



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0134713  
(43) 공개일자 2016년11월23일

(51) 국제특허분류(Int. Cl.)  
*G06F 9/54* (2006.01) *G06F 12/02* (2006.01)

(71) 출원인  
퀄컴 인코포레이티드  
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(52) CPC특허분류

*G06F 9/54* (2013.01)

(72) 벌명자  
부르드 알렉세이 블라디미로비치  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(21) 출원번호 10-2016-7027632

사크하르쉐테 스와프닐 프라딥쿠마르

(22) 출원일자(국제) 2015년03월05일

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

심사청구일자 없음

쉬 폐이

(85) 번역문제출일자 2016년10월05일

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(86) 국제출원번호 PCT/US2015/018987

스 페이

(87) 국제공개번호 WO 2015/142538

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

국제공개일자 2015년09월24일

(74) 대리인

(30) 우선권주장

14/219,696 2014년03월19일 미국(US)

특허법인코리아나

전체 청구항 수 : 총 30 항

(54) 발명의 명칭 태스크 간 통신을 지원하는 하드웨어-기반의 원자 동작들

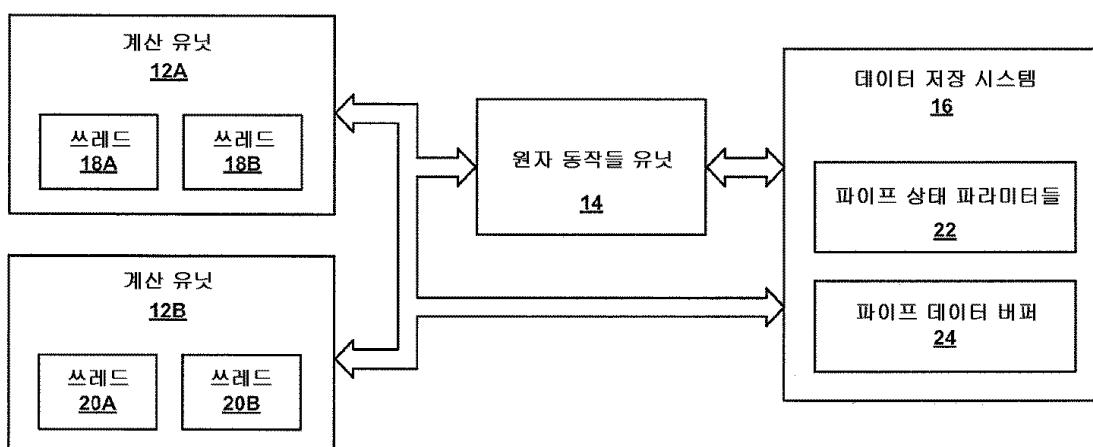
### (57) 요 약

본 개시물은 병렬 컴퓨팅 시스템에서 태스크 간 통신을 지원하는 기법들을 기술한다. 태스크 간 통신을 지원하는 기법들은 파이프의 상태를 유지하기 위해 하드웨어-기반의 원자 동작들을 이용할 수도 있다. 파이프는 여러 태스크들이 데이터 제작자들 또는 데이터 소비자들로서 버퍼와 상호작용 가능하게 하는 선입선출 (FIFO)-편

(뒷면에 계속)

### 대 표 도

10 ↗



성된 버퍼를 지칭할 수도 있다. 여러 파일 구현 예들은 다수의 상태 파라미터들을 이용하여, 파일의 상태를 정의할 수도 있다. 본 개시물에서 설명된 하드웨어-기반의 원자 동작들은 다수의 파일 상태 파라미터들을 원자 방식으로 수정할 수도 있다. 다수의 파일 상태 파라미터들을 원자 방식으로 수정하는 것은 그렇지 않다면, 다수의 제작자들 및/또는 다수의 소비자들이 파일의 상태를 동시에 수정하려고 시도할 때 일어나는 경합 조건들을 피할 수도 있다. 이러한 방법으로, 파일-기반의 태스크 간 통신은 병렬 컴퓨팅 시스템에서 지원될 수도 있다.

## (52) CPC특허분류

*G06F 9/526* (2013.01)

*G06F 9/546* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

하나 이상의 프로세서들에 의해, 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 관독-수정-기록 동작을 수행하는 단계를 포함하며,

상기 파이프는 태스크 간 통신을 지원하는 데이터 버퍼에 대응하는, 방법.

#### 청구항 2

제 1 항에 있어서,

상기 파이프 상태 파라미터들은 상기 파이프의 예약 상태를 나타내는 하나 이상의 파이프 예약 상태 파라미터들을 포함하는, 방법.

#### 청구항 3

제 2 항에 있어서,

상기 예약 상태 파라미터들은, 얼마나 많은 비커밋된 예약들이 상기 파이프에 현재 펜딩중인지를 나타내는 파라미터, 및 얼마나 많은 패킷 엔트리들이 상기 파이프에 현재 예약되어 있는지를 나타내는 파라미터 중 적어도 하나를 포함하는, 방법.

#### 청구항 4

제 1 항에 있어서,

상기 하나 이상의 프로세서들에 의해, 상기 하드웨어-기반의 원자 관독-수정-기록 동작을 수행하는 단계는, 인터럽트불가능한 동작들의 세트를 수행하는 단계를 포함하며,

상기 인터럽트불가능한 동작들의 세트는,

메모리 서브시스템으로부터 상기 파이프 상태 파라미터들에 대응하는 값들을 관독하는 동작;

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작; 및

상기 파이프 상태 파라미터들에 대한 상기 수정된 값들을 상기 메모리 서브시스템에 기록하는 동작을 포함하는, 방법.

#### 청구항 5

제 4 항에 있어서,

상기 2개 이상의 파이프 상태 파라미터들은 제 1 파이프 상태 파라미터 및 제 2 파이프 상태 파라미터를 포함하며,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은,

제 1 수정 기능에 기초하여 상기 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작; 및

제 2 수정 기능에 기초하여 상기 제 2 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작을 포함하며,

상기 제 2 수정 기능은 상기 제 1 수정 기능과는 상이한, 방법.

#### 청구항 6

제 4 항에 있어서,

상기 2개 이상의 파이프 상태 파라미터들은 제 1 파이프 상태 파라미터 및 제 2 파이프 상태 파라미터를 포함하

며,

상기 메모리 서브시스템으로부터 상기 파이프 상태 파라미터들에 대응하는 값들을 관독하는 동작은 제 1 파이프 상태 파라미터에 대응하는 제 1 값을 관독하는 동작을 포함하며,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은 상기 제 1 파이프 상태 파라미터에 대응하는 상기 제 1 값에 기초하여 상기 제 2 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작을 포함하는, 방법.

### 청구항 7

제 4 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은,

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 상기 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 결정하는 동작; 및

상기 파이프가 상기 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있다고 결정하는 것에 응답하여, 기록 예약이 상기 파이프에 대해 이루어졌다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 상기 수정된 값을 발생시키는 동작을 포함하는, 방법.

### 청구항 8

제 4 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값을 발생시키는 동작은,

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 상기 파이프가 가지는지 여부를 결정하는 동작; 및

패킷 데이터가 상기 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 상기 파이프가 가지고 있다고 결정하는 것에 응답하여, 판독 예약이 상기 파이프에 대해 이루어졌다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 수정된 값을 발생시키는 동작을 포함하는, 방법.

### 청구항 9

제 4 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값을 발생시키는 동작은,

상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하기 전에 상기 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 상기 파이프가 가지고 있다는 것을 상기 제 1 파이프 상태 파라미터에 대한 상기 수정된 값이 반영하도록, 상기 파이프 상태 파라미터 값들의 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작;

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 상기 파이프에 대한 모든 예약들이 커밋되었는지 여부를 결정하는 동작; 및

상기 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 커밋된 엔트리들이 상기 파이프에 배치되었다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 상기 수정된 값을 발생시키는 동작을 포함하는, 방법.

### 청구항 10

제 4 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값을 발생시키는 동작은,

상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하기 전에 상기 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 상기 파이프가 가지고 있다는 것을 상기 제 1 파이프 상태

파라미터에 대한 상기 수정된 값이 반영하도록, 상기 파이프 상태 파라미터 값들의 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작;

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 상기 파이프에 대한 모든 예약들이 커밋되었는지 여부를 결정하는 동작; 및

상기 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 커밋된 엔트리들이 상기 파이프로부터 제거되었다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 수정된 값을 발생시키는 동작을 포함하는, 방법.

## 청구항 11

제 1 항에 있어서,

상기 방법은, 상기 하나 이상의 프로세서들에 포함되는 하드웨어 유닛에 의해, 원자 동작을 수행해달라는 제 1 쓰레드로부터의 제 1 요청을 수신하는 단계를 더 포함하며,

상기 하드웨어-기반의 원자 판독-수정-기록 동작은 제 1 하드웨어-기반의 원자 판독-수정-기록 동작이며,

상기 하나 이상의 프로세서들에 의해, 상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 단계는, 상기 제 1 요청을 수신하는 것에 응답하여, 상기 하드웨어 유닛에 의해, 상기 제 1 요청을 서비스하기 위해 상기 제 1 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 단계를 더 포함하며,

상기 방법은,

상기 제 1 하드웨어-기반의 원자 판독-수정-기록 동작을 완료하기 전에, 상기 하드웨어 유닛에 의해, 원자 동작을 수행해달라는 제 2 쓰레드로부터의 제 2 요청을 수신하는 단계;

상기 제 2 요청을 수신하는 것에 응답하여, 상기 하드웨어 유닛에 의해, 상기 제 1 하드웨어-기반의 원자 판독-수정-기록 동작의 수행이 완료될 때까지, 상기 제 2 요청을 서비스하기 위한 제 2 하드웨어-기반의 원자 판독-수정-기록 동작의 수행을 개시하는 것을 대기하는 단계; 및

상기 제 1 하드웨어-기반의 원자 판독-수정-기록 동작의 수행이 완료된 후, 상기 하드웨어 유닛에 의해, 상기 제 2 요청을 서비스하기 위해 2개 이상의 파이프 상태 파라미터들을 수정하는 상기 제 2 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 단계를 더 포함하는, 방법.

## 청구항 12

제 1 항에 있어서,

상기 하나 이상의 프로세서들은 그래픽 프로세싱 유닛 (GPU) 을 포함하며,

상기 하나 이상의 프로세서들에 의해, 상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 단계는 상기 GPU 에 의해, 상기 파이프의 상태를 나타내는 상기 2개 이상의 파이프 상태 파라미터들을 수정하는 상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 단계를 포함하는, 방법.

## 청구항 13

제 1 항에 있어서,

상기 파이프 상태 파라미터들은 얼마나 많은 패킷들이 상기 파이프에 현재 저장되어 있는지를 나타내는 파라미터, 상기 파이프에 대한 시작 패킷을 저장하는 패킷 엔트리를 나타내는 파라미터, 얼마나 많은 비커밋된 예약들이 상기 파이프에 현재 펜딩중인지를 나타내는 파라미터, 및 얼마나 많은 패킷 엔트리들이 상기 파이프에 현재 예약되어 있는지를 나타내는 파라미터를 포함하는, 방법.

## 청구항 14

제 1 항에 있어서,

상기 데이터 버퍼는 태스크 간 통신을 지원하는 선입선출 (FIFO; First-In-First-Out)-편성된 데이터 버퍼를 포함하는, 방법.

**청구항 15**

파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 구성된 하나 이상의 프로세서들을 포함하며,

상기 파이프는 태스크 간 통신을 지원하는 선입선출 (FIFO)-편성된 데이터 버퍼에 대응하는, 디바이스.

**청구항 16**

제 15 항에 있어서,

상기 파이프 상태 파라미터들은 상기 파이프의 예약 상태를 나타내는 하나 이상의 파이프 예약 상태 파라미터들을 포함하는, 디바이스.

**청구항 17**

제 16 항에 있어서,

상기 예약 상태 파라미터들은 얼마나 많은 비커밋된 예약들이 상기 파이프에 현재 펜딩중인지를 나타내는 파라미터, 및 얼마나 많은 패킷 엔트리들이 상기 파이프에 현재 예약되어 있는지를 나타내는 파라미터 중 적어도 하나를 포함하는, 디바이스.

**청구항 18**

제 15 항에 있어서,

상기 하나 이상의 프로세서들에 의해, 상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 것은, 인터럽트불가능한 동작들의 세트를 수행하는 것을 포함하며,

상기 인터럽트불가능한 동작들의 세트는,

메모리 서브시스템으로부터 상기 파이프 상태 파라미터들에 대응하는 값들을 판독하는 동작;

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작; 및

상기 파이프 상태 파라미터들에 대한 상기 수정된 값들을 상기 메모리 서브시스템에 기록하는 동작을 포함하는, 디바이스.

**청구항 19**

제 18 항에 있어서,

상기 2개 이상의 파이프 상태 파라미터들은 제 1 파이프 상태 파라미터 및 제 2 파이프 상태 파라미터를 포함하며,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은,

제 1 수정 기능에 기초하여 상기 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작; 및

제 2 수정 기능에 기초하여 상기 제 2 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작을 포함하며,

상기 제 2 수정 기능은 상기 제 1 수정 기능과는 상이한, 디바이스.

**청구항 20**

제 18 항에 있어서,

상기 2개 이상의 파이프 상태 파라미터들은 제 1 파이프 상태 파라미터 및 제 2 파이프 상태 파라미터를 포함하며,

상기 메모리 서브시스템으로부터 상기 파이프 상태 파라미터들에 대응하는 값들을 판독하는 동작은 제 1 파이프 상태 파라미터에 대응하는 제 1 값을 판독하는 동작을 포함하며,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은, 상기 제 1 파이프 상태 파라미터에 대응하는 상기 제 1 값에 기초하여 상기 제 2 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작을 포함

하는, 디바이스.

### 청구항 21

제 18 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은,

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 상기 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 결정하는 동작; 및

상기 파이프가 상기 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있다고 결정하는 것에 응답하여, 기록 예약이 상기 파이프에 대해 이루어졌다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 상기 수정된 값들을 발생시키는 동작을 포함하는, 디바이스.

### 청구항 22

제 18 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은,

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 상기 파이프가 가지는지 여부를 결정하는 동작; 및

패킷 데이터가 상기 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 상기 파이프가 가지고 있다고 결정하는 것에 응답하여, 판독 예약이 상기 파이프에 대해 이루어졌다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작을 포함하는, 디바이스.

### 청구항 23

제 18 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은,

상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하기 전에 상기 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 상기 파이프가 가지고 있다는 것을 상기 제 1 파이프 상태 파라미터에 대한 상기 수정된 값이 반영하도록, 상기 파이프 상태 파라미터 값들의 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작;

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 상기 파이프에 대한 모든 예약들이 커밋되었는지 여부를 결정하는 동작; 및

상기 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 커밋된 엔트리들이 상기 파이프에 배치되었다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 상기 수정된 값들을 발생시키는 동작을 포함하는, 디바이스.

### 청구항 24

제 18 항에 있어서,

상기 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작은,

상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하기 전에 상기 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 상기 파이프가 가지고 있다는 것을 상기 제 1 파이프 상태 파라미터에 대한 상기 수정된 값이 반영하도록, 상기 파이프 상태 파라미터 값들의 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작;

상기 파이프 상태 파라미터들 중 하나 이상에 기초하여 상기 파이프에 대한 모든 예약들이 커밋되었는지 여부를 결정하는 동작; 및

상기 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 커밋된 엔트리들이 상기 파이프로부터

터 제거되었다는 것을 상기 수정된 값들이 반영하도록, 상기 파이프 상태 파라미터들에 대한 수정된 값을 빌생시키는 동작을 포함하는, 디바이스.

### 청구항 25

제 15 항에 있어서,

상기 하나 이상의 프로세서들은 그래픽 프로세싱 유닛 (GPU) 을 포함하며,

상기 하나 이상의 프로세서들은 또한, 상기 파이프의 상태를 나타내는 상기 2개 이상의 파이프 상태 파라미터들을 수정하는 상기 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 구성되는, 디바이스.

### 청구항 26

제 15 항에 있어서,

상기 파이프 상태 파라미터들은 얼마나 많은 패킷들이 상기 파이프에 현재 저장되어 있는지를 나타내는 파라미터, 상기 파이프에 대한 시작 패킷을 저장하는 패킷 엔트리를 나타내는 파라미터, 얼마나 많은 비커밋된 예약들이 상기 파이프에 현재 펜딩중인지를 나타내는 파라미터, 및 얼마나 많은 패킷 엔트리들이 상기 파이프에 현재 예약되어 있는지를 나타내는 파라미터를 포함하는, 디바이스.

### 청구항 27

제 15 항에 있어서,

상기 데이터 버퍼는 태스크 간 통신을 지원하는 선입선출 (FIFO)-편성된 데이터 버퍼를 포함하는, 디바이스.

### 청구항 28

제 15 항에 있어서,

상기 디바이스는, 무선 통신 디바이스 및 모바일 폰 핸드셋 중 적어도 하나를 포함하는, 디바이스.

### 청구항 29

파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 저장하는 수단으로서, 상기 파이프는 태스크 간 통신을 지원하는 선입선출 (FIFO)-편성된 데이터 버퍼에 대응하는, 상기 저장하는 수단; 및

상기 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 수단을 포함하는, 장치.

### 청구항 30

명령들을 저장하고 있는 비일시적 컴퓨터-판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 하나 이상의 프로세서들로 하여금,

파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 하며,

상기 파이프는 태스크 간 통신을 지원하는 선입선출 (FIFO)-편성된 데이터 버퍼에 대응하는, 비일시적 컴퓨터-판독가능 저장 매체.

## 발명의 설명

### 기술 분야

[0001]

본 개시물은 컴퓨팅 시스템들, 좀더 구체적으로는, 병렬 컴퓨팅 시스템들에 관한 것이다.

### 배경 기술

[0002]

병렬 프로그래밍 모델들은 계산 문제들을 해결하기 위해 태스크-병렬성 및 데이터-병렬성 중 하나 (또는, 양쪽) 을 지원할 수도 있다. 태스크-병렬성은 계산 문제들이 다수의 태스크들로 분할될 수 있도록 할 수도 있다.

태스크들은 하나 이상의 프로세서 코어들 상에서 순차적으로, 동시에, 및/또는 병렬로 실행될 수도 있다. 데이터-병렬성은, 데이터를 상이한 프로세싱 엘리먼트들에 분산시키고 프로세싱 엘리먼트들의 각각으로 하여금 그들의 할당된 데이터의 세트에 대해 동일한 동작들의 세트를 수행가능하게 함으로써, 동일한 동작들의 세트가 상이한 데이터의 세트들 상에서 병렬로 수행될 수 있도록 할 수도 있다.

[0003] 멀티-코어 프로세서들은 각각의 코어가 특정의 태스크를 실행하도록 구성되는 태스크-병렬성을 지원하기 위해 사용될 수도 있다. 일부의 경우, 멀티-코어 프로세서에서의 코어들 중 하나 이상은 데이터-병렬성을 지원하기 위해 다수의 프로세싱 엘리먼트들을 포함할 수도 있는 단일 명령, 다수의 데이터 (SIMD) 프로세서 또는 단일 프로그램, 다수의 데이터 (SPMD) 프로세서일 수도 있다. 이러한 경우, 데이터-레벨 병렬성을 지원하는 태스크들은 멀티-코어 프로세서 상에서 순차적으로 또는 병렬로 실행될 수도 있다.

[0004] 멀티-코어 중앙 처리 유닛 (CPU), 그래픽 프로세싱 유닛 (GPU), 디지털 신호 프로세서 (DSP), 셀 광대역 엔진 (Cell/B.E.) 프로세서, 등을 포함한, 여러 상이한 유형들의 프로세서들은 태스크-병렬성 및/또는 데이터-병렬성을 지원할 수도 있다. GPU들은 전통적으로 디스플레이에의 3차원 (3D) 그래픽스의 렌더링을 지원하도록 설계되었지만, 많은 현대의 GPU들에 포함되는 프로그래밍가능 세이더 아키텍처가 병렬 프로그래밍 모델을 이용하여 프로그래밍되는 범용, 비-그래픽스 특정의 프로그램들에서 발견되는 태스크-병렬성 및 데이터-병렬성 양쪽을 효율적으로 지원하기 위해 사용될 수도 있다. 비-그래픽스 특정의 프로그램들을 실행하기 위해 GPU의 병렬 아키텍처를 이용하는 것은 GPGPU (general-purpose computing on graphics processing units)로서 지칭될 수도 있다.

## 발명의 내용

### 과제의 해결 수단

[0005] 본 개시물은 병렬 컴퓨팅 시스템에서 태스크 간 통신을 지원하는 기법들을 기술한다. 태스크 간 통신을 지원하는 기법들은 파이프의 상태를 유지하기 위해 하드웨어-기반의 원자 동작들을 이용할 수도 있다. 파이프는 여러 태스크들이 데이터 제작자들 또는 데이터 소비자들로서 버퍼와 상호작용가능하게 하는 버퍼를 지칭할 수도 있다. 여러 파이프 구현예들은 파이프의 상태를 정의하기 위해 다수의 상태 파라미터들을 이용할 수도 있으며, 이를 상태 파라미터들은 데이터 제작자 또는 소비자가 파이프와 상호작용할 때마다 수정될 필요가 있을 수도 있다. 본 개시물에서 설명된 하드웨어-기반의 원자 동작들은 다수의 파이프 상태 파라미터들을 원자 방식으로 수정할 수도 있다. 다수의 파이프 상태 파라미터들을 원자 방식으로 수정하는 것은 그렇지 않다면, 다수의 제작자들 및/또는 다수의 소비자들이 파이프의 상태를 동시에 수정하려고 시도할 때 일어나는 경합 조건들을 피할 수도 있다. 이러한 방법으로, 파이프-기반의 태스크 간 통신은 병렬 컴퓨팅 시스템에서 지원될 수도 있다.

[0006] 일 예에서, 본 개시물은 하나 이상의 프로세서들에 의해, 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 단계를 포함하는 방법을 기술한다. 파이프는 태스크 간 통신을 지원하는 데이터 버퍼에 대응한다.

[0007] 다른 예에서, 본 개시물은 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 구성된 하나 이상의 프로세서들을 포함하는 디바이스를 기술한다. 파이프는 태스크 간 통신을 지원하는 데이터 버퍼에 대응한다.

[0008] 다른 예에서, 본 개시물은 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 저장하는 수단을 포함하는 장치를 기술한다. 파이프는 태스크 간 통신을 지원하는 데이터 버퍼에 대응한다. 본 장치는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하는 수단을 더 포함한다.

[0009] 다른 예에서, 본 개시물은 실행될 때, 하나 이상의 프로세서들로 하여금, 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 하는 명령들을 저장하는 비일시적 컴퓨터-판독가능 저장 매체를 기술한다. 파이프는 태스크 간 통신을 지원하는 데이터 버퍼에 대응한다.

[0010] 본 개시물의 하나 이상의 예들의 세부 사항들은 첨부도면 및 아래의 설명에서 개시된다. 본 개시물의 다른 특징들, 목적들, 및 이점들은 설명 및 도면들로부터, 그리고 청구범위로부터 명백히 알 수 있을 것이다.

## 도면의 간단한 설명

[0011]

도 1 은 본 개시물의 하드웨어-기반의 원자 동작들을 구현하는 예시적인 컴퓨팅 시스템을 예시하는 블록도이다.

도 2 는 본 개시물에 따른, 예시적인 원자 예약 기록 지령의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다.

도 3 은 본 개시물에 따른, 예시적인 원자 커밋 기록 지령 (atomic commit write command) 의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다.

도 4 는 본 개시물에 따른, 예시적인 원자 예약 판독 지령의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다.

도 5 는 본 개시물에 따른, 예시적인 원자 커밋 판독 지령의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다.

도 6 은 도 1 의 원자 동작들 유닛의 일 예가 더욱더 자세하게 예시된 도 1 의 예시적인 병렬 컴퓨팅 시스템의 블록도이다.

도 7 은 본 개시물에 따른, 데이터 저장 시스템과 상호작용하는 예시적인 원자 동작 데이터경로를 예시하는 개념도이다.

도 8 은 본 개시물의 하드웨어-기반의 원자 동작들을 구현하는데 사용될 수도 있는 예시적인 컴퓨팅 디바이스를 예시하는 블록도이다.

도 9 는 본 개시물에 따른, 그래픽 프로세싱 유닛 (GPU) 또는 다른 계산 디바이스에서 사용될 수도 있는 예시적인 계산 유닛을 예시하는 블록도이다.

도 10 은 예시적인 GPU 및 메모리 구성이 더욱더 자세하게 예시된 도 1 의 컴퓨팅 시스템의 일부를 예시하는 블록도이다.

도 11 은 본 개시물에 따른, 하드웨어-기반의 원자 동작들을 수행하는 예시적인 기법을 예시하는 흐름도이다.

도 12 는 본 개시물에 따른, 하드웨어-기반의 원자 동작들을 수행하는 다른 예시적인 기법을 예시하는 흐름도이다.

도 13 은 본 개시물에 따른, 원자 예약 기록 동작 및 원자 예약 판독 동작을 수행하는 예시적인 기법을 예시하는 흐름도이다.

도 14 는 본 개시물에 따른, 원자 커밋 기록 동작 및 원자 커밋 판독 동작을 수행하는 예시적인 기법을 예시하는 흐름도이다.

## 발명을 실시하기 위한 구체적인 내용

[0012]

본 개시물은 병렬 컴퓨팅 시스템에서 태스크 간 통신을 지원하는 기법들을 기술한다. 태스크 간 통신을 지원하는 기법들은 파일프의 상태를 유지하기 위해 하드웨어-기반의 원자 동작들을 이용할 수도 있다. 파일프는 여러 태스크들이 데이터 제작자들 또는 데이터 소비자들로서 버퍼와 상호작용 가능하게 하는 선입선출 (FIFO)-편성된 버퍼를 지칭할 수도 있다. 여러 파일프 구현예들은 파일프의 상태를 정의하기 위해 다수의 상태 파라미터들을 이용할 수도 있으며, 이를 상태 파라미터들은 데이터 제작자 또는 소비자가 파일프와 상호작용할 때마다 수정될 필요가 있을 수도 있다. 본 개시물에서 설명된 하드웨어-기반의 원자 동작들은 다수의 파일프 상태 파라미터들을 원자 방식으로 수정할 수도 있다. 다수의 파일프 상태 파라미터들을 원자 방식으로 수정하는 것은 그렇지 않다면, 다수의 제작자들 및/또는 다수의 소비자들이 파일프의 상태를 동시에 수정하려고 시도할 때 일어날 수도 있는 경합 조건들을 피할 수도 있다. 이러한 방법으로, 파일프-기반의 태스크 간 통신은 병렬 컴퓨팅 시스템에서 지원될 수도 있다.

[0013]

계산 문제는 하나 이상의 프로세서 코어들 상에서 순차적으로, 동시에, 및/또는 병렬로 실행되는 다수의 태스크들로 분할될 수도 있다. 태스크는 병렬 컴퓨팅 시스템에 의해 실행되는 프로그램을 지칭할 수도 있다. 쓰레드는 병렬 컴퓨팅 시스템에서 단일 프로세싱 엘리먼트에 의해 실행되는 태스크의 인스턴스를 지칭할 수도 있다. 태스크는 대안적으로 본원에서 커널로서 지칭될 수도 있으며, 쓰레드는 대안적으로 본원에서 작업-아이템으로서 지칭될 수도 있다. 일부의 경우, 병렬 컴퓨팅 시스템에 의해 실행되는 상이한 태스크들은 서로

통신할 필요가 있을 수도 있다. 예를 들어, 하나의 태스크는 다른 태스크에 의해 발생된 데이터에 기초하여 데이터를 발생시킬 수도 있다.

[0014] 파이프들은 상이한 태스크들이 서로 통신가능하게 하기 위해서 사용될 수도 있다. 프로그래머의 관점에서, 파이프는 여러 태스크들이 데이터 제작자들 또는 데이터 소비자들로서 버퍼와 상호작용가능하게 하는 FIFO-편성된 버퍼를 지칭할 수도 있다. FIFO-편성된 버퍼는 기록 인터페이스 및 판독 인터페이스가 FIFO 원리에 따라서 동작하는 버퍼를 지칭할 수도 있다. FIFO 원리에 따라서 동작하는 기록 인터페이스는 기록될 데이터를 버퍼에 이미 저장되어 있는 임의의 데이터의 끝에 첨부하는 인터페이스를 지칭할 수도 있다. FIFO 원리에 따라서 동작하는 판독 인터페이스는 버퍼에 기록되는 가장 빠른 시기에 미판독된 데이터를 선택하여 버퍼로부터 제거하는 인터페이스를 지칭할 수도 있다.

[0015] 파이프가 FIFO-편성된 버퍼를 구현할 수도 있기 때문에, 파이프는 또한 FIFO 원리에 따라서 동작하는 기록 및 판독 인터페이스들을 갖는다고 말할 수도 있다. 프로그래머의 관점으로부터, FIFO 원리에 따라서 동작하는 기록 인터페이스를 가지는 파이프에 대해, 데이터를 파이프에 기록하는 태스크는 데이터를 기록할 파이프에서 특정의 로케이션을 반드시 규정하지 않을 수도 있다. 대신, 파이프는 기록될 데이터를 파이프에 이미 저장되어 있는 임의의 데이터의 끝에 첨부할 수도 있다. 이와 유사하게, FIFO 원리에 따라서 동작하는 판독 인터페이스를 가지는 파이프에 대해, 파이프로부터 데이터를 판독하는 태스크는 데이터를 판독할 파이프에서 특정의 로케이션을 반드시 규정하지 않을 수도 있다. 대신, 파이프는 그 파이프에 기록되는 가장 이른 시기에 미판독된 데이터였던 파이프에 저장된 데이터를 선택하여 반환할 수도 있다. FIFO 원리에 따라서 동작하는 판독 및 기록 인터페이스들을 제공함으로써, 파이프는 프로그래머들이 데이터를 저장하는데 사용되는 메모리의 내부 조직과 관련될 필요 없이 제작자 및 소비자 태스크들이 서로 효율적으로 통신할 수도 있는 추상 데이터 구조를 제공할 수도 있다.

[0016] 본 개시물에서 설명되는 파이프들은 일부 예들에서, 데이터를 파이프에 기록하고 그로부터 판독하기 위한 데이터의 기본적인 유닛으로서 데이터 패킷들을 이용할 수도 있다. 공백이 아닌 파이프는 하나 이상의 패킷들의 시퀀스를 포함할 수도 있다. 파이프에 대한 시작 패킷은 파이프에 저장되는 패킷들의 시퀀스에서 제 1 패킷에 대응하는 패킷을 지칭할 수도 있다. 파이프에 대한 종료 패킷은 파이프에 저장되는 패킷들의 시퀀스에서 최종 패킷에 대응하는 패킷을 지칭할 수도 있다. 공백인 파이프는 어떠한 패킷들도 포함하지 않을 수도 있다.

[0017] 파이프는 패킷 엔트리들의 각각이 단일 데이터 패킷을 저장하는 것이 가능한, 데이터 패킷들을 저장하기 위한 복수의 패킷 엔트리들을 포함할 수도 있다. 비-공백 파이프에 대한 패킷들의 시퀀스는 그 파이프에 대한 인접한 패킷 엔트리들의 세트에 저장될 수도 있다. 일부의 경우, 패킷 엔트리들은 메모리 공간에 저장될 수도 있으며 인접한 패킷 엔트리들의 세트는 메모리 공간의 끝으로부터 메모리 공간의 시작까지 순환(wrap-around) 할 수도 있다. 비-공백 파이프에 대한 시작 패킷 엔트리는 파이프의 시작 패킷을 저장하는 패킷 엔트리를 지칭할 수도 있다. 비-공백 파이프에 대한 종료 패킷 엔트리는 파이프의 종료 패킷을 저장하는 패킷 엔트리를 지칭할 수도 있다. 제 1 패킷 오프셋 파라미터는 비-공백 파이프에 대한 시작 패킷 엔트리를 나타내는 값일 수도 있다. 공백 파이프에 대해, 제 1 패킷 오프셋 파라미터는 데이터가 파이프에 기록될 때 데이터가 기록될 제 1 패킷 엔트리를 나타내는 값일 수도 있다.

[0018] 여러 파이프 구현예들은 파이프의 상태를 정의하기 위해 다수의 상태 파라미터들을 이용할 수도 있으며, 이들 상태 파라미터들은 데이터 제작자 또는 소비자가 파이프와 상호작용할 때마다 수정될 필요가 있을 수도 있다.

병렬 컴퓨팅 시스템에서의 병렬성 때문에, 파이프는 각각이 파이프와 상호작용하여 파이프 상태 파라미터들을 동시에 수정하려고 잠재적으로 시도할 수도 있는 다수의 데이터 제작자들 및/또는 다수의 데이터 소비자들을 지원하는 것을 필요로 할 수도 있다. 다수의 데이터 제작자들 및/또는 소비자들이 다수의 파이프 상태 파라미터들을 잠재적으로 동일한 시간에 수정가능하게 하는 것은 경합 조건들을 초래할 수도 있으며, 이것은 파이프가 적절히 동작하는 것을 방해할 수도 있다.

[0019] 본 개시물의 일부 양태들에 따르면, 파이프의 상태를 유지하기 위해 하드웨어-기반의 원자 동작들을 수행하는 기법들이 기술된다. 예를 들어, 프로세서에 포함되는 하드웨어 유닛은 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 구성될 수도 있다. 2개 이상의 파이프 상태 파라미터들을 원자 방식으로 수정하기 위해 하드웨어-기반의 원자 판독-수정-기록 동작을 이용하는 것은, 그렇지 않다면, 다수의 데이터 제작자들 및/또는 소비자들이 파이프의 상태를 동시에 수정하려고 시도할 때 일어날 경합 조건들을 피할 수도 있다. 이러한 방법으로, 적합한 기능이 파이프

의 상태를 정의하기 위해 다수의 상태 파라미터들을 이용하는 파이프들에 대해 달성될 수도 있다.

[0020] 파이프의 상태를 정의하기 위해 다수의 상태 파라미터들을 이용하는 파이프의 일 예는 링 버퍼 (즉, 원형 버퍼)로서 구현되는 파이프이다. 링 버퍼로서 구현되는 파이프는 링 버퍼로서 편성되는 파이프 엔트리들을 가질 수도 있다. 링 버퍼는 마치 버퍼가 단-대-단 접속된 것처럼, 단일, 고정된-사이즈 버퍼를 이용하는 데이터 구조를 지칭할 수도 있다. 다시 말해서, 고정된-사이즈 버퍼의 반대 단부들에 로케이트된 패킷 엔트리들은 링 버퍼를 구현하기 위해 서로 인접한 것으로 개념적으로 간주될 수도 있다.

[0021] 링 버퍼가 고정된 시작 패킷 엔트리 및 고정된 종료 패킷 엔트리를 가지지 않기 때문에, 링 버퍼로서 구현되는 파이프의 상태를 완전히 정의하기 위해 다수의 상태 파라미터들이 일반적으로 요구된다. 예를 들어, 제 1 패킷 오프셋 파라미터는 파이프의 시작 패킷 엔트리를 가리킬 수도 있으며, 패킷 카운트 파라미터는 파이프에 포함된 패킷들의 개수를 나타낼 수도 있다. 다른 예로서, 제 1 패킷 오프셋 파라미터는 파이프의 시작 패킷 엔트리를 가리킬 수도 있으며 최종 패킷 오프셋 파라미터는 파이프의 최종 패킷 엔트리를 가리킬 수도 있다.

각각의 시간 데이터가 링 버퍼로서 구현되는 파이프에 추가되거나 또는 그로부터 제거되며, 상태 파라미터들의 각각은 업데이트될 필요가 있을 수도 있다. 따라서, 다수의 파이프 상태 파라미터들은 파이프가 링 버퍼로서 구현될 때 업데이트될 필요가 있을 수도 있다.

[0022] 파이프의 상태를 정의하기 위해 다수의 상태 파라미터들을 이용하는 파이프의 다른 예는 예약들을 지원하는 파이프이다. 파이프 예약들은 제작자 쓰레드가 파이프에서 인접한 패킷 엔트리들의 세트를 예약하고 기록 가능하게 할 수도 있으며, 소비자 쓰레드가 파이프에서 인접한 패킷 엔트리들의 세트를 예약하고 판독 가능하게 할 수도 있다. 파이프 예약들은 다수의 제작자들이 하나 보다 많은 패킷을 파이프에 동시에 기록하려고 시도하거나 및/또는 다수의 소비자들이 파이프로부터 하나 보다 많은 패킷을 동시에 판독하려고 시도할 때 일어날 수도 있는 인터리빙 문제들을 경감하기 위해 사용될 수도 있다.

