(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년06월25일
(11) 등록번호 10-0904518
(24) 등록일자 2009년06월17일

(51) Int. Cl.
*G11C 29/00* (2006.01)
(21) 출원번호 10-2008-0000260
(22) 출원일자 2008년01월02일
심사청구일자 2008년01월02일
(56) 선행기술조사문헌
US20070094560 A1
Ergin, O. et al. "Exploiting Narrow Values for Soft Error Tolerance," IEEE Computer Artchitecture Letters, Vol. 5, 2006.
장형범 등, "공정 변이 조건 하의 수율 인식 캐시 설계를 위한 캐시 메모리 내로우 밸류 검색", 2008 한국컴퓨터종합학술대회 논문집, Vol.35, No.1(B), 2008. 6.
전체 청구항 수 : 총 13 항

(73) 특허권자
**고려대학교 산학협력단**
서울 성북구 안암동5가1 고려대학교 내
(72) 발명자
**정성우**
서울 용산구 이촌동 한가람아파트 217-1002
**김도연**
서울 강남구 대치4동 922-4
*(뒷면에 계속)*
(74) 대리인
**나승택, 조영현**

심사관 : 장호근

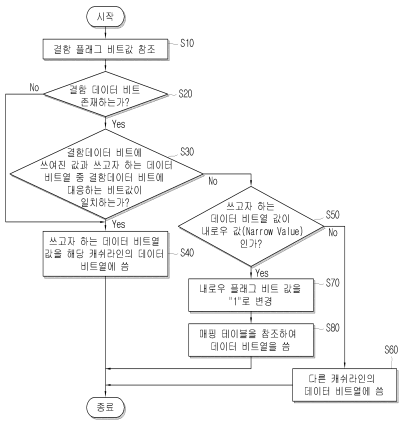
# (54) 결함 데이터 비트를 포함하는 캐쉬 메모리의 데이터비트열에 데이터를 쓰는 방법 및 데이터 비트열에서데이터를 읽는 방법

## (57) 요 약

본 발명은 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법 및 데이터 비트열에서 데이터를 읽는 방법에 관한 것으로, 특히 캐쉬 메모리에 결함 데이터 비트가 발생하더라도 결함 데이터 비트를 포함하는 캐쉬 라인을 그대로 활용함으로써, 캐쉬 메모리의 수율을 향상시킬 수 있는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법 및 데이터 비트열에서 데이터를 읽는 방법에 관한 것이다.

본 발명인 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법을 이루는 구성수단은, 해당 캐쉬 라인에 포함되는 결함(fault) 플래그(flag) 비트값을 참고하여 결함 데이터 비트가 존재하는지 판단하는 단계, 상기 판단 결과, 결함 데이터 비트가 존재하는 것으로 판단한 경우에, 상기 결함 데이터 비트에 쓰여진 값과 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트값이 일치하는지 판단하는 단계, 상기 판단 결과, 일치하면, 상기 쓰고자 하는 데이터 비트열 값을 상기 해당 캐쉬 라인의 데이터 비트열에 그대로 쓰고, 일치하지 않는 경우에는 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단한 후, 상기 해당 캐쉬 라인의 데이터 비트열 또는 다른 캐쉬 라인의 데이터 비트열에 쓰는 단계를 포함하여 이루어진 것을 특징으로 한다.

*대 표 도* - 도1



(72) 발명자

**이종성**

경기 의정부시 호원1동 우성아파트 201동 206호

**장형범**

서울 송파구 문정동 훼밀리아파트 304동 801호

**공준호**

서울 광진구 광장동 현대아파트 310동 1104호

**최진항**

강원 삼척시 남양동 백조맨션1차 101호

## *특허청구의 범위*

**청구항 1**

결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법에 있어서,

해당 캐쉬 라인에 포함되는 결함(fault) 플래그(flag) 비트값을 참고하여 결함 데이터 비트가 존재하는지 판단하는 단계;

상기 판단 결과, 결함 데이터 비트가 존재하는 것으로 판단한 경우에, 상기 결함 데이터 비트에 쓰여진 값과 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트값이 일치하는지 판단하는 단계;

판단 결과, 일치하면, 상기 쓰고자 하는 데이터 비트열 값을 상기 해당 캐쉬 라인의 데이터 비트열에 그대로 쓰고, 일치하지 않는 경우에는 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단한 후, 상기 해당 캐쉬 라인의 데이터 비트열 또는 다른 캐쉬 라인의 데이터 비트열에 쓰는 단계를 포함하여 이루어진 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법.

**청구항 2**

청구항 1에 있어서,

상기 결함 플래그 비트 값은 상기 캐쉬 메모리의 테스트 결과에 따라 세팅되는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법.

**청구항 3**

청구항 2에 있어서,

상기 결함 플래그 비트 값은 상기 캐쉬 메모리의 테스트 결과, 데이터 비트 열에 결함 비트가 포함된 경우에는 "1"로 고정 세팅되고, 결함 비트가 포함되지 않는 경우에는 "0"으로 세팅되는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법.

**청구항 4**

청구항 1에 있어서, 상기 결함 데이터 비트에 쓰여진 값과 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트값이 일치하지 않는 경우에는,

상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단하여, 내로우 값(narrow value)이 아닌 경우에는 다른 캐쉬 라인의 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓰고, 내로우 값(narrow value)인 경우에는 내로우 플래그 비트(narrow flag bit) 값을 매핑 테이블 참조 요구 값으로 변경한 후, 매핑 테이블(mapping table)을 참조하여 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓰는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법.

