



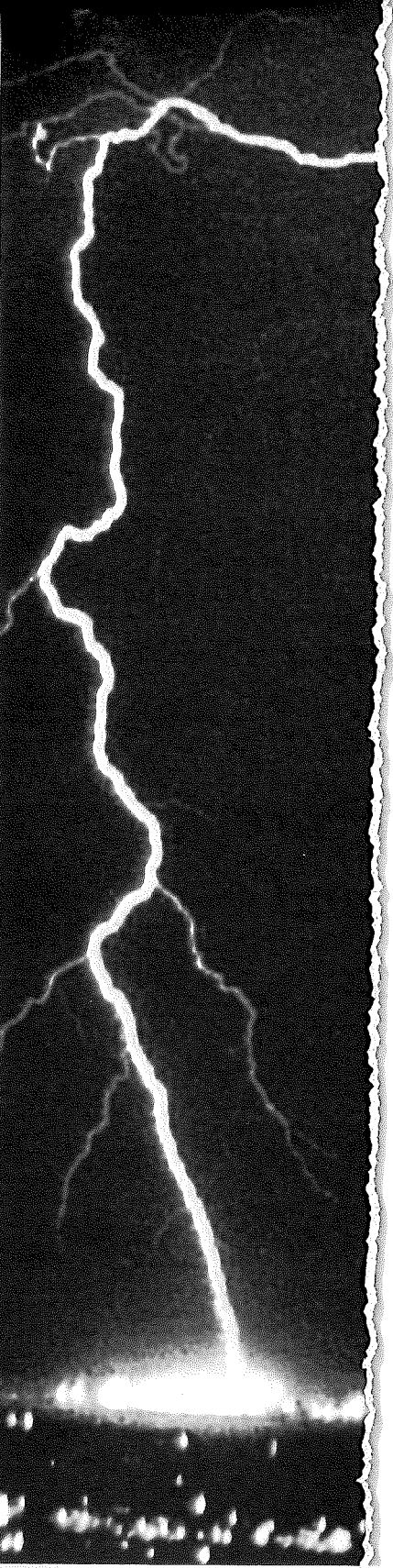
Designing with FLASH MEMORY

Brian Dipert and Markus Levy

The definitive guide to
designing flash memory
hardware and software
for components and
PCMCIA cards.

Designing with Flash Memory

*The definitive guide to
designing flash memory
hardware and software
for components and
PCMCIA cards*



Designing with Flash Memory

Brian Dipert & Markus Levy

*The definitive guide to
designing flash memory
hardware and software
for components and
PCMCIA cards*

*Annabooks
San Diego*

Designing with Flash Memory

BY

BRIAN DIPERT & MARKUS LEVY

PUBLISHED BY

Annabooks

11848 Bernardo Plaza Ct., Suite 110

San Diego, CA 92128

USA

619-673-0870

Copyright © Annabooks 1993, 1994

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the prior written permission of the publisher, except for the inclusion of brief quotations in a review.

Printed in the United States of America

ISBN 0-929392-17-5

Second Printing April 1994

Acknowledgements

Both Brian and Markus would like to thank their colleagues at Intel Corporation. Knowingly (or unknowingly), their knowledge and expertise was a superb resource base for answers to questions we uncovered as we wrote. Special recognition goes out to Dave Cobb, Russ Eslick, Kurt Robinson, John Garney, Clark Thurlo, Peter Torelli, Steve Wells, Saul Zales, Greg Komoto, Larry Leszczynski, and John Williams; thanks, guys! Thanks also goes out to Wink Saville at Saville Associates, David Lawrence at Ventura Micro, and Amir Ban at M-Systems.

Several years of staffing the customer and sales force support hotline at Intel have also taught us what *not* to do when designing with flash memory (and also given us some great ideas we hadn't considered ourselves). To the salespersons and design engineers we've worked with; thanks for an inexhaustible list of topics for this book. We hope we've included answers for at least most of the questions you've asked us!

To our "partner in crime" Mary Ann Hooker (You Know), a big round of applause for the great graphics work and for putting up with us and our seemingly endless revisions. If it weren't for you, the figures would have looked like kindergarden crayon work!

Markus: "Words cannot express the thanks and love that I give to Danielle for all her support and for all the times she patiently listened to me say 'It's almost done'. Well it's finally done. Thank you to Jacob (although you can't read this yet) for being a great kid while your Dad worked, worked and worked."

Brian: "I give thanks to the Creator God, to my family back in Indiana (thanks for all your encouragement!) and to Lil, for patiently putting up with all those nights and weekends I spent looking at a computer monitor instead of at her. I hereby retire the notebook PC (at least for a while) and echo Markus' comment.....it's finally done!".