[0023] 일반적으로, 다수의 제작자들 및/또는 다수의 소비자들이 하나 보다 많은 패킷을 파이프에 기록하거나 및/또는 하나 보다 많은 패킷을 파이프로부터 판독하려고 시도하는 문제는 파이프가 제작자들 또는 소비자들로서 그 파이프에 속박되는 다수의 태스크들을 지원하는 상황에서 일어날 수도 있다. 그 문제는 또한 단일 태스크가 제작자 또는 소비자로서 그 파이프에 속박되고 태스크의 실행이 병렬로 실행되는 복수의 쓰레드들 (즉, 태스크의 인스턴스들)에 분산되는 상황에서 일어날 수도 있다. 이것은 예를 들어, 데이터-병렬성을 지원하는 병렬 컴퓨팅 시스템 (예컨대, 단일 명령, 다수의 데이터 (SIMD) 실행 유닛 및/또는 단일 프로그램, 다수의 데이터 (SPMD) 실행 유닛을 포함하는 병렬 컴퓨팅 시스템)에서 일어날 수도 있다.

[0024] 예약들을 지원하는 파이프를 사용하기 위해, 제작자는 파이프에서 인접한 패킷 엔트리들의 세트를 기록용으로 예약하고, 패킷 데이터를 예약된 패킷 엔트리들에 기록하고, 그리고 예약된 패킷 엔트리들을 파이프에 커밋할 수도 있다. 이와 유사하게, 소비자는 파이프에서 인접한 패킷 엔트리들의 세트를 판독용으로 예약하고, 예약된 패킷 엔트리들로부터 패킷 데이터를 판독하고, 그리고 예약된 패킷 엔트리들을 파이프에 커밋할 수도 있다.

[0025] 다수의 제작자들이 파이프에 대한 패킷 엔트리들을 동시에 예약하거나 및/또는 예약된 패킷 엔트리들에 동시에 기록할 수도 있기 때문에, 다수의 기록 예약들이 파이프에 대해 주어진 시간에 미처리될 (즉, 커밋되지 않을) 수도 있다. 이러한 경우, 파이프는 데이터 패킷들이 파이프의 예약된 패킷 엔트리들에 기록되어 있는 것을 반영하기 위해 파이프 상태 파라미터들을 업데이트하기 전에, 모든 미처리된 기록 예약들이 커밋될 때까지 대기 할 수도 있다.

[0026] 이와 유사하게, 다수의 소비자들이 파이프에 대한 패킷 엔트리들을 동시에 예약하거나 및/또는 예약된 패킷 엔트리들을 동시에 판독할 수도 있기 때문에, 다수의 판독 예약들이 파이프에 대해 주어진 시간에 미처리될 (즉, 커밋되지 않을) 수도 있다. 이러한 경우, 파이프는 예약된 패킷 엔트리들에 포함되는 데이터 패킷들이 파이프로부터 판독하여 제거된 것을 반영하기 위해 파이프 상태 파라미터들을 업데이트하기 전에 모든 미처리된 판독 예약들이 커밋될 때까지 대기할 수도 있다.

[0027] 예약들을 지원하는 파이프의 상태를 유지하기 위해, 하나 이상의 파이프 상태 파라미터들이 파이프의 예약 상태를 정의하기 위해 요구될 수도 있다. 이들 파이프 상태 파라미터들은 예약 상태 파라미터들로서 지칭될 수도 있다. 예를 들어, 제 1 예약 상태 파라미터는 얼마나 많은 미처리된 예약들이 그 파이프에 존재하는지를 나타낼 수도 있으며, 제 2 예약 상태 파라미터는 얼마나 많은 패킷 엔트리들이 그 파이프에 예약되어 있는지를 나타낼 수도 있다. 패킷 엔트리들이 파이프에 예약되거나 및/또는 커밋될 때마다, 예약 상태 파라미터들은

업데이트될 필요가 있을 수도 있다. 게다가, 모든 미처리된 예약들이 파일에 커밋될 때, 예약 상태 파라미터들 및 다른 파일 상태 파라미터들이 업데이트될 필요가 있을 수도 있다. 따라서, 다수의 파일 상태 파라미터들은 패킷 엔트리들이 예약들을 지원하는 파일로부터 예약되거나 및/또는 그에 커밋될 때 업데이트될 필요가 있을 수도 있다.

[0028] 위에서 설명한 바와 같이, 파일가 링 버퍼로서 편성될 때 및/또는 파일가 예약들을 지원할 때, 다수의 파일 상태 파라미터들이 파일의 상태를 정의하기 위해 요구될 수도 있다. 병렬 컴퓨팅 시스템에서의 병렬성 때문에, 파일는 각각이 파일과 상호작용하여 파일 상태 파라미터들을 동시에 수정하려고 잠재적으로 시도 할 수도 있는 다수의 데이터 제작자들 및/또는 다수의 데이터 소비자들을 지원하는 것을 필요로 할 수도 있다.

다수의 데이터 제작자들 및/또는 소비자들이 다수의 파일 상태 파라미터들을 잠재적으로 동일한 시간에 수정 가능하게 하는 것은 경합 조건들을 초래할 수도 있으며, 이것은 파일가 적절히 동작하는 것을 방해할 수도 있다.

[0029] 본 개시물의 일부 양태들에 따르면, 파일의 상태를 유지하기 위해 하드웨어-기반의 원자 동작들을 수행하는 기법들이 기술된다. 예를 들어, 프로세서에 포함되는 하드웨어 유닛은 파일의 상태를 나타내는 2개 이상의 파일 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 구성될 수도 있다. 2개 이상의 파일 상태 파라미터들을 원자 방식으로 수정하기 위해 하드웨어-기반의 원자 판독-수정-기록 동작을 이용하는 것은, 그렇지 않다면, 다수의 데이터 제작자들 및/또는 소비자들이 파일의 상태를 동시에 수정하려고 시도할 때 일어날 경합 조건들을 피할 수도 있다. 이러한 방법으로, 적합한 기능이 예컨대, 링 버퍼들로서 구현되는 파일들 및/또는 예약들을 지원하는 파일들과 같은, 다수의 상태 파라미터들을 이용하는 파일들에 대해 달성될 수도 있다.

[0030] 다수의 파일 상태 파라미터들을 업데이트할 때 경합 조건들을 피하는 다른 솔루션은, 파일의 상태를 수정하려는 쓰레드에 의한 시도들을 그 쓰레드에 대한 컴파일된 코드의 중요 섹션에 배치하고, 코드의 중요 섹션을 강화하기 위해 뮤텍스 락들을 이용하는 것을 수반한다. 뮤텍스 락들 (mutex locks)은 어떤 2개의 쓰레드들이 그들의 중요 섹션들에서 동시에 실행하고 있지 않다는 것을 보장할 수도 있으며, 따라서 2개의 쓰레드들이 파일 상태 파라미터들을 동시에 수정할 수 없도록 보장한다. 뮤텍스 락들은 예컨대, 비교-교환 원자 동작들을 이용함으로써 구현될 수도 있다. 뮤텍스 락들이 사용될 때, 그러나, 많은 쓰레드들이 그 락에 대해 경합할 수도 있으며, 그에 따라서, 성공적인 락의 획득을 위해 다수의 시도들이 이루어지도록 초래할 수도 있다. 락을 획득하려는 쓰레드들에 의한 반복된 시도들은 성능 및/또는 전력 효율을 감소시킬 수도 있다.

[0031] 본 개시물에서 설명된 하드웨어-기반의 원자 동작들은, 그러나, 2개의 쓰레드들이 파일의 파일 상태 파라미터들을 동시에 수정할 수 없도록 보장할 수도 있으며, 뮤텍스 락들의 사용을 필요로 함이 없이 이러한 거동을 달성할 수도 있다. 예를 들어, 본 개시물의 하드웨어-기반의 원자 동작들을 구현하는 하드웨어 유닛이 파일에 대한 다수의 파일 상태 파라미터들을 수정하기 위해 다수의 쓰레드들로부터 다수의 요청들을 수신할 때, 하드웨어 유닛은, 제 1 쓰레드로부터의 제 1 요청과 연관된 모든 파일 상태 파라미터 수정들이 제 2 쓰레드로부터의 제 2 요청과 연관된 파일 상태 파라미터 수정들 중 임의의 파일 상태 파라미터 수정을 수행하기 전에 완료되도록, 그 요청들을 순차적으로 프로세싱할 수도 있다. 이러한 하드웨어 유닛을 이용하는 것은 다수의 동시에 실행하는 쓰레드들이 다수의 파일 상태 파라미터들을 동시에 수정하려고 시도하는 병렬 컴퓨팅 시스템들에서 적합한 파일 기능이 달성될 수 있도록 할 수도 있다. 이러한 방법으로, 파일-기반의 태스크 간 통신이, 파일 상태 파라미터들을 수정하는 소프트웨어-기반, 상호 제외 기법들을 이용하는 것과 연관된 성능 단점들 및/또는 소비 전력 단점들을 강요함이 없이, 병렬 컴퓨팅 시스템에서 제공될 수도 있다.

[0032] 일부 예들에서, 본 개시물에서 설명된 하드웨어-기반의 원자 동작들은 파일의 상태를 나타내는 2개 이상의 파일 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 포함할 수도 있다. 판독-수정-기록 동작은, 그 동작이 판독-수정-기록 동작을 요청한 태스크 또는 쓰레드를 동시에 실행하고 있는 다른 태스크들 및/또는 쓰레드들에 의해 요청될 수도 있는 임의의 다른 판독-수정-기록 동작들로부터의 인터럽트 없이 시작부터 끝까지 수행된다는 점에서 볼때 원자일 수도 있다. 다시 말해서, 원자 판독-수정-기록 동작의 실행 동안, 어떤 다른 쓰레드들 및/또는 태스크들도 판독-수정-기록 동작에 의해 수정되는 파일 상태 파라미터들을 판독하거나 또는 수정불가능할 수도 있다. 이러한 방법으로, 2개 이상의 태스크들 및/또는 쓰레드들이 파일의 파일 상태 파라미터들을 동시에 수정하려고 시도하는 경우에 경합 조건들이 회피될 수도 있다.

[0033] 원자 판독-수정-기록 동작은 원자 판독-수정-기록 동작이 하드웨어 유닛에 통신가능하게 커플링된 쓰레드들 및/또는 태스크들에 의해 호출될 수도 있는 단일, 불가분 동작으로서 하드웨어 유닛에 의해 구현될 수도 있다는 점

에서 하드웨어-기반일 수도 있다. 단일, 불가분 동작은 하드웨어 유닛이 판독-수정-기록 동작을 요청한 태스크 또는 쓰레드를 동시에 실행하고 있는 다른 태스크들 및/또는 쓰레드들에 의해 요청될 수도 있는 다른 판독-수정-기록 동작들을 수행하는 동작을 인터럽트함이 없이 시작부터 끝까지 그 동작을 수행한다는 점에서 불가분일 수도 있다. 예를 들어, 하드웨어 유닛이 원자 판독-수정-기록 동작을 수행해달라는 다수의 요청들을 수신하면, 하드웨어 유닛은 하나의 원자 판독-수정-기록 동작이 한꺼번에 수행되도록, 그리고 각각의 원자 판독-수정-기록 동작의 실행이 다른 원자 판독-수정-기록 동작을 수행하도록 시작하기 전에 완료되도록, 요청들의 각각에 대해 원자 판독-수정-기록 동작을 순차적으로 수행할 수도 있다. 하드웨어-기반의 원자 판독-수정-기록 동작들을 이용함으로써, 태스크들 및/또는 쓰레드들이 뮤텍스 락들을 이용할 필요 없이 동시에 실행하는 태스크들 및/또는 쓰레드들에 의해 다수의 파이프 상태 파라미터들이 수정될 수도 있다.

[0034] 일부 예들에서, 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하기 위해, 하드웨어 유닛은 인터럽트불가능한 동작들의 세트를 수행하도록 구성될 수도 있다. 예를 들어, 인터럽트불가능한 동작들의 세트는 파이프의 파이프 상태 파라미터들에 대응하는 값들을 메모리 서브시스템으로부터 판독하는 동작, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작, 및 파이프 상태 파라미터들에 대한 수정된 값들을 메모리 서브시스템에 기록하는 동작을 포함할 수도 있다. 동작들의 세트는 그 동작들이 다른 판독-수정-기록 동작들 (예컨대, 현재 실행하는 판독-수정-기록 동작을 요청한 태스크 또는 쓰레드를 동시에 실행하고 있는 태스크들 및/또는 쓰레드들에 의해 요청되는 다른 판독-수정 기록 동작들) 과 연관될 수도 있는 다른 동작들을 수행하는 실행을 인터럽트함이 없이 시작부터 끝까지 수행된다는 점에서 비-인터럽트가능할 수도 있다. 하드웨어 유닛으로 하여금 인터럽트불가능한 판독-수정-기록 시퀀스를 이러한 방식으로 수행하도록 하는 것은, 하드웨어 유닛으로 하여금 이러한 동작들을 요청할 수도 있는 동시에 실행하는 쓰레드들 및/또는 태스크들에 대해 원자 판독-수정-기록 동작들을 구현가능하게 한다.

[0035] 본 개시물의 일부 양태들에 따르면, 본 개시물에 따라서 하드웨어-기반의 원자 판독-수정-기록 동작들을 구현하는 하드웨어 유닛은 단일, 원자 판독-수정-기록 동작의 일부로서 상이한 파이프 상태 파라미터들에 대해 상이한 수정 동작들을 수행할 수도 있다. 예를 들어, 하드웨어 유닛은 제 1 수정 기능에 기초하여 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 제 1 수정 동작, 및 제 2 수정 기능에 기초하여 제 2 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 제 2 수정 동작을 수행할 수도 있다. 제 2 수정 기능은 제 1 수정 기능과는 상이할 수도 있다. 상이한 파이프 상태 파라미터들을 단일, 원자 판독-수정-기록 동작의 부분으로서 수정하기 위해 상이한 수정 기능들을 이용하는 것은 상이한 파이프 상태 파라미터들이 단일 원자 동작에서 상이한 방법들로 업데이트될 수 있도록 할 수도 있다.

[0036] 예를 들어, 원자 파이프 예약 동작을 지원하기 위해, 제 1 수정 기능이 예약된 패킷 카운트 파라미터를 파이프 예약용으로 요청되는 패킷들의 개수 만큼 증분시키는데 사용될 수도 있으며, 제 2 수정 기능이 예약 카운트를 1의 상수 값 만큼 증분시키는데 사용될 수도 있다. 다른 예로서, 원자 파이프 커밋 동작을 지원하기 위해, 제 1 수정 기능이 예약 카운트를 1의 상수 값 만큼 감분시키는데 사용될 수도 있으며, 제 2 수정 기능이 패킷 카운트 파라미터를 예약 카운트 파라미터 만큼 증분시키거나 또는 감분시키는데 사용될 수도 있으며, 제 3 수정 기능이 예약된 패킷 카운트 파라미터를 제로로 리셋시키기 위해 사용될 수도 있다. 다른 예들이 가능하며 본 개시물의 범위 이내이다.

[0037] 본 개시물의 추가적인 양태들에 따르면, 본 개시물에 따라서 하드웨어-기반의 원자 판독-수정-기록 동작들을 구현하는 하드웨어 유닛은 원자 동작의 실행 동안 메모리 서브시스템으로부터 획득되는 하나 이상의 추가적인 파이프 상태 파라미터들의 값에 기초하여 하나 이상의 파이프 상태 파라미터들의 값을 수정할 수도 있다. 예를 들어, 하드웨어 유닛은, 단일 판독-수정-기록 동작 동안, 파이프의 제 1 및 제 2 파이프 상태 파라미터들에 각각 대응하는 제 1 및 제 2 값을 판독하고, 그리고 제 2 파이프 상태 파라미터에 대한 수정된 값이 제 1 파이프 상태 파라미터에 대응하는 제 1 값에 기초하여 발생되도록, 제 1 및 제 2 파이프 상태 파라미터들에 대한 수정된 값을 발생시킬 수도 있다. 단일, 원자 판독-수정-기록 동작 동안 상이한 파이프 상태 파라미터 값에 기초하여 파이프 상태 파라미터 값을 수정하는 것은 파이프의 현재의 상태에 기초하여 파이프 상태 파라미터들이 단일 원자 동작으로 업데이트될 수 있도록 할 수도 있다.

[0038] 예를 들어, 원자 커밋 동작을 지원하기 위해, 패킷 카운트 파라미터는 커밋 동작이 판독 커밋 동작인지 또는 기록 커밋 동작인지 여부에 따라서, 예약된 패킷 카운트 파라미터에 의해 감분되거나 또는 증분될 수도 있다. 다른 예로서, 원자 커밋 판독 동작을 지원하기 위해, 제 1 패킷 오프셋 파라미터는 예약된 패킷 카운트 파라미터에 기초하여 조정될 수도 있다. 다른 예들이 가능하며 본 개시물의 범위 이내이다.

[0039]

본 개시물의 추가적인 양태들에 따르면, 상이한 파이프 상태 파라미터 값에 기초하여 파이프 상태 파라미터 값을 수정하는 것은 상이한 파이프 상태 파라미터 값에 기초하여 파이프 상태 파라미터 값을 조건부로 업데이트하는 것을 수반할 수도 있다. 파이프 상태 파라미터 값을 조건부로 업데이트하는 것은 상이한 파이프 상태 파라미터 값에 기초하여 다른 파이프 상태 파라미터 값들을 조건부로 업데이트하는 것은 파이프 상태 파라미터들이 파이프의 현재의 상태에 기초하여 업데이트되거나 또는 업데이트되지 않도록 할 수도 있다.

[0040]

본 개시물의 추가적인 양태들에 따르면, 본 개시물에 따라서 하드웨어-기반의 원자 판독-수정-기록 동작들을 구현하는 하드웨어 유닛은 파이프 상태 파라미터 값들을, 요청하는 쓰레드 및/또는 태스크로 반환할 수도 있다.

일부의 경우, 반환된 파이프 상태 파라미터 값들은 하드웨어-기반의 원자 판독-수정-기록 동작에 의해 수행되는 수정들 전에 파이프의 상태를 반영할 수도 있다. 값을 요청하는 쓰레드 및/또는 태스크로 반환하는 것은, 쓰레드 또는 태스크로 하여금, 어느 패킷 엔트리들이 예약 동작용으로 예약되어 있는지를, 그리고 예약된 패킷 엔트리들이 파이프에 대한 패킷 엔트리들을 저장하는 메모리 공간에서 어디에 저장되어 있는지를 결정 가능할 수도 있다.

[0041]

일부 예들에서, 본 개시물의 하드웨어-기반의 원자 판독-수정-기록 동작들은 원자 예약 기록 동작을 포함할 수도 있다. 원자 예약 기록 동작은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 결정할 수도 있다. 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있다고 결정하는 것에 응답하여, 원자 예약 기록 동작은 기록 예약이 파이프에 대해 이루어졌다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0042]

추가적인 예들에서, 본 개시물의 하드웨어-기반의 원자 판독-수정-기록 동작들은 원자 예약 판독 동작을 포함할 수도 있다. 원자 예약 판독 동작은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가지는지 여부를 결정할 수도 있다. 파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가진다고 결정하는 것에 응답하여, 원자 예약 판독 동작은 판독 예약이 파이프에 대해 이루어졌다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0043]

추가적인 예들에서, 본 개시물의 하드웨어-기반의 원자 판독-수정-기록 동작들은 원자 커밋 기록 동작을 포함할 수도 있다. 원자 커밋 기록 동작은 원자 커밋 기록 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 수정된 값이 반영하도록, 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시킬 수도 있다. 원자 커밋 기록 동작은 또한 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프에 대한 모든 예약들이 커밋되었는지 여부를 결정할 수도 있다. 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 원자 커밋 기록 동작은 커밋된 엔트리들이 파이프에 배치되어 있다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시킬 수도 있다.

[0044]

추가적인 예들에서, 본 개시물의 하드웨어-기반의 원자 판독-수정-기록 동작들은 원자 커밋 판독 동작을 포함할 수도 있다. 원자 커밋 판독 동작은 원자 커밋 판독 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 수정된 값이 반영하도록, 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시킬 수도 있다. 원자 커밋 판독 동작은 또한 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프에 대한 모든 예약들이 커밋되었는지 여부를 결정할 수도 있다. 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 원자 커밋 판독 동작은 커밋된 엔트리들이 파이프로부터 제거되었다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시킬 수도 있다.

[0045]

본 개시물의 추가적인 양태들에 따르면, 단일, 원자 판독-수정-기록 동작 동안 판독되고 기록되는 다수의 파이프 상태 파라미터들을 저장하는데 사용되는 비트수는 단일 메모리 트랜잭션 동안 데이터 저장 유닛으로부터 판독되거나 또는 그에 기록되는 데이터의 블록에 포함되는 비트들의 수보다 적거나 또는 동일할 수도 있다. 예를 들어, 다수의 파이프 상태 파라미터들을 저장하는데 사용되는 비트수는 메모리 워드에서의 비트수 및/또는 캐시 라인에서의 비트수보다 적거나 또는 동일할 수도 있다. 일부의 경우, 메모리 서브시스템에 저장된 파

이프들의 각각에 대한 파이프 상태 파라미터들은 파이프에 대한 파이프 상태 파라미터들의 모두가 메모리 서브 시스템의 단일 메모리 워드 및/또는 단일 캐시 라인에 저장되도록, 정렬될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 판독 트랜잭션 또는 하나 보다 많은 메모리 기록 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정가능할 수도 있다.

[0046] 일부 예들에서, 파이프의 상태를 정의하는 파이프 상태 파라미터들은 얼마나 많은 패킷들이 현재 파이프에 저장되어 있는지를 나타내는 패킷 카운트 파라미터, 및 파이프에 대한 시작 패킷을 저장하는 패킷 엔트리를 나타내는 제 1 패킷 오프셋 파라미터를 포함할 수도 있다. 이들 파라미터들은 링 버퍼로서 구현되는 파이프의 상태를 정의하기 위해 사용될 수도 있다.

[0047] 추가적인 예들에서, 파이프의 상태를 정의하는 파이프 상태 파라미터들은 얼마나 많은 비커밋된 예약들이 현재 파이프에 펜딩 중인지를 나타내는 예약 카운트 파라미터, 및 얼마나 많은 패킷 엔트리들이 현재 파이프에 예약되어 있는지를 나타내는 예약된 패킷 카운트 파라미터를 포함할 수도 있다. 이들 파라미터들은 예약 상태 파라미터들로서 지칭될 수도 있으며, 예약들을 지원하는 파이프의 상태를 정의하기 위해 사용될 수도 있다.

[0048] 추가적인 예들에서, 파이프의 상태를 정의하는 파이프 상태 파라미터들은 패킷 카운트 파라미터, 제 1 패킷 오프셋 파라미터, 예약 카운트 파라미터, 및 예약된 패킷 파라미터를 포함할 수도 있다. 이 파이프 상태 파라미터들의 조합이 예약들을 지원하고 링 버퍼로서 구현될 수도 있는 파이프의 상태를 정의하기 위해 사용될 수도 있다.

[0049] 본 개시물의 하드웨어-기반의 원자 판독-수정-기록 동작들이 파이프 상태 파라미터들의 수정에 대해 주로 설명되지만, 동일한 또는 유사한 유형들의 동작들이 상이한 유형들의 데이터에 적용될 수도 있는 것으로 이해되어야 한다. 일반적으로, 본 개시물의 하드웨어-기반의 원자 판독-수정-기록 동작들은 예를 들어, 단일 동작의 일부로서 2개 이상의 값들이 원자 방식으로 수정될 필요가 있는 애플리케이션들에서 사용될 수도 있다. 2개 이상의 값들을 원자 방식으로 수정하기 위해 하드웨어-기반의 원자 판독-수정-기록 동작들을 이용하는 것은 동시에 실행하는 태스크들 및/또는 쓰레드들이 락들 및/또는 코드의 중요 섹션들의 사용을 필요로 함이 없이, 공유된 오브젝트들을 수정가능하게 할 수도 있다.

[0050] 도 1 은 본 개시물의 하드웨어-기반의 원자 동작들을 구현하는 예시적인 컴퓨팅 시스템 (10) 을 예시하는 블록도이다. 컴퓨팅 시스템 (10) 은 하나 이상의 태스크들을 실행하도록 구성될 수도 있다. 태스크는 컴퓨팅 시스템 (10) 에 의해 실행되는 컴퓨터 프로그램을 지칭할 수도 있다. 일부 예들에서, 컴퓨팅 시스템 (10) 은 병렬 컴퓨팅 시스템일 수도 있다. 예를 들어, 컴퓨팅 시스템 (10) 은 2개 이상의 태스크들을 상이한 계산 유닛들 상에서 병렬로 실행할 수도 있다. 다른 예로서, 하나 이상의 태스크들에 대해, 컴퓨팅 시스템 (10) 은 개개의 태스크의 복수의 인스턴스들을 병렬로 실행할 수도 있다. 쓰레드는 컴퓨팅 시스템 (10) 에서 단일 프로세싱 엘리먼트에 의해 실행되는 태스크의 인스턴스를 지칭할 수도 있다.

[0051] 컴퓨팅 시스템 (10) 은 계산 유닛들 (12A-12B) (일괄하여, "계산 유닛들 (12)") , 원자 동작들 유닛 (14), 및 데이터 저장 시스템 (16) 을 포함한다. 도 1 에 나타낸 바와 같이, 계산 유닛들 (12) 은 원자 동작들 유닛 (14) 에 통신가능하게 커플링되며, 원자 동작들 유닛 (14) 은 데이터 저장 시스템 (16) 에 통신가능하게 커플링된다. 계산 유닛들 (12) 은 원자 동작들 유닛 (14) 을 통해서 간접적으로 데이터 저장 시스템 (16) 에 통신 가능하게 커플링될 수도 있다. 도 1 에 구체적으로 나타내지는 않지만, 계산 유닛들 (12) 은 일부 예들에서, 데이터 저장 시스템 (16) 에 간접적으로 커플링되는 것에 더해서, 데이터 저장 시스템 (16) 에 직접 통신가능하게 커플링될 수도 있다.

[0052] 계산 유닛들 (12) 은 하나 이상의 태스크들을 실행하도록 구성된다. 일부의 경우, 계산 유닛들 (12) 은 태스크들을 병렬로 실행할 수도 있다. 계산 유닛들 (12) 의 각각은 동일한 태스크 또는 상이한 태스크들을 실행할 수도 있다. 계산 유닛들 (12) 에 의해 실행되는 태스크들은 데이터-병렬성을 이용할 수도 있다. 따라서, 태스크의 실행이 쓰레드들의 각각이 상이한 데이터에 대해 동일한 프로그램/태스크를 실행하는 복수의 쓰레드들 (즉, 태스크의 인스턴스들) 에 분산될 수도 있다. 복수의 쓰레드들은 각각의 쓰레드 그룹에 대해, 쓰레드들이 병렬로 실행되는 쓰레드 그룹들에서 실행될 수도 있다. 이와 같이, 계산 유닛들 (12) 의 각각은 주어진 시점에 하나의 태스크를 실행할 수도 있지만, 태스크의 복수의 인스턴스들 (즉, 쓰레드들) 을 동시에 실행할 수도 있다. 쓰레드들은 대안적으로 작업-아이템들로서 지칭될 수도 있으며, 쓰레드 그룹들은 대안적으로 작업-그룹들로서 지칭될 수도 있다.

[0053] 도 1 에 나타낸 바와 같이, 계산 유닛들 (12) 은 하나 이상의 쓰레드들을 실행하도록 구성된다. 예를 들어,

계산 유닛 (12A) 은 쓰레드들 (18A, 18B) 을 실행하도록 구성되며, 계산 유닛 (12B) 은 쓰레드들 (20A, 20B) 을 실행하도록 구성된다.

[0054] 계산 유닛들 (12) 의 각각은 하나 이상의 프로세싱 엘리먼트들을 포함할 수도 있다. 계산 유닛들 (12) 에서의 프로세싱 엘리먼트들의 각각은 단일 쓰레드를 실행할 수도 있다. 계산 유닛들 (12) 중 하나 이상이 다수의 프로세싱 엘리먼트들을 포함하는 예들에서, 단일 계산 유닛 또는 계산 유닛들 (12) 에서의 프로세싱 엘리먼트들은 쓰레드들의 그룹 (예컨대, 작업-그룹) 을 병렬로 실행하도록 일괄하여 구성될 수도 있으며, 프로세싱 엘리먼트들의 각각은 쓰레드들의 그룹 (즉, 작업-그룹) 에서 쓰레드들의 개개의 하나를 실행할 수도 있다. 일부 예들에서, 계산 유닛들 (12) 의 각각은 또한 로컬 메모리, 명령 스토어, 상수 메모리, 등 중 하나 이상을 포함할 수도 있다.

[0055] 일부 예들에서, 계산 유닛들 (12) 의 단일 계산 유닛에서의 프로세싱 유닛들은 단일 명령, 다수의 데이터 (SIMD) 프로세서 및/또는 단일 프로그램, 다수의 데이터 (SPMD) 프로세서를 일괄하여 구현할 수도 있다. 양 쪽 유형들의 프로세서들은 다수의 프로세싱 엘리먼트들 상에서 복수의 태스크의 인스턴스들을 동시에 실행할 수도 있다. SIMD 프로세서에서, 모든 활성 프로세싱 엘리먼트들은 주어진 명령 사이를 동안 상이한 데이터에 대해 동일한 명령들을 실행할 수도 있으며, 프로세싱 엘리먼트들의 각각은 공통 프로그램 카운터를 공유할 수도 있다. SPMD 프로세서에서, 모든 프로세싱 엘리먼트들은 상이한 데이터에 대해 동일한 프로그램을 실행할 수도 있으며, 프로세싱 엘리먼트들의 각각은 그의 자신의 프로그램 카운터를 가질 수도 있다. SPMD 프로세서에 있어, 프로세싱 엘리먼트들은 동일한 프로그램의 상이한 명령들을 동시에 실행할 수도 있다.

[0056] 계산 유닛들 (12) 의 각각은 프로그래밍가능 계산 유닛일 수도 있다. 계산 유닛은 계산 유닛이 프로그램 또는 태스크를 실행하도록 구성된 하드웨어를 포함할 수도 있는 점에서, 프로그래밍가능할 수도 있다. 이에 반해, 비-프로그래밍가능 또는 고정-기능 하드웨어 (fixed-function hardware) 는 프로그램들을 실행하도록 구성되지 않은 하드웨어를 지칭할 수도 있다. 고정-기능 하드웨어가 구성가능할 수도 있지만, 하드웨어의 구성성 (configurability) 은 하드웨어에 의해 실행되는 사용자-정의된 프로그램에 기초하는 것과는 반대로, 고정-기능 하드웨어에 의해 수신된 하나 이상의 제어 신호들에 기초한다.

[0057] 일부 예들에서, 계산 유닛들 (12) 중 하나 (또는, 양쪽) 는 예를 들어, 개방 컴퓨팅 언어 (OpenCL™) API 와 같은, 병렬 프로그래밍 애플리케이션 프로그래밍 인터페이스 (API), 이종 컴퓨팅 플랫폼 API, 및/또는 코-프로세서 컴퓨팅 플랫폼 API 에 따라서 동작할 수도 있다. 추가적인 예들에서, 계산 유닛들 (12) 중 하나 (또는, 양쪽) 는 OpenCL™ 사양에 따라서 정의된 OpenCL™ 계산 유닛에 대응할 수도 있다. OpenCL™ API 에 관한 추가적인 세부 사항들은, 2013년 11월 14일, Khronos OpenCL 작업 그룹, 문서 수정안 19, 버전: 2.0, "OpenCL 사양"; 2013년 11월 14일, Khronos OpenCL 작업 그룹, 문서 수정안 19, 버전: 2.0, "OpenCL C 사양"; 및 2013년 11월 14일, Khronos OpenCL 작업 그룹, 문서 수정안 19, 버전: 2.0, "OpenCL 확장판 사양" 에서 발견될 수 있다.

[0058] 일부 예들에서, 계산 유닛들 (12) 의 각각은 하나 이상의 프로세싱 유닛들에 포함될 수도 있다. 프로세싱 유닛들은 예를 들어, 중앙 처리 유닛 (CPU), 멀티-코어 CPU, 그래픽 프로세싱 유닛 (GPU), 디지털 신호 프로세서 (DSP), 셀 광대역 엔진 (Cell/B.E.) 프로세서, 병렬 프로세서, 코-프로세서, 등을 포함할 수도 있다. 일부 예들에서, 프로세싱 유닛은 이종 컴퓨팅 플랫폼, 병렬 프로세싱 컴퓨팅 플랫폼, 및/또는 코-프로세서 컴퓨팅 플랫폼에서의 계산 디바이스일 수도 있다. 추가적인 예들에서, 프로세싱 유닛은 OpenCL™ 사양에 따라서 정의된 디바이스 및/또는 계산 디바이스에 대응할 수도 있다.

[0059] 일부 예들에서, 계산 유닛들 (12) 의 각각은 동일한 프로세싱 유닛 및/또는 계산 디바이스에 포함될 수도 있다. 예를 들어, 계산 유닛들 (12) 의 각각은 단일 그래픽 프로세싱 유닛 (GPU) 에 포함될 수도 있다. 추가적인 예들에서, 계산 유닛들 (12) 의 각각은 상이한 프로세싱 유닛들 및/또는 계산 디바이스들에 포함될 수도 있다. 예를 들어, 계산 유닛 (12A) 은 GPU 에 포함될 수도 있으며, 계산 유닛 (12B) 은 멀티-코어 CPU 에 포함될 수도 있다.

[0060] 계산 유닛들 (12) 중 하나 (또는, 양쪽) 가 GPU 에 포함되는 예들에서, 계산 유닛들 (12) 의 각각은 일부 예들에서, GPU 의 개개의 세이더 유닛에 대응할 수도 있다. GPU 의 세이더 유닛은 예컨대, 그래픽스-특정의 세이더 프로그램들 및/또는 범용 세이더 프로그램들 (예컨대, 계산 세이더 프로그램들) 과 같은, 세이더 프로그램들을 실행하도록 구성될 수도 있다.

[0061] 일부 예들에서, 호스트 프로그램과 함께 계산 유닛들 (12) 에 의해 실행되는 프로그램들 또는 태스크들은 컴퓨

팅 애플리케이션을 형성할 수도 있다. 호스트 프로그램은 계산 유닛들 (12) 을 포함하는 프로세싱 유닛들 중 하나 이상과 동일한 프로세싱 유닛 상에서 또는 상이한 프로세싱 유닛 상에서 실행할 수도 있다. 호스트 프로그램을 실행하는 프로세싱 유닛은 호스트 디바이스로서 지정될 수도 있다. 컴퓨팅 애플리케이션을 실행하기 위한 예시적인 구성은 CPU 및 GPU 를 포함할 수도 있으며, 여기서 CPU 는 호스트 프로그램을 실행하고 GPU 는 태스크들 및/또는 쓰레드들을 실행하는 계산 유닛들 (12) 을 포함한다.

[0062] 계산 유닛들 (12) 에 의해 실행되는 태스크들이 호스트 프로그램을 포함하는 컴퓨팅 애플리케이션의 일부인 예들에서, 계산 유닛들 (12) 에 의해 실행되는 태스크들에 대한 프로그램 코드는 호스트 프로그램에 의해 제공될 수도 있다. 일부의 경우, 호스트 프로세서는 태스크들에 대한 프로그램 코드를 계산 유닛들 (12) 에 포함되는 온칩 메모리에 및/또는 계산 유닛들 (12) 이 태스크들을 좀더 효율적으로 실행하도록 계산 유닛들 (12) 을 포함하는 하나 이상의 프로세싱 유닛들의 온칩 메모리에, 로드할 수도 있다. 일부 예들에서, 호스트 프로그램은 태스크에 대응하는 쓰레드들을 작업-그룹들 및 작업-그룹들 내 프로세싱 엘리먼트들에 맵핑하고, 그리고 계산 유닛들 (12) 중 하나 (또는, 양쪽) 을 포함하는 컴퓨팅 디바이스 상에서 태스크의 실행을 호출할 수도 있다. 계산 유닛들 (12) 중 하나 (또는, 양쪽) 가 일부인 프로세싱 유닛 (예컨대, 계산 디바이스) 에 포함되는 제어 유닛은 실행을 위해 특정의 작업-그룹들을 계산 유닛들 (12) 중 특정의 계산 유닛에 할당할 수도 있다.