**청구항 5**

청구항 4에 있어서,

상기 매핑 테이블은 상기 캐쉬 메모리의 테스트 결과에 따라 형성되는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법.

**청구항 6**

청구항 5에 있어서,

상기 매핑 테이블은 상기 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 포함하는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법.

**청구항 7**

청구항 4에 있어서,

상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인 경우에, 상기 쓰고자 하는 데이터 비트열 값 중에 상기 결함 데이터 비트의 위치에 쓰여질 비트 값을 상기 해당 캐쉬 라인에 포함되는 데이터 비트열의 최상위 비트에 순서대로 쓰는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법.

**청구항 8**

결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법에 있어서,

해당 캐쉬 라인에 포함되는 결함(fault) 플래그(flag) 비트값을 참고하여 결함 데이터 비트가 존재하는지 판단하는 단계;

상기 판단 결과, 결함 데이터 비트가 존재하는 것으로 판단한 경우에, 내로우 플래그 비트(narrow flag bit) 값을 참고하여 매핑 테이블의 참조 유무를 확인하는 단계;

상기 판단 결과, 매핑 테이블을 참조하지 않아도 되는 경우에는 상기 데이터 비트열에 쓰여진 값을 그대로 읽고, 매핑 테이블을 참조해야 되는 경우에는 매핑 테이블을 참조하여 데이터 비트열을 읽는 단계를 포함하여 이루어진 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법.

**청구항 9**

청구항 8에 있어서,

상기 결함 플래그 비트값은 상기 캐쉬 메모리의 테스트 결과에 따라 세팅되는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법.

**청구항 10**

청구항 9에 있어서,

상기 결함 플래그 비트 값은 상기 캐쉬 메모리의 테스트 결과, 데이터 비트 열에 결함 비트가 포함된 경우에는 "1"로 고정 세팅되고, 결함 비트가 포함되지 않는 경우에는 "0"으로 세팅되는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법.

**청구항 11**

청구항 8에 있어서,

상기 내로우 플래그 비트(narrow flag bit) 값이 "1"인 경우에는 상기 매핑 테이블을 참조하고, "0"인 경우에는 상기 매핑 테이블을 참조하지 않는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법.

**청구항 12**

청구항 8에 있어서,

상기 매핑 테이블은 상기 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 포함하는 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법.

**청구항 13**

청구항 12에 있어서, 상기 매핑 테이블을 참조하여 데이터 비트열을 읽는 단계는,

상기 데이터 비트열의 주소를 참조하여 해당 데이터 비트열을 읽어들인 후,상기 결함 데이터 비트 개수와 결함 데이터 비트의 위치를 참조하여, 상기 결함 데이터 비트 개수만큼 최상위 비트로부터 비트값을 읽어서 상기 결

함 데이터 비트에 순서대로 채우는 단계인 것을 특징으로 하는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법.

# 명 세 서

## 발명의 상세한 설명

### 기 술 분 야

<1> 본 발명은 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법 및 데이터 비트열에서 데이터를 읽는 방법에 관한 것으로, 특히 캐쉬 메모리에 결함 데이터 비트가 발생하더라도 결함 데이터 비트를 포함하는 캐쉬 라인을 그대로 활용함으로써, 캐쉬 메모리의 수율을 향상시킬 수 있는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법 및 데이터 비트열에서 데이터를 읽는 방법에 관한 것이다.

### 배 경 기 술

<2> 반도체 기술이 발전하면서 칩 제조 기술이 나노 스케일까지 내려갔다. 칩에 들어가는 트랜지스터의 크기가 점점 작아짐에 따라서 공정 변이 문제가 대두되었다. 이전에 큰 스케일의 칩에서는 잘 발생하지 않았던 문제로서, 각각의 트랜지스터가 제조시에 약간씩 다른 특성(유효채널 길이, 산화층 두께 등)을 지니게 된다.

<3> 이 경우 캐쉬 메모리에서 많이 사용하는 6T SRAM 셀에서 6개의 트랜지스터가 각각 조금씩 다른 특성을 지니게 되면 두가지 문제가 발생할 수 있다. 즉, 액세스 타임(access time)을 맞추지 못하는 경우와 read-write를 못하게 되는 문제가 발생한다.

<4> 이와 같은 문제가 발생하면, 결함이 발생한 캐쉬 라인 전체를 못쓰게 되고, direct-mapped cache의 경우에는 system failure까지 야기할 수 있다. set-associative 캐쉬의 경우에도 way를 다 쓸 수 있는 것이 아니라서 성능상에 상당한 손해를 보게 된다.

<5> 결과적으로, 결함 발생으로 인하여 사용하지 못하는 캐쉬 라인이 발생하기 때문에 캐쉬 메모리의 수율이 떨어지는 문제점이 발생한다.

### 발명의 내용

#### 해결 하고자하는 과제

<6> 본 발명은 상기와 같은 종래 기술의 문제점을 해결하기 위하여 창안된 것으로, 캐쉬 메모리에 결함 데이터 비트가 발생하더라도 결함 데이터 비트를 포함하는 캐쉬 라인을 그대로 활용함으로써, 캐쉬 메모리의 수율을 향상시킬 수 있는 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법 및 데이터 비트열에서 데이터를 읽는 방법을 제공하는 것을 그 목적으로 한다.

