

```

1 ;.PAGE 'Original UCSD Pascal v1.1'
2 ; Source file generated by Willi Kusche
3 ; using a 6502 disassembler
4 ;
5 ;
6 * LST OFF
7 ;
8 ; Assembly control
9 ;
10 ORIG DFLAG
11 *V11 DFLAG
12 PROCV EQU 2
13 INTPGS EQU 4
14 MAXVOL EQU 13
15 VERK EQU 2
16 FLAVK EQU $01
17 INITK2 EQU 64
18 INITK3 EQU $01
19 ;
20 ;
21 ;
22 INCLUDE MACROS.IAS
23 ;
24 ;
25 ;
26 INCLUDE MAPPGO.IAS
27 ;
28 ;
29 ;
30 RDMAIN MACRO
31 IFDEF BIG
32 STA RAMRDC
33 ENDF
34 ENDM
35 ;
36 ;
37 ;
38 RDAUX MACRO
39 IFDEF BIG
40 STA RAMRDS
41 ENDF
42 ENDM
43 ;
44 ;
45 ;
46 DSECT
47 ORG 0
48 POBASE
49 A00 DS 2
50 A02 DS 2
51 DS 54
52 A3A DS 2 ; New
53 DS 2
54 A3E DS 2
55 A40 DS 2 ; New
56 A42 DS 2 ; New
57 A44 DS 2 ; New

```

```

16 A46      DS      2
17      DS      8
18 BASPTR   DS      2
19 MSPTR    DS      2
20 JTBPTR   DS      2
21 SEGPTR   DS      2
22 IPCPTR   DS      2
23 HPPTR    DS      2
24 PSKPTR   DS      2
25  ifndef  ORIG
26  ifdef   BIG
27 BZ5E     DS      2
28 BZ60     DS      2
29 BZ62     DS      2
30 BZ64     DS      2
31 BZ66     DS      2
32  else
33      DS     10
34  endif
35  endif
36 OPRND    DS      2
37      DS      2
38 A62      DS      2
39 *A64
40 VARPTR
41 SUM      DS      2
42 SPTMP    DS      2
43 SRCPTR   DS      2
44 DSTPTR   DS      2
45 CMPTYP
46 A6C      DS      2
47 JMPPC    DS      3
48 JMPCSP   DS      3
49  ifdef   ORIG
50 NB8
51 PCDSIZ
52  endif
53 FPREG1   ; DS    6
54 A74      DS      2
55  ifndef  ORIG
56 PCDSIZ
57  endif
58 A76      DS      2
59 PROCNO
60 A78      DS      2
61 BLKNUM
62 FPREG2   ; DS    6
63 A7A      DS      2
64 A7C      DS      2
65 ARPTR
66 A7E      DS      2
67 FPREG3   ; DS    6
68 A80      DS      1
69 A81      DS      1
70 A82      DS      2
71  ifdef   ORIG
72 LCNTH

```

```

73  NBA
74  endi f
75  MLEXLV
76  A84      DS      1
77  A85      DS      1
78  SEGNO
79  A86      DS      2
80  TMSPTR   ; DS    2
81  FPRETA   ; DS    2
82  A88      DS      2
83  A8A      DS      2
84  TJTPTR
85  A8C      DS      2
86  A8E      DS      1
87  A8F      DS      1
88  A90      DS      2
89      DS      4
90  A96      DS      1
91  A97      DS      1
92  A98      DS      1
93      DS      3
94  A9C      DS      1
95  i f def  ORIG
96      DS      30
97  el se
98  ZA7      DS      2
99      DS      15
100 NB8      DS      2
101 LCNTH
102 NBA      DS      1
103 endi f
104      DS      2
105 ABD      DS      1
106 ABE      DS      1      ; New
107 ABF      DS      1
108 RETADR   DS      1
109 AC1      DS      1
110 AC2      DS      1
111 AC3      DS      1
112 AC4      DS      1
113 AC5      DS      1
114      DS      1
115 AC7      DS      1
116 REQLT    DS      1
117 REQTRK   DS      1
118 AFDATA
119 ACA      DS      1
120 ACTSEC   DS      1
121 ACTTRK   DS      1
122 ACTVOL   DS      1
123 HSL0T1
124 PRVTRK
125 HCKSUM   DS      1
126 RETRYC
127 HODDB
128 SAVEY
129 LOOPC

```

```

130 ACF      DS      1
131 ADO      DS      2
132 AD2      DS      1
133 AD3      DS      1
134      DS      2
135 AD6      DS      2
136 AD8      DS      2
137      DS      6
138 AEO      DS      1
139 AE1      DS      1
140      DS      2
141 RTPTR    DS      2
142 WTPTR    DS      2
143 UDJVP    DS      2
144      DS      6
145 SLLPTR   DS      2
146 SLRPTR   DS      2
147 HPOS     DS      1
148 VPOS     DS      1
149 TEMP     DS      2
150 SYCPTR   DS      2
151      i f n d e f  O R I G
152 CNFLGS   DS      1      ; Moved
153 NBF11    DS      1      ; New
154 ESCFLG   DS      1      ; Moved
155      DS      2
156 SVFLGS   DS      1      ; New
157      e n d i f
158      D E N D
27 ;
28 ;
29 ;
30 P1BASE    EQU     $0100
31 ;
32 ;   D i s k I / O r e f e r e n c e s   t o   p a g e s   t w o   a n d   t h r e e
33 ;
34      I N C L U D E      M A P D I O . I A S
  1 ;
  2 ;
  3 ;
  4 A0200    EQU     $0200
  5 ;
  6 ;
  7 ;
  8 BUFF6B   EQU     $0200
  9 ;
10 ;
11 ;
12 BUFF2B   EQU     $0300
13 ;
14 ;
15 ;
16 SLOTNO   EQU     $03A1
17 DRVNO    EQU     $03A2
18 A03A3    EQU     $03A3
19 A03A4    EQU     $03A4
20 A03A5    EQU     $03A5

```

```

21 XFRADR EQU $03A8
22 RDWRFL EQU $03AC
23 A03AD EQU $03AD
24 A03AE EQU $03AE
25 PRVSLT EQU $03AF
26 PRVDRV EQU $03B0
27 A03B1 EQU $03B1
35 ;
36 ; I/O slot references
37 ;
38 A0478 EQU $0478
39 A04F8 EQU $04F8
40 A05B8 EQU $05B8
41 A06F8 EQU $06F8
42 ;
43 ;
44 ;
45 DUMMY EQU $1000
46 ;
47 ;
48 ;
49 A6000 EQU $6000
50 A6200 EQU $6200
51 A6800 EQU $6800
52 A8000 EQU $8000
53 ;
54 ; System communications area
55 ;
56 INCLUDE MAPSYSC.IAS
1 DSECT
2 ;
3 ;
4 ;
5 ifdef BIG
6 ORG $BB5C
7 else
8 ORG $BD1C
9 endif
10 ;
11 ;
12 ;
13 ABD1C DS 2
14 ifdef BIG
15 ;
16 ;
17 ;
18 SXXTBL DS 64
19 endif
20 ;
21 ; Segment use counters
22 ;
23 SUCTBL DS INITK2
24 ;
25 ; Segment load addresses
26 ;
27 SLATBL DS INITK2
28 ;

```

```

29 ;
30 ;
31 IUTBL DS INITK2
32 ;
33 ; 'IORSLT' Result of last I/O call
34 ;
35 IORSLT DS 2
36 ;
37 ; 'XEQERR' Reason for 'EXECERROR' call
38 ;
39 ABDE0 DS 2
40 ;
41 ; 'SYSUNIT' Physical unit of bootload
42 ;
43 ABDE2 DS 2
44 DS 2 ; 'BUGSTATE' Debugger info
45 ;
46 ; 'GDIRP' Global directory pointer
47 ;
48 SYCGDP DS 2
49 ABDE8 DS 2 ;
50 SYCBAS DS 2 ;
51 SYCMS DS 2 ;
52 SYCJTB DS 2 ;
53 SYCSEG DS 2 ;
54 ABDF2 DS 2 ;
55 ABDF4 DS 2 ;
56 DS 58 ;
57 ABE30 DS 2 ;
58 DS 12 ;
59 ;
60 ;
61 ;
62 ifndef BIG
63 SYPTBL DS 192 ;
64 else
65 SYPTBL DS 384 ;
66 endif
67 XXX
68 DS 12
69 ABFOA DS 2
70 DS 2
71 ABFOE DS 1
72 LFFLAG DS 1
73 DS 1
74 ;
75 ;
76 ;
77 ifdef ORIG
78 NBF11
79 endif
80 ABF11 DS 1
81 ;
82 ;
83 ;
84 ifndef ORIG
85 ABF12 DS 1

```

```

86  else
87  ESCFLG  DS    1
88  endif
89  ;
90  ;
91  ;
92  RANDNO  DS    2
93  ;
94  ;
95  ;
96  ifndef ORIG
97  NBF15   DS    1
98  else
99  CNFLGS  DS    1
100 endif
101 ;
102 ;
103 ;
104 BREAK   DS    2
105 RPTR    DS    1
106 WPTR    DS    1
107 ABF1A   DS    2
108 SPCHAR  DS    1
109 I BRK   DS    2
110 I SYSC  DS    2
111 VER     DS    1
112 FLAVOR  DS    1
113 ABF23   DS    2
114     DS    2
115 ;
116 ; Slot types
117 ;
118 SLTYP5  DS    8
119 ;
120 ;
121 ;
122 XI TLOC DS    2
123 ABF31   DS    2
124 ABF33   DS    2
125     DS    33
126 ABF56   DS    42
127 ;
128 ;
129 ;
130     DEND
57 ;
58 ; I/O page
59 ;
60     INCLUDE     MAPIOP. IAS
1 ;
2 ; Pointer to I/O page
3 ;
4 PTRIOP  EQU    SC000
5 ;
6 ;
7 ;
8 KBD     EQU    SC000

```

```

9 ;
10 ; Soft switches
11 ;
12 ST080C EQU SC000 ; Clear 'STORE80'
13 ST080S EQU SC001 ; Set 'STORE80'
14 RAMRDC EQU SC002 ; Clear 'RAMRD'
15 RAMRDS EQU SC003 ; Set 'RAMRD'
16 RAMWRC EQU SC004 ; Clear 'RAMWRT'
17 RAMWRS EQU SC005 ; Set 'RAMWRT'
18 ALTZPC EQU SC008 ; Clear 'ALTZP'
19 ALTZPS EQU SC009 ; Set 'ALTZP'
20 ALTCHS EQU SC00F ; Set 'ALTCHAR'
21 ;
22 ;
23 ;
24 KBSTRB EQU SC010
25 ;
26 ; Speaker toggle
27 ;
28 AC030 EQU SC030
29 ;
30 ; Video modes
31 ;
32 AC050 EQU SC050
33 AC051 EQU SC051
34 AC052 EQU SC052
35 AC054 EQU SC054
36 AC057 EQU SC057
37 ;
38 ;
39 ;
40 AC060 EQU SC060 ; New
41 BUTNO EQU SC061 ; Open-Apple key
42 BUTN1 EQU SC062 ; Closed-Apple key ('option' key on newer
model s)
43 AC063 EQU SC063
44 ;
45 ; Language card bank switching control
46 ;
47 *AC081
48 ROMIN EQU SC081
49 *AC083
50 LCBNK2 EQU SC083
51 *AC08B
52 LCBNK1 EQU SC08B
53 ;
54 ; ???
55 ;
56 AC08A EQU SC08A
57 AC08E EQU SC08E
58 AC08F EQU SC08F
59 ;
60 ; Disk I/O definitions from page A-11 of "Beneath Apple ProDOS"
61 ;
62 DRVSMO EQU SC080
63 DRVSM1 EQU SC081
64 DRVSM2 EQU SC082

```



```

65  DRVSM4  EQU  SC084
66  DRVSM6  EQU  SC086
67  DRVOFF  EQU  SC088
68  DRVON   EQU  SC089
69  DRVSL1  EQU  SC08A
70  DRVSL2  EQU  SC08B
71  DRVRD   EQU  SC08C
72  DRVWR   EQU  SC08D
73  DRVADM  EQU  SC08E
74  DRVWRM  EQU  SC08F
75  ;
76  ;
77  ;
78  AC090   EQU  SC090
79  ACOBE   EQU  SC0BE
80  ACOBF   EQU  SC0BF
81  ;
82  ;
83  ;
84  AC1C1   EQU  SC1C1
85  AC800   EQU  SC800
86  AC84D   EQU  SC84D
87  AC9AA   EQU  SC9AA
88  ACFFF   EQU  SCFFF
61  ;
62  ;
63  ;
64  AD300   EQU  SD300
65  ;. PAGE
66  ;
67  ;
68  ;
69      ORG  $D000
70      INCLUDE  PCIINTERP.IAS
  1  ;. PAGE 'P-code routine address table'
  2  ;
  3  ;
  4  ;
D000 BBD6   5  PCTBL  DW  ABI
D002 B2EC   6      DW  ABR
D004 D9D6   7      DW  ADI
D006 C2EA   8      DW  ADR
D008 6BD5   9      DW  LAND
D00A 57DB  10     DW  DI F
D00C 39D8  11     DW  DVI
D00E 5AEB  12     DW  DVR
D010 7ED8  13     DW  CHK
D012 3FED  14     DW  FLO
D014 62ED  15     DW  FLT
D016 55DC  16     DW  INN
D018 20DB  17     DW  INT
D01A 7ED5  18     DW  LOR
D01C 66D8  19     DW  MODI
D01E 42D7  20     DW  MULPLY      ; MPI
D020 55EC  21     DW  MPR
D022 F1D6  22     DW  NGI
D024 COEC  23     DW  NGR

```

D026	91D5	24	DW	LNOT
D028	CCDC	25	DW	SRS
D02A	03D7	26	DW	SBI
D02C	09EB	27	DW	SBR
D02E	BADC	28	DW	SGS
D030	89D7	29	DW	SQI
D032	7DEC	30	DW	SQR
D034	7BD4	31	DW	STO
D036	48D9	32	DW	IXS
D038	79DB	33	DW	UNI
D03A	01D4	34	DW	LDE
D03C	30E6	35	DW	CSP
D03E	96D2	36	DW	LDCN
D040	E5DB	37	DW	ADJ
D042	5FD2	38	DW	FJP
D044	87D9	39	DW	INC
D046	6BD9	40	DW	IND
D048	9AD9	41	DW	IXA
D04A	43D3	42	DW	LAO
D04C	E5D8	43	DW	LSA
D04E	4BD4	44	DW	LAE
D050	57D5	45	DW	MOV
D052	25D3	46	DW	LDO
D054	07D9	47	DW	SAS
D056	69D3	48	DW	SRO
D058	9ED5	49	DW	XJP
D05A	3FE3	50	DW	RNP
D05C	53E2	51	DW	CIP
D05E	E8DD	52	DW	EQU
D060	E0DD	53	DW	GEQ
D062	D8DD	54	DW	GRT
D064	ADD3	55	DW	LDA
D066	95D4	56	DW	LDC
D068	E4DD	57	DW	LEQ
D06A	DCDD	58	DW	LES
D06C	87D3	59	DW	LOD
D06E	D4DD	60	DW	NEQ
D070	DBD3	61	DW	STR
D072	67D2	62	DW	UJP
D074	1CDA	63	DW	LDP
D076	72DA	64	DW	STP
D078	C8D4	65	DW	LDM
D07A	F6D4	66	DW	STM
D07C	23D5	67	DW	LDB
D07E	3DD5	68	DW	STB
D080	D9D9	69	DW	IXP
D082	2AE3	70	DW	RBP
D084	F9E2	71	DW	CBP
D086	65DF	72	DW	EQUI
D088	37DF	73	DW	GEQI
D08A	2FDF	74	DW	GRTI
D08C	D4D2	75	DW	LLA
D08E	9DD2	76	DW	LDCI
D090	33DF	77	DW	LEQI
D092	2BDF	78	DW	LESI
D094	B6D2	79	DW	LDL
D096	3BDF	80	DW	NEQI

DO98	FAD2	81	DW	STL	
DO9A	D4E2	82	DW	CXP	
DO9C	A1E2	83	DW	CLP	
DO9E	BDE2	84	DW	CGP	
DOAO	CDD8	85	DW	LPA	
DOA2	26D4	86	DW	STE	
DOA4	4DD2	87	DW	UPDBY1	; NOP
DOA6	EFD1	88	DW	ERR11	; EFJ
DOA8	EFD1	89	DW	ERR11	; NFJ
DOAA	2BE8	90	DW	BPT	
DOAC	AOD6	91	DW	XIT	
DOAE	4DD2	92	DW	UPDBY1	; NOP
DOBO	A9D2	93	DW	SLDL	
DOB2	A9D2	94	DW	SLDL	
DOB4	A9D2	95	DW	SLDL	
DOB6	A9D2	96	DW	SLDL	
DOB8	A9D2	97	DW	SLDL	
DOBA	A9D2	98	DW	SLDL	
DOBC	A9D2	99	DW	SLDL	
DOBE	A9D2	100	DW	SLDL	
DOCO	A9D2	101	DW	SLDL	
DOC2	A9D2	102	DW	SLDL	
DOC4	A9D2	103	DW	SLDL	
DOC6	A9D2	104	DW	SLDL	
DOC8	A9D2	105	DW	SLDL	
DOCA	A9D2	106	DW	SLDL	
DOCC	A9D2	107	DW	SLDL	
DOCE	A9D2	108	DW	SLDL	
DODO	18D3	109	DW	SLDO	
DOD2	18D3	110	DW	SLDO	
DOD4	18D3	111	DW	SLDO	
DOD6	18D3	112	DW	SLDO	
DOD8	18D3	113	DW	SLDO	
DODA	18D3	114	DW	SLDO	
DODC	18D3	115	DW	SLDO	
DODE	18D3	116	DW	SLDO	
DOEO	18D3	117	DW	SLDO	
DOE2	18D3	118	DW	SLDO	
DOE4	18D3	119	DW	SLDO	
DOE6	18D3	120	DW	SLDO	
DOE8	18D3	121	DW	SLDO	
DOEA	18D3	122	DW	SLDO	
DOEC	18D3	123	DW	SLDO	
DOEE	18D3	124	DW	SLDO	
DOFO	6AD4	125	DW	SI NDO	
DOF2	67D4	126	DW	SI ND	
DOF4	67D4	127	DW	SI ND	
DOF6	67D4	128	DW	SI ND	
DOF8	67D4	129	DW	SI ND	
DOFA	67D4	130	DW	SI ND	
DOFC	67D4	131	DW	SI ND	
DOFE	67D4	132	DW	SI ND	
		133	;.	PAGE	'Call system procedure routine address table'
		134	;		
		135	;		
		136	;		
D100	04EF	137	CSPTBL	DW	CSPO

```

D102 2FD6      138    DW    NEW
D104 A0E8      139    DW    MOVELR
D106 A0E8      140    DW    MOVELR
D108 84E7      141    DW    EXIT
D10A 69F0      142    DW    UNITRD
D10C 6EFO      143    DW    UNITWR
                144    ifndef ORIG
                145    DW    ERR11
                146    DW    ERR11
                147    else
D10E 3AE6      148    ifndef IIc
D110 40E6      149    DW    CSP7
                150    DW    TREES
                151    else
                152    DW    ERR11
                153    DW    ERR11
                154    endif
                155    endif
D112 41E8      156    DW    TIME
D114 B2E6      157    DW    FILLC
D116 F7E6      158    DW    SCAN
D118 27EF      159    DW    CSP12
D11A 0000      160    DW    0
D11C 0000      161    DW    0
D11E 0000      162    DW    0
D120 0000      163    DW    0
D122 0000      164    DW    0
D124 0000      165    DW    0
D126 0000      166    DW    0
D128 0000      167    DW    0
D12A 1CE6      168    DW    CSP21
D12C 26E6      169    DW    CSP22
D12E DOED      170    DW    TNC
D130 BBED      171    DW    RND
D132 EFD1      172    DW    ERR11 ; 25
D134 EFD1      173    DW    ERR11 ; 26
D136 EFD1      174    DW    ERR11 ; 27
D138 EFD1      175    DW    ERR11 ; 28
D13A EFD1      176    DW    ERR11 ; 29
D13C EFD1      177    DW    ERR11 ; 30
D13E EFD1      178    DW    ERR11 ; 31
D140 6BD6      179    DW    MRK
D142 82D6      180    DW    RLS
D144 F9EE      181    DW    PSHIOR
D146 OFEF      182    DW    CSP35
D148 E5ED      183    DW    PWR10
D14A 1DEF      184    DW    CSP37
D14C A5EF      185    DW    CSP38
D14E 33E8      186    DW    CSP39
D150 04E9      187    DW    FRE ; Return number of words of memory available
                188    ;
                189    ; Go to 'Move Initialization Code'
                190    ;
D152 4C75F2    191    JMOVIC  JMP  MINITC
                192    ;.PAGE 'Extract operand to memory'
                193    ;
                194    ;

```

```

195 ;
196 EBI G2M   RDAUX
    +   I FDEF BI G
    +   STA   RAMRDS
    +   ENDI F
197 ;
198 ;
199 ;
D155 B158    200   LDA   (I PCPTR) , Y
D157 1010    201   BNM   INXAT
202 ;
203 ;
204 ;
D159 855F    205   STA   OPRND+1
206 ;
207 ;
208 ;
D15B E658    209   INC   I PCPTR
D15D D002    210   BNZ   INXAS
D15F E659    211   INC   I PCPTR+1
212 ;
213 ;
214 ;
D161 B158    215   INXAS  LDA   (I PCPTR) , Y
D163 0A      216   ASL   A
D164 855E    217   STA   OPRND
D166 265F    218   ROL   OPRND+1
219 ;
220 ;
221 ;
222   RDMAI N
    +   I FDEF BI G
    +   STA   RAMRDC
    +   ENDI F
223 ;
224 ;
225 ;
D168 60      226   RTS
227 ;
228 ;
229 ;
D169 0A      230   INXAT  ASL   A
D16A 855E    231   STA   OPRND
D16C A900    232   LDA   #0
D16E 855F    233   STA   OPRND+1
234 ;
235 ;
236 ;
237   RDMAI N
    +   I FDEF BI G
    +   STA   RAMRDC
    +   ENDI F
238 ;
239 ;
240 ;
D170 60      241   RTS
242   ifndef ORIG

```

```

243 ;.PAGE 'Extract operand to register pair'
244 ;
245 ;
246 ;
247 EBI G2R    RDAUX
248 ;
249 ;
250 ;
251     LDA    (IPCPTR), Y
252     BNM    INXPC
253 ;
254 ;
255 ;
256     TAX
257 ;
258 ; Increment pointer
259 ;
260     INC    IPCPTR
261     BNZ    INXPB
262     INC    IPCPTR+1
263 ;
264 ;
265 ;
266 INXPB     LDA    (IPCPTR), Y
267         ASL    A
268         TAY
269         TXA
270         ROL    A
271         TAX
272         RDMAIN
273 ;
274 ;
275 ;
276     RTS
277 ;
278 ;
279 ;
280 INXPC     ASL    A
281         TAY
282         LDX    #0
283         RDMAIN
284 ;
285 ;
286 ;
287     RTS
288     endi f
289 ;.PAGE 'Subroutines'
290 ;
291 ;
292 ;
D171 A552   293 INXAU     LDA    MSPTR
D173 8562   294         STA    A62
D175 A553   295         LDA    MSPTR+1
D177 8563   296         STA    A62+1
297 ;
298 ;
299 ;

```

```

D179 E000      300      CPX      #0
D17B F012      301      BE       INXAY
                302      ;
                303      ;
                304      ;
D17D A000      305      INXAX     LDY      #0
D17F B162      306      LDA      (A62), Y
D181 8574      307      STA      A74
D183 C8        308      INY
D184 B162      309      LDA      (A62), Y
D186 8563      310      STA      A62+1
D188 A574      311      LDA      A74
D18A 8562      312      STA      A62
                313      ;
                314      ;
                315      ;
D18C CA        316      DEX
D18D DOEE      317      BNZ      INXAX
                318      ;
                319      ;
                320      ;
D18F 60        321      INXAY     RTS
                322      ; . PAGE
                323      ;
                324      ;
                325      ;
D190 ADE6BD    326      INXAZ     LDA      SYCGDP
D193 D006      327      BNZ      INXBB
D195 AEE7BD    328      LDX      SYCGDP+1
D198 D001      329      BNZ      INXBB
                330      ;
                331      ;
                332      ;
D19A 60        333      RTS
                334      ;
                335      ;
                336      ;
D19B 855A      337      INXBB     STA      HPPTR
D19D ADE7BD    338      LDA      SYCGDP+1
D1A0 855B      339      STA      HPPTR+1
                340      ;
                341      ;
                342      ;
D1A2 A900      343      LDA      #0
D1A4 8DE6BD    344      STA      SYCGDP
D1A7 A900      345      LDA      #0
D1A9 8DE7BD    346      STA      SYCGDP+1
                347      ;
                348      ;
                349      ;
D1AC 60        350      RTS
                351      ;
                352      ;
                353      ;
D1AD AFD1      354      INXBC     DW      AD1AF
D1AF FBD1      355      AD1AF     DW      INXBK
D1B1 00FF      356      DW      JCREAD

```

```

D1B3 DEBD      357      DW      IORSLT
D1B5 7400      358      DW      A74
                359      ;. PAGE
                360      ;
                361      ;
                362      ;
D1B7 A901      363      ERR1      LDA      #1
D1B9 1040      364      BNM      INXBK
                365      ;
                366      ;
                367      ;
D1BB A902      368      LDA      #2
D1BD 103C      369      BNM      INXBK
                370      ;
                371      ;
                372      ;
D1BF A903      373      ERR3      LDA      #3
D1C1 1038      374      BNM      INXBK
                375      ;
                376      ;
                377      ;
D1C3 A900      378      OOMERR     LDA      #0
D1C5 8DE6BD    379      STA      SYCGDP
D1C8 8DE7BD    380      STA      SYCGDP+1
D1CB 855A      381      STA      HPPTR
D1CD 855C      382      STA      PSKPTR
D1CF A980      383      LDA      #<A8000
D1D1 855D      384      STA      PSKPTR+1
D1D3 A962      385      LDA      #<A6200
D1D5 855B      386      STA      HPPTR+1
D1D7 A904      387      LDA      #4
                388      ;
                389      ; Uncondi tional branch
                390      ;
D1D9 1020      391      BNM      INXBK
                392      ;
                393      ;
                394      ;
D1DB A905      395      LDA      #5
                396      ;
                397      ; Uncondi tional branch
                398      ;
D1DD 101C      399      BNM      INXBK
                400      ;
                401      ;
                402      ;
D1DF A906      403      ERR6      LDA      #6
                404      ;
                405      ; Uncondi tional branch
                406      ;
D1E1 1018      407      BNM      INXBK
                408      ;
                409      ;
                410      ;
D1E3 A908      411      ERR8      LDA      #8
                412      ;
                413      ; Uncondi tional branch

```



```

D1E5 1014      414 ;
                415 BNM INXBK
                416 ;
                417 ;
                418 ;
D1E7 A909      419 ERR9 LDA #9
                420 ;
                421 ; Uncondi ti onal branch
                422 ;
D1E9 1010      423 BNM INXBK
                424 ;
                425 ;
                426 ;
D1EB A90A      427 ERR10 LDA #10
                428 ;
                429 ; Uncondi ti onal branch
                430 ;
D1ED 100C      431 BNM INXBK
                432 ifdef BIG
                433 ;
                434 ;
                435 ;
                436 ND219 LDA #<A8000
                437 STA BZ60+1
                438 LDA #<A6200
                439 STA BZ62+1
                440 ;
                441 ;
                442 ;
                443 LDA #16
                444 ;
                445 ; Uncondi ti onal branch
                446 ;
                447 BNM INXBK
                448 endif
                449 ;
                450 ;
                451 ;
D1EF A90B      452 ERR11 LDA #11
                453 ;
                454 ; Uncondi ti onal branch
                455 ;
D1F1 1008      456 BNM INXBK
                457 ;
                458 ;
                459 ;
D1F3 A90C      460 ERR12 LDA #12
                461 ;
                462 ; Uncondi ti onal branch
                463 ;
D1F5 1004      464 BNM INXBK
                465 ;
                466 ;
                467 ;
D1F7 A90D      468 ERR13 LDA #13
                469 ;
                470 ; Uncondi ti onal branch

```

```

D1F9 1000    471 ;
              472   BNM   INXBK
              473 ;
              474 ;
              475 ;
              476 INXBK
              477   ifndef ORIG
              478   ifdef  FOUR
              479   STA   ABDEO
              480   LDA   #0
              481   STA   ABDEO+1
              482 ;
              483 ;
              484 ;
              485   STA   RAMRDS
              486 ;
              487 ;
              488 ;
              489   CLB
              490   LDA   JTBPTR
              491   SBB   #2
              492   STA   A90
              493   LDA   JTBPTR+1
              494   SBB   #0
              495   STA   A90+1
              496 ;
              497 ;
              498 ;
              499   JSR   SETA7C
              500 ;
              501 ;
              502 ;
              503   CLB
              504   LDA   IPCPTR
              505   SBB   A7C
              506   STA   ABDF4
              507   LDA   IPCPTR+1
              508   SBB   A7C+1
              509   STA   ABDF4+1
              510 ;
              511 ;
              512 ;
              513   LDY   #0
              514   LDA   (JTBPTR), Y
              515   STA   SYCJTB
              516   LDA   (SEGPTR), Y
              517   STA   SYCSEG
              518   STY   SYCJTB+1
              519   STY   SYCSEG+1
              520 ;
              521 ;
              522 ;
              523   STA   RAMRDC
              524 ;
              525 ;
              526 ;
              527   LDA   #0
    
```

```

528     STA     SEGNO
529     ;
530     ;
531     ;
532     LDA     #2
533     STA     PROCNO
534     ;
535     ;
536     ;
537     JMP     INUCE
538     ;
539     ;
540     ;
541     else
542     CMP     #10
543     BE      INXPD
544     CLC
545     ADC     #>100
546     STA     IORSLT
547     LDA     #<100
548     STA     IORSLT+1
549     INXPD   LDX     #2
550     LDA     SUCTBL, X
551     BNZ     INXPE
552     LDA     SUCTBL+1, X
553     BNZ     INXPE
554     PHA
555     LDA     IORSLT
556     PHA
557     LDA     #0
558     BZ      INXPH
559     INXPE   LDA     #S01
560     LDY     #0
561     RDAUX
562     CMP     (SEGPTR), Y
563     RDMAIN
564     BE      INXPG
565     JSR     LODSEG
566     INXPG   LDA     #1
567     INXPH   STA     SEGNO
568     ;
569     ;
570     ;
571     LDA     #2
572     STA     PROCNO
573     ;
574     ;
575     ;
576     JSR     CALLX
577     ;
578     ;
579     ;
580     LDY     #0
581     LDA     BASPTR
582     STA     (MSPTR), Y
583     INY
584     LDA     BASPTR+1

```

```

585     STA     (MSPTR), Y
586     ;
587     ;
588     ;
589     LDY     #4
590     LDA     (MSPTR), Y
591     STA     A76
592     INY
593     LDA     (MSPTR), Y
594     STA     A76+1
595     ;
596     ;
597     ;
598     LDA     A76
599     CLB
600     SBB     #4
601     STA     A76
602     BNB     INXPI
603     DEC     A76+1
604     ;
605     ;
606     ;
607     INXPI   LDY     #0
608     LDA     A76
609     CLB
610     SBB     (A76), Y
611     LDY     #8
612     STA     (MSPTR), Y
613     LDY     #1
614     LDA     A76+1
615     SBB     (A76), Y
616     LDY     #9
617     STA     (MSPTR), Y
618     ;
619     ;
620     ;
621     endi f
622     endi f
623     ifdef  ORIG
624     ifndef IIc
D1FB 8DE0BD 625     STA     ABDE0
D1FE A900   626     LDA     #0
D200 8DE1BD 627     STA     ABDE0+1
628     ;
629     ;
630     ;
D203 A558   631     LDA     I PCPTR
D205 8DF4BD 632     STA     ABDF4
D208 A559   633     LDA     I PCPTR+1
D20A 8DF5BD 634     STA     ABDF4+1
635     ;
636     ;
637     ;
D20D A55C   638     LDA     PSKPTR
D20F 38     639     CLB
D210 E90E   640     SBB     #14
D212 8DE8BD 641     STA     ABDE8

```

```

D215 A55D      642    LDA    PSKPTR+1
D217 E900      643    SBB    #0
D219 8DE9BD    644    STA    ABDE8+1
              645    ;
              646    ;
              647    ;
D21C AD2CD2    648    LDA    I NXBW
D21F 8558      649    STA    I PCPTR
D221 AD2DD2    650    LDA    I NXBW+1
D224 8559      651    STA    I PCPTR+1
              652    else
              653    CMP    #10
              654    BE    I NXPD
              655    CLC
              656    ADC    #>100
              657    STA    I ORSLT
              658    LDA    #<100
              659    STA    I ORSLT+1
              660    I NXPD    LDX    #2
              661    LDA    SUCTBL, X
              662    BNZ    I NXPE
              663    LDA    SUCTBL+1, X
              664    BNZ    I NXPE
              665    PHA
              666    LDA    I ORSLT
              667    PHA
              668    LDA    #0
              669    BZ    I NXPH
              670    I NXPE    LDA    #S01
              671    LDY    #0
              672    RDAUX
              673    CMP    (SEGPTR), Y
              674    RDMAIN
              675    BE    I NXPG
              676    JSR    LODSEG
              677    I NXPG    LDA    #1
              678    I NXPH    STA    SEGNO
              679    ;
              680    ;
              681    ;
              682    LDA    #2
              683    STA    PROCNO
              684    ;
              685    ;
              686    ;
              687    JSR    CALLX
              688    ;
              689    ;
              690    ;
              691    LDY    #0
              692    LDA    BASPTR
              693    STA    (MSPTR), Y
              694    I NY
              695    LDA    BASPTR+1
              696    STA    (MSPTR), Y
              697    ;
              698    ;

```

```

699 ;
700 LDY #4
701 LDA (MSPTR), Y
702 STA A76
703 INY
704 LDA (MSPTR), Y
705 STA A76+1
706 ;
707 ;
708 ;
709 LDA A76
710 CLB
711 SBB #4
712 STA A76
713 BNB INXPI
714 DEC A76+1
715 ;
716 ;
717 ;
718 INXPI LDY #0
719 LDA A76
720 CLB
721 SBB (A76), Y
722 LDY #8
723 STA (MSPTR), Y
724 LDY #1
725 LDA A76+1
726 SBB (A76), Y
727 LDY #9
728 STA (MSPTR), Y
729 ;
730 ;
731 ;
732 endi f
733 endi f
734 ifndef FOUR
735 ;
736 ;
737 ;
D226 4C53D2 738 JMP EXECPC
739 endi f
740 ifdef ORIG
741 ifndef IIc
742 ;
743 ;
744 ;
D229 CD0002 745 AD229 CMP A0200
D22C 29D2 746 INXBW DW AD229
747 endi f
748 endi f
749 ;. PAGE '???'
750 ;
751 ;
752 ;
D22E A558 753 UPDBY3 LDA I PCPTR
D230 18 754 CLC
D231 6903 755 ADC #3

```

```

D233 8558      756      STA      I PCPTR
D235 901C      757      BNC      EXECPC
D237 E659      758      INC      I PCPTR+1
D239 B018      759      BC       EXECPC
              760
*****
              761      ;
              762      ;
              763      ;
D23B A558      764      UPDBY2   LDA      I PCPTR
D23D 18         765      CLC
D23E 6902      766      ADC      #2
D240 8558      767      STA      I PCPTR
D242 900F      768      BNC      EXECPC
D244 E659      769      INC      I PCPTR+1
D246 B00B      770      BC       EXECPC
              771
*****
              772      ;
              773      ;   Push literal (0..127) onto evaluation stack
              774      ;
D248 AA        775      PSHLI T   TAX
D249 98         776      TYA
D24A 48         777      PHA
D24B 8A         778      TXA
D24C 48         779      PHA
              780      ifndef ORIG
              781      ;
              782      ;
              783      ;
              784      INXPJ      INC      I PCPTR
              785      BNZ      INXPK
              786      INC      I PCPTR+1
              787      INXPK      LDA      (I PCPTR), Y
              788      BNM      PSHLI T
              789      ASL      A
              790      STA      INXPL+1
              791      RDMAIN
              792      INXPL      JMP      (PCTBL)
              793      endif
              794
*****
              795      ;
              796      ;   Increment P-code pointer
              797      ;
D24D E658      798      UPDBY1   INC      I PCPTR
D24F D002      799      BNZ      EXECPC
D251 E659      800      INC      I PCPTR+1
              801
*****
              802      ;
              803      ;   Execute next op code
              804      ;
              805      EXECPC      RDAUX
              +      I FDEF   BIG
              +      STA      RAMRDS
              +      ENDIF

```

```

D253 A000      806      LDY      #0
D255 B158      807      LDA      (I PCPTR), Y
D257 10EF      808      BNM      PSHLI T
D259 0A        809      ASL      A
                810      ifndef  ORIG
                811      STA      INXPM+1
                812      RDMAIN
                813      INXPM   JMP      (PCTBL)
                814      else
D25A 856F      815      STA      JMPPC+1
D25C 4C6E00    816      JMP      JMPPC
                817      endif
                818

```

\*\*\*\*\*

```

                819      ;
                820      ;   Jump if TOS is FALSE
                821      ;
D25F 68        822      FJP      PLA
                823      ifdef  ORIG
D260 AA        824      TAX
D261 68        825      PLA
D262 8A        826      TXA
                827      endif
D263 2901      828      AND      #$01
                829      ifndef  ORIG
                830      BZ      INXPN
                831      PLA
                832      JMP      UPDBY2
                833      else
D265 D0D4      834      BNZ      UPDBY2
                835      endif
                836

```

\*\*\*\*\*

```

                837      ;
                838      ;   Unconditional jump
                839      ;
                840      ifndef  ORIG
                841      INXPN   PLA
                842      endif
D267 18        843      UJP      CLC
                844      RDAUX
                +      IFDEF  BIG
                +      STA      RAMRDS
                +      ENDF
D268 A001      845      LDY      #1
D26A B158      846      LDA      (I PCPTR), Y
                847      RDMAIN
                +      IFDEF  BIG
                +      STA      RAMRDC
                +      ENDF
D26C 300B      848      BM      INXEX
D26E 6558      849      ADC      I PCPTR
D270 8558      850      STA      I PCPTR
D272 90C7      851      BNC      UPDBY2
D274 E659      852      INC      I PCPTR+1
                853      ;
                854      ;

```



```

855 ;
D276 4C3BD2 856 JMP UPDBY2
857 ;
858 ;
859 ;
D279 6554 860 INXEX ADC JTBPTR
D27B 8574 861 STA A74
D27D A9FF 862 LDA #$FF
D27F 6555 863 ADC JTBPTR+1
D281 8575 864 STA A74+1
865 ;
866 ;
867 ;
D283 A574 868 LDA A74
D285 38 869 CLB
D286 A000 870 LDY #0
871 RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
D288 F174 872 SBB (A74), Y
D28A 8558 873 STA IPCPTR
D28C A575 874 LDA A74+1
D28E C8 875 INY
D28F F174 876 SBB (A74), Y
D291 8559 877 STA IPCPTR+1
878 ;
879 ;
880 ;
D293 4C53D2 881 JMP EXECPC
882
*****
883 ;
884 ; Load constant NIL
885 ;
D296 A900 886 LDCN LDA #0
D298 48 887 PHA
D299 48 888 PHA
889 ;
890 ;
891 ;
D29A 4C4DD2 892 JMP UPDBY1
893
*****
894 ;
895 ; Load word literal
896 ;
D29D A002 897 LDCI LDY #2
898 RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
D29F B158 899 LDA (IPCPTR), Y
D2A1 48 900 PHA
D2A2 88 901 DEY
D2A3 B158 902 LDA (IPCPTR), Y
D2A5 48 903 PHA

```

```

D2A6 4C2ED2    904    JMP    UPDBY3
                905
*****
                906    ;
                907    ;   Load short local word
                908    ;
D2A9 E9A3     909    SLDL   SBB    #SA3
D2AB A8       910    TAY
D2AC B152     911    LDA    (MSPTR), Y
D2AE 48       912    PHA
D2AF 88       913    DEY
D2B0 B152     914    LDA    (MSPTR), Y
D2B2 48       915    PHA
                916    ;
                917    ;
                918    ;
D2B3 4C4DD2   919    JMP    UPDBY1
                920
*****
                921    ;
                922    ;   Load local word
                923    ;
D2B6 A001     924    LDL    LDY    #1
                925    i fndef ORIG
                926    JSR    EBIG2R
                927    ;
                928    ;
                929    ;
                930    CLC
                931    TYA
                932    ADC    MSPTR
                933    STA    VARPTR
                934    TXA
                935    ADC    MSPTR+1
                936    el se
D2B8 2055D1   937    JSR    EBIG2M
                938    ;
                939    ;
                940    ;
D2BB A552     941    LDA    MSPTR
D2BD 18       942    CLC
D2BE 655E     943    ADC    OPRND
D2C0 8564     944    STA    VARPTR
D2C2 A553     945    LDA    MSPTR+1
D2C4 655F     946    ADC    OPRND+1
                947    endi f
D2C6 8565     948    STA    VARPTR+1
                949    ;
                950    ;
                951    ;
D2C8 A00B     952    LDY    #11
D2CA B164     953    LDA    (VARPTR), Y
D2CC 48       954    PHA
D2CD 88       955    DEY
D2CE B164     956    LDA    (VARPTR), Y
D2D0 48       957    PHA
                958    ;

```

```

          959 ;
          960 ;
D2D1 4C3BD2 961     JMP     UPDBY2
          962
*****
          963 ;
          964 ;   Load local address
          965 ;
D2D4 A001 966 LLA     LDY     #1
          967   ifndef ORIG
          968     JSR     EBIG2R
          969 ;
          970 ;
          971 ;
          972     CLC
          973     TYA
          974     ADC     MSPTR
          975     TAY
          976     TXA
          977     ADC     MSPTR+1
          978     TAX
          979     CLC
          980     TYA
          981     ADC     #10
          982     TAY
          983     BNC     INXEY
          984     INX
          985 ;
          986 ;
          987 ;
          988 INXEY   TXA
          989     PHA
          990     TYA
          991     PHA
D2D6 2055D1 992   else
          993     JSR     EBIG2M
          994 ;
          995 ;
          996 ;
D2D9 A552 997     LDA     MSPTR
D2DB 18    998     CLC
D2DC 655E 999     ADC     OPRND
D2DE 8564 1000    STA     VARPTR
D2E0 A553 1001    LDA     MSPTR+1
D2E2 655F 1002    ADC     OPRND+1
D2E4 8565 1003    STA     VARPTR+1
          1004 ;
          1005 ;
          1006 ;
D2E6 A564 1007    LDA     VARPTR
D2E8 18    1008    CLC
D2E9 690A 1009    ADC     #10
D2EB 8564 1010    STA     VARPTR
D2ED 9002 1011    BNC     INXEY
D2EF E665 1012    INC     VARPTR+1
          1013 ;
          1014 ;

```

```

1015 ;
D2F1 A565 1016 INXEY LDA  VARPTR+1
D2F3 48 1017 PHA
D2F4 A564 1018 LDA  VARPTR
D2F6 48 1019 PHA
1020 endi f
1021 ;
1022 ;
1023 ;
D2F7 4C3BD2 1024 JMP  UPDBY2
1025
*****
1026 ;
1027 ; Store local word
1028 ;
D2FA A001 1029 STL  LDY  #1
1030 ifndef ORIG
1031 JSR  EBI G2R
1032 CLC
1033 TYA
1034 ADC  MSPTR
1035 STA  VARPTR
1036 TXA
1037 ADC  MSPTR+1
1038 STA  VARPTR+1
1039 else
D2FC 2055D1 1040 JSR  EBI G2M
1041 ;
1042 ;
1043 ;
D2FF A552 1044 LDA  MSPTR
D301 18 1045 CLC
D302 655E 1046 ADC  OPRND
D304 8564 1047 STA  VARPTR
D306 A553 1048 LDA  MSPTR+1
D308 655F 1049 ADC  OPRND+1
D30A 8565 1050 STA  VARPTR+1
1051 endi f
1052 ;
1053 ;
1054 ;
D30C A00A 1055 LDY  #10
D30E 68 1056 PLA
D30F 9164 1057 STA  (VARPTR), Y
D311 C8 1058 INY
D312 68 1059 PLA
D313 9164 1060 STA  (VARPTR), Y
1061 ;
1062 ;
1063 ;
D315 4C3BD2 1064 JMP  UPDBY2
1065
*****
1066 ;
1067 ; Load short global word
1068 ;
D318 E9C3 1069 SLDO  SBB  #$C3

```

```

D31A A8      1070    TAY
D31B B150    1071    LDA    (BASPTR), Y
D31D 48      1072    PHA
D31E 88      1073    DEY
D31F B150    1074    LDA    (BASPTR), Y
D321 48      1075    PHA
                1076    ;
                1077    ;
                1078    ;
D322 4C4DD2  1079    JMP    UPDBY1
                1080
*****
                1081    ;
                1082    ;   Load global word
                1083    ;
D325 A001    1084    LDO     LDY    #1
                1085    i f n d e f  O R I G
                1086    JSR    EBI G2R
                1087    ;
                1088    ;
                1089    ;
                1090    CLC
                1091    TYA
                1092    ADC    BASPTR
                1093    STA    VARPTR
                1094    TXA
                1095    ADC    BASPTR+1
                1096    STA    VARPTR+1
                1097    e l s e
D327 2055D1  1098    JSR    EBI G2M
                1099    ;
                1100    ;
                1101    ;
D32A A550    1102    LDA    BASPTR
D32C 18      1103    CLC
D32D 655E    1104    ADC    OPRND
D32F 8564    1105    STA    VARPTR
D331 A551    1106    LDA    BASPTR+1
D333 655F    1107    ADC    OPRND+1
D335 8565    1108    STA    VARPTR+1
                1109    e n d i f
                1110    ;
                1111    ;
                1112    ;
D337 A00B    1113    LDY    #11
D339 B164    1114    LDA    (VARPTR), Y
D33B 48      1115    PHA
D33C 88      1116    DEY
D33D B164    1117    LDA    (VARPTR), Y
D33F 48      1118    PHA
                1119    ;
                1120    ;
                1121    ;
D340 4C3BD2  1122    JMP    UPDBY2
                1123
*****
                1124    ;

```

```

1125 ; Load global address
1126 ;
D343 A001 1127 LAO LDY #1
1128 ifndef ORIG
1129 JSR EBIG2R
1130 ;
1131 ;
1132 ;
1133 CLC
1134 TYA
1135 ADC BASPTR
1136 TAY
1137 TXA
1138 ADC BASPTR+1
1139 TAX
1140 CLC
1141 TYA
1142 ADC #10
1143 TAY
1144 BNC INXEZ
1145 INC
1146 ;
1147 ;
1148 ;
1149 INXEZ TXA
1150 PHA
1151 TYA
1152 PHA
1153 else
D345 2055D1 1154 JSR EBIG2M
1155 ;
1156 ;
1157 ;
D348 A550 1158 LDA BASPTR
D34A 18 1159 CLC
D34B 655E 1160 ADC OPRND
D34D 8564 1161 STA VARPTR
D34F A551 1162 LDA BASPTR+1
D351 655F 1163 ADC OPRND+1
D353 8565 1164 STA VARPTR+1
1165 ;
1166 ;
1167 ;
D355 A564 1168 LDA VARPTR
D357 18 1169 CLC
D358 690A 1170 ADC #10
D35A 8564 1171 STA VARPTR
D35C 9002 1172 BNC INXEZ
D35E E665 1173 INC VARPTR+1
1174 ;
1175 ;
1176 ;
D360 A565 1177 INXEZ LDA VARPTR+1
D362 48 1178 PHA
D363 A564 1179 LDA VARPTR
D365 48 1180 PHA
1181 endif

```

```

1182 ;
1183 ;
1184 ;
D366 4C3BD2 1185     JMP     UPDBY2
1186
*****
1187 ;
1188 ;   Store global word
1189 ;
D369 A001   1190 SRO     LDY     #1
1191   ifndef ORIG
1192     JSR     EBIG2R
1193 ;
1194 ;
1195 ;
1196     CLC
1197     TYA
1198     ADC     BASPTR
1199     STA     VARPTR
1200     TXA
1201     ADC     BASPTR+1
1202     STA     VARPTR+1
1203   else
D36B 2055D1 1204     JSR     EBIG2M
1205 ;
1206 ;
1207 ;
D36E A550   1208     LDA     BASPTR
D370 18     1209     CLC
D371 655E   1210     ADC     OPRND
D373 8564   1211     STA     VARPTR
D375 A551   1212     LDA     BASPTR+1
D377 655F   1213     ADC     OPRND+1
D379 8565   1214     STA     VARPTR+1
1215   endif
1216 ;
1217 ;
1218 ;
D37B A00A   1219     LDY     #10
D37D 68     1220     PLA
D37E 9164   1221     STA     (VARPTR), Y
D380 C8     1222     INY
D381 68     1223     PLA
D382 9164   1224     STA     (VARPTR), Y
1225 ;
1226 ;
1227 ;
D384 4C3BD2 1228     JMP     UPDBY2
1229
*****
1230 ;
1231 ;   Load intermediate word
1232 ;
1233 LOD     RDAUX
+   IFDEF BIG
+   STA   RAMRDS
+   ENDF

```

```

D387 A001      1234      LDY   #1
D389 B158      1235      LDA   (I PCPTR), Y
              1236      RDMAI N
              +      I FDEF BI G
              +      STA   RAMRDC
              +      ENDI F
D38B AA        1237      TAX
D38C 2071D1    1238      JSR   I NXAU
              1239      ;
              1240      ;
              1241      ;
D38F A002      1242      LDY   #2
D391 2055D1    1243      JSR   EBI G2M
              1244      ;
              1245      ;
              1246      ;
D394 A55E      1247      LDA   OPRND
D396 18        1248      CLC
D397 6562      1249      ADC   A62
D399 8564      1250      STA   VARPTR
D39B A55F      1251      LDA   OPRND+1
D39D 6563      1252      ADC   A62+1
D39F 8565      1253      STA   VARPTR+1
              1254      ;
              1255      ;
              1256      ;
D3A1 A00B      1257      LDY   #11
D3A3 B164      1258      LDA   (VARPTR), Y
D3A5 48        1259      PHA
D3A6 88        1260      DEY
D3A7 B164      1261      LDA   (VARPTR), Y
D3A9 48        1262      PHA
              1263      ;
              1264      ;
              1265      ;
D3AA 4C2ED2    1266      JMP   UPDBY3
              1267
*****
              1268      ;
              1269      ;   Load intermediate address
              1270      ;
              1271      LDA   RDAUX
              +      I FDEF BI G
              +      STA   RAMRDS
              +      ENDI F
D3AD A001      1272      LDY   #1
D3AF B158      1273      LDA   (I PCPTR), Y
              1274      RDMAI N
              +      I FDEF BI G
              +      STA   RAMRDC
              +      ENDI F
D3B1 AA        1275      TAX
D3B2 2071D1    1276      JSR   I NXAU
              1277      ;
              1278      ;
              1279      ;
D3B5 A002      1280      LDY   #2

```



```

D3B7 2055D1    1281    JSR    EBI G2M
                1282    ;
                1283    ;
                1284    ;
D3BA A55E      1285    LDA    OPRND
D3BC 18         1286    CLC
D3BD 6562      1287    ADC    A62
D3BF 8564      1288    STA    VARPTR
D3C1 A55F      1289    LDA    OPRND+1
D3C3 6563      1290    ADC    A62+1
D3C5 8565      1291    STA    VARPTR+1
                1292    ;
                1293    ;
                1294    ;
D3C7 A564      1295    LDA    VARPTR
D3C9 18         1296    CLC
D3CA 690A      1297    ADC    #10
D3CC 8564      1298    STA    VARPTR
D3CE 9002      1299    BNC    INXFA
D3D0 E665      1300    INC    VARPTR+1
                1301    ;
                1302    ;
                1303    ;
D3D2 A565      1304    INXFA    LDA    VARPTR+1
D3D4 48         1305    PHA
D3D5 A564      1306    LDA    VARPTR
D3D7 48         1307    PHA
                1308    ;
                1309    ;
                1310    ;
D3D8 4C2ED2    1311    JMP    UPDBY3
                1312
*****
                1313    ;
                1314    ;    Store intermediate word
                1315    ;
                1316    STR      RDAUX
                +    IFDEF    BIG
                +    STA    RAMRDS
                +    ENDIF
D3DB A001      1317    LDY    #1
D3DD B158      1318    LDA    (IPCPTR), Y
                1319    RDMAIN
                +    IFDEF    BIG
                +    STA    RAMRDC
                +    ENDIF
D3DF AA        1320    TAX
D3E0 2071D1    1321    JSR    INXAU
                1322    ;
                1323    ;
                1324    ;
D3E3 A002      1325    LDY    #2
D3E5 2055D1    1326    JSR    EBI G2M
                1327    ;
                1328    ;
                1329    ;
D3E8 A55E      1330    LDA    OPRND

```

```

D3EA 18      1331    CLC
D3EB 6562    1332    ADC    A62
D3ED 8564    1333    STA    VARPTR
D3EF A55F    1334    LDA    OPRND+1
D3F1 6563    1335    ADC    A62+1
D3F3 8565    1336    STA    VARPTR+1
                1337    ;
                1338    ;
                1339    ;
D3F5 A00A    1340    LDY    #10
D3F7 68      1341    PLA
D3F8 9164    1342    STA    (VARPTR), Y
D3FA C8      1343    INY
D3FB 68      1344    PLA
D3FC 9164    1345    STA    (VARPTR), Y
                1346    ;
                1347    ;
                1348    ;
D3FE 4C2ED2  1349    JMP    UPDBY3
                1350
*****
                1351    ;
                1352    ;   Load extended word
                1353    ;
                1354    LDE      RDAUX
                +   IFDEF  BIG
                +   STA    RAMRDS
                +   ENDF
D401 A001    1355    LDY    #1
D403 B158    1356    LDA    (IPCPTR), Y
                1357    RDMAIN
                +   IFDEF  BIG
                +   STA    RAMRDC
                +   ENDF
D405 0A      1358    ASL    A
D406 AA      1359    TAX
D407 C8      1360    INY
D408 2055D1  1361    JSR    EBIG2M
                1362    ;
                1363    ;
                1364    ;
D40B BD9EBD  1365    LDA    IUTBL, X
D40E 18      1366    CLC
D40F 655E    1367    ADC    OPRND
D411 8574    1368    STA    A74
D413 BD9FBD  1369    LDA    IUTBL+1, X
D416 655F    1370    ADC    OPRND+1
D418 8575    1371    STA    A74+1
                1372    ;
                1373    ;
                1374    ;
D41A A001    1375    LDY    #1
D41C B174    1376    LDA    (A74), Y
D41E 48      1377    PHA
D41F 88      1378    DEY
D420 B174    1379    LDA    (A74), Y
D422 48      1380    PHA

```

```

1381 ;
1382 ;
1383 ;
D423 4C2ED2 1384 JMP UPDBY3
1385
*****
1386 ;
1387 ; Store extended word
1388 ;
1389 STE RDAUX
+ I FDEF BI G
+ STA RAMRDS
+ ENDI F
D426 A001 1390 LDY #1
D428 B158 1391 LDA (I PCPTR), Y
1392 RDMAI N
+ I FDEF BI G
+ STA RAMRDC
+ ENDI F
D42A 0A 1393 ASL A
D42B AA 1394 TAX
D42C C8 1395 I NY
D42D 2055D1 1396 JSR EBI G2M
1397 ;
1398 ;
1399 ;
D430 BD9EBD 1400 LDA I UTBL, X
D433 18 1401 CLC
D434 655E 1402 ADC OPRND
D436 8574 1403 STA A74
D438 BD9FBD 1404 LDA I UTBL+1, X
D43B 655F 1405 ADC OPRND+1
D43D 8575 1406 STA A74+1
1407 ;
1408 ;
1409 ;
D43F A000 1410 LDY #0
D441 68 1411 PLA
D442 9174 1412 STA (A74), Y
D444 C8 1413 I NY
D445 68 1414 PLA
D446 9174 1415 STA (A74), Y
1416 ;
1417 ;
1418 ;
D448 4C2ED2 1419 JMP UPDBY3
1420
*****
1421 ;
1422 ; Load extended address
1423 ;
1424 LAE RDAUX
+ I FDEF BI G
+ STA RAMRDS
+ ENDI F
D44B A001 1425 LDY #1
D44D B158 1426 LDA (I PCPTR), Y

```

```

1427      RDMAIN
          +      I FDEF BIG
          +      STA  RAMRDC
          +      ENDI F
D44F 0A      1428      ASL  A
D450 AA      1429      TAX
D451 C8      1430      I NY
D452 2055D1  1431      JSR  EBIG2M
          1432      ;
          1433      ;
          1434      ;
D455 BD9EBD  1435      LDA  IUTBL, X
D458 18      1436      CLC
D459 655E    1437      ADC  OPRND
D45B A8      1438      TAY
D45C BD9FBD  1439      LDA  IUTBL+1, X
D45F 655F    1440      ADC  OPRND+1
D461 48      1441      PHA
D462 98      1442      TYA
D463 48      1443      PHA
          1444      ;
          1445      ;
          1446      ;
D464 4C2ED2  1447      JMP  UPDBY3
          1448      ;.PAGE 'Short index and load word
          1449      ;
          1450      ;
          1451      ;
D467 E9F0    1452      SIND  SBB  #$F0
D469 A8      1453      TAY
D46A 68      1454      SINDO PLA
D46B 8574    1455      STA  A74
D46D 68      1456      PLA
D46E 8575    1457      STA  A74+1
          1458      ;
          1459      ;
          1460      ;
D470 C8      1461      I NY
D471 B174    1462      LDA  (A74), Y
D473 48      1463      PHA
D474 88      1464      DEY
D475 B174    1465      LDA  (A74), Y
D477 48      1466      PHA
          1467      ;
          1468      ;
          1469      ;
D478 4C4DD2  1470      JMP  UPDBY1
          1471
*****
          1472      ;
          1473      ; Store indirect word
          1474      ;
D47B 68      1475      STO  PLA
D47C 8574    1476      STA  A74
D47E 68      1477      PLA
D47F 8575    1478      STA  A74+1
          1479      ;

```

```

1480 ;
1481 ;
D481 68      1482     PLA
D482 8576    1483     STA   A76
D484 68      1484     PLA
D485 8577    1485     STA   A76+1
1486 ;
1487 ;
1488 ;
D487 A000    1489     LDY   #0
D489 A574    1490     LDA   A74
D48B 9176    1491     STA   (A76), Y
D48D C8      1492     INY
D48E A575    1493     LDA   A74+1
D490 9176    1494     STA   (A76), Y
1495 ;
1496 ;
1497 ;
D492 4C4DD2  1498     JMP   UPDBY1
1499
*****
1500 ;
1501 ;   Load multiple word address
1502 ;
1503 LDC      RDAUX
      +     IFDEF BIG
      +     STA   RAMRDS
      +     ENDF
1504 ifndef  ORIG
1505     LDY   #1
1506 else
D495 C8      1507     INY
1508 endif
D496 B158    1509     LDA   (IPCPTR), Y
1510     RDMAIN
      +     IFDEF BIG
      +     STA   RAMRDC
      +     ENDF
D498 AA      1511     TAX
D499 A8      1512     TAY
D49A C8      1513     INY
D49B 98      1514     TYA
D49C 0A      1515     ASL   A
D49D 8574    1516     STA   A74
1517 ;
1518 ;
1519 ;
D49F A558    1520     LDA   IPCPTR
D4A1 2901    1521     AND   #S01
D4A3 F006    1522     BZ   INXFB
1523 ;
1524 ;   Increment pointer
1525 ;
D4A5 E658    1526     INC   IPCPTR
D4A7 D002    1527     BNZ   INXFB
D4A9 E659    1528     INC   IPCPTR+1
1529 ;

```

```

1530 ;
1531 ;
D4AB A003 1532 INXFB LDY #3
1533 RDAUX
+ I FDEF BIG
+ STA RAMRDS
+ ENDI F
D4AD B158 1534 INXFD LDA (I PCPTR), Y
D4AF 48 1535 PHA
D4B0 88 1536 DEY
D4B1 B158 1537 LDA (I PCPTR), Y
D4B3 48 1538 PHA
D4B4 C8 1539 INY
D4B5 C8 1540 INY
D4B6 C8 1541 INY
D4B7 CA 1542 DEX
D4B8 DOF3 1543 BNZ INXFD
1544 ;
1545 ;
1546 ;
D4BA A558 1547 LDA I PCPTR
D4BC 18 1548 CLC
D4BD 6574 1549 ADC A74
D4BF 8558 1550 STA I PCPTR
D4C1 9002 1551 BNC INXFE
D4C3 E659 1552 INC I PCPTR+1
1553 ;
1554 ;
1555 ;
D4C5 4C53D2 1556 INXFE JMP EXECPC
1557

```

\*\*\*\*\*

```

1558 ;
1559 ; Load multiple words
1560 ;
1561 LDM RDAUX
+ I FDEF BIG
+ STA RAMRDS
+ ENDI F
1562 i fndef ORIG
1563 LDY #1
1564 el se
D4C8 C8 1565 INY
1566 endi f
D4C9 B158 1567 LDA (I PCPTR), Y
1568 RDMAIN
+ I FDEF BIG
+ STA RAMRDC
+ ENDI F
D4CB 8574 1569 STA A74
D4CD 0A 1570 ASL A
D4CE A8 1571 TAY
D4CF 8576 1572 STA A76
1573 ;
1574 ;
1575 ;
D4D1 BA 1576 TSX

```

```

D4D2 8A      1577    TXA
D4D3 38      1578    CLB
D4D4 E576    1579    SBB    A76
D4D6 901B    1580    BB     INXFG
D4D8 E908    1581    SBB    #8
D4DA 9017    1582    BB     INXFG
                1583    ;
                1584    ;   Initialize loop count
                1585    ;
D4DC A674    1586    LDX    A74
                1587    ;
                1588    ;
                1589    ;
D4DE 88      1590    DEY
                1591    ;
                1592    ;   Set source pointer
                1593    ;
D4DF 68      1594    PLA
D4E0 8568    1595    STA    SRCPTR
D4E2 68      1596    PLA
D4E3 8569    1597    STA    SRCPTR+1
                1598    ;
                1599    ;
                1600    ;
D4E5 B168    1601    INXFF   LDA    (SRCPTR), Y
D4E7 48      1602    PHA
D4E8 88      1603    DEY
D4E9 B168    1604    LDA    (SRCPTR), Y
D4EB 48      1605    PHA
D4EC 88      1606    DEY
D4ED CA      1607    DEX
D4EE DOF5    1608    BNZ    INXFF
                1609    ;
                1610    ;
                1611    ;
D4F0 4C3BD2  1612    JMP    UPDBY2
                1613    ;
                1614    ;
                1615    ;
D4F3 4CC3D1  1616    INXFG   JMP    OOMERR
                1617
*****
                1618    ;
                1619    ;   Store multiple words
                1620    ;
                1621    STM     RDAUX
                +     IFDEF  BIG
                +     STA    RAMRDS
                +     ENDF
                1622    ifndef ORIG
                1623    LDY    #1
                1624    else
D4F6 C8      1625    INY
                1626    endif
D4F7 B158    1627    LDA    (IPCPTR), Y
                1628    RDMAIN
                +     IFDEF  BIG

```

```

          + STA RAMRDC
          + ENDF
D4F9 8574 1629 STA A74
          1630 ;
          1631 ;
          1632 ;
D4FB BA 1633 TSX
D4FC 8A 1634 TXA
D4FD 18 1635 CLC
D4FE 6574 1636 ADC A74
D500 6574 1637 ADC A74
D502 AA 1638 TAX
D503 BD0101 1639 LDA P1BASE+1, X
D506 856A 1640 STA DSTPTR
D508 BD0201 1641 LDA P1BASE+2, X
D50B 856B 1642 STA DSTPTR+1
D50D 88 1643 DEY
          1644 ;
          1645 ; Initialize loop counter
          1646 ;
D50E A674 1647 LDX A74
          1648 ;
          1649 ; Go see if counter starts at zero
          1650 ;
D510 4C1CD5 1651 JMP INXFJ
          1652 ;
          1653 ;
          1654 ;
D513 68 1655 INXFI PLA
D514 916A 1656 STA (DSTPTR), Y
D516 C8 1657 INY
D517 68 1658 PLA
D518 916A 1659 STA (DSTPTR), Y
D51A C8 1660 INY
D51B CA 1661 DEX
D51C D0F5 1662 INXFJ BNZ INXFI
          1663 ;
          1664 ; Fix stack
          1665 ;
D51E 68 1666 PLA
D51F 68 1667 PLA
          1668 ;
          1669 ;
          1670 ;
D520 4C3BD2 1671 JMP UPDBY2
          1672
*****
          1673 ;
          1674 ; Load byte
          1675 ;
          1676 LDB
          1677 ifndef ORIG
          1678 TSX
          1679 CLC
          1680 PLA
          1681 ADC P1BASE+3, X
          1682 STA A74

```



```

1683   PLA
1684   ADC   P1BASE+4, X
1685   STA   A74+1
1686   TYA
1687   STA   P1BASE+4, X
1688   LDA   (A74), Y
1689   STA   P1BASE+3, X
1690   else
D523 68   1691   PLA
D524 8574 1692   STA   A74
D526 68   1693   PLA
D527 8575 1694   STA   A74+1
1695   ;
1696   ;
1697   ;
D529 18   1698   CLC
D52A 68   1699   PLA
D52B 6574 1700   ADC   A74
D52D 8574 1701   STA   A74
D52F 68   1702   PLA
D530 6575 1703   ADC   A74+1
D532 8575 1704   STA   A74+1
D534 A900 1705   LDA   #0
D536 48   1706   PHA
D537 B174 1707   LDA   (A74), Y
D539 48   1708   PHA
1709   endi f
1710   ;
1711   ;
1712   ;
D53A 4C4DD2 1713   JMP   UPDBY1
1714
*****
1715   ;
1716   ;   Store byte
1717   ;
D53D 68   1718   STB   PLA
D53E AA   1719   TAX
D53F 68   1720   PLA
1721   ;
1722   ;
1723   ;
D540 68   1724   PLA
D541 8574 1725   STA   A74
D543 68   1726   PLA
D544 8575 1727   STA   A74+1
1728   ;
1729   ;
1730   ;
D546 18   1731   CLC
D547 68   1732   PLA
D548 6574 1733   ADC   A74
D54A 8574 1734   STA   A74
D54C 68   1735   PLA
D54D 6575 1736   ADC   A74+1
D54F 8575 1737   STA   A74+1
D551 8A   1738   TXA

