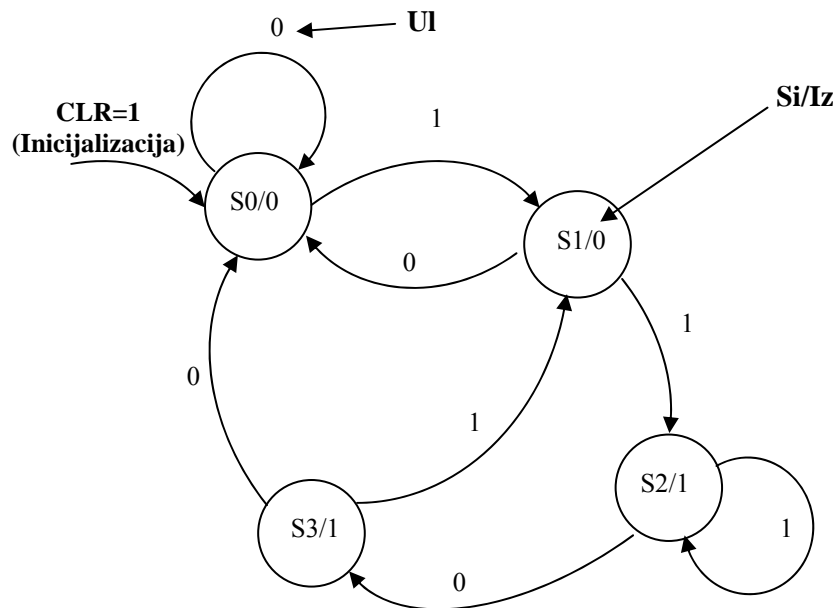


VIII DEO

REALIZACIJA SEKVENCIJALNIH MREŽA (ABEL)

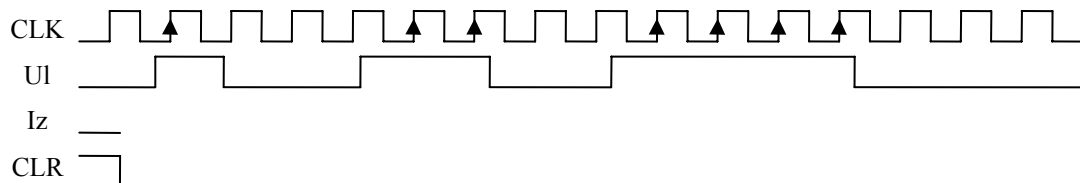
Zadatak 8.1.

a) Napisati *ABEL* program, kojim se u komponenti *PAL P16R4* realizuje sekvencijalna mreža data dijagramom stanja sa slike 8.1.1. Signalom *CLR=1* mreža se inicijalizuje u početno stanje *S0*.



Slika 8.1.1. Dijagram stanja sekvencijalne mreže

b) Generisati test vektore prema vremenskim dijagramima ulaznih signala, datih na slici 8.1.2. Na istom dijagramu doctrtati vremenski oblik izlaznog signala *Iz*.



Slika 8.1.2. Vremenski oblik ulaznih signala

REŠENJE:

```
module state_machine1
title 'zadatak 8.1'

U1 device 'P16R4';
"Definisanje koriscenih pinova komponente
clk, oe, clr      pin 1,11,19;
ul, iz           pin 2,12;
Q1, Q0          pin 14,15;

"Definisanje konstanti
H,L,X,Z,C      = 1,0, .X,..Z,..C.;

"Definisanje stanja mreze
S0 = 0; S1 = 1; S2 = 2; S3 = 3;
```

```

"Vektori
  Ulaz          = [clk, oe, clr, ul];
  Izlaz         = [iz, Q1, Q0];
  Stanje_mreze = [Q1, Q0];
equations
  ![Q1,Q0] := clr;

state_diagram Stanje_mreze
  State S0: iz = L;
    IF !ul THEN S0 ELSE S1;

  State S1: iz = L;
    IF !ul THEN S0 ELSE S2;

  State S2: iz = H;
    IF ul THEN S2 ELSE S3;

  State S3: iz = H;
    IF !ul THEN S0 ELSE S1;

test_vectors
  'Testiranje datog vremenskog dijagrama'
  (Ulaz          -> Izlaz)
" ([clk, oe, clr, ul]-> [iz, Q1, Q0])

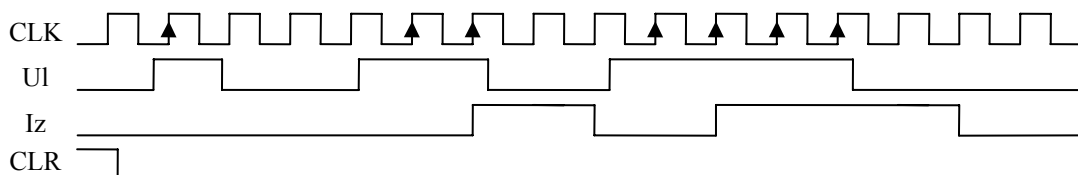
***** POČETAK VREMENSKOG DIJAGRAMA SA SLIKE 8.1.2 *****
[ C , L, H, 0 ] -> [ L, L, L ];
[ C , L, L, 1 ] -> [ L, L, H ];
[ C , L, L, 0 ] -> [ L, L, L ];
[ C , L, L, 1 ] -> [ L, L, H ];
[ C , L, L, 1 ] -> [ H, H, L ];
[ C , L, L, 0 ] -> [ H, H, H ];
[ C , L, L, 0 ] -> [ L, L, L ];
[ C , L, L, 1 ] -> [ L, L, H ];
[ C , L, L, 1 ] -> [ H, H, L ];
[ C , L, L, 1 ] -> [ H, H, L ];
[ C , L, L, 1 ] -> [ H, H, L ];
[ C , L, L, 0 ] -> [ H, H, H ];
[ C , L, L, 0 ] -> [ L, L, L ];
[ C , L, L, 0 ] -> [ L, L, L ];

***** KRAJ VREMENSKOG DIJAGRAMA SA SLIKE 8.1.2 *****
[ C , L, H, 1 ] -> [ L, L, L ];
[ C , L, L, 1 ] -> [ L, L, H ];
[ C , L, L, 1 ] -> [ H, H, L ];
[ C , L, L, 0 ] -> [ H, H, H ];
[ C , L, L, 1 ] -> [ L, L, H ];
[ C , H, L, 1 ] -> [ H, Z, Z ];

end state_machine1

