國立交通大學

電機與控制工程研究所

博士論文

高速高容量固態硬碟機之控制晶片設計研究

On Study of Controller Chip Design for High-Speed and High-Capacity Solid-State Drives

研 究 生：林傳生

指導教授：董蘭榮　博士

中華民國九十六年七月

高速高容量固態硬碟機之控制晶片設計研究

On Study of Controller Chip Design for High-Speed and High-Capacity Solid-State Drives

研 究 生：林傳生　　Student: Chuan-Sheng Lin

指導教授：董蘭榮　　Advisor: Lan-Rong Dung

國立交通大學

電機與控制工程學系

博士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of The Requirements

for The Degree of

Doctor of Philosophy

in

Electrical and Control Engineering

July 2007

Hsinchu, Taiwan, Republic of China

中 華 民 國 九十六 年 七 月

國立交通大學

博碩士論文全文電子檔著作權授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學電機與控制工程系所 ______________組， 95 學年度第 二 學期取得博士學位之論文。

論文題目：高速高容量固態硬碟機之控制晶片設計研究
指導教授：董蘭榮

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：

本校及台灣聯合大學系統區域網路	■ 中華民國 98 年 7 月 31 日公開
校外網際網路	■ 中華民國 98 年 7 月 31 日公開

■ 全文電子檔送交國家圖書館

授 權 人：林傳生

親筆簽名： 林傳生

中華民國 96 **年** 7 **月** 31 **日**

國立交通大學

博碩士紙本論文著作權授權書

（提供授權人裝訂於全文電子檔授權書之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學電機與控制工程系所 ______ 組， 95 學年度第 二 學期取得博士學位之論文。

論文題目：高速高容量固態硬碟機之控制晶片設計研究
指導教授：董蘭榮

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：______________，請將論文延至____年____月____日再公開。

授 權 人：林傳生

親筆簽名： 林傳生

中華民國 96 **年** 7 **月** 31 **日**

國家圖書館
博碩士論文電子檔案上網授權書

（提供授權人裝訂於紙本論文本校授權書之後）

ID:GT008712801

本授權書所授權之論文為授權人在國立交通大學電機與控制工程系所 95 學年度第二學期取得博士學位之論文。

論文題目：高速高容量固態硬碟機之控制晶片設計研究
指導教授：董蘭榮

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

※ 讀者基於非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：林傳生

親筆簽名：林傳生

民國96**年**7**月**31**日**

推薦函

一、事由：推薦電機與控制工程系博士班研究生林傳生提出論文以參加國立交通大學博士論文口試。

二、說明：本校電機與控制工程系博士班研究生林傳生已完成博士班規定之學科及論文研究訓練。

有關學科部分，林君以修必應修學分（請查學籍資料），通過資格考試；有關論文方面，楊君已完成“高速高容量固態硬碟機之控制晶片設計研究”初稿。其論文“***Chuan-Sheng Lin*** and Lan-Rong Dung, "A NAND Flash Memory Controller for SD/MMC Flash Memory Card," IEEE Transactions on Magnetics, Vol. 43, No. 2, pp.933-935, February 2007.”, “***Chuan-Sheng Lin*** and Lan-Rong Dung, "A Dual-Mode USB Interface Controller Chip Design for Low-Power Mobile Devices," WSEAS Transactions on Circuits and Systems, Issue 3, Volume 6, pp.380-388, March 2007.” 等期刊發表。另有論文“A NAND Flash Memory Controller for SD/MMC Memory Card”, “A Multi-mode Error-Correction-Code Architecture for Hybrid Multi-channel NAND Flash Memory Storage Systems.” 分別發表於研討會或以投稿於期刊正接受審查中；有關專利部分，共已取得美國專利 8 篇，中華民國(台灣)專利 38 篇;相關書籍著作 3 本（請參閱博士論文著作目錄）。

三、總言之，林君已具備國立交通大學電機與控制工程系博士班研究生應有之教育及訓練水準，因此推薦林君參加國立交通大學電機與控制工程系博士論文口試。

此致

國立交通大學電機與控制工程學系

電機與控制工程學系教授

董 蘭 榮

中華民國 96 年 6 月

國立交通大學

論文口試委員會審定書

本校電機與控制工程學系博士班林傳生君

所提論文：高速高容量固態硬碟機之控制晶片設計研究

合於博士資格水準，業經本委員會評審認可。

口試委員：董蘭榮　李鎮宜

孫台平　陳明達

張隆國

張文平

指導教授：董蘭榮

系 主 任：邱俊誠

中華民國 96 年 7 月 27 日

Department of Electrical and Control Engineering
National Chiao Tung University
Hsinchu, Taiwan R.O.C.

Date: July 27, 2007

We have carefully read the dissertation entitled On Study of Controller Chip Design for High-Speed and High-Capacity Solid-State Drive submitted by Chuan-Sheng Lin in partial fulfillment of the requirements of the degree of **DOCTOR OF PHILOSOPHY** and recommend its acceptance.

Tai-Ping Sun

Li-Pin Chang

Thesis Advisor:

Chairman: Chiou, Jin-Chern

Acknowledgement

First of all, I would like to thank my adviser, Professor Lan-Rong Dung. Without his support, inspiration, and encouragement, there is no possibility for me to accomplish my research works during the past years, and this dissertation will not be fulfilled, either. Moreover, his guidance has led me a positive attitude as facing the difficulties at work.

To my dissertation committee members: Professor C.Y. Lee, Professor T.P. Sun, Professor L.K. Chang, Professor L.P. Chang, and Dr. M.D. Chen. Their helpful comments and advices have made the improvements and significant refinements for this presentation.

To the teachers in my education courses: Professor J.Y. Yen and Professor Y.Y. Chen, for helping me a lot during my studying for the mater degree in NTU; Professor Edge Yeh, Professor W.C. Wang, Professor C.K. Sung, for introducing the fundamental knowledge to me in NTHU. Mr. A.M. Liu, Mr. L.H. Lu, Mr. M.C. Lee, for teaching me in high school. Also, the teachers brought me the initiation in my elementary school.

To the supports from industrial: C.T. Chang, CEO of Prolific; Simon Chen, Chairman of A-Data; C.Y. Liu, Director of ICL/ITRI; Randall Yang, President of M-systems Asia-Pacific; Austin Chen, President of Apacer; C.L. Lee, President of InnoDisk; W.T. Liu, Chairman of Carry Computer; Gordon Yu, President of Pretec, C.S. Yang, President of S-Devices, Shimon Chen, Chairman of KTC; H.P. Lin, President of FTC; W.K. Deng, VP of PQI; Their great supports led me good connections with industrials for the practical developments.

To the supports from flash memory vendors: Y.J. Choi, D.G. Kim, Simon Lee, Ken Yeh and Joni Yang from Samsung Electronics; Akihito Nishikawa, Hideshi Tanimoto, J.J. Tsai and Brian Shen from Toshiba Semiconductor; C.S. Sohn from Hynix Semiconductor; Jim Cooke, Peter Feeley, Calvin Ger and Alvin Lin from Micron Semiconductor; Rick Durante, Jack Chen and Edward Chiu from Intel Corporation. Their great supports gave me rich information and more understandings on the characteristics and application handling of flash memory devices.

To the friends from industrial: Dr. Gibson Chen, Vincent Hsieh, Wayne Chang and Angela Chang, we discussed and shared the knowledge of the related patents on flash storage systems. Dr. Amy Chou of PSC, we had many constructive discussions on the interfacing between the controller and flash memory devices. K.Y. Chen, Kevin Lin, K.H. Wang, T.S. Cheng, Grace Lee, Faith Huang, Vic Hsieh, R.J. Dai, Jimmy Ke, Tim Chiu, Jerry Lai, C.S. Liu, C.J. Chang, Y.C. Chang, J.S. Pan, P.J. Wu, Y.S. Wang, C.H. Kuo, Y.C. Hung, W.J. Wu, Y.G. Lee, Michael Lee, and all of the colleagues we worked with, I would like to appreciate their supporting during my studying.

To the members of SoC LAB, thank you all for helping me as I need hands, and sharing with me the painful and joyful things during the past years.

Especially, the deepest gratefulness goes to my wife. Her love and patience enabled me to carry out this work. To my parents and my family, their love encourages me as facing challenges.

Chuan-Sheng Lin

Jhubei, Taiwan.
July, 2007.

On Study of Controller Chip Design for High-Speed and High-Capacity Solid-State Drives

高速高容量固態硬碟機之控制晶片設計研究

Student: Chuan-Sheng Lin

Advisor: Lan-Rong Dung

A Thesis
Submitted to Department of Electrical and Control Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Electrical and Control Engineering
July 2007
Hsinchu, Taiwan, Republic of China

Table of Content

List of Figures

List of Tables

Terms and Abbreviations:

3C products Computer, Communication and Consumer electronics products.

3DES Triple(3) DES data encryption/decryption mechanism.

4C Entity LLC An organization formed by the four companies: Intel, IBM, Matsushita, and Toshiba, for managing the keys of CPRM mechanism. Web-site: http://www.4centity.com/

AC characteristics Alternative Current characteristics.

AES Advanced Encryption Standard, a national standard for data encryption and decryption scheme in the United States.

AKE Authentication Keys Exchange.

AP Application Programs.

ATA AT Attachment interface, a kind of PC interface standard for connection with data storage devices.

BCH Bose-Chaudhuri-Hocquenghem, an ECC code named after the three mathematicians.

BER Bit Error Rate, it’s a metric for the error probability of each bit.

CCI Copy Control Information, a data structure defined in the CPRM specification.

CD	**C**ompact **D**isc.
CLK	Clock signal for synchronous circuits.
CPR	**C**ost, **P**erformance and **R**eliability.
CPRM	**C**ontent **P**rotection for **R**ecordable **M**edia, a content protection scheme proposed by 4C entity LLC (http://www.4centity.com/).
CPU	**C**entral **P**rocessing **U**nit.
CRC	**C**yclic **R**edundancy **C**heck.
DC characteristics	**D**irect **C**urrent characteristics.
DES	**D**ata **E**ncryption **S**tandard.
DMA	**D**irect **M**emory **A**ccess.
DOS	**D**isk **O**perating **S**ystems.
DRAM	**D**ynamic **R**andom **A**ccess **M**emory.
DSC	**D**igital **S**till **C**amera.
DV	**D**igital **V**ideo recorder or **D**igital **V**ideo camcorder.
DVD	**D**igital **V**ersatile **D**isc.
DW	A **D**ouble **W**ord is a 32-bit word.
ECC	**E**rror **C**orrection **C**ode

EEPROM **E**rasable **P**rogrammable **R**ead-**O**nly-**M**emory.

Endec **En**coder / **Dec**oder of BCH ECC circuit.

FAT **F**ile **A**llocation **T**able, a table to store the clusters linkage in file systems, e.g., FAT, FAT16 (16-bit FAT file system) and FAT32 (32-bit file system).

FBC **F**lash **B**lock **C**aching, a data caching algorithm used in flash memory management.

FC **F**ibre-**C**hannel, a kind of high speed serial interface for connections among computer systems and peripherals.

FDD **F**loppy **D**isk **D**rives.

FEC **F**orward **E**rror **C**orrection.

FeRAM, FRAM **Fe**rroelectric **RAM**, a kind of non-volatile solid-state memory.

FPGA **F**ield **P**rogrammable **G**ates **A**rray, a category of the programmable logic device.

FSM **F**inite **S**tate **M**achine.

GB **G**iga **B**ytes, 2^{30} bytes.

HDD **H**ard **D**isk **D**rives.

i-SCSI Internet type SCSI interface.

I/O **I**nput and / or **O**utput.

IA devices	Information Appliance or Industrial Application devices.
IEEE 1394	IEEE (Institute of Electrical and Electronics Engineers, Inc., http://www.ieee.org/) 1394 Standard, an interface standard for connections among consumer electronics.
IP	Intellectual Property.
IPC	Industrial Personal Computer.
ISP	In System Programmability.
IT	Information Technology.
L2P	Logical block address to Physical block address mapping.
LBA	Logical Block Address.
LDPC	Low-Density Parity-Check code, a kind of iterative error correction code.
MB	Mega Bytes, 2^{20} bytes.
MB/sec	Mega Bytes per second, a metric for the data transfer rate measurement.
MCU	Micro-Controller Unit.
MKB	Media Key Block, a data structure defined in the CPRM specification.
MLC	Multiple Level Cell structure of NAND flash memory cell.

MMC **M**ulti **M**edia **C**ard, a kind of interface standard for small form-factor flash memory card.

MP3 **MPEG** (Moving Picture Experts Group, web-site at: http://www.chiariglione.org/mpeg/) audio compression / decompression at layer 3.

MRAM **M**agneto-resistive **R**andom **A**ccess **M**emory.

NCQ **N**ative **C**ommand **Q**ueuing.

NOP **N**umber **O**f Page **P**rogram cycles.

ODD **O**ptical **D**isk **D**rives.

OS **O**perating **S**ystems.

PATA **P**arallel **AT** **A**ttachment interface.

PC **P**ersonal **C**omputer.

PCI **P**eripheral **C**omponent **I**nterconnection, a kind of interface standard for the interconnection of the computer systems and the peripheral functions or devices.

PCMCIA-ATA **P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation – **ATA** interface, or called PC Card ATA interface.

PC-RAM, PRAM **P**hase-**C**hange **RAM**.

POS **P**oint **O**f **S**ales, an electronic device to handling the purchasing and ordering information.

RAID	**R**edundant **A**rray of **I**nexpensive **D**isks.
RAM	**R**andom **A**ccess **M**emory.
RCC	**R**edundant **C**heck **C**ode, an error detection checking mechanism as doing the MKB update, which is defined in the CPRM specification.
RISC	**R**educed **I**nstruction **S**et **C**omputer.
ROM	**R**ead **O**nly **M**emory.
RS	**R**eed-**S**olomon code, a class of linear block code of ECC.
RSA	Stands for the names of the 3 inventors: **R**ivest, **S**hamir, and **A**dleman.
SATA	**S**erial **AT** **A**ttachment interface, a kind of interface standard for data storage device in PC.
SCSI	**S**mall **C**omputer **S**ystem **I**nterface, a kind of interface standard for devices connection among computers and peripheral.
S-SCSI	**S**erial **SCSI**.
SD	**S**ecure **D**igital memory card, an industrial standard for memory cards, especially for flash memory cards.
SLC	**S**ingle **L**evel **C**ell structure of NAND flash memory cell.
SRAM	**S**tatic **R**andom **A**ccess **M**emory.

SSD	**S**olid-**S**tate **D**rive, a data storage device which is composed by a controller and non-volatile solid-state memory devices.
SSFDC	**S**olid-**S**tate **F**loppy **D**isk **C**onsortium.
SSM	**S**olid-**S**tate **M**emory.
t-EC	***t*** bits error correction capability in a ECC circuit.
TD-based	**T**ransfer **D**escriptor based.
UFD	USB Flash Disks.
UMD	**U**SB **M**obile Hard **D**isk Drives.
UMPC	**U**ltra **M**obile **P**ersonal **C**omputers.
USB	Universal Serial Bus
USB 2.0 PHY or USB-PHY	Component/Unit implemented as a Physical Layer of USB 2.0
UTMI	**U**SB 2.0 **T**ransceiver **M**acro-cell **I**nterface.
VHDL	**V**ery **H**igh-speed IC hardware **D**escription **L**anguage.
VLSI	**V**ery **L**arge **S**cale **I**ntegrated circuits.
***w*-bit**	A parallel I/O bus with ***w*** bits of bus width.

List of Symbols:

Units:

A	Ampere, the metric for electrical current.
mA	Milli (10^{-3}) Ampere.
uA or µA	Micro (10^{-6}) Ampere.
V	Volts, the metric for electrical voltage.
mV	Milli (10^{-3}) Volts.
uV or µV	Micro (10^{-6}) Volts.
GB	Giga (10^{9}) bytes.
Gbits/s	Giga (10^{9}) bits per second.
MB	Mega (10^{6}) bytes.
MB/s or Mbytes/s	Mega (10^{6}) bytes per second.
sec or s	Second.
ms	Milli (10^{-3}) second.
us or µs	Micro (10^{-6}) second.
ns	Nano (10^{-9}) second.
MHz	Mega (10^{6}) Hertz.

US$	Amount of US dollars.
yr	year.

Galois Field Algebra and ECC:

$GF(2^m)$	Galois Field over 2^m.
α	Primitive element of the Galois Field $GF(2^m)$.
D_i	The input data stream for BCH ECC encoding / decoding.
G	The companion matrix for t-EC BCH ECC over $GF(2^m)$.
g	The coefficient vector of the generator polynomial over $GF(2^m)$.
$G(x)$	The generator polynomial over $GF(2^m)$.
$mod(a(x), b(x))$	Modulus operation for the polynomial $a(x)$ / $b(x)$.
$m_i(x)$	The minimal polynomial of order i over $GF(2^m)$.
$p(x)$	The primitive polynomial over $GF(2^m)$.
$S_i(x)$	The syndrome polynomial of order i over $GF(2^m)$.
SQR	Square operation over $GF(2^m)$.
$\sigma(x)$	The error locator polynomial over $GF(2^m)$..

$\boldsymbol{\sigma_i}$ The coefficients of the error locator polynomial.

$\oplus$ The exclusive or-ed operation or logic circuit (additive operation over ***GF(2)***).

Timing Parameters:

TR Data Transfer Rate.

$\boldsymbol{t_F}$ The time of flash write 512 bytes data from buffer RAM.

$\boldsymbol{t_{F\_DI}}$ The data input time of flash write a page data from buffer RAM.

$\boldsymbol{t_{F\_PP}}$ The page program time of flash write a page data.

$\boldsymbol{t_H}$ The time of host send 512 bytes data to buffer RAM.

中文摘要

本論文主要針對固態硬碟機在使用高密度NAND型快閃記憶體時，因NAND型快閃記憶體的製程高度微縮與多層次(Multi-Level Cell)記憶技術所造成的性能劣化現象，而所必需要的控制晶片設計，提出了設計方法與電路架構。尤其是特別針對在高速高容量的固態硬碟機的控制晶片設計而言。近期，由於NAND型快閃記憶體密度的快速提昇以及成本的下降，使得使用NAND型快閃記憶體的固態硬碟機來取代傳統的硬式磁碟機，以作為可攜式裝置的資料存儲設備成為極佳的選擇。然而，如果欠缺了一個有效的控制晶片來處理這些高容量快閃記憶體的性能劣化問題，以及將整體系統的性能提昇，則此固態硬碟機將會有使用上的限制，或者甚至有可能變成為無法使用的裝置。在本論文中，針對了先進NAND快閃記憶體因為密度提高與MLC (Multi-Level Cell)技術所造成的資料讀寫與區塊抹除的干擾因素增加，因而導致位元讀寫錯誤率的升高，提出了使用Systolic Array方法來構建 *t*-EC *w*-位元 並列式BCH (**B**ose-**C**haudhuri-**H**ocquengham) ECC (**E**rror-**C**orrection **C**ode)的硬體電路，來彌補這些性能劣化的現象，降低其位元讀寫錯誤率至一個可靠的水平。針對使用混合式多通道的快閃記憶體所構成的高速高容量固態硬碟機，提出了多模BCH ECC的電路架構，以提供一個高效率、低成本及低功率消耗的ECC電路架構。而對於對單一快閃記憶體晶片所能提供的資料讀寫速率極度有限，提出了高效能的資料傳輸硬體電路架構，以使主機系統與快閃記憶體陣列之間的資料傳輸效能達到系統所能提供的最高的頻寬利用率。針對固態硬碟機裡所存儲的資料保全機制，提出了具硬體加速器方式的CPRM (**C**ontent **P**rotection for **R**ecordable **M**edia)功能實施，以得到一個高效能且低成本的資料保全架構。在本論文中，我們提出混合式的多通道非揮發性固態記憶體所組成的固態硬碟機架構及其控制晶片設計，則具有能綜合各種非揮發性記憶體之間的特性，而能達到整體固態硬碟機的最佳的成本與性能。這些非揮發性的固態記憶體中，除了NAND快閃記憶體之外，其他有如：FeRAM (**Fe**rroelectric RAM), MRAM (**M**agneto-resistive RAM), PRAM (**P**hase-changed RAM)等。整體言之，本論文針對高速高容量固態硬碟機的控制晶片設計，提出了具體的電路設計方法與控制晶片的電路架構，使得

一個合於產品規格的低成本、高資料傳輸率、高品質、壽命長的高速高容量固態硬碟機能具體的實施出來。並且，經由實際的晶片實作與測試，驗證了這些方法與電路架構的有效性與優越的性能。

Abstract

In this thesis, we present the key architectures of the controller chip design for solid-state drive. The key architectures of the controller chip were developed to cover the deficiencies of NAND Flash memory and to enhance the system performance, especially for the high-speed and high-capacity solid-state drive. Nowadays, the continuous price drop of NAND flash memory makes the solid-state drives affordable and a promising candidate to replace the hard disk drives in portables. However, the solid-state drives may become inefficient or even useless without the help of an intelligent controller chip, not to mention the fulfillment of high-speed and high-capacity. In this thesis, both innovative controller designs and circuit architectures were presented for solid-state drives to overcome the NAND flash device degradation caused by technology shrinking and thus improve the system performance of solid-state drives. A *t*-EC *w*-bit parallel BCH (**B**ose-**C**haudhuri-**H**ocquengham) ECC (**E**rror-**C**orrection-**C**ode) construction was first proposed to ensure data correctness under inherent high bit error rate caused by severe disturbance and interference in advanced high-density NAND flash memories. Then a multi-mode BCH ECC architecture was proposed to achieve high efficiency, low cost and low power consumption design of hybrid multi-channel high-capacity solid-state drives. Not only innovative ECC schemes, but also efficient hardware architecture was also presented to guarantee maximum bandwidth utilization for data transmission between a host system and NAND flash memory arrays no matter how slow the read/program/erase speed of a single NAND device is. In system performance, the development of wear-leveling, data caching, and flash block redundancy cover the degraded program and erase endurance of the advanced high-density NAND flash memory. Moreover, a CPRM (**C**ontent **P**rotection for **R**ecordable **M**edia) implementation with associated hardware accelerators provides excellent data security but little extra overhead. Finally, hybrid non-volatile solid-state memory array architecture was proposed to provide the best

cost/performance by benefiting from the advantages of kinds of non-volatile semiconductor memories, such as FeRAM (**Fe**rroelectric **RAM**), MRAM (**M**agneto-resistive **RAM**), PRAM (**P**hase-change **RAM**) and high-density NAND flash memory. In summary, the proposed systematic design methodology together with the efficient controller architecture ensures that the cost, performance, reliability and lifetime of high-speed and high-capacity solid-state drives can meet the product specifications. The effectiveness and performance of the presented controller design are also proven through the chip implementation and experimental results.

Chapter 1 Overview

In this chapter, we outline the framework of this dissertation. Section 1.1, we summarize the topics we presented in this dissertation. Section 1.2, we take a historical review on the solid-state drive development. Section 1.3, we reviewed the related works regarding to the topics presented in this dissertation. Section 1.4, we pointed out the motivations and objectives for this dissertation. Section 1.5, we describe the organization of this dissertation.

1.1 Overview of this Dissertation

We developed the controller chips design for the high-speed high-capacity solid-state drive. The solid-state drive is a quite important I/O device for the mass data storage in the mobile computers. The more advanced controller chip architecture and design for the high-speed high-capacity solid-state drive is necessary for covering the deficiencies of the more advanced high-density NAND flash memory. Although the continuously rapid process technology shrinking and the novel MLC (**M**ulti-**L**evel **C**ell) technology has increased the bit density of NAND flash memory dramatically, meanwhile it generates the higher BER (**B**it **E**rror **R**ate) during the data accessing, much less program / erase endurance cycles, and more severe disturbances and interferences during the operation of reading, programming, and erasing of the NAND flash memory. The presented stronger controller architecture is required to overcome such difficulties or deficiencies as using the advanced high-density NAND flash memory to compose a high-speed and high-capacity solid-state drive. The chips implementations and experiments show the effectiveness and good performance of the controller architectures we presented.

In this dissertation, the key architectures of the controller chip design for high-speed high-capacity solid-state drive have been investigated and developed. We presented: the *t*-EC *w*-bit parallel BCH (**B**ose-**C**haudhuri-**H**ocquengham) ECC (**E**rror-**C**orrection **C**ode) construction for covering the higher bit error rate of the advanced high-density NAND Flash memory, the multi-mode BCH ECC for hybrid multi-channel flash memory storage system, which can provide the optimal cost-performance by combining the different characteristics among different kinds of flash memory devices, the high efficient hardware architecture to provide the guaranteed maximum bandwidth utilization of the host interface even though the single flash memory device accessing speed is quite limited, the hardware accelerator based CPRM (**C**ontent **P**rotection for **R**ecordable **M**edia) implementation to provide the content protection of the stored data in the solid-state drive, and the configurations of hybrid multi-channel non-volatile solid-state memory array and its controller architecture to gain more excellent performance and reliability as combining with the other types of non-volatile solid-state memory.

A *t*-EC *w*-bit parallel BCH (**B**ose-**C**haudhuri-**H**ocquengham) ECC (**E**rror **C**orrection **C**ode) was invented and constructed to cover the higher innate bit error rate of the advanced high-density NAND flash memory. The higher bit error rate requires stronger error correction capability of the BCH ECC circuit, while it increases the complexity of the BCH ECC circuit design. The inherent parallel page-wise accessing of NAND Flash memory requires the parallel data I/O of the data accessing. The general *w*-bit parallel provides the most flexibility as designing the ECC circuit for NAND flash memory. The systematic construction procedure of the *t*-EC *w*-bit parallel BCH ECC by using the systolic array architecture was presented, which help reduce the design cycle time and the efficiency of the ECC circuit.

To support the hybrid multi-channel flash memory storage systems, a multi-mode BCH ECC circuit architecture was constructed. The hybrid multi-channel Flash memory array was presented to provide the optimal cost-performance for a high-speed and high-capacity solid-state drive by using multiple types of NAND flash memory. The separated ECC encoder / decoder module and the error corrector module are suitable for the flexible data flow control as supporting the multi-channel flash memory storage systems. The presented multi-mode BCH ECC circuit architecture shows the lower cost and lower power consumption for the solid-state drive controller chips design.

The high efficient hardware architecture for data buffering and transmitting was presented to provide the guaranteed maximum bandwidth utilization for the data transportation between the host system and the flash memory array regardless the very restricted data accessing speed in reading, programming, and erasing of a single flash memory device. The hardware architecture for high efficient data transportation contains: the buffer RAM and the buffer manager, the TD–based (**T**ransfer **D**escriptor based) flash memory sequencer, and the hybrid multi-channel flash memory array. The buffer RAM and the buffer manager was established with the enough bandwidth to satisfy the bandwidth of host interface defined in the specification. The enough bandwidth of the buffer RAM and buffer manager was constructed by the efficient DMA (**D**irect **M**emory **A**ccess) mechanism and multi-buffering by sharing the buffer RAM in spatial or timing windowing. The TD-based flash sequencer is composed of TD buffer, TD processor, and the flash memory access controller. The TD-based flash sequencer was designed to do the high efficient data accessing of the NAND flash memory array. The hybrid multi-channel flash memory array architecture was presented and composed by multiple flash memory devices. The hybrid composition of flash memory array can provide the optimal cost-performance combination. For example, SLC (**S**ingle-**L**evel **C**ell) with higher performance and good endurance, MLC (**M**ulti-**L**evel **C**ell) with lower cost and higher density, an efficient management on them can have the optimal overall system level cost-performance. The multi-channel architecture of flash memory array provides the parallelism to increase the data accessing speed as at the flash memory side.

The hardware accelerator based architecture of the CPRM implementation provides an efficient data security of the content data stored in the solid-state drive. The CPRM was developed by the 4C entity LLC (http://www.4centity.com/) for an efficient digital content protection of digital data exchange among the digital information appliances [26-27]. The hardware accelerator based architecture can be used with a MCU (**M**icro-**C**ontroller **U**nit). The hardware accelerators are working to help the data processing as the micro-controller is doing the AKE (**A**uthentication **K**eys **E**xchange) process and the secure read / write process defined in the CPRM specification. The architecture provides the high efficiency, high flexibility, low cost, and low power consumption for the supporting of the CPRM functions.

Besides the high-density NAND flash memory, there are other kinds of non-volatile solid-state memory, which have more excellent accessing speed, higher endurances, and longer data retention time than the advanced high-density NAND flash memory, such as: FeRAM (**Fe**rroelectric **RAM**) [100], MRAM (**M**agneto-resistive **RAM**) [101], PRAM (**P**hase-changed **RAM**) [102]. Their excellence performance can be used to compensate the poor endurance, and long erase / program time of the high-density NAND flash memory by composing the hybrid non-volatile solid-state storage systems. The presented hybrid non-volatile solid-state memory array structure and its controller architecture attain the optimal cost-performance consequence by leveraging the different characteristics among different kinds of non-volatile solid-state memory.

Finally, we have the chips implementation and experiments on the flash memory controllers design. We illustrated implementations and experiments of 3 designed flash memory controllers to show the effectiveness and performance results of controller chip design architecture we presented. The three implemented flash memory controllers are: a NAND flash memory controller for SD/MMC card, a NAND flash memory controller of dual-mode USB flash card for low-power mobile devices, and a NAND flash memory controller for SATA II solid-state drive. The calculated performance of a **120GB** SATA II solid-state drive is summarized as: **230MB/sec** in reading, and **205MB/sec** in writing. In the ending of this dissertation, conclusions and future works discussions give us the directions for doing the next steps.

1.2 Historical Review of Solid-State Drive

The concept for developing a solid-state storage drive to replace the magnetic drive comes from long time ago. It was in 1971 when Dr. Matsuoka joined Toshiba [48]. Since then, it has been over 30 years. The dream still has not become a solid truth till today. The main reason is the high bit cost. Even though the solid-state storage is more user-friendly than the rotating magnetic disk drive, the much cheaper bit cost and higher density of hard disk drive still dominated the storage device for the past decades in the computer industry.

The mass data storage device is one of the necessary I/O devices in the computer systems. In the modern computer system architecture, CPU, memory and its peripherals compose the main frame of the computer systems. The mass data storage device is one of the necessary peripherals of the CPU. It stores the data required for system, such as the OS (**O**perating **S**ystems) kernel, device drivers, and the AP (**A**pplication **P**rograms) files. It also keeps the saved data by users. It provides a permanent, non-volatile memory for the data used in the computer system operation. If we make the analogy of a computer system to a human body, the storage device is like the memory function of our brain. Then, we can imagine how important of the data storage device is.

The data storage device development in the IT (**I**nformation **T**echnology) industry is never stopping, since the first computer, Apple II, was introduced in the world. In the very initial stage, the magnetic tape drive was adopted for the data storage device for the computer. The tape drive cannot support random data access and the accessing speed is very slow. It could not keep the steps with the rapid progressing of CPU technology. Then, the rotating-type disk drive structure was introduced to have the random data access capability by cross tracks seeking. There were many formats and types of FDD (**F**loppy **D**isk **D**rives) and HDD (**H**ard **D**isk **D**rives) used for the computer systems. As the IT industry entered into the multi-media era, the large capacity, removable, and low cost ODD (**O**ptical **D**isk **D**rives) were adopted into the computer systems. They are CD-ROM, CD-R, CD-RW, DVD-ROM, DVD-RW, DVD-RAM, etc. In the most popular of current notebook computer, a 2.5” 160GB HDD and a DVD-RAM drive are the standard equipments of storage devices.

The non-volatile solid-state memory was invented as the micro-electronic process technology introduced. It uses the keeping and identifying the change of the non-volatile states in physical or electrical properties to store the data bit is ‘0’ or ‘1’. The first type of the non-volatile solid-state memory is the EPROM (**E**rasable **P**rogrammable **R**ead-**O**nly-**M**emory). The EPROM can be electrically programmed, but it needs UV (Ultra-Violet) light exposure to do the erase [1]. The EPROM can be used for the computer micro-code storage, but it cannot meet the requirement or electrically erasable of the mass data storage device. The EEPROM was invented to support the electrical erasure function. It might be the first invented non-volatile solid-state memory for mass data storage device, and the world first solid-state drive was made then, even though it is very low speed and very small capacity. Eventually, it was not a successful storage device due to the un-acceptable high cost. The block-wise simultaneous erasure mechanism was invented to improve the speed of the EEPROM, it called as flash memory named by Ariizumi in 1984 [48]. In 1987, Toshiba introduced the NAND flash memory structure with the smaller area of a single memory cell. It was the first light for the opportunity of the non-volatile solid-state memory can be used for the mass data storage device. The rapid progressing in process technology shrunk and the invention of MLC (Multi-Level Cell) technology pushes the bits density from 16Mbits in 1995 to 32Gbits in 2007. The density of NAND flash memory has got 2000X (11 folds) progress in the past 12 years. It nearly doubles the density per year [46].

The non-volatile solid-state memory is quite excellent as comparing with the inescapable mechanical movements of HDD and ODD. The advantages are: low power consumption, low heat dissipation, compact in size, shock free, anti-vibration, no acoustic noise, etc. The characteristics of the non-volatile solid-state memory are quite amazing, but the high bit cost limited its application field being very narrow in the past decades, like in the industrial equipments, automatic machines, or aero-space crafts. However, through the 10 folded of the past 10 years progressing in NAND flash memory density, the bit cost of the NAND flash memory was at around US$5 for 1 Giga byte now. It looks a 32GB solid-state drive with the price less than US$200 is foreseeable now. It is expected to see the solid-state drive will start displacing HDD in notebook computers from next year, 2008.

The solid-state drive era is coming based on the minimum required capacity at an affordable cost. To achieve the good performance of the solid-state drive, an excellent architecture of the controller chip for high-speed high-capacity solid-state drive is very important. Besides the cost and the capacity, the solid-state drive needs good quality, high performance, and high reliability as well. The rapid development of the NAND flash memory has led widespread applications for the NAND flash memory based storage devices. Such as the flash memory cards and UFD (**U**SB **F**lash **D**isks), are widely used in the 3C (**C**omputer, **C**ommunication and **C**onsumer electronics) products. The shipment of flash memory cards and UFDs in 2007 are anticipated to be 660 millions of units, and 125 millions of units, respectively (Source: Gartner, IDC). The flash memory cards are used for the storage media of DSC, cellular phone, MP3 player, etc. The UFD has replaced the floppy disks and become the indispensable removable storage media for the data transfer among PC and notebook computers. However, the application for solid-state drive is still not booming. The reason is caused not only by the not enough capacity but also the relatively high cost per Mega byte. But from this year, the situation might be changed. The emerging multi-media applications show a strong demand for higher density and lower cost, and drive the continuous process technology shrinking and compel the MLC (**M**ulti-**L**evel **C**ell) technology of flash memory. Samsung Semiconductor Co. demonstrated their 32-Gb NAND flash memory in ISSCC 2007. Meanwhile, the cost per GB (**G**iga **B**yte) of NAND flash memory was declined to US$5.0 in the first quarter of this year. The improvement of the non-volatile solid-state memory will keep going via the huge investment of the world’s top semiconductor companies (e.g., Intel, Samsung, Micron, Toshiba, Hynix, etc.). Consequently, the nice-enough capacity and reasonable cost of NAND flash memory is convincing the solid-state drive era to be coming. On the other hand, the progressing on the density increasing and multi-level cell technology induced the unwanted degradation effects on NAND flash memory. For example, the higher bit error rate, degraded program and erase endurance, more severe disturbance and interference phenomena during reading, programming, and erasing, problems for long-time data retention, and the restricted data accessing speed in reading, programming, and erasing of a single flash memory device, etc. These deficiencies in the advanced high-density non-volatile solid-state memory need a stronger controller to resolve the problems. The stronger controller with the effective circuit architecture and efficient managing algorithm is compellingly compulsory to ensure the reliability, performance, and lifetime of the high-speed and high-capacity solid-state drive.

1.3 Related Works

The controller chip design for solid-state drive has been developed for over 30 years. The related works of solid-state drive are the ECC circuit design for storage systems, the VLSI circuit architecture, the data security mechanism for content protection, the parallelism of multiple storage devices array, and hybrid multi-channel non-volatile solid-state memory devices.

The FEC (**F**orward **E**rror **C**orrection) is usually demanded for ensuring reliable data transmission in noisy communication channels, while maintaining a good system performance. There have been already a lot of algebraic ECC codes proposed for FEC purpose, in which the cyclic attribute of the BCH (**B**ose-**C**haudhuri-**H**ocquenghem) code makes an easy adoption for hardware circuit implementation and has become a common choice in most applications [2-3]. Different from the BCH code is suitable for correcting random bit errors, the derivative RS (**R**eed-**S**olomon) code was developed for burst errors correction. Both codes are of similar mathematical fundamentals, while the RS code can be regarded as an extension of the BCH code to be a non-binary, symbol-base operation. Nowadays, many applications adopt BCH or RS code as a standard ECC (**E**rror **C**orrection **C**ode), for example, CD (**C**ompact **D**isc), DVD (**D**igital **V**ersatile **D**isc), HDD (**H**ard **D**isk **D**rive), ether-net, wireless communications, etc. The parallel data I/O bus (×8 or ×16) of a NAND flash chip thus requires a parallel I/O BCH ECC codec, applied the parallel CRC (**C**yclic **R**edundancy **C**heck) concept was proposed by Zukowski [62]. A longer encoder polynomial for BCH code is spontaneous to have a stronger bit error correction capability. Some high speed architectures were presented by K. K. Parhi, et al [63-64, 66]. Also, Jun Zhang et al presented an optimized architecture for the long parallel BCH encoder [65]. In addition to the long encoder issue, there were time and power consumption issues solved by Chien's Search for error location [66-67]. These prior researches provide a formulated basis for a parallel BCH ECC code and circuit implementation.

The VLSI circuit design for micro-electronic technology has also been developed since 1980s [5-8, 37]. The smaller feature size of the semiconductor technology, the more transistors can be put on in a single chip. Thus, it also pushes the progressing of VLSI design techniques. In the highly regular and iterative VLSI architecture, the systolic array processors were introduced for the architecture and timing design by projection. The systolic array architecture has been applied to RS encoders and decoders, and showed good performance [68-70]. The systematic design approach of a systolic array processor can make the circuit design easy for implementation and easy for pipelining to fit the system level design specifications.

The rapid progressing in technology shrinking and the development of MLC (**M**ulti-**L**evel **C**ell) technology of NAND flash memory have increased the bit density and lowered the cost per MB of flash memory. The bit density of NAND flash memory was doubled per 12 months in the past decade. Nevertheless, the higher density of the NAND flash memory causes the poor performances and less endurance. The cost per MB versus to the performance and reliability of NAND flash memory has become a trade-off. In order to recover the performances and endurances loss of the higher density of the NAND flash memory, there are different strategies introduced to cover the degradation of advanced high-density NAND Flash memory [1, 47-51].

In the data cryptography, there are publications presented for discussing the data encryption and decryption, and public key cryptography algorithm design [4, 25, 36, 91-93, 108]. For example, the popular cryptography mechanism as: DES (**D**ata **E**ncryption **S**tandard) algorithm, RSA (stands for the names of the 3 inventors: **R**ivest, **S**hamir, and **A**dleman) algorithm, and ECC (**E**lliptic **C**urve **C**ryptography) algorithm are developed to increase the difficulties to crack the encrypted data and decrease the probability of the secure data been destroyed. In the history of developing of cryptography, the public-key cryptography might be the most important discovery. Moreover, we can say it might be the only one revolution from the long years “classic” cryptography into the “modern” cryptography. From the classic cryptography development point of view, almost all of the cryptosystems are using the fundamental data scrambling, symbol replacement, and data harsh function. There was a famous data encryption code developed in IBM laboratory, called Lucifer algorithm, it became a USA national standard of data encryption codes, called as DES. Besides the algorithm development of cryptosystems, there are some hardware circuit implementation have been presented [94-96]. The high-speed VLSI circuit architecture has been discussed to satisfy the increasing of computation due to more secure functions required for the demands of data portability. To increase the generality and flexibility of the hardware based VLSI processor for cryptography, there are scalable design architecture presented [97-99]. The scalability of the VLSI cryptography process provides the programmability of the circuit as using by a micro-processor, or a micro-controller.

To increase the data accessing or processing speed, the parallelism of computer architecture was developed [38]. The efficient computer architecture of parallelism, for example the pipelining, parallel structure, or super-scalar, can be implemented by VLSI techniques for the controller chips design of solid-state drive. The high efficient hardware architecture design can provide the enough data transportation bandwidth during the data transfer between the host system and storage device. To increase the accessing speed and to provide data redundancy of the data storage devices, the RAID

(**R**edundant **A**rray of **I**nexpensive **D**isks) technology was introduced [35]. The RAID level-0 can boost the higher data transfer by data block stripping. The RAID level-1 protects the data stored in the storage system by mirroring the data. The RAID level-5 provides the more utilization rate of the data storage capacity to provide the data parity redundant protection by *n* storage devices with the ratio of *(n-1)/n*.

There are several kinds of solid-state memory used in the computer, communication, industrial devices, car electronics, and consumer electronics, besides the NAND flash memory. Memory devices can be divided as the volatile memory type and the non-volatile memory type. The volatile memory is using for data storage as the device is powered, while the non-volatile memory can keep the data even if the power is off. The major volatile memory types are the DRAM and SRAM. Besides the NAND flash memory as the storage media for the mass data storage devices, there are other kinds of non-volatile solid-state memory device which has different characteristics of the memory cell. Such as FeRAM (**Fe**rroelectric **RAM**, or called **FRAM**), which can offers a number of advantages, notably lower power usage, faster write speed and a much greater maximum number (exceeding 10^{16} for 3.3 V devices) of write-erase cycles [100], MRAM (**M**agneto-resistive **R**andom **A**ccess **M**emory) which are using the magnetic element for storing bits, thus can have the almost un-limited endurance cycles and write the data without do the erase in advance [101], and PC-RAM (**P**hase-**C**hange **RAM**, or called **PRAM**) with its fast performance and high endurance for write cycles and data retention life span [102]. The different characteristics among these non-volatile solid-state memory devices can be combined together and formed a hybrid solid-state storage system, which can provide an optimization of cost, performance, and reliability in the system level point of view.

The Sandisk Corp. (http://www.sadisk.com/) and M-systems are the pioneers in the flash storage devices. Now they are the same company since the December 2006, Sandisk acquired the company of M-Systems. In 2007, the solid-state drive related products are very fruitful, and the related technology discussions and conferences, papers, patents are growing significantly. There are more than 50 companies (e.g., Sandisk, Simple Tech, Silicon Systems, A-Data, PQI, etc.) have solid-state drive products. The “Solid State Disks (SSDs) - all technologies” web-site shows the open-site articles, products information, and test reports for the solid-state drives (http://www.storagesearch.com/ssd.html). According to the survey on world-wide web, the current solid-state drives shown are: all using the SLC types NAND flash memory with 1-EC or 4-EC ECC circuit; the current available capacity of the solid-state drive is in range from 32GB to 160 GB; the host interfaces adopted are: SATA (**S**erial ATA), PATA (**P**arallel ATA, or IDE), SCSI, S-SCSI (Serial SCSI), iSCSI (Internet SCSI), FC (**F**ibre-**C**hannel); the data transfer rate is from 40MB/sec to 180MB/sec depends on the application fields.

1.4 Motivation and Objective

Without an effective controller chip for covering the deficiencies of the NAND flash memory, the solid-state drive will become futile. Without effective ECC circuit, the high bit error rate will cause data integrity issue. The data loss in a storage device would be a disaster. Without the hardware architecture to provide a smooth and high-enough data transfer rate, the data storing and retrieving speeds of the solid-state drives will become unacceptable. Moreover, the data security function supported by the controller can provide the content protection of the data stored in the solid-state drives. The talented hybrid multi-channel flash memory storage system can provide a better cost-performance at the overall system level point of view. Thus, an effective controller design for the high-speed and high-capacity solid-state drive is particularly important. The controller chip is like the “heart” of the solid-state drive.

The major goal of our research works is to find the systematic design methods and the efficient architectures of controller design for high-speed and high-capacity solid-state drive. The systematic design methods can shorten the controller chips design cycle, and easier to upgrade to have more advanced features. The efficient architecture can cover the deficiencies of the NAND flash memory devices. Such that we can ensure the cost, performance, reliability, and lifetime of solid-state drive be achieved for satisfying the requirements in the product specification.

