택된 작업부하 노드로의 입력 버퍼에 대한 크레디트들의 임계 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드와 비교한다. 스케줄러(500)가 크레디트들의 임계량을 수신하지 않았다면, 크레디트 비교기(506)는 스케줄러(500)가 연관되는 CBB에 할당된 다른 작업부하 노드 상에서 프로세스를 반복한다.

[0064] 도 5에 예시된 예에서, 스케줄러(500)가 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했다면, 크레디트 비교기(506)는 스케줄러(500)가 선택된 작업부하 노드에 대한 출력 버퍼에 데이터를 기입하기 위해 크레디트들의 임계량을 수신했는지를 결정한다. 예를 들어, 크레디트 비교기(506)는 선택된 작업부하 노드에 대한 출력 버퍼에 대한 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)로부터 수신된 크레디트들의 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드를, 출력 버퍼에 대한 크레디트들의 임계 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드와 비교한다.

[0065] 도 5의 예에서, 스케줄러(500)가 크레디트들의 임계량을 수신하지 않았을 경우, 크레디트 비교기(506)는 스케줄러(500)가 연관되는 CBB에 할당된 다른 작업부하 노드 상에서 프로세스를 반복한다. 스케줄러(500)가 출력 버퍼에 데이터를 기입하기 위해 크레디트들의 임계량을 수신했다면, 크레디트 비교기(506)는 선택된 작업부하 노드가 실행할 준비가 된 것을 표시한다. 후속하여, 크레디트 비교기(506)는 스케줄러(500)가 연관되는 CBB에 할당된 추가적인 작업부하 노드들에 대해 이 프로세스를 반복한다.

[0066] 일부 예들에서, 예시적인 크레디트 비교기(506)는 비교하기 위한 예시적인 수단을 구현한다. 비교 수단은 도 8의 적어도 블록들(804, 806, 808, 810, 및 812)에 의해 구현되는 것과 같은 실행가능 명령어들에 의해 구현된다. 예를 들어, 도 8의 블록들(804, 806, 808, 810 및 812)의 실행가능 명령어들은 도 9의 예에 예시된

예시적인 가속기(912) 및/또는 예시적인 프로세서(910)와 같은 적어도 하나의 프로세서 상에서 실행될 수 있다. 다른 예들에서, 비교 수단은 하드웨어 로직, 하드웨어로 구현되는 상태 머신들, 로직 회로, 및/또는 하드웨어, 소프트웨어, 및/또는 펌웨어의 임의의 다른 조합에 의해 구현된다.

[0067] 도 5의 예에서, 작업부하 노드 디스패처(508)는 스케줄러(500)가 연관되는 CBB 상에서 실행될, 스케줄러(500)가 연관되는 CBB에 할당된 하나 이상의 작업부하 노드들을 스케줄링하는 디바이스이다. 예를 들어, 스케줄러(500)가 연관되는 CBB에 할당된 작업부하 노드들이 분석된 후에, 작업부하 노드 디스패처(508)는 실행할 준비가 된 작업부하 노드들을 스케줄링한다. 예를 들어, 작업부하 노드 디스패처(508)는 라운드-로빈(round-robin) 스케줄과 같은 스케줄링 알고리즘에 기초하여 실행할 준비가 된 작업부하 노드들을 스케줄링한다. 작업부하 노드 디스패처(508)는 후속하여, 스케줄에 따라 작업부하 노드를 디스패치한다. 다른 예들에서, 작업부하 노드 디스패처(508)는 임의의 다른 적절한 중재 알고리즘을 이용하여, 실행할 준비가 된 작업부하 노드들을 스케줄링할 수 있다.

[0068] 도 5에 예시된 예에서, 디스패치된 작업부하 노드가 스케줄러(500)가 연관되는 CBB에 의해 실행될 때, 작업부하 인터페이스(502)는 입력 버퍼와 연관된 크레디트들을, 작업부하 인터페이스(502)가 크레디트들(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)을 수신했던 외부 디바이스에 전송한다. 작업부하 노드 디스패처(508)는 실행될, 스케줄 내의 추가적인 작업부하 노드들이 존재하는지를 추가적으로 결정한다. 스케줄 내에 추가적인 작업부하 노드들이 존재하는 경우, 작업부하 노드 디스패처(508)는 스케줄에서 다음 작업부하 노드를 디스패치한다.

[0069] 일부 예들에서, 예시적인 작업부하 노드 디스패처(508)는 디스패치를 위한 예시적인 수단을 구현한다. 디스패칭 수단은 도 8의 적어도 블록들(814, 816, 및 820)에 의해 구현되는 것과 같은 실행가능 명령어들에 의해 구현된다. 예를 들어, 도 8의 블록들(814, 816, 및 820)의 실행가능 명령어들은 도 9의 예에 예시된 예시적인 가속기(912) 및/또는 예시적인 프로세서(910)와 같은 적어도 하나의 프로세서 상에서 실행될 수 있다. 다른 예들에서, 디스패칭 수단은 하드웨어 로직, 하드웨어로 구현되는 상태 머신들, 로직 회로, 및/또는 하드웨어, 소프트웨어, 및/또는 펌웨어의 임의의 다른 조합에 의해 구현된다.

[0070] 본 명세서에 개시된 예들에서, 작업부하 인터페이스(502), 버퍼 크레디트 스토리지(504), 크레디트 비교기(506), 및 작업부하 노드 디스패처(508) 각각은 스케줄러(500)의 다른 요소들과 통신한다. 예를 들어, 작업부하 인터페이스(502), 버퍼 크레디트 스토리지(504), 크레디트 비교기(506), 및 작업부하 노드 디스패처(508)는 예시적인 통신 버스(510)를 통해 통신한다. 본 명세서에 개시된 일부 예들에서, 작업부하 인터페이스(502), 버퍼 크레디트 스토리지(504), 크레디트 비교기(506), 및 작업부하 노드 디스패처(508)는 임의의 적절한 유선 및/또는 무선 통신 시스템을 통해 통신할 수 있다. 추가적으로, 본 명세서에 개시된 일부 예들에서, 작업부하 인터페이스(502), 버퍼 크레디트 스토리지(504), 크레디트 비교기(506), 및 작업부하 노드 디스패처(508) 각각은 임의의 적절한 유선 및/또는 무선 통신 시스템을 통해 스케줄러(500) 외부의 임의의 컴포넌트와 통신할 수 있다.

[0071] 도 6은 도 5의 버퍼 크레디트 스토리지(504)의 추가 상세를 도시하는 예시적인 스케줄러(600)의 블록도이다. 스케줄러(600)는 도 3의 제1 스케줄러(326), 제2 스케줄러(328), 제3 스케줄러(330), 제4 스케줄러(332), 및/또는 제5 스케줄러(334), 및/또는 도 4의 제1 스케줄러(424), 제2 스케줄러(426), 제3 스케줄러(428) 및/또는 제4 스케줄러(430), 및/또는 도 5의 스케줄러(500), 및/또는 도 7의 제1 스케줄러(722), 제2 스케줄러(724), 제3 스케줄러(726), 및/또는 제4 스케줄러(728)의 예시적인 구현이다.

[0072] 도 6의 예에서, 스케줄러(600)는 예시적인 작업부하 인터페이스(502), 예시적인 버퍼 크레디트 스토리지(504), 예시적인 크레디트 비교기(506), 및 예시적인 작업부하 노드 디스패처(508)를 포함한다. 스케줄러(600)는, 스케줄러(600)가 연관되는 CBB가, 스케줄러(600)가 연관되는 CBB에 할당되었던 작업부하의 일부(예를 들어, 작업부하 노드)를 언제 실행할지를 결정하는 디바이스이다.

[0073] 도 6의 예시된 예에서, 작업부하 인터페이스(502)는 스케줄러(600), 버퍼 크레디트 스토리지(504) 및 작업부하 노드 디스패처(508) 외부의 하나 이상의 디바이스에 결합된다. 작업부하 인터페이스(502)는 스케줄러(600), 버퍼 크레디트 스토리지(504), 및/또는 작업부하 노드 디스패처(508) 외부의 다른 디바이스들과 통신하도록 구성되는 디바이스이다. 예를 들어, 작업부하 인터페이스(502)는 스케줄러(600)가 연관되는 CBB에 의해 실행될 작업부하 노드들을 수신하고/하거나 다르게는 획득할 수 있다. 추가적으로 또는 대안적으로, 작업부하 인터페이스(502)는 스케줄러(600) 외부의 하나 이상의 디바이스에 크레디트들을 송신하고 및/또는 그로부터 크레디트들을 수신할 수 있다. 또한, 작업부하 인터페이스(502)는 작업부하 노드로의 입력 버퍼들 및/또는 작업부하 노드

로부터의 출력 버퍼들에 대응하는 크레디트들을 버퍼 크레디트 스토리지(504) 내로 및/또는 밖으로 로딩할 수 있다.

[0074] 도 6에 예시된 예에서, 버퍼 크레디트 스토리지(504)는 작업부하 인터페이스(502), 크레디트 비교기(506), 및/또는 작업부하 노드 디스패처(508) 중 적어도 하나 사이의 공유 스토리지이다. 버퍼 크레디트 스토리지(504)는 스케줄러(500)에 로컬인 물리적 스토리지이다. 그러나, 다른 예들에서, 버퍼 크레디트 스토리지(504)는 스케줄러(500) 외부에 있을 수 있고 및/또는 다르게는 그에 대해 원격일 수 있다. 추가 예들에서, 버퍼 크레디트 스토리지(504)는 가상 스토리지일 수 있다. 도 5의 예에서, 버퍼 크레디트 스토리지(504)는 지속적 스토리지(예를 들어, ROM, PROM, EPROM, EEPROM 등)이다. 다른 예들에서, 버퍼 크레디트 스토리지(504)는 지속적 BIOS 또는 플래시 스토리지일 수 있다. 추가 예들에서, 버퍼 크레디트 스토리지(504)는 휘발성 메모리일 수 있다.

[0075] 도 6의 예에서, 버퍼 크레디트 스토리지(504)는 제1 작업부하 노드 WN[0], 제2 작업부하 노드 WN[1], 및 제n 작업부하 노드 WN[n]에 대응하는 행들을 포함하는 데이터 구조이다. 버퍼 크레디트 스토리지(504)는 제1 소비자에 대한 입력 버퍼(예를 들어, 소비자[0]), 제1 소비자에 대한 입력 버퍼(예를 들어, 소비자[1]), 제1 생산자에 대한 출력 버퍼(예를 들어, 생산자[0]), 및 제m 생산자에 대한 출력 버퍼(예를 들어, 생산자[m])에 대응하는 열들을 추가로 포함한다. 버퍼 크레디트 스토리지(504)는 각각의 작업부하 노드로의 입력 버퍼들 및/또는 각각의 작업부하 노드로부터의 출력 버퍼들에 대한 크레디트들의 임계 수에 대응하는 열을 추가로 포함한다.

[0076] 도 6의 예시된 예에서, 제1 작업부하 노드 WN[0], 제2 작업부하 노드 WN[1] 및 제n 작업부하 노드 WN[n] 각각은 스케줄러(600)가 연관되는 CBB에 할당된다. 버퍼 크레디트 스토리지(504)에서, 제1 작업부하 노드 WN[0], 제2 작업부하 노드 WN[1], 및 제n 작업부하 노드 WN[n]에 대응하는 행들과, 제1 소비자에 대한 입력 버퍼(예를 들어, 소비자[0]), 제1 소비자에 대한 입력 버퍼(예를 들어, 소비자[1]), 제1 생산자에 대한 출력 버퍼(예를 들어, 생산자[0]), 및 제m 생산자에 대한 출력 버퍼(예를 들어, 생산자[m])에 대응하는 열들 사이의 교차부는 그 버퍼에 대한 하나 이상의 외부 디바이스로부터 수신된 크레디트들의 수에 대응하는 필드들을 나타낸다. 더욱이, 각각의 작업부하 노드로의 입력 버퍼들 및/또는 각각의 작업부하 노드로부터의 출력 버퍼들에 대한 크레디트들의 임계 수에 대응하는 열은 스케줄러(600)가 연관되는 CBB가 각각의 작업부하 노드 상에서 동작할 수 있기 전에 버퍼에 대해 충족되어야 하는 크레디트들의 임계 수를 나타낸다.

[0077] 도 6의 예에서, 제1 작업부하 노드 WN[0], 제2 작업부하 노드 WN[1], 및 제n 작업부하 노드 WN[n]에 대응하는 행들과, 제1 소비자에 대한 입력 버퍼(예를 들어, 소비자[0]), 제1 소비자에 대한 입력 버퍼(예를 들어, 소비자[1])에 대응하는 열들 사이의 교차부에서의, 버퍼 크레디트 스토리지(504) 내의 필드들은 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)에 의해 제로의 값으로 초기화된다. 추가적으로, 제1 작업부하 노드 WN[0], 제2 작업부하 노드 WN[1], 및 제n 작업부하 노드 WN[n]에 대응하는 행들과, 제1 생산자에 대한 출력 버퍼(예를 들어, 생산자[0]), 및 제m 생산자에 대한 출력 버퍼(예를 들어, 생산자[m])에 대응하는 열들 사이의 교차점에서의, 버퍼 크레디트 스토리지(504)에서의 필드들은 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)에 의해, 연관된 버퍼에서 파티셔닝된 메모리의 양에 대응하는 값으로 초기화된다. 더욱이, 입력 버퍼들 및/또는 출력 버퍼들에 대한 크레디트들의 임계 수에 대응하는 열은 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322), 호스트 프로세서(306) 상에서 실행되는 소프트웨어 등)에 의해 초기화된다.