#### 과제 해결수단

<7> 상기와 같은 과제를 해결하기 위하여 제안된 본 발명인 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법을 이루는 구성수단은, 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법에 있어서, 해당 캐쉬 라인에 포함되는 결함(fault) 플래그(flag) 비트값을 참고하여 결함 데이터 비트가 존재하는지 판단하는 단계, 상기 판단 결과, 결함 데이터 비트가 존재하는 것으로 판단한 경우에, 상기 결함 데이터 비트에 쓰여진 값과 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트값이 일치하는지 판단하는 단계, 판단 결과, 일치하면, 상기 쓰고자 하는 데이터 비트열 값을 상기 해당 캐쉬 라인의 데이터 비트열에 그대로 쓰고, 일치하지 않는 경우에는 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단한 후, 상기 해당 캐쉬 라인의 데이터 비트열 또는 다른 캐쉬 라인의 데이터 비트열에 쓰는 단계를 포함하여 이루어진 것을 특징으로 한다.

<8> 또한, 상기 결함 플래그 비트 값은 상기 캐쉬 메모리의 테스트 결과에 따라 세팅되는 것을 특징으로 한다. 여기서, 상기 결함 플래그 비트 값은 상기 캐쉬 메모리의 테스트 결과, 데이터 비트 열에 결함 비트가 포함된 경우에는 "1"로 고정 세팅되고, 결함 비트가 포함되지 않는 경우에는 "0"으로 세팅되는 것을 특징으로 한다.

<9> 또한, 상기 결함 데이터 비트에 쓰여진 값과 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대

응하는 비트값이 일치하지 않는 경우에는, 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단하여, 내로우 값(narrow value)이 아닌 경우에는 다른 캐쉬 라인의 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓰고, 내로우 값(narrow value)인 경우에는 내로우 플래그 비트(narrow flag bit) 값을 매핑 테이블 참조 요구 값으로 변경한 후, 매핑 테이블(mapping table)을 참조하여 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓰는 것을 특징으로 한다.

<10> 또한, 상기 매핑 테이블은 상기 캐쉬 메모리의 테스트 결과에 따라 형성되는 것을 특징으로 한다. 여기서, 상기 매핑 테이블은 상기 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 포함하는 것을 특징으로 한다.

<11> 또한, 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인 경우에, 상기 쓰고자 하는 데이터 비트열 값 중에 상기 결함 데이터 비트의 위치에 쓰여질 비트 값을 상기 해당 캐쉬 라인에 포함되는 데이터 비트열의 최상위 비트에 순서대로 쓰는 것을 특징으로 한다.

<12> 한편, 상기와 같은 과제를 해결하기 위하여 제안된 또 다른 본 발명인 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법을 이루는 구성수단은, 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법에 있어서, 해당 캐쉬 라인에 포함되는 결함(fault) 플래그(flag) 비트값을 참고하여 결함 데이터 비트가 존재하는지 판단하는 단계, 상기 판단 결과, 결함 데이터 비트가 존재하는 것으로 판단한 경우에, 내로우 플래그 비트(narrow flag bit) 값을 참고하여 매핑 테이블의 참조 유무를 확인하는 단계, 상기 판단 결과, 매핑 테이블을 참조하지 않아도 되는 경우에는 상기 데이터 비트열에 쓰여진 값을 그대로 읽고, 매핑 테이블을 참조해야 되는 경우에는 매핑 테이블을 참조하여 데이터 비트열을 읽는 단계를 포함하여 이루어진 것을 특징으로 한다.

<13> 또한, 상기 결함 플래그 비트값은 상기 캐쉬 메모리의 테스트 결과에 따라 세팅되는 것을 특징으로 한다. 여기서, 상기 결함 플래그 비트 값은 상기 캐쉬 메모리의 테스트 결과, 데이터 비트 열에 결함 비트가 포함된 경우에는 "1"로 고정 세팅되고, 결함 비트가 포함되지 않는 경우에는 "0"으로 세팅되는 것을 특징으로 한다.

<14> 또한, 상기 내로우 플래그 비트(narrow flag bit) 값이 "1"인 경우에는 상기 매핑 테이블을 참조하고, "0"인 경우에는 상기 매핑 테이블을 참조하지 않는 것을 특징으로 한다.

<15> 또한, 상기 매핑 테이블은 상기 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 포함하는 것을 특징으로 한다. 여기서, 상기 매핑 테이블을 참조하여 데이터 비트열을 읽는 단계는, 상기 데이터 비트열의 주소를 참조하여 해당 데이터 비트열을 읽어들인 후,상기 결함 데이터 비트 개수와 결함 데이터 비트의 위치를 참조하여, 상기 결함 데이터 비트 개수만큼 최상위 비트로부터 비트값을 읽어서 상기 결함 데이터 비트에 순서대로 채우는 단계인 것을 특징으로 한다.

## *효 과*

<16> 상기와 같은 과제 및 수단을 가지는 본 발명인 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법 및 데이터 비트열에서 데이터를 읽는 방법에 의하면, 캐쉬 메모리에 결함 데이터 비트가 발생하더라도 결함 데이터 비트를 포함하는 캐쉬 라인을 그대로 활용할 수 있기 때문에, 캐쉬 메모리의 수율을 향상시킬 수 있는 장점이 있다.

<17> 즉, 본 발명에 의하면 공정 변이 하에서 어느 정도의 결함이 캐쉬 메모리 셀에 발생하여도 신뢰성에 문제가 되지 않으며 칩을 그대로 쓸 수 있게 된다. 이로 인하여 결함 발생으로 종래에 버려지는 칩들이 활용 가능하게 됨으로써 수율이 극대화되는 장점이 있다.

