



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2022년09월27일  
(11) 등록번호 10-2447816  
(24) 등록일자 2022년09월22일

(51) 국제특허분류(Int. Cl.)  
G06F 15/173 (2006.01) G06F 15/78 (2006.01)  
G06F 9/30 (2018.01) G06F 9/38 (2006.01)  
G06F 9/52 (2018.01) G06F 9/54 (2018.01)

(73) 특허권자  
마이크론 테크놀로지, 인크.  
미국, 아이다호, 보이세, 사우스 페더럴 웨이  
8000

(52) CPC특허분류  
G06F 15/17325 (2013.01)  
G06F 15/7825 (2013.01)

(72) 발명자  
브루어, 토니, 엠.  
미국 75093 텍사스주 플라노 마리너스 드라이브  
5225

(21) 출원번호 10-2020-7031296

(74) 대리인  
양영준, 백만기

(22) 출원일자(국제) 2019년03월31일

심사청구일자 2022년03월31일

(85) 번역문제출일자 2020년10월29일

(65) 공개번호 10-2020-0139741

(43) 공개일자 2020년12월14일

(86) 국제출원번호 PCT/US2019/025077

(87) 국제공개번호 WO 2019/191740

국제공개일자 2019년10월03일

(30) 우선권주장

62/651,134 2018년03월31일 미국(US)

16/371,054 2019년03월31일 미국(US)

(56) 선행기술조사문헌

EP02441013 A1

(뒷면에 계속)

전체 청구항 수 : 총 14 항

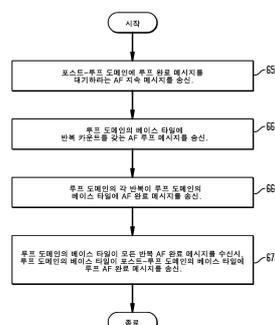
심사관 : 임재우

(54) 발명의 명칭 멀티 스레드, 자체 스케줄링 재구성 가능한 컴퓨팅 패브릭에 대한 다수의 유형의 스레드 식별자

(57) 요약

구성 가능한 컴퓨팅을 위한 대표적인 장치, 방법 및 시스템 실시 예들이 개시된다. 대표적인 시스템은 상호 접속 네트워크; 프로세서; 및 복수의 구성 가능한 회로 클러스터를 포함한다. 각 구성 가능한 회로 클러스터는 어레이로 배열되는 복수의 구성 가능한 회로; 어레이의 각 구성 가능한 회로에 연결되는 동기 네트워크; 및 어레이의 각 구성 가능한 회로에 연결되는 비동기 패킷 네트워크를 포함한다. 대표적인 구성 가능한 회로는 구성 가능한 계산 회로 및 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 마스터 동기 입력, 현재 데이터 경로 구성 명령, 및 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 갖는 구성 메모리를 포함한다.

대표도 - 도21



(52) CPC특허분류

*G06F 15/7867* (2013.01)

*G06F 9/3009* (2013.01)

*G06F 9/30134* (2013.01)

*G06F 9/3855* (2013.01)

*G06F 9/52* (2013.01)

*G06F 9/546* (2013.01)

(56) 선행기술조사문헌

KR1020110044465 A

KR1020130093571 A

KR1020150067722 A

US07635987 B1

US20180089140 A1

## 명세서

### 청구범위

#### 청구항 1

구성가능한 회로로서,

구성가능한 계산 회로;

상기 구성가능한 계산 회로에 결합된 제1 메모리 회로;

상기 구성가능한 계산 회로에 결합된 복수의 동기 네트워크 입력;

상기 구성가능한 계산 회로에 결합된 복수의 동기 네트워크 출력;

상기 구성가능한 계산 회로, 상기 복수의 동기 네트워크 입력, 및 상기 복수의 동기 네트워크 출력에 연결된 제2 구성 메모리 회로; 및

상기 구성가능한 계산 회로에 결합된 제어 회로를 포함하고, 상기 제어 회로는,

메모리 제어 회로;

스레드 제어 회로; 및

스레드 식별자의 각 유형이 루프 및 중첩 루프 실행에 대한 루프 레벨을 나타내는 복수의 유형의 스레드 식별자를 갖는 완료 테이블을 저장하는 복수의 제어 레지스터를 포함하고, 상기 복수의 제어 레지스터는 선택된 루프에 전용인 변수들에 대한 각 유형의 스레드 식별자 액세스를 허용하기 위한 스레드 식별자 스택의 상단을 더 저장하는, 구성가능한 회로.

#### 청구항 2

제1항에 있어서, 상기 제2 구성 메모리 회로는,

상기 구성 가능한 계산 회로의 내부 데이터 경로를 구성하기 위한 제1 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및

상기 제1 명령 메모리로부터 상기 제1 복수의 데이터 경로 구성 명령의 현재 데이터 경로 구성 명령의 선택을 위한, 다른 상이한 구성 회로로부터 상기 현재 데이터 경로 구성 명령 또는 명령 인덱스의 수신을 위한 상기 복수의 동기 네트워크 입력의 마스터 동기 네트워크 입력의 선택을 위한, 그리고 다음 구성 가능한 계산 회로의 상기 마스터 동기 네트워크 입력으로의 출력을 위한 다음 데이터 경로 구성 명령 또는 명령 인덱스의 선택을 위한 제2 복수의 데이터 경로 구성 명령 또는 명령 인덱스를 저장하는 제2 명령 메모리를 포함하는, 구성 가능한 회로.

#### 청구항 3

제1항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에 활성 루프 스레드 수의 루프 카운트를 더 저장하고, 스레드 식별자 풀에 스레드 식별자를 반환하는 비동기 패브릭 메시지의 수신에 응답하여, 상기 제어 회로는 상기 루프 카운트를 감소시키고, 상기 루프 카운트가 제로에 도달할 때, 비동기 패브릭 완료 메시지를 송신하는, 구성 가능한 회로.

#### 청구항 4

제1항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에, 상기 복수의 유형의 스레드 식별자들의 각각의 스레드 식별자에 대한 데이터 완료 카운트를 더 저장하고, 현재 스레드에 대한 각각의 스레드 식별자에 대해, 순서화된 스레드 실행을 제공하기 위해 상기 현재 스레드의 실행에 후속하는 실행을 위한 다음 스레드를 식별하는 다른 스레드 식별자를 더 저장하는, 구성가능한 회로.

#### 청구항 5

복수의 구성가능한 회로 중 구성가능한 회로로서,

구성가능한 계산 회로;

상기 구성가능한 계산 회로에 결합된 복수의 동기 네트워크 입력;

상기 구성가능한 계산 회로에 결합된 복수의 동기 네트워크 출력- 상기 복수의 동기 네트워크 출력은 동기 네트워크를 형성하기 위해 상기 복수의 구성가능한 회로의 인접한 구성가능한 회로들의 상기 복수의 동기 네트워크 입력에 직접 결합됨 -;

상기 구성가능한 계산 회로에 결합된 제어 회로 - 상기 제어 회로는,

스레드 제어 회로; 및

복수의 유형의 스레드 식별자들을 갖는 완료 테이블을 저장하는 복수의 제어 레지스터를 포함하고, 각각의 유형의 스레드 식별자는 루프 및 중첩 루프 실행에 대한 루프 레벨을 나타내고, 상기 복수의 제어 레지스터는 선택된 루프에 전용인 변수들에 대한 각각의 유형의 스레드 식별자 액세스를 허용하기 위한 스레드 식별자 스택의 상단을 더 저장함 -; 및

상기 구성가능한 계산 회로, 상기 제어 회로, 상기 복수의 동기 네트워크 입력, 및 상기 복수의 동기 네트워크 출력에 결합된 구성 메모리를 포함하고, 상기 구성 메모리는,

상기 구성가능한 계산 회로의 데이터 경로를 구성하기 위한 제1 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및

상기 복수의 동기 네트워크 입력 중 마스터 동기 네트워크 입력의 선택을 위한 제2 복수의 데이터 경로 구성 명령 또는 명령 인덱스를 저장하는 제2 명령 메모리를 포함하는, 구성가능한 회로.

**청구항 6**

제5항에 있어서, 상기 제2 복수의 데이터 경로 구성 명령 또는 명령 인덱스는 상기 구성가능한 계산 회로를 위한 상기 제1 명령 메모리로부터의 상기 제1 복수의 데이터 경로 구성 명령 중 현재 데이터 경로 구성 명령의 선택을 위한 그리고 다음 구성가능한 계산 회로의 상기 마스터 동기 네트워크 입력에 출력하기 위한 다음 구성가능한 계산 회로를 위한 다음 데이터 경로 구성 명령 또는 명령 인덱스의 선택을 위한 명령들 또는 명령 인덱스들을 더 포함하는, 구성가능한 회로.

**청구항 7**

제5항에 있어서, 상기 복수의 구성가능한 회로 중 하나 이상은 다른 구성가능한 회로들의 각각의 동기 네트워크 입력들에 결합된 각각의 동기 네트워크 출력들과 직렬로 결합되어 제1 베이스 타일을 갖는 제1 동기 도메인을 형성하고, 상기 복수의 구성가능한 회로 중 하나 이상은 다른 구성가능한 회로들의 각각의 동기 네트워크 입력들에 결합된 각각의 동기 네트워크 출력들과 직렬로 결합되어 제2 베이스 타일을 갖는 제2 동기 도메인을 형성하고, 상기 복수의 구성가능한 회로는 상기 제1 및 제2 동기 도메인들을 사용하여 복수의 계산을 수행하도록 적응되고, 상기 복수의 구성가능한 회로는 비동기 패킷 네트워크를 통해 복수의 제어 메시지를 생성 및 송신하도록 더 적응되고, 상기 복수의 제어 메시지는 상기 제2 동기 도메인으로부터의 하나 이상의 완료 메시지 및 상기 제1 동기 도메인으로부터 상기 제2 동기 도메인의 상기 제2 베이스 타일로의 하나 이상의 계속 메시지를 포함하는, 구성가능한 회로.

**청구항 8**

제5항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에 활성 루프 스레드 수의 루프 카운트를 더 저장하고, 스레드 식별자를 스레드 식별자 풀로 반환하는 비동기 패브릭 메시지의 수신에 응답하여, 상기 제어 회로는 상기 루프 카운트를 감소시키고, 상기 루프 카운트가 0에 도달할 때 비동기 패브릭 완료 메시지를 송신하는, 구성가능한 회로.

**청구항 9**

제5항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에, 상기 복수의 유형의 스레드 식별자들 중 각각의 스레드 식별자에 대한 대응하는 데이터 완료 카운트를 더 저장하고, 현재 스레드에 대한 각각의 스레드 식

별자에 대해, 순서화된 스레드 실행을 제공하기 위해 상기 현재 스레드의 실행에 후속하는 실행을 위한 다음 스레드를 식별하는 다른 스레드 식별자를 더 저장하고, 상기 스레드 제어 회로는 실행을 위한 선택된 스레드에 대한 상기 스레드 식별자에 대해, 상기 대응하는 데이터 완료 카운트가 0으로 감소되었고 상기 선택된 스레드에 대한 상기 스레드 식별자가 상기 다음 스레드로서 식별될 때, 상기 선택된 스레드를 큐잉하도록 적응되는, 구성 가능한 회로.

**청구항 10**

제6항에 있어서,

상기 구성가능한 계산 회로에 결합된 조건부 논리 회로를 더 포함하고, 상기 조건부 논리 회로는 상기 구성가능한 계산 회로로부터의 출력에 따라, 상기 다음 구성가능한 계산 회로의 상기 마스터 동기 네트워크 입력에 상이한 다음 데이터 경로 구성 명령 또는 명령 인덱스를 출력하기 위해 상기 다음 데이터 경로 구성 명령 또는 명령 인덱스를 수정함으로써 조건부 브랜칭을 제공하도록 적응되는, 구성가능한 회로.

**청구항 11**

제5항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에, 현재의 스레드에 대한 각각의 스레드 식별자에 대하여, 순서화된 스레드 실행을 제공하기 위해 상기 현재의 스레드의 실행에 후속하는 실행을 위한 다음 스레드를 식별하는 다른 스레드 식별자를 더 저장하는, 구성가능한 회로.

**청구항 12**

제5항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에 상기 복수의 유형의 스레드 식별자들 중 각각의 스레드 식별자에 대한 대응하는 데이터 완료 카운트를 더 저장하고, 상기 스레드 제어 회로는, 상기 복수의 상이한 스레드 중 각각의 스레드의 상기 스레드 식별자에 대해, 임의의 데이터 종속성들의 완료를 나타내기 위해 상기 대응하는 데이터 완료 카운트가 0으로 감소했을 때, 미리 결정된 케이던스 시에 실행할 복수의 상이한 스레드를 자체 스케줄링 및 큐잉하도록 적응되는, 구성가능한 회로.

**청구항 13**

제5항에 있어서, 상기 스레드 제어 회로는, 스레드의 스레드 식별자에 대해, 대응하는 데이터 완료 카운트가 0으로 감소되었고, 상기 스레드의 스레드 식별자가 다음 스레드로서 식별될 때, 미리 결정된 케이던스 시에 실행을 위해 상기 스레드를 자체 스케줄링 및 큐잉하도록 적응되는, 구성가능한 회로.

**청구항 14**

제5항에 있어서, 상기 스레드 제어 회로는,

실행할 준비가 된 하나 이상의 스레드에 대한 하나 이상의 스레드 식별자들을 저장하는 실행 큐;

실행을 허용하는 데이터 완료 카운트들을 갖는 스레드들에 대한 하나 이상의 스레드 식별자를 저장하거나, 할당된 스레드 식별자가 없는 스레드들을 저장하는 지속 큐; 및

재진입 큐를 포함하고, 상기 재진입 큐는 실행을 허용하는 데이터 완료 카운트들을 갖는 스레드들에 대한 하나 이상의 스레드 식별자 저장하여, 상기 지속 큐의 임의의 계산 스레드들에 앞서 상기 재진입 큐의 스레드들을 상기 실행 큐로 이동시킴으로써 지정된 카운트 시에 상기 재진입 큐의 상기 스레드들의 실행을 지속하는, 구성가능한 회로.

**청구항 15**

삭제

**청구항 16**

삭제

**청구항 17**

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

**발명의 설명**

**기술 분야**

[0001] **관련 출원 상호 참조**

[0002] 본 출원은 "Multiple Types of Thread Identifiers for a Multi-Threaded, Self-Scheduling Reconfigurable Computing Fabric(멀티 스레드, 자체 스케줄링 재구성 가능한 컴퓨팅 패브릭에 대한 다수의 유형의 스레드 식별자)"이라는 명칭으로 2018년 3월 31일자로 출원되어 공동 양도된 미국 가 특허 출원 번호 62/651,134(발명자 Tony M. Brewer)의 정규 출원으로서 이의 이익 및 우선권을 주장하고, 이들 모두는 이에 의해 여기에 그 전체가 제시된 바와 동일한 완전한 효력으로 그 전체가 참고로 여기에 통합된다(이하 "관련 출원들"로 지칭됨).

[0003] **기술분야**

[0004] 본 발명은 개괄적으로 구성 가능한 컴퓨팅 회로, 보다 구체적으로는 상호 접속 네트워크, 동적 재구성, 에너지 또는 전력 소비에 대한 동적 제어가 임베드된 구성 가능한 컴퓨팅 회로를 포함하는 이기종 컴퓨팅 시스템에 관한 것이다.

**배경 기술**

[0005] 기존의 많은 컴퓨팅 시스템은 계산 속도, 에너지(또는 전력) 소비 및 관련 열 방출 측면들에서 계산 처리 기능들에 대해 상당한 한계에 도달했다. 예를 들어, 인공 지능 및 기타 중요한 컴퓨팅 애플리케이션들을 수용하는 것과 같은 고급 컴퓨팅 기술들에 대한 요구가 증가함에 따라, 기존 컴퓨팅 솔루션들은 점점 더 부적절해졌다.

[0006] 따라서, 예를 들어 그리고 제한없이, 스펙트럼 기술, 기계 학습, 5G 네트워킹 알고리즘 및 대용량 스텐실 코드를 사용하는 그래프 클러스터링과 같은 합성 개구 레이더, 5G 기지국, 및 그래프 분석 애플리케이션들과 같은 감지, 통신 및 분석 애플리케이션들에 사용되는 고속 푸리에 변환(FFT, Fast Fourier Transform) 및 유한 임펄스 응답(FIR, Finite Impulse Response) 필터들의 계산과 같은 계산 집약적 커널들을 위한 고성능 및 에너지 효율적인 솔루션들을 제공할 수 있는 컴퓨팅 아키텍처에 대한 지속적인 요구가 있다.

[0007] 또한 이러한 다양한 애플리케이션의 어느 하나에 대해 구성될 수 있지만 가장 중요한 것은 동적 자체 구성 및 자체 재구성이 가능한 구성 가능한 컴퓨팅 아키텍처에 대한 지속적인 요구가 있다.

**발명의 내용**

[0008] 아래에서 더 상세히 설명될 바와 같이, 대표적인 장치, 시스템 및 방법은 예를 들어 그리고 제한없이, 스펙트럼 기술, 기계 학습, 5G 네트워킹 알고리즘 및 대용량 스텐실 코드를 사용하는 그래프 클러스터링과 같은 합성 개구 레이더, 5G 기지국, 및 그래프 분석 애플리케이션들과 같은 감지, 통신 및 분석 애플리케이션들에 사용되는 고속 푸리에 변환(FFT, Fast Fourier Transform) 및 유한 임펄스 응답(FIR, Finite Impulse Response) 필터들의 계산과 같은 계산 집약적 커널들을 위한 고성능 및 에너지 효율적인 솔루션들을 제공할 수 있는 컴퓨팅 아키텍처를 제공한다.

[0009] 중요하게는, 다양한 대표적인 실시 예는 이러한 다양한 애플리케이션의 어느 하나에 대해 구성될 수 있지만 가장 중요한 것은 자체 스케줄링, 동적 자체 구성 및 자체 재구성, 조건부 브랜칭, 비동기 시그널링을 위한 역압 제어, 순서화된 스레드 실행 및 루프 스레드 실행(데이터 종속성 포함), 데이터 종속성 및/또는 순서화 완료시 자동으로 스레드 실행 시작, 전용 변수들에 대한 루프 액세스 제공, 재진입 큐 사용, 및 중첩 루프들을 포함하여 고급 루프 실행을 위한 다양한 스레드 식별자 사용을 통한 루프 스레드들의 신속한 실행 제공이 가능한 멀티

스레드, 낮은 레벨로 세분화되는 구성 가능한 컴퓨팅 아키텍처를 제공한다.

- [0010] 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 구성 메모리를 포함할 수 있으며, 상기 구성 메모리는: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 상기 동기 네트워크 입력들의 마스터 동기 입력의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함한다.
- [0011] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 및 상기 구성 가능한 계산 회로, 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 구성 메모리를 포함할 수 있으며, 상기 구성 메모리는: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 상기 구성 가능한 계산 회로에 대한 현재 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함한다.
- [0012] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 및 상기 구성 가능한 계산 회로, 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 구성 메모리를 포함할 수 있으며, 상기 구성 메모리는: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함한다.
- [0013] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제어 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로를 포함할 수 있으며, 상기 구성 메모리 회로는: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 상기 동기 네트워크 입력들의 마스터 동기 입력의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함한다.
- [0014] 또 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로; 및 상기 구성 가능한 계산 회로에 연결되는 제어 회로로서, 메모리 제어 회로; 스레드 제어 회로; 및 복수의 제어 레지스터를 포함하는, 상기 제어 회로를 포함할 수 있다.
- [0015] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로, 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 구성 메모리로서: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함하는, 상기 구성 메모리; 및 상기 구성 가능한 컴퓨팅 회로에 연결되는 조건부 논리 회로로서, 상기 구성 가능한 컴퓨팅 회로로부터의 출력에 따라, 상기 복수의 동기 네트워크 출력 중 선택된 출력상에 제공되는 상기 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스를 변경함으로써 조건부 브랜칭을 제공하도록 구성되는, 상기 조건부 논리 회로를 포함할 수 있다.
- [0016] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제어 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 비동기 패킷 네트워크에 그리고 상기 제1 메모리 회로에 연결되는 비동기 네트워크 입력 큐; 비동기 네트워크 출력 큐; 및 상기 비동기 네트워크 출력 큐에 연결되는 흐름 제어 회로로서, 상기 비동기 네트워크 출력 큐에서 미리 결

정된 임계치가 도달될 때 중지 신호를 생성하도록 구성되는, 상기 흐름 제어 회로를 포함할 수 있다.

[0017] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로; 및 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 포함할 수 있으며, 상기 제어 회로는: 메모리 제어 회로; 스레드 제어 회로; 및 복수의 제어 레지스터로서, 스레드를 순서대로 실행하기 위해 복수의 스레드 식별자 및, 각 스레드 식별자에 대해, 현재 스레드를 실행한 다음 실행할 다음 스레드 식별자를 갖는 루프 테이블을 저장하는, 상기 복수의 제어 레지스터를 포함한다.

[0018] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로; 및 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 포함할 수 있으며, 상기 제어 회로는: 메모리 제어 회로; 복수의 제어 레지스터로서, 제1 데이터 완료 카운트를 갖는 완료 테이블을 저장하는, 상기 복수의 제어 레지스터; 및 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되었을 때 그것을 큐잉하도록 구성되는 스레드 제어 회로를 포함한다.

[0019] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력 및 출력; 비동기 네트워크 입력 큐 및 비동기 네트워크 출력 큐; 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로를 포함할 수 있으며, 상기 제2 구성 메모리는: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 상기 동기 네트워크 입력들의 마스터 동기 입력의 선택을 위한, 상기 구성 가능한 계산 회로에 대한 현재 데이터 경로 구성 명령의 선택을 위한, 그리고 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함하고; 상기 구성 가능한 회로는 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 더 포함하며, 상기 제어 회로는: 메모리 제어 회로; 복수의 제어 레지스터로서, 제1 데이터 완료 카운트를 갖는 완료 테이블을 저장하는, 상기 복수의 제어 레지스터; 및 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되었을 때 그것을 큐잉하도록 구성되는 스레드 제어 회로를 포함한다.

[0020] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로; 및 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 포함할 수 있으며, 상기 제어 회로는: 메모리 제어 회로; 복수의 제어 레지스터로서, 제1 데이터 완료 카운트를 갖는 완료 테이블을 저장하는, 상기 복수의 제어 레지스터; 및 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되고 그것의 스레드 식별자가 상기 다음 스레드일 때 그것을 큐잉하도록 구성되는 스레드 제어 회로를 포함한다.

[0021] 또 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로를 포함할 수 있고; 상기 구성 가능한 회로는 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 더 포함하며, 상기 제어 회로는: 메모리 제어 회로; 스레드 제어 회로; 및 스레드 식별자의 각 유형이 루프 및 중첩 루프 실행에 대한 루프 레벨을 나타내는 복수의 유형의 스레드 식별자를 갖는 완료 테이블을 저장하는 복수의 제어 레지스터로서, 선택된 루프에 전용인 변수들에 대한 각 유형의 스레드 식별자 액세스를 허용하기 위한 스레드 식별자 스택의 상단을 더 저장하는, 상기 복수의 제어 레지스터를 포함한다.

[0022] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결

되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로; 및 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 포함할 수 있으며, 상기 제어 회로는: 메모리 제어 회로; 복수의 제어 레지스터; 및 스레드 제어 회로로서: 실행을 허용하는 완료 카운트들은 갖지만 아직 할당된 스레드 식별자는 갖지 않는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장하는 지속 큐; 및 실행을 허용하는 완료 카운트들을 갖고 지정된 스포크 카운트시 재진입 큐의 상기 스레드들의 실행을 준비하도록 할당된 스레드 식별자를 갖는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장하는 상기 재진입 큐를 포함하는, 상기 스레드 제어 회로를 포함한다.

[0023] 다른 대표적인 실시 예에서, 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 입력; 상기 구성 가능한 계산 회로에 연결되는 복수의 동기 네트워크 출력; 및 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로; 및 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 포함할 수 있으며, 상기 제어 회로는: 메모리 제어 회로; 스레드 식별자 풀 및 활성 루프 스레드 수의 루프 카운트를 갖는 완료 테이블을 저장하는 복수의 제어 레지스터; 및 스레드 제어 회로를 포함하며, 상기 스레드 식별자 풀에 스레드 식별자를 반환하는 비동기 패브리크 메시지의 수신에 응답하여, 상기 제어 회로는 상기 루프 카운트를 감소시키고, 상기 루프 카운트가 제로에 도달할 때, 비동기 패브리크 완료 메시지를 송신한다.

[0024] 대표적인 실시 예에서, 시스템이 개시되며, 이는: 비동기 패킷 네트워크; 동기 네트워크; 및 어레이로 배열되는 복수의 구성 가능한 회로를 포함할 수 있으며, 상기 복수의 구성 가능한 회로의 각 구성 가능한 회로는 상기 동기 네트워크 및 상기 비동기 패킷 네트워크 양자에 연결되고, 상기 복수의 구성 가능한 회로는 상기 동기 네트워크를 사용하여 복수의 동기 도메인을 형성하여 복수의 계산을 수행하도록 구성되고, 상기 복수의 구성 가능한 회로는 복수의 제어 메시지를 생성하고 상기 비동기 패킷 네트워크를 통해 송신하도록 구성되며, 상기 복수의 제어 메시지는 하나 이상의 완료 메시지 및 계속 메시지를 포함한다.

[0025] 다른 대표적인 실시 예에서, 시스템은: 어레이로 배열되는 복수의 구성 가능한 회로; 상기 어레이의 상기 복수의 구성 가능한 회로의 각 구성 가능한 회로에 연결되는 동기 네트워크; 및 상기 어레이의 상기 복수의 구성 가능한 회로의 각 구성 가능한 회로에 연결되는 비동기 패킷 네트워크를 포함할 수 있다.

[0026] 다른 대표적인 실시 예에서, 시스템은: 상호 접속 네트워크; 상기 상호 접속 네트워크에 연결되는 프로세서; 및 상기 상호 접속 네트워크에 연결되는 복수의 구성 가능한 회로 클러스터를 포함할 수 있다.

[0027] 대표적인 실시 예에서, 시스템은: 상호 접속 네트워크; 상기 상호 접속 네트워크에 연결되는 프로세서; 상기 상호 접속 네트워크에 연결되는 호스트 인터페이스; 및 상기 상호 접속 네트워크에 연결되는 복수의 구성 가능한 회로 클러스터를 포함할 수 있으며, 상기 복수의 구성 가능한 회로 클러스터의 각 구성 가능한 회로 클러스터는: 어레이로 배열되는 복수의 구성 가능한 회로; 상기 어레이의 상기 복수의 구성 가능한 회로의 각 구성 가능한 회로에 연결되는 동기 네트워크; 상기 어레이의 상기 복수의 구성 가능한 회로의 각 구성 가능한 회로에 연결되는 비동기 패킷 네트워크; 상기 비동기 패킷 네트워크 및 상기 상호 접속 네트워크에 연결되는 메모리 인터페이스; 및 상기 비동기 패킷 네트워크 및 상기 상호 접속 네트워크에 연결되는 디스패치 인터페이스를 포함한다.

[0028] 다른 대표적인 실시 예에서, 시스템은: Folded Clos 구성 및 종점들과의 인터페이스들에 복수의 직접, 메쉬 연결을 갖는 제1 복수의 크로스바 스위치를 포함하는 계층 상호 접속 네트워크; 상기 상호 접속 네트워크에 연결되는 프로세서; 상기 상호 접속 네트워크에 연결되는 호스트 인터페이스; 및 상기 상호 접속 네트워크에 연결되는 복수의 구성 가능한 회로 클러스터를 포함할 수 있으며, 상기 복수의 구성 가능한 회로 클러스터의 각 구성 가능한 회로 클러스터는: 어레이로 배열되는 복수의 구성 가능한 회로; 상기 어레이의 상기 복수의 구성 가능한 회로의 각 구성 가능한 회로에 연결되고 상기 어레이의 인접한 구성 가능한 회로들 간에 복수의 직접 연결을 제공하는 동기 네트워크; 제2 복수의 크로스바 스위치를 포함하는 비동기 패킷 네트워크로서, 각 크로스바 스위치가 상기 어레이의 상기 복수의 구성 가능한 회로의 적어도 하나의 구성 가능한 회로 및 상기 제2 복수의 크로스바 스위치의 다른 크로스바 스위치에 연결되는, 상기 비동기 패킷 네트워크; 상기 비동기 패킷 네트워크 및 상기 상호 접속 네트워크에 연결되는 메모리 인터페이스; 및 상기 비동기 패킷 네트워크 및 상기 상호 접속 네트워크에 연결되는 디스패치 인터페이스를 포함한다.

- [0029] 다른 대표적인 실시 예에서, 시스템은: 상호 접속 네트워크; 상기 상호 접속 네트워크에 연결되는 프로세서; 상기 상호 접속 네트워크에 연결되는 호스트 인터페이스; 및 상기 상호 접속 네트워크에 연결되는 복수의 구성 가능한 회로 클러스터를 포함할 수 있으며, 상기 복수의 구성 가능한 회로 클러스터의 각 구성 가능한 회로 클러스터는: 동기 네트워크; 비동기 패킷 네트워크; 상기 비동기 패킷 네트워크 및 상기 상호 접속 네트워크에 연결되는 메모리 인터페이스; 상기 비동기 패킷 네트워크 및 상기 상호 접속 네트워크에 연결되는 디스패치 인터페이스; 및 어레이로 배열되는 복수의 구성 가능한 회로를 포함하고, 각 구성 가능한 회로는: 구성 가능한 계산 회로; 상기 구성 가능한 계산 회로에 연결되는 제어 회로를 포함하며, 상기 제어 회로는: 메모리 제어 회로; 스레드 제어 회로; 및 복수의 제어 레지스터; 상기 구성 가능한 계산 회로에 연결되는 제1 메모리 회로; 상기 구성 가능한 계산 회로 및 상기 동기 네트워크에 연결되는 복수의 동기 네트워크 입력 및 출력; 상기 비동기 패킷 네트워크에 연결되는 비동기 네트워크 입력 큐 및 비동기 네트워크 출력 큐; 상기 구성 가능한 계산 회로, 상기 제어 회로, 상기 동기 네트워크 입력들 및 상기 동기 네트워크 출력들에 연결되는 제2 구성 메모리 회로를 포함하며, 상기 구성 메모리 회로는: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 상기 동기 네트워크 입력들의 마스터 동기 입력의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함한다.
- [0030] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제2 명령 및 명령 인덱스 메모리는 상기 구성 가능한 계산 회로에 대한 현재 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 더 저장할 수 있다.
- [0031] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제2 명령 및 명령 인덱스 메모리는 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 더 저장할 수 있다.
- [0032] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 동기 네트워크 출력의 하나의 동기 네트워크 출력의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 더 저장할 수 있다.
- [0033] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 회로 또는 시스템은; 상기 제1 명령 메모리 및 상기 제2 명령 및 명령 인덱스 메모리에 연결되는 구성 메모리 멀티플렉서를 더 포함할 수 있다.
- [0034] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 메모리 멀티플렉서의 선택 입력은 제1 설정을 가지며, 상기 현재 데이터 경로 구성 명령은 상기 제2 명령 및 명령 인덱스 메모리로부터의 명령 인덱스를 사용하여 선택될 수 있다.
- [0035] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 메모리 멀티플렉서의 상기 선택 입력은 상기 제1 설정과 상이한 제2 설정을 가지며, 상기 현재 데이터 경로 구성 명령은 상기 마스터 동기 입력으로부터의 명령 인덱스를 사용하여 선택될 수 있다.
- [0036] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제2 명령 및 명령 인덱스 메모리는 상기 현재 데이터 경로 명령과 독립적으로 상기 구성 회로의 부분들의 구성을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 더 저장할 수 있다.
- [0037] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스 중 선택된 스포크 명령 및 데이터 경로 구성 명령 인덱스는 모듈로 스포크 카운트에 따라 선택될 수 있다.
- [0038] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 회로 또는 시스템은: 상기 구성 가능한 계산 회로에 연결되는 조건부 논리 회로를 포함할 수 있다.
- [0039] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 컴퓨팅 회로로부터의 출력에 따라, 상기 조건부 논리 회로는 상기 복수의 동기 네트워크 출력 중 선택된 출력상에 제공되는 상기 다음 데이터 경로 명령 인덱스를 변경하도록 구성될 수 있다.
- [0040] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 컴퓨팅 회로로부터의 출력에 따라, 상기 조건부 논리 회로는 상기 복수의 동기 네트워크 출력 중 선택된 출력상에 제공되는 상기 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스를 변경함으로써 조건부 브랜칭을 제공하도록 구성될 수 있다.
- [0041] 다양한 대표적인 실시 예의 어느 하나에서, 상기 조건부 논리 회로는 인에이블될 때, 상기 다음 데이터 경로 명령 또는 데이터 경로 명령 인덱스를 지정하기 위해 상기 구성 가능한 컴퓨팅 회로로부터의 상기 출력과 상기 다

음 데이터 경로 명령의 최하위 비트를 OR 연산함으로써 조건부 브랜칭을 제공하도록 구성될 수 있다.

- [0042] 다양한 대표적인 실시 예의 어느 하나에서, 상기 조건부 논리 회로는 인에이블될 때, 상기 다음 데이터 경로 명령 인덱스를 지정하기 위해 상기 구성 가능한 컴퓨팅 회로로부터의 상기 출력과 상기 다음 데이터 경로 명령 인덱스의 최하위 비트를 OR 연산함으로써 조건부 브랜칭을 제공하도록 구성될 수 있다.
- [0043] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 동기 네트워크 입력은: 동기 네트워크의 복수의 통신 라인에 연결되는 복수의 입력 레지스터; 및 상기 복수의 입력 레지스터에 그리고 상기 마스터 동기 입력의 선택을 위해 상기 제2 명령 및 명령 인덱스 메모리에 연결되는 입력 멀티플렉서를 포함할 수 있다.
- [0044] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 동기 네트워크 출력은: 상기 동기 네트워크의 복수의 통신 라인에 연결되는 복수의 출력 레지스터; 및 상기 구성 가능한 컴퓨팅 회로로부터의 출력의 선택을 위해 상기 구성 가능한 컴퓨팅 회로에 연결되는 출력 멀티플렉서를 포함할 수 있다.
- [0045] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 회로 또는 시스템은: 비동기 패킷 네트워크에 그리고 상기 메모리 회로에 연결되는 비동기 네트워크 입력 큐; 및 상기 출력 멀티플렉서에 연결되는 비동기 네트워크 출력 큐를 더 포함할 수 있다.
- [0046] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 회로 또는 시스템은: 상기 비동기 네트워크 입력 큐에 그리고 상기 비동기 네트워크 출력 큐에 연결되는 비동기 패브릭 상태 기계를 더 포함할 수 있으며, 상기 비동기 패브릭 상태 기계는 상기 비동기 패킷 네트워크로부터 수신되는 입력 데이터 패킷을 디코딩하도록 그리고 상기 비동기 패킷 네트워크상의 송신을 위해 출력 데이터 패킷을 어셈블하도록 구성된다.
- [0047] 다양한 대표적인 실시 예의 어느 하나에서, 상기 비동기 패킷 네트워크는 복수의 크로스바 스위치를 포함할 수 있으며, 각 크로스바 스위치는 복수의 구성 가능한 회로에 그리고 적어도 하나의 다른 크로스바 스위치에 연결된다.
- [0048] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 회로 또는 시스템은: 복수의 구성 가능한 회로의 어레이를 더 포함할 수 있으며, 각 구성 가능한 회로는 상기 복수의 동기 네트워크 입력 및 상기 복수의 동기 네트워크 출력을 통해 상기 동기 네트워크에 연결되고; 각 구성 가능한 회로는 상기 비동기 네트워크 입력 및 상기 비동기 네트워크 출력을 통해 상기 비동기 패킷 네트워크에 연결된다.
- [0049] 다양한 대표적인 실시 예의 어느 하나에서, 상기 동기 네트워크는 상기 복수의 구성 가능한 회로의 상기 어레이의 인접한 구성 가능한 회로들을 연결하는 복수의 직접 점대점 연결을 포함할 수 있다.
- [0050] 다양한 대표적인 실시 예의 어느 하나에서, 각 구성 가능한 회로는: 상기 복수의 입력 레지스터와 상기 복수의 출력 레지스터 간 직접 관통 연결을 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 직접 관통 연결은 상기 동기 네트워크상에 수신되는 제2 구성 가능한 회로로부터 상기 동기 네트워크상에 송신되는 제3 구성 가능한 회로로의 데이터 송신을 위한 직접 점대점 연결을 제공할 수 있다.
- [0051] 다양한 대표적인 실시 예의 어느 하나에서, 구성 가능한 계산 회로는 부호 있는 덧셈 및 부호 없는 덧셈, 절대값, 부정, 논리 NOT, 덧셈 및 부정, 뺄셈  $A - B$ , 역뺄셈  $B - A$ , 부호 있는 초과 및 부호 없는 초과, 부호 있는 이상 및 부호 없는 이상, 부호 있는 미만 및 부호 없는 미만, 부호 있는 이하 및 부호 없는 이하, 같거나 같지 않음의 비교, 논리 AND 연산, 논리 OR 연산, 논리 XOR 연산, 논리 NAND 연산, 논리 NOR 연산, 논리 NOT XOR 연산, 논리 AND NOT 연산, 논리 OR NOT 연산 및 정수와 부동 소수점 간의 상호 변환으로 이루어진 군으로부터 선택되는 적어도 하나의 정수 연산을 수행하도록 구성된 산술, 논리 및 비트 연산 회로를 포함할 수 있다.
- [0052] 다양한 대표적인 실시 예의 어느 하나에서, 구성 가능한 계산 회로는 부호 있는 덧셈 및 부호 없는 덧셈, 절대값, 부정, 논리 NOT, 덧셈 및 부정, 뺄셈  $A - B$ , 역뺄셈  $B - A$ , 부호 있는 초과 및 부호 없는 초과, 부호 있는 이상 및 부호 없는 이상, 부호 있는 미만 및 부호 없는 미만, 부호 있는 이하 및 부호 없는 이하, 같거나 같지 않음의 비교, 논리 AND 연산, 논리 OR 연산, 논리 XOR 연산, 논리 NAND 연산, 논리 NOR 연산, 논리 NOT XOR 연산, 논리 AND NOT 연산, 논리 OR NOT 연산, 정수와 부동 소수점 간의 상호 변환 및 이들의 조합들로 이루어진 군으로부터 선택되는 적어도 하나의 부동 소수점 연산을 수행하도록 구성된 산술, 논리 및 비트 연산 회로를 포함할 수 있다.
- [0053] 다양한 대표적인 실시 예의 어느 하나에서, 구성 가능한 계산 회로는 곱셈, 이동, 입력 전달, 부호 있는 곱셈 및 부호 없는 곱셈, 부호 있는 우방 이동 및 부호 없는 우방 이동, 부호 있는 좌방 이동 및 부호 없는 좌방 이동, 비트 순서 역전, 치환, 정수와 부동 소수점 간의 상호 변환 및 이들의 조합으로 이루어진 군으로부터 선택

되는 적어도 하나의 정수 연산을 수행하도록 구성된 곱셈 및 이동 연산 회로를 포함할 수 있다.