```

```

D552 9174      1739      STA   (A74), Y
              1740      ;
              1741      ;
              1742      ;
D554 4C4DD2    1743      JMP   UPDBY1
              1744
*****
              1745      ;
              1746      ;   Move words
              1747      ;
D557 68       1748      MOV   PLA
D558 8568     1749      STA   SRCPTR
D55A 68       1750      PLA
D55B 8569     1751      STA   SRCPTR+1
              1752      ;
              1753      ;
              1754      ;
D55D 68       1755      PLA
D55E 856A     1756      STA   DSTPTR
D560 68       1757      PLA
D561 856B     1758      STA   DSTPTR+1
              1759      ;
              1760      ;
              1761      ;
D563 A001     1762      LDY   #1
D565 2055D1   1763      JSR   EBIG2M
              1764      ;
              1765      ;
              1766      ;
D568 4CDDE8   1767      JMP   AE8DD
              1768
*****
              1769      ;
              1770      ;   Logical AND
              1771      ;
              1772      LAND
              1773      ifndef ORIG
              1774      TSX
              1775      LDA   P1BASE+3, X
              1776      AND   P1BASE+1, X
              1777      STA   P1BASE+3, X
              1778      LDA   P1BASE+4, X
              1779      AND   P1BASE+2, X
              1780      STA   P1BASE+4, X
              1781      INX
              1782      INX
              1783      TXS
              1784      else
D56B 68       1785      PLA
D56C 8574     1786      STA   A74
D56E 68       1787      PLA
D56F 8575     1788      STA   A74+1
D571 68       1789      PLA
D572 AA      1790      TAX
D573 68       1791      PLA
D574 2575     1792      AND   A74+1
D576 48       1793      PHA

```

```

D577 8A      1794    TXA
D578 2574    1795    AND   A74
D57A 48      1796    PHA
                1797    endi f
                1798    ;
                1799    ;
                1800    ;
D57B 4C4DD2 1801    JMP   UPDBY1
                1802
*****
                1803    ;
                1804    ;   Logi cal  OR
                1805    ;
                1806    LOR
                1807    i fndef  ORIG
                1808    TSX
                1809    LDA   P1BASE+3, X
                1810    ORA   P1BASE+1, X
                1811    STA   P1BASE+3, X
                1812    LDA   P1BASE+4, X
                1813    ORA   P1BASE+2, X
                1814    STA   P1BASE+4, X
                1815    INX
                1816    INX
                1817    TXS
                1818    el se
D57E 68      1819    PLA
D57F 8574    1820    STA   A74
D581 68      1821    PLA
D582 8575    1822    STA   A74+1
D584 68      1823    PLA
D585 AA      1824    TAX
D586 68      1825    PLA
D587 0575    1826    ORA   A74+1
D589 48      1827    PHA
D58A 8A      1828    TXA
D58B 0574    1829    ORA   A74
D58D 48      1830    PHA
                1831    endi f
                1832    ;
                1833    ;
                1834    ;
D58E 4C4DD2 1835    JMP   UPDBY1
                1836
*****
                1837    ;
                1838    ;   Logi cal  NOT
                1839    ;
D591 68      1840    LNOT   PLA
D592 AA      1841    TAX
D593 68      1842    PLA
D594 49FF    1843    EOR   #$FF
D596 48      1844    PHA
D597 8A      1845    TXA
D598 49FF    1846    EOR   #$FF
D59A 48      1847    PHA
                1848    ;

```

```

1849 ;
1850 ;
D59B 4C4DD2 1851     JMP     UPDBY1
1852     ;. PAGE 'Case j ump'
1853 ;
1854 ;
1855 ;
1856 XJP     RDAUX
      +     I FDEF BIG
      +     STA  RAMRDS
      +     ENDI F
1857 ;
1858 ; Insure P-code pointer is odd
1859 ;
D59E A901   1860     LDA  #%00000001
D5A0 0558   1861     ORA  I PCPTR
D5A2 8558   1862     STA  I PCPTR
1863 ;
1864 ; Extract case value
1865 ;
D5A4 68     1866     PLA
D5A5 858C   1867     STA  A8C
D5A7 68     1868     PLA
D5A8 858D   1869     STA  A8C+1
1870 ;
1871 ; Compare case value agai nst low bound
1872 ;
D5AA A002   1873     LDY  #2
D5AC 5158   1874     EOR  (I PCPTR), Y
D5AE 3011   1875     BM   I NXFK
D5B0 B158   1876     LDA  (I PCPTR), Y
D5B2 C58D   1877     CMP  A8C+1
D5B4 900F   1878     BL   I NXFL
D5B6 D069   1879     BNE  OUTBND
D5B8 88     1880     DEY
D5B9 A58C   1881     LDA  A8C
D5BB D158   1882     CMP  (I PCPTR), Y
D5BD B006   1883     BGE  I NXFL
D5BF 9060   1884     BL   OUTBND
D5C1 A58D   1885     I NXFK LDA  A8C+1
D5C3 305C   1886     BM   OUTBND
1887 ;
1888 ; Compare case value agai nst hi gh bound
1889 ;
D5C5 A004   1890     I NXFL LDY  #4
D5C7 B158   1891     LDA  (I PCPTR), Y
D5C9 458D   1892     EOR  A8C+1
D5CB 3011   1893     BM   I NXFN
D5CD A58D   1894     LDA  A8C+1
D5CF D158   1895     CMP  (I PCPTR), Y
D5D1 900F   1896     BL   I NXFO
D5D3 D04C   1897     BNE  OUTBND
D5D5 88     1898     DEY
D5D6 B158   1899     LDA  (I PCPTR), Y
D5D8 C58C   1900     CMP  A8C
D5DA B006   1901     BGE  I NXFO
D5DC 9043   1902     BL   OUTBND

```

```

D5DE A58D      1903  INXFN   LDA   A8C+1
D5E0 103F      1904      BNM   OUTBND
                1905  ;
                1906  ; Subtract low bound from case value
                1907  ;
D5E2 38        1908  INXFO   CLB
D5E3 A001      1909      LDY   #1
D5E5 A58C      1910      LDA   A8C
D5E7 F158      1911      SBB   (I PCPTR), Y
D5E9 858C      1912      STA   A8C
D5EB A58D      1913      LDA   A8C+1
D5ED C8        1914      INY
D5EE F158      1915      SBB   (I PCPTR), Y
D5F0 858D      1916      STA   A8C+1
                1917  ;
                1918  ; Multiply result by 2
                1919  ;
D5F2 068C      1920      ASL   A8C
D5F4 268D      1921      ROL   A8C+1
                1922  ;
                1923  ;
                1924  ;
D5F6 A558      1925      LDA   I PCPTR
D5F8 18         1926      CLC
D5F9 658C      1927      ADC   A8C
D5FB 858C      1928      STA   A8C
D5FD A559      1929      LDA   I PCPTR+1
D5FF 658D      1930      ADC   A8C+1
D601 858D      1931      STA   A8C+1
                1932  ;
                1933  ;
                1934  ;
D603 A58C      1935      LDA   A8C
D605 18         1936      CLC
D606 6907      1937      ADC   #7
D608 858C      1938      STA   A8C
D60A 9002      1939      BNC   I NXFQ
D60C E68D      1940      INC   A8C+1
                1941  ;
                1942  ;
                1943  ;
D60E A000      1944  INXFQ   LDY   #0
D610 38        1945      CLB
D611 A58C      1946      LDA   A8C
D613 F18C      1947      SBB   (A8C), Y
D615 8558      1948      STA   I PCPTR
D617 A58D      1949      LDA   A8C+1
D619 C8        1950      INY
D61A F18C      1951      SBB   (A8C), Y
D61C 8559      1952      STA   I PCPTR+1
                1953  ;
                1954  ; Go execute next op code
                1955  ;
D61E 4C53D2    1956      JMP   EXECPC
                1957  ;
                1958  ; Add 5 to pointer
                1959  ;

```

```

D621 A558      1960  OUTBND  LDA   IPCPTR
D623 18        1961      CLC
D624 6905     1962      ADC   #5
D626 8558     1963      STA   IPCPTR
D628 9002     1964      BNC   INXFX
D62A E659     1965      INC   IPCPTR+1
1966 ;
1967 ;   Go execute next op code
1968 ;
D62C 4C53D2   1969  INXFX   JMP   EXECPC
1970 ;.PAGE 'Allocate variable on heap'
1971 ;
1972 ;
1973 ;
D62F 2090D1   1974  NEW     JSR   INXAZ
1975 ;
1976 ;   Pull number of words to allocate
1977 ;
D632 68       1978      PLA
D633 8574     1979      STA   A74
D635 68       1980      PLA
D636 8575     1981      STA   A74+1
1982 ;
1983 ;   Pull address of pointer variable
1984 ;
D638 68       1985      PLA
D639 8576     1986      STA   A76
D63B 68       1987      PLA
D63C 8577     1988      STA   A76+1
1989 ;
1990 ;   Word count * 2 = byte count
1991 ;
D63E 0674     1992      ASL   A74
D640 2675     1993      ROL   A74+1
1994 ;
1995 ;   Move current heap pointer to variable pointer
1996 ;
D642 A000     1997      LDY   #0
D644 A55A     1998      LDA   HPPTR
D646 9176     1999      STA   (A76), Y
D648 C8       2000      INY
D649 A55B     2001      LDA   HPPTR+1
D64B 9176     2002      STA   (A76), Y
2003 ;
2004 ;   Update heap pointer
2005 ;
D64D A55A     2006      LDA   HPPTR
D64F 18       2007      CLC
D650 6574     2008      ADC   A74
D652 855A     2009      STA   HPPTR
D654 A55B     2010      LDA   HPPTR+1
D656 6575     2011      ADC   A74+1
D658 855B     2012      STA   HPPTR+1
2013 ;
2014 ;   Branch if not out of memory
2015 ;
D65A 38       2016      CLB

```

```

D65B A55C      2017    LDA    PSKPTR
D65D E55A      2018    SBB    HPPTR
D65F A55D      2019    LDA    PSKPTR+1
D661 E55B      2020    SBB    HPPTR+1
D663 B003      2021    BNB    INXFY
                2022    ;
                2023    ;   Go to error logic
                2024    ;
D665 4CC3D1    2025    JMP    OOMERR
                2026    ;
                2027    ;   Go update P-code pointer and execute next P-code
                2028    ;
D668 4C3BD2    2029    INXFY    JMP    UPDBY2
                2030    ;.PAGE 'Mark heap or release heap'
                2031    ;
                2032    ;
                2033    ;
D66B 2090D1    2034    MRK      JSR    INXAZ
                2035    ;
                2036    ;
                2037    ;
D66E 68        2038    PLA
D66F 8574      2039    STA    A74
D671 68        2040    PLA
D672 8575      2041    STA    A74+1
                2042    ;
                2043    ;
                2044    ;
D674 A000      2045    LDY    #0
D676 A55A      2046    LDA    HPPTR
D678 9174      2047    STA    (A74), Y
D67A C8        2048    INY
D67B A55B      2049    LDA    HPPTR+1
D67D 9174      2050    STA    (A74), Y
                2051    ;
                2052    ;   Go update P-code pointer and execute next P-code
                2053    ;
D67F 4C3BD2    2054    JMP    UPDBY2
                2055
*****
                2056    ;
                2057    ;
                2058    ;
D682 68        2059    RLS     PLA
D683 8574      2060    STA    A74
D685 68        2061    PLA
D686 8575      2062    STA    A74+1
                2063    ;
                2064    ;
                2065    ;
D688 A000      2066    LDY    #0
D68A B174      2067    LDA    (A74), Y
D68C 855A      2068    STA    HPPTR
D68E C8        2069    INY
D68F B174      2070    LDA    (A74), Y
D691 855B      2071    STA    HPPTR+1
                2072    ;

```

```

                2073 ;
                2074 ;
D693 A900      2075 LDA #0
D695 8DE6BD    2076 STA SYCGDP
D698 A900      2077 LDA #0
D69A 8DE7BD    2078 STA SYCGDP+1
                2079 ;
                2080 ; Go update P-code pointer and execute next P-code
                2081 ;
D69D 4C3BD2    2082 JMP UPDBY2
                2083 ;.PAGE 'Exit operating system'
                2084 ;
                2085 ; Move store instruction to $00..$02
                2086 ;
D6A0 A9AD      2087 XIT LDA #$AD
D6A2 8500      2088 STA A00
D6A4 A98A      2089 LDA #>AC08A
D6A6 8501      2090 STA A00+1
D6A8 A9C0      2091 LDA #<AC08A
D6AA 8502      2092 STA A00+2
                2093 ;
                2094 ; Move indirect jump instruction to $03..$05
                2095 ;
D6AC A96C      2096 LDA #$6C
D6AE 8503      2097 STA A00+3
D6B0 A9FC      2098 LDA #>AFFFC
D6B2 8504      2099 STA A00+4
D6B4 A9FF      2100 LDA #<AFFFC
D6B6 8505      2101 STA A00+5
                2102 ;
                2103 ; Cold boot system
                2104 ;
D6B8 4C0000    2105 JMP A00
                2106 ;.PAGE 'Absolute value of TOS'
                2107 ;
                2108 ; Branch if value on stack is negative
                2109 ;
D6BB 68        2110 ABI PLA
D6BC AA        2111 TAX
D6BD 68        2112 PLA
D6BE 3006      2113 BM INXFZ
                2114 ;
                2115 ; Push it back on stack
                2116 ;
D6C0 48        2117 PHA
D6C1 8A        2118 TXA
D6C2 48        2119 PHA
                2120 ;
                2121 ;
                2122 ;
D6C3 4C4DD2    2123 JMP UPDBY1
                2124 ;
                2125 ;
                2126 ;
D6C6 A8        2127 INXFZ TAY
D6C7 18        2128 CLC
D6C8 8A        2129 TXA

```



```

D6C9 49FF      2130      EOR      #$FF
D6CB 6901      2131      ADC      #1
D6CD AA        2132      TAX
D6CE 98        2133      TYA
D6CF 49FF      2134      EOR      #$FF
D6D1 6900      2135      ADC      #0
D6D3 48        2136      PHA
D6D4 8A        2137      TXA
D6D5 48        2138      PHA
                2139      ;
                2140      ;
                2141      ;
D6D6 4C4DD2    2142      JMP      UPDBY1
                2143      ;.PAGE 'Add TOS and TOS-1'
                2144      ;
                2145      ;
                2146      ;
                2147      i f n d e f  O R I G
                2148      A D I      T S X
                2149      C L C
                2150      L D A      P1BASE+3, X
                2151      A D C      P1BASE+1, X
                2152      S T A      P1BASE+3, X
                2153      L D A      P1BASE+4, X
                2154      A D C      P1BASE+2, X
                2155      S T A      P1BASE+4, X
                2156      I N X
                2157      I N X
                2158      T X S
                2159      ;
                2160      ;
                2161      ;
                2162      e l s e
D6D9 68        2163      A D I      P L A
D6DA 8574      2164      S T A      A74
D6DC 68        2165      P L A
D6DD 8575      2166      S T A      A74+1
                2167      ;
                2168      ;
                2169      ;
D6DF 68        2170      P L A
D6E0 A8        2171      T A Y
D6E1 68        2172      P L A
D6E2 AA        2173      T A X
                2174      ;
                2175      ;
                2176      ;
D6E3 98        2177      T Y A
D6E4 18        2178      C L C
D6E5 6574      2179      A D C      A74
D6E7 A8        2180      T A Y
D6E8 8A        2181      T X A
D6E9 6575      2182      A D C      A74+1
D6EB 48        2183      P H A
D6EC 98        2184      T Y A
D6ED 48        2185      P H A
                2186      e n d i f

```

```

2187 ;
2188 ;
2189 ;
D6EE 4C4DD2 2190     JMP     UPDBY1
2191 ;.PAGE 'Negate TOS'
2192 ;
2193 ;
2194 ;
D6F1 68      2195     NGI     PLA
D6F2 49FF    2196     EOR     #$FF
D6F4 18      2197     CLC
D6F5 6901    2198     ADC     #1
D6F7 AA      2199     TAX
D6F8 68      2200     PLA
D6F9 49FF    2201     EOR     #$FF
D6FB 6900    2202     ADC     #0
D6FD 48      2203     PHA
D6FE 8A      2204     TXA
D6FF 48      2205     PHA
2206 ;
2207 ;
2208 ;
D700 4C4DD2 2209     JMP     UPDBY1
2210 ;.PAGE 'Subtract TOS from TOS-1'
2211     ifndef ORIG
2212 ;
2213 ;   Set index to both operands
2214 ;
2215     SBI     TSX
2216 ;
2217 ;   Perform subtraction in place
2218 ;
2219     CLB
2220     LDA     P1BASE+3, X
2221     SBB     P1BASE+1, X
2222     STA     P1BASE+3, X
2223     LDA     P1BASE+4, X
2224     SBB     P1BASE+2, X
2225     STA     P1BASE+4, X
2226 ;
2227 ;   Set index to result
2228 ;
2229     INX
2230     INX
2231     TXS
2232     else
2233 ;
2234 ;   Pull subtrahend from stack
2235 ;
D703 68      2236     SBI     PLA
D704 8574    2237     STA     A74
D706 68      2238     PLA
D707 8575    2239     STA     A74+1
2240 ;
2241 ;   Pull minuend from stack
2242 ;
D709 68      2243     PLA

```

```

D70A A8      2244    TAY
D70B 68      2245    PLA
D70C AA      2246    TAX
                2247    ;
                2248    ; Perform subtraction and push result on stack
                2249    ;
D70D 98      2250    TYA
D70E 38      2251    CLB
D70F E574    2252    SBB    A74
D711 A8      2253    TAY
D712 8A      2254    TXA
D713 E575    2255    SBB    A74+1
D715 48      2256    PHA
D716 98      2257    TYA
D717 48      2258    PHA
                2259    endi f
                2260    ;
                2261    ;
                2262    ;
D718 4C4DD2  2263    JMP    UPDBY1
                2264    ;. PAGE ' Subroutines'
                2265    ;
                2266    ;
                2267    ;
D71B A900    2268    MULSUB  LDA    #0
D71D 8565    2269    STA    SUM+1
D71F 8564    2270    STA    SUM
                2271    ;
                2272    ;
                2273    ;
D721 A210    2274    LDX    #16
                2275    ;
                2276    ;
                2277    ;
D723 668B    2278    INXGC   ROR    A8A+1
D725 668A    2279    ROR    A8A
D727 900D    2280    BNC    INXGD
                2281    ;
                2282    ;
                2283    ;
D729 A564    2284    LDA    SUM
D72B 18      2285    CLC
D72C 6588    2286    ADC    A88
D72E 8564    2287    STA    SUM
D730 A565    2288    LDA    SUM+1
D732 6589    2289    ADC    A88+1
D734 8565    2290    STA    SUM+1
                2291    ;
                2292    ;
                2293    ;
D736 6665    2294    INXGD   ROR    SUM+1
D738 6664    2295    ROR    SUM
                2296    ;
                2297    ;
                2298    ;
D73A 668D    2299    ROR    A8C+1
D73C 668C    2300    ROR    A8C

```

```

                2301 ;
                2302 ;
                2303 ;
D73E CA        2304     DEX
D73F DOE2     2305     BNZ     I NXGC
                2306 ;
                2307 ;
                2308 ;
D741 60       2309     RTS
                2310
*****
                2311 ;
                2312 ;
                2313 ;
D742 68       2314     MULPLY  PLA
D743 858A     2315     STA     A8A
D745 68       2316     PLA
D746 858B     2317     STA     A8A+1
                2318 ;
                2319 ;
                2320 ;
D748 3009     2321     BM      I NXGF
                2322 ;
                2323 ;
                2324 ;
D74A 68       2325     PLA
D74B 8588     2326     STA     A88
D74D 68       2327     PLA
D74E 8589     2328     STA     A88+1
                2329 ;
                2330 ;
                2331 ;
D750 4C7DD7   2332     JMP     I NXGG
                2333 ;
                2334 ;
                2335 ;
D753 68       2336     I NXGF  PLA
D754 8588     2337     STA     A88
D756 68       2338     PLA
D757 8589     2339     STA     A88+1
                2340 ;
                2341 ;
                2342 ;
D759 1022     2343     BNM     I NXGG
                2344 ;
                2345 ;
                2346 ;
D75B A58A     2347     LDA     A8A
D75D 49FF     2348     EOR     #$FF
D75F 18       2349     CLC
D760 6901     2350     ADC     #1
D762 858A     2351     STA     A8A
D764 A58B     2352     LDA     A8A+1
D766 49FF     2353     EOR     #$FF
D768 6900     2354     ADC     #0
D76A 858B     2355     STA     A8A+1
                2356 ;

```

```

                2357 ;
                2358 ;
D76C A588      2359 LDA  A88
D76E 49FF      2360 EOR  #$FF
D770 18        2361 CLC
D771 6901      2362 ADC  #1
D773 8588      2363 STA  A88
D775 A589      2364 LDA  A88+1
D777 49FF      2365 EOR  #$FF
D779 6900      2366 ADC  #0
D77B 8589      2367 STA  A88+1
                2368 ;
                2369 ;
                2370 ;
D77D 201BD7    2371 INXGG JSR  MULSUB
                2372 ;
                2373 ;
                2374 ;
D780 A58D      2375 LDA  A8C+1
D782 48        2376 PHA
D783 A58C      2377 LDA  A8C
D785 48        2378 PHA
                2379 ;
                2380 ;
                2381 ;
D786 4C4DD2    2382 JMP  UPDBY1
                2383 ;.PAGE 'Square TOS'
                2384 ;
                2385 ;
                2386 ;
D789 BA        2387 SQI   TSX
D78A BD0201    2388 LDA  P1BASE+2, X
D78D 48        2389 PHA
D78E BD0101    2390 LDA  P1BASE+1, X
D791 48        2391 PHA
                2392 ;
                2393 ;
                2394 ;
                2395 * JSR  MULPLY
                2396 * RTS
D792 4C42D7    2397 JMP  MULPLY
                2398 ;.PAGE 'Common routine for integer divide and modulo'
                2399 ;
                2400 ;
                2401 ;
                2402 DI VI DE
D795 A000      2403 LDY  #0
D797 848A      2404 STY  A8A
                2405 ;
                2406 ;
                2407 ;
D799 8464      2408 STY  SUM
D79B 8465      2409 STY  SUM+1
                2410 ;
                2411 ;
                2412 ;
D79D 848C      2413 STY  A8C

```

```

D79F 848D      2414      STY   A8C+1
                2415      ;
                2416      ; Branch if divisor is not zero
                2417      ;
D7A1 A586      2418      LDA   A86
D7A3 D007      2419      BNZ   INXGL
D7A5 A587      2420      LDA   A86+1
D7A7 D003      2421      BNZ   INXGL
                2422      ;
                2423      ;
                2424      ;
D7A9 4CDFD1    2425      JMP   ERR6
                2426      ;
                2427      ; Branch if divisor is not negative
                2428      ;
D7AC A587      2429      INXGL  LDA   A86+1
D7AE 1013      2430      BNM   INXGN
                2431      ;
                2432      ; Complement divisor
                2433      ;
D7B0 A586      2434      LDA   A86
D7B2 49FF      2435      EOR   #$FF
D7B4 18        2436      CLC
D7B5 6901      2437      ADC   #1
D7B7 8586      2438      STA   A86
D7B9 A587      2439      LDA   A86+1
D7BB 49FF      2440      EOR   #$FF
D7BD 6900      2441      ADC   #0
D7BF 8587      2442      STA   A86+1
                2443      ;
                2444      ;
                2445      ;
D7C1 E68A      2446      INC   A8A
                2447      ;
                2448      ; Branch if dividend is not negative
                2449      ;
D7C3 A589      2450      INXGN  LDA   A88+1
D7C5 1013      2451      BNM   INXGO
                2452      ;
                2453      ; Complement dividend
                2454      ;
D7C7 A588      2455      LDA   A88
D7C9 49FF      2456      EOR   #$FF
D7CB 18        2457      CLC
D7CC 6901      2458      ADC   #1
D7CE 8588      2459      STA   A88
D7D0 A589      2460      LDA   A88+1
D7D2 49FF      2461      EOR   #$FF
D7D4 6900      2462      ADC   #0
D7D6 8589      2463      STA   A88+1
                2464      ;
                2465      ;
                2466      ;
D7D8 C68A      2467      DEC   A8A
                2468      ;
                2469      ;
                2470      ;

```

D7DA	38	2471	INXGO	CLB
D7DB	A588	2472	LDA	A88
D7DD	E586	2473	SBB	A86
D7DF	A589	2474	LDA	A88+1
D7E1	E587	2475	SBB	A86+1
D7E3	B00D	2476	BNB	INXGP
		2477	;	
		2478	;	
		2479	;	
D7E5	848C	2480	STY	A8C
D7E7	848D	2481	STY	A8C+1
		2482	;	
		2483	;	
		2484	;	
D7E9	A588	2485	LDA	A88
D7EB	8588	2486	STA	A88
D7ED	A589	2487	LDA	A88+1
D7EF	8589	2488	STA	A88+1
		2489	;	
		2490	;	
		2491	;	
D7F1	60	2492	RTS	
		2493	;	
		2494	;	
		2495	;	
D7F2	18	2496	INXGP	CLC
D7F3	A210	2497	LDX	#16
D7F5	A589	2498	LDA	A88+1
D7F7	D00A	2499	BNZ	INXGQ
		2500	;	
		2501	;	
		2502	;	
D7F9	A586	2503	LDA	A86
D7FB	8565	2504	STA	SUM+1
D7FD	A587	2505	LDA	A86+1
D7FF	8586	2506	STA	A86
		2507	;	
		2508	;	
		2509	;	
D801	A208	2510	LDX	#8
D803	18	2511	INXGQ	CLC
D804	6687	2512	ROR	A86+1
D806	D022	2513	BNZ	INXGT
D808	6686	2514	ROR	A86
D80A	D020	2515	BNZ	INXGU
D80C	6665	2516	INXGS	ROR SUM+1
D80E	6664	2517	ROR	SUM
D810	A588	2518	LDA	A88
D812	38	2519	CLB	
D813	E564	2520	SBB	SUM
D815	A8	2521	TAY	
D816	A589	2522	LDA	A88+1
D818	E565	2523	SBB	SUM+1
D81A	9015	2524	BB	INXGV
D81C	8589	2525	STA	A88+1
D81E	8488	2526	STY	A88
D820	38	2527	SEC	

D821	268C	2528	ROL	A8C
D823	268D	2529	ROL	A8C+1
D825	CA	2530	DEX	
D826	18	2531	CLC	
D827	DOE3	2532	BNZ	INXGS
		2533	;	
		2534	;	
		2535	;	
D829	60	2536	RTS	
		2537	;	
		2538	;	
		2539	;	
D82A	6686	2540	INXGT	ROR A86
D82C	6665	2541	INXGU	ROR SUM+1
D82E	6664	2542	ROR	SUM
D830	18	2543	CLC	
D831	268C	2544	INXGV	ROL A8C
D833	268D	2545	ROL	A8C+1
D835	CA	2546	DEX	
D836	DOCB	2547	BNZ	INXGQ
		2548	;	
		2549	;	
		2550	;	
D838	60	2551	RTS	
		2552	;	. PAGE 'Di vi de TOS- 1 by TOS'
		2553	;	
		2554	;	
		2555	;	
D839	68	2556	DVI	PLA
D83A	8586	2557	STA	A86
D83C	68	2558	PLA	
D83D	8587	2559	STA	A86+1
		2560	;	
		2561	;	
		2562	;	
D83F	68	2563	PLA	
D840	8588	2564	STA	A88
D842	68	2565	PLA	
D843	8589	2566	STA	A88+1
		2567	;	
		2568	;	
		2569	;	
D845	2095D7	2570	JSR	DI VI DE
		2571	;	
		2572	;	
		2573	;	
D848	A58A	2574	LDA	A8A
D84A	F011	2575	BZ	INXGW
		2576	;	
		2577	;	Complement quotient
		2578	;	
D84C	A58C	2579	LDA	A8C
D84E	49FF	2580	EOR	#\$FF
D850	18	2581	CLC	
D851	6901	2582	ADC	#1
D853	858C	2583	STA	A8C
D855	A58D	2584	LDA	A8C+1



```

D857 49FF      2585      EOR   #SFF
D859 6900      2586      ADC   #0
D85B 858D      2587      STA   A8C+1
                2588      ;
                2589      ;
                2590      ;
D85D A58D      2591      INXGW LDA   A8C+1
D85F 48        2592      PHA
D860 A58C      2593      LDA   A8C
D862 48        2594      PHA
                2595      ;
                2596      ;
                2597      ;
D863 4C4DD2    2598      JMP   UPDBY1
                2599      ;. PAGE 'Modul o'
                2600      ;
                2601      ;
                2602      ;
D866 68        2603      MODI   PLA
D867 8586      2604      STA   A86
D869 68        2605      PLA
D86A 8587      2606      STA   A86+1
                2607      ;
                2608      ;
                2609      ;
D86C 68        2610      PLA
D86D 8588      2611      STA   A88
D86F 68        2612      PLA
D870 8589      2613      STA   A88+1
                2614      ;
                2615      ;
                2616      ;
D872 2095D7    2617      JSR   DI VI DE
                2618      ;
                2619      ;
                2620      ;
D875 A589      2621      LDA   A88+1
D877 48        2622      PHA
D878 A588      2623      LDA   A88
D87A 48        2624      PHA
                2625      ;
                2626      ;
                2627      ;
D87B 4C4DD2    2628      JMP   UPDBY1
                2629      ;. PAGE 'Check agai nst subrange bounds'
                2630      ;
                2631      ;
                2632      ;
D87E 68        2633      CHK   PLA
D87F 8574      2634      STA   A74
D881 68        2635      PLA
D882 8575      2636      STA   A74+1
                2637      ;
                2638      ;
                2639      ;
D884 68        2640      PLA
D885 8576      2641      STA   A76

```

```

D887 68      2642    PLA
D888 8577    2643    STA    A76+1
                2644    ;
                2645    ;
                2646    ;
D88A BA     2647    TSX
D88B BD0101 2648    LDA    P1BASE+1, X
D88E 8578    2649    STA    A78
D890 BD0201 2650    LDA    P1BASE+2, X
D893 8579    2651    STA    A78+1
                2652    ;
                2653    ;
                2654    ;
D895 4577    2655    EOR    A76+1
D897 3010    2656    BM     I NXGX
D899 A577    2657    LDA    A76+1
D89B C579    2658    CMP    A78+1
D89D 900E    2659    BL     I NXGY
D89F D029    2660    BNE    CHKNG
D8A1 A578    2661    LDA    A78
D8A3 C576    2662    CMP    A76
D8A5 B006    2663    BGE    I NXGY
                2664    ;
                2665    ;    Uncondi ti onal  branch
                2666    ;
D8A7 9021    2667    BL     CHKNG
                2668    ;
                2669    ;
                2670    ;
D8A9 A579    2671    I NXGX    LDA    A78+1
D8AB 301D    2672    BM     CHKNG
                2673    ;
                2674    ;
                2675    ;
D8AD A575    2676    I NXGY    LDA    A74+1
D8AF 4579    2677    EOR    A78+1
D8B1 3010    2678    BM     I NXHA
D8B3 A579    2679    LDA    A78+1
D8B5 C575    2680    CMP    A74+1
D8B7 900E    2681    BL     CHKOK
D8B9 D00F    2682    BNE    CHKNG
D8BB A574    2683    LDA    A74
D8BD C578    2684    CMP    A78
D8BF B006    2685    BGE    CHKOK
                2686    ;
                2687    ;    Uncondi ti onal  branch
                2688    ;
D8C1 9007    2689    BL     CHKNG
                2690    ;
                2691    ;
                2692    ;
D8C3 A579    2693    I NXHA    LDA    A78+1
D8C5 1003    2694    BNM    CHKNG
                2695    ;
                2696    ;
                2697    ;
D8C7 4C4DD2 2698    CHKOK    JMP    UPDBY1

```

```

2699 ;
2700 ;
2701 ;
D8CA 4CB7D1 2702 CHKNG    JMP    ERR1
2703 ;. PAGE
2704   ifdef BIG
2705 ;
2706 ;
2707 ;
2708 ND9A7    LDA    BZ5E
2709     STA    A74
2710     LDA    BZ5E+1
2711     STA    A74+1
2712 ;
2713 ;
2714 ;
2715 NA LDA    A74+1
2716     BNZ    NB
2717     LDA    A74
2718     BNZ    NB
2719 ;
2720 ;
2721 ;
2722     RTS
2723 ;
2724 ;
2725 ;
2726 NB LDY    #0
2727     LDA    (A74), Y
2728     CMP    I PCPTR
2729     BNE    NC
2730     INY
2731     LDA    (A74), Y
2732     CMP    I PCPTR+1
2733     BNE    NC
2734 ;
2735 ;
2736 ;
2737     RTS
2738 ;
2739 ;
2740 ;
2741 NC LDY    #2
2742     LDA    (A74), Y
2743     TAX
2744     INY
2745     LDA    (A74), Y
2746     STA    A74+1
2747     STX    A74
2748 ;
2749 ;
2750 ;
2751     JMP    NA
2752   endi f
2753 ;. PAGE
2754 ;
2755 ;   Load a packed array

```

```

2756 ;
2757 LPA
2758   ifdef BIG
2759     JSR   ND9A7
2760 ;
2761 ;
2762 ;
2763     LDA   IPCPTR
2764     STA   A76
2765     LDA   IPCPTR+1
2766     STA   A76+1
2767   endif
2768 ;
2769 ;
2770 ;
D8CD E658 2771     INC   IPCPTR
D8CF D002 2772     BNZ   INXHJ
D8D1 E659 2773     INC   IPCPTR+1
2774 INXHJ
2775 ;
2776 ;
2777 ;
2778   ifdef BIG
2779     STA   RAMRDS
2780     LDY   #0
2781   endif
D8D3 B158 2782     LDA   (IPCPTR), Y
2783   ifndef BIG
D8D5 AA   2784     TAX
2785   else
2786     STA   RAMRDC
2787     STA   A78
2788   endif
2789 ;
2790 ;   Increment pointer
2791 ;
D8D6 E658 2792     INC   IPCPTR
D8D8 D002 2793     BNZ   INXHK
D8DA E659 2794     INC   IPCPTR+1
2795 INXHK
2796   ifndef BIG
2797 ;
2798 ;
2799 ;
D8DC A559 2800     LDA   IPCPTR+1
D8DE 48   2801     PHA
D8DF A558 2802     LDA   IPCPTR
D8E1 48   2803     PHA
2804   endif
2805 ;
2806 ;
2807 ;
D8E2 4CFAD8 2808     JMP   INXHM
2809 ;.PAGE 'Load constant string address'
2810 ;
2811 ;
2812 ;

```

```

2813 LSA
2814   ifdef BIG
2815     JSR   ND9A7
2816 ;
2817 ;
2818 ;
2819     LDA   I PCPTR
2820     STA   A76
2821     LDA   I PCPTR+1
2822     STA   A76+1
2823   endif
2824 ;
2825 ;   Increment pointer
2826 ;
D8E5 E658 2827     INC   I PCPTR
D8E7 D002 2828     BNZ   I NXHL
D8E9 E659 2829     INC   I PCPTR+1
2830 I NXHL
2831   ifndef BIG
2832 ;
2833 ;
2834 ;
D8EB A559 2835     LDA   I PCPTR+1
D8ED 48   2836     PHA
D8EE A558 2837     LDA   I PCPTR
D8FO 48   2838     PHA
2839 ;
2840 ;
2841 ;
D8F1 B158 2842     LDA   (I PCPTR), Y
D8F3 AA   2843     TAX
2844 ;
2845 ;   Increment pointer
2846 ;
D8F4 E658 2847     INC   I PCPTR
D8F6 D002 2848     BNZ   I NXHM
D8F8 E659 2849     INC   I PCPTR+1
2850 ;
2851 ;
2852 ;
D8FA 18   2853 I NXHM   CLC
D8FB 8A   2854     TXA
D8FC 6558 2855     ADC   I PCPTR
2856   else
2857     STA   RAMRDS
2858 ;
2859 ;
2860 ;
2861     LDY   #0
2862     LDA   (I PCPTR), Y
2863     STA   RAMRDC
2864     STA   A78
2865     INC   A78
2866 ;
2867 ;
2868 ;
2869 I NXHM   LDA   A74+1

```

```

2870    BNZ    XHMAI
2871    LDA    A74
2872    BNZ    XHMAI
2873    ;
2874    ;
2875    ;
2876    LDA    PSKPTR
2877    CLB
2878    SBB    A78
2879    STA    PSKPTR
2880    BNB    XHMAB
2881    DEC    PSKPTR+1
2882    XHMAB  LDA    PSKPTR
2883    AND    #$FE
2884    STA    PSKPTR
2885    ;
2886    ;
2887    ;
2888    LDA    PSKPTR+1
2889    PHA
2890    LDA    PSKPTR
2891    PHA
2892    ;
2893    ;
2894    ;
2895    LDY    A78
2896    DEY
2897    BM     XHMAD
2898    STA    RAMRDS
2899    XHMAC  LDA    (PCPTR), Y
2900    STA    (PSKPTR), Y
2901    DEY
2902    BNM    XHMAC
2903    STA    RAMRDC
2904    ;
2905    ;   Subtract 4 from pointer
2906    ;
2907    XHMAD  LDA    PSKPTR
2908    CLB
2909    SBB    #4
2910    STA    PSKPTR
2911    BNB    XHMAE
2912    DEC    PSKPTR+1
2913    ;
2914    ;
2915    ;
2916    XHMAE  LDY    #0
2917    LDA    A76
2918    STA    (PSKPTR), Y
2919    INY
2920    LDA    A76+1
2921    STA    (PSKPTR), Y
2922    INY
2923    LDA    BZ5E
2924    STA    (PSKPTR), Y
2925    INY
2926    LDA    BZ5E+1

```

```

2927     STA    (PSKPTR), Y
2928     ;
2929     ;
2930     ;
2931     LDA    PSKPTR
2932     STA    BZ5E
2933     LDA    PSKPTR+1
2934     STA    BZ5E+1
2935     ;
2936     ;
2937     ;
2938     CLB
2939     LDA    PSKPTR
2940     SBB    HPPTR
2941     LDA    PSKPTR+1
2942     SBB    HPPTR+1
2943     BNB    XHMAF
2944     ;
2945     ;
2946     ;
2947     JMP    OOMERR
2948     ;
2949     ;
2950     ;
2951 XHMAF    LDA    I PCPTR
2952         CLC
2953         ADC    A78
2954     endi f
D8FE 8558 2955     STA    I PCPTR
D900 9002 2956     BNC    I NXHO
D902 E659 2957     INC    I PCPTR+1
2958     ;
2959     ;
2960     ;
D904 4C53D2 2961 I NXHO    JMP    EXECPC
2962     ifdef BIG
2963     ;
2964     ;
2965     ;
2966 XHMAI    LDA    A74
2967         CLC
2968         ADC    #4
2969         STA    A74
2970         BNC    XHMAK
2971         INC    A74+1
2972     ;
2973     ;
2974     ;
2975 XHMAK    LDA    A74+1
2976         PHA
2977         LDA    A74
2978         PHA
2979     ;
2980     ;
2981     ;
2982         JMP    XHMAF
2983     endi f

```

```

2984 ;. PAGE 'String assign'
2985 ;
2986 ;
2987 ;
D907 68      2988 SAS      PLA
D908 8568    2989      STA      SRCPTR
D90A 68      2990      PLA
D90B 8569    2991      STA      SRCPTR+1
2992 ;
2993 ;
2994 ;
D90D 68      2995      PLA
D90E 856A    2996      STA      DSTPTR
D910 68      2997      PLA
D911 856B    2998      STA      DSTPTR+1
2999 ;
3000 ;
3001 ;
3002      RDAUX
      +      I FDEF  BIG
      +      STA      RAMRDS
      +      ENDI F
D913 A001    3003      LDY      #1
D915 B158    3004      LDA      (I PCPTR), Y
3005      RDMAIN
      +      I FDEF  BIG
      +      STA      RAMRDC
      +      ENDI F
D917 A669    3006      LDX      SRCPTR+1
D919 D012    3007      BNZ      I NXHP
D91B C901    3008      CMP      #1
D91D 9026    3009      BL       JERR13
D91F A901    3010      LDA      #1
D921 A000    3011      LDY      #0
D923 916A    3012      STA      (DSTPTR), Y
D925 A568    3013      LDA      SRCPTR
D927 C8      3014      I NY
D928 916A    3015      STA      (DSTPTR), Y
3016 ;
3017 ;
3018 ;
D92A 4C3BD2  3019      JMP      UPDBY2
3020 ;
3021 ;
3022 ;
D92D A000    3023 I NXHP      LDY      #0
D92F D168    3024      CMP      (SRCPTR), Y
D931 9012    3025      BL       JERR13
D933 B168    3026      LDA      (SRCPTR), Y
D935 916A    3027      STA      (DSTPTR), Y
D937 A8      3028      TAY
D938 4C40D9  3029      JMP      I NXHR
3030 ;
3031 ;
3032 ;
D93B B168    3033 I NXHQ      LDA      (SRCPTR), Y
D93D 916A    3034      STA      (DSTPTR), Y

```



```

D93F 88      3035      DEY
D940 DOF9    3036      INXHR      BNZ      INXHQ
              3037      ;
              3038      ;
              3039      ;
D942 4C3BD2 3040      JMP      UPDBY2
              3041      ;
              3042      ;
              3043      ;
D945 4CF7D1 3044      JERR13     JMP      ERR13
              3045      ;.PAGE 'Index string array'
              3046      ;
              3047      ;
              3048      ;
D948 BA      3049      IXS        TSX
D949 BDO201 3050      LDA      P1BASE+2, X
D94C DO17    3051      BNZ      INXHU
D94E BDO101 3052      LDA      P1BASE+1, X
D951 FO12    3053      BZ       INXHU
              3054      ;
              3055      ;
              3056      ;
D953 BC0301 3057      LDY      P1BASE+3, X
D956 8474    3058      STY      A74
D958 BC0401 3059      LDY      P1BASE+4, X
D95B 8475    3060      STY      A74+1
              3061      ;
              3062      ;
              3063      ;
D95D A000    3064      LDY      #0
D95F D174    3065      CMP      (A74), Y
D961 9005    3066      BL      INXHW
D963 FO03    3067      BE      INXHW
              3068      ;
              3069      ;
              3070      ;
D965 4CB7D1 3071      INXHU     JMP      ERR1
              3072      ;
              3073      ;
              3074      ;
D968 4C4DD2 3075      INXHW     JMP      UPDBY1
              3076      ;.PAGE 'Static index and load word'
              3077      ;
              3078      ;
              3079      ;
D96B A001    3080      IND      LDY      #1
D96D 2055D1 3081      JSR      EBI G2M
              3082      ;
              3083      ;
              3084      ;
D970 18      3085      CLC
D971 68      3086      PLA
D972 655E    3087      ADC      OPRND
D974 8578    3088      STA      A78
D976 68      3089      PLA
D977 655F    3090      ADC      OPRND+1
D979 8579    3091      STA      A78+1

```

```

3092 ;
3093 ;
3094 ;
D97B A001 3095 LDY #1
D97D B178 3096 LDA (A78), Y
D97F 48 3097 PHA
D980 88 3098 DEY
D981 B178 3099 LDA (A78), Y
D983 48 3100 PHA
3101 ;
3102 ;
3103 ;
D984 4C3BD2 3104 JMP UPDBY2
3105 ;.PAGE 'Increment field pointer'
3106 ;
3107 ;
3108 ;
D987 A001 3109 INC LDY #1
D989 2055D1 3110 JSR EBIG2M
3111 ;
3112 ;
3113 ;
D98C 18 3114 CLC
D98D 68 3115 PLA
D98E 655E 3116 ADC OPRND
D990 AA 3117 TAX
D991 68 3118 PLA
D992 655F 3119 ADC OPRND+1
D994 48 3120 PHA
D995 8A 3121 TXA
D996 48 3122 PHA
3123 ;
3124 ;
3125 ;
D997 4C3BD2 3126 JMP UPDBY2
3127 ;.PAGE 'Index array'
3128 ;
3129 ;
3130 ;
D99A 68 3131 IXA PLA
D99B 8588 3132 STA A88
D99D 68 3133 PLA
D99E 8589 3134 STA A88+1
3135 ;
3136 ;
3137 ;
D9A0 A001 3138 LDY #1
D9A2 2055D1 3139 JSR EBIG2M
3140 ;
3141 ;
3142 ;
D9A5 A55F 3143 LDA OPRND+1
D9A7 D00D 3144 BNZ INXHY
3145 ;
3146 ;
3147 ;
D9A9 A55E 3148 LDA OPRND

```

```

D9AB C902      3149      CMP   #S02
D9AD D007      3150      BNE   INXHY
                3151      ;
                3152      ;
                3153      ;
D9AF 0688      3154      ASL   A88
D9B1 2689      3155      ROL   A88+1
                3156      ;
                3157      ;
                3158      ;
D9B3 4CC9D9    3159      JMP   INXIA
                3160      ;
                3161      ;
                3162      ;
D9B6 A55E      3163      INXHY LDA   OPRND
D9B8 858A      3164      STA   A8A
D9BA A55F      3165      LDA   OPRND+1
D9BC 858B      3166      STA   A8A+1
D9BE 201BD7    3167      JSR   MULSUB
                3168      ;
                3169      ;
                3170      ;
D9C1 A58C      3171      LDA   A8C
D9C3 8588      3172      STA   A88
D9C5 A58D      3173      LDA   A8C+1
D9C7 8589      3174      STA   A88+1
                3175      ;
                3176      ;
                3177      ;
D9C9 18        3178      INXIA CLC
D9CA 68        3179      PLA
D9CB 6588      3180      ADC   A88
D9CD 8578      3181      STA   A78
D9CF 68        3182      PLA
D9D0 6589      3183      ADC   A88+1
D9D2 48        3184      PHA
D9D3 A578      3185      LDA   A78
D9D5 48        3186      PHA
                3187      ;
                3188      ;
                3189      ;
D9D6 4C3BD2    3190      JMP   UPDBY2
                3191      ;. PAGE 'Index packed array'
                3192      ;
                3193      ;
                3194      ;
                3195      IXP      RDAUX
                +      I FDEF  BI G
                +      STA   RAMRDS
                +      ENDI F
D9D9 A001      3196      LDY   #1
D9DB B158      3197      LDA   (I PCPTR), Y
                3198      RDMAI N
                +      I FDEF  BI G
                +      STA   RAMRDC
                +      ENDI F
D9DD 8586      3199      STA   A86

```

```

3200   RDAUX
      +   I FDEF BIG
      +   STA   RAMRDS
      +   ENDI F
D9DF A002 3201   LDY   #2
D9E1 B158 3202   LDA   (I PCPTR), Y
3203   RDMAI N
      +   I FDEF BIG
      +   STA   RAMRDC
      +   ENDI F
D9E3 857A 3204   STA   A7A
D9E5 A900 3205   LDA   #0
D9E7 8587 3206   STA   A86+1
D9E9 857B 3207   STA   A7A+1
D9EB 68   3208   PLA
D9EC 8588 3209   STA   A88
D9EE 68   3210   PLA
D9EF 8589 3211   STA   A88+1
D9F1 2095D7 3212   JSR   DI VI DE
D9F4 068C 3213   ASL   A8C
D9F6 268D 3214   ROL   A8C+1
D9F8 18   3215   CLC
D9F9 68   3216   PLA
D9FA 658C 3217   ADC   A8C
D9FC 8578 3218   STA   A78
D9FE 68   3219   PLA
D9FF 658D 3220   ADC   A8C+1
DA01 48   3221   PHA
DA02 A578 3222   LDA   A78
DA04 48   3223   PHA
DA05 A57B 3224   LDA   A7A+1
DA07 48   3225   PHA
DA08 A57A 3226   LDA   A7A
DAOA 48   3227   PHA
DAOB A900 3228   LDA   #0
DAOD A688 3229   LDX   A88
DAOF F006 3230   BZ   I NXI C
3231   ;
3232   ;
3233   ;
DA11 18   3234   CLC
DA12 657A 3235   I NXI B   ADC   A7A
DA14 CA   3236   DEX
DA15 DOFB 3237   BNZ   I NXI B
3238   ;
3239   ;
3240   ;
DA17 48   3241   I NXI C   PHA
DA18 48   3242   PHA
3243   ;
3244   ;
3245   ;
DA19 4C2ED2 3246   JMP   UPDBY3
3247   ;.PAGE 'Load a packed field'
3248   ;
3249   ;   Get offset into word in number of bits
3250   ;

```

```

DA1C 68      3251 LDP      PLA
DA1D 857C    3252      STA  A7C
DA1F 68      3253      PLA
DA20 857D    3254      STA  A7C+1
              3255 ;
              3256 ; Get number of bits in field
              3257 ;
DA22 68      3258      PLA
DA23 857A    3259      STA  A7A
DA25 68      3260      PLA
DA26 857B    3261      STA  A7A+1
              3262 ;
              3263 ; Get address of word to extract from
              3264 ;
DA28 68      3265      PLA
DA29 8578    3266      STA  A78
DA2B 68      3267      PLA
DA2C 8579    3268      STA  A78+1
              3269 ;
              3270 ; Get source word
              3271 ;
DA2E A000    3272      LDY  #0
DA30 B178    3273      LDA  (A78), Y
DA32 857E    3274      STA  A7E
DA34 C8      3275      INY
DA35 B178    3276      LDA  (A78), Y
DA37 857F    3277      STA  A7E+1
              3278 ;
              3279 ; Branch if field starts in high order byte
              3280 ;
DA39 A57C    3281      LDA  A7C
DA3B 38      3282      CLB
DA3C E908    3283      SBB  #8
DA3E 300F    3284      BM   INXID
              3285 ;
              3286 ;
              3287 ;
DA40 AA      3288      TAX
              3289 ;
              3290 ;
              3291 ;
DA41 A57E    3292      LDA  A7E
DA43 A47F    3293      LDY  A7E+1
DA45 847E    3294      STY  A7E
DA47 857F    3295      STA  A7E+1
              3296 ;
              3297 ;
              3298 ;
DA49 E000    3299      CPX  #0
DA4B D007    3300      BNE  INXIE
              3301 ;
              3302 ; Unconditional branch
              3303 ;
DA4D F00C    3304      BE   INXIG
              3305 ;
              3306 ;
              3307 ;

```

```

DA4F 6908      3308  INXID   ADC   #8
DA51 AA        3309      TAX
DA52 F007      3310      BZ    INXI G
                3311      ;
                3312      ; Right adjust source word
                3313      ;
DA54 667F      3314  INXI E   ROR   A7E+1
DA56 667E      3315      ROR   A7E
DA58 CA        3316      DEX
DA59 DOF9      3317      BNZ   INXI E
                3318      ;
                3319      ; Multiply adjusted offset by two
                3320      ;
DA5B A57A      3321  INXI G   LDA   A7A
DA5D 0A        3322      ASL   A
DA5E AA        3323      TAX
                3324      ;
                3325      ; Push extracted field onto stack
                3326      ;
DA5F A57E      3327      LDA   A7E
DA61 3D92DD    3328      AND   INXKF, X
DA64 857E      3329      STA   A7E
DA66 A57F      3330      LDA   A7E+1
DA68 3D93DD    3331      AND   INXKF+1, X
DA6B 48        3332      PHA
DA6C A57E      3333      LDA   A7E
DA6E 48        3334      PHA
                3335      ;
                3336      ;
                3337      ;
DA6F 4C4DD2    3338      JMP   UPDBY1
                3339      ;.PAGE 'Store into a packed field'
                3340      ;
                3341      ; Get TOS
                3342      ;
DA72 68        3343  STP     PLA
DA73 8574      3344      STA   A74
DA75 68        3345      PLA
DA76 8575      3346      STA   A74+1
                3347      ;
                3348      ; Get TOS - 1
                3349      ;
DA78 68        3350      PLA
DA79 857C      3351      STA   A7C
DA7B 68        3352      PLA
DA7C 857D      3353      STA   A7C+1
                3354      ;
                3355      ;
                3356      ;
DA7E 68        3357      PLA
DA7F 857A      3358      STA   A7A
DA81 68        3359      PLA
DA82 857B      3360      STA   A7A+1
                3361      ;
                3362      ;
                3363      ;
DA84 A57A      3364      LDA   A7A

```

```

DA86 OA      3365   ASL   A
DA87 AA      3366   TAX
DA88 BD92DD  3367   LDA   INXKF, X
DA8B 8576    3368   STA   A76
DA8D 2574    3369   AND   A74
DA8F 8574    3370   STA   A74
DA91 BD93DD  3371   LDA   INXKF+1, X
DA94 8577    3372   STA   A76+1
DA96 2575    3373   AND   A74+1
DA98 8575    3374   STA   A74+1
          3375   ;
          3376   ;
          3377   ;
DA9A A57C    3378   LDA   A7C
DA9C 38      3379   CLB
DA9D E908    3380   SBB   #8
DA9F 3017    3381   BM    INXII
          3382   ;
          3383   ;
          3384   ;
DAA1 AA      3385   TAX
          3386   ;
          3387   ;
          3388   ;
DAA2 A574    3389   LDA   A74
DAA4 A475    3390   LDY   A74+1
DAA6 8575    3391   STA   A74+1
DAA8 8474    3392   STY   A74
          3393   ;
          3394   ;
          3395   ;
DAAA A576    3396   LDA   A76
DAAC A477    3397   LDY   A76+1
DAAE 8577    3398   STA   A76+1
DAB0 8476    3399   STY   A76
          3400   ;
          3401   ;
          3402   ;
DAB2 E000    3403   CPX   #0
DAB4 D007    3404   BNE   INXIJ
          3405   ;
          3406   ; Unconditional branch
          3407   ;
DAB6 F010    3408   BE    INXIL
          3409   ;
          3410   ;
          3411   ;
DAB8 6908    3412   INXII   ADC   #8
DABA AA      3413   TAX
DABB F00B    3414   BZ    INXIL
          3415   ;
          3416   ;
          3417   ;
DABD 0674    3418   INXIJ   ASL   A74
DABF 2675    3419   ROL   A74+1
DAC1 0676    3420   ASL   A76
DAC3 2677    3421   ROL   A76+1

```

```

DAC5 CA          3422      DEX
DAC6 DOF5        3423      BNZ   INXI J
                 3424      ;
                 3425      ;
                 3426      ;
DAC8 68          3427      INXIL  PLA
DAC9 8578        3428      STA   A78
DACB 68          3429      PLA
DACC 8579        3430      STA   A78+1
                 3431      ;
                 3432      ;
                 3433      ;
DACE A000        3434      LDY   #0
DADO B178        3435      LDA   (A78), Y
DAD2 857E        3436      STA   A7E
DAD4 C8          3437      INY
DAD5 B178        3438      LDA   (A78), Y
DAD7 857F        3439      STA   A7E+1
DAD9 A576        3440      LDA   A76
DADB 49FF        3441      EOR   #$FF
DADD 257E        3442      AND   A7E
DADF 0574        3443      ORA   A74
DAE1 857E        3444      STA   A7E
DAE3 A577        3445      LDA   A76+1
DAE5 49FF        3446      EOR   #$FF
DAE7 257F        3447      AND   A7E+1
DAE9 0575        3448      ORA   A74+1
DAEB 857F        3449      STA   A7E+1
                 3450      ;
                 3451      ;
                 3452      ;
DAED A000        3453      LDY   #0
DAEF A57E        3454      LDA   A7E
DAF1 9178        3455      STA   (A78), Y
DAF3 C8          3456      INY
DAF4 A57F        3457      LDA   A7E+1
DAF6 9178        3458      STA   (A78), Y
                 3459      ;
                 3460      ;
                 3461      ;
DAF8 4C4DD2      3462      JMP   UPDBY1
                 3463      ;.PAGE 'Set logic - Subroutine for INT, DIF and UNI (AND, AND
NOT and OR)'
                 3464      ;
                 3465      ; Pull return address off stack and save it
                 3466      ;
DAFB 68          3467      FIXSET PLA
DAFC 858C        3468      STA   A8C
DAFE 68          3469      PLA
DAFF 858D        3470      STA   A8C+1
                 3471      ;
                 3472      ; Double low-order byte of TOS
                 3473      ;
DB01 68          3474      PLA
DB02 0A          3475      ASL   A
DB03 857C        3476      STA   A7C
                 3477      ;

```



```

3478 ; Discard high order byte of TOS
3479 ;
DB05 68 3480 PLA
3481 ;
3482 ;
3483 ;
DB06 BA 3484 TSX
DB07 8A 3485 TXA
DB08 18 3486 CLC
DB09 657C 3487 ADC A7C
DB0B A8 3488 TAY
DB0C C8 3489 INY
DB0D 847E 3490 STY A7E
3491 ;
3492 ;
3493 ;
DB0F B90001 3494 LDA P1BASE, Y
DB12 0A 3495 ASL A
3496 ;
3497 ;
3498 ;
DB13 C8 3499 INY
DB14 C8 3500 INY
DB15 8474 3501 STY A74
3502 ;
3503 ; Return to caller
3504 ;
DB17 E68C 3505 INC A8C
DB19 D002 3506 BNZ INXI Q
DB1B E68D 3507 INC A8C+1
3508 ;
3509 ;
3510 ;
DB1D 6C8C00 3511 INXI Q JMP (A8C)
3512 ;.PAGE 'Set logic - Intersection (AND)'
3513 ;
3514 ;
3515 ;
DB20 20FBDA 3516 INT JSR FIXSET
3517 ;
3518 ;
3519 ;
DB23 A67C 3520 LDX A7C
DB25 38 3521 SEC
DB26 E57C 3522 SBC A7C
DB28 1006 3523 BNM INXI R
3524 ;
3525 ;
3526 ;
DB2A 18 3527 CLC
DB2B 657C 3528 ADC A7C
DB2D AA 3529 TAX
DB2E A900 3530 LDA #0
DB30 8590 3531 INXI R STA A90
DB32 E000 3532 CPX #0
DB34 FO0B 3533 BE INXI T
3534 ;

```

```

3535 ;
3536 ;
DB36 68      3537 INXIS   PLA
DB37 390001  3538     AND   P1BASE, Y
DB3A 990001  3539     STA   P1BASE, Y
DB3D C8      3540     INY
DB3E CA      3541     DEX
DB3F DOF5    3542     BNZ   INXIS
3543 ;
3544 ;
3545 ;
DB41 A690    3546 INXIT   LDX   A90
DB43 FO0B    3547     BZ    INXIV
DB45 3009    3548     BM    INXIV
3549 ;
3550 ;
3551 ;
DB47 A900    3552     LDA   #0
DB49 990001  3553 INXIU   STA   P1BASE, Y
DB4C C8      3554     INY
DB4D CA      3555     DEX
DB4E DOF9    3556     BNZ   INXIU
3557 ;
3558 ;
3559 ;
DB50 A67E    3560 INXIV   LDX   A7E
DB52 CA      3561     DEX
DB53 9A      3562     TXS
3563 ;
3564 ;
3565 ;
DB54 4C4DD2  3566     JMP   UPDBY1
3567 ;.PAGE 'Set logic - Difference (AND NOT)'
3568 ;
3569 ;
3570 ;
DB57 20FBDA  3571 DIF     JSR   FIXSET
3572 ;
3573 ;
3574 ;
DB5A A67C    3575     LDX   A7C
DB5C C57C    3576     CMP   A7C
DB5E 1001    3577     BNM   INXIX
DB60 AA      3578     TAX
DB61 E000    3579 INXIX   CPX   #0
DB63 FOOD    3580     BE    INXIZ
3581 ;
3582 ;
3583 ;
DB65 68      3584 INXIY   PLA
DB66 49FF    3585     EOR   #$FF
DB68 390001  3586     AND   P1BASE, Y
DB6B 990001  3587     STA   P1BASE, Y
DB6E C8      3588     INY
DB6F CA      3589     DEX
DB70 DOF3    3590     BNZ   INXIY
3591 ;

```

```

3592 ;
3593 ;
DB72 A67E 3594 INXIZ   LDX   A7E
DB74 CA   3595     DEX
DB75 9A   3596     TXS
3597 ;
3598 ;
3599 ;
DB76 4C4DD2 3600     JMP   UPDBY1
3601 ;.PAGE 'Set Logic - Union (OR)'
3602 ;
3603 ;
3604 ;
DB79 20FBDA 3605 UNI     JSR   FIXSET
3606 ;
3607 ;
3608 ;
DB7C C57C   3609     CMP   A7C
DB7E 3018   3610     BM   INXJC
3611 ;
3612 ;
3613 ;
DB80 A67C   3614     LDX   A7C
DB82 E000   3615     CPX   #0
DB84 FO0B   3616     BE   INXJB
3617 ;
3618 ;
3619 ;
DB86 68     3620 INXJA   PLA
DB87 190001 3621     ORA   P1BASE, Y
DB8A 990001 3622     STA   P1BASE, Y
DB8D C8     3623     INY
DB8E CA     3624     DEX
DB8F DOF5   3625     BNZ   INXJA
3626 ;
3627 ;
3628 ;
DB91 A67E   3629 INXJB   LDX   A7E
DB93 CA     3630     DEX
DB94 9A     3631     TXS
3632 ;
3633 ;
3634 ;
DB95 4C4DD2 3635     JMP   UPDBY1
3636 ;
3637 ;
3638 ;
DB98 857A   3639 INXJC   STA   A7A
DB9A C900   3640     CMP   #0
DB9C D002   3641     BNE   INXJD
DB9E F01B   3642     BE   INXJF
DBA0 BA     3643 INXJD   TSX
DBA1 E8     3644     INX
DBA2 A474   3645     LDY   A74
DBA4 A57A   3646     LDA   A7A
DBA6 8578   3647     STA   A78
DBA8 A57B   3648     LDA   A7A+1

```

```

DBAA 8579      3649      STA   A78+1
DBAC B90001    3650      INXJE LDA   P1BASE, Y
DBAF 1D0001    3651      ORA   P1BASE, X
DBB2 9D0001    3652      STA   P1BASE, X
DBB5 C8        3653      INY
DBB6 E8        3654      INX
DBB7 C678     3655      DEC   A78
DBB9 DOF1     3656      BNZ   INXJE
                3657      ;
                3658      ;
                3659      ;
DBBB A574     3660      INXJF LDA   A74
DBBD 18        3661      CLC
DBBE 657A     3662      ADC   A7A
DBC0 AA       3663      TAX
DBC1 CA       3664      DEX
DBC2 A57C     3665      LDA   A7C
DBC4 8578     3666      STA   A78
DBC6 A57D     3667      LDA   A7C+1
DBC8 8579     3668      STA   A78+1
                3669      ;
                3670      ;
                3671      ;
DBCA A47E     3672      LDY   A7E
DBCC 88       3673      DEY
DBCD B90001    3674      INXJG LDA   P1BASE, Y
DBD0 9D0001    3675      STA   P1BASE, X
DBD3 88       3676      DEY
DBD4 CA       3677      DEX
DBD5 C678     3678      DEC   A78
DBD7 DOF4     3679      BNZ   INXJG
                3680      ;
                3681      ;
                3682      ;
DBD9 9A       3683      TXS
                3684      ;
                3685      ;
                3686      ;
DBDA 467C     3687      LSR   A7C
DBDC A57D     3688      LDA   A7C+1
DBDE 48       3689      PHA
DBDF A57C     3690      LDA   A7C
DBE1 48       3691      PHA
                3692      ;
                3693      ;
                3694      ;
DBE2 4C4DD2    3695      JMP   UPDBY1
                3696      ; .PAGE 'Set logic - Adjust'
                3697      ;
                3698      ;
                3699      ;
                3700      ADJ      RDAUX
                +      I FDEF  BIG
                +      STA   RAMRDS
                +      ENDF
DBE5 A001     3701      LDY   #1
DBE7 B158     3702      LDA   (I PCPTR), Y

```

```

3703      RDMAIN
          +      IFDEF BIG
          +      STA  RAMRDC
          +      ENDF
DBE9 0A   3704      ASL  A
DBEA 8584 3705      STA  A84
          3706      ;
          3707      ;
          3708      ;
DBEC 68   3709      PLA
DBED 8582 3710      STA  A82
DBEF 68   3711      PLA
DBFO 8583 3712      STA  A82+1
          3713      ;
          3714      ; Convert word count to byte count
          3715      ;
DBF2 0682 3716      ASL  A82
DBF4 2683 3717      ROL  A82+1
          3718      ;
          3719      ; Branch if current size <> required size
          3720      ;
DBF6 A582 3721      LDA  A82
DBF8 C584 3722      CMP  A84
DBFA D003 3723      BNE  INXJH
          3724      ;
          3725      ; Go execute next P-code
          3726      ;
DBFC 4C3BD2 3727      JMP  UPDBY2
          3728      ;
          3729      ; Branch if current size < required size
          3730      ;
DBFF 301B 3731      INXJH  BM  INXJJ
          3732      ;
          3733      ;
          3734      ;
DC01 BA   3735      TSX
DC02 8A   3736      TXA
DC03 18   3737      CLC
DC04 6584 3738      ADC  A84
DC06 A8   3739      TAY
DC07 8A   3740      TXA
DC08 18   3741      CLC
DC09 6582 3742      ADC  A82
DC0B AA   3743      TAX
DC0C B90001 3744      INXJI  LDA  P1BASE, Y
DC0F 9D0001 3745      STA  P1BASE, X
DC12 88   3746      DEY
DC13 CA   3747      DEX
DC14 C684 3748      DEC  A84
DC16 D0F4 3749      BNZ  INXJI
          3750      ;
          3751      ;
          3752      ;
DC18 9A   3753      TXS
          3754      ;
          3755      ; Go execute next P-code
          3756      ;