[0078] 도 6에 예시된 예에서, 크레디트 비교기(506)는 버퍼 크레디트 스토리지(504) 및 작업부하 노드 디스패처(508)에 결합된다. 크레디트 비교기(506)는, 스케줄러(600)가 연관되는 CBB에 할당된 작업부하 노드들로의 입력 버퍼들 및/또는 작업부하 노드들로부터의 출력 버퍼들에 대응하는 크레디트들의 임계 수가 수신되었는지를 결정하도록 구성되는 디바이스이다. 도 6의 예에서, 작업부하 노드 디스패처(508)는 작업부하 인터페이스(502), 버퍼 크레디트 스토리지(504), 크레디트 비교기(506), 및 스케줄러(600) 외부의 하나 이상의 디바이스에 결합된다. 작업부하 노드 디스패처(508)는, 예를 들어, 스케줄러(600)가 연관되는 CBB 상에서 실행될, 스케줄러(600)가 연관되는 CBB에 할당된 하나 이상의 작업부하 노드를 스케줄링하는 디바이스이다.

[0079] 도 6의 예에서, 동작 중에, 작업부하 인터페이스(502)가 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)로부터 작업부하 노드들을 수신하고/하거나 다르게는 획득할 때, 작업부하 인터페이스(502)는 작업부하 노드들을 작업부하 노드들에 대응하는 버퍼 크레디트 스토리지(504) 내의 각각의 필드들에 로딩한다. 또한, 크레디트 비교기(506)는 스케줄러(600)가 연관되는 CBB에 할당된 작업부하 노드를 선택한다.

[0080] 도 6의 예에서, 크레디트 비교기(506)는 스케줄러(600)가 선택된 작업부하 노드에 대한 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했는지를 결정한다. 예를 들어, 크레디트 비교기

(506)는 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)로부터 수신된 크레디트들의 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드를, 선택된 작업부하 노드로의 입력 버퍼에 대한 크레디트들의 임계 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드와 비교한다. 크레디트들의 임계 수는 스케줄러(600)가 연관되는 CBB가 소비자 작업부하 노드를 실행할 수 있기 전에 충족되어야 하는 입력 버퍼(예를 들어, 메모리(420) 내의 파티셔닝된 공간) 내의 데이터의 임계량에 대응한다. 스케줄러(600)가 크레디트들의 임계량을 수신하지 않았다면, 크레디트 비교기(506)는 스케줄러(600)가 연관되는 CBB에 할당된 다른 작업부하 노드 상에서 프로세스를 반복한다.

[0081] 도 6에 예시된 예에서, 스케줄러(600)가 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했다면, 크레디트 비교기(506)는 스케줄러(600)가 선택된 작업부하 노드에 대한 출력 버퍼에 데이터를 기입하기 위해 크레디트들의 임계량을 수신했는지를 결정한다. 예를 들어, 크레디트 비교기(506)는 선택된 작업부하 노드에 대한 출력 버퍼에 대한 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)로부터 수신된 크레디트들의 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드를, 출력 버퍼에 대한 크레디트들의 임계 수와 연관된 버퍼 크레디트 스토리지(504) 내의 필드와 비교한다. 크레디트들의 임계 수는 스케줄러(600)가 연관되는 CBB가 생산자 작업부하 노드를 실행할 수 있기 전에 충족되어야 하는 출력 버퍼(예를 들어, 메모리 내의 파티셔닝된 공간) 내의 공간의 임계량에 대응할 수 있다.

[0082] 도 6의 예에서, 스케줄러(600)가 크레디트들의 임계량을 수신하지 않았을 경우, 크레디트 비교기(506)는 스케줄러(600)가 연관되는 CBB에 할당된 다른 작업부하 노드 상에서 프로세스를 반복한다. 스케줄러(600)가 출력 버퍼에 데이터를 기입하기 위해 크레디트들의 임계량을 수신했다면, 크레디트 비교기(506)는 선택된 작업부하 노드가 실행할 준비가 된 것을 표시한다. 후속하여, 크레디트 비교기(506)는 스케줄러(600)가 연관되는 CBB에 할당된 추가적인 작업부하 노드들에 대해 이 프로세스를 반복한다.

[0083] 도 6의 예에서, 작업부하 노드 디스패처(508)는 스케줄러(600)가 연관되는 CBB 상에서 실행될, 스케줄러(600)가 연관되는 CBB에 할당된 하나 이상의 작업부하 노드들을 스케줄링하는 디바이스이다. 예를 들어, 스케줄러(600)가 연관되는 CBB에 할당된 작업부하 노드들이 분석된 후에, 작업부하 노드 디스패처(508)는 실행할 준비가 된 작업부하 노드들을 스케줄링한다. 예를 들어, 작업부하 노드 디스패처(508)는 라운드-로빈(round-robin) 스케줄과 같은 스케줄링 알고리즘에 기초하여 실행할 준비가 된 작업부하 노드들을 스케줄링한다. 작업부하 노드 디스패처(508)는 후속하여, 스케줄에 따라 작업부하 노드를 디스패치한다. 다른 예들에서, 작업부하 노드 디스패처(508)는 임의의 다른 적절한 중재 알고리즘을 이용하여, 실행할 준비가 된 작업부하 노드들을 스케줄링할 수 있다.

[0084] 도 6에 예시된 예에서, 디스패치된 작업부하 노드가 스케줄러(600)가 연관되는 CBB에 의해 실행될 때, 작업부하 인터페이스(502)는 입력 버퍼와 연관된 크레디트들을, 작업부하 인터페이스(502)가 크레디트들(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)을 수신했던 외부 디바이스에 전송한다. 작업부하 노드 디스패처(508)는 실행될, 스케줄 내의 추가적인 작업부하 노드들이 존재하는지를 추가적으로 결정한다. 스케줄 내에 추가적인 작업부하 노드들이 존재하는 경우, 작업부하 노드 디스패처(508)는 스케줄에서 다음 작업부하 노드를 디스패치한다.

[0085] 도 7은 버퍼들 및 파이프라이닝을 구현하는 이종 시스템의 가속기 상에서 실행하는 작업부하를 나타내는 예시적인 그래프(700)의 그래픽 예시이다. 예를 들어, 가속기는 제1 가속기(310a)이고, 이종 시스템은 도 3의 이종 시스템(304)이다. 예시적인 그래프(700)는 예시적인 제1 작업부하 노드(702)(WN[0]), 예시적인 제2 작업부하 노드(704)(WN[1]), 예시적인 제3 작업부하 노드(706)(WN[2]), 예시적인 제4 작업부하 노드(708)(WN[3]), 및 예시적인 제5 작업부하 노드(710)(WN[4])를 포함한다. 도 7의 예에서, 가속기는 작업부하 노드들을 다양한 CBB들에 할당하는 예시적인 크레디트 매니저(712)로부터의 스케줄에 기초하여 그래프(700)에 의해 표현되는 작업부하를 실행하도록 구성된다. 예를 들어, 크레디트 매니저(712) 및/또는 다른 컨트롤러는 제1 작업부하 노드(702)(WN[0])를 예시적인 제1 CBB(714), 제2 작업부하 노드(704)(WN[1])를 예시적인 제2 CBB(716)에, 제3 작업부하 노드(706)(WN[2])를 예시적인 제3 CBB(718)에, 제4 작업부하 노드(708)(WN[3])를 예시적인 제4 CBB(720)에, 그리고 제5 작업부하 노드(710)(WN[4])를 예시적인 제2 CBB(716)에 할당한다.

[0086] 도 7의 예에서, 예시적인 제1 CBB(714), 예시적인 제2 CBB(716), 예시적인 제3 CBB(718), 및 예시적인 제4 CBB(720) 각각은 예시적인 제1 스케줄러(722), 예시적인 제2 스케줄러(724), 예시적인 제3 스케줄러(726), 및 예시적인 제4 스케줄러(728)를 포함한다. 제1 스케줄러(722), 제2 스케줄러(724), 제3 스케줄러(726), 및 제4 스케줄러(728) 각각은 도 5의 스케줄러(500) 및/또는 도 6의 스케줄러(600)에 의해 구현될 수 있다.

[0087] 도 7의 예시된 예에서, 제1 작업부하 노드(702)(WN[0]) 및 제2 작업부하 노드(704)(WN[1])는 예시적인 제1 버퍼(730)와 연관된다. 제1 버퍼(730)는 제1 작업부하 노드(702)(WN[0])의 출력 버퍼 및 제2 작업부하 노드(704)(WN[1])로의 입력 버퍼이다. 제2 작업부하 노드(704)(WN[1]) 및 제3 작업부하 노드(706)(WN[2])는 예시적인 제2 버퍼(732)와 연관된다. 제2 버퍼(732)는 제2 작업부하 노드(704)(WN[1])의 출력 버퍼 및 제3 작업부하 노드(706)(WN[2])로의 입력 버퍼이다. 제4 작업부하 노드(708)(WN[3]) 및 제5 작업부하 노드(710)(WN[4])는 예시적인 제3 버퍼(734)와 연관된다. 제3 버퍼(734)는 제4 작업부하 노드(708)(WN[3])의 출력 버퍼 및 제5 작업부하 노드(710)(WN[4])로의 입력 버퍼이다. 제1 버퍼(730), 제2 버퍼(732) 및 제3 버퍼(734) 각각은 사이클릭 버퍼(cyclic buffer)에 의해 구현될 수 있다. 도 7의 예에서, 제1 버퍼(730), 제2 버퍼(732) 및 제3 버퍼(734) 각각은 가속기의 메모리의 5개 파티션을 포함하고, 그 각각은 데이터의 타일을 저장할 수 있다.

[0088] 도 7에 예시된 예에서, 제1 작업부하 노드(702)(WN[0])가 생산자 작업부하 노드이기 때문에, 크레디트 매니저(712)는 제1 버퍼(730)에 대한 5개의 크레디트로 제1 스케줄러(722)를 초기화한다. 유사하게, 제2 작업부하 노드(704)(WN[1])가 생산자 작업부하 노드이기 때문에, 크레디트 매니저(712)는 제2 버퍼(732)에 대한 5개의 크레디트로 제2 스케줄러(724)를 초기화한다. 추가적으로, 제4 작업부하 노드(708)(WN[3])가 생산자 작업부하 노드이기 때문에, 크레디트 매니저(712)는 제3 버퍼(734)에 대한 5개의 크레디트로 제4 스케줄러(728)를 초기화한다.

[0089] 제1 스케줄러(722), 제2 스케줄러(724), 및 제4 스케줄러(728) 각각에 제공되는 5개의 크레디트는 제1 버퍼(730), 제2 버퍼(732) 및 제3 버퍼(734)의 크기를 나타낸다. 추가적으로, 제2 작업부하 노드(704)(WN[1])가 또한 소비자 작업부하 노드이기 때문에, 크레디트 매니저(712)는 제1 버퍼(730)에 대한 제로 크레디트들로 제2 스케줄러(724)를 초기화한다. 더욱이, 제3 작업부하 노드(706)(WN[2])가 소비자 작업부하 노드이기 때문에, 크레디트 매니저(712)는 제2 버퍼(732)에 대한 제로 크레디트들로 제3 스케줄러(726)를 초기화한다. 또한, 제5 작업부하 노드(710)(WN[4])가 소비자 작업부하 노드이기 때문에, 크레디트 매니저(712)는 제3 버퍼(734)에 대한 제로 크레디트들로 제2 스케줄러(724)를 초기화한다.

[0090] 도 7의 예에서, 제1 스케줄러(722)가 제1 작업부하 노드(702)(WN[0])로의 입력 버퍼들 및 제1 작업부하 노드(702)(WN[0])로부터의 출력 버퍼들 둘 다에 대한 크레디트들의 임계 수를 수신했기 때문에, 제1 스케줄러(722)는 제1 작업부하 노드(702)(WN[0])를 디스패치하여 제1 CBB(714) 상에서 실행한다. 추가적으로, 제4 스케줄러(728)가 제4 작업부하 노드(708)(WN[3])로의 입력 버퍼들 및 제4 작업부하 노드(708)(WN[3])로부터의 출력 버퍼들 둘 다에 대한 크레디트들의 임계 수를 수신했기 때문에, 제4 스케줄러(728)는 제4 작업부하 노드(708)(WN[3])를 디스패치하여 제4 CBB(720) 상에서 실행한다. 제1 작업부하 노드(702)(WN[0])가 제1 CBB(714) 상에서 실행할 때, 제1 CBB(714)는 데이터를 제1 버퍼(730)에 송신한다. 유사하게, 제4 작업부하 노드(708)(WN[3])가 제4 CBB(720) 상에서 실행할 때, 제4 CBB(720)는 데이터를 제3 버퍼(734)에 송신한다.