- [0054] 다양한 대표적인 실시 예의 어느 하나에서, 구성 가능한 계산 회로는 곱셈, 이동, 입력 전달, 부호 있는 곱셈 및 부호 없는 곱셈, 부호 있는 우방 이동 및 부호 없는 우방 이동, 부호 있는 좌방 이동 및 부호 없는 좌방 이동, 비트 순서 역전, 치환, 정수와 부동 소수점 간의 상호 변환 및 이들의 조합으로 이루어진 군으로부터 선택되는 적어도 부동 소수점 연산을 수행하도록 구성된 곱셈 및 이동 연산 회로를 포함할 수 있다.
- [0055] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 구성 가능한 회로의 상기 어레이는 제1 상호 접속 네트워크에 더 연결될 수 있다.
- [0056] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 구성 가능한 회로의 상기 어레이는: 제3, 시스템 메모리 인터페이스 회로; 및 디스패치 인터페이스 회로를 더 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 디스패치 인터페이스 회로는 상기 제1 상호 접속 네트워크를 통해 작업 기술자 패킷을 수신하고, 상기 작업 기술자 패킷에 응답하여, 상기 복수의 구성 가능한 회로를 선택된 계산의 실행을 위해 구성하도록 상기 복수의 구성 가능한 회로에 대한 하나 이상의 데이터 및 제어 패킷을 생성하도록 구성될 수 있다.
- [0057] 다양한 대표적인 실시 예의 어느 하나에서, 상기 구성 가능한 회로 또는 시스템은: 상기 비동기 네트워크 출력 큐에 연결되는 흐름 제어 회로로서, 상기 비동기 네트워크 출력 큐에서 미리 결정된 임계치가 도달될 때 중지 신호를 생성하도록 구성되는, 상기 흐름 제어 회로를 더 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 중지 신호에 응답하여, 각 비동기 네트워크 출력 큐는 상기 비동기 패킷 네트워크상에 데이터 패킷들을 출력하는 것을 중지한다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 중지 신호에 응답하여, 각 구성 가능한 계산 회로는 그것의 현재 명령의 완료시 실행을 중지한다.
- [0058] 다양한 대표적인 실시 예의 어느 하나에서, 복수의 구성 가능한 회로의 상기 어레이의 제1 복수의 구성 가능한 회로는 상기 동기 네트워크를 통해 제1 미리 결정된 시퀀스로 제1 동기 도메인을 형성하도록 연결될 수 있고; 복수의 구성 가능한 회로의 상기 어레이의 제2 복수의 구성 가능한 회로는 상기 동기 네트워크를 통해 제2 미리 결정된 시퀀스로 제2 동기 도메인을 형성하도록 연결된다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 제1 동기 도메인은 상기 비동기 패킷 네트워크를 통해 상기 제2 동기 도메인으로 송신되는 지속 메시지를 생성하도록 구성될 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 제2 동기 도메인은 상기 비동기 패킷 네트워크를 통해 상기 제1 동기 도메인으로 송신되는 완료 메시지를 생성하도록 구성될 수 있다.
- [0059] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 제어 레지스터는 제1 데이터 완료 카운트를 갖는 완료 테이블을 저장할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 제어 레지스터는 제2 반복 카운트를 갖는 상기 완료 테이블을 더 저장한다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 제어 레지스터는 복수의 스레드 식별자 및, 각 스레드 식별자에 대해, 현재 스레드를 실행한 다음 실행할 다음 스레드 식별자를 갖는 루프 테이블을 더 저장할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 제어 레지스터는 상기 루프 테이블에, 처음 반복의 식별 및 마지막 반복의 식별을 더 저장할 수 있다.
- [0060] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되고 그것의 스레드 식별자가 상기 다음 스레드일 때 그것을 큐잉하도록 구성될 수 있다.
- [0061] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 임의의 데이터 종속성의 완료를 나타낼 때 그것을 큐잉하도록 구성될 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 완료 카운트는 복수의 스레드의 선택된 스레드마다, 상기 선택된 스레드의 실행 이전에, 수신될 완료 메시지들의 미리 결정된 수를 나타낼 수 있다.
- [0062] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 제어 레지스터는 스레드 식별자의 각 유형이 루프 및 중첩 루프 실행에 대한 루프 레벨을 나타내는 복수의 유형의 스레드 식별자들을 갖는 완료 테이블을 더 저장할 수 있다.
- [0063] 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 제어 레지스터는 활성 루프 스레드 수의 루프 카운트를 갖는 완료 테이블을 더 저장할 수 있고, 스레드 식별자 풀에 스레드 식별자를 반환하는 비동기 패브릭 메시지의 수신에 응답하여, 상기 제어 회로는 상기 루프 카운트를 감소시키고, 상기 루프 카운트가 제로에 도달할 때, 비동기 패브릭 완료 메시지를 송신한다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 복수의 제어 레지스터는 선택된 루프에 전용된 변수들에 대한 각 유형의 스레드 식별자 액세스를 허용하기 위한 스레드 식별자 스택의

상단을 더 저장할 수 있다.

- [0064] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는: 지속 큐; 및 재진입 큐를 더 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 지속 큐는 실행을 허용하는 완료 카운트들을 갖지만 아직 할당된 스레드 식별자는 갖지 않는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장한다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 재진입 큐는 실행을 허용하는 완료 카운트들을 갖고 할당된 스레드 식별자를 갖는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 재진입 큐 내 스레드 식별자를 갖는 임의의 스레드가 상기 지속 큐 내 스레드 식별자를 갖는 임의의 스레드의 실행 이전에 실행될 수 있다.
- [0065] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는: 우선 큐 내 스레드 식별자를 갖는 임의의 스레드가 상기 지속 큐 또는 상기 재진입 큐 내 스레드 식별자를 갖는 임의의 스레드의 실행 이전에 실행될 수 있는 상기 우선 큐를 더 포함할 수 있다.
- [0066] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는: 실행 큐 내 스레드 식별자를 갖는 임의의 스레드가 상기 스레드 식별자에 대한 스포크 카운트의 발생시 실행될 수 있는 상기 실행 큐를 더 포함할 수 있다.
- [0067] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제2 구성 메모리 회로는: 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 저장하는 제1 명령 메모리; 및 상기 동기 네트워크 입력들의 마스터 동기 입력의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 저장하는 제2 명령 및 명령 인덱스 메모리를 포함할 수 있다.
- [0068] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는 실행할 계산 스레드를 자체 스케줄링하도록 구성될 수 있다.
- [0069] 다양한 대표적인 실시 예의 어느 하나에서, 상기 조건부 논리 회로는 다음 구성 가능한 회로에 의해 실행할 상이한 제2 다음 명령으로 브랜칭하도록 구성될 수 있다.
- [0070] 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는 실행할 계산 스레드들을 순서화하도록 구성될 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는 실행할 루프 계산 스레드들을 순서화하도록 구성될 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 제어 회로는 데이터 종속성으로부터의 하나 이상의 완료 신호에 응답하여 계산 스레드들의 실행을 개시하도록 구성될 수 있다.
- [0071] 구성 가능한 회로를 구성하는 다양한 방법 실시 예가 또한 개시된다. 대표적인 방법 실시 예는: 제1 명령 메모리를 사용하여, 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 제공하는 단계; 및 제2 명령 및 명령 인덱스 메모리를 사용하여, 복수의 동기 네트워크 입력의 마스터 동기 입력의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계를 포함할 수 있다.
- [0072] 다양한 대표적인 실시 예의 어느 하나에서, 구성 가능한 회로를 구성하는 방법은: 제1 명령 메모리를 사용하여, 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 제공하는 단계; 및 제2 명령 및 명령 인덱스 메모리를 사용하여, 상기 구성 가능한 계산 회로에 대한 현재 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계를 포함할 수 있다.
- [0073] 다양한 대표적인 실시 예의 어느 하나에서, 구성 가능한 회로를 구성하는 방법은: 제1 명령 메모리를 사용하여, 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 복수의 데이터 경로 구성 명령을 제공하는 단계; 및 제2 명령 및 명령 인덱스 메모리를 사용하여, 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계를 포함할 수 있다.
- [0074] 또한 멀티 스레드 구성 가능한 회로의 스레드 실행을 제어하는 방법이 개시되며, 상기 구성 가능한 회로는 구성 가능한 계산 회로를 갖는다. 대표적인 방법 실시 예는: 조건부 논리 회로를 사용하여, 상기 구성 가능한 컴퓨팅 회로로부터의 출력에 따라, 다음 구성 가능한 회로에 제공되는 상기 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스를 변경함으로써 조건부 브랜칭을 제공하는 단계를 포함할 수 있다.
- [0075] 멀티 스레드 구성 가능한 회로의 스레드 실행을 제어하는 다른 대표적인 방법 실시 예는: 흐름 제어 회로를 사용하여, 비동기 네트워크 출력 큐에서 미리 결정된 임계치가 도달될 때 중지 신호를 생성하는 단계를 포함할 수 있다.
- [0076] 멀티 스레드 구성 가능한 회로의 스레드 실행을 제어하는 다른 대표적인 방법 실시 예는: 복수의 제어 레지스터

를 사용하여, 스레드를 순서대로 실행하기 위해 복수의 스레드 식별자 및, 각 스레드 식별자에 대해, 현재 스레드를 실행한 다음 실행할 다음 스레드 식별자를 갖는 루프 테이블을 저장하는 단계를 포함할 수 있다.

- [0077] 멀티 스레드 구성 가능한 회로의 스레드 실행을 제어하는 다른 대표적인 방법 실시 예는: 복수의 제어 레지스터를 사용하여, 제1 데이터 완료 카운트를 갖는 완료 테이블을 저장하는 단계; 및 스레드 제어 회로를 사용하여, 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되었을 때 그것을 큐잉하는 단계를 포함할 수 있다.
- [0078] 구성 가능한 계산 회로를 갖는 멀티 스레드 구성 가능한 회로의 스레드 실행을 구성 및 제어하는 방법이 개시되며, 대표적인 방법 실시 예는: 제1 명령 메모리를 사용하여, 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 구성 명령들을 사용하여 복수의 데이터 경로를 제공하는 단계; 제2 명령 및 명령 인덱스 메모리를 사용하여, 복수의 동기 네트워크 입력의 마스터 동기 입력의 선택을 위한, 상기 구성 가능한 계산 회로에 대한 현재 데이터 경로 구성 명령의 선택을 위한, 그리고 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계; 복수의 제어 레지스터를 사용하여, 제1 데이터 완료 카운트를 갖는 완료 테이블을 제공하는 단계; 및 스레드 제어 회로를 사용하여, 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되었을 때 그것을 큐잉하는 단계를 포함할 수 있다.
- [0079] 멀티 스레드 구성 가능한 회로의 스레드 실행을 구성 및 제어하는 다른 방법은: 제1 명령 메모리를 사용하여, 상기 구성 가능한 계산 회로의 데이터 경로를 구성하기 위한 구성 명령들을 사용하여 복수의 데이터 경로를 제공하는 단계; 제2 명령 및 명령 인덱스 메모리를 사용하여, 복수의 동기 네트워크 입력의 마스터 동기 입력의 선택을 위한, 상기 구성 가능한 계산 회로에 대한 현재 데이터 경로 구성 명령의 선택을 위한, 그리고 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계; 복수의 제어 레지스터를 사용하여, 제1 데이터 완료 카운트를 갖는 완료 테이블을 제공하는 단계; 및 스레드 제어 회로를 사용하여, 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되고 그것의 스레드 식별자가 상기 다음 스레드일 때 그것을 큐잉하는 단계를 포함할 수 있다.
- [0080] 멀티 스레드 구성 가능한 회로의 스레드 실행을 제어하는 다른 방법은: 복수의 제어 레지스터를 사용하여, 스레드 식별자의 각 유형이 루프 및 중첩 루프 실행에 대한 루프 레벨을 나타내는 복수의 유형의 스레드 식별자를 갖는 완료 테이블을 저장하는 단계를 포함할 수 있고, 상기 복수의 제어 레지스터는 스레드 식별자 스택의 상단을 더 저장하고; 선택된 루프에 전용된 변수들에 대한 각 유형의 스레드 식별자 액세스를 허용하는 단계를 포함할 수 있다.
- [0081] 멀티 스레드 구성 가능한 회로의 스레드 실행을 제어하는 다른 방법은: 복수의 제어 레지스터를 사용하여, 데이터 완료 카운트를 갖는 완료 테이블을 저장하는 단계; 스레드 제어 회로를 사용하여, 실행을 허용하는 완료 카운트들은 갖지만 아직 할당된 스레드 식별자는 갖지 않는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장하는 지속 큐를 제공하는 단계; 및 스레드 제어 회로를 사용하여, 실행을 허용하는 완료 카운트들을 갖고 지정된 스포크 카운트시 재진입 큐의 상기 스레드들의 실행을 준비하도록 할당된 스레드 식별자를 갖는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장하는 상기 재진입 큐를 제공하는 단계를 포함할 수 있다.
- [0082] 멀티 스레드 구성 가능한 회로의 스레드 실행을 제어하는 다른 방법은: 복수의 제어 레지스터를 사용하여, 스레드 식별자 풀 및 활성 루프 스레드 수의 루프 카운트를 갖는 완료 테이블을 저장하는 단계; 및 스레드 제어 회로를 사용하여, 상기 스레드 식별자 풀에 스레드 식별자를 반환하는 비동기 패브릭 메시지의 수신에 응답하여, 상기 루프 카운트를 감소시키는 단계, 및 상기 루프 카운트가 제로에 도달할 때, 비동기 패브릭 완료 메시지를 송신하는 단계를 포함할 수 있다.
- [0083] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제2 명령 및 명령 인덱스 메모리를 사용하여, 상기 구성 가능한 계산 회로에 대한 현재 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계를 더 포함할 수 있다.
- [0084] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제2 명령 및 명령 인덱스 메모리를 사용하여, 다음 구성 가능한 계산 회로에 대한 다음 데이터 경로 구성 명령의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계를 더 포함할 수 있다.
- [0085] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제2 명령 및 명령 인덱스 메모리를 사용하여, 상

기 복수의 동기 네트워크 입력의 동기 네트워크 출력의 선택을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계를 더 포함할 수 있다.

- [0086] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 구성 메모리 멀티플렉서를 사용하여, 상기 제2 명령 및 명령 인덱스 메모리로부터의 명령 인덱스를 사용하여 상기 현재 데이터 경로 구성 명령을 선택하기 위한 제1 선택 설정을 제공하는 단계를 더 포함할 수 있다.
- [0087] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 구성 메모리 멀티플렉서를 사용하여, 마스터 동기 입력으로부터의 명령 인덱스를 사용하여 상기 현재 데이터 경로 구성 명령을 선택하기 위한, 상기 제1 설정과 상이한 제2 선택 설정을 제공하는 단계를 더 포함할 수 있다.
- [0088] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제2 명령 및 명령 인덱스 메모리를 사용하여, 상기 현재 데이터 경로 명령과 독립적으로 상기 구성 회로의 부분들의 구성을 위한 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스를 제공하는 단계를 더 포함할 수 있다.
- [0089] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 구성 메모리 멀티플렉서를 사용하여, 모듈로 스포크 카운트에 따라 상기 복수의 스포크 명령 및 데이터 경로 구성 명령 인덱스 중 스포크 명령 및 데이터 경로 구성 명령 인덱스는 선택하는 단계를 더 포함할 수 있다.
- [0090] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 조건부 논리 회로를 사용하여 그리고 상기 구성 가능한 컴퓨팅 회로로부터의 출력에 따라, 상기 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스를 변경하는 단계를 더 포함할 수 있다.
- [0091] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 조건부 논리 회로를 사용하여 그리고 상기 구성 가능한 컴퓨팅 회로로부터의 출력에 따라, 상기 다음 데이터 경로 명령 또는 다음 데이터 경로 명령 인덱스를 변경함으로써 조건부 브랜칭을 제공하는 단계를 더 포함할 수 있다.
- [0092] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 조건부 논리 회로를 인에이블하는 단계; 및 상기 조건부 논리 회로를 사용하여 그리고 상기 구성 가능한 컴퓨팅 회로로부터의 출력에 따라, 상기 다음 데이터 경로 명령 또는 데이터 경로 명령 인덱스를 지정하기 위해 상기 구성 가능한 컴퓨팅 회로로부터의 상기 출력과 상기 다음 데이터 경로 명령의 최하위 비트를 OR 연산함으로써 조건부 브랜칭을 제공하는 단계를 더 포함할 수 있다.
- [0093] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 입력 멀티플렉서를 사용하여, 상기 마스터 동기 입력을 선택하는 단계를 더 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 출력 멀티플렉서를 사용하여, 상기 구성 가능한 컴퓨팅 회로로부터 출력을 선택하는 단계를 더 포함할 수 있다.
- [0094] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 비동기 네트워크 입력 큐에 그리고 비동기 네트워크 출력 큐에 연결되는 비동기 패브릭 상태 기계를 사용하여, 상기 비동기 패킷 네트워크로부터 수신되는 입력 데이터 패킷을 디코딩하는 단계 및 상기 비동기 패킷 네트워크상의 송신을 위해 출력 데이터 패킷을 어셈블하는 단계를 더 포함할 수 있다.
- [0095] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 동기 네트워크를 사용하여, 상기 복수의 구성 가능한 회로의 상기 어레이의 인접한 구성 가능한 회로들을 연결하는 복수의 직접 점대점 연결을 제공하는 단계를 더 포함할 수 있다.
- [0096] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 구성 가능한 회로를 사용하여, 복수의 입력 레지스터와 복수의 출력 레지스터 간 직접 관통 연결을 제공하는 단계를 더 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 직접 관통 연결은 상기 동기 네트워크상에 수신되는 제2 구성 가능한 회로로부터 상기 동기 네트워크상에 송신되는 제3 구성 가능한 회로의 데이터 송신을 위한 직접 점대점 연결을 제공한다.
- [0097] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 구성 가능한 계산 회로를 사용하여, 부호 있는 덧셈 및 부호 없는 덧셈, 절대 값, 부정, 논리 NOT, 덧셈 및 부정, 뺄셈  $A - B$ , 역뺄셈  $B - A$ , 부호 있는 초과 및 부호 없는 초과, 부호 있는 이상 및 부호 없는 이상, 부호 있는 미만 및 부호 없는 미만, 부호 있는 이하 및 부호 없는 이하, 같거나 같지 않음의 비교, 논리 AND 연산, 논리 OR 연산, 논리 XOR 연산, 논리 NAND 연산, 논리 NOR 연산, 논리 NOT XOR 연산, 논리 AND NOT 연산, 논리 OR NOT 연산 및 정수와 부동 소수점 간의 상호 변환으로 이루어진 군으로부터 선택되는 적어도 하나의 정수 또는 부동 소수점 연산을 수행하는 단계를 더 포함할 수 있다.

- [0098] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 구성 가능한 계산 회로를 사용하여, 곱셈, 이동, 입력 전달, 부호 있는 곱셈 및 부호 없는 곱셈, 부호 있는 우방 이동 및 부호 없는 우방 이동, 부호 있는 좌방 이동 및 부호 없는 좌방 이동, 비트 순서 역전, 치환, 정수와 부동 소수점 간의 상호 변환 및 이들의 조합으로 이루어진 군으로부터 선택되는 적어도 하나의 정수 또는 부동 소수점 연산을 수행하는 단계를 더 포함할 수 있다.
- [0099] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 디스패치 인터페이스 회로를 사용하여, 상기 제1 상호 접속 네트워크를 통해 작업 기술자 패킷을 수신하고, 상기 작업 기술자 패킷에 응답하여, 상기 복수의 구성 가능한 회로를 선택된 계산의 실행을 위해 구성하도록 상기 복수의 구성 가능한 회로에 대한 하나 이상의 데이터 및 제어 패킷을 생성하는 단계를 더 포함할 수 있다.
- [0100] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 흐름 제어 회로를 사용하여, 상기 비동기 네트워크 출력 큐에서 미리 결정된 임계치가 도달될 때 중지 신호를 생성하는 단계를 더 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 중지 신호에 응답하여, 각 비동기 네트워크 출력 큐는 상기 비동기 패킷 네트워크상에 데이터 패킷들을 출력하는 것을 중지한다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 중지 신호에 응답하여, 각 구성 가능한 계산 회로는 그것의 현재 명령의 완료시 실행을 중지한다.
- [0101] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 복수의 구성 가능한 회로의 상기 어레이의 제1 복수의 구성 가능한 회로를 상기 동기 네트워크를 통해 제1 미리 결정된 시퀀스로 제1 동기 도메인을 형성하도록 연결하는 단계; 및 복수의 구성 가능한 회로의 상기 어레이의 제2 복수의 구성 가능한 회로를 상기 동기 네트워크를 통해 제2 미리 결정된 시퀀스로 제2 동기 도메인을 형성하도록 연결하는 단계를 더 포함할 수 있다.
- [0102] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제1 동기 도메인으로부터 상기 제2 동기 도메인으로 상기 비동기 패킷 네트워크를 통해 송신할 지속 메시지를 생성하는 단계를 더 포함할 수 있다.
- [0103] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제2 동기 도메인으로부터 상기 제1 동기 도메인으로 상기 비동기 패킷 네트워크를 통해 송신할 완료 메시지를 생성하는 단계를 더 포함할 수 있다.
- [0104] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은 상기 복수의 제어 레지스터에 제1 데이터 완료 카운트를 갖는 완료 테이블을 저장하는 단계를 더 포함할 수 있다.
- [0105] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 복수의 제어 레지스터에 제2 반복 카운트를 갖는 상기 완료 테이블을 저장하는 단계를 더 포함할 수 있다.
- [0106] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 복수의 제어 레지스터에 복수의 스레드 식별자를 갖는 루프 테이블을 저장하는 단계, 및 각 스레드 식별자에 대해, 현재 스레드를 실행한 다음 실행할 다음 스레드 식별자를 저장하는 단계를 더 포함할 수 있다.
- [0107] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 복수의 제어 레지스터의 상기 루프 테이블에, 처음 반복의 식별 및 마지막 반복의 식별을 저장하는 단계를 더 포함할 수 있다.
- [0108] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제어 회로를 사용하여, 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되었을 때 그것을 큐잉하는 단계를 더 포함할 수 있다.
- [0109] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제어 회로를 사용하여, 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 제로로 감소되고 그것의 스레드 식별자가 상기 다음 스레드일 때 그것을 큐잉하는 단계를 더 포함할 수 있다.
- [0110] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제어 회로를 사용하여, 실행할 스레드의 스레드 식별자에 대해, 그것의 완료 카운트가 임의의 데이터 종속성의 완료를 나타낼 때 그것을 큐잉하는 단계를 더 포함할 수 있다. 다양한 대표적인 실시 예의 어느 하나에서, 상기 완료 카운트는 복수의 스레드의 선택된 스레드마다, 상기 선택된 스레드의 실행 이전에, 수신될 완료 메시지들의 미리 결정된 수를 나타낼 수 있다.
- [0111] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 복수의 제어 레지스터에, 스레드 식별자의 각 유형이 루프 및 중첩 루프 실행에 대한 루프 레벨을 나타내는 복수의 유형의 스레드 식별자들을 갖는 완료 테이블을 저장하는 단계를 더 포함할 수 있다.
- [0112] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 복수의 제어 레지스터에 활성 루프 스레드 수의 루프 카운트를 갖는 완료 테이블을 저장하는 단계, 및 스레드 식별자 풀에 스레드 식별자를 반환하는 비동기 패

브릭 메시지의 수신에 응답하여, 상기 제어 회로를 사용하여, 상기 루프 카운트를 감소시키는 단계, 및 상기 루프 카운트가 제로에 도달할 때, 비동기 패브릭 완료 메시지를 송신하는 단계를 더 포함할 수 있다.

- [0113] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 복수의 제어 레지스터에 선택된 루프에 전용인 변수들에 대한 각 유형의 스레드 식별자 액세스를 허용하기 위한 스레드 식별자 스택의 상단을 저장하는 단계를 더 포함할 수 있다.
- [0114] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 지속 큐를 사용하여, 실행을 허용하는 완료 카운트들은 갖지만 아직 할당된 스레드 식별자는 갖지 않는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장하는 단계를 더 포함할 수 있다.
- [0115] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 재진입 큐를 사용하여, 실행을 허용하는 완료 카운트들은 갖고 할당된 스레드 식별자는 갖는 계산 스레드들에 대해 하나 이상의 스레드 식별자를 저장하는 단계를 더 포함할 수 있다.
- [0116] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 재진입 큐 내 스레드 식별자를 갖는 임의의 스레드를 상기 지속 큐 내 스레드 식별자를 갖는 임의의 스레드의 실행 이전에 실행하는 단계를 더 포함할 수 있다.
- [0117] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 우선 큐 내 스레드 식별자를 갖는 임의의 스레드를 상기 지속 큐 또는 상기 재진입 큐 내 스레드 식별자를 갖는 임의의 스레드의 실행 이전에 실행하는 단계를 더 포함할 수 있다.
- [0118] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 실행 큐 내 임의의 스레드를 상기 스레드 식별자에 대한 스포크 카운트의 발생시 실행하는 단계를 더 포함할 수 있다.
- [0119] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 제어 회로를 사용하여, 실행할 계산 스레드를 자체 스케줄링하는 단계를 더 포함할 수 있다.
- [0120] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 조건부 논리 회로를 사용하여, 다음 구성 가능한 회로에 의해 실행할 상이한 제2 다음 명령으로 브랜칭하는 단계를 더 포함할 수 있다.
- [0121] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제어 회로를 사용하여, 실행할 계산 스레드들을 순서화하는 단계를 더 포함할 수 있다.
- [0122] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제어 회로를 사용하여, 실행할 루프 계산 스레드들을 순서화하는 단계를 더 포함할 수 있다.
- [0123] 다양한 대표적인 실시 예의 어느 하나에서, 상기 방법은: 상기 제어 회로를 사용하여, 데이터 종속성으로부터의 하나 이상의 완료 신호에 응답하여 계산 스레드들의 실행을 개시하는 단계를 더 포함할 수 있다.
- [0124] 본 발명의 많은 다른 이점 및 특징은 본 발명에 대한 다음의 상세한 설명 및 그 실시 예들로부터, 청구범위 및 첨부 도면들로부터 쉽게 명백해질 것이다.

**도면의 간단한 설명**

[0125] 본 발명의 목적들, 특징들 및 이점들은 첨부된 도면들과 관련하여 고려될 때 다음 개시 내용을 참조하면 보다 쉽게 이해될 것이며, 도면들에서 같은 참조 부호들은 다양한 도면에서 동일한 구성요소들을 식별하기 위해 사용되고, 알파벳 문자들이 있는 참조 부호들은 다양한 도면에서 선택된 구성요소 실시 예의 추가 유형, 인스턴스화 또는 변형을 식별하는 데 이용된다.

도면(또는 "도") 1은 하이브리드 컴퓨팅 시스템의 대표적인 제1 실시 예의 블록도이다.

도면(또는 "도") 2는 하이브리드 컴퓨팅 시스템의 대표적인 제2 실시 예의 블록도이다.

도면(또는 "도") 3은 하이브리드 컴퓨팅 시스템의 대표적인 제3 실시 예의 블록도이다.

도면(또는 "도") 4는 제1 상호 접속 네트워크에 연결되는 구성 가능한 컴퓨팅 회로를 갖는 하이브리드 스레딩 패브릭의 대표적인 실시 예의 블록도이다.

도면(또는 "도") 5는 하이브리드 스레딩 패브릭 회로 클러스터의 대표적인 실시 예의 일부의 상위 레벨 블록도이다.

도면(또는 "도") 6은 하이브리드 스테딩 패브릭 회로 클러스터 내 제2 상호 접속 네트워크의 상위 레벨 블록도이다.

도면(또는 "도") 7은 하이브리드 스테딩 패브릭 회로 클러스터의 대표적인 실시 예의 상세한 블록도이다.

도면(또는 "도") 8은 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로(타일)의 대표적인 실시 예의 상세한 블록도이다.

도면(또는 "도") 9, 9a 및 9b(여기서 도 9는 도 9a 및 도 9b로 나뉜다(그리고 도 9로 총칭된다))는 총괄하여 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로(타일)의 대표적인 실시 예의 상세한 블록도이다.

도면(또는 "도") 10은 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로(타일)의 메모리 제어 회로의 대표적인 실시 예의 상세한 블록도이다.

도면(또는 "도") 11은 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로(타일)의 스테드 제어 회로의 대표적인 실시 예의 상세한 블록도이다.

도면(또는 "도") 12는 동기 도메인들을 형성하는 대표적인 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들) 및 대표적인 비동기 패킷 네트워크 메시징의 도해이다.

도면(또는 "도") 13은 메모리 인터페이스의 대표적인 실시 예의 블록도이다.

도면(또는 "도") 14는 디스패치 인터페이스의 대표적인 실시 예의 블록도이다.

도면(또는 "도") 15는 임의적인 제1 네트워크 인터페이스의 대표적인 실시 예의 블록도이다.

도면(또는 "도") 16은 동기 도메인들을 형성하는 대표적인 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들) 및 하이브리드 스테딩 패브릭 회로 클러스터에 의한 계산의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징의 도해이다.

도면(또는 "도") 17a, 17b 및 17c(총괄하여 도 17)는 하이브리드 스테딩 패브릭 회로 클러스터에 의한 도 16의 계산의 수행을 위한 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들)에 의한 대표적인 비동기 패킷 네트워크 메시징 및 실행의 흐름도이다.

도면(또는 "도") 18은 동기 도메인들을 형성하는 대표적인 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들) 및 하이브리드 스테딩 패브릭 회로 클러스터에 의한 계산의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징의 도해이다.

도면(또는 "도") 19a 및 19b(총괄하여 도 19)는 하이브리드 스테딩 패브릭 회로 클러스터에 의한 도 18의 계산의 수행을 위한 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들)에 의한 대표적인 비동기 패킷 네트워크 메시징 및 실행의 흐름도이다.

도면(또는 "도") 20은 동기 도메인들을 형성하는 대표적인 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들) 및 하이브리드 스테딩 패브릭 회로 클러스터에 의한 계산의 루프의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징의 도해이다.

도면(또는 "도") 21은 하이브리드 스테딩 패브릭 회로 클러스터에 의한 도 20의 계산의 루프의 수행을 위한 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들)에 의한 대표적인 비동기 패킷 네트워크 메시징 및 실행의 흐름도이다.

도면(또는 "도") 22는 대표적인 흐름 제어 회로의 블록도이다.

도면(또는 "도") 23은 동기 도메인들을 형성하는 대표적인 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들) 및 하이브리드 스테딩 패브릭 회로 클러스터에 의한 계산의 루프의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징 및 동기 메시징의 도해이다.

도면(또는 "도") 24는 조건부 브랜칭 회로의 대표적인 실시 예의 블록 및 회로도이다.

**발명을 실시하기 위한 구체적인 내용**

본 발명은 많은 상이한 형태의 실시 예가 가능하지만, 본 개시가 본 발명의 원리들의 예증으로 간주되어야 하고 본 발명을 예시되는 구체적인 실시 예들로 제한하려는 것이 아니라는 것을 이해하여, 이의 구체적인 대표적인

[0126]

실시 예들이 도면들에 도시되고 상세히 여기에 설명될 것이다. 이와 관련하여, 본 발명에 따라 적어도 하나의 실시 예를 상세히 설명하기 전에, 본 발명은 그 적용에 있어서 위아래에 제시되거나, 도면들에 도시되거나, 또는 예들에 설명되는 대로 구성의 세부 사항들 및 구성요소들의 배열들로 제한되지 않는 것으로 이해되어야 한다. 본 발명에 따른 방법들 및 장치들은 다른 실시 예들이 가능하고 다양한 방식으로 실시 및 수행될 수 있다. 또한, 여기서, 뿐만 아니라 아래에 포함되는 요약에서 채용되는 어법 및 용어는 설명을 위한 것이고 제한하는 것으로 간주되어서는 안 된다는 것을 이해해야 한다.

[0127] 도 1, 도 2 및 도 3은 하이브리드 컴퓨팅 시스템(100A, 100B, 100C)(시스템(100)으로 총칭됨)의 대표적인 제1, 제2 및 제3 실시 예들의 블록도들이다. 도 4는 제1 상호 접속 네트워크(150)에 연결되는 구성 가능한 컴퓨팅 회로를 갖는 하이브리드 스레딩 패브릭("HTF", hybrid threading fabric)(200)의 대표적인 실시 예의 블록도이다. 도 5는 하이브리드 스레딩 패브릭 클러스터(205)의 대표적인 실시 예의 일부의 상위 레벨 블록도이다. 도 6은 하이브리드 스레딩 패브릭 클러스터(205) 내 제2 상호 접속 네트워크의 상위 레벨 블록도이다. 도 7은 하이브리드 스레딩 패브릭(HTF) 클러스터(205)의 대표적인 실시 예의 상세한 블록도이다.

[0128] 도 8은 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로(210)("타일"(210)이라 한다)의 대표적인 실시 예의 상세한 블록도이다. 도 9, 9a 및 9b(여기서 도 9는 도 9a 및 도 9b로 나뉜다(그리고 도 9로 총칭된다))는 총괄하여 타일(210)의 특정 대표적인 인스턴스화로서, 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로(210A)("타일"(210A)이라 한다)의 대표적인 실시 예의 상세한 블록도이다. 타일(210A)로 구체적으로 지칭되지 않는 한, 타일(210)에 대한 지칭은 개별적으로 그리고 총괄하여, 타일(210) 및 타일(210A)을 의미하고 지칭할 것이다. 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로(210)는 대표적인 실시 예에서 그러한 모든 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로(210)가 서로 동일하고 임의의 순서로 배열 및 연결될 수 있기 때문에, 즉, 각 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로(210)가 하이브리드 스레딩 패브릭 클러스터(205)를 형성하기 위해 "타일링"될 수 있기 때문에, "타일"(210)로 지칭된다.

[0129] 도 1 내지 도 9를 참조하면, 하이브리드 컴퓨팅 시스템(100)은 제1 상호 접속 네트워크(150)를 통해 하나 이상의 하이브리드 스레딩 패브릭("HTF", hybrid threading fabric) 회로(200)에 연결되는 하이브리드 스레딩 프로세서("HTP", hybrid threading processor)(300)를 포함한다. 여기서 사용될 때, 용어 "패브릭"은 이 경우 재구성 가능한 컴퓨팅 회로들인 컴퓨팅 회로들의 어레이를 의미하고 포함한다. 도 1, 도 2 및 도 3은 비교적 더 크고 더 작은 시스템들(100)을 형성하는 추가 구성 요소들을 포함하는 상이한 시스템(100A, 100B 및 100C) 배열들을 도시하며, 이들 중 어느 하나 그리고 모두는 본 개시 내용의 범위 내에 있다. 예를 들어 그리고 제한 없이, 각각 시스템 온 칩("SOC", system-on-a-chip)에 적합한 배열일 수 있는 도 1 및 도 2에 도시된 바와 같이, 하이브리드 컴퓨팅 시스템(100A, 100B)은 도시된 바와 같은 다양한 조합으로, 임의적으로, 메모리(125)에 연결될 수 있는 메모리 제어기(120)(또한 별개의 집적 회로일 수도 있음), 다양한 통신 인터페이스(130)(이들테면 PCIe 통신 인터페이스) 중 어느 하나, 하나 이상의 호스트 프로세서(들)(110), 및 호스트 인터페이스("HIF", host interface)(115)를 또한 포함할 수 있다. 예를 들어 그리고 제한 없이, 공통 기관(101)상의 "칩렛(chiplet)" 구성에 적합한 배열일 수 있는 도 3에 도시된 바와 같이, 하이브리드 컴퓨팅 시스템(100C)은 이러한 다른 구성 요소들을 포함하거나 포함하지 않고, 임의적으로 통신 인터페이스(130)를 또한 포함할 수 있다. 임의의 그리고 모든 이러한 배열은 본 개시의 범위 내이고, 집합체는 여기서 시스템(100)으로 지칭된다. 임의의 이러한 하이브리드 컴퓨팅 시스템들(100)은 또한 단일 운영 체제("OS", operating system) 하에서 작동하는 "노드"로 간주될 수 있고, 기타 그러한 로컬 및 원격 노드들에 연결될 수도 있다.