The controller chip design for the high-speed and high-capacity solid-state drive is essential. There are over 1000 solid-state storage related patents applied to the United States in 2006. Now, Taiwan is one of the major semiconductor memory production areas in the world. I believe we have the circumstances to build and even have the ability to dominate the solid-state drive industry. I hope through the academic and industrial research works on the controller chips design of solid-state drive can help Taiwan manufacturers to play the more important role in the solid-state drive industry. I would like to devote myself to try to fulfill the solid-state drive dream, even if I have little power and limited resources.

1.5 Organization of this Dissertation

In this dissertation, we have discussed the technology for high-speed and high-capacity solid-state drives, and presented the *t*-EC *w*-bit parallel BCH ECC construction by using systolic array, multi-mode ECC for hybrid multi-channel flash memory storage systems, advanced architecture for system performance enhancement, CPRM implementation via the hardware accelerator based architecture, the advantage of hybrid multi-channel non-volatile solid-state memory storage systems and its controller architecture. Finally, we show the effectiveness and the performance by the implementations of 3 flash memory controller chips design.

In chapter 2, we discussed the NAND flash memory controller design. The solid-state drive controller chip design is mainly started from the concepts of the established technology of storage systems. The NAND flash memory is the most successful non-volatile solid-state memory used for storage systems. Through the discussion of NAND flash memory controller design and some of the well-established magnetic mass storage device (Hard Disk Drive) technology, we can set up the core technology scenario of the controller chip design for high-speed and high-capacity solid-state drive.

In chapter 3, we presented the *t*-EC *w*-bit parallel BCH ECC construction by using systolic array. The BCH ECC is suitable for the data integrity enhancement of solid-state memory with its multiple random bits error correction. Especially, it consumes the minimal redundant bits required. Moreover, a systematic and efficient ECC codes construction method is shows potential, since the ECC in the flash memory controller chip is getting more important for the advanced MLC flash memory.

In chapter 4, we presented a multi-mode BCH ECC architecture for hybrid multi-channel flash memory. The multi-channel flash memory configuration is the way to have the high performance in high-capacity solid-state drive. The hybrid structure is constructed to have the best cost-performance combination by leveraging the different properties among different kinds of flash memory. The presented multi-mode ECC can support the low cost and low power consumption in keeping the system-level performance.

In chapter 5, we presented the hardware architecture design for the performance enhancement in high-capacity solid-state drive. The parallelism of flash memory configuration, high efficient buffer manager and the TD-based flash sequencer make the solid-state drive can assure the highest utilization of the host interface bandwidth which defined in the standard interface specification.

In chapter 6, we discussed the CPRM for digital content protection among the recording / playback devices and mass storage media or devices. The data security is getting more important as the digital data transportation is getting more convenient. We presented the hardware accelerator based architecture of CPRM for SD Card. The circuit was implemented and verified via the UMC 0.18um process technology.

In chapter 7, we presented the advantage of hybrid multi-channel non-volatile solid-state memory storage systems. The optimal balance among the multiple types of the solid-state memory can have the best CPR (**C**ost, **P**erformance and **R**eliability) index. The supporting for multiple types of non-volatile solid-state memory was discussed. Architecture of the controller chip for the hybrid multi-channel non-volatile solid-state memory storage systems was proposed and simulated.

In chapter 8, we have the controller chips implementations of this dissertation. We have shown 3 flash memory controllers design: a NAND flash memory controller for SD/MMC card; a NAND flash memory controller of dual-mode USB flash card for low-power mobile devices; a NAND flash memory controller for SATA solid-state drive. The chip architecture, simulation, implementation and test results show the realizations of the topics we discussed in this dissertation.

Finally, we have the suggestions for future works and research in chapter 9.

Chapter 2 NAND Flash Memory Controller

In this chapter, we summary the related technology of the NAND flash memory controller design. Section 2.1 is the overview of the NAND flash memory based storage device. In section 2.2, the host interface was discussed. In section 2.3, the property of NAND flash memory was described briefly. In section 2.4, the ECC for NAND flash memory storage was discussed. In section 2.5, the NAND flash memory management was described. In section 2.6, the configurations of multiple flash memory chips were discussed. In section 2.7, the data security issue was discussed. In section 2.8, the system level point of view was discussed. In section 2.9, we summarize the importance of each functional units of the NAND flash memory controller with respect to different application conditions.

2.1 Introduction

To manage and control the NAND flash memory array, a NAND flash memory controller is required for a NAND flash memory storage system. In Figure 2-1, there is a flash storage system connect with a host system though an interface bus. The flash storage system can be composed of 2 major functional portions: the flash controller and the flash memory array. The flash controller is used to bridge the host interface controller to access the data stored or to be stored in the flash memory array. The flash memory array is used to record the data sent from the host system. The CPU in the host system can use the flash memory array as a mass storage I/O device.

In Figure 2-2, a typical functional block diagram of the flash memory controller is shown. The flash sequencer circuit is designed to control the NAND type flash memory; the command protocols and data packets of the NAND flash memory are controlled by the flash sequencer. The ECC (**E**rror-**C**orrection **C**ode) circuit is designed to enhance the data integrity of the flash memory storage system; the data integrity is enhanced by doing ECC encoding in the flash memory data writing, and doing ECC decoding and error correction in the flash memory data reading. The buffer RAM and the buffer manager is designed for the data transportation between the host interface bus and flash memory bus. The micro-controller unit (MCU) and the RAM, ROM, and system firmware control the overall system level execution. The interface bus and interface controller are used to connect with a host system, such as: PC, DSC, cellular phone, notebook computer, etc. The interface controller executing the host issued commands by decoding and interpreting the commands and protocols defined by the Interface Bus Specifications.

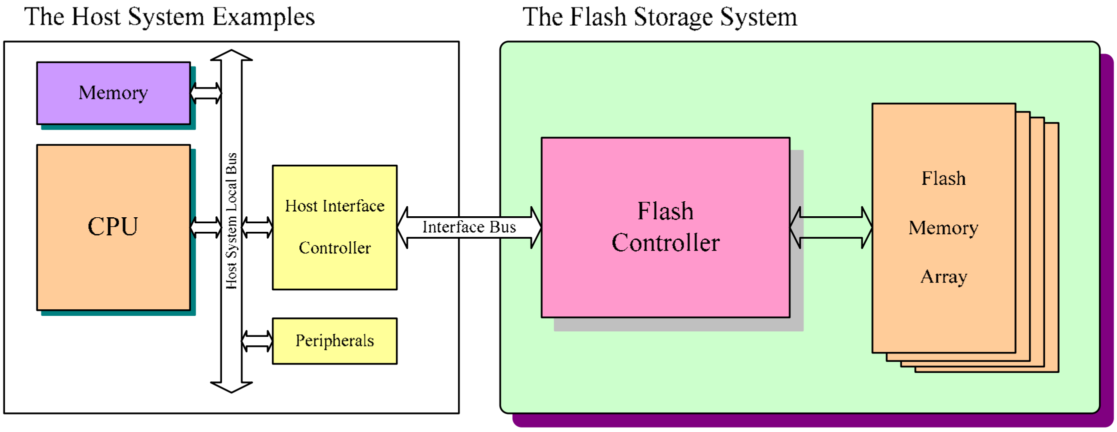


Figure 2-1. The functional block diagram of a flash storage system.

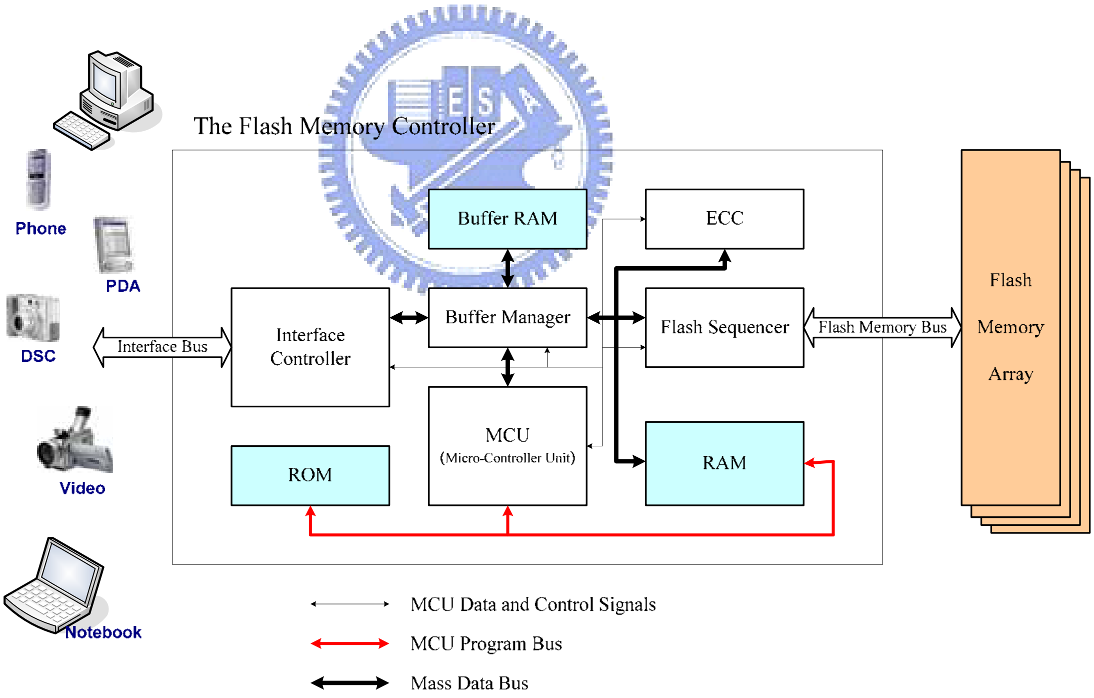


Figure 2-2. The functional block diagram of a typical flash memory controller.

2.2 Host Interfaces

To transfer the data between the CPU and the mass storage device, a host interface for mass data transportation is defined. In general, there are several kinds of host interface defined based on the target applications and the technology level. In Table 2-I, we have listed the major features of several host interface standards. The most popular interface for mass storage device, AT Attachment (ATA) specification [13], is specifically defined for PC mass storage devices, such as Hard Disk Drives, Optical Disk Drives, Solid-State Drives, etc. Serial ATA interface is the advanced high speed serialized ATA interface to support higher data rate up to 3.0 Gbit/s[1] for SATA II standard [14] [15]. The SCSI interface is another popular storage interface for more applications not only specific for PC [16]. The high speed PCI and PCI Express interface are defined for PC-based local bus application internally [17] [18]. Some internally embedded storage devices are using the PCI interface bus, such as RAID (Redundant Array of Inexpensive Disk Drives) devices [35]. The IEEE 1394 interface is a standard interface developed and targeted for multimedia application [19]. The peer-to-peer connectivity of IEEE 1394 creates an inter-operate ability among many Audio-Video devices. Some storage devices for consumer electronics are using IEEE 1394 interface as well. The USB (Universal Serial Bus) is another very popular I/O interface in PC and notebook computer [20]. The mass storage class of USB defines the mass storage device interface on USB bus. The hot-plug capability and plug-and-play function supporting make some removable storage devices use USB interface. Such as: UFD (**U**SB **F**lash **D**isks), UMD (**U**SB **M**obile Hard **D**isk Drives), etc.

A very important application field for the flash memory storage system is the flash memory cards. The flash memory card has become an inevitable peripheral for many

[1] Gbits/s stands for Giga (10^9) bits per second.

consumer electronic devices, such as: DSC (Digital Still Camera), cellular phones, handsets, MP3 player, DV (**D**igital **V**ideo recorder), etc. On the different application consideration, there are also many kinds of the host interfaces defined by the corresponding associations. The CompactFlash card interface is the first successful standard for small-form factor flash memory cards [21]. The CompactFlash interface is a PCMCIA-ATA interface flash memory card, but it is with a smaller mechanical size. The CompactFlash card interface was successfully adopted by DSC as a storage media to store the pictures. Then, the demand of much smaller devices in consumer electronics requires the size of the flash memory card need be simpler and smaller. The MMC (MultiMediaCard) card interface is developed by MMC Association [22]. The SD (Secure Digital) card defined another flash memory card interface with the security support with content protection [23] [24]. The Memory Stick interface is developed by SONY corporation and applied to the SONY branded devices [29]. Although there are so many different host interfaces for flash memory cards, they are quite similar at the functional and technical point of view. They are all in a kind of the flash memory storage systems with different host interface and mechanical form-factors.

The host interface is a necessary part of the flash memory storage systems. It needs to be defined by the target market and technology consideration. The major factors of the host interface can be divided into 4 categories: the general features, the electrical specification, the interface protocol, and physical specification.

In the general features of Host interface, the target applications, power consumption, hot-plug, plug-and-play are included. The target applications are the most important factor to define where the interface is adopted, and where is the device used. The power consumption defined based on how much power need for the transmitter and receiver. It's depends on how long and how fast of the data transfer is required in the application. The hot-plug supportability can support the storage device with the function of removable. The plug-and-play feature defines the support level in the host operating systems.

In the electrical specification of host interface, the bus bandwidth, interface signaling, the functional block of the hardware interface controller are included. The bus bandwidth of the host interface defines the maximum data transfer between the host and the mass storage device. The general metric for the bandwidth of data transfer rate is MB/sec. The signaling of the interface bus includes the DC and AC specification and signal assignment of the interface signals. In general, there are analog or digital types for interface signaling for data transportation. The functional block description of the interface controller defines how many the functional units and the connectivity and the signal and data flow among the functional units.

In the interface protocol of host interface, the data bus type, the protocol handshaking and operational state machines, and bus topology are included. There are generally two types of data bus of the interface: the serial or parallel bus. The serial interface can save space for the connectors and cables of the interface. While the parallel interface provides the simple logic to increase the data transfer rate in the comparatively lower clock rate. The protocol of the interface defines the data transfer and command handshaking mechanism of the interface. There are Master-Slave or Peer-to-Peer types. In general, the protocol of the interface defines the communication mechanism between the devices in the interface bus. The bus topology defines the interface bus using by the host system and the storage device. There are 4 types of bus topology using now: Master-slave, Point-to-Point, Tree, and Common Bus. The Master-Slave type was developed for the ATA standard. The Point-to-Point can have the independent bus bandwidth; thus, it can maintain a highest bandwidth of the data transfer. The new generation Serial ATA interface adopts the point-to-point connection. The tree-type bus topology is easier to manage at the host point of view, but the bus bandwidth will be shared to all the devices attached on the bus. The USB interface uses the tree-type bus topology. The common bus was developed to have multiple interface type in a bus, and can switch the interface mode automatically [109].

In the physical specification of host interface, the mechanical form-factor and dimensions, operating conditions, and testing criteria are included. The mechanical form-factor and dimensions of the interface cables, connectors, and the storage device are defined. The mechanical form-factors and dimensions show the physical outlines of all the components defined in the interface specification. The operating conditions define the operating environment factors of the storage devices. The testing criteria define how to qualify the storage devices to be compliant with the host interface.

In Table 2-I, the major factors of some typical host interface specifications are listed. The SATA II interface shows the highest bandwidth at 300 MB/s[2]. The interfaces of flash memory cards are lower speed, smaller size, and support hot-plug for hot insertion and removal. The analog serial signaling connection is for higher speed interface. From the target application point of view, the flash memory storage systems have three major application fields: the Solid-State Drive (SSD) for PC/Notebook, the flash memory cards for consumer electronics (e.g., DSC, Handsets, etc), the embedded flash modules for IA (**I**nformation **A**ppliance or **I**ndustrial **A**pplication) devices (e.g., IPC, POS, Medical Instrument, etc).

[2] MB/s stands for Mega (10^6) Bytes per second.

Table 2-I. The major factors of some typical host storage interfaces

Interfaces	Applications	Max. BW	Signaling	Bus Topology	Hot-Plug
Parallel ATA	PC Mass Storage	UDMA4: 66MB/s[1] UDMA5: 100MB/s UDMA6: 133MB/s	Digital Parallel	Master-slave	No: fixed Yes: removable
Serial ATA	PC Mass Storage	SATA I: 150MB/s SATA II: 300MB/s	Analog Serial	Point-to-Point	Yes
SCSI	Storage Devices	Ultra2: 40MB/s Ultra2W[4]: 80MB/s Ultra3: 160MB/s	Digital Parallel	Common Bus	No
Serial Attached SCSI	Storage Devices	300 MB/s	Analog Serial	Point-to-Point	No
PCI	Embedded	266 MB/s	Digital Parallel	Common Bus	No
PCI Express	Embedded, external I/O	2.5 Gbps[2] per lane	Analog, multi-lane	Point-to-Point	No: embedded Yes: removable
IEEE 1394	Multimedia AV Devices	a: 400Mbps[3] b: 800Mbps	Analog Serial	Point-to-Point	Yes
USB Mass Storage	Mobile Storage	1.1: 12Mbps 2.0: 480Mbps	Analog Serial	Root-and-Tree	Yes
CompactFlash Card	Flash Cards	CF 3.0: 66 CF 4.0: 133	Digital Parallel	Point-to-Point	Yes
MMC Card	Flash Cards	3.3: 20Mbps 4.0: 52MB/s	Digital Parallel	Common Bus	Yes
SD Card	Flash Cards	1.01: 12.5MB/s 1.1: 25MB/s	Digital Parallel	Point-to-Point	Yes
Memory Stick	Flash Cards	1.4: 2.5MB/s PRO: 20MB/s HG: 60MB/s	Digital Parallel	Point-to-Point	Yes

Note: 1. MB/s: Mega Bytes per second.
2. Gbps: Giga bits per second, same as Gb/s .
3. Mbps: Mega bits per second, same as Mb/s.
4. Ultra2W: SCSI Ultra 2 Wide.

2.3 NAND Flash Memory Brief

NAND flash memory is a kind of non-volatile solid-state memory composed by a NAND structural memory cell array. The page-wise[3] accessing for data reading and programming makes the NAND flash memory is only suitable for mass data storage, so-called the “Data Flash”. Data flash is different to the “Code Flash” in the application point of view; while the Code Flash memory is used for the micro-code accessing for a computer system, which the accessing for data reading is byte-wise or word-wise [1]. NAND flash architecture was first proposed by Toshiba in 1987 [48]. The cell structure of NAND flash let it to be the most highly integrated cell array architecture and to achieve the lowest cost per MB in the semiconductor mass storage field.

In this section, we describe the major characteristics, operations, and applications of NAND flash memory briefly. We describe the basics of NAND flash memory in Section 2.3.1; the deficiency of NAND flash memory is discussed in Section 2.3.2.

2.3.1 Basics of NAND Flash Memory

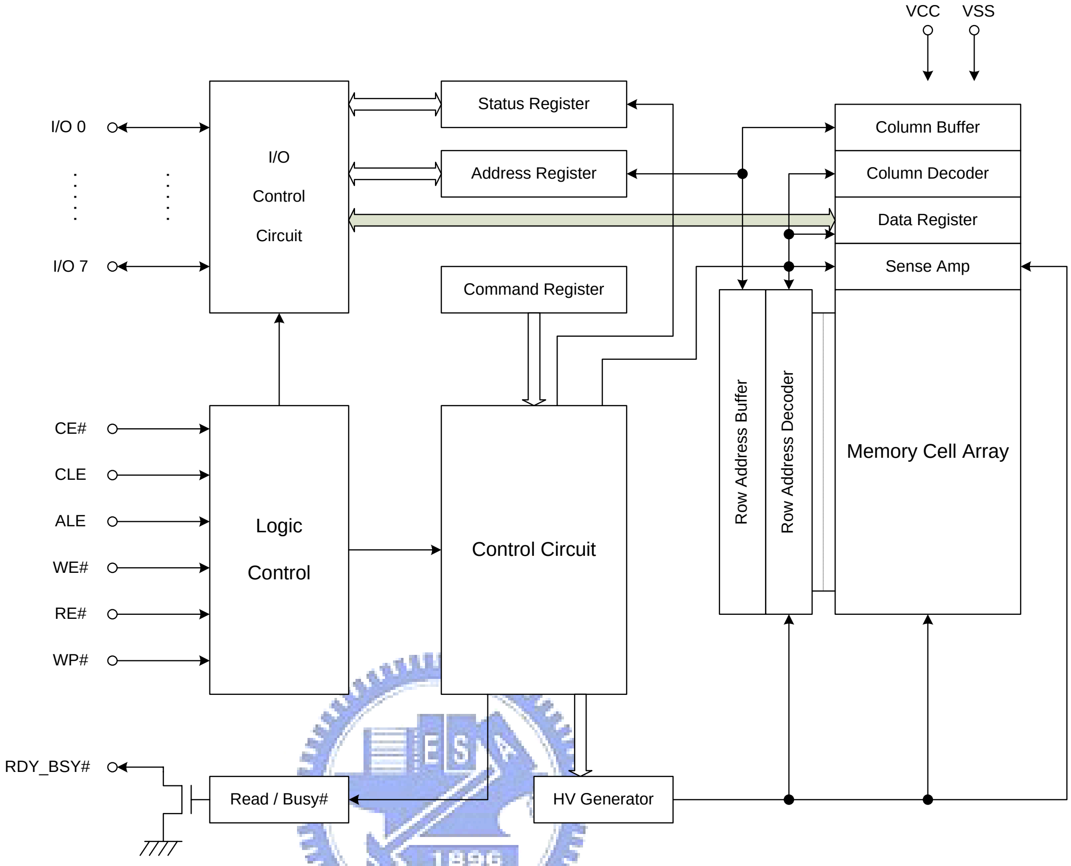
A typical architecture of the NAND flash memory devices is shown as in Figure 2-3. The NAND flash memory is consists of the memory cell array and some of the control logic circuits. The memory cell is addressed by column address as well as row address. The column address pointers address the byte position in a page; The row

[3] Some typical page sizes for page-wise accessing of NAND flash memory are: 512+16 bytes, 2048+64 bytes, and 4096 + 218 bytes.

address pointers address the page position in the memory cell array. Furthermore, several pages form a block, which is the basic erase unit of NAND flash memory. For example, a 32Gb NAND flash memory device can be formed by:

$$\begin{aligned}
&\text{1 Page} && = (4\text{K} + 218)\ \text{Bytes} \\
&\text{1 Block} && = 128\ \text{Pages} = 128*(4\text{K}+218)\ \text{Bytes} \\
& && = (512\text{K} + 27.2\text{K})\ \text{Bytes} \\
&\text{1 Device} && = 8192\ \text{Blocks} = 34{,}508.8\ \text{Mbits} \\
& && = 4313.6\ \text{MBytes}
\end{aligned} \tag{2-1}$$

The operation of the NAND flash memory is controlled by issuing the address and command from the I/O pads. There are 3 kinds of I/O phase in the operation: command phase, address input phase, and data phase. The operational commands are sent during the command phase by enabling the ***CLE*** (Command Latch Enable) signal; the memory address is sent during the address input phase by enabling the ***ALE*** (Address Latch Enable) signal. The data to be written or read is transferred during the data phase by disabling both the ***CLE*** and ***ALE*** signals. Typical operational modes of the NAND flash memory are shown as in Table 2-II. The data reading, programming, and block erasing are the 3 major operation modes in the NAND flash memory accessing. In the data reading, there are command/address input phase, NAND flash memory busy to ready time for reading the data from the cell array to the data register, and the sequential data output. In the data programming, there are command/address input phase, sequential data input, and NAND flash memory busy to ready time for page program. In the block erasing, there are command/address input phase, and NAND flash memory busy to ready time for block erase. The basic operations are shown as in Figure 2-4.



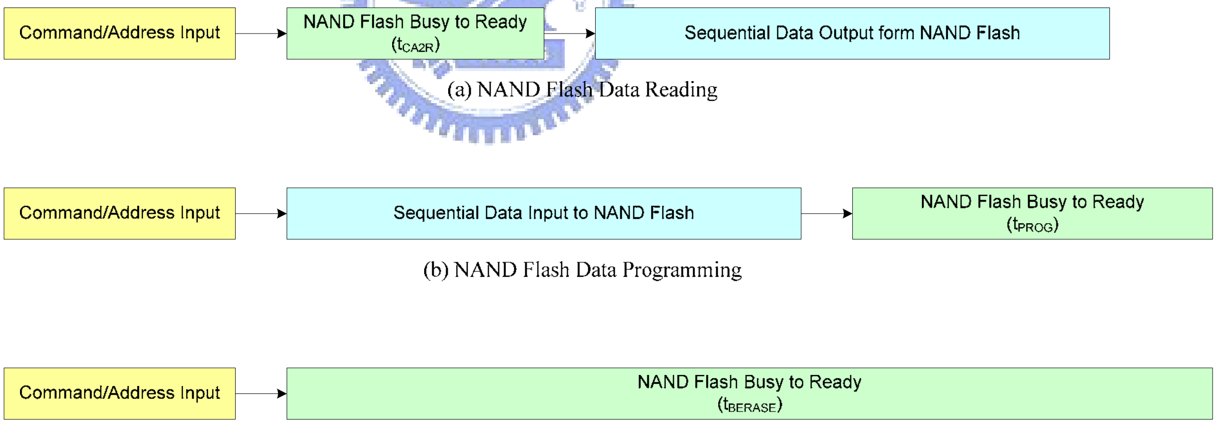
Source: Toshiba NAND Flash memory datasheet [53].

Figure 2-3. The typical functional block diagram of the NAND flash memory.

Table 2-II. The operation modes of NAND flash memory

Operation Mode	CLE	ALE	CE#	WE#	RE#	WP#
Command Input	H	L	L	↑	H	H or L
Data Input	L	L	L	↑	H	H
Address Input	L	H	L	↑	H	H or L
Sequential Data Output	L	L	L	H	↓	H or L
During Program (Busy)	H or L	H or L	H or L	H or L	H or L	H
During Erase (Busy)	H or L	H or L	H or L	H or L	H or L	H
During Read (Busy)	H or L	H or L	H	H or L	H or L	H or L
	H or L	H or L	L	H	H	H or L
Program, Erase Inhibit	H or L	H or L	H or L	H or L	H or L	L
Standby	X	X	H	X	X	H or L

Source: Toshiba NAND Flash memory datasheet [53].



(a) NAND Flash Data Reading

(b) NAND Flash Data Programming

(c) NAND Flash Block Erasing

Figure 2-4. Basic operation sequence of NAND flash memory.

The major AC timing factors that affect the performance of NAND flash significantly are described as below:

- The NAND flash busy to ready time in data reading, data programming, and block erasing is significant. A typical value of t_{CA2R} (busy time of cell array to register as flash data reading) of 2 bits per cell MLC is 50us; a typical value of t_{PROG} (busy time of flash page program) of 2 bits per cell MLC is **800us**; a typical value of t_{BERASE} (busy time of flash block erase) of 2 bits per cell MLC is **2ms**.

- The sequential data output time is equal to the read enable strobe cycle time, t_{RC}, times the total cycles. E.g., the 4K + 218 Bytes per page will need 4314 (=4096+218) cycles. A typical value of t_{RC} is **30ns**, So, the sequential data output time is 4314*30ns = **129.42us**.

- The sequential data input time is similar with the sequential data output time. It is equal to the write enable strobe cycle time, t_{WC}, times the total cycles. E.g., the 4K + 218 bytes per page will need 4314 cycles. A typical value of t_{WC} is **50ns**, so the sequential data output time is 4314*50ns = **215.7us**.

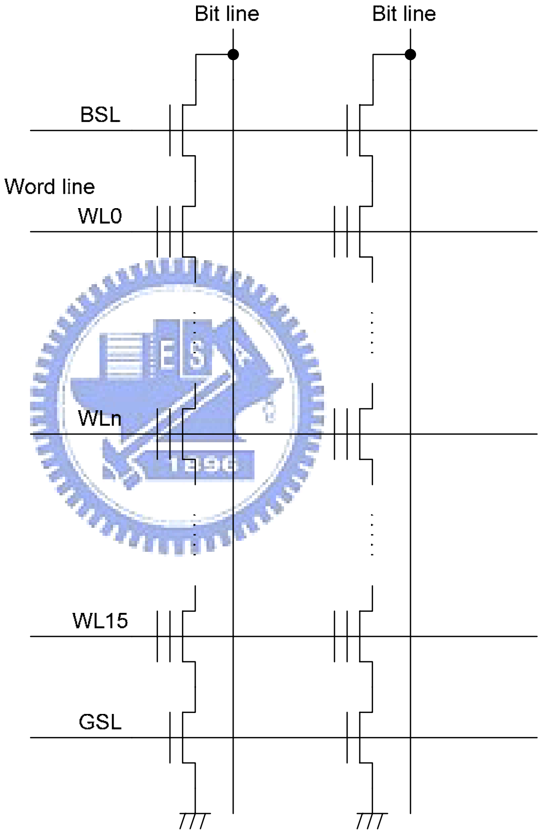
- The command/address input phase is in a few pulses of WE# signal rising. Thus, in the performance point of view, it is relative very small, and can be regarded as overheads to be negligible in brief performance estimation.

2.3.2 Deficiencies in NAND Flash Memory

The structure of NAND flash memory can reduce the total cell array area by word-line pitch scaling [1]. The basic structure of NAND architecture is shown as in

Figure 2-5. Since the parallel architecture and page-wise data accessing, the NAND flash memory was designed and targeting the application for mass data storage device. Because of the application for the mass data storage device, the NAND flash memory allows a certain percentage of defected blocks to increase the production yield and lower the cost. The initial defective blocks can be marked as “Bad” during the production testing stage. The existence of the “Bad” blocks does not affect the “Good” blocks because each block in the memory device is quite independent and individually isolated from the bit lines by block select transistors. Such initial permanent failures can be management by mapping table to replace the defective blocks by some reserved blocks, or lower the formatted capacity of the storage device. Moreover, the usage of the memory cell will cause the cycling of the programmed state (‘0’) and erased state (‘1’). Unlike the magnetic storage media, the NAND flash memory cell has wear-out problems. The endurance issue of the NAND flash memory will cause the block failure during use.

The wear-out problem of NAND flash memory will cause permanent failures; we called it as hard errors. On the other hand, there are some of soft errors caused by operating disturbances or over-programming. The soft errors are defined as they are recoverable by an appropriate and effective manipulation. The **over-programming** phenomenon is caused by the fault-operation during a page program action. It will affect the reading of the other bits in the same word-line. The over-programming issue can be cleared by a new block erasure operation. The **program disturbance** happens when the bit in the page is un-intentionally programmed from ‘1’ (erased state) to ‘0’ (programmed state). The program disturbance can also be cleared by a new block erasure operation. In addition, the program disturbance can be avoided or reduced by programming the pages of the block in sequential order or by stopping using the partial page program. The **read disturbance** happens like program disturbance. It affects the charges stored in the floating gate during the read page process. Move the block data from one block to a new one will clear the read disturbance. The soft errors can be detected by a read operation with the ECC (Error-Correction Code) check. The wear-out and disturbance issues are even more severe in the advanced shrunk technology and MLC flash memory. We summary the deficiency of NAND flash memory in Table 2-III.



Source: Cappelletti et al., [1]

Figure 2-5. Basic structure of NAND architecture.

Table 2-III. The deficiencies in NAND flash memory

Deficiency	Type	Effect
Initial defective blocks	Hard error	Program / Erase Failure.
Defective blocks during use	Hard error	Program / Erase Failure.
Over-programming	Soft error	Bit errors happen during read.
Program disturbance	Soft error	Bit errors happen during read.
Read disturbance	Soft error	Bit errors happen during read.
Page program, block erase	Problematic	Need erase before program.
Program must be in order	Problematic	Need sequential program in a block.
Partial page program inhibit	Problematic	Need buffer RAM.
Long program time	Weakness	Poor performance.
Long erase time	Weakness	Poor performance.
Limited endurance	Weakness	Permanent failure as Wear-out.

2.4 ECC (Error Correction Code)

The Error Correction Code is used to increase the data integrity by adding some redundant information [2-3]. A FEC (**F**orward **E**rror **C**orrection) mechanism can do the error correction independently at the receiver side. The data stored in the storage media is like the data transferred through a communication channel. The defect and the disturbances in the NAND flash memory operation is analogues to the noisy in a communication channel. In Figure 2-6, there is the NAND flash memory is used to store the data. The data written to the NAND flash memory and read back is similar to the data is sent from a transmitter to a receiver in the communication channel. The source data is the data to be written to the NAND flash memory, and the drain data is the data read back from the NAND flash memory. The ECC encoder generates the ECC parities as the redundancy of the user data which can be decoded by the ECC decoder. The ECC decoder can do the ECC decoding and check if the received is error-free or corrupted by noise. If the corrupted data is received, the ECC decoder will do the error correction by finding the error locations and the corresponding error patterns.

In practical, the ECC circuit was used to lower down the BER (**B**it **E**rror **R**ate) to be the lower bound of acceptability, thus let the whole system data integrity can meet the required specification. For example, the BER should be less than 10^{-15} as a data storage device for computer systems.

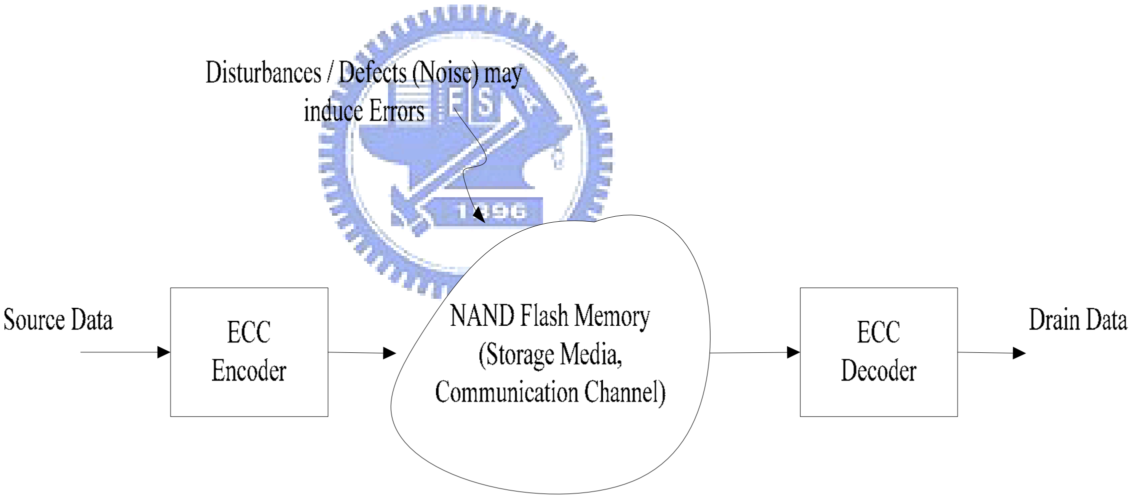

Disturbances / Defects (Noise) may induce Errors
Source Data
ECC Encoder
NAND Flash Memory (Storage Media, Communication Channel)
ECC Decoder
Drain Data

Figure 2-6. Illustration of ECC in the NAND flash memory storage.

There are some famous algebraic codes for doing the error correction in a memory-less communication channel. The RS (**R**eed-**S**olomon) Code is designed for the data error correction of burst errors. RS code is widely adopted in Hard Disk Drive and Optical Disk Drive industry. The BCH codes are used to correct the randomly happened errors in many systems. “BCH” is a name for the 3 historic mathematicians: **B**ose, **C**haudhuri, and **H**ocquengham. The BCH code is very suitable for the error correction for the randomly bit errors in the NAND flash memory, since the noise model of the NAND flash memory is the randomly bit error. In addition, the smaller redundant area requirement for the same EC (Error-Correction Capability) level compare to RS code make BCH is more suitable for the ECC strategy in NAND flash memory storage systems. In Table 2-IV shows the comparison between the BCH code and the RS code.

Table 2-IV. Comparisons between the BCH code and the RS code

Items	**BCH Code**	**RS Code**
Errors Correction	Random Bit Error	Burst Errors
Operation Unit	Binary	Non-binary (Symbol)
Encode Architecture	Long Cascade Shift Registers	Parallel Symbol-base Registers
Decode Architecture	• Syndrome Generators • Syndrome to Error Locator Polynomial • Chien's Search • Error Correction	• Syndrome Generators • Syndrome to Error Locator Polynomial • Chien's Search • Error Patterns Evaluation • Error Correction
Application Examples	Optical Communication, Semiconductor Storage	Optical Disk Drives, Magnetic Disk Drives.

The parity-check matrix is defined for the linear systematic block for t-error correction capability as below [3]:

$$H = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_0^3 & \alpha_1^3 & \cdots & \alpha_{n-1}^3 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{2t-1} & \alpha_1^{2t-1} & \cdots & \alpha_{n-1}^{2t-1} \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2\text{-}2)$$

Where *(α_0, α_1, ..., α_{n-1})* is a list of n distinct non-zero elements of *GF(2^m)*.

From the further induction of parity-check matrix, the generator polynomial of the general *t*-EC (*t*-error correction) BCH ECC code can be expressed as:

$$G(x) = \prod_{i=1}^{t} m_{2i-1}(x) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2\text{-}3)$$

In the equation (2-3), the *t* stands for *t*-error correction, *$m_1(x)$, $m_3(x)$, $m_5(x)$, ..., $m_{2t-1}(x)$* are the minimal polynomials of order *1, 3, 5, ..., 2t-1* of Galois Field *GF(2^m)*.

While the generator polynomial of a general *t*-SC (*t*-symbol Correction) Reed-Solomon ECC code can be expressed as:

$$G(x) = \prod_{j=1}^{2t} (x - \alpha^j) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2\text{-}4)$$

Where α in equation (2-4) is the primitive element of *GF(2^m)*.

Both the BCH and Reed-Solomon ECC use the generator polynomial to generate the ECC parity for errors checking and correction redundancy. If the data is error-free, the all zero values of Syndromes will be gotten by the decoding of BCH and Reed-Solomon ECC codes.

The decoding process of BCH ECC is done by doing the polynomial divisions and finding the remainder polynomial as the corresponding syndrome *($S_1(x)$, $S_3(x)$, ..., $S_{2t-1}(x)$)* by the following minimal polynomials *$m_1(x)$, $m_3(x)$, $m_5(x)$, ..., $m_{2t-1}(x)$*.

$$\begin{aligned} S_1(x) &= \mathrm{mod}(c(x), m_1(x)); \\ S_3(x) &= \mathrm{mod}(c(x), m_3(x)); \\ &\vdots \\ S_{2t-1}(x) &= \mathrm{mod}(c(x), m_{2t-1}(x)); \end{aligned} \qquad \ldots\ldots\ldots\ldots \qquad (2\text{-}5)$$

Where, the c(x) in equation (2-5) is the polynomial of the received code word.

The decoding process of Reed-Solomon ECC is done by doing the polynomial divisions and finding the remainder polynomial as the corresponding syndrome *($S_1(x)$, $S_2(x)$, ..., $S_{2t}(x)$)* by the following polynomials: *(x-α^1), (x-α^2), ..., (x-α^{2t})*.

$$\begin{aligned} S_1(x) &= \mathrm{mod}(c(x), (x-\alpha^1)); \\ S_2(x) &= \mathrm{mod}(c(x), (x-\alpha^2)); \\ &\vdots \\ S_{2t}(x) &= \mathrm{mod}(c(x), (x-\alpha^t)); \end{aligned} \qquad \ldots\ldots\ldots\ldots \qquad (2\text{-}6)$$

Where, the c(x) in equation (2-6) is the polynomial of the received code word.

If the received code word is corrupted by noise, the non-zero of syndromes will be decoded. The next step of both BCH and Reed-Solomon ECC is to find the error locator polynomial by solving the key equation. The key equation can be solved by the Berlekamp-Massey Algorithm [2]. The general format of an error locator polynomial for finding u errors is shown in equation (2-7).

$$\sigma(x) = \sum_{i=0}^{u} \sigma_i x^i \qquad \ldots\ldots\ldots\ldots \qquad (2\text{-}7)$$

Where, the σ_i in the equation (2-7) is called the coefficients of the error location polynomial.

To find the error locations, the iteration for $x = \alpha^i$ evaluating is processed. This process is called the Chien's Search. After finding the all of the error locations of the corrupted received code word, the bit error correction process of the BCH ECC code can be finished by doing the bit-inversion operation at the found error locations. While the Reed-Solomon ECC due to its symbol-base ECC code, it needs to find the error patterns at the corresponding error locations. The process is called as the error pattern evaluation.

There are 3 possible results after the decoding of the received data code word. Case (i), if all of the syndromes are all zero, the error-free of the received data code word result is judged. Case (ii), if the error locations and error patterns can be found and consistent with the mathematical formula, the correctable errors of the received data code word result is judged. The third condition is happened while none of the case (i) or case (ii). It is called as the un-correctable data error.

The ECC circuit architecture for BCH and RS code is quite similar. The RS code can be regarded as an extension of BCH code to a symbol base error correction code. Both of the BCH and RS code are in the class of cyclic linear block code. Figure 2-7 shows the general architecture for the BCH and RS ECC circuit. The ECC Registers are used to store the temporary data during the ECC encoding, ECC decoding, and finding the error locations and calculating error patterns. The key equation solver is designed to do the syndrome to error locator polynomial; the Berlekamp-Massey algorithm was constructed to do the operation. The error searcher is designed for finding the error locations by evaluating the error locator polynomial. Such evaluation of error locator polynomial is also called the Chien’s search algorithm. For the BCH ECC case, the corrector FSM (Finite State Machine) just doing the bit-inversion of the found error locations. For the RS ECC, there needs an operation to find the error pattern of each found error location, because RS is the symbol-base ECC code.

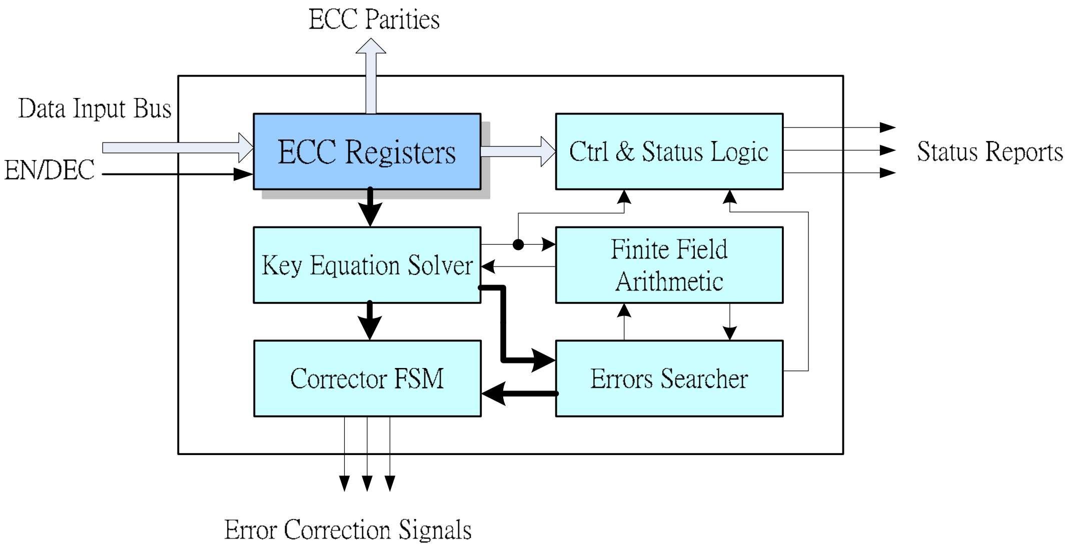


Figure 2-7. General architecture of ECC circuit.

2.5 Flash Memory Management

In this section, we discussed the flash memory management in the NAND flash memory controller. The flash memory management is related to algorithms on managing the logical block address to physical block address mapping, defective blocks management, wear-leveling on block erase, and the refreshing or scrubbing process of the NAND flash memory.

2.5.1 L2P (Logical to Physical) Block Mapping

The whole data transfer flow from the host computer to the storage devices can be divided 3 layers: The host file system level, the interface logical block address level, and the device physical block level, as shown in Figure 2-8 and Figure 2-9. In the flash data accessing, the L2P (Logical to Physical) block mapping and translation is necessary because of the different data accessing behavior in the logical side and the physical side [48-49]. In Figure 2-10, a typical L2P mapping was proposed by SSFDC (**S**olid-**S**tate **F**loppy **D**isk **C**onsortium) [33].