[0091] 도 7에 예시된 예에서, 제1 CBB(714) 및 제4 CBB(720) 각각이 각각 제1 작업부하 노드(702)(WN[0]) 및 제4 작업부하 노드(708)(WN[3])와 연관된 데이터의 타일들을 송신하기 때문에, 제1 스케줄러(722) 및 제4 스케줄러(728)는 각각 제1 CBB(714) 및 제4 CBB(720)로부터 제1 버퍼(730) 및 제3 버퍼(734)에 송신된 데이터의 각각의 타일에 대한 크레디트들을 크레디트 매니저(712)에 송신한다. 크레디트 매니저(712)는 제1 스케줄러(722)로부터 수신된 크레디트들을 제2 스케줄러(724)에 송신하고, 제4 스케줄러(728)로부터 수신된 크레디트들을 제2 스케줄러(724)에 송신한다. 제4 CBB(720)가 제4 작업부하 노드(708)(WN[3])를 실행할 때, 제4 CBB(720)는 제3 버퍼(734)에 저장하기 위한 데이터의 2개의 타일을 생성한다. 유사하게, 제1 CBB(714)가 제1 작업부하 노드(702)(WN[0])를 실행할 때, 제1 CBB(714)는 제1 버퍼(730)에 저장하기 위한 데이터의 5개의 타일을 생성한다.

[0092] 도 7의 예에서, 제4 CBB(720)가 제4 작업부하 노드(708)(WN[3])를 실행하는 것은, 제1 CBB(714)가 제1 작업부하 노드(702)(WN[0])를 실행하는 것보다 더 빠르다. 제2 버퍼(732)에서 이용가능한 메모리가 존재하지만, 제5 작업부하 노드(710)(WN[4])가 의존하는 데이터는 제2 작업부하 노드(704)(WN[1])가 의존하는 데이터가 준비되기 전에 준비된 데이터이기 때문에, 제2 스케줄러(724)는 제2 작업부하 노드(704)(WN[1])와 대조적으로 제2 CBB(716) 상에서 실행하기 위해 제5 작업부하 노드(710)(WN[4])를 선택한다.

[0093] 도 7의 예시된 예에서, 제5 작업부하 노드(710)(WN[4])가 제2 CBB(716) 상에서 실행하고 제2 CBB(716)가 제3 버퍼(734)에 저장되는 데이터의 타일들을 소비할 때, 제2 스케줄러(724)는 제3 버퍼(734)로부터의, 제2 CBB(716)에 의해 소비되는 데이터의 각각의 타일에 대한, 제3 버퍼(734)와 연관된 크레디트들을 크레디트 매니저(712)에 다시 전송한다. 후속하여, 제1 버퍼(730) 및 제2 버퍼(732)에 대한 크레디트들의 임계량을 충족시켰다면, 제2 스케줄러(724)는 제2 작업부하 노드(704)(WN[1])를 디스패치하여 제2 CBB(716) 상에서 실행한다. 제

2 CBB(716)가 제2 작업부하 노드(704)(WN[1])와 연관된 데이터의 타일들을 생성하고, 제2 버퍼(732)에 데이터를 출력할 때, 제2 스케줄러(724)는 제2 CBB(716)로부터 제2 버퍼(732)에 송신된 데이터의 각각의 타일에 대한, 제2 버퍼(732)와 연관된 크레디트들을 크레디트 매니저(712)에 전송한다.

[0094] 도 7의 예에서, 제2 스케줄러(724)로부터 제2 버퍼(732)와 연관된 크레디트들을 수신했다면, 크레디트 매니저(712)는 제2 버퍼(732)와 연관된 크레디트들을 제3 스케줄러(726)에 전송한다. 제3 스케줄러(726)가 제2 버퍼(732)와 연관된 크레디트들의 임계량을 수신할 때, 제3 스케줄러(726)는 제3 작업부하 노드(706)(WN[2])를 디스패치하여 제3 CBB(718) 상에서 실행한다. 제3 CBB(718)가 제3 작업부하 노드(706)(WN[2])를 실행하고 제3 CBB(718)가 제2 버퍼(732)에 저장되는 데이터의 타일들을 소비할 때, 제3 스케줄러(726)는 제3 CBB(718)에 의해 소비되는, 제2 버퍼(732)로부터의 데이터의 각각의 타일에 대한, 제2 버퍼(732)와 연관된 크레디트들을 크레디트 매니저(712)에 전송한다.

[0095] 추가적인 또는 대안적인 예들에서, 제1 CBB(714)는 도 4의 콘볼루션 엔진(412)에 대응할 수 있고, 제1 스케줄러(722)는 도 4의 제1 스케줄러(424)에 대응할 수 있다. 일부 예들에서, 제2 CBB(716)는 도 4의 RNN 엔진(416)에 대응할 수 있고, 제2 스케줄러(724)는 도 4의 제3 스케줄러(428)에 대응할 수 있다. 추가 예들에서, 제3 CBB(718)는 도 4의 DMA 유닛(414)에 대응할 수 있고, 제3 스케줄러(726)는 도 4의 제2 스케줄러(426)에 대응할 수 있다. 일부 예들에서, 제4 CBB(720)는 도 4의 DSP(418)에 대응할 수 있고, 제4 스케줄러(728)는 도 4의 제4 스케줄러(430)에 대응할 수 있다.

[0096] 도 3의 제1 스케줄러(326), 제2 스케줄러(328), 제3 스케줄러(330), 제4 스케줄러(332), 및/또는 제5 스케줄러(334), 및/또는 도 4의 제1 스케줄러(424), 제2 스케줄러(426), 제3 스케줄러(428), 및/또는 제4 스케줄러(430), 및/또는 도 7의 제1 스케줄러(722), 제2 스케줄러(724), 제3 스케줄러(726), 및/또는 제4 스케줄러(728)를 구현하는 예시적인 방식이 도 5 및/또는 도 6에 예시되지만, 도 5 및/또는 도 6에 예시된 요소들, 프로세스들 및/또는 디바이스들 중 하나 이상은 조합, 분할, 재배열, 생략, 제거 및/또는 임의의 다른 방식으로 구현될 수 있다. 또한, 도 5의 예시적인 작업부하 인터페이스(502), 예시적인 버퍼 크레디트 스토리지(504), 예시적인 크레디트 비교기(506), 예시적인 작업부하 노드 디스패처(508), 예시적인 통신 버스(510), 및/또는 보다 일반적으로, 예시적인 스케줄러(500) 및/또는 도 6의 예시적인 스케줄러(600)는 하드웨어, 소프트웨어, 펌웨어 및/또는 하드웨어, 소프트웨어 및/또는 펌웨어의 임의의 조합에 의해 구현될 수 있다. 따라서, 예를 들어, 도 5의 예시적인 작업부하 인터페이스(502), 예시적인 버퍼 크레디트 스토리지(504), 예시적인 크레디트 비교기(506), 예시적인 작업부하 노드 디스패처(508), 예시적인 통신 버스(510), 및/또는 보다 일반적으로는, 예시적인 스케줄러(500) 및/또는 도 6의 예시적인 스케줄러(600) 중 임의의 것은, 하나 이상의 아날로그 또는 디지털 회로(들), 로직 회로들, 프로그래머블 프로세서(들), 프로그래머블 컨트롤러(들), 그래픽 처리 유닛(들)(GPU(들)), 디지털 신호 프로세서(들)(DSP(들)), 애플리케이션 특정 통합 회로(들)(ASIC(들)), 프로그래머블 로직 디바이스(들)(PLD(들)) 및/또는 필드 프로그래머블 로직 디바이스(들)(FPLD(들))에 의해 구현될 수 있다. 순전히 소프트웨어 및/또는 펌웨어 구현을 커버하기 위해 본 특허의 장치 또는 시스템 청구항들 중 임의의 것을 읽을 때, 도 5의 예시적인 작업부하 인터페이스(502), 예시적인 버퍼 크레디트 스토리지(504), 예시적인 크레디트 비교기(506), 예시적인 작업부하 노드 디스패처(508), 예시적인 통신 버스(510), 및/또는 보다 일반적으로, 예시적인 스케줄러(500) 및/또는 도 6의 예시적인 스케줄러(600) 중 적어도 하나는 이로써 소프트웨어 및/또는 펌웨어를 포함하는 메모리와 같은 저장 디스크 또는 비일시적 컴퓨터 판독가능 저장 디바이스, 디지털 다기능 디스크(DVD), 컴팩트 디스크(CD), 블루레이 디스크(Blu-ray disk) 등을 포함하는 것으로 명시적으로 정의된다. 또한, 도 5의 예시적인 스케줄러(500) 및/또는 도 6의 예시적인 스케줄러(600)는 도 5 및/또는 도 6에 도시된 것들에 더하여 또는 그 대신에 하나 이상의 요소, 프로세스 및/또는 디바이스를 포함할 수 있고/있거나, 예시된 요소들, 프로세스들 및 디바이스들 중 임의의 것 또는 이들 모두의 둘 이상을 포함할 수 있다. 본 명세서에서 사용되는 바와 같이, "통신하는"이라는 문구는, 그 변형들을 포함하고, 하나 이상의 중간 컴포넌트를 통한 직접 통신 및/또는 간접 통신을 포괄하고, 직접적인 물리적(예를 들어, 유선) 통신 및/또는 끊임없는 통신을 요구하는 것이 아니라, 오히려 주기적 간격들, 스케줄링된 간격들, 비주기적 간격들, 및/또는 1회 이벤트들에서의 선택적 통신을 추가적으로 포함한다.

[0097] 도 5의 스케줄러(500) 및/또는 도 6의 스케줄러(600)를 구현하기 위한 예시적인 하드웨어 로직, 머신 판독가능 명령어들, 하드웨어로 구현되는 상태 머신들, 및/또는 이들의 임의의 조합을 나타내는 흐름도가 도 8에 도시된다. 머신 판독가능 명령어들은 도 9와 관련하여 이하에서 논의되는 예시적인 프로세서 플랫폼(900)에 도시된 가속기(912) 및/또는 프로세서(910)와 같은 컴퓨터 프로세서에 의한 실행을 위한 실행가능 프로그램의 하나 이상의 실행가능 프로그램 또는 부분(들)일 수 있다. 프로그램은 CD-ROM, 플로피 디스크, 하드 드라이브, DVD,

블루레이 디스크, 또는 프로세서(910) 및/또는 가속기(912)와 연관된 메모리와 같은 비일시적 컴퓨터 판독가능 저장 매체 상에 저장되는 소프트웨어로 구현될 수 있지만, 전체 프로그램 및/또는 그 일부는 대안적으로 프로세서(910) 및/또는 가속기(912) 이외의 디바이스에 의해 실행될 수 있고 및/또는 펌웨어 또는 전용 하드웨어로 구현될 수 있다. 또한, 예시적인 프로그램이 도 8에 예시된 흐름도를 참조하여 설명되었지만, 도 5의 예시적인 스케줄러(500) 및/또는 도 6의 스케줄러(600)를 구현하는 많은 다른 방법들이 대안적으로 사용될 수 있다. 예를 들어, 블록들의 실행 순서는 변경될 수 있고/있거나, 설명된 블록들 중 일부는 변경, 제거 또는 결합될 수 있다. 추가적으로 또는 대안적으로, 블록들 중 임의의 것 또는 전부는 소프트웨어 또는 펌웨어를 실행하지 않고 대응하는 동작을 수행하도록 구조화된 하나 이상의 하드웨어 회로(예를 들어, 이산 및/또는 통합 아날로그 및/또는 디지털 회로, FPGA, ASIC, 비교기, 연산 증폭기(op-amp), 로직 회로 등)에 의해 구현될 수 있다.

[0098] 본 명세서에 설명된 머신 판독가능 명령어들은 압축 포맷, 암호화 포맷, 단편화된 포맷, 컴파일된 포맷, 실행가능 포맷, 패키징된 포맷, 등 중 하나 이상으로 저장될 수 있다. 본 명세서에 설명된 바와 같은 머신 판독가능 명령어들은 머신 실행가능 명령어들을 생성, 제조, 및/또는 생산하기 위해 이용될 수 있는 데이터(예를 들어, 명령어들의 부분들, 코드, 코드의 표현들 등)로서 저장될 수 있다. 예를 들어, 머신 판독가능 명령어들은 단편화되고 하나 이상의 저장 디바이스 및/또는 컴퓨팅 디바이스(예를 들어, 서버들) 상에 저장될 수 있다. 머신 판독가능한 명령어들은, 그것들을 컴퓨팅 디바이스 및/또는 다른 머신에 의해 직접 판독가능하고, 해석가능하고, 및/또는 실행가능하게 하기 위해, 설치, 수정, 적응, 업데이트, 조합, 보충, 구성, 복호화, 압축해제, 언패킹, 분배, 재할당, 컴파일 등 중 하나 이상을 요구할 수 있다. 예를 들어, 머신 판독가능 명령어들은 개별적으로 압축되고, 암호화되고, 개별적인 컴퓨팅 디바이스들 상에 저장되는 다수의 부분에 저장될 수 있고, 여기서, 부분들은 복호화, 압축해제, 및 조합될 때, 본 명세서에 설명된 것과 같은 프로그램을 구현하는 실행가능 명령어들의 세트를 형성한다.

