DECUS
PROGRAM LIBRARY

DECUS NO.	8-668
TITLE	RAW - A REVERSE ASSMEBLER OF WINDSOR
AUTHOR	P. A. V. Thomas
COMPANY	University of Windsor Windsor, Ontario, Canada
DATE	November 2, 1973
SOURCE LANGUAGE	PAL III

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

RAW - A Reverse Assembler of Windsor

P.A.V. Thomas

ABSTRACT

This programme was written for a 4k PDP8 computer to obtain a symbolic programme from a binary programme produced by the PAL III assembler. The output obtained may be in the standard assembler output format or in a format suitable as input to the assembler for reassembling after modification. The programme will handle most of the standard mnemonics including EAE and floating point operations. The only known limitations are (i) a FIELD statement is not acceptable and (ii) subroutines with multiple arguments and/or returns will not give a properly formatted output but will have to be interpretted by the user.

0-1 Before using, load PATCH to correct counting error in S/R PRINT.

See LISTING @ 0144

0-2 Load Program to be DISSEMBLED and run ODT Masked Search to identify all HLT's

Operating Procedures

0. See Opposite

1. Load RAW Binary Tape.
2. Place Binary Tape of programme to be dis-assembled into the HS Reader.
3. Start programme from SA = 177.
4. Programme halts at location 223. Set Pass #1 by setting Switch Register bits 0 and 1 to 01 (as for PAL III Assembler). Hit "Continue" Key. Tape should be completely read in unless there are too many potential jump entries (greater than 1000 (Octal)) in which case a "Table Full" message will be typed out and no further dis-assembly can be carried out. Otherwise the computer should again halt at location 223.
5. Set Pass #2 by setting Switch Register bits 0 and 1 to 10 and replace programme tape (of (2)) in reader. Hit "Continue" Key again and (4) above repeats unless there are no "indirect jumps" in the programme in which case the programme tape will not be read in but a message will be typed out to this effect. In either case the computer will again stop at location 223.
6. Set Pass #3 by setting Switch Register bits 0 and 1 to 11 and replace programme tape (of (2)) in reader. Hit "Continue" Key again. The computer will now halt at location 273 and bit 11 of the Switch Register must be '0' if the output format is to be similar to that of a PAL III Listing or '1' if the output format is to be suitable for inputting to the PAL III Assembler. After setting this bit, hit the "Continue" Key again. The computer will then halt at location 2327 to accept the programme starting address which should be set in the Switch Register. After hitting the "Continue" Key the computer will again halt at 2327. Normally, ØØØØ should now be set in the Switch Register and the "Continue" Key again hit; the tape should now read in and the output listing should be printed on the TTY. If the programme being disassembled has any HLT instructions for which a "Continue" Key operation would be

followed, that is the programme contains somewhere the instruction sequence

HLT
(executable instruction)
.
.
.

then the address of the first instruction ((Location of HLT) + 1) must be loaded as a starting address by the following methods: after when the mainstarting address is loaded (as above) do not set ØØØØ into the Switch Register but set the address (location of HLT) + 1 and hit the "Continue" Key, which will load this address and again halt the computer at 2327 - up to a total of 8 addresses may be so loaded after which the termination code ØØØØ must be set in the Switch Register.

NOTE: If ØØØØ is a starting address change the contents of the following locations and use 7777 as the terminator.

Location	Change From	TO
2332	7ØØØ	11ØØ
2335	7ØØØ	13ØØ

7. The computer will finally stop at location 223 and a new programme tape can be dis-assembled by returning to step (4) above.

CAUTION

This programme has been developed for a 4k core PDP8 and no inclusion has been made for a FIELD statement which may cause a failure of the programme. Furthermore if there is an instruction JMS I 7(44Ø7) in your programme this is treated as an entry to the floating point package. If this is not desired you could try changing the content of Location 1235 from 11ØØ (TAD C1) to 7ØØØ (NOP) which I think should work.

SAMPLE PROGRAMME

The results obtained for a sample meaningless programme to show the various features are shown in the figures, in which the RAW output is shown together

with the original third pass listing for comparison purposes. In particular, the starting address is 3ØØ and locations 3Ø and 6Ø7 are restart addresses.