[0063] 도 1 의 예시적인 컴퓨팅 시스템 (10) 은 예시적인 목적들을 위해 2개의 쓰레드들을 각각 실행하는 2개의 계산 유닛들 (12) 을 포함하는 것으로 예시된다. 본 개시물의 기법들을 구현하는 다른 예시적인 컴퓨팅 시스템들은 동일한 또는 상이한 개수의 계산 유닛들을 가질 수도 있으며, 동일한 개수 또는 상이한 개수의 쓰레드들을 실행할 수도 있다. 계산 유닛들의 각각에 의해 동시에 실행되는 쓰레드들의 개수는 동일하거나 또는 상이할 수도 있다. 일부 예들에서, 본 개시물의 기법들을 구현하는 컴퓨팅 시스템은 단일 계산 유닛을 가질 수도 있다. 추가적인 예들에서, 본 개시물의 기법들을 구현하는 컴퓨팅 시스템은 단일 쓰레드를 실행하는 하나 이상의 계산 유닛들을 가질 수도 있다.

[0064] 원자 동작들 유닛 (14) 은 본 개시물에서 설명된 하드웨어-기반의 원자 동작들을 수행하도록 구성될 수도 있다. 일부 예들에서, 원자 동작들 유닛 (14) 은 파이프의 상태를 유지하는 하나 이상의 하드웨어-기반의 원자 판독-수정-기록 동작들을 수행할 수도 있다. 예를 들어, 원자 동작들 유닛 (14) 은 파이프가 태스크 간 통신을 지원하는 FIFO-편성된 데이터 버퍼에 대응하는 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행할 수도 있다. 일부 예들에서, 하드웨어-기반의 원자 판독-수정-기록 동작들은 판독 예약 동작, 기록 예약 동작, 판독 커밋 동작, 및 기록 커밋 동작을 포함할 수도 있다.

[0065] 원자 동작들 유닛 (14) 은 여러 원자 동작들을 수행하는 지령들을 계산 유닛들 (12) 중 하나 이상으로부터 수신하고, 그 지령들을 프로세싱할 수도 있다. 일부 예들에서, 원자 동작들 유닛 (14) 은 원자 동작들의 원자성을 보장하기 위해 지령들을 순차적으로 프로세싱할 수도 있다. 다시 말해서, 원자 동작들 유닛 (14) 은 후속 지령을 프로세싱하기 전에 단일 지령을 시작부터 끝까지 프로세싱할 수도 있다. 일부 예들에서, 지령들은 원자 예약 판독 지령, 원자 예약 기록 지령, 원자 커밋 판독 지령, 및 원자 커밋 기록 지령을 포함할 수도 있다.

[0066] 원자 동작들 유닛 (14) 은 원자 판독-수정-기록 동작들을 수행할 때 데이터 저장 시스템 (16) 과 상호작용할 수도 있다. 예를 들어, 원자 동작들 유닛 (14) 은 원자 판독-수정-기록 동작들의 판독 및 기록 구성요소들을 수행하기 위해 판독 요청들 및 기록 요청들을 데이터 저장 시스템 (16) 으로 발할 수도 있다.

[0067] 일부 예들에서, 원자 동작들 유닛 (14) 은 원자 지령을 프로세싱하는 것에 응답하여 하나 이상의 파이프 상태 파라미터들을 반환할 수도 있다. 예를 들어, 예약 동작의 일부로서, 원자 동작들 유닛 (14) 은 하나 이상의 파이프 상태 파라미터들을 그 요청하는 태스크로 반환할 수도 있다. 다른 예들에서, 원자 동작들 유닛 (14) 은 원자 지령을 실행하는 것에 응답하여 임의의 데이터를 반환하지 않을 수도 있다.

[0068] 원자 동작들 유닛 (14) 은 하나 이상의 프로세서들 상에서 구현되는 하드웨어 유닛일 수도 있다. 하드웨어 유닛은 본 개시물에 의해 원자 동작들 유닛 (14) 에 기인하는 기능을 구현하는 회로를 포함할 수도 있다. 예를 들어, 하드웨어 유닛은 일부 예들에서, 디지털 회로, 아날로그 회로, 산술 로직 유닛들 (ALU들), 데이터경로들, 등 중 하나 이상을 포함할 수도 있다.

[0069] 일부 예들에서, 원자 동작들 유닛 (14) 은 고정-기능 (즉, 비-프로그래밍가능) 하드웨어 유닛일 수도 있다. 다시 말해서, 원자 동작들 유닛 (14) 은 이러한 예들에서, 사용자-규정된 프로그램들을 수신하여 실행하도록

반드시 구성되지 않을 수도 있다.

[0070] 원자 동작들 유닛 (14) 은 계산 유닛들 (12) 중 하나 (또는, 양쪽) 과 동일한 프로세싱 유닛 상에서 또는 상이한 프로세싱 유닛 상에서 구현될 수도 있다. 일부 예들에서, 원자 동작들 유닛 (14) 은 계산 유닛들 (12) 중 하나 (또는, 양쪽) 을 또한 포함하는 그래픽 프로세싱 유닛 (GPU) 에 포함될 수도 있다. 추가적인 예들에서, 원자 동작들 유닛 (14) 은 이종 컴퓨팅 플랫폼, 병렬 프로세싱 컴퓨팅 플랫폼, 및/또는 코-프로세서 컴퓨팅 플랫폼에 포함되는 디바이스 및/또는 계산 디바이스 (예컨대, OpenCL™ 사양에 따라서 정의된 디바이스 및/또는 계산 디바이스) 에 포함될 수도 있다.

[0071] 일부 예들에서, 원자 동작들 유닛 (14) 은 데이터 저장 시스템 (16) 의 모두 또는 일부를 포함하는 메모리 서브시스템에 포함될 수도 있다. 예를 들어, 원자 동작들 유닛 (14) 은 프로세싱 유닛의 캐시 유닛의 부분 (예컨대, GPU 의 캐시 유닛) 일 수도 있다. 추가적인 예들에서, 원자 동작들 유닛 (14) 은 이종 컴퓨팅 플랫폼, 병렬 프로세싱 컴퓨팅 플랫폼, 및/또는 코-프로세서 컴퓨팅 플랫폼에 포함되는 디바이스 및/또는 계산 디바이스 (예컨대, OpenCL™ 사양에 따라서 정의된 디바이스 및/또는 계산 디바이스) 와 연관되는 메모리 서브시스템의 일부에 통합될 수도 있다.

[0072] 데이터 저장 시스템 (16) 은 태스크 간 통신에 사용될 파이프의 상태 및 콘텐츠를 나타내는 데이터를 저장하도록 구성될 수도 있다. 데이터 저장 시스템 (16) 은 파이프 상태 파라미터들 (22) 및 파이프 데이터 버퍼 (24) 를 포함한다. 데이터 저장 시스템 (16) 은 예컨대, 휘발성 메모리, 비-휘발성 메모리, 캐시, 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 등과 같은, 하나 이상의 데이터 저장 유닛들을 포함할 수도 있다.

[0073] 일부 예들에서, 데이터 저장 시스템 (16) 은 메모리 저장 유닛과 같은, 단일 저장 유닛을 포함할 수도 있다. 메모리 저장 유닛은 계산 유닛들 (12) 또는 계산 유닛들 (12) 을 포함하는 프로세싱 유닛의 외부에 있는 오프-칩 메모리를 포함하는 프로세싱 유닛 상에 로케이트되는 온칩 메모리일 수도 있다.

[0074] 추가적인 예들에서, 데이터 저장 시스템 (16) 은 다수의 저장 유닛들을 포함할 수도 있다. 예를 들어, 데이터 저장 시스템 (16) 은 캐시 저장 유닛 및 하부의 메모리 저장 유닛을 포함할 수도 있다. 하부의 메모리 저장 유닛은 파이프 상태 파라미터들 (22) 및/또는 파이프 데이터 버퍼 (24) 의 소스 복사본을 저장할 수도 있으며, 캐시 저장 유닛은 파이프 상태 파라미터들 (22) 및/또는 파이프 데이터 버퍼 (24) 의 캐시된 복사본을 저장할 수도 있다. 캐시된 복사본은 소스 복사본의 불완전한 버전일 수도 있다.

[0075] 데이터 저장 시스템 (16) 이 캐시 저장 유닛 및 하부의 메모리 저장 유닛을 포함하는 예들에서, 캐시 저장 유닛은 일부 예들에서, 온칩 캐시일 수도 있으며, 하부의 메모리 저장 유닛은 오프-칩 메모리 저장 유닛일 수도 있다. 다시 말해서, 캐시 저장 유닛은 계산 유닛들 (12) 을 포함하는 프로세싱 유닛 상에 로케이트될 수도 있거나 또는/및 메모리 저장 유닛은 계산 유닛들 (12) 을 포함하는 프로세싱 유닛 외부에 있을 수도 있다.

[0076] 파이프 상태 파라미터들 (22) 은 파이프의 상태를 나타낸다. 파이프 상태 파라미터들 (22) 은 얼마나 많은 패킷들이 현재 파이프에 저장되어 있는지를 나타내는 패킷 카운트 파라미터, 파이프에 대한 시작 패킷을 저장하는 패킷 엔트리를 나타내는 제 1 패킷 오프셋 파라미터, 얼마나 많은 비커밋된 예약들이 현재 파이프에 펜딩 중인지를 나타내는 예약 카운트 파라미터, 및 얼마나 많은 패킷 엔트리들이 현재 파이프에 예약되어 있는지를 나타내는 예약된 패킷 카운트 파라미터 중 하나 이상을 포함할 수도 있다.

[0077] 패킷 카운트 파라미터 및 제 1 패킷 오프셋 파라미터는 예를 들어, 링 버퍼로서 구현되는 파이프의 상태를 정의하기 위해 사용될 수도 있다. 예약 카운트 파라미터 및 예약된 패킷 카운트 파라미터는 예를 들어, 예약을 지원하는 파이프의 예약 상태를 정의하기 위해 사용될 수도 있다.

[0078] 위에서 설명된 파이프 상태 파라미터들 (22) 은 단지 본 개시물의 기법들과 함께 사용될 수도 있는 파이프 상태 파라미터들의 세트의 예들이다. 다른 예들에서, 동일한 개수 또는 상이한 개수의 파이프 상태 파라미터들이 사용될 수도 있으며, 사용되는 파이프 상태 파라미터들의 유형은 동일하거나 또는 상이할 수도 있다.

[0079] 파이프 데이터 버퍼 (24) 는 파이프의 데이터 콘텐츠에 대응하는 데이터 패킷들을 저장하도록 구성된 복수의 패킷 엔트리들을 포함할 수도 있다. 비-공백 파이프에 대해, 파이프 데이터 버퍼 (24) 는 인접한 패킷 엔트리들의 세트에 저장되는 패킷들의 시퀀스를 포함할 수도 있다. 일부의 경우, 인접한 패킷 엔트리들의 세트는 파이프 데이터 버퍼 (24) 의 끝으로부터 파이프 데이터 버퍼 (24) 의 처음까지 순환할 수도 있다. 공백 파이프에 대해, 파이프 데이터 버퍼 (24) 에서의 패킷 엔트리들의 각각은 비어 있을 수도 있다.

- [0080] 일부 예들에서, 파일 데이터 버퍼 (24) 는 링 버퍼로서 편성될 수도 있다. 링 버퍼는 마치 버퍼가 단-대-단 접속된 것처럼, 단일, 고정된-사이즈 버퍼를 이용하는 데이터 구조를 지칭할 수도 있다. 다시 말해서, 파일 데이터 버퍼 (24) 의 반대 단부들에 로케이트된 패킷 엔트리들은 링 버퍼를 구현하기 위해 서로 인접한 것으로 개념적으로 간주될 수도 있다.
- [0081] 파일 상태 파라미터들 (22) 및 파일 데이터 버퍼 (24) 는 FIFO-편성된 버퍼를 구현하기 위해 사용될 수도 있다. FIFO-편성된 버퍼는 기록 인터페이스 및 판독 인터페이스가 FIFO 원리에 따라서 동작하는 버퍼를 지칭할 수도 있다. FIFO 원리에 따라서 동작하는 기록 인터페이스는 기록될 데이터를 버퍼에 이미 저장되어 있는 임의의 데이터의 끝에 첨부하는 인터페이스를 지칭할 수도 있다. FIFO 원리에 따라서 동작하는 판독 인터페이스는 버퍼에 기록되는 가장 빠른 시기에 미판독된 데이터를 선택하여 버퍼로부터 제거하는 인터페이스를 지칭할 수도 있다.
- [0082] 일부 예들에서, 계산 유닛들 (12), 원자 동작들 유닛 (14), 및 데이터 저장 시스템 (16) 중 하나 이상의 모두 또는 일부는 프로세서 유닛의 일부일 수도 있다. 예를 들어, 계산 유닛들 (12), 원자 동작들 유닛 (14), 및 데이터 저장 시스템 (16) 의 부분에 대응하는 캐시 유닛은 GPU 에 모두 포함될 수도 있다. 이 예에서, 하부의 메모리 저장 유닛은 GPU 와 동일한 칩 상에 로케이트되지 않는 오프-칩 메모리에 포함될 수도 있다.
- [0083] 원자 동작들 유닛 (14) 이 컴퓨팅 디바이스의 일부인 예들에서, 계산 유닛 (12A) 및 계산 유닛 (12B) 이 상이한 컴퓨팅 디바이스들의 일부이면, 각각의 컴퓨팅 디바이스는 원자 동작들 유닛 (14) 을 포함할 수도 있다. 이러한 예들에서, 계산 유닛 (12A) 및 계산 유닛 (12B) 양쪽이 단일 원자 동작들 유닛 (14) 에 통신가능하게 커플링되는 대신, 도 1 에 나타낸 바와 같이, 계산 유닛 (12A) 및 계산 유닛 (12B) 은 그들의 개개의 컴퓨팅 디바이스들에서 그들의 개개의 원자 동작들 유닛들에 통신가능하게 커플링될 수도 있다.
- [0084] 컴퓨팅 시스템 (10) 의 동작이 이하 설명될 것이다. 아래에서 설명되는 예시적인 동작에서, 파일 (예컨대, 파일 상태 파라미터들 (22) 및 파일 데이터 버퍼 (24)) 는 데이터 저장 시스템 (16) 에 이미 할당되어 있으며, 태스크의 실행은 쓰레드들 (18A, 18B) 이 계산 유닛 (12A) 상에서 동시에 실행하고 있고 쓰레드들 (20A, 20B) 이 계산 유닛 (12B) 상에서 동시에 실행하고 있도록, 계산 유닛들 (12) 의 각각 상에서 호출되었다.
- [0085] 쓰레드들 (18, 20) 중 하나 이상이 원자 동작 지령을 원자 동작들 유닛 (14) 으로 발한다. 원자 동작들 유닛 (14) 은 원자 동작 지령을 실행한다. 예를 들어, 원자 동작들 유닛 (14) 은 원자 동작 지령에 대응하는 원자 동작을 수행할 수도 있다.
- [0086] 원자 동작들 유닛 (14) 에 의해 수행되는 원자 동작은 2개 이상의 파일 상태 파라미터들을 원자 방식으로 판독하고, 수정하고, 그리고 기록하는 원자 판독-수정-기록 동작일 수도 있다. 원자 판독-수정-기록 동작을 수행하기 위해, 원자 동작들 유닛 (14) 은 인터럽트불가능한 동작들의 세트를 수행할 수도 있다. 인터럽트 불가능한 동작들의 세트는 파일 상태 파라미터들 (22) 에 대응하는 값들을 데이터 저장 시스템 (16) 으로부터 판독하는 동작, 파일 상태 파라미터들 (22) 에 대한 수정된 값들을 발생시키는 동작, 및 파일 상태 파라미터들 (22) 에 대한 수정된 값들을 데이터 저장 시스템 (16) 에 기록하는 동작을 포함할 수도 있다.
- [0087] 판독 동작을 수행하기 위해, 원자 동작들 유닛 (14) 은 판독 요청을 데이터 저장 시스템 (16) 으로 발할 수도 있다. 데이터 저장 시스템 (16) 은 파일 상태 파라미터들 (22) 을 원자 동작들 유닛 (14) 에 제공할 수도 있다. 일부 예들에서, 파일에 대한 모든 파일 상태 파라미터들 (22) 은 단일 판독 요청에 응답하여 제공될 수도 있다. 추가적인 예들에서, 파일에 대한 모든 파일 상태 파라미터들 (22) 은 단일 판독 사이클의 일부로서 제공될 수도 있다. 추가적인 예들에서, 모든 파일 상태 파라미터들 (22) 은 단일 메모리 트랜잭션 동안 준비중인 단일 데이터 워드를 형성할 수도 있다. 수정 동작을 수행하기 위해, 원자 동작들 유닛 (14) 은 하나 이상의 데이터경로들을 이용할 수도 있다. 데이터 경로들은 예를 들어, 하나 이상의 ALU 들을 포함할 수도 있다. 기록 동작을 수행하기 위해, 원자 동작들 유닛 (14) 은 기록 요청을 데이터 저장 시스템 (16) 으로 발하고, 파일 상태 파라미터들 (22) 에 대한 수정된 값들을 데이터 저장 시스템 (16) 에 제공할 수도 있다.
- [0088] 원자 동작들 유닛 (14) 은 수행될 원자 동작의 유형에 기초하여, 수신된 파일 상태 파라미터 값들을 수정할 수도 있다. 원자 동작들의 예시적인 유형들은 원자 예약 판독 동작, 원자 예약 기록 동작, 원자 커밋 기록 동작, 및 원자 커밋 판독 동작을 포함한다.
- [0089] 일부 예들에서, 판독 동작을 수행한 후, 원자 동작들 유닛 (14) 은 데이터 저장 시스템 (16) 으로부터 판독된 파일 상태 파라미터 값들 중 하나 이상을 지령을 발한 요청하는 쓰레드로 반환하는 반환 동작을 수행할 수도

있다. 반환된 값들은 원자 판독-수정-기록 동작을 수행하기 직전 파이프의 상태에 대응할 수도 있다.

[0090] 원자 동작들 유닛 (14)은 계산 유닛들 (12)으로부터 수신된 원자 동작 지령의 유형에 기초하여 상이한 유형들의 원자 판독-수정 기록 동작들을 수행할 수도 있다. 예를 들어, 원자 동작들 유닛 (14)은 계산 유닛들 (12)로부터 원자 예약 기록 지령을 수신하는 것에 응답하여 원자 예약 기록 동작을 수행하거나, 계산 유닛들 (12)으로부터 원자 예약 판독 지령을 수신하는 것에 응답하여 원자 예약 판독 동작을 수행하거나, 계산 유닛들 (12)로부터 원자 커밋 기록 지령을 수신하는 것에 응답하여 원자 커밋 기록 동작을 수행하거나, 및/또는 계산 유닛들 (12)로부터 원자 커밋 판독 지령을 수신하는 것에 응답하여 원자 커밋 판독 동작을 수행할 수도 있다.

[0091] 원자 지령들은 대안적으로 본원에서 요청들로서 지칭될 수도 있으며, 지령을 발하는 쓰레드는 대안적으로 본원에서 요청자로서 지칭될 수도 있다. 예를 들어, 쓰레드 (18A)가 예약 기록 지령을 원자 동작들 유닛 (14)으로 발할 때, 쓰레드 (18A)는 기록 예약 요청을 발하는 요청자로서 말할 수도 있다.

[0092] 본 개시물에서 설명되는 원자 동작들 및 원자 지령들에 대한 예시적인 유즈 케이스가 이하 설명될 것이다. 애플리케이션 프로그래밍 인터페이스 (API)는 다수의 태스크들 및/또는 쓰레드들이 동시에 실행할 수도 있는 컴퓨팅 환경에서 태스크 간 통신을 가능하게 하는 파이프들에 대한 지원을 제공할 수도 있다. 파이프에서 인터리빙을 피하기 위해, API는 파이프에서 인접한 패킷 엔트리들의 세트들이 쓰레드들에 의해 판독 또는 기록 용으로 예약될 수 있도록 할 수도 있다. 이러한 파이프는 예약들을 지원한다고 말할 수도 있다.

[0093] 예약들을 지원하기 위해, API는 예약 파이프 지령들, 판독 및 기록 파이프 지령들, 및 커밋 파이프 지령들을 제공할 수도 있다. 이들 지령들은 태스크 및/또는 쓰레드로 하여금 파이프에 기록하고 그로부터 판독 가능하도록 하기 위해 프로그래머 및/또는 다른 소스 코드 발생기에 의해 소스 코드에 배치될 수도 있다.

[0094] 프로그래머의 관점으로부터, 데이터를 파이프에 기록하는데 사용되는 동작들의 일반적인 시퀀스는 다음과 같을 수도 있다: 기록을 위한 예약 패킷 엔트리들은, 데이터를 예약된 패킷 엔트리들에 기록하고, 그 예약된 패킷 엔트리들을 파이프에 커밋한다. 데이터를 파이프에 기록하는 동작들의 위에서 언급된 시퀀스를 수행하기 위해, 지령들의 기록 시퀀스는 소스 코드에 배치될 수도 있다. 지령들의 기록 시퀀스는 예약 기록 파이프 지령, 기록 파이프 지령, 및 커밋 기록 파이프 지령을 포함할 수도 있다. 일부 예들에서, 위에서 언급된 지령들의 기록 시퀀스에서의 지령들은 OpenCL™ 사양에 따라서 정의된, `reserve_write_pipe` 지령, `write_pipe` 지령, 및 `commit_write_pipe`에 각각 대응할 수도 있다.

[0095] 예약 기록 파이프 지령은 파이프에서 요청된 패킷 엔트리들의 개수를 판독용으로 예약하고, 그 예약을 나타내는 데이터 (예컨대, 예약 핸들)을 반환할 수도 있다. 기록 파이프 지령은 데이터 (예컨대, 패킷들)를 예약 핸들에 의해 표시된 예약된 패킷 엔트리들에 기록할 수도 있다. 커밋 기록 파이프 지령은 예약된 패킷 엔트리들을 파이프에 커밋할 수도 있다. 예약된 패킷 엔트리들을 파이프로 커밋하는 것은 제작자 태스크가 데이터를 파이프의 예약된 패킷 엔트리들에 기록하는 것을 종료하였다는 것을, 그리고 이러한 패킷 엔트리들이 소비자 태스크에 이용할 준비가 되어 있다는 것을 표시할 수도 있다.

[0096] 프로그래머의 관점으로부터, 파이프로부터 데이터를 판독하는데 사용되는 동작들의 일반적인 시퀀스는 다음과 같을 수도 있다: 판독을 위한 예약 패킷 엔트리들은, 데이터를 예약된 패킷 엔트리들로부터 판독하고, 예약된 패킷 엔트리들을 파이프에 커밋한다. 파이프로부터 데이터를 판독하는 위에서 언급된 동작들의 시퀀스를 수행하기 위해, 지령들의 판독 시퀀스는 소스 코드에 배치될 수도 있다. 지령들의 판독 시퀀스는 예약 판독 파이프 지령, 판독 파이프 지령, 및 커밋 판독 파이프 지령을 포함할 수도 있다. 일부 예들에서, 위에서 언급된 지령들의 판독 시퀀스에서의 지령들은 OpenCL™ 사양에 따라서 정의된, `reserve_read_pipe` 지령, `read_pipe` 지령, 및 `commit_read_pipe`에 각각 대응할 수도 있다.

[0097] 예약 판독 파이프 지령은 요청된 패킷 엔트리들의 개수를 판독용으로 예약하고, 예약을 나타내는 데이터 (예컨대, 예약 핸들)를 반환할 수도 있다. 판독 파이프 지령은 예약 핸들에 의해 표시된 예약된 패킷 엔트리들로부터 데이터 (예컨대, 패킷들)를 판독할 수도 있다. 커밋 판독 파이프 지령은 예약된 패킷 엔트리들을 파이프에 커밋할 수도 있다. 예약된 패킷 엔트리들을 파이프로 커밋하는 것은 소비자 태스크가 파이프의 예약된 패킷 엔트리들로부터 데이터를 판독하는 것을 종료하였다는 것을, 그리고 이러한 패킷 엔트리들이 파이프로부터 소거되었거나 및/또는 제거되었을 수도 있다는 것을 표시할 수도 있다.

[0098] (예약 파이프 및 커밋 파이프 지령들을 포함하는) 위에서 설명된 지령들의 판독 및 기록 시퀀스들은 소스 코드-레벨 지령들일 수도 있다. 컴파일러는 소스 코드-레벨 지령들을 컴파일하여, 컴파일된 코드-레벨 지령들을 포함하는 컴파일된 코드를 발생시킬 수도 있다. 컴파일된 코드-레벨 지령들은 본 개시물에서 설명되는 원자

예약 지령들 및 원자 커밋 지령들을 포함할 수도 있다.

[0099] 컴퓨팅 시스템 (10)의 예시적인 동작이, 쓰레드들 (18, 20) 중 하나가 파이프 예약들을 이용하여 데이터를 파이프에 기록하고 파이프 예약들을 이용하여 데이터를 파이프로부터 판독하는 경우들에 대해 이하 설명될 것이다. 이 예에서, 쓰레드들 (18, 20)의 각각은 본 개시물의 원자 예약 및 원자 커밋 지령들을 포함하는 컴파일된 코드를 실행할 수도 있다.

[0100] 이 설명의 목적들을 위해, 쓰레드 (18A)는 데이터를 파이프에 기록하는 것으로 설명될 것이며 쓰레드 (20A)는 파이프로부터 판독하는 것으로 설명될 것이다. 그러나, 다른 예들에서, 쓰레드들 (18, 20) 중 임의의 쓰레드가 유사한 방식으로 데이터를 파이프에 기록하고 데이터를 파이프로부터 판독할 수도 있다.

[0101] 데이터를 파이프에 기록하기 위해, 쓰레드 (18A)는 원자 예약 기록 지령을 원자 동작들 유닛 (14)으로 발할 수도 있다. 원자 예약 기록 지령은 데이터가 기록될 파이프를 나타내는 파라미터 및 기록용으로 예약될 요청된 패킷 엔트리들의 개수를 나타내는 파라미터 중 하나 (또는, 양쪽)를 포함할 수도 있다. 데이터가 기록될 파이프를 나타내는 파라미터 일부 예들에서, 파이프 어드레스 파라미터일 수도 있다. 예를 들어, 파이프 어드레스 파라미터는 파이프 상태 파라미터들 (22)이 액세스되고 있는 특정의 파이프에 대해 저장되는 메모리 서브시스템 (예컨대, 데이터 저장 시스템 (16))에서의 로케이션에 대응하는 메모리 어드레스일 수도 있다.

파이프 어드레스 파라미터는 데이터 저장 시스템 (16)이 하나 보다 많은 파이프에 대한 데이터를 포함할 때 데이터 저장 시스템 (16)에서 수정할 파이프가 어느 파이프인지를 식별하고, 파이프 데이터가 저장되는 데이터 저장 시스템 (16)에서의 로케이션을 식별하기 위해 사용될 수도 있다.

[0102] 원자 예약 기록 지령을 수신하는 것에 응답하여, 원자 동작들 유닛 (14)은 원자 예약 기록 동작을 수행할 수도 있다. 원자 예약 기록 동작은 본 개시물에서 설명되는 바와 같이 2개 이상의 파이프 상태 파라미터들을 수정하는 원자 판독-수정-기록 동작의 일 예일 수도 있다.

[0103] 원자 예약 기록 동작을 수행하기 위해, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들을 데이터 저장 시스템 (16)으로부터 획득하고, 데이터 저장 시스템 (16)으로부터 획득되는 파이프 상태 파라미터들 중 하나 이상을 쓰레드 (18A)로 반환할 수도 있다. 일부 예들에서, 원자 동작들 유닛 (14)에 의해 반환된 파이프 상태 파라미터들은 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터, 및 예약된 패킷 카운트 파라미터를 포함할 수도 있다.

[0104] 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 추가로 결정할 수도 있다. 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 기록 예약이 파이프에 대해 이루어졌다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다. 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 갖지 않는다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록 (즉, 기록 예약이 파이프에 대해 이루어지지 않았다는 것을 수정된 값들이 반영하도록) 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0105] 파이프 상태 파라미터들에 대한 수정된 값들을 결정한 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 (22)에 대한 수정된 값들을 데이터 저장 시스템 (16)에 기록할 수도 있다. 원자 예약 기록 지령에 기인하는 기능은 일부 예들에서, 인터럽트불가능한 동작들의 세트의 일부로서 수행될 수도 있다. 다시 말해서, 원자 동작들 유닛 (14)은 다른 원자 동작 지령들 (예컨대, 원자 예약 기록 지령을 발한 태스크 또는 쓰레드와 동시에 실행하고 있는 태스크들 및/또는 쓰레드들에 의해 발해지는 다른 원자 동작 지령들)과 연관될 수도 있는 다른 동작들을 수행하기 위해 실행을 인터럽트함이 없이 원자 예약 기록 지령에 기인하는 기능을 시작부터 끝까지 수행할 수도 있다. 인터럽트불가능한 동작들의 세트를 이 방식으로 수행하는 것은 원자 동작들 유닛 (14)으로 하여금, 심지어 동시에 실행하는 쓰레드들 및/또는 태스크들이 그 파이프를 기록용으로 동시에 예약하려고 시도할 수도 있는 경우에도 적합한 파이프의 기능을 보장하는 원자 예약 기록 지령을 구현가능하게 한다. 원자 예약 기록 지령의 예시적인 구현예에 관련한 추가적인 세부 사항들이 아래에서 도 2에 대해서 설명된다.

[0106] 쓰레드 (18A)는 반환된 파이프 상태 파라미터들에 기초하여 기록 예약이 성공적이었는지 여부를 결정할 수도 있다. 기록 예약이 성공적이었는지 여부를 결정하기 위해, 쓰레드 (18A)는 일부 예들에서, 반환된 파이프 상태 파라미터들에 기초하여, 원자 동작들 유닛 (14)이 아래에서 더 상세히 설명되는 것과 동일한 결정을 행할

수 있는 방법과 유사한 방법으로, 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 결정할 수도 있다. 반환된 파이프 상태 파라미터들은 예약 기록 동작의 실행 전 파이프의 상태를 나타낼 수도 있다.

[0107] 기록 예약이 성공적이지 않았으면, 쓰레드 (18A)는 기록 예약 동작의 완료에 응답하여 데이터를 파이프에 기록하거나 또는 데이터를 파이프에 커밋하지 않을 수도 있다. 일부 예들에서, 기록 예약이 성공적이지 않았으면, 쓰레드 (18A)는 여러 조건을 가지는 실행을 종료할 수도 있다. 추가적인 예들에서, 기록 예약이 성공적이지 않았으면, 쓰레드 (18A)는 그 조건을 처리하는 하나 이상의 동작들을 수행할 수도 있다. 예를 들어, 쓰레드 (18A)는 기록 예약을 수행하려고 시도하는 것을 중지하고 계속해서 다른 동작들을 수행할 수도 있다. 다른 예로서, 쓰레드 (18A)는 어떤 시간 기간 이후 기록 예약 동작을 반복하도록 시도할 수도 있다. 이것은 소비자 쓰레드 시간이 파이프로부터 하나 이상의 패킷들을 판독하는 것을 가능하게 하여, 후속 기록 예약 동작에서 예약될 수 있는 추가적인 패킷 엔트리들을 해방가능하게 한다.

[0108] 기록 예약이 성공적이었으면, 쓰레드 (18A)는 원자 기록 예약 지령을 프로세싱할 때 원자 동작들 유닛 (14)에 의해 반환된 하나 이상의 파이프 상태 파라미터들에 기초하여, 데이터를 기록할 파이프 데이터 버퍼 (24)의 물리적인 패킷 엔트리들을 결정할 수도 있다. 쓰레드 (18A)가 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터, 및 예약된 패킷 카운트 파라미터를 수신하는 예들에서, 쓰레드 (18A)는 예를 들어, 이들 파라미터들의 각각의 값들을 함께 가산함으로써, 데이터를 기록할 파이프 데이터 버퍼 (24)의 제 1 물리적인 패킷 엔트리를 결정할 수도 있다. 쓰레드 (18A)는 데이터를 결정된 제 1 물리적인 패킷 엔트리 및 다음 패킷 엔트리들에, 예약된 패킷 엔트리들의 개수까지 기록할 수도 있다. 예약된 패킷 엔트리들의 개수가 기록되기 전에 파이프 데이터 버퍼 (24)의 끝에 도달되면, 쓰레드 (18A)는 추가적인 물리적인 패킷 엔트리들을 기록하기 위해 파이프 데이터 버퍼 (24)의 시작으로 순환할 수도 있다.

[0109] 데이터를 파이프 데이터 버퍼 (24)에서의 예약된 패킷 엔트리들에 기록하기 위해, 쓰레드 (18A)는 일부 예들에서, 기록될 파이프와 연관되는 파이프 어드레스에 기초하여 그리고 데이터를 기록할 결정된 물리적인 패킷 엔트리들에 기초하여, 및/또는 반환된 파이프 상태 파라미터들에 기초하여, 패킷 엔트리들 중 하나 이상에 대한 물리 메모리 어드레스를 결정할 수도 있다. 일부 예들에서, 파이프 어드레스는 데이터 저장 시스템 (16)에 파이프를 할당한 호스트 프로그램에 의해 쓰레드 (18A)에 제공될 수도 있다. 이러한 예들에서, 파이프 어드레스는 일부 예들에서, 계산 유닛 (12A)의 온침 상수 메모리에 저장될 수도 있다. 예를 들어, 호스트 프로세서 상에서 실행하는 호스트 프로그램은 일부 예들에서, 파이프 어드레스를, 계산 유닛 (12A) 상에서 쓰레드 (18A)의 실행을 호출하는 함수의 인수로서 제공할 수도 있다 (예컨대, 상수 파라미터들이 커널 인수들로서 쓰레드 (18A)에 제공될 수도 있다).

[0110] 일부 예들에서, 쓰레드 (18A)는 하나 이상의 메모리 기록 요청들을 데이터 저장 시스템 (16)로 발하여 패킷 데이터를 데이터 저장 시스템 (16)에 포함되는 예약된 패킷 엔트리들에 기록할 수도 있다. 기록 요청들의 각각은 데이터를 기록할 데이터 저장 시스템 (16)에서의 물리적인 로케이션 (예컨대, 메모리 어드레스들) 및 기록될 데이터를 규정할 수도 있다. 기록 요청들은 패킷 데이터를 예약된 패킷 엔트리들의 모두에 일괄하여 기록할 수도 있다.

[0111] 쓰레드 (18A)는 일부 예들에서, 메모리 기록 요청들을 데이터 저장 시스템 (16) 및/또는 데이터 저장 시스템 (16)를 포함하는 메모리 서브시스템으로 직접 발할 수도 있다. 다시 말해서, 이러한 예들에서, 컴퓨팅 시스템 (10)은 계산 유닛 (12A)과 원자 동작들 유닛 (14)을 바이패스하는 데이터 저장 시스템 (16) 사이의 통신 인터페이스를 포함할 수도 있으며, 쓰레드 (18A)는 메모리 기록 요청들이 데이터 저장 시스템 (16)에 의해 수신되지만 원자 동작들 유닛 (14)을 통과하지 않도록, 그 통신 인터페이스를 통해서 메모리 기록 요청들을 데이터 저장 시스템 (16)으로 발할 수도 있다.

[0112] 패킷 데이터가 파이프 데이터 버퍼 (24)에서의 예약된 패킷 엔트리들에 기록되어진 후, 원자 동작들 유닛 (14)은 원자 커밋 기록 지령을 원자 동작들 유닛 (14)으로 발할 수도 있다. 원자 커밋 기록 지령은 쓰레드 (18A)가 데이터를 파이프의 예약된 패킷 엔트리들에 기록하는 것을 종료하였다는 것을, 그리고 이러한 패킷 엔트리들이 소비자 태스크에 이용할 준비가 되어 있다는 것을 표시할 수도 있다.

[0113] 원자 커밋 기록 지령은 데이터가 기록된 파이프를 나타내는 파라미터를 포함할 수도 있다. 데이터가 기록된 파이프를 나타내는 파라미터는 일부 예들에서, 파이프 어드레스 파라미터일 수도 있다. 예를 들어, 파이프 어드레스 파라미터는 파이프 상태 파라미터들 (22)이 액세스되고 있는 특정의 파이프에 대해 저장되는 메모리 서브시스템 (예컨대, 데이터 저장 시스템 (16))에서의 로케이션에 대응하는 메모리 어드레스일 수도 있다.