[0099] 다른 예에서, 머신 판독가능 명령어들은 컴퓨터에 의해 판독될 수 있는 상태로 저장되지만, 특정 컴퓨팅 디바이스 또는 다른 디바이스 상에서 명령어들을 실행하기 위해 라이브러리(예를 들어, 동적 링크 라이브러리(DLL)), 소프트웨어 개발 키트(SDK), 애플리케이션 프로그래밍 인터페이스(API) 등의 추가를 요구할 수 있다. 다른 예에서, 머신 판독가능 명령어들은 머신 판독가능 명령어들 및/또는 대응하는 프로그램(들)이 전체적으로 또는 부분적으로 실행될 수 있기 전에 구성될 필요가 있을 수 있다(예를 들어, 설정들이 저장됨, 데이터가 입력됨, 네트워크 어드레스들이 기록됨 등). 따라서, 개시된 머신 판독가능 명령어들 및/또는 대응하는 프로그램(들)은, 저장되거나 또는 다르게는 정지 상태에 있거나 또는 수송 중인 경우 머신 판독가능 명령어들 및/또는 프로그램(들)의 특정 포맷 또는 상태에 관계없이 이러한 머신 판독가능 명령어들 및/또는 프로그램(들)을 포괄하도록 의도된다.

[0100] 본 명세서에 설명된 머신 판독가능 명령어들은 임의의 과거, 현재 또는 미래의 명령어 언어, 스크립팅 언어, 프로그래밍 언어 등에 의해 표현될 수 있다. 예를 들어, 머신 판독가능 명령어들은 다음 언어들 중 임의의 것을 사용하여 표현될 수 있다: C, C++, Java, C#, Perl, Python, JavaScript, HTML(HyperText Markup Language), SQL(Structured Query Language), Swift 등.

[0101] 위에서 언급된 바와 같이, 도 8의 예시적인 프로세스들은, 정보가 임의의 지속 기간 동안(예를 들어, 연장된 시간 기간 동안, 영구적으로, 짧은 순간 동안, 일시적으로 버퍼링하는 동안, 및/또는 정보의 캐싱 동안) 저장되는, 하드 디스크 드라이브, 플래시 메모리, 판독 전용 메모리, 컴팩트 디스크, 디지털 다기능 디스크, 캐시, 랜덤 액세스 메모리 및/또는 임의의 다른 저장 디바이스 또는 저장 디스크와 같은 비일시적 컴퓨터 및/또는 머신 판독가능 매체 상에 저장되는 실행가능 명령어들(예를 들어, 컴퓨터 및/또는 머신 판독가능 명령어들)을 사용하여 구현될 수 있다. 본 명세서에서 사용되는 바와 같이, 비일시적 컴퓨터 판독가능 매체라는 용어는 임의의 타입의 컴퓨터 판독가능 저장 디바이스 및/또는 저장 디스크를 포함하고 전파 신호들을 배제하고 송신 매체를 배제하기 위해 명백히 정의된다.

[0102] "포함하는(including)" 및 "포함하는(comprising)"(및 그의 모든 형태들 및 시제들)은 개방형 용어(open ended term)들인 것으로 본 명세서에서 사용된다. 따라서, 청구항이 전제부로서 또는 임의의 종류의 청구항 기재 내에 있는 임의의 형태의 "포함하다(include)" 또는 "포함하다(comprise)"의 임의의 형태(예를 들어, 포함하다(comprises), 포함하다(includes), 포함하는(comprising), 포함하는(including), 갖는(having) 등)를 이용할 때마다, 추가적인 요소들, 용어들 등이 대응하는 청구항 또는 기재의 범위 밖에 속하지 않고서 존재할 수 있다는 것을 이해해야 한다. 본 명세서에 사용되는 바와 같이, 예를 들어, 청구항의 전제부에서 연결어(transition term)로서 "적어도(at least)"이라는 표현이 사용될 때, 그것은 "포함하는(comprising)" 및 "포함하는(including)"이라는 용어가 개방형(open ended)인 것과 동일한 방식으로 개방형이다. 용어 "및/또는"은, 예를

들어, A, B, 및/또는 C와 같은 형태로 사용될 때, (1) A 단독, (2) B 단독, (3) C 단독, (4) A와 B, (5) A와 C, (6) B와 C, 및 (7) A와 B와 C의 임의의 조합 또는 서브세트를 지칭한다. 구조들, 컴포넌트들, 아이템들, 객체들 및/또는 사물들을 설명하는 것의 맥락에서 본 명세서에서 사용되는 바와 같이, "A 및 B 중 적어도 하나"라는 문구는 (1) 적어도 하나의 A, (2) 적어도 하나의 B, 및 (3) 적어도 하나의 A 및 적어도 하나의 B 중 임의의 것을 포함하는 구현들을 지칭하기 위해 의도된다. 유사하게, 구조들, 컴포넌트들, 아이템들, 객체들 및/또는 사물들을 설명하는 맥락에서 본 명세서에서 사용되는 바와 같이, "A 또는 B 중 적어도 하나"라는 문구는 (1) 적어도 하나의 A, (2) 적어도 하나의 B, 및 (3) 적어도 하나의 A 및 적어도 하나의 B 중 임의의 것을 포함하는 구현들을 지칭하기 위해 의도된다. 처리들, 명령어들, 액션들, 활동들 및/또는 단계들의 수행 또는 실행을 설명하는 맥락에서 본 명세서에서 사용되는 바와 같이, "A 및 B 중 적어도 하나"라는 문구는 (1) 적어도 하나의 A, (2) 적어도 하나의 B, 및 (3) 적어도 하나의 A 및 적어도 하나의 B 중 임의의 것을 포함하는 구현들을 지칭하기 위해 의도된다. 유사하게, 처리들, 명령어들, 액션들, 활동들 및/또는 단계들의 수행 또는 실행을 설명하는 맥락에서 본 명세서에서 사용되는 바와 같이, "A 또는 B 중 적어도 하나"라는 문구는 (1) 적어도 하나의 A, (2) 적어도 하나의 B, 및 (3) 적어도 하나의 A 및 적어도 하나의 B 중 임의의 것을 포함하는 구현들을 지칭하기 위해 의도된다.

[0103] 본 명세서에서 사용되는 바와 같이, 단수 언급들(예를 들어, "a", "an", "제1(first)", "제2(second)" 등)은 복수를 배제하지 않는다. 본 명세서에서 사용되는 바와 같이, 용어 단수 표현("a" 또는 "an") 엔티티는 그 엔티티 중 하나 이상을 지칭한다. 용어 단수 표현 "a"(또는 "an"), "하나 이상" 및 "적어도 하나"는 본 명세서에서 상호교환가능하게 사용될 수 있다. 또한, 개별적으로 열거되지만, 복수의 수단, 요소들 또는 방법 액션들은 단일 유닛 또는 프로세서에 의해 구현될 수 있다. 추가적으로, 개별적 특징들이 상이한 예들 및 청구항들 내에 포함될 수 있지만, 이들은 가능하게는 조합될 수 있으며, 상이한 예들 및 청구항들 내의 포함은 특징들의 조합이 가능하지 않고 및/또는 유리하지 않다는 것을 암시하지 않는다.

[0104] 도 8은 도 5의 스케줄러(500) 및/또는 도 6의 스케줄러(600)를 구현하기 위해 실행될 수 있는 머신 판독가능 명령어들에 의해 구현될 수 있는 프로세스 800을 나타내는 흐름도이다. 프로세스 800은 블록 802에서 시작하고, 여기서 작업부하 인터페이스(502)는, 버퍼 크레디트 스토리지(504)에, 스케줄러(500) 및/또는 스케줄러(600)가 연관되는 CBB에 할당된 작업부하 노드들로의 입력 버퍼들 및/또는 작업부하 노드들로부터의 출력 버퍼들에 대응하는 크레디트들을 로딩한다.

[0105] 도 8에 예시된 예에서, 프로세스 800은 블록 804에서 계속되는데, 여기서 크레디트 비교기(506)는 스케줄러(500) 및/또는 스케줄러(600)가 연관되는 CBB에 할당된 작업부하 노드를 선택한다. 블록 806에서, 크레디트 비교기(506)는 스케줄러(500) 및/또는 스케줄러(600)가 선택된 작업부하 노드에 대한 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했는지를 결정한다. 예를 들어, 크레디트 비교기(506)는 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)로부터 수신된 크레디트들의 수와 연관된 어레이 또는 다른 데이터 구조 내의 필드를, 선택된 작업부하 노드로의 입력 버퍼에 대한 크레디트들 임계 수와 연관된 어레이 또는 다른 데이터 구조 내의 필드와 비교한다. 크레디트 비교기(506)가 스케줄러(500) 및/또는 스케줄러(600)가 선택된 작업부하 노드에 대한 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신하지 않았다고 결정하면(블록 806: 아니오), 프로세스 800은 블록 812로 진행한다.

[0106] 도 8의 예에서, 크레디트 비교기(506)가 스케줄러(500) 및/또는 스케줄러(600)가 입력 버퍼에 저장되는 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했다고 결정하면(블록 806: 예), 프로세스 800은 블록 808로 진행한다. 블록 808에서, 크레디트 비교기(506)는 스케줄러(500) 및/또는 스케줄러(600)가 선택된 작업부하 노드에 대한 출력 버퍼에 데이터를 기입하기 위해 크레디트들의 임계량을 수신했는지를 결정한다. 예를 들어, 크레디트 비교기(506)는 선택된 작업부하 노드에 대한 출력 버퍼에 대한 외부 디바이스(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)로부터 수신된 크레디트들의 수와 연관된 어레이 또는 다른 데이터 구조의 필드를, 출력 버퍼에 대한 크레디트들의 임계 수와 연관된 어레이 또는 다른 데이터 구조 내의 필드와 비교한다. 크레디트 비교기(506)가 스케줄러(500) 및/또는 스케줄러(600)가 크레디트들의 임계량을 수신하지 않았다고 결정하면(블록 808: 아니오), 프로세스 800은 블록 812로 진행한다. 크레디트 비교기(506)가 스케줄러(500) 및/또는 스케줄러(600)가 출력 버퍼에 데이터를 기입하기 위해 크레디트들의 임계량을 수신했다고 결정하면(블록 808: 예), 블록 810에서 크레디트 비교기(506)는 선택된 작업부하 노드가 실행할 준비가 된 것을 표시한다.

[0107] 도 8의 예에서, 블록 812에서, 크레디트 비교기(506)는 처리될 추가적인 작업부하 노드가 존재하는지를 결정한다. 크레디트 비교기(506)가 처리할 추가적인 작업부하 노드가 존재한다고 결정하면(블록 812: 예), 크레디트 비교기(506)는 추가적인 작업부하 노드를 선택하고 프로세스 800은 블록 806으로 진행한다. 크레디트 비교기

(506)가 처리할 추가적인 작업부하 노드가 존재하지 않는다고 결정하면(블록 812: 아니오), 프로세스 800은 블록 814로 진행한다.

[0108] 도 8의 예시된 예에서, 블록 814에서, 작업부하 노드 디스패처(508)는 실행 준비가 된 작업부하 노드들을 스케줄링한다. 블록 816에서, 작업부하 노드 디스패처(508)는 스케줄에 따라 작업부하 노드를 디스패치한다. 블록 818에서, 디스패치된 작업부하 노드가 스케줄러(500) 및/또는 스케줄러(600)가 연관되는 CBB에 의해 실행될 때, 작업부하 인터페이스(502)는 입력 버퍼와 연관된 크레디트들을, 작업부하 인터페이스(502)가 크레디트들(예를 들어, 크레디트 매니저(408), 컨트롤러(322) 등)을 수신했던 외부 디바이스에 전송한다.

[0109] 도 8에 예시된 예에서, 블록 820에서, 작업부하 노드 디스패처(508)는 실행될, 스케줄 내의 추가적인 작업부하 노드들이 존재하는지를 결정한다. 작업부하 노드 디스패처(508)가 스케줄 내에 추가적인 작업부하 노드들이 존재한다고 결정하면(블록 820: 예), 프로세스 800은 블록 816으로 진행한다. 작업부하 노드 디스패처(508)가 스케줄 내에 추가적인 작업부하 노드들이 존재하지 않는다고 결정하면(블록 820: 아니오), 프로세스 800은 블록 822로 진행한다.

[0110] 도 8의 예에서, 블록 822에서, 작업부하 인터페이스(502)는 동작을 계속할지를 결정한다. 예를 들어, 작업부하 인터페이스(502)로 하여금 동작을 계속하기로 결정하게 할 조건은 추가적인 작업부하 노드들을 수신하는 것을 포함한다. 작업부하 인터페이스(502)가 동작을 계속하기로 결정하는 경우(블록 822: 예), 프로세스 800은 블록 802로 진행한다. 작업부하 인터페이스(502)가 동작을 계속하지 않기로 결정하는 경우(블록 822: 아니오), 프로세스 800은 종료된다.

