반도체디스플레이기술학회지 제19권 제3호(2020년 9월)
Journal of the Semiconductor & Display Technology, Vol. 19, No. 3. September 2020.

에러 보정 코드를 이용한 비동기용 대용량 메모리 모듈의 성능 향상

안재현*·양 오*†·연준상**
*†청주대학교 반도체공학과
**우진산전㈜

Performance Improvement of Asynchronous Mass Memory Module Using Error Correction Code

Jae Hyun Ahn*, Oh Yang*† and Jun Sang Yeon**
*†Semiconductor Engineering of Cheongju University
**WOOJIN Industrial System Co. Ltd.

ABSTRACT

NAND flash memory is a non-volatile memory that retains stored data even without power supply. Internal memory used as a data storage device and solid-state drive (SSD) is used in portable devices such as smartphones and digital cameras. However, NAND flash memory carries the risk of electric shock, which can cause errors during read/write operations, so use error correction codes to ensure reliability. It efficiently recovers bad block information, which is a defect in NAND flash memory. BBT (Bad Block Table) is configured to manage data to increase stability, and as a result of experimenting with the error correction code algorithm, the bit error rate per page unit of 4Mbytes memory was on average 0ppm, and 100ppm without error correction code. Through the error correction code algorithm, data stability and reliability can be improved.

Key Words : Asynchronous Mass Memory Module, PSRAM, NAND Flash Memory, Error Correction Code, Bad Block Table

1. 서 론

임베디드 컴퓨터시스템 환경에서 SRAM (Static Random Access Memory)과 DRAM(Dynamic Random Access Memory)은 코드나 데이터를 저장하는 보편적인 메모리다[1]. DRAM은 백만분의 몇 초마다 리플래쉬(Reflash)를 해 주어야만 내용이 유지되지만 SRAM은 리플래쉬 없이도 내용이 유지되기 때문에 SRAM이 DRAM보다 사용하기 편리하다는 장점이 있다. 하지만 회로가 복잡하여 집적도가 낮고, 값비싼 단점이 있다. PSRAM은 SRAM과 DRAM의 장점을 이용한 제품으로, DRAM의 단순한 구조와 재 기록 회로를 내장한 메모리이다. 사용자의 측면에서 보면 SRAM과 같은 동작을 하지만 내부에서는 DRAM 동작을 하는 구조로 되어있다. 또한 최근 주로 사용되는 플래시 메모리에는 NOR형과 NAND형 두 가지가 있다. 낸드 플래시 메모리는 전원을 공급하지 않아도 저장된 데이터를 유지하는 비휘발성(non-volatile) 메모리로서, 최근 모바일 장치들이 대용량화 하면서 시장이 급격하게 성장하고 있다[2]. 주로 USB메모리, SSD(solid-state-drive)등 저장 매체인 스마트폰과 디지털 카메라 등에 많이 사용되고 있다[3]. 하지만 낸드 플래시 메모리는 전기적 충격에 취약하여 이로 인한 읽기/쓰기 작업 도중에 오류가 발생할 수 있기 때문에 에러

†E-mail: ohyang@cju.ac.kr

보정 코드(error correction code)를 사용하여 신뢰성을 보장한다[4]. 데이터를 에러로부터 보호하기 위한 ECC 연산에는 LDPC(Low-Density Parity-Check) 코드, 터보(Turbo) 코드, BCH(Bose-Chaudhuri-Hocquenghem) 등 다양한 기법들이 존재한다[5,6].

본 논문에서는 낸드 플래시 메모리의 Bad Block정보를 효율적으로 관리하기 위해 BBT(bad block table)을 설계하였고, 비동기용 대용량 메모리 모듈의 전원이 OFF가 되면 PSRAM의 데이터와 외부 컨트롤러를 통해 에러 보정 코드를 낸드 플래시 메모리에 쓴다. 다시 전원을 ON되면 낸드 플래시 메모리의 데이터와 에러 보정 코드를 PSRAM으로 읽게 되고 에러 보정 코드 알고리즘을 이용하여 비교하고 맞으면 정상, 틀리면 에러가 난 비트를 반전시켜 정상데이터로 보정하는 알고리즘을 설계하여 비동기용 메모리 모듈의 신뢰성 및 안정성을 향상시킬 수 있다.

