



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년08월08일
(11) 등록번호 10-1765830
(24) 등록일자 2017년08월01일

(51) 국제특허분류(Int. Cl.)
G06F 9/48 (2006.01) G06F 1/32 (2006.01)
G06F 9/38 (2006.01) G06F 9/50 (2006.01)
(52) CPC특허분류
G06F 9/4893 (2013.01)
G06F 1/3243 (2013.01)
(21) 출원번호 10-2016-0056308
(22) 출원일자 2016년05월09일
심사청구일자 2016년05월09일
(56) 선행기술조사문헌
KR1020140140943 A*
(뒷면에 계속)

(73) 특허권자
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
백웅기
울산광역시 울주군 언양읍 유니스트길 50
김규연
울산광역시 울주군 언양읍 유니스트길 50
(74) 대리인
유미특허법인

전체 청구항 수 : 총 16 항

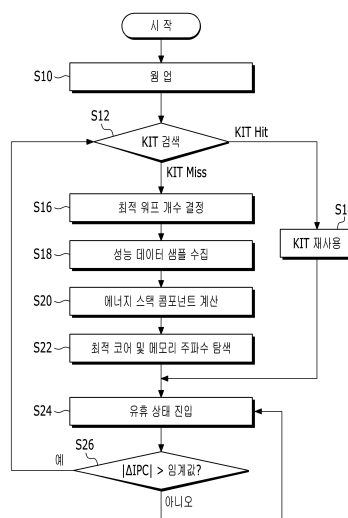
심사관 : 유진태

(54) 발명의 명칭 멀티 코어 시스템 및 그 구동 방법

(57) 요약

실시예에 따른 멀티 코어 시스템은 응용 프로그램에 포함된 복수의 작업 중 할당된 적어도 하나의 작업을 워프 단위로 처리하는 복수의 코어, 복수의 코어의 입출력 데이터를 저장하는 복수의 메모리, 그리고 응용 프로그램의 사이클 당 지시(IPC, instruction per cycle)를 최대화하는 워프 개수를 결정하고, 복수의 코어가 할당된 적어도 하나의 작업을 워프 개수 단위로 처리하도록 제어하는 전력 관리부를 포함한다.

대표도 - 도3



(52) CPC특허분류

G06F 9/3877 (2013.01)
G06F 9/4843 (2013.01)
G06F 9/5038 (2013.01)
G06F 9/5094 (2013.01)

(56) 선행기술조사문헌

KR1020130093995 A*
KR1020120106089 A
KR1020150067869 A
KR1020140134190 A
KR1020110075295 A

*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호 1711024637
부처명 미래창조과학부
연구관리전문기관 한국연구재단

연구사업명 일반사업

연구과제명 대규모 확장 가능한 고성능 저전력 매니코어 컴퓨팅을 위한 시스템 소프트웨어 및 하드웨어 기법 연구와 개발

기 여 율 1/2

주관기관 울산과학기술원

연구기간 2015.05.01 ~ 2016.04.30

이 발명을 지원한 국가연구개발사업

과제고유번호 1711028338
부처명 미래창조과학부
연구관리전문기관 정보통신산업진흥원

연구사업명 일반사업

연구과제명 빅데이터 처리 고도화 핵심 기술개발사업 총괄 및 고성능 컴퓨팅 기술을 활용한 성능 가속화 기술 개발

기 여 율 1/2

주관기관 울산과학기술원

연구기간 2015.06.01 ~ 2016.05.31

명세서

청구범위

청구항 1

응용 프로그램에 포함된 복수의 작업 중 할당된 적어도 하나의 작업을 워프 단위로 처리하는 복수의 코어,

상기 복수의 코어의 입출력 데이터를 저장하는 복수의 메모리, 그리고

상기 응용 프로그램의 사이클 당 지시(IPC, instruction per cycle)를 최대화하는 워프 개수를 결정하고, 상기 복수의 코어가 상기 할당된 적어도 하나의 작업을 상기 워프 개수 단위로 처리하도록 제어하는 전력 관리부를 포함하고,

상기 전력 관리부는,

복수의 작업에 대한 메타 데이터를 저장하는 전력 메모리, 그리고

상기 전력 메모리에서 상기 할당된 적어도 하나의 작업에 대한 메타 데이터를 검색하는 전력 제어부를 포함하며,

상기 전력 제어부는 상기 복수의 코어가 동작하는 코어 주파수 및 상기 복수의 메모리가 동작하는 메모리 주파수에 따른 성능 데이터를 복수로 수집하고, 상기 성능 데이터를 이용하여 하기의 수학적 식 1과 같은 성능 추정 모델을 설계하는,

멀티 코어 시스템.

[수학적 식 1]

$$T = \alpha_c \cdot f_c + \alpha_m \cdot f_m + \beta$$

여기서, T 는 상기 성능 데이터, α_c , α_m 및 β 는 상기 성능 추정 모델의 계수, f_c 는 코어 동작 주파수, f_m 는 메모리 동작 주파수

청구항 2

삭제

청구항 3

제1 항에 있어서,

상기 메타 데이터는,

작업 ID, 워프 개수, 소정 코어 주파수, 소정 메모리 주파수, 및 IPC에 대한 정보를 포함하는 멀티 코어 시스템.

청구항 4

제1 항에 있어서,

상기 전력 제어부는 상기 메타 데이터가 검색되지 않는 경우, 상기 워프 개수를 결정하는 멀티 코어 시스템.

청구항 5

삭제

청구항 6

제1 항에 있어서,

상기 복수의 성능 데이터는 상기 코어 주파수와 상기 메모리 주파수가 모두 최소일 때의 제1 성능 데이터, 상기 코어 주파수가 최소이고 상기 메모리 주파수가 최대일 때의 제2 성능 데이터, 및 상기 코어 주파수가 최대이고, 상기 메모리 주파수가 최소일 때의 제3 성능 데이터를 포함하는 멀티 코어 시스템.

청구항 7

제1 항에 있어서,

상기 성능 데이터는 하기의 수학적 식 2로 계산되는 멀티 코어 시스템.

[수학적 식 2]

$$T = \frac{I}{t}$$

여기서, I 는 폐기된 지시(instruction)의 개수, t 는 경과 시간

청구항 8

제1 항에 있어서,

상기 전력 제어부는 상기 코어 주파수와 상기 메모리 주파수가 모두 최소일 때의 제1 에너지 소비 데이터, 상기 코어 주파수가 최소이고 상기 메모리 주파수가 최대일 때의 제2 에너지 소비 데이터, 및 상기 코어 주파수가 최대이고, 상기 메모리 주파수가 최소일 때의 제3 에너지 소비 데이터를 수집하고, 상기 제1 에너지 소비 데이터, 상기 제2 에너지 소비 데이터 및 상기 제3 에너지 소비 데이터를 이용하여 하기의 수학적 식 3 내지 5로 복수의 에너지 스택 컴포넌트를 산출하는 멀티 코어 시스템.