파이프 어드레스 파라미터는 데이터 저장 시스템 (16) 이 하나 보다 많은 파이프에 대한 데이터를 포함할 때 데이터 저장 시스템 (16) 에서 수정할 파이프를 식별하고, 파이프 데이터가 저장되는 데이터 저장 시스템 (16) 에서의 로케이션을 식별하기 위해 사용될 수도 있다.

- [0114] 원자 커밋 기록 지령을 수신하는 것에 응답하여, 원자 동작들 유닛 (14) 은 원자 커밋 기록 동작을 수행할 수도 있다. 원자 커밋 기록 동작은 본 개시물에서 설명되는 바와 같이 2개 이상의 파이프 상태 파라미터들을 수정하는 원자 판독-수정-기록 동작의 일 예일 수도 있다.
- [0115] 원자 커밋 기록 동작을 수행하기 위해, 원자 동작들 유닛 (14) 은 데이터 저장 시스템 (16) 으로부터 파이프 상태 파라미터들을 획득할 수도 있다. 원자 예약 기록 동작과는 달리, 원자 동작들 유닛 (14) 은 일부 예들에서, 원자 커밋 기록 동작을 수행할 때, 데이터 저장 시스템 (16) 으로부터 획득되는 임의의 파이프 상태 파라미터들을 쓰레드 (18A) 로 반환하지 않을 수도 있다.
- [0116] 원자 동작들 유닛 (14) 은 원자 커밋 기록 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 수정된 값이 반영하도록, 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시킬 수도 있다. 일부 예들에서, 제 1 파이프 파라미터는 예약 카운트 파라미터일 수도 있으며, 원자 동작들 유닛 (14) 은 원자 커밋 기록 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 반영하기 위해, 예약 카운트 파라미터를 감분시킬 수도 있다.
- [0117] 원자 동작들 유닛 (14) 은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프에 대한 모든 예약들이 커밋되었는지 여부를 추가로 결정할 수도 있다. 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14) 은 커밋된 엔트리들이 파이프에 배치되어 있다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시킬 수도 있다. 일부 예들에서, 원자 동작들 유닛 (14) 은 패킷 카운트 파라미터를 예약된 패킷 카운트 파라미터와 동일한 양 만큼 증분시킬 수도 있으며, 커밋된 엔트리들이 파이프에 배치되어 있다는 것을 반영하기 위해, 예약된 패킷 카운트 파라미터를 제로로 리셋할 수도 있다. 파이프에 대한 모든 예약들이 커밋되지 않았다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14) 은 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록, 제 1 파이프 상태 파라미터 (예컨대, 예약 카운트 파라미터) 가 아닌, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.
- [0118] 파이프 상태 파라미터들에 대한 수정된 값들을 결정한 후, 원자 동작들 유닛 (14) 은 파이프 상태 파라미터들 (22) 에 대한 수정된 값들을 데이터 저장 시스템 (16) 에 기록할 수도 있다. 원자 커밋 기록 지령에 기인하는 기능은 일부 예들에서, 인터럽트불가능한 동작들의 세트의 일부로서 수행될 수도 있다. 다시 말해서, 원자 동작들 유닛 (14) 은 다른 원자 동작 지령들 (예컨대, 원자 커밋 기록 지령을 발한 태스크 또는 쓰레드와 동시에 실행하고 있는 태스크들 및/또는 쓰레드들에 의해 발해지는 다른 원자 동작 지령들) 과 연관될 수도 있는 다른 동작들을 수행하기 위해 실행을 인터럽트함이 없이 원자 커밋 기록 지령에 기인하는 기능을 시작부터 끝까지 수행할 수도 있다. 인터럽트불가능한 동작들의 세트를 이 방식으로 수행하는 것은 원자 동작들 유닛 (14) 으로 하여금, 심지어 동시에 실행하는 쓰레드들 및/또는 태스크들이 기록된 패킷 엔트리들을 파이프에 동시에 커밋하려고 시도할 수도 있는 경우에도 적합한 파이프의 기능을 보장하는 원자 커밋 기록 지령을 구현 가능하게 한다. 원자 커밋 기록 지령의 예시적인 구현예에 관한 추가적인 세부 사항들이 아래에서 도 3 에 대해서 설명된다.
- [0119] 파이프로부터 데이터를 판독하기 위해, 쓰레드 (20A) 는 원자 예약 판독 지령을 원자 동작들 유닛 (14) 으로 발할 수도 있다. 원자 예약 판독 지령은 데이터가 판독될 파이프를 나타내는 파라미터 및 판독용으로 예약될 패킷 엔트리들의 개수를 나타내는 파라미터 중 하나 (또는, 양쪽) 를 포함할 수도 있다. 데이터가 판독될 파이프를 나타내는 파라미터는 일부 예들에서, 파이프 어드레스 파라미터일 수도 있다. 예를 들어, 파이프 어드레스 파라미터는 파이프 상태 파라미터들 (22) 이 액세스되고 있는 특정의 파이프에 대해 저장되는 메모리 서브시스템 (예컨대, 데이터 저장 시스템 (16)) 에서의 로케이션에 대응하는 메모리 어드레스일 수도 있다. 파이프 어드레스 파라미터는 데이터 저장 시스템 (16) 이 하나 보다 많은 파이프에 대한 데이터를 포함할 때 데이터 저장 시스템 (16) 에서 액세스할 파이프를 식별하고, 파이프 데이터가 저장되는 데이터 저장 시스템 (16) 에서의 로케이션을 식별하기 위해 사용될 수도 있다.
- [0120] 원자 예약 판독 지령을 수신하는 것에 응답하여, 원자 동작들 유닛 (14) 은 원자 예약 판독 동작을 수행할 수도 있다. 원자 예약 판독 동작은 본 개시물에서 설명되는 바와 같이 2개 이상의 파이프 상태 파라미터들을 수

정하는 원자 판독-수정-기록 동작의 일 예일 수도 있다.

[0121] 원자 예약 판독 동작을 수행하기 위해, 원자 동작들 유닛 (14)은 데이터 저장 시스템 (16)으로부터 파이프 상태 파라미터들을 획득하고, 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가지는지 여부를 결정할 수도 있다. 파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가진다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 판독 예약이 파이프에 대해 이루어졌다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가지지 않는다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록 (즉, 판독 예약이 파이프에 대해 이루어지지 않았다는 것을 수정된 값들이 반영하도록) 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0122] 파이프 상태 파라미터들에 대한 수정된 값들을 결정한 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 (22)에 대한 수정된 값들을 데이터 저장 시스템 (16)에 기록할 수도 있다. 원자 예약 판독 지령에 기인하는 기능은 일부 예들에서, 인터럽트불가능한 동작들의 세트의 일부로서 수행될 수도 있다. 다시 말해서, 원자 동작들 유닛 (14)은 다른 원자 동작 지령들 (예컨대, 원자 예약 판독 지령을 발한 태스크 또는 쓰레드와 동시에 실행하고 있는 태스크들 및/또는 쓰레드들에 의해 발해지는 다른 원자 동작 지령들)과 연관될 수도 있는 다른 동작들을 수행하기 위해 실행을 인터럽트함이 없이 원자 예약 판독 지령에 기인하는 기능을 시작부터 끝까지 수행할 수도 있다. 인터럽트불가능한 동작들의 세트를 이 방식으로 수행하는 것은 원자 동작들 유닛 (14)으로 하여금, 심지어 동시에 실행하는 쓰레드들 및/또는 태스크들이 파이프를 판독용으로 동시에 예약하려고 시도할 수도 있는 경우에도 적합한 파이프의 기능을 보장하도록 원자 예약 판독 지령을 구현가능하게 한다.

원자 예약 판독 지령의 예시적인 구현예에 관련한 추가적인 세부 사항들이 아래에서 도 4에 대해서 설명된다.

[0123] 쓰레드 (20A)는 판독 예약이 성공적이었는지 여부를 결정할 수도 있다. 판독 예약이 성공적이었는지 여부를 결정하기 위해, 쓰레드 (20A)는 일부 예들에서, 반환된 파이프 상태 파라미터들에 기초하여, 원자 동작들 유닛 (14)이 아래에서 더 상세히 설명되는 것과 동일한 결정을 행할 수 있는 방법과 유사한 방법으로, 파이프가 판독 예약 요청을 지원하기에 충분한 비판독된 및 미예약된 패킷 엔트리들을 가지는지 여부를 결정할 수도 있다. 반환된 파이프 상태 파라미터들은 예약 판독 동작의 실행 전 파이프의 상태를 나타낼 수도 있다.

[0124] 판독 예약이 성공적이지 않았으면, 쓰레드 (20A)는 판독 예약 동작의 완료에 응답하여 데이터를 파이프로부터 판독하거나 또는 판독된 패킷들을 파이프에 커밋하지 않을 수도 있다. 일부 예들에서, 판독 예약이 성공적이지 않았으면, 쓰레드 (20A)는 여러 조건을 가지는 실행을 종료할 수도 있다. 추가적인 예들에서, 판독 예약이 성공적이지 않았으면, 쓰레드 (20A)는 그 조건을 처리하는 하나 이상의 동작들을 수행할 수도 있다. 예를 들어, 쓰레드 (20A)는 판독 예약을 수행하려고 시도하는 것을 중지하고 계속해서 다른 동작들을 수행할 수도 있다. 다른 예로서, 쓰레드 (20A)는 어떤 시간 기간 이후 판독 예약 동작을 반복하도록 시도할 수도 있다. 이것은 제작자 쓰레드 시간이 파이프에 하나 이상의 패킷들을 기록하는 것을 가능하게 하여, 후속 판독 예약 동작에서 예약될 수 있는 추가적인 패킷 엔트리들을 추가가능하게 한다.

[0125] 판독 예약이 성공적이었으면, 쓰레드 (20A)는 원자 판독 예약 지령을 프로세싱할 때, 원자 동작들 유닛 (14)에 의해 반환된 하나 이상의 파이프 상태 파라미터들에 기초하여, 데이터를 판독할 파이프 데이터 버퍼 (24)의 물리적인 패킷 엔트리들을 결정할 수도 있다. 쓰레드 (20A)가 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터, 및 예약된 패킷 카운트 파라미터를 수신하는 예들에서, 쓰레드 (20A)는 예를 들어, 제 1 패킷 오프셋 파라미터에 대응하는 패킷 엔트리를 이용함으로써, 데이터를 판독할 파이프 데이터 버퍼 (24)의 제 1 물리적인 패킷 엔트리를 결정할 수도 있다. 쓰레드 (20A)는 데이터를 결정된 제 1 물리적인 패킷 엔트리 및 다음 패킷 엔트리들로부터, 예약된 패킷 엔트리들의 개수까지 판독할 수도 있다. 예약된 패킷 엔트리들의 개수가 판독되기 전에 파이프 데이터 버퍼 (24)의 끝에 도달되면, 쓰레드 (20A)는 추가적인 물리적인 패킷 엔트리들을 판독하기 위해 파이프 데이터 버퍼 (24)의 시작으로 순환할 수도 있다.

[0126] 파이프 데이터 버퍼 (24)에서의 예약된 패킷 엔트리들로부터 데이터를 판독하기 위해, 쓰레드 (20A)는 일부 예들에서, 판독될 파이프와 연관되는 파이프 어드레스에 기초하여, 그리고 데이터를 판독할 결정된 물리적인 패킷 엔트리들에 기초하여, 및/또는 반환된 파이프 상태 파라미터들에 기초하여, 패킷 엔트리들 중 하나 이상에 대한 물리 메모리 어드레스를 결정할 수도 있다. 일부 예들에서, 파이프 어드레스는 데이터 저장 시스템

(16) 에 파이프를 할당한 호스트 프로그램에 의해 쓰레드 (20A)에 제공될 수도 있다. 이러한 예들에서, 파이프 어드레스는 일부 예들에서, 계산 유닛 (12B)의 온칩 상수 메모리에 저장될 수도 있다. 예를 들어, 호스트 프로세서 상에서 실행하는 호스트 프로그램은 일부 예들에서, 파이프 어드레스를, 계산 유닛 (12B) 상에서 쓰레드 (20A)의 실행을 호출하는 함수의 인수로서 제공할 수도 있다 (예컨대, 상수 파라미터들이 커널 인수들로서 계산 유닛 (12B)에 제공될 수도 있다).

[0127] 일부 예들에서, 쓰레드 (20A)는 하나 이상의 메모리 판독 요청들을 데이터 저장 시스템 (16)으로 발하여, 데이터 저장 시스템 (16)에 포함되는 예약된 패킷 엔트리들로부터 패킷 데이터를 판독할 수도 있다. 판독 요청들의 각각은 데이터를 판독할 데이터 저장 시스템 (16)에서의 물리적인 로케이션 (예컨대, 메모리 어드레스들)을 규정할 수도 있다. 판독 요청들은 예약된 패킷 엔트리들 모두로부터 패킷 데이터를 일괄하여 판독할 수도 있다.

[0128] 쓰레드 (20A)는 일부 예들에서, 메모리 판독 요청들을, 데이터 저장 시스템 (16) 및/또는 데이터 저장 시스템 (16)를 포함하는 메모리 서브시스템으로 직접 발할 수도 있다. 다시 말해서, 이러한 예들에서, 컴퓨팅 시스템 (10)은 계산 유닛 (12B)과 원자 동작들 유닛 (14)을 바이패스하는 데이터 저장 시스템 (16) 사이의 통신 인터페이스를 포함할 수도 있으며, 쓰레드 (20A)는 메모리 판독 요청들이 데이터 저장 시스템 (16)에 의해 수신되지만 원자 동작들 유닛 (14)을 통과하지 않도록, 그 통신 인터페이스를 통해서 메모리 판독 요청들을 데이터 저장 시스템 (16)으로 발할 수도 있다.

[0129] 패킷 데이터가 파이프 데이터 버퍼 (24)에서의 예약된 패킷 엔트리들로부터 판독되어진 후, 원자 동작들 유닛 (14)은 원자 커밋 판독 지령을 원자 동작들 유닛 (14)으로 발할 수도 있다. 원자 커밋 판독 지령은 쓰레드 (20A)가 파이프의 예약된 패킷 엔트리들로부터 데이터를 판독하는 것을 종료하였다는 것을, 그리고 이러한 패킷 엔트리들이 파이프로부터 제거될 준비가 되어 있다는 것을 표시할 수도 있다.

[0130] 원자 커밋 판독 지령은 데이터가 판독된 파이프를 나타내는 파라미터를 포함할 수도 있다. 데이터가 판독된 파이프를 나타내는 파라미터는 일부 예들에서, 파이프 어드레스 파라미터일 수도 있다. 예를 들어, 파이프 어드레스 파라미터는 파이프 상태 파라미터들 (22)이 액세스되고 있는 특정의 파이프에 대해 저장되는 메모리 서브시스템 (예컨대, 데이터 저장 시스템 (16))에서의 로케이션에 대응하는 메모리 어드레스일 수도 있다. 파이프 어드레스 파라미터는 데이터 저장 시스템 (16)이 하나 보다 많은 파이프에 대한 데이터를 포함할 때 데이터 저장 시스템 (16)에서 액세스할 파이프를 식별하고, 파이프 데이터가 저장되는 데이터 저장 시스템 (16)에서의 로케이션을 식별하기 위해 사용될 수도 있다.

[0131] 원자 커밋 판독 지령을 수신하는 것에 응답하여, 원자 동작들 유닛 (14)은 원자 커밋 판독 동작을 수행할 수도 있다. 원자 커밋 판독 동작은 본 개시물에서 설명되는 바와 같이 2개 이상의 파이프 상태 파라미터들을 수정하는 원자 판독-수정-기록 동작의 일 예일 수도 있다.

[0132] 원자 커밋 판독 동작을 수행하기 위해, 원자 동작들 유닛 (14)은 데이터 저장 시스템 (16)으로부터 파이프 상태 파라미터들을 획득할 수도 있다. 원자 예약 판독 동작과는 달리, 원자 동작들 유닛 (14)은 일부 예들에서, 원자 커밋 판독 동작을 수행할 때, 데이터 저장 시스템 (16)으로부터 획득되는 임의의 파이프 상태 파라미터들을 쓰레드 (20A)로 반환하지 않을 수도 있다.

[0133] 원자 동작들 유닛 (14)은 원자 커밋 판독 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 수정된 값이 반영하도록, 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시킬 수도 있다. 일부 예들에서, 제 1 파이프 파라미터는 예약 카운트 파라미터일 수도 있으며, 원자 동작들 유닛 (14)은 파이프가 원자 커밋 판독 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 반영하기 위해, 예약 카운트 파라미터를 감분시킬 수도 있다.

[0134] 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프에 대한 모든 예약들이 커밋되었는지 여부를 추가로 결정할 수도 있다. 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 커밋된 엔트리들이 파이프로부터 제거되었다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시킬 수도 있다. 일부 예들에서, 원자 동작들 유닛 (14)은 패킷 카운트 파라미터를 예약된 패킷 카운트 파라미터와 동일한 양 만큼 감분시킬 수도 있으며, 커밋된 엔트리들이 파이프로부터 제거되었다는 것을 반영하기 위해 예약된 패킷 카운트 파라미터를 제로로 리셋할 수도 있다. 이러한 예들에서, 원자 동작들 유닛 (14)은 또한 제 1 패킷 오프셋 파라미터를 예약

된 패킷 카운트 파라미터와 동일한 양 만큼 증분시킬 수도 있다. 제 1 패킷 오프셋 파라미터가 파이프 데이터 베퍼 (24) 의 끝을 지나서 증분되도록 예약된 패킷 엔트리들의 개수가 초래하면, 제 1 패킷 오프셋 파라미터는 파이프 데이터 베퍼 (24) 의 시작까지 링 베퍼-형 방식으로 순환될 수도 있다. 파이프에 대한 모든 예약들이 커밋되지 않았다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14) 은 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록, 제 1 파이프 상태 파라미터 (예컨대, 예약 카운트 파라미터) 가 아닌, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0135] 파이프 상태 파라미터들에 대한 수정된 값들을 결정한 후, 원자 동작들 유닛 (14) 은 파이프 상태 파라미터들 (22) 에 대한 수정된 값들을 데이터 저장 시스템 (16) 에 기록할 수도 있다. 원자 커밋 판독 지령에 기인하는 기능은 일부 예들에서, 인터럽트불가능한 동작들의 세트의 일부로서 수행될 수도 있다. 다시 말해서, 원자 동작들 유닛 (14) 은 다른 원자 동작 지령들 (예컨대, 원자 커밋 판독 지령을 발한 태스크 또는 쓰레드와 동시에 실행하고 있는 태스크들 및/또는 쓰레드들에 의해 발해지는 다른 원자 동작 지령들) 과 연관될 수도 있는 다른 동작들을 수행하기 위해 실행을 인터럽트함이 없이 원자 판독 커밋 지령에 기인하는 기능을 시작부터 끝까지 수행할 수도 있다. 인터럽트불가능한 동작들의 세트를 이 방식으로 수행하는 것은 원자 동작들 유닛 (14) 으로 하여금, 심지어 동시에 실행하는 쓰레드들 및/또는 태스크들이 판독된 패킷 엔트리들을 파이프에 동시에 커밋하려고 시도할 수도 있는 경우에도 적합한 파이프의 기능을 보장하는 원자 커밋 판독 지령을 구현 가능하게 한다. 원자 커밋 판독 지령의 예시적인 구현예에 관련한 추가적인 세부 사항들이 아래에서 도 5 에 대해서 설명된다.

[0136] 파이프는 패킷 엔트리들의 각각이 파이프에 대한 데이터 패킷을 저장하도록 구성가능한 복수의 패킷 엔트리들을 포함할 수도 있다. 파이프의 사이즈는 파이프에 포함되는 패킷 엔트리들의 개수로서 지정할 수도 있다. 일부 예들에서, 파이프들을 지원하는 API 는 프로그래머로 하여금 파이프의 임의의 사이즈를 특정의 한계들 내에서 규정가능하게 할 수도 있다. 그러나, API 를 지원하는 런타임 프로그램 (예컨대, 드라이버) 은 일부 예들에서, 파이프에 할당된 물리 메모리에 포함되는 패킷 엔트리들의 개수가 프로그래머에 의해 규정된 패킷 엔트리들의 개수와 상이하도록 파이프를 구현할 수도 있다. 예를 들어, 런타임 프로그램은 파이프들의 각각에 할당되는 물리 메모리에 포함되는 패킷 엔트리들의 개수가 항상 2 의 거듭제곱이 되도록 파이프들을 구현할 수도 있다. 비록 런타임 프로그램에 의해 물리 메모리에 할당된 패킷 엔트리들의 개수가 호스트 프로그램에 의해 규정된 패킷 엔트리들의 개수와 상이할 수도 있지만, 런타임 프로그램은, 파이프를 이용하고 있는 태스크들의 관점에서, 단지 호스트 프로그램에 의해 규정된 패킷 엔트리들의 개수만을 가지는 것처럼 파이프가 보이게 되는 방식으로 파이프의 상태를 제어할 수도 있다.

[0137] 예를 들어, 각각의 파이프에 할당되는 물리 메모리에 포함되는 패킷 엔트리들의 개수가 항상 2 의 거듭제곱이 되도록 런타임 프로그램이 파이프들을 구현하면 그리고 프로그래머가 11 개의 패킷 엔트리들의 파이프 사이즈를 규정하면, 런타임 프로그램은 물리 메모리가 파이프에 대해 16 개의 패킷 엔트리들을 포함하도록, 물리 메모리를 할당할 수도 있다. 이 예에서, 런타임 프로그램은 16 개의 패킷 엔트리들이 물리 메모리에 파이프에 대해 할당되었더라도, 파이프를 이용하고 있는 태스크들의 관점에서 단지 11 개의 패킷 엔트리들을 가지는 것처럼 파이프가 보이게 되는 방식으로 파이프의 상태를 제어할 수도 있다.

[0138] 소스 코드 및/또는 프로그래머의 관점에서 볼 때 파이프의 패킷 엔트리들은 본원에서 논리적 패킷 엔트리들로서 지정될 수도 있으며, 파이프에 대해 물리 메모리에 실제로 할당되는 패킷 엔트리들은 본원에서 물리적인 패킷 엔트리들로서 지정될 수도 있다. 이와 유사하게, 소스 코드 및/또는 프로그래머에 의해 규정되는 파이프의 사이즈 (예컨대, 패킷 엔트리들의 개수) 는 본원에서 파이프의 논리적 사이즈로서 지정될 수도 있으며, 파이프에 대해 실제로 메모리에 할당되는 패킷 엔트리들의 개수는 본원에서 파이프의 물리적인 사이즈로서 지정될 수도 있다.

[0139] 원자 예약 기록 지령으로 돌아가면, 데이터가 기록될 파이프를 나타내는 파라미터 및 판독용으로 예약될 패킷 엔트리들의 개수를 나타내는 파라미터를 포함하는 것에 더해서, 원자 예약 기록 지령은 일부 예들에서, 파이프의 논리적 사이즈를 나타내는 파라미터 및/또는 파이프의 물리적인 사이즈 마이너스 1 을 나타내는 파라미터를 더 포함할 수도 있다. 원자 예약 기록 동작은 파이프의 논리적 사이즈를 나타내는 파라미터를 이용하여, 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 결정할 수도 있다. 예를 들어, 원자 동작들 유닛 (14) 은 파이프의 논리적 사이즈로부터 파이프에 커밋된 패킷들의 총 개수 및 판독용으로 예약된 패킷 엔트리들의 총 개수를 감산하고, 그 차이가 판독용으로 예약되도록 예약 요청에 의해 요청된 패킷들의 개수보다 크거나 또는 동일한지 여부를 결정할 수도 있다. 그 차이가 판독용으로 예약되도록 예약 요청에 의해 요청된 패킷들의 개수보다 크거나 또는 동일하면, 원자 동작들 유닛 (14) 은

파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있다고 결정할 수도 있다. 그렇지 않고, 그 차이가 판독용으로 예약되도록 예약 요청에 의해 규정된 패킷들의 개수보다 크거나 또는 동일하지 않으면, 원자 동작들 유닛 (14)은 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 갖지 않는다고 결정할 수도 있다.

[0140] 이와 유사하게, 원자 예약 판독 지령에 대해, 일부 예들에서, 데이터가 판독될 파이프를 나타내는 파라미터 및 판독용으로 예약될 패킷 엔트리들의 개수를 나타내는 파라미터를 포함하는 것에 더해서, 원자 예약 판독 지령은 파이프의 논리적 사이즈를 나타내는 파라미터 및/또는 파이프의 물리적인 사이즈 마이너스 1 을 나타내는 파라미터를 더 포함할 수도 있다. 원자 예약 판독 동작은 파이프의 논리적 사이즈를 나타내는 파라미터를 이용하여, 파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가지는지 여부를 결정할 수도 있다. 예를 들어, 원자 동작들 유닛 (14)은 판독용으로 이전에 예약된 패킷들의 개수 및 판독용으로 예약되도록 예약 요청에 의해 요청된 패킷들의 개수를 가산하고, 그 합계가 파이프에 저장된 패킷들의 총 개수 (즉, 파이프에 기록되고 커밋되는 패킷들의 총 개수) 보다 작거나 또는 동일한지 여부를 결정할 수도 있다. 그 합계가 파이프에 저장된 패킷들의 총 개수보다 작거나 또는 동일하면, 원자 동작들 유닛 (14)은 파이프가 판독 예약 요청을 지원하기에 충분한 패킷 엔트리들을 가지고 있다고 결정할 수도 있다. 그렇지 않고, 그 합계가 파이프에 저장된 패킷들의 총 개수보다 작거나 또는 동일하지 않으면, 원자 동작들 유닛 (14)은 파이프가 판독 예약 요청을 지원하기에 충분한 패킷 엔트리들을 가지지 않는다고 결정할 수도 있다.

[0141] 또한, 원자 커밋 판독 지령에 대해, 일부 예들에서, 데이터가 판독된 파이프를 나타내는 파라미터를 포함하는 것에 더하여, 원자 판독 지령은 파이프의 물리적인 사이즈 마이너스 1 을 나타내는 파라미터를 더 포함할 수도 있다. 파이프의 물리적인 사이즈 마이너스 1 파라미터는 모든 판독 예약들이 커밋되었을 때 제 1 패킷 오프셋 파라미터를 업데이트하기 위해 원자 커밋 판독 지령에 의해 사용될 수도 있다. 예를 들어, 파이프의 물리적인 사이즈 마이너스 1 파라미터는 제 1 패킷 오프셋 파라미터가 증분되는 양이 제 1 패킷 오프셋 파라미터를 파이프 데이터 버퍼 (24)의 끝을 지나서 증분시킬 때 제 1 패킷 오프셋 파라미터가 물리적인 패킷 엔트리들의 끝으로부터 물리적인 패킷 엔트리들의 시작까지 순환가능하도록 하는 모듈로 연산을 구현하기 위해 사용될 수도 있다.

[0142] 일부 예들에서, 데이터가 기록될 파이프를 나타내는 파라미터, 파이프의 사이즈를 나타내는 파라미터, 및 파이프의 사이즈 마이너스 1 을 나타내는 파라미터 중 하나 이상이 데이터 저장 시스템 (16)에 파이프를 할당한 호스트 프로그램에 의해 쓰레드 (18A) 및/또는 쓰레드 (20A)에 제공될 수도 있다. 이러한 파라미터들이 파이프가 할당될 때 결정될 수도 있으며 쓰레드들 (18, 20)의 실행 전반을 통해서 일정하게 유지할 수도 있기 때문에, 이러한 파라미터들은 상수 파라미터들로서 지정될 수도 있다. 이러한 예들에서, 상수 파라미터들은 일부 예들에서, 계산 유닛 (12A) 및/또는 계산 유닛 (12B)의 온침 상수 메모리에 저장될 수도 있다. 호스트 프로세서 상에서 실행하는 호스트 프로그램은 일부 예들에서, 계산 유닛 (12A) 상의 쓰레드 (18A) 및/또는 계산 유닛 (12B) 상의 쓰레드 (20A)의 실행을 호출하는 함수의 인수들로서 상수 파라미터들을 제공할 수도 있다 (예컨대, 상수 파라미터들은 커널 인수들로서 쓰레드 (18A) 및 쓰레드 (20A)에 제공될 수도 있다).

[0143] 원자 예약 지령들 및 원자 커밋 지령들을 위해 원자 동작들 유닛 (14)에 의해 수행되는 예시적인 기능에 관련한 추가적인 세부 사항들이 도 2 내지 도 5에 대해 이하 설명될 것이다. 도 2 내지 도 5에 예시된 원자 동작 지령들의 각각에 대한 기능은 하드웨어로 구현될 수도 있으며, 지령들의 각각은 본 개시물에서 설명되는 바와 같이 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작의 일 예일 수도 있다.

[0144] 도 2는 본 개시물에 따른, 예시적인 원자 예약 기록 지령의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다. 의사-코드의 라인들 1-4는 이 예에서 액세스되거나 및/또는 수정될 파이프 상태 파라미터들을 어레이의 개개의 엘리먼트들에 맵핑한다. 파이프 상태 파라미터들은 파이프에 대한 시작 패킷을 저장하는 패킷 엔트리를 나타내는 제 1 패킷 오프셋 파라미터 (즉, "FST\_PKT"), 얼마나 많은 패킷들이 현재 파이프에 저장되어 있는지를 나타내는 패킷 카운트 파라미터 (즉, "PKT\_CNT"), 얼마나 많은 패킷 엔트리들이 현재 파이프에 예약되어 있는지를 나타내는 예약된 패킷 카운트 파라미터 (즉, "RES\_CNT"), 및 얼마나 많은 비커밋된 예약들이 현재 파이프에 펜딩 중인지를 나타내는 예약 카운트 파라미터 (즉, "NUM\_RES")를 포함한다.

그 맵핑은 제 1 패킷 오프셋 파라미터를 어레이의 제 1 엘리먼트에, 패킷 카운트 파라미터를 어레이의 제 2 엘리먼트에, 등등으로 맵핑한다.

- [0145] 의사-코드의 라인들 5-11 은 원자 예약 기록 지령에 대한 인터페이스를 규정한다. 라인 5 는 지령의 이름을 규정하며, 라인들 7-10 은 지령을 호출할 때 사용될 인수들 또는 파라미터들을 규정한다. 라인 7 은 액세스 될 파이프에 대한 파이프 상태 파라미터들 (22) 을 저장하는 데이터 저장 시스템 (16) 에서의 메모리의 영역을 가리킬 수도 있는 파이프 어드레스 파라미터를 규정한다. 일부 예들에서, 파이프 어드레스 파라미터는 머신 어드레스에 기초하는 32-비트 어드레스 또는 64-비트 어드레스일 수도 있다. 라인 8 은 데이터 저장 시스템 (16) 에 파이프에 할당되는 물리적인 패킷 엔트리들의 개수 마이너스 1 에 대응하는 물리 파이프 사이즈 마이너스 1 파라미터를 규정한다. 라인 9 는 파이프가 생성되었을 때 호스트 프로그램으로부터 소스 코드에 의해 요청된 논리 파이프 엔트리들의 수를 나타내는 논리 파이프 사이즈 파라미터 (즉, 프로그래머의 관점에서 파이프의 사이즈) 를 규정한다. 라인 10 은 요청하는 태스크가 판독용으로 예약하도록 요청하고 있는 패킷들의 개수를 나타내는 요청된 패킷들 파라미터의 개수를 규정한다. 일부 예들에서, 원자 예약 기록 지령에 대한 인터페이스는 물리 파이프 사이즈 마이너스 1 파라미터를 포함하지 않을 수도 있다.
- [0146] 원자 예약 기록 지령이 쓰레드에 의해 호출될 때, 원자 동작들 유닛 (14) 은 데이터 저장 시스템 (16) 으로부터 이 예에서 사용되는 4개의 파이프 상태 파라미터들을 판독하는 판독 동작을 수행할 수도 있다 (라인 13). 도 2 에 나타낸 바와 같이, 원자 동작들 유닛 (14) 은 데이터 저장 시스템 (16) 으로부터 모든 4개의 파이프 상태 파라미터들을 포함하는 어레이를 판독할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32 비트일 수도 있으며, 판독 동작은 단일 128-비트 로드 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 판독 동작 및/또는 로드 동작은 로드 원자 동작 동안 액세스되거나 및/또는 수정되는 파이프 상태 파라미터들의 모두를 로드하기 위해 데이터 저장 시스템 (16) 에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 판독 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정 가능할 수도 있다.
- [0147] 파이프 상태 파라미터들을 판독한 후, 원자 동작들 유닛 (14) 은 데이터 저장 시스템 (16) 으로부터 판독한 파이프 상태 파라미터들 중 3 개를 요청하는 쓰레드로 반환할 수도 있다 (라인들 14-15). 이 예에서 반환되는 3개의 파이프 상태 파라미터들은 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터 및 예약된 패킷 카운트 파라미터이다. 요청하는 쓰레드 (예컨대, 쓰레드 (18A)) 는 반환된 파이프 상태 파라미터들에 기초하여, 데이터를 기록할 파이프 데이터 버퍼 (24) 의 물리적인 패킷 엔트리들을 결정할 수도 있다. 예를 들어, 쓰레드 (18A) 는 예를 들어, 이들 파라미터들의 각각의 값들을 함께 가산함으로써, 데이터를 기록할 파이프 데이터 버퍼 (24) 의 제 1 물리적인 패킷 엔트리를 결정할 수도 있다. 쓰레드 (18A) 는 데이터를 결정된 제 1 물리적인 패킷 엔트리 및 다음 패킷 엔트리들에, 예약된 패킷 엔트리들의 개수까지 기록할 수도 있다. 예약된 패킷 엔트리들의 개수가 기록되기 전에 파이프 데이터 버퍼 (24) 의 끝에 도달되면, 쓰레드 (18A) 는 추가적인 물리적인 패킷 엔트리들을 기록하기 위해 파이프 데이터 버퍼 (24) 의 시작으로 순환할 수도 있다.
- [0148] 파이프 상태 파라미터들을 판독한 후, 원자 동작들 유닛 (14) 은 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 수정 동작을 수행할 수도 있다 (라인들 16-20). 수정 동작을 수행하기 위해, 원자 동작들 유닛 (14) 은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 결정할 수도 있다 (라인 16). 좀더 구체적으로, 이 예에서, 원자 동작들 유닛 (14) 은 논리 파이프 사이즈 마이너스 패킷 카운트 파라미터 마이너스 예약된 패킷 카운트 파라미터가 요청된 패킷들의 개수보다 크거나 또는 동일한지 여부를 결정할 수도 있다 (라인 16). 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14) 은 라인들 17-20 에서 규정된 바와 같이 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.
- [0149] 예를 들어, 원자 동작들 유닛 (14) 은 요청된 패킷들의 개수와 동일한 양 만큼 예약된 패킷 카운트 파라미터를 증분시키는 증분 동작을 수행할 수도 있으며 (라인 17), 요청된 패킷들의 개수가 제로와 동일하지 않으면, 원자 동작들 유닛 (14) 은 예약 카운트 파라미터를 1 만큼 증분시키는 1 증분 동작을 수행할 수도 있다 (라인들 18-20). 라인 18 의 if-절에서 하나의 동작에 의한 증분의 포함은 요청된 패킷들의 개수가 제로와 동일할 때 예약 카운트 파라미터가 수정되지 않도록 보장할 수도 있다. 사용자는 요청된 패킷들의 개수를 제로로 설정하고, 원자 예약 기록 지령을 발하여, 예약을 행함이 없이 (즉, 기록 예약이 일어났다는 것을 반영하기 위해 파라미터들을 수정함이 없이) 파이프의 현재의 상태를 획득할 수도 있다.
- [0150] 개념적으로, 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14) 은 요청된 패킷들의 개수가 제로보다 큰지 여부에 따라서 상이한 방법으로 파이프 상태 파라미터들을 수정할 수도 있다. 요청된 패킷들의 개수가 제로보다 크면, 원자 동작

들 유닛 (14) 은 기록 예약이 파일에 대해 이루어졌다는 것을 수정된 값들이 반영하도록, 파일 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다 (라인들 17-19). 요청된 패킷들의 개수가 제로와 동일하면, 원자 동작들 유닛 (14) 은 수정된 값들이 파일 상태 파라미터들에 대한 이전 값들과 동일하도록 (즉, 기록 예약이 파일에 대해 이루어지지 않았다는 것을 수정된 값들이 반영하도록) 파일 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다 (라인들 17-19).