```

```

DC19 4C3BD2    3757    JMP    UPDBY2
                3758    ;
                3759    ;
                3760    ;
DC1C A584     3761    INXJJ   LDA    A84
DC1E 38       3762    SEC
DC1F E582     3763    SBC    A82
DC21 8586     3764    STA    A86
DC23 C584     3765    CMP    A84
DC25 D00A     3766    BNE    INXJK
DC27 BA       3767    TSX
DC28 8A       3768    TXA
DC29 38       3769    SEC
DC2A E586     3770    SBC    A86
DC2C AA       3771    TAX
DC2D 9A       3772    TXS
DC2E E8       3773    INX
DC2F D016     3774    BNZ    INXJM
DC31 BA       3775    INXJK   TSX
DC32 8A       3776    TXA
DC33 A8       3777    TAY
DC34 C8       3778    INY
DC35 38       3779    SEC
DC36 E586     3780    SBC    A86
DC38 AA       3781    TAX
DC39 9A       3782    TXS
DC3A E8       3783    INX
DC3B B90001   3784    INXJL   LDA    P1BASE, Y
DC3E 9D0001   3785    STA    P1BASE, X
DC41 C8       3786    INY
DC42 E8       3787    INX
DC43 C682     3788    DEC    A82
DC45 D0F4     3789    BNZ    INXJL
                3790    ;
                3791    ;
                3792    ;
DC47 A486     3793    INXJM   LDY    A86
DC49 A900     3794    LDA    #0
DC4B 9D0001   3795    INXJN   STA    P1BASE, X
DC4E E8       3796    INX
DC4F 88       3797    DEY
DC50 D0F9     3798    BNZ    INXJN
                3799    ;
                3800    ; Go execute next P-code
                3801    ;
DC52 4C3BD2   3802    JMP    UPDBY2
                3803    ; .PAGE 'Set logic - Membership'
                3804    ;
                3805    ; Extract size of set
                3806    ;
DC55 68       3807    INN     PLA
DC56 8582     3808    STA    A82
DC58 68       3809    PLA
DC59 8583     3810    STA    A82+1
                3811    ;
                3812    ; Multiply size by 2
                3813    ;

```

DC5B 0682	3814	ASL	A82
DC5D 2683	3815	ROL	A82+1
	3816	;	
	3817	;	
	3818	;	
DC5F BA	3819	TSX	
DC60 8A	3820	TXA	
DC61 18	3821	CLC	
DC62 6582	3822	ADC	A82
DC64 A8	3823	TAY	
DC65 C8	3824	INY	
DC66 B90001	3825	LDA	P1BASE, Y
DC69 857C	3826	STA	A7C
DC6B C8	3827	INY	
DC6C B90001	3828	LDA	P1BASE, Y
DC6F 857D	3829	STA	A7C+1
DC71 8466	3830	STY	SPTEMP
	3831	;	
	3832	;	
	3833	;	
DC73 29FE	3834	AND	11111110
DC75 D031	3835	BNZ	INXJO
	3836	;	
	3837	;	
	3838	;	
DC77 A57C	3839	LDA	A7C
DC79 2907	3840	AND	00000111
DC7B 8575	3841	STA	A74+1
DC7D 667D	3842	ROR	A7C+1
DC7F A57C	3843	LDA	A7C
DC81 6A	3844	ROR	A
DC82 6A	3845	ROR	A
DC83 6A	3846	ROR	A
DC84 293F	3847	AND	00111111
DC86 8574	3848	STA	A74
	3849	;	
	3850	;	
	3851	;	
DC88 C582	3852	CMP	A82
DC8A 101C	3853	BNM	INXJO
	3854	;	
	3855	;	
	3856	;	
DC8C BA	3857	TSX	
DC8D E8	3858	INX	
DC8E 8A	3859	TXA	
DC8F 6574	3860	ADC	A74
DC91 AA	3861	TAX	
DC92 BD0001	3862	LDA	P1BASE, X
DC95 A475	3863	LDY	A74+1
DC97 39B2DC	3864	AND	BITTBL, Y
DC9A F00C	3865	BZ	INXJO
	3866	;	
	3867	;	
	3868	;	
DC9C A666	3869	LDX	SPTEMP
DC9E 9A	3870	TXS	

```

3871 ;
3872 ; Push TRUE
3873 ;
DC9F A900 3874 LDA #<1
DCA1 48 3875 PHA
DCA2 A901 3876 LDA #>1
DCA4 48 3877 PHA
3878 ;
3879 ;
3880 ;
DCA5 4C4DD2 3881 JMP UPDBY1
3882 ;
3883 ;
3884 ;
DCA8 A666 3885 INXJO LDX SPTEMP
DCAA 9A 3886 TXS
3887 ;
3888 ; Push FALSE
3889 ;
DCAB A900 3890 LDA #0
DCAD 48 3891 PHA
DCAE 48 3892 PHA
3893 ;
3894 ;
3895 ;
DCAF 4C4DD2 3896 JMP UPDBY1
3897 ;
3898 ;
3899 ;
DCB2 01 3900 BITTBL DB %00000001
DCB3 02 3901 DB %00000010
DCB4 04 3902 DB %00000100
DCB5 08 3903 DB %00001000
DCB6 10 3904 DB %00010000
DCB7 20 3905 DB %00100000
DCB8 40 3906 DB %01000000
DCB9 80 3907 DB %10000000
3908 ;.PAGE 'Set logic - Build singleton set'
3909 ;
3910 ;
3911 ;
DCBA 68 3912 SGS PLA
DCBB 8580 3913 STA A80
DCBD 68 3914 PLA
DCBE 8581 3915 STA A80+1
3916 ;
3917 ;
3918 ;
DCC0 A581 3919 LDA A80+1
DCC2 48 3920 PHA
DCC3 A580 3921 LDA A80
DCC5 48 3922 PHA
3923 ;
3924 ;
3925 ;
DCC6 A581 3926 LDA A80+1
DCC8 48 3927 PHA

```



```

DCC9 A580      3928      LDA   A80
DCCB 48        3929      PHA
                3930      ;.PAGE 'Set logic - Build a subrange set'
                3931      ;
                3932      ; Pull high bound
                3933      ;
DCCC 68        3934      SRS    PLA
DCCD 857A     3935      STA   A7A
DCCF 68        3936      PLA
DCDO 857B     3937      STA   A7A+1
                3938      ;
                3939      ; Pull low bound
                3940      ;
DCD2 68        3941      PLA
DCD3 857C     3942      STA   A7C
DCD5 68        3943      PLA
DCD6 857D     3944      STA   A7C+1
                3945      ;
                3946      ; Branch if low bound >= 0
                3947      ;
DCD8 A57D     3948      LDA   A7C+1
DCDA 1003     3949      BNM   INXJS
                3950      ;
                3951      ;
                3952      ;
DCDC 4C8BDD   3953      JMP   INXKD
                3954      ;
                3955      ; Branch if high bound < 512
                3956      ;
DCDF A57B     3957      INXJS  LDA   A7A+1
DCE1 29FE     3958      AND   #$FE
DCE3 F003     3959      BZ    INXJT
                3960      ;
                3961      ;
                3962      ;
DCE5 4C8BDD   3963      JMP   INXKD
                3964      ;
                3965      ;
                3966      ;
DCE8 38       3967      INXJT  SEC
DCE9 A57A     3968      LDA   A7A
DCEB E57C     3969      SBC   A7C
DCED 857E     3970      STA   A7E
DCEF A57B     3971      LDA   A7A+1
DCF1 E57D     3972      SBC   A7C+1
DCF3 857F     3973      STA   A7E+1
                3974      ;
                3975      ;
                3976      ;
DCF5 A57F     3977      LDA   A7E+1
DCF7 1003     3978      BNM   INXJU
                3979      ;
                3980      ;
                3981      ;
DCF9 4C84DD   3982      JMP   INXKC
                3983      ;
                3984      ;

```

```

3985 ;
DCFC A57C 3986 INXJU LDA A7C
DCFE 290F 3987 AND #SOF
DD00 0A 3988 ASL A
DD01 8575 3989 STA A74+1
DD03 A57C 3990 LDA A7C
DD05 A204 3991 LDX #4
DD07 667D 3992 INXJV ROR A7C+1
DD09 6A 3993 ROR A
DD0A CA 3994 DEX
DD0B DOFA 3995 BNZ INXJV
DD0D 8574 3996 STA A74
3997 ;
3998 ;
3999 ;
DD0F A57A 4000 LDA A7A
DD11 290F 4001 AND #SOF
DD13 0A 4002 ASL A
DD14 8577 4003 STA A76+1
DD16 A57A 4004 LDA A7A
DD18 A204 4005 LDX #4
DD1A 667B 4006 INXJW ROR A7A+1
DD1C 6A 4007 ROR A
DD1D CA 4008 DEX
DD1E DOFA 4009 BNZ INXJW
DD20 8576 4010 STA A76
4011 ;
4012 ;
4013 ;
DD22 38 4014 SEC
DD23 A576 4015 LDA A76
DD25 E574 4016 SBC A74
DD27 AA 4017 TAX
4018 ;
4019 ; Construct leftmost 16 bits of set
4020 ;
DD28 A477 4021 LDY A76+1
DD2A B994DD 4022 LDA INXKF+2, Y
DD2D 8578 4023 STA A78
DD2F B995DD 4024 LDA INXKF+3, Y
DD32 8579 4025 STA A78+1
4026 ;
4027 ;
4028 ;
DD34 E000 4029 CPX #0
DD36 F017 4030 BE INXJZ
4031 ;
4032 ; Push left edge of set
4033 ;
DD38 A579 4034 LDA A78+1
DD3A 48 4035 PHA
DD3B A578 4036 LDA A78
DD3D 48 4037 PHA
4038 ;
4039 ; Get ready to fill center of set
4040 ;
DD3E A9FF 4041 LDA #SFF

```

```

DD40 8578      4042      STA   A78
DD42 8579      4043      STA   A78+1
                4044      ;
                4045      ;   Unconditional branch
                4046      ;
DD44 D006      4047      BNZ   INXJY
                4048      ;
                4049      ;
                4050      ;
DD46 A579      4051      INXJX   LDA   A78+1
DD48 48        4052      PHA
DD49 A578      4053      LDA   A78
DD4B 48        4054      PHA
                4055      ;
                4056      ;   Branch if more 16 bit pieces to push
                4057      ;
DD4C CA        4058      INXJY   DEX
DD4D DOF7      4059      BNZ   INXJX
                4060      ;
                4061      ;   Construct rightmost 16 bits of set
                4062      ;
DD4F A578      4063      INXJZ   LDA   A78
DD51 A475      4064      LDY   A74+1
DD53 39B4DD    4065      AND   INXKL, Y
DD56 8578      4066      STA   A78
DD58 A579      4067      LDA   A78+1
DD5A 39B5DD    4068      AND   INXKL+1, Y
DD5D 8579      4069      STA   A78+1
                4070      ;
                4071      ;   Push right edge of set
                4072      ;
DD5F A579      4073      LDA   A78+1
DD61 48        4074      PHA
DD62 A578      4075      LDA   A78
DD64 48        4076      PHA
                4077      ;
                4078      ;   Get ready for zero fill
                4079      ;
DD65 A900      4080      LDA   #0
DD67 8578      4081      STA   A78
DD69 8579      4082      STA   A78+1
                4083      ;
                4084      ;   Branch if zero fill not needed
                4085      ;
DD6B A674      4086      LDX   A74
DD6D F009      4087      BZ    INXKB
                4088      ;
                4089      ;   Push zero fill
                4090      ;
DD6F A579      4091      INXKA   LDA   A78+1
DD71 48        4092      PHA
DD72 A578      4093      LDA   A78
DD74 48        4094      PHA
DD75 CA        4095      DEX
DD76 DOF7      4096      BNZ   INXKA
                4097      ;
                4098      ;   Push set size

```

```

4099 ;
DD78 A900 4100 INXKB LDA #0
DD7A 48 4101 PHA
DD7B 18 4102 CLC
DD7C A576 4103 LDA A76
DD7E 6901 4104 ADC #1
DD80 48 4105 PHA
4106 ;
4107 ;
4108 ;
DD81 4C4DD2 4109 JMP UPDBY1
4110 ;
4111 ;
4112 ;
DD84 A900 4113 INXKC LDA #0
DD86 48 4114 PHA
DD87 48 4115 PHA
4116 ;
4117 ;
4118 ;
DD88 4C4DD2 4119 JMP UPDBY1
4120 ;
4121 ;
4122 ;
DD8B A900 4123 INXKD LDA #0
DD8D 48 4124 PHA
DD8E 48 4125 PHA
4126 ;
4127 ;
4128 ;
DD8F 4CB7D1 4129 JMP ERR1
4130 ; .PAGE 'Masks for packed field manipulation and for set
construction'
4131 ;
4132 ;
4133 ;
DD92 0000 4134 INXKF DW %0000000000000000 ; $0000
DD94 0100 4135 DW %0000000000000001 ; $0001
DD96 0300 4136 DW %0000000000000011 ; $0003
DD98 0700 4137 DW %0000000000000111 ; $0007
DD9A 0F00 4138 DW %0000000000001111 ; $000F
DD9C 1F00 4139 DW %0000000000011111 ; $001F
DD9E 3F00 4140 DW %0000000000111111 ; $003F
DDA0 7F00 4141 DW %0000000001111111 ; $007F
DDA2 FF00 4142 DW %0000000011111111 ; $00FF
DDA4 FF01 4143 DW %0000000111111111 ; $01FF
DDA6 FF03 4144 DW %0000001111111111 ; $03FF
DDA8 FF07 4145 DW %0000011111111111 ; $07FF
DDAA FFOF 4146 DW %0000111111111111 ; $0FFF
DDAC FF1F 4147 DW %0001111111111111 ; $1FFF
DDAE FF3F 4148 DW %0011111111111111 ; $3FFF
DDB0 FF7F 4149 DW %0111111111111111 ; $7FFF
DDB2 FFFF 4150 DW %1111111111111111 ; $FFFF
4151 ;
4152 ; AND masks
4153 ;
DDB4 FFFF 4154 INXKL DW %1111111111111111 ; $FFFF

```

```

DDB6 FEFF      4155      DW      %1111111111111110 ; SFFFE
DDB8 FCFF      4156      DW      %1111111111111100 ; SFFFC
DDBA F8FF      4157      DW      %1111111111111000 ; SFFF8
DDBC F0FF      4158      DW      %1111111111110000 ; SFFF0
DDBE E0FF      4159      DW      %1111111111100000 ; SFFE0
DDC0 C0FF      4160      DW      %1111111111000000 ; SFFC0
DDC2 80FF      4161      DW      %1111111110000000 ; SFF80
DDC4 00FF      4162      DW      %1111111100000000 ; SFF00
DDC6 00FE      4163      DW      %1111111000000000 ; SFE00
DDC8 00FC      4164      DW      %1111110000000000 ; SFC00
DDCA 00F8      4165      DW      %1111100000000000 ; SF800
DDCC 00F0      4166      DW      %1111000000000000 ; SF000
DDCE 00E0      4167      DW      %1110000000000000 ; SE000
DDDO 00C0      4168      DW      %1100000000000000 ; SC000
DDD2 0080      4169      DW      %1000000000000000 ; $8000
                4170      ; . PAGE 'Compare logic'
                4171      ;
                4172      ;
                4173      ;
DDD4 A906      4174      NEQ      LDA      #6
DDD6 1012      4175      BNM      INXKN
                4176      ;
                4177      ;
                4178      ;
DDD8 A904      4179      GRT      LDA      #4
DDDA 100E      4180      BNM      INXKN
                4181      ;
                4182      ;
                4183      ;
DDDC A902      4184      LES      LDA      #2
DDDE 100A      4185      BNM      INXKN
                4186      ;
                4187      ;
                4188      ;
DDE0 A905      4189      GEQ      LDA      #5
DDE2 1006      4190      BNM      INXKN
                4191      ;
                4192      ;
                4193      ;
DDE4 A903      4194      LEQ      LDA      #3
DDE6 1002      4195      BNM      INXKN
                4196      ;
                4197      ;
                4198      ;
DDE8 A901      4199      EQU      LDA      #1
                4200      ;
                4201      ;
                4202      ;
DDEA 856C      4203      INXKN      STA      CMPTYP
                4204      ;
                4205      ;
                4206      ;
                4207      RDAUX
                +      IFDEF BIG
                +      STA      RAMRDS
                +      ENDIF
                4208      ;

```

```

4209 ; Extract data type code
4210 ;
DDEC A001 4211 LDY #1
DDEE B158 4212 LDA (IPCPTR),Y
4213 ;
4214 ;
4215 ;
4216 RDMAIN
+ IFDEF BIG
+ STA RAMRDC
+ ENDF
4217 ;
4218 ; Branch if not code for reals
4219 ;
DDF0 C902 4220 CMP #2
DDF2 D003 4221 BNE INXKS
4222 ;
4223 ; Go compare reals
4224 ;
DDF4 4CDCDE 4225 JMP CMPR
4226 ;
4227 ; Branch if not code for strings
4228 ;
DDF7 C904 4229 INXKS CMP #4
DDF9 D003 4230 BNE INXKT
4231 ;
4232 ; Go compare strings
4233 ;
DDFB 4C64DE 4234 JMP CMPSTR
4235 ;
4236 ; Branch if not code for booleans
4237 ;
DDFE C906 4238 INXKT CMP #6
DE00 D003 4239 BNE INXKU
4240 ;
4241 ; Go compare booleans
4242 ;
DE02 4C16DF 4243 JMP CMPB
4244 ;
4245 ; Branch if not code for sets
4246 ;
DE05 C908 4247 INXKU CMP #8
DE07 D003 4248 BNE INXKV
4249 ;
4250 ; Go compare sets
4251 ;
DE09 4C27E0 4252 JMP CMPSET
4253 ;
4254 ; Branch if code for byte arrays
4255 ;
DE0C C90A 4256 INXKV CMP #10
DE0E F002 4257 BE CMPBAR
4258 ;
4259 ; Default to compare words
4260 ;
DE10 D00C 4261 BNE INXKX
4262 ;

```

```

4263 ;
4264 ;
DE12 A002 4265 CMPBAR LDY #2
DE14 2055D1 4266 JSR EBI G2M
DE17 465F 4267 LSR OPRND+1
DE19 665E 4268 ROR OPRND
DE1B 4C23DE 4269 JMP I NXKY
4270 ;
4271 ;
4272 ;
DE1E A002 4273 I NXKX LDY #2
DE20 2055D1 4274 JSR EBI G2M
4275 ;
4276 ;
4277 ;
DE23 68 4278 I NXKY PLA
DE24 8576 4279 STA A76
DE26 68 4280 PLA
DE27 8577 4281 STA A76+1
DE29 68 4282 PLA
DE2A 8574 4283 STA A74
DE2C 68 4284 PLA
DE2D 8575 4285 STA A74+1
4286 ;
4287 ; Increment P-code pointer
4288 ;
DE2F E658 4289 INC I PCPTR
DE31 D002 4290 BNZ I NXKZ
DE33 E659 4291 INC I PCPTR+1
4292 ;
4293 ;
4294 ;
DE35 A000 4295 I NXKZ LDY #0
DE37 A65E 4296 LDX OPRND
4297 ;
4298 ;
4299 ;
DE39 4C4FDE 4300 JMP I NXLE
4301 ;
4302 ;
4303 ;
DE3C B174 4304 I NXLA LDA (A74), Y
DE3E D176 4305 CMP (A76), Y
DE40 D016 4306 BNE I NXLF
4307 ;
4308 ; Increment source pointer
4309 ;
DE42 E674 4310 INC A74
DE44 D002 4311 BNZ I NXLC
DE46 E675 4312 INC A74+1
4313 ;
4314 ; Increment destination pointer
4315 ;
DE48 E676 4316 I NXLC INC A76
DE4A D002 4317 BNZ I NXLD
DE4C E677 4318 INC A76+1
4319 ;

```

```

4320 ;
4321 ;
DE4E CA 4322 INXLD DEX
DE4F DOEB 4323 INXLE BNZ INXLA
DE51 C65F 4324 DEC OPRND+1
DE53 10E7 4325 BNM INXLA
4326 ;
4327 ;
4328 ;
DE55 4CC2DE 4329 JMP ISEQ
4330 ;
4331 ;
4332 ;
DE58 B003 4333 INXLF BC INXLG
4334 ;
4335 ;
4336 ;
DE5A 4CC6DE 4337 JMP ISLOW
4338 ;
4339 ;
4340 ;
DE5D 4CBEDE 4341 INXLG JMP ISHIGH
4342 ;
4343 ;
4344 ;
DE60 7E00 4345 ADRA7E DW A7E
DE62 8000 4346 ADRA80 DW A80
4347 ;.PAGE 'Compare logic - Strings'
4348 ;
4349 ;
4350 ;
DE64 68 4351 CMPSTR PLA
DE65 8576 4352 STA A76
DE67 68 4353 PLA
DE68 8577 4354 STA A76+1
DE6A D012 4355 BNZ INXLM
DE6C A576 4356 LDA A76
DE6E 857F 4357 STA A7E+1
DE70 A901 4358 LDA #1
DE72 857E 4359 STA A7E
DE74 AD60DE 4360 LDA ADRA7E
DE77 8576 4361 STA A76
DE79 AD61DE 4362 LDA ADRA7E+1
DE7C 8577 4363 STA A76+1
DE7E 68 4364 INXLM PLA
DE7F 8574 4365 STA A74
DE81 68 4366 PLA
DE82 8575 4367 STA A74+1
DE84 D012 4368 BNZ INXLN
DE86 A574 4369 LDA A74
DE88 8581 4370 STA A80+1
DE8A A901 4371 LDA #1
DE8C 8580 4372 STA A80
DE8E AD62DE 4373 LDA ADRA80
DE91 8574 4374 STA A74
DE93 AD63DE 4375 LDA ADRA80+1
DE96 8575 4376 STA A74+1

```



```

DE98 A000      4377  INXLN   LDY   #0
DE9A B174      4378      LDA   (A74), Y
DE9C D176      4379      CMP   (A76), Y
DE9E 9002      4380      BL    INXLO
DEA0 B176      4381      LDA   (A76), Y
DEA2 AA        4382  INXLO   TAX
DEA3 E8        4383      INX
DEA4 CA        4384  INXLP   DEX
DEA5 F00B      4385      BZ    INXLQ
DEA7 C8        4386      INY
DEA8 B174      4387      LDA   (A74), Y
DEAA D176      4388      CMP   (A76), Y
DEAC F0F6      4389      BE    INXLP
DEAE B00E      4390      BGE   I SHI GH
DEB0 9014      4391      BL    I SLOW
DEB2 A000      4392  INXLQ   LDY   #0
DEB4 B174      4393      LDA   (A74), Y
DEB6 D176      4394      CMP   (A76), Y
DEB8 900C      4395      BL    I SLOW
DEBA F006      4396      BE    I SEQ
DEBC D000      4397      BNE   I SHI GH
          4398 ;
          4399 ;
          4400 ;
DEBE A904      4401  I SHI GH   LDA   #4
DECO 1006      4402      BNM   INXMH
          4403 ;
          4404 ;
          4405 ;
DEC2 A901      4406  I SEQ     LDA   #1
DEC4 1002      4407      BNM   INXMH
          4408 ;
          4409 ;
          4410 ;
DEC6 A902      4411  I SLOW   LDA   #2
DEC8 256C      4412  INXMH   AND   CMPTYP
DECA F009      4413      BZ    FALSE1
          4414 ;
          4415 ;   Push TRUE onto stack
          4416 ;
DECC A900      4417      LDA   #<1
DECE 48        4418      PHA
DECF A901      4419      LDA   #>1
DED1 48        4420      PHA
          4421 ;
          4422 ;
          4423 ;
DED2 4C3BD2    4424      JMP   UPDBY2
          4425 ;
          4426 ;   Push FALSE onto stack
          4427 ;
DED5 A900      4428  FALSE1   LDA   #0
DED7 48        4429      PHA
DED8 48        4430      PHA
          4431 ;
          4432 ;
          4433 ;

```

```

DED9 4C3BD2    4434    JMP    UPDBY2
              4435    ;. PAGE 'Compare logic - Reals'
              4436    ;
              4437    ;
              4438    ;
DEDC A207     4439    CMPR    LDX    #7
DEDE 68       4440    INXML   PLA
DEDF 9574     4441    STA    A74, X
DEE1 CA       4442    DEX
DEE2 10FA     4443    BNM    INXML
DEE4 A001     4444    LDY    #1
DEE6 A980     4445    LDA    #S80
DEE8 4578     4446    EOR    A78
DEEA 8578     4447    STA    A78
DEEC 3006     4448    BM     INXMM
DEEE A574     4449    LDA    A74
DEFO 1002     4450    BNM    INXMM
DEF2 A0F4     4451    LDY    #SF4
DEF4 A980     4452    INXMM  LDA    #S80
DEF6 4574     4453    EOR    A74
DEF8 8574     4454    STA    A74
DEFA A200     4455    LDX    #0
DEFC B574     4456    INXMO  LDA    A74, X
DEFE D578     4457    CMP    A78, X
DFO0 D007     4458    BNE    INXMP
DFO2 E8       4459    INX
DFO3 E004     4460    CPX    #4
DFO5 D0F5     4461    BNE    INXMO
DFO7 FOB9     4462    BE     ISEQ
DFO9 98       4463    INXMP  TYA
DFOA 3005     4464    BM     INXMQ
DFOC BOBO     4465    BC     ISHIGH
DFOE 4CC6DE   4466    JMP    ISLOW
DF11 90AB     4467    INXMQ  BNC    ISHIGH
DF13 4CC6DE   4468    JMP    ISLOW
              4469    ;. PAGE 'Compare logic - Booleans'
              4470    ;
              4471    ; Use low order bit of first comparator
              4472    ;
DF16 68       4473    CMPB   PLA
DF17 2901     4474    AND    #%00000001
DF19 8574     4475    STA    A74
DF1B 68       4476    PLA
              4477    ;
              4478    ; Use low order bit of second comparator
              4479    ;
DF1C 68       4480    PLA
DF1D 2901     4481    AND    #%00000001
DF1F A8       4482    TAY
DF20 68       4483    PLA
              4484    ;
              4485    ;
              4486    ;
DF21 98       4487    TYA
DF22 C574     4488    CMP    A74
DF24 90A0     4489    BL     ISLOW
DF26 F09A     4490    BE     ISEQ

```

```

4491 ;
4492 ;
4493 ;
DF28 4CBEDE 4494     JMP     I SHIGH
4495 ;.PAGE 'Compare logic - Integers'
4496 ;
4497 ;
4498 ;
DF2B A902   4499     LESI     LDA     #2
DF2D 100E   4500     BNM     INXMS
4501 ;
4502 ;
4503 ;
DF2F A904   4504     GRTI     LDA     #4
DF31 100A   4505     BNM     INXMS
4506 ;
4507 ;
4508 ;
DF33 A903   4509     LEQI     LDA     #3
DF35 1006   4510     BNM     INXMS
4511 ;
4512 ;
4513 ;
DF37 A905   4514     GEQI     LDA     #5
DF39 1002   4515     BNM     INXMS
4516 ;
4517 ;
4518 ;
DF3B A906   4519     NEQI     LDA     #6
4520 ;
4521 ;
4522 ;
DF3D 856C   4523     INXMS    STA     CMPTYP
DF3F 68     4524     PLA
DF40 8576   4525     STA     A76
DF42 68     4526     PLA
DF43 8577   4527     STA     A76+1
DF45 68     4528     PLA
DF46 8574   4529     STA     A74
DF48 68     4530     PLA
DF49 8575   4531     STA     A74+1
DF4B 4577   4532     EOR     A76+1
DF4D 3010   4533     BM      INXMX
DF4F A575   4534     LDA     A74+1
DF51 C577   4535     CMP     A76+1
DF53 D006   4536     BNE     INXMW
DF55 A574   4537     LDA     A74
DF57 C576   4538     CMP     A76
DF59 F02F   4539     BE      INXNC
DF5B 9031   4540     INXMW   BL      INXND
DF5D B027   4541     BGE     INXNA
DF5F A575   4542     INXMX   LDA     A74+1
DF61 302B   4543     BM      INXND
DF63 1021   4544     BNM     INXNA
4545 ;.PAGE 'Compare logic - Integer equality'
4546 ;
4547 ;

```

```

4548 ;
4549   ifndef ORIG
4550   EQUI      TSX
4551       PLA
4552       CMP   P1BASE+3, X
4553       BNE   INXPP
4554       PLA
4555       CMP   P1BASE+4, X
4556       BNE   INXPQ
4557 ;
4558 ;   Push TRUE onto stack
4559 ;
4560       LDA   #<1
4561       STA   P1BASE+4, X
4562       LDA   #>1
4563       STA   P1BASE+3, X
4564 ;
4565 ;
4566 ;
4567       JMP   UPDBY1
4568 ;
4569 ;   Push FALSE onto stack
4570 ;
4571   INXPP   PLA
4572       LDA   #0
4573       STA   P1BASE+4, X
4574       STA   P1BASE+3, X
4575 ;
4576 ;
4577 ;
4578       JMP   UPDBY1
4579 ;
4580 ;   Push FALSE onto stack
4581 ;
4582   INXPQ   LDA   #0
4583       STA   P1BASE+4, X
4584       STA   P1BASE+3, X
4585       JMP   UPDBY1
4586 ;
4587 ;   Push TRUE onto stack
4588 ;
4589   TRUE    LDA   #<1
4590       PHA
4591       LDA   #>1
4592       PHA
4593 ;
4594 ;
4595 ;
4596       JMP   UPDBY1
4597   else
4598 ;
4599 ;
4600 ;
DF65 68   4601   EQUI      PLA
DF66 8574 4602       STA   A74
DF68 68   4603       PLA
DF69 8575 4604       STA   A74+1

```

```

DF6B 68      4605    PLA
DF6C C574    4606    CMP    A74
DF6E D00E    4607    BNE    INXMZ
DF70 68      4608    PLA
DF71 C575    4609    CMP    A74+1
DF73 D01F    4610    BNE    FALSE2
            4611    ;
            4612    ;   Push TRUE onto stack
            4613    ;
DF75 A900    4614    TRUE   LDA    #<1
DF77 48      4615    PHA
DF78 A901    4616    LDA    #>1
DF7A 48      4617    PHA
            4618    ;
            4619    ;
            4620    ;
DF7B 4C4DD2  4621    JMP    UPDBY1
            4622    ;
            4623    ;
            4624    ;
DF7E 68      4625    INXMZ   PLA
            4626    ;
            4627    ;   Push FALSE onto stack
            4628    ;
DF7F A900    4629    LDA    #0
DF81 48      4630    PHA
DF82 48      4631    PHA
            4632    ;
            4633    ;
            4634    ;
DF83 4C4DD2  4635    JMP    UPDBY1
            4636    endi f
            4637    ;
            4638    ;
            4639    ;
DF86 A904    4640    INXNA   LDA    #4
DF88 1006    4641    BNM    INXNF
            4642    ;
            4643    ;
            4644    ;
DF8A A901    4645    INXNC   LDA    #1
DF8C 1002    4646    BNM    INXNF
            4647    ;
            4648    ;
            4649    ;
DF8E A902    4650    INXND   LDA    #2
DF90 256C    4651    INXNF   AND    CMPTYP
DF92 DOE1    4652    BNZ    TRUE
            4653    ;
            4654    ;   Push FALSE onto stack
            4655    ;
DF94 A900    4656    FALSE2  LDA    #0
DF96 48      4657    PHA
DF97 48      4658    PHA
            4659    ;
            4660    ;
            4661    ;

```

```

DF98 4C4DD2    4662    JMP    UPDBY1
              4663    ;.PAGE 'Compare logic - Subroutine for Sets'
              4664    ;
              4665    ; Pull return address off stack and save it
              4666    ;
DF9B 68        4667    INXNI   PLA
DF9C 858E      4668    STA    A8E
DF9E 68        4669    PLA
DF9F 858F      4670    STA    A8E+1
              4671    ;
              4672    ;
              4673    ;
DFA1 68        4674    PLA
DFA2 0A        4675    ASL    A
DFA3 857C      4676    STA    A7C
DFA5 857E      4677    STA    A7E
DFA7 68        4678    PLA
DFA8 BA        4679    TSX
DFA9 E8        4680    INX
DFAA 8676      4681    STX    A76
DFAC 8A        4682    TXA
DFAD 18        4683    CLC
DFAE 657C      4684    ADC    A7C
DFB0 AA        4685    TAX
DFB1 BD0001    4686    LDA    P1BASE, X
DFB4 0A        4687    ASL    A
DFB5 857A      4688    STA    A7A
DFB7 C57E      4689    CMP    A7E
DFB9 1002      4690    BNM    INXNJ
DFBB 857E      4691    STA    A7E
DFBD 38        4692    INXNJ  SEC
DFBE E57C      4693    SBC    A7C
DFC0 8586      4694    STA    A86
DFC2 E8        4695    INX
DFC3 8A        4696    TXA
DFC4 18        4697    CLC
DFC5 657A      4698    ADC    A7A
DFC7 8580      4699    STA    A80
DFC9 E8        4700    INX
DFCA 8674      4701    STX    A74
              4702    ;
              4703    ; Return to caller
              4704    ;
DFCC E68E      4705    INC    A8E
DFCE D002      4706    BNZ    INXNK
DFD0 E68F      4707    INC    A8E+1
DFD2 6C8E00    4708    INXNK  JMP    (A8E)
testing'      4709    ;.PAGE 'Compare logic - Subroutines used by subset/superset
              4710    ;
              4711    ;
              4712    ;
DFD5 A900      4713    SUPSET LDA    #0
DFD7 38        4714    SEC
DFD8 E586      4715    SBC    A86
DFDA A8        4716    TAY
DFDB 4CE0DF    4717    JMP    INXNL

```

```

4718 ;
4719 ;
4720 ;
DFDE A486 4721 SUBSET LDY A86
4722 ;
4723 ;
4724 ;
DFE0 C000 4725 INXNL CPY #0
DFE2 F00C 4726 BE INXNN
DFE4 300A 4727 BM INXNN
DFE6 18 4728 CLC
DFE7 BD0001 4729 INXNM LDA P1BASE, X
DFEA D005 4730 BNZ INXNP
DFEC E8 4731 INX
DFED 88 4732 DEY
DFEE DOF7 4733 BNZ INXNM
DFF0 38 4734 INXNN SEC
DFF1 60 4735 INXNP RTS
4736 ;.PAGE 'Compare logic - Test if set A is equal to set B'
4737 ;
4738 ; Pull return address off stack and save it
4739 ;
DFF2 68 4740 SETEQ PLA
DFF3 858C 4741 STA A8C
DFF5 68 4742 PLA
DFF6 858D 4743 STA A8C+1
4744 ;
4745 ;
4746 ;
DFF8 209BDF 4747 JSR INXNI
4748 ;
4749 ;
4750 ;
DFFB A674 4751 LDX A74
DFFD A47E 4752 LDY A7E
DFFF F00A 4753 BZ INTZZ
E001 68 4754 INUAA PLA
E002 DD0001 4755 CMP P1BASE, X
E005 D016 4756 BNE INUAC
E007 E8 4757 INX
E008 88 4758 DEY
E009 DOF6 4759 BNZ INUAA
E00B A586 4760 INTZZ LDA A86
E00D 3006 4761 BM INUAB
E00F 20DEDF 4762 JSR SUBSET
E012 4C1EE0 4763 JMP INUAD
E015 BA 4764 INUAB TSX
E016 E8 4765 INX
E017 20D5DF 4766 JSR SUPSET
E01A 4C1EE0 4767 JMP INUAD
E01D 18 4768 INUAC CLC
4769 ;
4770 ; Increment return address
4771 ;
E01E E68C 4772 INUAD INC A8C
E020 D002 4773 BNZ INUAF
E022 E68D 4774 INC A8C+1

```

```

4775 ;
4776 ; Return to caller
4777 ;
E024 6C8C00 4778 INUAF    JMP    (A8C)
4779 ;.PAGE 'Compare logic - Sets'
4780 ;
4781 ; Branch if not testing for set equality
4782 ;
E027 A56C   4783 CMPSET   LDA    CMPTYP
E029 C901   4784      CMP    #1
E02B D003   4785      BNE   INUAG
4786 ;
4787 ; Go test for set equality
4788 ;
E02D 4C3DE0 4789      JMP    INUAI
4790 ;
4791 ; Branch if not testing for set inequality
4792 ;
E030 C906   4793 INUAG    CMP    #6
E032 D003   4794      BNE   INUAH
4795 ;
4796 ; Go test for set inequality
4797 ;
E034 4C43E0 4798      JMP    INUAJ
4799 ;
4800 ; Branch if not testing for subset
4801 ;
E037 C903   4802 INUAH    CMP    #3
E039 D02F   4803      BNE   INUAP
4804 ;
4805 ; Go test if subset
4806 ;
E03B F012   4807      BE    INUAM
4808 ;
4809 ; Set equality?
4810 ;
E03D 20F2DF 4811 INUAI    JSR    SETEQ
E040 4C8CEO 4812      JMP    STOANS
4813 ;
4814 ; Set inequality?
4815 ;
E043 20F2DF 4816 INUAJ    JSR    SETEQ
E046 B003   4817      BC    INUAK
E048 38     4818      SEC
E049 B001   4819      BC    INUAL
E04B 18     4820 INUAK    CLC
E04C 4C8CEO 4821 INUAL    JMP    STOANS
4822 ;
4823 ; Subset?
4824 ;
E04F 209BDF 4825 INUAM    JSR    INXNI
E052 A674   4826      LDX   A74
E054 A47E   4827      LDY   A7E
E056 F00C   4828      BZ    INUA0
E058 68     4829 INUAN    PLA
E059 49FF   4830      EOR   #$FF
E05B 3D0001 4831      AND   P1BASE, X

```



```

E05E D02B      4832      BNZ      INUAS
E060 E8        4833      INX
E061 88        4834      DEY
E062 D0F4      4835      BNZ      INUAN
E064 20DEDF    4836      INUAO    JSR      SUBSET
E067 4C8CEO    4837      JMP      STOANS
                4838      ;
                4839      ;   Superset?
                4840      ;
E06A 209BDF    4841      INUAP    JSR      INXNI
E06D A474      4842      LDY      A74
E06F BA        4843      TSX
E070 E8        4844      INX
E071 A57E      4845      LDA      A7E
E073 F010      4846      BZ       INUAR
E075 B90001    4847      INUAQ    LDA      P1BASE, Y
E078 49FF      4848      EOR      #SFF
E07A 3D0001    4849      AND      P1BASE, X
E07D D00C      4850      BNZ      INUAS
E07F E8        4851      INX
E080 C8        4852      INY
E081 C67E      4853      DEC      A7E
E083 D0F0      4854      BNZ      INUAQ
E085 20D5DF    4855      INUAR    JSR      SUPSET
E088 4C8CEO    4856      JMP      STOANS
E08B 18        4857      INUAS    CLC
                4858      ;
                4859      ;   Restore stack pointer
                4860      ;
E08C A680      4861      STOANS   LDX      A80
E08E 9A        4862      TXS
                4863      ;
                4864      ;   Convert carry flag to TRUE or FALSE
                4865      ;
E08F A000      4866      LDY      #0
E091 8485      4867      STY      A85
E093 9001      4868      BNC      INUAY
E095 C8        4869      INY
E096 8484      4870      INUAY    STY      A84
                4871      ;
                4872      ;   Push boolean result
                4873      ;
E098 A585      4874      LDA      A85
E09A 48        4875      PHA
E09B A584      4876      LDA      A84
E09D 48        4877      PHA
                4878      ;
                4879      ;
                4880      ;
E09E 4C3BD2    4881      JMP      UPDBY2
                4882      ;.PAGE 'Subroutine for external procedure call'
                4883      ;
                4884      ;   Pull return address off stack and save it
                4885      ;
EOA1 68        4886      CALLX    PLA
EOA2 858E      4887      STA      A8E
EOA4 68        4888      PLA

```

```

EOA5 858F      4889      STA   A8E+1
                4890      ;
                4891      ; Extract segment data from intrinsic unit table
                4892      ;
EOA7 A586      4893      LDA   SEGNO
EOA9 0A        4894      ASL   A
EOAA A8        4895      TAY
EOAB B99EBD    4896      LDA   IUTBL, Y
EOAE 8582      4897      STA   A82
EOB0 8574      4898      STA   A74
EOB2 B99FBD    4899      LDA   IUTBL+1, Y
EOB5 8583      4900      STA   A82+1
EOB7 8575      4901      STA   A74+1
                4902      ;
                4903      ;
                4904      ;
                4905      * JSR   INUBE
                4906      * RTS
EOB9 4CD2E0    4907      JMP   INUBE
                4908      ;.PAGE 'Call all procedures except external'
                4909      ;
                4910      ; Pull return address off stack and save it
                4911      ;
EOBC 68        4912      CALAXX  PLA
EOBD 858E      4913      STA   A8E
EOBF 68        4914      PLA
EOC0 858F      4915      STA   A8E+1
                4916      ;
                4917      ;
                4918      ;
EOC2 A556      4919      LDA   SEGPTR
EOC4 8582      4920      STA   A82
EOC6 8574      4921      STA   A74
EOC8 A557      4922      LDA   SEGPTR+1
EOCA 8583      4923      STA   A82+1
EOCC 8575      4924      STA   A74+1
                4925      ;
                4926      ;
                4927      ;
EOCE A9FF      4928      LDA   #- 1
EODO 8586      4929      STA   SEGNO
                4930      ;
                4931      ;
                4932      ;
                4933      * JSR   INUBE
                4934      * RTS
                4935      ;.PAGE 'Common logic for subroutine used by all procedure calls'
                4936      ;
                4937      ; Initialize working program stack pointer
                4938      ;
                4939      INUBE   RDAUX
                +   IFDEF  BIG
                +   STA   RAMRDS
                +   ENDF
EOD2 A55C      4940      LDA   PSKPTR
EOD4 857A      4941      STA   A7A
EOD6 A55D      4942      LDA   PSKPTR+1

```

```

EOD8 857B      4943      STA   A7A+1
                4944      ifdef ORIG
                4945      ;
                4946      ;   Update P-code pointer
                4947      ;
EODA A558      4948      LDA   IPCPTR
EODC 18        4949      CLC
EODD 6902      4950      ADC   #2
EODF 8558      4951      STA   IPCPTR
EOE1 9002      4952      BNC   INUBF
EOE3 E659      4953      INC   IPCPTR+1
                4954      INUBF
                4955      endif
                4956      ;
                4957      ;   Subtract procedure number times two from A74PTR
                4958      ;
EOE5 A574      4959      LDA   A74
EOE7 38        4960      CLB
EOE8 E578      4961      SBB   PROCNO
EOEA 8574      4962      STA   A74
EOEC B002      4963      BNB   INUBG
EOEE C675      4964      DEC   A74+1
EOF0 A574      4965      INUBG  LDA   A74
EOF2 38        4966      CLB
EOF3 E578      4967      SBB   PROCNO
EOF5 8574      4968      STA   A74
EOF7 B002      4969      BNB   INUBH
EOF9 C675      4970      DEC   A74+1
                4971      ;
                4972      ;   Set pointer into jump table
                4973      ;
EOFB A000      4974      INUBH  LDY   #0
EOFD 38        4975      CLB
EOFE A574      4976      LDA   A74
E100 F174      4977      SBB   (A74), Y
E102 857C      4978      STA   A7C
E104 A575      4979      LDA   A74+1
E106 C8        4980      INY
E107 F174      4981      SBB   (A74), Y
E109 857D      4982      STA   A7C+1
                4983      ;
                4984      ;   Branch if procedure not coded in assembly
                4985      ;
E10B A000      4986      LDY   #0
E10D B17C      4987      LDA   (A7C), Y
                4988      RDMAIN
                +      IFDEF BIG
                +      STA   RAMRDC
                +      ENDF
E10F D03D      4989      BNZ   INUBN
                4990      ;.PAGE 'Assembly language procedure'
                4991      ifndef ORIG
                4992      ;
                4993      ;
                4994      ;
                4995      LDA   IPCPTR
                4996      CLC

```

```

4997     ADC     #2
4998     STA     I PCPTR
4999     BNC     I NUBF
5000     INC     I PCPTR+1
5001     I NUBF
5002     else
E111 A586 5003     LDA     SEGNO
E113 3010 5004     BM      I NUBI
E115 0A   5005     ASL     A
E116 AA   5006     TAX
E117 38   5007     CLB
E118 BD1EBD 5008     LDA     SUCTBL, X
E11B E901 5009     SBB     #1
E11D 9D1EBD 5010     STA     SUCTBL, X
E120 B003 5011     BNB     I NUBI
E122 DE1FBD 5012     DEC     SUCTBL+1, X
5013     I NUBI
5014     endi f
5015     ;
5016     ; Subtract 2 from A7CPTR
5017     ;
5018     i fndef ORIG
5019     RDAUX
5020     JSR     S2A7C
5021     else
E125 A57C 5022     LDA     A7C
E127 38   5023     CLB
E128 E902 5024     SBB     #2
E12A 857C 5025     STA     A7C
E12C B002 5026     BNB     I NUBK
E12E C67D 5027     DEC     A7C+1
5028     I NUBK
5029     endi f
5030     ;
5031     ; Extract address of procedure
5032     ;
E130 38   5033     CLB
E131 A57C 5034     LDA     A7C
E133 F17C 5035     SBB     (A7C), Y
E135 8590 5036     STA     A90
E137 C8   5037     I NY
E138 A57D 5038     LDA     A7C+1
E13A F17C 5039     SBB     (A7C), Y
E13C 8591 5040     STA     A90+1
5041     RDMAIN
+       I FDEF BI G
+       STA     RAMRDC
+       ENDI F
5042     i fndef ORIG
5043     ;
5044     ;
5045     ;
5046     LDA     SEGNO
5047     STA     HSEGNO
5048     endi f
5049     ;
5050     ;

```

```

5051 ;
E13E AD4DE1 5052 LDA INUBL+1
E141 48 5053 PHA
E142 AD4CE1 5054 LDA INUBL
E145 48 5055 PHA
5056 ;
5057 ;
5058 ;
E146 6C9000 5059 JMP (A90)
5060 ;
5061 ;
5062 ;
5063 AE149
5064 ifndef ORIG
5065 LDA HSEGNO
5066 BM JXEM1
5067 JSR UNLSEG
5068 JXEM1
5069 endif
5070 ;
5071 ;
5072 ;
E149 4C53D2 5073 JMP EXECPC
5074 ;
5075 ;
5076 ;
E14C 48E1 5077 INUBL DW AE149-1
5078 ifndef ORIG
5079 ;
5080 ;
5081 ;
5082 HSEGNO DS 1
5083 endif
5084 ;.PAGE 'Calculate address of activation record'
5085 ;
5086 ; Alter pointer to point at size of activation record
5087 ;
E14E A57C 5088 INUBN LDA A7C
E150 38 5089 CLB
E151 E908 5090 SBB #8
E153 857C 5091 STA A7C
E155 B002 5092 BNB INUBO
E157 C67D 5093 DEC A7C+1
5094 ;
5095 ; Address = program stack pointer - size of activation record
5096 ;
E159 38 5097 INUBO CLB
E15A A000 5098 LDY #0
E15C A57A 5099 LDA A7A
5100 RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
E15E F17C 5101 SBB (A7C), Y
E160 857E 5102 STA ARPTR
E162 A57B 5103 LDA A7A+1
E164 C8 5104 INY

```

```

E165 F17C      5105      SBB      (A7C), Y
                5106      RDMAIN
                +      I FDEF BI G
                +      STA      RAMRDC
                +      ENDI F
E167 857F      5107      STA      ARPTR+1
E169 9040      5108      BB      JOOME1
                5109      ;
                5110      ; Branch if parameter size > 255
                5111      ;
E16B A003      5112      LDY      #3
                5113      RDAUX
                +      I FDEF BI G
                +      STA      RAMRDS
                +      ENDI F
E16D B17C      5114      LDA      (A7C), Y
                5115      RDMAIN
                +      I FDEF BI G
                +      STA      RAMRDC
                +      ENDI F
E16F D03A      5116      BNZ      JOOME1
                5117      ;
                5118      ; Subtract parameter size
                5119      ;
E171 88        5120      DEY
E172 A57E      5121      LDA      ARPTR
E174 38        5122      CLB
                5123      RDAUX
                +      I FDEF BI G
                +      STA      RAMRDS
                +      ENDI F
E175 F17C      5124      SBB      (A7C), Y
                5125      RDMAIN
                +      I FDEF BI G
                +      STA      RAMRDC
                +      ENDI F
E177 857E      5126      STA      ARPTR
E179 B008      5127      BNB      I NUBP
E17B A57F      5128      LDA      ARPTR+1
E17D E900      5129      SBB      #0
E17F 857F      5130      STA      ARPTR+1
E181 9028      5131      BB      JOOME1
                5132      ;. PAGE
                5133      ;
                5134      ; Subtract size of markstack
                5135      ;
E183 A57E      5136      I NUBP      LDA      ARPTR
                5137      i fndef ORIG
                5138      STA      DSTPTR
                5139      endi f
E185 38        5140      CLB
E186 E90C      5141      SBB      #12
E188 857E      5142      STA      ARPTR
E18A A57F      5143      LDA      ARPTR+1
                5144      i fndef ORIG
                5145      STA      DSTPTR+1
                5146      endi f

```

```

E18C E900      5147      SBB      #0
E18E 857F      5148      STA      ARPTR+1
E190 9019      5149      BB       JOOME1
                5150      ;
                5151      ; Branch if not out of memory
                5152      ;
E192 38        5153      CLB
E193 A57E      5154      LDA      ARPTR
E195 E55A      5155      SBB      HPPTR
E197 A57F      5156      LDA      ARPTR+1
E199 E55B      5157      SBB      HPPTR+1
E19B B011      5158      BNB      INUBU
                5159      ;
                5160      ;
                5161      ;
E19D 2090D1    5162      JSR      INXAZ
                5163      ;
                5164      ; Branch if not out of memory
                5165      ;
E1A0 38        5166      CLB
E1A1 A57E      5167      LDA      ARPTR
E1A3 E55A      5168      SBB      HPPTR
E1A5 A57F      5169      LDA      ARPTR+1
E1A7 E55B      5170      SBB      HPPTR+1
E1A9 B003      5171      BNB      INUBU
                5172      ;
                5173      ; Go to error logic
                5174      ;
E1AB 4CC3D1    5175      JOOME1   JMP      OOMERR
                5176      ;. PAGE
                5177      ;
                5178      ; Set initial displacement
                5179      ;
E1AE A002      5180      INUBU    LDY      #2
                5181      ;
                5182      ; Save markstack pointer
                5183      ;
                5184      ; ifndef ORIG
                5185      LDA      MSPTR
                5186      STA      (ARPTR), Y
                5187      INY
                5188      LDA      MSPTR+1
                5189      STA      (ARPTR), Y
                5190      INY
                5191      ;
                5192      ; Save jump table pointer
                5193      ;
                5194      LDA      JTBPTR
                5195      STA      (ARPTR), Y
                5196      INY
                5197      LDA      JTBPTR+1
                5198      STA      (ARPTR), Y
                5199      INY
                5200      ;
                5201      ; Save segment pointer
                5202      ;
                5203      LDA      SEGPTR

```

```

5204 STA (ARPTR), Y
5205 INY
5206 LDA SEGPTR+1
5207 STA (ARPTR), Y
5208 INY
5209 ;
5210 ; Save adjusted P-code pointer
5211 ;
5212 CLC
5213 LDA IPCPTR
5214 ADC #2
5215 STA (ARPTR), Y
5216 INY
5217 LDA IPCPTR+1
5218 ADC #0
5219 STA (ARPTR), Y
5220 ;
5221 ; Save markstack, jump table, segment and P-code pointers
5222 ;
5223 else
E1B0 B95000 5224 INUBW LDA MSPTR-2, Y
E1B3 917E 5225 STA (ARPTR), Y
E1B5 C8 5226 INY
E1B6 C00A 5227 CPY #10
E1B8 DOF6 5228 BNE INUBW
5229 ;
5230 ; Calculate address of parameter storage area
5231 ;
E1BA 18 5232 CLC
E1BB A57E 5233 LDA ARPTR
E1BD 690C 5234 ADC #12
E1BF 856A 5235 STA DSTPTR
E1C1 A57F 5236 LDA ARPTR+1
E1C3 6900 5237 ADC #0
E1C5 856B 5238 STA DSTPTR+1
5239 endif
5240 ;
5241 ; Move size of parameters to 'X' register
5242 ;
5243 RDAUX
+ I FDEF BIG
+ STA RAMRDS
+ ENDF
E1C7 A002 5244 LDY #2
E1C9 B17C 5245 LDA (A7C), Y
5246 RDMAIN
+ I FDEF BIG
+ STA RAMRDC
+ ENDF
E1CB A000 5247 LDY #0
E1CD AA 5248 TAX
5249 ;
5250 ; Pull parameters from stack and save them
5251 ;
E1CE 4CD6E1 5252 JMP INUBY
E1D1 68 5253 INUBX PLA
E1D2 916A 5254 STA (DSTPTR), Y

```



```

E1D4 C8      5255    INY
E1D5 CA      5256    DEX
E1D6 DOF9    5257    INUBY    BNZ    INUBX
              5258    ;
              5259    ;   Save stack pointer
              5260    ;
E1D8 BA      5261    TSX
E1D9 8A      5262    TXA
E1DA A00A    5263    LDY    #10
E1DC 917E    5264    STA    (ARPTR), Y
              5265    ;
              5266    ;
              5267    ;
E1DE A57C    5268    LDA    A7C
E1E0 18      5269    CLC
E1E1 6906    5270    ADC    #6
E1E3 8574    5271    STA    A74
E1E5 A57D    5272    LDA    A7C+1
E1E7 6900    5273    ADC    #0
E1E9 8575    5274    STA    A74+1
              5275    ;
              5276    ;   Set P-code start address
              5277    ;
E1EB A000    5278    LDY    #0
E1ED 38      5279    CLB
              5280    RDAUX
              +    IFDEF BIG
              +    STA    RAMRDS
              +    ENDF
E1EE A574    5281    LDA    A74
E1F0 F174    5282    SBB    (A74), Y
E1F2 8558    5283    STA    IPCPTR
E1F4 A575    5284    LDA    A74+1
E1F6 C8      5285    INY
E1F7 F174    5286    SBB    (A74), Y
E1F9 8559    5287    STA    IPCPTR+1
              5288    RDMAIN
              +    IFDEF BIG
              +    STA    RAMRDC
              +    ENDF
              5289    ;
              5290    ;   Set jump table pointer
              5291    ;
E1FB A57C    5292    LDA    A7C
E1FD 18      5293    CLC
E1FE 6908    5294    ADC    #8
E200 8554    5295    STA    JTBPTR
              5296    ifdef ORIG
E202 8DEEBD  5297    STA    SYCJTB
              5298    endif
E205 A57D    5299    LDA    A7C+1
E207 6900    5300    ADC    #0
E209 8555    5301    STA    JTBPTR+1
              5302    ifdef ORIG
E20B 8DEFBD  5303    STA    SYCJTB+1
              5304    endif
              5305    ;

```

```

5306 ; Set markstack address
5307 ;
E20E A57E 5308 LDA ARPTR
E210 8552 5309 STA MSPTR
E212 8DECBD 5310 STA SYCMS
E215 A57F 5311 LDA ARPTR+1
E217 8553 5312 STA MSPTR+1
E219 8DEDBD 5313 STA SYCMS+1
5314 ;
5315 ;
5316 ;
E21C A57E 5317 LDA ARPTR
E21E 38 5318 CLB
E21F E902 5319 SBB #PROCV
E221 8576 5320 STA A76
E223 A57F 5321 LDA ARPTR+1
E225 E900 5322 SBB #0
E227 8577 5323 STA A76+1
5324 ;
5325 ;
5326 ;
E229 A000 5327 LDY #0
5328 ifdef BIG
5329 ;
5330 ;
5331 ;
5332 LDA BZ5E
5333 STA (A76), Y
5334 INY
5335 LDA BZ5E+1
5336 STA (A76), Y
5337 INY
5338 endif
5339 ;
5340 ;
5341 ;
E22B A55C 5342 LDA PSKPTR
E22D 9176 5343 STA (A76), Y
E22F C8 5344 INY
E230 A55D 5345 LDA PSKPTR+1
E232 9176 5346 STA (A76), Y
5347 ;
5348 ;
5349 ;
E234 A576 5350 LDA A76
E236 855C 5351 STA PSKPTR
E238 A577 5352 LDA A76+1
E23A 855D 5353 STA PSKPTR+1
5354 ifdef BIG
5355 ;
5356 ;
5357 ;
5358 LDA #0
5359 STA BZ5E
5360 STA BZ5E+1
5361 endif
5362 ;

```

```

5363 ;
5364 ;
E23C A582 5365 LDA A82
E23E 8556 5366 STA SEGPTR
5367 ifdef ORIG
E240 8DF0BD 5368 STA SYCSEG
5369 endif
E243 A583 5370 LDA A82+1
E245 8557 5371 STA SEGPTR+1
5372 ifdef ORIG
E247 8DF1BD 5373 STA SYCSEG+1
5374 endif
5375 ;
5376 ; Increment return address
5377 ;
E24A E68E 5378 INC A8E
E24C D002 5379 BNZ INUBZ
E24E E68F 5380 INC A8E+1
5381 ;
5382 ; Return to caller
5383 ;
E250 6C8E00 5384 INUBZ JMP (A8E)
5385 ;.PAGE 'Call intermediate procedure'
5386 ;
5387 ; Extract procedure number
5388 ;
5389 CIP RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
E253 A001 5390 LDY #1
E255 B158 5391 LDA (IPCPTR), Y
5392 RDMAIN
+ IFDEF BIG
+ STA RAMRDC
+ ENDF
E257 8578 5393 STA PROCNO
5394 ;
5395 ;
5396 ;
E259 20BCEO 5397 JSR CALAXX
5398 ;
5399 ; Get lex level
5400 ;
5401 CPCOMA RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
E25C A009 5402 LDY #9
E25E B17C 5403 LDA (A7C), Y
5404 RDMAIN
+ IFDEF BIG
+ STA RAMRDC
+ ENDF
E260 8584 5405 STA MLEXLV
5406 ;
5407 ; Decrement lex level

```

```

5408 ;
E262 C684 5409 DEC MLEXLV
5410 ;
5411 ; Copy markstack pointer to temporary pointer
5412 ;
E264 A552 5413 LDA MSPTR
E266 8588 5414 STA TMSPTR
E268 A553 5415 LDA MSPTR+1
E26A 8589 5416 STA TMSPTR+1
5417 ;
5418 ;
5419 ;
E26C 4C7BE2 5420 JMP INUCC
5421 ;
5422 ; Extract address of next markstack in chain
5423 ;
E26F A002 5424 INUCB LDY #2
E271 B188 5425 LDA (TMSPTR), Y
E273 AA 5426 TAX
E274 C8 5427 INY
E275 B188 5428 LDA (TMSPTR), Y
E277 8589 5429 STA TMSPTR+1
E279 8688 5430 STX TMSPTR
5431 ; .PAGE
5432 ;
5433 ; Extract jump table pointer from markstack
5434 ;
E27B A004 5435 INUCC LDY #4
E27D B188 5436 LDA (TMSPTR), Y
E27F 858C 5437 STA TJTPTR
E281 C8 5438 INY
E282 B188 5439 LDA (TMSPTR), Y
E284 858D 5440 STA TJTPTR+1
5441 ;
5442 ; Extract lex level
5443 ;
E286 A001 5444 LDY #1
5445 RDAUX
+ I FDEF BIG
+ STA RAMRDS
+ ENDI F
E288 B18C 5446 LDA (TJTPTR), Y
5447 RDMAIN
+ I FDEF BIG
+ STA RAMRDC
+ ENDI F
5448 ;
5449 ; Branch if not at correct lex level
5450 ;
E28A C584 5451 CMP MLEXLV
E28C DOE1 5452 BNE INUCB
5453 ;
5454 ; Set static link of markstack
5455 ;
E28E A002 5456 LDY #2
E290 B188 5457 LDA (TMSPTR), Y
E292 A000 5458 LDY #0

```

```

E294 9152      5459      STA   (MSPTR), Y
E296 A003      5460      LDY   #3
E298 B188      5461      LDA   (TMSPTR), Y
E29A A001      5462      LDY   #1
E29C 9152      5463      STA   (MSPTR), Y
                5464      ;
                5465      ;   Go execute first P-code of called procedure
                5466      ;
E29E 4C53D2    5467      JMP   EXECPC
                5468      ;.PAGE 'Call local procedure'
                5469      ;
                5470      ;   Extract procedure number
                5471      ;
                5472      CLP      RDAUX
                +      I FDEF  BI G
                +      STA   RAMRDS
                +      ENDI F
E2A1 A001      5473      LDY   #1
E2A3 B158      5474      LDA   (IPCPTR), Y
                5475      RDMAI N
                +      I FDEF  BI G
                +      STA   RAMRDC
                +      ENDI F
E2A5 8578      5476      STA   PROCNO
                5477      ;
                5478      ;
                5479      ;
E2A7 20BCE0    5480      JSR   CALAXX
                5481      ;
                5482      ;   Set static link of markstack to old markstack pointer
                5483      ;
E2AA A002      5484      LDY   #2
E2AC B17E      5485      LDA   (ARPTR), Y
E2AE A000      5486      LDY   #0
E2B0 917E      5487      STA   (ARPTR), Y
E2B2 A003      5488      LDY   #3
E2B4 B17E      5489      LDA   (ARPTR), Y
E2B6 A001      5490      LDY   #1
E2B8 917E      5491      STA   (ARPTR), Y
                5492      ;
                5493      ;   Go execute first P-code of called procedure
                5494      ;
E2BA 4C53D2    5495      JMP   EXECPC
                5496      ;.PAGE 'Call global procedure'
                5497      ;
                5498      ;   Extract procedure number
                5499      ;
                5500      CGP      RDAUX
                +      I FDEF  BI G
                +      STA   RAMRDS
                +      ENDI F
E2BD A001      5501      LDY   #1
E2BF B158      5502      LDA   (IPCPTR), Y
                5503      RDMAI N
                +      I FDEF  BI G
                +      STA   RAMRDC
                +      ENDI F

```

```

E2C1 8578      5504      STA   PROCNO
                5505      ;
                5506      ;
                5507      ;
E2C3 20BCE0    5508      JSR   CALAXX
                5509      ;
                5510      ; Set static link of markstack to BASE
                5511      ;
E2C6 A000      5512      LDY   #0
E2C8 A550      5513      LDA   BASPTR
E2CA 917E      5514      STA   (ARPTR), Y
E2CC A551      5515      LDA   BASPTR+1
E2CE C8        5516      INY
E2CF 917E      5517      STA   (ARPTR), Y
                5518      ;
                5519      ; Go execute first P-code of called procedure
                5520      ;
E2D1 4C53D2    5521      JMP   EXECPC
                5522      ;.PAGE 'Call external or base procedure'
                5523      ;
                5524      ; Extract external procedure number
                5525      ;
                5526      CXP      RDAUX
                +      I FDEF  BIG
                +      STA   RAMRDS
                +      ENDI F
E2D4 A002      5527      LDY   #2
E2D6 B158      5528      LDA   (IPCPTR), Y
                5529      RDMAIN
                +      I FDEF  BIG
                +      STA   RAMRDC
                +      ENDI F
E2D8 8578      5530      STA   PROCNO
                5531      ;
                5532      ; Extract segment number
                5533      ;
                5534      RDAUX
                +      I FDEF  BIG
                +      STA   RAMRDS
                +      ENDI F
E2DA A001      5535      LDY   #1
E2DC B158      5536      LDA   (IPCPTR), Y
                5537      RDMAIN
                +      I FDEF  BIG
                +      STA   RAMRDC
                +      ENDI F
E2DE 8586      5538      STA   SEGNO
                5539      ;
                5540      ; Branch if segment number is zero
                5541      ;
E2E0 F003      5542      BZ    INUCD
                5543      ;
                5544      ; Load segment if not already in memory
                5545      ;
E2E2 20A5E4    5546      JSR   LODSEG
                5547      ;
                5548      ; Adjust PCode pointer for presence of segment number

```

```

5549 ;
E2E5 E658 5550 INUCD INC IPCPTR
E2E7 D002 5551 BNZ INUCE
E2E9 E659 5552 INC IPCPTR+1
5553 ;
5554 ;
5555 ;
E2EB 20A1E0 5556 INUCE JSR CALLX
5557 ;
5558 ; Branch if lex level is zero or minus one
5559 ;
5560 RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
E2EE A009 5561 LDY #9
E2FO B17C 5562 LDA (A7C), Y
5563 RDMAIN
+ IFDEF BIG
+ STA RAMRDC
+ ENDF
E2F2 F00E 5564 BZ CPCOMB
E2F4 300C 5565 BM CPCOMB
5566 ;
5567 ; Go use 'call intermediate procedure' logic
5568 ;
E2F6 4C5CE2 5569 JMP CPCOMA
5570
*****
5571 ;
5572 ; Extract base procedure number
5573 ;
5574 CBP RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
E2F9 A001 5575 LDY #1
E2FB B158 5576 LDA (IPCPTR), Y
5577 RDMAIN
+ IFDEF BIG
+ STA RAMRDC
+ ENDF
E2FD 8578 5578 STA PROCNO
5579 ;
5580 ;
5581 ;
E2FF 20BCE0 5582 JSR CALAXX
5583 ;.PAGE 'Common call procedure logic'
5584 ;
5585 ; Push BASE pointer onto stack
5586 ;
E302 A551 5587 CPCOMB LDA BASPTR+1
E304 48 5588 PHA
E305 A550 5589 LDA BASPTR
E307 48 5590 PHA
5591 ;
5592 ; Save stack pointer