## *발명의 실시를 위한 구체적인 내용*

<18> 이하, 첨부된 도면을 참조하여 상기와 같은 과제, 수단 및 효과를 가지는 본 발명인 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법 및 데이터 비트열에서 데이터를 읽는 방법에 관한 바람직한 실시예를 상세하게 설명한다.

<19> 도 1은 본 발명의 실시예에 따른 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법에 관한 절차도이다. 이를 참조하여 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법에 대하여 상세하게 설명한다.

<20> 먼저 캐쉬 메모리에 데이터들을 쓰기 위한 유닛(CPU)이 상기 캐쉬 메모리의 해당 캐쉬 라인에 포함되는 결함

(fault) 플래그(flag) 비트 값이 무엇인지 참고한다(s10). 상기 캐쉬 라인은 데이터가 쓰여지는 데이터 비트열과 특정 정보를 포함하고 있는 플래그 비트들을 포함한다.

<21> 상기 결함(fault) 플래그(flag) 비트는 본 발명에 적용되는 플래그 비트 중  하나이다. 상기 결함 플래그 비트는 해당 캐쉬 라인에 결함(fault)이 있는지에 대한 정보를 준다. 예를 들면, 상기 결함 플래그 비트 값이 "1"인 경우에는 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 비트가 포함됨을 의미하고, 상기 결함 플래그 비트 값이 "0"인 경우에는 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 비트가 포함되지 않음을 의미한다.

<22> 상기 결함 플래그 비트 값은 상기 캐쉬 메모리 칩을 완성한 후에 수행되는 케쉬 메모리의 테스트 결과에 따라 세팅되는 고정된 값이다. 즉, 캐쉬 메모리의 테스트 결과 캐쉬 라인에 포함되는 데이터 비트열에 결함(fault) 데이터 비트가 발생하면, 칩 제조자는 상기 결함 비트가 발생되는 캐쉬 라인에 포함되는 결함 플래그 비트 값을 "1"로 세팅하고, 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 발생하지 않으면 캐쉬 라인에 포함되는 결함 플래그 비트 값을 "0"으로 세팅한다.

<23> 상기 해당 캐쉬 라인에 포함되는 결함 플래그 비트 값을 참조한 결과, 결함 데이터 비트가 존재하는지를 판단한다(s20). 예를 들어, 결함 플래그 비트 값이 "1"인 경우에는 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 포함되는 것을 판단하고, "0"인 경우에는 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 포함되지 않는 것으로 판단한다. 상기 결함 데이터 비트는 상기 캐쉬 라인에 포함되는 데이터 비트열(데이터가 쓰여지는 비트열) 중에 결함이 발생한 비트를 의미한다.

<24> 상기 판단 결과, 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 포함되지 않는 경우로 판단되면, 해당 캐쉬 라인에 포함되는 데이터 비트열에 쓰고자 하는 데이터 비트열 값을 그대로 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 쓴다(s40).

<25> 만약 상기 판단 결과, 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 존재하는 경우로 판단되면(예를 들어, 상기 결함 플래그 비트 값이 "1"인 경우), 해당 캐쉬 라인에 쓰여진 데이터 비트열 중, 상기 결함 데이터 비트에 쓰여진 값과 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트의 값이 일치하는지를 판단한다(s30).

<26> 상기 결함 데이터 비트에 쓰여진 값이 무엇인지는 매핑 테이블을 참조하여 알아낼 수 있다. 즉, 매핑 테이블을 참조하여 결함 데이터 비트의 개수와 결함 데이터 비트의 위치를 알 수 있기 때문에, 상기 매핑 테이블을 참조하면 결함 데이터 비트에 쓰여진 값을 알아낼 수 있다.

<27> 상기와 같이, 상기 결함 데이터 비트의 위치(예를 들어, 캐쉬 라인의 데이터 비트열에서 하위비트부터 몇번째 위치한 비트인지에 대한 정보)와 쓰여진 값을 알아내면, 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트의 위치에 대응되는 비트의 값이 상기 결함 데이터 비트에 쓰여진 값과 일치하는지 판단할 수 있다.

<28> 예를 들어, 상기 캐쉬 라인에 포함되는 데이터 비트열에 쓰여진 값이 "0000 0000 0000 0011"이라 하고, 하위 두 비트가 결함 데이터 비트라 가정하자. 상기 하위 두 비트는 결함이 발생하여 비트 값이 모두 "1"로 고정된 상태이다. 이는 캐쉬 메모리 테스트 과정에서 확인할 수 있고, 칩 제조자는 결함 플래그 비트를 "1"(해당 데이터 비트열에 결함 데이터 비트가 존재한다는 정보)로 고정 세팅하고, 상기 데이터 비트열을 포함하는 캐쉬 라인의 주소, 상기 결함 데이터 비트 개수 및 결함 데이터 비트 위치에 대한 정보를 연결하여 매핑 테이블을 구성한다.

<29> 상기와 같은 상태에서, 상기 해당 캐쉬 라인의 데이터 비트열에 "1111"이라는 데이터 비트열 값을 쓰고자 하는 경우, 상기 해당 캐쉬 라인에 쓰여진 데이터 비트열 중, 상기 결함 데이터 비트에 쓰여진 값(두 하위 비트 "11" 값)과 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트(두 개의 하위 비트)의 값("11")이 일치하는지를 판단한다. 이 경우에는 마지막 두 하위 비트 값이 "11"로 일치하고 있다.

