



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년05월27일
 (11) 등록번호 10-1624636
 (24) 등록일자 2016년05월20일

(51) 국제특허분류(Int. Cl.)
 G06F 9/38 (2006.01) G06F 12/08 (2016.01)
 (52) CPC특허분류
 G06F 9/3804 (2013.01)
 G06F 12/0875 (2013.01)
 (21) 출원번호 10-2015-0029730
 (22) 출원일자 2015년03월03일
 심사청구일자 2015년03월03일
 (56) 선행기술조사문헌
 KR1019890008644 A
 JP06067881 A
 WO2014120703 A1

(73) 특허권자
 주식회사 에이디칩스
 경기도 안양시 동안구 학의로 282, 에이동 22층(관양동, 금강펜테리움아파트타워)
 (72) 발명자
 김상완
 경기도 안양시 동안구 달안로 78, 610동 107호 (비산동, 셋별한양아파트)
 차영호
 경기 안양시 동안구 학의로 146, 203동 1204호 (관양동, 한가람삼성아파트)
 (뒷면에 계속)
 (74) 대리인
 김정현

전체 청구항 수 : 총 13 항

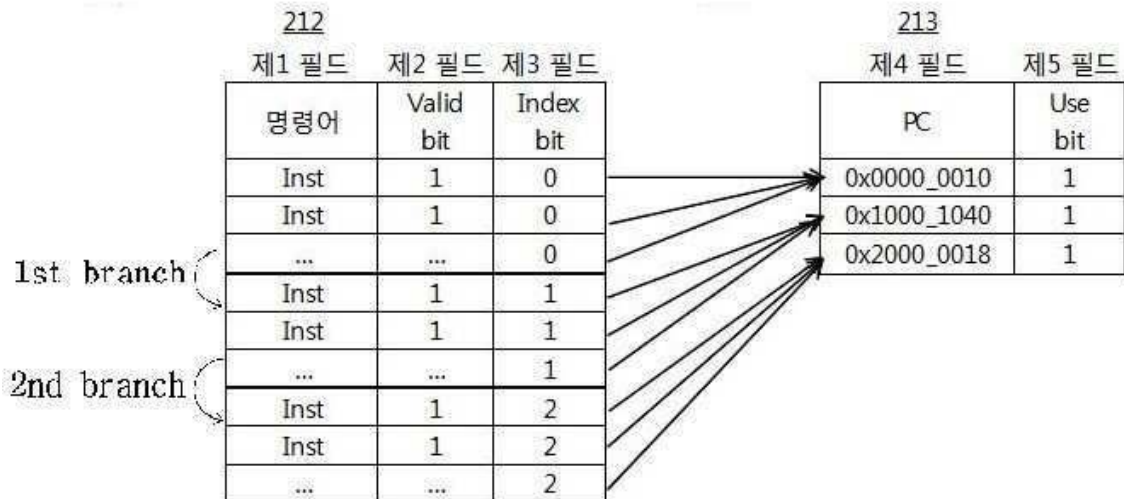
심사관 : 서광훈

(54) 발명의 명칭 **명령어 패치 장치 및 방법**

(57) 요약

본 발명은 명령어 패치 장치 및 방법에 대하여 개시한다. 본 발명의 일면에 따른 명령어 패치 유닛은, 각 분기에서 다음에 실행될 명령어의 주소를 저장하는 복수의 PC 버퍼; 실행될 명령어들과 상기 복수의 PC 버퍼 중 각 명령어와 연관된 PC 버퍼의 인덱스가 각기 기록되는 복수의 명령어 버퍼; 및 프로그램 메모리로부터 실행될 명령어를 하나씩 불러와 상기 복수의 명령어 버퍼에 순차적으로 저장하며, 분기 예측의 적중 이전에 상기 복수의 PC 버퍼 중에서 일 인덱스의 PC 버퍼를 이용하여 현재 분기에서 다음에 실행될 명령어를 표시하는 패치부를 포함하며, 상기 복수의 PC 버퍼의 개수는 상기 복수의 명령어 버퍼의 개수 미만인 것을 특징으로 한다.

대표도 - 도3



(52) CPC특허분류
G06F 9/3814 (2013.01)

(72) 발명자

김관영

경기도 성남시 분당구 정자일로213번길 5, 301동
1801호 (정자동, 아이파크분당3)

민병권

경기도 성남시 분당구 정자로 143, 209동 1601호
(정자동, 한솔마을LG아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호 10041664

부처명 지식경제부

연구관리전문기관 한국산업기술평가관리원

연구사업명 산업융합원천기술개발사업

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

기 여 율 1/1

주관기관 전자부품연구원

연구기간 2012.06.01 ~ 2016.05.31

명세서

청구범위

청구항 1

각 분기에서 다음에 실행될 명령어의 주소를 저장하는 복수의 PC 버퍼;

실행될 명령어들과 상기 복수의 PC 버퍼 중 각 명령어와 연관된 PC 버퍼의 인덱스가 각기 기록되는 복수의 명령어 버퍼; 및

프로그램 메모리로부터 실행될 명령어를 하나씩 불러와 상기 복수의 명령어 버퍼에 순차적으로 저장하며, 분기 예측의 적중 이전에 상기 복수의 PC 버퍼 중에서 일 인덱스의 PC 버퍼를 이용하여 현재 분기에서 다음에 실행될 명령어를 표시하는 패치부를 포함하며,

상기 복수의 PC 버퍼의 개수는 상기 복수의 명령어 버퍼의 개수 미만인 명령어 패치 유닛.

청구항 2

제1항에 있어서, 상기 패치부는,

상기 분기 예측의 적중 이전에는 상기 복수의 명령어 버퍼 내 명령어 중에서 실행완료된 명령어가 있으면, 상기 실행완료된 명령어가 있을 때마다 상기 일 인덱스의 PC 버퍼 내 상기 명령어의 주소를 상기 각 명령어의 크기만큼 증가시켜 상기 일 인덱스에 의해 상기 현재 분기에서 다음에 실행될 명령어의 주소를 표시하는 것인 명령어 패치 유닛.

