

DSECTRM2.S Display Disk Sectors in Real Mode
COMPILER: NASM 2.11 [Retro Unix 386 v1 - test]

```
-----
1      ; display disk sector data [ Retro Unix 386 v1 - test ]
2      ; DSECTRM2.S
3      ; by Erdogan Tan
4      ;
5      ;; Real Mode adaption (Standalone program, 386 RM)
6      ;; (without code optimization - RM adaption is
7      ;; left defective for easy PM adaption)
8      ;;
9      ;; Last Update: 21/02/2015
10     ;;
11
12     ; 25/11/2014
13
14     ESCKey      equ 1Bh      ;27
15     ENTERKey   equ 0Dh      ;13
16     SPACEKey   equ 20h      ;32
17     BACKSPC    equ 08h      ; 8
18     DELKey     equ 53E0h
19     F1Key      equ 3B00h
20     F2Key      equ 3C00h
21     F3Key      equ 3D00h ; 19/12/2014
22     HOMEKey    equ 47E0h
23     ENDKey     equ 4FE0h
24     PgUpKey    equ 49E0h
25     PgDnKey    equ 51E0h
26
27     ; 30/12/2014 (0B000h -> 9000h)
28     ; 10/12/2014
29     DPT_SEGM   equ 9000h ; FDPT segment (EDD v1.1, EDD v3)
30     ;
31     HD0_DPT    equ 0        ; Disk parameter table address for hd0
32     HD1_DPT    equ 32       ; Disk parameter table address for hd1
33     HD2_DPT    equ 64       ; Disk parameter table address for hd2
34     HD3_DPT    equ 96       ; Disk parameter table address for hd3
35
36     ; FDPT (Phoenix, Enhanced Disk Drive Specification v1.1, v3.0)
37     ; (HDPT: Programmer's Guide to the AMIBIOS, 1993)
38     ;
39     FDPT_CYLS  equ 0 ; 1 word, number of cylinders
40     FDPT_HDS   equ 2 ; 1 byte, number of heads
41     FDPT_TT    equ 3 ; 1 byte, A0h = translated FDPT with logical values
42                 ; otherwise it is standard FDPT with physical values
43     FDPT_PCMP  equ 5 ; 1 word, starting write precompensation cylinder
44                 ; (obsolete for IDE/ATA drives)
45     FDPT_CB    equ 8 ; 1 byte, drive control byte
46                 ; Bits 7-6 : Enable or disable retries (00h = enable)
```

```

47                                     ; Bit 5 : 1 = Defect map is located at last cyl. + 1
48                                     ; Bit 4 : Reserved. Always 0
49                                     ; Bit 3 : Set to 1 if more than 8 heads
50                                     ; Bit 2-0 : Reserved. Always 0
51 FDPT_LZ      equ 12 ; 1 word, landing zone (obsolete for IDE/ATA drives)
52 FDPT_SPT     equ 14 ; 1 byte, sectors per track
53
54 [BITS 16]
55
56         ;org 100h
57
58         ; 02/12/2014
59 00000000 BE[4E29]      mov     si, prg_msg
60         ; 19/12/2014
61 00000003 E8A520      call    print_msg
62         ;
63         %include 'drvinit.inc'      ; 15/12/2014
64 <1> ; DISK I/O SYSTEM INITIALIZATION - Erdogan Tan (Retro UNIX 386 v1 project)
65 <1>
66 <1> ; ////////// DISK I/O SYSTEM STRUCTURE INITIALIZATION //////////
67 <1>
68 <1>         ; 02/02/2015
69 <1>         ; 15/01/2015
70 <1>         ; 14/01/2015
71 <1>         ; 05/01/2015
72 <1>         ; 02/01/2015
73 <1>         ; 30/12/2014
74 <1>         ; 25/12/2014 (Translated HDPT)
75 <1>         ; 23/12/2014
76 <1>         ; 20/12/2014 (Fixed Disk ports)
77 <1>         ; 19/12/2014
78 <1>         ; 13/12/2014
79 <1>         ; 10/12/2014
80 <1>         ; (unix386.s)
81 <1> L0:
82 <1>         ; 24/11/2014
83 <1>         ; 23/11/2014
84 <1>         ; 19/11/2014, 13/11/2014
85 <1>         ; 12/11/2014 (Retro UNIX 386 v1 - beginning)
86 <1>         ; Detecting disk drives... (by help of ROM-BIOS)
87 00000006 BA7F00      <1> mov     dx, 7Fh
88 <1> L1:
89 00000009 FEC2      <1> inc     dl
90 0000000B B441      <1> mov     ah, 41h ; Check extensions present
91                 <1>         ; Phoenix EDD v1.1 - EDD v3
92 0000000D BBAA55      <1> mov     bx, 55AAh
93 00000010 CD13      <1> int     13h
94 00000012 721A      <1> jc      short L2
95 00000014 81FB55AA   <1> cmp     bx, 0AA55h

```

96	00000018	7514	<1>	jne	short L2
97	0000001A	FE06[4429]	<1>	inc	byte [hdc] ; count of hard disks (EDD present)
98	0000001E	8816[4C29]	<1>	mov	[last_drv], dl ; last hard disk number
99	00000022	BB[BA28]	<1>	mov	bx, hd0_type - 80h
100	00000025	01D3	<1>	add	bx, dx
101	00000027	880F	<1>	mov	[bx], cl ; Interface support bit map in CX
102			<1>		; Bit 0 - 1, Fixed disk access subset ready
103			<1>		; Bit 1 - 1, Drv locking and ejecting ready
104			<1>		; Bit 2 - 1, Enhanced Disk Drive Support
105			<1>		; (EDD) ready (DPTE ready)
106			<1>		; Bit 3 - 1, 64bit extensions are present
107			<1>		; (EDD-3)
108			<1>		; Bit 4 to 15 - 0, Reserved
109	00000029	80FA83	<1>	cmp	dl, 83h ; drive number < 83h
110	0000002C	72DB	<1>	jb	short L1
111			<1>	L2:	
112			<1>		; 23/11/2014
113			<1>		; 19/11/2014
114	0000002E	30D2	<1>	xor	dl, dl ; 0
115	00000030	66BE[38290000]	<1>	mov	esi, fd0_type
116			<1>	L3:	
117			<1>		; 14/01/2015
118	00000036	8816[4529]	<1>	mov	[drv], dl
119			<1>		
120	0000003A	B408	<1>	mov	ah, 08h ; Return drive parameters
121	0000003C	CD13	<1>	int	13h
122	0000003E	7215	<1>	jc	short L4
123			<1>		; BL = drive type (for floppy drives)
124			<1>		; DL = number of floppy drives
125			<1>		
126			<1>		; ES:DI = Address of DPT from BIOS
127			<1>		
128	00000040	67881E	<1>	mov	[esi], bl ; Drive type
129			<1>		; 4 = 1.44 MB, 80 track, 3 1/2"
130			<1>		; 14/01/2015
131	00000043	E82F03	<1>	call	set_disk_parms
132			<1>		; 10/12/2014
133	00000046	6681FE[38290000]	<1>	cmp	esi, fd0_type
134	0000004D	7706	<1>	ja	short L4
135	0000004F	6646	<1>	inc	esi ; fd1_type
136	00000051	B201	<1>	mov	dl, 1
137	00000053	EBE1	<1>	jmp	short L3
138			<1>	L4:	
139			<1>		; Older BIOS (INT 13h, AH = 48h is not available)
140	00000055	B27F	<1>	mov	dl, 7Fh
141			<1>		; 24/12/2014 (Temporary)
142	00000057	803E[4429]00	<1>	cmp	byte [hdc], 0 ; EDD present or not ?
143	0000005C	0F879000	<1>	ja	L10 ; yes, all fixed disk operations
144			<1>		; will be performed according to

145		<1>	; present EDD specification
146		<1> L6:	
147	00000060 FEC2	<1>	inc dl
148	00000062 8816[4529]	<1>	mov [drv], dl
149	00000066 8816[4C29]	<1>	mov [last_drv], dl ; 14/01/2015
150	0000006A B408	<1>	mov ah, 08h ; Return drive parameters
151	0000006C CD13	<1>	int 13h ; (conventional function)
152	0000006E 0F825E01	<1>	jc L13 ; fixed disk drive not ready
153	00000072 8816[4429]	<1>	mov [hdc], dl ; number of drives
154		<1>	;; 14/01/2013
155		<1>	;;push cx
156	00000076 E8FC02	<1>	call set_disk_parms
157		<1>	;;pop cx
158		<1>	;
159		<1>	;;and cl, 3Fh ; sectors per track (bits 0-6)
160	00000079 8A16[4529]	<1>	mov dl, [drv]
161	0000007D BB0401	<1>	mov bx, 65*4 ; hd0 parameters table (INT 41h)
162	00000080 80FA80	<1>	cmp dl, 80h
163	00000083 7603	<1>	jna short L7
164	00000085 83C314	<1>	add bx, 5*4 ; hd1 parameters table (INT 46h)
165		<1> L7:	
166	00000088 31C0	<1>	xor ax, ax
167	0000008A 8ED8	<1>	mov ds, ax
168	0000008C 8B37	<1>	mov si, [bx]
169	0000008E 8B4702	<1>	mov ax, [bx+2]
170	00000091 8ED8	<1>	mov ds, ax
171	00000093 3A4C0E	<1>	cmp cl, [si+FDPT_SPT] ; sectors per track
172	00000096 0F853201	<1>	jne L12 ; invalid FDPT
173	0000009A BF0000	<1>	mov di, HD0_DPT
174	0000009D 80FA80	<1>	cmp dl, 80h
175	000000A0 7603	<1>	jna short L8
176	000000A2 BF2000	<1>	mov di, HD1_DPT
177		<1> L8:	
178		<1>	; 30/12/2014
179	000000A5 B80090	<1>	mov ax, DPT_SEGM
180	000000A8 8EC0	<1>	mov es, ax
181		<1>	; 24/12/2014
182	000000AA B90800	<1>	mov cx, 8
183	000000AD F3A5	<1>	rep movsw ; copy 16 bytes to the kernel's DPT location
184	000000AF 8CC8	<1>	mov ax, cs
185	000000B1 8ED8	<1>	mov ds, ax
186		<1>	; 02/02/2015
187	000000B3 8A0E[4529]	<1>	mov cl, [drv]
188	000000B7 88CB	<1>	mov bl, cl
189	000000B9 B8F001	<1>	mov ax, 1F0h
190	000000BC 80E301	<1>	and bl, 1
191	000000BF 7406	<1>	jz short L9
192	000000C1 C0E304	<1>	shl bl, 4
193	000000C4 2D8000	<1>	sub ax, 1F0h-170h

194		<1> L9:	
195	000000C7 AB	<1>	stosw ; I/O PORT Base Address (1F0h, 170h)
196	000000C8 050602	<1>	add ax, 206h
197	000000CB AB	<1>	stosw ; CONTROL PORT Address (3F6h, 376h)
198	000000CC 88D8	<1>	mov al, bl
199	000000CE 04A0	<1>	add al, 0A0h
200	000000D0 AA	<1>	stosb ; Device/Head Register upper nibble
201		<1>	;
202	000000D1 FE06[4529]	<1>	inc byte [drv]
203	000000D5 BB[BA28]	<1>	mov bx, hd0_type - 80h
204	000000D8 01CB	<1>	add bx, cx
205	000000DA 800F80	<1>	or byte [bx], 80h ; present sign (when lower nibble is 0)
206	000000DD A0[4429]	<1>	mov al, [hdc]
207	000000E0 FEC8	<1>	dec al
208	000000E2 0F84EA00	<1>	jz L13
209	000000E6 80FA80	<1>	cmp dl, 80h
210	000000E9 0F8673FF	<1>	jna L6
211	000000ED E9E000	<1>	jmp L13
212		<1> L10:	
213	000000F0 FEC2	<1>	inc dl
214		<1>	; 25/12/2014
215	000000F2 8816[4529]	<1>	mov [drv], dl
216	000000F6 B408	<1>	mov ah, 08h ; Return drive parameters
217	000000F8 CD13	<1>	int 13h ; (conventional function)
218	000000FA 0F82D200	<1>	jc L13
219		<1>	; 14/01/2015
220	000000FE 8A16[4529]	<1>	mov dl, [drv]
221	00000102 52	<1>	push dx
222	00000103 51	<1>	push cx
223	00000104 E86E02	<1>	call set_disk_parms
224	00000107 59	<1>	pop cx
225	00000108 5A	<1>	pop dx
226		<1>	;
227	00000109 66BE[2C2A0000]	<1>	mov esi, _end ; 30 byte temporary buffer address
228		<1>	; at the '_end' of kernel.
229	0000010F 67C7061E00	<1>	mov word [esi], 30
230	00000114 B448	<1>	mov ah, 48h ; Get drive parameters (EDD function)
231	00000116 CD13	<1>	int 13h
232	00000118 0F82B400	<1>	jc L13
233		<1>	; 14/01/2015
234	0000011C 6629DB	<1>	sub ebx, ebx
235	0000011F 88D3	<1>	mov bl, dl
236	00000121 80EB80	<1>	sub bl, 80h
237	00000124 6681C3[3A290000]	<1>	add ebx, hd0_type
238	0000012B 678A03	<1>	mov al, [ebx]
239	0000012E 0C80	<1>	or al, 80h
240	00000130 678803	<1>	mov [ebx], al
241	00000133 6681EB[38290000]	<1>	sub ebx, hd0_type - 2 ; 15/01/2015
242	0000013A 6681C3[3E290000]	<1>	add ebx, drv_status

243	00000141	678803	<1>	mov	[ebx], al
244	00000144	66678B4610	<1>	mov	eax, [esi+16]
245	00000149	6681EB[3E290000]	<1>	sub	ebx, drv_status
246	00000150	66C1E302	<1>	shl	ebx, 2
247	00000154	6681C3[042A0000]	<1>	add	ebx, disk_size
248	0000015B	66678903	<1>	mov	[ebx], eax
249			<1>		; 30/12/2014
250	0000015F	BF0000	<1>	mov	di, HD0_DPT
251	00000162	88D0	<1>	mov	al, dl
252	00000164	83E003	<1>	and	ax, 3
253	00000167	C0E005	<1>	shl	al, 5 ; *32
254	0000016A	01C7	<1>	add	di, ax
255	0000016C	B80090	<1>	mov	ax, DPT_SEGM
256	0000016F	8EC0	<1>	mov	es, ax
257			<1>		;
258	00000171	88E8	<1>	mov	al, ch ; max. cylinder number (bits 0-7)
259	00000173	88CC	<1>	mov	ah, cl
260	00000175	C0EC06	<1>	shr	ah, 6 ; max. cylinder number (bits 8-9)
261	00000178	40	<1>	inc	ax ; logical cylinders (limit 1024)
262	00000179	AB	<1>	stosw	
263	0000017A	88F0	<1>	mov	al, dh ; max. head number
264	0000017C	FEC0	<1>	inc	al
265	0000017E	AA	<1>	stosb	; logical heads (limits 256)
266	0000017F	B0A0	<1>	mov	al, 0A0h ; Indicates translated table
267	00000181	AA	<1>	stosb	
268	00000182	8A440C	<1>	mov	al, [si+12]
269	00000185	AA	<1>	stosb	; physical sectors per track
270	00000186	31C0	<1>	xor	ax, ax
271			<1>	;dec	ax ; 02/01/2015
272	00000188	AB	<1>	stosw	; precompensation (obsolete)
273			<1>	;xor	al, al ; 02/01/2015
274	00000189	AA	<1>	stosb	; reserved
275	0000018A	B008	<1>	mov	al, 8 ; drive control byte
276			<1>		; (do not disable retries,
277			<1>		; more than 8 heads)
278	0000018C	AA	<1>	stosb	
279	0000018D	8B4404	<1>	mov	ax, [si+4]
280	00000190	AB	<1>	stosw	; physical number of cylinders
281			<1>	;push	ax ; 02/01/2015
282	00000191	8A4408	<1>	mov	al, [si+8]
283	00000194	AA	<1>	stosb	; physical num. of heads (limit 16)
284	00000195	29C0	<1>	sub	ax, ax
285			<1>	;pop	ax ; 02/01/2015
286	00000197	AB	<1>	stosw	; landing zone (obsolete)
287	00000198	88C8	<1>	mov	al, cl ; logical sectors per track (limit 63)
288	0000019A	243F	<1>	and	al, 3Fh
289	0000019C	AA	<1>	stosb	
290			<1>	;sub	al, al ; checksum
291			<1>	;stosb	

292		<1>	;
293	0000019D 83C61A	<1>	add si, 26 ; (BIOS) DPTE address pointer
294	000001A0 AD	<1>	lodsw
295	000001A1 50	<1>	push ax ; (BIOS) DPTE offset
296	000001A2 AD	<1>	lodsw
297	000001A3 50	<1>	push ax ; (BIOS) DPTE segment
298		<1>	;
299		<1>	; checksum calculation
300	000001A4 89FE	<1>	mov si, di
301	000001A6 06	<1>	push es
302	000001A7 1F	<1>	pop ds
303		<1>	;mov cx, 16
304	000001A8 B90F00	<1>	mov cx, 15
305	000001AB 29CE	<1>	sub si, cx
306	000001AD 30E4	<1>	xor ah, ah
307		<1>	;del cl
308		<1> L11:	
309	000001AF AC	<1>	lodsb
310	000001B0 00C4	<1>	add ah, al
311	000001B2 E2FB	<1>	loop L11
312		<1>	;
313	000001B4 88E0	<1>	mov al, ah
314	000001B6 F6D8	<1>	neg al ; -x+x = 0
315	000001B8 AA	<1>	stosb ; put checksum in byte 15 of the tbl
316		<1>	;
317	000001B9 1F	<1>	pop ds ; (BIOS) DPTE segment
318	000001BA 5E	<1>	pop si ; (BIOS) DPTE offset
319		<1>	;
320		<1>	; ES:DI points to DPTE (FDPTE) location
321		<1>	;mov cx, 8
322	000001BB B108	<1>	mov cl, 8
323	000001BD F3A5	<1>	rep movsw
324		<1>	;
325	000001BF 8CC8	<1>	mov ax, cs
326	000001C1 8ED8	<1>	mov ds, ax
327	000001C3 3A16[4C29]	<1>	cmp dl, [last_drv] ; 25/12/2014
328	000001C7 7307	<1>	jnb short L13
329	000001C9 E924FF	<1>	jmp L10
330		<1> L12:	
331		<1>	; Restore data registers
332	000001CC 8CC8	<1>	mov ax, cs
333	000001CE 8ED8	<1>	mov ds, ax
334		<1> L13:	
335		<1>	; 13/12/2014
336	000001D0 0E	<1>	push cs
337	000001D1 07	<1>	pop es
338		<1> L14:	
339	000001D2 B411	<1>	mov ah, 11h
340	000001D4 CD16	<1>	int 16h

341	000001D6	7406	<1>	jz	short L14K
342	000001D8	B010	<1>	mov	al, 10h
343	000001DA	CD16	<1>	int	16h
344	000001DC	EBF4	<1>	jmp	short L14
345			<1>	L14K:	
346			<1>	; /////	
347			<1>	; 24/11/2014	
348			<1>	; 19/11/2014	
349			<1>	; 14/11/2014	
350			<1>	; Temporary code for disk searching code check	
351			<1>	;	
352			<1>	; This code will show existing (usable) drives and also	
353			<1>	; will show EDD interface support status for hard disks	
354			<1>	; (If status bit 7 is 1, Identify Device info is ready,	
355			<1>	; no need to get it again in protected mode...)	
356			<1>	;	
357			<1>	; 13/11/2014	
358	000001DE	BB0700	<1>	mov	bx, 7
359	000001E1	B40E	<1>	mov	ah, 0Eh
360	000001E3	A0[3829]	<1>	mov	al, [fd0_type]
361	000001E6	20C0	<1>	and	al, al
362	000001E8	743D	<1>	jz	short L14a
363	000001EA	88C2	<1>	mov	dl, al
364	000001EC	B046	<1>	mov	al, 'F'
365	000001EE	CD10	<1>	int	10h
366	000001F0	B044	<1>	mov	al, 'D'
367	000001F2	CD10	<1>	int	10h
368	000001F4	B030	<1>	mov	al, '0'
369	000001F6	CD10	<1>	int	10h
370	000001F8	B020	<1>	mov	al, ' '
371	000001FA	CD10	<1>	int	10h
372	000001FC	E8B200	<1>	call	L14c
373	000001FF	B020	<1>	mov	al, ' '
374	00000201	CD10	<1>	int	10h
375			<1>	;	
376	00000203	A0[3929]	<1>	mov	al, [fd1_type]
377	00000206	20C0	<1>	and	al, al
378	00000208	741D	<1>	jz	short L14a
379	0000020A	88C2	<1>	mov	dl, al
380	0000020C	B046	<1>	mov	al, 'F'
381	0000020E	CD10	<1>	int	10h
382	00000210	B044	<1>	mov	al, 'D'
383	00000212	CD10	<1>	int	10h
384	00000214	B031	<1>	mov	al, '1'
385	00000216	CD10	<1>	int	10h
386	00000218	B020	<1>	mov	al, ' '
387	0000021A	CD10	<1>	int	10h
388	0000021C	E89200	<1>	call	L14c
389	0000021F	B020	<1>	mov	al, ' '

390	00000221	CD10	<1>	int	10h
391	00000223	B020	<1>	mov	al, ' '
392	00000225	CD10	<1>	int	10h
393			<1>	L14a:	
394	00000227	A0[3A29]	<1>	mov	al, [hd0_type]
395	0000022A	20C0	<1>	and	al, al
396	0000022C	7479	<1>	jz	short L14b
397	0000022E	88C2	<1>	mov	dl, al
398	00000230	B048	<1>	mov	al, 'H'
399	00000232	CD10	<1>	int	10h
400	00000234	B044	<1>	mov	al, 'D'
401	00000236	CD10	<1>	int	10h
402	00000238	B030	<1>	mov	al, '0'
403	0000023A	CD10	<1>	int	10h
404	0000023C	B020	<1>	mov	al, ' '
405	0000023E	CD10	<1>	int	10h
406	00000240	E86E00	<1>	call	L14c
407	00000243	B020	<1>	mov	al, ' '
408	00000245	CD10	<1>	int	10h
409			<1>	;	
410	00000247	A0[3B29]	<1>	mov	al, [hd1_type]
411	0000024A	20C0	<1>	and	al, al
412	0000024C	7459	<1>	jz	short L14b
413	0000024E	88C2	<1>	mov	dl, al
414	00000250	B048	<1>	mov	al, 'H'
415	00000252	CD10	<1>	int	10h
416	00000254	B044	<1>	mov	al, 'D'
417	00000256	CD10	<1>	int	10h
418	00000258	B031	<1>	mov	al, '1'
419	0000025A	CD10	<1>	int	10h
420	0000025C	B020	<1>	mov	al, ' '
421	0000025E	CD10	<1>	int	10h
422	00000260	E84E00	<1>	call	L14c
423	00000263	B020	<1>	mov	al, ' '
424	00000265	CD10	<1>	int	10h
425			<1>	;	
426	00000267	A0[3C29]	<1>	mov	al, [hd2_type]
427	0000026A	20C0	<1>	and	al, al
428	0000026C	7439	<1>	jz	short L14b
429	0000026E	88C2	<1>	mov	dl, al
430	00000270	B048	<1>	mov	al, 'H'
431	00000272	CD10	<1>	int	10h
432	00000274	B044	<1>	mov	al, 'D'
433	00000276	CD10	<1>	int	10h
434	00000278	B032	<1>	mov	al, '2'
435	0000027A	CD10	<1>	int	10h
436	0000027C	B020	<1>	mov	al, ' '
437	0000027E	CD10	<1>	int	10h
438	00000280	E82E00	<1>	call	L14c

439	00000283	B020	<1>	mov	al, ' '
440	00000285	CD10	<1>	int	10h
441			<1>	;	
442	00000287	A0[3D29]	<1>	mov	al, [hd3_type]
443	0000028A	20C0	<1>	and	al, al
444	0000028C	7419	<1>	jz	short L14b
445	0000028E	88C2	<1>	mov	dl, al
446	00000290	B048	<1>	mov	al, 'H'
447	00000292	CD10	<1>	int	10h
448	00000294	B044	<1>	mov	al, 'D'
449	00000296	CD10	<1>	int	10h
450	00000298	B033	<1>	mov	al, '3'
451	0000029A	CD10	<1>	int	10h
452	0000029C	B020	<1>	mov	al, ' '
453	0000029E	CD10	<1>	int	10h
454	000002A0	E80E00	<1>	call	L14c
455	000002A3	B020	<1>	mov	al, ' '
456	000002A5	CD10	<1>	int	10h
457			<1>	;	
458			<1>	L14b:	
459	000002A7	B00D	<1>	mov	al, 0Dh
460	000002A9	CD10	<1>	int	10h
461	000002AB	B00A	<1>	mov	al, 0Ah
462	000002AD	CD10	<1>	int	10h
463			<1>	;;xor	ah, ah
464			<1>	;;int	16h
465			<1>	;	
466	000002AF	EB17	<1>	jmp	short L15
467			<1>	;	
468			<1>	L14c:	
469	000002B1	88D6	<1>	mov	dh, dl
470	000002B3	C0EE04	<1>	shr	dh, 4
471	000002B6	80C630	<1>	add	dh, 30h
472	000002B9	80E20F	<1>	and	dl, 15
473	000002BC	80C230	<1>	add	dl, 30h
474	000002BF	88F0	<1>	mov	al, dh
475	000002C1	CD10	<1>	int	10h
476	000002C3	88D0	<1>	mov	al, dl
477	000002C5	CD10	<1>	int	10h
478	000002C7	C3	<1>	retn	
479			<1>	;	
480			<1>	;	end of temporary code for disk searching code check
481			<1>		
482			<1>	;	/////
483			<1>		
484			<1>	L15:	
485			<1>	;	11/12/2014
486			<1>	;	SAVE ORIGINAL/OLD INTERRUPT VECTORS & POINTERS
487			<1>	;;;	

488	000002C8	1E	<1>	push	ds
489	000002C9	31C0	<1>	xor	ax, ax
490	000002CB	8ED8	<1>	mov	ds, ax
491	000002CD	BE2000	<1>	mov	si, 08h*4 ; IRQ 0 - INT 08h
492	000002D0	BF[2B0B]	<1>	mov	di, ORG_INT08H
493	000002D3	66A5	<1>	movsd	
494	000002D5	BE3800	<1>	mov	si, 0Eh*4 ; IRQ 6 - INT 0Eh
495	000002D8	BF[2F0B]	<1>	mov	di, ORG_INT0EH
496	000002DB	66A5	<1>	movsd	
497	000002DD	BE4C00	<1>	mov	si, 13h*4 ; INT 13h
498	000002E0	BF[330B]	<1>	mov	di, ORG_INT13H
499	000002E3	66A5	<1>	movsd	
500	000002E5	BE0001	<1>	mov	si, 40h*4 ; INT 40h
501	000002E8	BF[3B0B]	<1>	mov	di, ORG_INT40H
502	000002EB	66A5	<1>	movsd	
503	000002ED	BED801	<1>	mov	si, 76h*4 ; IRQ 14 - INT 76h
504	000002F0	BF[3F0B]	<1>	mov	di, ORG_INT76H
505	000002F3	66A5	<1>	movsd	
506	000002F5	BEDC01	<1>	mov	si, 77h*4 ; IRQ 15 - INT 77h
507	000002F8	BF[430B]	<1>	mov	di, ORG_INT77H
508	000002FB	66A5	<1>	movsd	
509	000002FD	1F	<1>	pop	ds
510			<1>		
511	000002FE	06	<1>	push	es
512			<1>	;sub	ax, ax
513	000002FF	8EC0	<1>	mov	es, ax
514			<1>		
515	00000301	8B16[3A29]	<1>	mov	dx, [hd0_type] ; hd0, hd1
516	00000305	8B1E[3829]	<1>	mov	bx, [fd0_type] ; fd0, fd1
517			<1>		
518			<1>		; 23/12/2014
519	00000309	21DB	<1>	and	bx, bx
520	0000030B	7428	<1>	jz	short set_int08h
521			<1>		; supress 'jmp short T6'
522			<1>		; (activate fdc motor control code)
523	0000030D	C706[3917]9090	<1>	mov	word [T5], 9090h ; nop
524			<1>		; nop
525	00000313	B8[BB0B]	<1>	mov	ax, int_40h ; DISKETTE_IO_1
526	00000316	BF4C00	<1>	mov	di, 13h*4 ; INT 13h vector
527	00000319	AB	<1>	stosw	
528	0000031A	8CC8	<1>	mov	ax, cs
529	0000031C	AB	<1>	stosw	
530			<1>		
531	0000031D	B8[9C16]	<1>	mov	ax, int_0Eh ; IRQ 6 handler
532	00000320	BF3800	<1>	mov	di, 0Eh*4 ; IRQ 6 vector
533	00000323	AB	<1>	stosw	
534	00000324	8CC8	<1>	mov	ax, cs
535	00000326	AB	<1>	stosw	
536			<1>		

537	00000327	09D2	<1>	or	dx, dx	
538	00000329	750A	<1>	jnz	short set_int08h	
539			<1>		;	
540	0000032B	B8[B80B]	<1>	mov	ax, int_40h	; DISKETTE_IO_1
541	0000032E	BF0001	<1>	mov	di, 40h*4	; INT 40h vector
542	00000331	AB	<1>	stosw		
543	00000332	8CC8	<1>	mov	ax, cs	
544	00000334	AB	<1>	stosw		
545			<1>		;	
546			<1>	set_int08h:		
547	00000335	FA	<1>	cld		
548	00000336	B8[0D17]	<1>	mov	ax, int_08h	; IRQ 0 handler
549	00000339	BF2000	<1>	mov	di, 08h*4	; IRQ 0 vector
550	0000033C	AB	<1>	stosw		
551	0000033D	8CC8	<1>	mov	ax, cs	
552	0000033F	AB	<1>	stosw		
553	00000340	FB	<1>	sti		
554			<1>		;	
555	00000341	07	<1>	pop	es	
556			<1>		;	
557	00000342	09DB	<1>	or	bx, bx	
558	00000344	7403	<1>	jz	short dsksetup	
559	00000346	E86013	<1>	CALL	DSKETTE_SETUP; Initialize Floppy Disks	
560			<1>	dsksetup:		
561	00000349	09D2	<1>	or	dx, dx	
562	0000034B	0F84A900	<1>	jz	nodsk	
563	0000034F	E83614	<1>	call	DISK_SETUP	; Initialize Fixed Disks
564	00000352	0F83A200	<1>	jnc	dsksetok	
565	00000356	BE[5F03]	<1>	mov	si, setup_error_msg	
566	00000359	E84F1D	<1>	call	print_msg	
567	0000035C	E99900	<1>	jmp	dsksetok	
568			<1>	setup_error_msg:		
569	0000035F	0D0A	<1>	db	0Dh, 0Ah	
570	00000361	4469736B2053657475-	<1>	db	'Disk Setup Error!'	
571	0000036A	70204572726F7221	<1>			
572	00000372	0D0A00	<1>	db	0Dh, 0Ah, 0	
573			<1>			
574			<1>	set_disk_parms:		
575			<1>		; 14/01/2015	
576			<1>		;push ebx	
577	00000375	6629DB	<1>	sub	ebx, ebx	
578	00000378	8A1E[4529]	<1>	mov	bl, [drv]	
579	0000037C	80FB80	<1>	cmp	bl, 80h	
580	0000037F	7203	<1>	jb	short sdp0	
581	00000381	80EB7E	<1>	sub	bl, 7Eh	
582			<1>	sdp0:		
583	00000384	6681C3[3E290000]	<1>	add	ebx, drv_status	
584	0000038B	67C60380	<1>	mov	byte [ebx], 80h	; 'Present' flag
585			<1>		;	

```

586 0000038F 88E8 <1> mov al, ch ; last cylinder (bits 0-7)
587 00000391 88CC <1> mov ah, cl ;
588 00000393 C0EC06 <1> shr ah, 6 ; last cylinder (bits 8-9)
589 00000396 6681EB[3E290000] <1> sub ebx, drv_status
590 0000039D D0E3 <1> shl bl, 1
591 0000039F 6681C3[E0290000] <1> add ebx, cylinders
592 000003A6 40 <1> inc ax ; convert max. cyl number to cyl count
593 000003A7 678903 <1> mov [ebx], ax
594 000003AA 50 <1> push ax ; ** cylinders
595 000003AB 6681EB[E0290000] <1> sub ebx, cylinders
596 000003B2 6681C3[EC290000] <1> add ebx, heads
597 000003B9 30E4 <1> xor ah, ah
598 000003BB 88F0 <1> mov al, dh ; heads
599 000003BD 40 <1> inc ax
600 000003BE 678903 <1> mov [ebx], ax
601 000003C1 6681EB[EC290000] <1> sub ebx, heads
602 000003C8 6681C3[F8290000] <1> add ebx, spt
603 000003CF 30ED <1> xor ch, ch
604 000003D1 80E13F <1> and cl, 3Fh ; sectors (bits 0-6)
605 000003D4 67890B <1> mov [ebx], cx
606 000003D7 6681EB[F8290000] <1> sub ebx, spt
607 000003DE 66D1E3 <1> shl ebx, 1
608 000003E1 6681C3[042A0000] <1> add ebx, disk_size
609 <1> ; LBA size = cylinders * heads * secpertrack
610 000003E8 F7E1 <1> mul cx
611 000003EA 89C2 <1> mov dx, ax ; heads*spt
612 000003EC 58 <1> pop ax ; ** cylinders
613 000003ED 48 <1> dec ax ; 1 cylinder reserved (!?)
614 000003EE F7E2 <1> mul dx ; cylinders * (heads*spt)
615 000003F0 678903 <1> mov [ebx], ax
616 000003F3 67895302 <1> mov [ebx+2], dx
617 <1> ;
618 <1> ;pop ebx
619 000003F7 C3 <1> retn
620 <1>
621 <1> dsksetok:
622 <1> ;?
623 <1> nodsk:
624
625 ; ; 19/12/2014
626 ;display_dskprmt:
627 ; ;15/12/2014
628 ; xor dl, dl
629 ; ; 19/12/2014
630 ; ; Display disk parameters table
631 ;dskprmfmt:
632 ; ; mov byte [drv], dl ; 0
633 ; call dskprm
634 ; jc short dskprmh

```

```

635 ; cmp byte [drv], 0
636 ; ja short dskprmh
637 ; dec dl ; floppy drive count - 1
638 ; jz short dskprmh ; only 1 floppy drive
639 ; mov dl, 1
640 ; jmp short dskprmf
641 ;dskprmh:
642 ; mov dl, 80h
643 ;dskprmh:
644 ; ;mov byte [drv], dl ; 80h
645 ; call dskprm
646 ; jc short sccps
647 ; mov dl, byte [drv]
648 ; ; 25/12/2014
649 ; cmp dl, [last_drv]
650 ; jnb short sccps
651 ; inc dl
652 ; jmp short dskprmh
653 ; ;
654 ;sccps:
655 ; ; 28/11/2014
656 000003F8 30FF xor bh, bh
657 000003FA B403 mov ah, 03h ; get cursor position and shape
658 000003FC CD10 int 10h
659 000003FE 8916[3229] mov [cursor_posn], dx ; position
660 00000402 890E[3429] mov [cursor_shp], cx ; shape
661
662 display_sectors:
663 ; ; 04/12/2014
664 ; ; 02/12/2014
665 ; ; 01/12/2014
666 ; ; 28/11/2014
667 ; ; 27/11/2014
668 ; ; 26/11/2014
669 ;
670 00000406 E8F005 call hide_cursor
671 ; Save video page (before displaying sector)
672 00000409 1E push ds ; *
673 0000040A B800B8 mov ax, 0B800h
674 0000040D 8ED8 mov ds, ax ; *
675 0000040F 8EC0 mov es, ax ; *
676
677 ;mov esi, 0B8000h
678 00000411 6629F6 sub esi, esi ; *
679
680 00000414 6689F7 mov edi, esi
681 00000417 6681C7C05D0000 add edi, 5DC0h ; 4000*6 (video page 6)
682 0000041E 66B9E8030000 mov ecx, 1000
683 00000424 F366A5 rep movsd

```

684 00000427 1F	pop ds ; *
685	; Save cursor position
686 00000428 A1[3229]	mov ax, [cursor_posn] ; cursor pos.
687	; for video page 0
688 0000042B A3[3629]	mov [cursor_posb], ax
689 0000042E E8E604	call clear_frame
690 00000431 EB25	jmp short dscl_0
691	dscl_esc:
692 00000433 E89303	call restore_video_page
693	dscl_getc:
694 00000436 E86B03	call getch
695	;
696 00000439 3C1B	cmp al, ESCKey
697 0000043B 0F84F802	je dscl_exit
698 0000043F C606[0728]00	mov byte [dscmd], 0
699 00000444 3D003B	cmp ax, F1Key
700 00000447 740F	je short dscl_0
701	; 19/12/2014
702 00000449 FE06[0728]	inc byte [dscmd]
703 0000044D 3D003D	cmp ax, F3Key
704 00000450 0F853502	jne dscl_5
705 00000454 FE06[0728]	inc byte [dscmd] ; Display disk params.
706	dscl_0:
707 00000458 E84E03	call save_video_page
708 0000045B 66BE[F3280000]	mov esi, F1_ib ; F1 (Change drive)
709	; Inputbox address
710	dscl_ib:
711 00000461 E8FC04	call inputbox
712	; cursor position in DX
713 00000464 E89605	call show_cursor
714	; cursor blinks at current position
715	dscl_3:
716 00000467 E83A03	call getch
717 0000046A 3C1B	cmp al, ESCKey
718 0000046C 7505	jne short dscl_27
719 0000046E E88805	call hide_cursor
720 00000471 EBC0	jmp short dscl_esc
721	dscl_27:
722 00000473 3C20	cmp al, SPACEKey
723 00000475 743A	je short dscl_4
724 00000477 3C0D	cmp al, ENTERKey
725 00000479 7436	je short dscl_4
726	;
727 0000047B 6631DB	xor ebx, ebx ; 20/02/2015
728 0000047E 803E[0728]01	cmp byte [dscmd], 1
729 00000483 7457	je short dscl_12
730	;
731 00000485 3C30	cmp al, '0'
732 00000487 72DE	jb short dscl_3

733 00000489 3C35	cmp al, '5'
734 0000048B 77DA	ja short dscl_3
735 0000048D 668B3E[0228]	mov edi, [current_txtpos]
736 00000492 AA	stosb
737 00000493 2C30	sub al, '0'
738 00000495 88C2	mov dl, al
739 00000497 30F6	xor dh, dh
740 00000499 88C3	mov bl, al
741 0000049B C0E302	shl bl, 2 ; *4
742 0000049E 6681C3[B0270000]	add ebx, ds_sec ; current_sector
743 000004A5 66678B0B	mov ecx, [ebx]
744 000004A9 66BE[2C2A0000]	mov esi, buffer
745 000004AF EBB6	jmp short dscl_3
746	dscl_4:
747 000004B1 803E[AD27]00	cmp byte [inds], 0 ; display other half or not ?
748 000004B6 0F87F900	ja dscl_oh ; other half
749 000004BA 6651	push ecx
750	; save regs (ESI, EAX, DX)
751 000004BC E83A05	call hide_cursor
752	; restore regs (ESI, EAX, DX)
753 000004BF 6658	pop eax
754	; 19/12/2014
755 000004C1 803E[0728]01	cmp byte [dscmd], 1 ; Requested function ?
756 000004C6 0F849E00	je dscl_17 ; Change sector (F2)
757 000004CA 0F821E01	jb dscl_ns ; Change drive (F1)
758	; Display disk parameters (dscmd = 2)
759	; 31/12/2014
760 000004CE 80FA02	cmp dl, 2
761 000004D1 7203	jb short dscl_28
762 000004D3 80C27E	add dl, 7Eh
763	dscl_28:
764 000004D6 E8B719	call dskprm
765 000004D9 E957FF	jmp dscl_esc
766	dscl_12:
767 000004DC 3DE053	cmp ax, DELKey ; DEL key
768 000004DF 7404	je short dscl_bs
769 000004E1 3C08	cmp al, BACKSPC ; Backspace key
770 000004E3 752B	jne short dscl_13
771	dscl_bs:
772 000004E5 803E[0628]00	cmp byte [txtposoff], 0
773 000004EA 0F8679FF	jna dscl_3
774 000004EE FE0E[0628]	dec byte [txtposoff]
775 000004F2 FE0E[3229]	dec byte [cursor_posn]
776 000004F6 8B16[3229]	mov dx, [cursor_posn]
777 000004FA E82A24	call set_cpos
778 000004FD 6631DB	xor ebx, ebx
779 00000500 8A1E[0628]	mov bl, [txtposoff]
780 00000504 FE0E[0628]	dec byte [txtposoff]
781 00000508 FE0E[3229]	dec byte [cursor_posn]

782	0000050C	B020	mov	al, 20h
783	0000050E	EB15	jmp	short dscl_14
784			dscl_13:	
785	00000510	8A1E[0628]	mov	bl, [txtposoff]
786	00000514	80FB08	cmp	bl, 8
787	00000517	0F834CFF	jnb	dscl_3
788			;	
789	0000051B	3C30	cmp	al, '0'
790	0000051D	0F8246FF	jb	dscl_3
791	00000521	3C39	cmp	al, '9'
792	00000523	7729	ja	short dscl_15
793			dscl_14:	
794	00000525	D0E3	shl	bl, 1
795	00000527	668B36[0228]	mov	esi, [current_txtpos]
796	0000052C	6601F3	add	ebx, esi
797	0000052F	26678803	mov	[es:ebx], al
798			;	
799	00000533	803E[0628]08	cmp	byte [txtposoff], 8
800	00000538	0F8D2BFF	jge	dscl_3 ; JGE !
801	0000053C	FE06[0628]	inc	byte [txtposoff]
802	00000540	FE06[3229]	inc	byte [cursor_posn]
803	00000544	8B16[3229]	mov	dx, [cursor_posn]
804	00000548	E8DC23	call	set_cpos
805	0000054B	E919FF	jmp	dscl_3
806			dscl_15:	
807	0000054E	3C41	cmp	al, 'A'
808	00000550	0F8213FF	jb	dscl_3
809	00000554	3C46	cmp	al, 'F'
810	00000556	76CD	jna	short dscl_14
811			dscl_16:	
812	00000558	3C61	cmp	al, 'a'
813	0000055A	0F8209FF	jb	dscl_3
814	0000055E	3C66	cmp	al, 'f'
815	00000560	0F8703FF	ja	dscl_3
816	00000564	2C20	sub	al, 'a' - 'A'
817	00000566	EBBD	jmp	short dscl_14
818			;	
819			dscl_17:	
820	00000568	668B36[0228]	mov	esi, [current_txtpos]
821	0000056D	6631C0	xor	eax, eax
822	00000570	A2[0628]	mov	byte [txtposoff], al ; 0
823	00000573	6650	push	eax ; sector value (reset)
824	00000575	B800B8	mov	ax, 0B800h ; *
825	00000578	8ED8	mov	ds, ax ; *
826			dscl_18:	
827	0000057A	AD	lodsw	
828	0000057B	3C30	cmp	al, '0'
829	0000057D	7222	jb	short dscl_22
830			dscl_19:	

831	0000057F	6629C9	sub	ecx, ecx
832	00000582	66BB[9D270000]	mov	ebx, hexchrs
833			dsc1_20:	
834	00000588	2E673A03	cmp	al, [cs:ebx]
835	0000058C	7406	je	short dsc1_21
836			;cmp	cl, 15
837			;jnb	short dsc1_22
838	0000058E	FEC1	inc	cl
839	00000590	6643	inc	ebx
840	00000592	EBF4	jmp	short dsc1_20
841			dsc1_21:	
842	00000594	6658	pop	eax
843	00000596	66C1E004	shl	eax, 4 ; * 16
844	0000059A	6601C8	add	eax, ecx
845	0000059D	6650	push	eax
846	0000059F	EBD9	jmp	short dsc1_18
847			dsc1_22:	
848	000005A1	0E	push	cs
849	000005A2	1F	pop	ds
850	000005A3	8A16[AE27]	mov	dl, [ds_drv]
851	000005A7	30F6	xor	dh, dh
852	000005A9	6658	pop	eax
853	000005AB	66BE[2C2A0000]	mov	esi, buffer
854	000005B1	EB39	jmp	short dsc1_ns
855			dsc1_oh:	
856	000005B3	8A16[AE27]	mov	dl, [ds_drv]
857	000005B7	6631DB	xor	ebx, ebx
858	000005BA	88D3	mov	bl, dl
859	000005BC	C0E302	shl	bl, 2
860	000005BF	6681C3[B0270000]	add	ebx, ds_sec
861	000005C6	66678B03	mov	eax, [ebx]
862	000005CA	66BE[2C2A0000]	mov	esi, buffer
863			;	
864	000005D0	8A36[AF27]	mov	dh, [ds_drv+1]
865	000005D4	08F6	or	dh, dh
866	000005D6	7404	jz	short dsc1_nh ; second half of sector (0->1)
867	000005D8	30F6	xor	dh, dh ; reset (0)
868	000005DA	EB09	jmp	short dsc1_nx
869			dsc1_nh:	
870	000005DC	6681C600010000	add	esi, 256
871	000005E3	FEC6	inc	dh
872			dsc1_nx:	
873	000005E5	8836[AF27]	mov	[ds_drv+1], dh
874	000005E9	E98500	jmp	dsc1_25
875			dsc1_ns:	
876	000005EC	8836[AF27]	mov	[ds_drv+1], dh
877	000005F0	6631DB	xor	ebx, ebx
878	000005F3	88D3	mov	bl, dl
879	000005F5	C0E302	shl	bl, 2

880 000005F8 6681C3[B0270000]	add ebx, ds_sec
881 000005FF 3A16[AE27]	cmp dl, [ds_drv]
882 00000603 7506	jne short dscl_23
883 00000605 66673B03	cmp eax, [ebx]
884 00000609 7466	je short dscl_25
885	dscl_23:
886 0000060B 8816[AE27]	mov [ds_drv], dl
887	dscl_26:
888 0000060F 66678B0B	mov ecx, [ebx]
889 00000613 66890E[4829]	mov [prev_sec], ecx
890 00000618 66678903	mov [ebx], eax
891 0000061C E8F203	call read_disk_sector
892 0000061F 7333	jnc short dscl_24
893	dscl_rd_err:
894	;
895	;; Temporary, 15/12/2014
896 00000621 88E0	mov al, ah ; error code
897 00000623 BF[2029]	mov di, err_code_str
898 00000626 E80E19	call write_hex
899	;
900 00000629 66BE[11290000]	mov esi, dskr_err ; drive not ready or read error
901 0000062F E82E03	call inputbox
902 00000632 E86F01	call getch
903 00000635 E89101	call restore_video_page
904 00000638 6629DB	sub ebx, ebx
905 0000063B 8A1E[AE27]	mov bl, [ds_drv]
906 0000063F C0E302	shl bl, 2
907 00000642 6681C3[B0270000]	add ebx, ds_sec
908 00000649 66A1[4829]	mov eax, [prev_sec]
909 0000064D 66678903	mov [ebx], eax
910 00000651 E9E2FD	jmp dscl_getc
911	dscl_24:
912 00000654 8B16[AE27]	mov dx, [ds_drv]
913 00000658 6631DB	xor ebx, ebx
914 0000065B 88D3	mov bl, dl
915 0000065D C0E302	shl bl, 2
916 00000660 6681C3[B0270000]	add ebx, ds_sec
917 00000667 66678B03	mov eax, [ebx]
918 0000066B 66BE[2C2A0000]	mov esi, buffer
919	dscl_25:
920 00000671 E87501	call display_sector
921 00000674 E83201	call save_video_page
922 00000677 E9BCFD	jmp dscl_getc
923	dscl_11:
924 0000067A 66BE[2C2A0000]	mov esi, buffer
925 00000680 8A16[AE27]	mov dl, [ds_drv]
926 00000684 28F6	sub dh, dh ; 0 = first half of sector
927 00000686 E963FF	jmp dscl_ns
928	dscl_5:

929	00000689	3D003C	cmp	ax, F2Key
930	0000068C	7511	jne	short dscl_6
931	0000068E	E81801	call	save_video_page
932	00000691	66BE[01290000]	mov	esi, F2_ib ; F2 (Change sector)
933				; Inputbox address
934	00000697	C606[0728]01	mov	byte [dscmd], 1
935	0000069C	E9C2FD	jmp	dscl_ib
936			dscl_6:	
937	0000069F	3C20	cmp	al, SPACEKey
938	000006A1	0F840EFF	je	dscl_oh
939	000006A5	3C0D	cmp	al, ENTERKey
940	000006A7	0F8408FF	je	dscl_oh
941				;
942	000006AB	3DE047	cmp	ax, HOMEKey
943	000006AE	7505	jne	short dscl_7
944	000006B0	6631C0	xor	eax, eax
945	000006B3	EBC5	jmp	short dscl_11
946			dscl_7:	
947	000006B5	3DE04F	cmp	ax, ENDKey
948	000006B8	7519	jne	short dscl_8
949	000006BA	6631DB	xor	ebx, ebx
950	000006BD	8A1E[AE27]	mov	bl, [ds_drv]
951	000006C1	C0E302	shl	bl, 2
952	000006C4	6681C3[042A0000]	add	ebx, disk_size
953	000006CB	66678B03	mov	eax, [ebx]
954	000006CF	6648	dec	eax
955	000006D1	EBA7	jmp	short dscl_11
956			dscl_8:	
957	000006D3	3DE051	cmp	ax, PgDnKey
958	000006D6	7546	jne	short dscl_10
959	000006D8	E80F00	call	dscl_9
960	000006DB	6640	inc	eax
961	000006DD	6639C8	cmp	eax, ecx ; last sector
962	000006E0	0F8608FF	jna	dscl_ns
963	000006E4	6631C0	xor	eax, eax
964	000006E7	E925FF	jmp	dscl_26
965			dscl_9:	
966	000006EA	8A16[AE27]	mov	dl, [ds_drv]
967	000006EE	30F6	xor	dh, dh
968	000006F0	6629DB	sub	ebx, ebx
969	000006F3	88D3	mov	bl, dl
970	000006F5	C0E302	shl	bl, 2 ; *4
971	000006F8	6681C3[042A0000]	add	ebx, disk_size
972	000006FF	66678B0B	mov	ecx, [ebx]
973	00000703	6649	dec	ecx
974	00000705	6681EB[042A0000]	sub	ebx, disk_size
975	0000070C	6681C3[B0270000]	add	ebx, ds_sec ; current sector
976	00000713	66678B03	mov	eax, [ebx]
977	00000717	66BE[2C2A0000]	mov	esi, buffer

978	0000071D	C3		retn
979			dsc1_10:	
980	0000071E	3DE049		cmp ax, PgUpKey
981	00000721	0F8511FD		jne dsc1_getc
982	00000725	E8C2FF		call dsc1_9
983	00000728	6648		dec eax
984	0000072A	6639C8		cmp eax, ecx ; last sector
985	0000072D	0F86BBFE		jna dsc1_ns
986	00000731	6689C8		mov eax, ecx
987	00000734	E9D8FE		jmp dsc1_26
988				
989			dsc1_exit:	
990				;
991				; 11/12/2014
992				; RESTORE ORIGINAL/OLD INTERRUPT VECTORS & POINTERS
993				;;;
994				;push es
995	00000737	31C0		xor ax, ax
996	00000739	8EC0		mov es, ax
997	0000073B	BF2000		mov di, 08h*4 ; IRQ 0 - INT 08h
998	0000073E	BE[2B0B]		mov si, ORG_INT08H
999	00000741	66A5		movsd
1000	00000743	BF3800		mov di, 0Eh*4 ; IRQ 6 - INT 0Eh
1001	00000746	BE[2F0B]		mov si, ORG_INT0EH
1002	00000749	66A5		movsd
1003	0000074B	BF4C00		mov di, 13h*4 ; INT 13h
1004	0000074E	BE[330B]		mov si, ORG_INT13H
1005	00000751	66A5		movsd
1006	00000753	BF0001		mov di, 40h*4 ; INT 40h
1007	00000756	BE[3B0B]		mov si, ORG_INT40H
1008	00000759	66A5		movsd
1009	0000075B	BFD801		mov di, 76h*4 ; IRQ 14 - INT 76h
1010	0000075E	BE[3F0B]		mov si, ORG_INT76H
1011	00000761	66A5		movsd
1012	00000763	BFD001		mov di, 77h*4 ; IRQ 15 - INT 77h
1013	00000766	BE[430B]		mov si, ORG_INT77H
1014	00000769	66A5		movsd
1015				;pop es
1016				;
1017				; Restore video page (before displaying sector)
1018				;mov edi, 0B8000h
1019	0000076B	6629FF		sub edi, edi ; *
1020	0000076E	B800B8		mov ax, 0B800h ; *
1021	00000771	1E		push ds ; *
1022	00000772	8EC0		mov es, ax
1023	00000774	8ED8		mov ds, ax
1024	00000776	6689FE		mov esi, edi
1025	00000779	6681C6C05D0000		add esi, 5DC0h ; 4000*6 (video page 6)
1026	00000780	66B9E8030000		mov ecx, 1000

1027	00000786	F366A5	rep	movsd
1028	00000789	1F	pop	ds ; *
1029				; Restore cursor position
1030	0000078A	8B16[3629]	mov	dx, [cursor_posb] ; cursor pos.
1031				; for video page 6.
1032	0000078E	8916[3229]	mov	[cursor_posn], dx ; for video page 0.
1033				;
1034				; Set cursor position
1035	00000792	30FF	xor	bh, bh ; Video page 0
1036	00000794	B402	mov	ah, 2
1037	00000796	CD10	int	10h
1038				; Show standard blinking text cursor
1039				;mov ch, 6
1040				;mov cl, 7
1041	00000798	8B0E[3429]	mov	cx, [cursor_shp]
1042	0000079C	B401	mov	ah, 1
1043	0000079E	CD10	int	10h
1044			terminate:	
1045				;
1046	000007A0	CD20	int	20h ; Terminate
1047				;
1048	000007A2	CD19	int	19h
1049			;hltsys:	
1050				; hlt
1051				; jmp short hltsys
1052				
1053			getch:	
1054				; 28/11/2014 (DOS version)
1055				; Getchar by using INT 16h
1056	000007A4	B410	mov	ah, 10h
1057	000007A6	CD16	int	16h
1058	000007A8	C3	retn	
1059				
1060			save_video_page:	
1061				; 26/11/2014
1062				;
1063				; Save video page
1064				;mov esi, 0B8000h
1065	000007A9	6629F6	sub	esi, esi ; *
1066	000007AC	1E	push	ds ; *
1067	000007AD	B800B8	mov	ax, 0B800h ; *
1068	000007B0	8EC0	mov	es, ax ; *
1069	000007B2	8ED8	mov	ds, ax ; *
1070	000007B4	6689F7	mov	edi, esi
1071	000007B7	6681C7606D0000	add	edi, 6D60h ; 4000*7 (video page 7)
1072	000007BE	66B9E8030000	mov	ecx, 1000
1073	000007C4	F366A5	rep	movsd
1074	000007C7	1F	pop	ds ; *
1075	000007C8	C3	retn	

```

1076
1077
1078
1079
1080
1081 000007C9 6629F6
1082 000007CC B800B8
1083 000007CF 1E
1084 000007D0 8ED8
1085 000007D2 8EC0
1086 000007D4 6689F7
1087 000007D7 6681C6606D0000
1088 000007DE 66B9E8030000
1089 000007E4 F366A5
1090 000007E7 1F
1091 000007E8 C3
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118 000007E9 C606[AD27]01
1119
1120 000007EE 6650
1121 000007F0 E82401
1122 000007F3 6658
1123
1124 000007F5 6631DB

restore_video_page:
; 02/12/2014
; 26/11/2014
;mov esi, 0B8000h
sub esi, esi ; *
mov ax, 0B800h
push ds ; *
mov ds, ax ; *
mov es, ax ; *
mov edi, esi
add esi, 6D60h ; 4000*7 (video page 7)
mov ecx, 1000
rep movsd
pop ds ; *
retn

display_sector:
; 04/12/2014
; 30/11/2014
; 28/11/2014
; 25/11/2014
; display disk sector data (on tty0)
;
; INPUT ->
; ESI = sector buffer offset
; (sector size: 512 bytes)
; EAX = sector number
; DL = drive number (0,1,2,3,4,5,6)
; DH = portion control byte
; (0= first half of the sector,
; >0= second half of the sector)
; OUTPUT ->
; Video page 0 (0B8000h) will be filled
; with sector data
; (ESI points to byte 256 of the buffer
; or end of the buffer)
;
; Modified registers: eax, edx, ecx, ebx, esi, edi
;
;
;xor ecx, ecx ; reset for cx loop counts
mov byte [inds], 1 ; for ENTER key handling
;
push eax
call clear_frame
pop eax

dsfh:
xor ebx, ebx

```

1125	000007F8	08F6	or	dh, dh
1126	000007FA	7402	jz	short dsfh1
1127	000007FC	B310	mov	bl, 10h
1128			dsfh1:	
1129	000007FE	881E[C827]	mov	[paragr], bl ; Paragraph (16 bytes)
1130			;	
1131	00000802	88D3	mov	bl, dl
1132	00000804	C0E302	shl	bl, 2 ; *4
1133	00000807	6681C3[C9270000]	add	ebx, drv_names
1134	0000080E	66678B13	mov	edx, [ebx]
1135	00000812	668916[EA27]	mov	[drv_name], edx
1136	00000817	E8C200	call	dwordtohex
1137	0000081A	668916[F827]	mov	[sector_num], edx
1138	0000081F	66A3[FC27]	mov	[sector_num+4], eax
1139	00000823	B001	mov	al, 1
1140	00000825	B43F	mov	ah, 3Fh ; cyan background, white forecolor
1141	00000827	66BB[E1270000]	mov	ebx, dpheader
1142	0000082D	E8D400	call	print_line
1143	00000830	B015	mov	al, 21
1144			;mov	ah, 3Fh ; cyan background, white forecolor
1145	00000832	66BB[58280000]	mov	ebx, dpfooter1
1146	00000838	E8C900	call	print_line
1147	0000083B	B016	mov	al, 22
1148			;mov	ah, 3Fh ; cyan background, white forecolor
1149	0000083D	66BB[A2280000]	mov	ebx, dpfooter2
1150	00000843	E8BE00	call	print_line
1151			ds1:	
1152	00000846	66B910000000	mov	ecx, 16
1153			ds2:	
1154	0000084C	A0[C827]	mov	al, [paragr]
1155	0000084F	E84100	call	byteto hex
1156	00000852	A3[0E28]	mov	[sdline_1], ax
1157			;	
1158	00000855	6651	push	ecx
1159	00000857	1E	push	ds ; *
1160	00000858	07	pop	es ; *
1161	00000859	B110	mov	cl, 16
1162	0000085B	66BF[16280000]	mov	edi, sdline_2
1163			ds3:	
1164	00000861	AC	lodsb	
1165	00000862	E82E00	call	byteto hex
1166	00000865	AB	stosw	
1167	00000866	6647	inc	edi
1168	00000868	E2F7	loop	ds3
1169	0000086A	6683EE10	sub	esi, 16
1170	0000086E	6647	inc	edi
1171	00000870	B110	mov	cl, 16
1172	00000872	F3A4	rep	movsb
1173	00000874	6659	pop	ecx

1174	00000876	B800B8	mov	ax, 0B800h ; *
1175	00000879	8EC0	mov	es, ax ; *
1176	0000087B	B013	mov	al, 19 ; line (row) 3 to 24
1177	0000087D	28C8	sub	al, cl
1178	0000087F	B407	mov	ah, 07h ; Black background, light gray forecolor
1179	00000881	66BB[08280000]	mov	ebx, sdline
1180	00000887	E86400	call	print_line_80 ; 04/12/2014
1181	0000088A	E201	loop	ds4
1182	0000088C	C3	retn	
1183			ds4:	
1184	0000088D	FE06[C827]	inc	byte [paragr]
1185	00000891	EBB9	jmp	short ds2
1186				
1187				
1188				; Convert binary number to hexadecimal string
1189				; 01/12/2014
1190				; 25/11/2014
1191				;
1192				bytetohe:
1193				; INPUT ->
1194				; AL = byte (binary number)
1195				; OUTPUT ->
1196				; AX = hexadecimal string
1197	00000893	6653	push	ebx
1198	00000895	6631DB	xor	ebx, ebx
1199	00000898	88C3	mov	bl, al
1200	0000089A	C0EB04	shr	bl, 4
1201	0000089D	678A9B[9D270000]	mov	bl, [ebx+hexchrs]
1202	000008A4	86D8	xchg	bl, al
1203	000008A6	80E30F	and	bl, 0Fh
1204	000008A9	678AA3[9D270000]	mov	ah, [ebx+hexchrs]
1205	000008B0	665B	pop	ebx
1206	000008B2	C3	retn	
1207				
1208				wordtohe:
1209				; INPUT ->
1210				; AX = word (binary number)
1211				; OUTPUT ->
1212				; EAX = hexadecimal string
1213				;
1214	000008B3	6653	push	ebx
1215	000008B5	6631DB	xor	ebx, ebx
1216	000008B8	86E0	xchg	ah, al
1217	000008BA	50	push	ax
1218	000008BB	88E3	mov	bl, ah
1219	000008BD	C0EB04	shr	bl, 4
1220	000008C0	678A83[9D270000]	mov	al, [ebx+hexchrs]
1221	000008C7	88E3	mov	bl, ah
1222	000008C9	80E30F	and	bl, 0Fh

