

MONK

User's Manual

PALM BEACH SOFTWARE

**RT #1 BOX 119-X
OXFORD, FL 32684**

Revision 12/31/90

MONK is a simple debugging monitor for the 68000 microprocessor. It resides in 30K of memory and vectors all the Traps and Exceptions to its internal routines. There are also a number of User calls through Traps 0 - 14.

The prompt "MONK:" will be displayed when it is active and requesting a command. The following commands are available:

- A Display Assembly Code. Enter Hexadecimal Address. 11 lines of assembly code will be displayed. Any key will display 11 more lines and the ESC will return to MONK.
- B Set Breakpoint. Enter Hexadecimal Address. When the breakpoint is set the program will execute very slowly, because after each instruction the Trace exception will be taken and the program counter will be compared with the breakpoint address. If they are not equal the return from exception will be executed and the next instruction will be processed. This will continue until the breakpoint is reached when the PC, STACK & SR will be displayed and the MONK prompt will be on the screen. Any MONK commands can now be used. This Breakpoint will work in a ROM. Don't place a breakpoint immediately after a TRAP instruction.
- C Change Register. Enter Register Name and the present contents will be displayed, you may now enter the new data.
- D Display Memory in Hex and Ascii. Enter Hex Address. Dn arrow will display next 16 bytes of memory, Up arrow will display previous 16 bytes of memory, and Space will display next 256 bytes. ESC will return to MONK.
- F Re-enter REXDOS.
- G Continue Execution from Breakpoint. If the Breakpoint is still set then the (T)race bit will stay set, if the Breakpoint has been Killed then the (T)race bit will be cleared.
- I Load REXDOS from an IDE Drive. The Link Address must be in Track #0, Sector 1, at the Cylinder specified in the partition table on Sector #2, Track #0, Cylinder #0, bytes 250 & 251.
- J Jump To. Enter Hex Address.
- K Kill Breakpoints. The address set with the "B" command will be cleared. If the Quick Breakpoint is set the code will be replaced and the address will be put into the Program Counter and then Cleared.
- L Load S1-3 Files. The Motorola S1, S2 & S3 format files will be loaded. S7, S8 and S9 records will terminate the load, set the Transfer address, and run the program. If the Transfer address is \$0000 the loader will return to MONK. If the S7-9 record is missing you must manually terminate the load by entering any key. Use COM2 and 9600 baud.

- M Memory Examine and Change. Enter Hex Address and the Address and contents of that address will be displayed. Enter a Hex digit and the memory contents will be changed. The Up arrow will display the previous location and any other char will display the next location. ESC will return to MONK.
- N Calculate Checksum. Enter starting address and ending address. The sum of all bytes between the two addresses will be calculated. The byte at the ending address is not counted.
- O Boot OS-9 68K
- P Display Breakpoint and Quick Breakpoint Addresses.
- Q Set Quick Breakpoint. Enter Hexadecimal Address. The Word at the Address will be Saved along with the Address, and the Trap #5 command inserted. The Processor will run at full speed Until the Breakpoint is reached. The processor will Halt, Display the Status and enter MONK. The PC is now 2 bytes beyond the Breakpoint address. In order to continue you must use the "K" command to restore the instruction and set the PC. This Breakpoint can halt code in Supervisor Mode, but can not be set in a ROM.
- R Display Registers.
- S Execute one Instruction at PC Address. Does not stop in Supervisor Mode.
- T Test Memory. Enter Starting & Ending addresses. The ESC key will terminate the test. In case of error the address will be displayed and the bad and correct bytes shown.
- U Load REXDOS from 1772 DC, Physical Drive #0. Link Address must be in Track 0, Sector 1.
- V Load REXDOS from 37C65 DC, Physical Drive #0. Link Address must be in Track 0, Sector 1.
- W Load REXDOS from Hard Disk A. Link Address must be in Track #0, Sector 1, at the Cylinder specified in the MSDOS partition table on Sector #2, Track #0, Cylinder #0, bytes 250 & 251, Bytes 0 & 1. Bytes 2 - 9 must contain an image of the DIT, which will be put by the HFORMAT routine. Byte 3 must be \$80 for Drive A.
- X Change I/O. A Menu will be displayed, make your choice. When choosing the PC Keyboard be sure that the Caps Lock and Num Lock are off. The Selection made here will be preserved in the battery backed up RAM at \$FFOF49 and the next time you Cold Start MONK the saved selection will be initialized. The Autoboot byte at \$FFOF49 will be cleared.
- Y Load REXDOS from Hard Disk B. Link Address must be in Track #0, Sector 1, at the Cylinder specified in the MSDOS partition

