IDENTIFICATION: HIGH SPEED PUNCH (HSP-1) ROUTINE II

PURPOSE: To allow simple control of the HSP-1 fast paper tape punch for output in a flexible format. This routine can be used as an extension of the punching operation of DECAID I or the Octal Utility Package. In addition, it contains several subroutines which provide various operations to a program which might output on paper tape.

RESTRICTIONS: No less than three non-zero characters can be punched by this routine. Each word is punched in a binary format of 6-8-8 bits. By proper positioning in the word, six-bit Flexowriter codes can be punched by this routine.

SPACE: The program occupies one long line. In addition, it uses F11-F15 for temporary storage. Since the main program does not use the last two sectors of the long line, 254 and 255, the routine will operate in lines 05 or 06 without interference from DECAID I or the Octal Utility Package.

TIMING: The timing of the routine is determined by the speed of the punch, which is 110 characters per second. At this speed it requires approximately 7.5 seconds to punch one long line in standard binary format.

USE: The routine is available in decimal relocatable format and will operate from any long command line.

1. To enter the routine from DECAID I or the Octal Utility Package, it is only necessary to transfer to sector 000 of the correct line.

The punch motor will start and the light on the Flexowriter will come on, indicating keyboard control. It will now be possible to perform various operations by depressing keys on the keyboard.

a. To punch single characters on the tape, depress the BREAKPOINT switch and anything typed will be punched. When all identifying characters have been typed and punched, the BREAKPOINT switch should be raised.

   All the following operations require the BREAKPOINT switch to be in the raised position.

b. To punch one long line in binary format, type the line number (in octal) followed by a B (LLB). Any line from 00 - $77)_8$ may be punched. After punching the line, the routine will return control to the keyboard but will not turn off the punch. If LL is preceeded by "D" (DLLB), the line number specified will be taken to be decimal (base 10).

c. In order to turn off the punch and return to DECAID I or the Octal Utility Package, type a I. (Do not depress the ENABLE switch before typing the I or the punch will not be turned off).

d. To punch leader, the tape feed button on the punch front panel may be held down until the desired leader is produced. However, the routine will automatically punch 13 inches of leader and return to keyboard control when the L key is depressed. The BREAKPOINT switch must be in the raised position.

e. The program will also punch 13 inches of trailer automatically when an X is typed. In this case, after punching, the routine turns the punch off and returns to DECAID I or the Octal Utility Package.

2. Subroutines:

There are several subroutines contained in this punch routine; the calling sequence of each is similar:

1) Load B with return command.

2) Load A with argument, if needed.

3) Transfer to appropriate entrance.

The various subroutines are as follows:

a. Entrance 64: Punch one line in standard binary format, ending with check sum. The line number should be in the A register at a scaling of 21. It is not necessary to rearrange the most significant bit of the line number.

   If the sign of the A register is negative, a "W" code will be punched at the end of the binary block.

b. Entrance 96: Leader will be punched to any desired length. The number of frames of leader should be in the A register at a scale of 21. The A register should be positive. After entering this subroutine, the number of frames specified plus one will be punched and control returned to the calling program. There are nominally 10 frames per inch of tape.

c. Entrance 128: The contents of the A register will be punched as three characters, the first of 6 bits, the second and third of 8 each.

d. Entrance 160: Punch 13 inches of leader. No argument necessary in the A register.

e. Entrance 192: This subroutine will punch all or part of a line starting at sector 000 and proceeding up

through the sector indicated. It will also accumulate a logical sum of the information being punched and store it in F14. If desired, the calling program can pick this word up and have it punched as three characters by using the 128 entrance. The argument for this subroutine is a LOAD A (LDA) command with the sector number of the last sector to be punched, plus the appropriate line number. The LDA should not have either a sequence tag or index tag.

METHOD:

The words to be punched are picked up and broken into three characters of 6-8-8 bits. After each character has been positioned in the A register, the program goes into a wait-to-punch routine. This routine is a series of commands repeated four times around the line. A TES for punch busy is the first command executed. If the punch is busy, control passes to the next TES. If, however, the punch is ready for a character, a WOC command with the proper sector is added to the character in the A register. This command is stored in the fast line along with a return transfer, and control is transferred to the WOC. The WOC executes for one sector time and control is then transferred back to the main program.

In the time between characters, the new words are added to the check sum, the variable load command is incremented and tested, and the returns from the short line are set.

The one word punch and the punch from sector 000 routines modify appropriate parts of the general routine in order to handle these slightly different modes of operation. The general routine initializes any parts which may be modified by other routines.