[수학적 식 3]

$$E_C = \frac{E_{LL} - E_{HL} - (1 - \frac{T_{LL}}{T_{HL}})E_S}{k_{fC}}$$

[수학적 식 4]

$$E_M = \frac{E_{LL} - E_{LH} - (1 - \frac{T_{LL}}{T_{LH}})E_S}{k_{fM}}$$

[수학적 식 5]

$$E_S = \frac{(r_{fC}^2 r_{fM}^2 - 1)E_{LL} + k_{fC}E_{LH} + k_{fM}E_{HL}}{-1 + r_{fC}^2 r_{fM}^2 + k_{fC} \frac{T_{LL}}{T_{LH}} k_{fM} \frac{T_{LL}}{T_{HL}}}$$

여기서, E_C , E_M 및 E_S 는 상기 복수의 에너지 스택 컴포넌트, E_{LL} 은 상기 제1 에너지 소비 데이터, E_{LH} E_{LH} 은

상기 제2 에너지 소비 데이터, E_{HL} 은 상기 제3 에너지 소비 데이터, $r_{fC} = \frac{f_{CH}}{f_{CL}}$, $r_{fM} = \frac{f_{MH}}{f_{ML}}$, $k_{fC} = 1 - r_{fC}^2$, $k_{fM} = 1 - r_{fM}^2$

청구항 9

제8 항에 있어서,

상기 전력 제어부는 상기 복수의 에너지 스택 컴포넌트를 이용하여 하기의 수학적 식 6과 같은 에너지 소비 추정 모델을 설계하는 멀티 코어 시스템.

[수학식 6]

$$E_{f_C f_M} = \left(\frac{f_C}{f_{CL}}\right)^2 E_C + \left(\frac{f_M}{f_{ML}}\right)^2 E_M + \frac{T_{LL}}{T_{f_C f_M}} E_S$$

청구항 10

제9 항에 있어서,

상기 전력 제어부는 상기 성능 추정 모델 및 상기 에너지 소비 추정 모델을 이용하여, 현재 사이클 동안 제1 코어 주파수 및 제1 메모리 주파수에서의 제1 지시 당 줄 값(JPI, joule per instruction)을 계산하고, 상기 제1 지시 당 줄 값이, 이전 사이클 동안의 제2 코어 주파수 및 제2 메모리 주파수에서 계산된 제2 지시 당 줄 값 보다 높은 경우, 상기 제2 코어 주파수로 상기 복수의 코어를 동작시키고, 상기 제2 메모리 주파수로 상기 복수의 메모리를 동작시키는 멀티 코어 시스템.

청구항 11

응용 프로그램에 포함된 복수의 작업 중 할당된 적어도 하나의 작업을 워프 단위로 처리하는 복수의 코어, 상기 복수의 코어의 입출력 데이터를 저장하는 복수의 메모리, 복수의 작업에 대한 메타 데이터를 저장하는 전력 메모리, 그리고 상기 복수의 코어 및 상기 복수의 메모리를 연결하는 네트워크를 포함하는 멀티 코어 시스템의 구동 방법에 있어서,

상기 전력 메모리에서 상기 할당된 적어도 하나의 작업에 대한 메타 데이터를 검색하는 단계,

상기 응용 프로그램의 사이클 당 지시(IPC, instruction per cycle)를 최대화하는 워프 개수를 결정하는 단계,

상기 복수의 코어가 동작하는 코어 주파수 및 상기 복수의 메모리가 동작하는 메모리 주파수에 따른 성능 데이터를 복수로 수집하는 단계,

상기 성능 데이터를 이용하여 하기의 수학식 7과 같은 성능 추정 모델을 설계하는 단계, 그리고

상기 복수의 코어가 상기 할당된 적어도 하나의 작업을 상기 워프 개수 단위로 처리하도록 제어하는 단계

를 포함하는 멀티 코어 시스템의 구동 방법.

[수학식 7]

$$T = \alpha_C \cdot f_C + \alpha_M \cdot f_M + \beta$$

여기서, T 는 상기 성능 데이터, α_C , α_M 및 β 는 상기 성능 추정 모델의 계수, f_C 는 코어 동작 주파수, f_M 는 메모리 동작 주파수

청구항 12

삭제

청구항 13

제11 항에 있어서,

상기 메타 데이터는 작업 ID, 워프 개수, 소정 코어 주파수, 소정 메모리 주파수, 및 IPC에 대한 정보를 포함하는 멀티 코어 시스템의 구동 방법.

청구항 14

제11 항에 있어서,

상기 워프 개수를 결정하는 단계는, 상기 메타 데이터가 검색되지 않는 경우에 수행되는 멀티 코어 시스템의 구동 방법.

청구항 15

삭제

청구항 16

제11 항에 있어서,

상기 복수의 성능 데이터는 상기 코어 주파수와 상기 메모리 주파수가 모두 최소일 때의 제1 성능 데이터, 상기 코어 주파수가 최소이고 상기 메모리 주파수가 최대일 때의 제2 성능 데이터, 및 상기 코어 주파수가 최대이고, 상기 메모리 주파수가 최소일 때의 제3 성능 데이터를 포함하는 멀티 코어 시스템의 구동 방법.

청구항 17

제11 항에 있어서,

상기 성능 데이터는 하기의 수학식 8로 계산되는 멀티 코어 시스템의 구동 방법.

[수학식 8]

$$T = \frac{I}{t}$$

여기서, I 는 폐기된 지시(instruction)의 개수, t 는 경과 시간

청구항 18

제11 항에 있어서,

상기 성능 추정 모델을 설계하는 단계 이후에,

상기 코어 주파수와 상기 메모리 주파수가 모두 최소일 때의 제1 에너지 소비 데이터, 상기 코어 주파수가 최소이고 상기 메모리 주파수가 최대일 때의 제2 에너지 소비 데이터, 및 상기 코어 주파수가 최대이고, 상기 메모리 주파수가 최소일 때의 제3 에너지 소비 데이터를 수집하는 단계, 그리고

상기 제1 에너지 소비 데이터, 상기 제2 에너지 소비 데이터 및 상기 제3 에너지 소비 데이터를 이용하여, 하기의 수학식 9 내지 11로 복수의 에너지 스택 컴포넌트를 산출하는 단계를 더 포함하는 멀티 코어 시스템의 구동 방법.