table on Sector #2, Track #0, Cylinder #0, bytes 250 & 251, Bytes 0 & 1. Bytes 2 - 9 must contain an image of the DIT, which will be put by the HFORMAT routine. Byte 3 must be \$A0 for Drive B.

- Z Fill Memory. Enter Starting & Ending addresses and Fill Character.
- ? Help. Display the command set and the version number.

The following TRAP instruction are available:

TRAP #0	Warm Start
TRAP #1	Input Char, no Echo
TRAP #2	Output Char
TRAP #3	Restore Interrupts from Trap #4
TRAP #4	Set Level 7 Interrupt Mask
TRAP #5	Software Interrupt
TRAP #6	Check if Char Ready. Return NEZ if Ready.
TRAP #7	Clear KB Buffer
TRAP #8	Clear all interrupts
TRAP #9	Set Level 4 Interrupt Mask.
TRAP #10	Set Level 6 Interrupt Mask.
TRAP #11	Restore Interrupt Mask from Trap #10
TRAP #12	Set VGA modes. # in D7 will result in :
	01 - Set Video Mode 1 40x25 Text
	03 - Set Video Mode 3 80x25 Text
	0E - Set Video Mode E 640x200 Graphics
	10 - Set Video Mode 10 640x350 Graphics
	12 - Set Video Mode 12 640x480 Graphics
	13 - Set Video Mode 13 320x200 Graphics
	22 - Set Video Mode 22 132x44 Text
	23 - Set Video Mode 23 132x25 Text
	24 - Set Video Mode 24 132x28 Text
	26 - Set Video Mode 26 80x60 Text
	29 - Set Video Mode 29 800x600 Graphics
	2A - Set Video Mode 2A 100x40 Text
	2D - Set Video Mode 2D 640x350 Graphics
	2E - Set Video Mode 2E 640x480 Graphics
	2F - Set Video Mode 2F 640x400 Graphics
	30 - Set Video Mode 30 800x600 Graphics
	37 - Set Video Mode 37 1024x768 Graphics
	38 - Set Video Mode 38 1024x768 Graphics
	53 - Set Video Mode 53 80x50 Text
	80 - Screen Off
	81 - Screen On
	82 - Return Current Mode in D7
	83 - Initialize Mode 3
	84 - Set Default Mode
	85 - Load PAL, A1=data. Rtn/D7=Next Data
	86 - Load DAC, A1=data, Rtn/D7=Next Data
	87 - Rtn/D7=Base Memory Address
	88 - Rtn/D7=VGA Controller Address
	89 - Clear Graphics Memory, D0=color
	8A - Rtn/D7=Segment Register
	8B - Write D0 to Segment Register
	8C - Rtn/D7=Horizontal Resolution
	8D - Rtn/D7=Vertical Resolution
	8E - Rtn/D7=Number of Colors
	8F - Set Default Mode, D0=Mode
	90 - Set Graphics Position, D0=Column, D1=Row
	91 - Output Text in Graphics Mode, A1=String
	92 - Set Text Color, D0=color
	93 - Set Cursor Color, D0=color
	94 - Set Reverse Color, D0=color
	95 - Unlock VGA controller

96 - Switch to Terminal
 97 - Switch to PC & VGA Screen
 98 - Set Border Color, DO=color
 99 - Unlock Font
 9A - Restore Sequencer
 Rtn/D7=\$FF Error
 Trap #13 Reserved for Cache Management.
 TRAP #14 Manage Timer & Sound. # in D7 will result in:
 0 - Stop Timer & Return Count in D7.
 1 - Write DO to Preload Registers & Start
 Timer.
 2 - Wait for Timeout, Write DO to DAC & Reset
 ZDS.
 3 - Iniz Timer to 40 usec, (25kc).
 4 - Rtn/DO=Date,Hrs,Min,Sec
 5 - Rtn Duart #1 Addr in D7.
 6 - Rtn Duart #2 Addr in D7.
 7 - Rtn Pia Addr in D7.