[0111] 도 9는 도 5의 스케줄러(500) 및/또는 도 6의 스케줄러(600)의 하나 이상의 인스턴스화를 구현하기 위해 도 8의 명령어들을 실행하도록 구조화된 예시적인 프로세서 플랫폼(900)의 블록도이다. 프로세서 플랫폼(900)은, 예를 들어, 서버, 개인용 컴퓨터, 워크스테이션, 셀프-러닝 머신(self-learning machine)(예를 들어, 신경 네트워크), 모바일 디바이스(예를 들어, 셀 폰, 스마트폰, iPad™와 같은 태블릿), PDA(personal digital assistant), 인터넷 어플라이언스, DVD 플레이어, CD 플레이어, 디지털 비디오 레코더, 블루레이 플레이어, 게이밍 콘솔, 개인용 비디오 레코더, 셋톱 박스, 헤드셋 또는 다른 웨어러블 디바이스, 또는 임의의 다른 타입의 컴퓨팅 디바이스일 수 있다.

[0112] 예시된 예의 프로세서 플랫폼(900)은 프로세서(910) 및 가속기(912)를 포함한다. 예시된 예의 프로세서(910)는 하드웨어이다. 예를 들어, 프로세서(910)는, 임의의 원하는 패밀리 또는 제조자로부터의 하나 이상의 통합 회로, 로직 회로들, 마이크로프로세서들, GPU들, DSP들, 또는 컨트롤러들에 의해 구현될 수 있다. 하드웨어 프로세서는 반도체 기반(예를 들어, 실리콘 기반) 디바이스일 수 있다. 추가적으로, 가속기(912)는, 예를 들어, 임의의 원하는 패밀리 또는 제조자로부터의 하나 이상의 통합 회로, 로직 회로들, 마이크로프로세서들, GPU들, DSP들, FPGA들, VPU들, 컨트롤러들, 및/또는 다른 CBB들에 의해 구현될 수 있다. 예시된 예의 가속기(912)는 하드웨어이다. 하드웨어 가속기는 반도체 기반(예를 들어, 실리콘 기반) 디바이스일 수 있다. 이 예에서, 가속기(912)는 예시적인 콘볼루션 엔진(312), 예시적인 RNN 엔진(314), 예시적인 메모리(316), 예시적인 MMU(318), 예시적인 DSP(320), 예시적인 컨트롤러(322), 및 예시적인 DMA 유닛(324)을 구현한다. 또한, 예시적인 콘볼루션 엔진(312), 예시적인 RNN 엔진(314), 예시적인 DMA 유닛(324), 예시적인 DSP(320), 및 예시적인 컨트롤러(322) 각각은 예시적인 제1 스케줄러(326), 예시적인 제2 스케줄러(328), 예시적인 제3 스케줄러(330), 예시적인 제4 스케줄러(332), 및 예시적인 제5 스케줄러(334)를 각각 포함한다. 도 9의 예에서, 예시적인 제1 스케줄러(326), 예시적인 제2 스케줄러(328), 예시적인 제3 스케줄러(330), 예시적인 제4 스케줄러(332), 및 예시적인 제5 스케줄러(334) 각각은 예시적인 작업부하 인터페이스(502), 예시적인 버퍼 크레디트 스토리지(504), 예시적인 크레디트 비교기(506), 예시적인 작업부하 노드 디스패처(508), 및/또는 보다 일반적으로는 스케줄러(500)를 포함한다.

[0113] 추가적인 또는 대안적 예들에서, 프로세서(910)는 예시적인 콘볼루션 엔진(312), 예시적인 RNN 엔진(314), 예시적인 메모리(316), 예시적인 MMU(318), 예시적인 DSP(320), 예시적인 컨트롤러(322), 및 예시적인 DMA 유닛(324)을 구현한다. 또한, 이러한 추가적인 또는 대안적 예들에서, 예시적인 콘볼루션 엔진(312), 예시적인 RNN 엔진(314), 예시적인 DMA 유닛(324), 예시적인 DSP(320), 및 예시적인 컨트롤러(322) 각각은 예시적인 제1 스케줄러(326), 예시적인 제2 스케줄러(328), 예시적인 제3 스케줄러(330), 예시적인 제4 스케줄러(332), 및 예시적인 제5 스케줄러(334)를 각각 포함한다. 이러한 추가적인 또는 대안적 예들에서, 예시적인 제1 스케줄러(326), 예시적인 제2 스케줄러(328), 예시적인 제3 스케줄러(330), 예시적인 제4 스케줄러(332), 및 예시적인 제5 스케줄러(334) 각각은 예시적인 작업부하 인터페이스(502), 예시적인 버퍼 크레디트 스토리지(504), 예시적인 크레

디트 비교기(506), 예시적인 작업부하 노드 디스패처(508), 및/또는 보다 일반적으로는 스케줄러(500)를 포함한다.

[0114] 예시된 예의 프로세서(910)는 로컬 메모리(911)(예를 들어, 캐시)를 포함한다. 예시된 예의 프로세서(910)는, 버스(918)를 통해, 휘발성 메모리(914) 및 비휘발성 메모리(916)를 포함하는 메인 메모리와 통신한다. 또한, 예시된 예의 가속기(912)는 로컬 메모리(913)(예를 들어, 캐시)를 포함한다. 예시된 예의 가속기(912)는 버스(918)를 통해 휘발성 메모리(914) 및 비휘발성 메모리(916)를 포함하는 메인 메모리와 통신한다. 휘발성 메모리(914)는 SDRAM(Synchronous Dynamic Random Access Memory), DRAM(Dynamic Random Access Memory), RDRAM® (RAMBUS® Dynamic Random Access Memory) 및/또는 임의의 다른 타입의 랜덤 액세스 메모리 디바이스에 의해 구현될 수 있다. 비휘발성 메모리(916)는 플래시 메모리 및/또는 임의의 다른 원하는 타입의 메모리 디바이스에 의해 구현될 수 있다. 메인 메모리(914, 916)로의 액세스는 메모리 컨트롤러에 의해 제어된다.

[0115] 예시된 예의 프로세서 플랫폼(900)은 인터페이스 회로(920)를 또한 포함한다. 인터페이스 회로(920)는 이더넷 인터페이스, USB(universal serial bus), Bluetooth® 인터페이스, NFC(near field communication) 인터페이스, 및/또는 PCI 익스프레스 인터페이스와 같은 임의의 타입의 인터페이스 표준에 의해 구현될 수 있다.

[0116] 예시된 예에서, 하나 이상의 입력 디바이스(922)가 인터페이스 회로(920)에 연결된다. 입력 디바이스(들)(922)는 사용자가 프로세서(910) 및/또는 가속기(912)에 데이터 및/또는 커맨드들을 입력하는 것을 허용한다. 입력 디바이스(들)는, 예를 들어, 오디오 센서, 마이크로폰, 카메라(스틸 또는 비디오), 키보드, 버튼, 마우스, 터치스크린, 트랙-패드, 트랙볼, 이소포인트(isopoint) 및/또는 음성 인식 시스템에 의해 구현될 수 있다.

[0117] 하나 이상의 출력 디바이스(924)가 예시된 예의 인터페이스 회로(920)에 또한 연결된다. 출력 디바이스들(924)은, 예를 들어, 디스플레이 디바이스들(예를 들어, 발광 다이오드(LED), 유기 발광 다이오드(OLED), 액정 디스플레이(LCD), 음극선관 디스플레이(CRT), 인-플레이스 스위칭(IPS) 디스플레이, 터치스크린 등), 촉각 출력 디바이스, 프린터 및/또는 스피커에 의해 구현될 수 있다. 따라서, 예시된 예의 인터페이스 회로(920)는 통상적으로 그래픽 드라이버 카드, 그래픽 드라이버 칩 및/또는 그래픽 드라이버 프로세서를 포함한다.

[0118] 예시된 예의 인터페이스 회로(920)는 또한 네트워크(926)를 통해 외부 머신들(예를 들어, 임의의 종류의 컴퓨팅 디바이스들)과의 데이터 교환을 용이하게 하기 위한 송신기, 수신기, 송수신기, 모뎀, 주거용 게이트웨이, 무선 액세스 포인트, 및/또는 네트워크 인터페이스와 같은 통신 디바이스를 포함한다. 통신은, 예를 들어, 이더넷 연결, 디지털 가입자 회선(DSL) 연결, 전화선 연결, 동축 케이블 시스템, 위성 시스템, 라인-오브-사이트(line-of-site) 무선 시스템, 셀룰러 전화 시스템 등을 통할 수 있다.

[0119] 예시된 예의 프로세서 플랫폼(900)은 또한 소프트웨어 및/또는 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스(928)를 포함한다. 이러한 대용량 저장 디바이스들(928)의 예들은 플로피 디스크 드라이브들, 하드 드라이브 디스크들, 컴팩트 디스크 드라이브들, 블루레이 디스크 드라이브들, RAID(redundant array of independent disks) 시스템들, 및 DVD(digital versatile disk) 드라이브들을 포함한다.

[0120] 도 8의 머신 실행가능 명령어들(932)은 대용량 저장 디바이스(928)에, 휘발성 메모리(914)에, 비휘발성 메모리(916)에, 및/또는 CD 또는 DVD와 같은 이동식 비일시적 컴퓨터 판독가능 저장 매체 상에 저장될 수 있다.

[0121] 전술한 것으로부터, 작업부하의 정적 매핑의 비순차적 파이프라이닝된 실행을 가능하게 하는 예시적인 방법들, 장치 및 제조 물품들이 개시되었다는 것이 이해될 것이다. 더욱이, 작업부하 노드가 의존하는 데이터가 이용가능하고, 작업부하 노드를 실행하는 것에 의해 생성된 출력을 저장하기 위해 이용가능한 충분한 메모리가 존재할 때, 계산 빌딩 블록이 작업부하 노드들을 실행하는 것을 허용하는 예시적인 방법들, 장치들, 및 제조 물품들이 개시되었다. 추가적으로, 본 명세서에 개시된 예들은 작업부하 노드들이 스케줄 및/또는 다른 순서화와는 독립적으로 할당되는 계산 빌딩 블록들에 의해 작업부하 노드들이 실행되는 것을 허용한다. 개시된 방법들, 장치 및 제조 물품들은 처리 디바이스의 이용을 증가시킴으로써 컴퓨팅 디바이스를 사용하는 효율을 개선한다. 더욱이, 본 명세서에 개시된 예시적인 방법들, 장치 및 제조 물품들은 작업부하를 처리 및/또는 다르게는 실행하기 위해 처리 디바이스에 의해 이용되는 계산 사이클들의 수를 감소시킨다. 개시된 방법들, 장치 및 제조 물품들은 이에 따라 컴퓨터의 기능에서의 하나 이상의 개선(들)에 관한 것이다.

[0122] 작업부하의 정적 매핑의 비순차적 파이프라이닝된 실행을 가능하게 하기 위한 예시적인 방법들, 장치, 시스템들, 및 제조 물품들이 본 명세서에 개시된다. 추가 예들 및 이들의 조합들은 다음을 포함한다: 예 1은 장치를 포함하고, 이 장치는: 제1 수의 크레디트들을 메모리에 로딩하기 위한 인터페이스, 크레디트들의 제1 수

를, 버퍼 내의 메모리 가용성과 연관된 크레디트들의 임계 수와 비교하기 위한 비교기, 및 크레디트들의 제1 수가 크레디트들의 임계 수를 충족시킬 때, 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 작업부하의 작업부하 노드를 선택하기 위한 디스패처를 포함한다.

[0123] 예 2는 예 1의 장치를 포함하고, 인터페이스는, 인터페이스가 크레디트 매니저로부터 제1 수의 크레디트들을 수신할 때, 제1 수의 크레디트들을 메모리에 로딩하고, 작업부하 노드와 연관된 데이터의 하나 이상의 타일이 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록으로부터 버퍼에 송신될 때, 버퍼에 송신된 각각의 타일에 대한 크레디트를 크레디트 매니저에 송신한다.

[0124] 예 3은 예 1의 장치를 포함하고, 버퍼는 작업부하 노드와 연관된 출력 버퍼이고, 제1 수의 크레디트들은 출력 버퍼에 대응하고, 크레디트들의 임계 수는 출력 버퍼에서 메모리의 임계량에 대응한다.

[0125] 예 4는 예 1의 장치를 포함하고, 버퍼는 작업부하 노드와 연관된 입력 버퍼이고, 제1 수의 크레디트들은 입력 버퍼에 대응하고, 크레디트들의 임계 수는 입력 버퍼에서의 데이터의 임계량에 대응한다.