```

```

5593 ;
E308 BA      5594   TSX
E309 8A      5595   TXA
E30A A00A    5596   LDY   #10
E30C 917E    5597   STA   (ARPTR), Y
5598 ;
5599 ;   Set static link of markstack to static link of BASE's
markstack
5600 ;
E30E A000    5601   LDY   #0
E310 B150    5602   LDA   (BASPTR), Y
E312 917E    5603   STA   (ARPTR), Y
E314 C8      5604   INY
E315 B150    5605   LDA   (BASPTR), Y
E317 917E    5606   STA   (ARPTR), Y
5607 ;
5608 ;
5609 ;
E319 A57E    5610   LDA   ARPTR
E31B 8550    5611   STA   BASPTR
E31D 8DEABD  5612   STA   SYCBAS
E320 A57F    5613   LDA   ARPTR+1
E322 8551    5614   STA   BASPTR+1
E324 8DEBBD  5615   STA   SYCBAS+1
5616 ;
5617 ;   Go execute first P-code of called procedure
5618 ;
E327 4C53D2  5619   JMP   EXECPC
5620 ;.PAGE 'Return from procedure'
5621 ;
5622 ;   Restore stack pointer
5623 ;
E32A A00A    5624   RBP   LDY   #10
E32C B152    5625   LDA   (MSPTR), Y
E32E AA      5626   TAX
E32F 9A      5627   TXS
5628 ;
5629 ;   Restore BASE
5630 ;
E330 68      5631   PLA
E331 8550    5632   STA   BASPTR
E333 8DEABD  5633   STA   SYCBAS
E336 68      5634   PLA
E337 8551    5635   STA   BASPTR+1
E339 8DEBBD  5636   STA   SYCBAS+1
5637 ;
5638 ;
5639 ;
E33C 4C45E3  5640   JMP   INUCH
5641 ;
5642 ;   Restore stack pointer
5643 ;
E33F A00A    5644   RNP   LDY   #10
E341 B152    5645   LDA   (MSPTR), Y
E343 AA      5646   TAX
E344 9A      5647   TXS
5648 ;

```



```

5649 ;
5650 ;
E345 A552 5651 INUCH LDA MSPTR
E347 38 5652 CLB
E348 E902 5653 SBB #PROCV
E34A 8574 5654 STA A74
E34C A553 5655 LDA MSPTR+1
E34E E900 5656 SBB #0
E350 8575 5657 STA A74+1
5658 ;
5659 ;
5660 ;
E352 A000 5661 LDY #0
5662 ifdef BIG
5663 ;
5664 ;
5665 ;
5666 LDA (A74), Y
5667 STA BZ5E
5668 INY
5669 LDA (A74), Y
5670 STA BZ5E+1
5671 INY
5672 endif
5673 ;
5674 ;
5675 ;
E354 B174 5676 LDA (A74), Y
E356 855C 5677 STA PSKPTR
E358 C8 5678 INY
E359 B174 5679 LDA (A74), Y
E35B 855D 5680 STA PSKPTR+1
5681 ;
5682 ; Get count of words to return
5683 ;
5684 RDAUX
+ IFDEF BIG
+ STA RAMRDS
+ ENDF
E35D A001 5685 LDY #1
E35F B158 5686 LDA (IPCPTR), Y
5687 RDMAIN
+ IFDEF BIG
+ STA RAMRDC
+ ENDF
5688 ifndef ORIG
5689 ;
5690 ; Branch if not function
5691 ;
5692 BZ INUCJ
5693 endif
5694 ;
5695 ; Initialize loop control register
5696 ;
E361 0A 5697 ASL A
E362 18 5698 CLC
E363 690B 5699 ADC #11

```

```

E365 A8      5700      TAY
              5701      ;
              5702      ; Push function result onto evaluation stack
              5703      ;
              5704      I NUCI
              5705      i f n d e f  O R I G
              5706      LDA      (MSPTR), Y
              5707      PHA
              5708      DEY
              5709      LDA      (MSPTR), Y
              5710      PHA
              5711      DEY
              5712      CPY      #11
              5713      BNE      I NUCI
              5714      e l s e
E366 C00B    5715      CPY      #11
E368 F007    5716      BE      I NUCJ
E36A B152    5717      LDA      (MSPTR), Y
E36C 48      5718      PHA
E36D 88      5719      DEY
E36E 4C66E3  5720      JMP      I NUCI
              5721      e n d i f
              5722      ;
              5723      ;
              5724      ;
E371 A006    5725      I NUCJ      LDY      #6
E373 B152    5726      LDA      (MSPTR), Y
E375 8574    5727      STA      A74
E377 C8      5728      I NY
E378 B152    5729      LDA      (MSPTR), Y
E37A 8575    5730      STA      A74+1
              5731      RDAUX
              +      I F D E F  B I G
              +      STA      RAMRDS
              +      E N D I F
              5732      ;
              5733      ;
              5734      ;
E37C A000    5735      LDY      #0
E37E B156    5736      LDA      (SEGPTR), Y
E380 D174    5737      CMP      (A74), Y
              5738      R D M A I N
              +      I F D E F  B I G
              +      STA      RAMRDC
              +      E N D I F
E382 F011    5739      BE      I NUCK
              5740      ;
              5741      ; Remove segment from memory
              5742      ;
E384 20F7E5  5743      JSR      UNLSEG
              5744      ;
              5745      ;
              5746      ;
E387 A574    5747      LDA      A74
E389 8556    5748      STA      SEGPTR
              5749      i f d e f  O R I G
E38B 8DF0BD  5750      STA      SYCSEG

```

```

5751     endi f
E38E A575 5752     LDA     A74+1
E390 8557 5753     STA     SEGPTR+1
5754     ifdef  ORIG
E392 8DF1BD 5755     STA     SYCSEG+1
5756     endi f
5757     ;
5758     ;   Restore jump table pointer
5759     ;
E395 A004 5760 INUCK     LDY     #4
E397 B152 5761     LDA     (MSPTR), Y
E399 8554 5762     STA     JTBPTR
5763     ifdef  ORIG
E39B 8DEEBD 5764     STA     SYCJTB
5765     endi f
E39E C8   5766     INY
E39F B152 5767     LDA     (MSPTR), Y
E3A1 8555 5768     STA     JTBPTR+1
5769     ifdef  ORIG
E3A3 8DEFBD 5770     STA     SYCJTB+1
5771     endi f
5772     ;
5773     ;   Restore P-code pointer
5774     ;
E3A6 A008 5775     LDY     #8
E3A8 B152 5776     LDA     (MSPTR), Y
E3AA 8558 5777     STA     IPCPTR
E3AC C8   5778     INY
E3AD B152 5779     LDA     (MSPTR), Y
E3AF 8559 5780     STA     IPCPTR+1
5781     ;
5782     ;   Restore markstack pointer
5783     ;
E3B1 A003 5784     LDY     #3
E3B3 B152 5785     LDA     (MSPTR), Y
E3B5 AA   5786     TAX
E3B6 88   5787     DEY
E3B7 B152 5788     LDA     (MSPTR), Y
E3B9 8552 5789     STA     MSPTR
E3BB 8DECB D 5790     STA     SYCMS
E3BE 8653 5791     STX     MSPTR+1
E3C0 8EEDBD 5792     STX     SYCMS+1
5793     ;
5794     ;   Go execute next P-code
5795     ;
E3C3 4C53D2 5796     JMP     EXECPC
5797     ;.PAGE 'Set "A7C" pointer'
5798     ;
5799     ;
5800     ;
E3C6 A000 5801 SETA7C  LDY     #0
E3C8 38   5802     CLB
E3C9 A590 5803     LDA     A90
E3CB F190 5804     SBB     (A90), Y
E3CD 857C 5805     STA     A7C
E3CF A591 5806     LDA     A90+1
E3D1 C8   5807     INY

```

```

E3D2 F190      5808      SBB   (A90), Y
E3D4 857D      5809      STA   A7C+1
                5810      ;
                5811      ;
                5812      ;
E3D6 60        5813      RTS
                5814      ; .PAGE 'Compute ??? total'
                5815      ;
                5816      ; Initialize loop counter
                5817      ;
E3D7 A000      5818      INUCN   LDY   #0
E3D9 B17C      5819      LDA   (A7C), Y
E3DB AA        5820      TAX
E3DC E8        5821      INX
E3DD C8        5822      INY
E3DE B17C      5823      LDA   (A7C), Y
E3E0 8584      5824      STA   LCNTH
E3E2 4C04E4    5825      JMP   INUCT
                5826      ;
                5827      ; Update total at "A88"
                5828      ;
E3E5 A000      5829      INUCR   LDY   #0
E3E7 38        5830      CLB
E3E8 A57C      5831      LDA   A7C
E3EA F17C      5832      SBB   (A7C), Y
E3EC 8574      5833      STA   NB8
E3EE C8        5834      INY
E3EF A57D      5835      LDA   A7C+1
E3F1 F17C      5836      SBB   (A7C), Y
E3F3 8575      5837      STA   NB8+1
E3F5 18        5838      CLC
E3F6 88        5839      DEY
E3F7 B174      5840      LDA   (NB8), Y
E3F9 6588      5841      ADC   A88
E3FB 9174      5842      STA   (NB8), Y
E3FD C8        5843      INY
E3FE B174      5844      LDA   (NB8), Y
E400 6589      5845      ADC   A88+1
E402 9174      5846      STA   (NB8), Y
                5847      ;
                5848      ;
                5849      ;
                5850      INUCT
                5851      ifndef ORIG
                5852      JSR   S2A7C
                5853      else
E404 A57C      5854      LDA   A7C
E406 38        5855      CLB
E407 E902      5856      SBB   #2
E409 857C      5857      STA   A7C
E40B B002      5858      BNB   INUCU
E40D C67D      5859      DEC   A7C+1
                5860      INUCU
                5861      endif
                5862      ;
                5863      ;
                5864      ;

```

```

E40F CA      5865      DEX
E410 DOD3    5866      BNZ      INUCR
E412 C684    5867      DEC      LCNTH
E414 10CF    5868      BNM      INUCR
              5869      ;
              5870      ;      Return to caller
              5871      ;
E416 60      5872      RTS
              5873      ifndef ORIG
              5874      ;.PAGE 'Some subroutines'
              5875      ;
              5876      ;
              5877      ;
              5878      NE483      LDA      ABF33
              5879      BNM      EX1
              5880      ;
              5881      ;
              5882      ;
              5883      INC      HPPTR+1
              5884      INC      HPPTR+1
              5885      ;
              5886      ;
              5887      ;
              5888      EX1      RTS
              5889

```

\*\*\*\*\*

```

              5890      ;
              5891      ;
              5892      ;
              5893      NE48D      LDA      ABF33
              5894      BNM      EX2
              5895      ;
              5896      ;
              5897      ;
              5898      DEC      HPPTR+1
              5899      DEC      HPPTR+1
              5900      ;
              5901      ;
              5902      ;
              5903      EX2      RTS
              5904

```

\*\*\*\*\*

```

              5905      ;
              5906      ;
              5907      ;
              5908      NE497      LDA      PSKPTR
              5909      SBB      HPPTR
              5910      LDA      PSKPTR+1
              5911      SBB      HPPTR+1
              5912      BNB      NEWEX
              5913      ;
              5914      ;
              5915      ;
              5916      JSR      INXAZ
              5917      ;
              5918      ;
              5919      ;

```

```

5920     CLB
5921     LDA     PSKPTR
5922     SBB     HPPTR
5923     LDA     PSKPTR+1
5924     SBB     HPPTR+1
5925     BB      JOOME2
5926     ;
5927     ;
5928     ;
5929     NEWEX   RTS
5930     ifdef  BIG
5931
*****
5932     ;
5933     ;
5934     ;
5935     NE68D   JSR     NE483
5936     ;
5937     ;
5938     ;
5939     CLB
5940     LDA     PSKPTR
5941     SBB     HPPTR
5942     STA     A7E
5943     LDA     PSKPTR+1
5944     SBB     HPPTR+1
5945     STA     A7E+1
5946     ;
5947     ;
5948     ;
5949     JSR     NE48D
5950     ;
5951     ;
5952     ;
5953     LDA     #0
5954     STA     A7E
5955     LDA     A7E+1
5956     AND     #$FE
5957     STA     A7E+1
5958     ;
5959     ;
5960     ;
5961     RTS
5962     endif
5963     ;
5964     ;
5965     ;
5966     JOOME2   JMP     OOMERR
5967     endif
5968     ;.PAGE 'Read segment into memory'
5969     ifdef  BIG
5970     ;
5971     ;
5972     ;
5973     UCVA     JMP     ND219
5974     ;
5975     ;

```

```

5976 ;
5977 ADRSPT DW SYPTBL
5978 endif
5979 ;
5980 ; Multiply passed parameter by 6
5981 ;
E417 8574 5982 RDSEG STA A74
E419 0A 5983 ASL A
E41A 18 5984 CLC
E41B 6574 5985 ADC A74
5986 ifndef BIG
E41D 0A 5987 ASL A
E41E AA 5988 TAX
5989 else
5990 STA A74
5991 LDA #0
5992 STA A74+1
5993 ASL A74
5994 ROL A74+1
5995 ;
5996 ;
5997 ;
5998 CLC
5999 LDA A74
6000 ADC ADRSPT
6001 STA A40
6002 LDA A74+1
6003 ADC ADRSPT+1
6004 STA A40+1
6005 ;
6006 ;
6007 ;
6008 LDY #0
6009 LDA (A40), Y
6010 STA A74
6011 ;
6012 ;
6013 ;
6014 LDY #4
6015 LDA (A40), Y
6016 STA A76
6017 INY
6018 LDA (A40), Y
6019 STA A76+1
6020 ;
6021 ;
6022 ;
6023 LDA BZ60
6024 CLB
6025 SBB A76
6026 STA BZ60
6027 LDA BZ60+1
6028 SBB A76+1
6029 STA BZ60+1
6030 BB UCVA
6031 ;
6032 ;

```

```

6033 ;
6034 LDA BZ60
6035 SBB BZ62
6036 LDA BZ60+1
6037 SBB BZ62+1
6038 BB UCVAA
6039 ;
6040 ;
6041 ;
6042 LDA BZ60
6043 STA A82
6044 LDA BZ60+1
6045 STA A82+1
6046 ;
6047 ;
6048 ;
6049 LDY #2
6050 LDA (A40), Y
6051 STA A88
6052 INY
6053 LDA (A40), Y
6054 STA A88+1
6055 ;
6056 ;
6057 ;
6058 LDA A76
6059 STA A80
6060 LDA A76+1
6061 STA A80+1
6062 ;
6063 ;
6064 ;
6065 JSR NE68D
6066 BNZ UCVAB
6067 ;
6068 ;
6069 ;
6070 JSR INXAZ
6071 ;
6072 ;
6073 ;
6074 JSR NE68D
6075 BZ JOOME2
6076 ;
6077 ;
6078 ;
6079 UCVAB LDA A80
6080 STA A8E
6081 LDA A80+1
6082 STA A8E+1
6083 ;
6084 ;
6085 ;
6086 LDA A7E
6087 CMP A80
6088 LDA A7E+1
6089 SBB A80+1

```



```

6090      BNB   UCVAC
6091      ;
6092      ;
6093      ;
6094      LDA   A7E
6095      STA   A8E
6096      LDA   A7E+1
6097      STA   A8E+1
6098      UCVAC
6099      endi f
6100      ;
6101      ;   Push zero word onto stack
6102      ;
E41F A900 6103      LDA   #0
E421 48   6104      PHA
E422 48   6105      PHA
6106      ;
6107      ;   Push device code onto stack
6108      ;
E423 48   6109      PHA
6110      i fndef BI G
E424 BD3EBE 6111      LDA   SYPTBL, X
6112      el se
6113      LDA   A74
6114      endi f
E427 0A   6115      ASL   A
E428 A8   6116      TAY
E429 B9AFFE 6117      LDA   DSKNUM-1, Y
E42C F009 6118      BZ    I NUCW
E42E C9FF 6119      CMP   #$FF
E430 F06F 6120      BE    I NUDD
6121      i fndef BI G
E432 BD3EBE 6122      LDA   SYPTBL, X
6123      el se
6124      LDA   A74
6125      endi f
E435 1003 6126      BNM   I NUCX
E437 B9AEFE 6127      I NUCW  LDA   DSKNUM-2, Y
E43A 48   6128      I NUCX  PHA
6129      i fndef BI G
6130      ;
6131      ;
6132      ;
E43B BD42BE 6133      LDA   SYPTBL+4, X
E43E 8574 6134      STA   PCDSI Z
E440 BD43BE 6135      LDA   SYPTBL+5, X
E443 8575 6136      STA   PCDSI Z+1
E445 A55C 6137      LDA   PSKPTR
E447 38   6138      CLB
E448 E574 6139      SBB   PCDSI Z
E44A 855C 6140      STA   PSKPTR
E44C A55D 6141      LDA   PSKPTR+1
E44E E575 6142      SBB   PCDSI Z+1
E450 855D 6143      STA   PSKPTR+1
E452 9018 6144      BB    JOOME2
6145      i fndef ORI G
6146      JSR   NE483

```

```

        6147      JSR   NE497
        6148      JSR   NE48D
        6149      else
        6150      ;
        6151      ;
        6152      ;
E454 A55C      6153      LDA   PSKPTR
E456 E55A      6154      SBB   HPPTR
E458 A55D      6155      LDA   PSKPTR+1
E45A E55B      6156      SBB   HPPTR+1
E45C B011      6157      BNB   INUCZ
        6158      ;
        6159      ;
        6160      ;
E45E 2090D1    6161      JSR   INXAZ
        6162      ;
        6163      ;
        6164      ;
E461 38        6165      CLB
E462 A55C      6166      LDA   PSKPTR
E464 E55A      6167      SBB   HPPTR
E466 A55D      6168      LDA   PSKPTR+1
E468 E55B      6169      SBB   HPPTR+1
E46A B003      6170      BNB   INUCZ
        6171      ;
        6172      ;
        6173      ;
E46C 4CC3D1    6174      JOOME2   JMP   OOMERR
        6175      ;
        6176      ;
        6177      ;
        6178      INUCZ
        6179      endi f
        6180      ;
        6181      ;   Push load address onto stack
        6182      ;
E46F A55D      6183      LDA   PSKPTR+1
E471 858F      6184      STA   A8E+1
E473 48        6185      PHA
E474 A55C      6186      LDA   PSKPTR
E476 858E      6187      STA   A8E
E478 48        6188      PHA
        6189      ;
        6190      ;   Push size of P-code onto stack
        6191      ;
E479 A575      6192      LDA   PCDSIZ+1
E47B 48        6193      PHA
E47C A574      6194      LDA   PCDSIZ
E47E 48        6195      PHA
        6196      else
        6197      ;
        6198      ;
        6199      ;
        6200      JSR   NE483
        6201      ;
        6202      ;   Push load address onto stack
        6203      ;

```

```

6204 LDA HPPTR+1
6205 PHA
6206 LDA HPPTR
6207 PHA
6208 ;
6209 ;
6210 ;
6211 JSR NE48D
6212 ;
6213 ; Push size of P-code onto stack
6214 ;
6215 LDA A8E+1
6216 PHA
6217 LDA A8E
6218 PHA
6219 endi f
6220 ;
6221 ; Push block number onto stack
6222 ;
6223 i fndef BIG
E47F BD41BE 6224 LDA SYPTBL+3, X
6225 el se
6226 LDA A88+1
6227 endi f
E482 48 6228 PHA
6229 i fndef BIG
E483 BD40BE 6230 LDA SYPTBL+2, X
6231 el se
6232 LDA A88
6233 endi f
E486 48 6234 PHA
6235 ;
6236 ;
6237 ;
E487 B9AFFE 6238 LDA DSKNUM- 1, Y
E48A F00B 6239 BZ INUDB
6240 ;
6241 ;
6242 ;
6243 i fndef BIG
E48C BD3EBE 6244 LDA SYPTBL, X
6245 el se
6246 LDA A74
6247 endi f
E48F A200 6248 LDX #0
E491 203CFF 6249 JSR JUDDRV
6250 ;
6251 ;
6252 ;
E494 4C9AE4 6253 JMP INUDC
6254 ;
6255 ;
6256 ;
E497 2012FF 6257 INUDB JSR JDREAD
6258 ;
6259 ;
6260 ;

```

```

E49A E000      6261  INUDC   CPX   #0
E49C F006      6262      BE   INUDE
E49E 8EDEBD    6263      STX   IORSLT
E4A1 4CE7D1    6264  INUDD   JMP   ERR9
              6265  INUDE
              6266 ;.PAGE 'Move logic for 128K version'
              6267  ifdef BIG
              6268 ;
              6269 ;
              6270 ;
              6271      JSR   NE483
              6272 ;
              6273 ;
              6274 ;
              6275      LDA   HPPTR
              6276      STA   SRCPTR
              6277      LDA   HPPTR+1
              6278      STA   SRCPTR+1
              6279 ;
              6280 ;
              6281 ;
              6282      JSR   NE48D
              6283 ;
              6284 ;
              6285 ;
              6286      LDA   A82
              6287      STA   DSTPTR
              6288      LDA   A82+1
              6289      STA   DSTPTR+1
              6290 ;
              6291 ;
              6292 ;
              6293      STA   RAMWRS
              6294 ;
              6295 ;
              6296 ;
              6297      LDX   A8E+1
              6298      LDY   #0
              6299  UCVAJ   CPX   #0
              6300      BE   UCVAL
              6301      DEX
              6302  UCVAK   LDA   (SRCPTR), Y
              6303      STA   (DSTPTR), Y
              6304      INY
              6305      BNZ   UCVAK
              6306      INC   DSTPTR+1
              6307      INC   SRCPTR+1
              6308      JMP   UCVAJ
              6309  UCVAL   LDX   A8E
              6310      JMP   UCVAN
              6311  UCVAL   LDA   (SRCPTR), Y
              6312      STA   (DSTPTR), Y
              6313      INY
              6314      DEX
              6315  UCVAN   BNZ   UCVAL
              6316 ;
              6317 ;

```

```

6318 ;
6319 STA RAMWRC
6320 ;
6321 ;
6322 ;
6323 LDA A80
6324 CLB
6325 SBB A8E
6326 STA A80
6327 LDA A80+1
6328 SBB A8E+1
6329 STA A80+1
6330 ;
6331 ;
6332 ;
6333 BNZ UCVAO
6334 LDA A80
6335 BZ UCVAQ
6336 ;
6337 ;
6338 ;
6339 UCVAO LDA A82
6340 CLC
6341 ADC A8E
6342 STA A82
6343 LDA A82+1
6344 ADC A8E+1
6345 STA A82+1
6346 ;
6347 ;
6348 ;
6349 LDA A8E+1
6350 LSR A
6351 CLC
6352 ADC A88
6353 STA A88
6354 BNC UCVAP
6355 INC A88+1
6356 UCVAP JMP UCVAB
6357 UCVAQ
6358 endi f
E4A4 60 6359 RTS
6360 ;.PAGE 'Determine if segment is already in memory'
6361 ;
6362 ; Double segment number
6363 ;
E4A5 8586 6364 LODSEG STA SEGNO
E4A7 0A 6365 ASL A
E4A8 A8 6366 TAY
6367 ;
6368 ; Branch if use counter for this segment is zero
6369 ;
E4A9 B91EBD 6370 LDA SUCTBL, Y
E4AC D005 6371 BNZ INUDH
E4AE B91FBD 6372 LDA SUCTBL+1, Y
E4B1 FO0B 6373 BZ INUDJ
6374 ;

```

```

        6375 ; Increment use counter for this segment
        6376 ;
E4B3 98   6377 INUDH   TYA
E4B4 AA   6378     TAX
E4B5 FE1EBD 6379     INC   SUCTBL, X
E4B8 D003  6380     BNZ   INUDI
E4BA FE1FBD 6381     INC   SUCTBL+1, X
        6382 ;
        6383 ; Return to caller
        6384 ;
E4BD 60   6385 INUDI   RTS
        6386 ;.PAGE 'Load segment into memory'
        6387 ;
        6388 ; Save segment load address
        6389 ;
E4BE A55C 6390 INUDJ   LDA   PSKPTR
E4C0 995EBD 6391     STA   SLATBL, Y
E4C3 A55D  6392     LDA   PSKPTR+1
E4C5 995FBD 6393     STA   SLATBL+1, Y
        6394 ;
        6395 ; Set use counter for this segment to 1
        6396 ;
E4C8 A901 6397     LDA   #1
E4CA 991EBD 6398     STA   SUCTBL, Y
        6399 ifndef BIG
        6400 ;
        6401 ; SEGNO * 6 -> 'X' register
        6402 ;
E4CD 98   6403     TYA
E4CE 8574 6404     STA   A74
E4D0 0A   6405     ASL   A
E4D1 18   6406     CLC
E4D2 6574 6407     ADC   A74
E4D4 AA   6408     TAX
        6409 else
        6410 ;
        6411 ; SEGNO * 6 + A(SYPTBL) -> "A40"
        6412 ;
        6413 TYA
        6414 TAX
        6415 LSR   A
        6416 STA   A74
        6417 ASL   A
        6418 CLC
        6419 ADC   A74
        6420 STA   A74
        6421 LDA   #0
        6422 STA   A74+1
        6423 ASL   A74
        6424 ROL   A74+1
        6425 CLC
        6426 LDA   A74
        6427 ADC   ADRSPT
        6428 STA   A40
        6429 LDA   A74+1
        6430 ADC   ADRSPT+1
        6431 STA   A40+1

```

```

6432   endi f
6433   ;
6434   ;
6435   ;
6436   i fndef BIG
E4D5 BD40BE 6437   LDA   SYPTBL+2, X
E4D8 D024   6438   BNZ   I NUDK
E4DA BD41BE 6439   LDA   SYPTBL+3, X
E4DD D01F   6440   BNZ   I NUDK
6441   el se
6442   LDY   #2
6443   LDA   (A40), Y
6444   BNZ   I NUDK
6445   I NY
6446   LDA   (A40), Y
6447   BNZ   I NUDK
6448   endi f
6449   ;
6450   ;
6451   ;
6452   i fndef BIG
E4DF 38     6453   CLB
E4E0 A55C   6454   LDA   PSKPTR
E4E2 FD42BE 6455   SBB   SYPTBL+4, X
E4E5 855C   6456   STA   PSKPTR
E4E7 A55D   6457   LDA   PSKPTR+1
E4E9 FD43BE 6458   SBB   SYPTBL+5, X
E4EC 855D   6459   STA   PSKPTR+1
6460   el se
6461   LDY   #4
6462   CLB
6463   LDA   PSKPTR
6464   SBB   (A40), Y
6465   STA   PSKPTR
6466   LDA   PSKPTR+1
6467   I NY
6468   SBB   (A40), Y
6469   STA   PSKPTR+1
6470   TXA
6471   TAY
6472   endi f
6473   i fndef ORIG
6474   ;
6475   ;
6476   ;
6477   BB    JOOME3
6478   JSR   NE497
6479   endi f
6480   ;
6481   ;
6482   ;
E4EE A55C   6483   LDA   PSKPTR
E4F0 38     6484   CLB
E4F1 E902   6485   SBB   #2
E4F3 999EBD 6486   STA   I UTBL, Y
E4F6 A55D   6487   LDA   PSKPTR+1
E4F8 E900   6488   SBB   #0

```

```

E4FA 999FBD    6489    STA    IUTBL+1, Y
                6490    i fdef  BIG
                6491    ;
                6492    ;
                6493    ;
                6494    LDX    SEGNO
                6495    LDA    #0
                6496    STA    SXXTBL, X
                6497    endi f
                6498    ;
                6499    ;
                6500    ;
E4FD 60        6501    RTS
                6502    i fndef  ORIG
                6503    ;
                6504    ;
                6505    ;
                6506    JOOME3   JMP    OOMERR
                6507    endi f
                6508    INUDK
                6509    ;
                6510    ;
                6511    ;
                6512    i fndef  BIG
E4FE 38        6513    CLB
E4FF A55C      6514    LDA    PSKPTR
                6515    el se
                6516    TXA
                6517    TAY
                6518    CLB
                6519    LDA    BZ60
                6520    endi f
E501 E902      6521    SBB    #2
E503 999EBD    6522    STA    IUTBL, Y
E506 8590      6523    STA    A90
                6524    i fndef  BIG
E508 A55D      6525    LDA    PSKPTR+1
                6526    el se
                6527    LDA    BZ60+1
                6528    endi f
E50A E900      6529    SBB    #0
E50C 999FBD    6530    STA    IUTBL+1, Y
E50F 8591      6531    STA    A90+1
                6532    ;
                6533    ;
                6534    ;
E511 A586      6535    LDA    SEGNO
E513 2017E4    6536    JSR    RDSEG
                6537    i fdef  BIG
                6538    ;
                6539    ;
                6540    ;
                6541    STA    RAMRDS
                6542    ;
                6543    ;
                6544    ;
                6545    LDA    #- 1

```



```

6546     STA     A80
6547     STA     A80+1
6548     ;
6549     ;
6550     ;
6551     LDA     #0
6552     STA     A88
6553     STA     A88+1
6554     ;
6555     ;
6556     ;
6557     LDA     A90
6558     STA     A74
6559     LDA     A90+1
6560     STA     A74+1
6561     endi f
6562     ;
6563     ;
6564     ;
E516 A001 6565     LDY     #1
E518 B190 6566     LDA     (A90), Y
E51A 857A 6567     STA     A7A
6568     i fdef BIG
6569     ;
6570     ;
6571     ;
6572     CLB
6573     LDA     BZ60
6574     SBB     A7A
6575     STA     A84
6576     LDA     BZ60+1
6577     SBB     #0
6578     STA     A84+1
6579     ;
6580     ;
6581     ;
6582     CLB
6583     LDA     A84
6584     SBB     BZ62
6585     LDA     A84+1
6586     SBB     BZ62+1
6587     BNB     INZAA
6588     ;
6589     ;
6590     ;
6591     JMP     UCVA A
6592     ;
6593     ;
6594     ;
6595     INZAA     LDA     BZ60
6596     STA     A84
6597     LDA     BZ60+1
6598     STA     A84+1
6599     ;
6600     ;
6601     ;
6602     LDA     A84

```

```

6603     CLB
6604     SBB     #1
6605     STA     A84
6606     BNB     INZAB
6607     DEC     A84+1
6608     ;
6609     ;
6610     ;
6611     INZAB    LDA     #>512
6612     STA     A76
6613     LDA     #<512
6614     STA     A76+1
6615     ;
6616     ;
6617     ;
6618     STA     RAMWRS
6619     INZAC
6620     endi f
6621     ;. PAGE
6622     ;
6623     ;
6624     ;
6625     i fndef  ORIG
6626     JSR     S2A90
6627     el se
E51C A590 6628     LDA     A90
E51E 38   6629     CLB
E51F E902 6630     SBB     #2
E521 8590 6631     STA     A90
E523 B002 6632     BNB     INUDM
E525 C691 6633     DEC     A90+1
6634     INUDM
6635     endi f
E527 20C6E3 6636     JSR     SETA7C
6637     i fdef  BIG
6638     ;. PAGE
6639     ;
6640     ;
6641     ;
6642     LDY     #0
6643     LDA     (A7C), Y
6644     BZ     INZAD
6645     ;
6646     ;
6647     ;
6648     LDA     SEGNO
6649     BZ     INZAG
6650     ;
6651     ;
6652     ;
6653     LDA     #$80
6654     STA     (A84), Y
6655     ;
6656     ; Unconditi onal branch
6657     ;
6658     BNZ    INZAG
6659     ;

```

```

6660 ;
6661 ;
6662 INZAD LDA A7C
6663 CMP A80
6664 LDA A7C+1
6665 SBB A80+1
6666 BNB INZAE
6667 ;
6668 ;
6669 ;
6670 LDA A7C
6671 STA A80
6672 LDA A7C+1
6673 STA A80+1
6674 ;
6675 ;
6676 ;
6677 INZAE LDA A7C
6678 CMP A88
6679 LDA A7C+1
6680 SBB A88+1
6681 BB INZAF
6682 ;
6683 ;
6684 ;
6685 LDA A7C
6686 STA A88
6687 LDA A7C+1
6688 STA A88+1
6689 ;
6690 ;
6691 ;
6692 INZAF LDY #1
6693 LDA (A7C), Y
6694 DEY
6695 STA (A84), Y
6696 ;
6697 ;
6698 ;
6699 INZAG LDA A84
6700 CLB
6701 SBB #1
6702 STA A84
6703 BNB INZAI
6704 DEC A84+1
6705 ;
6706 ;
6707 ;
6708 INZAI DEC A7A
6709 BNZ INZAC
6710 ;
6711 ;
6712 ;
6713 STA RAMWRC
6714 ;
6715 ;
6716 ;

```

```

6717   LDA   A74
6718   STA   A90
6719   LDA   A74+1
6720   STA   A90+1
6721   ;
6722   ;
6723   ;
6724   LDA   #$FF
6725   CMP   A80
6726   BNE   INZAJ
6727   CMP   A80+1
6728   BNE   INZAJ
6729   ;
6730   ;
6731   ;
6732   JMP   INZBY
6733   ;
6734   ;
6735   ;
6736   INZAJ   LDA   A80
6737   CLB
6738   SBB   #2
6739   STA   A80
6740   BNB   INZAL
6741   DEC   A80+1
6742   ;
6743   ;
6744   ;
6745   INZAL   LDY   #0
6746   CLB
6747   LDA   A80
6748   SBB   (A80), Y
6749   TAX
6750   LDA   A80+1
6751   INY
6752   SBB   (A80), Y
6753   STA   A80+1
6754   STX   A80
6755   ;
6756   ;
6757   ;
6758   LDA   A88
6759   CLC
6760   ADC   #2
6761   STA   A88
6762   BNC   INZAM
6763   INC   A88+1
6764   ;
6765   ;
6766   ;
6767   INZAM   CLB
6768   LDA   A88
6769   SBB   A80
6770   STA   A82
6771   LDA   A88+1
6772   SBB   A80+1
6773   STA   A82+1

```

```

6774 ;
6775 ;
6776 ;
6777   CLB
6778   LDA   PSKPTR
6779   SBB   A82
6780   STA   PSKPTR
6781   LDA   PSKPTR+1
6782   SBB   A82+1
6783   STA   PSKPTR+1
6784 ;
6785 ;
6786 ;
6787   JSR   NE497
6788 ;
6789 ;
6790 ;
6791   CLB
6792   LDA   A80
6793   SBB   BZ60
6794   STA   A7E
6795   LDA   A80+1
6796   SBB   BZ60+1
6797   STA   A7E+1
6798 ;
6799 ;
6800 ;
6801   CLB
6802   LDA   PSKPTR
6803   SBB   A7E
6804   STA   A8E
6805   LDA   PSKPTR+1
6806   SBB   A7E+1
6807   STA   A8E+1
6808 ;
6809 ;
6810 ;
6811   LDY   #1
6812   LDA   (A90), Y
6813   STA   A7A
6814 ;
6815 ;
6816 ;
6817   JSR   S2A90
6818 ;
6819 ;
6820 ;
6821   JSR   SETA7C
6822   endi f
6823 ; . PAGE
6824 ;
6825 ;
6826 ;
E52A A001 6827   LDY   #1
E52C B17C 6828   LDA   (A7C), Y
E52E C901 6829   CMP   #1
E530 B035 6830   BGE   I NUDQ

```

```

E532 A57C      6831   LDA   A7C
E534 38        6832   CLB
E535 E908      6833   SBB   #8
E537 857C      6834   STA   A7C
E539 B002      6835   BNB   I NUDN
E53B C67D      6836   DEC   A7C+1
E53D 38        6837   I NUDN CLB
E53E A000      6838   LDY   #0
E540 A55C      6839   LDA   PSKPTR
E542 F17C      6840   SBB   (A7C), Y
E544 858C      6841   STA   A8C
E546 C8        6842   INY
E547 A55D      6843   LDA   PSKPTR+1
E549 F17C      6844   SBB   (A7C), Y
E54B 858D      6845   STA   A8C+1
E54D C8        6846   INY
E54E 38        6847   CLB
E54F A58C      6848   LDA   A8C
E551 F17C      6849   SBB   (A7C), Y
E553 858C      6850   STA   A8C
E555 B002      6851   BNB   I NUDO
E557 C68D      6852   DEC   A8C+1
E559 A58C      6853   I NUDO LDA   A8C
E55B 38        6854   CLB
E55C E90C      6855   SBB   #12
E55E 858C      6856   STA   A8C
E560 B002      6857   BNB   I NU DP
E562 C68D      6858   DEC   A8C+1
E564 4C6FE5    6859   I NU DP JMP   I NU DR
E567 A550      6860   I NU DQ LDA   BASPTR
E569 858C      6861   STA   A8C
E56B A551      6862   LDA   BASPTR+1
E56D 858D      6863   STA   A8C+1
                6864   I NU DR
                6865   i fdef BIG
                6866   STA   RAMWRS
                6867   endi f
E56F 20C6E3    6868   JSR   SETA7C
E572 A000      6869   LDY   #0
E574 B17C      6870   LDA   (A7C), Y
                6871   i fndef ORIG
                6872   BE    I NU DT
                6873   JMP   JSRUDX
                6874   I NU DT JSR   S2A7C
                6875   i fdef BIG
                6876   LDA   (A7C), Y
                6877   CLC
                6878   ADC   BZ60
                6879   STA   A7E
                6880   INY
                6881   LDA   (A7C), Y
                6882   ADC   BZ60+1
                6883   STA   A7E+1
                6884   ;
                6885   ;
                6886   ;
                6887   DEY

```

```

6888      CLB
6889      LDA      A7E
6890      SBB      A8E
6891      STA      (A7C), Y
6892      STA      (A76), Y
6893      INY
6894      LDA      A7E+1
6895      SBB      A8E+1
6896      STA      (A7C), Y
6897      STA      (A76), Y
6898      DEY
6899      endi f
6900      el se
E576 D06C 6901      BNZ      S2A90
E578 A57C 6902      LDA      A7C
E57A 38    6903      CLB
E57B E902 6904      SBB      #2
E57D 857C 6905      STA      A7C
E57F B002 6906      BNB      I NU DT
E581 C67D 6907      DEC      A7C+1
6908      I NU DT
6909      endi f
E583 38    6910      CLB
E584 A57C 6911      LDA      A7C
E586 F17C 6912      SBB      (A7C), Y
E588 858A 6913      STA      A8A
E58A C8    6914      INY
E58B A57D 6915      LDA      A7C+1
E58D F17C 6916      SBB      (A7C), Y
E58F 858B 6917      STA      A8A+1
6918      i f n d e f  O R I G
6919      JSR      S2A7C
6920      el se
E591 A57C 6921      LDA      A7C
E593 38    6922      CLB
E594 E902 6923      SBB      #2
E596 857C 6924      STA      A7C
E598 B002 6925      BNB      I NU DU
E59A C67D 6926      DEC      A7C+1
6927      I NU DU
6928      endi f
E59C A005 6929      LDY      #5
E59E B17C 6930      LDA      (A7C), Y
E5A0 F014 6931      BZ      I NU DV
E5A2 0A    6932      ASL      A
E5A3 A8    6933      TAY
6934      RDMAI N
        +      I FDEF  B I G
        +      STA      RAMRDC
        +      ENDI F
E5A4 B99EBD 6935      LDA      I UTBL, Y
E5A7 18    6936      CLC
E5A8 6902 6937      ADC      #2
E5AA 8588 6938      STA      A88
E5AC B99FBD 6939      LDA      I UTBL+1, Y
E5AF 6900 6940      ADC      #0
E5B1 8589 6941      STA      A88+1

```

```

6942      RDAUX
          +      I FDEF BI G
          +      STA  RAMRDS
          +      ENDI F
6943      ;
6944      ;
6945      ;
E5B3 4CBEE5 6946      JMP  I NUDW
6947      ;
6948      ;
6949      ;
E5B6 A58C 6950 I NUDV  LDA  A8C
E5B8 8588 6951      STA  A88
E5BA A58D 6952      LDA  A8C+1
E5BC 8589 6953      STA  A88+1
6954      ;
6955      ;
6956      ;
E5BE 20D7E3 6957 I NUDW  JSR  I NUCN
6958      ;
6959      ;
6960      ;
E5C1 A58E 6961      LDA  A8E
E5C3 8588 6962      STA  A88
E5C5 A58F 6963      LDA  A8E+1
E5C7 8589 6964      STA  A88+1
6965      ;
6966      ;
6967      ;
E5C9 20D7E3 6968      JSR  I NUCN
6969      ;
6970      ;
6971      ;
E5CC A58A 6972      LDA  A8A
E5CE 8588 6973      STA  A88
E5D0 A58B 6974      LDA  A8A+1
E5D2 8589 6975      STA  A88+1
6976      ;
6977      ;
6978      ;
E5D4 20D7E3 6979      JSR  I NUCN
6980      ;
6981      ;
6982      ;
E5D7 ADADD1 6983      LDA  I NXBC
E5DA 8588 6984      STA  A88
E5DC ADAED1 6985      LDA  I NXBC+1
E5DF 8589 6986      STA  A88+1
6987      ;
6988      ;
6989      ;
E5E1 20D7E3 6990      JSR  I NUCN
6991      ;
6992      ;
6993      ;
6994      i fndef ORIG
6995      JSRUDX JSR  S2A90

```



```

6996   ifdef BIG
6997     LDA   A76
6998     CLC
6999     ADC   #2
7000     STA   A76
7001     BNC   INZAX
7002     INC   A76+1
7003   INZAX
7004   endif
7005   else
E5E4 A590 7006 S2A90   LDA   A90
E5E6 38   7007     CLB
E5E7 E902 7008     SBB   #2
E5E9 8590 7009     STA   A90
E5EB B002 7010     BNB   INUDY
E5ED C691 7011     DEC   A90+1
7012   INUDY
7013   endif
E5EF C67A 7014     DEC   A7A
E5F1 F003 7015     BZ   INUDZ
E5F3 4C6FE5 7016     JMP   INUDR
7017   INUDZ
7018   ;.PAGE 'Move logic for 128K version'
7019   ifdef BIG
7020   ;
7021   ;
7022   ;
7023     STA   RAMWRC
7024   ;
7025   ; Initialize source pointer
7026   ;
7027     LDA   A80
7028     STA   SRCPTR
7029     LDA   A80+1
7030     STA   SRCPTR+1
7031   ;
7032   ; Initialize destination pointer
7033   ;
7034     LDA   PSKPTR
7035     STA   DSTPTR
7036     LDA   PSKPTR+1
7037     STA   DSTPTR+1
7038   ;
7039   ; Copy
7040   ;
7041     LDX   A82+1
7042     LDY   #0
7043   INZAZ   CPX   #0
7044     BE   INZBB
7045     DEX
7046   INZBA   LDA   (SRCPTR), Y
7047     STA   (DSTPTR), Y
7048     INY
7049     BNZ   INZBA
7050     INC   DSTPTR+1
7051     INC   SRCPTR+1
7052     JMP   INZAZ

```

```

7053 ;
7054 ;
7055 ;
7056 INZBB   LDX   A82
7057       JMP   INZBD
7058 INZBC   LDA   (SRCPTR), Y
7059       STA   (DSTPTR), Y
7060       INY
7061       DEX
7062 INZBD   BNZ   INZBC
7063 ;
7064 ;
7065 ;
7066       LDA   A74
7067       STA   A90
7068       LDA   A74+1
7069       STA   A90+1
7070 ;
7071 ;
7072 ;
7073       LDY   #1
7074       LDA   (A90), Y
7075       STA   A7A
7076 ;
7077 ;
7078 ;
7079       LDA   A90
7080       CLB
7081       SBB   A7A
7082       STA   A90
7083       BNB   INZBE
7084       DEC   A90+1
7085 ;
7086 ;
7087 ;
7088 INZBE   LDA   A90
7089       CLB
7090       SBB   A7A
7091       STA   A90
7092       BNB   INZBF
7093       DEC   A90+1
7094 ;
7095 ;
7096 ;
7097 INZBF   LDA   A90
7098       STA   A7C
7099       LDA   A90+1
7100       STA   A7C+1
7101 ;
7102 ;
7103 ;
7104       STA   RAMWRS
7105 ;. PAGE
7106 ;
7107 ;
7108 ;
7109 INZBG   LDA   A84

```

```

7110      CLC
7111      ADC      #1
7112      STA      A84
7113      BNC      INZBH
7114      INC      A84+1
7115      ;
7116      ;
7117      ;
7118      INZBH    LDA      A76
7119      CLB
7120      SBB      #2
7121      STA      A76
7122      BNB      INZBI
7123      DEC      A76+1
7124      ;
7125      ;
7126      ;
7127      INZBI    LDY      #0
7128      LDA      (A84), Y
7129      BM      INZBJ
7130      ;
7131      ;
7132      ;
7133      JSR      S2A7C
7134      ;
7135      ;
7136      ;
7137      LDY      #0
7138      LDA      #0
7139      STA      (A7C), Y
7140      LDA      (A84), Y
7141      INY
7142      STA      (A7C), Y
7143      ;
7144      ;
7145      ;
7146      CLB
7147      LDA      A90
7148      SBB      A7C
7149      STA      A7E
7150      LDA      A90+1
7151      SBB      A7C+1
7152      STA      A7E+1
7153      ;
7154      ;
7155      ;
7156      CLB
7157      LDY      #0
7158      LDA      (A90), Y
7159      SBB      A7E
7160      STA      A88
7161      INY
7162      LDA      (A90), Y
7163      SBB      A7E+1
7164      STA      A88+1
7165      ;
7166      ;

```

```

7167 ;
7168 LDY #0
7169 LDA A7E
7170 STA (A90), Y
7171 INY
7172 LDA A7E+1
7173 STA (A90), Y
7174 ;
7175 ;
7176 ;
7177 JSR S2A7C
7178 ;
7179 ;
7180 ;
7181 LDY #0
7182 LDA (A76), Y
7183 CLC
7184 ADC A88
7185 STA (A7C), Y
7186 INY
7187 LDA (A76), Y
7188 ADC A88+1
7189 STA (A7C), Y
7190 ;
7191 ;
7192 ;
7193 INZBJ LDA A90
7194 CLC
7195 ADC #2
7196 STA A90
7197 BNC INZBK
7198 INC A90+1
7199 ;
7200 ;
7201 ;
7202 INZBK DEC A7A
7203 BZ INZBL
7204 ;
7205 ;
7206 ;
7207 JMP INZBG
7208 ;
7209 ; Initialize source pointer
7210 ;
7211 INZBL LDA A80
7212 STA SRCPTR
7213 LDA A80+1
7214 STA SRCPTR+1
7215 ;
7216 ; Initialize destination pointer
7217 ;
7218 LDA A7C
7219 STA DSTPTR
7220 LDA A7C+1
7221 STA DSTPTR+1
7222 ;
7223 ;

```

```

7224 ;
7225   CLB
7226   LDA   SRCPTR
7227   SBB   BZ60
7228   STA   A82
7229   LDA   SRCPTR+1
7230   SBB   BZ60+1
7231   STA   A82+1
7232 ;
7233 ;
7234 ;
7235   CLB
7236   LDA   DSTPTR
7237   SBB   SRCPTR
7238   STA   A88
7239   LDA   DSTPTR+1
7240   SBB   SRCPTR+1
7241   STA   A88+1
7242 ;
7243 ;
7244 ;
7245   LDA   A74
7246   STA   A90
7247   LDA   A74+1
7248   STA   A90+1
7249 ;
7250 ;
7251 ;
7252   LDY   #1
7253   LDA   (A90), Y
7254   STA   A7A
7255 ;
7256 ;
7257 ;
7258 INZBM   JSR   S2A90
7259 ;
7260 ;
7261 ;
7262   LDY   #0
7263   LDA   (A84), Y
7264   BNM   INZBN
7265 ;
7266 ;
7267 ;
7268   LDY   #0
7269   CLB
7270   LDA   (A90), Y
7271   SBB   A88
7272   STA   (A90), Y
7273   INY
7274   LDA   (A90), Y
7275   SBB   A88+1
7276   STA   (A90), Y
7277 ;
7278 ;
7279 ;
7280 INZBN   LDA   A84

```

```

7281     CLB
7282     SBB     #1
7283     STA     A84
7284     BNB     INZBO
7285     DEC     A84+1
7286     ;
7287     ;
7288     ;
7289     INZBO    DEC     A7A
7290     BNZ     INZBM
7291     ;.PAGE
7292     ;
7293     ; Move page count to 'X' register
7294     ;
7295     LDX     A82+1
7296     ;
7297     ; Branch if no more pages to move
7298     ;
7299     INZBP    CPX     #0
7300     BE      INZBR
7301     ;
7302     ; Move a page
7303     ;
7304     LDY     #255
7305     DEC     SRCPTR+1
7306     DEC     DSTPTR+1
7307     DEX
7308     INZBQ    LDA     (SRCPTR), Y
7309     STA     (DSTPTR), Y
7310     DEY
7311     BNZ     INZBQ
7312     LDA     (SRCPTR), Y
7313     STA     (DSTPTR), Y
7314     ;
7315     ;
7316     ;
7317     JMP     INZBP
7318     ;
7319     ; Branch if move size a multiple of a page
7320     ;
7321     INZBR    LDY     A82
7322     BZ      INZBX
7323     ;
7324     ; Initialize source pointer
7325     ;
7326     LDA     SRCPTR
7327     CLB
7328     SBB     A82
7329     STA     SRCPTR
7330     BNB     INZBS
7331     DEC     SRCPTR+1
7332     ;
7333     ; Initialize destination pointer
7334     ;
7335     INZBS    LDA     DSTPTR
7336     CLB
7337     SBB     A82

```

```

7338     STA     DSTPTR
7339     BNB     INZBT
7340     DEC     DSTPTR+1
7341     ;
7342     ;   Decrement source pointer
7343     ;
7344     INZBT    LDA     SRCPTR
7345     CLB
7346     SBB     #1
7347     STA     SRCPTR
7348     BNB     INZBU
7349     DEC     SRCPTR+1
7350     ;
7351     ;   Decrement destination pointer
7352     ;
7353     INZBU    LDA     DSTPTR
7354     CLB
7355     SBB     #1
7356     STA     DSTPTR
7357     BNB     INZBV
7358     DEC     DSTPTR+1
7359     ;
7360     ;   Move ???
7361     ;
7362     INZBV    LDA     (SRCPTR), Y
7363     STA     (DSTPTR), Y
7364     DEY
7365     BNZ     INZBV
7366     ;. PAGE
7367     ;
7368     ;
7369     ;
7370     INZBX    LDA     BZ60
7371     CLC
7372     ADC     A88
7373     STA     BZ60
7374     LDA     BZ60+1
7375     ADC     A88+1
7376     STA     BZ60+1
7377     ;
7378     ;
7379     ;
7380     STA     RAMWRC
7381     ;
7382     ;
7383     ;
7384     INZBY    STA     RAMRDC
7385     ;
7386     ;
7387     ;
7388     LDX     SEGNO
7389     LDA     #$FF
7390     STA     SXXTBL, X
7391     endi f
7392     RTS
7393     ifndef ORIG
7394     ;. PAGE 'Subroutines for decrementing pointers'

```

E5F6 60

```

7395 ;
7396 ;
7397 ;
7398 S2A90 LDA A90
7399 CLB
7400 SBB #2
7401 STA A90
7402 BNB INUDY
7403 DEC A90+1
7404 INUDY RTS
7405 ;
7406 ;
7407 ;
7408 S2A7C LDA A7C
7409 CLB
7410 SBB #2
7411 STA A7C
7412 BNB INUCU
7413 DEC A7C+1
7414 INUCU RTS
7415 endif
7416 ;.PAGE 'Unload segment'
7417 ;
7418 ; 128K version needs segment number in 'Y' register
7419 ;
7420 UNLSEG
7421 ifdef BIG
7422 TAY
7423 endif
7424 ;
7425 ; Double segment number
7426 ;
E5F7 0A 7427 ASL A
E5F8 AA 7428 TAX
7429 ;
7430 ; Branch to subroutine exit if segment 0
7431 ;
E5F9 F020 7432 BZ INUED
7433 ;
7434 ; Decrement low order byte of use counter for this segment
7435 ;
E5FB 38 7436 CLB
E5FC BD1EBD 7437 LDA SUCTBL, X
E5FF E901 7438 SBB #1
E601 9D1EBD 7439 STA SUCTBL, X
7440 ;
7441 ; Branch if decrement to zero possible
7442 ;
E604 B004 7443 BNB INUEB
7444 ;
7445 ; Decrement high order byte
7446 ;
E606 DE1FBD 7447 DEC SUCTBL+1, X
7448 ;
7449 ; Return to caller
7450 ;
E609 60 7451 RTS

```



```

7452 ;
7453 ; Branch to subroutine exit if decremented use counter not zero
7454 ;
E60A D00F 7455 INUEB   BNZ   INUED
E60C BD1FBD 7456   LDA   SUCTBL+1, X
E60F D00A 7457   BNZ   INUED
7458   ifdef BIG
7459 ;
7460 ;
7461 ;
7462   LDA   SXXTBL, Y
7463   BZ    INUEC
7464   LDA   IUTBL, X
7465   CLC
7466   ADC   #2
7467   STA   BZ60
7468   LDA   IUTBL+1, X
7469   ADC   #0
7470   STA   BZ60+1
7471 INUEC
7472   endif
7473 ;
7474 ; Free memory by using segment load address as the stack
pointer
7475 ;
E611 BD5EBD 7476   LDA   SLATBL, X
E614 855C 7477   STA   PSKPTR
E616 BD5FBD 7478   LDA   SLATBL+1, X
E619 855D 7479   STA   PSKPTR+1
7480 ;
7481 ; Return to caller
7482 ;
E61B 60 7483 INUED   RTS
7484 ;.PAGE 'Load/Unload segment to/from memory'
7485 ;
7486 ; Load segment if not already in memory
7487 ;
E61C 68 7488 CSP21   PLA
E61D AA 7489   TAX
E61E 68 7490   PLA
E61F 8A 7491   TXA
E620 20A5E4 7492   JSR   LODSEG
7493 ;
7494 ;
7495 ;
E623 4C3BD2 7496   JMP   UPDBY2
7497 ;
7498 ; Remove segment from memory
7499 ;
E626 68 7500 CSP22   PLA
E627 AA 7501   TAX
E628 68 7502   PLA
E629 8A 7503   TXA
E62A 20F7E5 7504   JSR   UNLSEG
7505 ;
7506 ;
7507 ;

```

```

E62D 4C3BD2    7508    JMP    UPDBY2
               7509    ;.PAGE 'Call system procedure'
               7510    ;
               7511    ; Extract procedure number
               7512    ;
               7513    CSP      RDAUX
                   +    I FDEF  BIG
                   +    STA    RAMRDS
                   +    ENDI F
E630 A001     7514    LDY    #1
E632 B158     7515    LDA    (IPCPTR), Y
               7516    RDMAI N
                   +    I FDEF  BIG
                   +    STA    RAMRDC
                   +    ENDI F
               7517    ;
               7518    ; Doubled procedure number becomes low order byte of routine
address
               7519    ;
E634 0A       7520    ASL    A
               7521    ifndef ORIG
               7522    STA    CSPJMP+1
               7523    CSPJMP  JMP    (CSPTBL)
               7524    else
E635 8572     7525    STA    JMPCSP+1
E637 4C7100   7526    JMP    JMPCSP
               7527    endi f
               7528    ifdef  ORIG
               7529    ifndef IIc
               7530    ;
               7531    ;
               7532    ;
E63A 203FFF   7533    CSP7    JSR    JTREE
E63D 4C3BD2   7534    JMP    UPDBY2
               7535    ;.PAGE 'TREESEARCH'
               7536    ;
               7537    ;
               7538    ;
E640 68       7539    TREES    PLA
E641 8578     7540    STA    A78
E643 68       7541    PLA
E644 8579     7542    STA    A78+1
               7543    ;
               7544    ;
               7545    ;
E646 68       7546    PLA
E647 8574     7547    STA    A74
E649 68       7548    PLA
E64A 8575     7549    STA    A74+1
               7550    ;
               7551    ;
               7552    ;
E64C 68       7553    PLA
E64D 8576     7554    STA    A76
E64F 68       7555    PLA
E650 8577     7556    STA    A76+1
               7557    ;

```

```

7558 ;
7559 ;
E652 A208 7560 INUEG LDX #8
E654 A0FF 7561 LDY #- 1
E656 CA 7562 INUEI DEX
E657 3037 7563 BM INUEM
E659 C8 7564 INY
E65A B178 7565 LDA (A78), Y
E65C D176 7566 CMP (A76), Y
E65E F0F6 7567 BE INUEI
E660 3017 7568 BM INUEK
E662 A008 7569 LDY #8
E664 B176 7570 LDA (A76), Y
E666 AA 7571 TAX
E667 C8 7572 INY
E668 B176 7573 LDA (A76), Y
E66A C900 7574 CMP #0
E66C D004 7575 BNE INUEJ
E66E E000 7576 CPX #0
E670 F02C 7577 BE INUEO
E672 8676 7578 INUEJ STX A76
E674 8577 7579 STA A76+1
E676 4C52E6 7580 JMP INUEG
E679 A00A 7581 INUEK LDY #10
E67B B176 7582 LDA (A76), Y
E67D AA 7583 TAX
E67E C8 7584 INY
E67F B176 7585 LDA (A76), Y
E681 C900 7586 CMP #0
E683 D004 7587 BNE INUEL
E685 E000 7588 CPX #0
E687 F00E 7589 BE INUEN
E689 8676 7590 INUEL STX A76
E68B 8577 7591 STA A76+1
E68D 4C52E6 7592 JMP INUEG
7593 ;
7594 ;
7595 ;
E690 A900 7596 INUEM LDA #0
E692 48 7597 PHA
E693 48 7598 PHA
7599 ;
7600 ;
7601 ;
E694 4CA4E6 7602 JMP INUEP
7603 ;
7604 ;
7605 ;
E697 A9FF 7606 INUEN LDA #- 1
E699 48 7607 PHA
E69A 48 7608 PHA
7609 ;
7610 ;
7611 ;
E69B 4CA4E6 7612 JMP INUEP
7613 ;
7614 ;

```

```

7615 ;
E69E A900 7616 INUEO LDA #<1
E6A0 48 7617 PHA
E6A1 A901 7618 LDA #>1
E6A3 48 7619 PHA
7620 ;
7621 ;
7622 ;
E6A4 A000 7623 INUEP LDY #0
E6A6 A576 7624 LDA A76
E6A8 9174 7625 STA (A74), Y
E6AA C8 7626 INY
E6AB A577 7627 LDA A76+1
E6AD 9174 7628 STA (A74), Y
7629 ;
7630 ;
7631 ;
E6AF 4C3BD2 7632 JMP UPDBY2
7633 endi f
7634 endi f
7635 ; . PAGE ' FILLCHAR'
7636 ;
7637 ;
7638 ;
E6B2 68 7639 FILLC PLA
E6B3 AA 7640 TAX
E6B4 68 7641 PLA
7642 ;
7643 ;
7644 ;
E6B5 68 7645 PLA
E6B6 855E 7646 STA OPRND
E6B8 68 7647 PLA
E6B9 855F 7648 STA OPRND+1
7649 ;
7650 ;
7651 ;
E6BB 1007 7652 BNM INUER
7653 ;
7654 ;
7655 ;
E6BD 68 7656 PLA
E6BE 68 7657 PLA
E6BF 68 7658 PLA
E6C0 68 7659 PLA
7660 ;
7661 ;
7662 ;
E6C1 4C3BD2 7663 JMP UPDBY2
7664 ;
7665 ;
7666 ;
E6C4 68 7667 INUER PLA
E6C5 856A 7668 STA DSTPTR
E6C7 68 7669 PLA
E6C8 856B 7670 STA DSTPTR+1
7671 ;

```

```

7672 ;
7673 ;
E6CA 18      7674      CLC
E6CB 68      7675      PLA
E6CC 656A    7676      ADC      DSTPTR
E6CE 856A    7677      STA      DSTPTR
E6D0 68      7678      PLA
E6D1 656B    7679      ADC      DSTPTR+1
E6D3 856B    7680      STA      DSTPTR+1
7681 ;
7682 ;
7683 ;
E6D5 8A      7684      TXA
E6D6 A65F    7685      LDX      OPRND+1
E6D8 A000    7686      LDY      #0
E6DA E000    7687      I NUES   CPX      #0
E6DC F00B    7688      BE       I NUEU
E6DE CA      7689      DEX
E6DF 916A    7690      I NUET   STA      (DSTPTR), Y
E6E1 C8      7691      I NY
E6E2 DOFB    7692      BNZ      I NUET
E6E4 E66B    7693      INC      DSTPTR+1
E6E6 4CDAE6  7694      JMP      I NUES
E6E9 A65E    7695      I NUEU   LDX      OPRND
E6EB 4CF2E6  7696      JMP      I NUEW
E6EE 916A    7697      I NUEV   STA      (DSTPTR), Y
E6F0 C8      7698      I NY
E6F1 CA      7699      DEX
E6F2 DOFA    7700      I NUEW   BNZ      I NUEV
E6F4 4C3BD2  7701      JMP      UPDBY2
7702 ; . PAGE 'SCAN'
7703 ;
7704 ;
7705 ;
E6F7 68      7706      SCAN    PLA
E6F8 68      7707      PLA
E6F9 68      7708      PLA
E6FA 857C    7709      STA      A7C
E6FC 68      7710      PLA
E6FD 857D    7711      STA      A7C+1
E6FF 18      7712      CLC
E700 68      7713      PLA
E701 657C    7714      ADC      A7C
E703 857C    7715      STA      A7C
E705 8584    7716      STA      A84
E707 68      7717      PLA
E708 657D    7718      ADC      A7C+1
E70A 857D    7719      STA      A7C+1
E70C 8585    7720      STA      A85
E70E 68      7721      PLA
E70F 8588    7722      STA      A88
E711 68      7723      PLA
E712 68      7724      PLA
E713 858A    7725      STA      A8A
E715 68      7726      PLA
E716 68      7727      PLA
E717 855E    7728      STA      OPRND

```

E719	18	7729	CLC	
E71A	6584	7730	ADC	A84
E71C	8580	7731	STA	A80
E71E	68	7732	PLA	
E71F	855F	7733	STA	OPRND+1
E721	8586	7734	STA	A86
E723	6585	7735	ADC	A84+1
E725	8581	7736	STA	A80+1
E727	1011	7737	BNM	INUEX
E729	AA	7738	TAX	
E72A	A55E	7739	LDA	OPRND
E72C	49FF	7740	EOR	#SFF
E72E	18	7741	CLC	
E72F	6901	7742	ADC	#1
E731	855E	7743	STA	OPRND
E733	8A	7744	TXA	
E734	49FF	7745	EOR	#SFF
E736	6900	7746	ADC	#0
E738	855F	7747	STA	OPRND+1
E73A	A000	7748	INUEX	LDY #0
E73C	B17C	7749	INUEY	LDA (A7C), Y
E73E	C588	7750	CMP	A88
E740	D007	7751	BNE	INUFC
E742	A58A	7752	LDA	A8A
E744	F02E	7753	BZ	INUFG
E746	4C4DE7	7754	JMP	INUFD
E749	A58A	7755	INUFC	LDA A8A
E74B	D027	7756	BNZ	INUFG
E74D	A580	7757	INUFD	LDA A80
E74F	C57C	7758	CMP	A7C
E751	D006	7759	BNE	INUFE
E753	A581	7760	LDA	A81
E755	C57D	7761	CMP	A7C+1
E757	F01B	7762	BE	INUFG
E759	A586	7763	INUFE	LDA A86
E75B	3009	7764	BM	INUFF
E75D	E67C	7765	INC	A7C
E75F	D0DB	7766	BNZ	INUEY
E761	E67D	7767	INC	A7C+1
E763	4C3CE7	7768	JMP	INUEY
E766	A57C	7769	INUFF	LDA A7C
E768	38	7770	CLB	
E769	E901	7771	SBB	#1
E76B	857C	7772	STA	A7C
E76D	BOCD	7773	BNB	INUEY
E76F	C67D	7774	DEC	A7C+1
E771	4C3CE7	7775	JMP	INUEY
E774	A57C	7776	INUFG	LDA A7C
E776	38	7777	CLB	
E777	E584	7778	SBB	A84
E779	AA	7779	TAX	
E77A	A57D	7780	LDA	A7C+1
E77C	E585	7781	SBB	A85
E77E	48	7782	PHA	
E77F	8A	7783	TXA	
E780	48	7784	PHA	
E781	4C3BD2	7785	JMP	UPDBY2

```

7786 ; . PAGE 'EXIT'
7787 ;
7788 ;
7789 ;
E784 68 7790 EXIT PLA
E785 8578 7791 STA PROCNO
E787 68 7792 PLA
7793 ;
7794 ;
7795 ;
E788 68 7796 PLA
E789 8586 7797 STA SEGNO
E78B 68 7798 PLA
7799 ;
7800 ;
7801 ;
7802 RDAUX
+ I FDEF BIG
+ STA RAMRDS
+ ENDF
E78C A001 7803 LDY #1
E78E B154 7804 LDA (JTBPTR), Y
7805 RDMAIN
+ I FDEF BIG
+ STA RAMRDC
+ ENDF
E790 1003 7806 BNM INUFJ
7807 ;
7808 ;
7809 ;
E792 4CA0D6 7810 JMP XIT
7811 ;
7812 ;
7813 ;
E795 A554 7814 INUFJ LDA JTBPTR
E797 38 7815 CLB
E798 E904 7816 SBB #4
E79A 8574 7817 STA A74
E79C A555 7818 LDA JTBPTR+1
E79E E900 7819 SBB #0
E7A0 8575 7820 STA A74+1
7821 ;
7822 ;
7823 ;
7824 RDAUX
+ I FDEF BIG
+ STA RAMRDS
+ ENDF
E7A2 A000 7825 LDY #0
E7A4 A574 7826 LDA A74
E7A6 38 7827 CLB
E7A7 F174 7828 SBB (A74), Y
E7A9 8558 7829 STA IPCPTR
E7AB C8 7830 INY
E7AC A575 7831 LDA A74+1
E7AE F174 7832 SBB (A74), Y
E7B0 8559 7833 STA IPCPTR+1