본 논문의 구성은 비동기용 대용량 메모리 모듈의 구조와 사용된 낸드 플래시 메모리의 구조를 기술하고, BBT(Bad block table)설계에 대해 설명한다. 또한 쓰기와 읽기의 데이터 신뢰성을 높이기 위한 에러 보정 코드 알고리즘을 제시하였다. 이후 본론에서 제시된 보정 코드 알고리즘의 실험결과를 확인한후 본 논문의 타당성을 제시하고자 한다.

2. 비동기용 대용량 메모리 모듈

2.1 낸드 플래시 메모리 구조

낸드 플래시 메모리는 대표적인 비휘발성 메모리 소자 중 하나로 반도체 집적 기술의 발전에 따라 면적대비 높은 용량을 지니고 있다. 상대적으로 저렴한 비용으로 많은 양의 데이터를 저장하는 MLC (Multi-Level Cell), TLC (Triple-Level Cell) 낸드 플래시 메모리가 각광받고 있다. 하나의 셀 당 각각 2비트, 3비트를 저장하고 대용량 저장장치를 구성한다. 하지만 하나의 셀 당1비트를 저장하는 SLC (Single-Level Cell) 낸드 플래시 메모리에 비해 낮은 내구성과 신뢰도를 보여준다[7]. 본 논문에서 제시한 낸드 플래시 메모리는 Micron사의 SLC (Single-Level Cell)을 사용한 1Gbytes, Page size는 2048bytes인 메모리를 사용하였고 Fig. 1과 같다. 메모리 구조는 Fig. 2와 같이 낸드 플래시 메모리 어레이, 바이트 또는 워드로 데이터를 전송하고, 데이터 레지스터 및 캐시 레지스터를 통해 페이지 기반 작업을 하여 낸드 플래시 메모리 어레이를 프로그래밍 한다.

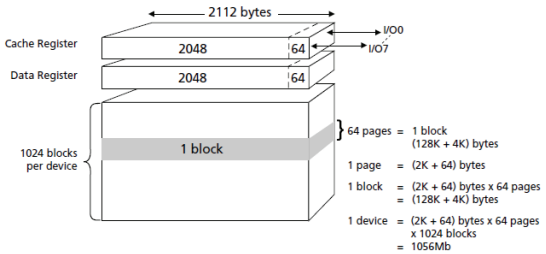


Fig. 1. NAND device and array organization.

이러한 낸드 플래시 메모리의 특성으로는 Serial 전송을 한다는 점과 Page 단위로 Read/Write를 한다는 점, 그리고 Bad Block을 가지고 있다는 점 등이 있다. 낸드 플래시 메모리의 경우 Bit Line을 한번에 묶어 큰 덩어리 단위로 전송하고 한 단위를 Page라고 한다. Spare영역은 보통 메타 데이터를 기록하기 위해 사용되며, 특히 Bad Block정보 및 에러 보정 코드 정보를 기록한다. Bad Block은 생산 중(Initial bad block)에 발생하기도 하고 사용 도중(Run-time bad block)에 발생하기도 한다[8].

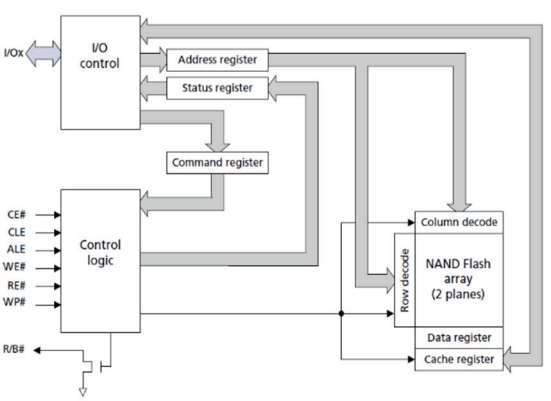


Fig. 2. NAND flash block diagram.