[수학식 9]

$$E_C = \frac{E_{LL} - E_{HL} - (1 - \frac{T_{LL}}{T_{HL}})E_S}{k_{fC}}$$

[수학식 10]

$$E_M = \frac{E_{LL} - E_{LH} - (1 - \frac{T_{LL}}{T_{LH}})E_S}{k_{fM}}$$

[수학식 11]

$$E_S = \frac{(r_{fC}^2 r_{fM}^2 - 1)E_{LL} + k_{fC}E_{LH} + k_{fM}E_{HL}}{-1 + r_{fC}^2 r_{fM}^2 + k_{fC} \frac{T_{LL}}{T_{LH}} k_{fM} \frac{T_{LL}}{T_{HL}}}$$

여기서, E_C , E_M 및 E_S 는 상기 복수의 에너지 스택 컴포넌트, E_{LL} 은 상기 제1 에너지 소비 데이터, E_{LH} 은 상

기 제2 에너지 소비 데이터, E_{HL} 은 상기 제3 에너지 소비 데이터, $r_{fC} = \frac{f_{CH}}{f_{CL}}$, $r_{fM} = \frac{f_{MH}}{f_{ML}}$, $k_{fC} = 1 - r_{fC}^2$,

$$k_{fM} = 1 - r_{fM}^2$$

청구항 19

제18 항에 있어서,

상기 복수의 에너지 스택 컴포넌트를 이용하여 하기의 수학적 식 12와 같은 에너지 소비 추정 모델을 설계하는 단계를 더 포함하는 멀티 코어 시스템의 구동 방법.

[수학적 식 12]

$$E_{f_C f_M} = \left(\frac{f_C}{f_{C_L}}\right)^2 E_C + \left(\frac{f_M}{f_{M_L}}\right)^2 E_M + \frac{T_{LL}}{T_{f_C f_M}} E_S$$

청구항 20

제19 항에 있어서,

상기 성능 추정 모델 및 상기 에너지 소비 추정 모델을 이용하여, 현재 사이클 동안 제1 코어 주파수 및 제1 메모리 주파수에서의 제1 지시 당 줄 값(JPI, joule per instruction)을 계산하는 단계, 그리고

상기 제1 지시 당 줄 값이, 이전 사이클 동안의 제2 코어 주파수 및 제2 메모리 주파수에서 계산된 제2 지시 당 줄 값 보다 높은 경우, 상기 제2 코어 주파수로 상기 복수의 코어를 동작시키고, 상기 제2 메모리 주파수로 상기 복수의 메모리를 동작시키는 단계를 더 포함하는 멀티 코어 시스템의 구동 방법.

발명의 설명

기술 분야

[0001] 본 개시는 멀티 코어 시스템 및 그 구동 방법에 관한 것이다.

배경 기술

[0002] 멀티 코어 시스템, 특히 범용 GPU(GPGPU, general purpose computing on graphics processing units)는 기존의 단순한 그래픽 가속장치(accelerator)로서의 역할을 넘어서 범용 CPU와 함께 이기종(heterogeneous) 시스템을 구성하는 스트리밍 멀티프로세서(streaming multiprocessor)로 발전하고 있다.

[0003] 고성능의 범용 GPU의 고 에너지 효율을 달성하기 위해, 범용 GPU의 하드웨어 구성요소의 동적 에너지 소비 및 정적 에너지 소비를 동적으로 밸런싱하여, 전체 에너지 소비를 최소화하는 적응형 아키텍처 기술의 중요성이 증가하고 있다.

[0004] 최근에는, 범용 GPU의 에너지 소비 절감을 위해 범용 GPU의 코어와 메모리 구성요소의 동적 전압 및 주파수 스케일링(DVFS, dynamic voltage and frequency scaling) 및 적응형 동시성 제어와 같은 적응 기술이 연구되고 있다.

발명의 내용

해결하려는 과제

[0005] 실시예는 동적 에너지 및 정적 에너지 소비를 효과적으로 밸런싱하는 멀티 코어 시스템 및 그 구동 방법을 제공하고자 한다.

과제의 해결 수단

[0006] 실시예에 따른 멀티 코어 시스템은 응용 프로그램에 포함된 복수의 작업 중 할당된 적어도 하나의 작업을 워프 단위로 처리하는 복수의 코어, 복수의 코어의 입출력 데이터를 저장하는 복수의 메모리, 그리고 응용 프로그램의 사이클 당 지시(IPC, instruction per cycle)를 최대화하는 워프 개수를 결정하고, 복수의 코어가 할당된 적어도 하나의 작업을 워프 개수 단위로 처리하도록 제어하는 전력 관리부를 포함한다.

[0007] 전력 관리부는, 복수의 작업에 대한 메타 데이터를 저장하는 전력 메모리, 그리고 전력 메모리에서 할당된 적어도 하나의 작업에 대한 메타 데이터를 검색하는 전력 제어부를 포함할 수 있다.

[0008] 메타 데이터는, 작업 ID, 워프 개수, 소정 코어 주파수, 소정 메모리 주파수, 및 IPC에 대한 정보를 포함할 수 있다.