[0151] 파일가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 갖지 않는다고 결정하는 것에 응답하여 또는 예약 카운트 파라미터가 제로와 동일한 것에 응답하여, 원자 동작들 유닛 (14) 은 파일 상태가 동일하게 유지하도록 파일 상태 파라미터 값들을 업데이트하지 않을 수도 있다. 파일 상태가 동일하게 유지하도록 파일 상태 파라미터 값들을 업데이트하지 않는 것은, 수정된 값들이 파일 상태 파라미터들에 대한 이전 값들과 동일하도록 (즉, 기록 예약이 파일에 대해 이루어지지 않았다는 것을 수정된 값들이 반영하도록) 파일 상태 파라미터들에 대한 수정된 값들을 발생시키는 것에 대응할 수도 있다.

[0152] 수정된 파일 상태 파라미터 값들을 발생시킨 후, 원자 동작들 유닛 (14) 은 이 예에서 사용되는 4개의 파일 상태 파라미터들을 데이터 저장 시스템 (16) 에 기록하는 기록 동작 (예컨대, 저장 동작) 을 수행할 수도 있다 (라인 21). 도 2 에 나타낸 바와 같이, 원자 동작들 유닛 (14) 은 모든 4개의 파일 상태 파라미터들을 포함하는 어레이를 데이터 저장 시스템 (16) 에 기록할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32 비트일 수도 있으며, 기록 동작은 단일 128-비트 저장 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 기록 동작 및/또는 저장 동작은 원자 동작 동안 액세스되거나 및/또는 수정되는 파일 상태 파라미터들 모두를 저장하기 위해 데이터 저장 시스템 (16) 에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 기록 트랜잭션을 필요로 함이 없이 다수의 파일 상태 파라미터들을 수정 가능할 수도 있다. 원자 예약 기록 동작의 프로세싱이 이제 완료된다.

[0153] 일부 예들에서, 원자 예약 기록 지령의 신택스는 다음 유형을 취할 수도 있다:

PIPE\_RSV\_WRITE Dest, Src0, Src1

[0155] 여기서, Dest 는 원자 예약 기록 지령에 의해 반환되는 출력 파일 상태 파라미터들을 수신할 3개의 GPR 레지스터들의 그룹에서 제 1 GPR 의 범용 레지스터 (GPR) 식별 (ID) 에 대응하며, Src0 은 요청된 패킷들의 개수를 포함하는 레지스터에 대한 GPR ID 에 대응하며, Src1 은 물리 파일 사이즈 마이너스 1, 논리 파일 사이즈, 및 파일 어드레스를 포함하는 4개의 상수 레지스터들의 그룹에서 제 1 레지스터에 대한 상수 ID 에 대응한다.

파일 어드레스는 파일 상태 파라미터들 (22) 을 가리키는 32-비트 또는 64-비트 어드레스를 포함하는 하나 또는 2개의 레지스터들에 저장될 수도 있다. 일부 예들에서, 물리 파일 사이즈 마이너스 1 이 2 의 거듭제곱 마이너스 1 이 되도록 물리 파일 사이즈는 2 의 거듭제곱일 수도 있다.

[0156] 위에서 설명한 바와 같이, 원자 예약 기록 동작은 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터, 및 예약된 패킷 카운트 파라미터를 반환할 수도 있다. 이와 같이, 지령은 반환 원자의 일 예일 수도 있다. 일부 예들에서, 계산 유닛들 (12) 중 하나 (또는, 양쪽) 는 각각의 퀘드 레지스터 그룹이 4개의 레지스터들을 포함하는 하나 이상의 퀘드 레지스터 그룹들을 포함할 수도 있으며, 4개의 레지스터들에 대한 데이터는 단일 캐시 트랜잭션 및/또는 메모리 트랜잭션의 일부로서 메모리로부터 판독되거나 및/또는 그에 저장될 수 있다. 이러한 예들에서, 원자 예약 기록 동작에 의해 반환되는 3개의 파일 상태 파라미터들은, 일부 예들에서, 퀘드 레지스터 그룹 내 3개의 연속된 레지스터들에서 반환된다. 이러한 방법으로, 3개의 파일 상태 파라미터들은 단일 캐시 트랜잭션 및/또는 메모리 트랜잭션의 일부로서 계산 유닛들 (12) 중 하나 (또는, 양쪽) 에 의해 폐치될 수도 있다.

[0157] 일부 예들에서, 원자 예약 기록 지령은 OpenCL™ C 사양에 따라서 정의된 reserve\_write\_pipe 지령을 구현하기 위해 사용될 수도 있다. 추가적인 예들에서, 예약 카운트 파라미터를 제로와 동일하게 설정함으로써, 원자 예약 기록 지령은 예약을 그 파일에 추가함이 없이 파일의 현재의 상태를 반환할 수도 있다. 이러한 방법으로, 원자 예약 기록 지령은 또한 OpenCL™ C 사양에 따라서 정의된 get\_pip\_num\_packets 지령을 구현하기 위해 사용될 수도 있다.

[0158] 도 3 은 본 개시물에 따른, 예시적인 원자 커밋 기록 지령의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다. 의사-코드의 라인들 1-4 는 이 예에서 액세스되거나 및/또는 수정될 파일 상태 파라미터들을 어레이의 개개의 엘리먼트들에 맵핑한다. 파일 상태 파라미터들 및 맵핑은 도 2 에 대해 위에서 설명한 것과 동일하다.

- [0159] 의사-코드의 라인 5 는 지령을 호출할 때 사용될 지령 및 인수 또는 파라미터의 이름을 포함하는 원자 커밋 기록 지령에 대한 인터페이스를 규정한다. 좀더 구체적으로, 원자 커밋 기록 지령은 파이프 상태 파라미터들(22) 을 저장하는 데이터 저장 시스템(16) 에서 메모리의 영역을 가리키는 파이프 어드레스 파라미터를 이용할 수도 있다. 일부 예들에서, 파이프 어드레스 파라미터는 머신 어드레스에 기초하는 32-비트 어드레스 또는 64-비트 어드레스일 수도 있다.
- [0160] 원자 커밋 기록 지령이 쓰레드에 의해 호출될 때, 원자 동작들 유닛(14) 은 데이터 저장 시스템(16) 으로부터 이 예에서 사용되는 4개의 파이프 상태 파라미터들을 판독하는 판독 동작을 수행할 수도 있다(라인 7). 도 3 에 나타낸 바와 같이, 원자 동작들 유닛(14) 은 데이터 저장 시스템(16) 으로부터 모든 4개의 파이프 상태 파라미터들을 포함하는 어레이를 판독할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32 비트일 수도 있으며, 판독 동작은 단일 128-비트 로드 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 판독 동작 및/또는 로드 동작은 로드 원자 동작 동안 액세스되거나 및/또는 수정되는 파이프 상태 파라미터들의 모두를 로드하기 위해 데이터 저장 시스템(16) 에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 판독 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정 가능할 수도 있다.
- [0161] 파이프 상태 파라미터들을 판독한 후, 원자 동작들 유닛(14) 은 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 수정 동작을 수행할 수도 있다(라인들 8-12). 수정 동작을 수행하기 위해, 원자 동작들 유닛(14) 은 예약 카운트 파라미터를 1 만큼 감분시키는 1 감분 동작을 수행할 수도 있다(라인 8). 예약 카운트 파라미터를 1 만큼 감분하는 것은 원자 커밋 기록 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 수정된 값이 반영하도록, 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 것에 대응할 수도 있다.
- [0162] 원자 동작들 유닛(14) 은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프에 대한 모든 예약들이 커밋되었는지 여부를 추가로 결정할 수도 있다(라인 9). 좀더 구체적으로, 이 예에서, 예약 카운트 파라미터를 1 만큼 감분한 후, 원자 동작들 유닛(14) 은 예약 카운트 파라미터가 제로와 동일한지 여부를 결정할 수도 있다.
- [0163] 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 원자 동작들 유닛(14) 은 커밋된 엔트리들이 파이프에 배치되어 있다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시킬 수도 있다(라인들 10-11). 좀더 구체적으로, 이 예에서, 원자 동작들 유닛(14) 은 패킷 카운트 파라미터를 예약된 패킷 카운트 파라미터와 동일한 양 만큼 증분시키는 증분 동작(라인 10), 및 예약된 패킷 카운트 파라미터를 제로로 리셋하는 리셋 동작을 수행할 수도 있다(라인 11).
- [0164] 파이프에 대한 모든 예약들이 커밋되지 않았다고 결정하는 것에 응답하여, 원자 동작들 유닛(14) 은 나머지 파이프 상태 파라미터들이 동일하게 유지하도록, 이미 업데이트된 예약 카운트 파라미터 외에 다른 파이프 상태 파라미터 값들을 업데이트하지 않을 수도 있다. 나머지 파이프 상태 파라미터들이 동일하게 유지하도록 나머지 파이프 상태 파라미터 값들을 업데이트하지 않는 것은, 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록, 나머지 파이프 상태 파라미터들에 대해 수정된 값들을 발생시키는 것에 대응할 수도 있다.
- [0165] 수정된 파이프 상태 파라미터 값들을 발생시킨 후, 원자 동작들 유닛(14) 은 이 예에서 사용되는 4개의 파이프 상태 파라미터들을 데이터 저장 시스템(16) 에 기록하는 기록 동작(예컨대, 저장 동작) 을 수행할 수도 있다(라인 13). 도 3 에 나타낸 바와 같이, 원자 동작들 유닛(14) 은 모든 4개의 파이프 상태 파라미터들을 포함하는 어레이를 데이터 저장 시스템(16) 에 기록할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32 비트일 수도 있으며, 기록 동작은 단일 128-비트 저장 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 기록 동작 및/또는 저장 동작은 원자 동작 동안 액세스되거나 및/또는 수정되는 파이프 상태 파라미터들 모두를 저장하기 위해 데이터 저장 시스템(16) 에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 기록 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정 가능할 수도 있다. 원자 커밋 기록 지령의 프로세싱이 이제 완료된다.
- [0166] 일부 예들에서, 원자 커밋 기록 지령의 선택스는 다음의 유형을 취할 수도 있다:
- [0167] PIPE\_CMT\_WRITE Src0
- [0168] 여기서, 어떤 Dest 레지스터도 규정되지 않으며, Src0 은 물리 파이프 사이즈 마이너스 1, 논리 파이프 사이즈,

및 파일 어드레스를 포함하는 4개의 상수 레지스터들의 그룹에서 제 1 레지스터에 대한 상수 ID에 대응한다. 파일 어드레스는 파일 상태 파라미터들 (22)을 가리키는 32-비트 또는 64-비트 어드레스를 포함하는 하나 또는 2개의 레지스터들에 저장될 수도 있다. 일부 예들에서, 물리 파일 사이즈 마이너스 1 이 2 의 거듭제곱 마이너스 1 이 되도록 물리 파일 사이즈는 2 의 거듭제곱일 수도 있다.

[0169] 원자 커밋 기록 지령은, 이 예에서, 어떤 값들도 반환하지 않는다. 이와 같이, 지령은 비-반환 원자의 일 예일 수도 있다. 일부 예들에서, 원자 커밋 기록 지령은 OpenCL™ C 사양에 따라서 정의된 바와 같은 commit\_write\_pipe 지령을 구현하기 위해 사용될 수도 있다.

[0170] 도 4 는 본 개시물에 따른, 예시적인 원자 예약 판독 지령의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다. 의사-코드의 라인들 1-4 는 이 예에서 액세스되거나 및/또는 수정될 파일 상태 파라미터들을 어레이의 개개의 엘리먼트들에 맵핑한다. 파일 상태 파라미터들 및 맵핑은 도 2 에 대해 위에서 설명한 것과 동일하다.

[0171] 의사-코드의 라인들 5-11 은 원자 예약 판독 지령에 대한 인터페이스를 규정한다. 라인 5 는 지령의 이름을 규정하며, 라인들 7-10 은 지령을 호출할 때 사용될 인수들 또는 파라미터들을 규정한다. 라인 7 은 파일 상태 파라미터들 (22)을 저장하는 데이터 저장 시스템 (16)에서 메모리의 영역을 가리킬 수도 있는 파일 어드레스 파라미터를 규정할 수도 있다. 일부 예들에서, 파일 어드레스 파라미터는 머신 어드레스에 기초하는 32-비트 어드레스 또는 64-비트 어드레스일 수도 있다. 라인 8 은 데이터 저장 시스템 (16)에 파일에 할당되는 물리적인 패킷 엔트리들의 개수 마이너스 1에 대응하는 물리 파일 사이즈 마이너스 1 파라미터를 규정한다. 라인 9 는 파일이 생성되었을 때 호스트 프로그램으로부터 소스 코드에 의해 요청된 논리 파일 엔트리들의 수를 나타내는 논리 파일 사이즈 파라미터 (즉, 프로그래머의 관점에서 파일의 사이즈)를 규정한다. 라인 10 은 요청하는 태스크가 기록용으로 요청하고 있는 패킷들의 개수를 나타내는 요청된 패킷들 파라미터의 개수를 규정한다. 일부 예들에서, 원자 예약 판독 지령에 대한 인터페이스는 물리 파일 사이즈 마이너스 1 파라미터를 포함하지 않을 수도 있다.

[0172] 원자 예약 판독 지령이 쓰레드에 의해 호출될 때, 원자 동작들 유닛 (14)은 데이터 저장 시스템 (16)으로부터 이 예에서 사용되는 4개의 파일 상태 파라미터들을 판독하는 판독 동작을 수행할 수도 있다 (라인 13). 도 4 에 나타낸 바와 같이, 원자 동작들 유닛 (14)은 데이터 저장 시스템 (16)으로부터 모든 4개의 파일 상태 파라미터들을 포함하는 어레이를 판독할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32 비트일 수도 있으며, 판독 동작은 단일 128-비트 로드 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 판독 동작 및/또는 로드 동작은 로드 원자 동작 동안 액세스되거나 및/또는 수정되는 파일 상태 파라미터들의 모두를 로드하기 위해 데이터 저장 시스템 (16)에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 판독 트랜잭션을 필요로 함이 없이 다수의 파일 상태 파라미터들을 수정 가능할 수도 있다.

[0173] 파일 상태 파라미터들을 판독한 후, 원자 동작들 유닛 (14)은 데이터 저장 시스템 (16)으로부터 판독한 파일 상태 파라미터들 중 3 개를 요청하는 쓰레드로 반환할 수도 있다 (라인들 14-15). 이 예에서 반환되는 3개의 파일 상태 파라미터들은 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터 및 예약된 패킷 카운트 파라미터이다. 요청하는 쓰레드 (예컨대, 쓰레드 (20A))는 반환된 파일 상태 파라미터들에 기초하여, 데이터를 판독할 파일 데이터 버퍼 (24)의 물리적인 패킷 엔트리들을 결정할 수도 있다. 예를 들어, 쓰레드 (20A)는 예를 들어, 제 1 오프셋 파라미터에 대응하는 패킷 엔트리를 제 1 패킷 엔트리로서 이용함으로써, 데이터를 판독할 파일 데이터 버퍼 (24)의 제 1 물리적인 패킷 엔트리를 결정할 수도 있다. 쓰레드 (20A)는 데이터를 결정된 제 1 물리적인 패킷 엔트리 및 다음 패킷 엔트리들로부터, 예약된 패킷 엔트리들의 개수까지 판독할 수도 있다. 예약된 패킷 엔트리들의 개수가 판독되기 전에 파일 데이터 버퍼 (24)의 끝에 도달되면, 쓰레드 (20A)는 추가적인 물리적인 패킷 엔트리들을 판독하기 위해 파일 데이터 버퍼 (24)의 시작으로 순환할 수도 있다.

[0174] 파일 상태 파라미터들을 판독한 후, 원자 동작들 유닛 (14)은 파일 상태 파라미터들에 대한 수정된 값을 발생시키는 수정 동작을 수행할 수도 있다 (라인들 16-20). 수정 동작을 수행하기 위해, 원자 동작들 유닛 (14)은 파일 상태 파라미터들 중 하나 이상에 기초하여, 파일이 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가지는지 여부를 결정할 수도 있다 (라인 16). 좀 더 구체적으로, 이 예에서, 원자 동작들 유닛 (14)은 예약된 패킷 카운트 파라미터 및 요청된 패킷들의 개수의 합계가 패킷 카운트 파라미터 미만이거나 또는 동일한지 여부를 결정할 수도 있다 (라인 16). 파일

가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가진다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 라인들 17-20에 규정된 바와 같이 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0175] 예를 들어, 원자 동작들 유닛 (14)은 요청된 패킷들의 개수와 동일한 양 만큼 예약된 패킷 카운트 파라미터를 충분시키는 충분 동작을 수행할 수도 있으며 (라인 17), 요청된 패킷들의 개수가 제로와 동일하지 않으면, 원자 동작들 유닛 (14)은 예약 카운트 파라미터를 1 만큼 충분시키는 1 충분 동작을 수행할 수도 있다 (라인들 18-20). 라인 18의 if-절에서 하나의 동작에 의한 충분의 포함은 요청된 패킷들의 개수가 제로와 동일할 때 예약 카운트 파라미터가 수정되지 않도록 보장할 수도 있다. 사용자는 요청된 패킷들의 개수를 제로로 설정하고, 원자 예약 판독 지령을 발하여, 예약을 행함이 없이 (즉, 판독 예약이 일어났다는 것을 반영하기 위해 파라미터들을 수정함이 없이) 파이프의 현재의 상태를 획득할 수도 있다.

[0176] 개념적으로, 파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가진다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 요청된 패킷들의 개수가 제로보다 큰지 여부에 따라서, 상이한 방법들로, 파이프 상태 파라미터들을 수정할 수도 있다. 요청된 패킷들의 개수가 제로보다 크면, 원자 동작들 유닛 (14)은 판독 예약이 파이프에 대해 이루어졌다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다 (라인들 17-19). 요청된 패킷들의 개수가 제로와 동일하면, 원자 동작들 유닛 (14)은 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록 (즉, 판독 예약이 파이프에 대해 이루어지지 않았다는 것을 수정된 값들이 반영하도록) 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다 (라인들 17-19).

[0177] 파이프가 패킷 데이터가 판독 예약 요청을 지원하기 위해 저장되는 충분한 비판독된 및 미예약된 패킷 엔트리들을 가지지 않는다고 결정하는 것에 응답하여 또는 예약 카운트 파라미터가 제로와 동일한 것에 응답하여, 원자 동작들 유닛 (14)은 파이프 상태가 동일하게 유지하도록, 파이프 상태 파라미터 값들을 업데이트하지 않을 수도 있다. 파이프 상태가 동일하게 유지하도록 파이프 상태 파라미터 값들을 업데이트하지 않는 것은, 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록 (즉, 판독 예약이 파이프에 대해 이루어지지 않았다는 것을 수정된 값들이 반영하도록) 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 것에 대응할 수도 있다.

[0178] 수정된 파이프 상태 파라미터 값들을 발생시킨 후, 원자 동작들 유닛 (14)은 이 예에서 사용되는 4개의 파이프 상태 파라미터들을 데이터 저장 시스템 (16)에 기록하는 기록 동작 (예컨대, 저장 동작)을 수행할 수도 있다 (라인 21). 도 4에 나타낸 바와 같이, 원자 동작들 유닛 (14)은 모든 4개의 파이프 상태 파라미터들을 포함하는 어레이를 데이터 저장 시스템 (16)에 기록할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32비트일 수도 있으며, 기록 동작은 단일 128-비트 저장 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 기록 동작 및/또는 저장 동작은 원자 동작 동안 액세스되거나 및/또는 수정되는 파이프 상태 파라미터들 모두를 저장하기 위해 데이터 저장 시스템 (16)에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 기록 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정 가능할 수도 있다. 원자 예약 판독 지령의 프로세싱이 이제 완료된다

[0179] 일부 예들에서, 원자 예약 판독 지령의 신택스는 다음의 유형을 취할 수도 있다:

[0180] PIPE\_RSV\_READ Dest, Src0, Src1

[0181] 여기서, Dest는 원자 예약 판독 지령에 의해 반환되는 출력 파이프 상태 파라미터들을 수신할 3개의 GPR 레지스터들의 그룹에서 제 1 GPR의 범용 레지스터 (GPR) 식별 (ID)에 대응하며, Src0은 요청된 패킷들의 개수를 포함하는 레지스터에 대한 GPR ID에 대응하며, Src1은 물리 파이프 사이즈 마이너스 1, 논리 파이프 사이즈, 및 파이프 어드레스를 포함하는 4개의 상수 레지스터들의 그룹에서 제 1 레지스터에 대한 상수 ID에 대응한다.

파이프 어드레스는 파이프 상태 파라미터들 (22)을 가리키는 32-비트 또는 64-비트 어드레스를 포함하는 하나 또는 2개의 레지스터들에 저장될 수도 있다. 일부 예들에서, 물리 파이프 사이즈 마이너스 1이 2의 거듭제곱 마이너스 1이 되도록 물리 파이프 사이즈는 2의 거듭제곱일 수도 있다.

[0182] 위에서 설명한 바와 같이, 원자 예약 판독 동작은 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터, 및 예약된 패킷 카운트 파라미터를 반환할 수도 있다. 이와 같이, 지령은 반환 원자의 일 예일 수도 있다. 일부 예들에서, 계산 유닛들 (12) 중 하나 (또는, 양쪽)는 각각의 큐드 레지스터 그룹이 4개의 레지스터들을 포함하는 하나 이상의 큐드 레지스터 그룹들을 포함할 수도 있으며, 4개의 레지스터들에 대한 데이터는 단일 캐시

트랜잭션 및/또는 메모리 트랜잭션의 일부로서 메모리로부터 판독되거나 및/또는 그에 저장될 수 있다. 이러한 예들에서, 원자 예약 판독 동작에 의해 반환되는 3개의 파이프 상태 파라미터들은, 일부 예들에서, 퀘드레지스터 그룹 내 3개의 연속된 레지스터들에서 반환된다. 이러한 방법으로, 3개의 파이프 상태 파라미터들은 단일 캐시 트랜잭션 및/또는 메모리 트랜잭션의 일부로서 계산 유닛들 (12) 중 하나 (또는, 양쪽)에 의해 폐치될 수도 있다.

[0183] 일부 예들에서, 원자 예약 판독 지령은 OpenCL™ C 사양에 따라서 정의된 `reserve_read_pipe` 지령을 구현하기 위해 사용될 수도 있다. 추가적인 예들에서, 예약 카운트 파라미터를 제로와 동일하게 설정함으로써, 원자 예약 판독 지령은 예약을 그 파이프에 추가함이 없이 파이프의 현재의 상태를 반환할 수도 있다. 이러한 방법으로, 원자 예약 판독 지령은 또한 OpenCL™ C 사양에 따라서 정의된 `get_pip_num_packets` 지령을 구현하기 위해 사용될 수도 있다.

[0184] 도 5는 본 개시물에 따른, 예시적인 원자 커밋 판독 지령의 인터페이스 및 기능을 정의하는 예시적인 의사-코드를 예시하는 개념도이다. 의사-코드의 라인들 1-4는 이 예에서 액세스되거나 및/또는 수정될 파이프 상태 파라미터들을 어레이의 개개의 엘리먼트들에 맵핑한다. 파이프 상태 파라미터들 및 맵핑은 도 2에 대해 위에서 설명한 것과 동일하다.

[0185] 의사-코드의 라인 5는 지령을 호출할 때 사용될 지령 및 인수들 또는 파라미터들의 이름을 포함하는 원자 커밋 판독 지령에 대한 인터페이스를 규정한다. 좀더 구체적으로, 원자 커밋 판독 지령은 파이프 상태 파라미터들 (22)을 저장하는 데이터 저장 시스템 (16)에서 메모리의 영역을 가리키는 파이프 어드레스 파라미터, 및 데이터 저장 시스템 (16)에 파이프에 대해 할당되는 물리적인 패킷 엔트리들의 개수 마이너스 1에 대응하는 물리 파이프 사이즈 마이너스 1 파라미터를 이용할 수도 있다. 일부 예들에서, 파이프 어드레스 파라미터는 머신 어드레스에 기초하는 32-비트 어드레스 또는 64-비트 어드레스일 수도 있다.

[0186] 원자 커밋 판독 지령이 쓰레드에 의해 호출될 때, 원자 동작들 유닛 (14)은 데이터 저장 시스템 (16)으로부터 이 예에서 사용되는 4개의 파이프 상태 파라미터들을 판독하는 판독 동작을 수행할 수도 있다 (라인 7). 도 5에 나타낸 바와 같이, 원자 동작들 유닛 (14)은 데이터 저장 시스템 (16)으로부터 모든 4개의 파이프 상태 파라미터들을 포함하는 어레이를 판독할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32 비트일 수도 있으며, 판독 동작은 단일 128-비트 로드 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 판독 동작 및/또는 로드 동작은 로드 원자 동작 동안 액세스되거나 및/또는 수정되는 파이프 상태 파라미터들의 모두를 로드하기 위해 데이터 저장 시스템 (16)에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 판독 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정 가능할 수도 있다.

[0187] 파이프 상태 파라미터들을 판독한 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 수정 동작을 수행할 수도 있다 (라인들 8-13). 수정 동작을 수행하기 위해, 원자 동작들 유닛 (14)은 예약 카운트 파라미터를 1 만큼 감분시키는 1 감분 동작을 수행할 수도 있다 (라인 8). 예약 카운트 파라미터를 1 만큼 감분하는 것은 원자 커밋 판독 커밋 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 수정된 값이 반영하도록, 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 것에 대응할 수도 있다.

[0188] 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프에 대한 모든 예약들이 커밋되었는지 여부를 추가로 결정할 수도 있다 (라인 9). 좀더 구체적으로, 이 예에서, 예약 카운트 파라미터를 1 만큼 감분한 후, 원자 동작들 유닛 (14)은 예약 카운트 파라미터가 제로와 동일한지 여부를 결정할 수도 있다.

[0189] 파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 커밋된 엔트리들이 파이프로부터 제거되었다는 것을 수정된 값들이 반영하도록, 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시킬 수도 있다 (라인들 10-12). 좀더 구체적으로, 이 예에서, 원자 동작들 유닛 (14)은 제 1 패킷 오프셋 파라미터를 예약된 패킷 카운트 파라미터와 동일한 양 만큼 증분시키는 모듈로-가산 동작을 수행하고, 파이프의 물리적인 사이즈가 계수인 상황에서, 그 결과에 대해 모듈로 동작을 수행할 수도 있다 (라인 10). 원자 동작들 유닛 (14)은 또한 패킷 카운트 파라미터를 예약된 패킷 카운트 파라미터와 동일한 양 만큼 감분시키는 감분 동작 (라인 11), 및 예약된 패킷 카운트 파라미터를 제로로 리셋하는 리셋 동작을 수행할 수도 있다 (라인 12).

- [0190] 파이프의 물리적인 사이즈 마이너스 1 파라미터는 모듈로-가산 동작을 수행하기 위해 원자 판독 커밋 지령에 의해 사용될 수도 있다. 좀더 구체적으로, 비트 방식 AND 연산이 제 1 패킷 오프셋 파라미터 및 예약된 패킷 카운트 파라미터의 합계를 제 1 피연산자로서, 그리고 물리 파이프 사이즈 마이너스 1 을 계수로서 이용하여 수행된다. 이것은 제 1 패킷 오프셋 파라미터가 충분되는 양이 제 1 패킷 오프셋 파라미터를 파이프 데이터 베퍼 (24) 의 끝을 지나서 충분되도록 초래할 때, 제 1 패킷 오프셋 파라미터가 물리적인 패킷 엔트리들의 끝으로부터 물리적인 패킷 엔트리들의 시작까지 링 베퍼-형 방식으로 순환가능하게 할 수도 있다.
- [0191] 파이프에 대한 모든 예약들이 커밋되지 않았다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14) 은 나머지 파이프 상태 파라미터들이 동일하게 유지하도록, 이미 업데이트된 예약 카운트 파라미터 외에 다른 파이프 상태 파라미터 값들을 업데이트하지 않을 수도 있다. 나머지 파이프 상태 파라미터들이 동일하게 유지하도록 나머지 파이프 상태 파라미터 값들을 업데이트하지 않는 것은, 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록, 나머지 파이프 상태 파라미터들에 대해 수정된 값들을 발생시키는 것에 대응할 수도 있다.
- [0192] 수정된 파이프 상태 파라미터 값들을 발생시킨 후, 원자 동작들 유닛 (14) 은 이 예에서 사용되는 4개의 파이프 상태 파라미터들을 데이터 저장 시스템 (16) 에 기록하는 기록 동작 (예컨대, 저장 동작) 을 수행할 수도 있다 (라인 13). 도 5 에 나타낸 바와 같이, 원자 동작들 유닛 (14) 은 모든 4개의 파이프 상태 파라미터들을 포함하는 어레이를 데이터 저장 시스템 (16) 에 기록할 수도 있다. 일부 예들에서, 파라미터들의 각각은 32 비트일 수도 있으며, 기록 동작은 단일 128-비트 저장 동작일 수도 있다. 다시 말해서, 일부 예들에서, 단일 기록 동작 및/또는 저장 동작은 원자 동작 동안 액세스되거나 및/또는 수정되는 파이프 상태 파라미터들을 모두를 저장하기 위해 데이터 저장 시스템 (16) 에 대해 사용될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 기록 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정 가능할 수도 있다. 원자 커밋 판독 지령의 프로세싱이 이제 완료된다.
- [0193] 일부 예들에서, 원자 커밋 판독 지령의 신택스는 다음의 유형을 취할 수도 있다:
- [0194] PIPE\_CMT\_READ Src0
- [0195] 여기서, 어떤 Dest 레지스터도 규정되지 않으며, Src0 은 물리 파이프 사이즈 마이너스 1, 논리 파이프 사이즈, 및 파이프 어드레스를 포함하는 4개의 상수 레지스터들의 그룹에서 제 1 레지스터에 대한 상수 ID 에 대응한다. 파이프 어드레스는 파이프 상태 파라미터들 (22) 을 가리키는 32-비트 또는 64-비트 어드레스를 포함하는 하나 또는 2개의 레지스터들에 저장될 수도 있다. 일부 예들에서, 물리 파이프 사이즈 마이너스 1 이 2 의 거듭제곱 마이너스 1 이 되도록 물리 파이프 사이즈는 2 의 거듭제곱일 수도 있다.
- [0196] 원자 커밋 판독 지령은, 이 예에서, 어떤 값들도 반환하지 않는다. 이와 같이, 지령은 비-반환 원자의 일 예일 수도 있다. 일부 예들에서, 원자 커밋 판독 지령은 OpenCL™ C 사양에 따라서 정의된 바와 같은 commit\_read\_pipe 지령을 구현하기 위해 사용될 수도 있다.
- [0197] 일부 예들에서, 단일 판독/로드 동작 및 단일 기록/저장 동작이 파이프 상태 파라미터들에 액세스하는데 사용되도록 하기 위해서, 원자 동작 동안 액세스되거나 및/또는 수정되는 파이프 상태 파라미터들을 저장하는데 사용되는 비트수는, 단일 메모리 트랜잭션 동안 데이터 저장 시스템 (16) 으로부터 판독되거나 또는 그에 기록되는 데이터의 블록에 포함되는 비트들의 수 미만이거나 또는 동일할 수도 있다. 예를 들어, 다수의 파이프 상태 파라미터들을 저장하는데 사용되는 비트수는 메모리 워드에서의 비트수 및/또는 캐시 라인에서의 비트수보다 적거나 또는 동일할 수도 있다. 일부의 경우, 메모리 서브시스템에 저장된 파이프들의 각각에 대한 파이프 상태 파라미터들은 파이프에 대한 파이프 상태 파라미터들의 모두가 메모리 서브시스템의 단일 메모리 워드 및/또는 단일 캐시 라인에 저장되도록, 정렬될 수도 있다. 이러한 방법으로, 원자 판독-수정-기록 동작은 하나 보다 많은 메모리 판독 트랜잭션 또는 하나 보다 많은 메모리 기록 트랜잭션을 필요로 함이 없이 다수의 파이프 상태 파라미터들을 수정 가능할 수도 있다.
- [0198] 일부 예들에서, 데이터가 기록될 파이프를 나타내는 파라미터, 파이프의 사이즈를 나타내는 파라미터, 및 파이프의 사이즈 마이너스 1 을 나타내는 파라미터 중 하나 이상이 데이터 저장 시스템 (16) 에 파이프를 할당한 호스트 프로그램에 의해 쓰레드 (18A) 및/또는 쓰레드 (20A) 에 제공될 수도 있다. 이러한 파라미터들이 파이프가 할당될 때 결정될 수도 있으며 쓰레드들 (18, 20) 의 수명 전체를 통해서 일정하게 유지할 수도 있기 때문에, 이러한 파라미터들은 상수 파라미터들로서 지정될 수도 있다. 이러한 예들에서, 상수 파라미터들은 일부 예들에서, 계산 유닛 (12A) 및/또는 계산 유닛 (12B) 의 온칩 상수 메모리에 저장될 수도 있다. 호스트

프로세서 상에서 실행하는 호스트 프로그램은 일부 예들에서, 계산 유닛 (12A) 상의 쓰레드 (18A) 및/또는 계산 유닛 (12B) 상의 쓰레드 (20A)의 실행을 호출하는 함수의 인수들로서 상수 파라미터들을 제공할 수도 있다(예컨대, 상수 파라미터들은 커널 인수들로서 쓰레드 (18A) 및 쓰레드 (20A)에 제공될 수도 있다).

[0199] 도 6 은 도 1 에 나타낸 원자 동작들 유닛 (14)의 일 예가 더욱더 자세하게 예시된 도 1 의 예시적인 컴퓨팅 시스템 (10)의 블록도이다. 도 6 에 나타낸 바와 같이, 원자 동작들 유닛 (14)은 지령 큐 (26), 제어 유닛 (28), 및 원자 동작 데이터경로들 (30)을 포함한다.

[0200] 지령 큐 (26)는 계산 유닛들 (12) 상에서 실행하고 있는 쓰레드들 (18, 20)로부터 수신되는 복수의 원자 동작 지령들을 저장할 수도 있다. 원자 동작 지령들은 예를 들어, 원자 예약 기록 지령, 원자 커밋 기록 지령, 원자 예약 판독 지령, 및 원자 커밋 판독 지령을 포함할 수도 있다. 지령들은 대안적으로 본원에서 요청들로서 지정될 수도 있다.

[0201] 쓰레드들 (18, 20) 중 하나로부터 지령을 수신하는 것에 응답하여, 지령 큐 (26)는 다른 지령이 현재 원자 동작들 유닛 (14)에 의해 프로세싱되고 있더라도 그 지령을 지령 큐 (26)에 저장할 수도 있다. 이것은 지령들을 발하기 전에 원자 동작들 유닛 (14)이 자유로워질 때까지 쓰레드들 (18, 20)이 대기할 필요 없이, 그리고, 바쁜 원자 동작들 유닛 (14)로 인해 쓰레드들 (18, 20)이 지령들을 다수회 발할 필요 없이, 원자 동작들 유닛 (14)으로 하여금, 계산 유닛들 (12) 상에서 동시에 실행하고 있는 복수의 쓰레드들 및/또는 태스크들로부터 복수의 지령들을 수신가능하게 할 수도 있다.

[0202] 제어 유닛 (28)은 지령 큐 (26)에 저장된 원자 동작 지령들을 프로세싱할 수도 있다. 지령을 프로세싱하기 위해, 제어 유닛 (28)은 프로세싱된 지령에 대응하는 원자 동작을 수행할 수도 있다. 예를 들어, 원자 예약 기록 지령을 프로세싱하기 위해, 제어 유닛 (28)은 본 개시물에서 설명되는 바와 같이 원자 예약 기록 동작을 수행할 수도 있다. 원자 커밋 기록 지령을 프로세싱하기 위해, 제어 유닛 (28)은 본 개시물에서 설명되는 바와 같이 원자 커밋 기록 동작을 수행할 수도 있다. 원자 예약 판독 지령을 프로세싱하기 위해, 제어 유닛 (28)은 본 개시물에서 설명되는 바와 같이 원자 커밋 판독 지령을 프로세싱하기 위해, 제어 유닛 (28)은 본 개시물에서 설명되는 바와 같이 원자 커밋 판독 동작을 수행할 수도 있다.