청구항 3

제1항에 있어서, 상기 패치부는,

상기 분기 예측이 적중하면, 상기 일 인덱스와 이어지는 타 인덱스의 PC 버퍼를 이용하여 상기 현재 분기에서 다음에 실행될 명령어의 주소를 표시하는 것인 명령어 패치 유닛.

청구항 4

제1항에 있어서, 상기 복수의 PC 버퍼 각각은,

각기 상기 인덱스로 구분되며 상기 각 분기에서 다음에 실행될 명령어의 주소를 저장하는 제4 필드와, 각 PC 버퍼의 사용중 여부를 표시하는 사용비트를 저장하는 제5 필드를 포함하는 것인 명령어 패치 유닛.

청구항 5

제1항에 있어서, 상기 복수의 명령어 버퍼의 각각은,

상기 실행될 명령어를 하나씩 저장하는 제1 필드, 상기 제1 필드 내 명령어와 연관된 PC 버퍼의 인덱스를 저장하는 제2 필드 및 상기 제1 필드 내 명령어의 유효성을 표시하는 제3필드를 포함하는 것인 명령어 패치 유닛.

청구항 6

제1항에 있어서, 상기 패치부는,

상기 복수의 PC 버퍼 및 상기 복수의 명령어 버퍼가 둘 중 적어도 하나가 풀(Full)이면, 상기 복수의 명령어 버퍼 중 적어도 하나 내 명령어가 실행됨에 따라 상기 복수의 PC 버퍼와 상기 복수의 명령어 버퍼가 둘 다 풀(Full)이 아닐 때까지 상기 프로그램 메모리로부터 실행될 명령어의 인출을 중단하는 것인 명령어 패치 유닛.

청구항 7

제6항에 있어서, 상기 패치부는,

상기 복수의 PC 버퍼 내 명령어 중 하나가 실행완료되면, 상기 실행완료된 명령어를 저장한 현재 명령어 버퍼의

유효성비트를 디스에이블(Disable)하고, 상기 실행완료된 명령어의 다음 명령어가 저장된 다음 명령어 버퍼의 PC 버퍼의 인덱스의 변화 여부를 확인하는 것인 명령어 패치 유닛.

청구항 8

제7항에 있어서, 상기 패치부는,

상기 다음 명령어 버퍼의 PC 버퍼의 인덱스가 변화됨을 확인하면, 상기 현재 명령어 버퍼에 의해 지정된 현재 PC 버퍼의 사용완료를 표시하고, 상기 다음 명령어 버퍼에 의해 지정되는 다음 PC 버퍼를 이용하여 상기 현재 분기에서 다음으로 실행될 명령어를 표시하는 것인 명령어 패치 유닛.

청구항 9

일 패치 프로세서에 의한 각 분기에서 다음에 실행될 명령어의 주소를 저장하는 복수의 PC 버퍼; 및 실행될 명령어들과 상기 복수의 PC 버퍼 중 각 명령어와 연관된 PC 버퍼의 인덱스가 각기 기록되며, 상기 복수의 PC 버퍼의 개수를 초과하는 복수의 명령어 버퍼를 이용한 명령어 패치 방법으로서,

상기 복수의 PC 버퍼 및 상기 복수의 명령어 버퍼가 둘 다 풀이 아니면, 프로그램 메모리로부터 실행될 명령어를 하나씩 불러오는 단계; 및

분기 예측이 적중하지 않으면, 상기 복수의 PC 버퍼 중에서 일 인덱스에 의해 지정되는 하나의 PC 버퍼를 이용하여 현재 분기에서 다음에 실행될 명령어를 표시하는 단계

를 포함하는 명령어 패치 방법.

청구항 10

제9항에 있어서, 상기 불러오는 단계는,

상기 복수의 PC 버퍼 및 상기 복수의 명령어 버퍼가 둘 중 적어도 하나가 풀(Full)이면, 상기 복수의 명령어 버퍼 중 적어도 하나 내 적어도 하나의 명령어가 실행완료되어 상기 복수의 PC 버퍼와 상기 복수의 명령어 버퍼가 둘 다 풀(Full)이 아닐 때까지 상기 프로그램 메모리로부터 실행될 명령어의 인출을 중단하는 단계

를 포함하는 것인 명령어 패치 방법.

청구항 11

제10항에 있어서,

상기 복수의 PC 버퍼 내 명령어 중 하나가 실행완료되면, 상기 실행완료된 명령어를 저장한 현재 명령어 버퍼의 유효성비트를 디스에이블(Disable)하는 단계 및

상기 실행완료된 명령어의 다음 명령어가 저장된 다음 명령어 버퍼의 PC 버퍼의 인덱스의 변화 여부를 확인하는 단계

를 더 포함하는 명령어 패치 방법.

청구항 12

제11항에 있어서, 상기 확인하는 단계는,

상기 다음 명령어 버퍼 내 PC 버퍼의 인덱스가 변화됨을 확인하면, 상기 현재 명령어 버퍼에 의해 지정된 현재 PC 버퍼의 사용 여부를 표시하는 사용비트를 디스에이블하는 단계; 및

상기 다음 명령어 버퍼에 의해 지정되는 다음 PC 버퍼를 이용하여 상기 현재 분기에서 다음으로 실행될 명령어를 표시하는 단계

를 포함하는 것인 명령어 패치 방법.

청구항 13

제9항에 있어서, 상기 표시하는 단계는,

상기 복수의 명령어 버퍼 내 명령어 중에서 실행완료된 명령어가 있으면, 상기 실행완료된 명령어가 있을 때마

다 상기 일 인덱스의 PC 버퍼 내 상기 명령어의 주소를 상기 각 명령어의 크기만큼 증가시켜 상기 일 인덱스에 의해 상기 현재 분기에서 다음에 실행될 명령어의 주소를 표시하는 단계를 포함하는 것인 명령어 패치 방법.