[0130] 하이브리드 스레딩은 여기서 사용될 때, 상이한 이기종 유형들의 처리 회로들(하드웨어)에 걸쳐, 이를테면 HTF 회로들(200)(재구성 가능한 컴퓨팅 패브릭)에 걸쳐 그리고 RISC-V 프로세서와 같은 프로세서에 걸쳐 다수의 계산 파이버 및 스레드를 생성할 수 있는 기능을 지칭한다. 하이브리드 스레딩은 또한 작업 스레드가 하나의 컴퓨터 요소로부터 다음 컴퓨터 요소로 전환하여 컴퓨터를 데이터가 위치되는 곳으로 이동시키는 프로그래밍 언어/스타일을 지칭하며, 이는 대표적인 실시 예들에서도 구현된다. 또한 대표적인 실시 예에서, HTP(300)은 RISC-V ISA 기반 멀티 스레드 프로세서이다. HTP(300)은 높은 클럭당 명령율을 유지하기 위해 배럴 스타일의 라운드-로빈 순시 스레드 스위칭을 제공한다. 호스트 프로세서(110)는 통상적으로 하이브리드 컴퓨팅 시스템(100) 내에 임베드될 수 있거나 PCIe 기반과 같은 통신 인터페이스(130)를 통해 하이브리드 컴퓨팅 시스템(100)에 연결되는 외부 호스트 프로세서일 수 있는 멀티 코어 프로세서이다. 이러한 프로세서들, 이를테면 HTP(300) 및 하나 이상의 호스트 프로세서(들)(110)는 아래에서 더 상세히 설명된다.

[0131] 메모리 제어기(120)는 전자 분야에 알려져 있거나 알려지게 될 바와 같이 구현될 수 있다. 대안적으로, 대표적인 실시 예에서, 메모리 제어기(120)는 관련 애플리케이션들에서 설명되는 바와 같이 구현될 수 있다. 메모리

(125)는 또한 전자 분야에 알려져 있거나 알려지게 될 바와 같이 구현될 수 있고, 아래에서 더 상세히 설명된다.

[0132] 여기서의 목적을 위해, HIF(115)는 양자 모두 제1 상호 접속 네트워크(150)를 통해 송신되는 작업 기술자 패킷들로서, 작업을 HTP(300) 및 HTF 회로들(200)로 전송하는 호스트 프로세서(110), 및 작업을 HTF 회로들(200)로 전송하는 HTP(300)를 제공한다. HIF(115)는 디스패치 회로 및 큐(약칭 "디스패치 큐"(105))를 포함하며, 이는 또한 HTF 회로들(200) 및/또는 HTP(300)에 제공되는 로드 및 리소스 가용성을 모니터링하기 위한 관리 기능을 제공한다. 자원들이 HTF 회로들(200) 및/또는 HTP(300)상에서 이용 가능할 때, 디스패치 큐(105)는 최소 로드를 갖는 HTF 회로(200) 및/또는 HTP(300) 자원을 결정한다. 다수의 HTF 회로 클러스터(205)가 동일하거나 유사한 작업 로딩을 갖는 경우, 그것은 가능하면 현재 동일한 커널을 실행하고 있는 HTF 회로 클러스터(205)를 선택한다(커널 구성 로딩 또는 다시 로딩할 필요가 없도록). HIF(115)의 유사한 기능은 또한 예를 들어, 특히 별개의 HIF(115)를 포함하지 않을 수 있는 시스템(100) 배열들의 경우, HTP(300)에 포함될 수 있다. 기타 HIF(115) 기능들은 아래에서 더 상세히 설명된다. HIF(115)는 예를 들어, (FIFO, 큐 등을 형성하는) 레지스터들을 갖는 하나 이상의 상태 기계과 같이, 전자 기술에 알려져 있거나 알려지게 될 바와 같이 구현될 수 있다.

[0133] 제1 상호 접속 네트워크(150)는 HTF 회로들(200), 하이브리드 스테딩 프로세서(300) 및 메모리 제어기(120), 통신 인터페이스(130), 및 호스트 프로세서(110)와 같은 기타 임의적 구성요소들 사이에서 그리고 그것들 중에서 데이터 패킷 라우팅을 제공하는 패킷 기반 통신 네트워크이다. 제1 상호 접속 네트워크(150)는 통상적으로 시스템(100) 실시 예에 따라, folded clos 구성을 갖는 복수의 크로스바 스위치 및 통상적으로 추가 연결들을 위한 메시 네트워크로서 구현된다. 본 개시의 목적을 위해, 제1 상호 접속 네트워크(150)는 비동기 스위칭 패브릭("AF", asynchronous switching fabric)의 일부를 형성하는데, 이는 데이터 패킷이 다양한 경로 중 어느 하나의 경로를 따라 라우팅되어, 라우팅에 따라, 어드레싱된 목적지에 임의의 선택된 패킷의 도착이 복수의 상이한 시간 중 임의의 시간에 발생할 수 있게 될 수 있다는 것을 의미한다. 이는 아래에서 더 상세히 논의될 제2 상호 접속 네트워크(250)의 동기 메시 통신 네트워크(275)와 대조적이다.

[0134] 차례로, HTF 회로(200)는 통상적으로 각 HTF 회로 클러스터(205)가 데이터 패킷 통신을 위해 제1 상호 접속 네트워크(150)에 연결되는 복수의 HTF 회로 클러스터(205)를 포함한다. 각 HTF 회로 클러스터(205)는 다른 HTF 회로 클러스터들(205) 각각과 독립적으로 동작할 수 있다. 차례로, 각 HTF 회로 클러스터(205)는 여기서 서로 같게 "타일들"(210)로 지칭되는 복수의 HTF 재구성 가능한 컴퓨팅 회로(210)의 어레이, 및 제2 상호 접속 네트워크(250)를 포함한다. 타일들(210)은 아래에서 더 상세히 논의될 두 가지 상이한 유형의 네트워크를 포함하는 제2 상호 접속 네트워크(250)에 임베드되거나 다른 방식으로 연결된다. 대표적인 실시 예에서, 각 HTF 회로 클러스터(205)는 또한 메모리 인터페이스(215), 임의적인 제1 네트워크 인터페이스(220)(이는 제1 상호 접속 네트워크(150)에 연결하기 위한 인터페이스를 제공한다) 및 디스패치 인터페이스(225)를 포함한다. 다양한 메모리 인터페이스(215), 디스패치 인터페이스(225) 및 임의적인 제1 네트워크 인터페이스(220)는 하나 이상의 상태 기계 회로와 같은 임의의 적절한 회로를 사용하여, 아래에서 더 상세히 지정되는 기능을 수행하도록 구현될 수 있다.

[0135] 개요로서, HTF 회로(200)는 상호 접속된 컴퓨팅 타일들(210)로 구성된 낮은 레벨로 세분화되는 재구성 가능한 컴퓨터 패브릭이다. 타일들(210)은 동기 메시 통신 네트워크(275)로 지칭되는 동기 패브릭과 상호 접속되어, 데이터가 큐잉 없이 하나의 타일(210)로부터 다른 타일(210)로 순회할 수 있게 한다. 이러한 동기 메시 통신 네트워크(275)는 함께 파이프라이닝된 많은 타일(210)이 산술 연산들을 통해 지속적인 데이터 흐름을 생성하게 하고, 하나 이상의 계산 스레드의 수행을 위해 동기 메시 통신 네트워크(275)를 통해 접속되는 타일들(210)의 그러한 각 파이프라인은 여기서 직렬 연결들, 병렬 연결들 그리고 가능하게는 브랜칭 연결들도 가질 수 있는 "동기 도메인(synchronous domain)"으로 지칭된다. 동기 도메인의 첫 번째 타일(210)은 여기서 "베이스" 타일(210)로 지칭된다.

[0136] 타일들(210)은 또한 비동기 패킷 네트워크(265)로 지칭되는 비동기 패브릭과 상호 접속되어 비동기 연산들에 의해 컴퓨터의 동기 도메인들이 브리징될 수 있게 하며, 대표적인 실시 예들에서 비동기 패킷 네트워크(265)상의 모든 패킷은 단일 클럭 사이클에서 전달될 수 있다. 이러한 비동기 작업들에는 동기 도메인 작업들의 개시, 하나의 동기 도메인으로부터 다른 동기 도메인으로의 데이터 전달, 시스템 메모리(125)에 액세스(판독 및 기록), 브랜칭 및 루핑 구성들의 수행이 포함된다. 동기 및 비동기 패브릭들은 함께, 타일들(210)이 고급 언어 구성들을 효율적으로 실행할 수 있게 한다. 비동기 패킷 네트워크(265)는 예를 들어 그리고 제한 없이, 적은 어드레싱을 요구하고, 단일 채널이 되고, 깊이 기반 역압으로 큐잉되며, 이를테면 128 비트의 데이터 경로로 패키징된 데이터 피연산자들을 이용하는 것을 포함하여, 많은 면에서 제1 상호 접속 네트워크(150)와 상이하다. 다양한 타일(210)의 내부 데이터 경로들은 또한 예를 들어 제한없이, 또한 128 비트라는 점에 유의해야 한다. 동기 도메

인들의 예들, 및 비동기 패킷 네트워크(265)를 통해 서로 통신하는 동기 도메인들의 예들이 예를 들어 제한 없이 도 16, 도 18 및 도 20에 도시된다.

[0137] 대표적인 실시 예에서, 대부분의 상황에서, 스레드(예를 들어, 커널) 실행 및 제어 시그널링은 이러한 두 가지 상이한 네트워크 간에서 구분되며, 스레드 실행은 동기 메시 통신 네트워크(275)를 사용하여 다양한 타일(210)의 복수의 동기 도메인을 형성하여 발생하고, 제어 시그널링은 다양한 타일(210) 사이에서 그리고 그것들 중에서 비동기 패킷 네트워크(265)를 통해 송신되는 메시징 패킷들을 사용하여 발생한다. 예를 들어, 복수의 구성 가능한 회로는 동기 메시 통신 네트워크(275)를 사용하여 복수의 동기 도메인을 형성하여 복수의 계산을 수행하도록 구성되고, 복수의 구성 가능한 회로는 복수의 제어 메시지를 생성하고 비동기 패킷 네트워크(265)를 통해 송신하도록 더 구성되며, 복수의 제어 메시지는 예를 들어 그리고 제한 없이, 하나 이상의 완료 메시지 및 계속 메시지를 포함한다.

[0138] 대표적인 실시 예에서, 제2 상호 접속 네트워크(250)는 통상적으로 도 6 및 도 7에 도시된 바와 같이, 제1 비동기 패킷 네트워크(265)가 제2 동기 메시 통신 네트워크(275)와 오버레이 또는 조합되어, 각각 타일들(210) 사이에 그리고 그것들 중에 데이터 통신을 제공하는 두 가지 상이한 유형의 네트워크를 포함한다. 비동기 패킷 네트워크(265)는 통상적으로 크로스바 스위치들(예를 들어 그리고 제한 없이, Clos 또는 Folded Clos 구성을 추가로 또는 임의적으로 가질 수도 있고 가지지 않을 수도 있다)로서 구현되는 복수의 AF 스위치(260), 및 AF 스위치들(260)을 타일(210)에 연결하여, 타일들(210)과 아래에서 논의될 다른 도시된 구성요소들 사이에 그리고 그것들 중에 데이터 패킷 통신을 제공하는 복수의 통신 라인(또는 와이어)(280, 285)으로 구성된다. 동기 메시 통신 네트워크(275)는 타일들(210) 사이의 그리고 그것들 중의 통신 라인들(또는 와이어들)(270)을 통한 복수의 직접(즉, 스위칭되지 않은, 점대 점) 연결을 제공하며, 이것들은 타일(210)의 입력들 및 출력들에서 레지스터 스테이징될 수 있으나, 그렇지 않으면 타일들(210) 간의 큐잉이 없고 데이터 패킷의 형성을 필요로 하지 않을 수 있다. (도 6에서, 타일들(210)이 제2 상호 접속 네트워크(250)에 임베드된 두 네트워크의 오버레이를 더 양호하게 설명하기 위해, 표기될 때, 타일들(210)은 동기 메시 통신 네트워크(275)의 정점들로 표현되고, AF 스위치들(260)은 "Xs"로 도시된다.)

[0139] 도 8을 참조하면, 타일(210)은 하나 이상의 구성 가능한 계산 회로(155), 제어 회로(145), 하나 이상의 메모리(325), 구성 메모리(예를 들어, RAM)(160), 동기 네트워크 입력(들)(135)(동기 메시 통신 네트워크(275)에 연결됨), 동기 네트워크 출력(들)(170)(또한 동기 메시 통신 네트워크(275)에 연결됨), 비동기 (패킷) 네트워크 입력(들)(140)(비동기 패킷 네트워크(265)에 연결됨), 및 비동기 (패킷) 네트워크 출력(들)(165)(또한 비동기 패킷 네트워크(265)에 연결됨)을 포함한다. 이러한 다양한 구성요소 각각은 도시된 바와 같이 다양한 조합으로 버스들(180, 185)을 통해 서로 연결된 것으로 도시되어 있다. 전자 기술 분야의 통상의 기술자들은 보다 적은 또는 보다 많은 구성요소가 연결들의 다양한 조합 중 임의의 조합으로, 타일(210)에 포함될 수 있음을 인식할 것이고, 이들 중 임의의 것 및 모두는 동등하고 본 개시의 범위 내에 있는 것으로 간주된다.

[0140] 이러한 다양한 구성요소 각각의 대표적인 예들이 도 9를 참조하여 아래에서 예시되고 논의된다. 예를 들어, 대표적인 실시 예에서, 하나 이상의 구성 가능한 계산 회로(155)는 예를 들어 그리고 제한 없이, 이를테면 중간 멀티플렉서들(365)을 통해 연관된 구성 기능들, 및 레지스터들(312)과 같은 연관된 레지스터들과 함께, 곱셈 및 이동 연산 회로(multiply and shift operation circuit)("MS Op")(305) 및 산술, 논리 및 비트 연산 회로(Arithmetic, Logical and Bit Operation circuit)("ALB Op")(310)로서 구현된다. 또한 대표적인 실시 예에서, 하나 이상의 구성 가능한 계산 회로(155)는 또한 예를 들어 그리고 제한 없이, 기록 마스크 생성기(375) 및 조건부 (브랜치) 논리 회로(370)를 포함할 수 있다. 또한 대표적인 실시 예에서, 제어 회로(145)는 예를 들어 그리고 제한 없이, 메모리 제어 회로(330), 스레드 제어 회로(335), 및 제어 레지스터들(340), 이를테면 타일(210A)에 대해 예시된 것들을 포함할 수 있다. 예들을 계속하여, 동기 네트워크 입력(들)(135)은 입력 레지스터들(350) 및 입력 멀티플렉서들(355)로 구성될 수 있고, 동기 네트워크 출력(들)(170)은 출력 레지스터들(380) 및 출력 멀티플렉서들(395)로 구성될 수 있고, 비동기 (패킷) 네트워크 입력(들)(140)은 AF 입력 큐들(360)로 구성될 수 있으며, 비동기 (패킷) 네트워크 출력(들)(165)은 AF 출력 큐들(390)로 구성될 수 있고, 또한 AF 메시지 상태 기계(345)를 포함하거나 공유할 수 있다.

[0141] 중요하게, 그리고 아래에서 더 상세히 논의되는 바와 같이, 구성 메모리(예를 들어, RAM)(160)는 구성 회로(이를테면 구성 메모리 멀티플렉서(372)) 및 상이한 구성 기능들을 수행하는 두 개의 상이한 구성 저장소, 즉 제1 명령 RAM(315)(이는 타일(210)의 내부 데이터 경로를 구성하는 데 사용됨) 및 제2 명령(및 인덱스) RAM(320)(여기서 "스포크(spoke)" RAM(320)으로 지칭됨)(이는 현재 명령과 무관한 타일(210)의 부분들을 구성하는 것, 현재 명령 및 다음 타일(210)의 명령을 선택하는 것, 그리고 특히, 마스터 동기 입력을 선택하는 것(모두 아래에서

더 상세히 논의됨)을 포함하여 다목적으로 사용됨)으로 구성된다.

[0142] 도 8 및 도 9에 도시된 바와 같이, 통신 라인들(또는 와이어들)(270)은 통신 라인들(또는 와이어들)(270A 및 270B)로 도시되어 있으며, 통신 라인들(또는 와이어들)(270A)은 입력 레지스터들(350)로 데이터를 공급하는 "입력들"(입력 통신 라인들(또는 와이어들))이고, 통신 라인들(또는 와이어들)(270B)은 출력 레지스터들(380)로부터 데이터를 이동하는 "출력들"(출력 통신 라인들(또는 와이어들))이다. 적인 실시 예에서, 아래에서 더 상세히 논의되는 바와 같이, 각 타일(210) 안팎으로, 각 인접한 타일(예를 들어, 동기 메시 통신 네트워크(275) 업 링크, 다운 링크, 좌측 링크 및 우측 링크)로부터 그리고 그것으로, 그리고 하나의 타일(210)로부터 다른 타일(210)로 제공되는 데이터 기록 마스크들, 중지 신호들, 및 명령들 또는 명령 인덱스들과 같은 다양한 신호의 분배를 위한 다른 구성요소들로부터 그리고 그것들로 통신 라인들(또는 와이어들)(270)의 복수의 세트 또는 버스가 있다. 대안적으로, 그리고 별도로 도시되지는 않았지만, 또한 HTF 회로 클러스터(205)의 임의의 타일(210)로부터 생성된 중지 신호가 HTF 회로 클러스터(205)의 다른 모든 타일(210)에 의해 제한된 수의 클록 사이클에서 즉시 수신될 수 있도록, 이를테면 중지 신호를 어췌트하기 위한 다양한 전용 통신 라인이 있을 수도 있다.

[0143] 동기 메시 통신 네트워크(275)를 형성하는 통신 라인들의 다양한 세트 또는 버스에 다양한 필드가 있음을 주목해야 한다. 예를 들어, 도 8 및 도 9는 각각, 착신 및 발신 통신 라인들(또는 와이어들)(270A 및 270B)의 네 개의 버스를 도시한다. 이러한 통신 라인들(또는 와이어들)(270A 및 270B)의 세트들 각각은 데이터, 명령 인덱스, 제어 정보 및 스프레드 정보(이를테면 TID, XID, 루프 종속성 정보, 유효 비트들의 선택을 위한 기록 마스크 비트들)와 같은 상이한 정보를 전달할 수 있다. 입력들(270A) 중 하나는 또한 아래에서 더 자세히 논의될 바와 같이, 예를 들어 그리고 제한 없이, 타일(210) 내부의 입력(출력의 피드백으로부터)을 포함하는 마스터 동기 입력으로 지정될 수 있으며, 이는 타일(210)의 각 시간 슬라이스마다 달라질 수 있으며, 이는 동기 도메인의 타일(210)에 대한 명령 인덱스에 대한 데이터를 가질 수 있다.

[0144] 또한, 아래에서 더 상세히 논의될 바와 같이, 동기 메시 통신 네트워크(275)상에 수신되고 하나 이상의 입력 레지스터(350)(동기 네트워크 입력(들)(135)의)에 홀딩되는 임의의 입력에 대해, 각 타일(210)은 그 입력을 동기 메시 통신 네트워크(275)의 다른 위치로의 출력을 위해 하나 이상의 출력 레지스터(380)(동기 네트워크 출력(들)(170)의)로 직접 전달할 수 있으며(통상적으로 단일 클록 사이클에서), 이에 의해 첫 번째 타일(210)이 하나 이상의 중간 두 번째 타일(210)을 통해, HTF 회로 클러스터(205) 내 임의의 다른 세 번째 타일과 통신할 수 있게 한다. 이러한 동기 메시 통신 네트워크(275)는 스프레드는 동기 도메인으로서 타일들(210) 사이에 그리고 그것들 중에 선택된 선택된 데이터 경로를 따라 시작되면, 데이터 처리의 완료가 고정된 시간 기간 내에 발생하도록, 타일들(210) 사이에 그리고 그것들 중에 정적으로 스케줄링된 동기 파이프 라인의 구성(및 재구성)을 가능하게 한다. 또한, 동기 메시 통신 네트워크(275)는 메모리(125)에 대한 액세스가 타일들(210) 사이에 그리고 그것들 중에 선택된 데이터 경로를 따라 수행되는 스프레드에 대한 계산을 완료하는 데 필요하지 않을 수 있기 때문에, 메모리(125)에 대한 임의의 필요한 액세스 수를 최소화하는 역할을 한다.

[0145] 비동기 패킷 네트워크(265)에서, 각 AF 스위치(260)는 통상적으로 통신 라인들(또는 와이어들)(280)을 통해 복수의 타일(210) 및 하나 이상의 다른 AF 스위치(260)에 연결된다. 또한, 하나 이상의 선택된 AF 스위치(260)는 또한 메모리 인터페이스(215), 임의적인 제1 네트워크 인터페이스(220), 및 디스패치 인터페이스(225) 중 하나 이상에(통신 라인들(또는 와이어들)(285)을 통해) 연결된다. 도시된 바와 같이, 예로서 그리고 제한 없이, HTF 회로 클러스터(205)는 하나의 디스패치 인터페이스(225), 두 개의 메모리 인터페이스(215), 및 두 개의 임의적인 제1 네트워크 인터페이스(220)를 포함한다. 또한 도시된 바와 같이, 예로서 그리고 제한 없이, 다른 AF 스위치들(260)에 연결되는 것에 더하여, AF 스위치들(260) 중 하나는 메모리 인터페이스(215), 임의적인 제1 네트워크 인터페이스(220), 및 디스패치 인터페이스(225)에 더 연결되는 한편, AF 스위치들(260) 중 다른 하나는 메모리 인터페이스(215) 및 임의적인 제1 네트워크 인터페이스(220)에 더 연결된다.

[0146] 선택된 실시 예에 따라, 메모리 인터페이스들(215) 각각 및 디스패치 인터페이스(225)는 또한 제1 상호 접속 네트워크(150)에 직접 연결될 수도 있으며, 제1 상호 접속 네트워크(150) 및 비동기 패킷 네트워크(265) 양자를 통해 데이터 패킷들을 수신, 생성 및 송신할 수 있는 기능은 HTF 회로 클러스터들(205)에서 이용되거나 그것에 포함되지 않는다. 예를 들어, 디스패치 인터페이스(225)는 제1 상호 접속 네트워크(150)로 그리고 그것으로부터의 데이터 패킷의 송신을 위해 다양한 타일(210) 중 임의의 것에 의해 이용될 수 있다. 다른 실시 예들에서, 메모리 인터페이스들(215) 및 디스패치 인터페이스(225) 중 임의의 것은 이를테면 제1 상호 접속 네트워크(150)에 필요한 추가 어드레싱을 제공하는 데 제1 네트워크 인터페이스(220)를 사용하기 위해, 제1 상호 접속 네트워크(150)를 통해 데이터 패킷들을 수신, 생성 및 송신하는 데 제1 네트워크 인터페이스(220)를 이용할 수 있다.

- [0147] 전자 기술분야의 통상의 기술자들은 AF 스위치들(260), 타일들(210), 임의적인 제1 네트워크 인터페이스들(220), 메모리 인터페이스들(215), 및 디스패치 인터페이스(225) 사이의 그리고 그것들 중의 연결들이 임의의 선택된 조합으로 발생할 수 있음을 인식할 것이며, 임의의 선택된 수의 구성요소 및 그러한 모든 구성요소 선택 및 조합은 동등하고 본 개시의 범위 내에 있는 것으로 간주된다. 전자 기술분야의 통상의 기술자들은 HTF 회로(200)를 복수의 HTF 클러스터(205)로 분할할 필요가 없고, 단지 다양한 구성요소 및 다양한 구성요소 사이의 그리고 그것들 중의 연결들을 설명하기 쉽도록 개념적 분할을 제공할 뿐임을 인식할 것이다. 예를 들어, HTF 회로 클러스터(205)는 16개의 타일(210)을 네 개의 AF 스위치(260), 하나의 디스패치 인터페이스(225), 두 개의 메모리 인터페이스(215), 및 두 개의 제1 네트워크 인터페이스(220)(임의적)와 갖는 것으로 도시되어 있지만, 보다 많거나 보다 적은 임의의 이러한 구성요소들이 HTF 회로 클러스터(205) 또는 HTF 회로(200) 중 하나 또는 양자에 포함될 수 있고, 아래에서 더 상세히 설명되는 바와 같이, 임의의 선택된 실시 예에 대해, HTF 회로 클러스터(205)는 임의의 선택된 시간에 활성화(예를 들어, 전원이 켜져 있고 작동 중)일 수 있는 구성요소의 수 및 유형이 달라지도록 분할될 수 있다.
- [0148] 동기 메시 통신 네트워크(275)는 데이터 큐잉 없이도 다수의 타일(210)이 파이프 라이닝될 수 있게 한다. 동기 도메인에 참여하는 모든 타일(210)은 하나의 파이프 라인 데이터 경로로 동작한다. 하나의 파이프 라인된 데이터 경로를 형성하는 그러한 타일 시퀀스(210)의 첫 번째 타일은 여기서 동기 도메인의 "베이스" 타일(210)로 지칭되고, 그러한 베이스 타일(210)은 파이프 라인된 타일들(210)을 통해 작업 스레드를 개시한다. 베이스 타일(210)은 여기서 "스포크 카운트(spoke count)"로 지칭되는 미리 정의된 케이던스에 관해 작업을 시작하는 역할을 한다. 예를 들어, 스포크 카운트가 3이면, 베이스 타일(210)은 매 3 클럭마다 작업을 개시할 수 있다. 또한 각 타일(210) 내의 계산들이 파이프 라인될 수 있어서, 현재 작업이 실행되는 동안 다음 작업을 위해 데이터가 입력되는 것과 같이 다른 명령들이 실행되는 동안 상이한 명령들의 부분들이 수행될 수 있게 된다는 점에 주목해야 한다.
- [0149] 각각의 타일들(210), 메모리 인터페이스들(215) 및 디스패치 인터페이스(225)는 임의의 선택된 HTF 회로 클러스터(205) 내에서, 목적지 또는 종점으로서 구별되거나 고유한 어드레스를 갖는다(예를 들어, 5-비트 폭의 종점 ID로서). 예를 들어, 타일들(210)은 0 - 15의 종점 ID들을 가질 수 있고, 메모리 인터페이스들(215(0 및 1))은 20 및 21의 종점 ID들을 가질 수 있으며, 디스패치 인터페이스(225)는 18의 종점 ID를 가질 수 있다(임의적인 제1 네트워크 인터페이스(220)에는 선택된 실시 예에서 그것이 포함되지 않는 한, 어떠한 어드레스도 제공되지 않음). 디스패치 인터페이스(225)는 아래에서 더 상세히 논의될 바와 같이, 다양한 동작을 위해 구성된 하나 이상의 타일(210)에 의해 수행될 작업을 포함하는 데이터 패킷(작업 기술자 패킷으로 지칭됨)을 수신한다. 작업 기술자 패킷은 하나 이상의 인수를 가질 것이며, 그 다음 디스패치 인터페이스(225)는 이를 AF 스위치들(260)을 통해 선택된 어드레스된 타일들(210)에 송신되는 패킷 또는 메시지(AF 메시지)로서 다양한 타일에 제공하거나 배포할 것이며, 또한, 통상적으로 데이터(인수(들))를 저장하기 위한 타일 메모리(325) 내 영역의 식별 및 관련 계산들 및 그것들의 완료를 추적 및 식별하는 데 이용되는 스레드 식별자("ID")를 저장할 것이다.
- [0150] 메시지들은 비동기 패킷 네트워크(265)를 통해 소스 종점으로부터 목적지 종점으로 라우팅된다. 상이한 소스들로부터 동일한 목적지로의 메시지들은 상이한 경로들을 취하고 상이한 혼잡 레벨들에 직면할 수 있다. 메시지들은 그것들이 발송될 때와 상이한 순서로 도착할 수 있다. 메시지 메커니즘들은 비결정적 도착 순서로 제대로 작동하도록 구성된다.
- [0151] 도 13은 메모리 인터페이스(215)의 대표적인 실시 예의 블록도이다. 도 13을 참조하면, 각 메모리 인터페이스(215)는 상태 기계(및 다른 논리 회로)(480), 하나 이상의 레지스터(485) 및 임의적으로 하나 이상의 큐(474)를 포함한다. 상태 기계(480)는 비동기 패킷 네트워크(265) 및 제1 상호 접속 네트워크(150)상에서 데이터 패킷들을 수신, 생성 및 송신한다. 레지스터들(485)은 타일들(210)의 가상 어드레스들, 주어진 노드 내에서 어드레싱되는 물리적, 그리고 가상 어드레스들을 물리적 어드레스들로 변환하기 위한 다양한 테이블과 같은 어드레싱 정보를 저장한다. 임의적인 큐들(474)은 제1 상호 접속 네트워크(150) 및/또는 비동기 패킷 네트워크(265)상에서 송신을 기다리는 메시지들을 저장한다.
- [0152] 메모리 인터페이스(215)는 HTF 회로 클러스터(205) 내 타일들(210)이 DRAM 메모리와 같은 시스템 메모리(125)에 요청들을 할 수 있게 한다. 메모리 인터페이스(215)에 의해 지원되는 메모리 요청 유형들은 로드, 저장 및 원자성이다. 메모리 인터페이스(215) 관점에서, 로드는 어드레스를 메모리(125)로 전송하고 데이터가 반환된다. 기록은 어드레스 및 데이터를 메모리(125) 양자로 전송하고 완료 메시지가 반환된다. 원자성 동작은 어드레스 및 데이터를 메모리(125)로 전송하고, 데이터가 반환된다. 단지 메모리로부터 데이터를 수신하는 원자(즉, 인출 및 증가)는 메모리 인터페이스(215)에 의해 로드 요청으로 핸들링된다는 점에 유의해야 한다. 모든 메모리 인터

페이스(215) 동작은 하나의 64-비트 가상 어드레스를 필요로 한다. 동작에 대한 데이터 크기는 단일 바이트에서 64 바이트까지 달라질 수 있다. 보다 큰 데이터 페이로드 크기들이 디바이스에 더 효율적이고 사용될 수 있으나; 데이터 페이로드 크기는 대용량 데이터 블록에 대한 액세스를 검출하는 고급 언어 컴파일러의 기능에 따라 결정될 것이다.

[0153] 도 14는 디스패치 인터페이스(225)의 대표적인 실시 예의 블록도이다. 도 14를 참조하면, 디스패치 인터페이스(225)는 상태 기계(및 다른 논리 회로)(470), 하나 이상의 레지스터(475) 및 하나 이상의 디스패치 큐(472)를 포함한다. 상태 기계(470)는 비동기 패킷 네트워크(265) 및 제1 상호 접속 네트워크(150)상에서 데이터 패킷들을 수신, 생성 및 송신한다. 레지스터들(475)은 타일들(210)의 가상 어드레스들 및 다양한 타일에 분산되는 구성들 및 작업 로드들을 추적하는 매우 다양한 테이블과 같은 어드레싱 정보를 저장한다. 디스패치 큐들(472)은 제1 상호 접속 네트워크(150) 및/또는 비동기 패킷 네트워크(265)상에서 송신을 기다리는 메시지들을 저장한다.

[0154] 상술된 바와 같이, 디스패치 인터페이스(225)는 이를테면 호스트 인터페이스(115)로부터, 제1 상호 접속 네트워크(150)를 통해 작업 기술자 콜 패킷(메시지들)을 수신한다. 작업 기술자 콜 패킷은 페이로드(예를 들어, 구성, 인수 값들 등), 콜 식별자(ID) 및 반환 정보(예를 들어, 해당 중점에 대한 결과들의 제공)와 같은 다양한 정보를 가질 것이다. 디스패치 인터페이스(225)는 데이터를 메모리들(325)에 기록하는 것을 포함하여 비동기 패킷 네트워크(265)를 통해 타일들(210)로 송신할 다양한 AF 데이터 메시지를 생성할 것이며, 이 타일(210)은 베이스 타일(210)(AF 완료 메시지의 송신을 위한 베이스 타일 ID), 스레드 ID(스레드 식별자 또는 "TID")이 될 것이고, 베이스 타일(210)에 지속 메시지(예를 들어, 각 TID에 대한 완료 및 기타 카운트들을 갖는)를 전송할 것이며, 그에 따라 베이스 타일(210)이 충분한 완료 메시지들을 수신하면 실행을 시작할 수 있게 된다. 디스패치 인터페이스(225)는 스레드 ID 및 XID별마다, 어느 타일(210)로 송신된 것을 추적하기 위해 레지스터들(475)에 다양한 테이블을 유지한다. 결과들이 생성되거나 실행들이 완료될 때, 디스패치 인터페이스(225)는 AF 데이터 메시지들(완료를 나타내고 데이터를 갖는) 또는 AF 완료 메시지들(완료를 나타내지만 데이터는 없음)을 수신할 것이다. 디스패치 인터페이스(225)는 또한 커널 실행이 완료되었음을 알기 위해 그것이 수신해야 할 완료 및 데이터 메시지 수의 다양한 카운트를 유지하고(레지스터들(475)에), 결과 데이터, 콜 ID, 반환 정보(예를 들어, 요청자의 어드레스)를 갖는 작업 기술자 반환 데이터 패킷들을 어셈블하고 제1 상호 접속 네트워크(150)를 통해 송신할 것이며, TID를 해제한다. 디스패치 인터페이스(225)의 추가 특징들 및 기능은 아래에서 더 상세히 설명된다.

[0155] 상술된 바와 같이, TID들의 다수의 레벨(또는 다수의 유형)이 이용될 수 있고 통상적으로 이용된다는 점에 유의해야 한다. 예를 들어, 디스패치 인터페이스(225)는 TID 풀로부터 그것이 베이스 타일(210)로 송신하는 제1 유형의 TID를 할당한다. 차례로, 베이스 타일(210)은 예를 들어 그리고 제한 없이, 이를테면 루프들 및 중첩 루프들에서 이용되는 스레드를 추적하기 위한 제2 및 제3 유형들의 TID들과 같은 추가 TID들을 할당할 수 있다. 그 다음 이러한 상이한 TID들을 이용하여 주어진 루프에 전용인 변수들에 액세스할 수도 있다. 예를 들어, 제1 유형의 TID는 외부 루프에 사용될 수 있고, 제2 및 제3 유형들의 TID들은 중첩 루프들의 반복들을 추적하는 데 이용될 수 있다.

[0156] 또한, 제1 상호 접속 네트워크(150)를 통해 다양한 메모리 요청을 추적하기 위해 별개의 트랜잭션 ID가 이용된다는 점에 유의해야 한다.

[0157] 도15는 임의적인 제1 네트워크 인터페이스의 대표적인 실시 예의 블록도이다. 도 15를 참조하면, 포함될 때 각 제1 네트워크 인터페이스(220)는 상태 기계(및 기타 논리 회로)(490) 및 하나 이상의 레지스터(495)를 포함한다. 상태 기계(490)는 비동기 패킷 네트워크(265) 및 제1 상호 접속 네트워크(150)상에서 데이터 패킷들을 수신, 생성 및 송신한다. 레지스터들(495)은 타일들(210)의 가상 어드레스들, 주어진 노드 내에서 어드레싱되는 물리적, 그리고 가상 어드레스들을 물리적 어드레스들로 변환하기 위한 다양한 테이블과 같은 어드레싱 정보를 저장한다.

[0158] 다시 도 9를 참조하면, 대표적인 HTF 재구성 가능한 컴퓨팅 회로(타일)(210A)는 적어도 하나의 곱셈 및 이동 연산 회로("MS Op")(305), 적어도 하나의 산술, 논리 및 비트 연산 회로("ALB Op")(310), 제1 명령 RAM(315), 여기서 "스포크" RAM(320)으로 지칭되는 제2 명령(및 인덱스) RAM(320), 하나 이상의 타일 메모리 회로(또는 메모리)(325)(메모리 "0"(325A), 메모리 "1"(325B) 내지 메모리 "N"(325C)로 도시되고, 개별적으로 그리고 총괄하여 메모리(325) 또는 타일 메모리(325)로 지칭됨)를 포함한다. 또한, 전술된 바와 같이, 대표적인 타일(210A)은 또한 통상적으로 통신 라인들(또는 와이어들)(270A, 270B)을 통해 동기 메시 통신 네트워크(275)에 연결되는 입력 레지스터들(350) 및 출력 레지스터들(380), 및 비동기 패킷 네트워크(265)의 통신 라인들(또는 와이어들)(280)을 통해 AF 스위치들(260)에 연결되는 AF 입력 큐들(360) 및 AF 출력 큐들(390)을 포함한다. 제어 회로

들(145)은 또한 통상적으로 타일(210A)에 대해 예시된 메모리 제어 회로(330), 스투드 제어 회로(335), 및 제어 레지스터들(340)에 포함된다. 비동기 패킷 네트워크(265)로부터 수신되거나 그것으로 제공되는 데이터 패킷들을 디코딩하고 준비(어셈블링)하기 위해, AF 메시지 상태 기계(345)도 또한 통상적으로 타일(210)에 포함된다. 타일(210)의 구성 가능성의 부분으로서, 하나 이상의 멀티플렉서는 통상적으로 MS Op(305) 및 ALB Op(310)로의 입력들의 선택을 위한 입력 멀티플렉서(355), 출력 멀티플렉서(395) 및 하나 이상의 중간 멀티플렉서(들)(365)로 도시된 것을 포함한다. 임의적으로, 기타 구성요소들, 이를테면 조건부(브랜치) 논리 회로(370), 기록 마스크 생성기(375) 및 흐름 제어 회로(385)(AF 출력 큐들(390)의 부분으로 포함된 것으로 도시되고 서로 함께, 별개의 흐름 제어 회로로서 제공될 수 있는)도 타일(210)에 포함될 수 있다. MS Op(305) 및 ALB Op(310)의 기능들은 아래에서 더 상세히 설명된다.