1223 000008CC 678AA3[9D270000]	mov ah, [ebx+hexchrs]
1224 000008D3 66C1E010	shl eax, 16
1225 000008D7 58	pop ax
1226 000008D8 665B	pop ebx
1227 000008DA EBB7	jmp short byteto hex
1228	;mov bl, al
1229	;shr bl, 4
1230	;mov bl, [ebx+hexchrs]
1231	;xchg bl, al
1232	;and bl, 0Fh
1233	;mov ah, [ebx+hexchrs]
1234	;pop ebx
1235	;retn
1236	
1237	dwordtohex:
1238	; INPUT ->
1239	; EAX = dword (binary number)
1240	; OUTPUT ->
1241	; EDX:EAX = hexadecimal string
1242	;
1243 000008DC 6650	push eax
1244 000008DE 66C1E810	shr eax, 16
1245 000008E2 E8CEFF	call wordtohex
1246 000008E5 6689C2	mov edx, eax
1247 000008E8 6658	pop eax
1248 000008EA E8C6FF	call wordtohex
1249 000008ED C3	retn
1250	
1251	print_line_80:
1252	; 04/12/2014
1253	; al = line (0 to 24)
1254	; ah = color attributes
1255	; ebx = 80 chars string address
1256 000008EE E85E00	call get_lpos
1257 000008F1 6651	push ecx
1258 000008F3 66B950000000	mov ecx, 80
1259	pl80:
1260 000008F9 678A03	mov al, [ebx]
1261 000008FC 6643	inc ebx
1262 000008FE AB	stosw
1263 000008FF E2F8	loop pl80
1264 00000901 6659	pop ecx
1265 00000903 C3	retn
1266	
1267	print_line:
1268	; al = line (0 to 24)
1269	; ah = color attributes
1270	; ebx = ASCIIZ string address
1271 00000904 E84800	call get_lpos

```

1272 00000907 6656
1273 00000909 6689DE
1274
1275 0000090C AC
1276 0000090D 20C0
1277 0000090F 7403
1278 00000911 AB
1279 00000912 EBF8
1280
1281 00000914 665E
1282 00000916 C3
1283
1284
1285 00000917 30C0
1286 00000919 E82600
1287 0000091C B001
1288 0000091E B43F
1289 00000920 E82100
1290 00000923 B001
1291
1292 00000925 40
1293 00000926 50
1294 00000927 E81800
1295 0000092A 58
1296 0000092B 3C13
1297 0000092D 72F6
1298
1299 0000092F B43F
1300
1301 00000931 FEC0
1302 00000933 50
1303 00000934 E80D00
1304 00000937 58
1305 00000938 3C17
1306 0000093A 72F5
1307 0000093C FEC0
1308 0000093E E80100
1309 00000941 C3
1310
1311
1312
1313 00000942 30E4
1314
1315
1316
1317 00000944 E80800
1318 00000947 B95000
1319 0000094A B020
1320 0000094C F3AB

```

```

        push    esi
        mov     esi,ebx
prl1:   lodsb
        and     al, al
        jz      short prl2
        stosw
        jmp     short prl1
prl2:   pop     esi
        retn

clear_frame:
        xor     al, al ; Line 0
        call    clear_line
        mov     al, 1
        mov     ah, 3Fh ; cyan background, white forecolor
        call    fill_color
        mov     al, 1
dscf0:  inc     ax
        push    ax
        call    clear_line
        pop     ax
        cmp     al, 19
        jb      short dscf0
        ;inc    al ; line 20
        mov     ah, 3Fh
dscf1:  inc     al
        push    ax
        call    fill_color
        pop     ax
        cmp     al, 23
        jb      short dscf1
        inc     al
        call    clear_line
        retn
        ;

clear_line:
        xor     ah, ah ; blank
fill_color:
        ; al = line (0 to 24)
        ; ah = color attributes
        call    get_lpos
        mov     cx, 80
        mov     al, 20h ; space/blank
        rep     stosw

```

1321 0000094E C3	retn
1322	
1323	get_lpos: ; Get line position in video buffer
1324 0000094F 50	push ax
1325 00000950 6631FF	xor edi, edi
1326 00000953 B4A0	mov ah, 80*2
1327 00000955 F6E4	mul ah
1328 00000957 89C7	mov di, ax
1329 00000959 B800B8	mov ax, 0B800h ; *
1330 0000095C 8EC0	mov es, ax ; *
1331 0000095E 58	pop ax
1332	;add edi, 0B8000h ; video page 0
1333	; *
1334 0000095F C3	retn
1335	
1336	inputbox:
1337	; 27/11/2014
1338	; Show an input box for user/keyboard input
1339	; INPUT ->
1340	; ESI = input structure address
1341	; OUTPUT ->
1342	; DX = cursor position for input
1343	; input box will be displayed (on tty0)
1344	;
1345	; Modified registers: ax, ebx, ecx, dx, esi, edi
1346	;
1347	
1348 00000960 C606[AD27]00	mov byte [inds], 0 ; for ENTER key handling
1349	;xor ecx, ecx
1350 00000965 31C9	xor cx, cx ; ecx = 0
1351	;mov ebx, 0B8000h
1352 00000967 B800B8	mov ax, 0B800h ; *
1353 0000096A 6631DB	xor ebx, ebx ; *
1354 0000096D 8EC0	mov es, ax ; *
1355 0000096F B81850	mov ax, 5018h ; 80, 24
1356 00000972 678B16	mov dx, [esi] ; box width (dl)
1357	; box height (dh)
1358 00000975 28F0	sub al, dh
1359 00000977 D0E8	shr al, 1
1360 00000979 A2[F228]	mov [ibcp+1], al ; row
1361 0000097C F6E4	mul ah
1362 0000097E D1E0	shl ax, 1 ; char + attribute
1363	;mov bx, ax
1364 00000980 01C3	add bx, ax ; *
1365 00000982 B050	mov al, 80
1366 00000984 28D0	sub al, dl
1367 00000986 D0E8	shr al, 1
1368 00000988 A2[F128]	mov [ibcp], al ; column
1369 0000098B D0E0	shl al, 1 ; char + attribute

1370	0000098D	28E4	sub	ah, ah
1371	0000098F	01C3	add	bx, ax
1372	00000991	678A6605	mov	ah, [esi+5] ; color attributes
1373	00000995	B020	mov	al, 20h ; space/blank
1374	00000997	88F1	mov	cl, dh ; height
1375				
1376	00000999	51	push	cx
1377	0000099A	88D1	mov	cl, dl
1378	0000099C	6689DF	mov	edi, ebx
1379	0000099F	F3AB	rep	stosw
1380	000009A1	59	pop	cx
1381	000009A2	81C3A000	add	bx, 80*2 ; number of columns * 2
1382	000009A6	E2F1	loop	ib0
1383				
1384				
1385	000009A8	6631FF	mov	edi, 0B8000h
1386	000009AB	A0[F228]	xor	edi, edi ; *
1387	000009AE	67024602	mov	al, [ibcp+1] ; row position
1388	000009B2	A2[F228]	add	al, [esi+2] ; label offset (row)
1389	000009B5	B4A0	mov	[ibcp+1], al
1390	000009B7	F6E4	mov	ah, 80*2
1391	000009B9	01C7	mul	ah
1392	000009BB	A0[F128]	add	di, ax
1393	000009BE	67024603	mov	al, [ibcp] ; column position
1394	000009C2	A2[F128]	add	al, [esi+3] ; label offset (column)
1395	000009C5	30E4	mov	[ibcp], al
1396	000009C7	D0E0	xor	ah, ah
1397	000009C9	01C7	shl	al, 1
1398	000009CB	6689F3	add	di, ax
1399	000009CE	6683C606	mov	ebx, esi
1400			add	esi, 6 ; Label offset
1401	000009D2	AC	ib2:	
1402	000009D3	08C0		lodsb
1403	000009D5	7406		or al, al
1404	000009D7	AA		jz short ib3
1405	000009D8	47		stosb
1406	000009D9	FEC1		inc di
1407	000009DB	EBF5		inc cl
1408				jmp short ib2
1409	000009DD	000E[F128]	ib3:	
1410	000009E1	66893E[0228]		add [ibcp], cl ; column position
1411				mov [current_txtpos], edi
1412	000009E6	678A4B04		;
1413	000009EA	08C9		mov cl, [ebx+4] ; input char count
1414	000009EC	740A		or cl, cl
1415	000009EE	B020		jz short ib5 ; message box (no input)
1416	000009F0	B407		mov al, 20h
1417				mov ah, 07h ; black background
1418				; light gray fore color
			ib4:	

1419	000009F2 F3AB	rep stosw
1420		;
1421	000009F4 8B16[F128]	mov dx, [ibcp] ; cursor position
1422		ib5:
1423	000009F8 C3	retn
1424		
1425		hide_cursor:
1426		; 01/12/2014
1427		; 27/11/2014
1428		;CH = cursor start line (bits 0-4)
1429		; and options (bits 5-7).
1430		;CL = bottom cursor line (bits 0-4).
1431		; when bit 5 of CH is set to 0, the cursor is visible.
1432		; when bit 5 is 1, the cursor is not visible.
1433		; hide blinking text cursor:
1434	000009F9 B520	mov ch, 32
1435	000009FB EB0D	jmp short hc_sc
1436		
1437		show_cursor:
1438		; 01/12/2014
1439		; 27/11/2014
1440		; dh = row
1441		; dl = column
1442	000009FD 8916[3229]	mov [cursor_posn], dx
1443	00000A01 28DB	sub bl, bl ; video page 0
1444	00000A03 E8251F	call set_cposx
1445		;
1446		;show box-shaped blinking text cursor
1447	00000A06 B500	mov ch, 0
1448		; 13/12/2014
1449	00000A08 B506	mov ch, 6
1450		hc_sc:
1451	00000A0A B107	mov cl, 7
1452	00000A0C B401	mov ah, 1
1453	00000A0E CD10	int 10h
1454	00000A10 C3	retn
1455		
1456		read_disk_sector:
1457		; 18/01/2015
1458		; 15/01/2015
1459		; 02/12/2014
1460		; 01/12/2014
1461	00000A11 6631DB	xor ebx, ebx
1462	00000A14 8A1E[AE27]	mov bl, [ds_drv]
1463	00000A18 88DA	mov dl, bl
1464	00000A1A 80FA02	cmp dl, 2
1465	00000A1D 7203	jb short rd0
1466	00000A1F 80C27E	add dl, 7Eh ; 80h, 81h, 82h, 83h
1467		rd0:

1468	00000A22	8816[4529]	mov	[drv], dl
1469	00000A26	6681C3[3E290000]	add	ebx, drv_status
1470	00000A2D	678A33	mov	dh, [ebx]
1471			rd1:	
1472	00000A30	80FEF0	cmp	dh, 0F0h ; 18/1/2015
1473	00000A33	F5	cmc	
1474	00000A34	7274	jc	short rd_lba_fails
1475			;	
1476	00000A36	6631DB	xor	ebx, ebx
1477	00000A39	8A1E[AE27]	mov	bl, [ds_drv]
1478	00000A3D	6689DE	mov	esi, ebx
1479	00000A40	C0E302	shl	bl, 2
1480	00000A43	6681C3[B0270000]	add	ebx, ds_sec
1481	00000A4A	66678B03	mov	eax, [ebx]
1482	00000A4E	6681EB[B0270000]	sub	ebx, ds_sec
1483	00000A55	6681C3[042A0000]	add	ebx, disk_size
1484	00000A5C	66673B03	cmp	eax, [ebx] ; Last sector + 1 (number of secs.)
1485	00000A60	F5	cmc	
1486	00000A61	7247	jc	short rd_lba_fails
1487			;	02/02/2015
1488	00000A63	F6C601	test	dh, 1 ; LBA ready ?
1489	00000A66	7443	jz	short rd_chs
1490			rd_lba:	
1491			;	LBA read (with temporary function)
1492			;	((Retro UNIX 386 v1 - DISK I/O Test))
1493	00000A68	6681C6[3E290000]	add	esi, drv_status
1494	00000A6F	6780268F	and	byte [esi], 8Fh ; clear error bits
1495			;	
1496	00000A73	6689C3	mov	ebx, eax
1497	00000A76	89D9	mov	cx, bx
1498	00000A78	66C1EB10	shr	ebx, 16
1499	00000A7C	8CCF	mov	di, cs
1500	00000A7E	8EC7	mov	es, di
1501	00000A80	BF[2C2A]	mov	di, buffer
1502	00000A83	8A16[4529]	mov	dl, [drv]
1503	00000A87	C606[2A0B]04	mov	byte [retry_count], 4
1504			rd_lba_retry:	
1505	00000A8C	B41B	mov	ah, 1Bh ; Temporary/Private function
1506	00000A8E	B001	mov	al, 1
1507	00000A90	CD13	int	13h
1508	00000A92	7316	jnc	short rd_lba_ok
1509	00000A94	80FC80	cmp	ah, 80h ; time out ?
1510	00000A97	7410	je	short rd_lba_rfails
1511	00000A99	FE0E[2A0B]	dec	byte [retry_count]
1512	00000A9D	740A	jz	short rd_lba_rfails
1513	00000A9F	B40D	mov	ah, 0Dh ; Alternate reset
1514	00000AA1	CD13	int	13h
1515	00000AA3	73E7	jnc	short rd_lba_retry
1516	00000AA5	67800EF0	or	byte [esi], 0F0h ; drive not ready !

```

1517
1518 00000AA9 F9
1519
1520
1521 00000AAA C3
1522
1523
1524
1525 00000AAB 66D1E6
1526 00000AAE 6689F3
1527 00000AB1 6631D2
1528 00000AB4 6629C9
1529 00000AB7 6681C3[F8290000]
1530 00000ABE 678B0B
1531
1532 00000AC1 66F7F1
1533 00000AC4 88D1
1534 00000AC6 FEC1
1535 00000AC8 51
1536 00000AC9 6689F3
1537 00000ACC 6681C3[EC290000]
1538 00000AD3 678B0B
1539 00000AD6 6631D2
1540
1541 00000AD9 66F7F1
1542 00000ADC 59
1543 00000ADD 88D6
1544 00000ADF 8A16[4529]
1545 00000AE3 88C5
1546 00000AE5 C0E406
1547 00000AE8 08E1
1548
1549 00000AEA 0E
1550 00000AEB 07
1551 00000AEC BB[2C2A]
1552
1553
1554
1555
1556
1557
1558 00000AEF 66D1EE
1559 00000AF2 6681C6[3E290000]
1560 00000AF9 6780268F
1561
1562 00000AFD C606[2A0B]04
1563
1564 00000B02 B402
1565 00000B04 B001

```

```

rd_lba_rfails:
    stc
rd_lba_fails:
rd_lba_ok:
    retn
    ;
    ; CHS read (convert LBA address to CHS values)
rd_chs:
    shl     esi, 1
    mov     ebx, esi
    xor     edx, edx ; 0
    sub     ecx, ecx
    add     ebx, spt
    mov     cx, [ebx] ; sector per track
    ; EDX:EAX = LBA
    div     ecx
    mov     cl, dl ; sector number - 1
    inc     cl ; sector number (1 based)
    push    cx
    mov     ebx, esi
    add     ebx, heads
    mov     cx, [ebx] ; heads
    xor     edx, edx
    ; EAX = cylinders * heads + head
    div     ecx
    pop     cx ; sector number
    mov     dh, dl ; head number
    mov     dl, [drv]
    mov     ch, al ; cylinder (bits 0-7)
    shl     ah, 6
    or      cl, ah ; cylinder (bits 8-9)
    ; sector (bits 0-7)
    push    cs
    pop     es
    mov     bx, buffer
    ; CL = sector (bits 0-6)
    ;     cylinder (bits 7-8 -> bits 8-9)
    ; CH = cylinder (bits 0-7)
    ; DH = head
    ; DL = drive
    ; 02/02/2015
    shr     esi, 1 ; drive index (byte alignment)
    add     esi, drv_status
    and     byte [esi], 8Fh ; clear error bits
    ;
    mov     byte [retry_count], 4
rd_retry:
    mov     ah, 02h ; read sectors
    mov     al, 1 ; sector count

```

```

1566 00000B06 CD13          int    13h
1567 00000B08 731F          jnc    short rd_ok
1568 00000B0A 80FC80        cmp    ah, 80h ; time out ?
1569 00000B0D 7406          je     short rd_rfails
1570 00000B0F FE0E[2A0B]      dec    byte [retry_count]
1571 00000B13 7502          jnz    short rd_reset
1572                          rd_rfails:
1573 00000B15 F9              stc
1574                          rd_fails:
1575 00000B16 C3              retn
1576                          rd_reset:
1577                          ; 02/02/2015
1578 00000B17 28E4          sub    ah, ah
1579 00000B19 80FA80        cmp    dl, 80h
1580 00000B1C 7202          jb     rd_fd_reset
1581 00000B1E B40D          mov    ah, 0Dh ; Alternate reset
1582                          rd_fd_reset:
1583 00000B20 CD13          int    13h
1584 00000B22 73DE          jnc    short rd_retry
1585 00000B24 67800EF0      or     byte [esi], 0F0h ; drive not ready !
1586 00000B28 F9              stc
1587                          rd_ok:
1588 00000B29 C3              retn
1589
1590                          retry_count: ; 16/12/2014
1591 00000B2A 00              db     0
1592                          ;rd_lba:
1593                          ; mov    [lba], eax
1594                          ; mov    ax, cs
1595                          ; mov    [tbuff+2], ax
1596                          ; mov    esi, disk_pkt
1597                          ; ;mov    dl, [drv]
1598                          ; mov    ah, 42h ; Extended read
1599                          ; int    13h
1600                          ; retn
1601
1602                          %include 'diskette.inc' ; 15/12/2014
1603 <1> ; DISKETTE I/O - Erdogan Tan (Retro UNIX 386 v1 project)
1604 <1> ;
1605 <1> ; 20/01/2015
1606 <1> ; 02/01/2015
1607 <1> ; 23/12/2014
1608 <1> ; 22/12/2014
1609 <1> ; 19/12/2014
1610 <1> ; 18/12/2014
1611 <1> ; 17/12/2014
1612 <1> ; 16/12/2014
1613 <1> ; Code (DELAY) modifications - AWARD BIOS 1999 (ADISK.EQU, COMMON.MAC)
1614 <1> ;

```

```

1615 <1> ; ADISK.EQU
1616 <1>
1617 <1> ;----- Wait control constants
1618 <1>
1619 <1> ;amount of time to wait while RESET is active.
1620 <1>
1621 <1> WAITCPU_RESET_ON EQU 21 ;Reset on must last at least 14us
1622 <1> ;at 250 KBS xfer rate.
1623 <1> ;see INTEL MCS, 1985, pg. 5-456
1624 <1>
1625 <1> WAITCPU_FOR_STATUS EQU 100 ;allow 30 microseconds for
1626 <1> ;status register to become valid
1627 <1> ;before re-reading.
1628 <1>
1629 <1> ;After sending a byte to NEC, status register may remain
1630 <1> ;incorrectly set for 24 us.
1631 <1>
1632 <1> WAITCPU_RQM_LOW EQU 24 ;number of loops to check for
1633 <1> ;RQM low.
1634 <1>
1635 <1> ; COMMON.MAC
1636 <1> ;
1637 <1> ; Timing macros
1638 <1> ;
1639 <1>
1640 <1> %macro SIODELAY 0 ; SHORT IODELAY
1641 <1> jmp short $+2
1642 <1> %endmacro
1643 <1>
1644 <1> %macro IODELAY 0 ; NORMAL IODELAY
1645 <1> jmp short $+2
1646 <1> jmp short $+2
1647 <1> %endmacro
1648 <1>
1649 <1> %macro NEWIODELAY 0
1650 <1> out 0ebh,al
1651 <1> %endmacro
1652 <1>
1653 <1> ; (According to) AWARD BIOS 1999 - ATORGS.ASM (dw -> equ, db -> equ)
1654 <1> ;;; WAIT_FOR_MEM
1655 <1> WAIT_FDU_INT_LO equ 017798 ; 2.5 secs in 30 micro units.
1656 <1> WAIT_FDU_INT_HI equ 1
1657 <1> ;;; WAIT_FOR_PORT
1658 <1> WAIT_FDU_SEND_LO equ 16667 ; .5 secons in 30 us units.
1659 <1> WAIT_FDU_SEND_HI equ 0
1660 <1> ;Time to wait while waiting for each byte of NEC results = .5
1661 <1> ;seconds. .5 seconds = 500,000 micros. 500,000/30 = 16,667.
1662 <1> WAIT_FDU_RESULTS_LO equ 16667 ; .5 seconds in 30 micro units.
1663 <1> WAIT_FDU_RESULTS_HI equ 0

```

```

1664 <1> ;;; WAIT_REFRESH
1665 <1> ;amount of time to wait for head settle, per unit in parameter
1666 <1> ;table = 1 ms.
1667 <1> WAIT_FDU_HEAD_SETTLE      equ    33          ; 1 ms in 30 micro units.
1668 <1>
1669 <1>
1670 <1> ; ////////////////////////////////// DISKETTE I/O //////////////////////////////////
1671 <1>
1672 <1> ; 11/12/2014 (copy from IBM PC-XT Model 286 BIOS - POSTEQU.INC)
1673 <1>
1674 <1> ;-----
1675 <1> ;      EQUATES USED BY POST AND BIOS      :
1676 <1> ;-----
1677 <1>
1678 <1> ;----- 8042 KEYBOARD INTERFACE AND DIAGNOSTIC CONTROL REGISTERS -----
1679 <1> PORT_A          EQU    060H          ; 8042 KEYBOARD SCAN CODE/CONTROL PORT
1680 <1> PORT_B          EQU    061H          ; PORT B READ/WRITE DIAGNOSTIC REGISTER
1681 <1> REFRESH_BIT EQU    00010000B        ; REFRESH TEST BIT
1682 <1>
1683 <1> ;-----
1684 <1> ;      CMOS EQUATES FOR THIS SYSTEM      :
1685 <1> ;-----
1686 <1> CMOS_PORT      EQU    070H          ; I/O ADDRESS OF CMOS ADDRESS PORT
1687 <1> CMOS_DATA      EQU    071H          ; I/O ADDRESS OF CMOS DATA PORT
1688 <1> NMI            EQU    10000000B      ; DISABLE NMI INTERRUPTS MASK -
1689 <1>                                     ; HIGH BIT OF CMOS LOCATION ADDRESS
1690 <1>
1691 <1> ;----- CMOS TABLE LOCATION ADDRESS'S ## -----
1692 <1> CMOS_DISKETTE  EQU    010H          ; DISKETTE DRIVE TYPE BYTE      ;
1693 <1> ;              EQU    011H          ; - RESERVED                    ;C
1694 <1> CMOS_DISK      EQU    012H          ; FIXED DISK TYPE BYTE          ;H
1695 <1> ;              EQU    013H          ; - RESERVED                    ;E
1696 <1> CMOS_EQUIP     EQU    014H          ; EQUIPMENT WORD LOW BYTE       ;C
1697 <1>
1698 <1> ;----- DISKETTE EQUATES -----
1699 <1> INT_FLAG       EQU    10000000B      ; INTERRUPT OCCURRENCE FLAG
1700 <1> DSK_CHG        EQU    10000000B      ; DISKETTE CHANGE FLAG MASK BIT
1701 <1> DETERMINED     EQU    00010000B      ; SET STATE DETERMINED IN STATE BITS
1702 <1> HOME           EQU    00010000B      ; TRACK 0 MASK
1703 <1> SENSE_DRV_ST   EQU    00000100B      ; SENSE DRIVE STATUS COMMAND
1704 <1> TRK_SLAP       EQU    030H           ; CRASH STOP (48 TPI DRIVES)
1705 <1> QUIET_SEEK     EQU    00AH          ; SEEK TO TRACK 10
1706 <1> ;MAX_DRV       EQU    2             ; MAX NUMBER OF DRIVES
1707 <1> HD12_SETTLE    EQU    15             ; 1.2 M HEAD SETTLE TIME
1708 <1> HD320_SETTLE   EQU    20             ; 320 K HEAD SETTLE TIME
1709 <1> MOTOR_WAIT     EQU    37             ; 2 SECONDS OF COUNTS FOR MOTOR TURN OFF
1710 <1>
1711 <1> ;----- DISKETTE ERRORS -----
1712 <1> ;TIME_OUT      EQU    080H          ; ATTACHMENT FAILED TO RESPOND

```

```

1713 <1> ;BAD_SEEK EQU 040H ; SEEK OPERATION FAILED
1714 <1> BAD_NEC EQU 020H ; DISKETTE CONTROLLER HAS FAILED
1715 <1> BAD_CRC EQU 010H ; BAD CRC ON DISKETTE READ
1716 <1> MED_NOT_FND EQU 00CH ; MEDIA TYPE NOT FOUND
1717 <1> DMA_BOUNDARY EQU 009H ; ATTEMPT TO DMA ACROSS 64K BOUNDARY
1718 <1> BAD_DMA EQU 008H ; DMA OVERRUN ON OPERATION
1719 <1> MEDIA_CHANGE EQU 006H ; MEDIA REMOVED ON DUAL ATTACH CARD
1720 <1> RECORD_NOT_FND EQU 004H ; REQUESTED SECTOR NOT FOUND
1721 <1> WRITE_PROTECT EQU 003H ; WRITE ATTEMPTED ON WRITE PROTECT DISK
1722 <1> BAD_ADDR_MARK EQU 002H ; ADDRESS MARK NOT FOUND
1723 <1> BAD_CMD EQU 001H ; BAD COMMAND PASSED TO DISKETTE I/O
1724 <1>
1725 <1> ;----- DISK CHANGE LINE EQUATES -----
1726 <1> NOCHGLN EQU 001H ; NO DISK CHANGE LINE AVAILABLE
1727 <1> CHGLN EQU 002H ; DISK CHANGE LINE AVAILABLE
1728 <1>
1729 <1> ;----- MEDIA/DRIVE STATE INDICATORS -----
1730 <1> TRK_CAPA EQU 0000001B ; 80 TRACK CAPABILITY
1731 <1> FMT_CAPA EQU 00000010B ; MULTIPLE FORMAT CAPABILITY (1.2M)
1732 <1> DRV_DET EQU 00000100B ; DRIVE DETERMINED
1733 <1> MED_DET EQU 00010000B ; MEDIA DETERMINED BIT
1734 <1> DBL_STEP EQU 00100000B ; DOUBLE STEP BIT
1735 <1> RATE_MSK EQU 11000000B ; MASK FOR CLEARING ALL BUT RATE
1736 <1> RATE_500 EQU 00000000B ; 500 KBS DATA RATE
1737 <1> RATE_300 EQU 01000000B ; 300 KBS DATA RATE
1738 <1> RATE_250 EQU 10000000B ; 250 KBS DATA RATE
1739 <1> STRT_MSK EQU 00001100B ; OPERATION START RATE MASK
1740 <1> SEND_MSK EQU 11000000B ; MASK FOR SEND RATE BITS
1741 <1>
1742 <1> ;----- MEDIA/DRIVE STATE INDICATORS COMPATIBILITY -----
1743 <1> M3D3U EQU 00000000B ; 360 MEDIA/DRIVE NOT ESTABLISHED
1744 <1> M3D1U EQU 00000001B ; 360 MEDIA,1.2DRIVE NOT ESTABLISHED
1745 <1> M1D1U EQU 00000010B ; 1.2 MEDIA/DRIVE NOT ESTABLISHED
1746 <1> MED_UNK EQU 00000111B ; NONE OF THE ABOVE
1747 <1>
1748 <1> ;----- INTERRUPT EQUATES -----
1749 <1> EOI EQU 020H ; END OF INTERRUPT COMMAND TO 8259
1750 <1> INTA00 EQU 020H ; 8259 PORT
1751 <1> INTA01 EQU 021H ; 8259 PORT
1752 <1> INTB00 EQU 0A0H ; 2ND 8259
1753 <1> INTB01 EQU 0A1H ;
1754 <1>
1755 <1> ;-----
1756 <1> DMA08 EQU 008H ; DMA STATUS REGISTER PORT ADDRESS
1757 <1> DMA EQU 000H ; DMA CH.0 ADDRESS REGISTER PORT ADDRESS
1758 <1> DMA18 EQU 0D0H ; 2ND DMA STATUS PORT ADDRESS
1759 <1> DMA1 EQU 0C0H ; 2ND DMA CH.0 ADDRESS REGISTER ADDRESS
1760 <1> ;-----
1761 <1> TIMER EQU 040H ; 8254 TIMER - BASE ADDRESS

```

```

1762 <1>
1763 <1> ;-----
1764 <1> DMA_PAGE EQU 081H ; START OF DMA PAGE REGISTERS
1765 <1>
1766 <1>
1767 <1> ; 11/12/2014 (copy from IBM PC-XT Model 286 BIOS - DSEG.INC)
1768 <1>
1769 <1> ;-----
1770 <1> ; 80286 INTERRUPT LOCATIONS :
1771 <1> ; REFERENCED BY POST & BIOS :
1772 <1> ;-----
1773 <1>
1774 <1> ;ABS0 SEGMENT AT 0 ; ADDRESS= 0000:0000
1775 <1>
1776 <1> ; ORG 4*008H
1777 <1> ;@INT_PTR DD ? ; HARDWARE INTERRUPT POINTER (8-F)
1778 <1>
1779 00000B2B 00000000 <1> ORG_INT08H: dd 0 ; IRQ 0 - Timer
1780 00000B2F 00000000 <1> ORG_INT0EH: dd 0 ; IRQ 6 - Diskette
1781 <1>
1782 <1> ; ORG 4*013H
1783 <1> ;@ORG_VECTOR DD ? ; DISKETTE/DISK INTERRUPT VECTOR
1784 <1>
1785 00000B33 00000000 <1> ORG_INT13H: dd 0 ; Original INT 13H
1786 <1>
1787 <1> ; ORG 4*01EH
1788 <1> ;@DISK_POINTER DD ? ; POINTER TO DISKETTE PARAMETER TABLE
1789 <1>
1790 00000B37 [AE0B] <1> DISK_POINTER: dw MD_TBL6 ; Pointer to Diskette Parameter Table
1791 00000B39 0000 <1> dw 0
1792 <1>
1793 <1> ; ORG 4*040H
1794 <1> ;@DISK_VECTOR DD ? ; POINTER TO DISKETTE INTERRUPT CODE
1795 <1>
1796 00000B3B 00000000 <1> ORG_INT40H: dd 0
1797 <1>
1798 <1> ; ORG 4*041H
1799 <1> ;@HF_TBL_VEC DD ? ; POINTER TO FIRST DISK PARAMETER TABLE
1800 <1>
1801 <1> ;HF_TBL_VEC: dd 0
1802 <1>
1803 <1> ; ORG 4*046H
1804 <1> ;@HF1_TBL_VEC DD ? ; POINTER TO SECOND DISK PARAMETER TABLE
1805 <1>
1806 <1> ;HF1_TBL_VEC: dd 0
1807 <1>
1808 <1> ; ORG 4*076H
1809 <1> ;@HDISK_INT DD ? ; POINTER TO FIXED DISK INTERRUPT CODE
1810 <1>

```

```

1811 00000B3F 00000000      <1> ORG_INT76H: dd      0          ; IRQ 14 - Fixed Disk (1st controller)
1812 00000B43 00000000      <1> ORG_INT77H: dd      0          ; IRQ 15 - Fixed Disk (2nd controller)
1813                          <1>
1814                          <1> ; ABS0                ENDS
1815                          <1>
1816                          <1> ; IBM PC-XT Model 286 source code ORGS.ASM (06/10/85) - 14/12/2014
1817                          <1> ;-----
1818                          <1> ; DISK_BASE                                :
1819                          <1> ;     THIS IS THE SET OF PARAMETERS REQUIRED FOR      :
1820                          <1> ;     DISKETTE OPERATION. THEY ARE POINTED AT BY THE  :
1821                          <1> ;     DATA VARIABLE @DISK_POINTER. TO MODIFY THE PARAMETERS, :
1822                          <1> ;     BUILD ANOTHER PARAMETER BLOCK AND POINT AT IT    :
1823                          <1> ;-----
1824                          <1>
1825                          <1> ;DISK_BASE:
1826                          <1> ;     DB      11011111B      ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
1827                          <1> ;     DB      2              ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
1828                          <1> ;     DB      MOTOR_WAIT    ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
1829                          <1> ;     DB      2              ; 512 BYTES/SECTOR
1830                          <1> ;     ;DB      15              ; EOT (LAST SECTOR ON TRACK)
1831                          <1> ;     db      18              ; (EOT for 1.44MB diskette)
1832                          <1> ;     DB      01BH            ; GAP LENGTH
1833                          <1> ;     DB      0FFH            ; DTL
1834                          <1> ;     ;DB      054H            ; GAP LENGTH FOR FORMAT
1835                          <1> ;     db      06ch            ; (for 1.44MB diskette)
1836                          <1> ;     DB      0F6H            ; FILL BYTE FOR FORMAT
1837                          <1> ;     DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
1838                          <1> ;     DB      8              ; MOTOR START TIME (1/8 SECONDS)
1839                          <1>
1840                          <1> ;-----
1841                          <1> ;     ROM BIOS DATA AREAS      :
1842                          <1> ;-----
1843                          <1>
1844                          <1> ;DATA          SEGMENT AT 40H          ; ADDRESS= 0040:0000
1845                          <1>
1846                          <1> ;@EQUIP_FLAG DW      ?          ; INSTALLED HARDWARE FLAGS
1847                          <1>
1848                          <1> ;-----
1849                          <1> ;     DISKETTE DATA AREAS      :
1850                          <1> ;-----
1851                          <1>
1852                          <1> ;@SEEK_STATUS      DB      ?          ; DRIVE RECALIBRATION STATUS
1853                          <1> ;                                ; BIT 3-0 = DRIVE 3-0 RECALIBRATION
1854                          <1> ;                                ; BEFORE NEXT SEEK IF BIT IS = 0
1855                          <1> ;@MOTOR_STATUS      DB      ?          ; MOTOR STATUS
1856                          <1> ;                                ; BIT 3-0 = DRIVE 3-0 CURRENTLY RUNNING
1857                          <1> ;                                ; BIT 7 = CURRENT OPERATION IS A WRITE
1858                          <1> ;@MOTOR_COUNT      DB      ?          ; TIME OUT COUNTER FOR MOTOR(S) TURN OFF
1859                          <1> ;@DSKETTE_STATUS DB      ?          ; RETURN CODE STATUS BYTE

```

```

1860 <1> ; ; CMD_BLOCK IN STACK FOR DISK OPERATION
1861 <1> ;@NEC_STATUS DB 7 DUP(?) ; STATUS BYTES FROM DISKETTE OPERATION
1862 <1>
1863 00000B47 00 <1> SEEK_STATUS: db 0
1864 00000B48 00 <1> MOTOR_STATUS: db 0
1865 00000B49 00 <1> MOTOR_COUNT: db 0
1866 00000B4A 00 <1> DSKETTE_STATUS: db 0
1867 00000B4B 0000000000000000 <1> NEC_STATUS: db 0,0,0,0,0,0,0
1868 <1>
1869 <1> ;-----
1870 <1> ; POST AND BIOS WORK DATA AREA :
1871 <1> ;-----
1872 <1>
1873 <1> ;@INTR_FLAG DB ? ; FLAG INDICATING AN INTERRUPT HAPPENED
1874 <1>
1875 <1> ;-----
1876 <1> ; TIMER DATA AREA :
1877 <1> ;-----
1878 <1>
1879 <1> ;@TIMER_LOW DW ? ; LOW WORD OF TIMER COUNT
1880 <1> ;@TIMER_HIGH DW ? ; HIGH WORD OF TIMER COUNT
1881 <1> ;@TIMER_OFL DB ? ; TIMER HAS ROLLED OVER SINCE LAST READ
1882 <1>
1883 <1> ; 17/12/2014 (IRQ 0 - INT 08H)
1884 <1> TIMER_LOW equ 46Ch ; Timer ticks (counter) @ 40h:006Ch
1885 <1> TIMER_HIGH equ 46Eh ; (18.2 timer ticks per second)
1886 <1> TIMER_OFL equ 470h ; Timer - 24 hours flag @ 40h:0070h
1887 <1>
1888 <1> ; MOTOR_COUNT and MOTOR_STATUS in BIOS Data Area
1889 <1> ORG_MOTOR_STATUS equ 43Fh ; Floppy disk motor status @ 40h:003Fh
1890 <1> ORG_MOTOR_COUNT equ 440h ; Floppy disk motor timeout @ 40h:0040h
1891 <1>
1892 <1> ;-----
1893 <1> ; SYSTEM DATA AREA :
1894 <1> ;-----
1895 <1>
1896 <1> ;@BIOS_BREAK DB ? ; BIT 7=1 IF BREAK KEY HAS BEEN PRESSED
1897 <1> ;@RESET_FLAG DW ? ; WORD=1234H IF KEYBOARD RESET UNDERWAY
1898 <1>
1899 <1> ;-----
1900 <1> ; FIXED DISK DATA AREAS :
1901 <1> ;-----
1902 <1>
1903 <1> ;@DISK_STATUS1 DB ? ; FIXED DISK STATUS
1904 <1> ;@HF_NUM DB ? ; COUNT OF FIXED DISK DRIVES
1905 <1> ;@CONTROL_BYTE DB ? ; HEAD CONTROL BYTE
1906 <1> ;@PORT_OFF DB ? ; RESERVED (PORT OFFSET)
1907 <1>
1908 <1> ;DISK_STATUS1: db 0

```

```

1909 <1> ;HF_NUM: db 0
1910 <1> ;CONTROL_BYTE: db 0
1911 <1> ;PORT_OFF: db 0
1912 <1>
1913 <1> ;-----
1914 <1> ; ADDITIONAL MEDIA DATA :
1915 <1> ;-----
1916 <1>
1917 <1> ;@LAstrate DB ? ; LAST DISKETTE DATA RATE SELECTED
1918 <1> ;@HF_STATUS DB ? ; STATUS REGISTER
1919 <1> ;@HF_ERROR DB ? ; ERROR REGISTER
1920 <1> ;@HF_INT_FLAG DB ? ; FIXED DISK INTERRUPT FLAG
1921 <1> ;@HF_CNTRL DB ? ; COMBO FIXED DISK/DISKETTE CARD BIT 0=1
1922 <1> ;@DSK_STATE DB ? ; DRIVE 0 MEDIA STATE
1923 <1> ; DB ? ; DRIVE 1 MEDIA STATE
1924 <1> ; DB ? ; DRIVE 0 OPERATION START STATE
1925 <1> ; DB ? ; DRIVE 1 OPERATION START STATE
1926 <1> ;@DSK_TRK DB ? ; DRIVE 0 PRESENT CYLINDER
1927 <1> ; DB ? ; DRIVE 1 PRESENT CYLINDER
1928 <1>
1929 00000B52 00 <1> LAstrate: db 0
1930 <1> ;HF_STATUS: db 0
1931 <1> ;HF_ERROR: db 0
1932 <1> ;HF_INT_FLAG: db 0
1933 <1> ;HF_CNTRL: db 0
1934 00000B53 00000000 <1> DSK_STATE: db 0,0,0,0
1935 00000B57 0000 <1> DSK_TRK: db 0,0
1936 <1>
1937 <1> ;-----
1938 <1> ; REAL TIME CLOCK DATA AREA :
1939 <1> ;-----
1940 <1>
1941 <1> ;@USER_FLAG DW ? ; OFFSET ADDRESS OF USERS WAIT FLAG
1942 <1> ;@USER_FLAG_SEG DW ? ; SEGMENT ADDRESS OF USER WAIT FLAG
1943 <1> ;@RTC_LOW DW ? ; LOW WORD OF USER WAIT FLAG
1944 <1> ;@RTC_HIGH DW ? ; HIGH WORD OF USER WAIT FLAG
1945 <1> ;@RTC_WAIT_FLAG DB ? ; WAIT ACTIVE FLAG (01=BUSY, 80=POSTED)
1946 <1> ; (00=POST ACKNOWLEDGED)
1947 <1>
1948 <1> ;DATA ENDS ; END OF BIOS DATA SEGMENT
1949 <1>
1950 <1>
1951 <1> ; 17/12/2014 (mov ax, [cfd])
1952 <1> ; 11/12/2014
1953 00000B59 00 <1> cfd: db 0 ; current floppy drive (for GET_PARM)
1954 <1> ; 17/12/2014 ; instead of 'DISK_PONTER'
1955 00000B5A 01 <1> pfd: db 1 ; previous floppy drive (for GET_PARM)
1956 <1> ; (initial value of 'pfd'
1957 <1> ; must be different then 'cfd' value

```

```

1958 <1> ; to force updating/initializing
1959 <1> ; current drive parameters)
1960 <1> ; 10/12/2014
1961 <1> ;
1962 <1> ;int40h:
1963 <1> ; pushf
1964 <1> ; push cs
1965 <1> ; cli
1966 <1> ; call DISKETTE_IO_1
1967 <1> ; retn
1968 <1>
1969 <1> ; DSKETTE ----- 04/21/86 DISKETTE BIOS
1970 <1> ; (IBM PC XT Model 286 System BIOS Source Code, 04-21-86)
1971 <1> ;
1972 <1>
1973 <1> ;-- INT13H -----
1974 <1> ; DISKETTE I/O
1975 <1> ; THIS INTERFACE PROVIDES ACCESS TO THE 5 1/4 INCH 360 KB,
1976 <1> ; 1.2 MB, 720 KB AND 1.44 MB DISKETTE DRIVES.
1977 <1> ; INPUT
1978 <1> ; (AH) = 00H RESET DISKETTE SYSTEM
1979 <1> ; HARD RESET TO NEC, PREPARE COMMAND, RECALIBRATE REQUIRED
1980 <1> ; ON ALL DRIVES
1981 <1> ;-----
1982 <1> ; (AH)= 01H READ THE STATUS OF THE SYSTEM INTO (AH)
1983 <1> ; @DISKETTE_STATUS FROM LAST OPERATION IS USED
1984 <1> ;-----
1985 <1> ; REGISTERS FOR READ/WRITE/VERIFY/FORMAT
1986 <1> ; (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
1987 <1> ; (DH) - HEAD NUMBER (0-1 ALLOWED, NOT VALUE CHECKED)
1988 <1> ; (CH) - TRACK NUMBER (NOT VALUE CHECKED)
1989 <1> ; MEDIA DRIVE TRACK NUMBER
1990 <1> ; 320/360 320/360 0-39
1991 <1> ; 320/360 1.2M 0-39
1992 <1> ; 1.2M 1.2M 0-79
1993 <1> ; 720K 720K 0-79
1994 <1> ; 1.44M 1.44M 0-79
1995 <1> ; (CL) - SECTOR NUMBER (NOT VALUE CHECKED, NOT USED FOR FORMAT)
1996 <1> ; MEDIA DRIVE SECTOR NUMBER
1997 <1> ; 320/360 320/360 1-8/9
1998 <1> ; 320/360 1.2M 1-8/9
1999 <1> ; 1.2M 1.2M 1-15
2000 <1> ; 720K 720K 1-9
2001 <1> ; 1.44M 1.44M 1-18
2002 <1> ; (AL) NUMBER OF SECTORS (NOT VALUE CHECKED)
2003 <1> ; MEDIA DRIVE MAX NUMBER OF SECTORS
2004 <1> ; 320/360 320/360 8/9
2005 <1> ; 320/360 1.2M 8/9
2006 <1> ; 1.2M 1.2M 15

```

```

2007      <1> ;          720K  720K          9
2008      <1> ;          1.44M 1.44M        18
2009      <1> ;
2010      <1> ;          (ES:BX) - ADDRESS OF BUFFER (NOT REQUIRED FOR VERIFY)
2011      <1> ;
2012      <1> ; -----
2013      <1> ;          (AH)= 02H  READ THE DESIRED SECTORS INTO MEMORY
2014      <1> ; -----
2015      <1> ;          (AH)= 03H  WRITE THE DESIRED SECTORS FROM MEMORY
2016      <1> ; -----
2017      <1> ;          (AH)= 04H  VERIFY THE DESIRED SECTORS
2018      <1> ; -----
2019      <1> ;          (AH)= 05H  FORMAT THE DESIRED TRACK
2020      <1> ;          (ES,BX) MUST POINT TO THE COLLECTION OF DESIRED ADDRESS FIELDS
2021      <1> ;          FOR THE          TRACK. EACH FIELD IS COMPOSED OF 4 BYTES, (C,H,R,N),
2022      <1> ;          WHERE C = TRACK NUMBER, H=HEAD NUMBER, R = SECTOR NUMBER,
2023      <1> ;          N= NUMBER OF BYTES PER SECTOR (00=128,01=256,02=512,03=1024),
2024      <1> ;          THERE MUST BE ONE ENTRY FOR EVERY SECTOR ON THE TRACK.
2025      <1> ;          THIS INFORMATION IS USED TO FIND THE REQUESTED SECTOR DURING
2026      <1> ;          READ/WRITE ACCESS.
2027      <1> ;          PRIOR TO FORMATTING A DISKETTE, IF THERE EXISTS MORE THAN
2028      <1> ;          ONE SUPPORTED MEDIA FORMAT TYPE WITHIN THE DRIVE IN QUESTION,
2029      <1> ;          THEN "SET DASD TYPE" (INT 13H, AH = 17H) OR 'SET MEDIA TYPE'
2030      <1> ;          (INT 13H, AH = 18H) MUST BE CALLED TO SET THE DISKETTE TYPE
2031      <1> ;          THAT IS TO BE FORMATTED. IF "SET DASD TYPE" OR "SET MEDIA TYPE"
2032      <1> ;          IS NOT CALLED, THE FORMAT ROUTINE WILL ASSUME THE
2033      <1> ;          MEDIA FORMAT TO BE THE MAXIMUM CAPACITY OF THE DRIVE.
2034      <1> ;
2035      <1> ;          THESE PARAMETERS OF DISK BASE MUST BE CHANGED IN ORDER TO
2036      <1> ;          FORMAT THE FOLLOWING MEDIAS:
2037      <1> ; -----
2038      <1> ;          : MEDIA   :      DRIVE      : PARM 1 : PARM 2 :
2039      <1> ;          : -----
2040      <1> ;          : 320K   : 320K/360K/1.2M : 50H   : 8     :
2041      <1> ;          : 360K   : 320K/360K/1.2M : 50H   : 9     :
2042      <1> ;          : 1.2M   : 1.2M           : 54H   : 15    :
2043      <1> ;          : 720K   : 720K/1.44M     : 50H   : 9     :
2044      <1> ;          : 1.44M   : 1.44M          : 6CH   : 18    :
2045      <1> ;          : -----
2046      <1> ;          NOTES: - PARM 1 = GAP LENGTH FOR FORMAT
2047      <1> ;          - PARM 2 = EOT (LAST SECTOR ON TRACK)
2048      <1> ;          - DISK BASE IS POINTED BY DISK POINTER LOCATED
2049      <1> ;          AT ABSOLUTE ADDRESS 0:78.
2050      <1> ;          - WHEN FORMAT OPERATIONS ARE COMPLETE, THE PARAMETERS
2051      <1> ;          SHOULD BE RESTORED TO THEIR RESPECTIVE INITIAL VALUES.
2052      <1> ; -----
2053      <1> ;          (AH) = 08H READ DRIVE PARAMETERS
2054      <1> ;          REGISTERS

```

```

2055      <1> ;      INPUT
2056      <1> ;      (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
2057      <1> ;      OUTPUT
2058      <1> ;      (ES:DI) POINTS TO DRIVE PARAMETER TABLE
2059      <1> ;      (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM NUMBER OF TRACKS
2060      <1> ;      (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
2061      <1> ;      BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
2062      <1> ;      (DH) - MAXIMUM HEAD NUMBER
2063      <1> ;      (DL) - NUMBER OF DISKETTE DRIVES INSTALLED
2064      <1> ;      (BH) - 0
2065      <1> ;      (BL) - BITS 7 THRU 4 - 0
2066      <1> ;      BITS 3 THRU 0 - VALID DRIVE TYPE VALUE IN CMOS
2067      <1> ;      (AX) - 0
2068      <1> ;      UNDER THE FOLLOWING CIRCUMSTANCES:
2069      <1> ;      (1) THE DRIVE NUMBER IS INVALID,
2070      <1> ;      (2) THE DRIVE TYPE IS UNKNOWN AND CMOS IS NOT PRESENT,
2071      <1> ;      (3) THE DRIVE TYPE IS UNKNOWN AND CMOS IS BAD,
2072      <1> ;      (4) OR THE DRIVE TYPE IS UNKNOWN AND THE CMOS DRIVE TYPE IS INVALID
2073      <1> ;      THEN ES,AX,BX,CX,DH,DI=0 ; DL=NUMBER OF DRIVES.
2074      <1> ;      IF NO DRIVES ARE PRESENT THEN: ES,AX,BX,CX,DX,DI=0.
2075      <1> ;      @DISKETTE_STATUS = 0 AND CY IS RESET.
2076      <1> ; -----
2077      <1> ;      (AH)= 15H  READ DASD TYPE
2078      <1> ;      OUTPUT REGISTERS
2079      <1> ;      (AH) - ON RETURN IF CARRY FLAG NOT SET, OTHERWISE ERROR
2080      <1> ;      00 - DRIVE NOT PRESENT
2081      <1> ;      01 - DISKETTE, NO CHANGE LINE AVAILABLE
2082      <1> ;      02 - DISKETTE, CHANGE LINE AVAILABLE
2083      <1> ;      03 - RESERVED (FIXED DISK)
2084      <1> ;      (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
2085      <1> ; -----
2086      <1> ;      (AH)= 16H  DISK CHANGE LINE STATUS
2087      <1> ;      OUTPUT REGISTERS
2088      <1> ;      (AH) - 00 - DISK CHANGE LINE NOT ACTIVE
2089      <1> ;      06 - DISK CHANGE LINE ACTIVE & CARRY BIT ON
2090      <1> ;      (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
2091      <1> ; -----
2092      <1> ;      (AH)= 17H  SET DASD TYPE FOR FORMAT
2093      <1> ;      INPUT REGISTERS
2094      <1> ;      (AL) - 00 - NOT USED
2095      <1> ;      01 - DISKETTE 320/360K IN 360K DRIVE
2096      <1> ;      02 - DISKETTE 360K IN 1.2M DRIVE
2097      <1> ;      03 - DISKETTE 1.2M IN 1.2M DRIVE
2098      <1> ;      04 - DISKETTE 720K IN 720K DRIVE
2099      <1> ;      (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED:
2100      <1> ;      (DO NOT USE WHEN DISKETTE ATTACH CARD USED)
2101      <1> ; -----
2102      <1> ;      (AH)= 18H  SET MEDIA TYPE FOR FORMAT
2103      <1> ;      INPUT REGISTERS

```

```

2104      <1> ;      (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM TRACKS
2105      <1> ;      (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
2106      <1> ;      BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
2107      <1> ;      (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHACKED)
2108      <1> ;      OUTPUT REGISTERS:
2109      <1> ;      (ES:DI) - POINTER TO DRIVE PARAMETERS TABLE FOR THIS MEDIA TYPE,
2110      <1> ;      UNCHANGED IF (AH) IS NON-ZERO
2111      <1> ;      (AH) - 00H, CY = 0, TRACK AND SECTORS/TRACK COMBINATION IS SUPPORTED
2112      <1> ;      - 01H, CY = 1, FUNCTION IS NOT AVAILABLE
2113      <1> ;      - 0CH, CY = 1, TRACK AND SECTORS/TRACK COMBINATION IS NOT SUPPORTED
2114      <1> ;      - 80H, CY = 1, TIME OUT (DISKETTE NOT PRESENT)
2115      <1> ; -----
2116      <1> ;      DISK CHANGE STATUS IS ONLY CHECKED WHEN A MEDIA SPECIFIED IS OTHER
2117      <1> ;      THAN 360 KB DRIVE. IF THE DISK CHANGE LINE IS FOUND TO BE
2118      <1> ;      ACTIVE THE FOLLOWING ACTIONS TAKE PLACE:
2119      <1> ;      ATTEMPT TO RESET DISK CHANGE LINE TO INACTIVE STATE.
2120      <1> ;      IF ATTEMPT SUCCEEDS SET DASD TYPE FOR FORMAT AND RETURN DISK
2121      <1> ;      CHANGE ERROR CODE
2122      <1> ;      IF ATTEMPT FAILS RETURN TIMEOUT ERROR CODE AND SET DASD TYPE
2123      <1> ;      TO A PREDETERMINED STATE INDICATING MEDIA TYPE UNKNOWN.
2124      <1> ;      IF THE DISK CHANGE LINE IN INACTIVE PERFORM SET DASD TYPE FOR FORMAT.
2125      <1> ;
2126      <1> ; DATA VARIABLE -- @DISK_POINTER
2127      <1> ;      DOUBLE WORD POINTER TO THE CURRENT SET OF DISKETTE PARAMETERS
2128      <1> ; -----
2129      <1> ; OUTPUT FOR ALL FUNCTIONS
2130      <1> ;      AH = STATUS OF OPERATION
2131      <1> ;      STATUS BITS ARE DEFINED IN THE EQUATES FOR @DISKETTE_STATUS
2132      <1> ;      VARIABLE IN THE DATA SEGMENT OF THIS MODULE
2133      <1> ;      CY = 0 SUCCESSFUL OPERATION (AH=0 ON RETURN, EXCEPT FOR READ DASD
2134      <1> ;      TYPE AH=(15)).
2135      <1> ;      CY = 1 FAILED OPERATION (AH HAS ERROR REASON)
2136      <1> ;      FOR READ/WRITE/VERIFY
2137      <1> ;      DS,BX,DX,CX PRESERVED
2138      <1> ;      NOTE: IF AN ERROR IS REPORTED BY THE DISKETTE CODE, THE APPROPRIATE
2139      <1> ;      ACTION IS TO RESET THE DISKETTE, THEN RETRY THE OPERATION.
2140      <1> ;      ON READ ACCESSES, NO MOTOR START DELAY IS TAKEN, SO THAT
2141      <1> ;      THREE RETRIES ARE REQUIRED ON READS TO ENSURE THAT THE
2142      <1> ;      PROBLEM IS NOT DUE TO MOTOR START-UP.
2143      <1> ; -----
2144      <1> ;
2145      <1> ; DISKETTE STATE MACHINE - ABSOLUTE ADDRESS 40:90 (DRIVE A) & 91 (DRIVE B)
2146      <1> ;
2147      <1> ; -----
2148      <1> ;      |   7   |   6   |   5   |   4   |   3   |   2   |   1   |   0   |
2149      <1> ;      |-----|-----|-----|-----|-----|-----|-----|-----|
2150      <1> ;
2151      <1> ;
2152      <1> ;      |   |   |   |   |   |   |   |   |

```

```

2153      <1> ;
2154      <1> ;
2155      <1> ;
2156      <1> ;
2157      <1> ;
2158      <1> ;
2159      <1> ;
2160      <1> ;
2161      <1> ;
2162      <1> ;
2163      <1> ;
2164      <1> ;
2165      <1> ;
2166      <1> ;
2167      <1> ;
2168      <1> ;
2169      <1> ;
2170      <1> ;
2171      <1> ;
2172      <1> ;
2173      <1> ;
2174      <1> ;
2175      <1> ;
2176      <1> ;
2177      <1> ;
2178      <1> ;
2179      <1> ;
2180      <1> ;
2181      <1> ;
2182      <1> ;
2183      <1> ;
2184      <1> ;
2185      <1> struc MD
2186      00000000 <res 00000001> <1> .SPEC1 resb 1 ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
2187      00000001 <res 00000001> <1> .SPEC2 resb 1 ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
2188      00000002 <res 00000001> <1> .OFF_TIM resb 1 ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
2189      00000003 <res 00000001> <1> .BYT_SEC resb 1 ; 512 BYTES/SECTOR
2190      00000004 <res 00000001> <1> .SEC_TRK resb 1 ; EOT (LAST SECTOR ON TRACK)
2191      00000005 <res 00000001> <1> .GAP resb 1 ; GAP LENGTH
2192      00000006 <res 00000001> <1> .DTL resb 1 ; DTL
2193      00000007 <res 00000001> <1> .GAP3 resb 1 ; GAP LENGTH FOR FORMAT
2194      00000008 <res 00000001> <1> .FIL_BYT resb 1 ; FILL BYTE FOR FORMAT
2195      00000009 <res 00000001> <1> .HD_TIM resb 1 ; HEAD SETTLE TIME (MILLISECONDS)
2196      0000000A <res 00000001> <1> .STR_TIM resb 1 ; MOTOR START TIME (1/8 SECONDS)
2197      0000000B <res 00000001> <1> .MAX_TRK resb 1 ; MAX. TRACK NUMBER
2198      0000000C <res 00000001> <1> .RATE resb 1 ; DATA TRANSFER RATE
2199      <1> endstruc
2200      <1>
2201      <1> BIT70FF EQU 7FH

```

```

2202          <1> BIT70N      EQU      80H
2203          <1>
2204          <1> ;-----
2205          <1> ;      DRIVE TYPE TABLE      :
2206          <1> ;-----
2207          <1> DR_TYPE:
2208          <1>          DB      01          ;DRIVE TYPE, MEDIA TABLE
2209          <1>          DW      MD_TBL1
2210          <1>          DB      02+BIT70N
2211          <1>          DW      MD_TBL2
2212          <1> DR_DEFAULT: DB      02
2213          <1>          DW      MD_TBL3
2214          <1>          DB      03
2215          <1>          DW      MD_TBL4
2216          <1>          DB      04+BIT70N
2217          <1>          DW      MD_TBL5
2218          <1>          DB      04
2219          <1>          DW      MD_TBL6
2220          <1> DR_TYPE_E      equ $          ; END OF TABLE
2221          <1> DR_CNT      EQU      (DR_TYPE_E-DR_TYPE)/3
2222          <1> ;-----
2223          <1> ;      MEDIA/DRIVE PARAMETER TABLES      :
2224          <1> ;-----
2225          <1> ;-----
2226          <1> ;      360 KB MEDIA IN 360 KB DRIVE      :
2227          <1> ;-----
2228          <1> MD_TBL1:
2229          <1>          DB      11011111B      ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
2230          <1>          DB      2              ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
2231          <1>          DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
2232          <1>          DB      2              ; 512 BYTES/SECTOR
2233          <1>          DB      09              ; EOT (LAST SECTOR ON TRACK)
2234          <1>          DB      02AH            ; GAP LENGTH
2235          <1>          DB      0FFH            ; DTL
2236          <1>          DB      050H            ; GAP LENGTH FOR FORMAT
2237          <1>          DB      0F6H            ; FILL BYTE FOR FORMAT
2238          <1>          DB      15              ; HEAD SETTLE TIME (MILLISECONDS)
2239          <1>          DB      8              ; MOTOR START TIME (1/8 SECONDS)
2240          <1>          DB      39              ; MAX. TRACK NUMBER
2241          <1>          DB      RATE_250        ; DATA TRANSFER RATE
2242          <1> ;-----
2243          <1> ;      360 KB MEDIA IN 1.2 MB DRIVE      :
2244          <1> ;-----
2245          <1> MD_TBL2:
2246          <1>          DB      11011111B      ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
2247          <1>          DB      2              ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
2248          <1>          DB      MOTOR_WAIT      ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
2249          <1>          DB      2              ; 512 BYTES/SECTOR
2250          <1>          DB      09              ; EOT (LAST SECTOR ON TRACK)

