

DOS 4.3 File Management System



Walland Philip Vrbancic, Jr.

DOS 4.3

File Management System

Walland Philip Vrbancic, Jr.

Copyright © 2020 January 1
Walland Philip Vrbancic, Jr.

All rights reserved.

No part of this publication may be reproduced,
stored in a retrieval system,
or transmitted, in any form or by any means,
electronic, mechanical, photocopying, recording, or otherwise,
without the prior written permission of the author.

DOS 4.3 File Management System
and this Publication are the
Confidential and Proprietary Intellectual Properties
of
Walland Philip Vrbancic, Jr.

ISBN 978-0-578-71547-6

*I am ever so proud to dedicate this Book on the
DOS 4.3 File Management System
and all my previous achievements
to my Parents Wally and Melba
who continuously nourished my intellectual curiosity.*

*I am ever so grateful to my partner
Carlton D. Wong
who delightfully pretends to understand
what the Hell I am talking about!*

If I have seen further than others it is because I have stood on the shoulders of giants.

~~~ Isaac Newton ~~~

# Disclaimer of All Liability

Do not use the DOS 4.3 File Management System or this Book for any mission-critical applications or for any purpose in which a software error or a software failure could cause you financial or material loss. The DOS 4.3 File Management System and this Book are designed to enhance your Apple ][ computing experience, but they may contain design flaws that could inhibit the proper operation of your computer or result in the loss of recorded data on any storage device connected to your computer. When using the DOS 4.3 File Management System or this Book, you assume all risks associated with the operation of your computer and the potential loss of your data. If these terms are not acceptable to you, please do not use the DOS 4.3 File Management System or this Book.

Walland Philip Vrbancic, Jr., the administrator of [applecored.net](http://applecored.net), makes no warranties either expressed or implied with respect to the DOS 4.3 File Management System or with respect to this Book, its quality, performance, or fitness for any particular purpose. Any risk of incidental or consequential damages resulting from the use of the DOS 4.3 File Management System or the use of information contained in this Book shall be assumed by you, the User. In no event will Walland Philip Vrbancic, Jr., or [applecored.net](http://applecored.net) be liable for any direct, indirect, incidental, or consequential damages resulting from any defect or deficiency in the DOS 4.3 File Management System or in this Book.

While all possible attempts have been made to ensure that the information contained within this publication is complete and accurate, the author shall have no liability or responsibility for any errors or omissions, or for any damages or data loss resulting from the use of the information, circuit diagrams, and example software programs contained herein. The author reserves the right to make any changes and/or any improvements to the DOS 4.3 File Management System or to the contents of this publication at any time and without any prior notice.

Apple and all Apple hardware and software brand names are trademarks of Apple Computer, Inc., registered in the United States and in other countries.  
All other brand names and trademarks are the property of their respective owners.

# Preface

When Brian Wiser and Bill Martens discovered my DOS 4.1 documentation and software at [applecored.net](http://applecored.net), they immediately contacted me and wanted Apple Pugetsound Program Library Exchange (A.P.P.L.E.) to publish my DOS 4.1 Manual. Ha! If only this would have happened back in 1982. That's when my co-worker, Randy at Rockwell, and I were actively reading many publications on Apple software and hardware, and Call-A.P.P.L.E. was one of our favorite publications. Needless to say, to be published by any of those computer journals at that time would have been crazy exciting, and certainly a cherished memory for a lifetime. I actually was very close to finishing DOS 4.1 when I agreed to have Call-A.P.P.L.E. publish the DOS 4.1 Manual, Build 45, and provide demo diskette images for DOS 4.1L and DOS 4.1H.

I wanted both versions of DOS 4.1 to provide the user with virtually the same computing experience, albeit the `HELP` command is found only in DOS 4.1H. This desire proved to be somewhat troublesome in that I was limited in memory for DOS 4.1L and I had ample memory for DOS 4.1H. It was the unused memory in DOS 4.1H that was the impetus to introduce the `HELP` command in the first place. At the onset I warned both Wiser and Martens that I could not stop creating more functionality in DOS 4.1, but they were rather insistent on printing the DOS 4.1 Manual for the Apple community as it was. I finished DOS 4.1 with Build 46. Only Build 46 can now be found at [applecored.net](http://applecored.net) as well as its respective PDF.

My next area of exploration for Apple DOS was an attempt to port DOS 4.1H to Auxiliary memory. I was absolutely successful, I might add, but I could not successfully design an interface between *Lisa* (my most favorite 65C02 assembler) and this DOS residing in Auxiliary memory. Over the course of several months in effort, I could not realize a viable solution that would be elegant, save memory, and provide the roadmap for interfacing other utilities and tools to this DOS. But this effort was certainly not wasted! I documented what I had learned about Main and Auxiliary memory management and moved forward to other areas of exploration.

I decided that I would use DOS 4.1H as my initial model for DOS 4.3. Yes, DOS 4.3 does retain the "H" designation for High memory. However, there is no DOS 4.3L. So, I simply refer to my new DOS as DOS 4.3. The question then became, can *Lisa* be ported and function in Auxiliary memory? The answer to that question turns out to be a resounding "Yes!" With DOS 4.3 in Main Language Card memory and *Lisa* in Auxiliary Language Card memory, the user has access to virtually all of Main memory below `0xBE00` for source code, object code, and the symbol list. I saw this configuration simply as an exercise of many potential and new possibilities.

Now, if I can relocate *Lisa* to Auxiliary memory, what about doing the same thing for *Big Mac*? I have to say that this challenge was a bit uneventful because relocating *Big Mac* to Auxiliary Language Card memory was even easier to accomplish. My main focus in *Big Mac* was to align *Sourceror* and *Big Mac* in terms of their *SWEET16* sourcing and assembling abilities, though I do not believe *Big Mac* has ever been able to assemble all of its own *SWEET16* opcodes. This task turned out to be an extraordinary undertaking: I wanted *Sourceror* and *Big Mac* to disassemble/assemble MY version of the *SWEET16* opcodes. I discovered that *Big Mac* could not even assemble its own unique *SWEET16* EVAL opcode. This tells me that Bredon probably did not even use *Big Mac* to assemble his own *Big Mac* source code. I have to confess that there still remain two *SWEET16* branches in my disassembled *Big Mac* source code that are wrong, and I do not know their solution to this day. They occur at memory addresses 0xD2C1 and 0xD2DF. Furthermore, I have yet to discover how to force their execution in order to analyze the resulting behavior in *Big Mac*. I suspect these particular instructions may be part of MACRO handling, something I have had no reason to use. *Big Mac* and DOS 4.3 now complement each other beautifully.

During my journey in developing the DOS 4.3 File Management System, I discovered many more layers of File Manager functionality that were almost coded correctly. It beleaguered me no end when I would issue a CLOSE statement on the Apple command line in DOS 3.3 or in DOS 4.1, and something would be flushed to the volume in focus. What was it? Why? I found even more examples of questionable logic, wrong logic, and desperate logic. I literally tore apart many of those “weird” routines used by the CLOSE statement so that now DOS commands will finish completely and data will be properly flushed. All these issues and many, many more have been resolved in DOS 4.3. The final frontier I tore apart was the RWTS manager and format algorithm. Using a utility of my own design to scan a track for its raw data and display the structure of that raw data allowed me to develop my own algorithm for a complete and revolutionary RWTS manager.

I know the user will discover many fascinating developments in the DOS 4.3 File Management System: he will be left wondering how he accomplished anything in a timely fashion without having had those developments in any other version of a previous Apple Disk Operating System. I would take that as my greatest compliment.

## Enjoy the ride!

# Table of Contents

|                                                 |           |
|-------------------------------------------------|-----------|
| <b>I. Designing Another New DOS.....</b>        | <b>1</b>  |
| 1. Introduction.....                            | 1         |
| 2. Brief Overview of DOS 4.3 .....              | 2         |
| 3. DOS 4.3 Software Development .....           | 3         |
| 4. Page-Zero Utilization.....                   | 5         |
| 5. DOS 4.3 VTOC Structure.....                  | 9         |
| 6. The VTOC Bitmap Definition .....             | 12        |
| 7. DOS 4.3 Catalog.....                         | 15        |
| 8. Booting DOS 4.3.....                         | 19        |