발명의 설명

기술 분야

[0001] 본 발명은 분기 예측 기술에 관한 것으로서, 더 구체적으로는 분기 예측 기법을 수행할 때의 명령어 패치 장치 및 방법에 관한 것이다.

배경 기술

[0002] 최근, 컴퓨팅 시스템은 명령 처리량을 증가시키기 위해 파이프라인식 구조를 가진 프로세서를 사용한다. 파이프라인식 프로세서에서는 제1 명령의 실제적인 실행이 완료되기 전에 제2 명령의 처리를 시작함으로써 레이턴시가 감소된다.

[0003] 중앙 처리 장치(CPU)의 명령에는 계산 결과에 따라 별도의 주소로 분기하는 브랜치 명령이 있는데, 파이프라인 구조에서는 분기가 생기면 파이프라인에 들어 있는 명령을 모두 버려 처리가 지연된다.

[0004] 이러한 현상을 분기 지연(Branch Penalty)이라고 하는데, 이를 방지하고자 분기 예측(Branch Prediction) 기법이 사용되고 있다. 이때, 분기 예측 기법은 CPU 명령의 분기 여부를 미리 예측하고 분기하면 파이프라인에 유입된 명령을 변화시켜 분기 지연 발생을 방지하는 기술이다.

[0005] 중앙 처리 장치는 분기 예측 기법을 사용하면서 실행할 명령어를 저장하는 명령어버퍼(Inst Buffer)를 운용하며, 명령어버퍼와 연계된 PC버퍼에 PC(Program Counter)도 함께 저장한다. 이는 PC에 의해 분기된 명령어를 실행한 후에 이어서 실행할 명령을 알기 위함이다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 전술한 바와 같은 기술적 배경에서 안출된 것으로서, 분기 예측 기법을 이용하는 중앙 처리 장치의 명령어 패치 장치 및 방법을 제공하는 것을 그 목적으로 한다.

[0007] 본 발명의 목적은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0008] 본 발명의 일면에 따른 명령어 패치 유닛은, 각 분기에서 다음에 실행될 명령어의 주소를 저장하는 복수의 PC 버퍼; 실행될 명령어들과 상기 복수의 PC 버퍼 중 각 명령어와 연관된 PC 버퍼의 인덱스가 각기 기록되는 복수의 명령어 버퍼; 및 프로그램 메모리로부터 실행될 명령어를 하나씩 불러와 상기 복수의 명령어 버퍼에 순차적으로 저장하며, 분기 예측의 적중 이전에 상기 복수의 PC 버퍼 중에서 일 인덱스의 PC 버퍼를 이용하여 현재 분기에서 다음에 실행될 명령어를 표시하는 패치부를 포함하며, 상기 복수의 PC 버퍼의 개수는 상기 복수의 명령어 버퍼의 개수 미만인 것을 특징으로 한다.

[0009] 본 발명의 다른 면에 따른 일 패치 프로세서에 의한 각 분기에서 다음에 실행될 명령어의 주소를 저장하는 복수의 PC 버퍼; 및 실행될 명령어들과 상기 복수의 PC 버퍼 중 각 명령어와 연관된 PC 버퍼의 인덱스가 각기 기록되며, 상기 복수의 PC 버퍼의 개수를 초과하는 복수의 명령어 버퍼를 이용한 명령어 패치 방법은, 상기 복수의 PC 버퍼 및 상기 복수의 명령어 버퍼가 둘 다 풀이 아니면, 프로그램 메모리로부터 실행될 명령어를 하나씩 불러오는 단계; 및 분기 예측이 적중하지 않으면, 상기 복수의 PC 버퍼 중에서 일 인덱스에 의해 지정되는 하나의 PC 버퍼를 이용하여 현재 분기에서 다음에 실행될 명령어를 표시하는 단계를 포함하는 것을 특징으로 한다.

발명의 효과

[0010] 본 발명에 따르면, 분기 예측 기법에서 사용되는 PC 버퍼의 수를 줄일 수 있다.

도면의 간단한 설명

- [0011] 도 1은 본 발명의 실시예에 따른 중앙 처리 장치를 도시한 구성도.
- 도 2는 본 발명의 실시예에 따른 명령어 패치 유닛을 도시한 구성도.
- 도 3은 본 발명의 실시예에 따른 복수의 명령어 버퍼와 복수의 PC 버퍼를 세부적으로 도시한 도면.
- 도 4는 본 발명의 실시예에 따른 인출 중단의 경우를 도시한 도면.
- 도 5a는 본 발명의 실시예에 따른 명령어 패치 방법을 도시한 흐름도.
- 도 5b는 본 발명의 실시예에 따른 명령어 패치 과정의 명령어 버퍼와 PC 버퍼를 도시한 도면.
- 도 6a는 본 발명의 실시예에 따른 명령어 실행 과정의 명령어 패치 방법을 도시한 흐름도.
- 도 6b 내지 6d는 본 발명의 실시예에 따른 명령어 실행 과정의 명령어 버퍼와 PC 버퍼를 도시한 도면.

