DECUS
PROGRAM LIBRARY

DECUS NO.	8-333
TITLE	8K PAL-D ASSEMBLER FOR 4K DISK MONITOR SYSTEM
AUTHOR	Charles H. Conley
COMPANY	Digital Equipment Corporation Maynard, Massachusetts
DATE	September 8, 1970
SOURCE LANGUAGE	PAL-D

8K PAL-D ASSEMBLER FOR 4K DISK MONITOR SYSTEM

DECUS Program Library Write-up DECUS No. 8-333

The 8K PAL-D Assembler differs from the 4K PAL-D Assembler as specified in the 1970 Programming Language Manual except as follows:

1) Input may be from any of the following:

*IN -	T:	Teletype reader
	R:	High-speed reader
	S: name	DF-32 disk or DECtape
	Sn: name	RF-08 disk
	Dn: name	DECtape

2) Binary output may be to:

*OUT -	T:	Teletype punch
	R:	High-speed punch
	S: name	DF-32 disk or DECtape
	Sn: name	RF-08 disk
	Dn: name	DECtape (On DECtape Systems only)

3) Listing output may be to:

*OPT -	T	Teletype
	R	High-speed punch
	L	LP-08 Line printer

The 8K PAL-D Assembler has the following pseudo-ops:

DECIMAL
* EJECT
* ENPUNCH
EXPUNGE
FIELD
FIXTAB
* IFDEF
* IFZERO
* NOPUNCH
OCTAL
PAGE
PAUSE
TEXT
XLIST
* ZBLOCK

*Pseudo-ops not found in 4K PAL-D but explained in the 1970 Programming Language Manual.

8K PAL-D provides space in the Symbol table for approximately 890 User defined symbols. It may be saved on the system device with the following command: .SAVE PAL8:0-5177, 6600-7577;200

8K PAL-D PROGRAMMING

The 8K PAL-D Assembler is similar to 4K PAL-D. The reader is advised to learn the 4K PAL-D Assembler by studying the appropriate sections of Chapter 13 of the Programming Language Manual, then return to this section to learn the additional features of 8K PAL-D. These additional features include assembler directives which permit operation of the Assembler to be controlled by the source program, page control, and the ability to expand to run in 12K of core.

CHARACTER SET

In addition to the characters allowed in 4K PAL-D, the following characters are given a special significance in 8K PAL-D: < >.

The angle brackets (< >) define the bounds of a conditional statement. The user should be especially cautious not to use angle brackets within a comment in any program containing a conditional assembly statement.

PSEUDO-OPERATORS

In addition to the pseudo-operators allowed in 4K PAL-D, the following pseudo-operators are unique to 8K PAL-D:

RESERVING FREE STORAGE

ZBLOCK n

Where n is an integer, ZBLOCK causes the Assembler to reserve n words of memory containing zeros, starting at the word indicated by the current location counter.

CONDITIONAL ASSEMBLY

IFDEF symbol <statements>

If the symbol indicated is previously defined, assemble the statements contained in the angle brackets. If undefined, ignore these statements. Any number of statements can be contained in the angle brackets and may consist of several lines of code. The format of the IFDEF statement requires a single space before and after the symbol.

IFZERO expression <statements>

If the evaluated (arithmetic or logical) expression is equal to zero, assemble the statements contained within the angle brackets; if the expression is non-zero, ignore these statements. Any number of statements can be contained in the angle brackets and may consist of several lines of code. The format of the IFZERO statement requires that the expression not contain any imbedded spaces and must have a single space preceding and following it.

BINARY OUTPUT CONTROL

NOPUNCH

Upon encountering this statement the Assembler continues to assemble the code, but ceases binary output.

ENPUNCH

This statement causes the Assembler to resume (or continue) binary output.

These two pseudo-operators are put into the source program and are ignored until pass 2 at which time they direct the Assembler to delete some section of code from the final binary punched tape.

For example, these pseudo-operators could be used where several programs have the same contents on page zero. When these programs are to be loaded and executed together, only one page zero need be punched.

PAGINATION OF OUTPUT LISTINGS

EJECT

The EJECT command causes the listing to jump to the top of the next page. The 8K PAL-D Assembler counts output lines and will format the user's program into neat, even pages with a page eject every 55 lines. If the user requires more frequent paging, he should use the EJECT pseudo-operator. A FORM FEED character within the source program will also cause a page eject.

The pagination process within the 8K PAL-D Assembler causes an output of carriage return/line feed pairs for the 33 ASR Teletype. For users with the 35 ASR Teletype who desire to output a FORM FEED character directly, changes should be made to modify the FORMI subroutine found in the 8K PAL-D listing.