2.2 낸드 플래시 메모리의 BBT(Bad block table) 설계

초기 불량 블록(Early Bad Blocks)은 반도체 제조 공정 상의 오류로 인해 발생하며 블록의 특정 위치에 기록된 배드 블록 표시를 초기 배드 블록 여부를 판별한다. 사용 도중 불량 블록(Later Bad Blocks)은 반복적인 지우기 및 쓰기 연산으로 인한 셀의 마모로 인해 발생하며, 지우기/쓰기 연산이 실패하는 경우 동작 중 배드 블록으로 판별한다[9]. 저장 장치의 데이터 무결성을 위해 배드 블록이 사용되지 않도록 보장하는 기법이 필수적이다. 배드 블록 관리기법의 정확성을 쉽게 검증하고 FTL(Flash translation layer) 및 파일 시스템의 구현 복잡도를 줄이기 위해 독립

적인 소프트웨어 계층에서 배드 블록을 관리하는 방안이 제안되었다[10]. BBT(Bad block table)은 NAND의 Physical block number와 접근하려는 Logical block number를 연결해 주는 방식이다. Bad block table 구현은 난이도가 높지만 제대로 구현하면 Random Access및 안정성도 굉장히 높아지게 되고 메모리 관리의 효율성이 증가한다.

Table 1. NAND flash memory map

Block 번호	Page	Kind	Sub Kind	Total Byte
0 (64page) (64+2048byte)	0 (1032byte)	Bad block table (Total 1024block)	Bad_block_table[0]	1024byte
			Bad_block_table[1]	
			…	
			Bad_block_table [1023]	
		Bad block table's 1024byte에 대한 CRC	CRC32's LL	4byte
			CRC32's LH	
			CRC32's HL	
			CRC32's HH	
		PSRAM's 4Mbyte에 대한 CRC	CRC32's LL	4byte
			CRC32's LH	
			CRC32's HL	
			CRC32's HH	
	1 (512byte)	첫 번째 유효 Block (128word)	10	2byte
			11	
			…	
			137	
		두 번째 유효 Block (128word)	200	2byte
			201	
			…	
			227	
	2 (32,768byte =16page)	첫 번째 각페이지별 CRC Data (128×64×4byte =32Kbyte)	0×12345678	4byte
			…	
			0×ABCDEF01	
	18 (32,768byte =16page)	두 번째 각페이지별 CRC Data (128×64×4byte =32Kbyte)	0×8765431	4byte
			…	
			0×11223344	
200	1st RAM	PSRAM	0×6000_0000	4Mbyte
201			0×6000_0001	
202			0×6000_0002	
			…	
			0×607F_FFFF	

본 논문에 사용된 낸드 플래시 메모리의 BBT(Bad block table)의 구도 설계는 Table 1과 같다. 첫번째 Page에는 Bad block table은 1024bytes로 구성되어 있고, Bad block table에 대한 CRC32(cyclic redundancy check) 4bytes, PSRAM에 대한 CRC 4bytes로 구성되어 있다. 두번째 Page는 첫번째 유효 Block의 데이터, 두번째 유효 Block 데이터와 각각 Page별 CRC 데이터도 포함 되어있다. 기존 방식은 모듈의 전원을 끄게 되면 PSRAM의 데이터를 낸드 플래시 메모리의 유효 Block에 저장하고, 전원을 키게 되면 PSRAM에 저장되어 있는 데이터를 읽게 되는데 이때 데이터의 CRC를 확인하여 정상 CRC는 데이터를 읽고, 비정상 CRC는 Error를 띄워 그 이후의 대책이 없다.

본 논문에서는 기존 CRC를 확인하고 Error가 발생하였을때 대처할 수 없는 부분을 에러 보정 코드를 사용한 알고리즘을 이용하여 에러 보정 코드를 보정하고 Error가 발생해도 보정되어 데이터를 정상적으로 읽을 수 있게 한다.