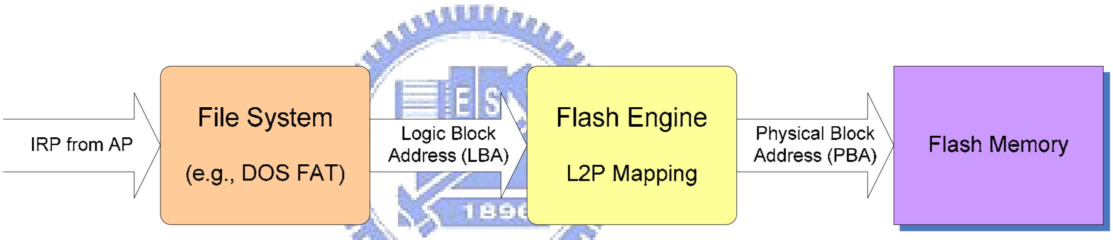


Figure 2-8. The data flow of host system to storage device.

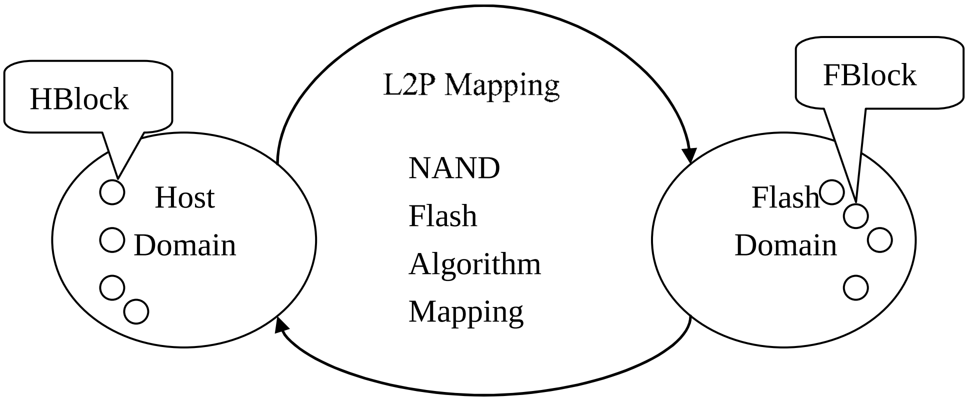
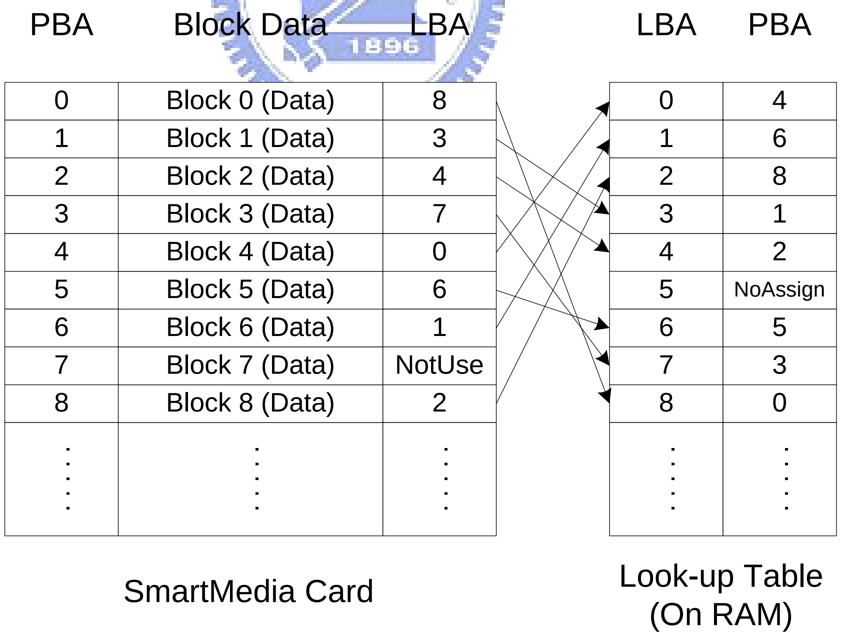


Figure 2-9. Illustration of L2P block mapping mechanism.



PBA	Block Data	LBA
0	Block 0 (Data)	8
1	Block 1 (Data)	3
2	Block 2 (Data)	4
3	Block 3 (Data)	7
4	Block 4 (Data)	0
5	Block 5 (Data)	6
6	Block 6 (Data)	1
7	Block 7 (Data)	NotUse
8	Block 8 (Data)	2
⋮	⋮	⋮

SmartMedia Card

LBA	PBA
0	4
1	6
2	8
3	1
4	2
5	NoAssign
6	5
7	3
8	0
⋮	⋮

Look-up Table
(On RAM)

Source: SmartMedia Interface Library, Software Edition [33].

Figure 2-10. Typical L2P block mapping algorithm of SmartMedia.

2.5.2 Defect Block Management

The defect block management is used to replace the defective blocks of the NAND flash memory with reserved spare blocks while the flash blocks were marked “bad” in the production test or become failure during in use. Doing the defective blocks scan and replacement in the production stage is called as “Statically Defect Management”; doing the failure block replacement during the usage is called as “Dynamically Defect Management” [48-49]. Figure 2-11 shows a typical mechanism for the NAND flash memory defect block Management. The Defect management table was constructed to do the mapping relationship between the original pointed block and the replacement block while the original pointed block is a defected block.

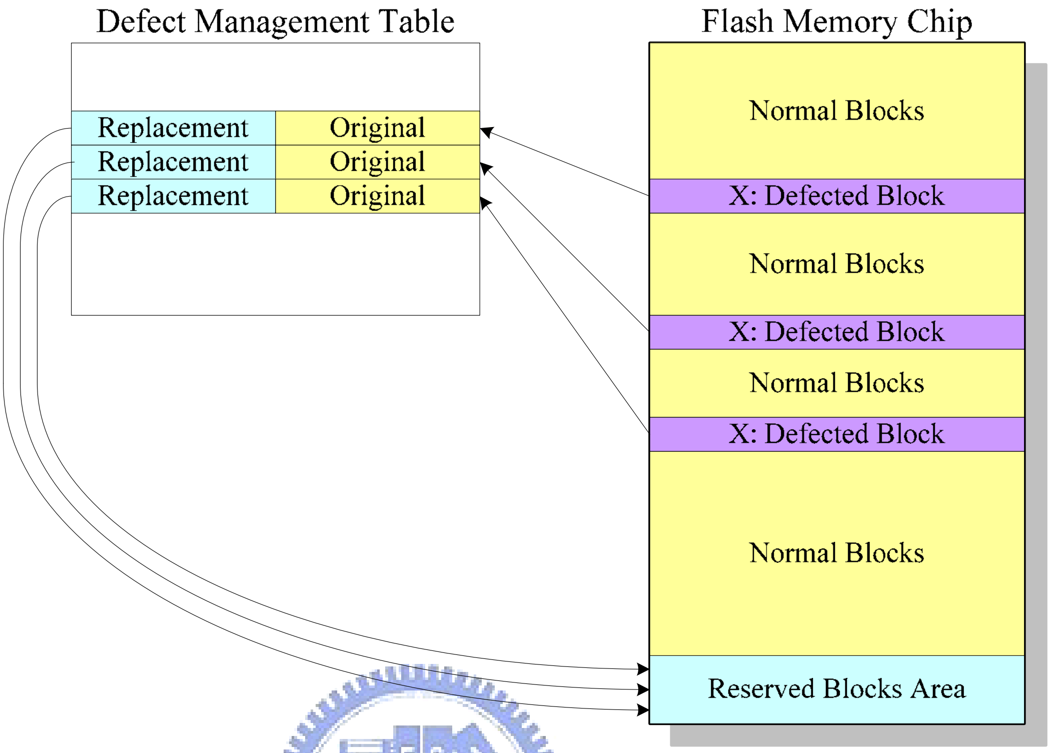


Figure 2-11. Typical defect block management mechanism of NAND flash memory.

2.5.3 Wear-Leveling

The wear-leveling issue is significantly important as the NAND flash memory is used for the solid-state drive. Since the original DOS (**D**isk **O**perating **S**ystems) file system was created for magnetic type storage device, the wear-out problem was not been seriously considered. The un-average usage work load of the each NAND flash memory block will cause the more frequently used area become wear-out. For example, the FAT (File Allocation Table) and directory of the DOS file system was updated very often as the files was in copying and deleting. Moreover, the NAND flash blocks storing the static data (write seldom but read often) are used in dramatically low frequency, and the reserved blank data area for the file system will not be used in operation in practical DOS file system operation.

To increase the product life time, a wear-leveling algorithm was implemented by the subroutines in the system firmware. As shown in Figure 2-12, the wear-leveling algorithm was created by the turn-around of all the flash blocks. The erase cycles of each block in the flash memory device, therefore, could be averaged through the wear-leveling algorithm [73].

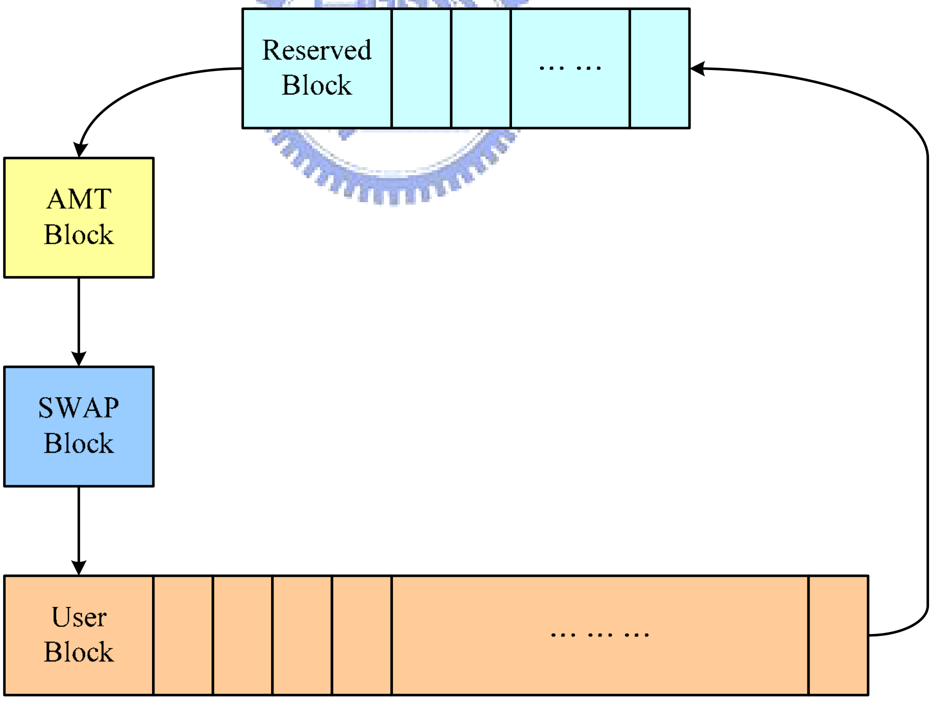
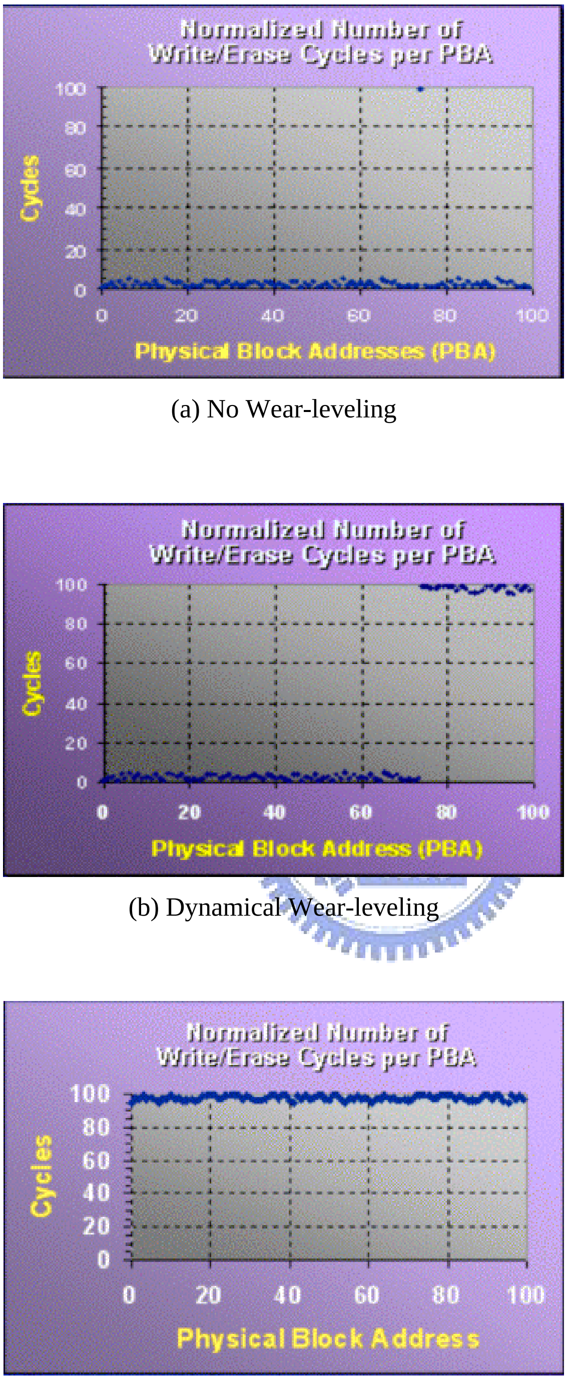


Figure 2-12. Typical wear-leveling mechanism of NAND flash memory.

Based on the wear-leveling on the dynamical data (usually to be read and updated files) or the statically data (write seldom but read often, or even write once read many), there are 2 levels of the wear-leveling defined: the dynamical wear-leveling and statically wear-leveling. The dynamical wear-leveling is only doing the wear-leveling on the dynamical data. It is simpler for implementation and affects the performance little, but it will have a result as 2 groups of wearing level. The usage level of static data is dramatically low as comparing with the dynamical data, even though the wearing level of the dynamical data is averaged. The static wear-leveling is doing the wear-leveling on the whole good flash memory blocks, thus, consequently it can make the wearing level for each NAND flash memory block in a small amount difference. Figure 2-13 shows the comparison of no wear-leveling, dynamical wear-leveling and static wear-leveling.

Instead of using the magnetic storage media oriented file system, such as DOS FAT, there are software drivers which were developed for NAND flash file system, like the Journaling type file system. These types of file systems use the NAND flash memory by solely in sequential order, thus, the work load for each block is evenly inherently. For example, the open source of JFFS2 (http://sources.redhat.com/jffs2/) and YAFFS (http://www.aleph1.co.uk/armlinux/projects/yaffs/) are available on the web-site.



(a) No Wear-leveling

(b) Dynamical Wear-leveling

(c) Static Wear-leveling

Note:
75% Static Data Simulated.

Source: Silicon Systems, http://www.storagesearch.com/siliconsys-art1.html

Figure 2-13. Comparison of dynamical wear-leveling and static wear-leveling.

2.5.4 Refreshing or Scrubbing

The charge-keeping quality in the floating gate structure of NAND flash memory is related to the disturbances and endurances. As the erase cycles increased of a NAND flash memory block, the charge-keeping quality decreased. The charge-keeping quality decrease will cause the bit sensing mechanism become difficult, thus, the bit error may happen. The charge-keeping problem caused by disturbances can be cleared by a new block erasure operation. A new block erasure operation can be happened as moving the data to a new block and erase it as blank block, this action is called the refreshing or scrubbing. The timing for doing the refreshing can be determined by counting the usage frequency or by detecting the error bits in reading [79].

Figure 2-14 shows a typical refreshing algorithm for NAND flash memory. The refreshing of the flash memory block is check during the read process. If the correctable bit errors happened, and the bit error count in a flash page is over a threshold, a refreshing operation will be executed. The flash block refreshing is done by the following procedure:

(1) Copy the block data to a cleaned spare block,

(2) Update the L2P table,

(3) Do the block erasure on the block and set it as a spare block.

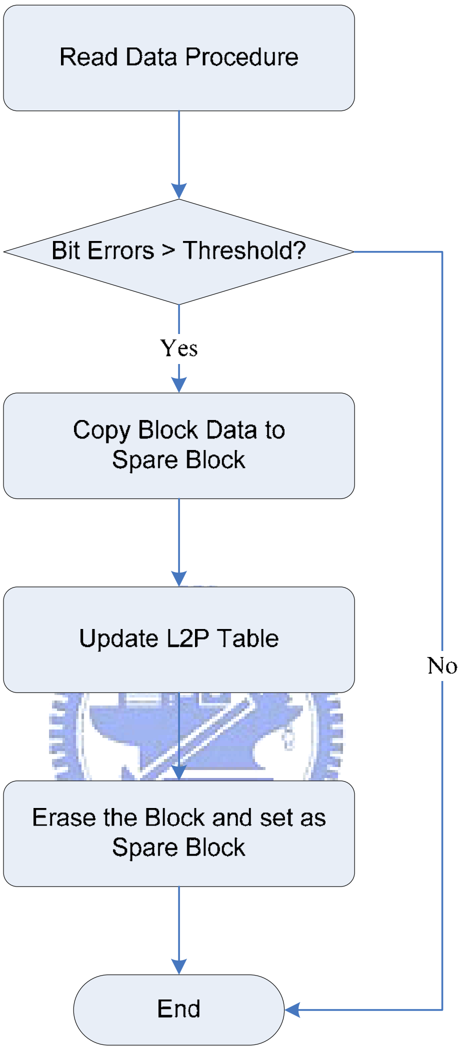

Read Data Procedure
Bit Errors > Threshold?
Yes
Copy Block Data to Spare Block
Update L2P Table
Erase the Block and set as Spare Block
No
End

Figure 2-14. Typical refreshing or scrubbing of NAND flash memory.

2.6 Flash Memory Configuration

The flash memory configuration is discussed in this section. The flash memory configuration was discussed about the organization structure as the multiple flash memory devices are used in a flash memory storage system. There are parallelisms of the NAND flash memory as well as the redundancies of the NAND flash memory. The parallelism in a flash storage system is used to improve the performance of data transfer rate. There are 2 types of parallelism of the NAND flash memory using in configuration: the multiple channels and interleave. The redundancy of the NAND Flash memory can further guarantee the data integrity besides the ECC, and the flash memory management. The RAID level-1 and level-5 were introduced to do the NAND flash memory redundancy.

2.6.1 Flash Memory Parallelism

Parallelism of the NAND flash memory in a flash storage system is used to improve the performance of data transfer rate. There are 2 types of parallelism of the NAND flash memory using in configuration: the multiple channels in vertical, and the interleave in horizontal [75]. Figure 2-15 shows a typical architecture of NAND flash memory parallelism by multi-channel. Figure 2-16 shows a typical architecture of NAND flash memory parallelism by interleave. The detailed operation and performance analysis and calculation were discussed in chapter 5.

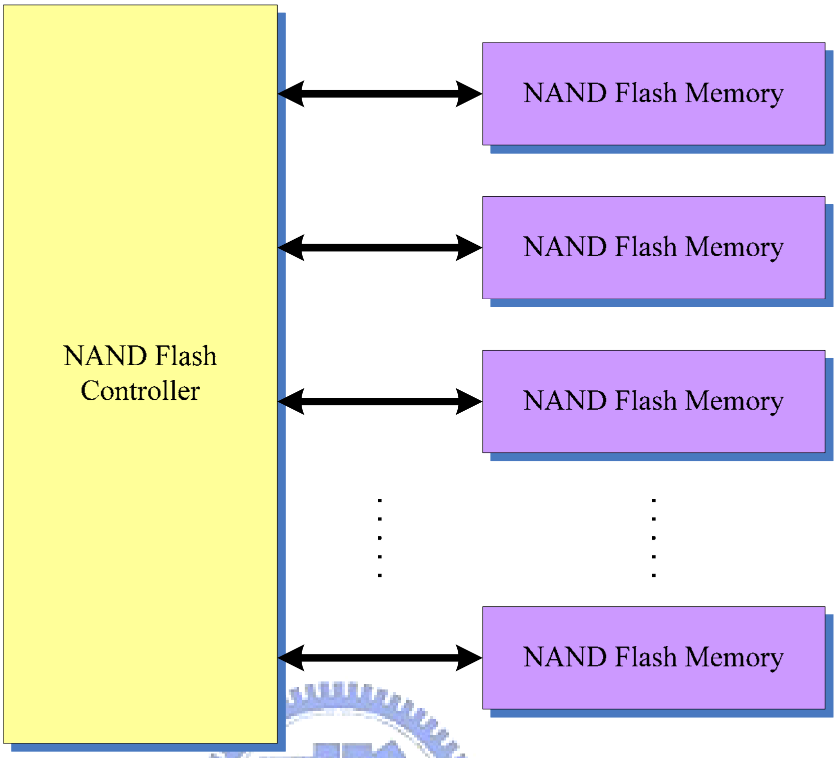
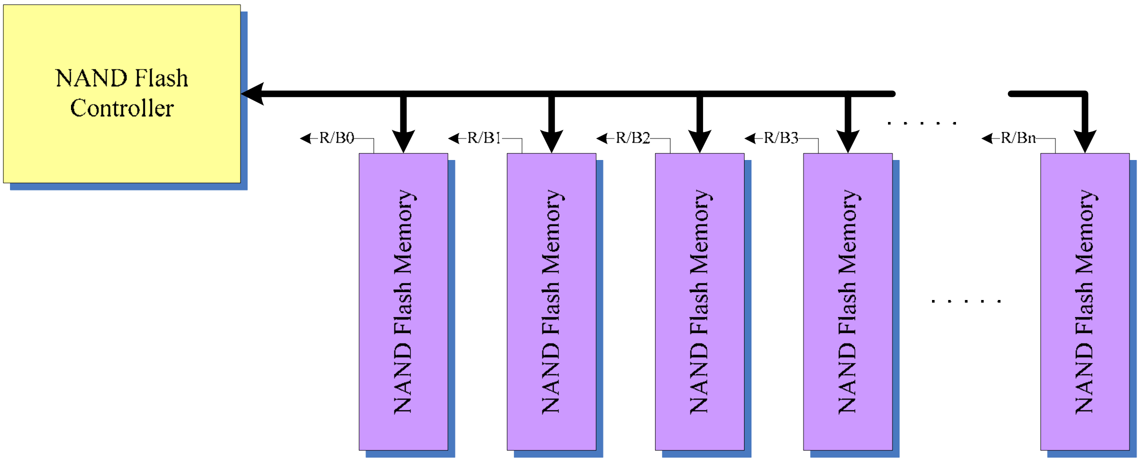


Figure 2-15. Typical architecture of NAND flash memory parallelism by multi-channel.



Note: R/B is the ready/busy indication signal of flash memory device.

Figure 2-16. Typical architecture of NAND flash memory parallelism by interleave.

2.6.2 Flash Memory Redundancy

Redundancy of the flash memory can enhance the data integrity, especially for the data retention problem as the endurance cycles of the NAND flash memory block is going to be expired. The redundancy strategy used in the NAND flash memory is similar to the RAID level 1, 5 of RAID technology developed [35]. Figure 2-17 shows a typical architecture of NAND flash memory redundancy.

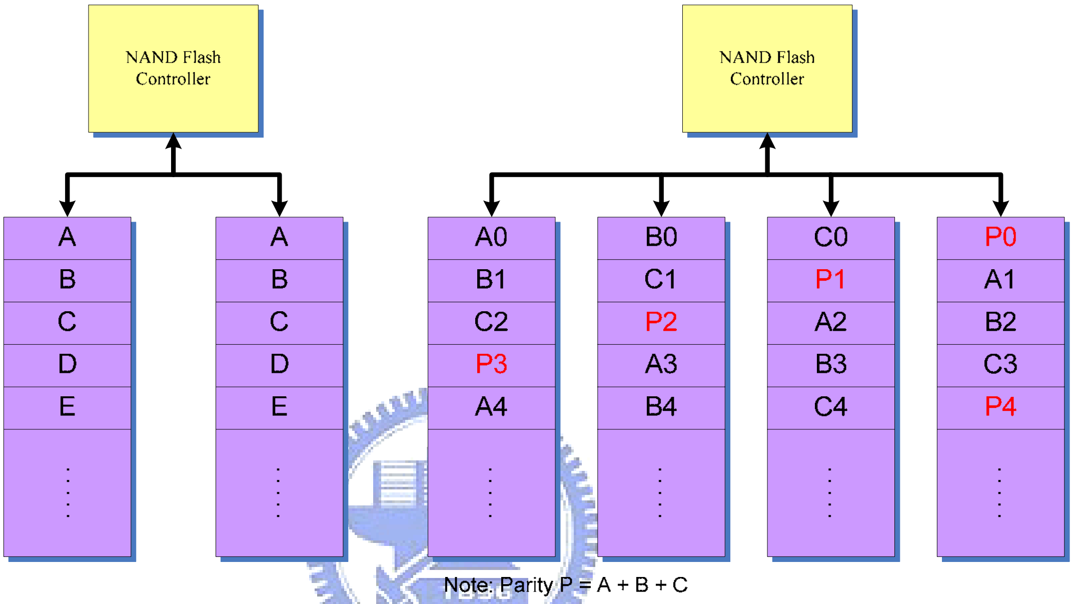
The RAID level-1 is doing the data redundancy by the mirror of the block data. The flash memory device #0 and device #1 will do the write data simultaneously (in parallel) to keep the data both in the flash memory device #0 and device #1. While as during the data reading, the read data can be taken either from device #0 or device #1. Once an un-correctable data error happened, another mirrored data can be attained from the other flash memory device.

The RAID level-5 constructed with 4 flash memory devices was shown in Figure 2-17, and the RAID level-5 is doing the data redundancy by the exclusive-OR formula as shown in equation (2-8).

$$P = A \oplus B \oplus C \qquad \text{. . . . .} \qquad \text{(2-8)}$$

Once one of the data block *A*, *B*, or *C* contains the un-correctable data error, the corrected data can be attained by solving the exclusive-OR formula by re-arranging the term of *A*, *B*, *C* and *P*. For example if block data *A* contains un-correctable data error, the corrected block data A^* can be attained by equation (2-9).

$$A^* = P \oplus B \oplus C \qquad \text{. . . . .} \qquad \text{(2-9)}$$



Note: Parity P = A + B + C

(a) Redundancy with Mirror RAID 1

(b) Redundancy with Mirror RAID 5

Figure 2-17. Typical architecture of NAND flash memory redundancy.

2.7 Cryptography and Security

The data security is important for copyright protection and privacy. The data security in the flash memory storage device can be implemented by the cryptographer circuit and AKE (**A**uthentication **K**ey **E**xchange) mechanism.

Figure 2-18 shows a typical architecture of NAND flash memory controller with a security controller circuit. The security controller was designed in conjunction with the MCU and Buffer, which can do the data encryption as writing the data to the flash memory array, and do the data decryption as reading the data from the flash memory.

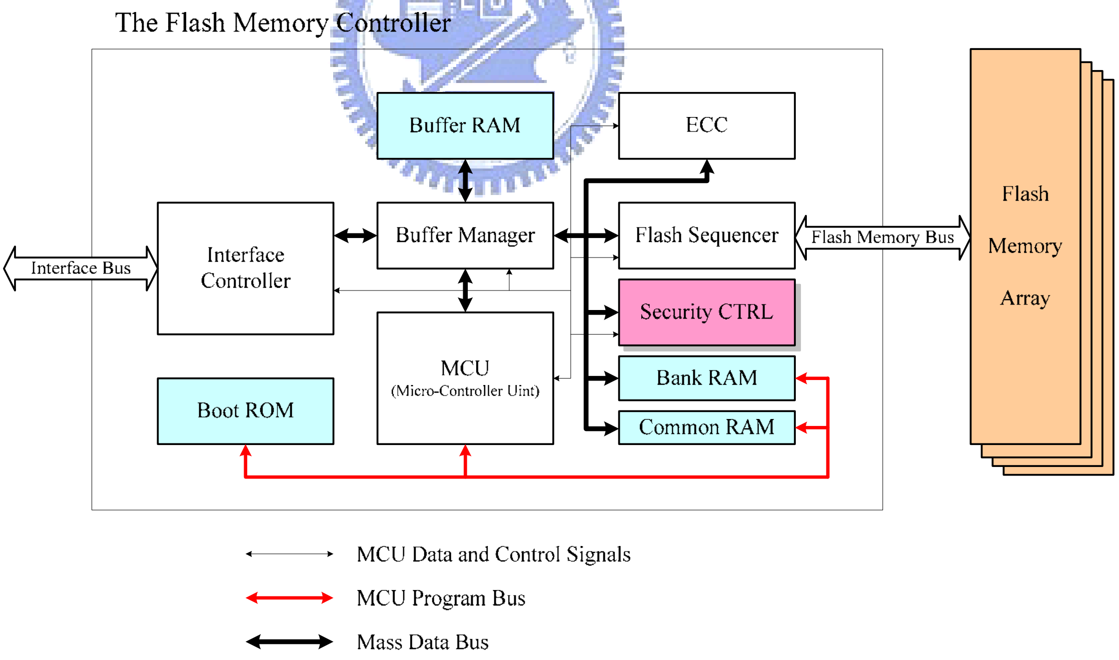


Figure 2-18. Typical architecture of NAND flash controller with security controller.

2.8 System-Level Performance

In this section, we discussed the system-level performance of the NAND flash memory storage systems controlled by the NAND flash memory controller. The data reading and writing, and command cache is used to decrease the latency time as reading and writing the data to the flash memory storage device. Thus, it increases the random block data accessing of the flash memory storage device. The compatibility of the NAND flash memory storage device is very important if it is used as a mobile storage device, such as flash memory cards, or UFD. The various flash memory supporting is meaningful while current flash memory sources are rich and versatile.

2.8.1 Data Read / Write, and Command Cache Operation

To improve the long latency time of Hard Disk Drive spindle speed-up, and disk head positioning, the read data caching and write data caching through a RAM buffer was invented [100]. The read look ahead data caching, and write buffer cache were adopted for the data caching in the accessing to Hard Disk Drive. Moreover, the commands order sent from host might not be suitable and in good efficient order for Hard Disk Drive. Thus, the NCQ (**N**ative **C**ommand **Q**ueuing) was developed to re-order the commands sent from host [103]. These data caching algorithms really improved the data accessing efficiency and save the power consumption of Hard Disk Drive.

As an analogy to the Hard Disk Drive, the data caching algorithm used in the Hard Disk Drive can be extended to the NAND flash memory storage system. The read look-ahead cache was designed by guessing the host reading data is in sequential order.

Thus, the flash controller can read the next subsequent data in advance as finishing the read command [76]. The write cache was designed to have a big enough size of buffer-RAM to accept the writing data from the Host. The data writing process was earlier completed as the data were written into the buffer RAM, instead of writing the data to the flash memory. The write cache saves lots of the latency time of the NAND flash block mapping calculation before start a writing data sequence to the NAND flash memory. In addition, to prevent the potentially accidental power failure during the write data, the non-volatile memory caching for write cache was introduced [105-106].

2.8.2 Compatibility

The compatibility issue raised when the NAND flash memory storage used as a removable storage device (e.g., the flash memory cards and USB flash disk). The incompatibility between the host system and the storage device causes the problem in inter-plug ability, and it makes the inconveniences for transferring information among host systems through the mobile flash memory storage device.

To solve the tedious compatibility validation job, a systematic way to verify the compatibility was developed by a golden host model [71]. The golden host model was established by the studying and experiments on the standard specification, and by doing the test on the major models in the marketing. Figure 2-19 illustrates the model matching methodology for compatibility analysis. The model matching methodology is a hardware test platform developed by a corporation in Taiwan, Prolific Technology, Inc. (http://www.prolific.com.tw). The main concept of the model matching methodology is assumed the greater similarity with the marketing leader brand, the better compatibility of the test sample will be. For example, in the SD card market, Sandisk (http://www.sandisk.com) is the leading brand in the flash card market.

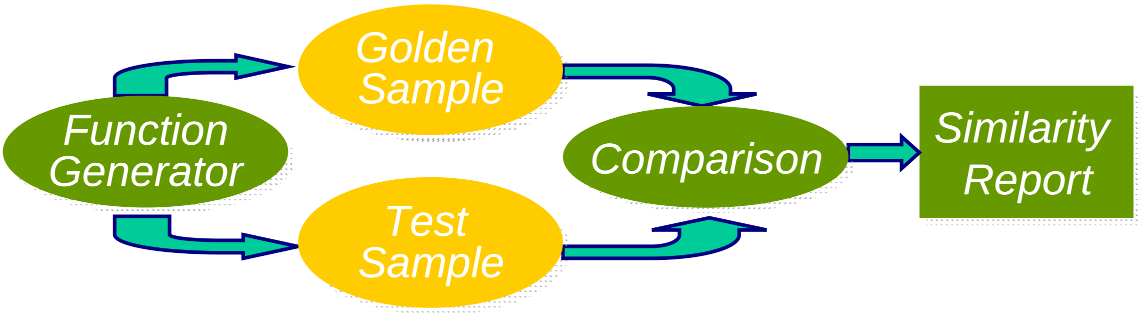


Figure 2-19. Illustration of model matching methodology for compatibility.

2.8.3 Various Flash Memory Supporting

Since the first commercial NAND flash memory device was shipped in 1989 by Toshiba, the NAND flash memory based solid-state mass storage devices was adopted in the market gradually. In 2005, the total capacity of NAND flash memory exceeded DRAM memory. The application of NAND flash memory is becoming very popular now. Nowadays, versatile of the NAND flash memory was provided by the major semiconductor companies in the world. The Intel, Toshiba, Samsung, Hynix, Micron, STMicroelectronic, etc. has the NAND flash memory product line in series. The NAND flash memory types are no longer simple now. Thus, to support the various flash memory types for a flash memory controller is becoming indispensable.

There are some strategies to support the various flash memory devices in the market. To read the device ID of the flash memory prior to access it is the popular method used in the flash memory controller. The drawback of the read flash ID algorithm is it needs to include all the firmware for the supported types of flash memory devices. It may be no problem in the earlier stage while the NAND flash memory types are still little. Another strategy is to have the firmware upgrade capability. Once the NAND flash memory type chosen, the corresponding firmware code was loaded to the flash controller. Thus, complete the one-to-one match of the controller and the NAND flash memory array.

2.9 Typical Flash Memory Controllers

The NAND flash memory is widely used as a storage media in many application fields. Typical flash memory controllers can be divided by 4 categories: flash memory cards, PC removable storage device, embedded flash memory modules, and solid-state drive. The necessary functional units of the NAND flash controller were described in this chapter. While in different kinds of applications, the required specifications of each functional unit may be different. Table 2-V shows the importance of each functional unit in the flash memory controller for typical applications.

Table 2-V. The importance of each functional unit for different applications

Category	Flash Cards	Mobile Storage	Embedded Flash	Solid-State Drive
Host Interface	SD, MMC, MS	USB, IEEE1394, PCIe	PCIe, IDE, SATA	IDE, SATA, PCIe,SCSI, SAS
ECC	* * * *	* * * *	* * * *	* * * * *
L2P efficiency	* * *	* * *	* * * *	* * * * *
Wear-Leveling	* * *	* * * *	* * * *	* * * * *
Refreshing	* * *	* * * *	* * * *	* * * * *
Security	* * * * *	* * *	* *	* *
Parallelism	* * *	* * *	* * * *	* * * * *
Cache	* * *	* * *	* * * *	* * * * *
Redundancy	* * *	* * *	* * * *	* * * * *
Compatibility	* * * * *	* * * * *	* * *	* * *

Note:

*****: Very Important

****: Important

***: Fair

**: Optional

Chapter 3 BCH ECC Circuit Implementation by Systolic Array

We presented a novel *t*-EC *w*-bit parallel data I/O BCH ECC circuit construction procedure for *t*-EC *w*-bit parallel data I/O bus by systolic array in this chapter. The *t*-EC *w*-bit parallel data I/O BCH ECC is necessary for the advanced high-density NAND flash memory chip. In section 3.1, we talk about the introduction and related works on the FEC (Forward Error Correction code). In section 3.2, we describe the construction procedure of this novel *t*-EC *w*-bit parallel data I/O BCH ECC circuit. In section 3.3, we illustrate the design procedure by a typical *4*-EC *16*-bit parallel BCH ECC circuit for 70nm MLC NAND flash memory which requires 4 bits random error correction capability. In section 3.4, we summarize the novel *t*-EC *w*-bit parallel data I/O BCH ECC circuit design by the chip implementation of the controller chip for SD/MMC memory cards via UMC 0.18um process.

3.1 Introduction

In this chapter, we presented a BCH ECC circuit construction procedure for *t*-EC *w*-bit parallel data I/O bus by systolic array. The *w*-bit parallel data I/O is required for the parallel data I/O bus (×8 or ×16) of a NAND flash memory chip.

The FEC (**F**orward **E**rror **C**orrection code) is usually demanded for ensuring reliable data transmission in noisy communication channels, while maintaining a good system performance. There have been already a lot of algebraic ECC codes proposed for FEC purpose, in which the cyclic attribute of the BCH (**B**ose-**C**haudhuri-**H**ocquenghem) code makes an easy adoption for hardware circuit implementation and has become a common choice in most applications [2,3]. Different from the BCH code is suitable for correcting random bit errors, the derivative RS (**R**eed-**S**olomon) code was developed for burst errors correction. Both codes are of similar mathematical fundamentals, while the RS code can be regarded as an extension of the BCH code to be a non-binary, symbol-base operation. Nowadays, many applications adopt BCH or RS code as a standard ECC (**E**rror **C**orrection **C**ode), for example, CD (**C**ompact **D**isc), DVD (**D**igital **V**ersatile **D**isc), HDD (**H**ard **D**isk **D**rive), Ether-Net, wireless communications, etc. The parallel data I/O bus (×8 or ×16) of a NAND flash chip thus requires a parallel BCH ECC codec, applied the parallel CRC (**C**yclic **R**edundancy **C**heck) concept was proposed by Zukowski [62]. A longer encoder polynomial for BCH code is spontaneous to have a stronger bit error correction capability. Some high speed architectures were presented by K. K. Parhi, et al [63-64, 66]. Also, Jun Zhang et al presented an optimized architecture for the long parallel BCH encoder [65]. In addition to the long encoder issue, there were time and power consumption issues solved by Chien's search for error location [66-67]. These prior researches provide a formulated basis for a parallel BCH ECC code and circuit implementation.

In the highly regular and iterative VLSI architecture, the systolic array processors were introduced for the architecture and timing design by projection. The systolic array architecture has been applied to RS encoders and decoders, and showed good performance [68-70]. The systematic design approach of a systolic array processor can make the circuit design easy for implementation and easy for pipelining to fit the system level design specifications. In this chapter, we presented a *t*-EC *w*-bit parallel BCH ECC code with incorporating the systolic array architecture. A typical application example of 4-EC 16-bit parallel BCH ECC circuit was designed for the NAND flash controller. The good performances were shown by the real chip realization for SD/MMC card.

3.2 Construction Procedure

In this section, we presented the construction procedure for the *t*-EC *w*-bit parallel BCH ECC circuit. The *w*-bit parallel data I/O is required for the parallel data I/O bus (×8 or ×16) of a NAND flash memory chip.

3.2.1 The Serial t-EC BCH ECC Code Construction

A general *t*-EC BCH ECC code can be constructed through the procedures shown as in Figure 3-1.

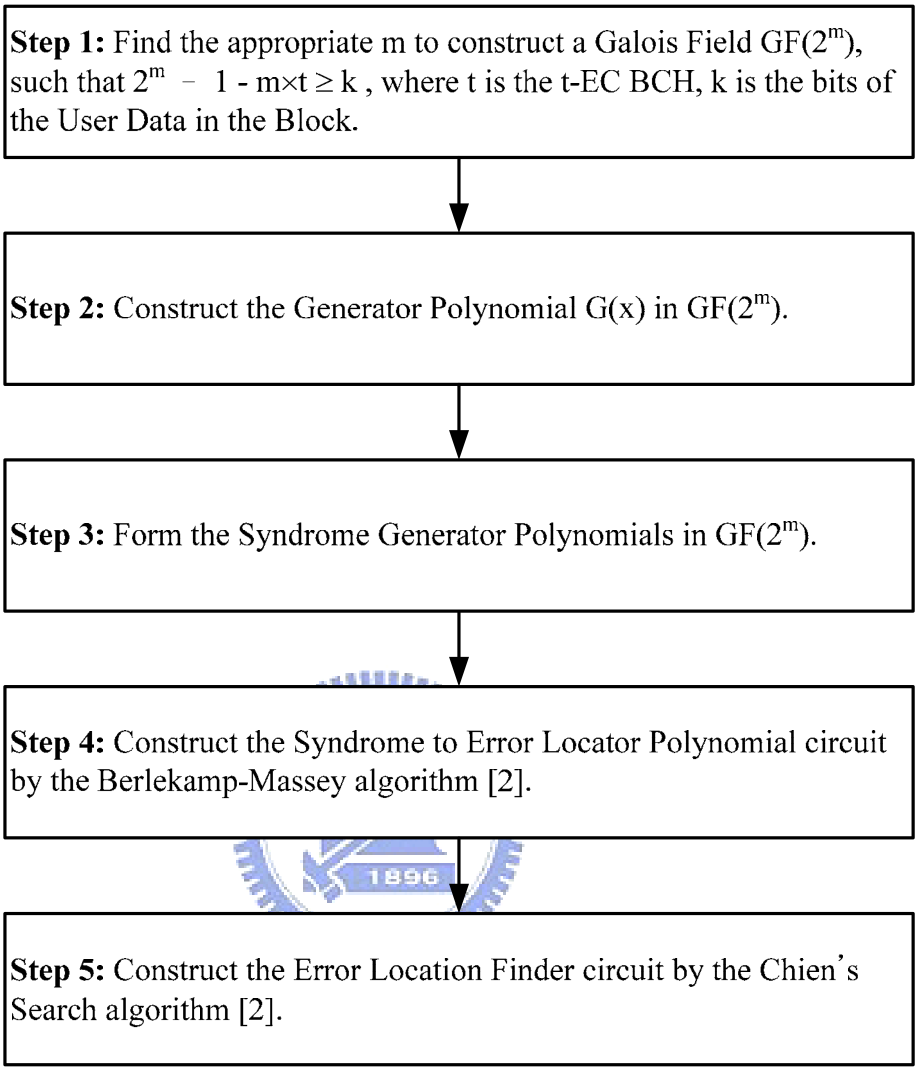

Step 1: Find the appropriate m to construct a Galois Field GF(2^m), such that $2^m - 1 - m \times t \geq k$, where t is the t-EC BCH, k is the bits of the User Data in the Block.
Step 2: Construct the Generator Polynomial G(x) in GF(2^m).
Step 3: Form the Syndrome Generator Polynomials in GF(2^m).
Step 4: Construct the Syndrome to Error Locator Polynomial circuit by the Berlekamp-Massey algorithm [2].
Step 5: Construct the Error Location Finder circuit by the Chien's Search algorithm [2].

Figure 3-1. The t-EC serial BCH code construction procedure

In **Step 1**, an adequate value for m should be selected to construct the $GF(2^m)$ Galois Field via a selected primitive polynomial $p(x)$, where m must satisfy $n = k + m \cdot t$, where t stands for the t-EC error correction capability, k is the bit length of user data, and n is the bit length of the encoded data block. All n, k, m, t here are natural numbers and satisfy the inequality equation: $n \leq 2^m - 1$, that is:

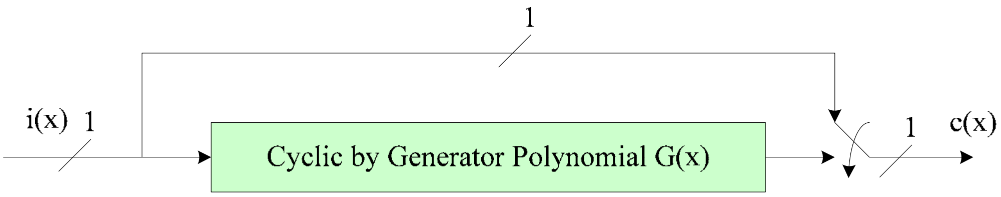
$$2^m - 1 - m \cdot t \geq k \qquad \text{……………………..} \quad (3\text{-}1)$$

In **Step 2**, a generator polynomial for *t*-EC in *GF(2*^{m}*)* can be formed by the following products of minimal polynomials, and can be expressed as:

$$
\begin{aligned}
G(x) &= m_1(x)m_3(x)m_5(x)\cdots m_{2t-1}(x) \\
&= \sum_{i=0}^{m\cdot t-1} a_i \cdot x^i
\end{aligned}
$$

Where, $m_1(x) = p(x)$, which is the primitive polynomial, and $m_{2t-1}(x) = p^{2t-1}(x)$.

