



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년07월09일
(11) 등록번호 10-1416766
(24) 등록일자 2014년07월02일

(51) 국제특허분류(Int. Cl.)
G06F 9/30 (2006.01)
(21) 출원번호 10-2013-0042986
(22) 출원일자 2013년04월18일
심사청구일자 2013년04월18일
(56) 선행기술조사문헌
KR1020010063191 A

(73) 특허권자
서울과학기술대학교 산학협력단
서울특별시 노원구 공릉로 232 (공릉동, 서울과학기술대학교)
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
주식회사 에이디칩스
경기도 안양시 동안구 학의로 282, 에이동 22층(관양동, 금강펜테리움아이티타워)
(72) 발명자
이승은
서울 강서구 허준로 23, 106동 601호 (가양동, 한강타운아파트)
정영섭
강원도 춘천시 우석로 15
(뒷면에 계속)
(74) 대리인
김정현

전체 청구항 수 : 총 3 항

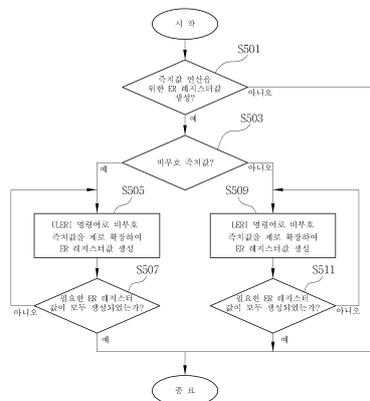
심사관 : 지정훈

(54) 발명의 명칭 EISC 프로세서에서의 즉시 값 연산 방법

(57) 요약

본 발명의 EISC(Extendable Instruction Set Computer) 프로세서에서의 즉시 값(Immediate value) 연산 방법에 있어서, 즉시 값(Immediate value) 연산을 위한 ER(Extension Register) 레지스터 값 생성 시, 비부호 즉시 값(Untyped Immediate value)을 사용하는지 여부를 확인하는 단계 및 비부호 즉시 값을 사용하는 경우, ULERI(Untyped Load Extension Register with Immediate) 명령어를 이용하여 상기 비부호 즉시 값을 제로 확장(Zero Extension)하는 방식으로 상기 ER 레지스터 값을 생성하는 단계를 포함한다. 본 발명에 의하면 EISC 프로세서에서의 16비트 명령어를 이용한 즉시 값 연산시에 불필요한 LERI 명령어의 수행을 방지함으로써, 연산 효율을 향상시킬 수 있는 효과가 있다.

대표도 - 도5



(72) 발명자

김상돈

강원 원주시 홍양로51번길 22, 101동 105호 (태장동, 태장주공1단지아파트)

서태원

서울 노원구 공릉로 126, 107동 502호 (공릉동, 두산힐스빌아파트)

김한이

서울 관악구 성현로 80, 127동 101호 (봉천동, 관악드림타운)

차영호

경기 안양시 동안구 학의로 146, 203동 1204호 (관양동, 한가람삼성아파트)

김관영

경기 성남시 분당구 정자일로213번길 5, 301동 1801호 (정자동, 아이파크분당)

이 발명을 지원한 국가연구개발사업

과제고유번호 10041664

부처명 지식경제부

연구사업명 산업융합원천기술개발사업

연구과제명 멀티 Shader GPU 통합형 멀티 코어 퓨전 프로세서 원천 기술 개발(The Development of Fusion Processor based on Multi-Shader GPU)

기여율 1/1

주관기관 전자부품연구원

연구기간 2012.06.01 ~ 2016.05.31

특허청구의 범위

청구항 1

EISC(Extendable Instruction Set Computer) 프로세서에서의 즉시 값(Immediate value) 연산 방법에 있어서, 즉시 값(Immediate value) 연산을 위한 ER(Extention Register) 레지스터 값 생성 시, 비부호 즉시 값(Unsigned Immediate value)을 사용하는지 여부를 확인하는 단계; 및

비부호 즉시 값을 사용하는 경우, ULERI(Unsigned Load Extension Register with Immediate) 명령어를 이용하여 상기 비부호 즉시 값을 제로 확장(Zero Extension)하는 방식으로 상기 ER 레지스터 값을 생성하는 단계를 포함하는 EISC 프로세서에서의 즉시 값 연산 방법.

