

# 낸드 플래시 메모리 오류 정정을 위한 병렬 BCH 복호기의 최적 설계

최효진, 지현우, 성원용  
 서울대학교 전기공학부

## 요약

In this work, we have developed a parallel BCH decoder for multi-level cell NAND flash memory. The decoder is designed to require minimum chip area as well as minimum power consumption for NAND flash memory applications. To achieve this goal, the parallel factor of each functional block is determined by using design exploitation techniques.

## 1. 서론

멀티 레벨 셀(MLC) 낸드 플래시는 저장 용량을 증가시키기 위해 셀당 복수개의 비트를 저장한다. 그러나, 단일 레벨 셀(SLC)에 비해, 읽기 동작 시 레벨간 간섭에 의한 오류가 발생할 확률이 크다. 또한, 읽기/쓰기 동작이 반복될수록 오류 발생 확률이 크게 증가하여 제품의 신뢰도가 떨어지는 문제가 발생한다. 이러한 문제를 보완하기 위하여 오류정정부호를 사용한다. BCH 코드는 통신 시스템 및 저장 시스템 등에서 복수개의 오류를 정정하기 위한 알고리즘으로 널리 사용되고 있다. 이진 BCH 코드는 심볼 단위를 사용하는 Reed-Solomon 코드와 비교하였을 때, 코드율이 높은 코드를 설계할 수 있고, 보다 간단한 회로를 구현할 수 있는 장점이 있어 낸드 플래시를 위한 오류정정부호로 더 적합하다.

본 논문에서는 멀티 레벨 셀 낸드 플래시의 오류 정정을 위한 BCH 코드의 병렬 복호기를 구현하였다. 각 기능 블록의 병렬 구조를 구현하고, 주어진 처리시간을 만족하면서 최소의 칩 면적과 전력 소모를 갖는 각 블록별 병렬도를 찾는 최적 설계를 수행하였다.

## 2. 이진 BCH 코드 복호기의 병렬 구현

이진 BCH 코드의 복호과정은 오류 검출과 정정으로 이루어진다. 오류 검출은 입력된 코드워드로부터 생성한 신드롬이 모두 0인지를 검사하는 과정으로, CGFM (constant Galois field multiplier)를 사용하여 구현한다 [1]. 한편, 복호과정에 인코더를 사용하여 오류를 검출할 수 있다[2]. 이 방법은 오류 발생 확률이 낮은 경우, CGFM을 사용하여 신드롬을 생성하지 않고 오류를 검출할 수 있으므로 낮은 전력 소모를 보인다. 오류가 검출되면, 생성된 신드롬으로부터 오류위치방정식의 계수를 구한다. 이 과정은 BCH 복호과정에서 하드웨어 복잡도가 가장 큰 부분으로 주로 BM (Berlekamp-Massey) 알고리즘을 사용한다. BM 알고리즘은 복잡도를 줄이기 위한 여러 가지

변형이 있으며[3][4], 본 논문에서는 SRiBM (simplified reconfigured inversionless BM) 알고리즘을 사용하였다. 마지막 과정은 오류위치방정식의 근을 구하는 과정으로, Chien search를 사용하였다. Chien search는 코드워드의 각 비트 값을 오류위치방정식에 대입하여 근일 경우 해당 비트가 오류임을 결정하는 방식이다.

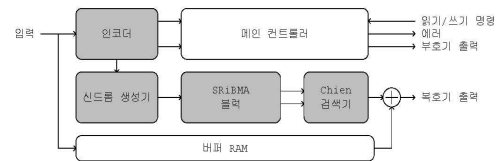


그림 1. BCH 복호기의 블록 다이어그램

BCH 복호기의 블록 다이어그램은 그림 1과 같다. 인코더는 p 비트 입력이 가능한 경우, 병렬도 p로 병렬화할 수 있다[5]. 병렬 신드롬 생성기와 Chien 검색기는 CGFM의 개수가 병렬도에 비례하여 증가한다[1]. 반면, SRiBMA 블록은 하드웨어 구현 복잡도가 크지만, 전체 복호 과정 중 매우 짧은 시간만 동작하므로, SRiBMA 블록에 사용된 GFM (Galois field multiplier)를 시분할로 공유하는 구조를 사용하였다. 최소로 사용할 수 있는 GFM의 개수는 2개이며, 이 때 SRiBMA 블록의 지연시간은 2t배가 된다(t는 오류정정능력)[4].

