

## CHAPTER 5: DEVELOPMENT TOOLS AND UTILITIES

1. SUPER GAMES

This section explains background loading, tape mapping and playability of super games and other programs that use OS\_7. Familiarity with OS\_7 is assumed. The information and examples in this section should help programmers who are accustomed to cartridge software adapt to tape-based software more easily.

1.1 Background Loading

The background loading software (see Subsection 1.7) is designed to load overlays from data pack to RAM while the program is executing. Care must be taken that data being loaded during execution does not destroy data that is controlling execution. A good approach is to use two buffers. One buffer is loaded while the other is controlling program execution.

1.2 Timing Considerations

Each background block read takes about one second assuming no retries and no repositioning. Retries take about 1.2 to 1.4 seconds and only happen if the checksum fails to compare on read. There are a maximum of three attempts to read a block on the tape, then the checksum failed code is sent. Repositioning takes as long as one second to find the current position on tape (approximately one second for every 80 inches of tape travel). Repositioning is automatically handled by the drive. For the NMI driven tape manager, additional overhead in transfer time averages 8 milliseconds (1/2 60hz clock tick). Buffering data for transfer to VDP-RAM results in Z80 CPU usage assuming use of WRITE\_VRAM. The following table shows the Z80 CPU time used in VDP-RAM writes.

TABLE 4: Z-80 CPU TIME FOR VDP-RAM WRITES

Number of bytes to transfer	Milliseconds
1	0.077
10	0.175
100	1.16
1000	11.25
2000	22.4
4000	44.8
8000	89.6

1.3 Mapping

The programmer must lay out the data to minimize load time, through mapping. Minimizing load time is the most crucial aspect of designing super games.

Since the data on tape is a memory image, it may be read directly into Z80 RAM and immediately executed. The programmer should diagram the tape blocks on a time line, following the timing considerations in Subsection 1.2. Some action on the screen must hide the loading process. Rewinding and positioning time must be included in calculating load size and load time.

An example of a time line diagram follows. It shows the screen actions and what is loaded in background while the action is taking place.

TIME IN SECONDS	SCREEN	LOADING	ADDRESS	BLOCK NUMBER
0	BLANK	COLD START MAIN	C800H	0
1			8000H	1
2			8400	2
3			8800	3
4	TITLE SCREEN	OVERLAY 3	8C00	4
5			9000	5
6			9400	6
7			9800	7
8	LOGO SCREEN		9C00	8
9			A000	9
10			A400	10
11			A800	11
12	SELECT OPTIONS		AC00	12
13			B000	13
14			B400	14
15			B800	15
16	GAME START	[*REWIND TO TRACK 1 SECTOR 1 5 SECONDS OF REWIND TIME FOLLOW]	BC00	16
17				128
18				
19				
20		OVERLAY 4		
21				
22			C000	129
23			C400	130
24		ETC...	C800	131
25			CC00	132

TIME LINE DIAGRAM

Overlay Control Blocks



To read memory images into RAM, three pieces of information are required. They are:

Start of memory buffer of transfer address	2 bytes
Start of memory image on tape or block number	2 bytes
Number of 1K Blocks in the memory image	1 byte

This information is organized into a 5-byte block called an Overlay Control Block (OCB). The overlay control blocks are organized into an Overlay Control Table (OCT). The OCT controls the loading of data from tape using the tape interface software described in Subsection 1.6. The table should be pointed to by a two byte pointer and it should be terminated by a byte set to FF. The Overlay Control Table should be loaded immediately after the cold start loader.

#### 1.4 Start of Game

EOS loads Block 0 (the cold start loader) to C800H and passes control to it. The cold start loader then initializes the system and loads enough of the main program to allow some user interaction to begin. The following rules define the interface to the cold start loader.

1. The main program is loaded to location 8000H.
2. Immediately following the ColecoVision OS vectors at 8000H, and the game name, is a pointer to the OCT. The first entry in the OCT describes the main program.
3. Control is passed to the main program by the vector at address 800AH as defined in the OS\_7 PRIME.
4. The main program must contain the OCT in the first 1K block.
5. The main program must contain the background loading routines described in Subsection 1.7. They are required to reside in the first 3K.
6. The main program must immediately display graphics or allow some user interaction.
7. Once control is passed to the main program, game play must start as soon as possible.
8. When control is transferred to the main program, Register B contains the boot device ID. Register B should be stored at the globally defined address DEVICE\_ID.

Subsection 1.5 is an example of the 8000H area code which interfaces to the cold start loader. Compare this to the area defined in OS\_7.

1.5

8000H Area Code- Interface to Cold Start Loader

FILE: CART:ROB  
LOCATION OBJECT CODE LINE SOURCE LINE

NEWLETT-PACKARD: CART (c) Coleco 1984

Fri, 18 May 1984, 14:59

PAGE 1

```

1  *280*
2
3
4  ; This is an example of the DCB pointer following the game name
5  ; section and how to store the boot device number for future use.
6  ; The RAM at 8000H is defined in greater detail in the ColecoVision
7  ; Programmers Manual.
8
9
10  ORG      8000H
11  EQU      0000
12  ; ** CARTRIDGE SOFTWARE POINTERS 8000H **
13  ;
14  MEX      55-AA
15  DEFN     ---
16  DEFN     ---
17  DEFN     ---
18  DEFN     ---
19  DEFN     MAIN_PROG
20
21  ; *****
22  ; *****
23  JP      RST 08
24  JP      RST 10
25  JP      RST 18
26  JP      RST 20
27  JP      RST 28
28  JP      RST 30
29
30  ; THIS IS THE MASKABLE INTERRUPT SOFT VECTOR
31  JP      MASK_INT
32
33  ; THIS IS THE MMI SOFT VECTOR.
34  JP      VDPINT
35
36  ; *****
37  ; *****
38  ; ** Game name section **
39
40  DEFN     "SUPER"
41  DEFN     "/GAMES ",IDM,"/1984"
42
43  ;
44  ;
45  ;
46  ; *****
47  ; *****
48  EXT      DCB
49
50  DEFN     DCB
51
52  ; A pointer to the DCB must follow the
53  ; date in the game name.
54  DEV_ID   DEFN     8
55  ; This is the default DEVICE_ID. See MAIN_PROG (below) for applications
56  ; booted from a different device.
57  ; *****
58  ; *****
59
60  ; *****
61  ; *****
62  ; *****
63  ; *****
64  ; *****
65  ; *****
66  ; *****
67  ; *****
68  ; *****
69  ; *****
70  ; *****
71  ; *****
72  ; *****
73  ; *****
74  ; *****
75  ; *****
76  ; *****
77  ; *****
78  ; *****
79  ; *****
80  ; *****
81  ; *****
82  ; *****
83  ; *****
84  ; *****
85  ; *****
86  ; *****
87  ; *****
88  ; *****
89  ; *****
90  ; *****
91  ; *****
92  ; *****
93  ; *****
94  ; *****
95  ; *****
96  ; *****
97  ; *****
98  ; *****
99  ; *****
100 ; *****

```



LOCATION OBJECT CODE LINE SOURCE LINE

```

57 *****
58 *****
59 8039 C9      RET      :Z80 restarts to be defined
60
61 :-----
62 VCPINT      RETN      :Non maskable interrupt processing-
63                :Normally used for critical timing:
64                :      music
65                :      processing timers
66                :      sprite motion processing
67
68 :-----
69 803C ED40    MASK_INT  RETI      :Z80 maskable interrupt vector-
70                :Normally used with spinner switch:
71                :      steering wheel
72                :      sports controller
73                :      roller controller
74
75            EXT      _HARD_INIT,COLDSTART
76 MAIN_PROG    LD
77            LD      A,B
78
79
80            LD      [DEV_ID],A
81
82 : The rest of the application follows...

```

```

: Main prog is entered with the Device ID in the B registe
: If it is not saved, the DDP manager or application will o
: run from Drive A.

```

Errors= 0

FILE: C:\1:R08

LINE# SYMBOL

75 COLDSTART  
53 DEV\_ID  
76 MAIN\_PROG  
69 MASK\_INT  
48 DCB  
59 RST  
62 VDPINT  
75 -HARD\_INIT  
10 ----

CROSS REFERENCE TABLE

TYPE REFERENCES

E 80  
A 19  
A 31  
E 50  
A 23,24,25,26,27,28  
A 34  
E  
A 15,16,17,18

## 1.6 Tape Interface Software

The tape interface software loads RAM from tape in background using the OCT structure defined in Subsection 1.7. The entry points for these modules are described in the following subsections. The tape manager programs are interrupt driven and should be called on every clock cycle (60Hz) to drive background loading. The OCT, DDP\_MANAGER (Subsection 1.10) and TAPE\_INTERFACE (Subsection 1.9) or DDP\_INTERFACE (Subsection 1.11) must be linked into the program.

### 1.6.1 Tape Manager Programs

The tape manager consists of two interchangeable parts dependent upon environment.