2.3 비동기용 대용량 메모리 모듈의 구조

NVSRAM 모듈의 구성으로는 Fig. 3과 같다. ST사의 Cortex-M7 STM32H750 MPU와 ISSI사의 4Mbytes메모리인 PSRAM, Micron사의 1Gbytes용량의 낸드 플래시 메모리로 구성되어 있다. 모듈의 테스트를 위한 테스트 보드에는 ST사의 Cortex-M7 STM32F746 MCU와 외부 전압을 변환하는 Voltage Regulator로 구성되어 있다. 전원이 있을 때는 PSRAM에 Data를 저장하고, 모듈의 전원이 꺼지게 되면 슈퍼커패시터를 통해 낸드 플래시 메모리에 저장된다.

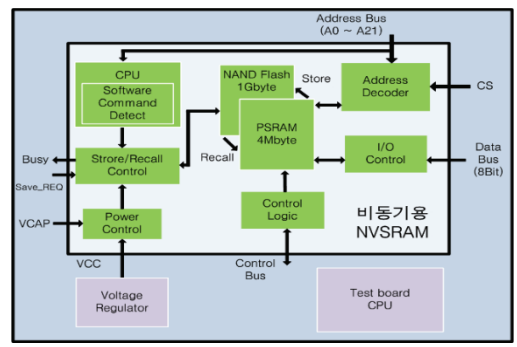


Fig. 3. NVSRAM module block diagram.

3. 에러 보정 코드 알고리즘 구현

에러 보정 코드는 낸드 플래시 메모리의 내부 컨트롤러에 존재하는 On-chip ECC와 낸드 플래시 메모리의 외부 컨트롤러를 사용하는 Off-chip ECC로 나뉠 수 있다. Off-chip ECC는 보드 레벨에서 신뢰성을 얻기 위해 사용되고, On-chip ECC는 MLC및 TLC에서 메모리 자체의 신뢰성 확보를 위해 자주 사용된다[11][12]. 본 논문에서 사용된 에러 보정 코드는 Off-chip ECC의 낸드 플래시 메모리를 사용하였고 내부에 에러 보정 코드 없이 CPU 내부의 에러 보정 코드를 사용하였다.

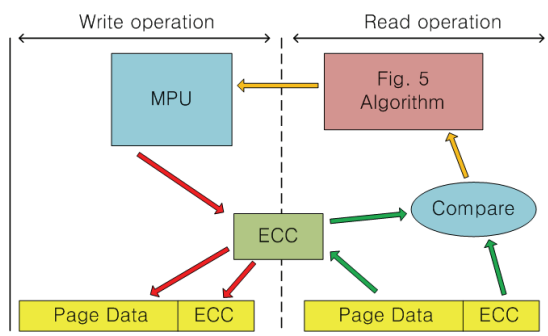


Fig. 4. ECC processing method of write and read operation.