```

2251	00000B7F	2A	<1>	DB	02AH	; GAP LENGTH
2252	00000B80	FF	<1>	DB	0FFH	; DTL
2253	00000B81	50	<1>	DB	050H	; GAP LENGTH FOR FORMAT
2254	00000B82	F6	<1>	DB	0F6H	; FILL BYTE FOR FORMAT
2255	00000B83	0F	<1>	DB	15	; HEAD SETTLE TIME (MILLISECONDS)
2256	00000B84	08	<1>	DB	8	; MOTOR START TIME (1/8 SECONDS)
2257	00000B85	27	<1>	DB	39	; MAX. TRACK NUMBER
2258	00000B86	40	<1>	DB	RATE_300	; DATA TRANSFER RATE
2259			<1>	;-----		
2260			<1>	; 1.2 MB MEDIA IN 1.2 MB DRIVE :		
2261			<1>	;-----		
2262			<1>	MD_TBL3:		
2263	00000B87	DF	<1>	DB	11011111B	; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
2264	00000B88	02	<1>	DB	2	; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
2265	00000B89	25	<1>	DB	MOTOR_WAIT	; WAIT TIME AFTER OPERATION TILL MOTOR OFF
2266	00000B8A	02	<1>	DB	2	; 512 BYTES/SECTOR
2267	00000B8B	0F	<1>	DB	15	; EOT (LAST SECTOR ON TRACK)
2268	00000B8C	1B	<1>	DB	01BH	; GAP LENGTH
2269	00000B8D	FF	<1>	DB	0FFH	; DTL
2270	00000B8E	54	<1>	DB	054H	; GAP LENGTH FOR FORMAT
2271	00000B8F	F6	<1>	DB	0F6H	; FILL BYTE FOR FORMAT
2272	00000B90	0F	<1>	DB	15	; HEAD SETTLE TIME (MILLISECONDS)
2273	00000B91	08	<1>	DB	8	; MOTOR START TIME (1/8 SECONDS)
2274	00000B92	4F	<1>	DB	79	; MAX. TRACK NUMBER
2275	00000B93	00	<1>	DB	RATE_500	; DATA TRANSFER RATE
2276			<1>	;-----		
2277			<1>	; 720 KB MEDIA IN 720 KB DRIVE :		
2278			<1>	;-----		
2279			<1>	MD_TBL4:		
2280	00000B94	DF	<1>	DB	11011111B	; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
2281	00000B95	02	<1>	DB	2	; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
2282	00000B96	25	<1>	DB	MOTOR_WAIT	; WAIT TIME AFTER OPERATION TILL MOTOR OFF
2283	00000B97	02	<1>	DB	2	; 512 BYTES/SECTOR
2284	00000B98	09	<1>	DB	09	; EOT (LAST SECTOR ON TRACK)
2285	00000B99	2A	<1>	DB	02AH	; GAP LENGTH
2286	00000B9A	FF	<1>	DB	0FFH	; DTL
2287	00000B9B	50	<1>	DB	050H	; GAP LENGTH FOR FORMAT
2288	00000B9C	F6	<1>	DB	0F6H	; FILL BYTE FOR FORMAT
2289	00000B9D	0F	<1>	DB	15	; HEAD SETTLE TIME (MILLISECONDS)
2290	00000B9E	08	<1>	DB	8	; MOTOR START TIME (1/8 SECONDS)
2291	00000B9F	4F	<1>	DB	79	; MAX. TRACK NUMBER
2292	00000BA0	80	<1>	DB	RATE_250	; DATA TRANSFER RATE
2293			<1>	;-----		
2294			<1>	; 720 KB MEDIA IN 1.44 MB DRIVE :		
2295			<1>	;-----		
2296			<1>	MD_TBL5:		
2297	00000BA1	DF	<1>	DB	11011111B	; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
2298	00000BA2	02	<1>	DB	2	; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
2299	00000BA3	25	<1>	DB	MOTOR_WAIT	; WAIT TIME AFTER OPERATION TILL MOTOR OFF

```

2300 00000BA4 02      <1>      DB      2              ; 512 BYTES/SECTOR
2301 00000BA5 09      <1>      DB      09             ; EOT (LAST SECTOR ON TRACK)
2302 00000BA6 2A      <1>      DB      02AH          ; GAP LENGTH
2303 00000BA7 FF      <1>      DB      0FFH          ; DTL
2304 00000BA8 50      <1>      DB      050H          ; GAP LENGTH FOR FORMAT
2305 00000BA9 F6      <1>      DB      0F6H          ; FILL BYTE FOR FORMAT
2306 00000BAA 0F      <1>      DB      15             ; HEAD SETTLE TIME (MILLISECONDS)
2307 00000BAB 08      <1>      DB      8              ; MOTOR START TIME (1/8 SECONDS)
2308 00000BAC 4F      <1>      DB      79             ; MAX. TRACK NUMBER
2309 00000BAD 80      <1>      DB      RATE_250        ; DATA TRANSFER RATE
2310                                     <1> ; -----
2311                                     <1> ; 1.44 MB MEDIA IN 1.44 MB DRIVE :
2312                                     <1> ; -----
2313                                     <1> MD_TBL6:
2314 00000BAE AF      <1>      DB      10101111B      ; SRT=A, HD UNLOAD=0F - 1ST SPECIFY BYTE
2315 00000BAF 02      <1>      DB      2              ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
2316 00000BB0 25      <1>      DB      MOTOR_WAIT     ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
2317 00000BB1 02      <1>      DB      2              ; 512 BYTES/SECTOR
2318 00000BB2 12      <1>      DB      18             ; EOT (LAST SECTOR ON TRACK)
2319 00000BB3 1B      <1>      DB      01BH          ; GAP LENGTH
2320 00000BB4 FF      <1>      DB      0FFH          ; DTL
2321 00000BB5 6C      <1>      DB      06CH          ; GAP LENGTH FOR FORMAT
2322 00000BB6 F6      <1>      DB      0F6H          ; FILL BYTE FOR FORMAT
2323 00000BB7 0F      <1>      DB      15             ; HEAD SETTLE TIME (MILLISECONDS)
2324 00000BB8 08      <1>      DB      8              ; MOTOR START TIME (1/8 SECONDS)
2325 00000BB9 4F      <1>      DB      79             ; MAX. TRACK NUMBER
2326 00000BBA 00      <1>      DB      RATE_500        ; DATA TRANSFER RATE
2327                                     <1>
2328                                     <1> ; 12/12/2014
2329                                     <1> int_40h: ; 11/12/2014
2330                                     <1>
2331                                     <1> ;jmp word far [CS:ORG_INT40H]
2332                                     <1>
2333                                     <1> DISKETTE_IO_1:
2334                                     <1>
2335                                     <1> ;jmp word far [CS:ORG_INT13H]
2336                                     <1>
2337 00000BBB FB      <1>      STI              ; INTERRUPTS BACK ON
2338 00000BBC 55      <1>      PUSH     BP           ; USER REGISTER
2339 00000BBD 57      <1>      PUSH     DI           ; USER REGISTER
2340 00000BBE 52      <1>      PUSH     DX           ; HEAD #, DRIVE # OR USER REGISTER
2341 00000BBF 53      <1>      PUSH     BX           ; BUFFER OFFSET PARAMETER OR REGISTER
2342 00000BC0 51      <1>      PUSH     CX           ; TRACK #-SECTOR # OR USER REGISTER
2343 00000BC1 89E5    <1>      MOV      BP,SP        ; BP => PARAMETER LIST DEP. ON AH
2344                                     <1> ; [BP] = SECTOR #
2345                                     <1> ; [BP+1] = TRACK #
2346                                     <1> ; [BP+2] = BUFFER OFFSET
2347                                     <1> ; FOR RETURN OF DRIVE PARAMETERS:
2348                                     <1> ; CL/[BP] = BITS 7&6 HI BITS OF MAX CYL

```

```

2349          <1>          ;          BITS 0-5 MAX SECTORS/TRACK
2350          <1>          ; CH/[BP+1] = LOW 8 BITS OF MAX CYL.
2351          <1>          ; BL/[BP+2] = BITS 7-4 = 0
2352          <1>          ;          BITS 3-0 = VALID CMOS TYPE
2353          <1>          ; BH/[BP+3] = 0
2354          <1>          ; DL/[BP+4] = # DRIVES INSTALLED
2355          <1>          ; DH/[BP+5] = MAX HEAD #
2356          <1>          ; DI/[BP+6] = OFFSET TO DISK BASE
2357 00000BC3 1E          <1>          PUSH    DS          ; BUFFER SEGMENT PARM OR USER REGISTER
2358 00000BC4 56          <1>          PUSH    SI          ; USER REGISTERS
2359          <1>          ;CALL    DDS          ; SEGMENT OF BIOS DATA AREA TO DS
2360 00000BC5 8CC9        <1>          mov     cx, cs
2361 00000BC7 8ED9        <1>          mov     ds, cx
2362 00000BC9 80FC19      <1>          CMP     AH,(FNC_TAE-FNC_TAB)/2      ; CHECK FOR > LARGEST FUNCTION
2363 00000BCC 7202        <1>          JB      short OK_FUNC          ; FUNCTION OK
2364 00000BCE B414        <1>          MOV     AH,14H          ; REPLACE WITH KNOWN INVALID FUNCTION
2365          <1> OK_FUNC:
2366 00000BD0 80FC01      <1>          CMP     AH,1          ; RESET OR STATUS ?
2367 00000BD3 760C        <1>          JBE     short OK_DRV          ; IF RESET OR STATUS DRIVE ALWAYS OK
2368 00000BD5 80FC08      <1>          CMP     AH,8          ; READ DRIVE PARMS ?
2369 00000BD8 7407        <1>          JZ      short OK_DRV          ; IF SO DRIVE CHECKED LATER
2370 00000BDA 80FA01      <1>          CMP     DL,1          ; DRIVES 0 AND 1 OK
2371 00000BDD 7602        <1>          JBE     short OK_DRV          ; IF 0 OR 1 THEN JUMP
2372 00000BDF B414        <1>          MOV     AH,14H          ; REPLACE WITH KNOWN INVALID FUNCTION
2373          <1> OK_DRV:
2374 00000BE1 88E1        <1>          MOV     CL,AH          ; CL = FUNCTION
2375 00000BE3 30ED        <1>          XOR     CH,CH          ; CX = FUNCTION
2376 00000BE5 D0E1        <1>          SHL     CL,1          ; FUNCTION TIMES 2
2377 00000BE7 BB[140C]    <1>          MOV     BX,FNC_TAB          ; LOAD START OF FUNCTION TABLE
2378 00000BEA 01CB        <1>          ADD     BX,CX          ; ADD OFFSET INTO TABLE => ROUTINE
2379 00000BEC 88F4        <1>          MOV     AH,DH          ; AX = HEAD #,# OF SECTORS OR DASD TYPE
2380 00000BEE 30F6        <1>          XOR     DH,DH          ; DX = DRIVE #
2381 00000BF0 89C6        <1>          MOV     SI,AX          ; SI = HEAD #,# OF SECTORS OR DASD TYPE
2382 00000BF2 89D7        <1>          MOV     DI,DX          ; DI = DRIVE #
2383          <1>          ;
2384          <1>          ; 11/12/2014
2385 00000BF4 8816[590B]  <1>          mov     [cfd], dl          ; current floppy drive (for 'GET_PARM')
2386          <1>          ;
2387 00000BF8 8A26[4A0B]  <1>          MOV     AH, [DSKETTE_STATUS]      ; LOAD STATUS TO AH FOR STATUS FUNCTION
2388 00000BFC C606[4A0B]00 <1>          MOV     byte [DSKETTE_STATUS],0      ; INITIALIZE FOR ALL OTHERS
2389          <1>
2390          <1> ;          THROUGHOUT THE DISKETTE BIOS, THE FOLLOWING INFORMATION IS CONTAINED IN
2391          <1> ;          THE FOLLOWING MEMORY LOCATIONS AND REGISTERS. NOT ALL DISKETTE BIOS
2392          <1> ;          FUNCTIONS REQUIRE ALL OF THESE PARAMETERS.
2393          <1> ;
2394          <1> ;          DI      : DRIVE #
2395          <1> ;          SI-HI   : HEAD #
2396          <1> ;          SI-LOW  : # OF SECTORS OR DASD TYPE FOR FORMAT
2397          <1> ;          ES      : BUFFER SEGMENT

```

```

2398      <1> ;           [BP]   : SECTOR #
2399      <1> ;           [BP+1] : TRACK #
2400      <1> ;           [BP+2] : BUFFER OFFSET
2401      <1> ;
2402      <1> ;           ACROSS CALLS TO SUBROUTINES THE CARRY FLAG (CY=1), WHERE INDICATED IN
2403      <1> ;           SUBROUTINE PROLOGUES, REPRESENTS AN EXCEPTION RETURN (NORMALLY AN ERROR
2404      <1> ;           CONDITION). IN MOST CASES, WHEN CY = 1, @DSKETTE_STATUS CONTAINS THE
2405      <1> ;           SPECIFIC ERROR CODE.
2406      <1> ;
2407      <1> ;           ; (AH) = @DSKETTE_STATUS
2408      00000C01 FF17      <1>      CALL    WORD [BX]           ; CALL THE REQUESTED FUNCTION
2409      00000C03 5E        <1>      POP     SI               ; RESTORE ALL REGISTERS
2410      00000C04 1F        <1>      POP     DS
2411      00000C05 59        <1>      POP     CX
2412      00000C06 5B        <1>      POP     BX
2413      00000C07 5A        <1>      POP     DX
2414      00000C08 5F        <1>      POP     DI
2415      00000C09 89E5      <1>      MOV     BP, SP
2416      00000C0B 50        <1>      PUSH    AX
2417      00000C0C 9C        <1>      PUSHF
2418      00000C0D 58        <1>      POP     AX
2419      00000C0E 894606     <1>      MOV     [BP+6], AX
2420      00000C11 58        <1>      POP     AX
2421      00000C12 5D        <1>      POP     BP
2422      00000C13 CF        <1>      IRET
2423      <1>
2424      <1> ;-----
2425      00000C14 [460C]     <1>      FNC_TAB    DW      DSK_RESET           ; AH = 00H; RESET
2426      00000C16 [9C0C]     <1>      DW      DSK_STATUS           ; AH = 01H; STATUS
2427      00000C18 [A80C]     <1>      DW      DSK_READ            ; AH = 02H; READ
2428      00000C1A [B40C]     <1>      DW      DSK_WRITE           ; AH = 03H; WRITE
2429      00000C1C [C00C]     <1>      DW      DSK_VERF            ; AH = 04H; VERIFY
2430      00000C1E [CC0C]     <1>      DW      DSK_FORMAT          ; AH = 05H; FORMAT
2431      00000C20 [260D]     <1>      DW      FNC_ERR             ; AH = 06H; INVALID
2432      00000C22 [260D]     <1>      DW      FNC_ERR             ; AH = 07H; INVALID
2433      00000C24 [300D]     <1>      DW      DSK_PARMS           ; AH = 08H; READ DRIVE PARAMETERS
2434      00000C26 [260D]     <1>      DW      FNC_ERR             ; AH = 09H; INVALID
2435      00000C28 [260D]     <1>      DW      FNC_ERR             ; AH = 0AH; INVALID
2436      00000C2A [260D]     <1>      DW      FNC_ERR             ; AH = 0BH; INVALID
2437      00000C2C [260D]     <1>      DW      FNC_ERR             ; AH = 0CH; INVALID
2438      00000C2E [260D]     <1>      DW      FNC_ERR             ; AH = 0DH; INVALID
2439      00000C30 [260D]     <1>      DW      FNC_ERR             ; AH = 0EH; INVALID
2440      00000C32 [260D]     <1>      DW      FNC_ERR             ; AH = 0FH; INVALID
2441      00000C34 [260D]     <1>      DW      FNC_ERR             ; AH = 10H; INVALID
2442      00000C36 [260D]     <1>      DW      FNC_ERR             ; AH = 11H; INVALID
2443      00000C38 [260D]     <1>      DW      FNC_ERR             ; AH = 12H; INVALID
2444      00000C3A [260D]     <1>      DW      FNC_ERR             ; AH = 13H; INVALID
2445      00000C3C [260D]     <1>      DW      FNC_ERR             ; AH = 14H; INVALID
2446      00000C3E [DD0D]     <1>      DW      DSK_TYPE            ; AH = 15H; READ DASD TYPE

```

2447 00000C40 [FF0D]	<1>	DW	DSK_CHANGE	; AH = 16H; CHANGE STATUS
2448 00000C42 [2A0E]	<1>	DW	FORMAT_SET	; AH = 17H; SET DASD TYPE
2449 00000C44 [8B0E]	<1>	DW	SET_MEDIA	; AH = 18H; SET MEDIA TYPE
2450	<1>	FNC_TAE EQU	\$	; END
2451	<1>			
2452	<1>			-----
2453	<1>		; DISK_RESET (AH = 00H)	
2454	<1>		RESET THE DISKETTE SYSTEM.	
2455	<1>			
2456	<1>		; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION	
2457	<1>			-----
2458	<1>		DSK_RESET:	
2459 00000C46 BAF203	<1>	MOV	DX, 03F2H	; ADAPTER CONTROL PORT
2460 00000C49 FA	<1>	CLI		; NO INTERRUPTS
2461 00000C4A A0[480B]	<1>	MOV	AL, [MOTOR_STATUS]	; GET DIGITAL OUTPUT REGISTER REFLECTION
2462 00000C4D 243F	<1>	AND	AL, 00111111B	; KEEP SELECTED AND MOTOR ON BITS
2463 00000C4F C0C004	<1>	ROL	AL, 4	; MOTOR VALUE TO HIGH NIBBLE
2464	<1>			; DRIVE SELECT TO LOW NIBBLE
2465 00000C52 0C08	<1>	OR	AL, 00001000B	; TURN ON INTERRUPT ENABLE
2466 00000C54 EE	<1>	OUT	DX, AL	; RESET THE ADAPTER
2467 00000C55 C606[470B]00	<1>	MOV	byte [SEEK_STATUS], 0	; SET RECALIBRATE REQUIRED ON ALL DRIVES
2468	<1>	; JMP	\$+2	; WAIT FOR I/O
2469	<1>	; JMP	\$+2	; WAIT FOR I/O (TO INSURE MINIMUM
2470	<1>			; PULSE WIDTH)
2471	<1>		; 19/12/2014	
2472	<1>		NEWIODELAY	
2473 00000C5A E6EB	<2>	out	0ebh, al	
2474	<1>			
2475	<1>		; 17/12/2014	
2476	<1>		; AWARD BIOS 1999 - RESETDRIVES (ADISK.ASM)	
2477	<1>	; mov	cx, WAITCPU_RESET_ON	; cx = 21 -- Min. 14 micro seconds !?
2478	<1>	; wdw1:		
2479	<1>	; NEWIODELAY		
2480	<1>	; loop	wdw1	
2481	<1>			
2482 00000C5C 0C04	<1>	OR	AL, 00000100B	; TURN OFF RESET BIT
2483 00000C5E EE	<1>	OUT	DX, AL	; RESET THE ADAPTER
2484	<1>		; 16/12/2014	
2485	<1>		IODELAY	
2486 00000C5F EB00	<2>	jmp	short \$+2	
2487 00000C61 EB00	<2>	jmp	short \$+2	
2488	<1>			
2489	<1>	; STI		; ENABLE THE INTERRUPTS
2490 00000C63 E87209	<1>	CALL	WAIT_INT	; WAIT FOR THE INTERRUPT
2491 00000C66 722D	<1>	JC	short DR_ERR	; IF ERROR, RETURN IT
2492 00000C68 B9C000	<1>	MOV	CX, 11000000B	; CL = EXPECTED @NEC_STATUS
2493	<1>		NXT_DRV:	
2494 00000C6B 51	<1>	PUSH	CX	; SAVE FOR CALL
2495 00000C6C B8[940C]	<1>	MOV	AX, DR_POP_ERR	; LOAD NEC_OUTPUT ERROR ADDRESS

2496	00000C6F	50	<1>	PUSH	AX	; "
2497	00000C70	B408	<1>	MOV	AH,08H	; SENSE INTERRUPT STATUS COMMAND
2498	00000C72	E89F08	<1>	CALL	NEC_OUTPUT	
2499	00000C75	58	<1>	POP	AX	; THROW AWAY ERROR RETURN
2500	00000C76	E88209	<1>	CALL	RESULTS	; READ IN THE RESULTS
2501	00000C79	59	<1>	POP	CX	; RESTORE AFTER CALL
2502	00000C7A	7219	<1>	JC	short DR_ERR	; ERROR RETURN
2503	00000C7C	3A0E[4B0B]	<1>	CMP	CL, [NEC_STATUS]	; TEST FOR DRIVE READY TRANSITION
2504	00000C80	7513	<1>	JNZ	short DR_ERR	; EVERYTHING OK
2505	00000C82	FEC1	<1>	INC	CL	; NEXT EXPECTED @NEC_STATUS
2506	00000C84	80F9C3	<1>	CMP	CL,11000011B	; ALL POSSIBLE DRIVES CLEARED
2507	00000C87	76E2	<1>	JBE	short NXT_DRV	; FALL THRU IF 11000100B OR >
2508			<1>			
2509	00000C89	E89802	<1>	CALL	SEND_SPEC	; SEND SPECIFY COMMAND TO NEC
2510			<1>	RESBAC:		
2511	00000C8C	E81907	<1>	CALL	SETUP_END	; VARIOUS CLEANUPS
2512	00000C8F	89F3	<1>	MOV	BX,SI	; GET SAVED AL TO BL
2513	00000C91	88D8	<1>	MOV	AL,BL	; PUT BACK FOR RETURN
2514	00000C93	C3	<1>	RETn		
2515			<1>	DR_POP_ERR:		
2516	00000C94	59	<1>	POP	CX	; CLEAR STACK
2517			<1>	DR_ERR:		
2518	00000C95	800E[4A0B]20	<1>	OR	byte [DSKETTE_STATUS],BAD_NEC	; SET ERROR CODE
2519	00000C9A	EBF0	<1>	JMP	SHORT RESBAC	; RETURN FROM RESET
2520			<1>			
2521			<1>			-----
2522			<1>		DISK_STATUS (AH = 01H)	
2523			<1>		DISKETTE STATUS.	
2524			<1>			
2525			<1>		ON ENTRY: AH : STATUS OF PREVIOUS OPERATION	
2526			<1>			
2527			<1>		ON EXIT: AH, @DSKETTE_STATUS, CY REFLECT STATUS OF PREVIOUS OPERATION.	
2528			<1>			-----
2529			<1>	DSK_STATUS:		
2530	00000C9C	8826[4A0B]	<1>	MOV	[DSKETTE_STATUS],AH	; PUT BACK FOR SETUP END
2531	00000CA0	E80507	<1>	CALL	SETUP_END	; VARIOUS CLEANUPS
2532	00000CA3	89F3	<1>	MOV	BX,SI	; GET SAVED AL TO BL
2533	00000CA5	88D8	<1>	MOV	AL,BL	; PUT BACK FOR RETURN
2534	00000CA7	C3	<1>	RETn		
2535			<1>			
2536			<1>			-----
2537			<1>		DISK_READ (AH = 02H)	
2538			<1>		DISKETTE READ.	
2539			<1>			
2540			<1>		ON ENTRY: DI : DRIVE #	
2541			<1>		SI-HI : HEAD #	
2542			<1>		SI-LOW : # OF SECTORS	
2543			<1>		ES : BUFFER SEGMENT	
2544			<1>		[BP] : SECTOR #	

2545		<1> ;	[BP+1] : TRACK #
2546		<1> ;	[BP+2] : BUFFER OFFSET
2547		<1> ;	
2548		<1> ;	ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
2549		<1> ;	-----
2550		<1>	DSK_READ:
2551	00000CA8 8026[480B]7F	<1>	AND byte [MOTOR_STATUS],01111111B ; INDICATE A READ OPERATION
2552	00000CAD B846E6	<1>	MOV AX,0E646H ; AX = NEC COMMAND, DMA COMMAND
2553	00000CB0 E84503	<1>	CALL RD_WR_VF ; COMMON READ/WRITE/VERIFY
2554	00000CB3 C3	<1>	RETn
2555		<1>	
2556		<1> ;	-----
2557		<1> ;	DISK_WRITE (AH = 03H)
2558		<1> ;	DISKETTE WRITE.
2559		<1> ;	
2560		<1> ;	ON ENTRY: DI : DRIVE #
2561		<1> ;	SI-HI : HEAD #
2562		<1> ;	SI-LOW : # OF SECTORS
2563		<1> ;	ES : BUFFER SEGMENT
2564		<1> ;	[BP] : SECTOR #
2565		<1> ;	[BP+1] : TRACK #
2566		<1> ;	[BP+2] : BUFFER OFFSET
2567		<1> ;	
2568		<1> ;	ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
2569		<1> ;	-----
2570		<1>	DSK_WRITE:
2571	00000CB4 B84AC5	<1>	MOV AX,0C54AH ; AX = NEC COMMAND, DMA COMMAND
2572	00000CB7 800E[480B]80	<1>	OR byte [MOTOR_STATUS],10000000B ; INDICATE WRITE OPERATION
2573	00000CBC E83903	<1>	CALL RD_WR_VF ; COMMON READ/WRITE/VERIFY
2574	00000CBF C3	<1>	RETn
2575		<1>	
2576		<1> ;	-----
2577		<1> ;	DISK_VERF (AH = 04H)
2578		<1> ;	DISKETTE VERIFY.
2579		<1> ;	
2580		<1> ;	ON ENTRY: DI : DRIVE #
2581		<1> ;	SI-HI : HEAD #
2582		<1> ;	SI-LOW : # OF SECTORS
2583		<1> ;	ES : BUFFER SEGMENT
2584		<1> ;	[BP] : SECTOR #
2585		<1> ;	[BP+1] : TRACK #
2586		<1> ;	[BP+2] : BUFFER OFFSET
2587		<1> ;	
2588		<1> ;	ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
2589		<1> ;	-----
2590		<1>	DSK_VERF:
2591	00000CC0 8026[480B]7F	<1>	AND byte [MOTOR_STATUS],01111111B ; INDICATE A READ OPERATION
2592	00000CC5 B842E6	<1>	MOV AX,0E642H ; AX = NEC COMMAND, DMA COMMAND
2593	00000CC8 E82D03	<1>	CALL RD_WR_VF ; COMMON READ/WRITE/VERIFY

2594	00000CCB C3	<1>	RETn	
2595		<1>		
2596		<1>	;	-----
2597		<1>	; DISK_FORMAT (AH = 05H)	
2598		<1>	; DISKETTE FORMAT.	
2599		<1>	;	
2600		<1>	; ON ENTRY: DI : DRIVE #	
2601		<1>	; SI-HI : HEAD #	
2602		<1>	; SI-LOW : # OF SECTORS	
2603		<1>	; ES : BUFFER SEGMENT	
2604		<1>	; [BP] : SECTOR #	
2605		<1>	; [BP+1] : TRACK #	
2606		<1>	; [BP+2] : BUFFER OFFSET	
2607		<1>	; @DISK_POINTER POINTS TO THE PARAMETER TABLE OF THIS DRIVE	
2608		<1>	;	
2609		<1>	; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION	
2610		<1>	;	-----
2611		<1>	DSK_FORMAT:	
2612	00000CCC E88A02	<1>	CALL XLAT_NEW	; TRANSLATE STATE TO PRESENT ARCH.
2613	00000CCF E81004	<1>	CALL FMT_INIT	; ESTABLISH STATE IF UNESTABLISHED
2614	00000CD2 800E[480B]80	<1>	OR byte [MOTOR_STATUS], 10000000B	; INDICATE WRITE OPERATION
2615	00000CD7 E85404	<1>	CALL MED_CHANGE	; CHECK MEDIA CHANGE AND RESET IF SO
2616	00000CDA 723F	<1>	JC short FM_DON	; MEDIA CHANGED, SKIP
2617	00000CDC E84502	<1>	CALL SEND_SPEC	; SEND SPECIFY COMMAND TO NEC
2618	00000CDF E8A004	<1>	CALL CHK_LASRATE	; ZF=1 ATTEMPT RATE IS SAME AS LAST RATE
2619	00000CE2 7403	<1>	JZ short FM_WR	; YES, SKIP SPECIFY COMMAND
2620	00000CE4 E88204	<1>	CALL SEND_RATE	; SEND DATA RATE TO CONTROLLER
2621		<1>	FM_WR:	
2622	00000CE7 E81005	<1>	CALL FMTDMA_SET	; SET UP THE DMA FOR FORMAT
2623	00000CEA 722F	<1>	JC short FM_DON	; RETURN WITH ERROR
2624	00000CEC B44D	<1>	MOV AH,04DH	; ESTABLISH THE FORMAT COMMAND
2625	00000CEE E86405	<1>	CALL NEC_INIT	; INITIALIZE THE NEC
2626	00000CF1 7228	<1>	JC short FM_DON	; ERROR - EXIT
2627	00000CF3 B8[1B0D]	<1>	MOV AX, FM_DON	; LOAD ERROR ADDRESS
2628	00000CF6 50	<1>	PUSH AX	; PUSH NEC_OUT ERROR RETURN
2629	00000CF7 B203	<1>	MOV DL,3	; BYTES/SECTOR VALUE TO NEC
2630	00000CF9 E84B07	<1>	CALL GET_PARM	
2631	00000CFC E81508	<1>	CALL NEC_OUTPUT	
2632	00000CFF B204	<1>	MOV DL,4	; SECTORS/TRACK VALUE TO NEC
2633	00000D01 E84307	<1>	CALL GET_PARM	
2634	00000D04 E80D08	<1>	CALL NEC_OUTPUT	
2635	00000D07 B207	<1>	MOV DL,7	; GAP LENGTH VALUE TO NEC
2636	00000D09 E83B07	<1>	CALL GET_PARM	
2637	00000D0C E80508	<1>	CALL NEC_OUTPUT	
2638	00000D0F B208	<1>	MOV DL,8	; FILLER BYTE TO NEC
2639	00000D11 E83307	<1>	CALL GET_PARM	
2640	00000D14 E8FD07	<1>	CALL NEC_OUTPUT	
2641	00000D17 58	<1>	POP AX	; THROW AWAY ERROR
2642	00000D18 E89405	<1>	CALL NEC_TERM	; TERMINATE, RECEIVE STATUS, ETC,

```

2643                                     <1> FM_DON:
2644 00000D1B E86102                 <1>      CALL   XLAT_OLD           ; TRANSLATE STATE TO COMPATIBLE MODE
2645 00000D1E E88706                 <1>      CALL   SETUP_END        ; VARIOUS CLEANUPS
2646 00000D21 89F3                   <1>      MOV     BX,SI           ; GET SAVED AL TO BL
2647 00000D23 88D8                   <1>      MOV     AL,BL           ; PUT BACK FOR RETURN
2648 00000D25 C3                     <1>      RETn
2649                                 <1>
2650                                 <1> ;-----
2651                                 <1> ; FNC_ERR
2652                                 <1> ;     INVALID FUNCTION REQUESTED OR INVALID DRIVE:
2653                                 <1> ;     SET BAD COMMAND IN STATUS.
2654                                 <1> ;
2655                                 <1> ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
2656                                 <1> ;-----
2657                                 <1> FNC_ERR:
2658 00000D26 89F0                   <1>      MOV     AX,SI           ; RESTORE AL
2659 00000D28 B401                   <1>      MOV     AH,BAD_CMD        ; SET BAD COMMAND ERROR
2660 00000D2A 8826[4A0B]             <1>      MOV     [DSKETTE_STATUS],AH ; STORE IN DATA AREA
2661 00000D2E F9                     <1>      STC                     ; SET CARRY INDICATING ERROR
2662 00000D2F C3                     <1>      RETn
2663                                 <1>
2664                                 <1> ;-----
2665                                 <1> ; DISK_PARMS (AH = 08H)
2666                                 <1> ;     READ DRIVE PARAMETERS.
2667                                 <1> ;
2668                                 <1> ; ON ENTRY:  DI : DRIVE #
2669                                 <1> ;
2670                                 <1> ; ON EXIT:  CL/[BP]   = BITS 7 & 6 HI 2 BITS OF MAX CYLINDER
2671                                 <1> ;                      BITS 0-5 MAX SECTORS/TRACK
2672                                 <1> ;                      CH/[BP+1] = LOW 8 BITS OF MAX CYLINDER
2673                                 <1> ;                      BL/[BP+2] = BITS 7-4 = 0
2674                                 <1> ;                      BITS 3-0 = VALID CMOS DRIVE TYPE
2675                                 <1> ;                      BH/[BP+3] = 0
2676                                 <1> ;                      DL/[BP+4] = # DRIVES INSTALLED (VALUE CHECKED)
2677                                 <1> ;                      DH/[BP+5] = MAX HEAD #
2678                                 <1> ;                      DI/[BP+6] = OFFSET TO DISK_BASE
2679                                 <1> ;                      ES          = SEGMENT OF DISK_BASE
2680                                 <1> ;                      AX          = 0
2681                                 <1> ;
2682                                 <1> ;
2683                                 <1> ;     NOTE : THE ABOVE INFORMATION IS STORED IN THE USERS STACK AT
2684                                 <1> ;     THE LOCATIONS WHERE THE MAIN ROUTINE WILL POP THEM
2685                                 <1> ;     INTO THE APPROPRIATE REGISTERS BEFORE RETURNING TO THE
2686                                 <1> ;     CALLER.
2687                                 <1> ;-----
2688                                 <1> DSK_PARMS:
2689 00000D30 E82602                 <1>      CALL   XLAT_NEW           ; TRANSLATE STATE TO PRESENT ARCH,
2690 00000D33 C746020000             <1>      MOV     WORD [BP+2],0      ; DRIVE TYPE = 0
2691                                 <1> ; MOV     AX, [EQUIP_FLAG]      ; LOAD EQUIPMENT FLAG FOR # DISKETTES
2691                                 <1> ; AND     AL,11000001B      ; KEEP DISKETTE DRIVE BITS

```

```

2692          <1>      ; MOV      DL,2                ; DISKETTE DRIVES = 2
2693          <1>      ; CMP      AL,01000001B        ; 2 DRIVES INSTALLED ?
2694          <1>      ; JZ       short STO_DL         ; IF YES JUMP
2695          <1>
2696          <1>      ; DEC      DL                    ; DISKETTE DRIVES = 1
2697          <1>      ; CMP      AL,00000001B        ; 1 DRIVE INSTALLED ?
2698          <1>      ; JNZ      short NON_DRV         ; IF NO JUMP
2699          <1>
2700 00000D38 28D2    <1>      sub      dl, dl
2701 00000D3A A1[3829] <1>      mov      ax, [fd0_type]
2702 00000D3D 21C0    <1>      and      ax, ax
2703 00000D3F 7468    <1>      jz       short NON_DRV
2704 00000D41 FEC2    <1>      inc      dl
2705 00000D43 20E4    <1>      and      ah, ah
2706 00000D45 7402    <1>      jz       short STO_DL
2707 00000D47 FEC2    <1>      inc      dl
2708          <1>      STO_DL:
2709 00000D49 885604   <1>      MOV      [BP+4],DL        ; STORE NUMBER OF DRIVES
2710 00000D4C 83FF01   <1>      CMP      DI,1                ; CHECK FOR VALID DRIVE
2711 00000D4F 775C     <1>      JA       short NON_DRV1        ; DRIVE INVALID
2712 00000D51 C6460501 <1>      MOV      BYTE [BP+5],1        ; MAXIMUM HEAD NUMBER = 1
2713 00000D55 E8E806   <1>      CALL     CMOS_TYPE        ; RETURN DRIVE TYPE IN AL
2714          <1>      ;JC      short CHK_EST        ; IF CMOS BAD CHECKSUM ESTABLISHED
2715          <1>      ;OR      AL,AL                ; TEST FOR NO DRIVE TYPE
2716 00000D58 7410     <1>      JZ       short CHK_EST        ; JUMP IF SO
2717 00000D5A E8A601   <1>      CALL     DR_TYPE_CHECK        ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
2718 00000D5D 720B     <1>      JC       short CHK_EST        ; TYPE NOT IN TABLE (POSSIBLE BAD CMOS)
2719 00000D5F 884602   <1>      MOV      [BP+2],AL        ; STORE VALID CMOS DRIVE TYPE
2720 00000D62 8A4F04   <1>      MOV      CL, [BX+MD.SEC_TRK]    ; GET SECTOR/TRACK
2721 00000D65 8A6F0B   <1>      MOV      CH, [BX+MD.MAX_TRK]    ; GET MAX. TRACK NUMBER
2722 00000D68 EB2E     <1>      JMP      SHORT STO_CX        ; CMOS GOOD, USE CMOS
2723          <1>      CHK_EST:
2724 00000D6A 8AA5[530B] <1>      MOV      AH, [DSK_STATE+DI]    ; LOAD STATE FOR THIS DRIVE
2725 00000D6E F6C410   <1>      TEST     AH,MED_DET        ; CHECK FOR ESTABLISHED STATE
2726 00000D71 743A     <1>      JZ       short NON_DRV1        ; CMOS BAD/INVALID OR UNESTABLISHED
2727          <1>      USE_EST:
2728 00000D73 80E4C0   <1>      AND      AH,RATE_MSK        ; ISOLATE STATE
2729 00000D76 80FC80   <1>      CMP      AH,RATE_250        ; RATE 250 ?
2730 00000D79 7550     <1>      JNE      short USE_EST2        ; NO, GO CHECK OTHER RATE
2731          <1>
2732          <1>      ;----- DATA RATE IS 250 KBS, TRY 360 KB TABLE FIRST
2733          <1>
2734 00000D7B B001     <1>      MOV      AL,01                ; DRIVE TYPE 1 (360KB)
2735 00000D7D E88301   <1>      CALL     DR_TYPE_CHECK        ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
2736 00000D80 8A4F04   <1>      MOV      CL, [BX+MD.SEC_TRK]    ; GET SECTOR/TRACK
2737 00000D83 8A6F0B   <1>      MOV      CH, [BX+MD.MAX_TRK]    ; GET MAX. TRACK NUMBER
2738 00000D86 F685[530B]01 <1>      TEST     byte [DSK_STATE+DI],TRK_CAPA ; 80 TRACK ?
2739 00000D8B 740B     <1>      JZ       short STO_CX        ; MUST BE 360KB DRIVE
2740          <1>

```

```

2741 <1> ;----- IT IS 1.44 MB DRIVE
2742 <1>
2743 <1> PARM144:
2744 00000D8D B004 <1> MOV AL,04 ; DRIVE TYPE 4 (1.44MB)
2745 00000D8F E87101 <1> CALL DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
2746 00000D92 8A4F04 <1> MOV CL, [BX+MD.SEC_TRK] ; GET SECTOR/TRACK
2747 00000D95 8A6F0B <1> MOV CH, [BX+MD.MAX_TRK] ; GET MAX. TRACK NUMBER
2748 <1> STO_CX:
2749 00000D98 894E00 <1> MOV [BP],CX ; SAVE POINTER IN STACK FOR RETURN
2750 <1> ES_DI:
2751 00000D9B 895E06 <1> MOV [BP+6],BX ; ADDRESS OF MEDIA/DRIVE PARM TABLE
2752 00000D9E 8CC8 <1> MOV AX,CS ; SEGMENT MEDIA/DRIVE PARAMETER TABLE
2753 00000DA0 8EC0 <1> MOV ES,AX ; ES IS SEGMENT OF TABLE
2754 <1> DP_OUT:
2755 00000DA2 E8DA01 <1> CALL XLAT_OLD ; TRANSLATE STATE TO COMPATIBLE MODE
2756 00000DA5 31C0 <1> XOR AX,AX ; CLEAR
2757 00000DA7 F8 <1> CLC
2758 00000DA8 C3 <1> RETn
2759 <1>
2760 <1> ;----- NO DRIYE PRESENT HANDLER
2761 <1>
2762 <1> NON_DRV:
2763 00000DA9 C6460400 <1> MOV BYTE [BP+4],0 ; CLEAR NUMBER OF DRIVES
2764 <1> NON_DRV1:
2765 00000DAD 81FF8000 <1> CMP DI,80H ; CHECK FOR FIXED MEDIA TYPE REQUEST
2766 00000DB1 7209 <1> JB short NON_DRV2 ; CONTINUE IF NOT REQUEST FALL THROUGH
2767 <1>
2768 <1> ;----- FIXED DISK REQUEST FALL THROUGH ERROR
2769 <1>
2770 00000DB3 E8C901 <1> CALL XLAT_OLD ; ELSE TRANSLATE TO COMPATIBLE MODE
2771 00000DB6 89F0 <1> MOV AX,SI ; RESTORE AL
2772 00000DB8 B401 <1> MOV AH,BAD_CMD ; SET BAD COMMAND ERROR
2773 00000DBA F9 <1> STC
2774 00000DBB C3 <1> RETn
2775 <1>
2776 <1> NON_DRV2:
2777 00000DBC 31C0 <1> XOR AX,AX ; CLEAR PARMS IF NO DRIVES OR CMOS BAD
2778 00000DBE 894600 <1> MOV [BP],AX ; TRACKS, SECTORS/TRACK = 0
2779 00000DC1 886605 <1> MOV [BP+5],AH ; HEAD = 0
2780 00000DC4 894606 <1> MOV [BP+6],AX ; OFFSET TO DISK_BASE = 0
2781 00000DC7 8EC0 <1> MOV ES,AX ; ES IS SEGMENT OF TABLE
2782 00000DC9 EBD7 <1> JMP SHORT DP_OUT
2783 <1>
2784 <1> ;----- DATA RATE IS EITHER 300 KBS OR 500 KBS, TRY 1.2 MB TABLE FIRST
2785 <1>
2786 <1> USE_EST2:
2787 00000DCB B002 <1> MOV AL,02 ; DRIVE TYPE 2 (1.2MB)
2788 00000DCD E83301 <1> CALL DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
2789 00000DD0 8A4F04 <1> MOV CL, [BX+MD.SEC_TRK] ; GET SECTOR/TRACK

```

2790 00000DD3 8A6F0B	<1>	MOV	CH, [BX+MD.MAX_TRK]	; GET MAX. TRACK NUMBER
2791 00000DD6 80FC40	<1>	CMP	AH, RATE_300	; RATE 300 ?
2792 00000DD9 74BD	<1>	JZ	short ST0_CX	; MUST BE 1.2MB DRIVE
2793 00000ddb EBB0	<1>	JMP	SHORT PARM144	; ELSE, IT IS 1.44MB DRIVE
2794	<1>			
2795	<1>			
2796	<1>			
2797	<1>			
2798	<1>			
2799	<1>			
2800	<1>			
2801	<1>			
2802	<1>			
2803	<1>			
2804 00000DDD E87901	<1>	CALL	XLAT_NEW	; TRANSLATE STATE TO PRESENT ARCH.
2805 00000DE0 8A85[530B]	<1>	MOV	AL, [DSK_STATE+DI]	; GET PRESENT STATE INFORMATION
2806 00000DE4 08C0	<1>	OR	AL, AL	; CHECK FOR NO DRIVE
2807 00000DE6 7413	<1>	JZ	short NO_DRV	
2808 00000DE8 B401	<1>	MOV	AH, NOCHGLN	; NO CHANGE LINE FOR 40 TRACK DRIVE
2809 00000DEA A801	<1>	TEST	AL, TRK_CAPA	; IS THIS DRIVE AN 80 TRACK DRIVE?
2810 00000DEC 7402	<1>	JZ	short DT_BACK	; IF NO JUMP
2811 00000DEE B402	<1>	MOV	AH, CHGLN	; CHANGE LINE FOR 80 TRACK DRIVE
2812	<1>			
2813 00000DF0 50	<1>	PUSH	AX	; SAVE RETURN VALUE
2814 00000DF1 E88B01	<1>	CALL	XLAT_OLD	; TRANSLATE STATE TO COMPATIBLE MODE
2815 00000DF4 58	<1>	POP	AX	; RESTORE RETURN VALUE
2816 00000DF5 F8	<1>	CLC		; NO ERROR
2817 00000DF6 89F3	<1>	MOV	BX, SI	; GET SAVED AL TO BL
2818 00000DF8 88D8	<1>	MOV	AL, BL	; PUT BACK FOR RETURN
2819 00000DFA C3	<1>	RETn		
2820	<1>			
2821 00000DFB 30E4	<1>	XOR	AH, AH	; NO DRIVE PRESENT OR UNKNOWN
2822 00000DFD EBF1	<1>	JMP	SHORT DT_BACK	
2823	<1>			
2824	<1>			
2825	<1>			
2826	<1>			
2827	<1>			
2828	<1>			
2829	<1>			
2830	<1>			
2831	<1>			
2832	<1>			
2833	<1>			
2834	<1>			
2835 00000DFF E85701	<1>	CALL	XLAT_NEW	; TRANSLATE STATE TO PRESENT ARCH.
2836 00000E02 8A85[530B]	<1>	MOV	AL, [DSK_STATE+DI]	; GET MEDIA STATE INFORMATION
2837 00000E06 08C0	<1>	OR	AL, AL	; DRIVE PRESENT ?
2838 00000E08 7419	<1>	JZ	short DC_NON	; JUMP IF NO DRIVE

2839 00000E0A A801	<1>	TEST	AL,TRK_CAPA	; 80 TRACK DRIVE ?
2840 00000E0C 7405	<1>	JZ	short SETIT	; IF S0 , CHECK CHANGE LINE
2841	<1>	DC0:		
2842 00000E0E E83A08	<1>	CALL	READ_DSKCHNG	; GO CHECK STATE OF DISK CHANGE LINE
2843 00000E11 7405	<1>	JZ	short FINIS	; CHANGE LINE NOT ACTIVE
2844	<1>			
2845 00000E13 C606[4A0B]06	<1>	SETIT:	MOV byte [DSKETTE_STATUS], MEDIA_CHANGE	; INDICATE MEDIA REMOVED
2846	<1>			
2847 00000E18 E86401	<1>	FINIS:	CALL XLAT_OLD	; TRANSLATE STATE TO COMPATIBLE MODE
2848 00000E1B E88A05	<1>	CALL	SETUP_END	; VARIOUS CLEANUPS
2849 00000E1E 89F3	<1>	MOV	BX,SI	; GET SAVED AL TO BL
2850 00000E20 88D8	<1>	MOV	AL,BL	; PUT BACK FOR RETURN
2851 00000E22 C3	<1>	RETn		
2852	<1>	DC_NON:		
2853 00000E23 800E[4A0B]80	<1>	OR	byte [DSKETTE_STATUS], TIME_OUT	; SET TIMEOUT, NO DRIVE
2854 00000E28 EBEE	<1>	JMP	SHORT FINIS	
2855	<1>			
2856	<1>			
2857	<1>			
2858	<1>			
2859	<1>			
2860	<1>			
2861	<1>			
2862	<1>			
2863	<1>			
2864	<1>			
2865	<1>			
2866	<1>			
2867	<1>			
2868	<1>			
2869 00000E2A E82C01	<1>	CALL	XLAT_NEW	; TRANSLATE STATE TO PRESENT ARCH.
2870 00000E2D 56	<1>	PUSH	SI	; SAVE DASD TYPE
2871 00000E2E 89F0	<1>	MOV	AX,SI	; AH = ? , AL , DASD TYPE
2872 00000E30 30E4	<1>	XOR	AH,AH	; AH , 0 , AL , DASD TYPE
2873 00000E32 89C6	<1>	MOV	SI,AX	; SI = DASD TYPE
2874 00000E34 80A5[530B]0F	<1>	AND	byte [DSK_STATE+DI], ~(MED_DET+DBL_STEP+RATE_MSK)	; CLEAR STATE
2875 00000E39 4E	<1>	DEC	SI	; CHECK FOR 320/360K MEDIA & DRIVE
2876 00000E3A 7507	<1>	JNZ	short NOT_320	; BYPASS IF NOT
2877 00000E3C 808D[530B]90	<1>	OR	byte [DSK_STATE+DI], MED_DET+RATE_250	; SET TO 320/360
2878 00000E41 EB37	<1>	JMP	SHORT S0	
2879	<1>			
2880	<1>	NOT_320:		
2881 00000E43 E8E802	<1>	CALL	MED_CHANGE	; CHECK FOR TIME_OUT
2882 00000E46 803E[4A0B]80	<1>	CMP	byte [DSKETTE_STATUS], TIME_OUT	
2883 00000E4B 742D	<1>	JZ	short S0	; IF TIME OUT TELL CALLER
2884	<1>	S3:		
2885 00000E4D 4E	<1>	DEC	SI	; CHECK FOR 320/360K IN 1.2M DRIVE
2886 00000E4E 7507	<1>	JNZ	short NOT_320_12	; BYPASS IF NOT
2887 00000E50 808D[530B]70	<1>	OR	byte [DSK_STATE+DI], MED_DET+DBL_STEP+RATE_300	; SET STATE

2888 00000E55 EB23	<1> JMP SHORT S0
2889	<1>
2890	<1> NOT_320_12:
2891 00000E57 4E	<1> DEC SI ; CHECK FOR 1.2M MEDIA IN 1.2M DRIVE
2892 00000E58 7507	<1> JNZ short NOT_12 ; BYPASS IF NOT
2893 00000E5A 808D[530B]10	<1> OR byte [DSK_STATE+DI], MED_DET+RATE_500 ; SET STATE VARIABLE
2894 00000E5F EB19	<1> JMP SHORT S0 ; RETURN TO CALLER
2895	<1>
2896	<1> NOT_12:
2897 00000E61 4E	<1> DEC SI ; CHECK FOR SET DASD TYPE 04
2898 00000E62 7520	<1> JNZ short FS_ERR ; BAD COMMAND EXIT IF NOT VALID TYPE
2899	<1>
2900 00000E64 F685[530B]04	<1> TEST byte [DSK_STATE+DI], DRV_DET ; DRIVE DETERMINED ?
2901 00000E69 7409	<1> JZ short ASSUME ; IF STILL NOT DETERMINED ASSUME
2902 00000E6B B050	<1> MOV AL,MED_DET+RATE_300
2903 00000E6D F685[530B]02	<1> TEST byte [DSK_STATE+DI], FMT_CAPA ; MULTIPLE FORMAT CAPABILITY ?
2904 00000E72 7502	<1> JNZ short OR_IT_IN ; IF 1.2 M THEN DATA RATE 300
2905	<1>
2906	<1> ASSUME:
2907 00000E74 B090	<1> MOV AL,MED_DET+RATE_250 ; SET UP
2908	<1>
2909	<1> OR_IT_IN:
2910 00000E76 0885[530B]	<1> OR byte [DSK_STATE+DI], AL ; OR IN THE CORRECT STATE
2911	<1> S0:
2912 00000E7A E80201	<1> CALL XLAT_OLD ; TRANSLATE STATE TO COMPATIBLE MODE
2913 00000E7D E82805	<1> CALL SETUP_END ; VARIOUS CLEANUPS
2914 00000E80 5B	<1> POP BX ; GET SAVED AL TO BL
2915 00000E81 88D8	<1> MOV AL,BL ; PUT BACK FOR RETURN
2916 00000E83 C3	<1> RETn
2917	<1>
2918	<1> FS_ERR:
2919 00000E84 C606[4A0B]01	<1> MOV byte [DSKETTE_STATUS], BAD_CMD ; UNKNOWN STATE,BAD COMMAND
2920 00000E89 EBEF	<1> JMP SHORT S0
2921	<1>
2922	<1> ; -----
2923	<1> ; SET_MEDIA (AH = 18H)
2924	<1> ; THIS ROUTINE SETS THE TYPE OF MEDIA AND DATA RATE
2925	<1> ; TO BE USED FOR THE FOLLOWING FORMAT OPERATION.
2926	<1> ;
2927	<1> ; ON ENTRY:
2928	<1> ; [BP] = SECTOR PER TRACK
2929	<1> ; [BP+1] = TRACK #
2930	<1> ; DI = DRIVE #
2931	<1> ;
2932	<1> ; ON EXIT:
2933	<1> ; @DSKETTE_STATUS REFLECTS STATUS
2934	<1> ; IF NO ERROR:
2935	<1> ; AH = 0
2936	<1> ; CY = 0

2937		<1> ;	ES = SEGMENT OF MEDIA/DRIVE PARAMETER TABLE
2938		<1> ;	DI/[BP+6] = OFFSET OF MEDIA/DRIVE PARAMETER TABLE
2939		<1> ;	IF ERROR:
2940		<1> ;	AH = @DSKETTE_STATUS
2941		<1> ;	CY = 1
2942		<1> ;	-----
2943		<1> SET_MEDIA:	
2944	00000E8B E8CB00	<1> CALL	XLAT_NEW ; TRANSLATE STATE TO PRESENT ARCH.
2945	00000E8E F685[530B]01	<1> TEST	byte [DSK_STATE+DI], TRK_CAPA ; CHECK FOR CHANGE LINE AVAILABLE
2946	00000E93 740F	<1> JZ	short SM_CMOS ; JUMP IF 40 TRACK DRIVE
2947	00000E95 E89602	<1> CALL	MED_CHANGE ; RESET CHANGE LINE
2948	00000E98 803E[4A0B]80	<1> CMP	byte [DSKETTE_STATUS], TIME_OUT ; IF TIME OUT TELL CALLER
2949	00000E9D 745D	<1> JE	short SM_RTN
2950	00000E9F C606[4A0B]00	<1> MOV	byte [DSKETTE_STATUS], 0 ; CLEAR STATUS
2951		<1> SM_CMOS:	
2952	00000EA4 E89905	<1> CALL	CMOS_TYPE ; RETURN DRIVE TYPE IN (AL)
2953		<1> ;JC	short MD_NOT_FND ; ERROR IN CMOS
2954		<1> ;OR	AL,AL ; TEST FOR NO DRIVE
2955	00000EA7 7453	<1> JZ	short SM_RTN ; RETURN IF SO
2956	00000EA9 E85700	<1> CALL	DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
2957	00000EAC 722B	<1> JC	short MD_NOT_FND ; TYPE NOT IN TABLE (BAD CMOS)
2958	00000EAE 57	<1> PUSH	DI ; SAVE REG.
2959	00000EAF 31DB	<1> XOR	BX,BX ; BX = INDEX TO DR. TYPE TABLE
2960	00000EB1 B90600	<1> MOV	CX,DR_CNT ; CX = LOOP COUNT
2961		<1> DR_SEARCH:	
2962	00000EB4 8AA7[5B0B]	<1> MOV	AH, [DR_TYPE+BX] ; GET DRIVE TYPE
2963	00000EB8 80E47F	<1> AND	AH,BIT70FF ; MASK OUT MSB
2964	00000EBB 38E0	<1> CMP	AL,AH ; DRIVE TYPE MATCH ?
2965	00000EBD 7514	<1> JNE	short NXT_MD ; NO, CHECK NEXT DRIVE TYPE
2966		<1> DR_FND:	
2967	00000EBF 8BBF[5C0B]	<1> MOV	DI, [DR_TYPE+BX+1] ; DI = MEDIA/DRIVE PARAM TABLE
2968		<1> MD_SEARCH:	
2969	00000EC3 8A6504	<1> MOV	AH, [DI+MD.SEC_TRK] ; GET SECTOR/TRACK
2970	00000EC6 386600	<1> CMP	[BP],AH ; MATCH?
2971	00000EC9 7508	<1> JNE	short NXT_MD ; NO, CHECK NEXT MEDIA
2972	00000ECB 8A650B	<1> MOV	AH, [DI+MD.MAX_TRK] ; GET MAX. TRACK #
2973	00000ECE 386601	<1> CMP	[BP+1],AH ; MATCH?
2974	00000ED1 740D	<1> JE	short MD_FND ; YES, GO GET RATE
2975		<1> NXT_MD:	
2976	00000ED3 83C303	<1> ADD	BX,3 ; CHECK NEXT DRIVE TYPE
2977	00000ED6 E2DC	<1> LOOP	DR_SEARCH
2978	00000ED8 5F	<1> POP	DI ; RESTORE REG.
2979		<1> MD_NOT_FND:	
2980	00000ED9 C606[4A0B]0C	<1> MOV	byte [DSKETTE_STATUS], MD_NOT_FND ; ERROR, MEDIA TYPE NOT FOUND
2981	00000EDE EB1C	<1> JMP	SHORT SM_RTN ; RETURN
2982		<1> MD_FND:	
2983	00000EE0 8A450C	<1> MOV	AL, [DI+MD.RATE] ; GET RATE
2984	00000EE3 3C40	<1> CMP	AL,RATE_300 ; DOUBLE STEP REQUIRED FOR RATE 300
2985	00000EE5 7502	<1> JNE	short MD_SET

```

2986 00000EE7 0C20      <1>      OR      AL, DBL_STEP
2987                    <1> MD_SET:
2988 00000EE9 897E06      <1>      MOV      [BP+6], DI          ; SAVE TABLE POINTER IN STACK
2989 00000EEC 0C10      <1>      OR      AL, MED_DET          ; SET MEDIA ESTABLISHED
2990 00000EEE 5F          <1>      POP      DI
2991 00000EEF 80A5[530B]0F <1>      AND      byte [DSK_STATE+DI], ~(MED_DET+DBL_STEP+RATE_MSK) ; CLEAR STATE
2992 00000EF4 0885[530B] <1>      OR      byte [DSK_STATE+DI], AL
2993 00000EF8 8CC8      <1>      MOV      AX, CS          ; SEGMENT OF MEDIA/DRIVE PARAMETER TABLE
2994 00000EFA 8EC0      <1>      MOV      ES, AX          ; ES IS SEGMENT OF TABLE
2995                    <1> SM_RTN:
2996 00000EFC E88000      <1>      CALL     XLAT_OLD          ; TRANSLATE STATE TO COMPATIBLE MODE
2997 00000EFF E8A604      <1>      CALL     SETUP_END        ; VARIOUS CLEANUPS
2998 00000F02 C3          <1>      RETn
2999                    <1>
3000                    <1> ; -----
3001                    <1> ; DR_TYPE_CHECK :
3002                    <1> ; CHECK IF THE GIVEN DRIVE TYPE IN REGISTER (AL) :
3003                    <1> ; IS SUPPORTED IN BIOS DRIVE TYPE TABLE :
3004                    <1> ; ON ENTRY: :
3005                    <1> ; AL = DRIVE TYPE :
3006                    <1> ; ON EXIT: :
3007                    <1> ; CS = SEGMENT MEDIA/DRIVE PARAMETER TABLE (CODE) :
3008                    <1> ; CY = 0 DRIVE TYPE SUPPORTED :
3009                    <1> ; BX = OFFSET TO MEDIA/DRIVE PARAMETER TABLE :
3010                    <1> ; CY = 1 DRIVE TYPE NOT SUPPORTED :
3011                    <1> ; REGISTERS ALTERED: BX :
3012                    <1> ; -----
3013                    <1> DR_TYPE_CHECK:
3014 00000F03 50          <1>      PUSH     AX
3015 00000F04 51          <1>      PUSH     CX
3016 00000F05 31DB      <1>      XOR      BX, BX          ; BX = INDEX TO DR_TYPE TABLE
3017 00000F07 B90600      <1>      MOV      CX, DR_CNT          ; CX = LOOP COUNT
3018                    <1> TYPE_CHK:
3019 00000F0A 8AA7[5B0B] <1>      MOV      AH, [DR_TYPE+BX]          ; GET DRIVE TYPE
3020 00000F0E 38E0      <1>      CMP      AL, AH          ; DRIVE TYPE MATCH?
3021 00000F10 740B      <1>      JE      short DR_TYPE_VALID ; YES, RETURN WITH CARRY RESET
3022 00000F12 83C303      <1>      ADD      BX, 3          ; CHECK NEXT DRIVE TYPE
3023 00000F15 E2F3      <1>      LOOP     TYPE_CHK
3024                    <1> ;
3025 00000F17 BB[AE0B] <1>      mov     bx, MD_TBL6          ; 1.44MB fd parameter table
3026                    <1> ; Default for GET_PARM (11/12/2014)
3027                    <1> ;
3028 00000F1A F9          <1>      STC          ; DRIVE TYPE NOT FOUND IN TABLE
3029 00000F1B EB04      <1>      JMP      SHORT TYPE_RTN
3030                    <1> DR_TYPE_VALID:
3031 00000F1D 8B9F[5C0B] <1>      MOV      BX, [DR_TYPE+BX+1] ; BX = MEDIA TABLE
3032                    <1> TYPE_RTN:
3033 00000F21 59          <1>      POP      CX
3034 00000F22 58          <1>      POP      AX

```

```

3035 00000F23 C3      <1>          RETn
3036                  <1>
3037                  <1> ;-----
3038                  <1> ; SEND_SPEC                                :
3039                  <1> ;     SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM      :
3040                  <1> ;     THE DRIVE PARAMETER TABLE POINTED BY @DISK_POINTER      :
3041                  <1> ; ON ENTRY: @DISK_POINTER = DRIVE PARAMETER TABLE      :
3042                  <1> ; ON EXIT:  NONE                                          :
3043                  <1> ; REGISTERS ALTERED: CX, DX                                :
3044                  <1> ;-----
3045                  <1> SEND_SPEC:
3046                  <1>     PUSH  AX                      ; SAVE AX
3047                  <1>     MOV   AX, SPECBAC                ; LOAD ERROR ADDRESS
3048                  <1>     PUSH  AX                      ; PUSH NEC_OUT ERROR RETURN
3049                  <1>     MOV   AH, 03H                    ; SPECIFY COMMAND
3050                  <1>     CALL  NEC_OUTPUT                ; OUTPUT THE COMMAND
3051                  <1>     SUB    DL, DL                    ; FIRST SPECIFY BYTE
3052                  <1>     CALL  GET_PARM                   ; GET PARAMETER TO AH
3053                  <1>     CALL  NEC_OUTPUT                ; OUTPUT THE COMMAND
3054                  <1>     MOV    DL, 1                     ; SECOND SPECIFY BYTE
3055                  <1>     CALL  GET_PARM                   ; GET PARAMETER TO AH
3056                  <1>     CALL  NEC_OUTPUT                ; OUTPUT THE COMMAND
3057                  <1>     POP    AX                      ; POP ERROR RETURN
3058                  <1> SPECBAC:
3059                  <1>     POP    AX                      ; RESTORE ORIGINAL AX VALUE
3060                  <1>     RETn
3061                  <1>
3062                  <1> ;-----
3063                  <1> ; SEND_SPEC_MD                                :
3064                  <1> ;     SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM      :
3065                  <1> ;     THE MEDIA/DRIVE PARAMETER TABLE POINTED BY (CS:BX)      :
3066                  <1> ; ON ENTRY: CS:BX = MEDIA/DRIVE PARAMETER TABLE      :
3067                  <1> ; ON EXIT:  NONE                                          :
3068                  <1> ; REGISTERS ALTERED: AX                                :
3069                  <1> ;-----
3070                  <1> SEND_SPEC_MD:
3071                  <1>     PUSH  AX                      ; SAVE RATE DATA
3072                  <1>     MOV   AX, SPEC_ESBAC                ; LOAD ERROR ADDRESS
3073                  <1>     PUSH  AX                      ; PUSH NEC_OUT ERROR RETURN
3074                  <1>     MOV   AH, 03H                    ; SPECIFY COMMAND
3075                  <1>     CALL  NEC_OUTPUT                ; OUTPUT THE COMMAND
3076                  <1>     MOV    AH, [BX+MD.SPEC1]          ; GET 1ST SPECIFY BYTE
3077                  <1>     CALL  NEC_OUTPUT                ; OUTPUT THE COMMAND
3078                  <1>     MOV    AH, [BX+MD.SPEC2]          ; GET SECOND SPECIFY BYTE
3079                  <1>     CALL  NEC_OUTPUT                ; OUTPUT THE COMMAND
3080                  <1>     POP    AX                      ; POP ERROR RETURN
3081                  <1> SPEC_ESBAC:
3082                  <1>     POP    AX                      ; RESTORE ORIGINAL AX VALUE
3083                  <1>     RETn