- [0159] 동기 메시 통신 네트워크(275)는 동기 도메인이 기능하기 위해 필요한 정보를 전달한다. 동기 메시 통신 네트워크(275)는 아래에 지정되는 필드들을 포함한다. 또한, 이러한 필드들에 사용되는 많은 파라미터는 또한 제어 레지스터들(340)에 저장되고, 복수의 타일(210)에 의해 형성되는 동기 도메인에서 실행될 스투드에 할당된다. 동기 메시 통신 네트워크(275)의 지정되는 필드들은 다음을 포함한다:
- [0160] 1. 통상적으로 64 비트의 필드 폭을 갖고, 동기 도메인에서 하나의 타일(210)로부터 다음 타일(210)로 전달되는 계산된 데이터를 포함하는 데이터.
- [0161] 2. 통상적으로 필드 폭이 8 비트이고, 다음 타일(210)에 대한 명령 RAM(315) 어드레스를 포함하는 "INSTR"로 약칭되는 명령 RAM(315) 어드레스. 베이스 타일(210)은 도메인의 첫 번째 타일(210)에 대한 명령 RAM(315) 어드레스를 지정한다. 후속 타일들(210)은 명령을 수정되지 않은 상태로 전달할 수 있거나, 조건부 실행(즉, 조건문 또는 스위치문)을 허용하는 다음 타일(210)에 대한 명령을 조건부로 변경할 수 있으며, 이는 아래에서 더 상세히 설명된다.
- [0162] 3. 통상적으로 8 비트의 필드 폭을 갖고, 커널의 스투드들에 대한 고유 식별자를 포함하는 스투드 식별자(여기서 "TID"로 지칭됨), 미리 결정된 수의 TID("TID 풀")가 제어 레지스터들(340)에 저장되고 가능하게는 스투드에 의한 사용에 이용 가능함(다른 스투드에 의해 아직 사용 중이 아닌 경우). TID는 동기 도메인의 베이스 타일(210)에 할당되고 타일 메모리(325)에 대한 판독 인덱스로서 사용될 수 있다. TID는 비동기 패킷 네트워크(265)를 통해 하나의 동기 도메인으로부터 다른 동기 도메인으로 전달될 수 있다. 다른 기능들 또는 계산들을 수행하기 위한 사용에 이용 가능한 유한 수의 TID가 있음에 따라, 결국 TID는 후속 재사용을 위해 할당된 베이스 타일의 TID 풀로 다시 해제되어야 한다. 해제는 비동기 패킷 네트워크(265)를 통해 송신되는 비동기 패킷 메시지를 사용하여 실현된다.
- [0163] 4. 통상적으로 8 비트의 필드 폭을 갖고, 하나의 동기 도메인으로부터 다른 동기 도메인으로 데이터를 전달하기 위한 고유 식별자를 포함하는 전달 식별자("XID"로 지칭됨), 미리 결정된 수의 XID("XID 풀")가 제어 레지스터들(340)에 저장되고 가능하게는 스투드에 의한 사용에 이용 가능함(다른 스투드에 의해 아직 사용 중이 아닌 경우). 전달은 "XID\_WR"로서 하나의 도메인으로부터 다른 도메인으로의 데이터의 직접 기록될 수 있거나, 또는 그것은 메모리(125) 판독("XID\_RD"로서)의 결과일 수 있으며 여기서 소스 도메인이 메모리(125)에 가상 어드레스를 전송하고 목적지 도메인이 메모리 판독 데이터를 수신한다. XID\_WR은 소스 도메인의 베이스 타일(210)에 할당된다. 소스 도메인의 XID\_WR은 목적이 도메인의 XID\_RD가 된다. XID\_WR은 목적지 도메인의 타일 메모리(325)에 대한 기록 인덱스로 사용될 수 있다. XID\_RD는 목적지 도메인에서 타일 메모리(325) 판독 인덱스로서 사용된다. 다른 기능들 또는 계산들을 수행하기 위한 사용에 이용 가능한 유한 수의 XID가 있음에 따라, 결국 XID는 후속 재사용을 위해 할당된 베이스 타일의 XID 풀로 다시 해제되어야 한다. 목적지 도메인은 또한 비동기 패킷 네트워크(265)를 통해, 소스 도메인의 베이스 타일(210)에 비동기 메시지를 전송함으로써 XID를 해제해야 한다.
- [0164] 동기 메시 통신 네트워크(275)는 데이터 및 제어 정보 양자를 제공한다. 제어 정보(INSTR, XID, TID)는 데이터 경로를 설정하는 데 사용되고, DATA 필드는 구성된 데이터 경로에 대한 소스로 선택될 수 있다. 제어 필드들은 데이터 필드보다 훨씬 일찍 (데이터 경로를 구성하는 데) 필요하다는 점에 유의한다. 타일(210)을 통한 동기 도메인 파이프 라인 지연을 최소화하기 위해, 제어 정보는 데이터보다 몇 클록 사이클 더 일찍 타일(210)에 도착한다.
- [0165] HTF 회로(200) 및 그것의 복합 HTF 회로 클러스터들(205) 및 그것들의 복합 타일들(210)의 아키텍처의 특히 창의적인 특징은 두 개의 상이한 구성 RAM, 데이터 경로 구성을 위한 명령 RAM(315), 및 임의의 선택되거나 주어진 데이터 경로와 독립적인 타일(210) 부분들의 구성, 명령 RAM(315)으로부터의 데이터 경로 명령들의 선택, 각 클록 사이클에 대한 마스터 동기 입력의 선택(이용 가능한 입력들(270A) 중)을 포함하는 다수의 기타 기능을 위

한 스포크 RAM(320)의 사용이다. 아래에서 더 상세히 논의될 바와 같이, 명령 RAM(315) 및 독립적인 스포크 RAM(320) 양자의 이러한 신규한 사용은 무엇보다도, 전체로서 HTF 회로 클러스터(205) 및 HTF 회로(200)의 동적 자체 구성 및 자체 재구성을 가능하게 한다.

[0166] 각 타일은 특정 연산에 대한 타일(210) 데이터 경로를 설정하기 위한 구성 정보, 즉, 예를 들어 곱셈, 이동, 덧셈 등이 타일(210)의 주어진 시간 슬라이스에, 그리고 그리고 어느 데이터(예를 들어, 메모리(325)로부터의 데이터, 또는 입력 레지스터(350)로부터의 데이터)를 사용하여 수행될 것인지 여부를 결정하는 데이터 경로 명령들을 포함하는 명령 RAM(315)을 갖는다. 명령 RAM(315)은 타일(210)이 시간 슬라이싱될 수 있게 하는 다수의 엔트리를 갖고, 파이프 라인된 동기 도메인에서 다수의 상이한 동작을 수행하며, 타일(210)의 대표적인 파이프 라인 섹션들(304, 306, 307, 308 및 309)이 도 9에 도시되어 있다. 임의의 주어진 명령은 또한 어느 입력들(270A)이 그러한 명령에 의해 이용될 데이터 및/또는 제어 정보를 가질 것인지를 지정할 수 있다. 또한, 각 시간 슬라이스는 도 24를 참조하여 논의될, 이전 타일(210) 시간 슬라이스 데이터 종속 조건부 동작들에 따라 상이한 명령들을 조건부로 수행할 수 있다. 명령 RAM(315) 내의 엔트리 수는 통상적으로 256 정도일 것이다. 그 수는 커널들을 HTF(200)로 포팅함으로써 얻은 경험에 따라 변할 수 있다.

[0167] 지원되는 명령 세트는 32- 및 64-비트 정수 및 부동 소수점 값들의 데이터 유형들을 갖는 애플리케이션들과 같은 타겟 애플리케이션들의 요구와 매칭되어야 한다. 기계 학습, 이미지 분석 및 5G 무선 처리와 같은 추가 애플리케이션들이 HTF(200)를 사용하여 수행할 수 있다. 이러한 전체 애플리케이션 세트에는 16, 32 및 64-비트 부동 소수점, 및 8, 16, 32 및 64-비트 정수 데이터 유형들을 필요할 것이다. 지원되는 명령 세트는 로드, 저장 및 산술 연산들을 위해 이러한 데이터 유형들을 지원해야 한다. 지원되는 동작들은 컴파일러가 고급 언어 소스를 타일(210) 명령들에 효율적으로 매핑할 수 있게 해야 한다. 대표적인 실시 예에서, 타일들(210)은 단일 명령 다중 데이터(SIMD, single instruction multiple data) 명령 변형을 포함하는 표준 고성능 프로세서와 동일한 명령 세트를 지원한다.

[0168] 스포크 RAM(320)은 다수의 기능을 갖고, 대표적인 실시 예들에서, 그러한 기능들 중 하나는 데이터 경로에 대한 현재 명령에 독립적인 타일(210)(의 시간 슬라이스)의 부분들을 구성하는 데 이용될 것이다, 즉, 스포크 RAM(320)에 홀딩되는 타일(210) 구성들은 타일(210)의 구성의 불변 부분들, 예를 들어, 상이한 데이터 경로 명령들에 걸쳐 동일하게 유지되는 타일(210)에 대한 설정들을 구성하는 데 사용될 수 있다. 예를 들어, 스포크 RAM(320)은 타일(210)의 어느 입력(예를 들어, 입력 통신 라인들(270A) 또는 입력 레지스터들(350)의 여러 세트 중 하나)이 입력 멀티플렉서(들)(355)의 선택 제어로서, 각 클록 사이클에 대해 마스터 동기 입력인지를 지정하는 데 사용된다. 이는 타일(210)의 주어진 시간 슬라이스에 대한 명령(명령 RAM(315)로부터의)을 선택하기 위한 명령 인덱스 및 스레드 ID(TID)가 마스터 동기 입력상에 제공되기 때문에 중요하다. 결과적으로, 주어진 타일(210)에 대한 입력(270A)상에 제공되는 실제 명령 인덱스가 달라질 수 있더라도(도 24를 참조하여 설명된 바와 같이), 그러한 선택된 명령 인덱스를 가질 입력들의 세트(270A)는 달라지지 않으며, 그에 따라 임의의 주어진 타일(210)은 그러한 선택된 명령 인덱스에 의해 지정된 명령과 무관하게, 그것이 선택된 명령 인덱스를 수신하는 데 사용할 입력을 미리 알게 된다. 스포크 RAM(320)에 홀딩되는 구성들은 또한 일반적으로 선택된 명령(또는 타임 슬라이스)에 대해 어느 출력들(270B)이 이용될 것인지를 지정한다. 스포크 RAM(320) 판독 어드레스 입력, 즉, 스포크 인덱스는 0으로부터 스포크 카운트 마이너스 1까지 카운트(모듈로)하는 카운터로부터 온다. HTF 회로 클러스터(205) 내 모든 타일(210)은 일반적으로 적절한 동기 도메인 동작을 갖기 위해 각 클록마다 동일한 스포크 RAM 입력 값을 가져야 한다. 스포크 RAM(320)은 또한 명령 인덱스들을 저장하고 또한 명령 RAM(315)으로부터 명령들을 선택하는 데 이용되므로, 동기 도메인의 베이스 타일(210)에 대해 스포크 RAM(320)의 카운트가 변경될 때 타일(210)에 의한 실행을 위해 일련의 명령이 선택될 수 있게 된다. 동기 도메인의 후속 타일들에 대해, 명령 인덱스는 동기 도메인의 이전 타일(210)에 의해 제공될 수 있다. 스포크 RAM(320)의 이러한 양태는 또한 HTF 회로 클러스터(205)의 동적 자체 구성 및 재구성을 가능하게 하여, 스포크 RAM(320)이 매우 창의적인 것으로 도 24를 참조하여 논의된다.

[0169] 스포크 RAM(320)은 또한 동기 입력(270A)이 타일 메모리(325)에 기록될 때를 지정한다. 이러한 상황은 타일 명령에 대해 다수의 입력이 필요하고, 입력들 중 하나가 일찍 도착하는 경우 발생한다. 일찍 도착한 입력은 타일 메모리(325)에 기록된 다음 나중에 다른 입력들이 도착했을 때 메모리(325)로부터 판독될 수 있다. 이러한 상황에서, 타일 메모리(325)는 FIFO로서 액세스된다. FIFO 판독 및 기록 포인터들은 타일 메모리 영역 램에 저장된다.

[0170] 각 타일(210)은 하나 이상의 메모리(325)를 포함하고, 통상적으로 각각은 데이터 경로(64-비트)의 폭이며, 깊이는 예를 들어, 512 내지 1024개의 요소의 범위 내일 것이다. 타일 메모리(325)는 데이터 경로 동작들을 지원하

는 데 필요한 데이터를 저장하는 데 사용된다. 저장된 데이터는 커널의 클러스터(205) 구성의 부분으로 로딩된 상수들이거나, 또는 데이터 흐름의 부분으로 계산된 변수들일 수 있다. 타일 메모리(325)는 다른 동기 도메인으로부터의 데이터 전달, 또는 다른 동기 도메인에 의해 개시된 로드 동작의 결과 중 어느 하나로서 동기 메시지 통신 네트워크(275)로부터 기록될 수 있다. 타일 메모리는 단지 동기 데이터 경로 명령 실행을 통해서만 판독된다.

[0171] 타일 메모리(325)는 통상적으로 영역들로 분할된다. 작은 타일 메모리 영역 RAM은 메모리 영역 액세스에 필요한 정보를 저장한다. 각 영역은 커널에서 상이한 변수를 나타낸다. 영역은 공유 변수(즉, 실행 중인 모든 스레드가 공유하는 변수)를 저장할 수 있다. 스칼라 공유 변수의 인덱스 값은 0이다. 공유 변수들의 어레이는 변수 인덱스 값을 갖는다. 영역은 TID 식별자로 인덱싱되는 스레드 전용 변수를 저장할 수 있다. 변수를 사용하여 하나의 동기 도메인으로부터 다음 도메인으로 데이터를 전달할 수 있다. 이 경우, 변수는 소스 동기 도메인에서 XID\_WR 식별자를 사용하여 기록되고, 목적지 도메인에서 XID\_RD 식별자를 사용하여 판독된다. 마지막으로, 다른 타일 데이터 입력들이 준비될 때까지 동기 데이터 경로에서 더 일찍 타일(210)에 의해 생성된 데이터를 일시적으로 저장하는 데 영역이 사용될 수 있다. 이 경우, 판독 및 기록 인덱스들은 FIFO 포인터들이다. FIFO 포인터들은 타일 메모리 영역 RAM에 저장된다.

[0172] 타일 메모리 영역 RAM에는 통상적으로 다음 필드들이 포함된다:

[0173] 1. 타일 메모리 영역 인덱스의 상위 비트들인 Region Index Upper. 하위 인덱스 비트들은 비동기 페브리크 메시지, TID, XID\_WR 또는 XID\_RD 식별자들 또는 FIFO 판독/기록 인덱스 값들로부터 얻어진다. Region Index Upper 비트들은 하위 인덱스 비트들과 OR' 연산되어 타일 메모리(325) 인덱스를 생성한다.

[0174] 2. 메모리 영역의 하위 인덱스의 폭인 Region SizeW. 메모리 영역의 크기는  $2^{\text{SizeW}}$  요소들이다.

[0175] 3. FIFO로서 동작하는 메모리 영역에 대한 판독 인덱스인 Region FIFO Read Index.

[0176] 4. FIFO로서 동작하는 메모리 영역에 대한 기록 인덱스인 Region FIFO Write Index. 타일은 HTF(200)에 대한 컴퓨터 동작들을 수행한다.

[0177] 컴퓨터 동작들은 타일(210) 내 데이터 경로를 구성함으로써 수행된다. 타일(210)에 대한 모든 컴퓨터를 수행하는 다음 두 개의 기능 블록이 있다: 곱하기 및 이동 연산 회로("MS Op")(305) 및 산술, 논리 및 비트 연산 회로("ALB Op")(310). S Op(305) 및 ALB Op(310)는 명령 RAM(315)로부터의 명령들에 따라 제어되고, 예를 들어 그리고 제한 없이, 곱셈 및 덧셈, 또는 이동 및 덧셈 같은 두 개의 파이프 라인된 연산들을 수행하도록 구성될 수 있다. (대표적인 실시 예에서, HTF(200)를 지원하는 모든 디바이스는 완전한 지원 명령 세트를 가질 것이다. 이는 모든 디바이스에 걸쳐 이진 호환성을 제공할 것이다. 그러나, 다이 사이즈 트레이드 오프를 충족하기 위한 기본 기능 세트 및 임의적인 명령 세트 클래스들을 갖는 것이 필요할 수 있다. 이러한 접근법은 RISC-V 명령 세트가 기본 및 다수의 임의적인 명령 서브 세트를 갖는 방식과 유사하다.) 도 9에 도시된 바와 같이, MS Op(305) 및 ALBL Op(310)의 출력들은 레지스터들(312)에 제공되거나, 출력 멀티플렉서(395), 조건부 논리 회로(370) 및/또는 기록 마스크 생성기(375)와 같은 다른 구성요소들에 직접 제공될 수 있다.

[0178] MS Op(305)에 의해 수행되는 다양한 연산은 예를 들어 그리고 제한 없이: 정수 및 부동 소수점 곱셈, 이동, 입력 중 어느 하나의 전달, 부호 있는 정수 및 부호 없는 정수 곱셈, 부호 있는 우방 이동 및 부호 없는 우방 이동, 부호 있는 좌방 이동 및 부호 없는 좌방 이동, 비트 순서 역전, 치환, 부동 소수점 연산들로서 이러한 임의의 그리고 모든 연산, 및 정수와 부동 소수점 간의 상호 변환, 이를테면 배 정도 하한 연산 또는 부동 소수점의 정수로의 변환을 포함한다. ALBL Op(310)에 의해 수행되는 다양한 연산은 예를 들어 그리고 제한 없이: 부호 있는 덧셈 및 부호 없는 덧셈, 절대 값, 부정, 논리 NOT, 덧셈 및 부정, 뺄셈 A - B, 역뺄셈 B - A, 부호 있는 초과 및 부호 없는 초과, 부호 있는 이상 및 부호 없는 이상, 부호 있는 미만 및 부호 없는 미만, 부호 있는 이하 및 부호 없는 이하, 비교(같거나 같지 않음), 논리 연산들(AND, OR, XOR, NAND, NOR, NOT XOR, AND NOT, OR NOT), 부동 소수점 연산들로서 이러한 임의의 그리고 모든 연산, 및 정수와 부동 소수점 간의 상호 변환, 이를테면 배 정도 하한 연산 또는 부동 소수점의 정수로의 변환을 포함한다.

[0179] ALB Op(310) 및 MS Op(305)에 대한 입력들은 동기 타일 입력들(270A)(레지스터들(350)에 홀딩되는), 내부 타일 메모리(325), 또는 명령 RAM(315) 내에 제공되는 작은 상수 값으로부터 온다. 타일(210) 데이터 경로 입력 소스들을 보여주는 다음 표 1은 ALB Op(310) 및 MS Op(305)에 대한 통상적인 입력들을 나열한다.

표 1

[0180]

소스명	소스 기술
SYNC_U	동기 메시 통신 네트워크(275) 업 링크
SYNC_D	동기 메시 통신 네트워크(275) 다운 링크
SYNC_L	동기 메시 통신 네트워크(275) 좌측 링크
SYNC_R	동기 메시 통신 네트워크(275) 우측 링크
TILE_OUT	타일(210) 내 ALB Op(310)의 출력.
RDMEMO_T	메모리 0 데이터 관독. 메모리(325) 영역은 마스터 동기 인터페이스로부터의 TID를 사용하여 인덱싱된다.
RDMEMO_X	메모리 0 데이터 관독. 메모리(325) 영역은 마스터 동기 인터페이스로부터의 XID를 사용하여 인덱싱된다.
RDMEMO_C	메모리 0 데이터 관독. 메모리(325) 영역은 명령 램 상수 값을 사용하여 인덱싱된다.
RDMEMO_V	메모리 0 데이터 관독. 메모리(325) 영역은 가변 인덱싱으로서 동기 입력으로부터 수신되는 값을 사용하여 인덱싱된다.
RDMEMO_F	메모리 0 데이터 관독. 메모리(325) 영역은 FIFO 순서를 사용하여 관독된다.
RDMEMO_Z	메모리 0 데이터 관독. 메모리(325) 영역은 값 0을 사용하여 인덱싱된다.
RDMEM1_T	메모리 1 데이터 관독. 메모리(325) 영역은 마스터 동기 인터페이스로부터의 TID를 사용하여 인덱싱된다.
RDMEM1_X	메모리 1 데이터 관독. 메모리(325) 영역은 마스터 동기 인터페이스로부터의 XID를 사용하여 인덱싱된다.
RDMEM1_C	메모리 1 데이터 관독. 메모리(325) 영역은 명령 램 상수 값을 사용하여 인덱싱된다.
RDMEM1_V	메모리 1 데이터 관독. 메모리(325) 영역은 가변 인덱싱으로서 동기 입력으로부터 수신되는 값을 사용하여 인덱싱된다.
RDMEM1_F	메모리 1 데이터 관독. 메모리(325) 영역은 FIFO 순서를 사용하여 관독된다.
RDMEM1_Z	메모리 1 데이터 관독. 메모리(325) 영역은 값 0을 사용하여 인덱싱된다.
CONST	데이터 경로 입력은 명령 내 0 확장 상수 값이다.
ITER_IDX	데이터 경로 입력은 아래에서 더 상세히 설명될 0 확장 루프 반복 값이다.
ITER_W	데이터 경로 입력은 0 확장 루프 반복자 너비 값이다. 자세한 내용은 루프 섹션을 참조한다.

[0181]

동기 메시 통신 네트워크(275)의 통신 라인들(270)의 부분으로서 타일(210)의 출력들(270B) 각각은 개별적으로 인에이블되어 디스에이블된 출력들의 클록 게이팅을 가능하게 한다. ALB Op(310)의 출력은 표 2에 제시된 바와 같이, 다수의 목적지로 전송될 수 있다.

표 2

[0182]

목적지명	목적지 기술
SYNC_U	동기 메시 통신 네트워크(275) 업 링크
SYNC_D	동기 메시 통신 네트워크(275) 다운 링크
SYNC_L	동기 메시 통신 네트워크(275) 좌측 링크
SYNC_R	동기 메시 통신 네트워크(275) 우측 링크
WRMEMO_Z	메모리 0 기록. 메모리(325) 영역은 인덱스로 값 0을 사용하여 기록된다.
WRMEMO_C	메모리 0 기록. 메모리(325) 영역은 명령 상수 필드를 인덱스로 사용하여 기록된다.
WRMEMO_T	메모리 0 기록. 메모리(325) 영역은 TID 값을 인덱스로 사용하여 기록된다.

[0183]

상위 레벨에서, 모두 대표적인 예들로서 그리고 제한 없이, 타일(210)의 일반적인 동작은 다음과 같다. 동기 메시 통신 네트워크(275) 및 다양한 타일(210)의 동기 도메인은 모두 프로그램 컴파일의 부분으로서 그리고 구성이 시스템에 로딩될 때 스케줄링된다. 일시 중지되거나 중지되지 않는 한, 타일(210)은 필요한 모든 입력이 준비될 때 그 동작들을 실행할 수 있다, 예를 들어, 데이터 변수들은 입력 레지스터들(350) 또는 메모리(325)에 있고, 레지스터들 또는 메모리로부터 관독되거나 취해져서 이용 가능하고 아래에서 더 상세히 설명된 바와 같이

사용된다. 대표적인 실시 예에서, 각 파이프 라인 스테이지는 단일 클록 사이클에서 동작할 수 있는 한편, 다른 대표적인 실시 예들에서는, 추가 클록 사이클들이 파이프 라인 스테이지마다 이용될 수 있다. 제1 파이프 라인 단계(304)에서, 데이터는 이를테면 AF 입력 큐들(360) 및 입력 레지스터들(350)로 입력되고, 임의적으로 메모리(325)에 직접 입력된다. 다음 파이프 라인 단계(306)에서, AF 메시지들은 AF 상태 기계(345)에 의해 디코딩되고 메모리(325)로 이동된다; AF 상태 기계(345)은 메모리(325)로부터 또는 출력 멀티플렉서들(395)로부터 수신된 데이터를 판독하고 비동기 패킷 네트워크(265)를 통한 송신을 위한 데이터 패킷을 생성한다; 입력 레지스터들(350)의 데이터는 메모리(325)로 이동되거나 (입력 멀티플렉서들(355) 및 중간 멀티플렉서들(365)을 사용하여) 피연산자 데이터로서 선택되거나, 동기 메시 통신 네트워크(275)상의 출력을 위해 출력 레지스터들(380)로 직접 전달된다; 예를 들면, 다음 파이프 라인 스테이지들(307 및 308) 중 하나 이상에서, 계산들은 ALB Op(310) 및/또는 MS Op(305)에 의해 수행되고, 기록 마스크들이 기록 마스크 생성기(375)에 의해 생성될 수 있으며, 명령들(또는 명령 인덱스들)이 조건부(브랜치) 논리 회로(370)의 테스트 조건들에 기초하여 선택될 수 있다. 다음 파이프 라인 스테이지(309)에서는, 출력 멀티플렉서들(395)을 사용하여 출력들이 선택되고, (AF 출력 큐들(390)에 저장되었을 수 있는) 출력 메시지들이 비동기 패킷 네트워크(265)상에 송신되며, 출력 레지스터들(380) 중 어느 하나의 출력 데이터는 동기 메시 통신 네트워크(275)상에 송신된다.

[0184] 도 10은 하이브리드 스테딩 패브릭 구성 가능한 컴퓨팅 회로(타일)(210)의 메모리 제어 회로(330)(관련 제어 레지스터들(340)과 함께)의 대표적인 실시 예의 상세한 블록도이다. 도 10은 메모리 제어 회로(330)의 타일 메모리(325) 판독 인덱싱 로직의 도해를 도시하고, 각 메모리(325)(별도로 도시되지 않음)에 대해 복제된다. 명령 RAM(315)은 타일 메모리(325)의 어느 영역이 액세스되고 있는지를 지정하는 필드, 및 액세스 인덱싱 모드를 지정하는 필드를 갖는다. 메모리 영역 RAM(405)(제어 레지스터들(340)의 부분)은 특정 영역에 대한 상위 메모리 어드레스 비트들을 제공하는 영역 판독 마스크를 지정한다. 마스크는 판독 인덱스 선택 맥스(403)에 의해 공급되는 하위 어드레스 비트들과 OR' 연산(OR 게이트(408))된다. 메모리 영역 RAM(405)은 또한 타일 메모리(325)가 FIFO 모드로 액세스될 때 판독 인덱스 값을 포함한다. RAM(405)의 판독 인덱스 값은 FIFO 모드로 액세스할 때 증분되고 다시 기록된다. 다양한 실시 예에서, 메모리 영역 RAM(405)은 또한 아래에서 설명되는 중첩 루프들을 통해 TID 스택의 상단을 유지할 수 있다.

[0185] 도 10은 또한 동기 메시 통신 네트워크(275)에 대한 제어 정보(INSTR, XID, TID)가 데이터 입력보다 몇 클럭 일찍 필요한 것을 도시한다. 이러한 이유로, 제어 정보는 데이터를 전송하기 몇 클럭 이전에 이전 타일(210)로부터 전송된다. 동기 메시 통신 네트워크(275) 정보의 이러한 스테이징은 타일(210)마다 전체 파이프 라인 스테이지들을 감소시키지만, 계산된 값을 타일 메모리들(325)에 대한 인덱스로 사용하는 것을 어렵게 만든다. 구체적으로, 동기 메시 통신 네트워크(275) 데이터가 타일 메모리들(325)에 대한 인덱스로서 사용되기에 너무 늦게 도착할 수 있다. 이 문제에 대한 아키텍처 해결책은 제어 레지스터들(340)의 가변 인덱스 레지스터의 이전 타일(210)로부터 계산된 인덱스를 제공하는 것이다. 이후에, 다른 입력(270A)이 가변 인덱스 레지스터가 타일 메모리(325) 인덱스로서 사용되게 한다.

[0186] 비동기 패킷 네트워크(265)는 동기 도메인에 대해 비동기적으로 발생하는 동작들을 수행하는 데 사용된다. 각 타일(210)은 도 9에 도시된 바와 같은 비동기 패킷 네트워크(265)에 대한 인터페이스를 포함한다. 인바운드 인터페이스(통신 라인들(280A)로부터의)는 즉시 처리될 수 없는 메시지들에 대한 스토리지를 제공하기 위한 AF 입력 큐들(360)(FIFO로서)이다. 유사하게, 아웃바운드 인터페이스(통신 라인들(280B)에 대한)는 즉시 발송될 수 없는 메시지들에 대한 스토리지를 제공하기 위한 AF 출력 큐들(390)(FIFO로서)이다. 비동기 패킷 네트워크(265)를 통한 메시지들은 데이터 메시지들 또는 제어 중 어느 하나로 분류될 수 있다. 데이터 메시지들은 타일 메모리들(325) 중 하나에 기록되는 64-비트 데이터 값을 포함한다. 제어 메시지들은 스레드 생성을 제어하고, 자원들(TID 또는 XID)을 해제하거나, 외부 메모리 참조를 발행하기 위한 것이다. 다음 표 3은 비동기 패킷 네트워크(265) 아웃바운드 메시지 동작들을 나열한다:

표 3

[0187]

동작명	동작 기술
FREE_XID	XID_RD를 해제하기 위해 동기 도메인의 베이스 타일(210)로 전송되는 메시지.
FREE_TID	TID를 해제하기 위해 동기 도메인의 베이스 타일(210)로 전송되는 메시지.
CONT_X	지정된 수의 완료 메시지가 수신된 후에 스레드가 개시되어야 함을 나타내기 위해 동기 도메인의 베이스 타일(210)로 전송되는 제1 유형의 지속 메시지.

CONT_T	지정된 수의 완료 메시지가 수신된 후에 스레드가 개시되어야 함을 나타내기 위해 동기 도메인의 베이스 타일(210)로 전송되는 제2 유형의 지속 메시지.
INNER_LOOP	스트립 마이닝된 루프 구성의 내부 루프를 개시하기 위해 전송되는 메시지. 메시지는 수행할 루프 반복 횟수를 지정한다. 각 반복마다 작업 스레드가 개시된다. 반복 인덱스는 데이터 경로 소스 멀티플렉서(365)에 대한 입력(ITER_IDX)으로서 베이스 타일(210) 내에서 이용 가능하다.
OUTER_LOOP	스트립 마이닝된 루프 구성의 외부 루프를 개시하기 위해 전송되는 메시지. 메시지는 수행할 루프 반복 횟수를 지정한다. 각 반복마다 작업 스레드가 개시된다. 반복 인덱스는 데이터 경로 소스 멀티플렉서(365)에 대한 입력(ITER_IDX)으로서 베이스 타일(210) 내에서 이용 가능하다.
COMP	동기 도메인 작업 스레드가 완료되었음을 나타내기 위해 전송되는 완료 메시지. 베이스 타일(210)은 후속 작업 스레드가 개시될 수 있게 하기 위해 콜 또는 계속 메시지를 수신하는 것과 함께 수신된 완료 메시지들을 카운트한다. 메시지들은 아래에서 설명될, 일시 중지 테이블 인덱스로서 TID 식별자를 전송한다.
CALL	동일한 또는 다른 동기 도메인상에서 작업 스레드를 계속하기 위해 전송되는 콜 메시지. TID 및/또는 XID는 임의적으로 작업 스레드가 개시될 때 할당될 수 있다.
CALL_DATA	동일한 또는 다른 동기 도메인상에서 작업 스레드를 계속하기 위해 전송되는 콜 데이터 메시지. TID 및/또는 XID는 임의적으로 작업 스레드가 개시될 때 할당될 수 있다. 이러한 메시지는 베이스 타일(210) 내 타일 메모리(325)에 기록될 128 비트(두 개의 64-비트 값)를 기록할 128 비트 값의 바이트들을 나타내는 마스크와 함께 전송한다. 이는 일반적으로 또한 모든 비동기 메시지의 경우에도 마찬가지이다.
DATA_R	목적지 타일(210)의 타일 메모리(325)에 기록하기 위해 전송되는 메시지. 목적지 타일의 메모리에 대한 기록 인덱스를 지정하는 데 TID 값이 사용된다. 타일 메모리(325)가 지정된 베이스 타일(210)에 기록되면 완료 메시지가 전송된다.
DATA_X	목적지 타일(210)의 타일 메모리(325)에 기록하기 위해 전송되는 메시지. 목적지 타일의 메모리에 대한 기록 인덱스를 지정하는 데 XID_WR 값이 사용된다. 타일 메모리(325)가 지정된 베이스 타일(210)에 기록되면 완료 메시지가 전송된다.
LD_ADDR_T	메모리 로드 동작을 위한 어드레스를 지정하기 위해 메모리 인터페이스(215)로 전송되는 메시지. 목적지 타일의 메모리에 대한 기록 인덱스로서 TID 식별자가 사용된다.
LD_ADDR_X	메모리 로드 동작을 위한 어드레스를 지정하기 위해 메모리 인터페이스(215)로 전송되는 메시지. 목적지 타일의 메모리에 대한 기록 인덱스로서 XID_WR 식별자가 사용된다.
LD_ADDR_Z	메모리 로드 동작을 위한 어드레스를 지정하기 위해 메모리 인터페이스(215)로 전송되는 메시지. 목적지 타일의 메모리에 대한 기록 인덱스로서 0이 사용된다.
ST_ADDR	메모리 저장 동작을 위한 어드레스를 지정하기 위해 메모리 인터페이스(215)로 전송되는 메시지.
ST_DATA	메모리 저장 동작을 위한 데이터를 지정하기 위해 메모리 인터페이스(215)로 전송되는 메시지.

[0188] 비동기 패킷 네트워크(265)는 상이한 동기 도메인들의 타일들(210)로부터 메시지들이 전송 및 수신될 수 있게 한다. 동기 도메인의 베이스 타일(210)이 TID를 할당하고 TID가 동일한 동기 도메인에 의해 해제될 때와 같이, 동기 도메인이 자체적으로 메시지를 전송하는 것이 합당한 상황은 거의 없다.

[0189] 비동기 패킷 네트워크(265)는 다양한 타일(210)의 동기 도메인이 비동기 패킷 네트워크(265) 또는 수신 종점들에 의해 수신, 라우팅 및 처리될 수 있는 것보다 더 빠르게 메시지들을 생성 및 전송하면 혼잡해질 수 있다. 이러한 상황에서 임의의 그러한 혼잡을 완화하기 위해 역압 메커니즘이 제공된다. 도 22는 대표적인 흐름 제어 회로(385)의 블록도이다. 일반적으로, HTF 회로 클러스터(205)마다 적어도 하나의 흐름 제어 회로(385)가 있다. 타일(210) 비동기 패트릭 출력 큐(390)는 비동기 패킷 네트워크(265)상에 전송되기를 기다리는 메시지들을 홀딩할 것이다. 미리 결정된 임계치가 출력 큐(390)에 제공되며, 이는 도달될 때, 타일(210)의 출력 큐(390)가 흐름 제어 회로(385)에 제공되는 통신 라인(384)상에 "중지" 신호(382)로서 어췌트되는 비트 설정과 같은 표시자를 생성하게 할 것이다. HTF 회로 클러스터(205)의 타일(210)로부터의 각 통신 라인은 흐름 제어 회로(385)로 제공된다. 흐름 제어 회로(385)는 하나 이상의 OR 게이트(386)를 가지며, 이는 타일들(210) 중 어느 하나가 중지 신호(382)를 생성하는 한, 영향을 받는 HTF 회로 클러스터(205) 내의 모든 타일(210)에 배포되는 정지 신호(382)

를 통신 라인(388)상에 계속 어썬트할 것이다.

[0190] 중지 신호(382)는 도시된 바와 같이 동기 메시 통신 네트워크(275) 또는 비동기 패킷 네트워크(265) 중 어느 하나의 부분이 아닌 전용 통신 라인(388)을 통해, 또는 동기 메시 통신 네트워크(275)를 통해 배포될 수 있다. 대표적인 실시 예에서, HTF 회로 클러스터(205) 내의 모든 타일 출력 큐(390)가 어썬트될 수 있는 하나의 중지 신호(382)가 있고, 중지 신호(382)가 어썬트될 때 그러한 HTF 회로 클러스터(205) 내의 모든 타일(210)은 홀딩(일시 중지 또는 중지)된다. 이러한 중지 신호(382)는 계속해서 모든 AF 입력 큐(360)가 AF 메시지를 및 패킷들을 수신할 수 있게 하여, 교착 상태를 피할 수 있지만, 또한 모든 동기 도메인 파이프 라인이 홀딩되거나 일시 중지되게 한다(또한 추가 AF 데이터 패킷들의 생성을 방지함). 중지 신호(382)는 비동기 패킷 네트워크(265)가 타일(210) 출력 큐(390)를 (트리거링 출력 큐(들)(390))의 출력 큐(390)의 메시지 수가 임계 레벨 아래로 떨어진 지점까지 드레인할 수 있게 한다. 출력 큐(390)의 크기가 임계 레벨 아래로 떨어지면, 통신 라인(384)을 통한 신호는 그 타일(210)에 대해 0으로 반환된다(중지 신호(382)는 더 이상 생성되지 않음). HTF 회로 클러스터(205) 내의 모든 타일(210)에 대해, 통신 라인(388)상의 신호 또한 0으로 복귀하는데, 이는 중지 신호가 더 이상 어썬트되지 않고, 타일들(210)상에서 중지 또는 일시 중지를 종료함을 의미한다.