[0203] 일부 예들에서, 제어 유닛 (28)은 하나의 원자 판독-수정-기록 동작이 한꺼번에 수행되도록, 그리고 각각의 원자 판독-수정-기록 동작의 실행이 다른 원자 판독-수정-기록 동작을 수행하도록 시작하기 전에 완료되도록, 지령들을 순차적으로 프로세싱할 수도 있다. 이와 같이 지령들을 프로세싱하는 것은 원자 동작들의 원자성을 보장할 수도 있다.

[0204] 일부 예들에서, 위에서 언급된 원자 동작들은 판독-수정-기록 동작들일 수도 있다. 이러한 예들에서, 원자 동작들 유닛 (14)은 병렬로 실행하는 복수의 쓰레드들로부터, 하드웨어-기반의 원자 판독-수정-기록 동작들을 수행해달라는 복수의 요청들을 수신하고, 하드웨어-기반의 원자 판독-수정-기록 동작들을 수행해달라는 복수의 요청들을 수신하는 것에 응답하여, 복수의 하드웨어-기반의 원자 판독-수정-기록 동작들을 순차적으로 수행할 수도 있다. 하드웨어-기반의 원자 판독-수정-기록 동작들의 각각은 복수의 요청들의 개개의 하나에 응답하여 수행될 수도 있다.

[0205] 원자 동작 지령들을 프로세싱할 때 수행되는 동작들은, 그 동작들이 원자 동작을 요청한 태스크 또는 쓰레드와 동시에 실행하고 있는 다른 태스크들 및/또는 쓰레드들에 의해 요청될 수도 있는 임의의 원자 동작들로부터의 인터럽트 없이, 시작부터 끝까지 수행된다는 점에서 원자일 수도 있다. 다시 말해서, 원자 동작의 실행 동안, 어떤 다른 쓰레드들 및/또는 태스크들도 현재 실행하는 원자 동작에 의해 수정되거나 및/또는 액세스되는 파이프 상태 파라미터들을 판독하거나 또는 수정불가능할 수도 있다. 이러한 방법으로, 2개 이상의 태스크들 및/또는 쓰레드들이 파이프의 파이프 상태 파라미터들을 동시에 수정하려고 시도하는 경우에 경합 조건들이 회피될 수도 있다.

[0206] 원자 동작들 유닛 (14)에 의해 수행되는 원자 동작들은 하드웨어-기반의 원자 동작들일 수도 있다. 원자 동작들은 그 원자 동작들이 원자 동작들 유닛 (14)에 통신가능하게 커플링된 쓰레드들 및/또는 태스크들에 의해 호출될 수도 있는 단일, 불가분 동작으로서 원자 동작들 유닛 (14)에 의해 구현될 수도 있다는 점에서 하드웨어-기반일 수도 있다. 단일, 불가분 동작은 원자 동작들 유닛 (14)이 원자 동작을 요청한 태스크 또는 쓰레드와 동시에 실행하고 있는 다른 태스크들 및/또는 쓰레드들에 의해 요청될 수도 있는 다른 원자 동작들을 수행하기 위해 동작을 인터럽트함이 없이, 시작부터 끝까지 동작을 수행한다는 점에서 불가분일 수도 있다.

- [0207] 도 6 의 예시적인 원자 동작들 유닛 (14) 에서, 하드웨어-기반의 원자 동작을 수행하기 위해, 원자 동작들 유닛 (14) 은 하나 이상의 원자 동작 데이터경로들 (30) 을 구현하는 회로를 포함할 수도 있다. 원자 동작 데이터경로들 (30) 의 각각은 비-프로그래밍가능 데이터경로일 수도 있다. 다시 말해서, 원자 동작 데이터경로들 (30) 의 각각은 사용자-규정된 프로그램이 데이터경로에 의해 다운로드되어 실행될 필요 없이, 개개의 데이터경로에 기인하는 기능을 수행하도록 구성될 수도 있다.
- [0208] 원자 동작 데이터경로들 (30) 은 일부 예들에서, 본 개시물에서 설명된 원자 동작들의 각각에 대한 하드웨어-기반의 데이터경로를 구현할 수도 있다. 일부 예들에서, 회로는 원자 동작들의 각각에 대한 별개의 하드웨어-기반의 데이터경로를 포함할 수도 있다. 추가적인 예들에서, 공통 하드웨어-기반의 데이터경로는 본 개시물에서 설명된 원자 동작들 중 2개 이상을 수행하도록 구성가능할 수도 있다. 일부 예들에서, 공통 하드웨어-기반의 데이터경로는 수정 동작이 수행될 원자 동작의 유형에 기초하여 상이한 수정 동작들을 수행하도록 구성 가능한 판독-수정-기록 동작 데이터경로일 수도 있다. 일부 예들에서, 원자 동작 데이터경로들 (30) 의 각각은 연관된 원자 동작 지령들의 수정 동작들을 수행하는 하나 이상의 ALU들을 포함할 수도 있다.
- [0209] 도 7 은 본 개시물에 따른, 데이터 저장 시스템 (16) 과 상호작용하는 예시적인 원자 동작 데이터경로 (32) 를 예시하는 개념도이다. 원자 동작 데이터경로 (32) 는 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하도록 구성될 수도 있다. 원자 동작 데이터경로 (32) 는 도 6 과 관련하여 예시되고 위에서 설명된 원자 동작 데이터경로들 (30) 중 하나의 일 예일 수도 있다. 원자 동작 데이터경로 (32) 는 판독 블록 (34), 수정 블록 (36), 및 기록 블록 (38) 을 포함한다.
- [0210] 판독 블록 (34) 은 데이터 저장 시스템 (16) 를 포함하는 메모리 서브시스템으로부터 파이프 상태 파라미터들 (22) 에 대응하는 값들을 판독하는 판독 동작을 수행하도록 구성될 수도 있다. 수정 블록 (36) 은 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 수정 동작을 수행하도록 구성될 수도 있다. 기록 블록 (38) 은 파이프 상태 파라미터들에 대한 수정된 값들을 데이터 저장 시스템 (16) 를 포함하는 메모리 서브시스템에 기록하는 기록 동작을 수행하도록 구성될 수도 있다.
- [0211] 판독 블록 (34), 수정 블록 (36), 및 기록 블록 (38) 의 각각은 이러한 구성요소들에 기인하는 기능들을 수행하도록 구성된 회로로 구현될 수도 있다. 일부 예들에서, 판독 블록 (34), 수정 블록 (36) 및 기록 블록 (38) 중 하나 이상은 원자 동작 데이터경로 (32) 에서의 별개의 하드웨어 블록들일 수도 있다. 추가적인 예들에서, 공통 하드웨어 블록은 판독 블록 (34), 수정 블록 (36) 및 기록 블록 (38) 중 하나 이상을 구현하도록 구성 가능할 수도 있다.
- [0212] 도 7 에 나타낸 바와 같이, 판독 블록 (34), 수정 블록 (36), 및 수정 블록 (36) 중 하나 이상은 원자 동작들 유닛 (14) 으로 발해진 원자 동작 지령과 함께 원자 동작들 유닛 (14) 에 제공된 지령 인수들 (즉, 파라미터들) 을 수신할 수도 있다. 지령 인수들은 예를 들어, 파이프 어드레스, 물리 파이프 사이즈 마이너스 1, 논리 파이프 사이즈, 및 요청된 패킷들의 개수 중 하나 이상을 포함할 수도 있다.
- [0213] 원자 동작 데이터경로 (32) 의 동작이 이하 설명될 것이다. 제어 유닛 (28) 은 원자 동작 지령을 프로세싱 하는 것에 응답하여, 원자 동작 데이터경로 (32) 를 이용하여 원자 판독-수정-기록 동작의 수행을 개시할 수도 있다. 제어 유닛 (28) 은 원자 판독-수정-기록 동작을 수행하기 위해 하나 이상의 지령 인수들을 원자 동작 데이터경로 (32) 에 제공할 수도 있다.
- [0214] 판독 블록 (34) 은 파이프 어드레스에 기초하여 판독 동작을 수행할 수도 있다. 파이프 어드레스는 파이프 상태 파라미터들 (22) 이 저장되는 데이터 저장 시스템 (16) 에서의 로케이션을 가리키는 메모리 어드레스일 수도 있다. 판독 동작을 수행하기 위해, 판독 블록 (34) 은 판독 요청을 데이터 저장 시스템 (16) 으로 발할 수도 있다. 판독 요청은 파이프 어드레스를 포함할 수도 있다. 판독 요청을 수신하는 것에 응답하여, 데이터 저장 시스템 (16) 은 파이프의 현재의 상태에 대응하는 파이프 상태 파라미터들 (PSP) 값들을 판독 블록 (34) 에 제공할 수도 있다. 판독 블록 (34) 은 PSP 값들을 수정 블록 (36) 에 제공할 수도 있다.
- [0215] 수정 블록 (36) 은 PSP 값들을 수신하고, 하나 이상의 수정 기능들에 기초하여 파이프 상태 파라미터들에 대한 수정된 PSP 값들을 발생시킬 수도 있다. 일부 예들에서, 수정 블록 (36) 은 논리 파이프 사이즈, 물리 파이프 사이즈 마이너스 1, 및 요청된 패킷들의 개수 중 하나 이상에 기초하여, 수정된 PSP 값들을 발생시킬 수도 있다. 수정 블록 (36) 은 수정된 PSP 값들을 기록 블록 (38) 에 제공할 수도 있다.
- [0216] 일부 예들에서, 수정된 PSP 값들을 발생시키기 위해 수정 블록 (36) 에 의해 사용되는 수정 기능들은 도 2 내지

도 5 에 나타낸 의사-코드에 표현된 수정 기능들에 대응할 수도 있다. 예를 들어, 원자 예약 기록 동작을 수행할 때, 수정 블록 (36) 은 도 2 에 나타낸 의사-코드의 라인들 16-20 에 나타낸 수정 기능들에 기초하여, 수정된 PSP 값들을 발생시킬 수도 있다. 다른 예로서, 원자 커밋 기록 동작을 수행할 때, 수정 블록 (36) 은 도 3 에 나타낸 의사-코드의 라인들 8-11 에 나타낸 수정 기능들에 기초하여, 수정된 PSP 값들을 발생시킬 수도 있다.

[0217] 추가적인 예로서, 원자 예약 판독 동작을 수행할 때, 수정 블록 (36) 은 도 4 에 나타낸 의사-코드의 라인들 16-20 에 나타낸 수정 기능들에 기초하여, 수정된 PSP 값들을 발생시킬 수도 있다. 추가적인 예로서, 원자 커밋 판독 동작을 수행할 때, 수정 블록 (36) 은 도 5 에 나타낸 의사-코드의 라인들 8-12 에 나타낸 수정 기능들에 기초하여, 수정된 PSP 값들을 발생시킬 수도 있다.

[0218] 기록 블록 (38) 은 파이프 어드레스에 기초하여 기록 동작을 수행할 수도 있다. 파이프 어드레스는 파이프 상태 파라미터들 (22) 이 저장되는 데이터 저장 시스템 (16) 에서의 로케이션을 가리키는 메모리 어드레스일 수도 있다. 기록 동작을 수행하기 위해, 기록 블록 (38) 은 기록 요청을 데이터 저장 시스템 (16) 으로 발할 수도 있으며, 데이터 저장 시스템 (16) 에의 저장을 위해 수정된 PSP 값들을 데이터 저장 시스템 (16) 에 제공할 수도 있다. 기록 요청은 파이프 어드레스를 포함할 수도 있다. 기록 요청을 수신하는 것에 응답하여, 데이터 저장 시스템 (16) 은 (예컨대, 파이프 상태 파라미터들 (22) 의 일부로서) 수정된 PSP 값을 데이터 저장 시스템 (16) 에 저장할 수도 있다. 원자 판독-수정-기록 동작이 이제 완료된다.

[0219] 원자 동작 데이터경로 (32) 는 판독-수정-기록 동작을 인터럽트불가능한 동작들의 세트로서 수행하도록 구성될 수도 있다. 다시 말해서, 원자 동작 데이터경로 (32) 는 현재의 판독-수정-기록 동작의 프로세싱을 위해 다른 판독-수정-기록 동작을 고려함이 없이, 현재의 판독-수정-기록 동작을 수행하도록 구성될 수도 있다. 이러한 방법으로, 하드웨어-기반의 원자 판독-수정-기록 동작은 이러한 동작들을 요청할 수도 있는 동시에 실행하는 쓰레드들 및/또는 태스크들에 대해 제공될 수도 있다.

[0220] 본 개시물의 일부 양태들에 따르면, 수정 블록 (36) 은 단일, 원자 판독-수정-기록 동작의 일부로서 상이한 파이프 상태 파라미터들에 대한 상이한 수정 동작들을 수행할 수도 있다. 예를 들어, 수정 블록 (36) 은 제 1 수정 기능에 기초하여 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 제 1 수정 동작, 및 제 2 수정 기능에 기초하여 제 2 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 제 2 수정 동작을 수행할 수도 있다. 제 2 수정 기능은 제 1 수정 기능과는 상이할 수도 있다. 이러한 방법으로, 상이한 파이프 상태 파라미터들은 단일 원자 동작의 일부로서 상이한 방법들로 업데이트될 수도 있다.

[0221] 예를 들어, 원자 예약 기록 동작 또는 원자 예약 판독 동작을 지원하기 위해, 제 1 수정 기능은 예약된 패킷 카운트 파라미터를 파이프 예약용으로 요청되는 패킷들의 개수 만큼 충분시키기 위해 사용될 수도 있으며 (도 2, 라인 17; 도 4, 라인 17), 제 2 수정 기능은 예약 카운트 파라미터를 1 의 상수 값 만큼 충분시키기 위해 사용될 수도 있다 (도 2, 라인 19; 도 4, 라인 19). 다른 예로서, 원자 커밋 기록 동작 또는 원자 커밋 판독 동작을 지원하기 위해, 제 1 수정 기능은 예약 카운트를 1 의 상수 값 만큼 감분시키기 위해 사용될 수도 있으며 (도 3, 라인 8; 도 5, 라인 8), 제 2 수정 기능은 패킷 카운트 파라미터를 예약 카운트 파라미터와 동일한 양 만큼 충분시키거나 또는 감분시키는데 사용될 수도 있으며 (도 3, 라인 10; 도 5, 라인 11), 제 3 수정 기능은 예약된 패킷 카운트 파라미터를 제로로 리셋시키기 위해 사용될 수도 있다 (도 3, 라인 11; 도 5, 라인 12).

다른 예들이 가능하며 본 개시물의 범위 이내이다.

[0222] 본 개시물의 추가적인 양태들에 따르면, 수정 블록 (36) 은 원자 동작의 실행 동안 메모리 서브시스템 (예컨대, 데이터 저장 시스템 (16)) 으로부터 획득되는 하나 이상의 추가적인 파이프 상태 파라미터들의 값에 기초하여, 하나 이상의 파이프 상태 파라미터들에 대한 수정된 값을 발생시킬 수도 있다. 예를 들어, 판독 블록 (34) 은 단일 원자 판독-수정-기록 동작 동안, 제 1 파이프 상태 파라미터에 대응하는 제 1 값 및 제 2 파이프 상태 파라미터에 대응하는 제 2 값을 판독할 수도 있다. 동일한 원자 판독-수정-기록 동작의 부분으로서, 수정 블록 (36) 은 제 1 파이프 상태 파라미터에 대응하는 제 1 값에 기초하여, 제 2 파이프 상태 파라미터에 대한 수정된 값을 발생시킬 수도 있다. 이러한 방법으로, 파이프 상태 파라미터들은 파이프의 현재의 상태에 기초하여 단일 원자 동작으로 업데이트될 수도 있다.

[0223] 예를 들어, 원자 커밋 동작을 지원하기 위해, 패킷 카운트 파라미터는 커밋 동작이 판독 커밋 동작인지 또는 기록 커밋 동작인지 여부에 따라서, 예약된 패킷 카운트 파라미터에 의해 감분되거나 또는 증분될 수도 있다 (도 3, 라인 10; 도 5, 라인 11). 다른 예로서, 원자 커밋 판독 동작을 지원하기 위해, 제 1 패킷 오프셋 파라미터는 예약된 패킷 카운트 파라미터에 기초하여 조정될 수도 있다 (도 5, 라인 10). 다른 예들이 가능하며

본 개시물의 범위 이내이다.

[0224] 본 개시물의 추가적인 양태들에 따르면, 수정 블록 (36)은 상이한 파이프 상태 파라미터 값에 기초하여 파이프 상태 파라미터 값을 조건부로 업데이트할 수도 있다. 파이프 상태 파라미터 값을 조건부로 업데이트하는 것은 상이한 파이프 상태 파라미터 값에 기초하여 파이프 상태 파라미터 값을 업데이트할지 또는 하지 않을지 여부를 결정하는 것을 수반할 수도 있다. 이러한 방법으로, 파이프 상태 파라미터들이 단일 원자 동작의 일부로서 파이프의 현재의 상태에 기초하여 업데이트되거나 또는 업데이트되지 않을 수도 있다.

[0225] 예를 들어, 원자 예약 기록 동작 또는 원자 예약 판독 동작을 지원하기 위해, 수정 블록 (36)은 패킷 카운트 파라미터 및 예약된 패킷 카운트 파라미터에 기초하여, 예약된 패킷 카운트 파라미터 및 예약 카운트 파라미터를 조건부로 업데이트할 수도 있다 (도 2, 라인들 16-20; 도 4, 라인들 16-20). 다른 예로서, 원자 커밋 기록 동작 또는 원자 커밋 판독 동작을 지원하기 위해, 수정 블록 (36)은 예약 카운트 파라미터에 기초하여, 패킷 카운트 파라미터 및 예약된 패킷 카운트 파라미터를 조건부로 업데이트할 수도 있다 (도 3, 라인들 9-12; 도 5, 라인들 9-13). 다른 예들이 가능하며 본 개시물의 범위 이내이다.

[0226] 도 8은 본 개시물의 하드웨어-기반의 원자 동작들을 구현하는데 사용될 수도 있는 예시적인 컴퓨팅 디바이스 (40)를 예시하는 블록도이다. 컴퓨팅 디바이스 (40)는 개인용 컴퓨터, 테스크탑 컴퓨터, 랩톱 컴퓨터, 컴퓨터 워크스테이션, 비디오 게임 플랫폼 또는 콘솔, (예컨대, 모바일 전화기, 셀룰러 전화기, 위성 전화기, 및/또는 모바일 폰 핸드셋과 같은) 무선 통신 디바이스, 유선 전화기, 인터넷 전화기, 휴대형 디바이스, 예컨대 휴대형 비디오 게임 디바이스 또는 개인 휴대정보 단말기 (PDA), 개인용 음악 플레이어, 비디오 플레이어, 디스플레이 디바이스, 텔레비전, 텔레비전 셋-탑 박스, 서버, 중간 네트워크 디바이스, 메인프레임 컴퓨터 또는 그래픽 데이터를 프로세싱하거나 및/또는 디스플레이하는 임의의 다른 유형의 디바이스를 포함할 수도 있다.

[0227] 도 8의 예에 예시된 바와 같이, 컴퓨팅 디바이스 (40)는 사용자 입력 인터페이스 (42), CPU (44), GPU (46), 메모리 제어기 (48), 메모리 (50), 디스플레이 인터페이스 (52), 디스플레이 (54) 및 버스 (56)를 포함한다. 사용자 입력 인터페이스 (42), CPU (44), GPU (46), 메모리 제어기 (48), 및 디스플레이 인터페이스 (52)는 버스 (20)를 이용하여 서로 통신할 수도 있다. 도 8에 나타낸 상이한 구성요소들 사이의 버스들 및 통신 인터페이스들의 특정의 구성은 단지 예시적이며, 컴퓨팅 디바이스의 다른 구성들 및/또는 동일한 또는 상이한 구성요소들을 가진 다른 그래픽스 프로세싱 시스템들이 본 개시물의 기법들을 구현하는데 이용될 수도 있다는 점에 유의해야 한다.

[0228] 사용자 입력 인터페이스 (42)는 하나 이상의 사용자 입력 디바이스들 (미도시)이 컴퓨팅 디바이스 (40)에 통신가능하게 커플링될 수 있도록 할 수도 있다. 사용자 입력 디바이스들은 사용자로 하여금 사용자 입력 인터페이스 (42)를 통해서 컴퓨팅 디바이스 (40)에 입력을 제공가능하게 할 수도 있다. 예시적인 사용자 입력 디바이스들은 키보드, 마우스, 트랙볼, 마이크로폰, 터치 패드, 터치-감지 또는 존재-민감한 디스플레이, 또는 다른 입력 디바이스를 포함한다. 터치-감지 또는 존재-민감한 디스플레이가 사용자 입력 디바이스로서 사용되는 예들에서, 사용자 입력 인터페이스 (42)의 모두 또는 일부가 디스플레이 (54)와 통합될 수도 있다.

[0229] CPU (44)는 컴퓨팅 디바이스 (40)의 동작을 제어하는 범용 또는 특수-목적 프로세서를 포함할 수도 있다. 도 8에 나타낸 바와 같이, CPU (44)는 호스트 프로그램 (58) 및 GPU 드라이버 (60)를 실행할 수도 있다. 호스트 프로그램 (58)은 GPU (46)에 대한 여러 태스크들의 실행을 오프-로드하는 컴퓨팅 애플리케이션을 위한 호스트 프로그램으로서 역할을 할 수도 있다. 일부 예들에서, 그 태스크들은 계산-집약적인 태스크들 및/또는 고도의 병렬성, 예컨대, 매트릭스 계산들, 신호 프로세싱 계산들, 통계적 알고리즘들, 분자의 모델링 애플리케이션들, 파이낸스 (finance) 애플리케이션들, 메디컬 이미징, 암호 해독 애플리케이션들, 등을 포함하는 태스크들일 수도 있다. CPU (44) 상에서 실행하는 호스트 프로그램 (58)은 예를 들어, 범용 컴퓨팅 애플리케이션, 병렬 프로세싱 애플리케이션, 스프레드 시트 애플리케이션, 워드 프로세서 애플리케이션, email 애플리케이션, 미디어 플레이어 애플리케이션, 비디오 게임 애플리케이션, 그래픽 사용자 인터페이스 애플리케이션, 운영 시스템, 그래픽스 애플리케이션, 또는 임의의 다른 유형의 애플리케이션을 포함한, 임의 종류의 소프트웨어 프로그램일 수도 있다.

[0230] GPU 드라이버 (60)는 호스트 프로그램 (58)으로부터 명령들을 수신하고 GPU (46)로 하여금 그 명령들을 서비스하게 할 수도 있다. 예를 들어, GPU 드라이버 (60)는 하나 이상의 지령들을 공식화하고, 지령들을 메모리 (50)에 배치하고, 그리고 GPU (46)에게 지령들을 실행하도록 명령할 수도 있다. 일부 예들에서, GPU 드라이버 (60)는 호스트 프로그램 (58)으로 하여금 파이프들을 생성가능하게 하고 계산 디바이스들 (예컨대, GPU (46)) 상에서 실행하는 태스크들로 하여금 데이터를 파이프에 기록하고 파이프로부터 데이터를 판독함으로

써 서로 통신가능하게 하는 API 를 지원하는 런타임 프로그램을 포함할 수도 있다.

[0231] 도 8 에 나타내지는 않지만, CPU (44) 는 또한 컴파일러를 실행할 수도 있다. 컴파일러는 하나 이상의 상이한 유형들의 태스크들에 대한 소스 코드 (예컨대, 커널들 및/또는 계산 세이더 프로그램들) 를 수신하고, 태스크들에 대한 컴파일된 코드를 발생시킬 수도 있다. GPU 드라이버 (60) 는 GPU (46) 에 의한 실행을 위해 태스크들 중 하나 이상에 대한 컴파일된 코드를 GPU (46) 상으로 (예컨대, GPU (46) 의 계산 유닛에 포함되는 명령 스토어에) 로드할 수도 있다.

[0232] 일부 예들에서, 호스트 프로그램 (58) 에 대한 소스 코드는 예를 들어, OpenCL™ API 와 같은, GPGPU API, 병렬 프로그래밍 API, 및/또는 이종 컴퓨팅 플랫폼 API 을 따를 수도 있다. 추가적인 예들에서, 컴파일되는 태스크들에 대한 소스 코드는 예를 들어, OpenCL™ C API 와 같은, GPGPU API, 병렬 프로그래밍 API, 및/또는 이종 컴퓨팅 플랫폼 API 를 따를 수도 있다.

[0233] GPU (46) 는 CPU (44) 에 의해 GPU (46) 로 발해지는 지령들을 실행하도록 구성될 수도 있다. GPU (46) 에 의해 실행되는 지령들은 범용 컴퓨팅 지령들, 태스크 실행 지령들 (예컨대, 커널 실행 지령들), 메모리 전송 지령들, 등을 포함할 수도 있다.

[0234] GPU (46) 는 CPU (44) 상에서 실행하는 애플리케이션들에 대한 범용 컴퓨팅을 수행하도록 구성될 수도 있다. 예를 들어, CPU (44) 상에서 실행하고 있는 호스트 프로그램 (58) 이 GPU (46) 에 대한 계산 태스크를 오프-로드 (off-load) 하기로 결정할 때, CPU (44) 는 범용 컴퓨팅 데이터를 GPU (46) 에 제공하고, 하나 이상의 범용 컴퓨팅 지령들을 GPU (46) 로 발할 수도 있다. 범용 컴퓨팅 지령들은 예컨대, 커널 실행 지령들, 메모리 전송 지령들, 등을 포함할 수도 있다. 일부 예들에서, CPU (44) 는 지령들 및 데이터를 GPU (46) 에 의해 액세스될 수도 있는 메모리 (50) 에 기록함으로써, 지령들 및 범용 컴퓨팅 데이터를 GPU (46) 에 제공할 수도 있다.

[0235] 도 8 에 나타낸 바와 같이, GPU (46) 는 원자 동작들 유닛 (14) 및 계산 유닛들 (12) 을 포함한다. 계산 유닛들 (12) 은 하나 이상의 태스크들을 실행할 수도 있다. 일부 예들에서, 계산 유닛들 (12) 은 다수의 태스크들을 동시에 실행할 수도 있다. 추가적인 예들에서, 계산 유닛들 (12) 중 하나 이상은 태스크의 다수의 인스턴스들을 동시에 실행할 수도 있다. 계산 유닛들 (12) 중 하나에 의해 실행되는 태스크의 인스턴스는 쓰레드 및/또는 작업-아이템으로서 지칭될 수도 있다. 태스크는 대안적으로 커널로서 지칭될 수도 있다.

원자 동작들 유닛 (14) 은 계산 유닛들 (12) 로부터 하나 이상의 상이한 유형들의 원자 동작 지령들을 수신하고, 본 개시물에서 설명되는 바와 같이 상이한 유형들의 원자 동작 지령들의 각각에 대응하는 하드웨어-기반의 원자 동작들을 수행하도록 구성될 수도 있다.

[0236] 일부 예들에서, 범용 컴퓨팅 지령들을 수행하는 것에 더하여, GPU (46) 는 하나 이상의 그래픽스 프리미티브들을 디스플레이 (54) 로 렌더링하기 위해 그래픽스 동작들을 수행하도록 구성될 수도 있다. 이러한 예들에서, CPU (44) 상에서 실행하는 소프트웨어 애플리케이션들 중 하나가 그래픽스 프로세싱을 필요로 할 때, CPU (44) 는 그래픽스 데이터를 GPU (46) 에 제공하고 하나 이상의 그래픽스 지령들을 GPU (46) 로 발할 수도 있다. 그래픽스 지령들은 예컨대, 그리기 호출 지령들, GPU 상태 프로그래밍 지령들, 메모리 전송 지령들, 블리팅 (blitting) 지령들, 등을 포함할 수도 있다. 그래픽스 데이터는 정점 버퍼들, 텍스쳐 데이터, 표면 데이터, 등을 포함할 수도 있다. 일부 예들에서, CPU (44) 는 지령들 및 그래픽스 데이터를 GPU (46) 에 의해 액세스될 수도 있는 메모리 (50) 에 기록함으로써, 지령들 및 그래픽스 데이터를 GPU (46) 에 제공할 수도 있다.

[0237] GPU (46) 는 일부 경우, CPU (44) 보다 좀더 효율적인 벡터 동작들의 프로세싱을 제공하는 고속-병렬 구조로 형성될 수도 있다. 예를 들어, GPU (46) 는 다수의 정점들, 제어 지점들, 광선들 및/또는 다른 데이터에 대해 병렬 방식으로 동작하도록 구성된 복수의 프로세싱 엘리먼트들을 포함할 수도 있다. GPU (46) 의 높은 병렬 성질은 일부 경우, GPU (46) 로 하여금, 고도의 병렬성을 포함하는 태스크들을 CPU (44) 보다 더 빨리 프로세싱 가능하게 할 수도 있다. 게다가, GPU (46) 의 고속 병렬 성질은 일부 경우, GPU (46) 로 하여금, CPU (44) 를 이용하여 이미지들을 렌더링하는 것보다 더 빨리 그래픽스 이미지들 (예컨대, GUI들 및 2차원 (2D) 및/또는 3차원 (3D) 그래픽스 장면들) 을 디스플레이 (54) 상으로 렌더링 가능하게 할 수도 있다.

[0238] GPU (46) 는 일부 경우, 컴퓨팅 디바이스 (40) 의 마더보더에 통합될 수도 있다. 다른 경우, GPU (46) 는 컴퓨팅 디바이스 (40) 의 마더보더에서의 포트에 설치된 그래픽스 카드 상에 존재할 수도 있거나 또는 아니면, 컴퓨팅 디바이스 (40) 와 상호운용하도록 구성된 주변장치 디바이스 내에 통합될 수도 있다. 추가적인 경우

들에서, GPU (46)는 시스템 온 칩 (SoC)을 형성하는 CPU (44)로서 동일한 마이크로칩 상에 로케이트될 수도 있다. GPU (46)는 하나 이상의 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 디지털 신호 프로세서들 (DSP들), 또는 다른 등가의 통합 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들을 포함할 수도 있다.

[0239] 일부 예들에서, GPU (46)는 메모리 (50)의 모두 또는 부분에 대해 캐싱 서비스들을 제공할 수도 있는 GPU 캐시를 포함할 수도 있다. 이러한 예들에서, GPU (46)는 오프-칩 메모리 대신, 로컬 스토리지를 이용하여, 데이터를 로컬로 프로세싱하기 위해 캐시를 이용할 수도 있다. 이것은 GPU (46)로 하여금, GPU (46)가 각각의 판독 및 기록 지령 동안 막대한 버스 트래픽을 겪을 수도 있는 버스 (56)를 통해서 메모리 (50)에 액세스할 필요성을 감소시킴으로써, 좀 더 효율적인 방법으로 동작가능하게 한다. 일부의 예들에서, 그러나, GPU (46)는 별개의 캐시를 포함하지 않고, 대신 버스 (56)를 통해서 메모리 (50)를 이용할 수도 있다. GPU 캐시는 예컨대, 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 등과 같은, 하나 이상의 휘발성 또는 비-휘발성 메모리 또는 저장 디바이스들을 포함할 수도 있다.

[0240] 메모리 제어기 (48)는 메모리 (50)로 그리고 그로부터의 데이터의 전송을 촉진한다. 예를 들어, 메모리 제어기 (48)는 메모리 판독 및 기록 지령들을 수신하고, 컴퓨팅 디바이스 (40)에서의 구성요소들에 메모리 서비스들을 제공하기 위해 메모리 (50)에 대해 이러한 지령들을 서비스할 수도 있다. 메모리 제어기 (48)은 메모리 (50)에 통신가능하게 커플링된다. 메모리 제어기 (48)가 CPU (44) 및 메모리 (50) 양쪽과는 별개인 프로세싱 모듈인 것으로 도 8의 예시적인 컴퓨팅 디바이스 (40)에서 예시되지만, 다른 예들에서, 메모리 제어기 (48)의 기능 중 일부 또는 모두는 CPU (44) 및 메모리 (50) 중 하나 또는 양자 상에서 구현될 수도 있다.

[0241] 메모리 (50)는 CPU (44)에 의한 실행을 위해서 액세스가능한 프로그램 모듈들 및/또는 명령들 및/또는 CPU (44) 상에서 실행하는 프로그램들에 의해 사용하기 위한 데이터를 저장할 수도 있다. 예를 들어, 메모리 (50)는 CPU (44) 상에서 실행하는 애플리케이션들과 연관된 프로그램 코드 및 그래픽스 데이터를 저장할 수도 있다. 게다가, 메모리 (50)는 컴퓨팅 디바이스 (40)의 다른 구성요소들에 의한 사용을 위한 및/또는 그에 의해 생성되는 정보를 저장할 수도 있다. 예를 들어, 메모리 (50)는 GPU (46)에 대한 디바이스 메모리로서 역할을 할 수도 있으며, GPU (46) 상에서 그에 의해 연산될 데이터 뿐만 아니라 GPU (46)에 의해 수행되는 연산들로부터 초래되는 데이터를 저장할 수도 있다. 예를 들어, 메모리 (50)는 버퍼 오브젝트들, 파이프 데이터, 또는 기타 등등의 임의의 조합을 저장할 수도 있다. 게다가, 메모리 (50)는 GPU (46)에 의해 프로세싱하는 지령 스트림들을 저장할 수도 있다 (예컨대, 지령 큐들). 메모리 (50)는 예를 들어, 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 판독 전용 메모리 (ROM), 소거가능한 프로그래밍가능 ROM (EPROM), 전기적 소거가능 프로그래밍가능 ROM (EEPROM), 플래시 메모리, 자기 데이터 매체 또는 광학 저장 매체와 같은, 하나 이상의 휘발성 또는 비-휘발성 메모리 또는 저장 디바이스들을 포함할 수도 있다. 일부 예들에서, 메모리 (50)는 도 1에 나타낸 데이터 저장 시스템 (16)의 모두 또는 일부에 대응할 수도 있다.

[0242] CPU (44) 및/또는 GPU (46)는 래스터화된 이미지 데이터를 메모리 (50) 내에 할당된 프레임 버퍼에 저장할 수도 있다. 디스플레이 인터페이스 (52)는 프레임 버퍼로부터 데이터를 취출하고, 래스터화된 이미지 데이터에 의해 표현되는 이미지를 디스플레이하도록 디스플레이 (54)를 구성할 수도 있다. 일부 예들에서, 디스플레이 인터페이스 (52)는 프레임 버퍼로부터 취출된 디지털 값들을 디스플레이 (54)에 의해 소비가능한 아날로그 신호로 변환하도록 구성되는 디지털-대-아날로그 변환기 (DAC)를 포함할 수도 있다. 다른 예들에서, 디스플레이 인터페이스 (52)는 프로세싱을 위해 디지털 값들을 디스플레이 (54)로 직접 전달할 수도 있다.

[0243] 디스플레이 (54)는 모니터, 텔레비전, 투사 디바이스, 액정 디스플레이 (LCD), 플라즈마 디스플레이 패널, 발광 다이오드 (LED) 어레이, 음극선관 (CRT) 디스플레이, 전자 종이, 표면-전도 전자방출 디스플레이 (SED), 레이저 텔레비전 디스플레이, 나노크리스탈 디스플레이 또는 또다른 유형의 디스플레이 유닛을 포함할 수도 있다. 디스플레이 (54)는 컴퓨팅 디바이스 (40) 내에 통합될 수도 있다. 예를 들어, 디스플레이 (54)는 모바일 전화기 핸드셋 또는 태블릿 컴퓨터의 스크린일 수도 있다. 대안적으로, 디스플레이 (54)는 유선 또는 무선 통신 링크를 통해서 컴퓨터 디바이스 (2)에 커플링된 독립형 디바이스일 수도 있다. 예를 들어, 디스플레이 (54)는 케이블 또는 무선 링크를 통해서 개인용 컴퓨터에 접속된 컴퓨터 모니터 또는 평판 패널 디스플레이일 수도 있다.

[0244] 버스 (56)는 제 1, 제 2 및 제 3세대 버스 구조들 및 프로토콜들, 공유된 버스 구조들 및 프로토콜들, 점-대-점 버스 구조들 및 프로토콜들, 단방향의 버스 구조들 및 프로토콜들, 및 양방향의 버스 구조들 및 프로토콜들

을 포함한, 버스 구조들과 버스 프로토콜들의 임의의 조합을 이용하여 구현될 수도 있다. 버스 (56) 를 구현하는데 사용될 수도 있는 상이한 버스 구조들 및 프로토콜들의 예들은 예컨대, HyperTransport 버스, InfiniBand 버스, AGP (Advanced Graphics Port) 버스, PCI (Peripheral Component Interconnect) 버스, PCI Express 버스, AMBA (Advanced Microcontroller Bus Architecture) AHB (Advanced High-performance Bus), AMBA APB (Advanced Peripheral Bus), 및 AMBA AXI (Advanced eXtensible Interface) 버스를 포함한다. 다른 유형들의 버스 구조들 및 프로토콜들이 또한 사용될 수도 있다.

