

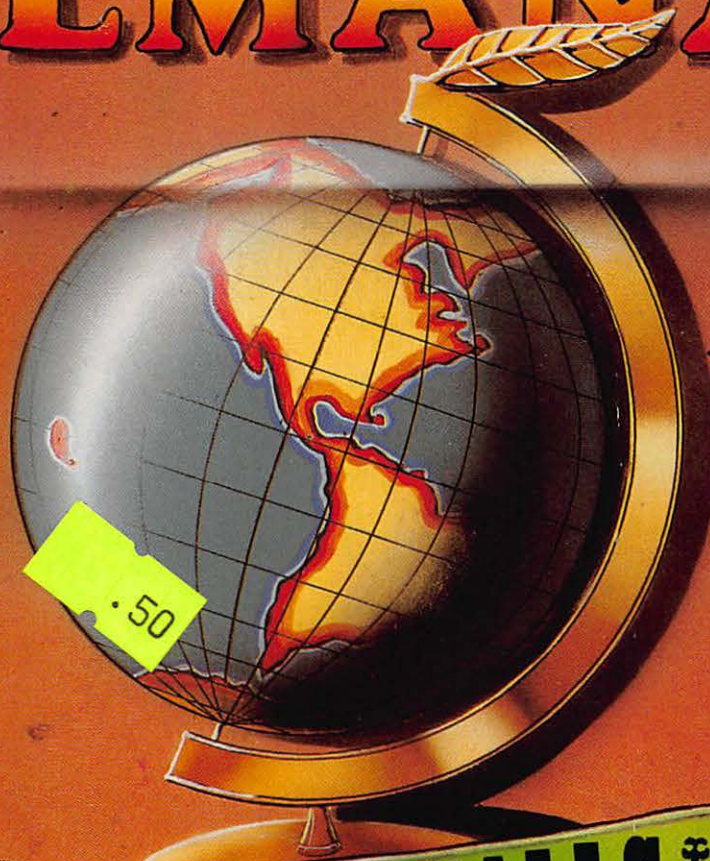


\$19.95

THE APPLE ALMANAC

1001

USEFUL THINGS



AMAZING
ASTOUNDING

FACTS



PEEKS & POKES & CALLS & ETC.

WRITTEN FOR YOU BY

ERIC E. GOEZ

&

WILLIAM B. SANDERS

The Apple Almanac

The Apple Almanac

By
Eric E. Goez
and
William B. Sanders, Ph.D.

Illustrated by
Martin Cannon



8943 Fullbright Ave.
Chatsworth, CA 91311-2750
(818) 709-1202



ISBN 0-88190-109-1

® COPYRIGHT 1983 BY DATAMOST INC.
ALL RIGHTS RESERVED

This manual is published and copyrighted by DATAMOST Inc. All rights are reserved by DATAMOST Inc. Copying, duplicating, selling or otherwise distributing this product is hereby expressly forbidden except by prior written consent of DATAMOST Inc.

The word APPLE and the Apple logo are registered trademarks of APPLE COMPUTER, INC.
APPLE COMPUTER, INC. was not in any way involved in the writing or other preparation of this manual, nor were the facts presented here reviewed for accuracy by that company. Use of the term APPLE should not be construed to represent any endorsement, official or otherwise, by APPLE COMPUTER, INC.

ACKNOWLEDGEMENTS

Our acknowledgements extend beyond the normal boundaries of most books since there are so many people who contributed in known and unknown ways. First, we owe a debt of gratitude to all the writers who took the time to write articles about the Apple Computer. They contributed to our overall understanding, and this book could not have been accomplished without their efforts. We are also grateful to CALL A.P.P.L.E, MICRO, NIBBLE, SOFTALK, APPLE ORCHARD, HARDCORE and other magazines which have published information and insights about the Apple provided by these writers.

The job of creating these charts from the ether was considerably aided by Neil Konzen's Program Line Editor, Peter Meyer's Routine Machine, Glen Bredon's Merlin assembler and Applesoft source listing, Don Worth's and Pieter Lechner's Beneath Apple DOS, Bert Kersey's APPLE MECHANIC, William E. Dougherty's APPLE II Monitor Peeled, Randy Hyde's DOS Source, Roger Wagner's Correspondent and Ed Zaron's Super-Text. Other utilities, articles, charts, discussions, books and programs too numerous to recall also helped. The SANDY APPLE PRESS, the Apple Corps of San Diego's club magazine, served as a training ground for formatting charts and learning about the Apple.

Individual Apple people who were directly influential or helpful include Roger Wagner, Val Golding, Bob Bishop, Brian Britt, Randy Hyde, Sam Janney and Bill Parker. The late Al Chaikin always supported us in endeavors with the Apple, and we are grateful to Al for all of his knowledge that he willingly shared. Finally, everybody in the Apple Corps of San Diego encouraged us in our project, and we are grateful to them all.

Marcia Carrozzo edited the manuscript so that humans, as well as computerists, could understand what we were attempting to communicate. Martin Cannon lightened the content considerably with his cartoons. Finally, Dave Gordon of DATAMOST provided us with all of the material needed to support this project. We are grateful to them all.

Our wives, Sandy and Eli, as well as our children Nerene and Heather and Billy and David, had to look at our backs as we hunched over a keyboard and gazed at monitors while absorbed with completing the book. Their understanding during this period is very much appreciated.

With all the gratitude for the help we received, we cannot share the blame for any of the book's shortcomings with others. Therefore, each of us will point out that errors, omissions or imperfections were solely the fault of the other.

TABLE OF CONTENTS

Alphabetic Table of Contents	9
Introduction	11
Part I — Groundwork	15
Hexadecimal to Decimal and Binary Chart	15
Decimal to Hexadecimal Conversion Chart	21
Hexadecimal Digit Addition Chart	26
Positive and Negative ASCII Character Codes	27
Expanded Positive/Negative ASCII Control Codes	29
Part II — BASIC	30
Applesoft/Integer BASIC & DOS Example Glossary	30
Applesoft BASIC List Interpreter Output Chart	44
Integer BASIC List Interpreter Chart	48
POKEs, PEEKs and CALLs	51
Sort Algorithms	55
Part III — DOS	59
RAM Disk Memory Cross Reference	59
DOS Buffer Charts	61
Track/Sector List Number Chart	77
DOS I/O Charts	78
Input Output Block Chart	78
Device Characteristics Table	79
File Manager Parameters List Chart	79
DOS Command Tables	81
DOS Command Entry Table	82
DOS Command Strings & Flags Table	83
DOS Keyword Charts	84
Keyword Name and Position Table	84
Keyword Value Min/Max Chart	85
DOS Error Message Charts	86
DOS Error Message Chart	86
DOS Error Index Chart	87
DOS VTOC Information	88
Volume Table of Contents (VTOC) Chart	88
Track Bit Map Chart	88
DOS File Information	98
Catalog/Directory Sectors Chart	98
File Name Entry Chart	100
File Classification Chart	104

Part IV — Text and Graphics 105

Text and Low Resolution Display Modes	105
Text and Low Resolution Screen Character Codes	105
Text and Low Resolution Memory Line Address Chart	109
Text and Low Resolution Tri-Line Charts	110
High Resolution Display Modes	112
High Resolution Horizontal Pixel Chart	112
High Resolution Line Chart	124
High Resolution Tri-Line Charts	129
High Resolution Screen Character Chart	134
High Resolution Shape Graphics	140
Decode High Resolution Shapes	140
Encode High Resolution Shapes	153
ASCII Character Shape Tables	175
ASCII Character Shape Table	175
Upper Case and Numbers	178
Upper/Lower Case and Numbers	179
ASCII Shape Table Generation Program	182

Part V — Assembly Language 185

Monitor Commands	185
Apple //e Monitor Comparison Chart	187
6502 Processor Status Flag Chart	188
6502 Opcodes : Numeric Order	190
6502 Opcodes : Alphabetic Order	194
Opcode Addressing Information	199
Branch Operand Offsets Chart	204
6502 Opcode Mnemonic Chart	208
R65C02 Added Mnemonics and Addressing Modes	209
R65C02 Opcodes	210
Numeric Listing	210
Alphabetic Listing	211
Conditioned Flags/Registers/Memory Chart	214
Applesoft BASIC Assembly Symbol Chart	219

ALPHABETIC TABLE OF CONTENTS

6502 Opcode Mnemonic Chart	208
6502 Opcodes - Alphabetic Order	211
6502 Opcodes - Numeric Order	210
6502 Processor Status Flag Chart	188
APPLE //e Monitor Comparison Chart	187
APPLESOFT BASIC Assembly Symbol Chart	219
APPLESOFT BASIC List Interpreter Output Chart	44
APPLESOFT/INTEGER BASIC & DOS Example Glossary	30
ASCII Character Shape Tables	175
ASCII Shape Table Generation Program	182
Branch Operand Offsets Chart	199
Catalog Directory Sectors Chart	98
Conditioned Flags/Registers/Memory Chart	214
Decimal to Hexadecimal Conversion Chart	21
Decode High Resolution Shapes	140
Device Characteristics Table	79
DOS Buffer Charts	61
DOS Command Entry Table	82
DOS Command Strings & Flags Table	83
DOS Error Index Chart	87
DOS Error Message Chart	86
Encode High Resolution Shapes	153
Expanded Positive-Negative ASCII Control Codes	29
File Classification Chart	104
File Manager Parameters List Chart	79
File Name Entry Chart	100
Hexadecimal Digit Addition Chart	26
Hexadecimal to Decimal and Binary Chart	15
High Resolution Horizontal Pixel Chart	112
High Resolution Line Chart	124
High Resolution Screen Character Chart	134
High Resolution Tri-Line Charts	129
Input Output Block Chart	78
Integer BASIC List Interpreter Chart	48
Keyword Name and Position Table	84
Keyword Value Min/Max Chart	85
Monitor Commands	185
Opcode Addressing Information	199

POKEs, PEEKs and CALLs	51
Positive and Negative ASCII Character Codes	27
R65C02 Added Mnemonics and Addressing Modes	209
R65C02 Opcodes	210
RAM/Disk Memory Cross Reference	60
Sort Algorithms	55
Text and Low Resolution Memory Line Address Chart	109
Text and Low Resolution Screen Character Codes	105
Text and Low Resolution Tri-Line Charts	110
Track Bit Map Chart	88
Track/Sector List Number Chart	77
Volume Table of Contents (VTOC) Chart	88

INTRODUCTION

This book began with the query, "Why doesn't somebody ...?" In our case, we were thinking that it would be a great idea to have a book with all the things we have to look up for the Apple in one place. Wouldn't it be nice to have a single source to find an ASCII chart, all our opcodes and a PEEK and POKE chart plus vital DOS information. While we're at it, why not have charts we use in graphics along with algorithms for doing various routines we always have to look up. Instead of having to remember where we put the 1979 copy of the club newsletter with the only chart of shape indexes, or whether it was CALL A.P.P.L.E. in 1981 or MICRO in 1980, (or was it VOL. 4 No. 2, 1983 NIBBLE?) where we saw that algorithm which finds the pointers for the beginning of an Integer BASIC program, we could pick up one book and know the information would be there.

Anyone who programs the Apple II, II+ or //e needs to look things up, and since we have been looking up things for years now, we have accumulated a good deal of knowledge of what programmers need. (Actually, we have a good deal of experience of what we need and are presumptuous enough to think others would like the same thing.) The problem was what to include in our book with "everything in one place" and how to best prepare the book to maximize its usefulness. In a very short period of time we found that an "everything" book for the Apple would weigh 203 pounds, 15 ounces and would cost more than the computer. That wouldn't do, but rather than curse the darkness, we decided to start with what we considered to be the most used and useful information and to proceed from there. Furthermore, we wanted to have information which would be applicable to a wide spectrum of programming problems and programmers. That meant we would have to consider both beginning and more advanced programmers. We were beginners once, and the charts we needed most then changed as we progressed, but we still go back to them on a number of occasions. (Who can remember the ASCII value for a percentage sign?) Also, we wanted to cover several different areas, including BASIC, DOS, graphics, assembly language and various general items. Finally, we wanted to explain what each chart did and how it could be used. Some of this is tutorial in that relative novices can read the explanation of a chart's use and then employ it to write a program. However, this book is not a general tutorial for beginners. If you are a beginner, you would be better off with THE ELEMENTARY APPLE, or some other beginner's book, and use this as a supplement. We have included a list of useful sources for learning about the various aspects of the Apple in case you have gaps in your knowledge about your computer.

To produce the charts, we toyed with the idea of looking up all the charts we knew of, writing the publishers for permission to use them and then cutting and pasting until we had the book we wanted. The problem with this idea was that we had some very firm ideas of how the charts should be organized and presented. Since the charts in the different sources had all sorts of formats, scope and styles, it would have looked like a crazy quilt instead of a book. The alternative was to generate all the charts ourselves, put the information into text files, and then do the final formatting and explanations with a word processor. The typesetter could then have a coherent body of manuscript rather than several scraps of paper. That was what we did. In this way, not only did we have the exact charts we wanted, we had them in the way we believed to be the most informative.

Since we could not get everything into this one volume, we decided to do later volumes until we had covered the full spectrum of information that "you have to look up." This is APPLE ALMA-NAC : Volume I, and in future volumes we intend to have more information of the same nature but covering those aspects of the Apple we did not have room enough to do here. Since the Apple

programmer is the ultimate judge of what is really useful and what is not, we invite you, the reader, to offer suggestions. Rather than grumbling about what we did *not* include in this volume, write to us and tell us what you would like to see in the next volume. In that way, instead of musing to yourself, "Why doesn't somebody...", you can take an active part in making what you want happen! Write to:

APPLE ALMANAC
DATAMOST, Inc.
8943 Fullbright Avenue
Chatsworth, CA 91311

CONVENTIONS

Most of the numbering and labeling conventions employed in this book are fairly standard and self-explanatory. However, some readers may not be familiar with them, so we will go over them here. (Throughout the book, there will be reminders of these conventions for convenience.)

DECIMAL, HEXADECIMAL and BINARY

We used both decimal and hexadecimal in most charts. This is because BASIC programmers access memory with decimal, while hexadecimal is used with machine/assembly language. Leading zeros (0's) are dropped from both hex and decimal numbers, as is the usual practice. Commas are used with four digit decimal numbers to make them easier to read, even though such commas are not used in writing programs. Where eight digit binary numbers are used, commas are placed in the center position with four bits on each side of the comma. (e.g., 0101,0101 or —+—, ++—). Hexadecimal numbers use no commas at all.

TRACK, SECTORS and BYTES

We use the TTSSBB notation for track, sectors and bytes. Track and byte values can be either one or two digits, while sector values (0-F) are always single hex digits. For example:

T T S B B
1 1 E 0 1

refers to Track \$11, Sector \$E, Byte \$01. Using only hex notations for TTSSBB is an exception to the convention of including both decimal and hexadecimal values. All references to disk storage is for 16 sector (DOS 3.3) formatted diskettes unless otherwise noted. In the text material, hexadecimal values are indicated with a leading dollar sign (e.g., \$12) to differentiate them from decimal numbers. In the charts, hexadecimal numbers and decimal numbers are differentiated by column heading.

LOOKING IT UP

The APPLE ALMANAC is organized into five parts. Each part contains several charts or other information pertaining to the general area described in the part heading. There is some overlap, and information you think is in one part may actually be in another. For example, a table for converting hexadecimal into decimal is in Part I: Groundwork. (If you are thinking of converting hexadecimal values into decimal for a BASIC program, you might look under Part II : BASIC.) To minimize the problem of finding things, we have made two Tables of Contents. The first is a standard one, listing the contents in chronological order. The second Table of Contents lists each chart, table or other information in alphabetical order. When you are using a certain set of charts for a project, it was found to be very helpful to use small, colored, plastic paper clips to mark the charts. In that way, it will be unnecessary to look up a currently used chart over and over again.

Overall, this book was written with the programmer in mind. What we included and failed to include were based on our collective experience with the Apple. It is important for us to know what you want and how you want it. So again, we invite you to write us to let us know and share your ideas.



HEXADECIMAL to DECIMAL and BINARY CHART

To use this chart for values of 0 through 255 (decimal) or \$0 through \$FF (hexadecimal), simply examine the left side (first 3 columns). When you have numbers larger than 255 decimal or \$FF hex, you must divide the number into "Low Byte" and "High Byte." Divide hexadecimal numbers into two parts made up of the rightmost two values and the left one or two numbers. For example, \$FC58 would be broken down into \$FC and \$58, with \$FC being the high byte and \$58 the low byte. Add two 0's to the high byte (left side of the number). For example, \$FC would now be \$FC00. On the right side (last three columns) find the value of the high byte; in this case \$FC00 equals 64,512 decimal. The low byte can be found on the right side. With our example, the value for \$58 is 88 decimal. Now, simply add the two together to find the decimal value of \$FC58.

$$\begin{array}{r} 64,512 \\ + 88 \\ \hline 64,600 \end{array}$$

With hex numbers having only 3 characters, remember to take the right two values *first*. For example, \$FC2 would be divided into \$F and \$C2; then two 0's would be added to the high byte, \$F, to get \$F00.

To change decimal to hexadecimal see DECIMAL to HEXADECIMAL CHART

LOW BYTE HEX (Increment by \$1)			HIGH BYTE HEX (Increment by \$100)		
HEX	DEC	BINARY	HEX	DEC	BINARY
0	0	0	0	0	0
1	1	1	100	256	1,0000,0000
2	2	10	200	512	10,0000,0000
3	3	11	300	768	11,0000,0000
4	4	100	400	1,024	100,0000,0000
5	5	101	500	1,280	101,0000,0000
6	6	110	600	1,536	110,0000,0000
7	7	111	700	1,792	111,0000,0000
8	8	1000	800	2,048	1000,0000,0000
9	9	1001	900	2,304	1001,0000,0000
A	10	1010	A00	2,560	1010,0000,0000
B	11	1011	B00	2,816	1011,0000,0000
C	12	1100	C00	3,072	1100,0000,0000
D	13	1101	D00	3,328	1101,0000,0000
E	14	1110	E00	3,584	1110,0000,0000
F	15	1111	F00	3,840	1111,0000,0000
10	16	1,0000	1000	4,096	1,0000,0000,0000
11	17	1,0001	1100	4,352	1,0001,0000,0000
12	18	1,0010	1200	4,608	1,0010,0000,0000
13	19	1,0011	1300	4,864	1,0011,0000,0000
14	20	1,0100	1400	5,120	1,0100,0000,0000
15	21	1,0101	1500	5,376	1,0101,0000,0000

HEX	DEC	BINARY	HEX	DEC	BINARY
16	22	1,0110	1600	5,632	1,0110,0000,0000
17	23	1,0111	1700	5,888	1,0111,0000,0000
18	24	1,1000	1800	6,144	1,1000,0000,0000
19	25	1,1001	1900	6,400	1,1001,0000,0000
1A	26	1,1010	1A00	6,656	1,1010,0000,0000
1B	27	1,1011	1B00	6,912	1,1011,0000,0000
1C	28	1,1100	1C00	7,168	1,1100,0000,0000
1D	29	1,1101	1D00	7,424	1,1101,0000,0000
1E	30	1,1110	1E00	7,680	1,1110,0000,0000
1F	31	1,1111	1F00	7,936	1,1111,0000,0000
20	32	10,0000	2000	8,192	10,0000,0000,0000
21	33	10,0001	2100	8,448	10,0001,0000,0000
22	34	10,0010	2200	8,704	10,0010,0000,0000
23	35	10,0011	2300	8,960	10,0011,0000,0000
24	36	10,0100	2400	9,216	10,0100,0000,0000
25	37	10,0101	2500	9,472	10,0101,0000,0000
26	38	10,0110	2600	9,728	10,0110,0000,0000
27	39	10,0111	2700	9,984	10,0111,0000,0000
28	40	10,1000	2800	10,240	10,1000,0000,0000
29	41	10,1001	2900	10,496	10,1001,0000,0000
2A	42	10,1010	2A00	10,752	10,1010,0000,0000
2B	43	10,1011	2B00	11,008	10,1011,0000,0000
2C	44	10,1100	2C00	11,264	10,1100,0000,0000
2D	45	10,1101	2D00	11,520	10,1101,0000,0000
2E	46	10,1110	2E00	11,776	10,1110,0000,0000
2F	47	10,1111	2F00	12,032	10,1111,0000,0000
30	48	11,0000	3000	12,288	11,0000,0000,0000
31	49	11,0001	3100	12,544	11,0001,0000,0000
32	50	11,0010	3200	12,800	11,0010,0000,0000
33	51	11,0011	3300	13,056	11,0011,0000,0000
34	52	11,0100	3400	13,312	11,0100,0000,0000
35	53	11,0101	3500	13,568	11,0101,0000,0000
36	54	11,0110	3600	13,824	11,0110,0000,0000
37	55	11,0111	3700	14,080	11,0111,0000,0000
38	56	11,1000	3800	14,336	11,1000,0000,0000
39	57	11,1001	3900	14,592	11,1001,0000,0000
3A	58	11,1010	3A00	14,848	11,1010,0000,0000
3B	59	11,1011	3B00	15,104	11,1011,0000,0000
3C	60	11,1100	3C00	15,360	11,1100,0000,0000
3D	61	11,1101	3D00	15,616	11,1101,0000,0000
3E	62	11,1110	3E00	15,872	11,1110,0000,0000
3F	63	11,1111	3F00	16,128	11,1111,0000,0000
40	64	100,0000	4000	16,384	100,0000,0000,0000
41	65	100,0001	4100	16,640	100,0001,0000,0000
42	66	100,0010	4200	16,896	100,0010,0000,0000
43	67	100,0011	4300	17,152	100,0011,0000,0000
44	68	100,0100	4400	17,408	100,0100,0000,0000
45	69	100,0101	4500	17,664	100,0101,0000,0000
46	70	100,0110	4600	17,920	100,0110,0000,0000

HEX	DEC	BINARY	HEX	DEC	BINARY
47	71	100,0111	4700	18,176	100,0111,0000,0000
48	72	100,1000	4800	18,432	100,1000,0000,0000
49	73	100,1001	4900	18,688	100,1001,0000,0000
4A	74	100,1010	4A00	18,944	100,1010,0000,0000
4B	75	100,1011	4B00	19,200	100,1011,0000,0000
4C	76	100,1100	4C00	19,456	100,1100,0000,0000
4D	77	100,1101	4D00	19,712	100,1101,0000,0000
4E	78	100,1110	4E00	19,968	100,1110,0000,0000
4F	79	100,1111	4F00	20,224	100,1111,0000,0000
50	80	101,0000	5000	20,480	101,0000,0000,0000
51	81	101,0001	5100	20,736	101,0001,0000,0000
52	82	101,0010	5200	20,992	101,0010,0000,0000
53	83	101,0011	5300	21,248	101,0011,0000,0000
54	84	101,0100	5400	21,504	101,0100,0000,0000
55	85	101,0101	5500	21,760	101,0101,0000,0000
56	86	101,0110	5600	22,016	101,0110,0000,0000
57	87	101,0111	5700	22,272	101,0111,0000,0000
58	88	101,1000	5800	22,528	101,1000,0000,0000
59	89	101,1001	5900	22,784	101,1001,0000,0000
5A	90	101,1010	5A00	23,040	101,1010,0000,0000
5B	91	101,1011	5B00	23,296	101,1011,0000,0000
5C	92	101,1100	5C00	23,552	101,1100,0000,0000
5D	93	101,1101	5D00	23,808	101,1101,0000,0000
5E	94	101,1110	5E00	24,064	101,1110,0000,0000
5F	95	101,1111	5F00	24,320	101,1111,0000,0000
60	96	110,0000	6000	24,576	110,0000,0000,0000
61	97	110,0001	6100	24,832	110,0001,0000,0000
62	98	110,0010	6200	25,088	110,0010,0000,0000
63	99	110,0011	6300	25,344	110,0011,0000,0000
64	100	110,0100	6400	25,600	110,0100,0000,0000
65	101	110,0101	6500	25,856	110,0101,0000,0000
66	102	110,0110	6600	26,112	110,0110,0000,0000
67	103	110,0111	6700	26,368	110,0111,0000,0000
68	104	110,1000	6800	26,624	110,1000,0000,0000
69	105	110,1001	6900	26,880	110,1001,0000,0000
6A	106	110,1010	6A00	27,136	110,1010,0000,0000
6B	107	110,1011	6B00	27,392	110,1011,0000,0000
6C	108	110,1100	6C00	27,648	110,1100,0000,0000
6D	109	110,1101	6D00	27,904	110,1101,0000,0000
6E	110	110,1110	6E00	28,160	110,1110,0000,0000
6F	111	110,1111	6F00	28,416	110,1111,0000,0000
70	112	111,0000	7000	28,672	111,0000,0000,0000
71	113	111,0001	7100	28,928	111,0001,0000,0000
72	114	111,0010	7200	29,184	111,0010,0000,0000
73	115	111,0011	7300	29,440	111,0011,0000,0000
74	116	111,0100	7400	29,696	111,0100,0000,0000
75	117	111,0101	7500	29,952	111,0101,0000,0000
76	118	111,0110	7600	30,208	111,0110,0000,0000
77	119	111,0111	7700	30,464	111,0111,0000,0000
78	120	111,1000	7800	30,720	111,1000,0000,0000

HEX	DEC	BINARY	HEX	DEC	BINARY
79	121	111,1001	7900	30,976	111,1001,0000,0000
7A	122	111,1010	7A00	31,232	111,1010,0000,0000
7B	123	111,1011	7B00	31,488	111,1011,0000,0000
7C	124	111,1100	7C00	31,744	111,1100,0000,0000
7D	125	111,1101	7D00	32,000	111,1101,0000,0000
7E	126	111,1110	7E00	32,256	111,1110,0000,0000
7F	127	111,1111	7F00	32,512	111,1111,0000,0000
80	128	1000,0000	8000	32,768	1000,0000,0000,0000
81	129	1000,0001	8100	33,024	1000,0001,0000,0000
82	130	1000,0010	8200	33,280	1000,0010,0000,0000
83	131	1000,0011	8300	33,536	1000,0011,0000,0000
84	132	1000,0100	8400	33,792	1000,0100,0000,0000
85	133	1000,0101	8500	34,048	1000,0101,0000,0000
86	134	1000,0110	8600	34,304	1000,0110,0000,0000
87	135	1000,0111	8700	34,560	1000,0111,0000,0000
88	136	1000,1000	8800	34,816	1000,1000,0000,0000
89	137	1000,1001	8900	35,072	1000,1001,0000,0000
8A	138	1000,1010	8A00	35,328	1000,1010,0000,0000
8B	139	1000,1011	8B00	35,584	1000,1011,0000,0000
8C	140	1000,1100	8C00	35,840	1000,1100,0000,0000
8D	141	1000,1101	8D00	36,096	1000,1101,0000,0000
8E	142	1000,1110	8E00	36,352	1000,1110,0000,0000
8F	143	1000,1111	8F00	36,608	1000,1111,0000,0000
90	144	1001,0000	9000	36,864	1001,0000,0000,0000
91	145	1001,0001	9100	37,120	1001,0001,0000,0000
92	146	1001,0010	9200	37,376	1001,0010,0000,0000
93	147	1001,0011	9300	37,632	1001,0011,0000,0000
94	148	1001,0100	9400	37,888	1001,0100,0000,0000
95	149	1001,0101	9500	38,144	1001,0101,0000,0000
96	150	1001,0110	9600	38,400	1001,0110,0000,0000
97	151	1001,0111	9700	38,656	1001,0111,0000,0000
98	152	1001,1000	9800	38,912	1001,1000,0000,0000
99	153	1001,1001	9900	39,168	1001,1001,0000,0000
9A	154	1001,1010	9A00	39,424	1001,1010,0000,0000
9B	155	1001,1011	9B00	39,680	1001,1011,0000,0000
9C	156	1001,1100	9C00	39,936	1001,1100,0000,0000
9D	157	1001,1101	9D00	40,192	1001,1101,0000,0000
9E	158	1001,1110	9E00	40,448	1001,1110,0000,0000
9F	159	1001,1111	9F00	40,704	1001,1111,0000,0000
A0	160	1010,0000	A000	40,960	1010,0000,0000,0000
A1	161	1010,0001	A100	41,216	1010,0001,0000,0000
A2	162	1010,0010	A200	41,472	1010,0010,0000,0000
A3	163	1010,0011	A300	41,728	1010,0011,0000,0000
A4	164	1010,0100	A400	41,984	1010,0100,0000,0000
A5	165	1010,0101	A500	42,240	1010,0101,0000,0000
A6	166	1010,0110	A600	42,496	1010,0110,0000,0000
A7	167	1010,0111	A700	42,752	1010,0111,0000,0000
A8	168	1010,1000	A800	43,008	1010,1000,0000,0000
A9	169	1010,1001	A900	43,264	1010,1001,0000,0000

HEX	DEC	BINARY	HEX	DEC	BINARY
AA	170	1010,1010	AA00	43,520	1010,1010,0000,0000
AB	171	1010,1011	AB00	43,776	1010,1011,0000,0000
AC	172	1010,1100	AC00	44,032	1010,1100,0000,0000
AD	173	1010,1101	AD00	44,288	1010,1101,0000,0000
AE	174	1010,1110	AE00	44,544	1010,1110,0000,0000
AF	175	1010,1111	AF00	44,800	1010,1111,0000,0000
B0	176	1011,0000	B000	45,056	1011,0000,0000,0000
B1	177	1011,0001	B100	45,312	1011,0001,0000,0000
B2	178	1011,0010	B200	45,568	1011,0010,0000,0000
B3	179	1011,0011	B300	45,824	1011,0011,0000,0000
B4	180	1011,0100	B400	46,080	1011,0100,0000,0000
B5	181	1011,0101	B500	46,336	1011,0101,0000,0000
B6	182	1011,0110	B600	46,592	1011,0110,0000,0000
B7	183	1011,0111	B700	46,848	1011,0111,0000,0000
B8	184	1011,1000	B800	47,104	1011,1000,0000,0000
B9	185	1011,1001	B900	47,360	1011,1001,0000,0000
BA	186	1011,1010	BA00	47,616	1011,1010,0000,0000
BB	187	1011,1011	BB00	47,872	1011,1011,0000,0000
BC	188	1011,1100	BC00	48,128	1011,1100,0000,0000
BD	189	1011,1101	BD00	48,384	1011,1101,0000,0000
BE	190	1011,1110	BE00	48,640	1011,1110,0000,0000
BF	191	1011,1111	BF00	48,896	1011,1111,0000,0000
C0	192	1100,0000	C000	49,152	1100,0000,0000,0000
C1	193	1100,0001	C100	49,408	1100,0001,0000,0000
C2	194	1100,0010	C200	49,664	1100,0010,0000,0000
C3	195	1100,0011	C300	49,920	1100,0011,0000,0000
C4	196	1100,0100	C400	50,176	1100,0100,0000,0000
C5	197	1100,0101	C500	50,432	1100,0101,0000,0000
C6	198	1100,0110	C600	50,688	1100,0110,0000,0000
C7	199	1100,0111	C700	50,944	1100,0111,0000,0000
C8	200	1100,1000	C800	51,200	1100,1000,0000,0000
C9	201	1100,1001	C900	51,456	1100,1001,0000,0000
CA	202	1100,1010	CA00	51,712	1100,1010,0000,0000
CB	203	1100,1011	CB00	51,968	1100,1011,0000,0000
CC	204	1100,1100	CC00	52,224	1100,1100,0000,0000
CD	205	1100,1101	CD00	52,480	1100,1101,0000,0000
CE	206	1100,1110	CE00	52,736	1100,1110,0000,0000
CF	207	1100,1111	CF00	52,992	1100,1111,0000,0000
D0	208	1101,0000	D000	53,248	1101,0000,0000,0000
D1	209	1101,0001	D100	53,504	1101,0001,0000,0000
D2	210	1101,0010	D200	53,760	1101,0010,0000,0000
D3	211	1101,0011	D300	54,016	1101,0011,0000,0000
D4	212	1101,0100	D400	54,272	1101,0100,0000,0000
D5	213	1101,0101	D500	54,528	1101,0101,0000,0000
D6	214	1101,0110	D600	54,784	1101,0110,0000,0000
D7	215	1101,0111	D700	55,040	1101,0111,0000,0000
D8	216	1101,1000	D800	55,296	1101,1000,0000,0000
D9	217	1101,1001	D900	55,552	1101,1001,0000,0000
DA	218	1101,1010	DA00	55,808	1101,1010,0000,0000

HEX	DEC	BINARY	HEX	DEC	BINARY
DB	219	1101,1011	DB00	56,064	1101,1011,0000,0000
DC	220	1101,1100	DC00	56,320	1101,1100,0000,0000
DD	221	1101,1101	DD00	56,576	1101,1101,0000,0000
DE	222	1101,1110	DE00	56,832	1101,1110,0000,0000
DF	223	1101,1111	DF00	57,088	1101,1111,0000,0000
E0	224	1110,0000	E000	57,344	1110,0000,0000,0000
E1	225	1110,0001	E100	57,600	1110,0001,0000,0000
E2	226	1110,0010	E200	57,856	1110,0010,0000,0000
E3	227	1110,0011	E300	58,112	1110,0011,0000,0000
E4	228	1110,0100	E400	58,368	1110,0100,0000,0000
E5	229	1110,0101	E500	58,624	1110,0101,0000,0000
E6	230	1110,0110	E600	58,880	1110,0110,0000,0000
E7	231	1110,0111	E700	59,136	1110,0111,0000,0000
E8	232	1110,1000	E800	59,392	1110,1000,0000,0000
E9	233	1110,1001	E900	59,648	1110,1001,0000,0000
EA	234	1110,1010	EA00	59,904	1110,1010,0000,0000
EB	235	1110,1011	EB00	60,160	1110,1011,0000,0000
EC	236	1110,1100	EC00	60,416	1110,1100,0000,0000
ED	237	1110,1101	ED00	60,672	1110,1101,0000,0000
EE	238	1110,1110	EE00	60,928	1110,1110,0000,0000
EF	239	1110,1111	EF00	61,184	1110,1111,0000,0000
F0	240	1111,0000	F000	61,440	1111,0000,0000,0000
F1	241	1111,0001	F100	61,696	1111,0001,0000,0000
F2	242	1111,0010	F200	61,952	1111,0010,0000,0000
F3	243	1111,0011	F300	62,208	1111,0011,0000,0000
F4	244	1111,0100	F400	62,464	1111,0100,0000,0000
F5	245	1111,0101	F500	62,720	1111,0101,0000,0000
F6	246	1111,0110	F600	62,976	1111,0110,0000,0000
F7	247	1111,0111	F700	63,232	1111,0111,0000,0000
F8	248	1111,1000	F800	63,488	1111,1000,0000,0000
F9	249	1111,1001	F900	63,744	1111,1001,0000,0000
FA	250	1111,1010	FA00	64,000	1111,1010,0000,0000
FB	251	1111,1011	FB00	64,256	1111,1011,0000,0000
FC	252	1111,1100	FC00	64,512	1111,1100,0000,0000
FD	253	1111,1101	FD00	64,768	1111,1101,0000,0000
FE	254	1111,1110	FE00	65,024	1111,1110,0000,0000
FF	255	1111,1111	FF00	65,280	1111,1111,0000,0000



DECIMAL to HEXADECIMAL CONVERSION CHART

To convert decimal numbers less than 1000 to hexadecimal, look up the decimal number in one of the five double columns on the left side of the chart. For example, the decimal number 622 can be found on the decimal side in the fourth double column. Its hex value is \$26E. To find values over 999, it is necessary to divide the number into two parts: thousands and what remains. For example, the decimal number 45,045 is divided into 45,000 and 45. First, find the hex value of 45. It is \$2D. Then in the far right double column, find the value of 45,000, which is \$AFC8. Finally add them together using the HEXADECIMAL DIGIT ADDITION CHART. The sum is \$AFF5. (See how this sum was reached on the hex addition chart.)

INC+1		INC+1		INC+1		INC+1		INC+1		INC+1,000	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
0	0	200	C8	400	190	600	258	800	320	1,000	3E8
1	1	201	C9	401	191	601	259	801	321	2,000	7D0
2	2	202	CA	402	192	602	25A	802	322	3,000	BB8
3	3	203	CB	403	193	603	25B	803	323	4,000	FA0
4	4	204	CC	404	194	604	25C	804	324	5,000	1388
5	5	205	CD	405	195	605	25D	805	325	6,000	1770
6	6	206	CE	406	196	606	25E	806	326	7,000	1B58
7	7	207	CF	407	197	607	25F	807	327	8,000	1F40
8	8	208	D0	408	198	608	260	808	328	9,000	2328
9	9	209	D1	409	199	609	261	809	329	10,000	2710
10	A	210	D2	410	19A	610	262	810	32A	11,000	2AF8
11	B	211	D3	411	19B	611	263	811	32B	12,000	2EE0
12	C	212	D4	412	19C	612	264	812	32C	13,000	32C8
13	D	213	D5	413	19D	613	265	813	32D	14,000	36B0
14	E	214	D6	414	19E	614	266	814	32E	15,000	3A98
15	F	215	D7	415	19F	615	267	815	32F	16,000	3E80
16	10	216	D8	416	1A0	616	268	816	330	17,000	4268
17	11	217	D9	417	1A1	617	269	817	331	18,000	4650
18	12	218	DA	418	1A2	618	26A	818	332	19,000	4A38
19	13	219	DB	419	1A3	619	26B	819	333	20,000	4E20
20	14	220	DC	420	1A4	620	26C	820	334	21,000	5208
21	15	221	DD	421	1A5	621	26D	821	335	22,000	55F0
22	16	222	DE	422	1A6	622	26E	822	336	23,000	59D8
23	17	223	DF	423	1A7	623	26F	823	337	24,000	5DC0
24	18	224	E0	424	1A8	624	270	824	338	25,000	61A8
25	19	225	E1	425	1A9	625	271	825	339	26,000	6590
26	1A	226	E2	426	1AA	626	272	826	33A	27,000	6978
27	1B	227	E3	427	1AB	627	273	827	33B	28,000	6D60
28	1C	228	E4	428	1AC	628	274	828	33C	29,000	7148
29	1D	229	E5	429	1AD	629	275	829	33D	30,000	7530
30	1E	230	E6	430	1AE	630	276	830	33E	31,000	7918
31	1F	231	E7	431	1AF	631	277	831	33F	32,000	7D00
32	20	232	E8	432	1B0	632	278	832	340	33,000	80E8
33	21	233	E9	433	1B1	633	279	833	341	34,000	84D0
34	22	234	EA	434	1B2	634	27A	834	342	35,000	88B8

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
35	23	235	EB	435	1B3	635	27B	835	343	36,000	8CA0
36	24	236	EC	436	1B4	636	27C	836	344	37,000	9088
37	25	237	ED	437	1B5	637	27D	837	345	38,000	9470
38	26	238	EE	438	1B6	638	27E	838	346	39,000	9858
39	27	239	EF	439	1B7	639	27F	839	347	40,000	9C40
40	28	240	F0	440	1B8	640	280	840	348	41,000	A028
41	29	241	F1	441	1B9	641	281	841	349	42,000	A410
42	2A	242	F2	442	1BA	642	282	842	34A	43,000	A7F8
43	2B	243	F3	443	1BB	643	283	843	34B	44,000	ABE0
44	2C	244	F4	444	1BC	644	284	844	34C	45,000	AFC8
45	2D	245	F5	445	1BD	645	285	845	34D	46,000	B3B0
46	2E	246	F6	446	1BE	646	286	846	34E	47,000	B798
47	2F	247	F7	447	1BF	647	287	847	34F	48,000	BB80
48	30	248	F8	448	1C0	648	288	848	350	49,000	BF68
49	31	249	F9	449	1C1	649	289	849	351	50,000	C350
50	32	250	FA	450	1C2	650	28A	850	352	51,000	C738
51	33	251	FB	451	1C3	651	28B	851	353	52,000	CB20
52	34	252	FC	452	1C4	652	28C	852	354	53,000	CF08
53	35	253	FD	453	1C5	653	28D	853	355	54,000	D2F0
54	36	254	FE	454	1C6	654	28E	854	356	55,000	D6D8
55	37	255	FF	455	1C7	655	28F	855	357	56,000	DAC0
56	38	256	100	456	1C8	656	290	856	358	57,000	DEA8
57	39	257	101	457	1C9	657	291	857	359	58,000	E290
58	3A	258	102	458	1CA	658	292	858	35A	59,000	E678
59	3B	259	103	459	1CB	659	293	859	35B	60,000	EA60
60	3C	260	104	460	1CC	660	294	860	35C	61,000	EE48
61	3D	261	105	461	1CD	661	295	861	35D	62,000	F230
62	3E	262	106	462	1CE	662	296	862	35E	63,000	F618
63	3F	263	107	463	1CF	663	297	863	35F	64,000	FA00
64	40	264	108	464	1D0	664	298	864	360	65,000	FDE8
65	41	265	109	465	1D1	665	299	865	361		
66	42	266	10A	466	1D2	666	29A	866	362		
67	43	267	10B	467	1D3	667	29B	867	363		
68	44	268	10C	468	1D4	668	29C	868	364		
69	45	269	10D	469	1D5	669	29D	869	365		
70	46	270	10E	470	1D6	670	29E	870	366		
71	47	271	10F	471	1D7	671	29F	871	367		
72	48	272	110	472	1D8	672	2A0	872	368		
73	49	273	111	473	1D9	673	2A1	873	369		
74	4A	274	112	474	1DA	674	2A2	874	36A		
75	4B	275	113	475	1DB	675	2A3	875	36B		
76	4C	276	114	476	1DC	676	2A4	876	36C		
77	4D	277	115	477	1DD	677	2A5	877	36D		
78	4E	278	116	478	1DE	678	2A6	878	36E		
79	4F	279	117	479	1DF	679	2A7	879	36F		
80	50	280	118	480	1E0	680	2A8	880	370		
81	51	281	119	481	1E1	681	2A9	881	371		
82	52	282	11A	482	1E2	682	2AA	882	372		
83	53	283	11B	483	1E3	683	2AB	883	373		
84	54	284	11C	484	1E4	684	2AC	884	374		

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
85	55	285	11D	485	1E5	685	2AD	885	375
86	56	286	11E	486	1E6	686	2AE	886	376
87	57	287	11F	487	1E7	687	2AF	887	377
88	58	288	120	488	1E8	688	2B0	888	378
89	59	289	121	489	1E9	689	2B1	889	379
90	5A	290	122	490	1EA	690	2B2	890	37A
91	5B	291	123	491	1EB	691	2B3	891	37B
92	5C	292	124	492	1EC	692	2B4	892	37C
93	5D	293	125	493	1ED	693	2B5	893	37D
94	5E	294	126	494	1EE	694	2B6	894	37E
95	5F	295	127	495	1EF	695	2B7	895	37F
96	60	296	128	496	1F0	696	2B8	896	380
97	61	297	129	497	1F1	697	2B9	897	381
98	62	298	12A	498	1F2	698	2BA	898	382
99	63	299	12B	499	1F3	699	2BB	899	383
100	64	300	12C	500	1F4	700	2BC	900	384
101	65	301	12D	501	1F5	701	2BD	901	385
102	66	302	12E	502	1F6	702	2BE	902	386
103	67	303	12F	503	1F7	703	2BF	903	387
104	68	304	130	504	1F8	704	2C0	904	388
105	69	305	131	505	1F9	705	2C1	905	389
106	6A	306	132	506	1FA	706	2C2	906	38A
107	6B	307	133	507	1FB	707	2C3	907	38B
108	6C	308	134	508	1FC	708	2C4	908	38C
109	6D	309	135	509	1FD	709	2C5	909	38D
110	6E	310	136	510	1FE	710	2C6	910	38E
111	6F	311	137	511	1FF	711	2C7	911	38F
112	70	312	138	512	200	712	2C8	912	390
113	71	313	139	513	201	713	2C9	913	391
114	72	314	13A	514	202	714	2CA	914	392
115	73	315	13B	515	203	715	2CB	915	393
116	74	316	13C	516	204	716	2CC	916	394
117	75	317	13D	517	205	717	2CD	917	395
118	76	318	13E	518	206	718	2CE	918	396
119	77	319	13F	519	207	719	2CF	919	397
120	78	320	140	520	208	720	2D0	920	398
121	79	321	141	521	209	721	2D1	921	399
122	7A	322	142	522	20A	722	2D2	922	39A
123	7B	323	143	523	20B	723	2D3	923	39B
124	7C	324	144	524	20C	724	2D4	924	39C
125	7D	325	145	525	20D	725	2D5	925	39D
126	7E	326	146	526	20E	726	2D6	926	39E
127	7F	327	147	527	20F	727	2D7	927	39F
128	80	328	148	528	210	728	2D8	928	3A0
129	81	329	149	529	211	729	2D9	929	3A1
130	82	330	14A	530	212	730	2DA	930	3A2
131	83	331	14B	531	213	731	2DB	931	3A3
132	84	332	14C	532	214	732	2DC	932	3A4
133	85	333	14D	533	215	733	2DD	933	3A5
134	86	334	14E	534	216	734	2DE	934	3A6

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
135	87	335	14F	535	217	735	2DF	935	3A7
136	88	336	150	536	218	736	2E0	936	3A8
137	89	337	151	537	219	737	2E1	937	3A9
138	8A	338	152	538	21A	738	2E2	938	3AA
139	8B	339	153	539	21B	739	2E3	939	3AB
140	8C	340	154	540	21C	740	2E4	940	3AC
141	8D	341	155	541	21D	741	2E5	941	3AD
142	8E	342	156	542	21E	742	2E6	942	3AE
143	8F	343	157	543	21F	743	2E7	943	3AF
144	90	344	158	544	220	744	2E8	944	3B0
145	91	345	159	545	221	745	2E9	945	3B1
146	92	346	15A	546	222	746	2EA	946	3B2
147	93	347	15B	547	223	747	2EB	947	3B3
148	94	348	15C	548	224	748	2EC	948	3B4
149	95	349	15D	549	225	749	2ED	949	3B5
150	96	350	15E	550	226	750	2EE	950	3B6
151	97	351	15F	551	227	751	2EF	951	3B7
152	98	352	160	552	228	752	2F0	952	3B8
153	99	353	161	553	229	753	2F1	953	3B9
154	9A	354	162	554	22A	754	2F2	954	3BA
155	9B	355	163	555	22B	755	2F3	955	3BB
156	9C	356	164	556	22C	756	2F4	956	3BC
157	9D	357	165	557	22D	757	2F5	957	3BD
158	9E	358	166	558	22E	758	2F6	958	3BE
159	9F	359	167	559	22F	759	2F7	959	3BF
160	A0	360	168	560	230	760	2F8	960	3C0
161	A1	361	169	561	231	761	2F9	961	3C1
162	A2	362	16A	562	232	762	2FA	962	3C2
163	A3	363	16B	563	233	763	2FB	963	3C3
164	A4	364	16C	564	234	764	2FC	964	3C4
165	A5	365	16D	565	235	765	2FD	965	3C5
166	A6	366	16E	566	236	766	2FE	966	3C6
167	A7	367	16F	567	237	767	2FF	967	3C7
168	A8	368	170	568	238	768	300	968	3C8
169	A9	369	171	569	239	769	301	969	3C9
170	AA	370	172	570	23A	770	302	970	3CA
171	AB	371	173	571	23B	771	303	971	3CB
172	AC	372	174	572	23C	772	304	972	3CC
173	AD	373	175	573	23D	773	305	973	3CD
174	AE	374	176	574	23E	774	306	974	3CE
175	AF	375	177	575	23F	775	307	975	3CF
176	B0	376	178	576	240	776	308	976	3D0
177	B1	377	179	577	241	777	309	977	3D1
178	B2	378	17A	578	242	778	30A	978	3D2
179	B3	379	17B	579	243	779	30B	979	3D3
180	B4	380	17C	580	244	780	30C	980	3D4
181	B5	381	17D	581	245	781	30D	981	3D5
182	B6	382	17E	582	246	782	30E	982	3D6
183	B7	383	17F	583	247	783	30F	983	3D7

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
184	B8	384	180	584	248	784	310	984	3D8
185	B9	385	181	585	249	785	311	985	3D9
186	BA	386	182	586	24A	786	312	986	3DA
187	BB	387	183	587	24B	787	313	987	3DB
188	BC	388	184	588	24C	788	314	988	3DC
189	BD	389	185	589	24D	789	315	989	3DD
190	BE	390	186	590	24E	790	316	990	3DE
191	BF	391	187	591	24F	791	317	991	3DF
192	C0	392	188	592	250	792	318	992	3E0
193	C1	393	189	593	251	793	319	993	3E1
194	C2	394	18A	594	252	794	31A	994	3E2
195	C3	395	18B	595	253	795	31B	995	3E3
196	C4	396	18C	596	254	796	31C	996	3E4
197	C5	397	18D	597	255	797	31D	997	3E5
198	C6	398	18E	598	256	798	31E	998	3E6
199	C7	399	18F	599	257	799	31F	999	3E7



HEXADECIMAL DIGIT ADDITION CHART

To add hexadecimal numbers, add the digits one at a time, carrying remainders as in normal math. To find the sum, simply find the row of the first number to be added and then the column of the second number to be added. Where the row and column intersect, you will find the sum. For example, we will add the sum of \$AFC8 and \$2D. First, as in decimal math, we begin with the rightmost digits, 8 and D. The intersection of Row 8 and Column D is "15." We write down the "5" and carry the "1", again as in decimal math. Now we find the sum of \$C + \$2 to be \$E. Adding the 1 we carried, we get \$F, and there is no carry. All that's left is to write down the \$AF to give us \$AFF5.

(1) carry the '1'

$$\begin{array}{r} \text{AFC8} \\ + 2\text{D} \\ \hline \text{AFF5} \end{array}$$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E



POSITIVE and NEGATIVE ASCII Character Codes

This chart has many uses. From BASIC, by POKEing in the decimal ASCII values to screen addresses 1024 to 2039, the character will appear at the screen location determined by the screen address. From machine language, the hex value can be loaded to the same range of screen addresses beginning at \$400. The chart can also be very useful for determining control codes for most printers and peripherals when information is transferred in non-showing ASCII codes, such as BS, FF, ESC and CR - ASCII characters in the range from 0 to 32 decimal. (Decimal 32/ hex \$20 is a "space.") The negative ASCII values represent the "High Bit On" condition, required by some word processors and programming situations. The "RUB" character is shown as an inverse square on Apple //e's and older Apples with lower case adaptors. Finally, ASCII character codes are important for the storage of string literals and data (e.g., text files).

POSITIVE ASCII VALUES			NEGATIVE ASCII VALUES			POSITIVE ASCII VALUES			NEGATIVE ASCII VALUES		
HEX	DEC	CHR	HEX	DEC		HEX	DEC	CHRS	HEX	DEC	
0	0	NUL	80	128		40	64	@	C0	192	
1	1	SOH	81	129		41	65	A	C1	193	
2	2	STX	82	130		42	66	B	C2	194	
3	3	ETX	83	131		43	67	C	C3	195	
4	4	EOT	84	132		44	68	D	C4	196	
5	5	ENQ	85	133		45	69	E	C5	197	
6	6	ACK	86	134		46	70	F	C6	198	
7	7	BEL	87	135		47	71	G	C7	199	
8	8	BS	88	136		48	72	H	C8	200	
9	9	HT	89	137		49	73	I	C9	201	
A	10	LF	8A	138		4A	74	J	CA	202	
B	11	VT	8B	139		4B	75	K	CB	203	
C	12	FF	8C	140		4C	76	L	CC	204	
D	13	CR	8D	141		4D	77	M	CD	205	
E	14	SO	8E	142		4E	78	N	CE	206	
F	15	SI	8F	143		4F	79	O	CF	207	
10	16	DLE	90	144		50	80	P	D0	208	
11	17	DC1	91	145		51	81	Q	D1	209	
12	18	DC2	92	146		52	82	R	D2	210	
13	19	DC3	93	147		53	83	S	D3	211	
14	20	DC4	94	148		54	84	T	D4	212	
15	21	NAK	95	149		55	85	U	D5	213	
16	22	SYN	96	150		56	86	V	D6	214	
17	23	ETB	97	151		57	87	W	D7	215	
18	24	CAN	98	152		58	88	X	D8	216	
19	25	EM	99	153		59	89	Y	D9	217	
1A	26	SUB	9A	154		5A	90	Z	DA	218	
1B	27	ESC	9B	155		5B	91	[	DB	219	
1C	28	FS	9C	156		5C	92	\	DC	220	
1D	29	GS	9D	157		5D	93	]	DD	221	
1E	30	RS	9E	158		5E	94	^	DE	222	
1F	31	US	9F	159		5F	95	_	DF	223	
20	32	SPACE	A0	160		60	96	`	E0	224	

POSITIVE			NEGATIVE			POSITIVE			NEGATIVE		
HEX	DEX	CHRS	HEX	DEC		HEX	DEC	CHRS	HEX	DEC	
21	33	!	A1	161		61	97	a	E1	225	
22	34	"	A2	162		62	98	b	E2	226	
23	35	#	A3	163		63	99	c	E3	227	
24	36	\$	A4	164		64	100	d	E4	228	
25	37	%	A5	165		65	101	e	E5	229	
26	38	&	A6	166		66	102	f	E6	230	
27	39	'	A7	167		67	103	g	E7	231	
28	40	(	A8	168		68	104	h	E8	232	
29	41	)	A9	169		69	105	i	E9	233	
2A	42	*	AA	170		6A	106	j	EA	234	
2B	43	+	AB	171		6B	107	k	EB	235	
2C	44	,	AC	172		6C	108	l	EC	236	
2D	45	-	AD	173		6D	109	m	ED	237	
2E	46	.	AE	174		6E	110	n	EE	238	
2F	47	/	AF	175		6F	111	o	EF	239	
30	48	0	B0	176		70	112	p	F0	240	
31	49	1	B1	177		71	113	q	F1	241	
32	50	2	B2	178		72	114	r	F2	242	
33	51	3	B3	179		73	115	s	F3	243	
34	52	4	B4	180		74	116	t	F4	244	
35	53	5	B5	181		75	117	u	F5	245	
36	54	6	B6	182		76	118	v	F6	246	
37	55	7	B7	183		77	119	w	F7	247	
38	56	8	B8	184		78	120	x	F8	248	
39	57	9	B9	185		79	121	y	F9	249	
3A	58	:	BA	186		7A	122	z	FA	250	
3B	59	;	BB	187		7B	123	{	FB	251	
3C	60	<	BC	188		7C	124		FC	252	
3D	61	=	BD	189		7D	125	}	FD	253	
3E	62	>	BE	190		7E	126	~	FE	254	
3F	63	?	BF	191		7F	127	RUB	FF	255	

EXPANDED POSITIVE/NEGATIVE ASCII CONTROL CODE MNEMONICS

This chart is useful for system programmers in setting up routines for printers, modems, communications and general I/O. (CTRL CD MN refers to control code mnemonic.)

POSITIVE ASCII CODES		NEGATIVE ASCII CODES		ASCII CTRL CD MN	EXPANDED MEANING
DEC	HEX	DEC	HEX		
0	0	128	80	NUL	Null
1	1	129	81	SOH	Start of Heading
2	2	130	82	STX	Start of text
3	3	131	83	ETX	End of text
4	4	132	84	EOT	End of transmission
5	5	133	85	ENQ	Inquiry
6	6	134	86	ACK	Acknowledge
7	7	135	87	BEL	Bell
8	8	136	88	BS	Backspace
9	9	137	89	HT	Horizontal tabulation
10	A	138	8A	LF	Linefeed
11	B	139	8B	VT	Vertical tabulation
12	C	140	8C	FF	Form feed
13	D	141	8D	CR	Carriage return
14	E	142	8E	SO	Shift out
15	F	143	8F	SI	Shift in
16	10	144	90	DLE	Data link escape
17	11	145	91	DC1	Device control 1
18	12	146	92	DC2	Device control 2
19	13	147	93	DC3	Device control 3
20	14	148	94	DC4	Device control 4
21	15	149	95	NAK	Negative acknowledge
22	16	150	96	SYN	Synchronous idle
23	17	151	97	ETB	End of transmission block
24	18	152	98	CAN	Cancel
25	19	153	99	EM	End of medium
26	1A	154	9A	SUB	Substitute
27	1B	155	9B	ESC	Escape
28	1C	156	9C	FS	File separator
29	1D	157	9D	GS	Group separator
30	1E	158	9E	RS	Record separator
31	1F	159	9F	US	Unit separator
127	7F	255	FF	RUB	Rub out

APPLESOFT/INTEGER BASIC DOS EXAMPLE GLOSSARY

This glossary is arranged in *alphabetical* order and contains Applesoft and Integer BASIC plus DOS commands. The examples are set up to show you how to use the commands and their proper syntax. In some cases when a command has different contexts of usage, more than a single example will be used. Some examples are given in the Immediate mode and some in the Program (deferred) mode <those with line numbers> and some with both. For clarification, results are given in some examples to show what a particular configuration would create. Commands with the Integer prompt (>) are available only in Integer (or have a unique feature) and those with an (a) are only Applesoft. DOS commands are indicated with (DOS).

NOTE: All DOS commands will work from either BASIC or from within the monitor or mini-assembler.

& Jumps to memory address \$3F4 to execute machine language subroutine. Works like CALL.

ABS() Gives the absolute value of a number or variable.

```
PRINT ABS(123.45)
```

AND Logic operator used in equations (assignments) and logic expressions.

```
140 IF A$ < > "Y" AND A$ < > "N" THEN GOTO 100
50 ON A$ = "Y" AND SUM AND CT GOTO 100, 200, 300
A = A$ = "Y" AND B$ = "Y"
POKE 6, A$ = "Y" AND F
```

APPEND (DOS) Adds data to end of existing sequential text file.

```
200 PRINT CHR$(4); "APPEND NAMES"
```

ASC() Returns ASCII value of first character in string.

```
PRINT ASC ("W") or A$ = "APPLE" : PRINT ASC(A $)
```

ATN() (a) Returns arctangent of number or variable.

```
PRINT ATN (123)
```

>AUTO Automatically enters line numbers in program. Increments can be included if default is different from 10.

```
AUTO 10
AUTO 2000,20 {Begin at line 200 and increment by 20.}
```


BLOAD (DOS) Load binary file into memory. Does not need address to load, but can be included.

<Immediate mode> BLOAD GRAPHICS FILE or BLOAD GRAPHICS FILE ,
A\$4000.

<Program mode> 100 PRINT CHR\$(4); "BLOAD GRAPHICS FILE"

BRUN (DOS) Runs binary file from disk.

<Immediate mode> BRUN SPACE APES

<Program mode> 100 PRINT CHR\$(4); "BRUN SPACE APES"

BSAVE (DOS) Saves binary file to disk. Must include both starting address and length in either hexadecimal or decimal.

<Hexadecimal> BSAVE ZOOM BLAST, A\$300, L\$3DF

<Decimal> BSAVE SPACE SHIP, A001, L1234

CALL Goes to machine subroutine at a given decimal address.

CALL -936 or CALL 768

CATALOG (DOS) Prints the contents of disk to screen, printer or other output device.

CATALOG,D1 or CATALOGD1 or CATALOG

>**CHAIN** (DOS) Allows user to maintain variables from one Integer BASIC program and RUN a second one.

CHAIN ACCOUNTANT #2

CHR\$() (a) Returns the character with a given decimal value.

PRINT CHR\$(65)

CLEAR (a) All variables are reset to zero.

120 CLEAR

>**CLR** All variables are reset to zero.

120 CLR

CLOSE Closes text file.

210 PRINT CHR\$(4); "CLOSE NAME LIST"

COLOR= Sets color in lo-res graphics in values from 0 to 15.

```
30 COLOR = 9
```

>CON Continue program after an END statement. Starts execution on line following where it stops program execution.

```
CON
```

CONT (a) Continue program after a STOP, END or CTRL-C.

```
CONT or 300 CONT (e.g., RUN 300 to continue)
```

COS() (a) Returns the cosine of variable or number.

```
PRINT COS(123)
```

DATA (a) Strings or numbers to be read.

```
1000 DATA 2, 345, HELLO, "WALK, TALK, AND BE HAPPY"
```

DEF FN() (a) Defines a function for simple real variable.

```
DEF FN A(X) = X * X  
PRINT FN A(4)
```

(Results = 16)

DEL Deletes range of line numbers.

```
DEL 120,200
```

DELETE (DOS) Deletes file from disk.

```
DELETE ADDING MACHINE
```

DIM (a) Allocates maximum range of array.

```
130 DIM A$ (100)
```


>DIM Defines a string variable's length to a maximum of 255.

```
20 DIM NA$(12)
30 NA$ = "THOMAS JONES"
```

DRAW N -- AT (a) Draws a hi-res shape numbered N on hi-res screen at X,Y coordinates.

```
150 DRAW 5 AT 100,40
```

>DSP Displays the value of a variable. DSP is not like a PRINT statement because every time the variable is changed it will be displayed.

```
10 DSP SUM
20 FOR COUNT = 1 TO 10
30 SUM = COUNT + SUM + 12
40 NEXT COUNT
```

END Terminates running of program and exits to Immediate mode. Required in Integer BASIC programs but optional in Applesoft.

```
200 END
```

EXEC (DOS). Executes the contents of a text file without removing program in memory.

```
EXEC PROGRAM SETUP
```

EXP() (a) Returns $e = 2.718289$ to indicated power.

```
PRINT EXP (3)
```

FLASH (a) Turns on flashing mode.

```
30 FLASH : PRINT "HELLO"
```

FN User defined function to be executed in program or in definition of variable. (See DEF FN).

```
300 PRINT FN C (44)
320 L = FN C (D)
```

FOR Sets up beginning of FOR/NEXT/STEP loop and top limit of loop.

```
40 FOR I = 1 TO 100
```

FP Sets to "floating point" or Applesoft language. Also clears memory.

```
FP
```

FRE() (a) Returns available memory.

```
PRINT FRE(0)
```

GET (a) Halts execution until single entry received from keyboard.

```
3 GET A or GET A$ or GET A%
```

GOSUB Branches to subroutine at given line number.

```
100 GOSUB 200
```

>**GOSUB** In Integer BASIC it is possible to branch to a variable.

```
40 LOOP = 300  
50 GOSUB LOOP
```

GOTO Branches to given line number.

```
100 GOTO 200
```

>**GOTO** Possible to branch to variable.

```
70 TEST = 1000  
80 IF A = 33 THEN GOTO TEST
```

GR Sets lo-res graphics mode.

```
120 GR
```

HCOLOR= (a) Set hi-res color to values of 0 to 7.

```
40 HCOLOR= 3
```

HGR (a) Sets page 1 hi-res graphics mode and clears hi-res screen 1 with bottom four lines as text.

```
120 HGR
```


HGR2 (a) Sets page 2 hi-res graphics mode and clears hi-res screen 2. All graphics and no text.

```
130 HGR2
```

HIMEM: Sets memory to highest desired location for a BASIC program and its variables within a range of 0 to 49152, depending on RAM memory.

```
10 HIMEM: 40123
```

HLIN -- AT Draws a horizontal line in lo-res graphics at given vertical position.

```
40 HLIN 10,20 AT 11
```

HOME (a) Clears screen and places cursor in upper left hand corner of text window.

```
HOME
```

HPlot -- TO (a) Plots points or lines on hi-res screen at given X,Y coordinates to another X,Y coordinate.

```
40 HPlot 10,50 : REM SINGLE POINT  
50 HPlot 0,0 TO 279,151 : REM RANGE OF POINTS (LINE)
```

HTAB (a) Horizontal tab position set.

```
50 HTAB 30 : PRINT "HERE"
```

IF/GOTO Branches to given line number if condition is met. (Note: The THEN is not required, but this works essentially the same as an IF/THEN statement.)

```
40 IF R > V GOTO 200
```

IF -- THEN Sets up conditional logic for execution.

```
60 IF A$ = "Q" THEN END
```

IN# Takes input from indicated slot #.

```
IN#6
```

INIT (DOS) Initializes diskette.

```
INIT HELLO
```

INPUT Halts program execution until string or numbers entered and RETURN key is pressed. May enter message within INPUT statement.

```
90 INPUT "ENTER WORD-> "; W$(I)
100 INPUT "ENTER NUMBER -> "; A
110 INPUT "ENTER INTEGER NUMBER -> "; N%
120 PRINT "HIT 'RETURN' TO CONTINUE ";
130 INPUT R$
>234 INPUT "CHOOSE NUMBER ", N (Note use of comma instead of semi-colon.)
```

INT (DOS) Sets to INTEGER BASIC language and clears memory.

INT

INT() (a) Returns the integer value of real variable or number.

```
PRINT INT (123.45)
```

INVERSE (a) Turns on Inverse mode.

```
50 INVERSE : PRINT "APPLE"
```

LEFT\$(,) (a) Returns specified number of characters from a given string beginning with character at far left.

```
10 A$ = "GOODBYE"
20 PRINT LEFT$ (A$,4)
(Results = GOOD)
```

LEN Returns the length in terms of number of characters of a specified string.

```
PRINT LEN(A$)
```

LET Used in specifying variable name in variable definition. Its use is optional.

```
30 LET X = 55
```

LIST Lists program currently in memory.

```
LIST
```

LOAD Loads Applesoft or Integer program specified.

```
LOAD CALENDAR PLOT
```


LOCK (DOS) Prevents file from being overwritten or deleted.

LOCK CALENDAR PLOT

LOG() (a) Returns natural logarithm (to base E) of specific number or variable.

PRINT LOG (15) or PRINT LOG (G)

LOMEM: Sets memory to lowest location available for a program within a range of - 2048 to 49152, depending on RAM memory.

LOMEM: 3245

>MAN Disables AUTO and returns line numbering to user.

MAN

MAXFILES (DOS) Reserves specified number of buffers for files within a range of 1-16 and reassigns HIMEM.

MAXFILES 10

MID\$(, ,) (a) Returns a portion of a string beginning with the nth character from the left for the number of characters indicated in the third position.

```
10 A$ = "WONDERFUL"  
20 PRINT MID$(A$,4,3)  
(Results = DER)
```

>MOD Returns the "modulo" or remainder of a division result.

```
PRINT 10 MOD 3  
(Results = 1)
```

MON (DOS) Turns on screen display of computer-disk communication.

MONCIO or MON C,I,O

NEW Clears program and variables in memory.

NEW

NEXT Sets the end of loop begun with FOR statement. Integer BASIC requires a variable after NEXT but it is optional in Applesoft.

```
10 FOR I = 1 TO 100
20 PRINT "THIS" ,
30 NEXT I
```

>NODSP Turns off DSP

NODSP

NOMON (DOS) Turns off screen display of computer-disk communication.

NOMONCIO or NOMON C,I,O

NORMAL (a) Returns screen to standard display from INVERSE or FLASH modes.

```
10 FLASH : PRINT "FLASHING MODE"
20 NORMAL: PRINT "NORMAL MODE"
```

NOT Logic negation in logic expression.

```
60 IF A NOT B THEN GOTO 100
70 C = NOT (D AND E)
```

NOTRACE Turns off TRACE mode.

NOTRACE

ON (a) Sets up computed GOTO and GOSUB.

```
190 ON A GOSUB 1000,2000,3000
```

ONERR Branches to specified line when error encountered.

```
40 ONERR GOTO 1000
```

OPEN (DOS) Creates or starts new text file.

```
500 PRINT CHR$(4); "OPEN NAME LIST"
```

OR Logic OR in logic expression.

```
130 IF A=10 OR B = 20 THEN GOTO 190
140 C = D OR E OR K
```


PDL() Returns value of specified paddle number 0 - 3.

```
PRINT PDL(0)
```

PEEK Returns memory byte's contents of given decimal location.

```
170 PRINT PEEK (768)
180 IF PEEK(768) = 5 THEN GOTO 200
```

PLOT Plots point in lo-res graphics in X,Y coordinates, visible in colors other than black.

```
PLOT 10,15
```

POKE Inserts given value in specified memory location.

```
POKE 768,10 (Sets memory location 768 to decimal value 10)
```

POP Used in GOSUB context, it removes top line number in the stack and makes the next line number the "return" point when the next RETURN is encountered.

```
10 GOSUB 100
20 END
100 GOSUB 200
200 PRINT "HELLO"
210 POP
220 RETURN
(Results = HELLO . Without POP, it would be HELLO HELLO.)
```

POS() (a) Gives the current horizontal position of the cursor.

```
10 PRINT "THIS LINE"; PRINT POS(0)
```

POSITION (DOS) Used in sequential text files to begin reading files at specified position rather than first record.

```
10 D$=CHR$(13) + CHR$(4)
20 PRINT D$ "OPEN NAME FILE" : PRINT D$ "POSITION NAME FILE,R9" : PRINT D$ "READ NAME FILE"
```

PR# Sends output to card in specified slot number.

```
PR#1
```

PRINT Outputs string, number, expression, function or variable to screen or printer.

```
PRINT 1;2;3; "GO"; F$; A; N%
```

READ (a) Enters DATA statement's contents into variable.

```
10 READ A : READ B$  
20 DATA 5, "BATS"
```

READ (DOS) Reads contents of text file.

```
40 PRINT CHR$(4) "READ NAME FILE"
```

RECALL (a) Loads array from tape that has been recorded with STORE.

```
RECALL Z
```

REM Non-executable command. Allows remarks in program lines.

```
10 BELL$ = CHR$(7): REM RINGS BELL
```

RENAME (DOS) Renames files on disk.

```
RENAME FAST SORT, SHELL SORT
```

RESTORE (a) Resets position of READ to first DATA statement.

```
10 FOR I = 1 TO 5 : READ A$(I) : NEXT  
20 RESTORE
```

RESUME (a) Goes to first statement of line where error occurred in error-handling routine.

```
10 ONERR GOTO 50  
20 INPUT V$  
30 END  
50 PRINT "ENTER ONLY INTEGER NUMBERS!" : RESUME
```

RETURN Returns program to next line after GOSUB command.

```
500 RETURN
```


RIGHT\$(,) (a) Returns the rightmost n characters of given string.

```
10 A$= "DATAMOST" : PRINT RIGHT$(A$,4)
```

(Results = MOST)

RND() Generates a random number less than 1 and greater than or equal to 0 in Applesoft. In Integer BASIC, RND returns integer number between 0 and number entered in parenthesis.

```
PRINT RND(5)
```

INT (RND (1) * (N) + 1) - Generates whole random numbers from 1 to N, with N being the upper limit of desired numbers.

```
>PRINT RND (5) : REM INTEGER BASIC - Generates whole random number  
from 0 to 5.
```

ROT= (a) Used to rotate shapes on hi-res screen. Can be set to angles between 0 to 255. Number of angular positions recognized depends on SCALE value of shape.

```
60 ROT= 64
```

RUN Executes program in memory or, if file name included, executes program from disk.

```
RUN FLYING MACHINE (executes disk file)
```

```
RUN (executes program in memory)
```

```
150 PRINT CHR$(4); "RUN FLYING MACHINE" (program mode)
```

SAVE Records program on disk.

```
SAVE GRAPH PLOT (Disk)  
(Tape)
```

SCALE= (a) Specifies size or scale of shape on hi-res screen.

```
40 SCALE= 5
```

SCRN(,) Returns the color code (0-15) in lo-res graphics of plot at X,Y coordinates.

```
30 PRINT SCRN (10,20)
```

SGN This will tell whether a given number or variable is less than 0, 0 or greater than 0. Essentially used to determine the sign of value as being positive or negative. Returns -1 if negative, 0 if 0 and + if number is greater than zero.

```
10 V = -10
```

```
20 PRINT SGN (V)
```

```
RESULTS = -1
```

SHLOAD (a) Command for loading shape table from tape.

```
SHLOAD
```

SIN() (a) Returns the sine of variable or number.

```
PRINT SIN(123)
```

SPC() (a) Prints specified number of spaces.

```
PRINT SPC(29); "HERE"
```

SPEED (a) Sets speed of execution from 0 to 255. (Default speed is 255.)

```
50 SPEED = 100
```

SQR() (a) Returns the square root of variable or number.

```
PRINT SQR(64)
```

STEP In FOR/NEXT loop, used to specify increments in loop.

```
20 FOR B = 10 TO 100 STEP 10  
RESULTS, B = 10 20 30 40 .... 100
```

STOP (a) Halts execution and prints line number where break occurs. (CONT command will re-start program at next instruction after STOP command.)

```
100 STOP
```

STORE (a) Records array values on tape.

```
STORE Z  
10 STORE Z
```

STR\$() (a) Converts number variable into string variable.

```
20 T= 123 : T$= STR$(T) : TT$= "$" + T$ + " .00"
```

TAB() (a) Sets horizontal tab from within a PRINT statement.

```
PRINT TAB(20); "HERE"
```

>**TAB** Works like HTAB in Applesoft BASIC.


```
10 TAB 10 : PRINT "HERE"
```

TAN() (a) Provides the tangent of number or variable.

```
40 T = 34 : V = 55
50 R = T + V : PRINT TAN(R)
```

TEXT Reset screen to text mode and full size.

```
TEXT
```

TRACE Turns on TRACE function for display of program execution. (Turned off with NOTRACE.)

```
10 TRACE
```

UNLOCK (DOS) Removes LOCK status from file on disk allowing it to be removed with DELETE statement or overwritten.

```
UNLOCK TAX CHART
```

VAL() (a) Used to convert string to numeric value.

```
30 H$ = "123" : PRINT VAL(H$)
```

VERIFY (DOS) Examines disk file to check for storage errors. If no such errors, nothing happens.

```
VERIFY TAX CHART
```

VLIN -- AT Draws vertical line in lo-res graphics at given horizontal position.

```
50 VLIN 1,30 AT 10
```

VTAB (a) Vertical tab position.

WAIT (a) Stops execution until memory location values meet given conditions.

```
80 WAIT -16384,128
```

WRITE (DOS) In making text files, specifies file name to which following PRINT statements will be written.

```
50 D$ = CHR$(13) + CHR$(4)
60 PRINT D$"OPEN NAMES": PRINT D$"WRITE NAMES"
70 PRINT "TOM" : PRINT "DICK" : PRINT "HARRY"
```

XDRAW (a) Used with shape tables to draw in complimentary color a specified shape on hi-res screen. Similar to DRAW.

```
70 XDRAW 3 AT 150,55
```

APPLESOFT BASIC LIST INTERPRETER OUTPUT CHART

This table represents Applesoft list output. The values on the left side of the chart (values 0-127 / \$0-\$7F) will be output to the screen or cause special conditions. (In the case of spaces, they appear as blanks on the chart. 235 <\$EB> is 2 spaces.) Values 1 through 31 and 127 (\$1-\$1F,\$7F) are special/control codes used with printers and other I/O devices. Zero (0) is reserved for the end of source line character and is not used in the ASCII/Tokenized source line.

On the right side of the chart, beginning with 128 (\$80), is the Applesoft token listing. The Applesoft token table begins at \$D0D0 and ends at \$D25F. Error messages begin at 236 (\$EC), and the Applesoft error message table begins at \$D260 and ends at \$D357.

One use of this chart is in inspecting disassembled code containing Applesoft programs. For example, using the monitor accessed through CALL -151, a disassembled listing of an Applesoft program might have B2 (\$B2) in the code. This is the token for the REM statement. Just for fun, change the B2 to an 89, and using CTRL-C return to Applesoft and LIST the program. Where the REM statement existed is now TEXT. This is because \$89 is the token for TEXT.

