מעבדה לVHDL – רמזור

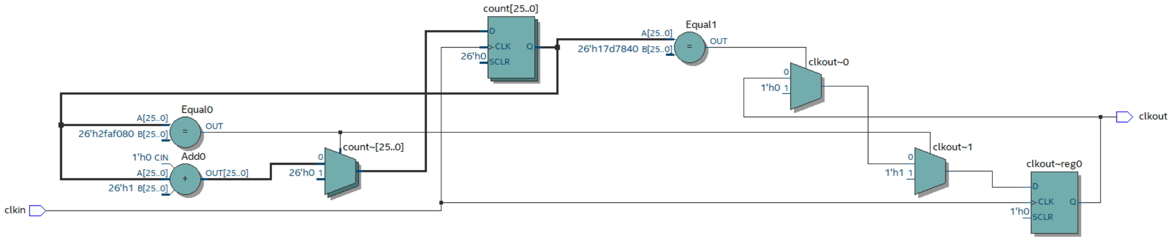
בתכנון עשינו שתי מכונות מצבים.
מכונת מצבים אחת תפעול הרמזור במצב יום\לילה והמכונת מצבים השנייה עבור תפעול מצב Emergency.
תכננו את המערכת כך שאנו עוצרים את המצב של המכונת מצבים שאחראית על מצבי עבודה יום\לילה כאשר אנחנו מפעילים את המצב Emergency.

בתכנון החלטנו שהמשתמש יכול לבחור בין האותות שעון שיכולים להיכנס למערכת.
אות שעון אחד בתדר 50[MHz] עם מחלק תדר ואות שעון שני שמחובר ללחצן.

התחלנו מלכתוב את התוכנית למחלק תדר:

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity Generator is
generic(N:  integer:= 50000000;
            DC: integer:= 25000000);
port(
clkin: in std_logic;
clkout: buffer std_logic:='1');
end;

architecture one of Generator is
signal count: integer range 0 to N := 0;
begin
process(clkin)
begin
if (clkin 'event and clkin = '1') then
      if (count = N) then
            count <= 0;
            clkout <= '1';
      elsif (count = DC) then
            count <= count + 1;
            clkout <= '0';
      else
            count <= count + 1;
      end if;
end if;
end process;
end;
```

:RTL

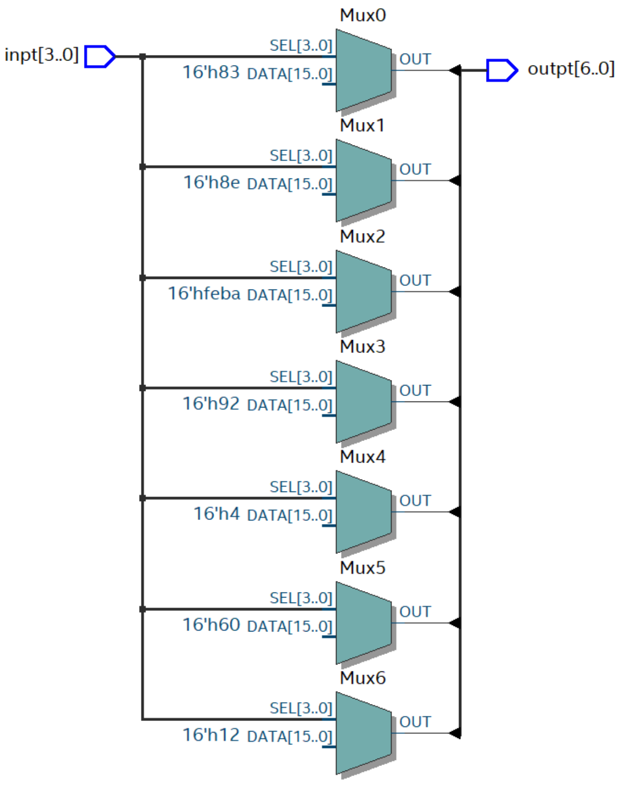


בנוסף רצינו להציג את הספירה של כמות הזמן שלוקח למכונת מצבים להתקדם.
לשם כך הצגנו את תוצאת המונה על תצוגת 7 המקטעים.

תוכנית למפענח תצוגה:

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Decoder7Segment is
port(
inpt:  in  std_logic_vector(3 downto 0);
outpt: out std_logic_vector(6 downto 0));
end;

architecture one of Decoder7Segment is
begin

with inpt select
outpt <= "1000000" when "0000", -- '0'
                  "1111001" when "0001", -- '1'
                  "0100100" when "0010", -- '2'
                  "0110000" when "0011", -- '3'
                  "0011001" when "0100", -- '4'
                  "0010010" when "0101", -- '5'
                  "0000010" when "0110", -- '6'
                  "1111000" when "0111", -- '7'
                  "0000000" when "1000", -- '8'
                  "0010000" when OTHERS; -- '9'
--                "" when "1010", -- 'A'
--                "" when "1011", -- 'B'
--                "" when "1100", -- 'C'
--                "" when "1101", -- 'D'
--                "" when "1110", -- 'E'
--                "" when "1111", -- 'F'  (OTHERS)

end;
```