BREAK or PAUSE (CNTL NUMLOCK) Will will cause the Program to halt after the next instruction in User Mode. If a Program is waiting for Input it will be in Supervisor Mode and you must depress a key to get the Halt.

When the "\$" is displayed you are to enter a Hex number, the last 8 digits entered are used. Backspace does not work. Terminate entry with a C/R.

At Power On, MONK will beep the speaker and then look for a Terminal connected to COM1. If not found it will proceed to initialize for a PC keyboard and initialize the previous selection made by the "X" command. If no selection exists it will try to find a display adaptor and initialize it. If for some reason you get a blank screen enter "X" and the character "M" for MGA, "C" for CGA, "E" for EGA or "V" for VGA to tell MONK the type of Display you want. If a Terminal is found, it will try 19.2K baud rate, and then default to 9600. Monk will accept input from either the PC keyboard or the Terminal keyboard; however, programs using Traps for I/O will use the selected I/O devices.

Keyboard input is to a buffer at \$FF0008. Interrupt Level 5 vector will point to the keyboard routine that has been selected. If Level 5 interrupts have been masked, then no keyboard input is allowed. MONK operates with Level 6 set and uses special routines to input directly from the Keyboards. Input characters are still placed in the buffer.

Input through Trap #1 at \$84 checks for a character in the buffer, if not there it will wait. When a character is available it will be returned in register DO. All other registers are preserved.

Output for MONK. Register DO.B is displayed through Trap #2 vector at \$88. This points to the output routine and can be changed by the user program. All other register are preserved.

SMARTTE the Smart Terminal Emulator

Output through the MGA, CGA, EGA, VGA cards and Terminal monitor is controlled by SMARTTE. The command sequences are a subset of the Televideo TV-905 and are compatible with Wyse WY-50 and other popular terminals.

The following keys and key combinations comprise the command set of SMARTTE.

CTL G	\$07	BELL
CTL H	\$08	BACKSPACE, the cursor will be moved right 1 space. If the cursor is at the home position, no action.
CTL I	\$09	TAB, If screen is write protected the cursor will be moved to the beginning of the next unprotected field. If write protect is off, no action.
CTL J	\$0A	LINE FEED, The cursor will advance 1 line. If the cursor is on line 24 the screen will scroll up one line and a new blank line inserted on line 24, if write protect is on the cursor will go to line 1.
CTL K	\$0B	UP ARROW, The cursor will go back 1 line. If the cursor is on line 1, no action.
CTL L	\$0C	RIGHT ARROW, The cursor will advance 1 position. If the cursor is at the end of a line it will go to the beginning of the next line. If write protect is on the cursor will move to the next unprotected character.
CTL M	\$0D	Carriage RETURN, The cursor will move to the beginning of the current line.
CTL ^	\$1E	HOME, The cursor will move to the top left corner of the screen. If write protect is on it will move to the first unprotected character position on the screen.
CTL _	\$1F	NEWLINE, The cursor will move to the beginning of the next line. If the cursor is on line 24 the screen will scroll up one line and a new blank line inserted on line 24, if write protect is on the cursor will go to line 1.
ESC "	\$1B22	UNLOCK KEYBOARD, Any characters in the input buffer will be cleared.
ESC #	\$1B23	LOCK KEYBOARD.
ESC &	\$1B26	SET WRITE PROTECTION, All characters written in

Write Protect Mode (low intensity) will be locked in their position on the screen. No scrolling will be allowed and the cursor can not be moved to a protected position.