| 9. DOS 4.3 Initialization.....                  | 26        |
| 10. The DOS 4.3 RWTS Interface.....             | 31        |
| 11. The DOS 4.3 File Manager Interface.....     | 38        |
| 12. DOS 4.3 Data Structures.....                | 46        |
| 13. DOS 4.3 Clock Access.....                   | 51        |
| 14. DOS 4.3 Error Processing.....               | 52        |
| 15. DOS 4.3 Chain Command .....                 | 54        |
| 16. ProDOS Disk I/O Algorithm .....             | 58        |
| 17. Building and Installing DOS 4.3 Images..... | 60        |
| 18. Using DOS 4.3 Commands.....                 | 61        |
| <br>                                            |           |
| <b>II. Apple ][ ROM Modifications.....</b>      | <b>63</b> |
| 1. Correct HLIN Drawing Algorithm .....         | 64        |
| 2. Soft Switches in the Apple //e.....          | 67        |
| 3. SWEET16 Metaprocessor.....                   | 75        |
| 4. Applesoft Garbage Collector .....            | 80        |
| 5. Building a New Apple //e ROM .....           | 85        |
| 6. Apple //e Character Generator ROM.....       | 89        |
| 7. Peripheral Slot Card Signature Bytes .....   | 92        |
| <br>                                            |           |
| <b>III. DOS 4.3 Commands.....</b>               | <b>95</b> |
| 1. System Commands .....                        | 100       |
| 2. File System Commands .....                   | 113       |
| 3. Applesoft File Commands .....                | 132       |
| 4. Binary File Commands .....                   | 137       |
| 5. Sequential Text File Commands.....           | 141       |
| 6. Random-Access Data File Commands .....       | 151       |



|                                                       |            |
|-------------------------------------------------------|------------|
| <b>IV. DOS 4.3 Assembly Language Routines .....</b>   | <b>159</b> |
| 1. DOSWARM Routine.....                               | 160        |
| 2. DOSCOLD Routine.....                               | 160        |
| 3. CALLFM Routine.....                                | 161        |
| 4. CALLRWTS Routine.....                              | 161        |
| 5. GETFMCB Routine.....                               | 162        |
| 6. RDCLKVSN Vector.....                               | 162        |
| 7. GETIOCB Routine.....                               | 163        |
| 8. PRERRADR Vector.....                               | 163        |
| 9. HOOKDOS Routine.....                               | 163        |
| 10. XFERADR Address.....                              | 164        |
| 11. AUTOBRK Routine.....                              | 164        |
| 12. AUTORSET Vector.....                              | 165        |
| 13. PWRSTATE Variable.....                            | 165        |
| 14. USRAHAND Routine.....                             | 166        |
| 15. USRYHAND Routine.....                             | 166        |
| 16. NMASKIRQ Routine.....                             | 167        |
| 17. MASKIRQ Vector.....                               | 167        |
| 18. GOTOMON2 Routine.....                             | 168        |
| 19. GOTOMON1 Routine.....                             | 169        |
| 20. BLDVRSN Variable.....                             | 169        |
| 21. BLDNMBR Variable.....                             | 169        |
| 22. MNGDISK Vector.....                               | 170        |
| 23. MNGVALS Vector.....                               | 171        |
| 24. MNGUSER Vector.....                               | 172        |
| 25. INITDOS Vector.....                               | 173        |
| 26. INITVAL Address.....                              | 173        |
| 27. BCFGNDX Variable.....                             | 174        |
| 28. NBUF1PG Variable.....                             | 174        |
| 29. BOOTADR Variable.....                             | 175        |
| 30. BOOTPGS Variable.....                             | 175        |
| <br>                                                  |            |
| <b>V. DOS 4.3 Operational Environment.....</b>        | <b>177</b> |
| 1. Disk Window.....                                   | 178        |
| 2. EPROM Operating System (EOS) for quikLoader.....   | 183        |
| 3. VTOC Manager (VMGR).....                           | 199        |
| 4. Big Mac.....                                       | 202        |
| 5. CFFA Card.....                                     | 212        |
| 6. Volume Manager (VOLMGR).....                       | 220        |
| 7. File Developer (FID).....                          | 225        |
| 8. Lazer's Interactive Symbolic Assembler (Lisa)..... | 228        |
| 9. Program Global Editor (PGE).....                   | 242        |
| 10. Global Program Line Editor (GPL).....             | 246        |
| 11. Axlon RAM Disk 320.....                           | 248        |
| 12. RanaSystems EliteThree.....                       | 255        |
| 13. First Class Peripherals Sider.....                | 258        |

|                                            |     |
|--------------------------------------------|-----|
| 14. Sourceror.....                         | 264 |
| 15. Applesoft Formatter.....               | 267 |
| 16. Binary File Installation (BFI) .....   | 269 |
| 17. SCRG PROMGRAMER.....                   | 272 |
| 18. Real Time Clock.....                   | 274 |
| 19. JFD Parallel Printer Buffer .....      | 281 |
| 20. Asynchronous Data Transfer (ADT) ..... | 287 |
| 21. TrackScan .....                        | 290 |
| 22. Apple ][+ Memory Upgrade .....         | 295 |
| 23. Apple ][+ Keyboard Modification .....  | 299 |
| 24. Last Concluding Thoughts.....          | 302 |

**VI. Autobiographical Information.....307**

SAMPLE

# List of Figures

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| Figure I.4.1. Page-Zero Memory Utilization .....                             | 6   |
| Figure I.5.1. DOS 4.3 Data Disk Volume VTOC .....                            | 9   |
| Figure I.7.1. DOS 4.3 First Volume Catalog Sector .....                      | 14  |
| Figure I.7.2. DOS 4.3 TSL Sector .....                                       | 17  |
| Figure I.8.1. Attaching a Disk Controller Slot Card Handler in DOS 4.3 ..... | 20  |
| Figure I.8.2. Requesting and Detaching a Slot Card Handler in DOS 4.3 .....  | 21  |
| Figure I.8.3. Using MNGUSER in DOS 4.3 .....                                 | 22  |
| Figure I.8.4. Accessing and Changing INITVALS in DOS 4.3 .....               | 24  |
| Figure I.8.5. Changing the Boot Configuration Structure in DOS 4.3 .....     | 25  |
| Figure I.9.1. Generating RDNIBLBT in Disk ][ Firmware .....                  | 26  |
| Figure I.9.2. Generating RDNIBL and WRNIBL in DOS 4.3 .....                  | 26  |
| Figure I.9.3. Reading the DOS Version in DOS 4.3 .....                       | 29  |
| Figure I.9.4. Reading the Date and Time in DOS 4.3 .....                     | 29  |
| Figure I.9.5. Big Mac Printing a File Manager Error in DOS 4.3 .....         | 30  |
| Figure I.10.1. Disk ][ Carriage and Cam Table .....                          | 33  |
| Figure I.10.2. Four Half-Phase Track Separation .....                        | 33  |
| Figure I.10.3. Three Half-Phase Track Separation .....                       | 34  |
| Figure I.10.4. Inside View of a Stepper Motor .....                          | 35  |
| Figure I.11.1. Using the File Manager Context Block in DOS 4.3 .....         | 41  |
| Figure I.11.2. File Manager Command Parameter List .....                     | 45  |
| Figure I.12.1. Accessing and Changing CMDVALS in DOS 4.3 .....               | 46  |
| Figure I.12.2. Lisa Reading LOADLEN Value in DOS 4.3 .....                   | 50  |
| Figure I.15.1. Example Applesoft Program Layout in Memory .....              | 53  |
| Figure II.1.1. First HLIN Code Adjustment .....                              | 63  |
| Figure II.1.2. Second HLIN Code Adjustment .....                             | 64  |
| Figure II.1.3. Applesoft HLIN Demonstration Program .....                    | 64  |
| Figure II.1.4. Original ROM HLIN Routine Display .....                       | 65  |
| Figure II.1.5. Modified ROM HLIN Routine Display .....                       | 66  |
| Figure II.2.1. Notes for Tables II.2.1 to II.2.6 .....                       | 71  |
| Figure II.5.1. UNIX Command File to Build the APPLE2E.SW16GC.3.ROM .....     | 88  |
| Figure II.6.1. International XML File .....                                  | 89  |
| Figure II.6.2. Inverse of New Character Set Bitmap TIFF File .....           | 90  |
| Figure II.6.3. Icon TIFF Bitmap File .....                                   | 91  |
| Figure II.6.4. Binary Character Set LORES Editor .....                       | 91  |
| Figure II.7.1. Video Firmware Note for the Apple //e .....                   | 92  |
| Figure III.1.1. CONFIG Command Display .....                                 | 101 |
| Figure III.1.2. DATE Command for Thunderclock Card Display .....             | 103 |
| Figure III.1.3. HELP Command Display .....                                   | 103 |
| Figure III.1.4. HELP Content in HELP Command .....                           | 104 |
| Figure III.1.5. INIT Content in HELP Command .....                           | 104 |
| Figure III.1.6. MAXFILES Command Display .....                               | 106 |
| Figure III.1.7. MON Command Display .....                                    | 107 |