Since the degree of the each minimal polynomial in *GF(2*^{m}*)*, $m_1(x), m_3(x), m_5(x), \cdots, m_{2t-1}(x)$ is *m*, the degree of the generator polynomial is thus, $m \cdot t$. The cyclic attribute of the generator polynomial can be implemented by a series of cascaded shift registers. An example of the 1-bit serial BCH ECC encoding circuit is shown as in Figure 3-2, where *i*(*x*) designate the input data stream of the *k* bits user data block and *c*(*x*) is the data stream of $(k + m \cdot t)$ bits encoded block.



Where the Cyclic Architecture by a Polynomial is as below:

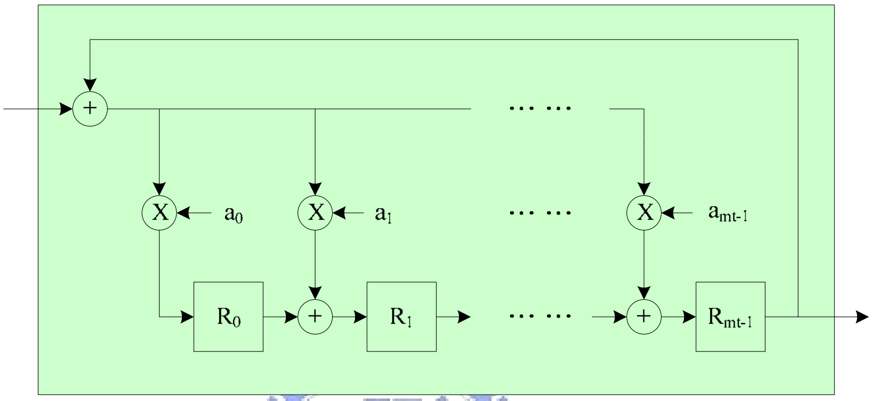


Figure 3-2. Block diagram of the 1-bit serial encoder

In **Step 3**, the syndrome generator polynomials for the BCH ECC decoder can be formed by each minimal polynomial. The Syndromes $S_1(x), S_3(x), S_5(x), \cdots, S_k(x)$ can then be generated from the received data stream, $r(x)$, and the minimal polynomials: $m_1(x), m_3(x), m_5(x), \cdots, m_k(x)$, respectively. In Figure 3-3, the block diagram of the 1-bit serial BCH ECC decoding circuit was shown, where k is equal to $(2t-1)$ for a t-EC BCH code. If all of the syndromes: $S_1(x), S_3(x), S_5(x), \cdots, S_k(x)$ are all “Zeros”, then no bit error happened in the received data stream. That is the received data is error-free.

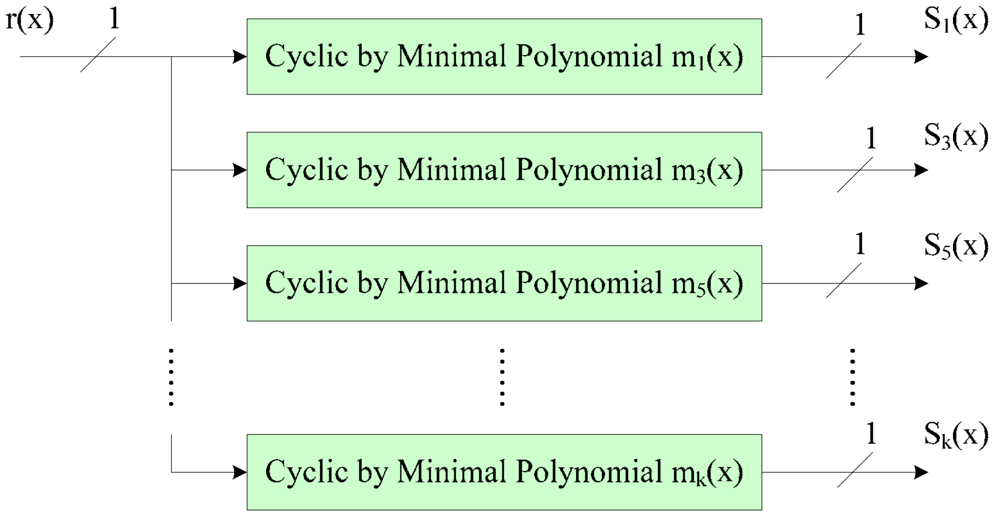

r(x)
1
Cyclic by Minimal Polynomial m1(x)
1
S1(x)
Cyclic by Minimal Polynomial m3(x)
1
S3(x)
Cyclic by Minimal Polynomial m5(x)
1
S5(x)
Cyclic by Minimal Polynomial mk(x)
1
Sk(x)

Figure 3-3. The syndromes generator circuit of 1-bit serial BCH decoder

In **Step 4**, by applying the Berlekamp-Massey algorithm, the coefficients of error locator polynomial can be got from generated syndromes in Step 3 [2].

In **Step 5,** the circuit to find the error locations was constructed by Chien's search algorithm [2, 66-67].

Through Step 1 to Step 5, a serial *t*-EC BCH ECC code can be constructed.

3.2.2 The *t*-EC *w*-bit Parallel BCH ECC Code Construction

For a general *t*-EC BCH Code, the generator polynomial can be expressed as

$$G(x) = m_1(x)m_3(x)m_5(x)\cdots m_{2t-1}(x) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \quad (3\text{-}2)$$

, where $m_j(x),\ j=1,3,5,2t-1$ are the minimal polynomials in *GF(2^m)*.

By expanding the products of $m_1(x), m_3(x), m_5(x), \cdots, m_{2t-1}(x)$, equation (3-2) can be expressed as

$$G(x)=\sum_{i=0}^{m\cdot t-1} a_i \cdot x^i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3\text{-}3)$$

, where a_i's are coefficients of the generator polynomial $G(x)$, $a_i \in GF(2)$. As shown in Figure 3-2, the cyclic operation of $G(x)$ can be implemented by a set of registers and XOR (Exclusive OR) gates. Totally, $m\cdot t$ registers are necessary to fulfill the cyclic operations.

$$G: R \to R,\ R \in GF(2)^{m\cdot t} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3\text{-}4)$$

Where, G can be regarded as a companion matrix.

$$[\mathrm{Reg}]_{i+1} = G\cdot[\mathrm{Reg}]_i + g\cdot D_i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3\text{-}5)$$

The detailed matrix form can be written as follows,

$$\begin{bmatrix} R_{m\cdot t} \\ R_{m\cdot t-1} \\ \vdots \\ R_2 \\ R_1 \end{bmatrix}_{i+1} = \begin{bmatrix} a_{m\cdot t-1} & 1 & 0 & \cdots & 0 \\ a_{m\cdot t-2} & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ a_1 & 0 & 0 & \cdots & 1 \\ a_0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} R_{m\cdot t} \\ R_{m\cdot t-1} \\ \vdots \\ R_2 \\ R_1 \end{bmatrix}_{i} \oplus \begin{bmatrix} a_{m\cdot t-1} \\ a_{m\cdot t-2} \\ \vdots \\ a_1 \\ a_0 \end{bmatrix} \cdot D_i \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3\text{-}6)$$

In serial connection, the matrix equation can be expressed as in equation (3-5). Moreover, for a continuously serial data input stream, {…, D_i, D_{i+1}, D_{i+2}, …, D_{i+w-1}, …}, the *w*-bit parallel formula can be formed as:

$$\begin{aligned}
[\mathrm{Reg}]_{i+1} &= G \cdot [\mathrm{Reg}]_i + g \cdot D_i \\
[\mathrm{Reg}]_{i+2} &= G \cdot [\mathrm{Reg}]_{i+1} + g \cdot D_{i+1} \\
&\vdots \\
[\mathrm{Reg}]_{i+w} &= G \cdot [\mathrm{Reg}]_{i+w-1} + g \cdot D_{i+w-1}
\end{aligned}$$

By *w*-times functional composition of the above equations, an analytical equation was deducted as in equation (3-7).

$$[\mathrm{Reg}]_{i+w} \quad = \quad G^w \cdot [\mathrm{Reg}]_i + \sum_{j=0}^{w-1} G^j \cdot g \cdot D_{i+w-j-1} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \quad (3\text{-}7)$$

Based on the same concept and operations, the induction of the parallel syndrome generator polynomials is similar to the generator polynomials. The encoder and the syndrome generators for *t*-EC *w*-bit parallel BCH ECC were shown in Figure 3-4 and 3-5 respectively.

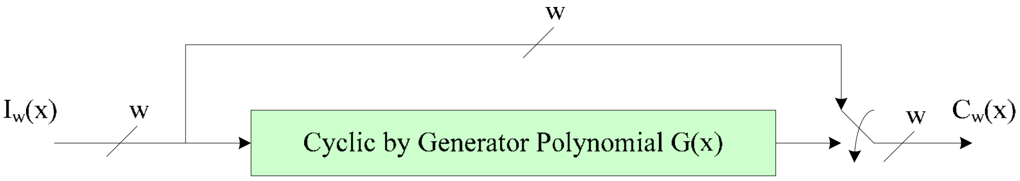


Figure 3-4. The encoder for *t*-EC *w*-bit parallel BCH ECC.

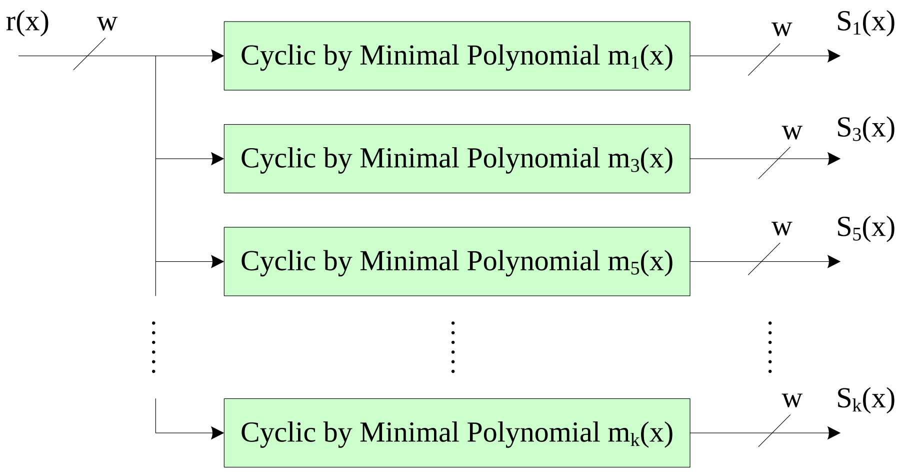

r(x)
w
Cyclic by Minimal Polynomial m1(x)
w
S1(x)
Cyclic by Minimal Polynomial m3(x)
w
S3(x)
Cyclic by Minimal Polynomial m5(x)
w
S5(x)
Cyclic by Minimal Polynomial mk(x)
w
Sk(x)

Figure 3-5. The syndrome generators for *t*-EC *w*-bit parallel BCH ECC.

From the matrix operation of the equation (3-5), a general basic equation for the systolic array processing can be expressed as

$$R_{i+1,j} = a_i \cdot R_{i,mt} + a_i \cdot D_i + R_{i,j-1} \qquad \text{……………………..} \quad (3\text{-}8)$$

, where the boundary conditions are

$$R_{i+1,mt} = a_i \cdot R_{i,mt} + a_i \cdot D_i + R_{i,mt-1} \qquad \text{……………………..} \quad (3\text{-}9)$$

$$R_{i+1,1} = a_i \cdot R_{i,mt} + a_i \cdot D_i \qquad \text{……………………..} \quad (3\text{-}10)$$

Based on the general basic equation of a *t*-EC *w*-bit parallel BCH code, the basic operation module for the matrix array was as shown in Figure 3-6, which contains two AND gates and two XOR gates. The coefficients a_i's are determined by the generator polynomial or the syndrome generator polynomials of the constructed BCH code. To complete the matrix equation (3-6) by the basic operation module, an array architecture was adopted. The array architecture of the *n*-bit input data stream for the BCH generator polynomial or syndrome generator polynomials was shown as in Figure 3-7.

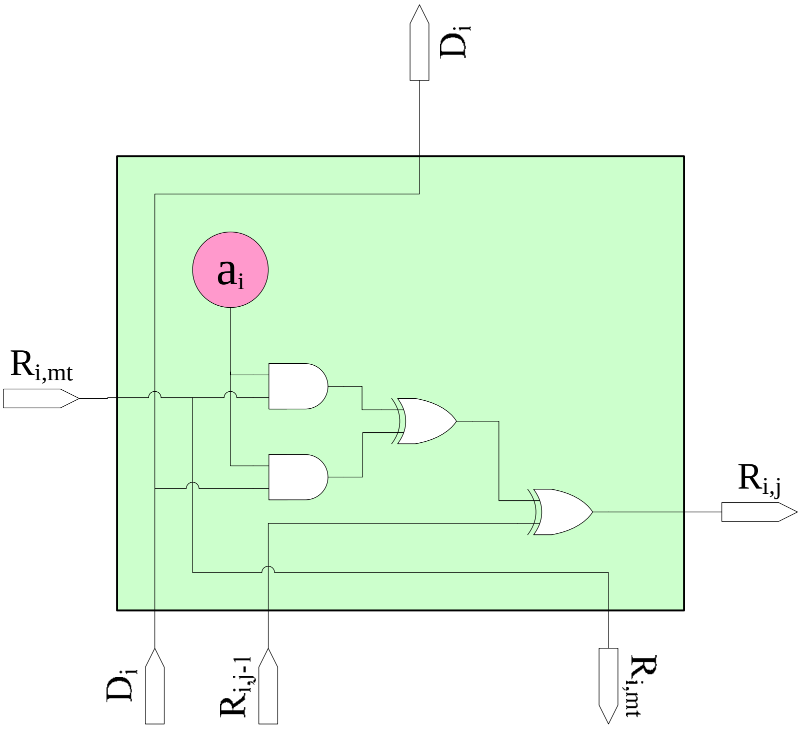


Figure 3-6. The basic operation module for the systolic array

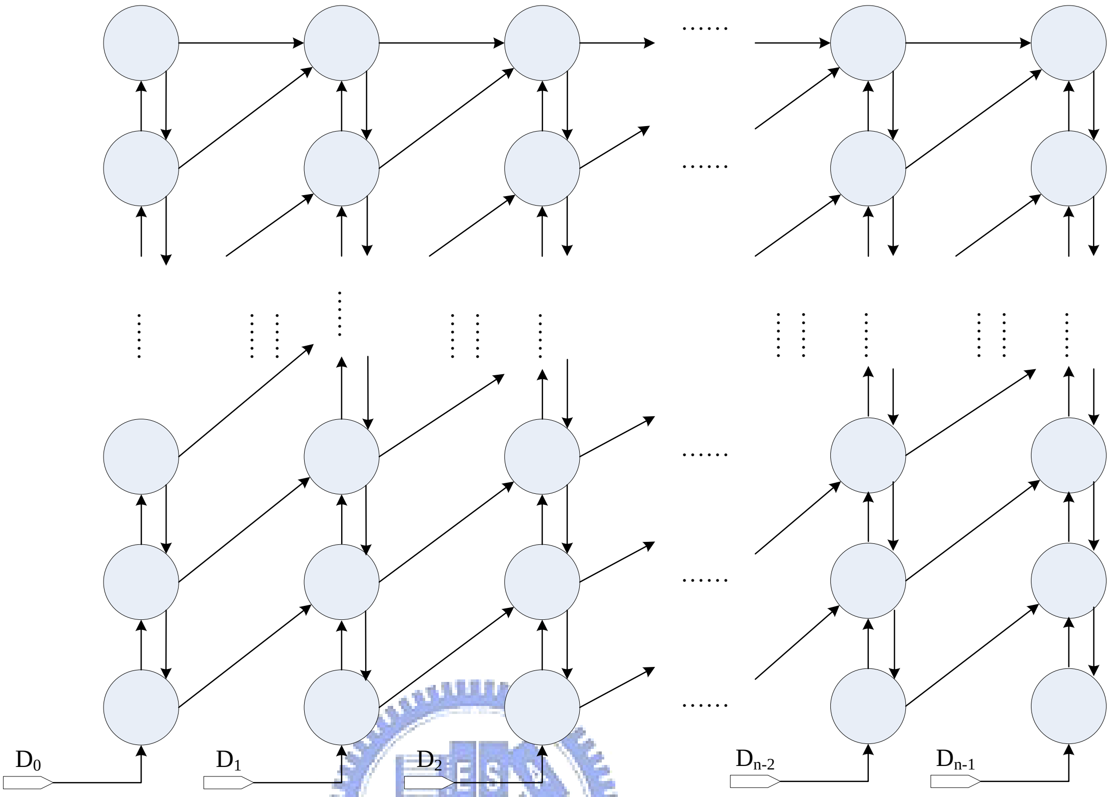


Figure 3-7. The array structure of *n*-bit data stream BCH encoder/decoder

The *w*-bit parallel BCH encoder/decoder circuits can be constructed by a *w*-bit folded of the whole array structure as illustrated in Figure 3-8.

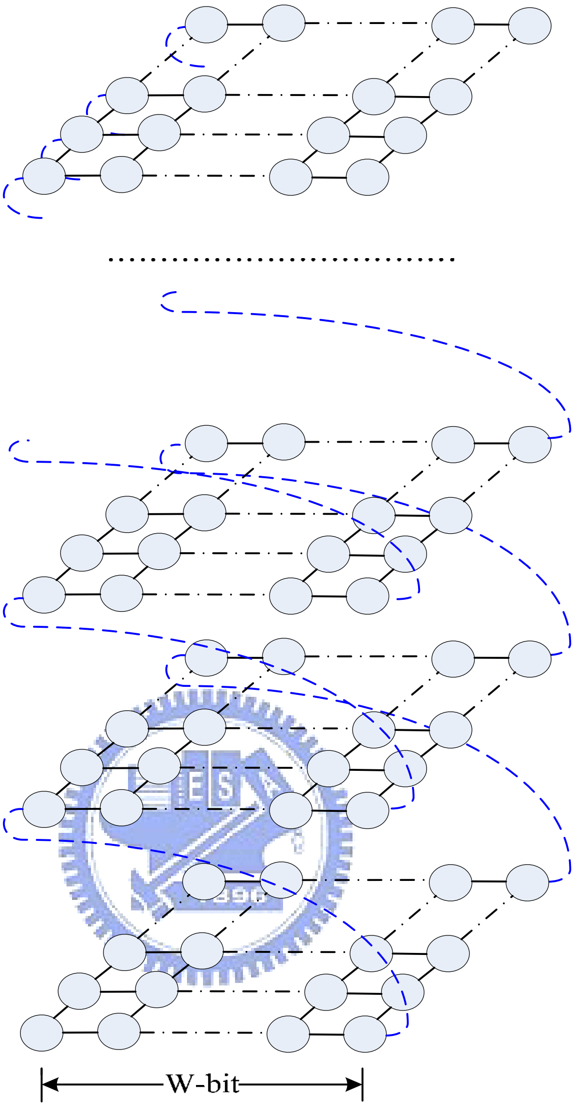


Figure 3-8. The *w*-bit folded structure for *w*-bit parallel operation

By doing the “Z-axis” projection, the array structure of *w*-bit parallel systolic array of a BCH encoder and syndrome generator was as shown in Figure 3-9. The dotted line places in between each w-bit parallel folding array means the registers for data latching it each clock.

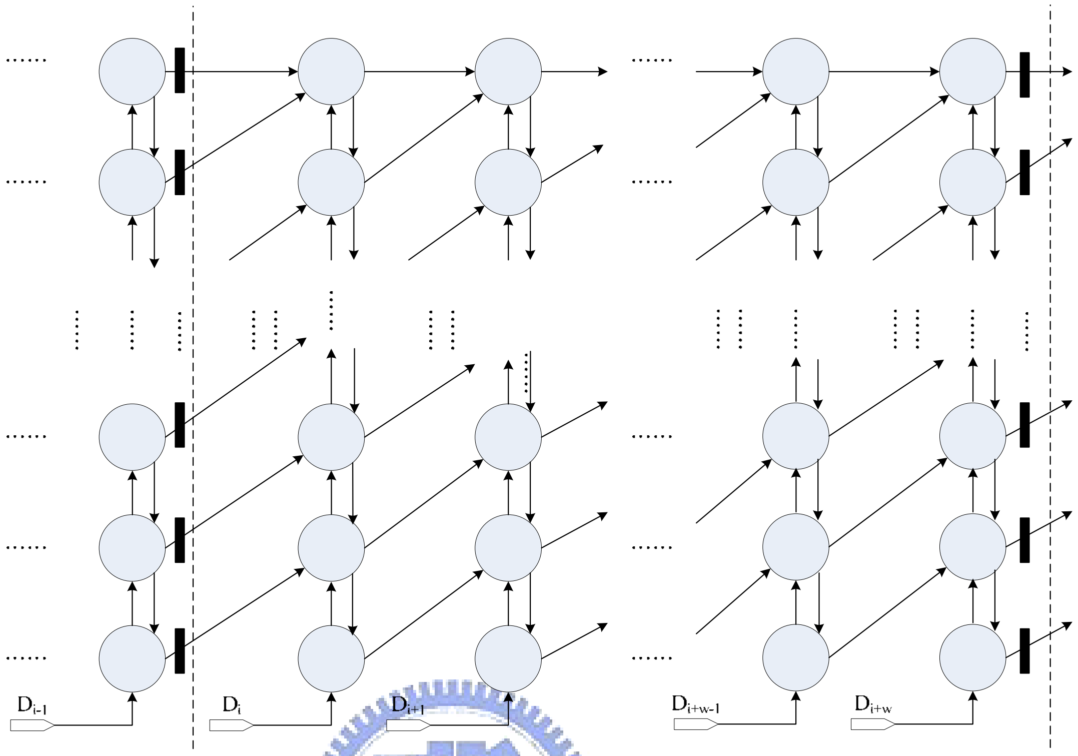


Figure 3-9. The *w*-bit parallel systolic array structure

3.3 A 4-EC 16-bit Parallel BCH ECC Circuit Design Example

For the presented NAND flash controller in this paper, a *4*-EC *16*-bit parallel BCH ECC circuit was implemented to meet the *4*-bit ECC (per 528-byte page) requirement of most MLC flash memory chips. The specifications of the BCH ECC circuit of NAND flash controller for SD/MMC flash memory card were summarized in Table 3-I.

Table 3-I. The specifications of the BCH ECC circuit

Items	**Specifications**
Random bit correction	4-EC; 4 random bits error correction
Data bus width	16 bits
Clock frequency	60 MHz
User data size	512 bytes = 4096 bits

The 4-EC capability for each *4096*-bit user data requires a minimum m of 13 to meet equation (3-1). That is

$$2^{13}-1-13\cdot 4=8139>4096 \qquad \text{……………………….. (3-11)}$$

Thus, a Finite Field GF(2^{13}) was formed by the primitive polynomial

$$p(x)=x^{13}+x^{4}+x^{3}+x+1 \qquad \text{……………………….. (3-12)}$$

And, the minimal polynomials $m_1(x), m_3(x), m_5(x), m_7(x)$ for constructing the generator polynomial are

$$\begin{aligned}
m_1(x)&=x^{13}+x^{4}+x^{3}+x+1\\
m_3(x)&=x^{13}+x^{10}+x^{9}+x^{7}+x^{5}+x^{4}+1\\
m_5(x)&=x^{13}+x^{11}+x^{8}+x^{7}+x^{4}+x+1\\
m_7(x)&=x^{13}+x^{10}+x^{9}+x^{8}+x^{6}+x^{3}+x^{2}+x+1
\end{aligned}$$

Take the vector notation for easy operation as below

$$\begin{array}{lcllllllllllllll} & & x^{13} & x^{12} & x^{11} & x^{10} & x^{9} & x^{8} & x^{7} & x^{6} & x^{5} & x^{4} & x^{3} & x^{2} & x & 1 \\ m_1(x) & = & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ m_3(x) & = & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ m_5(x) & = & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ m_7(x) & = & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

The generator polynomial of the 4-EC BCH ECC, $G(x)$ is thus expressed as

$$G(x) = m_1(x) \cdot m_3(x) \cdot m_5(x) \cdot m_7(x)$$

The 52-bit generator polynomial for BCH(4148,4096) 4-EC ECC was formed as below.

$G(x)$=[1 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1
0 1 0 1 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 1 0 1 1] ………….. (3-13)

The companion matrix G was therefore shown as:

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 \\ 1 & 0 & 1 & 0 & & & & & & 0 \\ 0 & 0 & 0 & 1 & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \ddots & & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ddots & & & & \vdots \\ 1 & 0 & 0 & 0 & 0 & 0 & \ddots & & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 \end{bmatrix}, \quad g = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad \text{………} \quad (3\text{-}14)$$

With the companion matrix G and the vector g, the systolic array circuit can be implemented as statements in the above.

A systolic array was applied to a *t*-EC *w*-bit parallel BCH codec (The encoder and syndrome generator). Table 3-II summarized the comparison between the proposed systolic array *t*-EC *w*-bit parallel BCH code and other prior methods. The systolic array design method can provide a systematic approach for the *t*-EC *w*-bit parallel BCH ECC circuit design with no unbalanced fan-out or clock skew issues as shown in Figure 3-9. A *4*-EC *16*-bit parallel BCH ECC capability for SD/MMC flash memory cards was designed. As the discussion on chip implementations and experiments in Section 8.1 of Chapter 8, the test results show the good performance and low power consumption of this designed circuit.

Table 3-II. Comparison between different BCH implementation methods

Items	**Serial**	**Formulated w-bit Parallel**	**Systolic Array w-bit Parallel**
Unbalanced Fan-out	Yes	Yes	No
Bit rate / Clock	1	1/w	1/w
Area	Small	Middle	Middle
Pipe-lining	No	Difficult	Systematic
Long clock tree	Yes	No	No

3.4 Summary

The systematic circuit construction and implementation of *t*-EC *w*-bit parallel BCH ECC was presented. The *t*-EC *w*-bit Parallel BCH ECC was designed by using systolic array processor, and a *4*-EC *16*-bit parallel BCH ECC was designed for the NAND flash memory controller. With this successful *4*-EC *16*-bit parallel BCH ECC circuit, the presented design method can be easily applied to a general *t*-EC *w*-bit parallel BCH ECC circuit application. The real chip realization and the test results by the experiment show the good performances of the controller chip. The micro-graph of the NAND flash memory controller chip for SD/MMC flash memory card is shown as in Figure 3-10. The detail information was shown in the section 8.1.

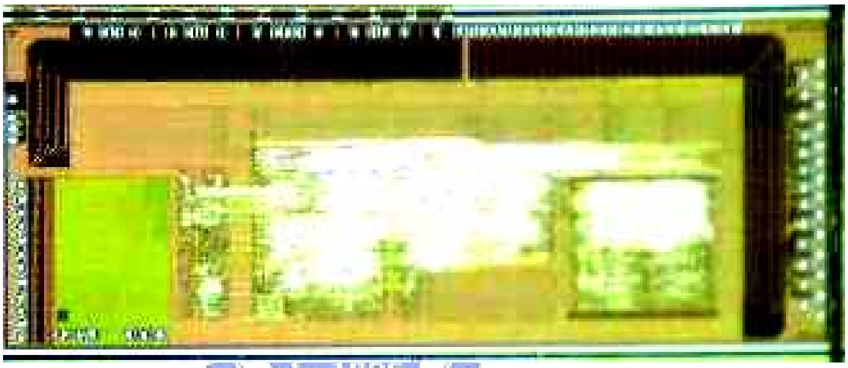


Figure 3-10. The micro-graph of the silicon die of the controller chip with UMC 0.18um Process

Chapter 4 Multi-Mode BCH ECC for Hybrid Multi-Channel Flash Memory

In this chapter, we presented a multi-mode Error-Correction-Code (ECC) architecture for hybrid multi-channel NAND flash memory systems, which optimizes the overall system cost-performance of the flash memory storage systems by leveraging the properties of flash memory structure, performance and endurance in different grade of NAND flash memory. The implementation of the circuit shows 37% of the circuit size and power consumption savings of the proposed architecture. In section 4.1, we talk the introduction on the background and related works. In section 4.2, we describe the proposed multi-mode BCH ECC circuit architecture. In section 4.3, we describe the ECC circuit architecture for hybrid multi-channel flash memory systems. In section 4.4, we show the effectiveness of the proposed architecture by circuit implementation. In section 4.5, we have the summary on the proposed multi-mode BCH ECC circuit for hybrid multi-channel flash memory systems.

4.1 Introduction

The rapid progressing in process technology shrinking and the development of MLC (Multi-Level Cell) technology of NAND flash memory have increased the bit density and lowered the cost per MB of flash memory. The bit density of NAND flash memory was doubled per 12 months in the past years. Nevertheless, the higher density of the NAND flash memory causes the poor performances and less endurance. The cost per MB versus to the performance and reliability of NAND flash memory has become a trade-off. In order to recover the performances and endurances loss of the higher density of the NAND flash memory, there is different operating features introduced and more bits error correction capability requirements during the data accessing [1, 47-51]. Table I shows the summary of the major features among different grade of the NAND flash memory devices [52-58]. In regarding with the NAND flash memory applications, there are some major factors among the different grade of NAND flash memory devices, shown as below:

- ECC Capability: For example, *1*-EC, *4*-EC, *8*-EC, *12*-EC per sector (512 or 528 Bytes).
- Endurances: the more shrinking of technology and higher MLC cell structure, cause more severe problems in endurance cycles.
- Data retention period: it degrades as the operating cycles tend to the maximum endurance cycles.
- Busy time of page program and block erase: it relates to the data accessing performance of NAND flash memory.
- Page format and memory array structure: it defines data accessing method of the NAND flash memory.

- I/O timing in data reading and writing: it relates to the data accessing performance of NAND flash memory.

Table 4-I. The features of different NAND flash memory devices.

Flash Type	SLC	>60nm 4LC	50nm 4LC	40nm 4LC
Max. Monolithic Density	Half of 4LC	8Gb	16Gb	32Gb
ECC Requirement	1 bit per 528 Bytes	4 bits per 528 Bytes	8 bits per 528 Bytes	12 bits per 528 Bytes
Page Format (Bytes)	2048+64	2048+64	4096+218	4096+218
Spare per 512 Bytes	16 Bytes	16 Bytes	27.25 Bytes (218 bits)	27.25 Bytes (218 bits)
Endurance Cycles	100K	10K	5~10K	1.5~5K
Min. NOP[1]	4	1	1	1

Note: 1. NOP means the acceptable Number of Page Program cycles.

As we discussed in chapter 2, to recover the randomly bit errors occurred during the NAND flash memory data accessing, the ECC circuit was designed in the NAND flash memory controller. With randomly bit errors phenomenon during the flash memory data accessing, the BCH code is more suitable than the Reed-Solomon code. The BCH code consumes less redundant data than Reed-Solomon code. The different grade of NAND flash memory requires different ECC capability, as shown in Table 4-I. Thus, the different EC (**E**rror-**C**orrection Capability) of BCH ECC circuit was required for supporting the different type of NAND flash memory. Table 4-II shows the typical information of different EC of BCH ECC circuits.

Table 4-II. The comparison of *4*-EC, *8*-EC and *12*-EC BCH ECC circuits.

BCH ECC Type	*4*-EC	*8*-EC	*12*-EC
Redundant bits	52 bits	104 bits	156 bits
Typical Gate Counts	15K	25K	40K

In some flash memory applications, they are related to file systems level, such as: in the application of the real-time embedded systems [85-88]. In such system-level and large-scale applications, to provide a higher flash accessing bandwidth by multi-channel is useful and meaningful. Meanwhile, the poor performance of lower cost MLC flash memory can be improved by parallel accessing. Furthermore, an optimized cost-performance of the memory cost, system performance and reliability can be achieved by managing the hybrid flash memory systems which are composed of multiple types of flash memory devices. The overall system cost-performance can be optimized by leveraging the memory cost per MB, endurance and performance among the different grade of NAND flash memory chips.

In a hybrid multi-channel flash memory system, the ECC circuit of the NAND flash controller needs to cover all the types of the NAND flash memory in the flash memory system. The more shrinking process technology and more bits per cell of the MLC type NAND flash memory (e.g., 16LC for 4bits per cell), it requires stronger error correction capability of the ECC circuit. Thus, the ECC circuit size in the NAND flash memory controller has become a significant factor. A proposed multi-mode ECC circuit can save the circuit size by sharing the common function units in each mode. The proper ECC mode can be set based on the corresponding NAND flash memory type. In addition, to support multi-channel flash memory system, a proper ECC circuit architecture is also required for the parallel operation in each flash memory channel with the reasonable circuit size.

4.2 The Multi-Mode BCH ECC Architecture

In this section, we presented a multi-mode BCH ECC architecture for hybrid multi-channel NAND flash memory storage system. In Figure 4-1, the overall system functional block diagram of a novel NAND flash memory controller with multi-mode ECC architecture for hybrid multi-channel flash memory storage systems is shown. There are host interface bus and host interface controller to connect with a host system, such as: PC, DSC, cellular phone, notebook computer, etc. The multiple flash sequencer circuits support multi-channel flash memory array. Buffer RAM and buffer manager control the data transportation between the flash memory and the host. The micro-controller unit (MCU) and the code banking architecture provide the environment for firmware execution. The multi-mode BCH ECC was designed for the data integrity of the hybrid multi-channel flash memory array.

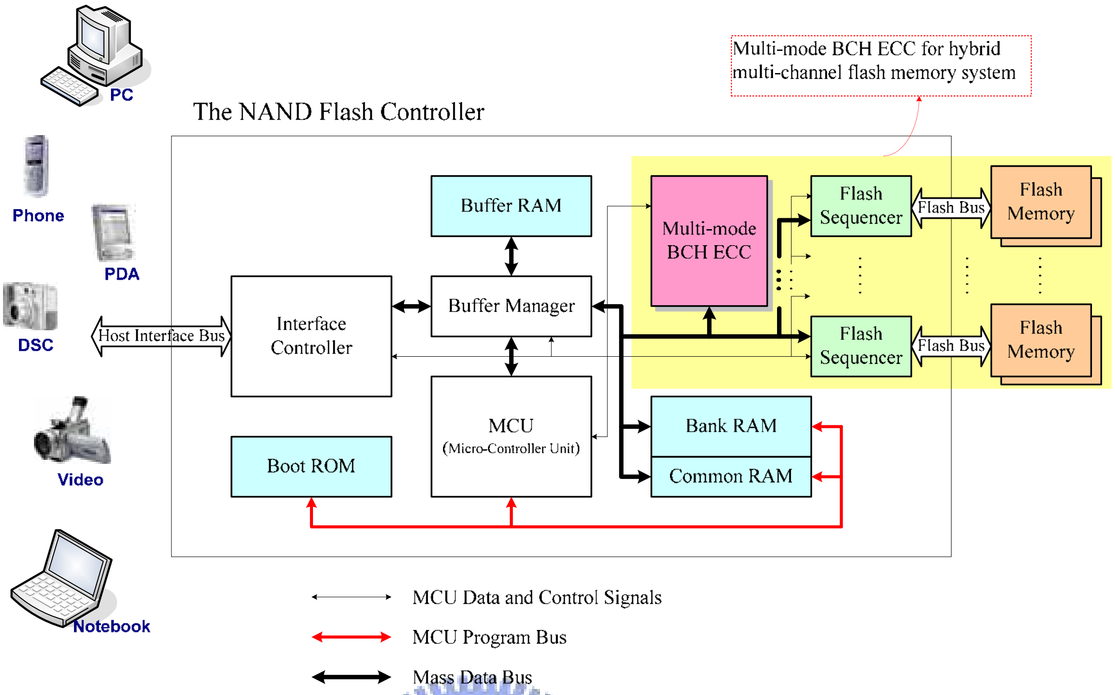


Figure 4-1. The overall system functional block diagram of the controller.

In Figure 4-2, a separated ***Endec*** (Encoder and Decoder circuit for BCH ECC) and error corrector architecture was illustrated. The architecture is designed for cost-performance consideration in supporting the multi-channel flash memory operation. The BCH ECC encoder is used to generate the ECC parities as writing data to flash memory, while the BCH ECC decoder is used to generate the decoded syndromes as reading data from flash memory. We combined the BCH ECC encoder and decoder circuit together since the flash memory reading and writing will not occur at the same time. The *t*-EC *w*-bit parallel BCH ECC ***Endec*** circuit was constructed as the procedures in Chapter 3. The error corrector circuit was designed by three sub-modules: syndromes to error locator polynomial, error location search, and registers file. The "syndrome to error locator polynomial" circuit was implemented by Berlekamp-Massey algorithm. The error location search circuit was implemented by 4-step parallel Chien’s search algorithm. The registers file is shared for the syndrome to error locator polynomial circuit and error location search circuit. Table 4-III shows the typical gate count of the circuit design and synthesis.

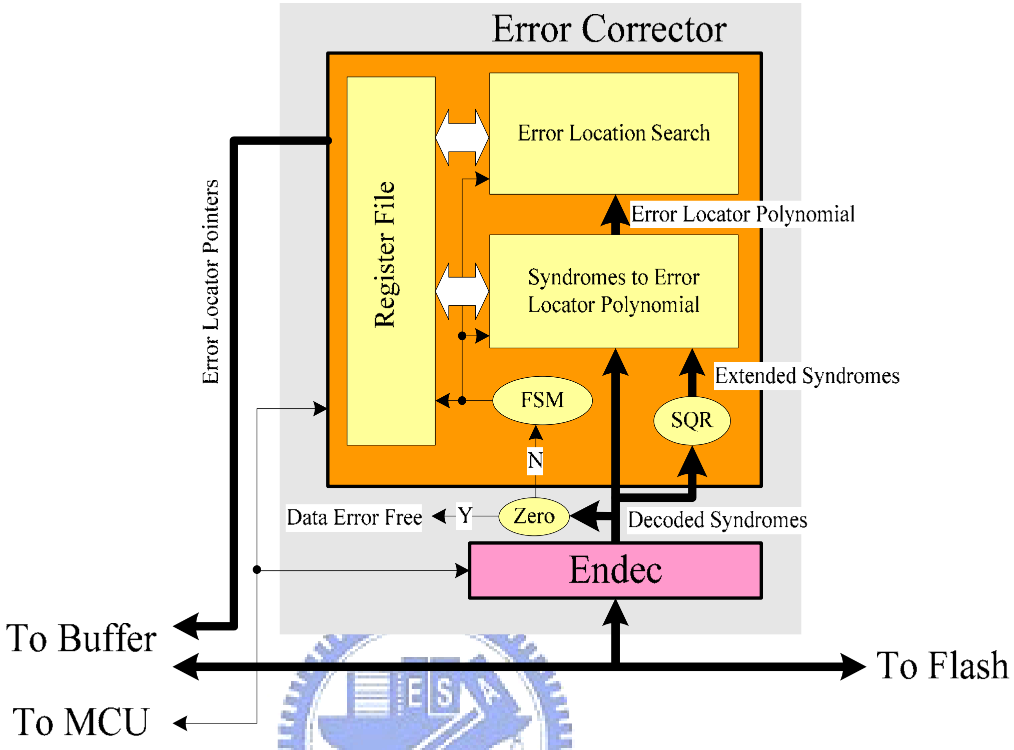


Figure 4-2. The separated ***Endec*** / corrector architecture of BCH ECC.

Table 4-III. Sizes of ***Endec*** and corrector in 12-EC and 8-EC BCH ECC.

Circuit Module	8-EC BCH ECC		12-EC BCH ECC	
	Endec	Corrector	***Endec***	Corrector
Gate Counts	4.4K	19.7K	7.5K	29.5K
Ratio	1.00	4.48	1.00	3.93
Difference	N/A	15.3K	N/A	22K

Based on the circuit architecture in Figure 4-2, supporting multi-mode operation of the BCH ECC circuit was designed on ***Endec*** module and corrector module respectively. The multi-mode operation of BCH encoder and decoder was designed by sharing the registers and adjusting the combinational functions in different mode. Figure 4-3 shows the architecture of the multi-mode ***Endec*** circuit. The encoding combinational functions in each mode: m0, m1, … , mn, were denoted by $f_{EN\_m0}(\bullet)$, $f_{EN\_m1}(\bullet)$, … , $f_{EN\_mn}(\bullet)$, and the decoding combinational functions of each syndrome generation function: S_1, S_3, S_5, … , S_{2t-1} for the minimal polynomials: $m_1(x)$, m3(x), m5(x), …, m2t-1, were denoted by $f_{DE\_S1}(\bullet)$, $f_{DE\_S3}(\bullet)$, $f_{DE\_S5}(\bullet)$, … , $f_{DE\_SN}(\bullet)$. The ***Endec*** circuit is operated cyclically by the selected combinational functions and the synchronous clock of encoding and decoding.

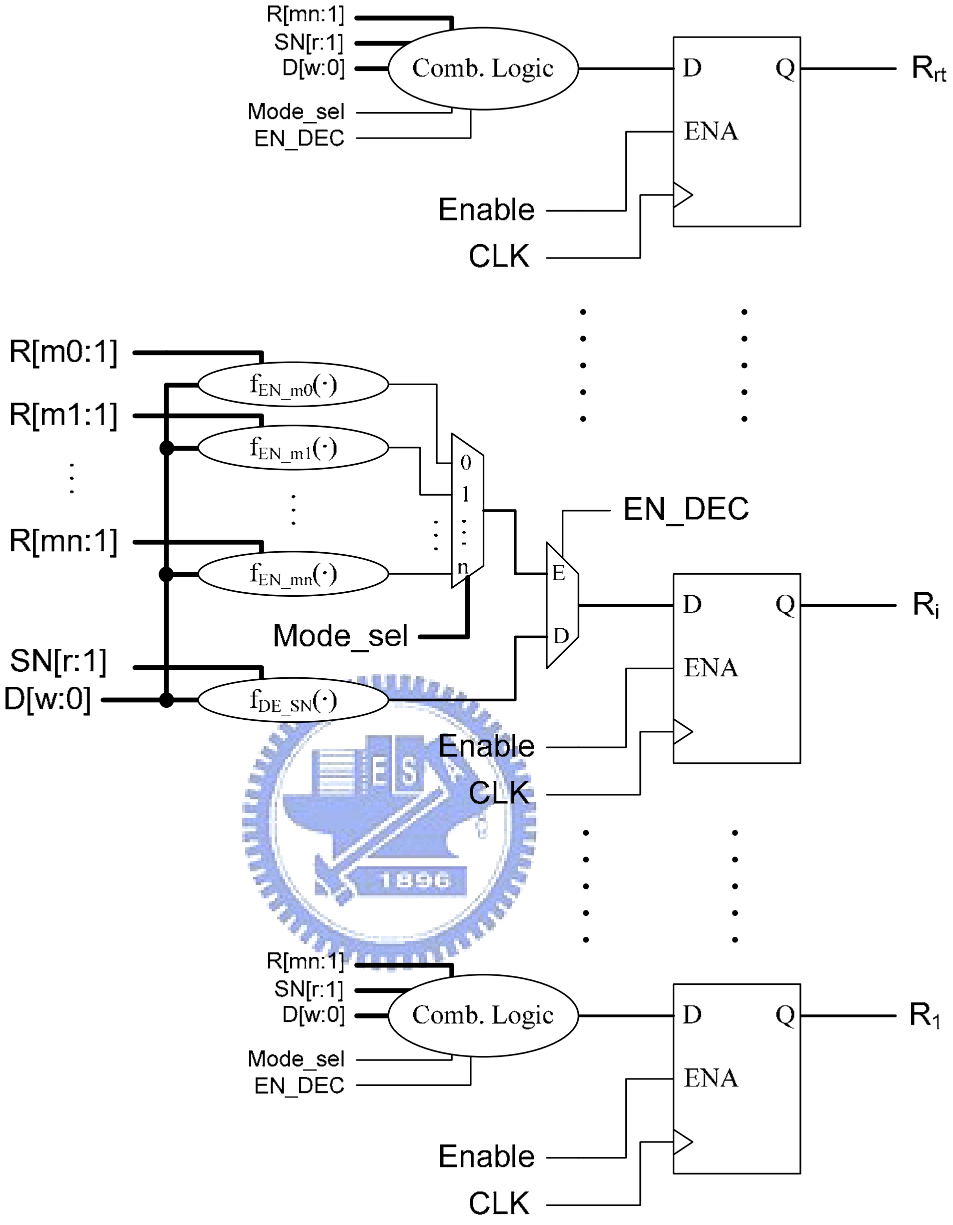


Figure 4-3. The architecture of multi-mode BCH ECC ***Endec*** circuit.