ESC '	\$1B27	CLEAR WRITE PROTECTION, Remove Write Protection.
ESC (\$1B28	WRITE HIGH INTENSITY CHARACTERS(Write Unprotected), Normal condition.
ESC)	\$1B29	WRITE LOW INTENSITY CHARACTERS(Write Protected), All characters will be written in low intensity and if Write Protect is Set they will be locked in position.
ESC *	\$1B2A	CLEAR ALL, Screen will be cleared to Unprotected spaces (high intensity), write protection will be removed, any attributes will be cleared and the cursor moved to Home.
ESC . n	\$1B2E	CURSOR ATTRIBUTES, The next byte sent will determine the visual attributes to be displayed on the screen.
	1	\$31 Change cursor color.
	2	\$32 Restore cursor color.
ESC 4	\$1B34	SEND LINE TO CURSOR, The unprotected characters will be sent from the beginning of the line up to and including the cursor position.
ESC 5	\$1B35	SEND PAGE TO CURSOR, The unprotected characters will be sent from the beginning of the screen up to and including the cursor position.
ESC :	\$1B3A	CLEAR UNPROTECTED TO NULLS, Clear all unprotected characters to null characters.
ESC ;	\$1B3B	CLEAR UNPROTECTED TO SPACES, Clear all unprotected characters to space characters.
ESC = r,c	\$1B3D	MOVE CURSOR TO ROW & COLUMN, The next two characters sent specify the Row and Column positions with an offset of \$20. Space, Space is Home position.
ESC ?	\$1B3F	READ CURSOR POSITION, The next three characters returned will be the Row & Column position of the cursor with an offset of \$20 and a C/R.
ESC E	\$1B45	INSERT LINE, All lines including the cursor line will be moved down 1 line, the last line will be lost. A line of spaces will be inserted at the cursor line. The cursor will

stay put. If Write Protect is Set no action.

ESC G n \$1B47 SCREEN ATTRIBUTES, The next byte sent will determine the visual attributes to be displayed on the screen.

0	\$30	Stop all attributes.
2	\$32	Start Blinking.
4	\$34	Start Reverse Video
6	\$36	Start Blink & Reverse Video.
8	\$38	Start Underline.
:	\$3A	Start Underline & Blink.

ESC I \$1B49 BACKTAB, If Write Protect is Set the cursor will move backwards to the beginning of the Unprotected field, if the cursor is at the beginning of a field it will move to the beginning of the previous unprotected field. If Protection is Clear it cursor will Backspace.

ESC Q \$1B51 INSERT CHARACTER, All characters on the line, from the cursor position including the character under the cursor are moved to the right 1 position and a space inserted at the cursor position. If Write Protect is on, only those character in the unprotected field are moved.

ESC R \$1B52 DELETE LINE, The cursor line is deleted, all following lines are moved up and a blank line inserted at line 24. If Write Protect is Set, no action.

ESC T \$1B54 CLEAR TO EOL, The cursor line is cleared to spaces from the cursor to the end of the line. If Write Protect is Set then only clearing to the end of the current unprotected field.

ESC W \$1B57 DELETE CHARACTER, The character under the cursor is Deleted and all the characters on the cursor line are moved left 1 position. A space is put at the last position on the cursor line. If Write Protect is Set only the characters in the current unprotected field will be moved.

ESC Y \$1B59 CLEAR TO EOP, The screen is cleared from the cursor to the end of line 24. If Write Protect is Set, no action.

ESC f \$1B66 WRITE STATUS LINE, The Status line (line 25)

will be cleared and the next characters received will be displayed on line 25 in Reverse Video. A C/R will terminate entry and restore the cursor to its original position. The RIGHT ARROW Key (\$OC) will move the cursor nondestructively.

ESC C/R \$1BOD DISPLAY Carriage RETURN, A left pointing arrow will be displayed.

THE IBM KEYBOARD

The IBM Keyboard is fully decoded and returns the following character(s) for each key press. Function keys return 3 characters \$01,\$XX,\$0D. Some other special keys return 2 characters \$1B,\$XX. (\$--) means no effect.

