bcc	title CHIO Phase 1 Test Routines		prefix/class-number.revision PCNR/M-17
checked	authors Paul Heckel	approval date 12/9/69	revision date
checked		classification Manual	
approved Mel		distribution Company Private	pages 7

ABSTRACT and CONTENTS

This document describes a QSPL program that tests the CHIO to see if it appears to be working. The program does not make an exhaustive test of the CHIO but it should catch any gross errors. Bringing up the CHIO will consist of getting this program working (plus some extra debugging). At that time this document will be reissued with an appendix showing the output involved in the run.

LOADING THE CHIO

Before any testing can occur, the CHIO must be loaded. This means that its scratch pad registers and the core that it requires should both be properly initialized. This may already be done for one reason or another; if it is not, then it may be accomplished by saying: .LOAD;G.

RUNNING THE TEST

The test is started by saying .TEST;̂G. The user should keep a listing of the program with him because extensive error diognostics are not given. It is very easy for one familiar with DDT to find the error however. After each major test a message is output indicating that that test was sucessful. Test failure is indicated by a short error message "TEST FAILED" and a halt; the user should use DDT to find the location of the error and then check the listing to see the test that was being performed. DDT can also be used to find out the precise reason for the test failure.

CPU/CHIO INTERFACE TESTS

The first test is a test of the CHIO's block move instructions. First it will output a message indicating the expected new contents of core after the block moves, and then it prints out the values. They should be compared to see if the test succeeded. In actual practice, this is really a test of the basic loop of the CHIO. Most likely it will fail by typing out nothing due to a basic failure in the main flow of program control, rather than fail because of an error in block move.

The next test checks the scratch pad register load and store instructions. Again, it will print out a message asking for verification.

The third test checks the field load and store instructions. If an error occurs it will stop. A quick check of the listing will tell the reason. It will have failed because a PUTFIELD at location Ln failed, or the corresponding GETFIELD at location Mn failed; the failure will occur at location Mn plus some small constant.

The next test checks the read and write string operations for gross errors, and errors that might occur due to buffer malfunction. Each line from Ø to 9 has the string LINEM, n repeatedly written in it until the buffers are exhausted. (n is A for line Ø, B for line 1, etc.) The characters are then read out of the buffers and printed on the teletype. This operation is done three times and the result should be the same each time. The first read from the buffer is done by RCND, and all of the others are done by RSTB. The number of characters requested starts at zero for RCND and is incremented by 2 for each new line. For RSTB, it starts at 21 and is decremented by 2 for each new line. Thus, the output would look like this for the first two lines

```
(No characters for first RCND)
LINEA,ALINEA,ALINEA,A
LINEA,ALINEA,ALINEA,A
LI
LINEB,BLINEB,BLINEB
,BLINEB,BLINEB,BLIN
EB,B
```

The next test is for a more detailed test of the read and write string command. It will output messages to a teletype to indicate the expected character output. It will stop if it finds an illegal condition.

TESTING HIGH SPEED LINES

None of the previous tests have tested any actual input/output devices. The high speed line test subroutine can be used to do that. The selected output line, SOUTL (normally Ø) should be fed into the selected input line SINL (normally 1). The 2400 baud line test will send a string out on the 2400 baud line and then read it in again, printing it on the teletype and inserting a carriage return after each break character.

TESTING THE TELETYPE LINES

A teletype line TTYT (normally Ø) will next be tested. It will ask for a string to be supplied. The user types the requested string (or any other string terminated by a carriage return) on the specified teletype and it will type the string back to the user (on the selected teletype) inserting a carriage return after each break character. The system will then stay in this state, waiting for other strings to be typed in until the user hits rubout on the control teletype to return control to DDT. The characters will be typed out in a particular order, first the characters until the first break character, then an inserted carriage return, then those characters echoed, then the characters to the next break, etc. Non-echoable characters will be typed out once because they are not echoed.

GENERAL INFORMATION ABOUT THE TESTS

Associated with each of the above tests is a switch which may be turned off (set to zero) to prevent the selected test from being called. Any particular test may be selected by branching to the beginning of the test, and putting a breakpoint at the beginning of the next test.

Whenever a process is awakened, a message indicating this is typed out on the teletype. Running of this test procedure will affect several teletypes and will require a system restart to continue.