```

```

3084
3085
3086
3087
3088
3089
3090
3091
3092
3093 00000F59 83FF01
3094 00000F5C 771C
3095 00000F5E 80BD[530B]00
3096 00000F63 7416
3097 00000F65 89F9
3098 00000F67 C0E102
3099 00000F6A A0[7117]
3100 00000F6D D2C8
3101 00000F6F 2407
3102 00000F71 80A5[530B]F8
3103 00000F76 0885[530B]
3104
3105 00000F7A C3
3106
3107 00000F7B E8D706
3108 00000F7E C3
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118 00000F7F 83FF01
3119 00000F82 7773
3120 00000F84 80BD[530B]00
3121 00000F89 746C
3122
3123
3124
3125 00000F8B 89F9
3126 00000F8D C0E102
3127 00000F90 B402
3128 00000F92 D2CC
3129 00000F94 8426[7117]
3130 00000F98 7516
3131
3132

<1>
<1> ;-----
<1> ; XLAT_NEW
<1> ; TRANSLATES DISKETTE STATE LOCATIONS FROM COMPATIBLE
<1> ; MODE TO NEW ARCHITECTURE.
<1> ;
<1> ; ON ENTRY: DI = DRIVE #
<1> ;-----
<1> XLAT_NEW:
<1>     CMP     DI,1                ; VALID DRIVE
<1>     JA      short XN_OUT        ; IF INVALID BACK
<1>     CMP     byte [DSK_STATE+DI], 0      ; NO DRIVE ?
<1>     JZ      short DO_DET        ; IF NO DRIVE ATTEMPT DETERMINE
<1>     MOV     CX,DI                ; CX = DRIVE NUMBER
<1>     SHL     CL,2                ; CL = SHIFT COUNT, A=0, B=4
<1>     MOV     AL, [HF_CNTRL]        ; DRIVE INFORMATION
<1>     ROR     AL,CL                ; TO LOW NIBBLE
<1>     AND     AL,DRV_DET+FMT_CAPA+TRK_CAPA ; KEEP DRIVE BITS
<1>     AND     byte [DSK_STATE+DI], ~(DRV_DET+FMT_CAPA+TRK_CAPA)
<1>     OR      [DSK_STATE+DI], AL      ; UPDATE DRIVE STATE
<1> XN_OUT:
<1>     RETn
<1> DO_DET:
<1>     CALL    DRIVE_DET            ; TRY TO DETERMINE
<1>     RETn
<1>
<1> ;-----
<1> ; XLAT_OLD
<1> ; TRANSLATES DISKETTE STATE LOCATIONS FROM NEW
<1> ; ARCHITECTURE TO COMPATIBLE MODE.
<1> ;
<1> ; ON ENTRY: DI = DRIVE
<1> ;-----
<1> XLAT_OLD:
<1>     CMP     DI,1                ; VALID DRIVE ?
<1>     JA      short X0_OUT        ; IF INVALID BACK
<1>     CMP     byte [DSK_STATE+DI],0      ; NO DRIVE ?
<1>     JZ      short X0_OUT        ; IF NO DRIVE TRANSLATE DONE
<1>
<1> ;----- TEST FOR SAVED DRIVE INFORMATION ALREADY SET
<1>
<1>     MOV     CX,DI                ; CX = DRIVE NUMBER
<1>     SHL     CL,2                ; CL = SHIFT COUNT, A=0, B=4
<1>     MOV     AH,FMT_CAPA          ; LOAD MULTIPLE DATA RATE BIT MASK
<1>     ROR     AH,CL                ; ROTATE BY MASK
<1>     TEST    [HF_CNTRL], AH        ; MULTIPLE-DATA RATE DETERMINED ?
<1>     JNZ     short SAVE_SET        ; IF SO, NO NEED TO RE-SAVE
<1>
<1> ;----- ERASE DRIVE BITS IN @HF_CNTRL FOR THIS DRIVE

```

```

3133
3134 00000F9A B407      <1>
3135 00000F9C D2CC      <1>      MOV    AH,DRV_DET+_FMT_CAPA+TRK_CAPA ; MASK TO KEEP
3136 00000F9E F6D4      <1>      ROR    AH,CL ; FIX MASK TO KEEP
3137 00000FA0 2026[7117] <1>      NOT    AH ; TRANSLATE MASK
3138                      <1>      AND    [HF_CNTRL], AH ; KEEP BITS FROM OTHER DRIVE INTACT
3139                      <1> ;----- ACCESS CURRENT DRIVE BITS AND STORE IN @HF_CNTRL
3140                      <1>
3141 00000FA4 8A85[530B] <1>      MOV    AL,[DSK_STATE+DI] ; ACCESS STATE
3142 00000FA8 2407      <1>      AND    AL,DRV_DET+_FMT_CAPA+TRK_CAPA ; KEEP DRIVE BITS
3143 00000FAA D2C8      <1>      ROR    AL,CL ; FIX FOR THIS DRIVE
3144 00000FAC 0806[7117] <1>      OR     [HF_CNTRL], AL ; UPDATE SAVED DRIVE STATE
3145                      <1>
3146                      <1> ;----- TRANSLATE TO COMPATIBILITY MODE
3147                      <1>
3148                      <1> SAVE_SET:
3149 00000FB0 8AA5[530B] <1>      MOV    AH,[DSK_STATE+DI] ; ACCESS STATE
3150 00000FB4 88E7      <1>      MOV    BH,AH ; TO BH FOR LATER
3151 00000FB6 80E4C0     <1>      AND    AH,RATE_MSK ; KEEP ONLY RATE
3152 00000FB9 80FC00     <1>      CMP    AH,RATE_500 ; RATE 500 ?
3153 00000FBC 7410      <1>      JZ     short CHK_144 ; YES 1.2/1.2 OR 1.44/1.44
3154 00000FBE B001      <1>      MOV    AL,M3D1U ; AL = 360 IN 1.2 UNESTABLISHED
3155 00000FC0 80FC40     <1>      CMP    AH,RATE_300 ; RATE 300 ?
3156 00000FC3 7516      <1>      JNZ    short CHK_250 ; NO, 360/360, 720/720 OR 720/1.44
3157 00000FC5 F6C720     <1>      TEST   BH,DBL_STEP ; CHECK FOR DOUBLE STEP
3158 00000FC8 751D      <1>      JNZ    short TST_DET ; MUST BE 360 IN 1.2
3159                      <1> UNKNO:
3160 00000FCA B007      <1>      MOV    AL,MED_UNK ; NONE OF THE ABOVE
3161 00000FCC EB20      <1>      JMP     SHORT AL_SET ; PROCESS COMPLETE
3162                      <1> CHK_144:
3163 00000FCE E86F04     <1>      CALL   CMOS_TYPE ; RETURN DRIVE TYPE IN (AL)
3164                      <1>      ;JC     short UNKNO ; ERROR, SET 'NONE OF ABOVE'
3165 00000FD1 74F7      <1>      jz     short UNKNO
3166 00000FD3 3C02      <1>      CMP    AL,2 ; 1.2MB DRIVE ?
3167 00000FD5 75F3      <1>      JNE    short UNKNO ; NO, GO SET 'NONE OF ABOVE'
3168 00000FD7 B002      <1>      MOV    AL,M1D1U ; AL = 1.2 IN 1.2 UNESTABLISHED
3169 00000FD9 EB0C      <1>      JMP     SHORT TST_DET
3170                      <1> CHK_250:
3171 00000FDB B000      <1>      MOV    AL,M3D3U ; AL = 360 IN 360 UNESTABLISHED
3172 00000FDD 80FC80     <1>      CMP    AH,RATE_250 ; RATE 250 ?
3173 00000FE0 75E8      <1>      JNZ    short UNKNO ; IF SO FALL IHRU
3174 00000FE2 F6C701     <1>      TEST   BH,TRK_CAPA ; 80 TRACK CAPABILITY ?
3175 00000FE5 75E3      <1>      JNZ    short UNKNO ; IF SO JUMP, FALL THRU TEST DET
3176                      <1> TST_DET:
3177 00000FE7 F6C710     <1>      TEST   BH,MED_DET ; DETERMINED ?
3178 00000FEA 7402      <1>      JZ     short AL_SET ; IF NOT THEN SET
3179 00000FEC 0403      <1>      ADD    AL,3 ; MAKE DETERMINED/ESTABLISHED
3180                      <1> AL_SET:
3181 00000FEE 80A5[530B]F8 <1>      AND    byte [DSK_STATE+DI], ~(DRV_DET+_FMT_CAPA+TRK_CAPA) ; CLEAR DRIVE

```

3182 00000FF3 0885[530B]	<1> OR [DSK_STATE+DI], AL ; REPLACE WITH COMPATIBLE MODE
3183	<1> XO_OUT:
3184 00000FF7 C3	<1> RETn
3185	<1>
3186	<1> ; -----
3187	<1> ; RD_WR_VF
3188	<1> ; COMMON READ, WRITE AND VERIFY:
3189	<1> ; MAIN LOOP FOR STATE RETRIES.
3190	<1> ;
3191	<1> ; ON ENTRY: AH = READ/WRITE/VERIFY NEC PARAMETER
3192	<1> ; AL = READ/WRITE/VERIFY DMA PARAMETER
3193	<1> ;
3194	<1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
3195	<1> ; -----
3196	<1> RD_WR_VF:
3197 00000FF8 50	<1> PUSH AX ; SAVE DMA, NEC PARAMETERS
3198 00000FF9 E85DFF	<1> CALL XLAT_NEW ; TRANSLATE STATE TO PRESENT ARCH.
3199 00000FFC E8B200	<1> CALL SETUP_STATE ; INITIALIZE START AND END RATE
3200 00000FFF 58	<1> POP AX ; RESTORE READ/WRITE/VERIFY
3201	<1> DO_AGAIN:
3202 00001000 50	<1> PUSH AX ; SAVE READ/WRITE/VERIFY PARAMETER
3203 00001001 E82A01	<1> CALL MED_CHANGE ; MEDIA CHANGE AND RESET IF CHANGED
3204 00001004 58	<1> POP AX ; RESTORE READ/WRITE/VERIFY
3205 00001005 0F829900	<1> JC RWV_END ; MEDIA CHANGE ERROR OR TIME-OUT
3206	<1> RWV:
3207 00001009 50	<1> PUSH AX ; SAVE READ/WRITE/VERIFY PARAMETER
3208 0000100A 8AB5[530B]	<1> MOV DH, [DSK_STATE+DI] ; GET RATE STATE OF THIS DRIVE
3209 0000100E 80E6C0	<1> AND DH, RATE_MSK ; KEEP ONLY RATE
3210 00001011 E82C04	<1> CALL CMOS_TYPE ; RETURN DRIVE TYPE IN AL (AL)
3211	<1> ;JC short RWV_ASSUME ; ERROR IN CMOS
3212 00001014 7445	<1> jz short RWV_ASSUME
3213 00001016 3C01	<1> CMP AL,1 ; 40 TRACK DRIVE?
3214 00001018 750B	<1> JNE short RWV_1 ; NO, BYPASS CMOS VALIDITY CHECK
3215 0000101A F685[530B]01	<1> TEST byte [DSK_STATE+DI], TRK_CAPA ; CHECK FOR 40 TRACK DRIVE
3216 0000101F 7411	<1> JZ short RWV_2 ; YES, CMOS IS CORRECT
3217 00001021 B002	<1> MOV AL,2 ; CHANGE TO 1.2M
3218 00001023 EB0D	<1> JMP SHORT RWV_2
3219	<1> RWV_1:
3220 00001025 720B	<1> JB short RWV_2 ; NO DRIVE SPECIFIED, CONTINUE
3221 00001027 F685[530B]01	<1> TEST byte [DSK_STATE+DI], TRK_CAPA ; IS IT REALLY 40 TRACK?
3222 0000102C 7504	<1> JNZ short RWV_2 ; NO, 80 TRACK
3223 0000102E B001	<1> MOV AL,1 ; IT IS 40 TRACK, FIX CMOS VALUE
3224 00001030 EB04	<1> jmp short rww_3
3225	<1> RWV_2:
3226 00001032 08C0	<1> OR AL,AL ; TEST FOR NO DRIVE
3227 00001034 7425	<1> JZ short RWV_ASSUME ; ASSUME TYPE, USE MAX TRACK
3228	<1> rww_3:
3229 00001036 E8CAFE	<1> CALL DR_TYPE_CHECK ; RTN CS:BX = MEDIA/DRIVE PARAM TBL.
3230 00001039 7220	<1> JC short RWV_ASSUME ; TYPE NOT IN TABLE (BAD CMOS)

3231		<1>		
3232		<1>	;-----	SEARCH FOR MEDIA/DRIVE PARAMETER TABLE
3233		<1>		
3234	0000103B 57	<1>	PUSH	DI ; SAVE DRIVE #
3235	0000103C 31DB	<1>	XOR	BX,BX ; BX = INDEX TO DR_TYPE TABLE
3236	0000103E B90600	<1>	MOV	CX,DR_CNT ; CX = LOOP COUNT
3237		<1>	RWV_DR_SEARCH:	
3238	00001041 8AA7[5B0B]	<1>	MOV	AH, [DR_TYPE+BX] ; GET DRIVE TYPE
3239	00001045 80E47F	<1>	AND	AH,BIT70FF ; MASK OUT MSB
3240	00001048 38E0	<1>	CMP	AL,AH ; DRIVE TYPE MATCH?
3241	0000104A 7509	<1>	JNE	short RWV_NXT_MD ; NO, CHECK NEXT DRIVE TYPE
3242		<1>	RWV_DR_FND:	
3243	0000104C 8BBF[5C0B]	<1>	MOV	DI, [DR_TYPE+BX+1] ; DI = MEDIA/DRIVE PARAMETER TABLE
3244		<1>	RWV_MD_SEARCH:	
3245	00001050 3A750C	<1>	CMP	DH, [DI+MD.RATE] ; MATCH?
3246	00001053 7415	<1>	JE	short RWV_MD_FND ; YES, GO GET 1ST SPECIFY BYTE
3247		<1>	RWV_NXT_MD:	
3248	00001055 83C303	<1>	ADD	BX,3 ; CHECK NEXT DRIVE TYPE
3249	00001058 E2E7	<1>	LOOP	RWV_DR_SEARCH
3250	0000105A 5F	<1>	POP	DI ; RESTORE DRIVE #
3251		<1>		
3252		<1>	;-----	ASSUME PRIMARY DRIVE IS INSTALLED AS SHIPPED
3253		<1>		
3254		<1>	RWV_ASSUME:	
3255	0000105B BB[6D0B]	<1>	MOV	BX, MD_TBL1 ; POINT TO 40 TRACK 250 KBS
3256	0000105E F685[530B]01	<1>	TEST	byte [DSK_STATE+DI], TRK_CAPA ; TEST FOR 80 TRACK
3257	00001063 7408	<1>	JZ	short RWV_MD_FND1 ; MUST BE 40 TRACK
3258	00001065 BB[870B]	<1>	MOV	BX, MD_TBL3 ; POINT TO 80 TRACK 500 KBS
3259	00001068 EB03	<1>	JMP	short RWV_MD_FND1 ; GO SPECIFY PARAMETERS
3260		<1>		
3261		<1>	;-----	CS:BX POINTS TO MEDIA/DRIVE PARAMETER TABLE
3262		<1>		
3263		<1>	RWV_MD_FND:	
3264	0000106A 89FB	<1>	MOV	BX,DI ; BX = MEDIA/DRIVE PARAMETER TABLE
3265	0000106C 5F	<1>	POP	DI ; RESTORE DRIVE #
3266		<1>		
3267		<1>	;-----	SEND THE SPECIFY COMMAND TO THE CONTROLLER
3268		<1>		
3269		<1>	RWV_MD_FND1:	
3270	0000106D E8D1FE	<1>	CALL	SEND_SPEC_MD
3271	00001070 E80F01	<1>	CALL	CHK_LASRATE ; ZF=1 ATTEMP RATE IS SAME AS LAST RATE
3272	00001073 7403	<1>	JZ	short RWV_DBL ; YES,SKIP SEND RATE COMMAND
3273	00001075 E8F100	<1>	CALL	SEND_RATE ; SEND DATA RATE TO NEC
3274		<1>	RWV_DBL:	
3275	00001078 53	<1>	PUSH	BX ; SAVE MEDIA/DRIVE PARAM TBL ADDRESS
3276	00001079 E84603	<1>	CALL	SETUP_DBL ; CHECK FOR DOUBLE STEP
3277	0000107C 5B	<1>	POP	BX ; RESTORE ADDRESS
3278	0000107D 721A	<1>	JC	short CHK_RET ; ERROR FROM READ ID, POSSIBLE RETRY
3279	0000107F 58	<1>	POP	AX ; RESTORE NEC, DMA COMMAND

```

3280 00001080 50      <1>      PUSH    AX                ; SAVE NEC COMMAND
3281 00001081 53      <1>      PUSH    BX                ; SAVE MEDIA/DRIVE PARAM TBL ADDRESS
3282 00001082 E80D01   <1>      CALL    DMA_SETUP          ; SET UP THE DMA
3283 00001085 5B      <1>      POP     BX                ; RESTORE ADDRESS
3284 00001086 58      <1>      POP     AX                ; RESTORE NEC COMMAND
3285 00001087 721F    <1>      JC      short RWV_BAC          ; CHECK FOR DMA BOUNDARY ERROR
3286 00001089 50      <1>      PUSH    AX                ; SAVE NEC COMMAND
3287 0000108A 53      <1>      PUSH    BX                ; SAVE MEDIA/DRIVE PARAM TBL ADDRESS
3288 0000108B E8C701   <1>      CALL    NEC_INIT              ; INITIALIZE NEC
3289 0000108E 5B      <1>      POP     BX                ; RESTORE ADDRESS
3290 0000108F 7208    <1>      JC      short CHK_RET          ; ERROR - EXIT
3291 00001091 E8E601   <1>      CALL    RWV_COM              ; OP CODE COMMON TO READ/WRITE/VERIFY
3292 00001094 7203    <1>      JC      short CHK_RET          ; ERROR - EXIT
3293 00001096 E81602   <1>      CALL    NEC_TERM              ; TERMINATE, GET STATUS, ETC.
3294                                <1> CHK_RET:
3295 00001099 E89D02   <1>      CALL    RETRY              ; CHECK FOR, SETUP RETRY
3296 0000109C 58      <1>      POP     AX                ; RESTORE READ/WRITE/VERIFY PARAMETER
3297 0000109D 7303    <1>      JNC    short RWV_END          ; CY = 0 NO RETRY
3298 0000109F E95EFF    <1>      JMP     DO_AGAIN              ; CY = 1 MEANS RETRY
3299                                <1> RWV_END:
3300 000010A2 E85C02   <1>      CALL    DSTATE              ; ESTABLISH STATE IF SUCCESSFUL
3301 000010A5 E8D302   <1>      CALL    NUM_TRANS            ; AL = NUMBER TRANSFERRED
3302                                <1> RWV_BAC:
3303 000010A8 50      <1>      PUSH    AX                ; SAVE NUMBER TRANSFERRED
3304 000010A9 E8D3FE    <1>      CALL    XLAT_OLD            ; TRANSLATE STATE TO COMPATIBLE MODE
3305 000010AC 58      <1>      POP     AX                ; RESTORE NUMBER TRANSFERRED
3306 000010AD E8F802   <1>      CALL    SETUP_END          ; VARIOUS CLEANUPS
3307 000010B0 C3      <1>      RETn
3308                                <1>
3309                                <1> ;-----
3310                                <1> ; SETUP_STATE:      INITIALIZES START AND END RATES.
3311                                <1> ;-----
3312                                <1> SETUP_STATE:
3313 000010B1 F685[530B]10 <1>      TEST    byte [DSK_STATE+DI], MED_DET ; MEDIA DETERMINED ?
3314 000010B6 7529      <1>      JNZ     short J1C              ; NO STATES IF DETERMINED
3315 000010B8 B84000     <1>      MOV     AX,(RATE_500*256)+RATE_300 ; AH = START RATE, AL = END RATE
3316 000010BB F685[530B]04 <1>      TEST    byte [DSK_STATE+DI], DRV_DET ; DRIVE ?
3317 000010C0 740A      <1>      JZ      short AX_SET          ; DO NOT KNOW DRIVE
3318 000010C2 F685[530B]02 <1>      TEST    byte [DSK_STATE+DI], FMT_CAPA ; MULTI-RATE?
3319 000010C7 7503      <1>      JNZ     short AX_SET          ; JUMP IF YES
3320 000010C9 B88080     <1>      MOV     AX,RATE_250*257          ; START A END RATE 250 FOR 360 DRIVE
3321                                <1> AX_SET:
3322 000010CC 80A5[530B]1F <1>      AND     byte [DSK_STATE+DI], ~(RATE_MSK+DBL_STEP) ; TURN OFF THE RATE
3323 000010D1 08A5[530B]   <1>      OR      [DSK_STATE+DI], AH ; RATE FIRST TO TRY
3324 000010D5 8026[520B]F3 <1>      AND     byte [LAstrate], ~STRT_MSK ; ERASE LAST TO TRY RATE BITS
3325 000010DA C0C804     <1>      ROR     AL,4 ; TO OPERATION LAST RATE LOCATION
3326 000010DD 0806[520B]   <1>      OR      [LAstrate], AL ; LAST RATE
3327                                <1> J1C:
3328 000010E1 C3      <1>      RETn

```

```

3329 <1>
3330 <1> ;-----
3331 <1> ; FMT_INIT: ESTABLISH STATE IF UNESTABLISHED AT FORMAT TIME.
3332 <1> ;-----
3333 <1> FMT_INIT:
3334 000010E2 F685[530B]10 <1> TEST byte [DSK_STATE+DI], MED_DET ; IS MEDIA ESTABLISHED
3335 000010E7 7540 <1> JNZ short F1_OUT ; IF SO RETURN
3336 000010E9 E85403 <1> CALL CMOS_TYPE ; RETURN DRIVE TYPE IN AL
3337 <1> ;JC short CL_DRV ; ERROR IN CMOS ASSUME NO DRIVE
3338 000010EC 743C <1> jz short CL_DRV
3339 000010EE FEC8 <1> DEC AL ; MAKE ZERO ORIGIN
3340 <1> ;JS short CL_DRV ; NO DRIVE IF AL 0
3341 000010F0 8AA5[530B] <1> MOV AH, [DSK_STATE+DI] ; AH = CURRENT STATE
3342 000010F4 80E40F <1> AND AH, ~(MED_DET+DBL_STEP+RATE_MSK) ; CLEAR
3343 000010F7 08C0 <1> OR AL,AL ; CHECK FOR 360
3344 000010F9 7505 <1> JNZ short N_360 ; IF 360 WILL BE 0
3345 000010FB 80CC90 <1> OR AH,MED_DET+RATE_250 ; ESTABLISH MEDIA
3346 000010FE EB25 <1> JMP SHORT SKP_STATE ; SKIP OTHER STATE PROCESSING
3347 <1> N_360:
3348 00001100 FEC8 <1> DEC AL ; 1.2 M DRIVE
3349 00001102 7505 <1> JNZ short N_12 ; JUMP IF NOT
3350 <1> F1_RATE:
3351 00001104 80CC10 <1> OR AH,MED_DET+RATE_500 ; SET FORMAT RATE
3352 00001107 EB1C <1> JMP SHORT SKP_STATE ; SKIP OTHER STATE PROCESSING
3353 <1> N_12:
3354 00001109 FEC8 <1> DEC AL ; CHECK FOR TYPE 3
3355 0000110B 750F <1> JNZ short N_720 ; JUMP IF NOT
3356 0000110D F6C404 <1> TEST AH,DRV_DET ; IS DRIVE DETERMINED
3357 00001110 7410 <1> JZ short ISNT_12 ; TREAT AS NON 1.2 DRIVE
3358 00001112 F6C402 <1> TEST AH,FMT_CAPA ; IS 1.2M
3359 00001115 740B <1> JZ short ISNT_12 ; JUMP IF NOT
3360 00001117 80CC50 <1> OR AH,MED_DET+RATE_300 ; RATE 300
3361 0000111A EB09 <1> JMP SHORT SKP_STATE ; CONTINUE
3362 <1> N_720:
3363 0000111C FEC8 <1> DEC AL ; CHECK FOR TYPE 4
3364 0000111E 750A <1> JNZ short CL_DRV ; NO DRIVE, CMOS BAD
3365 00001120 EBE2 <1> JMP SHORT F1_RATE
3366 <1> ISNT_12:
3367 00001122 80CC90 <1> OR AH,MED_DET+RATE_250 ; MUST BE RATE 250
3368 <1>
3369 <1> SKP_STATE:
3370 00001125 88A5[530B] <1> MOV [DSK_STATE+DI], AH ; STORE AWAY
3371 <1> F1_OUT:
3372 00001129 C3 <1> RETn
3373 <1> CL_DRV:
3374 0000112A 30E4 <1> XOR AH,AH ; CLEAR STATE
3375 0000112C EBF7 <1> JMP SHORT SKP_STATE ; SAVE IT
3376 <1>
3377 <1> ;-----

```

```

3378          <1> ; MED_CHANGE
3379          <1> ;     CHECKS FOR MEDIA CHANGE, RESETS MEDIA CHANGE,
3380          <1> ;     CHECKS MEDIA CHANGE AGAIN.
3381          <1> ;
3382          <1> ; ON EXIT:  CY = 1 MEANS MEDIA CHANGE OR TIMEOUT
3383          <1> ;     @DSKETTE_STATUS = ERROR CODE
3384          <1> ; -----
3385          <1> MED_CHANGE:
3386 0000112E E81A05          <1>     CALL    READ_DSKCHNG      ; READ DISK CHANCE LINE STATE
3387 00001131 7434           <1>     JZ      short MC_OUT      ; BYPASS HANDLING DISK CHANGE LINE
3388 00001133 80A5[530B]EF   <1>     AND     byte [DSK_STATE+DI], ~MED_DET ; CLEAR STATE FOR THIS DRIVE
3389          <1>
3390          <1> ;     THIS SEQUENCE ENSURES WHENEVER A DISKETTE IS CHANGED THAT
3391          <1> ;     ON THE NEXT OPERATION THE REQUIRED MOTOR START UP TIME WILL
3392          <1> ;     BE WAITED. (DRIVE MOTOR MAY GO OFF UPON DOOR OPENING).
3393          <1>
3394 00001138 89F9           <1>     MOV     CX,DI             ; CL = DRIVE 0
3395 0000113A B001           <1>     MOV     AL,1              ; MOTOR ON BIT MASK
3396 0000113C D2E0           <1>     SHL     AL,CL             ; TO APPROPRIATE POSITION
3397 0000113E F6D0           <1>     NOT     AL                ; KEEP ALL BUT MOTOR ON
3398 00001140 FA            <1>     CLI                     ; NO INTERRUPTS
3399 00001141 2006[480B]     <1>     AND     [MOTOR_STATUS], AL ; TURN MOTOR OFF INDICATOR
3400 00001145 FB            <1>     STI                     ; INTERRUPTS ENABLED
3401 00001146 E82F03       <1>     CALL    MOTOR_ON          ; TURN MOTOR ON
3402          <1>
3403          <1> ;----- THIS SEQUENCE OF SEEKS IS USED TO RESET DISKETTE CHANGE SIGNAL
3404          <1>
3405 00001149 E8FAFA       <1>     CALL    DSK_RESET          ; RESET NEC
3406 0000114C B501         <1>     MOV     CH,01H           ; MOVE TO CYLINDER 1
3407 0000114E E8EA03       <1>     CALL    SEEK              ; ISSUE SEEK
3408 00001151 30ED         <1>     XOR     CH,CH            ; MOVE TO CYLINDER 0
3409 00001153 E8E503       <1>     CALL    SEEK              ; ISSUE SEEK
3410 00001156 C606[4A0B]06 <1>     MOV     byte [DSKETTE_STATUS], MEDIA_CHANGE ; STORE IN STATUS
3411          <1> OK1:
3412 0000115B E8ED04       <1>     CALL    READ_DSKCHNG      ; CHECK MEDIA CHANGED AGAIN
3413 0000115E 7405         <1>     JZ      short OK2          ; IF ACTIVE, NO DISKETTE, TIMEOUT
3414          <1> OK4:
3415 00001160 C606[4A0B]80 <1>     MOV     byte [DSKETTE_STATUS], TIME_OUT ; TIMEOUT IF DRIVE EMPTY
3416          <1> OK2:
3417 00001165 F9           <1>     STC                     ; MEDIA CHANGED, SET CY
3418 00001166 C3           <1>     RETn
3419          <1> MC_OUT:
3420 00001167 F8           <1>     CLC                     ; NO MEDIA CHANGED, CLEAR CY
3421 00001168 C3           <1>     RETn
3422          <1>
3423          <1> ;-----
3424          <1> ; SEND_RATE
3425          <1> ;     SENDS DATA RATE COMMAND TO NEC
3426          <1> ; ON ENTRY:  DI = DRIVE #

```

```

3427      <1> ; ON EXIT:  NONE
3428      <1> ; REGISTERS ALTERED: DX
3429      <1> ;-----
3430      <1> SEND_RATE:
3431      <1>     PUSH  AX                ; SAVE REG.
3432      <1>     AND   byte [LAstrate], ~SEND_MSK ; ELSE CLEAR LAST RATE ATTEMPTED
3433      <1>     MOV   AL, [DSK_STATE+DI] ; GET RATE STATE OF THIS DRIVE
3434      <1>     AND   AL, SEND_MSK       ; KEEP ONLY RATE BITS
3435      <1>     OR    [LAstrate], AL      ; SAVE NEW RATE FOR NEXT CHECK
3436      <1>     ROL   AL, 2              ; MOVE TO BIT OUTPUT POSITIONS
3437      <1>     MOV   DX, 03F7H         ; OUTPUT NEW DATA RATE
3438      <1>     OUT   DX, AL
3439      <1>     POP   AX                ; RESTORE REG.
3440      <1>     RETn
3441      <1>
3442      <1> ;-----
3443      <1> ; CHK_LAstrate
3444      <1> ;     CHECK PREVIOUS DATE RATE SNT TO THE CONTROLLER.
3445      <1> ; ON ENTRY:
3446      <1> ;     DI = DRIVE #
3447      <1> ; ON EXIT:
3448      <1> ;     ZF = 1 DATA RATE IS THE SAME AS THE LAST RATE SENT TO NEC
3449      <1> ;     ZF = 0 DATA RATE IS DIFFERENT FROM LAST RATE
3450      <1> ; REGISTERS ALTERED: DX
3451      <1> ;-----
3452      <1> CHK_LAstrate:
3453      <1>     PUSH  AX                ; SAVE REG
3454      <1>     AND   AH, [LAstrate]    ; GET LAST DATA RATE SELECTED
3455      <1>     MOV   AL, [DSK_STATE+DI] ; GET RATE STATE OF THIS DRIVE
3456      <1>     AND   AX, SEND_MSK*257  ; KEEP ONLY RATE BITS OF BOTH
3457      <1>     CMP   AL, AH            ; COMPARE TO PREVIOUSLY TRIED
3458      <1>     ; ZF = 1 RATE IS THE SAME
3459      <1>     POP   AX                ; RESTORE REG.
3460      <1>     RETn
3461      <1>
3462      <1> ;-----
3463      <1> ; DMA_SETUP
3464      <1> ;     THIS ROUTINE SETS UP THE DMA FOR READ/WRITE/VERIFY OPERATIONS.
3465      <1> ;
3466      <1> ; ON ENTRY:  AL = DMA COMMAND
3467      <1> ;
3468      <1> ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
3469      <1> ;-----
3470      <1>
3471      <1> ; SI = Head #, # of Sectors or DASD Type
3472      <1>
3473      <1> ; 16/12/2014 (IODELAY)
3474      <1>
3475      <1> DMA_SETUP:

```

3476 00001192 FA	<1>	CLI	; DISABLE INTERRUPTS DURING DMA SET-UP
3477 00001193 E60C	<1>	OUT DMA+12,AL	; SET THE FIRST/LA5T F/F
3478	<1>	;JMP \$+2	; WAIT FOR I/O
3479	<1>	IODELAY	
3480 00001195 EB00	<2>	jmp short \$+2	
3481 00001197 EB00	<2>	jmp short \$+2	
3482 00001199 E60B	<1>	OUT DMA+11,AL	; OUTPUT THE MODE BYTE
3483	<1>	;SIODELAY	
3484 0000119B 3C42	<1>	CMP AL, 42H	; DMA VERIFY COMMAND
3485 0000119D 7504	<1>	JNE short NOT_VERF	; NO
3486 0000119F 31C0	<1>	XOR AX, AX	; START ADDRESS
3487 000011A1 EB10	<1>	JMP SHORT J33	
3488	<1>	NOT_VERF:	
3489 000011A3 8CC0	<1>	MOV AX,ES	; GET THE ES VALUE
3490 000011A5 C1C004	<1>	ROL AX,4	; ROTATE LEFT
3491 000011A8 88C5	<1>	MOV CH,AL	; GET HIGHEST NIBBLE OF ES TO CH
3492 000011AA 24F0	<1>	AND AL,11110000B	; ZERO THE LOW NIBBLE FROM SEGMENT
3493 000011AC 034602	<1>	ADD AX,[BP+2]	; TEST FOR CARRY FROM ADDITION
3494 000011AF 7302	<1>	JNC short J33	
3495 000011B1 FEC5	<1>	INC CH	; CARRY MEANS HIGH 4 BITS MUST BE INC
3496	<1>	J33:	
3497 000011B3 50	<1>	PUSH AX	; SAVE START ADDRESS
3498 000011B4 E604	<1>	OUT DMA+4,AL	; OUTPUT LOW ADDRESS
3499	<1>	;JMP \$+2	; WAIT FOR I/O
3500	<1>	IODELAY	
3501 000011B6 EB00	<2>	jmp short \$+2	
3502 000011B8 EB00	<2>	jmp short \$+2	
3503 000011BA 88E0	<1>	MOV AL,AH	
3504 000011BC E604	<1>	OUT DMA+4,AL	; OUTPUT HIGH ADDRESS
3505 000011BE 88E8	<1>	MOV AL,CH	; GET HIGH 4 BITS
3506	<1>	;JMP \$+2	; I/O WAIT STATE
3507	<1>	IODELAY	
3508 000011C0 EB00	<2>	jmp short \$+2	
3509 000011C2 EB00	<2>	jmp short \$+2	
3510 000011C4 240F	<1>	AND AL,00001111B	
3511 000011C6 E681	<1>	OUT 081H,AL	; OUTPUT HIGH 4 BITS TO PAGE REGISTER
3512	<1>	;SIODELAY	
3513	<1>		
3514	<1>	;----- DETERMINE COUNT	
3515	<1>		
3516 000011C8 89F0	<1>	MOV AX, SI	; AL = # OF SECTORS
3517 000011CA 86C4	<1>	XCHG AL, AH	; AH = # OF SECTORS
3518 000011CC 28C0	<1>	SUB AL, AL	; AL = 0, AX = # SECTORS * 256
3519 000011CE D1E8	<1>	SHR AX, 1	; AX = # SECTORS * 128
3520 000011D0 50	<1>	PUSH AX	; SAVE # OF SECTORS * 128
3521 000011D1 B203	<1>	MOV DL, 3	; GET BYTES/SECTOR PARAMETER
3522 000011D3 E87102	<1>	CALL GET_PARM	; "
3523 000011D6 88E1	<1>	MOV CL,AH	; SHIFT COUNT (0=128, 1=256, 2=512 ETC)
3524 000011D8 58	<1>	POP AX	; AX = # SECTORS * 128

3525 000011D9 D3E0	<1>	SHL AX, CL	; SHIFT BY PARAMETER VALUE
3526 000011DB 48	<1>	DEC AX	; -1 FOR DMA VALUE
3527 000011DC 50	<1>	PUSH AX	; SAVE COUNT VALUE
3528 000011DD E605	<1>	OUT DMA+5, AL	; LOW BYTE OF COUNT
3529	<1>	; JMP \$+2	; WAIT FOR I/O
3530	<1>	IODELAY	
3531 000011DF EB00	<2>	jmp short \$+2	
3532 000011E1 EB00	<2>	jmp short \$+2	
3533 000011E3 88E0	<1>	MOV AL, AH	
3534 000011E5 E605	<1>	OUT DMA+5, AL	; HIGH BYTE OF COUNT
3535	<1>	; IODELAY	
3536 000011E7 FB	<1>	STI	; RE-ENABLE INTERRUPTS
3537 000011E8 59	<1>	POP CX	; RECOVER COUNT VALUE
3538 000011E9 58	<1>	POP AX	; RECOVER ADDRESS VALUE
3539 000011EA 01C8	<1>	ADD AX, CX	; ADD, TEST FOR 64K OVERFLOW
3540 000011EC B002	<1>	MOV AL, 2	; MODE FOR 8237
3541	<1>	; JMP \$+2	; WAIT FOR I/O
3542	<1>	SIODELAY	
3543 000011EE EB00	<2>	jmp short \$+2	
3544 000011F0 E60A	<1>	OUT DMA+10, AL	; INITIALIZE THE DISKETTE CHANNEL
3545 000011F2 7305	<1>	JNC short NO_BAD	; CHECK FOR ERROR
3546 000011F4 C606[4A0B]09	<1>	MOV byte [DSKETTE_STATUS], DMA_BOUNDARY	; SET ERROR
3547	<1>	NO_BAD:	
3548 000011F9 C3	<1>	RETn	; CY SET BY ABOVE IF ERROR
3549	<1>		
3550	<1>	; -----	
3551	<1>	; FMTDMA_SET	
3552	<1>	; THIS ROUTINE SETS UP THE DMA CONTROLLER FOR A FORMAT OPERATION.	
3553	<1>	;	
3554	<1>	; ON ENTRY: NOTHING REQUIRED	
3555	<1>	;	
3556	<1>	; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION	
3557	<1>	; -----	
3558	<1>		
3559	<1>	FMTDMA_SET:	
3560 000011FA B04A	<1>	MOV AL, 04AH	; WILL WRITE TO THE DISKETTE
3561 000011FC FA	<1>	CLI	; DISABLE INTERRUPTS DURING DMA SET-UP
3562 000011FD E60C	<1>	OUT DMA+12, AL	; SET THE FIRST/LA5T F/F
3563	<1>	; JMP \$+2	; WAIT FOR I/O
3564	<1>	IODELAY	
3565 000011FF EB00	<2>	jmp short \$+2	
3566 00001201 EB00	<2>	jmp short \$+2	
3567 00001203 E60B	<1>	OUT DMA+11, AL	; OUTPUT THE MODE BYTE
3568	<1>		
3569 00001205 8CC0	<1>	MOV AX, ES	; GET THE ES VALUE
3570 00001207 C1C004	<1>	ROL AX, 4	; ROTATE LEFT
3571 0000120A 88C5	<1>	MOV CH, AL	; GET HIGHEST NIBBLE OF ES TO CH
3572 0000120C 24F0	<1>	AND AL, 11110000B	; ZERO THE LOW NIBBLE FROM SEGMENT
3573 0000120E 034602	<1>	ADD AX, [BP+2]	; TEST FOR CARRY FROM ADDITION

3574 00001211 7302	<1>	JNC	short J33A	
3575 00001213 FEC5	<1>	INC	CH	; CARRY MEANS HIGH 4 BITS MUST BE INC
3576	<1>	J33A:		
3577 00001215 50	<1>	PUSH	AX	; SAVE START ADDRESS
3578 00001216 E604	<1>	OUT	DMA+4,AL	; OUTPUT LOW ADDRESS
3579	<1>	;JMP	\$+2	; WAIT FOR I/O
3580	<1>	IODELAY		
3581 00001218 EB00	<2>	jmp	short \$+2	
3582 0000121A EB00	<2>	jmp	short \$+2	
3583 0000121C 88E0	<1>	MOV	AL,AH	
3584 0000121E E604	<1>	OUT	DMA+4,AL	; OUTPUT HIGH ADDRESS
3585 00001220 88E8	<1>	MOV	AL,CH	; GET HIGH 4 BITS
3586	<1>	;JMP	\$+2	; I/O WAIT STATE
3587	<1>	IODELAY		
3588 00001222 EB00	<2>	jmp	short \$+2	
3589 00001224 EB00	<2>	jmp	short \$+2	
3590 00001226 240F	<1>	AND	AL,00001111B	
3591 00001228 E681	<1>	OUT	081H,AL	; OUTPUT HIGH 4 BITS TO PAGE REGISTER
3592	<1>			
3593	<1>	;-----	DETERMINE COUNT	
3594	<1>			
3595 0000122A B204	<1>	MOV	DL, 4	; SECTORS/TRACK VALUE IN PARM TABLE
3596 0000122C E81802	<1>	CALL	GET_PARM	; "
3597 0000122F 86C4	<1>	XCHG	AL, AH	; AL = SECTORS/TRACK VALUE
3598 00001231 28E4	<1>	SUB	AH, AH	; AX = SECTORS/TRACK VALUE
3599 00001233 C1E002	<1>	SHL	AX, 2	; AX = SEC/TRK * 4 (OFFSET C,H,R,N)
3600 00001236 48	<1>	DEC	AX	; -1 FOR DMA VALUE
3601 00001237 50	<1>	PUSH	AX	; SAVE # OF BYTES TO BE TRANSFERED
3602 00001238 E605	<1>	OUT	DMA+5,AL	; LOW BYTE OF COUNT
3603	<1>	;JMP	\$+2	; WAIT FOR I/O
3604	<1>	IODELAY		
3605 0000123A EB00	<2>	jmp	short \$+2	
3606 0000123C EB00	<2>	jmp	short \$+2	
3607 0000123E 88E0	<1>	MOV	AL, AH	
3608 00001240 E605	<1>	OUT	DMA+5,AL	; HIGH BYTE OF COUNT
3609 00001242 FB	<1>	STI		; RE-ENABLE INTERRUPTS
3610 00001243 59	<1>	POP	CX	; RECOVER COUNT VALUE
3611 00001244 58	<1>	POP	AX	; RECOVER ADDRESS VALUE
3612 00001245 01C8	<1>	ADD	AX, CX	; ADD, TEST FOR 64K OVERFLOW
3613 00001247 B002	<1>	MOV	AL, 2	; MODE FOR 8237
3614	<1>	;JMP	\$+2	; WAIT FOR I/O
3615	<1>	SIODELAY		
3616 00001249 EB00	<2>	jmp	short \$+2	
3617 0000124B E60A	<1>	OUT	DMA+10, AL	; INITIALIZE THE DISKETTE CHANNEL
3618 0000124D 7305	<1>	JNC	short FMTDMA_OK	; CHECK FOR ERROR
3619 0000124F C606[4A0B]09	<1>	MOV	byte [DSKETTE_STATUS], DMA_BOUNDARY	; SET ERROR
3620	<1>	FMTDMA_OK:		
3621 00001254 C3	<1>	RETn		; CY SET BY ABOVE IF ERROR
3622	<1>			

```

3623                                     <1> ;-----
3624                                     <1> ; NEC_INIT
3625                                     <1> ;   THIS ROUTINE SEEKS TO THE REQUESTED TRACK AND INITIALIZES
3626                                     <1> ;   THE NEC FOR THE READ/WRITE/VERIFY/FORMAT OPERATION.
3627                                     <1> ;
3628                                     <1> ; ON ENTRY:  AH = NEC COMMAND TO BE PERFORMED
3629                                     <1> ;
3630                                     <1> ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
3631                                     <1> ;-----
3632                                     <1> NEC_INIT:
3633 00001255 50                         <1>     PUSH  AX                ; SAVE NEC COMMAND
3634 00001256 E81F02                     <1>     CALL  MOTOR_ON          ; TURN MOTOR ON FOR SPECIFIC DRIVE
3635                                     <1> ;
3636                                     <1> ;----- DO THE SEEK OPERATION
3637                                     <1> ;
3638 00001259 8A6E01                     <1>     MOV   CH, [BP+1]          ; CH = TRACK #
3639 0000125C E8DC02                     <1>     CALL  SEEK                ; MOVE TO CORRECT TRACK
3640 0000125F 58                         <1>     POP   AX                ; RECOVER COMMAND
3641 00001260 7217                       <1>     JC    short ER_1          ; ERROR ON SEEK
3642 00001262 BB[7912]                   <1>     MOV   BX, ER_1          ; LOAD ERROR ADDRESS
3643 00001265 53                         <1>     PUSH  BX                ; PUSH NEC_OUT ERROR RETURN
3644                                     <1> ;
3645                                     <1> ;----- SEND OUT THE PARAMETERS TO THE CONTROLLER
3646                                     <1> ;
3647 00001266 E8AB02                     <1>     CALL  NEC_OUTPUT          ; OUTPUT THE OPERATION COMMAND
3648 00001269 89F0                       <1>     MOV   AX, SI            ; AH = HEAD #
3649 0000126B 89FB                       <1>     MOV   BX, DI            ; BL = DRIVE #
3650 0000126D C0E402                     <1>     SAL   AH, 2            ; MOVE IT TO BIT 2
3651 00001270 80E404                     <1>     AND   AH, 00000100B      ; ISOLATE THAT BIT
3652 00001273 08DC                       <1>     OR    AH, BL            ; OR IN THE DRIVE NUMBER
3653 00001275 E89C02                     <1>     CALL  NEC_OUTPUT          ; FALL THRU CY SET IF ERROR
3654 00001278 5B                         <1>     POP   BX                ; THROW AWAY ERROR RETURN
3655                                     <1> ER_1:
3656 00001279 C3                         <1>     RETn
3657                                     <1> ;
3658                                     <1> ;-----
3659                                     <1> ; RWV_COM
3660                                     <1> ;   THIS ROUTINE SENDS PARAMETERS TO THE NEC SPECIFIC TO THE
3661                                     <1> ;   READ/WRITE/VERIFY OPERATIONS.
3662                                     <1> ;
3663                                     <1> ; ON ENTRY:  CS:BX = ADDRESS OF MEDIA/DRIVE PARAMETER TABLE
3664                                     <1> ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
3665                                     <1> ;-----
3666                                     <1> RWV_COM:
3667 0000127A B8[AE12]                   <1>     MOV   AX, ER_2          ; LOAD ERROR ADDRESS
3668 0000127D 50                         <1>     PUSH  AX                ; PUSH NEC_OUT ERROR RETURN
3669 0000127E 8A6601                     <1>     MOV   AH, [BP+1]          ; OUTPUT TRACK #
3670 00001281 E89002                     <1>     CALL  NEC_OUTPUT
3671 00001284 89F0                       <1>     MOV   AX, SI            ; OUTPUT HEAD #

```

3672	00001286	E88B02	<1>	CALL	NEC_OUTPUT	
3673	00001289	8A6600	<1>	MOV	AH, [BP]	; OUTPUT SECTOR #
3674	0000128C	E88502	<1>	CALL	NEC_OUTPUT	
3675	0000128F	B203	<1>	MOV	DL, 3	; BYTES/SECTOR PARAMETER FROM BLOCK
3676	00001291	E8B301	<1>	CALL	GET_PARM	; ... TO THE NEC
3677	00001294	E87D02	<1>	CALL	NEC_OUTPUT	; OUTPUT TO CONTROLLER
3678	00001297	B204	<1>	MOV	DL, 4	; EOT PARAMETER FROM BLOCK
3679	00001299	E8AB01	<1>	CALL	GET_PARM	; ... TO THE NEC
3680	0000129C	E87502	<1>	CALL	NEC_OUTPUT	; OUTPUT TO CONTROLLER
3681	0000129F	8A6705	<1>	MOV	AH, [BX+MD.GAP]	; GET GAP LENGTH
3682			<1>	_R15:		
3683	000012A2	E86F02	<1>	CALL	NEC_OUTPUT	
3684	000012A5	B206	<1>	MOV	DL, 6	; DTL PARAMETER FROM BLOCK
3685	000012A7	E89D01	<1>	CALL	GET_PARM	; TO THE NEC
3686	000012AA	E86702	<1>	CALL	NEC_OUTPUT	; OUTPUT TO CONTROLLER
3687	000012AD	58	<1>	POP	AX	; THROW AWAY ERROR EXIT
3688			<1>	ER_2:		
3689	000012AE	C3	<1>	RETn		
3690			<1>			
3691			<1>			
3692			<1>			
3693			<1>			
3694			<1>			
3695			<1>			
3696			<1>			
3697			<1>			
3698			<1>			
3699			<1>			
3700			<1>			
3701			<1>			
3702	000012AF	56	<1>	PUSH	SI	; SAVE HEAD #, # OF SECTORS
3703	000012B0	E82503	<1>	CALL	WAIT_INT	; WAIT FOR THE INTERRUPT
3704	000012B3	9C	<1>	PUSHF		
3705	000012B4	E84403	<1>	CALL	RESULTS	; GET THE NEC STATUS
3706	000012B7	7245	<1>	JC	short SET_END_POP	
3707	000012B9	9D	<1>	POPF		
3708	000012BA	723A	<1>	JC	short SET_END	; LOOK FOR ERROR
3709			<1>			
3710			<1>			
3711			<1>			
3712	000012BC	FC	<1>	CLD		; SET THE CORRECT DIRECTION
3713	000012BD	BE[4B0B]	<1>	MOV	SI, NEC_STATUS	; POINT TO STATUS FIELD
3714	000012C0	AC	<1>	lodsb		; GET ST0
3715	000012C1	24C0	<1>	AND	AL, 11000000B	; TEST FOR NORMAL TERMINATION
3716	000012C3	7431	<1>	JZ	short SET_END	
3717	000012C5	3C40	<1>	CMP	AL, 01000000B	; TEST FOR ABNORMAL TERMINATION
3718	000012C7	7527	<1>	JNZ	short J18	; NOT ABNORMAL, BAD NEC
3719			<1>			
3720			<1>			

```

3721 <1>
3722 000012C9 AC <1> lodsb ; GET ST1
3723 000012CA D0E0 <1> SAL AL,1 ; TEST FOR EDT FOUND
3724 000012CC B404 <1> MOV AH,RECORD_NOT_FND
3725 000012CE 7222 <1> JC short J19
3726 000012D0 C0E002 <1> SAL AL,2
3727 000012D3 B410 <1> MOV AH,BAD_CRC
3728 000012D5 721B <1> JC short J19
3729 000012D7 D0E0 <1> SAL AL,1 ; TEST FOR DMA OVERRUN
3730 000012D9 B408 <1> MOV AH,BAD_DMA
3731 000012DB 7215 <1> JC short J19
3732 000012DD C0E002 <1> SAL AL,2 ; TEST FOR RECORD NOT FOUND
3733 000012E0 B404 <1> MOV AH,RECORD_NOT_FND
3734 000012E2 720E <1> JC short J19
3735 000012E4 D0E0 <1> SAL AL,1
3736 000012E6 B403 <1> MOV AH,WRITE_PROTECT ; TEST FOR WRITE_PROTECT
3737 000012E8 7208 <1> JC short J19
3738 000012EA D0E0 <1> SAL AL,1 ; TEST MISSING ADDRESS MARK
3739 000012EC B402 <1> MOV AH,BAD_ADDR_MARK
3740 000012EE 7202 <1> JC short J19
3741 <1>
3742 <1> ;----- NEC MUST HAVE FAILED
3743 <1> J18:
3744 000012F0 B420 <1> MOV AH,BAD_NEC
3745 <1> J19:
3746 000012F2 0826[4A0B] <1> OR [DSKETTE_STATUS], AH
3747 <1> SET_END:
3748 000012F6 803E[4A0B]01 <1> CMP byte [DSKETTE_STATUS], 1 ; SET ERROR CONDITION
3749 000012FB F5 <1> CMC
3750 000012FC 5E <1> POP SI
3751 000012FD C3 <1> RETn ; RESTORE HEAD #, # OF SECTORS
3752 <1>
3753 <1> SET_END_POP:
3754 000012FE 9D <1> POPF
3755 000012FF EBF5 <1> JMP SHORT SET_END
3756 <1>
3757 <1> ;-----
3758 <1> ; DSTATE: ESTABLISH STATE UPON SUCCESSFUL OPERATION.
3759 <1> ;-----
3760 <1> DSTATE:
3761 00001301 803E[4A0B]00 <1> CMP byte [DSKETTE_STATUS],0 ; CHECK FOR ERROR
3762 00001306 7530 <1> JNZ short SETBAC ; IF ERROR JUMP
3763 00001308 808D[530B]10 <1> OR byte [DSK_STATE+DI],MED_DET ; NO ERROR, MARK MEDIA AS DETERMINED
3764 0000130D F685[530B]04 <1> TEST byte [DSK_STATE+DI],DRV_DET ; DRIVE DETERMINED ?
3765 00001312 7524 <1> JNZ short SETBAC ; IF DETERMINED NO TRY TO DETERMINE
3766 00001314 8A85[530B] <1> MOV AL,[DSK_STATE+DI] ; LOAD STATE
3767 00001318 24C0 <1> AND AL,RATE_MSK ; KEEP ONLY RATE
3768 0000131A 3C80 <1> CMP AL,RATE_250 ; RATE 250 ?
3769 0000131C 7515 <1> JNE short M_12 ; NO, MUST BE 1.2M OR 1.44M DRIVE

```

```

3770                                     <1>
3771                                     <1> ;----- CHECK IF IT IS 1.44M
3772                                     <1>
3773 0000131E E81F01                 <1>      CALL    CMOS_TYPE          ; RETURN DRIVE TYPE IN (AL)
3774                                     <1>      ;JC      short M_12          ; CMOS BAD
3775 00001321 7410                     <1>      jz       short M_12
3776 00001323 3C04                     <1>      CMP     AL, 4              ; 1.44MB DRIVE ?
3777 00001325 740C                     <1>      JE      short M_12          ; YES
3778                                     <1> M_720:
3779 00001327 80A5[530B]FD             <1>      AND     byte [DSK_STATE+DI], ~FMT_CAPA ; TURN OFF FORMAT CAPABILITY
3780 0000132C 808D[530B]04             <1>      OR      byte [DSK_STATE+DI], DRV_DET ; MARK DRIVE DETERMINED
3781 00001331 EB05                     <1>      JMP     SHORT SETBAC        ; BACK
3782                                     <1> M_12:
3783 00001333 808D[530B]06             <1>      OR      byte [DSK_STATE+DI], DRV_DET+FMT_CAPA
3784                                     <1>                                     ; TURN ON DETERMINED & FMT CAPA
3785                                     <1> SETBAC:
3786 00001338 C3                       <1>      RETn
3787                                     <1>
3788                                     <1> ;-----
3789                                     <1> ; RETRY
3790                                     <1> ; DETERMINES WHETHER A RETRY IS NECESSARY.
3791                                     <1> ; IF RETRY IS REQUIRED THEN STATE INFORMATION IS UPDATED FOR RETRY.
3792                                     <1> ;
3793                                     <1> ; ON EXIT: CY = 1 FOR RETRY, CY = 0 FOR NO RETRY
3794                                     <1> ;-----
3795                                     <1> RETRY:
3796 00001339 803E[4A0B]00             <1>      CMP     byte [DSKETTE_STATUS],0 ; GET STATUS OF OPERATION
3797 0000133E 7439                     <1>      JZ      short NO_RETRY ; SUCCESSFUL OPERATION
3798 00001340 803E[4A0B]80             <1>      CMP     byte [DSKETTE_STATUS],TIME_OUT ; IF TIME OUT NO RETRY
3799 00001345 7432                     <1>      JZ      short NO_RETRY
3800 00001347 8AA5[530B]             <1>      MOV     AH,[DSK_STATE+DI] ; GET MEDIA STATE OF DRIVE
3801 0000134B F6C410                 <1>      TEST    AH,MED_DET ; ESTABLISHED/DETERMINED ?
3802 0000134E 7529                     <1>      JNZ     short NO_RETRY ; IF ESTABLISHED STATE THEN TRUE ERROR
3803 00001350 80E4C0                 <1>      AND     AH,RATE_MSK ; ISOLATE RATE
3804 00001353 8A2E[520B]             <1>      MOV     CH,[LAstrate] ; GET START OPERATION STATE
3805 00001357 C0C504                 <1>      ROL     CH,4 ; TO CORRESPONDING BITS
3806 0000135A 80E5C0                 <1>      AND     CH,RATE_MSK ; ISOLATE RATE BITS
3807 0000135D 38E5                     <1>      CMP     CH,AH ; ALL RATES TRIED
3808 0000135F 7418                     <1>      JE      short NO_RETRY ; IF YES, THEN TRUE ERROR
3809                                     <1>
3810                                     <1> ; SETUP STATE INDICATOR FOR RETRY ATTEMPT TO NEXT RATE
3811                                     <1> ; 00000000B (500) -> 10000000B (250)
3812                                     <1> ; 10000000B (250) -> 01000000B (300)
3813                                     <1> ; 01000000B (300) -> 00000000B (500)
3814                                     <1>
3815 00001361 80FC01                 <1>      CMP     AH,RATE_500+1 ; SET CY FOR RATE 500
3816 00001364 D0DC                     <1>      RCR     AH,1 ; TO NEXT STATE
3817 00001366 80E4C0                 <1>      AND     AH,RATE_MSK ; KEEP ONLY RATE BITS
3818 00001369 80A5[530B]1F             <1>      AND     byte [DSK_STATE+DI], ~(RATE_MSK+DBL_STEP)

```

```

3819                                <1>                                ; RATE, DBL STEP OFF
3820 0000136E 08A5[530B]          <1>      OR      [DSK_STATE+DI],AH      ; TURN ON NEW RATE
3821 00001372 C606[4A0B]00        <1>      MOV      byte [DSKETTE_STATUS],0      ; RESET STATUS FOR RETRY
3822 00001377 F9                  <1>      STC                          ; SET CARRY FOR RETRY
3823 00001378 C3                  <1>      RETn                          ; RETRY RETURN
3824                                <1>
3825                                <1> NO_RETRY:
3826 00001379 F8                  <1>      CLC                          ; CLEAR CARRY NO RETRY
3827 0000137A C3                  <1>      RETn                          ; NO RETRY RETURN
3828                                <1>
3829                                <1> ;-----
3830                                <1> ; NUM_TRANS
3831                                <1> ;     THIS ROUTINE CALCULATES THE NUMBER OF SECTORS THAT WERE
3832                                <1> ;     ACTUALLY TRANSFERRED TO/FROM THE DISKETTE.
3833                                <1> ;
3834                                <1> ; ON ENTRY: [BP+1] = TRACK
3835                                <1> ;             SI-HI = HEAD
3836                                <1> ;             [BP] = START SECTOR
3837                                <1> ;
3838                                <1> ; ON EXIT:  AL = NUMBER ACTUALLY TRANSFERRED
3839                                <1> ;-----
3840                                <1> NUM_TRANS:
3841 0000137B 30C0                  <1>      XOR      AL,AL      ; CLEAR FOR ERROR
3842 0000137D 803E[4A0B]00        <1>      CMP      byte [DSKETTE_STATUS],0      ; CHECK FOR ERROR
3843 00001382 7523                  <1>      JNZ      NT_OUT      ; IF ERROR 0 TRANSFERRED
3844 00001384 B204                  <1>      MOV      DL,4      ; SECTORS/TRACK OFFSET TO DL
3845 00001386 E8BE00              <1>      CALL     GET_PARM      ; AH = SECTORS/TRACK
3846 00001389 8A1E[500B]          <1>      MOV      BL, [NEC_STATUS+5]      ; GET ENDING SECTOR
3847 0000138D 89F1                  <1>      MOV      CX,SI      ; CH = HEAD # STARTED
3848 0000138F 3A2E[4F0B]          <1>      CMP      CH, [NEC_STATUS+4]      ; GET HEAD ENDED UP ON
3849 00001393 750B                  <1>      JNZ      DIF_HD      ; IF ON SAME HEAD, THEN NO ADJUST
3850 00001395 8A2E[4E0B]          <1>      MOV      CH, [NEC_STATUS+3]      ; GET TRACK ENDED UP ON
3851 00001399 3A6E01              <1>      CMP      CH, [BP+1]      ; IS IT ASKED FOR TRACK
3852 0000139C 7404                  <1>      JZ       short SAME_TRK      ; IF SAME TRACK NO INCREASE
3853 0000139E 00E3                  <1>      ADD      BL,AH      ; ADD SECTORS/TRACK
3854                                <1> DIF_HD:
3855 000013A0 00E3                  <1>      ADD      BL,AH      ; ADD SECTORS/TRACK
3856                                <1> SAME_TRK:
3857 000013A2 2A5E00              <1>      SUB      BL, [BP]      ; SUBTRACT START FROM END
3858 000013A5 88D8                  <1>      MOV      AL,BL      ; TO AL
3859                                <1> NT_OUT:
3860 000013A7 C3                  <1>      RETn
3861                                <1>
3862                                <1> ;-----
3863                                <1> ; SETUP_END
3864                                <1> ;     RESTORES @MOTOR_COUNT TO PARAMETER PROVIDED IN TABLE
3865                                <1> ;     AND LOADS @DSKETTE_STATUS TO AH, AND SETS CY.
3866                                <1> ;
3867                                <1> ; ON EXIT:

```

```

3868      <1> ;      AH, @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
3869      <1> ;-----
3870      <1> SETUP_END:
3871      000013A8 B202      <1>      MOV     DL,2          ; GET THE MOTOR WAIT PARAMETER
3872      000013AA 50        <1>      PUSH    AX              ; SAVE NUMBER TRANSFERRED
3873      000013AB E89900    <1>      CALL    GET_PARM
3874      000013AE 8826[490B] <1>      MOV     [MOTOR_COUNT],AH    ; STORE UPON RETURN
3875      000013B2 58        <1>      POP     AX              ; RESTORE NUMBER TRANSFERRED
3876      000013B3 8A26[4A0B] <1>      MOV     AH, [DSKETTE_STATUS] ; GET STATUS OF OPERATION
3877      000013B7 08E4      <1>      OR      AH,AH          ; CHECK FOR ERROR
3878      000013B9 7402      <1>      JZ      short NUN_ERR      ; NO ERROR
3879      000013BB 30C0      <1>      XOR     AL,AL          ; CLEAR NUMBER RETURNED
3880      <1> NUN_ERR:
3881      000013BD 80FC01    <1>      CMP     AH,1          ; SET THE CARRY FLAG TO INDICATE
3882      000013C0 F5        <1>      CMC              ; SUCCESS OR FAILURE
3883      000013C1 C3        <1>      RETN
3884      <1>
3885      <1> ;-----
3886      <1> ; SETUP_DBL
3887      <1> ;      CHECK DOUBLE STEP.
3888      <1> ;
3889      <1> ; ON ENTRY : DI = DRIVE
3890      <1> ;
3891      <1> ; ON EXIT : CY = 1 MEANS ERROR
3892      <1> ;-----
3893      <1> SETUP_DBL:
3894      000013C2 8AA5[530B] <1>      MOV     AH, [DSK_STATE+DI] ; ACCESS STATE
3895      000013C6 F6C410    <1>      TEST    AH,MED_DET      ; ESTABLISHED STATE ?
3896      000013C9 755E      <1>      JNZ     short NO_DBL      ; IF ESTABLISHED THEN DOUBLE DONE
3897      <1>
3898      <1> ;-----      CHECK FOR TRACK 0 TO SPEED UP ACKNOWLEDGE OF UNFORMATTED DISKETTE
3899      <1>
3900      000013CB C606[470B]00 <1>      MOV     byte [SEEK_STATUS],0 ; SET RECALIBRATE REQUIRED ON ALL DRIVES
3901      000013D0 E8A500    <1>      CALL    MOTOR_ON      ; ENSURE MOTOR STAY ON
3902      000013D3 B500      <1>      MOV     CH,0          ; LOAD TRACK 0
3903      000013D5 E86301    <1>      CALL    SEEK          ; SEEK TO TRACK 0
3904      000013D8 E85000    <1>      CALL    READ_ID       ; READ ID FUNCTION
3905      000013DB 7237      <1>      JC      short SD_ERR      ; IF ERROR NO TRACK 0
3906      <1>
3907      <1> ;-----      INITIALIZE START AND MAX TRACKS (TIMES 2 FOR BOTH HEADS)
3908      <1>
3909      000013DD B95004      <1>      MOV     CX,0450H      ; START, MAX TRACKS
3910      000013E0 F685[530B]01 <1>      TEST    byte [DSK_STATE+DI],TRK_CAPA ; TEST FOR 80 TRACK CAPABILITY
3911      000013E5 7402      <1>      JZ      short CNT_OK      ; IF NOT COUNT IS SETUP
3912      000013E7 B1A0      <1>      MOV     CL,0A0H      ; MAXIMUM TRACK 1.2 MB
3913      <1>
3914      <1> ;      ATTEMPT READ ID OF ALL TRACKS, ALL HEADS UNTIL SUCCESS; UPON SUCCESS,
3915      <1> ;      MUST SEE IF ASKED FOR TRACK IN SINGLE STEP MODE = TRACK ID READ; IF NOT
3916      <1> ;      THEN SET DOUBLE STEP ON.

