효율적인 오류검출 방식의 Nand 플래시 컨트롤러

백정택* · 이용환

*김포공과대학교

A NAND Flash Controller with Efficient Error Detection Unit

Chung-taek Baik* · Yong-hwan Lee

*Kumoh National Institute of Technology
E-mail: ctaik@kumoh.ac.kr

요 약

NAND 플래시 메모리는 최근 많은 디지털 기기에서 사용되고 있으며 그 용량과 성능면에서의 발전이 급격히 이루어지고 있다. NAND 플래시 메모리는 잃고 쓰기 회수에 제한이 있어 이 수명이 다하면 데이터의 신뢰성을 보장하기 어렵다. 이 때문에 NAND 플래시 데이터의 오류를 검출하는 ECC(Error Correction Code) 알고리즘의 적용이 필수적인 시기이다. 기존에는 ECC 알고리즘을 논리 게이트로 구현하였으나 본 논문에서는 독립 테이블 방식을 사용하여 신뢰성과 데이터 처리 시간을 향상시키고자 한다.

ABSTRACT

Recently, NAND flash memory is widely used for digital equipments and its capacity and performance are rapidly improving. The limit on the number of writings and readings to/from NAND flash memory does not guarantee the integrity of its data. Therefore, ECC algorithm should be applied to the NAND flash controller. To reduce the access time, we use the look-up table to implement the ECC algorithm instead of the conventional logic gates.

키워드

Mobile, NAND Flash controller, SRAM interface

I. 서 론

최근 NAND 플래시 메모리의 휴대폰, 캡코더, 아이팟, 디지털카메라, USB 플래시 드라이브 등에 족 널리 사용이 확대되면서 수요가 급증하고 있다. 이러한 원인은 대용량 기기의 요전인 크기가 작고 가벼우며 외부충격에도 강하고 가격이 저렴하기 때문에 플래시 메모리가 모두 만족하기 때문이다. 반도체산업협회(SIA)의 보고에 따르면 NAND 플래시 메모리 판매는 계속성장하며 2009년에는 350달러까지 증가할 것으로 보고 있다.

또한 사용자의 데이터 저장 요구량의 증가로 메모리 용량 확대에 대한 연구가 활발히 진행 중이다. 2002년 국제반도체학회 학술회의(ISSCC)에서 삼성전자의 황광규 사장이 메모리 신성장론을 발표하였다. 그 논문은 반도체의 집적도가 2배로 증가하는 시간이 1년으로 단축되었으며 두의 법칙을 뒷받침 논리 있다고 한다. 2006년에는 DVD급 영화 20편을 저장할 수 있는 기술이 공개되었다. 2007년에는 32기가바이트 Nand 플래시 메모리를 개발하였다. 따라서 본 논문에서는 대용량의 NAND 플래시 메모리의 신뢰성과 데이터 처리 속도 향상을 위한 NAND 플래시 컨트롤러의 ECC(erro correcting code)를 설계한다.

논문의 구성은 다음과 같다. 2장에서는 NAND 플래시 메모리의 특성에 대해 알아보고 3장에서는 NAND 플래시 메모리 컨트롤러 구조에 관해 설명한다. 4장에서는 예명표 알고리즘을 5장에서는 독립 테이블 구조의 ECC에 대해 알아본다. 마지막으로 성능 분석 및 결론으로 구성된다.

II. NAND 플래시 메모리 특성

NAND 플래시 메모리는 전원이 깨져도 기록된 내용이 지워지지 않는 비휘발성 저장장치이며 제
조 단자가 쌓고 대용량의 저장이 가능하여 휴대형 전자기기의 저장매체로 주로 사용된다.
반면 쓰기 횟수가 100,000회로 제한되어 있으므로 제거의 일이거나가 불가능하며, 하나의 블록 내에서 데이터를 순차적으로 쓰여지는 계획이 있고, 린타임 메드블록(Run-Time Bad Block)이 발생할 경우 이를 처리해야 하며, 지우는 단위 프로그램의 단위가 다르다는 특성이다. 이러한 난드 플래시 메모리 사용하기 위해서는 호스트 컨트롤러(Flash Translation Layer)가 필요하며 이 컨트롤러는 각각의 플래시 메모리 소자 특성에 따라 다르다.

그림 4. 난드 플래시 메모리 구조

그림 1. 난드 플래시 메모리의 구조를 나타낸다. 난드 플래시 메모리의 한 페이지는 2,112바이트이며 한 블록은 64 페이지로 구성된다. 또한 읽기와 쓰기 명령을 받으면 개시 테이터와 데이터 해지스트를 통해서 페이지 단위로 실행하며 지우기 명령은 블록단위로 실행된다. 쓰기 명령은 우선 특별 데이터를 모두 지운 후에 쓰기 연산을 실행하므로 다소 시간이 걸린다.

그림 5. 난드 플래시 메모리의 한 페이지 구조

2,048바이트의 한 페이지 데이터 영역은 네 개의 512바이트 볼록으로 나눠진다. 각각의 볼록 데이터는 오류 검출을 위한 정보로 인코딩되어 난드 플래시의 2,101 - 2,112영역의 3바이트에 저장된다. 그리고 나머지 52바이트 영역(Spare Area)은 컨트롤러의 영역으로 사용한다.

III. 난드 플래시 메모리 컨트롤러 구조

본 논문에서 제공하는 난드 플래시 컨트롤러는 호스트 디비아스와의 인터페이스에서 일반 RAM처럼 간단한 메모리 메핑이 가능하도록 지원한다[2]. 그림 3.은 난드 플래시 컨트롤러 전체 블록도를 나타낸다[1]. 내부 구조는 데이터 페스 모듈과 에러 정정 모듈이 통합된 형태로 구성된다. 난드 플래시 메모리에 데이터를 쓰고 읽을 때 페이지 단위로 동작하기 때문에 임시로 데이터를 저장하는 버퍼가 필요하다. 데이터 페스 모듈에는 2,112바이트 버퍼가 2개가 있어서 호스트 시스템과 플래시 컨트롤러가 동시에 버퍼를 사용할 수 있다.