KEY #	LOWER CASE	UPPER CASE	CONTROL	NUMLOCK	ASCII
1	\$1B	\$9B	\$1B	\$--	ESC
2	\$31	\$21	\$00	\$--	1,!,NUL,1,!
3	\$32	\$40	\$00	\$--	2,@,NUL,2,@
4	\$33	\$23	\$00	\$--	3,#,NUL,3,#
5	\$34	\$24	\$00	\$--	4,\$,NUL,4,\$
6	\$35	\$25	\$00	\$--	5,%,NUL,5,%
7	\$36	\$5E	\$1E	\$--	6,^,RS,6,^
8	\$37	\$26	\$00	\$--	7,&,NUL,7,&
9	\$38	\$2A	\$00	\$--	8,* .NUL,8,*
10	\$39	\$28	\$00	\$--	9,(,NUL,9,(
11	\$30	\$29	\$00	\$--	0,),NUL,0,)
12	\$2D	\$5F	\$1F	\$--	-,_,US,-, _
13	\$3D	\$2B	\$00	\$--	=,+ ,NUL,=,+
14	\$08	\$08	\$08	\$--	BS,BS,BS,BS
15	\$09	\$1B49	\$1B31	\$--	HT,BTAB,TAB
16	\$71	\$51	\$11	\$--	q,Q,DC1,q,Q
17	\$77	\$57	\$17	\$--	w,W,ETB,w,W
18	\$65	\$45	\$05	\$--	e,E,END,e,E
19	\$72	\$52	\$12	\$--	r,R,DC2,r,R
20	\$74	\$54	\$14	\$--	t,T;DC4,t,T
21	\$79	\$59	\$19	\$--	y,Y,EM,y,Y
22	\$75	\$55	\$15	\$--	u,U,NAK,u,U
23	\$69	\$49	\$09	\$--	i,I,HT,i,I
24	\$6F	\$4F	\$0F	\$--	o,O,SI,o,O
25	\$70	\$50	\$10	\$--	p,P,DLE,p,P
26	\$5B	\$7B	\$1B	\$--	[,¼,ESC,[,¼
27	\$5D	\$7D	\$1D	\$--],¾,GS,],¾
28	\$0D	\$0D	\$0D	\$--	C/R,C/R,C/R
29	\$--	\$--	\$--	\$--	CONTROL KEY
30	\$61	\$41	\$01	\$--	a,A,SOH,a,A
31	\$73	\$53	\$13	\$--	s,S,DC3,s,S
32	\$64	\$44	\$04	\$--	d,D,EOT,d,D
33	\$66	\$46	\$06	\$--	f,F,ACK,f,F
34	\$67	\$47	\$07	\$--	g,G,BEL,g,G
35	\$68	\$48	\$08	\$--	h,H,BS,h,H
36	\$6A	\$4A	\$0A	\$--	j,J,LF,j,J
37	\$6B	\$4B	\$0B	\$--	k,K,VT,k,K
38	\$6C	\$4C	\$0C	\$--	l,L,FF,l,L
39	\$3B	\$3A	\$3B	\$--	;
40	\$27	\$22	\$27	\$--	'" ;
41	\$60	\$7E	\$00	\$--	° ,§,NUL,° ,§
42	\$--	\$--	\$--	\$--	LEFT SHIFT
43	\$5C	\$7C	\$1C	\$--	±,!,FS,±,!
44	\$7A	\$5A	\$1A	\$--	z,Z,SUR,z,Z
45	\$78	\$58	\$18	\$--	x,X,CAN,x,X
46	\$63	\$43	\$03	\$--	c,C,ETX,c,C
47	\$76	\$56	\$16	\$--	v,V,SYN,v,V
48	\$62	\$42	\$02	\$--	b,B,STX,b,B