<30> 상기와 같은 판단 결과, 상기 해당 캐쉬 라인에 쓰여진 데이터 비트열 중, 상기 결함 데이터 비트에 쓰여진 값과 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트의 값이 일치한 경우에는 상기 쓰고자 하는 데이터 비트열 값을 상기 해당 캐쉬 라인의 데이터 비트열에 그대로 쓴다(s40).

<31> 이 경우에는, 결함에 의하여 특정 비트 값으로 고정된 비트값(상기 예에서 "11"로 고정)이 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 쓰여질 데이터 비트열의 값이 일치하기 때문에, 결과적으로 정확한 데이터를 쓰는 결과가 되기 때문에 상기 쓰고자 하는 데이터 비트열 값을 상기 해당 캐쉬 라인의 데이터 비트열에 그대로 쓰더라도 문제가 없다.

<32> 반면, 상기 판단 결과(s30 단계), 상기 해당 캐쉬 라인에 쓰여진 데이터 비트열 중, 상기 결함 데이터 비트에 쓰여진 값과 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트의 값이 일치하지 않는 경우에는 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단한다(s50).

<33> 여기서 내로우 값(narrow value)이란 상기 쓰고자 하는 데이터 비트열 값 중, 절반에 해당하는 상위 비트(예를 들면, 상기 데이터 비트열이 32비트인 경우 상위 16비트)가 "0"의 값을 가진 데이터 비트열의 값을 의미한다.

<34> 상기와 같이, 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단한 후에는 상기 해당 캐쉬 라인의 데이터 비트열(결함 데이터 비트를 포함하고 있는 데이터 비트열)에 쓰고자 하는 데이터 비트열 값을 쓰거나(s70, s80), 다른 캐쉬 라인의 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓴다(s60).

<35> 결국, 상기 결함 데이터 비트에 쓰여진 값과 상기 쓰고자 하는 데이터 비트열 중, 상기 결함 데이터 비트에 대응하는 비트값이 일치하지 않는 경우에는, 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단하여, 내로우 값(narrow value)이 아닌 경우에는 다른 캐쉬 라인의 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓴다(s60).

<36> 이 경우는 후술하겠지만, 상기 쓰고자 하는 데이터 비트열 값 중, 상기 결함 데이터 비트에 쓰여질 비트의 값을 해당 캐쉬 라인의 데이터 비트열의 상위 비트에 옮겨서 쓰는 경우, 정상적인 비트 값을 변경할 수 있기 때문에, 안정성을 고려하여 다른 캐쉬 라인의 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓴다.

<37> 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인지를 판단한 결과, 내로우 값(narrow value)인 경우에는 내로우 플래그 비트(narrow flag bit) 값을 매핑 테이블 참조 요구 값으로 변경한 후(예를 들어 내로우 플래그 비트 값이 "0"인 경우 "1"로 변경), 매핑 테이블(mapping table)을 참조하여 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 쓴다.

<38> 그런데, 여기서 중요한 사항은 상기 매핑 테이블을 참조하여 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 상기 쓰고자 하는 데이터 비트열 값을 그대로 쓰는 것이 아니라, 적절한 데이터 비트의 이동을 적용하여 쓴다.

<39> 구체적으로, 상기 쓰고자 하는 데이터 비트열 값이 내로우 값(narrow value)인 경우에, 상기 쓰고자 하는 데이터 비트열 값 중에 상기 결함 데이터 비트의 위치에 쓰여질 비트 값을 상기 해당 캐쉬 라인에 포함되는 데이터 비트열의 최상위 비트에 순서대로 쓴다.

<40> 도 2를 참조하여 설명하면, 다음과 같은 절차로 쓰고자 하는 데이터 비트열 값이 상기 해당 캐쉬 라인의 데이터 비트열에 쓰여진다. 도 2의 (a)는 결함이 발생한 캐쉬 라인의 데이터 비트열을 예시한다. 즉, 16 비트의 데이터 비트열 중, 마지막 두 개의 비트가 연속적으로 결함이 발생한 것을 예시한다. 이 결함 데이터 비트는 결함에 의하여 모두 "1"로 고정 값이 쓰여져 있다.

<41> 이 상태에서, 쓰고자 하는 데이터 비트열 값이 "1111"이라면, 상기 결함 데이터 비트에 쓰여진 값("11")과 상기 쓰고자 하는 데이터 비트열 중, 마지막 두 개이 비트 값("11")이 일치하기 때문에, 상기 쓰고자 하는 데이터 비트열 값을 상기 결함 데이터 비트를 포함한 데이터 비트열에 그대로 써도 무방하다.

<42> 그런데, 만약 쓰고자 하는 데이터 비트열 값이 도 2의 (b)에 도시된 바와 같다면, 상기 쓰고자 하는 데이터 비트열 값을 상기 결함 데이터 비트를 포함한 데이터 비트열에 그대로 쓰면 안된다. 왜냐하면, 상기 쓰고자 하는 데이터 비트열 값은 16비트 중 마지막 네 개의 비트 값이 "1100"의 값인데, 이를 상기 결함 데이터 비트를 포함한 데이터 비트열에 그대로 쓰게 되면, "1100"이 쓰여지는 것이 아니라 "1111"이 쓰여지게 되어서 정상적으로 데이터가 쓰여졌다고 할 수 없다.