[0245] 본 개시물에 따르면, GPU (46) (예컨대, 원자 동작들 유닛 (14)) 는 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행할 수도 있다. 파이프는 테스크 간 통신을 지원하는 선입선출 (FIFO)-편성된 데이터 버퍼에 대응할 수도 있다. 2개 이상의 파이프 상태 파라미터 값들을 원자 방식으로 수정하기 위해 하드웨어-기반의 원자 판독-수정-기록 동작들을 수행하는 것은, 동시에 실행하는 테스크들 및/또는 쓰레드들이 락들 및/또는 중요 코드의 섹션들의 사용을 필요로 함이 없이, 파이프의 상태를 수정가능하게 할 수도 있다.

[0246] 일부 예들에서, 하드웨어-기반의 원자 판독-수정-기록 동작을 수행하기 위해, GPU (46) (예컨대, 원자 동작들 유닛 (14)) 는 인터럽트불가능한 동작들의 세트를 수행할 수도 있다. 인터럽트불가능한 동작들의 세트는 메모리 서브시스템 (예컨대, GPU (46) 에 포함되는 캐시 및/또는 메모리 (50)) 으로부터 파이프 상태 파라미터들에 대응하는 값들을 판독하는 동작, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작, 및 파이프 상태 파라미터들에 대한 수정된 값들을 메모리 서브시스템에 기록하는 동작을 포함할 수도 있다.

[0247] 컴퓨팅 디바이스 (40) 의 예시적인 동작이 이하 설명될 것이다. 호스트 프로그램 (58) 은 파이프 오브젝트를 생성하라는 명령을 GPU 드라이버 (60) 로 발할 수도 있다. 명령은 생성될 파이프의 논리적 사이즈를 규정할 수도 있다. 파이프의 논리적 사이즈는 파이프에 포함할 논리적 패킷 엔트리들의 개수를 지정할 수도 있다.

[0248] 파이프 오브젝트를 생성하라는 명령을 수신하는 것에 응답하여, GPU 드라이버 (60) 는 파이프 오브젝트에 대해 메모리 (50) 에서의 메모리 공간을 할당할 수도 있다. 할당된 메모리 공간은 도 1 에 예시된 바와 같은, 파이프 상태 파라미터들 (22) 및 파이프 데이터 버퍼 (24) 용으로 할당된 메모리 공간을 포함할 수도 있다. 일부 예들에서, 파이프에 할당된 물리적인 패킷 엔트리들의 개수는 파이프에 대해 호스트 프로그램 (58) 에 의해 규정된 논리적 패킷 엔트리들의 개수와 반드시 동일하지 않을 수도 있다. 예를 들어, GPU 드라이버 (60) 는 호스트 프로그램 (58) 에 의해 요청된 논리적 패킷 엔트리들의 개수보다 크거나 또는 동일한 물리적인 패킷 엔트리들의 개수가 2 의 1승이 되도록, 파이프에 대해 메모리를 할당할 수도 있다. 파이프를 할당한 후, GPU 드라이버 (60) 는 방금 생성된 파이프를 식별하는 핸들을 반환할 수도 있다.

[0249] 호스트 프로그램 (58) 은 GPU (46) 에 의해 실행될 테스크에 기초하여 테스크 오브젝트 (예컨대, 커널 오브젝트) 를 생성할 수도 있다. 호스트 프로그램 (58) 은 테스크 오브젝트에 대한 인수들 중 하나를 이전에 생성된 파이프에 대응하게 설정할 (예컨대, 그 인수를 파이프에 대한 핸들과 동일하게 설정할) 수도 있다. 호스트 프로그램 (58) 은 테스크 오브젝트가 GPU (46) 에 의해 실행되도록 하는 명령을 GPU 드라이버 (60) 로 발할 수도 있다.

[0250] 테스크 오브젝트가 실행되도록 하는 명령을 수신하는 것에 응답하여, GPU 드라이버 (60) 는 GPU (46) 로 하여금, 계산 유닛들 (12) 중 하나 이상 상에서 테스크 오브젝트에 대응하는 테스크를 실행하도록 할 수도 있다. 예를 들어, GPU 드라이버 (60) 는 GPU (46) 에 대한 지령 큐에, GPU (46) 로 하여금 테스크를 실행하도록 하는 지령을 배치할 수도 있다.

[0251] 일부 예들에서, GPU 드라이버 (60) 는 GPU (46) 에 파이프의 특성을 규정하는 하나 이상의 파라미터들 및/또는 인수들을 제공할 수도 있다. 예를 들어, GPU 드라이버 (60) 는 GPU (46) 에, 파이프에 대한 파이프 어드레스, 파이프에 대한 물리 파이프 사이즈 마이너스 1, 및/또는 파이프에 대한 논리 파이프 사이즈 중 하나 이상을 제공할 수도 있다. 일부 예들에서, GPU 드라이버 (60) 및/또는 GPU (46) 는 테스크를 실행하고 있는 계산 유닛들 (12) 의 상수 메모리 및/또는 상수 레지스터들이 파이프의 특성을 규정하는 파라미터들로 팝풀레이트되도록 할 수도 있다.

[0252] GPU (46) 로 하여금 테스크를 실행하도록 명령하는 지령을 수신하는 것에 응답하여, GPU (46) 는 그 테스크를 계산 유닛들 (12) 중 하나 이상에 할당한다. 테스크를 실행하도록 할당된 계산 유닛들 (12) 중 하나 이상이 테스크를 실행한다. 일부 예들에서, 테스크를 실행하는 것은 복수의 쓰레드들 (즉, 테스크의 인스턴스들)

을 병렬로 실행하는 것을 수반할 수도 있다.

[0253] 계산 유닛들 (12) 상에서 동시에 실행하고 있는 태스크 및/또는 쓰레드들은 하나 이상의 원자 동작 지령들을 원자 동작들 유닛 (14)으로 발한다. 원자 동작 지령들을 수신하는 것에 응답하여, 원자 동작들 유닛 (14)은 본 개시물에서 설명되는 바와 같이 하나 이상의 하드웨어-기반의 원자 동작들을 수행한다. 태스크 및/또는 쓰레드들의 수행이 완료될 때까지 계속된다.

[0254] 도 9 는 본 개시물에 따른, GPU (46) 또는 다른 계산 디바이스에서 사용될 수도 있는 예시적인 계산 유닛 (12A)을 예시하는 블록도이다. 계산 유닛 (12A)은 계산 유닛 (12A)에 액세스 가능한 태스크들에 대한 컴파일된 프로그램 코드에 기초하여 하나 이상의 상이한 유형들의 태스크들을 실행하도록 구성될 수도 있다. 계산 유닛 (12A)은 제어 유닛 (72), 명령 스토어 (74), 프로세싱 엘리먼트들 (76A-76H) (일괄하여 "프로세싱 엘리먼트들 (76)", 레지스터들 (78), 상수 메모리 (80) 및 로컬 메모리 (82)를 포함한다.

[0255] 명령 스토어 (74)는 계산 유닛 (12A)에 의해 실행될 하나 이상의 태스크들 (예컨대, 커널들)에 대한 프로그램 명령들의 모두 또는 일부를 저장하도록 구성된다. 명령 스토어 (74)는 예컨대, 휘발성 메모리, 비-휘발성 메모리, 캐시, 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 등을 포함한, 임의 종류의 저장 유닛일 수도 있다. 명령 스토어 (74)가 캐시일 때, 명령 스토어 (74)는 계산 유닛 (12A) 외부에 있는 메모리에 저장되어 있는 프로그램 명령들을 캐시할 수도 있다. 명령 스토어 (74)가 계산 유닛 (12A)의 내부에 있는 것으로 예시되지만, 다른 예들에서, 명령 스토어 (74)는 계산 유닛 (12A)의 외부에 있을 수도 있다.

[0256] 프로세싱 엘리먼트들 (76)은 태스크 프로그램의 쓰레드들을 실행하도록 구성된다. 프로세싱 엘리먼트들 (76)의 각각은 상이한 쓰레드를 실행할 수도 있다. 쓰레드는 쓰레드에 특유한 데이터 아이템에 대해 실행되는 태스크의 인스턴스를 지칭할 수도 있다. 따라서, 프로세싱 엘리먼트들 (76)의 각각이 잠재적으로 상이한 데이터 아이템들에 대해 태스크의 인스턴스를 실행한다고 말할 수도 있다. 공통 시점에서 프로세싱 엘리먼트들 (76A-76H) 상에서 병렬로 실행하고 있는 쓰레드들의 컬렉션은 쓰레드들의 파 (wave)로서 지칭될 수도 있다.

[0257] 도 9 의 예시적인 계산 유닛 (12A)에서, 프로세싱 엘리먼트들 (76)은 단일-명령, 다수의-데이터 (SIMD) 프로세싱 엘리먼트들일 수도 있다. SIMD 프로세싱 엘리먼트들은 활성화될 때, 상이한 데이터에 대해 동일한 명령을 동시에 실행하도록 구성되는 프로세싱 엘리먼트들을 지칭한다. 이것은 프로세싱 엘리먼트들 (76)로 하여금, 상이한 데이터 아이템들에 대해 복수의 태스크의 쓰레드들을 병렬로 실행 가능하도록 할 수도 있다. 일부의 경우, 프로세싱 엘리먼트들 (76)의 각각은 명령 스토어 (74)에 포함된 명령을 가리키는 공통 프로그램 카운터에 기초하여 태스크의 명령들을 실행할 수도 있다.

[0258] 프로세싱 엘리먼트들 (76) 중 하나 이상이 비활성화되면, 이러한 프로세싱 엘리먼트들 (76)은 주어진 명령 사이를 동안 프로그램 명령을 실행하지 않는다. 일부의 경우, 제어 유닛 (72)은 분기 조건이 일부 쓰레드들에 대해 만족되지만 다른 쓰레드들에 대해서는 만족되지 않는 조건부 분기 명령들을 구현하기 위해 프로세싱 엘리먼트들 (76) 중 하나 이상을 비활성화할 수도 있다.

[0259] 일부 예들에서, 프로세싱 엘리먼트들 (76)의 각각은 산술 로직 유닛 (ALU)을 포함하거나 및/또는 그에 대응할 수도 있다. 추가적인 예들에서, 프로세싱 엘리먼트들 (76)의 각각은 ALU 기능을 구현할 수도 있다. ALU 기능은 가산, 감산, 곱셈, 등을 포함할 수도 있다. 추가적인 예들에서, 프로세싱 엘리먼트들 (76)의 각각은 스칼라 ALU 또는 벡터 ALU 일 수도 있다. 스칼라 ALU는 스칼라 데이터 아이템들에 대해 동작할 수도 있으며, 벡터 ALU는 벡터 데이터 아이템들에 대해 동작할 수도 있다. 스칼라 데이터 아이템은 스칼라에 대한 단일 성분에 대응하는 단일 값은 포함할 수도 있다. 벡터 데이터 아이템은 벡터의 다수의 성분들에 대응하는 다수의 값들을 포함할 수도 있다.

[0260] 프로세싱 엘리먼트들 (76)의 각각은 명령 스토어 (74)로부터 명령들을 판독하거나 및/또는 레지스터들 (78), 상수 메모리 (80), 로컬 메모리 (82) 및 외부 메모리 (82) 중 하나 이상으로부터 데이터 아이템들을 판독할 수도 있다. 프로세싱 엘리먼트들 (76)의 각각은 출력 데이터를 레지스터들 (78), 로컬 메모리 (82) 및 외부 메모리 중 하나 이상에 기록할 수도 있다.

[0261] 레지스터들 (78)은 여러 프로세싱 엘리먼트들 (76)에 동적으로 할당될 수도 있다. 일부의 경우, 일부 또는 모두 레지스터들 (78)은 계산 유닛 (12A) 상에서 실행하는 여러 쓰레드들에 대한 입력 레지스터들 및/또는 출력 레지스터들로서 기능할 수도 있다. 입력 레지스터는 태스크 프로그램에 대한 입력 데이터 아이템들을

저장하는 레지스터를 지칭할 수도 있으며, 출력 레지스터는 태스크 프로그램에 대한 출력 데이터 아이템들을 저장하는 레지스터를 지칭할 수도 있다.

[0262] 상수 메모리 (80) 는 계산 유닛 (12A) 상에서 실행하는 태스크들에 의해 실행되는 상수 값들을 저장할 수도 있다. 상수 메모리 (80) 는 예컨대, 휘발성 메모리, 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 전기적으로 소거가능한 프로그래밍가능 판독 전용 메모리 (EEPROM), 등을 포함한, 임의 종류의 메모리일 수도 있다.

[0263] 로컬 메모리 (82) 는 계산 유닛 (12A) 상에서 실행하는 태스크들에 의한 사용을 위해 메모리 스토리지를 제공할 수도 있다. 로컬 메모리 (82) 는 예컨대, 휘발성 메모리, 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 등을 포함한, 임의 종류의 메모리일 수도 있다. 일부 예들에서, 로컬 메모리 (82) 에 대한 어드레스 공간은 계산 유닛 (12A) 에 포함되는 프로세싱 엘리먼트들 (76) 에 로컬일 수도 있다. 다시 말해서, GPU (46) 의 다른 계산 유닛들 및/또는 다른 부분들은 로컬 메모리 (82) 에 직접 액세스불가능할 수도 있다. 이와 유사하게, 호스트 디바이스는 로컬 메모리 (82) 에 직접 액세스불가능할 수도 있다. 일부 예들에서, 로컬 메모리 (82) 는 계산 유닛 (12A) 및/또는 GPU (46) 와 동일한 칩 상에서 구현될 수도 있다.

[0264] 계산 유닛 (12A) 의 예시적인 동작이 이하 설명될 것이다. GPU (46) 는 실행을 위해 태스크를 계산 유닛 (12A) 에 할당할 수도 있다. GPU (46) 는 태스크에 대한 컴파일 코드를 명령 스토어 (74) 에 저장할 수도 있다. GPU (46) 는 태스크에 대한 상수 파라미터들을 상수 메모리 (80) 에, 그리고 태스크에 대한 입력 파라미터들을 레지스터들 (78) 에 저장할 수도 있다. 프로세싱 엘리먼트들 (76) 은 태스크의 복수의 인스턴스들 (즉, 쓰레드들) 을 병렬로 실행한다. 프로세싱 엘리먼트들 (76) 상에서 실행하는 쓰레드들 중 하나 이상이 본 개시물에서 설명되는 원자 동작 지령들 중 하나 이상을 발한다.

[0265] 원자 동작들 지령을 발하기 위해, 프로세싱 엘리먼트 (76A) 는 원자 동작 지령에 대한 입력 파라미터들을 포함하는 레지스터들 (78) 중 하나 이상을 규정할 수도 있다. 프로세싱 엘리먼트 (76A) 는 원자 동작 지령을 원자 동작들 유닛 (14) 으로 발할 수도 있다.

[0266] 일부 예들에서, 원자 커밋 기록 및 원자 커밋 판독 지령들에 대해, 어떤 반환 값들도 원자 동작들 유닛 (14) 에 의해 제공되지 않는다. 원자 예약 기록 및 원자 예약 판독 지령들에 대해, 반환 값들이 제공될 수도 있다. 예를 들어, 원자 동작들 유닛 (14) 은 제 1 패킷 오프셋 파라미터, 패킷 카운트 파라미터, 및 예약된 패킷 카운트 파라미터를 반환할 수도 있다. 일부 예들에서, 레지스터들 (78) 은 각각의 큐드 레지스터 그룹이 4 개의 레지스터들을 포함하는 하나 이상의 큐드 레지스터 그룹들을 포함할 수도 있으며, 4개의 레지스터들에 대한 데이터는 단일 캐시 및/또는 메모리 트랜잭션의 일부로서 메모리로부터 판독되거나 및/또는 그에 저장될 수 있다. 이러한 예들에서, 원자 예약 기록 동작 및/또는 원자 예약 판독 동작에 의해 반환되는 3개의 파이프 상태 파라미터들은 일부 예들에서, 레지스터들 (78) 의 큐드 레지스터 그룹 내 3개의 연속된 레지스터들에서 반환될 수도 있다. 이러한 방법으로, 3개의 파이프 상태 파라미터들이 단일 캐시 및/또는 메모리 트랜잭션의 일부로서 계산 유닛 (12A) 에 의해 폐치될 수도 있다.

[0267] 도 10 은 예시적인 GPU (46) 및 메모리 구성이 더욱 자세하게 예시된 도 1 의 컴퓨팅 시스템 (10) 의 일부를 예시하는 블록도이다. 도 10 에 예시된 구성은 도 1 의 컴퓨팅 시스템 (10) 및/또는 도 8 의 컴퓨팅 디바이스 (40) 를 구현하기 위해 사용될 수도 있다.

[0268] 도 10 에 나타낸 바와 같이, GPU (46) 는 캐시 유닛 (84) 을 포함한다. 캐시 유닛 (84) 은 GPU (46) 에서의 계산 유닛들 (12) 에 대한 캐싱 서비스들을 제공할 수도 있다. 캐시 유닛 (84) 은 지령 큐 (26), 캐시 제어 유닛 (86), 원자 동작 데이터경로들 (30), 및 캐시 저장 유닛 (88) 을 포함한다. 캐시 저장 유닛 (88) 은 캐시된 파이프 상태 파라미터들 (90) 및 캐시된 파이프 데이터 버퍼 (92) 중 하나 (또는, 양쪽) 를 포함할 수도 있다. 캐시된 파이프 상태 파라미터들 (90) 은 파이프 상태 파라미터들 (22) 의 캐시된 복사본일 수도 있으며, 캐시된 파이프 데이터 버퍼 (92) 는 파이프 데이터 버퍼 (24) 의 캐시된 복사본일 수도 있다. 일부 예들에서, 캐시된 복사본들이 캐시되기 때문에, 이러한 복사본들은 그들의 개개의 소스 데이터의 불완전한 및/또는 부분 복사본들일 수도 있다.

[0269] 캐시 유닛 (84) 및 메모리 (50) 는 메모리 서브시스템을 형성할 수도 있으며, 캐시 저장 유닛 (88) 및 메모리 (50) 는 메모리 서브시스템에서의 데이터 저장 시스템을 형성할 수도 있다. 예를 들어, 데이터 저장 시스템은 도 1 에서의 데이터 저장 시스템 (16) 에 대응할 수도 있다.

[0270] 지령 큐 (26) 는 계산 유닛들 (12) 로부터 수신된 메모리 액세스 지령들 및 계산 유닛들 (12) 로부터 수신된 원

자 동작 지령들을 저장하도록 구성될 수도 있다. 캐시 제어 유닛 (86)은 메모리 액세스 지령들 및 원자 동작 지령들을 프로세싱할 수도 있다. 일부 예들에서, 캐시 제어 유닛 (86)은 하나의 지령이 한번에 프로세싱되도록 지령들을 순차적으로 프로세싱할 수도 있다.

[0271] 메모리 액세스 지령들에 대해, 캐시 제어 유닛 (86)은 대응하는 판독 요청들 및 기록 요청들을 캐시 저장 유닛 (88) 및 메모리 (50) 중 하나 (또는, 양쪽)로 발할 수도 있다. 원자 동작 지령들에 대해, 캐시 제어 유닛 (86)은 원자 동작 데이터경로들 (30) 중 하나 이상으로 하여금 그 지령에 대응하는 하드웨어-기반의 원자 동작을 수행하도록 할 수도 있다. 원자 동작 데이터경로들 (30)에 의해 구현되는 하드웨어-기반의 원자 동작들은 원자 동작들을 수행하기 위해 판독 요청들 및 기록 요청들을 캐시 저장 유닛 (88) 및 메모리 (50) 중 하나 (또는, 양쪽)로 발할 수도 있다.

[0272] 도 10에 나타낸 바와 같이, 원자 동작들 유닛 (14) (도 6)에 의해 수행되는 하드웨어-기반의 원자 동작들은 일부 예들에서, GPU (46)의 캐시 유닛 (84)에 통합될 수도 있다. 원자 동작들 유닛 (14)에 의해 수행되는 하드웨어-기반의 원자 동작들을 GPU (46)의 캐시 유닛 (84)에 통합하는 것은, 유사한 기능들을 수행하는 하드웨어의 일부가 원자 동작들 유닛 (14)과 캐시 유닛 (84) 사이에 공유될 수 있도록 할 수도 있다. 이러한 방법으로, 좀더 컴팩트한 회로가 본 개시물의 하드웨어-기반의 원자 동작들을 구현하는데 사용될 수도 있다.

[0273] 도 11은 본 개시물에 따른, 하드웨어-기반의 원자 동작들을 수행하는 예시적인 기법을 예시하는 흐름도이다. 도 11에 예시된 예시적인 기법은 컴퓨팅 시스템 (10)의 원자 동작들 유닛 (14)에 대해 설명된다. 일부 예들에서, 원자 동작들 유닛 (14)은 예컨대, GPU (예컨대, GPU (46)), CPU, DSP, 등과 같은, 프로세서 및/또는 프로세싱 유닛에 포함될 수도 있다.

[0274] 원자 동작들 유닛 (14)은 계산 유닛 (12A)로부터 원자 동작 지령을 수신한다 (100). 원자 동작들 유닛 (14)은 파이프의 상태를 나타내는 2개 이상의 파이프 상태 파라미터들을 수정하는 하드웨어-기반의 원자 판독-수정-기록 동작을 수행한다 (102). 파이프는 인터-쓰레드 통신을 지원하기 위한 FIFO-편성된 데이터 버퍼에 대응할 수도 있다.

[0275] 판독-수정-기록 동작은, 그 동작이 판독-수정-기록 동작을 요청한 태스크 또는 쓰레드를 동시에 실행하고 있는 다른 태스크들 및/또는 쓰레드들에 의해 요청될 수도 있는 임의의 다른 판독-수정-기록 동작들로부터의 인터럽트 없이 시작부터 끝까지 수행된다는 점에서 볼때 원자일 수도 있다. 다시 말해서, 원자 판독-수정-기록 동작의 실행 동안, 어떤 다른 쓰레드들 및/또는 태스크들도 현재-실행하는 판독-수정-기록 동작에 의해 수정되는 파이프 상태 파라미터들을 판독하거나 또는 수정불가능할 수도 있다. 이러한 방법으로, 2개 이상의 태스크들 및/또는 쓰레드들이 파이프의 파이프 상태 파라미터들을 동시에 수정하려고 시도하는 경우에 경합 조건들이 회피될 수도 있다.

[0276] 원자 판독-수정-기록 동작은 원자 판독-수정-기록 동작이 하드웨어 유닛에 통신가능하게 커플링된 쓰레드들 및/또는 태스크들에 의해 호출될 수도 있는 단일, 불가분 동작으로서 하드웨어 유닛에 의해 구현될 수도 있다는 점에서 하드웨어-기반일 수도 있다. 단일, 불가분 동작은 하드웨어 유닛이 판독-수정-기록 동작을 요청한 태스크 또는 쓰레드를 동시에 실행하고 있는 다른 태스크들 및/또는 쓰레드들에 의해 요청될 수도 있는 다른 판독-수정-기록 동작들을 수행하는 동작을 인터럽트함이 없이 시작부터 끝까지 그 동작을 수행한다는 점에서 불가분일 수도 있다. 예를 들어, 하드웨어 유닛이 원자 판독-수정-기록 동작을 수행해달라는 다수의 요청들을 수신하면, 하드웨어 유닛은 하나의 원자 판독-수정-기록 동작이 한꺼번에 수행되도록, 그리고 각각의 원자 판독-수정-기록 동작의 실행이 다른 원자 판독-수정-기록 동작을 수행하도록 시작하기 전에 완료되도록, 요청들의 각각에 대해 원자 판독-수정-기록 동작을 순차적으로 수행할 수도 있다. 하드웨어-기반의 원자 판독-수정-기록 동작들을 이용함으로써, 태스크들 및/또는 쓰레드들이 뮤텍스 락들을 이용할 필요 없이 동시에 실행하는 태스크들 및/또는 쓰레드들에 의해 다수의 파이프 상태 파라미터들이 수정될 수도 있다.

[0277] 일부 예들에서, 원자 동작들 유닛 (14)은 하드웨어-기반의 원자 판독-수정-기록 동작들을 수행해 달라는 복수의 요청들 (즉, 지령들)을 병렬로 실행하는 복수의 쓰레드들로부터 수신할 수도 있다. 이러한 예들에서, 원자 동작들 유닛 (14)은 하드웨어-기반의 원자 판독-수정-기록 동작들을 수행해달라는 복수의 요청들을 수신하는 것에 응답하여, 복수의 하드웨어-기반의 원자 판독-수정-기록 동작들을 순차적으로 수행할 수도 있다. 하드웨어-기반의 원자 판독-수정-기록 동작들의 각각은 복수의 요청들의 개개의 하나에 응답하여 수행될 수도 있다. 복수의 하드웨어-기반의 원자 판독-수정-기록 동작들을 순차적으로 수행하는 것은, 제 1 쓰레드로부터의 제 1 요청과 연관된 모든 파이프 상태 파라미터 수정들이 제 2 쓰레드로부터의 제 2 요청과 연관된 파이프 상태 파라미터 수정들 중 임의의 파이프 상태 파라미터 수정을 수행하기 전에 완료되도록, 하드웨어-기반의 원

자 판독-수정-기록 동작들을 수행하는 것을 수반할 수도 있다.

[0278] 예를 들어, 원자 동작들 유닛 (14)은 원자 동작을 수행해 달라는 제 1 쓰레드로부터의 제 1 요청을 수신할 수도 있다. 제 1 요청을 수신하는 것에 응답하여, 원자 동작들 유닛 (14)은 제 1 요청을 서비스하기 위해 제 1 하드웨어-기반의 원자 판독-수정-기록 동작을 수행할 수도 있다. 제 1 하드웨어-기반의 원자 판독-수정-기록 동작을 완료하기 전에, 원자 동작들 유닛 (14)은 원자 동작을 수행해달라는 제 2 쓰레드로부터의 제 2 요청을 수신할 수도 있다. 제 2 요청을 수신하는 것에 응답하여, 원자 동작들 유닛 (14)은 제 1 하드웨어-기반의 원자 판독-수정-기록 동작의 수행이 완료될 때까지, 제 2 요청을 서비스하기 위한 제 2 하드웨어-기반의 원자 판독-수정-기록 동작의 수행을 개시하는 것을 대기할 수도 있다. 제 1 하드웨어-기반의 원자 판독-수정-기록 동작의 수행이 완료된 후, 원자 동작들 유닛 (14)은 제 2 요청을 서비스하기 위해 2개 이상의 파이프 상태 파라미터들을 수정하는 제 2 하드웨어-기반의 원자 판독-수정-기록 동작을 수행할 수도 있다.

[0279] 하드웨어-기반의 원자 판독-수정-기록 동작들을 위에서 설명한 방법으로 순차적으로 수행하는 것은, 원자 동작들 유닛 (14)으로 하여금 동작들의 원자성을 보장가능하게 할 수도 있다. 이러한 방법으로, 파이프-기반의 태스크 간 통신이, 파이프 상태 파라미터들을 수정하는 소프트웨어-기반, 상호 제외 기법들을 이용하는 것과 연관된 성능 단점들 및/또는 소비 전력 단점들을 강요함이 없이, 병렬 컴퓨팅 시스템에서 제공될 수도 있다.

[0280] 일부 예들에서, 파이프 상태 파라미터들은 파이프의 예약 상태를 나타내는 하나 이상의 파이프 예약 상태 파라미터들을 포함할 수도 있다. 예를 들어, 예약 상태 파라미터들은 얼마나 많은 비커밋된 예약들이 현재 파이프에 펜딩 중인지를 나타내는 파라미터, 및 얼마나 많은 패킷 엔트리들이 현재 파이프에 예약되어 있는지를 나타내는 파라미터 중 적어도 하나를 포함할 수도 있다. 추가적인 예들에서, 파이프 상태 파라미터들은 얼마나 많은 패킷들이 현재 파이프에 저장되어 있는지를 나타내는 파라미터, 파이프에 대한 시작 패킷을 저장하는 패킷 엔트리를 나타내는 파라미터, 얼마나 많은 비커밋된 예약들이 현재 파이프에 펜딩 중인지를 나타내는 파라미터, 및 얼마나 많은 패킷 엔트리들이 현재 파이프에 예약되어 있는지를 나타내는 파라미터를 포함할 수도 있다.

[0281] 도 12 는 본 개시물에 따른, 하드웨어-기반의 원자 동작들을 수행하는 다른 예시적인 기법을 예시하는 흐름도이다. 도 12 에 예시된 예시적인 기법은 컴퓨팅 시스템 (10) 의 원자 동작들 유닛 (14)에 대해 설명된다. 일부 예들에서, 원자 동작들 유닛 (14)은 예컨대, GPU (예컨대, GPU (46)), CPU, DSP, 등과 같은, 프로세서 및/또는 프로세싱 유닛에 포함될 수도 있다. 일부 예들에서, 도 13 에 나타낸 예시적인 기법은 도 11 에 예시된 프로세스 블록 (102)을 구현할 수도 있다.

[0282] 원자 동작들 유닛 (14)은 2개 이상의 파이프 상태 파라미터들에 대응하는 값들을 메모리 서브시스템으로부터 판독하는 판독 동작을 수행한다 (104). 원자 동작들 유닛 (14)은 2개 이상의 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 수정 동작을 수행한다 (106). 원자 동작들 유닛 (14)은 파이프 상태 파라미터들에 대한 수정된 값들을 메모리 서브시스템에 기록하는 기록 동작을 수행한다 (108).

[0283] 일부 예들에서, 메모리 서브시스템은 데이터 저장 시스템 (16)을 포함할 수도 있다. 이러한 예들에서, 메모리 서브시스템에 값들을 기록하고 메모리 서브시스템으로부터 값들을 판독하는 것은 메모리 서브시스템에 포함되는 데이터 저장 시스템 (16)에 값들을 기록하고 그로부터 값들을 판독하는 것을 수반할 수도 있다.

[0284] 일부 예들에서, 프로세스 박스들 (104, 106, 및 108)은 인터럽트불가능한 동작들의 세트의 일부로서 수행될 수도 있다. 동작들은 그 동작들이 다른 판독-수정-기록 동작들 (예컨대, 현재 실행하는 판독-수정-기록 동작을 요청한 태스크 또는 쓰레드를 동시에 실행하고 있는 태스크들 및/또는 쓰레드들에 의해 요청되는 다른 판독-수정 기록 동작들) 과 연관될 수도 있는 다른 동작들을 수행하는 실행을 인터럽트함이 없이 시작부터 끝까지 수행된다는 점에서 비-인터럽트가능할 수도 있다. 하드웨어 유닛으로 하여금 인터럽트불가능한 판독-수정-기록 시퀀스를 이러한 방식으로 수행하도록 하는 것은, 하드웨어 유닛으로 하여금 이러한 동작들을 요청할 수도 있는 동시에 실행하는 쓰레드들 및/또는 태스크들에 대해 원자 판독-수정-기록 동작들을 구현가능하게 한다.

[0285] 도 13 은 본 개시물에 따른, 원자 예약 기록 동작 및 원자 예약 판독 동작을 수행하는 예시적인 기법을 예시하는 흐름도이다. 도 13 에 예시된 예시적인 기법은 컴퓨팅 시스템 (10)의 원자 동작들 유닛 (14)에 대해 설명된다. 일부 예들에서, 원자 동작들 유닛 (14)은 예컨대, GPU (예컨대, GPU (46)), CPU, DSP, 등과 같은, 프로세서 및/또는 프로세싱 유닛에 포함될 수도 있다. 일부 예들에서, 도 13 에 나타낸 예시적인 기법은 도 12 에 나타낸 예시적인 기법을 구현할 수도 있다.

[0286] 원자 동작들 유닛 (14)은 파이프 상태 파라미터 (PSP) 값들을 메모리 서브시스템으로부터 판독하는 판독 동작

을 수행한다 (110). 관독 동작을 수행한 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터 값들 중 하나 이상을 요청하는 쓰레드 및/또는 태스크로 반환하는 반환 동작을 수행한다 (112).

[0287] 관독 동작을 수행한 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 수정 동작을 수행한다 (블록들 114-118). 수정 동작을 수행하기 위해, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 중 하나 이상에 기초하여 파이프가 예약 요청을 지원하기에 충분한 가용 엔트리들을 가지고 있는지 여부를 결정하는 동작을 수행한다 (114). 파이프가 예약 요청을 지원하기에 충분한 가용 엔트리들을 가지고 있다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 예약이 파이프에 대해 이루어졌다는 것을 수정된 값들이 반영하도록 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작을 수행한다 (116).

[0288] 파이프가 예약 요청을 지원하기에 충분한 가용 엔트리들을 가지고 있지 않다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 예약이 파이프에 대해 이루어졌다는 것을 수정된 값들이 반영하도록 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 동작을 수행하지 않을 수도 있다 (118). 다시 말해서, 원자 동작들 유닛 (14)은 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록 (즉, 예약이 파이프에 대해 이루어지지 않았다는 것을 수정된 값들이 반영하도록) 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0289] 원자 예약 기록 동작이 수행되는 예들에서, 파이프가 예약 요청을 지원하기에 충분한 가용 엔트리들을 가지고 있는지 여부를 결정하기 위해, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 중 하나 이상에 기초하여, 파이프가 기록 예약 요청을 지원하기에 충분한 비커밋된 및 미예약된 엔트리들을 가지고 있는지 여부를 결정할 수도 있다. 원자 예약 관독 동작이 수행되는 예들에서, 파이프가 예약 요청을 지원하기에 충분한 가용 엔트리들을 가지고 있는지 여부를 결정하기 위해, 원자 동작들 유닛 (14)은 파이프가 패킷 데이터가 관독 예약 요청을 지원하기 위해 저장되는 충분한 비관독된 및 미예약된 패킷 엔트리들을 가지고 있는지 여부를 결정할 수도 있다.

[0290] 수정된 값들을 발생시킨 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들에 대한 수정된 값들을 메모리 서브시스템에 기록하는 기록 동작을 수행한다 (108). 예약 동작이 이제 완료된다

[0291] 일부 예들에서, 예약 동작은 예약 기록 동작일 수도 있다. 이러한 예들에서, 결정 블록 (114)은 일부 예들에서, 도 2 의 라인 16 상에 나타낸 수식을 평가함으로써 구현될 수도 있다. 게다가, 프로세스 블록 (116)에서 설명된 바와 같이 예약이 파이프에 대해 이루어졌다는 것을 반영하는 수정된 파이프 상태 파라미터 값들을 발생시키기 위해, 원자 동작들 유닛 (14)은 도 2 의 라인들 17-20 에 규정된 방정식들에 기초하여, 파이프 상태 파라미터 값들을 업데이트할 수도 있다.

[0292] 추가적인 예들에서, 예약 동작은 예약 관독 동작일 수도 있다. 이러한 예들에서, 결정 블록 (114)은 일부 예들에서, 도 4 의 라인 16 상에 나타낸 수식을 평가함으로써 구현될 수도 있다. 게다가, 프로세스 블록 (116)에서 설명된 바와 같이 예약이 파이프에 대해 이루어졌다는 것을 반영하는 수정된 파이프 상태 파라미터 값들을 발생시키기 위해, 원자 동작들 유닛 (14)은 도 4 의 라인들 17-10 에 규정된 방정식들에 기초하여, 파이프 상태 파라미터 값들을 업데이트할 수도 있다.

[0293] 도 14 는 본 개시물에 따른, 원자 커밋 기록 동작 및 원자 커밋 관독 동작을 수행하는 예시적인 기법을 예시하는 흐름도이다. 도 14 에 예시된 예시적인 기법은 컴퓨팅 시스템 (10)의 원자 동작들 유닛 (14)에 대해 설명된다. 일부 예들에서, 원자 동작들 유닛 (14)은 예컨대, GPU (예컨대, GPU (46)), CPU, DSP, 등과 같은, 프로세서 및/또는 프로세싱 유닛에 포함될 수도 있다. 일부 예들에서, 도 14 에 나타낸 예시적인 기법은 도 12 에 나타낸 예시적인 기법을 구현할 수도 있다.