Figure 1 is a normal programme which shows that labels have the address prefixed with an L and variables are previxed with a V; in some cases both will occur, as at locations 24 and 26. Also constants in excess of 3777 are treated as being negative (see location 1ØØ). Finally each new memory page is stared even if not so in the original (see line between locations 177 and 2ØØ). Figure 2 shows how floating point instructions (triggered by the 44Ø7 at location 2ØØ) are presented; note that only 4 characters are used so that SQROOT becomes SQRT; also the three locations for a variable are all labelled rather than just the first (locations 223 - 225). Some EAE instructions are also shown including some with arguments such as the MUY at location 21Ø.

Figure 3 shows the results obtained of multiple arguments with subroutine calls. Locations 4Ø1 and 4Ø2 are arguments which become translated to equivalent instructions; however, the actual numbers are still available in the 2nd column of the output and cause no real problem as investigation of an actual subroutine would enable the user to determine how many arguments are required. If, however, an argument is in the range 5ØØØ to 5777 this appears as a jump (JMP) and in general some of the following items will then appear erroneously as variables; this is shown at locations 6Ø2 - 6Ø6; the change to instructions at 6Ø7 is due to this location having been set as a restart address after the halt at 6Ø6. Also shown here is the effect of having more than three mnemonics in a microinstruction in which the first three are printed terminated by the word FULL; the user can easily add the missing terms. The last line, location 613, should always be ignored as it is produced by the checksum punched on the source (binary) tape.

Figure 4, shows Figure 1 as it appears if the user chooses the option of outputting the information in the alternate format. It will be seen that it is acceptable to the PAL III assembler, the octal forms being separated off

with a comment slash.

REFERENCES:

1. "Symbolic Dis-assembler Design", P.A.V. Thomas, Proceedings, Canadian Western DECUS Symposium (1973).

2. "RAW - A Reverse Assembler", P.A.V. Thomas, Proceedings, DECUS EUROPE Seminar (1973).

```
                *7
0007   5600     5600

                *20
0020   0000     SR,0
0021   1033     TAD A
0022   0034     AND B
0023   2033     ISZ A
0024   5420     JMP I SR
0025   2020     ISZ SR
0026   1420     TAD I SR
0027   7402     HLT
0030   3435     DCA I C
0031   2020     ISZ SR
0032   5420     JMP I SR
0033   0003     A,3
0034   0100     B,100
0035   0213     C,K

                *100
0100   7630     D,-150

                *300
0300   6046     TLS
0301   7340     ST, CLA CLL CMA
0302   1100     TAD D
0303   7700     SMA CLA
0304   5177     JMP NEX
0305   6041     TSF
0306   5305     JMP .-1
0307   6046     TLS
0310   4020     JMS SR
0311   3713     DCA I E
0312   7402     HLT
0313   0211     E,J

                *177
0177   7300     NEX, CLA CLL


0200   4407     JMS I 7
```

3rd Pass Listing

```
                *0007
0007   5600     L0007,L5600

                *0020
0020   0000     L0020, 0000
0021   1033            TAD     V0033
0022   0034            AND     V0034
0023   2033            ISZ     V0033
0024   5420            JMP   I L0020
0025   2020            ISZ     V0020
0026   1420            TAD   I V0020
0027   7402            HLT
0030   3435            DCA   I V0035
0031   2020            ISZ     V0020
0032   5420            JMP   I L0020
0033   0003     V0033, 0003
0034   0100     V0034, 0100
0035   0213     V0035, 0213

                *0100
0100   7630     V0100,-0150

                *0300
0300   6046            IOT   04 6
0301   7340     L0301,CLA CLL CMA
0302   1100            TAD     V0100
0303   7700            SMA CLA
0304   5177            JMP     L0177
0305   6041     L0305,IOT   04 1
0306   5305            JMP     L0305
0307   6046            IOT   04 6
0310   4020            JMS     L0020
0311   3713            DCA   I V0313
0312   7402            HLT
0313   0211     V0313, 0211

                *0177
0177   7300     L0177,CLA CLL

                *0200
0200   4407            JMS   I L0007
```