49	\$6E	\$4E	\$0E	\$--	n , N , S O , n , N
50	\$6D	\$4D	\$0D	\$--	m , M , C R , m , M
51	\$2C	\$3C	\$2C	\$--	, , < , , , , <
52	\$2E	\$3E	\$2E	\$--	, , > , , , , >
53	\$2F	\$3F	\$2F	\$--	/ , ? , / , / , ?
54	\$--	\$--	\$--	\$--	RIGHT SHIFT
55	\$2A	\$2A	\$2A	\$--	GREY STAR
56	\$--	\$--	\$--	\$--	ALT KEY
57	\$20	\$20	\$7F	\$--	SP , SP , DEL , SP
58	\$--	\$--	\$--	\$--	CAPS LOCK
59	\$01420D	\$01620D	\$01420D	\$--	(F1) PREV
60	\$01430D	\$01630D	\$01430D	\$--	(F2) NEXT
61	\$01440D	\$01640D	\$01440D	\$--	(F3) LOCAT
62	\$01450D	\$01650D	\$01450D	\$--	(F4) LOAD
63	\$01460D	\$01660D	\$01460D	\$--	(F5) TEST
64	\$01470D	\$01670D	\$01470D	\$--	(F6) FIRST
65	\$01490D	\$01690D	\$01490D	\$--	(F7) ADD
66	\$014A0D	\$016A0D	\$014A0D	\$--	(F8) REPL
67	\$014B0D	\$016B0D	\$014B0D	\$--	(F9) EXIT
68	\$014C0D	\$016C0D	\$014C0D	\$--	(F10) DIAL
69	\$--	\$--	HALT	\$--	NUM LOCK KEY
70	\$1B40	\$1B37	\$1B37	\$--	SCROLL LOCK
71	\$1E	\$1E	\$1E	\$37	HOME (7)
72	\$0B	\$0B	\$0B	\$38	UP ARROW (8)
73	\$01410D	\$01610D	\$01410D	\$39	PGUP (9)
74	\$1B54	\$1B59	\$1B54	\$--	CLR EOL , EOP (G-)
75	\$08	\$08	\$08	\$34	LEFT ARROW (4)
76	\$0A	\$0A	\$0A	\$35	(5)
77	\$0C	\$0C	\$0C	\$36	RIGHT ARROW (6)
78	\$1B4B	\$1B4B	\$1B4B	\$--	INS TEXT (G+)
79	\$01480D	\$016801	\$014801	\$31	END (1)
80	\$0A	\$0A	\$0A	\$32	DOWN ARROW (2)
81	\$01400D	\$01600D	\$01400D	\$33	PGDN (3)
82	\$1B51	\$1B51	\$1B45	\$30	INS (G)
83	\$1B57	\$1B57	\$1B52	\$2E	DEL (.)
84	\$2A	\$2A	\$2A	\$--	(SYSREG) ALT
85	\$014D0D	\$016D0D	\$014D0D	\$--	(F11) DELETE
86	\$014E0D	\$016E0D	\$014D0D	\$--	(F12) HELP

If CAPSLOCK is on, the Shift key will output lower case alpha characters. When the ALT key is depressed the characters output will be \$01,char,\$0D. Cntl SPACE will output the DEL character(\$7F).