The multi-mode operation of BCH corrector circuit was designed by adjusting number of syndromes and the degree of error locator polynomial in different mode.

Figure 4-4 shows the architecture of the multi-mode corrector circuit. The registers Q_1, Q_2, …, Q_n in the register file were used to store the data during the calculation of error locator polynomial and the error location searching. There are three phases in the error correction process: (i) calculation of syndrome to error locator polynomial (**S** in Figure 4-4.); (ii) search of error locations by Chien's search (**C** in Figure 4-4.); (iii) errors found and store the error location pointers (**F** in Figure 4-4.). As doing the syndromes to error locator polynomial by Berlekamp-Massey algorithm, there is a Galois field MAC (**M**ultiplier and **AC**cumulator) designed and used by the FSM (Finite State Machine). The coefficients for Chien's search were initialized and updated in each clocking step by $f_{CHIEN\_ini}(\bullet)$ and $f_{CHIEN\_upd}(\bullet)$. The error locations pointers were found by evaluating the error locator polynomial, and were stored by latch the values of counter.

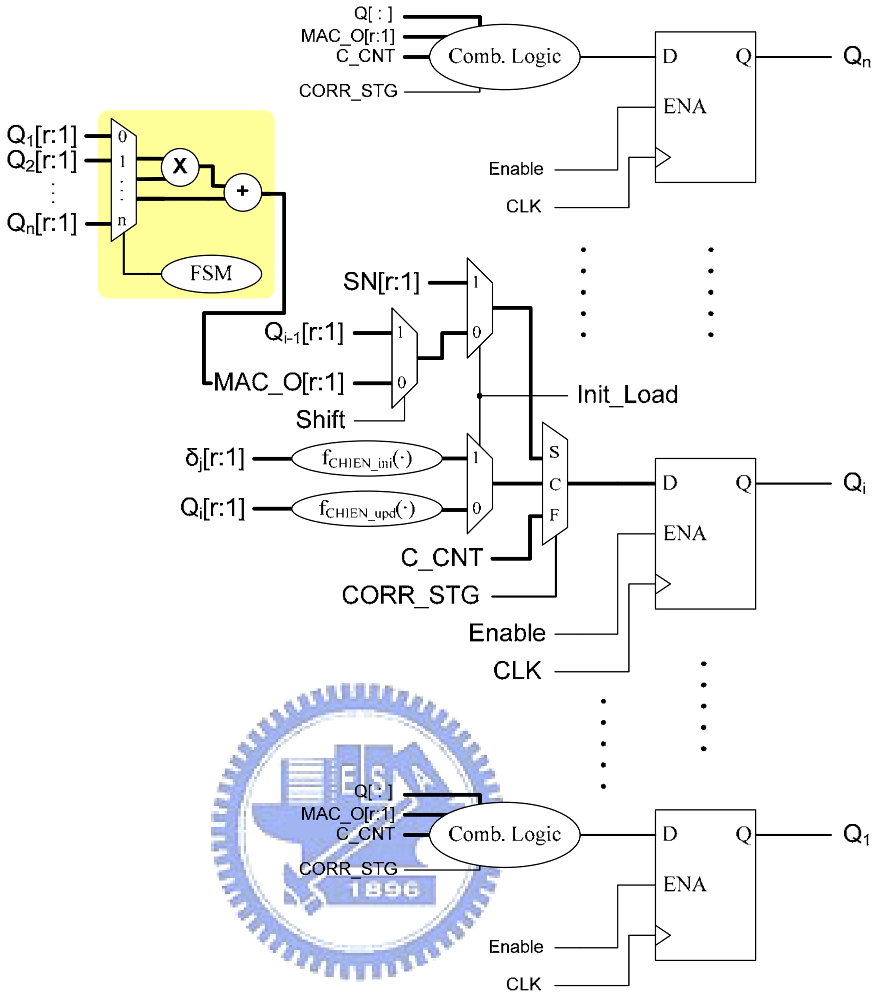


Figure 4-4. The architecture of multi-mode BCH ECC corrector circuit.

4.3 The Architecture for Hybrid Multi-Channel Flash Memory

From the discussion in Section 4.2, the BCH ECC ***Endec*** circuit is smaller than the

BCH ECC corrector circuit. The ***Endec*** was smaller in circuit size and no latency clock required for ECC parities calculation in flash memory writing and syndromes calculation in flash memory reading. In practical flash memory application, the bit error rate is relatively low, thus the corrector circuit is not operated often. To support the multi-channel flash memory operation, the multiple ECC circuits for each channel are necessary for parallel operation. In figure 4-5, multiple BCH ECC ***Endec*** circuits with a common BCH ECC corrector was proposed to support the multi-channel flash memory systems. In the encoding phase of flash memory writing, only the encoding function of the ***Endec*** circuit is active. While in the decoding phase of flash memory reading, the decoding function of the ***Endec*** circuit is active first, and the corrector circuit will be only activated if a non-zero of decoded syndromes detected. Thus, it lowers the circuit size and power consumption of the ECC circuit in the multi-channel flash memory data accessing.

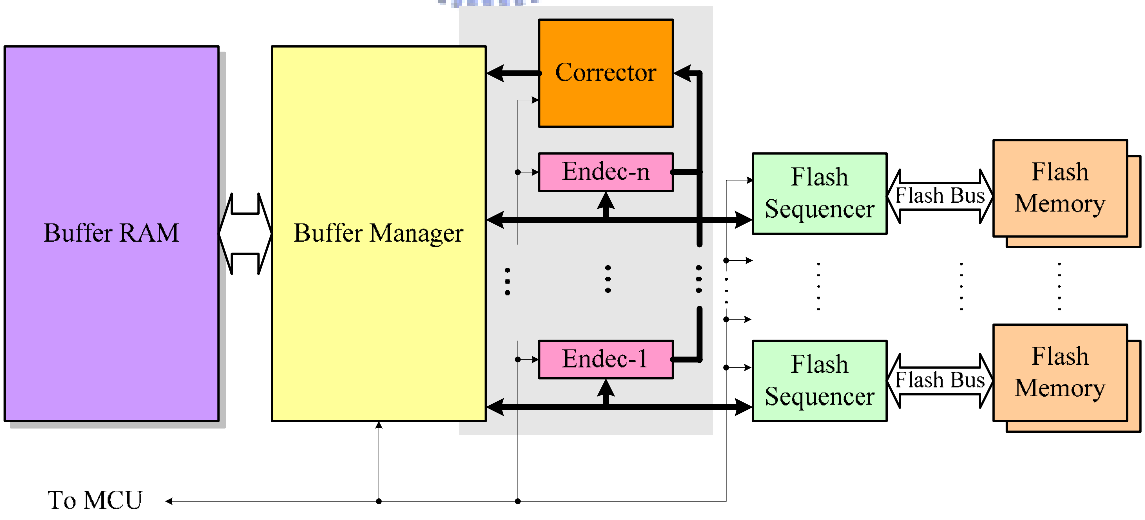


Figure 4-5. The separated ***Endec*** and corrector for multi-channel flash memory system.

4.4 The Circuit Implementation Results

In this section, we show the circuit implementation results of the proposed multi-mode BCH ECC circuit architecture. We used VHDL (**V**ery **H**igh-speed IC hardware **D**escription **L**anguage) for the circuit design entry, and the VHDL codes of the circuit design were compiled by Altera Quartus II software platform (please refer to the web-site: http://www.altera.com), and the design was targeted to the Altera FPGA Stratix II device (EP2S30F672C4) for circuit size comparison. Table 4-IV shows the comparatively cost saving by the proposed multi-mode architecture. The higher of the bit error correction capability, the more significant ratio in cost reduction. For example, the cost saving ratio in 12-EC/8-EC/4-EC triple mode is 37.17%. As for supporting of a flash memory storage system with the four-channel and two-type ECC (12-EC and 8-EC) flash memory, we made the comparison among the conventional BCH ECC algorithm, the inversion-less BCH ECC algorithm, and our proposed architecture in Table 4-V. The significant advantage in area of the proposed architecture was shown as in Table 4-V. As a consequence, when supporting the advanced MLC NAND Flash memory and hybrid multi-channel NAND Flash memory array, the proposed multi-mode BCH ECC circuit architecture lowers the controller chip size by sharing the common logics, and minimizes power consumption by cutting off the un-necessary operations in Error correction process.

Table 4-IV. The cost saving of 12-EC/8-EC/4-EC multi-mode ECC

ECC Type	Comb.[1]	Ratio	Reg.[2]	Ratio	ALUT[3]	Ratio
8-EC + 4-EC	4322	1.00	1749	1.00	4538	1.00
8-EC/4-EC Dual	3383	0.7827	1110	0.6346	4035	0.8892
Cost Saving	939	21.73%	639	36.54%	503	11.08%
12-EC + 4-EC	5691	1.00	1957	1.00	6529	1.00
12-EC/4-EC Dual	4758	0.8361	1588	0.8114	5710	0.8746
Cost Saving	933	16.39%	369	18.86%	819	12.54%
12-EC + 8-EC	7103	1.00	2698	1.00	8157	1.00
12-EC/8-EC Dual	4871	0.6858	1588	0.5886	5826	0.7142
Cost Saving	2232	31.42%	1110	41.14%	2331	28.58%
12-EC + 8-EC + 4-EC	8558	1.00	3067	1.00	9612	1.00
12-EC/8-EC/4-EC Trio	5055	0.5907	1588	0.5178	6039	0.6283
Cost Saving	3503	40.93%	1479	48.22%	3573	37.17%

Note: 1. Comb.: stands for combinational functions.

2. Reg.: stands for registers.

3. ALUT is the "Adaptive Look-Up Tables" of the Altera Stratix II FPGA. It is a kind of metrics for circuit size in the FPGA.

Table 4-V. The comparison among different BCH ECC schemes for four-channel two-type ECC (12-EC and 8-EC) flash memory storage system.

ECC Type	Comb.	Reg.	ALUT	EQ. G/C*
Conventional 12-EC + 8-EC [2-4, 62-64]	13,009	3,540	14,650	~100K
Inversion-less 12-EC + 8-EC [110]	19,480	6,352	23,304	~150K
Proposed dual-mode 12-EC/8-EC	8,578	2,125	9,785	~65K

Note *: EQ. G/C: stands for the approximately equivalent gates count.

4.5 Summary

The hybrid multi-channel flash memory array was constructed to optimize the cost-performance of a flash memory based storage systems by supporting multiple types of NAND flash memory devices, which leverages the bit density, cost per MB, endurance cycles and programming speed of various NAND flash memory types. A proposed multi-mode BCH ECC circuit architecture was designed for the hybrid multi-channel flash memory array. The results of the circuit implementation show the advantage of smaller circuit size and lower power consumption.

Chapter 5 Performance Enhancement

In this chapter, we present the schemes for the performance enhancement on the flash memory storage systems. In section 5.1, we give the briefly introduction on the background and related works. In section 5.2, we propose the multi-channel and multi-buffering schemes for the performance enhancement on the storage systems by multiple flash memory chips. In section 5.3, we propose the flash memory accessing parallelism by the configurations of flash memory chips. In section 5.4, we propose the high bandwidth buffer DMA to provide an enough bandwidth for any typical host interface application. In section 5.5, we present the flash block caching to improve the randomly or small size block data accessing. In section 5.6, a novel high efficiency TD-based (Transfer Descriptor) flash memory sequencer circuit was presented. In section 5.7, we summarize the schemes we discussed for the performance enhancement by hardware architecture.

5.1 Introduction

As we discussed in Section 2.3, the basic characteristics of NAND flash memory confine the operation and data accessing speed on the NAND flash memory. It’s not as convenient as with the popular volatile memory (e.g., DRAM or SRAM), nor as with magnetic storage devices. Thus, a flash memory controller was constructed to compensate the performance weakness of NAND flash memory. In this chapter, we presented some schemes to do the performance enhancement on the controller design for solid-state drives.

The first point we need consider for the performance on flash memory controller would be how fast of the host-side interface will be. That is the maximum bandwidth of the host interface. The maximum bandwidth of the host interface defines the ideal maximum data transfer rate of the flash memory storage systems. For example, in the flash memory cards application, the performance issue is not so significant, since the host interface is comparative slow. However, as for the applications in the solid-state drive, the flash memory performance issue is quite important. The maximum bandwidth for SATA II standard is **300 MB/sec**. To achieve such higher performance in a solid-state drive is not obviously, nor easily. Some techniques implemented on the flash memory controller for the performance enhancement of the solid-state drives will help the overall system performance to reach the maximum bandwidth of host interface, for example, 300 MB/sec in SATA II interface.

There are two factors to measure the performance index on the data transportation: the throughput and the latency. For the sustained sequential large block data transportation, the throughput is more important than latency, since the latency time is just affect the ready to transfer time, and comparatively very minor portion in the whole data transportation. On the other hand, as for the case of randomly small block data transportation, the latency is more significant than the throughput. In the data transfer of a mass data storage device, the data transfer rate for the sequential data accessing (sequential read and sequential write) is like a data transportation of a continuously large block data, so the throughput is our major concern. On the other hand, the data transfer rate of randomly data accessing (random read and random write) is more likely with a data transportation of a small block data, so the latency becomes the major consideration.

5.2 Multi-Buffer and Multi-Channel Architecture

The data transfer rate between the host and the flash memory storage device must be faster enough for some applications, For example, the applications of real-time video recording. Besides, in the application of data transportation from the host to the memory card, the sooner is the better. The faster data can be transferred, the more time could be saved. In general, the write speed of the NAND Flash memory is relatively slower than the host side bandwidth. In MMC 4.0 specification [http://www.mmca.org], the Host side bandwidth in Burst can be up to 52 Mbytes/sec. In SD 1.1 specification [http://www.sdcard.org], the Host side bandwidth in Burst can be up to 25 Mbytes/sec. These host defined interface speed is higher than the bandwidth in writing of the typical NAND flash memory. To increase the system bandwidth to meet the system performance requirement, the multi-channel of flash memory, and multi-buffering of the buffer in the controller was introduced.

5.2.1 Data Transmitting Analysis of Multi-Buffer and Multi-Channel

In Figure 5-1, some cases of the multi-channel and multi-buffering scheme were discussed. In the following discussions, some of the conditions were assumed:

- The flash-side bandwidth is lower than the host-side. That is $t_F > t_H$.

- The overheads of the flash-side is higher than the host-side. That is $t_2 > t_1$.
- The flash side sequencer re-trigger time is lower than the algorithmic handling time. That is: $t_3 > t_4$

Based on the assumptions of practical condition, the following 3 cases of the multi-channel and multi-buffering scheme were analyzed.

The case 1 is the single flash channel and dual buffer architecture. Through the data transferring analysis, we can find the estimated data transfer rate is:

$$TR1 = 512 Bytes/(t_4 + t_F)$$

The case 2 is the dual flash channel and dual buffer architecture. Through the data transferring analysis, we can find the estimated data transfer rate is:

$$TR2 = 1024 Bytes/(t_1 + t_H + t_2 + t_F)$$

The case 3 is the dual flash channel and quad buffer architecture. Through the data transferring analysis, we can find the estimated data transfer rate is:

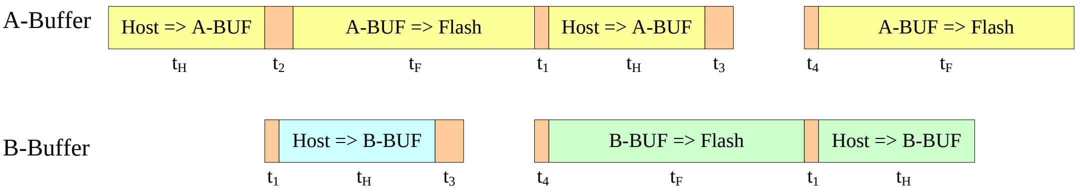
$$TR3 = \begin{cases} 512 Bytes/(t_1 + t_H), if (t_F + t_2) \le (2 \cdot (t_H + t_1) + t_2) \\ 1024 Bytes/(t_4 + t_F), if (t_F + t_2) > (2 \cdot (t_H + t_1) + t_2) \end{cases}$$

In Figure 5-2, the simulation for different flash channel and buffer scheme was compared. In general, the more flash channels and more buffers will get the higher data transfer rate of the flash card. The simulation data was created by the following conditions:

- The host bus is the HS-MMC at 52 Mbytes /sec in burst.
- The overhead of host to buffer transfer, t_1 = 1 μsec.
- The overhead of buffer to flash transfer, t_2 = 3 μsec.
- The overhead of buffer to flash handling, t_3 = 2.5 μsec.
- The re-trigger of buffer to flash transfer, *t4* = 0.8 μsec.

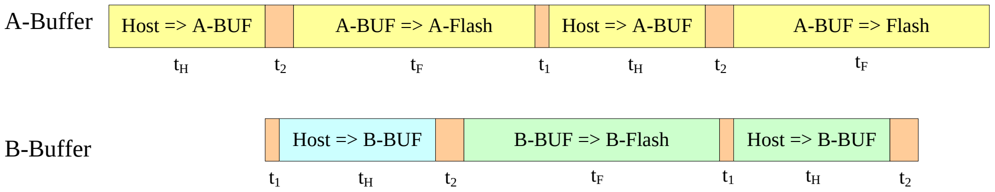
By calculating the estimated data transfer rate in different case versus the ratio of the host bandwidth to the flash bandwidth. The simulation results were shown in Figure 5-2. Although the calculation on the multi-buffer, multi-channel configuration is only for 4 cases, the other cases are quite similar, and it very easy to get the formula as the analysis method we presented in this section.

Case 1: Single Channel, Dual Buffer



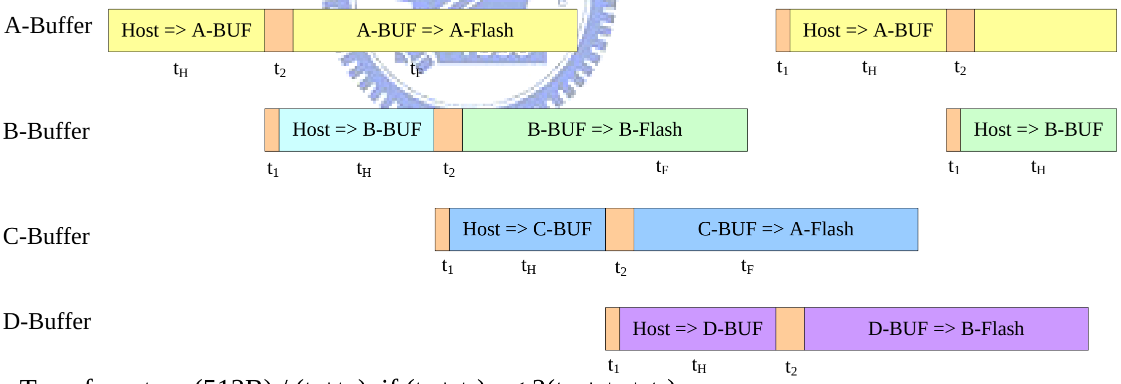
Transfer rate = $512B*2/(2(t_4+t_F))$, assume $t_F+t_4 > t_H+t_3$

Case 2: Dual Channel, Dual Buffer



Transfer rate = $(512B*2) / (t_1+t_H+t_2+t_F)$, assume $t_F > t_H$, $t_2 > t_1$

Case 3: Dual Channel, Quad Buffer



Transfer rate = $(512B) / (t_1+t_H)$, if $(t_F + t_2) =< 2(t_H + t_1 + t_2)$

Transfer rate = $(512B*4) / (2(t_F + t_4))$, if $(t_F + t_2) > 2(t_H + t_1 + t_2)$

Where,

t_1 = The overhead of set Host to Buffer Transfer
t_2 = The overhead of set Buffer to Flash Transfer
t_H = The time of Host send 512 Bytes data to Buffer (Host bus BandWidth)
t_F = The time of Flash write 512 Bytes data from Buffer (Flash Write BandWidth)
t_3 = The overhead of set Buffer to Flash Transfer
t_4 = The trigger overhaed of set Buffer to Flash Transfer

Figure 5-1. The flash channel and buffer scheme analysis.

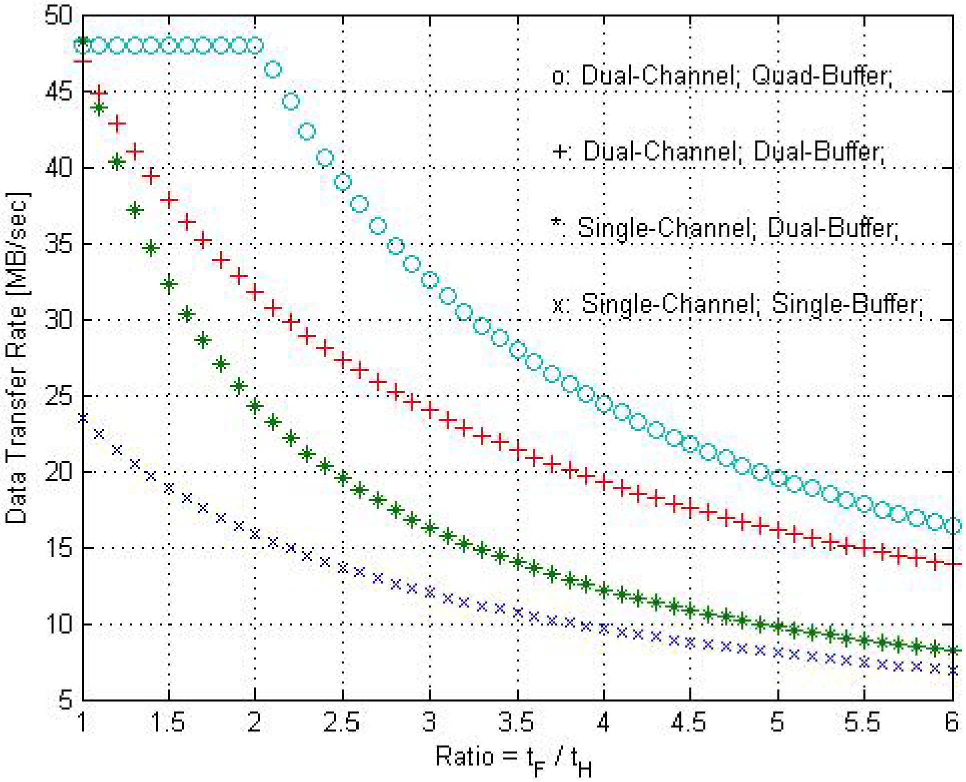


Figure 5-2. The simulation of different flash channel, buffering schemes.

5.2.2 A Quad-Buffer and Dual-Channel Architecture for SD/MMC Card

According to the above analysis on section 5.2.1, the architecture of dual-channel and quad-buffering is suitable for higher performance SD and MMC flash memory cards. The architecture was shown as in Figure 5-3. Two flash sequencer circuits for the flash A-Port and flash B-Port. The quad-buffer was composed by 4 SRAMs with the size of 512Bytes. A buffer manager was designed here to handle the buffer memory switch and control.

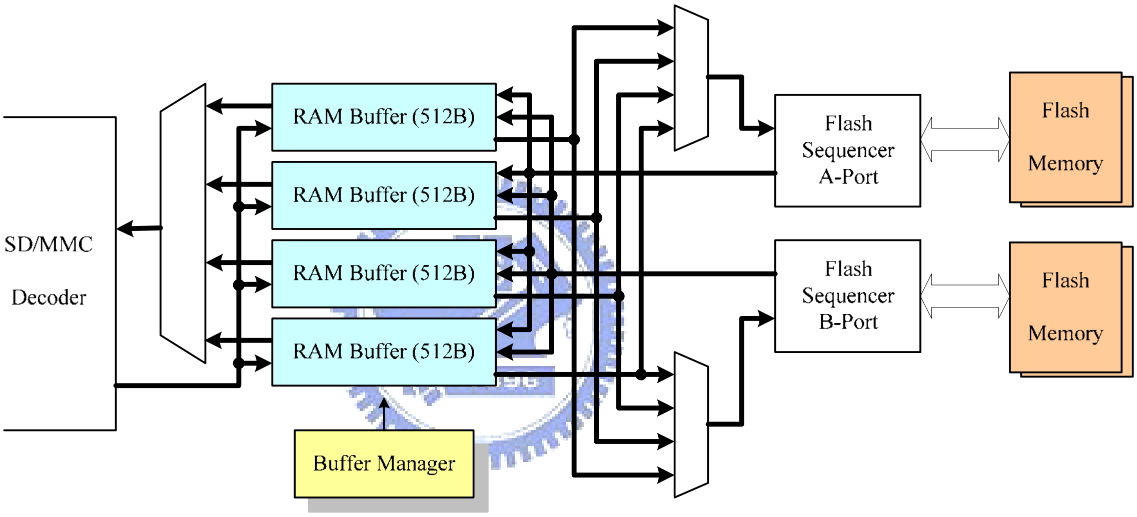


Figure 5-3. The dual-channel quad-buffering architecture

5.3 Parallelism of Flash Memory Data Accessing

The multi-buffer and multi-channel architecture might be very suitable for the flash controller with on-chip RAM. In the small form-factor flash memory card case, the size of the flash memory card is too small to have external bigger RAM buffer, for example DRAM. But in the application case for solid-state drive, the capacity is high and the data transfer rate performance requirement is also high. Thus, a large size of DRAM memory buffer is very useful for the performance enhancement with the flash parallelism, the read cache, and the write cache.

5.3.1 Data Transportation Analysis on Flash Memory Interleave

As our introduction of the flash memory interleave parallelism in Section 2.10, the flash memory interleave in a flash memory channel is similar to the pipeline architecture with 2 stages: the page data in and the page program, as shown in Figure 5-4. The popular naming for such data flow is called as “2-way interleave” in field of flash memory storage systems. In the interleave data flow of a flash memory channel, each flash memory has independent Read/Busy# signals, thus the long time programming or erasing can be being done as the data input is sent to another flash memory device. In Figure 5-4, there are 2 flash memory devices: Flash-A and Flash-B in the flash memory channel. For the data transportation analysis on flash memory interleave, we have the following assumptions:

a) A dual-port buffer is used for the data transportation between host-side and flash-side; the host-side interface has infinite bandwidth. (That means the bandwidth of host-side always larger than the bandwidth of flash-side).

b) The flash page program time (t_{F_PP}) is larger or equal than the flash data input time (t_{F_DI}).

c) The overheads of issuing flash commands and address cycles are negligible.

According to the timing chart analysis, the data transfer rate (*DR*) of the 2-way interleave can be estimated by:

DR = 2*(Bytes of the Block) / ($t_{F\_DI} + t_{F\_PP}$);

In Figure 5-4 (b), it shows the ideal case if the t_{F_DI} is equal to t_{F_PP}. It can gain a double speed compare to non-interleave case.

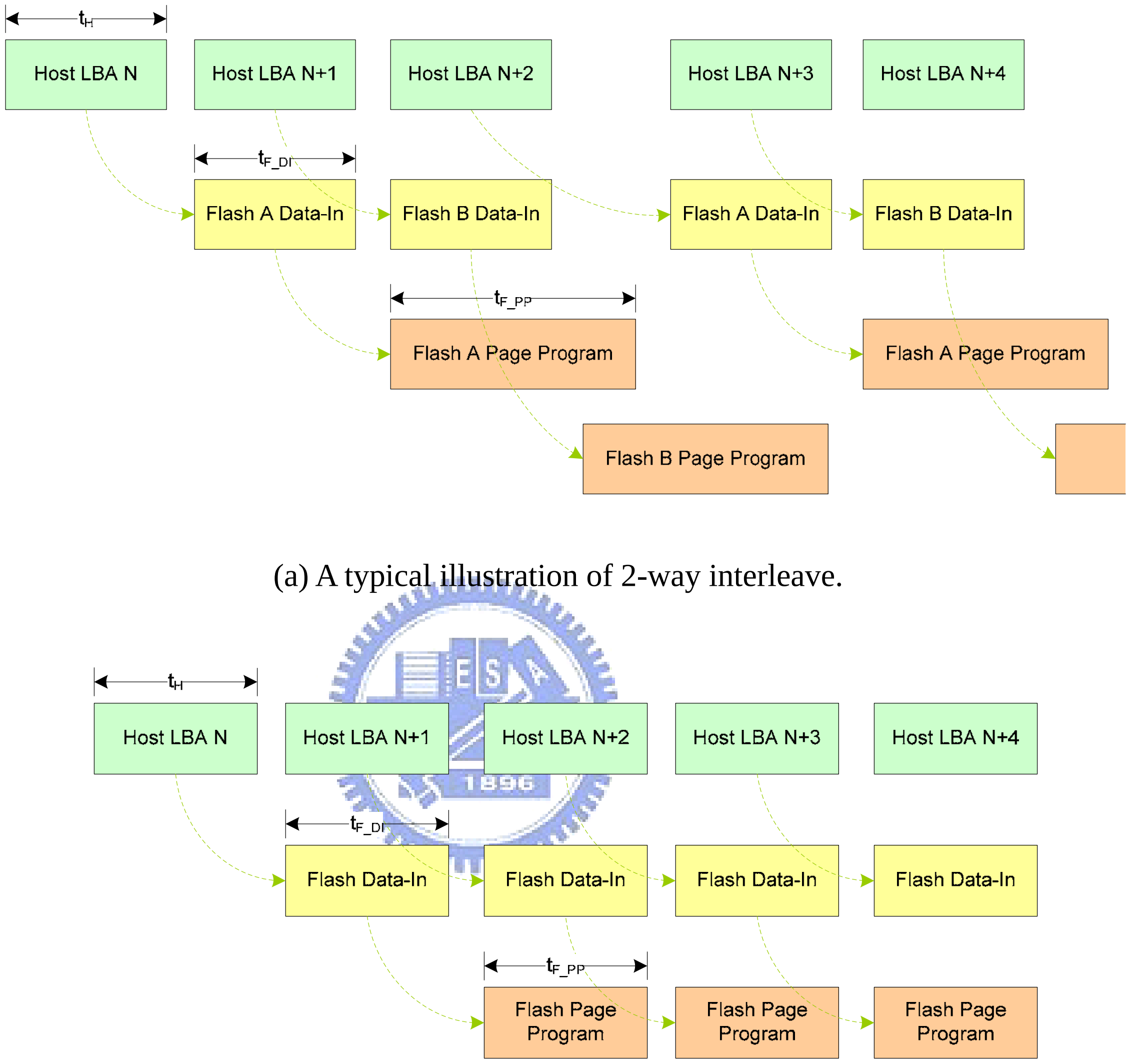
A rule of thumb to have the maximum gain in performance is:

a) Calculate the floor function of (t_{F_PP}/t_{F_DI}), and find the integer number,

$k = \text{floor}(t_{F\_PP}/t_{F\_DI})$;

b) The (k+1)-way interleave will get the maximum data transfer rate as:

DR = (k+1)*(Bytes of the Block) / ($t_{F\_DI} + t_{F\_PP}$);



(a) A typical illustration of 2-way interleave.

(b) An Ideal Case of Interleave Meets the Max. Bandwidth of Host Interface.

Figure 5-4. Illustration of 2-way interleave data flow.

5.3.2 Data Transportation Analysis on Flash Memory Multi-Channel

Comparing to the flash memory interleave sharing the data bus, the multi-channel configuration can send the data input and page program to each channel simultaneously. The analysis on flash memory multi-channel is relative simple. The multi-channel data transfer rate (*DR*) is just the *DR* of single channel times the number of channels, as shown in Figure 5-5. Assume a m-channel flash memory systems, the data transfer rate (*DR*) is:

$$DR = m*(\text{Bytes of the Block}) / (t_{F\_DI} + t_{F\_PP});$$

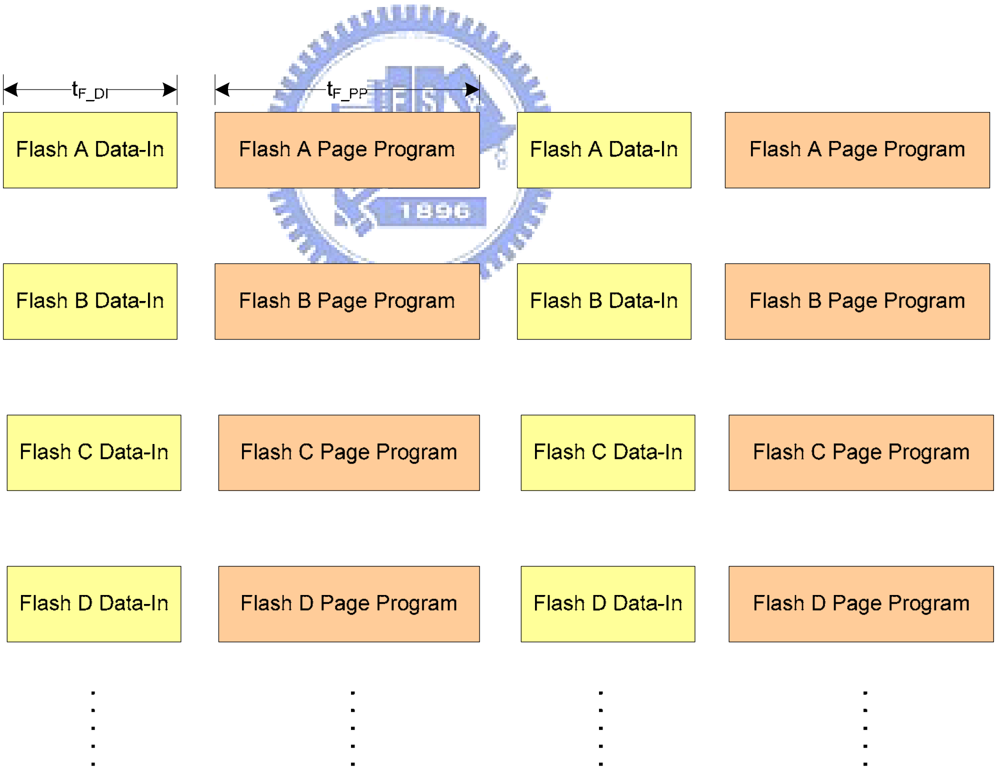


Figure 5-5. Illustration of multi-channel data flow.

Although the data transfer rate of the 2-way interleave and 2-channel is almost the same, we need to point out the 2-channel configuration will have the following 2 advantages:

1) The latency (data transfer start to the first block data recorded) is only half as comparing to the 2-way interleave.

2) The overhead is also the half as comparing to the 2-way interleave, even though we assume it's negligible on the above analysis.

5.3.3 Data Transportation Analysis on Flash Memory Multi-Channel and Interleave

As combining the flash memory interleave and multi-channel, the data transfer rate (*DR*) can be estimated by combining the analysis on section 5.3.1 and 5.3.2.

DR = m*(k+1)*(Bytes of the Block) / (t_{F_DI} + t_{F_PP});

The illustrative data flow of the configuration is as shown in Figure 5-6. The data flow of a dual-channel with 2-way interleave configuration was shown in Figure 5-6.

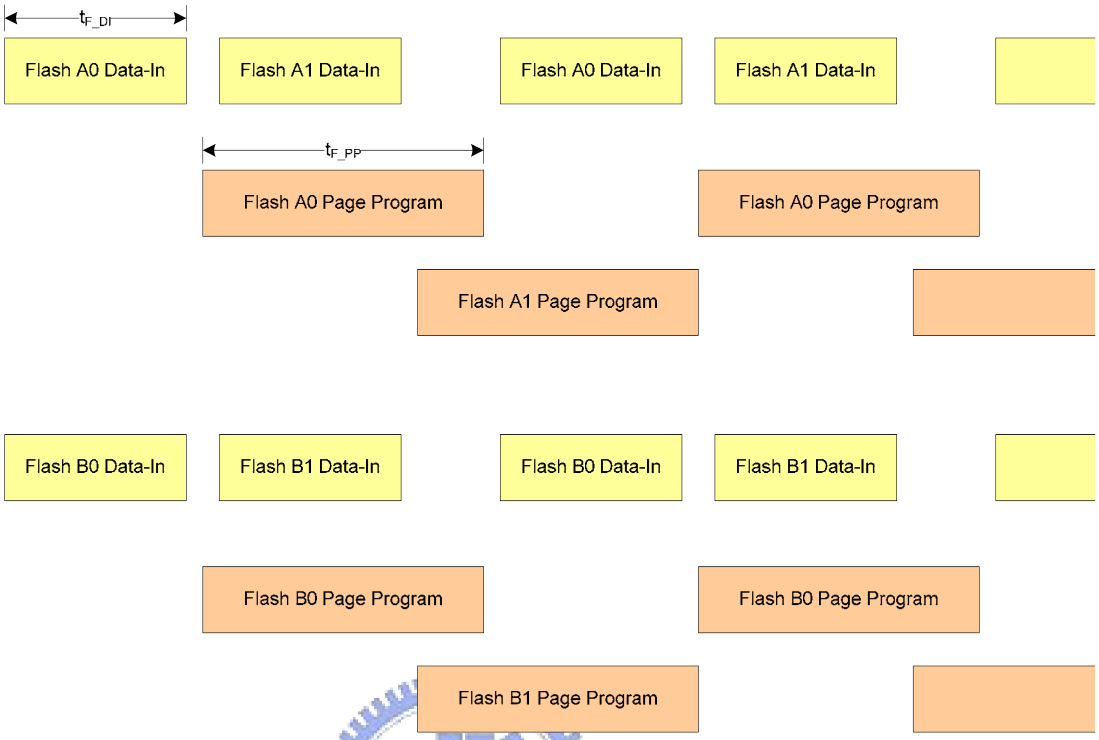


Figure 5-6. Illustration of multi-channel and interleave data flow.

5.4 High Bandwidth Buffer DMA

Architecture of high-bandwidth Buffer DMA was presented in the section. In Figure 5-7, a single block RAM but with high bandwidth Buffer was controlled by a DMA controller (also called as the buffer manager). The accessing timing window of the buffer RAM was controlled by a FSM (Finite State Machine), thus, the bandwidth budge was shared by the multiple sources. The proposed specification is as:

1) The host-side has the guaranteed bandwidth budget of half of the

maximum bandwidth. The other half of the bandwidth was shared by ECC corrector, MCU, and flash channels.

2) The priority order of the half bandwidth sharing is as:

ECC corrector => MCU => flash memory channels.

The state diagram of the finite state machine is as shown in Figure 5-8. A typical timing window example of the buffer RAM accessing is as shown in Figure 5-9.

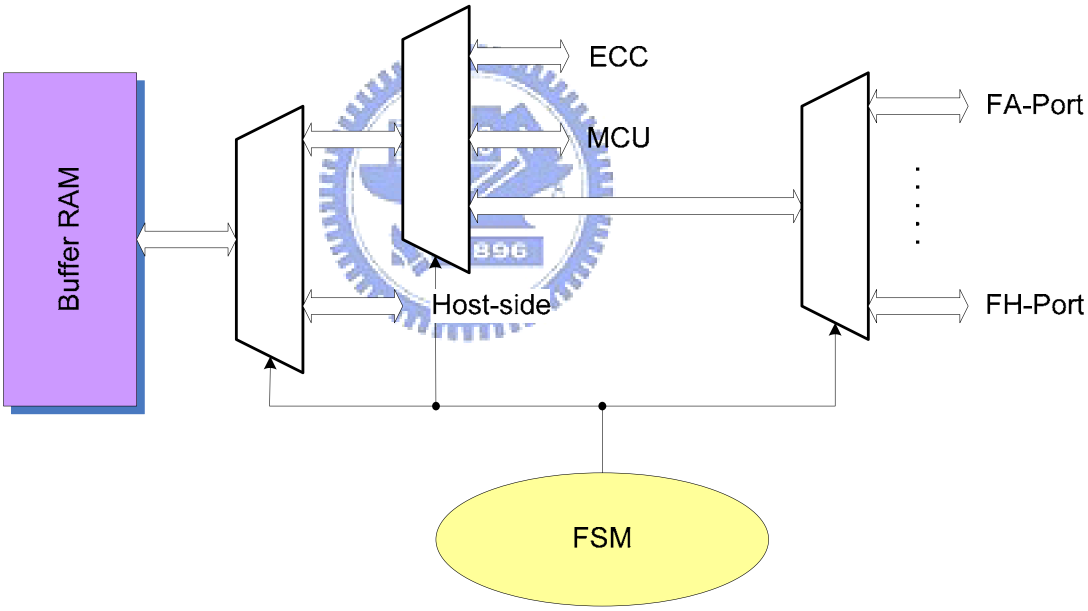


Figure 5-7. The block diagram of timing widowing multi-port buffer RAM.

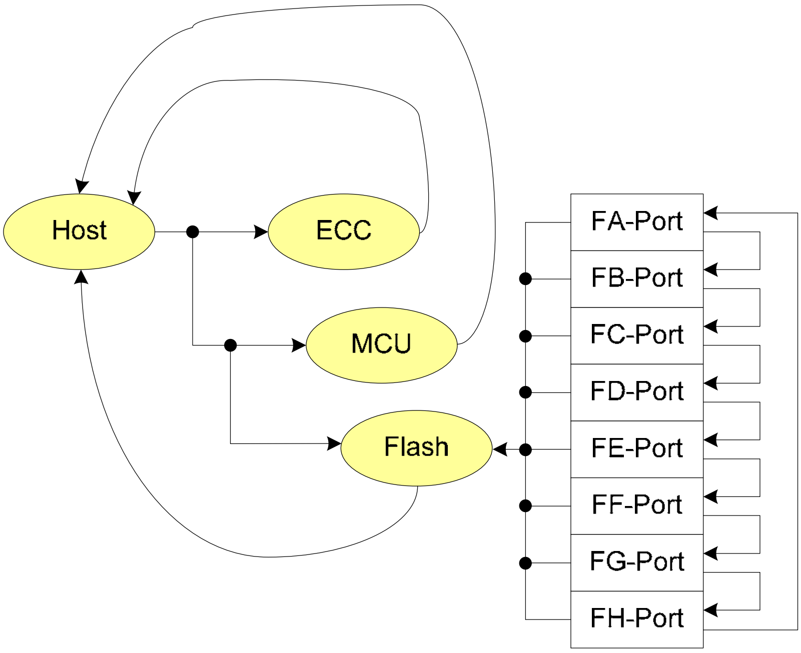
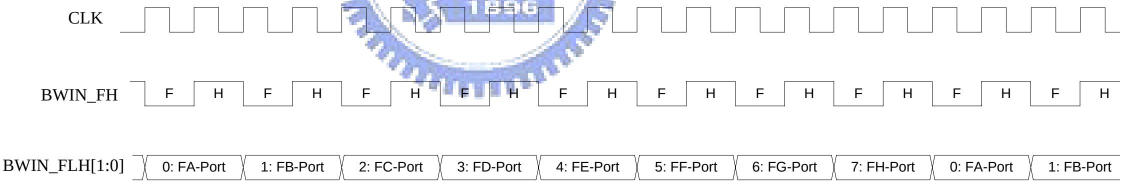


Figure 5-8. The state transition diagram of the finite state machine.



Note:

1. BWIN_FH: Flash-side (0) Accessing or Host-side (1) Accessing.
2. The Buffer Output Valid after Ta (Access Delay of RAM) after the Accessed.
3. BWIN_FLH: Buffer Window for Flash-side Accessing.
   a) 0: Flash A-Port.
   b) 1: Flash B-Port.
   c) 2: Flash C-Port.
   d) 3: Flash D-Port.
   e) 4: Flash E-Port.
   f) 5: Flash F-Port.
   g) 6: Flash G-Port.
   h) 7: Flash H-Port.
   i) 8: ECC Read.
   j) 9: ECC Write. (Do the in-buffer Error Correction)
4. ECC Correction, and MCU accessing is intervened in prioroty.

Figure 5-9. A typical timing window of Buffer RAM Accessing.