```

3917		<1>	
3918		<1>	CNT_OK:
3919	000013E9 C606[490B]FF	<1>	MOV byte [MOTOR_COUNT], 0FFH ; ENSURE MOTOR STAYS ON FOR OPERATION
3920	000013EE 51	<1>	PUSH CX ; SAVE TRACK, COUNT
3921	000013EF C606[4A0B]00	<1>	MOV byte [DSKETTE_STATUS], 0 ; CLEAR STATUS, EXPECT ERRORS
3922	000013F4 31C0	<1>	XOR AX, AX ; CLEAR AX
3923	000013F6 D0ED	<1>	SHR CH, 1 ; HALVE TRACK, CY = HEAD
3924	000013F8 C0D003	<1>	RCL AL, 3 ; AX = HEAD IN CORRECT BIT
3925	000013FB 50	<1>	PUSH AX ; SAVE HEAD
3926	000013FC E83C01	<1>	CALL SEEK ; SEEK TO TRACK
3927	000013FF 58	<1>	POP AX ; RESTORE HEAD
3928	00001400 09C7	<1>	OR DI, AX ; DI = HEAD OR'ED DRIVE
3929	00001402 E82600	<1>	CALL READ_ID ; READ ID HEAD 0
3930	00001405 9C	<1>	PUSHF ; SAVE RETURN FROM READ_ID
3931	00001406 81E7FB00	<1>	AND DI, 11111011B ; TURN OFF HEAD 1 BIT
3932	0000140A 9D	<1>	POPF ; RESTORE ERROR RETURN
3933	0000140B 59	<1>	POP CX ; RESTORE COUNT
3934	0000140C 7308	<1>	JNC short DO_CHK ; IF OK, ASKED = RETURNED TRACK ?
3935	0000140E FEC5	<1>	INC CH ; INC FOR NEXT TRACK
3936	00001410 38CD	<1>	CMP CH, CL ; REACHED MAXIMUM YET
3937	00001412 75D5	<1>	JNZ short CNT_OK ; CONTINUE TILL ALL TRIED
3938		<1>	
3939		<1>	;----- FALL THRU, READ ID FAILED FOR ALL TRACKS
3940		<1>	
3941		<1>	SD_ERR:
3942	00001414 F9	<1>	STC ; SET CARRY FOR ERROR
3943	00001415 C3	<1>	RETn ; SETUP_DBL ERROR EXIT
3944		<1>	
3945		<1>	DO_CHK:
3946	00001416 8A0E[4E0B]	<1>	MOV CL, [NEC_STATUS+3] ; LOAD RETURNED TRACK
3947	0000141A 888D[570B]	<1>	MOV [DSK_TRK+DI], CL ; STORE TRACK NUMBER
3948	0000141E D0ED	<1>	SHR CH, 1 ; HALVE TRACK
3949	00001420 38CD	<1>	CMP CH, CL ; IS IT THE SAME AS ASKED FOR TRACK
3950	00001422 7405	<1>	JZ short NO_DBL ; IF SAME THEN NO DOUBLE STEP
3951	00001424 808D[530B]20	<1>	OR byte [DSK_STATE+DI], DBL_STEP ; TURN ON DOUBLE STEP REQUIRED
3952		<1>	NO_DBL:
3953	00001429 F8	<1>	CLC ; CLEAR ERROR FLAG
3954	0000142A C3	<1>	RETn
3955		<1>	
3956		<1>	;-----
3957		<1>	; READ_ID
3958		<1>	; READ ID FUNCTION.
3959		<1>	;
3960		<1>	; ON ENTRY: DI : BIT 2 = HEAD; BITS 1,0 = DRIVE
3961		<1>	;
3962		<1>	; ON EXIT: DI : BIT 2 IS RESET, BITS 1,0 = DRIVE
3963		<1>	; @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
3964		<1>	;-----
3965		<1>	READ_ID:

3966 0000142B B8[3F14]	<1>	MOV	AX, ER_3	; MOVE NEC OUTPUT ERROR ADDRESS
3967 0000142E 50	<1>	PUSH	AX	
3968 0000142F B44A	<1>	MOV	AH, 4AH	; READ ID COMMAND
3969 00001431 E8E000	<1>	CALL	NEC_OUTPUT	; TO CONTROLLER
3970 00001434 89F8	<1>	MOV	AX, DI	; DRIVE # TO AH, HEAD 0
3971 00001436 88C4	<1>	MOV	AH, AL	
3972 00001438 E8D900	<1>	CALL	NEC_OUTPUT	; TO CONTROLLER
3973 0000143B E871FE	<1>	CALL	NEC_TERM	; WAIT FOR OPERATION, GET STATUS
3974 0000143E 58	<1>	POP	AX	; THROW AWAY ERROR ADDRESS
3975	<1>	ER_3:		
3976 0000143F C3	<1>	RETN		
3977	<1>			
3978	<1>	;-----		
3979	<1>	; CMOS_TYPE		
3980	<1>	; RETURNS DISKETTE TYPE FROM CMOS		
3981	<1>	;		
3982	<1>	; ON ENTRY: DI = DRIVE #		
3983	<1>	;		
3984	<1>	; ON EXIT: AL = TYPE; CY REFLECTS STATUS		
3985	<1>	;-----		
3986	<1>			
3987	<1>	CMOS_TYPE: ; 11/12/2014		
3988 00001440 8A85[3829]	<1>	mov	al, [DI+fd0_type]	
3989 00001444 20C0	<1>	and	al, al ; 18/12/2014	
3990 00001446 C3	<1>	retn		
3991	<1>			
3992	<1>	; CMOS_TYPE:		
3993	<1>	MOV	AL, CMOS_DIAG	; CMOS DIAGNOSTIC STATUS BYTE ADDRESS
3994	<1>	CALL	CMOS_READ	; GET CMOS STATUS
3995	<1>	TEST	AL, BAD_BAT+BAD_CKSUM	; BATTERY GOOD AND CHECKSUM VALID
3996	<1>	STC		; SET CY = 1 INDICATING ERROR FOR RETURN
3997	<1>	JNZ	short BAD_CM	; ERROR IF EITHER BIT ON
3998	<1>	MOV	AL, CMOS_DISKETTE	; ADDRESS OF DISKETTE BYTE IN CMOS
3999	<1>	CALL	CMOS_READ	; GET DISKETTE BYTE
4000	<1>	OR	DI, DI	; SEE WHICH DRIVE IN QUESTION
4001	<1>	JNZ	short TB	; IF DRIVE 1, DATA IN LOW NIBBLE
4002	<1>	ROR	AL, 4	; EXCHANGE NIBBLES IF SECOND DRIVE
4003	<1>	; TB:		
4004	<1>	AND	AL, 0FH	; KEEP ONLY DRIVE DATA, RESET CY, 0
4005	<1>	; BAD_CM:		
4006	<1>	RETN		
4007	<1>	; CY, STATUS OF READ		
4008	<1>	;-----		
4009	<1>	; GET_PARM		
4010	<1>	; THIS ROUTINE FETCHES THE INDEXED POINTER FROM THE DISK_BASE		
4011	<1>	; BLOCK POINTED TO BY THE DATA VARIABLE @DISK_POINTER. A BYTE FROM		
4012	<1>	; THAT TABLE IS THEN MOVED INTO AH, THE INDEX OF THAT BYTE BEING		
4013	<1>	; THE PARAMETER IN DL.		
4014	<1>	;		

```

4015 <1> ; ON ENTRY: DL = INDEX OF BYTE TO BE FETCHED
4016 <1> ;
4017 <1> ; ON EXIT: AH = THAT BYTE FROM BLOCK
4018 <1> ; AL,DH DESTROYED
4019 <1> ;-----
4020 <1> GET_PARM:
4021 <1> ;PUSH DS
4022 00001447 56 <1> PUSH SI
4023 <1> ;SUB AX,AX ; DS = 0, BIOS DATA AREA
4024 <1> ;MOV DS,AX
4025 <1> ;;mov ax, cs
4026 <1> ;;mov ds, ax
4027 00001448 87D3 <1> XCHG DX,BX ; BL = INDEX
4028 0000144A 28FF <1> SUB BH,BH ; BX = INDEX
4029 <1> ;LDS SI, [DISK_POINTER] ; POINT TO BLOCK
4030 <1> ;
4031 <1> ; 17/12/2014
4032 0000144C A1[590B] <1> mov ax, [cfd] ; current (AL) and previous fd (AH)
4033 0000144F 38E0 <1> cmp al, ah
4034 00001451 741B <1> je short gpndc
4035 00001453 A2[5A0B] <1> mov [pfd], al ; current drive -> previous drive
4036 00001456 53 <1> push bx
4037 00001457 88C3 <1> mov bl, al
4038 <1> ; 11/12/2014
4039 00001459 8A87[3829] <1> mov al, [BX+fd0_type] ; Drive type (0,1,2,3,4)
4040 <1> ; 18/12/2014
4041 0000145D 20C0 <1> and al, al
4042 0000145F 7505 <1> jnz short gpdtc
4043 00001461 BB[AE0B] <1> mov bx, MD_TBL6 ; 1.44 MB param. tbl. (default)
4044 00001464 EB03 <1> jmp short gpdpu
4045 <1> gpdtc:
4046 00001466 E89AFA <1> call DR_TYPE_CHECK
4047 <1> ; cf = 1 -> BX points to 1.44MB fd parameter table (default)
4048 <1> gpdpu:
4049 00001469 891E[370B] <1> mov [DISK_POINTER], bx
4050 0000146D 5B <1> pop bx
4051 <1> gpndc:
4052 0000146E 8B36[370B] <1> mov si, [DISK_POINTER]
4053 00001472 8A20 <1> MOV AH, [SI+BX] ; GET THE WORD
4054 00001474 87D3 <1> XCHG DX,BX ; RESTORE BX
4055 00001476 5E <1> POP SI
4056 <1> ;POP DS
4057 00001477 C3 <1> RETn
4058 <1>
4059 <1> ;-----
4060 <1> ; MOTOR_ON
4061 <1> ; TURN MOTOR ON AND WAIT FOR MOTOR START UP TIME. THE @MOTOR_COUNT
4062 <1> ; IS REPLACED WITH A SUFFICIENTLY HIGH NUMBER (0FFH) TO ENSURE
4063 <1> ; THAT THE MOTOR DOES NOT GO OFF DURING THE OPERATION. IF THE

```

```

4064 <1> ; MOTOR NEEDED TO BE TURNED ON, THE MULTI-TASKING HOOK FUNCTION
4065 <1> ; (AX=90FDH, INT 15) IS CALLED TELLING THE OPERATING SYSTEM
4066 <1> ; THAT THE BIOS IS ABOUT TO WAIT FOR MOTOR START UP. IF THIS
4067 <1> ; FUNCTION RETURNS WITH CY = 1, IT MEANS THAT THE MINIMUM WAIT
4068 <1> ; HAS BEEN COMPLETED. AT THIS POINT A CHECK IS MADE TO ENSURE
4069 <1> ; THAT THE MOTOR WASN'T TURNED OFF BY THE TIMER. IF THE HOOK DID
4070 <1> ; NOT WAIT, THE WAIT FUNCTION (AH=086H) IS CALLED TO WAIT THE
4071 <1> ; PRESCRIBED AMOUNT OF TIME. IF THE CARRY FLAG IS SET ON RETURN,
4072 <1> ; IT MEANS THAT THE FUNCTION IS IN USE AND DID NOT PERFORM THE
4073 <1> ; WAIT. A TIMER 1 WAIT LOOP WILL THEN DO THE WAIT.
4074 <1> ;
4075 <1> ; ON ENTRY: DI = DRIVE #
4076 <1> ; ON EXIT: AX,CX,DX DESTROYED
4077 <1> ;-----
4078 <1> MOTOR_ON:
4079 00001478 53 <1> PUSH BX ; SAVE REG.
4080 00001479 E82000 <1> CALL TURN_ON ; TURN ON MOTOR
4081 0000147C 721C <1> JC short MOT_IS_ON ; IF CY=1 NO WAIT
4082 0000147E E8FEFA <1> CALL XLAT_OLD ; TRANSLATE STATE TO COMPATIBLE MODE
4083 00001481 E8D5FA <1> CALL XLAT_NEW ; TRANSLATE STATE TO PRESENT ARCH,
4084 <1> ;CALL TURN_ON ; CHECK AGAIN IF MOTOR ON
4085 <1> ;JC MOT_IS_ON ; IF NO WAIT MEANS IT IS ON
4086 <1> M_WAIT:
4087 00001484 B20A <1> MOV DL,10 ; GET THE MOTOR WAIT PARAMETER
4088 00001486 E8BEFF <1> CALL GET_PARM
4089 <1> ;MOV AL,AH ; AL = MOTOR WAIT PARAMETER
4090 <1> ;XOR AH,AH ; AX = MOTOR WAIT PARAMETER
4091 <1> ;CMP AL,8 ; SEE IF AT LEAST A SECOND IS SPECIFIED
4092 00001489 80FC08 <1> cmp ah, 8
4093 <1> ;JAE short GP2 ; IF YES, CONTINUE
4094 0000148C 7702 <1> ja short J13
4095 <1> ;MOV AL,8 ; ONE SECOND WAIT FOR MOTOR START UP
4096 0000148E B408 <1> mov ah, 8
4097 <1>
4098 <1> ;----- AS CONTAINS NUMBER OF 1/8 SECONDS (125000 MICROSECONDS) TO WAIT
4099 <1> GP2:
4100 <1> ;----- FOLLOWING LOOPS REQUIRED WHEN RTC WAIT FUNCTION IS ALREADY IN USE
4101 <1> J13: ; WAIT FOR 1/8 SECOND PER (AL)
4102 00001490 B95E20 <1> MOV CX,8286 ; COUNT FOR 1/8 SECOND AT 15.085737 US
4103 00001493 E86402 <1> CALL WAITF ; GO TO FIXED WAIT ROUTINE
4104 <1> ;DEC AL ; DECREMENT TIME VALUE
4105 00001496 FECC <1> dec ah
4106 00001498 75F6 <1> JNZ short J13 ; ARE WE DONE YET
4107 <1> MOT_IS_ON:
4108 0000149A 5B <1> POP BX ; RESTORE REG.
4109 0000149B C3 <1> RETn
4110 <1>
4111 <1> ;-----
4112 <1> ; TURN_ON

```

```

4113          <1> ;      TURN MOTOR ON AND RETURN WAIT STATE.
4114          <1> ;
4115          <1> ; ON ENTRY:  DI = DRIVE #
4116          <1> ;
4117          <1> ; ON EXIT:   CY = 0 MEANS WAIT REQUIRED
4118          <1> ;           CY = 1 MEANS NO WAIT REQUIRED
4119          <1> ;           AX,BX,CX,DX DESTROYED
4120          <1> ; -----
4121          <1> TURN_ON:
4122 0000149C 89FB          <1>      MOV     BX,DI          ; BX = DRIVE #
4123 0000149E 88D9          <1>      MOV     CL,BL          ; CL = DRIVE #
4124 000014A0 C0C304        <1>      ROL     BL,4          ; BL = DRIVE SELECT
4125 000014A3 FA            <1>      CLI             ; NO INTERRUPTS WHILE DETERMINING STATUS
4126 000014A4 C606[490B]FF <1>      MOV     byte [MOTOR_COUNT],0FFH ; ENSURE MOTOR STAYS ON FOR OPERATION
4127 000014A9 A0[480B]      <1>      MOV     AL, [MOTOR_STATUS] ; GET DIGITAL OUTPUT REGISTER REFLECTION
4128 000014AC 2430          <1>      AND     AL,00110000B ; KEEP ONLY DRIVE SELECT BITS
4129 000014AE B401          <1>      MOV     AH,1          ; MASK FOR DETERMINING MOTOR BIT
4130 000014B0 D2E4          <1>      SHL     AH,CL          ; AH = MOTOR ON, A=00000001, B=00000010
4131          <1>
4132          <1> ; AL = DRIVE SELECT FROM @MOTOR_STATUS
4133          <1> ; BL = DRIVE SELECT DESIRED
4134          <1> ; AH = MOTOR ON MASK DESIRED
4135          <1>
4136 000014B2 38D8          <1>      CMP     AL,BL          ; REQUESTED DRIVE ALREADY SELECTED ?
4137 000014B4 7506          <1>      JNZ     short TURN_IT_ON ; IF NOT SELECTED JUMP
4138 000014B6 8426[480B]      <1>      TEST    AH, [MOTOR_STATUS] ; TEST MOTOR ON BIT
4139 000014BA 752C          <1>      JNZ     short NO_MOT_WAIT ; JUMP IF MOTOR ON AND SELECTED
4140          <1>
4141          <1> TURN_IT_ON:
4142 000014BC 08DC          <1>      OR      AH,BL          ; AH = DRIVE SELECT AND MOTOR ON
4143 000014BE 8A3E[480B]      <1>      MOV     BH, [MOTOR_STATUS] ; SAVE COPY OF @MOTOR_STATUS BEFORE
4144 000014C2 80E70F          <1>      AND     BH,00001111B ; KEEP ONLY MOTOR BITS
4145 000014C5 8026[480B]CF   <1>      AND     byte [MOTOR_STATUS],11001111B ; CLEAR OUT DRIVE SELECT
4146 000014CA 0826[480B]      <1>      OR      byte [MOTOR_STATUS],AH ; OR IN DRIVE SELECTED AND MOTOR ON
4147 000014CE A0[480B]      <1>      MOV     AL, [MOTOR_STATUS] ; GET DIGITAL OUTPUT REGISTER REFLECTION
4148 000014D1 88C3          <1>      MOV     BL,AL          ; BL=@MOTOR_STATUS AFTER, BH=BEFORE
4149 000014D3 80E30F          <1>      AND     BL,00001111B ; KEEP ONLY MOTOR BITS
4150 000014D6 FB            <1>      STI             ; ENABLE INTERRUPTS AGAIN
4151 000014D7 243F          <1>      AND     AL,00111111B ; STRIP AWAY UNWANTED BITS
4152 000014D9 C0C004        <1>      ROL     AL,4          ; PUT BITS IN DESIRED POSITIONS
4153 000014DC 0C0C          <1>      OR      AL,00001100B ; NO RESET, ENABLE DMA/INTERRUPT
4154 000014DE BAF203        <1>      MOV     DX,03F2H ; SELECT DRIVE AND TURN ON MOTOR
4155 000014E1 EE            <1>      OUT     DX,AL
4156 000014E2 38FB          <1>      CMP     BL,BH          ; NEW MOTOR TURNED ON ?
4157 000014E4 7402          <1>      JZ      short NO_MOT_WAIT ; NO WAIT REQUIRED IF JUST SELECT
4158 000014E6 F8            <1>      CLC             ; (re)SET CARRY MEANING WAIT
4159 000014E7 C3            <1>      RETn
4160          <1>
4161          <1> NO_MOT_WAIT:

```

```

4162 000014E8 F9      <1>      STC                      ; SET NO WAIT REQUIRED
4163 000014E9 FB      <1>      STI                      ; INTERRUPTS BACK ON
4164 000014EA C3      <1>      RETn
4165                  <1>
4166                  <1> ; -----
4167                  <1> ; HD_WAIT
4168                  <1> ;     WAIT FOR HEAD SETTLE TIME.
4169                  <1> ;
4170                  <1> ; ON ENTRY:  DI = DRIVE #
4171                  <1> ;
4172                  <1> ; ON EXIT:   AX,BX,CX,DX DESTROYED
4173                  <1> ; -----
4174                  <1> HD_WAIT:
4175 000014EB B209      <1>      MOV     DL,9                ; GET HEAD SETTLE PARAMETER
4176 000014ED E857FF    <1>      CALL    GET_PARM
4177 000014F0 08E4      <1>      or      ah, ah ; 17/12/2014
4178 000014F2 7515      <1>      jnz     short DO_WAT
4179 000014F4 F606[480B]80 <1>      TEST    byte [MOTOR_STATUS],10000000B ; SEE IF A WRITE OPERATION
4180                  <1>      ;JZ     short ISNT_WRITE    ; IF NOT, DO NOT ENFORCE ANY VALUES
4181                  <1>      ;OR     AH,AH                ; CHECK FOR ANY WAIT?
4182                  <1>      ;JNZ    short DO_WAT         ; IF THERE DO NOT ENFORCE
4183 000014F9 7418      <1>      jz      short HW_DONE
4184 000014FB B40F      <1>      MOV     AH,HD12_SETTLE        ; LOAD 1.2M HEAD SETTLE MINIMUM
4185 000014FD 8A85[530B] <1>      MOV     AL,[DSK_STATE+DI] ; LOAD STATE
4186 00001501 24C0      <1>      AND     AL,RATE_MSK         ; KEEP ONLY RATE
4187 00001503 3C80      <1>      CMP     AL,RATE_250         ; 1.2 M DRIVE ?
4188 00001505 7502      <1>      JNZ     short DO_WAT         ; DEFAULT HEAD SETTLE LOADED
4189                  <1> ;GP3:
4190 00001507 B414      <1>      MOV     AH,HD320_SETTLE        ; USE 320/360 HEAD SETTLE
4191                  <1>      ;     JMP     SHORT DO_WAT
4192                  <1>
4193                  <1> ;ISNT_WRITE:
4194                  <1>      ;     OR     AH,AH                ; CHECK FOR NO WAIT
4195                  <1>      ;     JZ     short HW_DONE        ; IF NOT WRITE AND 0 ITS OK
4196                  <1>
4197                  <1> ;-----      AH CONTAINS NUMBER OF MILLISECONDS TO WAIT
4198                  <1> DO_WAT:
4199                  <1>      ;     MOV     AL,AH                ; AL = # MILLISECONDS
4200                  <1>      ;     ;XOR    AH,AH                ; AX = # MILLISECONDS
4201                  <1> J29:          <1>      ;     ; 1 MILLISECOND LOOP
4202                  <1>      ;mov     cx, WAIT_FDU_HEAD_SETTLE ; 33 ; 1 ms in 30 micro units.
4203 00001509 B94200      <1>      MOV     CX,66                ; COUNT AT 15.085737 US PER COUNT
4204 0000150C E8EB01      <1>      CALL    WAITF                ; DELAY FOR 1 MILLISECOND
4205                  <1>      ;DEC     AL                    ; DECREMENT THE COUNT
4206 0000150F FECC      <1>      dec     ah
4207 00001511 75F6      <1>      JNZ     short J29                ; DO AL MILLISECOND # OF TIMES
4208                  <1> HW_DONE:
4209 00001513 C3          <1>      RETn
4210                  <1>

```

```

4211 <1> ;-----
4212 <1> ; NEC_OUTPUT
4213 <1> ;     THIS ROUTINE SENDS A BYTE TO THE NEC CONTROLLER AFTER TESTING
4214 <1> ;     FOR CORRECT DIRECTION AND CONTROLLER READY THIS ROUTINE WILL
4215 <1> ;     TIME OUT IF THE BYTE IS NOT ACCEPTED WITHIN A REASONABLE AMOUNT
4216 <1> ;     OF TIME, SETTING THE DISKETTE STATUS ON COMPLETION.
4217 <1> ;
4218 <1> ; ON ENTRY: AH = BYTE TO BE OUTPUT
4219 <1> ;
4220 <1> ; ON EXIT:  CY = 0  SUCCESS
4221 <1> ;           CY = 1  FAILURE -- DISKETTE STATUS UPDATED
4222 <1> ;           IF A FAILURE HAS OCCURRED, THE RETURN IS MADE ONE LEVEL
4223 <1> ;           HIGHER THAN THE CALLER OF NEC_OUTPUT. THIS REMOVES THE
4224 <1> ;           REQUIREMENT OF TESTING AFTER EVERY CALL OF NEC_OUTPUT.
4225 <1> ;           AX,CX,DX DESTROYED
4226 <1> ;-----
4227 <1> ;
4228 <1> ; 09/12/2014 [Erdogan Tan]
4229 <1> ;     (from 'PS2 Hardware Interface Tech. Ref. May 88', Page 09-05.)
4230 <1> ; Diskette Drive Controller Status Register (3F4h)
4231 <1> ;     This read only register facilitates the transfer of data between
4232 <1> ;     the system microprocessor and the controller.
4233 <1> ; Bit 7 - When set to 1, the Data register is ready to transfer data
4234 <1> ;     with the system microprocessor.
4235 <1> ; Bit 6 - The direction of data transfer. If this bit is set to 0,
4236 <1> ;     the transfer is to the controller.
4237 <1> ; Bit 5 - When this bit is set to 1, the controller is in the non-DMA mode.
4238 <1> ; Bit 4 - When this bit is set to 1, a Read or Write command is being executed.
4239 <1> ; Bit 3 - Reserved.
4240 <1> ; Bit 2 - Reserved.
4241 <1> ; Bit 1 - When this bit is set to 1, diskette drive 1 is in the seek mode.
4242 <1> ; Bit 0 - When this bit is set to 1, diskette drive 1 is in the seek mode.
4243 <1> ;
4244 <1> ; Data Register (3F5h)
4245 <1> ; This read/write register passes data, commands and parameters, and provides
4246 <1> ; diskette status information.
4247 <1> ;
4248 <1> NEC_OUTPUT:
4249 <1>     ;PUSH  BX                ; SAVE REG.
4250 00001514 BAF403 <1>     MOV    DX,03F4H        ; STATUS PORT
4251 <1>     ;MOV   BL,2              ; HIGH ORDER COUNTER
4252 <1>     ;XOR   CX,CX            ; COUNT FOR TIME OUT
4253 <1>     ; 16/12/2014
4254 <1>     ; waiting for (max.) 0.5 seconds
4255 00001517 C606[4629]00 <1>     mov     byte [wait_count], 0
4256 <1>     ;
4257 <1>     ; 17/12/2014
4258 <1>     ; Modified from AWARD BIOS 1999 - ADISK.ASM - SEND_COMMAND
4259 <1>     ;

```

```

4260      <1>      ;WAIT_FOR_PORT:    Waits for a bit at a port pointed to by DX to
4261      <1>      ;                  go on.
4262      <1>      ;INPUT:
4263      <1>      ;      AH=Mask for isolation bits.
4264      <1>      ;      AL=pattern to look for.
4265      <1>      ;      DX=Port to test for
4266      <1>      ;      BH:CX=Number of memory refresh periods to delay.
4267      <1>      ;                  (normally 30 microseconds per period.)
4268      <1>      ;
4269      <1>      ;WFP_SHORT:
4270      <1>      ;      Wait for port if refresh cycle is short (15-80 Us range).
4271      <1>      ;
4272      <1>      ;
4273      <1>      ;      mov     bl, WAIT_FDU_SEND_HI+1      ; 0+1
4274      <1>      ;      mov     cx, WAIT_FDU_SEND_LO       ; 16667
4275      <1>      ;
4276      <1>      ;WFPS_OUTER_LP:
4277      <1>      ;
4278      <1>      ;WFPS_CHECK_PORT:
4279      <1>      ;;;J23:
4280      <1>      ;      IN      AL,DX                      ; GET STATUS
4281      <1>      ;      AND      AL,11000000B              ; KEEP STATUS AND DIRECTION
4282      <1>      ;      CMP      AL,10000000B              ; STATUS 1 AND DIRECTION 0 ?
4283      <1>      ;      JZ       short J27                  ; STATUS AND DIRECTION OK
4284      <1>      ;WFPS_HI:
4285      <1>      ;      IN      AL, PORT_B      ;061h ; SYS1 ; wait for hi to lo
4286      <1>      ;      TEST     AL,010H          ;          ; transition on memory
4287      <1>      ;      JNZ      SHORT WFPS_HI          ; refresh.
4288      <1>      ;WFPS_LO:
4289      <1>      ;      IN      AL, PORT_B          ; SYS1
4290      <1>      ;      TEST     AL,010H
4291      <1>      ;      JZ       SHORT WFPS_LO
4292      <1>      ;      LOOP     SHORT WFPS_CHECK_PORT
4293      <1>      ;
4294      <1>      ;      dec     bl
4295      <1>      ;      jnz      short WFPS_OUTER_LP
4296      <1>      ;      jmp      short WFPS_TIMEOUT ; fail
4297      <1>      ;
4298      <1>      J23:
4299      0000151C EC      <1>      IN      AL,DX                      ; GET STATUS
4300      0000151D 24C0    <1>      AND      AL,11000000B              ; KEEP STATUS AND DIRECTION
4301      0000151F 3C80    <1>      CMP      AL,10000000B              ; STATUS 1 AND DIRECTION 0 ?
4302      00001521 740F    <1>      JZ       short J27                  ; STATUS AND DIRECTION OK
4303      <1>      ;LOOP J23                      ; CONTINUE TILL CX EXHAUSTED
4304      <1>      ;DEC BL                          ; DECREMENT COUNTER
4305      <1>      ;JNZ short J23                  ; REPEAT TILL DELAY FINISHED, CX = 0
4306      <1>      ;
4307      <1>      ;16/12/2014
4308      00001523 803E[4629]0A <1>      cmp     byte [wait_count], 10      ; (10/18.2 seconds)

```

```

4309 00001528 72F2      <1>      jb      short J23
4310                    <1>
4311                    <1> ;WFPS_TIMEOUT:
4312                    <1>
4313                    <1> ;-----      FALL THRU TO ERROR RETURN
4314                    <1>
4315 0000152A 800E[4A0B]80 <1>      OR      byte [DSKETTE_STATUS],TIME_OUT
4316                    <1>      ;POP    BX      ; RESTORE REG.
4317 0000152F 58         <1>      POP    AX      ; DISCARD THE RETURN ADDRESS
4318 00001530 F9         <1>      STC      ; INDICATE ERROR TO CALLER
4319 00001531 C3         <1>      RETn
4320                    <1>
4321                    <1> ;-----      DIRECTION AND STATUS OK; OUTPUT BYTE
4322                    <1>
4323                    <1> J27:
4324 00001532 88E0        <1>      MOV     AL,AH      ; GET BYTE TO OUTPUT
4325 00001534 42         <1>      INC     DX      ; DATA PORT = STATUS PORT + 1
4326 00001535 EE         <1>      OUT     DX,AL      ; OUTPUT THE BYTE
4327                    <1>      IODELAY
4328 00001536 EB00        <2>      jmp short $+2
4329 00001538 EB00        <2>      jmp short $+2
4330                    <1>      ;
4331                    <1>      ;PUSHF      ; SAVE FLAGS
4332                    <1>      ;MOV     CX, 3      ; 30 TO 45 MICROSECONDS WAIT FOR
4333                    <1>      ;CALL    WAITF      ; NEC FLAGS UPDATE CYCLE
4334                    <1>      ;POPF      ; RESTORE FLAGS FOR EXIT
4335                    <1>      ;POP     BX      ; RESTORE REG
4336 0000153A C3         <1>      RETn      ; CY = 0 FROM TEST INSTRUCTION
4337                    <1>
4338                    <1> ;-----
4339                    <1> ; SEEK
4340                    <1> ;      THIS ROUTINE WILL MOVE THE HEAD ON THE NAMED DRIVE TO THE NAMED
4341                    <1> ;      TRACK. IF THE DRIVE HAS NOT BEEN ACCESSED SINCE THE DRIVE
4342                    <1> ;      RESET COMMAND WAS ISSUED, THE DRIVE WILL BE RECALIBRATED.
4343                    <1> ;
4344                    <1> ; ON ENTRY: DI = DRIVE #
4345                    <1> ;      CH = TRACK #
4346                    <1> ;
4347                    <1> ; ON EXIT: @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
4348                    <1> ;      AX,BX,CX DX DESTROYED
4349                    <1> ;-----
4350                    <1> SEEK:
4351 0000153B 89FB        <1>      MOV     BX,DI      ; BX = DRIVE #
4352 0000153D B001        <1>      MOV     AL,1      ; ESTABLISH MASK FOR RECALIBRATE TEST
4353 0000153F 86CB        <1>      XCHG    CL,BL      ; SET DRIVE VALULE INTO CL
4354 00001541 D2C0        <1>      ROL     AL,CL      ; SHIFT MASK BY THE DRIVE VALUE
4355 00001543 86CB        <1>      XCHG    CL,BL      ; RECOVER TRACK VALUE
4356 00001545 8406[470B] <1>      TEST    AL,[SEEK_STATUS] ; TEST FOR RECALIBRATE REQUIRED
4357 00001549 751C        <1>      JNZ     short J28A      ; JUMP IF RECALIBRATE NOT REQUIRED

```

4358		<1>	
4359	0000154B 0806[470B]	<1>	OR [SEEK_STATUS],AL ; TURN ON THE NO RECALIBRATE BIT IN FLAG
4360	0000154F E84800	<1>	CALL RECAL ; RECALIBRATE DRIVE
4361	00001552 730A	<1>	JNC short AFT_RECAL ; RECALIBRATE DONE
4362		<1>	
4363		<1>	;----- ISSUE RECALIBRATE FOR 80 TRACK DISKETTES
4364		<1>	
4365	00001554 C606[4A0B]00	<1>	MOV byte [DSKETTE_STATUS],0 ; CLEAR OUT INVALID STATUS
4366	00001559 E83E00	<1>	CALL RECAL ; RECALIBRATE DRIVE
4367	0000155C 723B	<1>	JC short RB ; IF RECALIBRATE FAILS TWICE THEN ERROR
4368		<1>	
4369		<1>	AFT_RECAL:
4370	0000155E C685[570B]00	<1>	MOV byte [DSK_TRK+DI],0 ; SAVE NEW CYLINDER AS PRESENT POSITION
4371	00001563 08ED	<1>	OR CH,CH ; CHECK FOR SEEK TO TRACK 0
4372	00001565 742D	<1>	JZ short DO_WAIT ; HEAD SETTLE, CY = 0 IF JUMP
4373		<1>	
4374		<1>	;----- DRIVE IS IN SYNCHRONIZATION WITH CONTROLLER, SEEK TO TRACK
4375		<1>	
4376	00001567 F685[530B]20	<1>	J28A: TEST byte [DSK_STATE+DI],DBL_STEP ; CHECK FOR DOUBLE STEP REQUIRED
4377	0000156C 7402	<1>	JZ short R7 ; SINGLE STEP REQUIRED BYPASS DOUBLE
4378	0000156E D0E5	<1>	SHL CH,1 ; DOUBLE NUMBER OF STEP TO TAKE
4379		<1>	
4380	00001570 3AAD[570B]	<1>	R7: CMP CH, [DSK_TRK+DI] ; SEE IF ALREADY AT THE DESIRED TRACK
4381	00001574 7423	<1>	JE short RB ; IF YES, DO NOT NEED TO SEEK
4382		<1>	
4383	00001576 BA[9915]	<1>	MOV DX, NEC_ERR ; LOAD RETURN ADDRESS
4384	00001579 52	<1>	PUSH DX ; ON STACK FOR NEC OUTPUT ERROR
4385	0000157A 88AD[570B]	<1>	MOV [DSK_TRK+DI],CH ; SAVE NEW CYLINDER AS PRESENT POSITION
4386	0000157E B40F	<1>	MOV AH,0FH ; SEEK COMMAND TO NEC
4387	00001580 E891FF	<1>	CALL NEC_OUTPUT
4388	00001583 89FB	<1>	MOV BX,DI ; BX = DRIVE #
4389	00001585 88DC	<1>	MOV AH,BL ; OUTPUT DRIVE NUMBER
4390	00001587 E88AFF	<1>	CALL NEC_OUTPUT
4391	0000158A 8AA5[570B]	<1>	MOV AH, [DSK_TRK+DI] ; GET CYLINDER NUMBER
4392	0000158E E883FF	<1>	CALL NEC_OUTPUT
4393	00001591 E81D00	<1>	CALL CHK_STAT_2 ; ENDING INTERRUPT AND SENSE STATUS
4394		<1>	
4395		<1>	;----- WAIT FOR HEAD SETTLE
4396		<1>	
4397		<1>	DO_WAIT:
4398	00001594 9C	<1>	PUSHF ; SAVE STATUS
4399	00001595 E853FF	<1>	CALL HD_WAIT ; WAIT FOR HEAD SETTLE TIME
4400	00001598 9D	<1>	POPF ; RESTORE STATUS
4401		<1>	RB:
4402		<1>	NEC_ERR:
4403	00001599 C3	<1>	RETn ; RETURN TO CALLER
4404		<1>	
4405		<1>	;-----
4406		<1>	; RECAL

```

4407          <1> ;      RECALIBRATE DRIVE
4408          <1> ;
4409          <1> ; ON ENTRY:  DI = DRIVE #
4410          <1> ;
4411          <1> ; ON EXIT:   CY REFLECTS STATUS OF OPERATION.
4412          <1> ;-----
4413          <1> RECAL:
4414 0000159A 51          <1>      PUSH    CX
4415 0000159B B8[AF15]   <1>      MOV     AX, RC_BACK      ; LOAD NEC_OUTPUT ERROR
4416 0000159E 50          <1>      PUSH    AX
4417 0000159F B407       <1>      MOV     AH, 07H      ; RECALIBRATE COMMAND
4418 000015A1 E870FF      <1>      CALL    NEC_OUTPUT
4419 000015A4 89FB       <1>      MOV     BX, DI      ; BX = DRIVE #
4420 000015A6 88DC       <1>      MOV     AH, BL
4421 000015A8 E869FF      <1>      CALL    NEC_OUTPUT      ; OUTPUT THE DRIVE NUMBER
4422 000015AB E80300      <1>      CALL    CHK_STAT_2      ; GET THE INTERRUPT AND SENSE INT STATUS
4423 000015AE 58          <1>      POP     AX      ; THROW AWAY ERROR
4424          <1> RC_BACK:
4425 000015AF 59          <1>      POP     CX
4426 000015B0 C3          <1>      RETn
4427          <1>
4428          <1> ;-----
4429          <1> ; CHK_STAT_2
4430          <1> ;      THIS ROUTINE HANDLES THE INTERRUPT RECEIVED AFTER RECALIBRATE,
4431          <1> ;      OR SEEK TO THE ADAPTER. THE INTERRUPT IS WAITED FOR, THE
4432          <1> ;      INTERRUPT STATUS SENSED, AND THE RESULT RETURNED TO THE CALLER.
4433          <1> ;
4434          <1> ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
4435          <1> ;-----
4436          <1> CHK_STAT_2:
4437 000015B1 B8[CF15]   <1>      MOV     AX, CS_BACK      ; LOAD NEC_OUTPUT ERROR ADDRESS
4438 000015B4 50          <1>      PUSH    AX
4439 000015B5 E82000      <1>      CALL    WAIT_INT      ; WAIT FOR THE INTERRUPT
4440 000015B8 7214       <1>      JC      short J34      ; IF ERROR, RETURN IT
4441 000015BA B408       <1>      MOV     AH, 08H      ; SENSE INTERRUPT STATUS COMMAND
4442 000015BC E855FF      <1>      CALL    NEC_OUTPUT
4443 000015BF E83900      <1>      CALL    RESULTS      ; READ IN THE RESULTS
4444 000015C2 720A       <1>      JC      short J34
4445 000015C4 A0[4B0B]   <1>      MOV     AL, [NEC_STATUS]      ; GET THE FIRST STATUS BYTE
4446 000015C7 2460       <1>      AND     AL, 01100000B      ; ISOLATE THE BITS
4447 000015C9 3C60       <1>      CMP     AL, 01100000B      ; TEST FOR CORRECT VALUE
4448 000015CB 7403       <1>      JZ      short J35      ; IF ERROR, GO MARK IT
4449 000015CD F8          <1>      CLC      ; GOOD RETURN
4450          <1> J34:
4451 000015CE 58          <1>      POP     AX      ; THROW AWAY ERROR RETURN
4452          <1> CS_BACK:
4453 000015CF C3          <1>      RETn
4454          <1> J35:
4455 000015D0 800E[4A0B]40 <1>      OR      byte [DSKETTE_STATUS], BAD_SEEK

```

```

4456 000015D5 F9      <1>      STC                      ; ERROR RETURN CODE
4457 000015D6 EBF6    <1>      JMP      SHORT J34
4458                  <1>
4459                  <1> ;-----
4460                  <1> ; WAIT_INT
4461                  <1> ;     THIS ROUTINE WAITS FOR AN INTERRUPT TO OCCUR A TIME OUT ROUTINE
4462                  <1> ;     TAKES PLACE DURING THE WAIT, SO THAT AN ERROR MAY BE RETURNED
4463                  <1> ;     IF THE DRIVE IS NOT READY.
4464                  <1> ;
4465                  <1> ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
4466                  <1> ;-----
4467                  <1>
4468                  <1> ; 17/12/2014
4469                  <1> ; 2.5 seconds waiting !
4470                  <1> ; (AWARD BIOS - 1999, WAIT_FDU_INT_LOW, WAIT_FDU_INT_HI)
4471                  <1> ; amount of time to wait for completion interrupt from NEC.
4472                  <1>
4473                  <1>
4474                  <1> WAIT_INT:
4475 000015D8 FB        <1>      STI                      ; TURN ON INTERRUPTS, JUST IN CASE
4476 000015D9 F8        <1>      CLC                      ; CLEAR TIMEOUT INDICATOR
4477                  <1>      ;MOV  BL,10                ; CLEAR THE COUNTERS
4478                  <1>      ;XOR  CX,CX                ; FOR 2 SECOND WAIT
4479                  <1>
4480                  <1>      ; Modification from AWARD BIOS - 1999 (ATORGS.ASM, WAIT
4481                  <1>      ;
4482                  <1>      ;WAIT_FOR_MEM:
4483                  <1>      ;     Waits for a bit at a specified memory location pointed
4484                  <1>      ;     to by ES:[DI] to become set.
4485                  <1>      ;INPUT:
4486                  <1>      ;     AH=Mask to test with.
4487                  <1>      ;     ES:[DI] = memory location to watch.
4488                  <1>      ;     BH:CX=Number of memory refresh periods to delay.
4489                  <1>      ;     (normally 30 microseconds per period.)
4490                  <1>
4491                  <1>      ; waiting for (max.) 2.5 secs in 30 micro units.
4492                  <1> ;     mov cx, WAIT_FDU_INT_LO      ; 017798
4493                  <1> ;;     mov bl, WAIT_FDU_INT_HI
4494                  <1> ;     mov bl, WAIT_FDU_INT_HI + 1
4495                  <1>
4496                  <1> ;WFMS_CHECK_MEM:
4497                  <1> ;     test  byte [SEEK_STATUS],INT_FLAG ; TEST FOR INTERRUPT OCCURRING
4498                  <1> ;     jnz   shortr J37
4499                  <1> ;WFMS_HI:
4500                  <1> ;     IN    AL,PORT_B  ; 061h   ; SYS1, wait for lo to hi
4501                  <1> ;     TEST  AL,010H                ; transition on memory
4502                  <1> ;     JNZ   SHORT WFMS_HI          ; refresh.
4503                  <1> ;WFMS_LO:
4504                  <1> ;     IN    AL,PORT_B                ;SYS1

```

4505	<1> ;	TEST	AL,010H	
4506	<1> ;	JZ	SHORT WFMS_LO	
4507	<1> ;	LOOP	SHORT WFMS_CHECK_MEM	
4508	<1> ;	WFMS_OUTER_LP:		
4509	<1> ;;	or	bl, bl	; check outer counter
4510	<1> ;;	jz	short J36A	; WFMS_TIMEOUT
4511	<1> ;	dec	bl	
4512	<1> ;	jz	short J36A	
4513	<1> ;	jmp	short WFMS_CHECK_MEM	
4514	<1>			
4515	<1>		;17/12/2014	
4516	<1>		;16/12/2014	
4517 000015DA C606[4629]00	<1>	mov	byte [wait_count], 0	; Reset (INT 08H) counter
4518	<1>	J36:		
4519 000015DF F606[470B]80	<1>	TEST	byte [SEEK_STATUS],INT_FLAG	; TEST FOR INTERRUPT OCCURRING
4520 000015E4 750D	<1>	JNZ	short J37	
4521	<1>		;16/12/2014	
4522	<1>	;LOOP	J36	; COUNT DOWN WHILE WAITING
4523	<1>	;DEC	BL	; SECOND LEVEL COUNTER
4524	<1>	;JNZ	short J36	
4525 000015E6 803E[4629]2E	<1>	cmp	byte [wait_count], 46	; (46/18.2 seconds)
4526 000015EB 72F2	<1>	jb	short J36	
4527	<1>			
4528	<1>	;WFMS_TIMEOUT:		
4529	<1>	J36A:		
4530 000015ED 800E[4A0B]80	<1>	OR	byte [DSKETTE_STATUS], TIME_OUT	; NOTHING HAPPENED
4531 000015F2 F9	<1>	STC		; ERROR RETURN
4532	<1>	J37:		
4533 000015F3 9C	<1>	PUSHF		; SAVE CURRENT CARRY
4534 000015F4 8026[470B]7F	<1>	AND	byte [SEEK_STATUS], ~INT_FLAG	; TURN OFF INTERRUPT FLAG
4535 000015F9 9D	<1>	POPF		; RECOVER CARRY
4536 000015FA C3	<1>	RETn		; GOOD RETURN CODE
4537	<1>			
4538	<1>			
4539	<1>	; RESULTS		
4540	<1>		THIS ROUTINE WILL READ ANYTHING THAT THE NEC CONTROLLER RETURNS	
4541	<1>		FOLLOWING AN INTERRUPT.	
4542	<1>			
4543	<1>	; ON EXIT:	@DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.	
4544	<1>		AX,BX,CX,DX DESTROYED	
4545	<1>			
4546	<1>	RESULTS:		
4547 000015FB 57	<1>	PUSH	DI	
4548 000015FC BF[4B0B]	<1>	MOV	DI, NEC_STATUS	; POINTER TO DATA AREA
4549 000015FF B307	<1>	MOV	BL,7	; MAX STATUS BYTES
4550 00001601 BAF403	<1>	MOV	DX,03F4H	; STATUS PORT
4551	<1>			
4552	<1>	;-----	WAIT FOR REQUEST FOR MASTER	
4553	<1>			

```

4554 <1> _R10:
4555 <1> ; 16/12/2014
4556 <1> ; wait for (max) 0.5 seconds
4557 <1> ;MOV BH,2 ; HIGH ORDER COUNTER
4558 <1> ;XOR CX,CX ; COUNTER
4559 <1>
4560 <1> ;Time to wait while waiting for each byte of NEC results = .5
4561 <1> ;seconds. .5 seconds = 500,000 micros. 500,000/30 = 16,667.
4562 00001604 B91B41 <1> mov cx, WAIT_FDU_RESULTS_LO ; 16667
4563 00001607 B701 <1> mov bh, WAIT_FDU_RESULTS_HI+1 ; 0+1
4564 <1>
4565 <1> WFPSR_OUTER_LP:
4566 <1> ;
4567 <1> WFPSR_CHECK_PORT:
4568 <1> J39: ; WAIT FOR MASTER
4569 00001609 EC <1> IN AL,DX ; GET STATUS
4570 0000160A 24C0 <1> AND AL,11000000B ; KEEP ONLY STATUS AND DIRECTION
4571 0000160C 3CC0 <1> CMP AL,11000000B ; STATUS 1 AND DIRECTION 1 ?
4572 0000160E 741A <1> JZ short J42 ; STATUS AND DIRECTION OK
4573 <1> WFPSR_HI:
4574 00001610 E461 <1> IN AL, PORT_B ;061h ; SYS1; wait for hi to lo
4575 00001612 A810 <1> TEST AL,010H ; transition on memory
4576 00001614 75FA <1> JNZ SHORT WFPSR_HI ; refresh.
4577 <1> WFPSR_LO:
4578 00001616 E461 <1> IN AL, PORT_B ; SYS1
4579 00001618 A810 <1> TEST AL,010H
4580 0000161A 74FA <1> JZ SHORT WFPSR_LO
4581 0000161C E2EB <1> LOOP WFPSR_CHECK_PORT
4582 <1> ;
4583 0000161E FECF <1> dec bh
4584 00001620 75E7 <1> jnz short WFPSR_OUTER_LP
4585 <1> ;jmp short WFPSR_TIMEOUT ; fail
4586 <1>
4587 <1> ;;mov byte [wait_count], 0
4588 <1> ;J39: ; WAIT FOR MASTER
4589 <1> ; IN AL,DX ; GET STATUS
4590 <1> ; AND AL,11000000B ; KEEP ONLY STATUS AND DIRECTION
4591 <1> ; CMP AL,11000000B ; STATUS 1 AND DIRECTION 1 ?
4592 <1> ; JZ short J42 ; STATUS AND DIRECTION OK
4593 <1> ;LOOP J39 ; LOOP TILL TIMEOUT
4594 <1> ;DEC BH ; DECREMENT HIGH ORDER COUNTER
4595 <1> ;JNZ short J39 ; REPEAT TILL DELAY DONE
4596 <1> ;
4597 <1> ;;cmp byte [wait_count], 10 ; (10/18.2 seconds)
4598 <1> ;;jb short J39
4599 <1>
4600 <1> ;WFPSR_TIMEOUT:
4601 00001622 800E[4A0B]80 <1> OR byte [DSKETTE_STATUS],TIME_OUT
4602 00001627 F9 <1> STC ; SET ERROR RETURN

```

```

4603 00001628 EB1F          <1>          JMP      SHORT POPRES          ; POP REGISTERS AND RETURN
4604          <1>
4605          <1> ;-----          READ IN THE STATUS
4606          <1>
4607          <1> J42:
4608          <1>          ;JMP      $+2          ; I/O DELAY
4609 0000162A 42          <1>          INC      DX          ; POINT AT DATA PORT
4610 0000162B EC          <1>          IN       AL,DX          ; GET THE DATA
4611          <1>          ; 16/12/2014
4612          <1>          NEWIODELAY
4613 0000162C E6EB        <2> out 0ebh,al
4614 0000162E 8805        <1>          MOV      [DI],AL          ; STORE THE BYTE
4615 00001630 47          <1>          INC      DI          ; INCREMENT THE POINTER
4616          <1>          ; 16/12/2014
4617          <1>          ; push    cx
4618          <1>          ; mov     cx, 30
4619          <1> ;wdw2:
4620          <1>          ; NEWIODELAY
4621          <1>          ; loop    wdw2
4622          <1>          ; pop     cx
4623          <1>
4624 00001631 B90300        <1>          MOV      CX,3          ; MINIMUM 24 MICROSECONDS FOR NEC
4625 00001634 E8C300        <1>          CALL    WAITF          ; WAIT 30 TO 45 MICROSECONDS
4626 00001637 4A          <1>          DEC      DX          ; POINT AT STATUS PORT
4627 00001638 EC          <1>          IN       AL,DX          ; GET STATUS
4628          <1>          ; 16/12/2014
4629          <1>          NEWIODELAY
4630 00001639 E6EB        <2> out 0ebh,al
4631          <1>          ;
4632 0000163B A810          <1>          TEST     AL,00010000B          ; TEST FOR NEC STILL BUSY
4633 0000163D 740A          <1>          JZ       short POPRES          ; RESULTS DONE ?
4634          <1>
4635 0000163F FECB          <1>          DEC      BL          ; DECREMENT THE STATUS COUNTER
4636 00001641 75C1          <1>          JNZ      short _R10          ; GO BACK FOR MORE
4637 00001643 800E[4A0B]20 <1>          OR       byte [DSKETTE_STATUS],BAD_NEC ; TOO MANY STATUS BYTES
4638 00001648 F9          <1>          STC          ; SET ERROR FLAG
4639          <1>
4640          <1> ;-----          RESULT OPERATION IS DONE
4641          <1> POPRES:
4642 00001649 5F          <1>          POP      DI
4643 0000164A C3          <1>          RETn          ; RETURN WITH CARRY SET
4644          <1>
4645          <1> ;-----
4646          <1> ; READ_DSKCHNG
4647          <1> ; READS THE STATE OF THE DISK CHANGE LINE.
4648          <1> ;
4649          <1> ; ON ENTRY: DI = DRIVE #
4650          <1> ;
4651          <1> ; ON EXIT: DI = DRIVE #

```

```

4652          <1> ;           ZF = 0 : DISK CHANGE LINE INACTIVE
4653          <1> ;           ZF = 1 : DISK CHANGE LINE ACTIVE
4654          <1> ;           AX,CX,DX DESTROYED
4655          <1> ; -----
4656          <1> READ_DSKCHNG:
4657          <1>         CALL  MOTOR_ON           ; TURN ON THE MOTOR IF OFF
4658          <1>         MOV   DX,03F7H           ; ADDRESS DIGITAL INPUT REGISTER
4659          <1>         IN    AL,DX               ; INPUT DIGITAL INPUT REGISTER
4660          <1>         TEST  AL,DSK_CHG         ; CHECK FOR DISK CHANGE LINE ACTIVE
4661          <1>         RETN                    ; RETURN TO CALLER WITH ZERO FLAG SET
4662          <1>
4663          <1> ; -----
4664          <1> ; DRIVE_DET
4665          <1> ;     DETERMINES WHETHER DRIVE IS 80 OR 40 TRACKS AND
4666          <1> ;     UPDATES STATE INFORMATION ACCORDINGLY.
4667          <1> ;
4668          <1> ; ON ENTRY:  DI = DRIVE #
4669          <1> ; -----
4670          <1> DRIVE_DET:
4671          <1>         CALL  MOTOR_ON           ; TURN ON MOTOR IF NOT ALREADY ON
4672          <1>         CALL  RECAL               ; RECALIBRATE DRIVE
4673          <1>         JC    short DD_BAC        ; ASSUME NO DRIVE PRESENT
4674          <1>         MOV   CH,TRK_SLAP        ; SEEK TO TRACK 48
4675          <1>         CALL  SEEK
4676          <1>         JC    short DD_BAC        ; ERROR NO DRIVE
4677          <1>         MOV   CH,QUIET_SEEK+1     ; SEEK TO TRACK 10
4678          <1> SK_GIN:
4679          <1>         DEC   CH                 ; DECREMENT TO NEXT TRACK
4680          <1>         PUSH  CX                 ; SAVE TRACK
4681          <1>         CALL  SEEK
4682          <1>         JC    short POP_BAC        ; POP AND RETURN
4683          <1>         MOV   AX, POP_BAC         ; LOAD NEC OUTPUT ERROR ADDRESS
4684          <1>         PUSH  AX
4685          <1>         MOV   AH,SENSE_DRV_ST     ; SENSE DRIVE STATUS COMMAND BYTE
4686          <1>         CALL  NEC_OUTPUT          ; OUTPUT TO NEC
4687          <1>         MOV   AX,DI               ; AL = DRIVE
4688          <1>         MOV   AH,AL               ; AH = DRIVE
4689          <1>         CALL  NEC_OUTPUT          ; OUTPUT TO NEC
4690          <1>         CALL  RESULTS             ; GO GET STATUS
4691          <1>         POP   AX                 ; THROW AWAY ERROR ADDRESS
4692          <1>         POP   CX                 ; RESTORE TRACK
4693          <1>         TEST  byte [NEC_STATUS], HOME ; TRACK 0 ?
4694          <1>         JZ    short SK_GIN        ; GO TILL TRACK 0
4695          <1>         OR    CH,CH               ; IS HOME AT TRACK 0
4696          <1>         JZ    short IS_80        ; MUST BE 80 TRACK DRIVE
4697          <1>
4698          <1> ;     DRIVE IS A 360; SET DRIVE TO DETERMINED;
4699          <1> ;     SET MEDIA TO DETERMINED AT RATE 250.
4700          <1>

```

4701 0000168E 808D[530B]94	<1> OR byte [DSK_STATE+DI], DRV_DET+MED_DET+RATE_250
4702 00001693 C3	<1> RETn ; ALL INFORMATION SET
4703	<1> IS_80:
4704 00001694 808D[530B]01	<1> OR byte [DSK_STATE+DI], TRK_CAPA ; SETUP 80 TRACK CAPABILITY
4705	<1> DD_BAC:
4706 00001699 C3	<1> RETn
4707	<1> POP_BAC:
4708 0000169A 59	<1> POP CX ; THROW AWAY
4709 0000169B C3	<1> RETn
4710	<1>
4711	<1> int_0Eh: ; 11/12/2014
4712	<1>
4713	<1> ;--- HARDWARE INT 0EH -- (IRQ LEVEL 6) -----
4714	<1> ; DISK_INT
4715	<1> ; THIS ROUTINE HANDLES THE DISKETTE INTERRUPT.
4716	<1> ;
4717	<1> ; ON EXIT: THE INTERRUPT FLAG IS SET IN @SEEK_STATUS.
4718	<1> ;-----
4719	<1> DISK_INT_1:
4720	<1>
4721	<1> ;jmp word far [CS:ORG_INT0EH]
4722	<1>
4723 0000169C 50	<1> PUSH AX ; SAVE WORK REGISTER
4724 0000169D 2E800E[470B]80	<1> OR byte [CS:SEEK_STATUS], INT_FLAG ; TURN ON INTERRUPT OCCURRED
4725 000016A3 B020	<1> MOV AL,E0I ; END OF INTERRUPT MARKER
4726 000016A5 E620	<1> OUT INTA00,AL ; INTERRUPT CONTROL PORT
4727 000016A7 58	<1> POP AX ; RECOVER REGISTER
4728 000016A8 CF	<1> IRET ; RETURN FROM INTERRUPT
4729	<1>
4730	<1> ;-----
4731	<1> ; DSKETTE_SETUP
4732	<1> ; THIS ROUTINE DOES A PRELIMINARY CHECK TO SEE WHAT TYPE OF
4733	<1> ; DISKETTE DRIVES ARE ATTACH TO THE SYSTEM.
4734	<1> ;-----
4735	<1> DSKETTE_SETUP:
4736 000016A9 50	<1> PUSH AX ; SAVE REGISTERS
4737 000016AA 53	<1> PUSH BX
4738 000016AB 51	<1> PUSH CX
4739 000016AC 52	<1> PUSH DX
4740 000016AD 57	<1> PUSH DI
4741	<1> ;PUSH DS
4742	<1> ; 14/12/2014
4743	<1> ;mov word [DISK_POINTER], MD_TBL6
4744 000016AE 8C0E[390B]	<1> mov word [DISK_POINTER+2], cs
4745	<1> ;
4746	<1> ;OR byte [RTC_WAIT_FLAG], 1 ; NO RTC WAIT, FORCE USE OF LOOP
4747 000016B2 31FF	<1> XOR DI,DI ; INITIALIZE DRIVE POINTER
4748 000016B4 C706[530B]0000	<1> MOV WORD [DSK_STATE],0 ; INITIALIZE STATES
4749 000016BA 8026[520B]33	<1> AND byte [LAstrate], ~(STRT_MSK+SEND_MSK) ; CLEAR START & SEND