APPLESOFT LIST OUTPUT			APPLESOFT LIST OUTPUT			APSFT TABLE MEMORY ORDER
DEC	HEX		DEC	HEX		
0	0		128	80	" END "	1
1	1	SOH	129	81	" FOR "	2
2	2	STX	130	82	" NEXT "	3
3	3	ETX	131	83	" DATA "	4
4	4	EOT	132	84	" INPUT "	5
5	5	ENQ	133	85	" DEL "	6
6	6	ACK	134	86	" DIM "	7
7	7	BEL	135	87	" READ "	8
8	8	BS	136	88	" GR "	9
9	9	HT	137	89	" TEXT "	10
10	A	LF	138	8A	" PR# "	11
11	B	VT	139	8B	" IN# "	12
12	C	FF	140	8C	" CALL "	13
13	D	CR	141	8D	" PLOT "	14
14	E	SO	142	8E	" HLIN "	15
15	F	SI	143	8F	" VLIN "	16
16	10	DLE	144	90	" HGR2 "	17
17	11	DC1	145	91	" HGR "	18
18	12	DC2	146	92	" HCOLOR= "	19
19	13	DC3	147	93	" HPLOT "	20
20	14	DC4	148	94	" DRAW "	21
21	15	NAK	149	95	" XDRAW "	22
22	16	SYN	150	96	" HTAB "	23
23	17	ETB	151	97	" HOME "	24
24	18	CAN	152	98	" ROT= "	25
25	19	EM	153	99	" SCALE= "	26
26	1A	SUB	154	9A	" SHLOAD "	27
27	1B	ESC	155	9B	" TRACE "	28

DEC	HEX	APPLESOFT LIST OUTPUT	DEC	HEX	APPLESOFT LIST OUTPUT	APSFT TABLE MEMORY ORDER
28	1C	FS	156	9C	" NOTRACE "	29
29	1D	GS	157	9D	" NORMAL "	30
30	1E	RS	158	9E	" INVERSE "	31
31	1F	US	159	9F	" FLASH "	32
32	20	"SPACE"	160	A0	" COLOR= "	33
33	21	!	161	A1	" POP "	34
34	22	"	162	A2	" VTAB "	35
35	23	#	163	A3	" HIMEM: "	36
36	24	\$	164	A4	" LOMEM: "	37
37	25	%	165	A5	" ONERR "	38
38	26	&	166	A6	" RESUME "	39
39	27	'	167	A7	" RECALL "	40
40	28	(	168	A8	" STORE "	41
41	29	)	169	A9	" SPEED= "	42
42	2A	*	170	AA	" LET "	43
43	2B	+	171	AB	" GOTO "	44
44	2C	,	172	AC	" RUN "	45
45	2D	-	173	AD	" IF "	46
46	2E	.	174	AE	" RESTORE "	47
47	2F	/	175	AF	" & "	48
48	30	0	176	B0	" GOSUB "	49
49	31	1	177	B1	" RETURN "	50
50	32	2	178	B2	" REM "	51
51	33	3	179	B3	" STOP "	52
52	34	4	180	B4	" ON "	53
53	35	5	181	B5	" WAIT "	54
54	36	6	182	B6	" LOAD "	55
55	37	7	183	B7	" SAVE "	56
56	38	8	184	B8	" DEF "	57
57	39	9	185	B9	" POKE "	58
58	3A	:	186	BA	" PRINT "	59
59	3B	;	187	BB	" CONT "	60
60	3C	<	188	BC	" LIST "	61
61	3D	=	189	BD	" CLEAR "	62
62	3E	>	190	BE	" GET "	63
63	3F	?	191	BF	" NEW "	64
64	40	@	192	C0	" TAB("	65
65	41	A	193	C1	" TO "	66
66	42	B	194	C2	" FN "	67
67	43	C	195	C3	" SPC("	68
68	44	D	196	C4	" THEN "	69
69	45	E	197	C5	" AT "	70
70	46	F	198	C6	" NOT "	71
71	47	G	199	C7	" STEP "	72
72	48	H	200	C8	" + "	73
73	49	I	201	C9	" - "	74
74	4A	J	202	CA	" * "	75
75	4B	K	203	CB	" / "	76
76	4C	L	204	CC	" ^ "	77
77	4D	M	205	CD	" AND "	78

DEC	HEX	APPLESOFT LIST OUTPUT	DEC	HEX	APPLESOFT LIST OUTPUT	APSFT TABLE MEMORY ORDER
78	4E	N	206	CE	" OR "	79
79	4F	O	207	CF	" > "	80
80	50	P	208	D0	" = "	81
81	51	Q	209	D1	" < "	82
82	52	R	210	D2	" SGN "	83
83	53	S	211	D3	" INT "	84
84	54	T	212	D4	" ABS "	85
85	55	U	213	D5	" USR "	86
86	56	V	214	D6	" FRE "	87
87	57	W	215	D7	" SCRN("	88
88	58	X	216	D8	" PDL "	89
89	59	Y	217	D9	" POS "	90
90	5A	Z	218	DA	" SQR "	91
91	5B	[	219	DB	" RND "	92
92	5C	\	220	DC	" LOG "	93
93	5D	]	221	DD	" EXP "	94
94	5E	^	222	DE	" COS "	95
95	5F	_	223	DF	" SIN "	96
96	60	`	224	E0	" TAN "	97
97	61	a	225	E1	" ATN "	98
98	62	b	226	E2	" PEEK "	99
99	63	c	227	E3	" LEN "	100
100	64	d	228	E4	" STR\$ "	101
101	65	e	229	E5	" VAL "	102
102	66	f	230	E6	" ASC "	103
103	67	g	231	E7	" CHR\$ "	104
104	68	h	232	E8	" LEFT\$ "	105
105	69	i	233	E9	" RIGHT\$ "	106
106	6A	j	234	EA	" MID\$ "	107
107	6B	k	235	EB	" NUL NUL~ "	
108	6C	l	236	EC	" SYNTAX "	1
109	6D	m	237	ED	" RETURN WITHOUT GOSUB "	2
110	6E	n	238	EE	" OUT OF DATA "	3
111	6F	o	239	EF	" ILLEGAL QUANTITY "	4
112	70	p	240	F0	" OVERFLOW "	5
113	71	q	241	F1	" OUT OF MEMORY "	6
114	72	r	242	F2	" UNDEF'D STATEMENT "	7
115	73	s	243	F3	" BAD SUBSCRIPT "	8
116	74	t	244	F4	" REDIM'D ARRAY "	9
117	75	u	245	F5	" DIVISION BY ZERO "	10
118	76	v	246	F6	" ILLEGAL DIRECT "	11
119	77	w	247	F7	" TYPE MISMATCH "	12
120	78	x	248	F8	" STRING TOO LONG "	13
121	79	y	249	F9	" FORMULA TOO COMPLEX "	14
122	7A	z	250	FA	" CAN'T CONTINUE "	15
123	7B	{	251	FB	" UNDEF'D FUNCTION "	16
124	7C		252	FC	" ERROR `BEL NUL NUL~" "	17
125	7D	}	253	FD	" h "	
126	7E	~	254	FE	" h "	
127	7F	RUB	255	FF	" h "	

INTEGER BASIC LIST INTERPRETER CHART

This chart is helpful for examining numeric byte listings (e.g., hex dumps) of Integer BASIC programs, and generally advancing your understanding of Integer BASIC. The strings from 0-127 (\$0-7F) are shown in quotes even though the quotes are not part of the output string. Rather, the quotation marks show leading and trailing spaces. The "SPACE" (decimal 160 / hex \$A0) is spelled out as {SPACE} for clarification. The {EOL} after decimal/hex 1 represents the "end of line" terminator and does not list as output. Control codes (128-159 / \$80-\$9F) are indented, representing single value outputs and are shown in their mnemonic form.

For example the following line in BASIC,

```
10 TEXT : REM
```

looks as follows in numeric byte listings:

```
95D2 - 07 0A 00 4B 03 5D
```

The line number 10 is coded as 0A, the string literal TEXT is 4B, the colon 03 and the REM statement, 5D.

It should be noted that different token values are employed in Applesoft BASIC, and changing from Integer to Applesoft requires the use of the appropriate table. For example, in Applesoft, the token for a space is \$20, while it is \$A0 in Integer BASIC. Likewise, reserved words have different tokens. "TEXT" is \$89 in Applesoft and \$4B in Integer BASIC.

To examine an Integer BASIC program, enter the monitor with CALL -151, and list the memory locations where your program is loaded. (Once in the monitor, key in CA.CB to obtain the beginning address of an Integer program.) By examining the tokens in this chart, you can see the ways in which your program is translated into code that the Integer BASIC run-time interpreter "understands." It is possible to make changes in the monitor and see the results in your listing, including "illegal" code. For example, change the 5D to a DD in the above example.

INTERPRETED LIST VALUE		RESULTANT STRING EXPANDED TOKEN	INTERPRETED LIST VALUE		RESULTANT STRING NEG ASCII CHAR SET
DEC	HEX		DEC	HEX	
0	0	" HIMEM: "	128	80	NUL
1	1	" EOL "	129	81	SOH
2	2	" "	130	82	STX
3	3	" :	131	83	ETX
4	4	" LOAD "	132	84	EOT
5	5	" SAVE "	133	85	ENQ
6	6	" CON "	134	86	ACK
7	7	" RUN "	135	87	BEL
8	8	" RUN "	136	88	BS

INTERPRETED LIST VALUE			RESULTANT STRING EXPANDED TOKEN		INTERPRETED LIST VALUE			RESULTANT STRING NEG ASCII CHAR SET	
DEC	HEX				DEC	HEX			
9	9	" DEL "			137	89		HT	
10	A	" , "			138	8A		LF	
11	B	" NEW "			139	8B		VT	
12	C	" CLR "			140	8C		FF	
13	D	" AUTO "			141	8D		CR	
14	E	" , "			142	8E		SO	
15	F	" MAN "			143	8F		SI	
16	10	" HIMEM: "			144	90		DLE	
17	11	" LOMEM: "			145	91		DC1	
18	12	" + "			146	92		DC2	
19	13	" - "			147	93		DC3	
20	14	" * "			148	94		DC4	
21	15	" / "			149	95		NAK	
22	16	" = "			150	96		SYN	
23	17	" # "			151	97		ETB	
24	18	" > = "			152	98		CAN	
25	19	" > "			153	99		EM	
26	1A	" < = "			154	9A		SUB	
27	1B	" < > "			155	9B		ESC	
28	1C	" < "			156	9C		FS	
29	1D	" AND "			157	9D		GS	
30	1E	" OR "			158	9E		RS	
31	1F	" MOD "			159	9F		US	
32	20	" ^ "			160	A0		"SPACE"	
33	21	" + "			161	A1		!	
34	22	" ("			162	A2		"	
35	23	" , "			163	A3		#	
36	24	" THEN "			164	A4		\$	
37	25	" THEN "			165	A5		%	
38	26	" , "			166	A6		&	
39	27	" , "			167	A7		'	
40	28	" " "			168	A8		(	
41	29	" " "			169	A9		)	
42	2A	" ("			170	AA		*	
44	2C	" ! "			172	AC		,	
45	2D	" ("			173	AD		-	
46	2E	" PEEK "			174	AE		.	
47	2F	" RND "			175	AF		/	
48	30	" SGN "			176	B0		Ø	
49	31	" ABS "			177	B1		1	
50	32	" PDL "			178	B2		2	
51	33	" RNDX "			179	B3		3	
52	34	" ("			180	B4		4	
53	35	" + "			181	B5		5	
54	36	" - "			182	B6		6	
55	37	" NOT "			183	B7		7	
56	38	" ("			184	B8		8	
57	39	" = "			185	B9		9	
58	3A	" # "			186	BA		:	
59	3B	" LEN ("			187	BB		;	

INTERPRETED LIST VALUE			RESULTANT STRING EXPANDED TOKEN		INTERPRETED LIST VALUE			RESULTANT STRING NEG ASCII CHAR SET	
DEC	HEX				DEC	HEX			
60	3C	"	ASC ("		188	BC	<		
61	3D	"	SCRN ("		189	BD	=		
62	3E	"	,		190	BE	>		
63	3F	"	("		191	BF	?		
64	40	"	\$ "		192	C0	@		
65	41	"	\$ "		193	C1	A		
66	42	"	("		194	C2	B		
67	43	"	,		195	C3	C		
68	44	"	,		196	C4	D		
69	45	"	;		197	C5	E		
70	46	"	;		198	C6	F		
71	47	"	;		199	C7	G		
72	48	"	,		200	C8	H		
73	49	"	,		201	C9	I		
74	4A	"	,		202	CA	J		
75	4B	"	TEXT "		203	CB	K		
76	4C	"	GR "		204	CC	L		
77	4D	"	CALL "		205	CD	M		
78	4E	"	DIM "		206	CE	N		
79	4F	"	DIM "		207	CF	O		
80	50	"	TAB "		208	D0	P		
81	51	"	END "		209	D1	Q		
82	52	"	INPUT "		210	D2	R		
83	53	"	INPUT "		211	D3	S		
84	54	"	INPUT "		212	D4	T		
85	55	"	FOR "		213	D5	U		
86	56	"	= "		214	D6	V		
87	57	"	TO "		215	D7	W		
88	58	"	STEP "		216	D8	X		
89	59	"	NEXT "		217	D9	Y		
90	5A	"	,		218	DA	Z		
91	5B	"	RETURN "		219	DB	[		
92	5C	"	GOSUB "		220	DC	\		
93	5D	"	REM "		221	DD	]		
94	5E	"	LET "		222	DE	^		
95	5F	"	GOTO "		223	DF	_		
96	60	"	IF "		224	E0	`		
97	61	"	PRINT "		225	E1	a		
98	62	"	PRINT "		226	E2	b		
99	63	"	PRINT "		227	E3	c		
100	64	"	POKE "		228	E4	d		
101	65	"	,		229	E5	e		
102	66	"	COLOR = "		230	E6	f		
103	67	"	PLOT "		231	E7	g		
104	68	"	,		232	E8	h		
105	69	"	HLIN "		233	E9	i		
106	6A	"	,		234	EA	j		
107	6B	"	AT "		235	EB	k		
108	6C	"	VLIN "		236	EC	l		

INTERPRETED LIST VALUE			RESULTANT STRING EXPANDED TOKEN		INTERPRETED LIST VALUE			RESULTANT STRING NEG ASCII CHAR SET	
DEC	HEX				DEC	HEX			
109	6D	" , "			237	ED	m		
110	6E	" AT "			238	EE	n		
111	6F	" VTAB "			239	EF	o		
112	70	" = "			240	F0	p		
113	71	" = "			241	F1	q		
114	72	") "			242	F2	r		
115	73	") "			243	F3	s		
116	74	" LIST "			244	F4	t		
117	75	" , "			245	F5	u		
118	76	" LIST "			246	F6	v		
119	77	" POP "			247	F7	w		
120	78	" NODSP "			248	F8	x		
121	79	" NODSP "			249	F9	y		
122	7A	" NOTRACE "			250	FA	z		
123	7B	" DSP "			251	FB	{		
124	7C	" DSP "			252	FC			
125	7D	" TRACE "			253	FD	}		
126	7E	" PR# "			254	FE	~		
127	7F	" IN# "			255	FF		RUB	

POKES, PEEKS and CALLS

This chart is designed for the BASIC programmer who would like to improve or enhance his/her programming by using the built-in machine level subroutines of the Apple computer. For the novice, the use of POKES, PEEKs and CALLs is the first step to machine/assembly level programming. And for the veteran programmer, it is a handy chart for looking up decimal values for various functions.

Most POKES simply enter a value into a memory location while PEEKs return the value of what is stored in that location. However, certain POKES and PEEKs make the computer do special things, such as changing the size of the text window or making sounds. The CALL command, on the other hand, executes either built-in or programmed machine coded subroutines. If you CALL -1401, your computer will boot itself. (It will also boot on CALL -1370.) All that's happening is that the built-in subroutines in Apple's monitor are being cranked up and executed. In a lot of programs, you will see CALL 768. That's because the program has a machine level program loaded in memory at hex location \$300, a common place to store machine code in RAM memory. Since all CALLs, as well as POKES and PEEKs, have to be entered in decimal, and 768 is the decimal equivalent to \$300, the CALL is to 768 instead of \$300.

A convention we will use is providing negative numbers where appropriate. In the old days when Apple first came out, the only BASIC was Integer BASIC. When using values above 32767, it was necessary to use the difference between the value and 65536. For example, to enter the monitor, most use CALL -151. However, the subroutine to jump into the monitor is actually at decimal location 65385. By subtracting 65536 from 65385, though, we get -151, which is read by the computer the same as it would read 65385. It is perfectly all right to CALL the higher number on all Apple II+'s and Apple IIe's (or Apple II's with Applesoft on ROM or in a language card), but when using Integer BASIC, you have to use the negative number. So, instead of using the longer positive numbers over 32767, we will incorporate the negative numbers. In that way, everybody can use them.

Since the Apple stores values in hexadecimal with the low byte before the high byte, (backwards!), when PEEKing a location using two byte values, it is necessary to convert the value to a useful decimal one. By using

$$A = \text{PEEK}(\text{XXXX}) + 256 * \text{PEEK}(\text{XXXX}+1)$$

where XXXX is the first of two adjacent decimal locations in memory, the variable A will store the correct decimal value for you. For example, when BSAVEing a binary program, it is necessary to specify the starting address and length of the program. The pointers for the beginning of a binary program are at \$AA72 and \$AA73 (43634 and 43635), and the pointers for the length of the program are at \$AA60 and \$AA61 (43616 and 43617). To find the decimal values for the most recently BLOADED binary program we would enter

```
B = PEEK (43634) + 256 * PEEK (43635)
L = PEEK (43616) + 256 * PEEK (43617)
PRINT "BEGINNING = "; B : PRINT "LENGTH = "; L
```

When we BSAVE the program we would enter

```
BSAVE PROGRAM NAME, A 1234, L 567
```

with the actual values of 1234 and 567 being whatever values were returned by the variable B and L. (Note: When BSAVEing with hexadecimal values, we use A\$ and L\$.)

The following chart of POKEs, PEEKs and CALLs is arranged in terms of the most common uses of the addresses. This list is by no means exhaustive, but rather it is devised to provide a quick reference to some of the most commonly used POKE, PEEK and CALL addresses, along with typical parameters, that BASIC programmers will need. It is possible to POKE locations which are listed under PEEK, and vice versa. In fact, since most of the charts in this book have decimal values, you can access any of them you want with the POKE, PEEK or CALL, but unless you're careful, strange things might happen.

POKES

COMMAND	PARAMETERS
POKE 32 Set left edge text widow	0 - 39
POKE 33 Set width of text window	1 - 40
POKE 34 Set top edge of text window	2 - 22
POKE 35 Set bottom edge of text window	1 - 24
POKE 50 Text display (normal, inverse, flash)	255, 63, 127
POKE 216 Cancel ONERR	0
POKE -16151 Turns on Drive (Good for cleaning heads)	0
POKE -16152 Turns off Drive	0
POKE -16135 Turns on Drive 1	0
POKE -16316 Turns on Drive 2	0
POKE -16297 High resolution graphics	0
POKE -16298 Low resolution graphics	0
POKE -16299 Page 2	0
POKE -16300 Page 1	0
POKE -16301 Text & Graphics	0
POKE -16302 All graphics	0
POKE -16303 Text	0
POKE -16304 Graphics	0
POKE -16368 Clear keyboard strobe	0
POKE 49167 Turn on alternate chr. set Apple //e	0
POKE 49166 Turn off alternate chr. set Apple //e	0

NOTE: PEEK values can be PRINTed or stored as variables.

PEEKs

COMMAND FORMAT

RESULTS

PEEK (103) + 256 * PEEK (104)	Beginning address of FP program
PEEK (202) + 256 * PEEK (203)	Beginning address of INT program
PEEK (232) + 256 * PEEK (233)	Beginning address of shape table
PEEK (225) + 256 * PEEK (225)	Horizontal position of last HPLOT
PEEK (226)	Vertical position of last HPLOT
PEEK (43634) + 256 * PEEK (43635)	Beginning address of BIN program
PEEK (43616) + 256 * PEEK (43617)	Length of BIN program
PEEK (-16384)	Last keypress entered
PEEK (-16336)	When defined as variable, clicks speaker
PEEK (-16287)	Reads PDL (0) button >127 if pressed
PEEK (-16286)	Reads PDL (1) button >127 if pressed
PEEK (-16285)	Reads PDL (2) button >127 if pressed
PEEK (-16284)	Reads PDL (3) button >127 if pressed
PEEK (222)	Code for error on ONERR message
0 NEXT without FOR	22 RETURN without GOSUB
1 Language not available	42 BASIC out of DATA
2,3 Range error	53 Illegal quantity
4 Write protected	69 Overflow error
5 End of data (DOS)	77 Out of memory
6 File not found	90 Undefined statement
7 Volume mismatch	107 Bad subscript
8 IO error	120 Redimensioned array
9 Disk full	133 Division by zero
10 Locked file	163 Type mismatch
11 DOS Syntax error	176 String too long
12 No buffers available	191 Formula too complex
13 File type mismatch	224 Undefined function
14 Program too large	254 Bad input response
15 Not direct command	255 Control-C break
16 BASIC syntax error (FP)	

CALLS

CALL VALUE

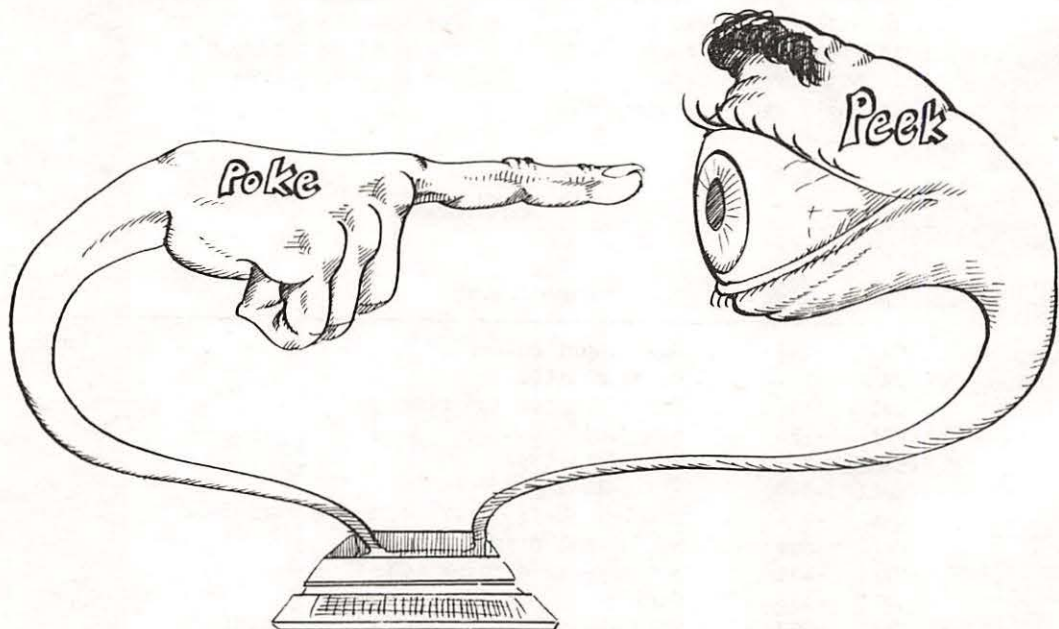
SUBROUTINE

CALL -144	Scan input buffer
CALL -151	Enter monitor
CALL -167	Enter monitor and reset
CALL -198	Ring bell
CALL -259	Read from tape
CALL -310	Write to tape
CALL -321	Display the A, X, Y, P, and S registers
CALL -380	Set normal display mode
CALL -381	Set inverse display mode
CALL -458	Verify
CALL -550	Print hexadecimal value of accumulator

CALL VALUE

SUBROUTINE

CALL -657	Allows input of commas and colons
CALL -662	Get line of input with prompt, no linefeed
CALL -665	Get line of input with prompt, linefeed
CALL -678	Wait until RETURN is pressed
CALL -756	Wait until any key is pressed
CALL -856	Delay of POKE 69,dd with dd = delay (0 -255)
CALL -868	Clear from cursor to right
CALL -875	Clear whole line of text
CALL -912	Scroll one line
CALL -922	Carriage return
CALL -936	Clear screen and home cursor
CALL -958	Clear text from cursor to bottom of screen
CALL -998	Move cursor up one line
CALL -1002	Re-connect DOS
CALL -1008	Move cursor left one space
CALL -1036	Move cursor right one space
CALL -1216	Set GR (especially useful in INT BASIC)
CALL -1321	Display registers
CALL -1370	Boot disk (common)
CALL -1401	Boot disk and impress friends (uncommon)
CALL -1728	Display hexadecimal values of X and Y registers
CALL -2458	Enter mini-assembler
CALL -3106	HGR2 (especially useful in INT BASIC)
CALL -3116	HGR (especially useful in INT BASIC)
CALL -3776	Save INTEGER
CALL 42350	CATALOG



SORT ALGORITHMS

The following are three common sort algorithms. They have been set up for sorting strings, but the same algorithms will sort numbers. The array variable A\$ is used with all for purposes of illustration. The most common, shortest and slowest is the BUBBLE SORT. It is good for short lists and longer lists that are partially sorted. The second is the SHELL-METZNER (or SHELL) SORT. This algorithm works faster than the BUBBLE SORT, but it is more complicated. It also works faster than the QUICK SORT on short and medium sized lists. Finally, the QUICK SORT works the best on longer lists and was benchmarked to be the best of the three on longer, wholly unsorted lists.

To use the sorts, simply enter the strings in the array A\$ on one end of the sort and the output on the other. For example, the following routine attached to the sorts will sort any number of strings for you:

```
10 TEXT : HOME
20 INPUT "HOW MANY STRINGS TO ENTER";N
30 DIM A$(N)
40 FOR I = 1 TO N: INPUT "ENTER STRING => ";A$(I):
   NEXT
50 HOME

5000 REM SORTED OUTPUT
5010 FOR I = 1 TO N: PRINT A$(I) : NEXT
```

To see the differences for yourself, try the above input format with different lists of strings.

```
100 FLAG = 0
110 FOR I = 1 TO N - 1
120 IF A$(I) > A$(I + 1) THEN FL
    AG = 1:TEMP$ = A$(I):A$(I) =
    A$(I + 1):A$(I + 1) = TEMP$
130 NEXT
140 IF FLAG THEN 100
```

```

2000 REM SHELL SORT - STRING = A$(I) N = NUMBER OF CASES
2010 X = (2 ^ INT ( LOG (N) / LOG (2))) - 1
2020 X = INT (X / 2)
2030 IF X < 1 THEN 2160
2040 FOR J = 1 TO X
2050 FOR K = J + X TO N STEP X
2060 I = K
2070 T$ = A$(I)
2080 IF A$(I - X) < = T$ THEN 2120
2090 A$(I) = A$(I - X)
2100 I = I - X
2110 IF I > X THEN 2080
2120 A$(I) = T$
2130 NEXT K
2140 NEXT J
2150 GOTO 2020
2160 REM SORT DONE

3000 REM QUICKSORT
3010 REM SORTS ON A$(N)

```



```

3020 W1 = 1
3030 X(1) = 1
3040 Z(1) = N
3050 X1 = X(W1)
3060 Z1 = Z(W1)
3070 W1 = W1 - 1
3080 X2 = X1
3090 Z2 = Z1
3100 X$ = A$( INT ((X1 + Z1) / 2))
3110 REM
3120 IF A$(X2) > = X$ THEN 3150
3130 X2 = X2 + 1
3140 GOTO 3120
3150 REM
3160 IF X$ > = A$(Z2) THEN 3190
3170 Z2 = Z2 - 1
3180 GOTO 3160
3190 IF X2 > Z2 THEN 3260
3200 W = W + 1
3210 T$ = A$(X2)
3220 A$(X2) = A$(Z2)
3230 A$(Z2) = T$
3240 X2 = X2 + 1
3250 Z2 = Z2 - 1
3260 IF X2 < = Z2 THEN 3110

```

```
3270 IF X2 > = Z1 THEN 3310
3280 W1 = W1 + 1
3290 X(W1) = X2
3300 Z(W1) = Z1
3310 Z1 = Z2
3320 IF X1 < Z1 THEN 3080
3330 IF W1 > 0 THEN 3050
3340 REM SORT COMPLETE
```


RAM/DISK MEMORY CROSS REFERENCE

This table can be used for both 16 sector and the older 13 sector formatted disks. The 48K RAM addresses are the same as the 64K machines (Apple //e's and Apple II+/II's with 64K RAM). In examining DOS it is possible, using this chart, to locate modifications in DOS, make your own modifications and generally cross reference locations in RAM and DOS memory. The first two lines (indented on the left and right margins) represent a master disk's code to relocate DOS into different sized machines.

For example, the title seen when you CATALOG a disk is usually DISK VOLUME. This is located at \$B3AF - \$B3BA (45999 - 46010). By RUNNING the following program you can see that:

```
10 TEXT : HOME
20 N = 1023
30 FOR I = 46010 TO 45999 STEP -1
40 N = N + 1
50 POKE N, PEEK (I) : NEXT I
60 END
```

However, you can change the contents of that message as well. Try the following program, and then CATALOG your disk:

```
10 TEXT : HOME
20 FOR I = 46010 TO 45999 STEP -1 : READ D
30 POKE I, D : NEXT
40 DATA 1, 16, 16, 32, 1, 12, 13, 1, 14, 1, 3, 32
50 END
```

Some abbreviations were used so the table could be condensed, and certain notations were necessary:

*All mastered disks load here and then relocate.

**TTSBB - T = Track, S = Sector, B = Byte (the double T's and B's reflect two digit values of tracks and bytes).

***T S B - T = Track, S = Sector, B = Byte.

RAM MEMORY 48K DOS RAM ADDRESS		16K DOS RAM ADDRESS*		DISK MEMORY LOC 16-SECTOR OFFSET TTSBB**		DISK MEM LOC 13-SEC*** OFFSET T S B		DISK MEM LOC 13-SEC OFFSET T S B	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
		6,912	1B00	2,560	A00	2,560	0 A 0		
		7,168	1C00	2,816	B00	2,816	0 B 0		
40,192	9D00	7,424	1D00	3,072	C00	3,072	0 C 0	2,560	0 A 0
40,448	9E00	7,680	1E00	3,328	D00	3,328	1 0 0	2,816	0 B 0
40,704	9F00	7,936	1F00	3,584	E00	3,584	1 1 0	3,072	0 C 0
40,960	A000	8,192	2000	3,840	F00	3,840	1 2 0	3,328	1 0 0
41,216	A100	8,448	2100	4,096	1000	4,096	1 3 0	3,584	1 1 0
41,472	A200	8,704	2200	4,352	1100	4,352	1 4 0	3,840	1 2 0

RAM MEMORY 48K DOS RAM ADDRESS		16K DOS RAM ADDRESS*		DISK MEMORY LOC 16-SECTOR OFFSET TTSBB**		DISK MEM LOC 13-SEC*** OFFSET T S B		DISK MEM LOC 13-SEC OFFSET T S B	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
41,728	A300	8,960	2300	4,608	1200	4,608	1 5 0	4,096	1 3 0
41,984	A400	9,216	2400	4,864	1300	4,864	1 6 0	4,352	1 4 0
42,240	A500	9,472	2500	5,120	1400	5,120	1 7 0	4,608	1 5 0
42,496	A600	9,728	2600	5,376	1500	5,376	1 8 0	4,864	1 6 0
42,752	A700	9,984	2700	5,632	1600	5,632	1 9 0	5,120	1 7 0
43,008	A800	10,240	2800	5,888	1700	5,888	1 A 0	5,376	1 8 0
43,264	A900	10,496	2900	6,144	1800	6,144	1 B 0	5,632	1 9 0
43,520	AA00	10,752	2A00	6,400	1900	6,400	1 C 0	5,888	1 A 0
43,776	AB00	11,008	2B00	6,656	1A00	6,656	2 0 0	6,144	1 B 0
44,032	AC00	11,264	2C00	6,912	1B00	6,912	2 1 0	6,400	1 C 0
44,288	AD00	11,520	2D00	7,168	1C00	7,168	2 2 0	6,656	2 0 0
44,544	AE00	11,776	2E00	7,424	1D00	7,424	2 3 0	6,912	2 1 0
44,800	AF00	12,032	2F00	7,680	1E00	7,680	2 4 0	7,168	2 2 0
45,056	B000	12,288	3000	7,936	1F00	7,936	2 5 0	7,424	2 3 0
45,312	B100	12,544	3100	8,192	2000	8,192	2 6 0	7,680	2 4 0
45,568	B200	12,800	3200	8,448	2100	8,448	2 7 0	7,936	2 5 0
45,824	B300	13,056	3300	8,704	2200	8,704	2 8 0	8,192	2 6 0
46,080	B400	13,312	3400	8,960	2300	8,960	2 9 0	8,448	2 7 0
46,336	B500	13,568	3500	9,216	2400	9,216	2 A 0	8,704	2 8 0
46,592	B600	13,824	3600	0	0	0 0 0	0	0 0 0	0
46,848	B700	14,080	3700	256	100	256	0 1 0	256	0 1 0
47,104	B800	14,336	3800	512	200	512	0 2 0	512	0 2 0
47,360	B900	14,592	3900	768	300	768	0 3 0	768	0 3 0
47,616	BA00	14,848	3A00	1,024	400	1,024	0 4 0	1,024	0 4 0
47,872	BB00	15,104	3B00	1,280	500	1,280	0 5 0	1,280	0 5 0
48,128	BC00	15,360	3C00	1,536	600	1,536	0 6 0	1,536	0 6 0
48,384	BD00	15,616	3D00	1,792	700	1,792	0 7 0	1,792	0 7 0
48,640	BE00	15,872	3E00	2,048	800	2,048	0 8 0	2,048	0 8 0
48,896	BF00	16,128	3F00	2,304	900	2,304	0 9 0	2,304	0 9 0

Slave boots
directly
here.

Master boots
and moves code
up to highest
point.

Master and
Slave boot here
16 Sector

13 Sector

RAM MEMORY

DISK
MEMORY

DISK MEMORY



DOS BUFFER CHARTS

When DOS is loaded in your Apple, it is in RAM memory. Since it is in RAM, it can be changed, and much of the information to be examined or changed is stored in the DOS Buffers. The following set of charts shows how these buffers are organized so that they may be used in disk access programming or examining other programs using DOS. There are a total of 16 buffers, each divided into three sections--Data Sector, Track/Sector List and File Manager Work Area.

In the far left hand column is the relative byte offset from an address for locating the different buffers in each buffer section. Each new section, it should be noted, begins with 0, and Section 1 has two 0 offsets. Section 1 has the following organization:

DEC	HEX	
0	0	Start
594	252	End
0	0	Data Buffer
256	100	Track Sector List Buffer
512	200	File Manager Work Area
557	22D	File Name Buffer
587	24B	Pointer
589	24D	Pointer
591	24F	Pointer
593	251	Pointer

To find the address of the end of DOS Buffer #3, for example, examine the second row with offsets of 594 (\$252) and then go over to the DOS BUFFER #3 column and find the address to be 38,994 (\$9852). Similarly, to find the File Manager Work Area of a buffer, #9 we'll say, first we would locate DOS Buffer #9, and then going to the fifth row of the far left column we would locate 512 (\$200), and then go over to the DOS Buffer #9 column and find the address to be 35,342 (\$8A0E). To locate addresses in the Track/Sector List Buffer or File Manager Work Area, find the offset in the far left hand column in the appropriate Section area, and then cross index to the desired buffer column. Conversely, start with the known buffer and section, and cross index to find the address and/or offset. The addresses are listed sequentially, from top to bottom. That is, the higher numbered buffers have the lower numbered addresses in RAM memory.

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #1 (IN COM. MEMORY)		DOS BUFFER #2		DOS BUFFER #3		DOS BUFFER #4	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX

SECTION 1: DATA SECTOR

0	0	39,590	9AA6	38,995	9853	38,400	9600	37,805	93AD
594	252	40,184	9CF8	39,589	9AA5	38,994	9852	38,399	95FF
0	0	39,590	9AA6	38,995	9853	38,400	9600	37,805	93AD
256	100	39,846	9BA6	39,251	9953	38,656	9700	38,061	94AD
512	200	40,102	9CA6	39,507	9A53	38,912	9800	38,317	95AD
557	22D	40,147	9CD3	39,552	9A80	38,957	982D	38,362	95DA
587	24B	40,177	9CF1	39,582	9A9E	38,987	984B	38,392	95F8
589	24D	40,179	9CF3	39,584	9AA0	38,989	984D	38,394	95FA
591	24F	40,181	9CF5	39,586	9AA2	38,991	984F	38,396	95FC
593	251	40,183	9CF7	39,588	9AA4	38,993	9851	38,398	95FE

SECTION 2: TRACK/SECTOR LIST BUFFERS

0	0	39,846	9BA6	39,251	9953	38,656	9700	38,061	94AD
1	1	39,847	9BA7	39,252	9954	38,657	9701	38,062	94AE
2	2	39,848	9BA8	39,253	9955	38,658	9702	38,063	94AF
3	3	39,849	9BA9	39,254	9956	38,659	9703	38,064	94B0
5	5	39,851	9BAB	39,256	9958	38,661	9705	38,066	94B2
7	7	39,853	9BAD	39,258	995A	38,663	9707	38,068	94B4
12	C	39,858	9BB2	39,263	995F	38,668	970C	38,073	94B9
14	E	39,860	9BB4	39,265	9961	38,670	970E	38,075	94BB
16	10	39,862	9BB6	39,267	9963	38,672	9710	38,077	94BD
18	12	39,864	9BB8	39,269	9965	38,674	9712	38,079	94BF
20	14	39,866	9BBA	39,271	9967	38,676	9714	38,081	94C1
22	16	39,868	9BBC	39,273	9969	38,678	9716	38,083	94C3
24	18	39,870	9BBE	39,275	996B	38,680	9718	38,085	94C5
26	1A	39,872	9BC0	39,277	996D	38,682	971A	38,087	94C7
28	1C	39,874	9BC2	39,279	996F	38,684	971C	38,089	94C9
30	1E	39,876	9BC4	39,281	9971	38,686	971E	38,091	94CB
32	20	39,878	9BC6	39,283	9973	38,688	9720	38,093	94CD
34	22	39,880	9BC8	39,285	9975	38,690	9722	38,095	94CF
36	24	39,882	9BCA	39,287	9977	38,692	9724	38,097	94D1
38	26	39,884	9BCC	39,289	9979	38,694	9726	38,099	94D3
40	28	39,886	9BCE	39,291	997B	38,696	9728	38,101	94D5
42	2A	39,888	9BD0	39,293	997D	38,698	972A	38,103	94D7
44	2C	39,890	9BD2	39,295	997F	38,700	972C	38,105	94D9
46	2E	39,892	9BD4	39,297	9981	38,702	972E	38,107	94DB
48	30	39,894	9BD6	39,299	9983	38,704	9730	38,109	94DD
50	32	39,896	9BD8	39,301	9985	38,706	9732	38,111	94DF
52	34	39,898	9BDA	39,303	9987	38,708	9734	38,113	94E1
54	36	39,900	9BDC	39,305	9989	38,710	9736	38,115	94E3
56	38	39,902	9BDE	39,307	998B	38,712	9738	38,117	94E5
58	3A	39,904	9BE0	39,309	998D	38,714	973A	38,119	94E7

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #1 (IN COM. MEMORY)		DOS BUFFER #2		DOS BUFFER #3		DOS BUFFER #4	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
60	3C	39,906	9BE2	39,311	998F	38,716	973C	38,121	94E9
62	3E	39,908	9BE4	39,313	9991	38,718	973E	38,123	94EB
64	40	39,910	9BE6	39,315	9993	38,720	9740	38,125	94ED
66	42	39,912	9BE8	39,317	9995	38,722	9742	38,127	94EF
68	44	39,914	9BEA	39,319	9997	38,724	9744	38,129	94F1
70	46	39,916	9BEC	39,321	9999	38,726	9746	38,131	94F3
72	48	39,918	9BEE	39,323	999B	38,728	9748	38,133	94F5
74	4A	39,920	9BF0	39,325	999D	38,730	974A	38,135	94F7
76	4C	39,922	9BF2	39,327	999F	38,732	974C	38,137	94F9
78	4E	39,924	9BF4	39,329	99A1	38,734	974E	38,139	94FB
80	50	39,926	9BF6	39,331	99A3	38,736	9750	38,141	94FD
82	52	39,928	9BF8	39,333	99A5	38,738	9752	38,143	94FF
84	54	39,930	9BFA	39,335	99A7	38,740	9754	38,145	9501
86	56	39,932	9BFC	39,337	99A9	38,742	9756	38,147	9503
88	58	39,934	9BFE	39,339	99AB	38,744	9758	38,149	9505
90	5A	39,936	9C00	39,341	99AD	38,746	975A	38,151	9507
92	5C	39,938	9C02	39,343	99AF	38,748	975C	38,153	9509
94	5E	39,940	9C04	39,345	99B1	38,750	975E	38,155	950B
96	60	39,942	9C06	39,347	99B3	38,752	9760	38,157	950D
98	62	39,944	9C08	39,349	99B5	38,754	9762	38,159	950F
100	64	39,946	9C0A	39,351	99B7	38,756	9764	38,161	9511
102	66	39,948	9C0C	39,353	99B9	38,758	9766	38,163	9513
104	68	39,950	9C0E	39,355	99BB	38,760	9768	38,165	9515
106	6A	39,952	9C10	39,357	99BD	38,762	976A	38,167	9517
108	6C	39,954	9C12	39,359	99BF	38,764	976C	38,169	9519
110	6E	39,956	9C14	39,361	99C1	38,766	976E	38,171	951B
112	70	39,958	9C16	39,363	99C3	38,768	9770	38,173	951D
114	72	39,960	9C18	39,365	99C5	38,770	9772	38,175	951F
116	74	39,962	9C1A	39,367	99C7	38,772	9774	38,177	9521
118	76	39,964	9C1C	39,369	99C9	38,774	9776	38,179	9523
120	78	39,966	9C1E	39,371	99CB	38,776	9778	38,181	9525
122	7A	39,968	9C20	39,373	99CD	38,778	977A	38,183	9527
124	7C	39,970	9C22	39,375	99CF	38,780	977C	38,185	9529
126	7E	39,972	9C24	39,377	99D1	38,782	977E	38,187	952B
128	80	39,974	9C26	39,379	99D3	38,784	9780	38,189	952D
130	82	39,976	9C28	39,381	99D5	38,786	9782	38,191	952F
132	84	39,978	9C2A	39,383	99D7	38,788	9784	38,193	9531
134	86	39,980	9C2C	39,385	99D9	38,790	9786	38,195	9533
136	88	39,982	9C2E	39,387	99DB	38,792	9788	38,197	9535
138	8A	39,984	9C30	39,389	99DD	38,794	978A	38,199	9537
140	8C	39,986	9C32	39,391	99DF	38,796	978C	38,201	9539
142	8E	39,988	9C34	39,393	99E1	38,798	978E	38,203	953B
144	90	39,990	9C36	39,395	99E3	38,800	9790	38,205	953D
146	92	39,992	9C38	39,397	99E5	38,802	9792	38,207	953F
148	94	39,994	9C3A	39,399	99E7	38,804	9794	38,209	9541
150	96	39,996	9C3C	39,401	99E9	38,806	9796	38,211	9543
152	98	39,998	9C3E	39,403	99EB	38,808	9798	38,213	9545
154	9A	40,000	9C40	39,405	99ED	38,810	979A	38,215	9547
156	9C	40,002	9C42	39,407	99EF	38,812	979C	38,217	9549

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #1 (IN COM. MEMORY)		DOS BUFFER #2		DOS BUFFER #3		DOS BUFFER #4	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
158	9E	40,004	9C44	39,409	99F1	38,814	979E	38,219	954B
160	A0	40,006	9C46	39,411	99F3	38,816	97A0	38,221	954D
162	A2	40,008	9C48	39,413	99F5	38,818	97A2	38,223	954F
164	A4	40,010	9C4A	39,415	99F7	38,820	97A4	38,225	9551
166	A6	40,012	9C4C	39,417	99F9	38,822	97A6	38,227	9553
168	A8	40,014	9C4E	39,419	99FB	38,824	97A8	38,229	9555
170	AA	40,016	9C50	39,421	99FD	38,826	97AA	38,231	9557
172	AC	40,018	9C52	39,423	99FF	38,828	97AC	38,233	9559
174	AE	40,020	9C54	39,425	9A01	38,830	97AE	38,235	955B
176	B0	40,022	9C56	39,427	9A03	38,832	97B0	38,237	955D
178	B2	40,024	9C58	39,429	9A05	38,834	97B2	38,239	955F
180	B4	40,026	9C5A	39,431	9A07	38,836	97B4	38,241	9561
182	B6	40,028	9C5C	39,433	9A09	38,838	97B6	38,243	9563
184	B8	40,030	9C5E	39,435	9A0B	38,840	97B8	38,245	9565
186	BA	40,032	9C60	39,437	9A0D	38,842	97BA	38,247	9567
188	BC	40,034	9C62	39,439	9A0F	38,844	97BC	38,249	9569
190	BE	40,036	9C64	39,441	9A11	38,846	97BE	38,251	956B
192	C0	40,038	9C66	39,443	9A13	38,848	97C0	38,253	956D
194	C2	40,040	9C68	39,445	9A15	38,850	97C2	38,255	956F
196	C4	40,042	9C6A	39,447	9A17	38,852	97C4	38,257	9571
198	C6	40,044	9C6C	39,449	9A19	38,854	97C6	38,259	9573
200	C8	40,046	9C6E	39,451	9A1B	38,856	97C8	38,261	9575
202	CA	40,048	9C70	39,453	9A1D	38,858	97CA	38,263	9577
204	CC	40,050	9C72	39,455	9A1F	38,860	97CC	38,265	9579
206	CE	40,052	9C74	39,457	9A21	38,862	97CE	38,267	957B
208	D0	40,054	9C76	39,459	9A23	38,864	97D0	38,269	957D
210	D2	40,056	9C78	39,461	9A25	38,866	97D2	38,271	957F
212	D4	40,058	9C7A	39,463	9A27	38,868	97D4	38,273	9581
214	D6	40,060	9C7C	39,465	9A29	38,870	97D6	38,275	9583
216	D8	40,062	9C7E	39,467	9A2B	38,872	97D8	38,277	9585
218	DA	40,064	9C80	39,469	9A2D	38,874	97DA	38,279	9587
220	DC	40,066	9C82	39,471	9A2F	38,876	97DC	38,281	9589
222	DE	40,068	9C84	39,473	9A31	38,878	97DE	38,283	958B
224	E0	40,070	9C86	39,475	9A33	38,880	97E0	38,285	958D
226	E2	40,072	9C88	39,477	9A35	38,882	97E2	38,287	958F
228	E4	40,074	9C8A	39,479	9A37	38,884	97E4	38,289	9591
230	E6	40,076	9C8C	39,481	9A39	38,886	97E6	38,291	9593
232	E8	40,078	9C8E	39,483	9A3B	38,888	97E8	38,293	9595
234	EA	40,080	9C90	39,485	9A3D	38,890	97EA	38,295	9597
236	EC	40,082	9C92	39,487	9A3F	38,892	97EC	38,297	9599
238	EE	40,084	9C94	39,489	9A41	38,894	97EE	38,299	959B
240	F0	40,086	9C96	39,491	9A43	38,896	97F0	38,301	959D
242	F2	40,088	9C98	39,493	9A45	38,898	97F2	38,303	959F
244	F4	40,090	9C9A	39,495	9A47	38,900	97F4	38,305	95A1
246	F6	40,092	9C9C	39,497	9A49	38,902	97F6	38,307	95A3
248	F8	40,094	9C9E	39,499	9A4B	38,904	97F8	38,309	95A5
250	FA	40,096	9CA0	39,501	9A4D	38,906	97FA	38,311	95A7
252	FC	40,098	9CA2	39,503	9A4F	38,908	97FC	38,313	95A9
254	FE	40,100	9CA4	39,505	9A51	38,910	97FE	38,315	95AB

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #1 (IN COM. MEMORY)		DOS BUFFER #2		DOS BUFFER #3		DOS BUFFER #4	
--------------------------------------	--	--------------------------------------	--	------------------	--	------------------	--	------------------	--

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

SECTION 3: FILE MANAGER WORK AREA

0	0	40,102	9CA6	39,507	9A53	38,912	9800	38,317	95AD
2	2	40,104	9CA8	39,509	9A55	38,914	9802	38,319	95AF
4	4	40,106	9CAA	39,511	9A57	38,916	9804	38,321	95B1
5	5	40,107	9CAB	39,512	9A58	38,917	9805	38,322	95B2
7	7	40,109	9CAD	39,514	9A5A	38,919	9807	38,324	95B4
8	8	40,110	9CAE	39,515	9A5B	38,920	9808	38,325	95B5
9	9	40,111	9CAF	39,516	9A5C	38,921	9809	38,326	95B6
11	B	40,113	9CB1	39,518	9A5E	38,923	980B	38,328	95B8
13	D	40,115	9CB3	39,520	9A60	38,925	980D	38,330	95BA
15	F	40,117	9CB5	39,522	9A62	38,927	980F	38,332	95BC
17	11	40,119	9CB7	39,524	9A64	38,929	9811	38,334	95BE
19	13	40,121	9CB9	39,526	9A66	38,931	9813	38,336	95C0
21	15	40,123	9CBB	39,528	9A68	38,933	9815	38,338	95C2
22	16	40,124	9CBC	39,529	9A69	38,934	9816	38,339	95C3
23	17	40,125	9CBD	39,530	9A6A	38,935	9817	38,340	95C4
25	19	40,127	9CBF	39,532	9A6C	38,937	9819	38,342	95C6
27	1B	40,129	9CC1	39,534	9A6E	38,939	981B	38,344	95C8
29	1D	40,131	9CC3	39,536	9A70	38,941	981D	38,346	95CA
31	1F	40,133	9CC5	39,538	9A72	38,943	981F	38,348	95CC
32	20	40,134	9CC6	39,539	9A73	38,944	9820	38,349	95CD
33	21	40,135	9CC7	39,540	9A74	38,945	9821	38,350	95CE
37	25	40,139	9CCB	39,544	9A78	38,949	9825	38,354	95D2
38	26	40,140	9CCC	39,545	9A79	38,950	9826	38,355	95D3
39	27	40,141	9CCD	39,546	9A7A	38,951	9827	38,356	95D4
40	28	40,142	9CCE	39,547	9A7B	38,952	9828	38,357	95D5
41	29	40,143	9CCF	39,548	9A7C	38,953	9829	38,358	95D6
42	2A	40,144	9CD0	39,549	9A7D	38,954	982A	38,359	95D7

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #5		DOS BUFFER #6		DOS BUFFER #7		DOS BUFFER #8	
--------------------------------------	--	------------------	--	------------------	--	------------------	--	------------------	--

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

SECTION 1: DATA SECTOR

0	0	37,210	915A	36,615	8F07	36,020	8CB4	35,425	8A61
594	252	37,804	93AC	37,209	9159	36,614	8F06	36,019	8CB3
0	0	37,210	915A	36,615	8F07	36,020	8CB4	35,425	8A61
256	100	37,466	925A	36,871	9007	36,276	8DB4	35,681	8B61
512	200	37,722	935A	37,127	9107	36,532	8EB4	35,937	8C61
557	22D	37,767	9387	37,172	9134	36,577	8EE1	35,982	8C8E
587	24B	37,797	93A5	37,202	9152	36,607	8EFF	36,012	8CAC
589	24D	37,799	93A7	37,204	9154	36,609	8F01	36,014	8CAE
591	24F	37,801	93A9	37,206	9156	36,611	8F03	36,016	8CB0
593	251	37,803	93AB	37,208	9158	36,613	8F05	36,018	8CB2

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #5		DOS BUFFER #6		DOS BUFFER #7		DOS BUFFER #8	
--------------------------------------	--	------------------	--	------------------	--	------------------	--	------------------	--

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

SECTION #2: TRACK/SECTOR LIST BUFFER

0	0	37,466	925A	36,871	9007	36,276	8DB4	35,681	8B61
1	1	37,467	925B	36,872	9008	36,277	8DB5	35,682	8B62
2	2	37,468	925C	36,873	9009	36,278	8DB6	35,683	8B63
3	3	37,469	925D	36,874	900A	36,279	8DB7	35,684	8B64
5	5	37,471	925F	36,876	900C	36,281	8DB9	35,686	8B66
7	7	37,473	9261	36,878	900E	36,283	8DBB	35,688	8B68
12	C	37,478	9266	36,883	9013	36,288	8DC0	35,693	8B6D
14	E	37,480	9268	36,885	9015	36,290	8DC2	35,695	8B6F
16	10	37,482	926A	36,887	9017	36,292	8DC4	35,697	8B71
18	12	37,484	926C	36,889	9019	36,294	8DC6	35,699	8B73
20	14	37,486	926E	36,891	901B	36,296	8DC8	35,701	8B75
22	16	37,488	9270	36,893	901D	36,298	8DCA	35,703	8B77
24	18	37,490	9272	36,895	901F	36,300	8DCC	35,705	8B79
26	1A	37,492	9274	36,897	9021	36,302	8DCE	35,707	8B7B
28	1C	37,494	9276	36,899	9023	36,304	8DD0	35,709	8B7D
30	1E	37,496	9278	36,901	9025	36,306	8DD2	35,711	8B7F
32	20	37,498	927A	36,903	9027	36,308	8DD4	35,713	8B81
34	22	37,500	927C	36,905	9029	36,310	8DD6	35,715	8B83
36	24	37,502	927E	36,907	902B	36,312	8DD8	35,717	8B85
38	26	37,504	9280	36,909	902D	36,314	8DDA	35,719	8B87
40	28	37,506	9282	36,911	902F	36,316	8DDC	35,721	8B89
42	2A	37,508	9284	36,913	9031	36,318	8DDE	35,723	8B8B
44	2C	37,510	9286	36,915	9033	36,320	8DE0	35,725	8B8D
46	2E	37,512	9288	36,917	9035	36,322	8DE2	35,727	8B8F
48	30	37,514	928A	36,919	9037	36,324	8DE4	35,729	8B91
50	32	37,516	928C	36,921	9039	36,326	8DE6	35,731	8B93
52	34	37,518	928E	36,923	903B	36,328	8DE8	35,733	8B95
54	36	37,520	9290	36,925	903D	36,330	8DEA	35,735	8B97
56	38	37,522	9292	36,927	903F	36,332	8DEC	35,737	8B99
58	3A	37,524	9294	36,929	9041	36,334	8DEE	35,739	8B9B
60	3C	37,526	9296	36,931	9043	36,336	8DF0	35,741	8B9D
62	3E	37,528	9298	36,933	9045	36,338	8DF2	35,743	8B9F
64	40	37,530	929A	36,935	9047	36,340	8DF4	35,745	8BA1
66	42	37,532	929C	36,937	9049	36,342	8DF6	35,747	8BA3
68	44	37,534	929E	36,939	904B	36,344	8DF8	35,749	8BA5
70	46	37,536	92A0	36,941	904D	36,346	8DFA	35,751	8BA7
72	48	37,538	92A2	36,943	904F	36,348	8DFC	35,753	8BA9
74	4A	37,540	92A4	36,945	9051	36,350	8DFE	35,755	8BAB
76	4C	37,542	92A6	36,947	9053	36,352	8E00	35,757	8BAD
78	4E	37,544	92A8	36,949	9055	36,354	8E02	35,759	8BAF
80	50	37,546	92AA	36,951	9057	36,356	8E04	35,761	8BB1
82	52	37,548	92AC	36,953	9059	36,358	8E06	35,763	8BB3
84	54	37,550	92AE	36,955	905B	36,360	8E08	35,765	8BB5
86	56	37,552	92B0	36,957	905D	36,362	8E0A	35,767	8BB7
88	58	37,554	92B2	36,959	905F	36,364	8E0C	35,769	8BB9
90	5A	37,556	92B4	36,961	9061	36,366	8E0E	35,771	8BBB
92	5C	37,558	92B6	36,963	9063	36,368	8E10	35,773	8BBD
94	5E	37,560	92B8	36,965	9065	36,370	8E12	35,775	8BBF
96	60	37,562	92BA	36,967	9067	36,372	8E14	35,777	8BC1
98	62	37,564	92BC	36,969	9069	36,374	8E16	35,779	8BC3

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #5		DOS BUFFER #6		DOS BUFFER #7		DOS BUFFER #8	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
100	64	37,566	92BE	36,971	906B	36,376	8E18	35,781	8BC5
102	66	37,568	92C0	36,973	906D	36,378	8E1A	35,783	8BC7
104	68	37,570	92C2	36,975	906F	36,380	8E1C	35,785	8BC9
106	6A	37,572	92C4	36,977	9071	36,382	8E1E	35,787	8BCB
108	6C	37,574	92C6	36,979	9073	36,384	8E20	35,789	8BCD
110	6E	37,576	92C8	36,981	9075	36,386	8E22	35,791	8BCF
112	70	37,578	92CA	36,983	9077	36,388	8E24	35,793	8BD1
114	72	37,580	92CC	36,985	9079	36,390	8E26	35,795	8BD3
116	74	37,582	92CE	36,987	907B	36,392	8E28	35,797	8BD5
118	76	37,584	92D0	36,989	907D	36,394	8E2A	35,799	8BD7
120	78	37,586	92D2	36,991	907F	36,396	8E2C	35,801	8BD9
122	7A	37,588	92D4	36,993	9081	36,398	8E2E	35,803	8BDB
124	7C	37,590	92D6	36,995	9083	36,400	8E30	35,805	8BDD
126	7E	37,592	92D8	36,997	9085	36,402	8E32	35,807	8BDF
128	80	37,594	92DA	36,999	9087	36,404	8E34	35,809	8BE1
130	82	37,596	92DC	37,001	9089	36,406	8E36	35,811	8BE3
132	84	37,598	92DE	37,003	908B	36,408	8E38	35,813	8BE5
134	86	37,600	92E0	37,005	908D	36,410	8E3A	35,815	8BE7
136	88	37,602	92E2	37,007	908F	36,412	8E3C	35,817	8BE9
138	8A	37,604	92E4	37,009	9091	36,414	8E3E	35,819	8BEB
140	8C	37,606	92E6	37,011	9093	36,416	8E40	35,821	8BED
142	8E	37,608	92E8	37,013	9095	36,418	8E42	35,823	8BEF
144	90	37,610	92EA	37,015	9097	36,420	8E44	35,825	8BF1
146	92	37,612	92EC	37,017	9099	36,422	8E46	35,827	8BF3
148	94	37,614	92EE	37,019	909B	36,424	8E48	35,829	8BF5
150	96	37,616	92F0	37,021	909D	36,426	8E4A	35,831	8BF7
152	98	37,618	92F2	37,023	909F	36,428	8E4C	35,833	8BF9
154	9A	37,620	92F4	37,025	90A1	36,430	8E4E	35,835	8BFB
156	9C	37,622	92F6	37,027	90A3	36,432	8E50	35,837	8BFD
158	9E	37,624	92F8	37,029	90A5	36,434	8E52	35,839	8BFF
160	A0	37,626	92FA	37,031	90A7	36,436	8E54	35,841	8C01
162	A2	37,628	92FC	37,033	90A9	36,438	8E56	35,843	8C03
164	A4	37,630	92FE	37,035	90AB	36,440	8E58	35,845	8C05
166	A6	37,632	9300	37,037	90AD	36,442	8E5A	35,847	8C07
168	A8	37,634	9302	37,039	90AF	36,444	8E5C	35,849	8C09
170	AA	37,636	9304	37,041	90B1	36,446	8E5E	35,851	8C0B
172	AC	37,638	9306	37,043	90B3	36,448	8E60	35,853	8C0D
174	AE	37,640	9308	37,045	90B5	36,450	8E62	35,855	8C0F
176	B0	37,642	930A	37,047	90B7	36,452	8E64	35,857	8C11
178	B2	37,644	930C	37,049	90B9	36,454	8E66	35,859	8C13
180	B4	37,646	930E	37,051	90BB	36,456	8E68	35,861	8C15
182	B6	37,648	9310	37,053	90BD	36,458	8E6A	35,863	8C17
184	B8	37,650	9312	37,055	90BF	36,460	8E6C	35,865	8C19
186	BA	37,652	9314	37,057	90C1	36,462	8E6E	35,867	8C1B
188	BC	37,654	9316	37,059	90C3	36,464	8E70	35,869	8C1D
190	BE	37,656	9318	37,061	90C5	36,466	8E72	35,871	8C1F
192	C0	37,658	931A	37,063	90C7	36,468	8E74	35,873	8C21
194	C2	37,660	931C	37,065	90C9	36,470	8E76	35,875	8C23
196	C4	37,662	931E	37,067	90CB	36,472	8E78	35,877	8C25
198	C6	37,664	9320	37,069	90CD	36,474	8E7A	35,879	8C27

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #5		DOS BUFFER #6		DOS BUFFER #7		DOS BUFFER #8	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
200	C8	37,666	9322	37,071	90CF	36,476	8E7C	35,881	8C29
202	CA	37,668	9324	37,073	90D1	36,478	8E7E	35,883	8C2B
204	CC	37,670	9326	37,075	90D3	36,480	8E80	35,885	8C2D
206	CE	37,672	9328	37,077	90D5	36,482	8E82	35,887	8C2F
208	D0	37,674	932A	37,079	90D7	36,484	8E84	35,889	8C31
210	D2	37,676	932C	37,081	90D9	36,486	8E86	35,891	8C33
212	D4	37,678	932E	37,083	90DB	36,488	8E88	35,893	8C35
214	D6	37,680	9330	37,085	90DD	36,490	8E8A	35,895	8C37
216	D8	37,682	9332	37,087	90DF	36,492	8E8C	35,897	8C39
218	DA	37,684	9334	37,089	90E1	36,494	8E8E	35,899	8C3B
220	DC	37,686	9336	37,091	90E3	36,496	8E90	35,901	8C3D
222	DE	37,688	9338	37,093	90E5	36,498	8E92	35,903	8C3F
224	E0	37,690	933A	37,095	90E7	36,500	8E94	35,905	8C41
226	E2	37,692	933C	37,097	90E9	36,502	8E96	35,907	8C43
228	E4	37,694	933E	37,099	90EB	36,504	8E98	35,909	8C45
230	E6	37,696	9340	37,101	90ED	36,506	8E9A	35,911	8C47
232	E8	37,698	9342	37,103	90EF	36,508	8E9C	35,913	8C49
234	EA	37,700	9344	37,105	90F1	36,510	8E9E	35,915	8C4B
236	EC	37,702	9346	37,107	90F3	36,512	8EA0	35,917	8C4D
238	EE	37,704	9348	37,109	90F5	36,514	8EA2	35,919	8C4F
240	F0	37,706	934A	37,111	90F7	36,516	8EA4	35,921	8C51
242	F2	37,708	934C	37,113	90F9	36,518	8EA6	35,923	8C53
244	F4	37,710	934E	37,115	90FB	36,520	8EA8	35,925	8C55
246	F6	37,712	9350	37,117	90FD	36,522	8EAA	35,927	8C57
248	F8	37,714	9352	37,119	90FF	36,524	8EAC	35,929	8C59
250	FA	37,716	9354	37,121	9101	36,526	8EAE	35,931	8C5B
252	FC	37,718	9356	37,123	9103	36,528	8EB0	35,933	8C5D
254	FE	37,720	9358	37,125	9105	36,530	8EB2	35,935	8C5F

SECTION 3: FILE MANAGER WORK AREA

0	0	37,722	935A	37,127	9107	36,532	8EB4	35,937	8C61
2	2	37,724	935C	37,129	9109	36,534	8EB6	35,939	8C63
4	4	37,726	935E	37,131	910B	36,536	8EB8	35,941	8C65
5	5	37,727	935F	37,132	910C	36,537	8EB9	35,942	8C66
7	7	37,729	9361	37,134	910E	36,539	8EBB	35,944	8C68
8	8	37,730	9362	37,135	910F	36,540	8EBC	35,945	8C69
9	9	37,731	9363	37,136	9110	36,541	8EBD	35,946	8C6A
11	B	37,733	9365	37,138	9112	36,543	8EBF	35,948	8C6C
13	D	37,735	9367	37,140	9114	36,545	8EC1	35,950	8C6E
15	F	37,737	9369	37,142	9116	36,547	8EC3	35,952	8C70
17	11	37,739	936B	37,144	9118	36,549	8EC5	35,954	8C72
19	13	37,741	936D	37,146	911A	36,551	8EC7	35,956	8C74
21	15	37,743	936F	37,148	911C	36,553	8EC9	35,958	8C76
22	16	37,744	9370	37,149	911D	36,554	8ECA	35,959	8C77

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #5		DOS BUFFER #6		DOS BUFFER #7		DOS BUFFER #8	
--------------------------------------	--	------------------	--	------------------	--	------------------	--	------------------	--

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
23	17	37,745	9371	37,150	911E	36,555	8ECB	35,960	8C78
25	19	37,747	9373	37,152	9120	36,557	8ECD	35,962	8C7A
27	1B	37,749	9375	37,154	9122	36,559	8ECF	35,964	8C7C
29	1D	37,751	9377	37,156	9124	36,561	8ED1	35,966	8C7E
31	1F	37,753	9379	37,158	9126	36,563	8ED3	35,968	8C80
32	20	37,754	937A	37,159	9127	36,564	8ED4	35,969	8C81
33	21	37,755	937B	37,160	9128	36,565	8ED5	35,970	8C82
37	25	37,759	937F	37,164	912C	36,569	8ED9	35,974	8C86
38	26	37,760	9380	37,165	912D	36,570	8EDA	35,975	8C87
39	27	37,761	9381	37,166	912E	36,571	8EDB	35,976	8C88
40	28	37,762	9382	37,167	912F	36,572	8EDC	35,977	8C89
41	29	37,763	9383	37,168	9130	36,573	8EDD	35,978	8C8A
42	2A	37,764	9384	37,169	9131	36,574	8EDE	35,979	8C8B

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #9		DOS BUFFER #10		DOS BUFFER #11		DOS BUFFER #12	
--------------------------------------	--	------------------	--	-------------------	--	-------------------	--	-------------------	--

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

SECTION 1: DATA SECTOR

0	0	34,830	880E	34,235	85BB	33,640	8368	33,045	8115
594	252	35,424	8A60	34,829	880D	34,234	85BA	33,639	8367

0	0	34,830	880E	34,235	85BB	33,640	8368	33,045	8115
256	100	35,086	890E	34,491	86BB	33,896	8468	33,301	8215
512	200	35,342	8A0E	34,747	87BB	34,152	8568	33,557	8315
557	22D	35,387	8A3B	34,792	87E8	34,197	8595	33,602	8342
587	24B	35,417	8A59	34,822	8806	34,227	85B3	33,632	8360
589	24D	35,419	8A5B	34,824	8808	34,229	85B5	33,634	8362
591	24F	35,421	8A5D	34,826	880A	34,231	85B7	33,636	8364
593	251	35,423	8A5F	34,828	880C	34,233	85B9	33,638	8366

SECTION #2: TRACK/SECTOR LIST BUFFER

0	0	35,086	890E	34,491	86BB	33,896	8468	33,301	8215
1	1	35,087	890F	34,492	86BC	33,897	8469	33,302	8216
2	2	35,088	8910	34,493	86BD	33,898	846A	33,303	8217
3	3	35,089	8911	34,494	86BE	33,899	846B	33,304	8218
5	5	35,091	8913	34,496	86C0	33,901	846D	33,306	821A
7	7	35,093	8915	34,498	86C2	33,903	846F	33,308	821C
12	C	35,098	891A	34,503	86C7	33,908	8474	33,313	8221
14	E	35,100	891C	34,505	86C9	33,910	8476	33,315	8223

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #9		DOS BUFFER #10		DOS BUFFER #11		DOS BUFFER #12	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
16	10	35,102	891E	34,507	86CB	33,912	8478	33,317	8225
18	12	35,104	8920	34,509	86CD	33,914	847A	33,319	8227
20	14	35,106	8922	34,511	86CF	33,916	847C	33,321	8229
22	16	35,108	8924	34,513	86D1	33,918	847E	33,323	822B
24	18	35,110	8926	34,515	86D3	33,920	8480	33,325	822D
26	1A	35,112	8928	34,517	86D5	33,922	8482	33,327	822F
28	1C	35,114	892A	34,519	86D7	33,924	8484	33,329	8231
30	1E	35,116	892C	34,521	86D9	33,926	8486	33,331	8233
32	20	35,118	892E	34,523	86DB	33,928	8488	33,333	8235
34	22	35,120	8930	34,525	86DD	33,930	848A	33,335	8237
36	24	35,122	8932	34,527	86DF	33,932	848C	33,337	8239
38	26	35,124	8934	34,529	86E1	33,934	848E	33,339	823B
40	28	35,126	8936	34,531	86E3	33,936	8490	33,341	823D
42	2A	35,128	8938	34,533	86E5	33,938	8492	33,343	823F
44	2C	35,130	893A	34,535	86E7	33,940	8494	33,345	8241
46	2E	35,132	893C	34,537	86E9	33,942	8496	33,347	8243
48	30	35,134	893E	34,539	86EB	33,944	8498	33,349	8245
50	32	35,136	8940	34,541	86ED	33,946	849A	33,351	8247
52	34	35,138	8942	34,543	86EF	33,948	849C	33,353	8249
54	36	35,140	8944	34,545	86F1	33,950	849E	33,355	824B
56	38	35,142	8946	34,547	86F3	33,952	84A0	33,357	824D
58	3A	35,144	8948	34,549	86F5	33,954	84A2	33,359	824F
60	3C	35,146	894A	34,551	86F7	33,956	84A4	33,361	8251
62	3E	35,148	894C	34,553	86F9	33,958	84A6	33,363	8253
64	40	35,150	894E	34,555	86FB	33,960	84A8	33,365	8255
66	42	35,152	8950	34,557	86FD	33,962	84AA	33,367	8257
68	44	35,154	8952	34,559	86FF	33,964	84AC	33,369	8259
70	46	35,156	8954	34,561	8701	33,966	84AE	33,371	825B
72	48	35,158	8956	34,563	8703	33,968	84B0	33,373	825D
74	4A	35,160	8958	34,565	8705	33,970	84B2	33,375	825F
76	4C	35,162	895A	34,567	8707	33,972	84B4	33,377	8261
78	4E	35,164	895C	34,569	8709	33,974	84B6	33,379	8263
80	50	35,166	895E	34,571	870B	33,976	84B8	33,381	8265
82	52	35,168	8960	34,573	870D	33,978	84BA	33,383	8267
84	54	35,170	8962	34,575	870F	33,980	84BC	33,385	8269
86	56	35,172	8964	34,577	8711	33,982	84BE	33,387	826B
88	58	35,174	8966	34,579	8713	33,984	84C0	33,389	826D
90	5A	35,176	8968	34,581	8715	33,986	84C2	33,391	826F
92	5C	35,178	896A	34,583	8717	33,988	84C4	33,393	8271
94	5E	35,180	896C	34,585	8719	33,990	84C6	33,395	8273
96	60	35,182	896E	34,587	871B	33,992	84C8	33,397	8275
98	62	35,184	8970	34,589	871D	33,994	84CA	33,399	8277
100	64	35,186	8972	34,591	871F	33,996	84CC	33,401	8279
102	66	35,188	8974	34,593	8721	33,998	84CE	33,403	827B
104	68	35,190	8976	34,595	8723	34,000	84D0	33,405	827D
106	6A	35,192	8978	34,597	8725	34,002	84D2	33,407	827F
108	6C	35,194	897A	34,599	8727	34,004	84D4	33,409	8281
110	6E	35,196	897C	34,601	8729	34,006	84D6	33,411	8283
112	70	35,198	897E	34,603	872B	34,008	84D8	33,413	8285

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #9		DOS BUFFER #10		DOS BUFFER #11		DOS BUFFER #12	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
114	72	35,200	8980	34,605	872D	34,010	84DA	33,415	8287
116	74	35,202	8982	34,607	872F	34,012	84DC	33,417	8289
118	76	35,204	8984	34,609	8731	34,014	84DE	33,419	828B
120	78	35,206	8986	34,611	8733	34,016	84E0	33,421	828D
122	7A	35,208	8988	34,613	8735	34,018	84E2	33,423	828F
124	7C	35,210	898A	34,615	8737	34,020	84E4	33,425	8291
126	7E	35,212	898C	34,617	8739	34,022	84E6	33,427	8293
128	80	35,214	898E	34,619	873B	34,024	84E8	33,429	8295
130	82	35,216	8990	34,621	873D	34,026	84EA	33,431	8297
132	84	35,218	8992	34,623	873F	34,028	84EC	33,433	8299
134	86	35,220	8994	34,625	8741	34,030	84EE	33,435	829B
136	88	35,222	8996	34,627	8743	34,032	84F0	33,437	829D
138	8A	35,224	8998	34,629	8745	34,034	84F2	33,439	829F
140	8C	35,226	899A	34,631	8747	34,036	84F4	33,441	82A1
142	8E	35,228	899C	34,633	8749	34,038	84F6	33,443	82A3
144	90	35,230	899E	34,635	874B	34,040	84F8	33,445	82A5
146	92	35,232	89A0	34,637	874D	34,042	84FA	33,447	82A7
148	94	35,234	89A2	34,639	874F	34,044	84FC	33,449	82A9
150	96	35,236	89A4	34,641	8751	34,046	84FE	33,451	82AB
152	98	35,238	89A6	34,643	8753	34,048	8500	33,453	82AD
154	9A	35,240	89A8	34,645	8755	34,050	8502	33,455	82AF
156	9C	35,242	89AA	34,647	8757	34,052	8504	33,457	82B1
158	9E	35,244	89AC	34,649	8759	34,054	8506	33,459	82B3
160	A0	35,246	89AE	34,651	875B	34,056	8508	33,461	82B5
162	A2	35,248	89B0	34,653	875D	34,058	850A	33,463	82B7
164	A4	35,250	89B2	34,655	875F	34,060	850C	33,465	82B9
166	A6	35,252	89B4	34,657	8761	34,062	850E	33,467	82BB
168	A8	35,254	89B6	34,659	8763	34,064	8510	33,469	82BD
170	AA	35,256	89B8	34,661	8765	34,066	8512	33,471	82BF
172	AC	35,258	89BA	34,663	8767	34,068	8514	33,473	82C1
174	AE	35,260	89BC	34,665	8769	34,070	8516	33,475	82C3
176	B0	35,262	89BE	34,667	876B	34,072	8518	33,477	82C5
178	B2	35,264	89C0	34,669	876D	34,074	851A	33,479	82C7
180	B4	35,266	89C2	34,671	876F	34,076	851C	33,481	82C9
182	B6	35,268	89C4	34,673	8771	34,078	851E	33,483	82CB
184	B8	35,270	89C6	34,675	8773	34,080	8520	33,485	82CD
186	BA	35,272	89C8	34,677	8775	34,082	8522	33,487	82CF
188	BC	35,274	89CA	34,679	8777	34,084	8524	33,489	82D1
190	BE	35,276	89CC	34,681	8779	34,086	8526	33,491	82D3
192	C0	35,278	89CE	34,683	877B	34,088	8528	33,493	82D5
194	C2	35,280	89D0	34,685	877D	34,090	852A	33,495	82D7
196	C4	35,282	89D2	34,687	877F	34,092	852C	33,497	82D9
198	C6	35,284	89D4	34,689	8781	34,094	852E	33,499	82DB
200	C8	35,286	89D6	34,691	8783	34,096	8530	33,501	82DD
202	CA	35,288	89D8	34,693	8785	34,098	8532	33,503	82DF
204	CC	35,290	89DA	34,695	8787	34,100	8534	33,505	82E1
206	CE	35,292	89DC	34,697	8789	34,102	8536	33,507	82E3
208	D0	35,294	89DE	34,699	878B	34,104	8538	33,509	82E5
210	D2	35,296	89E0	34,701	878D	34,106	853A	33,511	82E7
212	D4	35,298	89E2	34,703	878F	34,108	853C	33,513	82E9

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #9		DOS BUFFER #10		DOS BUFFER #11		DOS BUFFER #12	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
214	D6	35,300	89E4	34,705	8791	34,110	853E	33,515	82EB
216	D8	35,302	89E6	34,707	8793	34,112	8540	33,517	82ED
218	DA	35,304	89E8	34,709	8795	34,114	8542	33,519	82EF
220	DC	35,306	89EA	34,711	8797	34,116	8544	33,521	82F1
222	DE	35,308	89EC	34,713	8799	34,118	8546	33,523	82F3
224	E0	35,310	89EE	34,715	879B	34,120	8548	33,525	82F5
226	E2	35,312	89F0	34,717	879D	34,122	854A	33,527	82F7
228	E4	35,314	89F2	34,719	879F	34,124	854C	33,529	82F9
230	E6	35,316	89F4	34,721	87A1	34,126	854E	33,531	82FB
232	E8	35,318	89F6	34,723	87A3	34,128	8550	33,533	82FD
234	EA	35,320	89F8	34,725	87A5	34,130	8552	33,535	82FF
236	EC	35,322	89FA	34,727	87A7	34,132	8554	33,537	8301
238	EE	35,324	89FC	34,729	87A9	34,134	8556	33,539	8303
240	F0	35,326	89FE	34,731	87AB	34,136	8558	33,541	8305
242	F2	35,328	8A00	34,733	87AD	34,138	855A	33,543	8307
244	F4	35,330	8A02	34,735	87AF	34,140	855C	33,545	8309
246	F6	35,332	8A04	34,737	87B1	34,142	855E	33,547	830B
248	F8	35,334	8A06	34,739	87B3	34,144	8560	33,549	830D
250	FA	35,336	8A08	34,741	87B5	34,146	8562	33,551	830F
252	FC	35,338	8A0A	34,743	87B7	34,148	8564	33,553	8311
254	FE	35,340	8A0C	34,745	87B9	34,150	8566	33,555	8313

SECTION 3: FILE MANAGER WORK AREA

0	0	35,342	8A0E	34,747	87BB	34,152	8568	33,557	8315
2	2	35,344	8A10	34,749	87BD	34,154	856A	33,559	8317
4	4	35,346	8A12	34,751	87BF	34,156	856C	33,561	8319
5	5	35,347	8A13	34,752	87C0	34,157	856D	33,562	831A
7	7	35,349	8A15	34,754	87C2	34,159	856F	33,564	831C
8	8	35,350	8A16	34,755	87C3	34,160	8570	33,565	831D
9	9	35,351	8A17	34,756	87C4	34,161	8571	33,566	831E
11	B	35,353	8A19	34,758	87C6	34,163	8573	33,568	8320
13	D	35,355	8A1B	34,760	87C8	34,165	8575	33,570	8322
15	F	35,357	8A1D	34,762	87CA	34,167	8577	33,572	8324
17	11	35,359	8A1F	34,764	87CC	34,169	8579	33,574	8326
19	13	35,361	8A21	34,766	87CE	34,171	857B	33,576	8328
21	15	35,363	8A23	34,768	87D0	34,173	857D	33,578	832A
22	16	35,364	8A24	34,769	87D1	34,174	857E	33,579	832B
23	17	35,365	8A25	34,770	87D2	34,175	857F	33,580	832C
25	19	35,367	8A27	34,772	87D4	34,177	8581	33,582	832E
27	1B	35,369	8A29	34,774	87D6	34,179	8583	33,584	8330
29	1D	35,371	8A2B	34,776	87D8	34,181	8585	33,586	8332
31	1F	35,373	8A2D	34,778	87DA	34,183	8587	33,588	8334

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #9		DOS BUFFER #10		DOS BUFFER #11		DOS BUFFER #12	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
32	20	35,374	8A2E	34,779	87DB	34,184	8588	33,589	8335
33	21	35,375	8A2F	34,780	87DC	34,185	8589	33,590	8336
37	25	35,379	8A33	34,784	87E0	34,189	858D	33,594	833A
38	26	35,380	8A34	34,785	87E1	34,190	858E	33,595	833B
39	27	35,381	8A35	34,786	87E2	34,191	858F	33,596	833C
40	28	35,382	8A36	34,787	87E3	34,192	8590	33,597	833D
41	29	35,383	8A37	34,788	87E4	34,193	8591	33,598	833E
42	2A	35,384	8A38	34,789	87E5	34,194	8592	33,599	833F

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #13		DOS BUFFER #14		DOS BUFFER #15		DOS BUFFER #16	
DEX	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX

SECTION 1: DATA SECTOR

0	0	32,450	7EC2	31,855	7C6F	31,260	7A1C	30,665	77C9
594	252	33,044	8114	32,449	7EC1	31,854	7C6E	31,259	7A1B
0	0	32,450	7EC2	31,855	7C6F	31,260	7A1C	30,665	77C9
256	100	32,706	7FC2	32,111	7D6F	31,516	7B1C	30,921	78C9
512	200	32,962	80C2	32,367	7E6F	31,772	7C1C	31,177	79C9
557	22D	33,007	80EF	32,412	7E9C	31,817	7C49	31,222	79F6
587	24B	33,037	810D	32,442	7EBA	31,847	7C67	31,252	7A14
589	24D	33,039	810F	32,444	7EBC	31,849	7C69	31,254	7A16
591	24F	33,041	8111	32,446	7EBE	31,851	7C6B	31,256	7A18
593	251	33,043	8113	32,448	7EC0	31,853	7C6D	31,258	7A1A

SECTION #2: TRACK/SECTOR LIST BUFFER

0	0	32,706	7FC2	32,111	7D6F	31,516	7B1C	30,921	78C9
1	1	32,707	7FC3	32,112	7D70	31,517	7B1D	30,922	78CA
2	2	32,708	7FC4	32,113	7D71	31,518	7B1E	30,923	78CB
3	3	32,709	7FC5	32,114	7D72	31,519	7B1F	30,924	78CC
5	5	32,711	7FC7	32,116	7D74	31,521	7B21	30,926	78CE
7	7	32,713	7FC9	32,118	7D76	31,523	7B23	30,928	78D0
12	C	32,718	7FCE	32,123	7D7B	31,528	7B28	30,933	78D5
14	E	32,720	7FD0	32,125	7D7D	31,530	7B2A	30,935	78D7
16	10	32,722	7FD2	32,127	7D7F	31,532	7B2C	30,937	78D9
18	12	32,724	7FD4	32,129	7D81	31,534	7B2E	30,939	78DB
20	14	32,726	7FD6	32,131	7D83	31,536	7B30	30,941	78DD
22	16	32,728	7FD8	32,133	7D85	31,538	7B32	30,943	78DF
24	18	32,730	7FDA	32,135	7D87	31,540	7B34	30,945	78E1

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #13		DOS BUFFER #14		DOS BUFFER #15		DOS BUFFER #16	
DEX	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
26	1A	32,732	7FDC	32,137	7D89	31,542	7B36	30,947	78E3
28	1C	32,734	7FDE	32,139	7D8B	31,544	7B38	30,949	78E5
30	1E	32,736	7FE0	32,141	7D8D	31,546	7B3A	30,951	78E7
32	20	32,738	7FE2	32,143	7D8F	31,548	7B3C	30,953	78E9
34	22	32,740	7FE4	32,145	7D91	31,550	7B3E	30,955	78EB
36	24	32,742	7FE6	32,147	7D93	31,552	7B40	30,957	78ED
38	26	32,744	7FE8	32,149	7D95	31,554	7B42	30,959	78EF
40	28	32,746	7FEA	32,151	7D97	31,556	7B44	30,961	78F1
42	2A	32,748	7FEC	32,153	7D99	31,558	7B46	30,963	78F3
44	2C	32,750	7FEE	32,155	7D9B	31,560	7B48	30,965	78F5
46	2E	32,752	7FF0	32,157	7D9D	31,562	7B4A	30,967	78F7
48	30	32,754	7FF2	32,159	7D9F	31,564	7B4C	30,969	78F9
50	32	32,756	7FF4	32,161	7DA1	31,566	7B4E	30,971	78FB
52	34	32,758	7FF6	32,163	7DA3	31,568	7B50	30,973	78FD
54	36	32,760	7FF8	32,165	7DA5	31,570	7B52	30,975	78FF
56	38	32,762	7FFA	32,167	7DA7	31,572	7B54	30,977	7901
58	3A	32,764	7FFC	32,169	7DA9	31,574	7B56	30,979	7903
60	3C	32,766	7FFE	32,171	7DAB	31,576	7B58	30,981	7905
62	3E	32,768	8000	32,173	7DAD	31,578	7B5A	30,983	7907
64	40	32,770	8002	32,175	7DAF	31,580	7B5C	30,985	7909
66	42	32,772	8004	32,177	7DB1	31,582	7B5E	30,987	790B
68	44	32,774	8006	32,179	7DB3	31,584	7B60	30,989	790D
70	46	32,776	8008	32,181	7DB5	31,586	7B62	30,991	790F
72	48	32,778	800A	32,183	7DB7	31,588	7B64	30,993	7911
74	4A	32,780	800C	32,185	7DB9	31,590	7B66	30,995	7913
76	4C	32,782	800E	32,187	7DBB	31,592	7B68	30,997	7915
78	4E	32,784	8010	32,189	7DBD	31,594	7B6A	30,999	7917
80	50	32,786	8012	32,191	7DBF	31,596	7B6C	31,001	7919
82	52	32,788	8014	32,193	7DC1	31,598	7B6E	31,003	791B
84	54	32,790	8016	32,195	7DC3	31,600	7B70	31,005	791D
86	56	32,792	8018	32,197	7DC5	31,602	7B72	31,007	791F
88	58	32,794	801A	32,199	7DC7	31,604	7B74	31,009	7921
90	5A	32,796	801C	32,201	7DC9	31,606	7B76	31,011	7923
92	5C	32,798	801E	32,203	7DCB	31,608	7B78	31,013	7925
94	5E	32,800	8020	32,205	7DCD	31,610	7B7A	31,015	7927
96	60	32,802	8022	32,207	7DCF	31,612	7B7C	31,017	7929
98	62	32,804	8024	32,209	7DD1	31,614	7B7E	31,019	792B
100	64	32,806	8026	32,211	7DD3	31,616	7B80	31,021	792D
102	66	32,808	8028	32,213	7DD5	31,618	7B82	31,023	792F
104	68	32,810	802A	32,215	7DD7	31,620	7B84	31,025	7931
106	6A	32,812	802C	32,217	7DD9	31,622	7B86	31,027	7933
108	6C	32,814	802E	32,219	7ddb	31,624	7B88	31,029	7935
110	6E	32,816	8030	32,221	7DDD	31,626	7B8A	31,031	7937
112	70	32,818	8032	32,223	7DDF	31,628	7B8C	31,033	7939
114	72	32,820	8034	32,225	7DE1	31,630	7B8E	31,035	793B
116	74	32,822	8036	32,227	7DE3	31,632	7B90	31,037	793D
118	76	32,824	8038	32,229	7DE5	31,634	7B92	31,039	793F
120	78	32,826	803A	32,231	7DE7	31,636	7B94	31,041	7941
122	7A	32,828	803C	32,233	7DE9	31,638	7B96	31,043	7943
124	7C	32,830	803E	32,235	7DEB	31,640	7B98	31,045	7945

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #13		DOS BUFFER #14		DOS BUFFER #15		DOS BUFFER #16	
DEX	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
126	7E	32,832	8040	32,237	7DED	31,642	7B9A	31,047	7947
128	80	32,834	8042	32,239	7DEF	31,644	7B9C	31,049	7949
130	82	32,836	8044	32,241	7DF1	31,646	7B9E	31,051	794B
132	84	32,838	8046	32,243	7DF3	31,648	7BA0	31,053	794D
134	86	32,840	8048	32,245	7DF5	31,650	7BA2	31,055	794F
136	88	32,842	804A	32,247	7DF7	31,652	7BA4	31,057	7951
138	8A	32,844	804C	32,249	7DF9	31,654	7BA6	31,059	7953
140	8C	32,846	804E	32,251	7DFB	31,656	7BA8	31,061	7955
142	8E	32,848	8050	32,253	7DFD	31,658	7BAA	31,063	7957
144	90	32,850	8052	32,255	7DFE	31,660	7BAC	31,065	7959
146	92	32,852	8054	32,257	7E01	31,662	7BAE	31,067	795B
148	94	32,854	8056	32,259	7E03	31,664	7BB0	31,069	795D
150	96	32,856	8058	32,261	7E05	31,666	7BB2	31,071	795F
152	98	32,858	805A	32,263	7E07	31,668	7BB4	31,073	7961
154	9A	32,860	805C	32,265	7E09	31,670	7BB6	31,075	7963
156	9C	32,862	805E	32,267	7E0B	31,672	7BB8	31,077	7965
158	9E	32,864	8060	32,269	7E0D	31,674	7BBA	31,079	7967
160	A0	32,866	8062	32,271	7E0F	31,676	7BBC	31,081	7969
162	A2	32,868	8064	32,273	7E11	31,678	7BBE	31,083	796B
164	A4	32,870	8066	32,275	7E13	31,680	7BC0	31,085	796D
166	A6	32,872	8068	32,277	7E15	31,682	7BC2	31,087	796F
168	A8	32,874	806A	32,279	7E17	31,684	7BC4	31,089	7971
170	AA	32,876	806C	32,281	7E19	31,686	7BC6	31,091	7973
172	AC	32,878	806E	32,283	7E1B	31,688	7BC8	31,093	7975
174	AE	32,880	8070	32,285	7E1D	31,690	7BCA	31,095	7977
176	B0	32,882	8072	32,287	7E1F	31,692	7BCC	31,097	7979
178	B2	32,884	8074	32,289	7E21	31,694	7BCE	31,099	797B
180	B4	32,886	8076	32,291	7E23	31,696	7BD0	31,101	797D
182	B6	32,888	8078	32,293	7E25	31,698	7BD2	31,103	797F
184	B8	32,890	807A	32,295	7E27	31,700	7BD4	31,105	7981
186	BA	32,892	807C	32,297	7E29	31,702	7BD6	31,107	7983
188	BC	32,894	807E	32,299	7E2B	31,704	7BD8	31,109	7985
190	BE	32,896	8080	32,301	7E2D	31,706	7BDA	31,111	7987
192	C0	32,898	8082	32,303	7E2F	31,708	7BDC	31,113	7989
194	C2	32,900	8084	32,305	7E31	31,710	7BDE	31,115	798B
196	C4	32,902	8086	32,307	7E33	31,712	7BE0	31,117	798D
198	C6	32,904	8088	32,309	7E35	31,714	7BE2	31,119	798F
200	C8	32,906	808A	32,311	7E37	31,716	7BE4	31,121	7991
202	CA	32,908	808C	32,313	7E39	31,718	7BE6	31,123	7993
204	CC	32,910	808E	32,315	7E3B	31,720	7BE8	31,125	7995
206	CE	32,912	8090	32,317	7E3D	31,722	7BEA	31,127	7997
208	D0	32,914	8092	32,319	7E3F	31,724	7BEC	31,129	7999
210	D2	32,916	8094	32,321	7E41	31,726	7BEE	31,131	799B
212	D4	32,918	8096	32,323	7E43	31,728	7BF0	31,133	799D
214	D6	32,920	8098	32,325	7E45	31,730	7BF2	31,135	799F
216	D8	32,922	809A	32,327	7E47	31,732	7BF4	31,137	79A1
218	DA	32,924	809C	32,329	7E49	31,734	7BF6	31,139	79A3
220	DC	32,926	809E	32,331	7E4B	31,736	7BF8	31,141	79A5
222	DE	32,928	80A0	32,333	7E4D	31,738	7BFA	31,143	79A7

RELATIVE BYTE OFFSET FROM ADDR		DOS BUFFER #13		DOS BUFFER #14		DOS BUFFER #15		DOS BUFFER #16	
DEX	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
224	E0	32,930	80A2	32,335	7E4F	31,740	7BFC	31,145	79A9
226	E2	32,932	80A4	32,337	7E51	31,742	7BFE	31,147	79AB
228	E4	32,934	80A6	32,339	7E53	31,744	7C00	31,149	79AD
230	E6	32,936	80A8	32,341	7E55	31,746	7C02	31,151	79AF
232	E8	32,938	80AA	32,343	7E57	31,748	7C04	31,153	79B1
234	EA	32,940	80AC	32,345	7E59	31,750	7C06	31,155	79B3
236	EC	32,942	80AE	32,347	7E5B	31,752	7C08	31,157	79B5
238	EE	32,944	80B0	32,349	7E5D	31,754	7C0A	31,159	79B7
240	F0	32,946	80B2	32,351	7E5F	31,756	7C0C	31,161	79B9
242	F2	32,948	80B4	32,353	7E61	31,758	7C0E	31,163	79BB
244	F4	32,950	80B6	32,355	7E63	31,760	7C10	31,165	79BD
246	F6	32,952	80B8	32,357	7E65	31,762	7C12	31,167	79BF
248	F8	32,954	80BA	32,359	7E67	31,764	7C14	31,169	79C1
250	FA	32,956	80BC	32,361	7E69	31,766	7C16	31,171	79C3
252	FC	32,958	80BE	32,363	7E6B	31,768	7C18	31,173	79C5
254	FE	32,960	80C0	32,365	7E6D	31,770	7C1A	31,175	79C7

SECTION 3: FILE MANAGER WORK AREA

0	0	32,962	80C2	32,367	7E6F	31,772	7C1C	31,177	79C9
2	2	32,964	80C4	32,369	7E71	31,774	7C1E	31,179	79CB
4	4	32,966	80C6	32,371	7E73	31,776	7C20	31,181	79CD
5	5	32,967	80C7	32,372	7E74	31,777	7C21	31,182	79CE
7	7	32,969	80C9	32,374	7E76	31,779	7C23	31,184	79D0
8	8	32,970	80CA	32,375	7E77	31,780	7C24	31,185	79D1
9	9	32,971	80CB	32,376	7E78	31,781	7C25	31,186	79D2
11	B	32,973	80CD	32,378	7E7A	31,783	7C27	31,188	79D4
13	D	32,975	80CF	32,380	7E7C	31,785	7C29	31,190	79D6
15	F	32,977	80D1	32,382	7E7E	31,787	7C2B	31,192	79D8
17	11	32,979	80D3	32,384	7E80	31,789	7C2D	31,194	79DA
19	13	32,981	80D5	32,386	7E82	31,791	7C2F	31,196	79DC
21	15	32,983	80D7	32,388	7E84	31,793	7C31	31,198	79DE
22	16	32,984	80D8	32,389	7E85	31,794	7C32	31,199	79DF
23	17	32,985	80D9	32,390	7E86	31,795	7C33	31,200	79E0
25	19	32,987	80DB	32,392	7E88	31,797	7C35	31,202	79E2
27	1B	32,989	80DD	32,394	7E8A	31,799	7C37	31,204	79E4
29	1D	32,991	80DF	32,396	7E8C	31,801	7C39	31,206	79E6
31	1F	32,993	80E1	32,398	7E8E	31,803	7C3B	31,208	79E8
32	20	32,994	80E2	32,399	7E8F	31,804	7C3C	31,209	79E9
33	21	32,995	80E3	32,400	7E90	31,805	7C3D	31,210	79EA
37	25	32,999	80E7	32,404	7E94	31,809	7C41	31,214	79EE
38	26	33,000	80E8	32,405	7E95	31,810	7C42	31,215	79EF
39	27	33,001	80E9	32,406	7E96	31,811	7C43	31,216	79F0
40	28	33,002	80EA	32,407	7E97	31,812	7C44	31,217	79F1
41	29	33,003	80EB	32,408	7E98	31,813	7C45	31,218	79F2
42	2A	33,004	80EC	32,409	7E99	31,814	7C46	31,219	79F3

TRACK/SECTOR LIST NEXT FILE SECTOR CHART

All files saved on disk have a Track/Sector List. Basically, these lists tell where the file's data sectors are stored. The catalog sector points to the Track/Sector List, and the Track/Sector List points to the sectors where the data are stored. Each list has the hexadecimal value for the track and sector (pairs), in that order, for each data sector.

With this chart, it is possible to easily determine each sequential data sector position where a file's data are stored. On the left side of each column is the decimal and hexadecimal value for the Track/Sector List relative byte offset (TSL REL BYTE OF). By cross indexing the byte offset with the File Data Sector Position (FILE DATA SEC POSN), one can easily determine where file data information is stored on the disk. If the data sectors exceed 122, the highest position indicated on the chart, additional Track/Sector List sectors will be added.

TSL REL FILE DATA TSL REL FILE DATA TSL REL FILE DATA TSL REL FILE DATA
BYTE OF SEC POSN BYTE OF SEC POSN BYTE OF SEC POSN BYTE OF SEC POSN

DEC	HEX		DEC	HEX		DEC	HEX		DEC	HEX	
12	C	1st	72	48	31st	132	84	61st	192	C0	91st
14	E	2nd	74	4A	32nd	134	86	62nd	194	C2	92nd
16	10	3rd	76	4C	33rd	136	88	63rd	196	C4	93rd
18	12	4th	78	4E	34th	138	8A	64th	198	C6	94th
20	14	5th	80	50	35th	140	8C	65th	200	C8	95th
22	16	6th	82	52	36th	142	8E	66th	202	CA	96th
24	18	7th	84	54	37th	144	90	67th	204	CC	97th
26	1A	8th	86	56	38th	146	92	68th	206	CE	98th
28	1C	9th	88	58	39th	148	94	69th	208	D0	99th
30	1E	10th	90	5A	40th	150	96	70th	210	D2	100th
32	20	11th	92	5C	41st	152	98	71st	212	D4	101st
34	22	12th	94	5E	42nd	154	9A	72nd	214	D6	102nd
36	24	13th	96	60	43rd	156	9C	73rd	216	D8	103rd
38	26	14th	98	62	44th	158	9E	74th	218	DA	104th
40	28	15th	100	64	45th	160	A0	75th	220	DC	105th
42	2A	16th	102	66	46th	162	A2	76th	222	DE	106th
44	2C	17th	104	68	47th	164	A4	77th	224	E0	107th
46	2E	18th	106	6A	48th	166	A6	78th	226	E2	108th
48	30	19th	108	6C	49th	168	A8	79th	228	E4	109th
50	32	20th	110	6E	50th	170	AA	80th	230	E6	110th
52	34	21st	112	70	51st	172	AC	81st	232	E8	111th
54	36	22nd	114	72	52nd	174	AE	82nd	234	EA	112th
56	38	23rd	116	74	53rd	176	B0	83rd	236	EC	113th
58	3A	24th	118	76	54th	178	B2	84th	238	EE	114th
60	3C	25th	120	78	55th	180	B4	85th	240	F0	115th
62	3E	26th	122	7A	56th	182	B6	86th	242	F2	116th
64	40	27th	124	7C	57th	184	B8	87th	244	F4	117th
66	42	28th	126	7E	58th	186	BA	88th	246	F6	118th
68	44	29th	128	80	59th	188	BC	89th	248	F8	119th
70	46	30th	130	82	60th	190	BE	90th	250	FA	120th
									252	FC	121st
									254	FE	122nd

DOS INPUT/OUTPUT CHARTS

The following three charts deal with various aspects of DOS stored in RAM memory as they relate to disk storage locations. The leftmost column of each chart is the relative byte offset, in hex and decimal. The middle column is the location in RAM, again in hex and decimal, and the rightmost column is the hex storage location on the disk. The abbreviation "TTSBB" refers to Track Sector Byte. (Since all sectors {0-F} are one digit values, the "S" is singular. Both track and byte values can be two digit values.) The value of many tracks is zero, and therefore not shown in the "TT" column, since DOS is stored on track 0, 1 and 2. (e.g., Track 0, Sector 1, Byte C) appears on the chart as:

TTSBB
1FC

The File Manager is a section of DOS which is told by the Command Interpreter of DOS to place an image of RAM memory into disk memory (or vice versa). The Command Interpreter gives the File Manager instructions on what it wants done by placing the instructions into the "File Manager Parameters List." The File Manager (FM) figures out where information is to be stored to the diskette using the VTOC. The FM builds the track/sector lists and updates the directory sectors. The FM also requests information from the RWTS and places it into position in RAM memory.

The FM cannot actually save or retrieve information to or from the diskette. Only RWTS can do that. The FM places its instructions into the "IOB," then RWTS does its job. The "DCT" is used by RWTS to describe what kind of device (disk drive) is connected to the system.

Each chart can be used in examining RAM to identify the corresponding location on the disk for the various DOS functions or vice versa if examining track and sectors. By cross-indexing the different locations or offsets, it is possible to find the corresponding RAM, sectors or offset. Each chart has separate abbreviations for column headings.

INPUT OUTPUT BLOCK CHART (PARAMETER LIST FOR RWTS)

IOB REL BYTE OF = INPUT OUTPUT BLOCK RELATIVE BYTE
OFFSET

IOB ADDRESS LOC IN DOS (RAM) = IOB ADDRESS
LOCATIONS IN DOS (RAM)

IOB ST LOC ON DISK = IOB STORAGE LOCATION ON DISK

IOB REL BYTE OF	IOB ADDRESS IN DOS (RAM)	IOB ST LOC ON DISK
--------------------	-----------------------------	-----------------------

				TTSBB
DEC	HEX	DEC	HEX	HEX
0	0	47,080	B7E8	1E8
1	1	47,081	B7E9	1E9
2	2	47,082	B7EA	1EA
3	3	47,083	B7EB	1EB
4	4	47,084	B7EC	1EC
5	5	47,085	B7ED	1ED
6	6	47,086	B7EE	1EE
8	8	47,088	B7F0	1F0
10	A	47,090	B7F2	1F2
11	B	47,091	B7F3	1F3

12	C	47,092	B7F4	1F4
13	D	47,093	B7F5	1F5
14	E	47,094	B7F6	1F6
15	F	47,095	B7F7	1F7
16	10	47,096	B7F8	1F8

DEVICE CHARACTERISTICS TABLE

DCT REL BYTE OF = DEVICE CHARACTERISTICS TABLE
 RELATIVE BYTE OFFSET
 DCT ADDR LOC IN DOS (RAM) = DCT ADDRESS LOCATIONS
 IN DOS (RAM)
 DCT ST LOC ON DISK = DCT STORAGE LOCATIONS ON DISK

IOB REL BYTE OF	IOB ADDRESS IN DOS (RAM)	IOB ST LOC ON DISK
--------------------	-----------------------------	-----------------------

D	H	DEC	HEX	HEX
0	0	47,099	B7FB	1FB
1	1	47,100	B7FC	1FC
2	2	47,101	B7FD	1FD

FILE MANAGER PARAMETERS LIST CHART

DCT REL BYTE OF	DCT ADDR LOC IN DOS (RAM)	DCT ST LOC ON DISK
--------------------	------------------------------	-----------------------

FMPL = FILE MANAGER PARAMETERS LIST
 FMPL REL OFFSET = FMPL RELATIVE OFFSET BYTE
 FMPL ADDR LOC IN DOS (RAM) = FMPL ADDRESS LOCATIONS
 IN DOS (RAM)
 FMPL ST LOC DISK = FMPL STORAGE LOCATIONS ON DISK

FMPL REL OFFSET	FMPL ADDR LOC IN DOS (RAM)	FMPL ST LOC DISK
--------------------	-------------------------------	---------------------

DEC	HEX	DEC	HEX	HEX
0	0	46,523	B5BB	24BB
1	1	46,524	B5BC	24BC
2	2	46,525	B5BD	24BD
10	A	46,533	B5C5	24C5
11	B	46,534	B5C6	24C6
12	C	46,535	B5C7	24C7
14	E	46,537	B5C9	24C9
16	10	46,539	B5CB	24CB

DOS COMMAND TABLES

The following two charts contain various relevant addresses for DOS commands. It is important to note that for a 48K slave, the address on the disk will be the same as the RTS address in a 48K RAM DOS as shown. However, a "mastered" disk will always show the RTS addresses for a 16K RAM DOS on the DOS image stored on the diskette. For example, a 48K slave RTS address will show \$A54E but a "mastered" disk will show \$254E. The first chart has the addresses for entry points for DOS commands issued from BASIC. In the first three columns are the DOS command entry routines' table locations in RAM and on disk. (All disk locations are on Track 0 so no value is listed under the "TT" column. Only the Sector and Byte values are shown.) For example, the entry point pointer for MON is at 40,256 (\$9D40) in RAM, and Track \$0, Sector \$C, Byte \$22 on disk. The right three columns of the chart have the RTS entry addresses for 48K of RAM, the actual DOS routine address and the storage location on disk. Note that the entry RTS addresses are all one less than the actual DOS routine addresses. The actual DOS routine addresses are all stored on Track \$1 of the disk.

The second table shows the addresses for the DOS strings in the first three columns, and the command flag addresses in the second three. The actual ASCII values for each DOS command are stored at the addresses in RAM and locations on disk shown. Depending on the length of the command name, different numbers of bytes are used. For example, FP (\$46 \$D0) takes up two bytes, while EXEC (\$45 \$58 \$45 \$C3) takes up four. (On disk, all information is on Track \$1.) The fourth column has the addresses for the optional and required flags on each command, given in hex. Several of these command entries are identical, such as CHAIN, DELETE, LOCK, UNLOCK and VERIFY. Finally, the last two columns show the addresses of DOS command flags where two bytes of information are stored indicating the set flags. Both the RAM address and disk location are given.

Using the second table and the KEYWORD NAME AND POSITION TABLE CHART, it is possible to make the NOMON command accept the "D#" parameter. Before changing DOS, enter

```
NOMON,I,O,C,D2
```

You will get a SYNTAX ERROR. Now locate the address of the DOS CMD FLG (43,309/\$A920). Enter the monitor and enter

```
A92D <RETURN>
```

Then press RETURN again and you will see 40 80..., which is also shown on this table. The second byte (\$80) allows only I,O,C parameters. Now look at the KEYWORD NAME AND POSITION CHART across D's line in column KW FLG BIT PS and you will see a \$20. Logical AND this to \$80 (\$20) and you have \$A0. (\$80 AND \$20 = \$A0). Now enter

```
A92E: A0 <RETURN>
3D0G <RETURN>
NOMON,I,O,C,D2
```

This time there will be no SYNTAX ERROR. If you type CATALOG (assuming you have a second disk drive), it will now list the files on the screen.

DOS CMD = DOS COMMAND
 DOS CMD ENTRY TBL DOS ROU RTS ADDR = DOS COMMAND ENTRY TABLE OF DOS
 ROUTINE RTS ADDRESSES
 DOS CMD EN TBL DK = DOS COMMAND ENTRY TABLE LOCATIONS ON DISK
 DOS ROUTINE EN TRS AD 48K RAM = DOS ROUTINE ENTRY RTS ADDRESS IN TABLE
 FOR 48K RAM
 ACT DOS ROU ADD DK = ACTUAL DOS ROUTINE ADDRESS STORED ON DISK
 TTSBB = TRACK SECTOR BYTE

DOS COMMAND ENTRY TABLE CHART

DOS CMD NAME DK	DOS CMD ENTRY TBL DOS ROU RTS ADDR	DOS CMD EN TBL TTSBB	DOS ROUTINE DK EN RTS AD 48K RAM	ACTUAL DOS ROUTINE ADDR 48K RAM	ACT DOS ROU ADD TTSBB
	DEC HEX	HEX	DEC HEX	DEC HEX	HEX
INIT	40,222 9D1E	C1E	42,318 A54E	42,319 A54F	144F
LOAD	40,224 9D20	C20	42,002 A412	42,003 A413	1313
SAVE	40,226 9D22	C22	41,878 A396	41,879 A397	1297
RUN	40,228 9D24	C24	42,192 A4D0	42,193 A4D1	13D1
CHAIN	40,230 9D26	C26	42,223 A4EF	42,224 A4F0	13F0
DELETE	40,232 9D28	C28	41,570 A262	41,571 A263	1163
LOCK	40,234 9D2A	C2A	41,584 A270	41,585 A271	1171
UNLOCK	40,236 9D2C	C2C	41,588 A274	41,589 A275	1175
CLOSE	40,238 9D2E	C2E	41,705 A2E9	41,706 A2EA	11EA
READ	40,240 9D30	C30	42,266 A51A	42,267 A51B	141B
EXEC	40,242 9D32	C32	42,437 A5C5	42,438 A5C6	14C6
WRITE	40,244 9D34	C34	42,255 A50F	42,256 A510	1410
POSITION	40,246 9D36	C36	42,460 A5DC	42,461 A5DD	14DD
OPEN	40,248 9D38	C38	41,634 A2A2	41,635 A2A3	11A3
APPEND	40,250 9D3A	C3A	41,623 A297	41,624 A298	1198
RENAME	40,252 9D3C	C3C	41,600 A280	41,601 A281	1181
CATALOG	40,254 9D3E	C3E	42,349 A56D	42,350 A56E	146E
MON	40,256 9D40	C40	41,522 A232	41,523 A233	1133
NOMON	40,258 9D42	C42	41,532 A23C	41,533 A23D	113D
PR#	40,260 9D44	C44	41,512 A228	41,513 A229	1129
IN#	40,262 9D46	C46	41,517 A22D	41,518 A22E	112E
MAXFILES	40,264 9D48	C48	41,552 A250	41,553 A251	1151
FP	40,266 9D4A	C4A	42,361 A579	42,362 A57A	147A
INT	40,268 9D4C	C4C	42,397 A59D	42,398 A59E	149E
BSAVE	40,270 9D4E	C4E	41,776 A330	41,777 A331	1231
BLOAD	40,272 9D50	C50	41,820 A35C	41,821 A35D	125D
BRUN	40,274 9D52	C52	41,869 A38D	41,870 A38E	128E
VERIFY	40,276 9D54	C54	41,596 A27C	41,597 A27D	117D

DOS CMD = DOS Command
 TTSBB = Track Sector Byte
 DOS CMD NM ST = DOS CMD Name String Literal
 DOS CMD NAME ST LIT ADR RAM = DOS CMD Name String Literal Address in RAM
 DOS CMD DISK = DOS CMD Name Literal Storage on Disk
 DOS CMD FLAGS = DOS CMD Optional and Required Parameters Flags
 DOS CMD FLGS ADDR RAM = DOS CMD Flags Address Locations in RAM
 DCF ST DISK = DOS CMD Flags Storage Location on Disk

DOS COMMAND STRINGS & FLAGS TABLE CHART

DOS CMD NM ST LIT	DOS CMD NAME ST LIT ADR RAM	DOS CMD DISK	DOS CMD FLAGS	DOS CMD FLGS ADDR RAM	DOS CMD FLGS ADDR RAM	DCF ST DISK
	DEC	HEX	HEX	HEX	DEC	HEX
INIT	43, 140	A884	1784	2170	43, 273	A909
LOAD	43, 144	A888	1788	A070	43, 275	A90B
SAVE	43, 148	A88C	178C	A170	43, 277	A90D
RUN	43, 152	A890	1790	A070	43, 279	A90F
CHAIN	43, 155	A893	1793	2070	43, 281	A911
DELETE	43, 160	A898	1798	2070	43, 283	A913
LOCK	43, 166	A89E	179E	2070	43, 285	A915
UNLOCK	43, 170	A8A2	17A2	2070	43, 287	A917
CLOSE	43, 176	A8A8	17A8	6000	43, 289	A919
READ	43, 181	A8AD	17AD	2206	43, 291	A91B
EXEC	43, 185	A8B1	17B1	2074	43, 293	A91D
WRITE	43, 189	A8B5	17B5	2206	43, 295	A91F
POSITION	43, 194	A8BA	17BA	2204	43, 297	A921
OPEN	43, 202	A8C2	17C2	2378	43, 299	A923
APPEND	43, 206	A8C6	17C6	2270	43, 301	A925
RENAME	43, 212	A8CC	17CC	3070	43, 303	A927
CATALOG	43, 218	A8D2	17D2	4070	43, 305	A929
MON	43, 225	A8D9	17D9	4080	43, 307	A92B
NOMON	43, 228	A8DC	17DC	4080	43, 309	A92D
PR#	43, 233	A8E1	17E1	800	43, 311	A92F
IN#	43, 236	A8E4	17E4	800	43, 313	A931
MAXFILES	43, 239	A8E7	17E7	400	43, 315	A933
FP	43, 247	A8EF	17EF	4070	43, 317	A935
INT	43, 249	A8F1	17F1	4000	43, 319	A937
BSAVE	43, 252	A8F4	17F4	2179	43, 321	A939
BLOAD	43, 257	A8F9	17F9	2071	43, 323	A93B
BRUN	43, 262	A8FE	17FE	2071	43, 325	A93D
VERIFY	43, 266	A902	1802	2070	43, 327	A93F

DATA	RAM	DISK	DATA	RAM	DISK
------	-----	------	------	-----	------

DOS PARAMETER KEYWORD CHARTS

The various parameters associated with DOS commands are referred to as "keywords." The following one-letter abbreviations are used with DOS parameter keywords:

V = VOLUME
D = DISK DRIVE NUMBER
S = SLOT NUMBER
L = LENGTH OF BINARY FILE
R = RELATIVE FIELD (SEQ FILE)
R = RECORD NUMBER (RANDOM ACCESS FILE)
B = BYTE
A = STARTING ADDRESS OF BINARY FILE
C = COMMANDS
I = INPUT
O = OUTPUT

The Keyword Name and Position Table Chart shows the keyword bit positions and flag positions locations in RAM and on disk. Using this table, it is possible to examine RAM to see what flags are set in byte \$AA65, indicating the presence of a keyword in a DOS command.

The second table, Keyword Value Min/Max Chart, has a dual purpose. First, it shows the minimum and maximum values a keyword can have. For example, as most realize, the maximum value for a disk's Volume Number is 254, and the minimum value is 0 (not 1). Second, the chart shows where these minimum and maximum values are stored in RAM and on disk. There are no values listed for C, I and O since their values are not numeric. (E.g., MON C,I,O <- no values given.)

KEYWORD NAME AND POSITION TABLE CHART

KW = KEYWORD DOS CMD = DOS COMMAND TTSBB = TRACK SECTOR BYTE
KW NM = DOS CMD KEYWORD NAME
DOS CMD KWS AD NAME IN RAM = DOS CMD KEYWORDS NAME ADDRESS IN RAM
KW ST DISK = DOS CMD NAME STORAGE LOCATIONS ON DISK
KW FLG BIT PS = KEYWORD FLAG BIT POSITIONS
KW FLG DISK = KEYWORD FLAG BIT POSITIONS ON DISK

KW NM	DOS CMD KWS AD NAME IN RAM		KW ST DISK	KW FLG BIT PS	KW FLAG BIT POSITION IN RAM		KW FLG DISK
	DEC	HEX			DEC	HEX	
V	43,329	A941	1841	40	43,339	A94B	184B
D	43,330	A942	1842	20	43,340	A94C	184C
S	43,331	A943	1843	10	43,341	A94D	184D
L	43,332	A944	1844	8	43,342	A94E	184E
R	43,333	A945	1845	4	43,343	A94F	184F
B	43,334	A946	1846	2	43,344	A950	1850
A	43,335	A947	1847	1	43,345	A951	1851
*C	43,336	A948	1848	C0	43,346	A952	1852
*I	43,337	A949	1849	A0	43,347	A953	1853
*O	43,338	A94A	184A	90	43,348	A954	1854

**NOTE: C,I,O are allowed by a value of \$80 being ANDed (they are grouped together.)*

KEYWORD VALUE MIN/MAX CHART

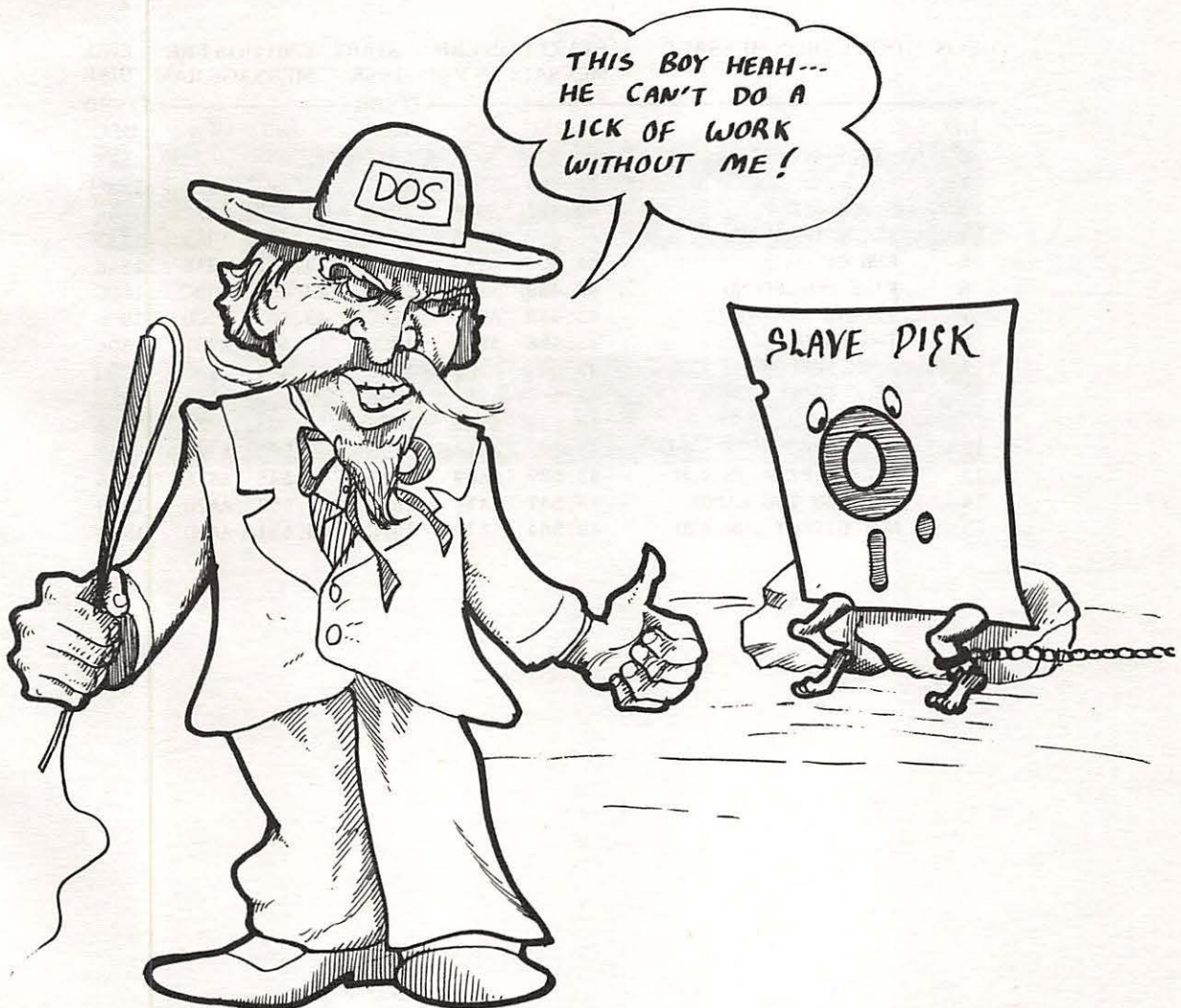
KW = KEYWORD TTSBB = TRACK SECTOR BYTE

MIN (MAX) KW VALUE = MINIMUM (MAXIMUM) KEYWORD VALUE

MIN (MAX) KW VALUE LOC IN RAM = Minimum (MAX) Keyword Value Location in RAM

MIN (MAX) KW LOC DK = Minimum (MAXIMUM) Keyword Storage Location on Disk

KW	MIN KW	MIN KW VALUE		MIN KW	KW	MAX KW	MAX KW VALUE		MAX KW
	VALUE	LOC IN RAM		LOC DK		VALUE	LOC IN RAM		LOC DK
	DEC	DEC	HEX	TTSBB- HEX		DEC	DEC	HEX	TTSBB- HEX
V	0	43,349	A955	1855	V	254	43,351	A957	1857
D	1	43,353	A959	1859	D	2	43,355	A95B	185B
S	1	43,357	A95D	185D	S	7	43,359	A95F	185F
L	1	43,361	A961	1861	L	32,767	43,363	A963	1863
R	0	43,365	A965	1865	R	32,767	43,367	A967	1867
B	0	43,369	A969	1869	B	32,767	43,371	A96B	186B
A	0	43,373	A96D	186D	A	65,535	43,375	A96F	186F



DOS ERROR MESSAGE CHART

This chart shows the storage of DOS error messages in RAM and on disk. Each message is stored in ASCII hex code, and it is a simple matter to modify the messages for a customized set. For example, FILE TYPE MISMATCH, begins at \$AA09 and ends at \$AA1A. By entering the monitor and replacing the code beginning at \$AA09, you can change the content of that message. If you enter

AA09: 54 48 41 54 20 57 4F 4E 27 54 20 57 4F 52 CB 20 20 20

when you try to RUN a "T" (text) file, you will get THAT WON'T WORK, instead of FILE TYPE MISMATCH. Permanent changes can be made for a customized DOS by making changes to the storage locations on disk.

DECN	DOS ERROR MESSAGE	START DOS ERR		START	END DOS ERR		END
		MESSAGE IN RAM		DISK	MESSAGE RAM		DISK
TTSBB							
DEC		DEC	HEX	HEX	DEC	HEX	DEC
0	<CR> <BELL> <CR>	43,377	A971	1871	43,379	A973	1873
1	LANGUAGE NOT AVAILABLE	43,380	A974	1874	43,401	A989	1889
2 & 3	RANGE ERROR	43,402	A98A	188A	43,412	A994	1894
4	WRITE PROTECTED	43,413	A995	1895	43,427	A9A3	18A3
5	END OF DATA	43,428	A9A4	18A4	43,438	A9AE	18AE
6	FILE NOT FOUND	43,439	A9AF	18AF	43,452	A9BC	18BC
7	VOLUME MISMATCH	43,453	A9BD	18BD	43,467	A9CB	18CB
8	I/O ERROR	43,468	A9CC	18CC	43,476	A9D4	18D4
9	DISK FULL	43,477	A9D5	18D5	43,485	A9DD	18DD
10	FILE LOCKED	43,486	A9DE	18DE	43,496	A9E8	18E8
11	SYNTAX ERROR	43,497	A9E9	18E9	43,508	A9F4	18F4
12	NO BUFFERS AVAILABLE	43,509	A9F5	18F5	43,528	AA08	1908
13	FILE TYPE MISMATCH	43,529	AA09	1909	43,546	AA1A	191A
14	PROGRAM TOO LARGE	43,547	AA1B	191B	43,563	AA2B	192B
15	NOT DIRECT COMMAND	43,564	AA2C	192C	43,581	AA3D	193D

DOS ERROR INDEX CHART

DC EC# = DOS Error Code Number

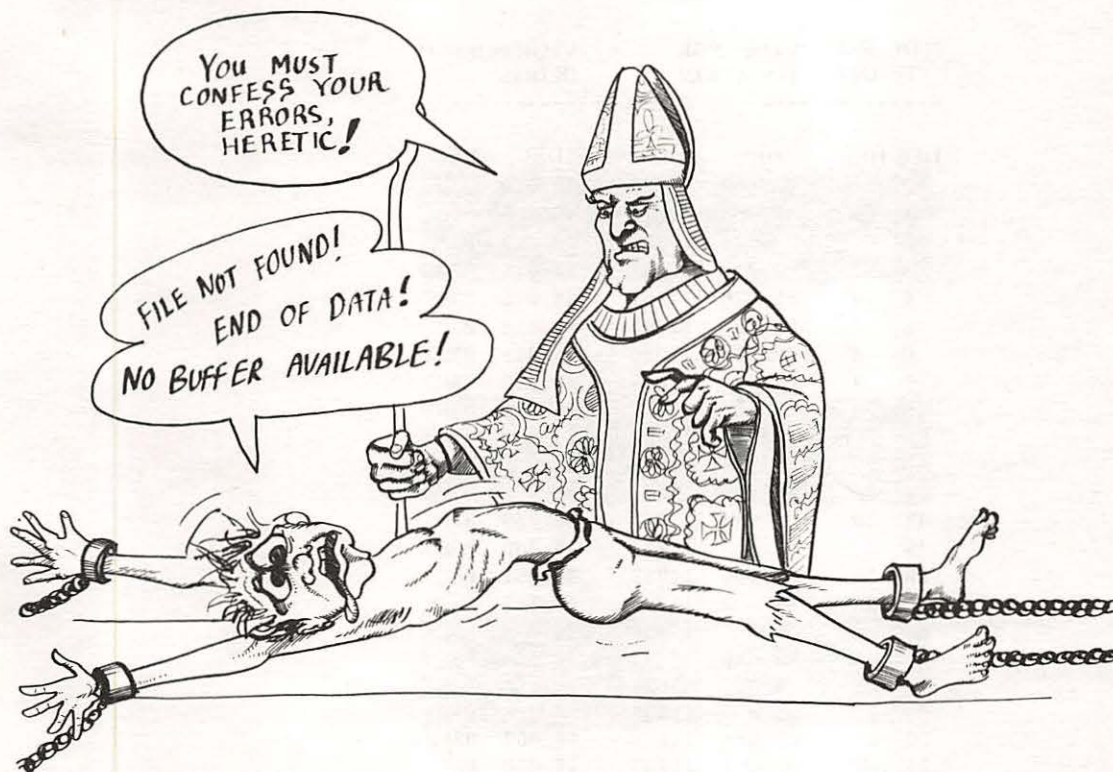
DOS ERR IN OFF = DOS Error Message Index Offset

DOS ERR MS IN OFF LOC RAM = DOS Error Message Index Offset Location in RAM

DOS ER S DISK = DOS Error Message Index Offset Storage on Disk

TTSBB = Track Sector Byte

DOS EC#	DOS ERROR MESSAGE	DOS ERR IN OFF	DOS ERR MS IN OFF LOC RAM	DOS ER S DISK
DEC		DEC HEX	DEC HEX	TTSBB HEX
0	<CR> <BELL> <CR>	0 0	43,583 AA3F	193F
1	LANGUAGE NOT AVAILABLE	3 3	43,584 AA40	1940
2	RANGE ERROR	25 19	43,585 AA41	1941
3	RANGE ERROR	25 19	43,586 AA42	1942
4	WRITE PROTECTED	36 24	43,587 AA43	1943
5	END OF DATA	51 33	43,588 AA44	1944
6	FILE NOT FOUND	62 3E	43,589 AA45	1945
7	VOLUME MISMATCH	76 4C	43,590 AA46	1946
8	I/O ERROR	91 5B	43,591 AA47	1947
9	DISK FULL	100 64	43,592 AA48	1948
10	FILE LOCKED	109 6D	43,593 AA49	1949
11	SYNTAX ERROR	120 78	43,594 AA4A	194A
12	NO BUFFERS AVAILABLE	132 84	43,595 AA4B	194B
13	FILE TYPE MISMATCH	152 98	43,596 AA4C	194C
14	PROGRAM TOO LARGE	170 AA	43,597 AA4D	194D
15	NOT DIRECT COMMAND	187 BB	43,598 AA4E	194E



VOLUME TABLE OF CONTENTS (VTOC) CHART

The VTOC, located on track \$11 (17) sector \$0 of the diskette, stores the information relating to the location of files. Since files are often in non-contiguous locations, the VTOC is vital for locating files and file data. The first \$37 bytes are used to store the following:

- \$01 First catalog sector track-number.
- \$02 First catalog sector sector-number.
- \$03 Version of DOS used to initialize disk - (02 for DOS 3.2 and 03 for DOS 3.3).
- \$06 Volume number.
- \$27 Limit of track/sector pairs possible to contain in single track/sector list sector.
- \$30 Track where sectors were last stored.
- \$31 Positive or negative direction of track usage .
- \$34 Total of tracks on diskette. Usually 35,
but with special disk operating systems, it is possible to have up to 50.
- \$35 Indicates tracks per sector. \$0D for DOS 3.2 (or earlier) and \$10 for DOS 3.3.
- \$36 Low byte for number of bytes per sector.
- \$37 High byte for number of bytes per sector.

The remaining information pertains to the bit map storage of data on the disk. The first and second bytes of the track bit maps are used by 16 sector DOS (3.3). Only part of these bytes are used by 13 sector DOS (3.2 and earlier). The last two bytes are not used at all by either 13 or 16 sector DOS.

VTOC REL BYTE OFF		VTOC DISK LOCATION		VTOC BUFFER IN DOS	
-----TTSBB-----					
DEC	HEX	DEX	HEX	DEC	HEX
0	0	69,632	11000	46,011	B3BB
1	1	69,633	11001	46,012	B3BC
2	2	69,634	11002	46,013	B3BD
3	3	69,635	11003	46,014	B3BE
4	4	69,636	11004	46,015	B3BF
5	5	69,637	11005	46,016	B3C0
6	6	69,638	11006	46,017	B3C1
7	7	69,639	11007	46,018	B3C2
38	26	69,670	11026	46,049	B3E1
39	27	69,671	11027	46,050	B3E2
40	28	69,672	11028	46,051	B3E3
47	2F	69,679	1102F	46,058	B3EA
48	30	69,680	11030	46,059	B3EB
49	31	69,681	11031	46,060	B3EC
50	32	69,682	11032	46,061	B3ED
51	33	69,683	11033	46,062	B3EE
52	34	69,684	11034	46,063	B3EF
53	35	69,685	11035	46,064	B3F0
54	36	69,686	11036	46,065	B3F1
55	37	69,687	11037	46,066	B3F2
56	38	69,688	11038	46,067	B3F3
59	3B	69,691	1103B	46,070	B3F6

VTOC REL BYTE OFF		VTOC DISK LOCATION		VTOC BUFFER IN DOS		
-----TTSBB-----						
DEC	HEX	DEX	HEX	DEC	HEX	Track Bit Map
60	3C	69,692	1103C	46,071	B3F7	Track 1 (1) Bit Map
63	3F	69,695	1103F	46,074	B3FA	
64	40	69,696	11040	46,075	B3FB	Track 2 (2) Bit Map
67	43	69,699	11043	46,078	B3FE	
68	44	69,700	11044	46,079	B3FF	Track 3 (3) Bit Map
71	47	69,703	11047	46,082	B402	
72	48	69,704	11048	46,083	B403	Track 4 (4) Bit Map
75	4B	69,707	1104B	46,086	B406	
76	4C	69,708	1104C	46,087	B407	Track 5 (5) Bit Map
79	4F	69,711	1104F	46,090	B40A	
80	50	69,712	11050	46,091	B40B	Track 6 (6) Bit Map
83	53	69,715	11053	46,094	B40E	
84	54	69,716	11054	46,095	B40F	Track 7 (7) Bit Map
87	57	69,719	11057	46,098	B412	
88	58	69,720	11058	46,099	B413	Track 8 (8) Bit Map
91	5B	69,723	1105B	46,102	B416	
92	5C	69,724	1105C	46,103	B417	Track 9 (9) Bit Map
95	5F	69,727	1105F	46,106	B41A	
96	60	69,728	11060	46,107	B41B	Track 10 (A) Bit Map
99	63	69,731	11063	46,110	B41E	
100	64	69,732	11064	46,111	B41F	Track 11 (B) Bit Map
103	67	69,735	11067	46,114	B422	
104	68	69,736	11068	46,115	B423	Track 12 (C) Bit Map
107	6B	69,739	1106B	46,118	B426	
108	6C	69,740	1106C	46,119	B427	Track 13 (D) Bit Map
111	6F	69,743	1106F	46,122	B42A	
112	70	69,744	11070	46,123	B42B	Track 14 (E) Bit Map
115	73	69,747	11073	46,126	B42E	
116	74	69,748	11074	46,127	B42F	Track 15 (F) Bit Map
119	77	69,751	11077	46,130	B432	
120	78	69,752	11078	46,131	B433	Track 16 (10) Bit Map
123	7B	69,755	1107B	46,134	B436	
124	7C	69,756	1107C	46,135	B437	Track 17 (11) Bit Map
127	7F	69,759	1107F	46,138	B43A	
128	80	69,760	11080	46,139	B43B	Track 18 (12) Bit Map
131	83	69,763	11083	46,142	B43E	
132	84	69,764	11084	46,143	B43F	Track 19 (13) Bit Map
135	87	69,767	11087	46,146	B442	
136	88	69,768	11088	46,147	B443	Track 20 (14) Bit Map
139	8B	69,771	1108B	46,150	B446	
140	8C	69,772	1108C	46,151	B447	Track 21 (15) Bit Map
143	8F	69,775	1108F	46,154	B44A	
144	90	69,776	11090	46,155	B44B	Track 22 (16) Bit Map
147	93	69,779	11093	46,158	B44E	
148	94	69,780	11094	46,159	B44F	Track 23 (17) Bit Map
151	97	69,783	11097	46,162	B452	
152	98	69,784	11098	46,163	B453	Track 24 (18) Bit Map
155	9B	69,787	1109B	46,166	B456	
156	9C	69,788	1109C	46,167	B457	Track 25 (19) Bit Map

VTOC REL BYTE OFF		VTOC DISK LOCATION		VTOC BUFFER IN DOS		
-----TTSBB-----						
DEC	HEX	DEX	HEX	DEC	HEX	Track Bit Map
159	9F	69,791	1109F	46,170	B45A	
160	A0	69,792	110A0	46,171	B45B	Track 26 (1A) Bit Map
163	A3	69,795	110A3	46,174	B45E	
164	A4	69,796	110A4	46,175	B45F	Track 27 (1B) Bit Map
167	A7	69,799	110A7	46,178	B462	
168	A8	69,800	110A8	46,179	B463	Track 28 (1C) Bit Map
171	AB	69,803	110AB	46,182	B466	
172	AC	69,804	110AC	46,183	B467	Track 29 (1D) Bit Map
175	AF	69,807	110AF	46,186	B46A	
176	B0	69,808	110B0	46,187	B46B	Track 30 (1E) Bit Map
179	B3	69,811	110B3	46,190	B46E	
180	B4	69,812	110B4	46,191	B46F	Track 31 (1F) Bit Map
183	B7	69,815	110B7	46,194	B472	
184	B8	69,816	110B8	46,195	B473	Track 32 (20) Bit Map
187	BB	69,819	110BB	46,198	B476	
188	BC	69,820	110BC	46,199	B477	Track 33 (21) Bit Map
191	BF	69,823	110BF	46,202	B47A	
192	C0	69,824	110C0	46,203	B47B	Track 34 (22) Bit Map
195	C3	69,827	110C3	46,206	B47E	

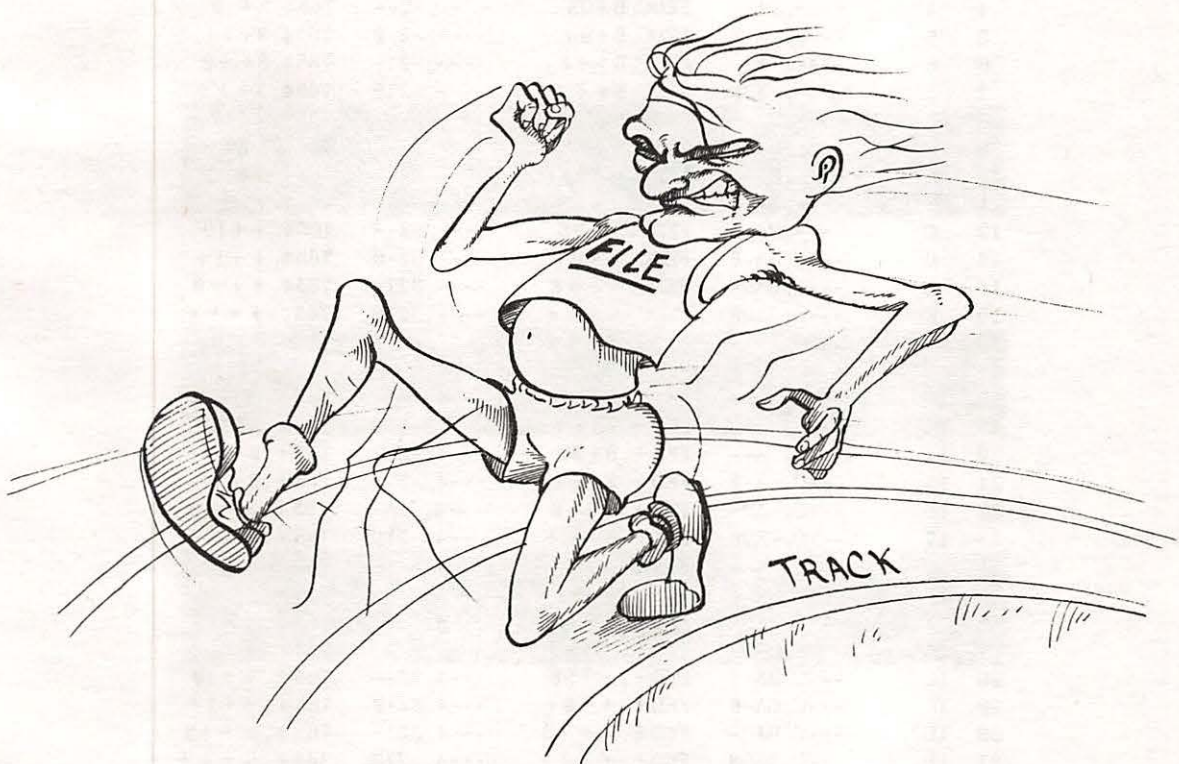
The following are not used by DOS as released by Apple Computer, Inc., but they are employed by some modified disk operating systems on the Apple computer.

196	C4	69,828	110C4	46,207	B47F	Track 35 (23) Bit Map
199	C7	69,831	110C7	46,210	B482	
200	C8	69,832	110C8	46,211	B483	Track 36 (24) Bit Map
203	CB	69,835	110CB	46,214	B486	
204	CC	69,836	110CC	46,215	B487	Track 37 (25) Bit Map
207	CF	69,839	110CF	46,218	B48A	
208	D0	69,840	110D0	46,219	B48B	Track 38 (26) Bit Map
211	D3	69,843	110D3	46,222	B48E	
212	D4	69,844	110D4	46,223	B48F	Track 39 (27) Bit Map
215	D7	69,847	110D7	46,226	B492	
216	D8	69,848	110D8	46,227	B493	Track 40 (28) Bit Map
219	DB	69,851	110DB	46,230	B496	
220	DC	69,852	110DC	46,231	B497	Track 41 (29) Bit Map
223	DF	69,855	110DF	46,234	B49A	
224	E0	69,856	110E0	46,235	B49B	Track 42 (2A) Bit Map
227	E3	69,859	110E3	46,238	B49E	
228	E4	69,860	110E4	46,239	B49F	Track 43 (2B) Bit Map
231	E7	69,863	110E7	46,242	B4A2	
232	E8	69,864	110E8	46,243	B4A3	Track 44 (2C) Bit Map
235	EB	69,867	110EB	46,246	B4A6	
236	EC	69,868	110EC	46,247	B4A7	Track 45 (2D) Bit Map
239	EF	69,871	110EF	46,250	B4AA	
240	F0	69,872	110F0	46,251	B4AB	Track 46 (2E) Bit Map
243	F3	69,875	110F3	46,254	B4AE	
244	F4	69,876	110F4	46,255	B4AF	Track 47 (2F) Bit Map

VTOC REL BYTE OFF	VTOC DISK LOCATION	VTOC BUFFER IN DOS
----------------------	-----------------------	-----------------------

-----TTSBB-----

DEC	HEX	DEX	HEX	DEC	HEX	Track Bit Map
247	F7	69,879	110F7	46,258	B4B2	
248	F8	69,880	110F8	46,259	B4B3	Track 48 (30) Bit Map
251	FB	69,883	110FB	46,262	B4B6	
252	FC	69,884	110FC	46,263	B4B7	Track 49 (31) Bit Map
255	FF	69,887	110FF	46,266	B4BA	



TRACK BIT MAP CHART

This chart shows the bit maps of free and used disk space. A plus (+) sign represents a set bit (it is on), and a minus (-) sign represents a cleared bit (it is off). Set bits are equal to 1 and cleared bits equal 0. Relative bytes #2 and #3 are not used by DOS but reserved for later use. Track usage information is stored in binary format in 4-byte groups.

BIT MAP BYTE VALUE		RELATIVE BYTE #0 OF TK BIT MAPS IN VTOC SECTOR		RELATIVE BYTE #1 OF TK BIT MAPS IN VTOR SECTOR	
DEC	HEX	FREE	USED	FREE	USED
0	0	----,----	FEDC, BA98	----,----	7654, 3210
1	1	----,---8	FEDC, BA9+	----,---0	7654, 321+
2	2	----,--9-	FEDC, BA+8	----,--1-	7654, 32+0
3	3	----,--98	FEDC, BA++	----,--10	7654, 32++
4	4	----,-A--	FEDC, B+98	----,-2--	7654, 3+10
5	5	----,-A-8	FEDC, B+9+	----,-2-0	7654, 3+1+
6	6	----,-A9-	FEDC, B++8	----,-21-	7654, 3++0
7	7	----,-A98	FEDC, B+++	----,-210	7654, 3+++
8	8	----,B---	FEDC, +A98	----,3---	7654, +210
9	9	----,B--8	FEDC, +A9+	----,3--0	7654, +21+
10	A	----,B-9-	FEDC, +A+8	----,3-1-	7654, +2+0
11	B	----,B-98	FEDC, +A++	----,3-10	7654, +2++
12	C	----,BA--	FEDC, ++98	----,32--	7654, ++10
13	D	----,BA-8	FEDC, ++9+	----,32-0	7654, ++1+
14	E	----,BA9-	FEDC, +++8	----,321-	7654, +++0
15	F	----,BA98	FEDC, ++++	----,3210	7654, ++++
16	10	---C,----	FED+, BA98	---4,----	765+, 3210
17	11	---C,---8	FED+, BA9+	---4,---0	765+, 321+
18	12	---C,--9-	FED+, BA+8	---4,--1-	765+, 32+0
19	13	---C,--98	FED+, BA++	---4,--10	765+, 32++
20	14	---C,-A--	FED+, B+98	---4,-2--	765+, 3+10
21	15	---C,-A-8	FED+, B+9+	---4,-2-0	765+, 3+1+
22	16	---C,-A9-	FED+, B++8	---4,-21-	765+, 3++0
23	17	---C,-A98	FED+, B+++	---4,-210	765+, 3+++
24	18	---C,B---	FED+, +A98	---4,3---	765+, +210
25	19	---C,B--8	FED+, +A9+	---4,3--0	765+, +21+
26	1A	---C,B-9-	FED+, +A+8	---4,3-1-	765+, +2+0
27	1B	---C,B-98	FED+, +A++	---4,3-10	765+, +2++
28	1C	---C,BA--	FED+, ++98	---4,32--	765+, ++10
29	1D	---C,BA-8	FED+, ++9+	---4,32-0	765+, ++1+
30	1E	---C,BA9-	FED+, +++8	---4,321-	765+, +++0
31	1F	---C,BA98	FED+, ++++	---4,3210	765+, ++++
32	20	--D-,----	FE+C, BA98	--5-,----	76+4, 3210
33	21	--D-,---8	FE+C, BA9+	--5-,---0	76+4, 321+
34	22	--D-,--9-	FE+C, BA+8	--5-,--1-	76+4, 32+0
35	23	--D-,--98	FE+C, BA++	--5-,--10	76+4, 32++
36	24	--D-,A--	FE+C, B+98	--5-,2--	76+4, 3+10
37	25	--D-,A-8	FE+C, B+9+	--5-,2-0	76+4, 3+1+
38	26	--D-,A9-	FE+C, B++8	--5-,21-	76+4, 3++0
39	27	--D-,A98	FE+C, B+++	--5-,210	76+4, 3+++

BIT MAP BYTE VALUE		RELATIVE BYTE #0 OF TK BIT MAPS IN VTOC SECTOR		RELATIVE #1 OF TK BIT MAPS IN VTOR SECTOR	
DEC	HEX	FREE	USED	FREE	USED
40	28	--D-,B---	FE+C,+A98	--5-,3---	76+4,+210
41	29	--D-,B--8	FE+C,+A9+	--5-,3--0	76+4,+21+
42	2A	--D-,B-9-	FE+C,+A+8	--5-,3-1-	76+4,+2+0
43	2B	--D-,B-98	FE+C,+A++	--5-,3-10	76+4,+2++
44	2C	--D-,BA--	FE+C,++98	--5-,32--	76+4,++10
45	2D	--D-,BA-8	FE+C,++9+	--5-,32-0	76+4,++1+
46	2E	--D-,BA9-	FE+C,+++8	--5-,321-	76+4,+++0
47	2F	--D-,BA98	FE+C,++++	--5-,3210	76+4,++++
48	30	--DC,----	FE++,BA98	--54,----	76++,3210
49	31	--DC,---8	FE++,BA9+	--54,---0	76++,321+
50	32	--DC,--9-	FE++,BA+8	--54,--1-	76++,32+0
51	33	--DC,--98	FE++,BA++	--54,--10	76++,32++
52	34	--DC,-A--	FE++,B+98	--54,-2--	76++,3+10
53	35	--DC,-A-8	FE++,B+9+	--54,-2-0	76++,3+1+
54	36	--DC,-A9-	FE++,B++8	--54,-21-	76++,3++0
55	37	--DC,-A98	FE++,B+++	--54,-210	76++,3+++
56	38	--DC,B---	FE++,+A98	--54,3---	76++,+210
57	39	--DC,B--8	FE++,+A9+	--54,3--0	76++,+21+
58	3A	--DC,B-9-	FE++,+A+8	--54,3-1-	76++,+2+0
59	3B	--DC,B-98	FE++,+A++	--54,3-10	76++,+2++
60	3C	--DC,BA--	FE++,++98	--54,32--	76++,++10
61	3D	--DC,BA-8	FE++,++9+	--54,32-0	76++,++1+
62	3E	--DC,BA9-	FE++,+++8	--54,321-	76++,+++0
63	3F	--DC,BA98	FE++,++++	--54,3210	76++,++++
64	40	-E--,----	F+DC,BA98	-6--,----	7+54,3210
65	41	-E--,---8	F+DC,BA9+	-6--,---0	7+54,321+
66	42	-E--,--9-	F+DC,BA+8	-6--,--1-	7+54,32+0
67	43	-E--,--98	F+DC,BA++	-6--,--10	7+54,32++
68	44	-E--,A--	F+DC,B+98	-6--,2--	7+54,3+10
69	45	-E--,A-8	F+DC,B+9+	-6--,2-0	7+54,3+1+
70	46	-E--,A9-	F+DC,B++8	-6--,21-	7+54,3++0
71	47	-E--,A98	F+DC,B+++	-6--,210	7+54,3+++
72	48	-E--,B---	F+DC,+A98	-6--,3---	7+54,+210
73	49	-E--,B--8	F+DC,+A9+	-6--,3--0	7+54,+21+
74	4A	-E--,B-9-	F+DC,+A+8	-6--,3-1-	7+54,+2+0
75	4B	-E--,B-98	F+DC,+A++	-6--,3-10	7+54,+2++
76	4C	-E--,BA--	F+DC,++98	-6--,32--	7+54,++10
77	4D	-E--,BA-8	F+DC,++9+	-6--,32-0	7+54,++1+
78	4E	-E--,BA9-	F+DC,+++8	-6--,321-	7+54,+++0
79	4F	-E--,BA98	F+DC,++++	-6--,3210	7+54,++++
80	50	-E-C,----	F+D+,BA98	-6-4,----	7+5+,3210
81	51	-E-C,---8	F+D+,BA9+	-6-4,---0	7+5+,321+
82	52	-E-C,--9-	F+D+,BA+8	-6-4,--1-	7+5+,32+0
83	53	-E-C,--98	F+D+,BA++	-6-4,--10	7+5+,32++
84	54	-E-C,-A--	F+D+,B+98	-6-4,-2--	7+5+,3+10
85	55	-E-C,-A-8	F+D+,B+9+	-6-4,-2-0	7+5+,3+1+
86	56	-E-C,-A9-	F+D+,B++8	-6-4,-21-	7+5+,3++0
87	57	-E-C,-A98	F+D+,B+++	-6-4,-210	7+5+,3+++
88	58	-E-C,B---	F+D+,+A98	-6-4,3---	7+5+,+210

BIT MAP BYTE VALUE		RELATIVE BYTE #0 OF TK BIT MAPS IN VTOC SECTOR		RELATIVE #1 OF TK BIT MAPS IN VTOR SECTOR	
DEC	HEX	FREE	USED	FREE	USED
89	59	-E-C, B--8	F+D+, +A9+	-6-4, 3--0	7+5+, +21+
90	5A	-E-C, B-9-	F+D+, +A+8	-6-4, 3-1-	7+5+, +2+0
91	5B	-E-C, B-98	F+D+, +A++	-6-4, 3-10	7+5+, +2++
92	5C	-E-C, BA--	F+D+, ++98	-6-4, 32--	7+5+, ++10
93	5D	-E-C, BA-8	F+D+, ++9+	-6-4, 32-0	7+5+, ++1+
94	5E	-E-C, BA9-	F+D+, +++8	-6-4, 321-	7+5+, +++0
95	5F	-E-C, BA98	F+D+, ++++	-6-4, 3210	7+5+, ++++
96	60	-ED-, ----	F++C, BA98	-65-, ----	7++4, 3210
97	61	-ED-, ---8	F++C, BA9+	-65-, ---0	7++4, 321+
98	62	-ED-, --9-	F++C, BA+8	-65-, --1-	7++4, 32+0
99	63	-ED-, --98	F++C, BA++	-65-, --10	7++4, 32++
100	64	-ED-, -A--	F++C, B+98	-65-, -2--	7++4, 3+10
101	65	-ED-, -A-8	F++C, B+9+	-65-, -2-0	7++4, 3+1+
102	66	-ED-, -A9-	F++C, B++8	-65-, -21-	7++4, 3++0
103	67	-ED-, -A98	F++C, B+++	-65-, -210	7++4, 3+++
104	68	-ED-, B---	F++C, +A98	-65-, 3---	7++4, +210
105	69	-ED-, B--8	F++C, +A9+	-65-, 3--0	7++4, +21+
106	6A	-ED-, B-9-	F++C, +A+8	-65-, 3-1-	7++4, +2+0
107	6B	-ED-, B-98	F++C, +A++	-65-, 3-10	7++4, +2++
108	6C	-ED-, BA--	F++C, ++98	-65-, 32--	7++4, ++10
109	6D	-ED-, BA-8	F++C, ++9+	-65-, 32-0	7++4, ++1+
110	6E	-ED-, BA9-	F++C, +++8	-65-, 321-	7++4, +++0
111	6F	-ED-, BA98	F++C, ++++	-65-, 3210	7++4, ++++
112	70	-EDC, ----	F+++ , BA98	-654, ----	7+++ , 3210
113	71	-EDC, ---8	F+++ , BA9+	-654, ---0	7+++ , 321+
114	72	-EDC, --9-	F+++ , BA+8	-654, --1-	7+++ , 32+0
115	73	-EDC, --98	F+++ , BA++	-654, --10	7+++ , 32++
116	74	-EDC, -A--	F+++ , B+98	-654, -2--	7+++ , 3+10
117	75	-EDC, -A-8	F+++ , B+9+	-654, -2-0	7+++ , 3+1+
118	76	-EDC, -A9-	F+++ , B++8	-654, -21-	7+++ , 3++0
119	77	-EDC, -A98	F+++ , B+++	-654, -210	7+++ , 3+++
120	78	-EDC, B---	F+++ , +A98	-654, 3---	7+++ , +210
121	79	-EDC, B--8	F+++ , +A9+	-654, 3--0	7+++ , +21+
122	7A	-EDC, B-9-	F+++ , +A+8	-654, 3-1-	7+++ , +2+0
123	7B	-EDC, B-98	F+++ , +A++	-654, 3-10	7+++ , +2++
124	7C	-EDC, BA--	F+++ , ++98	-654, 32--	7+++ , ++10
125	7D	-EDC, BA-8	F+++ , ++9+	-654, 32-0	7+++ , ++1+
126	7E	-EDC, BA9-	F+++ , +++8	-654, 321-	7+++ , +++0
127	7F	-EDC, BA98	F+++ , ++++	-654, 3210	7+++ , ++++
128	80	F---, ----	+EDC, BA98	7---, ----	+654, 3210
129	81	F---, ---8	+EDC, BA9+	7---, ---0	+654, 321+
130	82	F---, --9-	+EDC, BA+8	7---, --1-	+654, 32+0
131	83	F---, --98	+EDC, BA++	7---, --10	+654, 32++
132	84	F---, -A--	+EDC, B+98	7---, -2--	+654, 3+10
133	85	F---, -A-8	+EDC, B+9+	7---, -2-0	+654, 3+1+
134	86	F---, -A9-	+EDC, B++8	7---, -21-	+654, 3++0
135	87	F---, -A98	+EDC, B+++	7---, -210	+654, 3+++
136	88	F---, B---	+EDC, +A98	7---, 3---	+654, +210
137	89	F---, B--8	+EDC, +A9+	7---, 3--0	+654, +21+

BIT MAP BYTE VALUE		RELATIVE BYTE #0 OF TK BIT MAPS IN VTOC SECTOR		RELATIVE #1 OF TK BIT MAPS IN VTOR SECTOR	
DEC	HEX	FREE	USED	FREE	USED
138	8A	F---, B-9-	+EDC, +A+8	7---, 3-1-	+654, +2+0
139	8B	F---, B-98	+EDC, +A++	7---, 3-10	+654, +2++
140	8C	F---, BA--	+EDC, ++98	7---, 32--	+654, ++10
141	8D	F---, BA-8	+EDC, ++9+	7---, 32-0	+654, ++1+
142	8E	F---, BA9-	+EDC, +++8	7---, 321-	+654, +++0
143	8F	F---, BA98	+EDC, ++++	7---, 3210	+654, ++++
144	90	F--C, ----	+ED+, BA98	7--4, ----	+65+, 3210
145	91	F--C, ---8	+ED+, BA9+	7--4, ---0	+65+, 321+
146	92	F--C, --9-	+ED+, BA+8	7--4, --1-	+65+, 32+0
147	93	F--C, --98	+ED+, BA++	7--4, --10	+65+, 32++
148	94	F--C, -A--	+ED+, B+98	7--4, -2--	+65+, 3+10
149	95	F--C, -A-8	+ED+, B+9+	7--4, -2-0	+65+, 3+1+
150	96	F--C, -A9-	+ED+, B++8	7--4, -21-	+65+, 3++0
151	97	F--C, -A98	+ED+, B+++	7--4, -210	+65+, 3+++
152	98	F--C, B---	+ED+, +A98	7--4, 3---	+65+, +210
153	99	F--C, B--8	+ED+, +A9+	7--4, 3--0	+65+, +21+
154	9A	F--C, B-9-	+ED+, +A+8	7--4, 3-1-	+65+, +2+0
155	9B	F--C, B-98	+ED+, +A++	7--4, 3-10	+65+, +2++
156	9C	F--C, BA--	+ED+, ++98	7--4, 32--	+65+, ++10
157	9D	F--C, BA-8	+ED+, ++9+	7--4, 32-0	+65+, ++1+
158	9E	F--C, BA9-	+ED+, +++8	7--4, 321-	+65+, +++0
159	9F	F--C, BA98	+ED+, ++++	7--4, 3210	+65+, ++++
160	A0	F-D-, ----	+E+C, BA98	7-5-, ----	+6+4, 3210
161	A1	F-D-, ---8	+E+C, BA9+	7-5-, ---0	+6+4, 321+
162	A2	F-D-, --9-	+E+C, BA+8	7-5-, --1-	+6+4, 32+0
163	A3	F-D-, --98	+E+C, BA++	7-5-, --10	+6+4, 32++
164	A4	F-D-, -A--	+E+C, B+98	7-5-, -2--	+6+4, 3+10
165	A5	F-D-, -A-8	+E+C, B+9+	7-5-, -2-0	+6+4, 3+1+
166	A6	F-D-, -A9-	+E+C, B++8	7-5-, -21-	+6+4, 3++0
167	A7	F-D-, -A98	+E+C, B+++	7-5-, -210	+6+4, 3+++
168	A8	F-D-, B---	+E+C, +A98	7-5-, 3---	+6+4, +210
169	A9	F-D-, B--8	+E+C, +A9+	7-5-, 3--0	+6+4, +21+
170	AA	F-D-, B-9-	+E+C, +A+8	7-5-, 3-1-	+6+4, +2+0
171	AB	F-D-, B-98	+E+C, +A++	7-5-, 3-10	+6+4, +2++
172	AC	F-D-, BA--	+E+C, ++98	7-5-, 32--	+6+4, ++10
173	AD	F-D-, BA-8	+E+C, ++9+	7-5-, 32-0	+6+4, ++1+
174	AE	F-D-, BA9-	+E+C, +++8	7-5-, 321-	+6+4, +++0
175	AF	F-D-, BA98	+E+C, ++++	7-5-, 3210	+6+4, ++++
176	B0	F-DC, ----	+E+++, BA98	7-54, ----	+6+++, 3210
177	B1	F-DC, ---8	+E+++, BA9+	7-54, ---0	+6+++, 321+
178	B2	F-DC, --9-	+E+++, BA+8	7-54, --1-	+6+++, 32+0
179	B3	F-DC, --98	+E+++, BA++	7-54, --10	+6+++, 32++
180	B4	F-DC, -A--	+E+++, B+98	7-54, -2--	+6+++, 3+10
181	B5	F-DC, -A-8	+E+++, B+9+	7-54, -2-0	+6+++, 3+1+
182	B6	F-DC, -A9-	+E+++, B++8	7-54, -21-	+6+++, 3++0
183	B7	F-DC, -A98	+E+++, B+++	7-54, -210	+6+++, 3+++
184	B8	F-DC, B---	+E+++, +A98	7-54, 3---	+6+++, +210
185	B9	F-DC, B--8	+E+++, +A9+	7-54, 3--0	+6+++, +21+
186	BA	F-DC, B-9-	+E+++, +A+8	7-54, 3-1-	+6+++, +2+0

BIT MAP BYTE VALUE		RELATIVE BYTE #0 OF TK BIT MAPS IN VTOC SECTOR		RELATIVE #1 OF TK BIT MAPS IN VTOR SECTOR	
DEC	HEX	FREE	USED	FREE	USED
187	BB	F-DC, B-98	+E+, +A++	7-54, 3-10	+6+, +2++
188	BC	F-DC, BA--	+E+, ++98	7-54, 32--	+6+, ++10
189	BD	F-DC, BA-8	+E+, ++9+	7-54, 32-0	+6+, ++1+
190	BE	F-DC, BA9-	+E+, +++8	7-54, 321-	+6+, +++0
191	BF	F-DC, BA98	+E+, ++++	7-54, 3210	+6+, ++++
192	C0	FE--, ----	+DC, BA98	76--, ----	++54, 3210
193	C1	FE--, ---8	+DC, BA9+	76--, ---0	++54, 321+
194	C2	FE--, --9-	+DC, BA+8	76--, --1-	++54, 32+0
195	C3	FE--, --98	+DC, BA++	76--, --10	++54, 32++
196	C4	FE--, -A--	+DC, B+98	76--, -2--	++54, 3+10
197	C5	FE--, -A-8	+DC, B+9+	76--, -2-0	++54, 3+1+
198	C6	FE--, -A9-	+DC, B++8	76--, -21-	++54, 3++0
199	C7	FE--, -A98	+DC, B+++	76--, -210	++54, 3+++
200	C8	FE--, B---	+DC, +A98	76--, 3---	++54, +210
201	C9	FE--, B--8	+DC, +A9+	76--, 3--0	++54, +21+
202	CA	FE--, B-9-	+DC, +A+8	76--, 3-1-	++54, +2+0
203	CB	FE--, B-98	+DC, +A++	76--, 3-10	++54, +2++
204	CC	FE--, BA--	+DC, ++98	76--, 32--	++54, ++10
205	CD	FE--, BA-8	+DC, ++9+	76--, 32-0	++54, ++1+
206	CE	FE--, BA9-	+DC, +++8	76--, 321-	++54, +++0
207	CF	FE--, BA98	+DC, ++++	76--, 3210	++54, ++++
208	D0	FE-C, ----	+D+, BA98	76-4, ----	++5+, 3210
209	D1	FE-C, ---8	+D+, BA9+	76-4, ---0	++5+, 321+
210	D2	FE-C, --9-	+D+, BA+8	76-4, --1-	++5+, 32+0
211	D3	FE-C, --98	+D+, BA++	76-4, --10	++5+, 32++
212	D4	FE-C, -A--	+D+, B+98	76-4, -2--	++5+, 3+10
213	D5	FE-C, -A-8	+D+, B+9+	76-4, -2-0	++5+, 3+1+
214	D6	FE-C, -A9-	+D+, B++8	76-4, -21-	++5+, 3++0
215	D7	FE-C, -A98	+D+, B+++	76-4, -210	++5+, 3+++
216	D8	FE-C, B---	+D+, +A98	76-4, 3---	++5+, +210
217	D9	FE-C, B--8	+D+, +A9+	76-4, 3--0	++5+, +21+
218	DA	FE-C, B-9-	+D+, +A+8	76-4, 3-1-	++5+, +2+0
219	DB	FE-C, B-98	+D+, +A++	76-4, 3-10	++5+, +2++
220	DC	FE-C, BA--	+D+, ++98	76-4, 32--	++5+, ++10
221	DD	FE-C, BA-8	+D+, ++9+	76-4, 32-0	++5+, ++1+
222	DE	FE-C, BA9-	+D+, +++8	76-4, 321-	++5+, +++0
223	DF	FE-C, BA98	+D+, ++++	76-4, 3210	++5+, ++++
224	E0	FED-, ----	++C, BA98	765-, ----	++4+, 3210
225	E1	FED-, ---8	++C, BA9+	765-, ---0	++4+, 321+
226	E2	FED-, --9-	++C, BA+8	765-, --1-	++4+, 32+0
227	E3	FED-, --98	++C, BA++	765-, --10	++4+, 32++
228	E4	FED-, -A--	++C, B+98	765-, -2--	++4+, 3+10
229	E5	FED-, -A-8	++C, B+9+	765-, -2-0	++4+, 3+1+
230	E6	FED-, -A9-	++C, B++8	765-, -21-	++4+, 3++0
231	E7	FED-, -A98	++C, B+++	765-, -210	++4+, 3+++
232	E8	FED-, B---	++C, +A98	765-, 3---	++4+, +210
233	E9	FED-, B--8	++C, +A9+	765-, 3--0	++4+, +21+
234	EA	FED-, B-9-	++C, +A+8	765-, 3-1-	++4+, +2+0
235	EB	FED-, B-98	++C, +A++	765-, 3-10	++4+, +2++

BIT MAP BYTE VALUE		RELATIVE BYTE #0 OF TK BIT MAPS IN VTOC SECTOR		RELATIVE #1 OF TK BIT MAPS IN VTOR SECTOR	
DEC	HEX	FREE	USED	FREE	USED
236	EC	FED-, BA--	+++C, ++98	765-, 32--	+++4, ++10
237	ED	FED-, BA-8	+++C, ++9+	765-, 32-0	+++4, ++1+
238	EE	FED-, BA9-	+++C, +++8	765-, 321-	+++4, +++0
239	EF	FED-, BA98	+++C, ++++	765-, 3210	+++4, ++++
240	F0	FEDC, ----	++++, BA98	7654, ----	++++, 3210
241	F1	FEDC, ---8	++++, BA9+	7654, ---0	++++, 321+
242	F2	FEDC, --9-	++++, BA+8	7654, --1-	++++, 32+0
243	F3	FEDC, --98	++++, BA++	7654, --10	++++, 32++
244	F4	FEDC, -A--	++++, B+98	7654, -2--	++++, 3+10
245	F5	FEDC, -A-8	++++, B+9+	7654, -2-0	++++, 3+1+
246	F6	FEDC, -A9-	++++, B++8	7654, -21-	++++, 3++0
247	F7	FEDC, -A98	++++, B+++	7654, -210	++++, 3+++
248	F8	FEDC, B---	++++, +A98	7654, 3---	++++, +210
249	F9	FEDC, B--8	++++, +A9+	7654, 3--0	++++, +21+
250	FA	FEDC, B-9-	++++, +A+8	7654, 3-1-	++++, +2+0
251	FB	FEDC, B-98	++++, +A++	7654, 3-10	++++, +2++
252	FC	FEDC, BA--	++++, ++98	7654, 32--	++++, ++10
253	FD	FEDC, BA-8	++++, ++9+	7654, 32-0	++++, ++1+
254	FE	FEDC, BA9-	++++, +++8	7654, 321-	++++, +++0
255	FF	FEDC, BA98	++++, ++++	7654, 3210	++++, ++++

CATALOG/DIRECTORY SECTORS CHART

This chart can be used to locate information in the Catalog/Directory on Track \$11 (17 decimal) Sectors \$F to \$1 (15 to 1 decimal) on diskette and in memory. The relative byte offsets and Catalog/Directory Sector Buffer in DOS (RAM) are given in both decimal and hex. The storage locations of sectors 1 through 15 are given in hex only in the TTSBB (Track Sector Byte) notation.