```

```

7834 ;
7835 ;
7836 ;
E7B2 88      7837      DEY
E7B3 A578    7838      LDA      PROCNO
E7B5 D154    7839      CMP      (JTBPTR), Y
E7B7 D009    7840      BNE      INUFK
E7B9 A586    7841      LDA      SEGNO
E7BB D156    7842      CMP      (SEGPTR), Y
E7BD D003    7843      BNE      INUFK
7844 ;
7845 ;
7846 ;
E7BF 4C53D2  7847      JMP      EXECPC
7848 ;
7849 ;
7850 ;
7851 INUFK    RDMAIN
      +      IFDEF  BIG
      +      STA      RAMRDC
      +      ENDF
E7C2 A552    7852      LDA      MSPTR
E7C4 8588    7853      STA      A88
E7C6 A553    7854      LDA      MSPTR+1
E7C8 8589    7855      STA      A88+1
7856 ;
7857 ;
7858 ;
E7CA 4CD9E7  7859      JMP      INUFO
7860 ;
7861 ;
7862 ;
E7CD A002    7863 INUFM      LDY      #2
E7CF B188    7864      LDA      (A88), Y
E7D1 AA      7865      TAX
E7D2 C8      7866      INY
E7D3 B188    7867      LDA      (A88), Y
E7D5 8589    7868      STA      A88+1
E7D7 8688    7869      STX      A88
7870 ;
7871 ;
7872 ;
E7D9 A004    7873 INUFO      LDY      #4
E7DB B188    7874      LDA      (A88), Y
E7DD 8576    7875      STA      A76
E7DF C8      7876      INY
E7E0 B188    7877      LDA      (A88), Y
E7E2 8577    7878      STA      A76+1
7879      RDAUX
      +      IFDEF  BIG
      +      STA      RAMRDS
      +      ENDF
7880 ;
7881 ;
7882 ;
E7E4 A001    7883      LDY      #1
E7E6 B176    7884      LDA      (A76), Y

```



```

7885      RDMAIN
          +      I FDEF BI G
          +      STA      RAMRDC
          +      ENDI F
E7E8 1003 7886      BNM      I NUFP
          7887      ;
          7888      ;
          7889      ;
E7EA 4CBFD1 7890      JMP      ERR3
          7891      ;
          7892      ;
          7893      ;
E7ED A576 7894      I NUFP      LDA      A76
E7EF 38    7895      CLB
E7F0 E904 7896      SBB      #4
E7F2 8576 7897      STA      A76
E7F4 B002 7898      BNB      I NUFQ
E7F6 C677 7899      DEC      A76+1
          7900      ;
          7901      ;
          7902      ;
E7F8 A000 7903      I NUFQ      LDY      #0
E7FA A576 7904      LDA      A76
E7FC 38    7905      CLB
          7906      RDAUX
          +      I FDEF BI G
          +      STA      RAMRDS
          +      ENDI F
E7FD F176 7907      SBB      (A76), Y
          7908      RDMAIN
          +      I FDEF BI G
          +      STA      RAMRDC
          +      ENDI F
E7FF A008 7909      LDY      #8
E801 9188 7910      STA      (A88), Y
E803 A001 7911      LDY      #1
E805 A577 7912      LDA      A76+1
          7913      RDAUX
          +      I FDEF BI G
          +      STA      RAMRDS
          +      ENDI F
E807 F176 7914      SBB      (A76), Y
          7915      RDMAIN
          +      I FDEF BI G
          +      STA      RAMRDC
          +      ENDI F
E809 A009 7916      LDY      #9
E80B 9188 7917      STA      (A88), Y
          7918      ;
          7919      ;
          7920      ;
E80D A004 7921      LDY      #4
E80F A578 7922      LDA      PROCNO
          7923      RDAUX
          +      I FDEF BI G
          +      STA      RAMRDS
          +      ENDI F

```

```

E811 D176      7924      CMP      (A76), Y
                7925      RDMAIN
                +      I FDEF BI G
                +      STA      RAMRDC
                +      ENDI F
                7926      i fndef ORIG
                7927      BNEUFM
                7928      endi f
E813 DOB8      7929      BNE      INUFM
                7930      ;
                7931      ;
                7932      ;
E815 A006      7933      LDY      #6
E817 B188      7934      LDA      (A88), Y
E819 8574      7935      STA      A74
E81B C8        7936      I NY
E81C B188      7937      LDA      (A88), Y
E81E 8575      7938      STA      A74+1
                7939      ;
                7940      ;
                7941      ;
E820 A586      7942      LDA      SEGNO
E822 A000      7943      LDY      #0
                7944      RDAUX
                +      I FDEF BI G
                +      STA      RAMRDS
                +      ENDI F
E824 D174      7945      CMP      (A74), Y
                7946      RDMAIN
                +      I FDEF BI G
                +      STA      RAMRDC
                +      ENDI F
                7947      i fndef ORIG
                7948      BNE      BNEUFM
                7949      el se
E826 DOA5      7950      BNE      INUFM
                7951      endi f
                7952      ;
                7953      ;
                7954      ;
E828 4C53D2    7955      JMP      EXECPC
                7956      ;.PAGE 'Other op codes'
                7957      ;
                7958      ; Breakpoi nt
                7959      ;
E82B A001      7960      BPT      LDY      #1
E82D 2055D1    7961      JSR      EBI G2M
                7962      ;
                7963      ;
                7964      ;
E830 4C3BD2    7965      JMP      UPDBY2
                7966
*****
                7967      ;
                7968      ; Add 2 to pointer
                7969      ;
E833 A558      7970      CSP39      LDA      IPCPTR
    
```

```

E835 18          7971    CLC
E836 6902       7972    ADC    #2
E838 8558       7973    STA    IPCPTR
E83A 9002       7974    BNC    INUFR
E83C E659       7975    INC    IPCPTR+1
                7976    ;
                7977    ;
                7978    ;
E83E 4CE3D1     7979    INUFR    JMP    ERR8
                7980

```

\*\*\*\*\*

```

                7981    ;
                7982    ;    Ti me
                7983    ;
E841 68         7984    TIME    PLA
E842 8574       7985    STA    A74
E844 68         7986    PLA
E845 8575       7987    STA    A74+1
                7988    ;
                7989    ;
                7990    ;
E847 68         7991    PLA
E848 8576       7992    STA    A76
E84A 68         7993    PLA
E84B 8577       7994    STA    A76+1
                7995    ;
                7996    ;
                7997    ;
E84D A000       7998    LDY    #0
E84F A900       7999    LDA    #0
E851 9174       8000    STA    (A74), Y
E853 9176       8001    STA    (A76), Y
E855 C8         8002    INY
E856 9174       8003    STA    (A74), Y
E858 9176       8004    STA    (A76), Y
                8005    ;
                8006    ;
                8007    ;
E85A 4C3BD2     8008    JMP    UPDBY2
                8009

```

\*\*\*\*\*

```

                8010    ;
                8011    ;
                8012    ;
E85D A568       8013    INUFS    LDA    SRCPTR
E85F 18         8014    CLC
E860 655E       8015    ADC    OPRND
E862 8568       8016    STA    SRCPTR
E864 A569       8017    LDA    SRCPTR+1
E866 655F       8018    ADC    OPRND+1
E868 8569       8019    STA    SRCPTR+1
                8020    ;
                8021    ;
                8022    ;
E86A A56A       8023    LDA    DSTPTR
E86C 18         8024    CLC
E86D 655E       8025    ADC    OPRND

```

```

E86F 856A      8026      STA   DSTPTR
E871 A56B      8027      LDA   DSTPTR+1
E873 655F      8028      ADC   OPRND+1
E875 856B      8029      STA   DSTPTR+1
                8030      ;
                8031      ;
                8032      ;
E877 A0FF      8033      LDY   #$FF
E879 A65F      8034      LDX   OPRND+1
E87B C669      8035      INUFT DEC   SRCPTR+1
E87D C66B      8036      DEC   DSTPTR+1
E87F E000      8037      CPX   #0
E881 F00D      8038      BE    INUFV
E883 CA        8039      DEX
E884 B168      8040      INUFU LDA   (SRCPTR), Y
E886 916A      8041      STA   (DSTPTR), Y
E888 88        8042      DEY
E889 C0FF      8043      CPY   #$FF
E88B D0F7      8044      BNE   INUFU
E88D 4C7BE8    8045      JMP   INUFT
E890 A65E      8046      INUFV LDX   OPRND
E892 4C9BE8    8047      JMP   INUFX
E895 B168      8048      INUFW LDA   (SRCPTR), Y
E897 916A      8049      STA   (DSTPTR), Y
E899 88        8050      DEY
E89A CA        8051      DEX
E89B D0F8      8052      INUFX BNZ   INUFW
                8053      ;
                8054      ;
                8055      ;
E89D 4C3BD2    8056      JMP   UPDBY2
                8057      ; . PAGE 'MOVELEFT or MOVERIGHT'
                8058      ;
                8059      ;
                8060      ;
E8A0 68        8061      MOVELR PLA
E8A1 855E      8062      STA   OPRND
E8A3 68        8063      PLA
E8A4 855F      8064      STA   OPRND+1
                8065      ;
                8066      ;
                8067      ;
                8068      ifndef ORIG
E8A6 100B      8069      BM    INUFY
                8070      else
                8071      BNM   INUFY
                8072      ;
                8073      ;
                8074      ;
E8A8 68        8075      PLA
E8A9 68        8076      PLA
E8AA 68        8077      PLA
E8AB 68        8078      PLA
E8AC 68        8079      PLA
E8AD 68        8080      PLA
E8AE 68        8081      PLA
E8AF 68        8082      PLA

```

```

      8083 ;
      8084 ;
      8085 ;
E8B0 4C3BD2 8086   JMP   UPDBY2
      8087   INUFY
      8088   endi f
      8089 ;
      8090 ;
      8091 ;
E8B3 68      8092   PLA
E8B4 856A    8093   STA   DSTPTR
E8B6 68      8094   PLA
E8B7 856B    8095   STA   DSTPTR+1
      8096 ;
      8097 ;
      8098 ;
E8B9 18      8099   CLC
E8BA 68      8100   PLA
E8BB 656A    8101   ADC   DSTPTR
E8BD 856A    8102   STA   DSTPTR
E8BF 68      8103   PLA
E8C0 656B    8104   ADC   DSTPTR+1
E8C2 856B    8105   STA   DSTPTR+1
      8106 ;
      8107 ;
      8108 ;
E8C4 68      8109   PLA
E8C5 8568    8110   STA   SRCPTR
E8C7 68      8111   PLA
E8C8 8569    8112   STA   SRCPTR+1
      8113 ;
      8114 ;
      8115 ;
E8CA 18      8116   CLC
E8CB 68      8117   PLA
E8CC 6568    8118   ADC   SRCPTR
E8CE 8568    8119   STA   SRCPTR
E8D0 68      8120   PLA
E8D1 6569    8121   ADC   SRCPTR+1
E8D3 8569    8122   STA   SRCPTR+1
      8123 ;
      8124 ;
      8125 ;
      8126   RDAUX
      +   I FDEF BI G
      +   STA   RAMRDS
      +   ENDI F
E8D5 A001    8127   LDY   #1
E8D7 B158    8128   LDA   (I PCPTR), Y
      8129   RDMAIN
      +   I FDEF BI G
      +   STA   RAMRDC
      +   ENDI F
E8D9 C902    8130   CMP   #2
E8DB D080    8131   BNE   I NUFS
      8132 ;
      8133 ;

```

```

8134 ;
E8DD A65F 8135 AE8DD LDX OPRND+1
E8DF A000 8136 LDY #0
E8E1 E000 8137 INUFZ CPX #0
E8E3 F00F 8138 BE INUGB
E8E5 CA 8139 DEX
E8E6 B168 8140 INUGA LDA (SRCPTR), Y
E8E8 916A 8141 STA (DSTPTR), Y
E8EA C8 8142 INY
E8EB DOF9 8143 BNZ INUGA
E8ED E66B 8144 INC DSTPTR+1
E8EF E669 8145 INC SRCPTR+1
E8F1 4CE1E8 8146 JMP INUFZ
E8F4 A65E 8147 INUGB LDX OPRND
E8F6 4CFFE8 8148 JMP INUGD
E8F9 B168 8149 INUGC LDA (SRCPTR), Y
E8FB 916A 8150 STA (DSTPTR), Y
E8FD C8 8151 INY
E8FE CA 8152 DEX
E8FF DOF8 8153 INUGD BNZ INUGC
8154 ;
8155 ;
8156 ;
E901 4C3BD2 8157 JMP UPDBY2
8158 i f n d e f  O R I G
8159 ;
8160 ;
8161 ;
8162 INUFY PLA
8163 PLA
8164 PLA
8165 PLA
8166 PLA
8167 PLA
8168 PLA
8169 PLA
8170 ;
8171 ;
8172 ;
8173 JMP UPDBY2
8174 e n d i f
8175 ; . PAGE 'Free memory'
8176 ;
8177 ; Branch if GDIR pointer is zero
8178 ;
E904 38 8179 FRE CLB
E905 ADE7BD 8180 LDA SYCGDP+1
E908 FOOE 8181 BZ INUGE
8182 ;
8183 ; Use GDIR pointer
8184 ;
E90A A55C 8185 LDA PSKPTR
E90C EDE6BD 8186 SBB SYCGDP
E90F AA 8187 TAX
E910 A55D 8188 LDA PSKPTR+1
E912 EDE7BD 8189 SBB SYCGDP+1
8190 ;

```

```

      8191 ;
      8192 ;
E915 4C21E9 8193   JMP   INUGF
      8194 ;
      8195 ;   Use heap pointer
      8196 ;
E918 A55C   8197   INUGE   LDA   PSKPTR
E91A E55A   8198   SBB   HPPTR
E91C AA     8199   TAX
E91D A55D   8200   LDA   PSKPTR+1
E91F E55B   8201   SBB   HPPTR+1
      8202 ;
      8203 ;   Divide result by 2 and push it
      8204 ;
E921 4A     8205   INUGF   LSR   A
E922 48     8206   PHA
E923 8A     8207   TXA
E924 6A     8208   ROR   A
E925 48     8209   PHA
      8210 ;
      8211 ;
      8212 ;
E926 4C3BD2 8213   JMP   UPDBY2
      8214 ;.PAGE 'Pull real value from stack'
      8215 ;
      8216 ;   Save (return address - 1)
      8217 ;
E929 68     8218   PULLFP  PLA
E92A 8588   8219   STA   FPRETA
E92C 68     8220   PLA
E92D 8589   8221   STA   FPRETA+1
      8222 ;
      8223 ;
      8224 ;
E92F A900   8225   LDA   #0
E931 9505   8226   STA   POBASE+5, X
E933 9500   8227   STA   POBASE, X
E935 9501   8228   STA   POBASE+1, X
      8229 ;
      8230 ;
      8231 ;
E937 68     8232   PLA
E938 9504   8233   STA   POBASE+4, X
E93A 68     8234   PLA
E93B 9503   8235   STA   POBASE+3, X
E93D 68     8236   PLA
E93E 2A     8237   ROL   A
E93F 3601   8238   ROL   POBASE+1, X
E941 38     8239   SEC
E942 6A     8240   ROR   A
E943 9502   8241   STA   POBASE+2, X
E945 68     8242   PLA
E946 18     8243   CLC
E947 2A     8244   ROL   A
E948 7600   8245   ROR   POBASE, X
E94A 1501   8246   ORA   POBASE+1, X
E94C 9501   8247   STA   POBASE+1, X

```

```

      8248 ;
      8249 ; Return to caller
      8250 ;
E94E E688 8251 INC FPRETA
E950 D002 8252 BNZ INUGR
E952 E689 8253 INC FPRETA+1
E954 6C8800 8254 INUGR JMP (FPRETA)
      8255 ;.PAGE 'Push real value on stack'
      8256 ;
      8257 ; Save (return address - 1)
      8258 ;
E957 68 8259 PUSHFP PLA
E958 8588 8260 STA FPRETA
E95A 68 8261 PLA
E95B 8589 8262 STA FPRETA+1
      8263 ;
      8264 ;
      8265 ;
E95D B501 8266 LDA POBASE+1, X
E95F 18 8267 CLC
E960 6A 8268 ROR A
E961 1500 8269 ORA POBASE, X
E963 48 8270 PHA
E964 B502 8271 LDA POBASE+2, X
E966 2A 8272 ROL A
E967 7601 8273 ROR POBASE+1, X
E969 6A 8274 ROR A
E96A 48 8275 PHA
E96B B503 8276 LDA POBASE+3, X
E96D 48 8277 PHA
E96E B504 8278 LDA POBASE+4, X
E970 48 8279 PHA
      8280 ;
      8281 ; Return to caller
      8282 ;
E971 E688 8283 INC FPRETA
E973 D002 8284 BNZ INUHF
E975 E689 8285 INC FPRETA+1
E977 6C8800 8286 INUHF JMP (FPRETA)
      8287 ;.PAGE 'FP subroutines'
      8288 ;
      8289 ;
      8290 ;
E97A 901D 8291 INUHG BNC INUHN
      8292 ;
      8293 ;
      8294 ;
E97C 6682 8295 ROR FPREG3+2
E97E 6683 8296 ROR FPREG3+3
E980 6684 8297 ROR FPREG3+4
E982 6685 8298 ROR FPREG3+5
E984 9006 8299 BNC INUHM
      8300 ;
      8301 ;
      8302 ;
E986 A901 8303 LDA #$01
E988 0585 8304 ORA FPREG3+5

```



```

E98A 8585      8305      STA   FPREG3+5
               8306      ;
               8307      ;
               8308      ;
E98C E681      8309  I NUHM   INC   FPREG3+1
               8310      ;
               8311      ;
               8312      ;
E98E A9FF      8313      LDA   #$FF
E990 C581      8314      CMP   FPREG3+1
E992 D005      8315      BNE   I NUHN
               8316      ;
               8317      ;
               8318      ;
E994 68        8319      PLA
E995 68        8320      PLA
               8321      ;
               8322      ;
               8323      ;
E996 4CF3D1    8324      JMP   ERR12
               8325      ;
               8326      ;
               8327      ;
E999 60        8328  I NUHN   RTS
               8329
*****
               8330      ;
               8331      ;
               8332      ;
E99A A582      8333  I NUHP   LDA   FPREG3+2
E99C 301B      8334      BM    I NUHU
               8335      ;
               8336      ;
               8337      ;
E99E 18        8338      CLC
E99F 2685      8339      ROL   FPREG3+5
E9A1 2684      8340  I NUHR   ROL   FPREG3+4
E9A3 2683      8341      ROL   FPREG3+3
E9A5 2682      8342      ROL   FPREG3+2
E9A7 3007      8343      BM    I NUHS
               8344      ;
               8345      ;
               8346      ;
E9A9 18        8347      CLC
E9AA C681      8348      DEC   FPREG3+1
E9AC D0F3      8349      BNZ   I NUHR
               8350      ;
               8351      ;
               8352      ;
E9AE F004      8353      BZ    I NUHT
               8354      ;
               8355      ;
               8356      ;
E9B0 C681      8357  I NUHS   DEC   FPREG3+1
E9B2 D005      8358      BNZ   I NUHU
               8359      ;
               8360      ;

```

```

8361 ;
E9B4 68      8362 INUHT   PLA
E9B5 68      8363     PLA
            8364 ;
            8365 ;
            8366 ;
E9B6 4CF3D1  8367     JMP    ERR12
            8368 ;
            8369 ;
            8370 ;
E9B9 60      8371 INUHU   RTS
            8372
*****
            8373 ;
            8374 ;
            8375 ;
E9BA 18      8376 INUHW   CLC
E9BB 2685    8377     ROL   FPREG3+5
E9BD 9018    8378     BNC   INUID
E9BF E684    8379     INC   FPREG3+4
E9C1 D006    8380     BNZ   INUIA
E9C3 E683    8381     INC   FPREG3+3
E9C5 D002    8382     BNZ   INUIA
E9C7 E682    8383     INC   FPREG3+2
E9C9 A585    8384 INUIA   LDA   FPREG3+5
E9CB D004    8385     BNZ   INUI C
E9CD 4684    8386     LSR   FPREG3+4
E9CF 0684    8387     ASL   FPREG3+4
E9D1 A582    8388 INUI C   LDA   FPREG3+2
E9D3 18      8389     CLC
E9D4 D001    8390     BNZ   INUID
E9D6 38      8391     SEC
E9D7 60      8392 INUID   RTS
            8393 ; . PAGE
            8394 ;
            8395 ;
            8396 ;
E9D8 A575    8397 INUI F   LDA   FPREG1+1
E9DA 8581    8398     STA   FPREG3+1
E9DC 38      8399     SEC
E9DD E57B    8400     SBC   FPREG2+1
E9DF B01B    8401     BC    INUI J
E9E1 49FF    8402     EOR   #$FF
E9E3 AA      8403     TAX
E9E4 E8      8404     INX
E9E5 A57B    8405     LDA   FPREG2+1
E9E7 8581    8406     STA   FPREG3+1
E9E9 4676    8407 INUI H   LSR   FPREG1+2
E9EB 6677    8408     ROR   FPREG1+3
E9ED 6678    8409     ROR   FPREG1+4
E9EF 6679    8410     ROR   FPREG1+5
E9F1 9006    8411     BNC   INUI I
E9F3 A901    8412     LDA   #$01
E9F5 0579    8413     ORA   FPREG1+5
E9F7 8579    8414     STA   FPREG1+5
E9F9 CA      8415 INUI I   DEX
E9FA DOED    8416     BNZ   INUI H

```

E9FC F014	8417	I NUI J	BZ	I NUI M
E9FE AA	8418	TAX		
E9FF 467C	8419	I NUI K	LSR	FPREG2+2
EA01 667D	8420	ROR	FPREG2+3	
EA03 667E	8421	ROR	FPREG2+4	
EA05 667F	8422	ROR	FPREG2+5	
EA07 9006	8423	BNC	I NUI L	
EA09 A901	8424	LDA	#S01	
EA0B 057F	8425	ORA	FPREG2+5	
EA0D 857F	8426	STA	FPREG2+5	
EA0F CA	8427	I NUI L	DEX	
EA10 DOED	8428	BNZ	I NUI K	
EA12 60	8429	I NUI M	RTS	
	8430	; . PAGE		
	8431	;		
	8432	;		
	8433	;		
EA13 20D8E9	8434	FADD	JSR	I NUI F
EA16 18	8435	CLC		
EA17 A579	8436	LDA	FPREG1+5	
EA19 657F	8437	ADC	FPREG2+5	
EA1B 8585	8438	STA	FPREG3+5	
EA1D A578	8439	LDA	FPREG1+4	
EA1F 657E	8440	ADC	FPREG2+4	
EA21 8584	8441	STA	FPREG3+4	
EA23 A577	8442	LDA	FPREG1+3	
EA25 657D	8443	ADC	FPREG2+3	
EA27 8583	8444	STA	FPREG3+3	
EA29 A576	8445	LDA	FPREG1+2	
EA2B 657C	8446	ADC	FPREG2+2	
EA2D 8582	8447	STA	FPREG3+2	
EA2F 207AE9	8448	JSR	I NUHG	
EA32 20BAE9	8449	JSR	I NUHW	
EA35 207AE9	8450	JSR	I NUHG	
EA38 60	8451	RTS		
	8452	; . PAGE		
	8453	;		
	8454	;		
	8455	;		
EA39 20D8E9	8456	FSUB	JSR	I NUI F
EA3C 38	8457	SEC		
EA3D A57F	8458	LDA	FPREG2+5	
EA3F E579	8459	SBC	FPREG1+5	
EA41 8585	8460	STA	FPREG3+5	
EA43 A57E	8461	LDA	FPREG2+4	
EA45 E578	8462	SBC	FPREG1+4	
EA47 8584	8463	STA	FPREG3+4	
EA49 A57D	8464	LDA	FPREG2+3	
EA4B E577	8465	SBC	FPREG1+3	
EA4D 8583	8466	STA	FPREG3+3	
EA4F A57C	8467	LDA	FPREG2+2	
EA51 E576	8468	SBC	FPREG1+2	
EA53 8582	8469	STA	FPREG3+2	
EA55 A582	8470	LDA	FPREG3+2	
EA57 D00F	8471	BNZ	I NUI R	
EA59 A583	8472	LDA	FPREG3+3	
EA5B D00B	8473	BNZ	I NUI R	

EA5D	A584	8474	LDA	FPREG3+4
EA5F	D007	8475	BNZ	INUIR
EA61	A900	8476	LDA	#0
EA63	8581	8477	STA	FPREG3+1
EA65	8580	8478	STA	FPREG3
EA67	60	8479	RTS	
EA68	209AE9	8480	INUIR	JSR INUHP
EA6B	20BAE9	8481	JSR	INUHW
EA6E	207AE9	8482	JSR	INUHG
EA71	60	8483	RTS	
		8484	;. PAGE	
		8485	;	
		8486	;	
		8487	;	
EA72	A575	8488	FCMP	LDA FPREG1+1
EA74	C57B	8489	CMP	FPREG2+1
EA76	9022	8490	BL	INUIW
EA78	D024	8491	BNE	INUJB
EA7A	A576	8492	LDA	FPREG1+2
EA7C	C57C	8493	CMP	FPREG2+2
EA7E	901A	8494	BL	INUIW
EA80	D01C	8495	BNE	INUJB
EA82	A577	8496	LDA	FPREG1+3
EA84	C57D	8497	CMP	FPREG2+3
EA86	9012	8498	BL	INUIW
EA88	D014	8499	BNE	INUJB
EA8A	A578	8500	LDA	FPREG1+4
EA8C	C57E	8501	CMP	FPREG2+4
EA8E	900A	8502	BL	INUIW
EA90	D00C	8503	BNE	INUJB
EA92	A579	8504	LDA	FPREG1+5
EA94	C57F	8505	CMP	FPREG2+5
EA96	9002	8506	BL	INUIW
EA98	D004	8507	BNE	INUJB
EA9A	A57A	8508	INUIW	LDA FPREG2
EA9C	38	8509	SEC	
EA9D	60	8510	RTS	
EA9E	A675	8511	INUJB	LDX FPREG1+1
EAA0	A47B	8512	LDY	FPREG2+1
EAA2	8475	8513	STY	FPREG1+1
EAA4	867B	8514	STX	FPREG2+1
EAA6	A676	8515	LDX	FPREG1+2
EAA8	A47C	8516	LDY	FPREG2+2
EAAA	8476	8517	STY	FPREG1+2
EAAC	867C	8518	STX	FPREG2+2
EAAE	A677	8519	LDX	FPREG1+3
EAB0	A47D	8520	LDY	FPREG2+3
EAB2	8477	8521	STY	FPREG1+3
EAB4	867D	8522	STX	FPREG2+3
EAB6	A678	8523	LDX	FPREG1+4
EAB8	A47E	8524	LDY	FPREG2+4
EABA	8478	8525	STY	FPREG1+4
EABC	867E	8526	STX	FPREG2+4
EABE	A574	8527	LDA	FPREG1
EAC0	18	8528	CLC	
EAC1	60	8529	RTS	
		8530	;. PAGE	'Floating point - Add'

```

8531 ;
8532 ;
8533 ;
EAC2 A274 8534 ADR LDX #>FPREG1
EAC4 2029E9 8535 JSR PULLFP
EAC7 A27A 8536 LDX #>FPREG2
EAC9 2029E9 8537 JSR PULLFP
8538 ;
8539 ; Branch if zero
8540 ;
EACC A575 8541 LDA FPREG1+1
EACE D008 8542 BNZ INUJG
8543 ;
8544 ; Skip add logic
8545 ;
EADO A27A 8546 LDX #>FPREG2
EAD2 2057E9 8547 JSR PUSHFP
8548 ;
8549 ;
8550 ;
EAD5 4C4DD2 8551 JMP UPDBY1
8552 ;
8553 ; Branch if zero
8554 ;
EAD8 A57B 8555 INUJG LDA FPREG2+1
EADA D008 8556 BNZ INUJH
8557 ;
8558 ; Skip add logic
8559 ;
EADC A274 8560 LDX #>FPREG1
EADE 2057E9 8561 JSR PUSHFP
8562 ;
8563 ;
8564 ;
EAE1 4C4DD2 8565 JMP UPDBY1
8566 ;
8567 ;
8568 ;
EAE4 A574 8569 INUJH LDA FPREG1
EAE6 457A 8570 EOR FPREG2
EAE8 D00F 8571 BNZ INUJI
EAEA A574 8572 LDA FPREG1
EAEC 8580 8573 STA FPREG3
EAEE 2013EA 8574 JSR FADD
EAF1 A280 8575 LDX #>FPREG3
EAF3 2057E9 8576 JSR PUSHFP
EAF6 4C4DD2 8577 JMP UPDBY1
EAF9 2072EA 8578 INUJI JSR FCMP
E AFC 8580 8579 STA FPREG3
EAFE 2039EA 8580 JSR FSUB
8581 ;
8582 ; Push result
8583 ;
EB01 A280 8584 LDX #>FPREG3
EB03 2057E9 8585 JSR PUSHFP
8586 ;
8587 ;

```

```

8588 ;
EB06 4C4DD2 8589 JMP UPDBY1
8590 ;.PAGE 'Floating point - Subtract'
8591 ;
8592 ;
8593 ;
EB09 A274 8594 SBR LDX #>FPREG1
EBOB 2029E9 8595 JSR PULLFP
EBOE A27A 8596 LDX #>FPREG2
EB10 2029E9 8597 JSR PULLFP
8598 ;
8599 ;
8600 ;
EB13 A575 8601 LDA FPREG1+1
EB15 D008 8602 BNZ INUJJ
8603 ;
8604 ;
8605 ;
EB17 A27A 8606 LDX #>FPREG2
EB19 2057E9 8607 JSR PUSHFP
8608 ;
8609 ;
8610 ;
EB1C 4C4DD2 8611 JMP UPDBY1
8612 ;
8613 ;
8614 ;
EB1F A57B 8615 INUJJ LDA FPREG2+1
EB21 D00E 8616 BNZ INUJK
8617 ;
8618 ;
8619 ;
EB23 A574 8620 LDA FPREG1
EB25 4980 8621 EOR #$80
EB27 8574 8622 STA FPREG1
8623 ;
8624 ;
8625 ;
EB29 A274 8626 LDX #>FPREG1
EB2B 2057E9 8627 JSR PUSHFP
8628 ;
8629 ;
8630 ;
EB2E 4C4DD2 8631 JMP UPDBY1
8632 ;
8633 ;
8634 ;
EB31 A574 8635 INUJK LDA FPREG1
EB33 457A 8636 EOR FPREG2
EB35 FOOF 8637 BZ INUJL
8638 ;
8639 ;
8640 ;
EB37 A57A 8641 LDA FPREG2
EB39 8580 8642 STA FPREG3
EB3B 2013EA 8643 JSR FADD
8644 ;

```

```

8645 ;
8646 ;
EB3E A280 8647 LDX #>FPREG3
EB40 2057E9 8648 JSR PUSHFP
8649 ;
8650 ;
8651 ;
EB43 4C4DD2 8652 JMP UPDBY1
8653 ;
8654 ;
8655 ;
EB46 2072EA 8656 INUJL JSR FCMP
EB49 B002 8657 BC INUJM
EB4B 4980 8658 EOR #$80
EB4D 8580 8659 INUJM STA FPREG3
8660 ;
8661 ;
8662 ;
EB4F 2039EA 8663 JSR FSUB
8664 ;
8665 ;
8666 ;
EB52 A280 8667 LDX #>FPREG3
EB54 2057E9 8668 JSR PUSHFP
8669 ;
8670 ;
8671 ;
EB57 4C4DD2 8672 JMP UPDBY1
8673 ; PAGE 'Floating point - Divide'
8674 ;
8675 ; Pull divisor off stack
8676 ;
EB5A A274 8677 DVR LDX #>FPREG1
EB5C 2029E9 8678 JSR PULLFP
8679 ;
8680 ; Pull dividend off stack
8681 ;
EB5F A27A 8682 LDX #>FPREG2
EB61 2029E9 8683 JSR PULLFP
8684 ;
8685 ; Branch if divisor is not zero
8686 ;
EB64 A575 8687 LDA FPREG1+1
EB66 D003 8688 BNZ INUJN
8689 ;
8690 ;
8691 ;
EB68 4CDFD1 8692 JMP ERR6
8693 ;
8694 ; Branch if dividend is not zero
8695 ;
EB6B A57B 8696 INUJN LDA FPREG2+1
EB6D D009 8697 BNZ INUJO
8698 ;
8699 ; Push zero result on stack
8700 ;
EB6F A900 8701 LDA #0

```

```

EB71 48      8702    PHA
EB72 48      8703    PHA
EB73 48      8704    PHA
EB74 48      8705    PHA
           8706    ;
           8707    ;
           8708    ;
EB75 4C4DD2  8709    JMP    UPDBY1
           8710    ;
           8711    ;
           8712    ;
EB78 A574    8713    INUJO   LDA    FPREG1
EB7A 457A    8714    EOR    FPREG2
EB7C 8580    8715    STA    FPREG3
EB7E A57B    8716    LDA    FPREG2+1
EB80 38      8717    SEC
EB81 E575    8718    SBC    FPREG1+1
EB83 900B    8719    BNC    INUJP
EB85 18      8720    CLC
EB86 697F    8721    ADC    #$7F
EB88 B00C    8722    BC     INUJQ
EB8A C9FF    8723    CMP    #$FF
EB8C F008    8724    BE     INUJQ
EB8E 9009    8725    BL     INUJT
EB90 697F    8726    INUJP  ADC    #$7F
EB92 F002    8727    BZ     INUJQ
EB94 1003    8728    BNM    INUJT
EB96 4CF3D1  8729    INUJQ  JMP    ERR12
EB99 8581    8730    INUJT  STA    FPREG3+1
           8731    ;
           8732    ;
           8733    ;
EB9B A220    8734    LDX    #32
EB9D 38      8735    INUJV  SEC
EB9E A57F    8736    LDA    FPREG2+5
EBA0 E579    8737    SBC    FPREG1+5
EBA2 8587    8738    STA    A86+1
EBA4 A57E    8739    LDA    FPREG2+4
EBA6 E578    8740    SBC    FPREG1+4
EBA8 8586    8741    STA    A86
EBA A57D    8742    LDA    FPREG2+3
EBAC E577    8743    SBC    FPREG1+3
EBAE A8      8744    TAY
EBAF A57C    8745    LDA    FPREG2+2
EBB1 E576    8746    SBC    FPREG1+2
EBB3 900D    8747    BNC    INUJW
EBB5 857C    8748    STA    FPREG2+2
EBB7 847D    8749    STY    FPREG2+3
EBB9 A586    8750    LDA    A86
EBBB 857E    8751    STA    FPREG2+4
EBBD A587    8752    LDA    A86+1
EBBF 857F    8753    STA    FPREG2+5
EBC1 38      8754    SEC
EBC2 2685    8755    INUJW  ROL    FPREG3+5
EBC4 2684    8756    ROL    FPREG3+4
EBC6 2683    8757    ROL    FPREG3+3
EBC8 2682    8758    ROL    FPREG3+2

```



```

EBCA 4676      8759      LSR   FPREG1+2
EBCC 6677      8760      ROR   FPREG1+3
EBCE 6678      8761      ROR   FPREG1+4
EBDO 6679      8762      ROR   FPREG1+5
EBD2 CA        8763      DEX
EBD3 DOC8      8764      BNZ   INUJV
           8765      ;
           8766      ;
           8767      ;
EBD5 209AE9    8768      JSR   INUHP
EBD8 20BAE9    8769      JSR   INUHW
EBDB 207AE9    8770      JSR   INUHG
           8771      ;
           8772      ;
           8773      ;
EBDE A280      8774      LDX   #>FPREG3
EBEO 2057E9    8775      JSR   PUSHFP
           8776      ;
           8777      ;
           8778      ;
EBE3 4C4DD2    8779      JMP   UPDBY1
           8780
*****
           8781      ;
           8782      ;
           8783      ;
EBE6 A574      8784      INUJX  LDA   FPREG1
EBE8 457A      8785      EOR   FPREG2
EBEA 8580      8786      STA   FPREG3
EBEC A575      8787      LDA   FPREG1+1
EBEE 38        8788      SEC
EBEF E97F      8789      SBC   #$7F
EBF1 8575      8790      STA   FPREG1+1
EBF3 A57B      8791      LDA   FPREG2+1
EBF5 38        8792      SEC
EBF6 E97F      8793      SBC   #$7F
EBF8 B8        8794      CLV
EBF9 18        8795      CLC
EBFA 6575      8796      ADC   FPREG1+1
EBFC 5005      8797      BNV   INUKB
EBFE 68        8798      INUJZ  PLA
EBFF 68        8799      PLA
EC00 4CF3D1    8800      JMP   ERR12
EC03 18        8801      INUKB  CLC
EC04 697F      8802      ADC   #$7F
EC06 F0F6      8803      BZ    INUJZ
EC08 C9FF      8804      CMP   #$FF
EC0A F0F2      8805      BE    INUJZ
ECOC 8581      8806      STA   FPREG3+1
ECOE A900      8807      LDA   #0
EC10 8582      8808      STA   FPREG3+2
EC12 8583      8809      STA   FPREG3+3
EC14 8584      8810      STA   FPREG3+4
EC16 8585      8811      STA   FPREG3+5
EC18 A218      8812      LDX   #$18
EC1A 4676      8813      INUKC  LSR   FPREG1+2
EC1C 6677      8814      ROR   FPREG1+3

```

```

EC1E 6678      8815      ROR      FPREG1+4
EC20 9013      8816      BNC      INUKE
EC22 18        8817      CLC
EC23 A584      8818      LDA      FPREG3+4
EC25 657E      8819      ADC      FPREG2+4
EC27 8584      8820      STA      FPREG3+4
EC29 A583      8821      LDA      FPREG3+3
EC2B 657D      8822      ADC      FPREG2+3
EC2D 8583      8823      STA      FPREG3+3
EC2F A582      8824      LDA      FPREG3+2
EC31 657C      8825      ADC      FPREG2+2
EC33 8582      8826      STA      FPREG3+2
EC35 CA        8827      INUKE    DEX
EC36 F013      8828      BZ       INUKF
EC38 6682      8829      ROR      FPREG3+2
EC3A 6683      8830      ROR      FPREG3+3
EC3C 6684      8831      ROR      FPREG3+4
EC3E 6685      8832      ROR      FPREG3+5
EC40 90D8      8833      BNC      INUKC
EC42 A585      8834      LDA      FPREG3+5
EC44 0901      8835      ORA      #$01
EC46 8585      8836      STA      FPREG3+5
EC48 4C1AEC    8837      JMP      INUKC
EC4B 207AE9    8838      INUKF    JSR      INUHG
EC4E 20BAE9    8839      JSR      INUHW
EC51 207AE9    8840      JSR      INUHG
EC54 60        8841      RTS
                8842      ;.PAGE 'Floating point - Multiply'
                8843      ;
                8844      ;
                8845      ;
EC55 A274      8846      MPR      LDX      #>FPREG1
EC57 2029E9    8847      JSR      PULLFP
EC5A A27A      8848      LDX      #>FPREG2
EC5C 2029E9    8849      JSR      PULLFP
                8850      ;
                8851      ;
                8852      ;
EC5F A575      8853      LDA      FPREG1+1
EC61 F006      8854      BZ       INUKG
EC63 A57B      8855      LDA      FPREG2+1
EC65 F002      8856      BZ       INUKG
                8857      ;
                8858      ; Unconditional branch
                8859      ;
EC67 D009      8860      BNZ      INUKI
                8861      ;
                8862      ;
                8863      ;
EC69 A900      8864      INUKG    LDA      #0
EC6B 48        8865      PHA
EC6C 48        8866      PHA
EC6D 48        8867      PHA
EC6E 48        8868      PHA
                8869      ;
                8870      ;
                8871      ;

```

```

EC6F 4C4DD2      8872      JMP      UPDBY1
                  8873      ;
                  8874      ;
                  8875      ;
EC72 20E6EB      8876  I NUKI      JSR      I NUJX
                  8877      ;
                  8878      ;
                  8879      ;
EC75 A280        8880      LDX      #>FPREG3
EC77 2057E9      8881      JSR      PUSHFP
                  8882      ;
                  8883      ;
                  8884      ;
EC7A 4C4DD2      8885      JMP      UPDBY1
                  8886      ;.PAGE 'Floating point - Square'
                  8887      ;
                  8888      ; Pull value to be squared off stack
                  8889      ;
EC7D A274        8890  SQR      LDX      #>FPREG1
EC7F 2029E9      8891      JSR      PULLFP
                  8892      ;
                  8893      ;
                  8894      ;
EC82 A575        8895      LDA      FPREG1+1
EC84 D009        8896      BNZ      I NUKJ
                  8897      ;
                  8898      ;
                  8899      ;
EC86 A900        8900      LDA      #0
EC88 48          8901      PHA
EC89 48          8902      PHA
EC8A 48          8903      PHA
EC8B 48          8904      PHA
                  8905      ;
                  8906      ;
                  8907      ;
EC8C 4C4DD2      8908      JMP      UPDBY1
                  8909      ;
                  8910      ;
                  8911      ;
EC8F A574        8912  I NUKJ      LDA      FPREG1
EC91 857A        8913      STA      FPREG2
EC93 A575        8914      LDA      FPREG1+1
EC95 857B        8915      STA      FPREG2+1
EC97 A576        8916      LDA      FPREG1+2
EC99 857C        8917      STA      FPREG2+2
EC9B A577        8918      LDA      FPREG1+3
EC9D 857D        8919      STA      FPREG2+3
EC9F A578        8920      LDA      FPREG1+4
ECA1 857E        8921      STA      FPREG2+4
ECA3 A579        8922      LDA      FPREG1+5
ECA5 857F        8923      STA      FPREG2+5
                  8924      ;
                  8925      ;
                  8926      ;
ECA7 20E6EB      8927      JSR      I NUJX
                  8928      ;

```

```

      8929 ;
      8930 ;
ECAA A280 8931 LDX #>FPREG3
ECAC 2057E9 8932 JSR PUSHFP
      8933 ;
      8934 ;
      8935 ;
ECAF 4C4DD2 8936 JMP UPDBY1
      8937
*****
      8938 ;
      8939 ; Absolute value
      8940 ;
ECB2 BA 8941 ABR TSX
ECB3 E8 8942 INX
ECB4 E8 8943 INX
ECB5 E8 8944 INX
ECB6 E8 8945 INX
ECB7 1E0001 8946 ASL P1BASE, X
ECBA 5E0001 8947 LSR P1BASE, X
      8948 ;
      8949 ;
      8950 ;
ECBD 4C4DD2 8951 JMP UPDBY1
      8952
*****
      8953 ;
      8954 ; Negate real
      8955 ;
ECC0 BA 8956 NGR TSX
ECC1 BD0301 8957 LDA P1BASE+3, X
ECC4 8574 8958 STA FPREG1
ECC6 BD0401 8959 LDA P1BASE+4, X
ECC9 8576 8960 STA FPREG1+2
ECCB 2674 8961 ROL FPREG1
ECCD 2676 8962 ROL FPREG1+2
ECCF F00C 8963 BZ I NUKK
ECD1 E8 8964 INX
ECD2 E8 8965 INX
ECD3 E8 8966 INX
ECD4 E8 8967 INX
ECD5 BD0001 8968 LDA P1BASE, X
ECD8 4980 8969 EOR #$80
ECDA 9D0001 8970 STA P1BASE, X
      8971 ;
      8972 ;
      8973 ;
ECDD 4C4DD2 8974 I NUKK JMP UPDBY1
      8975 ; . PAGE
      8976 ;
      8977 ; Save (return address - 1)
      8978 ;
ECE0 68 8979 FLOAT PLA
ECE1 8588 8980 STA FPRETA
ECE3 68 8981 PLA
ECE4 8589 8982 STA FPRETA+1
      8983 ;

```

```

      8984 ;
      8985 ;
ECE6 68      8986     PLA
ECE7 8586    8987     STA   A86
ECE9 68      8988     PLA
ECEA 8587    8989     STA   A86+1
      8990 ;
      8991 ;
      8992 ;
ECEC A587    8993     LDA   A86+1
ECEE D012    8994     BNZ   I NUKN
ECFO A586    8995     LDA   A86
ECF2 D00E    8996     BNZ   I NUKN
      8997     i fdef  ORIG
      8998     i fndef  II c
      8999 ;
      9000 ;
      9001 ;
ECF4 A900    9002     LDA   #0
      9003     endi f
      9004     endi f
ECF6 8580    9005     STA   FPREG3
ECF8 8581    9006     STA   FPREG3+1
ECFA 8582    9007     STA   FPREG3+2
ECFC 8583    9008     STA   FPREG3+3
ECFE 8584    9009     STA   FPREG3+4
      9010 ;
      9011 ;   Unconditional branch
      9012 ;
ED00 F034    9013     BZ    I NUKR
      9014 ;
      9015 ;
      9016 ;
ED02 A900    9017     I NUKN   LDA   #0
ED04 8580    9018     STA   FPREG3
      9019     i fndef  ORIG
      9020 ;
      9021 ;
      9022 ;
      9023     LDX   #$8E
      9024     LDA   A86+1
      9025     BNM   I NUKQ
      9026 ;
      9027 ;
      9028 ;
      9029     LDA   #$80
      9030     STA   FPREG3
      9031     el se
      9032 ;
      9033 ;
      9034 ;
      9035     i fdef  II c
      9036     LDX   #$8E
      9037     endi f
ED06 A587    9038     LDA   A86+1
      9039     i fdef  II c
      9040     BNM   I NUKQ

```

```

ED08 1015      9041      el se
                9042      BNM   I NUKP
                9043      endi f
                9044      i fdef II c
                9045      ;
                9046      ;
                9047      ;
                9048      LDA   #$80
                9049      STA   FPREG3
                9050      endi f
                9051      endi f
                9052      ;
                9053      ;
                9054      ;
ED0A 18        9055      CLC
ED0B A586      9056      LDA   A86
ED0D 49FF      9057      EOR   #$FF
ED0F 6901      9058      ADC   #1
ED11 8586      9059      STA   A86
ED13 A587      9060      LDA   A86+1
ED15 49FF      9061      EOR   #$FF
ED17 6900      9062      ADC   #0
ED19 8587      9063      STA   A86+1
                9064      i fndef ORIG
                9065      BM    I NUKP
                9066      el se
                9067      i fndef II c
ED1B A980      9068      LDA   #$80
ED1D 8580      9069      STA   FPREG3
ED1F A28E      9070      I NUKP  LDX   #$8E
                9071      el se
                9072      BM    I NUKP
                9073      endi f
                9074      endi f
ED21 CA        9075      I NUKQ  DEX
ED22 0686      9076      ASL   A86
ED24 2687      9077      ROL   A86+1
ED26 10F9      9078      BNM   I NUKQ
                9079      i fndef ORIG
                9080      I NUKP
                9081      el se
                9082      i fdef II c
                9083      I NUKP
                9084      endi f
                9085      endi f
ED28 8681      9086      STX   FPREG3+1
ED2A A587      9087      LDA   A86+1
ED2C 8582      9088      STA   FPREG3+2
ED2E A586      9089      LDA   A86
ED30 8583      9090      STA   FPREG3+3
ED32 A900      9091      LDA   #0
ED34 8584      9092      STA   FPREG3+4
                9093      ;
                9094      ;   Return to caller
                9095      ;
ED36 E688      9096      I NUKR  INC   FPRETA
ED38 D002      9097      BNZ   I NUKS

```

```

ED3A E689      9098      INC      FPRETA+1
ED3C 6C8800    9099      INUKS      JMP      (FPRETA)
                9100      ;.PAGE 'Integer to real conversion'
                9101      ;
                9102      ; Save TOS
                9103      ;
ED3F 68        9104      FLO        PLA
ED40 8579      9105      STA        FPREG1+5
ED42 68        9106      PLA
ED43 8578      9107      STA        FPREG1+4
ED45 68        9108      PLA
ED46 8577      9109      STA        FPREG1+3
ED48 68        9110      PLA
ED49 8576      9111      STA        FPREG1+2
                9112      ;
                9113      ;
                9114      ;
ED4B 20E0EC    9115      JSR        FLOAT
ED4E A280      9116      LDX        #>FPREG3
ED50 2057E9    9117      JSR        PUSHFP
                9118      ;
                9119      ;
                9120      ;
ED53 A576      9121      LDA        FPREG1+2
ED55 48        9122      PHA
ED56 A577      9123      LDA        FPREG1+3
ED58 48        9124      PHA
ED59 A578      9125      LDA        FPREG1+4
ED5B 48        9126      PHA
ED5C A579      9127      LDA        FPREG1+5
ED5E 48        9128      PHA
                9129      ;
                9130      ;
                9131      ;
ED5F 4C4DD2    9132      JMP        UPDBY1
                9133      ;
                9134      ; Fl oat TOS
                9135      ;
ED62 20E0EC    9136      FLT        JSR        FLOAT
ED65 A280      9137      LDX        #>FPREG3
ED67 2057E9    9138      JSR        PUSHFP
ED6A 4C4DD2    9139      JMP        UPDBY1
                9140      ;.PAGE 'Real to integer subroutine'
                9141      ;
                9142      ;
                9143      ;
ED6D 38        9144      FIX        SEC
ED6E A98D      9145      LDA        #$8D
ED70 E575      9146      SBC        FPREG1+1
ED72 B005      9147      BC        INUKW
                9148      ;
                9149      ;
                9150      ;
ED74 68        9151      INUKV      PLA
ED75 68        9152      PLA
                9153      ;
                9154      ;

```

```

ED76 4CF3D1    9155 ;
                9156   JMP   ERR12
                9157 ;
                9158 ;
                9159 ;
ED79 AA        9160 I NUKW   TAX
ED7A E018      9161   CPX   #$18
ED7C 3009      9162   BM    I NUKX
                9163 ;
                9164 ;
                9165 ;
ED7E A900      9166   LDA   #0
ED80 8576      9167   STA   FPREG1+2
ED82 8577      9168   STA   FPREG1+3
ED84 8578      9169   STA   FPREG1+4
                9170 ;
                9171 ;
                9172 ;
ED86 60        9173   RTS
                9174 ;
                9175 ;
                9176 ;
ED87 4676      9177 I NUKX   LSR   FPREG1+2
ED89 6677      9178   ROR   FPREG1+3
ED8B 6678      9179   ROR   FPREG1+4
                9180 ;
                9181 ;
                9182 ;
ED8D CA        9183   DEX
ED8E 10F7      9184   BNM   I NUKX
                9185 ;
                9186 ;   Branch if truncating
                9187 ;
ED90 A586      9188   LDA   A86
ED92 F00F      9189   BZ    I NUKZ
                9190 ;
                9191 ;
                9192 ;
ED94 A578      9193   LDA   FPREG1+4
ED96 100B      9194   BNM   I NUKZ
ED98 E677      9195   INC   FPREG1+3
ED9A D007      9196   BNZ   I NUKZ
ED9C E676      9197   INC   FPREG1+2
ED9E 1003      9198   BNM   I NUKZ
                9199 ;
                9200 ;
                9201 ;
EDA0 4C74ED    9202   JMP   I NUKV
                9203 ;
                9204 ;
                9205 ;
EDA3 A574      9206 I NUKZ   LDA   FPREG1
EDA5 1013      9207   BNM   I NULD
                9208 ;
                9209 ;
                9210 ;
EDA7 A576      9211   LDA   FPREG1+2

```



```

EDA9 49FF      9212      EOR      #SFF
EDAB 8576      9213      STA      FPREG1+2
EDAD A577      9214      LDA      FPREG1+3
EDAF 49FF      9215      EOR      #SFF
EDB1 18        9216      CLC
EDB2 6901      9217      ADC      #1
EDB4 8577      9218      STA      FPREG1+3
EDB6 9002      9219      BNC      I NULD
EDB8 E676      9220      INC      FPREG1+2
                9221      ;
                9222      ;
                9223      ;
EDBA 60        9224      I NULD    RTS
                9225      ;.PAGE 'Real to integer conversion'
                9226      ;
                9227      ; Round
                9228      ;
EDBB A901      9229      RND      LDA      #1
EDBD 8586      9230      STA      A86
                9231      ;
                9232      ; Pull real value off stack
                9233      ;
EDBF A274      9234      LDX      #>FPREG1
EDC1 2029E9    9235      JSR      PULLFP
                9236      ;
                9237      ;
                9238      ;
EDC4 206DED    9239      JSR      FIX
                9240      ;
                9241      ; Push result on stack
                9242      ;
EDC7 A576      9243      LDA      FPREG1+2
EDC9 48        9244      PHA
EDCA A577      9245      LDA      FPREG1+3
EDCC 48        9246      PHA
                9247      ;
                9248      ;
                9249      ;
EDCD 4C3BD2    9250      JMP      UPDBY2
                9251
*****
                9252      ;
                9253      ; Truncate
                9254      ;
EDD0 A900      9255      TNC      LDA      #0
EDD2 8586      9256      STA      A86
                9257      ;
                9258      ; Pull real value off stack
                9259      ;
EDD4 A274      9260      LDX      #>FPREG1
EDD6 2029E9    9261      JSR      PULLFP
                9262      ;
                9263      ;
                9264      ;
EDD9 206DED    9265      JSR      FIX
                9266      ;
                9267      ; Push result on stack

```

```

9268 ;
EDDC A576 9269 LDA FPREG1+2
EDDE 48 9270 PHA
EDDF A577 9271 LDA FPREG1+3
EDE1 48 9272 PHA
9273 ;
9274 ;
9275 ;
EDE2 4C3BD2 9276 JMP UPDBY2
9277 ;.PAGE 'Power of ten'
9278 ;
9279 ; Pull integer exponent from stack
9280 ;
EDE5 68 9281 PWR10 PLA
EDE6 AA 9282 TAX
EDE7 68 9283 PLA
9284 ;
9285 ; Branch if exponent > 38
9286 ;
EDE8 D01B 9287 BNZ INULF
EDEA E027 9288 CPX #39
EDEE 1017 9289 BNM INULF
9290 ;
9291 ; Multiply exponent by 4
9292 ;
EDEE 8A 9293 TXA
EDEF 0A 9294 ASL A
EDFO 0A 9295 ASL A
EDF1 AA 9296 TAX
9297 ;
9298 ; Push result value from table
9299 ;
EDF2 BDOFEE 9300 LDA P10TBL+1, X
EDF5 48 9301 PHA
EDF6 BDOEEE 9302 LDA P10TBL, X
EDF9 48 9303 PHA
EDFA BD11EE 9304 LDA P10TBL+3, X
EDFD 48 9305 PHA
EDFE BD10EE 9306 LDA P10TBL+2, X
EEO1 48 9307 PHA
9308 ;
9309 ;
9310 ;
EEO2 4C3BD2 9311 JMP UPDBY2
9312 ;
9313 ; Push zero
9314 ;
EEO5 A900 9315 INULF LDA #0
EEO7 48 9316 PHA
EEO8 48 9317 PHA
EEO9 48 9318 PHA
EEOA 48 9319 PHA
9320 ;
9321 ;
9322 ;
EEOB 4CF3D1 9323 JMP ERR12
9324 ;

```

```

9325 ;
9326 ;
EE0E 803F0000 9327 P10TBL DB $80, $3F, $00, $00
EE12 20410000 9328 DB $20, $41, $00, $00
EE16 C8420000 9329 DB $C8, $42, $00, $00
EE1A 7A440000 9330 DB $7A, $44, $00, $00
EE1E 1C460040 9331 DB $1C, $46, $00, $40
EE22 C3470050 9332 DB $C3, $47, $00, $50
EE26 74490024 9333 DB $74, $49, $00, $24
EE2A 184B8096 9334 DB $18, $4B, $80, $96
EE2E BE4C20BC 9335 DB $BE, $4C, $20, $BC
EE32 6E4E286B 9336 DB $6E, $4E, $28, $6B
EE36 1550F902 9337 DB $15, $50, $F9, $02
EE3A BA51B743 9338 DB $BA, $51, $B7, $43
EE3E 6853A5D4 9339 DB $68, $53, $A5, $D4
EE42 1155E784 9340 DB $11, $55, $E7, $84
EE46 B55621E6 9341 DB $B5, $56, $21, $E6
EE4A 6358A95F 9342 DB $63, $58, $A9, $5F
EE4E 0E5ACA1B 9343 DB $0E, $5A, $CA, $1B
EE52 B15BBCA2 9344 DB $B1, $5B, $BC, $A2
EE56 5E5D6B0B 9345 DB $5E, $5D, $6B, $0B
EE5A 0A5F23C7 9346 DB $0A, $5F, $23, $C7
EE5E AD60EC78 9347 DB $AD, $60, $EC, $78
EE62 586227D7 9348 DB $58, $62, $27, $D7
EE66 07647886 9349 DB $07, $64, $78, $86
EE6A A9651668 9350 DB $A9, $65, $16, $68
EE6E 53671CC2 9351 DB $53, $67, $1C, $C2
EE72 04695159 9352 DB $04, $69, $51, $59
EE76 A56AA66F 9353 DB $A5, $6A, $A6, $6F
EE7A 4E6C8FCB 9354 DB $4E, $6C, $8F, $CB
EE7E 016E393F 9355 DB $01, $6E, $39, $3F
EE82 A16F088F 9356 DB $A1, $6F, $08, $8F
EE86 4971CAF2 9357 DB $49, $71, $CA, $F2
EE8A FC727C6F 9358 DB $FC, $72, $7C, $6F
EE8E 9D74AEC5 9359 DB $9D, $74, $AE, $C5
EE92 45761937 9360 DB $45, $76, $19, $37
EE96 F677DF84 9361 DB $F6, $77, $DF, $84
EE9A 9A790C13 9362 DB $9A, $79, $0C, $13
EE9E 407BCE97 9363 DB $40, $7B, $CE, $97
EEA2 F07CC2BD 9364 DB $F0, $7C, $C2, $BD
EEA6 967E9C76 9365 DB $96, $7E, $9C, $76
9366 ;. PAGE 'More support for CSP'
9367 ;
9368 ;
9369 ;
EEAA E3D1 9370 I NULL DW ERR8
EEAC 00 9371 I NULO DB $00
9372 ;
9373 ;
9374 ;
EEAD 03FF 9375 I NULP DW JCWRT
EEAF 03FF 9376 DW JCWRT
EEB1 21FF 9377 DW JGRAFW
EEB3 0000 9378 DW 0
EEB5 0000 9379 DW 0
EEB7 09FF 9380 DW JPRTW
EEB9 0000 9381 DW 0

```

```

EEBB 1BFF      9382      DW      JREMW
                9383      ;
                9384      ;
                9385      ;
EEBD 00FF      9386      I NULS      DW      JCREAD
EEBF 00FF      9387      DW      JCREAD
EEC1 0000      9388      DW      0
EEC3 0000      9389      DW      0
EEC5 0000      9390      DW      0
EEC7 0000      9391      DW      0
EEC9 18FF      9392      DW      JREMR
EECB 0000      9393      DW      0
                9394      ; . PAGE
                9395      ;
                9396      ;   Validate volume number
                9397      ;
EECD 2980      9398      VALVNO      AND      #$80
EECF D00A      9399      BNZ      I NULZ
EED1 8A        9400      TXA
EED2 C90D      9401      CMP      #MAXVOL
EED4 B019      9402      BGE      I NUMA
EED6 C901      9403      CMP      #1
EED8 9015      9404      BL      I NUMA
                9405      ;
                9406      ;
                9407      ;
EEDA 60        9408      RTS
                9409      ;
                9410      ;
                9411      ;
EEDB 8A        9412      I NULZ      TXA
EEDC 297F      9413      AND      #$7F
EEDE 858C      9414      STA      A8C
EEE0 C910      9415      CMP      #$10
EEE2 B00B      9416      BGE      I NUMA
EEE4 0A        9417      ASL      A
EEE5 658C      9418      ADC      A8C
EEE7 A8        9419      TAY
EEE8 B982FE    9420      LDA      UDJMPV+2, Y
EEEB F002      9421      BZ      I NUMA
EEED 8A        9422      TXA
                9423      ;
                9424      ;
                9425      ;
EEEE 60        9426      RTS
                9427      ;
                9428      ;
                9429      ;
EEEF A202      9430      I NUMA      LDX      #2
EEF1 8EDEBD    9431      STX      I ORSLT
                9432      ;
                9433      ;
                9434      ;
EEF4 68        9435      PLA
EEF5 68        9436      PLA
                9437      ;
                9438      ;

```

```

9439 ;
EEF6 4C3BD2 9440 JMP UPDBY2
9441
*****
9442 ;
9443 ; Push IORESULT onto stack
9444 ;
9445 PSHI OR
EEF9 ADDFBD 9446 LDA IORSLT+1
EEFC 48 9447 PHA
EEFD ADDEBD 9448 LDA IORSLT
EFO0 48 9449 PHA
9450 ;
9451 ;
9452 ;
EFO1 4C3BD2 9453 JMP UPDBY2
9454
*****
9455 ;
9456 ;
9457 ;
EFO4 ADDEBD 9458 CSPO LDA IORSLT
EFO7 FO03 9459 BZ I NUME
9460 ;
9461 ;
9462 ;
EFO9 4CEBD1 9463 JMP ERR10
9464 ;
9465 ;
9466 ;
EFOC 4C3BD2 9467 I NUME JMP UPDBY2
9468
*****
9469 ;
9470 ;
9471 ;
EFOF 68 9472 CSP35 PLA
EF10 AA 9473 TAX
EF11 68 9474 PLA
EF12 8A 9475 TXA
9476 ;
9477 ;
9478 ;
EF13 20CDEE 9479 JSR VALVNO
9480 ;
9481 ;
9482 ;
EF16 A900 9483 LDA #0
EF18 48 9484 PHA
EF19 48 9485 PHA
9486 ;
9487 ;
9488 ;
EF1A 4C3BD2 9489 JMP UPDBY2
9490
*****
9491 ;

```

```

          9492 ;
          9493 ;
EF1D 68   9494 CSP37   PLA
EF1E AA   9495     TAX
EF1F 68   9496     PLA
EF20 8A   9497     TXA
          9498 ;
          9499 ;
          9500 ;
EF21 20CDEE 9501     JSR   VALVNO
          9502 ;
          9503 ;
          9504 ;
EF24 4C3BD2 9505     JMP   UPDBY2
          9506
*****
          9507 ;
          9508 ;
          9509 ;
EF27 A904  9510 CSP12   LDA   #S04
EF29 857C  9511     STA   A7C
          9512 ;
          9513 ;
          9514 ;
EF2B 68   9515     PLA
EF2C 8582  9516     STA   A82
EF2E 68   9517     PLA
EF2F 8583  9518     STA   A82+1
          9519 ;
          9520 ;
          9521 ;
EF31 68   9522     PLA
EF32 857A  9523     STA   A7A
EF34 68   9524     PLA
EF35 857B  9525     STA   A7A+1
          9526 ;
          9527 ;
          9528 ;
EF37 18   9529     CLC
EF38 68   9530     PLA
EF39 657A  9531     ADC   A7A
EF3B 857A  9532     STA   A7A
EF3D 68   9533     PLA
EF3E 657B  9534     ADC   A7A+1
EF40 857B  9535     STA   A7A+1
          9536 ;
          9537 ;
          9538 ;
EF42 68   9539     PLA
EF43 AA   9540     TAX
EF44 68   9541     PLA
EF45 8A   9542     TXA
EF46 20CDEE 9543     JSR   VALVNO
          9544 ;
          9545 ;
          9546 ;
EF49 A583  9547     LDA   A82+1

```

```

EF4B 48      9548    PHA
EF4C A582    9549    LDA    A82
EF4E 48      9550    PHA
              9551    ;
              9552    ;
              9553    ;
EF4F A57B    9554    LDA    A7A+1
EF51 48      9555    PHA
EF52 A57A    9556    LDA    A7A
EF54 48      9557    PHA
              9558    ;
              9559    ;
              9560    ;
EF55 8A      9561    TXA
EF56 2980    9562    AND    #$80
EF58 F002    9563    BZ     I NUMF
              9564    ;
              9565    ;   Uncondi tional  branch
              9566    ;
EF5A D058    9567    BNZ    I NUMM
              9568    ;
              9569    ;
              9570    ;
EF5C 8A      9571    I NUMF    TXA
              9572    i fdef  ORIG
EF5D C907    9573    CMP    #$07
EF5F D003    9574    BNE    I NUMG
EF61 A908    9575    LDA    #$08
EF63 AA      9576    TAX
              9577    I NUMG
              9578    endi f
EF64 0A      9579    ASL    A
EF65 A8      9580    TAY
EF66 B9AFFE  9581    LDA    DSKNUM- 1, Y
EF69 F006    9582    BZ     I NUMH
EF6B C9FF    9583    CMP    #$FF
EF6D F00B    9584    BE     I NUMI
EF6F D05F    9585    BNE    I NUMP
              9586    ;
              9587    ;
              9588    ;
EF71 B9AEFE  9589    I NUMH    LDA    DSKNUM- 2, Y
EF74 2030FF  9590    JSR    JDSKS
EF77 4C30F0  9591    JMP    STOI OR
              9592    ;
              9593    ;
              9594    ;
EF7A 8A      9595    I NUMI    TXA
EF7B A8      9596    TAY
EF7C A204    9597    LDX    #4
EF7E C003    9598    CPY    #3
EF80 B006    9599    BGE    I NUMJ
EF82 202AFF  9600    JSR    JCONS
EF85 4C30F0  9601    JMP    STOI OR
              9602    ;
              9603    ;
              9604    ;

```

```

EF88 C006      9605  INUMJ   CPY   #6
EF8A D006      9606      BNE   INUMK
EF8C 202DFF    9607      JSR   JPRTS
EF8F 4C30F0    9608      JMP   STOIOR
              9609      ;
              9610      ;
              9611      ;
              9612  INUMK
              9613  i fndef ORIG
              9614      CPY   #7
              9615      BE    JSRLST
              9616  endi f
EF92 C008      9617      CPY   #8
EF94 D006      9618      BNE   INUML
              9619  i fndef ORIG
              9620  JSRLST
              9621  endi f
EF96 2033FF    9622      JSR   JREMS
EF99 4C30F0    9623      JMP   STOIOR
              9624      ;
              9625      ;
              9626      ;
EF9C A209      9627  INUML   LDX   #9
              9628      ;
              9629      ;
              9630      ;
EF9E 68        9631      PLA
EF9F 68        9632      PLA
EFA0 68        9633      PLA
EFA1 68        9634      PLA
              9635      ;
              9636      ;
              9637      ;
EFA2 4C30F0    9638      JMP   STOIOR
              9639  ;. PAGE 'System procedure 38'
              9640      ;
              9641      ;
              9642      ;
EFA5 A902      9643  CSP38   LDA   #S02
EFA7 857C      9644      STA   A7C
              9645      ;
              9646      ;
              9647      ;
EFA9 68        9648      PLA
EFAA AA        9649      TAX
EFAB 68        9650      PLA
EFAC 8A        9651      TXA
EFAD 20CDEE    9652      JSR   VALVNO
              9653      ;
              9654      ;
              9655      ;
EFB0 2980      9656      AND   #S80
EFB2 F009      9657      BZ    INUMN
              9658      ;
              9659      ;
              9660      ;
EFB4 8A        9661  INUMM   TXA