#### TAPE\_MANAGER

The program TAPE\_MANAGER (shown in Subsection 1.8) is designed for use on the HP64000. This program allows for simulated tape I/O via the HP disk, and should be used in conjunction with TAPE\_INTERFACE.

#### DDP\_MANAGER

This program, shown in Subsection 1.10, replaces the tape manager in working games. Entry points and interface are identical to the TAPE\_MANAGER except that this module uses EOS calls to manipulate the tape. DDP\_MANAGER should be used in conjunction with DDP\_INTERFACE.

The final version of the game should have the DDP\_MANAGER installed. The tape managers are fully interchangeable.



### 1.6.2 TAPE INTERFACE

TAPE\_INTERFACE consists of a set of entry points and data passed in the accumulator. DDP\_INTERFACE is similar to TAPE\_INTERFACE, except that some labels are different.

<u>Entry Points</u>	<u>Accumulator</u>	<u>Comments</u>
LOAD_OVERLAY	OVERLAY_NUMBER	The code uses the overlay number to look up OCT information. The overlay is loaded from tape to Z80 RAM.
WRITE_OVERLAY	OVERLAY_NUMBER	As in LOAD_OVERLAY, but data flow is from RAM to tape.
ABORT_TAPE	-----	Aborts all tape functions. Resets tape. Makes tape ready to immediately receive new commands. Does not reposition tape.
TEST_TAPE	-----	Returns status of tape.

Status from TEST-TAPE may show the following conditions.

0	OK
1	Checksum failed
2	Block not found
3	Tape not present
4	Device not present

The checksum shows that after three retries, the tape block was not read correctly and the data transferred by the read command is not valid.

1.7 Background Loading Software

FILE: OCB:IDS      MEWLETT-PACKARD: DCS (c) Coleco 1982 Confidential  
LOCATION OBJECT CODE LINE      SOURCE LINE      PAGE 1

1 \*ZRO"  
2 NAME "REV 0 - DTT"  
3  
4 DESCR\_DCB      MACRO      -GOTO ENDESCR\_DCB  
5  
6  
7  
8 Author:      DTT  
9 Project:      WAFER, A132  
10 Starting date: 25mar83  
11  
12 \*\*\*\*\*  
13 \* OCB      \*\*\*\*\*  
14      DTT      \*  
15 \*\*\*\*\*  
16  
17 Rev.      Date      Name      Change  
18      1  
19      0      25mar83      DTT      Initial Pseudo code  
20  
21 NAME: OCB (OVERLAY CONTROL BLOCK TABLE)  
22 THE OCB DESCRIBED BELOW IS AN EXAMPLE ONLY AND DOES NOT DESCRIBE  
23 ANY GAME. I AM SHOWING A MAIN PROGRAM STARTING AT 8000H.  
24 THE BACKUP COPY OF THE COLD START LOADER ALSO GETS LOADED AT 8000H.  
25 THERE ARE 18 OVERLAYS DESCRIBED IN THIS OCT. THE LAST TWO ARE  
26 SPARE. TWO DESCRIBE VANITY SCREEN AND DATA  
27  
28 ENDESCR\_DCB:  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000

0000  
0000  
0000 8000  
0002 0001  
0004 10  
0005

0000H,1,16  
7/12/83 coded and tested  
0000H  
1  
16  
03000H,1,16,16

MACRO      EP1,EP2,EP3  
REV 0 DTT, 7/12/83 coded and tested  
DEFW      EP1  
DEFW      EP2  
DEFB      EP3  
MEND

PROG

OCB      0000H,1,16  
REV 0 DTT, 7/12/83 coded and tested  
DEFW      0000H  
DEFW      1  
DEFB      16  
OCB      03000H,1,16,16

MAIN PROGRAM  
TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED, IN RAM)  
BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
NUMBER OF 1K BLOCKS TO TRANSFER  
BACKUP MAIN PROGRAM

```
0005 8000
0007 0011
0009 10
000A
55
000A 2400
000C 0021
000E 07
000F
56
000F C000
0011 0028
0013 08
0014
57
0014 2400
0016 0030
0018 06
0019
58
0019 C000
001B 0036
001D 08
001E
59
001E 2400
0020 003E
0022 06
0023
60
0023 C000
0025 0044
0027 06
0028
61
0028 2400
002A 004A
002C 04
002D
62
002D 7C00
002F 0081
0031 01
0032
63
0032 C000
0034 008E
0036 0E
0037
64
0037 2400
0039 009C
003B 07
003C
65
003C C000
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 08000H
+ DEF 1+16
+ DEF 16
55 DCB 02400H,1+16+16,7
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 02400H
+ DEF 1+16+16
+ DEF 7
56 DCB 0C000H,1+16+16+7,8
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 0C000H
+ DEF 1+16+16+7
+ DEF 8
57 DCB 02400H,1+16+16+7+8,6
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 02400H
+ DEF 1+16+16+7+8
+ DEF 6
58 DCB 0C000H,1+16+16+7+8+6,8
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 0C000H
+ DEF 1+16+16+7+8+6
+ DEF 8
59 DCB 02400H,1+16+16+7+8+6+8,6
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 02400H
+ DEF 1+16+16+7+8+6+8
+ DEF 6
60 DCB 0C000H,1+16+16+7+8+6+8+6,6
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 0C000H
+ DEF 1+16+16+7+8+6+8+6
+ DEF 6
61 DCB 02400H,1+16+16+7+8+6+8+6+6,4
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 02400H
+ DEF 1+16+16+7+8+6+8+6+6
+ DEF 4
62 DCB 07C00H,128+1,1
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 07C00H
+ DEF 128+1
+ DEF 1
63 DCB 0C000H,128+1+13,14
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 0C000H
+ DEF 128+1+13
+ DEF 14
64 DCB 02400H,128+1+13+14,7
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 02400H
+ DEF 128+1+13+14
+ DEF 7
65 DCB 0C000H,128+1+13+14+7,1
+ * REV 0 DTT. 7/12/83 coded and tested
+ DEF 0C000H
```

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 3

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 4

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 5

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 6

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 7

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 8

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 9

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:REWIND (NEVER ACTUAL EXECUTEABLE CODE)

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 11

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 12

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 13

:TRANSFER ADDRESS OF THE OVERLAY (WHERE DOES IT GET LOADED IN RAM)  
:BLOCK NUMBER OF THE FIRST BLOCK IN THE OVERLAY  
:NUMBER OF 1K BLOCKS TO TRANSFER  
:OVERLAY 13



LOCATION	OBJECT	CODE	LINE	SOURCE	LINE
----------	--------	------	------	--------	------

[illegible]