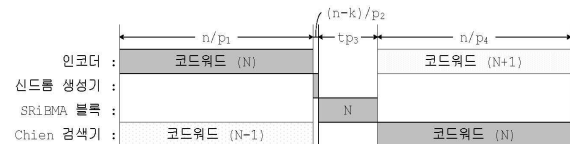


그림 2. 타이밍 다이어그램 (파이프라인)

각 블록의 병렬도를 p1, p2, 1/p3, p4 (SRiBMA 블록의 경우 시분할처리를 하였으므로 병렬도의 역수를 사용하였다)라 했을 때, 한 코드워드를 디코딩하는 과정을 타이밍 다이어그램으로 나타내면 그림 2와 같다. 연속된 코드워드에 대해 인코더와 Chien 검색기가 동시에 동작하는 파이프라인 구조를 가정하였다.

한 개의 코드워드를 디코딩하는 평균 처리 시간(sec)은 각 블록의 병렬도와 클럭 주파수(f<sub>clk</sub>), 코드워드의 길이(n), 메시지의 길이(k), 오류정정능력(t), 오류가 발생하지 않을 확률(P<sub>0</sub>)에 따라 다음과 같이 결정된다. 파이프라인에 의해 중첩된 시간은 p1과 p4 중, 작은 값에 의해 결정된다.

$$P_0 \frac{n}{\min\{p_1, p_4\}} + (1 - P_0) \left( \frac{n}{\min\{p_1, p_4\}} + \frac{n-k}{p_2} + tp_3 \right) / f_{clk} \quad (1)$$

## 3. 최적 설계

각 블록의 면적은 병렬도에 비례하여 증가하는 추세를

보인다. 그 이유는 필요한 레지스터의 개수와 XOR 게이트의 개수가 병렬도에 비례하여 증가하기 때문이다. 필요한 처리량(throughput)이 고정되어 있다고 가정할 때, 각 블록의 에너지 소모는 병렬도에 비례하여 감소한다. 따라서 다음과 같이 각 블록의 칩 면적과 에너지 소모를 병렬도에 대한 1차식으로 표현할 수 있다. Synopsys Design Compiler를 사용한 게이트 수준의 합성 결과와 PrimeTime을 사용하여 예측한 전력 소모 값으로 계수를 결정하였다(0.25 um 스탠다드 셀 라이브러리, 공급 전압:2.5V). 사용한 BCH 코드는  $n=4200$ ,  $k=4096$ ,  $t=8$ 이며,  $P_0 = 0.885$ 를 가정하였다.

	칩 면적 (mm <sup>2</sup> )	에너지 (uJ)
인코더	$0.0011 p_1 + 0.0185$	$0.385 / p_1 + 0.021$
신드롬생성기	$0.0378 p_2 + 0.0250$	$0.015 / p_2 + 0.002$
SRiBMA	$0.4663 / p_3 + 0.0621$	$0.001 p_3 + 0.006$
Chien검색기	$0.0193 p_4 + 0.0161$	$0.148 / p_4 + 0.239$

평균 처리 속도에 대한 요구사항이 주어졌을 경우, 최소의 칩 면적과 전력 소모를 갖는 각 블록의 최적 병렬도를 찾는 문제는 다음과 같은 다중 목적함수의 최적화 문제로 표현할 수 있다(a,c는 칩 면적과 에너지 수식의 1차계수의 벡터이며 b,d는 상수항의 합.  $t(x)$  : 평균처리시간.  $t_R$  : 처리시간 요구사항).

$$\begin{aligned} & \text{minimize } [\mu_1(x), \mu_2(x)] \\ & \text{s.t. } t(x) \leq t_R \\ & \mu_1(x) = ax + b, \mu_2(x) = (cx + d)/t(x), x = [p_1 p_2 1/p_3 p_4] \end{aligned} \quad (2)$$

이 문제의 해를 구하기 위해서,  $\mu(x) = \mu_1(x) + \gamma \cdot \mu_2(x)$  형태의 비용 함수를 정의하였다.  $\mu(x)$ 가 x에 대한 선형 함수가 아니므로, linear programming[6] 등의 방법을 사용할 수 없다. 대신,  $p_1=p_2=p_4=\{1,2,4,8,16,32,64\}$ ,  $p_3=\{1,2,8,16\}$ 에 대해 exhaustive search 방식으로 최적화를 수행하였다.  $\gamma$  값에 따른 칩 면적과 평균 전력 소모는 그림 3,4와 같다.