At the end of the chart (the column to the right of sector 15) is the Catalog/Directory sector buffer in DOS (RAM). Since the second chart in this pair is simply a horizontal extension of the first, cross indexing can extend from this last column up to the first chart.

Sectors 1-9										
REL BYTE OFFSET		1ST	2ND	3RD	4TH	5TH	6TH	7TH	8TH	9TH
		TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB
DEC	HEX									
0	0	11F00	11E00	11D00	11C00	11B00	11A00	11900	11800	11700
1	1	11F01	11E01	11D01	11C01	11B01	11A01	11901	11801	11701
2	2	11F02	11E02	11D02	11C02	11B02	11A02	11902	11802	11702
3	3	11F03	11E03	11D03	11C03	11B03	11A03	11903	11803	11703
10	A	11F0A	11E0A	11D0A	11C0A	11B0A	11A0A	1190A	1180A	1170A
11	B	11F0B	11E0B	11D0B	11C0B	11B0B	11A0B	1190B	1180B	1170B
45	2D	11F2D	11E2D	11D2D	11C2D	11B2D	11A2D	1192D	1182D	1172D
46	2E	11F2E	11E2E	11D2E	11C2E	11B2E	11A2E	1192E	1182E	1172E
80	50	11F50	11E50	11D50	11C50	11B50	11A50	11950	11850	11750
81	51	11F51	11E51	11D51	11C51	11B51	11A51	11951	11851	11751
115	73	11F73	11E73	11D73	11C73	11B73	11A73	11973	11873	11773
116	74	11F74	11E74	11D74	11C74	11B74	11A74	11974	11874	11774
150	96	11F96	11E96	11D96	11C96	11B96	11A96	11996	11896	11796
151	97	11F97	11E97	11D97	11C97	11B97	11A97	11997	11897	11797
185	B9	11FB9	11EB9	11DB9	11CB9	11BB9	11AB9	119B9	118B9	117B9
186	BA	11FBA	11EBA	11DBA	11CBA	11BBA	11ABA	119BA	118BA	117BA
220	DC	11FDC	11EDC	11DDC	11CDC	11BDC	11ADC	119DC	118DC	117DC
221	DD	11FDD	11EDD	11DDD	11CDD	11BDD	11ADD	119DD	118DD	117DD
255	FF	11FFF	11EFF	11DFF	11CFF	11BFF	11AFF	119FF	118FF	117FF

		Sectors 10-15							
REL BYTE	10TH	11TH	12TH	13TH	14TH	15TH	CAT/DIR SEC		
OFFSET							BUF DOX (RAM		
		TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB		
DEC	HEX							DEX	HEX
0	0	11600	11500	11400	11300	11200	11100	46,267	B4BB
1	1	11601	11501	11401	11301	11201	11101	46,268	B4BC
2	2	11602	11502	11402	11302	11202	11102	46,269	B4BD
3	3	11603	11503	11403	11303	11203	11103	46,270	B4BE
10	A	1160A	1150A	1140A	1130A	1120A	1110A	46,277	B4C5
11	B	1160B	1150B	1140B	1130B	1120B	1110B	46,278	B4C6
45	2D	1162D	1152D	1142D	1132D	1122D	1112D	46,312	B4E8
46	2E	1162E	1152E	1142E	1132E	1122E	1112E	46,313	B4E9
80	50	11650	11550	11450	11350	11250	11150	46,347	B50B
81	51	11651	11551	11451	11351	11251	11151	46,348	B50C
115	73	11673	11573	11473	11373	11273	11173	46,382	B52E
116	74	11674	11574	11474	11374	11274	11174	46,383	B52F
150	96	11696	11596	11496	11396	11296	11196	46,417	B551
151	97	11697	11597	11497	11397	11297	11197	46,418	B552
185	B9	116B9	115B9	114B9	113B9	112B9	111B9	46,452	B574
186	BA	116BA	115BA	114BA	113BA	112BA	111BA	46,453	B575
220	DC	116DC	115DC	114DC	113DC	112DC	111DC	46,487	B597
221	DD	116DD	115DD	114DD	113DD	112DD	111DD	46,488	B598
255	FF	116FF	115FF	114FF	113FF	112FF	111FF	46,522	B5BA

FILE NAME ENTRY CHART

This chart helps in reading/modifying file names on your disks. Each of 105 possible file names is associated with certain storage locations in the catalog/directory sectors.

C/D FILE NAME # = CATALOG/DIRECTORY FILE NAME NUMBER
 TRK # 1ST TSL = TRACK NUMBER OF FIRST TRACK SECTOR LIST
 (*Relative byte offset #0 gets put into relative byte #20 when file is deleted)
 SEC # 1ST TSL = SECTOR NUMBER OF FIRST TRACK SECTOR LIST
 FI TYP & LOCK STATUS = FILE TYPE AND LOCK STATUS

If locked \$80 is added to file type

Unlocked files have \$0 added.

\$00 = Text file

\$01 = INTEGER BASIC file

\$02 = APPLESOFT BASIC file

\$04 = BINARY file

\$08 = S file

\$10 = R file

\$20 = A type file

\$40 = B type file

ST OF FI NA = START OF FILE NAME

END OF FI NA = END OF FILE- NAME

(**Relative byte #20, the 30th character of the file name, is lost forever if file is deleted.)

LEN FI LO = LENGTH OF FILE LOW BYTE

LEN FI HI = LENGTH OF FILE HIGH BYTE

(Also see the FILE CLASSIFICATION CHART)

The bottom portion of the chart gives the RAM locations for the same information (current sector examined/used.)

C/D FILE NAME #	TRK # 1st TSL*	SEC # 1st TSL	FI TYP & LOCK STATUS	ST OF FI NA	END OF FI NA **	LEN FI LO	LEN FI HI
	0/\$0	1/\$1	2/\$2	3/\$3	32/\$20	33/\$21	34/\$22
	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB
1st	11F0B	11F0C	11F0D	11F0E	11F2B	11F2C	11F2D
2nd	11F2E	11F2F	11F30	11F31	11F4E	11F4F	11F50
3rd	11F51	11F52	11F53	11F54	11F71	11F72	11F73
4th	11F74	11F75	11F76	11F77	11F94	11F95	11F96
5th	11F97	11F98	11F99	11F9A	11FB7	11FB8	11FB9
6th	11FBA	11FBB	11FBC	11FBD	11FDA	11FDB	11FDC
7th	11FDD	11FDE	11FDF	11FE0	11FFD	11FFE	11FFE
8th	11E0B	11E0C	11E0D	11E0E	11E2B	11E2C	11E2D
9th	11E2E	11E2F	11E30	11E31	11E4E	11E4F	11E50
10th	11E51	11E52	11E53	11E54	11E71	11E72	11E73
11th	11E74	11E75	11E76	11E77	11E94	11E95	11E96
12th	11E97	11E98	11E99	11E9A	11EB7	11EB8	11EB9
13th	11EBA	11EBB	11EBC	11EBD	11EDA	11EDB	11EDC

C/D FILE TRK #	SEC #	FI TYP	ST OF	END OF	LEN	LEN	
NAME # 1st	1st	& LOCK	FI NA	FI NA	FI LO	FI HI	
TSL*	TSL	STATUS		**			
0/\$0	1/\$1	2/\$2	3/\$3	32/\$20	33/\$21	34/\$22	
TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	
14th	11EDD	11EDE	11EDF	11EE0	11EFD	11EFE	11EFF
15th	11D0B	11D0C	11D0D	11D0E	11D2B	11D2C	11D2D
16th	11D2E	11D2F	11D30	11D31	11D4E	11D4F	11D50
17th	11D51	11D52	11D53	11D54	11D71	11D72	11D73
18th	11D74	11D75	11D76	11D77	11D94	11D95	11D96
19th	11D97	11D98	11D99	11D9A	11DB7	11DB8	11DB9
20th	11DBA	11DBB	11DBC	11DBD	11DDA	11DDB	11DDC
21st	11DDD	11DDE	11DDF	11DE0	11DFD	11DFE	11DFF
22nd	11C0B	11C0C	11C0D	11C0E	11C2B	11C2C	11C2D
23rd	11C2E	11C2F	11C30	11C31	11C4E	11C4F	11C50
24th	11C51	11C52	11C53	11C54	11C71	11C72	11C73
25th	11C74	11C75	11C76	11C77	11C94	11C95	11C96
26th	11C97	11C98	11C99	11C9A	11CB7	11CB8	11CB9
27th	11CBA	11CBB	11CBC	11CBD	11CDA	11CDB	11CDC
28th	11CDD	11CDE	11CDF	11CE0	11CFD	11CFE	11CFF
29th	11B0B	11B0C	11B0D	11B0E	11B2B	11B2C	11B2D
30th	11B2E	11B2F	11B30	11B31	11B4E	11B4F	11B50
31st	11B51	11B52	11B53	11B54	11B71	11B72	11B73
32nd	11B74	11B75	11B76	11B77	11B94	11B95	11B96
33rd	11B97	11B98	11B99	11B9A	11BB7	11BB8	11BB9
34th	11BBA	11BBB	11BBC	11BBD	11BDA	11BDB	11BDC
35th	11BDD	11BDE	11BDF	11BE0	11BFD	11BFE	11BFF
36th	11A0B	11A0C	11A0D	11A0E	11A2B	11A2C	11A2D
37th	11A2E	11A2F	11A30	11A31	11A4E	11A4F	11A50
38th	11A51	11A52	11A53	11A54	11A71	11A72	11A73
39th	11A74	11A75	11A76	11A77	11A94	11A95	11A96
40th	11A97	11A98	11A99	11A9A	11AB7	11AB8	11AB9
41st	11ABA	11ABB	11ABC	11ABD	11ADA	11ADB	11ADC
42nd	11ADD	11ADE	11ADF	11AE0	11AFD	11AFE	11AFF
43rd	1190B	1190C	1190D	1190E	1192B	1192C	1192D
44th	1192E	1192F	11930	11931	1194E	1194F	11950
45th	11951	11952	11953	11954	11971	11972	11973
46th	11974	11975	11976	11977	11994	11995	11996
47th	11997	11998	11999	1199A	119B7	119B8	119B9
48th	119BA	119BB	119BC	119BD	119DA	119DB	119DC
49th	119DD	119DE	119DF	119E0	119FD	119FE	119FF
50th	1180B	1180C	1180D	1180E	1182B	1182C	1182D
51st	1182E	1182F	11830	11831	1184E	1184F	11850
52nd	11851	11852	11853	11854	11871	11872	11873
53rd	11874	11875	11876	11877	11894	11895	11896
54th	11897	11898	11899	1189A	118B7	118B8	118B9
55th	118BA	118BB	118BC	118BD	118DA	118DB	118DC
56th	118DD	118DE	118DF	118E0	118FD	118FE	118FF
57th	1170B	1170C	1170D	1170E	1172B	1172C	1172D
58th	1172E	1172F	11730	11731	1174E	1174F	11750
59th	11751	11752	11753	11754	11771	11772	11773
60th	11774	11775	11776	11777	11794	11795	11796
61st	11797	11798	11799	1179A	117B7	117B8	117B9
62nd	117BA	117BB	117BC	117BD	117DA	117DB	117DC
63rd	117DD	117DE	117DF	117E0	117FD	117FE	117FF

C/D FILE TRK #	SEC #	FI TYP	ST OF	END OF	LEN	LEN	
NAME # 1st	1st	& LOCK	FI NA	FI NA	FI LO	FI HI	
TSL*	TSL	STATUS		**			
0/\$0	1/\$1	2/\$2	3/\$3	32/\$20	33/\$21	34/\$22	
TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	TTSBB	
64th	1160B	1160C	1160D	1160E	1162B	1162C	1162D
65th	1162E	1162F	11630	11631	1164E	1164F	11650
66th	11651	11652	11653	11654	11671	11672	11673
67th	11674	11675	11676	11677	11694	11695	11696
68th	11697	11698	11699	1169A	116B7	116B8	116B9
69th	116BA	116BB	116BC	116BD	116DA	116DB	116DC
70th	116DD	116DE	116DF	116E0	116FD	116FE	116FF
71st	1150B	1150C	1150D	1150E	1152B	1152C	1152D
72nd	1152E	1152F	11530	11531	1154E	1154F	11550
73rd	11551	11552	11553	11554	11571	11572	11573
74th	11574	11575	11576	11577	11594	11595	11596
75th	11597	11598	11599	1159A	115B7	115B8	115B9
76th	115BA	115BB	115BC	115BD	115DA	115DB	115DC
77th	115DD	115DE	115DF	115E0	115FD	115FE	115FF
78th	1140B	1140C	1140D	1140E	1142B	1142C	1142D
79th	1142E	1142F	11430	11431	1144E	1144F	11450
80th	11451	11452	11453	11454	11471	11472	11473
81st	11474	11475	11476	11477	11494	11495	11496
82nd	11497	11498	11499	1149A	114B7	114B8	114B9
83rd	114BA	114BB	114BC	114BD	114DA	114DB	114DC
84th	114DD	114DE	114DF	114E0	114FD	114FE	114FF
85th	1130B	1130C	1130D	1130E	1132B	1132C	1132D
86th	1132E	1132F	11330	11331	1134E	1134F	11350
87th	11351	11352	11353	11354	11371	11372	11373
88th	11374	11375	11376	11377	11394	11395	11396
89th	11397	11398	11399	1139A	113B7	113B8	113B9
90th	113BA	113BB	113BC	113BD	113DA	113DB	113DC
91st	113DD	113DE	113DF	113E0	113FD	113FE	113FF
92nd	1120B	1120C	1120D	1120E	1122B	1122C	1122D
93rd	1122E	1122F	11230	11231	1124E	1124F	11250
94th	11251	11252	11253	11254	11271	11272	11273
95th	11274	11275	11276	11277	11294	11295	11296
96th	11297	11298	11299	1129A	112B7	112B8	112B9
97th	112BA	112BB	112BC	112BD	112DA	112DB	112DC
98th	112DD	112DE	112DF	112E0	112FD	112FE	112FF
99th	1110B	1110C	1110D	1110E	1112B	1112C	1112D
100th	1112E	1112F	11130	11131	1114E	1114F	11150
101st	11151	11152	11153	11154	11171	11172	11173
102nd	11174	11175	11176	11177	11194	11195	11196
103rd	11197	11198	11199	1119A	111B7	111B8	111B9
104th	111BA	111BB	111BC	111BD	111DA	111DB	111DC
105th	111DD	111DE	111DF	111E0	111FD	111FE	111FF

	RAM ADDRESS	RAM ADDRESS	RAM ADDRESS	RAM ADDRESS	RAM ADDRESS	RAM ADDRESS	RAM ADDRESS
DEC 1	46,278	46,279	46,280	46,281	46,310	46,311	46,312
HEX 1	B4C6	B4C7	B4C8	B4C9	B4E6	B4E7	B4E8
DEC 2	46,313	46,314	46,315	46,316	46,345	46,346	46,347
HEX 2	B4E9	B4EA	B4EB	B4EC	B509	B50A	B50B
DEC 3	46,348	46,349	46,350	46,351	46,380	46,381	46,382
HEX 3	B50C	B50D	B50E	B50F	B52C	B52D	B52E
DEC 4	46,383	46,384	46,385	46,386	46,415	46,416	46,417
HEX 4	B52F	B530	B531	B532	B54F	B550	B551
DEC 5	46,418	46,419	46,420	46,421	46,450	46,451	46,452
HEX 5	B552	B553	B554	B555	B572	B573	B574
DEC 6	46,453	46,454	46,455	46,456	46,485	46,486	46,487
HEX 6	B575	B576	B577	B578	B595	B596	B597
DEC 7	46,488	46,489	46,490	46,491	46,520	46,521	46,522
HEX 7	B598	B599	B59A	B59B	B5B8	B5B9	B5BA

FILE CLASSIFICATION CHART

On the FILE NAME ENTRY CHART, there is an abbreviated list of how files are classified in a catalog sector. This chart is simply an expanded description of that classification

FI CLASS FILE CLASSIFICATION
BYTE VAL

DEC HEX

0	0	Unlocked Text File
1	1	Unlocked Integer BASIC File
2	2	Unlocked Applesoft BASIC File
4	4	Unlocked Binary File
8	8	Unlocked S (unused file type)
16	10	Unlocked Relocatable 6502 Code File
32	20	Unlocked A (unused file type)
64	40	Unlocked B (unused file type)
128	80	*Locked Text File
129	81	*Locked Integer BASIC File
130	82	*Locked Applesoft BASIC File
132	84	*Locked Binary File
136	88	*Locked S (unused file type)
144	90	*Locked Relocatable 6502 Code File
160	A0	*Locked A (unused file type)
192	C0	*Locked B (unused file type)

TEXT and LOW RESOLUTION SCREEN CHARACTER CODES (40 COLUMN MODE)

This chart can be used for either low resolution colors or text characters, depending on whether you want to use screen memory for lores graphics or text. The "TXT" columns refer to different kinds of text output to the screen with "I" indicating inverse, "F" flashing and "N" normal. Top and bottom lores blocks indicate the color of the blocks on the lores screen. Try POKEing in different decimal values to the screen addresses used for text (1024-2039) after entering GR to see the different color combinations. For example, try the following one line program:

```
10 GR : POKE 1335, 156
```

(NOTE: The symbols for codes 224, 251-255 may be different with different character generators.)

BOTTOM LORES BLOCK					TOP LORES BLOCK				
DEC	HEX	TXT	BLOCK		DEC	HEX	TXT	BLOCK	
0	0	@I	BLACK0	BLACK0	128	80	@N	BROWN8	BLACK0
1	1	AI	BLACK0	MAGENTA1	129	81	AN	BROWN8	MAGENTA1
2	2	BI	BLACK0	D BLUE2	130	82	BN	BROWN8	D BLUE2
3	3	CI	BLACK0	PURPLE3	131	83	CN	BROWN8	PURPLE3
4	4	DI	BLACK0	D GREEN4	132	84	DN	BROWN8	D GREEN4
5	5	EI	BLACK0	GREY5	133	85	EN	BROWN8	GREY5
6	6	FI	BLACK0	M BLUE6	134	86	FN	BROWN8	M BLUE6
7	7	GI	BLACK0	L BLUE7	135	87	GN	BROWN8	L BLUE7
8	8	HI	BLACK0	BROWN8	136	88	HN	BROWN8	BROWN8
9	9	II	BLACK0	ORANGE9	137	89	IN	BROWN8	ORANGE9
10	A	JI	BLACK0	GREY10	138	8A	JN	BROWN8	GREY10
11	B	KI	BLACK0	PINK11	139	8B	KN	BROWN8	PINK11
12	C	LI	BLACK0	GREEN12	140	8C	LN	BROWN8	GREEN12
13	D	MI	BLACK0	YELLOW13	141	8D	MN	BROWN8	YELLOW13
14	E	NI	BLACK0	AQUA14	142	8E	NN	BROWN8	AQUA14
15	F	OI	BLACK0	WHITE15	143	8F	ON	BROWN8	WHITE15
16	10	PI	MAGENTA1	BLACK0	144	90	PN	ORANGE9	BLACK0
17	11	QI	MAGENTA1	MAGENTA1	145	91	QN	ORANGE9	MAGENTA1
18	12	RI	MAGENTA1	D BLUE2	146	92	RN	ORANGE9	D BLUE2
19	13	SI	MAGENTA1	PURPLE3	147	93	SN	ORANGE9	PURPLE3
20	14	TI	MAGENTA1	D GREEN4	148	94	TN	ORANGE9	D GREEN4
21	15	UI	MAGENTA1	GREY5	149	95	UN	ORANGE9	GREY5
22	16	VI	MAGENTA1	M BLUE6	150	96	VN	ORANGE9	M BLUE6
23	17	WI	MAGENTA1	L BLUE7	151	97	WN	ORANGE9	L BLUE7
24	18	XI	MAGENTA1	BROWN8	152	98	XN	ORANGE9	BROWN8
25	19	YI	MAGENTA1	ORANGE9	153	99	YN	ORANGE9	ORANGE9
26	1A	ZI	MAGENTA1	GREY10	154	9A	ZN	ORANGE9	GREY10
27	1B	[I	MAGENTA1	PINK11	155	9B	[N	ORANGE9	PINK11
28	1C	\I	MAGENTA1	GREEN12	156	9C	\N	ORANGE9	GREEN12
29	1D	I	MAGENTA1	YELLOW13	157	9D	N	ORANGE9	YELLOW13
30	1E	^I	MAGENTA1	AQUA14	158	9E	^N	ORANGE9	AQUA14
31	1F	_I	MAGENTA1	WHITE15	159	9F	_N	ORANGE9	WHITE15

BOTTOM LORES DEC HEX TXT BLOCK				TOP LORES BLOCK	BOTTOM LORES DEC HEX TXT BLOCK				TOP LORES BLOCK
32	20	I	D BLUE2	BLACK0	160	A0	N	GREY10	BLACK0
33	21	!I	D BLUE2	MAGENTA1	161	A1	!N	GREY10	MAGENTA1
34	22	"I	D BLUE2	D BLUE2	162	A2	"N	GREY10	D BLUE2
35	23	#I	D BLUE2	PURPLE3	163	A3	#N	GREY10	PURPLE3
36	24	\$I	D BLUE2	D GREEN4	164	A4	\$N	GREY10	D GREEN4
37	25	%I	D BLUE2	GREY5	165	A5	%N	GREY10	GREY5
38	26	&I	D BLUE2	M BLUE6	166	A6	&N	GREY10	M BLUE6
39	27	'I	D BLUE2	L BLUE7	167	A7	'N	GREY10	L BLUE7
40	28	(I	D BLUE2	BROWN8	168	A8	(N	GREY10	BROWN8
41	29	)I	D BLUE2	ORANGE9	169	A9	)N	GREY10	ORANGE9
42	2A	*I	D BLUE2	GREY10	170	AA	*N	GREY10	GREY10
43	2B	+I	D BLUE2	PINK11	171	AB	+N	GREY10	PINK11
44	2C	,I	D BLUE2	GREEN12	172	AC	,N	GREY10	GREEN12
45	2D	-I	D BLUE2	YELLOW13	173	AD	-N	GREY10	YELLOW13
46	2E	.I	D BLUE2	AQUA14	174	AE	.N	GREY10	AQUA14
47	2F	/I	D BLUE2	WHITE15	175	AF	/N	GREY10	WHITE15
48	30	0I	PURPLE3	BLACK0	176	B0	0N	PINK11	BLACK0
49	31	1I	PURPLE3	MAGENTA1	177	B1	1N	PINK11	MAGENTA1
50	32	2I	PURPLE3	D BLUE2	178	B2	2N	PINK11	D BLUE2
51	33	3I	PURPLE3	PURPLE3	179	B3	3N	PINK11	PURPLE3
52	34	4I	PURPLE3	D GREEN4	180	B4	4N	PINK11	D GREEN4
53	35	5I	PURPLE3	GREY5	181	B5	5N	PINK11	GREY5
54	36	6I	PURPLE3	M BLUE6	182	B6	6N	PINK11	M BLUE6
55	37	7I	PURPLE3	L BLUE7	183	B7	7N	PINK11	L BLUE7
56	38	8I	PURPLE3	BROWN8	184	B8	8N	PINK11	BROWN8
57	39	9I	PURPLE3	ORANGE9	185	B9	9N	PINK11	ORANGE9
58	3A	:I	PURPLE3	GREY10	186	BA	:N	PINK11	GREY10
59	3B	;I	PURPLE3	PINK11	187	BB	;N	PINK11	PINK11
60	3C	<I	PURPLE3	GREEN12	188	BC	<N	PINK11	GREEN12
61	3D	=I	PURPLE3	YELLOW13	189	BD	=N	PINK11	YELLOW13
62	3E	>I	PURPLE3	AQUA14	190	BE	>N	PINK11	AQUA14
63	3F	?I	PURPLE3	WHITE15	191	BF	?N	PINK11	WHITE15
64	40	@F	D GREEN4	BLACK0	192	C0	@N	GREEN12	BLACK0
65	41	AF	D GREEN4	MAGENTA1	193	C1	AN	GREEN12	MAGENTA1
66	42	BF	D GREEN4	D BLUE2	194	C2	BN	GREEN12	D BLUE2
67	43	CF	D GREEN4	PURPLE3	195	C3	CN	GREEN12	PURPLE3
68	44	DF	D GREEN4	D GREEN4	196	C4	DN	GREEN12	D GREEN4
69	45	EF	D GREEN4	GREY5	197	C5	EN	GREEN12	GREY5
70	46	FF	D GREEN4	M BLUE6	198	C6	FN	GREEN12	M BLUE6
71	47	GF	D GREEN4	L BLUE7	199	C7	GN	GREEN12	L BLUE7
72	48	HF	D GREEN4	BROWN8	200	C8	HN	GREEN12	BROWN8
73	49	IF	D GREEN4	ORANGE9	201	C9	IN	GREEN12	ORANGE9
74	4A	JF	D GREEN4	GREY10	202	CA	JN	GREEN12	GREY10
75	4B	KF	D GREEN4	PINK11	203	CB	KN	GREEN12	PINK11
76	4C	LF	D GREEN4	GREEN12	204	CC	LN	GREEN12	GREEN12
77	4D	MF	D GREEN4	YELLOW13	205	CD	MN	GREEN12	YELLOW13
78	4E	NF	D GREEN4	AQUA14	206	CE	NN	GREEN12	AQUA14
79	4F	OF	D GREEN4	WHITE15	207	CF	ON	GREEN12	WHITE15
80	50	PF	GREY5	BLACK0	208	D0	PN	YELLOW13	BLACK0

BOTTOM LORES					TOP LORES				
DEC	HEX	TXT	BLOCK		DEC	HEX	TXT	BLOCK	
81	51	QF	GREY5	MAGENTA1	209	D1	QN	YELLOW13	MAGENTA1
82	52	RF	GREY5	D BLUE2	210	D2	RN	YELLOW13	D BLUE2
83	53	SF	GREY5	PURPLE3	211	D3	SN	YELLOW13	PURPLE3
84	54	TF	GREY5	D GREEN4	212	D4	TN	YELLOW13	D GREEN4
85	55	UF	GREY5	GREY5	213	D5	UN	YELLOW13	GREY5
86	56	VF	GREY5	M BLUE6	214	D6	VN	YELLOW13	M BLUE6
87	57	WF	GREY5	L BLUE7	215	D7	WN	YELLOW13	L BLUE7
88	58	XF	GREY5	BROWN8	216	D8	XN	YELLOW13	BROWN8
89	59	YF	GREY5	ORANGE9	217	D9	YN	YELLOW13	ORANGE9
90	5A	ZF	GREY5	GREY10	218	DA	ZN	YELLOW13	GREY10
91	5B	[F	GREY5	PINK11	219	DB	IN	YELLOW13	PINK11
92	5C	\F	GREY5	GREEN12	220	DC	\N	YELLOW13	GREEN12
93	5D	]F	GREY5	YELLOW13	221	DD	]N	YELLOW13	YELLOW13
94	5E	`F	GREY5	AQUA14	222	DE	`N	YELLOW13	AQUA14
95	5F	_F	GREY5	WHITE15	223	DF	_N	YELLOW13	WHITE15
96	60	F	M BLUE6	BLACK0	224	E0	N	AQUA14	BLACK0
97	61	!F	M BLUE6	MAGENTA1	225	E1	!N	AQUA14	MAGENTA1
98	62	"F	M BLUE6	D BLUE2	226	E2	"N	AQUA14	D BLUE2
99	63	#F	M BLUE6	PURPLE3	227	E3	#N	AQUA14	PURPLE3
100	64	\$F	M BLUE6	D GREEN4	228	E4	\$N	AQUA14	D GREEN4
101	65	%F	M BLUE6	GREY5	229	E5	%N	AQUA14	GREY5
102	66	&F	M BLUE6	M BLUE6	230	E6	&N	AQUA14	M BLUE6
103	67	'F	M BLUE6	L BLUE7	231	E7	'N	AQUA14	L BLUE7
104	68	(F	M BLUE6	BROWN8	232	E8	(N	AQUA14	BROWN8
105	69	)F	M BLUE6	ORANGE9	233	E9	)N	AQUA14	ORANGE9
106	6A	*F	M BLUE6	GREY10	234	EA	*N	AQUA14	GREY10
107	6B	+F	M BLUE6	PINK11	235	EB	+N	AQUA14	PINK11
108	6C	,F	M BLUE6	GREEN12	236	EC	,N	AQUA14	GREEN12
109	6D	-F	M BLUE6	YELLOW13	237	ED	-N	AQUA14	YELLOW13
110	6E	.F	M BLUE6	AQUA14	238	EE	.N	AQUA14	AQUA14
111	6F	/F	M BLUE6	WHITE15	239	EF	/N	AQUA14	WHITE15
112	70	0F	L BLUE7	BLACK0	240	F0	0N	WHITE15	BLACK0
113	71	1F	L BLUE7	MAGENTA1	241	F1	1N	WHITE15	MAGENTA1
114	72	2F	L BLUE7	D BLUE2	242	F2	2N	WHITE15	D BLUE2
115	73	3F	L BLUE7	PURPLE3	243	F3	3N	WHITE15	PURPLE3
116	74	4F	L BLUE7	D GREEN4	244	F4	4N	WHITE15	D GREEN4
117	75	5F	L BLUE7	GREY5	245	F5	5N	WHITE15	GREY5
118	76	6F	L BLUE7	M BLUE6	246	F6	6N	WHITE15	M BLUE6
119	77	7F	L BLUE7	L BLUE7	247	F7	7N	WHITE15	L BLUE7
120	78	8F	L BLUE7	BROWN8	248	F8	8N	WHITE15	BROWN8
121	79	9F	L BLUE7	ORANGE9	249	F9	9N	WHITE15	ORANGE9
122	7A	:F	L BLUE7	GREY10	250	FA	:N	WHITE15	GREY10
123	7B	;F	L BLUE7	PINK11	251	FB	;N	WHITE15	PINK11
124	7C	<F	L BLUE7	GREEN12	252	FC	<N	WHITE15	GREEN12
125	7D	=F	L BLUE7	YELLOW13	253	FD	=N	WHITE15	YELLOW13
126	7E	>F	L BLUE7	AQUA14	254	FE	>N	WHITE15	AQUA14
127	7F	?F	L BLUE7	WHITE15	255	FF	?N	WHITE15	WHITE15

BOTTOM LORES				TOP LORES BLOCK
DEC	HEX	TXT	BLOCK	
224	E0	'N	AQUA14	BLACK0
225	E1	aN	AQUA14	MAGENTA1
226	E2	bN	AQUA14	D BLUE2
227	E3	cN	AQUA14	PURPLE3
228	E4	dN	AQUA14	D GREEN4
229	E5	eN	AQUA14	GREY5
230	E6	fN	AQUA14	M BLUE6
231	E7	gN	AQUA14	L BLUE7
232	E8	hN	AQUA14	BROWN8
233	E9	iN	AQUA14	ORANGE9
234	EA	jN	AQUA14	GREY10
235	EB	kN	AQUA14	PINK11
236	EC	lN	AQUA14	GREEN12
237	ED	mN	AQUA14	YELLOW13
238	EE	nN	AQUA14	AQUA14
239	EF	oN	AQUA14	WHITE15
240	F0	pN	WHITE15	BLACK0
241	F1	qN	WHITE15	MAGENTA1
242	F2	rN	WHITE15	D BLUE2
243	F3	sN	WHITE15	PURPLE3
244	F4	tN	WHITE15	D GREEN4
245	F5	uN	WHITE15	GREY5
246	F6	vN	WHITE15	M BLUE6
247	F7	wN	WHITE15	L BLUE7
248	F8	xN	WHITE15	BROWN8
249	F9	yN	WHITE15	ORANGE9
250	FA	zN	WHITE15	GREY10
251	FB	{N	WHITE15	PINK11
252	FC	lN	WHITE15	GREEN12
253	FD	}N	WHITE15	YELLOW13
254	FE	^N	WHITE15	AQUA14
255	FF	ON	WHITE15	WHITE15

For Apple //e's and Apple II's with
lower case adaptors

TEXT/LOW RESOLUTION SCREEN MEMORY LINE ADDRESS CHART

This chart is useful for locating screen addresses for either text characters or lores blocks in Page 1 and Page 2 of screen memory. The leftmost column for both Page 1 and 2 provides text screen line number (VTAB or vertical position) in decimal. The decimal and hex address values are given as the first two columns of each division. By POKEing in values at the decimal location given in that row, it is possible to easily place text or lores plots. In lores mixed mode, the bottom four rows are shown in text, and the rows above in lores graphics, and when in hires mixed mode only those bottom four lines are used for text display. (See the hires graphics line address chart for hires lines.)

For example, the following program will put a lores block in the middle of the screen:

```
Line #12 = 1,448. Add 20 to get to center of line.
10 GR : POKE 1448 + 20, 19 : END
RUN
```

The following abbreviations are used:

TX SC BA LN = Text screen basic line number.

TX WIN SET = Text window setting.

LO (HI) NI LR LN = Low (high) nibble of lores line number.

PAGE 1 TEXT/LORES						PAGE 2 TEXT/LORES					
TX TX	1ST BYTES OF	LAST BYTE OF	1ST BYTE OF	LAST BYTE OF	LO HI	TX WIN	LINE ADDR	LN ADDR	1ST BYTE OF	LAST BYTE OF	NI NI
BA SET	PA 1 TX LR	PG 1 TX LR	PG 2 TX LR	PG 2 TX LR	LR LR	LN					LN LN
	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	
1	0	1,024	400	1,063	427	2,048	800	2,087	827	1	2
2	1	1,152	480	1,191	4A7	2,176	880	2,215	8A7	3	4
3	2	1,280	500	1,319	527	2,304	900	2,343	927	5	6
4	3	1,408	580	1,447	5A7	2,432	980	2,471	9A7	7	8
5	4	1,536	600	1,575	627	2,560	A00	2,599	A27	9	10
6	5	1,664	680	1,703	6A7	2,688	A80	2,727	AA7	11	12
7	6	1,792	700	1,831	727	2,816	B00	2,855	B27	13	14
8	7	1,920	780	1,959	7A7	2,944	B80	2,983	BA7	15	16
9	8	1,064	428	1,103	44F	2,088	828	2,127	84F	17	18
10	9	1,192	4A8	1,231	4CF	2,216	8A8	2,255	8CF	19	20
11	10	1,320	528	1,359	54F	2,344	928	2,383	94F	21	22
12	11	1,448	5A8	1,487	5CF	2,472	9A8	2,511	9CF	23	24
13	12	1,576	628	1,615	64F	2,600	A28	2,639	A4F	25	26
14	13	1,704	6A8	1,743	6CF	2,728	AA8	2,767	ACF	27	28
15	14	1,832	728	1,871	74F	2,856	B28	2,895	B4F	29	30
16	15	1,960	7A8	1,999	7CF	2,984	BA8	3,023	BCF	31	32
17	16	1,104	450	1,143	477	2,128	850	2,167	877	33	34
18	17	1,232	4D0	1,271	4F7	2,256	8D0	2,295	8F7	35	36
19	18	1,360	550	1,399	577	2,384	950	2,423	977	37	38
20	19	1,488	5D0	1,527	5F7	2,512	9D0	2,551	9F7	39	40
21	20	1,616	650	1,655	677	2,640	A50	2,679	A77	41	42
22	21	1,744	6D0	1,783	6F7	2,768	AD0	2,807	AF7	43	44
23	22	1,872	750	1,911	777	2,896	B50	2,935	B77	45	46
24	23	2,000	7D0	2,039	7F7	3,024	BD0	3,063	BF7	47	48

TEXT AND LOW RESOLUTION TRI-LINE CHARTS

Tri-lines are made up of 128 bytes arranged in three 40 byte groups with one 8 byte gap at the end.

: _____ 40 _____ : _____ 40 _____ : _____ 40 _____ : 8 _____ :

Using these charts it is possible to look up sequential and non-sequential line output on the text and lores screen. They are very useful addresses for fast screen alterations in assembly level programming. From BASIC, text and lores color can easily be POKEd in using the tri-lines where sequential location is important. For example, the following program draws tri-line number 4 in lores graphics:

```
10 GR : FOR I = 1408 TO 1527 : POKE I, 19 : NEXT
```

As can be seen when the program is run, the bars are spaced apart on the screen. However, the following program, using the first line *group*, will put the lines together in a block.

```
10 GR : FOR I = 1280 TO 1319 : POKE I, 19 : NEXT
20 FOR I = 1408 TO 1447 : POKE I, 19 : NEXT
30 FOR I = 1536 TO 1575 : POKE I, 19 : NEXT
```

Text output can also be formatted with tri-lines, the main focus being on the sequential or non-sequential locations in terms of *groups*, not addresses. In the mixed lores mode, lines 5 through 8 in the third line group are displayed in text. Thus, using the addresses in those lines and groups, placing text in mixed lores graphics can be greatly enhanced.

PAGE 1 TEXT/LORES TRI-LINE CHART

TRI LN NO.	START 1 FIRST LN GROUP	START 2 2ND LINE GROUP	START 3 3RD LINE GROUP	GAP I/O BYTES	END OF TRI-LN
	DEC HEX	DEC HEX	DEC HEX	DEC HEX	DEC HEX
1	1,024 400	1,064 428	1,104 450	1,144 478	1,151 47F
2	1,152 480	1,192 4A8	1,232 4D0	1,272 4F8	1,279 4FF
3	1,280 500	1,320 528	1,360 550	1,400 578	1,407 57F
4	1,408 580	1,448 5A8	1,488 5D0	1,528 5F8	1,535 5FF
5	1,536 600	1,576 628	1,616 650	1,656 678	1,663 67F
6	1,664 680	1,704 6A8	1,744 6D0	1,784 6F8	1,791 6FF
7	1,792 700	1,832 728	1,872 750	1,912 778	1,919 77F
8	1,920 780	1,960 7A8	2,000 7D0	2,040 7F8	2,047 7FF

PAGE 2 TEXT/LORES TRI-LINE CHART

TRI LN NO.	START 1 FIRST LN GROUP	START 2 2ND LINE GROUP	START 3 3RD LINE GROUP	GAP I/O BYTES	END OF TRI-LN
	DEC HEX	DEC HEX	DEC HEX	DEC HEX	DEC HEX
1	2,048 800	2,088 828	2,128 850	2,168 878	2,175 87F
2	2,176 880	2,216 8A8	2,256 8D0	2,296 8F8	2,303 8FF
3	2,304 900	2,344 928	2,384 950	2,424 978	2,431 97F
4	2,432 980	2,472 9A8	2,512 9D0	2,552 9F8	2,559 9FF
5	2,560 A00	2,600 A28	2,640 A50	2,680 A78	2,687 A7F
6	2,688 A80	2,728 AA8	2,768 AD0	2,808 AF8	2,815 AFF
7	2,816 B00	2,856 B28	2,896 B50	2,936 B78	2,943 B7F
8	2,944 B80	2,984 BA8	3,024 BD0	3,064 BF8	3,071 BFF

TRI-LINE MODE DISPLAY FOR PAGE 1 AND PAGE 2

TRI LN NO.	TEXT MODE DISPLAY SCREEN LINES			LORES MODE DISPLAY SCREEN LINES		
	#1	#2	#3	#1	#2	#3
1	1	9	17	1 & 2	17 & 18	33 & 34
2	2	10	18	3 & 4	19 & 20	35 & 36
3	3	11	19	5 & 6	21 & 22	37 & 38
4	4	12	20	7 & 8	23 & 24	39 & 40
5	5	13	21	9 & 10	25 & 26	41 & 42
6	6	14	22	11 & 12	27 & 28	43 & 44
7	7	15	23	13 & 14	29 & 30	45 & 46
8	8	16	24	15 & 16	31 & 32	47 & 48

MIXING GRAPHICS AND TEXT



HIGH RESOLUTION HORIZONTAL PIXEL CHART

This chart can be used for accessing individual pixels in high resolution graphics. Individual pixels can be turned on and off at any location on the screen. Both the "traditional" and the "modern" modes of plotting pixels are represented on this chart. One side of the chart is set up for the "traditional," and other side for "modern." The traditional mode has 280 plot positions horizontally made up of 40 clusters of seven bits (pixels). The first two columns can be used for this form of plotting. There are two plotting modes for the modern view of hires graphics. One mode is for black and white plotting with 560 horizontal points (pixels) per line, and the other is for color with 140 points. Each color point represents four consecutive black and white points. Depending upon which approach is taken in programming graphics, use the appropriate columns.

The following abbreviations are employed:

DBO = Decimal Byte Offset

HBO = Hexadecimal Byte Offset

HPLT X/C = HPLT X-coordinate S/P = Screen Pixel

SCREEN PIX COL = Screen Pixel Color

BIT 0-6 & HBIT ST = Bit 0-6 to set and high bit status

SC CLM PX LIT = Screen Column's Pixel Lit

MODERN X-CO = Modern Hires Color Mode X-coordinate value.

TRADITIONAL			MODERN/BLACK AND WHITE			MODERN		
DBO	HBO	HPLT X/C	S/P	PIX COL	BIT 0-6 & HBIT ST	SC	CLM PX LIT	X-CO
0	0	0	0	VIOLET	0000,0001	+0112233445566	0	
			1	BLUE	1000,0001	0+112233445566		
		1	2	GREEN	0000,0010	00+12233445566		
			3	ORANGE	1000,0010	001+2233445566		
		2	4	VIOLET	0000,0100	0011+233445566	1	
			5	BLUE	1000,0100	00112+33445566		
		3	6	GREEN	0000,1000	001122+3445566		
			7	ORANGE	1000,1000	0011223+445566		
		4	8	VIOLET	0001,0000	00112233+45566	2	
			9	BLUE	1001,0000	001122334+5566		
		5	10	GREEN	0010,0000	0011223344+566		
			11	ORANGE	1010,0000	00112233445+66		
		6	12	VIOLET	0100,0000	001122334455+6	3	
			13	BLUE	1100,0000	0011223344556+		
1	1	7	14	GREEN	0000,0001	+0112233445566		
			15	ORANGE	1000,0001	0+112233445566		
		8	16	VIOLET	0000,0010	00+12233445566	4	
			17	BLUE	1000,0010	001+2233445566		
		9	18	GREEN	0000,0100	0011+233445566		
			19	ORANGE	1000,0100	00112+33445566		

TRADITIONAL			MODERN/BLACK AND WHITE			MODERN		
HPLT			SCREEN BIT 0-6					
DBO	HBO	X/C	S/P	PIX	COL & HBIT	ST	SC	CLM PX LIT X-CO
			10	20	VIOLET	0000,1000	001122+3445566	5
				21	BLUE	1000,1000	0011223+445566	
			11	22	GREEN	0001,0000	00112233+45566	
				23	ORANGE	1001,0000	001122334+5566	
			12	24	VIOLET	0010,0000	0011223344+566	6
				25	BLUE	1010,0000	00112233445+66	
			13	26	GREEN	0100,0000	001122334455+6	
				27	ORANGE	1100,0000	0011223344556+	
2	2		14	28	VIOLET	0000,0001	+0112233445566	7
				29	BLUE	1000,0001	0+112233445566	
			15	30	GREEN	0000,0010	00+12233445566	
				31	ORANGE	1000,0010	001+2233445566	
			16	32	VIOLET	0000,0100	0011+233445566	8
				33	BLUE	1000,0100	00112+33445566	
			17	34	GREEN	0000,1000	001122+3445566	
				35	ORANGE	1000,1000	0011223+445566	
			18	36	VIOLET	0001,0000	00112233+45566	9
				37	BLUE	1001,0000	001122334+5566	
			19	38	GREEN	0010,0000	0011223344+566	
				39	ORANGE	1010,0000	00112233445+66	
			20	40	VIOLET	0100,0000	001122334455+6	10
				41	BLUE	1100,0000	0011223344556+	
3	3		21	42	GREEN	0000,0001	+0112233445566	
				43	ORANGE	1000,0001	0+112233445566	
			22	44	VIOLET	0000,0010	00+12233445566	11
				45	BLUE	1000,0010	001+2233445566	
			23	46	GREEN	0000,0100	0011+233445566	
				47	ORANGE	1000,0100	00112+33445566	
			24	48	VIOLET	0000,1000	001122+3445566	12
				49	BLUE	1000,1000	0011223+445566	
			25	50	GREEN	0001,0000	00112233+45566	
				51	ORANGE	1001,0000	001122334+5566	
			26	52	VIOLET	0010,0000	0011223344+566	13
				53	BLUE	1010,0000	00112233445+66	
			27	54	GREEN	0100,0000	001122334455+6	
				55	ORANGE	1100,0000	0011223344556+	
4	4		28	56	VIOLET	0000,0001	+0112233445566	14
				57	BLUE	1000,0001	0+112233445566	
			29	58	GREEN	0000,0010	00+12233445566	
				59	ORANGE	1000,0010	001+2233445566	
			30	60	VIOLET	0000,0100	0011+233445566	15
				61	BLUE	1000,0100	00112+33445566	
			31	62	GREEN	0000,1000	001122+3445566	
				63	ORANGE	1000,1000	0011223+445566	
			32	64	VIOLET	0001,0000	00112233+45566	16
				65	BLUE	1001,0000	001122334+5566	
			33	66	GREEN	0010,0000	0011223344+566	
				67	ORANGE	1010,0000	00112233445+66	
			34	68	VIOLET	0100,0000	001122334455+6	17

TRADITIONAL			MODERN/BLACK AND WHITE				MODERN			
HPLT			SCREEN BIT 0-6							
DBO	HBO	X/C	S/P	PIX	COL	HBIT	ST	SC	CLM	PX LIT X-CO
5	5	35	69	BLUE	1100,0000	0011223344556+				
			70	GREEN	0000,0001	+0112233445566				
			71	ORANGE	1000,0001	0+112233445566				
		36	72	VIOLET	0000,0010	00+12233445566				18
			73	BLUE	1000,0010	001+2233445566				
		37	74	GREEN	0000,0100	0011+233445566				
			75	ORANGE	1000,0100	00112+33445566				
		38	76	VIOLET	0000,1000	001122+3445566				19
			77	BLUE	1000,1000	0011223+445566				
		39	78	GREEN	0001,0000	00112233+45566				
			79	ORANGE	1001,0000	001122334+5566				
		40	80	VIOLET	0010,0000	0011223344+566				20
			81	BLUE	1010,0000	00112233445+66				
		41	82	GREEN	0100,0000	001122334455+6				
			83	ORANGE	1100,0000	0011223344556+				
6	6	42	84	VIOLET	0000,0001	+0112233445566				21
			85	BLUE	1000,0001	0+112233445566				
		43	86	GREEN	0000,0010	00+12233445566				
			87	ORANGE	1000,0010	001+2233445566				
		44	88	VIOLET	0000,0100	0011+233445566				22
			89	BLUE	1000,0100	00112+33445566				
		45	90	GREEN	0000,1000	001122+3445566				
			91	ORANGE	1000,1000	0011223+445566				
		46	92	VIOLET	0001,0000	00112233+45566				23
			93	BLUE	1001,0000	001122334+5566				
		47	94	GREEN	0010,0000	0011223344+566				
			95	ORANGE	1010,0000	00112233445+66				
		48	96	VIOLET	0100,0000	001122334455+6				24
			97	BLUE	1100,0000	0011223344556+				
7	7	49	98	GREEN	0000,0001	+0112233445566				
			99	ORANGE	1000,0001	0+112233445566				
		50	100	VIOLET	0000,0010	00+12233445566				25
			101	BLUE	1000,0010	001+2233445566				
		51	102	GREEN	0000,0100	0011+233445566				
			103	ORANGE	1000,0100	00112+33445566				
		52	104	VIOLET	0000,1000	001122+3445566				26
			105	BLUE	1000,1000	0011223+445566				
		53	106	GREEN	0001,0000	00112233+45566				
			107	ORANGE	1001,0000	001122334+5566				
		54	108	VIOLET	0010,0000	0011223344+566				27
			109	BLUE	1010,0000	00112233445+66				
		55	110	GREEN	0100,0000	001122334455+6				
			111	ORANGE	1100,0000	0011223344556+				
8	8	56	112	VIOLET	0000,0001	+0112233445566				28
			113	BLUE	1000,0001	0+112233445566				
		57	114	GREEN	0000,0010	00+12233445566				
			115	ORANGE	1000,0010	001+2233445566				
		58	116	VIOLET	0000,0100	0011+233445566				29
			117	BLUE	1000,0100	00112+33445566				
		59	118	GREEN	0000,1000	001122+3445566				

TRADITIONAL		MODERN/BLACK AND WHITE		MODERN	
HPLT		SCREEN BIT 0-6		MODERN	
DBO	HBO	X/C	S/P PIX COL & HBIT ST	SC CLM PX LIT	X-CO
			119 ORANGE 1000,1000	0011223+445566	
		60 120	VIOLET 0001,0000	00112233+45566	30
		121	BLUE 1001,0000	001122334+5566	
		61 122	GREEN 0010,0000	0011223344+566	
		123	ORANGE 1010,0000	00112233445+66	
		62 124	VIOLET 0100,0000	001122334455+6	31
		125	BLUE 1100,0000	0011223344556+	
9	9	63 126	GREEN 0000,0001	+0112233445566	
		127	ORANGE 1000,0001	0+112233445566	
		64 128	VIOLET 0000,0010	00+12233445566	32
		129	BLUE 1000,0010	001+2233445566	
		65 130	GREEN 0000,0100	0011+233445566	
		131	ORANGE 1000,0100	00112+33445566	
		66 132	VIOLET 0000,1000	001122+3445566	33
		133	BLUE 1000,1000	0011223+445566	
		67 134	GREEN 0001,0000	00112233+45566	
		135	ORANGE 1001,0000	001122334+5566	
		68 136	VIOLET 0010,0000	0011223344+566	34
		137	BLUE 1010,0000	00112233445+66	
		69 138	GREEN 0100,0000	001122334455+6	
		139	ORANGE 1100,0000	0011223344556+	
10	A	70 140	VIOLET 0000,0001	+0112233445566	35
		141	BLUE 1000,0001	0+112233445566	
		71 142	GREEN 0000,0010	00+12233445566	
		143	ORANGE 1000,0010	001+2233445566	
		72 144	VIOLET 0000,0100	0011+233445566	36
		145	BLUE 1000,0100	00112+33445566	
		73 146	GREEN 0000,1000	001122+3445566	
		147	ORANGE 1000,1000	0011223+445566	
		74 148	VIOLET 0001,0000	00112233+45566	37
		149	BLUE 1001,0000	001122334+5566	
		75 150	GREEN 0010,0000	0011223344+566	
		151	ORANGE 1010,0000	00112233445+66	
		76 152	VIOLET 0100,0000	001122334455+6	38
		153	BLUE 1100,0000	0011223344556+	
11	B	77 154	GREEN 0000,0001	+0112233445566	
		155	ORANGE 1000,0001	0+112233445566	
		78 156	VIOLET 0000,0010	00+12233445566	39
		157	BLUE 1000,0010	001+2233445566	
		79 158	GREEN 0000,0100	0011+233445566	
		159	ORANGE 1000,0100	00112+33445566	
		80 160	VIOLET 0000,1000	001122+3445566	40
		161	BLUE 1000,1000	0011223+445566	
		81 162	GREEN 0001,0000	00112233+45566	
		163	ORANGE 1001,0000	001122334+5566	
		82 164	VIOLET 0010,0000	0011223344+566	41
		165	BLUE 1010,0000	00112233445+66	
		83 166	GREEN 0100,0000	001122334455+6	
		167	ORANGE 1100,0000	0011223344556+	
12	C	84 168	VIOLET 0000,0001	+0112233445566	42

TRADITIONAL		MODERN/BLACK AND WHITE				MODERN		
HPLT		SCREEN BIT 0-6						
DBO	HBO	X/C	S/P	PIX	COL & HBIT	ST	SC	CLM PX LIT X-CO
			169	BLUE	1000,0001	0+112233445566		
	85	170	GREEN	0000,0010	00+12233445566			
		171	ORANGE	1000,0010	001+2233445566			
	86	172	VIOLET	0000,0100	0011+233445566	43		
		173	BLUE	1000,0100	00112+33445566			
	87	174	GREEN	0000,1000	001122+3445566			
		175	ORANGE	1000,1000	0011223+445566			
	88	176	VIOLET	0001,0000	00112233+45566	44		
		177	BLUE	1001,0000	001122334+5566			
	89	178	GREEN	0010,0000	0011223344+566			
		179	ORANGE	1010,0000	00112233445+66			
	90	180	VIOLET	0100,0000	001122334455+6	45		
		181	BLUE	1100,0000	0011223344556+			
13	D	91	182	GREEN	0000,0001	+0112233445566		
		183	ORANGE	1000,0001	0+112233445566			
		92	184	VIOLET	0000,0010	00+12233445566	46	
		185	BLUE	1000,0010	001+2233445566			
	93	186	GREEN	0000,0100	0011+233445566			
		187	ORANGE	1000,0100	00112+33445566			
	94	188	VIOLET	0000,1000	001122+3445566	47		
		189	BLUE	1000,1000	0011223+445566			
	95	190	GREEN	0001,0000	00112233+45566			
		191	ORANGE	1001,0000	001122334+5566			
	96	192	VIOLET	0010,0000	0011223344+566	48		
		193	BLUE	1010,0000	00112233445+66			
	97	194	GREEN	0100,0000	001122334455+6			
		195	ORANGE	1100,0000	0011223344556+			
14	E	98	196	VIOLET	0000,0001	+0112233445566	49	
		197	BLUE	1000,0001	0+112233445566			
	99	198	GREEN	0000,0010	00+12233445566			
		199	ORANGE	1000,0010	001+2233445566			
	100	200	VIOLET	0000,0100	0011+233445566	50		
		201	BLUE	1000,0100	00112+33445566			
	101	202	GREEN	0000,1000	001122+3445566			
		203	ORANGE	1000,1000	0011223+445566			
	102	204	VIOLET	0001,0000	00112233+45566	51		
		205	BLUE	1001,0000	001122334+5566			
	103	206	GREEN	0010,0000	0011223344+566			
		207	ORANGE	1010,0000	00112233445+66			
	104	208	VIOLET	0100,0000	001122334455+6	52		
		209	BLUE	1100,0000	0011223344556+			
15	F	105	210	GREEN	0000,0001	+0112233445566		
		211	ORANGE	1000,0001	0+112233445566			
	106	212	VIOLET	0000,0010	00+12233445566	53		
		213	BLUE	1000,0010	001+2233445566			
	107	214	GREEN	0000,0100	0011+233445566			
		215	ORANGE	1000,0100	00112+33445566			
	108	216	VIOLET	0000,1000	001122+3445566	54		
		217	BLUE	1000,1000	0011223+445566			
	109	218	GREEN	0001,0000	00112233+45566			

TRADITIONAL		MODERN/BLACK AND WHITE		MODERN	
HPLT		SCREEN BIT 0-6			
DBO	HBO	X/C	S/P PIX COL & HBIT ST	SC CLM PX LIT	X-CO
		219	ORANGE 1001,0000	001122334+5566	
	110	220	VIOLET 0010,0000	0011223344+566	55
		221	BLUE 1010,0000	00112233445+66	
	111	222	GREEN 0100,0000	001122334455+6	
		223	ORANGE 1100,0000	0011223344556+	
16	10	112	224	VIOLET 0000,0001	+0112233445566 56
		225	BLUE 1000,0001	0+112233445566	
	113	226	GREEN 0000,0010	00+12233445566	
		227	ORANGE 1000,0010	001+2233445566	
	114	228	VIOLET 0000,0100	0011+233445566	57
		229	BLUE 1000,0100	00112+33445566	
	115	230	GREEN 0000,1000	001122+3445566	
		231	ORANGE 1000,1000	0011223+445566	
	116	232	VIOLET 0001,0000	00112233+45566	58
		233	BLUE 1001,0000	001122334+5566	
	117	234	GREEN 0010,0000	0011223344+566	
		235	ORANGE 1010,0000	00112233445+66	
	118	236	VIOLET 0100,0000	001122334455+6	59
		237	BLUE 1100,0000	0011223344556+	
17	11	119	238	GREEN 0000,0001	+0112233445566
		239	ORANGE 1000,0001	0+112233445566	
	120	240	VIOLET 0000,0010	00+12233445566	60
		241	BLUE 1000,0010	001+2233445566	
	121	242	GREEN 0000,0100	0011+233445566	
		243	ORANGE 1000,0100	00112+33445566	
	122	244	VIOLET 0000,1000	001122+3445566	61
		245	BLUE 1000,1000	0011223+445566	
	123	246	GREEN 0001,0000	00112233+45566	
		247	ORANGE 1001,0000	001122334+5566	
	124	248	VIOLET 0010,0000	0011223344+566	62
		249	BLUE 1010,0000	00112233445+66	
	125	250	GREEN 0100,0000	001122334455+6	
		251	ORANGE 1100,0000	0011223344556+	
18	12	126	252	VIOLET 0000,0001	+0112233445566 63
		253	BLUE 1000,0001	0+112233445566	
	127	254	GREEN 0000,0010	00+12233445566	
		255	ORANGE 1000,0010	001+2233445566	
	128	256	VIOLET 0000,0100	0011+233445566	64
		257	BLUE 1000,0100	00112+33445566	
	129	258	GREEN 0000,1000	001122+3445566	
		259	ORANGE 1000,1000	0011223+445566	
	130	260	VIOLET 0001,0000	00112233+45566	65
		261	BLUE 1001,0000	001122334+5566	
	131	262	GREEN 0010,0000	0011223344+566	
		263	ORANGE 1010,0000	00112233445+66	
	132	264	VIOLET 0100,0000	001122334455+6	66
		265	BLUE 1100,0000	0011223344556+	
19	13	133	266	GREEN 0000,0001	+0112233445566
		267	ORANGE 1000,0001	0+112233445566	
	134	268	VIOLET 0000,0010	00+12233445566	67

TRADITIONAL			MODERN/BLACK AND WHITE					MODERN		
HPLT			SCREEN BIT 0-6							
DBO	HBO	X/C	S/P	PIX	COL	HBIT	ST	SC	CLM	PX LIT X-CO
			269	BLUE	1000	0010	001+2233445566			
135			270	GREEN	0000	0100	0011+233445566			
			271	ORANGE	1000	0100	00112+33445566			
136			272	VIOLET	0000	1000	001122+3445566	68		
			273	BLUE	1000	1000	0011223+445566			
137			274	GREEN	0001	0000	00112233+45566			
			275	ORANGE	1001	0000	001122334+5566			
138			276	VIOLET	0010	0000	0011223344+566	69		
			277	BLUE	1010	0000	00112233445+66			
139			278	GREEN	0100	0000	001122334455+6			
			279	ORANGE	1100	0000	0011223344556+			
20	14	140	280	VIOLET	0000	0001	+0112233445566	70		
			281	BLUE	1000	0001	0+112233445566			
		141	282	GREEN	0000	0010	00+12233445566			
			283	ORANGE	1000	0010	001+2233445566			
		142	284	VIOLET	0000	0100	0011+233445566	71		
			285	BLUE	1000	0100	00112+33445566			
		143	286	GREEN	0000	1000	001122+3445566			
			287	ORANGE	1000	1000	0011223+445566			
		144	288	VIOLET	0001	0000	00112233+45566	72		
			289	BLUE	1001	0000	001122334+5566			
		145	290	GREEN	0010	0000	0011223344+566			
			291	ORANGE	1010	0000	00112233445+66			
		146	292	VIOLET	0100	0000	001122334455+6	73		
			293	BLUE	1100	0000	0011223344556+			
21	15	147	294	GREEN	0000	0001	+0112233445566			
			295	ORANGE	1000	0001	0+112233445566			
		148	296	VIOLET	0000	0010	00+12233445566	74		
			297	BLUE	1000	0010	001+2233445566			
		149	298	GREEN	0000	0100	0011+233445566			
			299	ORANGE	1000	0100	00112+33445566			
		150	300	VIOLET	0000	1000	001122+3445566	75		
			301	BLUE	1000	1000	0011223+445566			
		151	302	GREEN	0001	0000	00112233+45566			
			303	ORANGE	1001	0000	001122334+5566			
		152	304	VIOLET	0010	0000	0011223344+566	76		
			305	BLUE	1010	0000	00112233445+66			
		153	306	GREEN	0100	0000	001122334455+6			
			307	ORANGE	1100	0000	0011223344556+			
22	16	154	308	VIOLET	0000	0001	+0112233445566	77		
			309	BLUE	1000	0001	0+112233445566			
		155	310	GREEN	0000	0010	00+12233445566			
			311	ORANGE	1000	0010	001+2233445566			
		156	312	VIOLET	0000	0100	0011+233445566	78		
			313	BLUE	1000	0100	00112+33445566			
		157	314	GREEN	0000	1000	001122+3445566			
			315	ORANGE	1000	1000	0011223+445566			
		158	316	VIOLET	0001	0000	00112233+45566	79		
			317	BLUE	1001	0000	001122334+5566			
		159	318	GREEN	0010	0000	0011223344+566			

TRADITIONAL		MODERN/BLACK AND WHITE				MODERN	
HPLT		SCREEN BIT 0-6				X-CO	
DBO	HBO	X/C	S/P	PIX	COL & HBIT	ST	SC CLM PX LIT
			319	ORANGE	1010, 0000	00112233445+66	
		160	320	VIOLET	0100, 0000	001122334455+6	80
			321	BLUE	1100, 0000	0011223344556+	
23	17	161	322	GREEN	0000, 0001	+0112233445566	
			323	ORANGE	1000, 0001	0+112233445566	
		162	324	VIOLET	0000, 0010	00+12233445566	81
			325	BLUE	1000, 0010	001+2233445566	
		163	326	GREEN	0000, 0100	0011+233445566	
			327	ORANGE	1000, 0100	00112+33445566	
		164	328	VIOLET	0000, 1000	001122+3445566	82
			329	BLUE	1000, 1000	0011223+445566	
		165	330	GREEN	0001, 0000	00112233+45566	
			331	ORANGE	1001, 0000	001122334+5566	
		166	332	VIOLET	0010, 0000	0011223344+566	83
			333	BLUE	1010, 0000	00112233445+66	
		167	334	GREEN	0100, 0000	001122334455+6	
			335	ORANGE	1100, 0000	0011223344556+	
24	18	168	336	VIOLET	0000, 0001	+0112233445566	84
			337	BLUE	1000, 0001	0+112233445566	
		169	338	GREEN	0000, 0010	00+12233445566	
			339	ORANGE	1000, 0010	001+2233445566	
		170	340	VIOLET	0000, 0100	0011+233445566	85
			341	BLUE	1000, 0100	00112+33445566	
		171	342	GREEN	0000, 1000	001122+3445566	
			343	ORANGE	1000, 1000	0011223+445566	
		172	344	VIOLET	0001, 0000	00112233+45566	86
			345	BLUE	1001, 0000	001122334+5566	
		173	346	GREEN	0010, 0000	0011223344+566	
			347	ORANGE	1010, 0000	00112233445+66	
		174	348	VIOLET	0100, 0000	001122334455+6	87
			349	BLUE	1100, 0000	0011223344556+	
25	19	175	350	GREEN	0000, 0001	+0112233445566	
			351	ORANGE	1000, 0001	0+112233445566	
		176	352	VIOLET	0000, 0010	00+12233445566	88
			353	BLUE	1000, 0010	001+2233445566	
		177	354	GREEN	0000, 0100	0011+233445566	
			355	ORANGE	1000, 0100	00112+33445566	
		178	356	VIOLET	0000, 1000	001122+3445566	89
			357	BLUE	1000, 1000	0011223+445566	
		179	358	GREEN	0001, 0000	00112233+45566	
			359	ORANGE	1001, 0000	001122334+5566	
		180	360	VIOLET	0010, 0000	0011223344+566	90
			361	BLUE	1010, 0000	00112233445+66	
		181	362	GREEN	0100, 0000	001122334455+6	
			363	ORANGE	1100, 0000	0011223344556+	
26	1A	182	364	VIOLET	0000, 0001	+0112233445566	91
			365	BLUE	1000, 0001	0+112233445566	
		183	366	GREEN	0000, 0010	00+12233445566	
			367	ORANGE	1000, 0010	001+2233445566	
		184	368	VIOLET	0000, 0100	0011+233445566	92

TRADITIONAL			MODERN/BLACK AND WHITE				MODERN		
HPLT			SCREEN BIT 0-6						
DBO	HBO	X/C	S/P	PIX	COL & HBIT	ST	SC	CLM	PX LIT X-CO
			369	BLUE	1000,0100	00112+33445566			
	185	370	GREEN	0000,1000	001122+3445566				
		371	ORANGE	1000,1000	0011223+445566				
	186	372	VIOLET	0001,0000	00112233+45566	93			
		373	BLUE	1001,0000	001122334+5566				
	187	374	GREEN	0010,0000	0011223344+566				
		375	ORANGE	1010,0000	00112233445+66				
	188	376	VIOLET	0100,0000	001122334455+6	94			
		377	BLUE	1100,0000	0011223344556+				
27	1B	189	378	GREEN	0000,0001	+0112233445566			
		379	ORANGE	1000,0001	0+112233445566				
	190	380	VIOLET	0000,0010	00+12233445566	95			
		381	BLUE	1000,0010	001+2233445566				
	191	382	GREEN	0000,0100	0011+233445566				
		383	ORANGE	1000,0100	00112+33445566				
	192	384	VIOLET	0000,1000	001122+3445566	96			
		385	BLUE	1000,1000	0011223+445566				
	193	386	GREEN	0001,0000	00112233+45566				
		387	ORANGE	1001,0000	001122334+5566				
	194	388	VIOLET	0010,0000	0011223344+566	97			
		389	BLUE	1010,0000	00112233445+66				
	195	390	GREEN	0100,0000	001122334455+6				
		391	ORANGE	1100,0000	0011223344556+				
28	1C	196	392	VIOLET	0000,0001	+0112233445566	98		
		393	BLUE	1000,0001	0+112233445566				
	197	394	GREEN	0000,0010	00+12233445566				
		395	ORANGE	1000,0010	001+2233445566				
	198	396	VIOLET	0000,0100	0011+233445566	99			
		397	BLUE	1000,0100	00112+33445566				
	199	398	GREEN	0000,1000	001122+3445566				
		399	ORANGE	1000,1000	0011223+445566				
	200	400	VIOLET	0001,0000	00112233+45566	100			
		401	BLUE	1001,0000	001122334+5566				
	201	402	GREEN	0010,0000	0011223344+566				
		403	ORANGE	1010,0000	00112233445+66				
	202	404	VIOLET	0100,0000	001122334455+6	101			
		405	BLUE	1100,0000	0011223344556+				
29	1D	203	406	GREEN	0000,0001	+0112233445566			
		407	ORANGE	1000,0001	0+112233445566				
	204	408	VIOLET	0000,0010	00+12233445566	102			
		409	BLUE	1000,0010	001+2233445566				
	205	410	GREEN	0000,0100	0011+233445566				
		411	ORANGE	1000,0100	00112+33445566				
	206	412	VIOLET	0000,1000	001122+3445566	103			
		413	BLUE	1000,1000	0011223+445566				
	207	414	GREEN	0001,0000	00112233+45566				
		415	ORANGE	1001,0000	001122334+5566				
	208	416	VIOLET	0010,0000	0011223344+566	104			
		417	BLUE	1010,0000	00112233445+66				
	209	418	GREEN	0100,0000	001122334455+6				

TRADITIONAL	MODERN/BLACK AND WHITE
HPLT	SCREEN BIT 0-6
DBO HBO X/C	S/P PIX COL & HBIT ST SC CLM PX LIT X-CO

		419	ORANGE	1100,0000	0011223344556+	
30	1E	210	420	VIOLET	0000,0001	+0112233445566 105
		421	BLUE	1000,0001	0+112233445566	
		211	422	GREEN	0000,0010	00+12233445566
		423	ORANGE	1000,0010	001+2233445566	
		212	424	VIOLET	0000,0100	0011+233445566 106
		425	BLUE	1000,0100	00112+33445566	
		213	426	GREEN	0000,1000	001122+3445566
		427	ORANGE	1000,1000	0011223+445566	
		214	428	VIOLET	0001,0000	00112233+45566 107
		429	BLUE	1001,0000	001122334+5566	
		215	430	GREEN	0010,0000	0011223344+566
		431	ORANGE	1010,0000	00112233445+66	
		216	432	VIOLET	0100,0000	001122334455+6 108
		433	BLUE	1100,0000	0011223344556+	
31	1F	217	434	GREEN	0000,0001	+0112233445566
		435	ORANGE	1000,0001	0+112233445566	
		218	436	VIOLET	0000,0010	00+12233445566 109
		437	BLUE	1000,0010	001+2233445566	
		219	438	GREEN	0000,0100	0011+233445566
		439	ORANGE	1000,0100	00112+33445566	
		220	440	VIOLET	0000,1000	001122+3445566 110
		441	BLUE	1000,1000	0011223+445566	
		221	442	GREEN	0001,0000	00112233+45566
		443	ORANGE	1001,0000	001122334+5566	
		222	444	VIOLET	0010,0000	0011223344+566 111
		445	BLUE	1010,0000	00112233445+66	
		223	446	GREEN	0100,0000	001122334455+6
		447	ORANGE	1100,0000	0011223344556+	
32	20	224	448	VIOLET	0000,0001	+0112233445566 112
		449	BLUE	1000,0001	0+112233445566	
		225	450	GREEN	0000,0010	00+12233445566
		451	ORANGE	1000,0010	001+2233445566	
		226	452	VIOLET	0000,0100	0011+233445566 113
		453	BLUE	1000,0100	00112+33445566	
		227	454	GREEN	0000,1000	001122+3445566
		455	ORANGE	1000,1000	0011223+445566	
		228	456	VIOLET	0001,0000	00112233+45566 114
		457	BLUE	1001,0000	001122334+5566	
		229	458	GREEN	0010,0000	0011223344+566
		459	ORANGE	1010,0000	00112233445+66	
		230	460	VIOLET	0100,0000	001122334455+6 115
		461	BLUE	1100,0000	0011223344556+	
33	21	231	462	GREEN	0000,0001	+0112233445566
		463	ORANGE	1000,0001	0+112233445566	
		232	464	VIOLET	0000,0010	00+12233445566 116
		465	BLUE	1000,0010	001+2233445566	
		233	466	GREEN	0000,0100	0011+233445566
		467	ORANGE	1000,0100	00112+33445566	

TRADITIONAL			MODERN/BLACK AND WHITE				MODERN	
	HPLT		SCREEN BIT 0-6					
DBO	HBO	X/C	S/P	PIX	COL & HBIT	ST	SC	CLM PX LIT X-CO

			234	468	VIOLET	0000,1000	001122+3445566	117
			469	BLUE	1000,1000	0011223+445566		
			235	470	GREEN	0001,0000	00112233+45566	
			471	ORANGE	1001,0000	001122334+5566		
			236	472	VIOLET	0010,0000	0011223344+566	118
			473	BLUE	1010,0000	00112233445+66		
			237	474	GREEN	0100,0000	001122334455+6	
			475	ORANGE	1100,0000	0011223344556+		
34	22		238	476	VIOLET	0000,0001	+0112233445566	119
			477	BLUE	1000,0001	0+112233445566		
			239	478	GREEN	0000,0010	00+12233445566	
			479	ORANGE	1000,0010	001+2233445566		
			240	480	VIOLET	0000,0100	0011+233445566	120
			481	BLUE	1000,0100	00112+33445566		
			241	482	GREEN	0000,1000	001122+3445566	
			483	ORANGE	1000,1000	0011223+445566		
			242	484	VIOLET	0001,0000	00112233+45566	121
			485	BLUE	1001,0000	001122334+5566		
			243	486	GREEN	0010,0000	0011223344+566	
			487	ORANGE	1010,0000	00112233445+66		
			244	488	VIOLET	0100,0000	001122334455+6	122
			489	BLUE	1100,0000	0011223344556+		
35	23		245	490	GREEN	0000,0001	+0112233445566	
			491	ORANGE	1000,0001	0+112233445566		
			246	492	VIOLET	0000,0010	00+12233445566	123
			493	BLUE	1000,0010	001+2233445566		
			247	494	GREEN	0000,0100	0011+233445566	
			495	ORANGE	1000,0100	00112+33445566		
			248	496	VIOLET	0000,1000	001122+3445566	124
			497	BLUE	1000,1000	0011223+445566		
			249	498	GREEN	0001,0000	00112233+45566	
			499	ORANGE	1001,0000	001122334+5566		
			250	500	VIOLET	0010,0000	0011223344+566	125
			501	BLUE	1010,0000	00112233445+66		
			251	502	GREEN	0100,0000	001122334455+6	
			503	ORANGE	1100,0000	0011223344556+		
36	24		252	504	VIOLET	0000,0001	+0112233445566	126
			505	BLUE	1000,0001	0+112233445566		
			253	506	GREEN	0000,0010	00+12233445566	
			507	ORANGE	1000,0010	001+2233445566		
			254	508	VIOLET	0000,0100	0011+233445566	127
			509	BLUE	1000,0100	00112+33445566		
			255	510	GREEN	0000,1000	001122+3445566	
			511	ORANGE	1000,1000	0011223+445566		
			256	512	VIOLET	0001,0000	00112233+45566	128
			513	BLUE	1001,0000	001122334+5566		
			257	514	GREEN	0010,0000	0011223344+566	
			515	ORANGE	1010,0000	00112233445+66		
			258	516	VIOLET	0100,0000	001122334455+6	129
			517	BLUE	1100,0000	0011223344556+		

TRADITIONAL		MODERN/BLACK AND WHITE										MODERN	
HPLT		SCREEN BIT 0-6											
DBO	HBO	X/C	S/P	PIX	COL	&	HBIT	ST	SC	CLM	PX	LIT	X-CO
37	25	259	518	GREEN	0000	,	0001		+	0112233445566			
			519	ORANGE	1000	,	0001		0	+112233445566			
		260	520	VIOLET	0000	,	0010		00	+12233445566		130	
			521	BLUE	1000	,	0010		00	1+2233445566			
		261	522	GREEN	0000	,	0100		00	11+233445566			
			523	ORANGE	1000	,	0100		00	112+33445566			
		262	524	VIOLET	0000	,	1000		00	1122+3445566		131	
			525	BLUE	1000	,	1000		00	11223+445566			
		263	526	GREEN	0001	,	0000		00	112233+45566			
			527	ORANGE	1001	,	0000		00	1122334+5566			
		264	528	VIOLET	0010	,	0000		00	11223344+566		132	
			529	BLUE	1010	,	0000		00	112233445+66			
		265	530	GREEN	0100	,	0000		00	1122334455+6			
			531	ORANGE	1100	,	0000		00	11223344556+			
38	26	266	532	VIOLET	0000	,	0001		+	0112233445566		133	
			533	BLUE	1000	,	0001		0	+112233445566			
		267	534	GREEN	0000	,	0010		00	+12233445566			
			535	ORANGE	1000	,	0010		00	1+2233445566			
		268	536	VIOLET	0000	,	0100		00	11+233445566		134	
			537	BLUE	1000	,	0100		00	112+33445566			
		269	538	GREEN	0000	,	1000		00	1122+3445566			
			539	ORANGE	1000	,	1000		00	11223+445566			
		270	540	VIOLET	0001	,	0000		00	112233+45566		135	
			541	BLUE	1001	,	0000		00	1122334+5566			
		271	542	GREEN	0010	,	0000		00	11223344+566			
			543	ORANGE	1010	,	0000		00	112233445+66			
		272	544	VIOLET	0100	,	0000		00	1122334455+6		136	
			545	BLUE	1100	,	0000		00	11223344556+			
39	27	273	546	GREEN	0000	,	0001		+	0112233445566			
			547	ORANGE	1000	,	0001		0	+112233445566			
		274	548	VIOLET	0000	,	0010		00	+12233445566		137	
			549	BLUE	1000	,	0010		00	1+2233445566			
		275	550	GREEN	0000	,	0100		00	11+233445566			
			551	ORANGE	1000	,	0100		00	112+33445566			
		276	552	VIOLET	0000	,	1000		00	1122+3445566		138	
			553	BLUE	1000	,	1000		00	11223+445566			
		277	554	GREEN	0001	,	0000		00	112233+45566			
			555	ORANGE	1001	,	0000		00	1122334+5566			
		278	556	VIOLET	0010	,	0000		00	11223344+566		139	
			557	BLUE	1010	,	0000		00	112233445+66			
		279	558	GREEN	0100	,	0000		00	1122334455+6			
			559	ORANGE	1100	,	0000		00	11223344556+			

HIGH RESOLUTION MEMORY LINE CHART

This chart provides the RAM addresses for hires memory lines. There are 40 bytes in each hires memory line, and each byte contains one hires character code. Each hires character code displays a pattern of dots to the hires screen. The dot patterns can occupy up to seven dots at a time. Hence 7 dots times 40 bytes gives the 280 positions associated with hires graphics. Actually each byte controls the lit or unlit condition of 14 consecutive pixels. (See the HIRES SCREEN CHARACTER CHART.)