Errors: 0

FILE: OCB:TDS

LINE# SYMBOL

52 OCB  
35 OCB\_PTR

CROSS REFERENCE TABLE

TYPE REFERENCES

P 34  
E

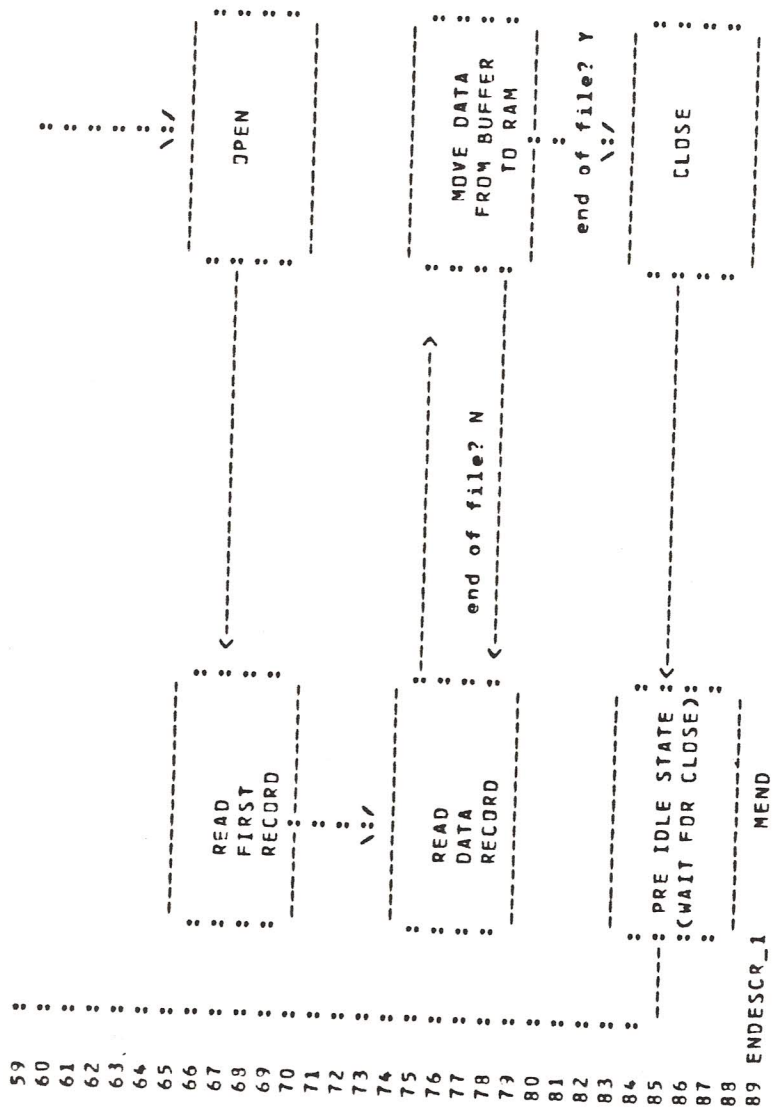
# 1.8 TAPE MANAGER

```

FILE: TAPE_MANAGER:OS      MENLETT-PACKARD: TAPE_MANAGER (c) Coloco 1983 Confidential
LOCATION OBJECT CODE LINE  SOURCE LINE
1 "Z80"
3 NAME "Rev 6 - DTT"
5 DESCR_
6 MACRO
7 -GOTO ENDESCR_1
8 Author: DTT
9 Project:
10 Starting date:10Feb83
11
12 Prom release Date:
13 Prom release Rev:
14
15 Header Rev: 3
16
17 *****
18 *
19 * TAPE_MANAGER DTT *
20 *
21 *****
22
23 Rev History (one line note indicating the change)
24
25 Rev. Date Change
26 6 11/16/83 DTT ADDED DEVICE ID TO SIMULATE DEVICE INDEPENDENCE
27 5 12Sep83 DTT MODIFIED ERROR SYSTEM TO WRITE ERRORS TO CSA AREA
28 4 08Aug83 DTT SIMULATES WRITES!
29 3 07Jul83 DTT ADDED KILL_TAPE/CSA
30 2 30Jun83 DTT STATE MACHINE FOR MULTI TASKING
31 1 05Apr83 DTT BINK ON I/O ERROR, REMOVE TIMING STUFF, EI AND DI ADDED.
32 0 10Feb83 DTT Initial Pseudo code
33
34 NAME: TAPE_MANAGER (OVERLAY CONTROL)
35
36
37 FUNCTION: (LOAD OVERLAYS VIA SIMULATED I/O WITH THE HP64000)
38
39
40 INPUTS: (ACCUMULATOR = OVERLAY NUMBER)
41
42
43 OUTPUTS: (OVERLAY IS LOADED TO RAM ADDRESS)
44
45
46 PSEUDOCODE (PASCAL type pseudocode of procedure.)
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
25
```



LOCATION OBJECT CODE LINE SOURCE LINE



LOCATION OBJECT CODE LINE SOURCE LINE

```
91 DESC_a      MACRO
92 *GOTO      DESC_a
93 *****
94 ***** Each transition state looks like this:
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 DESC_a      MEND
```

from previous  
state  
\\:

TRANSITION  
I/O errors? Y  
ERROR STATE  
(terminate)  
\\:

sim i/o  
function  
complete? N  
TEST FOR  
ABORT  
REQUEST  
no  
abort : to next  
state  
\\:

to state 6  
to state 7  
\\:

LOCATION OBJECT CODE LINE SOURCE LINE

```

123 DESC_2      MACR3
124 .GOTO      DESC_2
125 *
126 *****
127 *
128

```

COMMON ATTRIBUTE AREAS (COMMUNICATION BUFFER WITH HP)  
ALL VALUES ARE HEXIDECIMAL

```

1) ASSIGN FILENAME TO CA
   CA      CA+1      CA+2      CA+n
   -----
   : 8A :LENGTH :FILENAME -----> : USER ID : NOT
   :    :BYTE   :UP TO 9 BYTES : UP TO 5 BYTES : USED----->
   -----
   LENGTH_BYTE := (((LENGTH OF FILE NAME)+1)/2)-1)*32) + ((LENGTH OF USER ID)/(2)*8)
   FILENAME MUST BE ODD NUMBER OF BYTES LONG MAY BE PADDED WITH ONE SPACE.
   USER ID MUST BE EVEN NUMBER OF BYTES LONG MAY BE PADDED WITH ONE SPACE.

```

```

2) OPEN
   CA      CA+1      CA+2
   -----
   : 81 : 04 : 00 : NOT
   :    :    :    : USED----->
   -----

```

CA+1 MUST BE 04HEX DESIGNATES ABSOLUTE FILE WILL BE OPEVED.  
CA+2 MUST BE 00HEX DESIGNATES DISK NUMBER.

```

3) CLOSE
   CA
   -----
   : 82 : NOT
   :    : USED----->
   -----

```

```

4) READ
   CA      CA+1      CA+2      CA+4      CA+6
   -----
   : 87 : 80 : #BYTES TO LOAD : N/A : N/A : LOAD ADDRESS
   :    :    : MSB : LSB :    : MSB : LSB
   -----

```

CA+1 DEFINES THE BUFFER LENGTH IN WORDS - 1 MUST BE LOADED BEFORE CALLING SIM I/O  
CA+2 DEFINES NUMBER OF BYTES TO MOVE TO RAM FROM THE SIM I/O BUFFER  
CA+6 DEFINES THE RAM ADDRESS TO LOAD TO  
CA+8 (NOT SHOWN) IS THE START OF THE SIM I/O BUFFER  
NOTE: CA+2 AND CA+4 ARE ONE WORD LONG BUT NOT STORED AS 280 WORDS.  
THE 280 EXPECTS WORD VALUES TO BE STORED LSB/MSB.

```

171 DESC_2      MEND
172
173
174

```



LOCATION OBJECT CODE LINE SOURCE LINE

```

176 NEXT_STATE MACRO &P1
177 LD A,&P1
178 LD C,TAPE_STATE,&A
179 LD HL,CSTATE_VECTORS+&P1+&P1J
180 LD C,EXT_STATE_ADDRESS,&HL
181 JP END_OF_STATE_MACHINE
182 DESC_B MEND
183 :SUBROUTINES CALLED:
184 : EXT
185
186 :OPERATING SYSTEM CALLS:
187
188 :EXTERNAL DATA AREAS USED:
189 EXT KILL_TAPE
190 EXT WRITE_TAPE
191 EXT CSA
192 EXT TAPE_STATE
193 EXT OVERLAY_NUMBER
194
195 :GLOBAL DATA AREAS DEFINED:
196 GLB DEVICE_ID
197
198 :LOCAL EQUATES
199 CA EQU 07400H
200 BUF_LEN EQU CA+1
201 FILETYPE EQU CA+1
202 DISC_NUM EQU FILETYPE+1
203 REC_LEN EQU CA+2
204 BUFFER EQU CA+8
205 RNAM_BUF EQU CA+1
206 LOADADDR EQU CA+4
207
208 MAXBUFLN EQU 128
209 ABSOLUTE EQU 004H
210 OPEN EQU 081H
211 CLOSE EQU 082H
212 READ EQU 087H
213 WRITE EQU 089H
214 RENAME EQU 08AH
215
216 :GLOBAL EQUATES
217 : INCLUDE equate file name
218
:COMMON ATTRIBUTES AREA FOR SIM I/O
:
:FILETYPE ADDRESS IN C_A_
:DISK NUMBER ADDRESS IN C_A_
:RECORD LENGTH ADDRESS IN C_A_
:I-O BUFFER
:RENAME BUFFER AREA
:ADDRESS OF MEM ADDRESS OF OVERLAY
:
:UP TO 128 WORDS MAY BE READ IN
:FILE TYPE IS ABSOLUTE

```

:SOMETHING IN THE COMMAND BUFFER!

LOCATION OBJECT CODE LINE SOURCE LINE

```

0000      220      PROG
          221      GLB      TAPE_MANAGER
          222
          223      GLB      INITIALIZE_TAPE
          224      INITIALIZE_TAPE:
          225      GLB      INIT_TAPE
          226      INIT_TAPE:
          227      LD      A,0
          228      LD      [CA],A
          229      LD      [CSA],A
          230      DEC     A
          231      LD      [COVERLAY_NUMBER],A
          232      NEXT_STATE 0
          233      LD      A,0
          234      LD      [TAPE_STATE],A
          235      LD      HL,[STATE_VECTORS+0+0]
          236      LD      [NEXT_STATE_ADDRESS],HL
          237      LD      [END_OF_STATE_MACHINE]
          238      JP      RET
          239
          240      TAPE_MANAGER:
          241      BEGIN
          242
          243      LD      A,[CA]
          244      OR      A
          245      JP      M,[END_OF_STATE_MACHINE]
          246      JP      Z,[AB_REQ]
          247      CP      1
          248      JP      NZ,ERROR
          249
          250      AT THIS POINT ANY SIM I/O FUNCTIONS ARE COMPLETE: TEST FOR ABORTS (KILLS)
          251
          252      GLB      AB_REQ
          253      LD      A,[CSA]
          254      CP      KILL_TAPE
          255      JR      NZ,CASE_STATE
          256      LD      A,[TAPE_STATE]
          257
          258      CP      3
          259      JP      C,STATE_PRE_IDLE
          260      CP      6
          261      JP      C,STATE_CLOSE
          262
          263      FALL THRU TO CASE STATEMENT
          264
          265      CASE
          266      TAPE_STATE,(IDLE,RENAME,OPEN,READ1,READ2,MOVE2VRAM,CLOSE,PRE_IDLE)
          267
          268      CASE_STATE:
          269
          270      LD      HL,[NEXT_STATE_ADDRESS]
          271      JP      [HL]

```

:CLEAR THE HP64000 AREA

:AND THE TAPE CONTROL STATUS AREA

:MAKE THE OVERLAY NUMBER -1 (INVALID)

:SET THE IDLE STATE!