발명을 실시하기 위한 구체적인 내용

- [0012] 본 발명의 전술한 목적 및 그 이외의 목적과 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 것이며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 한편, 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소, 단계, 동작 및/또는 소자는 하나 이상의 다른 구성요소, 단계, 동작 및/또는 소자의 존재 또는 추가를 배제하지 않는다.
- [0013] 이제 본 발명의 실시예에 대하여 첨부한 도면을 참조하여 상세히 설명하기로 한다. 도 1은 본 발명의 실시예에 따른 중앙 처리 장치를 도시한 구성도이다.
- [0014] 도 1에 도시된 바와 같이, 본 발명의 실시예에 따른 중앙 처리 장치(20)는 패치 유닛(2100), 디코더(2200), 분기 예측기(2400) 및 실행 유닛(2300)을 포함한다.
- [0015] 패치 유닛(2100)은 프로그램 메모리(10)에 명령어의 주소를 제공하고, 해당 주소의 명령어를 가져와 명령어 버퍼에 저장하고, 실행할 명령어의 주소를 PC 버퍼에 저장한다. 패치 유닛(2100)의 구체 구성에 대해서는 도 2와 함께 후술한다.
- [0016] 디코더(2200)는 패치 유닛(2100)으로부터 실행할 명령어를 가져와 디코딩하며, 디코딩된 코드를 실행 유닛(2300)에 제공한다.
- [0017] 실행 유닛(2300)은 디코딩된 명령어를 실행하고, 각 명령어의 실행 완료를 패치 유닛(2100)에 알린다.
- [0018] 분기 예측기(2400)는 패치 유닛(2100)으로부터 불러온 명령어의 주소를 수신하고 기설정된 분기 예측 알고리즘에 의해 예측된 분기 예측 주소와 일치하면(이하, "분기 예측 적중"이라고 함), 패치 유닛(2100)에 분기 예측의 적중을 알린다. 반면, 분기 예측기(2400)는 패치 유닛(2100)으로부터의 명령어의 주소와 분기 예측 주소와 일치하지 않으면, 분기 예측에 실패한 것으로 판단한다.
- [0019] 이하, 도 2를 참조하여 본 발명의 실시예에 따른 명령어 패치 유닛에 대하여 설명한다. 도 2는 본 발명의 실시예에 따른 명령어 패치 유닛을 도시한 구성도이고, 도 3은 본 발명의 실시예에 따른 복수의 명령어 버퍼와 복수의 PC 버퍼를 세부적으로 도시한 도면이고, 도 4는 본 발명의 실시예에 따른 인출 중단의 경우를 도시한 도면이다.
- [0020] 도 2에 도시된 바와 같이, 본 발명의 실시예에 따른 명령어 패치 유닛(2100)은 복수의 명령어 버퍼(212), 복수의 PC 버퍼(213) 및 패치부(211)를 포함한다.

- [0021] 각 명령어 버퍼(212)는 각기 실행될 명령어 및 각 명령어와 관련된 정보들을 저장하는 제2 및 제3필드로 구성된다. 제1 필드 내지 제3필드는 각기 1:1로 대응된다.
- [0022] 상세하게는, 도 3과 같이, 제1 필드에는 각기 실행될 명령어가 순차적으로 저장되며, 제2 필드에는 각 명령어의 유효성(즉, 실행완료 여부)을 나타내는 유효성비트(Valid Bit)가 저장되며, 제3필드에는 각 명령어와 관련된 PC 버퍼의 인덱스를 나타내는 인덱스비트(Index Bit)가 저장된다. 도 3에서는 복수의 PC 버퍼(213)가 총 3개인 경우를 예로 들어 도시하였다.
- [0023] 여기서, 유효성비트는 1비트로 그와 쌍을 이루는 제1 필드 내 명령어의 인에이블 또는 디스에이블을 표시한다.
- [0024] 상세하게는, 유효성비트는 그가 가리키는 제1 필드 내 해당 명령어가 실행되지 않은 상태이면, 인에이블(예컨대, '1'로 설정)되어 해당 명령어가 유효함을 나타낸다. 반면, 유효성비트는 해당 명령어가 실행 완료되면, 디스에이블(예컨대, '0'으로 설정)되어 해당 명령어가 더 이상 유효하지 않음을 나타낸다. 이때, 유효성비트는 명령어의 유효시에 0으로 설정되고 유효하지 않을 시에는 1로 설정될 수도 있음은 물론이다.