본 논문에서는 기존에 데이터 오류를 탐지하기 위한 오류 탐지 코드로 CRC(Cyclic Redundancy Check)방식을 사용하였지만 제안된 방식은 Error가 발생되었을 때 대처가 불가능 하다는 점을 보안하였다[13]. 사용된 에러 보정 코드 알고리즘은 Fig. 4와 같이 모듈의 전원을 끌 때 PSRAM에 있는 데이터와 MPU에서 생성된 에러 보정 코드와 함께 낸드 플래시 메모리에 쓰게 되고, 모듈의 전원을 입력하게 되면 낸드 플래시 메모리의 데이터와 에러보정 코드를 PSRAM으로 읽게 되는데 이때 쓰고 읽을 때 에러 보정 코드를 비교하며, 낸드 플래시 메모리에 2048bytes의 데이터와 4bytes의 에러 보정 코드를 저장한다. 에러 보정 코드 알고리즘은 Fig. 5와 같이 쓰고/읽을 때 발생한 에러 보정 코드를 비교하여 에러가 발생된 Error bit를 찾고, 찾은 Error bit를 에러 보정 코드 알고리즘에 의해 정상 데이터로 보정하게 된다.

```
__ramfunc void ECC_data_recover( dword Result_ECC, dword rd_buf[])
{
    byte error_bit = 0;              //0~31
    word error_line = 0;             //0~511

    ////////////////// Error bit searching (dword is 31)
    if( Result_ECC & 0x02 )          /////// bit 0 of error position [ Max position is 31 (=0x1F) ]
        error_bit = 0x01;
    if( Result_ECC & 0x08 )          /////// bit 1 of error position
        error_bit |= 0x02;
    if( Result_ECC & 0x20 )          /////// bit 2 of error position
        error_bit |= 0x04;
    if( Result_ECC & 0x80 )          /////// bit 3 of error position
        error_bit |= 0x08;
    if( Result_ECC & 0x200 )         /////// bit 4 of error position
        error_bit |= 0x10;

    ////////////////// Error line searching
    if( Result_ECC & 0x800 )         /////// bit 0 of error line [ Max line is 511 (=0x1FF) ]
        error_line = 0x01;
    if( Result_ECC & 0x2000 )        /////// bit 1 of error line
        error_line |= 0x02;
    if( Result_ECC & 0x8000 )        /////// bit 2 of error line
        error_line |= 0x04;
    if( Result_ECC & 0x20000 )       /////// bit 3 of error line
        error_line |= 0x08;


    if( Result_ECC & 0x80000 )       /////// bit 4 of error line
        error_line |= 0x10;
    if( Result_ECC & 0x200000 )      /////// bit 5 of error line
        error_line |= 0x20;
    if( Result_ECC & 0x800000 )      /////// bit 6 of error line
        error_line |= 0x40;
    if( Result_ECC & 0x2000000 )     /////// bit 7 of error line
        error_line |= 0x80;
    if( Result_ECC & 0x8000000 )     /////// bit 8 of error line
        error_line |= 0x100;

    //Read data 보정
    error_line &= 0x1FF;             //max 512
    error_bit  &= 0x1F;              //max 31
    rd_buf[error_line] ^= (1<<error_bit);
}
```

Fig. 5. ECC algorithm for error correction code.

4. 실험 결과

비동기용 대용량 메모리 모듈을 Fig. 6과 같이 설계하였으며, 사용된 메모리는 4Mbytes PSRAM과 1Gbytes 낸드 플래시 메모리가 사용되었다. 낸드 플래시 메모리는 내부에서 에러 보정 코드를 컨트롤 할 수 없고, 외부 MPU를 통해 에러 보정 코드를 컨트롤 하였다. 또한 모듈의 전원이 꺼졌을 때 전원을 유지하도록 5V/0.47F인 슈퍼커패시터를 사용하였다.

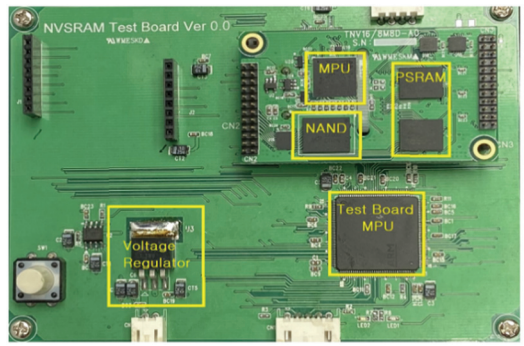
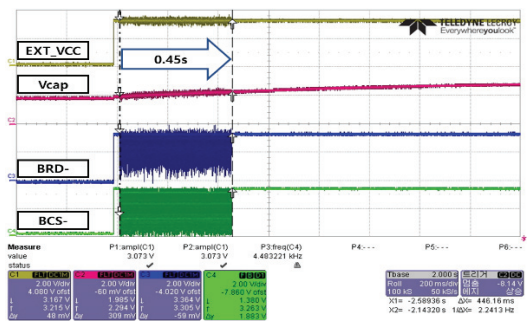
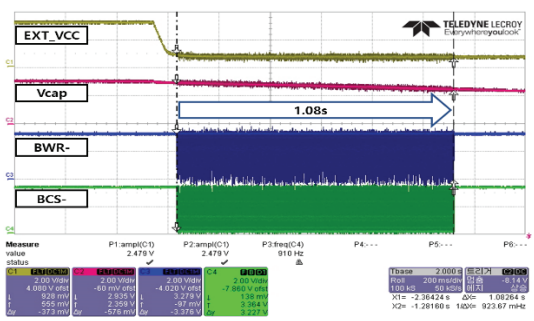


Fig. 6. Prototype for testing the ECC algorithm of NVSRAM.