To change any byte (hence any pixel) on the high resolution screen, locate the starting bytes of the desired hires memory line. Now add the relative offset necessary to locate the actual address of your byte. Almost every hires game with fast graphics has much of this chart's addresses in a similar table in memory. This allows them to move hires graphic shapes much faster about the screen.

When in mixed hires graphics within text, hires memory lines 160 through 191 are not shown. Instead, text memory lines 21 through 24 are shown. (See TEXT AND LOW RESOLUTION MEMORY LINE Chart.)

The following program illustrates how this chart can be used. It will clear the hires screen to a background color of your choice. (We used violet.) NOTE: We had a little fun naming our FOR/NEXT loops. Make sure you key them in as printed - LOOP, L00P, LDDP & LQQP.

```
10 EVEN = 85 : REM V-V-V-V : VVV V
20 ODD = 42 : REM -V-V-V- : - VVV
30 HGR2 : REM PAGE 2 HIRES SCREEN
100 FOR LOOP = 16384 TO 16464 STEP 40
110 : FOR L00P = 0 TO 896 STEP 128
120 :: FOR LQQP = 0 TO 7168 STEP 1024
130 ::: LDDP = 0 TO 38 STEP 2
140 :::: POKE LOOP + L00P + LQQP + LDDP, EVEN
150 :::: POKE LOOP + L00P + LQQP + LDDP + 1, ODD
160 ::: NEXT LDDP
170 :: NEXT LQQP
180 : NEXT L00P
190 NEXT LOOP
```

Since the hires graphics screen is composed of even and odd columns, we look up the values for violet in the HIGH RESOLUTION SCREEN CHARACTER CHART. These values are shown in Lines 10 and 20. Line 30, of course, leaves you viewing all of hires graphics page 2. Lines 100 and 190 form the outside loop which steps through the three memory lines of the tri-line. The pattern of hires memory lines can be seen to emerge when the program is RUN. This pattern is reflected by the four FOR/NEXT loops. The two POKEs place the designated values into the odd and even columns.

HGR SCREEN BASIC LN NO.	PAGE 1 HGR ADDRESS OF 1ST BYTE IN LINE	PAGE 1 HGR ADDRESS OF LAST BYTE IN LINE	PAGE 2 HGR ADDRESS OF 1ST BYTE IN LINE	PAGE 2 HGR ADDRESS OF LAST BYTE IN LINE
DEC HEX	DEC HEX	DEC HEC	DEC HEX	DEC HEX
0	\$0	8,192 \$2000	8,231 \$2027	16,384 \$4000
1	1	9,216 2400	9,255 2427	17,408 4400
2	2	10,240 2800	10,279 2827	18,432 4800
3	3	11,264 2C00	11,303 2C27	19,456 4C00
4	4	12,288 3000	12,327 3027	20,480 5000
5	5	13,312 3400	13,351 3427	21,504 5400
6	6	14,336 3800	14,375 3827	22,528 5800
7	7	15,360 3C00	15,399 3C27	23,552 5C00
8	8	8,320 2080	8,359 20A7	16,512 4080
9	9	9,344 2480	9,383 24A7	17,536 4480
10	A	10,368 2880	10,407 28A7	18,560 4880
11	B	11,392 2C80	11,431 2CA7	19,584 4C80
12	C	12,416 3080	12,455 30A7	20,608 5080
13	D	13,440 3480	13,479 34A7	21,632 5480
14	E	14,464 3880	14,503 38A7	22,656 5880
15	F	15,488 3C80	15,527 3CA7	23,680 5C80
16	10	8,448 2100	8,487 2127	16,640 4100
17	11	9,472 2500	9,511 2527	17,664 4500
18	12	10,496 2900	10,535 2927	18,688 4900
19	13	11,520 2D00	11,559 2D27	19,712 4D00
20	14	12,544 3100	12,583 3127	20,736 5100
21	15	13,568 3500	13,607 3527	21,760 5500
22	16	14,592 3900	14,631 3927	22,784 5900
23	17	15,616 3D00	15,655 3D27	23,808 5D00
24	18	8,576 2180	8,615 21A7	16,768 4180
25	19	9,600 2580	9,639 25A7	17,792 4580
26	1A	10,624 2980	10,663 29A7	18,816 4980
27	1B	11,648 2D80	11,687 2DA7	19,840 4D80
28	1C	12,672 3180	12,711 31A7	20,864 5180
29	1D	13,696 3580	13,735 35A7	21,888 5580
30	1E	14,720 3980	14,759 39A7	22,912 5980
31	1F	15,744 3D80	15,783 3DA7	23,936 5D80
32	20	8,704 2200	8,743 2227	16,896 4200
33	21	9,728 2600	9,767 2627	17,920 4600
34	22	10,752 2A00	10,791 2A27	18,944 4A00
35	23	11,776 2E00	11,815 2E27	19,968 4E00
36	24	12,800 3200	12,839 3227	20,992 5200
37	25	13,824 3600	13,863 3627	22,016 5600
38	26	14,848 3A00	14,887 3A27	23,040 5A00
39	27	15,872 3E00	15,911 3E27	24,064 5E00
40	28	8,832 2280	8,871 22A7	17,024 4280
41	29	9,856 2680	9,895 26A7	18,048 4680
42	2A	10,880 2A80	10,919 2AA7	19,072 4A80
43	2B	11,904 2E80	11,943 2EA7	20,096 4E80
44	2C	12,928 3280	12,967 32A7	21,120 5280
45	2D	13,952 3680	13,991 36A7	22,144 5680
46	2E	14,976 3A80	15,015 3AA7	23,168 5A80
47	2F	16,000 3E80	16,039 3EA7	24,192 5E80
48	30	8,960 2300	8,999 2327	17,152 4300

BASIC LN NO.		1ST BYTE IN LINE		LAST BYTE IN LINE		1ST BYTE IN LINE		LAST BYTE IN LINE	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
49	31	9,984	2700	10,023	2727	18,176	4700	18,215	4727
50	32	11,008	2B00	11,047	2B27	19,200	4B00	19,239	4B27
51	33	12,032	2F00	12,071	2F27	20,224	4F00	20,263	4F27
52	34	13,056	3300	13,095	3327	21,248	5300	21,287	5327
53	35	14,080	3700	14,119	3727	22,272	5700	22,311	5727
54	36	15,104	3B00	15,143	3B27	23,296	5B00	23,335	5B27
55	37	16,128	3F00	16,167	3F27	24,320	5F00	24,359	5F27
56	38	9,088	2380	9,127	23A7	17,280	4380	17,319	43A7
57	39	10,112	2780	10,151	27A7	18,304	4780	18,343	47A7
58	3A	11,136	2B80	11,175	2BA7	19,328	4B80	19,367	4BA7
59	3B	12,160	2F80	12,199	2FA7	20,352	4F80	20,391	4FA7
60	3C	13,184	3380	13,223	33A7	21,376	5380	21,415	53A7
61	3D	14,208	3780	14,247	37A7	22,400	5780	22,439	57A7
62	3E	15,232	3B80	15,271	3BA7	23,424	5B80	23,463	5BA7
63	3F	16,256	3F80	16,295	3FA7	24,448	5F80	24,487	5FA7
64	40	8,232	2028	8,271	204F	16,424	4028	16,463	404F
65	41	9,256	2428	9,295	244F	17,448	4428	17,487	444F
66	42	10,280	2828	10,319	284F	18,472	4828	18,511	484F
67	43	11,304	2C28	11,343	2C4F	19,496	4C28	19,535	4C4F
68	44	12,328	3028	12,367	304F	20,520	5028	20,559	504F
69	45	13,352	3428	13,391	344F	21,544	5428	21,583	544F
70	46	14,376	3828	14,415	384F	22,568	5828	22,607	584F
71	47	15,400	3C28	15,439	3C4F	23,592	5C28	23,631	5C4F
72	48	8,360	20A8	8,399	20CF	16,552	40A8	16,591	40CF
73	49	9,384	24A8	9,423	24CF	17,576	44A8	17,615	44CF
74	4A	10,408	28A8	10,447	28CF	18,600	48A8	18,639	48CF
75	4B	11,432	2CA8	11,471	2CCF	19,624	4CA8	19,663	4CCF
76	4C	12,456	30A8	12,495	30CF	20,648	50A8	20,687	50CF
77	4D	13,480	34A8	13,519	34CF	21,672	54A8	21,711	54CF
78	4E	14,504	38A8	14,543	38CF	22,696	58A8	22,735	58CF
79	4F	15,528	3CA8	15,567	3CCF	23,720	5CA8	23,759	5CCF
80	50	8,488	2128	8,527	214F	16,680	4128	16,719	414F
81	51	9,512	2528	9,551	254F	17,704	4528	17,743	454F
82	52	10,536	2928	10,575	294F	18,728	4928	18,767	494F
83	53	11,560	2D28	11,599	2D4F	19,752	4D28	19,791	4D4F
84	54	12,584	3128	12,623	314F	20,776	5128	20,815	514F
85	55	13,608	3528	13,647	354F	21,800	5528	21,839	554F
86	56	14,632	3928	14,671	394F	22,824	5928	22,863	594F
87	57	15,656	3D28	15,695	3D4F	23,848	5D28	23,887	5D4F
88	58	8,616	21A8	8,655	21CF	16,808	41A8	16,847	41CF
89	59	9,640	25A8	9,679	25CF	17,832	45A8	17,871	45CF
90	5A	10,664	29A8	10,703	29CF	18,856	49A8	18,895	49CF
91	5B	11,688	2DA8	11,727	2DCF	19,880	4DA8	19,919	4DCF
92	5C	12,712	31A8	12,751	31CF	20,904	51A8	20,943	51CF
93	5D	13,736	35A8	13,775	35CF	21,928	55A8	21,967	55CF
94	5E	14,760	39A8	14,799	39CF	22,952	59A8	22,991	59CF
95	5F	15,784	3DA8	15,823	3DCF	23,976	5DA8	24,015	5DCF
96	60	8,744	2228	8,783	224F	16,936	4228	16,975	424F
97	61	9,768	2628	9,807	264F	17,960	4628	17,999	464F

BASIC LN NO.		1ST BYTE IN LINE		LAST BYTE IN LINE		1ST BYTE IN LINE		LAST BYTE IN LINE	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
98	62	10,792	2A28	10,831	2A4F	18,984	4A28	19,023	4A4F
99	63	11,816	2E28	11,855	2E4F	20,008	4E28	20,047	4E4F
100	64	12,840	3228	12,879	324F	21,032	5228	21,071	524F
101	65	13,864	3628	13,903	364F	22,056	5628	22,095	564F
102	66	14,888	3A28	14,927	3A4F	23,080	5A28	23,119	5A4F
103	67	15,912	3E28	15,951	3E4F	24,104	5E28	24,143	5E4F
104	68	8,872	22A8	8,911	22CF	17,064	42A8	17,103	42CF
105	69	9,896	26A8	9,935	26CF	18,088	46A8	18,127	46CF
106	6A	10,920	2AA8	10,959	2ACF	19,112	4AA8	19,151	4ACF
107	6B	11,944	2EA8	11,983	2ECF	20,136	4EA8	20,175	4ECF
108	6C	12,968	32A8	13,007	32CF	21,160	52A8	21,199	52CF
109	6D	13,992	36A8	14,031	36CF	22,184	56A8	22,223	56CF
110	6E	15,016	3AA8	15,055	3ACF	23,208	5AA8	23,247	5ACF
111	6F	16,040	3EA8	16,079	3ECF	24,232	5EA8	24,271	5ECF
112	70	9,000	2328	9,039	234F	17,192	4328	17,231	434F
113	71	10,024	2728	10,063	274F	18,216	4728	18,255	474F
114	72	11,048	2B28	11,087	2B4F	19,240	4B28	19,279	4B4F
115	73	12,072	2F28	12,111	2F4F	20,264	4F28	20,303	4F4F
116	74	13,096	3328	13,135	334F	21,288	5328	21,327	534F
117	75	14,120	3728	14,159	374F	22,312	5728	22,351	574F
118	76	15,144	3B28	15,183	3B4F	23,336	5B28	23,375	5B4F
119	77	16,168	3F28	16,207	3F4F	24,360	5F28	24,399	5F4F
120	78	9,128	23A8	9,167	23CF	17,320	43A8	17,359	43CF
121	79	10,152	27A8	10,191	27CF	18,344	47A8	18,383	47CF
122	7A	11,176	2BA8	11,215	2BCF	19,368	4BA8	19,407	4BCF
123	7B	12,200	2FA8	12,239	2FCF	20,392	4FA8	20,431	4FCF
124	7C	13,224	33A8	13,263	33CF	21,416	53A8	21,455	53CF
125	7D	14,248	37A8	14,287	37CF	22,440	57A8	22,479	57CF
126	7E	15,272	3BA8	15,311	3BCF	23,464	5BA8	23,503	5BCF
127	7F	16,296	3FA8	16,335	3FCF	24,488	5FA8	24,527	5FCF
128	80	8,272	2050	8,311	2077	16,464	4050	16,503	4077
129	81	9,296	2450	9,335	2477	17,488	4450	17,527	4477
130	82	10,320	2850	10,359	2877	18,512	4850	18,551	4877
131	83	11,344	2C50	11,383	2C77	19,536	4C50	19,575	4C77
132	84	12,368	3050	12,407	3077	20,560	5050	20,599	5077
133	85	13,392	3450	13,431	3477	21,584	5450	21,623	5477
134	86	14,416	3850	14,455	3877	22,608	5850	22,647	5877
135	87	15,440	3C50	15,479	3C77	23,632	5C50	23,671	5C77
136	88	8,400	20D0	8,439	20F7	16,592	40D0	16,631	40F7
137	89	9,424	24D0	9,463	24F7	17,616	44D0	17,655	44F7
138	8A	10,448	28D0	10,487	28F7	18,640	48D0	18,679	48F7
139	8B	11,472	2CD0	11,511	2CF7	19,664	4CD0	19,703	4CF7
140	8C	12,496	30D0	12,535	30F7	20,688	50D0	20,727	50F7
141	8D	13,520	34D0	13,559	34F7	21,712	54D0	21,751	54F7
142	8E	14,544	38D0	14,583	38F7	22,736	58D0	22,775	58F7
143	8F	15,568	3CD0	15,607	3CF7	23,760	5CD0	23,799	5CF7
144	90	8,528	2150	8,567	2177	16,720	4150	16,759	4177
145	91	9,552	2550	9,591	2577	17,744	4550	17,783	4577
146	92	10,576	2950	10,615	2977	18,768	4950	18,807	4977
147	93	11,600	2D50	11,639	2D77	19,792	4D50	19,831	4D77

BASIC LN NO.		1ST BYTE IN LINE		LAST BYTE IN LINE		1ST BYTE IN LINE		LAST BYTE IN LINE	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
148	94	12,624	3150	12,663	3177	20,816	5150	20,855	5177
149	95	13,648	3550	13,687	3577	21,840	5550	21,879	5577
150	96	14,672	3950	14,711	3977	22,864	5950	22,903	5977
151	97	15,696	3D50	15,735	3D77	23,888	5D50	23,927	5D77
152	98	8,656	21D0	8,695	21F7	16,848	41D0	16,887	41F7
153	99	9,680	25D0	9,719	25F7	17,872	45D0	17,911	45F7
154	9A	10,704	29D0	10,743	29F7	18,896	49D0	18,935	49F7
155	9B	11,728	2DD0	11,767	2DF7	19,920	4DD0	19,959	4DF7
156	9C	12,752	31D0	12,791	31F7	20,944	51D0	20,983	51F7
157	9D	13,776	35D0	13,815	35F7	21,968	55D0	22,007	55F7
158	9E	14,800	39D0	14,839	39F7	22,992	59D0	23,031	59F7
159	9F	15,824	3DD0	15,863	3DF7	24,016	5DD0	24,055	5DF7
160	A0	8,784	2250	8,823	2277	16,976	4250	17,015	4277
161	A1	9,808	2650	9,847	2677	18,000	4650	18,039	4677
162	A2	10,832	2A50	10,871	2A77	19,024	4A50	19,063	4A77
163	A3	11,856	2E50	11,895	2E77	20,048	4E50	20,087	4E77
164	A4	12,880	3250	12,919	3277	21,072	5250	21,111	5277
165	A5	13,904	3650	13,943	3677	22,096	5650	22,135	5677
166	A6	14,928	3A50	14,967	3A77	23,120	5A50	23,159	5A77
167	A7	15,952	3E50	15,991	3E77	24,144	5E50	24,183	5E77
168	A8	8,912	22D0	8,951	22F7	17,168	42D0	17,143	42F7
169	A9	9,936	26D0	9,975	26F7	18,128	46D0	18,167	46F7
170	AA	10,960	2AD0	10,999	2AF7	19,152	4AD0	19,191	4AF7
171	AB	11,984	2ED0	12,023	2EF7	20,176	4ED0	20,215	4EF7
172	AC	13,008	32D0	13,047	32F7	21,200	52D0	21,239	52F7
173	AD	14,032	36D0	14,071	36F7	22,224	56D0	22,263	56F7
174	AE	15,056	3AD0	15,095	3AF7	23,248	5AD0	23,287	5AF7
175	AF	16,080	3ED0	16,119	3EF7	24,272	5ED0	24,311	5EF7
176	B0	9,040	2350	9,079	2377	17,232	4350	17,271	4377
177	B1	10,064	2750	10,103	2777	18,256	4750	18,295	4777
178	B2	11,088	2B50	11,127	2B77	19,280	4B50	19,319	4B77
179	B3	12,112	2F50	12,151	2F77	20,304	4F50	20,343	4F77
180	B4	13,136	3350	13,175	3377	21,328	5350	21,367	5377
181	B5	14,160	3750	14,199	3777	22,352	5750	22,391	5777
182	B6	15,184	3B50	15,223	3B77	23,376	5B50	23,415	5B77
183	B7	16,208	3F50	16,247	3F77	24,400	5F50	24,439	5F77
184	B8	9,168	23D0	9,207	23F7	17,360	43D0	17,399	43F7
185	B9	10,192	27D0	10,231	27F7	18,384	47D0	18,423	47F7
186	BA	11,216	2BD0	11,255	2BF7	19,408	4BD0	19,447	4BF7
187	BB	12,240	2FD0	12,279	2FF7	20,432	4FD0	20,471	4FF7
188	BC	13,264	33D0	13,303	33F7	21,456	53D0	21,495	53F7
189	BD	14,288	37D0	14,327	37F7	22,480	57D0	22,519	57F7
190	BE	15,312	3BD0	15,351	3BF7	23,504	5BD0	23,543	5BF7
191	BF	16,336	3FD0	16,375	3FF7	24,528	5FD0	24,567	5FF7

HIGH RESOLUTION TRI-LINE CHARTS

The addresses on the following charts give the addresses for the tri-lines on the high resolution screens. Each tri-line is composed of three hires memory lines (see TEXT AND LOW RESOLUTION TRI-LINE CHARTS). The three lines of each tri-line make up the top third, middle third and bottom third of the high resolution display screen. The following program demonstrates this phenomenon:

```

10 TEXT : HOME : HGR2
100 FOR L1 = 16384 TO 24448 STEP 128
110 EVEN = 213
120 ODD = 170
130 FOR L2 = 0 TO 38 STEP 2
140 POKE L2 + L1,EVEN
150 POKE L2 + L1 + 1,ODD
160 NEXT L2
200 EVEN = 42
210 ODD = 85
220 FOR L2 = 40 TO 78 STEP 2
230 POKE L1 + L2,EVEN
240 POKE L1 + L2,ODD
250 NEXT L2
300 EVEN = 170
310 ODD = 213
320 FOR L2 = 80 TO 118 STEP 2
330 POKE L1 + L2,EVEN
340 POKE L1 + L2,ODD
350 NEXT L2
360 NEXT L1

```

HIRES TRI-LINE CHART PAGE 1

TRI LN NO.	START 1 FIRST LINE GROUP		START 2 SECOND LINE GROUP		START 3 THIRD LINE GROUP		GAP THESE BYTES NOT DISPLAYED		END OF TRI-LINE	
	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
1	8,192	2000	8,232	2028	8,272	2050	8,312	2078	8,319	207F
2	8,320	2080	8,360	20A8	8,400	20D0	8,440	20F8	8,447	20FF
3	8,448	2100	8,488	2128	8,528	2150	8,568	2178	8,575	217F
4	8,576	2180	8,616	21A8	8,656	21D0	8,696	21F8	8,703	21FF
5	8,704	2200	8,744	2228	8,784	2250	8,824	2278	8,831	227F
6	8,832	2280	8,872	22A8	8,912	22D0	8,952	22F8	8,959	22FF
7	8,960	2300	9,000	2328	9,040	2350	9,080	2378	9,087	237F
8	9,088	2380	9,128	23A8	9,168	23D0	9,208	23F8	9,215	23FF
9	9,216	2400	9,256	2428	9,296	2450	9,336	2478	9,343	247F
10	9,344	2480	9,384	24A8	9,424	24D0	9,464	24F8	9,471	24FF
11	9,472	2500	9,512	2528	9,552	2550	9,592	2578	9,599	257F
12	9,600	2580	9,640	25A8	9,680	25D0	9,720	25F8	9,727	25FF

LN NO.	FIRST LINE GROUP		SECOND LINE GROUP		THIRD LINE GROUP		THESE BYTES NOT DISPLAYED		END OF TRI-LINE	
	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
13	9,728	2600	9,768	2628	9,808	2650	9,848	2678	9,855	267F
14	9,856	2680	9,896	26A8	9,936	26D0	9,976	26F8	9,983	26FF
15	9,984	2700	10,024	2728	10,064	2750	10,104	2778	10,111	277F
16	10,112	2780	10,152	27A8	10,192	27D0	10,232	27F8	10,239	27FF
17	10,240	2800	10,280	2828	10,320	2850	10,360	2878	10,367	287F
18	10,368	2880	10,408	28A8	10,448	28D0	10,488	28F8	10,495	28FF
19	10,496	2900	10,536	2928	10,576	2950	10,616	2978	10,623	297F
20	10,624	2980	10,664	29A8	10,704	29D0	10,744	29F8	10,751	29FF
21	10,752	2A00	10,792	2A28	10,832	2A50	10,872	2A78	10,879	2A7F
22	10,880	2A80	10,920	2AA8	10,960	2AD0	11,000	2AF8	11,007	2AFF
23	11,008	2B00	11,048	2B28	11,088	2B50	11,128	2B78	11,135	2B7F
24	11,136	2B80	11,176	2BA8	11,216	2BD0	11,256	2BF8	11,263	2BFF
25	11,264	2C00	11,304	2C28	11,344	2C50	11,384	2C78	11,391	2C7F
26	11,392	2C80	11,432	2CA8	11,472	2CD0	11,512	2CF8	11,519	2CFF
27	11,520	2D00	11,560	2D28	11,600	2D50	11,640	2D78	11,647	2D7F
28	11,648	2D80	11,688	2DA8	11,728	2DD0	11,768	2DF8	11,775	2DFF
29	11,776	2E00	11,816	2E28	11,856	2E50	11,896	2E78	11,903	2E7F
30	11,904	2E80	11,944	2EA8	11,984	2ED0	12,024	2EF8	12,031	2EFF
31	12,032	2F00	12,072	2F28	12,112	2F50	12,152	2F78	12,159	2F7F
32	12,160	2F80	12,200	2FA8	12,240	2FD0	12,280	2FF8	12,287	2FFF
33	12,288	3000	12,328	3028	12,368	3050	12,408	3078	12,415	307F
34	12,416	3080	12,456	30A8	12,496	30D0	12,536	30F8	12,543	30FF
35	12,544	3100	12,584	3128	12,624	3150	12,664	3178	12,671	317F
36	12,672	3180	12,712	31A8	12,752	31D0	12,792	31F8	12,799	31FF
37	12,800	3200	12,840	3228	12,880	3250	12,920	3278	12,927	327F
38	12,928	3280	12,968	32A8	13,008	32D0	13,048	32F8	13,055	32FF
39	13,056	3300	13,096	3328	13,136	3350	13,176	3378	13,183	337F
40	13,184	3380	13,224	33A8	13,264	33D0	13,304	33F8	13,311	33FF
41	13,312	3400	13,352	3428	13,392	3450	13,432	3478	13,439	347F
42	13,440	3480	13,480	34A8	13,520	34D0	13,560	34F8	13,567	34FF
43	13,568	3500	13,608	3528	13,648	3550	13,688	3578	13,695	357F
44	13,696	3580	13,736	35A8	13,776	35D0	13,816	35F8	13,823	35FF
45	13,824	3600	13,864	3628	13,904	3650	13,944	3678	13,951	367F
46	13,952	3680	13,992	36A8	14,032	36D0	14,072	36F8	14,079	36FF
47	14,080	3700	14,120	3728	14,160	3750	14,200	3778	14,207	377F
48	14,208	3780	14,248	37A8	14,288	37D0	14,328	37F8	14,335	37FF
49	14,336	3800	14,376	3828	14,416	3850	14,456	3878	14,463	387F
50	14,464	3880	14,504	38A8	14,544	38D0	14,584	38F8	14,591	38FF
51	14,592	3900	14,632	3928	14,672	3950	14,712	3978	14,719	397F
52	14,720	3980	14,760	39A8	14,800	39D0	14,840	39F8	14,847	39FF
53	14,848	3A00	14,888	3A28	14,928	3A50	14,968	3A78	14,975	3A7F
54	14,976	3A80	15,016	3AA8	15,056	3AD0	15,096	3AF8	15,103	3AFF
55	15,104	3B00	15,144	3B28	15,184	3B50	15,224	3B78	15,231	3B7F
56	15,232	3B80	15,272	3BA8	15,312	3BD0	15,352	3BF8	15,359	3BFF
57	15,360	3C00	15,400	3C28	15,440	3C50	15,480	3C78	15,487	3C7F
58	15,488	3C80	15,528	3CA8	15,568	3CD0	15,608	3CF8	15,615	3CFF
59	15,616	3D00	15,656	3D28	15,696	3D50	15,736	3D78	15,743	3D7F
60	15,744	3D80	15,784	3DA8	15,824	3DD0	15,864	3DF8	15,871	3DFF
61	15,872	3E00	15,912	3E28	15,952	3E50	15,992	3E78	15,999	3E7F

LN NO.	FIRST LINE GROUP		SECOND LINE GROUP		THIRD LINE GROUP		THESE BYTES NOT DISPLAYED		END OF TRI-LINE	
	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
62	16,000	3E80	16,040	3EA8	16,080	3ED0	16,120	3EF8	16,127	3EFF
63	16,128	3F00	16,168	3F28	16,208	3F50	16,248	3F78	16,255	3F7F
64	16,256	3F80	16,296	3FA8	16,336	3FD0	16,376	3FF8	16,383	3FFF

HIRES TRI-LINE CHART PAGE 2

1	16,384	4000	16,424	4028	16,464	4050	16,504	4078	16,511	407F
2	16,512	4080	16,552	40A8	16,592	40D0	16,632	40F8	16,639	40FF
3	16,640	4100	16,680	4128	16,720	4150	16,760	4178	16,767	417F
4	16,768	4180	16,808	41A8	16,848	41D0	16,888	41F8	16,895	41FF
5	16,896	4200	16,936	4228	16,976	4250	17,016	4278	17,023	427F
6	17,024	4280	17,064	42A8	17,104	42D0	17,144	42F8	17,151	42FF
7	17,152	4300	17,192	4328	17,232	4350	17,272	4378	17,279	437F
8	17,280	4380	17,320	43A8	17,360	43D0	17,400	43F8	17,407	43FF
9	17,408	4400	17,448	4428	17,488	4450	17,528	4478	17,535	447F
10	17,536	4480	17,576	44A8	17,616	44D0	17,656	44F8	17,663	44FF
11	17,664	4500	17,704	4528	17,744	4550	17,784	4578	17,791	457F
12	17,792	4580	17,832	45A8	17,872	45D0	17,912	45F8	17,919	45FF
13	17,920	4600	17,960	4628	18,000	4650	18,040	4678	18,047	467F
14	18,048	4680	18,088	46A8	18,128	46D0	18,168	46F8	18,175	46FF
15	18,176	4700	18,216	4728	18,256	4750	18,296	4778	18,303	477F
16	18,304	4780	18,344	47A8	18,384	47D0	18,424	47F8	18,431	47FF
17	18,432	4800	18,472	4828	18,512	4850	18,552	4878	18,559	487F
18	18,560	4880	18,600	48A8	18,640	48D0	18,680	48F8	18,687	48FF
19	18,688	4900	18,728	4928	18,768	4950	18,808	4978	18,815	497F
20	18,816	4980	18,856	49A8	18,896	49D0	18,936	49F8	18,943	49FF
21	18,944	4A00	18,984	4A28	19,024	4A50	19,064	4A78	19,071	4A7F
22	19,072	4A80	19,112	4AA8	19,152	4AD0	19,192	4AF8	19,199	4AFF
23	19,200	4B00	19,240	4B28	19,280	4B50	19,320	4B78	19,327	4B7F
24	19,328	4B80	19,368	4BA8	19,408	4BD0	19,448	4BF8	19,455	4BFF
25	19,456	4C00	19,496	4C28	19,536	4C50	19,576	4C78	19,583	4C7F
26	19,584	4C80	19,624	4CA8	19,664	4CD0	19,704	4CF8	19,711	4CFF
27	19,712	4D00	19,752	4D28	19,792	4D50	19,832	4D78	19,839	4D7F
28	19,840	4D80	19,880	4DA8	19,920	4DD0	19,960	4DF8	19,967	4DFF
29	19,968	4E00	20,008	4E28	20,048	4E50	20,088	4E78	20,095	4E7F
30	20,096	4E80	20,136	4EA8	20,176	4ED0	20,216	4EF8	20,223	4EFF
31	20,224	4F00	20,264	4F28	20,304	4F50	20,344	4F78	20,351	4F7F
32	20,352	4F80	20,392	4FA8	20,432	4FD0	20,472	4FF8	20,479	4FFF
33	20,480	5000	20,520	5028	20,560	5050	20,600	5078	20,607	507F
34	20,608	5080	20,648	50A8	20,688	50D0	20,728	50F8	20,735	50FF
35	20,736	5100	20,776	5128	20,816	5150	20,856	5178	20,863	517F
36	20,864	5180	20,904	51A8	20,944	51D0	20,984	51F8	20,991	51FF
37	20,992	5200	21,032	5228	21,072	5250	21,112	5278	21,119	527F
38	21,120	5280	21,160	52A8	21,200	52D0	21,240	52F8	21,247	52FF
39	21,248	5300	21,288	5328	21,328	5350	21,368	5378	21,375	537F
40	21,376	5380	21,416	53A8	21,456	53D0	21,496	53F8	21,503	53FF
41	21,504	5400	21,544	5428	21,584	5450	21,624	5478	21,631	547F

LN NO.	FIRST LINE GROUP		SECOND LINE GROUP		THIRD LINE GROUP		THESE BYTES NOT DISPLAYED		END OF TRI-LINE	
	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
42	21,632	5480	21,672	54A8	21,712	54D0	21,752	54F8	21,759	54FF
43	21,760	5500	21,800	5528	21,840	5550	21,880	5578	21,887	557F
44	21,888	5580	21,928	55A8	21,968	55D0	22,008	55F8	22,015	55FF
45	22,016	5600	22,056	5628	22,096	5650	22,136	5678	22,143	567F
46	22,144	5680	22,184	56A8	22,224	56D0	22,264	56F8	22,271	56FF
47	22,272	5700	22,312	5728	22,352	5750	22,392	5778	22,399	577F
48	22,400	5780	22,440	57A8	22,480	57D0	22,520	57F8	22,527	57FF
49	22,528	5800	22,568	5828	22,608	5850	22,648	5878	22,655	587F
50	22,656	5880	22,696	58A8	22,736	58D0	22,776	58F8	22,783	58FF
51	22,784	5900	22,824	5928	22,864	5950	22,904	5978	22,911	597F
52	22,912	5980	22,952	59A8	22,992	59D0	23,032	59F8	23,039	59FF
53	23,040	5A00	23,080	5A28	23,120	5A50	23,160	5A78	23,167	5A7F
54	23,168	5A80	23,208	5AA8	23,248	5AD0	23,288	5AF8	23,295	5AFF
55	23,296	5B00	23,336	5B28	23,376	5B50	23,416	5B78	23,423	5B7F
56	23,424	5B80	23,464	5BA8	23,504	5BD0	23,544	5BF8	23,551	5BFF
57	23,552	5C00	23,592	5C28	23,632	5C50	23,672	5C78	23,679	5C7F
58	23,680	5C80	23,720	5CA8	23,760	5CD0	23,800	5CF8	23,807	5CFF
59	23,808	5D00	23,848	5D28	23,888	5D50	23,928	5D78	23,935	5D7F
60	23,936	5D80	23,976	5DA8	24,016	5DD0	24,056	5DF8	24,063	5DFF
61	24,064	5E00	24,104	5E28	24,144	5E50	24,184	5E78	24,191	5E7F
62	24,192	5E80	24,232	5EA8	24,272	5ED0	24,312	5EF8	24,319	5EFF
63	24,320	5F00	24,360	5F28	24,400	5F50	24,440	5F78	24,447	5F7F
64	24,448	5F80	24,488	5FA8	24,528	5FD0	24,568	5FF8	24,575	5FFF

HIRES TRI-LINES

TRI- LINE			
NUMBER	#1	#2	#3
1	0	64	128
2	8	72	136
3	16	80	144
4	24	88	152
5	32	96	160
6	40	104	168
7	48	112	176
8	56	120	184
9	1	65	129
10	9	73	137
11	17	81	145
12	25	89	153
13	33	97	161
14	41	105	169
15	49	113	177

TRI- LINE NUMBER #1	#2	#3
16	57	121
17	2	66
18	10	74
19	18	82
20	26	90
21	34	98
22	42	106
23	50	114
24	58	122
25	3	67
26	11	75
27	19	83
28	27	91
29	35	99
30	43	107
31	51	115
32	59	123
33	4	68
34	12	76
35	20	84
36	28	92
37	36	100
38	44	108
39	52	116
40	60	124
41	5	69
42	13	77
43	21	85
44	29	93
45	37	101
46	45	109
47	53	117
48	61	125
49	6	70
50	14	78
51	22	86
52	30	94
53	38	102
54	46	110
55	54	118
56	62	126
57	7	71
58	15	79
59	23	87
60	31	95
61	39	103
62	47	111
63	55	119
64	63	127

HIGH RESOLUTION SCREEN CHARACTER CHART

This chart presents the bit patterns for the hires screen characters generated in resolutions of 280 (traditional), 560 and 140 (modern). The screen character bit patterns show what color (V = violet, G = green, W = white, O = orange, B = blue) pixel results from entering the chart values in decimal or hexadecimal into hires screen addresses. By noting the various patterns generated by different values loaded into hires screen locations, this table is very useful for creating shapes and fonts as well as interesting bit pattern graphics.

The following program will print out all 255 bit patterns to give you a visual idea of what patterns will appear on your screen.

```
10 HGR : N = 1
20 FOR I = 8192 TO 18382 STEP 40
30 POKE I,N : N = N + 1 : NEXT I
```

TRADITIONAL					MODERN			
BYTE VALUE	BIT 280	PAT HGR	BIT 280	PAT HGR	BIT 560	PAT HGR	C 140	PAT HR
		SCN CHR		EVEN		ODD		EVEN
						B&W MODE		ODD
DEC HEX								
0	0	0123456	-----	-----	00112233445566		---	-
1	1	+123456	V-----	G-----	+0112233445566		V--	G---
2	2	0+23456	-G-----	-V-----	00+12233445566		G--	-V--
3	3	+ +23456	WW-----	WW-----	+0+12233445566		W--	G V--
4	4	01+3456	--V----	--G----	0011+233445566		-V-	-G--
5	5	+1+3456	V-V----	G-G----	+011+233445566		VV-	G G--
6	6	0++3456	-WW----	-WW----	00+1+233445566		GV-	-W--
7	7	+ + +3456	WWW----	WWW----	+0+1+233445566		WV-	G W--
8	8	012+456	---G---	---V---	001122+3445566		-G-	- -V-
9	9	+12+456	V--G---	G--V---	+01122+3445566		VG-	G -V-
10	A	0+2+456	-G-G---	-V-V---	00+122+3445566		GG-	- -VV-
11	B	+ +2+456	WW-G---	WW-V---	+0+122+3445566		WG-	G -V-
12	C	01++456	--WW---	--VV---	0011+2+3445566		-W-	- -GV-
13	D	+1++456	V-WW---	G-WW---	+011+2+3445566		VW-	G -GV-
14	E	0+++456	-WWW---	-VVV---	00+1+2+3445566		GW-	- -WV-
15	F	+ + + +456	WWWW---	VVVV---	+0+1+2+3445566		WW-	G -WV-
16	10	0123+56	----V--	----G--	00112233+45566		--V-	- - -G-
17	11	+123+56	V---V--	G---G--	+0112233+45566		V-V-	G - -G-
18	12	0+23+56	-G---V--	-V---G--	00+12233+45566		G-V-	- -VG-
19	13	+ +23+56	WW--V--	WW--G--	+0+12233+45566		W-V-	G -VG-
20	14	01+3+56	--V-V--	--G-G--	0011+233+45566		-VV-	- -GG-
21	15	+1+3+56	V-V-V--	G-G-G--	+011+233+45566		VVV-	G -GG-
22	16	0++3+56	-WW-V--	-VV-G--	00+1+233+45566		GVV-	- -WG-
23	17	+ + +3+56	WWW-V--	VVV-G--	+0+1+233+45566		WVV-	G -WG-
24	18	012++56	---WW--	---VV--	001122+3+45566		-GV-	- - -W-
25	19	+12++56	V---WW--	G---VV--	+01122+3+45566		VGv-	G - -W-
26	1A	0+2++56	-G---WW--	-V---VV--	00+122+3+45566		GGV-	- - -VW-
27	1B	+ +2++56	WW--WW--	VV--VV--	+0+122+3+45566		WGV-	G -VW-
28	1C	01+++56	--WWW--	--VVV--	0011+2+3+45566		-WV-	- -GW-

BYTE VALUE	280 HGR SCN CHR	280 HGR EVEN	280 HGR ODD	560 HGR B&W MODE	140 HR EVEN	140 HR ODD
DEC HEX						
29	1D	+1+++56	V-WWW--	G-WWW--	+011+2+3+45566	VWV - G GW-
30	1E	0++++56	-WWW--	-WWW--	00+1+2+3+45566	GWV - - WW-
31	1F	+++++56	WWWW--	WWWW--	+0+1+2+3+45566	WWV - G WW-
32	20	01234+6	-----G-	-----V-	0011223344+566	--G - - --V
33	21	+1234+6	V----G-	G----V-	+011223344+566	V-G - G --V
34	22	0+234+6	-G---G-	-V---V-	00+1223344+566	G-G - - V-V
35	23	++234+6	WW---G-	WW---V-	+0+1223344+566	W-G - G V-V
36	24	01+34+6	--V--G-	--G--V-	0011+23344+566	-VG - - G-V
37	25	+1+34+6	V-V--G-	G-G--V-	+011+23344+566	VVG - G G-V
38	26	0++34+6	-WW--G-	-WW--V-	00+1+23344+566	GVG - - W-V
39	27	+++34+6	WWW--G-	WWW--V-	+0+1+23344+566	WVG - G W-V
40	28	012+4+6	---G-G-	---V-V-	001122+344+566	-GG - - -VV
41	29	+12+4+6	V--G-G-	G--V-V-	+01122+344+566	VGG - G -VV
42	2A	0+2+4+6	-G-G-G-	-V-V-V-	00+122+344+566	GGG - - VVV
43	2B	++2+4+6	WW-G-G-	WW-V-V-	+0+122+344+566	WGG - G VVV
44	2C	01+4+6	--WW-G-	--WW-V-	0011+2+344+566	-WG - - GVV
45	2D	+1+4+6	V-WW-G-	G-WW-V-	+011+2+344+566	VWG - G GVV
46	2E	0++4+6	-WWW-G-	-WWW-V-	00+1+2+344+566	GWG - - WVV
47	2F	+++4+6	WWW-G-	WWW-V-	+0+1+2+344+566	WWG - G WVV
48	30	0123++6	----WW-	----WW-	00112233+4+566	--W - - -GV
49	31	+123++6	V---WW-	G---WW-	+0112233+4+566	V-W - G -GV
50	32	0+23++6	-G---WW-	-V---WW-	00+12233+4+566	G-W - - VGV
51	33	++23++6	WW---WW-	WW---WW-	+0+12233+4+566	W-W - G VGV
52	34	01+3++6	--V-WW-	--G-WW-	0011+233+4+566	-VW - - GGV
53	35	+1+3++6	V-V-WW-	G-G-WW-	+011+233+4+566	VVW - G GGV
54	36	0++3++6	-WW-WW-	-WW-WW-	00+1+233+4+566	GVW - - WGV
55	37	+++3++6	WWW-WW-	WWW-WW-	+0+1+233+4+566	WVW - G WGV
56	38	012++6	---WWW-	---WWW-	001122+3+4+566	-GW - - -WV
57	39	+12++6	V---WWW-	G---WWW-	+01122+3+4+566	VGW - G -WV
58	3A	0+2++6	-G---WWW-	-V---WWW-	00+122+3+4+566	GGW - - VVW
59	3B	++2++6	WW---WWW-	WW---WWW-	+0+122+3+4+566	WGW - G VVW
60	3C	01+++6	--WWW-	--WWW-	0011+2+3+4+566	-WW - - GWV
61	3D	+1+++6	V-WWW-	G-WWW-	+011+2+3+4+566	VWW - G GWV
62	3E	0++++6	-WWWW-	-WWWW-	00+1+2+3+4+566	GWV - - WWV
63	3F	+++++6	WWWWW-	WWWWW-	+0+1+2+3+4+566	WWW - G WWV
64	40	012345+	-----V	-----G	001122334455+6	--- V - --G
65	41	+12345+	V-----V	G-----G	+01122334455+6	V-- V G --G
66	42	0+2345+	-G-----V	-V-----G	00+122334455+6	G-- V - V-G
67	43	++2345+	WW-----V	WW-----G	+0+122334455+6	W-- V G V-G
68	44	01+345+	--V---V	--G---G	0011+2334455+6	-V- V - G-G
69	45	+1+345+	V-V---V	G-G---G	+011+2334455+6	VV- V G G-G
70	46	0++345+	-WW---V	-WW---G	00+1+2334455+6	GV- V - W-G
71	47	+++345+	WWW---V	WWW---G	+0+1+2334455+6	WV- V G W-G
72	48	012+45+	---G---V	---V---G	001122+34455+6	-G- V - -VG
73	49	+12+45+	V--G---V	G--V---G	+01122+34455+6	VG- V G -VG
74	4A	0+2+45+	-G-G---V	-V-V---G	00+122+34455+6	GG- V - VVG
75	4B	++2+45+	WW-G---V	WW-V---G	+0+122+34455+6	WG- V G VVG
76	4C	01++45+	--WW---V	--WW---G	0011+2+34455+6	-W- V - GVG
77	4D	+1++45+	V-WW---V	G-WW---G	+011+2+34455+6	VW- V G GVG
78	4E	0++45+	-WWW---V	-WWW---G	00+1+2+34455+6	GW- V - WVG

	BYTE VALUE	280 HGR SCN CHR	280 HGR EVEN	280 HGR ODD	560 HGR B&W MODE	140 HR EVEN	140 HR ODD
DEC HEX							
79	4F	++++45+	WWW--V	WWW--G	+0+1+2+34455+6	WW- V	G WVG
80	50	0123+5+	----V-V	----G-G	00112233+455+6	--V V	- -GG
81	51	+123+5+	V--V-V	G--G-G	+0112233+455+6	V-V V	G -GG
82	52	0+23+5+	-G--V-V	-V--G-G	00+12233+455+6	G-V V	- VGG
83	53	++23+5+	WW--V-V	WW--G-G	+0+12233+455+6	W-V V	G VGG
84	54	01+3+5+	--V-V-V	--G-G-G	0011+233+455+6	-VV V	- GGG
85	55	+1+3+5+	V-V-V-V	G-G-G-G	+011+233+455+6	VVV V	G GGG
86	56	0++3+5+	-WW-V-V	-WW-G-G	00+1+233+455+6	GVV V	- WGG
87	57	+++3+5+	WWW-V-V	WWW-G-G	+0+1+233+455+6	WVV V	G WGG
88	58	012++5+	---WW-V	---WW-G	001122+3+455+6	-GV V	- -WG
89	59	+12++5+	V--WW-V	G--WW-G	+01122+3+455+6	VG V	G -WG
90	5A	0+2++5+	-G--WW-V	-V--WW-G	00+122+3+455+6	GGV V	- VWG
91	5B	++2++5+	WW--WW-V	WW--WW-G	+0+122+3+455+6	WGV V	G VWG
92	5C	01++5+	--WWW-V	--WWW-G	0011+2+3+455+6	-WV V	- GWG
93	5D	+1++5+	V-WWW-V	G-WWW-G	+011+2+3+455+6	VWV V	G GWG
94	5E	0++++5+	-WWW-V	-WWW-G	00+1+2+3+455+6	GWV V	- WWG
95	5F	++++5+	WWWWW-V	WWWWW-G	+0+1+2+3+455+6	WVV V	G WWG
96	60	01234++	-----WW	-----WW	0011223344+5+6	--G V	- --W
97	61	+1234++	V-----WW	G-----WW	+011223344+5+6	V-G V	G --W
98	62	0+234++	-G-----WW	-V-----WW	00+1223344+5+6	G-G V	- V-W
99	63	++234++	WW-----WW	WW-----WW	+0+1223344+5+6	W-G V	G V-W
100	64	01+34++	--V--WW	--G--WW	0011+23344+5+6	-VG V	- G-W
101	65	+1+34++	V-V--WW	G-G--WW	+011+23344+5+6	VVG V	G G-W
102	66	0++34++	-WW--WW	-WW--WW	00+1+23344+5+6	GVG V	- W-W
103	67	+++34++	WWW--WW	WWW--WW	+0+1+23344+5+6	WVG V	G W-W
104	68	012+4++	---G-WW	---V-WW	001122+344+5+6	-GG V	- -VW
105	69	+12+4++	V--G-WW	G--V-WW	+01122+344+5+6	VGG V	G -VW
106	6A	0+2+4++	-G-G-WW	-V-V-WW	00+122+344+5+6	GGG V	- VWV
107	6B	++2+4++	WW-G-WW	WW-V-WW	+0+122+344+5+6	WGG V	G VWV
108	6C	01++4++	--WW-WW	--WW-WW	0011+2+344+5+6	-WG V	- GVW
109	6D	+1++4++	V-WW-WW	G-WW-WW	+011+2+344+5+6	VWG V	G GVW
110	6E	0+++4++	-WWW-WW	-WWW-WW	00+1+2+344+5+6	GWG V	- WVW
111	6F	++++4++	WWWW-WW	WWWW-WW	+0+1+2+344+5+6	WVG V	G WVW
112	70	0123+++	----WWW	----WWW	00112233+4+5+6	--W V	- -GW
113	71	+123+++	V---WWW	G---WWW	+0112233+4+5+6	V-W V	G -GW
114	72	0+23+++	-G---WWW	-V---WWW	00+12233+4+5+6	G-W V	- VGW
115	73	++23+++	WW---WWW	WW---WWW	+0+12233+4+5+6	W-W V	G VGW
116	74	01+3+++	--V-WWW	--G-WWW	0011+233+4+5+6	-VW V	- GGW
117	75	+1+3+++	V-V-WWW	G-G-WWW	+011+233+4+5+6	VVW V	G GGW
118	76	0++3+++	-WW-WWW	-WW-WWW	00+1+233+4+5+6	GVW V	- WGW
119	77	+++3+++	WWW-WWW	WWW-WWW	+0+1+233+4+5+6	WVW V	G WGW
120	78	012++++	---WWWW	---WWWW	001122+3+4+5+6	-GW V	- -WW
121	79	+12++++	V--WWWW	G--WWWW	+01122+3+4+5+6	VGW V	G -WW
122	7A	0+2++++	-G-WWWW	-V-WWWW	00+122+3+4+5+6	GGW V	- VWW
123	7B	++2++++	WW-WWWW	WW-WWWW	+0+122+3+4+5+6	WGW V	G VWW
124	7C	01+++++	--WWWWW	--WWWWW	0011+2+3+4+5+6	-WV V	- GWW
125	7D	+1+++++	V-WWWWW	G-WWWWW	+011+2+3+4+5+6	VVW V	G GWW
126	7E	0+++++	-WWWWW	-WWWWW	00+1+2+3+4+5+6	GWV V	- WWW
127	7F	++++++	WWWWWWW	WWWWWWW	+0+1+2+3+4+5+6	WVV V	G WWW

BYTE VALUE	280 HGR SCN CHR	280 HGR EVEN	280 HGR ODD	560 HGR B&W MODE	140 HR EVEN	140 HR ODD
DEC HEX						
128	80	0123456	-----	00112233445566	---	---
129	81	+123456	B-----	0+112233445566	B--	O---
130	82	0+23456	-O-----	001+2233445566	O--	-B--
131	83	++23456	WW-----	0+1+2233445566	W--	O B--
132	84	01+3456	--B-----	00112+33445566	-B-	-O--
133	85	+1+3456	B-B----	0+112+33445566	BB-	O O--
134	86	0++3456	-WW----	001+2+33445566	OB-	-W--
135	87	++3456	WWW----	0+1+2+33445566	WB-	O W--
136	88	012+456	---O----	0011223+445566	-O-	-B-
137	89	+12+456	B--O----	0+11223+445566	BO-	O -B-
138	8A	0+2+456	-O-O----	001+223+445566	O-B-	-BB-
139	8B	++2+456	WW-O----	0+1+223+445566	WO-	O BB-
140	8C	01++456	--WW----	00112+3+445566	-W-	-OB-
141	8D	+1++456	B-WW----	0+112+3+445566	BW-	O OB-
142	8E	0+++456	-WWW----	001+2+3+445566	OW-	-WB-
143	8F	+++456	WWWW----	0+1+2+3+445566	WW-	O WB-
144	90	0123+56	----B--	001122334+5566	--B-	-O-
145	91	+123+56	B---B--	0+1122334+5566	B-B-	O -O-
146	92	0+23+56	-O--B--	001+22334+5566	O-B-	-BO-
147	93	++23+56	WW--B--	0+1+22334+5566	W-B-	O BO-
148	94	01+3+56	--B-B--	00112+334+5566	-BB-	-OO-
149	95	+1+3+56	B-B-B--	0+112+334+5566	BBB-	O OO-
150	96	0++3+56	-WW-B--	001+2+334+5566	BBB-	-WO-
151	97	++3+56	WWW-B--	0+1+2+334+5566	WBB-	O WO-
152	98	012++56	---WW--	0011223+4+5566	-OB-	-W-
153	99	+12++56	B--WW--	0+11223+4+5566	BOB-	O -W-
154	9A	0+2++56	-O-WW--	001+223+4+5566	OBB-	-BW-
155	9B	++2++56	WW-WW--	0+1+223+4+5566	WOB-	O BW-
156	9C	01++56	--WWW--	00112+3+4+5566	-WB-	-OW-
157	9D	+1++56	B-WWW--	0+112+3+4+5566	BWB-	O OW-
158	9E	0+++56	-WWW--	001+2+3+4+5566	OWB-	-WW-
159	9F	++++56	WWWW--	0+1+2+3+4+5566	WWB-	O WW-
160	A0	01234+6	-----O-	00112233445+66	--O-	--B
161	A1	+1234+6	B----O-	0+112233445+66	B-O-	O --B
162	A2	0+234+6	-O---O-	001+2233445+66	O-O-	-B-B
163	A3	++234+6	WW---O-	0+1+2233445+66	W-O-	O B-B
164	A4	01+34+6	--B--O-	00112+33445+66	-BO-	-O-B
165	A5	+1+34+6	B-B--O-	0+112+33445+66	BBO-	O O-B
166	A6	0++34+6	-WW--O-	001+2+33445+66	OBO-	-W-B
167	A7	++34+6	WWW--O-	0+1+2+33445+66	WBO-	O W-B
168	A8	012+4+6	---O-O-	0011223+445+66	-OO-	-BB
169	A9	+12+4+6	B--O-O-	0+11223+445+66	BOO-	O -BB
170	AA	0+2+4+6	-O-O-O-	001+223+445+66	OOO-	-BBB
171	AB	++2+4+6	WW-O-O-	0+1+223+445+66	WOO-	O BBB
172	AC	01++4+6	--WW-O-	00112+3+445+66	-WO-	-OBB
173	AD	+1++4+6	B-WW-O-	0+112+3+445+66	BWO-	O OBB
174	AE	0+++4+6	-WWW-O-	001+2+3+445+66	OWO-	-WBB
175	AF	+++4+6	WWWW-O-	0+1+2+3+445+66	WWO-	O WBB
176	B0	0123++6	----WW-	001122334+5+66	--W-	-OB
177	B1	+123++6	B---WW-	0+1122334+5+66	B-W-	O -OB

	BYTE VALUE	280 HGR SCN CHR	280 HGR EVEN	280 HGR ODD	560 HGR B&W MODE	140 HR EVEN	140 HR ODD
DEC HEX							
178	B2	0+23++6	-O--WW-	-B--WW-	001+22334+5+66	O-W -	- BOB
179	B3	++23++6	WW--WW-	WW--WW-	0+1+22334+5+66	W-W -	O BOB
180	B4	01+3++6	--B-WW-	--O-WW-	00112+334+5+66	-BW -	- OOB
181	B5	+1+3++6	B-B-WW-	O-O-WW-	0+112+334+5+66	BBW -	O OOB
182	B6	0++3++6	-WW-WW-	-WW-WW-	001+2+334+5+66	OBW -	- WOB
183	B7	+++3++6	WWW-WW-	WWW-WW-	0+1+2+334+5+66	WBW -	O WOB
184	B8	012++6	---WWW-	---WWW-	0011223+4+5+66	-OW -	- -WB
185	B9	+12++6	B---WWW-	O---WWW-	0+11223+4+5+66	BOW -	O -WB
186	BA	0+2++6	-O-WWW-	-B-WWW-	001+223+4+5+66	OOW -	- BWB
187	BB	++2++6	WW-WWW-	WW-WWW-	0+1+223+4+5+66	WOW -	O BWB
188	BC	01+++6	--WWW-	--WWW-	00112+3+4+5+66	-WW -	- OWB
189	BD	+1++++6	B-WWWW-	O-WWWW-	0+112+3+4+5+66	BBW -	O OWB
190	BE	0++++6	-WWWWW-	-WWWWW-	001+2+3+4+5+66	OOW -	- WWB
191	BF	+++++6	WWWWW-	WWWWW-	0+1+2+3+4+5+66	WWW -	O WWB
192	C0	012345+	-----B	-----O	0011223344556+	--- B	- --O
193	C1	+12345+	B-----B	O-----O	0+11223344556+	B-- B	O --O
194	C2	0+2345+	-O-----B	-B-----O	001+223344556+	O-- B	- B-O
195	C3	++2345+	WW----B	WW----O	0+1+223344556+	W-- B	O B-O
196	C4	01+345+	--B---B	--O---O	00112+3344556+	-B- B	- O-O
197	C5	+1+345+	B-B---B	O-O---O	0+112+3344556+	BB- B	O O-O
198	C6	0++345+	-WW---B	-WW---O	001+2+3344556+	OB- B	- W-O
199	C7	++345+	WWW---B	WWW---O	0+1+2+3344556+	WB- B	O W-O
200	C8	012+45+	---O--B	---B--O	0011223+44556+	-O- B	- -BO
201	C9	+12+45+	B--O--B	O--B--O	0+11223+44556+	BO- B	O -BO
202	CA	0+2+45+	-O-O--B	-B-B--O	001+223+44556+	OO- B	- BBO
203	CB	++2+45+	WW-O--B	WW-B--O	0+1+223+44556+	WO- B	O BBO
204	CC	01++45+	--WW--B	--WW--O	00112+3+44556+	-W- B	- OBO
205	CD	+1++45+	B-WW--B	O-WW--O	0+112+3+44556+	BW- B	O OBO
206	CE	0++45+	-WWW--B	-WWW--O	001+2+3+44556+	OW- B	- WBO
207	CF	+++45+	WWW--B	WWW--O	0+1+2+3+44556+	WW- B	O WBO
208	D0	0123+5+	----B-B	----O-O	001122334+556+	--B B	- -OO
209	D1	+123+5+	B---B-B	O---O-O	0+1122334+556+	B-B B	O -OO
210	D2	0+23+5+	-O--B-B	-B--O-O	001+22334+556+	O-B B	- BOO
211	D3	++23+5+	WW--B-B	WW--O-O	0+1+22334+556+	W-B B	O BOO
212	D4	01+3+5+	--B-B-B	--O-O-O	00112+334+556+	-BB B	- OOO
213	D5	+1+3+5+	B-B-B-B	O-O-O-O	0+112+334+556+	BBB B	O OOO
214	D6	0++3+5+	-WW-B-B	-WW-O-O	001+2+334+556+	OB B	- WOO
215	D7	++3+5+	WWW-B-B	WWW-O-O	0+1+2+334+556+	WB B	O WOO
216	D8	012++5+	---WW-B	---WW-O	0011223+4+556+	-OB B	- -WO
217	D9	+12++5+	B--WW-B	O--WW-O	0+11223+4+556+	BOB B	O -WO
218	DA	0+2++5+	-O-WW-B	-B-WW-O	001+223+4+556+	O B B	- BWO
219	DB	++2++5+	WW-WW-B	WW-WW-O	0+1+223+4+556+	W B B	O BWO
220	DC	01++5+	--WWW-B	--WWW-O	00112+3+4+556+	-WB B	- OWO
221	DD	+1++5+	B-WWW-B	O-WWW-O	0+112+3+4+556+	BWB B	O OWO
222	DE	0++++5+	-WWWW-B	-WWWW-O	001+2+3+4+556+	OWB B	- WWO
223	DF	+++++5+	WWWWW-B	WWWWW-O	0+1+2+3+4+556+	WWB B	O WWO
224	E0	01234++	-----WW	-----WW	00112233445+6+	--O B	- --W
225	E1	+1234++	B-----WW	O-----WW	0+112233445+6+	B-O B	O --W
226	E2	0+234++	-O-----WW	-B-----WW	001+2233445+6+	O-O B	- B-W
227	E3	++234++	WW---WW	WW---WW	0+1+2233445+6+	W-O B	O B-W

BYTE VALUE	280 HGR SCN CHR	280 HGR EVEN	280 HGR ODD	560 HGR B&W MODE	140 HR EVEN	140 HR ODD
DEC HEX						
228 E4	01+34++	--B--WW	--O--WW	00112+33445+6+	-BO B	- O-W
229 E5	+1+34++	B-B--WW	O-O--WW	0+112+33445+6+	BBO B	O O-W
230 E6	0++34++	-WW--WW	-WW--WW	001+2+33445+6+	OBO B	- W-W
231 E7	+++34++	WWW--WW	WWW--WW	0+1+2+33445+6+	WBO B	O W-W
232 E8	012+4++	---O-WW	---B-WW	0011223+445+6+	-OO B	- -BW
233 E9	+12+4++	B--O-WW	O--B-WW	0+11223+445+6+	BOO B	O -BW
234 EA	0+2+4++	-O-O-WW	-B-B-WW	001+223+445+6+	OOO B	- BBW
235 EB	++2+4++	WW-O-WW	WW-B-WW	0+1+223+445+6+	WOO B	O BBW
236 EC	01++4++	--WW-WW	--WW-WW	00112+3+445+6+	-WO B	- OBW
237 ED	+1++4++	B-WW-WW	O-WW-WW	0+112+3+445+6+	BWO B	O OBW
238 EE	0+++4++	-WWW-WW	-WWW-WW	001+2+3+445+6+	OWO B	- WBW
239 EF	++++4++	WWW-WW	WWW-WW	0+1+2+3+445+6+	WWO B	O WBW
240 F0	0123+++	----WWW	----WWW	001122334+5+6+	--W B	- -OW
241 F1	+123+++	B---WWW	O---WWW	0+1122334+5+6+	B-W B	O -OW
242 F2	0+23+++	-O---WWW	-B---WWW	001+22334+5+6+	O-W B	- BOW
243 F3	++23+++	WW---WWW	WW---WWW	0+1+22334+5+6+	W-W B	O BOW
244 F4	01+3+++	--B-WWW	--O-WWW	00112+334+5+6+	-BW B	- OOW
245 F5	+1+3+++	B-B-WWW	O-O-WWW	0+112+334+5+6+	BBW B	O OOW
246 F6	0++3+++	-WW-WWW	-WW-WWW	001+2+334+5+6+	OBW B	- WOW
247 F7	+++3+++	WWW-WWW	WWW-WWW	0+1+2+334+5+6+	WBW B	O WOW
248 F8	012++++	---WWW	---WWW	0011223+4+5+6+	-OW B	- -WW
249 F9	+12++++	B---WWW	O---WWW	0+11223+4+5+6+	BOW B	O -WW
250 FA	0+2++++	-O---WWW	-B---WWW	001+223+4+5+6+	OOW B	- BWW
251 FB	++2++++	WW---WWW	WW---WWW	0+1+223+4+5+6+	WOW B	O BWW
252 FC	01++++	--WWW	--WWW	00112+3+4+5+6+	-WW B	- OWW
253 FD	+1++++	B-WWWW	O-WWWW	0+112+3+4+5+6+	BWW B	O OWW
254 FE	0++++	-WWWWW	-WWWWW	001+2+3+4+5+6+	OWW B	- WWW
255 FF	+++++	WWWWW	WWWWW	0+1+2+3+4+5+6+	WWW B	O WWW



DECODE HIGH RESOLUTION SHAPES

This table represents the 256 combinations of moving and plotting/not plotting shapes. The vectors displayed in this table can be understood in terms of where a plot occurs and the ending location for the next plot/move. Each *byte* has up to two plots and an ending location. The ending move is represented in parentheses as an X,Y coordinate. The X is the horizontal movement and Y the vertical. Positive numbers represent upward movement on the vertical axis and towards the right on the horizontal. Negative numbers represent down vertically and left horizontally. A "0" coordinate/offset is "no movement" in a given direction. For example, (1,0) means moving to the right 1 and 0 up or down.

The numbers to the left of the location number are X,Y coordinate offsets and are the location of the plots or ending location relative to location number 12 (0,0). The following table shows these locations relative to "12" at the 0,0 location:

Ending Location Table

		1 -1, 2	2 0, 2	3 1, 2		
	4 -2, 1	5 -1, 1	6 0, 1	7 1, 1	8 2, 1	
9 -3, 0	10 -2, 0	11 -1, 0	12 0, 0	13 1, 0	14 2, 0	15 3, 0
	16 -2, -1	17 -1, -1	18 0, -1	19 1, -1	20 2, -1	
		21 -1, -2	22 0, -2	23 1, -2		
			24 0, -3			

Plotting occurs in some of the bytes of a shape but not in others. There are two coordinates; those for dots plotted (left side of each column) and the ending coordinates (right side of each column), while for others there are only the ending coordinates. Each, however, can be understood using the Location Table and examining the X,Y values. For example, byte 224 (\$E0) moves up two vertically and one to the left. Along the way it plots a point (location 6), continuing movement without plotting and finally ending up in position "1" on the Location Table. The "DOT PAT FROM CUR LOC" (dot pattern from current location) column shows the patterns of points plotted with a "+" for plotted points. Each byte is assumed to begin at position "12" no matter what the previous byte in a shape table was. For example, the partial shape pattern for byte 224 resulted in the plot point ending at location "1", but in determining the next plot with a new byte, we consider its starting position to be location "12" on the Location Table.