- [0025] 또한, 인덱스비트는 모든 PC 버퍼(213)를 인덱싱 가능한 크기로 구성된다. 예를 들어, 복수의 PC 버퍼(213)가 총 4개인 경우, 인덱스비트는 4개의 PC 버퍼를 각기 다른 인덱스로 지정할 수 있도록 2비트로 설정될 수 있다.
- [0026] 각 PC 버퍼(213)는 실행될 명령어의 주소(PC) 및 그 정보를 저장하는 제4 필드 및 제5 필드로 구성된다. 도 3과 같이, 제4 필드는 각 분기에서 다음으로 실행될 명령어의 주소(PC)가 저장되며, 제5 필드에는 제4 필드(또는, PC 버퍼)의 사용 여부를 표시하는 사용비트(Use bit)가 저장된다. 도 3에 도시된 바와 같이, 본 발명에서는 동일한 분기에서 복수의 명령어에 대해 동일한 PC 버퍼를 사용하여 다음에 실행하거나 불러올 명령어의 주소를 표현할 수 있다. 따라서, 복수의 PC 버퍼(213)의 개수는 복수의 명령어 버퍼(212)의 개수 미만이다. 이 같이, 본 발명에서는 복수의 명령어에 대해 하나의 PC 버퍼를 사용하여 PC 버퍼가 명령어 버퍼와 1:1로 구비되어 필요한 PC 버퍼의 수가 많아지는 문제를 방지할 수 있다.
- [0027] 여기서, 사용비트는 1비트로 구성되며, 그와 쌍을 이루는 제4 필드 또는 PC 버퍼의 사용 여부를 표시한다. 예컨대, 해당 PC 버퍼가 사용중이면, 사용비트는 인에이블되며, 해당 PC 버퍼가 사용완료되면, 사용비트는 디스에이블된다.
- [0028] 한편, 제1 필드와 제4 필드의 크기는 각 명령어의 크기에 대응할 수 있다. 예를 들어, 각 명령어의 크기가 32bit일 경우 제1 필드와 제4 필드의 크기는 32비트일 수 있다.
- [0029] 패치부(211)는 프로그램 메모리(10)로부터 실행될 명령어를 하나씩 불러와(Load), 복수의 명령어 버퍼(212)에 순차적으로 저장시키되, 각 분기에서 첫 번째 명령어의 첫 번째 명령어의 실행 이전에는 도 3과 같이, 각 분기에서 첫 번째 명령어의 주소를 PC 버퍼에 저장시킨다.
- [0030] 여기서, 패치부(211)는 명령어를 불러오기 전에 복수의 PC 버퍼(213) 및 복수의 명령어 버퍼(212) 중 적어도 하나가 풀(Full)이 아닌지를 확인하고, 둘 다 풀이 아닌 경우에만 명령어를 가져온다.
- [0031] 반면, 패치부(211)는 복수의 PC 버퍼(213) 및 복수의 명령어 버퍼(212) 중 적어도 하나가 풀이면, 적어도 하나의 명령어의 실행이 완료될 때까지 명령어의 인출을 중단한다(Fatch Stop).
- [0032] 예를 들어, 도 4와 같이, 패치부(211)는 프로그램 메모리(10)로부터 세 번째 분기의 첫 번째 명령어를 가져오려 하지만, 명령어 버퍼(212)의 공간은 남아 있지만, 복수의 PC 버퍼(213)가 모두 사용중(Use bit가 모두 1임)임을 확인한다. 따라서, 패치부(211)는 더 이상 명령어를 가져오지 못하고, 복수의 PC 버퍼(213) 중 적어도 하나가 빌 때까지 명령어의 인출을 중단한다.
- [0033] 패치부(211)는 명령어를 가져온 후 분기 예측의 적중 여부를 확인하고, 분기 예측의 적중 상태가 아니면, 불러온 명령어를 복수의 명령어 버퍼(212)에 저장한다. 이때, 패치부(211)는 이전에 설정된 PC 버퍼의 인덱스가 있으면, 즉, 불러온 명령어가 해당 분기의 두 번째 이후의 명령어이면, 불러온 명령어의 주소를 PC 버퍼에 별도로 기록하지 않는다. 반면, 패치부(211)는 이전에 설정된 인덱스가 없으면, 즉, 불러온 명령어가 분기 후 첫 번째 명령어이면, PC 버퍼의 인덱스를 증가시킨 후 증가된 인덱스에 대응하는 PC 버퍼에 불러온 명령어의 주소를 기록한다.
- [0034] 예를 들어, 패치부(211)는 프로그램 메모리로부터 첫 번째 명령어 A를 가져오면, 제1 필드에는 명령어 A를 저장하고, 제2 필드에 첫 번째 PC 버퍼의 인덱스인 0을 기록한다. 또한, 인덱스 0의 PC 버퍼에 명령어 A의 주소를 기록한다. 이후, 패치부(211)는 명령어 B를 가져오면, 복수의 PC 버퍼(213)와 복수의 명령어 버퍼(212)가 들