[0126] 예 5는 예 1의 장치를 포함하고, 버퍼는 제1 버퍼이고, 크레디트들의 임계 수는 크레디트들의 제1 임계 수이고, 비교기는 크레디트들의 제2 수를 제2 버퍼에서의 메모리 가용성과 연관된 크레디트들의 제2 임계 수와 비교하기 위한 것이고, 디스패처는 (1) 크레디트들의 제1 수가 크레디트들의 제1 임계 수를 충족시키고 (2) 크레디트들의 제2 수가 크레디트들의 제2 임계 수를 충족시킬 때, 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 작업부하 노드를 선택하기 위한 것이다.

[0127] 예 6은 예 5의 장치를 포함하고, 제2 버퍼는 작업부하 노드와 연관된 입력 버퍼이고, 제2 수의 크레디트들은 입력 버퍼에 대응하고, 크레디트들의 제2 임계 수는 입력 버퍼에서의 데이터의 임계량에 대응한다.

[0128] 예 7은 예 1의 장치를 포함하고, 크레디트들의 임계 수는 크레디트들의 제1 임계 수이고, 작업부하 노드는 제1 작업부하 노드이고, (1) 크레디트들의 제1 수가 크레디트들의 제1 임계 수를 충족시키고 (2) 크레디트들의 제2 수가 크레디트들의 제2 임계 수를 충족시킬 때, 디스패처는 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 제1 작업부하 노드 및 제2 작업부하 노드를 스케줄링하기 위한 것이다.

[0129] 예 8은, 실행될 때, 적어도 하나의 프로세서로 하여금, 적어도 제1 수의 크레디트들을 메모리에 로딩하게 하고, 크레디트들의 제1 수를 버퍼에서의 메모리 가용성과 연관된 크레디트들의 임계 수와 비교하게 하고, 크레디트들의 제1 수가 크레디트들의 임계 수를 충족시킬 때, 계산 빌딩 블록에서 실행될 작업부하의 작업부하 노드를 선택하게 하는 명령어들을 포함하는 비일시적 컴퓨터 판독가능 저장 매체를 포함한다.

[0130] 예 9는 예 8의 비일시적 컴퓨터 판독가능 저장 매체를 포함하고, 명령어들은 실행될 때, 적어도 하나의 프로세서로 하여금, 제1 수의 크레디트들이 크레디트 매니저로부터 수신될 때, 제1 수의 크레디트들을 메모리에 로딩하게 하고, 작업부하 노드와 연관된 데이터의 하나 이상의 타일이 계산 빌딩 블록으로부터 버퍼에 송신될 때, 버퍼에 송신된 각각의 타일에 대한 크레디트를 크레디트 매니저에 송신하게 한다.

[0131] 예 10은 예 8의 비일시적 컴퓨터 판독가능 저장 매체를 포함하고, 버퍼는 작업부하 노드와 연관된 출력 버퍼이고, 제1 수의 크레디트들은 출력 버퍼에 대응하고, 크레디트들의 임계 수는 출력 버퍼에서의 메모리의 임계량에 대응한다.

[0132] 예 11은 예 8의 비일시적 컴퓨터 판독가능 저장 매체를 포함하고, 버퍼는 작업부하 노드와 연관된 입력 버퍼이고, 제1 수의 크레디트들은 입력 버퍼에 대응하고, 크레디트들의 임계 수는 입력 버퍼에서의 데이터의 임계량에 대응한다.

[0133] 예 12는 예 8의 비일시적 컴퓨터 판독가능 저장 매체를 포함하고, 버퍼는 제1 버퍼이고, 크레디트들의 임계 수는 크레디트들의 제1 임계 수이고, 명령어들은 실행될 때, 적어도 하나의 프로세서로 하여금 크레디트들의 제2 수를 제2 버퍼에서의 메모리 가용성과 연관된 크레디트들의 제2 임계 수와 비교하게 하고, (1) 크레디트들의 제1 수가 크레디트들의 제1 임계 수를 충족시키고 (2) 크레디트들의 제2 수가 크레디트들의 제2 임계 수를 충족시킬 때, 계산 빌딩 블록에서 실행될 작업부하 노드를 선택하게 한다.

[0134] 예 13은 예 12의 비일시적 컴퓨터 판독가능 저장 매체를 포함하고, 제2 버퍼는 작업부하 노드와 연관된 입력 버퍼이고, 제2 수의 크레디트들은 제2 버퍼에 대응하고, 크레디트들의 제2 임계 수는 입력 버퍼에서의 데이터의 임계량에 대응한다.

[0135] 예 14는 예 8의 비일시적 컴퓨터 판독가능 저장 매체를 포함하고, 크레디트들의 임계 수는 크레디트들의 제1 임

계 수이고, 작업부하 노드는 제1 작업부하 노드이고, 명령어들은 실행될 때, 적어도 하나의 프로세서로 하여금, (1) 크레디트들의 제1 수가 크레디트들의 제1 임계 수를 충족시키고 (2) 크레디트들의 제2 수가 크레디트들의 제2 임계 수를 충족시킬 때, 계산 빌딩 블록에서 실행될 제1 작업부하 노드 및 제2 작업부하 노드를 스케줄링하게 한다.

[0136] 예 15는 장치를 포함하고, 이 장치는: 인터페이싱하기 위한 수단- 인터페이싱하기 위한 수단은 제1 수의 크레디트들을 메모리에 로딩하기 위한 것임 -, 비교하기 위한 수단- 비교하기 위한 수단은 크레디트들의 제1 수를 버퍼에서의 메모리 가용성과 연관된 크레디트들의 임계 수와 비교하기 위한 것임 -, 및 디스패치하기 위한 수단- 디스패치하기 위한 수단은 크레디트들의 제1 수가 크레디트들의 임계 수를 충족시킬 때, 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 작업부하의 작업부하 노드를 선택하기 위한 것임 -을 포함한다.

[0137] 예 16은 예 15의 장치를 포함하고, 인터페이싱하기 위한 수단은, 인터페이싱하기 위한 수단이 크레디트 매니저로부터 제1 수의 크레디트들을 수신할 때, 제1 수의 크레디트들을 메모리에 로딩하고, 작업부하 노드와 연관된 데이터의 하나 이상의 타일이 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록으로부터 버퍼에 송신될 때, 버퍼에 송신된 각각의 타일에 대한 크레디트를 크레디트 매니저에 송신하기 위한 것이다.

[0138] 예 17은 예 15의 장치를 포함하고, 버퍼는 작업부하 노드와 연관된 출력 버퍼이고, 제1 수의 크레디트들은 출력 버퍼에 대응하고, 크레디트들의 임계 수는 출력 버퍼에서 메모리의 임계량에 대응한다.

[0139] 예 18은 예 15의 장치를 포함하고, 버퍼는 작업부하 노드와 연관된 입력 버퍼이고, 제1 수의 크레디트들은 입력 버퍼에 대응하고, 크레디트들의 임계 수는 입력 버퍼에서의 데이터의 임계량에 대응한다.

[0140] 예 19는 예 15의 장치를 포함하고, 버퍼는 제1 버퍼이고, 크레디트들의 임계 수는 크레디트들의 제1 임계 수이고, 비교하기 위한 수단은 크레디트들의 제2 수를 제2 버퍼에서의 메모리 가용성과 연관된 크레디트들의 제2 임계 수와 비교하기 위한 것이고, 디스패치하기 위한 수단은 (1) 크레디트들의 제1 수가 크레디트들의 제1 임계 수를 충족시키고 (2) 크레디트들의 제2 수가 크레디트들의 제2 임계 수를 충족시킬 때, 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 작업부하 노드를 선택하기 위한 것이다.

[0141] 예 20은 예 19의 장치를 포함하고, 제2 버퍼는 작업부하 노드와 연관된 입력 버퍼이고, 제2 수의 크레디트들은 입력 버퍼에 대응하고, 크레디트들의 제2 임계 수는 입력 버퍼에서의 데이터의 임계량에 대응한다.

[0142] 예 21은 예 15의 장치를 포함하고, 크레디트들의 임계 수는 크레디트들의 제1 임계 수이고, 작업부하 노드는 제1 작업부하 노드이고, (1) 크레디트들의 제1 수가 크레디트들의 제1 임계 수를 충족시키고 (2) 크레디트들의 제2 수가 크레디트들의 제2 임계 수를 충족시킬 때, 디스패치하기 위한 수단은 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 제1 작업부하 노드 및 제2 작업부하 노드를 스케줄링하기 위한 것이다.

[0143] 예 22는 방법을 포함하고, 이 방법은: 제1 수의 크레디트들을 메모리에 로딩하는 단계, 크레디트들의 제1 수를 버퍼에서의 메모리 가용성과 연관된 크레디트들의 임계 수와 비교하는 단계, 크레디트들의 제1 수가 크레디트들의 임계 수를 충족시킬 때, 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록에서 실행될 작업부하의 작업부하 노드를 선택하는 단계를 포함한다.

[0144] 예 23은 예 22의 방법을 포함하고, 이 방법은, 제1 수의 크레디트들이 크레디트 매니저로부터 수신될 때, 제1 수의 크레디트들을 메모리에 로딩하는 단계, 및 작업부하 노드와 연관된 데이터의 하나 이상의 타일이 하나 이상의 계산 빌딩 블록 중 제1 계산 빌딩 블록으로부터 버퍼에 송신될 때, 버퍼에 송신된 각각의 타일에 대한 크레디트를 크레디트 매니저에 송신하는 단계를 추가로 포함한다.

[0145] 예 24는 예 22의 방법을 포함하고, 버퍼는 작업부하 노드와 연관된 출력 버퍼이고, 제1 수의 크레디트들은 출력 버퍼에 대응하고, 크레디트들의 임계 수는 출력 버퍼에서 메모리의 임계량에 대응한다.

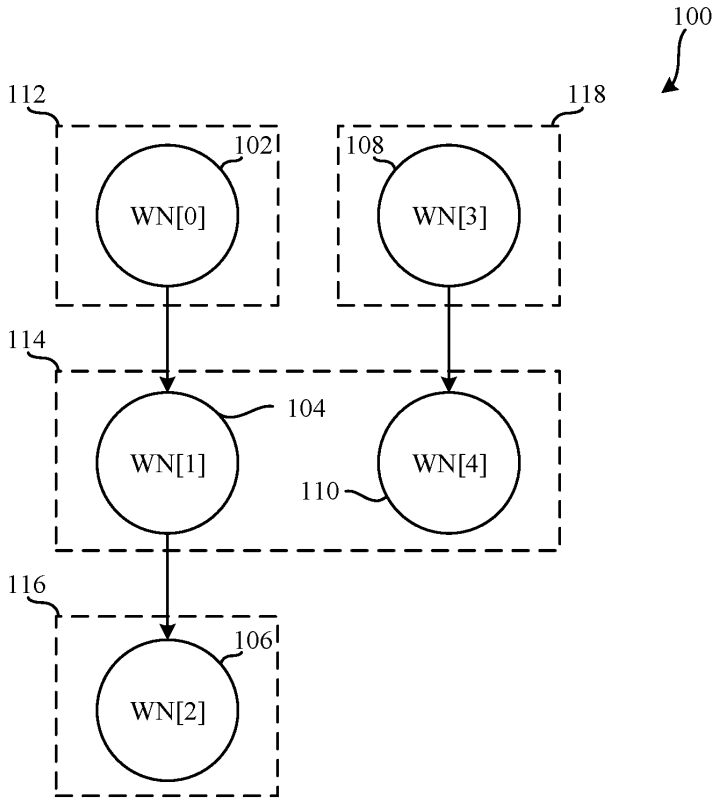
[0146] 예 25는 예 22의 방법을 포함하고, 버퍼는 작업부하 노드와 연관된 입력 버퍼이고, 제1 수의 크레디트들은 입력 버퍼에 대응하고, 크레디트들의 임계 수는 입력 버퍼에서의 데이터의 임계량에 대응한다.

[0147] 특정의 예시적인 방법들, 장치 및 제조 물품들이 본 명세서에 개시되었지만, 본 특허의 적용 범위는 이에 제한되지는 않는다. 오히려, 이 특허는 명백히 이 특허의 특허청구범위 안에 있는 모든 방법들, 장치 및 제조 물품들을 포함한다.