- [0009] 전력 제어부는 메타 데이터가 검색되지 않는 경우, 워프 개수를 결정할 수 있다.
- [0010] 전력 제어부는 복수의 코어 및 메모리 주파수에 따른 성능 데이터를 복수로 수집하고, 성능 데이터를 이용하여 성능 추정 모델을 설계할 수 있다.
- [0011] 복수의 성능 데이터는 복수의 코어가 동작하는 코어 주파수와 복수의 메모리가 동작하는 메모리 주파수가 모두 최소일 때의 제1 성능 데이터, 코어 주파수가 최소이고 메모리 주파수가 최대일 때의 제2 성능 데이터, 및 코어 주파수가 최대이고, 메모리 주파수가 최소일 때의 제3 성능 데이터를 포함할 수 있다.
- [0012] 성능 데이터는 폐기된 지시(instruction)의 개수 및 경과 시간을 이용하여 계산될 수 있다.
- [0013] 전력 제어부는 복수의 코어가 동작하는 코어 주파수와 복수의 메모리가 동작하는 메모리 주파수가 모두 최소일 때의 제1 에너지 소비 데이터, 코어 주파수가 최소이고 메모리 주파수가 최대일 때의 제2 에너지 소비 데이터, 및 코어 주파수가 최대이고, 메모리 주파수가 최소일 때의 제3 에너지 소비 데이터를 수집하고, 제1 에너지 소비 데이터, 제2 에너지 소비 데이터 및 제3 에너지 소비 데이터를 이용하여 복수의 에너지 스택 컴포넌트를 산출할 수 있다.
- [0014] 전력 제어부는 복수의 에너지 스택 컴포넌트를 이용하여 에너지 소비 추정 모델을 설계할 수 있다.
- [0015] 전력 제어부는 성능 추정 모델 및 에너지 소비 추정 모델을 이용하여, 현재 사이클 동안 제1 코어 주파수 및 제1 메모리 주파수에서의 제1 지시 당 줄 값(JPI, joule per instruction)을 계산하고, 제1 지시 당 줄 값이, 이전 사이클 동안의 제2 코어 주파수 및 제2 메모리 주파수에서 계산된 제2 지시 당 줄 값 보다 높은 경우, 제2 코어 주파수로 복수의 코어를 동작시키고, 제2 메모리 주파수로 복수의 메모리를 동작시킬 수 있다.
- [0016] 실시예에 따른 멀티 코어 시스템의 구동 방법은, 응용 프로그램에 포함된 복수의 작업 중 할당된 적어도 하나의 작업을 워프 단위로 처리하는 복수의 코어, 복수의 코어의 입출력 데이터를 저장하는 복수의 메모리, 그리고 복수의 코어 및 복수의 메모리를 연결하는 네트워크를 포함하는 멀티 코어 시스템의 구동 방법에 있어서, 응용 프로그램의 사이클 당 지시(IPC, instruction per cycle)를 최대화하는 워프 개수를 결정하는 단계, 그리고 복수의 코어가 할당된 적어도 하나의 작업을 워프 개수 단위로 처리하도록 제어하는 단계를 포함한다.
- [0017] 멀티 코어 시스템은 복수의 작업에 대한 메타 데이터를 저장하는 전력 메모리를 더 포함하고, 전력 메모리에서 할당된 적어도 하나의 작업에 대한 메타 데이터를 검색하는 단계를 더 포함할 수 있다.
- [0018] 메타 데이터는 작업 ID, 워프 개수, 소정 코어 주파수, 소정 메모리 주파수, 및 IPC에 대한 정보를 포함할 수 있다.
- [0019] 워프 개수를 결정하는 단계는, 메타 데이터가 검색되지 않는 경우에 수행될 수 있다.
- [0020] 복수의 코어 및 메모리 주파수에 따른 성능 데이터를 복수로 수집하는 단계, 그리고 성능 데이터를 이용하여 성능 추정 모델을 설계하는 단계를 더 포함할 수 있다.
- [0021] 복수의 성능 데이터는 복수의 코어가 동작하는 코어 주파수와 복수의 메모리가 동작하는 메모리 주파수가 모두 최소일 때의 제1 성능 데이터, 코어 주파수가 최소이고 메모리 주파수가 최대일 때의 제2 성능 데이터, 및 코어 주파수가 최대이고, 메모리 주파수가 최소일 때의 제3 성능 데이터를 포함할 수 있다.
- [0022] 성능 데이터는 폐기된 지시의 개수 및 경과 시간을 이용하여 계산될 수 있다.
- [0023] 복수의 코어가 동작하는 코어 주파수와 복수의 메모리가 동작하는 메모리 주파수가 모두 최소일 때의 제1 에너지 소비 데이터, 코어 주파수가 최소이고 메모리 주파수가 최대일 때의 제2 에너지 소비 데이터, 및 코어 주파수가 최대이고, 메모리 주파수가 최소일 때의 제3 에너지 소비 데이터를 수집하는 단계, 그리고 제1 에너지 소비 데이터, 제2 에너지 소비 데이터 및 제3 에너지 소비 데이터를 이용하여 복수의 에너지 스택 컴포넌트를 산출하는 단계를 더 포함할 수 있다.
- [0024] 복수의 에너지 스택 컴포넌트를 이용하여 에너지 소비 추정 모델을 설계하는 단계를 더 포함할 수 있다.
- [0025] 성능 추정 모델 및 에너지 소비 추정 모델을 이용하여, 현재 사이클 동안 제1 코어 주파수 및 제1 메모리 주파수에서의 제1 지시 당 줄 값(JPI, joule per instruction)을 계산하는 단계, 그리고 제1 지시 당 줄 값이, 이전 사이클 동안의 제2 코어 주파수 및 제2 메모리 주파수에서 계산된 제2 지시 당 줄 값 보다 높은 경우, 제2 코어 주파수로 복수의 코어를 동작시키고, 제2 메모리 주파수로 복수의 메모리를 동작시키는 단계를 더 포함할 수 있다.

발명의 효과

- [0026] 실시예에 따른 멀티 코어 시스템 및 그 구동 방법의 효과에 대해 설명하면 다음과 같다.
- [0027] 실시예들 중 적어도 하나에 의하면, 에너지-딜레이 프로덕트(energy-delay product) 및 에너지-딜레이 스퀘어드 프로덕트(energy-delay-squared product)를 감소시킬 수 있다.
- [0028] 실시예들 중 적어도 하나에 의하면, 높은 에너지 효율 이득을 달성할 수 있다.

도면의 간단한 설명

- [0029] 도 1은 실시예의 멀티 코어 시스템을 나타낸 블록도(block diagram)이다.
- 도 2는 도 1의 멀티 코어 시스템의 코어 및 메모리를 구체적으로 나타낸 블록도이다.
- 도 3은 실시예에 따른 멀티 코어 시스템의 구동 방법을 나타낸 순서도이다.