다 풀이 아니고 명령어 B가 분기 명령어가 아닌 경우, 제1 필드에 명령어 B를 저장하고 PC 버퍼에는 별도의 기록을 하지 않는다.

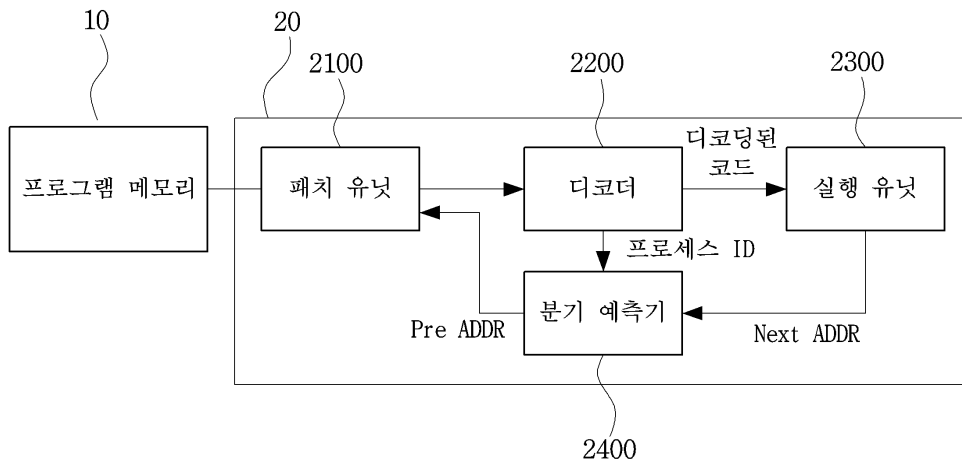
- [0035] 패치부(211)는 복수의 명령어 버퍼(212) 내 명령어 중 하나가 실행됨을 확인하면, 실행된 명령어를 저장하는 명령어 버퍼의 유효성비트를 디스에이블한다.
- [0036] 또한, 패치부(211)는 다음에 실행될 명령어를 확인하고, 현재 실행된 명령어와 비교할 때 다음 실행될 명령어의 PC 버퍼의 인덱스비트가 증가되었으면, 현재 실행된 명령어의 주소를 저장하는 PC 버퍼의 제5 필드를 디스에이블한다.
- [0037] 이때, 패치부(211)는 다음에 실행될 명령어의 PC 버퍼의 인덱스비트가 증가되지 않았으면, 현재 사용중인 PC 버퍼에 저장된 명령어의 주소를 명령어의 크기(예컨대, 4비트)만큼 증가시킨다. 이 같이, 본 발명에서는 동일한 분기에서 현재 실행되는 명령어와 다음에 실행되는 명령어의 주소는 명령어의 크기만큼 차이 있다는 특징을 이용하여 동일한 분기에서는 하나의 PC 버퍼를 이용하여 명령어를 실행할 수 있다.
- [0038] 한편, 전술한 예에서 패치부(211)는 불러온 명령어를 명령어 버퍼(212)에 저장하는 과정과 실행 유닛(2300)으로부터 현재 실행된 명령어를 확인하고 실행된 명령어를 저장한 명령어 버퍼(212)의 Valid bit와 그 주소를 저장한 PC 버퍼의 주소를 업카운트하거나, 그 사용비트를 디스에이블하는 과정은 병렬로 진행한다.
- [0039] 이와 같이, 본 발명의 실시예는 하나의 분기 내 복수의 명령어에 대해 하나의 PC 버퍼를 사용하므로 분기 예측 기법을 위해 사용되는 전체 PC 버퍼의 개수를 줄일 수 있다.
- [0040] 이하, 도 5a 및 6b를 참조하여 본 발명의 실시예에 따른 명령어 패치 방법에 대하여 설명한다. 도 5a는 본 발명의 실시예에 따른 명령어 패치 방법을 도시한 흐름도이고, 도 5b는 본 발명의 실시예에 따른 명령어 패치 과정의 명령어 버퍼와 PC 버퍼를 도시한 도면이다.
- [0041] 도 5를 참조하면, 패치 유닛(2100)은 명령어를 불러오기 전에 복수의 명령어 버퍼(212) 및 복수의 PC 버퍼(213)가 둘 다 풀이 아닌지를 확인한다(S500).
- [0042] 복수의 명령어 버퍼(212) 및 복수의 PC 버퍼(213)가 둘 다 풀이 아니면, 패치 유닛(2100)은 프로그램 메모리로부터 명령어를 불러온 후(S510), 분기 예측의 적중 여부를 확인한다(S520).
- [0043] 패치 유닛(2100)은 분기 예측이 적중되지 않음을 확인하면, 불러온 명령어를 명령어 버퍼에 저장하고, 해당 명령어 버퍼의 유효성비트를 인에이블한다(S530).
- [0044] 이때, 패치 유닛(2100)은 이전에 불러온 명령어를 저장한 명령어 버퍼의 다음에 위치한 명령어 버퍼(212)에 불러온 명령어를 저장할 수 있다. 여기서, 패치 유닛(2100)은 불러온 명령어를 저장한 명령어 버퍼(212) 내 PC 버퍼 인덱스비트를 이전에 불러온 명령어의 PC 버퍼 인덱스비트와 동일하게 설정할 수 있다. 만약, 불러온 명령어가 첫 번째로 불러온 명령어이면, 패치 유닛(2100)은 인덱스비트를 0으로 설정하여 해당 명령어를 저장한 명령어 버퍼가 첫 번째 PC 버퍼를 지정하도록 할 수 있다.
- [0045] 반면, 패치 유닛(2100)은 분기 예측이 적중됨을 확인하면 즉, 다음의 명령어가 분기 명령어임을 확인하면, 불러온 명령어를 다음에 사용될 명령어 버퍼에 저장하면서, 현재 사용중인 PC 버퍼가 아닌 다른 PC 버퍼를 지정한다(S540). 이때, 패치 유닛(2100)은 불러온 명령어를 저장하는 명령어 버퍼의 PC 버퍼 인덱스비트를 증가시켜 설정한다.
- [0046] 복수의 명령어 버퍼(212) 및 복수의 PC 버퍼(213) 중 적어도 하나가 풀이면, 패치 유닛(2100)은 도 4와 같이, 명령어를 불러오지 않고 대기하면서(Fatch Stop), 복수의 명령어 버퍼(212) 및 복수의 PC 버퍼(213)가 둘 다 풀 아닐때까지 대기한다(S550). 이때, 적어도 하나의 명령어가 실행되어야 명령어 버퍼(212) 및 PC 버퍼가 빌(Empty) 수 있으므로, 패치 유닛(2100)은 적어도 하나의 명령어의 실행완료될 때까지 대기할 수 있다.
- [0047] 도 5b와 같이, 명령어의 실행 없이 아닌 명령어의 패치만 수행되었을 경우, PC 버퍼는 각 분기에서 첫 번째로 패치(인출)된 명령어의 주소를 저장할 수 있다.
- [0048] 이하, 도 6a 내지 6d를 참조하여 본 발명의 실시예에 따른 명령어 실행중 명령어 버퍼와 PC 버퍼의 사용에 대하여 살펴본다. 도 6a는 본 발명의 실시예에 따른 명령어 실행 과정의 명령어 패치 방법을 도시한 흐름도이고,

도 6b 내지 6d는 본 발명의 실시예에 따른 명령어 실행 과정의 명령어 버퍼와 PC 버퍼를 도시한 도면이다.

