מעבדה מס' 6: מכונת מצבים

במשימה זאת יש לתכנן מערכת ספרתית סינכרונית המשחקת נגד המשתמש (בן אנוש) במשחק Tik-Tak (איקס עיגול). מטרת המשחק היא יצירת רצף של שלושה תאים באחד מששת הצירים (שלושה אופקיים ושלושה אנכיים) ושני אלכסונים. את המערכת הנ"ל יש לממש באמצעות מכונת מצבים. המכונה תקבל כקלט מספר שלם (integer) בין 0 לבין 8 המייצג מספר התא (איור 1) בו בחר השחקן המשחק נגד המכונה.

איור 1

0	1	2
3	4	5
6	7	8

פלט המכונה הוא גם מספר שלם- מספר התא בו בחרה המכונה. בנוסף המכונה תנהל אות המודיע על סטאטוס המשחק באופן הבא:

00– משחק שטרם הסתיים

01- משחק שהסתיים בניצחון הבן אדם

10- משחק בו ניצחה המכונה

11- משחק אשר הסתיים בתיקו

הנחות התכן:

- אסטרטגיה של המכונה היא הניצחון
- אין לעסוק בבדיקת תקינות הקלט, הן מבחינת ערך הנקלט (0-8) והן מבחינת סימון התא שכבר נבחר. מניחים השחקן הוא נבון ולא יזין ערכים שאינם תקינים.

אילוצי התכן:

- יש לממש מערכת הכוללת מספר המצבים מזערי ככל האפשר
- אין להיעזר בזיכרון נוסף לזיכרון של מכונת מצבים
- בגרסה הראשונה המכונה היא השחקן המתחיל
- בגרסה הסופית השחקן המתחיל יכול להיות גם בן אדם

להלן דוגמא של סביבת סימולציה המאפשרת הרצת סימולציה בצורה אינטראקטיבית:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use std.textio.all;
use ieee.std_logic_textio.all;
-----------------------------------------------------------------------------
-
entity tb_example is
end entity tb_example;
-----------------------------------------------------------------------------
-
architecture arc_tb_example of tb_example is
   signal clk      : std_logic              :='0';   -- [in]
   signal rst      : std_logic              :='0';   -- [in]
   signal player_o : integer range 0 to 15 := 0;    -- [in]
   signal en       : std_logic              :='0';   -- [in]
   signal player_x : integer range 0 to 15;         -- [out]
   signal done     : std_logic;                     -- [out]
   signal status   : std_logic_vector(1 downto 0);  -- [out] "00" normal game,
begin
   DUT : entity work.fsm
   port map (clk      => clk,      -- [in  std_logic]
             rst      => rst,      -- [in  std_logic]
             player_o => player_o, -- [in  integer range 0 to 15]
             en       => en,       -- [in  std_logic]
             player_x => player_x, -- [out integer range 0 to 15]
             done     => done,     -- [out std_logic]
             status   => status);  -- [out std_logic_vector(1 downto 0)]

   clk <= not clk after 10 ns;
   rst <= '1'     after 20 ns;

   process
      variable lr, lw : line;
      variable o_sel  : integer;
   begin
      wait until rst = '1';
      wait on player_x;
      wait until rising_edge(clk);
      l1 : loop
         wait until rising_edge(clk);
         write(lw, string'("x player selects ") & integer'image(player_x));
         -- writing x player (a machine) choice to command shell
         writeline(output, lw);
         exit l1 when (status /= "00");-- "00" normal game
         wait until rising_edge(clk);
         -- waiting for o player step from command shell and data collection
         readline(input, lr);
         read(lr, o_sel);
         player_o <= o_sel;
         en       <= '1';
         wait until rising_edge(clk);
         wait for 1 ns;
         en       <= '0';
         wait on player_x, done, status;
      end loop l1;
      write(lw, string'("end of game with status ") &
      integer'image(conv_integer(status)));
      writeline(output, lw);
      wait;
   end process ;
end architecture arc_tb_example;
```

קובץ זה זמין גם בhttp://www.abramovbenjamin.net/labs/tb_example.pdf.

דיאגרמה המתארת את כל המצבים במשחק: http://www.abramovbenjamin.net/labs/game.gif