```



```

EFB5 A67C      9662      LDX  A7C
EFB7 2039FF    9663      JSR  JUSER
EFBA 4C30FO    9664      JMP  STOIOR
                9665      ;
                9666      ;
                9667      ;
EFBD 8A        9668      I NUMN  TXA
EFBE C907      9669      CMP   #7
EFCO D003      9670      BNE  I NUMO
                9671      ;
                9672      ;
                9673      ;
EFC2 A908      9674      LDA  #8
EFC4 AA        9675      TAX
                9676      ;
                9677      ;
                9678      ;
EFC5 0A        9679      I NUMO  ASL  A
EFC6 A8        9680      TAY
EFC7 B9AFFE    9681      LDA  DSKNUM- 1, Y
EFCA FOOD      9682      BZ   I NUMQ
EFCC C9FF      9683      CMP  #$FF
EFCE F012      9684      BE   I NUMR
EFD0 8A        9685      I NUMP  TXA
EFD1 A67C      9686      LDX  A7C
EFD3 203CFF    9687      JSR  JUDDRV
EFD6 4C30FO    9688      JMP  STOIOR
                9689      ;
                9690      ;
                9691      ;
EFD9 B9AEFE    9692      I NUMQ  LDA  DSKNUM- 2, Y
EFDC 2015FF    9693      JSR  JDI NIT
EFD F 4C30FO    9694      JMP  STOIOR
                9695      ;
                9696      ;
                9697      ;
EFE2 8A        9698      I NUMR  TXA
EFE3 A8        9699      TAY
EFE4 A202      9700      LDX  #2
EFE6 C003      9701      CPY  #3
EFE8 B01C      9702      BGE  I NUMS
                9703      ;
                9704      ;
                9705      ;
EFEA A900      9706      LDA  #0
EFEC 8597      9707      STA  A97
EFEE 8598      9708      STA  A98
                9709      ;
                9710      ;
                9711      ;
EFF0 ADABEE    9712      LDA  I NULL+1
EFF3 48        9713      PHA
EFF4 ADA AEE    9714      LDA  I NULL
EFF7 48        9715      PHA
                9716      ;
                9717      ;
                9718      ;

```

```

EFF8 AD18F2    9719    LDA    ADRIOR+1
EFFB 48        9720    PHA
EFFC AD17F2    9721    LDA    ADRIOR
EFFF 48        9722    PHA
                9723    ;
                9724    ;
                9725    ;
FO00 2006FF    9726    JSR    JCINIT
                9727    ;
                9728    ;
                9729    ;
FO03 4C30F0    9730    JMP    STOIOR
                9731    ;
                9732    ;
                9733    ;
FO06 C006      9734    INUMS   CPY    #6
FO08 D00A      9735    BNE    INUMT
                9736    ;
                9737    ;
                9738    ;
FO0A A900      9739    LDA    #0
FO0C 859C      9740    STA    A9C
                9741    ;
                9742    ;
                9743    ;
FO0E 200CFF    9744    JSR    JPRTI
FO11 4C30F0    9745    JMP    STOIOR
                9746    ;
                9747    ;
                9748    ;
FO14 C008      9749    INUMT   CPY    #8
FO16 D006      9750    BNE    INUMU
                9751    ;
                9752    ;
                9753    ;
                9754    ifndef ORIG
                9755    LDA    #0
                9756    STA    ZA7
                9757    STA    ZA7+1
                9758    endif
FO18 201EFF    9759    JSR    JREMI
FO1B 4C30F0    9760    JMP    STOIOR
                9761    ;
                9762    ;
                9763    ;
FO1E C003      9764    INUMU   CPY    #3
FO20 D003      9765    BNE    INUMV
FO22 2024FF    9766    JSR    JGRAFI
FO25 A209      9767    INUMV   LDX    #9
FO27 4C30F0    9768    JMP    STOIOR
                9769    ;
                9770    ;
                9771    ;
FO2A ADACEE    9772    UNTOKX  LDA    INULO
FO2D 8D1CBF    9773    STA    SPCHAR
                9774    ;
                9775    ;

```

```

9776 ;
F030 8EDEBD 9777 STOI OR STX IORSLT
9778 ;
9779 ;
9780 ;
F033 4C3BD2 9781 JMP UPDBY2
9782 ; . PAGE '???'
9783 ;
9784 ;
9785 ;
F036 68 9786 INUNN PLA
F037 858E 9787 STA A8E
F039 68 9788 PLA
F03A 858F 9789 STA A8E+1
9790 ;
9791 ;
9792 ;
F03C A583 9793 LDA A82+1
F03E 48 9794 PHA
F03F A582 9795 LDA A82
F041 48 9796 PHA
9797 ;
9798 ;
9799 ;
F042 A900 9800 LDA #$00
F044 48 9801 PHA
F045 8A 9802 TXA
F046 48 9803 PHA
9804 ;
9805 ;
9806 ;
F047 A57B 9807 LDA A7A+1
F049 48 9808 PHA
F04A A57A 9809 LDA A7A
F04C 48 9810 PHA
9811 ;
9812 ;
9813 ;
F04D A579 9814 LDA A78+1
F04F 48 9815 PHA
F050 A578 9816 LDA A78
F052 48 9817 PHA
9818 ;
9819 ;
9820 ;
F053 A577 9821 LDA A76+1
F055 48 9822 PHA
F056 A576 9823 LDA A76
F058 48 9824 PHA
9825 ;
9826 ;
9827 ;
F059 A58F 9828 LDA A8E+1
F05B 48 9829 PHA
F05C A58E 9830 LDA A8E
F05E 48 9831 PHA
9832 ;

```

```

          9833 ;
          9834 ;
F05F 60   9835   RTS
          9836 ;. PAGE '???'
          9837 ;
          9838 ;
          9839 ;
F060 ADACEE 9840 I NUNQ   LDA   I NULO
F063 8D1CBF 9841   STA   SPCHAR
          9842 ;
          9843 ;
          9844 ;
F066 4C3BD2 9845   JMP   UPDBY2
          9846
*****
          9847 ;
          9848 ;
          9849 ;
F069 A900   9850 UNI TRD   LDA   #0
F06B 4C70F0 9851   JMP   UNI TRW
          9852 ;
          9853 ;
          9854 ;
F06E A901   9855 UNI TWR   LDA   #1
          9856 ;
          9857 ;
          9858 ;
F070 857C   9859 UNI TRW   STA   A7C
          9860 ;
          9861 ;
          9862 ;
F072 A001   9863   LDY   #1
F074 847E   9864   STY   A7E
F076 8480   9865   STY   A80
F078 AC1CBF 9866   LDY   SPCHAR
F07B 8CACEE 9867   STY   I NULO
F07E 68     9868   PLA
F07F AA     9869   TAX
F080 8582   9870   STA   A82
F082 4A     9871   LSR   A
F083 4A     9872   LSR   A
F084 4A     9873   LSR   A
F085 4A     9874   LSR   A
F086 2903   9875   AND   #$03
F088 0D1CBF 9876   ORA   SPCHAR
F08B 8D1CBF 9877   STA   SPCHAR
F08E A000   9878   LDY   #0
F090 8CDEBD 9879   STY   I ORSLT
F093 8A     9880   TXA
F094 2904   9881   AND   #$04
F096 D002   9882   BNZ   I NUNU
F098 847E   9883   STY   A7E
F09A 8A     9884 I NUNU   TXA
F09B 2908   9885   AND   #$08
F09D D002   9886   BNZ   I NUNV
F09F 8480   9887   STY   A80
FOA1 68     9888 I NUNV   PLA

```

FOA2	8583	9889	STA	A82+1
FOA4	68	9890	PLA	
FOA5	8576	9891	STA	A76
FOA7	68	9892	PLA	
FOA8	8577	9893	STA	A76+1
FOAA	68	9894	PLA	
FOAB	8578	9895	STA	A78
FOAD	F011	9896	BZ	I NUNZ
FOAF	68	9897	PLA	
FOBO	8579	9898	STA	A78+1
FOB2	3003	9899	BM	UNTERX
FOB4	4CC7F0	9900	JMP	I NUOA
		9901	;	
		9902	;	
		9903	;	
FOB7	68	9904	UNTERX	PLA
FOB8	68	9905	PLA	
FOB9	68	9906	PLA	
FOBA	68	9907	PLA	
FOBB	68	9908	PLA	
FOBC	68	9909	PLA	
FOBD	4C3BD2	9910	JMP	UPDBY2
		9911	;	
		9912	;	
		9913	;	
FOCO	68	9914	I NUNZ	PLA
FOC1	FOF4	9915	BZ	UNTERX
FOC3	30F2	9916	BM	UNTERX
FOC5	8579	9917	STA	A78+1
FOC7	68	9918	I NUOA	PLA
FOC8	857A	9919	STA	A7A
FOCA	68	9920	PLA	
FOCB	857B	9921	STA	A7A+1
FOCD	18	9922	CLC	
FOCE	68	9923	PLA	
FOCF	657A	9924	ADC	A7A
FOD1	857A	9925	STA	A7A
FOD3	68	9926	PLA	
FOD4	657B	9927	ADC	A7A+1
FOD6	857B	9928	STA	A7A+1
FOD8	68	9929	PLA	
FOD9	8584	9930	STA	A84
FODB	AA	9931	TAX	
FODC	68	9932	PLA	
FODD	8A	9933	TXA	
FODE	20CDEE	9934	JSR	VALVNO
FOE1	2980	9935	AND	#\$80
FOE3	FOOC	9936	BZ	I NUOB
FOE5	2036F0	9937	JSR	I NUNN
FOE8	8A	9938	TXA	
FOE9	A67C	9939	LDX	A7C
FOEB	2039FF	9940	JSR	JUSER
FOEE	4C2AF0	9941	JMP	UNTOKX
		9942	;	
		9943	;	
		9944	;	
FOF1	8A	9945	I NUOB	TXA

```

FOF2 OA      9946   ASL   A
FOF3 A8      9947   TAY
FOF4 B9AFFE  9948   LDA   DSKNUM- 1, Y
FOF7 F010    9949   BZ    I NUOC
FOF9 C9FF    9950   CMP   #$FF
FOFB F022    9951   BE    I NUOE
FOFD 2036FO  9952   JSR   I NUNN
F100 8A      9953   TXA
F101 A67C    9954   LDX   A7C
F103 203CFF  9955   JSR   JUDDRV
F106 4C2AFO  9956   JMP   UNTOKX
          9957   ;
          9958   ;
          9959   ;
F109 BEAEFE  9960   I NUOC   LDX   DSKNUM- 2, Y
F10C 2036FO  9961   JSR   I NUNN
F10F A57C    9962   LDA   A7C
F111 D006    9963   BNZ   I NUOD
F113 2012FF  9964   JSR   JDREAD
F116 4C2AFO  9965   JMP   UNTOKX
          9966   ;
          9967   ;
          9968   ;
F119 200FFF  9969   I NUOD   JSR   JDWRT
F11C 4C2AFO  9970   JMP   UNTOKX
          9971   ;
          9972   ;
          9973   ;
F11F A901    9974   I NUOE   LDA   #1
F121 E001    9975   CPX   #1
F123 D002    9976   BNE   I NUOF
F125 A900    9977   LDA   #0
F127 8574    9978   I NUOF   STA   A74
          9979   i fndef   ORIG
          9980   CPX   #9
          9981   BL   TXA
          9982   LDX   #2
          9983   BNZ   JUOKX
          9984   TXA
          9985   endi f
F129 8A      9986   TXA
F12A OA      9987   ASL   A
F12B AA      9988   TAX
F12C BDABEE  9989   LDA   I NULP- 2, X
F12F 858A    9990   STA   A8A
F131 BDACEE  9991   LDA   I NULP- 1, X
F134 858B    9992   STA   A8A+1
F136 BDBBEE  9993   LDA   I NULS- 2, X
F139 8586    9994   STA   A86
F13B BDBCEE  9995   LDA   I NULS- 1, X
F13E 8587    9996   STA   A86+1
F140 A57C    9997   LDA   A7C
F142 F009    9998   BZ    I NUOH
F144 A58B    9999   LDA   A8A+1
F146 D009    10000  BNZ   I NUOI
F148 A203    10001  I NUOG   LDX   #3
F14A 4C2AFO  10002  JUOKX   JMP   UNTOKX

```

```

F14D A587      10003  I NUOH   LDA   A86+1
F14F F0F7      10004      BZ    INUOG
F151 A678      10005  I NUOI   LDX   A78
F153 4CA1F1    10006      JMP    INUOV
F156 A57C      10007  I NUOJ   LDA   A7C
F158 D038      10008      BNZ   INUOR
F15A 2002F2    10009      JSR   INUPN
F15D A200      10010      LDX   #0
F15F 817A      10011      STA   (A7A, X)
F161 A47E      10012      LDY   A7E
F163 D023      10013      BNZ   INUOP
F165 CD30BE    10014      CMP   ABE30
F168 D01E      10015      BNE   INUOP
F16A A001      10016      LDY   #1
F16C C484      10017      CPY   A84
F16E F003      10018      BE    INUOL
F170 4C60FO    10019      JMP    INUNQ
F173 A900      10020  I NUOL   LDA   #0
F175 817A      10021  I NUOM   STA   (A7A, X)
F177 E67A      10022      INC   A7A
F179 D002      10023      BNZ   INUOO
F17B E67B      10024      INC   A7A+1
F17D C678      10025  I NUOO   DEC   A78
F17F DOF4      10026      BNZ   INUOM
F181 C679      10027      DEC   A78+1
F183 10F0      10028      BNM   INUOM
F185 4C60FO    10029      JMP    INUNQ
F188 A474      10030  I NUOP   LDY   A74
F18A D00D      10031      BNZ   INUOS
F18C 20AAF1    10032      JSR   INUOW
F18F 4C99F1    10033      JMP    INUOS
F192 A200      10034  I NUOR   LDX   #0
F194 A17A      10035      LDA   (A7A, X)
F196 20AAF1    10036      JSR   INUOW
F199 E67A      10037  I NUOS   INC   A7A
F19B D002      10038      BNZ   INUOU
F19D E67B      10039      INC   A7A+1
F19F C678      10040  I NUOU   DEC   A78
F1A1 DOB3      10041  I NUOV   BNZ   INUOJ
F1A3 C679      10042      DEC   A78+1
F1A5 10AF      10043      BNM   INUOJ
F1A7 4C60FO    10044      JMP    INUNQ
10045

```

\*\*\*\*\*

```

10046 ;
10047 ;
10048 ;
F1AA A47E      10049  I NUOW   LDY   A7E
F1AC D028      10050      BNZ   INUPB
F1AE A484      10051      LDY   A84
F1B0 B696      10052      LDX   A96, Y
F1B2 F018      10053      BZ    INUPA
F1B4 38        10054      SEC
F1B5 E91F      10055      SBC   #S1F
F1B7 858E      10056      STA   A8E
F1B9 A920      10057  I NUOY   LDA   #S20
F1BB C68E      10058      DEC   A8E

```

```

F1BD F006      10059      BZ      INUOZ
F1BF 20F1F1   10060      JSR     INUPJ
F1C2 4CB9F1   10061      JMP     INUOY
F1C5 A200     10062      INUOZ   LDX    #0
F1C7 A484     10063      LDY     A84
F1C9 9696     10064      STX     A96, Y
F1CB 60       10065      RTS
F1CC C910     10066      INUPA   CMP    #$10
F1CE D006     10067      BNE     INUPB
F1D0 A901     10068      LDA     #$01
F1D2 999600   10069      STA     A96, Y
F1D5 60       10070      RTS
F1D6 A480     10071      INUPB   LDY    A80
F1D8 D00D     10072      BNZ     INUPD
F1DA C90D     10073      CMP     #$0D
F1DC D009     10074      BNE     INUPD
F1DE 20F1F1   10075      JSR     INUPJ
F1E1 A90A     10076      LDA     #$0A
F1E3 20F1F1   10077      JSR     INUPJ
F1E6 60       10078      RTS
F1E7 20F1F1   10079      INUPD   JSR    INUPJ
F1EA 60       10080      RTS
                10081      ;
                10082      ;
                10083      ;
F1EB ECF1     10084      INUPF   DW    AF1ED- 1
                10085      ;
                10086      ;
                10087      ;
F1ED 8EDEBD   10088      AF1ED   STX    I ORSLT
F1F0 60       10089      RTS
                10090
*****
                10091      ;
                10092      ;
                10093      ;
F1F1 A8       10094      INUPJ   TAY
F1F2 ADECF1   10095      LDA     INUPF+1
F1F5 48       10096      PHA
F1F6 ADEBF1   10097      LDA     INUPF
F1F9 48       10098      PHA
F1FA 98       10099      TYA
F1FB A484     10100      LDY     A84
F1FD A201     10101      LDX     #1
F1FF 6C8A00   10102      JMP     (A8A)
                10103
*****
                10104      ;
                10105      ;
                10106      ;
F202 A8       10107      INUPN   TAY
F203 ADECF1   10108      LDA     INUPF+1
F206 48       10109      PHA
F207 ADEBF1   10110      LDA     INUPF
F20A 48       10111      PHA
F20B 98       10112      TYA
F20C A484     10113      LDY     A84

```



```

F20E A200      10114      LDX      #0
F210 6C8600   10115      JMP      (A86)
                71      INCLUDE      INITO.IAS
                1      ;.PAGE 'Initialization - Address constants'
                2      ;
                3      ;
                4      ;
F213 1CBD      5      ADRHIM   DW      ABD1C
F215 3EBE      6      ASYSPT   DW      SYPTBL
F217 DEBD      7      ADRIOR   DW      IORSLT
                8      ifdef  ORIG
                9      ;
                10     ;
                11     ;
F219 00D1     12     ACSPT    DW      CSPTBL
F21B 00D0     13     APCT    DW      PCTBL
                14     endif
                15     ;
                16     ;
                17     ;
F21D A0D6     18     ADRXIT   DW      XIT
F21F 0000     19     AF21F   DW      0
F221 00       20     AF221   DB      0
F222 00       21     AF222   DB      0
                22     ifdef  BIG
                23     ;
                24     ;
                25     ;
                26     ADRM1   DW      A0800
                27     ADRM2   DW      PTRIOP           ; $C000
                28     endif
                29     ;.PAGE 'Initialization - String constants'
                30     ifdef  ORIG
                31     ;
                32     ;
                33     ;
F223 0D535953 34     SYSPK    STR      'SYSTEM.PASCAL'
                35     endif
                36     ;
                37     ;
                38     ;
F231 0D       39     INUPQ    DB      $0D
F232 496E7365 40     ASC      'Insert boot disk with SYSTEM.PASCAL'
F255 0DOA     41     DB      $0D,$0A
F257 6F6E2069 42     ASC      'on it, then press '
                43     ifndef  ORIG
                44     ASC      'RETURN'
                45     else
F269 52455345 46     ASC      'RESET'
                47     endif
F26E 23       48     ASC      '#'
                49     ifndef  ORIG
                50     ;
                51     ;
                52     ;
                53     SYSPK    STR      'SYSTEM.PASCAL'
                54     ASC      ' is not V1.3#'

```

```

55 ;
56 ;
57 ;
58   ifdef BIG
59 MEMRK   DB   SOD
60   ASC   '128K memory required#'
61   endif
62 ;
63 ;
64 ;
65 AF26F   DW   INULP
66   DW   INULS
67 ;
68 ;
69 ;
70 JINTJT   JMP   (NFFE2)
71 ;
72 ;
73 ;
74 ACD2MV   DW   RELOC
75   endif
76   ifdef ORIG
77 ;
78 ;
79 ;
80 AF26F
81   ifndef IIc
F26F ADEEBD 82   LDA   SYCJTB
F272 EE     83   DB   $EE
84   else
85   ADC   $EE,X
86   STA   $EE
87   endif
88 ;
89 ;
90 ;
F273 75F2   91 ACD2MV   DW   MINITC
92   endif
93 ;.PAGE 'Initialization - Copy init code to non-bank RAM'
94 ;
95 ;   Set source pointer
96 ;
F275 AD73F2 97 MINITC   LDA   ACD2MV
F278 8500   98   STA   A00
F27A AD74F2 99   LDA   ACD2MV+1
F27D 8501   100  STA   A00+1
101 ;
102 ;   Set destination pointer
103 ;
F27F A968   104   LDA   #<A6800
F281 8503   105   STA   A02+1
F283 A900   106   LDA   #>A6800
F285 8502   107   STA   A02
108 ;
109 ;   Clear index registers
110 ;
111 * LDA   #0

```

---

Apple II Pascal Information

---

```

F287 AA      112      TAX
F288 A8      113      TAY
              114      ;
              115      ; Copy a byte
              116      ;
F289 B100    117      INUPT   LDA   (A00), Y
F28B 9102    118      STA   (A02), Y
              119      ;
              120      ; Branch if page not copied yet
              121      ;
F28D C8      122      INY
F28E D0F9    123      BNZ   INUPT
              124      ;
              125      ; Add 256 to source and destination pointers
              126      ;
F290 E601    127      INC   A00+1
F292 E603    128      INC   A02+1
              129      ;
              130      ; Increment page count
              131      ;
F294 E8      132      INX
              133      ;
              134      ; Branch if desired number of pages not copied yet
              135      ;
F295 E004    136      CPX   #INTPGS
F297 D0F0    137      BNE   INUPT
              138      ;
              139      ;
              140      ;
              141      ifdef ORIG
F299 4C2768  142      * JMP   A6800+RELOC-MINI TC
              143      db $4c, $27, $68
              144      else
              145      JMP   A6800
              146      endif
              147      ;
              148      ;
              149      ;
              150      RELOC
              72      INCLUDE   INIT2.IAS
              1      ;.PAGE 'Part 2 of initialization'
              2      ;
              3      ; Initialize stack pointer
              4      ;
F29C A2FF    5        LDX   #$FF
F29E 9A      6        TXS
              7        ;
              8        ; Clear decimal flag (again)
              9        ;
F29F D8      10       CLD
              11      ;
              12      ;
              13      ;
F2A0 ADAEE   14       LDA   I NULL
F2A3 8D1DBF  15       STA   I BRK
F2A6 ADABEE  16       LDA   I NULL+1
F2A9 8D1EBF  17       STA   I BRK+1

```

```

18 ;
19 ;
20 ;
F2AC AD1DF2 21 LDA ADRXI T
F2AF 8D2FBF 22 STA XI TL0C
F2B2 AD1EF2 23 LDA ADRXI T+1
F2B5 8D30BF 24 STA XI TL0C+1
25 ;
26 ;
27 ;
F2B8 AD17F2 28 LDA ADRI OR
F2BB 8D1FBF 29 STA I SYSC
F2BE AD18F2 30 LDA ADRI OR+1
F2C1 8D20BF 31 STA I SYSC+1
32 ;
33 ;
34 ;
35 i f n d e f  O R I G
36 LDA #FLAVK
37 STA FLAVOR
38 LDA #VERK
39 STA VER
40 e l s e
F2C4 A902 41 LDA #VERK
F2C6 8D21BF 42 STA VER
F2C9 A901 43 LDA #FLAVK
F2CB 8D22BF 44 STA FLAVOR
45 e n d i f
46 ;
47 ;
48 ;
F2CE AD6FF2 49 LDA AF26F
F2D1 85E6 50 STA WTPTR
F2D3 AD70F2 51 LDA AF26F+1
F2D6 85E7 52 STA WTPTR+1
53 ;
54 ;
55 ;
F2D8 AD71F2 56 LDA AF26F+2
F2DB 85E4 57 STA RTPTR
F2DD AD72F2 58 LDA AF26F+3
F2E0 85E5 59 STA RTPTR+1
60 i f d e f  B I G
61 ; . PAGE
62 ;
63 ;
64 ;
65 LDA ADRM1
66 STA BZ62
67 LDA ADRM1+1
68 STA BZ62+1
69 ;
70 ;
71 ;
72 LDA #>SYSTEMEM
73 STA BZ64
74 STA BZ66

```

```

75     LDA    #<SYSTEMEM
76     STA    BZ64+1
77     STA    BZ66+1
78     endi f
79     ;. PAGE
80     ;
81     ; Move 42 mystery bytes to SYSCOM
82     ;
F2E2 A029 83     LDY    #41
84     INVAA
85     ifndef ORIG
86     LDA    INVAB+RELOC- CLRMEM, Y
87     el se
F2E4 B908F3 88     LDA    INVAB, Y
89     endi f
F2E7 9956BF 90     STA    ABF56, Y
F2EA 88     91     DEY
F2EB 10F7   92     BNM    INVAA
93     ;
94     ; Move address of 42 mystery bytes to SYSCOM
95     ;
F2ED A956   96     LDA    #>ABF56
F2EF 8D23BF 97     STA    ABF23
F2F2 A9BF   98     LDA    #<ABF56
F2F4 8D24BF 99     STA    ABF23+1
100    ;
101    ;
102    ;
F2F7 ADB7FE 103    LDA    DSKNUM+7
F2FA F036   104    BZ     INVAC
105    ;
106    ;
107    ;
F2FC A202   108    LDX    #$02
F2FE A904   109    LDA    #$04
F300 A008   110    LDY    #$08
F302 203CFF 111    JSR    JUDDRV
F305 18     112    CLC
F306 902F   113    BNC    INVAD
114    ;
115    ;
116    ;
F308 COD2   117    INVAB   DB     $C0, $D2
F30A 5CD5   118    DB     $5C, $D5
F30C 7AD3   119    DB     $7A, $D3
F30E OFD2   120    DB     $0F, $D2
F310 20B3B3 121    DB     $20, $B3, $B3
F313 A54685 122    DB     $A5, $46, $85
F316 4CB3   123    DB     $4C, $B3
F318 B34CB3B3 124    DB     $B3, $4C, $B3, $B3, $0E, $49, $AB, $45
F320 CD484A68 125    DB     $CD, $48, $4A, $68, $2A, $C9, $04, $2A
F328 854660A4 126    DB     $85, $46, $60, $A4, $46, $24, $00
F32F 4C10D3 127    DB     $4C, $10, $D3
128    ;
129    ;
130    ;
F332 A904   131    INVAC   LDA    #$04

```

```

F334 2015FF      132      JSR      JDINIT
                  133      ;. PAGE
                  134      ;
                  135      ;   Clear registers
                  136      ;
                  137      i f n d e f  O R I G
F337 A200        138      I N V A D      L D A      # 0
F339 A900        139      T A X
                  140      e l s e
F337 A200        141      I N V A D      L D X      # 0
F339 A900        142      L D A      # 0
                  143      e n d i f
                  144      ;
                  145      ;   Clear 256 bytes
                  146      ;
F33B 9DDEBD     147      I N V A E      S T A      I O R S L T , X
F33E E8          148      I N X
F33F DOFA        149      B N Z      I N V A E
                  150      ;
                  151      ;
                  152      ;
F341 9DDEBE     153      I N V A F      S T A      I O R S L T + 2 5 6 , X
F344 E8          154      I N X
F345 E020        155      C P X      # X X X - I O R S L T - 2 5 6
F347 DOF8        156      B N E      I N V A F
                  157      ;. PAGE 'Load directory'
                  158      ;
                  159      ;   Push zero word onto stack
                  160      ;
F349 A900        161      L D A      # 0
F34B 48          162      P H A
F34C 48          163      P H A
                  164      ;
                  165      ;   Push ??? onto stack
                  166      ;
                  167      * L D A      # 0
F34D 48          168      P H A
F34E ADB7FE     169      L D A      D S K N U M + 7
F351 F004        170      B Z      I N V A G
F353 A904        171      L D A      # $ 0 4
F355 1003        172      B N M      I N V A H
F357 ADB6FE     173      I N V A G      L D A      D S K N U M + 6
F35A 48          174      I N V A H      P H A
                  175      ;
                  176      ;   Push directory read address onto stack
                  177      ;
F35B A960        178      L D A      # < A 6 0 0 0
F35D 48          179      P H A
F35E 8581        180      S T A      A 8 0 + 1
F360 A900        181      L D A      # > A 6 0 0 0
F362 48          182      P H A
F363 8580        183      S T A      A 8 0
                  184      ;
                  185      ;   Push directory block count onto stack
                  186      ;
F365 A908        187      L D A      # > 8
F367 48          188      P H A

```

```

F368 A900      189      LDA    #<8
F36A 48        190      PHA
                191      ;
                192      ;   Push starting block number onto stack
                193      ;
                194      * LDA    #<2
F36B 48        195      PHA
F36C A902      196      LDA    #>2
F36E 48        197      PHA
                198      ;
                199      ;   Read directory into memory
                200      ;
F36F ADB7FE    201      LDA    DSKNUM+7
F372 FOOC      202      BZ     INVAI
                203      ;
                204      ;
                205      ;
F374 A200      206      LDX    #$00
F376 A904      207      LDA    #04
F378 A008      208      LDY    #08
F37A 203CFF    209      JSR    JUDDRV
F37D 18        210      CLC
F37E 9003      211      BNC    BEGBQ
                212      ;
                213      ;
                214      ;
F380 2012FF    215      INVAI    JSR    JDREAD
                216      ;
                217      ;
                218      ;
                219      BEGBQ
                220      i f n d e f  O R I G
                221      ; . PAGE '???'
                222      ;
                223      ;
                224      ;
                225      LDA    A6000+25
                226      AND    #08
                227      BNZ    BEGBR
                228      ;
                229      ;
                230      ;
                231      BIT    ABF31
                232      BNV    BEGBS
                233      LDA    AC060
                234      BNM    BEGBS
                235      BEGBR    LDA    #00
                236      BZ     BEGBU
                237      BEGBS    LDA    ABFOE
                238      STA    SLTYP5+3
                239      ;
                240      ;
                241      ;
                242      LDY    #30
                243      ;
                244      ;   Read and write RAM; use bank #2
                245      ;

```

```

246     LDX   LCBNK2
247     ;
248     ;
249     ;
250     JSR   JINTJT
251     ;
252     ; Read and write RAM; use bank #1
253     ;
254     LDA   LCBNK1
255     ;
256     ;
257     ;
258     LDA   #$00
259     CPX   #$00
260     BNE   BEGBU
261     LDA   #$04
262     BEGBU STA   ABFOE
263     ifdef BIG
264     ;
265     ;
266     ;
267     LDA   ABF31
268     AND   #$02
269     BNZ   INVAJ
270     TAY
271     STY   AF222
272     INVAM LDA   MEMRK, Y
273     CMP   #' #'
274     BE    *
275     JSR   JCWRT
276     INC   AF222
277     LDY   AF222
278     CLC
279     BNC   INVAM
280     INVAJ
281     endif
282     endif
283     ; .PAGE 'Search directory for SYSTEM.PASCAL'
284     ;
285     ; Set count of entries in entire directory
286     ;
F383 A24E 287     LDX   #78
288     ifndef ORIG
289     ;
290     ; Set directory entry pointer
291     ;
292     LDA   #<A6000
293     STA   A80+1
294     LDA   #>A6000
295     STA   A80
296     endif
297     ;
298     ; Point at name field
299     ;
F385 18   300     CLC
F386 A580 301     LDA   A80
F388 6906 302     ADC   #6

```



```

F38A 8580      303      STA      A80
F38C 900D      304      BNC      INVAL
F38E E681      305      INC      A80+1
                306      ;
                307      ; Increment directory entry pointer
                308      ;
F390 18        309      INVAK    CLC
F391 A580      310      LDA      A80
F393 691A      311      ADC      #26
F395 8580      312      STA      A80
F397 9002      313      BNC      INVAL
F399 E681      314      INC      A80+1
                315      ;
                316      ; Branch if more entries to check
                317      ;
F39B CA        318      INVAL    DEX
F39C D053      319      BNZ      INVAR
                320      ;
                321      ;
                322      ;
                323      ifndef  ORIG
                324      TXA
                325      else
F39E A900      326      LDA      #0
                327      endif
F3A0 8597      328      STA      A96+1
F3A2 8598      329      STA      A96+2
                330      ;
                331      ;
                332      ;
F3A4 ADABEE    333      LDA      I NULL+1
F3A7 48        334      PHA
F3A8 ADAEEE    335      LDA      I NULL
F3AB 48        336      PHA
F3AC AD18F2    337      LDA      ADRI OR+1
F3AF 48        338      PHA
F3B0 AD17F2    339      LDA      ADRI OR
F3B3 48        340      PHA
F3B4 2006FF    341      JSR      JCINI T
                342      ;
                343      ; Initialize for display of two line "Insert boot disk..."
                344      ;
F3B7 A000      345      LDY      #0
F3B9 8C22F2    346      STY      AF222
                347      ;
                348      ; Branch if message display not complete
                349      ;
F3BC B931F2    350      INVAN    LDA      I NUPQ, Y
F3BF C923      351      CMP      #' #'
F3C1 D022      352      BNE      INVAQ
                353      ifndef  ORIG
                354      ;
                355      ;
                356      ;
                357      LDA      AF221
                358      BNZ      BEGCA
                359      ;

```

```

360 ;
361 ;
362   INC   AF221
363 ;
364 ;   Wait for press of any key
365 ;
366 BEGCA   LDA   KBD
367       BNM   BEGCA
368 ;
369 ;   Clear stobre
370 ;
371   STA   KBSTRB
372 ;
373 ;   Branch if not return key pressed
374 ;
375   AND   #$7F
376   CMP   #$0D
377   BNE   BEGCA
378 ;
379 ;
380 ;
381   JMP   (AFFF8)
382   else
383 ;
384 ;
385 ;
F3C3 AD21F2 386   LDA   AF221
F3C6 DOFE   387 INVAO   BNZ   INVAO
388 ;
389 ;
390 ;
F3C8 ADFCFF 391   LDA   AFFFC
F3CB 8D1FF2 392   STA   AF21F
F3CE ADFDFF 393   LDA   AFFFC+1
F3D1 8D20F2 394   STA   AF21F+1
395 ;
396 ;
397 ;
F3D4 ADF8FF 398   LDA   AFFF8
F3D7 8DFCFF 399   STA   AFFFC
F3DA ADF9FF 400   LDA   AFFF8+1
F3DD 8DFDFF 401   STA   AFFFC+1
402 ;
403 ;
404 ;
F3E0 EE21F2 405   INC   AF221
F3E3 DOFE   406 INVAP   BNZ   INVAP
407   endi f
408 ;
409 ;   Display character of message
410 ;
F3E5 2003FF 411 INVAQ   JSR   JCWRT
412 ;
413 ;   Increment index into message
414 ;
F3E8 EE22F2 415   INC   AF222
F3EB AC22F2 416   LDY   AF222

```

```

417 ;
418 ; Go continue message display
419 ;
F3EE 18 420 CLC
F3EF 90CB 421 BNC INVAN
422 ;
423 ; Compare entry file name against "SYSTEM.PASCAL"
424 ;
F3F1 AC23F2 425 INVAR LDY SYSPK
F3F4 B180 426 INVAS LDA (A80),Y
F3F6 D923F2 427 CMP SYSPK,Y
F3F9 D095 428 BNE INVAK
F3FB 88 429 DEY
F3FC 10F6 430 BNM INVAS
431 ;.PAGE 'Load SYSTEM.PASCAL'
432 ;
433 ; Calculate address of start of entry
434 ;
F3FE A580 435 LDA A80
F400 38 436 SEC
F401 E906 437 SBC #6
F403 8580 438 STA A80
F405 B002 439 BC INVAT
F407 C681 440 DEC A80+1
441 ;
442 ; Extract starting block number
443 ;
F409 A000 444 INVAT LDY #0
F40B B180 445 LDA (A80),Y
F40D 857A 446 STA BLKNUM
F40F C8 447 INY
F410 B180 448 LDA (A80),Y
F412 857B 449 STA BLKNUM+1
450 ;
451 ;
452 ;
F414 A900 453 LDA #0
F416 48 454 PHA
F417 48 455 PHA
456 ;
457 ;
458 ;
459 * LDA #0
F418 48 460 PHA
F419 ADB7FE 461 LDA DSKNUM+7
F41C F004 462 BZ INVAV
F41E A904 463 LDA #S04
F420 1003 464 BNM INVAV
F422 ADB6FE 465 INVAV LDA DSKNUM+6
F425 48 466 INVAV PHA
467 ;
468 ;
469 ;
F426 A960 470 LDA #<A6000
F428 48 471 PHA
F429 A900 472 LDA #>A6000
F42B 48 473 PHA

```

```

474 ;
475 ;
476 ;
F42C A901 477 LDA #>INITK3
F42E 48 478 PHA
F42F A900 479 LDA #<INITK3
F431 48 480 PHA
481 ;
482 ;
483 ;
F432 A57B 484 LDA BLKNUM+1
F434 48 485 PHA
F435 A57A 486 LDA BLKNUM
F437 48 487 PHA
488 ;
489 ; Read SYSTEM.PASCAL into memory
490 ;
F438 ADB7FE 491 LDA DSKNUM+7
F43B FOOC 492 BZ INVAW
493 ;
494 ;
495 ;
F43D A200 496 LDX #$00
F43F A904 497 LDA #$04
F441 A008 498 LDY #$08
F443 203CFF 499 JSR JUDDRV
F446 18 500 CLC
F447 9003 501 BNC INVAX
502 ;
503 ;
504 ;
F449 2012FF 505 INVAW JSR JDREAD
506 ; .PAGE 'Initialize SYSCOMM tables'
507 ;
508 ; Clear segment use counter table
509 ;
510 INVAX
511 ifndef ORIG
512 LDA #0
513 TAX
514 else
F44C A200 515 LDX #0
F44E A900 516 LDA #0
517 endif
F450 9D1EBD 518 INVAY STA SUCTBL, X
F453 E8 519 INX
F454 E040 520 CPX #INITK2
F456 DOF8 521 BNE INVAY
522 ;
523 ; Initialize destination pointer
524 ;
F458 AD15F2 525 LDA ASYSPT
F45B 8574 526 STA A74
F45D AD16F2 527 LDA ASYSPT+1
F460 8575 528 STA A74+1
529 ifndef ORIG
530 ;

```

```

531 ;
532 ;
533 LDA #<A6000+256
534 STA A76+1
535 LDA #>A6000+256
536 STA A76
537 LDY #3
538 LDA (A76), Y
539 DEC A76+1
540 ;
541 ; Branch if correct version of SYSTEM.PASCAL
542 ;
543 AND #$E0
544 CMP #$C0
545 BE BEGCO
546 ;
547 ; Display "...not V1.3" and enter limbo
548 ;
549 LDY #0
550 STY AF222
551 BEGCM LDA SYSPK, Y
552 CMP #' #'
553 BEGCM BE BEGCM
554 JSR JCWRT
555 INC AF222
556 LDY AF222
557 BNM BEGCM
558 BEGCO
559 else
560 ;
561 ;
562 ;
F462 A960 563 LDA #<A6000
F464 8577 564 STA A76+1
F466 A900 565 LDA #>A6000
F468 8576 566 STA A76
567 endif
568 ;.PAGE 'Build ??? table from segment data in .CODE file'
569 ;
570 ;
571 ;
F46A A210 572 LDX #16
F46C A000 573 INVAZ LDY #0
F46E A904 574 LDA #$04
F470 9174 575 STA (A74), Y
576 ifndef ORIG
577 TYA
578 endif
F472 C8 579 INY
580 ifdef ORIG
F473 A900 581 LDA #0
582 endif
F475 9174 583 STA (A74), Y
F477 A574 584 LDA A74
F479 18 585 CLC
F47A 6902 586 ADC #2
F47C 8574 587 STA A74

```

```

F47E 9002      588      BNC      INVBA
F480 E675      589      INC      A74+1
F482 A000      590      INVBA    LDY      #0
F484 18        591      CLC
F485 B176      592      LDA      (A76), Y
F487 657A      593      ADC      BLKNUM
F489 9174      594      STA      (A74), Y
F48B C8        595      INY
F48C B176      596      LDA      (A76), Y
F48E 657B      597      ADC      BLKNUM+1
F490 9174      598      STA      (A74), Y
F492 C8        599      INY
F493 B176      600      LDA      (A76), Y
F495 9174      601      STA      (A74), Y
F497 C8        602      INY
F498 B176      603      LDA      (A76), Y
F49A 9174      604      STA      (A74), Y
605      ;
606      ; Update destination pointer
607      ;
F49C A574      608      LDA      A74
F49E 18        609      CLC
F49F 6904      610      ADC      #4
F4A1 8574      611      STA      A74
F4A3 9002      612      BNC      INVBB
F4A5 E675      613      INC      A74+1
614      ;
615      ; Update source pointer
616      ;
F4A7 A576      617      INVBB    LDA      A76
F4A9 18        618      CLC
F4AA 6904      619      ADC      #4
F4AC 8576      620      STA      A76
F4AE 9002      621      BNC      INVBC
F4B0 E677      622      INC      A76+1
623      ;
624      ; Branch if more segments
625      ;
F4B2 CA        626      INVBC    DEX
F4B3 DOB7      627      BNZ      INVAZ
628      ; .PAGE '???'
629      ifdef BIG
630      ;
631      ;
632      ;
633      LDA      ADRM2
634      STA      BZ60
635      LDA      ADRM2+1
636      STA      BZ60+1
637      endif
638      ;
639      ;
640      ;
F4B5 AD13F2    641      LDA      ADRHIM
F4B8 855C      642      STA      PSKPTR
F4BA 8552      643      STA      MSPTR
F4BC 8DF2BD    644      STA      ABDF2

```

```

F4BF AD14F2    645    LDA    ADRHI M+1
F4C2 855D     646    STA    PSKPTR+1
F4C4 8553     647    STA    MSPTR+1
F4C6 8DF3BD   648    STA    ABDF2+1
649    ;
650    ;   Read mystery segment into memory
651    ;
652    i f n d e f   O R I G
653        LDA    #$0C
654    e l s e
F4C9 A900     655        LDA    #$00
656    e n d i f
F4CB 855B     657        STA    HPPTR+1
F4CD A90F     658        LDA    #15
F4CF 2017E4   659        JSR    RDSEG
660    ;
661    ;
662    ;
663    i f d e f   B I G
664        LDA    BZ60
665    e l s e
F4D2 A55C     666        LDA    PSKPTR
667    e n d i f
F4D4 856C     668        STA    A6C
669    i f d e f   B I G
670        LDA    BZ60+1
671    e l s e
F4D6 A55D     672        LDA    PSKPTR+1
673    e n d i f
F4D8 856D     674        STA    A6C+1
675    ;
676    ;
677    ;
678    i f d e f   B I G
679        LDA    #$FD ; ***
680        STA    BZ60+1
681        LDA    #$FC ; ***
682        STA    BZ60
683    e l s e
F4DA A9FE     684        LDA    #$FE ; ***
F4DC 855D     685        STA    PSKPTR+1
F4DE A97C     686        LDA    #$7C ; ***
F4E0 855C     687        STA    PSKPTR
688    e n d i f
689    ;
690    ;
691    ;
F4E2 A900     692        LDA    #$00
F4E4 20A5E4   693        JSR    LODSEG
694    ;
695    ;
696    ;
F4E7 A56C     697        LDA    A6C
698    i f d e f   B I G
699        STA    BZ60
700    e l s e
F4E9 855C     701        STA    PSKPTR

```

```

702   endi f
F4EB A56D 703   LDA   A6C+1
704   ifdef BIG
705     STA   BZ60+1
706   el se
F4ED 855D 707     STA   PSKPTR+1
708   endi f
709   ;
710   ;
711   ;
F4EF A901 712     LDA   #$01
F4F1 8578 713     STA   PROCNO
F4F3 20A1E0 714     JSR   CALLX
715   ;
716   ;
717   ;
F4F6 A000 718     LDY   #0
F4F8 A552 719     LDA   MSPTR
F4FA 8550 720     STA   BASPTR
F4FC 9152 721     STA   (MSPTR), Y
F4FE A002 722     LDY   #2
F500 9152 723     STA   (MSPTR), Y
F502 A553 724     LDA   MSPTR+1
F504 8551 725     STA   BASPTR+1
F506 88    726     DEY
F507 9152 727     STA   (MSPTR), Y
F509 A003 728     LDY   #3
F50B 9152 729     STA   (MSPTR), Y
730   ;
731   ;
732   ;
F50D A00C 733     LDY   #12
F50F AD17F2 734     LDA   ADRI OR
F512 9152 735     STA   (MSPTR), Y
F514 C8    736     INY
F515 AD18F2 737     LDA   ADRI OR+1
F518 9152 738     STA   (MSPTR), Y
739   ;
740   ;
741   ;
F51A ADOEBF 742     LDA   ABFOE
F51D 2904 743     AND   #$04
F51F F006 744     BZ   INVBD
745   ;
746   ;
747   ;
F521 A908 748     LDA   #$08
F523 855B 749     STA   HPPTR+1
F525 1004 750     BNM   INVBE
751   ;
752   ;
753   ;
F527 A90C 754   INVBD   LDA   #$0C
F529 855B 755     STA   HPPTR+1
756   ;
757   ;
758   ;

```



```

F52B A904      759  INVBE   LDA   #$04
F52D 855A      760     STA   HPPTR
              761     ;
              762     ;
              763     ;
              764  ifndef ORIG
              765     LDX   #<$0100
              766     STX   SPTEMP+1
              767  else
F52F A901      768     LDA   #<$0100
F531 8567      769     STA   SPTEMP+1
              770  endif
              771     ;
              772     ; Clear part of zero page
              773     ;
              774  ifndef ORIG
              775     DEX
              776     TXA
              777  else
F533 A200      778     LDX   #0
F535 A900      779     LDA   #0
              780  endif
F537 9596      781  INVBF   STA   A96, X
F539 E8        782     INX
F53A E00D      783     CPX   #13
F53C D0F9      784     BNE   INVBF
              785  ifdef  ORIG
              786     ;
              787     ; Initialize JMP instructions in zero page
              788     ;
F53E A96C      789     LDA   #$6C
F540 8571      790     STA   JMPCSP
F542 856E      791     STA   JMPPC
F544 AD1CF2    792     LDA   APCT+1
F547 8570      793     STA   JMPPC+2
F549 AD1AF2    794     LDA   ACSPT+1
F54C 8573      795     STA   JMPCSP+2
              796  endif
              797     ;
              798     ;
              799     ;
F54E A904      800     LDA   #$04
F550 8DE2BD    801     STA   ABDE2
              802  ifndef ORIG
              803  ;.PAGE 'Logic relocated from original version or not in original
version'
              804     ;
              805     ; Unconditional branch
              806     ;
              807     BNZ   BEGDA
              808     ;
              809     ; Compute total of 256 locations
              810     ;
              811  CKPAGE  LDA   #0
              812     TAX
              813     TAY
              814  CKNX   CLC

```

```

815     ADC     (AC5), Y
816     BNC     NOCRY
817     INX
818     NOCRY   INY
819     BNZ     CKNX
820     RTS
821     ;
822     ;
823     ;
824     CN05T   DB     $03, $18, $38, $48
825     CN07T   DB     $3C, $38, $18, $48
826     ;
827     ;
828     ;
829     JVECTS  DW     UDJMPV
830     DW     DSKNUM
831     DW     BIOS
832     DW     BIOSAF
833     ;
834     ; Zero code just executed
835     ;
836     BEGDA   LDX     #<A6000
837     STX     ABD+1
838     LDA     #>A6000
839     STA     ABD
840     TAY
841     CLRLP   CPY     #>CLRLP
842     BNE     BEGDD
843     CPX     #<CLRLP
844     BE     BEGDE
845     BEGDD   STA     (ABD), Y
846     INY
847     BNZ     CLRLP
848     INC     ABD+1
849     INX
850     BNZ     CLRLP
851     BEGDE
852     else
853     ;.PAGE 'Logic from original version'
854     ;
855     ;
856     ;
F553 AD21F2 857     LDA     AF221
F556 F00C   858     BZ     INVBG
F558 AD1FF2 859     LDA     AF21F
F55B 8DFCFF 860     STA     AFFFC
F55E AD20F2 861     LDA     AF21F+1
F561 8DFDFD 862     STA     AFFFC+1
863     INVBG
864     endi f
865     ;
866     ;
867     ;
F564 4C53D2 868     JMP     EXECPC
F567 00000000 73     DS     9
74     INCLUDE   PADV11.IAS
1     ;

```

```

2 ;
3 ;
4  ifndef IIc
F570 70      5      DB      $70
6      el se
7      DB      $38
8      endi f
F571 25000002 9      DB      $25, $00, $00, $02, $00, $01, $01
F578 A008    10     LDY      #$08
F57A 203CFF  11     JSR      JUDDRV
F57D 18      12     CLC
F57E 9003    13     BNC      I NVBH
F580 2012FF  14     JSR      JDREAD
F583 A24E    15     I NVBH   LDX      #78
F585 18      16     CLC
F586 A580    17     LDA      A80
F588 6906    18     ADC      #6
F58A 8580    19     STA      A80
F58C 900D    20     BNC      I NVBJ
F58E E681    21     INC      A80+1
F590 18      22     I NVBI   CLC
F591 A580    23     LDA      A80
F593 691A    24     ADC      #26
F595 8580    25     STA      A80
F597 9002    26     BNC      I NVBJ
F599 E681    27     INC      A80+1
F59B CA      28     I NVBJ   DEX
F59C D053    29     BNZ      I NVBP
F59E A900    30     LDA      #0
F5A0 8597    31     STA      A97
F5A2 8598    32     STA      A98
F5A4 ADABEE  33     LDA      I NULL+1
F5A7 48      34     PHA
F5A8 ADA AEE  35     LDA      I NULL
F5AB 48      36     PHA
F5AC AD18F2  37     LDA      ADRI OR+1
F5AF 48      38     PHA
F5B0 AD17F2  39     LDA      ADRI OR
F5B3 48      40     PHA
F5B4 2006FF  41     JSR      JCI NIT
F5B7 A000    42     LDY      #0
F5B9 8C22F2  43     STY      AF222
F5BC B931F2  44     I NVBL   LDA      I NUPQ, Y
F5BF C923    45     CMP      #$23
F5C1 D022    46     BNE      I NVB0
F5C3 AD21F2  47     LDA      AF221
F5C6 DOFE    48     I NVBM   BNZ      I NVBM
F5C8 ADFCFF  49     LDA      AFFF C
F5CB 8D1FF2  50     STA      AF21F
F5CE ADFDFF  51     LDA      AFFF C+1
F5D1 8D20F2  52     STA      AF21F+1
F5D4 ADF8FF  53     LDA      AFFF8
F5D7 8DFCFF  54     STA      AFFF C
F5DA ADF9FF  55     LDA      AFFF8+1
F5DD 8DFDFF  56     STA      AFFF C+1
F5E0 EE21F2  57     INC      AF221
F5E3 DOFE    58     I NVBN   BNZ      I NVBN

```

```

F5E5 2003FF      59  INVBO   JSR   JCWRT
F5E8 EE22F2      60      INC   AF222
F5EB AC22F2      61      LDY   AF222
F5EE 18          62      CLC
F5EF 90CB        63      BNC   INVBL
F5F1 AC23F2      64  INVBP   LDY   SYSPK
F5F4 B180        65  INVBQ   LDA   (A80), Y
F5F6 D923F2      66      CMP   SYSPK, Y
F5F9 D095        67      BNE   INVBI
F5FB 88          68      DEY
F5FC 10F6        69      BNM   INVBQ
F5FE A580        70      LDA   A80
71      ifdef IIc
72      SEC
73      SBC   #$06
74      STA   $80
75      BC   LF5D1
76      DEC   $80+1
77  LF5D1      LDY   #0
78      LDA   ($80), Y
79      STA   $7A
80      INY
81      LDA   ($80), Y
82      STA   $7A+1
83      LDA   #$00
84      PHA
85      PHA
86      PHA
87      LDA   $FEB7
88      BZ   LF5EA
89      LDA   #$04
90      BNM   LF5ED
91  LF5EA      LDA   $FEB6
92  LF5ED      PHA
93      LDA   #$60
94      PHA
95      LDA   #$00
96      PHA
97      LDA   #$01
98      PHA
99      LDA   #$00
100     PHA
101     LDA   $7A+1
102     PHA
103     LDA   $7A
104     PHA
105     endi f
106     ;
107     ;
108     ;
F600 54484953    109     ASC   ' TH I S I S I T '
F608 0B0001      110     DB   11, 0, 1
F60B 00000000    111     DS   5
F610 54484953    112     ASC   ' TH I S I S I T '
F618 060001      113     DB   6, 0, 1
F61B 00000000    114     DS   5
F620 20202020    115     ASC   ' '

```

```

F628 00000000    75    DS    2136
                76    INCLUDE    IOHI.IAS
                1    ;. PAGE 'Tables'
                2    ;
                3    ;
                4    ;
FE80 4C0000     5    UDJMPV    JMP    A00
FE83 4C0000     6    JMP    A00
FE86 4C0000     7    JMP    A00
FE89 4C0000     8    JMP    A00
FE8C 4C0000     9    JMP    A00
FE8F 4C0000    10    JMP    A00
FE92 4C0000    11    JMP    A00
FE95 4C0000    12    JMP    A00
FE98 4C0000    13    JMP    A00
FE9B 4C0000    14    JMP    A00
FE9E 4C0000    15    JMP    A00
FEA1 4C0000    16    JMP    A00
FEA4 4C0000    17    JMP    A00
FEA7 4C0000    18    JMP    A00
FEAA 4C0000    19    JMP    A00
FEAD 4C0000    20    JMP    A00
                21    ;
                22    ;
                23    ;
FEBO FFFF      24    DSKNUM    DW    -1    ; CONSOLE
FEB2 FFFF      25    DW    -1    ; SYSTEM
FEB4 FFFF      26    DW    -1    ; not used
FEB6 0000      27    DW    0    ; Slot 6, drive 1
FEB8 0100      28    DW    1    ; Slot 6, drive 2
FEBA FFFF      29    DW    -1    ; PRINTER
FEBC FFFF      30    DW    -1    ; REMIN
FEBE FFFF      31    DW    -1    ; REMOUT
FECO 0400      32    DW    4    ; Slot 4, drive 1
FEC2 0500      33    DW    5    ; Slot 4, drive 2
FEC4 0200      34    DW    2    ; Slot 5, drive 1
FEC6 0300      35    DW    3    ; Slot 5, drive 2
                36    ifndef ORIG
                37    DW    -1
                38    DW    -1
                39    DW    -1
                40    DW    -1
                41    DW    -1
                42    DW    -1
                43    DW    -1
                44    DW    -1
                45    endif
                46    ;. PAGE 'Go to appropriate user defined driver'
                47    ;
                48    ;
                49    ;
FEC8 85D2     50    URWIS    STA    AD2
                51    ifndef ORIG
                52    ASL    A
                53    ASL    A
                54    SBB    AD2
                55    TAY

```

```

56     LDA    #<DSKNUM
57     PHA
58     TYA
59     JMP    COMN
60     else
FECA 297F 61     AND    #$7F
FECC 85D3 62     STA    AD3
FECE 0A    63     ASL    A
FECF 18    64     CLC
FEDO 65D3 65     ADC    AD3
FED2 6D96D6 66     ADC    JVECTS
FED5 85D3 67     STA    AD3
FED7 A900 68     LDA    #0
FED9 6D97D6 69     ADC    JVECTS+1
FEDC 85D4 70     STA    AD3+1
FEDE A5D2 71     LDA    AD2
FEEO 6CD300 72     JMP    (AD3)
73     endif
74     ;.PAGE 'Odds and ends'
75     ifdef ORIG
76     ;
77     ;   Read and write RAM; use bank #1
78     ;
FEE3 AD8BC0 79     TOBRK    LDA    LCBNK1
80     ;
81     ;
82     ;
FEE6 6C16BF 83     JMP    (BREAK)
84
*****
85     ;
86     ;   Read and write RAM; use bank #2
87     ;
FEE9 AD83C0 88     NOTOST    LDA    LCBNK2
89     ;
90     ;
91     ;
FEEC 4C9ED6 92     JMP    CLRMEM
93     endif
94
*****
95     ;
96     ;   Read and write RAM; use bank #1
97     ;
FEEF AD8BC0 98     AFEEF    LDA    LCBNK1
99     ;
100    ;
101    ;
FEF2 6C2FBF 102    JMP    (XI TLOC)
103
*****
104    ;
105    ;   Read and write RAM; use bank #1
106    ;
FEF5 AD8BC0 107    OST      LDA    LCBNK1
108    ;
109    ;   Go to a jump to 'MINITC' (Move INI Tialization Code)

```

```

110 ;
FEF8 4C52D1 111     JMP     JMOVIC           ; INI TO
112     ifndef ORIG
113
*****
114 ;
115 ;   Read and write RAM; use bank #1
116 ;
117     TOBRK     LDA     LCBNK1
118 ;
119 ;
120 ;
121     JMP     (BREAK)
122 ;
123 ;
124 ;
125     DS     8
126     else
127
*****
128 ;
129 ;
130 ;
FEFB A980 131     DOH12     LDA     #$80
FEFD 85EO 132     STA     AEO
133 ;
134 ;
135 ;
FEFF 60 136     RTS
137     endif
138 ;. PAGE 'Main BIOS jump table'
139 ;
140 ;
141 ;
142     BIOS
FF00 209EFF 143     JCREAD     JSR     SAVRET           ; Console read
FF03 209EFF 144     JCWRT     JSR     SAVRET           ; Console write
FF06 209EFF 145     JCI NIT   JSR     SAVRET           ; Console init
FF09 209EFF 146     JPRTW     JSR     SAVRET           ; Printer write
FF0C 209EFF 147     JPRTI     JSR     SAVRET           ; Printer init
FF0F 209EFF 148     JDWRT     JSR     SAVRET           ; Disk write
FF12 209EFF 149     JDREAD     JSR     SAVRET           ; Disk read
FF15 209EFF 150     JDI NIT   JSR     SAVRET           ; Disk init
FF18 209EFF 151     JREMR     JSR     SAVRET           ; Remote read
FF1B 209EFF 152     JREMW     JSR     SAVRET           ; Remote write
FF1E 209EFF 153     JREMI     JSR     SAVRET           ; Remote init
FF21 209EFF 154     JGRAFW     JSR     SAVRET           ; Graphic write
FF24 209EFF 155     JGRAFI     JSR     SAVRET           ; Graphic init
FF27 209EFF 156     JSR     SAVRET           ; Printer read
FF2A 209EFF 157     JCONS     JSR     SAVRET           ; Console status
FF2D 209EFF 158     JPRTS     JSR     SAVRET           ; Printer status
FF30 209EFF 159     JDSKS     JSR     SAVRET           ; Disk status
FF33 209EFF 160     JREMS     JSR     SAVRET           ; Remote status
FF36 209EFF 161     AFF36     JSR     SAVRET           ; To get to CONCK from CONCKVEC
FF39 209EFF 162     JUSER     JSR     SAVRET           ; User read, write, init, status
FF3C 209EFF 163     JUDDRV     JSR     SAVRET           ; Substitute disk driver
164     ifdef ORIG

```

```

165 AFF3F
FF3F 209EFF 166 JTREE JSR SAVRET ; ID search
167 endif
168 ;.PAGE 'Go to address of substitute disk driver'
169 ;
170 ; Save 'A' register
171 ;
FF42 85D2 172 PSUBDR STA AD2
173 ;
174 ; Push address from table onto stack
175 ;
FF44 B9AFFE 176 LDA DSKNUM-1, Y
FF47 48 177 PHA
FF48 B9AEFE 178 LDA DSKNUM-2, Y
FF4B 48 179 COMN PHA
180 ;
181 ; Restore 'A' register
182 ;
FF4C A5D2 183 LDA AD2
184 ;
185 ; Go to address pushed onto stack
186 ;
FF4E 60 187 RTS
188 ;.PAGE 'Odds and ends'
189 ifndef ORIG
190 ;
191 ;
192 ;
193 DS 6
194 ;
195 ;
196 ;
197 NFF52 DW DOFF
198 DW INVERT
199 DW INTAG
200 else
201 ;
202 ;
203 ;
FF4F A980 204 DOH14 LDA #$80
FF51 85E1 205 STA AE1
FF53 0A 206 ASL A
FF54 85E0 207 STA AEO
FF56 60 208 RTS
209 ;
210 ;
211 ;
FF57 00 212 DS 1
213 endif
214 ;
215 ;
216 ;
FF58 60 217 INWAE RTS
218 ;.PAGE 'Offset jump table'
219 ;
220 ;
221 ;

```



```

FF59 4C72D7    222 JCONCK  JMP  CONCK
                223 ;
                224 ; Console support
                225 ;
FF5C 4CC6D8    226 BIOSAF  JMP  CREAD
FF5F 4C50D9    227      JMP  CWRI TE
FF62 4C98D8    228      JMP  CI NI T
                229 ;
                230 ; Printer support
                231 ;
FF65 4CC3D9    232      JMP  PWRI TE
FF68 4CEFD8    233      JMP  PI NI T
                234 ;
                235 ; Disk support
                236 ;
FF6B 4C28D0    237      JMP  DWRI TE
FF6E 4C2CDO    238      JMP  DREAD
FF71 4C83D6    239      JMP  DI NI T
                240 ;
                241 ; Remote support
                242 ;
FF74 4CE5D9    243      JMP  RREAD
FF77 4C9CD9    244      JMP  RWRI TE
FF7A 4C1CD9    245      JMP  RI NI T
                246 ;
                247 ;
                248 ;
FF7D 4C58FF    249      JMP  INWAE
FF80 4C18D9    250      JMP  GI NI T
                251 ;
                252 ;
                253 ;
FF83 4C58FF    254      JMP  INWAE
                255 ;
                256 ;
                257 ;
FF86 4CB0DC    258      JMP  STATC
FF89 4CA0DC    259      JMP  STATPR
FF8C 4CC5DC    260      JMP  STATD
                261 ifdef ORIG
FF8F 4CA0DC    262      JMP  STATPR
                263 else
                264      JMP  STATR
                265 endif
                266 ;
                267 ;
                268 ;
FF92 4C72D7    269      JMP  CONCK
FF95 4CC8FE    270      JMP  URWIS
FF98 4C42FF    271      JMP  PSUBDR
                272 ifdef ORIG
                273 ifndef IIc
FF9B 4CE4DC    274      JMP  ADCE4
                275 else
                276      JMP  INWAE
                277 endif
                278 endif

```

```

279 ;. PAGE ' Switch between main logic and I/O logic'
280 ;
281 ;
282 ;
FF9E 85D2 283 SAVRET STA AD2
284 ;
285 ;
286 ;
FFA0 68 287 PLA
FFA1 18 288 CLC
FFA2 695A 289 ADC #JCONCK-BIOS+1
FFA4 85D3 290 STA AD3
FFA6 68 291 PLA
FFA7 6900 292 ADC #0
FFA9 85D4 293 STA AD3+1
294 ;
295 ;
296 ;
FFAB 68 297 PLA
FFAC 8D1ABF 298 STA ABF1A
FFAF 68 299 PLA
FFB0 8D1BBF 300 STA ABF1A+1
301 ;
302 ; Read and write RAM; use bank #2
303 ;
FFB3 AD83C0 304 LDA LCBNK2
305 ;
306 ;
307 ;
FFB6 A5D2 308 LDA AD2
309 ;
310 ;
311 ;
FFB8 20CBFF 312 JSR INWBC
313 ;
314 ;
315 ;
FFBB 85D2 316 AFFBB STA AD2
317 ;
318 ;
319 ;
FFBD AD1BBF 320 LDA ABF1A+1
FFC0 48 321 PHA
FFC1 AD1ABF 322 LDA ABF1A
FFC4 48 323 PHA
324 ;
325 ; Read and write RAM; use bank #1
326 ;
FFC5 AD8BC0 327 LDA LCBNK1
328 ;
329 ;
330 ;
FFC8 A5D2 331 LDA AD2
332 ;
333 ;
334 ;
FFCA 60 335 RTS

```

```

336 ;
337 ;
338 ;
FFCB 6CD300 339 INWBC    JMP    (AD3)
340 ;.PAGE 'More odds and ends'
341  ifdef ORIG
342 ;
343 ;
344 ;
FFCE 5D5E40 345 XLTIN    ASC    ' ] ^@'
FFD1 1D1E00 346     DB    $1D, $1E, $00
347 ;
348 ;
349 ;
FFD4 4D4E505D 350 XLTOUT   ASC    ' MNP] ^@'
351     endi f
352     i fndef II c
353 ;
354 ;
355 ;
FFDA AD11BF 356 HTAB     LDA    NBF11
FFDD C914 357     CMP    #20
FFDF B004 358     BGE   TAB3
359 ;
360 ;
361 ;
FFE1 A928 362 AFFE1    LDA    #40
FFE3 D002 363     BNZ   INWBE
FFE5 A900 364 TAB3     LDA    #$00
FFE7 38 365 INWBE    CLB
FFE8 ED11BF 366     SBB   NBF11
FFEB 4CFEDB 367     JMP   INTNR
368     el se
369     DB    0, 0, 0, 0, 0, 0, 0, 0
370     DB    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
371     endi f
372     i fndef ORIG
373 ;
374 ;
375 ;
376     DS    5
377 ;
378 ;
379 ;
380 NFFE2    DW    GENIT
381 ;
382 ;
383 ;
384 XLTIN    DB    SOB, SOC, SOF
385     ASC   ' ] ^@'
386     DB    $1D, $1E, $00
387 ;
388 ;
389 ;
390 XLTOUT   ASC    '['
391     DB    $0A, $0B
392     ASC   ' MNP] ^@'

```

```

393 ;
394 ;
395 ;
396     DW      4
397     el se
398 ;
399 ;
400 ;
FFEE 00000000 401     DS      10
402     endi f
403 ;
404 ;
405 ;
406     AFFF8
407     i fndef  ORIG
408     DW      OST
409     el se
FFF8 E9FE     410     DW      NOTOST
411     endi f
FFFA EFFE     412     DW      AFEEF
FFFC EFFE     413     AFFFC   DW      AFEEF
FFFE EFFE     414     DW      AFEEF
77 ;
78 ;
79 ;
80     ORG     $D000
81     INCLUDE V111OP1.IAS
1 ;
2 ;
3 ;
D000 00       4     SVFLGS  DB      0
5 ;
6 ;   Save CPU flags
7 ;
8 *INTAD
D001 08       9     CCONCK  PHP
10 ;
11 ;   Save 'A' register
12 ;
D002 48       13     PHA
14 ;
15 ;   Set CPU flags from 'SVFLGS'
16 ;
D003 AD00D0   17     LDA     SVFLGS
D006 48       18     PHA
D007 28       19     PLP
20 ;
21 ;
22 ;
D008 EA       23     NOP
24 ;
25 ;   Inhibit interrupts
26 ;
D009 78       27     SEI
28 ;
29 ;
30 ;