4750 000016BF 800E[520B]C0	<1>	OR	byte [LASTRATE],SEND_MSK ; INITIALIZE SENT TO IMPOSSIBLE
4751 000016C4 C606[470B]00	<1>	MOV	byte [SEEK_STATUS],0 ; INDICATE RECALIBRATE NEEDED
4752 000016C9 C606[490B]00	<1>	MOV	byte [MOTOR_COUNT],0 ; INITIALIZE MOTOR COUNT
4753 000016CE C606[480B]00	<1>	MOV	byte [MOTOR_STATUS],0 ; INITIALIZE DRIVES TO OFF STATE
4754 000016D3 C606[4A0B]00	<1>	MOV	byte [DSKETTE_STATUS],0 ; NO ERRORS
4755	<1>	SUP0:	
4756 000016D8 E87AFF	<1>	CALL	DRIVE_DET ; DETERMINE DRIVE
4757 000016DB E8A1F8	<1>	CALL	XLAT_OLD ; TRANSLATE STATE TO COMPATIBLE MODE
4758	<1>		; 02/01/2015
4759	<1>	;INC	DI ; POINT TO NEXT DRIVE
4760	<1>	;CMP	DI,MAX_DRV ; SEE IF DONE
4761	<1>	;JNZ	short SUP0 ; REPEAT FOR EACH ORIVE
4762 000016DE 803E[3929]00	<1>	cmp	byte [fd1_type], 0
4763 000016E3 7607	<1>	jna	short sup1
4764 000016E5 09FF	<1>	or	di, di
4765 000016E7 7503	<1>	jnz	short sup1
4766 000016E9 47	<1>	inc	di
4767 000016EA EBEC	<1>	jmp	short SUP0
4768	<1>	sup1:	
4769 000016EC C606[470B]00	<1>	MOV	byte [SEEK_STATUS],0 ; FORCE RECALIBRATE
4770	<1>	;AND	byte [RTC_WAIT_FLAG],0FEH ; ALLOW FOR RTC WAIT
4771 000016F1 E8B4FC	<1>	CALL	SETUP_END ; VARIOUS CLEANUPS
4772	<1>	;POP	DS ; RESTORE CALLERS REGISTERS
4773 000016F4 5F	<1>	POP	DI
4774 000016F5 5A	<1>	POP	DX
4775 000016F6 59	<1>	POP	CX
4776 000016F7 5B	<1>	POP	BX
4777 000016F8 58	<1>	POP	AX
4778 000016F9 C3	<1>	RETn	
4779	<1>		
4780	<1>	; WAITF	
4781	<1>	; /// IBM PC-XT Model 286 System BIOS Source Code - Test 4 - 06/10/85 ///	
4782	<1>		
4783	<1>	;---WAITF-----	
4784	<1>	; FIXED TIME WAIT ROUTINE (HARDWARE CONTROLLED - NOT PROCESSOR)	
4785	<1>	; ENTRY:	
4786	<1>	; (CX) = COUNT OF 15.085737 MICROSECOND INTERVALS TO WAIT	
4787	<1>	; MEMORY REFRESH TIMER 1 OUTPUT USED AS REFERENCE	
4788	<1>	; EXIT:	
4789	<1>	; AFTER (CX) TIME COUNT (PLUS OR MINUS 16 MICROSECONDS)	
4790	<1>	; (CX) = 0	
4791	<1>	;-----	
4792	<1>		
4793	<1>	; Refresh period: 30 micro seconds (15-80 us)	
4794	<1>	; (16/12/2014 - AWARDBIOS 1999 - ATORG.S.ASM, WAIT_REFRESH)	
4795	<1>		
4796	<1>	WAITF: ; DELAY FOR (CX)*15.085737 US	
4797 000016FA 50	<1>	PUSH	AX ; SAVE WORK REGISTER (AH)
4798	<1>	; 16/12/2014	

```

4799 000016FB D1E9      <1>      shr    cx, 1                ; convert to count of 30 micro seconds
4800                    <1> ;17/12/2014
4801                    <1> ;WAITF1:
4802                    <1> ;      IN     AL, PORT_B    ;061h  ; READ CURRENT COUNTER OUTPUT STATUS
4803                    <1> ;      AND     AL, REFRESH_BIT ;00010000b ; MASK FOR REFRESH DETERMINE BIT
4804                    <1> ;      CMP     AL, AH          ; DID IT JUST CHANGE
4805                    <1> ;      JE      short WAITF1      ; WAIT FOR A CHANGE IN OUTPUT LINE
4806                    <1> ;      MOV     AH, AL          ; SAVE NEW FLAG STATE
4807                    <1> ;      LOOP    WAITF1          ; DECREMENT HALF CYCLES TILL COUNT END
4808                    <1> ;
4809                    <1> ; 17/12/2014
4810                    <1> ;
4811                    <1> ; Modification from 'WAIT_REFRESH' procedure of AWARD BIOS - 1999
4812                    <1> ;
4813                    <1> ;WAIT_REFRESH: Uses port 61, bit 4 to have CPU speed independent waiting.
4814                    <1> ;      INPUT:  CX = number of refresh periods to wait
4815                    <1> ;                      (refresh periods = 1 per 30 microseconds on most machines)
4816                    <1> WR_STATE_0:
4817 000016FD E461        <1>      IN     AL,PORT_B        ; IN AL,SYS1
4818 000016FF A810        <1>      TEST    AL,010H
4819 00001701 74FA        <1>      JZ      SHORT WR_STATE_0
4820                    <1> WR_STATE_1:
4821 00001703 E461        <1>      IN     AL,PORT_B        ; IN AL,SYS1
4822 00001705 A810        <1>      TEST    AL,010H
4823 00001707 75FA        <1>      JNZ     SHORT WR_STATE_1
4824 00001709 E2F2        <1>      LOOP    WR_STATE_0
4825                    <1> ;
4826 0000170B 58          <1>      POP     AX              ; RESTORE (AH)
4827 0000170C C3          <1>      RETn             ; (CX) = 0
4828                    <1> ;
4829                    <1> ; //////////////////////////////////
4830                    <1> ;
4831                    <1> ; 22/12/2014
4832                    <1> ; 17/12/2014 (Complete Timer Interrupt has been moved here)
4833                    <1> ;                      (Due to MOTOR_COUNT conflicts... Original timer interrupt
4834                    <1> ;                      was turning off the motor by using original MOTOR_COUNT in the
4835                    <1> ;                      BIOS Data Area !)
4836                    <1> ; 16/12/2014
4837                    <1> ; 12/12/2014
4838                    <1> ; 11/12/2014
4839                    <1> ; IBM PC-XT Model 286 Source Code - BIOS2.ASM (06/10/85)
4840                    <1> ;
4841                    <1> ;-- HARDWARE INT 08 H - ( IRQ LEVEL 0 ) -----
4842                    <1> ;      THIS ROUTINE HANDLES THE TIMER INTERRUPT FROM FROM CHANNEL 0 OF      :
4843                    <1> ;      THE 8254 TIMER. INPUT FREQUENCY IS 1.19318 MHZ AND THE DIVISOR      :
4844                    <1> ;      IS 65536, RESULTING IN APPROXIMATELY 18.2 INTERRUPTS EVERY SECOND.      :
4845                    <1> ;      :
4846                    <1> ;      THE INTERRUPT HANDLER MAINTAINS A COUNT (40:6C) OF INTERRUPTS SINCE      :
4847                    <1> ;      POWER ON TIME, WHICH MAY BE USED TO ESTABLISH TIME OF DAY.      :

```

```

4848 <1> ; THE INTERRUPT HANDLER ALSO DECREASES THE MOTOR CONTROL COUNT (40:40) :
4849 <1> ; OF THE DISKETTE, AND WHEN IT EXPIRES, WILL TURN OFF THE :
4850 <1> ; DISKETTE MOTOR(s), AND RESET THE MOTOR RUNNING FLAGS. :
4851 <1> ; THE INTERRUPT HANDLER WILL ALSO INVOKE A USER ROUTINE THROUGH :
4852 <1> ; INTERRUPT 1CH AT EVERY TIME TICK. THE USER MUST CODE A :
4853 <1> ; ROUTINE AND PLACE THE CORRECT ADDRESS IN THE VECTOR TABLE. :
4854 <1> ;-----
4855 <1> ;
4856 <1> ;
4857 <1> int_08h: ; Timer
4858 0000170D FB <1> STI ; INTERRUPTS BACK ON
4859 0000170E 1E <1> PUSH DS
4860 0000170F 50 <1> PUSH AX
4861 00001710 52 <1> PUSH DX ; SAVE MACHINE STATE
4862 00001711 31C0 <1> xor ax,ax
4863 00001713 8ED8 <1> mov ds,ax
4864 00001715 FF066C04 <1> INC word [TIMER_LOW] ; INCREMENT TIME
4865 00001719 7504 <1> JNZ short T4 ; GO TO TEST_DAY
4866 0000171B FF066E04 <1> INC word [TIMER_HIGH] ; INCREMENT HIGH WORD OF TIME
4867 <1> T4: ; TEST_DAY
4868 0000171F 833E6E0418 <1> CMP word [TIMER_HIGH],018H ; TEST FOR COUNT EQUALING 24 HOURS
4869 00001724 7513 <1> JNZ short T5 ; GO TO DISKETTE_CTL
4870 00001726 813E6C04B000 <1> CMP word [TIMER_LOW],0B0H
4871 0000172C 750B <1> JNZ short T5 ; GO TO DISKETTE_CTL
4872 <1>
4873 <1> ;----- TIMER HAS GONE 24 HOURS
4874 <1>
4875 <1> ;SUB AX,AX
4876 0000172E A36E04 <1> MOV word [TIMER_HIGH],AX
4877 00001731 A36C04 <1> MOV word [TIMER_LOW],AX
4878 00001734 C606700401 <1> MOV byte [TIMER_OFL],1
4879 <1>
4880 <1> ;----- TEST FOR DISKETTE TIME OUT
4881 <1>
4882 <1> T5:
4883 <1> ; 23/12/2014
4884 00001739 EB20 <1> jmp short T6 ; will be replaced with nop, nop
4885 <1> ; (9090h) if a floppy disk
4886 <1> ; is detected.
4887 0000173B 2EA0[490B] <1> mov al,[CS:MOTOR_COUNT]
4888 0000173F FEC8 <1> dec al
4889 00001741 2EA2[490B] <1> mov [CS:MOTOR_COUNT], al ; DECREMENT DISKETTE MOTOR CONTROL
4890 00001745 A24004 <1> mov [ORG_MOTOR_COUNT], al
4891 00001748 7511 <1> JNZ short T6 ; RETURN IF COUNT NOT OUT
4892 0000174A B0F0 <1> mov al,0F0h
4893 0000174C 2E2006[480B] <1> AND [CS:MOTOR_STATUS],al ; TURN OFF MOTOR RUNNING BITS
4894 00001751 20063F04 <1> and [ORG_MOTOR_STATUS], al
4895 00001755 B00C <1> MOV AL,0CH ; bit 3 = enable IRQ & DMA,
4896 <1> ; bit 2 = enable controller

```

```

4897                                     <1>                                     ;      1 = normal operation
4898                                     <1>                                     ;      0 = reset
4899                                     <1>                                     ; bit 0, 1 = drive select
4900                                     <1>                                     ; bit 4-7 = motor running bits
4901 00001757 BAF203                     <1>      MOV     DX,03F2H                     ; FDC CTL PORT
4902 0000175A EE                         <1>      OUT     DX,AL                       ; TURN OFF THE MOTOR
4903                                     <1> T6:
4904 0000175B 2EFF06[4629]               <1>      inc     word [CS:wait_count]       ; 22/12/2014 (byte -> word)
4905                                     <1>                                     ; TIMER TICK INTERRUPT
4906                                     <1>      ; DS = 0 (DS <> CS) !
4907 00001760 CD1C                       <1>      INT     1CH                       ; TRANSFER CONTROL TO A USER ROUTINE
4908                                     <1>      ;
4909 00001762 5A                         <1>      POP     DX                       ; RESTORE (DX)
4910 00001763 B020                       <1>      MOV     AL,E0I                     ; GET END OF INTERRUPT MASK
4911 00001765 FA                         <1>      CLI                      ; DISABLE INTERRUPTS TILL STACK CLEARED
4912 00001766 E620                       <1>      OUT     INTA00,AL                 ; END OF INTERRUPT TO 8259 - 1
4913 00001768 58                         <1>      POP     AX
4914 00001769 1F                         <1>      POP     DS                       ; RESET MACHINE STATE
4915 0000176A CF                         <1>      IRET                      ; RETURN FROM INTERRUPT
4916                                     <1>
4917                                     <1> ; ///////////////////////////////////
4918                                     <1>      %include 'disk.inc'           ; 23/12/2014
4919                                     <1> ; FIXED DISK I/O - Erdogan Tan (Retro UNIX 386 v1 project)
4920                                     <1> ;
4921                                     <1> ; 21/02/2015
4922                                     <1> ;
4923                                     <1> ; Original Source Code:
4924                                     <1> ; DISK ----- 09/25/85 FIXED DISK BIOS
4925                                     <1> ; (IBM PC XT Model 286 System BIOS Source Code, 04-21-86)
4926                                     <1> ;
4927                                     <1> ; Modifications: by reference of AWARD BIOS 1999 (D1A0622)
4928                                     <1> ; Source Code - ATORGS.ASM, AHDSK.ASM
4929                                     <1> ;
4930                                     <1>
4931                                     <1> ;The wait for controller to be not busy is 10 seconds.
4932                                     <1> ;10,000,000 / 30 = 333,333. 333,333 decimal = 051615h
4933                                     <1> WAIT_HDU_CTLR_BUSY_LO     equ    1615h
4934                                     <1> WAIT_HDU_CTLR_BUSY_HI     equ    05h
4935                                     <1>
4936                                     <1> ;The wait for controller to issue completion interrupt is 10 seconds.
4937                                     <1> ;10,000,000 / 30 = 333,333. 333,333 decimal = 051615h
4938                                     <1> WAIT_HDU_INT_LO      equ    1615h
4939                                     <1> WAIT_HDU_INT_HI      equ    05h
4940                                     <1>
4941                                     <1> ;The wait for Data request on read and write longs is
4942                                     <1> ;2000 us. (?)
4943                                     <1> WAIT_HDU_DRQ_LO      equ    1000    ; 03E8h
4944                                     <1> WAIT_HDU_DRQ_HI      equ    0
4945                                     <1>

```

```

4946 <1> ; Port 61h (PORT_B)
4947 <1> SYS1 equ 61h ; PORT_B (diskette.inc)
4948 <1>
4949 <1> ; 23/12/2014
4950 <1> %define CMD_BLOCK BP-8
4951 <1>
4952 <1>
4953 <1> ; << diskette.inc >>
4954 <1> ; ++++++
4955 <1> ;
4956 <1> ;-----
4957 <1> ; ROM BIOS DATA AREAS :
4958 <1> ;-----
4959 <1>
4960 <1> ;DATA SEGMENT AT 40H ; ADDRESS= 0040:0000
4961 <1>
4962 <1> ;-----
4963 <1> ; FIXED DISK DATA AREAS :
4964 <1> ;-----
4965 <1>
4966 0000176B 00 <1> DISK_STATUS1: DB 0 ; FIXED DISK STATUS
4967 0000176C 00 <1> HF_NUM: DB 0 ; COUNT OF FIXED DISK DRIVES
4968 0000176D 00 <1> CONTROL_BYTE: DB 0 ; HEAD CONTROL BYTE
4969 <1> ;@PORT_OFF DB ? ; RESERVED (PORT OFFSET)
4970 <1>
4971 <1> ;-----
4972 <1> ; ADDITIONAL MEDIA DATA :
4973 <1> ;-----
4974 <1>
4975 <1> ;@LAstrate DB ? ; LAST DISKETTE DATA RATE SELECTED
4976 0000176E 00 <1> HF_STATUS DB 0 ; STATUS REGISTER
4977 0000176F 00 <1> HF_ERROR DB 0 ; ERROR REGISTER
4978 00001770 00 <1> HF_INT_FLAG DB 0 ; FIXED DISK INTERRUPT FLAG
4979 00001771 00 <1> HF_CNTRL DB 0 ; COMBO FIXED DISK/DISKETTE CARD BIT 0=1
4980 <1> ;@DSK_STATE DB ? ; DRIVE 0 MEDIA STATE
4981 <1> ; DB ? ; DRIVE 1 MEDIA STATE
4982 <1> ; DB ? ; DRIVE 0 OPERATION START STATE
4983 <1> ; DB ? ; DRIVE 1 OPERATION START STATE
4984 <1> ;@DSK_TRK DB ? ; DRIVE 0 PRESENT CYLINDER
4985 <1> ; DB ? ; DRIVE 1 PRESENT CYLINDER
4986 <1>
4987 <1> ;DATA ENDS ; END OF BIOS DATA SEGMENT
4988 <1> ;
4989 <1> ; ++++++
4990 <1>
4991 <1>
4992 <1> ;--- INT 13H -----
4993 <1> ;
4994 <1> ; FIXED DISK I/O INTERFACE :

```

```

4995      <1> ;
4996      <1> ; THIS INTERFACE PROVIDES ACCESS TO 5 1/4" FIXED DISKS THROUGH :
4997      <1> ; THE IBM FIXED DISK CONTROLLER. :
4998      <1> ;
4999      <1> ; THE BIOS ROUTINES ARE MEANT TO BE ACCESSED THROUGH :
5000      <1> ; SOFTWARE INTERRUPTS ONLY. ANY ADDRESSES PRESENT IN :
5001      <1> ; THESE LISTINGS ARE INCLUDED ONLY FOR COMPLETENESS, :
5002      <1> ; NOT FOR REFERENCE. APPLICATIONS WHICH REFERENCE ANY :
5003      <1> ; ABSOLUTE ADDRESSES WITHIN THE CODE SEGMENTS OF BIOS :
5004      <1> ; VIOLATE THE STRUCTURE AND DESIGN OF BIOS. :
5005      <1> ;
5006      <1> ;-----:
5007      <1> ;
5008      <1> ; INPUT (AH)= HEX COMMAND VALUE :
5009      <1> ;
5010      <1> ; (AH)= 00H RESET DISK (DL = 80H,81H) / DISKETTE :
5011      <1> ; (AH)= 01H READ THE STATUS OF THE LAST DISK OPERATION INTO (AL) :
5012      <1> ; NOTE: DL < 80H - DISKETTE :
5013      <1> ; DL > 80H - DISK :
5014      <1> ; (AH)= 02H READ THE DESIRED SECTORS INTO MEMORY :
5015      <1> ; (AH)= 03H WRITE THE DESIRED SECTORS FROM MEMORY :
5016      <1> ; (AH)= 04H VERIFY THE DESIRED SECTORS :
5017      <1> ; (AH)= 05H FORMAT THE DESIRED TRACK :
5018      <1> ; (AH)= 06H UNUSED :
5019      <1> ; (AH)= 07H UNUSED :
5020      <1> ; (AH)= 08H RETURN THE CURRENT DRIVE PARAMETERS :
5021      <1> ; (AH)= 09H INITIALIZE DRIVE PAIR CHARACTERISTICS :
5022      <1> ; INTERRUPT 41 POINTS TO DATA BLOCK FOR DRIVE 0 :
5023      <1> ; INTERRUPT 46 POINTS TO DATA BLOCK FOR DRIVE 1 :
5024      <1> ; (AH)= 0AH READ LONG :
5025      <1> ; (AH)= 0BH WRITE LONG (READ & WRITE LONG ENCOMPASS 512 + 4 BYTES ECC) :
5026      <1> ; (AH)= 0CH SEEK :
5027      <1> ; (AH)= 0DH ALTERNATE DISK RESET (SEE DL) :
5028      <1> ; (AH)= 0EH UNUSED :
5029      <1> ; (AH)= 0FH UNUSED :
5030      <1> ; (AH)= 10H TEST DRIVE READY :
5031      <1> ; (AH)= 11H RECALIBRATE :
5032      <1> ; (AH)= 12H UNUSED :
5033      <1> ; (AH)= 13H UNUSED :
5034      <1> ; (AH)= 14H CONTROLLER INTERNAL DIAGNOSTIC :
5035      <1> ; (AH)= 15H READ DASD TYPE :
5036      <1> ;
5037      <1> ;-----:
5038      <1> ;
5039      <1> ; REGISTERS USED FOR FIXED DISK OPERATIONS :
5040      <1> ;
5041      <1> ; (DL) - DRIVE NUMBER (80H-81H FOR DISK. VALUE CHECKED) :
5042      <1> ; (DH) - HEAD NUMBER (0-15 ALLOWED, NOT VALUE CHECKED) :
5043      <1> ; (CH) - CYLINDER NUMBER (0-1023, NOT VALUE CHECKED)(SEE CL):

```

```

5044      <1> ;          (CL)  -  SECTOR NUMBER      (1-17, NOT VALUE CHECKED)      :
5045      <1> ;          :
5046      <1> ;          NOTE: HIGH 2 BITS OF CYLINDER NUMBER ARE PLACED      :
5047      <1> ;          IN THE HIGH 2 BITS OF THE CL REGISTER      :
5048      <1> ;          (10 BITS TOTAL)      :
5049      <1> ;          :
5050      <1> ;          (AL)  -  NUMBER OF SECTORS (MAXIMUM POSSIBLE RANGE 1-80H,      :
5051      <1> ;          FOR READ/WRITE LONG 1-79H)      :
5052      <1> ;          :
5053      <1> ;          (ES:BX) - ADDRESS OF BUFFER FOR READS AND WRITES,      :
5054      <1> ;          (NOT REQUIRED FOR VERIFY)      :
5055      <1> ;          :
5056      <1> ;          FORMAT (AH=5) ES:BX POINTS TO A 512 BYTE BUFFER. THE FIRST      :
5057      <1> ;          2*(SECTORS/TRACK) BYTES CONTAIN F,N FOR EACH SECTOR.:
5058      <1> ;          F = 00H FOR A GOOD SECTOR      :
5059      <1> ;          80H FOR A BAD SECTOR      :
5060      <1> ;          N = SECTOR NUMBER      :
5061      <1> ;          FOR AN INTERLEAVE OF 2 AND 17 SECTORS/TRACK      :
5062      <1> ;          THE TABLE SHOULD BE:      :
5063      <1> ;          :
5064      <1> ;          DB    00H,01H,00H,0AH,00H,02H,00H,0BH,00H,03H,00H,0CH      :
5065      <1> ;          DB    00H,04H,00H,0DH,00H,05H,00H,0EH,00H,06H,00H,0FH      :
5066      <1> ;          DB    00H,07H,00H,10H,00H,08H,00H,11H,00H,09H      :
5067      <1> ;          :
5068      <1> ; -----
5069      <1> ;
5070      <1> ; -----
5071      <1> ; OUTPUT      :
5072      <1> ; AH = STATUS OF CURRENT OPERATION      :
5073      <1> ; STATUS BITS ARE DEFINED IN THE EQUATES BELOW      :
5074      <1> ; CY = 0 SUCCESSFUL OPERATION (AH=0 ON RETURN)      :
5075      <1> ; CY = 1 FAILED OPERATION (AH HAS ERROR REASON)      :
5076      <1> ;          :
5077      <1> ; NOTE: ERROR 11H INDICATES THAT THE DATA READ HAD A RECOVERABLE      :
5078      <1> ; ERROR WHICH WAS CORRECTED BY THE ECC ALGORITHM. THE DATA      :
5079      <1> ; IS PROBABLY GOOD, HOWEVER THE BIOS ROUTINE INDICATES AN      :
5080      <1> ; ERROR TO ALLOW THE CONTROLLING PROGRAM A CHANCE TO DECIDE      :
5081      <1> ; FOR ITSELF. THE ERROR MAY NOT RECUR IF THE DATA IS      :
5082      <1> ; REWRITTEN.      :
5083      <1> ;          :
5084      <1> ; IF DRIVE PARAMETERS WERE REQUESTED (DL >= 80H),      :
5085      <1> ; INPUT:      :
5086      <1> ; (DL) = DRIVE NUMBER      :
5087      <1> ; OUTPUT:      :
5088      <1> ; (DL) = NUMBER OF CONSECUTIVE ACKNOWLEDGING DRIVES ATTACHED (1-2) :
5089      <1> ; (CONTROLLER CARD ZERO TALLY ONLY)      :
5090      <1> ; (DH) = MAXIMUM USEABLE VALUE FOR HEAD NUMBER      :
5091      <1> ; (CH) = MAXIMUM USEABLE VALUE FOR CYLINDER NUMBER      :
5092      <1> ; (CL) = MAXIMUM USEABLE VALUE FOR SECTOR NUMBER      :

```

```

5093      <1> ;                AND CYLINDER NUMBER HIGH BITS                :
5094      <1> ;                :
5095      <1> ;      IF READ DASD TYPE WAS REQUESTED,                :
5096      <1> ;                :
5097      <1> ;      AH = 0 - NOT PRESENT                :
5098      <1> ;                1 - DISKETTE - NO CHANGE LINE AVAILABLE        :
5099      <1> ;                2 - DISKETTE - CHANGE LINE AVAILABLE          :
5100      <1> ;                3 - FIXED DISK                :
5101      <1> ;                :
5102      <1> ;      CX,DX = NUMBER OF 512 BYTE BLOCKS WHEN AH = 3        :
5103      <1> ;                :
5104      <1> ;      REGISTERS WILL BE PRESERVED EXCEPT WHEN THEY ARE USED TO RETURN :
5105      <1> ;      INFORMATION.                :
5106      <1> ;                :
5107      <1> ;      NOTE: IF AN ERROR IS REPORTED BY THE DISK CODE, THE APPROPRIATE :
5108      <1> ;      ACTION IS TO RESET THE DISK, THEN RETRY THE OPERATION.    :
5109      <1> ;                :
5110      <1> ; -----
5111      <1> ;
5112      <1> SENSE_FAIL EQU 0FFH ; NOT IMPLEMENTED
5113      <1> NO_ERR EQU 0E0H ; STATUS ERROR/ERROR REGISTER=0
5114      <1> WRITE_FAULT EQU 0CCH ; WRITE FAULT ON SELECTED DRIVE
5115      <1> UNDEF_ERR EQU 0BBH ; UNDEFINED ERROR OCCURRED
5116      <1> NOT_RDY EQU 0AAH ; DRIVE NOT READY
5117      <1> TIME_OUT EQU 80H ; ATTACHMENT FAILED TO RESPOND
5118      <1> BAD_SEEK EQU 40H ; SEEK OPERATION FAILED
5119      <1> BAD_CNTLRL EQU 20H ; CONTROLLER HAS FAILED
5120      <1> DATA_CORRECTED EQU 11H ; ECC CORRECTED DATA ERROR
5121      <1> BAD_ECC EQU 10H ; BAD ECC ON DISK READ
5122      <1> BAD_TRACK EQU 0BH ; NOT IMPLEMENTED
5123      <1> BAD_SECTOR EQU 0AH ; BAD SECTOR FLAG DETECTED
5124      <1> ;DMA_BOUNDARY EQU 09H ; DATA EXTENDS TOO FAR
5125      <1> INIT_FAIL EQU 07H ; DRIVE PARAMETER ACTIVITY FAILED
5126      <1> BAD_RESET EQU 05H ; RESET FAILED
5127      <1> ;RECORD_NOT_FND EQU 04H ; REQUESTED SECTOR NOT FOUND
5128      <1> ;BAD_ADDR_MARK EQU 02H ; ADDRESS MARK NOT FOUND
5129      <1> ;BAD_CMD EQU 01H ; BAD COMMAND PASSED TO DISK I/O
5130      <1> ;
5131      <1> ; -----
5132      <1> ;
5133      <1> ; FIXED DISK PARAMETER TABLE :
5134      <1> ; - THE TABLE IS COMPOSED OF A BLOCK DEFINED AS: :
5135      <1> ; :
5136      <1> ; +0 (1 WORD) - MAXIMUM NUMBER OF CYLINDERS :
5137      <1> ; +2 (1 BYTE) - MAXIMUM NUMBER OF HEADS :
5138      <1> ; +3 (1 WORD) - NOT USED/SEE PC-XT :
5139      <1> ; +5 (1 WORD) - STARTING WRITE PRECOMPENSATION CYL :
5140      <1> ; +7 (1 BYTE) - MAXIMUM ECC DATA BURST LENGTH :
5141      <1> ; +8 (1 BYTE) - CONTROL BYTE :

```

```

5142 <1> ;          BIT 7 DISABLE RETRIES -OR- :
5143 <1> ;          BIT 6 DISABLE RETRIES :
5144 <1> ;          BIT 3 MORE THAN 8 HEADS :
5145 <1> ; +9 (3 BYTES)- NOT USED/SEE PC-XT :
5146 <1> ; +12 (1 WORD) - LANDING ZONE :
5147 <1> ; +14 (1 BYTE) - NUMBER OF SECTORS/TRACK :
5148 <1> ; +15 (1 BYTE) - RESERVED FOR FUTURE USE :
5149 <1> ; :
5150 <1> ; - TO DYNAMICALLY DEFINE A SET OF PARAMETERS :
5151 <1> ; BUILD A TABLE FOR UP TO 15 TYPES AND PLACE :
5152 <1> ; THE CORRESPONDING VECTOR INTO INTERRUPT 41 :
5153 <1> ; FOR DRIVE 0 AND INTERRUPT 46 FOR DRIVE 1. :
5154 <1> ; :
5155 <1> ; -----
5156 <1> ;
5157 <1> ; -----
5158 <1> ; :
5159 <1> ; HARDWARE SPECIFIC VALUES :
5160 <1> ; :
5161 <1> ; - CONTROLLER I/O PORT :
5162 <1> ; :
5163 <1> ; > WHEN READ FROM: :
5164 <1> ; HF_PORT+0 - READ DATA (FROM CONTROLLER TO CPU) :
5165 <1> ; HF_PORT+1 - GET ERROR REGISTER :
5166 <1> ; HF_PORT+2 - GET SECTOR COUNT :
5167 <1> ; HF_PORT+3 - GET SECTOR NUMBER :
5168 <1> ; HF_PORT+4 - GET CYLINDER LOW :
5169 <1> ; HF_PORT+5 - GET CYLINDER HIGH (2 BITS) :
5170 <1> ; HF_PORT+6 - GET SIZE/DRIVE/HEAD :
5171 <1> ; HF_PORT+7 - GET STATUS REGISTER :
5172 <1> ; :
5173 <1> ; > WHEN WRITTEN TO: :
5174 <1> ; HF_PORT+0 - WRITE DATA (FROM CPU TO CONTROLLER) :
5175 <1> ; HF_PORT+1 - SET PRECOMPENSATION CYLINDER :
5176 <1> ; HF_PORT+2 - SET SECTOR COUNT :
5177 <1> ; HF_PORT+3 - SET SECTOR NUMBER :
5178 <1> ; HF_PORT+4 - SET CYLINDER LOW :
5179 <1> ; HF_PORT+5 - SET CYLINDER HIGH (2 BITS) :
5180 <1> ; HF_PORT+6 - SET SIZE/DRIVE/HEAD :
5181 <1> ; HF_PORT+7 - SET COMMAND REGISTER :
5182 <1> ; :
5183 <1> ; -----
5184 <1> ;
5185 <1> ; HF_PORT EQU 01F0H ; DISK PORT
5186 <1> ; HF1_PORT equ 0170h
5187 <1> ; HF_REG_PORT EQU 03F6H
5188 <1> ; HF1_REG_PORT equ 0376h
5189 <1> ;
5190 <1> align 2

```

```

5191 <1>
5192 00001772 F001 <1> HF_PORT: dw 1F0h ; Default = 1F0h
5193 <1> ; (170h)
5194 00001774 F603 <1> HF_REG_PORT: dw 3F6h ; HF_PORT + 206h
5195 <1>
5196 <1> HDC1_BASEPORT equ 1F0h
5197 <1> HDC2_BASEPORT equ 170h
5198 <1>
5199 <1> ; 05/01/2015
5200 00001776 00 <1> hf_m_s: db 0 ; (0 = Master, 1 = Slave)
5201 <1>
5202 00001777 90 <1> align 2
5203 <1>
5204 <1> ;----- STATUS REGISTER
5205 <1>
5206 <1> ST_ERROR EQU 00000001B ;
5207 <1> ST_INDEX EQU 00000010B ;
5208 <1> ST_CORRCTD EQU 00000100B ; ECC CORRECTION SUCCESSFUL
5209 <1> ST_DRQ EQU 00001000B ;
5210 <1> ST_SEEK_COMPL EQU 00010000B ; SEEK COMPLETE
5211 <1> ST_WRT_FLT EQU 00100000B ; WRITE FAULT
5212 <1> ST_READY EQU 01000000B ;
5213 <1> ST_BUSY EQU 10000000B ;
5214 <1>
5215 <1> ;----- ERROR REGISTER
5216 <1>
5217 <1> ERR_DAM EQU 00000001B ; DATA ADDRESS MARK NOT FOUND
5218 <1> ERR_TRK_0 EQU 00000010B ; TRACK 0 NOT FOUND ON RECAL
5219 <1> ERR_ABORT EQU 00000100B ; ABORTED COMMAND
5220 <1> ; EQU 00001000B ; NOT USED
5221 <1> ERR_ID EQU 00010000B ; ID NOT FOUND
5222 <1> ; EQU 00100000B ; NOT USED
5223 <1> ERR_DATA_ECC EQU 01000000B
5224 <1> ERR_BAD_BLOCK EQU 10000000B
5225 <1>
5226 <1>
5227 <1> RECAL_CMD EQU 00010000B ; DRIVE RECAL (10H)
5228 <1> READ_CMD EQU 00100000B ; READ (20H)
5229 <1> WRITE_CMD EQU 00110000B ; WRITE (30H)
5230 <1> VERIFY_CMD EQU 01000000B ; VERIFY (40H)
5231 <1> FMTTRK_CMD EQU 01010000B ; FORMAT TRACK (50H)
5232 <1> INIT_CMD EQU 01100000B ; INITIALIZE (60H)
5233 <1> SEEK_CMD EQU 01110000B ; SEEK (70H)
5234 <1> DIAG_CMD EQU 10010000B ; DIAGNOSTIC (90H)
5235 <1> SET_PARM_CMD EQU 10010001B ; DRIVE PARMS (91H)
5236 <1> NO_RETRIES EQU 00000001B ; CHD MODIFIER (01H)
5237 <1> ECC_MODE EQU 00000010B ; CMD MODIFIER (02H)
5238 <1> BUFFER_MODE EQU 00001000B ; CMD MODIFIER (08H)
5239 <1>

```

```

5240 <1> ;MAX_FILE EQU 2
5241 <1> ;S_MAX_FILE EQU 2
5242 <1> MAX_FILE equ 4 ; 22/12/2014
5243 <1> S_MAX_FILE equ 4 ; 22/12/2014
5244 <1>
5245 <1> DELAY_1 EQU 25H ; DELAY FOR OPERATION COMPLETE
5246 <1> DELAY_2 EQU 0600H ; DELAY FOR READY
5247 <1> DELAY_3 EQU 0100H ; DELAY FOR DATA REQUEST
5248 <1>
5249 <1> HF_FAIL EQU 08H ; CMOS FLAG IN BYTE 0EH
5250 <1>
5251 <1> ;----- COMMAND BLOCK REFERENCE
5252 <1>
5253 <1> ;CMD_BLOCK EQU BP-8 ; @CMD_BLOCK REFERENCES BLOCK HEAD IN SS
5254 <1> ; (BP) POINTS TO COMMAND BLOCK TAIL
5255 <1> ; AS DEFINED BY THE "ENTER" PARMS
5256 <1> ; 19/12/2014
5257 <1> ORG_VECTOR equ 4*13h ; INT 13h vector
5258 <1> DISK_VECTOR equ 4*40h ; INT 40h vector (for floppy disks)
5259 <1> ;HDISK_INT equ 4*76h ; Primary HDC - Hardware interrupt (IRQ14)
5260 <1> HDISK_INT1 equ 4*76h ; Primary HDC - Hardware interrupt (IRQ14)
5261 <1> HDISK_INT2 equ 4*77h ; Secondary HDC - Hardware interrupt (IRQ15)
5262 <1> ;HF_TBL_VEC equ 4*41h ; Pointer to 1st fixed disk parameter table
5263 <1> ;HF1_TBL_VEC equ 4*46h ; Pointer to 2nd fixed disk parameter table
5264 <1> ;
5265 <1> ;HF_TBL_VEC: dd 0 ; Primary master disk param. tbl. pointer
5266 <1> ;HF1_TBL_VEC: dd 0 ; Primary slave disk param. tbl. pointer
5267 <1> HF_TBL_VEC: ; 22/12/2014
5268 00001778 00000000 <1> HDPM_TBL_VEC: dd 0 ; Primary master disk param. tbl. pointer
5269 0000177C 00000000 <1> HDPS_TBL_VEC: dd 0 ; Primary slave disk param. tbl. pointer
5270 00001780 00000000 <1> HDPM_TBL_VEC: dd 0 ; Secondary master disk param. tbl. pointer
5271 00001784 00000000 <1> HDSS_TBL_VEC: dd 0 ; Secondary slave disk param. tbl. pointer
5272 <1>
5273 <1> ;-----
5274 <1> ; FIXED DISK DATA AREAS :
5275 <1> ;-----
5276 <1>
5277 <1> ;@DISK_STATUS1 DB ? ; FIXED DISK STATUS
5278 <1> ;@HF_NUM DB ? ; COUNT OF FIXED DISK DRIVES
5279 <1> ;@CONTROL_BYTE DB ? ; HEAD CONTROL BYTE
5280 <1> ;@PORT_OFF DB ? ; RESERVED (PORT OFFSET)
5281 <1> ;
5282 <1> ;port1_off db 0 ; Hard disk controller 1 - port offset
5283 <1> ;port2_off db 0 ; Hard idsk controller 2 - port offset
5284 <1>
5285 <1> align 2
5286 <1>
5287 <1> ;-----
5288 <1> ; FIXED DISK I/O SETUP :

```

```

5289      <1> ;
5290      <1> ; - ESTABLISH TRANSFER VECTORS FOR THE FIXED DISK :
5291      <1> ; - PERFORM POWER ON DIAGNOSTICS :
5292      <1> ; SHOULD AN ERROR OCCUR A "1701" MESSAGE IS DISPLAYED :
5293      <1> ;
5294      <1> ;-----
5295      <1>
5296      <1> DISK_SETUP:
5297 00001788 FA <1> CLI
5298      <1> ;MOV AX,ABS0 ; GET ABSOLUTE SEGMENT
5299 00001789 31C0 <1> xor ax,ax
5300 0000178B 8ED8 <1> MOV DS,AX ; SET SEGMENT REGISTER
5301 0000178D A14C00 <1> MOV AX, word [ORG_VECTOR] ; GET DISKETTE VECTOR
5302 00001790 A30001 <1> MOV word [DISK_VECTOR],AX ; INTO INT 40H
5303 00001793 A14E00 <1> MOV AX, word [ORG_VECTOR+2]
5304 00001796 A30201 <1> MOV word [DISK_VECTOR+2],AX
5305 00001799 C7064C00[0A18] <1> MOV word [ORG_VECTOR],DISK_IO ; FIXED DISK HANDLER
5306 0000179F 8C0E4E00 <1> MOV word [ORG_VECTOR+2],CS
5307      <1> ; 1st controller (primary master, slave) - IRQ 14
5308      <1> ;MOV word [HDISK_INT],HD_INT ; FIXED DISK INTERRUPT
5309 000017A3 C706D801[511E] <1> mov word [HDISK_INT1],HD_INT ;
5310      <1> ;MOV word [HDISK_INT+2],CS
5311 000017A9 8C0EDA01 <1> mov word [HDISK_INT1+2],CS
5312      <1> ; 2nd controller (secondary master, slave) - IRQ 15
5313 000017AD C706DC01[6A1E] <1> mov word [HDISK_INT2],HD1_INT ;
5314 000017B3 8C0EDE01 <1> mov word [HDISK_INT2+2],CS
5315      <1> ;
5316      <1> ;MOV word [HF_TBL_VEC],HD0_DPT ; PARM TABLE DRIVE 80
5317      <1> ;MOV word [HF_TBL_VEC+2],DPT_SEGM
5318      <1> ;MOV word [HF1_TBL_VEC],HD1_DPT ; PARM TABLE DRIVE 81
5319      <1> ;MOV word [HF1_TBL_VEC+2],DPT_SEGM
5320 000017B7 0E <1> push cs
5321 000017B8 1F <1> pop ds
5322 000017B9 C706[7817]0000 <1> mov word [HDP_M_TBL_VEC],HD0_DPT ; PARM TABLE DRIVE 80h
5323 000017BF C706[7A17]0090 <1> mov word [HDP_M_TBL_VEC+2],DPT_SEGM
5324 000017C5 C706[7C17]2000 <1> mov word [HDPS_TBL_VEC],HD1_DPT ; PARM TABLE DRIVE 81h
5325 000017CB C706[7E17]0090 <1> mov word [HDPS_TBL_VEC+2],DPT_SEGM
5326 000017D1 C706[8017]4000 <1> mov word [HDSM_TBL_VEC],HD2_DPT ; PARM TABLE DRIVE 82h
5327 000017D7 C706[8217]0090 <1> mov word [HDSM_TBL_VEC+2],DPT_SEGM
5328 000017DD C706[8417]6000 <1> mov word [HDSS_TBL_VEC],HD3_DPT ; PARM TABLE DRIVE 83h
5329 000017E3 C706[8617]0090 <1> mov word [HDSS_TBL_VEC+2],DPT_SEGM
5330      <1> ;
5331      <1> ; IN AL,INTB01 ; TURN ON SECOND INTERRUPT CHIP
5332      <1> ; AND AL,0BFH
5333      <1> ; and al, 3Fh ; enable IRQ 14 and IRQ 15
5334      <1> ; JMP $+2
5335      <1> ; IODELAY
5336      <1> ; OUT INTB01,AL
5337      <1> ; IODELAY

```

```

5338          <1> ;      IN      AL,INTA01          ; LET INTERRUPTS PASS THRU TO
5339          <1> ;      AND      AL,0FBH          ; SECOND CHIP
5340          <1> ;      ;JMP      $+2
5341          <1> ;      IODELAY
5342          <1> ;      OUT      INTA01,AL
5343          <1> ;
5344 000017E9 FB          <1>      STI
5345          <1> ;      ;PUSH    DS          ; MOVE ABS0 POINTER TO
5346          <1> ;      ;POP      ES          ; EXTRA SEGMENT POINTER
5347          <1> ;      ;CALL    DDS          ; ESTABLISH DATA SEGMENT
5348          <1> ;      MOV      byte [DISK_STATUS1],0      ; RESET THE STATUS INDICATOR
5349          <1> ;      MOV      byte [HF_NUM],0          ; ZERO NUMBER OF FIXED DISKS
5350          <1> ;      MOV      byte [CONTROL_BYTE],0
5351          <1> ;      ;MOV      byte [PORT_OFF],0      ; ZERO CARD OFFSET
5352          <1> ;      ; 20/12/2014 - private code by Erdogan Tan
5353          <1> ;      ; (out of original PC-AT, PC-XT BIOS code)
5354 000017EA BE[3A29]    <1>      mov     si, hde_type
5355 000017ED B90400      <1>      mov     cx, 4
5356          <1> hde_l:
5357          <1>      lodsb
5358 000017F1 3C80        <1>      cmp     al, 80h          ; 8?h = existing
5359 000017F3 7206        <1>      jb      short _L4
5360 000017F5 FE06[6C17]  <1>      inc     byte [HF_NUM]      ; + 1 hard (fixed) disk drives
5361 000017F9 E2F5        <1>      loop    hde_l
5362          <1> _L4:          ; 0 <= [HF_NUM] =< 4
5363          <1> ;_L4:
5364          <1> ;
5365          <1> ; 31/12/2014 - cancel controller diagnostics here
5366          <1> ;      ;mov     cx, 3      ; 26/12/2014 (Award BIOS 1999)
5367          <1> ;      mov     cl, 3
5368          <1> ;
5369          <1> ;      MOV      DL,80H          ; CHECK THE CONTROLLER
5370          <1> ;hdc_dl:
5371          <1> ;      MOV      AH,14H          ; USE CONTROLLER DIAGNOSTIC COMMAND
5372          <1> ;      INT      13H          ; CALL BIOS WITH DIAGNOSTIC COMMAND
5373          <1> ;      ;JC      short CTL_ERRX      ; DISPLAY ERROR MESSAGE IF BAD RETURN
5374          <1> ;      ;jc      short POD_DONE ;22/12/2014
5375          <1> ;      jnc      short hdc_reset0
5376          <1> ;      loop    hdc_dl
5377          <1> ;      ; 27/12/2014
5378          <1> ;      stc
5379          <1> ;      retn
5380          <1> ;
5381          <1> ;hdc_reset0:
5382          <1> ;      ; 18/01/2015
5383 000017FB 8A0E[6C17]  <1>      mov     cl, [HF_NUM]
5384          <1> ;      ;and     cl, cl
5385          <1> ;      ;jz      short POD_DONE
5386          <1> ;

```

5387	000017FF B27F	<1>	mov	dl, 7Fh	
5388		<1>	hdc_reset1:		
5389	00001801 FEC2	<1>	inc	dl	
5390		<1>		; 31/12/2015	
5391		<1>	push	dx	
5392		<1>	push	cx	
5393		<1>	push	ds	
5394		<1>	sub	ax, ax	
5395		<1>	mov	ds, ax	
5396		<1>	MOV	AX, [TIMER_LOW]	; GET START TIMER COUNTS
5397		<1>	pop	ds	
5398		<1>	MOV	BX, AX	
5399		<1>	ADD	AX, 6*182	; 60 SECONDS* 18.2
5400		<1>	MOV	CX, AX	
5401		<1>	mov	word [wait_count], 0	; 22/12/2014 (reset wait counter)
5402		<1>			
5403		<1>		; 31/12/2014 - cancel HD_RESET_1	
5404		<1>	CALL	HD_RESET_1	; SET UP DRIVE 0, (1,2,3)
5405		<1>	pop	cx	
5406		<1>	pop	dx	
5407		<1>			
5408		<1>		; 18/01/2015	
5409	00001803 B40D	<1>	mov	ah, 0Dh	; ALTERNATE RESET
5410	00001805 CD13	<1>	int	13h	
5411		<1>			
5412	00001807 E2F8	<1>	loop	hdc_reset1	
5413		<1>	POD_DONE:		
5414	00001809 C3	<1>	RETn		
5415		<1>			
5416		<1>	;	-----	POD_ERROR
5417		<1>			
5418		<1>	;	CTL_ERRX:	
5419		<1>	MOV	SI, OFFSET F1782	; CONTROLLER ERROR
5420		<1>	CALL	SET_FAIL	; DO NOT IPL FROM DISK
5421		<1>	CALL	E_MSG	; DISPLAY ERROR AND SET (BP) ERROR FLAG
5422		<1>	JMP	short POD_DONE	
5423		<1>			
5424		<1>	;	HD_RESET_1:	
5425		<1>	;	PUSH BX	; SAVE TIMER LIMITS
5426		<1>	;	PUSH CX	
5427		<1>	;	RES_1: MOV AH, 09H	; SET DRIVE PARAMETERS
5428		<1>	;	INT 13H	
5429		<1>	;	JC short RES_2	
5430		<1>	;	MOV AH, 11H	; RECALIBRATE DRIVE
5431		<1>	;	INT 13H	
5432		<1>	;	JNC short RES_CHK	; DRIVE OK
5433		<1>	;	RES_2: CALL POD_TCHK	; CHECK TIME OUT
5434		<1>	;	cmp word [wait_count], 6*182	; waiting time (in timer ticks)
5435		<1>	;		; (30 seconds)

```

5436      <1> ;      ;cmc
5437      <1> ;      ;JNC short RES_1
5438      <1> ;      ;jb short RES_1
5439      <1> ;;RES_FL: ;MOV SI,OFFSET F1781 ; INDICATE DISK 1 FAILURE;
5440      <1> ;      ;TEST DL,1
5441      <1> ;      ;JNZ RES_E1
5442      <1> ;      ;MOV SI,OFFSET F1780 ; INDICATE DISK 0 FAILURE
5443      <1> ;      ;CALL SET_FAIL ; DO NOT TRY TO IPL DISK 0
5444      <1> ;      ;JMP SHORT RES_E1
5445      <1> ;RES_ER: ; 22/12/2014
5446      <1> ;RES_OK:
5447      <1> ;      ;POP CX ; RESTORE TIMER LIMITS
5448      <1> ;      ;POP BX
5449      <1> ;      ;RETn
5450      <1> ;
5451      <1> ;RES_RS: MOV AH,00H ; RESET THE DRIVE
5452      <1> ;      ;INT 13H
5453      <1> ;RES_CK: MOV AH,08H ; GET MAX CYLINDER,HEAD,SECTOR
5454      <1> ;      ;MOV BL,DL ; SAVE DRIVE CODE
5455      <1> ;      ;INT 13H
5456      <1> ;      ;JC short RES_ER
5457      <1> ;      ;MOV word [NEC_STATUS],CX ; SAVE MAX CYLINDER, SECTOR
5458      <1> ;      ;MOV DL,BL ; RESTORE DRIVE CODE
5459      <1> ;RES_3: MOV AX,0401H ; VERIFY THE LAST SECTOR
5460      <1> ;      ;INT 13H
5461      <1> ;      ;JNC short RES_OK ; VERIFY OK
5462      <1> ;      ;CMP AH,BAD_SECTOR ; OK ALSO IF JUST ID READ
5463      <1> ;      ;JE short RES_OK
5464      <1> ;      ;CMP AH,DATA_CORRECTED
5465      <1> ;      ;JE short RES_OK
5466      <1> ;      ;CMP AH,BAD_ECC
5467      <1> ;      ;JE short RES_OK
5468      <1> ;      ;CALL POD_TCHK ; CHECK FOR TIME OUT
5469      <1> ;      ;cmp word [wait_count], 6*182 ; waiting time (in timer ticks)
5470      <1> ;      ; (60 seconds)
5471      <1> ;      ;cmc
5472      <1> ;      ;JC short RES_ER ; FAILED
5473      <1> ;      ;MOV CX,word [NEC_STATUS] ; GET SECTOR ADDRESS, AND CYLINDER
5474      <1> ;      ;MOV AL,CL ; SEPARATE OUT SECTOR NUMBER
5475      <1> ;      ;AND AL,3FH
5476      <1> ;      ;DEC AL ; TRY PREVIOUS ONE
5477      <1> ;      ;JZ short RES_RS ; WE'VE TRIED ALL SECTORS ON TRACK
5478      <1> ;      ;AND CL,0C0H ; KEEP CYLINDER BITS
5479      <1> ;      ;OR CL,AL ; MERGE SECTOR WITH CYLINDER BITS
5480      <1> ;      ;MOV word [NEC_STATUS],CX ; SAVE CYLINDER, NEW SECTOR NUMBER
5481      <1> ;      ;JMP short RES_3 ; TRY AGAIN
5482      <1> ;;RES_ER: MOV SI,OFFSET F1791 ; INDICATE DISK 1 ERROR
5483      <1> ;      ;TEST DL,1
5484      <1> ;      ;JNZ short RES_E1

```

```

5485      <1> ;      ;MOV    SI,OFFSET F1790      ; INDICATE DISK 0 ERROR
5486      <1> ;;RES_E1:
5487      <1> ;      ;CALL    E_MSG              ; DISPLAY ERROR AND SET (BP) ERROR FLAG
5488      <1> ;;RES_OK:
5489      <1> ;      ;POP     CX                  ; RESTORE TIMER LIMITS
5490      <1> ;      ;POP     BX
5491      <1> ;      ;RETn
5492      <1>
5493      <1> ;SET_FAIL:
5494      <1> ;MOV     AX,X*(CMOS_DIAG+NMI)        ; GET CMOS ERROR BYTE
5495      <1> ;CALL    CMOS_READ
5496      <1> ;OR      AL,HF_FAIL                  ; SET DO NOT IPL FROM DISK FLAG
5497      <1> ;XCHG    AH,AL                      ; SAVE IT
5498      <1> ;CALL    CMOS_WRITE                 ; PUT IT OUT
5499      <1> ;RETn
5500      <1>
5501      <1> ;POD_TCHK:                          ; CHECK FOR 30 SECOND TIME OUT
5502      <1> ;POP     AX                        ; SAVE RETURN
5503      <1> ;POP     CX                        ; GET TIME OUT LIMITS
5504      <1> ;POP     BX
5505      <1> ;PUSH    BX                        ; AND SAVE THEM AGAIN
5506      <1> ;PUSH    CX
5507      <1> ;PUSH    AX
5508      <1> ;push    ds
5509      <1> ;xor     ax, ax
5510      <1> ;mov     ds, ax                    ; RESTORE RETURN
5511      <1> ;MOV     AX, [TIMER_LOW]           ; AX = CURRENT TIME
5512      <1> ;                                           ; BX = START TIME
5513      <1> ;                                           ; CX = END TIME
5514      <1> ;pop     ds
5515      <1> ;CMP     BX,CX
5516      <1> ;JB      short TCHK1              ; START < END
5517      <1> ;CMP     BX,AX
5518      <1> ;JB      short TCHKG              ; END < START < CURRENT
5519      <1> ;JMP     SHORT TCHK2              ; END, CURRENT < START
5520      <1> ;TCHK1:   CMP     AX,BX
5521      <1> ;      ;JB      short TCHKNG       ; CURRENT < START < END
5522      <1> ;TCHK2:   CMP     AX,CX
5523      <1> ;      ;JB      short TCHKG       ; START < CURRENT < END
5524      <1> ;      ;OR      CURRENT < END < START
5525      <1> ;TCHKNG: STC                      ; CARRY SET INDICATES TIME OUT
5526      <1> ;      RETn
5527      <1> ;TCHKG:   CLC                      ; INDICATE STILL TIME
5528      <1> ;      RETn
5529      <1>
5530      <1> ;int_13h:
5531      <1>
5532      <1> ;-----
5533      <1> ;      FIXED DISK BIOS ENTRY POINT      :

```

```

5534 <1> ;-----
5535 <1>
5536 <1> DISK_IO:
5537 0000180A 80FA80 <1> CMP DL,80H ; TEST FOR FIXED DISK DRIVE
5538 0000180D 7305 <1> JAE short A1 ; YES, HANDLE HERE
5539 0000180F CD40 <1> INT 40H ; DISKETTE HANDLER
5540 <1> RET_2:
5541 00001811 CA0200 <1> RETf 2 ; BACK TO CALLER
5542 <1> A1:
5543 00001814 FB <1> STI ; ENABLE INTERRUPTS
5544 <1> ;; 04/01/2015
5545 <1> ;; OR AH,AH
5546 <1> ;; JNZ short A2
5547 <1> ;; INT 40H ; RESET NEC WHEN AH=0
5548 <1> ;; SUB AH,AH
5549 00001815 80FA83 <1> CMP DL,(80H + S_MAX_FILE - 1)
5550 00001818 77F7 <1> JA short RET_2
5551 <1> ; 18/01/2015
5552 <1> ; 04/01/2015
5553 0000181A 08E4 <1> or ah,ah
5554 0000181C 7425 <1> jz short A4
5555 0000181E 80FC0D <1> cmp ah, 0Dh ; Alternate reset
5556 00001821 7504 <1> jne short A2
5557 00001823 28E4 <1> sub ah,ah ; Reset
5558 00001825 EB1C <1> jmp short A4
5559 <1> A2:
5560 00001827 80FC08 <1> CMP AH,08H ; GET PARAMETERS IS A SPECIAL CASE
5561 <1> ;JNZ short A3
5562 <1> ;JMP GET_PARM_N
5563 0000182A 0F848702 <1> je GET_PARM_N
5564 0000182E 80FC15 <1> A3: CMP AH,15H ; READ DASD TYPE IS ALSO
5565 <1> ;JNZ short A4
5566 <1> ;JMP READ_DASD_TYPE
5567 00001831 0F844102 <1> je READ_DASD_TYPE
5568 <1> ; 02/02/2015
5569 00001835 80FC1D <1> cmp ah, 1Dh ;(Temporary for Retro UNIX 386 v1)
5570 <1> ; 12/01/2015
5571 00001838 F5 <1> cmc
5572 00001839 7308 <1> jnc short A4
5573 <1> ; 30/01/2015
5574 0000183B 2EC606[6B17]01 <1> mov byte [CS:DISK_STATUS1],BAD_CMD ; COMMAND ERROR
5575 00001841 EBCE <1> jmp short RET_2
5576 <1> A4: ; SAVE REGISTERS DURING OPERATION
5577 00001843 C8080000 <1> ENTER 8,0 ; SAVE (BP) AND MAKE ROOM FOR @CMD_BLOCK
5578 00001847 53 <1> PUSH BX ; IN THE STACK, THE COMMAND BLOCK IS:
5579 00001848 51 <1> PUSH CX ; @CMD_BLOCK == BYTE PTR [BP]-8
5580 00001849 52 <1> PUSH DX
5581 0000184A 1E <1> PUSH DS
5582 0000184B 06 <1> PUSH ES

```

```

5583 0000184C 56      <1>      PUSH    SI
5584 0000184D 57      <1>      PUSH    DI
5585                    <1>      ;;04/01/2015
5586                    <1>      ;;OR     AH,AH          ; CHECK FOR RESET
5587                    <1>      ;;JNZ   short A5
5588                    <1>      ;;MOV   DL,80H          ; FORCE DRIVE 80 FOR RESET
5589                    <1>      ;;A5:
5590 0000184E 0E      <1>      push    cs
5591 0000184F 1F      <1>      pop     ds
5592 00001850 E84D00  <1>      CALL   DISK_IO_CONT      ; PERFORM THE OPERATION
5593                    <1>      ;CALL   DDS          ; ESTABLISH SEGMENT
5594 00001853 8A26[6B17] <1>      MOV     AH,[DISK_STATUS1] ; GET STATUS FROM OPERATION
5595 00001857 80FC01  <1>      CMP     AH,1          ; SET THE CARRY FLAG TO INDICATE
5596 0000185A F5      <1>      CMC          ; SUCCESS OR FAILURE
5597 0000185B 5F      <1>      POP     DI          ; RESTORE REGISTERS
5598 0000185C 5E      <1>      POP     SI
5599 0000185D 07      <1>      POP     ES
5600 0000185E 1F      <1>      POP     DS
5601 0000185F 5A      <1>      POP     DX
5602 00001860 59      <1>      POP     CX
5603 00001861 5B      <1>      POP     BX
5604 00001862 C9      <1>      LEAVE          ; ADJUST (SP) AND RESTORE (BP)
5605 00001863 CA0200  <1>      RETf     2          ; THROW AWAY SAVED FLAGS
5606                    <1>
5607                    <1>      M1:          ; FUNCTION TRANSFER TABLE
5608 00001866 [DE19]  <1>      DW      DISK_RESET      ; 000H
5609 00001868 [381A]  <1>      DW      RETURN_STATUS ; 001H
5610 0000186A [411A]  <1>      DW      DISK_READ      ; 002H
5611 0000186C [481A]  <1>      DW      DISK_WRITE     ; 003H
5612 0000186E [4F1A]  <1>      DW      DISK_VERF      ; 004H
5613 00001870 [611A]  <1>      DW      FMT_TRK        ; 005H
5614 00001872 [D619]  <1>      DW      BAD_COMMAND     ; 006H FORMAT BAD SECTORS
5615 00001874 [D619]  <1>      DW      BAD_COMMAND     ; 007H FORMAT DRIVE
5616 00001876 [D619]  <1>      DW      BAD_COMMAND     ; 008H RETURN PARAMETERS
5617 00001878 [0E1B]  <1>      DW      INIT_DRV        ; 009H
5618 0000187A [611B]  <1>      DW      RD_LONG        ; 00AH
5619 0000187C [681B]  <1>      DW      WR_LONG        ; 00BH
5620 0000187E [6F1B]  <1>      DW      DISK_SEEK     ; 00CH
5621 00001880 [DE19]  <1>      DW      DISK_RESET     ; 00DH
5622 00001882 [D619]  <1>      DW      BAD_COMMAND     ; 00EH READ BUFFER
5623 00001884 [D619]  <1>      DW      BAD_COMMAND     ; 00FH WRITE BUFFER
5624 00001886 [8D1B]  <1>      DW      TST_RDY        ; 010H
5625 00001888 [A81B]  <1>      DW      HDISK_RECAL     ; 011H
5626 0000188A [D619]  <1>      DW      BAD_COMMAND     ; 012H MEMORY DIAGNOSTIC
5627 0000188C [D619]  <1>      DW      BAD_COMMAND     ; 013H DRIVE DIAGNOSTIC
5628 0000188E [D01B]  <1>      DW      CTLR_DIAGNOSTIC ; 014H CONTROLLER DIAGNOSTIC
5629                    <1>      ; 02/02/2015 (Temporary - Retro UNIX 386 v1 - DISK I/O test)
5630 00001890 [D619]  <1>      dw      BAD_COMMAND     ; 015h
5631 00001892 [D619]  <1>      dw      BAD_COMMAND     ; 016h