RAW Output

FIGURE 1

```
0201  5223  FGET F
0202  0002  SQROOT
0203  6626  FPUT I G
0204  0000  FEXT
0205  7200  CLA
0206  1227  TAD H
0207  7421  MQL
0210  7405  MUY
0211  0000  J,0
0212  7407  DVI
0213  0000  K,0
0214  7420  SNL
0215  5222  JMP ERR
0216  3230  DCA L
0217  7701  MQA CLA
0220  3231  DCA M
0221  5377  JMP NEXT
0222  7402  ERR,HLT
0223  0003  F,3
0224  2000  2000
0225  0000  0
0226  3000  G,3000
0227  7500  H,-300
0230  0000  L,0
0231  0000  M,0

      *377
0377  7000  NEXT, NOP
```

```
0201  5223        FGET   V0223
0202  0002        SQRT
0203  6626        FPUT I V0226
0204  0000        FEXT
0205  7200        CLA
0206  1227        TAD    V0227
0207  7421        MQL
0210  7405        MUY
0211  0000  V0211, 0000
0212  7407        DVI
0213  0000  V0213, 0000
0214  7420        SNL
0215  5222        JMP    L0222
0216  3230        DCA    V0230
0217  7701        MQA CLA
0220  3231        DCA    V0231
0221  5377        JMP    L0377
0222  7402  L0222,HLT
0223  0003  V0223, 0003
0224  2000  V0224, 2000
0225  0000  V0225, 0000
0226  3000  V0226, 3000
0227  7500  V0227,-0300
0230  0000  V0230, 0000
0231  0000  V0231, 0000

      *0377
0377  7000  L0377,NOP
```

FIGURE 2

```
0400  4605  JMS I N
0401  3000  3000
0402  0020  20
0403  7200  CLA
0404  5206  JMP .+2
0405  0020  N,SR
0406  7120  STL
0407  1216  TAD P
0410  7430  SZL
0411  7360  CLA CLL CMA CML

0412  3215  DCA Q
0413  4617  JMS I R
0414  7402  HLT
0415  0000  Q,0
0416  4000  P,4000
0417  0600  R,SR3

            *600
0600  0000  SR3,0
0601  4020  JMS SR
0602  5000  5000
0603  1000  1000
0604  7200  CLA
0605  1212  TAD S
0606  7402  HLT
0607  7200  CLA
0610  5611  JMP I STI
0611  0301  STI,ST
0612  7000  S,7000
```

```
            *0400
0400  4605        JMS  I L0405
0401  3000        DCA    V0000
0402  0020        AND    V0020
0403  7200        CLA
0404  5206        JMP    L0406
0405  0020  L0405,L0020
0406  7120  L0406,CLL CML
0407  1216        TAD    V0416
0410  7430        SZL
0411  7360        CLA CLL CMA
        /  FULL
0412  3215        DCA    V0415
0413  4617        JMS  I L0417
0414  7402        HLT
0415  0000  V0415, 0000
0416  4000  V0416,-4000
0417  0600  L0417,L0600

            *0600
0600  0000  L0600, 0000
0601  4020        JMS    L0020
0602  5000        JMP    L0000
0603  1000  V0603, 1000
0604  7200  V0604,-0600
0605  1212  V0605, 1212
0606  7402  V0606,-0376
0607  7200        CLA
0610  5611        JMP  I L0611
0611  0301  L0611,L0301
0612  7000  V0612,-1000
0613  0242  V0613, 0242
```

FIGURE 3

```
          *0007
L0007,L5600              /0007  5600

          *0020
L0020, 0000              /0020  0000
       TAD    V0033      /0021  1033
       AND    V0034      /0022  0034
       ISZ    V0033      /0023  2033
       JMP  I L0020      /0024  5420
       ISZ    V0020      /0025  2020
       TAD  I V0020      /0026  1420
       HLT               /0027  7402
       DCA  I V0035      /0030  3435
       ISZ    V0020      /0031  2020
       JMP  I L0020      /0032  5420
V0033, 0003              /0033  0003
V0034, 0100              /0034  0100
V0035, 0213              /0035  0213

          *0100
V0100,-0150              /0100  7630

          *0300
       IOT  04 6         /0300  6046
L0301,CLA CLL CMA        /0301  7340
       TAD    V0100      /0302  1100
       SMA CLA           /0303  7700
       JMP    L0177      /0304  5177
L0305,IOT  04 1          /0305  6041
       JMP    L0305      /0306  5305
       IOT  04 6         /0307  6046
       JMS    L0020      /0310  4020
       DCA  I V0313      /0311  3713
       HLT               /0312  7402
V0313, 0211              /0313  0211

          *0177
L0177,CLA CLL            /0177  7300

          *0200
       JMS  I L0007      /0200  4407
```

FIGURE 4