:SOMETHING IN THE COMMAND BUFFER!

:TEST THE STATUS OF THE FILE

:IF THERE IS A COMMAND IN THE BUFFER

:END OF FILE ON READ FROM HP64000

:IF COMMAND IS TO KILL TAPE COMMAND

:CHECK THE STATE OF THE TAPE

:STATE 0,1,2

:FILE NOT OPENED

:STATE 3,4,5

:FILE OPENED TRY TO CLOSE IT

:IF STATE = 6,7

:FILE IS TRYING TO CLOSE

LOCATION OBJECT CODE LINE SOURCE LINE

```

0043 0068      DEFN STATE_RENAME      : 1
0045 00A2      DEFN STATE_OPEN      : 2
0047 00C2      DEFN STATE_READ_1    : 3
0049 00DA      DEFN STATE_READ_2    : 4
004B 00E2      DEFN STATE_MOVE2RAM  : 5
004D 012A      DEFN STATE_CLOSE     : 6
004F 0130      DEFN STATE_PRE_IDLE  : 7
0051 0154      DEFN WRITE_1

272 *          IF THE MACHINE IS IDLE IT'S OK TO TEST FOR ANOTHER READ REQUEST
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 STATE_IDLE:
284 LD
285 OR
286 JP
287 *
288 *
289 *
290 *
291 *
292 STATE_RENAME:
293 LD
294 LD
295 LD
296 LD
297 LD
298 *
299 *
300 *
301 LD
302 OV_2_ASCII_1:
303 OR
304 SBC
305 INC
306 JR
307 DEC
308 ADD
309 LD
310 ADD
311 LD
312 LD
313 CP
314 JR
315 LD
316 OV_2_ASCII_2:
317 LD
318 INC
319 LD
320 LD
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```



## LOCATION OBJECT CODE LINE SOURCE LINE

```
008F 3E8A      324 LD      A,RENAME
0091 327400    325 LD      [CA],A
0094          326 *
0094          327 NEXT_STATE 2
0094 3E02      LD      A,2
0096 320000    LD      [TAPE_STATE],A
0099 2A0045    LD      HL,[STATE_VECTORS+2+2J]
009C 220000    LD      [NEXT_STATE_ADDRESS],HL
009F C3016A    JP      END_OF_STATE_MACHINE

00A2          328 *
00A2          329 *
00A2          330 STATE_OPEN:
00A2          331 *
00A2          332 *
00A2          333 *
00A2          334 OPEN THE FILE
00A2 3E04      LD      A,ABSOLUTE
00A4 327401    LD      [FILETYPE],A
00A7 3E00      LD      A,0
00A9 327402    LD      [DISC_NUM],A
00AC 3E81      LD      A,OPEN
00AE 327400    LD      [CA],A
00B1 3A0000    LD      A,[CCSA]
00B4          NEXT_STATE 3
00B4 3E03      LD      A,3
00B6 320000    LD      [TAPE_STATE],A
00B9 2A0047    LD      HL,[STATE_VECTORS+3+3J]
00BC 220000    LD      [NEXT_STATE_ADDRESS],HL
00BF C3016A    JP      END_OF_STATE_MACHINE
00C2          342 STATE_READ_1:
00C2          343 *
00C2          344 *
00C2          345 *
00C2 3E80      LD      A,MAXBUFLN
00C4 327401    LD      [BUF_LEN],A
00C7 3E87      LD      A,READ
00C9 327400    LD      [CA],A
00CC          NEXT_STATE 4
00CC 3E04      LD      A,4
00CE 320000    LD      [TAPE_STATE],A
00D1 2A0049    LD      HL,[STATE_VECTORS+4+4J]
00D4 220000    LD      [NEXT_STATE_ADDRESS],HL
00D7 C3016A    JP      END_OF_STATE_MACHINE

00DA          352 *
00DA          353 *
00DA          354 *
00DA          355 *
00DA          356 STATE_READ_2:
00DA 3E80      LD      A,MAXBUFLN
00DC 327401    LD      [BUF_LEN],A
00DF 3E87      LD      A,READ
00E1 327400    LD      [CA],A
00E4          NEXT_STATE 5
00E4 3E05      LD      A,5
00E6 320000    LD      [TAPE_STATE],A
00E9 2A0043    LD      HL,[STATE_VECTORS+5+5J]

:NEXT OPEN
:SOMETHING IN THE COMMAND BUFFER!

:STATE 2

:LOAD THE FILE TYPE
:DISK NUMBER
:OPNE THE CURRENT FILE
:NEXT STATE READ OR WRITE?
:NEXT STATE = READ_1
:SOMETHING IN THE COMMAND BUFFER!

:STATE 3

:SET THE INPUT BUFFER LENGTH
:
:READ THE FIRST RECORD

:SOMETHING IN THE COMMAND BUFFER!

:STATE 4
:SET THE INPUT BUFFER LENGTH
:
:
:NEXT STATE = READ A DATA RECORD
:SOMETHING IN THE COMMAND BUFFER!
```

LOCATION OBJECT CODE LINE SOURCE LINE

```

00EC 220000      *      LD      [NEXT_STATE_ADDRESS],HL
00EF C3016A      *      JP      END_OF_STATE_MACHINE
363
364 *
365 *
366 *      MOVE THE LAST BUFFER TO RAM
00F2
00F2 3A7400      LD      A,[CA]
00F5 B7          DR      A
00F6 280E        JR      Z,NO_EOF
371 *
372 *
373
374
375
376
377
378
379 *
380
381
382
383
384
385
386
387
388
389
390 *
391
392
393 *
394 *
395 *
396 STATE_CLOSE:
397      LD      A,CLOSE
398      LD      [CA],A
399
400
401 *
402 *
NEXT_STATE 6
LD      A,6
LD      [TAPE_STATE],A
LD      HL,[STATE_VECTORS+6*6]
LD      [NEXT_STATE_ADDRESS],HL
JP      END_OF_STATE_MACHINE
375 NO_EOF:
LD      HL,[REC_LEN]
LD      C,H
LD      B,L
LD      HL,[LOADADDR]
LD      D,L
LD      E,H
LD      HL,BUFFER
LD      A,[CCSA]
CP      WRITE_TAPE
JR      Z,WRITE_1
LDIR
NEXT_STATE 4
LD      A,4
LD      [TAPE_STATE],A
LD      HL,[STATE_VECTORS+4*4]
LD      [NEXT_STATE_ADDRESS],HL
JP      END_OF_STATE_MACHINE
393 *
394 *
395 *
396 STATE_CLOSE:
397      LD      A,CLOSE
398      LD      [CA],A
399
400
401 *
402 *
NEXT_STATE 7
LD      A,7
LD      [TAPE_STATE],A
LD      HL,[STATE_VECTORS+7*7]
LD      [NEXT_STATE_ADDRESS],HL
JP      END_OF_STATE_MACHINE
401 *
402 *
:STATE 5
:CHECK THE RETURN STATUS
:END OF FILE?
:NEXT STATE = CLOSE FILE
:SOMETHING IN THE COMMAND BUFFER!
:MOVE THE CURRENT OVERLAY TO MEMORY
:GET THE NUMBER OF BYTES TO MOVE TO RAM
:GET THE LOAD ADDRESS AND CHECK TO SEE IF IT IS IN VRAM
:FROM ADDRESS
:NEXT STATE = READ A DATA RECORD
:SOMETHING IN THE COMMAND BUFFER!
:STATE 6
:CLOSE THE FILES
:NEXT STATE = PRE-IDLE
:SOMETHING IN THE COMMAND BUFFER!

```

LOCATION OBJECT CODE LINE

```

0130      NEXT STATE IS IDLE
0130      STATE_PRE_IDLE:
0130      LD A,0
0130      LD [CSAJ],A
0130      DEC A
0130      LD COVERLAY_NUMBERJ,A
0130      NEXT_STATE 0
0130      LD A,0
0130      LD [TAPE_STATEJ],A
0130      HL,[CSTATE_VECTORS+0*0J]
0130      [NEXT_STATE_ADDRESSJ],HL
0130      END_OF_STATE_MACHINE
0130      JP

0130      WRITE OUT 250 DATA BYTES TO AN OVERLAY! GH 83YI
0130      :GET HERE FROM MOVE2RAM!
0130      :SOMETHING IN THE COMMAND BUFFER!