DOT PAT FROM CUR				DOT PAT FROM CUR					
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
0	0	---	---	NO MOVE TERMINATE SHAPE	128	80	---	---	6 (0, 1)
1	1	---	---	13 (1, 0)	129	81	---	---	13 (1, 0)
2	2	---	---	18 (0, -1)	130	82	---	---	18 (0, -1)
3	3	---	---	11 (-1, 0)	131	83	---	---	11 (-1, 0)
4	4	--- -+- ---	12 (0, 0)	6 (0, 1)	132	84	--- -+- ---	12 (0, 0)	6 (0, 1)
5	5	--- -+- ---	12 (0, 0)	13 (1, 0)	133	85	--- -+- ---	12 (0, 0)	13 (1, 0)
6	6	--- -+- ---	12 (0, 0)	18 (0, -1)	134	86	--- -+- ---	12 (0, 0)	18 (0, -1)
7	7	--- -+- ---	12 (0, 0)	11 (-1, 0)	135	87	--- -+- ---	12 (0, 0)	11 (-1, 0)
8	8	---	---	7 (1, 1)	136	88	---	---	13 (1, 0)
9	9	---	---	14 (2, 0)	137	89	---	---	20 (2, -1)
10	A	---	---	19 (1, -1)	138	8A	---	---	23 (1, -2)
11	B	---	---	12 (0, 0)	139	8B	---	---	18 (0, -1)

DOT PAT FROM CUR				DOT PAT FROM CUR					
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
12	C	--- -+- ---	12 (0, 0)	7 (1, 1)	140	8C	--- -+- ---	12 (0, 0)	13 (1, 0)
13	D	--- -+- ---	12 (0, 0)	14 (2, 0)	141	8D	--- -+- ---	12 (0, 0)	20 (2, -1)
14	E	--- -+- ---	12 (0, 0)	19 (1, -1)	142	8E	--- -+- ---	12 (0, 0)	23 (1, -2)
15	F	--- -+- ---	12 (0, 0)	12 (0, 0)	143	8F	--- -+- ---	12 (0, 0)	18 (0, -1)
16	10	--- --- ---		12 (0, 0)	144	90	--- --- ---		18 (0, -1)
17	11	--- --- ---		19 (1, -1)	145	91	--- --- ---		23 (1, -2)
18	12	--- --- ---		22 (0, -2)	146	92	--- --- ---		24 (0, -3)
19	13	--- --- ---		17 (-1, -1)	147	93	--- --- ---		21 (-1, -2)
20	14	--- -+- ---	12 (0, 0)	12 (0, 0)	148	94	--- -+- ---	12 (0, 0)	18 (0, -1)
21	15	--- -+- ---	12 (0, 0)	19 (1, -1)	149	95	--- -+- ---	12 (0, 0)	23 (1, -2)
22	16	--- -+- ---	12 (0, 0)	22 (0, -2)	150	96	--- -+- ---	12 (0, 0)	24 (0, -3)
23	17	--- -+- ---	12 (0, 0)	17 (-1, -1)	151	97	--- -+- ---	12 (0, 0)	21 (-1, -2)

DOT PAT FROM CUR				DOT PAT FROM CUR					
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
24	18	---	---	5 (-1, 1)	152	98	---	---	11 (-1, 0)
25	19	---	---	12 (0, 0)	153	99	---	---	18 (0, -1)
26	1A	---	---	17 (-1, -1)	154	9A	---	---	21 (-1, -2)
27	1B	---	---	10 (-2, 0)	155	9B	---	---	16 (-2, -1)
28	1C	+-	12 (0, 0)	5 (-1, 1)	156	9C	+-	12 (0, 0)	11 (-1, 0)
29	1D	+-	12 (0, 0)	12 (0, 0)	157	9D	+-	12 (0, 0)	18 (0, -1)
30	1E	+-	12 (0, 0)	17 (-1, -1)	158	9E	+-	12 (0, 0)	21 (-1, -2)
31	1F	+-	12 (0, 0)	10 (-2, 0)	159	9F	+-	12 (0, 0)	16 (-2, -1)
32	20	+-	6 (0, 1)	2 (0, 2)	160	A0	+-	6 (0, 1)	6 (0, 1)
33	21	++	13 (1, 0)	7 (1, 1)	161	A1	++	13 (1, 0)	13 (1, 0)
34	22	++	18 (0, -1)	12 (0, 0)	162	A2	++	18 (0, -1)	18 (0, -1)
35	23	++	11 (-1, 0)	5 (-1, 1)	163	A3	++	11 (-1, 0)	11 (-1, 0)

DOT PAT FROM CUR					DOT PAT FROM CUR				
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
36	24	-+- -+- ---	12 (0, 0) 6 (0, 1)	2 (0, 2)	164	A4	-+- -+- ---	12 (0, 0) 6 (0, 1)	6 (0, 1)
37	25	--- -++ ---	12 (0, 0) 13 (1, 0)	7 (1, 1)	165	A5	--- -++ ---	12 (0, 0) 13 (1, 0)	13 (1, 0)
38	26	--- -+- -+-	12 (0, 0) 18 (0, -1)	12 (0, 0)	166	A6	--- -+- -+-	12 (0, 0) 18 (0, -1)	18 (0, -1)
39	27	--- ++- ---	12 (0, 0) 11 (-1, 0)	5 (-1, 1)	167	A7	--- ++- ---	12 (0, 0) 11 (-1, 0)	11 (-1, 0)
40	28	-+- --- ---	6 (0, 1)	7 (1, 1)	168	A8	-+- --- ---	6 (0, 1)	13 (1, 0)
41	29	--- --+ ---	13 (1, 0)	14 (2, 0)	169	A9	--- --+ ---	13 (1, 0)	20 (2, -1)
42	2A	--- --- -+-	18 (0, -1)	19 (1, -1)	170	AA	--- --- -+-	18 (0, -1)	23 (1, -2)
43	2B	--- +-- ---	11 (-1, 0)	12 (0, 0)	171	AB	--- +-- ---	11 (-1, 0)	18 (0, -1)
44	2C	-+- -+- ---	12 (0, 0) 6 (0, 1)	7 (1, 1)	172	AC	-+- -+- ---	12 (0, 0) 6 (0, 1)	13 (1, 0)
45	2D	--- -++ ---	12 (0, 0) 13 (1, 0)	14 (2, 0)	173	AD	--- -++ ---	12 (0, 0) 13 (1, 0)	20 (2, -1)
46	2E	--- -+- -+-	12 (0, 0) 18 (0, -1)	19 (1, -1)	174	AE	--- -+- -+-	12 (0, 0) 18 (0, -1)	23 (1, -2)
47	2F	--- ++- ---	12 (0, 0) 11 (-1, 0)	12 (0, 0)	175	AF	--- ++- ---	12 (0, 0) 11 (-1, 0)	18 (0, -1)
48	30	-+- --- ---	6 (0, 1)	12 (0, 0)	176	B0	-+- --- ---	6 (0, 1)	18 (0, -1)

DOT PAT FROM CUR				DOT PAT FROM CUR					
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
49	31	---	---	13 (1, 0) 19 (1, -1)	177	B1	---	---	13 (1, 0) 23 (1, -2)
		---	---				---	---	
50	32	---	---	18 (0, -1) 22 (0, -2)	178	B2	---	---	18 (0, -1) 24 (0, -3)
		---	---				---	---	
51	33	---	---	11 (-1, 0) 17 (-1, -1)	179	B3	---	---	11 (-1, 0) 21 (-1, -2)
		---	---				---	---	
52	34	---	---	12 (0, 0) 12 (0, 0)	180	B4	---	---	12 (0, 0) 18 (0, -1)
		---	---	6 (0, 1)			---	---	6 (0, 1)
53	35	---	---	12 (0, 0) 19 (1, -1)	181	B5	---	---	12 (0, 0) 23 (1, -2)
		---	---	13 (1, 0)			---	---	13 (1, 0)
54	36	---	---	12 (0, 0) 22 (0, -2)	182	B6	---	---	12 (0, 0) 24 (0, -3)
		---	---	18 (0, -1)			---	---	18 (0, -1)
55	37	---	---	12 (0, 0) 17 (-1, -1)	183	B7	---	---	12 (0, 0) 21 (-1, -2)
		---	---	11 (-1, 0)			---	---	11 (-1, 0)
56	38	---	---	6 (0, 1) 5 (-1, 1)	184	B8	---	---	6 (0, 1) 11 (-1, 0)
		---	---				---	---	
57	39	---	---	13 (1, 0) 12 (0, 0)	185	B9	---	---	13 (1, 0) 18 (0, -1)
		---	---				---	---	
58	3A	---	---	18 (0, -1) 17 (-1, -1)	186	BA	---	---	18 (0, -1) 21 (-1, -2)
		---	---				---	---	
59	3B	---	---	11 (-1, 0) 10 (-2, 0)	187	BB	---	---	11 (-1, 0) 16 (-2, -1)
		---	---				---	---	
60	3C	---	---	12 (0, 0) 5 (-1, 1)	188	BC	---	---	12 (0, 0) 11 (-1, 0)
		---	---	6 (0, 1)			---	---	6 (0, 1)
61	3D	---	---	12 (0, 0) 12 (0, 0)	189	BD	---	---	12 (0, 0) 18 (0, -1)
		---	---	13 (1, 0)			---	---	13 (1, 0)

DOT PAT FROM				DOT PAT FROM			
DEC	HEX	CUR	DOTS	DEC	HEX	CUR	DOTS
LOC			PLOTTED	LOC			PLOTTED
ENDING COORDINATES				ENDING COORDINATES			
62	3E	---	12 (0, 0)	190	BE	---	12 (0, 0)
		---	18 (0, -1)			---	18 (0, -1)
63	3F	---	12 (0, 0)	191	BF	---	12 (0, 0)
		---	11 (-1, 0)			---	11 (-1, 0)
64	40	---		192	C0	---	
		---	3 (1, 2)			---	1 (-1, 2)
65	41	---		193	C1	---	
		---	8 (2, 1)			---	6 (0, 1)
66	42	---		194	C2	---	
		---	13 (1, 0)			---	11 (-1, 0)
67	43	---		195	C3	---	
		---	6 (0, 1)			---	4 (-2, 1)
68	44	---	12 (0, 0)	196	C4	---	12 (0, 0)
		---	3 (1, 2)			---	1 (-1, 2)
69	45	---	12 (0, 0)	197	C5	---	12 (0, 0)
		---	8 (2, 1)			---	6 (0, 1)
70	46	---	12 (0, 0)	198	C6	---	12 (0, 0)
		---	13 (1, 0)			---	11 (-1, 0)
71	47	---	12 (0, 0)	199	C7	---	12 (0, 0)
		---	6 (0, 1)			---	4 (-2, 1)
72	48	---		200	C8	---	
		---	8 (2, 1)			---	6 (0, 1)
73	49	---		201	C9	---	
		---	15 (3, 0)			---	13 (1, 0)
74	4A	---		202	CA	---	
		---	20 (2, -1)			---	18 (0, -1)

DOT PAT FROM CUR				DOT PAT FROM CUR					
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
75	4B	---	---	13 (1, 0)	203	CB	---	---	11 (-1, 0)
76	4C	---	---		204	CC	---	---	
77	4D	---	---		205	CD	---	---	
78	4E	---	---		206	CE	---	---	
79	4F	---	---		207	CF	---	---	
80	50	---	---		208	D0	---	---	
81	51	---	---		209	D1	---	---	
82	52	---	---		210	D2	---	---	
83	53	---	---		211	D3	---	---	
84	54	---	---		212	D4	---	---	
85	55	---	---		213	D5	---	---	
86	56	---	---		214	D6	---	---	
87	57	---	---		215	D7	---	---	

DOT PAT FROM CUR				DOT PAT FROM CUR					
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
88	58	---		6 (0, 1)	216	D8	---		4 (-2, 1)
89	59	---		13 (1, 0)	217	D9	---		11 (-1, 0)
90	5A	---		18 (0, -1)	218	DA	---		16 (-2, -1)
91	5B	---		11 (-1, 0)	219	DB	---		9 (-3, 0)
92	5C	-+-	12 (0, 0)	6 (0, 1)	220	DC	-+-	12 (0, 0)	4 (-2, 1)
93	5D	-+-	12 (0, 0)	13 (1, 0)	221	DD	-+-	12 (0, 0)	11 (-1, 0)
94	5E	-+-	12 (0, 0)	18 (0, -1)	222	DE	-+-	12 (0, 0)	16 (-2, -1)
95	5F	-+-	12 (0, 0)	11 (-1, 0)	223	DF	-+-	12 (0, 0)	9 (-3, 0)
96	60	-+-	6 (0, 1)	3 (1, 2)	224	E0	-+-	6 (0, 1)	1 (-1, 2)
97	61	--+	13 (1, 0)	8 (2, 1)	225	E1	--+	13 (1, 0)	6 (0, 1)
98	62	-+-	18 (0, -1)	13 (1, 0)	226	E2	-+-	18 (0, -1)	11 (-1, 0)
99	63	+--	11 (-1, 0)	6 (0, 1)	227	E3	+--	11 (-1, 0)	4 (-2, 1)

DOT PAT FROM CUR					DOT PAT FROM CUR				
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
100	64	--- -+- ---	12 (0, 0) 6 (0, 1)	3 (1, 2)	228	E4	--- -+- ---	12 (0, 0) 6 (0, 1)	1 (-1, 2)
101	65	--- -++ ---	12 (0, 0) 13 (1, 0)	8 (2, 1)	229	E5	--- -++ ---	12 (0, 0) 13 (1, 0)	6 (0, 1)
102	66	--- -+- -+-	12 (0, 0) 18 (0, -1)	13 (1, 0)	230	E6	--- -+- -+-	12 (0, 0) 18 (0, -1)	11 (-1, 0)
103	67	--- ++- ---	12 (0, 0) 11 (-1, 0)	6 (0, 1)	231	E7	--- ++- ---	12 (0, 0) 11 (-1, 0)	4 (-2, 1)
104	68	--- -+- ---	6 (0, 1)	8 (2, 1)	232	E8	--- -+- ---	6 (0, 1)	6 (0, 1)
105	69	--- --+ ---	13 (1, 0)	15 (3, 0)	233	E9	--- --+ ---	13 (1, 0)	13 (1, 0)
106	6A	--- --- -+-	18 (0, -1)	20 (2, -1)	234	EA	--- --- -+-	18 (0, -1)	18 (0, -1)
107	6B	--- +-- ---	11 (-1, 0)	13 (1, 0)	235	EB	--- +-- ---	11 (-1, 0)	11 (-1, 0)
108	6C	--- -+- ---	12 (0, 0) 6 (0, 1)	8 (2, 1)	236	EC	--- -+- ---	12 (0, 0) 6 (0, 1)	6 (0, 1)
109	6D	--- -++ ---	12 (0, 0) 13 (1, 0)	15 (3, 0)	237	ED	--- -++ ---	12 (0, 0) 13 (1, 0)	13 (1, 0)
110	6E	--- -+- -+-	12 (0, 0) 18 (0, -1)	20 (2, -1)	238	EE	--- -+- -+-	12 (0, 0) 18 (0, -1)	18 (0, -1)
111	6F	--- ++- ---	12 (0, 0) 11 (-1, 0)	13 (1, 0)	239	EF	--- ++- ---	12 (0, 0) 11 (-1, 0)	11 (-1, 0)
112	70	--- -+- ---	6 (0, 1)	13 (1, 0)	240	F0	--- -+- ---	6 (0, 1)	11 (-1, 0)

DOT PAT FROM CUR					DOT PAT FROM CUR				
DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES	DEC	HEX	LOC	DOTS PLOTTED	ENDING COORDINATES
113	71	---	---	13 (1, 0) 20 (2, -1)	241	F1	---	---	13 (1, 0) 18 (0, -1)
		---	---				---	---	
114	72	---	---	18 (0, -1) 23 (1, -2)	242	F2	---	---	18 (0, -1) 21 (-1, -2)
		---	---				---	---	
115	73	---	---	11 (-1, 0) 18 (0, -1)	243	F3	---	---	11 (-1, 0) 16 (-2, -1)
		---	---				---	---	
116	74	---	---	12 (0, 0) 13 (1, 0)	244	F4	---	---	12 (0, 0) 11 (-1, 0)
		---	---	6 (0, 1)			---	---	6 (0, 1)
117	75	---	---	12 (0, 0) 20 (2, -1)	245	F5	---	---	12 (0, 0) 18 (0, -1)
		---	---	13 (1, 0)			---	---	13 (1, 0)
118	76	---	---	12 (0, 0) 23 (1, -2)	246	F6	---	---	12 (0, 0) 21 (-1, -2)
		---	---	18 (0, -1)			---	---	18 (0, -1)
119	77	---	---	12 (0, 0) 18 (0, -1)	247	F7	---	---	12 (0, 0) 16 (-2, -1)
		---	---	11 (-1, 0)			---	---	11 (-1, 0)
120	78	---	---	6 (0, 1) 6 (0, 1)	248	F8	---	---	6 (0, 1) 4 (-2, 1)
		---	---				---	---	
121	79	---	---	13 (1, 0) 13 (1, 0)	249	F9	---	---	13 (1, 0) 11 (-1, 0)
		---	---				---	---	
122	7A	---	---	18 (0, -1) 18 (0, -1)	250	FA	---	---	18 (0, -1) 16 (-2, -1)
		---	---				---	---	
123	7B	---	---	11 (-1, 0) 11 (-1, 0)	251	FB	---	---	11 (-1, 0) 9 (-3, 0)
		---	---				---	---	
124	7C	---	---	12 (0, 0) 6 (0, 1)	252	FC	---	---	12 (0, 0) 4 (-2, 1)
		---	---	6 (0, 1)			---	---	6 (0, 1)



ENCODE HIGH RESOLUTION SHAPES

This chart can be extremely useful for designing hires "shapes." It will take some getting used to, but once it is mastered, it will *substantially* aid in creating shape tables. The method for creating shapes with this chart is unique; so we will go through an example to illustrate how to employ it correctly. Like a previous shape chart in this book, this one uses an "Ending Position Location" table. The *current* location of a shape is always '12' (0,0) and all other positions are in relation to the current position. Each column heading shows a combination of nine locations with + and - signs. The center is where a new plot/move always begins, and it is always '12' on the Position Location table. The - (minus) marks represent nonplotted pixels, and the + represent plots. For example, the second column on the chart shows:

```
12 (0,0)
 6 (0,1)
+-
+-
--
```

It means that the first position '12', which is 0 horizontal and 0 vertical, was plotted, and then position '6', directly above 12 on the Position Location Table, which is 0 horizontal and 1 vertical, was also plotted.

The next step is to determine where the shape cursor position will end up. The "Ending Location Table" below is used for that. Beginning at location 12, decide where, relative to 12, you want to go. For example, suppose you wanted to plot position 12 and 18 and then end up at location 24.

```
+ 12
+ 18
- 22
- 24
```

That's going straight down. Rather than trying to calculate plots and moves, simply find two symbols indicating plots, one at location 12 and the second at location 18. It looks like this:

```
12 (0,0)
18 (0,-1)
--
+-
+-
```

Now look in the column "MOVE TO ENDING LOCATION" for an ending location of 24, the very bottom row of your chart. Now cross index the column and the row and you will find the values 182 and B6. The first number is the decimal value and the second, the hexadecimal value for your shape byte entry. Simply enter that value. By treating your shape creations in this manner, you will save a lot of time and give yourself complete control over their creation. The following example shows how to make a "right arrow" shape. First, draw the shape:


```

      *
    -> -> * *
  * * * * *
      * * <
Start   * End

```

To begin, we will plot position 12 and 13 and end on position 14. (The starting position is indicated by the carat ' '). First, we look up a plot of 12 and 13. It is in column 9. Then we cross index that column with an ending position of 14. Our value is \$2D. Now we are on the third point from the start. Since we want to go to the tip of the arrow, we proceed in the same manner again using \$2D. Now we are at the fifth point. Again we enter \$2D. After this plot/move, we are at the tip of the arrow. Now we want to plot the tip and move up and to the left. Since there is no configuration to plot/move/plot in that manner, we simply have to plot and move. Looking at our Ending Location Table, we want to end up in position 5. To do this we go to the column which only plots our starting point (*always* position '12') and then cross index it with the ending location of 5. There we find \$1C. We complete the arrow by the same series of examining the plots we want and ending position until we have the last position plotted. In hex, we get the following values to make our arrow:

\$2D \$2D \$2D \$1C \$1C \$36 \$32 \$0C \$0F

Now we enter that in memory. Using the monitor, we can enter it, starting at \$300 as follows:

```

   *1  *2  *3  *4  *5
300: 01 00 04 00 2D 2D 2D 1C

           *6
308: 1C 36 32 0C 0F 00

```

(NOTE: *1 one byte value for number of shapes. *2 any value (unused byte). *3 low byte and *4 high byte of the relative offset from the first byte in the index to the first shape. *5 beginning of shape. *6 end of shape.)

Now we set the shape table pointers at \$E8 and \$E9 to point to our starting location of \$300:

E8: 00 03

That done, the shape is all set to go. Using DRAW, ROT, SCALE and other graphics/shape commands it is possible to use the shape created.

Ending Location Table

		1 -1, 2	2 0, 2	3 1, 2		
	4 -2, 1	5 -1, 1	6 0, 1	7 1, 1	8 2, 1	
9 -3, 0	10 -2, 0	11 -1, 0	12 0, 0	13 1, 0	14 2, 0	15 3, 0
	16 -2, -1	17 -1, -1	18 0, -1	19 1, -1	20 2, -1	
		21 -1, -2	22 0, -2	23 1, -2		
			24 0, -3			

Ending Location Table (A)

		1	2	3		
	4	5	6	7	8	
9	10	11	12	13	14	15
	16	17	18	19	20	
		21	22	23		
			24			

Ending Location Table (B)

		-1, 2	0, 2	1, 2		
	-2, 1	-1, 1	0, 1	1, 1	2, 1	
-3, 0	-2, 0	-1, 0	0, 0	1, 0	2, 0	3, 0
	-2, -1	-1, -1	0, -1	1, -1	2, -1	
		-1, -2	0, -2	1, -2		
			0, -3			

MOVE TO ENDING LOCATION	6 (0, 1) -+- --- ---	11 (-1, 0) --- +-- ---	12 (0, 0) --- -+- ---	13 (1, 0) --- ---+ ---	18 (0, -1) --- --- --+-
PS# XYOFF	DEC HEX	DEC HEX	DEC DEX	DEC HEX	DEC HEX
1 (-1, 2)	224 E0	* *	196 C4	* *	* *
2 (0, 2)	32 20	* *	* *	* *	* *
3 (1, 2)	96 60	* *	68 44	* *	* *
4 (-2, 1)	248 F8	227 E3	199 C7 220 DC	* *	* *
5 (-1, 1)	56 38	35 23	28 1C	* *	* *
6 (0, 1)	120 78 160 A0 232 E8	99 63	4 4 71 47 92 5C 132 84 197 C5 204 CC	225 E1	* *
7 (1, 1)	40 28	* *	12 C	33 21	* *
8 (2, 1)	104 68	* *	69 45 76 4C	97 61	* *
9 (-3, 0)	* *	251 FB	223 DF	* *	* *
10 (-2, 0)	* *	59 3B	31 1F	* *	* *
11 (-1, 0)	184 B8 240 F0	123 7B 163 A3 235 EB	7 7 95 5F 135 87 156 9C 198 C6 207 CF 212 D4 221 DD	249 F9	226 E2
12 (0, 0)	48 30	43 2B	15 F 20 14 29 1D	57 39	34 22
13 (1, 0)	112 70 168 A8	107 6B	5 5 70 46 79 4F 84 54 93 5D 133 85 140 8C 205 CD	121 79 161 A1 233 E9	98 62

MOVE TO	NO PLOT	12 (0, 0) 6 (0, 1)	12 (0, 0) 11 (-1, 0)	12 (0, 0) 13 (1, 0)	12 (0, 0) 18 (0, -1)
ENDING LOCATION	---	--- --+- ---	--- ++- ---	--- -++ ---	--- --+- ---
PS# XYOFF	DEC HEX	DEC HEX	DEC HEX	DEC HEX	DEC HEX
1 (-1, 2)	192 C0	228 E4	* *	* *	* *
2 (0, 2)	* *	36 24	* *	* *	* *
3 (1, 2)	64 40	100 64	* *	* *	* *
4 (-2, 1)	195 C3 216 D8	252 FC	231 E7	* *	* *
5 (-1, 1)	24 18	60 3C	39 27	* *	* *
6 (0, 1)	67 43 88 58 128 80 193 C1 200 C8	124 7C 164 A4 236 EC	103 67	229 E5	* *
7 (1, 1)	8 8	44 2C	* *	37 25	* *
8 (2, 1)	65 41 72 48	108 6C	* *	101 65	* *
9 (-3, 0)	219 DB	* *	255 FF	* *	* *
10 (-2, 0)	27 1B	* *	63 3F	* *	* *
11 (-1, 0)	3 3 91 5B 131 83 152 98 194 C2 203 CB 208 D0 217 D9	188 BC 244 F4	127 7F 167 A7 239 EF	253 FD	230 E6
12 (0, 0)	11 B 16 10 25 19	52 34	47 2F	61 3D	38 26
13 (1, 0)	1 1 66 42 75 4B 80 50 89 59 129 81 136 88 201 C9	116 74 172 AC	111 6F	125 7D 165 A5 237 ED	102 66

MOVE TO ENDING LOCATION	6 (0, 1) -+- --- ---	11 (-1, 0) --- +-- ---	12 (0, 0) --- -+- ---	13 (1, 0) --- ---+ ---	18 (0, -1) --- --- -+-
PS# XYOFF	DEC HEX	DEC HEX	DEC DEX	DEC HEX	DEC HEX
14 (2, 0)	* *	* *	13 D	41 29	* *
15 (3, 0)	* *	* *	77 4D	105 69	* *
16 (-2, -1)	* *	187 BB 243 F3	159 9F 215 D7 222 DE	* *	250 FA
17 (-1, -1)	* *	51 33	23 17 30 1E	* *	58 3A
18 (0, -1)	176 B0	115 73 171 AB	6 6 87 57 94 5E 134 86 143 8F 148 94 157 9D 206 CE 213 D5	185 B9 241 F1	122 7A 162 A2 234 EA
19 (1, -1)	* *	* *	14 E 21 15	49 31	42 2A
20 (2, -1)	* *	* *	78 4E 85 55 141 8D	113 71 169 A9	106 6A
21 (-1, -2)	* *	179 B3	151 97 158 9E 214 D6	* *	186 BA 242 F2
22 (0, -2)	* *	* *	22 16	* *	50 32
23 (1, -2)	* *	* *	86 56 142 8E 149 95	177 B1	114 72 170 AA
24 (0, -3)	* *	* *	150 96	* *	178 B2

MOVE TO	NO PLOT	12 (0, 0) 6 (0, 1)	12 (0, 0) 11 (-1, 0)	12 (0, 0) 13 (1, 0)	12 (0, 0) 18 (0, -1)
ENDING LOCATION	---	-+-	---	---	---
PS# XYOFF	DEC HEX	DEC HEX	DEC HEX	DEC HEX	DEC HEX
14 (2, 0)	9 9	* *	* *	45 2D	* *
15 (3, 0)	73 49	* *	* *	109 6D	* *
16 (-2, -1)	155 9B 211 D3 218 DA	* *	191 BF 247 F7	* *	254 FE
17 (-1, -1)	19 13 26 1A	* *	55 37	* *	62 3E
18 (0, -1)	2 2 83 53 90 5A 130 82 139 8B 144 90 153 99 202 CA 209 D1	180 B4	119 77 175 AF	189 BD 245 F5	126 7E 166 A6 238 EE
19 (1, -1)	10 A 17 11	* *	* *	53 35	46 2E
20 (2, -1)	74 4A 81 51 137 89	* *	* *	117 75 173 AD	110 6E
21 (-1, -2)	147 93 154 9A 210 D2	* *	183 B7	* *	190 BE 246 F6
22 (0, -2)	18 12	* *	* *	* *	54 36
23 (1, -2)	82 52 138 8A 145 91	* *	* *	181 B5	118 76 174 AE
24 (0, -3)	146 92	* *	* *	* *	182 B6

ASCII CHARACTER SHAPES

The following table shows ASCII characters generated as shapes. Each character is shown as a set of plot points with S indicating the beginning of the plotting points and E, the end. A lower case "s" or "e" means that the starting or ending point was plotted.

To the right of each character diagram is the information for creating the character as a shape. The DATA statements are the decimal values to be entered in BASIC programs. Below the DATA values are the hexadecimal values for the shape. The colon (:) to the left of the hex numbers is a reminder that in the monitor, the values have to be entered after the colon prompt. Below the hexadecimal numbers is the table and shape number for each character. For example, the shape for the question mark "?" is located in Shape Table 1, Shape Number 32.

	S - - - - - E	DATA 73, 1, 73, 0
		: 49 1 49 0
		(SHAPE TABLE #, SHAPE #) -- (1, 1)
!	- - + - - - - - - + - - - - - - + - - - - - - + - - - - - - + - - - - - - + - - - - S - + - - - - E	DATA 9, 196, 33, 36, 180, 146, 74, 73, 0
		: 9 C4 21 24 B4 92 4A 49 0
		(SHAPE TABLE #, SHAPE #) -- (1, 2)
"	+ + - + + - - - + + - + + - - - + + - + + - - - - - - - - - - - - - - - - - - S - - - - - E	DATA 64, 192, 229, 229, 109, 53, 119, 119, 146, 73, 0
		: 40 C0 E5 E5 6D 35 77 77 92 49 0
		(SHAPE TABLE #, SHAPE #) -- (1, 3)
#	- + - + - - - - - + - + - - - - + + + + - - - - - + - + - - - - + + + + - - - - - + - + - - - - S + - + - - - E	DATA 33, 60, 12, 60, 12, 108, 190, 45, 23, 23, 45, 23, 46, 73, 0
		: 21 3C C 3C C 6C BE 2D 17 17 2D 17 2E 49 0
		(SHAPE TABLE #, SHAPE #) -- (1, 4)
\$	- - + - - - - - - + + + - - - - + - + - - - - - - + + + - - - - - - + - + - - - + + + + - - - - S - + - - - - E	DATA 40, 45, 12, 28, 63, 28, 12, 45, 253, 176, 22, 22, 13, 73, 0
		: 28 2D C 1C 3F 1C C 2D FD B0 16 16 D 49 0
		(SHAPE TABLE #, SHAPE #) -- (1, 5)

% ++ -- -- --
 ++ -- + --
 -- + -- --
 -- + -- --
 - + -- --
 + -- + + --
 S -- + + -- E

DATA 96, 12, 12, 12, 199, 27, 189, 109, 146, 2, 102, 126, 73, 0

: 60 C C C C7 1B BD 6D 92 2 66 7E 49 0

(SHAPE TABLE #, SHAPE #) -- (1, 6)

& - + -- -- --
 + - + -- --
 + - + -- --
 - + -- -- --
 + + + - + --
 + - - + -- --
 S + + - + -- E

DATA 67, 44, 228, 100, 14, 182, 13, 23, 159, 109, 77, 0

: 43 2C E4 64 E B6 D 17 9F 6D 4D 0

(SHAPE TABLE #, SHAPE #) -- (1, 7)

' - + + -- -- --
 - + + -- -- --
 - + + -- -- --
 -- -- -- -- --
 -- -- -- -- --
 -- -- -- -- --
 S -- -- -- -- E

DATA 64, 64, 36, 55, 182, 73, 18, 73, 0

: 40 40 24 37 B6 49 12 49 0

(SHAPE TABLE #, SHAPE #) -- (1, 8)

(-- - + -- -- --
 -- - + -- -- --
 -- - + -- -- --
 -- - + -- -- --
 -- - + -- -- --
 -- - + -- -- --
 S -- - + -- -- E

DATA 65, 36, 36, 12, 150, 146, 5, 73, 0

: 41 24 24 C 96 92 5 49 0

(SHAPE TABLE #, SHAPE #) -- (1, 9)

) - + -- -- -- --
 -- - + -- -- --
 -- - + -- -- --
 -- - + -- -- --
 -- - + -- -- --
 -- - + -- -- --
 S + -- -- -- -- E

DATA 97, 36, 36, 28, 150, 146, 73, 73, 0

: 61 24 24 1C 96 92 49 49 0

(SHAPE TABLE #, SHAPE #) -- (1, 10)

* -- - + -- -- --
 + - + - + -- --
 - + + + -- -- --
 + + + + + -- --
 - + + + -- -- --
 + - + - + -- --
 S - + -- -- -- E

DATA 96, 37, 63, 69, 231, 69, 110, 23, 46, 23, 151, 108, 6, 73, 0

: 60 25 3F 45 E7 45 6E 17 2E 17 97 6C 6 49 0

(SHAPE TABLE #, SHAPE #) -- (1, 11)

+	- - + - - - - - - + - - - - + + + + - - - - - + - - - - - - + - - - - S - - - - - E	DATA 65, 228, 111, 36, 86, 181, 2, 73, 0 : 41 E4 6F 24 56 B5 2 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 12)
'	- + + - - - - - + + - - - - - - + - - - - S + - - - - E	DATA 97, 196, 189, 109, 18, 73, 0 : 61 C4 BD 6D 12 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 13)
-	+ + + + - - - - - - - - - - - - - S - - - - - E	DATA 64, 24, 45, 45, 150, 73, 0 : 40 18 2D 2D 96 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 14)
.	- + + - - - - S + + - - - E	DATA 8, 189, 109, 73, 0 : 8 BD 6D 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 15)
/	- - - - + - - - - - - + - - - - - + - - - - - + - - - - - + - - - - - - S - - - - - E	DATA 96, 12, 12, 12, 150, 18, 73, 0 : 60 C C C 96 12 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 16)
0	- + + + - - - + - - - + - - - + - - + + - - - + - + - + - - - + + - - + - - - + - - - + - - - S + + + - - - E	DATA 96, 12, 12, 37, 28, 63, 23, 54, 182, 41, 101, 36, 150, 73, 0 : 60 C C 25 1C 3F 17 36 B6 29 65 24 96 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 17) (2, 1) (3, 1)
1	- - + - - - - - + + - - - - - - + - - - - - - + - - - - - - + - - - - - - + - - - - S + + + - - - E	DATA 41, 36, 36, 36, 23, 150, 82, 41, 73, 0 : 29 24 24 24 17 96 52 29 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 18) (2, 2) (3, 2)
2	- + + + - - - + - - - + - - - - - - - + - - - - - + + - - - - - + - - - - - + - - - - - - S + + + + - - - E	DATA 100, 12, 101, 228, 63, 23, 150, 82, 45, 45, 9, 0 64 C 65 E4 3F 17 96 52 2D 2D 9 0 (SHAPE TABLE #, SHAPE #) -- (1, 19) (2, 3) (3, 3)

3 - + + + - - - - DATA 41,101,228,103,1,228,63,23,150,114,73,73,0
 + - - - + - - -
 - - - + - - -
 - - + + - - - -
 - - - + - - -
 + - - - + - - -
 S + + + - - - E
 : 29 65 E4 67 1 E4 3F 17 96 72 49 49 0
 (SHAPE TABLE #, SHAPE #) -- (1,20) (2,4) (3,4)

4 - - - + - - - - DATA 73,36,36,36,23,23,23,46,109,22,73,0
 - - + + - - - -
 - + - + - - - -
 + - - + - - - -
 + + + + + - - -
 - - - + - - - -
 S - - + - - - E
 : 49 24 24 24 17 17 17 2E 6D 16 49 0
 (SHAPE TABLE #, SHAPE #) -- (1,21) (2,5) (3,5)

5 + + + + + - - - DATA 112,45,12,36,28,63,39,44,45,181,146,2,73,0
 + - - - - - - -
 + + + + - - - -
 - - - - + - - -
 - - - + - - -
 + - - - + - - -
 S + + + - - - E
 : 70 2D C 24 1C 3F 27 2C 2D B5 92 2 49 0
 (SHAPE TABLE #, SHAPE #) -- (1,22) (2,6) (3,6)

6 - - + + - - - - DATA 225,36,229,12,12,181,19,117,190,111,73,0
 - + - - - - - -
 + - - - - - - -
 + + + + - - - -
 + - - - + - - -
 + - - - + - - -
 S + + + - - - E
 : E1 24 E5 C C B5 13 75 BE 6F 49 0
 (SHAPE TABLE #, SHAPE #) -- (1,23) (2,7) (3,7)

7 + + + + + - - - DATA 33,100,12,12,60,255,109,146,146,1,73,0
 - - - - + - - -
 - - - + - - - -
 - - + - - - - -
 - + - - - - - -
 - + - - - - - -
 S + - - - - - E
 : 21 64 C C 3C FF 6D 92 92 1 49 0
 (SHAPE TABLE #, SHAPE #) -- (1,24) (2,8) (3,8)

8 - + + + - - - - DATA 225,100,28,100,45,14,190,119,1,190,111,73,0
 + - - - + - - -
 + - - - + - - -
 - + + + - - - -
 + - - - + - - -
 + - - - + - - -
 S + + + - - - E
 : E1 64 1C 64 2D E BE 77 1 BE 6F 49 0
 (SHAPE TABLE #, SHAPE #) -- (1,25) (2,9) (3,9)

9 - + + + - - - - DATA 1, 101, 12, 36, 228, 63, 23, 118, 45, 86, 2, 73, 0
+ - - - + - - -
+ - - - + - - - : 1 65 C 24 E4 3F 17 76 2D 56 2 49 0
- + + + - - - -
- - - - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 26) (2, 10) (3, 10)
- - - + - - - -
S + + - - - - E

: - + + - - - - - DATA 8, 229, 229, 67, 229, 109, 146, 18, 73, 0
- + + - - - - -
- - - - - - - - : 8 E5 E5 43 E5 6D 92 12 49 0
- + + - - - - -
- + + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 27)
S - - - - - E

; - + + - - - - - DATA 37, 229, 229, 67, 229, 109, 146, 18, 73, 0
- + + - - - - -
- - - - - - - - : 25 E5 E5 43 E5 6D 92 12 49 0
- + + - - - - -
- + + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 28)
S + - - - - E

< - - - + + - - - DATA 65, 28, 28, 12, 12, 12, 181, 146, 2, 47, 73, 0
- - + - - - - -
- + - - - - - - : 41 1C 1C C C C B5 92 2 2F 49 0
+ - - - - - - -
- + - - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 29)
- - + - - - - -
S - - + + - - E

= + + + + - - - - DATA 192, 41, 109, 255, 192, 45, 45, 150, 2, 73, 0
- - - - - - - -
+ + + + - - - - : C0 29 6D FF C0 2D 2D 96 2 49 0
- - - - - - - -
S - - - - - E (SHAPE TABLE #, SHAPE #) -- (1, 30)

> + + - - - - - - DATA 101, 12, 12, 28, 28, 199, 109, 146, 146, 1, 73, 0
- - + - - - - -
- - - + - - - - : 65 C C 1C 1C C7 6D 92 92 1 49 0
- - - - + - - -
- - - + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 31)
- - + - - - - -
S + - - - - E

? - + + + - - - - DATA 64, 97, 12, 228, 63, 23, 150, 82, 105, 73, 0
+ - - - + - - -
- - - - + - - - : 40 61 C E4 3F 17 96 52 69 49 0
- - - + - - - -
- - + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 32)
- - - - - - - -
S - + - - - E

@ - + + + - - - - DATA 225, 36, 36, 12, 45, 14, 62, 23, 46, 12, 150, 27, 45, 73, 0
 + - - - + - - -
 + - - + + - - - : E1 24 24 C 2D E 3E 17 2E C 96 1B 2D 49 0
 + - + - + - - -
 + - + + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 33)
 + - - - - - - -
 S + + + - - - E

A - - + - - - - - DATA 36, 36, 12, 12, 14, 14, 254, 19, 45, 53, 126, 73, 0
 - + - + - - - -
 + - - - + - - - : 24 24 C C E E FE 13 2D 35 7E 49 0
 + - - - + - - -
 + + + + + - - - (SHAPE TABLE #, SHAPE #) -- (1, 34) (2, 11) (3, 11)
 + - - - + - - -
 S - - - + - - E

B + + + + - - - - DATA 36, 36, 36, 45, 173, 190, 63, 77, 2, 190, 63, 77, 73, 0
 + - - - + - - -
 + - - - + - - - : 24 24 24 2D AD BE 3F 4D 2 BE 3F 4D 49 0
 + + + + - - - -
 + - - - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 35) (2, 12) (3, 12)
 + - - - + - - -
 S + + + - - - E

C - + + + - - - - DATA 67, 36, 36, 12, 45, 14, 150, 250, 19, 45, 77, 1, 0
 + - - - + - - -
 + - - - - - - - : 43 24 24 C 2D E 96 FA 13 2D 4D 1 0
 + - - - - - - -
 + - - - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 36) (2, 13) (3, 13)
 + - - - + - - -
 S + + + - - - E

D + + + - - - - - DATA 36, 36, 36, 45, 14, 14, 54, 23, 23, 111, 73, 1, 0
 + - - + - - - -
 + - - - + - - - : 24 24 24 2D E E 36 17 17 6F 49 1 0
 + - - - + - - -
 + - - - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 37) (2, 14) (3, 14)
 + - - + - - - -
 S + + - - - - E

E + + + + + - - - DATA 36, 36, 36, 45, 45, 150, 59, 183, 42, 45, 77, 0
 + - - - - - - -
 + - - - - - - - : 24 24 24 2D 2D 96 3B B7 2A 2D 4D 0
 + + + + - - - -
 + - - - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 38) (2, 15) (3, 15)
 + - - - - - - -
 S + + + + - - E

F + + + + + - - - DATA 36, 36, 36, 45, 45, 150, 251, 109, 146, 73, 0
 + - - - - - - -
 + - - - - - - - : 24 24 24 2D 2D 96 FB 6D 92 49 0
 + + + + - - - -
 + - - - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 39) (2, 16) (3, 16)
 + - - - - - - -
 S - - - - - - E

G - + + + - - - DATA 32, 36, 100, 45, 14, 150, 47, 250, 19, 45, 77, 1, 0
 + - - - + - -
 + - - - - - - : 20 24 64 2D E 96 2F FA 13 2D 4D 1 0
 + - - - - - -
 + - - + + - - (SHAPE TABLE #, SHAPE #) -- (1, 40) (2, 17) (3, 17)
 + - - - + - -
 S + + + - - - E

H + - - - + - - DATA 36, 36, 36, 150, 41, 45, 36, 180, 18, 54, 77, 0
 + - - - + - -
 + - - - + - - : 24 24 24 96 29 2D 24 B4 12 36 4D 0
 + + + + + - -
 + - - - + - - (SHAPE TABLE #, SHAPE #) -- (1, 41) (2, 18) (3, 18)
 + - - - + - -
 S - - - + - - E

I - + + + - - - DATA 41, 36, 36, 228, 45, 150, 146, 5, 73, 0
 - - + - - - -
 - - + - - - - : 29 24 24 E4 2D 96 92 5 49 0
 - - + - - - -
 - - + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 42) (2, 19) (3, 19)
 - - + - - - -
 S + + + - - - E

J - - - - + - - DATA 112, 45, 12, 36, 36, 180, 146, 2, 73, 0
 - - - - + - -
 - - - - + - - : 70 2D C 24 24 B4 92 2 49 0
 - - - - + - -
 - - - - + - - (SHAPE TABLE #, SHAPE #) -- (1, 43) (2, 20) (3, 20)
 + - - - + - -
 S + + + - - - E

K + - - - + - - DATA 36, 36, 36, 77, 185, 23, 23, 14, 14, 14, 77, 0
 + - - + - - -
 + - + - - - - : 24 24 24 4D B9 17 17 E E E 4D 0
 + + - - - - -
 + - + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 44) (2, 21) (3, 21)
 + - - + - - -
 S - - - + - - E

L + - - - - - - DATA 36, 36, 36, 150, 146, 41, 45, 77, 0
 + - - - - - -
 + - - - - - - : 24 24 24 96 92 29 2D 4D 0
 + - - - - - -
 + - - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 45) (2, 22) (3, 22)
 + - - - - - -
 S + + + + - - E

M + - - - + - - - DATA 36,36,36,14,14,12,12,54,54,54,77,Ø
 + + - + + - - -
 + - + - + - - - : 24 24 24 E E C C 36 36 36 4D Ø
 + - - - + - - -
 + - - - + - - - (SHAPE TABLE #, SHAPE #) -- (1,46) (2,23) (3,23)
 + - - - + - - -
 S - - - + - - E

N + - - - + - - - DATA 36,36,36,14,46,54,53,37,36,36,15Ø,146,77,Ø
 + + - - + - - -
 + + + - + - - - : 24 24 24 E 2E 36 35 25 24 24 96 92 4D Ø
 + - + - + - - -
 + - + + + - - - (SHAPE TABLE #, SHAPE #) -- (1,47) (2,24) (3,24)
 + - - + + - - -
 S - - - + - - E

O - + + + - - - - DATA 225,36,36,12,45,14,54,54,159,45,73,Ø
 + - - - + - - -
 + - - - + - - - : E1 24 24 C 2D E 36 36 9F 2D 49 Ø
 + - - - + - - -
 + - - - + - - - (SHAPE TABLE #, SHAPE #) -- (1,48) (2,25) (3,25)
 + - - - + - - -
 S + + + - - - E

P + + + + - - - - DATA 36,36,36,45,117,19Ø,63,15Ø,73,73,Ø
 + - - - + - - -
 + - - - + - - - : 24 24 24 2D 75 BE 3F 96 49 49 Ø
 + + + + - - - -
 + - - - - - - - (SHAPE TABLE #, SHAPE #) -- (1,49) (2,26) (3,26)
 + - - - - - - -
 S - - - - - - E

Q - + + + - - - - DATA 225,36,36,12,45,14,54,254,14,23,13,77,Ø
 + - - - + - - -
 + - - - + - - - : E1 24 24 C 2D E 36 FE E 17 D 4D Ø
 + - - - + - - -
 + - + - + - - - (SHAPE TABLE #, SHAPE #) -- (1,5Ø) (2,27) (3,27)
 + - - + - - - -
 S + + - + - - E

R + + + + - - - - DATA 36,36,36,45,117,19Ø,63,14,14,14,77,Ø
 + - - - + - - -
 + - - - + - - - : 24 24 24 2D 75 BE 3F E E E 4D Ø
 + + + + - - - -
 + - + - - - - - (SHAPE TABLE #, SHAPE #) -- (1,51) (2,28) (3,28)
 + - - + - - - -
 S - - - + - - E

S - + + + - - - - DATA 112, 45, 12, 228, 63, 28, 100, 45, 14, 150, 18, 73, 0
 + - - - + - - -
 + - - - - - - - : 70 2D C E4 3F 1C 64 2D E 96 12 49 0
 - + + + - - - -
 - - - - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 52) (2, 29) (3, 29)
 + - - - + - - -
 S + + + - - - E

T + + + + - - - - DATA 9, 36, 36, 228, 111, 45, 150, 146, 73, 0
 - - + - - - - -
 - - + - - - - - : 9 24 24 E4 6F 2D 96 92 49 0
 - - + - - - - -
 - - + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 53) (2, 30) (3, 30)
 - - + - - - - -
 S - + - - - - E

U + - - - + - - - DATA 225, 36, 36, 108, 9, 54, 54, 190, 111, 73, 0
 + - - - + - - -
 + - - - + - - - : E1 24 24 6C 9 36 36 BE 6F 49 0
 + - - - + - - -
 + - - - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 54) (2, 31) (3, 31)
 + - - - + - - -
 S + + + - - - E

V + - - - + - - - DATA 8, 60, 36, 36, 77, 49, 54, 62, 190, 13, 73, 0
 + - - - + - - -
 + - - - + - - - : 8 3C 24 24 4D 31 36 3E BE D 49 0
 + - - - + - - -
 + + - + + - - - (SHAPE TABLE #, SHAPE #) -- (1, 55) (2, 32) (3, 32)
 - + - + - - - -
 S - + - - - - E

W + - - - + - - - DATA 36, 36, 36, 77, 49, 54, 54, 255, 12, 86, 1, 77, 0
 + - - - + - - -
 + - - - + - - - : 24 24 24 4D 31 36 36 FF C 56 1 4D 0
 + - - - + - - -
 + - + - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 56) (2, 33) (3, 33)
 + + - + + - - -
 S - - - + - - E

X + - - - + - - - DATA 100, 12, 12, 12, 252, 27, 118, 86, 113, 126, 73, 0
 + - - - + - - -
 - + - + - - - - : 64 C C C FC 1B 76 56 71 7E 49 0
 - - + - - - - -
 - + - + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 57) (2, 34) (3, 34)
 + - - - + - - -
 S - - - + - - E

Y	+ - - - + - - - + - - - + - - - - + - + - - - - - - + - - - - - - - + - - - - - - - + - - - - - S - + - - - - E	DATA 9, 36, 228, 28, 108, 9, 190, 150, 10, 73, 0 : 9 24 E4 1C 6C 9 BE 96 A 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 58) (2, 35) (3, 35)
Z	+ + + + + - - - - - - - + - - - - - - + - - - - - - + - - - - - - + - - - - - - + - - - - - - - S + + + + - - E	DATA 100, 12, 12, 12, 60, 255, 181, 146, 42, 45, 77, 0 : 64 C C C 3C FF B5 92 2A 2D 4D 0 (SHAPE TABLE #, SHAPE #) -- (1, 59) (2, 36) (3, 36)
[	- + + + + - - - - + - - - - - - - + - - - - - - - + - - - - - - - + - - - - - - - + - - - - - - S + + + + - - E	DATA 1, 229, 36, 36, 44, 45, 150, 146, 47, 73, 0 : 1 E5 24 24 2C 2D 96 92 2F 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 60)
/	+ - - - - - - - - + - - - - - - - - + - - - - - - - - + - - - - - - - - + - - - S - - - - - - E	DATA 73, 8, 28, 28, 28, 28, 150, 82, 73, 73, 0 : 49 8 1C 1C 1C 1C 96 52 49 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 61)
]	+ + + + - - - - - - - + - - - - - - - + - - - - - - - + - - - - - - - + - - - - - - - + - - - - S + + + - - - E	DATA 45, 37, 36, 36, 60, 63, 150, 146, 1, 73, 73, 0 : 2D 25 24 24 3C 3F 96 92 1 49 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 62)
↑	- - + - - - - - - + + + - - - - + - + - + - - - - - + - - - - - - - + - - - - - - - + - - - - - S - + - - - - E	DATA 9, 36, 36, 252, 98, 12, 14, 14, 150, 2, 73, 0 : 9 24 24 FC 62 C E E 96 2 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 63)
—	S - - - - - E - - - - - - - + + + + + - - -	DATA 147, 45, 45, 45, 196, 73, 0 : 93 2D 2D 2D C4 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 64)

\ - + - - - - - DATA 64, 192, 64, 14, 14, 14, 146, 73, 0
 - - + - - - -
 - - - + - - - : 40 C0 40 E E E 92 49 0
 - - - - - - -
 - - - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 65)
 - - - - - - -
 S - - - - - E

a + + + - - - - DATA 32, 44, 45, 228, 63, 86, 18, 101, 14, 6, 73, 0
 - - - + - - -
 + + + + - - - : 20 2C 2D E4 3F 56 12 65 E 6 49 0
 + - - + - - -
 + + + - + - - (SHAPE TABLE #, SHAPE #) -- (1, 66) (3, 37)
 S - - - - - E

b - + - - - - - DATA 40, 36, 36, 180, 1, 117, 190, 119, 1, 73, 0
 - + - - - - -
 - + + + - - - : 28 24 24 B4 1 75 BE 77 1 49 0
 - + - - + - -
 - + - - + - - (SHAPE TABLE #, SHAPE #) -- (1, 67) (3, 38)
 + + + + - - -
 S - - - - - E

c - + + + - - - DATA 64, 35, 100, 45, 14, 150, 255, 109, 2, 73, 0
 + - - - + - -
 + - - - - - -
 + - - - - - - : 40 23 64 2D E 96 FF 6D 2 49 0
 - + + + + - -
 S - - - - - E (SHAPE TABLE #, SHAPE #) -- (1, 68) (3, 39)

d - - - + - - - DATA 65, 231, 100, 101, 180, 54, 46, 6, 73, 0
 - - - + - - -
 - + + + - - - : 41 E7 64 65 B4 36 2E 6 49 0
 + - - + - - -
 + - - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 69) (3, 40)
 - + + + + - -
 S - - - - - E

e - + + + - - - DATA 8, 28, 36, 12, 5, 117, 62, 63, 86, 117, 73, 0
 + - - - + - -
 + + + + + - - : 8 1C 24 C 5 75 3E 3F 56 75 49 0
 + - - - - - -
 - + + + - - - (SHAPE TABLE #, SHAPE #) -- (1, 70) (3, 41)
 S - - - - - E

f - - + - - - - DATA 8, 36, 111, 28, 100, 14, 150, 82, 73, 0
 - + - + - - -
 - + - - - - -
 + + + - - - - : 8 24 6F 1C 64 E 96 52 49 0
 - + - - - - -
 - + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 71) (3, 42)
 S - - - - - E

g - + + - - - - - DATA 1, 101, 60, 231, 100, 117, 182, 10, 73, 0
 + - - + - - - -
 + - - + - - - - : 1 65 3C E7 64 75 B6 A 49 0
 - + + + - - - -
 - - - + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 72) (3, 43)
 S + + - - - - E

h + - - - - - - - DATA 67, 36, 36, 180, 1, 117, 54, 14, 73, 0
 + - - - - - - -
 + + + - - - - - : 43 24 24 B4 1 75 36 E 49 0
 + - - + - - - -
 + - - + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 73) (3, 44)
 + - - + - - - -
 S - - - - - - E

i - - + - - - - - DATA 8, 45, 28, 36, 103, 176, 146, 74, 73, 0
 - - - - - - - -
 - + + - - - - - : 8 2D 1C 24 67 B0 92 4A 49 0
 - - + - - - - -
 - - + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 74) (3, 45)
 - + + + - - - -
 S - - - - - - E

j - - - + - - - - DATA 1, 225, 13, 36, 60, 68, 150, 146, 1, 73, 0
 - - + + - - - -
 - - - + - - - - : 1 E1 D 24 3C 44 96 92 1 49 0
 - - - + - - - -
 - + - + - - - - (SHAPE TABLE #, SHAPE #) -- (1, 75) (3, 46)
 S - + - - - - E

k + - - - - - - - DATA 32, 12, 39, 36, 77, 186, 22, 14, 14, 73, 0
 + - - + - - - -
 + - + - - - - - : 20 C 27 24 4D BA 16 E E 49 0
 + + - - - - - -
 + - + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 76) (3, 47)
 + - - + - - - -
 S - - - - - - E

l - + + - - - - - DATA 8, 45, 28, 36, 36, 111, 146, 146, 1, 73, 0
 - - + - - - - -
 - - + - - - - - : 8 2D 1C 24 24 6F 92 92 1 49 0
 - - + - - - - -
 - - + - - - - - (SHAPE TABLE #, SHAPE #) -- (1, 77) (3, 48)
 - + + + - - - -
 S - - - - - - E

m + + - + - - - - DATA 32, 36, 44, 14, 54, 110, 36, 228, 150, 82, 73, 0
 + - + - + - - -
 + - + - + - - - : 20 24 2C E 36 6E 24 E4 96 52 49 0
 + - + - + - - -
 + - + - + - - - (SHAPE TABLE #, SHAPE #) -- (1, 78) (3, 49)
 S - - - - - - E

n	+++----- +--+- +--+- +--+- +--+- S-----E	DATA 32,36,44,117,54,118,73,0 :20 24 2C 75 36 76 49 0 (SHAPE TABLE #, SHAPE #) -- (1,79) (3,50)
o	-++++----- +----- +----- +----- -++++----- S-----E	DATA 65,253,28,4,100,5,117,6,182,73,0 :41 FD 1C 4 64 5 75 6 B6 49 0 (SHAPE TABLE #, SHAPE #) -- (1,80) (3,51)
p	+++----- +--+- +--+- +++----- +----- S-----E	DATA 36,36,44,117,190,111,82,73,0 :24 24 2C 75 BE 6F 52 49 0 (SHAPE TABLE #, SHAPE #) -- (1,81) (3,52)
q	-++++----- +--+- +--+- -++++----- -++----- S--++-----E	DATA 64,229,3,100,45,54,54,46,77,0 :40 E5 3 64 2D 36 36 2E 4D 0 (SHAPE TABLE #, SHAPE #) -- (1,82) (3,53)
r	+--+----- ++----- +----- +----- +----- S-----E	DATA 67,36,36,14,12,117,146,2,73,0 :43 24 24 E C 75 92 2 49 0 (SHAPE TABLE #, SHAPE #) -- (1,83) (3,54)
s	-++++----- +----- -++++----- -----+----- ++++----- S-----E	DATA 67,45,101,28,7,231,12,45,181,146,73,0 :43 2D 65 1C 7 E7 C 2D B5 92 49 0 (SHAPE TABLE #, SHAPE #) -- (1,84) (3,55)
t	--+- --+- -++----- --+- --+- --+- S--+-E	DATA 73,28,36,44,31,12,180,146,74,73,0 :49 1C 24 2C 1F C B4 92 4A 49 0 (SHAPE TABLE #, SHAPE #) -- (1,85) (3,56)

u	+ - - + - - - - + - - + - - - - + - - + - - - - + - - + - - - - - + + + - - - - S - - - - - E	DATA 65, 231, 36, 108, 49, 54, 46, 6, 73, 0 : 41 E7 24 6C 31 36 2E 6 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 86) (3, 57)
v	+ - - - + - - - + - - - + - - - - + - + - - - - - + - + - - - - - - + - - - - - S - - - - - E	DATA 65, 28, 228, 108, 9, 190, 190, 74, 73, 0 : 41 1C E4 6C 9 BE BE 4A 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 87) (3, 58)
w	+ - - - + - - - + - - - + - - - + - + - + - - - + - + - + - - - - + - + - - - - S - - - - - E	DATA 8, 28, 36, 108, 9, 182, 252, 118, 14, 73, 0 : 8 1C 24 6C 9 B6 FC 76 E 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 88) (3, 59)
x	+ - - - + - - - - + - + - - - - - - + - - - - - - + - + - - - - + - - - + - - - S - - - - - E	DATA 96, 12, 12, 12, 223, 115, 86, 113, 6, 73, 0 : 60 C C C DF 73 56 71 6 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 89) (3, 60)
y	+ - - + - - - - + - - + - - - - + - - + - - - - - + + + - - - - - - - + - - - - S + + - - - E	DATA 1, 101, 36, 36, 223, 54, 14, 181, 9, 73, 0 : 1 65 24 24 DF 36 E B5 9 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 90) (3, 61)
z	+ + + + + - - - - - - + - - - - - - + - - - - - - + - - - - - - + + + + + - - - S - - - - - E	DATA 96, 12, 12, 12, 63, 63, 150, 10, 45, 53, 73, 0 : 60 C C C 3F 3F 96 A 2D 35 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 91) (3, 62)
{	- - - + + - - - - - + - - - - - - - + - - - - - - + - - - - - - - - + - - - - - - - + - - - - - S - - + + - - E	DATA 65, 228, 12, 100, 181, 146, 58, 5, 73, 0 : 41 E4 C 64 B5 92 3A 5 49 0 (SHAPE TABLE #, SHAPE #) -- (1, 92)


```

|  --+-----
   --+-----
   --+-----
   --+-----
   --+-----
   --+-----
   S -+----- E
DATA 9,36,36,36,150,146,9,73,0
:9 24 24 24 96 92 9 49 0
(SHAPE TABLE #, SHAPE #) -- (1,93)

```

```

}  ++-----
   --+-----
   --+-----
   --+-----
   --+-----
   --+-----
   S +----- E
DATA 101,100,28,228,111,146,146,9,73,0
:65 64 1C E4 6F 92 92 9 49 0
(SHAPE TABLE #, SHAPE #) -- (1,94)

```

```

~  -----+-----
   -++++-----
   +-----
   -----
   S ----- E
DATA 64,99,45,12,150,2,73,0
:40 63 2D C 96 2 49 0
(SHAPE TABLE #, SHAPE #) -- (1,95)

```

ASCII CHARACTER SHAPE TABLES CODE

The following memory dumps will create shape tables of all the ASCII codes. The first table generates shapes of all ASCII code, including symbols. The second table generates upper case and numbers only, and the third table generates upper and lower case along with the numbers but without symbols. These three dumps are presented to make it convenient to use the ASCII character set, or portions of it, in your programs. All code is based on the shape tables referenced in the ASCII SHAPE CHART.

ASCII SHAPE TABLE CODE #1 (Entire ASCII set)

ADDR.	MACHINE CODE
6000-	5F 00 C0 00 C4 00 CD 00
6008-	D8 00 E7 00 F6 00 04 01
6010-	10 01 19 01 22 01 2B 01
6018-	3A 01 43 01 4A 01 51 01
6020-	56 01 5E 01 6D 01 77 01
6028-	83 01 90 01 9C 01 AA 01
6030-	B6 01 C2 01 CF 01 DC 01
6038-	E6 01 F0 01 FC 01 07 02
6040-	13 02 1E 02 2D 02 3A 02
6048-	48 02 55 02 62 02 6E 02
6050-	79 02 86 02 92 02 9C 02
6058-	A6 02 B2 02 BB 02 C7 02
6060-	D5 02 E1 02 EC 02 F9 02
6068-	05 03 12 03 1C 03 27 03
6070-	33 03 40 03 4C 03 57 03
6078-	63 03 6E 03 79 03 85 03
6080-	91 03 98 03 A1 03 AD 03
6088-	B8 03 C3 03 CD 03 D9 03
6090-	E3 03 ED 03 F7 03 01 04
6098-	0C 04 17 04 22 04 2E 04
60A0-	36 04 41 04 4A 04 54 04
60A8-	5E 04 6A 04 75 04 7F 04
60B0-	89 04 94 04 9F 04 AA 04
60B8-	B6 04 C0 04 C9 04 D3 04
60C0-	49 01 49 00 09 C4 21 24
60C8-	B4 92 4A 49 00 40 C0 E5
60D0-	E5 6D 35 77 77 92 49 00
60D8-	21 3C 0C 3C 0C 6C BE 2D
60E0-	17 17 2D 17 2E 49 00 28
60E8-	2D 0C 1C 3F 1C 0C 2D FD
60F0-	B0 16 16 0D 49 00 60 0C
60F8-	0C 0C C7 1B BD 6D 92 02
6100-	66 7E 49 00 43 2C E4 64
6108-	0E B6 0D 17 9F 6D 4D 00
6110-	40 40 24 37 B6 49 12 49
6118-	00 41 24 24 0C 96 92 05
6120-	49 00 61 24 24 1C 96 92
6128-	49 49 00 60 25 3F 45 E7
6130-	45 6E 17 2E 17 97 6C 06
6138-	49 00 41 E4 6F 24 56 B5
6140-	02 49 00 61 C4 BD 6D 12
6148-	49 00 40 18 2D 2D 96 49

ADDR.	MACHINE CODE
6150-	00 08 BD 6D 49 00 60 0C
6158-	0C 0C 96 12 49 00 60 0C
6160-	0C 25 1C 3F 17 36 B6 29
6168-	65 24 96 49 00 29 24 24
6170-	24 17 96 52 29 49 00 64
6178-	0C 65 E4 3F 17 96 52 2D
6180-	2D 09 00 29 65 E4 67 01
6188-	E4 3F 17 96 72 49 49 00
6190-	49 24 24 24 17 17 17 2E
6198-	6D 16 49 00 70 2D 0C 24
61A0-	1C 3F 27 2C 2D B5 92 02
61A8-	49 00 E1 24 E5 0C 0C B5
61B0-	13 75 BE 6F 49 00 21 64
61B8-	0C 0C 3C FF 6D 92 92 01
61C0-	49 00 E1 64 1C 64 2D 0E
61C8-	BE 77 01 BE 6F 49 00 01
61D0-	65 0C 24 E4 3F 17 76 2D
61D8-	56 02 49 00 08 E5 E5 43
61E0-	E5 6D 92 12 49 00 25 E5
61E8-	E5 43 E5 6D 92 12 49 00
61F0-	41 1C 1C 0C 0C 0C B5 92
61F8-	02 2F 49 00 C0 29 6D FF
6200-	C0 2D 2D 96 02 49 00 65
6208-	0C 0C 1C 1C C7 6D 92 92
6210-	01 49 00 40 61 0C E4 3F
6218-	17 96 52 69 49 00 E1 24
6220-	24 0C 2D 0E 3E 17 2E 0C
6228-	96 1B 2D 49 00 24 24 0C
6230-	0C 0E 0E FE 13 2D 35 7E
6238-	49 00 24 24 24 2D AD BE
6240-	3F 4D 02 BE 3F 4D 49 00
6248-	43 24 24 0C 2D 0E 96 FA
6250-	13 2D 4D 01 00 24 24 24
6258-	2D 0E 0E 36 17 17 6F 49
6260-	01 00 24 24 24 2D 2D 96
6268-	3B B7 2A 2D 4D 00 24 24
6270-	24 2D 2D 96 FB 6D 92 49
6278-	00 20 24 64 2D 0E 96 2F
6280-	FA 13 2D 4D 01 00 24 24
6288-	24 96 29 2D 24 B4 12 36
6290-	4D 00 29 24 24 E4 2D 96
6298-	92 05 49 00 70 2D 0C 24
62A0-	24 B4 92 02 49 00 24 24
62A8-	24 4D B9 17 17 0E 0E 0E
62B0-	4D 00 24 24 24 96 92 29
62B8-	2D 4D 00 24 24 24 0E 0E
62C0-	0C 0C 36 36 36 4D 00 24
62C8-	24 24 0E 2E 36 35 25 24
62D0-	24 96 92 4D 00 E1 24 24
62D8-	0C 2D 0E 36 36 9F 2D 49
62E0-	00 24 24 24 2D 75 BE 3F
62E8-	96 49 49 00 E1 24 24 0C
62F0-	2D 0E 36 FE 0E 17 0D 4D
62F8-	00 24 24 24 2D 75 BE 3F

ADDR. MACHINE CODE

```

6300- 0E 0E 0E 4D 00 70 2D 0C
6308- E4 3F 1C 64 2D 0E 96 12
6310- 49 00 09 24 24 E4 6F 2D
6318- 96 92 49 00 E1 24 24 6C
6320- 09 36 36 BE 6F 49 00 08
6328- 3C 24 24 4D 31 36 3E BE
6330- 0D 49 00 24 24 24 4D 31
6338- 36 36 FF 0C 56 01 4D 00
6340- 64 0C 0C 0C FC 1B 76 56
6348- 71 7E 49 00 09 24 E4 1C
6350- 6C 09 BE 96 0A 49 00 64
6358- 0C 0C 0C 3C FF B5 92 2A
6360- 2D 4D 00 01 E5 24 24 2C
6368- 2D 96 92 2F 49 00 49 08
6370- 1C 1C 1C 1C 96 52 49 49
6378- 00 2D 25 24 24 3C 3F 96
6380- 92 01 49 49 00 09 24 24
6388- FC 62 0C 0E 0E 96 02 49
6390- 00 93 2D 2D 2D C4 49 00
6398- 40 C0 40 0E 0E 0E 92 49
63A0- 00 20 2C 2D E4 3F 56 12
63A8- 65 0E 06 49 00 28 24 24
63B0- B4 01 75 BE 77 01 49 00
63B8- 40 23 64 2D 0E 96 FF 6D
63C0- 02 49 00 41 E7 64 65 B4
63C8- 36 2E 06 49 00 08 1C 24
63D0- 0C 05 75 3E 3F 56 75 49
63D8- 00 08 24 6F 1C 64 0E 96
63E0- 52 49 00 01 65 3C E7 64
63E8- 75 B6 0A 49 00 43 24 24
63F0- B4 01 75 36 0E 49 00 08
63F8- 2D 1C 24 67 B0 92 4A 49
6400- 00 01 E1 0D 24 3C 44 96
6408- 92 01 49 00 20 0C 27 24
6410- 4D BA 16 0E 0E 49 00 08
6418- 2D 1C 24 24 6F 92 92 01
6420- 49 00 20 24 2C 0E 36 6E
6428- 24 E4 96 52 49 00 20 24
6430- 2C 75 36 76 49 00 41 FD
6438- 1C 04 64 05 75 06 B6 49
6440- 00 24 24 2C 75 BE 6F 52
6448- 49 00 40 E5 03 64 2D 36
6450- 36 2E 4D 00 43 24 24 0E
6458- 0C 75 92 02 49 00 43 2D
6460- 65 1C 07 E7 0C 2D B5 92
6468- 49 00 49 1C 24 2C 1F 0C
6470- B4 92 4A 49 00 41 E7 24
6478- 6C 31 36 2E 06 49 00 41
6480- 1C E4 6C 09 BE BE 4A 49
6488- 00 08 1C 24 6C 09 B6 FC
6490- 76 0E 49 00 60 0C 0C 0C
6498- DF 73 56 71 06 49 00 01
64A0- 65 24 24 DF 36 0E B5 09
64A8- 49 00 60 0C 0C 0C 3F 3F

```


ADDR. MACHINE CODE

```

64B0- 96 0A 2D 35 49 00 41 E4
64B8- 0C 64 B5 92 3A 05 49 00
64C0- 09 24 24 24 96 92 09 49
64C8- 00 65 64 1C E4 6F 92 92
64D0- 09 49 00 40 63 2D 0C 96
64D8- 02 49 00

```

ASCII SHAPE CODE #2 (Capital letters and numbers)

ADDR. MACHINE CODE

```

6000- 24 00 4A 00 59 00 63 00
6008- 6F 00 7C 00 88 00 96 00
6010- A2 00 AE 00 BB 00 C8 00
6018- D5 00 E3 00 F0 00 FD 00
6020- 09 01 14 01 21 01 2D 01
6028- 37 01 41 01 4D 01 56 01
6030- 62 01 70 01 7C 01 87 01
6038- 94 01 A0 01 AD 01 B7 01
6040- C2 01 CE 01 DB 01 E7 01
6048- F2 01 60 0C 0C 25 1C 3F
6050- 17 36 B6 29 65 24 96 49
6058- 00 29 24 24 24 17 96 52
6060- 29 49 00 64 0C 65 E4 3F
6068- 17 96 52 2D 2D 09 00 29
6070- 65 E4 67 01 E4 3F 17 96
6078- 72 49 49 00 49 24 24 24
6080- 17 17 17 2E 6D 16 49 00
6088- 70 2D 0C 24 1C 3F 27 2C
6090- 2D B5 92 02 49 00 E1 24
6098- E5 0C 0C B5 13 75 BE 6F
60A0- 49 00 21 64 0C 0C 3C FF
60A8- 6D 92 92 01 49 00 E1 64
60B0- 1C 64 2D 0E BE 77 01 BE
60B8- 6F 49 00 01 65 0C 24 E4
60C0- 3F 17 76 2D 56 02 49 00
60C8- 24 24 0C 0C 0E 0E FE 13
60D0- 2D 35 7E 49 00 24 24 24
60D8- 2D AD BE 3F 4D 02 BE 3F
60E0- 4D 49 00 43 24 24 0C 2D
60E8- 0E 96 FA 13 2D 4D 01 00
60F0- 24 24 24 2D 0E 0E 36 17
60F8- 17 6F 49 01 00 24 24 24
6100- 2D 2D 96 3B B7 2A 2D 4D
6108- 00 24 24 24 2D 2D 96 FB
6110- 6D 92 49 00 20 24 64 2D
6118- 0E 96 2F FA 13 2D 4D 01
6120- 00 24 24 24 96 29 2D 24
6128- B4 12 36 4D 00 29 24 24
6130- E4 2D 96 92 05 49 00 70
6138- 2D 0C 24 24 B4 92 02 49
6140- 00 24 24 24 4D B9 17 17

```

ADDR.	MACHINE CODE
6148-	0E 0E 0E 4D 00 24 24 24
6150-	96 92 29 2D 4D 00 24 24
6158-	24 0E 0E 0C 0C 36 36 36
6160-	4D 00 24 24 24 0E 2E 36
6168-	35 25 24 24 96 92 4D 00
6170-	E1 24 24 0C 2D 0E 36 36
6178-	9F 2D 49 00 24 24 24 2D
6180-	75 BE 3F 96 49 49 00 E1
6188-	24 24 0C 2D 0E 36 FE 0E
6190-	17 0D 4D 00 24 24 24 2D
6198-	75 BE 3F 0E 0E 0E 4D 00
61A0-	70 2D 0C E4 3F 1C 64 2D
61A8-	0E 96 12 49 00 09 24 24
61B0-	E4 6F 2D 96 92 49 00 E1
61B8-	24 24 6C 09 36 36 BE 6F
61C0-	49 00 08 3C 24 24 4D 31
61C8-	36 3E BE 0D 49 00 24 24
61D0-	24 4D 31 36 36 FF 0C 56
61D8-	01 4D 00 64 0C 0C 0C FC
61E0-	1B 76 56 71 7E 49 00 09
61E8-	24 E4 1C 6C 09 BE 96 0A
61F0-	49 00 64 0C 0C 0C 3C FF
61F8-	B5 92 2A 2D 4D 00

ASCII SHAPE CODE #3 (Capital letters, lower case and numbers)

ADDR.	MACHINE CODE
6000-	3E 00 7E 00 8D 00 97 00
6008-	A3 00 B0 00 BC 00 CA 00
6010-	D6 00 E2 00 EF 00 FC 00
6018-	09 01 17 01 24 01 31 01
6020-	3D 01 48 01 55 01 61 01
6028-	6B 01 75 01 81 01 8A 01
6030-	96 01 A4 01 B0 01 BB 01
6038-	C8 01 D4 01 E1 01 EB 01
6040-	F6 01 02 02 0F 02 1B 02
6048-	26 02 32 02 3E 02 49 02
6050-	54 02 5E 02 6A 02 74 02
6058-	7E 02 88 02 92 02 9D 02
6060-	A8 02 B3 02 BF 02 C7 02
6068-	D2 02 DB 02 E5 02 EF 02
6070-	FB 02 06 03 10 03 1A 03
6078-	25 03 30 03 3B 03 60 0C
6080-	0C 25 1C 3F 17 36 B6 29
6088-	65 24 96 49 00 29 24 24
6090-	24 17 96 52 29 49 00 64
6098-	0C 65 E4 3F 17 96 52 2D
60A0-	2D 09 00 29 65 E4 67 01
60A8-	E4 3F 17 96 72 49 49 00
60B0-	49 24 24 24 17 17 17 2E
60B8-	6D 16 49 00 70 2D 0C 24
60C0-	1C 3F 27 2C 2D B5 92 02

ADDR.	MACHINE CODE														
60C8-	49	00	E1	24	E5	0C	0C	B5							
60D0-	13	75	BE	6F	49	00	21	64							
60D8-	0C	0C	3C	FF	6D	92	92	01							
60E0-	49	00	E1	64	1C	64	2D	0E							
60E8-	BE	77	01	BE	6F	49	00	01							
60F0-	65	0C	24	E4	3F	17	76	2D							
60F8-	56	02	49	00	24	24	0C	0C							
6100-	0E	0E	FE	13	2D	35	7E	49							
6108-	00	24	24	24	2D	AD	BE	3F							
6110-	4D	02	BE	3F	4D	49	00	43							
6118-	24	24	0C	2D	0E	96	FA	13							
6120-	2D	4D	01	00	24	24	24	2D							
6128-	0E	0E	36	17	17	6F	49	01							
6130-	00	24	24	24	2D	2D	96	3B							
6138-	B7	2A	2D	4D	00	24	24	24							
6140-	2D	2D	96	FB	6D	92	49	00							
6148-	20	24	64	2D	0E	96	2F	FA							
6150-	13	2D	4D	01	00	24	24	24							
6158-	96	29	2D	24	B4	12	36	4D							
6160-	00	29	24	24	E4	2D	96	92							
6168-	05	49	00	70	2D	0C	24	24							
6170-	B4	92	02	49	00	24	24	24							
6178-	4D	B9	17	17	0E	0E	0E	4D							
6180-	00	24	24	24	96	92	29	2D							
6188-	4D	00	24	24	24	0E	0E	0C							
6190-	0C	36	36	36	4D	00	24	24							
6198-	24	0E	2E	36	35	25	24	24							
61A0-	96	92	4D	00	E1	24	24	0C							
61A8-	2D	0E	36	36	9F	2D	49	00							
61B0-	24	24	24	2D	75	BE	3F	96							
61B8-	49	49	00	E1	24	24	0C	2D							
61C0-	0E	36	FE	0E	17	0D	4D	00							
61C8-	24	24	24	2D	75	BE	3F	0E							
61D0-	0E	0E	4D	00	70	2D	0C	E4							
61D8-	3F	1C	64	2D	0E	96	12	49							
61E0-	00	09	24	24	E4	6F	2D	96							
61E8-	92	49	00	E1	24	24	6C	09							
61F0-	36	36	BE	6F	49	00	08	3C							
61F8-	24	24	4D	31	36	3E	BE	0D							
6200-	49	00	24	24	24	4D	31	36							
6208-	36	FF	0C	56	01	4D	00	64							
6210-	0C	0C	0C	FC	1B	76	56	71							
6218-	7E	49	00	09	24	E4	1C	6C							
6220-	09	BE	96	0A	49	00	64	0C							
6228-	0C	0C	3C	FF	B5	92	2A	2D							
6230-	4D	00	20	2C	2D	E4	3F	56							
6238-	12	65	0E	06	49	00	28	24							
6240-	24	B4	01	75	BE	77	01	49							
6248-	00	40	23	64	2D	0E	96	FF							
6250-	6D	02	49	00	41	E7	64	65							
6258-	B4	36	2E	06	49	00	08	1C							
6260-	24	0C	05	75	3E	3F	56	75							
6268-	49	00	08	24	6F	1C	64	0E							
6270-	96	52	49	00	01	65	3C	E7							

ADDR.	MACHINE CODE														
6278-	64	75	B6	0A	49	00	43	24							
6280-	24	B4	01	75	36	0E	49	00							
6288-	08	2D	1C	24	67	B0	92	4A							
6290-	49	00	01	E1	0D	24	3C	44							
6298-	96	92	01	49	00	20	0C	27							
62A0-	24	4D	BA	16	0E	0E	49	00							
62A8-	08	2D	1C	24	24	6F	92	92							
62B0-	01	49	00	20	24	2C	0E	36							
62B8-	6E	24	E4	96	52	49	00	20							
62C0-	24	2C	75	36	76	49	00	41							
62C8-	FD	1C	04	64	05	75	06	B6							
62D0-	49	00	24	24	2C	75	BE	6F							
62D8-	52	49	00	40	E5	03	64	2D							
62E0-	36	36	2E	4D	00	43	24	24							
62E8-	0E	0C	75	92	02	49	00	43							
62F0-	2D	65	1C	07	E7	0C	2D	B5							
62F8-	92	49	00	49	1C	24	2C	1F							
6300-	0C	B4	92	4A	49	00	41	E7							
6308-	24	6C	31	36	2E	06	49	00							
6310-	41	1C	E4	6C	09	BE	BE	4A							
6318-	49	00	08	1C	24	6C	09	B6							
6320-	FC	76	0E	49	00	60	0C	0C							
6328-	0C	DF	73	56	71	06	49	00							
6330-	01	65	24	24	DF	36	0E	B5							
6338-	09	49	00	60	0C	0C	0C	3F							
6340-	3F	96	0A	2D	35	49	00								

ASCII SHAPE TABLE GENERATION PROGRAM

The following program will store all the ASCII shapes consecutively, build a shape table index and save all of it as binary files for you. This program can be used for building any other shape tables you desire as well. However, you had better change the file name used in Line 610 unless you want all your other shape programs saved as "ASCII SHAPES"!!

Finally, to help you understand what each data statement does, the line numbers of each DATA statement correspond to each character's high (negative) ASCII values.

```

10  ONERR  GOTO 30
20  READ A: POKE 20480 + I,A:I = I + 1: GOTO 20
30  POKE 216,0:ND = 20480 + I - 1: DIM AD(255):AD(1) = 20480:CT =
    1
40  FOR I = 20480 TO ND - 1: IF PEEK (I) = 0 THEN CT = CT +
    1:AD(CT) = I + 1
50  NEXT
60  S = 20480 - (CT * 2 + 2): POKE S,CT: POKE S + 1,0
70  FOR I = 1 TO CT:A = AD(I) - S: POKE S + 2 * I,A - (INT (A /
    256) * 256): POKE S + 2 * I + 1,INT (A / 256): NEXT
80  POKE 232,S - (INT (S / 256) * 256): POKE 233,INT (S / 256)
160 DATA 73,1,73,0: REM <SPACE>
161 DATA 9,196,33,36,180,146,74,73,0: REM !
162 DATA 64,192,229,229,109,53,119,119,146,73,0: REM "
163 DATA 33,60,12,60,12,108,190,45,23,23,45,23,46,73,0: REM #
164 DATA 40,45,12,28,63,28,12,45,253,176,22,22,13,73,0: REM $
165 DATA 96,12,12,12,199,27,189,109,146,2,102,126,73,0: REM %
166 DATA 67,44,228,100,14,182,13,23,159,109,77,0: REM &
167 DATA 64,64,36,55,182,73,18,73,0: REM '
168 DATA 65,36,36,12,150,146,5,73,0: REM (
169 DATA 97,36,36,28,150,146,73,73,0: REM )
170 DATA 96,37,63,69,231,69,110,23,46,23,151,108,6,73,0: REM *
171 DATA 65,228,111,36,86,181,2,73,0: REM +
172 DATA 97,196,189,109,18,73,0: REM ,
173 DATA 64,24,45,45,150,73,0: REM -
174 DATA 8,189,109,73,0: REM .
175 DATA 96,12,12,12,150,18,73,0: REM /
176 DATA 96,12,12,37,28,63,23,54,182,41,101,36,150,73,0: REM 0
177 DATA 41,36,36,36,23,150,82,41,73,0: REM 1
178 DATA 100,12,101,228,63,23,150,82,45,45,9,0: REM 2
179 DATA 41,101,228,103,1,228,63,23,150,114,73,73,0: REM 3
180 DATA 73,36,36,36,23,23,23,46,109,22,73,0: REM 4
181 DATA 112,45,12,36,28,63,39,44,45,181,146,2,73,0: REM 5
182 DATA 225,36,229,12,12,181,19,117,190,111,73,0: REM 6
183 DATA 33,100,12,12,60,255,109,146,146,1,73,0: REM 7
184 DATA 225,100,28,100,45,14,190,119,1,190,111,73,0: REM 8
185 DATA 1,101,12,36,228,63,23,118,45,86,2,73,0: REM 9
186 DATA 8,229,229,67,229,109,146,18,73,0: REM :
187 DATA 37,229,229,67,229,109,146,18,73,0: REM ;

```


188 DATA 65,28,28,12,12,12,181,146,2,47,73,0: REM <
189 DATA 192,41,109,255,192,45,45,150,2,73,0: REM =
190 DATA 101,12,12,28,28,199,109,146,146,1,73,0: REM >
191 DATA 64,97,12,228,63,23,150,82,105,73,0: REM ?
192 DATA 225,36,36,12,45,14,62,23,46,12,150,27,45,73,0: REM E
193 DATA 36,36,12,12,14,14,254,19,45,53,126,73,0: REM A
194 DATA 36,36,36,45,173,190,63,77,2,190,63,77,73,0: REM B
195 DATA 67,36,36,12,45,14,150,250,19,45,77,1,0: REM C
196 DATA 36,36,36,45,14,14,54,23,23,111,73,1,0: REM D
197 DATA 36,36,36,45,45,150,59,183,42,45,77,0: REM E
198 DATA 36,36,36,45,45,150,251,109,146,73,0: REM F
199 DATA 32,36,100,45,14,150,47,250,19,45,77,1,0: REM G
200 DATA 36,36,36,150,41,45,36,180,18,54,77,0: REM H
201 DATA 41,36,36,228,45,150,146,5,73,0: REM I
202 DATA 112,45,12,36,36,180,146,2,73,0: REM J
203 DATA 36,36,36,77,185,23,23,14,14,14,77,0: REM K
204 DATA 36,36,36,150,146,41,45,77,0: REM L
205 DATA 36,36,36,14,14,12,12,54,54,54,77,0: REM M
206 DATA 36,36,36,14,46,54,53,37,36,36,150,146,77,0: REM N
207 DATA 225,36,36,12,45,14,54,54,159,45,73,0: REM O
208 DATA 36,36,36,45,117,190,63,150,73,73,0: REM P
209 DATA 225,36,36,12,45,14,54,254,14,23,13,77,0: REM Q
210 DATA 36,36,36,45,117,190,63,14,14,14,77,0: REM R
211 DATA 112,45,12,228,63,28,100,45,14,150,18,73,0: REM S
212 DATA 9,36,36,228,111,45,150,146,73,0: REM T
213 DATA 225,36,36,108,9,54,54,190,111,73,0: REM U
214 DATA 8,60,36,36,77,49,54,62,190,13,73,0: REM V
215 DATA 36,36,36,77,49,54,54,255,12,86,1,77,0: REM W
216 DATA 100,12,12,12,252,27,118,86,113,126,73,0: REM X
217 DATA 9,36,228,28,108,9,190,150,10,73,0: REM Y
218 DATA 100,12,12,12,60,255,181,146,42,45,77,0: REM Z
219 DATA 1,229,36,36,44,45,150,146,47,73,0: REM [
220 DATA 73,8,28,28,28,28,150,82,73,73,0: REM \
221 DATA 45,37,36,36,60,63,150,146,1,73,73,0: REM RIGHT BRACKET]
222 DATA 9,36,36,252,98,12,14,14,150,2,73,0: REM ^
223 DATA 147,45,45,45,196,73,0: REM _
224 DATA 64,192,64,14,14,14,146,73,0: REM `
225 DATA 32,44,45,228,63,86,18,101,14,6,73,0: REM a
226 DATA 40,36,36,180,1,117,190,119,1,73,0: REM b
227 DATA 64,35,100,45,14,150,255,109,2,73,0: REM c
228 DATA 65,231,100,101,180,54,46,6,73,0: REM d
229 DATA 8,28,36,12,5,117,62,63,86,117,73,0: REM e
230 DATA 8,36,111,28,100,14,150,82,73,0: REM f
231 DATA 1,101,60,231,100,117,182,10,73,0: REM g
232 DATA 67,36,36,180,1,117,54,14,73,0: REM h
233 DATA 8,45,28,36,103,176,146,74,73,0: REM i
234 DATA 1,225,13,36,60,68,150,146,1,73,0: REM j
235 DATA 32,12,39,36,77,186,22,14,14,73,0: REM k
236 DATA 8,45,28,36,36,111,146,146,1,73,0: REM l
237 DATA 32,36,44,14,54,110,36,228,150,82,73,0: REM m


```

238 DATA 32,36,44,117,54,118,73,0: REM n
239 DATA 65,253,28,4,100,5,117,6,182,73,0: REM o
240 DATA 36,36,44,117,190,111,82,73,0: REM p
241 DATA 64,229,3,100,45,54,54,46,77,0: REM q
242 DATA 67,36,36,14,12,117,146,2,73,0: REM r
243 DATA 67,45,101,28,7,231,12,45,181,146,73,0: REM s
244 DATA 73,28,36,44,31,12,180,146,74,73,0: REM t
245 DATA 65,231,36,108,49,54,46,6,73,0: REM u
246 DATA 65,28,228,108,9,190,190,74,73,0: REM v
247 DATA 8,28,36,108,9,182,252,118,14,73,0: REM w
248 DATA 96,12,12,12,223,115,86,113,6,73,0: REM x
249 DATA 1,101,36,36,223,54,14,181,9,73,0: REM y
250 DATA 96,12,12,12,63,63,150,10,45,53,73,0: REM z
251 DATA 65,228,12,100,181,146,58,5,73,0: REM {
252 DATA 9,36,36,36,150,146,9,73,0: REM !
253 DATA 101,100,28,228,111,146,146,9,73,0: REM }
254 DATA 64,99,45,12,150,2,73,0: REM ~
290 POKE 216,0
500 HGR : HOME : VTAB 21: HCOLOR= 3: ROT= 0: SCALE= 1: HPLOT
    0,159
510 FOR I = 32 TO 126
520 SH = SH + 1: IF F = 0 THEN GOSUB 1000: HOME : VTAB 21: PRINT
    CHR$(I): HGR : DRAW SH AT 1,159: GOTO 540
530 PRINT CHR$(I): DRAW SH
540 F = F + 1: IF F = 17 THEN F = 0
550 NEXT
600 LL = ND - S + 1
610 PRINT : PRINT CHR$(4)"BSAVE ASCII SHAPES,A"S",L"LL
999 END
1000 WAIT - 16384,128: GET A$: RETURN

```

MONITOR COMMANDS

The monitor inside your Apple has a limited but useful set of commands for examining, creating, moving and executing programs. The monitor is accessed from BASIC with CALL -151, and is prompted with an asterisk (*). To exit the monitor, enter CTRL-C. The following commands and examples show how to work with the monitor.