<43> 따라서, 본 발명에서는 상기 쓰고자 하는 데이터 비트열 값 중, 상기 결함 데이터 비트에 쓰여지게 될 비트의 값은 상위 비트에 쓴다. 구체적으로 결함 데이터 비트에 쓰여질 비트 값을 상기 캐쉬 라인의 데이터 비트열에서 최상위 비트부터 순차적으로 쓰게된다. 예를 들어, 캐쉬 라인의 데이터 비트열 중, 마지막 두 개의 비트가 결함 데이터 비트이면, 상기 쓰고자 하는 데이터 비트열 중, 마지막 두 개의 비트의 값은 상기 캐쉬 라인의 데이터 비트열 중, 최상위 비트와 다음 상위 비트에 순차적으로 쓰여진다.

<44> 도 2의 (c)는 상기 쓰고자 하는 데이터 비트열의 마지막 두 개의 비트 값 "00"(도 2의 (b)에 도시된 마지막 두 개의 비트 값)을 상기 캐쉬 라인의 데이터 비트열 중, 최상위 비트와 다음 상위 비트에 쓰여진 것을 보여준다.

<45> 이 경우는 상기 쓰고자 하는 데이터 비트열 값이 도 2의 (b)에 도시된 바와 같이 내로우 값(narrow value)이기 때문에 적용 가능하다. 즉, 도 2의 (b)에 도시된 바와 같이, 절반의 상위 비트(최상위 비트로부터 8번째 까지)

가 모두 "0"의 값을 가지기 때문에, 상기 결함 데이터 비트에 쓰여질 비트 값을 상위 비트로 이동시켜 쓰게 되어도 무방하다.

<46> 여기서, 상술한 매핑 테이블은 상기 캐쉬 메모리의 테스트 결과에 따라 형성된다. 즉, 완성된 캐쉬 메모리 칩을 제조자가 테스트하여, 결함 데이터 비트가 발생한 경우에는 도 3에 도시된 바와 같이, 해당 캐쉬 라인의 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 연결하여 매핑 테이블을 구성한다.

<47> 상기 매핑 테이블은 상기 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 포함하기 때문에, 상기 쓰고자 하는 데이터 비트열 값 중, 상기 결함 데이터 비트에 쓰여질 비트 값을 상위 비트로 이동시켜 쓸 수 있게 된다. 즉, 매핑 테이블을 참조하여 결함 데이터 비트의 개수 및 위치를 알 수 있기 때문에, 상기 결함 데이터 비트에 쓰여질 비트 값을 상위 비트로 이동시켜 쓸 수 있게 되는 것이다.

<48> 상술한 사항은 결함 데이터 비트가 포함된 캐쉬 메모리의 캐쉬 라인에 소정의 쓰고자 하는 데이터 비트열 값을 쓰는 방법에 대하여 설명하였다. 이하에서는 도 4를 참조하여 결함 데이터 비트를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법에 대하여 설명한다.

<49> 먼저 캐쉬 메모리에 데이터들을 읽기 위한 유닛(CPU)이 상기 캐쉬 메모리의 해당 캐쉬 라인에 포함되는 결함(fault) 플래그(flag) 비트 값이 무엇인지 참고한다(s110). 상기 캐쉬 라인은 데이터가 쓰여지는 데이터 비트열과 특정 정보를 포함하고 있는 플래그 비트들을 포함한다.

<50> 상기 결함(fault) 플래그(flag) 비트는 본 발명에 적용되는 플래그 비트 중  하나이다. 상기 결함 플래그 비트는 해당 캐쉬 라인에 결함(fault)이 있는지에 대한 정보를 준다. 예를 들면, 상기 결함 플래그 비트 값이 "1"인 경우에는 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 비트가 포함됨을 의미하고, 상기 결함 플래그 비트 값이 "0"인 경우에는 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 비트가 포함되지 않음을 의미한다.

<51> 상기 결함 플래그 비트 값은 상기 캐쉬 메모리 칩을 완성한 후에 수행되는 케쉬 메모리의 테스트 결과에 따라 세팅되는 고정된 값이다. 즉, 캐쉬 메모리의 테스트 결과 캐쉬 라인에 포함되는 데이터 비트열에 결함(fault) 데이터 비트가 발생하면, 칩 제조자는 상기 결함 비트가 발생되는 캐쉬 라인에 포함되는 결함 플래그 비트 값을 "1"로 세팅하고, 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 발생하지 않으면 캐쉬 라인에 포함되는 결함 플래그 비트 값을 "0"으로 세팅한다.

<52> 상기 해당 캐쉬 라인에 포함되는 결함 플래그 비트 값을 참조한 결과, 결함 데이터 비트가 존재하는지를 판단한다(s120). 예를 들어, 결함 플래그 비트 값이 "1"인 경우에는 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 포함되는 것을 판단하고, "0"인 경우에는 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 포함되지 않는 것으로 판단한다. 상기 결함 데이터 비트는 상기 캐쉬 라인에 포함되는 데이터 비트열(데이터가 쓰여지는 비트열) 중에 결함이 발생한 비트를 의미한다.

<53> 상기 판단 결과, 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 포함되지 않는 경우로 판단되면, 해당 캐쉬 라인에 포함되는 데이터 비트열에 쓰여진 값을 그대로 읽어들인다(s150). 이 경우는 상기 데이터 비트열에 결함 데이터 비트가 포함되지 않기 때문에, 상기 해당 캐쉬 라인에 포함되는 데이터 비트열에 쓰여진 데이터는 정상적으로 쓰여진 데이터로 인정된다. 따라서 상기 데이터 비트열에 쓰여진 값을 값을 그대로 읽어도 문제가 발생하지 않는다.

