

dec

PERMANENT  
MEMORANDUM

M-1120

PAGE 1 OF 7

DATE

August 9, 1961

SUBJECT Checkerboard Program For 1,024 and 4,096 Word Memories  
TO PDP Distribution List

ABSTRACT

This is a maintenance program for checking the performance and reliability of magnetic core memory and its sense amplifiers. The program is designed for Ferrite Core memories to give maximum noise on the sense windings while a margin test is performed on the sense amplifiers.

FROM

Steve Lambert

APPROVED BY

*Harlan E. Anderson*

## INTRODUCTION

There are four possible configurations of checkerboard patterns generated by this program. Each pattern is designed for a particular type of memory, i.e., one for every read, write and sense winding configuration. To select a checkerboard pattern, put up one of the sense switches as mentioned in the notes. After selecting the desired pattern, place the checkerboard tape in the reader, turn reader on, and activate the read-in switch. If one of the following sense switches is up (3,4,5, or 6), the program will begin immediately loading the selected pattern. However, if no sense switches are selected, then the reader will continue to read in the next program on tape.

## CHECKING AND COMPLEMENT CHECKING

During the initial loading of the selected checkerboard pattern, a test is made on each register for errors resulting from bits lost or gained. After the checker pattern has been written into memory, an individual register complement and check test is made. Should an error be found either during initial loading or individual complement and checking, the computer will come to a halt, the address of the register being tested will appear in the In-Out and its contents in the accumulator. Should an error appear during testing, it is recommended that one press the continue button to find other faulty registers. Once memory has been completely checked and no errors found, the program restarts itself, this time writing the complement of the present pattern. The program continues indefinitely within this closed loop until the operator stops the program or a faulty component brings the computer to a halt.

It is possible to determine what part of the program the computer is executing by observing the program flags. When program flag 1 is on, checkerboard is either writing or checking the pattern as shown in the notes. If program flag 1 is off, then checkerboard is writing the complement of this pattern. During the time that program flag 2 is on, the program is doing the initial loading or complement loading of the selected pattern. When 2 is off, the program is complementing and checking individual registers. When a register is complement checked, the contents of this register are first checked, then complemented, and written back into the same register, brought out again and checked, complemented, written in, brought out and checked. To determine whether or not the program is doing a first or second complement, program flag 6 is on during the first complement and off during the second complement.

NOTES:

When complement checking individual addresses with sense switch 1 up, one cannot check register 7777 on the low checkerboard or 7600 on the high for 4,096 word memories. This is due to an indexing feature within the program.

Checkerboard

SS #1      Up, gives the operator the option of selecting a register from 0 to 7600 with test word switches for a continuous complement and check.

SS #2      Down, is for 4096 memory locations. Up, is for 1024 word memory.

SS #3      Pattern A                    100110011001100. . . .  
   01100110011  
   011001100  
   1001100  
   10011  
   011  
   0  
   1

SS #4      Pattern B                    1100110011001100. . . .  
   11001100110011  
   0011001100  
   00110011  
   1100.  
   11  
   0  
   0

SS #5      Pattern C                    1100110011001100. . . .  
   001100110011  
   0011001100  
   11001100  
   110011  
   0011  
   00  
   1  
   0

SS #6      Pattern D                    100110011001100. . . .  
   10011001100

011001100  
0110011  
10011  
100  
0  
0

### OPERATING INSTRUCTIONS

1. Set the sense switches to the desired positions.
2. Turn on the marginal check switch, Panel 3D. Be sure that the switch marked +a+b is thrown to the +a side.
3. Read in the checkerboard tape.
4. Begin varying the margins in the positive direction until the computer halts. There are two normal halts in the checkerboard program, either high or low version. In the high version, the first halt, which is in the checking part of the program, is at register 7623. The other halt is register 7634, which is in the complement checking part of the program.
5. After a halt has occurred, write down the address at which the halt appeared and the contents of this address. Reset the margins to +10 and start the program at either 0 or 7700 depending on which checkerboard you are using. To verify the positive margin, increase +10 again until the computer halts. Record the results in the computer log book and bring the +10 back down to normal. Now vary the positive margin towards 0 until the computer halts. Note the memory address at which the halt occurred. Record all test information in the computer log book.
6. Should bit errors occur when varying the margins + or -, note which bit it is and then check with a scope both the slice and gain in the sense amplifier packages located in Bay 3D. The best method is to look with a scope on the checkpoints of the sense amplifiers while syncing the scope on the defer level, either pin R or Z on 1K13. After varying either the slice or gain controls or both, recheck the margins.

```
,Checkerboard for 1024 & 4096 word memories
,generates 4 checker patterns
,HIGH CHECKER
,7/18/61
,s. Lambert
opd          nop 760000
org 7600
start        764207
             stf 11
             stf 12
             dzm q
write        jsp pattern      ,initial load & complement load
             lac n
isop2        nop              ,inst. is determined by pat. gen.
             dac * q
             jmp check
             idx q
             sas final
             jmp write
             clf 2
             dzm q
check        jsp pattern      ,load check
             lio q
             lac * q
isop         nop              ,inst. is determined by pat. gen.
             sas n
             halt
             szf 2
             jmp isop2 & 3
comp         stf 16           ,individual reg. complementing &
                                 ,checking
             lac * q
             cma
             dac * q
             xor * q
             sza
             hlt
             szf * 6
             jmp c & 3
             clf 6
             jmp comp & 1
             idx q
             szs 10
             lat              ,hit same reg with compliment check
             dpa q
             sas final
```

```

                                jmp check
                                lac 0
                                sza *
                                jmp c & 5
                                cla
                                dac n
                                clf 1
                                jmp start & 2
                                cma
                                dac n
                                jmp start & 1
pattern  dap exit
                                szs 20
                                jmp chg
                                law 5077
                                dap chl          ,sets pattern for 4096 word memory
                                law 3077
                                dap ch2
                                dap ch3
                                lac q
                                dac m
                                szs 60
                                jmp pd
chl      sar s6
                                szs 50
                                jmp pc
                                szs 40
                                jmp pb
                                szs 30
                                jmp pa
                                jmp read          ,reads in new machine language tape
pa      xor m          ,pattern a
                                dac m
                                rar s1
                                xor m
                                rar s1
                                jmp out
pc      sza *          ,pattern c
                                jmp pa
                                sub one
                                jmp pa
pb      xor m          ,pattern b
                                rar s2
                                jmp out
pd      cli          ,pattern d
ch2     rcr s6
```

```
          sza *
          jmp pb
          sub one
ch3       dap m
          rcl s6
          xor m
          rar s2
          cma
out       spa *
          jmp c & 5
          law 1000
          dap isop
          dap isop2
exit     jmp
          cla
chg       jmp c - 4
          law 3037
          dap ch2
          dap ch3
          law 5037           ,sets pattern for 1024 word memory
          dap chl
          jmp chl - 4
one      l
q        0
m        0
n        777777
final    7600
read     rpb
          dio m
          lac m
          dap stop
          dap stop & 2
          and stop
          sad stop
stop     jmp
          rpb
          dio
          jmp read
jmp start end .
```