COMMAND	MEANING	EXAMPLE
addr1.addr2	Examine range of memory beginning at addr1 and ending at addr2. As soon as RETURN is pressed, hexadecimal values stored in addr1 - addr2 will be displayed.	300.31D
addr L	Lists the next twenty 6502 instructions and or unknown code (???) bytes; so 20-60 bytes can be displayed. If no address is indicated, the listing will be from the last address. If more than a single L is entered, 20 instructions per L will be listed.	300L L LL
addr G	Program, beginning at addr will be executed.	300G
addr1 < addr2.addr3 M	Compares memory in the range from addr2 through addr3, to the memory contents beginning at addr1 through the total number of addresses in addr2 through addr3. (V is for "verify.")	350<300.31D V
addr : value(s)	Enters value (hexadecimal) into addr. Once the address has been specified it is possible to enter up to 85 values that will be stored in consecutive addresses beginning at addr. Spaces must be inserted between 2 digit values to be entered.	300 : A0 301 : 12 BF
I	Sets output to inverse, but the input remains normal.	I
N	Sets all output to normal.	N
S	Sets up STEP mode whereby each command is executed one instruction at a time. Used in finding program bugs. Multiple "S's" can be entered to see several executed steps at one time. This is the most useful way to examine the contents of the various registers displayed for each step. (Only in old monitor ROM.)	300S SSSS

COMMAND	MEANING	EXAMPLE
T	Sets up TRACE mode, similar to STEP but steps through program rapidly until a BRK is found. This procedure is almost impossible to follow, but it can locate bugs. (Only in old monitor ROM.)	T
CTRL-E	Shows the current values of the Accumulator (A), X Register (X), Y Register (Y), Process Register (P) and Stack Pointer(S).	CTRL-E A=01 X=00 Y=D1 P=C0 S=58
CTRL-B	Return to BASIC without saving the resident BASIC program. This has the same results as issuing NEW from BASIC.	CTRL-B
CTRL-C	Return to BASIC and saving all program values and the program itself.	CTRL-C
CTRL-P	Treated as a PRINTER command to a specified slot number. Usually used with a "6" to boot DOS from monitor.	6 CTRL-P
CTRL-Y	Jumps to \$3F8. The contents of \$3F8 can be changed to JMP to your own progra, or it is possible to change \$3F8 to be executed there. IF, for example you enter: 3F8: 20 58 FC 60 pressing CTRL-Y will result in the same state as a HOME command.	CTRL-Y
+ -	Addition and subtraction of 2 digit dexadecimal values is possible. All that is required is to enter the value, a plus or minus sign and press RETURN. The addition or subtraction does not display an overflow (carry) or underflow (borrow). The user is expected to realize this will occur.	*A3 + 0F *BD - AA

APPLE //e MONITOR COMPARISON CHART

There are some differences between the Apple II Autostart Monitor ROM and the Apple //e's monitor ROM. This table shows the bytes which are different in the Apple //e's monitor ROM.

RANGE STARTING ADDRESS		RANGE ENDING ADDRESS		NUMBER OF BYTES IN RANGE	
DEC	HEX	DEC	HEX	DEC	HEX
64,117	FA75	64,122	FA7A	6	6
64,266	FB0A	64,269	FB0D	4	4
64,337	FB51	64,340	FB54	4	4
64,419	FBA3	64,419	FBA3	1	1
64,435	FBB3	64,447	FBBF	13	D
64,578	FC42	64,609	FC61	32	20
64,624	FC70	64,679	FCA7	56	38
64,795	FD1B	64,813	FD2D	19	13
64,816	FD30	64,816	FD30	1	1
64,834	FD42	64,835	FD43	2	2
64,899	FD83	64,899	FD83	1	1
65,199	FEAF	65,199	FEAF	1	1
65,221	FEC5	65,225	FEC9	5	5

6502 PROCESSOR STATUS FLAG CHART

This table shows the 256 (\$0-\$FF) different possible combinations of the status flags on the processor status register. The "set" flags are designated with codes for each flag:

NEGATIVE = N OVERFLOW = V DECIMAL = D. ns INTERRUPT = I
 BREAK = B ZERO = Z CARRY = C
 BITS (UNASSIGNED) = + or - for clear or set

When examining machine code listings, the condition of the status register is important. This table will be useful in determining the condition of the flags, tracing, single stepping or break opcode.

\$0-\$3F		\$40-\$7F		\$80-\$BF		\$C0-\$FF	
HEX	FLAGS SET	HEX	FLAGS SET	HEX	FLAGS SET	HEX	FLAGS SET
0	-----	40	-V-----	80	N-----	C0	NV-----
1	-----C	41	-V-----C	81	N-----C	C1	NV-----C
2	-----Z-	42	-V-----Z-	82	N-----Z-	C2	NV-----Z-
3	-----ZC	43	-V-----ZC	83	N-----ZC	C3	NV-----ZC
4	----I--	44	-V---I--	84	N---I--	C4	NV---I--
5	----I-C	45	-V---I-C	85	N---I-C	C5	NV---I-C
6	----IZ-	46	-V---IZ-	86	N---IZ-	C6	NV---IZ-
7	----IZC	47	-V---IZC	87	N---IZC	C7	NV---IZC
8	----D---	48	-V--D---	88	N--D---	C8	NV--D---
9	----D-C	49	-V--D-C	89	N--D-C	C9	NV--D-C
A	----D-Z-	4A	-V--D-Z-	8A	N--D-Z-	CA	NV--D-Z-
B	----D-ZC	4B	-V--D-ZC	8B	N--D-ZC	CB	NV--D-ZC
C	----DI--	4C	-V--DI--	8C	N--DI--	CC	NV--DI--
D	----DI-C	4D	-V--DI-C	8D	N--DI-C	CD	NV--DI-C
E	----DIZ-	4E	-V--DIZ-	8E	N--DIZ-	CE	NV--DIZ-
F	----DIZC	4F	-V--DIZC	8F	N--DIZC	CF	NV--DIZC
10	---B----	50	-V-B----	90	N--B----	D0	NV-B----
11	---B---C	51	-V-B---C	91	N--B---C	D1	NV-B---C
12	---B--Z-	52	-V-B--Z-	92	N--B--Z-	D2	NV-B--Z-
13	---B--ZC	53	-V-B--ZC	93	N--B--ZC	D3	NV-B--ZC
14	---B-I--	54	-V-B-I--	94	N--B-I--	D4	NV-B-I--
15	---B-I-C	55	-V-B-I-C	95	N--B-I-C	D5	NV-B-I-C
16	---B-IZ-	56	-V-B-IZ-	96	N--B-IZ-	D6	NV-B-IZ-
17	---B-IZC	57	-V-B-IZC	97	N--B-IZC	D7	NV-B-IZC
18	---BD---	58	-V-BD---	98	N--BD---	D8	NV-BD---
19	---BD-C	59	-V-BD-C	99	N--BD-C	D9	NV-BD-C
1A	---BD-Z-	5A	-V-BD-Z-	9A	N--BD-Z-	DA	NV-BD-Z-
1B	---BD-ZC	5B	-V-BD-ZC	9B	N--BD-ZC	DB	NV-BD-ZC
1C	---BDI--	5C	-V-BDI--	9C	N--BDI--	DC	NV-BDI--
1D	---BDI-C	5D	-V-BDI-C	9D	N--BDI-C	DD	NV-BDI-C
1E	---BDIZ-	5E	-V-BDIZ-	9E	N--BDIZ-	DE	NV-BDIZ-
1F	---BDIZC	5F	-V-BDIZC	9F	N--BDIZC	DF	NV-BDIZC

\$0-\$3F		\$40-\$7F		\$80-\$BF		\$C0-\$FF	
HEX	FLAGS SET	HEX	FLAGS SET	HEX	FLAGS SET	HEX	FLAGS SET
20	---+-----	60	-V+-----	A0	N-+-----	E0	NV+-----
21	---+----C	61	-V+----C	A1	N-+----C	E1	NV+----C
22	---+---Z-	62	-V+---Z-	A2	N-+---Z-	E2	NV+---Z-
23	---+---ZC	63	-V+---ZC	A3	N-+---ZC	E3	NV+---ZC
24	---+--I--	64	-V+--I--	A4	N-+--I--	E4	NV+--I--
25	---+--I-C	65	-V+--I-C	A5	N-+--I-C	E5	NV+--I-C
26	---+--IZ-	66	-V+--IZ-	A6	N-+--IZ-	E6	NV+--IZ-
27	---+--IZC	67	-V+--IZC	A7	N-+--IZC	E7	NV+--IZC
28	---+D---	68	-V+D---	A8	N-+D---	E8	NV+D---
29	---+D--C	69	-V+D--C	A9	N-+D--C	E9	NV+D--C
2A	---+D-Z-	6A	-V+D-Z-	AA	N-+D-Z-	EA	NV+D-Z-
2B	---+D-ZC	6B	-V+D-ZC	AB	N-+D-ZC	EB	NV+D-ZC
2C	---+DI--	6C	-V+DI--	AC	N-+DI--	EC	NV+DI--
2D	---+DI-C	6D	-V+DI-C	AD	N-+DI-C	ED	NV+DI-C
2E	---+DIZ-	6E	-V+DIZ-	AE	N-+DIZ-	EE	NV+DIZ-
2F	---+DIZC	6F	-V+DIZC	AF	N-+DIZC	EF	NV+DIZC
30	---+B----	70	-V+B----	B0	N-+B----	F0	NV+B----
31	---+B---C	71	-V+B---C	B1	N-+B---C	F1	NV+B---C
32	---+B--Z-	72	-V+B--Z-	B2	N-+B--Z-	F2	NV+B--Z-
33	---+B--ZC	73	-V+B--ZC	B3	N-+B--ZC	F3	NV+B--ZC
34	---+B-I--	74	-V+B-I--	B4	N-+B-I--	F4	NV+B-I--
35	---+B-I-C	75	-V+B-I-C	B5	N-+B-I-C	F5	NV+B-I-C
36	---+B-IZ-	76	-V+B-IZ-	B6	N-+B-IZ-	F6	NV+B-IZ-
37	---+B-IZC	77	-V+B-IZC	B7	N-+B-IZC	F7	NV+B-IZC
38	---+BD---	78	-V+BD---	B8	N-+BD---	F8	NV+BD---
39	---+BD--C	79	-V+BD--C	B9	N-+BD--C	F9	NV+BD--C
3A	---+BD-Z-	7A	-V+BD-Z-	BA	N-+BD-Z-	FA	NV+BD-Z-
3B	---+BD-ZC	7B	-V+BD-ZC	BB	N-+BD-ZC	FB	NV+BD-ZC
3C	---+BDI--	7C	-V+BDI--	BC	N-+BDI--	FC	NV+BDI--
3D	---+BDI-C	7D	-V+BDI-C	BD	N-+BDI-C	FD	NV+BDI-C
3E	---+BDIZ-	7E	-V+BDIZ-	BE	N-+BDIZ-	FE	NV+BDIZ-
3F	---+BDIZC	7F	-V+BDIZC	BF	N-+BDIZC	FF	NV+BDIZC

6502 OPCODES : NUMERIC ORDER

In examining machine language routines listings, it is often necessary to know the mnemonics and addressing modes for the various assembly language opcodes. Using this chart, arranged numerically by machine language opcode, in both decimal and hexadecimal, the user can quickly find the meaning of a machine language opcode. For example, \$AC (172 decimal) can quickly be found to be LDY in the "absolute" addressing mode.

For those wishing to translate or create small machine language routines from BASIC using PEEK and POKE statements, the decimal values can be translated into opcodes. For example, the following program can be used to examine a machine level routine located at \$300-\$306 (768-774 decimal):

```
10 FOR I = 768 TO 774
20 PRINT PEEK (I),
30 NEXT I
```

Suppose you got the following output:

```
32      88      252
32     228      251
96
```

First, look up 32 in the decimal column. That turns out to be JSR (jump to subroutine). This means the next two bytes will be the subroutine address, with the low byte followed by the high byte. Using the DECIMAL TO HEX CONVERSION CHART, you will find that 88 = \$58 and 252 = \$FC. That means the first instruction was to jump to the subroutine at \$FC58. (Remember, to get the correct address, we had to reverse the bytes of the addresses.) The second "32" is another JSR, and we find the hex values to be 228 = \$E4 and 251 = \$FB, reversed to \$FBE4. Finally, we look up 96, and not surprisingly it is \$60, the machine language opcode for RTS. So we have found the little machine language program clears the screen (JSR \$FC58) and rings a bell (JSR \$FBE4), and then returns to its origins with an RTS.

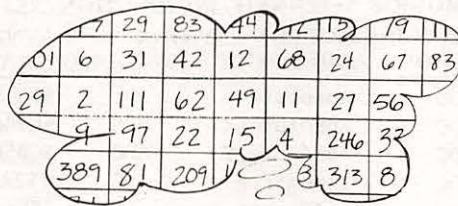
The dashed (-) entries are byte values with no assigned opcodes or defined functions. There are 105 such bytes, possibly available for later releases of new enhanced versions of the 6502 microprocessor. (See the 65C02 opcode chart for additional opcodes.)

MACH LAN ASSEMBLY LANGUAGE				MACH LAN ASSEMBLY LANGUAGE			
OPCODE		OPCODE		OPCODE		OPCODE	
DEC	HEX	MNEMON	ADDRESSING MODE	DEC	HEX	ADDRESSING MODE	
0	0	BRK	Implied	128	80	-	
1	1	ORA	(Indirect,X)	129	81	STA	(Indirect,X)
2	2	-		130	82	-	
3	3	-		131	83	-	

MACH LAN OPCODE	ASSEMBLY LANGUAGE OPCODE		MACH LAN OPCODE	ASSEMBLY LANGUAGE OPCODE
DEC	HEX	MNEMON ADDRESSING MODE	DEC	HEX ADDRESSING MODE
4	4	-	132	84 STY Zero Page
5	5	ORA Zero Page	133	85 STA Zero Page
6	6	ASL Zero Page	134	86 STX Zero Page
7	7	-	135	87 -
8	8	PHP Implied	136	88 DEY Implied
9	9	ORA Immediate	137	89 -
10	A	ASL Accumulator	138	8A TXA Implied
11	B	-	139	8B -
12	C	-	140	8C STY Absolute
13	D	ORA Absolute	141	8D STA Absolute
14	E	ASL Absolute	142	8E STX Absolute
15	F	-	143	8F -
16	10	BPL Relative	144	90 BCC Relative
17	11	ORA (Indirect), Y	145	91 STA (Indirect), Y
18	12	-	146	92 -
19	13	-	147	93 -
20	14	-	148	94 STY Zero Page, X
21	15	ORA Zero Page, X	149	95 STA Zero Page, X
22	16	ASL Zero Page, X	150	96 STX Zero Page, Y
23	17	-	151	97 -
24	18	CLC Implied	152	98 TYA Implied
25	19	ORA Absolute, Y	153	99 STA Absolute, Y
26	1A	-	154	9A TXS Implied
27	1B	-	155	9B -
28	1C	-	156	9C -
29	1D	ORA Absolute, X	157	9D STA Absolute, X
30	1E	ASL Absolute, X	158	9E -
31	1F	-	159	9F -
32	20	JSR Absolute	160	A0 LDY Immediate
33	21	AND (Indirect, X)	161	A1 LDA (Indirect, X)
34	22	-	162	A2 LDX Immediate
35	23	-	163	A3 -
36	24	BIT Zero Page	164	A4 LDY Zero Page
37	25	AND Zero Page	165	A5 LDA Zero Page
38	26	ROL Zero Page	166	A6 LDX Zero Page
39	27	-	167	A7 -
40	28	PLP Implied	168	A8 TAY Implied
41	29	AND Immediate	169	A9 LDA Immediate
42	2A	ROL Accumulator	170	AA TAX Implied
43	2B	-	171	AB -
44	2C	BIT Absolute	172	AC LDY Absolute
45	2D	AND Absolute	173	AD LDA Absolute
46	2E	ROL Absolute	174	AE LDX Absolute
47	2F	-	175	AF -
48	30	BMI Relative	176	B0 BCS Relative
49	31	AND (Indirect), Y	177	B1 LDA (Indirect), Y
50	32	-	178	B2 -
51	33	-	179	B3 -
52	34	-	180	B4 LDY Zero Page, X

MACH LAN OPCODE		ASSEMBLY LANGUAGE OPCODE		MACH LAN OPCODE		ASSEMBLY LANGUAGE OPCODE	
DEC	HEX	MNEMON	ADDRESSING MODE	DEC	HEX	MNEMON	ADDRESSING MODE
53	35	AND	Zero Page,X	181	B5	LDA	Zero Page,X
54	36	ROL	Zero Page,X	182	B6	LDX	Zero Page,Y
55	37	-		183	B7	-	
56	38	SEC	Implied	184	B8	CLV	Implied
57	39	AND	Absolute,Y	185	B9	LDA	Absolute,Y
58	3A	-		186	BA	TSX	Implied
59	3B	-		187	BB	-	
60	3C	-		188	BC	LDY	Absolute,X
61	3D	AND	Absolute,X	189	BD	LDA	Absolute,X
62	3E	ROL	Absolute,X	190	BE	LDX	Absolute,Y
63	3F	-		191	BF	-	
64	40	RTI	Implied	192	C0	CPY	Immediate
65	41	EOR	(Indirect,X)	193	C1	CMP	(Indirect,X)
66	42	-		194	C2	-	
67	43	-		195	C3	-	
68	44	-		196	C4	CPY	Zero Page
69	45	EOR	Zero Page	197	C5	CMP	Zero Page
70	46	LSR	Zero Page	198	C6	DEC	Zero Page
71	47	-		199	C7	-	
72	48	PHA	Implied	200	C8	INY	Implied
73	49	EOR	Immediate	201	C9	CMP	Immediate
74	4A	LSR	Accumulator	202	CA	DEX	Implied
75	4B	-		203	CB	-	
76	4C	JMP	Absolute	204	CC	CPY	Absolute
77	4D	EOR	Absolute	205	CD	CMP	Absolute
78	4E	LSR	Absolute	206	CE	DEC	Absolute
79	4F	-		207	CF	-	
80	50	BVC	Relative	208	D0	BNE	Relative
81	51	EOR	(Indirect),Y	209	D1	CMP	(Indirect),Y
82	52	-		210	D2	-	
83	53	-		211	D3	-	
84	54	-		212	D4	-	
85	55	EOR	Zero Page,X	213	D5	CMP	Zero Page,X
86	56	LSR	Zero Page,X	214	D6	DEC	Zero Page,X
87	57	-		215	D7	-	
88	58	CLI	Implied	216	D8	CLD	Implied
89	59	EOR	Absolute,Y	217	D9	CMP	Absolute,Y
90	5A	-		218	DA	-	
91	5B	-		219	DB	-	
92	5C	-		220	DC	-	
93	5D	EOR	Absolute,X	221	DD	CMP	Absolute,X
94	5E	LSR	Absolute,X	222	DE	DEC	Absolute,X
95	5F	-		223	DF	-	
96	60	RTS	Implied	224	E0	CPX	Immediate
97	61	ADC	(Indirect,X)	225	E1	SBC	(Indirect,X)
98	62	-		226	E2	-	
99	63	-		227	E3	-	
100	64	-		228	E4	CPX	Zero Page
101	65	ADC	Zero Page	229	E5	SBC	Zero Page

MACH LAN OPCODE	ASSEMBLY LANGUAGE OPCODE			MACH LAN OPCODE	ASSEMBLY LANGUAGE OPCODE		
DEC	HEX	MNEMON	ADDRESSING MODE	DEC	HEX	MNEMON	ADDRESSING MODE
102	66	ROR	Zero Page	230	E6	INC	Zero Page
103	67	-		231	E7	-	
104	68	PLA	Implied	232	E8	INX	Implied
105	69	ADC	Immediate	233	E9	SBC	Immediate
106	6A	ROR	Accumulator	234	EA	NOP	Implied
107	6B	-		235	EB	-	
108	6C	JMP	(Indirect)	236	EC	CPX	Absolute
109	6D	ADC	Absolute	237	ED	SBC	Absolute
110	6E	ROR	Absolute	238	EE	INC	Absolute
111	6F	-		239	EF	-	
112	70	BVS	Relative	240	F0	BEQ	Relative
113	71	ADC	(Indirect), Y	241	F1	SBC	(Indirect), Y
114	72	-		242	F2	-	
115	73	-		243	F3	-	
116	74	-		244	F4	-	
117	75	ADC	Zero Page, X	245	F5	SBC	Zero Page, X
118	76	ROR	Zero Page, X	246	F6	INC	Zero Page, X
119	77	-		247	F7	-	
120	78	SEI	Implied	248	F8	SED	Implied
121	79	ADC	Absolute, Y	249	F9	SBC	Absolute, Y
122	7A	-		250	FA	-	
123	7B	-		251	FB	-	
124	7C	-		252	FC	-	
125	7D	ADC	Absolute, X	253	FD	SBC	Absolute, X
126	7E	ROR	Absolute, X	254	FE	INC	Absolute, X
127	7F	-		255	FF	-	



6502 OPCODES: ALPHABETIC ORDER

This chart provides a quick reference for looking up opcodes used in assembly language programming. The chart is arranged alphabetically by mnemonic opcode, including multiple addressing modes where applicable. It should be noted that different opcodes have different machine language opcodes when in different addressing modes. The cycle times for branches are explained in the text.

The far right column shows the number of cycles taken for an instruction under different circumstances. Those instructions using "ABSOLUTE,X", "ABSOLUTE,Y", and "(INDIRECT),Y" modes and showing a "+1" following the cycling time, must have +1 added to their execution time if the index register (X or Y), when added to the base address, causes the high byte to be incremented. The first number, reading from left to right, is the execution time when no branch is taken. A further +1 is added to the timing cycle if a branch is taken and the destination address' high byte is greater or less than the instruction's address.

The example operands are typical ones for the various instructions (e.g., \$FC58, \$D260) used to illustrate syntax in different modes.

The following abbreviations were used:

ASSEM LAN INSTRUCTION SYNTAX = Assembly language instruction syntax without labels.

MACH LAN INSTRUCT SYNTAX = Machine language instruction syntax.

MACH LAN TIMING (CYCLES) FOR EA INST = Machine language timing (cycles) for each instruction.

MACHINE LANGUAGE		ASSEMBLY LANGUAGE MNEMONIC & ALTERNATE ADDRESSING MODES		ASSEM LAN INSTRUCTION SYNTAX	MACH LAN INSTRUCT SYNTAX	MACH LAN TIMING (CYCLES) FOR EA INST			CYC
OPCODE DEC	HEX	OPCODE	ADRES MODE	OPCODE	OPERAND	OPCD	OPRND		
109	6D	ADC	Absolute	ADC	\$FC58	6D	58 FC		4
125	7D	ADC	Absolute,X	ADC	\$D260,X	7D	60 D2		4 +1
121	79	ADC	Absolute,Y	ADC	\$C050,Y	79	50 C0		4 +1
105	69	ADC	Immediate	ADC	#\$2A	69	2A		2
97	61	ADC	(Indirect,X)	ADC	(\$E4,X)	61	E4		6
113	71	ADC	(Indirect),Y	ADC	(\$6),Y	71	06		5 +1
101	65	ADC	Zero Page	ADC	\$B7	65	B7		3
117	75	ADC	Zero Page,X	ADC	\$9D,X	75	9D		4
45	2D	AND	Absolute	AND	\$FC58	2D	58 FC		4
61	3D	AND	Absolute,X	AND	\$D260,X	3D	60 D2		4 +1
57	39	AND	Absolute,Y	AND	\$C050,Y	39	50 C0		4 +1
41	29	AND	Immediate	AND	#\$2A	29	2A		2
33	21	AND	(Indirect,X)	AND	(\$E4,X)	21	E4		6
49	31	AND	(Indirect),Y	AND	(\$6),Y	31	06		5 +1
37	25	AND	Zero Page	AND	\$B7	25	B7		3
53	35	AND	Zero Page,X	AND	\$9D,X	35	9D		4

MACHINE LANGUAG	ASSEMBLY LANGUAGE MNEMONIC & ALTERNATE	ASSEM LAN INSTRUCTION	MACH LAN INSTRUCT	MACH LAN TIMING (CYCLES)					
OPCODE DEC	HEX	ADDRESSING MODES OPCODE	ADRES MODE	SYNTAX OPCODE	SYNTAX OPERAND	FOR EA INST OPCD	OPRND	CYC	
14	E	ASL	Absolute	ASL	\$FC58	0E	58 FC	6	
30	1E	ASL	Absolute, X	ASL	\$D260, X	1E	60 D2	7	
10	A	ASL	Accumulator	ASL		0A		2	
6	6	ASL	Zero Page	ASL	\$B7	06	B7	5	
22	16	ASL	Zero Page, X	ASL	\$9D, X	16	9D	6	
144	90	BCC	Relative	BCC	\$FC22	90	CA	2	+1 +1
176	B0	BCS	Relative	BCS	\$FC22	B0	CA	2	+1 +1
240	F0	BEQ	Relative	BEQ	\$FC22	F0	CA	2	+1 +1
44	2C	BIT	Absolute	BIT	\$FC58	2C	58 FC	4	
36	24	BIT	Zero Page	BIT	\$B7	24	B7	3	
48	30	BMI	Relative	BMI	\$FC22	30	CA	2	+1 +1
208	D0	BNE	Relative	BNE	\$FC22	D0	CA	2	+1 +1
16	10	BPL	Relative	BPL	\$FC22	10	CA	2	+1 +1
0	0	BRK	Implied	BRK		00		7	
80	50	BVC	Relative	BVC	\$FC22	50	CA	2	+1 +1
112	70	BVS	Relative	BVS	\$FC22	70	CA	2	+1 +1
24	18	CLC	Implied	CLC		18		2	
216	D8	CLD	Implied	CLD		D8		2	
88	58	CLI	Implied	CLI		58		2	
184	B8	CLV	Implied	CLV		B8		2	
205	CD	CMP	Absolute	CMP	\$FC58	CD	58 FC	4	
221	DD	CMP	Absolute, X	CMP	\$D260, X	DD	60 D2	4	+1
217	D9	CMP	Absolute, Y	CMP	\$C050, Y	D9	50 C0	4	+1
201	C9	CMP	Immediate	CMP	#\$2A	C9	2A	2	
193	C1	CMP	(Indirect, X)	CMP	(\$E4, X)	C1	E4	6	
209	D1	CMP	(Indirect), Y	CMP	(\$6), Y	D1	06	5	+1
197	C5	CMP	Zero Page	CMP	\$B7	C5	B7	3	
213	D5	CMP	Zero Page, X	CMP	\$9D, X	D5	9D	4	
236	EC	CPX	Absolute	CPX	\$FC58	EC	58 FC	4	
224	E0	CPX	Immediate	CPX	#\$2A	E0	2A	2	
228	E4	CPX	Zero Page	CPX	\$B7	E4	B7	3	

MACHINE LANGUAG	ASSEMBLY LANGUAGE MNEMONIC & ALTERNATE	ASSEM LAN INSTRUCTION	MACH LAN INSTRUCT	MACH LAN TIMING (CYCLES)				
OPCODE	ADDRESSING	MODES	SYNTAX	SYNTAX	FOR EA	INST		
DEC	HEX	OPCODE	ADRES MODE	OPCODE	OPERAND	OPCD	OPRND	CYC
204	CC	CPY	Absolute	CPY	\$FC58	CC	58 FC	4
192	C0	CPY	Immediate	CPY	#\$2A	C0	2A	2
196	C4	CPY	Zero Page	CPY	\$B7	C4	B7	3
206	CE	DEC	Absolute	DEC	\$FC58	CE	58 FC	6
222	DE	DEC	Absolute, X	DEC	\$D260, X	DE	60 D2	7
198	C6	DEC	Zero Page	DEC	\$B7	C6	B7	5
214	D6	DEC	Zero Page, X	DEC	\$9D, X	D6	9D	6
202	CA	DEX	Implied	DEX		CA		2
136	88	DEY	Implied	DEY		88		2
77	4D	EOR	Absolute	EOR	\$FC58	4D	58 FC	4
93	5D	EOR	Absolute, X	EOR	\$D260, X	5D	60 D2	4 +1
89	59	EOR	Absolute, Y	EOR	\$C050, Y	59	50 C0	4 +1
73	49	EOR	Immediate	EOR	#\$2A	49	2A	2
65	41	EOR	(Indirect, X)	EOR	(\$E4, X)	41	E4	6
81	51	EOR	(Indirect), Y	EOR	(\$6), Y	51	06	5 +1
69	45	EOR	Zero Page	EOR	\$B7	45	B7	3
85	55	EOR	Zero Page, X	EOR	\$9D, X	55	9D	4
238	EE	INC	Absolute	INC	\$FC58	EE	58 FC	6
254	FE	INC	Absolute, X	INC	\$D260, X	FE	60 D2	7
230	E6	INC	Zero Page	INC	\$B7	E6	B7	5
246	F6	INC	Zero Page, X	INC	\$9D, X	F6	9D	6
232	E8	INX	Implied	INX		E8		2
200	C8	INY	Implied	INY		C8		2
76	4C	JMP	Absolute	JMP	\$FC58	4C	58 FC	3
108	6C	JMP	(Indirect)	JMP	(\$3D1)	6C	D1 03	5
32	20	JSR	Absolute	JSR	\$FC58	20	58 FC	6
173	AD	LDA	Absolute	LDA	\$FC58	AD	58 FC	4
189	BD	LDA	Absolute, X	LDA	\$D260, X	BD	60 D2	4 +1
185	B9	LDA	Absolute, Y	LDA	\$C050, Y	B9	50 C0	4 +1
169	A9	LDA	Immediate	LDA	#\$2A	A9	2A	2
161	A1	LDA	(Indirect, X)	LDA	(\$E4, X)	A1	E4	6
177	B1	LDA	(Indirect), Y	LDA	(\$6), Y	B1	06	5 +1
165	A5	LDA	Zero Page	LDA	\$B7	A5	B7	3
181	B5	LDA	Zero Page, X	LDA	\$9D, X	B5	9D	4
174	AE	LDX	Absolute	LDX	\$FC58	AE	58 FC	4
190	BE	LDX	Absolute, Y	LDX	\$C050, Y	BE	50 C0	4 +1
162	A2	LDX	Immediate	LDX	#\$2A	A2	2A	2
166	A6	LDX	Zero Page	LDX	\$B7	A6	B7	3
182	B6	LDX	Zero Page, Y	LDX	\$A3, Y	B6	A3	4

MACHINE ASSEMBLY LANGUAGE ASSEMBLY LANGUAGE MACHINE MACHINE
LANGUAGE MNEMONIC & ALTERNATE INSTRUCTION INSTRUCTION TIMING (CYCLES)

OPCODE	ADDRESSING MODES	SYNTAX	SYNTAX	FOR EA INST				
DEC	HEX	OPCODE	ADRES MODE	OPCODE	OPERAND	OPCD	OPRND	CYC
172	AC	LDY	Absolute	LDY	\$FC58	AC	58 FC	4
188	BC	LDY	Absolute, X	LDY	\$D260, X	BC	60 D2	4 +1
160	A0	LDY	Immediate	LDY	#\$2A	A0	2A	2
164	A4	LDY	Zero Page	LDY	\$B7	A4	B7	3
180	B4	LDY	Zero Page, X	LDY	\$9D, X	B4	9D	4
78	4E	LSR	Absolute	LSR	\$FC58	4E	58 FC	6
94	5E	LSR	Absolute, X	LSR	\$D260, X	5E	60 D2	7
74	4A	LSR	Accumulator	LSR		4A		2
70	46	LSR	Zero Page	LSR	\$B7	46	B7	5
86	56	LSR	Zero Page, X	LSR	\$9D, X	56	9D	6
234	EA	NOP	Implied	NOP		EA		2
13	D	ORA	Absolute	ORA	\$FC58	0D	58 FC	4
29	1D	ORA	Absolute, X	ORA	\$D260, X	1D	60 D2	4 +1
25	19	ORA	Absolute, Y	ORA	\$C050, Y	19	50 C0	4 +1
9	9	ORA	Immediate	ORA	#\$2A	09	2A	2
1	1	ORA	(Indirect, X)	ORA	(\$E4, X)	01	E4	6
17	11	ORA	(Indirect), Y	ORA	(\$6), Y	11	06	5 +1
5	5	ORA	Zero Page	ORA	\$B7	05	B7	3
21	15	ORA	Zero Page, X	ORA	\$9D, X	15	9D	4
72	48	PHA	Implied	PHA		48		3
8	8	PHP	Implied	PHP		08		3
104	68	PLA	Implied	PLA		68		4
40	28	PLP	Implied	PLP		28		4
46	2E	ROL	Absolute	ROL	\$FC58	2E	58 FC	6
62	3E	ROL	Absolute, X	ROL	\$D260, X	3E	60 D2	7
42	2A	ROL	Accumulator	ROL		2A		2
38	26	ROL	Zero Page	ROL	\$B7	26	B7	5
54	36	ROL	Zero Page, X	ROL	\$9D, X	36	9D	6
110	6E	ROR	Absolute	ROR	\$FC58	6E	58 FC	6
126	7E	ROR	Absolute, X	ROR	\$D260, X	7E	60 D2	7
106	6A	ROR	Accumulator	ROR		6A		2
102	66	ROR	Zero Page	ROR	\$B7	66	B7	5
118	76	ROR	Zero Page, X	ROR	\$9D, X	76	9D	6
64	40	RTI	Implied	RTI		40		6
96	60	RTS	Implied	RTS		60		6
237	ED	SBC	Absolute	SBC	\$FC58	ED	58 FC	4
253	FD	SBC	Absolute, X	SBC	\$D260, X	FD	60 D2	4 +1

MACHINE LANGUAG		ASSEMBLY LANGUAGE MNEMONIC & ALTERNATE		ASSEM LAN INSTRUCTION	MACH LAN INSTRUCT	MACH LAN TIMING (CYCLES)		
OPCODE		ADDRESSING MODES		SYNTAX	SYNTAX	FOR EA INST		CYC
DEC	HEX	OPCODE	ADRES MODE	OPCODE	OPERAND	OPCD	OPRND	
249	F9	SBC	Absolute, Y	SBC	\$C050, Y	F9	50 C0	4 +1
233	E9	SBC	Immediate	SBC	#\$2A	E9	2A	2
225	E1	SBC	(Indirect, X)	SBC	(\$E4, X)	E1	E4	6
241	F1	SBC	(Indirect), Y	SBC	(\$6), Y	F1	06	5 +1
229	E5	SBC	Zero Page	SBC	\$B7	E5	B7	3
245	F5	SBC	Zero Page, X	SBC	\$9D, X	F5	9D	4
56	38	SEC	Implied	SEC		38		2
248	F8	SED	Implied	SED		F8		2
120	78	SEI	Implied	SEI		78		2
141	8D	STA	Absolute	STA	\$FC58	8D	58 FC	4
157	9D	STA	Absolute, X	STA	\$D260, X	9D	60 D2	5
153	99	STA	Absolute, Y	STA	\$C050, Y	99	50 C0	5
129	81	STA	(Indirect, X)	STA	(\$E4, X)	81	E4	6
145	91	STA	(Indirect), Y	STA	(\$6), Y	91	06	6
133	85	STA	Zero Page	STA	\$B7	85	B7	3
149	95	STA	Zero Page, X	STA	\$9D, X	95	9D	4
142	8E	STX	Absolute	STX	\$FC58	8E	58 FC	4
134	86	STX	Zero Page	STX	\$B7	86	B7	3
150	96	STX	Zero Page, Y	STX	\$A3, Y	96	A3	4
140	8C	STY	Absolute	STY	\$FC58	8C	58 FC	4
132	84	STY	Zero Page	STY	\$B7	84	B7	3
148	94	STY	Zero Page, X	STY	\$9D, X	94	9D	4
170	AA	TAX	Implied	TAX		AA		2
168	A8	TAY	Implied	TAY		A8		2
186	BA	TSX	Implied	TSX		BA		2
138	8A	TXA	Implied	TXA		8A		2
154	9A	TXS	Implied	TXS		9A		2
152	98	TYA	Implied	TYA		98		2

OPCODE ADDRESSING INFORMATION

This chart is for assembly level programming where different addressing modes are employed. Using this chart, the programmer can quickly look up whether the opcode is 1, 2 or 3 bytes in a given addressing mode and the natures of the addressing. Both the decimal and hex codes are given for the machine language opcodes.

In disassembled listings, the hex opcodes are given next to an address, and this chart will show the user what byte pattern can be expected and memory used for every instruction. For example, the following segment uses 1, 2 and 3 byte opcodes:

```
LDA  #$14      (2 bytes)
PHA                      (1 byte )
JSR  $F8D0     (3 bytes)
```

Disassembled, the program would look like this:

```
300: A9 14      LDA    #$14
302: 48          PHA
303: 20 D0 F8    JSR    $F8D0
```

Examining the chart, we can see that LDA in the Immediate addressing mode is 2 bytes (1 opcode + 1 operand) with the machine opcode of A9. In the Absolute mode, on the other hand, the opcode LDA is a 3 byte instruction (1 opcode + 2 operand) with a machine opcode of AD instead of A9. Likewise, PHA in the Implied mode is a 1 byte instruction (1 opcode + 0 operand) and JSR in the Absolute mode is a 3 byte instruction (1 opcode + 2 operand.)

ASSEMBLY LANGUAGE OPCODE				ADDRESSING MODE INFORMATION
DEC	HEX	MNEMON		ADDRESSING
109	6D	ADC	Absolute	3-byte INSTRUCTION
45	2D	AND	Absolute	OPCODE + 2-byte OPERAND
14	E	ASL	Absolute	OPERAND forms an address in
44	2C	BIT	Absolute	low byte high byte format of
205	CD	CMP	Absolute	where data is to be found.
236	EC	CPX	Absolute	
204	CC	CPY	Absolute	
206	CE	DEC	Absolute	
77	4D	EOR	Absolute	
238	EE	INC	Absolute	
76	4C	JMP	Absolute	
32	20	JSR	Absolute	
173	AD	LDA	Absolute	
174	AE	LDX	Absolute	
172	AC	LDY	Absolute	

ASSEMBLY LANGUAGE OP CODE				ADDRESSING MODE INFORMATION
78	4E	LSR	Absolute	
13	D	ORA	Absolute	
46	2E	ROL	Absolute	
110	6E	ROR	Absolute	
237	ED	SBC	Absolute	
141	8D	STA	Absolute	
142	8E	STX	Absolute	
140	8C	STY	Absolute	
125	7D	ADC	Absolute,X	3-byte INSTRUCTION OPCODE + 2-byte OPERAND OPERAND forms an address in low byte high byte format to which the X-register is added. This computed address is where the data is to be found.
61	3D	AND	Absolute,X	
30	1E	ASL	Absolute,X	
221	DD	CMP	Absolute,X	
222	DE	DEC	Absolute,X	
93	5D	EOR	Absolute,X	
254	FE	INC	Absolute,X	
189	BD	LDA	Absolute,X	
188	BC	LDY	Absolute,X	
94	5E	LSR	Absolute,X	
29	1D	ORA	Absolute,X	
62	3E	ROL	Absolute,X	
126	7E	ROR	Absolute,X	
253	FD	SBC	Absolute,X	
157	9D	STA	Absolute,X	
121	79	ADC	Absolute,Y	3-byte INSTRUCTION OPCODE + 2-byte OPERAND OPERAND forms an address in low byte high byte format to which the Y-register is added. This computed address is where the data is to be found.
57	39	AND	Absolute,Y	
217	D9	CMP	Absolute,Y	
89	59	EOR	Absolute,Y	
185	B9	LDA	Absolute,Y	
190	BE	LDX	Absolute,Y	
25	19	ORA	Absolute,Y	
249	F9	SBC	Absolute,Y	
153	99	STA	Absolute,Y	
10	A	ASL	Accumulator	1-byte INSTRUCTION OPCODE (no OPERAND)
74	4A	LSR	Accumulator	
42	2A	ROL	Accumulator	
106	6A	ROR	Accumulator	
105	69	ADC	Immediate	2-byte INSTRUCTION OPCODE + 1-byte OPERAND OPERAND is the data.
41	29	AND	Immediate	
201	C9	CMP	Immediate	
224	E0	CPX	Immediate	
192	C0	CPY	Immediate	
73	49	EOR	Immediate	
169	A9	LDA	Immediate	
162	A2	LDX	Immediate	
160	A0	LDY	Immediate	

ASSEMBLY LANGUAGE OPCODE				ADDRESSING MODE INFORMATION
9	9	ORA	Immediate	
233	E9	SBC	Immediate	
0	0	BRK	Implied	1-byte INSTRUCTION
24	18	CLC	Implied	OPCODE (no OPERAND)
216	D8	CLD	Implied	
88	58	CLI	Implied	
184	B8	CLV	Implied	
202	CA	DEX	Implied	
136	88	DEY	Implied	
232	E8	INX	Implied	
200	C8	INY	Implied	
234	EA	NOP	Implied	
72	48	PHA	Implied	
8	8	PHP	Implied	
104	68	PLA	Implied	
40	28	PLP	Implied	
64	40	RTI	Implied	
96	60	RTS	Implied	
56	38	SEC	Implied	
248	F8	SED	Implied	
120	78	SEI	Implied	
170	AA	TAX	Implied	
168	A8	TAY	Implied	
186	BA	TSX	Implied	
138	8A	TXA	Implied	
154	9A	TXS	Implied	
152	98	TYA	Implied	
108	6C	JMP	(Indirect)	3-byte INSTRUCTION OPCODE + 2-byte OPERAND OPERAND forms an address in low byte high byte format. This address (pointer) and the next address location hold the address where the program execution will continue.
97	61	ADC	(Indirect,X)	2-byte INSTRUCTION
33	21	AND	(Indirect,X)	OPCODE + 1-byte OPERAND
193	C1	CMP	(Indirect,X)	OPERAND forms a zero page address (pointer) to which the X-register is added.
65	41	EOR	(Indirect,X)	This computed address and the next zero page location hold the address to where the data is to be found.
161	A1	LDA	(Indirect,X)	
1	1	ORA	(Indirect,X)	
225	E1	SBC	(Indirect,x)	
129	81	STA	(Indirect,X)	

ASSEMBLY LANGUAGE OPCODE				ADDRESSING MODE INFORMATION
113	71	ADC	(Indirect),Y	2-byte INSTRUCTION
49	31	AND	(Indirect),Y	OPCODE + 1-byte OPERAND
209	D1	CMP	(Indirect),Y	OPERAND forms a zero page
81	51	EOR	(Indirect),Y	address (pointer) . This
177	B1	LDA	(Indirect),Y	address and the next zero page
17	11	ORA	(Indirect),Y	location hold an address to
241	F1	SBC	(Indirect),Y	which the Y-register is added.
145	91	STA	(Indirect),Y	This computed address is where the data is to be found.
144	90	BCC	Relative	2-byte INSTRUCTION
176	B0	BCS	Relative	OPCODE + 1-byte OPERAND
240	F0	BEQ	Relative	OPERAND is a signed offset
48	30	BMI	Relative	which is added/subtracted
208	D0	BNE	Relative	from the Program Counter.
16	10	BPL	Relative	If the branch is taken then the
80	50	BVC	Relative	6502 Microprocessor begins
112	70	BVS	Relative	executing code at this computed address.
101	65	ADC	Zero Page	2-byte INSTRUCTION
37	25	AND	Zero Page	OPCODE + 1-byte OPERAND
6	6	ASL	Zero Page	OPERAND forms a zero page address
36	24	BIT	Zero Page	where the data is to be found.
197	C5	CMP	Zero Page	
228	E4	CPX	Zero Page	
196	C4	CPY	Zero Page	
198	C6	DEC	Zero Page	
69	45	EOR	Zero Page	
230	E6	INC	Zero Page	
165	A5	LDA	Zero Page	
166	A6	LDX	Zero Page	
164	A4	LDY	Zero Page	
70	46	LSR	Zero Page	
5	5	ORA	Zero Page	
38	26	ROL	Zero Page	
102	66	ROR	Zero Page	
229	E5	SBC	Zero Page	
133	85	STA	Zero Page	
134	86	STX	Zero Page	
132	84	STY	Zero Page	
117	75	ADC	Zero Page,X	2-byte INSTRUCTION
53	35	AND	Zero Page,X	OPCODE + 1-byte OPERAND
22	16	ASL	Zero Page,X	OPERAND forms a zero page address
213	D5	CMP	Zero Page,X	to which the X-register is added.
214	D6	DEC	Zero Page,X	This computed address is the
85	55	EOR	Zero Page,X	zero page location of the data.
246	F6	INC	Zero Page,X	

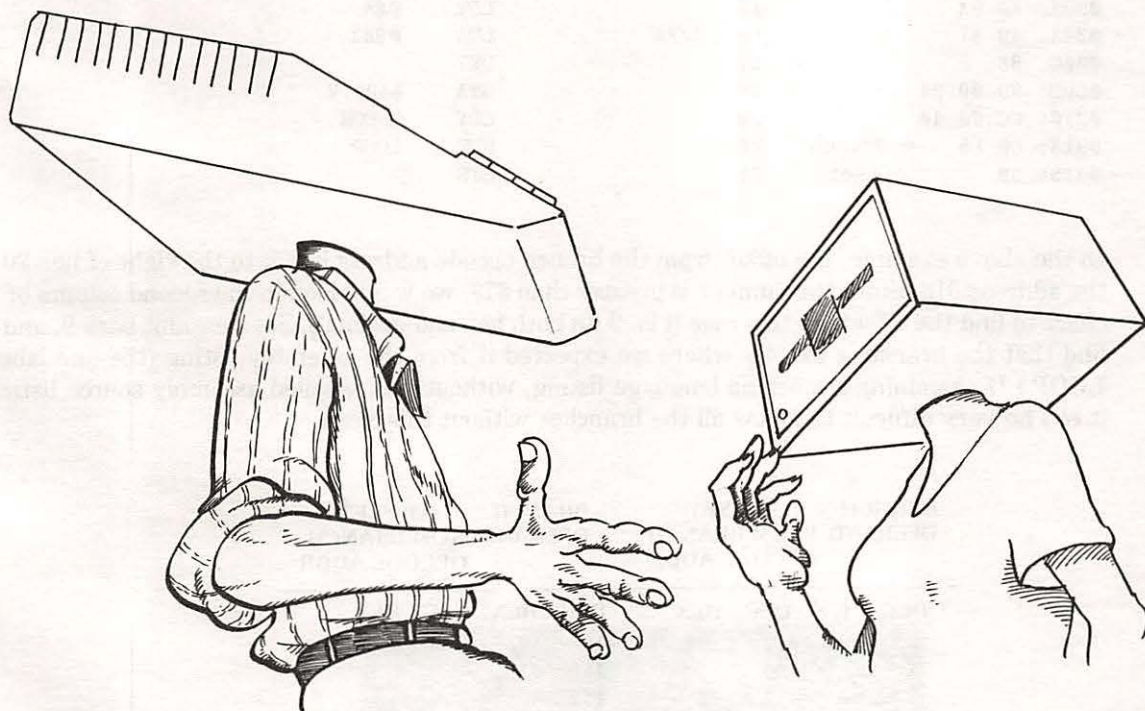
ASSEMBLY LANGUAGE OPCODE

ADDRESSING MODE INFORMATION

181	B5	LDA	Zero Page, X
180	B4	LDY	Zero Page, X
86	56	LSR	Zero Page, X
21	15	ORA	Zero Page, X
54	36	ROL	Zero Page, X
118	76	ROR	Zero Page, X
245	F5	SBC	Zero Page, X
149	95	STA	Zero Page, X
148	94	STY	Zero Page, X

182	B6	LDX	Zero Page, Y
150	96	STX	Zero Page, Y

2-byte INSTRUCTION
 OPCODE + 1-byte OPERAND
 OPERAND forms a zero page address
 to which the Y-register is added.
 This computed address is the
 zero page location of the data.



BRANCH OPERAND OFFSETS CHART

This chart is used to determine forward and backward branches in machine/assembly level programming. Usually in assembly level programming, a label is used with a branch within the program (e.g. BNE LOOP) with the label being an address locator. However, there is a maximum forward and backward branch range available, and using this chart, the programmer can examine those limitations. Likewise, it is possible to see the offset from the branch opcode address so that branch ranges can be examined in machine code listings to find branch locations. The following example will be used to explain how to use the chart. (In case you're interested, all this program does is to print ten A's in the upper right hand corner of the screen.)

	1	*****		
	2	* BRANCH *		
	3	*****		
	4			
	5		OBJ	\$300
	6		ORG	\$300
	7			
	8	HOME	EQU	\$FC58
	9	CHECK	EQU	\$4000
	10			
	11			
0300: A9 00	12		LDA	#\$0
0302: 8D 00 40	13		STA	CHECK
0305: 20 58 FC	14		JSR	HOME
0308: A0 0A	15		LDY	#\$A
030A: A9 81	16	LOOP	LDA	#\$81
030C: 88	17		DEY	
030D: 99 00 04	18		STA	\$400, Y
0310: CC 00 40	19		CPY	CHECK
0313: D0 F5 ← Branch	20		BNE	LOOP
0315: 60 offset	21		RTS	

In the above example, the offset from the branch opcode address is F5, to the right of line 20, at the address 313. Since the number is greater than \$79, we would look in the second column of the chart to find the offset. In this case it is -9 (in both hex and decimal). So, we count back 9, and we find that the branch is exactly where we expected it from the assembly listing (the line labeled LOOP.) If examining a machine language listing, without well labelled assembly source listings, it can be very difficult to follow all the branches without this chart.

BRANCH OPERAND		OFFSET FROM BRANCH OPCODE ADDR		BRANCH OPERAND		OFFSET FROM BRANCH OPCODE ADDR	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
0	0	2	2	128	80	-126	-7E
1	1	3	3	129	81	-125	-7D
2	2	4	4	130	82	-124	-7C
3	3	5	5	131	83	-123	-7B
4	4	6	6	132	84	-122	-7A
5	5	7	7	133	85	-121	-79
6	6	8	8	134	86	-120	-78

BRANCH OPERAND	OFFSET FROM BRANCH OPCODE ADDR	BRANCH OPERAND	OFFSET FROM BRANCH OPCODE ADDR
-------------------	--------------------------------------	-------------------	--------------------------------------

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
7	7	9	9	135	87	-119	-77
8	8	10	A	136	88	-118	-76
9	9	11	B	137	89	-117	-75
10	A	12	C	138	8A	-116	-74
11	B	13	D	139	8B	-115	-73
12	C	14	E	140	8C	-114	-72
13	D	15	F	141	8D	-113	-71
14	E	16	10	142	8E	-112	-70
15	F	17	11	143	8F	-111	-6F
16	10	18	12	144	90	-110	-6E
17	11	19	13	145	91	-109	-6D
18	12	20	14	146	92	-108	-6C
19	13	21	15	147	93	-107	-6B
20	14	22	16	148	94	-106	-6A
21	15	23	17	149	95	-105	-69
22	16	24	18	150	96	-104	-68
23	17	25	19	151	97	-103	-67
24	18	26	1A	152	98	-102	-66
25	19	27	1B	153	99	-101	-65
26	1A	28	1C	154	9A	-100	-64
27	1B	29	1D	155	9B	-99	-63
28	1C	30	1E	156	9C	-98	-62
29	1D	31	1F	157	9D	-97	-61
30	1E	32	20	158	9E	-96	-60
31	1F	33	21	159	9F	-95	-5F
32	20	34	22	160	A0	-94	-5E
33	21	35	23	161	A1	-93	-5D
34	22	36	24	162	A2	-92	-5C
35	23	37	25	163	A3	-91	-5B
36	24	38	26	164	A4	-90	-5A
37	25	39	27	165	A5	-89	-59
38	26	40	28	166	A6	-88	-58
39	27	41	29	167	A7	-87	-57
40	28	42	2A	168	A8	-86	-56
41	29	43	2B	169	A9	-85	-55
42	2A	44	2C	170	AA	-84	-54
43	2B	45	2D	171	AB	-83	-53
44	2C	46	2E	172	AC	-82	-52
45	2D	47	2F	173	AD	-81	-51
46	2E	48	30	174	AE	-80	-50
47	2F	49	31	175	AF	-79	-4F
48	30	50	32	176	B0	-78	-4E
49	31	51	33	177	B1	-77	-4D
50	32	52	34	178	B2	-76	-4C
51	33	53	35	179	B3	-75	-4B
52	34	54	36	180	B4	-74	-4A
53	35	55	37	181	B5	-73	-49
54	36	56	38	182	B6	-72	-48
55	37	57	39	183	B7	-71	-47
56	38	58	3A	184	B8	-70	-46
57	39	59	3B	185	B9	-69	-45
58	3A	60	3C	186	BA	-68	-44

BRANCH OPERAND		OFFSET FROM BRANCH OPCODE ADDR		BRANCH OPERAND		OFFSET FROM BRANCH OPCODE ADDR	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
59	3B	61	3D	187	BB	-67	-43
60	3C	62	3E	188	BC	-66	-42
61	3D	63	3F	189	BD	-65	-41
62	3E	64	40	190	BE	-64	-40
63	3F	65	41	191	BF	-63	-3F
64	40	66	42	192	C0	-62	-3E
65	41	67	43	193	C1	-61	-3D
66	42	68	44	194	C2	-60	-3C
67	43	69	45	195	C3	-59	-3B
68	44	70	46	196	C4	-58	-3A
69	45	71	47	197	C5	-57	-39
70	46	72	48	198	C6	-56	-38
71	47	73	49	199	C7	-55	-37
72	48	74	4A	200	C8	-54	-36
73	49	75	4B	201	C9	-53	-35
74	4A	76	4C	202	CA	-52	-34
75	4B	77	4D	203	CB	-51	-33
76	4C	78	4E	204	CC	-50	-32
77	4D	79	4F	205	CD	-49	-31
78	4E	80	50	206	CE	-48	-30
79	4F	81	51	207	CF	-47	-2F
80	50	82	52	208	D0	-46	-2E
81	51	83	53	209	D1	-45	-2D
82	52	84	54	210	D2	-44	-2C
83	53	85	55	211	D3	-43	-2B
84	54	86	56	212	D4	-42	-2A
85	55	87	57	213	D5	-41	-29
86	56	88	58	214	D6	-40	-28
87	57	89	59	215	D7	-39	-27
88	58	90	5A	216	D8	-38	-26
89	59	91	5B	217	D9	-37	-25
90	5A	92	5C	218	DA	-36	-24
91	5B	93	5D	219	DB	-35	-23
92	5C	94	5E	220	DC	-34	-22
93	5D	95	5F	221	DD	-33	-21
94	5E	96	60	222	DE	-32	-20
95	5F	97	61	223	DF	-31	-1F
96	60	98	62	224	E0	-30	-1E
97	61	99	63	225	E1	-29	-1D
98	62	100	64	226	E2	-28	-1C
99	63	101	65	227	E3	-27	-1B
100	64	102	66	228	E4	-26	-1A
101	65	103	67	229	E5	-25	-19
102	66	104	68	230	E6	-24	-18
103	67	105	69	231	E7	-23	-17
104	68	106	6A	232	E8	-22	-16
105	69	107	6B	233	E9	-21	-15
106	6A	108	6C	234	EA	-20	-14
107	6B	109	6D	235	EB	-19	-13
108	6C	110	6E	236	EC	-18	-12
109	6D	111	6F	237	ED	-17	-11

BRANCH OPERAND		OFFSET FROM BRANCH OPCODE ADDR		BRANCH OPERAND		OFFSET FROM BRANCH OPCODE ADDR	
DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
110	6E	112	70	238	EE	-16	-10
111	6F	113	71	239	EF	-15	-F
112	70	114	72	240	F0	-14	-E
113	71	115	73	241	F1	-13	-D
114	72	116	74	242	F2	-12	-C
115	73	117	75	243	F3	-11	-B
116	74	118	76	244	F4	-10	-A
117	75	119	77	245	F5	-9	-9
118	76	120	78	246	F6	-8	-8
119	77	121	79	247	F7	-7	-7
120	78	122	7A	248	F8	-6	-6
121	79	123	7B	249	F9	-5	-5
122	7A	124	7C	250	FA	-4	-4
123	7B	125	7D	251	FB	-3	-3
124	7C	126	7E	252	FC	-2	-2
125	7D	127	7F	253	FD	-1	-1
126	7E	128	80	254	FE	0	0
127	7F	129	81	255	FF	1	1

6502 OPCODE MNEMONIC CHART

This chart is useful for a quick look-up of the different meanings of 6502 opcodes. Other charts in this book provide greater detail regarding tokens, flags set, etc. for 6502 opcodes. However, for a general reference for the assembly level programmer, this will be very handy. (NOTE: See also 65C02 OPCODES)

MNEMON TRANSLATION		MNEMON TRANSLATION	
ADC	Add Memory to Carry to A-reg	LDA	Load A-reg from Memory
AND	Logical AND memory to A-reg	LDX	Load X-reg from Memory
ASL	Arithmetic Shift bits Left	LDY	Load Y-reg from Memory
		LSR	Logical Shift bits Right
BCC	Branch if Carry Clear		
BCS	Branch if Carry Set	NOP	No Operation
BEQ	Branch if Equal Zero		
BIT	Test A-reg bits with memory	ORA	Logical OR Memory and A-reg
BMI	Branch if Minus		
BNE	Branch if Not Equal to zero	PHA	Push A-reg onto Stack
BPL	Branch if Plus	PHP	Push P-reg onto Stack
BRK	Software Interrupt - BREAK	PLA	Pull A-reg from Stack
BVC	Branch if Overflow Clear	PLP	Pull P-reg from Stack
BVS	Branch if Overflow Set		
		ROL	Rotate bits Left
CLC	Clear Carry flag	ROR	Rotate bits Right
CLD	Clear Decimal flag	RTI	Return from Interrupt
CLI	Clear Interrupt flag	RTS	Return from Subroutine
CLV	Clear Overflow flag		
CMP	Compare Memory and A-reg	SBC	Subtract Memory from A-reg
CPX	Compare Memory and X-reg	SEC	Set Carry flag
CPY	Compare Memory and Y-reg	SED	Set Decimal flag
		SEI	Set Interrupt flag
DEC	Decrement Memory by 1	STA	Store A-reg into Memory
DEX	Decrement X-reg by 1	STX	Store X-reg into Memory
DEY	Decrement Y-reg by 1	STY	Store Y-reg into Memory
EOR	Exclusive OR Memory and A-reg	TAX	Transfer A-reg into X-reg
		TAY	Transfer A-reg into Y-reg
INC	Increment Memory by 1	TSX	Transfer S-reg into X-reg
INX	Increment X-reg by 1	TXA	Transfer X-reg into A-reg
INY	Increment Y-reg by 1	TXS	Transfer X-reg into S-reg
		TYA	Transfer Y-reg into A-reg
JMP	Jump PC-reg to new address		
JSR	Jump but Save Return address		

*R65C02 ADDED MNEUMONICS AND ADDRESSING MODES

The new Rockwell R65C02 microprocessor can be used in the Apple simply by replacing the current 6502 chip with the R65C02. All opcodes in the R65C02 are compatible with those in the 6502, but some new ones have been added. The added mnemonics are listed in the first of the following two charts. In the second chart, the new addressing modes are presented. For those programmers interested in purchasing the new microprocessor or for those who have one, these charts will be necessary for the transition if the user wishes to access the added features of the R65C02.

**R65C02 is Rockwell's designation for the microprocessor chip, but other manufacturers with their own chip designation are expected to be producing essentially the same chip microprocessor.*

R65C02 ADDED OPCODE MNEUMONICS CHART

BBR	Branch on Bit Reset
BBS	Branch on Bit Set
BRA	Branch Always
PHX	Push X-reg onto Stack
PHY	Push Y-reg onto Stack
PLX	Pull X-reg from Stack
PLY	Pull Y-reg from Stack
RMB	Reset Memory Bit
SMB	Set Memory Bit
STZ	Store Zero
TRB	Test and Reset Bits
TSB	Test and Set Bits

NEW ADDRESSING MODES

INDEXED ABSOLUTE INDIRECT

" " 3-byte INSTRUCTION OPCODE + 2-byte OPERAND. OPERAND forms an address in low byte high byte format to which the X-register is added. This computed address and the next address location hold the address to where the data is to be found.

ZERO PAGE INDIRECT

2-byte INSTRUCTION OPCODE + 1-byte OPERAND OPERAND forms a zero page address (pointer). This address and the next zero page address hold the address to where the data is to be found.

(IN CURRENT LITERATURE PLEASE NOTE:)

The original INDIRECT addressing mode is being referred to as ABSOLUTE INDIRECT. And the new addressing mode ZERO PAGE INDIRECT is being referred to as INDIRECT. This convention will unfortunately confuse many.

R65C02 OPCODES

The following two charts provide a numeric and alphabetic listing of the opcodes added on the R65C02 microprocessor. Compare the first chart with the 6502 OPCODES chart, and you will notice that several blank areas are now used for the new opcodes. For those users putting the R65C02 chip in their Apple, these charts will provide handy reference.

R65C02 NEW OPCODES (NUMERIC LISTING)

MACH LAN ASSEMBLY LANGUAGE
OPCODE OPCODE

HEX DEC MNEMON ADDRESSING MODE

4	4	TSB	Zero Page
7	7	RMB0	Zero Page
12	C	TSB	Absolute
15	F	BBR0	Zero Page
18	12	ORA	(Zero Page)
20	14	TRB	Zero Page
23	17	RMB1	Zero Page
26	1A	INC	Accumulator
28	1C	TRB	Absolute
31	1F	BBR1	Zero Page
39	27	RMB2	Zero Page
47	2F	BBR2	Zero Page
50	32	AND	(Zero Page)
52	34	BIT	Zero Page, X
55	37	RMB3	Zero Page
58	3A	DEC	Accumulator
60	3C	BIT	Absolute, X
63	3F	BBR3	Zero Page
71	47	RMB4	Zero Page
79	4F	BBR4	Zero Page
82	52	EOR	(Zero Page)
87	57	RMB5	Zero Page
90	5A	PHY	Implied
95	5F	BBR5	Zero Page
100	64	STZ	Zero Page
103	67	RMB6	Zero Page
111	6F	BBR6	Zero Page
114	72	ADC	(Zero Page)
116	74	STZ	Zero Page, X
119	77	RMB7	Zero Page
122	7A	PLY	Implied
124	7C	JMP	(Absolute, X)
127	7F	BBR7	Zero Page
128	80	BRA	Relative
135	87	SMB0	Zero Page

MACH LAN ASSEMBLY LANGUAGE
OPCODE OPCODE

137	89	BIT	Immediate
143	8F	BBS0	Zero Page
146	92	STA	(Zero Page)
151	97	SMB1	Zero Page
156	9C	STZ	Absolute
158	9E	STZ	Absolute, X
159	9F	BBS1	Zero Page
167	A7	SMB2	Zero Page
175	AF	BBS2	Zero Page
178	B2	LDA	(Zero Page)
183	B7	SMB3	Zero Page
191	BF	BBS3	Zero Page
199	C7	SMB4	Zero Page
207	CF	BBS4	Zero Page
210	D2	CMP	(Zero Page)
215	D7	SMB5	Zero Page
218	DA	PHX	Implied
223	DF	BBS5	Zero Page
231	E7	SMB6	Zero Page
239	EF	BBS6	Zero Page
242	F2	SBC	(Zero Page)
247	F7	SMB7	Zero Page
250	FA	PLX	Implied
255	FF	BBS7	Zero Page

R65C02 OPCODES (ALPHABETIC LISTING)

MACH LAN ASSEMBLY LANGUAGE
OPCODE OPCODE

HEX	DEC	MNEMON	ADDRESSING MODE
114	72	ADC	(Zero Page)
50	32	AND	(Zero Page)
15	F	BBR0	Zero Page
31	1F	BBR1	Zero Page
47	2F	BBR2	Zero Page
63	3F	BBR3	Zero Page
79	4F	BBR4	Zero Page
95	5F	BBR5	Zero Page
111	6F	BBR6	Zero Page
127	7F	BBR7	Zero Page
143	8F	BBS0	Zero Page
159	9F	BBS1	Zero Page
175	AF	BBS2	Zero Page
191	BF	BBS3	Zero Page
207	CF	BBS4	Zero Page
223	DF	BBS5	Zero Page
239	EF	BBS6	Zero Page
255	FF	BBS7	Zero Page

MACH LAN ASSEMBLY LANGUAGE
OPCODE OPCODE

HEX	DEC	MNEMON	ADDRESSING MODE
60	3C	BIT	Absolute,X
137	89	BIT	Immediate
52	34	BIT	Zero Page,X
128	80	BRA	Relative
210	D2	CMP	(Zero Page)
58	3A	DEC	Accumulator
82	52	EOR	(Zero Page)
26	1A	INC	Accumulator
124	7C	JMP	(Absolute,X)
178	B2	LDA	(Zero Page)
18	12	ORA	(Zero Page)
218	DA	PHX	Implied
90	5A	PHY	Implied
250	FA	PLX	Implied
122	7A	PLY	Implied
7	7	RMB0	Zero Page
23	17	RMB1	Zero Page
39	27	RMB2	Zero Page
55	37	RMB3	Zero Page
71	47	RMB4	Zero Page
87	57	RMB5	Zero Page
103	67	RMB6	Zero Page
119	77	RMB7	Zero Page
242	F2	SBC	(Zero Page)
135	87	SMB0	Zero Page
151	97	SMB1	Zero Page
167	A7	SMB2	Zero Page
183	B7	SMB3	Zero Page
199	C7	SMB4	Zero Page
215	D7	SMB5	Zero Page
231	E7	SMB6	Zero Page
247	F7	SMB7	Zero Page
146	92	STA	(Zero Page)
156	9C	STZ	Absolute
158	9E	STZ	Absolute,X
100	64	STZ	Zero Page
116	74	STZ	Zero Page,X

MACH LAN		ASSEMBLY LANGUAGE	
OPCODE		OPCODE	
HEX	DEC	MNEMON	ADDRESSING MODE
28	1C	TRB	Absolute
20	14	TRB	Zero Page
12	C	TSB	Absolute
4	4	TSB	Zero Page



CONDITIONED FLAGS/REGISTERS/MEMORY CHART

This chart shows which flags are affected by different opcodes in machine/assembly level programming. Additionally, it shows which registers are affected and any modifications in memory.

The left column shows the opcodes in machine language, both decimal and hex, and assembly language, along with the addressing mode. The middle column shows the flags affected in the process status register using the following abbreviations:

N - Negative V - Overflow B - Break Command I - Interrupt disable
Z - Zero C - Carry.

The right column shows whether the accumulator (A), X or Y register (X, Y), processor status register (P) and stack pointer (S) along with the program counter (PC) are modified by the command. Memory modifications are indicated by MEM.

There are three hardware lines (interrupts) which also cause alterations:

<u>INTERRUPT</u>	<u>JUMP THROUGH</u>
NMI	\$FFFA & \$FFFB
RESET	\$FFFC & \$FFF D
IRQ	\$FFFE & \$FFFF

An NMI saves the PCH, the PCL and the P-register to the stack and then makes the above jump. An IRQ saves the PCH , PCL and the P-register to the stack and then makes the jump, but it sets the I flag and the break flag is cleared. RESET saves nothing, but it does take control from anything else before its jump.

MACHINE		ASSEMBLY LANGUAGE		CONDITIONED	KNOWN FLAG	CONDITIONED	
LAN OPCD		OPCODES & ADDRES.		FLAGS	VALUES	REGISTERS & MEMORY	
DEC	HEX	MNEMON	ADRS MODE			6502	MEM
109	6D	ADC	Absolute	NV...ZC		A	P
125	7D	ADC	Absolute,X	NV...ZC		A	P
121	79	ADC	Absolute,Y	NV...ZC		A	P
105	69	ADC	Immediate	NV...ZC		A	P
97	61	ADC	(Indirect,X)	NV...ZC		A	P
113	71	ADC	(Indirect),Y	NV...ZC		A	P
101	65	ADC	Zero Page	NV...ZC		A	P
117	75	ADC	Zero Page,X	NV...ZC		A	P
45	2D	AND	Absolute	N....Z.		A	P
61	3D	AND	Absolute,X	N....Z.		A	P
57	39	AND	Absolute,Y	N....Z.		A	P
41	29	AND	Immediate	N....Z.		A	P
33	21	AND	(Indirect,X)	N....Z.		A	P
49	31	AND	(Indirect),Y	N....Z.		A	P
37	25	AND	Zero Page	N....Z.		A	P
53	35	AND	Zero Page,X	N....Z.		A	P

MACHINE LAN	OPCD	ASSEMBLY LANGUAGE OPCODES & ADDRES.	CONDITIONED FLAGS	KNOWN FLAG VALUES	CONDITIONED REGISTERS & MEMORY
14	E	ASL Absolute	N....ZC		P MEM
30	1E	ASL Absolute,X	N....ZC		P MEM
10	A	ASL Accumulator	N....ZC	A	P
6	6	ASL Zero Page	N....ZC		P MEM
22	16	ASL Zero Page,X	N....ZC		P MEM
144	90	BCC Relative			PC
176	B0	BCS Relative			PC
240	F0	BEQ Relative			PC
44	2C	BIT Absolute	NV....Z.	N=bit7 V=bit6	P
36	24	BIT Zero Page	NV....Z.	N=bit7 V=bit6	P
48	30	BMI Relative			PC
208	D0	BNE Relative			PC
16	10	BPL Relative			PC
0	0	BRK Implied	...B.I..	B=1 I=1	P S PC
80	50	BVC Relative			PC
112	70	BVS Relative			PC
24	18	CLC Implied	C	C=0	P
216	D8	CLD Implied	D...	D=0	P
88	58	CLI Implied	I..	I=0	P
184	B8	CLV Implied	.V.....	V=0	P
205	CD	CMP Absolute	N....ZC		P
221	DD	CMP Absolute,X	N....ZC		P
217	D9	CMP Absolute,Y	N....ZC		P
201	C9	CMP Immediate	N....ZC		P
193	C1	CMP (Indirect,X)	N....ZC		P
209	D1	CMP (Indirect),Y	N....ZC		P
197	C5	CMP Zero Page	N....ZC		P
213	D5	CMP Zero Page,X	N....ZC		P
236	EC	CPX Absolute	N....ZC		P
224	E0	CPX Immediate	N....ZC		P
228	E4	CPX Zero Page	N....ZC		P
204	CC	CPY Absolute	N....ZC		P
192	C0	CPY Immediate	N....ZC		P
196	C4	CPY Zero Page	N....ZC		P

MACHINE LAN	OPCD	ASSEMBLY LANGUAGE OPCODES & ADDRES.	CONDITIONED FLAGS	KNOWN FLAG VALUES	CONDITIONED REGISTERS & MEMORY
206	CE	DEC	Absolute	N....Z.	P MEM
222	DE	DEC	Absolute,X	N....Z.	P MEM
198	C6	DEC	Zero Page	N....Z.	P MEM
214	D6	DEC	Zero Page,X	N....Z.	P MEM
202	CA	DEX	Implied	N....Z.	X P
136	88	DEY	Implied	N....Z.	Y P
77	4D	EOR	Absolute	N....Z.	A P
93	5D	EOR	Absolute,X	N....Z.	A P
89	59	EOR	Absolute,Y	N....Z.	A P
73	49	EOR	Immediate	N....Z.	A P
65	41	EOR	(Indirect,X)	N....Z.	A P
81	51	EOR	(Indirect),Y	N....Z.	A P
69	45	EOR	Zero Page	N....Z.	A P
85	55	EOR	Zero Page,X	N....Z.	A P
238	EE	INC	Absolute	N....Z.	P MEM
254	FE	INC	Absolute,X	N....Z.	P MEM
230	E6	INC	Zero Page	N....Z.	P MEM
246	F6	INC	Zero Page,X	N....Z.	P MEM
232	E8	INX	Implied	N....Z.	X P
200	C8	INY	Implied	N....Z.	Y P
76	4C	JMP	Absolute		PC
108	6C	JMP	(Indirect)		PC
32	20	JSR	Absolute		S PC MEM
173	AD	LDA	Absolute	N....Z.	A P
189	BD	LDA	Absolute,X	N....Z.	A P
185	B9	LDA	Absolute,Y	N....Z.	A P
169	A9	LDA	Immediate	N....Z.	A P
161	A1	LDA	(Indirect,X)	N....Z.	A P
177	B1	LDA	(Indirect),Y	N....Z.	A P
165	A5	LDA	Zero Page	N....Z.	A P
181	B5	LDA	Zero Page,X	N....Z.	A P
174	AE	LDX	Absolute	N....Z.	X P
190	BE	LDX	Absolute,Y	N....Z.	X P
162	A2	LDX	Immediate	N....Z.	X P
166	A6	LDX	Zero Page	N....Z.	X P
182	B6	LDX	Zero Page,Y	N....Z.	X P
172	AC	LDY	Absolute	N....Z.	Y P
188	BC	LDY	Absolute,X	N....Z.	Y P
160	A0	LDY	Immediate	N....Z.	Y P
164	A4	LDY	Zero Page	N....Z.	Y P
180	B4	LDY	Zero Page,X	N....Z.	Y P

MACHINE LAN OPCODE	ASSEMBLY LANGUAGE OPCODES & ADDRES.	CONDITIONED FLAGS	KNOWN FLAG VALUES	CONDITIONED REGISTERS & MEMORY
78 4E	LSR Absolute	N....ZC	N=0	P MEM
94 5E	LSR Absolute,X	N....ZC	N=0	P MEM
74 4A	LSR Accumulator	N....ZC	N=0	A P
70 46	LSR Zero Page	N....ZC	N=0	P MEM
86 56	LSR Zero Page,X	N....ZC	N=0	P MEM
234 EA	NOP Implied			
13 D	ORA Absolute	N....Z.		A P
29 1D	ORA Absolute,X	N....Z.		A P
25 19	ORA Absolute,Y	N....Z.		A P
9 9	ORA Immediate	N....Z.		A P
1 1	ORA (Indirect,X)	N....Z.		A P
17 11	ORA (Indirect),Y	N....Z.		A P
5 5	ORA Zero Page	N....Z.		A P
21 15	ORA Zero Page,X	N....Z.		A P
72 48	PHA Implied			S MEM
8 8	PHP Implied			S MEM
104 68	PLA Implied	N....Z.		A P S
40 28	PLP Implied	NV+BDIZC		P S
46 2E	ROL Absolute	N....ZC		P MEM
62 3E	ROL Absolute,X	N....ZC		P MEM
42 2A	ROL Accumulator	N....ZC		A P
38 26	ROL Zero Page	N....ZC		P MEM
54 36	ROL Zero Page,X	N....ZC		P MEM
110 6E	ROR Absolute	N....ZC		P MEM
126 7E	ROR Absolute,X	N....ZC		P MEM
106 6A	ROR Accumulator	N....ZC		A P
102 66	ROR Zero Page	N....ZC		P MEM
118 76	ROR Zero Page,X	N....ZC		P MEM
64 40	RTI Implied	NV+BDIZC		P S PC
96 60	RTS Implied			S PC
237 ED	SBC Absolute	NV....ZC		A P
253 FD	SBC Absolute,X	NV....ZC		A P
249 F9	SBC Absolute,Y	NV....ZC		A P
233 E9	SBC Immediate	NV....ZC		A P
225 E1	SBC (Indirect,X)	NV....ZC		A P
241 F1	SBC (Indirect),Y	NV....ZC		A P
229 E5	SBC Zero Page	NV....ZC		A P
245 F5	SBC Zero Page,X	NV....ZC		A P
56 38	SEC Implied	C	C=1	P

MACHINE LAN	OPCD	ASSEMBLY LANGUAGE OPCODES & ADDRES.		CONDITIONED FLAGS	KNOWN FLAG VALUES	CONDITIONED REGISTERS & MEMORY	
248	F8	SED	Implied	D...	D=1	P	
120	78	SEI	Implied	I..	I=1	P	
141	8D	STA	Absolute				MEM
157	9D	STA	Absolute,X				MEM
153	99	STA	Absolute,Y				MEM
129	81	STA	(Indirect,X)				MEM
145	91	STA	(Indirect),Y				MEM
133	85	STA	Zero Page				MEM
149	95	STA	Zero Page,X				MEM
142	8E	STX	Absolute				MEM
134	86	STX	Zero Page				MEM
150	96	STX	Zero Page,Y				MEM
140	8C	STY	Absolute				MEM
132	84	STY	Zero Page				MEM
148	94	STY	Zero Page,X				MEM
170	AA	TAX	Implied	N....Z.		X P	
168	A8	TAY	Implied	N....Z.		Y P	
186	BA	TSX	Implied	N....Z.		X P	
138	8A	TXA	Implied	N....Z.		A P	
154	9A	TXS	Implied				S
152	98	TYA	Implied	N....Z.		A P	

APPLESOFT BASIC ASSEMBLY SYMBOL CHART

This chart represents an attempt to standardize address labels on the Apple. The values and labels in this chart are from Applesoft source code, generated on the Merlin Assembler by Southwestern Data System from Glen Bredon's interpretation of Applesoft code. While any label would work with an address, these labels are widely used and most easily read by programmers. For example, HOME is the label for address \$FC58 in the monitor source code(decimal 64,600 or -936). A JSR to that location or HOME or CALL -936 from BASIC all clear the screen and home the cursor. Using the HOME label for \$FC58 instead of an alternative, such as CLEAR, CLS or some other non-standard label, provides users with a way of easily communicating what they are doing to others and a reminder to themselves of what their own programs are doing.