0130      NEXT_STATE 6
0130      LD A,6
0130      LD [CTAPE_STATEJ],A
0130      HL,[CSTATE_VECTORS+6*6J]
0130      [NEXT_STATE_ADDRESSJ],HL
0130      END_OF_STATE_MACHINE
0130      JP

0130      (Ordinarily registers are restored; retain only the pushes and pops you need.)
0130      :SAVE THE OVERLAY NUMBER IN A
0130      :B HAS THE FUNCTION CODE
0130      :C HAS THE ERROR CODE

0130      LD [HLJ] HAS JUST CAUSED A SINK TO OCCUR ON THE HP64000
0130      THIS IS TO LET THE USER KNOW THERE HAS BEEN AN ERROR

0130      XOR A
0130      LD [CAJ],A
0130      INC A
0130      LD [CSAJ],A
0130      LD HL,[CSTATE_VECTORS+0000J]
0130      [NEXT_STATE_ADDRESSJ],HL

```

LOCATION OBJECT CODE LINE

SOURCE LINE

0186 C9

RET

450 :  
451 :  
452 : END {TAPE\_ERR}  
453 :  
454 :  
455 :  
456 :  
457 :  
458 :  
459 :  
460 :  
461 :  
462 :  
463 :  
464 :  
465 :  
466 :  
467 :  
468 :  
469 :  
470 :  
471 :  
472 :  
473 :  
474 :  
475 :  
476 :  
477 :  
478 :  
479 :  
480 :  
481 :  
482 :  
483 :

OVERLAY NAMES

:LENGTH :FILENAME :-----> : USER ID :  
:BYTE :UP TO 9 BYTES : UP TO 5 BYTES :LENGTH\_BYTE := (((LENGTH OF FILE NAME)+1)/2)-1)\*32) + ((LENGTH OF USER ID)/2)\*8)  
FILENAME MUST BE ODD NUMBER OF BYTES LONG MAY BE PADDED WITH ONE SPACE.  
USER ID MUST BE EVEN NUMBER OF BYTES LONG MAY BE PADDED WITH ONE SPACE.

OVERLAY NAME 0

0187

0187 40

0188 4F564C5F31

DVLAY0 DEF8

NAM\_0 ASCII "DVL\_1"

USID0 ASCII ""

FL\_NM\_LN0 EQU ((USID0-NAM\_0+1)/2)-1

US\_ID\_LN0 EQU (\$-USID0)/2

SAMPLE\_NAME\_LEN EQU \$-SAMPLE\_NAME-2

DATA

GLB NEXT\_STATE\_ADDRESS

NEXT\_STATE\_ADDRESS DEFS 2

DEVICE\_ID DEFS 1

0000

0002

Errors=

0

:LENGTH DESCRIPTION BYTE  
:MUST BE ODD NUMBER OF LETTERS  
:MUST BE EVEN NUMBER OF LETTERS  
:LENGTH OF FILENAME IN WORDS  
:LENGTH OF USER ID IN WORDS  
:NUMBER OF BYTES TO MOVE = 2:POINTER TO NEXT ENTRY STATE  
:DEVICE INDEPENDENCE SIMULATED



LINE#	SYMBOL	TYPE	REFERENCES
209	ABSOLUTE	A	334
248	AB_REQ	P	241,247
204	BUFFER	A	383
200	BUF_LEN	A	347,358
193	CA	A	200,201,203,204,205,206,228,238,325,339,349,360,368,398,418,445
265	CASE_STATE	P	250
211	CLOSE	A	397
191	CSA	E	229,248,284,340,385,407,447
483	DEVICE_ID	D	196
202	DISC_NUM	A	337
426	END_OF_STATE_MA	P	181,240,286
431	ERRR	P	243
201	FILETYPE	A	202,335
476	FL_NM_LNO	A	473
224	INITIALIZE_TAPE	P	223
226	INIT_TAPE	P	225
189	KILL_TAPE	E	249
206	LOADADDR	A	380
208	MAXBUFLN	A	346,357
474	NAM_0	P	476
482	NEXT_STATE_ADDR	D	180,268,449,481
375	NO_EOF	P	370
210	OPEN	A	338
193	OVERLAY_NUMBER	E	231,293,409,435
473	OVLAY0	P	
302	OV_2_ASCII_1	P	306
316	OV_2_ASCII_2	P	314
212	READ	A	348,359
203	REC_LEN	A	376
214	RENAME	A	324
205	RNAM_BUF	A	295
472	SAMPLE_NAME	P	294,478
478	SAMPLE_NAME_LEN	A	296
396	STATE_CLOSE	P	257,277
283	STATE_IDLE	P	271
367	STATE_MOVE2RAM	P	276
330	STATE_OPEN	P	273
405	STATE_PRE_IDLE	P	254,278
342	STATE_READ_1	P	274
356	STATE_READ_2	P	275
292	STATE_RENAME	P	272
270	STATE_VECTORS	P	179,448
235	TAPE_MANAGER	P	221
192	TAPE_STATE	E	178,251,437
475	USIO0	P	476,477
477	US_ID_LNO	A	473
213	WRITE	A	417
414	WRITE_1	P	279,387
190	WRITE_TAPE	E	386

## 1.9

TAPE INTERFACE

Tue, 15 May 1984, 20:30 PAGE 1

FILE: TAPE\_INTE:TOS

LOCATION OBJECT CODE LINE SOURCE LINE  
HEWLETT-PACKARD: TAPE\_INTERFACE (c) Coleco 1983 Confidential

```
1 ^Z80^
3 NAME ^Rev 02 - GPB^
4
5 De_TAPE_INTERFACE MACRO
6     .GOTO Ede_TAPE_INTERFACE :Header Rev. 5
7
8 Project: TAPE C101
9
10 *****
11 *
12 * TAPE_INTERFACE DTT *
13 *
14 *****
15
16 Rev History
17 Rev. Date Name Change
18 2 11/2/83 GPB CHANGED RANGE TO RANGE_ (duplicate symbol problem)
19 1 9/13/83 DTT CHANGED TO ALLOW ERROR RETIRES
20 0 7/5/83 DTT Initial Pseudo code
21
22 Function:
23 REQUEST READS AND WRITES AS DEFINED IN DCB.
24 REQUESTS ABORT_TAPE.
25 TEST STATUS OF TAPE REQUEST.
26
27
28 Ede_TAPE_INTERFACE MEND
29 Pseudo_code_TAPE_INTERFACE MACRO :Pseudocode macro area
30 BEGIN;
31 STORE OVERLAY_NUMBER
32 HL := POINTER TO DCB := OVERLAY_NUMBER*5 + OVERLAY_TABLE_POINTER
33 MOVE DCB TO CSA
34 IF WRITE THEN
35     SEND WRITE_COMMAND
36 ELSE
37     SEND READ_COMMAND
38 ENDIF
39 END
40
41 .GOTO Ede_TAPE_INTERFACE
42
43
44 Ede_TAPE_INTERFACE MEND
```

LOCATION OBJECT CODE LINE SOURCE LINE

```

46 :Subroutines called
47 : EXT
48
49 :Subroutines defined
50 GLB TEST_TAPE
51 GLB ABORT_TAPE
52 GLB LOAD_OVERLAY
53 GLB WRITE_OVERLAY
54 GLB CALC_OCB_ADDR
55 GLB WRITE_BLOCKS
56 GLB LOAD_BLOCKS
57 GLB BLOCK_IO
58 : GLB
59
60 :Operating system calls
61 : EXT
62
63 :Inputs/Outputs passed in registers
64 : A = OVERLAY NUMBER 1 thru N
65 : A <> 0 = ERROR
66
67 :External data areas used
68 EXT OCB_PTR
69
70 :Global data areas defined
71 GLB OVERLAY_NUMBER
72 GLB TAPE_STATE
73 GLB CSA
74 GLB XFER_ADDR
75 GLB BLOCK_NUM
76 GLB RANGE_
77 : GLB
78
79 :Local equates
80 : EQU
81
82 :Global equates
83 GLB READ_TAPE
84 GLB WRITE_TAPE
85 GLB KILL_TAPE
86 READ_TAPE EQU 81H
87 WRITE_TAPE EQU 82H
88 KILL_TAPE EQU 87H
89 : INCLUDE File_name:userid
90

```

<0081>  
<0082>  
<0087>

:POINTER TO THE OVERLAY CONTROL TABLE

LOCATION OBJECT CODE LINE SOURCE LINE