[0191] 동기 도메인의 제1 또는 "베이스" 타일(210)은 다중 타일(210) 동기 파이프 라인을 통해 작업 스레드들을 개시할 책임을 갖는다. 미리 결정된 케이던스상에서 새로운 스레드가 개시될 수 있다. 상술된 바와 같이, 여기서 케이던스 간격이 "스포크 카운트"로 지칭된다. 예를 들어, 스포크 카운트가 3이면, 새로운 작업 스레드가 3 클럭마다 베이스 타일(210)상에서 개시될 수 있다. 새로운 스레드 시작이 스킵되면(예를 들어, 시작할 준비가 된 스레드가 없음), 다른 스레드가 시작될 수 있기 전에 풀 스포크 카운트가 대기되어야 한다. 1보다 큰 스포크 카운트는 각 물리적 타일(210)이 동기 파이프 라인 내에서 여러 번 사용될 수 있게 한다. 예로서, 동기 도메인이 하나의 물리적 타일상에서 실행되고 스포크 카운트가 1이면 동기 도메인에는 단지 하나의 타일 시간 슬라이스만 포함될 수 있다. 이러한 예에서 스포크 카운트가 4이면, 동기 도메인에는 네 개의 타일 시간 슬라이스가 포함될 수 있다. 통상적으로, 동기 도메인은 동기 메시 통신 네트워크(275)의 동기 링크들에 의해 상호 접속된 다수의 타일(210)에 의해 실행된다. 동기 도메인은 클러스터(205) 내 타일들(210)의 서브 세트에 제한되지 않는다, 즉, 다수의 동기 도메인이 클러스터(205)의 타일들(210)을 공유할 수 있다. 하나의 타일(210)은 다수의 동기 도메인, 예를 들어, 타일(210)이 동기 도메인 "A"상에서 작동하는 스포크 0; 타일(210)이 동기 도메인 "B"상에서 작동하는 스포크 1; 타일(210)이 동기 도메인 "A"상에서 작동하는 스포크 2; 및 타일(210)이 동기 도메인 "C"상에서 작동하는 스포크 3에 참여할 수 있다. 타일에 대한 스레드 제어는 도 11을 참조하여 아래에서 설명된다.

[0192] 도 11은 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로(타일)(210)의 스레드 제어 회로(335)(관련 제어 레지스터들(340)과 함께)의 대표적인 실시 예의 상세한 블록도이다. 도 11에 도시된 바와 같이, 제어 레지스터들(340)에는 여러 레지스터들, 즉, TID 풀 레지스터(410), XID 풀 레지스터(415), 일시 중지 테이블(420) 및 완료 테이블(422)이 포함된다. 다양한 실시 예에서, 완료 테이블(422)의 데이터는 서로 같게 일시 중지 테이블(420)에 홀딩될 수 있고, 그 반대로 마찬가지이다. 스레드 제어 회로(335)는 계속 큐(430), 재진입 큐(445), 스레드 제어 멀티플렉서(435), 실행 큐(440), 반복 증분(447), 반복 인덱스(460) 및 루프 반복 카운트(465)를 포함한다. 대안적으로, 계속 큐(430) 및 실행 큐(440)는 제어 레지스터(340)에서 서로 같게 구현될 수 있다.

[0193] 도 12는 제1 및 제2 동기 도메인들(526, 538)을 형성하는 타일들(210) 및 대표적인 비동기 패킷 네트워크 메시징의 도해이다. 비동기 패킷 네트워크(265)가 갖는 한 가지 어려움은 필요한 데이터가 상이한 시간들에 타일들(210)에 도착할 수 있다는 것이며, 이는 시작된 스레드가 고정된 파이프 라인 지연으로 완료될 때까지 실행될 수 있다는 것을 보장하는 것을 어렵게 할 수 있다. 대표적인 실시 예들에서, 동기 도메인을 형성하는 타일(210)은 이를테면 이용 가능한 필요한 데이터, 임의의 필요한 변수들(이들 모두는 비동기 패킷 네트워크(265)를 통해 타일들로 배포됨)을 가짐으로써, 모든 자원이 준비될 때까지 컴퓨터 스레드를 실행하지 않고 그에 따라 임의의 다양한 시간에 지정된 타일(210)에 도달할 수 있다. 또한, 데이터는 시스템 메모리(125)로부터 관독되고 비동기 패킷 네트워크(265)를 통해 전달되었을 수 있고, 그에 따라 또한 임의의 다양한 시간에 지정된 타일(210)에 도달했을 수 있다.

[0194] 고정된 파이프 라인 지연으로 스레드가 완료될 때까지 실행되도록 제공하기 위해, 대표적인 실시 예들은 동기 도메인의 베이스 타일(210)에서 스레드의 TID에 의해 인덱싱되는 완료 테이블(422)(또는 일시 중지 테이블(420))을 제공한다. 완료 테이블(422)(또는 일시 중지 테이블(420))은 스레드의 실행을 개시하기 이전에 수신되어야 하는 종속성 완료 카운트를 유지한다. 완료 테이블(422)(또는 일시 중지 테이블(420))은 리셋시에 0으로 초기화되는 "완료 카운트"라는 명칭의 필드를 포함한다. 카운트 필드를 변경하는 데 두 가지 유형의 AF 메시지

가 사용된다. 첫 번째 메시지 유형은 스레드 시작 또는 계속 메시지이고, 카운트만큼 필드를 증분시켜 동기 도메인에서 스레드가 시작될 수 있기 전에 관찰되어야 하는 종속성 수를 나타낸다. 두 번째 AF 메시지 유형은 완료 메시지이고 카운트 필드를 1씩 감소시켜 완료 메시지가 수신되었음을 나타낸다. 스레드 시작 메시지가 수신되고, 완료 카운트 필드가 0에 도달하면, 스레드가 시작될 준비가 된 것이다.

[0195] 도 12에 도시된 바와 같이, 제1 동기 도메인(526)의 타일(210B)은 비동기 패킷 네트워크(265)를 통해 AF 메모리 로드 메시지(293)를 메모리 인터페이스(215)로 송신했으며, 이는 차례로 요청된 데이터를 획득하기 위한(메시지(297)로 반환됨) 또 다른 메시지(296)를 제1 상호 접속 네트워크(150)를 통해 시스템 메모리(125)에 생성할 것이다. 그러나, 그 데이터는 제2 동기 도메인(538)의 타일(201E)에 의해 이용되고 그것으로 송신된다(메시지(294)). 제1 동기 도메인(526)이 파이프 라인의 그것의 부분을 완료함에 따라, 제1 동기 도메인(526)의 타일들(210C) 중 하나는 제2 동기 도메인(538)의 베이스 타일(210D)에 AF 계속 메시지(291)를 송신한다. 그러한 AF 계속 메시지(291)는 제1 동기 도메인(526)의 스레드의 TID(예를 들어, TID = 1, 도 12의 다른 메시지들에도 포함되어 있음), 및 제2 동기 도메인(538)의 베이스 타일(210D)이 그것이 하나의 완료 메시지를 받을 때까지 제2 동기 도메인(538)상에서 스레드를 시작하기 위해 대기할 것임을 나타내는 완료 카운트 1을 포함한다.

[0196] 따라서, 타일(210)이 도 12의 타일(210E)과 같은 데이터를 수신할 때, 완료 메시지(스레드 ID(TID)를 갖는)를 다시 베이스 타일(210), 여기서는 제2 동기 도메인(538)의 베이스 타일(210D)로 전송함으로써 수신을 확인 응답한다. 베이스 타일(210)에 제공되고 완료 카운트로서 완료 테이블(422)(또는 일시 중지 테이블(420))에 저장되는 구성의 부분으로(초기 구성에서 또는 계속 메시지의 부분으로서), 베이스 타일(210D)은 베이스 타일(210)이 동기 도메인, 이 경우, 제2 동기 도메인(538)의 타일들(210)에 의한 실행을 시작하기 위해 수신해야 하는 그러한 완료 메시지 수를 알고 있다. 완료 메시지들이 베이스 타일(210)에 의해 수신됨에 따라, 그러한 TID를 갖는 특정 스레드에 대해, 일시 중지 테이블의 완료 카운트가 감소되고, 해당 스레드에 대해 모든 필요한 완료 메시지가 수신되었음을 나타내는 0에 도달할 때, 베이스 타일(210)은 스레드의 실행을 시작할 수 있다. 실행을 시작하기 위해, 스레드의 TID는 (타일(210)의 적절한 시간 슬라이스에 대한 적절한 스포크 카운트에서) 실행하도록 선택되는 것으로부터 계속 큐(430)로 전달된다. 스레드 실행 동안 결정되고 동기 메시 통신 네트워크(275)를 통해 동기 도메인의 타일들(210) 간에 전달될 수 있는 데이터에는 완료 메시지가 필요하지 않다는 것에 유의해야 한다.

[0197] 이러한 유형의 스레드 제어에는 몇 가지 이점이 있다. 이러한 스레드 제어는 스레드를 시작하기 이전에 모든 종속성이 완료될 때까지 대기하므로, 시작된 스레드가 고정된 동기 실행 시간을 가질 수 있게 한다. 고정된 실행 시간은 FIFO 대신 파이프 라인 전체에서 레지스터 스테이지들을 사용할 수 있게 한다. 또한, 타일(210)의 하나의 스레드가 실행을 대기하고 있는 동안, 다른 스레드들이 그 타일(210)상에서 실행될 수 있어, 훨씬 더 높은 전체 처리량을 제공하고, 유휴 시간을 최소화하며 사용되지 않는 자원들을 최소화할 수 있다.

[0198] (예를 들어, 컴퓨터 파이버를 형성하는 관련 컴퓨터 스레드들에 대한) 다수의 스레드의 성능과 같이 동기 도메인들에 걸쳐 이어질 때 유사한 제어가 제공된다. 예를 들어, 제1 동기 도메인은 다음 동기 도메인의 베이스 타일(210)에, 비동기 패킷 네트워크(265)를 통해 송신되는 지속 메시지에, 다음 동기 도메인의 베이스 타일(210)에 다음의 스레드 실행을 시작하기 위해 수신해야 할 완료 메시지 수를 알릴 것이다. 또한 예를 들어, 동기 도메인들에 걸쳐 이어지는 반복 루프에 대해, 제1 동기 도메인은 다음 동기 도메인의 베이스 타일(210)에, 비동기 패킷 네트워크(265)를 통해 송신되는 루프 메시지(루프 카운트 및 동일한 TID를 가짐)에, 다음 동기 도메인의 베이스 타일(210)에 다음의 스레드 실행을 시작하기 위해 수신해야 할 완료 메시지 수를 알릴 것이다.

[0199] 또한 이를테면 제1 데이터가 제1 타일(210)로부터 이용 가능한 한편, 제2 데이터가 여전히 제2 타일(210)에 의해 결정될 때, 다양한 지연이 구현될 필요가 있을 수 있으며, 양자는 제3 타일(210)에 의한 다음 계산에 요구될 것임이 언급되어야 한다. 그러한 경우들에서, 제1 데이터를 생성한 제1 타일(210)의 출력 레지스터들(380) 또는 제3 타일(210)의 타일 메모리(325) 중 어느 하나에 지연이 도입될 수 있다. 이러한 지연 메커니즘은 또한 제2 타일(210)을 관통하여 제1 타일(210)로부터 제3 타일(210)로 전달될 수 있는 데이터에도 적용 가능하다.

[0200] 일시 중지 테이블(420)은 필요한 모든 완료 메시지가 수신될 때까지 타일(210)에서 새로운 동기 스레드의 생성을 홀딩 또는 일시 중지하는 데 사용된다. 이전 동기 도메인으로부터의 스레드는 새로운 동기 스레드에 대해 예상되는 완료 메시지 수, 및 모든 완료 메시지가 수신되었을 때 취할 행위들을 포함하는 메시지를 베이스 타일(210)로 전송한다. 행위들에는 콜, 계속 또는 루프가 포함된다. 많은 일시 중지 동작들은 통상적으로 동시에 활성화된다. 특정 일시 중지 동작에 대한 모든 메시지(즉, 일시 중지 및 완료 메시지들의 세트)는 각각의 메시지들 내에 동일한 일시 중지 인덱스를 가질 것이다. 일시 중지 인덱스는 전송 타일(210)로부터의 TID이다. 일시

중지 테이블(420) 엔트리들은 0의 완료 델타 카운트로 비활성화되도록 초기화된다. 일시 중지 메시지를 수신하면 필요한 완료 카운트 수만큼 델타 카운트가 증가되고, 일시 중지 테이블(420) 엔트리가 활성화로 설정된다. 완료 메시지를 수신하면 델타 카운트가 1씩 감소한다. 완료 메시지는 연관된 일시 중지 메시지 이전에 도착하여, 델타 카운트가 음수가 될 수 있다는 점에 유의해야 한다. 일시 중지 테이블(420) 엔트리가 0의 델타 카운트로 활성화될 때, 연관된 활동(예를 들어, 새로운 스레드)이 개시된다(그리고 일시 중지 테이블(420) 엔트리는 비활성화된다).

- [0201] 연속(또는 콜) 큐(430)는 동기 도메인상에서 시작될 준비가 된 스레드들을 홀딩한다. 콜 동작에 대한 모든 완료가 수신될 때 스레드는 지속 큐(430)로 푸시된다. 지속 큐(430) 내 스레드들은 스레드가 동기 도메인상에서 시작될 수 있기 전에 할당될 TID 및/또는 XID를 요구할 수 있다는 점에 유의해야 한다, 예를 들어, 모든 TID가 사용 중인 경우, 지속 큐(430) 내 스레드들은 TID가 해제되고 이용 가능해지면 시작될 수 있다, 즉, TID 및/또는 XID들이 이용 가능해질 때까지 스레드가 대기할 수 있다.
- [0202] 재진입 큐(445)는 동기 도메인상에서 시작될 준비가 된 스레드들을 홀딩하며, 그러한 스레드들의 실행은 지속 큐(430) 내의 것들보다 우선 순위를 갖는다. 스레드는 계속 동작에 대한 모든 완료가 수신될 때 재진입 큐(445)로 푸시되고, 스레드는 이미 TID를 갖는다. 재진입 큐(445) 내 스레드들은 TID의 할당을 요구할 수 없다는 점에 유의해야 한다. 교착 상태 상황을 피하기 위해 별도의 재진입 및 계속(또는 지속) 큐들(445, 430)이 제공된다. 특수한 유형의 계속 동작은 루프이다. 루프 메시지에는 루프 반복 카운트가 포함된다. 카운트는 일시 중지 동작이 완료되면 스레드가 시작될 횟수를 지정하는 데 사용된다.
- [0203] 우선 큐 내 스레드 식별자를 갖는 임의의 스레드가 지속 큐(430) 또는 재진입 큐(445) 내 스레드 식별자를 갖는 임의의 스레드의 실행 이전에 실행되도록, 임의적인 우선 큐(425)가 또한 구현될 수 있다.
- [0204] 반복 인덱스(460) 상태는 루프 동작을 위한 스레드들을 시작할 때 사용된다. 반복 인덱스(460)는 0으로 초기화되고 각 스레드 시작마다 증분된다. 반복 인덱스(460)는 계속 큐(430)로부터의 스레드 정보와 함께 실행 큐(440)로 푸시된다. 반복 인덱스(460)는 동기 도메인의 제1 타일(베이스 타일)(210) 내의 데이터 경로 입력 멀티플렉서(365)에 대한 선택으로서 이용 가능하다.
- [0205] 루프 반복 카운트(465)는 루프 메시지의 부분으로서 수신되고, 일시 중지 테이블(420)에 저장되고, 계속 큐(430)로 푸시된 다음, 루프 동작을 위해 적절한 수의 스레드가 시작된 때를 결정하는 데 사용된다.
- [0206] 실행 큐(440)는 TID들 및/또는 XID들을 할당한 실행 준비가 된 스레드들을 홀딩하고 적절한 스포크 카운트 클러들이 발생했을 때 실행할 수 있다. TID 풀(410)은 동기 도메인상에서 시작될 때 스레드들에 고유한 스레드 식별자들(TID들)을 제공한다. 지속 큐(430) 내 스레드들만이 TID를 획득할 수 있다. XID 풀(415)은 동기 도메인상에서 시작될 때 스레드들에 고유한 전달 식별자들(XID들)을 제공한다. 계속 큐(430)로부터의 스레드들은 XID를 획득할 수 있다. 할당된 XID는 시작된 스레드에 대한 XID\_WR이 된다.
- [0207] 실행될 임의의 주어진 또는 선택된 프로그램에 대해, 임의의 적절한 또는 선택된 프로그래밍 언어로 작성되거나 생성된 해당 프로그램에 대한 코드 또는 명령들은 HTP(300) 및 HTF 회로들(200)에 대한 명령들을 포함하여 시스템(100)에 대해 컴파일되고 그것으로 로딩되며, 이는 선택된 구성을 시스템(100)에 제공하기 위해 호스트 프로세서(110)에 적용 가능할 수 있다. 결과적으로, 하나 이상의 선택된 계산을 위한 다양한 명령들의 세트가 각 타일(210)의 명령 RAM들(315) 및 스포크 RAM들(320)에 로딩되고, 각 타일(210)의 메모리 인터페이스들(215) 및 디스패치 인터페이스(225)에 유지되는 다양한 레지스터 중 임의의 것으로 로딩되어, HTF 회로들(200)에 대한 구성들을 제공하고, 프로그램에 따라, 또한 HTP(300)로도 로딩된다.
- [0208] 커널은 예를 들어 그리고 제한 없이, 하나 이상의 HTF 회로(200)에 의한 성능을 위해 통상적으로 호스트 프로세서(110) 또는 HTP(300)에 의해 생성되는 0 이상의 인수들을 포함하는 작업 기술자 메시지로 시작된다. 인수들은 작업 기술자 AF 메시지 내에 디스패치 인터페이스(225)로 전송된다. 이러한 인수들은 스레드-특정 입력 값들을 제공한다. 각각의 운영 체제("OS")를 사용하는 호스트 프로세서(110) 또는 HTP(300)는 타일 메모리(325) 위치를 초기화하는 커널에 "호스트" 메시지를 전송할 수 있으며, 그러한 호스트 메시지들은 비-스레드 특정 값들을 제공한다. 통상적인 예는 모든 커널 스레드에 의해 사용하는 데이터 구조의 베이스 어드레스를 전송하는 호스트 메시지이다.
- [0209] 커널로 전송되는 호스트 메시지는 해당 커널이 로딩되는 모든 HTF 회로 클러스터(205)로 전송된다. 또한, 호스트 메시지들을 전송하고 커널 디스패치들을 전송하는 순서가 유지된다. 호스트 메시지를 전송하면 메시지를 전송하기 이전에 해당 커널이 기본적으로 유향 상태가 된다. 완료 메시지들은 새로운 동기 스레드들을 시작하기

이전에 타일 메모리(325) 기록들이 완료되었음을 보장한다.

[0210] 비동기 패킷 네트워크(265)를 통한 제어 메시징은 다음과 같다:

[0211] (1) 디스패치 인터페이스(225)는 호스트 메시지를 수신하고 AF 데이터 메시지를 목적지 타일(210)로 전송한다. 목적지 타일(210)은 선택된 메모리에 AF 데이터 메시지의 데이터를 기록한다.

[0212] (2) 목적지 타일(210)은 타일 기록이 완료됨을 확인 응답하는 AF 완료 메시지를 디스패치 인터페이스(225)로 전송한다.

[0213] (3) 디스패치 인터페이스(225)는 모든 메시지 기록이 확인 응답될 때까지 모든 새로운 커널 스레드 시작을 홀딩한다. 확인 응답되면, 디스패치 인터페이스(225)는 스레드를 시작하기 위해 AF 콜 메시지를 동기 도메인의 베이스 타일로 송신한다.

[0214] 디스패치 인터페이스(225)는 다음을 포함하여 HTF 회로 클러스터(205)를 관리하는 역할을 한다: (1) 프로세스에 의한 사용을 위해 HTF 회로 클러스터(205)를 준비하기 위해 시스템(100) 소프트웨어와의 상호 작용; (2) HTF 회로 클러스터(205)를 하나 이상의 커널 구성으로 로딩하는 것을 포함하여 HTF 회로 클러스터(205)의 타일들(210)로 작업을 디스패칭하는 것; (3) 중단점 및 예외들을 위해 HTF 회로 클러스터(205)의 컨텍스트들을 메모리(125)에 저장 및 복원. 상술된 바와 같이, 디스패치 인터페이스(225)의 레지스터들(475)은 대표적인 실시 예들에서 이용되는 임의의 메시징을 추적하는 것과 같이, 다양한 타일(210) 중 임의의 것으로 디스패칭되고 그것으로부터 수신된 것을 추적하기 위해 매우 다양한 테이블을 포함할 수 있다. 이러한 동작들을 수행하는 데 이용되는 디스패치 인터페이스(225) 원시 동작들이 표 4에 나열된다.

표 4

[0215]

원시 동작	위치	개시 주체	동작 설명
HTF 클러스터 로드 커널 구성	HTF 디스패치 인터페이스	애플리케이션	HTF 회로 클러스터(205)는 필요한 커널 구성이 현재 로딩된 구성과 매칭되는지 결정하기 위해 각각의 수신된 작업 기술자를 체크한다. 작업 기술자의 커널 구성이 현재 로딩된 구성과 매칭되지 않으면, HTF 회로 클러스터(205)는 모든 이전 작업이 완료될 때까지 대기하고 새로운 커널 구성을 로딩한다. 각 작업 기술자는 필수 커널 구성에 대한 가상 어드레스를 갖는다.
HTF 클러스터 리셋	HTF 디스패치 인터페이스	OS	새로운 커널 구성 또는 커널 컨텍스트가 로딩될 수 있게 하도록 HTF 회로 클러스터(205) 내 모든 상태를 리셋한다.
HTF 클러스터 저장 컨텍스트	HTF 디스패치 인터페이스	OS	HTF 회로 클러스터(205)는 예외 또는 중단점으로 인해 실행을 일시 중지할 수 있다. HTF 회로 클러스터(205)는 이벤트를 알리기 위해 인터럽트를 OS에 전송한다. OS는 디버거 액세스를 위해 프로세스 컨텍스트가 메모리에 저장되어야 하는지를 결정한다. 프로세스 컨텍스트가 필요하다면, OS는 HTF 회로 클러스터(205)의 디스패치 인터페이스(225)와 상호 작용함으로써 동작을 개시한다.
HTF 클러스터 로드 컨텍스트	HTF 디스패치 인터페이스	OS	HTF 회로 클러스터(205)에 대한 컨텍스트는 실행을 재개하기 위한 준비로 메모리로부터 로딩될 수 있다. OS는 HTF 회로 클러스터(205)의 디스패치 인터페이스(225)와 상호 작용함으로써 동작을 개시한다.
HTF 클러스터 일시 중지	HTF 디스패치 인터페이스	OS	OS는 자체의 프로세스가 중지되어야 할 때 실행 중인 HTF 회로 클러스터들(205)상에서 실행을 일시 중지할 필요가 있을 수 있다. 프로세서 또는 상이한 HTF 회로 클러스터(205)에 의해 예외 또는 중단점이 발생하거나, 프로세스가 Linux 신호를 수신한 경우 프로세스가 중지되어야 할 수 있다. OS는 디스패치 인터페이스(225)와 상호 작용함으로써 일시 중지를 개시한다.

HTF 클러스터 재개	HTF 디스패치 인터페이스	OS	일시 중지된 HTF 회로 클러스터(205)의 실행은 HTF 회로 클러스터(205)의 일시 중지 신호를 제거함으로써 재개될 수 있다. OS는 HTF 회로 클러스터(205)의 디스패치 인터페이스(225)와 상호 작용함으로써 재개를 개시한다.
HTF 클러스터 유휴	HTF 디스패치 인터페이스	OS	OS는 HTF 회로 클러스터(205)가 유휴 상태이고 새로운 동작을 받아 들일 준비가 되는 때를 결정할 필요가 있을 수 있다. 디스패치 인터페이스(225)는 다양한 명령을 수행하는 다수의 상태 기계를 갖는다. 이러한 명령들은 컨텍스트 로드, 컨텍스트 저장, 일시 중지 및 구성 로드를 포함한다. OS는 명령을 발행하기 이전에 HTF 회로 클러스터(205)가 유휴 상태를 보장해야 한다.

[0216] 도 16 및 17은 시스템(100) 내의 메시징 및 스레드 제어의 예를 제공하며, 동기 메시 통신 네트워크(275) 및 비동기 패킷 네트워크(265)가 함께 작동하여 간단한 커널을 실행하는 방법을 보여주기 위해 제공된 예시적인 계산은 여기서, 간단한 식  $R = *A + B$ 를 푼다. 그러한 메시징을 보여주기 위해, 계산은 두 개의 상이한 동기 도메인(526 및 538)에 걸쳐 분할되었다. 변수 B는 호스트 메시지로써 모든 HTF 회로 클러스터(205)에 전달되고, 어드레스 A는 작업 기술자 패킷에 콜에 대한 인수로서 전달된다. 결과 R은 제1 상호 접속 네트워크(150)를 통해 반환 데이터 패킷에 다시 전달된다. 이 예는 거의 컴퓨터를 하지 않으므로 수행되는 컴퓨터당 메시지 수가 매우 높다. HTF 회로들(200)은 컴퓨터당 메시지 수가 낮도록 루프 내에서 상당한 계산이 수행될 때 훨씬 더 높은 성능을 갖는다.

[0217] 도 16은 동기 도메인들을 형성하는 대표적인 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들)(210) 및 HTF 회로 클러스터(205)에 의한 계산의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징의 도해이다. 도 17은 HTF 회로 클러스터(205)에 의한 도 16의 계산의 수행을 위한 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들)에 의한 대표적인 비동기 패킷 네트워크 메시징 및 실행의 흐름도이다. 시작하기 위해, 호스트 프로세서(110)는 노드 내의 모든 HTF 회로 클러스터(205)에 메시지(504)를 전송한다(단계 506). 메시지는 변수 B의 값이다. 메시지는 프로세스와 연관된 HIF(115)(도 1 및 2에 도시됨)의 디스패치 큐(105)에 기록되는 하나의 데이터 패킷(통상적으로 작업 기술자 패킷으로 지칭됨)에 포함된다. HIF(115)는 디스패치 큐(105)로부터 메시지를 관독하고 프로세스에 할당된 각각의 HTF 회로 클러스터(205)에 패킷의 사본을 전송한다. 할당된 HTF 회로 클러스터(205)의 디스패치 인터페이스(225)는 패킷을 수신한다. 또한 HIF(115)는 모든 HTP(300) 및 HTF(200) 자원에 걸쳐 다양한 로드 밸런싱 기능을 수행한다는 점에 유의해야 한다.

[0218] 호스트 프로세서(110)는 프로세스에 할당된 하나의 HTF 회로 클러스터(205)에 콜 메시지(508)를 전송한다(단계 510). 호스트 프로세서(110)는 수동으로 특정 HTF 회로 클러스터(205)를 타겟으로 하여 커널을 실행하거나, 또는 HTF 회로 클러스터(205)가 자동으로 선택될 수 있게 할 수 있다. 호스트 프로세서(110)는 프로세스와 연관된 디스패치 큐에 콜 파라미터들을 기록한다. 콜 파라미터들은 커널 어드레스, 시작 명령 및 하나의 인수(변수 A의 어드레스)를 포함한다. 호스트 인터페이스(HIF)(115)는 큐잉된 메시지를 관독하고 제1 상호 접속 네트워크(150)상의 데이터 패킷으로서 메시지를 할당된 HTF 회로 클러스터(205), 통상적으로 최소 로드를 갖는 HTF 회로 클러스터(205)에 포워딩한다.

[0219] 디스패치 인터페이스(225)는 호스트 메시지(변수 B의 값)를 수신하고, HTF 회로 클러스터(205)에 대한 모든 이전 콜이 완료될 때까지 대기한 다음, 비동기 패킷 네트워크(265)를 통해 AF 메시지(512)를 사용하여 첫 번째 선택된 목적지 타일(210H)로 값을 전송한다(512). 디스패치 인터페이스(225)는 목적지 타일(210H), 해당 타일(210H)에 대한 타일 메모리(325) 및 메모리 영역(RAM(405) 내)을 나타내는 각각의 가능한 호스트 메시지에 대해 레지스터들(475)에 저장된 정보 테이블을 갖는다. 타일(210H)은 메시지 정보를 사용하여 타일(210H)의 메모리(325)에 값을 기록하고, 값이 타일 메모리에 기록되면, 기록 완료 AF 메시지(516)가 비동기 패킷 네트워크(265)를 통해 다시 디스패치 인터페이스(225)로 전송된다(단계 518).

[0220] 디스패치 인터페이스(225)는 모든 메시지 완료 메시지(이 경우에는 단지 하나의 메시지)가 도착하기를 대기한다. 모든 완료 메시지가 도착하면, 디스패치 인터페이스(225)는 AF 메시지(520)의 콜 인수(변수 A의 어드레스)를 타일 메모리(325)에 기록될 값에 대한 제2 선택된 목적지 타일(210B)로 전송한다(단계 522). 디스패치 인터페이스(225)는 목적지 타일(210B), 해당 타일(210B)에 대한 타일 메모리(325) 및 메모리 영역(RAM(405))

내)을 나타내는 레지스터들(475)에 저장된 콜 인수 테이블을 갖는다.

- [0221] 그 다음 디스패치 인터페이스(225)는 AF 콜 메시지(524)를 제1 동기 도메인(526)의 베이스 타일(210A)로 전송한다(단계 528). AF 콜 메시지는 콜이 동기 타일(210) 파이프 라인을 통해 실행을 시작할 수 있기 전에 하나의 완료 메시지가 수신되어야 함을 나타낸다. 필요한 완료 메시지가 도착하지 않으면 콜은 일시 중지된다.
- [0222] 값이 해당 타일(210B)에 대한 타일 메모리(325)에 기록되면, 기록 완료 메시지(530)가 타일(210B)에 의해 비동기 패킷 네트워크(265)를 통해 제1 동기 도메인(526)의 베이스 타일(210A)로 전송된다(단계 532).
- [0223] 베이스 타일(210A)은 콜 메시지(524) 및 필요한 완료 메시지(530) 양자를 수신했고, 이제 동기 도메인(526)(타일 파이프 라인)상에서 실행을 개시할 준비가 되었다. 베이스 타일(210A)은 동기 메시 통신 네트워크(275)를 통해 타일(210B)에 개시 명령 및 유효한 신호(534)를 제공함으로써 실행을 개시한다(단계 536). 베이스 타일(210A)은 XID 풀(415)로부터 제1 동기 도메인(526)에서 사용하기 위한 XID 값을 할당한다. XID 풀(415)이 비어 있다면, 베이스 타일(210A)은 동기 파이프 라인을 시작하기 위해 XID가 이용 가능해질 때까지 대기해야 한다.
- [0224] 실행이 진행됨에 따라, 제1 동기 도메인(526) 내의 타일(210B) 또는 다른 타일(210E)은 AF 계속 메시지(540)를 제2 동기 도메인(538)의 베이스 타일(210C)에 전송한다(단계 542). 계속 메시지는 제2 동기 도메인(538)이 실행을 개시할 수 있기 전에 도착해야 하는 필요한 완료 메시지 수(이 경우 하나의 완료 메시지)를 포함한다. 계속 메시지에는 또한 전달 ID(XID)가 포함된다. XID는 하나의 동기 도메인(526)에서 기록 인덱스로서 사용된 다음, 다음 동기 도메인(538)에서 판독 인덱스로서 사용된다. XID는 하나의 동기 도메인으로부터 다음 도메인으로 공통 타일 메모리 인덱스를 제공한다.
- [0225] 제1 동기 도메인(526) 내의 타일(210B) 또는 다른 타일(210F)은 AF 메모리 로드 메시지(544)를 HTF 회로 클러스터(205)의 메모리 인터페이스(215)에 전송한다(단계 546). 메시지는 요청 ID, 가상 어드레스, 및 목적지 타일(210G) 메모리(325)에 로드 데이터를 기록하기 위한 인덱스로서 사용될 XID를 포함한다.
- [0226] 메모리 인터페이스(215)는 AF 로드 메시지를 수신하고 가상 어드레스를 노드 로컬 물리 어드레스 또는 원격 가상 어드레스로 변환한다. 메모리 인터페이스(215)는 AF 메시지의 요청 ID를 사용하여 메모리 요청에 대한 파라미터들을 포함하는 레지스터들(485)에 저장된 요청 테이블에 인덱싱한다. 메모리 인터페이스(215)는 요청 테이블로부터 변환된 어드레스 및 크기 정보를 갖는 제1 상호 접속 네트워크(150)에 대한 로드 메모리 요청 패킷(548)을 발행한다(단계 550).
- [0227] 이어서 메모리 인터페이스(215)는 로드 데이터(변수 A에 대한 값)을 갖는 메모리 응답 패킷(552)을 제1 상호 접속 네트워크(150)를 통해 수신한다(단계 554). 메모리 인터페이스(215)는 AF 메시지(556)를 제2 동기 도메인(538) 내의 타일(210G)에 전송한다(단계 558). AF 메시지는 변수 A에 대한 값을 포함하고 값은 레지스터들(485)에 저장된 요청 테이블로부터의 파라미터를 사용하여 타일 메모리에 기록된다.
- [0228] 값이 타일 메모리에 기록되면, AF 기록 완료 메시지(560)가 비동기 패킷 네트워크(265)를 통해 제2 동기 도메인(538)의 베이스 타일(210C)로 전송된다(단계 562).
- [0229] 제2 동기 도메인(538)의 베이스 타일(210C)은 계속 메시지(540) 및 필요한 완료 메시지(560) 양자를 수신하고 제2 동기 도메인(538)(타일 파이프 라인)상에서 실행을 개시할 준비가 된다. 베이스 타일(210C)은 타일(210H)과 같은, 제2 동기 도메인(538)의 타일(210)에 초기 명령 및 유효한 신호(564)를 제공함으로써 실행을 개시한다(단계 566). 베이스 타일(210C)은 또한 XID 풀로부터 제2 동기 도메인(538)에서 사용하기 위한 XID 값을 할당한다.
- [0230] 제2 동기 도메인 내의 타일(210H)은 호스트 메시지로부터 전달된 B 값 및 시스템 메모리(125)로부터 판독된 A 값의 추가 연산을 수행한다(단계 568). 결과 값은 식의 R 값이다.
- [0231] 제2 동기 도메인 내의 타일(210J)은 R 값을 포함하는 AF 메시지(570)를 디스패치 인터페이스(225)로 전송한다(단계 572). AF 메시지는 베이스 타일(210A)로부터 할당된 XID 값을 포함한다. XID 값은 값들이 판독될 때까지 반환 파라미터들 및 제1 상호 접속 네트워크(150)를 통한 송신을 위해 생성된 반환 메시지를 홀딩하는 레지스터들(475)에 저장된 테이블에 대한 디스패치 인터페이스(225) 내의 인덱스로서 사용된다.
- [0232] 제2 동기 도메인(타일(210K))으로부터의 AF 메시지(574)는 XID 풀로 반환되도록 제1 동기 도메인에 할당된 XID 값을 베이스 타일(210A)로 다시 전송한다(단계 576). 디스패치 인터페이스(225)로부터의 제1 상호 접속 네트워크(150) 메시지(578)는 HIF(115)로 전송된다(단계 580). HIF는 반환 작업 기술자를 디스패치 반환 큐에 기록한다. 제1 상호 접속 네트워크(150)가 반환 패킷을 전송하면, XID 값은 XID 풀로 리턴되도록 디스패치 인터페이스

(225)에 의해 AF 메시지(582)에 제2 동기 도메인(538)의 베이스 타일(210C)로 전송된다(단계 584).