The left side of the chart has the labels listed in order by address, and the right side of the chart has an alphabetical listing by label. When using an address and looking for a label, use the left side of the chart, and when searching for an address given a label, use the right side. The "REF" column provides space for the user to note page numbers elsewhere in this book for more detailed descriptions of subroutines associated with a label. The "Special Label Function" column shows locations/labels which are Macro-Definitions (MD) and where helpful or informative labels in Applesoft source code listings exist a question mark (?) is used. The "M" refers to a label used within a Macro-definition. This label will take on a different value each time the Macro-Definition is used, with the last address value being shown in the symbol table. (The entire symbol table was reproduced even though many of these addresses have little value to most programmers.)

SP LB = Special Label Function

ASSEM LAN AD/DATA LB = Assembly Language Address/Label

SP LB	ASSEM LAN AD/DATA LB	ADDRESS/DATA VALUE	REF	SP LB	ASSEM LAN AD/DATA LB	ADDRESS/DATA VALUE	REF
		HEX	DEC			HEX	DEC
	SAVEOBJ	0	0		A1H	3D	61
	GOWARM	0	0		A1L	3C	60
	GOSTROUT	3	3		A2H	3F	63
	USR	A	10		A2L	3E	62
	CHARAC	D	13		ABS	EBAF	60,335
	ENDCHR	E	14	MD AD	D000	53,248	
	PNTR	F	15		AD0	E7B9	59,321
	NUMDIM	F	15		AD1	E7C6	59,334
	DIMFLG	10	16		AD2	E7CE	59,342
	VALTYP	11	17		AD3	E7EA	59,370
	DATAFLG	13	19		AD4	E7EE	59,374
	GARFLG	13	19		AD5	E7FA	59,386
	SUBFLG	14	20	MD ADD	D000	53,248	
	INPUTFLG	15	21		ADDACC	ECD5	60,629
	CPRMASK	16	22		ADDON	D998	55,704
	SIGNFLG	16	22		ADEX2	EA10	59,920
	SHAPEL	1A	26		ADEX3	EA1B	59,931
	SHAPEH	1B	27		ADEX4	EA26	59,942
	HCOLOR1	1C	28		ADEXP	EA0E	59,918
	COUNTH	1D	29		ADJEXP	EC9E	60,574

SP ASSEM LAN ADDRESS/DATA REF SP ASSEM LAN ADDRESS/DATA REF
LB AD/DATA LB VALUE LB AD/DATA LB VALUE

	HEX	DEC		HEX	DEC
CH	24	36	ADMAN	E855	59,477
GBASL	26	38	AEX	ECA0	60,576
GBASH	27	39	AMPER	3F5	1,013
H2	2C	44	AND	DF55	57,173
V2	2D	45	ARG	A5	165
HMASK	30	48	ARGSGN	AA	170
INVFLG	32	50	ARITH	DDC5	56,773
PROMPT	33	51	ARRAY	E11E	57,630
A1L	3C	60	ARYFOUND	E19E	57,758
A1H	3D	61	ARYLOOP	E16D	57,709
A2L	3E	62	ARYNAM?	E179	57,721
A2H	3F	63	ARYPNT	94	148
LINNUM	50	80	ARYSTR	E50C	58,636
TEMPPT	52	82	ARYTAB	6B	107
LASTPT	53	83	ARYVA2	E4CA	58,570
TEMPST	55	85	ARYVA3	E4CE	58,574
INDEX	5E	94	ARYVAR	E4C2	58,562
DEST	60	96	ARYVGO	E4D9	58,585
RESULT	62	98	ASC	E6E5	59,109
TXTTAB	67	103	ASCHEX	DA12	55,826
VARTAB	69	105	AT?	F218	61,976
ARYTAB	6B	107	ATN	F09E	61,598
STREND	6D	109	ATN1	F0A6	61,606
FRETOP	6F	111	ATN2	F0B4	61,620
FRESPC	71	113	ATN3	F0C7	61,639
MEMSIZ	73	115	ATNSER	F0CE	61,646
CURLIN	75	117	AYINT	E10C	57,612
OLDLIN	77	119	AtoFAC	E850	59,472
OLDTEXT	79	121	AtoFACS	E852	59,474
DATLIN	7B	123	BADNAM	DF4	57,332
DATPTR	7D	125	BADSUBS	D2CB	53,963
INPTR	7F	127	BILLION	ED14	60,692
for	81	129	BILmIONE	ED0F	60,687
VARNAM	81	129	? BKGND	F3F6	62,454
data	83	131	BKGND1	F3FE	62,462
VARPNT	83	131	BLTU	D393	54,163
FORPNT	85	133	BLTU2	D39A	54,170
TXPSV	87	135	M BPD	E67F	59,007
LASTOP	87	135	BREAKIN	D35D	54,109
CPRTYP	89	137	BSB	E049	57,417
FNCNAM	8A	138	MD BUMP	D000	53,248
TEMP3	8A	138	BYPASS	E012	57,362
DSCPTR	8C	140	CALL	F1D5	61,909
DSCLEN	8F	143	CANTCON	D332	54,066
JMPADRS	90	144	CAT	E597	58,775
LENGTH	91	145	CH	24	36
EXTRASV	92	146	CHARAC	D	13
TEMP1	93	147	CHKCLS	DEB8	57,016
ARYPNT	94	148	CHKCOM	DEBE	57,022
HIGHDS	94	148	CHKDIM	E1AA	57,770
HIGHTR	96	150	CHKDP	EC66	60,518

SP ASSEM LAN ADDRESS/DATA REF			SP ASSEM LAN ADDRESS/DATA REF		
LB	AD/DATA LB	VALUE	LB	AD/DATA LB	VALUE
	HEX	DEC		HEX	DEC
TEMP2	98	152	CHKMEM	D3D6	54,230
INDX	99	153	CHKNUM	DD6A	56,682
TMPEXP	99	153	CHKOPN	DEBB	57,019
EXPON	9A	154	CHKSTR	DD6C	56,684
LOWTR	9B	155	CHKTYP	DDB4	56,756
DPFLG	9B	155	CHKVAL	DD6D	56,685
EXPSGN	9C	156	CHRGET	B1	177
DSCTMP	9D	157	CHRGOT	B7	183
FAC	9D	157	CHRSTR	E646	58,950
VPNT	A0	160	CLEAR	D66A	54,890
pop	A1	161	CLEARC	D66C	54,892
FACSGN	A2	162	CMDTABL	D000	53,248
SERLEN	A3	163	CMPBM1	ED62	60,770
FPGEN	A4	164	CMPDONE	DFC1	57,281
ARG	A5	165	CMPHM	ED6D	60,781
ARGSGN	AA	170	CMPLOOP	DFAA	57,258
goto	AB	171	MD CMPR	D000	53,248
SGNCPR	AB	171	COLCOUNT	EA	234
STRNG1	AB	171	COLDST	F128	61,736
EXTRAFAC	AC	172	COLON?	D842	55,362
SERPNT	AD	173	COLOR	F24F	62,031
STRNG2	AD	173	COLORTB	F6F6	63,222
PRGEND	AF	175	COMBYTE	E74C	59,212
gosub	B0	176	COMPARE	DDE4	56,804
CHRGET	B1	177	CON	D8A1	55,457
rem	B2	178	CONINT	E6FB	59,131
CHRGOT	B7	183	CONT	D896	55,446
TXTPTR	B8	184	CONUPK	E9E3	59,875
print	BA	186	COPSTR	DA9A	55,962
tab	C0	192	COPY	DAB7	55,991
to	C1	193	COS	EFEA	61,418
fn	C2	194	COSTBL	F5BA	62,906
spc	C3	195	COUNTED	EDE7	60,903
then	C4	196	COUNTH	1D	29
at	C5	197	COUT	FDED	65,005
not	C6	198	CPRMASK	16	22
step	C7	199	CPROP	DD98	56,728
plus	C8	200	CPRTYP	89	137
minus	C9	201	CRDO	DAFB	56,059
RNDSEED	C9	201	CSHFT2	F47E	62,590
equal	D0	208	? CSHIFT	F47C	62,588
DXL	D0	208	CTRC?	D86C	55,404
DXH	D1	209	CURLIN	75	117
sgn	D2	210	CURLSV	F6	246
DY	D2	210	CV2	DD74	56,692
QDRNT	D3	211	CY	E45F	58,463
EL	D4	212	DATA	D995	55,701
EH	D5	213	DATAFLG	13	19
LOCK	D6	214	DATAN	D9A3	55,715

SP ASSEM LAN	ADDRESS/DATA	REF	SP ASSEM LAN	ADDRESS/DATA	REF
LB AD/DATA LB	VALUE		LB AD/DATA LB	VALUE	

	HEX	DEC		HEX	DEC
scrn	D7	215	DATIN	DC69	56,425
ERRFLG	D8	216	DATLIN	7B	123
ERRLIN	DA	218	DATPTR	7D	125
ERRPOS	DC	220	MD DECR	D000	53,248
ERRNUM	DE	222	DECTBL	EE69	61,033
ERRSTK	DF	223	DEF	E313	58,131
X0L	E0	224	DEL	F331	62,257
X0H	E1	225	DESC?	DA8C	55,948
Y0	E2	226	DEST	60	96
HCOLORZ	E4	228	DFLTDIM	E1F7	57,847
HNDX	E5	229	DIM	DFD9	57,305
HPAG	E6	230	DIMFLG	10	16
SCALEZ	E7	231	DIMLUP	E253	57,939
leftstr	E8	232	DIMOK	E26F	57,967
SHAPEPNT	E8	232	DIMOK2	E270	57,968
COLCOUNT	EA	234	DIR?	DBC7	56,263
FIRST	F0	240	DIRECT	D7E5	55,269
SPEEDZ	F1	241	DIV	EA5E	59,998
TRCFLG	F2	242	DIV10	EA55	59,989
ORMASK	F3	243	DIVZ	EAE1	60,129
TXTPSV	F4	244	DIVbyZRO	D2E5	53,989
CURLSV	F6	246	DMTH	DE41	56,897
REMSTK	F8	248	DOCMP	DFB5	57,269
ROTZ	F9	249	DOERRMSG	D419	54,297
STACK	100	256	DOMATH	DE43	56,899
IN	200	512	DOMTH	DE3A	56,890
AMPER	3F5	1,013	DONE	D610	54,800
KEY	C000	49,152	DOREENT	DB90	56,208
TXTCLE	C050	49,232	DOSPC	DB35	56,117
MIXCLR	C052	49,234	DOWN	F505	62,725
MIXSET	C053	49,235	DOWN1	F52A	62,762
LOWSCR	C054	49,236	DOWN2	F52C	62,764
HISCR	C055	49,237	DOWN3	F524	62,756
LORES	C056	49,238	DP1	F741	63,297
HIRES	C057	49,239	DP2	F747	63,303
MD INCR	D000	53,248	DP3	F766	63,334
MD DECR	D000	53,248	DPDIG	EC87	60,551
MD CMPR	D000	53,248	DPFLG	9B	155
MD TR	D000	53,248	DPL	ED9F	60,831
MD TRDB	D000	53,248	DPLEFT	ECA9	60,585
MD TRX	D000	53,248	DPLOC	ED9E	60,830
MD TRAX	D000	53,248	DPRIGHT	ECB2	60,594
MD TRAY	D000	53,248	DRAW	F769	63,337
MD TRXY	D000	53,248	? DRAW0	F601	62,977
MD AD	D000	53,248	DRAW1	F605	62,981
MD ADD	D000	53,248	DRAW2	F626	63,014
MD SB	D000	53,248	DRAW3	F630	63,024
MD SUB	D000	53,248	DRAW4	F63D	63,037
MD BUMP	D000	53,248	DRAW5	F648	63,048
MD PUSH	D000	53,248	DRWPNT	F72D	63,277
MD PULL	D000	53,248	DSCLN	8F	143

SP ASSEM LAN ADDRESS/DATA REF				SP ASSEM LAN ADDRESS/DATA REF			
LB AD/DATA LB		VALUE		LB AD/DATA LB		VALUE	
		HEX	DEC			HEX	DEC
MD	NEG	D000	53,248		DSCPTR	8C	140
	CMDTABL	D000	53,248		DSCTMP	9D	157
	UNFNC	D080	53,376		DV1	E538	58,680
	MATHTBL	D0B2	53,426		DV2	E542	58,690
	MINUS	D0C7	53,447		DVAR	E523	58,659
	EQUAL	D0CA	53,450		DVARS	E519	58,649
	PLUS	D0CD	53,453		DVARTS	E552	58,706
	TOKTABL	D0D0	53,456		DXH	D1	209
	ERRMSG	D260	53,856		DXL	D0	208
	NXwoFOR	D260	53,856		DY	D2	210
	SYNTAXERR	D270	53,872		EH	D5	213
	RTNwoGSB	D276	53,878		EL	D4	212
	OofDATA	D28A	53,898		END	D870	55,408
	ILLQUAN	D295	53,909		END2	D871	55,409
	OVFLOW	D2A5	53,925		END3	D888	55,432
	OofMEM	D2AD	53,933		END4	D88A	55,434
	UNDSTAT	D2BA	53,946		ENDCHR	E	14
	BADSUBS	D2CB	53,963		ENDFOR	DD55	56,661
	REdimARR	D2D8	53,976		ENDRNG	D6C4	54,980
	DIVbyZRO	D2E5	53,989		EQ1C	F5B9	62,905
	ILLDIR	D2F5	54,005		EQ3	F4CD	62,669
	TYPEMISS	D303	54,019		EQ4	F508	62,728
	STRtoLNG	D310	54,032		EQUAL	D0CA	53,450
	FORMtoCX	D31F	54,047		EQL	DED0	57,040
	CANTCON	D332	54,066		EQUOP	DE98	56,984
	UNDFUNC	D340	54,080		ERFLG?	D863	55,395
	ERRIN	D350	54,096		ERLIN	DB7F	56,191
	INMSG	D358	54,104		ERLUP	D41F	54,303
	BREAKIN	D35D	54,109		ERRDIR	E306	58,118
	GTFORPNT	D365	54,117		ERRFLG	D8	216
	FNDFOR	D36A	54,122		ERRIN	D350	54,096
	SAMEFOR?	D37F	54,143		ERRLIN	DA	218
	NXFOR	D38B	54,155		ERRMSG	D260	53,856
	RET1	D392	54,162		ERRNUM	DE	222
	BLTU	D393	54,163		ERROR	D412	54,290
	BLTU2	D39A	54,170		ERRPOS	DC	220
	SETEND	D3B7	54,199		ERRSTK	DF	223
	MVBYT	D3C3	54,211		EVAL	EC61	60,513
	NXBYT	D3C7	54,215		EVD	ECBE	60,606
	NXPAG	D3CE	54,222		EVDONE	ECB9	60,601
	CHKMEM	D3D6	54,230		EXIG	DCDF	56,543
	REASON	D3E3	54,243		EXIT	DE5D	56,925
	RS1	D3ED	54,253		EXP	EF09	61,193
	RS2	D3F1	54,257		EXPON	9A	154
	RS3	D3FC	54,268		EXPSER	EEE0	61,152
	RET2	D40F	54,287		EXPSGN	9C	156
	MEMERR	D410	54,288		EXTRAFAFAC	AC	172
	ERROR	D412	54,290		EXTRASV	92	146
	DOERRMSG	D419	54,297		FAC	9D	157
	ERLUP	D41F	54,303		FACSGN	A2	162
	PRNTIN?	D431	54,321		FACSTRNG	ED36	60,726

SP	ASSEM	LAN	ADDRESS/DATA	REF	SP	ASSEM	LAN	ADDRESS/DATA	REF
LB	AD/DATA	LB	VALUE		LB	AD/DATA	LB	VALUE	
		HEX	DEC				HEX	DEC	
RESTART	D43C	54,332		FADD	E7BE	59,326			
NXLIN	D45C	54,364		FADDH	E7A0	59,296			
NL1	D49F	54,431		FADDT	E7C1	59,329			
MVDWN	D4A7	54,439		FALSE	DF5D	57,181			
NEWLN?	D4B5	54,453		FC1	EBE9	60,393			
MVPRG	D4D1	54,481		FC2	EBEF	60,399			
INSRTLIN	D4EA	54,506		FCOMP	EBB2	60,338			
LINKSET	D4F2	54,514		FCOMP2	EBB4	60,340			
NXLINK	D4FE	54,526		FD1	EA80	60,032			
PUTLINK	D50F	54,543		FD2	EA96	60,054			
FINDEOL	D511	54,545		FD3	EEA3	60,067			
INLIN	D52C	54,572		FD4	EEA6	60,070			
INLIN2	D52E	54,574		FD5	EAB4	60,084			
GDBUFS	D539	54,585		FD6	EAD1	60,113			
STRIP	D541	54,593		FD7	EAD5	60,117			
NOI	D54C	54,604		FDIV	EA66	60,006			
INCHR	D553	54,611		FDIVT	EA69	60,009			
GETIN	D559	54,617		FE1	E415	58,389			
PARSE	D56C	54,636		FE2	E41F	58,399			
NXCHR	D56D	54,637		FEND	E3F7	58,359			
SE	D578	54,648		FEVLOOP	DD86	56,710			
TOK?	D588	54,664		FIN	EC4A	60,490			
ISTOK?	D590	54,672		FIN2	EC4E	60,494			
NY	D5A2	54,690		FIN3	EC5D	60,509			
NX	D5A7	54,695		FINDATA	DCA0	56,480			
LIN	D5A8	54,696		FINDEOL	D511	54,545			
PUTTOK	D5CB	54,731		FIRST	F0	240			
PUTIN	D5CD	54,733		FL1	D61E	54,814			
SSF	D5E0	54,752		FL2	D635	54,837			
REM?	D5E2	54,754		FLASH	F280	62,080			
SHFTIN	D5E9	54,761		FLO1	EB9B	60,315			
SHIN	D5F2	54,770		FLO2	EBA0	60,320			
SKIPTOK	D5F9	54,777		FLOAT	EB93	60,307			
SK2	D5FD	54,781		FLOOP	E832	59,442			
PLU?	D604	54,788		FM1	E9B0	59,824			
DONE	D610	54,800		FM2	E9B5	59,829			
FNDLIN	D61A	54,810		FM3	E9B8	59,832			
FL1	D61E	54,814		FM4	E9D4	59,860			
FL2	D635	54,837		FMU	E987	59,783			
GETLINK	D63E	54,846		FMULT	E97F	59,775			
NOSUCH	D647	54,855		FMULTT	E982	59,778			
RET3	D648	54,856		FN?	DEA4	56,996			
NEW	D649	54,857		FNC?	E341	58,177			
SCRATCH	D64B	54,859		FNCDATA	E3AF	58,287			
SETPTRS	D665	54,885		FNCNAM	8A	138			
CLEAR	D66A	54,890		FNDARY	E169	57,705			
CLEARC	D66C	54,892		FNDELEM	E24B	57,931			
STKINI	D683	54,915		FNDFOR	D36A	54,122			
RET4	D696	54,934		FNDLIN	D61A	54,810			
STXTPT	D697	54,935		FNDMEMHI	F181	61,825			
LIST	D6A5	54,949		FNDVAR	E488	58,504			
STRTRNG	D6B1	54,961		FOR	D766	55,142			

SP ASSEM LAN	ADDRESS/DATA	REF	SP ASSEM LAN	ADDRESS/DATA	REF
LB AD/DATA LB	VALUE		LB AD/DATA LB	VALUE	

	HEX	DEC		HEX	DEC
ENDRNG	D6C4	54,980	FOR2	D777	55,159
MAINLST	D6CC	54,988	FORMtoCX	D31F	54,047
NXLST	D6DA	55,002	FORPNT	85	133
LSTD?	D6F5	55,029	FOUT	ED34	60,724
LST1LIN	D6F7	55,031	FPGEN	A4	164
LISTLOOP	D6FE	55,038	FPWRT	EE97	61,079
SENDCHR	D702	55,042	FR1	E874	59,508
NCR	D712	55,058	FR2	E880	59,520
LISTED	D724	55,076	FR3	E88D	59,533
GETCHR	D72C	55,084	FRE	E2DE	58,078
GC	D731	55,089	FRE2	E2E5	58,085
TOKEN?	D734	55,092	FREFAC	E600	58,880
SKPTK	D746	55,110	FRESC	71	113
TOKLP	D749	55,113	FRESTR	E5FD	58,877
PRTOK	D750	55,120	FRETMP	E604	58,884
TOKLUP	D755	55,125	FRETMS	E635	58,933
TOKDONE	D75F	55,135	FRETOP	6F	111
FOR	D766	55,142	FRMEVL	DD7B	56,699
FOR2	D777	55,159	FRMEVL2	DD95	56,725
STEP	D7AF	55,215	FRNUM	DD67	56,679
ONESTEP	D7C3	55,235	FROUND	E88F	59,535
NEWSTT	D7D2	55,250	FSUB	E7A7	59,303
DIRCT	D7E5	55,269	FSUBT	E7AA	59,306
TRACE?	D805	55,301	FULL	E474	58,484
TR1	D81D	55,325	FUNCT	E354	58,196
GOEND	D826	55,334	GARBAG	E484	58,500
GOCMD	D828	55,336	GARFLG	13	19
GOCMD2	D82A	55,338	GBASH	27	39
NOTOK	D83F	55,359	GBASL	26	38
COLON?	D842	55,362	GC	D731	55,089
JSY	D846	55,366	GD	E0F9	57,593
RESTORE	D849	55,369	GDBUFS	D539	54,585
SETDA	D853	55,379	GERR	DD0D	56,589
RET5	D857	55,383	GET	DBA0	56,224
ISCNTC	D858	55,384	GETADR	E752	59,218
GK	D860	55,392	GETARY	E0ED	57,581
ERFLG?	D863	55,395	GETARY2	E0EF	57,583
CTRC?	D86C	55,404	GETARYPT	F7D9	63,449
STOP	D86E	55,406	GETBYT	E6F8	59,128
END	D870	55,408	GETCHR	D72C	55,084
END2	D871	55,409	GETEXP	ECE8	60,648
END3	D888	55,432	GETIN	D559	54,617
END4	D88A	55,434	GETLINK	D63E	54,846
GOSTART	D893	55,443	GETLN	FD6A	64,874
CONT	D896	55,446	GETOLD	E3A9	58,281
CON	D8A1	55,457	GETSPA	E452	58,450
RET6	D8AF	55,471	GETSPC	E454	58,452
SAVE	D8B0	55,472	GETSTR	E6DC	59,100
LOAD	D8C9	55,497	GETVAL	DE60	56,928
JLNK	D8ED	55,533	GGERR	F6E6	63,206
VARTIO	D8F0	55,536	GIQ	E948	59,720
PROGIO	D901	55,553	GIVAYF	E2F2	58,098

SP ASSEM LAN ADDRESS/DATA REF			SP ASSEM LAN ADDRESS/DATA REF		
LB	AD/DATA LB	VALUE	LB	AD/DATA LB	VALUE
		HEX DEC			HEX DEC
RUN	D912	55,570	GK	D860	55,392
RUNLINE	D91B	55,579	GME	E26C	57,964
GOSUB	D921	55,585	GO1	D955	55,637
GOLINE	D935	55,605	GO2	D959	55,641
GOTO	D93E	55,614	GOCMD	D828	55,336
GO1	D955	55,637	GOCMD2	D82A	55,338
GO2	D959	55,641	GOEND	D826	55,334
RET7	D96A	55,658	GOERR	F206	61,958
POP	D96B	55,659	GOEX	DE38	56,888
UNDERR	D97C	55,676	GOGEX	EC8A	60,554
GSYNER	D981	55,681	GOGO	E514	58,644
RETURN	D984	55,684	GOIQ	E6F2	59,122
DATA	D995	55,701	GOLINE	D935	55,605
ADDON	D998	55,704	GONEWST	DD52	56,658
RET8	D9A2	55,714	GOROUT	DF3F	57,151
DATAN	D9A3	55,715	GOSTART	D893	55,443
REMN	D9A6	55,718	GOSTROUT	3	3
RM1	D9AE	55,726	GOSUB	D921	55,585
RM2	D9B6	55,734	GOTFOR	DD0F	56,591
PULL3	D9C5	55,749	GOTO	D93E	55,614
IF	D9C9	55,753	GOTO?	D9F4	55,796
TRUE?	D9D8	55,768	GOTSPA	E469	58,473
REM	D9DC	55,772	GOWARM	0	0
IFTRUE	D9E1	55,777	GR	F390	62,352
JGOCMD	D9E9	55,785	GRBPAS	E562	58,722
ONGOTO	D9EC	55,788	GSE	E269	57,961
GOTO?	D9F4	55,796	GSYNER	D981	55,681
ONCNT	D9F8	55,800	GTBYTC	E6F5	59,125
NXNUM	DA00	55,808	GTFORPNT	D365	54,117
RET9	DA0B	55,819	GTLT	E011	57,361
LINGET	DA0C	55,820	GTNUM	E746	59,206
ASCHEX	DA12	55,826	H2	2C	44
NXDIG	DA40	55,872	HALF	EE64	61,028
LET	DA46	55,878	HALFneg	E937	59,703
LET2	DA63	55,907	HANDLERR	F2E9	62,185
LETREAL	DA77	55,927	? HCLR	F3F2	62,450
LETSTR	DA7A	55,930	HCOLOR	F6E9	63,209
PUTSTR	DA7B	55,931	HCOLOR1	1C	28
DESC?	DA8C	55,948	HCOLORZ	E4	228
COPSTR	DA9A	55,962	HCOUNT	F58B	62,859
NEWDESC	DAA1	55,969	? HFIND	F5CB	62,923
COPY	DAB7	55,991	HFIND1	F5F0	62,960
PRSTRING	DACF	56,015	HFIND2	F5FE	62,974
PRINT	DAD5	56,021	HFNS	F6B9	63,161
PRINT2	DAD7	56,023	HFNS1	F6CD	63,181
CRDO	DAFB	56,059	HGLIN	F53A	62,778
NEGATE	DB00	56,064	HGR	F3E2	62,434
RET10	DB02	56,066	HGR2	F3D8	62,424
TAB	DB03	56,067	HIGHDS	94	148
NXCLM	DB0E	56,078	HIGHTR	96	150

SP ASSEM LAN	ADDRESS/DATA	REF	SP ASSEM LAN	ADDRESS/DATA	REF
LB AD/DATA LB	VALUE		LB AD/DATA LB	VALUE	
	HEX	DEC		HEX	DEC
TABWHERE	DB16	56,086	HIMEM	F286	62,086
SPC?	DB21	56,097	HIRES	C057	49,239
TABIT	DB2B	56,107	HISCR	C055	49,237
NXSPC	DB2C	56,108	HLIN	F232	62,002
NEXTCHR	DB2F	56,111	HLIN2	F550	62,800
DOSPC	DB35	56,117	HLIN3	F568	62,824
STROUT	DB3A	56,122	HLIN4	F59E	62,878
STRPRT	DB3D	56,125	HLINE	F819	63,513
NXCHAR	DB44	56,132	? HLINRL	F530	62,768
OUTSP	DB57	56,151	HMASK	30	48
OUTQUES	DB5A	56,154	HMmiTNTH	ED0A	60,682
OUTDO	DB5C	56,156	HNDX	E5	229
SEND	DB64	56,164	HOME	FC58	64,600
INPUTERR	DB71	56,177	HP2	F708	63,240
READERR	DB7B	56,187	HP3	F70F	63,247
ERLIN	DB7F	56,191	HPAG	E6	230
INPERR	DB86	56,198	HPLOT	F6FE	63,230
RESPERR	DB87	56,199	HPLOT0	F457	62,551
DOREENT	DB90	56,208	HPOSN	F411	62,481
GET	DBA0	56,224	HPOSN1	F441	62,529
INPUT	DBB2	56,242	HPOSN2	F442	62,530
QOUT	DBC4	56,260	HTAB	F7E7	63,463
DIR?	DBC7	56,263	HTAB1	F7EC	63,468
NXIN	DBDC	56,284	HTAB2	F7FA	63,482
READ	DBE2	56,290	IF	D9C9	55,753
ZF	DBE9	56,297	IFTRUE	D9E1	55,777
MAININP	DBEB	56,299	ILLDIR	D2F5	54,005
NXINP	DBF1	56,305	ILLQUAN	D295	53,909
SNDQ?	DC1F	56,351	IN	200	512
STXP	DC27	56,359	INCHR	D553	54,611
INSTART	DC2B	56,363	MD INCR	D000	53,248
PUTCHR	DC3F	56,383	INDEX	5E	94
PENCHR	DC4B	56,395	INDX	99	153
PECHR	DC4C	56,396	INLIN	D52C	54,572
SKP	DC57	56,407	INLIN2	D52E	54,574
NUMIN	DC63	56,419	INMSG	D358	54,104
DATIN	DC69	56,425	INNU	F1DE	61,918
WNX	DC72	56,434	INPDONE	DCC6	56,518
SWPNT	DC7E	56,446	INPERR	DB86	56,198
INPFIN	DC99	56,473	INPFIN	DC99	56,473
FINDATA	DCA0	56,480	INPORT	FE8B	65,163
NXS	DCB9	56,505	INPRT	ED19	60,697
INPDONE	DCC6	56,518	INPTR	7F	127
NTD	DCD1	56,529	INPUT	DBB2	56,242
RET11	DCDE	56,542	INPUTERR	DB71	56,177
EXIG	DCDF	56,543	INPUTFLG	15	21
REENT	DCEF	56,559	INS1	E660	58,976
NEXT	DCF9	56,569	INS2	E666	58,982
VARNXT	DCFF	56,575	INS3	E667	58,983
SKPV	DD02	56,578	INS4	E668	58,984
GERR	DD0D	56,589	INS5	E6A2	59,042

SP ASSEM LAN	ADDRESS/DATA	REF	SP ASSEM LAN	ADDRESS/DATA	REF
LB AD/DATA LB	VALUE		LB AD/DATA LB	VALUE	

	HEX	DEC		HEX	DEC
GOTFOR	DD0F	56,591	INSRTDIG	ECC1	60,609
GONEWST	DD52	56,658	INSRTLIN	D4EA	54,506
ENDFOR	DD55	56,661	INSTART	DC2B	56,363
FRMNUM	DD67	56,679	INSTRNG	E6B9	59,065
CHKNUM	DD6A	56,682	INT	EC23	60,451
CHKSTR	DD6C	56,684	INTPART	ED89	60,809
CHKVAL	DD6D	56,685	INTVAR?	E026	57,382
RET12	DD73	56,691	INVERSE	F277	62,071
CV2	DD74	56,692	INVFLG	32	50
MISMTCH	DD76	56,694	IQERR	E199	57,753
JERROR	DD78	56,696	ISCNTC	D858	55,384
FRMEVL	DD7B	56,699	ISLETC	E07D	57,469
FEVLOOP	DD86	56,710	ISTOK?	D590	54,672
FRMEVL2	DD95	56,725	JARY	E046	57,414
CPROP	DD98	56,728	JD10	ED7F	60,799
CHKTYP	DDB4	56,756	JER	E19B	57,755
ARITH	DDC5	56,773	JERR	E432	58,418
PREFTEST	DDCD	56,781	JERROR	DD78	56,696
NXOP	DDD6	56,790	JF	DFCA	57,290
SAVOP	DDD7	56,791	JGOCMD	D9E9	55,785
COMPARE	DDE4	56,804	JLNK	D8ED	55,533
M ND	DDEE	56,814	JM10	ED78	60,792
PREFNC	DDF6	56,822	JMM	F296	62,102
PSHMAD	DDFD	56,829	JMPADRS	90	144
SNTXERR	DE0D	56,845	JOV	EA36	59,958
PSHF	DE10	56,848	JSY	D846	55,366
PSHFACX	DE15	56,853	JSYN	F32E	62,254
PUSHFAC	DE20	56,864	KEY	C000	49,152
NOTMATH	DE35	56,885	LASTOP	87	135
GOEX	DE38	56,888	LASTPT	53	83
DOMTH	DE3A	56,890	? LEFT	F467	62,567
DMTH	DE41	56,897	LEFT1	F471	62,577
DOMATH	DE43	56,899	LEFT2	F476	62,582
EXIT	DE5D	56,925	LEFTSTR	E65A	58,970
GETVAL	DE60	56,928	LEN	E6D6	59,094
SKIP	DE64	56,932	LENGTH	91	145
NUMBER	DE69	56,937	LET	DA46	55,878
VAR?	DE6C	56,940	LET2	DA63	55,907
STRTXT	DE81	56,961	LETREAL	DA77	55,927
ST1	DE8A	56,970	LETSTR	DA7A	55,930
NOT?	DE90	56,976	LFTRT	F465	62,565
EQUOP	DE98	56,984	LG2	E94B	59,723
NOTZ	DE9F	56,991	LIN	D5A8	54,696
FN?	DEA4	56,996	LINCOOR	F209	61,961
SGN?	DEAB	57,003	LINGET	DA0C	55,820
PARCHK	DEB2	57,010	LINKSET	D4F2	54,514
CHKCLS	DEB8	57,016	LINNUM	50	80
CHKOPN	DEBB	57,019	LINPRT	ED24	60,708
CHKCOM	DEBE	57,022	LIST	D6A5	54,949
SYNCHR	DEC0	57,024	LISTED	D724	55,076
SYNERR	DEC9	57,033	LISTLOOP	D6FE	55,038

SP	ASSEM	LAN	ADDRESS/DATA	REF	SP	ASSEM	LAN	ADDRESS/DATA	REF
LB	AD/DATA	LB	VALUE		LB	AD/DATA	LB	VALUE	

	HEX	DEC		HEX	DEC
MIN	DECE	57,038	LOAD	D8C9	55,497
EQU	DED0	57,040	LOCK	D6	214
VARL	DED5	57,045	LOG	E941	59,713
VR1	DEE5	57,061	LOGSER	E918	59,672
VR2	DEF6	57,078	LOGe	EEDB	61,147
SCREEN	DEF9	57,081	LOGtwo	E93C	59,708
UNARY	DF0C	57,100	LOMEM	F2A6	62,118
NOTinstr	DF3A	57,146	LORES	C056	49,238
GOROUT	DF3F	57,151	LOWSCR	C054	49,236
OR	DF4F	57,167	LOWTR	9B	155
AND	DF55	57,173	LR1	F46E	62,574
FALSE	DF5D	57,181	LRUD1	F4B3	62,643
TRUE	DF60	57,184	LRUD2	F4B4	62,644
POSOP	DF65	57,189	LRUD3	F4C4	62,660
STRCMP	DF7D	57,213	LRUD4	F4C8	62,664
SFS	DFA5	57,253	LRUDX1	F49C	62,620
CMPLOOP	DFAA	57,258	LRUDX2	F49D	62,621
NUMCMP	DFB0	57,264	LST1LIN	D6F7	55,031
DOCMP	DFB5	57,269	LSTD?	D6F5	55,029
CMPDONE	DFC1	57,281	MAF	EB66	60,262
JF	DFCA	57,290	MAF2	EB68	60,264
PDL	DFCD	57,293	MAININP	DBEB	56,299
NXDIM	DFD6	57,302	MAINLST	D6CC	54,988
DIM	DFD9	57,305	MAKARY	E1C1	57,793
PTRGET	DFE3	57,315	MAKDIGIT	EDEE	60,910
PTRGET2	DFE8	57,320	MAKINT	E102	57,602
PTRGET3	DFEA	57,322	MAKSTR	EDBD	60,861
BADNAM	DFF4	57,332	MARKEND	EE5A	61,018
NAMOK	DFF7	57,335	MATHTBL	D0B2	53,426
MORNAM	E007	57,351	MB	ED57	60,759
GTLT	E011	57,361	MEMERR	D410	54,288
BYPASS	E012	57,362	MEMFOUND	F195	61,845
STRNG?	E01C	57,372	MEMSIZ	73	115
INTVAR?	E026	57,382	MFA	EB55	60,245
NIN	E036	57,398	MFA2	EB59	60,249
SCDCH	E03D	57,405	MI1	E119	57,625
JARY	E046	57,414	MI2	E11B	57,627
BSB	E049	57,417	MIDSTR	E691	59,025
VSEARCH	E04F	57,423	MIN	DECE	57,038
NXVAR	E059	57,433	MINUS	D0C7	53,447
NV1	E05B	57,435	MISMTCH	DD76	56,694
NV2	E065	57,445	MIXCLR	C052	49,234
NXPTR	E073	57,459	MIXSET	C053	49,235
ISLETC	E07D	57,469	MKINT	E108	57,608
RTN1	E086	57,478	MOK	F3D2	62,418
NOTFND	E087	57,479	MONPLOT	F800	63,488
TWOBRK	E09A	57,498	MONREAD	FEFD	65,277
NEWVAR	E09C	57,500	MONREAD2	FF02	65,282
NWV	E0B2	57,522	MONWAIT	FCA8	64,680
SETVPNT	E0DE	57,566	MORNAM	E007	57,351

SP ASSEM LAN ADDRESS/DATA REF				SP ASSEM LAN ADDRESS/DATA REF			
LB AD/DATA LB		VALUE		LB AD/DATA LB		VALUE	
		HEX	DEC			HEX	DEC
SVP		E0E8	57,576			MOV1F	EB21 60,193
GETARY		E0ED	57,581			MOV2F	EB1E 60,190
GETARY2		E0EF	57,583			MOVAF	EB63 60,259
GD		E0F9	57,593			MOVDWN	F365 62,309
NEGNUM		E0FE	57,598			MOVESTR	E5E6 58,854
MAKINT		E102	57,602			MOVEX	F57C 62,844
MKINT		E108	57,608			MOVEX2	F581 62,849
AYINT		E10C	57,612			MOVFA	EB53 60,243
MI1		E119	57,625			MOVFM	EAF9 60,153
MI2		E11B	57,627			MOVINS	E5D4 58,836
ARRAY		E11E	57,630			MOVIT	F367 62,311
NXTDIM		E12C	57,644			MOVMF	EB2B 60,203
FNDARY		E169	57,705			MOVML	EB23 60,195
ARYLOOP		E16D	57,709			MOVSTR	E5E2 58,850
ARYNAM?		E179	57,721			MSKTBL	F5B2 62,898
NXARY		E188	57,736			MSLUP	EDC1 60,865
SUBERR		E196	57,750			MU1	E2B6 58,038
IQERR		E199	57,753			MU2	E2C0 58,048
JER		E19B	57,755			MU3	E2D9 58,073
ARYFOUND		E19E	57,758			MUL10	EA39 59,961
CHKDIM		E1AA	57,770			MULT	E2AD 58,029
NOTFOUND		E1B8	57,784			MVBACK	EE19 60,953
MAKARY		E1C1	57,793			MVBYT	D3C3 54,211
NINT		E1D5	57,813			MVDG	ECF7 60,663
RAR		E1DE	57,822			MVDWN	D4A7 54,439
SAVDIM		E1E7	57,831			MVPRG	D4D1 54,481
DFLTDIM		E1F7	57,847			MVS2	E5EA 58,858
ZARY		E21A	57,882			MVS3	E5F3 58,867
ZLUP		E229	57,897			MVZP	F152 61,778
NXPG		E22E	57,902			NAMOK	DFF7 57,335
FNDELEM		E24B	57,931			NB	E630 58,928
DIMLUP		E253	57,939			NCR	D712 55,058
GSE		E269	57,961			ND	DDEE 56,814
GME		E26C	57,964			NDP	ECC8 60,616
DIMOK		E26F	57,967			NDX	F388 62,344
DIMOK2		E270	57,968			NEEDEX?	EE26 60,966
NXDM		E281	57,985			NEG	D000 53,248
NINTA		E292	58,002			NEG2	E8A4 59,556
RARY		E298	58,008			NEGATE	DB00 56,064
RTN2		E2AC	58,028			NEGFAC	E89E 59,550
MULT		E2AD	58,029			NEGNUM	E0FE 57,598
MU1		E2B6	58,038			NEGOP	EED0 61,136
MU2		E2C0	58,048			NEW	D649 54,857
MU3		E2D9	58,073			NEWDESC	DAA1 55,969
FRE		E2DE	58,078			NEWLN?	D4B5 54,453
FRE2		E2E5	58,085			NEWNDX	F478 62,584
GIVAYF		E2F2	58,098			NEWSTT	D7D2 55,250
POS		E2FF	58,111			NEWVAR	E09C 57,500
SNGFLT		E301	58,113			NEXT	DCF9 56,569
ERRDIR		E306	58,118			NEXTCHR	DB2F 56,111
UNDFNC		E30E	58,126			NI	F6B4 63,156

SP ASSEM LAN	ADDRESS/DATA	REF	SP ASSEM LAN	ADDRESS/DATA	REF
LB AD/DATA LB	VALUE		LB AD/DATA LB	VALUE	
	HEX DEC			HEX DEC	
DEF	E313 58,131		NIN	E036 57,398	
FNC?	E341 58,177		NINT	E1D5 57,813	
FUNCT	E354 58,196		NINTA	E292 58,002	
SAVOLD	E378 58,232		NL1	D49F 54,431	
GETOLD	E3A9 58,281		NOI	D54C 54,604	
FNCDATA	E3AF 58,287		NORMAL	F273 62,067	
STR	E3C5 58,309		NOSUCH	D647 54,855	
STRINI	E3D5 58,325		NOT?	DE90 56,976	
STRSPA	E3DD 58,333		NOTFND	E087 57,479	
STRLIT	E3E7 58,343		NOTFOUND	E1B8 57,784	
STRLT2	E3ED 58,349		NOTMATH	DE35 56,885	
FEND	E3F7 58,359		NOTOK	D83F 55,359	
QUO?	E404 58,372		NOTRACE	F26F 62,063	
ZEND	E408 58,376		NOTZ	DE9F 56,991	
NZ	E409 58,377		NOTZE	ED4F 60,751	
FE1	E415 58,389		NOTinstr	DF3A 57,146	
FE2	E41F 58,399		NRM	F279 62,073	
PUTNEW	E42A 58,410		NTD	DCD1 56,529	
JERR	E432 58,418		NTL	E5B7 58,807	
PUTEMP	E435 58,421		NUM10	EA50 59,984	
GETSPA	E452 58,450		NUMBER	DE69 56,937	
GETSPC	E454 58,452		NUMCMP	DFB0 57,264	
CY	E45F 58,463		NUMDIM	F 15	
GOTSPA	E469 58,473		NUMIN	DC63 56,419	
FULL	E474 58,484		NV1	E05B 57,435	
GARBAG	E484 58,500		NV2	E065 57,445	
FNDVAR	E488 58,504		NWV	E0B2 57,522	
TVAR	E4A0 58,528		NX	D5A7 54,695	
SVAR	E4A9 58,537		NXARY	E188 57,736	
SVAR	E4B5 58,549		NXBYT	D3C7 54,215	
SVARGO	E4BD 58,557		NXCHAR	DB44 56,132	
ARYVAR	E4C2 58,562		NXCHR	D56D 54,637	
ARYVA2	E4CA 58,570		NXCLM	DB0E 56,078	
ARYVA3	E4CE 58,574		NXDIG	DA40 55,872	
ARYVGO	E4D9 58,585		NXDIGIT	EC64 60,516	
ARYSTR	E50C 58,636		NXDIM	DFD6 57,302	
GOGO	E514 58,644		NXDM	E281 57,985	
DVAR	E519 58,649		NXFOR	D38B 54,155	
DVAR	E523 58,659		NXIN	DBDC 56,284	
DV1	E538 58,680		NXINP	DBF1 56,305	
DV2	E542 58,690		NXLIN	D45C 54,364	
DVARTS	E552 58,706		NXLINK	D4FE 54,526	
VDONE	E55D 58,717		NXLST	D6DA 55,002	
GRBPAS	E562 58,722		NXNUM	DA00 55,808	
CAT	E597 58,775		NXOP	DDD6 56,790	
NTL	E5B7 58,807		NXPAG	D3CE 54,222	
MOVINS	E5D4 58,836		NXPG	E22E 57,902	
MOVSTR	E5E2 58,850		NXPTR	E073 57,459	
MOVESTR	E5E6 58,854		NXS	DCB9 56,505	
MVS2	E5EA 58,858		NXSFT	E8DC 59,612	

SP ASSEM LAN ADDRESS/DATA REF				SP ASSEM LAN ADDRESS/DATA REF			
LB AD/DATA LB		VALUE		LB AD/DATA LB		VALUE	
		HEX	DEC			HEX	DEC
M	MVS3	E5F3	58,867		NXSPC	DB2C	56,108
	FRESTR	E5FD	58,877		NXTDIM	E12C	57,644
	FREFAC	E600	58,880		NXTERM	EF96	61,334
	FRETMP	E604	58,884		NXVAR	E059	57,433
	NB	E630	58,928		NXwoFOR	D260	53,856
	FRETMS	E635	58,933		NY	D5A2	54,690
	RTN3	E645	58,949		NZ	E409	58,377
	CHRSTR	E646	58,950		ODDSER	EF5C	61,276
	LEFTSTR	E65A	58,970		OLDLIN	77	119
	INS1	E660	58,976		OLDTEXT	79	121
	INS2	E666	58,982		ONCNT	D9F8	55,800
	INS3	E667	58,983		ONE	E913	59,667
	INS4	E668	58,984		ONERR	F2CB	62,155
	BPD	E67F	59,007		ONESTEP	D7C3	55,235
	RIGHTSTR	E686	59,014		ONGOTO	D9EC	55,788
	MIDSTR	E691	59,025		OOR	EF24	61,220
	INS5	E6A2	59,042		OR	DF4F	57,167
	INSTRNG	E6B9	59,065		ORMASK	F3	243
	LEN	E6D6	59,094		OUTDO	DB5C	56,156
	GETSTR	E6DC	59,100		OUTOFRNG	EA2B	59,947
	ASC	E6E5	59,109		OUTPORT	FE95	65,173
	GOIQ	E6F2	59,122		OUTQUES	DB5A	56,154
	GTBYTC	E6F5	59,125		OUTSP	DB57	56,151
	GETBYT	E6F8	59,128		OVERFLOW	E8D5	59,605
	CONINT	E6FB	59,131		OvFLOW	D2A5	53,925
	VAL	E707	59,143		OofDATA	D28A	53,898
	VL2	E70F	59,151		OofMEM	D2AD	53,933
	VL3	E727	59,175		PARCHK	DEB2	57,010
	POINT	E73D	59,197		PARITY?	EDE5	60,901
	GTNUM	E746	59,206		PARSE	D56C	54,636
	COMBYTE	E74C	59,212		PDL	DFCD	57,293
	GETADR	E752	59,218		PECHR	DC4C	56,396
	PEEK	E764	59,236		PEEK	E764	59,236
	POKE	E77B	59,259		PENCHR	DC4B	56,395
	WAIT	E784	59,268		PIdoub	F06B	61,547
	WT2	E793	59,283		PIhalf	F066	61,542
	WT3	E797	59,287		PLOT	F225	61,989
	RTN4	E79F	59,295		PLOTFNS	F1EC	61,932
	FADDH	E7A0	59,296		PLU?	D604	54,788
	FSUB	E7A7	59,303		PLUS	D0CD	53,453
	FSUBT	E7AA	59,306		PLUSEPS	E8C6	59,590
	AD0	E7B9	59,321		PNTR	F	15
	FADD	E7BE	59,326		PNTSTK	EE5F	61,023
	FADDT	E7C1	59,329		POINT	E73D	59,197
	AD1	E7C6	59,334		POKE	E77B	59,259
	AD2	E7CE	59,342		POP	D96B	55,659
	AD3	E7EA	59,370		POS	E2FF	58,111
	AD4	E7EE	59,374		POSOP	DF65	57,189
	AD5	E7FA	59,386		PREAD	FB1E	64,286
	SUBMAN	E806	59,398		PREFNC	DDF6	56,822
	SGNIF	E829	59,433		PREFTEST	DDCD	56,781

SP ASSEM LAN ADDRESS/DATA REF SP ASSEM LAN ADDRESS/DATA REF
LB AD/DATA LB VALUE LB AD/DATA LB VALUE

	HEX	DEC		HEX	DEC
SIGNIF	E82E	59,438	PRGEND	AF	175
FLOOP	E832	59,442	PRINT	DAD5	56,021
ZEROFAC	E84E	59,470	PRINT2	DAD7	56,023
AtoFAC	E850	59,472	? PRNTFAC	ED2E	60,718
AtoFACS	E852	59,474	PRNTIN?	D431	54,321
ADMAN	E855	59,477	PRNU	F1E5	61,925
FR1	E874	59,508	PROGIO	D901	55,553
FR2	E880	59,520	PROMPT	33	51
FR3	E88D	59,533	PRSTR	ED31	60,721
FROUND	E88F	59,535	PRSTRING	DACF	56,015
RTN5	E89D	59,549	PRTOK	D750	55,120
NEGFAC	E89E	59,550	PSHF	DE10	56,848
NEG2	E8A4	59,556	PSHFACX	DE15	56,853
PLUSEPS	E8C6	59,590	PSHMAD	DDFD	56,829
RTN6	E8D4	59,604	PTRGET	DFE3	57,315
OVERFLOW	E8D5	59,605	PTRGET2	DFE8	57,320
SHFTRES	E8DA	59,610	PTRGET3	DFA	57,322
NXSFT	E8DC	59,612	MD PULL	D000	53,248
SHIFT	E8F0	59,632	PULL3	D9C5	55,749
SH1	E8FD	59,645	MD PUSH	D000	53,248
SH2	E903	59,651	PUSHFAC	DE20	56,864
SHFTR	E907	59,655	PUTCHR	DC3F	56,383
SH3	E911	59,665	PUTDP	EDAA	60,842
ONE	E913	59,667	PUTEMP	E435	58,421
LOGSER	E918	59,672	PUTEX	EE36	60,982
SQRhalf	E92D	59,693	PUTIN	D5CD	54,733
SQRtwo	E932	59,698	PUTLINK	D50F	54,543
HALFneg	E937	59,703	PUTNEW	E42A	58,410
LOGtwo	E93C	59,708	PUTSTR	DA7B	55,931
LOG	E941	59,713	PUTOK	D5CB	54,731
GIQ	E948	59,720	PW1	EEA0	61,088
LG2	E94B	59,723	PW2	EEBA	61,114
FMULT	E97F	59,775	QDRNT	D3	211
FMULTT	E982	59,778	QI1	EC06	60,422
FMU	E987	59,783	QI2	EC12	60,434
FM1	E9B0	59,824	QINT	EBF2	60,402
FM2	E9B5	59,829	QOUT	DBC4	56,260
FM3	E9B8	59,832	QUARTER	F070	61,552
FM4	E9D4	59,860	QUO?	E404	58,372
RTN7	E9E2	59,874	RAR	E1DE	57,822
CONUPK	E9E3	59,875	RARY	E298	58,008
ADEXP	EA0E	59,918	RD1	EFCC	61,388
ADEX2	EA10	59,920	RD2	EFE7	61,415
ADEX3	EA1B	59,931	RD2BIT	FCFA	64,762
ADEX4	EA26	59,942	RDKEY	FD0C	64,780
OUTOFRNG	EA2B	59,947	READ	DBE2	56,290
ZERO	EA31	59,953	READERR	DB7B	56,187
JOB	EA36	59,958	REASON	D3E3	54,243
MUL10	EA39	59,961	RECALL	F3BC	62,396
RTN8	EA4F	59,983	REENT	DCEF	56,559
NUM10	EA50	59,984	REM	D9DC	55,772

SP ASSEM LAN	ADDRESS/DATA	REF	SP ASSEM LAN	ADDRESS/DATA	REF
LB AD/DATA LB	VALUE		LB AD/DATA LB	VALUE	
	HEX DEC			HEX DEC	
DIV10	EA55 59,989		REM?	D5E2 54,754	
DIV	EA5E 59,998		REMN	D9A6 55,718	
FDIV	EA66 60,006		REMSTK	F8 248	
FDIVT	EA69 60,009		RES>FAC	EAE6 60,134	
FD1	EA80 60,032		RESPERR	DB87 56,199	
FD2	EA96 60,054		RESTART	D43C 54,332	
FD3	AAA3 60,067		RESTORE	D849 55,369	
FD4	AAA6 60,070		RESULT	62 98	
FD5	EAB4 60,084		RESUME	F318 62,232	
FD6	EAD1 60,113		RET1	D392 54,162	
FD7	EAD5 60,117		RET10	DB02 56,066	
DIVZ	EAE1 60,129		RET11	DCDE 56,542	
RES>FAC	EAE6 60,134		RET12	DD73 56,691	
MOVFM	EA9 60,153		RET2	D40F 54,287	
MOV2F	EB1E 60,190		RET3	D648 54,856	
MOV1F	EB21 60,193		RET4	D696 54,934	
? MOVML	EB23 60,195		RET5	D857 55,383	
SETFOR	EB27 60,199		RET6	D8AF 55,471	
MOVMF	EB2B 60,203		RET7	D96A 55,658	
MOVFA	EB53 60,243		RET8	D9A2 55,714	
MFA	EB55 60,245		RET9	DA0B 55,819	
MFA2	EB59 60,249		RETURN	D984 55,684	
MOVAF	EB63 60,259		REdimARR	D2D8 53,976	
MAF	EB66 60,262		RIGHT	F48A 62,602	
MAF2	EB68 60,264		RIGHTSTR	E686 59,014	
RTN9	EB71 60,273		RM1	D9AE 55,726	
RNDB	EB72 60,274		RM2	D9B6 55,734	
ROUND	EB7A 60,282		RND	EFAE 61,358	
SIGN	EB82 60,290		RNDADJ1	EFA6 61,350	
SIGN1	EB86 60,294		RNDADJ2	EFAA 61,354	
SIGN2	EB88 60,296		RNDB	EB72 60,274	
RTN10	EB8F 60,303		RNDSEED	C9 201	
SGN	EB90 60,304		ROT	F721 63,265	
FLOAT	EB93 60,307		ROTZ	F9 249	
FLO1	EB9B 60,315		ROUN	ED86 60,806	
FLO2	EBA0 60,320		ROUND	EB7A 60,282	
ABS	EBAF 60,335		RS1	D3ED 54,253	
FCOMP	EBB2 60,338		RS2	D3F1 54,257	
FCOMP2	EBB4 60,340		RS3	D3FC 54,268	
FC1	EBE9 60,393		RTN1	E086 57,478	
FC2	EBEF 60,399		RTN10	EB8F 60,303	
QINT	EBF2 60,402		RTN11	EC11 60,433	
QI1	EC06 60,422		RTN12	EC49 60,489	
RTN11	EC11 60,433		RTN13	EEDA 61,146	
QI2	EC12 60,434		RTN14	EFA5 61,349	
INT	EC23 60,451		RTN15	F0CD 61,645	
ZFAC	EC40 60,480		RTN16	F122 61,730	
RTN12	EC49 60,489		RTN2	E2AC 58,028	
FIN	EC4A 60,490		RTN3	E645 58,949	
FIN2	EC4E 60,494		RTN4	E79F 59,295	

SP ASSEM LAN ADDRESS/DATA REF	SP ASSEM LAN ADDRESS/DATA REF
LB AD/DATA LB VALUE	LB AD/DATA LB VALUE

	HEX	DEC		HEX	DEC
FIN3	EC5D	60,509	RTN5	E89D	59,549
EVAL	EC61	60,513	RTN6	E8D4	59,604
NXDIGIT	EC64	60,516	RTN7	E9E2	59,874
CHKDP	EC66	60,518	RTN8	EA4F	59,983
SETSGN	EC85	60,549	RTN9	EB71	60,273
DPDIG	EC87	60,551	RTNwoGSB	D276	53,878
GOGEX	EC8A	60,554	RTS1	F489	62,601
SGNCHK	EC8C	60,556	RTS2	F600	62,976
SETDP	EC98	60,568	RTS3	F6F5	63,221
ADJEXP	EC9E	60,574	RUN	D912	55,570
AEX	ECA0	60,576	RUNLINE	D91B	55,579
DPLEFT	ECA9	60,585	SAMEFOR?	D37F	54,143
DPRIGHT	ECB2	60,594	SAVDIM	E1E7	57,831
EVDONE	ECB9	60,601	SAVE	D8B0	55,472
EVD	ECBE	60,606	SAVEOBJ	0	0
INSRTDIG	ECC1	60,609	SAVOLD	E378	58,232
NDP	ECC8	60,616	SAVOP	DDD7	56,791
ADDACC	ECD5	60,629	SAVY	EE09	60,937
GETEXP	ECE8	60,648	MD SB	D000	53,248
MVDG	ECF7	60,663	SCALE	F727	63,271
STEX	ED05	60,677	SCALEZ	E7	231
HMmiTNTH	ED0A	60,682	SCDCH	E03D	57,405
BILmiONE	ED0F	60,687	SCREEN	DEF9	57,081
BILLION	ED14	60,692	SCRN	F871	63,601
INPRT	ED19	60,697	SCRTCH	D64B	54,859
LINPRT	ED24	60,708	SE	D578	54,648
? PRNTFAC	ED2E	60,718	SEND	DB64	56,164
PRSTR	ED31	60,721	SENDCHR	D702	55,042
FOUT	ED34	60,724	SERIES	EF72	61,298
FACSTRNG	ED36	60,726	SERLEN	A3	163
SFSG	ED41	60,737	SERLOOP	EF89	61,321
NOTZE	ED4F	60,751	SERMAIN	EF76	61,302
MB	ED57	60,759	SERPNT	AD	173
STE	ED60	60,768	SETCOL	F864	63,588
CMPBM1	ED62	60,770	SETDA	D853	55,379
CMPHM	ED6D	60,781	SETDP	EC98	60,568
JM10	ED78	60,792	SETEND	D3B7	54,199
JD10	ED7F	60,799	SETFOR	EB27	60,199
ROUN	ED86	60,806	SETGR	FB40	64,320
INTPART	ED89	60,809	SETHI	F299	62,105
DPLOC	ED9E	60,830	SETHPG	F3EA	62,442
DPL	ED9F	60,831	SETPTRS	D665	54,885
PUTDP	EDAA	60,842	SETSGN	EC85	60,549
SVY	EDBB	60,859	SETTXT	FB39	64,313
MAKSTR	EDBD	60,861	SETVPNT	E0DE	57,566
MSLUP	EDC1	60,865	SFS	DFA5	57,253
PARITY?	EDE5	60,901	SFSG	ED41	60,737
COUNTED	EDE7	60,903	SGN	EB90	60,304
MAKDIGIT	EDEE	60,910	SGN?	DEAB	57,003
SAVY	EE09	60,937	SGNCHK	EC8C	60,556
MVBACK	EE19	60,953	SGNCPR	AB	171

SP ASSEM LAN ADDRESS/DATA REF			SP ASSEM LAN ADDRESS/DATA REF		
LB AD/DATA LB	VALUE		LB AD/DATA LB	VALUE	
	HEX	DEC		HEX	DEC
NEEDX?	EE26	60,966	SGNIF	E829	59,433
PUTEX	EE36	60,982	SH1	E8FD	59,645
WHATX	EE42	60,994	SH2	E903	59,651
WNDUP	EE57	61,015	SH3	E911	59,665
MARKEND	EE5A	61,018	SHAPEH	1B	27
PNTSTK	EE5F	61,023	SHAPEL	1A	26
HALF	EE64	61,028	SHAPEPNT	E8	232
DECTBL	EE69	61,033	SHFTIN	D5E9	54,761
TEND	EE8D	61,069	SHFTR	E907	59,655
SQR	EE8D	61,069	SHFTRES	E8DA	59,610
FPWRT	EE97	61,079	SHIFT	E8F0	59,632
PW1	EEA0	61,088	SHIN	D5F2	54,770
PW2	EEBA	61,114	SHLOAD	F775	63,349
NEGOP	EED0	61,136	SI	F27B	62,075
RTN13	EEDA	61,146	SI1	F023	61,475
LOGe	EEDB	61,147	SI2	F026	61,478
EXPSE	EEE0	61,152	SI3	F033	61,491
EXP	EF09	61,193	SIGN	EB82	60,290
X1	EF19	61,209	SIGN1	EB86	60,294
OR	EF24	61,220	SIGN2	EB88	60,296
X2	EF27	61,223	SIGNFLG	16	22
X3	EF37	61,239	SIGNIF	E82E	59,438
ODDSER	EF5C	61,276	SIN	EFF1	61,425
SERIES	EF72	61,298	SINSER	F075	61,557
SERMAIN	EF76	61,302	SK2	D5FD	54,781
SS	EF85	61,317	SKIP	DE64	56,932
SERLOOP	EF89	61,321	SKIPTOK	D5F9	54,777
NXTERM	EF96	61,334	SKP	DC57	56,407
RTN14	EFA5	61,349	SKPTK	D746	55,110
RNDADJ1	EFA6	61,350	SKPV	DD02	56,578
RNDADJ2	EFAA	61,354	SL1	F796	63,382
RND	EFAE	61,358	SL2	F7A0	63,392
RD1	EFCC	61,388	SL3	F7A3	63,395
RD2	EFE7	61,415	SLI	F3AF	62,383
COS	EFEA	61,418	SNDQ?	DC1F	56,351
SIN	EFF1	61,425	SNGFLT	E301	58,113
SI1	F023	61,475	SNTXERR	DE0D	56,845
SI2	F026	61,478	SPC?	DB21	56,097
SI3	F033	61,491	SPEED	F262	62,050
TAN	F03A	61,498	SPEEDZ	F1	241
TAN2	F062	61,538	SQR	EE8D	61,069
PIhalf	F066	61,542	SQRhalf	E92D	59,693
PIoub	F06B	61,547	SQRtwo	E932	59,698
QUARTER	F070	61,552	SS	EF85	61,317
SINSER	F075	61,557	SSF	D5E0	54,752
ATN	F09E	61,598	ST1	DE8A	56,970
ATN1	F0A6	61,606	STACK	100	256
ATN2	F0B4	61,620	STE	ED60	60,768
ATN3	F0C7	61,639	STEP	D7AF	55,215
RTN15	F0CD	61,645	STEX	ED05	60,677
ATNSER	F0CE	61,646	STKINI	D683	54,915

SP ASSEM LAN ADDRESS/DATA REF SP ASSEM LAN ADDRESS/DATA REF
LB AD/DATA LB VALUE LB AD/DATA LB VALUE

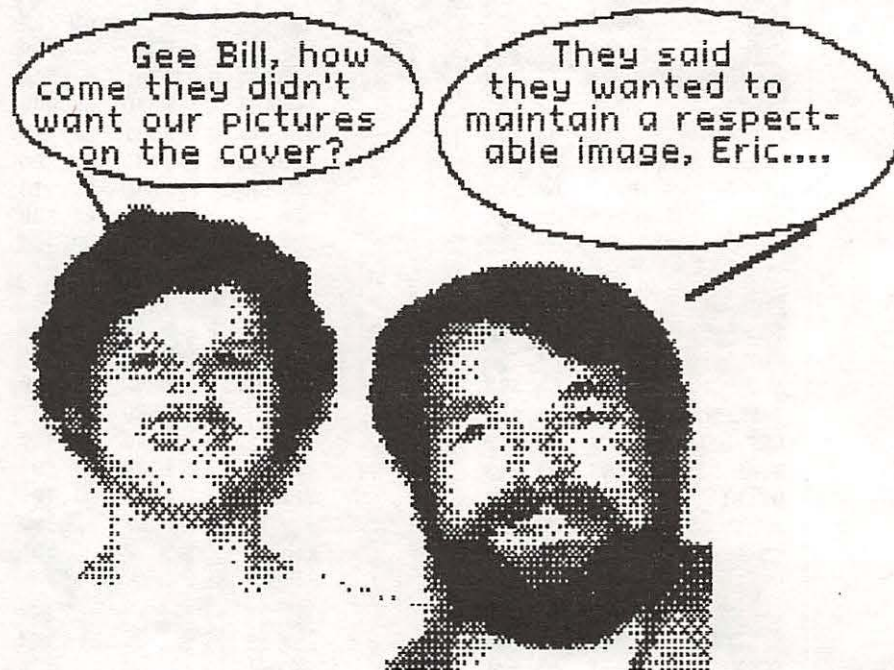
	HEX	DEC		HEX	DEC
ZPSTUFF	F10B	61,707	STOP	D86E	55,406
RTN16	F122	61,730	STORE	F39F	62,367
COLDST	F128	61,736	STR	E3C5	58,309
MVZP	F152	61,778	STRCMP	DF7D	57,213
FNDMEMHI	F181	61,825	STREND	6D	109
MEMFOUND	F195	61,845	STRINI	E3D5	58,325
CALL	F1D5	61,909	STRIP	D541	54,593
INNU	F1DE	61,918	STRLIT	E3E7	58,343
PRNU	F1E5	61,925	STRLT2	E3ED	58,349
PLOTFNS	F1EC	61,932	STRNG1	AB	171
GOERR	F206	61,958	STRNG2	AD	173
LINCOOR	F209	61,961	STRNG?	E01C	57,372
AT?	F218	61,976	STROUT	DB3A	56,122
PLOT	F225	61,989	STRPRT	DB3D	56,125
HLIN	F232	62,002	STRSPA	E3DD	58,333
VLIN	F241	62,017	STRTRNG	D6B1	54,961
COLOR	F24F	62,031	STRTXT	DE81	56,961
VTAB	F256	62,038	STRtoLNG	D310	54,032
SPEED	F262	62,050	STXP	DC27	56,359
TRACE	F26D	62,061	STXTPT	D697	54,935
NOTRACE	F26F	62,063	MD SUB	D000	53,248
NORMAL	F273	62,067	SUBERR	E196	57,750
INVERSE	F277	62,071	SUBFLG	14	20
NRM	F279	62,073	SUBMAN	E806	59,398
SI	F27B	62,075	SVAR	E4B5	58,549
FLASH	F280	62,080	SVARGO	E4BD	58,557
HIMEM	F286	62,086	SVARS	E4A9	58,537
JMM	F296	62,102	SVP	E0E8	57,576
SETHI	F299	62,105	SVY	EDBB	60,859
LOMEM	F2A6	62,118	SWPNT	DC7E	56,446
ONERR	F2CB	62,155	SYNCHR	DEC0	57,024
HANDLERR	F2E9	62,185	SYNERR	DEC9	57,033
RESUME	F318	62,232	SYNTAXERR	D270	53,872
JSYN	F32E	62,254	TAB	DB03	56,067
DEL	F331	62,257	TABIT	DB2B	56,107
MOVDWN	F365	62,309	TABV	FB5B	64,347
MOVIT	F367	62,311	TABWHERE	DB16	56,086
NDX	F388	62,344	TAN	F03A	61,498
GR	F390	62,352	TAN2	F062	61,538
TEXT	F399	62,361	TAPEFNT	F7BC	63,420
STORE	F39F	62,367	TEMP1	93	147
SLI	F3AF	62,383	TEMP2	98	152
RECALL	F3BC	62,396	TEMP3	8A	138
MOK	F3D2	62,418	TEMPPT	52	82
HGR2	F3D8	62,424	TEMPST	55	85
HGR	F3E2	62,434	TEND	EE8D	61,069
SETHPG	F3EA	62,442	TEXT	F399	62,361
? HCLR	F3F2	62,450	TMPEXP	99	153
? BKGND	F3F6	62,454	TOK?	D588	54,664
BKGND1	F3FE	62,462	TOKDONE	D75F	55,135
HPOSN	F411	62,481	TOKEN?	D734	55,092

SP	ASSEM	LAN	ADDRESS/DATA	REF	SP	ASSEM	LAN	ADDRESS/DATA	REF
LB	AD/DATA	LB	VALUE		LB	AD/DATA	LB	VALUE	
			HEX	DEC				HEX	DEC
			HPOSN1	F441 62,529				TOKLP	D749 55,113
			HPOSN2	F442 62,530				TOKLUP	D755 55,125
			HPL0T0	F457 62,551				TOKTABL	D000 53,456
			LFTRT	F465 62,565		MD	TR	D000	53,248
?			LEFT	F467 62,567			TR1	D81D	55,325
			LR1	F46E 62,574			TRACE	F26D	62,061
			LEFT1	F471 62,577			TRACE?	D805	55,301
			LEFT2	F476 62,582		MD	TRAX	D000	53,248
			NEWDX	F478 62,584		MD	TRAY	D000	53,248
?			CSHIFT	F47C 62,588			TRCFLG	F2	242
			CSHFT2	F47E 62,590		MD	TRDB	D000	53,248
			RTS1	F489 62,601			TRUE	DF60	57,184
			RIGHT	F48A 62,602			TRUE?	D9D8	55,768
			LRUDX1	F49C 62,620		MD	TRX	D000	53,248
			LRUDX2	F49D 62,621		MD	TRXY	D000	53,248
			LRUD1	F4B3 62,643			TVAR	E4A0	58,528
			LRUD2	F4B4 62,644			TWOBRK	E09A	57,498
			LRUD3	F4C4 62,660			TXPSV	87	135
			LRUD4	F4C8 62,664			TXTCLR	C050	49,232
			EQ3	F4CD 62,669			TXTPSV	F4	244
			UPDOWN	F4D3 62,675			TXTPTR	B8	184
?			UP	F4D5 62,677			TXTTAB	67	103
			UP1	F4EB 62,699			TYPEMISS	D303	54,019
			UP5	F4F6 62,710			UNARY	DF0C	57,100
			UP2	F4FB 62,715			UNDERR	D97C	55,676
			UP3	F4FD 62,717			UNDFNC	E30E	58,126
			UP4	F4FF 62,719			UNDFUNC	D340	54,080
			UPDWN1	F501 62,721			UNDSTAT	D2BA	53,946
			DOWN	F505 62,725			UNFNC	D080	53,376
			EQ4	F508 62,728	?	UP	F4D5	62,677	
			DOWN3	F524 62,756		UP1	F4EB	62,699	
			DOWN1	F52A 62,762		UP2	F4FB	62,715	
			DOWN2	F52C 62,764		UP3	F4FD	62,717	
?			HLINRL	F530 62,768		UP4	F4FF	62,719	
			HGLIN	F53A 62,778		UP5	F4F6	62,710	
			HLIN2	F550 62,800		UPDOWN	F4D3	62,675	
			HLIN3	F568 62,824		UPDWN1	F501	62,721	
			MOVEX	F57C 62,844		USR	A	10	
			MOVEX2	F581 62,849		V2	2D	45	
			HCOUNT	F58B 62,859		VAL	E707	59,143	
			HLIN4	F59E 62,878		VALTYP	11	17	
			MSKTBL	F5B2 62,898		VAR?	DE6C	56,940	
			EQ1C	F5B9 62,905		VARL	DED5	57,045	
			COSTBL	F5BA 62,906		VARNAM	81	129	
?			HFIND	F5CB 62,923		VARNXT	DCFF	56,575	
			HFIND1	F5F0 62,960		VARPNT	83	131	
			HFIND2	F5FE 62,974		VARTAB	69	105	
			RTS2	F600 62,976		VARTIO	D8F0	55,536	
?			DRAW0	F601 62,977		VDONE	E55D	58,717	
			DRAW1	F605 62,981		VL2	E70F	59,151	
			DRAW2	F626 63,014		VL3	E727	59,175	
			DRAW3	F630 63,024		VLIN	F241	62,017	

SP	ASSEM	LAN	ADDRESS/DATA	REF	SP	ASSEM	LAN	ADDRESS/DATA	REF
LB	AD/DATA	LB	VALUE		LB	AD/DATA	LB	VALUE	

		HEX	DEC			HEX	DEC
	DRAW4	F63D	63,037		VLINE	F828	63,528
	DRAW5	F648	63,048		VPNT	A0	160
?	XDRAW0	F65D	63,069		VR1	DEE5	57,061
	XDRAW1	F661	63,073		VR2	DEF6	57,078
	XDRAW2	F682	63,106		VSEARCH	E04F	57,423
	XDRAW3	F68C	63,116		VTAB	F256	62,038
	XDRAW4	F699	63,129		WAIT	E784	59,268
	XDRAW5	F6A4	63,140		WHATX	EE42	60,994
M	NI	F6B4	63,156		WNDUP	EE57	61,015
	HFNS	F6B9	63,161		WNX	DC72	56,434
	HFNS1	F6CD	63,181		WRITE	FECD	65,229
	GGERR	F6E6	63,206		WT2	E793	59,283
	HCOLOR	F6E9	63,209		WT3	E797	59,287
	RTS3	F6F5	63,221		X0H	E1	225
	COLORTBL	F6F6	63,222		X0L	E0	224
	HPL0T	F6FE	63,230		X1	EF19	61,209
	HP2	F708	63,240		X2	EF27	61,223
	HP3	F70F	63,247		X3	EF37	61,239
	ROT	F721	63,265		XDRAW	F76F	63,343
	SCALE	F727	63,271	?	XDRAW0	F65D	63,069
	DRWPNT	F72D	63,277		XDRAW1	F661	63,073
	DP1	F741	63,297		XDRAW2	F682	63,106
	DP2	F747	63,303		XDRAW3	F68C	63,116
	DP3	F766	63,334		XDRAW4	F699	63,129
	DRAW	F769	63,337		XDRAW5	F6A4	63,140
	XDRAW	F76F	63,343		Y0	E2	226
	SHLOAD	F775	63,349		ZARY	E21A	57,882
	SL1	F796	63,382		ZEND	E408	58,376
	SL2	F7A0	63,392		ZERO	EA31	59,953
	SL3	F7A3	63,395		ZEROFAC	E84E	59,470
	TAPEPNT	F7BC	63,420		ZF	DBE9	56,297
	GETARYPT	F7D9	63,449		ZFAC	EC40	60,480
	HTAB	F7E7	63,463		ZLUP	E229	57,897
	HTAB1	F7EC	63,468		ZPSTUFF	F10B	61,707
	HTAB2	F7FA	63,482		at	C5	197
	MONPLOT	F800	63,488		data	83	131
	HLINE	F819	63,513		equal	D0	208
	VLINE	F828	63,528		fn	C2	194
	SETCOL	F864	63,588		for	81	129
	SCRN	F871	63,601		gosub	B0	176
	PREAD	FB1E	64,286		goto	AB	171
	SETTXT	FB39	64,313		leftstr	E8	232
	SETGR	FB40	64,320		minus	C9	201
	TABV	FB5B	64,347		not	C6	198
	HOME	FC58	64,600		plus	C8	200
	MONWAIT	FCA8	64,680		pop	A1	161
	RD2BIT	FCFA	64,762		print	BA	186
	RDKEY	FD0C	64,780		rem	B2	178
	GETLN	FD6A	64,874		scrn	D7	215
	COUT	FDED	65,005		sgn	D2	210
	INPORT	FE8B	65,163		spc	C3	195

SP	ASSEM	LAN	ADDRESS/DATA	REF	SP	ASSEM	LAN	ADDRESS/DATA	REF
LB	AD/DATA	LB	VALUE		LB	AD/DATA	LB	VALUE	
			HEX	DEC				HEX	DEC
OUTPORT			FE95	65,173	step			C7	199
WRITE			FECB	65,229	tab			C0	192
MONREAD			FEFD	65,277	then			C4	196
MONREAD2			FF02	65,282	to			C1	193



THE APPLE ALMANAC

Here it is! The complete "everything you always wanted to know" book for beginning, intermediate and advanced Apple programmers.

This Almanac has something for everyone!

- Pokes, Peeks, Calls — arranged with Peek formulas and special Apple //e 80 column card Pokes.
- Alphabetical listing of all Applesoft, Integer and DOS Commands, with examples.
- Common Sort algorithms using strings, with common driver.
- RAM/DISK memory cross-reference organized by 48K RAM memory addressing.
- Applesoft and Integer BASIC List interpreter output chart.
- Opcode addressing information chart. Explains the addressing mode and charts every opcode that uses it.
- New opcodes for the 65C02 chip.
- Full Track/Sector List number chart.
- DOS command tables.
- Catalog/Director sectors detailed by addresses.
- HIRES horizontal pixel and screen character charts.
- Tri-line text/LORES and HIRES Line displays for page 1 and 2.
- Conditioned flags/Registers/Memory chart.
- DOS buffers, I/O buffers, DCT, FMPL charts.
- Keyword name and position table.
- Keyword value, DOS error message value, DOS error index, file name entry, track track bit map and shape byte encoding charts.
- Applesoft BASIC Source Symbol chart and all charts on the R65C02 chip.
- Apple II+, Apple //e monitor comparison chart.
- And MORE!!!

So if you're just starting out on the road to programming, or are an old pro, The Apple Almanac has all the information you'll ever need!

ISBN 0-88190-109-1



DATAMOSTTM

8943 Fullbright Avenue, Chatsworth, CA 91311-2750
(818) 709-1202