```
33
34 ;Inputs/Outputs passed in registers
35 ; CALLED EVERY 50th SECOND BY THE NMI
36
37
38 TAPE1 EQU 08H
39 READ_TAPE EQU 81H
40 WRITE_TAPE EQU 82H
41 KILL_TAPE EQU 87H
42 INCLUDE P_DCB_EQU:EOS
+ ;THESE OUR EQUATES THAT ARE USED BY THE EOS PROGRAMS TO REFERENCE
+ ;PCB AND DCB INFORMATION
+
+
+ ;PCB EQUATES
+
+ P_COM_STAT EQU 0 ; THIS IS THE COMMAND/STATUS BYTE
+
+ P_REL_ADDR EQU 1 ; THIS IS THE RELOCATION ADDRESS
+ P_REL_ADDR_LO EQU P_REL_ADDR+0
+ P_REL_ADDR_HI EQU P_REL_ADDR+1
+
+ P_NUM_DCBS EQU 3 ; THIS IS THE NUMBER OF DCBS DEFINED
+
+
+ P_SIZE EQU 4 ; THE NUMBER OF BYTES IN THE PCB
+
+
+ ;DCB EQUATES
+
+ D_COM_STAT EQU 0 ; THE COMMAND STATUS BYTE
+
+ D_BUF_ADR EQU 1 ; ADDRESS OF THE DATA BUFFER
+ D_BUF_ADR_LO EQU D_BUF_ADR+0
+ D_BUF_ADR_HI EQU D_BUF_ADR+1
+
+ D_BUF_LEN EQU 3 ; THE LENGTH OF THE DATA BUFFER
+ D_BUF_LEN_LO EQU D_BUF_LEN+0
+ D_BUF_LEN_HI EQU D_BUF_LEN+1
+
+ D_SECT_NUM EQU 5 ; THE BLOCK DEVICE SECTOR NUMBER
+
+ D_SEC_DEV_ID EQU 9 ; SECONDARY DEVICE ID
+
+ D_RET_COUNT EQU 14 ; THE NUMBER OF TIMES A COMMAND WILL
+ ; BE RETRIED.
+ D_RET_COUNT_LO EQU D_RET_COUNT+0
+ D_RET_COUNT_HI EQU D_RET_COUNT+1
+
+ D_DEV_ADDR EQU 16 ; THE DEVICE ADDRESS (ID)
+
+ D_MAX_MSG_LEN EQU 17 ; THE MAX LENGTH OF A DATA STRING
+
+ D_MAX_MSG_LEN_LO EQU D_MAX_MSG_LEN+0
+ D_MAX_MSG_LEN_HI EQU D_MAX_MSG_LEN+1
+
<0008>
<0091>
<0082>
<0097>

<0000>
<0001>
<0001>
<0002>
<0003>
<0004>

<0000>
<0001>
<0001>
<0002>
<0003>
<0003>
<0003>
<0004>
<0005>
<0009>
<000E>
<000E>
<000F>
<0010>
<0011>
<0011>
<0012>
```



LOCATION OBJECT CODE LINE SOURCE LINE

```

92 PRJG
93
94 : TEST FOR COMPLETION OF IO REQUEST
95 TEST_TAPE
96 LD A,[CSA]
97 OR A
98 JP EXIT_TAPE
99 #
100 #
101 #
102 ABORT_TAPE
103 LD
104 LD
105 JP
106 #
107 #
108
109 CALC_OCB_ADDR:
110 DEC A
111 C,A
112 LD B,0
113 LD HL,[OCB_PTR]
114 ADD HL,BC
115 ADD HL,BC
116 ADD HL,BC
117 ADD HL,BC
118 ADD HL,BC
119 RET
120 #
121 WRITE_OVERLAY:
122 SCF
123 JR LD_1
124 #
125 #
126 LOAD_OVERLAY:
127 OR A
128 #
129 #
130 LD_1:
131 : BEGIN
132 #
133 #
134 #
135 LD OVERLAY_NUMBER],A
136 #
137 #
138 PUSH
139 CALL
140 POP
141 #
142 JR NC,LOAD_BLOCKS
143
144 WRITE_BLOCKS:
145 LD
146 JR
147 LOAD_BLOCKS:
148 LD
149
0000
0000 3A0002
0003 B7
0004 C3003D

0007 3E87
0009 320002
000C C3003D

000F 3D
000F 4F
0010 0600
0013 2A0000
0016 09
0017 09
0018 09
0019 09
001A 09
001B C9

001C
001C 37
001D 1801

001F
001F B7

0020

0020 320000

0023 F5
0024 C0000F
0027 F1

0028 3004

002A 3E82
002C 1802
002E
002E 3E81

```

:ADDR = OCB\_TABLE(OV\_NUM-1\*5)  
:GET THE OVERLAY NUMBER IN C

:POINTER TO THE OCB TABLE  
:OCB\_ADDR = OVERLAY\_NUM\*5 + START\_OF\_TABLE

:SET CARRY FLAG IF WRITE INSTRUCTION!

:RESET CARRY FLAG IF READ

(Ordinarily registers are restored; retain only the pushes and pops you need.)

:FOR DEBUGGING PURPOSES

:SAVE THE CARRY FLAG IF SET

:WRITE INSTRUCTIONS GO HERE

:READ INSTRUCTIONS GO HERE!

LOCATION OBJECT CODE LINE SOURCE LINE

```
0030
0030 110003 149 BLOCK_ID:
0033 010005 150 LD
0036 ED80 151 LD
0038 110002 152 LDIR
003B 12 153 LD
003C AF 154 LD
155 XOR
156 GLB
157 EXIT_TAPE:
003D 003D C9 158 RET
159 *****
160 DATA
161 OVERLAY_NUMBER DEFS 1
162 TAPE_STATE DEFS 1
163 CSA DEFS 6
164 XFER_ADDR EQU CSA+1
165 BLOCK_NUM EQU XFER_ADDR+2
166 RANGE_ EQU BLOCK_NUM+2
<0003>
<0005>
<0007>
*****
:
:
: POINT TO THE COMMAND STATUS AREA
: NUMBER OF BYTES TO MOVE
:
: SET THE COMMAND IN THE CSA BUFFER
*****
```

Errors= 0

LINE#	SYMBOL	TYPE	REFERENCES
102	ABORT_TAPE	P	51
149	BLOCK_IO	P	57,146
165	BLOCK_NUM	D	75,166
109	CALC_DCB_ADDR	P	54,139
163	CSA	D	73,96,104,150,153,164
157	EXIT_TAPE	P	98,105,156
88	KILL_TAPE	A	85,103
147	LOAD_BLOCKS	P	56,142
126	LOAD_OVERLAY	P	52
130	LD_1	P	123
68	DCB_PTR	E	113
161	OVERLAY_NUMBER	D	71,135
166	RANGE	D	76
86	READ_TAPE	A	83,148
162	TAPE_STATE	D	72
95	TEST_TAPE	P	50
**	WRITE	U	34
144	WRITE_BLOCKS	P	55
121	WRITE_OVERLAY	P	53
87	WRITE_TAPE	A	84,145
164	XFER_ADDR	D	74,165

1.10

DDP MANAGER

PAGE 1

Tue, 15 May 1984, 20:29

FILE: DDP\_MANAG:IOS      HEWLETT-PACKARD: DDP\_MANAGER (c) Coleco 1983 Confidential

LOCATION OBJECT CODE LINE      SOURCE LINE

```
1 ^280^
2 NAME ^Rev 01 - DTT^
3
4
5 De_DDP_MANAGER MACRO
6     .GOTO Ede_DDP_MANAGER
7
8 Project: H132, VS
9
10 *****
11 *
12 # DDP_MANAGER DTT *
13 *
14 *****
15
16 Rev History
17 Rev. Date      Name      Change
18 1
19 0 9/9/83 DTT      DEVICE_ID --> DEV_ID
20 Initial Pseudo code
21
22 Function:
23
24     CONTROLS THE DIGITAL DATA PACK FOR READS AND WRITES SETUP BY TAPE_INTERFACE
25
26 Ede_DDP_MANAGER MEND
27 Pseudo_code_DDP_MANAGER MACRO :Pseudocode macro area
28     .GOTO Ep_DDP_MANAGER
29
30
31 Ep_DDP_MANAGER MEND
```



LOCATION OBJECT CODE LINE SOURCE LINE