청구항 2

제1항에 있어서,

즉시 값 연산을 위한 ER 레지스터 값 생성 시, 부호 즉시 값(Signed Immediate value)을 사용하는 경우, LERI(Load Extension Register with Immediate) 명령어를 이용하여 상기 즉시 값을 부호 확장(Sign Extension)하는 방식으로 상기 ER 레지스터 값을 생성하는 단계를 더 포함하는 것을 특징으로 하는 EISC 프로세서에서의 즉시 값 연산 방법.

청구항 3

제2항에 있어서,

상기 EISC 프로세서는 32 비트(bit) 프로세서로서 32 비트(bit)의 즉시 값 연산을 수행하고,

상기 ULERI 명령어 및 상기 LERI 명령어의 길이는 16 비트(bit)인 것임을 특징으로 하는 EISC 프로세서에서의 즉시 값 연산 방법.

명세서

기술분야

[0001] 본 발명은 16 비트 인스트럭션(Instruction)을 사용하는 32비트 프로세서인 EISC(Extendable Instruction Set Computer)에 관한 것으로서, 더욱 상세하게는 EISC에서 즉시 값(Immediate value) 연산을 위한 ER 레지스터(Extention Register) 값을 효율적으로 생성하는 방법에 관한 것이다.

배경기술

[0002] 종래의 CISC(Complex Instruction Set Computer) 기반의 마이크로 프로세서는 코드 밀도에 있어서 장점이 있으나 고속화에 문제점을 지니고 있으며, RISC(Reduce Instruction Set Computer) 기반의 마이크로 프로세서는 대부분 32-비트(bit) 크기의 명령어를 지님으로 프로그램 코드의 크기가 지나치게 커진다는 단점을 지니고 있다.

[0003] EISC(Extendable Instruction Set Computer)는 수행 프로그램의 크기 및 메모리 접근의 회수를 매우 효과적으로 줄일 수 있도록 설계된 명령어 셋으로서, 확장 명령어를 이용하여 명령어의 즉시 값 및 변위 등의 값을 자유롭게 확장할 수 있는 형태를 지닌 아키텍처로서 종래의 CISC와 RISC의 장점을 합친 형태를 취하고 있다.

[0004] EISC는 기본적으로 RISC가 갖고 있는 간단한 구조의 하드웨어를 취하면서도 CISC의 장점을 추가하여 높은 성능을 갖게 하였고, 코드 밀도가 높아 종래의 RISC 프로세서와 비교하여 약 60%, CISC 프로세서와 비교하여 약 80% 정도로 프로그램 크기가 작다는 장점을 지니고 있다. 따라서, 코드 밀도가 중시되는 내장형 응용 분야에 있어서

장점을 지니고 있다.

- [0005] EISC는 명령어 확장기능이 있는 컴퓨터 아키텍처이다. 즉, 보통의 프로세서는 명령어 오퍼랜드의 크기가 한정되어 있는 반면에 EISC 아키텍처를 채용하고 있는 프로세서는 내부에 ER(Extention Register, 확장 레지스터)와 확장 플래그(e_flag)를 갖추고 있어 명령어 오퍼랜드(operand)의 크기를 확장 레지스터의 크기만큼 확장할 수 있다.
- [0006] 예를 들어, 명령어 오퍼랜드의 크기가 16비트이고 확장 레지스터의 크기가 32비트이면 명령어 오퍼랜드는 16비트+32비트까지 확장될 수 있다. 그러므로 하드웨어를 단순화할 수 있어 임베디드 프로세서로서 장점을 갖고 있다. 이런 명령어 확장을 위한 LERI(Load Extention Register Immediate)라는 명령어가 존재한다.
- [0007] LERI 명령어는 임의의 값을 ER에 세트하고 e_flag를 세트하는 역할을 한다. 그러면 LERI 명령어 후에 실행되는 다른 명령어는 e_flag가 세트되었는지 체크하여, 세트되어 있다면 ER의 값을 자기 자신의 오퍼랜드(operand)에 더하여 동작을 수행하게 된다. 이런 방식으로 명령어의 오퍼랜드를 확장하게 된다.
- [0008] LERI명령어는 2비트 Op-Code와 14비트 즉시 값(immediate value)을 가지는 명령어로서 ER(Extension Register)에 즉시 값을 저장하고, 이후에 즉시 값의 확장이 필요한 명령어에서 ER의 값을 인출하여 해당 명령어의 즉시 값과 덧붙이는 구조를 지니고 있다. 이러한 구조의 장점은 짧은 즉시 값 크기로 인하여 발생 가능한 문제를 효과적으로 해결 할 수 있다는 것이나, LERI의 부가로 인하여 코드의 길이가 길어질 수 있다는 문제와 성능의 저하를 가져 올 수 있다는 문제를 지니고 있다. 따라서, EISC기반의 프로세서는 이 LERI를 효과적으로 처리하는 것이 중요하다.
- [0009] EISC 프로세서는 32 비트 프로세서(bit Processor)이므로 32 비트 Immediate 연산을 수행할 수 있으며, 16 비트(bit) 인스트럭션(Instruction)을 사용하기 때문에, Immediate 연산을 위한 인스트럭션(Instruction)에 4 비트(bit)의 즉시 값이 포함될 수 있으며, 더 큰 크기의 즉시 값을 사용하기 위해서는 별도의 방법이 필요하다. 따라서 종래 EISC 프로세서는 큰 값의 Immediate값이 필요한 경우 Immediate 연산을 처리하기 전에 LERI명령어로 ER 레지스터 값을 생성하며 Immediate 연산에서 미리 생성된 ER 레지스터 값을 사용한다.
- [0010] ER 레지스터는 LERI 명령어의 14 비트 즉시 값(Immediate value)을 부호 확장(Sign Extension)하여 생성된다. 그러나, 이러한 종래 방식에서는 비부호 즉시 값(Unsigned Immediate value)를 사용하는 경우에도 ER 레지스터를 무조건 부호 확장(Sign Extension)하기 때문에 추가적인 인스트럭션(Instruction)이 필요할 수 있으며, 이로 인하여 코드 밀도가 낮아지는 문제점이 발생한다.