- [0233] 도 16 및 도 17의 이러한 예에서, 스레드 제어를 위해 이용될 수 있는 매우 다양한 AF 메시지를 보여주기 위해 많은 타일(210)이 이용되었다는 점에 유의해야 한다. 실제로는, 특히 그러한 간단한 계산을 위해, 하나의 동기 도메인 내에서 계산을 완전히 수행하기 위해 훨씬 더 적은 타일(210)이 이용될 수 있다.
- [0234] 또 다른 메시징 예는 또한 비동기 패킷 네트워크(265)상에서 AF 완료 및 계속 메시지들을 사용하는, 도 18 및 도 19에서의 다수의 동기 도메인에 걸친 스레드 제어에 대해 제공된다. 도 18은 동기 도메인들을 형성하는 대표적인 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들) 및 하이브리드 스레딩 패브릭 회로 클러스터에 의한 계산의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징의 도해이다. 도 19는 하이브리드 스레딩 패브릭 회로 클러스터에 의한 도 18의 계산의 수행을 위한 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들)에 의한 대표적인 비동기 패킷 네트워크 메시징 및 실행의 흐름도이다.
- [0235] 이러한 예에서, 디스패치 인터페이스(225)는 제1 동기 도메인(526)의 베이스 타일(210A)로 메시지를 전송한다. 메시지는 제1 동기 도메인(526)상에서 스레드를 시작한다. 스레드는 스레드 계속 메시지를 제2 동기 도메인(538)에 전송한다. 계속 메시지는 지정된 수의 완료 메시지가 수신되었을 때 제2 동기 도메인(538)상에서 스레드가 시작될 것임을 나타낸다. 제1 동기 도메인(526)은 완료 메시지를 제2 동기 도메인(538)에 전송하여 일시 중지를 완료하고 동기 제2 스레드를 시작한다. 제2 스레드는 디스패치 인터페이스(225)에 제2 동기 스레드가 완료되어 디스패칭된 커널을 완료했음을 나타내는 완료 메시지를 다시 전송한다. TID 및 XID 식별자들을 해제하는 추가 메시지들이 도 18에 도시되어 있다.
- [0236] 디스패치 인터페이스(225)는 작업 기술자 패킷(602)을 수신했고, 올바른 커널 구성이 로딩되었는지를 확실하게 했으며, XID 및 TID 폴들이 비어 있지 않다고 결정하여, 디스패치 인터페이스(225) 내의 레지스터들(475)에 저장된 TID 및 XID 폴들로부터 새로운 작업 스레드에 대한 XID 및 TID 값들을 얻는다(단계 604). 디스패치 인터페이스(225)는 AF 콜 메시지(606)(할당된 XID 및 TID 값들, 예를 들어, XID = 3 그리고 (제1 유형) TID = 11을 갖는)를 제1 동기 도메인(526)의 베이스 타일(210A)로 전송함으로써 커널 실행을 시작한다(단계 608). 베이스 타일(210A)은 AF 콜 메시지(606)를 수신하고, TID 및 XID 폴들(410, 415)이 비어 있지 않다고 결정하며, TID 및 XID 값들(예를 들어, XID\_WR = 7 그리고 (제2 유형) TID = 1)을 할당하고, 스포크 램(320)은 타일 데이터 경로에 대한 입력으로 베이스 타일을 선택하고 있으며(단계 610), 그에 따라 그것은 (예를 들어, 조건부 실행에 관해 아래에서 더 상세히 논의될 바와 같이) 이전 타일(210)에 의해 제공되었을 수 있는 명령 인덱스로부터의 명령을 잠재적으로 실행하기 보다, 스포크 램(320)에 홀딩된 명령 인덱스에 의해 지정되는 첫 번째 지정된 명령으로 실행을 시작하게 된다.
- [0237] 베이스 타일(210A)은 제1 동기 도메인(526)을 통해 할당된 TID 값(예를 들어, (제2 유형) TID = 1), AF 콜 메시지(606)로부터 값을 할당받은 XID\_RD(예를 들어, XID\_RD = 3), XID 폴로부터 값을 할당받은 XID\_WR(예를 들어, XID\_WR = 7), 및 AF 콜 메시지(606)로부터의 TID(예를 들어, (제1 유형) TID = 11)를 갖는 제1 스레드(612)를 시작한다(단계 614).
- [0238] 계산이 제1 동기 도메인(526)에서 진행됨에 따라, 제1 동기 도메인(526) 내의 다른 타일(210B)은 AF 계속 메시지(616)를 제2 동기 도메인(538)의 베이스 타일(210D)로 전송한다(단계 618). AF 계속 메시지(616)는 적절한 수의 완료 메시지가 도착했을 때 제2 동기 도메인(538)상에서 제2 스레드를 시작하는 데 필요한 정보를 제공한다. AF 계속 메시지(616)는 필요한 완료 메시지 수를 지정하는 값을 갖는 완료 카운트 필드를 포함한다. 제1 동기 도메인(526) 내의 타일들(210C) 중 하나는 또한 디스패치 인터페이스(225)에 자유 XID(예를 들어, XID = 3) 메시지(641)를 송신한다.
- [0239] AF 계속 메시지(616)는 목적지 베이스 타일(210D)상의 일시 중지 테이블(420)에 대한 인덱스로서 TID 또는 XID\_WR 값 중 어느 하나를 포함할 수 있다. 일시 중지 테이블은 수신된 완료 메시지들을 모으고 필요한 수가 도착하고 새로운 스레드가 시작될 수 있는 때를 결정한다(단계 620). AF 계속 메시지(616)를 전송하는 타일(210B)은 선택된 TID 또는 XID\_WR 값을 PID(일시 중지 테이블 인덱스)로서 전송하고 동기 도메인의 다운 스트림 TID 값을 선택한 값으로 변경한다(예를 들어, (제1 유형) TID = 11, (제2 유형) TID = 1). 이러한 새로운 TID 값은 베이스 타일(210D)의 일시 중지 테이블(420)에 대한 인덱스로서 사용될 모든 AF 완료 메시지에 전달된다.
- [0240] TID 값(예를 들어, (제1 유형) TID = 11)을 갖는 AF 완료 메시지(622)가 제2 동기 도메인(538)의 베이스 타일(210D)로 전송된다(단계 624). AF 완료 메시지(622)는 베이스 타일(210D)의 일시 중지 테이블(420)의 델타 카운트 필드의 엔트리를 감소시킨다. AF 완료 메시지(622) 및 AF 계속 메시지(616)는 임의의 순서로 도착할 수

있다. 도착할 마지막 메시지는 AF 계속 메시지(616)가 도착했고 일시 중지 테이블(420)의 델타 카운트 필드가 0에 도달했음을 관찰할 것이다. 이러한 조건은 일시 중지가 완료되었고 제2 동기 스레드(626)가 시작될 수 있음을 나타낸다. 베이스 타일(210D)은 또한 일시 중지 동작이 완료되었음을 결정하거나 관찰하고, XID 식별자 풀이 비어 있지 않다고 결정하며, XID를 할당하고(예를 들어, XID = 5), 스포크 램은 타일 데이터 경로에 대한 입력으로서 베이스 타일을 선택하고 있다(단계 628).

[0241] 그 다음 베이스 타일(210D)은 AF 계속 메시지(616)로부터 획득된 값(예를 들어, (제1 유형) TID = 11, (제2 유형) TID = 1, XID\_RD = 7)이 할당된 TID 및 XID\_RD를 갖는 제2 동기 스레드(626)를 제2 동기 도메인(538)을 통해 시작한다(단계 630). XID\_WR에는 단계 628에서 XID 풀로부터 얻어진 값이 할당된다(예를 들어, XID\_WR = 5).

[0242] 제2 동기 스레드(626)의 계산들이 완료되었을 때, 여러 하우스 키퍼링 메시지가 제2 동기 도메인(538)의 다양한 타일(210)에 의해 전송된다. AF 자유 TID 메시지(632)가 제1 동기 도메인(526)의 베이스 타일(210A)로 전송되고(단계 634), 수신 베이스 타일(210A)은 TID 값을 TID 풀(410)에 추가하여(단계 636), 다시 한번 사용 가능하게 된다. AF 자유 XID 메시지(638)는 제1 동기 도메인(526)의 베이스 타일(210A)로 전송되고(단계 640), 수신 베이스 타일(210)은 XID 값을 XID 풀(415)에 추가하여(단계 642), 또한 다시 한번 사용 가능하게 된다. 제2 동기 스레드(626)가 완료되었음을 나타내는 AF 완료 메시지(644)가 디스패치 인터페이스(225)로 전송된다(단계 646). 디스패치 인터페이스(225)는 예상되는 완료 메시지들의 카운트를 갖는다. AF 완료 메시지(644)는 디스패치 인터페이스에 대한 제2 동기 도메인(538)의 XID\_WR 값 및 TID 값((제1 유형) TID = 11)을 포함한다. 그 다음 디스패치 인터페이스(225)는 AF 자유 XID 메시지(648)를 제2 동기 도메인(538)의 기본타일(210D)로 전송한다(단계 650). 수신 베이스 타일(210D)은 XID 값을 XID 풀(415)에 추가하여(단계 652), 다시 한번 사용 가능하게 된다.

[0243] 데이터 전달 동작을 사용하여 하나의 동기 도메인으로부터 다음 도메인으로 데이터를 전달한다. 통상적으로, 데이터 전달은 메모리(125)로부터 데이터를 획득하는 로드 동작과 함께 사용된다. 로드 데이터가 제2 동기 도메인(538)에 도착하면 제1 동기 도메인(526)으로부터 계산된 데이터가 제2 동기 도메인(538)에 필요하게 된다. 이 경우, 모든 로드 및 데이터 전달 동작들로부터의 총 완료 메시지 카운트를 포함하는 하나의 일시 중지가 제1 동기 도메인(526)으로부터 제2 동기 도메인(538)으로 전송된다.

[0244] 그 다음 동기 도메인들 간의 데이터 전달 동작은 단계 624의 변형을 이용한다. 단계 624에서 AF 완료 메시지(622)를 전송하는 대신, 제1 동기 도메인(526)은 데이터를 갖는 AF 데이터 메시지를 제2 동기 도메인(538)에 전송한다. 제2 동기 도메인(538) 내의 목적지 타일(210)은 AF 데이터 메시지 내의 데이터를 선택된 타일 메모리(325)에 기록한다. 그 다음 AF 데이터 메시지를 수신하는 타일(210)은 제2 동기 도메인(538)의 베이스 타일(210)에 AF 완료 메시지를 전송한다. 제2 동기 도메인(538)의 베이스 타일(210)은 로드 데이터가 제2 동기 도메인(538)에 도착하면 제2 동기 도메인(538)상에서 제2 스레드를 시작할 수 있다.

[0245] 동기 도메인들에 걸쳐 반복되는 스레드 루프에 대한 제어는 유사한 제어 메시징 기법을 이용한다. 루프 메시지 흐름은 하나의 루프 메시지로부터 다수의 동기 도메인 시작을 가능하게 한다. 시작된 동기 스레드들 각각은 반복 인덱스에 액세스할 수 있다. 도 20은 동기 도메인들을 형성하는 대표적인 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들) 및 하이브리드 스레딩 패브릭 회로 클러스터에 의한 계산의 루프의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징의 도해이다. 도 21은 하이브리드 스레딩 패브릭 회로 클러스터에 의한 도 20의 계산의 루프의 수행을 위한 하이브리드 스레딩 패브릭 구성 가능한 컴퓨팅 회로들(타일들)에 의한 대표적인 비동기 패킷 네트워크 메시징 및 실행의 흐름도이다.

[0246] 도 20은 세 개의 동기 도메인, 제1 동기 도메인(526), 제2 동기 도메인(538) 및 제3 동기 도메인(654)을 도시한다. 제1 동기 도메인(526)은 프리-루프 설정에 사용되고, 제2 동기 도메인(538)은 스레드 수에 대한 반복 카운트(IterCnt)로 시작되며, 마지막 제3 동기 도메인(654)은 포스트-루프이다. 루프들은 또한 추가 인덱싱 계층들을 사용하여 중첩될 수도 있으며, 아래에서 더 상세히 논의된다.

[0247] 다시 도 11을 참조하면, 제어 레지스터들(340)은 완료 테이블(422)(또는 일시 중지 테이블(420))을 포함한다. 루프들의 경우, 두 가지 유형의 완료 정보가 완료 테이블(422)에 유지되며, 하나는 상술된 바와 같이, 스레드가 시작될 수 있기 전에 도착해야 하는 완료 메시지 수와 관련된 제1 완료 카운트이고, 다른 하나는 시작되고 완료된 루프 스레드 수를 추적하기 위한 제2 루프 또는 반복(완료) 카운트이다. 루프는 루프 카운트(및 아래에서 논의될 다양한 TID)를 포함하는 AF 루프 메시지를 동기 도메인의 베이스 타일(210)로 전송함으로써 시작된다. 루프 카운트는 완료 테이블(422)(또는 일시 중지 테이블(420))에 저장되고, 동기 도메인상에서 새로운 스레드가 시작되는 횟수를 결정하는 데 사용된다. 일 실시 예에서, 각 스레드는 TID 풀(410)로부터 얻어지는 새로운 TID로 시작된다. 각 활성 스레드는 예를 들어, 스레드 전용 변수들을 허용하는 고유한 TID를 갖는다. 중첩된 루프

들의 스레드들은 그 자체의 TID의 데이터 또는 변수들과, 외부 루프의 TID들에 액세스될 수 있다. 아래에서 논의되는 제2 실시 예에서, TID들은 루프의 연속적인 스레드들에 의해 재사용된다.

[0248] TID들은 스레드가 종료될 때 동기 도메인 내의 타일로부터 전송되는 AF 메시지에 의해 TID 풀(410)로 반환되며, 이는 AF 완료 메시지가거나 제2 실시 예의 경우 AF 재진입 메시지가 될 수 있다. 이는 또한 베이스 타일(210)에 대한 자유 TID 메시지에 의해 실현될 수 있다. TID를 풀에 반환하거나 TID를 재사용하는 AF 메시지는 또한 루프 베이스 타일(210)에 의해 완료 테이블(422)(또는 일시 중지 테이블(420))의 루프 카운트에 활성 루프 스레드 수의 카운트를 유지하는 데 사용된다. 활성 루프 스레드 수가 0에 도달할 때, 루프가 완료된다. 루프 카운트가 0이 되어 루프 완료가 검출될 때, 완료를 알리는 AF 완료 메시지가 포스트-루프 동기 도메인으로 전송된다. 이러한 메커니즘은 중첩 루프들에 대해 최소(0이 아닌 경우) 유휴 사이클들을 제공하므로 성능이 향상된다.

[0249] 도 20 및 도 21을 참조하면, 제1 동기 도메인(526)(타일(210B)로 도시되어 있지만, 제1 동기 도메인(526)의 임의의 다른 타일로부터 있을 수 있음)은 AF 계속 메시지(656)를 제3 포스트-루프 동기 도메인(654)의 베이스 타일(210D)로 전송하여(단계 658), 루프 완료 메시지(제2 동기 도메인(538)으로부터 올)를 대기한다. 타일(210B)로 도시된 제1 동기 도메인(526) 내의 타일들 중 하나는 반복(루프) 카운트를 갖는 AF 루프 메시지(660)를 루프 도메인(이는 제2 동기 도메인(538))의 베이스 타일(210C)로 전송한다(단계 664). 베이스 타일(210C)은 제2 동기 도메인(538)상에서 루프(IterCnt) 스레드들(662, 예를 들어, 662<sub>0</sub>, 662<sub>1</sub> 내지 662<sub>N-1</sub>, 여기서 "N"은 반복 카운트(IterCnt))를 시작한다. 각 스레드(662)는 동일한 TID 및 XID\_RD 식별자들을 갖는다. XID\_WR 식별자는 인에이블되는 경우 루프 베이스 타일(210C)에 의해 할당된다. 반복 인덱스(즉, 0으로부터 IterCnt-1(N-1)로 순서화됨)는 루프 도메인의 베이스 타일(210C)에 데이터 경로 멀티플렉서 선택으로 액세스 가능하다.

[0250] 그 다음 루프 도메인의 각 반복은 AF 완료 메시지(666)를 제2 동기(루프) 도메인(538)의 베이스 타일(210C)로 다시 전송한다(단계 668). 도 20에 도시된 제2 동기 도메인(538)은 실제로 여러 동기 도메인일 수 있음을 유의해야 한다. 다수의 동기 도메인이 루프를 형성하는 경우, 루프의 마지막 동기 도메인의 스레드들은 AF 완료 메시지(666)를 전송해야 하므로, 포스트-루프 제3 동기 도메인(654)은 모든 루프 동작이 완료될 때까지 적절히 대기한다. 제2 동기(루프) 도메인(538)의 베이스 타일(210C)이 모든 반복 AF 완료 메시지(666)를 수신하면, 그것은 루프 AF 완료 메시지(또는 AF 계속 메시지)(670)를 제3(포스트-루프) 동기 도메인(654)의 베이스 타일(210D)로 전송한다.

[0251] 중첩 및 이중 중첩 루프를 포함하는 루프의 경우, 예를 들어 그리고 제한 없이, 재진입 큐(445) 및 가장 바깥쪽 루프에 대해 TID<sub>2</sub>, 중간 루프에 대해 TID<sub>1</sub>, 그리고 가장 안쪽 루프에 대해 TID<sub>0</sub>과 같은 추가 서브-TID들을 포함하여, 이를테면 유휴 시간을 최소화하기 위해 여러 추가의 그리고 신규한 특징들이 이용된다. 루프에서 실행 중인 각 스레드는 또한 50번의 반복이 있을 외부 루프에 대해 TID<sub>2</sub>들 0 - 49와 같이 고유한 TID를 가지며, 이는 또한 예를 들어 그리고 제한 없이, 각 반복이 실행을 완료할 때 대응하는 완료 메시지에 이용된다.

[0252] 다시 도 11을 참조하면, 효율적인 루프를 지원하고 유휴 시간을 최소화하기 위해 여겨 새로운 메커니즘이 제공된다. 예를 들어, 데이터-중속 종료 조건들(예를 들어, "반복(while)" 루프들)이 있는 루프들은 루프가 실행될 때 종료 조건이 계산되어야 한다. 또한 루프의 제어 및 실행에 대해, TID 풀(410)로부터 모든 TID가 할당되었지만, 실행을 위한 큐의 헤드에 있는 스레드가 이용 가능한 TID들의 부족으로 인해 실행할 수 없어, 다른 루프 스레드들을 차단하는 새로운 루프인 경우 잠재적인 교착 상태 문제가 발생할 수 있으며 이는 그것들의 할당된 TID들을 완료하고 해제할 수 없다. 따라서, 대표적인 실시 예들에서, 제어 레지스터들(340)은 새로운 루프들을 개시하기 위한 제1 큐(비루프 스레드들에도 이용되는 지속 큐(430)), 및 루프 지속을 위한 제2 별개의 큐(재진입 큐(445))를 이용하여, 실행할 준비가 된 스레드들을 위한 두 개의 별개의 큐를 포함한다. 지속 큐(430)는 전술된 바와 같이, 스레드를 시작하기 위해 TID 풀(410)로부터 TID를 할당한다. 재진입 큐(445)는 루프 스레드의 각 반복이 이전에 할당된 TID들을 갖는 AF 재진입 메시지를 실행하고 송신함에 따라, 이전에 할당된 TID를 사용한다. 재진입 큐(445) 내 임의의 스레드(TID)는 다른 큐들(지속 큐(430))에 있을 수 있는 스레드들(TID들)보다 앞서 실행 큐(440)로 이동될 것이다. 결과적으로, 루프가 시작되면, 루프 반복이 매우 빠르게 발생하며, 루프의 각 다음 스레드는 별개의 재진입 큐(445)를 사용하여 빠르게 시작되고, 나아가 교착 상태 문제가 발생할 가능성이 없다. 또한, 재진입 큐(445)는 데이터-중속 종료 조건들을 갖는 루프들에 대해 매우 중요한 이러한 빠른 실행을 가능하게 하며, 이는 이제 데이터-중속 종료 조건들을 생성하는 마지막 반복까지 인터럽트 없이 효과적으로 실행될 수 있다.

[0253] 다시 도 9 및 도 10을 참조하면, 제어 레지스터들(340)은 메모리 영역 RAM(405)을 포함한다. 다양한 실시 예에

서, 메모리 영역 RAM(405)은 또한 아래에서 설명되는 중첩 루프들을 통해 TID 스택의 상단(식별자들을 갖게)을 유지할 수 있다. 상술한 바와 같이, 각 중첩 루프는 새로운(또는 재사용된) TID 세트의 스레드들을 개시한다. 루프의 스레드들은 그것의 TID와 외부 루프 스레드들의 TID들에 액세스해야 할 수 있다. 각 중첩 루프 스레드의 TID들에 액세스하면 상술된 상이한 레벨 또는 유형들의 TID들, TID<sub>0</sub>, TID<sub>1</sub> 및 TID<sub>2</sub>와 같은 각 스레드의 전용 변수들에 액세스할 수 있다. 스택 상단 TID 식별자는 활성 스레드의 TID를 나타낸다. 스택 상단 TID 식별자는 세 가지 TID(TID<sub>0</sub>, TID<sub>1</sub> 및 TID<sub>2</sub>) 중 다양한 동작에 사용되는 TID를 선택하는 데 사용된다. 이러한 세 개의 TID 및 스택 상단 TID 식별자는 동기 메시 통신 네트워크(275)상에 송신되는 동기 패브릭 제어 정보(또는 메시지들)에 포함되므로, 각 스레드에 알려지게 된다. 다수의 TID가 동기 패브릭 메시지 내에 포함되고 스택 상단 TID 식별자를 포함하기 때문에, 다수의 TID를 사용하면 중첩 루프의 스레드가 중첩 루프 스레드들 내의 임의의 레벨로부터의 변수들에 액세스할 수 있다. 선택된 TID와 타일 메모리 영역 RAM(405) 식별자는 전용 스레드 변수에 액세스하는 데 사용된다.

[0254] 본 개시의 또 다른 신규한 특징은 루프 반복 종속성을 핸들링하기 위해 루프 스레드 실행을 순서화하는 메커니즘이며, 이는 또한 비동기 패킷 네트워크(265)를 통해 수신된 완료 메시지들 및 데이터의 임의의 지연을 수용한다. 도 23은 동기 도메인들을 형성하는 타일들(210) 및 하이브리드 스레딩 패브릭 회로 클러스터에 의한 계산의 루프의 수행을 위한 대표적인 비동기 패킷 네트워크 메시징 및 동기 메시징의 도해이다. 도 23에 도시된 바와 같이, 다수의 동기 도메인(682, 684 및 686)은 프리-루프 제1 동기 도메인(526) 및 포스트-루프(제5) 동기 도메인(654)에 더하여, 루프 계산, 제2 동기 도메인(682), 제3 동기 도메인(684), 제4 동기 도메인(686)의 성능에 관여한다. 루프 계산은 중첩 루프들을 포함한 임의의 종류의 루프일 수 있고, 이 경우 다양한 루프 내에 데이터 종속성이 있다. 예를 들어, 이러한 데이터 종속성은 비동기 패킷 네트워크(265)를 통한 AF 메시징을 수반하여, 이를테면 메모리(125)로부터 정보가 필요할 때, 한 번의 반복 내에서 발생할 수 있다. 결과적으로, 스레드 실행은 임의의 특정 스레드가 완료 카운트 0을 가질 때(이는 해당 스레드에 대한 모든 완료 메시지가 도착하여, 해당 스레드가 어떠한 데이터도 대기하고 있지 않음을 의미함)마다 뿐만 아니라 정의된 순서로 진행되어야 한다.

[0255] 순서화된 루프 스레드 실행을 제공하기 위해, 대표적인 실시 예들에서, 추가 메시징 및 추가 필드들이 각 루프 반복에 대해, 완료 테이블(422)에 이용된다. 루프 베이스 타일(210B)은 동기 메시 통신 네트워크(275)를 통해 각 동기 도메인(682, 684, 686)을 통해 동기 메시지들(688)에 전달되는(즉, 주어진 해당 동기 도메인의 모든 연속적인 타일(210)에 전달되는) 네 개의 정보(각 루프 반복에 대해), 그리고 비동기 패킷 네트워크(265)를 통해 연속적인 동기 도메인들의 베이스 타일들(210)로 AF 계속 메시지들(692)(이는 그 다음 주어진 동기 도메인의 각각의 연속적인 타일(210)로 전달된다)을 제공한다. 그러한 네 개의 정보 필드는 그 다음 완료 테이블(422)에 저장 및 인덱싱되고 루프 실행이 진행됨에 따라 비교에 이용된다. 네 가지 정보는 다음과 같다: 루프에 대한 스레드 세트의 제1 스레드를 나타내는 제1 플래그, 루프에 대한 스레드 세트의 마지막 스레드를 나타내는 제2 플래그, 현재 스레드에 대한 TID, 및 다음 스레드에 대한 TID. 현재 스레드에 대한 TID는 TID 폴로부터 얻어지고, 다음 스레드에 대한 TID는 다음 스레드에 제공될 폴로부터의 TID이다. 이러한 네 가지 정보는 각 연속 동기 도메인의 베이스 타일에 의해 스레드 시작을 순서화하는 데 사용된다. 스레드는 그것의 종속성 카운트가 0에 도달하고 스레드가 루프의 첫 번째 스레드이거나 스레드 TID가 이전에 시작된 스레드에 대한 다음 TID와 같은 경우 시작될 수 있다.

[0256] 달리 말하면, 스레드 제어 회로(330)(일반적으로 다양한 상태 기계를 포함함)는 모든 데이터 완료를 수신한(그에 따라 그 외 실행할 준비가 될) 각 스레드에 대해, 스레드가 실행할 다음 스레드인지( 다음 스레드 ID, 예를 들어, TID = 4를 갖는) 여부에 대해 완료 테이블(422)을 검사하고, 그렇다면, 해당 스레드(TID = 4)를 실행 큐(440)로 이동하고, 그렇지 않다면, 해당 스레드(예를 들어, 데이터 완료 카운트가 0이 되었지만 TID = 5를 갖는)를 시작하지 않고 TID가 다음으로 시작할 인덱스를 유지한다. 다음 TID를 갖는 스레드에 대한 데이터 완료가 0으로 감소하여(이 경우 TID = 4), 모든 완료 메시지가 도착했을 때, 해당 스레드는 실행을 위해 큐잉되고 실행될 스레드(TID = 4)는 또한 다음 TID(이 경우, 그것의 당므 TID는 TID = 5)를 갖는다. 따라서, TID = 4를 갖는 스레드가 완료되었을 때, 스레드 제어 회로(330)는 완료 테이블(422)을 체크하고 이제 스레드(TID = 5)가 다음 스레드 ID임을 결정하며, 실행을 위해 해당 스레드를 큐잉한다. 그 다음 스레드 ID가 마지막 TID일 때, 그것의 실행 이후, AF 완료 메시지(656)가 포스트-루프 베이스 타일(이 경우 210E)로 송신될 수 있다. 완료 테이블(422)에 추가 필드들의 이러한 사용은 스레드 실행의 특정 순서가 유지되어야 하는 임의의 상황으로 확장될 수 있다는 점에 유의해야 한다.

[0257] 도 24는 조건부 브랜칭 회로(370)의 대표적인 실시 예의 블록 및 회로도이다. 상술된 제1, 제2 및 제3 동기 도메인들과 같은 동기 도메인은 동기 메시 통신 네트워크(275)를 통해 시퀀스 또는 직렬로 연결된 일련의 상호 접

속된 타일이다. 스프레드의 실행은 베이스 타일(210)로 지칭되는 동기 도메인의 첫 번째 타일(210)에서 시작하고, 동기 메시 통신 네트워크(275)의 구성된 연결들을 통해 거기서부터 동기 도메인의 다른 타일들(210)로 진행한다. 도 24에 도시된 바와 같이, 타일(210)이 동기 도메인의 베이스 타일(210)로서 구성되었을 때(그러한 구성들이 런타임에 앞서 HTF 회로(200)로 로딩됨에 따라), 구성 메모리 멀티플렉서(372)의 선택(374)은 1과 같게 설정되며, 이에 의해 명령 RAM(315)으로부터 명령들을 선택하기 위한 명령 인덱스를 제공하기 위한 스포크 RAM(320)을 선택한다. 동기 도메인의 다른 모든 타일(210)에 대해, 구성 메모리 멀티플렉서(372)의 선택(374)은 0과 같게 설정되며, 이에 의해 동기 도메인의 타일들의(210)의 시퀀스에서의 이전 타일(210)에 의해 제공된 명령 인덱스를 선택한다. 결과적으로, 베이스 타일(210)은 (상술된 바와 같이, 마스터 동기 입력들을 지정한) 통신 라인들(또는 와이어들)(270B 및 270A)의 지정된 필드들(또는 부분들)을 통해 도메인의 다음, 두 번째 타일에 실행될 명령 인덱스(또는 명령)를 제공한다. 기본적으로, 그 후, 이러한 다음 타일(210), 및 동기 도메인의 각각의 연속적인 타일(210)은 정적 구성으로서, 실행을 위해 연결된 타일들(210)의 각 다음 타일(210)에 동일한 명령을 제공할 것이다.

[0258] 그러나, 대표적인 실시 예들에서, 스포크 RAM(320), 명령 RAM(315), 및 조건부 브랜칭 회로(370)를 사용하는 동적 자체 구성을 위한 메커니즘이 제공된다. 도 24를 참조하면, 현재 타일(210)에 대해, ALB Op(310)는 예를 들어, 하나의 입력이 제2 입력보다 큰지 여부와 같은 테스트 조건의 결과인 출력을 생성하도록 구성될 수 있다. 그러한 테스트 조건 출력은 통신 라인들(또는 와이어들)(378)상의 조건부 브랜칭 회로(370)에 제공된다. 조건부 브랜칭 회로(370)가 인에이블될 때(라인들(또는 와이어들)(379)상에 제공되는 명령의 하나 이상의 비트를 통해), 테스트 조건 출력은 동기 도메인의 다음 타일(210)에 제공되는 다음 명령 인덱스(또는 명령)를 선택하는데, 이를테면 다음 타일(210)에 대한 "X" 명령 또는 "Y" 명령 간에서 선택하는 데 이용되어, 첫 번째 또는 두 번째 명령이 선택될 때 데이터 경로의 조건부 브랜칭을 제공한다. 그러한 조건부 브랜칭은 또한 이를테면 다음 타일(210)이 또한 조건부 브랜칭을 제공하도록 인에이블될 때, 캐스케이드될 수도 있다. 다음 타일들(210) 중 하나 이상에 대한 다음 명령을 선택함으로써, 동적 자체 구성 및 자체 재구성이 각각의 그러한 HTF 회로 클러스터(205)에서 인에이블된다.

[0259] 대표적인 실시 예에서, 조건부 브랜칭 회로(370)는 테스트 조건 결과에 따라, 두 개의 상이한 명령 사이에서 선택 또는 토글링하도록 배열되었다. 브랜치 인에이블은 현재(또는 현재 다음) 명령의 필드들 중 하나에 제공되고, 조건부 브랜칭 회로(370)의 AND 게이트(362)에 제공되며, 여기서 테스트 조건 출력과 AND 처리된다. 테스트 조건 출력이 논리 "0" 또는 "1"인 것에 따라, AND 게이트(362)는 논리 "0" 또는 "1"을 출력으로 생성할 것이며, 이는 OR 게이트(364)에 대한 입력으로서 제공된다. 현재 다음 명령 인덱스의 선택된 필드의 다른 지정된 비트, 통상적으로 다음 명령 인덱스의 최하위 비트("LSB")도 OR 게이트(364)에 제공되며, 여기서 AND 게이트(362)의 출력과 OR 연산된다. 다음 명령 인덱스의 LSB는 0이고, 그것이 AND 게이트(362) 출력의 논리 "1"과 OR 연산된다면, 다음 출력되는 다음 명령 인덱스가 1씩 증분되어, 상이한 다음 명령 인덱스를 다음 타일(210)에 제공한다. 다음 명령 인덱스의 LSB는 0이고, 그것이 AND 게이트(362) 출력의 논리 "0"과 OR 연산된다면, 다음 출력되는 다음 명령 인덱스가 1씩 증분되지 않아, 동일한 다음 명령 인덱스를 다음 타일(210)에 제공한다. 결과적으로, 현재 타일(210)은 연결된 타일(210)이 실행하기 위한 대체 명령을 조건부로 지정하여, HTF 회로 클러스터(205)에서 하나 이상의 사례 설명문의 수행을 인에이블한다. 대체 명령은 현재 타일의 명령과 동기 도메인의 다음 타일(210)에 대한 다음 명령 인덱스로서 제공되는 대체 명령 간을 선택하기 위해 현재 타일의 데이터 경로가 부울 조건 값을 생성하고, 부울 값을 사용함으로써 선택된다. 또한 결과적으로, 현재 타일(210)은 다음 타일(210) 등을 동적으로 구성하여, 각 HTF 회로 클러스터(205)에서 동적 자체 구성 및 자체 재구성되게 한다.

[0260] 대표적인 실시 예들의 많은 이점이 쉽게 명백하다. 대표적인 장치, 시스템 및 방법은 예를 들어 그리고 제한없이, 스펙트럼 기술, 기계 학습, 5G 네트워킹 알고리즘 및 대용량 스텔실 코드를 사용하는 그래프 클러스터링과 같은 합성 개구 레이더, 5G 기지국, 및 그래프 분석 애플리케이션들과 같은 감지, 통신 및 분석 애플리케이션들에 사용되는 고속 푸리에 변환(FFT, Fast Fourier Transform) 및 유한 임펄스 응답(FIR, Finite Impulse Response) 필터들의 계산과 같은 계산 집약적 커널들을 위한 고성능 및 에너지 효율적인 솔루션들을 제공할 수 있는 컴퓨팅 아키텍처를 제공한다.

[0261] 여기서 사용될 때, "프로세서"(110, 300)는 임의의 유형의 프로세서 또는 제어기일 수 있고, 여기서 논의된 기능을 수행하도록 구성, 설계, 프로그래밍 또는 다른 방식으로 구성되는 하나 이상의 프로세서(들)(110, 300)로서 구현될 수 있다. 프로세서 또는 제어기라는 용어가 여기서 사용될 때, 프로세서(110, 300)는 단일 집적 회로("IC")의 사용을 포함할 수 있거나, 또는 복수의 집적 회로 또는 함께 연결, 배열 또는 그룹화되는 다른 구성요소, 이를테면 제어기, 마이크로 프로세서, 디지털 신호 프로세서("DSP"), 어레이 프로세서, 그래픽 또는 이미지

프로세서, 병렬 프로세서, 다중 코어 프로세서, 맞춤형 IC, 주문형 집적 회로("ASIC"), 필드 프로그램 가능한 게이트 어레이("FPGA"), 적응형 컴퓨팅 IC, 관련 메모리(이러테면 RAM, DRAM 및 ROM), 및 기타 IC 및 구성요소(아날로그 또는 디지털)의 사용을 포함할 수 있다. 결과적으로, 여기서 사용될 때, 프로세서 또는 제어기라는 용어는 마이크로 프로세서 메모리 또는 추가 RAM, DRAM, SDRAM, SRAM, MRAM, ROM, FLASH, EPROM 또는 E<sup>2</sup>PROM과 같은 관련 메모리와 함께 여기서 논의된 기능들을 수행하는 단일 IC, 또는 맞춤형 IC, ASIC, 프로세서, 마이크로 프로세서, 제어기, FPGA, 적응형 컴퓨팅 IC 또는 몇몇 기타 집적 회로 그룹의 배열을 의미하고 포함하는 것으로 이해되어야 한다. 연관된 메모리를 갖는 프로세서(110, 300)는 여기서 논의된 바와 같이, 본 발명의 방법론을 수행하기 위해 (프로그래밍, FPGA 상호 접속 또는 하드웨어에 내장을 통해) 구성될 수 있다. 예를 들어, 방법론은 프로세서(110, 300)가 동작 가능(즉, 전원이 켜지고 작동)할 때 후속 실행을 위해 프로그램 명령 세트 또는 다른 코드(또는 동등한 구성 또는 다른 프로그램)로서 그것의 연관된 메모리(및/또는 메모리(125)) 및 기타 동등한 구성요소들과 함께 프로세서(110, 300)에서 프로그래밍되고 저장될 수 있다. 동등하게, 프로세서(110, 300)가 전체적으로 또는 부분적으로 FPGA, 맞춤형 IC 및/또는 ASIC로 구현될 수 있을 때, FPGA, 맞춤형 IC 또는 ASIC는 또한 본 발명의 방법론을 구현하기 위해 설계, 구성 및/또는 하드웨어에 내장될 수 있다. 예를 들어, 프로세서(110, 300)는 "프로세서" 또는 "제어기"로 통칭되는 아날로그 및/또는 디지털 회로, 제어기, 마이크로 프로세서, DSP 및/또는 ASIC의 배열로 구현될 수 있으며, 이들은 각각 가능한 메모리(125)와 함께를 포함하여, 본 발명의 방법론을 구현하도록 하드웨어에 내장, 프로그래밍, 설계 또는 구성된다.

[0262] 데이터 저장소(또는 데이터베이스)를 포함할 수 있는 메모리(125, 325)는 제한 없이 RAM, FLASH, DRAM, SDRAM, SRAM, MRAM, FeRAM, ROM, EPROM 또는 E<sup>2</sup>PROM, 또는 임의의 다른 형태의 메모리 디바이스, 이를테면 자기 하드 드라이브, 광학 드라이브, 자기 디스크 또는 테이프 드라이브, 하드 디스크 드라이브, 기타 기계 판독 가능한 스토리지 또는 메모리 매체 이를테면 플로피 디스크, CDROM, CD-RW, 디지털 다용도 디스크(DVD) 또는 기타 광학 메모리, 또는 선택된 실시 예에 따라 알려져 있거나 알려지게 될 임의의 다른 유형의 메모리, 저장 매체, 또는 데이터 스토리지 장치 또는 회로를 포함하여, 이에 제한되지는 않지만, 메모리 집적 회로("IC") 또는 집적 회로의 메모리 부분(이를테면 프로세서(130) 또는 프로세서 IC 내의 상주 메모리)(휘발성 또는 비휘발성)(이동식 또는 비이동식)을 포함하여, 현재 알려져 있거나 미래에 이용 가능하게 될 정보의 저장 또는 통신을 위한 임의의 컴퓨터 또는 기타 기계 판독 가능한 데이터 저장 매체, 메모리 디바이스 또는 기타 저장 또는 통신 디바이스 내를 포함하여 임의의 수의 형태로 구현될 수 있다. 메모리(125, 325)는 다양한 록업 테이블, 파라미터, 계수, 기타 정보 및 데이터, 프로그램 또는 명령(본 발명의 소프트웨어의) 및 다른 유형들의 테이블들 이를테면 데이터베이스 테이블들을 저장하도록 구성될 수 있다.