5.5 Flash Block Caching

The Flash Block Caching (FBC) is using some reserved flash blocks to cache the data as the data block size is much less than the flash memory block size in the write sequence from host-side to flash side. For example, in the case of the DOS file system is updating the FAT or directory; the data block size is relative small, and it is updated by constant repetitions. And such operations are quite often as copying or updating files to the storage device. As shown in Figure 5-10, the most frequently accessed page data is cached in a specified cache block. The write data time can be saved just as the page program time, instead of (page program time) + (data copy time) + (block erase time).

The flow sequence of 1 page data write without flash block caching is:

1) Find a spare block (clean block), and program the data into the page.

2) Copy the effective data to the spare block.

3) Update the L2P table, and the spare block become a used block.

4) Erase the dirty block, and let it become a new spare block.

The flow sequence of 1 page data write with flash block caching is:

1) Program the data into the new page of cache block.

2) Update the L2P table.

The effectiveness of the flash block caching algorithm was examined by a data storage device performance benchmarking tool in the PC platform, called HD Bench (web-site, http://www.hdbench.net/). In the copy files operation of DOS FAT file systems, the write data time to update the FAT and directory can be saved by the flash block caching algorithm. In our experiments, we observed if the data block size is less than or equal to 8 sectors (1 sector = 512 bytes), the updating frequency of block data is comparatively often. Thus, we treat the data block size less or equal than 8 sectors as the frequent accessed data. And the frequent accessed data will be handling by the flash block caching algorithm. The performance test results was shown as in Table 5-I. In Table 5-I, the significant write performance enhancement by 1.434 and 1.376 was verified of FBC enabled as comparing the FBC was not enabled in the Microsoft Windows XP environment of FAT16 and FAT32 file format respectively.

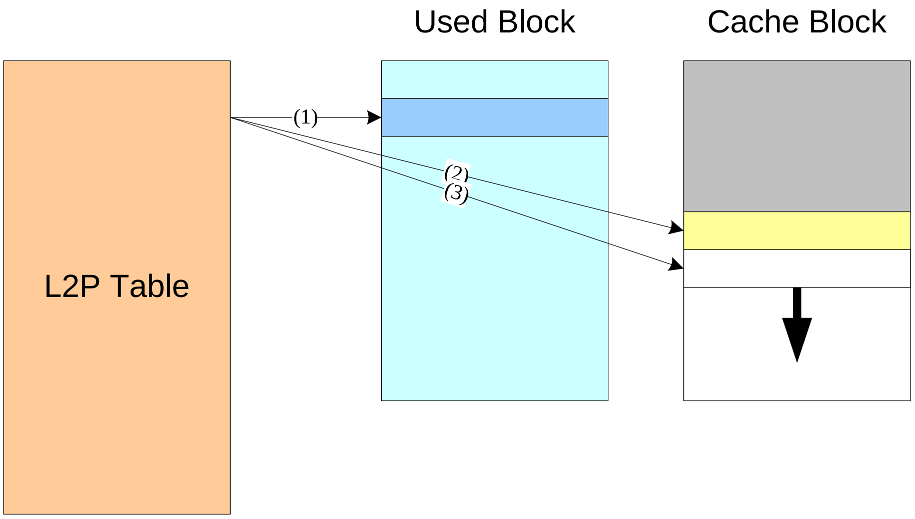


Figure 5-10. Illustration of flash block caching.

Table 5-I. The performance enhancement by flash block caching

File System Type	Sequential Write Performance[1]		Performance Gain
	Without FBC[2]	With FBC	
Windows XP FAT16	13.24 MB/s	18.98 MB/s	1.434
Windows XP FAT32	11.20 MB/s	15.41 MB/s	1.376

Note:

1. The sequential write performance was measured by the HDBench Version:3.4 (http://www.hdbench.net/) Benchmarking Tool for data storage devices in PC platform.
2. With FBC means the flash block caching was enabled; without FBC means the flash block caching was disabled.

5.6 Transfer Descriptor (TD) -based Flash Sequencer

A Transfer Descriptor (TD) is a data structure which contains the data transfer information for a transaction. The total transfer of a transaction is executed by executing of a series of TD. In Figure 5-11, a TD-based flash Sequencer was presented. Series of TDs were stored in the TD Buffer, which were pre-coded by the MCU. After the TD execution command (Run TD) was issued from MCU, the TD processor loads the TDs from the TD buffer, and then executed and reported the execution status. The TD processor controls the flash access controller, which is designed to issues the controller signals and transferring the data between the flash memory and the DMA for the buffer RAM.

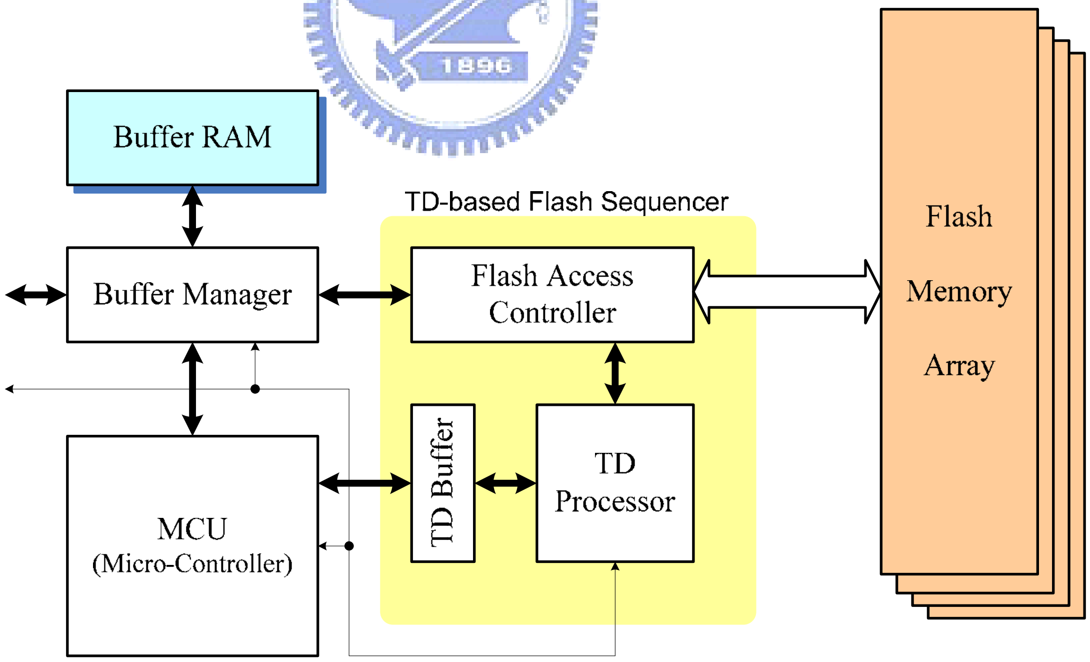


Figure 5-11. The functional block diagram of the TD-based flash sequencer.

There are 3 stages of the TD processor designed: load LD, TD execution, and status report. A 3-stage pipeline structure was adopted in the TD processor design, shown as in Figure 5-12.

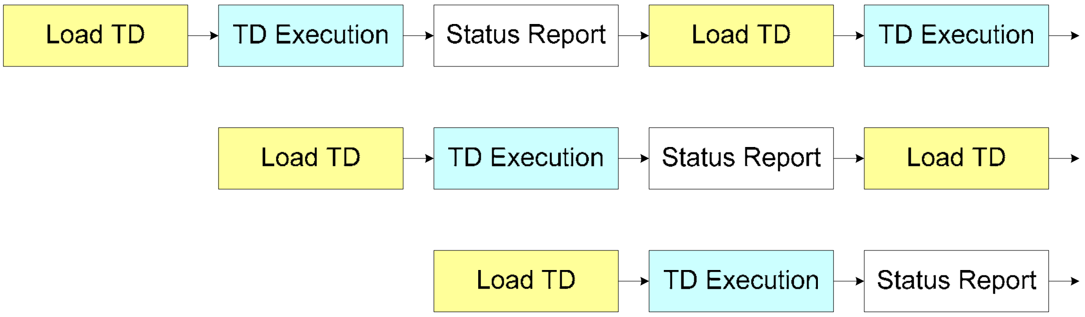
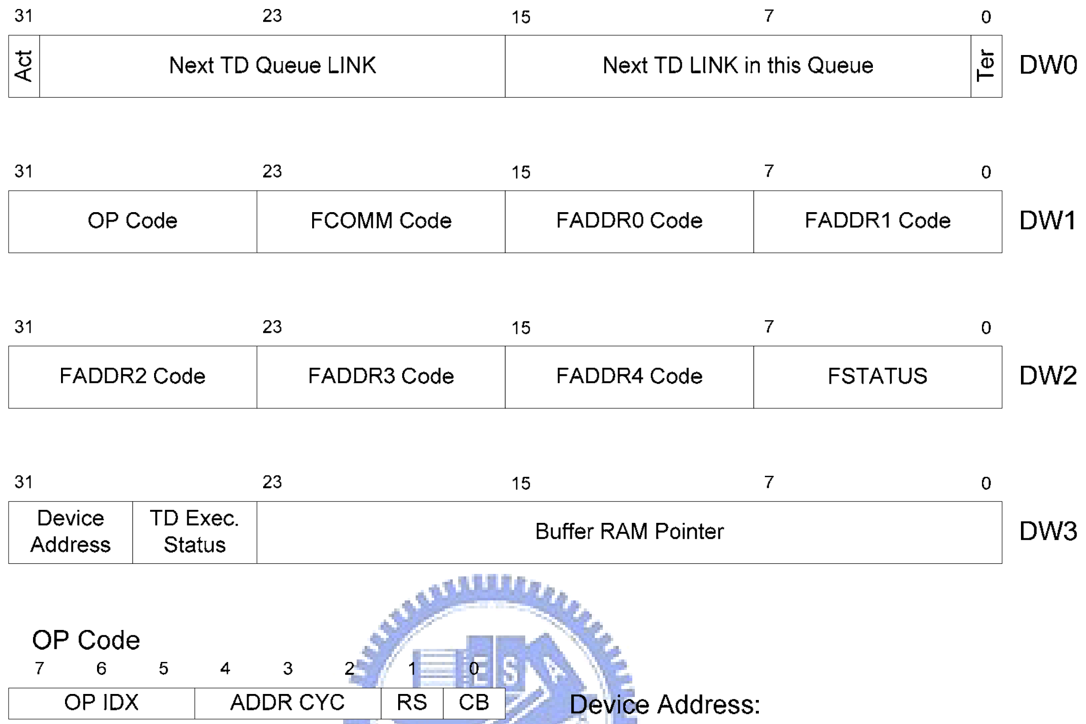


Figure 5-12. The 3-stage pipeline structure of the TD processor.

A typical data structure of TD is shown in Figure 5-13. The TD occupied 4 double words (DW; the Double Word is a 32-bit word). It contains the information of the TD linkage in the firs DW. The second and third DW contains the information for flash memory accessing. The 4th DW contains the buffer DMA information.

A typical linkage of queue of TD execution is shown as in Figure 5-14. The execution of the TD structure contains 3 linked TD queues. The first Queue has 5 TDs, the second 6 TDs, and the third 4 TDs.



Device Address:
Flash device address in the channel 0~15

OP IDX 000: Data Read
OP IDX 001: Data Write
OP IDX 100: Command Only
OP IDX 110: Command Address
OP IDX 010: Address Only

TD Exec. Status:
TD Execution Successfully Done,
ECC Correctable Errors, Un-correctable Data Error.
Flash Program Fail, Erase Fail

ADDR CYC: Address Cycles 1~5
RS: Read Flash Status in the TD.
CB: Check Flash Busy to Ready in the TD.

Figure 5-13. The data structure of the TD for flash memory accessing.

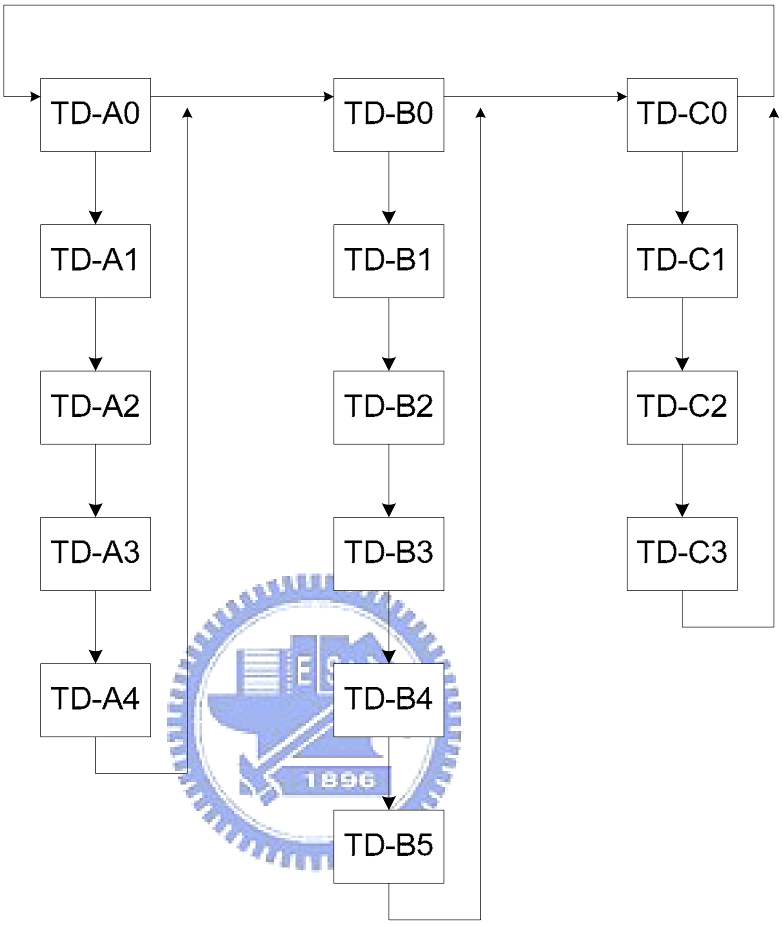


Figure 5-14. A typical linkage of TD queue in TD execution.

5.7 Summary

The goal of the performance enhancement of the controller for the flash memory system is to maintain a fast enough speed to meet the maximum bandwidth of host interface. Thus, the bandwidth of the buffer RAM need be wide enough. The bandwidth of the buffer RAM can be improved by the faster accessing speed or wider bus width. A multi-port accessing of buffer RAM can be implemented by spatial (multi-buffering) or timing (timing window control by FSM). An efficient flash block caching algorithm is presented to resolve the bottleneck as the small block data accessing is required. A TD-based flash sequencer provides high efficiency operation on the multi-channel with flash memory chips interleave support for the flash memory array. Through this combinations of the circuit architecture and system firmware operating, the data transmission between the host and the flash memory can be maintained as the highest bandwidth of the host interface can provide.

Chapter 6 CPRM Implementation

A hardware accelerator-based CPRM circuit was presented in this chapter. In section 6.1, we give the brief introduction on the related works. In section 6.2, In section 6.2, we briefly describe the operations of the SD-CPRM. In section 6.3, we presented our circuit architecture. In section 6.4, we discuss the verification and validation of the SD-CPRM circuit. Finally, we have the summary in Section 6.5.

6.1 Introduction

CPRM (**C**ontent **P**rotection for **R**ecordable **M**edia) is a very important class of the data cryptography, which are used as the digital data content protection for the data transferring among the versatile information appliance devices. In the data cryptography, there are publications presented for discussing the data encryption and decryption, and public key cryptography algorithm design [4, 25, 36, 91-93, 108]. For example the popular cryptography mechanism as: DES (**D**ata **E**ncryption **S**tandard) algorithm, RSA (stands for the names of the 3 inventors: **R**ivest, **S**hamir, and **A**dleman) algorithm, and ECC (**E**lliptic **C**urve **C**ryptography) algorithm are developed to increase the difficulties to crack the encrypted data and decrease the probability of the secure data been destroyed. In the history of the developing of cryptography, the public-key cryptography might be the most important discovery. Moreover, we can say it might be the only revolution from the long years used “classic” cryptography into the “modern” cryptography. From the classic cryptography development point of view, almost all of the cryptosystems are using the fundamental data scrambling, symbol replacement, and data harsh function. There was a famous data encryption code developed in IBM laboratory, called Lucifer algorithm, it became a USA national standard of data encryption codes, called as DES. Besides the algorithm development of cryptosystems, there are some hardware circuit implementation have been presented [94-96]. The high-speed VLSI circuit architecture has been discussed to satisfy the increasing of computation due to more secure functions required for data portability. To increase the generality and flexibility of the hardware based VLSI processor for cryptography, there are scalable design architecture presented [97-99]. The scalability of the VLSI cryptography process provides the programmability of the circuit as using by a micro-processor, or micro-controller.

The CPRM was developed for the digital content protection mechanism between the storage media and the playing devices as well as recording devices [26-27]. The content protection for digital data in a storage media or storage device can be briefly divided into 2 parts: the cryptography for data encryption and decryption, and the key authentication mechanism and management. The CPRM is managed by a organization called 4C entity, LLC (http://www.4centity.com/). The 4C stands for 4 companies. Those are: IBM, Intel, MEI (Matsushita Electronics Inc.) and Toshiba. The CPRM uses the C2 (**C**ryptomeria **C**ipher) Block cipher as the cryptographer for data encryption and decryption [28]; uses RSA algorithm for AKE (**A**uthentication **K**ey **E**xchange) and key management [26]. Through the AKE process, the device and the media can get the same session key. The data storage and transportation between the device and the media in this AKE session can be encrypted and decrypted by the C2 block cipher via this session key.

Originally, CPRM was developed as a content protection mechanism for DVD (**D**igital **V**ersatile **D**isc). After the successful of CPRM for DVD, a small form-factor flash memory card, SD (**S**ecure **D**igital) card, adopted it as the standard for the digital content protection on the flash memory card. That standard is called SD-CPRM which is defined in the SD card security specification [24]. The CPRM standard now is becoming the most popular content protection mechanism in flash memory cards. It is a mandatory function support in the SD card. It is important to discuss the implementation of CPRM in a flash memory controller because of not only its popularity, but also the similar implementation methodology of other advanced security algorithms in a flash memory controller. For example, the advanced data security or content protection can be implemented by the ECC (**E**lliptic **C**urve **C**oding) as the public key exchange and management algorithm, and the AES (**A**dvanced **E**ncryption **S**tandard) for the advanced data encryption and decryption engine.

6.2 SD-CPRM Operation Brief

The C2 block cipher is adopted for the data encryption and decryption engine in SD-CPRM. The C2 block cipher is a Feistel network-based block cipher designed. Table 6-I shows the C2 cipher basic characteristics.

Table 6-I. Basic characteristics of C2 block cipher.

Input Block Size	64 bits
Output Block Size	64 bits
Input Key Size	56 bits
Number of Rounds	10

There are functions defined in the C2 block cipher specification [28], and we implemented these functions into hardware circuit.

- C2_E: Encryption in Electronic Codebook (ECB) Mode, as defined in ISO 8372 or ISO/IEC 10116.

  y = C2_E(k, d);

  where k is a 56-bit key, d is 64-bit data value to be encrypted, and C2_E returns the 64-bit result y.

  Decryption using the C2 cipher in ECB mode is represented by the function

- C2_D: Decryption in ECB Mode.

  y = C2_D(k, d);

where k is a 56-bit key, d is 64-bit data value to be decrypted, and C2_E returns the 64-bit result y.

- C2_ECBC: Encryption in Converted Cipher Block Chaining (C-CBC) Mode. The Cipher Block Chaining mode is chaining the encryption / decryption frames by propagating the result of the frame to the initialization of the encryption / decryption of the frame. It can provide a stronger protection of encrypted data.

  C2_ECBC(k, d)

  where k is a 56-bit key, d is a frame of data to be encrypted, and C2_ECBC returns the encrypted frame.

- C2_DCBC: Decryption in C-CBC Mode.

  C2_DCBC(k, d)

  where k is a 56-bit key, d is a frame of data to be encrypted, and C2_ECBC returns the encrypted frame.

- C2_G: CPPM uses a cryptographic one-way function based on the C2 encryption algorithm. This function is called the C2 One-way Function, and is represented by.

  y = C2_G(d1,d2) = C2_E(d1, d2) + d2;

  where d1 is a 56-bit key, d2 is 64-bit data value to be encrypted, and C2_G returns the 64-bit result y.

The functional block diagram of the SD-CPRM mechanism was shown as in Figure 6-1. There are 4 major processes in the SD-CPRM content protection mechanism:

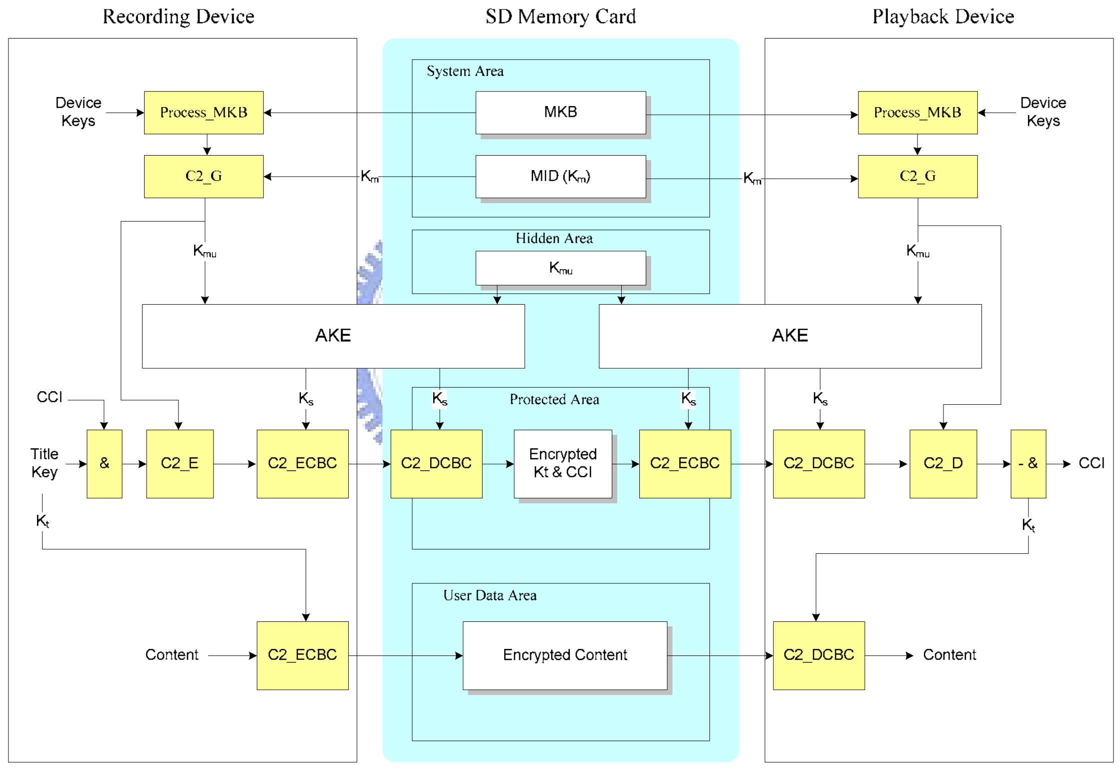
- The Process MKB (**M**edia **K**ey **B**lock): The content recoding device and the playback device get the Media Key (K_m) by processing the MKB read with the Device Keys (K_{d1}~K_{d16}) from the SD memory card. Then, the Media Unit Key (K_{mu}) can be obtained by doing the C2_G function of the Media ID (MID) and Media Key.

- AKE Process: the AKE process must be done in every secure transfer session. The session key is obtained after the successful AKE verification. The AKE process between the recording / playback device and the SD memory card is processed by Challenge 1 => Challenge 2 => Response 2 => Response 1. The functional blocks of the AKE process was shown as in Figure 6-2.

- Secure data transportation for encrypted Title Key (Kt) and CCI (**C**opy **C**ontrol **I**nformation): The Title Keys are used for the key in encryption and decryption of the content data. The CCI is used for Copy Control Information for the management of copyright. Two layers of encryption is designed for the data transportation between the recording / playback device and the SD memory card.

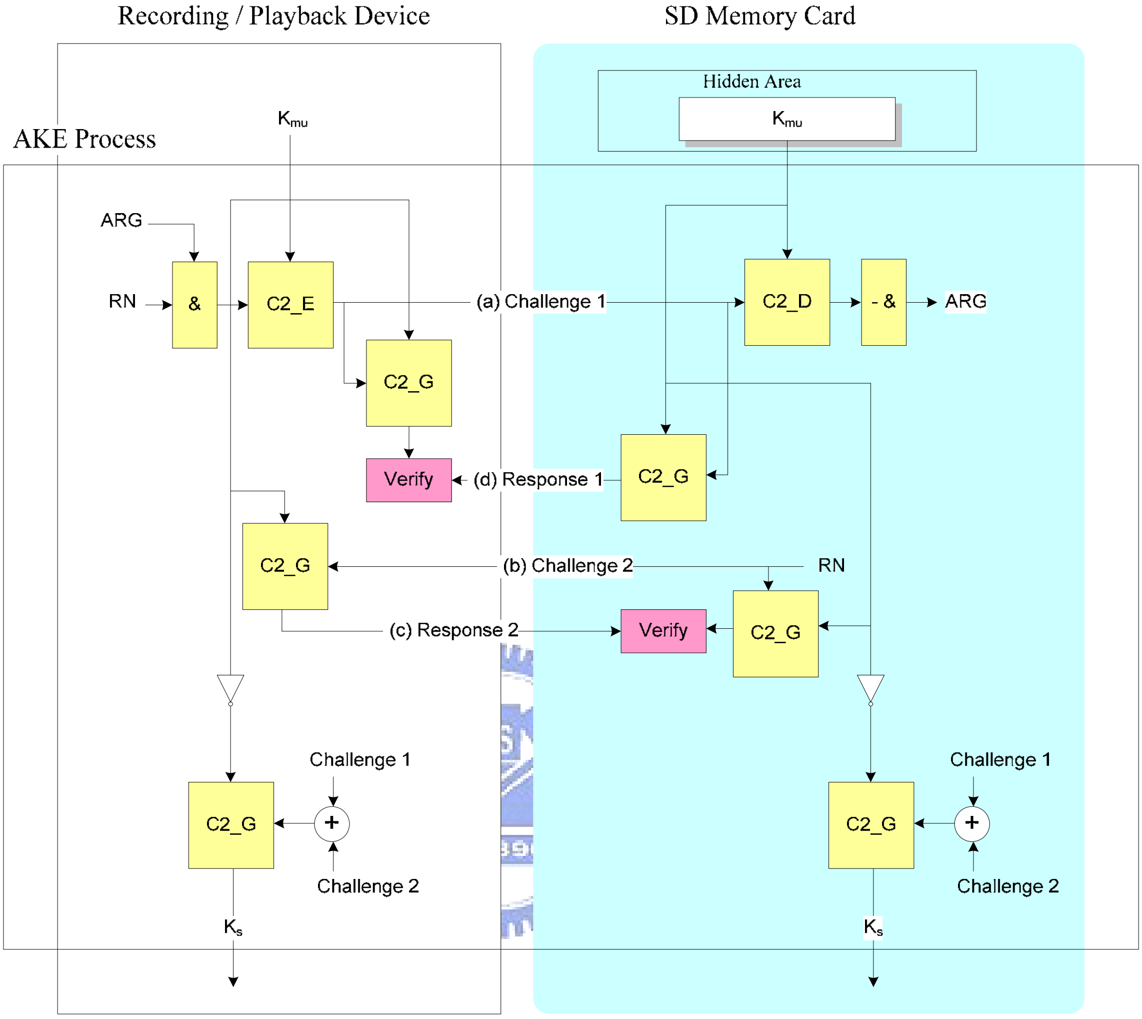
- Encrypted content transportation: The encrypted content data is stored in the user data area of SD memory card. The recording device encrypted the content data by C2_ECBC function at the device side, while the playback device decrypted the encrypted content data at the device side. The data transportation for SD memory card is same as the normal user data, since there is no security cipher needed for the encrypted content.

As shown in Figure 6-1 and 6-2, the basic hardware circuits for SD-CPRM are: C2_E, C2_D, C2_G, C2_ECBC, C2_DCBC. To cope with the control flow of CPRM, the SD card security specification defined the commands and states [24]. Through these hardware accelerators for C2 block cipher, The firmware can then handle the SD-CPRM protocols defined in the SD card security specification in easier programming and shorter code length.



Source: CPRM Specification [26].

Figure 6-1. The functional block diagram of SD-CPRM mechanism.



Source: CPRM Specification [26].

Figure 6-2. The functional block diagram of AKE process.

6.3 Architecture

The hardware accelerator-based module for implementation of SD-CPRM in the flash controller for SD memory card is shown as in Figure 6-3. The functional block marked as “SD-CPRM” is the sub-module circuit for the hardware accelerators in SD-CPRM operation. The micro-controller unit (MCU) controls the SD-CPRM module circuit, during the AKE process, secure reading, and secure writing. Through receiving the instructions from the MCU, the SD-CPRM sub-module will activate the corresponding actions to fulfill the C2 block cipher operations of the data encryption or decryption for the data in buffer RAM, as shown in Figure 6-4. In non-secure data transportation between the SD card Interface controller and the flash memory, the SD-CPRM sub-module will be completely shut off. Thus, it does not affect the normal data reading and writing of the SD card, both in performance and power consumption.

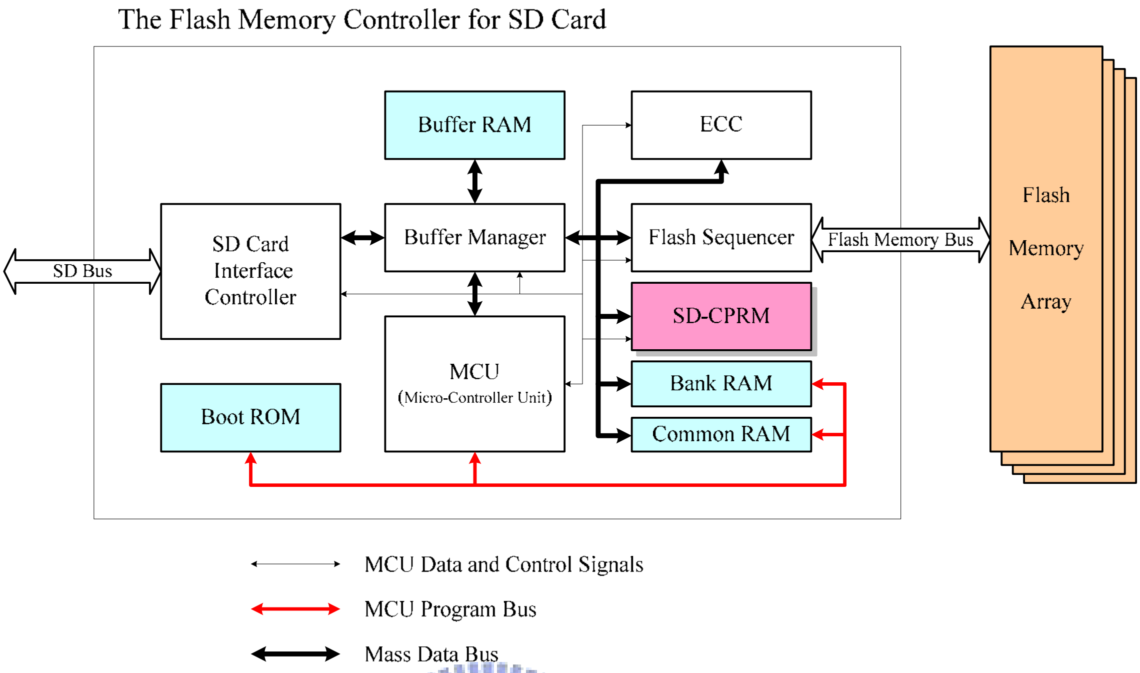


Figure 6-3. Functional block diagram of the CPRM hardware accelerator.

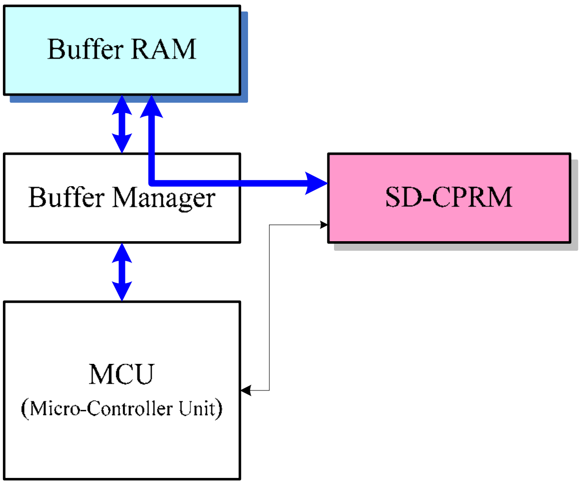
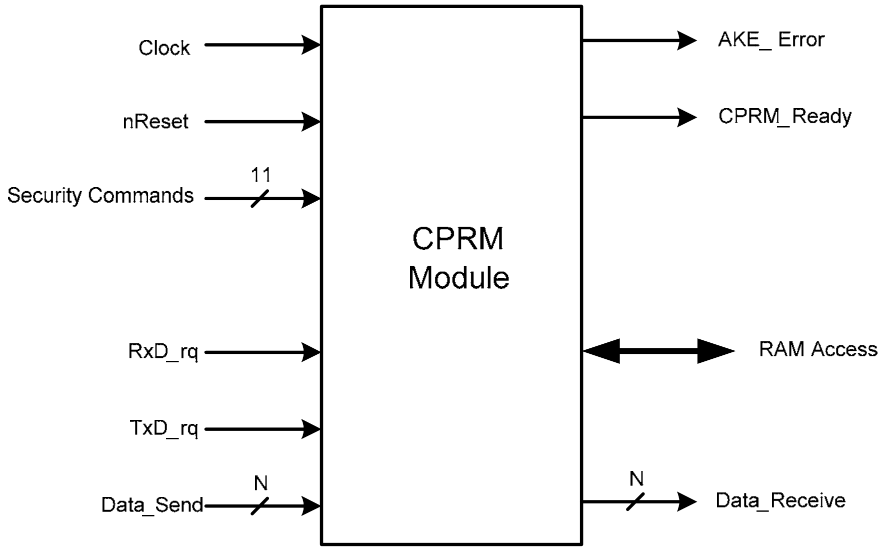


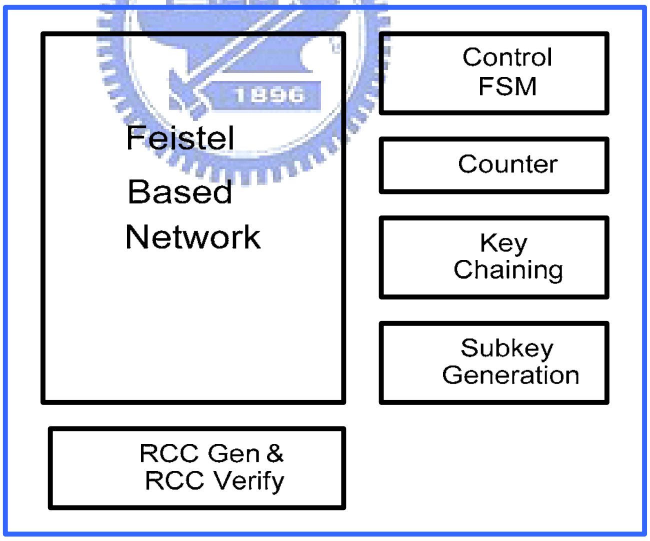
Figure 6-4. The operations between SD-CPRM, buffer, and MCU.

The functional block diagram of the SD-CPRM hardware accelerators sub-module circuit is shown as in Figure 6-5. In Figure 6-5(a), it shows the top entity and the interface signals of the SD-CPRM sub-module circuit. The interface of the SD-CPRM sub-module can be categorized with 2: the MCU related signals for control command receiving and statuses report; the buffer RAM related signals for sub-keys calculation and data encryption / decryption. The secret constant and the seed for random number generation were initialized from loading the data stored in the flash memory. The secret constant is never updated when the SD-Card been manufactured and shipped. While the seed stored in the flash memory for random number generation will be updated for every round of random number generation. The continuous seed update guarantees the truly pseudo random number generation.

The core function in the SD-CPRM sub-module is the Feistel based network, which is doing the main operation in calculating the C2 block cipher functions. The key chaining block is a cooperated circuit to do the CBC mode encryption and decryption. The Subkey generation block is designed to generate the sub-keys. The RCC (**R**edundant **C**heck **C**ode) Gen & Verify is used while the updating MKB process, which is used for the CPRM renewable support as defined in the SD card security specification.



(a) The top entity of SD-CPRM sub-module.



(b) The functional blocks in the SD-CPRM sub-module circuit.

Figure 6-5. The functional block diagram of the SD-CPRM sub-module circuit.

6.4 Verification and Validation

The verification of the designed hardware accelerator modules for C2 block cipher calculation can be verified by a golden test pattern, which can be attained from 4C Entity, LLC. In addition, the consistence between C2_E and C2D was verified by the test pattern for ECB mode verification, shown as in Figure 6-6. The consistence between C2_ECBC and C2_DCBC was verified by the test pattern for CBC mode verification, shown as in Figure 6-7.

The validation of the designed SD-CPRM function was done by through the several recording devices and playback devices, as shown in Figure 6-8. The on market Panasonic and Toshiba SD card readers and MP3 players were used for the validation test.

```
**** Test Data for ECB_E / ECB_D ****

k : Encryption Key (56 bits)

P : Plaintext (64 bits)

C : Ciphertext (64 bits)

sk[0],…,sk[9]: Sub Key


k =   5e916a ef341fa3

P = 89067f2b e2a60d6f  (Plaintext);C = 8fe65fe4 f7ba8005  (Ciphertext )

sk[0] = ef342cb3, sk[1] = 3f46c172, sk[2] = 7a45b43c, sk[3] = 5779a10d, sk[4] = 41fa3c69

sk[5] = 6bd2340d, sk[6] = 5abbd767, sk[7] = 9a0fd55f, sk[8] = a35e94fa, sk[9] = 22d5eb08


sbox[256] = {

  0xB6,0xAA,0xEB,0xB3,0x35,0x5D,0xEE,0xB1,0x72,0x33,0x05,0x13,0x6D,0xC7,0x6C,0x27,

  0x25,0x54,0xE9,0x4C,0xDE,0xC3,0x21,0x39,0xA9,0xAB,0xD6,0xDF,0xE8,0x71,0x94,0xAE,

 0x16,0x44,0x76,0xCD,0xB7,0x78,0x20,0xF0,0xC1,0x9F,0xCF,0xAF,0x0F,0xCB,0x59,0x83,

 0x3A,0x5E,0xB8,0xB5,0xF3,0x47,0x80,0xC2,0xF6,0x14,0xE6,0x69,0xFC,0x17,0xE0,0xE5,

 0x79,0xF9,0x12,0xBF,0x3C,0xB4,0x66,0xAD,0xF7,0x65,0x95,0xF4,0x4E,0x02,0xA0,0x07,

 0x4D,0x2F,0x0D,0x7E,0xE4,0xEF,0xA1,0x8C,0x6E,0xD2,0xFD,0x19,0x1C,0x82,0x42,0xBB,

 0x9A,0x43,0xC6,0xE2,0x1F,0xF2,0x75,0x1A,0x63,0x45,0xD1,0x30,0x81,0x7F,0x8E,0x62,

 0x3B,0xA4,0xFB,0x1E,0x5F,0xBC,0xB0,0x40,0x8B,0x74,0x38,0x8A,0xC4,0x73,0x9C,0x09,

 0xD4,0xED,0xD3,0x5A,0x60,0x48,0xC5,0x9E,0x01,0xCC,0x34,0x1B,0x58,0x36,0x23,0x88,

 0x7A,0x90,0x9B,0x8F,0xBD,0x3F,0xB9,0x57,0xA2,0x3E,0x04,0xB2,0x49,0x37,0x5C,0x7D,

 0x61,0x4A,0xA6,0x67,0xEC,0x7C,0x0E,0x96,0xDD,0xE3,0x2C,0x56,0x08,0x0C,0x8D,0x2B,

 0x6A,0xFE,0xEA,0xA3,0xCA,0x3D,0x91,0xE7,0xC9,0xAC,0x03,0xD5,0x89,0x86,0xDC,0x10,

 0x55,0x77,0xC8,0xD7,0x97,0x24,0x46,0x9D,0x0A,0x1D,0x22,0xD9,0xFF,0x5B,0x52,0xD8,

 0x00,0xFA,0x53,0x26,0x29,0x2E,0x2A,0x11,0xC0,0x6F,0x4F,0x7B,0x28,0x99,0x41,0x92,

 0xDB,0xF8,0x50,0xA8,0x51,0xA5,0x4B,0x93,0x87,0xDA,0x06,0x85,0x2D,0xBA,0x0B,0x98,

 0x70,0x6B,0xBE,0xF1,0x18,0xD0,0x31,0x68,0x15,0x84,0x64,0xE1,0xCE,0xA7,0xF5,0x32 };
```

Figure 6-6. The test pattern for ECB circuit verifications.

```
   **** Test Data for C2_CBC(Encryption) / C2_CBC(Decryption) ****
k : Encryption Key (56 bits)
P : Plaintext (24 bytes)
    P0 is the first block to be encrypted,
    the second is P1, and P2 is the last.
C : Cipher text (24 bytes)

k  =   7cb3c4 db094713
P0 = a24632d8  24320844  -->  C0 =  50fc09d1  691c5102
P1 = 7d8111df  8ce24172  -->  C1 =  541d322f  68e7fd79
P2 = 76be42d7  0db14418  -->  C2 =  91a80c3d  9d9f310d
sbox[256] = {
  0xB6,0xAA,0xEB,0xB3,0x35,0x5D,0xEE,0xB1,0x72,0x33,0x05,0x13,0x6D,0xC7,0x6C,0x27,
  0x25,0x54,0xE9,0x4C,0xDE,0xC3,0x21,0x39,0xA9,0xAB,0xD6,0xDF,0xE8,0x71,0x94,0xAE,
 0x16,0x44,0x76,0xCD,0xB7,0x78,0x20,0xF0,0xC1,0x9F,0xCF,0xAF,0x0F,0xCB,0x59,0x83,
 0x3A,0x5E,0xB8,0xB5,0xF3,0x47,0x80,0xC2,0xF6,0x14,0xE6,0x69,0xFC,0x17,0xE0,0xE5,
 0x79,0xF9,0x12,0xBF,0x3C,0xB4,0x66,0xAD,0xF7,0x65,0x95,0xF4,0x4E,0x02,0xA0,0x07,
 0x4D,0x2F,0x0D,0x7E,0xE4,0xEF,0xA1,0x8C,0x6E,0xD2,0xFD,0x19,0x1C,0x82,0x42,0xBB,
 0x9A,0x43,0xC6,0xE2,0x1F,0xF2,0x75,0x1A,0x63,0x45,0xD1,0x30,0x81,0x7F,0x8E,0x62,
 0x3B,0xA4,0xFB,0x1E,0x5F,0xBC,0xB0,0x40,0x8B,0x74,0x38,0x8A,0xC4,0x73,0x9C,0x09,
 0xD4,0xED,0xD3,0x5A,0x60,0x48,0xC5,0x9E,0x01,0xCC,0x34,0x1B,0x58,0x36,0x23,0x88,
 0x7A,0x90,0x9B,0x8F,0xBD,0x3F,0xB9,0x57,0xA2,0x3E,0x04,0xB2,0x49,0x37,0x5C,0x7D,
 0x61,0x4A,0xA6,0x67,0xEC,0x7C,0x0E,0x96,0xDD,0xE3,0x2C,0x56,0x08,0x0C,0x8D,0x2B,
 0x6A,0xFE,0xEA,0xA3,0xCA,0x3D,0x91,0xE7,0xC9,0xAC,0x03,0xD5,0x89,0x86,0xDC,0x10,
 0x55,0x77,0xC8,0xD7,0x97,0x24,0x46,0x9D,0x0A,0x1D,0x22,0xD9,0xFF,0x5B,0x52,0xD8,
 0x00,0xFA,0x53,0x26,0x29,0x2E,0x2A,0x11,0xC0,0x6F,0x4F,0x7B,0x28,0x99,0x41,0x92,
 0xDB,0xF8,0x50,0xA8,0x51,0xA5,0x4B,0x93,0x87,0xDA,0x06,0x85,0x2D,0xBA,0x0B,0x98,
 0x70,0x6B,0xBE,0xF1,0x18,0xD0,0x31,0x68,0x15,0x84,0x64,0xE1,0xCE,0xA7,0xF5,0x32 };
```

Figure 6-7. The test pattern for CBC circuit verifications.