모듈을 통해 낸드 플래시 메모리의 Bad block table을 설계하였으며, 에러 보정 코드 알고리즘을 통해 비동기용 대용량 메모리 모듈의 비트 에러율을 비교하였다.



(a) NVSRAM module at Power ON



(b) NVSRAM module at Power OFF

Fig. 7. Time of power ON/OFF.

Fig. 7은 모듈의 전원을 ON했을 때 낸드 플래시 메모리에서 PSRAM으로 읽을 때 약 0.45초가 걸리며 2048bytes의 데이터와 4bytes의 에러 보정 코드를 읽게 된다. 그리고 모듈의 전원을 OFF했을땐 전원이 바로 꺼지지 않고 슈퍼커패시터에 의해 PSRAM에서 낸드 플래시 메모리로 데이터

를 보내게 되며 약 1.08초가 경과되었다. 읽을 때와 같이 2048bytes의 데이터와 4bytes의 에러 보정 코드를 쓰게 된다. 읽고 쓸 때의 에러 보정 코드를 비교하여 비트 에러율을 확인하였으며, 에러 보정 코드 알고리즘을 사용하지 않았을 때 총3번의 실험 결과 page단위당 평균 100ppm의 에러율이 나왔다. 에러 보정 코드 알고리즘을 사용하였을 때는 page단위당 평균 0ppm의 에러율이 나오게 되면서 비트 에러 보정을 통한 데이터 손실이 줄어든 것을 확인할 수 있었다.

5. 결 론

낸드 플래시 메모리는 전기적 충격에 약하여 읽기/쓰기 작업 도중 오류가 발생할 수 있기 때문에 정확한 데이터를 송수신하기 위해서는 에러를 검출하는 과정이 필수적으로 수반되어야 한다[14]. 그래서 에러 보정 코드를 사용하여 신뢰성을 보장한다. 기존에 사용하던 데이터 에러 검사의 경우엔 CRC(cyclic redundancy check)를 사용하지만 Error가 발생했을 경우 대처가 불가능 하다는 점을 개선하기 위해 에러 보정 코드를 이용한 알고리즘을 설계하였다. 에러 보정 코드 알고리즘은 데이터 오류로 인한 ECC가 발생했을 때 ECC를 보정하여 오류를 줄이고 신뢰성을 높였다. 사용된 모듈은 낸드 플래시 메모리와 PSRAM을 사용한 비동기용 대용량 메모리 모듈이며, 내부 컨트롤이 되지 않는 낸드 플래시 메모리를 사용하여 외부 MPU를 이용한 컨트롤 로직을 설계하였다. 또한 낸드 플래시 메모리의 불량인 Bad Block정보를 효율적으로 관리하기 위해 BBT(bad block table)구성하여 안정성을 높였다. 에러 보정 코드 알고리즘을 설계하여 모듈이 켜지고 꺼질 때, 즉 읽기/쓰기 작업 도중 생길 수 있는 에러를 최소화였다. 에러 보정 코드 알고리즘을 이용해 실험 한 결과 4Mbytes 메모리의 page단위당 비트 에러율은 평균 0ppm의 결과가 나왔으며, 에러 보정 코드를 이용하지 않았을 시 100ppm의 비트 에러율과 비교하면 보정을 통한 데이터 안정성 및 신뢰도 증가를 확인할 수 있다. 기존 CRC방식과 비교하였을 때 시간은 조금 늘어 날수 있지만 데이터 손실에 대한 안정성과 신뢰도를 확보할 수 있다는 큰 장점이 있다.