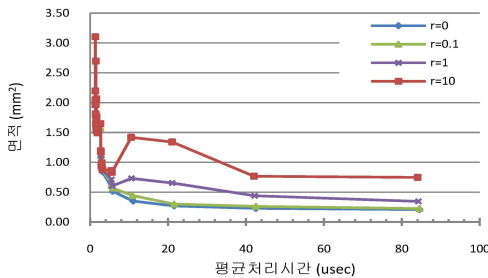


그림 3. 평균처리시간에 따른 최적 면적

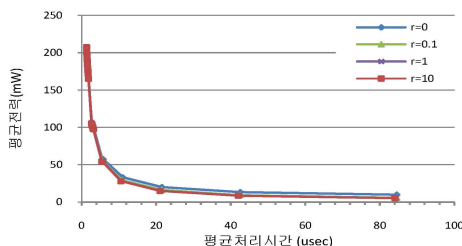
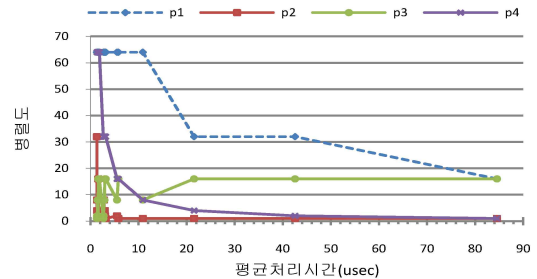


그림 4. 평균처리시간에 따른 최적 전력

$\gamma=0$ 인 경우, 칩 면적에 대한 최적화를 수행한다.  $\gamma=0.1$  일 때, 칩 면적이 0.5% 이내로 증가하면서, 평균전력 소모

는  $\gamma=10$ 인 경우와 유사한 결과를 얻었다.  $\gamma=0.1$ 인 경우의 각 블록의 병렬도는 그림 5와 같다. P0 값이 클 경우, 디코딩 과정은 오류 검출을 위해 인코더만 사용하므로, 평균 처리시간과 전력 소모는 인코더의 병렬도에 크게 영향을 받는다. 반면, 인코더의 면적은 전체 면적에서 차지하는 비중이 작기 때문에, 빠른 처리 시간이 요구되는 경우 병렬도가 높은 인코더를 설계해야 한다.



#### 4. 결론

본 논문에서는 병렬 BCH 복호기의 최적 설계를 연구하였다. 각 기능 블록별 칩 면적과 에너지 소모를 병렬도에 대한 함수로 근사하고, 최소의 칩 면적과 전력소모를 얻기 위한 병렬도를 구하는 최적 설계를 수행하였다.

본 논문은 지식경제부 출연금으로 ETRI, 시스템반도체 산업진흥센터에서 수행한 IT SoC 핵심설계인력양성사업의 연구결과입니다.

#### 참고 문헌

1. S. Lin, D. J. Costello, "Error Control Coding", Prentice-Hall, 1983
2. W. Liu, J. Rho, W. Sung, "Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories", SiPS, pp248, 2006
3. H. O. Burton, "Inversionless decoding of binary BCH codes", IEEE Trans. Inform. Theory, vol IT-17, pp464, 1971
4. D. V. Sarwate, N. R. Shanbhag, "High-speed architectures for Reed-Solomon decoders", IEEE Trans. on VLSI Syst., vol. 9, no. 5, pp641, 2001
5. J. Zhang, Z. Wang, Q. Hu, J. Xiao, "Optimized design for high-speed parallel BCH encoder", IEEE Int. Workshop on VLSI Design and Video Technology, pp97, 2005
6. S. Boyd and L. Vandenberghe, "Convex Optimization", Cambridge University Press, 2004