발명을 실시하기 위한 구체적인 내용

- [0030] 그러면 첨부한 도면을 참고로 하여 실시예에 대하여 실시예가 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 실시예는 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다.
- [0031] 또한, 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함" 한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0032] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.
- [0033] 비록 제1, 제2 등과 같은 서수가 구성요소들을 서술하기 위해서 사용되나, 이들 구성요소들은 서수에 의해 제한되지 않는다. 이들 용어들은 동일한 용어로 정의되는 구성요소들을 구별하기 위하여 사용하는 것이다. 따라서, 발명을 실시하기 위한 구체적인 내용에서 언급되는 제1 구성요소는 특허청구범위에서 다른 서수로 구별될 수 있다.
- [0034] 먼저, 도 1 및 도 2를 참조하여, 멀티 코어 시스템에 대하여 설명한다.
- [0035] 도 1은 실시예의 멀티 코어 시스템을 나타낸 블록도이고, 도 2는 도 1의 멀티 코어 시스템의 코어 및 메모리를 구체적으로 나타낸 블록도이다.
- [0036] 도 1에 도시된 바와 같이, 멀티 코어 시스템은, 멀티 코어 GPU(10)와 외부 메모리 그룹(20)을 포함할 수 있다. 멀티 코어 GPU(10)는 작업 스케줄러(110), 전력 관리부(120), 복수의 코어(C1-Cn), 복수의 메모리(M11-M1m) 및 이들을 상호 연결하는 내부 연결 네트워크(14)를 포함할 수 있다.
- [0037] 작업 스케줄러(110)(job scheduler)는 멀티 코어(C1-Cn)를 이루는 각 코어(C1-Cn)의 자원(resource) 상태에 따라 작업(job)을 각 코어(C1-Cn)에 분배한다.
- [0038] 멀티 코어 시스템에서 응용 프로그램을 수행할 때, 하나의 응용 프로그램(workload)은 복수의 커널(kernel)을 포함할 수 있다. 하나의 커널은 여러 개의 스레드 묶음(CTAs)으로 구성된다. 하나의 스레드 묶음(CTA, cooperative thread array)은 다수의 워프들(warps)을 포함하고, 워프(warp)는 복수의 스레드를 포함할 수 있다.
- [0039] 작업 스케줄러(110)는 코어(C1-Cn)의 자원 상태를 고려하여, 각 코어(C1-Cn)가 처리할 수 있는 개수의 스레드 묶음을 할당한다.
- [0040] 각각의 코어(C1-Cn)는 할당된 하나의 스레드 묶음 또는 다수의 스레드 묶음을 처리한다. 이때, 코어(C1-Cn) 각각은 한 사이클(1 cycle)에 동시에 처리할 수 있는 단위로 나누어 관리하기 위하여 레지스터 파일 및 소규모의 캐시 또는 온-칩 메모리 요소를 포함할 수 있다.
- [0041] 복수의 메모리(M11-M1m)는 코어(C1-Cn)와 작업 스케줄러(110)에서 필요한 각종 입출력 데이터를 저장한다. 그리

고, 메모리들(M11-M1m)은 모든 코어(C1-Cn)에 의해 공유될 수 있다.

- [0042] 내부 연결 네트워크(14)는 작업 스케줄러(110), 코어들(C1-Cn) 및 메모리들(M11-M1m)을 서로 연결하며, 데이터와 제어 신호의 전달 경로가 된다.
- [0043] 다음으로, 외부 메모리 그룹(20)은 데이터를 임시로 저장하는 버퍼 역할을 수행할 수 있다. 멀티 코어 GPU(10) 내의 복수의 메모리들(M11-M1m)은 복수의 외부 메모리들(M21-M2m)을 액세스할 수 있다. 각각의 외부 메모리들(M21-M2m)은 다이내믹 랜덤 액세스 메모리(DRAM) 또는 스테틱 랜덤 액세스 메모리(SRAM)로 구현될 수 있으며 이에 한정되지 않는다.
- [0044] 도 2를 함께 참조하면, 하나의 코어(Ci)는 워프 스케줄러(210), 산술 논리 연산부(220)(arithmetic and logic unit, 이하 ALU)들, 메모리 액세스부(230)(load/store (LDST) unit)들, 및 L1 캐시(240)를 포함할 수 있다. 그리고, 하나의 메모리(M1j)는 L2 캐시(250)와 메모리 제어부(260)를 포함할 수 있다. 코어(Ci)와 메모리(M1j)는 모두 내부 연결 네트워크(14)로 연결될 수 있다.
- [0045] 워프 스케줄러(210)는 할당된 적어도 하나의 스레드 묶음 내의 워프들 중 ALU들(220)에 의해 실행될 워프를 결정할 수 있다. 워프 스케줄러(210)는 활성 워프 개수(active warp count)에 따라, ALU들(220)에 의해 동시에 처리될 워프의 개수를 결정할 수 있다.
- [0046] 워프 스케줄링은, 각 워프에 의해 실행되고 있는 커널로부터의 특정한 명령들이 각각의 코어(Ci) 내의 ALU들(220) 중 하나에 디스패치(dispatch)되게 한다.
- [0047] ALU는 디스패치된 워프 내의 스레드를 동시에 처리할 수 있다.
- [0048] 메모리 액세스부(230)는 L1 캐시(240) 또는 L2 캐시(250)로부터 데이터를 읽어 오거나, L1 캐시(240) 또는 L2 캐시(250)에 데이터를 기록할 수 있다.
- [0049] L1 캐시(240)는 코어(Ci) 내에서 처리되는 작업, 명령어 및 데이터 등을 임시 저장할 수 있다. L1 캐시(240)는 접근 속도가 빠를 뿐만 아니라, 코어(Ci)마다 독립적으로 처리될 수 있으므로 시스템의 성능 향상에 기여한다.
- [0050] 메모리 제어부(260)는 L2 캐시(250)로부터 출력되는 데이터를 해석하고, L2 캐시(250)로 입력되는 데이터를 저장 및 삭제하는 등, L2 캐시(250)의 데이터를 관리하고 제어할 수 있다.
- [0051] 다시 도 1을 참조하여 설명하면, 실시예의 멀티 코어 시스템(10, 20)은, 두 개의 전압 및 주파수(VF) 도메인, 즉 코어 VF 도메인 및 메모리 VF 도메인에 기초하여 동작하는 것으로 가정한다. 코어(C1-Cn)들은 코어 VF 도메인에 기초하여 동작하고, 내부 연결 네트워크(14), 메모리들(M11-M1m) 및 외부 메모리들(M21-M2m)은 메모리 VF 도메인에 기초하여 동작하는 것으로 가정한다.
- [0052] 전력 관리부(120)는 전력 메모리(122)와 전력 제어부(124)를 포함할 수 있다.
- [0053] 전력 메모리(122)는 소규모의 캐시와 커널 정보 테이블(kernel information table, 이하, KIT)을 포함할 수 있다. KIT는 커널 ID, 최적 워프 개수, 최적 코어 주파수, 최적 메모리 주파수, 및 커널의 이전 실행으로부터 저장된 사이클 당 지시들(instruction per cycle, 이하, IPC)과 같은 커널의 메타데이터를 관리할 수 있다.
- [0054] 예를 들어, 전력 메모리(122)에는 8 개의 KIT가 저장될 수 있고, 각각의 KIT는 4 비트의 커널 ID, 5 비트의 최적 워프 개수 정보, 2 비트의 코어 주파수 정보, 2 비트의 메모리 주파수 정보, 그리고 16 비트 IPC 정보를 포함하여, 총 29비트로 구성될 수 있다. 이러한 KIT는 LRU 알고리즘(Least-recently-used)에 따라, 업데이트될 수 있다.
- [0055] 전력 제어부(124)는 복수의 에너지 스택에 근거하여 에너지 추정 모델을 동적으로 생성하고, 에너지 추정 모델을 이용하여 최적 시스템 상태를 탐색할 수 있다.
- [0056] 복수의 에너지 스택은 멀티 코어 시스템에서의 주요 에너지 소비 컴포넌트를 구성하며, 코어 다이내믹(core dynamic), 메모리 다이내믹(memory dynamic), 및 정적 에너지 소비(static energy consumption)로 구분될 수 있다.
- [0057] 예를 들어, 코어 다이내믹은 코어 VF 도메인에 기초하여 동작하는 구성 요소들에 의해 소비되는 에너지 소비 컴포넌트이고, 메모리 다이내믹은 메모리 VF 도메인에 기초하여 동작하는 구성요소들에 의해 소비되는 에너지 소비 컴포넌트이다. 그리고, 정적 에너지 소비는 유휴 상태(idle state)에서 멀티 코어 시스템 내에서 소비되는 에너지 소비 컴포넌트이다.