T

MEMORY MAP

```

00000000 - 000003FF  EXCEPTION VECTORS
0024 - TRACE & BREAKPOINT VECTOR
0074 - LEVEL 5 VECTOR (KEYBOARD)
007C - LEVEL 7 VECTOR (ABORT)
0080 - TRAP #0 WARM START
0084 - TRAP #1 INPUT CHAR
0088 - TRAP #2 OUTPUT CHAR
008C - TRAP #3 DISABLE INTERRUPTS
0090 - TRAP #4 ENABLE INTERRUPTS
0094 - TRAP #5 SOFTWARE INTERRUPT
0098 - TRAP #6 CHARACTER READY
009C - TRAP #7 CLEAR KB BUFFER
00A0 - TRAP #8 CLEAR INTERRUPT MASK
00A4 - TRAP #9 SET LEVEL 4 INTERRUPT MASK
00A8 - TRAP #10 SET LEVEL 6 INTERRUPT MASK
00AC - TRAP #11 RESTOR TRAP #10 MASK
00FC - MOTOR OFF VECTOR (37C65)

00D40000 - 00D4FFFF  CHARACTER FONT MEMORY

00D60000 - 00D60800  MONOGRAPHICS MEMORY

00D70000 - 00DFFFFF  COLOR MEMORY
D93A01 - IDE CONTROLLER BASE ADDRESS
D93A05 - IDE ERROR REG
D93A09 - IDE SECTOR COUNT REG
D93A0D - IDE SECTOR NUMBER
D93A11 - IDE CYL # LOW
D93A15 - IDE CYL # HIGH
D93A19 - IDE SIZE, DRIVE & HEAD REG
D93A1D - IDE STATUS REG (READ)
D93A1D - IDE COMMAND REG (WRITE)
D93A3A - IDE DIGITAL OUTPUT REG
D93C01 - IDE READ/WRITE DATA REG

00F80000 - 00F8FFFF  ROM
F80000 - F87FFF MONK
F88000 - F8BFFF OS-9

00FA0641 - 00FA0645  HARD DISK CONTROLER REGS
FA0641 - DATA
FA0643 - STATUS
FA0645 - SELECT PULSE

00FA0769 - 00FA0771  MGA BOARD REGISTERS
FA0769 - INDEX
FA076B - DATA
FA0771 - CONTROL
FA0779 - PRINTER DATA PORT
FA077B - PRINTER STATUS PORT
FA077D - PRINTER CONTROL PORT

```

00FA0781 - 00FA07BD EGA & VGA REGISTERS
FA0781 - ATTRIBUTE CONTROLER
FA0785 - MISC OUTPUT REG
FA0787 - VGA ENABLE
FA0789 - SEQUENCER INDEX REG
FA078B - SEQUENCER DATA REG
FA078D - PEL MASK REG
FA078F - DAC STATE REG
FA0791 - PEL ADDRESS WRITE MODE
FA0793 - PEL DATA REG
FA0795 - GRAPHICS 2 POS REG
FA0799 - GRAPHICS 1 POS REG
FA079D - GRAPHICS INDEX REG
FA079F - GRAPHICS REGISTER (DATA)
FA07A9 - CRTC INDEX REG
FA07AB - CRTC REGISTERS (DATA)
FA07B5 - FEATURE CONTROL REG

00FA07A9 - 00FA07B1 CGA BOARD REGISTERS
FA07A9 - INDEX REG
FA07AB - DATA REGISTERS
FA07B1 - CONTROL REG

00FA07E5 - 00FA07EF 37C65 DC REGISTERS
FA07E5 - OPERATION REG
FA07E9 - MASTER STATUS REG
FA07EB - DATA & COMMAND REG
FA07EF - CONTROL REG

00FE0000 - 00FE003F DUART #1
FE0001 - MODE REG A
FE0003 - STATUS/CLOCK REG A
FE0005 - COMMAND REG A
FE0007 - RECEIVE/TRANSMIT BUFFER A
FE0009 - AUX CONTROL REG
FE000B - INTERRUPT STATUS/MASK REG
FE000D - COUNT/TIMER HIGH
FE000F - COUNT/TIMER LOW
FE0011 - MODE REG B
FE0013 - STATUS/CLOCK REG B
FE0015 - COMMAND REG B
FE0017 - RECEIVE/TRANSMIT BUFFER B
FE001B - OPCR OUTPUT PORT CONFIGURATION

00FE0040 - 00FE007F DUART #2
FE0041 - MODE REG A
FE0043 - STATUS/CLOCK REG A
FE0045 - COMMAND REG A
FE0047 - RECEIVE/TRANSMIT BUFFER A
FE0049 - AUX CONTROL REG
FE004B - INTERRUPT STATUS/MASK REG
FE004D - COUNT/TIMER HIGH
FE004F COUNT/TIMER LOW
FE0051 - MODE REG B
FE0053 - STATUS/CLOCK REG B
FE0055 - COMMAND REG B
FE0057 - RECEIVE/TRANSMIT BUFFER B
FE005B - OPCR OUTPUT PORT CONFIGURATION