```

5632 00001894 [D619]	<1>	dw	BAD_COMMAND	; 017h
5633 00001896 [D619]	<1>	dw	BAD_COMMAND	; 018h
5634 00001898 [D619]	<1>	dw	BAD_COMMAND	; 019h
5635 0000189A [D619]	<1>	dw	BAD_COMMAND	; 01Ah
5636 0000189C [411A]	<1>	dw	DISK_READ	; 01Bh ; LBA read
5637 0000189E [481A]	<1>	dw	DISK_WRITE	; 01Ch ; LBA write
5638	<1>	M1L	EQU	\$-M1
5639	<1>			
5640	<1>		DISK_IO_CONT:	
5641	<1>		;CALL DDS	; ESTABLISH SEGMENT
5642 000018A0 80FC01	<1>		CMP AH,01H	; RETURN STATUS
5643	<1>		;JNZ short SU0	
5644	<1>		;JMP RETURN_STATUS	
5645 000018A3 0F849101	<1>		je RETURN_STATUS	
5646	<1>	SU0:		
5647 000018A7 C606[6B17]00	<1>		MOV byte [DISK_STATUS1],0	; RESET THE STATUS INDICATOR
5648	<1>		;;PUSH BX	; SAVE DATA ADDRESS
5649 000018AC 89DE	<1>		mov si, bx ;;	14/02/2015
5650 000018AE 8A1E[6C17]	<1>		MOV BL,[HF_NUM]	; GET NUMBER OF DRIVES
5651	<1>		;; 04/01/2015	
5652	<1>		;;PUSH AX	
5653 000018B2 80E27F	<1>		AND DL,7FH	; GET DRIVE AS 0 OR 1
5654	<1>			; (get drive number as 0 to 3)
5655 000018B5 38D3	<1>		CMP BL,DL	
5656	<1>		;;JBE BAD_COMMAND_POP	; INVALID DRIVE
5657 000018B7 0F861B01	<1>		jbe BAD_COMMAND ;;	14/02/2015
5658	<1>		;	
5659	<1>		;;03/01/2015	
5660 000018BB 88D3	<1>		mov bl, dl	
5661 000018BD 28FF	<1>		sub bh, bh	
5662 000018BF 883E[4D29]	<1>		mov [LBAMode], bh	; 0
5663 000018C3 F687[3A29]01	<1>		test byte [bx+hd0_type], 1	; LBA ready ?
5664 000018C8 7404	<1>		jz short su1	; no
5665 000018CA FE06[4D29]	<1>		inc byte [LBAMode]	
5666	<1>	su1:		
5667	<1>		;04/01/2015	
5668 000018CE 50	<1>		push ax ;	***
5669 000018CF 06	<1>		PUSH ES ;	**
5670 000018D0 52	<1>		PUSH DX ;	*
5671 000018D1 50	<1>		push ax	
5672 000018D2 E86A05	<1>		CALL GET_VEC	; GET DISK PARAMETERS
5673	<1>		; 02/02/2015	
5674 000018D5 268B4710	<1>		mov ax, [ES:BX+16]	; I/O port base address (1F0h, 170h)
5675 000018D9 A3[7217]	<1>		mov [HF_PORT], ax	
5676 000018DC 268B5712	<1>		mov dx, [ES:BX+18]	; control port address (3F6h, 376h)
5677 000018E0 8916[7417]	<1>		mov [HF_REG_PORT], dx	
5678 000018E4 268A4714	<1>		mov al, [ES:BX+20]	; head register upper nibble (A0h,B0h,E0h,F0h)
5679 000018E8 C0E804	<1>		shr al, 4	
5680 000018EB 2401	<1>		and al, 1	

```

5681 000018ED A2[7617]      <1>      mov     [hf_m_s], al
5682                        <1>      ;
5683                        <1>      ; 03/01/2015
5684 000018F0 268A4708      <1>      MOV     AL,byte [ES:BX+8] ; GET CONTROL BYTE MODIFIER
5685                        <1>      ;MOV    DX,[HF_REG_PORT] ; Device Control register
5686 000018F4 EE            <1>      OUT     DX,AL ; SET EXTRA HEAD OPTION
5687                        <1>      ; Control Byte: (= 08h, here)
5688                        <1>      ; bit 0 - 0
5689                        <1>      ; bit 1 - nIEN (1 = disable irq)
5690                        <1>      ; bit 2 - SRST (software RESET)
5691                        <1>      ; bit 3 - use extra heads (8 to 15)
5692                        <1>      ; -always set to 1-
5693                        <1>      ; (bits 3 to 7 are reserved)
5694                        <1>      ; for ATA devices)
5695 000018F5 8A26[6D17]      <1>      MOV     AH,[CONTROL_BYTE] ; SET EXTRA HEAD OPTION IN
5696 000018F9 80E4C0          <1>      AND     AH,0C0H ; CONTROL BYTE
5697 000018FC 08C4           <1>      OR      AH,AL
5698 000018FE 8826[6D17]      <1>      MOV     [CONTROL_BYTE],AH
5699                        <1>      ; 04/01/2015
5700 00001902 58            <1>      pop     ax
5701 00001903 5A            <1>      pop     dx ; * ;; 14/02/2015
5702 00001904 20E4          <1>      and     ah, ah ; Reset function ?
5703 00001906 7505          <1>      jnz     short su2
5704                        <1>      ;;pop dx ; * ;; 14/02/2015
5705 00001908 07            <1>      pop     es ; **
5706 00001909 58            <1>      pop     ax ; ***
5707                        <1>      ;;pop bx
5708 0000190A E9D100         <1>      jmp     DISK_RESET
5709                        <1>      su2:
5710 0000190D 803E[4D29]00    <1>      cmp     byte [LBAMode], 0
5711 00001912 7669          <1>      jna     short su3
5712                        <1>      ;
5713                        <1>      ; 02/02/2015 (LBA read/write function calls)
5714 00001914 80FC1B         <1>      cmp     ah, 1Bh
5715 00001917 720F          <1>      jb      short lbarw1
5716 00001919 80FC1C         <1>      cmp     ah, 1Ch
5717 0000191C 7764          <1>      ja      short invldfnc
5718                        <1>      ;;pop dx ; * ; 14/02/2015
5719 0000191E 89C8          <1>      mov     ax, cx ; lower word of LBA address (bits 0-15)
5720                        <1>      ;; 14/02/2015
5721 00001920 88D1          <1>      mov     cl, dl ; 14/02/2015
5722                        <1>      ;;mov dx, bx
5723 00001922 89F2          <1>      mov     dx, si ; higher word of LBA address (bits 16-23)
5724                        <1>      ;;mov bx, di
5725 00001924 89FE          <1>      mov     si, di ; Buffer offset
5726 00001926 EB35          <1>      jmp     short lbarw2
5727                        <1>      lbarw1:
5728                        <1>      ; convert CHS to LBA
5729                        <1>      ;

```

```

5730          <1>      ; LBA calculation - AWARD BIOS - 1999 - AHDSK.ASM
5731          <1>      ; LBA = "# of Heads" * Sectors/Track * Cylinder + Head * Sectors/Track
5732          <1>      ;      + Sector - 1
5733 00001928 52      <1>      push    dx ; * ;; 14/02/2015
5734 00001929 30F6    <1>      xor     dh, dh
5735 0000192B 268A570E <1>      mov     dl, [ES:BX+14] ; sectors per track (logical)
5736 0000192F 30E4    <1>      xor     ah, ah
5737 00001931 268A4702 <1>      mov     al, [ES:BX+2]; heads (logical)
5738 00001935 FEC8    <1>      dec     al
5739 00001937 40      <1>      inc     ax ; 0 = 256
5740 00001938 F7E2    <1>      mul     dx
5741          <1>      ; AX = # of Heads" * Sectors/Track
5742 0000193A 89CA    <1>      mov     dx, cx
5743 0000193C 83E13F  <1>      and     cx, 3Fh ; sector (1 to 63)
5744 0000193F 86D6    <1>      xchg    dl, dh
5745 00001941 C0EE06  <1>      shr     dh, 6
5746          <1>      ; DX = cylinder (0 to 1023)
5747 00001944 F7E2    <1>      mul     dx
5748          <1>      ; DX:AX = # of Heads" * Sectors/Track * Cylinder
5749 00001946 FEC9    <1>      dec     cl ; sector - 1
5750 00001948 01C8    <1>      add     ax, cx
5751 0000194A 83D200  <1>      adc     dx, 0
5752          <1>      ; DX:AX = # of Heads" * Sectors/Track * Cylinder + Sector -1
5753 0000194D 59      <1>      pop     cx ; * ; ch = head, cl = drive number (zero based)
5754 0000194E 52      <1>      push    dx
5755 0000194F 50      <1>      push    ax
5756 00001950 268A470E <1>      mov     al, [ES:BX+14] ; sectors per track (logical)
5757 00001954 F6E5    <1>      mul     ch
5758          <1>      ; AX = Head * Sectors/Track
5759 00001956 5A      <1>      pop     dx
5760 00001957 01D0    <1>      add     ax, dx
5761 00001959 5A      <1>      pop     dx
5762 0000195A 83D200  <1>      adc     dx, 0 ; add carry bit
5763          <1>      lbarw2:
5764 0000195D C646F800 <1>      mov     byte [CMD_BLOCK], 0 ; Features Register
5765          <1>      ; NOTE: Features register (1F1h, 171h)
5766          <1>      ; is not used for ATA device R/W functions.
5767          <1>      ; It is old/obsolete 'write precompensation'
5768          <1>      ; register and error register
5769          <1>      ; for old ATA/IDE devices.
5770          <1>      ; 18/01/2014
5771 00001961 8A2E[7617] <1>      mov     ch, [hf_m_s] ; Drive 0 (master) or 1 (slave)
5772 00001965 C0E504  <1>      shl     ch, 4 ; bit 4 (drive bit)
5773 00001968 80CDE0  <1>      or      ch, 0E0h ; bit 5 = 1
5774          <1>      ; bit 6 = 1 = LBA mode
5775          <1>      ; bit 7 = 1
5776 0000196B 80E60F  <1>      and     dh, 0Fh ; LBA byte 4 (bits 24 to 27)
5777 0000196E 08EE    <1>      or      dh, ch
5778          <1>      ;mov    [CMD_BLOCK+2], al ; LBA byte 1 (bits 0 to 7)

```

```

5779          <1>          ; (Sector Number Register)
5780          <1>          ;mov  [CMD_BLOCK+3], ah ; LBA byte 2 (bits 8 to 15)
5781          <1>          ; (Cylinder Low Register)
5782 00001970 8946FA      <1>          mov  [CMD_BLOCK+2], ax ; LBA byte 1, 2
5783 00001973 8856FC      <1>          mov  [CMD_BLOCK+4], dl ; LBA byte 3 (bits 16 to 23)
5784          <1>          ; (Cylinder High Register)
5785          <1>          ;mov  [CMD_BLOCK+5], dh ; LBA byte 4 (bits 24 to 27)
5786          <1>          ; (Drive/Head Register)
5787 00001976 8956FC      <1>          mov  [CMD_BLOCK+4], dx ; LBA byte 4, LBA & DEV select bits
5788          <1>          ;14/02/2015
5789 00001979 88CA        <1>          mov  dl, cl ; Drive number (INIT_DRV)
5790 0000197B EB35        <1>          jmp   short su4
5791          <1> su3:
5792          <1>          ; 02/02/2015
5793          <1>          ; (Temporary functions 1Bh & 1Ch are not valid for CHS mode)
5794 0000197D 80FC14      <1>          cmp   ah, 14h
5795 00001980 7604        <1>          jna   short chsfnc
5796          <1> invldfnc:
5797          <1>          ; 14/02/2015
5798 00001982 07          <1>          pop   es ; **
5799 00001983 58          <1>          pop   ax ; ***
5800          <1>          ;jmp   short BAD_COMMAND_POP
5801 00001984 EB50        <1>          jmp   short BAD_COMMAND
5802          <1> chsfnc:
5803 00001986 268B4705     <1>          MOV    AX,word [ES:BX+5] ; GET WRITE PRE-COMPENSATION CYLINDER
5804 0000198A C1E802      <1>          SHR    AX,2
5805 0000198D 8846F8      <1>          MOV    [CMD_BLOCK],AL
5806          <1>          ;MOV   AL,byte [ES:BX+8] ; GET CONTROL BYTE MODIFIER
5807          <1>          ;PUSH   DX
5808          <1>          ;MOV   DX,[HF_REG_PORT]
5809          <1>          ;OUT    DX,AL ; SET EXTRA HEAD OPTION
5810          <1>          ;;POP   DX ; * ;; 21/02/2015 (14/02/2015!)
5811          <1>          ;POP   ES ; **
5812          <1>          ;MOV   AH,[CONTROL_BYTE] ; SET EXTRA HEAD OPTION IN
5813          <1>          ;AND   AH,0C0H ; CONTROL BYTE
5814          <1>          ;OR    AH,AL
5815          <1>          ;MOV   [CONTROL_BYTE],AH
5816          <1>          ;
5817 00001990 88C8          <1>          MOV    AL,CL ; GET SECTOR NUMBER
5818 00001992 243F          <1>          AND    AL,3FH
5819 00001994 8846FA      <1>          MOV    [CMD_BLOCK+2],AL
5820 00001997 886EFB      <1>          MOV    [CMD_BLOCK+3],CH ; GET CYLINDER NUMBER
5821 0000199A 88C8          <1>          MOV    AL,CL
5822 0000199C C0E806      <1>          SHR    AL,6
5823 0000199F 8846FC      <1>          MOV    [CMD_BLOCK+4],AL ; CYLINDER HIGH ORDER 2 BITS
5824          <1>          ;;05/01/2015
5825          <1>          ;;MOV   AL,DL ; DRIVE NUMBER
5826 000019A2 A0[7617]    <1>          mov    al, [hf_m_s]
5827 000019A5 C0E004      <1>          SHL    AL,4

```

```

5828 000019A8 80E60F      <1>      AND    DH,0FH          ; HEAD NUMBER
5829 000019AB 08F0        <1>      OR     AL,DH
5830                          <1>      ;OR    AL,80H or 20H
5831 000019AD 0CA0        <1>      OR     AL,80h+20h      ; ECC AND 512 BYTE SECTORS
5832 000019AF 8846FD      <1>      MOV    [CMD_BLOCK+5],AL    ; ECC/SIZE/DRIVE/HEAD
5833                          <1> su4:
5834 000019B2 07          <1>      POP     ES ; **
5835                          <1>      ;; 14/02/2015
5836                          <1>      ;;POP    AX
5837                          <1>      ;;MOV    [CMD_BLOCK+1],AL    ; SECTOR COUNT
5838                          <1>      ;;PUSH   AX
5839                          <1>      ;;MOV    AL,AH          ; GET INTO LOW BYTE
5840                          <1>      ;;XOR    AH,AH          ; ZERO HIGH BYTE
5841                          <1>      ;;SAL    AX,1            ; *2 FOR TABLE LOOKUP
5842 000019B3 5B          <1>      pop     bx ; *** ax
5843 000019B4 885EF9      <1>      mov     [CMD_BLOCK+1], bl
5844 000019B7 88FB        <1>      mov     bl, bh
5845 000019B9 30FF        <1>      xor     bh, bh
5846 000019BB D1E3        <1>      sal     bx, 1
5847                          <1>      ;;MOV    SI,AX          ; PUT INTO SI FOR BRANCH
5848                          <1>      ;;CMP    AX,M1L          ; TEST WITHIN RANGE
5849                          <1>      ;;JNB    short BAD_COMMAND_POP
5850 000019BD 83FB3A      <1>      cmp     bx, M1L
5851 000019C0 7314        <1>      jnb     short BAD_COMMAND
5852 000019C2 87DE        <1>      xchg    bx, si
5853                          <1>      ;;POP    AX          ; RESTORE AX
5854                          <1>      ;;POP    BX          ; AND DATA ADDRESS
5855                          <1>
5856                          <1>      ;PUSH   CX
5857                          <1>      ;PUSH   AX          ; ADJUST ES:BX
5858 000019C4 89D9        <1>      MOV     CX,BX          ; GET 3 HIGH ORDER NIBBLES OF BX
5859 000019C6 C1E904      <1>      SHR     CX,4
5860 000019C9 8CC0        <1>      MOV     AX,ES
5861 000019CB 01C8        <1>      ADD     AX,CX
5862 000019CD 8EC0        <1>      MOV     ES,AX
5863 000019CF 83E30F      <1>      AND     BX,000FH      ; ES:BX CHANGED TO ES:000X
5864                          <1>      ;POP    AX
5865                          <1>      ;POP    CX
5866                          <1>      ;JMP    word [CS:SI+M1]
5867 000019D2 FFA4[6618]   <1>      jmp     word [SI+M1]
5868                          <1>      ;;BAD_COMMAND_POP:
5869                          <1>      ;;      POP    AX
5870                          <1>      ;;      POP    BX
5871                          <1>      BAD_COMMAND:
5872 000019D6 C606[6B17]01 <1>      MOV     byte [DISK_STATUS1],BAD_CMD ; COMMAND ERROR
5873 000019DB B000        <1>      MOV     AL,0
5874 000019DD C3          <1>      RETn
5875                          <1>
5876                          <1> ;-----

```

```

5877          <1> ;      RESET THE DISK SYSTEM (AH=00H) :
5878          <1> ;-----
5879          <1>
5880          <1> ; 18-1-2015 : one controller reset (not other one)
5881          <1>
5882          <1> DISK_RESET:
5883 000019DE FA          <1>      CLI
5884 000019DF E4A1        <1>      IN      AL,INTB01          ; GET THE MASK REGISTER
5885                                <1>      ;JMP      $+2
5886                                <1>      IODELAY
5887 000019E1 EB00        <2>      jmp short $+2
5888 000019E3 EB00        <2>      jmp short $+2
5889                                <1>      ;AND      AL,0BFH          ; ENABLE FIXED DISK INTERRUPT
5890 000019E5 243F        <1>      and      al,3Fh          ; 22/12/2014 (IRQ 14 & IRQ 15)
5891 000019E7 E6A1        <1>      OUT      INTB01,AL
5892 000019E9 FB          <1>      STI          ; START INTERRUPTS
5893                                <1>      ; 14/02/2015
5894 000019EA 89D7        <1>      mov     di, dx
5895                                <1>      ; 04/01/2015
5896                                <1>      ; xor     di,di
5897          <1> drst0:
5898 000019EC B004        <1>      MOV      AL,04H ; bit 2 - SRST
5899                                <1>      ;MOV     DX,HF_REG_PORT
5900 000019EE 8B16[7417] <1>      MOV     DX,[HF_REG_PORT]
5901 000019F2 EE          <1>      OUT      DX,AL          ; RESET
5902                                <1>      ; MOV     CX,10          ; DELAY COUNT
5903                                <1>      ;DRD: DEC     CX
5904                                <1>      ; JNZ     short DRD          ; WAIT 4.8 MICRO-SEC
5905 000019F3 B90200      <1>      mov     cx,2          ; wait for 30 micro seconds
5906 000019F6 E801FD      <1>      call     WAITF          ; (Award Bios 1999 - WAIT_REFRESH,
5907                                <1>                                ; 40 micro seconds)
5908 000019F9 A0[6D17]    <1>      mov     al,[CONTROL_BYTE]
5909 000019FC 240F        <1>      AND      AL,0FH          ; SET HEAD OPTION
5910 000019FE EE          <1>      OUT      DX,AL          ; TURN RESET OFF
5911 000019FF E85103      <1>      CALL     NOT_BUSY
5912 00001A02 750F        <1>      JNZ     short DRERR          ; TIME OUT ON RESET
5913 00001A04 8B16[7217] <1>      MOV     DX,[HF_PORT]
5914 00001A08 FEC2        <1>      inc     dl ; HF_PORT+1
5915                                <1>      ; 02/01/2015 - Award BIOS 1999 - AHDSK.ASM
5916 00001A0A B10A        <1>      mov     cl, 10
5917          <1> drst1:
5918 00001A0C EC          <1>      IN      AL,DX          ; GET RESET STATUS
5919 00001A0D 3C01        <1>      CMP     AL,1
5920                                <1>      ; 04/01/2015
5921 00001A0F 7408        <1>      jz      short drst2
5922                                <1>      ;JNZ     short DRERR          ; BAD RESET STATUS
5923                                <1>      ; Drive/Head Register - bit 4
5924 00001A11 E2F9        <1>      loop    drst1
5925          <1> DRERR:

```

```

5926 00001A13 C606[6B17]05      <1>      MOV     byte [DISK_STATUS1],BAD_RESET ; CARD FAILED
5927 00001A18 C3                  <1>      RETn
5928                               <1> drst2:
5929                               <1>      ; 14/02/2015
5930 00001A19 89FA                <1>      mov     dx,di
5931                               <1> ;drst3:
5932                               <1>      ; 05/01/2015
5933                               <1>      ; shl     di,1
5934                               <1>      ; 04/01/2015
5935                               <1>      ; mov     ax,[di+hd_cports]
5936                               <1>      ; cmp     ax,[HF_REG_PORT]
5937                               <1>      ; je      short drst4
5938                               <1>      ; mov     [HF_REG_PORT], ax
5939                               <1>      ; 03/01/2015
5940                               <1>      ; mov     ax,[di+hd_ports]
5941                               <1>      ; mov     [HF_PORT], ax
5942                               <1>      ; 05/01/2014
5943                               <1>      ; shr     di,1
5944                               <1>      ; 04/01/2015
5945                               <1>      ; jmp     short drst0 ; reset other controller
5946                               <1> ;drst4:
5947                               <1>      ; 05/01/2015
5948                               <1>      ; shr     di,1
5949                               <1>      ; mov     al,[di+hd_dregs]
5950                               <1>      ; and     al,10h ; bit 4 only
5951                               <1>      ; shr     al,4 ; bit 4 -> bit 0
5952                               <1>      ; mov     [hf_m_s], al ; (0 = master, 1 = slave)
5953                               <1>      ;
5954 00001A1B A0[7617]              <1>      mov     al, [hf_m_s] ; 18/01/2015
5955 00001A1E A801                  <1>      test    al,1
5956                               <1>      ; jnz     short drst6
5957 00001A20 7510                  <1>      jnz     short drst4
5958 00001A22 8066FDEF              <1>      AND     byte [CMD_BLOCK+5],0EFH ; SET TO DRIVE 0
5959                               <1> ;drst5:
5960                               <1> drst3:
5961 00001A26 E8E500                <1>      CALL    INIT_DRV           ; SET MAX HEADS
5962                               <1>      ;mov     dx,di
5963 00001A29 E87C01                <1>      CALL    HDISK_RECAL        ; RECAL TO RESET SEEK SPEED
5964                               <1>      ; 04/01/2014
5965                               <1>      ; inc     di
5966                               <1>      ; mov     dx,di
5967                               <1>      ; cmp     dl,[HF_NUM]
5968                               <1>      ; jb      short drst3
5969                               <1> ;DRE:
5970 00001A2C C606[6B17]00          <1>      MOV     byte [DISK_STATUS1],0      ; IGNORE ANY SET UP ERRORS
5971 00001A31 C3                  <1>      RETn
5972                               <1> ;drst6:
5973                               <1> drst4:      ; Drive/Head Register - bit 4
5974 00001A32 804EFD10              <1>      OR      byte [CMD_BLOCK+5],010H ; SET TO DRIVE 1

```

5975		<1>	;jmp	short drst5	
5976	00001A36 EBEE	<1>	jmp	short drst3	
5977		<1>			
5978		<1>	;-----		
5979		<1>	; DISK STATUS ROUTINE (AH = 01H) :		
5980		<1>	;-----		
5981		<1>			
5982		<1>	RETURN_STATUS:		
5983	00001A38 A0[6B17]	<1>	MOV	AL,[DISK_STATUS1]	; OBTAIN PREVIOUS STATUS
5984	00001A3B C606[6B17]00	<1>	MOV	byte [DISK_STATUS1],0	; RESET STATUS
5985	00001A40 C3	<1>	RETn		
5986		<1>			
5987		<1>	;-----		
5988		<1>	; DISK READ ROUTINE (AH = 02H) :		
5989		<1>	;-----		
5990		<1>			
5991		<1>	DISK_READ:		
5992	00001A41 C646FE20	<1>	MOV	byte [CMD_BLOCK+6],READ_CMD	
5993	00001A45 E9CF01	<1>	JMP	COMMANDI	
5994		<1>			
5995		<1>	;-----		
5996		<1>	; DISK WRITE ROUTINE (AH = 03H) :		
5997		<1>	;-----		
5998		<1>			
5999		<1>	DISK_WRITE:		
6000	00001A48 C646FE30	<1>	MOV	byte [CMD_BLOCK+6],WRITE_CMD	
6001	00001A4C E90902	<1>	JMP	COMMANDO	
6002		<1>			
6003		<1>	;-----		
6004		<1>	; DISK VERIFY (AH = 04H) :		
6005		<1>	;-----		
6006		<1>			
6007		<1>	DISK_VERF:		
6008	00001A4F C646FE40	<1>	MOV	byte [CMD_BLOCK+6],VERIFY_CMD	
6009	00001A53 E86002	<1>	CALL	COMMAND	
6010	00001A56 7508	<1>	JNZ	short VERF_EXIT	; CONTROLLER STILL BUSY
6011	00001A58 E8C202	<1>	CALL	_WAIT	; (Original: CALL WAIT)
6012	00001A5B 7503	<1>	JNZ	short VERF_EXIT	; TIME OUT
6013	00001A5D E84503	<1>	CALL	CHECK_STATUS	
6014		<1>	VERF_EXIT:		
6015	00001A60 C3	<1>	RETn		
6016		<1>			
6017		<1>	;-----		
6018		<1>	; FORMATTING (AH = 05H) :		
6019		<1>	;-----		
6020		<1>			
6021		<1>	FMT_TRK: ; FORMAT TRACK (AH = 005H)		
6022	00001A61 C646FE50	<1>	MOV	byte [CMD_BLOCK+6],FMTTRK_CMD	
6023	00001A65 06	<1>	PUSH	ES	

6024	00001A66	53	<1>	PUSH	BX	
6025	00001A67	E8D503	<1>	CALL	GET_VEC	; GET DISK PARAMETERS ADDRESS
6026	00001A6A	268A470E	<1>	MOV	AL,[ES:BX+14]	; GET SECTORS/TRACK
6027	00001A6E	8846F9	<1>	MOV	[CMD_BLOCK+1],AL	; SET SECTOR COUNT IN COMMAND
6028	00001A71	5B	<1>	POP	BX	
6029	00001A72	07	<1>	POP	ES	
6030	00001A73	E9E701	<1>	JMP	CMD_OF	; GO EXECUTE THE COMMAND
6031			<1>			
6032			<1>			
6033			<1>			
6034			<1>			
6035			<1>			
6036			<1>	READ_DASD_TYPE:		
6037			<1>	READ_D_T:		; GET DRIVE PARAMETERS
6038	00001A76	1E	<1>	PUSH	DS	; SAVE REGISTERS
6039	00001A77	06	<1>	PUSH	ES	
6040	00001A78	53	<1>	PUSH	BX	
6041			<1>			
6042	00001A79	0E	<1>	;CALL	DDS	; ESTABLISH ADDRESSING
6043	00001A7A	1F	<1>	push	cs	
6044	00001A7B	C606[6B17]00	<1>	pop	ds	
6045	00001A80	8A1E[6C17]	<1>	MOV	byte [DISK_STATUS1],0	
6046	00001A84	80E27F	<1>	MOV	BL,[HF_NUM]	; GET NUMBER OF DRIVES
6047	00001A87	38D3	<1>	AND	DL,7FH	; GET DRIVE NUMBER
6048	00001A89	7622	<1>	CMP	BL,DL	
6049	00001A8B	E8B103	<1>	JBE	short RDT_NOT_PRESENT	; RETURN DRIVE NOT PRESENT
6050	00001A8E	268A4702	<1>	CALL	GET_VEC	; GET DISK PARAMETER ADDRESS
6051	00001A92	268A4F0E	<1>	MOV	AL,[ES:BX+2]	; HEADS
6052	00001A96	F6E9	<1>	MOV	CL,[ES:BX+14]	
6053	00001A98	268B0F	<1>	IMUL	CL	; * NUMBER OF SECTORS
6054			<1>	MOV	CX,[ES:BX]	; MAX NUMBER OF CYLINDERS
6055			<1>			
6056			<1>			
6057			<1>			
6058	00001A9B	49	<1>			
6059			<1>	DEC	CX	; LEAVE ONE FOR DIAGNOSTICS
6060	00001A9C	F7E9	<1>			
6061	00001A9E	89D1	<1>	IMUL	CX	; NUMBER OF SECTORS
6062	00001AA0	89C2	<1>	MOV	CX,DX	; HIGH ORDER HALF
6063			<1>	MOV	DX,AX	; LOW ORDER HALF
6064	00001AA2	28C0	<1>	;SUB	AX,AX	
6065	00001AA4	B403	<1>	sub	al, al	
6066	00001AA6	5B	<1>	MOV	AH,03H	; INDICATE FIXED DISK
6067	00001AA7	07	<1>	RDT2: POP	BX	; RESTORE REGISTERS
6068	00001AA8	1F	<1>	POP	ES	
6069	00001AA9	F8	<1>	POP	DS	
6070	00001AAA	CA0200	<1>	CLC		; CLEAR CARRY
6071			<1>	RETf	2	
6072	00001AAD	29C0	<1>	RDT_NOT_PRESENT:		
			<1>	SUB	AX,AX	; DRIVE NOT PRESENT RETURN

6073	00001AAF 89C1	<1>	MOV	CX,AX	; ZERO BLOCK COUNT
6074	00001AB1 89C2	<1>	MOV	DX,AX	
6075	00001AB3 EBF1	<1>	JMP	short RDT2	
6076		<1>			
6077		<1>			
6078		<1>			
6079		<1>			
6080		<1>			
6081		<1>	GET_PARM_N:		
6082		<1>	;GET_PARM:		; GET DRIVE PARAMETERS
6083	00001AB5 1E	<1>	PUSH	DS	; SAVE REGISTERS
6084		<1>	;PUSH	ES	
6085	00001AB6 53	<1>	PUSH	BX	
6086		<1>	;MOV	AX,ABS0	; ESTABLISH ADDRESSING
6087		<1>	;MOV	DS,AX	
6088		<1>	;TEST	DL,1	; CHECK FOR DRIVE 1
6089		<1>	;JZ	short G0	
6090		<1>	;LES	BX,@HF1_TBL_VEC	
6091		<1>	;JMP	SHORT G1	
6092		<1>	;G0:	LES	BX,@HF_TBL_VEC
6093		<1>	;G1:		
6094		<1>	;CALL	DDS	; ESTABLISH SEGMENT
6095		<1>		; 22/12/2014	
6096	00001AB7 0E	<1>	push	cs	
6097	00001AB8 1F	<1>	pop	ds	
6098	00001AB9 80EA80	<1>	SUB	DL,80H	
6099	00001ABC 80FA04	<1>	CMP	DL,MAX_FILE	; TEST WITHIN RANGE
6100	00001ABF 733D	<1>	JAE	short G4	
6101		<1>			
6102		<1>		; 22/12/2014	
6103	00001AC1 88D3	<1>	mov	bl, dl	
6104	00001AC3 30FF	<1>	xor	bh, bh	
6105	00001AC5 C0E302	<1>	shl	bl, 2	; convert index to offset
6106	00001AC8 81C3[7817]	<1>	add	bx, HF_TBL_VEC	
6107	00001ACC 8B4702	<1>	mov	ax, [bx+2]	
6108	00001ACF 8EC0	<1>	mov	es, ax	; dpt segment
6109	00001AD1 8B1F	<1>	mov	bx, [bx]	; dpt offset
6110		<1>			
6111	00001AD3 C606[6B17]00	<1>	MOV	byte [DISK_STATUS1],0	
6112	00001AD8 268B07	<1>	MOV	AX,[ES:BX]	; MAX NUMBER OF CYLINDERS
6113		<1>	;SUB	AX,2	; ADJUST FOR 0-N
6114	00001ADB 48	<1>	dec	ax	; max. cylinder number
6115	00001ADC 88C5	<1>	MOV	CH,AL	
6116	00001ADE 250003	<1>	AND	AX,0300H	; HIGH TWO BITS OF CYLINDER
6117	00001AE1 D1E8	<1>	SHR	AX,1	
6118	00001AE3 D1E8	<1>	SHR	AX,1	
6119	00001AE5 260A470E	<1>	OR	AL,[ES:BX+14]	; SECTORS
6120	00001AE9 88C1	<1>	MOV	CL,AL	
6121	00001AEB 268A7702	<1>	MOV	DH,[ES:BX+2]	; HEADS

```

6122 00001AEF FECE          <1>      DEC    DH              ; 0-N RANGE
6123 00001AF1 8A16[6C17]    <1>      MOV    DL,[HF_NUM]          ; DRIVE COUNT
6124 00001AF5 29C0          <1>      SUB    AX,AX
6125                          <1>      ;27/12/2014
6126                          <1>      ; ES:DI = Address of disk parameter table from BIOS
6127                          <1>      ;(Programmer's Guide to the AMIBIOS - 1993)
6128 00001AF7 89DF          <1>      mov    di, bx              ; HDPT offset
6129                          <1>  G5:
6130 00001AF9 5B             <1>      POP    BX              ; RESTORE REGISTERS
6131                          <1>      ;POP    ES
6132 00001AFA 1F             <1>      POP    DS
6133 00001AFB CA0200         <1>      RETf    2
6134                          <1>  G4:
6135 00001AFE C606[6B17]07    <1>      MOV    byte [DISK_STATUS1],INIT_FAIL ; OPERATION FAILED
6136 00001B03 B407          <1>      MOV    AH,INIT_FAIL
6137 00001B05 28C0          <1>      SUB    AL,AL
6138 00001B07 29D2          <1>      SUB    DX,DX
6139 00001B09 29C9          <1>      SUB    CX,CX
6140 00001B0B F9            <1>      STC              ; SET ERROR FLAG
6141 00001B0C EBEB          <1>      JMP    short G5
6142                          <1>
6143                          <1> ;-----
6144                          <1> ;      INITIALIZE DRIVE      (AH = 09H) :
6145                          <1> ;-----
6146                          <1>      ; 03/01/2015
6147                          <1>      ; According to ATA-ATAPI specification v2.0 to v5.0
6148                          <1>      ; logical sector per logical track
6149                          <1>      ; and logical heads - 1 would be set but
6150                          <1>      ; it is seen as it will be good
6151                          <1>      ; if physical parameters will be set here
6152                          <1>      ; because, number of heads <= 16.
6153                          <1>      ; (logical heads usually more than 16)
6154                          <1>      ; NOTE: ATA logical parameters (software C, H, S)
6155                          <1>      ;      == INT 13h physical parameters
6156                          <1>
6157                          <1> ;INIT_DRV:
6158                          <1> ;      MOV    byte [CMD_BLOCK+6],SET_PARM_CMD
6159                          <1> ;      CALL   GET_VEC          ; ES:BX -> PARAMETER BLOCK
6160                          <1> ;      MOV    AL,[ES:BX+2]        ; GET NUMBER OF HEADS
6161                          <1> ;      DEC    AL              ; CONVERT TO 0-INDEX
6162                          <1> ;      MOV    AH,[CMD_BLOCK+5]    ; GET SDH REGISTER
6163                          <1> ;      AND    AH,0F0H          ; CHANGE HEAD NUMBER
6164                          <1> ;      OR     AH,AL             ; TO MAX HEAD
6165                          <1> ;      MOV    [CMD_BLOCK+5],AH
6166                          <1> ;      MOV    AL,[ES:BX+14]        ; MAX SECTOR NUMBER
6167                          <1> ;      MOV    [CMD_BLOCK+1],AL
6168                          <1> ;      SUB    AX,AX
6169                          <1> ;      MOV    [CMD_BLOCK+3],AL      ; ZERO FLAGS
6170                          <1> ;      CALL   COMMAND          ; TELL CONTROLLER

```

6171	<1> ;	JNZ	short INIT_EXIT	; CONTROLLER BUSY ERROR
6172	<1> ;	CALL	NOT_BUSY	; WAIT FOR IT TO BE DONE
6173	<1> ;	JNZ	short INIT_EXIT	; TIME OUT
6174	<1> ;	CALL	CHECK_STATUS	
6175	<1> ;	INIT_EXIT:		
6176	<1> ;	RETn		
6177	<1>			
6178	<1> ;	04/01/2015		
6179	<1> ;	02/01/2015	- Derived from from AWARD BIOS 1999	
6180	<1> ;		AHDSK.ASM - INIT_DRIVE	
6181	<1>	INIT_DRV:		
6182	00001B0E	30E4	<1>	xor ah,ah
6183	00001B10	B00B	<1>	mov al,11 ; Physical heads from translated HDPT
6184	00001B12	3826[4D29]	<1>	cmp [LBAMode], ah ; 0
6185	00001B16	7702	<1>	ja short idrv0
6186	00001B18	B002	<1>	mov al,2 ; Physical heads from standard HDPT
6187	<1>	idrv0:		
6188	<1>			; DL = drive number (0 based)
6189	00001B1A	E82203	<1>	call GET_VEC
6190	00001B1D	53	<1>	push bx
6191	00001B1E	01C3	<1>	add bx,ax
6192	<1>			; 05/01/2015
6193	00001B20	8A26[7617]	<1>	mov ah, [hf_m_s] ; drive number (0= master, 1= slave)
6194	<1>			;and ah,1
6195	00001B24	C0E404	<1>	shl ah,4
6196	00001B27	80CCA0	<1>	or ah,0A0h ; Drive/Head register - 10100000b (A0h)
6197	00001B2A	268A07	<1>	mov al,[es:bx]
6198	00001B2D	FEC8	<1>	dec al ; last head number
6199	<1>			;and al,0Fh
6200	00001B2F	08E0	<1>	or al,ah ; lower 4 bits for head number
6201	<1>			
6202	00001B31	C646FE91	<1>	mov byte [CMD_BLOCK+6],SET_PARM_CMD
6203	00001B35	8846FD	<1>	mov [CMD_BLOCK+5],al
6204	00001B38	5B	<1>	pop bx
6205	00001B39	B004	<1>	mov al,4 ; Physical sec per track from translated HDPT
6206	00001B3B	803E[4D29]00	<1>	cmp byte [LBAMode], 0
6207	00001B40	7702	<1>	ja short idrv1
6208	00001B42	B00E	<1>	mov al,14 ; Physical sec per track from standard HDPT
6209	<1>	idrv1:		
6210	00001B44	30E4	<1>	xor ah,ah
6211	00001B46	01C3	<1>	add bx,ax
6212	00001B48	268A07	<1>	mov al,[es:bx]
6213	<1>			; sector number
6214	00001B4B	8846F9	<1>	mov [CMD_BLOCK+1],al
6215	00001B4E	28C0	<1>	sub al,al
6216	00001B50	8846FB	<1>	mov [CMD_BLOCK+3],al ; ZERO FLAGS
6217	00001B53	E86001	<1>	call COMMAND ; TELL CONTROLLER
6218	00001B56	7508	<1>	jnz short INIT_EXIT ; CONTROLLER BUSY ERROR
6219	00001B58	E8F801	<1>	call NOT_BUSY ; WAIT FOR IT TO BE DONE

6220 00001B5B 7503	<1> jnz short INIT_EXIT ; TIME OUT
6221 00001B5D E84502	<1> call CHECK_STATUS
6222	<1> INIT_EXIT:
6223 00001B60 C3	<1> RETn
6224	<1>
6225	<1> ;-----
6226	<1> ; READ LONG (AH = 0AH) :
6227	<1> ;-----
6228	<1>
6229	<1> RD_LONG:
6230	<1> ;MOV @CMD_BLOCK+6,READ_CMD OR ECC_MODE
6231 00001B61 C646FE22	<1> mov byte [CMD_BLOCK+6],READ_CMD + ECC_MODE
6232 00001B65 E9AF00	<1> JMP COMMANDI
6233	<1>
6234	<1> ;-----
6235	<1> ; WRITE LONG (AH = 0BH) :
6236	<1> ;-----
6237	<1>
6238	<1> WR_LONG:
6239	<1> ;MOV @CMD_BLOCK+6,WRITE_CMD OR ECC_MODE
6240 00001B68 C646FE32	<1> MOV byte [CMD_BLOCK+6],WRITE_CMD + ECC_MODE
6241 00001B6C E9E900	<1> JMP COMMANDO
6242	<1>
6243	<1> ;-----
6244	<1> ; SEEK (AH = 0CH) :
6245	<1> ;-----
6246	<1>
6247	<1> DISK_SEEK:
6248 00001B6F C646FE70	<1> MOV byte [CMD_BLOCK+6],SEEK_CMD
6249 00001B73 E84001	<1> CALL COMMAND
6250 00001B76 7514	<1> JNZ short DS_EXIT ; CONTROLLER BUSY ERROR
6251 00001B78 E8A201	<1> CALL _WAIT
6252 00001B7B 750F	<1> JNZ DS_EXIT ; TIME OUT ON SEEK
6253 00001B7D E82502	<1> CALL CHECK_STATUS
6254 00001B80 803E[6B17]40	<1> CMP byte [DISK_STATUS1],BAD_SEEK
6255 00001B85 7505	<1> JNE short DS_EXIT
6256 00001B87 C606[6B17]00	<1> MOV byte [DISK_STATUS1],0
6257	<1> DS_EXIT:
6258 00001B8C C3	<1> RETn
6259	<1>
6260	<1> ;-----
6261	<1> ; TEST DISK READY (AH = 10H) :
6262	<1> ;-----
6263	<1>
6264	<1> TST_RDY: ; WAIT FOR CONTROLLER
6265 00001B8D E8C301	<1> CALL NOT_BUSY
6266 00001B90 7515	<1> JNZ short TR_EX
6267 00001B92 8A46FD	<1> MOV AL,[CMD_BLOCK+5] ; SELECT DRIVE
6268 00001B95 8B16[7217]	<1> MOV DX,[HF_PORT]

6269	00001B99	80C206	<1>	add	d1,6	
6270	00001B9C	EE	<1>	OUT	DX,AL	
6271	00001B9D	E81702	<1>	CALL	CHECK_ST	; CHECK STATUS ONLY
6272	00001BA0	7505	<1>	JNZ	short TR_EX	
6273	00001BA2	C606[6B17]00	<1>	MOV	byte [DISK_STATUS1],0	; WIPE OUT DATA CORRECTED ERROR
6274			<1>	TR_EX:		
6275	00001BA7	C3	<1>	RETn		
6276			<1>			
6277			<1>	;-----		
6278			<1>	; RECALIBRATE	(AH = 11H) :	
6279			<1>	;-----		
6280			<1>			
6281			<1>	HDISK_RECAL:		
6282	00001BA8	C646FE10	<1>	MOV	byte [CMD_BLOCK+6],RECAL_CMD ; 10h, 16	
6283	00001BAC	E80701	<1>	CALL	COMMAND	; START THE OPERATION
6284	00001BAF	7519	<1>	JNZ	short RECAL_EXIT	; ERROR
6285	00001BB1	E86901	<1>	CALL	_WAIT	; WAIT FOR COMPLETION
6286	00001BB4	7405	<1>	JZ	short RECAL_X	; TIME OUT ONE OK ?
6287	00001BB6	E86401	<1>	CALL	_WAIT	; WAIT FOR COMPLETION LONGER
6288	00001BB9	750F	<1>	JNZ	short RECAL_EXIT	; TIME OUT TWO TIMES IS ERROR
6289			<1>	RECAL_X:		
6290	00001BBB	E8E701	<1>	CALL	CHECK_STATUS	
6291	00001BBE	803E[6B17]40	<1>	CMP	byte [DISK_STATUS1],BAD_SEEK ; SEEK NOT COMPLETE	
6292	00001BC3	7505	<1>	JNE	short RECAL_EXIT	; IS OK
6293	00001BC5	C606[6B17]00	<1>	MOV	byte [DISK_STATUS1],0	
6294			<1>	RECAL_EXIT:		
6295	00001BCA	803E[6B17]00	<1>	CMP	byte [DISK_STATUS1],0	
6296	00001BCF	C3	<1>	RETn		
6297			<1>			
6298			<1>	;-----		
6299			<1>	; CONTROLLER DIAGNOSTIC (AH = 14H) :		
6300			<1>	;-----		
6301			<1>			
6302			<1>	CTLR_DIAGNOSTIC:		
6303	00001BD0	FA	<1>	CLI		; DISABLE INTERRUPTS WHILE CHANGING MASK
6304	00001BD1	E4A1	<1>	IN	AL,INTB01	; TURN ON SECOND INTERRUPT CHIP
6305			<1>	;AND	AL,0BFH	
6306	00001BD3	243F	<1>	and	al, 3Fh	; enable IRQ 14 & IRQ 15
6307			<1>	;JMP	\$+2	
6308			<1>	IODELAY		
6309	00001BD5	EB00	<2>	jmp	short \$+2	
6310	00001BD7	EB00	<2>	jmp	short \$+2	
6311	00001BD9	E6A1	<1>	OUT	INTB01,AL	
6312			<1>	IODELAY		
6313	00001BDB	EB00	<2>	jmp	short \$+2	
6314	00001BDD	EB00	<2>	jmp	short \$+2	
6315	00001BDF	E421	<1>	IN	AL,INTA01	; LET INTERRUPTS PASS THRU TO
6316	00001BE1	24FB	<1>	AND	AL,0FBH	; SECOND CHIP
6317			<1>	;JMP	\$+2	

6318		<1>	IODELAY	
6319	00001BE3 EB00	<2>	jmp short \$+2	
6320	00001BE5 EB00	<2>	jmp short \$+2	
6321	00001BE7 E621	<1>	OUT	INTA01,AL
6322	00001BE9 FB	<1>	STI	
6323	00001BEA E86601	<1>	CALL	NOT_BUSY ; WAIT FOR CARD
6324	00001BED 7521	<1>	JNZ	short CD_ERR ; BAD CARD
6325		<1>	;MOV	DX, HF_PORT+7
6326	00001BEF 8B16[7217]	<1>	mov	dx, [HF_PORT]
6327	00001BF3 80C207	<1>	add	dl, 7
6328	00001BF6 B090	<1>	MOV	AL,DIAG_CMD ; START DIAGNOSE
6329	00001BF8 EE	<1>	OUT	DX,AL
6330	00001BF9 E85701	<1>	CALL	NOT_BUSY ; WAIT FOR IT TO COMPLETE
6331	00001BFC B480	<1>	MOV	AH,TIME_OUT
6332	00001BFE 7512	<1>	JNZ	short CD_EXIT ; TIME OUT ON DIAGNOSTIC
6333		<1>	;MOV	DX,HF_PORT+1 ; GET ERROR REGISTER
6334	00001C00 8B16[7217]	<1>	mov	dx, [HF_PORT]
6335	00001C04 FEC2	<1>	inc	dl
6336	00001C06 EC	<1>	IN	AL,DX
6337	00001C07 A2[6F17]	<1>	MOV	[HF_ERROR],AL ; SAVE IT
6338	00001C0A B400	<1>	MOV	AH,0
6339	00001C0C 3C01	<1>	CMP	AL,1 ; CHECK FOR ALL OK
6340	00001C0E 7402	<1>	JE	SHORT CD_EXIT
6341	00001C10 B420	<1>	CD_ERR: MOV	AH,BAD_CNTL
6342		<1>	CD_EXIT:	
6343	00001C12 8826[6B17]	<1>	MOV	[DISK_STATUS1],AH
6344	00001C16 C3	<1>	RETn	
6345		<1>		
6346		<1>	;-----	
6347		<1>	; COMMANDI :	
6348		<1>	; REPEATEDLY INPUTS DATA TILL :	
6349		<1>	; NSECTOR RETURNS ZERO :	
6350		<1>	;-----	
6351		<1>	COMMANDI:	
6352	00001C17 E80202	<1>	CALL	CHECK_DMA ; CHECK 64K BOUNDARY ERROR
6353	00001C1A 723B	<1>	JC	short CMD_ABORT
6354	00001C1C 89DF	<1>	MOV	DI,BX
6355	00001C1E E89500	<1>	CALL	COMMAND ; OUTPUT COMMAND
6356	00001C21 7534	<1>	JNZ	short CMD_ABORT
6357		<1>	CMD_I1:	
6358	00001C23 E8F700	<1>	CALL	_WAIT ; WAIT FOR DATA REQUEST INTERRUPT
6359	00001C26 752F	<1>	JNZ	short TM_OUT ; TIME OUT
6360	00001C28 B90001	<1>	MOV	CX,256 ; SECTOR SIZE IN WORDS
6361		<1>	;MOV	DX,HF_PORT
6362	00001C2B 8B16[7217]	<1>	mov	dx, [HF_PORT]
6363	00001C2F FA	<1>	CLI	
6364	00001C30 FC	<1>	CLD	
6365	00001C31 F36D	<1>	REP	INSW ; GET THE SECTOR
6366	00001C33 FB	<1>	STI	

6367 00001C34 F646FE02	<1> TEST byte [CMD_BLOCK+6],ECC_MODE ; CHECK FOR NORMAL INPUT
6368 00001C38 7413	<1> JZ CMD_I3
6369 00001C3A E84401	<1> CALL WAIT_DRQ ; WAIT FOR DATA REQUEST
6370 00001C3D 7218	<1> JC short TM_OUT
6371	<1> ;MOV DX,HF_PORT
6372 00001C3F 8B16[7217]	<1> mov dx,[HF_PORT]
6373 00001C43 B90400	<1> MOV CX,4 ; GET ECC BYTES
6374 00001C46 EC	<1> CMD_I2: IN AL,DX
6375 00001C47 268805	<1> MOV [ES:DI],AL ; GO SLOW FOR BOARD
6376 00001C4A 47	<1> INC DI
6377 00001C4B E2F9	<1> LOOP CMD_I2
6378 00001C4D E85501	<1> CMD_I3: CALL CHECK_STATUS
6379 00001C50 7505	<1> JNZ short CMD_ABORT ; ERROR RETURNED
6380 00001C52 FE4EF9	<1> DEC byte [CMD_BLOCK+1] ; CHECK FOR MORE
6381 00001C55 75CC	<1> JNZ SHORT CMD_I1
6382	<1> CMD_ABORT:
6383 00001C57 C3	<1> TM_OUT: RETn
6384	<1>
6385	<1> ;-----
6386	<1> ; COMMANDO :
6387	<1> ; REPEATEDLY OUTPUTS DATA TILL :
6388	<1> ; NSECTOR RETURNS ZERO :
6389	<1> ;-----
6390	<1> COMMANDO:
6391 00001C58 E8C101	<1> CALL CHECK_DMA ; CHECK 64K BOUNDARY ERROR
6392 00001C5B 72FA	<1> JC short CMD_ABORT
6393 00001C5D 89DE	<1> CMD_OF: MOV SI,BX
6394 00001C5F E85400	<1> CALL COMMAND ; OUTPUT COMMAND
6395 00001C62 75F3	<1> JNZ short CMD_ABORT
6396 00001C64 E81A01	<1> CALL WAIT_DRQ ; WAIT FOR DATA REQUEST
6397 00001C67 72EE	<1> JC short TM_OUT ; TOO LONG
6398 00001C69 1E	<1> CMD_01: PUSH DS
6399	<1> ;PUSH ES ; MOVE ES TO DS
6400	<1> ;POP DS
6401	<1> ;MOV CX,256 ; PUT THE DATA OUT TO THE CARD
6402	<1> ;MOV DX,HF_PORT
6403	<1> ; 01/02/2015
6404 00001C6A 8B16[7217]	<1> mov dx,[HF_PORT]
6405 00001C6E 06	<1> push es
6406 00001C6F 1F	<1> pop ds
6407 00001C70 B90001	<1> mov cx, 256
6408 00001C73 FA	<1> CLI
6409 00001C74 FC	<1> CLD
6410 00001C75 F36F	<1> REP OUTSW
6411 00001C77 FB	<1> STI
6412 00001C78 1F	<1> POP DS ; RESTORE DS
6413 00001C79 F646FE02	<1> TEST byte [CMD_BLOCK+6],ECC_MODE ; CHECK FOR NORMAL OUTPUT
6414 00001C7D 7413	<1> JZ short CMD_03
6415 00001C7F E8FF00	<1> CALL WAIT_DRQ ; WAIT FOR DATA REQUEST

6416 00001C82 72D3	<1> JC short TM_OUT	
6417	<1> ;MOV DX,HF_PORT	
6418 00001C84 8B16[7217]	<1> mov dx, [HF_PORT]	
6419 00001C88 B90400	<1> MOV CX,4	; OUTPUT THE ECC BYTES
6420 00001C8B 268A04	<1> CMD_02: MOV AL,[ES:SI]	
6421 00001C8E EE	<1> OUT DX,AL	
6422 00001C8F 46	<1> INC SI	
6423 00001C90 E2F9	<1> LOOP CMD_02	
6424	<1> CMD_03:	
6425 00001C92 E88800	<1> CALL _WAIT	; WAIT FOR SECTOR COMPLETE INTERRUPT
6426 00001C95 75C0	<1> JNZ short TM_OUT	; ERROR RETURNED
6427 00001C97 E80B01	<1> CALL CHECK_STATUS	
6428 00001C9A 75BB	<1> JNZ short CMD_ABORT	
6429 00001C9C F606[6E17]08	<1> TEST byte [HF_STATUS],ST_DRQ	; CHECK FOR MORE
6430 00001CA1 75C6	<1> JNZ SHORT CMD_01	
6431	<1> ;MOV DX,HF_PORT+2	; CHECK RESIDUAL SECTOR COUNT
6432 00001CA3 8B16[7217]	<1> mov dx, [HF_PORT]	
6433	<1> ;add dl, 2	
6434 00001CA7 FEC2	<1> inc dl	
6435 00001CA9 FEC2	<1> inc dl	
6436 00001CAB EC	<1> IN AL,DX	
6437 00001CAC A8FF	<1> TEST AL,0FFH	
6438 00001CAE 7405	<1> JZ short CMD_04	; COUNT = 0 OK
6439 00001CB0 C606[6B17]BB	<1> MOV byte [DISK_STATUS1],UNDEF_ERR	
6440	<1>	; OPERATION ABORTED - PARTIAL TRANSFER
6441	<1> CMD_04:	
6442 00001CB5 C3	<1> RETn	
6443	<1>	
6444	<1> ;-----	
6445	<1> ; COMMAND :	
6446	<1> ; THIS ROUTINE OUTPUTS THE COMMAND BLOCK :	
6447	<1> ; OUTPUT :	
6448	<1> ; BL = STATUS :	
6449	<1> ; BH = ERROR REGISTER :	
6450	<1> ;-----	
6451	<1>	
6452	<1> COMMAND:	
6453 00001CB6 53	<1> PUSH BX	; WAIT FOR SEEK COMPLETE AND READY
6454	<1> ;;MOV CX,DELAY_2	; SET INITIAL DELAY BEFORE TEST
6455	<1> COMMAND1:	
6456	<1> ;;PUSH CX	; SAVE LOOP COUNT
6457 00001CB7 E8D3FE	<1> CALL TST_RDY	; CHECK DRIVE READY
6458	<1> ;;POP CX	
6459 00001CBA 7413	<1> JZ short COMMAND2	; DRIVE IS READY
6460 00001CBC 803E[6B17]80	<1> CMP byte [DISK_STATUS1],TIME_OUT	; TST_RDY TIMED OUT--GIVE UP
6461	<1> ;JZ short CMD_TIMEOUT	
6462	<1> ;;LOOP COMMAND1	; KEEP TRYING FOR A WHILE
6463	<1> ;JMP SHORT COMMAND4	; ITS NOT GOING TO GET READY
6464 00001CC1 7505	<1> jne short COMMAND4	

6465		<1> CMD_TIMEOUT:
6466	00001CC3 C606[6B17]20	<1> MOV byte [DISK_STATUS1],BAD_CNTL
6467		<1> COMMAND4:
6468	00001CC8 5B	<1> POP BX
6469	00001CC9 803E[6B17]00	<1> CMP byte [DISK_STATUS1],0 ; SET CONDITION CODE FOR CALLER
6470	00001CCE C3	<1> RETn
6471		<1> COMMAND2:
6472	00001CCF 5B	<1> POP BX
6473	00001CD0 57	<1> PUSH DI
6474	00001CD1 C606[7017]00	<1> MOV byte [HF_INT_FLAG],0 ; RESET INTERRUPT FLAG
6475	00001CD6 FA	<1> CLI ; INHIBIT INTERRUPTS WHILE CHANGING MASK
6476	00001CD7 E4A1	<1> IN AL,INTB01 ; TURN ON SECOND INTERRUPT CHIP
6477		<1> ;AND AL,0BFH
6478	00001CD9 243F	<1> and al, 3Fh ; Enable IRQ 14 & 15
6479		<1> ;JMP \$+2
6480		<1> IODELAY
6481	00001CDB EB00	<2> jmp short \$+2
6482	00001CDD EB00	<2> jmp short \$+2
6483	00001CDF E6A1	<1> OUT INTB01,AL
6484	00001CE1 E421	<1> IN AL,INTA01 ; LET INTERRUPTS PASS THRU TO
6485	00001CE3 24FB	<1> AND AL,0FBH ; SECOND CHIP
6486		<1> ;JMP \$+2
6487		<1> IODELAY
6488	00001CE5 EB00	<2> jmp short \$+2
6489	00001CE7 EB00	<2> jmp short \$+2
6490	00001CE9 E621	<1> OUT INTA01,AL
6491	00001CEB FB	<1> STI
6492	00001CEC 31FF	<1> XOR DI,DI ; INDEX THE COMMAND TABLE
6493		<1> ;MOV DX,HF_PORT+1 ; DISK ADDRESS
6494	00001CEE 8B16[7217]	<1> mov dx, [HF_PORT]
6495	00001CF2 FEC2	<1> inc dl
6496	00001CF4 F606[6D17]C0	<1> TEST byte [CONTROL_BYTE],0C0H ; CHECK FOR RETRY SUPPRESSION
6497	00001CF9 7411	<1> JZ short COMMAND3
6498	00001CFB 8A46FE	<1> MOV AL, [CMD_BLOCK+6] ; YES-GET OPERATION CODE
6499	00001CFE 24F0	<1> AND AL,0F0H ; GET RID OF MODIFIERS
6500	00001D00 3C20	<1> CMP AL,20H ; 20H-40H IS READ, WRITE, VERIFY
6501	00001D02 7208	<1> JB short COMMAND3
6502	00001D04 3C40	<1> CMP AL,40H
6503	00001D06 7704	<1> JA short COMMAND3
6504	00001D08 804EFE01	<1> OR byte [CMD_BLOCK+6],NO_RETRIES
6505		<1> ; VALID OPERATION FOR RETRY SUPPRESS
6506		<1> COMMAND3:
6507	00001D0C 8A43F8	<1> MOV AL, [CMD_BLOCK+DI] ; GET THE COMMAND STRING BYTE
6508	00001D0F EE	<1> OUT DX,AL ; GIVE IT TO CONTROLLER
6509		<1> IODELAY
6510	00001D10 EB00	<2> jmp short \$+2
6511	00001D12 EB00	<2> jmp short \$+2
6512	00001D14 47	<1> INC DI ; NEXT BYTE IN COMMAND BLOCK
6513	00001D15 42	<1> INC DX ; NEXT DISK ADAPTER REGISTER

6514	00001D16	83FF07	<1>	cmp	di, 7 ; 1/1/2015	; ALL DONE?
6515	00001D19	75F1	<1>	JNZ	short COMMAND3	; NO--GO DO NEXT ONE
6516	00001D1B	5F	<1>	POP	DI	
6517	00001D1C	C3	<1>	RETn		; ZERO FLAG IS SET
6518			<1>			
6519			<1>	;CMD_TIMEOUT:		
6520			<1>	MOV	byte [DISK_STATUS1],BAD_CNTL	
6521			<1>	;COMMAND4:		
6522			<1>	POP	BX	
6523			<1>	CMP	[DISK_STATUS1],0	; SET CONDITION CODE FOR CALLER
6524			<1>	RETn		
6525			<1>			
6526			<1>	;-----		
6527			<1>	WAIT	FOR INTERRUPT	:
6528			<1>	;-----		
6529			<1>	;WAIT:		
6530			<1>	_WAIT:		
6531	00001D1D	FB	<1>	STI		; MAKE SURE INTERRUPTS ARE ON
6532			<1>	SUB	CX,CX	; SET INITIAL DELAY BEFORE TEST
6533			<1>	CLC		
6534			<1>	MOV	AX,9000H	; DEVICE WAIT INTERRUPT
6535			<1>	INT	15H	
6536			<1>	JC	WT2	; DEVICE TIMED OUT
6537			<1>	MOV	BL,DELAY_1	; SET DELAY COUNT
6538			<1>			
6539			<1>	mov	bl, WAIT_HDU_INT_HI	
6540	00001D1E	B306	<1>	mov	bl, WAIT_HDU_INT_HI + 1	
6541	00001D20	B91516	<1>	mov	cx, WAIT_HDU_INT_LO	
6542			<1>			; (AWARD BIOS -> WAIT_FOR_MEM)
6543			<1>	;	-----	WAIT LOOP
6544			<1>			
6545			<1>	WT1:		
6546			<1>	TEST	byte [HF_INT_FLAG],80H	; TEST FOR INTERRUPT
6547	00001D23	F606[7017]C0	<1>	test	byte [HF_INT_FLAG],0C0h	
6548			<1>	;LOOPZ WT1		
6549	00001D28	7519	<1>	JNZ	short WT3	; INTERRUPT--LETS GO
6550			<1>	DEC	BL	
6551			<1>	JNZ	short WT1	; KEEP TRYING FOR A WHILE
6552			<1>			
6553			<1>	WT1_hi:		
6554	00001D2A	E461	<1>	in	al, SYS1 ; 61h (PORT_B)	; wait for lo to hi
6555	00001D2C	A810	<1>	test	al, 10h	; transition on memory
6556	00001D2E	75FA	<1>	jnz	short WT1_hi	; refresh.
6557			<1>	WT1_lo:		
6558	00001D30	E461	<1>	in	al, SYS1	; 061h (PORT_B)
6559	00001D32	A810	<1>	test	al, 10h	
6560	00001D34	74FA	<1>	jz	short WT1_lo	
6561	00001D36	E2EB	<1>	loop	WT1	
6562			<1>	or	bl, bl	

6563		<1>	;jz	short WT2	
6564		<1>	;dec	b1	
6565		<1>	;jmp	short WT1	
6566	00001D38	FECB	<1>	dec	b1
6567	00001D3A	75E7	<1>	jnz	short WT1
6568		<1>			
6569	00001D3C	C606[6B17]80	<1>	WT2: MOV	byte [DISK_STATUS1],TIME_OUT ; REPORT TIME OUT ERROR
6570	00001D41	EB0A	<1>	JMP	SHORT WT4
6571	00001D43	C606[6B17]00	<1>	WT3: MOV	byte [DISK_STATUS1],0
6572	00001D48	C606[7017]00	<1>	MOV	byte [HF_INT_FLAG],0
6573	00001D4D	803E[6B17]00	<1>	WT4: CMP	byte [DISK_STATUS1],0 ; SET CONDITION CODE FOR CALLER
6574	00001D52	C3	<1>	RETn	
6575		<1>			
6576		<1>			
6577		<1>	;-----		
6578		<1>	; WAIT FOR CONTROLLER NOT BUSY	:	
6579		<1>	;-----		
6580	00001D53	FB	<1>	NOT_BUSY:	
6581	00001D54	53	<1>	STI	; MAKE SURE INTERRUPTS ARE ON
6582			<1>	PUSH	BX
6583	00001D55	8B16[7217]	<1>	;SUB	CX,CX ; SET INITIAL DELAY BEFORE TEST
6584	00001D59	80C207	<1>	mov	DX, [HF_PORT]
6585			<1>	add	d1, 7 ; Status port (HF_PORT+7)
6586			<1>	;MOV	BL,DELAY_1
6587	00001D5C	B91516	<1>		; wait for 10 seconds
6588			<1>	mov	cx, WAIT_HDU_INT_LO ; 1615h
6589	00001D5F	B306	<1>	;mov	b1, WAIT_HDU_INT_HI ; 05h
6590			<1>	mov	b1, WAIT_HDU_INT_HI + 1
6591			<1>	;	
6592			<1>	;; mov	byte [wait_count], 0 ; Reset wait counter
6593	00001D61	EC	<1>	NB1:	
6594			<1>	IN	AL,DX ; CHECK STATUS
6595	00001D62	2480	<1>	;TEST	AL,ST_BUSY
6596			<1>	and	al, ST_BUSY
6597	00001D64	7414	<1>	;LOOPNZ	NB1
6598			<1>	JZ	short NB2 ; NOT BUSY--LETS GO
6599			<1>	;DEC	BL
6600			<1>	;JNZ	short NB1 ; KEEP TRYING FOR A WHILE
6601	00001D66	E461	<1>	NB1_hi: IN	AL,SYS1 ; wait for hi to lo
6602	00001D68	A810	<1>	TEST	AL,010H ; transition on memory
6603	00001D6A	75FA	<1>	JNZ	SHORT NB1_hi ; refresh.
6604	00001D6C	E461	<1>	NB1_lo: IN	AL,SYS1
6605	00001D6E	A810	<1>	TEST	AL,010H
6606	00001D70	74FA	<1>	JZ	short NB1_lo
6607	00001D72	E2ED	<1>	LOOP	NB1
6608	00001D74	FECB	<1>	dec	b1
6609	00001D76	75E9	<1>	jnz	short NB1
6610			<1>	;	
6611			<1>	;; cmp	byte [wait_count], 182 ; 10 seconds (182 timer ticks)