<54> 반면, 상기 판단 결과(s120 단계에서의 판단 결과), 해당 캐쉬 라인에 포함되는 데이터 비트열에 결함 데이터 비트가 존재하는 것으로 판단한 경우에는 내로우 플래그 비트(narrow flag bit) 값을 참고하여 매핑 테이블을 참조 유무를 확인한다(s130, s140). 즉, 내로우 플래그 비트 값을 참고하여 매핑 테이블을 참조하지 않고 해당 캐쉬 라인에 포함되는 데이터 비트열에 쓰여진 값을 그대로 읽거나, 매핑 테이블을 참조하여 해당 캐쉬 라인에 포함되는 데이터 비트열에 쓰여진 값을 가공하여 읽어들인다.

<55> 상기 내로우 플래그 비트는 본 발명에서 적용되는 플래그 비트로서, 상기 내로우 플래그 비트(narrow flag bit) 값이 "1"인 경우에는 상기 매핑 테이블을 참조하고, "0"인 경우에는 상기 매핑 테이블을 참조하지 않는 것을 의미한다.

<56> 상술한 결함 데이터 비트를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법에서, 결함 데이터 비트가 존재하고(즉, 결함 플래그 비트가 "1"), 상기 결함 데이터 비트에 쓰여진 값과 쓰고자 하는 데이터 비트열

중, 상기 결함 데이터 비트에 대응하는 비트값이 일치한 경우(이 때, 내로우 플래그 비트는 "0"으로 유지 또는 변경)에는 쓰고자 하는 데이터 비트열 값을 해당 캐쉬 라인의 데이터 비트열에 그대로 쓰여지게 하였다. 따라서, 내로우 플래그 비트 값을 참고하여 매핑 테이블을 참조하지 않아도 된다면(내로우 플래그 비트 값이 "0"), 해당 캐쉬 라인에 포함되는 데이터 비트열에 쓰여진 값을 그대로 읽어도 데이터 변형이 발생하지 않아 무방하다(s150).

<57> 그러나, 매핑 테이블을 참조해야 하는 경우(내로우 플래그 비트 값이 "1")에는 매핑 테이블을 참조하여 상기 해당 캐쉬 라인의 데이터 비트열에 쓰여져 있는 데이터를 가공하여 읽어들인다(s160).

<58> 상기 매핑 테이블은 상기 캐쉬 메모리의 테스트 결과에 따라 형성된다. 즉, 완성된 캐쉬 메모리 칩을 제조자가 테스트하여, 결함 데이터 비트가 발생한 경우에는 도 3에 도시된 바와 같이, 해당 캐쉬 라인의 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 연결하여 매핑 테이블을 구성한다.

<59> 상기 매핑 테이블은 상기 데이터 비트열의 주소, 결함 데이터 비트의 개수 및 결함 데이터 비트의 위치에 대한 정보를 포함하기 때문에, 이 정보를 이용하여 상기 해당 캐쉬 라인의 데이터 비트열에 쓰여져 있는 데이터를 가공하여 정상적으로 읽어들일 수 있다.

<60> 상기 매핑 테이블을 참조하여 데이터 비트열을 읽는 단계는 상기 데이터 비트열의 주소를 참조하여 해당 데이터 비트열을 읽어들인 후,상기 결함 데이터 비트 개수와 결함 데이터 비트의 위치를 참조하여, 상기 결함 데이터 비트 개수만큼 최상위 비트로부터 비트값을 읽어서 상기 결함 데이터 비트에 순서대로 채우는 과정을 거친다.

<61> 구체적으로 설명하면, 이 경우에 상기 캐쉬 라인의 데이터 비트열에 쓰여진 데이터는 도 2의 (c)에 도시된 바와 같다. 즉, 마지막 두 개의 비트에 쓰여져야 할 비트 값("00")이 최상위 비트와 다음 상위 비트에 쓰여져 있는 상태이다. 이와 같이 쓰여져 있는 데이터를 정상적으로 복원하여 읽기 위해서는 매핑 테이블을 참조해야 한다.

<62> 즉, 매핑 테이블을 참조하면, 마지막 두 개의 비트가 결함 데이터 비트인 것을 알 수 있다. 따라서, 상기 캐쉬 라인의 데이터 비트열에 쓰여져 있는 데이터 비트열(예로서, 도 2의 (c)에 도시된 데이터 비트열)을 읽어들일 때는 매핑 테이블을 참조하여 결함 데이터 비트 개수만큼 최상위 비트로부터 비트값을 순서대로 읽어서 매핑 테이블을 참조하여 결함 데이터 비트에 순서대로 채워서 읽어들인다. 그러면, 원래의 정상적인 데이터 비트열을 완성할 수 있다.

## *도면의 간단한 설명*

<63> 도 1은 본 발명의 실시예에 따른 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법의 절차도이다.