Information provided in this publication is derived from various sources, standards, and analyses. Any errors or omissions shall not imply any liability for direct or indirect consequences arising from the use of this information. The publisher and authors make no warranty for the correctness or for the use of this information, and assume no liability for direct or indirect damages of any kind arising from technical interpretations or technical explanations in this book, for typographical or printing errors, or for any subsequent changes.

The publisher and authors reserve the right to make changes in this publication without notice and without incurring any liability.

The publisher and editor would like to extend his thanks to Marco Thompson and Gary Lorenc of Doctor Design, Inc. and to Ray Weiss of EDN who reviewed the book on a very tight schedule while it was still in three ring binder form. Thanks also to Rich Sjoberg who did the artwork for the cover.

Cover design by John P. Choisser, Annabooks.

Cover photo by Peter Menzel, Napa, California (707-255-3528). Peter caught this lightning bolt along with dozens of others one night in Tucson, Arizona. For more of Peter's work, see the *National Geographic* July 1993 issue. Peter has taken photos all over the world for *Geo*, *Life*, *Forbes*, *Stern*, *U.S. News & World Report*, *Smithsonian*, *The New York Times Magazine*, and many others. The glow at the bottom of this bolt is probably not the flash memory at the University of Arizona being re-programmed all at once. It is more likely the stadium, which puts out a pretty healthy glow of its own, lighting or no.

The following trademarks mentioned in this book are the property of the companies listed: ETOX, i386, i486, Pentium, FlashFile, and Boot Block are the property of Intel Corporation. Microsoft, MS-DOS, MS-FFS, and Windows are trademarks of Microsoft Corporation. Other trademarks are credited as they appear.

Contents

Chapter One: Introduction	1
Flash Memory Compared to Other Memories	3
ROM (Read-Only-Memory)	4
RAM (Random-Access-Memory).....	4
EEPROM (Electrically-Eraseable-Programmable-Read-Only-Memory)	5
Magnetic Mass Storage.....	5
An Emerging Alternative: Flash Memory.....	5
A Preview of Chapters to Follow.....	7
Chapter Two: Flash Memory Applications.....	11
Data Accumulation	12
Medical Instrumentation	12
Flight Recorders.....	13
More Data Accumulation Examples.....	13
Why Flash Memory for Data Acquisition?.....	14
Data/Lookup Table Storage	14
PBX Switcher.....	14
Laser Printers	15
Why Flash Memory For Data/Lookup Table Storage?.....	16
Embedded Code Storage	16
PC BIOS	16
Digital Cellular Phones	18
More Embedded Code Storage Applications.....	18
Why Flash Memory for Embedded Code Storage?	18
File Storage	19
Flash Memory Promotes Longer Battery Life	19
HDD Densities with FDD Interchange	20
Summary	21
Chapter Three: Flash Memory Technologies	23
NOR Flash Memory	25
Program.....	27
Erase.....	28
Negative Gate Erase.....	29
Overerase	29
NOR Flash Memory Specifications	30

Flash EEPROM	32
Erase	33
Programming	34
Flash EEPROM Memory Specifications	35
NAND Flash Memory	36
Program and Erase	36
NAND Flash Memory Specifications	36
What's All This <i>Cycling</i> Stuff, Anyway?	38
Failure Analysis	40
Oxide Breakdown	40
Electron Trapup	41
Mean Time Before Failure	41
Extended Cycling-The Vendor's Options	42
Extended Cycling-What Can You Do?	43
Summary	44
Chapter Four: Packaging Options and Update Alternatives	45
Packaging Options	46
DIP (Dual In-Line Package)	46
LCC (Leaded/Leadless Chip Carrier)	49
SOJ (Small-Outline J-Lead)	52
SOP (Small Outline Package)	54
TSOP (Thin Small Outline Package)	54
SIMM (Single In-Line Leadless Memory Module)	59
PCMCIA Flash Memory Cards	62
Flash Drives	66
Update Options	68
Off-Board PROM Programming	68
On-Board Update	69
In-System Write	71
Summary	72
Chapter Five: Hardware Interfacing to Flash Memory Components	73
Hardware Interfacing Fundamentals	73
Chip Enable	75
Addresses	75
Data In/Out	76
Output Enable	76
Write Enable	77