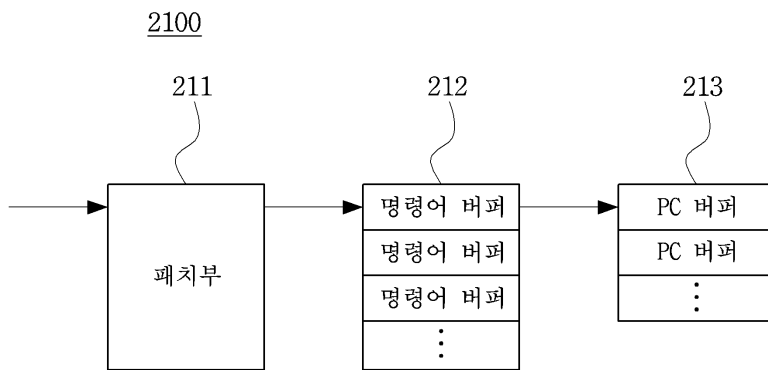
- [0049] 도 6a와 같이, 패치 유닛(2100)은 명령어의 인출 과정과 병렬적으로 실행완료된 명령어가 있는지를 확인한다(S600). 이때, 패치 유닛(2100)은 실행 유닛(2300)으로부터 실행완료된 명령어의 대한 보고를 수신하여 명령어의 실행완료를 확인할 수 있다.
- [0050] 패치 유닛(2100)은 실행완료된 명령어가 저장된 명령어 버퍼의 유효성비트를 디스에이블한다(S610). 이때, 패치 유닛(2100)은 실행 유닛(2300)으로부터 이전에 디코더(2200)로 전달한 명령어의 실행완료를 보고받으면, 실행완료된 명령어의 유효성비트를 디스에이블할 수 있다.
- [0051] 패치 유닛(2100)은 현재 실행완료된 명령어의 명령어 버퍼와 비교할 때 다음으로 실행될 명령어가 저장된 명령어 버퍼의 PC 버퍼 인덱스비트가 증가하는지를 확인한다(S630). 다시 말해, 다음으로 실행될 명령어가 분기 명령어일 경우 다른 PC 버퍼를 사용해야 하므로, 패치 유닛(2100)은 명령어 버퍼의 인덱스비트의 변화를 모니터링하는 것이다.
- [0052] 패치 유닛(2100)은 다음으로 실행될 명령어가 저장된 명령어 버퍼의 PC 버퍼 인덱스가 증가되지 않았으면, 현재 사용중인 PC 버퍼의 PC 값만 명령어의 크기만큼 증가시킨다(S640).
- [0053] 반면, 패치 유닛(2100)은 다음으로 실행될 명령어가 저장된 명령어 버퍼의 PC 버퍼 인덱스가 증가되면, 현재 사용중인 PC 버퍼의 사용비트를 디스에이블한다(S650).
- [0054] 이하, 도 6b 내지 6d를 참조하여 전술한 과정의 구체적 예를 설명한다.
- [0055] 도 6b와 같이, 패치 유닛(2100)은 첫 번째 명령어 버퍼 내 명령어를 디코더(2200)로 전달한 후 즉, 첫 번째 명령어의 실행중(Execute)에, 첫 번째 명령어 버퍼가 가리키는 첫 번째 PC 버퍼 내 PC값을 명령어의 크기(4)만큼 증가시킨다(S611). 즉, 패치 유닛(2100)은 첫 번째 PC 버퍼내 다음으로 실행될 명령어의 주소가 저장되도록 한다.
- [0056] 도 6c와 같이, 패치 유닛(2100)은 첫 번째 명령어 버퍼 내 명령어의 실행완료를 실행 유닛(2300)으로부터 확인하면, 첫 번째 명령어의 유효성비트를 디스에이블한다(S612).
- [0057] 이어서, 도 6c와 같이, 패치 유닛(2100)은 두 번째 명령어 실행할 명령어(두 번째 명령어 버퍼 내 명령어)를 디코더(2200)로 전달하고, PC 버퍼 내 PC 값을 명령어의 크기(4)만큼 증가시킨다(도 6c의 S613).
- [0058] 그 다음으로, 도 6d와 같이, 패치 유닛(2100)은 두 번째 명령어의 실행완료를 보고받으면, 두 번째 명령어의 유효성비트를 디스에이블하고(S614), 세 번째 명령어를 확인한다.
- [0059] 이때, 패치 유닛(2100)은 두 번째 명령어의 인덱스비트와 비교하여 세 번째 명령어의 인덱스비트가 증가하는지 여부를 확인하고, 인덱스비트가 증가함을 확인하면, 이전의 사용중인 PC 버퍼의 사용비트를 디스에이블한다(S631).
- [0060] 그리고, 패치 유닛(2100)은 증가된 인덱스비트에 대응하는 두 번째 PC 버퍼를 지정하고 세 번째 명령어를 디코더(2200)로 전달하며, 두 번째 PC 버퍼의 PC를 명령어의 크기만큼 증가시킨다(S632).
- [0061] 이와 같이, 본 발명의 실시예는 하나의 분기 내 복수의 명령어에 대해 하나의 PC 버퍼를 사용하므로 분기 예측 기법을 위해 사용되는 전체 PC 버퍼의 개수를 줄일 수 있다.
- [0062] 이상, 본 발명의 구성에 대하여 첨부 도면을 참조하여 상세히 설명하였으나, 이는 예시에 불과한 것으로서, 본 발명이 속하는 기술분야에 통상의 지식을 가진자라면 본 발명의 기술적 사상의 범위 내에서 다양한 변형과 변경이 가능함은 물론이다. 따라서 본 발명의 보호 범위는 전술한 실시예에 국한되어서는 아니되며 이하의 특허청구 범위의 기재에 의하여 정해져야 할 것이다.

도면

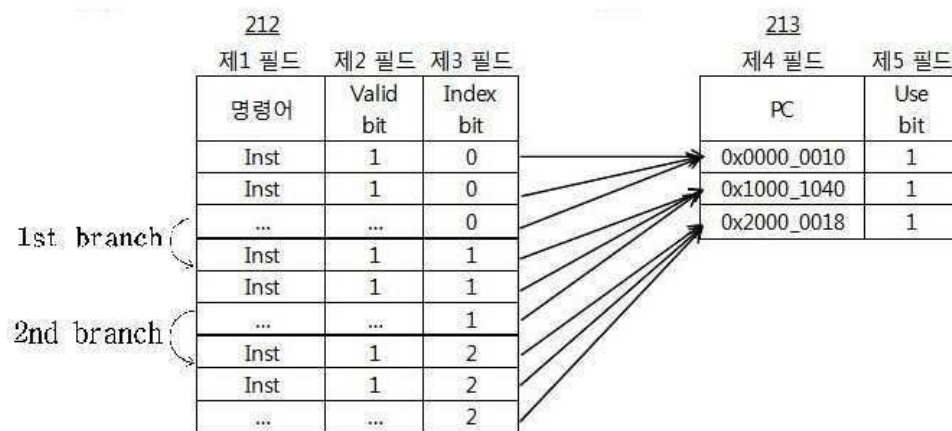
도면1



도면2



도면3



도면4

212

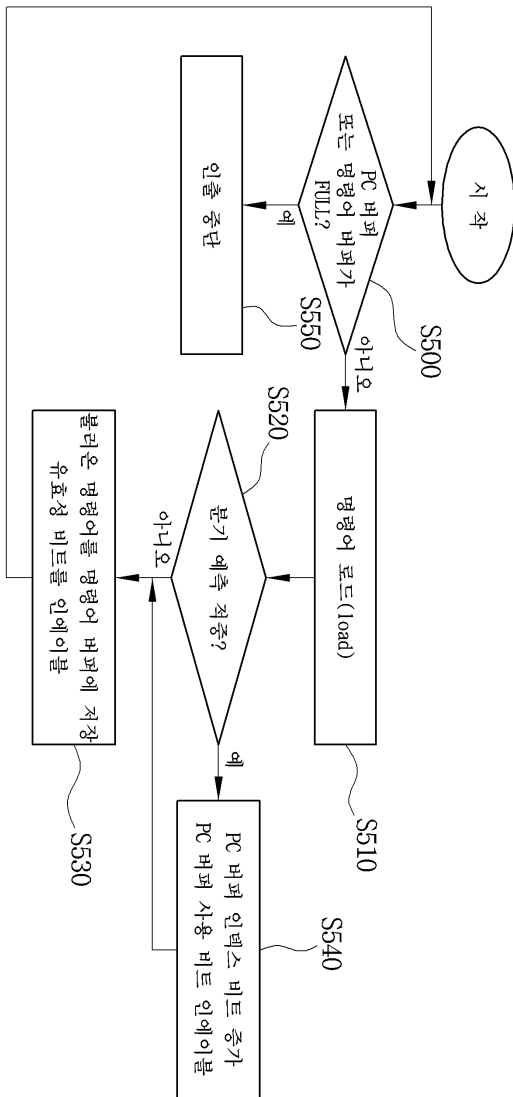
명령어	Valid bit	Index bit
Inst	1	0
Inst	1	0
Inst	1	1
Inst	1	1
Inst	1	2
Inst	1	2
		0
		0