```

<0013> * D_DEV_TYPE EQU 19 ; THE DEVICE TYPE, BLOCKED OR CHARACTER
<0014> * D_STATUS_FLAGS EQU 20 ; DEVICE DEPENDENT STATUS FLAGS
<0015> * D_SIZE EQU 21 ; THE NUMBER OF BYTES IN THE DCB
*
* :DEVICE ID'S FOR THE KEYBOARD, PRINTER, AND TAPE DRIVE
*
<0001> * KEYBOARD_ID EQU 1 ; KYBD ID
<0002> * PRINTER_ID EQU 2 ; PRINTER ID
<0008> * TAPE_ID EQU 8 ; TAPE DRIVE ID
<0002> * ERROR_RETRY EQU 2 ; MAX RETRY'S ON ERRORS, READ_BLOCK AND WRITE_BLOCK
<000F> * MAX_DEV_ADDR EQU 15 ; HIGHEST POSSIBLE DEVICE ADDRESS
*
* :PCB COMMAND EQUATES
*
<0000> * PCB_IDLE EQU 0 ; THIS IS AN IDLE STATE
<0001> * PCB_SYNC1 EQU 1 ; SYNC BYTE 1
<0081> * PCB_SYNC1_ACK EQU PCB_SYNC1+80H
<0002> * PCB_SYNC2 EQU 2 ; SYNC BYTE 2
<0082> * PCB_SYNC2_ACK EQU PCB_SYNC2+80H
<0003> * PCB_SNA EQU 3 ; SET NEW PCB ADDRESS
<0083> * PCB_SNA_ACK EQU PCB_SNA+80H
<0004> * PCB_RESET EQU 4 ; RESET ALL NODES
<0084> * PCB_RESET_ACK EQU PCB_RESET+80H
<0005> * PCB_WAIT EQU 5 ;
<0085> * PCB_WAIT_ACK EQU PCB_WAIT+80H
*
* :DCB COMMAND EQUATES
*
<0000> * DCB_IDLE EQU 00 ;
<0001> * DCB_STATUS EQU 01 ; REQUEST STATUS
<0002> * DCB_RESET EQU 02 ; RESET NODE
<0003> * DCB_WR EQU 03 ; WRITE DATA TO DEVICE
<0004> * DCB_RD EQU 04 ; READ DATA FROM DEVICE
*
<FEC0> * INIT_PCB_ADDR EQU 0FEC0H ; INITIAL ADDRESS OF THE PCB

```

LOCATION OBJECT CODE LINE SOURCE LINE

```
* :GENERAL USAGE EQUATES FOR USE WITH DCB INFO
*
* CMND_COMPLETE_BIT EQU 7 : THIS IS THE BIT THAT INDICATES THE
* : COMMAND HAS BEEN PROCESSED.
* CMND_FIN_STATUS EQU 80H : THIS IS THE STATUS OF A COMMAND
* : THAT COMPLETED WITH NO ERRORS
* KBD_NAK EQU 8CH : INDICATES NO KEY READY
* PR_NAK EQU 86H : INDICATES THE PRINTER IS BUSY
* ETX EQU 03H : END OF DATA STRING INDICATOR
* TIMEOUT EQU 98H : DEVICE TIMED OUT
*
*3 INCLUDE EOS_ERRS:EOS
*
* DCB_NOT_FOUND EQU 1 : THERE WAS NO DCB FOR THE DEVICE REQUESTED.
* DCB_BUSY EQU 2 : DCB IS BUSY
* DCB_IDLE_ERR EQU 3 : DCB IS IDLE
*
* NO_DATE_ERR EQU 4
* NO_FILE_ERR EQU 5
* FILE_EXISTS_ERR EQU 6
* NO_FCB_ERR EQU 7
* MATCH_ERR EQU 8
* BAD_FNUM_ERR EQU 9
* EOF_ERR EQU 10
* TOO_BIG_ERR EQU 11
* FULL_DIR_ERR EQU 12
* FULL_TAPE_ERR EQU 13
* FILE_NM_ERR EQU 14
* RENAME_ERR EQU 15
* DELETE_ERR EQU 16
* RANGE_ERR EQU 17
*
* CANT_SYNC1 EQU 18
* CANT_SYNC2 EQU 19
* PRI_ERR EQU 20
*
* RQ_TP_STAT_ERR EQU 21
* DEVICE_DEPD_ERR EQU 22
*
*45 #EXTERNAL DATA AREAS USED:
*46
*47 EXT _START_RD_1_BLOCK
*48 EXT _FIND_DCB
*49 EXT _START_WR_1_BLOCK
*50 EXT _END_RD_1_BLOCK
*51
*52 NEXT_STATE MACRO EPI
*53 LD A,EPI
*54 LD CD_TAPE_STATES,A
*55 LD HL,[STATE_VECTORS+EPI+EPI]
*56 LD CNEXT_STATE_ADDRESS,HL
*57 JP
*58 MEND
*59
```

:SOMETHING IN THE COMMAND BUFFER!

:DLS(8/28/83)  
:DLS(8/30/83)  
:DLS(8/30/83)  
:DLS(8/31/83)



LOCATION OBJECT CODE LINE SOURCE LINE

```

117 *
118 STATE_1:
119 AB_REQ: LD A,CD_CSAJ
120 CP KILL_TAPE
121 JP Z,INIT_DDP
122 *
123 *
124 *
125
126 LD HL,CXFER_ADDRJ
127 LD DE,CBLOCK_NUMJ
128 LD BC,0000H
129 LD A,CD_CSAJ
130 CP READ_TAPE
131 LD A,CDEV_IDJ
132 JR NZ,EDS_TAPE_WRITE
133 EDS_TAPE_READ:
134 CALL _START_RD_1_BLOCK
135 JR RET_ADDR
136 EDS_TAPE_WRITE:
137 CALL _START_WR_1_BLOCK
138
139
140 RET_ADDR:
141
142
143 *
144 *
145 *
146 STATE_2:
147 LD A,CDEV_IDJ
148 CALL _END_RD_1_BLOCK
149 JP NC,_END_OF_STATE_MACHINE
150 *
151 *
152 *
153 *
154
155 JR NZ,STATE_1
156 *
157 *
158 *
159 *
160 STATE_2_OK:
161 *
162 LD A,CDEV_IDJ
163 CALL _FIND_DCB
164 LD C,IV+D_COM_STATJ,DCB_STATUS
165 NEXT_STATE_3
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```



LOCATION OBJECT CODE LINE SOURCE LINE

```
0075 2200D2      +      LD CNEXT_STATE_ADDRESSJ,HL
0078 C300CF      +      JP END_OF_STATE_MACHINE
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #

0078          STATE_3:
0079 3A00D1      LD A,CDEV_IDJ
007E C00000      CALL _FIND_DC3
0081 F0C9007E      BIT CMND_COMPLETE_BIT,CII+D_COM_STATJ
0085 CA00CF      JP Z,END_OF_STATE_MACHINE
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #

0088 FD7E00      LD A,CII+D_COM_STATJ
0088 FE80      CP 080H
008D 20D4      JR NZ,STATE_2_OK
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #

008F 3A00D1      LD A,CDEV_IDJ
0092 E6F0      AND OF0H
0094 F07E14      LD A,CII+D_STATUS_FLAGSJ
0097 2808      JR Z,DEV_0_CHECK
0099 C81F      RR A
009B C81F      RR A
009D C81F      RR A
009F C81F      RR A
00A1          DEV_0_CHECK:
00A1 E60F      AND OFH
00A3 C200C6      JP NZ,ERROR
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #

00A6 2A0003      LD HL,XFER_ADDRJ
00A9 ED5B0005      DE,CBLOCK_NUMJ
00AD 010400      LD BC,0400H
00B0 3A0007      LD A,C[RANGEJ]
00B3 30          DEC A
00B4 CA0000      JP Z,INIT_DDP
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #

00B7 320007      LD [RANGEJ],A
00BA 13          INC DE
00BB ED530005      LD CBLOCK_NUMJ,DE
00BF 09          ADD HL,BC
00C0 220003      LD [XFER_ADDRJ],HL
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #

00C3 C30026      JP STATE_1
214 #
215 #
216 #
217 #
218 #
219 #
220 #

00C6          ERROR PROCESSING FROM REQUEST STATUS
215 #
216 #
217 #
218 #
219 #
220 #

Errors occur after request status
Possible errors are 1=CRC check (bad data on tape)
```

LOCATION OBJECT CODE LINE SOURCE LINE

```
221 *
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *
265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *
```