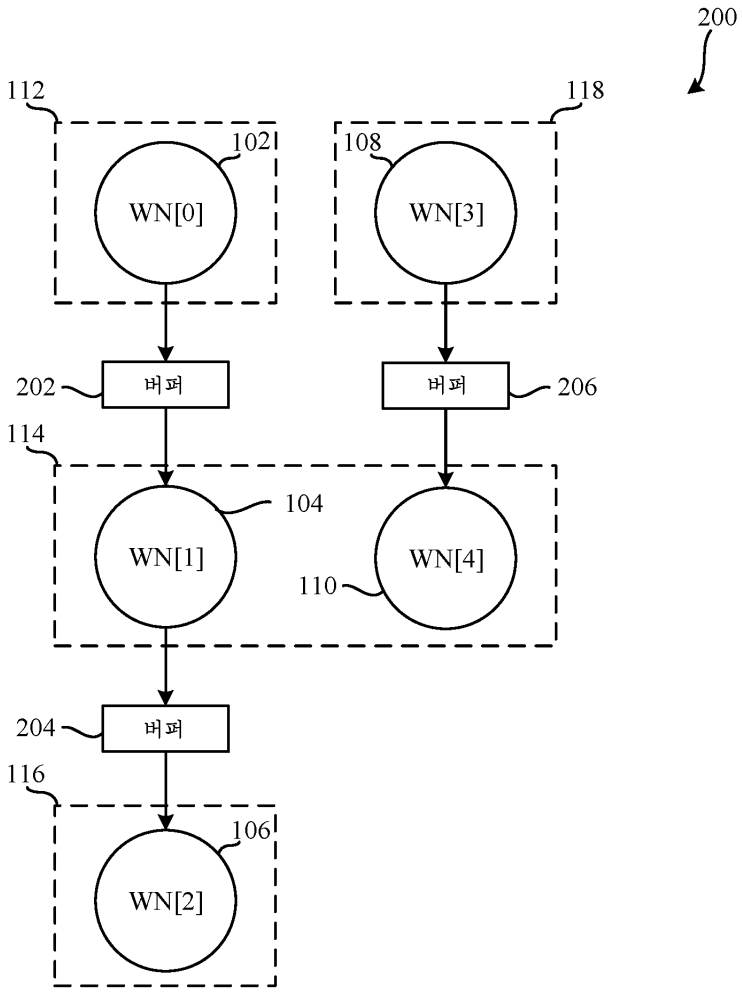
[0148] 이하의 청구항들은 이로써 이 참조에 의해 이 상세한 설명에 통합되며, 각각의 청구항은 본 개시의 별개의 실시예로서 자립한다.

도면

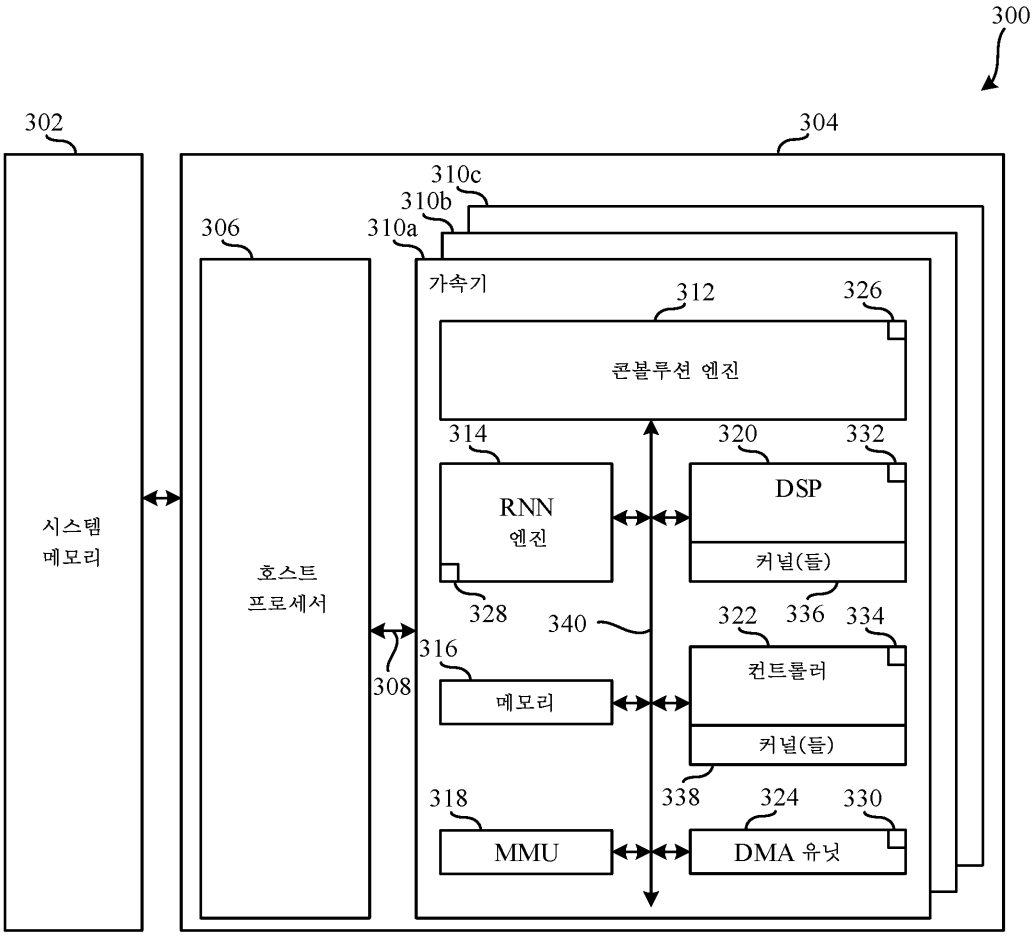
도면1



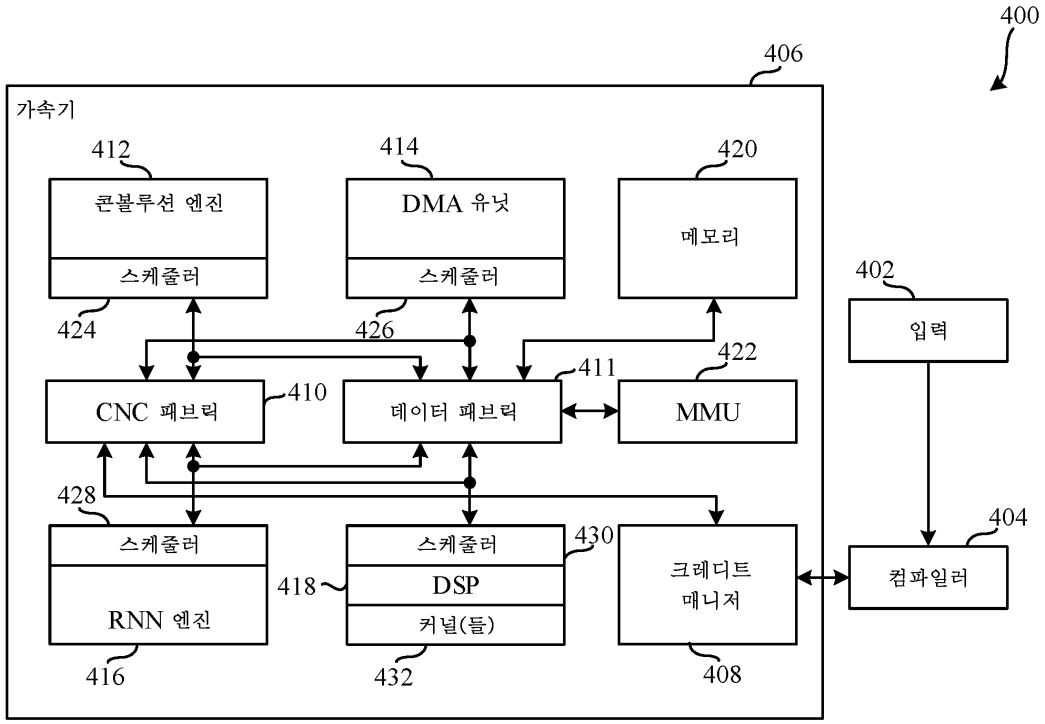
도면2


200
112
118
102
108
WN[0]
WN[3]
202
버퍼
버퍼
206
114
104
WN[1]
WN[4]
110
204
버퍼
116
106
WN[2]

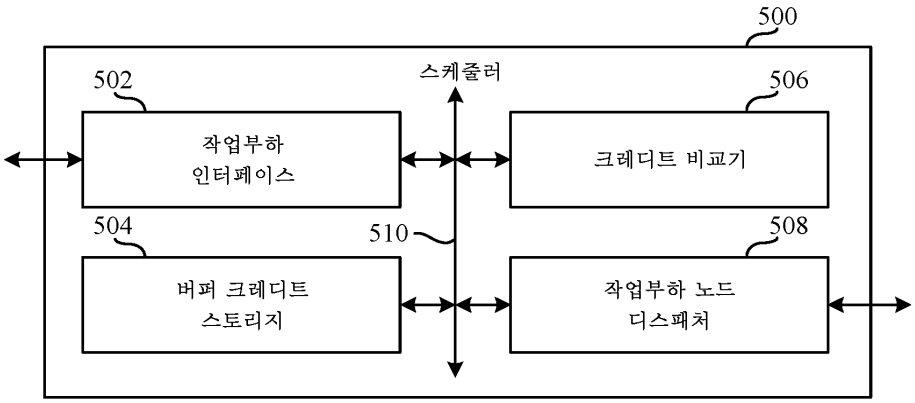
도면3


300
302
304
310c
310b
310a
306
가속기
312
326
콘볼루션 엔진
314
320
332
RNN
엔진
DSP
커널(들)
328
340
322
336
334
316
308
컨트롤러
메모리
커널(들)
338
318
324
330
MMU
DMA 유닛
시스템
메모리
호스트
프로세서

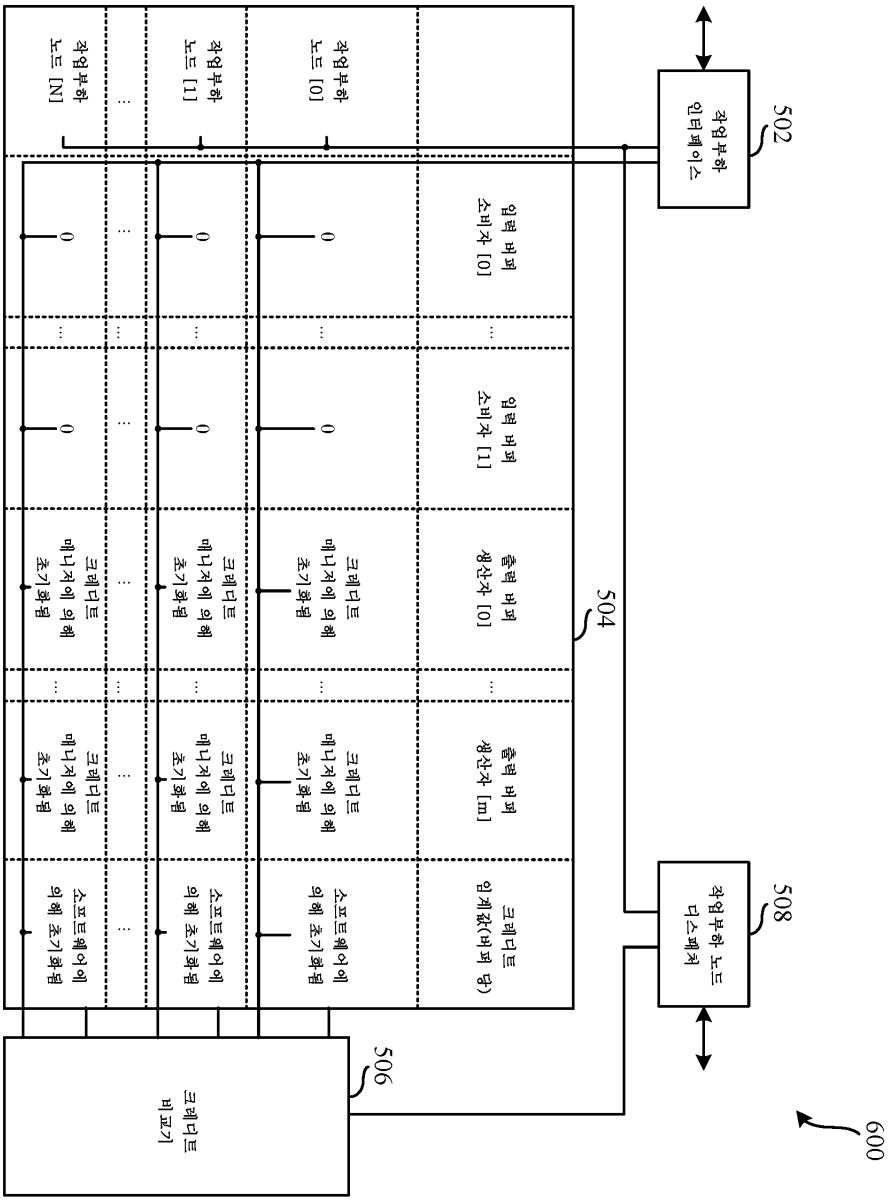
도면4


400
406
가속기
412
414
420
콘볼루션 엔진
스케줄러
DMA 유닛
스케줄러
메모리
402
입력
424
426
422
411
CNC 패브릭
410
데이터 패브릭
MMU
428
404
스케줄러
RNN 엔진
스케줄러
DSP
커널(들)
430
418
크레디트
매니저
컴파일러
416
432
408