$\overline{\text{WE}}$ -Less Flash Memories	78
The V_{PP} Program/Erase Voltage	79
Switching V_{PP}	80
V_{PP} Feedback	81
Advanced Hardware Interfacing	82
The $\overline{\text{PWD}}$ Input	82
RY/ $\overline{\text{BY}}$ Output	84
Interpreting Datasheet AC Parameters	85
General Observations	90
Naming Conventions	91
Capacitive Loading and Effects	92
AC Read Characteristics	93
Read Specification Clarifications	94
AC Write Characteristics	95
Write Specification Clarifications	96
Performance Enhancements	98
Caching	98
Shadowing	98
Hardware Interleaving	99
Summary	104
Chapter Six: Power Requirements and Design Techniques	105
The V_{CC} Operating Voltage	106
Read Mode (I_{CCR})	108
Standby Mode (I_{CCS})	109
Deep Powerdown Mode (I_{CCD})	110
Program Mode ($I_{\text{CCW}}/I_{\text{CCP}}$)	111
Erase Mode (I_{CCE})	112
The V_{PP} Program/Erase Voltage	113
Read/Standby Mode (I_{PPR} and I_{PPS})	114
Deep Powerdown Mode (I_{PPD})	114
Program Mode ($I_{\text{PPW}}/I_{\text{PPP}}$)	115
Erase Mode (I_{PPE})	115
V_{PP} Generation Techniques	117
Directly from a 12V Regulated Supply	117
Converting from 12V Unregulated	118
Converting from a Lower Voltage	118
Converting from a Higher Voltage	120

General Characteristics of Voltage Converters	121
Totally Modular Solutions	122
Bypass and Decoupling Capacitive Filtering	122
Decoupling Capacitors- V_{CC}	123
Bypass Capacitors- V_{CC}	124
Decoupling Capacitors- V_{PP}	124
Mixed-Voltage System Design	125
3.3 Volt to 5 Volt Interfaces	125
5V to 3.3V Interfaces	127
Bidirectional Bus Interface	130
Power Management Techniques	130
Summary	132
Chapter Seven: Software Interfacing to Flash Memory	133
Why Is Flash Memory Controlled By System Software?	134
EPROM Programming Algorithm	134
Flash Memory Programming	136
The NOR Bulk-Erase Flash Memory Algorithms	139
The Program Algorithm	140
The Chip Erase Algorithm	143
Summary of First-Generation Programming/Erase Characteristics	148
The NOR Fully-Automated Flash Memory Algorithms	149
Intel Automated Program Algorithm	151
Intel Automated Block Erase Algorithm	156
Intel Automated Erase Suspend/Resume Algorithm ..	160
Alternative Automated Algorithms	162
General Automated Algorithm Techniques-Multiple Block Erase	168
General Automated Algorithm Techniques-Page Programming	168
General Automated Algorithm Techniques-Aborting Internal Automation	168
General Automated Algorithm Techniques-The $RY/\overline{BY}$ Output	169
Software Polling or Hardware Interrupt: Which Should You Use?	169
Update Routines	170

Flexible Design Techniques.....	172
System Boot Code Contents	173
Software Interface to Flash Cards, SIMMs and Multi- Component Arrays	174
Parallel Program of Non-Automated Flash Memories	176
Parallel Erase of Non-Automated Flash Memories	179
Parallel Program/Erase of Automated Flash Memories	183
Summary	184
Chapter Eight: Hardware Interfacing Considerations for Flash Cards	187
A Flash Memory Array Within a Card	187
PCMCIA Flash Memory Cards.....	188
PCMCIA 1.0	188
PCMCIA 2.0	189
PCMCIA Signal Definitions	193
Host System Implementations.....	211
Implementing PCMCIA 2.0 Hardware	220
Proprietary or Commercial Interface Controllers	220
Supporting Hardware for PCMCIA-Interface Controllers.....	220
Accessing Flash Memory Cards with PCMCIA- Interface Controllers	222
More On Buffering	222
Summary	225
Chapter Nine: Flash Memory File Systems	227
Introduction	227
Flash Memory Solid-State Drive Form Factors	227
Flash Memory Solid-State Drives Require Special Drivers.....	228
Disk-Drive Basics	229
DOS Data Structures	231
Device Drivers	233
Flash File System Designs	238
Measuring Drive Usage	239
The Disk-Drive Emulators	239
Flash Optimized File Systems.....	240