|                                                                         |     |
|-------------------------------------------------------------------------|-----|
| Figure III.1.8. NOMON Command Display .....                             | 107 |
| Figure III.1.9. PHASE Command Display .....                             | 109 |
| Figure III.1.10. PHASE 3 VTOC Using a TS Command Display .....          | 110 |
| Figure III.1.11. SV Command Display .....                               | 111 |
| Figure III.2.1. CATALOG and CAT Command Display .....                   | 114 |
| Figure III.2.2. LS R Command Display .....                              | 114 |
| Figure III.2.3. CD Command Display .....                                | 115 |
| Figure III.2.4. DELETE Command Display .....                            | 116 |
| Figure III.2.5. DIFF Command Display .....                              | 117 |
| Figure III.2.6. GREP Command Display .....                              | 118 |
| Figure III.2.7. INIT Command for Boot Disk .....                        | 119 |
| Figure III.2.8. INIT Command for Data Disk .....                        | 120 |
| Figure III.2.9. LIST Command Display .....                              | 122 |
| Figure III.2.10. LOCK File Command Display .....                        | 123 |
| Figure III.2.11. LOCK Volume Command Display .....                      | 124 |
| Figure III.2.12. RENAME Command Display .....                           | 125 |
| Figure III.2.13. TOUCH Command Display .....                            | 125 |
| Figure III.2.14. TS Command of an Empty Data Disk VTOC Display .....    | 127 |
| Figure III.2.15. UNLOCK File Command Display .....                      | 127 |
| Figure III.2.16. UNLOCK Volume Command Display .....                    | 129 |
| Figure III.2.17. URM Command Display .....                              | 129 |
| Figure III.2.18. VERIFY Command Display .....                           | 130 |
| Figure III.2.19. WTS Command Display .....                              | 131 |
| Figure III.3.1. Listing of START and PROGRAM2 CHAIN Display .....       | 133 |
| Figure III.3.2. Output of START and PROGRAM2 CHAIN Display .....        | 133 |
| Figure III.3.3. LOAD and SAVE Commands Display .....                    | 135 |
| Figure III.3.4. SAVE Commands Display .....                             | 135 |
| Figure III.4.1. BLOAD and BSAVE Commands Display .....                  | 137 |
| Figure III.4.2. BRUN Command Display .....                              | 138 |
| Figure III.4.3. LLOAD and LSAVE Commands Display .....                  | 139 |
| Figure III.5.1. OPEN, WRITE, and CLOSE Commands Display .....           | 142 |
| Figure III.5.2. APPEND Command Display .....                            | 143 |
| Figure III.5.3. EXEC Command Display .....                              | 144 |
| Figure III.5.4. No PROMPT, "EXEC,Rr" Command Display .....              | 145 |
| Figure III.5.5. POSITION and READ Commands Display .....                | 146 |
| Figure III.5.6. "READ,Bb" Command Display .....                         | 147 |
| Figure III.5.7. TLOAD and TSAVE Command Display .....                   | 149 |
| Figure III.5.8. TW Command Display .....                                | 149 |
| Figure III.6.1. OPEN, WRITE, and CLOSE Commands Display .....           | 152 |
| Figure III.6.2. Contents of RTEST.T Display .....                       | 153 |
| Figure III.6.3. READ and RUN Command Display .....                      | 154 |
| Figure III.6.4. Random-Access Data File CREATE .....                    | 155 |
| Figure IV.0.1. Direct and Indirect Approach for a Subroutine Call ..... | 159 |
| Figure V.1.1. Disk Window Startup Screen .....                          | 179 |
| Figure V.1.2. Select T/S Mode .....                                     | 179 |
| Figure V.1.3. Edit Data Screen .....                                    | 180 |
| Figure V.1.4. Write Sector Data Screen .....                            | 180 |
| Figure V.1.5. Print Sector Data Screen .....                            | 181 |
| Figure V.1.6. Disk Window Error Message Display .....                   | 181 |

|                |                                                       |     |
|----------------|-------------------------------------------------------|-----|
| Figure V.2.1   | quikLoader Schematic with Circuit Modifications ..... | 184 |
| Figure V.2.2.  | EOS Commands after RESET .....                        | 188 |
| Figure V.2.3.  | EOS Catalog for quikLoader EPROM 0, Part 1 .....      | 188 |
| Figure V.2.4.  | EOS Catalog for quikLoader EPROM 0, Part 2 .....      | 189 |
| Figure V.2.5.  | Example Code for QLBINEOS Utilization .....           | 194 |
| Figure V.2.6.  | EOS1 Image File Creation .....                        | 198 |
| Figure V.2.7.  | EOS2 Image File Creation .....                        | 198 |
| Figure V.3.1.  | VMGR Option Menu .....                                | 199 |
| Figure V.3.2.  | VTOC Contents for Option 2 and 3 .....                | 200 |
| Figure V.3.3.  | VTOC Sector Bitmap Contents for Option 4 and 5 .....  | 200 |
| Figure V.4.1.  | Big Mac Main Menu .....                               | 202 |
| Figure V.4.2.  | Big Mac Mnemonic Hashing Algorithm .....              | 204 |
| Figure V.6.1.  | VOLMGR Product Warning Screen .....                   | 220 |
| Figure V.6.2.  | VOLMGR Main Menu .....                                | 221 |
| Figure V.6.3.  | VOLMGR Manage Firmware Menu .....                     | 221 |
| Figure V.6.4.  | VOLMGR Manage CompactFlash Menu .....                 | 222 |
| Figure V.6.5.  | VOLMGR Device Identity Contents .....                 | 222 |
| Figure V.6.6.  | VOLMGR Manage Drives Menu .....                       | 223 |
| Figure V.6.7.  | VOLMGR Manage Volumes Menu .....                      | 223 |
| Figure V.6.8.  | VOLMGR Manage User DOS Images Menu .....              | 224 |
| Figure V.7.1.  | FID Main Menu .....                                   | 225 |
| Figure V.8.1.  | Lisa80 Startup Screen .....                           | 228 |
| Figure V.8.2.  | Lisa80 Source Code List Screen .....                  | 229 |
| Figure V.8.3.  | Lisa80 SETSCRN Routine .....                          | 230 |
| Figure V.8.4.  | Lisa80 READSCRN Routine .....                         | 231 |
| Figure V.8.5.  | Lisa80 SAVESCRN Routine .....                         | 231 |
| Figure V.8.6.  | Lisa80 SETUP80 Utility .....                          | 232 |
| Figure V.8.7.  | Lisa Screen Editing Definitions .....                 | 232 |
| Figure V.8.8.  | DOS 3.3 Source Code Volume .....                      | 235 |
| Figure V.8.9.  | EOS Image Segment Files .....                         | 235 |
| Figure V.8.10. | EOS1 Image Creation .....                             | 236 |
| Figure V.8.11. | EOS2 Image Creation .....                             | 236 |
| Figure V.9.1.  | Applesoft LIST Function .....                         | 243 |
| Figure V.9.2.  | PGE LIST Function .....                               | 243 |
| Figure V.9.3.  | PGE RENUMBER Function .....                           | 244 |
| Figure V.10.1. | GPLE FIND Function .....                              | 246 |
| Figure V.11.1. | Original RAM Card Hardware Circuit Diagram .....      | 250 |
| Figure V.11.2. | Modified RAM Card Hardware Circuit Diagram .....      | 251 |
| Figure V.11.3. | RAM Card Hardware Modifications .....                 | 252 |
| Figure V.11.4. | RAM Disk/RAM Card Connect Program .....               | 254 |
| Figure V.13.1. | Sider Connect Program .....                           | 262 |
| Figure V.13.2. | Sider Entry Points .....                              | 262 |
| Figure V.14.1. | Sourceror Initialization .....                        | 264 |
| Figure V.14.2. | Sourceror Startup/Help Screen .....                   | 265 |
| Figure V.14.3. | Sourceror Monitor Source Listing .....                | 265 |
| Figure V.15.1. | Applesoft Program Listing .....                       | 267 |
| Figure V.15.2. | Applesoft Program Programmatically Formatted .....    | 268 |
| Figure V.16.1. | BFI Main Menu .....                                   | 270 |
| Figure V.16.2. | BFI Peripheral Selection .....                        | 270 |

|                                                          |     |
|----------------------------------------------------------|-----|
| Figure V.16.3. BFI Installation Report on BFI.....       | 271 |
| Figure V.17.1. PROMGRAMER Configuration.....             | 272 |
| Figure V.17.2. PROMGRAMER Expanded Command Menu .....    | 273 |
| Figure V.18.1. Real Time Clock Circuit Diagram.....      | 275 |
| Figure V.20.1. ADT Window .....                          | 287 |
| Figure V.20.2. ADT Configuration.....                    | 288 |
| Figure V.20.3. ADT Software Credits .....                | 288 |
| Figure V.21.1. TrackScan Main Menu .....                 | 291 |
| Figure V.21.2. TrackScan for Track 0x00 .....            | 293 |
| Figure V.21.3. TrackScan for Track 0x11 .....            | 293 |
| Figure V.22.1. Apple ][+ Satellite Circuit Diagram ..... | 295 |
| Figure V.23.1. Apple ][+ Keyboard Modification.....      | 299 |

SAMPLE

# List of Tables

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
| Table I.3.1. Apple ][ Memory Utilization with DOS 4.3 .....                       | 4  |
| Table I.4.1. DOS 4.3 Page-Zero Utilization – Part 1 .....                         | 7  |
| Table I.4.2. DOS 4.3 Page-Zero Utilization – Part 2 .....                         | 8  |