호스트에서 데이터를 쓰면 컨트롤러 내부에 있는 버퍼에 페이지 단위의 데이터를 쓴다. 컨트롤러는 난드 플래시 메모리의 쓰기 주소에 있는 데이터를 지우거나 다른 페이지로 옮기고 메모리에 쓰려면 할당된 메모리가 있는 버퍼를 선택한다. 버퍼의 데이터는 ECC 모듈을 통해 에러 검출을 위해 인코딩 정보를 생성한 후 데이터와 함께 난드 플래시 메모리에 페이지 단위로 저장된다.
반대로 데이터를 읽을 때는 난드 플래시의 범위의 데이터와 인코딩된 정보를 컨트롤러로 읽는다. 데이터는 ECC 모듈에서 에러 비트를 생성하며 그 페러정보는 기존에 쓰기 실행에서 인코딩된 정보와 비교하여 에러를 진단한다. 그러므로 에러가 있으면 정정하거나 데이터를 다시 요청하여 에러가 없으면 호스트가 사용하는 버퍼에 데이터 값을 저장한 후 SRAM 인터페이스 컨트롤러 신호에 따라 출력하게 된다.

IV. 난드 플래시 메모리에 ECC 해밍 코드 알고리즘

난드 플래시 메모리 한 페이지의 내부 512바이트 볼록에 적용되는 해밍코드 정보는 24바이트(3바이트)에 저장된다. 그림 4.는 볼록 페러티 비트 테이블을 보여준다.
그림 7. 블록 페어리 비트 테이블

P1은 512 비트에서 모든 1, 3, 5, 7 비트에 대한 Column 페어리 비트이고 P1'은 0, 2, 4, 6 비트에 대한 Column 페어리 비트를 나타낸다.

\[
\begin{align*}
p1' &= \text{bit6 xor bit4 xor bit2 xor bit0} \\
p1 &= \text{bit7 xor bit5 xor bit3 xor bit1} \\
p2' &= \text{bit5 xor bit4 xor bit1 xor bit0} \\
p2 &= \text{bit7 xor bit6 xor bit3 xor bit2}
\end{align*}
\]

P8'는 512 비트에서 모든 학수 열의 XOR 연산을 통하여 계산된다. Row 페어리 비트 (rowparity) 계산은 다음과 같다[2].

\[
\text{rowparity} = \text{bit7 xor bit6 xor bit5 xor bit4 xor bit3 xor bit2 xor bit1 xor bit0}
\]

\[
P8' = P8' \text{ xor (rowparity} \text{ xor !rownumber[0]})
\]

그림 8. ECC 인코딩 레지스터

그림 5는 각각의 인코딩 정보에 대한 페어리 비트를 계산하기 위한 24비트 레지스터이다.

V. 토큲 테이블 ECC 설계


결과 페어리(Result parity)를 생성할 때 XOR 트리로 사용하지 않고 Datain의 4비트를 사용하여 토큲 테이블 데이터를 출력하였다. 토큲 테이블은 16비트 두 개로 구성되어 서로 반전된 형태이다. 그래서 결과 페어리 값에 따라 토큲 테이블이 선택된다. 또한 Row의 토큲 테이블은 8비트에 대한 256 비트의 데이터를 선택적으로 사용하여
출력하였다. 독립 테이블을 사용하여 설계한 Encoding Data Generator 병목은 Verilog HDL로 기술되었으며 Modelsim을 이용하여 기능 검증을 하고 Synopsys와 0.35um CMOS 표준 셀 라이브러리를 이용하여 합성하였다. XOR 트리 구조로 설계된 것과 독립 테이블로 설계한 병목의 설 명적과 동작 속도를 비교하였다. 그 결과는 아래 표 1.과 같다.

<table>
<thead>
<tr>
<th></th>
<th>XOR</th>
<th>독립 테이블</th>
<th>성능</th>
</tr>
</thead>
<tbody>
<tr>
<td>섬 면적</td>
<td>1033.61</td>
<td>1744.53</td>
<td></td>
</tr>
<tr>
<td>동작속도</td>
<td>(7.6ns)</td>
<td>(5.6ns)</td>
<td></td>
</tr>
<tr>
<td>data arrival time(6.6)</td>
<td>data arrival time(4.64)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

표 1. XOR 트리 구조와 LUT 합성결과 비교

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Condition Name :</td>
</tr>
<tr>
<td>V300WTP0850</td>
</tr>
<tr>
<td>Process :1.30</td>
</tr>
<tr>
<td>Temperature :85.00</td>
</tr>
<tr>
<td>Voltage :3.00</td>
</tr>
</tbody>
</table>


※ 2007년도 IT-Soc 핵심설계인력양성 사업의 SoC 전공실습프로그램 지원에 대한 연구결과의 일부임.

VI. 결 론

본 논문에서 휴대형, 디지털카메라, USB 플래시드라이브 등과 같은 대용량의 메모리 요구하는 모바일 기기의 특성을 고려하여 동작 속도 증가를 위한 난드 플래시 컨트롤러 ECC 모듈을 독립 테이블로 설계하였다. 그리고 기존의 XOR 트리 구조와 지연시간, 면적을 비교한 결과 지연시간은 작아지고 면적은 증가하였다. 따라서 대용량의 난드 플래시 메모리의 고속처리를 위한 난드 플래시 메모리 컨트롤러로 사용이 적합하다.

참고문헌