The Disk-Drive Emulators.....	241
Primitive Flash File Systems	242
Full-Featured Disk-Drive Emulators	247
Flash Optimized FSSD's	252
Accessing the Flash-Optimized FSSD	253
Microsoft's Flash File System Design Criteria.....	253
Functional Description	256
Flash File System Evaluation	265
Performance - File Transfer Rate	265
Performance - Clean-Up Efficiency	267
Performance - Hot and Cold File Management.....	268
Reliability - Cycle Leveling	268
Reliability - Failure Recovery Modes	269
System Level Issues - File System Overhead.....	270
System Level Issues - Ease of Use	270
Summary.....	271
Chapter Ten: PCMCIA Software	273
Introduction	273
The Areas of Software Compatibility	274
The PCMCIA-ExCA Relationship	277
Flash File System Models	279
The Original Flash File System Model	279
What's Really Necessary?.....	282
Socket Services.....	282
Defining the Adapter Hardware	283
Accessing Socket Services	284
Installing Socket Services.....	286
The Socket Services Functions.....	287
Non-Specific Functions	289
Adapter Functions	292
Window Functions	304
Fields In The I/O Window Characteristics Table.....	317
Window Size	318
Detecting Card Insertion.....	347
Error Detection and Correction Functions	347
Socket Service Design Considerations and Benefits.....	348
The Card Information Structure	349

Accessing the Card Information Structure.....	350
The Device Information Tuple.....	356
Card Services.....	363
What is Card Services?	364
Do You Need Card Services?.....	366
Flash Card Memory Technology Drivers	366
Why Support New Cards?.....	367
Flash Card Driver Functions	367
Interfacing to the Flash Card Driver	368
Installing the Flash Card Drivers	368
Summary	371
Appendix A: Flash Memory Component Vendors	373
Appendix B: Flash Memory Card/Drive Vendors	375
Appendix C: Flash Memory Component and Card Programmers	379
Appendix D: Component and Card Socket and Adapter Vendors.....	383
Appendix E: 12V Converters	387
Appendix F: Flash Memory Card Readers and Writers.....	391
Appendix G: Flash File Systems.....	395
Appendix H: PCMCIA and Software Vendors	397
Appendix I: PCMCIA Compliance Testing Facilities	399
Appendix J: PCMCIA Card Types.....	401
Appendix K: PCMCIA Controller Register Functions and Vendors...	403
Appendix L: INT 21H Standard Disk-Related Functions.....	409
Appendix M: Sample Flash File System Benchmarking Code.....	411

Sector-Level Modification Method

To achieve an even higher degree of functionality, a slightly more advanced flash device driver that performs sector-level modification may be employed. Every time the user modifies a file, the flash memory devices containing the FAT, directory, and associated file sectors get copied into system RAM. In RAM, the file system modifies the contents, updating the FAT and directory and inserting the user's changes into the file. (This method would not have to perform the FAT bit-flipping used in the previous example, because the FAT is completely rewritten with every change). Afterwards, the file system's software erases the corresponding flash memory devices and writes the modified file structures and file contents back into flash memory. Figure 9.11 shows this sequence for a simple single-device FSSD.

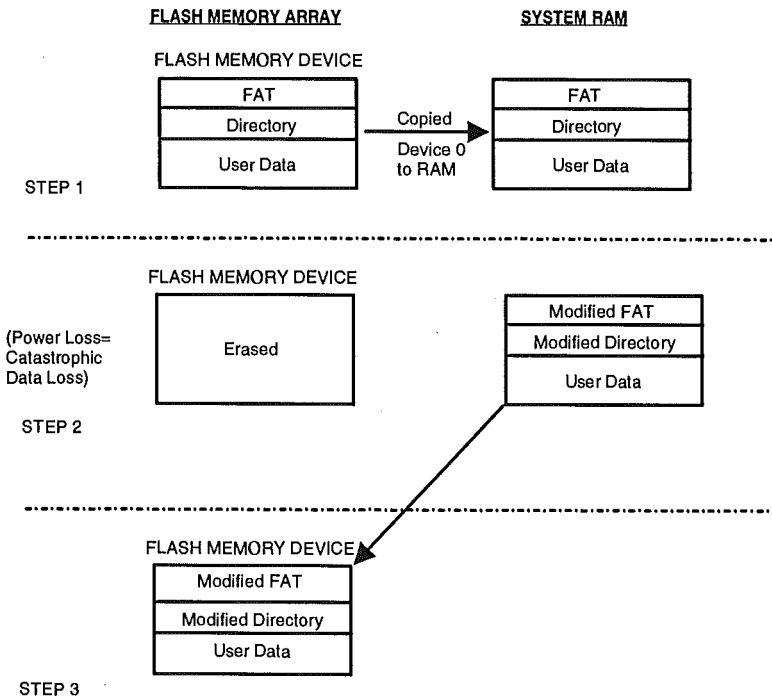


Figure 9.11: Sector-Level Modification Requires Considerable Overhead

If you're thinking, "Geez, this must be a long, slow process".....YOU'RE RIGHT! Remember that flash memory devices typically require one second to erase. A modification could constitute more than just a few erases for a file fragmented among many flash memory devices. Coming from a perspective that disk drives do not need to perform erases, these multiple one second erases become very visible in performance to the end user. Next you should ask, "What if power goes down during the erase process?". All the data temporarily stored in RAM goes away forever! However, analogous to the disk-image approach, if the user only requires occasional application changes, this flash file system may be adequate. Also realize that when performing excessive modifications, the flash memory devices in the FSSD experience an unequal amount of cycling, especially those containing the FAT and directory structures.