[0263] 상술된 바와 같이, 프로세서(110, 300)는 예를 들어, 본 발명의 방법론을 수행하기 위해 본 발명의 소프트웨어 및 데이터 구조를 사용하여, 하드웨어에 내장되거나 프로그래밍된다. 결과적으로, 구성 메모리(160)의 다양한 명령을 포함하여 본 발명의 시스템 및 관련 방법들은 상술된 비밀시적 컴퓨터 판독 가능한 매체 내에 구현되는 명령 세트 및/또는 메타 데이터와 같은 그러한 프로그래밍 또는 기타 명령들을 제공하는 소프트웨어로서 구현될 수 있다. 또한, 메타 데이터는 또한 록업 테이블 또는 데이터베이스의 다양한 데이터 구조를 정의하는 데 사용될 수도 있다. 그러한 소프트웨어는 예로서 그리고 제한 없이, 소스 또는 객체 코드의 형태일 수 있다. 소스 코드는 추가로 어떤 형태의 명령들 또는 객체 코드(어셈블리 언어 명령들 또는 구성 정보를 포함하는)로 컴파일될 수 있다. 본 발명의 소프트웨어, 소스 코드 또는 메타 데이터는 C, C++, Matlab, SystemC, LISA, XML, Java, Brew, SQL 및 그 변형(예를 들어, SQL 99 또는 SQL의 독점 버전들), DB2, Oracle 또는 다양한 하드웨어 정의 또는 하드웨어 모델링 언어(예를 들어, Verilog, VHDL, RTL) 및 결과 데이터베이스 파일들(예를 들어, GDSII)을 포함하여 여기서 논의된 기능을 수행하는 임의의 다른 유형의 프로그래밍 언어와 같은 임의의 유형의 코드로서 구현될 수 있다. 결과적으로, 여기서 동등하게 사용될 때, "구성", "프로그램 구성", "소프트웨어 구성" 또는 "소프트웨어"는 (예를 들어, 프로세서 또는 컴퓨터에 인스턴스화되거나 로딩되고 프로세서(110, 300)를 포함하여 실행될 때) 지정된 관련 기능 또는 방법론을 제공하거나 제공하는 것으로 해석될 수 있는 임의의 구문 또는 서명들을 갖는 임의의 유형의 임의의 프로그래밍 언어를 의미하고 지칭한다.

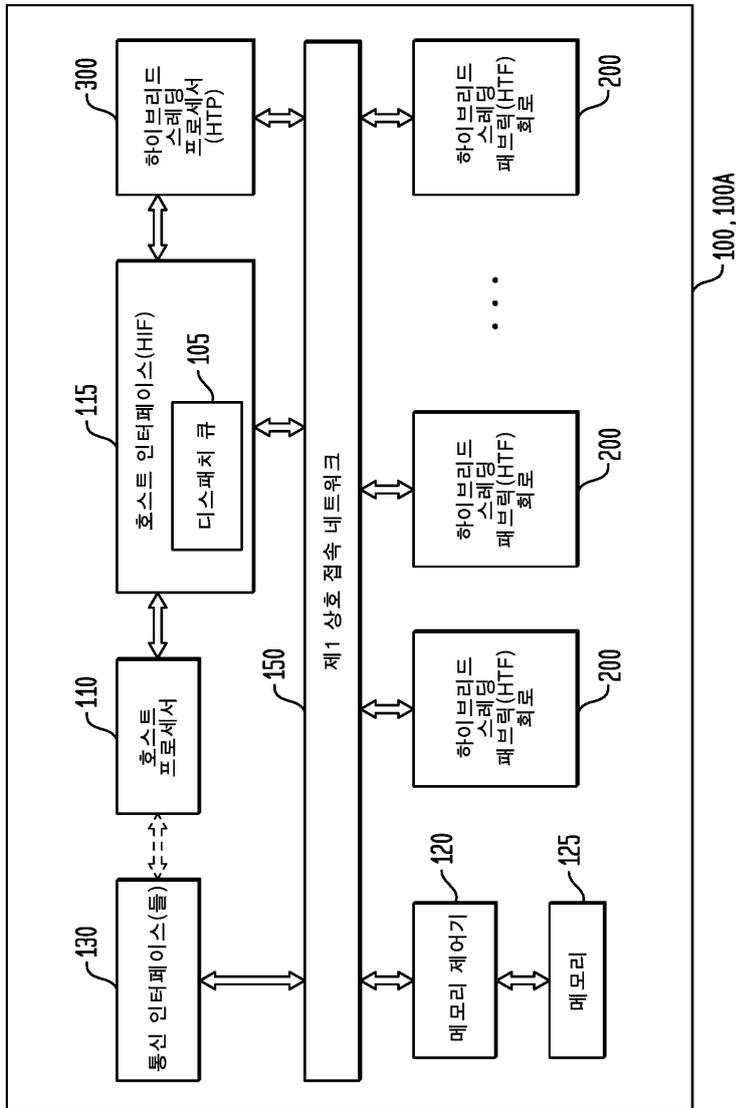
[0264] 본 발명의 소프트웨어, 메타 데이터 또는 기타 소스 코드 및 임의의 결과 비트 파일(객체 코드, 데이터베이스 또는 록업 테이블)은 상술된 바와 같이, 메모리(125, 325), 예를 들어, 플로피 디스크, CDROM, CD-RW, DVD 자기 하드 드라이브, 광학 드라이브 또는 임의의 다른 유형의 데이터 스토리지 장치 또는 매체와 관련하여 상술된 것과 같은 컴퓨터 판독 가능한 명령들, 데이터 구조들, 프로그램 모듈들 또는 기타 데이터로서, 임의의 컴퓨터 또는 다른 기계 판독 가능한 데이터 스토리지 매체와 같은 임의의 유형의 비밀시적 저장 매체 내에 구현될 수 있다.

- [0265] 통신 인터페이스(들)(130)는 관련 채널, 네트워크 또는 버스에 대한 적절한 연결에 이용된다; 예를 들어, 통신 인터페이스(들)(130)는 유선 또는 무선 인터페이스를 위한 임피던스 매칭, 드라이버들 및 기타 기능들을 제공할 수 있고, 무선 인터페이스를 위한 복조 및 아날로그-디지털 변환을 제공할 수 있으며, 프로세서 (110, 300) 및/또는 메모리(125)를 위해 각각 다른 디바이스들과의 물리적 인터페이스를 제공할 수 있다. 일반적으로, 통신 인터페이스(들)(130)는 선택된 실시 예에 따라, 프로그램 명령, 파라미터, 구성 정보, 제어 메시지, 데이터 및 기타 관련 정보와 같은 데이터를 수신 및 송신하는 데 사용된다.
- [0266] 통신 인터페이스(들)(130)는 HTF(200) 및/또는 프로세서(110, 300)와 임의의 유형의 네트워크 또는 무선 디바이스 간에, 이를테면 무선, 광 또는 유선으로, 그리고 임의의 적용 가능한 표준(예를 들어, PCI, USB, RJ 45, 이더넷(예를 들어 그리고 제한 없이, 고속 이더넷, 기가비트 이더넷, 300ase-TX, 300ase-FX 등), IEEE 802.11, 블루투스, WCDMA, 와이파이, GSM, GPRS, EDGE, 3G 및 상술된 다른 표준들 및 시스템들)을 사용하여 데이터 통신을 제공하기 위해 해당 기술분야에 알려져 있거나 알려지게 될 수 있는 바와 같이 구현될 수 있고, 임피던스 매칭 기능, 고전압 제어 버스와 인터페이스하기 위한 저전압 프로세서용 전압 변환, 유선 또는 무선 송수신기, 및 프로세서(130)로부터의 시그널링에 응답하여 다양한 라인 또는 커넥터를 켜거나 끄는 다양한 스위칭 메커니즘(예를 들어, 트랜지스터)를 포함할 수 있다. 또한, 통신 인터페이스(들)(130)는 또한 예를 들어, 디스플레이상의 출력을 위해 실시간으로 정보를 수신하기 위해, 예를 들어, 이를테면 하드웨어 내장 또는 RF 또는 적외선 시그널링을 통해, 시스템(100) 외부로 신호들을 수신 및/또는 송신하도록 구성될 수도 있다. 통신 인터페이스(들)(130)는 임의의 선택된 아키텍처를 사용하여 임의의 유형의 버스 또는 네트워크 구조 또는 매체에 대한 연결을 제공할 수 있다. 예로서 그리고 제한 없이, 그러한 아키텍처들은 ISA(Industry Standard Architecture) 버스, EISA(Enhanced ISA) 버스, MCA(Micro Channel Architecture) 버스, PCI(Peripheral Component Interconnect) 버스, SAN 버스 또는 임의의 기타 통신 또는 시그널링 매체, 이를테면 이더넷, ISDN, T1, 위성, 무선 등을 포함한다.
- [0267] 본 발명은 본 발명의 원리들의 예증으로 간주되어야 하고 본 발명을 이의 구체적인 구체적인 실시 예들로 제한하려는 것이 아니다. 이와 관련하여, 본 발명은 그 적용에 있어서 위아래에 제시되거나, 도면들에 도시되거나, 또는 예들에 설명되는 대로 구성의 세부 사항들 및 구성요소들의 배열들로 제한되지 않는 것으로 이해되어야 한다. 본 발명에 따른 시스템들, 방법들 및 장치들은 다른 실시 예들이 가능하고 다양한 방식으로 실시 및 수행될 수 있다.
- [0268] 본 발명은 구체적인 실시 예들과 관련하여 설명되었지만, 이러한 실시 예들은 단지 예시일 뿐 본 발명을 제한하는 것은 아니다. 여기서의 설명에서, 본 발명의 실시 예들에 대한 완전한 이해를 제공하기 위해, 전자 부품, 전자 및 구조적 연결, 재료 및 구조적 변형의 예들과 같은 많은 구체적인 세부 사항이 제공된다. 그러나, 관련 기술의 통상의 기술자들은 본 발명의 실시 예들이 하나 이상의 구체적인 세부 사항 없이 또는 다른 장치, 시스템, 어셈블리, 구성요소, 재료, 부품 등과 함께 실시될 수 있음을 인식할 것이다. 다른 경우들에서, 잘 알려져 있는 구조, 재료 또는 동작은 본 발명의 실시 예들의 양태들을 모호하게하는 것을 피하기 위해 구체적으로 도시되거나 상세하게 설명되지 않는다. 또한, 다양한 도면은 축척에 따라 그려지지 않고 제한적인 것으로 간주되어서는 안 된다.
- [0269] 본 명세서 전체에서 "일 실시 예", "실시 예" 또는 특정 "실시 예"에 대한 언급은 실시 예와 관련하여 설명된 특정 특징, 구조 또는 특성이 본 발명의 적어도 하나의 실시 예에 포함된다는 것을 의미하고 모든 실시 예에서 반드시 그런 것은 아니며, 또한 반드시 동일한 실시 예를 지칭하는 것은 아니다. 더욱이, 본 발명의 임의의 특정 실시 예들의 특정 특징, 구조 또는 특성은 임의의 적절한 방식으로 그리고 다른 특징의 대응하는 사용 없이 선택된 특징의 사용을 포함하여 하나 이상의 다른 실시 예와 임의의 적절한 조합으로 조합될 수 있다. 또한, 본 발명의 본질적인 범위 및 사상에 특정 응용, 상황 또는 재료를 적용하기 위해 많은 수정이 이루어질 수 있다. 여기서 설명되고 예시된 본 발명의 실시 예의 다른 변형 및 수정이 여기서의 교시에 비추어 가능하며 본 발명의 사상 및 범위의 일부로 간주되어야 함을 이해해야 한다.
- [0270] 여기서의 숫자 범위의 인용에 대해, 동일한 정밀도를 갖는 그 사이의 각각의 개재되는 숫자가 명시적으로 고려된다. 예를 들어, 6-9 범위의 경우, 6 및 9 이외에 숫자 7 및 8이 고려되고, 범위 6.0-7.0의 경우, 숫자 6.0, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9 및 7.0이 명시적으로 고려된다. 또한, 범위 내의 모든 개재되는 하위 범위는 임의의 조합으로 고려되며 본 개시의 범위 내에 있다. 예를 들어, 5 - 10의 범위의 경우, 하위 범위 5 - 6, 5 - 7, 5 - 8, 5 - 9, 6 - 7, 6 - 8, 6 - 9, 6 - 10, 7 - 8, 7 - 9, 7 - 10, 8 - 9, 8 - 10 및 9 - 10이 고려되고 개시된 범위의 범위 내에 있다.

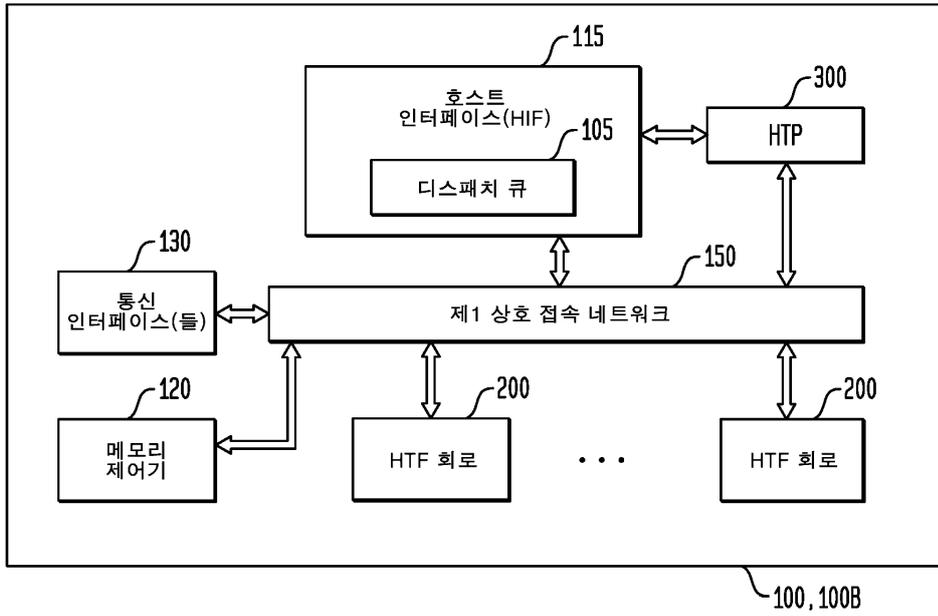
- [0271] 또한, 도면들에 도시된 요소들 중 하나 이상은 더 분리되거나 통합된 방식으로 구현될 수 있고, 본 출원에 따라 유용할 수 있는 것처럼 특정 경우에 제거되거나 작동 불가능하게 될 수도 있음을 알 수 있을 것이다. 일체로 형성된 구성요소들의 조합들은 또한 특히 개별 구성요소들의 분리 또는 조합이 불분명하거나 식별할 수 없는 실시예들에 대해, 본 발명의 범위 내에 있다. 또한, "연결" 또는 "연결 가능한"과 같은 다양한 형태를 포함하여 여기서 사용되는 "연결된"이라는 용어의 사용은 일체로 형성된 구성요소들 및 다른 구성요소를 통해 연결되는 구성요소들을 포함하여, 그러한 직접 또는 간접적인 전기적, 구조적 또는 자기적 연결, 연결 또는 부착을 위한 임의의 직접 또는 간접 전기, 구조 또는 자기 연결, 연결 또는 부착 또는 적응 또는 기능을 의미하고 포함한다.
- [0272] 신호들과 관련하여, 여기서는 주어진 메트릭을 "나타내거나" 주어진 메트릭을 "대표하는" 파라미터들을 지칭하며, 여기서 메트릭은 조절기 또는 그 입력들 또는 출력들의 적어도 부분의 상태의 척도이다.
- [0273] 파라미터가 측정 항목과 직접적으로 관련되어 있어 파라미터를 조정하면 측정 항목을 만족스럽게 조정할 수 있는 파라미터가 측정 항목을 나타내는 것으로 간주된다. 파라미터가 메트릭의 배수 또는 일부를 나타내는 경우 메트릭의 허용 가능한 표현으로 간주될 수 있다.
- [0274] 더욱이, 도면들의 모든 신호 화살표는 달리 구체적으로 언급하지 않는 한 예시일 뿐 제한이 아닌 것으로 간주되어야 한다. 특히 분리 또는 조합하는 능력이 불분명하거나 예측 가능한 경우, 단계들의 구성요소들의 조합들도 본 발명의 범위 내로 고려될 것이다. 여기서 다음 청구범위에서 사용되는 이접 용어 "또는"은 일반적으로 달리 표기되지 않는 한, 접속 및 이접 의미를 모두 갖는("배타적 또는" 의미에 국한되지 않음) "및/또는"을 의미하는 것으로 의도된다. 여기서의 설명 및 다음 청구범위 전체에 사용될 때, "한", "하나의" 그리고 "상기"는 문맥이 달리 명확하게 지시하지 않는 한 복수 참조를 포함한다. 또한 여기서의 설명 및 다음 청구범위 전체에 걸쳐 사용될 때 "내"의 의미는 문맥이 달리 명확하게 지시하지 않는 한 "내" 및 "상"을 포함한다.
- [0275] 발명의 내용 또는 요약에서 설명된 것을 포함하여 본 발명의 예시된 실시예들에 대한 앞에서의 설명은 여기에 개시된 정확한 형태들로 본 발명을 총망라하거나 제한하려는 의도가 아니다. 상술된 바로부터, 본 발명의 신규한 개념의 사상 및 범위에서 벗어나지 않고 다양한 변형, 수정 및 대체가 의도되고 실행될 수 있음을 알 수 있을 것이다. 여기서 예시된 특정 방법 및 장치에 대한 제한은 의도되거나 추론되어서는 안 된다는 것을 이해해야 한다. 물론, 이는 청구범위의 범위 내에 있는 모든 그러한 수정을 첨부된 청구범위에 의해 포함하도록 의도된다.