LOADING AND OPERATING PROCEDURES

Saving 8K PAL-D

The 8K PAL-D Assembler is supplied on binary coded paper tape. It is loaded using the Binary Loader as explained in Appendix C2.

The 8K PAL-D Assembler may be saved on the system device as a system program. This is done by typing the following SAVE instruction:

```
.SAVE PAL8! Ø-5177,66ØØ-7577;2ØØ
```

The Assembler is now saved as a system program. The programmer may now type PAL8, which brings the assembler into core for use with symbolic source programs.

Output devices are the same for 8K PAL-D as for 4K PAL-D. When 8K PAL-D requests the input file(s), the user may select up to ten (10) input files. Valid input devices for 8K PAL-D are as follows:

Device Designator	Device
T:	Teletype
R:	High-speed reader/punch
S: name	DF 32 disk
Sn: name	RF 08 disk
DO: name through D7: name	DECtape

Symbol Table

The symbol table for 8K PAL-D provides space for approximately 890 (decimal) user defined symbols. When the SE (symbol table exceeded) error message occurs, assembly is terminated and control is returned to the Monitor. The user file .SYM is not used by 8K PAL-D.

12K VERSION OF 8K PAL-D

The 8K PAL-D Assembler must be reassembled to run in 12K of core. The 12K version has a larger symbol table, but assembles at a slower pace. The changes to be made are documented on page 1 of the 8K PAL-D listing.

ADDENDUM TO DECUS NO. 8-333

HARVARD UNIVERSITY

DEPARTMENT OF PHYSICS

LYMAN LABORATORY OF PHYSICS
CAMBRIDGE, MASSACHUSETTS 02138

September 18, 1974

Ms. Mary Hogan
Digital Equipment Corp. Users Society
Maynard, Massachusetts 01754

Dear Ms. Hogan:

We have run across a problem with 8K PAL-D (DECUS 8-333) on our PDP 8/E. You will perhaps want to include this note in the file for that program.

When the 8K PAL-D assembler is used with paper tape input from the PC04 high speed reader which strobes data by the feed hole signal, timing problems can occur which depend on the program being assembled. The problem is that while the reader "stop delay" one-shot (for single character reading rate) can hold off the reader flag for as long as 60 msec., 8K PAL-D waits only 36 msec. (on the PDP 8/E) before deciding that the tape has run out. For slower computers than the 8/E, this time will be longer and the problem may not occur.

The following patch will increase the loop time from 36 msec. to 82 msec.:

in location	4173	deposit	1173
"	1173	"	1400
"	1174	"	1400
"	1175	"	7200
"	1176	"	5777
"	1177	"	4141

Assembly time is not noticably increased by the change, since the reader flag can terminate the loop at any time. The change is completely compatible with standard (i.e. stepping motor) PC04 readers.

Thank you,

Larry A. Cohen
Research Assistant

LAC/cad

MAY 22 1975

ADDENDUM TO 8-333

Los Angeles Office
21120 Vanowen Street
Post Office Box 633
Canoga Park, CA 91305
Telephone (213) 347-8360

Bolt Beranek and Newman Inc.
Consulting Development Research

20 May 1975

Ms. Ferne Halley
Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754

Subject: Problem with PDP-8 8K PAL-D Assembler
(DECUS No. 8-333)

Reference: BBN Letter of 22 January 1975 (Attachment 1)

Dear Ms. Halley:

I would like to take this opportunity to thank you for sending us the source listing for the 8K PAL-D Assembler and for guiding our attention to specific parts of the code. After some review of the code, I believe we have found the cause of the problem regarding paper tape input files described in our 22 January letter.

I. STATEMENT OF THE PROBLEM

The problem is briefly defined as follows: Upon reaching the end of a paper tape input file, and after reading to the physical end of the tape, the assembler attempts to reprocess a portion of the last input buffer's worth of code. It thinks this code should be appended to the end of the file (often resulting in page overflow, double defined symbols, etc.

II. CAUSE OF THE PROBLEM

The cause of the problem centers around the buffered input scheme used for paper tape files. The assembler reads 128 (200_8) characters, stores them in a one page buffer, and sets up a working pointer (PTBUFP) and counter (CHRCNT) for retrieval of characters from the buffer. It then processes one "buffer's worth" of code, after which another 128 characters are read into the buffer and the process continues until the physical end of the tape is detected. Upon reaching the end of the tape, however, only part of the input buffer will be filled (since it is very unlikely that the last character on the paper tape corresponds exactly to the 128th character entered into the buffer). In this case

the working counter (CHRCNT) should be set to account for only a partial filling of the buffer. The problem is that CHRCNT is always set to indicate 128 characters are in the buffer, even under the end-of-tape condition.