| Table I.4.3. Available Page-Zero Locations Summary .....                          | 8  |
| Table I.5.1. DOS 4.3 VTOC Structure Block Definition .....                        | 10 |
| Table I.5.2. Free Sector Bitmap for Each Track Having 16 Sectors .....            | 10 |
| Table I.5.3. DOS 4.3 Date and Time Definition in Variable Order .....             | 11 |
| Table I.6.1. Free Sector Bitmap for Each Track Having 32 Sectors in DOS 3.3 ..... | 13 |
| Table I.7.1. DOS 4.3 Volume Catalog Entry .....                                   | 15 |
| Table I.7.2. DOS 4.3 Catalog Sector Data Offsets for File Entries .....           | 15 |
| Table I.7.3. DOS 4.3 File Type Byte Description .....                             | 16 |
| Table I.7.4. DOS 4.3 TSL Structure Block Definition .....                         | 17 |
| Table I.8.1. DOS 4.3 Boot and Data Management Structure Definition .....          | 18 |
| Table I.8.2. DOS 4.3 Boot Configuration Structure .....                           | 18 |
| Table I.8.3. DOS 4.3 Disk Track/Sector Mapping to Memory Address .....            | 20 |
| Table I.8.4. DOS 4.3 File Image Mapping to Memory Address .....                   | 22 |
| Table I.8.5. DOS 4.3 INITVALS Structure Definition .....                          | 23 |
| Table I.9.1. DOS 4.3 Page 0x03 Interface Routines and Vectors .....               | 28 |
| Table I.10.1. RWTS Input/Output Context Block Definition .....                    | 31 |
| Table I.10.2. RWTS Command Codes .....                                            | 31 |
| Table I.10.3. RWTS Error Codes .....                                              | 32 |
| Table I.11.1. File Manager Context Block Definition .....                         | 38 |
| Table I.11.2. File Manager Command Codes .....                                    | 39 |
| Table I.11.3. File Manager Read and Write Command Subcodes .....                  | 39 |
| Table I.11.4. File Manager Subcode Utilization .....                              | 40 |
| Table I.11.5. File Manager FMINITCD Boot Type (SUBCODE) .....                     | 40 |
| Table I.11.6. File Manager VTOCVALS Structure Initialization Data .....           | 42 |
| Table I.11.7. DOS 4.3 Error Messages and Sources .....                            | 43 |
| Table I.12.1. CMDVALS Data Structure Definition .....                             | 47 |
| Table I.12.2. File Manager Workarea Data Structure Definition .....               | 48 |
| Table I.12.3. File Manager File Buffer Definition .....                           | 49 |
| Table I.13.1. Supported Clock Cards in DOS 4.3 .....                              | 51 |
| Table I.15.1. Applesoft Simple Variable Descriptor Definitions .....              | 55 |
| Table I.15.2. Applesoft Array Variable Descriptor Definitions .....               | 55 |
| Table I.16.1. Comparison of DOS 4.3 and ProDOS RWTS .....                         | 57 |
| Table II.2.1. New Memory Management and Video Soft Switches .....                 | 68 |
| Table II.2.2. New Soft Switch Status Flags .....                                  | 69 |
| Table II.2.3. Original Input/Output Control Soft Switches, Part 1 .....           | 69 |
| Table II.2.4. Original Input/Output Control Soft Switches, Part 2 .....           | 70 |
| Table II.2.5. Original Memory Management Soft Switches .....                      | 70 |
| Table II.2.6. Original Disk ][ Control Soft Switches .....                        | 71 |
| Table II.2.7. Zip Chip Control Soft Switches .....                                | 72 |



|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| Table II.2.8. CFFA Control Soft Switches .....                                   | 72  |
| Table II.2.9. quikLoader Control Soft Switches .....                             | 73  |
| Table II.2.10. Sider, RAM Disk, RAM Card, and Rana Control Soft Switches .....   | 73  |
| Table II.3.1. SWEET16 Register Descriptions .....                                | 76  |
| Table II.3.2. SWEET16 Non-Register Opcodes .....                                 | 76  |
| Table II.3.3. SWEET16 Register Opcodes .....                                     | 77  |
| Table II.4.1. Simple Variable Descriptor Processing in Bongers' Pass 1 .....     | 81  |
| Table II.4.2. Array Variable Element Processing in Bongers' Pass 1 .....         | 81  |
| Table II.4.3. Garbage Collector Comparison and Verification Timing Results ..... | 84  |
| Table II.5.1. Disabled Applesoft Commands .....                                  | 85  |
| Table II.5.2. Transformations to Build a New Apple //e ROM .....                 | 86  |
| Table II.7.1. Peripheral Slot Card Signature Bytes .....                         | 93  |
| Table II.7.2. Revised Disk Drive Peripheral Slot Card Signature Bytes .....      | 94  |
| Table III.0.1. DOS 4.3 Command Valid Keyword Table .....                         | 95  |
| Table III.0.2. DOS 4.3 Command Table .....                                       | 96  |
| Table III.0.3. DOS 4.3 Keyword Name and Range Table .....                        | 97  |
| Table III.0.4. DOS 4.3 Keywords and Keyword Value Items .....                    | 98  |
| Table III.1.1. DOS 4.3 System Commands .....                                     | 100 |
| Table III.1.2. CONFIG Command Bit Summary .....                                  | 101 |
| Table III.1.3. DOS 4.3 File Buffer Memory Locations .....                        | 106 |
| Table III.2.1. DOS 4.3 File System Commands .....                                | 112 |
| Table III.2.2. Available Data Sectors for 36 Tracks, 16/32 Sectors/Track .....   | 121 |
| Table III.2.3. Total Sectors in Initialized Volumes .....                        | 121 |
| Table III.3.1. DOS 4.3 Applesoft File Commands .....                             | 132 |
| Table III.4.1. DOS 4.3 Binary File Commands .....                                | 136 |
| Table III.5.1. DOS 4.3 Sequential Text File Commands .....                       | 141 |
| Table III.6.1. DOS 4.3 Random-Access Data File Commands .....                    | 150 |
| Table IV.17.1. Interrupt Handler System Status Byte Definition .....             | 168 |
| Table V.2.1. quikLoader Bank Switching .....                                     | 185 |
| Table V.2.2. quikLoader Firmware Entry Points .....                              | 186 |
| Table V.2.3. quikLoader EPROM 0 Containing EOS and Programs .....                | 187 |
| Table V.2.4. EOS File Types .....                                                | 187 |
| Table V.2.5. EOS Catalog File Entry Structure .....                              | 192 |
| Table V.2.6. BINEOS Catalog File Entry .....                                     | 196 |
| Table V.4.1. Hash Table 1 – Assembly Directives .....                            | 205 |
| Table V.4.2. Hash Table 2 – Text Directives .....                                | 205 |
| Table V.4.3. Hash Table 3 – Macro Directives .....                               | 206 |
| Table V.4.4. Hash Table 4 – Single Byte Instructions .....                       | 207 |
| Table V.4.5. Hash Table 5 – Branch Instructions .....                            | 208 |
| Table V.4.6. Hash Table 6 – Single Byte SWEET16 Register Instructions .....      | 208 |
| Table V.4.7. Hash Table 7 – Multiple Addressing Mode Instructions .....          | 209 |
| Table V.4.8. Format Byte Definition .....                                        | 210 |
| Table V.5.1. CFFA Card Firmware Entry Points .....                               | 212 |
| Table V.5.2. CompactFlash Card CHS Parameters .....                              | 213 |
| Table V.5.3. CFFA Firmware Interface Control Registers .....                     | 213 |
| Table V.5.4. CFFA Firmware DVTS Variable Range .....                             | 214 |
| Table V.5.5. CompactFlash Card Drive/Volume Parameters .....                     | 215 |
| Table V.5.6. Block Utilization for a 1.0 GB CompactFlash Card .....              | 215 |
| Table V.5.7. CFFA Image Mapper for DOS 4.3 .....                                 | 216 |

|                                                                         |     |
|-------------------------------------------------------------------------|-----|
| Table V.5.8. DOS 3.3 Patches for CFFA.....                              | 218 |
| Table V.8.1. Lisa USR Command .....                                     | 233 |
| Table V.8.2. Lisa Command-Line Commands .....                           | 240 |
| Table V.11.1. RAM Card Memory Configuration Soft Switches .....         | 249 |
| Table V.11.2. RAM Disk 320 Firmware Entry Points .....                  | 253 |
| Table V.12.1. Rana Disk Firmware Entry Points .....                     | 256 |
| Table V.13.1. Sider Logical Block Structure.....                        | 259 |
| Table V.13.2. Modified Sider Logical Block Structure.....               | 259 |
| Table V.13.3. Sider Firmware Entry Points.....                          | 260 |
| Table V.18.1. SaRonix RTC58321 Real Time Clock Pinout .....             | 274 |
| Table V.18.2. Real Time Clock Peripheral Slot Card I/O Addresses.....   | 276 |
| Table V.18.3. Real Time Clock Configuration Register .....              | 276 |
| Table V.18.4. Interrupt Rate Selection .....                            | 277 |
| Table V.18.5. Real Time Clock Registers .....                           | 277 |
| Table V.18.6. Clock Firmware Entry Points.....                          | 279 |
| Table V.19.1. 8035 Microprocessor Memory Map .....                      | 284 |
