| NUMBER: | Digital - 4 - 11 - U |
| NAME:   | RIM Puncher |
| AUTHOR: | J. M. Graetz - DEC |
| DATE:   | September 28, 1962 |
| SPECS:  | 1038 registers: 100-102, 7000-7002-7202 |
|         | Tapes: P10-DEC, RIM, SA 100, RIM, SA 7000 |
| NEEDED: | RIM Loader (I-1) |
| PURPOSE:| To punch a Readin-Mode tape from any area of core memory. |
The RIM puncher will punch a read-in-mode tape with start block from any area of core. The tape format of the output consists of thirty inches or about 3 1/2 fanfold units (ffu) of leader, data blocks in read-in-mode format, a start block consisting of a jump instruction and a blank dummy word to stop the tape reader, and a few inches of trailer.

Usage:
1. Read in the desired version (high or low) of RIM Puncher.
2. AC Switch zero must be down. Set the first address of the block to be punched in the AC Switches and press continue.
3. When program stops, set the final address of the block in the ACS and press continue. (If ACS₀ is up at this point, the program will refuse to proceed.) caution.
4a. If ACS₀ is down when the program stops punching, pressing continue at this point will cause the address now in the ACS to be taken as the first address of a new block of data to be punched. In this case, the procedure is repeated from step 2. pressing continue indicates that
4b. If ACS₀ is up, the address in the ACS will be taken as a starting location and a start block followed by trailer will be punched.
5. If a new tape is desired after the start block has been punched, put ACS₀ down and repeat from step 2.

Error Stops:
There are no error stops in RIM Puncher. The only halts are those indicated above.
RIM PUNCHER MK IIIa  28-9-62

/ACS-0 down, first address in ACS and continue.
/On halt, last address in ACS and continue.
/On halt, ACS-0 down if new block, up if start block. Address
/in ACS and continue.
/For new tapes, repeat procedure from the top.

/Two versions:  low--SA 100; high--SA 7000

rimp,
  hlt
  lam -400
  jms feed
  las

rim2,
  xor (dac
  dac blast
  hlt
  las
  spa
  jmp rim2 2
  add (dac 1
  dac blast
  pch,
  lac blast
  sad blast
  jmp pch
  jms pib
  lac blast 1
  jms pib
  isz blast
  jmp pch
  lam -20
  jms feed
  hlt
  las
  sna
  jmp rim2
  xor (200000
  jms pib
  cla
  jms pib
  lam -131
  jms feed
  jmp rimp

feed,
  0
  dac hold
  pls 10
  psf
  jmp -1
  isz hold
  jmp feed 2
  jmp feed 1
pib, 0
dac temp
lac temp
rtl
dac temp
ral
plsa
psf
jmp .-1
isz chrc
jmp plin
jmp pib 1
bfst, 0
blast, 0
chrc, 0
temp=feed
hold=pib
chrc/
start rimp