- [0058] 전력 관리부(120)는 전력 메모리(122)의 KIT로부터, 커널을 처리하기 위한 최적 워프 개수, 최적 코어 주파수 및 최적 메모리 주파수로 멀티 코어 시스템이 구동하도록 제어할 수 있다.
- [0059] 또는, 전력 관리부(120)는 KIT에 현재 실행되는 커널의 메타 데이터가 저장되어 있지 않은 경우, 적응형 워프 제한(AWL, adaptive warp limiting) 및 동적 전압 및 주파수 스케일링(DVFS, dynamic voltage and frequency scaling)을 통해, 최적 워프 개수, 최적 코어 주파수 및 최적 메모리 주파수를 탐색할 수 있다.
- [0060] 이러한 전력 관리부(120)의 구체적인 동작에 대해서 도 3을 참조하여 함께 설명한다.
- [0061] 도 3은 실시예에 따른 멀티 코어 시스템의 구동 방법을 나타낸 순서도이다.
- [0062] 먼저, 전력 제어부(124)는 코어 주파수 및 메모리 주파수들을 최대 값으로 설정하는 초기 상태로 멀티 코어 시스템의 상태를 설정(S10)한다.
- [0063] 이러한 워 업 단계(S10) 동안, 전력 제어부(124)는 어떠한 모니터링 또는 적응 활동도 수행하지 않고, 한 사이클 동안 대상 응용 프로그램을 실행한다. 워 업 단계의 목적은 충분히 멀티 코어 시스템을 워 업시킨 후에 수집된 정상 상태 런타임 데이터에 근거하여, 후속하는 적응 활동이 수행되는 것을 보장하기 위함이다.
- [0064] 워 업 단계(S10) 후, 전력 제어부(124)는 KIT를 검색(S12)한다. KIT 내에 대상 응용 프로그램에 포함된 커널들에 관련된 커널 정보가 저장되어 있지 않은 경우, 전력 제어부(124)는 최적 워프 개수를 결정(S16)한다.
- [0065] 전력 제어부(124)는 IPC를 최대화하는 최적 워프 개수를 동적으로 결정할 수 있다. 최적 워프 개수는 순차적 방식 또는 병렬적 방식으로 결정될 수 있다.
- [0066] 순차적인 방식에 따르면, 각각의 코어들(C1-Cn)은 주기적으로 응용 프로그램의 IPC를 샘플링하고, 전력 제어부(124)는 샘플링된 IPC의 변화 추세에 근거하여, 활성 워프 개수를 결정할 수 있다.
- [0067] 예를 들어, 코어들(C1-Cn) 각각은 최대 개수의 활성 워프 개수로부터 시작하여, 현재 주기 동안 샘플링된 IPC가 이전 주기 동안 샘플링된 IPC 동안 더 적어질 때까지, 활성 워프 개수를 반으로 감소시키는 이진 검색(binary search)을 수행한다. 그리고, 코어들(C1-Cn) 각각은 최적 활성 워프 개수를 결정하기 위해, 이진 검색된 활성 워프 개수에서부터 점차 그 개수를 증가시키는 선형 검색(linear search)을 수행한다.
- [0068] 병렬적인 방식에 따르면, 코어들(C1-Cn)은 소정 사이클 동안 최대 및 최소 워프 개수 범위 내에서, 서로 다른 활성 워프 개수로 응용 프로그램을 각각 실행할 수 있다. 병렬적인 방식의 경우, 최대 워프 개수가 코어들(C1-Cn)의 개수보다 더 크면, 최적 워프 개수를 결정하기 위해 다수의 주기들이 요구된다.
- [0069] 따라서, 순차적인 방식에 비해 병렬적인 방식은 활성 워프 개수에 대한 병렬적 검색에 의해, 최적 워프 개수를 더 빠르게 결정할 수 있다는 것이다. 그러나, 병렬적인 방식의 주된 단점은, 다른 코어들에서 실행되는 워프들이 광범위한 성능 특성들을 나타내면, 최적 워프 개수가 아닌, 차선의 워프 개수로 활성 워프 개수가 결정될 수 있다는 것이다.
- [0070] 다음으로, 최적의 활성 워프 개수가 검색되면, 전력 제어부(124)는 성능 데이터 샘플을 수집(S18)한다. 전력 제어부(124)는 소정 사이클 동안, 대상 응용 프로그램을 세가지 조건에서 실행함으로써, 세가지 성능 데이터 샘플들을 수집할 수 있다.
- [0071] 성능 데이터는 초당 폐기된 지시의 개수를 나타내는 스루풋(T)으로 결정될 수 있다. 스루풋은 하기의 수학적 식 1로 계산될 수 있다.

수학적 식 1

$$T = \frac{I}{t}$$

[0072] 여기서, I 는 폐기된 인스트럭션의 개수이며, t 는 경과 시간이다.

[0073] 세가지 성능 데이터 샘플은 코어 주파수와 메모리 주파수가 모두 최소일 때의 스루풋(T_{LL} , 코어 주파수가 최소

이고 메모리 주파수가 최대일 때의 스루풋(T_{LH}), 및 코어 주파수가 최대이고, 메모리 주파수가 최소일 때의 스루풋(T_{HL})이다.

[0075] 이때, 전력 제어부(124)는, 세가지 조건에서 대상 응용 프로그램을 실행할 때의 멀티 코어 시스템의 전체 에너지 소비량(E_{LL} , E_{LH} , 및 E_{HL})을 수집한다.

[0076] 다음으로, 전력 제어부(124)는 수집한 성능 데이터 샘플을 이용하여, 에너지 스택 컴포넌트를 계산(S20)한다.