This type of flash device driver may function best using some of the smaller-sectored flash memory devices, such as Toshiba's NAND devices.

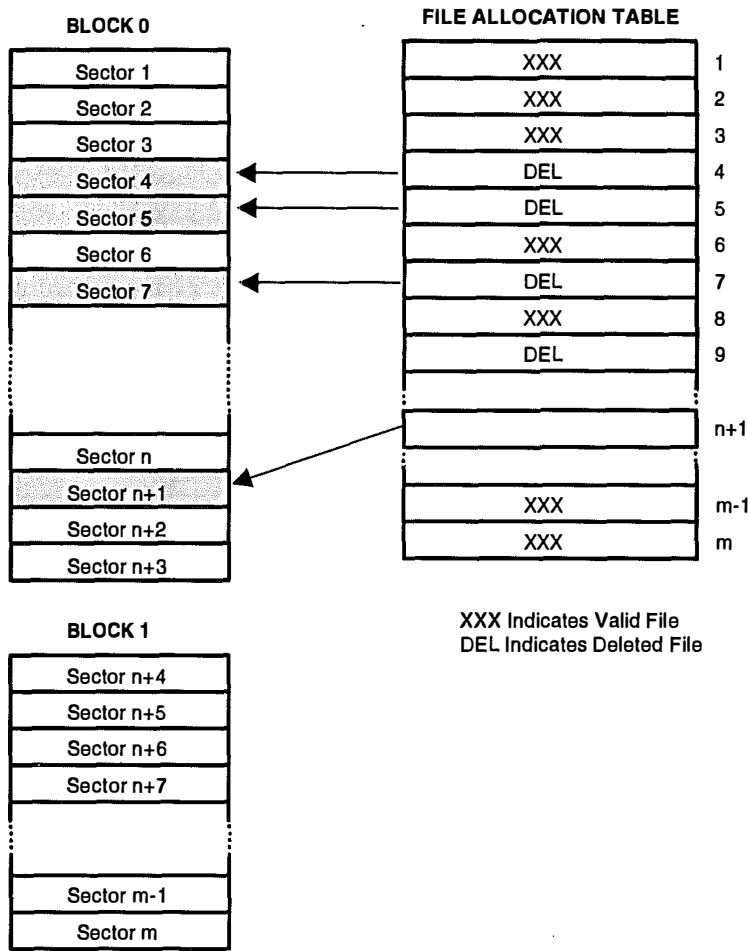
Full-Featured Disk-Drive Emulators

The example FSSDs we have seen up to this point have had very minimal functionality. Some of the noteworthy enhancements exhibited by real disk-drive emulators include:

- Virtually unlimited read and write capability
- Reclamation of memory space containing deleted files
- Compatible with all disk-related system commands

Today, several full-featured DDEs can be obtained from companies like M-Systems (True Flash File System, TFFS) and SCM. The file systems (e.g., FAT file system) supporting these drives access the DDE using all the standard mechanisms (disk interrupts and DOS function calls). When these file systems initialize, they install the disk interrupt filters we've discussed (just like the primitive flash file systems had to). Although these DDEs utilize a DOS FAT format and manage the media in terms of sectors, they differ considerably from the methods described earlier because they incorporate a logical (rather than physical) sector addressing scheme. This eliminates the dependency on any type of "fixed-in-flash" disk-drive data structures and allows the FAT and

directories to move around. This means that the FAT, directory and user files can be modified simply by relocating them to free flash memory.



- o Each block is an erasable flash unit.
- o Fat entries point to physical location of a sector within a specific flash memory block.
- o Full-featured DDEs can remove 'dirty' space occupied by deleted files.

Figure 9.12: One-to-One Correspondence Between FAT Entries and Sectors

You are welcome to send us comments or questions concerning this or other Annabooks products, or to request a catalog of our products and seminars.

Annabooks
11848 Bernardo Center Drive, Suite 110
San Diego, CA 92128

616-673-0870

1-800-462-1042

616-673-1432 FAX