

1. IDENTIFICATION
- 1.1 Digital-8-5-U-Sym
- 1.2 Binary Punch (ASR 33 or 75A)
- 1.3 March 1, 1965

2. ABSTRACT

This program provides a means of punching out information contained in selected blocks of core memory as binary-coded tape via the ASR-33 perforated tape punch or via the high-speed punch 75A.

3. REQUIREMENTS

3.1 Storage

This program occupies 75 (decimal) core memory locations.

3.3 Equipment

Basic PDP-8 with standard ASR-33 or in addition the 75A high-speed punch.

4. USAGE

4.1 Loading

This program is loaded by means of the Binary Loader. See DEC-08-LBAA-D for a complete discussion of the Binary Loader and its use.

4.2 Calling Sequence

None. This program cannot be called as a subroutine.

4.3 Switch Settings

The switch register is used to enter initial and final addresses of blocks to be punched as well as the number of blocks to be punched.

4.4 Start Up/Entry

This program is used in the following manner.

4.4.1 Assuming the program is in memory as listed in 10.4, place the starting address 7465 in the switch register and press the LOAD ADDRESS key.

4.4.2 Press the START key. Leader will be punched and the computer will halt. Set the number of blocks to be punched into the switch register and press CONTINUE.

4.4.3 The computer will halt. Set the initial address of the block to be punched into the switch register and press CONTINUE.

4.4.4 The computer will halt. Set the final address of the block to be punched into the switch register and press CONTINUE.

4.4.4.1 Note that the final address must be greater than the initial address.

4.4.5 The indicated block of data will be punched. If only one block has been called for, the trailer will punch and the computer will halt. If more than one block has been called for, the computer will halt at step 4.4.3 waiting for a new initial address. The second block will punch following completion of steps 4.4.3 and 4.4.4, etc.

## 7. METHOD

### 7.1 Discussion

This is a basic routine used to produce tapes acceptable by the Binary Loader  
See DEC-08-LBAA-D.

Within each block, an initial address into which data is to be loaded is punched as the first two characters. Following the initial address, each 2-character group represents the binary contents of a computer word. At the end of each block, a 2-character checksum is punched.

Reference to Section 11.1, Flow Chart, will illustrate the computational approach. Basically data is picked up from memory, the most significant half shifted right and punched, and the least significant half masked out and punched.

A similar process is followed with respect to the initial address and the checksum which is accumulated character by character as a block is punched.

## 8. FORMAT

### 8.1 External Data

See DEC-08-LBAA-D for a complete discussion of tape format.

## 9. EXECUTION TIME

This routine is output limited.

## 10. PROGRAM

### 10.4 Program Listing

A listing of this program with BPUN located at 7465 is as follows:

```
          /BIN PUNCH 75A
          *7465
7465     7300     BPUN,      CLA CLL
7466     6026           PLS          /INITIAL PUNCH
```

7467	3366		DCA CKSM	/CLEAR CHECK-SUM
7470	4330		JMS PLOT	/GO PUNCH LEADER CODES
7471	7402		HLT	/SET SWITCHES = NUMBER OF BLOCKS
7472	7602		LAS	
7473	7041		CIA	
7474	3367		DCA NB	/STORE MINUS NUMBER OF BLOCKS
7475	7402	NXBL,	HLT	/SET SWITCHES = INITIAL ADDRESS OF /BLOCK
7476	7604		LAS	
7477	3370		DCA IA	
7500	7402		HLT	/SET SWITCHES = FINAL ADDRESS OF /BLOCK
7501	7604		LAS	
7502	7001		IAC	
7503	3371		DCA FA	
7504	1370		TAD IA	
7505	7120		STL	/TO PUNCH IA AS ORIGIN
7506	4341	PUNL,	JMS BINP	/GO PUNCH WORD AS TWO LINES OF /TAPE
7507	1370		TAD IA	
7510	7041		CIA	
7511	1371		TAD FA	/AC = FA - IA
7512	7650		SNA CLA	/WAS IT LAST WORD OF BLOCK?
7513	5320		JMP .+5	/IT WAS THE LAST WORD
7514	1770		TAD I IA	/GET WORD TO PUNCH
7515	7100		CLL	/NOT AN ORIGIN
7516	2370		ISZ IA	/JUST INDEX IA
7517	5306		JMP PUNL	
7520	2367		ISZ NB	/IS THERE ANOTHER BLOCK?
7521	5275		JMP NXBL	/HANDLE NEXT BLOCK
7522	1366		TAD CKSM	
7523	7100		CLL	
7524	4341		JMS BINP	/GO PUNCH CHECK SUM
7525	4330		JMS PLOT	/GO PUNCH TRAILER CODES
7526	7402		HLT	/DONE
7527	5265		JMP BPUN	
7530	0000	PLOT,	0	
7531	7300		CLA CLL	
7532	1372		TAD M212	/TO PUNCH 212 OCTAL LEADER /TRAILER CODES
7533	3373		DCA CTR1	

7534	1374		TAD C200	/LEADER TRAILER CODE
7535	4361		JMS PUN	/PUNCH C (AC)
7536	2373		ISZ CTR1	/ANOTHER L-T CODE OR NOT?
7537	5335		JMP .-2	/GO PUNCH ANOTHER.
7540	5730		JMP I PLOT	/EXIT
7541	0000	BINP,	0	
7542	3375		DCA TEM1	
7543	1375		TAD TEM1	
7544	7012		RTR	
7545	7012		RTR	
7546	7012		RTR	
7547	0376		AND SL 7	/FIRST TWO OCTAL DIGITS IN AC 5-11
7550	4361		JMS PUN	/PUNCH C (AC)
7551	1366		TAD CKSM	
7552	3366		DCA CKSM	
7553	1375		TAD TEM1	
7554	0377		AND SL6	/LAST TWO OCTAL DIGITS IN AC 6-11
7555	4361		JMS PUN	/PUNCH C (AC)
7556	1366		TAD CKSM	
7557	3366		DCA CKSM	
7560	5741		JMP I BINP	/EXIT
7561	0000	PUN,	0	/ROUTINE TO PUNCH C (AC)
7562	6021		PSF	/AND EXIT WITH C (AC)
7563	5362		JMP .-1	/UNALTERED
7564	6026		PLS	/PUNCH IT
7565	5761		JMP I PUN	/EXIT
7566	0000	CKSM,	0	
7567	0000	NB,	0	
7570	0000	IA,	0	
7571	0000	FA,	0	
7572	7566	M212,	-212	
7573	0000	CTR1,	0	
7574	0200	C200,	200	
7575	0000	TEM1,	0	
7576	0177	SL7,	177	
7577	0077	SL6,	77	

To use this program with the ASR-33 make the following changes:

7466	6046	TLS	/INITIAL PUNCH
7562	6041	TSF	/AND EXIT WITH C (AC)
7564	6046	TLS	/PUNCH IT

II. DIAGRAM

II.1 Flow Chart

Note that in this diagram circles represent subroutine utilization not connectives.