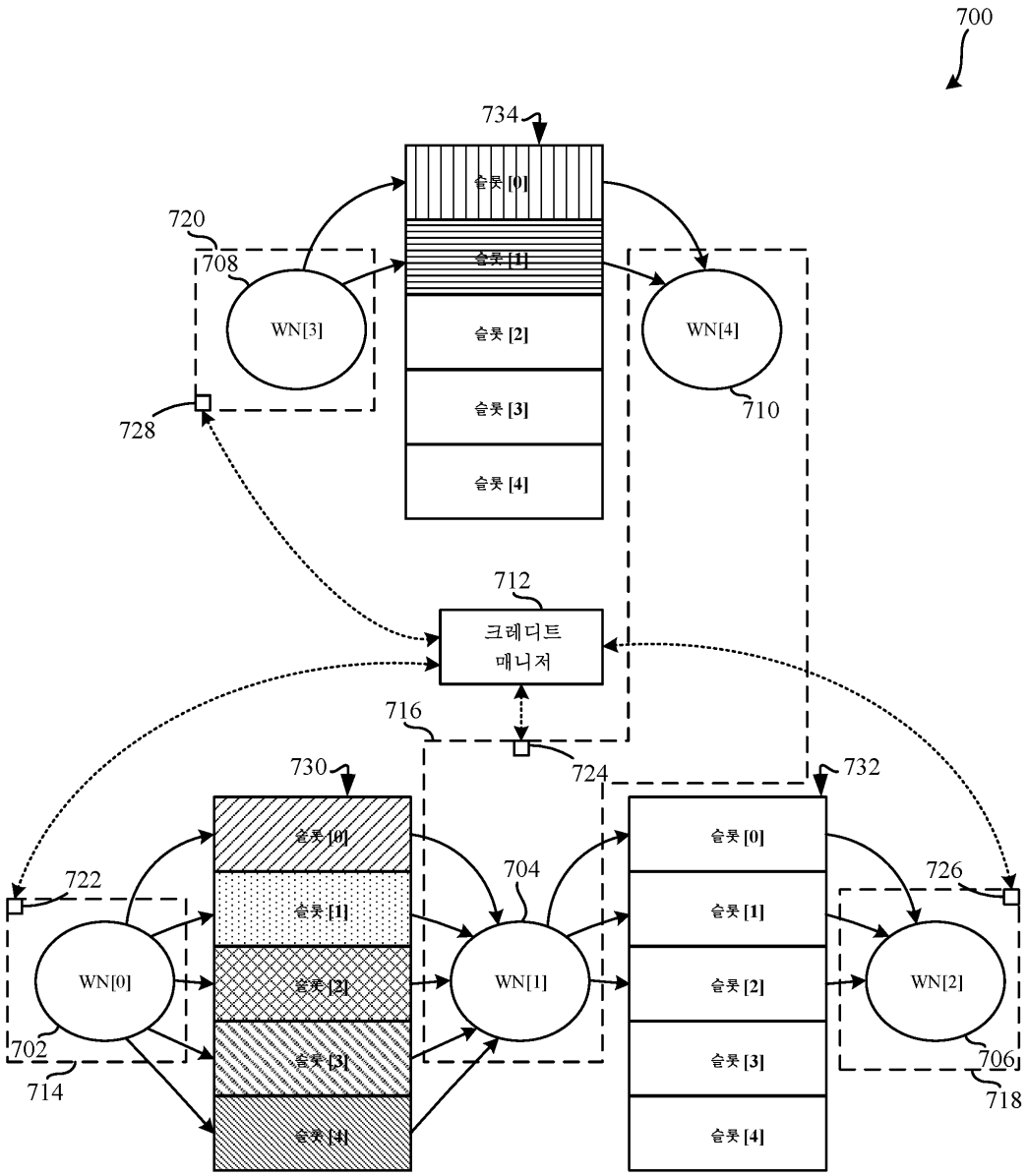
도면5


500
스케줄러
502
작업부하
인터페이스
506
크레디트 비교기
504
510
508
버퍼 크레디트
스토리지
작업부하 노드
디스패처

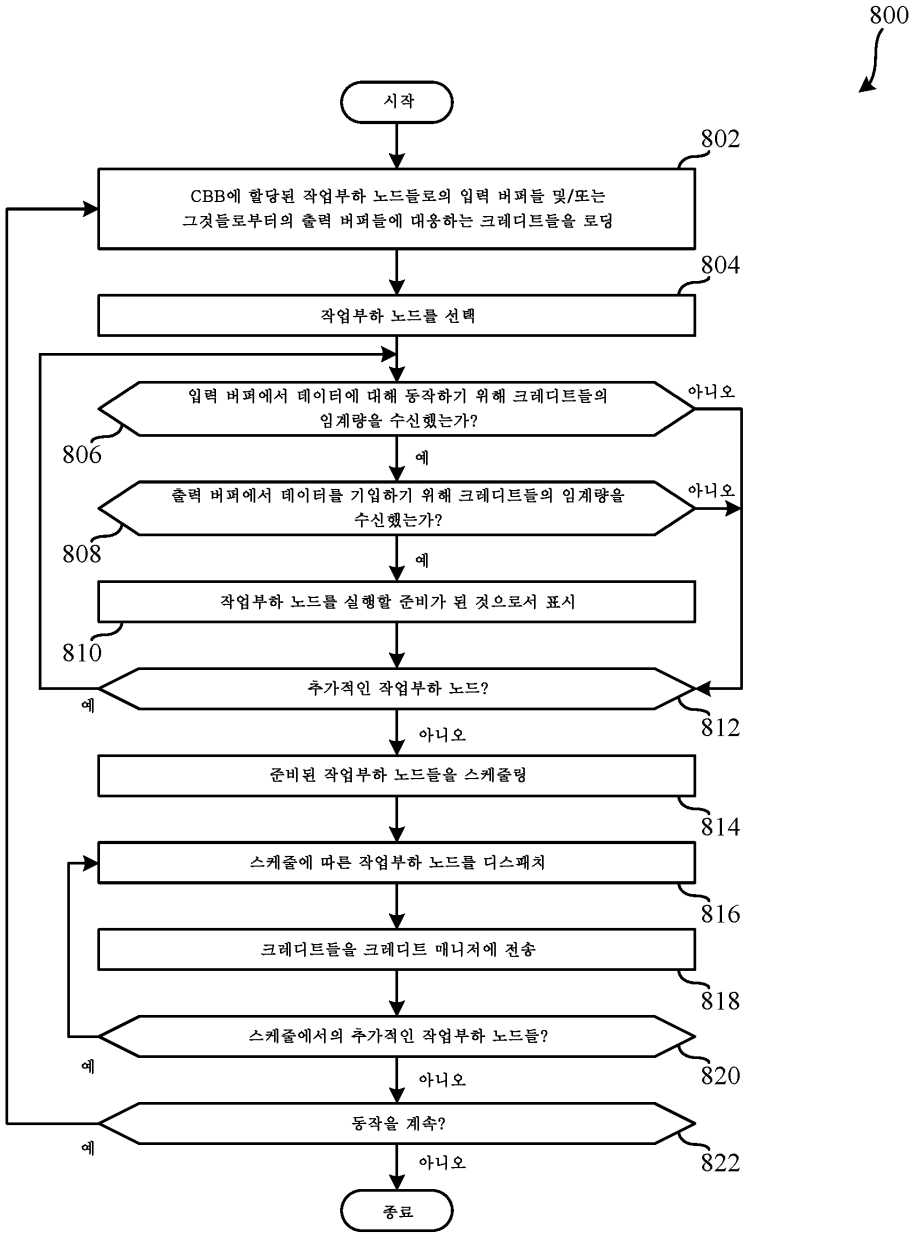
도면6


작업부하
노드 [N]
작업부하
노드 [1]
작업부하
노드 [0]
작업부하
인터페이스
502
입력 버퍼
소비자 [0]
입력 버퍼
소비자 [1]
출력 버퍼
생산자 [0]
출력 버퍼
생산자 [m]
크레디트
임계값(버퍼 당)
크레디트
메니저에 의해
초기화됨
소프트웨어에
의해 초기화됨
504
작업부하 노드
디스패처
508
크레디트
비교기
506
600

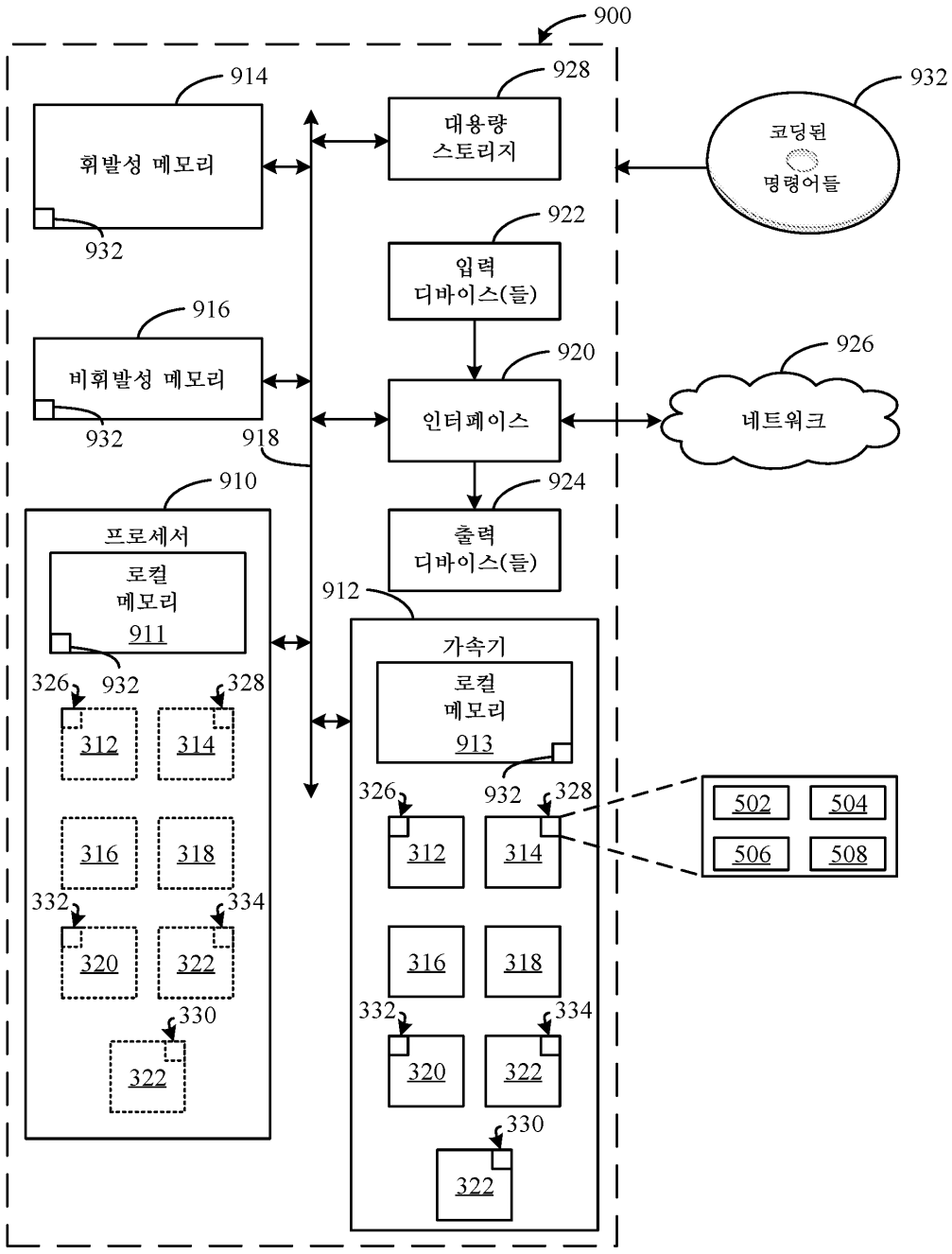
도면7


700
734
슬롯 [0]
슬롯 [1]
슬롯 [2]
슬롯 [3]
슬롯 [4]
720
708
WN[3]
WN[4]
710
728
712
크레디트
매니저
716
724
730
732
722
704
726
WN[0]
WN[1]
WN[2]
702
706
714
718

도면8


800
시작
802
CBB에 할당된 작업부하 노드들로의 입력 버퍼들 및/또는 그것들로부터의 출력 버퍼들에 대응하는 크레디트들을 로딩
804
작업부하 노드를 선택
입력 버퍼에서 데이터에 대해 동작하기 위해 크레디트들의 임계량을 수신했는가?
806
아니오
예
출력 버퍼에서 데이터를 기입하기 위해 크레디트들의 임계량을 수신했는가?
808
아니오
예
작업부하 노드를 실행할 준비가 된 것으로서 표시
810
추가적인 작업부하 노드?
812
예
아니오
준비된 작업부하 노드들을 스케줄링
814
스케줄에 따른 작업부하 노드를 디스패치
816
크레디트들을 크레디트 매니저에 전송
818
스케줄에서의 추가적인 작업부하 노드들?
820
예
아니오
동작을 계속?
822
예
아니오
종료

도면9


900
914
휘발성 메모리
932
928
대용량 스토리지
932
코딩된
명령어들
922
입력 디바이스(들)
916
비휘발성 메모리
932
920
인터페이스
926
네트워크
918
910
924
출력 디바이스(들)
프로세서
로컬 메모리
911
912
가속기
로컬 메모리
913
326
932
328
312
314
316
318
332
334
320
322
330
322
326
932
328
502
504
506
508
312
314
316
318
332
334
320
322
330
322