선행기술문헌

특허문헌

- [0011] (특허문헌 0001) 대한민국 등록특허 10-0888675

발명의 내용

해결하려는 과제

- [0012] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, EISC 프로세서에서의 즉시 값 연산시에 불필요한 LERI 명령어의 수행을 방지함으로써, 연산 효율을 향상시킬 수 있는 즉시 값 연산 방법을 제공하는데 그 목적이 있다.
- [0013] 본 발명의 목적은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

- [0014] 이와 같은 목적을 달성하기 위한 본 발명의 EISC(Extendable Instruction Set Computer) 프로세서에서의 즉시

값(Immediate value) 연산 방법에 있어서, 즉치 값(Immediate value) 연산을 위한 ER(Extention Register) 레지스터 값 생성 시, 비부호 즉치 값(Unsigned Immediate value)을 사용하는지 여부를 확인하는 단계 및 비부호 즉치 값을 사용하는 경우, ULERI(Unsigned Load Extension Register with Immediate) 명령어를 이용하여 상기 비부호 즉치 값을 제로 확장(Zero Extension)하는 방식으로 상기 ER 레지스터 값을 생성하는 단계를 포함한다.

[0015] 즉치 값 연산을 위한 ER 레지스터 값 생성 시, 부호 즉치 값(Signed Immediate value)을 사용하는 경우, LERI(Load Extension Register with Immediate) 명령어를 이용하여 상기 즉치 값을 부호 확장(Sign Extension)하는 방식으로 상기 ER 레지스터 값을 생성하는 단계를 더 포함할 수 있다.

[0016] 상기 EISC 프로세서는 32 비트(bit) 프로세서로서 32 비트(bit)의 즉치 값 연산을 수행하고, 상기 ULERI 명령어 및 상기 LERI 명령어의 길이는 16 비트(bit)로 구성될 수 있다.

발명의 효과

[0017] 본 발명에 의하면 EISC 프로세서에서의 16비트 명령어를 이용한 즉치 값 연산시에 불필요한 LERI 명령어의 수행을 방지함으로써, 연산 효율을 향상시킬 수 있는 효과가 있다. 특히, 본 발명은 신호처리 분야 등의 비부호 값(Unsigned value)을 주로 사용하는 연산에서 매우 효율적이며, 코드 밀도를 높일 수 있고, 프로세서의 연산 효율을 높일 수 있는 효과가 있다.

도면의 간단한 설명

- [0018] 도 1은 LERI 명령어를 이용한 ER 레지스터 값 생성과정을 보여주는 도면이다.
- 도 2는 복수의 LERI 명령어를 이용한 ER 레지스터 값 생성과정을 보여주는 도면이다.
- 도 3은 LERI 명령어를 이용하여 비부호 즉치 값(Unsigned Immediate value)을 생성하는 과정을 보여주는 도면이다.
- 도 4는 ULERI 명령어를 이용하여 비부호 즉치 값을 생성하는 과정을 보여주는 도면이다.
- 도 5는 본 발명의 일 실시예에 따른 EISC 프로세서에서의 즉치 값 연산 방법을 보여주는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0019] 이하, 첨부된 도면을 참조해서 본 발명의 실시예를 상세히 설명하면 다음과 같다. 우선 각 도면의 구성 요소들에 참조 부호를 부가함에 있어서, 동일한 구성 요소들에 한해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 그리고, 본 발명을 설명함에 있어서, 관련된 공지 기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다. 또한, 명세서 전반에 걸쳐서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있다는 것을 의미한다.