III. FIX FOR THE PROBLEM

Attachment 2 shows a relatively straightforward fix for the problem. Unfortunately this fix requires reassembly of the 8K PAL-D code.

IV. PATCH FOR THE PROBLEM

For users with a current binary tape of 8K PAL-D, the four-instruction patch shown in Attachment 3 will accomplish the same result as the fix shown in Attachment 2. The patch is as follows:

LOCATION	OLD CONTENTS	NEW CONTENTS
4172	5357	7610
4174	N/A	1052
4175	N/A	7040
4176	N/A	5360

* * * * ** * * * *

We have used the patch outlined in section IV above and it has worked quite satisfactorily.

It might be well for a member of the DEC staff to review both the fix and the patch in case there may be some undesirable ramifications.

Very truly yours,

BOLT BERANEK AND NEWMAN INC.

Richard D. Horonjeff

R. D. Horonjeff

Enclosures

RDH:dlw

ATTACHMENT 1 Chron.

Los Angeles Office
21120 Vanowen Street
Post Office Box 633
Canoga Park, CA 91305
Telephone (213) 347-8360

Bolt Beranek and Newman Inc.
Architectural and Engineering Acoustics

22 January 1975

Ms. Ferne Halley
Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754

Subject: Problem with PDP-8 8K PAL-D Assembler
(DECUS No. 8-333)

Dear Ms. Halley:

In following up our telephone conversation today, I am outlining herein the problem we have been experiencing with the above referenced assembler.

I. BBN-LA PDP-8 Equipment Configuration

PDP-8 w/8K memory (4K core, 4K Calcomp/Galaxies semi-conductor)
DF-32 Disk w/32K storage
High Speed Paper Tape Reader/Punch
Teletype

miscellaneous non-DEC peripherals

II. Statement of the Problem

The problem is briefly defined as follows:

Upon reaching the end of a paper tape input file, and after reading to the physical end of the tape, the assembler attempts to reprocess a portion of the last input buffer's worth of code. It thinks this code should be appended to the end of the file (often resulting in page overflows, double defined symbols, etc.). An example is discussed in Section III below.

The problem occurs consistently with all paper tape files (unless the file is the last one containing the end-of-program ($) symbol). The same code read from a file on disk will be processed correctly. The problem occurs on all passes of the assembly.

The 4K PAL-D assembler will process paper tape files correctly (however, an early version of 4K PAL-D did have problems of a slightly different nature with paper tape input).

III. Example of the Problem

A short piece of code was used to exemplify the problem. The example consists of two input files. On paper tape they are two tapes. On disk they are ASCII files S:A1 and S:A2. A listing (using the disk editor) is shown in Attachment 1.

Attachment 2 shows the third pass listing from 8K PAL-D using the ASCII files on disk as source input. Note that the assembly proceeds correctly in this case.

Attachment 3 shows the third pass listing from 8K PAL-D using paper tape in the high-speed reader as source input. Note that after the last code on the first file (DUMMY2, 0) has been assembled, the assembler attempts to reprocess code which is already in its input buffer area.

I am enclosing two paper tapes containing the aforementioned ASCII files.

I hope the information I have included will aid you in locating the source of the problem. If I may be of any assistance please do not hesitate to write or call. Thank you for your consideration in this matter.

Very truly yours,

BOLT BERANEK AND NEWMAN INC.