도면

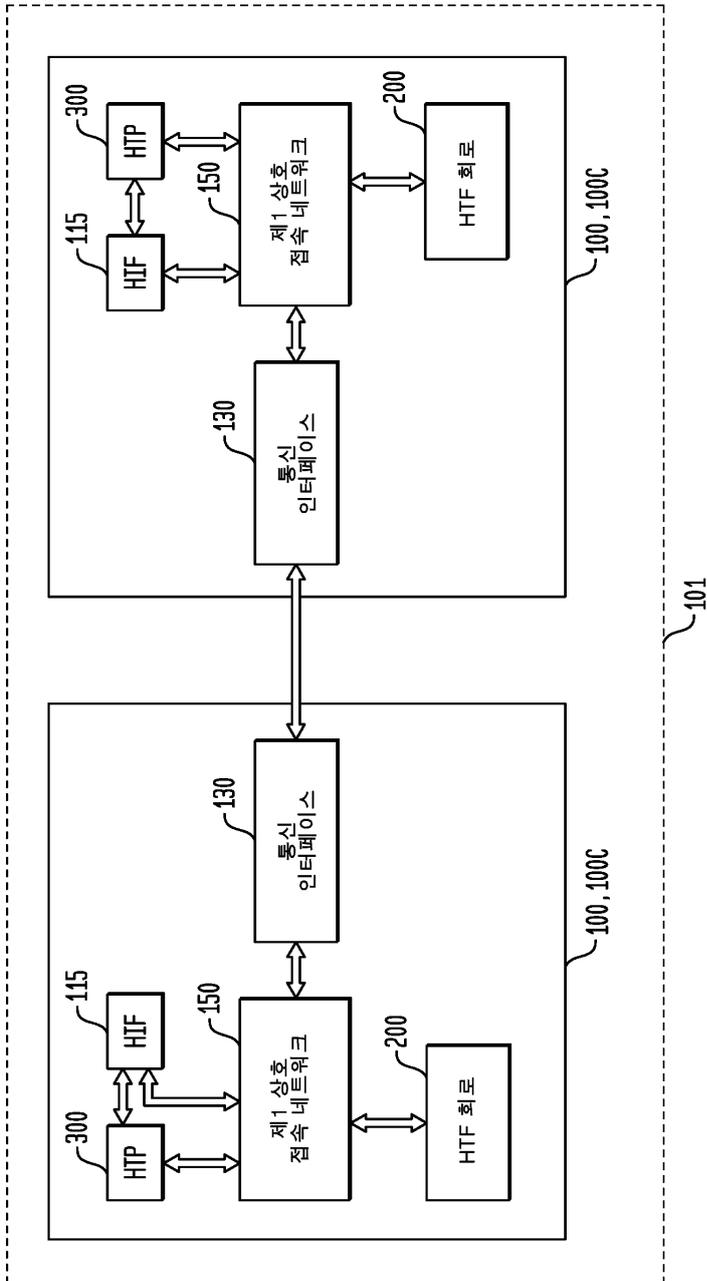
도면1



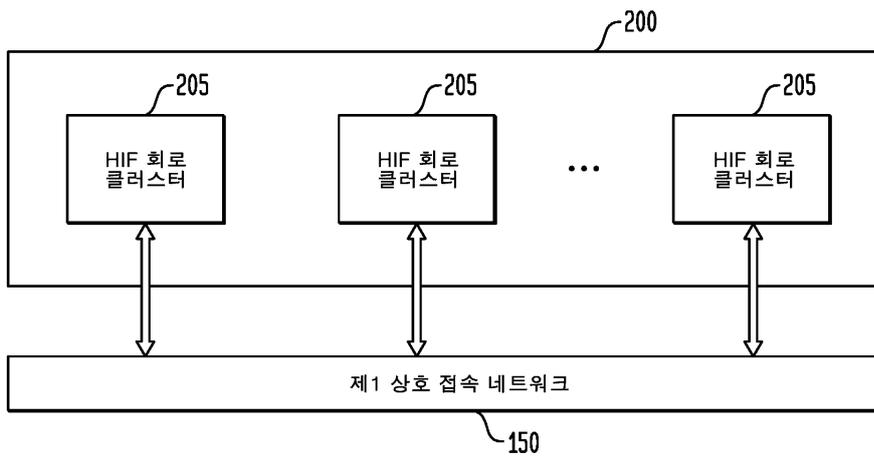
도면2



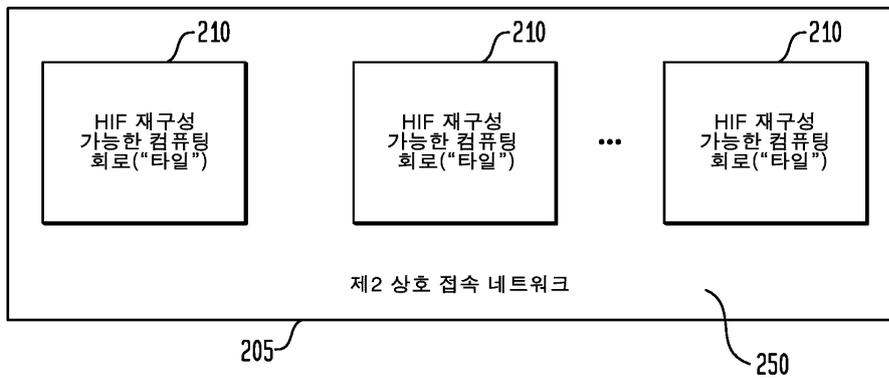
도면3



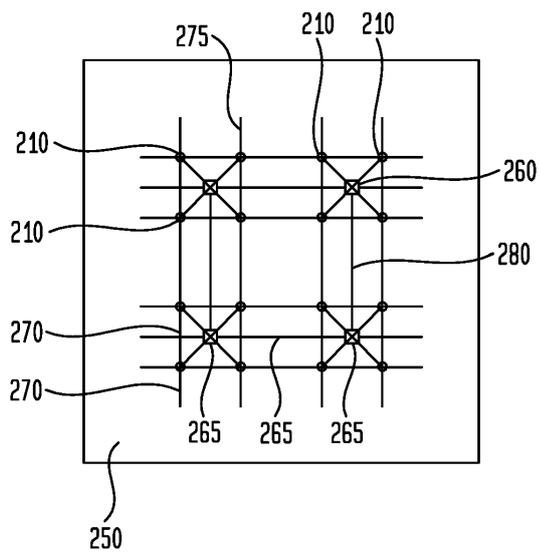
도면4



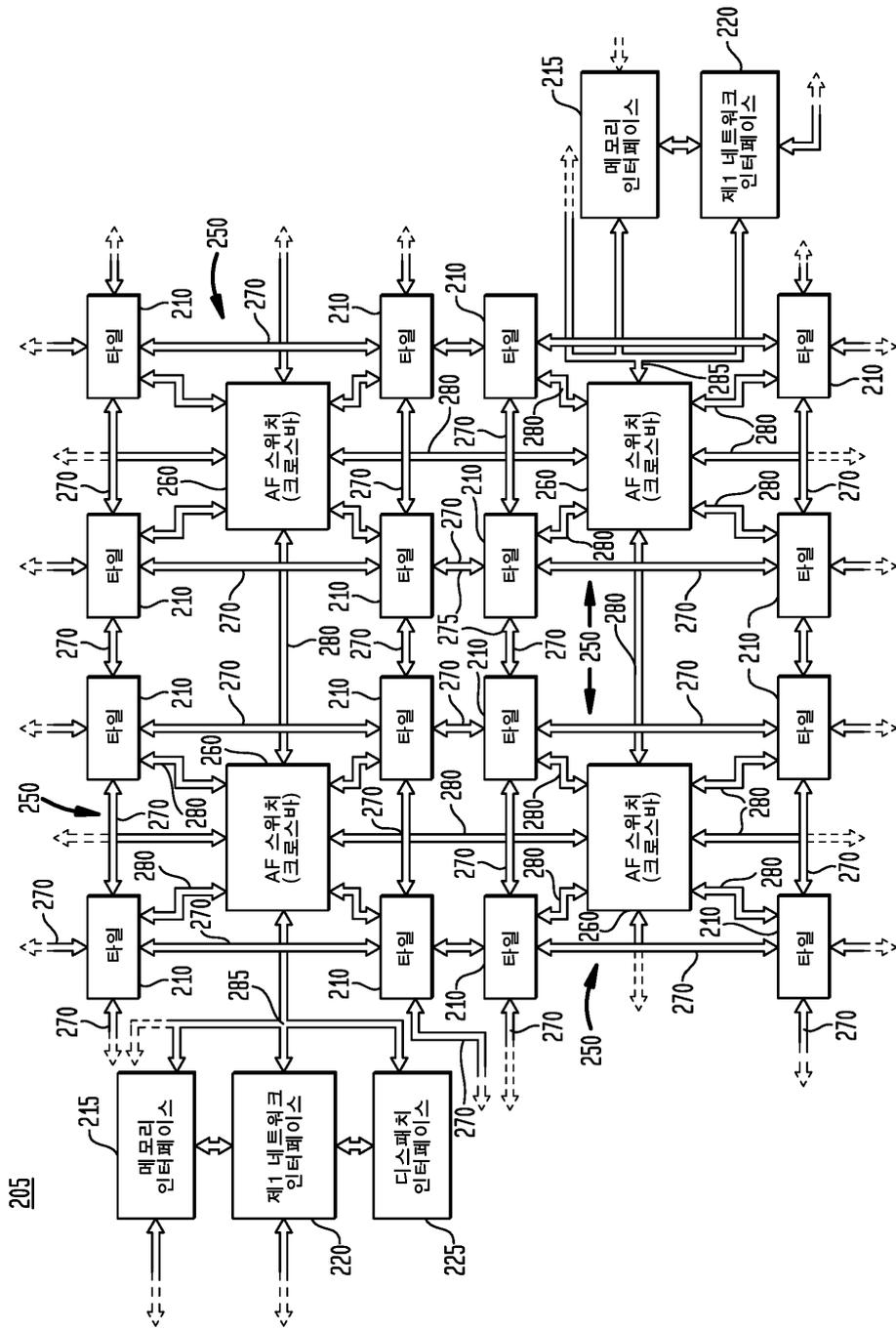
도면5



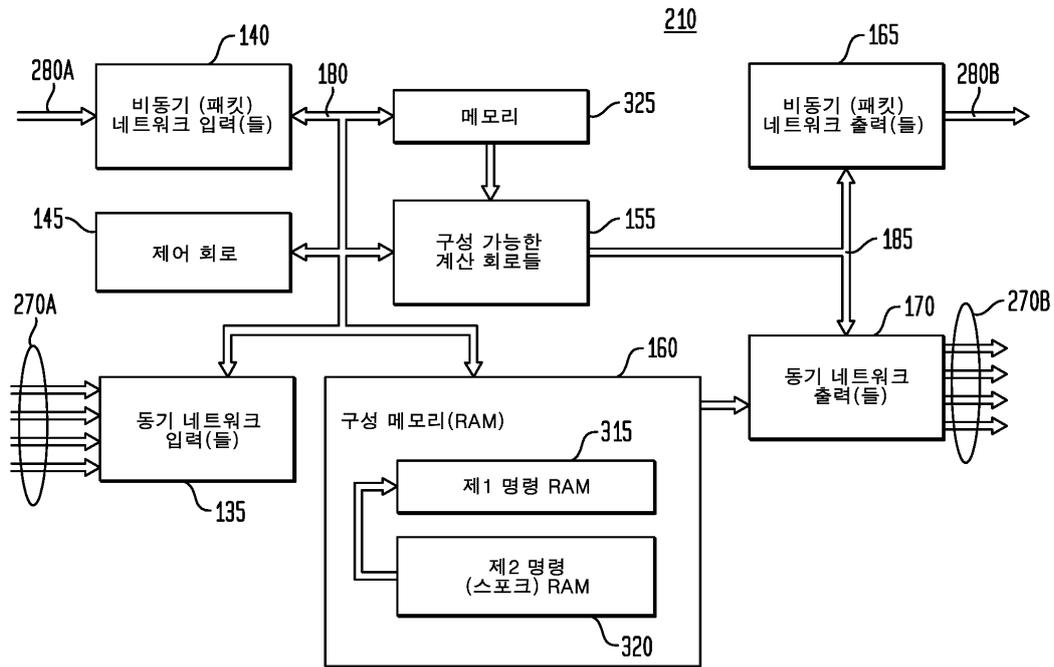
도면6



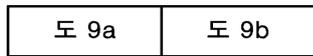
도면7



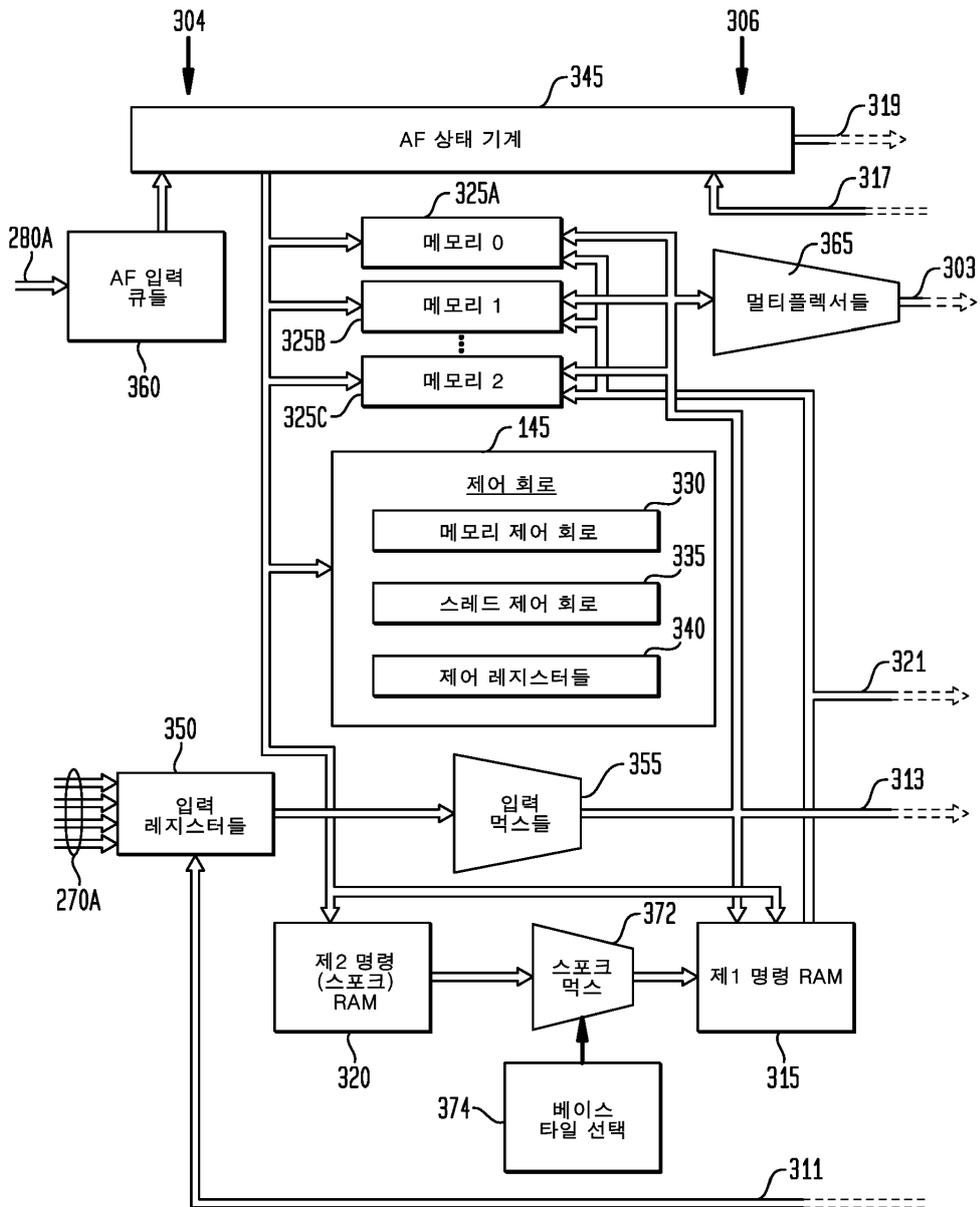
도면8



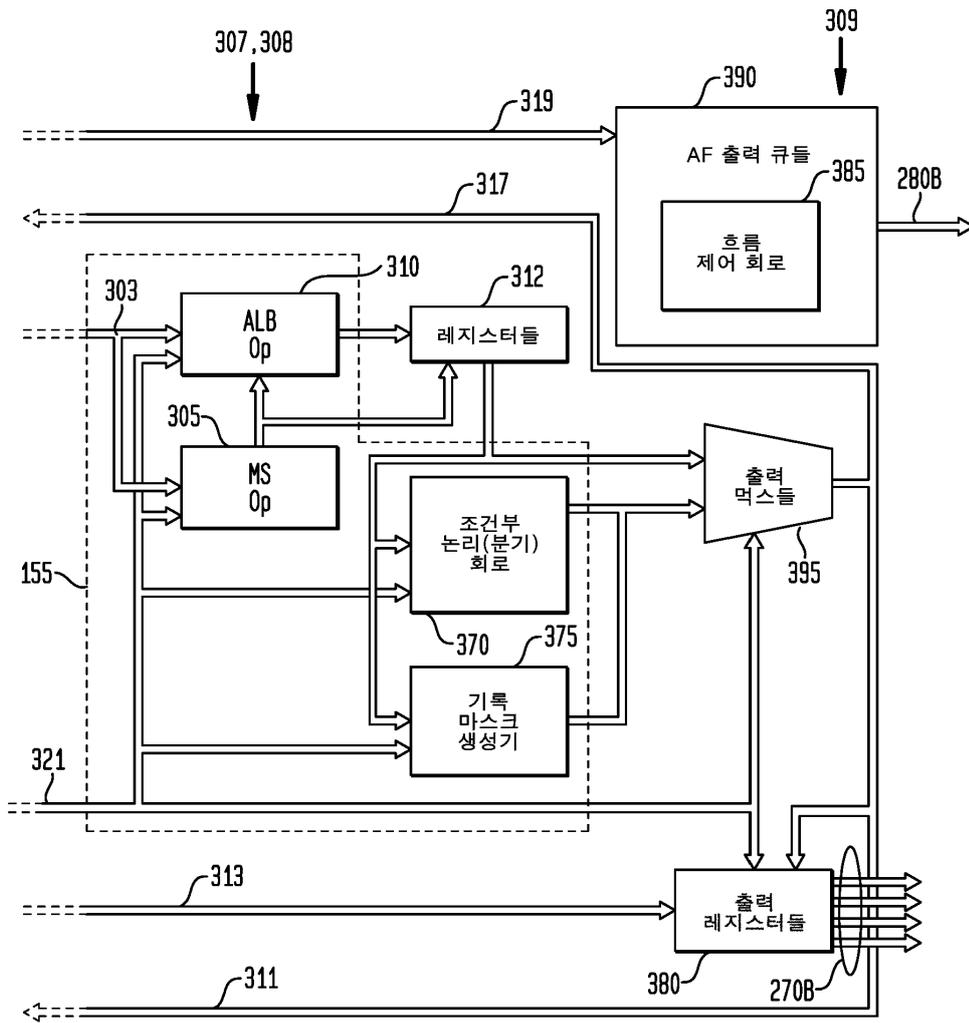
도면9



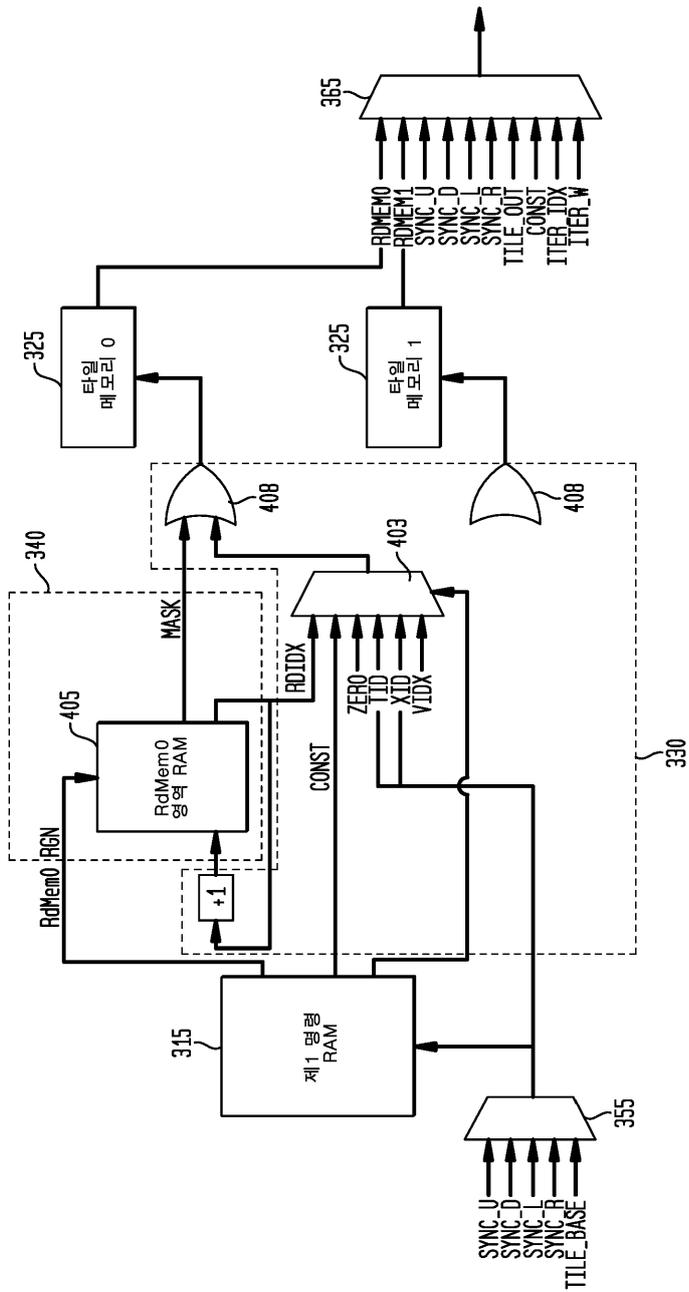
도면9a



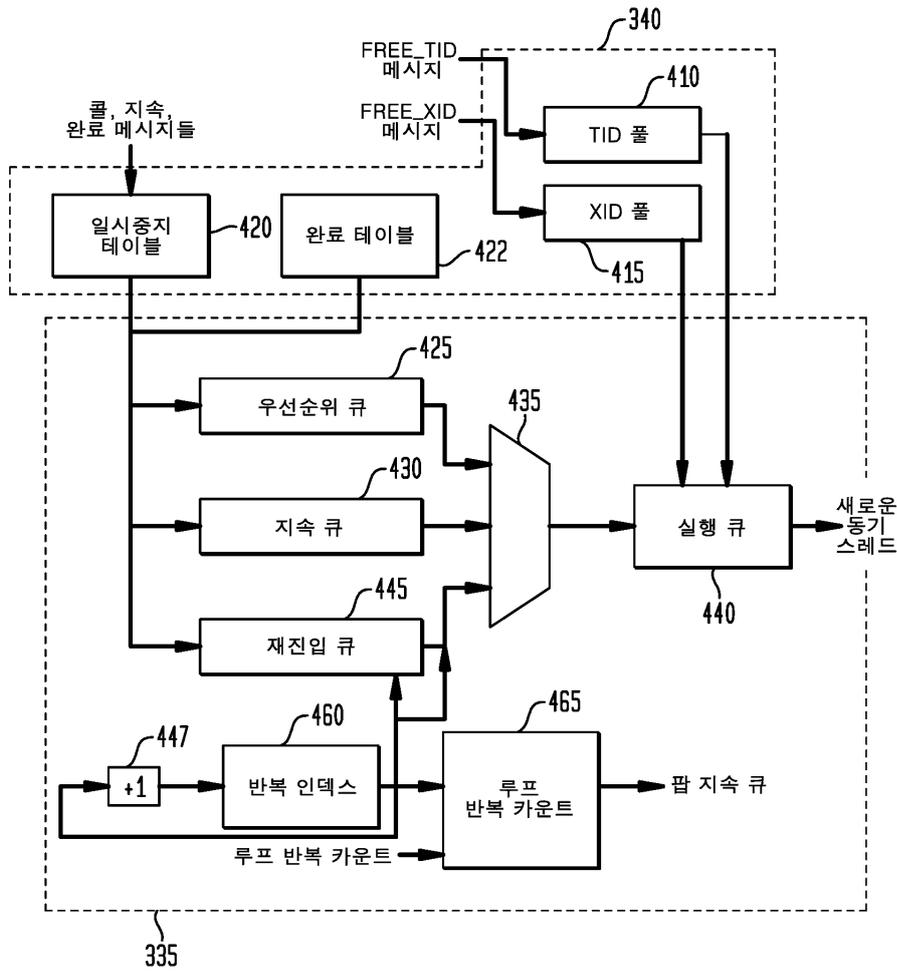
도면9b



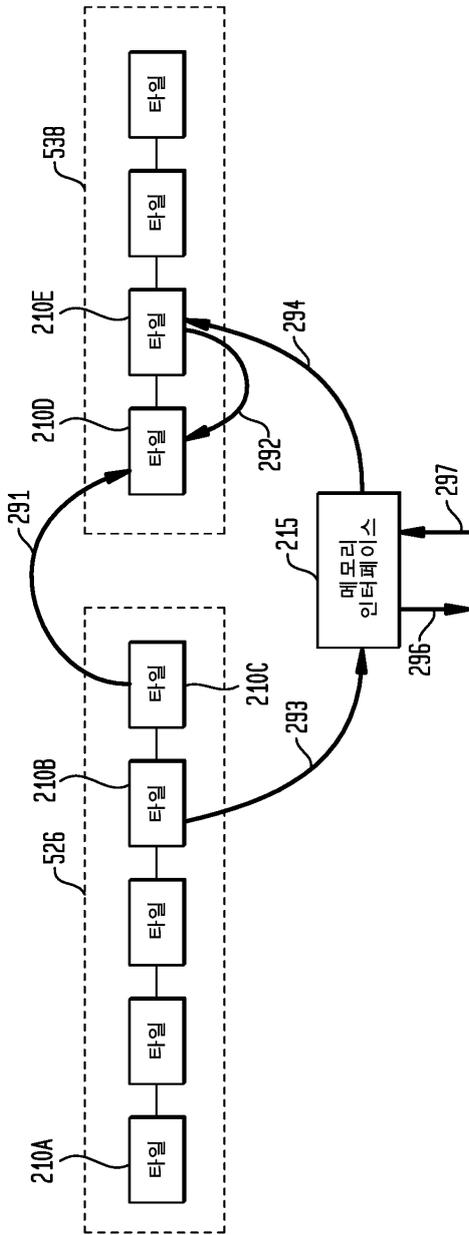
도면10



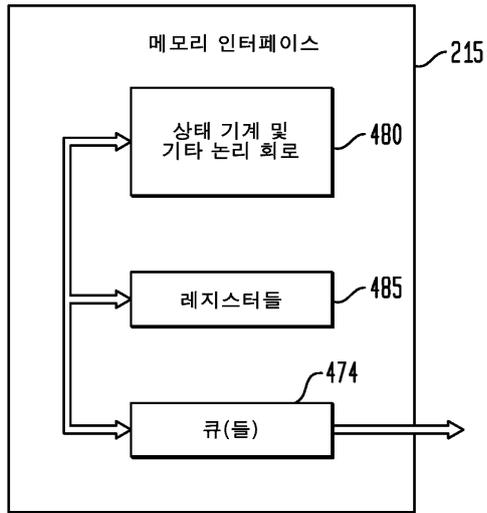
도면11



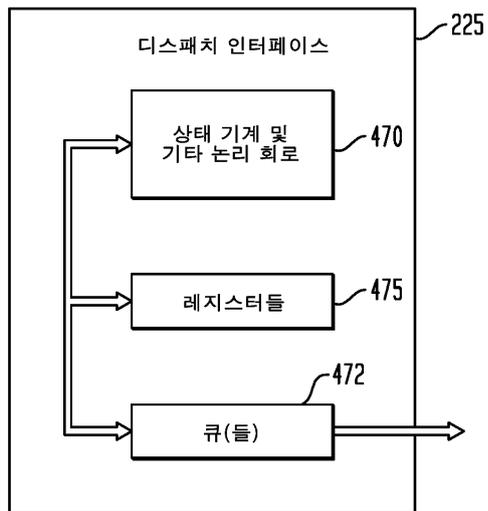
도면12



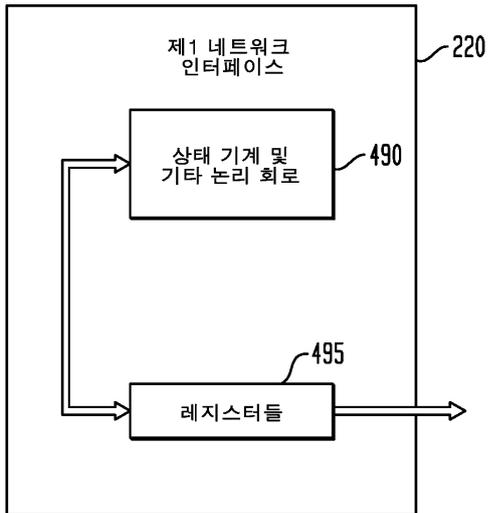
도면13



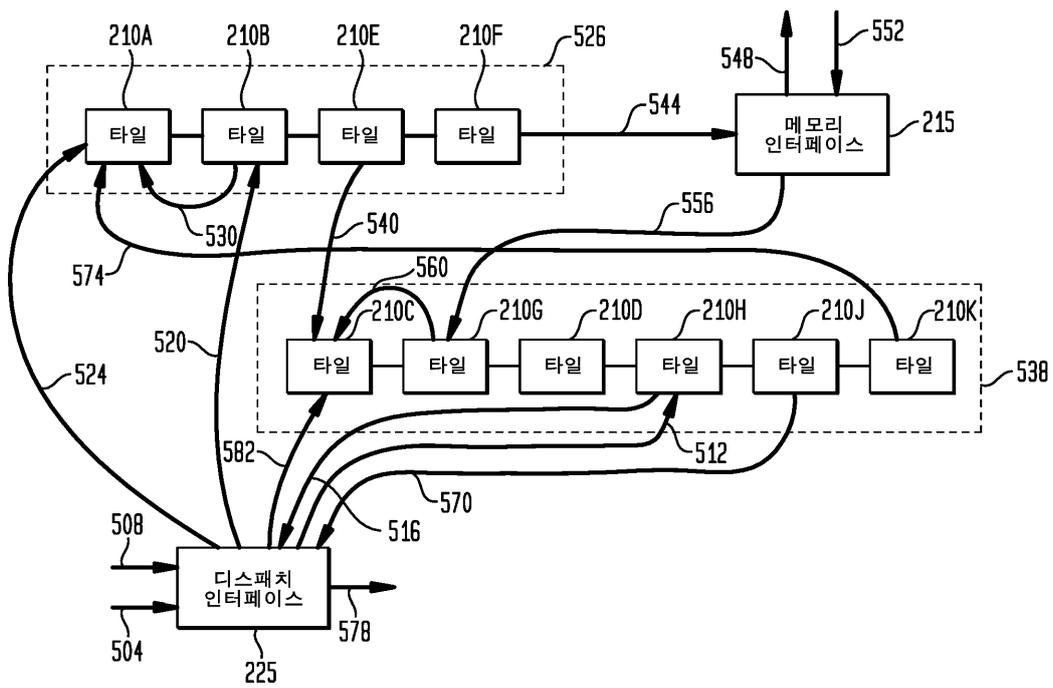
도면14



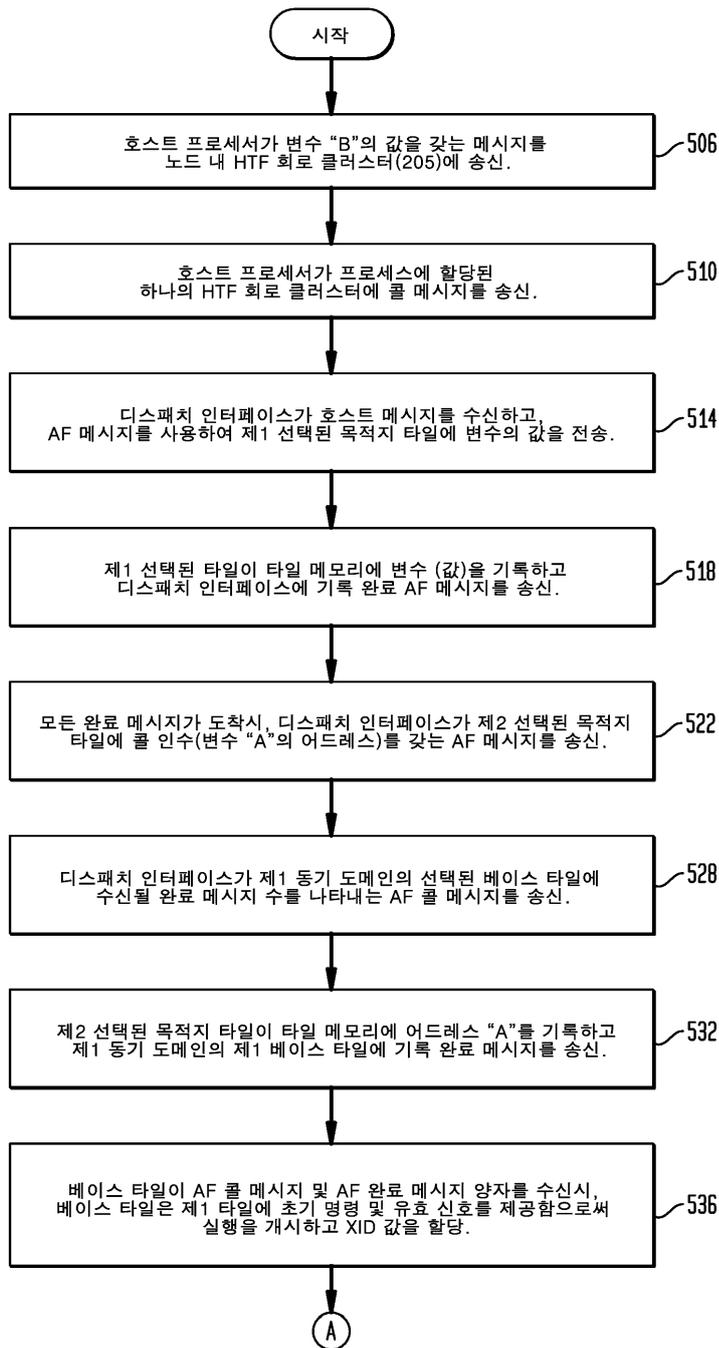
도면15



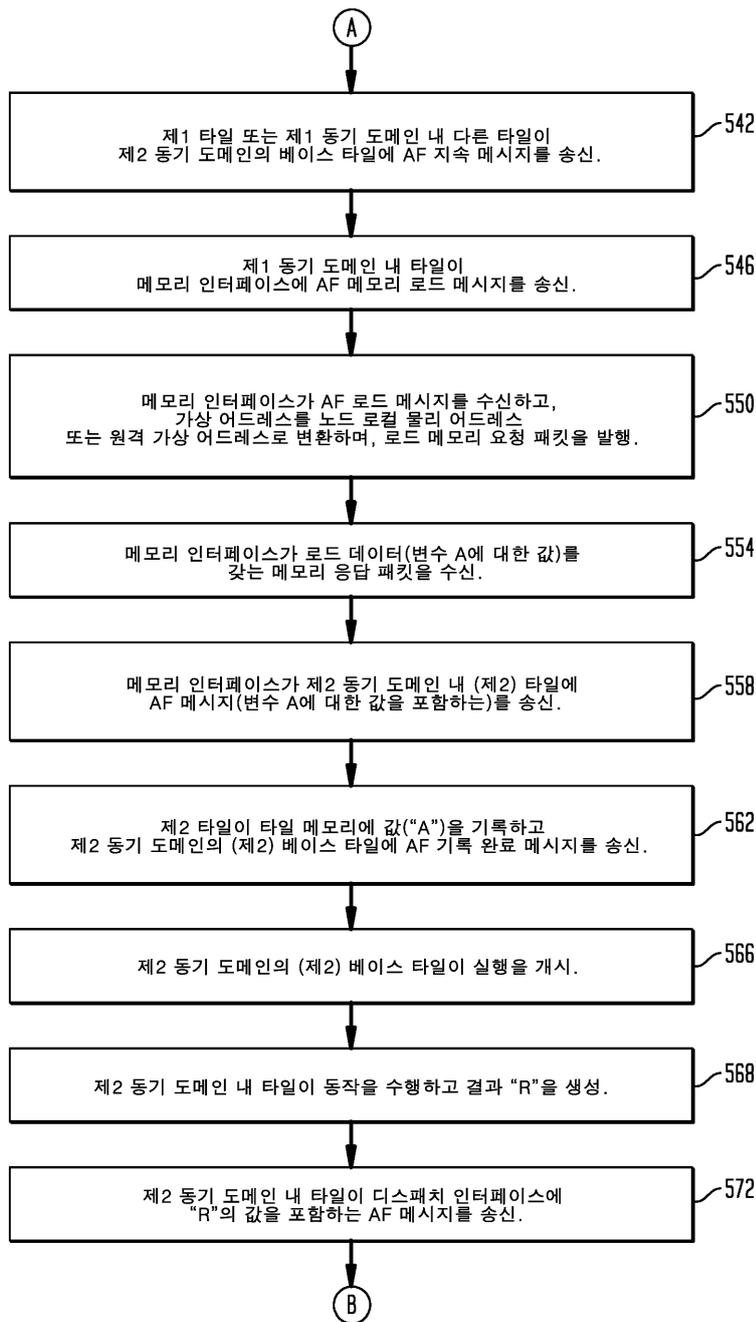
도면16



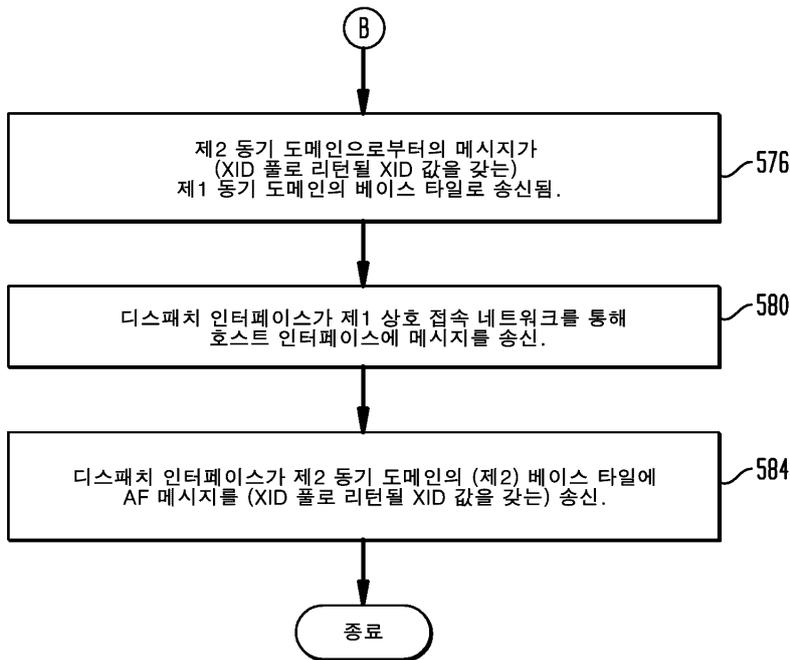
도면17a



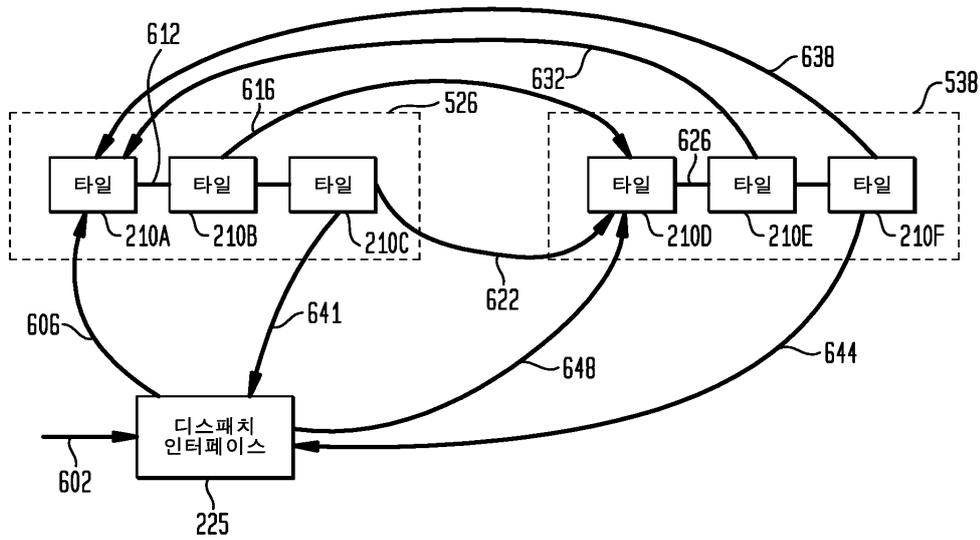
도면17b



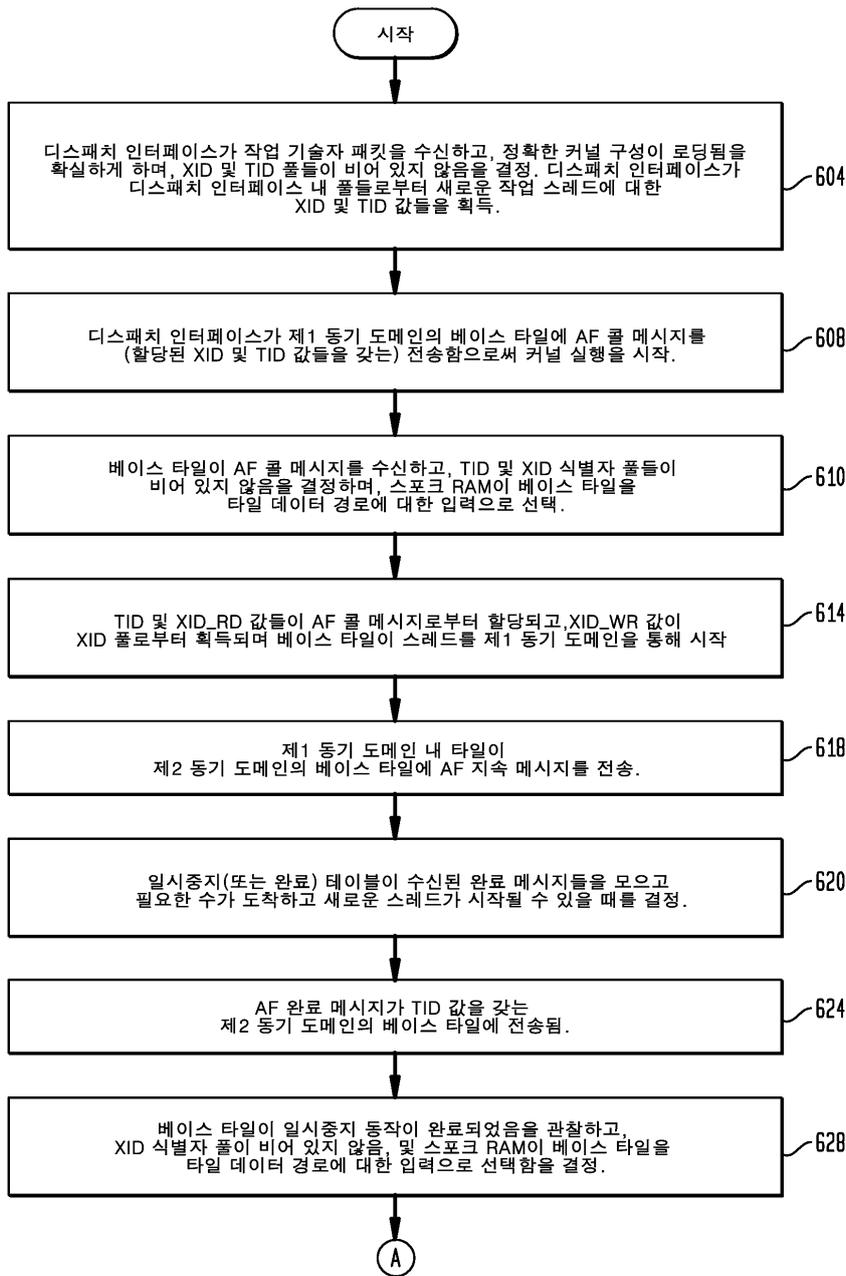
도면17c



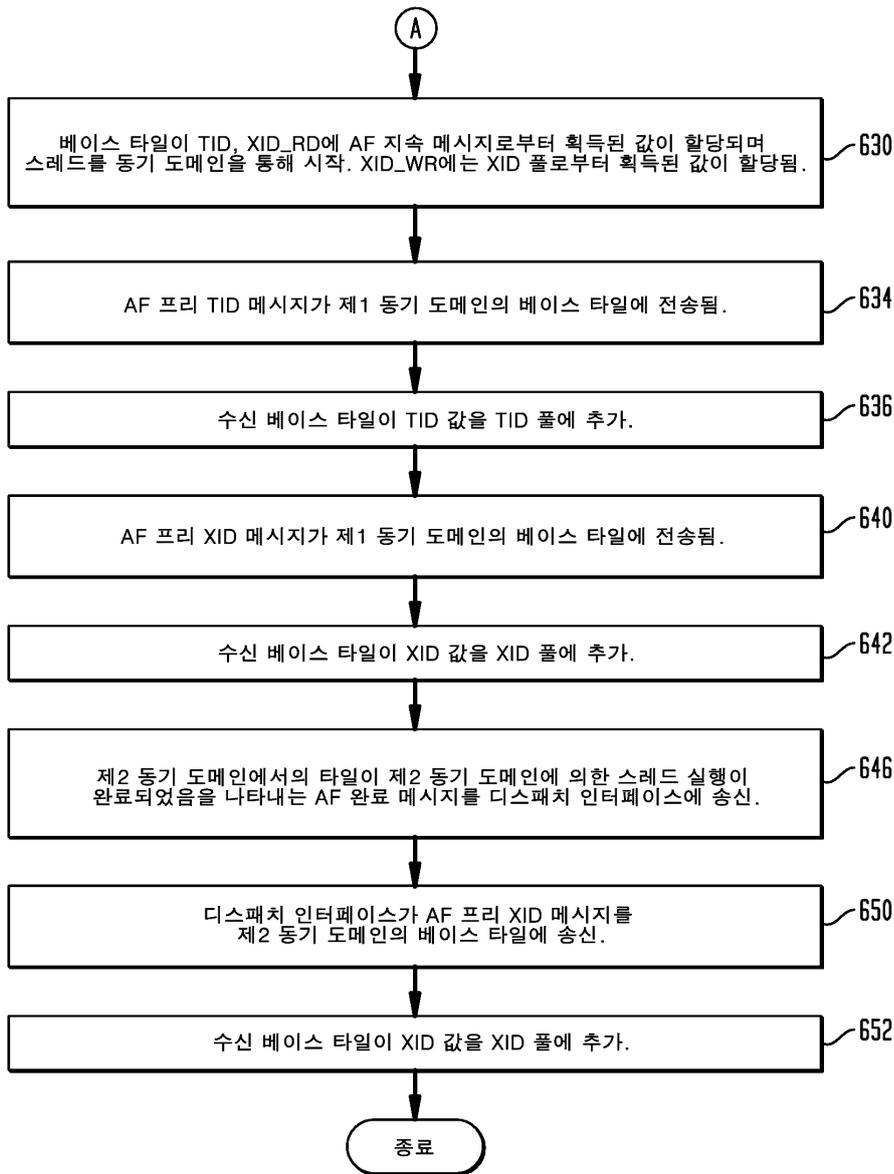
도면18



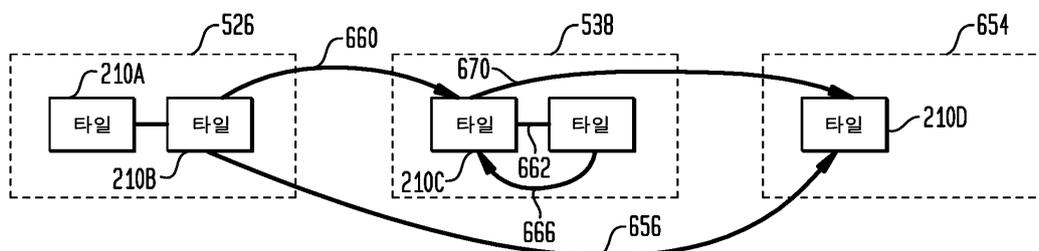
도면19a



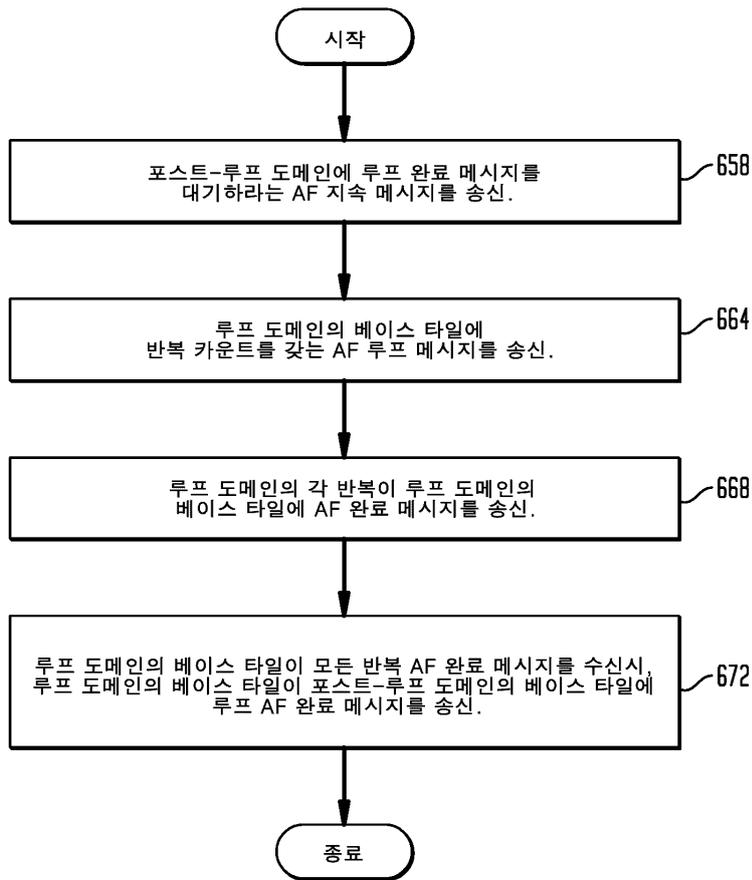
도면19b



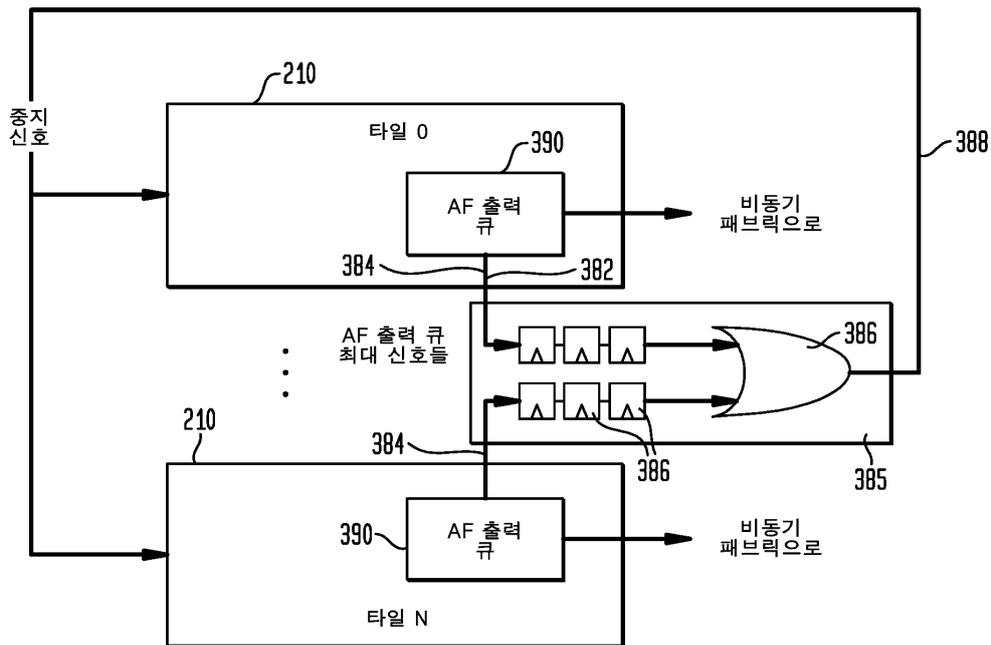
도면20



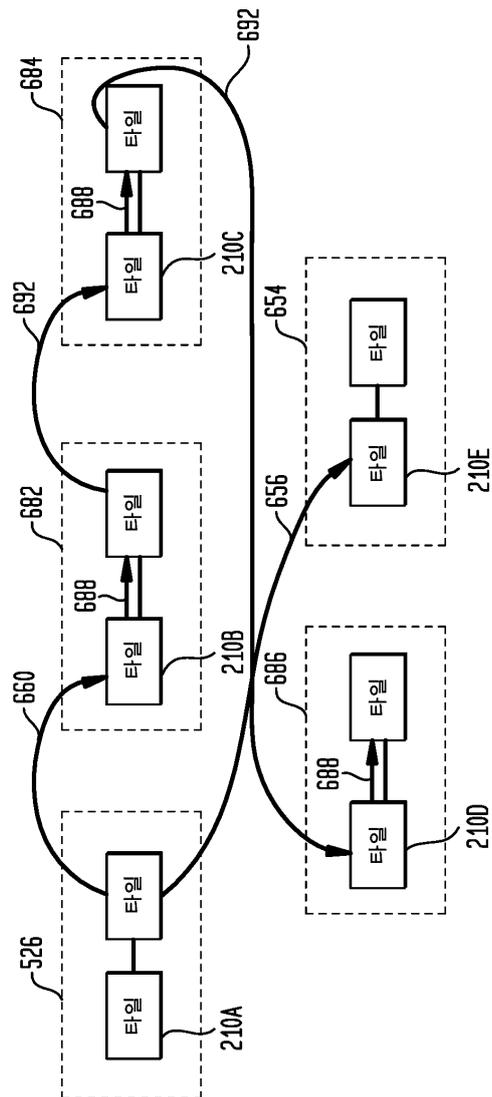
도면21



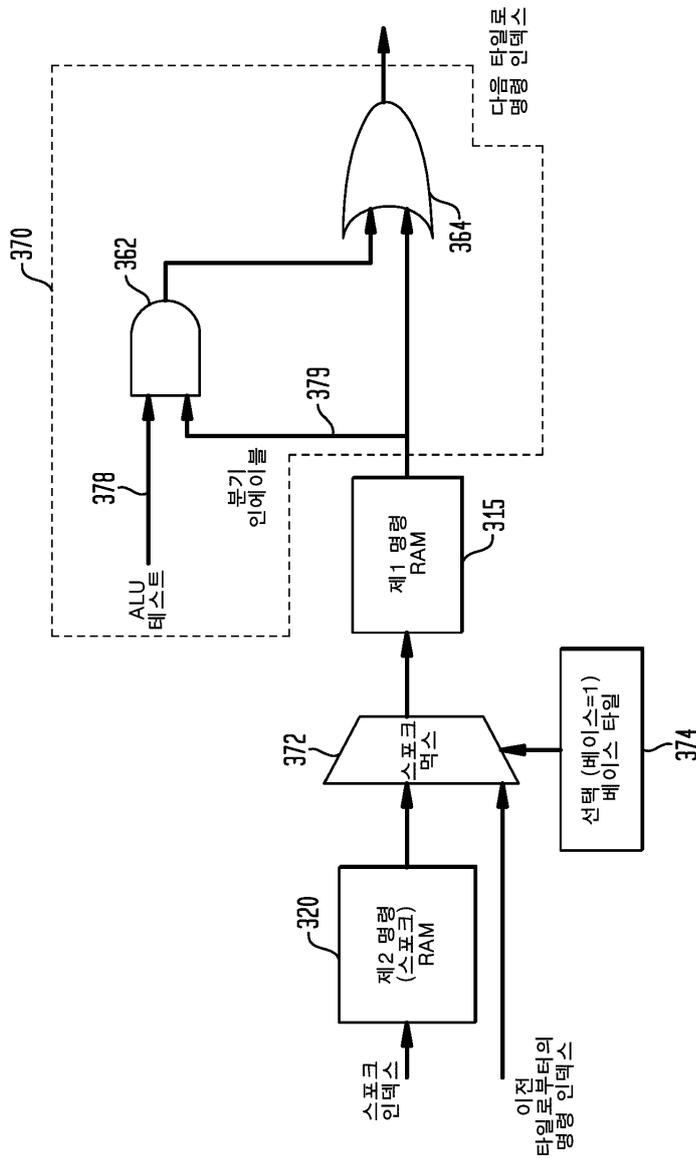
도면22



도면23



도면24



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 12

【변경전】

제5항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에 상기 복수의 유형의 스레드 식별자들 중 각각의 스레드 식별자에 대한 대응하는 데이터 완료 카운트를 더 저장하고, 상기 스레드 제어 회로는, 상기 복수의 상이한 스레드 중 각각의 스레드의 상기 스레드 식별자에 대해, 임의의 데이터 종속성들의 완료를 나타내기 위해 상기 대응하는 데이터 완료 카운트가 0으로 감소했을 때, 미리 결정된 케이던스 시에 실행할 복수의 상이한 스레드를 자체 스케줄링 및 큐잉하도록 적응되는, 구성가능한 회로.

【변경후】

제5항에 있어서, 상기 복수의 제어 레지스터는 상기 완료 테이블에 상기 복수의 유형의 스레드 식별자들 중 각각의 스레드 식별자에 대한 대응하는 데이터 완료 카운트를 더 저장하고, 상기 스레드 제어 회로는, 상기 복수의 상이한 스레드 중 각각의 스레드의 상기 스레드 식별자에 대해, 임의의 데이터 종속성들의 완료를 나타내기 위해 상기 대응하는 데이터 완료 카운트가 0으로 감소했을 때, 미리 결정된 케이던스 시에 실행할 복수의 상이한 스레드를 자체 스케줄링 및 큐잉하도록 적응되는, 구성가능한 회로.