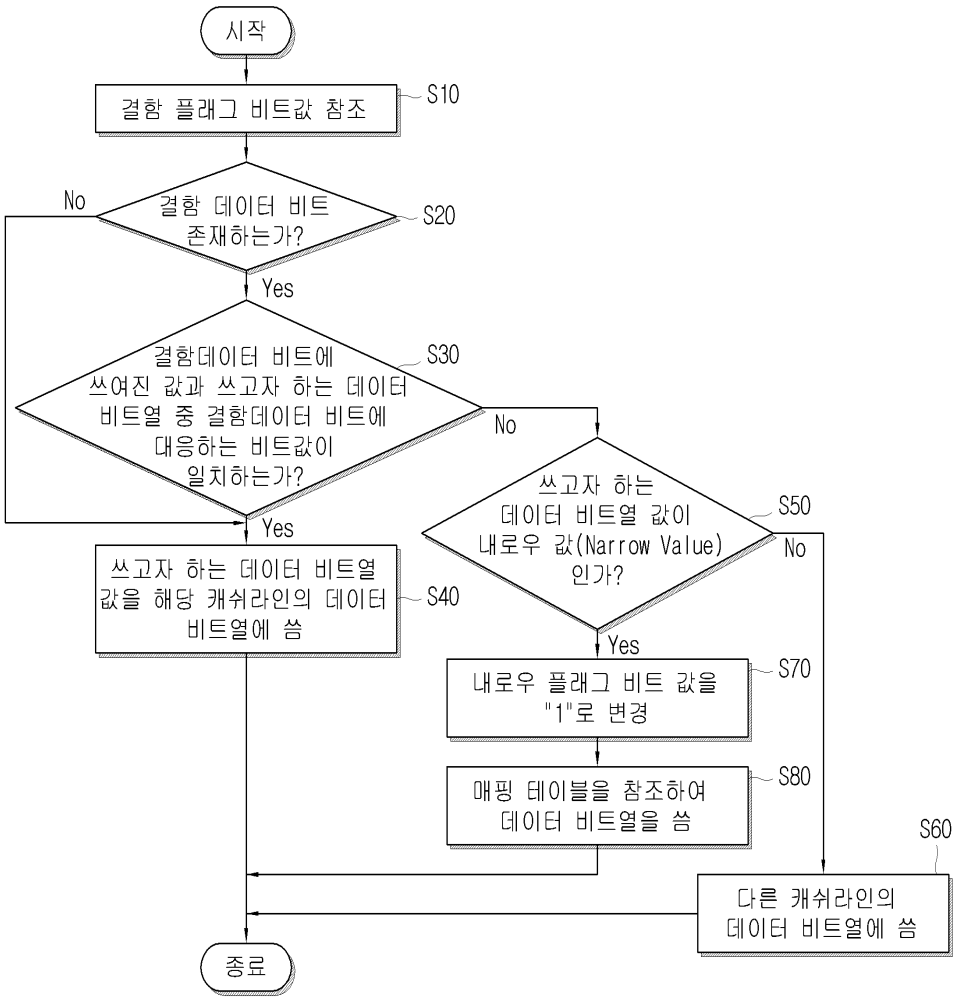
<64> 도 2는 본 발명의 실시예에 따른 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에 데이터를 쓰는 방법의 이해를 위한 데이터 비트열의 예시도이다.

<65> 도 3은 본 발명에 적용되는 매핑 테이블의 예시도이다.

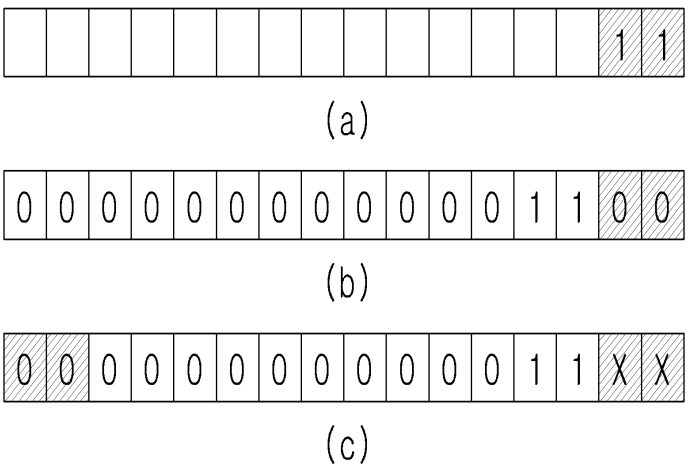
<66> 도 4는 본 발명의 또 다른 실시예에 따른 결함(fault) 데이터 비트(bit)를 포함하는 캐쉬 메모리의 데이터 비트열에서 데이터를 읽는 방법의 절차도이다.

## 도면

### 도면1


시작
결함 플래그 비트값 참조
S10
결함 데이터 비트 존재하는가?
S20
No
Yes
결함데이터 비트에 쓰여진 값과 쓰고자 하는 데이터 비트열 중 결함데이터 비트에 대응하는 비트값이 일치하는가?
S30
No
Yes
쓰고자 하는 데이터 비트열 값이 내로우 값(Narrow Value) 인가?
S50
No
Yes
쓰고자 하는 데이터 비트열 값을 해당 캐쉬라인의 데이터 비트열에 씀
S40
내로우 플래그 비트 값을 "1"로 변경
S70
매핑 테이블을 참조하여 데이터 비트열을 씀
S80
다른 캐쉬라인의 데이터 비트열에 씀
S60
종료

### 도면2


1 1
(a)
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
(b)
0 0 0 0 0 0 0 0 0 0 0 0 1 1 X X
(c)

도면3

| Address | # of faulty bits | Position of faulty cell |
|---|---|---|
| 0x1234abcd | 2 | 14 |
| : | : | : |
| : | : | : |

도면4