| Table V.19.2. User Flag 1 .....                                         | 284 |
| Table V.19.3. Port Utilization.....                                     | 285 |
| Table V.19.4. SELRB0 Utilization.....                                   | 285 |
| Table V.19.5. SELRB1 Utilization.....                                   | 286 |
| Table V.19.6. Primary R3 R/W Block Number Bits .....                    | 286 |
| Table V.19.7. Secondary R3 System Flag Bits.....                        | 286 |
| Table V.22.1. Apple ][+ Satellite Circuit Board Connections.....        | 296 |
| Table V.22.2. Apple ][+ Satellite Circuit Board Operation, Part 1 ..... | 297 |
| Table V.22.3. Apple ][+ Satellite Circuit Board Operation, Part 2 ..... | 298 |
| Table V.23.1. 74LS153 Truth Table.....                                  | 300 |
| Table V.23.2. Generation of Unavailable Characters.....                 | 300 |

# I. Designing Another New DOS

This publication describes the process and products I created when I decided to design and program an enhanced Disk Operating System (DOS) for my Apple //e. Wherever I am able, I have included schematic diagrams, code samples, equations, figures, tables, and representative screen shots to help explain what I have created and the reasons why I did so. As in my previous design of an Apple ][ DOS, i.e. DOS 4.1L and DOS 4.1H, this has been an incredible journey for me. With DOS 4.3 I have again re-imagined that time when I mostly lived, breathed, and worked on Apple ][ computers, hardware, and software development continuously for a good period of my life many, many years ago.

## 1. Introduction

I have been an avid Apple ][ computer enthusiast, hobbyist, and professional software programmer since 1983 when I became the proud owner of an Apple ][+ computer. Besides the Apple ][+, my initial system included an Apple ][ Language Card, a Disk ][ with an Apple ][ Disk Controller slot card, an Amdek color monitor, and an Epson MX100 printer with a Grappler+ Printer Interface slot card. During those early years I designed and built my own Apple ][ peripheral slot cards, made electrical and hardware modifications to my Apple ][+ motherboard and keyboard, and wrote a substantial number of software programs using Applesoft BASIC (Applesoft hereafter) and 6502 assembly language. I soon acquired a Videx UltraTerm video display slot card and a Microsoft Z80 slot card. With the Z80 card I began writing Fortran programs that analyzed tomographic reconstructions of the human spinal column. A year or so later I added the Southern California Research Group quikLoader and PROMGRAMMER slot cards, a Johnathon Freeman Designs (JFD) Parallel Printer Buffer, and an Axlon RAM Disk 320 and its interface slot card to my system.

I used C language in my professional programming career for the software development of ultra-high-speed data collection systems for tactical radar and sensor development. Now that I am retired from the aerospace industry I have always wanted to dig into, tear apart, and learn the intricacies of the last available DOS for the Apple ][+. That DOS, DOS 3.3, was published on August 25, 1980. Then I recently came across another version of DOS 3.3 published years later on January 1, 1983. That DOS contains even more patches for the DOS APPEND command and a patch for Apple //e initialization. What I learned from the 1980 publication flabbergasted me: the software is exciting in its originality and concept vis-à-vis it was released just after the publication of Integer BASIC. However, I found the software to be somewhat juvenile in structure and implementation. Apparently, very little attention was given to software design and review. It appeared to me Apple made a strong push to release “something or anything” to consumers and vendors in order to begin marketing software products on diskettes. And history does reveal that Apple Computer did outsource DOS and contracted for it to be delivered within 35 days for \$13,000 in April, 1978. Paul Laughton at Shepardson Microsystems wrote Apple’s initial disk operating system using Hollerith cards, a card reader, and a minicomputer.

Now that I have the time and the continuing curiosity to delve into Apple ][ DOS, I have the unique opportunity to create my own version of DOS that contains the power and the flexibility I always thought DOS ought to and could have. I call this version of Apple ][ DOS, DOS 4.3 File Management System, and it requires an Apple ][ that contains Language Card memory. This document describes my eighth build of DOS 4.3. What a ride I have been on! Why? To see what I could do with this brilliant machine and its magnificent architecture!

## 2. Brief Overview of DOS 4.3

I know there are a great many ProDOS users in the Apple ][ community, but I never became interested in ProDOS. The work I did at Hughes Aircraft in the mid 1980's consisted of using assembly language for programming an operating system executive and interface driver routines on Gould SEL 2780, 6780, and 9780 mainframe computers. These computers hosted a proprietary operating system that allowed our team to simulate a radar processor traveling above the earth's surface in virtually real time. In order to accomplish that goal and simulate real time navigation, the computer's file system was essentially flat: each user had their own directory, and these user directories contained no subdirectories. I was very comfortable with the idea of a flat file system and it was very much like Apple's DOS 3.3. I was simply not comfortable with a slew of subdirectories exemplified by Apple's ProDOS. My thought was always "How does one remember the path to follow in order to find anything?" With the advent of the Macintosh computer and later when I became familiar with the UNIX file system, my subdirectory fears vanished and I cannot imagine a modern computer file system without subdirectories. However, I still remain passionate about Apple ][ DOS and I leave ProDOS to those who are comfortable with that operating system architecture. Though what I have seen of ProDOS recently, I believe it could definitely use a facelift, seriously. I also believe that ProDOS is better suited on a machine with a 16-bit processor much like that found in the Apple //gs.

I am sure many are curious and want to know what is new and different in DOS 4.3, and what makes this version of the DOS File Management System so special. Looking back over my previous build manuals for DOS 4.1, I realized that I should have included this vital build enhancement information with every build, if only for historical reasons. Like, which build did I solve the Track 0x00 utilization quest? Which build did I start labeling volumes? Which build did I solve the Disk Full logic error? Taken all together, I have done an incredible amount of research, writing, and software development to reach DOS 4.3, Build 8. And, to say the least, I have done an incredible amount of testing for every function under normal and abnormal (i.e. error) conditions. However small the list of items unique to DOS 4.3 may appear, I have spent countless hours developing and testing those items alone and in concert with the entire DOS 4.3 command repertoire.

DOS 4.3 is specifically designed to reside in Main Language Card memory. Language Card memory begins at address 0xD000, and it includes two banks of memory from 0xD000 to 0xDFFF and one bank of memory from 0xE000 to 0xFFFF. The user has complete freedom to use all memory below 0xBE00 where HIMEM is set. The foundation for DOS 4.3 was DOS 4.1H which introduced the HELP command. However, I have completely redesigned HELP in how it looks and how it works in DOS 4.3, though it provides the same information. I have added a companion command to the DOS TS command called WTS. WTS (Write Track/Sector) allows the user to modify a single byte at a time on any sector of a disk volume. And, DOS 4.3 introduces the DOS TOUCH command that will update any file's date and time stamp. Of course, if a file is locked nothing about the file can be changed nor can the file be deleted, unless there is an override available. DOS 4.3 provides that override for the DOS RENAME, TOUCH, and DELETE commands. Thus, DOS 4.3 introduces two additional File Manager opcodes to support WTS and TOUCH. DOS 4.3 introduces the DOS PHASE command which allows the user to set the number of Disk ][ stepper motor half-phases between adjacent tracks. Finally, DOS 4.3 introduces the CONFIG command which gives the user ultimate control over many DOS 4.3 display and input functions. Not that I allowed a single error to reside in any DOS 4.1 build, there are sometimes better programming methodologies. DOS 4.3 does contain many of the DOS 4.1 routines rewritten having a far better design in subverting virtually all possible DOS command programming consequences due to user naiveté. Let's begin with some software design strategies.

### 3. DOS 4.3 Software Development

In order to design reliable software for a particular machine or platform, one must understand the machine's complete architecture. I believe this design approach is fully applicable to the Apple ][ computer: either code or data occupies fixed addressable memory where some defined memory locations are reserved for the stack, text, graphics, control, and peripheral slot cards. Code is further restricted in the Apple ][ by the rather limited 6502-microprocessor Instruction Set. My obvious goal strategy is to design software in such a way as to create the most functionality with the least amount of code and data space. I believe this methodology will yield the highest degree of code effectiveness.