[0294] 원자 동작들 유닛 (14)은 파이프 상태 파라미터 (PSP) 값들을 메모리 서브시스템으로부터 관독하는 관독 동작을 수행한다 (122). 관독 동작을 수행한 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들에 대한 수정된 값들을 발생시키는 수정 동작을 수행한다 (블록들 124-130). 수정 동작을 수행하기 위해, 원자 동작들 유닛 (14)은 예약이 커밋되었다는 것을 반영하기 위해 제 1 파이프 상태 파라미터 (예컨대, 예약 카운트 파라미터)에 대한 수정된 값을 발생시킬 수도 있다 (124). 다시 말해서, 원자 동작들 유닛 (14)은 커밋 동작을 수행하기 전에 파이프가 얼마나 많은 비커밋된 예약들을 갖는지에 관하여 하나 더 적은 비커밋된 예약을 파이프가 가지고 있다는 것을 제 1 파이프 상태 파라미터에 대한 수정된 값이 반영하도록 제 1 파이프 상태 파라미터에 대한 수정된 값을 발생시키는 동작을 수행할 수도 있다.

[0295] 원자 동작들 유닛 (14)은 파이프 상태 파라미터들 중 하나 이상에 기초하여 파이프에 대한 모든 예약들이 커밋

되었는지 여부를 결정하는 동작을 수행한다 (126).      파이프에 대한 모든 예약들이 커밋되었다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 엔트리들이 파이프에 커밋되었다는 것을 수정된 값들이 반영하도록 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시키는 동작을 수행한다 (128).

[0296]      파이프에 대한 모든 예약들이 커밋되지 않았다고 결정하는 것에 응답하여, 원자 동작들 유닛 (14)은 커밋된 엔트리들이 파이프에 커밋되었다는 것을 수정된 값들이 반영하도록 파이프 상태 파라미터들에 대한 하나 이상의 수정된 값들을 발생시키는 동작을 수행하지 않는다 (130).      다시 말해서, 원자 동작들 유닛 (14)은 수정된 값들이 파이프 상태 파라미터들에 대한 이전 값들과 동일하도록, 제 1 파이프 상태 파라미터 (예컨대, 예약 카운트 파라미터) 가 아닌, 파이프 상태 파라미터들에 대한 수정된 값들을 발생시킬 수도 있다.

[0297]      원자 커밋 기록 동작이 수행되는 예들에서, 엔트리들이 파이프에 커밋되었다는 것을 반영하기 위해 파이프 상태 파라미터 값들을 수정하는 것은 커밋된 엔트리들이 파이프에 배치되어 있고 소비자 태스크들에 이용가능하다는 것을 반영하기 위해 파이프 상태 파라미터 값들을 수정하는 것을 수반할 수도 있다.      원자 커밋 판독 동작이 수행되는 예들에서, 엔트리들이 파이프에 커밋되었다는 것을 반영하기 위해 파이프 상태 파라미터 값들을 수정하는 것은, 커밋된 엔트리들이 파이프로부터 제거되었다는 것을 반영하기 위해 파이프 상태 파라미터 값들을 수정하는 것을 수반할 수도 있다.

[0298]      수정된 값들을 발생시킨 후, 원자 동작들 유닛 (14)은 파이프 상태 파라미터들에 대한 수정된 값들을 메모리 서브시스템에 기록하는 기록 동작을 수행한다 (132).      커밋 동작이 이제 완료된다.

[0299]      일부 예들에서, 커밋 동작은 커밋 기록 동작일 수도 있다.      이러한 예들에서, 프로세스 블록 (124)은 일부 예들에서, 도 3 의 라인 8 상에 나타낸 방정식에 기초하여 구현될 수도 있다.      게다가, 결정 블록 (126)은 일부 예들에서, 도 3 의 라인 9 상에 나타낸 수식을 평가함으로써 구현될 수도 있다.      또, 프로세스 블록 (128)에서 설명된 바와 같이 엔트리들이 파이프에 커밋되었다는 것을 반영하는 수정된 파이프 상태 파라미터 값들을 발생시키기 위해, 원자 동작들 유닛 (14)은 도 3 의 라인들 10-11 에 규정된 방정식들에 기초하여, 파이프 상태 파라미터 값들을 업데이트할 수도 있다.

[0300]      추가적인 예들에서, 커밋 동작은 커밋 판독 동작일 수도 있다.      이러한 예들에서, 프로세스 블록 (124)은 일부 예들에서, 도 5 의 라인 8 상에 나타낸 방정식에 기초하여 구현될 수도 있다.      게다가, 결정 블록 (126)은 일부 예들에서, 도 5 의 라인 9 상에 나타낸 수식을 평가함으로써 구현될 수도 있다.      또, 프로세스 블록 (128)에서 설명된 바와 같이 엔트리들이 파이프에 커밋되었다는 것을 반영하는 수정된 파이프 상태 파라미터 값들을 발생시키기 위해, 원자 동작들 유닛 (14)은 도 5 의 라인들 10-12 에 규정된 방정식들에 기초하여, 파이프 상태 파라미터 값들을 업데이트할 수도 있다.

[0301]      일부 예들에서, 본 개시물의 기법들은 OpenCL™ 2.0 파이프들에 대한 상대적으로 효율적인 솔루션을 구현할 수도 있다.      이 기법들은 일부 예들에서, 특수화된 원자 기능들을 이용하여, OpenCL™ 2.0 파이프들을 구현할 수도 있다.

[0302]      OpenCL™ 2.0 파이프는 예약 피처 (reservation feature) 를 통해서 다수의 제작자/다수의 소비자들에 의해 사용될 수 있는 FIFO 로서 역할을 한다.      파이프에 액세스하는 것에 대한 의미들은 reserve(cnt) 함수를 호출하는 것을 포함할 수도 있으며, 여기서, cnt 는 예약을 위한 요청된 패킷들의 개수를 나타낸다. reserve(cnt) 함수가 성공적이면, reserve(cnt) 함수는 예약 식별 값 (즉, reserve\_id) 을 반환할 수도 있다.      파이프에 액세스하는 것에 대한 의미들은 reserve(cnt) 함수가 성공적이면, 0 내지 cnt-1 의 인덱스 및 reserved\_id 를 이용하여 판독 파이프 지령을 호출하는 것을 더 포함할 수도 있다.      인덱스는 어느 패킷 엔트리들이 파이프로부터 판독되는지를 표시할 수도 있다.      파이프에 액세스하는 것에 대한 의미들은 파이프를 판독하는 것이 종료되었으며 패킷들이 파이프로부터 제거될 수도 있다는 것을 나타내는 commit(reserve\_id) 지령을 더 포함할 수도 있다.

[0303]      파이프의 상태를 유지하는 것은 다수의 카운터들의 동시적인 (즉, 원자) 수정을 필요로 할 수도 있다.      이것은 뮤텍스 락들을 이용하여 달성될 수도 있으며, 이것은 결국 비교-교환 원자 동작들을 이용하여 구현될 수 있다.      그러나, 뮤텍스 락들을 이용하는 것은, 다수의 작업-아이템들이 락에 대해 경합할 수도 있기 때문에, 비효율적일 수도 있다.      이것은 성공적인 획득을 위해 요구되는 다수의 시도들을 초래하여, 전력을 낭비할 수도 있다.

[0304]      본 개시물의 기법들은 일부 예들에서, 동작 당 단일 하드웨어 명령 (single hardware instruction per operation) 을 이용하여, 파이프 예약, 커밋 및 쿼리 동작들을 지원하도록 설계될 수도 있는, 하드웨어 원자 동

작들을 이용할 수도 있다. 일부 예들에서, 본 개시물에서 설명된 원자 동작들은 전역 GPU 원자학 (atomics) 을 구현하는 GPU 파이프라인의 일부에서 구현될 수도 있다. 추가적인 예들에서, 본 개시물에서 설명된 원자 동작들은 구현예가 공유된 가상 메모리를 지원하면, 호스트 CPU 와 GPU 사이에 파이프가 사용될 수 있도록 할 수도 있다.

[0305] 일부 예들에서, 본 개시물의 기법들은 GPU 의 세이더 블록 내에서 내장 파이프 기능들을 수행할 수도 있으며, 세이더 블록에 대한 명령 세트 아키텍처 (ISA) 는 본 개시물에서 설명된 원자 동작 지령들 중 하나 이상을 포함 할 수도 있다. 추가적인 예들에서, 본 개시물의 기법들은 전용 하드웨어 회로들을 이용하여, 원자 동작들을 구현할 수도 있다.

[0306] 일부 예들에서, 본 개시물의 기법들은 특수화된 원자학을 GPU 의 통합된 캐시 블록 (예컨대, L2 캐시 블록) 에 추가할 수도 있다. 특수화된 원자학은 일부 예들에서, 추가적인 ISA 명령들로 액세스될 수도 있다. 일부 예들에서, 파이프들은 예약 메커니즘 (예컨대, reserve\_read, read, commit\_read, 또는, reserve\_write, write, commit\_write) 을 이용하여 액세스될 수도 있다.

[0307] 일부 예들에서, 파이프에 대한 파이프 상태는 다음 4개의 32-비트 수들로 표현될 수도 있다: (1) 제 1 패킷 오프셋; (2) 패킷 카운트; (3) 예약된 패킷 카운트; 및 (4) 예약 카운트. 이러한 예들에서, 이를 32-비트 수들은 일부 예들에서, 본 개시물에서 설명된 파이프 상태 파라미터들에 대응할 수도 있다. 이러한 예들에서, 4개의 32-비트 수들 (예컨대, 파이프 상태 파라미터들) 은 단일 메모리 트랜잭션으로 판독/기록하는 것이 가능하도록, 128-비트 블록 내에 128-비트 정렬로 로케이트될 수도 있다.

[0308] 일부 예들에서, 파이프의 할당된 사이즈는 2 의 거듭제곱일 수도 있다. 그러나, 파이프의 실제 심도는 할당된 사이즈보다 작을 수 있다. 이것은 OpenCL™ API 의 유연성에 기인할 수도 있다.

[0309] 본 개시물의 기법들은 일부 예들에서, OpenCL™ 파이프 예약, 커밋 및 쿼리에 대한 지원을 제공할 수도 있다. OpenCL™ 파이프들은 판독 (즉, 입력 파이프) 용으로 또는 기록 (즉, 출력 파이프) 용으로 사용될 수도 있다. 커널은 일부 예들에서, 주어진 파이프로부터 판독하거나 또는 그에 기록할 수도 있지만, 양쪽 모두는 아니다.

[0310] 일부 예들에서, 본 개시물의 기법들은 GPU 의 세이더 프로세서, 스트림 프로세서, 및/또는 통합된 캐시에서 구현될 수도 있다. 일부 예들에서, 본 개시물의 기법들은 커널 당 비제한 파이프들의 개수가 지원될 (예컨대, 파이프 상태를 유지하기 위해 오직 가용 메모리 및 상수 레지스터들에 의해 제한될) 수 있도록 할 수도 있다. 추가적인 예들에서, 파이프 상태를 메모리에 유지하는 피처는 호스트/디바이스 전체에 걸쳐서 또는 다수의 디바이스들 전체에 걸쳐서 파이프들을 포함하기 위해 OpenCL™ 2.0 의 공유된 가상 메모리 피처와 커플링될 수도 있다.

[0311] 일부 예들에서, 파이프는 FIFO 로서 편성된 데이터를 저장하는 메모리 오브젝트일 수도 있다. 파이프는 대안적으로 파이프 오브젝트로서 지정될 수도 있다. 일부의 경우, 파이프 오브젝트는 파이프로부터 판독하거나 및/또는 그에 기록하는 내장 기능들을 이용하여 액세스될 수도 있다. 추가적인 경우에, 파이프 오브젝트들은 호스트 디바이스로부터 액세스불가능할 수도 있다. 일부 예들에서, 파이프 오브젝트는 다음 정보를 캡슐화할 수도 있다: 바이트 단위 패킷 사이즈, 패킷들에서의 파이프의 최대 용량, 현재 파이프에서의 패킷들의 개수를 나타내는 정보, 및 파이프에 포함되는 데이터 패킷들.

[0312] 일부 예들에서, 파이프 메모리 오브젝트는 데이터 아이템들의 순서화된 시퀀스에 개념적으로 대응할 수도 있다. 예를 들어, 파이프는 2개의 종점들, 즉, 데이터 아이템들이 삽입되는 기록 종점, 및 데이터 아이템들이 제거되는 판독 종점을 가질 수도 있다. 일부 예들에서, 임의의 시간에, 오직 하나의 커널 인스턴스 (예컨대, 태스크) 만이 파이프에 기록될 수도 있으며, 오직 하나의 커널 인스턴스 (예컨대, 태스크) 만이 파이프로부터 판독될 수도 있다. 일부 예들에서, 파이프는 하나의 커널 인스턴스가 파이프의 기록 종점에 (제작자로서) 접속되는 반면 다른 커널 인스턴스가 파이프의 판독 종점에 (소비자로서) 접속되는 제작자-소비자 설계 패턴을 지원할 수도 있다.

[0313] 일부의 경우, 파이프는 메모리 오브젝트의 유형일 수도 있다. 메모리 오브젝트는 일부 예들에서, 전역 메모리 (예컨대, 시스템 메모리) 의 콘텐츠를 포함할 수도 있다. 일부 예들에서, 메모리 오브젝트는 전역 메모리의 참조 카운트된 영역에 대한 핸들을 지정할 수도 있다.

[0314] 본 개시물에서 설명하는 기법들은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로, 적어도 부분적으로, 구현될 수도 있다. 예를 들어, 설명되는 기법들의 여러 양태들은, 하나 이상의 마이크로프로세서들,

디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 또는 임의의 다른 등가의 통합 또는 이산 로직 회로, 뿐만 아니라, 이러한 구성요소들의 임의의 조합들을 포함한, 하나 이상의 프로세서들 내에서 구현될 수도 있다. 용어 "프로세서" 또는 "프로세싱 회로"는 일반적으로 단독으로 또는 다른 로직 회로와 조합한, 전술한 로직 회로 중 임의의 로직 회로, 또는 프로세싱을 수행하는 별개의 하드웨어와 같은 임의의 다른 등가 회로를 지칭할 수도 있다.

[0315] 이러한 하드웨어, 소프트웨어, 및 펌웨어는 본 개시물에서 설명하는 여러 동작들 및 기능들을 지원하기 위해 동일한 디바이스 내에 또는 별개의 디바이스들 내에 구현될 수도 있다. 게다가, 설명되는 유닛들, 모듈들 또는 구성요소들 중 임의의 것은 별개의 상호운용가능한 로직 디바이스들과 함께 또는 별개로 구현될 수도 있다.

모듈들 또는 유닛들과는 상이한 특성들의 서술은 상이한 기능적 양태들을 강조하려고 의도된 것이며, 이러한 모듈들 또는 유닛들이 별개의 하드웨어 또는 소프트웨어 구성요소들로 실현되어야 한다는 것을 반드시 암시하지는 않는다. 대신, 하나 이상의 모듈들 또는 유닛들과 연관되는 기능은 별개의 하드웨어, 펌웨어, 및/또는 소프트웨어 구성요소들에 의해 수행되거나, 또는 공통의 또는 별개의 하드웨어 또는 소프트웨어 구성요소들 내에 통합될 수도 있다.

[0316] 본 개시물에서 설명하는 기법들은 또한 명령들을 저장하는 컴퓨터-판독가능 저장 매체와 같은 컴퓨터-판독가능 매체에 저장되거나, 수록되거나 또는 인코딩될 수도 있다. 컴퓨터-판독가능 매체에 수록되거나 또는 인코딩되는 명령들은 하나 이상의 프로세서들로 하여금, 예컨대, 명령들이 하나 이상의 프로세서들에 의해 실행될 때, 본원에서 설명되는 기법들을 수행하도록 할 수도 있다. 일부 예들에서, 컴퓨터-판독가능 매체는 비일시적 컴퓨터-판독가능 저장 매체일 수도 있다. 컴퓨터 판독가능 저장 매체들은 랜덤 액세스 메모리 (RAM), 판독 전용 메모리 (ROM), 프로그래밍가능 판독전용 메모리 (PROM), 소거가능한 프로그래밍가능 판독전용 메모리 (EPROM), 전자적으로 소거가능한 프로그래밍가능 판독전용 메모리 (EEPROM), 플래시 메모리, 하드 디스크, CD-ROM, 플로피 디스크, 카세트, 자기 매체들, 광학 매체들, 또는 유형인 다른 컴퓨터 판독가능 저장 매체들을 포함할 수도 있다.

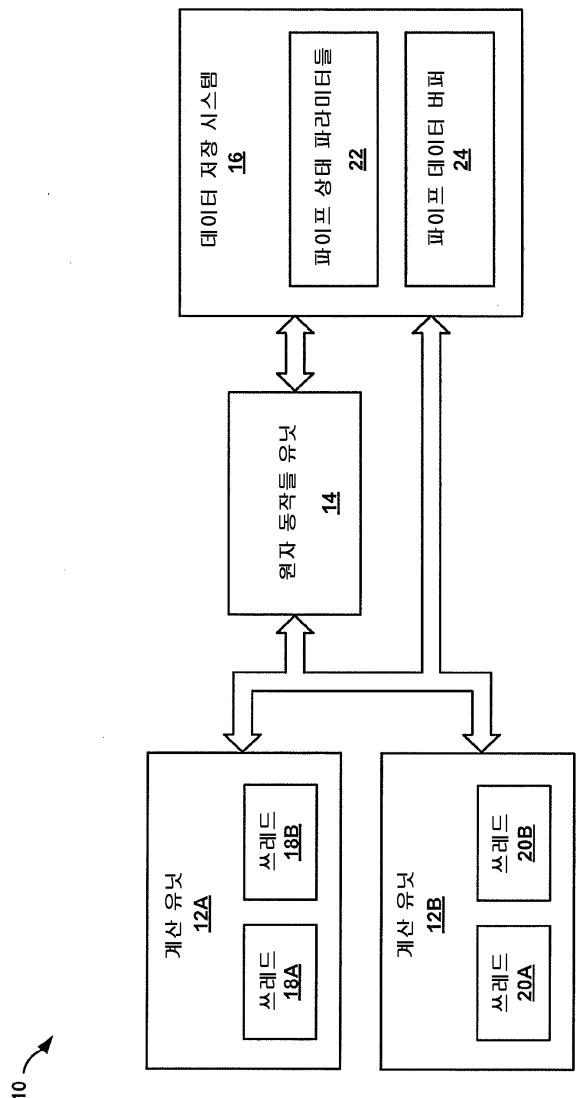
[0317] 컴퓨터-판독가능 매체는 위에서 리스트된 것들과 같은, 유형의 저장매체에 대응하는 컴퓨터-판독가능 저장 매체들을 포함할 수도 있다. 컴퓨터-판독가능 매체는 또한 예컨대, 통신 프로토콜에 따라서 한 장소로부터 또 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함한, 통신 매체들을 포함할 수도 있다.

이러한 방법으로, 어구 "컴퓨터-판독가능 매체들"은 일반적으로 (1) 비일시적 유형의 컴퓨터-판독가능 저장 매체, 및 (2) 일시적인 신호 또는 반송파와 같은 비-유형의 컴퓨터-판독가능 통신 매체에 대응할 수도 있다.

[0318] 여러 양태들 및 예들이 설명되었다. 그러나, 다음 청구항들의 범위로부터 일탈함이 없이 본 개시물의 구조 또는 기법들에 대해 변경들이 이루어질 수 있다.

도면

도면1



10 ↗

## 도면2

```

1 #define FST_PKT 0 /* First packet offset */
2 #define PKT_CNT 1 /* Packet count */
3 #define RES_CNT 2 /* Reserved packet count */
4 #define NUM_RES 3 /* Number of reservations */

5 atomic_reserve_write
6 {
7     uint32 *pipe_addr,
8     uint32 pipe_size_minus_one,
9     uint32 max_depth,
10    uint32 num_packets
11 }

12 {
13     uint32 cnts[4] = load(pipe_addr); // 128-bit load
14
15     // Return 96-bit to the caller
16     return cnts[FST_PKT], cnts[PKT_CNT], cnts[RES_CNT];
17
18     if((max_depth - cnts[PKT_CNT] - cnts[RES_CNT]) >= num_packets) {
19         cnts[RES_CNT] += num_packets;
20         if(num_packet != 0) {
21             cnts[NUM_RES]++;
22         }
23     }
}

```

## 도면3

```

1 #define FST_PKT 0 /* First packet offset */
2 #define PKT_CNT 1 /* Packet count */
3 #define RES_CNT 2 /* Reserved packet count */
4 #define NUM_RES 3 /* Number of reservations */

5 void atomic_commit_write(uint32 pipe_addr)
6 {
7     uint32 cnts[4] = load(pipe_addr); // 128 bit load
8     cnts[NUM_RES]--;
9     if(cnts[NUM_RES] == 0){
10         cnts[PKT_CNT] += cnts[RES_CNT];
11         cnts[RES_CNT] = 0;
12     }
13     store(pipe_addr, cnts); // store the modified pipe counters back to memory.
14 }

```

## 도면4

```

1 #define FST_PKT 0 /* First packet offset */
2 #define PKT_CNT 1 /* Packet count */
3 #define RES_CNT 2 /* Reserved packet count */
4 #define NUM_RES 3 /* Number of reservations */

5 atomic_reserve_read
6 {
7     uint32 *pipe_addr,
8     uint32 pipe_size_minus_one,
9     uint32 max_depth,
10    uint32 num_packets
11 }
12 {
13     uint32 cnts[4] = load(pipe_addr); // 128-bit load
14     // Return 96-bit to the caller
15     return cnts[FST_PKT], cnts[PKT_CNT], cnts[RES_CNT];
16     if ((cnts[RES_CNT] + num_packets) <= cnts[PKT_CNT]) {
17         cnts[RES_CNT] += num_packets;
18         if (num_packets != 0) {
19             cnts[NUM_RES]++;
20         }
21         store(pipe_addr, cnts); // 128-bit store
22     }
23 }

```

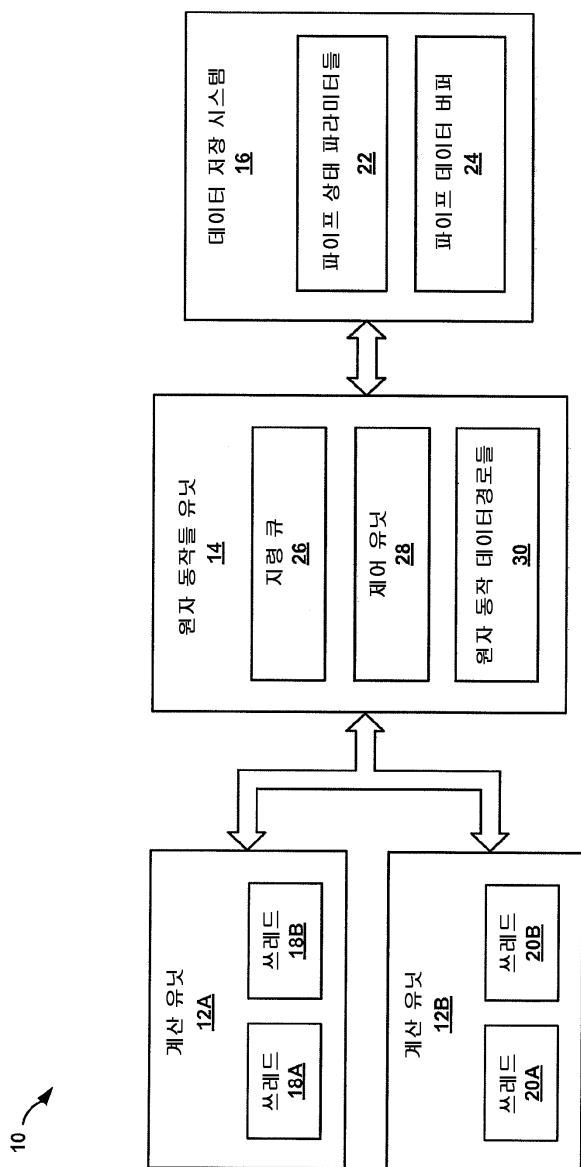
도면5

```

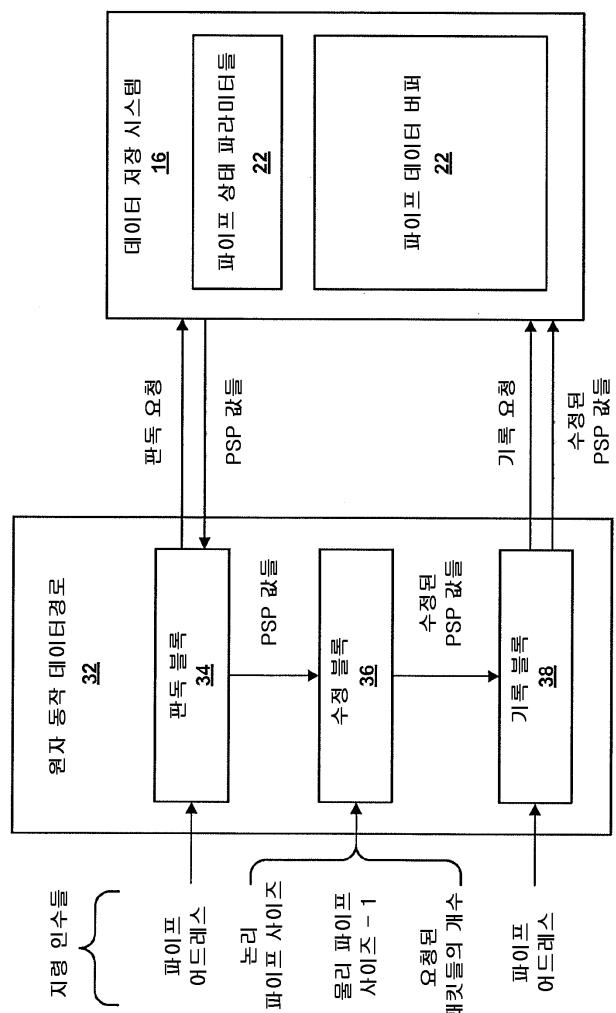
1 #define FST_PKT 0 /* First packet offset */
2 #define PKT_CNT 1 /* Packet count */
3 #define RES_CNT 2 /* Reserved packet count */
4 #define NUM_RES 3 /* Number of reservations */

5 void atomic_commit_read(uint32 *pipe_addr, uint32 pipe_size_minus_one)
6 {
7     uint32 cnts[4] = load(pipe_addr); // 128 bit load
8     cnts[NUM_RES]--;
9     if(cnts[NUM_RES] == 0){
10         cnts[FST_PKT] = (cnts[FST_PKT] + cnts[RES_CNT]) & pipe_size_minus_one;
11         cnts[PKT_CNT] -= cnts[RES_CNT];
12         cnts[RES_CNT] = 0;
13     }
14     store(pipe_addr, cnts); // store the modified pipe counters back to memory.
15 }
```

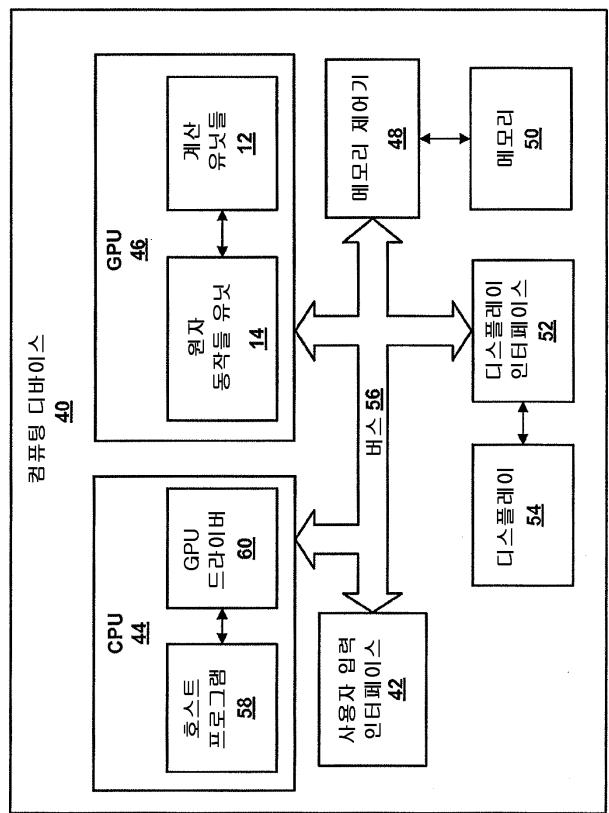
도면6



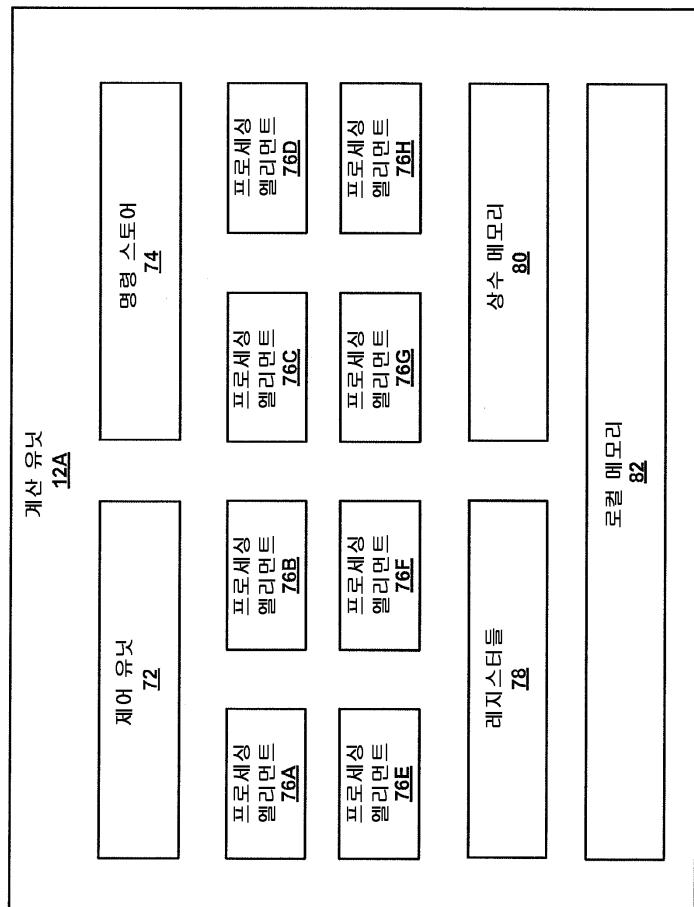
## 도면7



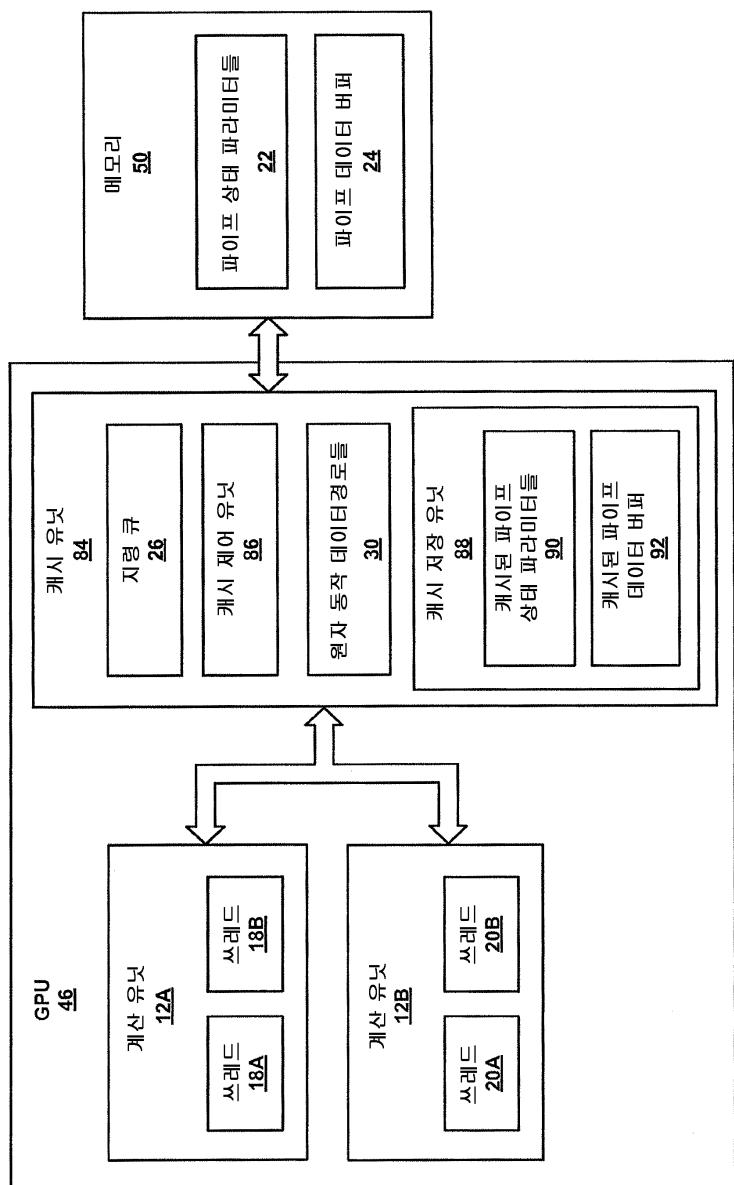
## 도면8



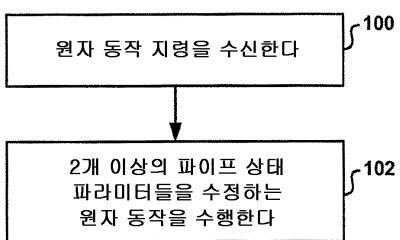
## 도면9



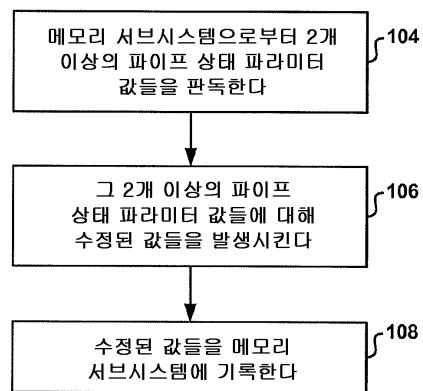
도면10



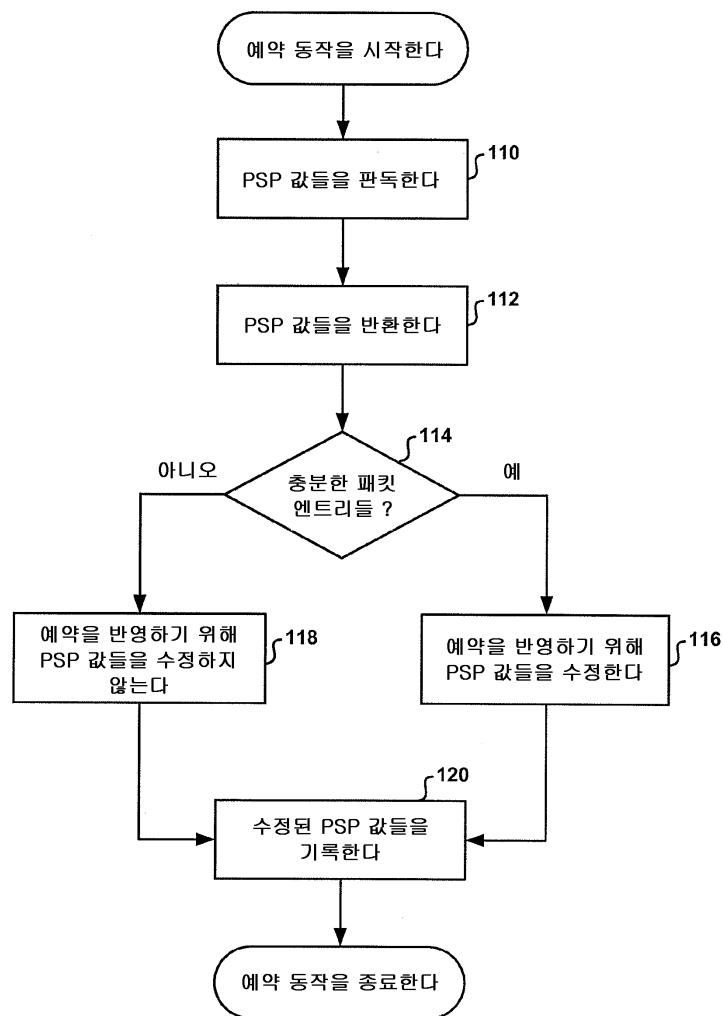
도면11



## 도면12



## 도면13



## 도면14