Richard D. Horonjeff

RDH:ms

ATTACHMENT 2 "THE FIX"


```
04104  0000  PUTWD,  0
04105  7006          RTL
04106  7006          RTL
04107  3332          DCA PTRIN
04110  1332          TAD PTRIN
04111  0147          AND P7400
04112  1720          TAD I OBUFP
04113  3720          DCA I OBUFP
04114  5704          JMP I PUTWD

04115  1324  FILLED, TAD P10
04116  4462          JMS I ERR1
04117  5403          JMP I MONITOR
04120  6377  OBUFP,  OUTBUF-1
04121  1742  PAUSEP, WAITT+1
04122  4400  FNBLOK, FINDBL
04123  4462  SAMIOP, SAMIO
04124  0010  P10,    10
04125  0000  ENDCNT, 0
04126  0000  OEND,   0
04127  6600  P6600,  6600
04130  7775  THIRDW, 7775
04131  7600  OUTCNT, 7600

04132  0000  PTRIN,  0               /READ A CHARACTER FROM HIGH SPEED PAPER TA
04133  2052          ISZ CHRCNT      /HOW DOES THE BUFFER LOOK
04134  5365          JMP PTPKUP      /NOT EMPTY, GET CHARACTER
04135  1151          TAD P7600       /IS EMPTY, REINITIALIZE THE COUNTER
04136  3052          DCA CHRCNT
04137  1327          TAD P6600
04140  3053          DCA PTBUFP      /REINITIALIZE THE POINTER
04141  6011  PTREAD, RSF
04142  5346          JMP .+4
04143  3200          DCA PACK
04144  6016          RFC RRB
04145  5351          JMP .+4
04146  2200          ISZ PACK
04147  5773          JMP I MORTIM
04150  5370          JMP PTEMTY
04151  7450          SNA
04152  5341          JMP PTREAD
04153  3453          DCA I PTBUFP
04154  2053          ISZ PTBUFP
04155  2052          ISZ CHRCNT
04156  5341          JMP PTREAD
04157  7240  PTINIT, CLA ~~CMA~~
04160  1151          TAD P7600
04161  3052          DCA CHRCNT      /REINITIALIZE THE CHARACTER COUNTER
04162  1327          TAD P6600
04163  3053          DCA PTBUFP      /REINITIALIZE THE BUFFER POINTER
04164  5333          JMP PTRIN+1

04165  1453  PTPKUP, TAD I PTBUFP    /PICK UP A CHARACTER
04166  2053          ISZ PTBUFP
```

TAD CHRCNT
CMA

```
04104  0000  PUTWD,  0
04105  7006          RTL
04106  7006          RTL
04107  3332          DCA PTRIN
04110  1332          TAD PTRIN
04111  0147          AND P7400
04112  1720          TAD I OBUFP
04113  3720          DCA I OBUFP
04114  5704          JMP I PUTWD

04115  1324  FILLED, TAD P10
04116  4462          JMS I ERR1
04117  5403          JMP I MONITOR
04120  6377  OBUFP,  OUTBUF-1
04121  1742  PAUSEP, WAITT+1
04122  4400  FNBLOK, FINDBL
04123  4462  SAMIOP, SAMIO
04124  0010  P10,    10
04125  0000  ENDCNT, 0
04126  0000  OEND,   0
04127  6600  P6600,  6600
04130  7775  THIRDW, 7775
04131  7600  OUTCNT, 7600

04132  0000  PTRIN,  0                /READ A CHARACTER FROM HIGH SPEED PAPER TAP
04133  2052          ISZ CHRCNT       /HOW DOES THE BUFFER LOOK
04134  5365          JMP PTPKUP       /NOT EMPTY, GET CHARACTER
04135  1151          TAD P7600        /IS EMPTY, REINITIALIZE THE COUNTER
04136  3052          DCA CHRCNT
04137  1327          TAD P6600
04140  3053          DCA PTBUFP       /REINITIALIZE THE POINTER
04141  6011  PTREAD, RSF
04142  5346          JMP .+4
04143  3200          DCA PACK
04144  6016          RFC RRB
04145  5351          JMP .+4
04146  2200          ISZ PACK
04147  5773          JMP I MORTIM
04150  5370          JMP PTEMTY
04151  7450          SNA
04152  5341          JMP PTREAD
04153  3453          DCA I PTBUFP
04154  2053          ISZ PTBUFP
04155  2052          ISZ CHRCNT
04156  5341          JMP PTREAD
04157  7240  PTINIT, CLA CMA
04160  1151          TAD P7600
04161  3052          DCA CHRCNT       /REINITIALIZE THE CHARACTER COUNTER
04162  1327          TAD P6600
04163  3053          DCA PTBUFP       /REINITIALIZE THE BUFFER POINTER
04164  5333          JMP PTRIN+1

04165  1453  PTPKUP, TAD I PTBUFP     /PICK UP A CHARACTER
04166  2053          ISZ PTBUFP
```

```
04167  5732           JMP I PTRIN
04170  2325  PTEMTY,  ISZ ENDCNT
04171  5721           JMP I PAUSEP
04172  ~~5357~~ 7610  ~~JMP PTINIT~~ SKP CLA
04173  4141  MORTIM,  PTREAD

4174  1052            TAD CHRCNT
4175  7040            CMA
4176  5360            JMP PTINIT+1
```