dec PERMANENT MEMORANDUM

M-1129

PAGE 1 OF 5

DATE October 2, 1961

SUBJECT Octal Dump & Read-In-Mode Punch

TO PDP Distribution List

ABSTRACT This program provides a means for typing or punching out information from any section of core memory. The console typewriter provides the control information along with sense switch #1.

FROM

Steve Lambert

APPROVED BY

Harlan E. Anderson

OPERATING PROCEDURE

#1 Read-In Octal Dump - RIM Punch Tape. (Three tapes are presently available starting at registers 0, 4000, and 7000.)

#2 Set SS #1 to desired position. (SS #1 Up, read-in-mode tape will be punched upon completion of control statement "r". Down, Octal type-out will occur upon completion of control statement "f".)

#3 Type in starting address followed by the letter "b". (Leading zeros may be omitted.)

#4 Type in final address and the letter "f". (Leading zeros may be omitted. Octal Dump will occur after "f" is typed.)

#5 Type in an address and the letter "j" for a jump statement that will be punched at the end of the RIM tape. (This is not necessary if the jump statement is not desired.)

#6 Type "r" and the RIM tape will be punched. (Upon completion of a punch or type-out, the program will reset itself for entry of more control statements.)

Note: Should a mistake be made during type-in of control statements, type "b" as shown below or restart Octal Dump - RIM Punch program.

```
0b31 ← mistake
b0b30f
```

Example:

```
0b30f                          SS #1 down, type-out occurs.
↑↑↑↑↑
do not space
between characters
↓↓↓↓↓↓↓↓
0b30f5jr                       SS #1 Up, Punch-out results.
```

```
, Octal Dump & Read-In-Mode Punch
, 9/20/61
, S. Lambert
opd                 cliaf 764207
start               dzm s
                    dzm e
                    cliaf
                    dzm temp
read                szf * 1
                    jmp read
                    cliaf
                    tyi
                    rcr s6
                    sad b
                    jmp begin
                    sad f
                    jmp final
                    sad j
                    jmp loop
                    sad r
                    jmp punch
                    rar s6
                    rar s6
                    and seven
                    add temp
                    ral s3
                    dac temp
                    jmp read
begin               lac temp
                    rar s3
                    dap s
                    dap a
                    jmp start & 2
loop                lac read & 1
                    dac jump
                    lac temp
                    rar s3
                    dap jump
                    jmp start & 2
final               lac temp
                    rar s3
                    dap e
                    idx e
                    szs 10
                    jmp start & 2
                    lio cr
                    tyo *
                    law * 4
                    dac temp
                    lac s
                    sad e
                    jmp return
                    rcl s6
                    dac a
sr                  lac a
                    cli
                    rcl s3
                    dac a
                    jda type
                    isp temp
                    jmp sr
                    lio tab
                    tyo *
                    law * 6
                    dac temp
                    lac * s
                    dac a
pr                  ac a
                    cli
                    rcl s3
                    dac a
                    jda type
                    isp temp
                    jmp pr
                    idx s
                    jmp final & 6
type                0
                    dap xt
                    dio type
                    lac type
                    sza
                    jmp pt
                    lio zero
                    tyo *
xt                  jmp
pt                  lio type
                    tyo *
                    jmp xt
punch               law * 200          , adjustment for beginning leader
                    dac temp
                    cli
                    ppa *
                    isp temp
                    jmp ¢ - 3
```

```
pun                 lio a
                    jsp out
                    lio * a
                    jsp out
                    idx a
                    idx s
                    sas e
                    jmp pun
                    lio jump
                    jsp out
                    law * 30              , adjustment for final leader
                    dac temp
                    cli
                    ppa *
                    isp temp
                    jmp ¢ - 3
return              lac deposit
                    dac a
                    dzm jump
                    jmp start
out                 dap exit
                    ppb *
                    ril s6
                    ppb *
                    ril s6
                    ppb *
exit                jmp
r                   510000
j                   410000
b                   620000
f                   660000
s                   0
e                   0
a                   320000
temp                0
jump                0
deposit             320000
zero                20
tab                 36
cr                  77
seven               7
jmp start end.
```