I use Gerard Putter's application Virtual ][, Version 9.3, to create my software applications, and that is the platform I use to perform the initial, though simulated testing. Once I am satisfied with a program or a utility operating within the Virtual ][ application, I transfer the volume image containing that program or utility to an Enhanced Apple //e. I have found some discrepancies between Virtual ][ and my Enhanced Apple //e particularly in enabling Language Card memory: two successive writes to memory address 0xC083 does **not** write enable Language Card Bank 2 memory in my Enhanced Apple //e as it does in Virtual ][. Two successive reads of memory address 0xC083 functions the same in both my Enhanced Apple //e and in Virtual ][ as they should. I have brought this to the attention of Mr. Putter. Also, Main memory is not initialized at power-up in quite the same way in my Enhanced Apple //e as it is in Virtual ][. I believe DOS 3.3 always assumed that an Apple ][ will power-up with all of page-zero memory set to 0xFF. Virtual ][ also makes this same assumption. I know I have been caught unaware that all of Auxiliary page-zero memory is not always set to 0xFF at power-up. Therefore, I have included a call to SETNORM during Boot Stage 2 to ensure that page-zero memory location 0x32 is, indeed, set to 0xFF. I have used AUXMOVE to manually "hide" some ProDOS code in Auxiliary memory within Virtual ][. The code disappears (is overwritten) when I boot with DOS 4.3. This does not happen in the Enhanced Apple //e: the code can still be safely found in Auxiliary memory after a reboot. Always, always, always make final tests on **real** hardware.

Before beginning any discussion of a complicated subject like a disk operating system or file management system for the Apple ][, it is usually easier to understand such a system if each component of that system is shown as part of a Big Picture. That Big Picture is shown in Table I.3.1. Though certainly not to any scale, Table I.3.1 shows how memory is utilized in the Apple ][ and where the basic hardware and software components are found in Main memory. I exclude any discussion of Auxiliary memory as found in the Apple //e at this time. The basic components shown in Table I.3.1 are the 6502 microprocessor memory requirements, the DOS vectors and routines, text and LORES graphic pages, HIRRES graphic pages, DOS file buffers, DOS software manager locations, Soft Switches, peripheral-card memory, Read/Write Track/Sector (RWTS) and HELP routines, Applesoft interpreter, and the ROM and RAM Monitor. The following pages will discuss the Apple ][ memory utilization in great detail. It may be helpful to refer to Table I.3.1 occasionally in order to fully understand how those details relate to the entire hardware and software management of the Apple ][ computer by the DOS 4.3 File Management System.

If any of the components shown in Table I.3.1 are unfamiliar, it would be to your advantage now to locate one or more Apple publications and refresh your understanding of that component. Even the *Apple ][ Reference Manual* that came with my Apple ][+ computer contains invaluable information applicable to the entire family of Apple ][ computers. I even own a few **SAMS** Publications that have provided me with enhanced understanding of many of the components shown in Table I.3.1.

| Memory Page | Description                                                                       | Description                   |
|-------------|-----------------------------------------------------------------------------------|-------------------------------|
| 0x00        | Page-zero variables, pointers, routines, and special addressing modes             |                               |
| 0x01        | Stack for the 6502 microprocessor                                                 |                               |
| 0x02        | Input buffer, Applesoft interpretation buffer                                     |                               |
| 0x03        | User buffer, DOS vectors and routines                                             |                               |
| 0x04–0x07   | Text or LORES graphics Page 1                                                     |                               |
| 0x08–0x0B   | Applesoft program start, Text or LORES graphics Page 2, or available for software |                               |
| 0x0C–0x1F   | Available for software                                                            |                               |
| 0x20–0x3F   | HIRES graphics Page 1, or available for software                                  |                               |
| 0x40–0x5F   | HIRES graphics Page 2, or available for software                                  |                               |
| 0x60–0xBD   | Available for software                                                            |                               |
| 0xBE–0xBF   | DOS 4.3 HIMEM, DOS 4.3 Language Card interface, DOS 4.3 bootstrap routines        |                               |
| 0xC0        | System Soft Switches                                                              |                               |
| 0xC1–0xC7   | Peripheral-card ROM memory for slots 1-7, or CX ROM                               |                               |
| 0xC8–0xCF   | Peripheral-card expansion ROM memory for slots 1-7, or CX ROM                     |                               |
| 0xD0–0xDF   | Bank 2, ROM Applesoft Interpreter, DOS 4.3 Command and File Managers              | Bank 1, DOS 4.3 RWTS and HELP |
| 0xE0–0xEB   | ROM Applesoft Interpreter, DOS 4.3 Command and File Managers                      |                               |
| 0xEC–0xEF   | ROM Applesoft Interpreter, DOS 4.3 working variables and file buffers             |                               |
| 0xF0–0xF7   | ROM Applesoft Interpreter, DOS 4.3 file buffers                                   |                               |
| 0xF8–0xFF   | ROM Monitor and RAM Monitor                                                       |                               |

Table I.3.1. Apple ][ Memory Utilization with DOS 4.3

The Apple ][ computer is truly a brilliant machine and it has a magnificent architecture. I hope you find my presentation of DOS 4.3 vis-à-vis the Apple ][ computer interesting, enlightening, and useful in view of your own hardware and software experiences with this delightful computer.

## 4. Page-Zero Utilization

The Instruction Set for the 6502-microprocessor (and the 65C02 processor as well) includes special processor instructions that utilize variables located in the first 256 bytes, or page, of addressable memory, that is, locations 0x0000 to 0x00FF. I designate this area of memory “page-zero.” When Steve Wozniak designed the Apple Monitor, he allocated a number of page-zero locations for its variables and pointers. Similarly, Applesoft, DOS, and virtually all other user assembly language programs use page-zero locations in order to utilize those special instructions. The 6502-microprocessor contains an accumulator, the A-register, and two index registers, the X-register and the Y-register. Page-zero instructions using these registers include load and store instructions, indexed load and store instructions, indexed indirect addressing instructions using the X-register, and indirect indexed addressing instructions using the Y-register. Page-zero wraparound occurs with page-zero indexed addressing and indexed indirect addressing instructions using the X-register and page-zero indexed addressing instructions of the X-register using the Y-register, but not with indirect indexed addressing instructions using the Y-register. Yes, it is a little confusing, but not too complicated.

When developing a user assembly language program, it is critical to select page-zero locations that do not conflict with the Apple Monitor, Applesoft, or DOS depending on whether those ROM and language card applications are important to the user program. Knowing which page-zero locations are used by or critical to resident applications can greatly simplify the selection of unused or available page-zero locations. Because DOS 3.3 supports Integer BASIC, a few page-zero locations were used to process that file type. DOS 4.3 also uses those same page-zero locations for processing the Applesoft CHAIN command, for example, and other DOS command enhancements. There are definitely obvious page-zero locations that cannot be used except for how they were intended, like the horizontal and vertical cursor locations CH and CV, respectively. Then, there are less obvious, rather dubious page-zero locations that are used by some Applesoft commands from 0x00 to 0x1F. These page-zero locations are fair game for user programs that do not use the Applesoft interpreter or Steve Wozniak’s *SWEET16* interpreter. Figure I.4.1 shows all the used and the unused page-zero locations and the applications that use those particular locations according to my references and the best of my ability to decipher the code that uses those locations. The shaded locations in Figure I.4.1 are unused page-zero locations that probably are not used by the Apple //e Monitor or Applesoft, so they are more than likely the better locations to select. Tables I.4.1 and I.4.2 lists all the page-zero locations utilized by DOS 4.3. Table I.4.3 summarizes all the available page-zero locations not utilized by the ROM routines and DOS 4.3. Keep in mind that indirect indexed addressing instructions using the Y-register do require a page-zero byte-pair, so it is even more critical that neither address byte is clobbered by software external to a user’s assembly language program.