[0077] 전력 제어부(124)는 단순 리니어 모델을 이용하여 시스템 상태의 성능을 추정할 수 있다. 성능 추정을 위한 단순 리니어 모델은 하기의 수학식 2와 같이 나타낼 수 있다.

수학식 2

$$T = \alpha_C \cdot f_C + \alpha_M \cdot f_M + \beta$$

[0079] 여기서, α_C , α_M 및 β 는 성능 추정 모델의 계수이고, f_C 는 코어 주파수며, f_M 는 메모리 주파수다.

[0080] 전력 제어부(124)는 스루풋들(T_{LL} , T_{LH} , T_{HL})과 코어 주파수 및 메모리 주파수 값들을 상기의 수학식 2에 대입하여, 성능 추정 모델의 세 계수를 계산할 수 있다.

[0081] 그러면, 전력 제어부(124)는 세가지 조건에서의 에너지 소비량(E_{LL} , E_{LH} , 및 E_{HL})과 하기의 수학식 3 내지 5를 이용하여, 세가지 에너지 스택 컴포넌트(E_C , E_M 및 E_S)를 산출할 수 있다.

수학식 3

$$E_C = \frac{E_{LL} - E_{HL} - (1 - \frac{T_{LL}}{T_{HL}})E_S}{k_{fC}}$$

수학식 4

$$E_M = \frac{E_{LL} - E_{LH} - (1 - \frac{T_{LL}}{T_{LH}})E_S}{k_{fM}}$$

수학식 5

$$E_S = \frac{(r_{fC}^2 r_{fM}^2 - 1)E_{LL} + k_{fC}E_{LH} + k_{fM}E_{HL}}{-1 + r_{fC}^2 r_{fM}^2 + k_{fC} \frac{T_{LL}}{T_{LH}} k_{fM} \frac{T_{LL}}{T_{HL}}}$$

[0085] 여기서, $r_{fC} = \frac{f_{CH}}{f_{CL}}$, $r_{fM} = \frac{f_{MH}}{f_{ML}}$, $k_{fC} = 1 - r_{fC}^2$ 및 $k_{fM} = 1 - r_{fM}^2$ 이다.

[0086] 다음으로, 전력 제어부(124)는 에너지 스택 컴포넌트를 이용하여, 최적의 코어 주파수 및 메모리 주파수를 탐색(S22)한다. 전력 제어부(124)는 매 사이클에서, 대상 응용 프로그램의 성능 및 에너지 소비를 주기적으로 샘플링한다.

[0087] 구체적으로, 전력 제어부(124)는 상기의 수학식 2의 성능 추정 모델을 이용하여, 임의의 코어 주파수 및 메모리

주파수에서의 성능 데이터를 샘플링할 수 있다.

[0088] 또한, 전력 제어부(124)는 하기의 수학적 식 6을 이용하여, 임의의 코어 주파수 및 메모리 주파수에서의 에너지 소비 데이터를 샘플링할 수 있다.

수학적 식 6

$$E_{f_C f_M} = \left(\frac{f_C}{f_{CL}}\right)^2 E_C + \left(\frac{f_M}{f_{ML}}\right)^2 E_M + \frac{T_{LL}}{T_{f_C f_M}} E_S$$

[0090] 그러면, 전력 제어부(124)는 성능 데이터 및 에너지 소비 데이터로부터 현재 기간의 한 개의 지시당 소비된 에너지, 즉 지시 당 줄(joule per instruction, 이하, JPI)을 계산하고, 이를 이전 기간의 JPI와 비교한다.

[0091] 현재 기간의 JPI가 이전 기간의 JPI보다 높은 경우, 전력 제어부(124)는 시스템 상태를 이전 시스템 상태로 되돌리는 폴백을 수행하고, 탐색 단계를 종료하며, 유휴 단계로 진입(S24)한다.

[0092] 예를 들어, 전력 제어부(124)는 수학적 식 2의 성능 추정 모델 및 수학적 식 6의 에너지 소비 추정 모델을 이용하여, 현재 사이클 동안 제1 코어 주파수 및 제1 메모리 주파수에서의 제1 지시 당 줄 값을 계산하고, 제1 지시 당 줄 값이 이전 사이클 동안의 제2 코어 주파수 및 제2 메모리 주파수에서 계산된 제2 지시 당 줄 값 보다 높은 경우, 제2 코어 주파수 및 제2 메모리 주파수를 최적의 코어 주파수 및 메모리 주파수로 결정한다.

[0093] 현재 기간의 JPI가 이전 기간의 JPI보다 낮은 경우, 전력 제어부(124)는 최적의 코어 주파수 및 메모리 주파수를 계속 탐색한다. 전력 제어부(124)는 incremental and greedy 방식으로 코어 주파수 및 메모리 주파수를 탐색할 수 있다. incremental and greedy 방식은 코어 주파수 및 메모리 주파수를 현재의 코어 주파수 및 메모리 주파수 값으로부터 점진적으로 변경함으로써, 최적 코어 주파수 및 메모리 주파수를 탐색할 수 있다.

[0094] 예를 들어, 현재 시스템 상태는 (f_{C_i}, f_{M_j}) 이고, 가장 최근 탐색되었던 이전 시스템 상태는 $(f_{C_i}, f_{M_{j+1}})$ 인 경우, 후보 시스템 상태는 $(f_{C_{i-1}}, f_{M_j})$, $(f_{C_{i+1}}, f_{M_j})$ 및 $(f_{C_i}, f_{M_{j-1}})$ 이다. 이때 주파수 값들은 첨자들의 순서대로 정렬된다.

[0095] 한편, S12단계에서, KIT 내에 대상 응용 프로그램에 포함된 커널들에 관련된 커널 정보가 저장되어 있는 경우, 전력 제어부(124)는 커널 정보를 이용하여 커널을 처리하기 위한 최적 워프 개수, 최적 코어 주파수 및 최적 메모리 주파수로 멀티 코어 시스템이 구동하도록 제어(S14)하고, 유휴 상태로 진입(S24)한다.

[0096] 유휴 단계(S24) 이후, 전력 제어부(124)는 대상 응용 프로그램의 IPC를 주기적으로 샘플링한다. 두 개의 연속한 IPC 샘플들 간의 차이(ΔIPC)가 임계값을 초과하는 경우, 전력 제어부(124)는 대상 응용 프로그램 변경을 감지하고, 유휴 단계(S24)를 중단하며, S12 단계로 진입한다.

[0097] 실시예에 따른 멀티 코어 시스템 및 그 구동 방법은, 멀티 코어 시스템의 주요 에너지 소모 컴포넌트(즉, 코어 다이내믹, 메모리 다이내믹 및 정적 에너지 소비)를 산출하고, 에너지 추정 모델을 동적으로 생성하며, 에너지 효율의 관점에서 최적의 시스템 상태를 찾기 위해 에너지 추정 모델에 근거하여 코어 주파수 및 메모리 주파수를 탐색한다.