00FE0080 - 00FE00B5 68230 PIA REGISTERS
FE0081 POCR GENERAL CONTROL REG
FE0085 PADDR DATA DIRECTION REG A
FE0087 PBDDR DATA DIRECTION REG B
FE008D PACR CONTROL REG A
FE008F PBCR CONTROL REG B
FE0091 PADR DATA REG A
FE0093 PBDR DATA REG B
FE009B PSR PORT STATUS REG

00FE00FE - 00FE0107 1772 DISK CONTROLER REGISTERS
FE00FE - DRIVE
FE0101 - STATUS(R), COMMAND(W)
FE0103 - TRACK
FE0105 - SECTOR
FE0107 - DATA

00FE01C1 - 00FE01C3 KEYBOARD REGISTERS
FE01C1 - STATUS
FE01C3 - DATA

```

00FF0000 - 00FF0FFF RAM (ODD BYTES ARE BATTERY BACKED UP)
00FF0000 - 00FF0003 KB OUTPUT POINTER
00FF0004 - 00FF0007 KB INPUT POINTER
00FF0008 - 00FF07D0 KEYBOARD BUFFER
00FF07E1 - 00FF07F4 37C65 DC TEMPS
           00FF07F3 IDE SECTOR SAVE ADDR
           00FF07F4 IDE DISK ADDR (H,L,T,S)
           00FF07F8 IDE DIT ADDRESS
00FF0800 - 00FF08FF BOOT BUFFER
00FF0900 - 00FF0F00 SUPERVISOR STACK
00FF0F04 - 00FF0F4B MONK TEMPS
           FFOF04 - BREAKPOINT ADDR
           FFOF08 - TRACE FLAG
           FFOF0A - BUSS ERROR TEMP
           FFOF14 - STATUS REG TEMP
           FFOF16 - LEVEL 6 INT TEMP
           FFOF18 - QUICK BP ADDR
           FFOF1C - QUICK BP CODE
           FFOF20 - CTL PRESS = TRUE
           FFOF21 - SHIFT PRESS = TRUE
           FFOF22 - ALT PRESS = TRUE
           FFOF23 - S LOCK ON = TRUE
           FFOF24 - NUM LOCK ON = TRUE
           FFOF25 - LOW INTENS = TRUE
           FFOF26 - ATTRIBUTE BYTE
           FFOF27 - WRITE PROT = TRUE
           FFOF28 - CURRENT MEM POS
           FFOF2A - OLD MEMORY POS
           FFOF2C - LINE ADDRESS
           FFOF3E - NEXT COM ADDRESS
           FFOF32 - ADAPT BASE ADDR
           FFOF36 - FILL WORD
           FFOF39 - SCREEN COLOR
           FFOF3A - CURSOR ATTRIBUTE
           FFOF3B - REV VIDEO COLOR
           FFOF49 - I/O PREFERENCE
           FFOF4B - AUTO BOOT BYTE
           FFOF52 - SPEED 70% = TRUE
           FFOF53 - VIDEO MODE
           FFOF54 - DEFAULT MODE
           FFOF55 - INTERLACE SW
           FFOF56 - OSCILLATOR BYTE
           FFOF57 - # OF COLORS
           FFOF58 - SCREEN WIDTH
           FFOF5A - SCREEN SIZE
           FFOF5E - X-RESOLUTION
           FFOF62 - Y-RESOLUTION
           FFOF66 - GRAPHICS COL #
           FFOF6A - GRAPHICS ROW #
           FFOF6E - HORIZONTAL SIZE
           FFOF70 - GRAPHICS COLOR
           FFOF7F - BYTE,WORD,LONG SW
           FFOF80 - DECODER WORK AREA
           FFOF84 - OPCODE TEMP

00FF0FF0 - 00FF0FFF CLOCK REGISTERS

```

FFOFF1 - CONTROL
FFOFF3 - SECONDS 00-59
FFOFF5 - MINUTES 00-59
FFOFF7 - HOURS 00-23
FFOFF9 - DAY 00-07
FFOFFB - DATE 01-31
FFOFFD - MONTH 01-12
FFOFFF - YEAR 00-99