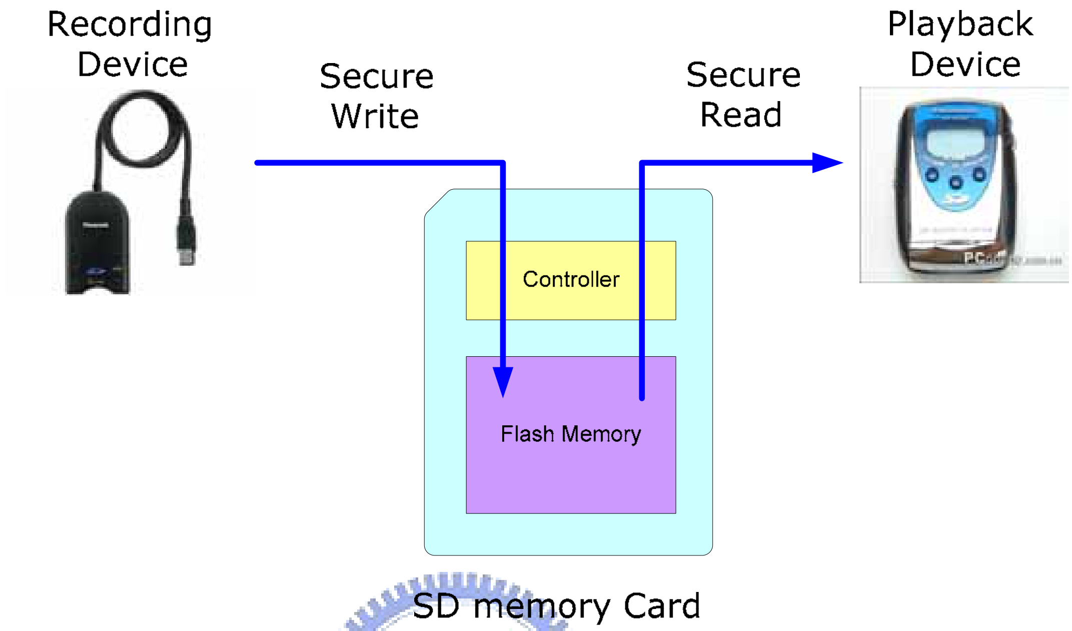


Figure 6-8. The illustration of SD-CPRM function validation.

6.5 Summary

A SD-CPRM function with hardware accelerators was implemented, verified, and validated. The circuit size is around 8K gate count through UMC 0.18um via Faraday Technology (http://www.faraday-tech.com) cell library, which is relative very small in the SD memory card controller. But it saved lots of tedious firmware programming job to support the C2 block cipher functions in the SD Card. Moreover, it increased the performance of the security reading and writing of SD Card. The micro-graph of a SD memory card controller chip by using the presented hardware accelerator based architecture for the SD-CPRM function is shown as in Figure 6-9. The performance of the implemented circuit was shown as in Table 6-II.

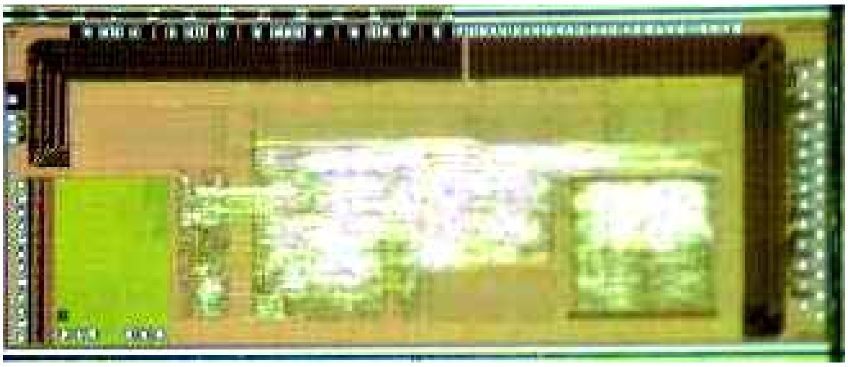


Figure 6-9. The micro-graph of a SD memory card controller chip by using the presented hardware accelerator based architecture for the SD-CPRM function.

Table 6-II. Performance of the hardware accelerator-based CPRM circuit.

Process Technology	UMC 0.18um
Gate Counts	~8K
Operating frequency	60 MHz

The concept of hardware accelerators for supporting the security function in the SD-Card can be extended to the development of the specific cryptograph processor which can be configured, and supports the programmability by firmware or software.

The specific cryptograph processor can not only be used in the major standard encryption method, like AES, DES, 3DES (Triple DES), etc., but also to be configurable for proprietary data security schemes. The proprietary data encryption / decryption schemes might not be convenient for data transportation, but it is probably more secure for data protection.

Chapter 7 Multi-Type Solid-State Memory Supporting

In this chapter, the multiple types of the solid-state memory characteristics were discussed, and controller architecture for supporting the multiple types of the solid-state memory was presented. In section 7.1, a brief introduction on the background and related works were given. In section 7.2, we present the handling methods for the hybrid non-volatile solid-state memory array. In section 7.3, we propose the novel firmware code banking and ISP (In-System Programmability) for universal flash memory supporting, and on-line maintenance and upgrade was discussed. In section 7.4, we proposed a control circuit architecture for the hybrid solid-state drives. Finally, section 7.5 is the summary.

7.1 Introduction

There are several kinds of solid-state memory used in the computer, communication, industrial, car electronics, and consumer electronics besides the NAND flash memory. They can be divided as the volatile memory and non-volatile memory. The volatile memory are using for data storage as the device is powered, while the non-volatile memory can keep the data even if the power is off. The major volatile memory types are the DRAM and SRAM. Besides the NAND flash memory as the storage media for the mass data storage devices, there are kinds of non-volatile solid-state memory device which has different characteristics of the memory cell. Such as FeRAM (Ferroelectric RAM, or called FRAM), which can offers a number of advantages, notably lower power usage, faster write speed and a much greater maximum number (exceeding 10^{16} for 3.3 V devices) of write-erase cycles [100], MRAM (Magneto-resistive Random Access Memory) which are using the magnetic element for storing bits, thus can have the almost un-limited endurance cycles and write the data without do the erase in advance [101], and PC-RAM (Phase-Change RAM, or called PRAM) with its fast performance and high endurance for write cycles and data retention life span [102]. The different characteristics among these non-volatile solid-state memory devices can be combined together and formed a hybrid solid-state storage system, which can provide an optimization of cost, performance, and reliability in the system level point of view.

In this chapter, we discussed the advanced features for solid-state drive, which can be made by multiple types of non-volatile solid-state memory devices. The solid-state drive composed of multiple types of non-volatile solid-state memory devices can be called as a hybrid solid-state drive, since the memory type might be NAND flash (both SLC, MLC), FeRAM, MRAM, PRAM, etc. To support the multiple types of non-volatile solid-state memory, the controller needs some added functional features, besides the functional units supporting for NAND flash memory.

7.2 Handling of Hybrid Non-Volatile Memory Array

In this section, we discussed how to handling the multi-type solid-state memory in the controller. In Table 7-I, the features comparison of NAND-SLC, NAND-MLC, FeRAM, MRAM, and PRAM were listed. The NAND flash memory has the lowest cost and largest capacity advantage, but it has some weaknesses, such as: slow speed in data writing, limited endurance cycles, and limited data retention period for reliability. The FeRAM, or MRAM, is more expensive and lower density, but the excellent write performance and much higher endurance cycles may help the NAND flash memory on the small block data write caching, and it can be as the storage area for the host frequent accessed data (e.g., the FAT and directory of DOS file system).

Table 7-I. Feature comparisons among non-volatile solid-state memory.

Types	NAND SLC	NAND MLC	FeRAM	MRAM	PRAM
Max. Density (in 1Q'2007)	8 Gb	16 Gb	4 Mb	16 Mb	512 Mb
I/F	NAND	NAND	SRAM	SRAM	N/A
Read cycle time	25 ns	30 ns	35 ns	35 ns	N/A
Write cycle time	30 ns	30 ns	35 ns	35 ns	N/A
Program time	200 us	800 us	No need	No need	No need
Pre-Erase need	Yes	Yes	No	No	No
Typical Endurance	100K	10K	10^{14}	> 10^{16}	10^{8}
Data Retention	10 yr.	3 yr.	> 10 yr.	20 yr.	> 10 yr.
US$/MB (y2006)	0.01	0.007	~20	~30	N/A

The higher performance of FeRAM and MRAM non-volatile memory can be used as a non-volatile cache memory as host writing data to the NAND flash memory array. The application of non-volatile solid state memory cache in rotating-type disk storage device has been introduced in the 1990s [105-106]. The faster response of the non-volatile solid-state memory can enhance the disk writing performance of the Hard Disk Drive [107]. Moreover, the quicker read data availability of the non-volatile solid-state memory can help the operating systems top boot up in shorter time than Hard Disk Drive, called as "Instant-On" [108]. The advantages of the non-volatile solid-state memory are not only providing the quicker response in both data reading and data writing, but also it can keep the effective data if an accidental power failure happened.

The accessing interface for the current FeRAM and MRAM non-volatile memory is not a NAND type interface. The accessing interface is a JEDEC SRAM (**S**tatic **R**andom **A**ccess **M**emory) type interface instead. Like most of SRAM memories, they provide the page access mode for faster continuous sequential data reading and writing besides the randomly word-wise access mode. The page mode access is more suitable for the data accessing in a mass data storage device. So, we need modify the flash access controller circuit in the flash sequencer, refer to Section 5.6. The flash access controller circuit needs to support both the NAND-type and SRAM-type accessing interface. The circuit to support both accessing types will be slightly larger than to support the NAND-type interface only.

Besides the hardware circuit modification for FeRAM and MRAM support, it needs the algorithmic procedures to handle the data flow control and data allocation management. We proposed the frequently access data in the solid-state drive to be stored in the FeRAM or MRAM memory devices, for example the frequently accessed data by host file systems, the identified user hot data, the tables for the flash memory management, and the cache blocks for write data cache.

The host frequently accessed data can be stored in the FeRAM or MRAM memory, for example, the FAT and directory of DOS file systems. The data area can be identified by examining the format in the solid-state drive. Thus, the solid-state drive controller can allocate this data in the FeRAM or MRAM memory to have the quicker responses, and higher endurances. Moreover, some algorithms for the “Hot” data identification by observing the host data accessing behavior have been developed [85-88]. The hot data identification can not only identify the frequently accessed data of file systems, but also the frequently accessed user’s data, for example, the file been editing by the user.

The tables for NAND Flash operating algorithm were updated more frequently than the other blocks for storing user’s data. Moreover, they are much more important data since it will affect the whole system functionality of the solid-state drive. For example, the L2P tables record the addressing mapping relationships between the logical block and physical block; they are updated once a new mapping relationship was established by data writing. The tables were updated in almost every data writing sequence. If the L2P tables lost, the logical to physical linkage will also be lost. Thus, the stored data cannot be retrieved any more. The defect management tables might be updated very often, but it records the defective blocks and its replacement in the solid-state drive. It is as important as the L2P tables, and it cannot be lost in any conditions. The caching blocks, temporary blocks, and special blocks for keeping system parameters in flash algorithm operation, are all related to the data linking structures of the flash memory blocks. They are used to store the system parameters and need be treated like the L2P tables.

The better performance and higher endurance cycles of FeRAM and MRAM non-volatile solid-state memory than the NAND flash memory let it can be used as a non-volatile memory cache in the solid-state drive. The direct overwrite characteristics of FeRAM or MRAM has the advantage to gain the faster data writing performance by writing the data to the non-volatile memory immediately without doing the flash blocks management in the NAND flash memory array. Furthermore, the higher endurances of FeRAM and MRAM can lower the erase cycles of the NAND flash memory if the host is writing the small data length in the same flash memory block. A typical cache mapping strategy is to have the data mapping in a block size. The cache committing can be executed if the block accessed by Host was changed. In addition, as a “Non-volatile cache memory” for data cache of host writing, it will not have problem to do the write data cache, since the data won’t be lost even an accidental power failure happens.

7.3 Firmware Code-Banking and ISP Architecture

In this section, we presented a hardware structure for the firmware code-banking, which provides the ISP (**I**n-**S**ystem **P**rogrammability) capability. By using the firmware code-banking and ISP architecture, the solid-state drive can support the conceptual universal flash memory support and the on-line maintenance and upgrade.

7.3.1 Firmware Code-Banking Architecture

The architecture for the code-banking was shown in Figure 7-1. The micro-controller unit used in the designed NAND flash controller was the 1-T (one instruction cycle per system clock) RISC (**R**educed **I**nstruction **S**et **C**omputer) type, C51 instruction compatible controller. The boot ROM was the masked ROM (Read Only Memory) which stores the boot codes for the Micro-controller in booting. In the booting process, the Micro-controller was being initialized, the common code and bank code #0, stored in the specified area of the flash memory, were then loaded through the code loader to the common RAM and bank RAM respectively. After the Micro-controller booted, the system program will be switched to run the codes in common RAM and bank RAM to handle the commands passed from host. By the code-banking architecture, the whole system firmware can be separated by several banks: Bank code #0, Bank code #1, …, Bank code #n. The banked codes would be executed bank by bank when loaded by the code loader to the bank RAM. Since the whole system firmware was divided into the common code and the bank codes stored in the NAND flash memory of the SD/MMC memory card, the system firmware can be upgraded by the vendor commands and protocols from the host side. Therefore, the designed code-banking architecture supported the In System Programmability (ISP) functionality.

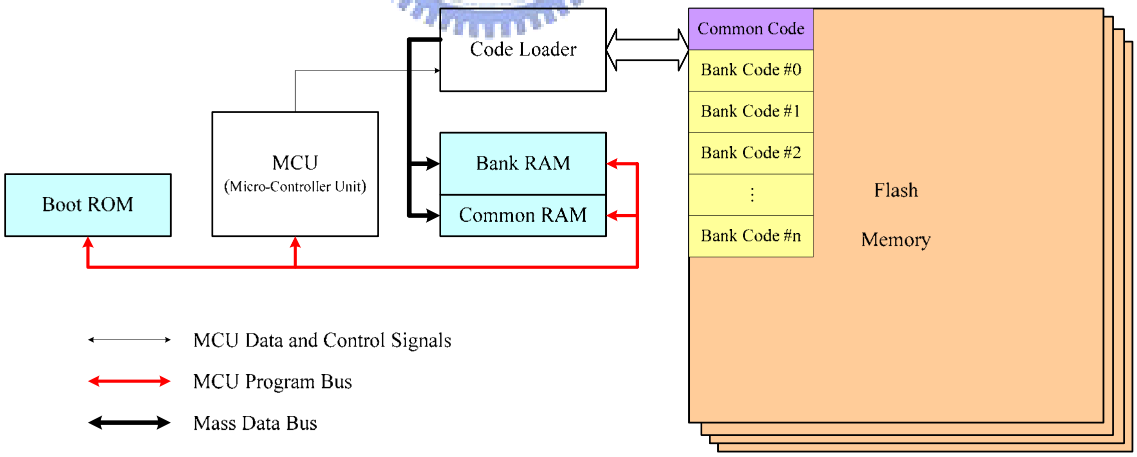


Figure 7-1. The code-banking architecture

7.3.2 Concept for Universal Flash Memory Support

To support the various kinds of the NAND Flash memory devices in a controller, a specified flash parameters table was created to record some system operation parameters of the NAND flash memory. A typical table for the flash memory parameters was shown as in Figure 7-2. The table was started by a Start-Tag, and ended by an End-of-Table Flag. The specified flash parameters table was composed of some useful parameters such as: Total Capacity, Total Physical Blocks, Pages per Block, …, etc. With the code-banking / ISP function and the specified flash memory parameters, the various kinds of NAND flash memory devices can be supported in the same controller chip. As shown in Figure 7-3, the specified flash memory parameters were recorded in the production phase, since the kind of the NAND flash memory was known in the production stage, or the kind of the NAND flash memory could be identified before mounted to the SD/MMC memory card. After the production phase, the specified flash memory parameters and the corresponding system firmware were built in the flash memory in the SD/MMC memory card. The controller could then be booted, loaded the appropriate system banking codes, and read back the specified flash memory parameters, thus, the controller can access the NAND flash memory in a proper way, and perform well.

Start of Table Tag	Total Capacity	Total Blocks	Pages Per Block	Page Size (Bytes)	Read Cycle Time	Write Cycle Time	Address Cycles	Special Commands	……	End of Table Flag

Figure 7-2. A typical table for the flash memory parameters

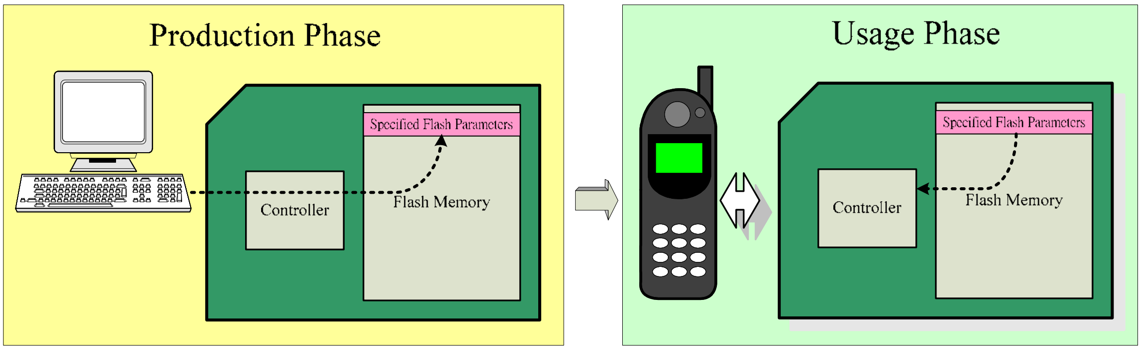


Figure 7-3. The illustrative diagram for various kinds of flash memories

7.3.3 On-line Maintenance and Upgrade

The maintenance of the solid-state drive can be supported by either implementing the S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology) feature which defined in the ATA standard [13], or defining the proprietary vendor maintenance commands. This maintenance and upgrade scheme can be done on stand alone computer, or through the internet for on-line checking and upgrade. The solid-state drives maker can have the most updated information on the web-site for the user’s on-line real-time check.

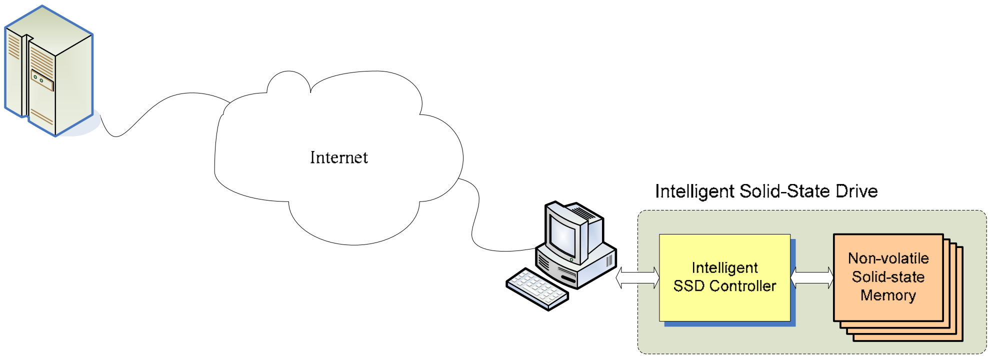

Internet
Intelligent Solid-State Drive
Intelligent SSD Controller
Non-volatile Solid-state Memory

Figure 7-4. On-line maintenance and upgrade through internet.

7.4 Controller Architecture for Hybrid Multi-Channel Solid-State Memory Array

The controller architecture to support the Multi-channel hybrid solid-state memory array was presented in this section. Through the discussion in chapter 3, 4, 5, and the above of this chapter, we conclude a controller architecture for solid-state drive with multi-channel, hybrid, non-volatile solid-state memory array, as shown in Figure 7-5.

The flash sequencer circuit is modified as Solid-State Memory (SSM) sequencer, which supports two kinds of memory access interface: the NAND type interface as well as the SRAM type. The NAND type interface is configured for the versatile NAND flash memory. The SRAM type interface is configured for supporting of FeRAM and MRAM. The multi-mode BCH ECC circuit is adopted to support the different ECC capability requirement of different non-volatile solid-state memory. An off-chip DRAM buffer is connected by the buffer manager, which provides the more buffer space for MCU in system-level operation.

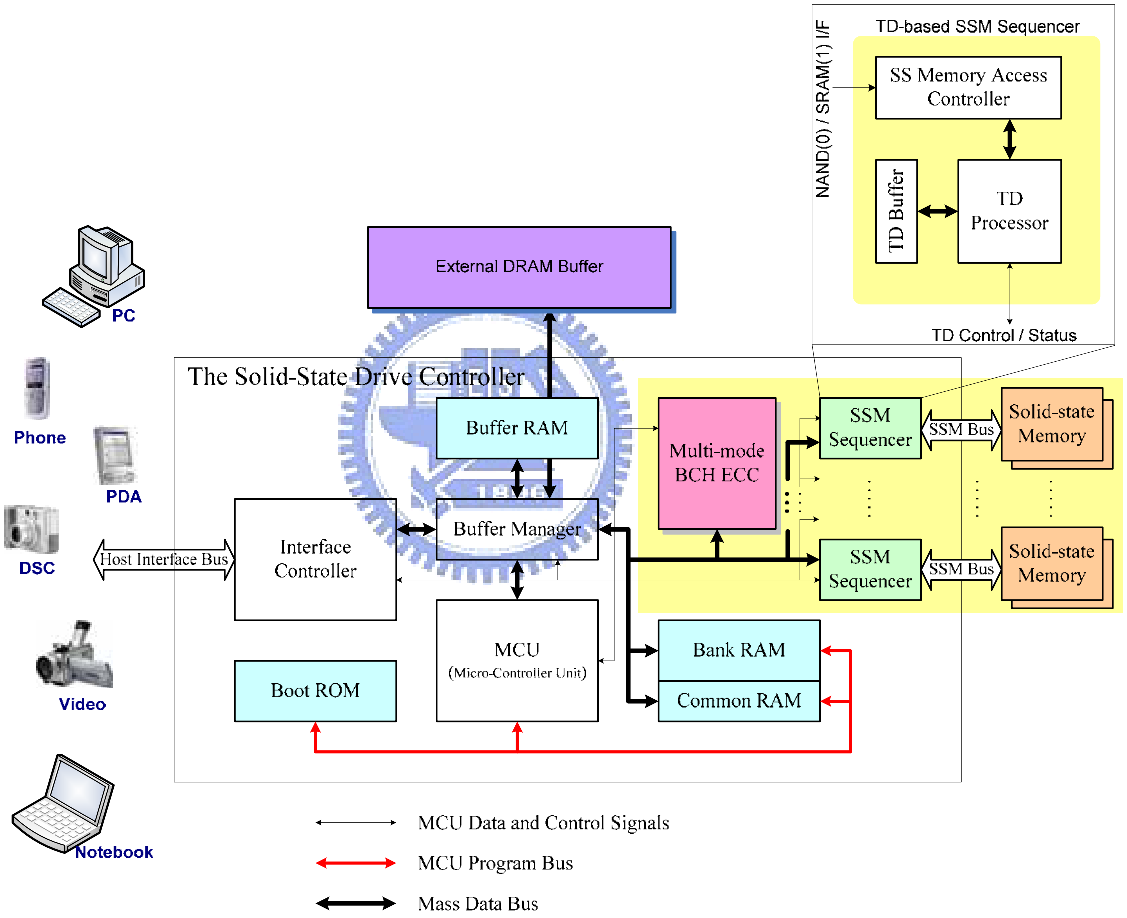


Figure 7-5. The functional block diagram of the controller support for hybrid multi-channel non-volatile solid-state memory array.

7.5 Summary

The hybrid multi-channel non-volatile solid-state memory array can be constructed to optimize the cost-performance of a solid-state drive, which leverages the bit density, cost per MB, endurance cycles and programming speed of various non-volatile solid-state memory types. The good performance types of the non-volatile memory devices can be used as the “Non-volatile cache memory” and the “Non-volatile Buffer RAM” to compensate the slow speed and poor endurance of the NAND flash memory, even though it’s small density and expensive. The FeRAM, PRAM is going to be on market in 2008, and MRAM could be on 2009, thus, we can foresee the good performance of the hybrid solid-state drive will provide the high performance, high reliable, reasonable cost, low lower, low heat, shock free, and no acoustics mass data storage device for 3C (**C**omputer, **C**ommunication, **C**onsumer electronics).

Chapter 8 Implementation of Flash Controllers

In this chapter, three flash memory controller chips implementation were presented: a NAND flash controller for SD/MMC card, a NAND flash controller for dual mode USB, and architecture design for high-speed SATA II hybrid flash memory controller.

8.1 A NAND Flash Controller for SD/MMC Card

The functional block diagram of the NAND flash controller was shown in Figure 8-1. The major functions of the controller can be divided as: the *t*-EC *w*-bit parallel BCH ECC circuit, the code-banking structure and firmware In-System Programmability (ISP), the defect management and wear-leveling algorithm, and the multi-channel and multi-buffering mechanism. The ECC circuit was designed here to enhance the data integrity and reliability of the data stored in the flash memory. The code-banking structure for the micro-controller complying with firmware ISP function can provide the firmware upgrade to support various kinds of flash memory devices. The defect management algorithm can increase the yield of the flash memory and prolong the product life cycle by replacing the defected (Bad) block with the reserved good/virgin (clean data) blocks. The wear-leveling algorithm was also introduced to prolong the product life cycle by preventing the flash memory blocks from un-balanced usage. The multi-channel and multi-buffering mechanism was designed to increase the data transfer rate at the flash memory side (flash memory bus) and to reach the maximum bandwidth of the host-side interface bus (SD or MMC bus in Figure 8-1).

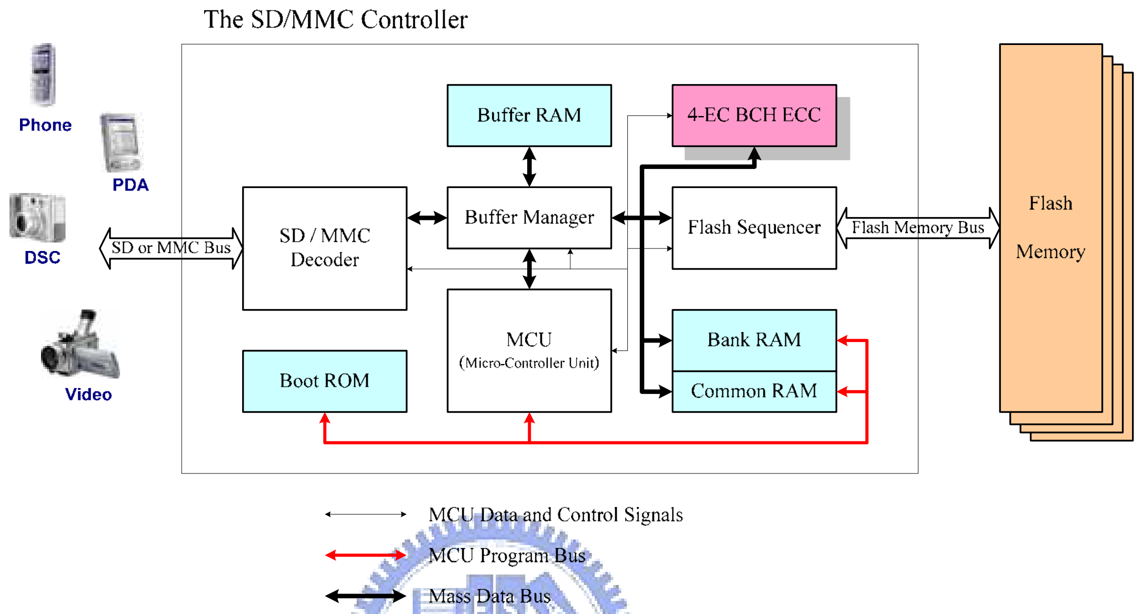


Figure 8-1. The block diagram of the NAND flash controller for SD/MMC card.

With the designed chip architecture shown in Figure 8-1, the circuit of the chip was designed and implemented by the UMC 0.18 um CMOS process. The chip layout of the designed NAND flash controller was shown in Figure 8-2. And the silicon photograph of the controller die was shown in Figure 8-3.

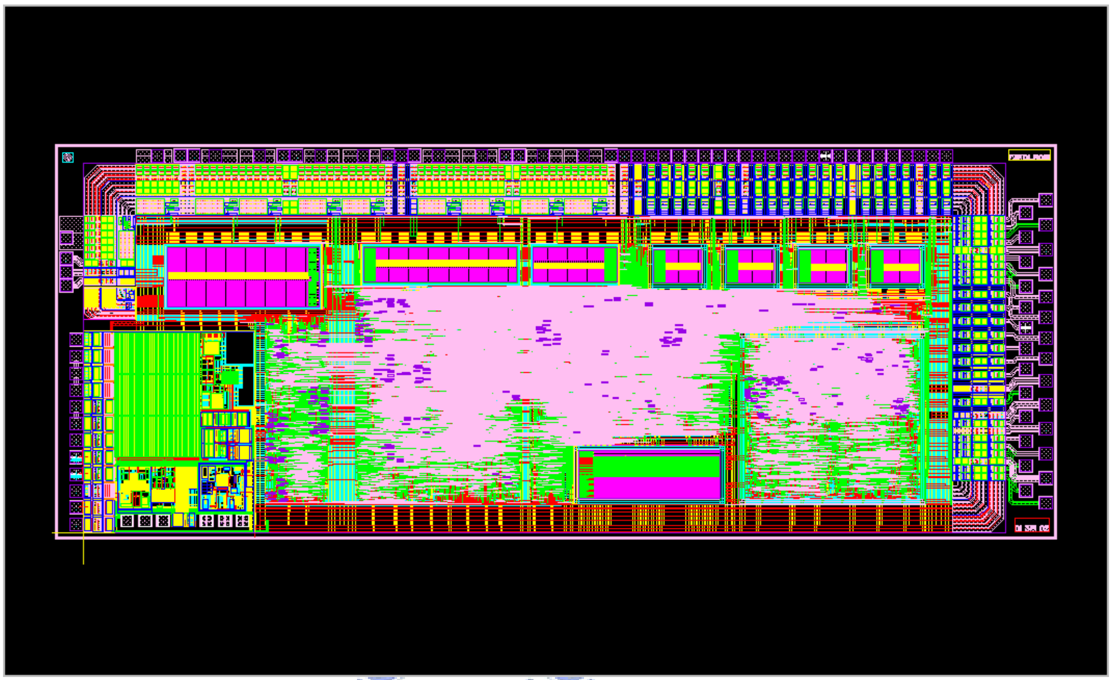


Figure 8-2. The chip layout of the NAND flash controller.

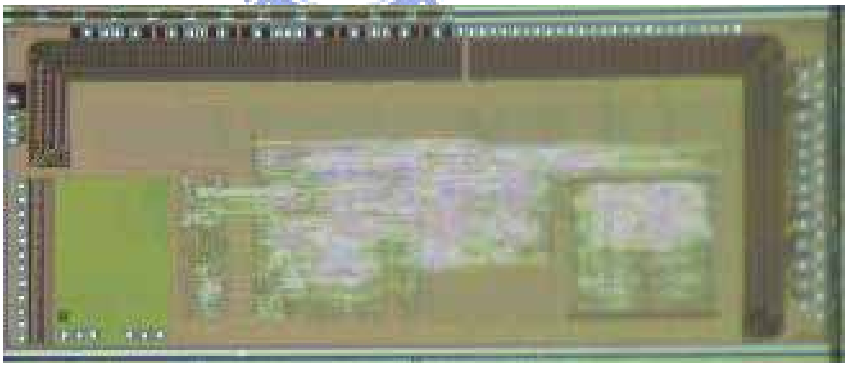


Figure 8-3. The photograph of the silicon die with UMC 0.18um process.

The system performance of the designed NAND flash controller was evaluated and tested by the certified platform of MMCA (http://www.mmca.org) and SDA (http://www.sdcard.org). The test platform was provided by Testmetrix, Inc.

(http://www.testmetrix.com). The Testmetrix machine can test and evaluate the system performance and reliability on the SD or MMC card to be tested. The summary of the system performance test results of the designed NAND flash controller with 2 Samsung K9F1G NAND flash memory devices in dual channel configuration were shown in Table 8-I. By implementing the code-banking / ISP, the various kinds of the NAND flash memory were tested for data read / write process and the supported items are listed in Table 8-II.

Table 8-I. The system performance test results

Test Items	Results*
1. Sequential Multi-Block Read @ MMC 8-bit, 52 MHz	42.33 MB/sec
2. Sequential Multi-Block Write @ MMC 8-bit, 52 MHz	19.86 MB/sec
3. Random Multi-Block Read @ MMC 8-bit, 52 MHz	41.03 MB/sec
4. Random Multi-Block Write @ MMC 8-bit, 52 MHz	12.77 MB/sec
5. Suspend Current @ 3.3V	120 uA
6. Operating Current at Sustained Data Write @ 3.3V	30.95 mA
7. Operating Current at Sustained Data Read @ 3.3V	20.23 mA

*The test conditions:

1. Test Platform: Testmetrix for SD/MMC Card
2. Flash Configuration: 2x Samsung K9F1G08U0M in Dual Channel

Table 8-II. The NAND flash memory support list

Company	Capacity and Model (Part) Numbers
Samsung	See the website at: http://www.samsung.com/Products/semiconductor
Toshiba	See the website at: http://www.toshiba.com/taec/Catalog
Hynix	See the website at: http://www.hynix.com/datasheet/eng/flash
Micron	See the website at: http://www.micron.com/products/nand
Fujitsu	See the website at: http://www.fujitsu.com/global/service/microelectronics/product
ST	See the website at: http://www.st.com/stonline/products/families/memories/fl_nand
Spansion	See the website at: http://www.spansion.com/flash_memory_products

8.2 A NAND Flash Controller for Dual-mode USB Flash Card

A dual-mode USB interface can reduce the operation power consumption among the data accessing between the host device and the USB flash memory card [69]. The operation illustration of the dual-mode USB interface is shown as in Figure 8-4. There are 2 buses in the architecture: the standard USB and the low power USB. The standard

USB bus uses the differential pair signaling which is defined in the USB specification. In the standard USB mode, there needs a USB-PHY (**PHY**sical layer circuit of **USB** 2.0)circuit between the differential signaling and the digital UTMI bus. The low power USB bus can be used in the USB interface applications which are not using the long cable connection, for example, the application to mobile devices. In between the low power USB bus and the UTMI (**U**SB 2.0 **T**ransceiver **M**acro-cell **I**nterface) bus, there is a digital wrapper circuit to bridge these two buses together. The standard USB bus and the low power USB bus was integrated as a common bus in the device side, which was used to connect to the host devices. When the dual-mode USB device was connected to a PC/Internet host, the standard USB bus was adopted. On the other hand, the low power USB bus was used as the host-side mobile devices been connected. The interface mode detector was designed as a function block to detect which the standard USB or low power USB bus been connected. It provided the auto-detection and auto-switch of the interface on the common bus.

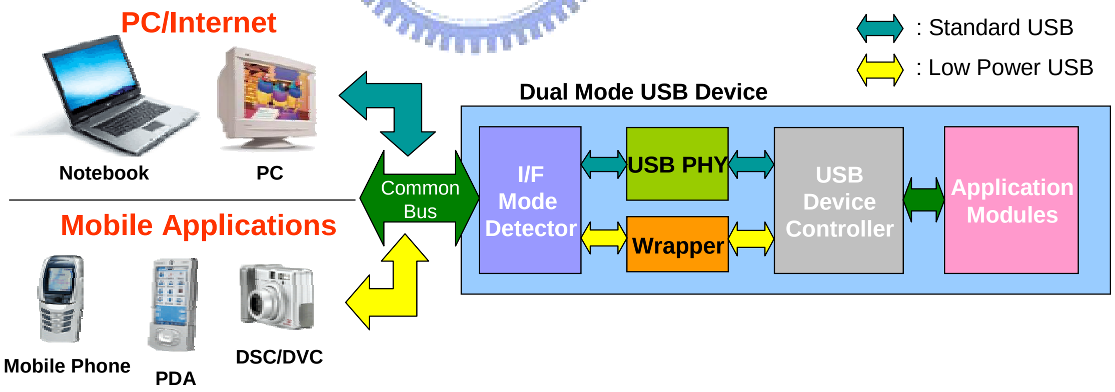


Figure 8-4. The operation illustration diagram of the dual-mode USB.

A controller chip for the dual mode USB flash card was implemented and presented in the section. The block diagram of the implemented controller chip was shown in Figure 8-5. The controller can detect if a standard USB or a low power USB bus attached by a specified hardware signal send from the host device with low power USB interface capability. The bus controller unit was designed to control the USB-PHY or the digital wrapper circuit to be enabled and connect to the host devices. If the dual mode USB flash card is connected to the power sensitive device (battery powered), the low power USB interface can then be selected, and the operation power can be saved. If the dual mode USB flash card is connected to the power insensitive device (AC powered) with standard USB port built-in, the standard USB interface can then be selected, the command communication and the data transfer between the host and the device will be executed by the standard USB interface. The USB device controller was located after the USB-PHY and the digital wrapper circuit. These function units handled the dual mode USB bus protocols, and the interface operations. Besides the dual mode USB interface inside the controller, there were some typical function units designed to accomplish the flash memory data accessing, buffer management, and so on. The micro-controller with boot ROM, common RAM and bank RAM were designed to handle the system coordination and control of the dual mode USB flash card.

The chip layout of the designed dual mode USB flash controller chip was shown in Figure 8-6. By using the UMC 0.18 um CMOS process, a real chip was fabricated and shown the chip die photograph in Figure 8-7.

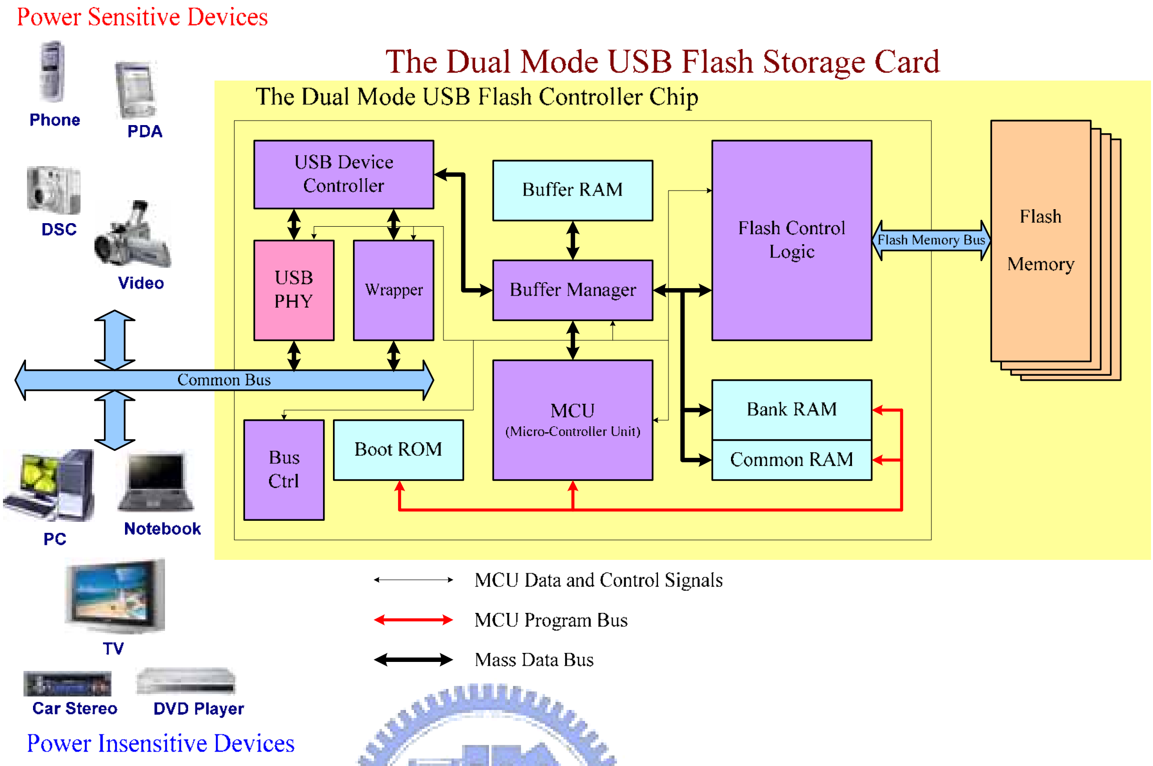


Figure 8-5. The block diagram of the dual-mode USB flash controller

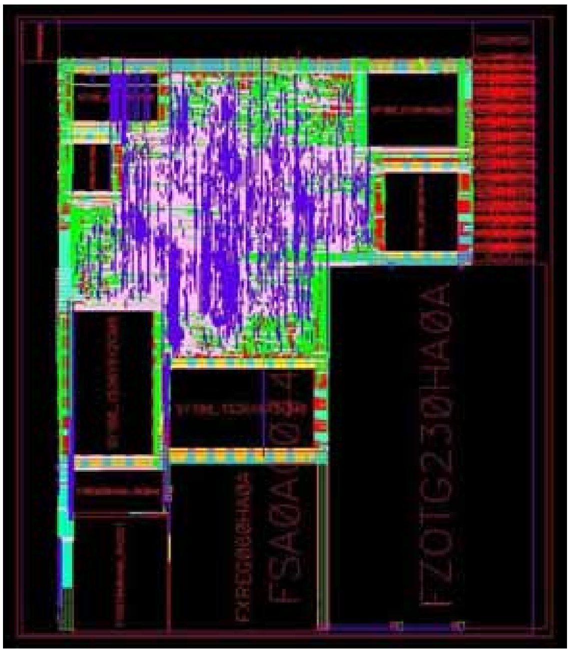


Figure 8-6. The diagram of the chip layout

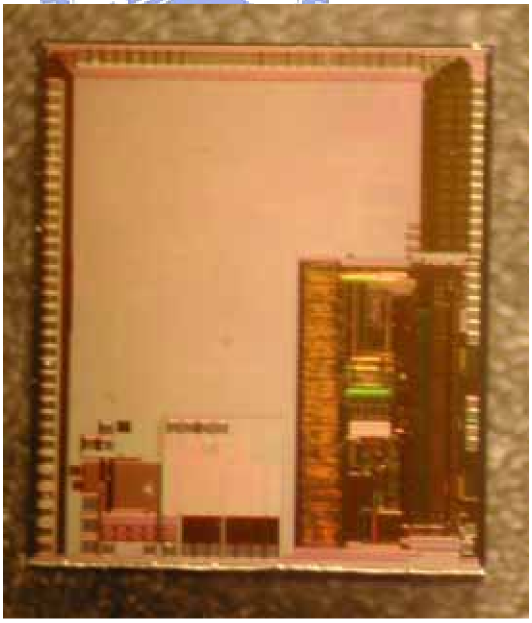


Figure 8-7. The photograph of the die of the designed controller chip

A platform to measure the power consumption and the performance was shown in Figure 8-8. For the power consumption, the current of the power supply source during the operation mode were measured. Two of the operation modes were measured: the sustained data reading and the sustained data writing. The performance was measured by the data transfer rate on the interface bus. The data transfer rate was defined as the user data transferred during a period of time in the sustained reading and writing. The test results of the measurement were shown in Table 8-III.