감사의 글

본 연구는 2020년도 청주대학교 연구장학과 중소벤처기업부의 “산학연 Collabo R&D 예비연구사업(R&D, S2899935)”으로 지원받아 수행된 연구결과입니다.

참고문헌

1. Jungwon Kim, Seungkyun Kim, Jeajin Lee, Changhee Jung, Duk-Kyun Woo, “Memory Hierarchy Optimization in Embedded Systems using On-Chip SRAM”, Journal of KIISE: Computer Systems and Theory 36(2), pp. 102-110, 2009.4.
2. Myeong Kyun Kim, Oh Yang, Won Sup Chung, “Implementation of the FAT32 File System using PLC and CF Memory”, Journal of the Semiconductor & Display Technology, Vol. 11, No. 2. June 2012.
3. Lee Ki-jun, Lee Myung-Kyu, Shin Bum-kyu, Gong Joon-jin, “NAND flash memory storage device Error control code application”, The Journal of The Korean Institute of Communication Sciences 32(6), pp. 16-22, 2015.05.
4. Dong-Hwan Lee, Kwang-Hee Phark, Shao-hu Peng, Deok-Hwan Kim, “IRAW ECC for ensuring ECC Integrity of Flash memory”, The Journal of The Korean Institute of Communication Sciences 35(2B), pp. 451-454, 2008.10.
5. Jisu Kwon, Daejin Park, “Acceleration of ECC Computation for Robust Massive Data Reception under GPU-based Embedded Systems”, Journal of the Korea Institute of Information and Communication Engineering 24(7), pp. 956-962, 2020.7.
6. Jae-bin Lee, Geon-Myung Kim, Yi-Hyun Park, Seung-Ho Lim, “NAND Flash Memory Storage with Rate-Compatible LDPC”, Journal of Academic Presentation of the Korean Society of Information Sciences, pp. 1519-1521, 2019.06.
7. Yongju Song, Young Ik Eom, “Analysis of Hot/Cold Data Separation Scheme for Data Retention Error on NAND Flash Memory”, Journal of Academic Presentation of the Korean Society of Information Sciences, pp. 1395-1397, 2017.06.
8. Seong Ryeol Kim, “The Proposed of the Encryption Method and Designed of the Secure Key Using Initial Bad Block Information Physical Address of NAND Flash Memory”, Journal of the Korea Institute of Information and Communication Engineering 20(12), pp. 2282-2288, 2016.12.
9. Hongseok Kim, Ki Jun Kim, Sang Lyul Min, “Fast address translation of bad block management scheme in flash memory storage devices using bloom filters”, Journal of Academic Presentation of the Korean Society of Information Sciences, pp. 1053-1055, 2014.12.
10. Hongseok Kim, Eyee Hyun Nam, “Design and implementation of bad block management layer for high performance flash memory-based storage devices”, Journal of Academic Presentation of the Korean Society of Information Sciences 39(2A), pp. 90-92, 2012.11.
11. Stefano Gregori, Alessandro Cabrini, Osama Khouri,

Guido Torelli, “On-chip error correction techniques for new-generation flash memories,” Proc. of IEEE, Vol.91, No.4, pp. 602-616, 2003.
12. Wei Liu, Junrye Rho, Wonyong Sung, “Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories,” IEEE Workshop on Signal Processing Systems (SIPS), pp. 248-253, 2006.
13. Namgi Kim, “Variable CRC Scheme for Efficient Data Transmission Control for IEEE 802.16e Wireless Network”, The Journal of Korean Institute of Information Technology 7(3), pp. 109-115, 2009.6.
14. Oh Yang, Seungkyu Ock, “The Design of Multi-channel Synchronous Communication IC Using FPGA”, Journal of the Semiconductor & Display Technology, Vol. 10, No. 3. September 2011.

접수일: 2020년 9월 17일, 심사일: 2020년 9월 22일, 게재확정일: 2020년 9월 23일