PERMANENT MEMORANDUM NO. M-1112

SUBJECT: Binary Punch and Load Package DATE: June 26, 1961

TO: PDP Distribution List FROM: Computer Division

This routine allows the user to punch out any area of memory in the binary format*, specifying the desired area with typed commands. To load memory with a tape in binary format, it is necessary to utilize a binary loader. Since it is often inconvenient to load the binary loader into memory before reading in a binary format tape, this package will punch out a Read-In-Mode version of a binary loader on the first part of a tape thereby allowing the tape produced to load itself. The binary punch has three parameters: the location of the first register in the block to punch, the location of the last register in the block to punch or the length of the block, and the place to jump to after reading in the block.

The loader will read-in the specified block and jump to the specified location. If it is desired to punch a succession of blocks no jump location should be specified until the last block. If a block has been specified, typing the letter "P" will punch out the block without leader and will have trailer only if a jump has been specified. If the letter "R" is typed, the package will punch out leader, then a rim binary loader, then the block, and then trailer if a jump was specified.

The commands to specify a block are as follows:

0000b beginning of block
0000f final location of block
0000l length of block (use either "f" or "l")
0000j jump location
(Leading zeros need not be typed.)
(All numbers must be octal.)

A carriage return will erase a number any time before the letter is typed.

This package is available in three locations:

High..... 7520 - 7777 SA 7777
Medium..... 4000 - 4256 SA 4000
Low.....0001 - 0257 SA 0001

When the package is loaded it will be running and ready to accept typewriter commands. One typewriter command not yet described is "s". This will cause a binary tape to be read in (assuming it's in the reader and ready to go). After the tape has been read in the loader will jump to the location specified at the end of the binary tape unless SS#1 is up, in which case it will halt. The binary punch produces a check sum at the end of each block punched and the loader will halt at the end of read in if this sum didn't check. For the packages at the three

* See Appendix A

locations the reading of the program counter for a halt at check sum error and a halt because SS#1 up are shown below:

High	Check sum error	PC reads 7745
	Halt because SS#1 up	PC reads 7725
Medium	Check sum error	PC reads 4225
	Halt because SS#1 up	PC reads 4205
Low	Check sum error	PC reads 0226
	Halt because SS#1 up	PC reads 0206

The location in memory of the RIM Binary loader produced by the routine is shown below:

High	7700 - 7753
Medium	4160 - 4233
Low	0161 - 0234

Storage Requirements:

High	7520 - 7777
Medium	4000 - 4256
Low	0001 - 0257

Modification

The binary punch and load package as described above has been modified to include some new features and has been relocated slightly at its various positions in core. The positions occupied by the three punch and load packages are noted on the attached data sheets. The most important new feature is the sense switch #2 option for checking a tape against core. With sense switch #2 down, the loader will operate in the normal manner. With sense switch #2 up, the loader will check the data on the tape against the contents of core. If a difference is found, the loader will halt at a location noted on the attached data sheet and will display in the in-out register the word from tape and in the accumulator the word from memory. While the loader is performing this check, it will compute and check the check sum in the normal manner and will, if sense switch #1 is down, execute the jump to the program being read in at the end.

The symbol ">" has been defined to mean put a zero in the register which is assembling the octal number. Previously this was accomplished by a carriage return. Typing anything other than a control symbol will cause an input to the register. Therefore, if a carriage return or any undesired type-out is performed, use the symbol ">" to clear this register before typing in any further commands.

The command "b" will now have the added function of deleting a previously designated jump order.

Please note on the data sheet the new position occupied by the loader itself in the lo and mid packages.

An attempt to punch out the binary punch and load package using the binary punch and load package to accomplish this will lead to difficulties and should not be attempted except by following exactly the punch out record that is on the leader of the binary punch and load package tape.

Program Control Rider for Binary Punch and Load Package

A sub program of the Binary Punch and Load Package is provided to allow operation of the package from core. Binary blocks may be punched with or without a RIM loader and a jump block may be designated.