RTL:

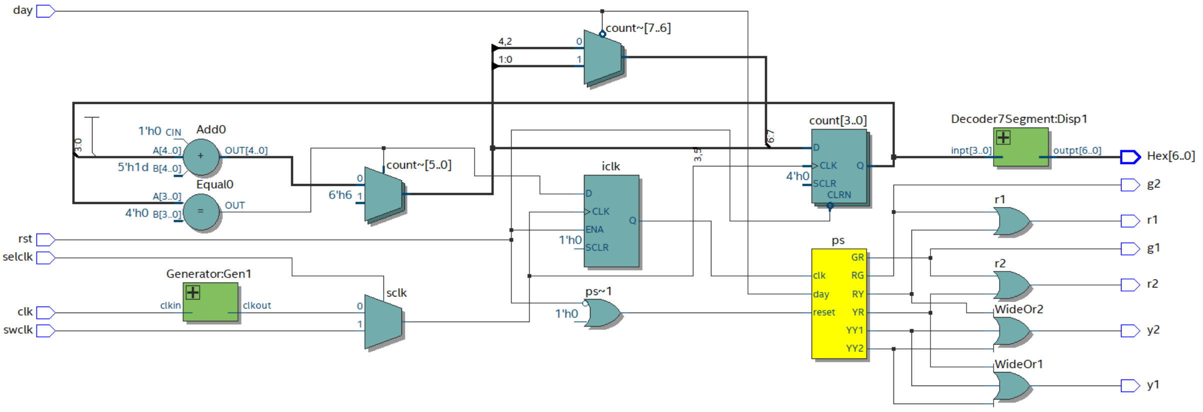


התוכנית עבור המכונת מצבים ששולטת על המצב יום\לילה של המערכת:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity Traffic_Light is
port(
swclk,selclk: in std_logic;
day,clk,rst: in std_logic;
y1,y2,r1,r2,g1,g2: out std_logic;
Hex: out std_logic_vector(6 downto 0));
end;

architecture one of Traffic_Light is

--/ Components \--
component Generator
generic(N:  integer:= 50000000;
              DC: integer:= 25000000);
port(clkin: in std_logic;
clkout: buffer std_logic:='1');
end component;

component Decoder7Segment
port(inpt:  in  std_logic_vector(3 downto 0);
outpt: out std_logic_vector(6 downto 0));
end component;
--/ Components \--

type state is (YY1,YR,GR,YY2,RY,RG,NN);
signal ns,ps: state;
signal clk1hz,iclk, sclk: std_logic;
signal count: std_logic_vector(3 downto 0);
begin

process (iclk,rst)
begin
if(rst = '0') then ps <= YY1;
elsif (iclk 'event and iclk = '1') then ps <= ns; end if;
end process;

process (ps)
begin
case ps is
when YY1 => if (day = '0') then ns <= NN; else ns <= YR;  end if;
when YR  => if (day = '0') then ns <= NN; else ns <= GR;  end if;
when GR  => if (day = '0') then ns <= NN; else ns <= YY2; end if;
when YY2 => if (day = '0') then ns <= NN; else ns <= RY;  end if;
when RY  => if (day = '0') then ns <= NN; else ns <= RG;  end if;
when RG  => if (day = '0') then ns <= NN; else ns <= YY1; end if;
when NN  => ns <= YY2;
end case;