[0098] 실시예에 따른 멀티 코어 시스템 및 그 구동 방법은 최대 코어 주파수 및 최대 메모리 주파수에서 동작하는 기존의 멀티 코어 시스템 보다 확연히 높은 에너지 효율을 달성할 수 있다. 실시예에 따른 멀티 코어 시스템 및 그 구동 방법은 에너지 소모 컴포넌트들의 균형을 유지하여 높은 에너지 효율을 달성할 수 있다.

[0099] 또한, 실시예에 따른 멀티 코어 시스템 및 그 구동 방법은 워프들의 단위를 제어하여, 최적의 워프 개수에서 대상 응용 프로그램을 실행함으로써 더 높은 에너지 효율을 달성할 수 있다.

[0100] 이상에서 실시예의 바람직한 실시예에 대하여 상세하게 설명하였지만 실시예의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 실시예의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 실시예의 권리범위에 속하는 것이다.

부호의 설명

- [0101]
- 10: 멀티 코어 GPU

20: 외부 메모리 그룹

110: 작업 스케줄러

120: 전력 관리부

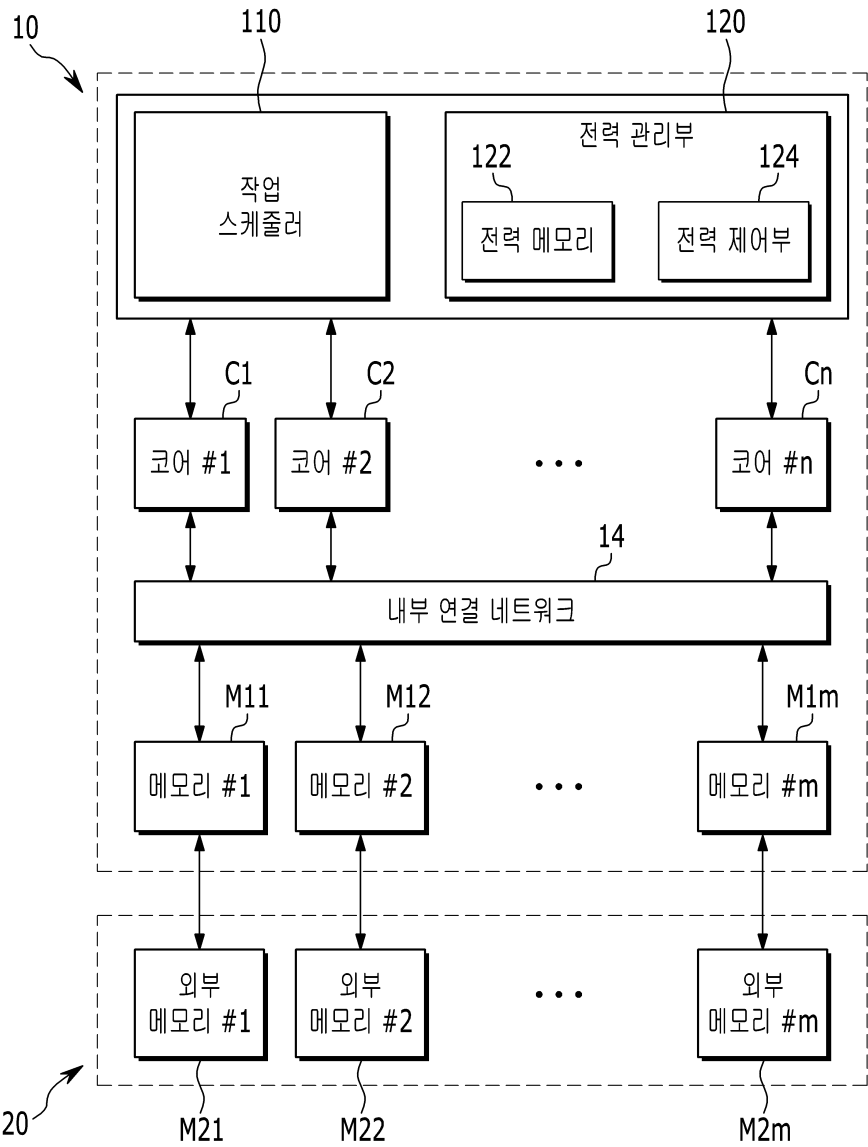
C1-Cn: 코어들

M11-M1m: 메모리들

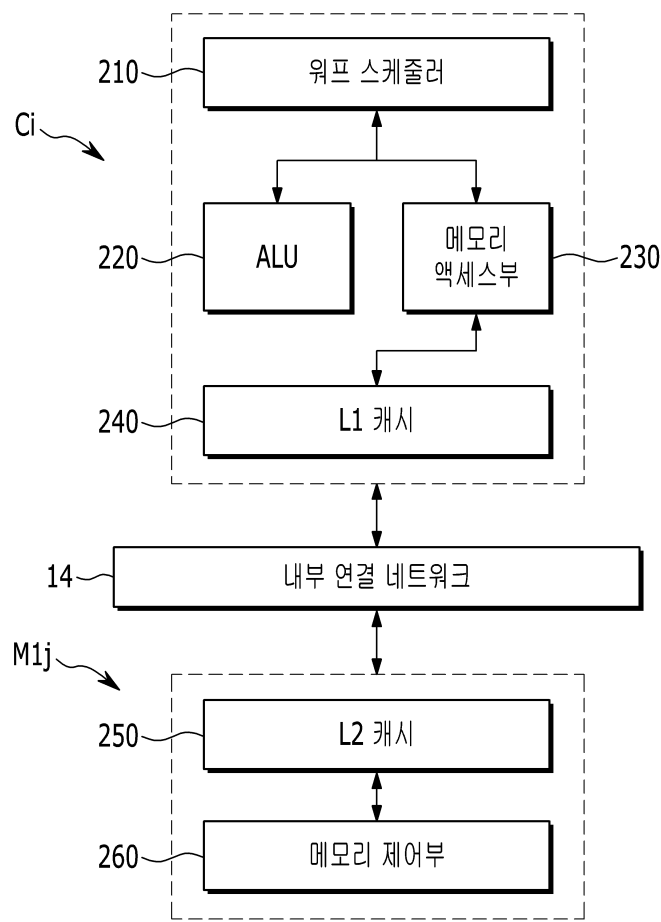
M21-M2m: 외부 메모리들

도면

도면1



도면2



도면3