1st branch
2nd branch
3rd branch

213

PC	Use bit
0x0000_0010	1
0x1000_1040	1
0x2000_0018	1

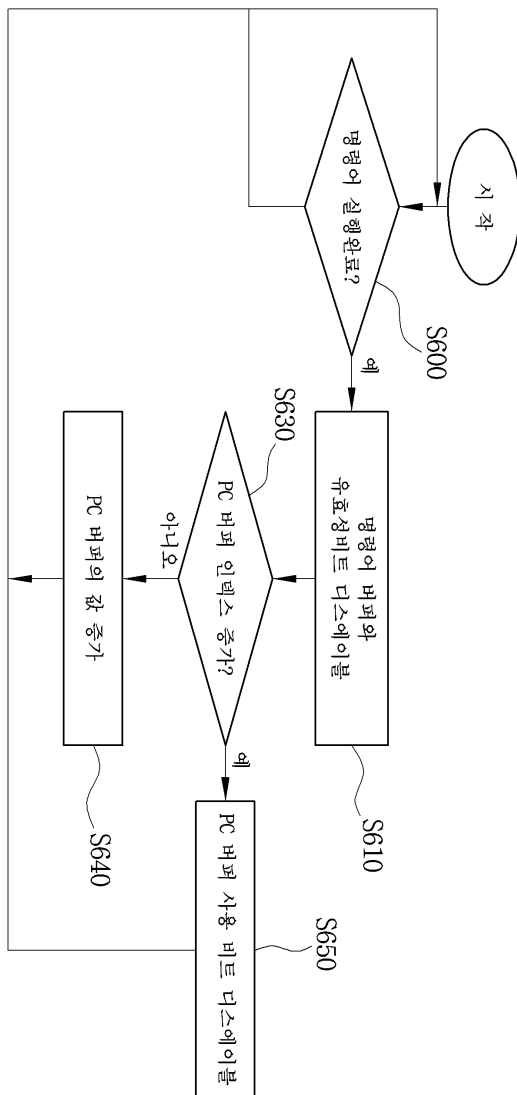
도면5a



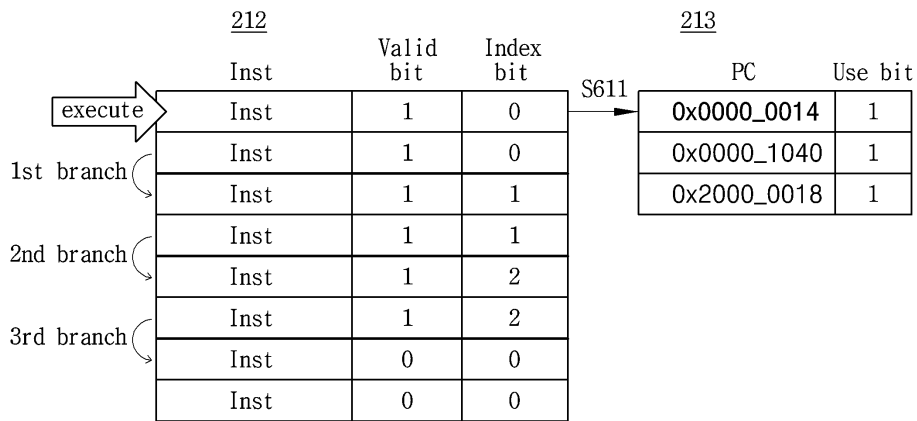
도면5b

	<u>212</u> Inst	Valid bit	Index bit	<u>213</u> PC	Use bit
1st branch	Inst	1	0	0x0000_0014	1
	Inst	1	0	0x0000_1040	1
	Inst	1	1	0x2000_0018	1
2nd branch	Inst	1	1		
	Inst	1	2		
3rd branch	Inst	1	2		
	Inst	0	0		
	Inst	0	0		

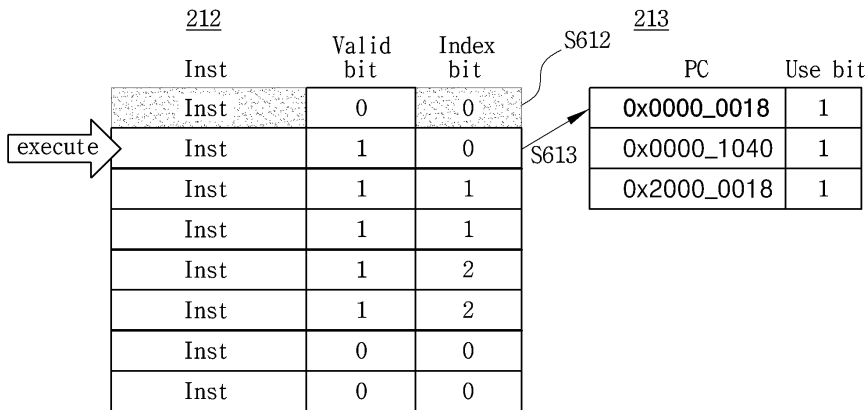
도면6a



도면6b



도면6c



도면6d