case ps is
when YY1 => y1 <= '1'; y2 <= '1'; g1 <= '0'; g2 <= '0'; r1 <= '0'; r2 <=
'0';
when YR  => y1 <= '1'; y2 <= '0'; g1 <= '0'; g2 <= '0'; r1 <= '0'; r2 <=
'1';
when GR  => y1 <= '0'; y2 <= '0'; g1 <= '1'; g2 <= '0'; r1 <= '0'; r2 <=
'1';
```

```
when YY2 => y1 <= '1'; y2 <= '1'; g1 <= '0'; g2 <= '0'; r1 <= '0'; r2 <=
'0';
when RY  => y1 <= '0'; y2 <= '1'; g1 <= '0'; g2 <= '0'; r1 <= '1'; r2 <=
'0';
when RG  => y1 <= '0'; y2 <= '0'; g1 <= '0'; g2 <= '1'; r1 <= '1'; r2 <=
'0';
when NN  => y1 <= '0'; y2 <= '0'; g1 <= '0'; g2 <= '0'; r1 <= '0'; r2 <=
'0';
end case;

end process;

process(sclk,rst)
begin
if (rst = '0') then count <= "0000";
elsif (sclk 'event and sclk = '1') then
      if (day = '0') then -- Night
            if (count = "0000") then
                  count <= "0010";
                  iclk <= '1';
            else
                  count <= count - 1;
                  iclk <= '0';
            end if;
      else