There are certainly common page-zero locations that all software routines can use as temporary variables and pointers. The 6502-microprocessor is not time-shared and there is no context switching between routines, so if a routine uses some common page-zero locations, it should complete all processing using those locations and not expect to find its results sometime later. Examples of common page-zero locations would be A1L/A1H at 0x3C/0x3D, A2L/A2H at 0x3E/0x3F, A3L/A3H at 0x40/0x41, A4L/A4H at 0x42/0x43, OPRND at 0x44, and the first three bytes of DSCTMP at 0x9D:0x9F. Using these page-zero locations to move or copy data would be safe and not interfere with the Monitor, Applesoft, or DOS processing. Actually, several Monitor routines require that some of these locations just mentioned contain your data before using those routines. The Monitor routine MOVE at 0xFE2C is one such example. It is really up to the user to confirm and verify that the selected page-zero memory locations do not interfere with other routines external to and required by the user software.

| <b>0x</b> | <b>0</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>A</b>  | <b>B</b>  | <b>C</b>  | <b>D</b>  | <b>E</b>   | <b>F</b>   |
|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|------------|------------|
| <b>00</b> | 12<br>34 | 134      | 34       | 34       | 34       | 4        |          |          |          |          | 4         | 4         | 4         | 4         | 4          | 4          |
| <b>10</b> | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4         | 4         | 4         | 24        | 2          | 3          |
| <b>20</b> | 134      | 134      | 134      | 134      | 13<br>46 | 134      | 14<br>56 | 14<br>56 | 13<br>46 | 13<br>46 | 13<br>456 | 13<br>456 | 14<br>56  | 14<br>56  | 123<br>456 | 123<br>456 |
| <b>30</b> | 14       | 12       | 134      | 12<br>46 | 123      | 12<br>36 | 136      | 136      | 136      | 136      | 123       | 123       | 12<br>345 | 12<br>345 | 123<br>456 | 123<br>456 |
| <b>40</b> | 156      | 156      | 12<br>36 | 12<br>36 | 12<br>36 | 1        | 1        | 1        | 1        | 1        | 56        | 56        | 6         | 6         | 13         | 13         |
| <b>50</b> | 346      | 346      | 4        | 4        | 4        | 4        | 24       | 24       | 24       | 24       | 46        | 6         | 6         | 6         | 4          | 4          |
| <b>60</b> | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 46       | 46       | 46       | 46        | 46        | 46        | 346       | 346        | 346        |
| <b>70</b> | 346      | 4        | 4        | 346      | 346      | 4        | 46       | 4        | 4        | 4        | 4         | 4         | 4         | 4         | 4          | 4          |
| <b>80</b> | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4         | 4         | 4         | 4         | 4          | 4          |
| <b>90</b> | 4        | 4        | 4        | 4        | 34       | 34       | 4        | 4        | 4        | 4        | 4         | 34        | 34        | 4         | 4          | 4          |
| <b>A0</b> | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4         | 4         | 4         | 4         | 4          | 46         |
| <b>B0</b> | 46       | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4         | 4         | 4         | 4         | 4          | 4          |
| <b>C0</b> | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4        | 4         | 4         | 4         | 4         |            |            |
| <b>D0</b> | 4        | 4        | 4        | 4        | 4        | 4        | 46       |          | 46       | 6        | 4         | 4         | 4         | 4         | 4          | 4          |
| <b>E0</b> | 4        | 4        | 4        |          | 4        | 4        | 4        | 4        | 34       | 34       | 4         |           |           |           |            |            |
| <b>F0</b> | 4        | 4        | 4        | 14       | 14       | 4        | 4        | 4        | 4        | 4        |           |           |           |           |            | 34         |

Figure I.4.1. Page-Zero Memory Utilization

**Key**

1 – used by the Monitor

2 – used by the Mini Assembler

3 – used by the Apple //e CX ROM

4 – used by Applesoft

5 – used by RWTS

6 – used by DOS 4.3



| <b>Address</b> | <b>Parameter</b> | <b>Description</b>                          |
|----------------|------------------|---------------------------------------------|
| 0x24           | CH               | horizontal cursor location                  |
| 0x25           | CV               | vertical cursor location                    |
| 0x26           | BUFADRZ          | ROM boot data field buffer address          |
| 0x26           | TEMPZ            | RWTS temporary 8-bit variable               |
| 0x27           | TEMP2Z           | RWTS temporary 8-bit variable               |
| 0x28           | BASEZ            | text screen line address                    |
| 0x2A           | ASPTRSAV         | DOS CHAIN array descriptor addresses        |
| 0x2A           | CURTRKZ          | RWTS requested track                        |
| 0x2B           | DRVFLAG          | RWTS data-changing drive flag               |
| 0x2B           | SLOT16Z          | boot slot * 16                              |
| 0x2B           | SYNCNT           | RWTS format sync byte count                 |
| 0x2C           | DATAFNDZ         | RWTS address field address                  |
| 0x2D           | SECFNDZ          | RWTS address field sector found             |
| 0x2E           | TRKFNDZ          | RWTS address field track found              |
| 0x2F           | VOLFNDZ          | RWTS address field volume found             |
| 0x32           | INVFLG           | text screen inverse/normal flag             |
| 0x33           | PROMPT           | text screen prompt character                |
| 0x35           | PAGECNT          | boot/initialization DOS image page count    |
| 0x36           | CSWL             | output device handler address               |
| 0x38           | KSWL             | input device handler address                |
| 0x3C           | ROMTEMPZ         | ROM boot temporary 8-bit variable           |
| 0x3C           | MOTORTIM         | RWTS motor on-time 16-bit count             |
| 0x3C           | A1               | general purpose temporary 16-bit variable   |
| 0x3D           | ROMSECTR         | ROM boot requested sector                   |
| 0x3E           | BUFADR2Z         | RWTS data field buffer address              |
| 0x3E           | ODDBITSZ         | RWTS temporary 8-bit variable               |
| 0x3E           | A2               | general purpose temporary 16-bit variable   |
| 0x3F           | SECTORZ          | RWTS address field sector                   |
| 0x40           | ROMDATA          | ROM boot address field track found          |
| 0x40           | FILEBUFZ         | file context block parameter buffer address |
| 0x40           | TRACKZ           | RWTS address field track                    |
| 0x41           | ROMTRACK         | ROM boot requested track                    |
| 0x41           | VOLUMEZ          | RWTS address field volume                   |
| 0x42           | A4               | general purpose temporary 16-bit variable   |
| 0x42           | BUFADRZ          | general purpose sector data buffer address  |
| 0x44           | DIRINDX          | VTOC and TSL data index                     |
| 0x4A           | IOBADR           | RWTS IOCB buffer address                    |
| 0x4C           | DOSPTR           | DOS general purpose pointer address         |

Table I.4.1. DOS 4.3 Page-Zero Utilization – Part 1

|      |          |                                         |
|------|----------|-----------------------------------------|
| 0x50 | LINNUM   | Applesoft line number 16-bit variable   |
| 0x5A | DOSTEMP1 | DOS general purpose 8-bit variable      |
| 0x5B | DOSTEMP2 | DOS general purpose 8-bit variable      |
| 0x5C | DOSBUFR  | DOS general purpose buffer address      |
| 0x67 | ASPGMST  | Applesoft program start address         |
| 0x69 | ASVARS   | Applesoft simple variables pointer      |
| 0x6B | ASARYS   | Applesoft array pointer                 |
| 0x6D | ARYEND   | Applesoft end of array pointer          |
| 0x6F | ASSTRS   | Applesoft end of string storage pointer |
| 0x73 | ASHIMEM  | Applesoft HIMEM address                 |
| 0x76 | ASRUN    | Applesoft RUN flag                      |
| 0x9D | DSCTMP   | Applesoft temporary string descriptor   |
| 0xAF | ASPEND   | Applesoft end of program address        |
| 0xD6 | PROTECT  | Applesoft program write-protect flag    |
| 0xD8 | ASONERR  | Applesoft ONERR error flag              |
| 0xD9 | RKEYWORD | DOS R keyword 8-bit variable            |

Table I.4.2. DOS 4.3 Page-Zero Utilization – Part 2

| Start | End  | Description  |
|-------|------|--------------|
| 0x06  | 0x09 | 4 bytes free |
| 0x1E  | 0x1E | 1 byte free  |
| 0xCE  | 0xCF | 2 bytes free |
| 0xD7  | 0xD7 | 1 byte free  |
| 0xE3  | 0xE3 | 1 byte free  |
| 0xEB  | 0xEF | 5 bytes free |
| 0xFA  | 0xFE | 5 bytes free |

Table I.4.3. Available Page-Zero Locations Summary