```

```

DO0A 2072D7    31    JSR    CONCK
                32    ;
                33    ; Restore 'A' register
                34    ;
DO0D 68        35    PLA
                36    ;
                37    ; Restore CPU flags
                38    ;
DO0E 28        39    PLP
                40    ;
                41    ; Return to caller
                42    ;
DO0F 60        43    RTS
                44
*****
                45    ;
                46    ;
                47    ;
DO10 ADEACO    48    INTAG    LDA    AC08A+$60
                49    ;
                50    ;
                51    ;
                52    ;
DO13 A000      53    LDY    #0
DO15 8CA303    54    STY    A03A3
DO18 8C7E04    55    STY    A0478+6
DO1B 8CFE04    56    STY    A04F8+6
                57    ;
                58    ;
                59    ;
DO1E C8        60    INY
DO1F 8CB003    61    STY    PRVDRV
                62    ;
                63    ;
                64    ;
DO22 A960      65    LDA    #$60
DO24 8DAF03    66    STA    PRVSLT
                67    ;
                68    ; Return to caller
                69    ;
DO27 60        70    RTS
                71    ;.PAGE 'Disk I/O'
                72    ;
                73    ;
                74    ;
DO28 A002      75    DWRI TE    LDY    #2
DO2A D002      76    BNZ    INTAH
DO2C A001      77    DREAD    LDY    #1
DO2E 8CAC03    78    INTAH    STY    RDWRFL
                79    ;
                80    ; Save return address
                81    ;
DO31 68        82    PLA
DO32 85C0      83    STA    RETADR
DO34 68        84    PLA
DO35 85C1      85    STA    RETADR+1
                86    ;

```

```

      87 ;
      88 ;
D037 68      89   PLA
D038 85BE    90   STA   ABD+1
D03A 0A      91   ASL   A
D03B 290F    92   AND   #SOF
D03D 85BD    93   STA   ABD
D03F 68      94   PLA
D040 4A      95   LSR   A
D041 85C2    96   STA   AC2
D043 A5BE    97   LDA   ABD+1
D045 6A      98   ROR   A
D046 4A      99   LSR   A
D047 4A     100   LSR   A
D048 8DA403 101   STA   A03A4
      102 ;
      103 ;
      104 ;
D04B 68     105   PLA
D04C 85BE    106   STA   ABD+1
D04E C901    107   CMP   #S01
D050 68     108   PLA
D051 6900    109   ADC   #0
D053 88     110   DEY
D054 F003    111   BZ    INTAI
D056 88     112   DEY
D057 84BE    113   STY   ABD+1
D059 85BF    114   INTAI STA   ABF
      115 ;
      116 ;
      117 ;
D05B 68     118   PLA
D05C 8DA803 119   STA   XFRADR
D05F 85C4    120   STA   AC4
D061 68     121   PLA
D062 8DA903 122   STA   XFRADR+1
      123 ;
      124 ; Convert drive code to slot and drive numbers
      125 ;
D065 68     126   PLA
D066 4A     127   LSR   A
D067 A8     128   TAY
D068 A901    129   LDA   #1
D06A 6900    130   ADC   #0
D06C 8DA203 131   STA   DRVNO
D06F B980D6 132   LDA   SLTTBL, Y
D072 8DA103 133   STA   SLOTNO
      134 ;
      135 ;
      136 ;
D075 68     137   PLA
D076 AA     138   TAX
D077 68     139   PLA
D078 68     140   PLA
D079 8A     141   TXA
D07A A208    142   LDX   #S08
D07C 05C2    143   ORA   AC2

```

```

D07E D061      144      BNZ   INTAP
                145      ;
                146      ;
                147      ;
D080 98        148      TYA
D081 2084D6    149      JSR   INTGN
D084 D05B      150      BNZ   INTAP
                151      ;
                152      ;
                153      ;
D086 A5BF      154      LDA   ABF
D088 F055      155      BZ    INTAN
                156      ;
                157      ;   Clear $DO. . $DF
                158      ;
D08A A00F      159      INTAJ  LDY   #15
D08C A900      160      LDA   #0
D08E 99D000    161      INTAK  STA   ADO, Y
D091 88        162      DEY
D092 10FA      163      BNM   INTAK
                164      ;
                165      ;
                166      ;
D094 A910      167      LDA   #$10
D096 38        168      CLB
D097 E5BD      169      SBB   ABD
D099 C5BF      170      CMP   ABF
D09B 9002      171      BL    INTAL
D09D A5BF      172      LDA   ABF
D09F 85C6      173      INTAL  STA   AC5+1
DOA1 85C3      174      STA   AC3
                175      ;
                176      ;
                177      ;
DOA3 08        178      PHP
DOA4 68        179      PLA
DOA5 2904      180      AND   #$04
DOA7 8D00D0    181      STA   SVFLGS
                182      ;
                183      ;
                184      ;
DOAA 78        185      SEI
                186      ;
                187      ;
                188      ;
DOAB 20E8D0    189      JSR   INTAS
                190      ;
                191      ;
                192      ;
DOAE 08        193      PHP
DOAF 68        194      PLA
DOBO 0D00D0    195      ORA   SVFLGS
DOB3 48        196      PHA
DOB4 28        197      PLP
                198      ;
                199      ;
                200      ;

```

```

DOB5 AEAD03      201      LDX  A03AD
DOB8 B027        202      BC   INTAP
                203      ;
                204      ;
                205      ;
DOBA A900        206      LDA  #0
DOBC 85BD        207      STA  ABD
                208      ;
                209      ;
                210      ;
DOBE EEA403      211      INC  A03A4
                212      ;
                213      ;
                214      ;
DOC1 ADA903      215      LDA  XFRADR+1
DOC4 65C6        216      ADC  AC5+1
DOC6 8DA903      217      STA  XFRADR+1
                218      ;
                219      ;
                220      ;
DOC9 A5BF        221      LDA  ABF
DOCB 38          222      CLB
DOCC E5C6        223      SBB  AC5+1
DOCE 85BF        224      STA  ABF
                225      ;
                226      ;
                227      ;
DODO DOB8        228      BNZ  INTAJ
                229      ;
                230      ;
                231      ;
DOD2 A4BE        232      LDY  ABD+1
DOD4 F009        233      BZ   INTAN
                234      ;
                235      ;
                236      ;
DOD6 88          237      INTAM  DEY
                238      ;
                239      ;
                240      ;
DOD7 B90002      241      LDA  A0200, Y
DODA 91C4        242      STA  (AC4), Y
                243      ;
                244      ;
                245      ;
DODC 98          246      TYA
DODD DOF7        247      BNZ  INTAM
                248      ;
                249      ;
                250      ;
DODF A200        251      INTAN  LDX  #0
                252      ;
                253      ;   Push saved return address on stack
                254      ;
DOE1 A5C1        255      INTAP  LDA  RETADR+1
DOE3 48          256      PHA
DOE4 A5C0        257      LDA  RETADR

```



```

DOE6 48      258      PHA
              259      ;
              260      ; Return to caller
              261      ;
DOE7 60      262      RTS
              82      INCLUDE      DISKII.IAS
              1      ; PAGE '???'
              2      ;
              3      ;
              4      ;
DOE8 A002    5      INTAS      LDY      #$02
DOEA 8CFFDF  6      STY      AFFFF
              7      ;
              8      ;
              9      ;
DOED A00A    10     LDY      #$0A
DOEF 8CFDDF  11     STY      AFFFF
              12     ;
              13     ; Branch if slot number hasn't changed
              14     ;
DOF2 AEA103  15     LDX      SLOTNO
DOF5 ECAF03  16     CPX      PRVSLT
DOF8 F011    17     BE      INTAU
              18     ;
              19     ;
              20     ;
DOFA AEAFO3  21     LDX      PRVSLT
              22     ifndef ORIG
              23     BZ      INTAZ
              24     endif
              25     ;
              26     ;
              27     ;
DOFD 2001D0  28     INTAT      JSR      CCONCK      ; INTAD
              29     ;
              30     ;
              31     ;
D100 2082D3  32     JSR      INTDV
D103 DOF8    33     BNZ      INTAT
              34     ;
              35     ;
              36     ;
              37     ifndef ORIG
              38     INTAZ
              39     endif
D105 AEA103  40     LDX      SLOTNO
D108 8EAF03  41     STX      PRVSLT
              42     ;
              43     ;
              44     ;
D10B 2082D3  45     INTAU      JSR      INTDV
D10E 08      46     PHP
              47     ;
              48     ; Turn drive on
              49     ;
D10F BD89C0  50     LDA      DRVON, X
              51     ;

```

```

52 ;
53 ;
D112 ADA203 54 LDA DRVNO
D115 CDB003 55 CMP PRVDRV
D118 F007 56 BE INTAV
57 ;
58 ;
59 ;
D11A 8DB003 60 STA PRVDRV
61 ;
62 ;
63 ;
D11D 28 64 PLP
D11E A000 65 LDY #$00
D120 08 66 PHP
67 ;
68 ;
69 ;
D121 2901 70 INTAV AND #$01
D123 4901 71 EOR #$01
D125 ODA103 72 ORA SLOTNO
D128 A8 73 TAY
D129 D98AC0 74 CMP DRVSL1, Y
D12C 4A 75 LSR A
D12D 08 76 PHP
D12E 4A 77 LSR A
D12F 4A 78 LSR A
D130 28 79 PLP
D131 6A 80 ROR A
D132 85C7 81 STA AC7
82 ;
83 ;
84 ;
D134 A9EF 85 LDA #>$D8EF
D136 8546 86 STA A46
D138 A9D8 87 LDA #<$D8EF
D13A 8547 88 STA A46+1
89 ;
90 ;
91 ;
D13C 28 92 PLP
D13D 08 93 PHP
D13E D00B 94 BNZ INTAX
95 ;
96 ;
97 ;
D140 A008 98 LDY #8
99 INTAW
100 ifdef ORIG
D142 2037D6 101 JSR DELAY
102 else
103 JSR DLAYCC
104 endif
D145 88 105 DEY
D146 DOFA 106 BNZ INTAW
107 ;
108 ;

```

```

109 ;
D148 AEA103 110 LDX SLOTNO
111 ;
112 ;
113 ;
D14B ADA403 114 INTAX LDA A03A4
D14E 206BD2 115 JSR INTCH
116 ;
117 ;
118 ;
D151 28 119 PLP
D152 D020 120 BNZ INTBC
121 ;
122 ; Branch if reading
123 ;
D154 ADAC03 124 LDA RDWRFL
D157 C902 125 CMP #2
D159 D010 126 BNE INTBB
127 ;
128 ;
129 ;
D15B A012 130 INTAY LDY #18
131 ;
132 ;
133 ;
D15D 88 134 INTBA DEY
D15E DOFD 135 BNZ INTBA
136 ;
137 ;
138 ;
D160 E646 139 INC A46
D162 DOF7 140 BNZ INTAY
D164 2001D0 141 JSR CCONCK ; INTAD
D167 E647 142 INC A46+1
D169 DOFO 143 BNZ INTAY
144 ;
145 ;
146 ;
D16B 2082D3 147 INTBB JSR INTDV
D16E D004 148 BNZ INTBC
149 ;
150 ;
151 ;
D170 A909 152 LDA #$09
D172 D058 153 BNZ INTBN
154 ;
155 ;
156 ;
D174 ADAC03 157 INTBC LDA RDWRFL
D177 6A 158 ROR A
D178 08 159 PHP
160 ;
161 ;
162 ;
D179 A9FF 163 LDA #$FF
D17B 8DA503 164 STA A03A5
165 ;

```

```

166 ;
167 ;
D17E A080 168 INTBE LDY #$80
D180 8CFEDF 169 STY ADFFE
170 ;
171 ;
172 ;
D183 CEFEDF 173 INTBF DEC ADFFE
D186 3020 174 BM INTBJ
175 ;
176 ;
177 ;
D188 2001D0 178 JSR CCONCK ; INTAD
179 ;
180 ; Read address field
181 ;
D18B AEA103 182 LDX SLOTNO
D18E 2024D3 183 JSR READAF
D191 BOFO 184 BC INTBF
185 ;
186 ;
187 ;
D193 A5CC 188 LDA ACTTRK
D195 CDFBDF 189 CMP CURTRK
D198 FO38 190 BE INTBO
191 ;
192 ;
193 ;
D19A A4C7 194 LDY AC7
D19C 0A 195 ASL A
D19D 997804 196 STA A0478, Y
197 ;
198 ;
199 ;
D1A0 ADFBDF 200 LDA CURTRK
D1A3 CEFDDF 201 DEC ADFFD
D1A6 D01B 202 BNZ INTBK
203 ;
204 ;
205 ;
D1A8 CEFFDF 206 INTBJ DEC ADFFF
D1AB FO1C 207 BZ INTBL
208 ;
209 ;
210 ;
D1AD ADFBDF 211 LDA CURTRK
D1B0 48 212 PHA
213 ;
214 ;
215 ;
D1B1 A90A 216 LDA #$0A
D1B3 8DFDDF 217 STA ADFFD
218 ;
219 ;
220 ;
D1B6 A960 221 LDA #$60
D1B8 A4C7 222 LDY AC7

```

```

D1BA 997804      223      STA   A0478, Y
                  224      ;
                  225      ;
                  226      ;
D1BD A900        227      LDA   #S00
D1BF 206BD2      228      JSR   INTCH
                  229      ;
                  230      ;
                  231      ;
D1C2 68          232      PLA
                  233      ;
                  234      ;
                  235      ;
D1C3 206BD2      236      INTBK   JSR   INTCH
                  237      ifdef ORIG
D1C6 4C7ED1      238      JMP   INTBE
                  239      else
                  240      BNM   INTBE
                  241      endif
                  242      ;
                  243      ;
                  244      ;
D1C9 A940        245      INTBL   LDA   #S40
D1CB 28          246      INTBM   PLP
                  247      ;
                  248      ;
                  249      ;
D1CC 4C59D2      250      INTBN   JMP   WRNG
                  251      ;
                  252      ;
                  253      ;
                  254      ifdef ORIG
D1CF 4C57D2      255      JMP   WROK
                  256      endif
                  257      ;
                  258      ;
                  259      ;
D1D2 ADA303      260      INTBO   LDA   A03A3
D1D5 48          261      PHA
                  262      ;
                  263      ;
                  264      ;
D1D6 A5CD        265      LDA   ACTVOL
D1D8 8DAE03      266      STA   A03AE
                  267      ;
                  268      ;
                  269      ;
D1DB 68          270      PLA
D1DC F008        271      BZ    INTBP
                  272      ;
                  273      ;
                  274      ;
D1DE C5CD        275      CMP   ACTVOL
D1E0 F004        276      BE   INTBP
                  277      ;
                  278      ;
                  279      ;

```

```

D1E2 A920      280      LDA   #$20
D1E4 DOE5      281      BNZ   INTBM
                282      ;
                283      ;
                284      ;
D1E6 28        285      INTBP   PLP
D1E7 08        286      PHP
D1E8 B013      287      BC     INTBR
                288      ;
                289      ;
                290      ;
D1EA A5CB      291      LDA   ACTSEC
D1EC CDA503    292      CMP   A03A5
D1EF F070      293      BE    WRSEC
                294      ;
                295      ;
                296      ;
D1F1 18        297      CLC
D1F2 6901      298      ADC   #$01
D1F4 290F      299      AND   #$0F
D1F6 CDA503    300      CMP   A03A5
D1F9 F088      301      BE    INTBF
                302      ;
                303      ;
                304      ;
D1FB 85CB      305      STA   ACTSEC
                306      ;
                307      ;
                308      ;
D1FD A4CB      309      INTBR   LDY   ACTSEC
D1FF B9D000    310      LDA   ADO, Y
D202 F003      311      BZ    INTBS
                312      ;
                313      ;
                314      ;
D204 4C83D1    315      JMP   INTBF
                316      ;
                317      ;
                318      ;
D207 98        319      INTBS   TYA
D208 4A        320      LSR   A
D209 9002      321      BNC  INTBT
                322      ;
                323      ;
                324      ;
D20B 0908      325      ORA   #$08
                326      ;
                327      ;
                328      ;
D20D 38        329      INTBT   CLB
D20E E5BD      330      SBB   ABD
D210 902E      331      BB    INTBX
                332      ;
                333      ;
                334      ;
D212 A8        335      TAY
D213 18        336      CLC

```

```

D214 6DA903    337    ADC    XFRADR+1
D217 C8        338    INY
D218 C4BF      339    CPY    ABF
D21A 900E      340    BL     INTBU
D21C D022      341    BNE    INTBX
D21E A4BE      342    LDY    ABD+1
D220 F008      343    BZ     INTBU
D222 85C5      344    STA    AC5
                345    ;
                346    ;
                347    ;
D224 A000      348    LDY    #$00
D226 A902      349    LDA    #$02
D228 B003      350    BC     INTBW
D22A ACA803    351    INTBU  LDY    XFRADR
D22D 843E      352    INTBW  STY    A3E
D22F 853F      353    STA    A3E+1
D231 28        354    PLP
D232 08        355    PHP
D233 B00E      356    BC     INTCB
D235 2058D6    357    JSR    INTGI
D238 AEA103    358    LDX    SLOTNO
D23B A5CB      359    LDA    ACTSEC
D23D 8DA503    360    STA    A03A5
D240 4C83D1    361    INTBX  JMP    INTBF
                362    ;
                363    ;
                364    ;
D243 2002D5    365    INTCB  JSR    READD
D246 B0F8      366    BC     INTBX
                367    ;
                368    ;
                369    ;
D248 A4CB      370    INTCC  LDY    ACTSEC
D24A A9FF      371    LDA    #$FF
D24C 99D000    372    STA    ADO, Y
D24F 8DA503    373    STA    A03A5
                374    ;
                375    ;
                376    ;
D252 C6C3      377    DEC    AC3
D254 DOEA      378    BNZ    INTBX
                379    ;
                380    ; Fix stack pointer
                381    ;
D256 28        382    PLP
                383    ;
                384    ;
                385    ;
D257 18        386    WROK   CLC
D258 24        387    DB     $24
D259 38        388    WRNG   SEC
D25A 8DAD03    389    STA    A03AD
                390    ;
                391    ; Turn off motor
                392    ;
D25D BD88C0    393    LDA    DRVOFF, X

```

```

394 ;
395 ; Return to caller
396 ;
D260 60 397 RTS
398 ;
399 ; Write data field
400 ;
D261 208CD2 401 WRSEC JSR WRDF
402 ;
403 ; Branch if good write
404 ;
D264 90E2 405 BNC INTCC
406 ;
407 ; Fix stack pointer
408 ;
D266 68 409 PLA
410 ;
411 ; Indicate write protect error
412 ;
D267 A910 413 LDA #$10
414 ;
415 ; Unconditional branch
416 ;
D269 BOEE 417 BC WRNG
418
*****
419 ;
420 ;
421 ;
D26B 0A 422 INTCH ASL A
D26C 48 423 PHA
424 ;
425 ;
426 ;
D26D BD80C0 427 LDA DRVSM0, X
D270 BD82C0 428 LDA DRVSM2, X
D273 BD84C0 429 LDA DRVSM4, X
D276 BD86C0 430 LDA DRVSM6, X
431 ;
432 ;
433 ;
D279 A4C7 434 LDY AC7
D27B B97804 435 LDA A0478, Y
D27E 8DFBDF 436 STA CURTRK
437 ;
438 ;
439 ;
D281 68 440 PLA
D282 997804 441 STA A0478, Y
442 ;
443 ;
444 ;
D285 20DAD5 445 JSR SEEK
446 ;
447 ;
448 ;
D288 4EFBDF 449 LSR CURTRK

```



```

450 ;
451 ; Return to caller
452 ;
D28B 60 453 RTS
454 ;.PAGE 'Write data field'
455 ;
456 ;
457 ;
D28C 38 458 WRDF SEC
459 ;
460 ;
461 ;
D28D 86CE 462 STX HSLOT1
D28F 8EFADF 463 STX HSLOT2
464 ;
465 ;
466 ;
D292 BD8DC0 467 LDA DRVWR, X
D295 BD8ECO 468 LDA DRVVRDM, X
D298 307C 469 BM WRDAD
470 ;
471 ;
472 ;
D29A AD0003 473 LDA BUFF2B
D29D 85CF 474 STA ACF
475 ;
476 ;
477 ;
D29F A9FF 478 LDA #$FF
D2A1 9D8FC0 479 STA DRVWRM, X
480 ;
481 ;
482 ;
D2A4 1D8CC0 483 ORA DRVVRD, X
D2A7 48 484 PHA
D2A8 68 485 PLA
D2A9 EA 486 NOP
487 ;
488 ; Write 4 autosync bytes
489 ;
D2AA A004 490 LDY #4
D2AC 48 491 AUTOLP PHA
D2AD 68 492 PLA
D2AE 201BD3 493 JSR WRCHE2
D2B1 88 494 DEY
D2B2 DOF8 495 BNZ AUTOLP
496 ;
497 ; Write 1st data field prologue byte
498 ;
D2B4 A9D5 499 LDA #$D5
D2B6 201AD3 500 JSR WRCHE1
501 ;
502 ; Write 2nd data field prologue byte
503 ;
D2B9 A9AA 504 LDA #$AA
D2BB 201AD3 505 JSR WRCHE1
506 ;

```

```

507 ; Write 3rd data field prologue byte
508 ;
D2BE A9AD 509 LDA #SAD
D2C0 201AD3 510 JSR WRCHE1
511 ;
512 ;
513 ;
D2C3 98 514 TYA
515 ;
516 ;
517 ;
D2C4 A056 518 LDY #86
D2C6 D003 519 BNZ WRDAB
520 ;
521 ;
522 ;
D2C8 B90003 523 WRDAA LDA BUFF2B, Y
D2CB 59FF02 524 WRDAB EOR BUFF2B-1, Y
D2CE AA 525 TAX
D2CF BD03D4 526 LDA WRXLTT, X
D2D2 A6CE 527 LDX HSL0T1
D2D4 9D8DC0 528 STA DRVWR, X
D2D7 BD8CC0 529 LDA DRVRD, X
530 ;
531 ;
532 ;
D2DA 88 533 DEY
D2DB DOEB 534 BNZ WRDAA
535 ;
536 ;
537 ;
D2DD A5CF 538 LDA ACF
D2DF EA 539 NOP
540 ;
541 ;
542 ;
D2E0 590002 543 WRDAC EOR BUFF6B, Y
D2E3 AA 544 TAX
D2E4 BD03D4 545 LDA WRXLTT, X
D2E7 AEFADF 546 LDX HSL0T2
D2EA 9D8DC0 547 STA DRVWR, X
D2ED BD8CC0 548 LDA DRVRD, X
D2F0 B90002 549 LDA BUFF6B, Y
550 ;
551 ;
552 ;
D2F3 C8 553 INY
D2F4 DOEA 554 BNZ WRDAC
555 ;
556 ; Write checksum
557 ;
D2F6 AA 558 TAX
D2F7 BD03D4 559 LDA WRXLTT, X
D2FA A6CE 560 LDX HSL0T1
D2FC 201DD3 561 JSR WRCHE3
562 ;
563 ; Write 1st data field epilogue byte

```

```

564 ;
D2FF A9DE 565 LDA #SDE
D301 201AD3 566 JSR WRCHE1
567 ;
568 ; Write 2nd data field epilogue byte
569 ;
D304 A9AA 570 LDA #SAA
D306 201AD3 571 JSR WRCHE1
572 ;
573 ; Write 3rd data field epilogue byte
574 ;
D309 A9EB 575 LDA #SEB
D30B 201AD3 576 JSR WRCHE1
577 ;
578 ; Write a gap byte
579 ;
D30E A9FF 580 LDA #SFF
D310 201AD3 581 JSR WRCHE1
582 ;
583 ;
584 ;
D313 BD8ECO 585 LDA DRVWDM, X
D316 BD8CCO 586 WRDAD LDA DRVWDM, X
D319 60 587 RTS
588
*****
589 ;
590 ;
591 ;
D31A 18 592 WRCHE1 CLC
593 * JSR WRCHE2
594 * RTS
595
*****
596 ;
597 ;
598 ;
D31B 48 599 WRCHE2 PHA
D31C 68 600 PLA
601 * JSR WRCHE3
602 * RTS
603
*****
604 ;
605 ;
606 ;
D31D 9D8DCO 607 WRCHE3 STA DRVWR, X
D320 1D8CCO 608 ORA DRVWR, X
D323 60 609 RTS
610 ;.PAGE 'Read address field'
611 ;
612 ;
613 ;
D324 A0FC 614 READAF LDY #-4
D326 84CF 615 STY RETRYC
616 ;
617 ;

```

```

618 ;
D328 C8      619  CHKTRY  INY
D329 D004    620      BNZ  WAITA1
              621 ;
              622 ;
              623 ;
D32B E6CF    624      INC  RETRYC
D32D F051    625      BZ   ERREX
              626 ;
              627 ;  Read a byte
              628 ;
D32F BD8CC0  629  WAITA1  LDA   DRV RD, X
D332 10FB    630      BNM  WAITA1
              631 ;
              632 ;  Branch if not 1st prologue byte
              633 ;
D334 C9D5    634  CHKP B1  CMP   #$D5
D336 DOF0    635      BNE  CHKTRY
              636 ;
              637 ;
              638 ;
D338 EA      639      NOP
              640 ;
              641 ;  Read a byte
              642 ;
D339 BD8CC0  643  WAITA2  LDA   DRV RD, X
D33C 10FB    644      BNM  WAITA2
              645 ;
              646 ;  Branch if not 2nd prologue byte
              647 ;
D33E C9AA    648      CMP   #$AA
D340 DOF2    649      BNE  CHKP B1
              650 ;
              651 ;
              652 ;
D342 A003    653      LDY   #3
              654 ;
              655 ;  Read a byte
              656 ;
D344 BD8CC0  657  WAITA3  LDA   DRV RD, X
D347 10FB    658      BNM  WAITA3
              659 ;
              660 ;  Branch if not address field prologue
              661 ;
D349 C996    662      CMP   #$96
D34B DOE7    663      BNE  CHKP B1
              664 ; .PAGE
              665 ;
              666 ;  Clear checksum
              667 ;
D34D A900    668      LDA   #0
              669 ;
              670 ;
              671 ;
D34F 85CE    672  DSKFM   STA   HCKSUM
              673 ;
              674 ;  Read an odd bits address field byte

```

```

675 ;
D351 BD8CC0 676 WAI TA4 LDA DRVRD, X
D354 10FB 677 BNM WAI TA4
678 ;
679 ;
680 ;
D356 2A 681 ROL A
D357 85CF 682 STA HODDB
683 ;
684 ; Read an even bits address field byte
685 ;
D359 BD8CC0 686 WAI TA5 LDA DRVRD, X
D35C 10FB 687 BNM WAI TA5
688 ;
689 ;
690 ;
D35E 25CF 691 AND HODDB
D360 99CA00 692 STA AFDATA, Y
693 ;
694 ;
695 ;
D363 45CE 696 EOR HCKSUM
697 ;
698 ;
699 ;
D365 88 700 DEY
D366 10E7 701 BNM DSKFM
702 ;
703 ; Branch if checksum error
704 ;
D368 A8 705 TAY
D369 D015 706 BNZ ERREX
707 ;
708 ; Read a byte
709 ;
D36B BD8CC0 710 WAI TA6 LDA DRVRD, X
D36E 10FB 711 BNM WAI TA6
712 ;
713 ; Branch if not first epilogue byte
714 ;
D370 C9DE 715 CMP #$DE
D372 D00C 716 BNE ERREX
717 ;
718 ;
719 ;
D374 EA 720 NOP
721 ;
722 ; Read a byte
723 ;
D375 BD8CC0 724 WAI TA7 LDA DRVRD, X
D378 10FB 725 BNM WAI TA7
726 ;
727 ; Branch if not 2nd epilogue byte
728 ;
D37A C9AA 729 CMP #$AA
D37C D002 730 BNE ERREX
731 ;

```

```

732 ;
733 ;
D37E 18 734 CLC
D37F 60 735 RTS
736 ;
737 ;
738 ;
D380 38 739 ERREX SEC
D381 60 740 RTS
741
*****
742 ;
743 ;
744 ;
D382 A000 745 INTDV LDY #0
746 ;
747 ;
748 ;
D384 BD8CC0 749 INTDY LDA DRVRD, X
750 ;
751 ; Time delay
752 ;
D387 2094D3 753 JSR INTDZ
D38A 48 754 PHA
D38B 68 755 PLA
756 ;
757 ; Branch if drive still moving???
758 ;
D38C DD8CC0 759 CMP DRVRD, X
D38F D003 760 BNE INTDZ
761 ;
762 ;
763 ;
D391 88 764 DEY
D392 DOF0 765 BNZ INTDY
766 ;
767 ; Return to caller
768 ;
D394 60 769 INTDZ RTS
770 ; .PAGE 'Tables'
771 ;
772 ;
773 ;
D395 00 774 DB $00
775 ;
776 ;
777 ;
D396 0004 778 DB $00, $04
D398 9899080C 779 DB $98, $99, $08, $0C
D39C 9C101418 780 DB $9C, $10, $14, $18
D3A0 A0A1A2A3 781 DB $A0, $A1, $A2, $A3
D3A4 A4A51C20 782 DB $A4, $A5, $1C, $20
D3A8 A8A9AA24 783 DB $A8, $A9, $AA, $24
D3AC 282C3034 784 DB $28, $2C, $30, $34
D3B0 B0B1383C 785 DB $B0, $B1, $38, $3C
D3B4 4044484C 786 DB $40, $44, $48, $4C
D3B8 B8505458 787 DB $B8, $50, $54, $58

```

D3BC	5C606468	788	DB	\$5C, \$60, \$64, \$68
D3C0	C0C1C2C3	789	DB	\$C0, \$C1, \$C2, \$C3
D3C4	C4C5C6C7	790	DB	\$C4, \$C5, \$C6, \$C7
D3C8	C8C9CA6C	791	DB	\$C8, \$C9, \$CA, \$6C
D3CC	CC707478	792	DB	\$CC, \$70, \$74, \$78
D3D0	D0D1D27C	793	DB	\$D0, \$D1, \$D2, \$7C
D3D4	D4D58084	794	DB	\$D4, \$D5, \$80, \$84
D3D8	D8888C90	795	DB	\$D8, \$88, \$8C, \$90
D3DC	94989CA0	796	DB	\$94, \$98, \$9C, \$A0
D3E0	E0E1E2E3	797	DB	\$E0, \$E1, \$E2, \$E3
D3E4	E4A4A8AC	798	DB	\$E4, \$A4, \$A8, \$AC
D3E8	E8B0B4B8	799	DB	\$E8, \$B0, \$B4, \$B8
D3EC	BCC0C4C8	800	DB	\$BC, \$C0, \$C4, \$C8
D3F0	FOF1CCD0	801	DB	\$F0, \$F1, \$CC, \$D0
D3F4	D4D8DCE0	802	DB	\$D4, \$D8, \$DC, \$E0
D3F8	F8E4E8EC	803	DB	\$F8, \$E4, \$E8, \$EC
D3FC	FOF4F8FC	804	DB	\$F0, \$F4, \$F8, \$FC
		805	;	
		806	;	
		807	;	
D400	00	808	DSKCA	DB \$00
D401	00	809	DSKCB	DB \$00
D402	00	810	DSKCC	DB \$00
		811	;	
		812	;	
		813	;	
D403	96020000	814	WRXLTT	DB \$96, \$02, \$00, \$00
D407	97010000	815	DB	\$97, \$01, \$00, \$00
D40B	9A030000	816	DB	\$9A, \$03, \$00, \$00
D40F	9B000200	817	DB	\$9B, \$00, \$02, \$00
D413	9D020200	818	DB	\$9D, \$02, \$02, \$00
D417	9E010200	819	DB	\$9E, \$01, \$02, \$00
D41B	9F030200	820	DB	\$9F, \$03, \$02, \$00
D41F	A6000100	821	DB	\$A6, \$00, \$01, \$00
D423	A7020100	822	DB	\$A7, \$02, \$01, \$00
D427	AB010100	823	DB	\$AB, \$01, \$01, \$00
D42B	AC030100	824	DB	\$AC, \$03, \$01, \$00
D42F	AD000300	825	DB	\$AD, \$00, \$03, \$00
D433	AE020300	826	DB	\$AE, \$02, \$03, \$00
D437	AF010300	827	DB	\$AF, \$01, \$03, \$00
D43B	B2030300	828	DB	\$B2, \$03, \$03, \$00
D43F	B3000002	829	DB	\$B3, \$00, \$00, \$02
D443	B4020002	830	DB	\$B4, \$02, \$00, \$02
D447	B5010002	831	DB	\$B5, \$01, \$00, \$02
D44B	B6030002	832	DB	\$B6, \$03, \$00, \$02
D44F	B7000202	833	DB	\$B7, \$00, \$02, \$02
D453	B9020202	834	DB	\$B9, \$02, \$02, \$02
D457	BA010202	835	DB	\$BA, \$01, \$02, \$02
D45B	BB030202	836	DB	\$BB, \$03, \$02, \$02
D45F	BC000102	837	DB	\$BC, \$00, \$01, \$02
D463	BD020102	838	DB	\$BD, \$02, \$01, \$02
D467	BE010102	839	DB	\$BE, \$01, \$01, \$02
D46B	BF030102	840	DB	\$BF, \$03, \$01, \$02
D46F	CB000302	841	DB	\$CB, \$00, \$03, \$02
D473	CD020302	842	DB	\$CD, \$02, \$03, \$02
D477	CE010302	843	DB	\$CE, \$01, \$03, \$02
D47B	CF030302	844	DB	\$CF, \$03, \$03, \$02

```

D47F D3000001 845 DB SD3, $00, $00, $01
D483 D6020001 846 DB SD6, $02, $00, $01
D487 D7010001 847 DB SD7, $01, $00, $01
D48B D9030001 848 DB SD9, $03, $00, $01
D48F DA000201 849 DB SDA, $00, $02, $01
D493 DB020201 850 DB SDB, $02, $02, $01
D497 DC010201 851 DB SDC, $01, $02, $01
D49B DD030201 852 DB SDD, $03, $02, $01
D49F DE000101 853 DB SDE, $00, $01, $01
D4A3 DF020101 854 DB SDF, $02, $01, $01
D4A7 E5010101 855 DB SE5, $01, $01, $01
D4AB E6030101 856 DB SE6, $03, $01, $01
D4AF E7000301 857 DB SE7, $00, $03, $01
D4B3 E9020301 858 DB SE9, $02, $03, $01
D4B7 EA010301 859 DB SEA, $01, $03, $01
D4BB EB030301 860 DB SEB, $03, $03, $01
D4BF EC000003 861 DB SEC, $00, $00, $03
D4C3 ED020003 862 DB SED, $02, $00, $03
D4C7 EE010003 863 DB SEE, $01, $00, $03
D4CB EF030003 864 DB SEF, $03, $00, $03
D4CF F2000203 865 DB SF2, $00, $02, $03
D4D3 F3020203 866 DB SF3, $02, $02, $03
D4D7 F4010203 867 DB SF4, $01, $02, $03
D4DB F5030203 868 DB SF5, $03, $02, $03
D4DF F6000103 869 DB SF6, $00, $01, $03
D4E3 F7020103 870 DB SF7, $02, $01, $03
D4E7 F9010103 871 DB SF9, $01, $01, $03
D4EB FA030103 872 DB SFA, $03, $01, $03
D4EF FB000303 873 DB SFB, $00, $03, $03
D4F3 FC020303 874 DB SFC, $02, $03, $03
D4F7 FD010303 875 DB SFD, $01, $03, $03
D4FB FE030303 876 DB SFE, $03, $03, $03
D4FF FF 877 DB SFF
878 ;.PAGE 'Read data field'
879 ;
880 ;
881 ;
D500 38 882 RDFNG1 SEC
D501 60 883 RTS
884 ;
885 ;
886 ;
D502 86CE 887 READDF STX HSL0T1
888 ;
889 ; Set low byte of port addresses that read into 'X' register
890 ;
D504 8A 891 TXA
D505 098C 892 ORA #$8C
D507 8D61D5 893 STA WAI TD4+1
D50A 8D78D5 894 STA WAI TD5+1
D50D 8D8ED5 895 STA WAI TD6+1
D510 8DA2D5 896 STA WAI TD7+1
D513 8DB7D5 897 STA WAI TD8+1
898 ;
899 ;
900 ;
D516 A53E 901 LDA A3E

```



```

D518 A43F      902     LDY   A3E+1
D51A 8DB4D5   903     STA   IMM3+1
D51D 8CB5D5   904     STY   IMM3+2
          905     ;
          906     ;
          907     ;
D520 38       908     CLB
D521 E954     909     SBB   #84
D523 B002     910     BNB   DSKFW
D525 88       911     DEY
D526 38       912     CLB
D527 8D9CD5   913     DSKFW  STA   IMM2+1
D52A 8C9DD5   914     STY   IMM2+2
          915     ;
          916     ;
          917     ;
D52D E957     918     SBB   #87
D52F B001     919     BNB   DSKFX
D531 88       920     DEY
D532 8D75D5   921     DSKFX  STA   IMM1+1
D535 8C76D5   922     STY   IMM1+2
          923     ;
          924     ;   Set retry count
          925     ;
D538 A020     926     LDY   #32
          927     ;
          928     ;
          929     ;
D53A 88       930     DSKFY  DEY
D53B FOC3     931     BZ    RDFNG1
          932     ;
          933     ;   Read a byte
          934     ;
D53D BD8CC0   935     WAI TD1  LDA   DRVRD, X
D540 10FB     936     BNM   WAI TD1
          937     ;
          938     ;   Branch if not 1st prologue byte
          939     ;
D542 49D5     940     DSKGA  EOR   #$D5
D544 DOF4     941     BNZ   DSKFY
          942     ;
          943     ;
          944     ;
D546 EA       945     NOP
          946     ;
          947     ;   Read a byte
          948     ;
D547 BD8CC0   949     WAI TD2  LDA   DRVRD, X
D54A 10FB     950     BNM   WAI TD2
          951     ;
          952     ;   Branch if not 2nd prologue byte
          953     ;
D54C C9AA     954     CMP   #$AA
D54E DOF2     955     BNE   DSKGA
          956     ;
          957     ;
          958     ;

```

```

D550 EA          959      NOP
                960      ;
                961      ; Read a byte
                962      ;
D551 BD8CC0     963      WAI TD3   LDA   DRV RD, X
D554 10FB       964      BNM   WAI TD3
                965      ;
                966      ; Branch if not data field
                967      ;
D556 C9AD       968      CMP   #$AD
D558 DOE8       969      BNE   DSKGA
                970      ;
                971      ;
                972      ;
D55A A0AA       973      LDY   #-86
                974      ;
                975      ; Insure checksum starts at zero
                976      ;
D55C A900       977      LDA   #0
                978      ;
                979      ; Save current checksum value
                980      ;
D55E 85CF       981      DSKGE   STA   ACF
                982      ;
                983      ; Read a byte
                984      ;
D560 AEECC0     985      WAI TD4   LDX   DRV RD+$60
D563 10FB       986      BNM   WAI TD4
                987      ;
                988      ;
                989      ;
D565 BD00D3     990      LDA   AD300, X
D568 995602     991      STA   BU FF6B+86, Y
D56B 45CF       992      EOR   ACF
                993      ;
                994      ;
                995      ;
D56D C8         996      I NY
D56E DOEE       997      BNZ   DSKGE
                998      ;
                999      ;
                1000     ;
D570 A0AA       1001     LDY   #$AA
D572 D003       1002     BNZ   WAI TD5
                1003     ;
                1004     ;
                1005     ;
D574 990010     1006     IMM1   STA   DUMMY, Y           ; User buffer address - 161
                1007     ;
                1008     ; Read a byte
                1009     ;
D577 AEECC0     1010     WAI TD5   LDX   DRV RD+$60
D57A 10FB       1011     BNM   WAI TD5
                1012     ;
                1013     ;
                1014     ;
D57C 5D00D3     1015     EOR   AD300, X

```

---

Apple II Pascal Information

---

```

D57F BE5602    1016    LDX    BUFF6B+86, Y
D582 5D00D4    1017    EOR    DSKCA, X
                1018    ;
                1019    ;
                1020    ;
D585 C8        1021    INY
D586 DOEC      1022    BNZ    IMM1
                1023    ;
                1024    ;
                1025    ;
D588 48        1026    PHA
D589 29FC      1027    AND    #%11111100
                1028    ;
                1029    ;
                1030    ;
D58B A0AA      1031    LDY    #$AA
                1032    ;
                1033    ; Read a byte
                1034    ;
D58D AEECC0    1035    WAITD6  LDX    DRV RD+$60
D590 10FB      1036    BNM    WAITD6
                1037    ;
                1038    ;
                1039    ;
D592 5D00D3    1040    EOR    AD300, X
D595 BE5602    1041    LDX    BUFF6B+86, Y
D598 5D01D4    1042    EOR    DSKCB, X
D59B 990010    1043    IMM2    STA    DUMMY, Y           ; User buffer address - 84
                1044    ;
                1045    ;
                1046    ;
D59E C8        1047    INY
D59F DOEC      1048    BNZ    WAITD6
                1049    ;
                1050    ; Read a byte
                1051    ;
D5A1 AEECC0    1052    WAITD7  LDX    DRV RD+$60
D5A4 10FB      1053    BNM    WAITD7
                1054    ;
                1055    ;
                1056    ;
D5A6 29FC      1057    AND    #%11111100
                1058    ;
                1059    ;
                1060    ;
D5A8 A0AC      1061    LDY    #- 84
                1062    ;
                1063    ;
                1064    ;
D5AA 5D00D3    1065    DSKGV    EOR    AD300, X
D5AD BE5402    1066    LDX    BUFF6B+84, Y
D5B0 5D02D4    1067    EOR    DSKCC, X
D5B3 990010    1068    IMM3    STA    DUMMY, Y
                1069    ;
                1070    ; Read a byte
                1071    ;
D5B6 AEECC0    1072    WAITD8  LDX    DRV RD+$60

```

```

D5B9 10FB      1073      BNM   WAI TD8
                1074      ;
                1075      ;
                1076      ;
D5BB C8       1077      INY
D5BC DOEC     1078      BNZ   DSKGV
                1079      ;
                1080      ; Branch if checksum error
                1081      ;
D5BE 29FC     1082      AND   #%11111100
D5C0 5D00D3   1083      EOR   AD300, X
D5C3 A6CE     1084      LDX   HSL0T1
D5C5 A8       1085      TAY
D5C6 D009     1086      BNZ   RDFNG2
                1087      ;
                1088      ; Read a byte
                1089      ;
D5C8 BD8CC0   1090      WAI TD9  LDA   DRVRD, X
D5CB 10FB     1091      BNM   WAI TD9
                1092      ;
                1093      ; Branch if not 1st epilogue byte
                1094      ;
D5CD C9DE     1095      CMP   #$DE
D5CF F002     1096      BE    RДФOK
                1097      ;
                1098      ;
                1099      ;
D5D1 38       1100      RДФNG2  SEC
D5D2 24       1101      DB    $24
                1102      ;
                1103      ;
                1104      ;
D5D3 18       1105      RДФOK   CLC
D5D4 68       1106      PLA
D5D5 A055     1107      LDY   #85
D5D7 913E     1108      STA   (A3E), Y
D5D9 60       1109      RTS
                1110      ; .PAGE 'Position read head'
                1111      ;
                1112      ;
                1113      ;
D5DA 86C8     1114      SEEK   STX   REQSLT
                1115      ;
                1116      ; Save number of track to seek to
                1117      ;
D5DC 85C9     1118      STA   REQTRK
                1119      ;
                1120      ; Branch if no head movement needed
                1121      ;
D5DE CDFBDF   1122      CMP   CURTRK
D5E1 F053     1123      BE    SEEKEX
                1124      ;
                1125      ;
                1126      ;
D5E3 A900     1127      LDA   #0
D5E5 85CF     1128      STA   LOOPC
                1129      ;

```

```

1130 ;
1131 ;
D5E7 ADFBDF 1132 SEEKLP LDA CURTRK
D5EA 85CE 1133 STA PRVTRK
1134 ;
1135 ;
1136 ;
D5EC 38 1137 CLB
D5ED E5C9 1138 SBB REQTRK
D5EF F033 1139 BZ DSKBR
1140 ;
1141 ;
1142 ;
D5F1 B007 1143 BNB DSKBN
1144 ;
1145 ;
1146 ;
D5F3 49FF 1147 EOR #$FF
1148 ;
1149 ;
1150 ;
D5F5 EEFBDF 1151 INC CURTRK
1152 ;
1153 ;
1154 ;
D5F8 9005 1155 BB DSKBO
1156 ;
1157 ;
1158 ;
D5FA 69FE 1159 DSKBN ADC #-2
1160 ;
1161 ;
1162 ;
D5FC CEFBDF 1163 DEC CURTRK
1164 ;
1165 ;
1166 ;
D5FF C5CF 1167 DSKBO CMP LOOPC
D601 9002 1168 BL DSKBP
1169 ;
1170 ;
1171 ;
D603 A5CF 1172 LDA LOOPC
1173 ;
1174 ;
1175 ;
D605 C908 1176 DSKBP CMP #8
D607 B001 1177 BGE DSKBQ
1178 ;
1179 ;
1180 ;
D609 A8 1181 TAY
1182 ;
1183 ;
1184 ;
D60A 38 1185 DSKBQ SEC
D60B 2028D6 1186 JSR ONPH

```

```

1187 ;
1188 ;
1189 ;
D60E B948D6 1190 LDA AMDTB1, Y
1191 ifdef ORIG
D611 2037D6 1192 JSR DELAY
1193 else
1194 JSR DLAYCC
1195 endif
1196 ;
1197 ;
1198 ;
D614 A5CE 1199 LDA PRVTRK
D616 18 1200 CLC
D617 202BD6 1201 JSR INTFX
1202 ;
1203 ;
1204 ;
D61A B950D6 1205 LDA AMDTB2, Y
1206 ifdef ORIG
D61D 2037D6 1207 JSR DELAY
1208 else
1209 JSR DLAYCC
1210 endif
1211 ;
1212 ;
1213 ;
D620 E6CF 1214 INC LOOPC
D622 DOC3 1215 BNZ SEEKLP
1216 ;
1217 ;
1218 ;
1219 DSKBR
1220 ifdef ORIG
D624 2037D6 1221 JSR DELAY
1222 else
1223 JSR DLAYCC
1224 endif
1225 ;
1226 ;
1227 ;
D627 18 1228 CLC
1229 * JSR ONPH
1230 * RTS
1231
*****
1232 ;
1233 ;
1234 ;
D628 ADFBDF 1235 ONPH LDA CURTRK
D62B 2903 1236 INTFX AND #$03
D62D 2A 1237 ROL A
D62E 05C8 1238 ORA REQSLT
D630 AA 1239 TAX
D631 BD80C0 1240 LDA DRVSMO, X
1241 ;
1242 ; Restore slot number index value

```

```

1243 ;
D634 A6C8 1244 LDX  REQSLT
          1245 ;
          1246 ; Return to caller
          1247 ;
D636 60   1248 SEEKEX  RTS
          1249
*****
          1250 ;
          1251 ;
          1252 ;
D637 A211 1253 DELAY  LDX  #17
          1254 ;
          1255 ;
          1256 ;
D639 CA   1257 DSKFC  DEX
D63A DOFD 1258 BNZ  DSKFC
          1259 ;
          1260 ;
          1261 ;
D63C E646 1262 INC  A46
D63E D002 1263 BNZ  DSKFD
D640 E647 1264 INC  A46+1
          1265 ;
          1266 ;
          1267 ;
D642 38   1268 DSKFD  CLB
D643 E901 1269 SBB  #1
D645 DOFO 1270 BNZ  DELAY
          1271 ;
          1272 ;
          1273 ;
D647 60   1274 RTS
          1275 ifndef ORIG
          1276
*****
          1277 ;
          1278 ;
          1279 ;
          1280 DLAYCC JSR  DELAY
          1281 JSR  CONCK
          1282 RTS
          1283 endif
          1284 ;
          1285 ; Arm move delay table 1
          1286 ;
D648 01302824 1287 AMDTB1 DB 1, 48, 40, 36, 32, 30, 29, 28
          1288 ;
          1289 ; Arm move delay table 2
          1290 ;
D650 702C2622 1291 AMDTB2 DB 112, 44, 38, 34, 31, 30, 29, 28
          1292 ;. PAGE '???'
          1293 ;
          1294 ;
          1295 ;
D658 A056 1296 INTGI  LDY  #86
D65A A900 1297 LDA  #0

```

```

D65C 99FF02    1298 INTGJ   STA   BUFF2B-1, Y
D65F 88        1299     DEY
D660 DOFA      1300     BNZ   INTGJ
                1301 ;
                1302 ;
                1303 ;
D662 A255     1304 INTGK   LDX   #85
D664 B13E     1305 INTGL   LDA   (A3E), Y
D666 29FC     1306     AND   #$FC
D668 990002   1307     STA   BUFF6B, Y
D66B 513E     1308     EOR   (A3E), Y
D66D C8       1309     INY
D66E C902     1310     CMP   #$02
D670 1D0003   1311     ORA   BUFF2B, X
D673 6A       1312     ROR   A
D674 6A       1313     ROR   A
D675 9D0003   1314     STA   BUFF2B, X
D678 CA       1315     DEX
D679 10E9     1316     BNM   INTGL
D67B C002     1317     CPY   #2
D67D DOE3     1318     BNE   INTGK
                1319 ;
                1320 ;   Return to caller
                1321 ;
D67F 60       1322     RTS
                83     INCLUDE   V1110P2.IAS
                1     ;
                2     ;   Slot table
                3     ;
D680 605040   4     SLTTBL  DB   $60, $50, $40
                5     ; .PAGE 'Disk initialization'
                6     ;
                7     ;
                8     ;
D683 4A       9     DINIT   LSR   A
D684 C903    10     INTGN   CMP   #$03
D686 B00B    11     BGE   INTGO
D688 4903    12     EOR   #$03
D68A AA      13     TAX
D68B BD2ABF  14     LDA   SLTYP3+3, X
D68E 4902    15     EOR   #$02
D690 AA      16     TAX
D691 F002    17     BZ   INTGP
                18 ;
                19 ;
                20 ;
D693 A209    21     INTGO   LDX   #9
                22 ;
                23 ;
                24 ;
D695 60      25     INTGP   RTS
                84     INCLUDE   INIT1.IAS
                1     ; .PAGE 'Table of addresses for page zero'
                2     ifdef ORIG
                3     ;
                4     ;
                5     ;

```



```

D696 80FE      6  JVECTS  DW    UDJMPV
D698 B0FE      7      DW    DSKNUM
D69A 00FF      8      DW    BIOS
D69C 5CFF      9      DW    BIOSAF
                10     endi f
                11     ; .PAGE 'Clear memory'
                12     ;
                13     ; Clear decimal flag
                14     ;
D69E D8        15  CLRMEM  CLD
                16     ;
                17     ; Disable interrupts
                18     ;
D69F 78        19     SEI
                20     ;
                21     ; Set pointer to location zero
                22     ;
D6A0 A900      23     LDA    #$00
D6A2 85BD      24     STA    ABD
D6A4 85BE      25     STA    ABD+1
                26     ;
                27     ; Clear 'X' and 'Y' registers
                28     ;
                29     * LDA    #0
D6A6 A8        30     TAY
D6A7 AA        31     TAX
                32     ;
                33     ; Clear a page
                34     ;
D6A8 91BD      35  INTGR   STA    (ABD), Y
D6AA C8        36     INY
D6AB DOFB      37     BNZ    INTGR
                38     ;
                39     ; Add 256 to pointer
                40     ;
D6AD E6BE      41     INC    ABD+1
                42     ;
                43     ; Increment page number
                44     ;
D6AF E8        45     INX
                46     ifndef ORIG
                47     ;
                48     ; Zero $0000..$03FF, $0800..$67FF, $6D00..$BFFF
                49     ;
                50     CPX    #$04
                51     BE    BEGAE
                52     CPX    #$C0
                53     BE    BEGAF
                54     CPX    #$68
                55     BNE    INTGR
                56     LDX    #$6D
                57     STX    ABD+1
                58     BNZ    INTGR
                59     BEGAE  LDX    #$08
                60     STX    ABD+1
                61     BNZ    INTGR
                62     BEGAF

```

```

63     else
64     ;
65     ;   Zero $0000.. $BFFF
66     ;
D6B0 E0C0 67     CPX   #$C0
D6B2 D0F4 68     BNE   INTGR
69     endif
70     ;.PAGE 'Slot scan'
71     ;
72     ;   Point to slot 7
73     ;
D6B4 A0C7 74     LDY   #<$C700
75     ;
76     ;   Store current slot page
77     ;
D6B6 84C6 78     INTGT  STY   AC5+1
79     ;
80     ;   Sum page and save sum
81     ;
82     ifndef ORIG
83         JSR   CKPAGE+RELOC-CLRMEM
84     else
D6B8 2058D7 85     JSR   CKPAGE
86     endif
D6BB 85D0 87     STA   ADO
D6BD 86D1 88     STX   ADO+1
89     ;
90     ;   Sum page again
91     ;
92     ifndef ORIG
93         JSR   CKPAGE+RELOC-CLRMEM
94     else
D6BF 2058D7 95     JSR   CKPAGE
96     endif
97     ;
98     ;   Branch if not ROM page
99     ;
D6C2 E000 100    CPX   #$00
D6C4 F03F 101    BE    INTHD
D6C6 C5D0 102    CMP   ADO
D6C8 D03B 103    BNE   INTHD
D6CA E4D1 104    CPX   ADO+1
D6CC D037 105    BNE   INTHD
106    ifdef ORIG
107    ;
108    ;
109    ;
D6CE F008 110    BE    INTGX
111    ;
112    ;
113    ;
D6D0 03183848 114   CN05T  DB    $03, $18, $38, $48
D6D4 3C381848 115   CN07T  DB    $3C, $38, $18, $48
116    ;
117    ;
118    ;
119    INTGX

```

```

120   endi f
121   ;
122   ;
123   ;
D6D8 A205   124   LDX   #5
D6DA A005   125   INTGY  LDY   #5
D6DC B1C5   126   LDA   (AC5), Y
D6DE DDCED6 127   CMP   CNO5T-2, X
D6E1 D009   128   BNE   INTGZ
129   ifndef ORIG
130   CMP   #$03
131   BE    BEGAN
132   endi f
D6E3 A007   133   LDY   #7
D6E5 B1C5   134   LDA   (AC5), Y
D6E7 DDD2D6 135   CMP   CNO7T-2, X
D6EA F005   136   BE    INTHA
D6EC CA     137   INTGZ  DEX
D6ED E002   138   CPX   #2
D6EF B0E9   139   BGE   INTGY
D6F1 E004   140   INTHA  CPX   #4
D6F3 D00A   141   BNE   INTHB
142   ;
143   ;
144   ;
D6F5 A00B   145   LDY   #11
D6F7 B1C5   146   LDA   (AC5), Y
D6F9 C901   147   CMP   #$01
D6FB D002   148   BNE   INTHB
149   ;
150   ;
151   ;
D6FD A206   152   LDX   #6
153   ifndef ORIG
154   ;
155   ;   Unconditional branch
156   ;
157   BNZ   INTHB
158   ;
159   ;
160   ;
161   BEGAN  LDY   #1
162   LDA   (AC5), Y
163   CMP   #$20
164   BNE   INTGZ
165   ;
166   ;
167   ;
168   LDY   #3
169   LDA   (AC5), Y
170   CMP   #$00
171   BNE   INTGZ
172   ;
173   ;
174   ;
175   LDY   #255
176   LDA   (AC5), Y

```

```

177     BZ     INTHB
178     LDY   #254
179     LDA   (AC5),Y
180     AND   #S03
181     BZ     INTGZ
182     LDA   #S80
183     STA   ABF33
184     LDX   #7
185     endif
186     ;
187     ;
188     ;
D6FF A4C6 189 INTHB     LDY   AC5+1
D701 8A   190     TXA
D702 9967BE 191     STA   SLTYP5-$C0,Y
D705 A4C6 192 INTHD     LDY   AC5+1
D707 88   193     DEY
D708 C0C0 194     CPY   #S00
D70A D0AA 195     BNE   INTGT
196     ifdef ORIG
197     ;
198     ; Unconditional branch
199     ;
D70C F004 200     BE     INTHH
201     else
202     ;.PAGE 'More stuff'
203     ;
204     ;
205     ;
206     LDX   SLTYP5+3
207     STX   ABFOE
208     LDA   #S00
209     STA   SLTYP5+3
210     ;
211     ; Initialize for move of instruction sequence
212     ;
213     LDY   #4-1
214     ;
215     ; Unconditional branch
216     ;
217     BNZ   INTHI
218     endif
219     ;
220     ; Instruction sequence
221     ;
D70E 2036FF 222 INTHG     JSR   AFF36
D711 60   223     RTS
224     ifdef ORIG
225     ;
226     ; Initialize for move of instruction sequence
227     ;
D712 A003 228 INTHH     LDY   #4-1
229     endif
230     ;
231     ; Move ' JSR AFF36' and ' RTS' to SYSCOM
232     ;
233     INTHI

```

```

234   ifndef ORIG
235     LDA   INTHG+RELOC-CLRMEM, Y
236   else
D714 B90ED7 237     LDA   INTHG, Y
238   endif
D717 990ABF 239     STA   ABFOA, Y
D71A 88      240     DEY
D71B 10F7    241     BNM   INTHI
242   ;
243   ;   Move four addresses to page zero
244   ;
D71D A007    245     LDY   #8-1
246   INTHJ
247   ifndef ORIG
248     LDA   JVECTS+RELOC-CLRMEM, Y
249   else
D71F B996D6 250     LDA   JVECTS, Y
251   endif
D722 99E800 252     STA   UDJVP, Y
D725 88      253     DEY
D726 10F7    254     BNM   INTHJ
255   ifndef IIc
256   ;
257   ;
258   ;
D728 A980    259     LDA   #$80
D72A 85E1    260     STA   AE1
261   ;
262   ;   Set text mode
263   ;
D72C AD51C0 264     LDA   AC051
265   else
266   ;
267   ;
268   ;
269     LDA   #0
270     STA   AE1
271   ;
272   ;   Set ??? mode
273   ;
274     LDA   AC050
275   endif
276   ;
277   ;   Set bottom for graphics
278   ;
D72F AD52C0 279     LDA   AC052
280   ;
281   ;   Select primary page
282   ;
D732 AD54C0 283     LDA   AC054
284   ;
285   ;   Select hi res graphics
286   ;
D735 AD57C0 287     LDA   AC057
288   ;
289   ;   Clear keyboard strobe
290   ;

```

```

D738 AD10C0  291   LDA   KBSTRB
              292   ifndef ORIG
              293   ;.PAGE 'Logic for versions after 1.0'
              294   ;
              295   ;   Activate alternate character set
              296   ;
              297   STA   ALTCHS
              298   ;
              299   ;   Read and write RAM; use bank #2
              300   ;
              301   LDA   LCBNK2
              302   ;
              303   ;   Clear screen
              304   ;
              305   LDA   BEGDK+1+RELOC- CLRMEM
              306   PHA
              307   LDA   BEGDK+RELOC- CLRMEM
              308   PHA
              309   JMP   (NFF52)
              310   ;
              311   ;   INVERT
              312   ;
              313   BEGDH   LDA   BEGDL+1+RELOC- CLRMEM
              314   PHA
              315   LDA   BEGDL+RELOC- CLRMEM
              316   PHA
              317   JMP   (NFF52+2)
              318   ;
              319   ;   INTAG
              320   ;
              321   BEGDI   LDA   BEGDM+1+RELOC- CLRMEM
              322   PHA
              323   LDA   BEGDM+RELOC- CLRMEM
              324   PHA
              325   JMP   (NFF52+4)
              326   ;
              327   ;   Read and write RAM; use bank #1
              328   ;
              329   BEGDJ   LDA   LCBNK1
              330   ;
              331   ;
              332   ;
              333   BNZ   BEGAY
              334   ;
              335   ;
              336   ;
              337   BEGDK   DW    BEGDH- 1
              338   BEGDL   DW    BEGDI - 1
              339   BEGDM   DW    BEGDJ- 1
              340   ;
              341   ;   Read ROM; write RAM; use bank #2
              342   ;
              343   BEGAY   LDA   ROMI N
              344   LDA   ROMI N
              345   ;
              346   ;   Get ROM version byte
              347   ;

```

```

348     LDA   VERZIN
349     ;
350     ;   Get ROM revision byte
351     ;
352     LDY   ZIDBYT
353     ;
354     ;   Read and write RAM; use bank #1
355     ;
356     LDX   LCBNK1
357     LDX   LCBNK1
358     ;
359     ;
360     ;
361     LDX   #$80
362     ;
363     ;   Branch if Apple IIe ROM
364     ;
365     CMP   #$06
366     BE    BEGAZ
367     ;
368     ;   Branch if open-apple key pressed
369     ;
370     LDA   BUTNO
371     BM    BEGBF
372     ;
373     ;
374     ;
375     STX   ABF11
376     ;
377     ;   Unconditional branch
378     ;
379     BNM   BEGBF
380     ;
381     ;
382     ;
383     BEGAZ   STX   ABF11
384     ;
385     ;
386     ;
387     TYA
388     AND   #$FO
389     CMP   #$EO
390     BE    BEGGB
391     ;
392     ;
393     ;
394     CMP   #$OO
395     BE    BEGBA
396     ;
397     ;
398     ;
399     LDA   #$A3
400     ;
401     ;   Unconditional branch
402     ;
403     BM    BEGBC
404     ;

```

```

405 ;
406 ;
407 BEGBA LDA #C3
408 ;
409 ; Unconditional branch
410 ;
411 BM BEGBC
412 ;
413 ;
414 ;
415 BEGBB STX ABF31
416 ;
417 ;
418 ;
419 LDA #83
420 ;
421 ; Set 'ALTZP' softswitch
422 ;
423 STA ALTZPS
424 ;
425 ;
426 ;
427 STA AF800
428 ;
429 ;
430 ;
431 LDA #81
432 STA AFC00
433 ;
434 ;
435 ;
436 LDA AF800
437 ;
438 ; Clear 'ALTZP' softswitch
439 ;
440 STA ALTZPC
441 ;
442 ;
443 ;
444 CMP #83
445 BE BEGBC
446 ;
447 ;
448 ;
449 CMP #81
450 BNE BEGBF
451 ;
452 ;
453 ;
454 BEGBC STA ABF31
455 ;
456 ;
457 ;
458 BEGBF LDA #02
459 STA CNFLGS
460 endif
461 ifdef ORIG

```



```

462 ; .PAGE 'Logic for version 1.0'
463 ; ifndef IIc
464 ;
465 ;
466 ;
D73B 203DDB 467 ; JSR DOFF
468 ;
469 ;
470 ;
D73E 20C7DB 471 ; JSR INVERT
472 ;
473 ;
474 ;
D741 2010D0 475 ; JSR INTAG
476 ;
477 ;
478 ;
D744 AD2ABF 479 ; LDA SLTYP5+3
D747 A030 480 ; LDY #$30
D749 20F8D8 481 ; JSR INTJT
D74C E000 482 ; CPX #0
D74E D005 483 ; BNE INTHK
484 ;
485 ;
486 ;
D750 A904 487 ; LDA #4
D752 8D0EBF 488 ; STA ABFOE
489 ;
490 ;
491 ;
D755 4CF5FE 492 ; INTHK JMP OST
493 ; el se
494 ; JSR INTAG
495 ; LDA #$00
496 ; STA $F4
497 ; STA $F5
498 ; STA $F3
499 ; STA $F0
500 ; LDA #$08
501 ; STA $F1
502 ; LDA #$28
503 ; STA $F2
504 ; NOP
505 ; NOP
506 ; NOP
507 ; NOP
508 ; NOP
509 ; JMP $FEF5
510 ; endi f
511 ;
*****
512 ;
513 ; Compute sum of memory page
514 ;
D758 A900 515 ; CKPAGE LDA #0
D75A AA 516 ; TAX
D75B A8 517 ; TAY

```

```

518 ;
519 ;
520 ;
D75C 18 521 CKNX CLC
D75D 71C5 522 ADC (AC5), Y
D75F 9001 523 BNC NOCRY
D761 E8 524 INX
D762 C8 525 NOCRY INY
D763 DOF7 526 BNZ CKNX
527 ;
528 ; Return page sum in 'X' register and accumulator
529 ;
D765 60 530 RTS
531 endi f
85 INCLUDE CHARI00. IAS
1 ;
2 ;
3 ;
D766 38 4 TDELAY CLB
5 ;
6 ; Push loop control value
7 ;
D767 48 8 INTHR PHA
9 ;
10 ;
11 ;
D768 E901 12 INTHS SBB #1
D76A DOFC 13 BNZ INTHS
14 ;
15 ; Decrement loop control value
16 ;
D76C 68 17 PLA
D76D E901 18 SBB #1
19 ;
20 ;
21 ;
D76F DOF6 22 BNZ INTHR
23 ;
24 ; Return to caller
25 ;
D771 60 26 RTS
27
*****
28 ;
29 ; Save CPU state
30 ;
D772 08 31 CONCK PHP
D773 48 32 PHA
D774 8A 33 TXA
D775 48 34 PHA
D776 98 35 TYA
D777 48 36 PHA
37 ;
38 ; Increment 16 bit random seed
39 ;
D778 EE13BF 40 INTIC INC RANDNO
D77B D003 41 BNZ INTID

```

```

D77D EE14BF      42      INC   RANDNO+1
                  43  INTID
                  44  ;
                  45  ;
                  46  ;
                  47  ifndef IIc
D780 AD2ABF      48      LDA   SLTYP3+3
D783 C903        49      CMP   #3
D785 F040        50      BE    COMCK
                  51  ;
                  52  ;
                  53  ;
D787 C904        54      CMP   #4
D789 F05E        55      BE    INTIK
                  56  ;
                  57  ;
                  58  ;
D78B C906        59      CMP   #6
D78D F05D        60      BE    INTIM
                  61  endif
                  62  ;
                  63  ; Branch if no key pressed
                  64  ;
D78F AD00C0      65      LDA   KBD
                  66  ifndef IIc
D792 1055        67      BNM   INTIK
                  68  else
                  69      BNM   INTIS
                  70  endif
                  71  ;
                  72  ; Clear keyboard strobe
                  73  ;
D794 8D10C0      74      STA   KBSTRB
                  75  ;
                  76  ; Mask off high order bit
                  77  ;
D797 297F        78      AND   #%01111111
                  79  ;
                  80  ; Save character in 'X' register
                  81  ;
D799 AA          82      TAX
                  83  ifndef IIc
                  84  ;
                  85  ; Branch if not checking for special characters
                  86  ;
D79A AD1CBF      87      LDA   SPCHAR
D79D 6A          88      ROR   A
D79E B037        89      BC    INTIJ
                  90  ;
                  91  ;
                  92  ;
D7A0 8A          93      TXA
D7A1 C90B        94      CMP   #SOB
D7A3 D002        95      BNE   INTIE
                  96  ;
                  97  ;
                  98  ;

