

Programming Staff Note 14

Computation Center

October 30, 1963

FROM: Janet Allen

SUBJECT: Diagnostic Programs for the 7094 CTSS Background

The following diagnostic programs are available to be run as background to the CTSS:

(1) CORE sets all of core higher than the program and then continually checks for proper values.

(2) CNSOL sets all of the lights and registers on the console and then checks them.

(3) TAGCK sets all seven index registers, then runs in both tag modes, checking to see that the correct tag mode is in effect and that the index registers contain the correct values.

(4) TAPCK writes a tape on each of Channels A and B, then reads and checks the values read, backspaces, skips records, and reads in the wrong mode.

Another diagnostic program, similar to TAPCK, which runs with traps enabled, disabled, and inhibited, is currently being debugged. All four programs are in absolute row binary and use a two card loader to load on-line. Each program, when sense switch two is down, prints on-line, using NPRINT, at the completion of each pass. Sense switch two in the normal position causes this printout to be skipped. When any error occurs, the program prints an error indication and the contents of the register (if applicable) and halts. Pressing start will cause the program to continue checking. The programs run in infinite loops, returning to restore the set values only upon the occurrence of an end.

#### I. Operating Instructions

All programs begin at location 62(octal). Thus, if it

is desired to begin again without loading, the operator may transfer to that location. Operating instructions for all four programs are the same.

1. Deck--two card loader  
Main program  
Subroutine NPRINT  
Transfer card
2. Ready deck card reader.
3. TAPCK only--scratch tapes on A4, B1.
4. Sense switch two down if printing at end of each pass is desired.
5. Keys 21, 34, and 35 must be down to load cards while CTSS is in operation.
6. Clear and load cards.
7. On any halt (HTR \*+1), check printer.  
Press start to continue testing.

## 11. Program Descriptions

A. CØRE initially stores 1's in memory locations higher than those used by CØRE and INPRINT. The routine then begins checking each location for all 1's. On finding an incorrect value, CØRE sets an error switch prints "ERRØR AT LØCATION xxxxx", and halts (HTR \*+1). Pressing start causes the routine to continue checking. At the end of each pass, if sense switch two is down, the routine prints "PASS CØMPLETE." If the error switch has been set, return is made to store all 1's. If no error has occurred, the routine begins checking again.

B. CNSØL initially sets the indicators, enters 7-tag mode, sets all index registers, sense lights, the accumulator and MQ registers, and turns on the divide check, overflow, and I/Ø indicators. It then begins checking.

On any error the routine stores the contents of the register (if applicable), sets index register one to the location of the BCD name of the register in error, and enters

ERPRT. This subroutine turns off the I/O light, enters 3-tag mode, prints the error indication and contents, and halts. Pressing start will cause ERPRT to enter the 7-tag mode, restore the registers it has used, and return to the main program to continue checking. The main routine then restores whatever register was in error if ERPRT has not done so.

Tests are first made on index registers 1, 2, and 4. The tag mode is checked and the index registers 3, 5, 6, 7. Checks are then made on the accumulator, MQ, sense indicators, sense lights, and I/O, divide check, and overflow indicators. The test is now complete. If sense switch two is down, the routine enters the 3-tag mode and prints. It then re-enters the 7-tag mode and returns to the beginning to restore the I/O, divide check, and overflow indicators and the MQ and accumulator, and resume testing.

C. TAGCK initially entering the 7-tag mode, sets all seven index registers. It then re-enters the 3-tag mode and, using tags of 3, 5, 6, and 7, checks that it is in the 3-tag mode. The 7-tag mode is entered, and tags 3, 5, 6, and 7, are again used to check the tag mode. This completes the pass; and, if sense switch two is down, printout occurs. On any error, printout occurs, giving the tag mode and the tag being used, before checking is continued.

D. TAPCK---A test of tape and channel operations without channel traps, TAPCK initially writes two tapes, one on Channel A and one on Channel B. Each tape then contains one file of four binary records, followed by two end of file marks, and a file of two BCD records, also followed by two end of file marks. The routine then begins testing.

If an unexpected condition occurs, the routine places in the accumulator the location of a format statement to be used by NPRINT and enters subroutine PRTCHN, if the channel is to be printed, or PRTERR if the channel does not apply (as in I/O check). Both enter subroutine PRNTR, which prints an error line containing channel indication if applicable, followed by the line specified by the format statement. The subroutine then halts. Pressing start causes PRNTR to return, depending on what error occurred, either to continue testing or to re-write the tapes and begin again.

On a real tape redundancy, occurring ten consecutive times, the printed line indicates that the tape must be changed.

Subroutine TAPEX does the actual tape reading of a record. It is entered with the location of the read select in the accumulator and in location (1,4) the locations of the I/O command used to write the record and the I/O command to be used to read it. TAPEX sets up the correct channel and tape commands and reads the record. It checks for redundancy and end of file. An end of file causes an error printout and return to the beginning to rewrite the tapes. On a redundancy check, TAPEX sets an indicator for later reference.

If neither has occurred, TAPEX computes the number of words read and compares this with the number of words that were written. If they are the same, the routine compares each word read with each word written. If these compare correctly, TAPEX checks to see if a redundancy occurred. If it has, TAPEX enters a subroutine which prints this on-line. Return is then made to the main testing. If the word count was wrong or if a word read didn't compare correctly, TAPEX enters routines which first check on the redundancy indicator. If a redundancy

did occur, the routine backspaces and reads again, making five consecutive tries. If a redundancy still occurs, the routine enters the print subroutine. If no redundancy was indicated, an error printout occurs, and return is made to rewrite the tapes.

The program begins testing by reading record one from both tapes using subroutine TAPEX and skipping record two. It then tests the I/Ø light and, if it is on, enters a subroutine to print and return. Using TAPEX, the routine reads the second record on both tapes. The routine skips records three and four. An end of file causes an error printout and return to this point in the routine. The two end of file marks are then read. If either one is not present, an error printout occurs, and return is made to the beginning in order to rewrite the tapes.

The routine enters TAPEX to read the first record (BCD) of the second file. It then tries to read the second record, also BCD, in binary. If a redundancy does occur, an error printout is made, and the testing continues. The routine backspaces a file over the second file and the second end of file mark, backspaces another file over the first end of file mark, backspaces a record over the fourth record of the first file, and then enters TAPEX to read this record.

As a final test, the routine backspaces a file, which should return to the load point, executes two record backspaces which should cause a delay until it gets to the load point, and then checks the begin tape indicator. If this is not on, an error printout occurs. If sense switch two is down, the routine prints "ONE PASS COMPLETE" and then returns to the first test.

haa