6612	<1> ;;	jb	short NB1	
6613	<1>			
6614	<1>	;MOV	[DISK_STATUS1],TIME_OUT	; REPORT TIME OUT ERROR
6615	<1>	;JMP	SHORT NB3	
6616 00001D78 B080	<1>	mov	al, TIME_OUT	
6617	<1> NB2:			
6618	<1>	;MOV	byte [DISK_STATUS1],0	
6619	<1> ;NB3:			
6620 00001D7A 5B	<1>	POP	BX	
6621 00001D7B A2[6B17]	<1>	mov	[DISK_STATUS1], al	;;; will be set after return
6622	<1>	;CMP	byte [DISK_STATUS1],0	; SET CONDITION CODE FOR CALLER
6623 00001D7E 08C0	<1>	or	al, al	; (zf = 0 --> timeout)
6624 00001D80 C3	<1>	RETn		
6625	<1>			
6626	<1>			
6627	<1>	; WAIT FOR DATA REQUEST	:	
6628	<1>			
6629	<1> WAIT_DRQ:			
6630	<1>	;MOV	CX,DELAY_3	
6631	<1>	;MOV	DX,HF_PORT+7	
6632 00001D81 8B16[7217]	<1>	mov	dx, [HF_PORT]	
6633 00001D85 80C207	<1>	add	d1, 7	
6634	<1>	;MOV	bl, WAIT_HDU_DRQ_HI	; 0
6635 00001D88 B9E803	<1>	MOV	cx, WAIT_HDU_DRQ_LO	; 1000 (30 milli seconds)
6636	<1>			; (but it is written as 2000
6637	<1>			; micro seconds in ATORGS.ASM file
6638	<1>			; of Award Bios - 1999, D1A0622)
6639	<1>			
6640 00001D8B EC	<1> WQ_1: IN	AL,DX		; GET STATUS
6641 00001D8C A808	<1>	TEST	AL,ST_DRQ	; WAIT FOR DRQ
6642 00001D8E 7514	<1>	JNZ	short WQ_OK	
6643	<1>	;LOOP	WQ_1	; KEEP TRYING FOR A SHORT WHILE
6644	<1> WQ_hi:			
6645 00001D90 E461	<1>	IN	AL,SYS1	; wait for hi to lo
6646 00001D92 A810	<1>	TEST	AL,010H	; transition on memory
6647 00001D94 75FA	<1>	JNZ	SHORT WQ_hi	; refresh.
6648 00001D96 E461	<1> WQ_lo: IN	AL,SYS1		
6649 00001D98 A810	<1>	TEST	AL,010H	
6650 00001D9A 74FA	<1>	JZ	SHORT WQ_lo	
6651 00001D9C E2ED	<1>	LOOP	WQ_1	
6652	<1>			
6653 00001D9E C606[6B17]80	<1>	MOV	byte [DISK_STATUS1],TIME_OUT	; ERROR
6654 00001DA3 F9	<1>	STC		
6655	<1> WQ_OK:			
6656 00001DA4 C3	<1>	RETn		
6657	<1> ;WQ_OK:		;CLC	
6658	<1>	; RETn		
6659	<1>			
6660	<1>			

6661	<1> ;	CHECK FIXED DISK STATUS :
6662	<1> ;	-----
6663	<1>	CHECK_STATUS:
6664 00001DA5 E80F00	<1>	CALL CHECK_ST ; CHECK THE STATUS BYTE
6665 00001DA8 7507	<1>	JNZ short CHECK_S1 ; AN ERROR WAS FOUND
6666 00001DAA A801	<1>	TEST AL,ST_ERROR ; WERE THERE ANY OTHER ERRORS
6667 00001DAC 7403	<1>	JZ short CHECK_S1 ; NO ERROR REPORTED
6668 00001DAE E83E00	<1>	CALL CHECK_ER ; ERROR REPORTED
6669	<1>	CHECK_S1:
6670 00001DB1 803E[6B17]00	<1>	CMP byte [DISK_STATUS1],0 ; SET STATUS FOR CALLER
6671 00001DB6 C3	<1>	RETn
6672	<1>	
6673	<1> ;	-----
6674	<1> ;	CHECK FIXED DISK STATUS BYTE :
6675	<1> ;	-----
6676	<1>	CHECK_ST:
6677	<1>	;MOV DX,HF_PORT+7 ; GET THE STATUS
6678 00001DB7 8B16[7217]	<1>	mov dx, [HF_PORT]
6679 00001DBB 80C207	<1>	add dl, 7
6680 00001DBE EC	<1>	IN AL,DX
6681 00001DBF A2[6E17]	<1>	MOV [HF_STATUS],AL
6682 00001DC2 B400	<1>	MOV AH,0
6683 00001DC4 A880	<1>	TEST AL,ST_BUSY ; IF STILL BUSY
6684 00001DC6 751A	<1>	JNZ short CKST_EXIT ; REPORT OK
6685 00001DC8 B4CC	<1>	MOV AH,WRITE_FAULT
6686 00001DCA A820	<1>	TEST AL,ST_WRT_FLT ; CHECK FOR WRITE FAULT
6687 00001DCC 7514	<1>	JNZ short CKST_EXIT
6688 00001DCE B4AA	<1>	MOV AH,NOT_RDY
6689 00001DD0 A840	<1>	TEST AL,ST_READY ; CHECK FOR NOT READY
6690 00001DD2 740E	<1>	JZ short CKST_EXIT
6691 00001DD4 B440	<1>	MOV AH,BAD_SEEK
6692 00001DD6 A810	<1>	TEST AL,ST_SEEK_COMPL ; CHECK FOR SEEK NOT COMPLETE
6693 00001DD8 7408	<1>	JZ short CKST_EXIT
6694 00001DDA B411	<1>	MOV AH,DATA_CORRECTED
6695 00001DDC A804	<1>	TEST AL,ST_CORRCTD ; CHECK FOR CORRECTED ECC
6696 00001DDE 7502	<1>	JNZ short CKST_EXIT
6697 00001DE0 B400	<1>	MOV AH,0
6698	<1>	CKST_EXIT:
6699 00001DE2 8826[6B17]	<1>	MOV [DISK_STATUS1],AH ; SET ERROR FLAG
6700 00001DE6 80FC11	<1>	CMP AH,DATA_CORRECTED ; KEEP GOING WITH DATA CORRECTED
6701 00001DE9 7403	<1>	JZ short CKST_EX1
6702 00001DEB 80FC00	<1>	CMP AH,0
6703	<1>	CKST_EX1:
6704 00001DEE C3	<1>	RETn
6705	<1>	
6706	<1> ;	-----
6707	<1> ;	CHECK FIXED DISK ERROR REGISTER :
6708	<1> ;	-----
6709	<1>	CHECK_ER:

6710		<1>	;MOV DX, HF_PORT+1	; GET THE ERROR REGISTER
6711	00001DEF 8B16[7217]	<1>	mov dx, [HF_PORT]	;
6712	00001DF3 FEC2	<1>	inc dl	
6713	00001DF5 EC	<1>	IN AL,DX	
6714	00001DF6 A2[6F17]	<1>	MOV [HF_ERROR],AL	
6715	00001DF9 53	<1>	PUSH BX	
6716	00001DFA B90800	<1>	MOV CX,8	; TEST ALL 8 BITS
6717	00001DFD D0E0	<1>	CK1: SHL AL,1	; MOVE NEXT ERROR BIT TO CARRY
6718	00001DFF 7202	<1>	JC short CK2	; FOUND THE ERROR
6719	00001E01 E2FA	<1>	LOOP CK1	; KEEP TRYING
6720	00001E03 BB[131E]	<1>	CK2: MOV BX,ERR_TBL	; COMPUTE ADDRESS OF
6721	00001E06 01CB	<1>	ADD BX,CX	; ERROR CODE
6722		<1>	;MOV AH,BYTE [CS:BX]	; GET ERROR CODE
6723	00001E08 8A27	<1>	mov ah, byte [bx]	
6724	00001E0A 8826[6B17]	<1>	CKEX: MOV [DISK_STATUS1],AH	; SAVE ERROR CODE
6725	00001E0E 5B	<1>	POP BX	
6726	00001E0F 80FC00	<1>	CMP AH,0	
6727	00001E12 C3	<1>	RETn	
6728		<1>	ERR_TBL:	
6729	00001E13 E0	<1>	DB NO_ERR	
6730	00001E14 024001BB	<1>	DB BAD_ADDR_MARK,BAD_SEEK,BAD_CMD,UNDEF_ERR	
6731	00001E18 04BB100A	<1>	DB RECORD_NOT_FND,UNDEF_ERR,BAD_ECC,BAD_SECTOR	
6732		<1>		
6733		<1>	;-----	
6734		<1>	; CHECK_DMA :	
6735		<1>	; -CHECK ES:BX AND # SECTORS TO MAKE SURE THAT IT WILL :	
6736		<1>	; FIT WITHOUT SEGMENT OVERFLOW. :	
6737		<1>	; -ES:BX HAS BEEN REVISED TO THE FORMAT SSSS:000X :	
6738		<1>	; -OK IF # SECTORS < 80H (7FH IF LONG READ OR WRITE) :	
6739		<1>	; -OK IF # SECTORS = 80H (7FH) AND BX <= 00H (04H) :	
6740		<1>	; -ERROR OTHERWISE :	
6741		<1>	;-----	
6742		<1>	CHECK_DMA:	
6743	00001E1C 50	<1>	PUSH AX	; SAVE REGISTERS
6744	00001E1D B80080	<1>	MOV AX,8000H	; AH = MAX # SECTORS AL = MAX OFFSET
6745	00001E20 F646FE02	<1>	TEST byte [CMD_BLOCK+6],ECC_MODE	
6746	00001E24 7403	<1>	JZ short CKD1	
6747	00001E26 B8047F	<1>	MOV AX,7F04H	; ECC IS 4 MORE BYTES
6748	00001E29 3A66F9	<1>	CKD1: CMP AH, [CMD_BLOCK+1]	; NUMBER OF SECTORS
6749	00001E2C 7706	<1>	JA short CKDOK	; IT WILL FIT
6750	00001E2E 7207	<1>	JB short CKDERR	; TOO MANY
6751	00001E30 38D8	<1>	CMP AL,BL	; CHECK OFFSET ON MAX SECTORS
6752	00001E32 7203	<1>	JB short CKDERR	; ERROR
6753	00001E34 F8	<1>	CKDOK: CLC	; CLEAR CARRY
6754	00001E35 58	<1>	POP AX	
6755	00001E36 C3	<1>	RETn	; NORMAL RETURN
6756	00001E37 F9	<1>	CKDERR: STC	; INDICATE ERROR
6757	00001E38 C606[6B17]09	<1>	MOV byte [DISK_STATUS1],DMA_BOUNDARY	
6758	00001E3D 58	<1>	POP AX	

```

6759 00001E3E C3      <1>      RETn
6760                  <1>
6761                  <1> ;-----
6762                  <1> ;      SET UP ES:BX-> DISK PARMS :
6763                  <1> ;-----
6764                  <1>
6765                  <1> ; INPUT -> DL = 0 based drive number
6766                  <1> ; OUTPUT -> ES:BX = disk parameter table address
6767                  <1>
6768                  <1> GET_VEC:
6769                  <1>      ;SUB    AX,AX                ; GET DISK PARAMETER ADDRESS
6770                  <1>      ;MOV    ES,AX
6771                  <1>      ;TEST   DL,1
6772                  <1>      ;JZ     short GV_0
6773                  <1> ;      LES    BX,[HF1_TBL_VEC]    ; ES:BX -> DRIVE PARAMETERS
6774                  <1> ;      JMP    SHORT GV_EXIT
6775                  <1> ;GV_0:
6776                  <1> ;      LES    BX,[HF_TBL_VEC]      ; ES:BX -> DRIVE PARAMETERS
6777                  <1> ;
6778                  <1>      xor     bh, bh
6779 00001E41 88D3      <1>      mov     bl, dl
6780                  <1>      ;;02/01/2015
6781                  <1>      ;;shl    bl, 1                ; port address offset
6782                  <1>      ;;mov    ax, [bx+hd_ports]    ; Base port address (1F0h, 170h)
6783                  <1>      ;;shl    bl, 1                ; dpt pointer offset
6784 00001E43 C0E302    <1>      shl     bl, 2 ;;
6785 00001E46 81C3[7817] <1>      add     bx, HF_TBL_VEC        ; Disk parameter table pointer
6786 00001E4A FF7702    <1>      push    word [bx+2]        ; dpt segment
6787 00001E4D 07        <1>      pop     es
6788 00001E4E 8B1F      <1>      mov     bx, [bx]            ; dpt offset
6789                  <1> ;GV_EXIT:
6790 00001E50 C3        <1>      RETn
6791                  <1>
6792                  <1>
6793                  <1> ;--- HARDWARE INT 76H -- ( IRQ LEVEL 14 ) -----
6794                  <1> ;
6795                  <1> ;      FIXED DISK INTERRUPT ROUTINE
6796                  <1> ;
6797                  <1> ;-----
6798                  <1>
6799                  <1> ; 22/12/2014
6800                  <1> ; IBM PC-XT Model 286 System BIOS Source Code - DISK.ASM (HD_INT)
6801                  <1> ;      '11/15/85'
6802                  <1> ; AWARD BIOS 1999 (D1A0622)
6803                  <1> ;      Source Code - ATORGS.ASM (INT_HDISK, INT_HDISK1)
6804                  <1>
6805                  <1> ;int_76h:
6806                  <1> HD_INT:
6807 00001E51 50        <1>      PUSH    AX

```

```

6808          <1>          ;PUSH DS
6809          <1>          ;CALL DDS
6810          <1>          ;
6811          <1>          ;MOV @HF_INT_FLAG,0FFH ; ALL DONE
6812 00001E52 2EC606[7017]FF <1>          mov byte [CS:HF_INT_FLAG], 0FFh
6813          <1>          ;
6814 00001E58 52 <1>          push dx
6815 00001E59 BAF701 <1>          mov dx, HDC1_BASEPORT+7 ; Status Register (1F7h)
6816          <1>          ; Clear Controller
6817          <1> Clear_IRQ1415: ; (Award BIOS - 1999)
6818 00001E5C EC <1>          in al, dx ;
6819 00001E5D 5A <1>          pop dx ;
6820          <1>          NEWIODELAY
6821 00001E5E E6EB <2>          out 0ebh,al
6822          <1>          ;
6823 00001E60 B020 <1>          MOV AL,E0I ; NON-SPECIFIC END OF INTERRUPT
6824 00001E62 E6A0 <1>          OUT INTB00,AL ; FOR CONTROLLER #2
6825          <1>          ;JMP $+2 ; WAIT
6826          <1>          NEWIODELAY
6827 00001E64 E6EB <2>          out 0ebh,al
6828 00001E66 E620 <1>          OUT INTA00,AL ; FOR CONTROLLER #1
6829          <1>          ;POP DS
6830          <1>          ;STI ; RE-ENABLE INTERRUPTS
6831          <1>          ;MOV AX,9100H ; DEVICE POST
6832          <1>          ;INT 15H ; INTERRUPT
6833 00001E68 58 <1>          POP AX
6834 00001E69 CF <1>          IRET ; RETURN FROM INTERRUPT
6835          <1>
6836          <1> ;+++++ HARDWARE INT 77H ++ ( IRQ LEVEL 15 ) ++++++
6837          <1> ; :
6838          <1> ; FIXED DISK INTERRUPT ROUTINE :
6839          <1> ; :
6840          <1> ;+++++
6841          <1>
6842          <1> ;int_77h:
6843          <1> HD1_INT:
6844 00001E6A 50 <1>          PUSH AX
6845          <1>          ;PUSH DS
6846          <1>          ;CALL DDS
6847          <1>          ;
6848          <1>          ;MOV @HF_INT_FLAG,0FFH ; ALL DONE
6849 00001E6B 2E800E[7017]C0 <1>          or byte [CS:HF_INT_FLAG],0C0h
6850          <1>          ;
6851 00001E71 52 <1>          push dx
6852 00001E72 BA7701 <1>          mov dx, HDC2_BASEPORT+7 ; Status Register (177h)
6853          <1>          ; Clear Controller (Award BIOS 1999)
6854 00001E75 EBE5 <1>          jmp short Clear_IRQ1415
6855
6856          ; 19/12/2014

```

```

6857 ; 15/12/2014
6858 clear_screen:
6859 ;mov ah, 0Fh ; get video mode
6860 ;int 10h
6861 ;; al = video mode
6862 ;mov ah, 0 ; set video mode (clears screen)
6863 ;int 10h
6864 ;retn
6865 ; 19/12/2014
6866 00001E77 06 push es
6867 00001E78 57 push di
6868 00001E79 BF00B8 mov di, 0B800h
6869 00001E7C 8EC7 mov es, di
6870 00001E7E 29FF sub di, di
6871 00001E80 B9D007 mov cx, 2000
6872 00001E83 B82007 mov ax, 0720h ; light gray char space (blank)
6873 00001E86 F3AB rep stosw
6874 00001E88 5F pop di
6875 00001E89 07 pop es
6876 00001E8A 31D2 xor dx, dx ; column 0, row 0
6877 00001E8C E99C0A jmp set_cposx ; set cursor position
6878
6879 rfdp_err:
6880 ;23/12/2014
6881 00001E8F C3 retn
6882
6883 dskprm:
6884 ; 20/02/2015
6885 ; 23/12/2014
6886 ; 19/12/2014
6887 ; DISPLAY DISK PARAMETERS TABLE
6888 ;
6889 ; INPUT -> DL = Disk/Drive #
6890 ;
6891 00001E90 8816[4529] mov byte [drv], dl ; 0,1,80h,81h,82h,83h
6892 ;
6893 00001E94 B408 mov ah, 08h ; Return drive parameters (conventional)
6894 00001E96 CD13 int 13h
6895 00001E98 72F5 jc short rfdp_err
6896 ;
6897 ;push bx ; 20/02/2015
6898 ; 23/12/2014
6899 00001E9A E8DAFF call clear_screen ; clear video page 0
6900 ;pop bx ; 20/02/2015
6901 ;
6902 00001E9D F606[4529]80 test byte [drv], 80h
6903 00001EA2 0F85AE00 jnz print_hdpt
6904 00001EA6 80C330 add bl, 30h ; '0'
6905 00001EA9 881E[FD20] mov byte [flpdtype], bl

```

```

6906                                     ; floppy disk drive type
6907                                     ; (1=360K, 2=1.2M, 3=720K, 4=1.44M)
6908 print_flpdpt:
6909                                     ; Writing the Diskette Parameter Table on screen
6910 mov     si, di
6911 mov     ax, es
6912 mov     ds, ax
6913 mov     ax, cs
6914 mov     es, ax
6915 lodsb   ; bits 0-3: SRT step rate time
6916         ; bits 4-7: head unload time
6917 mov     di, rSrtHdUnld
6918 call    write_hex
6919 lodsb   ; bit 0: 1=use DMA
6920         ; bits 2-7: head load time
6921 mov     di, rDmaHdLd
6922 call    write_hex
6923 lodsb   ; 55-ms increments
6924         ; before turning disk motor off
6925 mov     di, bMotorOff
6926 call    write_hex
6927 lodsb   ; sector size
6928         ; (0=128, 1=256, 2=512, 3=1024)
6929 mov     di, bSectSize
6930 call    write_hex
6931 lodsb   ; EOT (last sector on a track)
6932 mov     di, bLastTrack
6933 call    write_hex
6934 lodsb   ; gap length
6935         ; for read/write operations
6936 mov     di, bGapLen
6937 call    write_hex
6938 lodsb   ; DTL (Data Transfer Length)
6939         ; max transfer when length not set
6940 mov     di, bDTL
6941 call    write_hex
6942 lodsb   ; gap length for format operation
6943 mov     di, bGapFmt
6944 call    write_hex
6945 lodsb   ; fill character for format
6946         ; (normally F6H)
6947 mov     di, bFillChar
6948 call    write_hex
6949 lodsb   ; head-settle time
6950         ; (in milliseconds)
6951 mov     di, bHdSettle
6952 call    write_hex
6953 lodsb   ; motor-startup time
6954         ; (in 1/8th-second intervals)

```

6955 00001EFE BF[0B23]	mov di, bMotorOn
6956 00001F01 E83300	call write_hex
6957	; 19/12/2014
6958	; (extension, not in original bios function)
6959 00001F04 AC	lodsb ; Max. track number
6960 00001F05 BF[4523]	mov di, bMaxTrack
6961 00001F08 E82C00	call write_hex
6962 00001F0B AC	lodsb ; Data transfer rate
6963 00001F0C BF[6123]	mov di, bDataRate
6964 00001F0F E82500	call write_hex
6965	;
6966 00001F12 8CC8	mov ax, cs
6967 00001F14 8ED8	mov ds, ax
6968	;;mov al, byte [ds_drv]
6969 00001F16 A0[4529]	mov al, [drv] ; 11/01/2015
6970 00001F19 0430	add al, 30h ; '0'
6971 00001F1B A2[C620]	mov byte [flpdnum], al
6972 00001F1E BE[BE20]	mov si, FLPDPT
6973 00001F21 E88701	call print_msg
6974 00001F24 C3	retn
6975	
6976	write_dhex:
6977 00001F25 88E3	mov bl, ah
6978 00001F27 D0EB	shr bl, 1
6979 00001F29 D0EB	shr bl, 1
6980 00001F2B D0EB	shr bl, 1
6981 00001F2D D0EB	shr bl, 1
6982 00001F2F E81400	call dhgd
6983 00001F32 88E3	mov bl, ah
6984 00001F34 E80F00	call dhgd
6985	
6986	write_hex:
6987 00001F37 88C3	mov bl, al
6988 00001F39 D0EB	shr bl, 1
6989 00001F3B D0EB	shr bl, 1
6990 00001F3D D0EB	shr bl, 1
6991 00001F3F D0EB	shr bl, 1
6992 00001F41 E80200	call dhgd
6993 00001F44 88C3	mov bl, al
6994	;call dhgd
6995	;retn
6996	dhgd:
6997 00001F46 50	push ax
6998 00001F47 83E30F	and bx, 0Fh
6999 00001F4A 81C3[9D27]	add bx, hex_digits
7000 00001F4E 2E8A07	mov al, byte [CS:BX]
7001 00001F51 AA	stosb
7002 00001F52 58	pop ax
7003 00001F53 C3	retn

7004	
7005	print_hdpt:
7006	; 20/02/2015
7007	; 11/01/2015
7008	; 30/12/2014
7009	; 27/12/2014
7010	; 25/12/2014
7011	; 23/12/2014
7012	;; ES:DI = DPT address
7013	;
7014	00001F54 BE0000 mov si, HD0_DPT
7015	00001F57 8A1E[4529] mov bl, [drv] ; 20/02/2015
7016	00001F5B 83E303 and bx, 3
7017	00001F5E 88D8 mov al, bl
7018	00001F60 0402 add al, 2
7019	00001F62 A2[AE27] mov [ds_drv], al
7020	;
7021	00001F65 C0E305 shl bl, 5 ; * 32
7022	00001F68 01DE add si, bx
7023	00001F6A B80090 mov ax, DPT_SEGM
7024	00001F6D 8ED8 mov ds, ax
7025	00001F6F 8CC8 mov ax, cs
7026	00001F71 8EC0 mov es, ax
7027	; 25/12/2014
7028	00001F73 807C03A0 cmp byte [SI+3], 0A0h ; Translated table
7029	00001F77 0F849A00 je print_thdpt ; indicator
7030	;
7031	; Writing Fixed Disk Parameter Table on screen
7032	00001F7B AD lodsw ; Number of Cylinders
7033	00001F7C BF[D423] mov di, cylnum
7034	00001F7F E8A3FF call write_dhex
7035	00001F82 AC lodsb ; Number of Heads
7036	00001F83 BF[F123] mov di, headnum
7037	00001F86 E8AEFF call write_hex
7038	00001F89 AC lodsb ; Reserved
7039	00001F8A BF[0C24] mov di, rsvd3
7040	00001F8D E8A7FF call write_hex
7041	00001F90 AC lodsb ; Reserved
7042	00001F91 BF[2724] mov di, rsvd4
7043	00001F94 E8A0FF call write_hex
7044	00001F97 AD lodsw ; Precompensation (Obsolete)
7045	00001F98 BF[4224] mov di, pcompnum
7046	00001F9B E887FF call write_dhex
7047	00001F9E AC lodsb ; Reserved
7048	00001F9F BF[5F24] mov di, rsvd7
7049	00001FA2 E892FF call write_hex
7050	00001FA5 AC lodsb ; Drive Control Byte
7051	00001FA6 BF[7A24] mov di, dcbnum
7052	00001FA9 E88BFF call write_hex

7053	00001FAC	AD	lodsw ; Reserved
7054	00001FAD	BF[9524]	mov di, rsvd9
7055	00001FB0	E872FF	call write_dhex
7056	00001FB3	AC	lodsb ; Reserved
7057	00001FB4	BF[B224]	mov di, rsvd11
7058	00001FB7	E87DFF	call write_hex
7059	00001FBA	AD	lodsw ; Landing Zone (Obsolete)
7060	00001FBB	BF[CD24]	mov di, lzonenum
7061	00001FBE	E864FF	call write_dhex
7062	00001FC1	AC	lodsb ; Sectors per Track
7063	00001FC2	BF[EA24]	mov di, psptnum
7064	00001FC5	E86FFF	call write_hex
7065	00001FC8	AC	lodsb ; Reserved
7066	00001FC9	BF[0525]	mov di, rsvd15
7067	00001FCC	E868FF	call write_hex
7068			; 23/12/2014
7069			; (extension, not in original bios function)
7070	00001FCF	AD	lodsw ; I/O Port Base Address
7071	00001FD0	BF[2225]	mov di, bPortAddr
7072	00001FD3	E84FFF	call write_dhex
7073			; 06/01/2015
7074	00001FD6	AD	lodsw ; Control Port Address
7075	00001FD7	BF[3F25]	mov di, cPortAddr
7076	00001FDA	E848FF	call write_dhex
7077	00001FDD	AC	lodsb ; Head Register Upper Nibble
7078	00001FDE	BF[5C25]	mov di, hregupnib
7079	00001FE1	E853FF	call write_hex
7080			;
7081	00001FE4	8CC8	mov ax, cs
7082	00001FE6	8ED8	mov ds, ax
7083	00001FE8	A0[AE27]	mov al, [ds_drv]
7084	00001FEB	88C3	mov bl, al ; 06/01/2015
7085	00001FED	0430	add al, '0'
7086	00001FEF	A2[9C23]	mov [dsknum], al
7087			; 06/01/2015
7088	00001FF2	28FF	sub bh, bh
7089	00001FF4	C0E302	shl bl, 2
7090	00001FF7	89DE	mov si, bx
7091	00001FF9	81C6[042A]	add si, disk_size
7092	00001FFD	8B4402	mov ax, [si+2]
7093	00002000	BF[7925]	mov di, disksize
7094	00002003	E81FFF	call write_dhex
7095	00002006	8B04	mov ax, [si]
7096	00002008	BF[7D25]	mov di, disksize+4
7097	0000200B	E817FF	call write_dhex
7098			;
7099	0000200E	BE[9423]	mov si, HDPT
7100	00002011	E89700	call print_msg
7101	00002014	C3	retn

7102		print_thdpt:
7103		
7104		; 25/12/2014
7105		; Writing the Translated FDPT on screen
7106		; (PHOENIX - EDD specification v1.1)
7107	00002015 AD	lodsw ; Logical Numbers of Cylinders, Limit 1024
7108	00002016 BF[D225]	mov di, lcylnum
7109	00002019 E809FF	call write_dhex
7110	0000201C AC	lodsb ; Logical Numbers of Heads, Limit 256
7111	0000201D BF[EF25]	mov di, lheadnum
7112	00002020 E814FF	call write_hex
7113	00002023 AC	lodsb ; A0h signature, indicates translated table
7114	00002024 BF[0A26]	mov di, tsignum
7115	00002027 E80DFF	call write_hex
7116	0000202A AC	lodsb ; Physical Sectors per Track
7117	0000202B BF[2526]	mov di, tpsptnum
7118	0000202E E806FF	call write_hex
7119	00002031 AD	lodsw ; Precompensation (Obsolete)
7120	00002032 BF[4026]	mov di, tpcompnum
7121	00002035 E8EDFE	call write_dhex
7122	00002038 AC	lodsb ; Reserved
7123	00002039 BF[6926]	mov di, trsvd7
7124	0000203C E8F8FE	call write_hex
7125	0000203F AC	lodsb ; Drive Control Byte
7126	00002040 BF[8426]	mov di, tdcbnum
7127	00002043 E8F1FE	call write_hex
7128	00002046 AD	lodsw ; Physical Cylinders, limit 65536
7129	00002047 BF[9F26]	mov di, tpcylnum
7130	0000204A E8D8FE	call write_dhex
7131	0000204D AC	lodsb ; Physical Heads, limit 16
7132	0000204E BF[BC26]	mov di, tpheadnum
7133	00002051 E8E3FE	call write_hex
7134	00002054 AD	lodsw ; Landing Zone (Obsolete)
7135	00002055 BF[D726]	mov di, tlzonenum
7136	00002058 E8CAFE	call write_dhex
7137	0000205B AC	lodsb ; Logical Sectors per Track, Limit 63
7138	0000205C BF[0027]	mov di, lsptnum
7139	0000205F E8D5FE	call write_hex
7140	00002062 AC	lodsb ; Checksum for translated FDPT
7141	00002063 BF[1B27]	mov di, checksum
7142	00002066 E8CEFE	call write_hex
7143		; 23/12/2014
7144		; (extension, not in original bios function)
7145	00002069 AD	lodsw ; I/O Port Base Address
7146	0000206A BF[3827]	mov di, tbPortAddr
7147	0000206D E8B5FE	call write_dhex
7148		; 06/01/2015
7149	00002070 AD	lodsw ; Control Port Address
7150	00002071 BF[5527]	mov di, tcPortAddr

7151	00002074	E8AEFE	call	write_dhex
7152	00002077	AC	lodsb	; Head Register Upper Nibble
7153	00002078	BF[7227]	mov	di, thregupnib
7154	0000207B	E8B9FE	call	write_hex
7155			;	
7156	0000207E	8CC8	mov	ax, cs
7157	00002080	8ED8	mov	ds, ax
7158	00002082	A0[AE27]	mov	al, [ds_drv]
7159	00002085	88C3	mov	bl, al ; 06/01/2015
7160	00002087	0430	add	al, '0'
7161	00002089	A2[8F25]	mov	[tdsknum], al
7162			;	06/01/2015
7163	0000208C	28FF	sub	bh, bh
7164	0000208E	C0E302	shl	bl, 2
7165	00002091	89DE	mov	si, bx
7166	00002093	81C6[042A]	add	si, disk_size
7167	00002097	8B4402	mov	ax, [si+2]
7168	0000209A	BF[8F27]	mov	di, tdisksize
7169	0000209D	E885FE	call	write_dhex
7170	000020A0	8B04	mov	ax, [si]
7171	000020A2	BF[9327]	mov	di, tdisksize+4
7172	000020A5	E87DFE	call	write_dhex
7173			;	
7174	000020A8	BE[8725]	mov	si, THDPT
7175			;	call print_msg
7176			;	retn
7177				
7178			print_msg:	
7179	000020AB	BB0700	mov	bx, 7
7180	000020AE	B40E	mov	ah, 0Eh
7181			pmsg_loop:	
7182	000020B0	AC	lodsb	
7183	000020B1	20C0	and	al, al
7184	000020B3	7404	jz	short pmsg_ok
7185	000020B5	CD10	int	10h
7186	000020B7	EBF7	jmp	short pmsg_loop
7187			pmsg_ok:	
7188	000020B9	B410	mov	ah, 10h ; Getchar
7189	000020BB	CD16	int	16h
7190	000020BD	C3	retn	
7191			;	
7192			FLPDPT:	
7193	000020BE	07	db	07h
7194	000020BF	0D0A	db	0Dh, 0Ah
7195	000020C1	4469736B20	db	'Disk '
7196			flpdnum:	
7197	000020C6	58202D20	db	'X - '
7198	000020CA	4449534B4554544520-	db	'DISKETTE PARAMETER TABLE'
7199	000020D3	504152414D45544552-		

```

7200 000020DC 205441424C45
7201 000020E2 0D0A0D0A          db 0Dh, 0Ah, 0DH, 0Ah
7202 000020E6 5479706520202020-  db 'Type          : '
7203 000020EF 2020202020202020-
7204 000020F8 2020203A20
7205
7206 000020FD 58202020          flpdtype:
7207 00002101 5B2031203D20333630-  db 'X          '
7208 0000210A 4B2C2032203D20312E-  db '[ 1 = 360K, 2 = 1.2M, 3 = 720K, 4 = 1.44M ]'
7209 00002113 324D2C2033203D2037-
7210 0000211C 32304B2C2034203D20-
7211 00002125 312E34344D205D
7212 0000212C 0D0A0D0A          db 0Dh, 0Ah, 0DH, 0Ah
7213 00002130 535254202D20486561-  db 'SRT - Head Unld Time : '
7214 00002139 6420556E6C64205469-
7215 00002142 6D65203A20
7216
7217 00002147 585868202862697473-  rSrtHdUnld:
7218 00002150 20302D333A20535254-  db 'XXh (bits 0-3: SRT, bits 4-7: head unload time)'
7219 00002159 2C206269747320342D-
7220 00002162 373A20686561642075-
7221 0000216B 6E6C6F61642074696D-
7222 00002174 6529
7223 00002176 0D0A          db 0Dh, 0Ah
7224 00002178 444D41202D20486561-  db 'DMA - Head Load Time : '
7225 00002181 64204C6F6164205469-
7226 0000218A 6D65203A20
7227
7228 0000218F 585868202862697420-  rDmaHdLd:
7229 00002198 303A2031203D20444D-  db 'XXh (bit 0: 1 = DMA, bits 2-7: head load time)'
7230 000021A1 412C20626974732032-
7231 000021AA 2D373A206865616420-
7232 000021B3 6C6F61642074696D65-
7233 000021BC 29
7234 000021BD 0D0A          db 0Dh, 0Ah
7235 000021BF 4D6F746F72204F6666-  db 'Motor Off Count      : '
7236 000021C8 20436F756E74202020-
7237 000021D1 2020203A20
7238
7239 000021D6 585868202877697468-  bMotorOff:
7240 000021DF 2035356D7320696372-  db 'XXh (with 55ms icrements before turning off)'
7241 000021E8 656D656E7473206265-
7242 000021F1 666F7265207475726E-
7243 000021FA 696E67206F666629
7244 00002202 0D0A          db 0Dh, 0Ah
7245 00002204 536563746F72205369-  db 'Sector Size          : '
7246 0000220D 7A65202020202020-
7247 00002216 2020203A20
7248

```

7249 0000221B 585868202832203D20-	db 'XXh (2 = 512 bytes)'
7250 00002224 353132206279746573-	
7251 0000222D 29	
7252 0000222E 0D0A	db 0Dh, 0Ah
7253 00002230 4C6173742053656374-	db 'Last Sect on a Track : '
7254 00002239 206F6E206120547261-	
7255 00002242 636B203A20	
7256	bLastTrack:
7257 00002247 585868	db 'XXh'
7258 0000224A 0D0A	db 0Dh, 0Ah
7259 0000224C 476170204C656E6774-	db 'Gap Length (R/W) : '
7260 00002255 68202028522F572920-	
7261 0000225E 2020203A20	
7262	bGapLen:
7263 00002263 585868	db 'XXh'
7264 00002266 0D0A	db 0Dh, 0Ah
7265 00002268 44617461205472616E-	db 'Data Transfer Length : '
7266 00002271 73666572204C656E67-	
7267 0000227A 7468203A20	
7268	bDTL:
7269 0000227F 585868	db 'XXh'
7270 00002282 0D0A	db 0Dh, 0Ah
7271 00002284 476170204C656E6774-	db 'Gap Length (Format) : '
7272 0000228D 682028466F726D6174-	
7273 00002296 2920203A20	
7274	bGapFmt:
7275 0000229B 585868	db 'XXh'
7276 0000229E 0D0A	db 0Dh, 0Ah
7277 000022A0 46696C6C2043686172-	db 'Fill Char for format : '
7278 000022A9 20666F7220666F726D-	
7279 000022B2 6174203A20	
7280	bFillChar:
7281 000022B7 58586820286E6F726D-	db 'XXh (normally F6h)'
7282 000022C0 616C6C792046366829	
7283 000022C9 0D0A	db 0Dh, 0Ah
7284 000022CB 486561642053657474-	db 'Head Settle Time : '
7285 000022D4 6C652054696D652020-	
7286 000022DD 2020203A20	
7287	bHdSettle:
7288 000022E2 585868206D696C6C69-	db 'XXh milliseconds'
7289 000022EB 7365636F6E6473	
7290 000022F2 0D0A	db 0Dh, 0Ah
7291 000022F4 4D6F746F7220537461-	db 'Motor Startup Time : '
7292 000022FD 727475702054696D65-	
7293 00002306 2020203A20	
7294	bMotorOn:
7295 0000230B 5858682028696E2031-	db 'XXh (in 1/8th second intervals)'
7296 00002314 2F387468207365636F-	
7297 0000231D 6E6420696E74657276-	

7298 00002326 616C7329	
7299 0000232A 0D0A	db 0Dh, 0Ah
7300	; 19/12/2014
7301 0000232C 0D0A	db 0Dh, 0Ah
7302 0000232E 4D6178696D756D2054-	db 'Maximum Track Number : '
7303 00002337 7261636B204E756D62-	
7304 00002340 6572203A20	
7305	bMaxTrack:
7306 00002345 585868	db 'XXh'
7307 00002348 0D0A	db 0Dh, 0Ah
7308 0000234A 44617461205472616E-	db 'Data Transfer Rate : '
7309 00002353 736665722052617465-	
7310 0000235C 2020203A20	
7311	bDataRate:
7312 00002361 585868202830306820-	db 'XXh (00h = 500KBS, 40h = 300KBS, 80H = 250KBS)'
7313 0000236A 3D203530304B42532C-	
7314 00002373 20343068203D203330-	
7315 0000237C 304B42532C20383048-	
7316 00002385 203D203235304B4253-	
7317 0000238E 29	
7318 0000238F 0D0A	db 0Dh, 0Ah
7319 00002391 0D0A00	db 0Dh, 0Ah, 0
7320	
7321	; 23/12/2014
7322	HDPT:
7323 00002394 07	db 07h
7324 00002395 0D0A	db 0Dh, 0Ah
7325 00002397 4469736B20	db 'Disk '
7326	dsknum:
7327 0000239C 58202D20	db 'X - '
7328 000023A0 464958454420444953-	db 'FIXED DISK PARAMETER TABLE'
7329 000023A9 4B20504152414D4554-	
7330 000023B2 4552205441424C45	
7331 000023BA 0D0A0D0A	db 0Dh, 0Ah, 0Dh, 0Ah
7332 000023BE 4E756D626572206F66-	db 'Number of Cylinders : '
7333 000023C7 2043796C696E646572-	
7334 000023D0 73203A20	
7335	cylnum:
7336 000023D4 5858585868	db 'XXXXh'
7337 000023D9 0D0A	db 0Dh, 0Ah
7338 000023DB 4E756D626572206F66-	db 'Number of Heads : '
7339 000023E4 204865616473202020-	
7340 000023ED 20203A20	
7341	headnum:
7342 000023F1 585868	db 'XXh'
7343 000023F4 0D0A	db 0Dh, 0Ah
7344 000023F6 526573657276656420-	db 'Reserved : '
7345 000023FF 2020202020202020-	
7346 00002408 20203A20	

7347			rsvd3:	
7348	0000240C	585868	db 'XXh'	
7349	0000240F	0D0A	db 0Dh, 0Ah	
7350	00002411	526573657276656420-	db 'Reserved	:
7351	0000241A	2020202020202020-		
7352	00002423	20203A20		
7353			rsvd4:	
7354	00002427	585868	db 'XXh'	
7355	0000242A	0D0A	db 0Dh, 0Ah	
7356	0000242C	507265636F6D70656E-	db 'Precompensation	:
7357	00002435	736174696F6E202020-		
7358	0000243E	20203A20		
7359			pcompnum:	
7360	00002442	5858585868	db 'XXXXh'	
7361	00002447	0D0A	db 0Dh, 0Ah	
7362	00002449	526573657276656420-	db 'Reserved	:
7363	00002452	2020202020202020-		
7364	0000245B	20203A20		
7365			rsvd7:	
7366	0000245F	585868	db 'XXh'	
7367	00002462	0D0A	db 0Dh, 0Ah	
7368	00002464	447269766520436F6E-	db 'Drive Control Byte	:
7369	0000246D	74726F6C2042797465-		
7370	00002476	20203A20		
7371			dcbnum:	
7372	0000247A	585868	db 'XXh'	
7373	0000247D	0D0A	db 0Dh, 0Ah	
7374	0000247F	526573657276656420-	db 'Reserved	:
7375	00002488	2020202020202020-		
7376	00002491	20203A20		
7377			rsvd9:	
7378	00002495	5858585868	db 'XXXXh'	
7379	0000249A	0D0A	db 0Dh, 0Ah	
7380	0000249C	526573657276656420-	db 'Reserved	:
7381	000024A5	2020202020202020-		
7382	000024AE	20203A20		
7383			rsvd11:	
7384	000024B2	585868	db 'XXh'	
7385	000024B5	0D0A	db 0Dh, 0Ah	
7386	000024B7	4C616E64696E67205A-	db 'Landing Zone	:
7387	000024C0	6F6E652020202020-		
7388	000024C9	20203A20		
7389			lzonenum:	
7390	000024CD	5858585868	db 'XXXXh'	
7391	000024D2	0D0A	db 0Dh, 0Ah	
7392	000024D4	536563746F72732070-	db 'Sectors per Track	:
7393	000024DD	657220547261636B20-		
7394	000024E6	20203A20		
7395			psptnum:	

7396	000024EA	585868	db 'XXh'
7397	000024ED	0D0A	db 0Dh, 0Ah
7398	000024EF	526573657276656420-	db 'Reserved : '
7399	000024F8	2020202020202020-	
7400	00002501	20203A20	
7401			rsvd15:
7402	00002505	585868	db 'XXh'
7403	00002508	0D0A	db 0Dh, 0Ah
7404	0000250A	0D0A	db 0Dh, 0Ah
7405	0000250C	492F4F20506F727420-	db 'I/O Port Base Addr : '
7406	00002515	426173652041646472-	
7407	0000251E	20203A20	
7408			bPortAddr:
7409	00002522	5858585868	db 'XXXXh'
7410	00002527	0D0A	db 0Dh, 0Ah
7411	00002529	436F6E74726F6C2050-	db 'Control Port Addr : '
7412	00002532	6F7274204164647220-	
7413	0000253B	20203A20	
7414			cPortAddr:
7415	0000253F	5858585868	db 'XXXXh'
7416	00002544	0D0A	db 0Dh, 0Ah
7417	00002546	486561642052656720-	db 'Head Reg Upp Nibb : '
7418	0000254F	557070204E69626220-	
7419	00002558	20203A20	
7420			hregupnib:
7421	0000255C	585868	db 'XXh'
7422	0000255F	0D0A	db 0Dh, 0Ah
7423	00002561	0D0A	db 0Dh, 0Ah
7424	00002563	53697A652028696E20-	db 'Size (in sectors) : '
7425	0000256C	736563746F72732920-	
7426	00002575	20203A20	
7427			disksize:
7428	00002579	585858585858585868	db 'XXXXXXXXh'
7429	00002582	0D0A	db 0Dh, 0Ah
7430	00002584	0D0A00	db 0Dh, 0Ah, 0
7431			
7432			THDPT:
7433	00002587	07	db 07h
7434	00002588	0D0A	db 0Dh, 0Ah
7435	0000258A	4469736B20	db 'Disk '
7436			tdsknum:
7437	0000258F	58202D20	db 'X - '
7438	00002593	5452414E534C415445-	db 'TRANSLATED FIXED DISK PARAMETER TABLE'
7439	0000259C	442046495845442044-	
7440	000025A5	49534B20504152414D-	
7441	000025AE	45544552205441424C-	
7442	000025B7	45	
7443	000025B8	0D0A0D0A	db 0Dh, 0Ah, 0Dh, 0Ah
7444	000025BC	4C6F676963616C2043-	db 'Logical Cylinders : '

7445	000025C5	796C696E6465727320-	
7446	000025CE	20203A20	
7447			lcylnum:
7448	000025D2	5858585868	db 'XXXXh'
7449	000025D7	0D0A	db 0Dh, 0Ah
7450	000025D9	4C6F676963616C2048-	db 'Logical Heads : '
7451	000025E2	656164732020202020-	
7452	000025EB	20203A20	
7453			lheadnum:
7454	000025EF	585868	db 'XXh'
7455	000025F2	0D0A	db 0Dh, 0Ah
7456	000025F4	5369676E6174757265-	db 'Signature : '
7457	000025FD	202020202020202020-	
7458	00002606	20203A20	
7459			tsignum:
7460	0000260A	585868	db 'XXh'
7461	0000260D	0D0A	db 0Dh, 0Ah
7462	0000260F	506879205365632070-	db 'Phy Sec per Track : '
7463	00002618	657220547261636B20-	
7464	00002621	20203A20	
7465			tpsptnum:
7466	00002625	585868	db 'XXh'
7467	00002628	0D0A	db 0Dh, 0Ah
7468	0000262A	507265636F6D70656E-	db 'Precompensation : '
7469	00002633	736174696F6E202020-	
7470	0000263C	20203A20	
7471			tpcompnum:
7472	00002640	58585858682020284F-	db 'XXXXh (Obsolete)'
7473	00002649	62736F6C65746529	
7474	00002651	0D0A	db 0Dh, 0Ah
7475	00002653	526573657276656420-	db 'Reserved : '
7476	0000265C	202020202020202020-	
7477	00002665	20203A20	
7478			trsvd7:
7479	00002669	585868	db 'XXh'
7480	0000266C	0D0A	db 0Dh, 0Ah
7481	0000266E	447269766520436F6E-	db 'Drive Control Byte : '
7482	00002677	74726F6C2042797465-	
7483	00002680	20203A20	
7484			tdcbnum:
7485	00002684	585868	db 'XXh'
7486	00002687	0D0A	db 0Dh, 0Ah
7487	00002689	506879736963616C20-	db 'Physical Cylinders : '
7488	00002692	43796C696E64657273-	
7489	0000269B	20203A20	
7490			tpcylnum:
7491	0000269F	5858585868	db 'XXXXh'
7492	000026A4	0D0A	db 0Dh, 0Ah
7493	000026A6	506879736963616C20-	db 'Physical Heads : '

```

7494 000026AF 486561647320202020-
7495 000026B8 20203A20
7496
7497 000026BC 585868
7498 000026BF 0D0A
7499 000026C1 4C616E64696E67205A-
7500 000026CA 6F6E65202020202020-
7501 000026D3 20203A20
7502
7503 000026D7 58585858682020284F-
7504 000026E0 62736F6C65746529
7505 000026E8 0D0A
7506 000026EA 4C6F67696320536563-
7507 000026F3 207065722054726B20-
7508 000026FC 20203A20
7509
7510 00002700 585868
7511 00002703 0D0A
7512 00002705 436865636B73756D20-
7513 0000270E 202020202020202020-
7514 00002717 20203A20
7515
7516 0000271B 585868
7517 0000271E 0D0A
7518 00002720 0D0A
7519 00002722 492F4F20506F727420-
7520 0000272B 426173652041646472-
7521 00002734 20203A20
7522
7523 00002738 5858585868
7524 0000273D 0D0A
7525 0000273F 436F6E74726F6C2050-
7526 00002748 6F7274204164647220-
7527 00002751 20203A20
7528
7529 00002755 5858585868
7530 0000275A 0D0A
7531 0000275C 486561642052656720-
7532 00002765 557070204E69626220-
7533 0000276E 20203A20
7534
7535 00002772 585868
7536 00002775 0D0A
7537 00002777 0D0A
7538 00002779 53697A652028696E20-
7539 00002782 736563746F72732920-
7540 0000278B 20203A20
7541
7542 0000278F 5858585858585868

```

```

tpheadnum:
    db 'XXh'
    db 0Dh, 0Ah
    db 'Landing Zone' : '

tlzonenum:
    db 'XXXXh (Obsolete)'

    db 0Dh, 0Ah
    db 'Logic Sec per Trk' : '

lsptnum:
    db 'XXh'
    db 0Dh, 0Ah
    db 'Checksum' : '

checksum:
    db 'XXh'
    db 0Dh, 0Ah
    db 0Dh, 0Ah
    db 'I/O Port Base Addr' : '

tbPortAddr:
    db 'XXXXh'
    db 0Dh, 0Ah
    db 'Control Port Addr' : '

tcPortAddr:
    db 'XXXXh'
    db 0Dh, 0Ah
    db 'Head Reg Upp Nibb' : '

thregupnib:
    db 'XXh'
    db 0Dh, 0Ah
    db 0Dh, 0Ah
    db 'Size (in sectors)' : '

tdisksize:
    db 'XXXXXXXXh'

```

7543 00002798 0D0A	db 0Dh, 0Ah
7544 0000279A 0D0A00	db 0Dh, 0Ah, 0
7545	
7546	hex_digits:
7547	hexchrs:
7548 0000279D 303132333435363738-	db '0123456789ABCDEF'
7549 000027A6 39414243444546	
7550	
7551	inds:
7552 000027AD 00	db 0
7553	ds_drv:
7554 000027AE FF	db 0FFh ; Current drive (on display)
7555 000027AF 00	db 0 ; Current half (0 or >0)
7556	ds_sec:
7557 000027B0 00000000	dd 0 ; Current sector (on display), drv 0
7558 000027B4 00000000	dd 0 ; Current sector (on display), drv 1
7559 000027B8 00000000	dd 0 ; Current sector (on display), drv 2
7560 000027BC 00000000	dd 0 ; Current sector (on display), drv 3
7561 000027C0 00000000	dd 0 ; Current sector (on display), drv 4
7562 000027C4 00000000	dd 0 ; Current sector (on display), drv 5
7563	paragr:
7564 000027C8 00	db 0
7565	
7566	drv_names:
7567 000027C9 666430206664312068-	db 'fd0 fd1 hd0 hd1 hd2 hd3 '
7568 000027D2 643020686431206864-	
7569 000027DB 322068643320	
7570	
7571	dpheader:
7572 000027E1 204472697665203A20	db ' Drive : '
7573	drv_name:
7574 000027EA 3030302020	db '000 '
7575 000027EF 536563746F72203A20	db 'Sector : '
7576	sector_num:
7577 000027F8 464646464646464668	db 'FFFFFFFFh'
7578 00002801 00	db 0
7579	
7580	current_txtpos:
7581 00002802 00000000	dd 0
7582	txtposoff:
7583 00002806 00	db 0 ; txtpos offset for sector number input
7584	dscmd:
7585 00002807 00	db 0 ; 0 = change drive
7586	
7587	
7588	
7589 00002808 204279746520	db ' Byte '
7590	sdline_1:
7591 0000280E 30303068	db '000h'

7592	00002812	202D2020	db ' - '
7593			sdline_2:
7594	00002816	303020303020303020-	db '00 00 00 00 00 00 00 00 '
7595	0000281F	303020303020303020-	
7596	00002828	303020303020	
7597	0000282E	303020303020303020-	db '00 00 00 00 00 00 00 00 '
7598	00002837	303020303020303020-	
7599	00002840	303020303020	
7600	00002846	20	db ' '
7601			sdline_3:
7602	00002847	2E2E2E2E2E2E2E2E-	db '.....'
7603	00002850	2E2E2E2E2E2E2E	
7604	00002857	20	db 20h
7605			
7606			dpfooter1:
7607	00002858	204631203D20436861-	db ' F1 = Change Drive '
7608	00002861	6E6765204472697665-	
7609	0000286A	2020	
7610	0000286C	486F6D65203D204669-	db 'Home = First Sector '
7611	00002875	72737420536563746F-	
7612	0000287E	7220	
7613	00002880	50675570203D205072-	db 'PgUp = Previous Sector '
7614	00002889	6576696F7573205365-	
7615	00002892	63746F7220	
7616	00002897	455343203D20455849-	db 'ESC = EXIT'
7617	000028A0	54	
7618	000028A1	00	db 0
7619			dpfooter2:
7620	000028A2	204632203D20436861-	db ' F2 = Change Sector '
7621	000028AB	6E676520536563746F-	
7622	000028B4	7220	
7623	000028B6	456E64203D204C6173-	db 'End = Last Sector '
7624	000028BF	7420536563746F7220-	
7625	000028C8	2020	
7626	000028CA	5067446F776E203D20-	db 'PgDown = Next Sector '
7627	000028D3	4E6578742053656374-	
7628	000028DC	6F72202020	
7629	000028E1	454E544552203D2050-	db 'ENTER = Prv/Nxt'
7630	000028EA	72762F4E7874	
7631	000028F0	00	db 0
7632			ibcp:
7633	000028F1	00	db 0 ; input box - row position
7634	000028F2	00	db 0 ; input box - column position
7635			F1_ib:
7636	000028F3	10	db 16 ; box width (columns)
7637	000028F4	03	db 3 ; box height (rows)
7638	000028F5	01	db 1 ; label offset (vertical)
7639	000028F6	01	db 1 ; label offset (horizontal)
7640	000028F7	01	db 1 ; text (input) size

7641	000028F8	4E	db 4Eh ; box color
7642	000028F9	44726976653A20	db 'Drive: ' ; Label
7643	00002900	00	db 0
7644			
7645			F2_ib:
7646	00002901	14	db 20 ; box width (columns)
7647	00002902	03	db 3 ; box height (rows)
7648	00002903	01	db 1 ; label offset (vertical)
7649	00002904	01	db 1 ; label offset (horizontal)
7650	00002905	08	db 8 ; text (input) size
7651	00002906	4E	db 4Eh ; box color
7652	00002907	536563746F72203A20	db 'Sector : ' ; Label
7653	00002910	00	db 0
7654			
7655			dskr_err:
7656			;db 33 ; box width (columns)
7657	00002911	11	db 17
7658	00002912	03	db 3 ; box height (rows)
7659	00002913	01	db 1 ; label offset (vertical)
7660	00002914	01	db 1 ; label offset (horizontal)
7661	00002915	00	db 0 ; text (input) size
7662	00002916	4E	db 4Eh ; box color
7663			;db 'Drive not ready or read error !' ; Label
7664	00002917	204572726F72203A20	db ' Error : '
7665			err_code_str:
7666	00002920	303068202120	db '00h ! '
7667	00002926	00	db 0
7668			
7669			; Additional functions, variables/pointers for
7670			; Real Mode adaption (out of unix386.s) variables/pointers
7671			
7672			;28/11/2014
7673			set_cpos:
7674			;mov bh, bl
7675	00002927	8B16[3229]	mov dx, [cursor_posn] ; dh = row, dl = column
7676			set_cposx:
7677			; DX = cursor position
7678	0000292B	B402	mov ah, 02h ; Set cursor position
7679	0000292D	30FF	xor bh, bh ; for video page 0
7680	0000292F	CD10	int 10h
7681	00002931	C3	retn
7682			
7683			align 2
7684			
7685			; 28/11/2014
7686	00002932	0000	cursor_posn: dw 0
7687			; 01/12/2014
7688	00002934	0000	cursor_shp: dw 0
7689			; 12/02/2015 (bugfix)

```

7690 00002936 0000      cursor_posb: dw 0 ; (cursor position backup, for video page 0)
7691
7692      ;; 10/12/2014
7693      ;; 24/11/2014 (Retro UNIX 386 v1)
7694      ; Physical drive type & flags
7695 00002938 00      fd0_type:  db 0 ; floppy drive type
7696 00002939 00      fd1_type:  db 0 ; 4 = 1.44 Mb, 80 track, 3.5" (18 spt)
7697                      ; 6 = 2.88 Mb, 80 track, 3.5" (36 spt)
7698                      ; 3 = 720 Kb, 80 track, 3.5" (9 spt)
7699                      ; 2 = 1.2 Mb, 80 track, 5.25" (15 spt)
7700                      ; 1 = 360 Kb, 40 track, 5.25" (9 spt)
7701 0000293A 00      hd0_type:  db 0 ; EDD status for hd0 (bit 7 = present flag)
7702 0000293B 00      hd1_type:  db 0 ; EDD status for hd1 (bit 7 = present flag)
7703 0000293C 00      hd2_type:  db 0 ; EDD status for hd2 (bit 7 = present flag)
7704 0000293D 00      hd3_type:  db 0 ; EDD status for hd3 (bit 7 = present flag)
7705                      ; bit 0 - Fixed disk access subset supported
7706                      ; bit 1 - Drive locking and ejecting
7707                      ; bit 2 - Enhanced disk drive support
7708                      ; bit 3 = Reserved (64 bit EDD support)
7709                      ; (If bit 0 is '1' Retro UNIX 386 v1
7710                      ; will interpret it as 'LBA ready'!)
7711      ; 01/12/2014
7712 0000293E 0000      drv_status: dw 0 ; fd0, fd1 (FFh = failure, 80h = existing)
7713 00002940 00000000      dd 0 ; hd0, hd1 hd2, hd3 (FFh = failure)
7714                      ; (80h - 87h = existing)
7715                      ; (bit 0 = 1 : LBA ready)
7716      ; 10/12/2014
7717 00002944 00      hdc:  db 0 ; Hard (Fixed) disk count
7718 00002945 00      drv:  db 0 ; physical drive number (0, 1, 80h, 81h, 82h, 83h)
7719
7720      ; 23/12/2014 (dw)
7721      ; 16/12/2014 (db)
7722 00002946 0000      wait_count: dw 0 ; INT 08h timer ticks for disk functions
7723
7724      ; 02/12/2014
7725 00002948 00000000      prev_sec:  dd 0 ; previous sector (before reading)
7726 0000294C 00      last_drv:  db 0 ; 25/12/2014 - physical drv num of last hard disk
7727
7728      ; 03/01/2015
7729 0000294D 00      LBAMode: db 0
7730
7731      prg_msg:
7732 0000294E 0D0A07      db 0Dh, 0Ah, 07h
7733 00002951 4469736B2052656164-      db 'Disk Read Utility - Retro UNIX 386 v1 Disk I/O test.'
7734 0000295A 205574696C69747920-
7735 00002963 2D20526574726F2055-
7736 0000296C 4E4958203338362076-
7737 00002975 31204469736B20492F-
7738 0000297E 4F20746573742E

```

```

7739 00002985 0D0A                db 0Dh, 0Ah
7740 00002987 6279204572646F6761-   db 'by Erdogan Tan [21/02/2015]'
7741 00002990 6E2054616E20205B32-
7742 00002999 312F30322F32303135-
7743 000029A2 5D
7744 000029A3 0D0A0D0A                db 0Dh, 0Ah, 0Dh, 0Ah
7745 000029A7 28507265737320616E-   db '(Press any key to continue...)'
7746 000029B0 79206B657920746F20-
7747 000029B9 636F6E74696E75652E-
7748 000029C2 2E2E29
7749 000029C5 0D0A00                db 0Dh, 0Ah, 0
7750
7751                                ; 16/12/2014
7752                                drv_not_ready:
7753 000029C8 070D0A                db 07h, 0Dh, 0Ah
7754 000029CB 4472697665206E6F74-   db 'Drive not ready !' ; Temporary
7755 000029D4 2072656164792021
7756 000029DC 0D0A00                db 0Dh, 0Ah, 0
7757
7758 000029DF 90                align 2
7759
7760 000029E0 000000000000000000-   cylinders : dw 0, 0, 0, 0, 0, 0
7761 000029E9 000000
7762 000029EC 000000000000000000-   heads : dw 0, 0, 0, 0, 0, 0
7763 000029F5 000000
7764 000029F8 000000000000000000-   spt : dw 0, 0, 0, 0, 0, 0
7765 00002A01 000000
7766 00002A04 000000000000000000-   disk_size : dd 0, 0, 0, 0, 0, 0
7767 00002A0D 000000000000000000-
7768 00002A16 000000000000
7769
7770                                disk_pkt:
7771 00002A1C 10                db 16 ; Packet size
7772 00002A1D 00                db 0
7773 00002A1E 01                db 1 ; Transfer count (1 sector)
7774 00002A1F 00                db 0
7775                                tbuff:
7776 00002A20 [2C2A]           dw buffer ; offset
7777 00002A22 0000            dw 0 ; segment
7778                                lba:
7779 00002A24 00000000           dd 0
7780 00002A28 00000000           dd 0
7781                                align 2
7782
7783                                buffer:
7784                                _end:

```