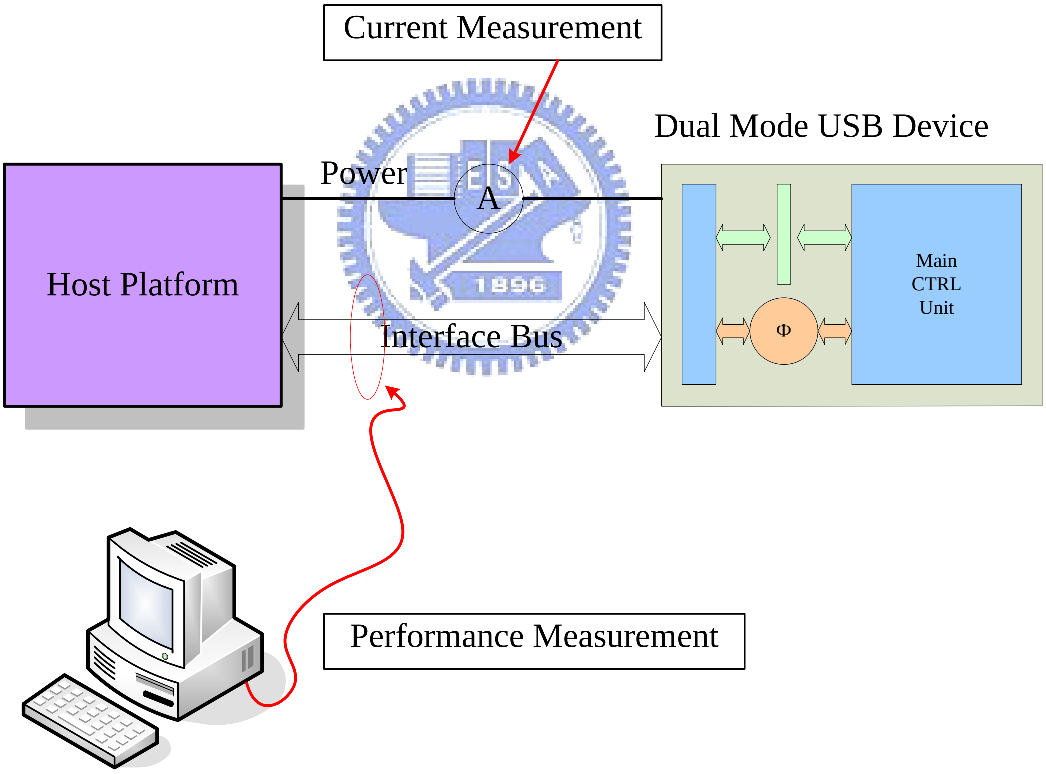


Figure 8-8. The platform for current and performance measurement

Table 8-III. The experiment results.

Items	Operation in Standard USB Bus Mode	Operation in Low Power USB Bus Mode
Power Supply VDD	+5V of USB Bus Power	+3.3V
Sustained Read Data Transfer Rate	30 MBytes / sec	31 MBytes /sec
Current Consumption at Sustained Data Read	85 mA	21 mA
Sustained Write Data Transfer Rate	18 MBytes / sec	19 MBytes /sec
Current Consumption at Sustained Data Write	95 mA	30 mA

8.3 An Architecture for High-Speed SATA Flash Controller

A high-speed SATA flash controller to support hybrid multi-channel flash memory array was shown as in Figure 8-9. The SATA II interface controller contains the physical layer, link layer, and AHCI controller to handle the SATA II command protocols and data transportation. A 12-EC and 8-EC dual-mode BCH ECC circuit is used to support the "4KB + 218B" page format as well as "2KB + 64B" page format for different kind of NAND flash memory device. The 12-EC and 8-EC BCH ECC can meet all the error correction capability required for the current flash memory devices in the market. The 8 channels flash memory bus was constructed by 8 flash sequencers in-parallel denoted as Flash #0 ~ Flash #7. The On-chip buffer RAM is used for faster cache required as the flash memory data accessing or store the operating variables or tables accessed by the MCU. The external buffer RAM has bigger size for data cache or buffering during the mass data transferring between the host and the flash memory devices. The MCU, boot ROM, and code-banking architecture are used for system firmware operation.

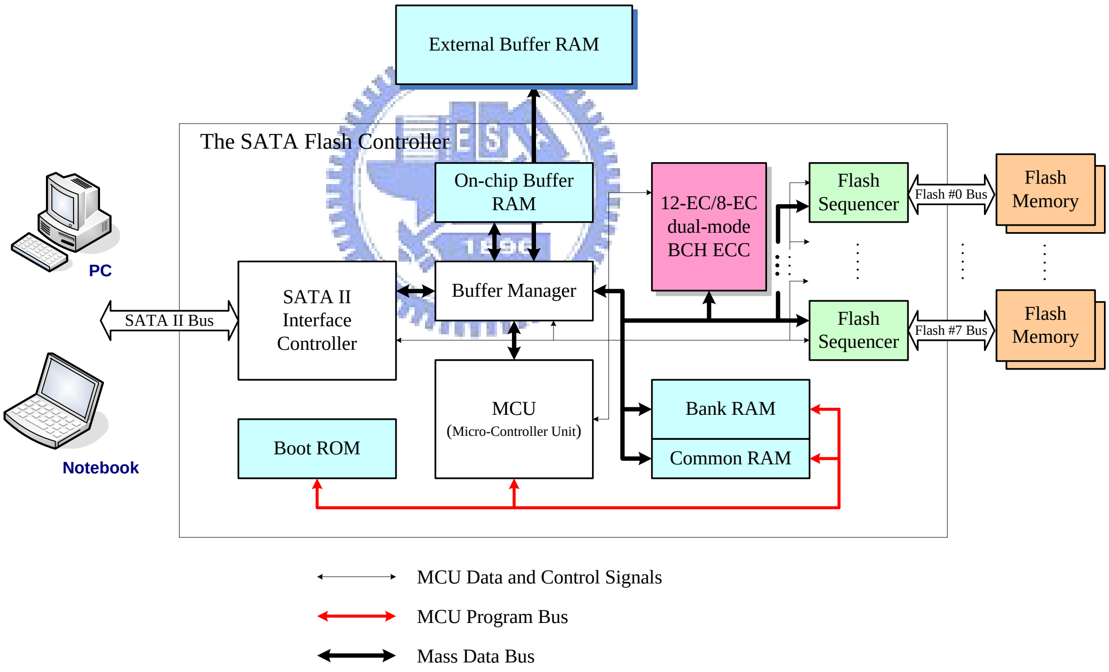


Figure 8-9. The block diagram of the high-speed SATA flash controller

The high-speed SATA II flash controller can be used to construct a high-speed, high capacity solid-state drive with hybrid multi-channel flash memory array. The proposed highest capacity of the solid-state drive can be get by 8 channels, and 4 flash memory devices in 4-way interleave operation for each channel, shown as in Figure 8-10. The flash memory channel #0 is used for 4 16-Gb SLC memory devices, while the flash memory channel #1~7 are used for total 28 (=7*4) 32-Gb MLC flash memory devices. Thus, the capacity is: 4*2GB + 28*4GB = 120GB. The maximum sequentially data transfer rate in reading and writing can be calculated as below:

The assumptions are listed as below:

- The flash parameter data is referred to the datasheet of Samsung 32Gb MLC K9LBG08U0M.
- The page program time of MLC flash memory: t_{MLC_PROG} = 800us for 4KB Page.
- Dummy busy time for 2-play program: t_{MLC_DBSY} = 0.5us
- The read/write cycle time of MLC flash memory: t_{MLC_IOCYC} = 25 ns.
- The read busy time of MLC flash memory: $t_{MLC\_R\_BSY}$ = 60 us.
- The command overheads are estimated as 10us per page in reading and 5% writing.
- Use 4KB Page 2-plane program to double the sequentially
- The SLC flash memory chips in channel #0 is using for accessing the file system hot data. We assume the speed parameters are the same with MLC ones for conservative estimation, although the speed of SLC is almost several times faster than MLC.

The estimated flash read performance is:

Read performance per flash memory channel:

= 4096 Bytes / [(4096*25ns) + 10us]

(P.S.: 60us read busy is cancelled by interleave)

= 36.44 MB/sec.

8 channels in parallel operation = 36.44*8 = 291.52 MB/sec.

The estimated flash write performance is:

Write performance per flash memory channel:

= 4096*2 Bytes / [max(4096*2*25ns, (800us + 0.5us)/4) + 50us]

(P.S.: Data-In time = 204.8us > Program time / 4 = 200.15us in 4-way interleave

= 32.15 MB/sec.

8 channels in parallel operation = 32.15*8 = 257.2 MB/sec.

The ideal bus bandwidth for SATA II is 300 MB/sec. The time-multiplexing dual-port buffer accessing of the host-side and the flash-side data transfer guarantee the optimal flash read / write performance can be maintained in the whole data transfer since 300 MB/sec is larger than flash read speed 291.52 MB/sec and flash write speed 257.2 MB/sec. However, in the practical file system data accessing, there is some command protocol overhead and bus transition handshaking time during the data transportation. The 20% overhead of the interface protocol is estimated. Thus, the estimated overall system performance is: 230 MB/sec in data reading; 205 MB/sec in data writing.

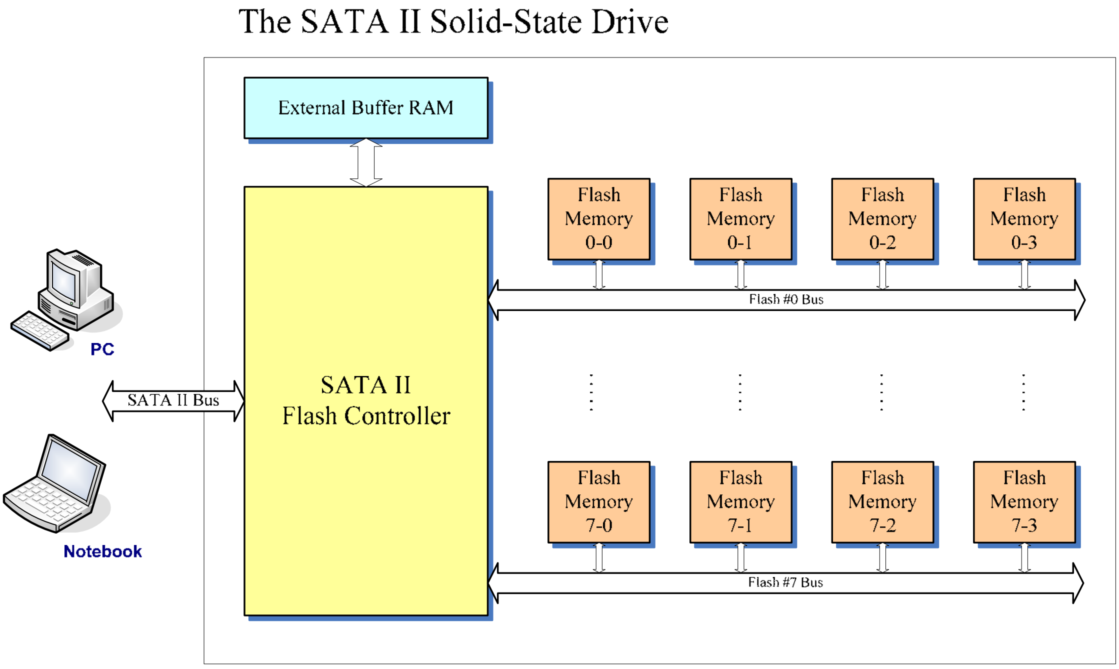


Figure 8-10. The architecture of the 120GB SATA solid-state drive.

Chapter 9 Conclusions

According to the research and discussion in this dissertation, the reliable high-speed and high-capacity solid-state drives will be required for the mobile computers, such as: notebook computers, UMPC (**U**ltra **M**obile **P**ersonal **C**omputers). We discussed and analyzed all the functional units of the solid-state drive controller in this dissertation. We presented a *t*-EC *w*-bit parallel BCH ECC circuit by the systolic array method to enhance the data integrity of the flash memory storage system. The hardware architectures for the performance enhancement were discussed and simulated. In the security of the flash storage system, we presented a circuit implementation for SD card CPRM. Architecture of hybrid multi-channel flash memory storage system was presented to attain the cost-performance optimization in the high capacity solid-state drive. The silicon realization of the controller chips have shown the architectures and design methods we discussed.

Based on the results shown in this dissertation, we think there are some future works for research as below:

- **Advanced Wear-Leveling**:
  The wear-leveling issues in the solid-state drive will become more important in the future, since the more advanced MLC flash memory will have worse in endurance. The advanced wear-leveling algorithm for hybrid non-volatile solid-state memory storage system to prolong the product lifetime will be very significant.

- **Advanced Data Security Mechanism**:

  The electronic data is very convenient to transfer and easy to duplicate, but the copyright protection is weak on the other hand. An effective data security mechanism for the copyright protection of the electronic IP (**I**ntellectual **P**roperty) is becoming more and more important. The solid-state drive contains an active controller circuit, so a more complicated and advanced data security algorithm can be implemented on it. An advanced data security mechanism shall be guaranteed-secure, effective, and convenient for the users.

- **Enhancement of data integrity**:

  To overcome the more severe impact of the disturbances during the flash memory operation of the advanced MLC flash memory, the more powerful mechanism for the data integrity enhancement of the solid-state drive is required. In the ECC of the flash controller, the linear block code like BCH or Reed-Solomon might not be enough. The more advanced ECC for the solid-state drive controller is suggested, such as: the iterative decoding algorithm, like LDPC (**L**ow-**D**ensity **P**arity-**C**heck code), or Turbo code (a kind of iterative error correction code used in the noisy communication channels).

- **Advanced Flash memory refreshing mechanism**:

  The charge decay or electrons fly-away phenomenon of the floating gate of the NAND flash memory will become serious in advanced MLC flash memory. An efficient and effective refreshing mechanism can reduce the problem, so that a long enough data retention time can be guaranteed.

- **Intelligent Self-adaptive Capability**:

  The data stored in the storage device for the computer system is like the blood for the human body. The data loss or drive failure of the solid-state drive is a disaster for the users. An intelligent self-adaptive capability of the solid-state drive can maintain the data structure, check the system parameter and do regular self-diagnosis periodically. The intelligent solid-state drive can even upgrade the firmware and maintain the health of flash data link via internet as the computer is on-line.

Bibliography

[1] Paolo Cappelletti, Carla Golla, Piero Olivo and Enrico Zanoni, Flash Memories, Kluwer Academic Publishers, 2000.

[2] D.R.Hankerson, D.G.Hoffman, D.A.Leonard, C.C.Lindner, K.T.Phelps, C.A.Rodger, J.R.Wall, Coding Theory and Cryptography, Marcel Dekker Inc., 2000.

[3] Robert J. McEliece, The Theory of Information and Coding, 2nd edition, Cambridge, 2002.

[4] William Stallings, Cryptograph and Network Security – Principles and Practice, Third Edition, Prentice-Hall, 2004.

[5] S. Y. Kung, VLSI Array Processors, Prentice-Hall, 1988.

[6] Vijay K. Madisetti, VLSI Digital Signal Processors – An Introduction to Rapid Prototyping and Design Synthesis, Butterworth-Heinemann, 1995.

[7] Michael John Sebastian Smith, Application-Specific Integrated Circuits, Addison-Wesley, 1997.

[8] Abdellatif Bellaouar and Mohamed I. Elmasry, Low-Power Digital VLSI Design Circuits and Systems, Kluwer Academic Publishers,

[9] Samir Palnitkar, Verilog HDL – A Guide to Digital Design and Syhthesis, Prentice-Hall, 1996.

[10] Douglas L. Perry, VHDL, 2nd edition, McGraw-Hill, 1994.

[11] Synopsys Inc., VHDL Compiler Reference. http://www.synopsys.com

[12] Altera Inc., MAX+PLUS II VHDL Manual. http://www.altera.com

[13] INCITS Technical Committee T13, AT Attachment, http://www.t13.org/

[14] Serial ATA International Organization (SATA-IO), Serial ATA: High Speed Serialized AT Attachment, http://www.sata-io.org/

[15] Intel Corporation, Serial ATA Advanced Host Controller Interface (AHCI), http://www.intel.com/technology/serialata/ahci.htm

[16] INCITS Technical Committee T10, SCSI Storage Interfaces, http://www.t10.org/

[17] PCI SIG, PCI (Peripheral Component Interconnect) Standard Specifications, http://www.pcisig.com/

[18] PCI SIG, PCI Express Standard Specifications, http://www.pcisig.com/

[19] IEEE p1394a Working Group (2000). IEEE Std 1394a-2000 High Performance Serial Bus - Amendment 1. IEEE.

[20] USB-IF, Universal Serial Bus (USB) Specification Revision 2.0, April 2001. http://www.usb.org/

[21] CompactFlash Association, CF+ and CompactFlash Specifications, http://www.compactflash.org/

[22] Multimedia Card Association, MultiMediaCard (MMC) System Specifications, http://www.mmca.org/

[23] SD Card Association, SD Card Physical Layer Specification, http://www.sdcard.org/

[24] SD Card Association, SD Card Security Specification, http://www.sdcard.org/

[25] Data Encryption Standard (DES), FIPS PUB 46-3, Oct. 1999.

[26] CPRM (Content Protection for Recordable Media) Specification, 4C Entity, http://www.4centity.com/

[27] Michael Ripley et al, “Content Protection in the Digital Home,” Intel Technology Journal, Vol. 6, Issue 4, 2002.

[28] C2 Block Cipher Specification Version 1.00, 4C Entity, January 2003. http://www.4centity.com/

[29] Memory Stick Association, Memory Stick Standard Specifications, http://www.memorystick.org/

[30] ONFI (Open NAND Flash Interface) Association, ONFI Specifications, http://www.onfi.org/

[31] SmartMedia ECC Reference Manual Ver:2.1 (Software & Hardware), Toshiba Corporation, September 1999.

[32] SmartMedia Software Algorithm Guidelines Version 1.10, SSFDC Forum Technical Committee, April 2003.

[33] SmartMedia Interface Library (SMIL) Software Edition Version 1.10, Toshiba Corporation, February 2004.

[34] SmartMedia Interface Library (SMIL) Hardware Edition Version 1.00, Toshiba Corporation, July 2000.

[35] Paul Massiglia, The RAID book – Storage System Technology Handbook, 6th edition, RAID Advisory Board, 1997, ISBN 1-8799-90-9.

[36] Douglas R. Stinson, Cryptography – Theory and Practice, Second Edition, CRC Press LLC, 2002.

[37] Neil H. E. Weste and Kamran Eshraghian, Principles of CMOS VLSI Design – A Systems Perspective, Second Edition, Addison-Wesley, 1994.

[38] Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, 1993.

[39] Howard W. Johnson and Martin Graham, High-Speed Digital Design – A Handbook of Black Magic, Prentice-Hall, 1993.

[40] Don Anderson, USB System Architecture, MINDSHARE INC., Addition-Wesley Developer's Press, 1997.

[41] Don Anderson, USB System Architecture (USB 2.0), MINDSHARE INC., Addition-Wesley Developer's Press, 2001.

[42] Michael D. Ciletti, Advanced Digital Design with the Verilog HDL,

Prentice-Hall, 2003.

[43] Weng Fook Lee, Verilog Coding for Logic Syhthesis, John Wiley & Sons, 2003.

[44] Kevin Skahill, VHDL for Programmable Logic, Addison-Wesley, 1996.

[45] George Lawton, "Improved Flash Memory Grows in Popularity," Computer, IEEE Computer Society, Jan. 2006.

[46] Chang-Gyu Hwang, "Nanotechnology Enables a New Memory Growth Model," Proceedings of the IEEE, Vol. 91, No. 11, Nov. 2003.

[47] Sang Lyul Min Eyee Hyun Nam, "Current Trends in Flash Memory Technology," Asia and South Pacific Conference on Design Automation, January 2006.

[48] Atsushi Inoue and Doug Wong, "NAND Flash Applications Design," Toshiba Semiconductor Co., April 2003.

[49] Samsung Semiconductor Co., "Application Note for NAND Flash Memory," December 1999.

[50] Intel Corporation, "Understanding the Flash Translation Layer (FTL) Specification," Application Note 648, 1998. http://developer.intel.com/design/flcomp/applnots/297816.htm.

[51] Gal, E., and Toledo, S., "Mapping structures for flash memories: techniques and open problems," In Proceedings of the IEEE International Conference on Software-Science, Technology and Engineering, SwSTE 05, Feb. 22 - 23,

2005.

[52] Samsung Electronics Co., NAND Flash Memory Data Book, 2007, http://www.samsung.com/.

[53] Toshiba Semiconductor Co., NAND Flash Memory Data Book, 2007, http://www.semicon.toshiba.co.jp/eng/product/memory/selection/.

[54] Hynix Semiconductor Inc., NAND Flash Memory – Technical Data Sheet, http://www.hynix.com/eng/02_products/03_flash/index.jsp

[55] Intel Corporation, NAND Flash Memory Data Sheet, http://www.intel.com/design/flash/nand/index.htm

[56] Micron Technology Inc., NAND Flash Memory Data Sheet, http://www.micron.com/products/nand/

[57] STMicroelectronics, NAND Flash Memory Data Sheet, http://www.st.com/stonline/products/families/memories/fl_nand/

[58] SPANSION Technology Corporation, Flash Memory Data Sheet, http://www.spansion.com/flash_memory_products/

[59] Stefano Gregori, Alessandro Cabrini, Osama Khouri and Guido Torelli, "On-Chip Error Correcting Techniques for New-Generation Flash Memories," Proceedings of the IEEE, Vol. 91, No. 4, April 2003.

[60] Toru Tanzawa et al, "A Compact On-Chip ECC for Low Cost Flash Memories," IEEE Journal of Solid-State Circuits, Vol. 32, No. 5, May 1997.

[61] H. C. Chang, C. C. Lin, T. Y. Hsiao, J. T.Wu, and T. H.Wang, "Multilevel

memory systems using error control codes," Proc. Int. Symp. Circuits and Systems (ISCAS), pp. II 393–II 396, 2004.

[62] Tong-Bi Pei and Charles Zukowski, "High-speed Parallel CRC Circuits in VLSI," IEEE Trans on Communications, Vol. 40, No. 4, Apr. 1992. pp 653-657.

[63] Xinmiao Zhang and K. K. Parhi, "High-Speed Architectures for Parallel Long BCH Encoders," IEEE Trans on Very Large Scale Integration (VLSI) Systems, Vol. 12, No. 5, Jul. 2005. pp 872-877.

[64] K. K. Parhi, "Eliminating the fan-out bottleneck in parallel long BCH encoders," Journal, IEEE Trans on Circuits and Systems, 3(51), pp. 512-516, 2004.

[65] Jun Zhang, Zhi-Gong Wang, Qing-Sheng Hu and Jie Xiao, "Optimized Design for High-speed Parallel BCH Encoder," IEEE Int. Workshop VLSI Design & Video Tech. 2005, pp. 97-100.

[66] Yanni Chen and K. K. Parhi, "Small Area Parallel Chien Search Architectures for Long BCH Codes," IEEE Trans on Very Large Scale Integration (VLSI) Systems, Vol. 12, No. 5, May 2004, pp. 545-549.

[67] Yuejian Wu, "Low Power Decoding of BCH Codes," INCAS 2004, pp. II 369-372.

[68] Keiichi Iwamura, Yasunori Dohi and Hideki Imai, "A Design of Reed-Solomon Decoder with Systolic-Array Structure," Journal, IEEE Trans on Computers, Vol. 44, No. 1. Jan. 1995, pp. 118-122.

[69] Howard M. Shao and Irving S. Reed, "On the VLSI Design of a Pipeline Reed-Solomon Decoder Using Systolic Arrays," Journal, IEEE Trans on Computers, Vol. 31, No. 10, Oct. 1988, pp. 1273-1280.

[70] Gadiel Seroussi, "A Systolic Reed-Solomon Encoder," IEEE Trans on Information Theory, Vol. 37, No. 4, Jul. 1991, pp. 1217-1220.

[71] Chanik Park, Seungmo Cho, Jaewook Lee and Hyungjun Park, "Co-Validation Environment for Memory Card Compatibility Test: A Case Study," Proceedings of 15th IEEE International Workshop on Rapid System Prototyping, June 2004, pp. 62-65.

[72] Chuan-Sheng Lin and Lan-Rong Dung, "A NAND Flash Memory Controller for SD/MMC Flash Memory Card," IEEE Trans. Magnetics, Vol. 43, No. 2, pp. 933-935, Feb. 2007.

[73] Chuan-Sheng Lin, Kuang-Yuan Chen, Yu-Hsian Wang and Lan-Rong Dung, "A NAND Flash Memory Controller for SD/MMC Flash Memory Card," 13th ICECS (13th IEEE International Conference on Electronics, Circuits, and Systems), Dec. 2006.

[74] Chuan Sheng Lin, Chen Nan Lai, Kuang Yuan Chen, "Memory Array Apparatus With Reduced Data Accessing Time and Method for the same," US Patent No. 6,718,406 B2, April 2004.

[75] Petro Estakhri and Berhanu Iman, "Increasing the Memory Performance of Flash Memory Devices by Writing Sectors Simultaneously to Multiple Flash Memory Devices," US Patent No. 6,757,800 B1, June 2004.

[76] Kevin M. Conley and Reuven Elhamias, "Flash Controller Cache

Architecture," US Patent No. 7,173,863 B2, February 2007.

[77] Chanson Lin, Joe Shyu, "Apparatus and Method Accessing Flash Memory," US Patent No. 6,237,110 B1, May 2001.

[78] Chen Nan Lai, Yao Tse Chang, Kuo-Hong Wang, Chanson Lin, "Algorithm of Flash Memory Capable of Quickly Building Table and Preventing Improper Operation and Control System Thereof," US Patent No. 6,711,663 B2, March 2004.

[79] Carlos J. Gonzalez and Kevin M. Conley, "Flash Memory Data Correction and Scrub Techniques," US Patent No. 7,224,607 B2, May 2007.

[80] Chanson Lin and Joe Shyu, "Memory Access Control Device, And Its Control Method," US Patent No. 6,167,549, December 2000.

[81] Ken Takeuchi and Tomoharu Tanaka, "A Dual-Page Programming Scheme for High-Speed Multigigabit-Scale NAND Flash Memories," IEEE Journal of Solid-State Circuits, Vol. 36, No. 1, November 2001.

[82] Robert C Chang, Bahman Qawami and Farshid Sabet-Sharghi, "Method and Apparatus for Performing Block Caching in a Non-Volatile Memory System," US Patent No. 7,174,440 B2, February 2007.

[83] Woodhouse, D., "JFFS2: The Journaling Flash File System," http://sources.redhat.com/jffs2/jffs2.pdf.

[84] Yet Another Flash Filing System (YAFFS). http://www.aleph1.co.uk/yaffs.

[85] L.P. Chang and T.-W. Kuo, "An Efficient Management Scheme for

Large-Scale Flash-Memory Storage Systems," Proc. ACM Symp. Applied Computing, Mar. 2004.

[86] L.P. Chang and T.W. Kuo, "A Real-Time Garbage Collection Mechanism for Flash Memory Storage System in Embedded Systems," Proc. Eighth Int'l Conf. Real-Time Computing Systems and Applications, 2002.

[87] Seung-Ho Lim, and Kyu-Ho Park, "An Efficient NAND Flash File System for Flash Memory Storage," IEEE Trans. on Computers, Vol. 55, No. 7, July 2006.

[88] Sudeep Jain Yann-Hang Lee, "Real-Time Support of Flash Memory File System for Embedded Applications," Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06), 2006.

[89] Taehee Cho et al, "A Dual-Mode NAND Flash Memory: 1-Gb Multilevel and High-Performance 512-Mb Single-Level Modes," IEEE Journal of Solid-State Circuits, Vol. 36, No. 1, November 2001.

[90] Jae-Duk Lee, Jeong-Hyuk Choi, Donggun Park, and Kinam Kim, "Data Retention Characteristics of Sub-100 nm NAND Flash Memory Cells," IEEE Electron Device Letters, Vol. 24, No. 12, December 2003.

[91] C. Paar, P. Fleischmann, P. S-Rodriguez, "Fast Arithmetic for Public-Key Algorithms in Galois Fields with Composite Exponents," IEEE Trans. on Computers, October 1999, Vol. 48, No.10, pp.1025-1034.

[92] Neal Koblitz, "Elliptic Curve Cryptosystems," Mathematics of Computation,

48, N.177 (1987), pp.203-209.

[93] R. L. Rivest, A. Shamir, and L. Adleman, “Public Key Cryptography,” CACM 21, 1978, pp.120-126.

[94] Ernest F. Brickell, “A Survey of Hardware Implementations of RSA,” in Gilles Brassard, editor, Advances in Cryptology - Crypto ’89, pp. 368-370.

[95] C.-C. Yang, T.-S. Chang, and C.-W. Jen, “ A New RSA Cryptosystem Hardware Design based on Montgomery’s Algorithm,” IEEE Trans. on Circuits and Systems II, Vol.45, pp.908-913, July 1998.

[96] S. Moon, J. Park, and Y. Lee, “Fast VLSI Arithmetic Algorithms for High-Security Elliptic Curve Cryptographic Applications,” IEEE Trans. on Consumer Electronics, August 2001, Vol. 47, No. 3, pp. 700-708.

[97] J. Goodman and A. P. Chandrakasan, “An Energy-Efficient Reconfigurable Public-Key Cryptography Processor,” IEEE Journal of Solid-State Circuits, November 2001, Vol.36, No. 11, pp.1808-1820.

[98] M. Aydos, T. Yanik, and C.K. Koc, “High-Speed Implementation of an ECC-Based Wireless Authentication Protocol on an ARM Microprocessor,” IEE Proc. Commun., October 2001, Vol. 148, No. 5, pp.273-279.

[99] S. R. Dusse, B. S. Kaliski Jr., “A Cryptographic Library for the Motorola DSP 56000,” Proceedings of EUROCRYPT ’90, Springer LNCS 473, 1991.

[100] Ali Sheikholeslami and P. G. Gulak: A survey of circuit innovations in Ferroelectric random-access memories, Proceedings of the IEEE, Vol. 88, No. 3, pp. 667-689, May 2000.

[101] Magnetoresistive Random Access Memory, Freescale Semiconductor, Inc. June 2006. http://www.freescale.com/files/memory/doc/

[102] Ovonic Unified Memory Technical Presentation , Ovonyx Inc., http://ovonyx.com/technology/technical-presentation.html

[103] Brian Dees, “Native command queuing - advanced performance in desktop storage,” IEEE Potentials, Vol. 24, Issue 4, pp.4-7, Oct. / Nov. 2005.

[104] Prabuddha Biswas, K. K. Ramakrishnan and Don Towsley, “Exploiting Non-volatile Memory in Disks for Write Caching,” Computer Science Department, University of Massachusetts, 1994. available at: http://citeseer.ist.psu.edu/306928.html

[105] Baker, M., et al, “Non-Volatile Memory for Fast, Reliable File Systems,” Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOSV), October, 1992.

[106] David B. Anderson et al, “Disc Storage System Having a Non-Volatile Cache to Store Write Data in The Event of a Power Failure,” US Patent No. 6,295,577 B1, Sep. 2001.

[107] Behram Mario DaCosta, “Non-Volatile Memory System For Instant-On,” US Patent No. 6,564,286 B2, May 2003.

[108] Charles Severance, “Linking Computers and Consumer Electronics,” Standards, February 1997.

[109] Chuan-Sheng Lin, and Lan-Rong Dung, “A Dual-Mode USB Interface

Controller Chip Design for Low-Power Mobile Devices," WSEAS Transactions on Circuits and Systems, Issue 3, Volume 6, pp.380-388, March 2007.

[110] .I. Reed, M. Shih, and T. Truong, "VLSI design of inverse-free Berlekamp–Massey algorithm," Proc. IEE, pt. E, vol. 138, pp. 295–298, September 1991.

[111] 林傳生, 使用 VHDL 電路設計語言之數位電路設計, 第六版, 儒林圖書, 2002.

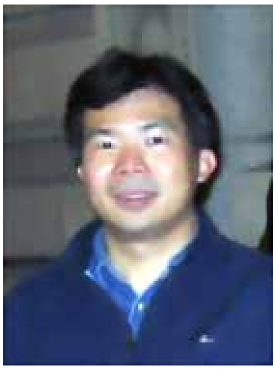
[112] 林傳生, MATLAB 之使用與應用, 第十一版, 儒林圖書, 2004.

[113] 鍾慶豐, 近代密碼學與其應用, 儒林圖書, 2005.

[114] 陳希孟, 蕭亮星等合著, 硬式磁碟機原理, 碁峰資訊, 1995.

[115] 林銀議, 數位通訊原理 – 編碼與消息理論, 五南出版社, 2005.

作者簡歷與著作(Author's Information and Publications)



姓名： 林傳生(Chuan-Sheng Lin)

指導教授： 董蘭榮(Lan-Rong Dung)

學號： 8712801

出生年月日: 民國 56 年 9 月 18 日

聯絡地址: 新竹縣竹北市文采街 36 巷 26 號
聯絡電話: 0922-289329
e-mail: chanson@ntu.edu.tw
chan.ece87g@nctu.edu.tw

學歷：

1.	71 年 9 月~74 年 6 月	台灣省立宜蘭高級中學
2.	74 年 9 月~78 年 6 月	國立清華大學動力機械工程學系
3.	78 年 9 月~80 年 6 月	國立台灣大學機械工程學系研究所 系統控制組
4.	87 年 9 月~迄今	國立交通大學電機與控制工程學系博士學位

經歷：

1.	82 年 6 月~83 年 6 月	弘一科技 伺服控制工程師
2.	83 年 7 月~84 年 12 月	工研院光電所 電子工程師
3.	85 年 1 月~92 年 11 月	太和科技 研發協理、研發副總、總經理
4.	93 年 1 月~94 年 11 月	瑞程科技 總經理
5.	94 年 12 月~迄今	旺玖科技 副總經理

Publications

Journal Papers

1. ***Chuan-Sheng Lin*** and Lan-Rong Dung, "A NAND Flash Memory Controller for SD/MMC Flash Memory Card," IEEE Transactions on Magnetics, Vol. 43, No. 2, pp.933-935, February 2007.

2. ***Chuan-Sheng Lin*** and Lan-Rong Dung, "**A** Dual-Mode USB Interface Controller Chip Design for Low-Power Mobile Devices," WSEAS Transactions on Circuits and Systems, Issue 3, Volume 6, pp.380-388, March 2007.

3. ***林傳生,*** "硬式磁碟機之讀寫技術原理介紹," 電子月刊, 第 25 期, August 1997. (中文期刊)

Conference Papers

1. ***Chuan-Sheng Lin***, Kuang-Yuan Chen, Yu-Hsian Wang, and Lan-Rong Dung, "A NAND Flash Memory Controller for SD/MMC Flash Memory Card," 13th ICECS (13th IEEE International Conference on Electronics, Circuits, and Systems), December 2006. (Nice, France.)

2. ***Chuan-Sheng Lin***, Kuang-Yuan Chen, and Lan-Rong Dung, "A NAND Flash Memory Controller for SD/MMC Flash Memory Card," APDSC'06 (Asia-Pacific Data Storage Conference 2006), August 2006. (Hsinchu, Taiwan.)

3. ***Jia-Yush Yen, Chuan-Sheng Lin***, Chung-Han Li, and Yung-Yaw Chen, "Servo Controller Design For An Optical Disk Drive Using Fuzzy Control Algorithm," FUZZY-IEEE IEEE Int. Conference on Fuzzy Systems, San Diego, CA, March 8-12, 1992.

Patent (US)

1. Chung-Liang Lee (Taipei, TW); ***Chanson Lin (Jhubei, TW)***; Ken Tsai (Taipei, TW), "Structure of USB Compatible Application Apparatus," US Patent 7,165,998 B2, Jan. 23, 2007.

2. Chen Nan Lai (Hsinchu, TW); ***Chanson Lin (Hsinchu, TW)***; Tsair-Jinn Cheng (Hsinchu, TW), "Detection Method used in Adaptor Capable of Inserting Various Kinds of Memory Cards," US Patent 6,725,291 B2, Apr. 20, 2004.

3. ***Chuan Sheng Lin*** **(Hsinchu, TW)**; Chen Nan Lai (Hsinchu, TW); Kuang Yuan Chen (Hsinchu, TW), “Memory Array Apparatus With Reduced Data Accessing Time and Method for the same,” US Patent 6,718,406 B2, Apr. 6, 2004.

4. Chen Nan Lai (Hsinchu, TW); Yao Tse Chang (Hsinchu, TW); Kuo-Hong Wang (Hsinchu, TW); ***Chanson Lin (Hsinchu, TW)***, "Algorithm of Flash Memory Capable of Quickly Building Table and Preventing Improper Operation and Control System Thereof,” US Patent 6,711,663 B2, Mar. 23, 2004.

5. Chen Nan Lai (Hsinchu, TW); Tsair-Jinn Cheng (Hsinchu, TW); Shang Chin Chien (Hsinchu, TW); ***Chanson Lin (Hsinchu, TW)***, "Control Device Applicable to Flash Memory Card and Method for Building Partial Lookup Table,” US Patent 6,704,852 B2, Mar. 9, 2004.

6. Shimon Chen (Hsinchu, TW); ***Chuan Sheng Lin (Hsinchu, TW)***; Yu-Ting Chiu (Hsinchu, TW); Cheng Wei Yang (Hsinchu, TW), “RAID Device for Establishing a Direct Passage Between a Host Computer and a Hard Disk By a Data Hub Selectively Passing Only Data To Be Accessed,” US Patent 6,671,751 B1, Dec. 30, 2003.

7. ***Chanson Lin (Hsinchu, TW)***; Joe Shyu (Hsinchu, TW), “Apparatus and Method Accessing Flash Memory,” US Patent 6,237,110 B1, May 22, 2001.

8. ***Chanson Lin (Hsinchu, TW)***; Joe Shyu (Hsinchu, TW), “Memory Access Control Device, And Its Control Method,” US Patent 6,167,549, Dec. 26, 2000.

Patent (Taiwan, ROC)

1. 王裕賢; ***林傳生***; 吳東賢; 蘇建彰; 林高正; 徐慶鐘; 陳光原, “快閃記憶體儲存系統”, 中華民國專利 I275101, Mar. 1, 2007.

2. 李鐘亮; ***林傳生***; 蔡瑞隆, “可防止靜電破壞之記憶裝置”, 中華民國專利 I274440, Feb. 21, 2007.

3. ***林傳生***; 李鐘亮; 戴瑞鎔, “具有多重介面功能之記憶卡及其傳輸模式選擇方法”, 中華民國專利 I271659, Jan. 21, 2007.

4. 王裕賢; ***林傳生***; 洪裕智, “多程式組韌體執行方法及裝置”, 中華民國專利 I270006, Jan. 1, 2007.

5. 賴振楠; 鄭才進; 簡尚進; ***林傳生***, “可適用於快閃記憶卡之控制裝置及其建構方法”, 中華民國專利 I264678, Oct. 21, 2006.

6. ***林傳生***; 王裕賢; 黃宏嘉; 范伯政, “具多緩衝器之資料儲存器及其存取方法”, 中華民國專利 I254881, May 11, 2006.

7. ***林傳生***; 李鐘亮; 徐慶鐘, “資料儲存裝置”, 中華民國專利 I248617, Feb. 1, 2006.

8. 戴瑞鎔; 李鐘亮; ***林傳生***, “可降低工作電流之 USB 傳輸介面裝置”, 中華民國專利 I247216, Jan. 11, 2006.

9. 李鐘亮; ***林傳生***; 蔡瑞隆, “SATA 傳輸介面之改良構造及其應用裝置”, 中華民國專利 I241758, Oct. 11, 2005.

10. 李鐘亮; ***林傳生***; 蔡瑞隆, “可提高隔絕溫溼度效益的 CF 儲存碟結構”, 中華民國專利 M268659, Jun. 21, 2005.

11. 李鐘亮; ***林傳生***; 蔡瑞隆, “USB 介面裝置之改良結構”, 中華民國專利 M267691, Jun. 11, 2005.

12. ***林傳生***; 李鐘亮, “可延遲影像播放時間之預錄影像機”, 中華民國專利 M261943, Apr. 11, 2005.

13. 賴振楠; 張耀澤; ***林傳生***, “一種在及閘系統中增進效率且增長使用壽命的控制方法”, 中華民國專利 I230947, Apr. 11, 2005.

14. ***林傳生***; 李鐘亮; 戴瑞鎔, “具有多重介面功能之記憶卡”, 中華民國專利 M258369, Mar. 1, 2005.

15. 戴瑞鎔; 李鐘亮; ***林傳生***, “USB 傳輸介面裝置”, 中華民國專利 M258347, Mar. 1, 2005.

16. 賴振楠; ***林傳生***, “內建基本輸入輸出系統之小型儲存裝置”, 中華民國專利 M241755, Aug. 21, 2004.

17. 賴振楠; ***林傳生***, “具快取設計之資料儲存裝置”, 中華民國專利 M240662, Aug. 11, 2004.

18. ***林傳生***; 邱裕庭; 顏址良; 陳景湖; 王國鴻, “儲存媒體之資料保密裝置及資料保密方法”, 中華民國專利 00591630, Jun. 11, 2004.

19. 賴振楠; ***林傳生***, “一種可立即偵測剩餘記憶體容量的儲存裝置及其方法”, 中華民國專利 00591418, Jun. 11, 2004.

20. 賴振楠; ***林傳生***; 鄭才進, “用於可插接多種記憶卡轉接器之偵測方法”, 中華民國專利 00584808, Apr. 21, 2004.

21. 賴振楠; ***林傳生***, “一種包含有主機端 USB 介面之多功能晶片”, 中華民國專利 00568356, Dec. 21, 2003.

22. 賴振楠; ***林傳生***, “記憶卡轉接裝置”, 中華民國專利 00566625, Dec. 11, 2003.

23. 陳希孟; ***林傳生***; 邱裕庭; 鄭才進; 徐慶鐘, “多重讀卡光碟機”, 中華民國專利 00555130, Sept. 21, 2003.

24. ***林傳生***; 賴振楠, “可獨立操作記憶卡轉存裝置”, 中華民國專利 00539188, Jun. 21, 2003.

25. ***林傳生***; 賴振楠; 王國鴻, “一種在快閃記憶體儲存裝置中可自動建立檔案系統之方法”, 中華民國專利 00531752, May 11, 2003.

26. 賴振楠; 張耀澤; 王國鴻; ***林傳生***, “一種快閃記憶體中快速且能防止不正常斷電的演算法”, 中華民國專利 00531698, May 11, 2003.

27. ***林傳生***; 賴振楠; 陳光原, “可縮短資料存取時間之記憶體陣列裝置及其資料存取方法”, 中華民國專利 00531696, May 11, 2003.

28. ***林傳生***; 邱裕庭; 賴振楠; 陳光原; 樓仁傑, “可支援多實體設備之中繼控制裝置”, 中華民國專利 00512946, Dec. 1, 2002.

29. 徐慶鐘; ***林傳生***, “可提高資料存取可靠度的控制裝置及方法”, 中華民國專利 00461993, Nov. 1, 2001.

30. ***林傳生***; 邱裕庭; 賴振楠 陳光原, “具有熱插拔功效之硬碟抽取裝置”, 中華民國專利 00460021, Oct. 11, 2001.

31. 陳希孟; ***林傳生***; 邱裕庭; 楊政偉, “具有命令處理與資料存取配接傳輸功能之硬碟櫃”, 中華民國專利 00450403, Aug. 11, 2001.

32. 賴振楠; ***林傳生***, “可攜式無線資料轉存裝置”, 中華民國專利 00449691, Aug. 11, 2001.

33. ***林傳生***; 邱裕庭; 陳希孟,“可增加資料可靠度之矽硬碟機結構改良”, 中華民國專利 00408839, Oct. 11, 2000.

34. 陳希孟; ***林傳生***; 邱裕庭; 黃意翔, “數字錶自動讀錶裝置”, 中華民國專利 00404504, Sept. 1, 2000.

35. ***林傳生***; 徐慶鐘, “快閃記憶體之連結架構與演算法”, 中華民國專利 0033648, Jun. 11, 1998.

36. 仰崇仁; ***林傳生***; 邱裕庭; 陳希孟, “可內建於電腦系統中之硬碟櫃”, 中華民國專利 00303948, Apr. 21, 1997.

Books

1. ***林傳生***, 數位訊號處理器(DSP)簡介與應用，全華圖書, 1996-1997.
2. ***林傳生***, MATLAB 之使用與應用 (含 SIMULINK FOR WINDOWS), 儒林圖書, 1998-2004 (共十二版).
3. ***林傳生***, 使用 VHDL 電路設計語言之數位電路設計，儒林圖書，2000-2007 (共七版).
4. ***林傳生***, 悟性控制於複合驅動器之應用，台灣大學 碩士論文, June 1991.