[0020] 본 발명은 EISC(Extendable Instruction Set Computer) 프로세서에서의 즉치 값(Immediate value) 연산 방법을 제안한다.

[0021] 본 발명의 즉치 값(Immediate value) 연산 방법에서는 기존 사용중이던 LERI(Load Extension Register with Immediate)(ER 레지스터 부호 확장) 명령어와, 본 발명에서 새롭게 추가된 ULERI(Unsigned Load Extension Register with Immediate)(ER 레지스터 비부호 확장) 명령어를 사용하여 효율적으로 즉치 값 연산을 수행하는 방법을 제안한다.

[0022] 도 1은 LERI 명령어를 이용한 ER(Extention Register) 레지스터 값 생성과정을 보여주는 도면이다.

[0023] 도 1을 참조하면, EISC 프로세서는 LERI 명령어의 14비트(bit) 즉치 값을 32비트로 부호 확장(sign extension)하는 방식으로 ER 레지스터 값을 생성한다.

[0024] 이때, 하나의 LERI 명령어로 생성할 수 있는 값은 14 비트로 제한되며, 더 큰 크기의 즉치 값을 생성하기 위해

서는 복수의 LERI 명령어가 필요하다.

- [0025] 예를 들어, 부호 변수(Signed variable) 16진수 0x21FF가 32비트로 확장되는 경우, 부호 확장(Sign extension)하여 0xFFFFE1FF로 ER 레지스터에 저장되어야 하며, 도 1과 같이, LERI 명령어를 사용하여 처리될 수 있다.
- [0026] 도 2는 복수의 LERI 명령어를 이용한 ER 레지스터 값 생성과정을 보여주는 도면이다.
- [0027] 도 2를 참조하면, EISC 프로세서는 복수의 LERI 명령어를 이용하여 ER 레지스터 값을 생성한다. 즉, 부호확장하여 생성된 기존의 ER 레지스터 값과 새로 처리되는 LERI 명령어의 즉시 값을 사용하여 보다 큰 범위의 즉시 값을 생성할 수 있다. 이와 같은 방법으로 최대 32비트까지 즉시 값을 생성할 수 있다.
- [0028] 도 2에서는 LERI 명령어에서 14비트를 카피(copy)하고, 기존 ER 레지스터 값에서 18비트를 카피하는 방식으로 32비트의 ER 레지스터 값을 생성하는 과정을 보여주고 있다.
- [0029] 도 3은 LERI 명령어를 이용하여 비부호 즉시 값(Unsigned Immediate value)을 생성하는 과정을 보여주는 도면이다.
- [0030] 도 3에서는 비부호 변수(Unsigned variable) 16진수 0x21FF가 32비트로 확장되는 경우, 0x000021FF로 ER 레지스터에 저장되어야 하며, LERI 명령어를 사용하여 처리되는 과정이 도시되어 있다.
- [0031] 도 3을 참조하면, 2개의 LERI 명령어를 사용하여 비부호 16진수 즉시 값을 생성하는 과정에서, 먼저 LERI 명령어를 처리한다. 이 과정에서는 ER 레지스터에 32비트 데이터 0x00000000값이 생성된다(①).
- [0032] 다음, 기존의 ER 레지스터 값과 LERI 명령어의 즉시 값(0x21FF)을 이용하여 0x000021FF 값을 생성한다(②).
- [0033] 이처럼, EISC 프로세서에서 LERI 명령어를 이용하여 비부호 즉시 값을 생성하는 경우, 2개의 LERI 명령어가 필요한 구조이다.
- [0034] 도 4는 ULERI 명령어를 이용하여 비부호 즉시 값을 생성하는 과정을 보여주는 도면이다.
- [0035] 도 4를 참조하면, EISC 프로세서에서 ULERI(Unsigned Load Extension Register with Immediate) 명령어를 이용하여 비부호 즉시 값을 생성한다. 즉, EISC 프로세서는 ULERI 명령어를 통해 제로 확장(zero extension)하는 방식으로 ER 레지스터 값을 생성한다. 도 4에서는 제로 확장하여 32비트 데이터인 0x000021FF 값을 생성한다.
- [0036] 도 5는 본 발명의 일 실시예에 따른 EISC 프로세서에서의 즉시 값 연산 방법을 보여주는 흐름도이다.
- [0037] 도 5를 참조하면, EISC(Extendable Instruction Set Computer) 프로세서에서의 즉시 값(Immediate value) 연산 방법에 있어서, EISC 프로세서는 즉시 값(Immediate value) 연산을 위한 ER(Extention Register) 레지스터 값 생성이 필요한지 확인한다(S501).
- [0038] 즉시 값 연산을 위한 ER 레지스터 값 생성이 필요하면, 비부호 즉시 값(Unsigned Immediate value)을 사용하는지 여부를 확인한다(S503).
- [0039] 비부호 즉시 값을 사용하는 경우, ULERI(Unsigned Load Extension Register with Immediate) 명령어를 이용하여 비부호 즉시 값을 제로 확장(Zero Extension)하는 방식으로 ER 레지스터 값을 생성한다(S505).
- [0040] 다음, 필요한 ER 레지스터 값이 모두 생성되었는지 여부를 확인한다(S507). 필요한 ER 레지스터 값이 모두 생성될 때까지 ER 레지스터 값을 생성하고, 필요한 ER 레지스터 값이 모두 생성되면 절차를 종료한다.
- [0041] 본 발명에서 즉시 값 연산을 위한 ER 레지스터 값 생성 시, 부호 즉시 값(Signed Immediate value)을 사용하는 경우, LERI(Load Extension Register with Immediate) 명령어를 이용하여 즉시 값을 부호 확장(Sign Extension)하는 방식으로 ER 레지스터 값을 생성한다(S509).
- [0042] 다음, 필요한 ER 레지스터 값이 모두 생성되었는지 여부를 확인한다(S511). 필요한 ER 레지스터 값이 모두 생성될 때까지 ER 레지스터 값을 생성하고, 필요한 ER 레지스터 값이 모두 생성되면 절차를 종료한다.
- [0043] 본 발명에서 EISC 프로세서는 32 비트(bit) 프로세서로서 32 비트(bit)의 즉시 값 연산을 수행하고, ULERI 명령

