KEY ASSIGNMENT Vb

INPUT STATE

```
       -20       -10         0         10        20        30        40
        .         .          .          .         .         .         .
0000000000000000000M00Y000X0000111X0011110X0100110X0110010X1000000Y00000000000
                          ^
Input:UTMADD.INP          1                                               0
       Locate                Copy Item           Write dir           Simulate
   1 Y   Y  R  2           ?   ?  R   6      13 M   B  R 15           ?   ?  L 18
     0   A  L  1         7 0   A  L   8         ?   ?  L 13       19 A   1  L 21
     1   B  L  1           1   B  L   9      14 M   A  R 15          B   1  R 20
     ?   ?  L  1           ?   ?  R   7         ?   ?  L 14          ?   ?  L 19
   2 X   X  R  7         8 Y   Y  R 11       15 A   0  R 15       20 0   M  R 22
     A   0  R  3           ?   ?  L   8         B   1  R 15          1   M  R 23
     B   1  R  4         9 Y   Y  R 10          X   X  R 16          ?   ?  R 20
     ?   ?  R  2           ?   ?  L   9         ?   ?  R 15       21 0   M  R 22
   3 0   A  L  5        10 0   B  R 12       16 0   0  L 17          1   M  R 23
     1   B  R  6           1   B  R 12          1   1  L 17          ?   ?  L 21
     ?   ?  R  3           X   X  L 13          ?   ?  R 16       22 S   A  L  1
   4 0   A  R  6           ?   ?  R 10       17 A   0  L 17          ?   ?  R 22
     1   B  L  5        11 0   A  R 12          B   1  L 17       23 S   B  L  1
     ?   ?  R  4           1   A  R 12          0   S  L 18          ?   ?  R 23
   5 Y   Y  R  2           X   X  L 14          1   S  L 19
     ?   ?  L  5           ?   ?  R 11          ?   ?  L 17
   6 X   X  L  1        12 X   X  R   7      18 A   0  L 21
     Y   Y  R  0           ?   ?  R 12          B   0  R 20
0       -10        0         10        20        30        40        50
          .         .         .         .         .         .         .
000000000M1110Y11AXAAAABBBXAABBBBAXABAABBAXABBAABAXBA00000Y00000000000000000000
                                                           ^
Input:UTMADD.INP                                          HALT             3093
       Locate                Copy Item           Write dir           Simulate
   1 Y   Y  R  2           ?   ?  R   6      13 M   B  R 15           ?   ?  L 18
     0   A  L  1         7 0   A  L   8         ?   ?  L 13       19 A   1  L 21
     1   B  L  1           1   B  L   9      14 M   A  R 15          B   1  R 20
     ?   ?  L  1           ?   ?  R   7         ?   ?  L 14          ?   ?  L 19
   2 X   X  R  7         8 Y   Y  R 11       15 A   0  R 15       20 0   M  R 22
     A   0  R  3           ?   ?  L   8         B   1  R 15          1   M  R 23
     B   1  R  4         9 Y   Y  R 10          X   X  R 16          ?   ?  R 20
     ?   ?  R  2           ?   ?  L   9         ?   ?  R 15       21 0   M  R 22
   3 0   A  L  5        10 0   B  R 12       16 0   0  L 17          1   M  R 23
     1   B  R  6           1   B  R 12          1   1  L 17          ?   ?  L 21
     ?   ?  R  3           X   X  L 13          ?   ?  R 16       22 S   A  L  1
   4 0   A  R  6           ?   ?  R 10       17 A   0  L 17          ?   ?  R 22
     1   B  L  5        11 0   A  R 12          B   1  L 17       23 S   B  L  1
     ?   ?  R  4           1   A  R 12          0   S  L 18          ?   ?  R 23
   5 Y   Y  R  2           X   X  L 14          1   S  L 19
     ?   ?  L  5           ?   ?  R 11          ?   ?  L 17
   6 X   X  L  1        12 X   X  R   7      18 A   0  L 21
     Y   Y  R  0           ?   ?  R 12          B   0  R 20
<ESC>=Quit  HelpUTM    Step-by-step  Continuous  HiSpeed    EditTape
```

Vb) The essential difference between the Halting Problem (HP) and the Clocked Halting
Problem (CHP) is that in the latter case, the CHP does NOT involve issue of whether or
not any Turing Machine P with input tape i (P,i) will ULTIMATELY halt, simply whether
or for any (P,i) there exists a universal tester E that will answer y/n as to whether

or not it will halt within the bounds of t steps (some finite upper bound.) Recall
the Unsolavability of Halting Problem (Alternative Proof) handout. One can
construct
(without contradiction) a semi-tester E_a that halts iff (P,i)halts (for any (P,i))
For in the 'diagonal' case in which E_a is fed d_(E_a) (a description of itself)
then
this states E_a halts iff E_a halts (a tautology). The CHP is simply a more
restrictive
version of semitester E_a (i.e., falls within E_a's extension, or the set of all
(P,i)
such that (P,i) halt.) In other words, let S_Halt = {(P,i) | (P,i) halt}. Let
S_Halt_t = {(P,i)| (P,i) halt within t steps}. Clearly, S_Halt_t is a subset
of S_Halt, so there would exist a modified semitester (E_a + (a counter) .. call it
E_(a_t) whose output comprises the follwing cases:

case 1: All inputs from the set: S_Halt_t. Then E_(a_t) halts and outputs Y
case 2: All inputs from the set:
    S_Halt - S_Halt_t = {(P,i) | (P,i) halt in (t+1) steps or more}
Then E_(a_t) halts and outputs N
case 3: Any input from a set NoHalt = {(P,i)| (P,i) don't halt}
Then, (trivially) since any such input would continue past t steps, then,
as in case 2, E_(a_t) halts and outputs N