Errors= 0

LINE#	SYMBOL	TYPE	REFERENCES
119	A9_REQ	P	
43	BAD_FNUM_ERR	A	
262	BLOCK_NUM	D	127,199,209,263
43	CANT_SYNC1	A	
43	CANT_SYNC2	A	
103	CASE_STATE	P	
42	CMND_COMPLETE_B	A	172
42	CMND_FIN_STATUS	A	
43	DCB_BUSY	A	
42	DCB_IDLE	A	
43	DCB_IDLE_ERR	A	
43	DCB_NOT_FOUND	A	
42	DCB_RD	A	
42	DCB_RESET	A	
42	DCB_STATUS	A	164
42	DCB_WR	A	
92	DOP_MANAGER	P	89
43	DELETE_ERR	A	
43	DEVICE_DEPD_ERR	A	
192	DEV_O_CHECK	P	187
241	DEV_ID	P	62,131,147,162,170,184
42	D_BUF_ADR	A	42,42
42	D_BUF_ADR_HI	A	
42	D_BUF_ADR_LO	A	
42	D_BUF_LEN	A	42,42
42	D_BUF_LEN_HI	A	
42	D_BUF_LEN_LO	A	
42	D_COM_STAT	A	164,172,177
260	D_CSA	A	70,112,119,129,226,249,261
42	D_DEV_ADDR	A	
42	D_DEV_TYPE	A	
42	D_MAX_MSG_LEN	A	42,42
42	D_MAX_MSG_LEN_HI	A	
42	D_MAX_MSG_LEN_LO	A	
258	D_OVERLAY_NUMBE	D	73,247
42	D_RET_COUNT	A	42,42
42	D_RET_COUNT_HI	A	
42	D_RET_COUNT_LO	A	
42	D_SECT_NUM	A	
42	D_SEC_DEV_ID	A	
42	D_SIZE	A	
42	D_STATUS_FLAGS	A	186
259	D_TAPE_STATE	D	54,71,248
233	END_OF_STATE_MA	P	57,114,149,173
43	EOF_ERR	A	
133	EOS_TAPE_READ	P	
136	EOS_TAPE_WRITE	P	132
218	ERROR	P	194
42	ERROR_RETRY	A	
42	ETX	A	
43	FILE_EXISTS_ERR	A	
43	FILE_NM_ERR	A	
43	FULL_DIR_ERR	A	
43	FULL_TAPE_ERR	A	
64	INITIALIZE_ODP	P	63
65	INITIALIZE_TAPE	P	63
68	INIT_CODE	P	82

LINE#	SYMBOL	TYPE	REFERENCES
67	INIT_DDP	P	63,121,203
42	INIT_PC3_ADDR	A	
66	INIT_TAPE	P	63
42	KBD_NAK	A	
42	KEYBOARD_ID	A	
41	KILL_TAPE	A	120
82	LEN_INIT	A	86
43	MATCH_ERR	A	
42	MAX_DEV_ADDR	A	
253	NEXT_STATE_ADDR	P	56,75,106,228
43	NO_DATE_ERR	A	
43	NO_FCB_ERR	A	
43	NO_FILE_ERR	A	
42	PCB_IDLE	A	
42	PCB_RESET	A	42
42	PCB_RESET_ACK	A	
42	PCB_SNA	A	42
42	PCB_SNA_ACK	A	
42	PCB_SYNC1	A	42
42	PCB_SYNC1_ACK	A	
42	PCB_SYNC2	A	42
42	PCB_SYNC2_ACK	A	
42	PCB_WAIT	A	42
42	PCB_WAIT_ACK	A	
42	PRINTER_ID	A	
43	PRT_ERR	A	
42	PR_NAK	A	
42	P_COM_STAT	A	
42	P_NUM_DCBs	A	
42	P_REL_ADDR	A	42,42
42	P_REL_ADDR_HI	A	
42	P_REL_ADDR_LO	A	
42	P_SIZE	A	
263	RANGE	D	201,207
43	RANGE_ERR	A	
39	READ_TAPE	A	130
43	RENAME_ERR	A	
140	RET_ADDR	P	135
43	RQ_TP_STAT_ERR	A	
118	STATE_1	P	79,155,213
146	STATE_2	P	80
160	STATE_2_OK	P	179
169	STATE_3	P	81
111	STATE_IDLE	P	78
77	STATE_VECTORS	P	55,74,227
38	TAPE1	A	241
42	TAPE_ID	A	
91	TAPE_MANAGER	P	90
42	TIMEOUT	A	
43	TOO_BIG_ERR	A	
40	WRITE_TAPE	A	
261	XFER_ADDR	D	126,198,211,262
50	-END_RO_1_BLOCK	E	148
48	-FIND_DCB	E	163,171
47	-START_RO_1_BLO	E	134
49	-START_WR_1_BLO	E	137



1.11

DDP INTERFACE

FILE: DDP\_INTER:TOS      HEWLETT-PACKARD: DDP\_INTERFACE      (c) Coleco 1984 Confidential      Mon, 21 May 1984, 16:35      PAGE 1

LOCATION OBJECT CODE LINE      SOURCE LINE

```
1 *280*
3 NAME ^Rev 01 - DTT^
4
5 De_DDP_INTERFACE MACRO
6     -GOTO Ed_DDP_INTERFACE :Header Rev. 5
7
8 Project: TAPE, C101
9
10 *****
11 *
12 * DDP_INTERFACE DTT *
13 *
14 *****
15
16 Rev History
17 Rev. Date Name Change
18 1 9/13/83 DTT CHANGED TO ALLOW ERROR RETIRES
19 0 7/5/83 DTT Initial Pseudo code
20
21 Function:
22 REQUEST READS AND WRITES AS DEFINED IN DCB.
23 REQUESTS ABORT_TAPE.
24 TEST STATUS OF TAPE REQUEST.
25
26
27 Ed_DDP_INTERFACE MEND
28 Pseudo_Code_DDP_INTERFACE MACRO :Pseudocode macro area
29 BEGIN:
30 STORE D-OVERLAY_NUMBER
31 HL := POINTER TO DCB := D-OVERLAY_NUMBER5 + OVERLAY_TABLE_POINTER
32 MOVE DCB TO D_CSA
33 IF WRITE THEN
34 SEND WRITE_COMMAND
35 ELSE
36 SEND READ_COMMAND
37 ENDIF
38 END
39
40 -GOTO Ed_DDP_INTERFACE
41
42
43 Ed_DDP_INTERFACE MEND
```

LOCATION OBJECT CODE LINE SOURCE LINE

```

45 :Subroutines called
46 : EXT
47 :
48 :Subroutines defined
49 GLJ TEST_TAPE
50 GLB ABORT_TAPE
51 GLB LOAD_OVERLAY
52 GLB WRITE_OVERLAY
53 GLB CALC_DCB_ADDR
54 GLB WRITE_BLOCKS
55 GLB LOAD_BLOCKS
56 GLB BLOCK_IO
57 :
58 :
59 :Operating system calls
60 : EXT
61 :
62 :Inputs/Outputs passed in registers
63 : A = JOVERLAY NUMBER 1 thru N
64 : A <> 0 = ERROR
65 :
66 :External data areas used
67 EXT DCB_PTR
68 :
69 :Global data areas defined
70 EXT D_OVERLAY_NUMBER
71 EXT D_TAPE_STATE
72 EXT D_CSA
73 : GL3
74 :
75 :Local equates
76 : EQU
77 :
78 :Global equates
79 GLB READ_TAPE
80 GLB WRITE_TAPE
81 GLB KILL_TAPE
82 READ_TAPE EQU 81H
83 WRITE_TAPE EQU 82H
84 KILL_TAPE EQU 87H
85 : INCLUDE File_name:userld
86

```

```

<0081>
<0082>
<0087>

```

: POINTER TO THE OVERLAY CONTROL TABLE

LOCATION	OBJECT	CODE	LINE	SOURCE	LINE
----------	--------	------	------	--------	------

```

83 PRGJG
89
90 ; TEST FOR COMPLETION OF IO REQUEST
91 TEST_TAPE
92 LD A,CD_CSAJ
93 OR A
94 JP EXIT_TAPE
95 #
96 #
97 #
98
99 ABORT_TAPE
100 LD
101 LD
102 JP
103 #
104
105 CALC_OCB_ADDR:
106 DEC A
107 LD C,A
108 LD B,0
109 LD HL,CCCB_PTRJ
110 ADD HL,BC
111 ADD HL,BC
112 ADD HL,BC
113 ADD HL,BC
114 ADD HL,BC
115 RET
116 #
117 WRITE_OVERLAY:
118 SCF
119 JR LO_1
120 #
121 #
122 LOAD_OVERLAY: A
123 OR
124 #
125 #
126 LO_1:
127 ; BEGIN
128 #
129 #
130 #
131 LD [D_OVERLAY_NUMBER],A
132 #
133 #
134 PUSH
135 CALL
136 PJP
137 #
138 JR NC,LOAD_BLOCKS
139
140 WRITE_BLOCKS:
141 LD
142 LD
143 LOAD_BLOCKS:
144 LD
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

0



LINE#	SYMBOL	TYPE	REFERENCES
98	ABORT_TAPE	P	50
145	BLOCK_ID	P	56,142
105	CALC_OCB_ADOR	P	53,135
72	D_CSA	E	92,100,146,149
70	D_OVERLAY_NUM3E	E	131
71	D_TAPE_STATE	E	
153	EXIT_TAPE	P	94,101,152
84	KILL_TAPE	A	81,99
143	LOAD_BLOCKS	P	55,138
122	LOAD_OVERLAY	P	51
126	LG_1	P	119
67	OCB_PIR	E	109
82	READ_TAPE	A	79,144
91	TEST_TAPE	P	43
**	WRITE	J	33
140	WRITE_BLOCKS	P	54
117	WRITE_OVERLAY	P	52
83	WRITE_TAPE	A	80,141