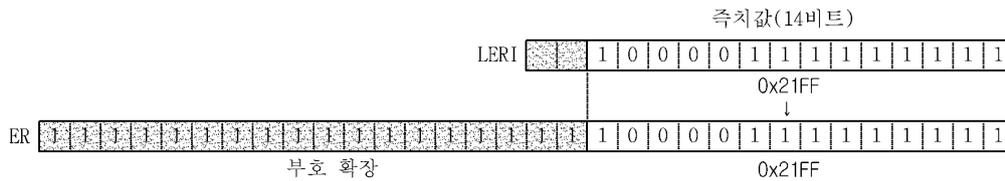
어 및 LERI 명령어의 길이는 16 비트(bit)이다.

[0044]

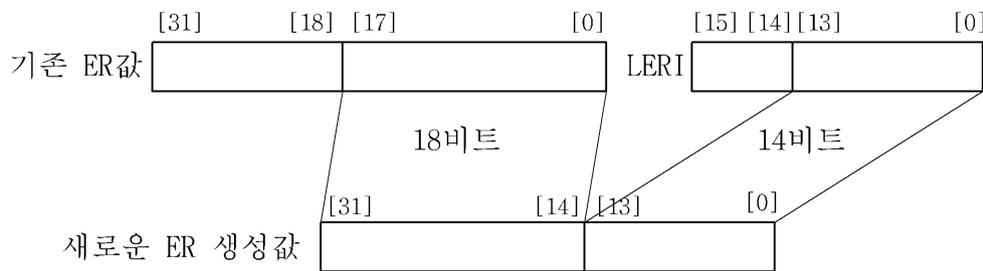
이상 본 발명을 몇 가지 바람직한 실시예를 사용하여 설명하였으나, 이들 실시예는 예시적인 것이며 한정적인 것이 아니다. 본 발명이 속하는 기술분야에서 통상의 지식을 지닌 자라면 본 발명의 사상과 첨부된 특허청구범위에 제시된 권리범위에서 벗어나지 않으면서 다양한 변화와 수정을 가할 수 있음을 이해할 것이다.

도면

도면1



도면2



도면3