```

```

D7A5 A95B      99      LDA   #$5B
                100 ;
                101 ;
                102 ;
D7A7 C901     103 INTIE   CMP   #1
D7A9 D00E     104      BNE   INTIF
                105 ;
                106 ;
                107 ;
D7AB 20DAFF   108      JSR   HTAB
                109 ;
                110 ; Reset auto-follow bit
                111 ;
D7AE AD15BF   112      LDA   CNFLGS
D7B1 29FE     113      AND   #%11111110
D7B3 8D15BF   114      STA   CNFLGS
                115 ;
                116 ;
                117 ;
D7B6 4C33D8   118      JMP   INTIS
                119 ;
                120 ; Branch if not ^Z
                121 ;
D7B9 C91A     122 INTIF   CMP   #$1A
D7BB D07E     123      BNE   INTJA
                124 ;
                125 ;
                126 ;
D7BD AD15BF   127      LDA   CNFLGS
D7C0 0901     128      ORA   #%00000001
D7C2 8D15BF   129      STA   CNFLGS
                130 ;
                131 ; Unconditional branch
                132 ;
D7C5 D06C     133      BNZ   INTIS
                134 ;
*****
                135 ;
                136 ; Branch if comm. character not available
                137 ;
D7C7 ADBECO   138 COMCK   LDA   ACOBE
D7CA 4A       139      LSR   A
D7CB 9066     140      BNC   INTIS
                141 ;
                142 ; Get character from UART
                143 ;
D7CD ADBFC0   144      LDA   ACOBF
                145 ;
                146 ; Strip off high order bit
                147 ;
D7D0 297F     148 GOTCH   AND   #$7F
                149 ;
                150 ;
                151 ;
D7D2 AA       152 INTII   TAX
                153      endi f
                154 ;

```

```

155 ;
156 ;
D7D3 AD1CBF 157 LDA SPCHAR
D7D6 6A 158 ROR A
D7D7 6A 159 INTIJ ROR A
160 ifndef IIc
D7D8 B041 161 BC INTIP
D7DA 8A 162 TXA
163 else
164 TXA
165 BC INTIP
166 endif
D7DB A055 167 LDY #$55
D7DD D1F8 168 CMP (SYCPTR), Y
D7DF D01A 169 BNE INTIN
170 ;
171 ; Toggle stop bit
172 ;
D7E1 AD15BF 173 LDA CNFLGS
D7E4 4980 174 EOR #$80
D7E6 8D15BF 175 STA CNFLGS
176 ;
177 ;
178 ;
D7E9 4C33D8 179 INTIK JMP INTIS
180 ifndef IIc
181 ;
182 ;
183 ;
D7EC A901 184 INTIM LDA #$01
D7EE A030 185 LDY #$30
D7F0 2024DA 186 JSR INTLD
D7F3 903E 187 BNC INTIS
188 ;
189 ;
190 ;
D7F5 200CDA 191 JSR FREAD1
192 ;
193 ;
194 ;
D7F8 4CD0D7 195 JMP GOTCH
196 endif
197 ;
198 ;
199 ;
D7FB 88 200 INTIN DEY
D7FC D1F8 201 CMP (SYCPTR), Y
D7FE D00B 202 BNE NOTBRK
203 ;
204 ; Reset flush and stop bits
205 ;
D800 AD15BF 206 LDA CNFLGS
D803 293F 207 AND #%00111111
D805 8D15BF 208 STA CNFLGS
209 ;
210 ;
211 ;

```

```

D808 4CE3FE      212      JMP      TOBRK
                213      ;
                214      ;
                215      ;
D80B 88          216      NOTBRK   DEY
D80C D1F8        217      CMP      (SYCPTR), Y
D80E D00C        218      BNE      INTIQ
                219      ;
                220      ; Toggle flush bit
                221      ;
D810 AD15BF      222      LDA      CNFLGS
D813 4940        223      EOR      #$40
D815 8D15BF      224      STA      CNFLGS
                225      ;
                226      ;
                227      ;
D818 4C33D8      228      JMP      INTIS
                229      ;
                230      ;
                231      ;
                232      INTIP
                233      ifndef IIc
D81B 8A          234      TXA
                235      endif
D81C AE19BF      236      INTIQ   LDX      WPTR
D81F 2090D8      237      JSR      IXZE78
D822 EC18BF      238      CPX      RPTR
D825 D006        239      BNE      INTIR
                240      ;
                241      ;
                242      ;
D827 201ADB      243      JSR      DOBEL
                244      ;
                245      ;
                246      ;
D82A 4C33D8      247      JMP      INTIS
                248      ;
                249      ;
                250      ;
D82D 8E19BF      251      INTIR   STX      WPTR
D830 9DB103      252      STA      A03B1, X
                253      ;
                254      ;
                255      ;
D833 2C15BF      256      INTIS   BIT      CNFLGS
D836 1051        257      BNM      INTJN
                258      ;
                259      ;
                260      ;
D838 4C78D7      261      JMP      INTIC
                262      ifndef IIc
                263      ;
                264      ; Save keypress
                265      ;
D83B AA          266      INTJA   TAX
                267      ;
                268      ;

```

```

269 ;
D83C 2C63C0 270 BIT AC063
D83F 1015 271 BNM INTJC
272 ;
273 ;
274 ;
D841 A5E1 275 LDA AE1
D843 2A 276 ROL A
277 ;
278 ; Restore keypress
279 ;
D844 8A 280 TXA
281 ;
282 ;
283 ;
D845 B01E 284 BC INTJG
285 ;
286 ; Branch if not upper case character
287 ;
D847 C941 288 CMP #' A
D849 901A 289 BL INTJG
D84B C95B 290 CMP #' Z+1
D84D B016 291 BGE INTJG
292 ;
293 ; Convert to lower case
294 ;
D84F 0920 295 ORA #%00100000
296 ;
297 ; Unconditional branch
298 ;
D851 D003 299 BNZ INTJC
300 ;
301 ;
302 ;
D853 4CD2D7 303 INTJB JMP INTII
304 ;
305 ;
306 ;
D856 A205 307 INTJC LDX #6-1
D858 DDCEFF 308 INTJE CMP XLTI N, X
D85B F005 309 BE INTJF
D85D CA 310 DEX
D85E 10F8 311 BNM INTJE
312 ;
313 ; Unconditional branch
314 ;
D860 3003 315 BM INTJG
316 ;
317 ;
318 ;
D862 BDD4FF 319 INTJF LDA XLTOU T, X
320 ;
321 ;
322 ;
D865 48 323 INTJG PHA
324 ;
325 ;

```

```

326 ;
D866 A5E1 327 LDA AE1
D868 2940 328 AND #%01000000
D86A F004 329 BZ INTJK
330 ;
331 ;
332 ;
D86C A900 333 LDA #0
D86E 85E1 334 STA AE1
335 ;
336 ;
337 ;
D870 68 338 INTJK PLA
D871 C905 339 CMP #$05
D873 D007 340 BNE INTJL
341 ;
342 ;
343 ;
D875 A5E1 344 LDA AE1
D877 4980 345 EOR #%10000000
346 ;
347 ;
348 ;
D879 4C82D8 349 JMP INTJM
350 ;
351 ; Branch if not ^W
352 ;
D87C C917 353 INTJL CMP #$17
D87E D0D3 354 BNE INTJB
355 ;
356 ;
357 ;
D880 A9C0 358 LDA #%11000000
359 ;
360 ;
361 ;
D882 85E1 362 INTJM STA AE1
363 ;
364 ;
365 ;
D884 20FBFE 366 JSR DOH12
D887 DOAA 367 BNZ INTIS
368 endif
369 ;
370 ; Restore CPU state
371 ;
D889 68 372 INTJN PLA
D88A A8 373 TAY
D88B 68 374 PLA
D88C AA 375 TAX
D88D 68 376 PLA
D88E 28 377 PLP
378 ;
379 ; Return to caller
380 ;
D88F 60 381 RTS
382 ;

```



```

383 ; Increment X register, clear X register if = 78
384 ;
D890 E8 385 IXZE78 INX
D891 E04E 386 CPX #78
D893 D002 387 BNE INTJQ
D895 A200 388 LDX #0
D897 60 389 INTJQ RTS
390 ; PAGE 'Console initialization'
391 ;
392 ; Pull return address from stack
393 ;
D898 68 394 CINIT PLA
D899 85F6 395 STA TEMP
D89B 68 396 PLA
D89C 85F7 397 STA TEMP+1
398 ;
399 ; Save pointer to SYSCOM area
400 ;
D89E 68 401 PLA
D89F 85F8 402 STA SYCPTR
D8A1 68 403 PLA
D8A2 85F9 404 STA SYCPTR+1
405 ;
406 ; Save break address
407 ;
D8A4 68 408 PLA
D8A5 8D16BF 409 STA BREAK
D8A8 68 410 PLA
D8A9 8D17BF 411 STA BREAK+1
412 ;
413 ; Push return address back on stack
414 ;
D8AC A5F7 415 LDA TEMP+1
D8AE 48 416 PHA
D8AF A5F6 417 LDA TEMP
D8B1 48 418 PHA
419 ;
420 ; Flush type-ahead buffer
421 ;
D8B2 AD18BF 422 LDA RPTR
D8B5 8D19BF 423 STA WPTR
424 ;
425 ; Reset stop, flush and auto-follow bits
426 ;
D8B8 AD15BF 427 LDA CNFLGS
D8BB 293E 428 AND #$3E
D8BD 8D15BF 429 STA CNFLGS
430 ifndef IIc
431 ;
432 ; Horizontal shift full left
433 ;
D8C0 20E5FF 434 JSR TAB3
435 endif
436 ;
437 ;
438 ;
D8C3 A200 439 LDX #0

```

```

440 ;
441 ; Return to caller
442 ;
D8C5 60 443 RTS
444 ;.PAGE 'Console read'
445 ;
446 ;
447 ;
D8C6 2059DC 448 CREAD JSR INTOA
449 ifndef IIc
450 ;
451 ;
452 ;
D8C9 A030 453 LDY #$30
454 ;
455 ;
456 ;
D8CB AD2ABF 457 LDA SLTYP+3
D8CE C904 458 CMP #$04
D8D0 D006 459 BNE INTJS
460 ;
461 ;
462 ;
D8D2 2015DA 463 JSR RSER
D8D5 297F 464 AND #$7F
465 ;
466 ; Return to caller
467 ;
D8D7 60 468 RTS
469 INTJS
470 endif
471 ;
472 ;
473 ;
D8D8 2072D7 474 JSR CONCK
D8DB AE18BF 475 LDX RPTR
D8DE EC19BF 476 CPX WPTR
477 ifndef IIc
D8E1 FOE3 478 BE CREAD
479 else
480 BE CREAD+3
481 endif
D8E3 2090D8 482 JSR IXZE78
D8E6 8E18BF 483 STX RPTR
484 ifdef IIc
485 JSR $DE8A
486 LDX RPTR
487 endif
D8E9 BDB103 488 LDA A03B1,X
D8EC A200 489 LDX #0
490 ;
491 ; Return to caller
492 ;
D8EE 60 493 RTS
494 ;.PAGE 'Printer initialization'
495 ;
496 ;

```

```

497 ;
D8EF A010 498 PINIT LDY #S10
499 ;
500 ; Branch if printer card
501 ;
D8F1 AD28BF 502 LDA SLTYP5+1
D8F4 C905 503 CMP #5
D8F6 F035 504 BE CLRI01
505 ;
506 ; Branch if serial card
507 ;
D8F8 C904 508 INTJT CMP #4
D8FA F034 509 BE ISER
510 ;
511 ; Branch if com card
512 ;
D8FC C903 513 CMP #3
D8FE F023 514 BE ICOM
515 ;
516 ;
517 ;
D900 C906 518 CMP #6
D902 F003 519 BE FINIT
520 ;
521 ; Set IOResult
522 ;
D904 A209 523 LDX #9
524 ;
525 ; Return to caller
526 ;
D906 60 527 RTS
528 ; PAGE 'Firmware initialization'
529 ;
530 ;
531 ;
D907 48 532 FINIT PHA
D908 2039D9 533 JSR SER1
534 ;
535 ;
536 ;
D90B A00D 537 LDY #13
D90D B1F6 538 FVEC1 LDA (TEMP),Y
D90F 85F6 539 STA TEMP
D911 ACF806 540 LDY A06F8
D914 68 541 PLA
D915 6CF600 542 JMP (TEMP)
543
*****
544 ;
545 ;
546 ;
547 GINIT
548 ifndef IIc
D918 AD51C0 549 LDA AC051
550 else
551 LDA AC050
552 endif

```

```

553 ;
554 ;
555 ;
D91B 60 556 RTS
557 ;.PAGE 'Remote initialization'
558 ;
559 ;
560 ;
D91C AD29BF 561 RINIT LDA SLTYP5+2
562 ;
563 ;
564 ;
D91F A020 565 LDY #$20
566 ;
567 ; Unconditional branch
568 ;
D921 D0D5 569 BNZ INTJT
570 ;.PAGE 'Initialize com card'
571 ;
572 ;
573 ;
D923 A903 574 ICOM LDA #$03
D925 998ECO 575 STA AC08E, Y
576 ;
577 ;
578 ;
D928 A915 579 LDA #$15
D92A 998ECO 580 STA AC08E, Y
581 ;
582 ; IOResult always zero
583 ;
D92D A200 584 CLRIO1 LDX #0
585 ;
586 ; Return to caller
587 ;
D92F 60 588 RTS
589 ;.PAGE 'Initialize serial card'
590 ;
591 ;
592 ;
D930 2039D9 593 ISER JSR SER1
594 ;
595 ; Set up slot dependents
596 ;
D933 2000C8 597 JSR AC800
598 ;
599 ; IOResult always zero
600 ;
D936 A200 601 LDX #0
602 ;
603 ;
604 ;
D938 60 605 RTS
606 ;.PAGE 'Assorted serial card setup'
607 ;
608 ;
609 ;

```

```

D939 8CF806      610 SER1      STY      A06F8
D93C 98          611      TYA
D93D 4A          612      LSR      A
D93E 4A          613      LSR      A
D93F 4A          614      LSR      A
D940 4A          615      LSR      A
D941 09C0        616      ORA      #$CO
D943 AA          617      TAX
                618 ;
                619 ;
                620 ;
D944 A900        621      LDA      #0
D946 85F6        622      STA      TEMP
D948 86F7        623      STX      TEMP+1
                624 ;
                625 ; Turn off all C8 ROMs
                626 ;
D94A ADFFCF      627      LDA      ACFFF
                628 ;
                629 ; Select C8 bank
                630 ;
D94D B1F6        631      LDA      (TEMP), Y
                632 ;
                633 ; Return to caller
                634 ;
D94F 60          635      RTS
                636 ;.PAGE 'Console initialization'
                637 ;
                638 ;
                639 ;
D950 2072D7      640 CWRITE      JSR      CONCK
                641      ifdef IIc
                642 ;
                643 ;
                644 ;
                645      JSR      $D97A
                646      endif
                647      ifndef IIc
                648 ;
                649 ;
                650 ;
D953 2C15BF      651      BIT      CNFLGS
D956 7020        652      BV      INTKJ
                653 ;
                654 ;
                655 ;
D958 AA          656      TAX
                657 ;
                658 ;
                659 ;
D959 A030        660      LDY      #$30
D95B AD2ABF      661      LDA      SLTYP3+3
D95E C903        662      CMP      #3
D960 F050        663      BE      WCOM
                664 ;
                665 ;
                666 ;

```

```

D962 C904      667      CMP      #4
D964 F024      668      BE        WSER
                669      ;
                670      ;
                671      ;
D966 C906      672      CMP      #6
D968 F011      673      BE        WFIRM
                674      ;
                675      ;
                676      ;
D96A 8A        677      TXA
D96B 85F6      678      STA      TEMP
                679      ;
                680      ;
                681      ;
D96D 20C7DB    682      JSR      INVERT
D970 A4F4      683      LDY      HPOS
D972 2062DA    684      JSR      VOUT2
D975 20C7DB    685      JSR      INVERT
                686      endi f
                687      ;
                688      ;
                689      ;
D978 A200      690      INTKJ   LDX      #0
                691      ;
                692      ;
                693      ;
D97A 60        694      RTS
                695      ;.PAGE 'Write to firmware'
                696      ;
                697      ;
                698      ;
D97B 8A        699      WFIRM   TXA
D97C 48        700      PHA
D97D A900      701      LDA      #0
D97F 202DDA    702      JSR      IOWAIT
                703      ;
                704      ;
                705      ;
D982 2039D9    706      JSR      SER1
                707      ;
                708      ;
                709      ;
D985 A00F      710      LDY      #$0F
                711      ;
                712      ;
                713      ;
D987 4C0DD9    714      JMP      FVEC1
                715      ;.PAGE 'Write to serial card'
                716      ;
                717      ;
                718      ;
D98A 2072D7    719      WSER     JSR      CONCK
                720      ;
                721      ; Save data byte on stack
                722      ;
                723      ;
D98D 8A        723      TXA

```

```

D98E 48      724    PHA
              725    ;
              726    ;
              727    ;
D98F 2039D9 728    JSR    SER1
              729    ;
              730    ; Set up data byte
              731    ;
D992 68      732    PLA
D993 9DB805  733    STA    A05B8,X
              734    ;
              735    ; Send it (shout)
              736    ;
D996 20AAC9 737    JSR    AC9AA
              738    ;
              739    ; IOResult always zero
              740    ;
D999 A200    741    LDX    #0
              742    ;
              743    ; Return to caller
              744    ;
D99B 60      745    RTS
              746    ;.PAGE 'Remote write'
              747    ;
              748    ;
              749    ;
D99C AA      750    RWRI TE    TAX
D99D AD29BF  751    LDA    SLTYP S+2
D9A0 A020    752    LDY    #$20
              753    ;
              754    ; Unconditional branch
              755    ;
D9A2 D032    756    BNZ    INTKV
              757    ;.PAGE 'Write to printer card in slot 1'
              758    ;
              759    ;
              760    ;
D9A4 2072D7 761    INTKP    JSR    CONCK
              762    ;
              763    ; Branch if printer not ready
              764    ;
D9A7 ADC1C1 765    LDA    AC1C1
D9AA 30F8    766    BM    INTKP
              767    ;
              768    ; Send character
              769    ;
D9AC 8E90C0 770    STX    AC090
              771    ;
              772    ; IOResult always zero
              773    ;
D9AF A200    774    LDX    #0
              775    ;
              776    ; Return to caller
              777    ;
D9B1 60      778    RTS
              779    ;.PAGE 'Write to com card'
              780    ;

```

```

781 ;
782 ;
D9B2 2072D7 783 WCOM JSR CONCK
784 ;
785 ; Branch if UART not ready
786 ;
D9B5 B98ECO 787 LDA AC08E, Y
D9B8 2902 788 AND #$02
D9BA F0F6 789 BZ WCOM
790 ;
791 ; Send character
792 ;
D9BC 8A 793 TXA
D9BD 998FC0 794 STA AC08F, Y
795 ;
796 ; IOResult always zero
797 ;
D9C0 A200 798 LDX #0
799 ;
800 ; Return to caller
801 ;
D9C2 60 802 RTS
803 ; PAGE 'Printer write'
804 ;
805 ;
806 ;
D9C3 AA 807 PWRITE TAX
808 ;
809 ;
810 ;
D9C4 ADOFBF 811 LDA LFFLAG
D9C7 1004 812 BNM INTKU
813 ;
814 ;
815 ;
D9C9 E00A 816 CPX #SOA
D9CB FOAB 817 BE INTKJ
818 ;
819 ;
820 ;
D9CD A010 821 INTKU LDY #$10
822 ;
823 ;
824 ;
D9CF AD28BF 825 LDA SLTYP5+1
D9D2 C905 826 CMP #5
D9D4 FOCE 827 BE INTKP
828 ;
829 ;
830 ;
D9D6 C904 831 INTKV CMP #4
D9D8 FOBO 832 BE WSER
833 ;
834 ;
835 ;
D9DA C903 836 CMP #3
D9DC FOD4 837 BE WCOM

```



```

838 ;
839 ;
840 ;
D9DE C906 841     CMP    #6
D9EO F099 842     BE     WFIRM
843 ;
844 ;
845 ;
D9E2 A209 846 INTKW   LDX    #9
847 ;
848 ;
849 ;
D9E4 60   850     RTS
851 ;.PAGE 'Remote read'
852 ;
853 ;
854 ;
D9E5 AD29BF 855 RREAD   LDA    SLTYP+2
D9E8 A020 856     LDY    #$20
D9EA C904 857     CMP    #4
D9EC F027 858     BE     RSER
859 ;
860 ;
861 ;
D9EE C903 862     CMP    #3
D9FO F006 863     BE     RCOM
864 ;
865 ;
866 ;
D9F2 C906 867     CMP    #6
D9F4 F011 868     BE     RFIRM
869 ;
870 ; Unconditional branch
871 ;
D9F6 DOEA 872     BNE   INTKW
873 ;.PAGE 'Read from com card'
874 ;
875 ;
876 ;
D9F8 2072D7 877 RCOM    JSR    CONCK
878 ;
879 ; Branch if character not ready
880 ;
D9FB B98ECO 881     LDA    AC08E, Y
D9FE 4A     882     LSR    A
D9FF 90F7 883     BNC   RCOM
884 ;
885 ;
886 ;
DA01 B98FC0 887     LDA    AC08F, Y
888 ;
889 ; IOResult always zero
890 ;
DA04 A200 891     LDX    #0
892 ;
893 ; Return to caller
894 ;

```

```

DA06 60      895      RTS
              896      ;.PAGE 'Read from firmware'
              897      ;
              898      ;
              899      ;
DA07 A901    900      RFIRM   LDA   #1
DA09 202DDA  901      JSR     IOWAIT
              902      ;
              903      ;
              904      ;
DA0C 2039D9  905      FREAD1  JSR   SER1
              906      ;
              907      ;
              908      ;
DA0F 48      909      PHA
              910      ;
              911      ;
              912      ;
DA10 A00E    913      LDY     #$0E
              914      ;
              915      ;
              916      ;
DA12 4C0DD9  917      JMP     FVEC1
              918      ;.PAGE 'Read from serial card'
              919      ;
              920      ;
              921      ;
DA15 2072D7  922      RSER    JSR   CONCK
              923      ;
              924      ;
              925      ;
DA18 2039D9  926      JSR     SER1
              927      ;
              928      ; Get a byte (shiftin)
              929      ;
DA1B 204DC8  930      JSR     AC84D
              931      ;
              932      ;
              933      ;
DA1E BDB805  934      LDA     A05B8, X
              935      ;
              936      ; IOResult always zero
              937      ;
DA21 A200    938      LDX     #0
              939      ;
              940      ; Return to caller
              941      ;
DA23 60      942      RTS
              943
*****
              944      ;
              945      ;
              946      ;
DA24 48      947      INTLD   PHA
DA25 2039D9  948      JSR     SER1
DA28 A010    949      LDY     #$10
DA2A 4C0DD9  950      JMP     FVEC1

```

```

951
*****
952 ;
953 ;
954 ;
DA2D 2072D7 955 IOWAIT JSR CONCK
DA30 48 956 PHA
DA31 2024DA 957 JSR INTLD
DA34 68 958 PLA
DA35 90F6 959 BNC IOWAIT
960 ;
961 ;
962 ;
DA37 60 963 RTS
86 INCLUDE INCTERM.IAS
1 ifdef ORIG
2 ;
3 ;
4 ;
DA38 1B 5 CCHTBL DB $1B ; Escape
DA39 1E 6 DB $1E ; ???
DA3A 0D 7 DB $0D ; Carriage return
DA3B 0A 8 DB $0A ; Linefeed
DA3C 07 9 DB $07 ; Bell character
DA3D 1F 10 DB $1F ; Cursor up
DA3E 1C 11 DB $1C ;
DA3F 08 12 DB $08 ; Backspace
DA40 0C 13 DB $0C ; Form feed
DA41 19 14 DB $19 ;
DA42 0B 15 DB $0B ; Vertical tab
DA43 1D 16 DB $1D ;
DA44 12 17 DB $12 ;
DA45 14 18 DB $14 ;
19 ENDCCT
20 ;
21 ;
22 ;
DA46 CFDA 23 INTLJ DW DOESC
DA48 D5DA 24 DW DOH1E
DA4A DBDA 25 DW DOCR
DA4C EODA 26 DW DOLF
DA4E 1ADB 27 DW DOBEL
DA50 2DDB 28 DW CURSUP
DA52 79DB 29 DW ADVC
DA54 36DB 30 DW DOBS
DA56 3DDB 31 DW DOFF
DA58 43DB 32 DW HOMEC
DA5A 4CDB 33 DW DOVT
DA5C 6ADB 34 DW INTML
DA5E FBFE 35 DW DOH12
DA60 4FFF 36 DW DOH14
37 endi f
38 ;
39 ;
40 ;
DA62 AD12BF 41 VOUT2 LDA ESCFLG
DA65 F038 42 BZ INTLT

```

```

43 ;
44 ;
45 ;
DA67 C902 46     CMP    #2
DA69 9030 47     BL     INTLR
48 ;
49 ;
50 ;
DA6B F012 51     BE     INTLO
52 ;
53 ; Branch if character in $00..$1F
54 ;
DA6D A5F6 55     LDA    TEMP
DA6F 38   56     SEC
DA70 E920 57     SBC    #$20
DA72 3004 58     BM     INTLM
59 ;
60 ; Branch if character in $20..$6F
61 ;
DA74 C950 62     CMP    #$50
DA76 9002 63     BL     INTLN
64 ;
65 ;
66 ;
DA78 A900 67     INTLM   LDA    #0
68 ;
69 ;
70 ;
DA7A 85F7 71     INTLN   STA    TEMP+1
72 ;
73 ;
74 ;
75     ifndef ORIG
76     INTLR   DEC    ESCFLG
77     RTS
78     else
DA7C 4C9BDA 79     JMP    INTLR
80     endif
81 ;
82 ;
83 ;
DA7F A5F6 84     INTLO   LDA    TEMP
DA81 38   85     SEC
DA82 E920 86     SBC    #$20
DA84 3004 87     BM     INTLP
DA86 C918 88     CMP    #$18
DA88 9002 89     BL     INTLQ
90 ;
91 ;
92 ;
DA8A A900 93     INTLP   LDA    #0
DA8C 85F5 94     INTLQ   STA    VPOS
95 ;
96 ;
97 ;
DA8E 20CCDB 98     JSR    CVP2SP
99 ;

```

```

100 ;
101 ;
DA91 A5F7 102 LDA TEMP+1
DA93 85F4 103 STA HPOS
104 ;
105 ;
106 ;
DA95 A900 107 LDA #0
108 ;
109 ;
110 ;
DA97 8D12BF 111 STESCF STA ESCFLG
112 ;
113 ;
114 ;
DA9A 60 115 RTS
116 ifdef ORIG
117 ;
118 ;
119 ;
DA9B CE12BF 120 INTLR DEC ESCFLG
DA9E 60 121 RTS
122 endif
123 ;
124 ;
125 ;
DA9F A5F6 126 INTLT LDA TEMP
127 ifdef ORIG
DAA1 297F 128 AND #$7F
129 endif
DAA3 C920 130 CMP #$20
DAA5 900D 131 BL INTLV
132 ;
133 ;
134 ;
135 ifndef ORIG
136 AND #$7F
137 endif
DAA7 C960 138 CMP #$60
139 ifndef ORIG
140 LDA TEMP
141 endif
DAA9 9002 142 BL INTLU
143 ;
144 ;
145 ;
146 ifndef ORIG
147 BIT ABF31
148 BM INTLY
149 endif
DAAB E920 150 SBC #$20
DAAD 293F 151 INTLU AND #$3F
152 ifndef ORIG
153 INTLY EOR #$80
154 else
DAAF 0980 155 ORA #$80
156 endif

```

```

DAB1 4C76DB      157      JMP      WCADVC
                  158      ;.PAGE 'Handle output of $00..$1F'
                  159      ;
                  160      ; Set count of characters in table
                  161      ;
DAB4 A20D        162      INTLV      LDX      #ENDCCT-CCHTBL-1
DAB6 DD38DA      163      INTLW      CMP      CCHTBL, X
DAB9 F004        164      BE        INTLX
DABB CA          165      DEX
DABC 10F8        166      BNM      INTLW
                  167      ;
                  168      ; Return to caller
                  169      ;
DABE 60          170      RTS
                  171      ;
                  172      ;
                  173      ;
DABF 8A          174      INTLX      TXA
DAC0 0A          175      ASL      A
DAC1 AA          176      TAX
DAC2 BD47DA      177      LDA      INTLJ+1, X
DAC5 85C0        178      STA      ABF+1
DAC7 BD46DA      179      LDA      INTLJ, X
DACA 85BF        180      STA      ABF
                  181      ifndef ORIG
                  182      ;
                  183      ;
                  184      ;
                  185      LDA      #$80
                  186      endif
                  187      ;
                  188      ;
                  189      ;
DACC 6CBF00      190      JMP      (ABF)
                  191      ifndef ORIG
                  192      ;
*****
                  193      ;
                  194      ;
                  195      ;
                  196      DOESC      LDA      #1
                  197      ;
                  198      ; Unconditional branch
                  199      ;
                  200      BNZ      STESCF
                  201      ;
*****
                  202      ;
                  203      ;
                  204      ;
                  205      DOH1E
                  206      LDA      #3
                  207      ;
                  208      ; Unconditional branch
                  209      ;
                  210      BNZ      STESCF
                  211      else

```

```

212
*****
213 ;
214 ;
215 ;
DACF A901 216 DOESC LDA #1
DAD1 8D12BF 217 STA ESCFLG
DAD4 60 218 RTS
219
*****
220 ;
221 ;
222 ;
DAD5 A903 223 DOH1E LDA #3
DAD7 8D12BF 224 STA ESCFLG
DADA 60 225 RTS
226 endi f
227
*****
228 ;
229 ;
230 ;
DADB A000 231 DOCR LDY #0
DADD 84F4 232 STY HPOS
233 ;
234 ;
235 ;
DADF 60 236 RTS
237
*****
238 ;
239 ;
240 ;
DAE0 A5F5 241 DOLF LDA VPOS
DAE2 C917 242 CMP #23
DAE4 F005 243 BE INTMA
244 ;
245 ;
246 ;
DAE6 E6F5 247 INTLZ INC VPOS
248 ;
249 ;
250 ;
251 ifndef ORIG
252 BNM INTMJ
253 else
DAE8 4CCDB 254 JMP CVP2SP
255 endi f
256 ;
257 ;
258 ;
DAEB A5F4 259 INTMA LDA HPOS
DAED 48 260 PHA
261 ;
262 ;
263 ;
DAEE A200 264 LDX #0

```

```

DAFO 86F5      265      STX   VPOS
                266      ;
                267      ;
                268      ;
DAF2 20CCDB    269      JSR   CVP2SP
                270      ;
                271      ;
                272      ;
                273      INTMB
                274      i fndef ORIG
                275      LDY   #3
                276      LOOP  LDA   SLLPTR, Y
                277      STA   AC1, Y
                278      DEY
                279      BNM   LOOP
                280      el se
DAF5 20EFDB    281      JSR   INTNP
                282      endi f
                283      ;
                284      ;
                285      ;
DAF8 E8        286      INX
DAF9 86F5      287      STX   VPOS
                288      ;
                289      ;
                290      ;
DAFB 20CCDB    291      JSR   CVP2SP
                292      ;
                293      ;
                294      ;
DAFE A027      295      LDY   #39
DBO0 B1F0      296      INTMC  LDA   (SLLPTR), Y
DBO2 91C1      297      STA   (AC1), Y
DBO4 B1F2      298      LDA   (SLRPTR), Y
DBO6 91C3      299      STA   (AC3), Y
DBO8 88        300      DEY
DBO9 10F5      301      BNM   INTMC
                302      ;
                303      ;
                304      ;
DBOB E017      305      CPX   #23
DBOD DOE6      306      BNE   INTMB
                307      ;
                308      ;
                309      ;
                310      i fndef ORIG
                311      JSR   DOCR
                312      el se
DBOF A900      313      LDA   #0
DB11 85F4      314      STA   HPOS
                315      endi f
                316      ;
                317      ;
                318      ;
DB13 206ADB    319      JSR   INTML
                320      ;
                321      ;

```



```

322 ;
DB16 68      323   PLA
DB17 85F4    324   STA   HPOS
              325 ;
              326 ;   Return to caller
              327 ;
DB19 60      328   RTS
              329 ; .PAGE
              330 ;
              331 ;   Delay 64 periods
              332 ;
DB1A A940    333 DOBEL   LDA   #64
DB1C 2066D7  334   JSR   TDELAY
              335 ;
              336 ;   Set loop control value
              337 ;
DB1F A0C0    338   LDY   #192
              339 ;
              340 ;   Delay 12 periods
              341 ;
DB21 A90C    342 INTME   LDA   #12
DB23 2066D7  343   JSR   TDELAY
              344 ;
              345 ;   Toggle speaker
              346 ;
DB26 AD30C0  347   LDA   AC030
              348 ;
              349 ;   Decrement loop control value
              350 ;
DB29 88      351   DEY
              352 ;
              353 ;   Branch if loop not done
              354 ;
DB2A D0F5    355   BNZ   INTME
              356 ;
              357 ;   Retrun to caller
              358 ;
              359 ; *INTMF
DB2C 60      360 ANRTS   RTS
              361
*****
              362 ;
              363 ;
              364 ;
DB2D A5F5    365 CURSUP  LDA   VPOS
DB2F F0FB    366   BZ    ANRTS ; INTMF
              367 ;
              368 ;
              369 ;
DB31 C6F5    370   DEC   VPOS
              371   ifndef ORIG
              372   BNM   INTMJ
              373   else
DB33 4CCDB   374   JMP   CVP2SP
              375   endif
              376
*****

```

```

377 ;
378 ;
379 ;
DB36 C000 380 DOBS   CPY   #0
DB38 F002 381   BE   INTMG
382 ;
383 ;
384 ;
DB3A C6F4 385   DEC   HPOS
386 ;
387 ;
388 ;
DB3C 60   389 INTMG   RTS
390
*****
391 ;
392 ;
393 ;
DB3D 2043DB 394 DOFF   JSR   HOME C
395 ;
396 ;
397 ;
398   i fdef ORIG
DB40 4C4CDB 399   JMP   DOVT
400   endi f
401   i fdef ORIG
402
*****
403 ;
404 ;
405 ;
DB43 A000 406 HOME C   LDY   #0
DB45 84F5 407   STY   VPOS
DB47 84F4 408   STY   HPOS
DB49 4CCCDB 409   JMP   CVP2SP
410   endi f
411
*****
412 ;
413 ;
414 ;
DB4C A5F4 415 DOVT   LDA   HPOS
DB4E 48   416   PHA
DB4F A5F5 417   LDA   VPOS
DB51 48   418   PHA
DB52 206ADB 419 INTMK   JSR   INTML
DB55 20DBDA 420   JSR   DOCR
DB58 20E6DA 421   JSR   INTLZ
DB5B A5F5 422   LDA   VPOS
DB5D C918 423   CMP   #24
DB5F 90F1 424   BL   INTMK
DB61 68   425   PLA
DB62 85F5 426   STA   VPOS
DB64 68   427   PLA
428   i fndef ORIG
429   BNM   SETHP
430 ;

```

```

431 ;
432 ;
433 HOMECLDA #0
434 STA VPOS
435 SETHP
436 endi f
DB65 85F4 437 STA HPOS
438 i fndef ORIG
439 INTMJ
440 endi f
DB67 4CCCDB 441 JMP CVP2SP
442 ;
443 ;
444 ;
DB6A A6F4 445 INTML LDX HPOS
DB6C A9A0 446 INTMN LDA #SAO
DB6E 2076DB 447 JSR WCADVC
DB71 90F9 448 BNC INTMN
DB73 86F4 449 STX HPOS
DB75 60 450 RTS
451 i fndef ORIG
452 ;. PAGE
453 ;
454 ;
455 ;
456 DOH12 BIT ABF31
457 BM GA
458 ;
459 ;
460 ;
461 NDEA9 LDA #S80
462 STA AEO
463 ;
464 ;
465 ;
466 GA RTS
467
*****
468 ;
469 ;
470 ;
471 DOH14 STA AE1
472 ASL A
473 STA AEO
474 ;
475 ;
476 ;
477 RTS
478
*****
479 ;
480 ;
481 ;
482 DOH05 ASL A
483 DOH06 STA ABF12
484 ;
485 ;

```

```

486 ;
487   RTS
488
*****
489
490 ;
491 ;
492 ;
493 DOHOF   ASL   A
494 DOHOE   STA   INTLY+1
495 ;
496 ;
497 ;
498   RTS
499   endi f
500 ;.PAGE 'Write character to screen'
501 ;
502 ;   Write character to screen
503 ;
DB76 2083DB 504 WCADVC   JSR   WRCH2S
505 ;
506 ;   Branch if cursor at end of line
507 ;
DB79 A4F4   508 ADVC     LDY   HPOS
DB7B C04F   509 CPY     #79
DB7D B003   510 BGE     INTMQ
511 ;
512 ;   Advance cursor
513 ;
DB7F C8     514 INY
DB80 84F4   515 STY     HPOS
516 ;
517 ;   Return to caller
518 ;
DB82 60     519 INTMQ   RTS
520
*****
521 ;
522 ;   Save character on stack
523 ;
DB83 48     524 WRCH2S  PHA
525 ;
526 ;   Branch if null character
527 ;
528   ifn def ORIG
529     TAY
530     BZ   INTMT
531   el se
DB84 C900   532     CMP   #0
DB86 F014   533     BE   INTMT
534   endi f
535 ;
536 ;
537 ;
DB88 24E0   538     BIT   AEO
DB8A 1010   539     BNM  INTMT
540 ;

```

```

541 ;
542 ;
DB8C A5F6 543 LDA TEMP
DB8E 297F 544 AND #$7F
DB90 C941 545 CMP #' A
DB92 9008 546 BL INTMT
DB94 C95B 547 CMP #' Z+1
DB96 B004 548 BGE INTMT
DB98 68 549 PLA
DB99 4980 550 EOR #$80
DB9B 48 551 PHA
552 ;
553 ;
554 ;
DB9C A5F4 555 INTMT LDA HPOS
DB9E 38 556 SEC
DB9F ED11BF 557 SBC NBF11
DBA2 300F 558 BM INTMY
DBA4 C928 559 CMP #40
DBA6 B011 560 BGE INTMZ
DBA8 A8 561 TAY
DBA9 68 562 PLA
DBAA D004 563 BNZ INTMX
DBAC B1F0 564 LDA (SLLPTR), Y
DBAE 4980 565 EOR #$80
DBBO 91F0 566 INTMX STA (SLLPTR), Y
567 ;
568 ; Return to caller
569 ;
DBB2 60 570 RTS
571 ;
572 ;
573 ;
DBB3 18 574 INTMY CLC
DBB4 6928 575 ADC #40
576 ifndef ORIG
577 BNM INTNA
578 else
DBB6 4CBCDB 579 JMP INTNA
580 endif
DBB9 38 581 INTMZ SEC
DBBA E928 582 SBC #40
DBBC A8 583 INTNA TAY
DBBD 68 584 PLA
DBBE D004 585 BNZ INTNB
DBC0 B1F2 586 LDA (SLRPTR), Y
DBC2 4980 587 EOR #$80
DBC4 91F2 588 INTNB STA (SLRPTR), Y
589 ;
590 ; Return to caller
591 ;
DBC6 60 592 RTS
593 ifdef ORIG
594 ;
595 ;
596 ;
DBC7 A900 597 INVERT LDA #0

```

```

DBC9 4C83DB    598      JMP    WRCH2S
                599      endi f
                600      ;.PAGE 'Convert vertical position to screen addresses'
                601      ;
                602      ;
                603      ;
DBCC A5F5      604      CVP2SP  LDA    VPOS
DBCE 4A        605      LSR    A
DBCF 2903      606      AND    #$03
DBD1 0904      607      ORA    #$04
DBD3 85F1      608      STA    SLLPTR+1
DBD5 A5F5      609      LDA    VPOS
DBD7 2918      610      AND    #$18
DBD9 9002      611      BNC    INTNO
DBDB 697F      612      ADC    #128-1
DBDD 85F0      613      INTNO  STA    SLLPTR
DBDF 0A        614      ASL    A
DBE0 0A        615      ASL    A
DBE1 05F0      616      ORA    SLLPTR
DBE3 85F0      617      STA    SLLPTR
DBE5 85F2      618      STA    SLRPTR
DBE7 A5F1      619      LDA    SLLPTR+1
DBE9 18        620      CLC
DBEA 6904      621      ADC    #<1024
DBEC 85F3      622      STA    SLRPTR+1
                623      ifndef ORIG
                624      INTNQ
                625      endi f
                626      ;
                627      ; Return to caller
                628      ;
DBEE 60        629      RTS
                630      ifdef ORIG
                631      ;
                632      ;
                633      ;
DBEF A5F1      634      INTNP  LDA    SLLPTR+1
DBF1 85C2      635      STA    AC1+1
DBF3 A5F3      636      LDA    SLRPTR+1
DBF5 85C4      637      STA    AC3+1
DBF7 A5F0      638      LDA    SLLPTR
DBF9 85C1      639      STA    AC1
DBFB 85C3      640      STA    AC3
                641      ;
                642      ; Return to caller
                643      ;
DBFD 60        644      INTNQ  RTS
                645      endi f
                646      ;
                647      ;
                648      ;
DBFE AA        649      INTNR  TAX
DBFF FOFC      650      BZ    INTNQ
DC01 A5F5      651      LDA    VPOS
DC03 48        652      PHA
DC04 8A        653      TXA
DC05 48        654      PHA

```

```

DC06 38          655      SEC
DC07 E928       656      SBC      #40
DC09 1003       657      BNM      INTNT
DC0B 18         658      CLC
DC0C 6950       659      ADC      #80
DC0E 48         660      INTNT    PHA
                661      ;
                662      ;
                663      ;
DC0F A917       664      LDA      #23
DC11 85F5       665      STA      VPOS
                666      ;
                667      ;
                668      ;
DC13 20CCDB     669      INTNU    JSR      CVP2SP
                670      ;
                671      ;
                672      ;
DC16 A027       673      LDY      #39
DC18 B1F0       674      INTNV    LDA      (SLLPTR), Y
DC1A 990002     675      STA      A0200, Y
DC1D B1F2       676      LDA      (SLRPTR), Y
DC1F 992802     677      STA      A0200+40, Y
DC22 88         678      DEY
DC23 10F3       679      BNM      INTNV
                680      ;
                681      ;
                682      ;
DC25 68         683      PLA
DC26 48         684      PHA
DC27 AA         685      TAX
                686      ;
                687      ;
                688      ;
                689      ifndef ORIG
                690      LDA      #SLLPTR
                691      JSR      INTNS
                692      LDA      #SLRPTR
                693      JSR      INTNS
                694      else
DC28 A027       695      LDY      #39
DC2A CA         696      INTNW    DEX
DC2B 1002       697      BNM      INTNX
DC2D A24F       698      LDX      #79
DC2F BD0002     699      INTNX    LDA      A0200, X
DC32 91F0       700      STA      (SLLPTR), Y
DC34 88         701      DEY
DC35 10F3       702      BNM      INTNW
DC37 A027       703      LDY      #39
DC39 CA         704      INTNY    DEX
DC3A 1002       705      BNM      INTNZ
                706      ;
                707      ;
                708      ;
DC3C A24F       709      LDX      #79
DC3E BD0002     710      INTNZ    LDA      A0200, X
DC41 91F2       711      STA      (SLRPTR), Y

```

```

DC43 88          712      DEY
DC44 10F3       713      BNM      INTNY
                714      endi f
DC46 C6F5       715      DEC      VPOS
DC48 10C9       716      BNM      INTNU
                717      ;
                718      ;
                719      ;
DC4A 68         720      PLA
DC4B 68         721      PLA
                722      ;
                723      ;
                724      ;
DC4C 18         725      CLC
DC4D 6D11BF     726      ADC      NBF11
DC50 8D11BF     727      STA      NBF11
DC53 68         728      PLA
DC54 85F5       729      STA      VPOS
                730      ;
                731      ;
                732      ;
DC56 4CCDB      733      JMP      CVP2SP
                734      i fndef ORIG
                735      ;
                736      ;
                737      ;
                738      INTNS   STA      GB+1
                739      LDY      #39
                740      INTNW   DEX
                741      BNM      INTNX
                742      LDX      #79
                743      INTNX   LDA      A0200, X
                744      GB STA   (A00), Y
                745      DEY
                746      BNM      INTNW
                747      ;
                748      ;
                749      ;
                750      RTS
                751      endi f
                752      ;
                753      ;
                754      ;
DC59 AD15BF     755      INTOA   LDA      CNFLGS
DC5C 4A         756      LSR      A
DC5D 9013       757      BNC      INTOC
DC5F A5F4       758      LDA      HPOS
DC61 C914       759      CMP      #20
DC63 B003       760      BGE      INTOB
                761      ;
                762      ;
                763      ;
DC65 4CE5FF     764      JMP      TAB3
                765      ;
                766      ;
                767      ;
DC68 38         768      INTOB   SEC

```



```

DC69 ED11BF      769      SBC      NBF11
DC6C 3014        770      BM       INTOF
DC6E C925        771      CMP       #37
                772      ifndef  ORIG
                773      BL       INTOC
                774      else
DC70 B001        775      BGE      INTOD
DC72 60          776      INTOC     RTS
                777      INTOD
                778      endif
                779      ;
                780      ;
                781      ;
DC73 A4F4        782      LDY      HPOS
DC75 C04D        783      CPY      #77
DC77 B006        784      BGE      INTOE
DC79 38          785      SEC
DC7A E924        786      SBC      #36
                787      ifndef  ORIG
                788      INTOF     JMP      INTNR
                789      INTOE     JMP      AFFE1
                790      INTOC     RTS
                791      else
DC7C 4CFEDB      792      JMP      INTNR
DC7F 4CE1FF      793      INTOE     JMP      AFFE1
DC82 4CFEDB      794      INTOF     JMP      INTNR
                795      endif
                796      ifndef  ORIG
                797      ;
                798      ;
                799      ;
                800      CCHTBL   DB      $1B      ; Escape
                801      DB      $1E      ;
                802      DB      $0D      ; Carriage return
                803      DB      $0A      ; Linefeed
                804      DB      $07      ; Bell character
                805      DB      $1F      ; Cursor up
                806      DB      $1C      ;
                807      DB      $08      ; Backspace
                808      DB      $0C      ; Form feed
                809      DB      $19      ;
                810      DB      $0B      ; Vertical tab
                811      DB      $1D      ;
                812      DB      $12      ;
                813      DB      $14      ;
                814      DB      $05      ;
                815      DB      $06      ;
                816      DB      $0E      ;
                817      DB      $0F      ;
                818      ENDCCT
                819      ;
                820      ;
                821      ;
                822      INTLJ    DW      DOESC
                823      DW      DOH1E
                824      DW      DOCR
                825      DW      DOLF

```

```

826     DW     DOBEL
827     DW     CURSUP
828     DW     ADVC
829     DW     DOBS
830     DW     DOFF
831     DW     HOMECE
832     DW     DOVT
833     DW     INTML
834     DW     DOH12
835     DW     DOH14
836     DW     DOH05
837     DW     DOH06
838     DW     DOHOE
839     DW     DOHOF
840     ;
841     ;
842     ;
843     DS     8
844     endif
87     lst ON
88     INCLUDE STATUSO.IAS
      1 ;
      2 ; Save return address
      3 ;
DC85 68      4 XARGOS PLA
DC86 85BD    5 STA ABD
DC88 68      6 PLA
DC89 85BE    7 STA ABD+1
      8 ;
      9 ; Discard integer value
     10 ;
DC8B 68     11 PLA
DC8C 68     12 PLA
     13 ;
     14 ; Extract pointer from stack and save it
     15 ;
DC8D 68     16 PLA
DC8E 85D8   17 STA AD8
DC90 68     18 PLA
DC91 85D9   19 STA AD8+1
     20 ;
     21 ; Extract ??? from stack and save it
     22 ;
DC93 68     23 PLA
DC94 85D6   24 STA AD6
DC96 68     25 PLA
DC97 85D7   26 STA AD6+1
     27 ;
     28 ; Restore return address
     29 ;
DC99 A5BE   30 LDA ABD+1
DC9B 48     31 PHA
DC9C A5BD   32 LDA ABD
DC9E 48     33 PHA
     34 ;
     35 ; Return to caller
     36 ;

```

```

DC9F 60          37      RTS
                 38
*****
                 39 ;
                 40 ; Extract arguents off stack
                 41 ;
DCA0 2085DC     42 STATPR JSR XARGOS
                 43 ;
                 44 ;
                 45 ;
DCA3 A900      46 INTOJ LDA #0
DCA5 A8        47 TAY
DCA6 91D8      48 STA (AD8), Y
DCA8 C8        49 INTOK INY
DCA9 91D8      50 STA (AD8), Y
                 51 ;
                 52 ;
                 53 ;
DCAB A200      54 INTOL LDX #0
                 55 ;
                 56 ;
                 57 ;
DCAD 4CBBFF    58 JMP AFFBB
                 59
*****
                 60 ;
                 61 ; Extract arguents off stack
                 62 ;
DCB0 2085DC     63 STATC JSR XARGOS
                 64 ;
                 65 ;
                 66 ;
DCB3 A5D6      67 LDA AD6
DCB5 FOEC      68 BZ INTOJ
                 69 ;
                 70 ;
                 71 ;
DCB7 38        72 SEC
DCB8 AD19BF    73 LDA WPTR
DCBB ED18BF    74 SBC RPTR
DCBE A000      75 LDY #0
DCC0 91D8      76 STA (AD8), Y
                 77 ;
                 78 ;
                 79 ;
DCC2 98        80 TYA
                 81 ;
                 82 ; Unconditi onal branch
                 83 ;
DCC3 FOE3      84 BZ INTOK
                 85
*****
                 86 ;
                 87 ; Extract arguents off stack
                 88 ;
DCC5 2085DC     89 STATD JSR XARGOS
                 90 ;

```

```

91 ;
92 ;
DCC8 A900 93 LDA #0
DCCA A007 94 LDY #8-1
DCCC 91D8 95 INTOM STA (AD8), Y
DCCE 88 96 DEY
DCCF 10FB 97 BNM INTOM
98 ;
99 ;
100 ;
DCD1 A003 101 LDY #3
DCD3 A901 102 LDA #$01
DCD5 91D8 103 STA (AD8), Y
DCD7 C8 104 INY
DCD8 A910 105 LDA #$10
DCDA 91D8 106 STA (AD8), Y
107 ;
108 ;
109 ;
DCDC A006 110 LDY #6
DCDE A923 111 LDA #$23
DCEO 91D8 112 STA (AD8), Y
113 ;
114 ; Unconditional branch
115 ;
DCE2 DOC7 116 BNZ INTOL
89 INCLUDE INCLDS.IAS
1 ;.PAGE 'ID search'
2 ;
3 ;
4 ;
DCE4 68 5 ADCE4 PLA
DCE5 8574 6 STA A74
DCE7 68 7 PLA
DCE8 8575 8 STA A74+1
9 ;
10 ;
11 ;
DCEA 68 12 PLA
DCEB 858A 13 STA A8A
DCED 68 14 PLA
DCEE 858B 15 STA A8A+1
16 ;
17 ;
18 ;
DCFO 68 19 PLA
DCF1 858C 20 STA A8C
DCF3 68 21 PLA
DCF4 858D 22 STA A8C+1
23 ;
24 ;
25 ;
DCF6 A000 26 LDY #0
DCF8 18 27 CLC
DCF9 B18C 28 LDA (A8C), Y
DCFB 658A 29 ADC A8A
DCFD 858E 30 STA A8E

```

```

DCFF C8          31      INY
DD00 B18C       32      LDA      (A8C), Y
DD02 658B       33      ADC      A8A+1
DD04 858F       34      STA      A8E+1
                35      ;
                36      ;
                37      ;
DD06 A920       38      LDA      #$20
DD08 857E       39      STA      A7E
DD0A 857F       40      STA      A7E+1
DD0C 8580       41      STA      A7E+2
DD0E 8581       42      STA      A7E+3
DD10 8582       43      STA      A7E+4
DD12 8583       44      STA      A7E+5
DD14 8584       45      STA      A7E+6
DD16 8585       46      STA      A7E+7
                47      ;
                48      ;
                49      ;
DD18 A000       50      LDY      #0
DD1A A200       51      LDX      #0
DD1C B18E       52      LDA      (A8E), Y
DD1E C961       53      CMP      #' a
DD20 9007       54      BL      INTON
DD22 C97B       55      CMP      #' z+1
DD24 B003       56      BGE      INTON
DD26 38         57      SEC
DD27 E920       58      SBC      #$20
DD29 857E       59      INTON   STA      A7E
DD2B C8         60      INTOP   INY
                61      ;
                62      ;
                63      ;
DD2C B18E       64      LDA      (A8E), Y
DD2E C961       65      CMP      #' a
DD30 900A       66      BL      INTOS
DD32 C97B       67      CMP      #' z+1
DD34 B006       68      BGE      INTOS
DD36 38         69      SEC
DD37 E920       70      SBC      #$20
                71      ;
                72      ;
                73      ;
DD39 4C54DD     74      JMP      INTOW
                75      ;
                76      ;
                77      ;
DD3C C930       78      INTOS   CMP      #' 0
DD3E 9006       79      BL      INTOU
DD40 C93A       80      CMP      #' 9+1
DD42 B002       81      BGE      INTOU
DD44 900E       82      BL      INTOW
DD46 C941       83      INTOU   CMP      #' A
DD48 9014       84      BL      INTOZ
DD4A C95B       85      CMP      #' Z+1
DD4C 9006       86      BL      INTOW
DD4E C95F       87      CMP      #$5F

```

DD50	D00C	88	BNE	INTOZ
DD52	F0D7	89	BE	INTOP
DD54	E8	90	INTOW	INX
DD55	E008	91	CPX	#8
DD57	B0D2	92	BGE	INTOP
DD59	957E	93	STA	A7E, X
DD5B	4C2BDD	94	JMP	INTOP
DD5E	88	95	INTOZ	DEY
DD5F	98	96	TYA	
DD60	A000	97	LDY	#0
DD62	18	98	CLC	
DD63	718C	99	ADC	(A8C), Y
DD65	918C	100	STA	(A8C), Y
DD67	C8	101	INY	
DD68	B18C	102	LDA	(A8C), Y
DD6A	6900	103	ADC	#0
DD6C	918C	104	STA	(A8C), Y
DD6E	A57E	105	LDA	A7E
DD70	38	106	SEC	
DD71	E941	107	SBC	#\$41
DD73	0A	108	ASL	A
DD74	A8	109	TAY	
DD75	B9F5DD	110	LDA	RWREFS, Y
DD78	8588	111	STA	A88
DD7A	C8	112	INY	
DD7B	B9F5DD	113	LDA	RWREFS, Y
DD7E	8589	114	STA	A88+1
DD80	A000	115	LDY	#0
DD82	B188	116	LDA	(A88), Y
DD84	8586	117	STA	A86
DD86	A200	118	LDX	#0
DD88	A001	119	LDY	#1
DD8A	E8	120	INTPB	INX
DD8B	C8	121	INY	
DD8C	B188	122	LDA	(A88), Y
DD8E	D57E	123	CMP	A7E, X
DD90	D006	124	BNE	INTPD
DD92	E007	125	CPX	#7
DD94	DOF4	126	BNE	INTPB
DD96	F016	127	BE	INTPF
DD98	C686	128	INTPD	DEC A86
DD9A	F034	129	BZ	INTPG
DD9C	A200	130	LDX	#0
DD9E	A001	131	LDY	#1
DDA0	A588	132	LDA	A88
DDA2	18	133	CLC	
DDA3	690B	134	ADC	#\$0B
DDA5	8588	135	STA	A88
DDA7	9002	136	BNC	INTPE
DDA9	E689	137	INC	A88+1
DDAB	4C8ADD	138	INTPE	JMP INTPB
DDAE	C8	139	INTPF	INY
DDAF	B188	140	LDA	(A88), Y
DDB1	857C	141	STA	A7C
DDB3	C8	142	INY	
DDB4	B188	143	LDA	(A88), Y
DDB6	857D	144	STA	A7C+1

```

DDB8 A002      145      LDY      #2
DDBA A57C      146      LDA      A7C
DDBC 918C      147      STA      (A8C), Y
DDBE C8        148      INY
DDBF A900      149      LDA      #0
DDC1 918C      150      STA      (A8C), Y
DDC3 C8        151      INY
DDC4 A57D      152      LDA      A7C+1
DDC6 918C      153      STA      (A8C), Y
DDC8 C8        154      INY
DDC9 A900      155      LDA      #0
DDCB 918C      156      STA      (A8C), Y
DDCD 4CEEDD    157      JMP      INTPI
DDDO A002      158      INTPG   LDY      #2
DDD2 A900      159      LDA      #0
DDD4 918C      160      STA      (A8C), Y
DDD6 C8        161      INY
DDD7 918C      162      STA      (A8C), Y
DDD9 C8        163      INY
DDDA C8        164      INY
DDDB 918C      165      STA      (A8C), Y
DDDD 88        166      DEY
DDDE A915      167      LDA      #$15
DDE0 918C      168      STA      (A8C), Y
DDE2 A00E      169      LDY      #SOE
DDE4 A207      170      LDX      #7
DDE6 88        171      INTPH   DEY
DDE7 B57E      172      LDA      A7E, X
DDE9 918C      173      STA      (A8C), Y
DDEB CA        174      DEX
DDEC 10F8      175      BNM      INTPH
DDEE A575      176      INTPI   LDA      A74+1
DDF0 48        177      PHA
DDF1 A574      178      LDA      A74
DDF3 48        179      PHA
DDF4 60        180      RTS

```

\*\*\*\*\*

```

182 ;
183 ;
184 ;
DDF5 2CDE      185      RWREFS  DW      ANDK
DDF7 42DE      186      DW      BEGK
DDF9 4DDE      187      DW      CASK
DDFB 63DE      188      DW      DI VK
DDFD 84DE      189      DW      ENDK
DDFF A5DE      190      DW      FORK
DE01 D1DE      191      DW      GOTOK
DE03 29DE      192      DW      ATSK
DE05 DCDE      193      DW      IFK
DE07 29DE      194      DW      ATSK
DE09 29DE      195      DW      ATSK
DE0B 08DF      196      DW      LABK
DE0D 13DF      197      DW      MODK
DE0F 1EDF      198      DW      NOTK
DE11 29DF      199      DW      OFK
DE13 3FDF      200      DW      PROCK

```

```

DE15 29DE      201    DW    ATSK
DE17 60DF      202    DW    REPK
DE19 76DF      203    DW    SETK
DE1B 97DF      204    DW    THENK
DE1D B8DF      205    DW    UNITK
DE1F D9DF      206    DW    VARK
DE21 E4DF      207    DW    WHILK
DE23 29DE      208    DW    ATSK
DE25 29DE      209    DW    ATSK
DE27 29DE      210    DW    ATSK

```

211

\*\*\*\*\*

```

                212    ;
                213    ;
                214    ;
DE29 00        215    ATSK    DB    $00
DE2A 40        216        ASC    '@'
DE2B 23        217        DB    $23
DE2C 02        218    ANDK    DB    $02
DE2D 414E4420  219        ASC    'AND'
DE35 2702      220        DB    $27, $02
DE37 00        221        DB    $00
DE38 41525241  222        ASC    'ARRAY'
DE40 2C00      223        DB    $2C, $00
DE42 01        224    BEGK    DB    $01
DE43 42454749  225        ASC    'BEGIN'
DE4B 1300      226        DB    $13, $00
DE4D 02        227    CASK    DB    $02
DE4E 43415345  228        ASC    'CASE'
DE56 1500      229        DB    $15, $00
DE58 00        230        DB    $00
DE59 434F4E53  231        ASC    'CONST'
DE61 1C00      232        DB    $1C, $00
DE63 03        233    DIVK    DB    $03
DE64 44495620  234        ASC    'DIV'
DE6C 2703      235        DB    $27, $03
DE6E 00        236        DB    $00
DE6F 444F2020  237        ASC    'DO'
DE77 0600      238        DB    $06, $00
DE79 00        239        DB    $00
DE7A 444F574E  240        ASC    'DOWNT0'
DE82 0800      241        DB    $08, $00
DE84 03        242    ENDK    DB    $03
DE85 454E4420  243        ASC    'END'
DE8D 0900      244        DB    $09, $00
DE8F 00        245        DB    $00
DE90 454C5345  246        ASC    'ELSE'
DE98 0D00      247        DB    $0D, $00
DE9A 00        248        DB    $00
DE9B 45585445  249        ASC    'EXTERNAL'
DEA3 3500      250        DB    $35, $00
DEA5 04        251    FORK    DB    $04
DEA6 464F5220  252        ASC    'FOR'
DEAE 1800      253        DB    $18, $00
DEBO 00        254        DB    $00
DEB1 46494C45  255        ASC    'FILE'
DEB9 2E00      256        DB    $2E, $00

```



DEBB 00	257	DB	\$00
DEBC 464F5257	258	ASC	' FORWARD '
DEC4 2200	259	DB	\$22, \$00
DEC6 00	260	DB	\$00
DEC7 46554E43	261	ASC	' FUNCTION'
DECF 2000	262	DB	\$20, \$00
DED1 01	263	GOTOK	DB \$01
DED2 474F544F	264	ASC	' GOTO '
DEDA 1A00	265	DB	\$1A, \$00
DEDC 04	266	IFK	DB \$04
DEDD 49462020	267	ASC	' IF '
DEE5 1400	268	DB	\$14, \$00
DEE7 00	269	DB	\$00
DEE8 494D504C	270	ASC	' IMPLEMEN'
DEF0 3400	271	DB	\$34, \$00
DEF2 00	272	DB	\$00
DEF3 494E2020	273	ASC	' IN '
DEFB 290E	274	DB	\$29, \$0E
DEFD 00	275	DB	\$00
DEFE 494E5445	276	ASC	' INTERFAC'
DF06 3300	277	DB	\$33, \$00
DF08 01	278	LABK	DB \$01
DF09 4C414245	279	ASC	' LABEL '
DF11 1B00	280	DB	\$1B, \$00
DF13 01	281	MODK	DB \$01
DF14 4D4F4420	282	ASC	' MOD '
DF1C 2704	283	DB	\$27, \$04
DF1E 01	284	NOTK	DB \$01
DF1F 4E4F5420	285	ASC	' NOT '
DF27 2600	286	DB	\$26, \$00
DF29 02	287	OFK	DB \$02
DF2A 4F462020	288	ASC	' OF '
DF32 0B00	289	DB	\$0B, \$00
DF34 00	290	DB	\$00
DF35 4F522020	291	ASC	' OR '
DF3D 2807	292	DB	\$28, \$07
DF3F 03	293	PROCK	DB \$03
DF40 50524F43	294	ASC	' PROCEDUR'
DF48 1F00	295	DB	\$1F, \$00
DF4A 00	296	DB	\$00
DF4B 50524F47	297	ASC	' PROGRAM '
DF53 2100	298	DB	\$21, \$00
DF55 00	299	DB	\$00
DF56 5041434B	300	ASC	' PACKED '
DF5E 2B00	301	DB	\$2B, \$00
DF60 02	302	REPK	DB \$02
DF61 52455045	303	ASC	' REPEAT '
DF69 1600	304	DB	\$16, \$00
DF6B 00	305	DB	\$00
DF6C 5245434F	306	ASC	' RECORD '
DF74 2D00	307	DB	\$2D, \$00
DF76 03	308	SETK	DB \$03
DF77 53455420	309	ASC	' SET '
DF7F 2A00	310	DB	\$2A, \$00
DF81 00	311	DB	\$00
DF82 5345474D	312	ASC	' SEGMENT '
DF8A 2100	313	DB	\$21, \$00

```

DF8C 00          314    DB    $00
DF8D 53455041   315    ASC    ' SEPARATE'
DF95 3600       316    DB    $36, $00
DF97 03         317    THENK  DB    $03
DF98 5448454E   318    ASC    ' THEN '
DFA0 0C00       319    DB    $0C, $00
DFA2 00         320    DB    $00
DFA3 544F2020   321    ASC    ' TO '
DFAB 0700       322    DB    $07, $00
DFAD 00         323    DB    $00
DFAE 54595045   324    ASC    ' TYPE '
DFB6 1D00       325    DB    $1D, $00
DFB8 03         326    UNITK  DB    $03
DFB9 554E4954   327    ASC    ' UNIT '
DFC1 3200       328    DB    $32, $00
DFC3 00         329    DB    $00
DFC4 554E5449   330    ASC    ' UNTIL '
DFCC 0A00       331    DB    $0A, $00
DFCE 00         332    DB    $00
DFCF 55534553   333    ASC    ' USES '
DFD7 3100       334    DB    $31, $00
DFD9 01         335    VARK   DB    $01
DFDA 56415220   336    ASC    ' VAR '
DFE2 1E00       337    DB    $1E, $00
DFE4 02         338    WHI LK DB    $02
DFE5 5748494C   339    ASC    ' WHILE '
DFED 1700       340    DB    $17, $00
DFEF 00         341    DB    $00
DFF0 57495448   342    ASC    ' WITH '
DFF8 1900       343    DB    $19, $00
                344    ;
                345    ;
                346    ;
DFFA 00         347    HSL0T2 DB    $00
DFFB 0000       348    CURTRK DB    $00, $00
DFFD 00         349    ADFFD  DB    $00
DFFE 00         350    ADFFE  DB    $00
DFFF 00         351    ADFFF  DB    $00

```

###