```
DOS 4.3H
(c) 2020
Walland Philip Urbancic Jr

M=4308H P=04 T=DOS 4.3H Demo Disk
B=4308H boot L=0x2020 01/01/20 08:28:48
S=6 D=01 U=000 F=0510 01/01/20 08:28:48
001 0x12,0x0F HELLO
```

This publication describes the process and products I created when I decided to design and program an enhanced Disk Operating System (DOS) for my Apple //e. Wherever I am able, I have included code samples, schematic diagrams, equations, figures, tables, and representative screen shots to help explain what I have created and the reasons why I did so. As in my previous design of an Apple ][ DOS, i.e. DOS 4.1L and DOS 4.1H that I released in 2018, this has been an incredible journey for me. With DOS 4.3, I have again re-imagined that time when I mostly lived, breathed, and worked on Apple ][ hardware and software development continuously for a good period of my life many, many years ago. The list that follows are some of the features that I have engineered into the DOS 4.3 File Management System.

DOS 4.3 boots directly into Language Card memory and sets HIMEM to 0xBE00. All five file buffers are fully contained in Language Card memory.

DOS 4.3 supports the full integration of the *Lisa* and *Big Mac* assemblers in Auxiliary Memory. Both assemblers now support the full 65C02 instruction set. *Sourceror* and *Big Mac* both support all *SWEET16* opcodes. My ROM image supports the 65C02, *SWEET16*, and *GARBAGE* based on Bongers' algorithm.

DOS 4.3 supports most Clock Cards which play an integral role in Volume and File timestamps. Volumes may be titled, numbered, locked, and initialized to boot or to store data exclusively. Even track 0x00 is used for data. Files are loaded and saved using an acceleration algorithm.

DOS 4.3 supports Applesoft CHAIN, File DIFF, File GREP, File LIST, File Undelete, Sector Display, Sector Modify, Syntactical HELP, DOS Processing Control, Variable Management, Half-Phase Tracking, and full lowercase command and argument input.

DOS 4.3 copies files using a modified *FID* that accepts Volume and Phase Number parameters.

DOS 4.3 can read DOS 3.3 Volumes and files.

DOS 4.3 can initialize Volumes with up to 48 tracks having either 16 or 32 sectors per track.

All the source code and all the schematics can be requested through [www.applecored.net](http://www.applecored.net).