            if (count = "0000") then
                  count <= "1000";
                  iclk <= '1';
            else
                  count <= count - 1;
                  iclk <= '0';
            end if;
      end if;
end if;
end process;
sclk <= swclk when (selclk = '1') else clk1hz;
-- define mux that select clk (clk = KEY {or} clk = 50[MHz])
Gen1:  Generator generic map (50000000,25000000) port map (clk,clk1hz);
Disp1: Decoder7Segment port map (count,Hex);
end;
```

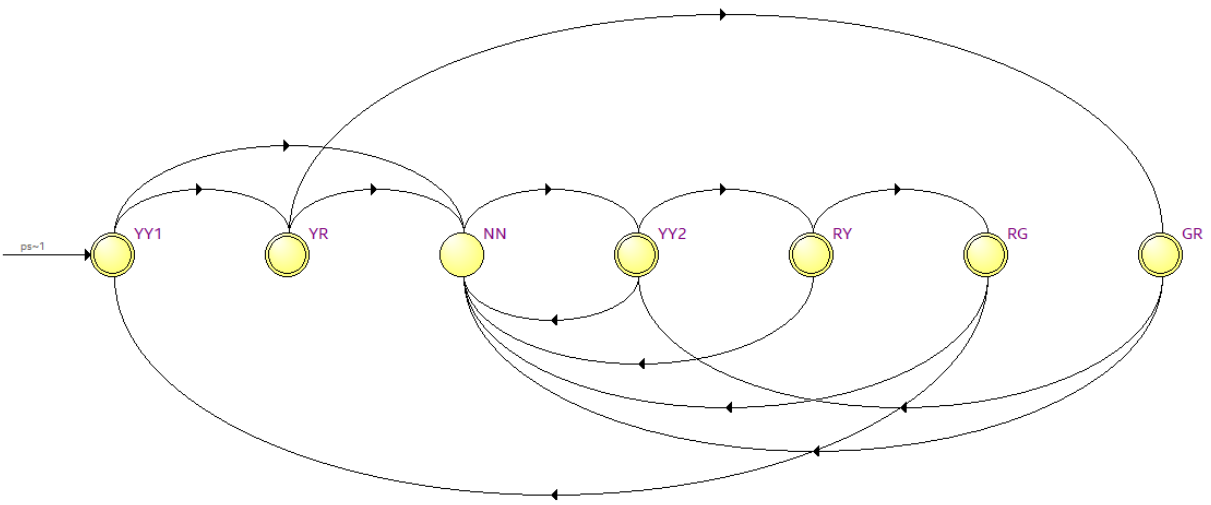
:RTL



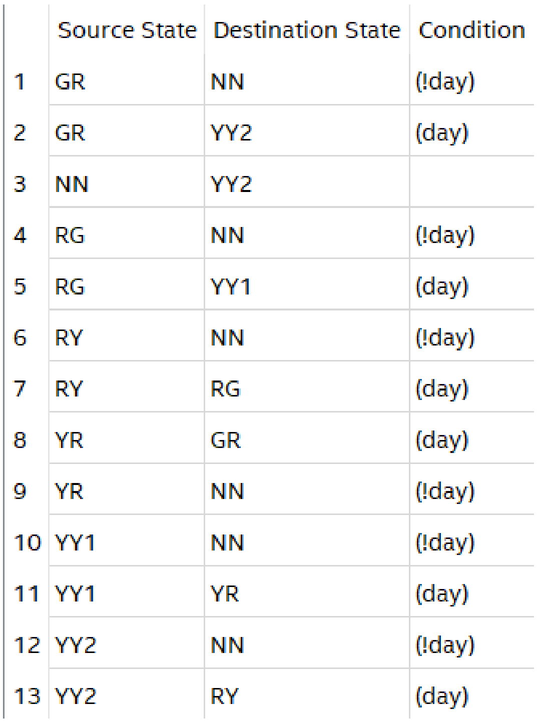
206912354

אריאל אוחיון

דיאגרמת מצבים:



טבלת מעברים:



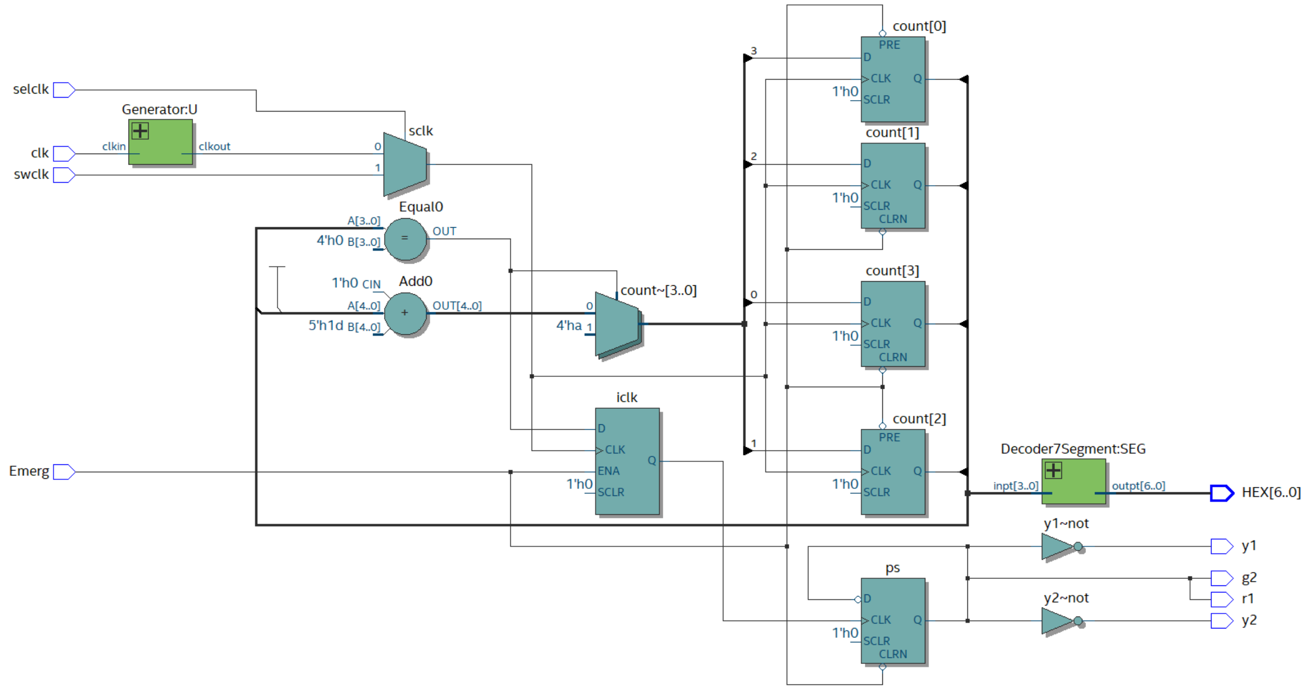
	Source State	Destination State	Condition
1	GR	NN	(!day)
2	GR	YY2	(day)
3	NN	YY2	
4	RG	NN	(!day)
5	RG	YY1	(day)
6	RY	NN	(!day)
7	RY	RG	(day)
8	YR	GR	(day)
9	YR	NN	(!day)
10	YY1	NN	(!day)
11	YY1	YR	(day)
12	YY2	NN	(!day)
13	YY2	RY	(day)

התוכנית עבור המכונת מצבים ששולטת על המצב Emergency של המערכת:

```
library ieee;
use ieee.std_Logic_1164.all;
use ieee.std_logic_unsigned.all;

entity EmergencySystem is
port(
Emerg,clk,swclk,selclk: in std_Logic;
y1,y2,r1,g2: out std_logic;
HEX: out std_logic_vector(6 downto 0));
end;

architecture one of EmergencySystem is
component Generator
generic(N:  integer:= 50000000;
              DC: integer:= 25000000);
port(clkin: in std_logic;
clkout: buffer std_logic:='1');
end component;

component Decoder7Segment
port(
inpt:  in  std_logic_vector(3 downto 0);
outpt: out std_logic_vector(6 downto 0));
end component;

type state is (YY1,RG);
signal ps,ns:state;
signal clk1hz,iclk,sclk: std_logic;
signal count: std_logic_vector(3 downto 0);
begin

process(Emerg,iclk)
begin
if (emerg = '0') then ps<=YY1;
elsif (iclk 'event and iclk = '1') then ps<=ns; end if;
end process;
process(ps)
begin
case ps is
when YY1 => ns <= RG;  y1 <= '1'; y2 <= '1'; r1 <= '0'; g2 <= '0';
when RG  => ns <= YY1; y1 <= '0'; y2 <= '0'; r1 <= '1'; g2 <= '1';
end case;
end process;
process(sclk)
begin
if (emerg = '0') then count <= "0101";
elsif (sclk 'event and sclk = '1') then
      if (count = "0000") then
            count <= "0101";
            iclk  <= '1';
      else
            count <= count - 1;
            iclk  <= '0';
      end if;
end if;
end process;
```

```
sclk <= swclk when (selclk='1') else clk1hz;
U: Generator generic map (50000000, 25000000) port map (clk,clk1hz);
SEG: Decoder7Segment port map (count,HEX);
end;
```

RTL:

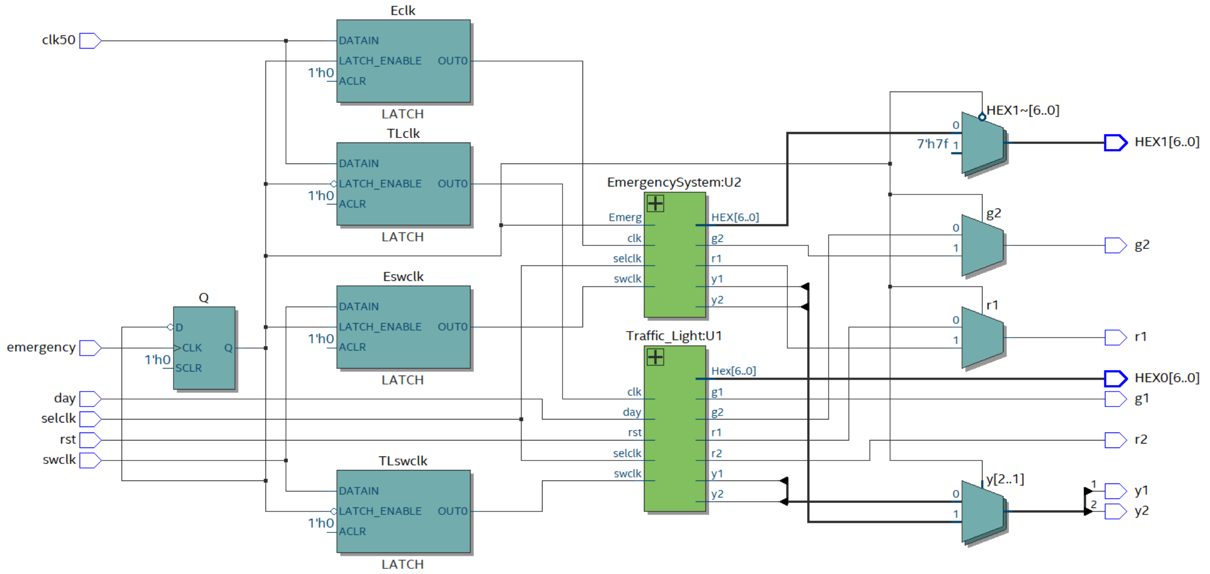


התוכנית הסופית שכוללת את כל הרכיבים שממלאים את הדרישה:

```
library ieee;
use ieee.std_logic_1164.all;

entity Full_Traffic_Light is
port(
day,rst,emergency,swclk,selclk,clk50:in std_logic;
HEX0,HEX1: out std_logic_vector(6 downto 0);
y1,y2,r1,r2,g1,g2: out std_logic);
end;

architecture one of Full_Traffic_Light is

component Traffic_Light
port(
swclk,selclk: in std_logic;
day,clk,rst: in std_logic;
y1,y2,r1,r2,g1,g2: out std_logic;
Hex: out std_logic_vector(6 downto 0));
end component;

component EmergencySystem
port(
Emerg,clk,swclk,selclk: in std_Logic;
y1,y2,r1,g2: out std_logic;
HEX: out std_logic_vector(6 downto 0));
```

```
end component;

signal TLswclk,TLclk: std_logic;
signal TLy1,TLy2,TLr1,TLg2: std_logic;
signal Eswclk,Eclk:   std_logic;
signal Ey1,Ey2,Er1,Eg2: std_logic;

signal sel: std_logic;
signal Q: std_logic:='0';
signal HEX: std_logic_vector(6 downto 0);

begin
U1: Traffic_Light   port map
(TLswclk,selclk,day,TLclk,rst,TLy1,TLy2,TLr1,r2,g1,TLg2,HEX0);
U2: EmergencySystem port map (Q,Eclk,Eswclk,selclk,Ey1,Ey2,Er1,Eg2,HEX);

process(emergency)
begin
if (emergency 'event and emergency = '1') then Q <= not Q; end if;
end process;
HEX1 <= (others => '1') when (Q = '0') else HEX;
process (Q)
begin
if (Q = '1') then -- Emergency Enabled
      Eclk<=clk50;
      Eswclk<=swclk;
      y1<=Ey1;
      y2<=Ey2;
      r1<=Er1;
      g2<=Eg2;
else
      TLclk<=clk50;
      TLswclk<=swclk;
      y1<=TLy1;
      y2<=TLy2;
      r1<=TLr1;
      g2<=TLg2;
end if;
end process;
end;
```

:RTL


Eclk
clk50
DATAIN
LATCH_ENABLE
OUT0
1'h0
ACLR
LATCH
TLclk
DATAIN
LATCH_ENABLE
OUT0
1'h0
ACLR
LATCH
EmergencySystem:U2
Emerg
clk
selclk
swclk
HEX[6..0]
g2
r1
y1
y2
HEX1~[6..0]
7'h7f
0
1
HEX1[6..0]
g2
g2
r1
r1
Q
emergency
D
CLK
Q
1'h0
SCLR
Eswclk
DATAIN
LATCH_ENABLE
OUT0
1'h0
ACLR
LATCH
Traffic_Light:U1
clk
day
rst
selclk
swclk
Hex[6..0]
g1
g2
r1
r2
y1
y2
HEX0[6..0]
g1
r2
day
selclk
rst
swclk
TLswclk
DATAIN
LATCH_ENABLE
OUT0
1'h0
ACLR
LATCH
y[2..1]
y1
y2