To designate a jump block:

Load the in-out register with the jump location.
jsp pj, (Location of pj for each of the packages is noted on the data sheet).
r, Returns here after setting up the jump.

To designate a block with a RIM loader and punch it:

Load the accumulator with the beginning of the block.
Load the in-out register with the length of the block.
jda pbr, (Location of pbr for each of the packages is noted on the data sheet.)
r, Returns here after punching the block designated with a RIM loader and the jump block if it was designated previously.

To designate a block without a RIM loader and punch it:

Load the accumulator with the beginning of the block.
Load the in-out register with the length of the block
jda pb, (Location of pb for each of the packages is noted on the date sheet).
r, Returns here after punching the block designated and a jump block if it was designated previously.

Example: To punch a block from 1000-1477 with a RIM loader; then a block from 2153-2200 with a jump to 1000.

```
law 500
rcl s9
rcl s9        ,set length in io
law 1000      ,set beginning in ac
jda pbr       ,punch block with RIM loader
law 1000
rcl s9
```

```
        rcl  s9        ,set jump location in io
        jsp  pj        ,set the jump
        law  16
        rcl  s9
        rcl  s9        ,set length in io
        law  2153      ,set beginning in ac
        jda  pb        ,punch block and jump block
        hlt
```

or:

```
        lac  rc-beginning1
        lio  rc-length1
        jda  pbr
        lio  pbr
        lio  rc-beginning 1
        jsp  pj
        lac  rc-beginning2
        lio  rc-length2
        jda  pb
        hlt
rc-beginning1  1000
rc-length1     500
rc-beginning2  2153
rc-length2     16
```

DATA SHEET FOR THE BINARY PUNCH AND LOAD PACKAGES - II

	Lo	Mid	Hi
Start at	100	4100	7500 or 7777
Registers used	1-277	4001-4277	7500-7777
Loader uses registers	1-77	4001-4077	7700-7777
If SS#1 up; after loading routine halts at pc reading:	34	4034	7733
Check sum error - routine halts with pc reading: (ac contains computer sum; io contains sum on tape)	55	4055	7754
Tape to core comparison error- routine halts with pc reading: (ac contains word from core; io contains word from tape)	24	4024	7723
Address of offending word in memory is in address part of register:	20	4020	7717
The program control rider goes in registers	300-324	4300-4324	7453-7477
pb is at	300	4300	7453
pbr is at	313	4313	7466
pj is at	322	4322	7475

CONTROL CHARACTERS FOR BINARY PUNCH AND LOAD PACKAGE

Let α_i be an octal number of up to six digits. Then:

αb	designates a block to begin at α and erases any previously designated "j"
α_1l	designates a block of length α_1
α_2f	designates a block to end at α_2
α_3j	designates a block to be punch which will cause the binary loader to jump to α_3
p	causes the block defined by the characters above to be punched out. Either "f" or "l" should be used and a "j" should be used if this is the last block to be punched out. Trailer will be punched only if a "j" was used. Leader will not be punched.
r	causes leader to be punched, then punches a RIM loader with a jump to itself and finishe up with "p"
>	clears the register which assembles the octal number
s	starts the binary loader; a tape must be in the reader and the clutch engaged before the "s" is typed

SENSE SWITCH OPTIONS

SS#1	SS#1 up causes the binary loader to halt after reading in a binary tape rather than jumping to the register designated by the jump block. After a halt the jump may be performed by depressing the "CONTINUE" switch.
SS#2	SS#2 up causes the binary loader to compare the words on the tape with the registers in memory into which they would have been stored had ss#2 been down. A halt occurs if a difference arises. The check sum is computed as during a normal loading operation and the jump at the end of the tape will be performed unless ss#1 is up.

APPENDIX A

Binary Tape Format

Location Block

Initial Location
Length
Check Sum

Program Block

Check Sum

Jump Block

JMP Location
Check Sum

Length of block may be as long as you want.