```

Na slici 8.1.3. prikazani su vremenski dijagrami ulaznih i izlaznog signala I_z .

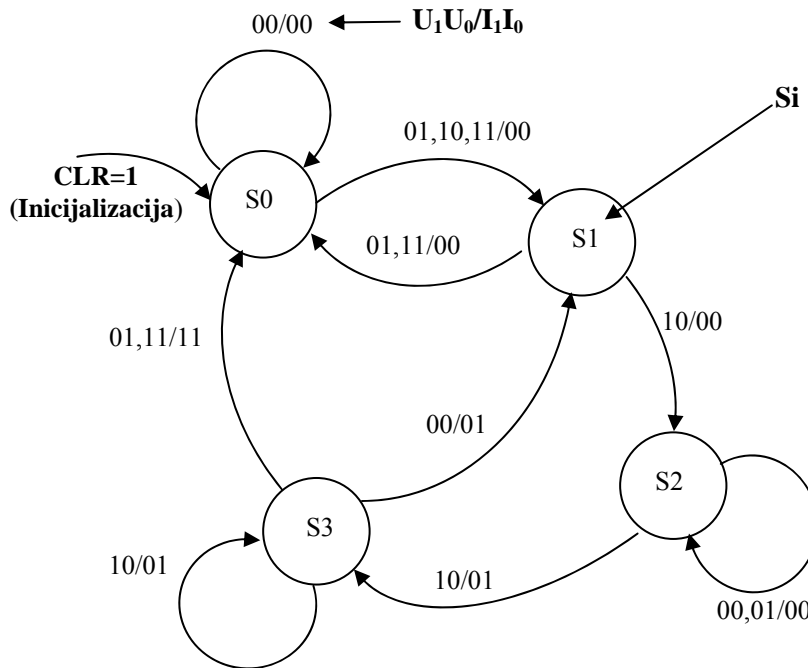


Slika 8.1.3. Vremenski dijagrami ulaznih i izlaznih signala sekvencijalne mreže

Zadatak 8.2.

a) Napisati *ABEL* program, kojim se u komponenti *PAL P16R4* realizuje sekvencijalna mreža data dijagramom stanja sa slike 8.2.1. Signalom *CLR=1* mreža se inicijalizuje u početno stanje *S0*.

b) Realizovati sekvencijalnu mrežu pomoću *STATE_DIAGRAM* i *WITH* konstrukcije.



Slika 8.2.1. Dijagram stanja sekvencijalne mreže

REŠENJE:

a) Jednačine pobude i izlaza, nakon minimizacije (*pogledati zadatak 7.6*), date su jednačinama:

Funkcija ulaza D flipflopa 1 biće:

$$D_1 = Q_1 Q_0 + Q_0 U_1 \bar{U}_0 + Q_1 U_1 \bar{U}_0 = Q_1^+$$

Funkcija ulaza D flipflopa 0 biće:

$$D_0 = Q_1 \bar{U}_1 \bar{U}_0 + Q_1 Q_0 \bar{U}_1 + \bar{Q}_1 \bar{Q}_0 U_0 + \bar{Q}_1 U_1 \bar{U}_0 = Q_0^+$$

Logička funkcija izlaza I_1 data je izrazom:

$$I_1 = Q_1 \bar{Q}_0 U_0$$

Logička funkcija izlaza I_0 data je izrazom:

$$I_0 = Q_1 \bar{Q}_0 + Q_1 U_1$$

Napomena: Treba obratiti pažnju, obzirom da je mreža *Mealy*-evog tipa, da nakon generisanja test vektora pri čemu je vrednost signala $CLK=C$. (taktni impuls oblika 010) izlazni signali su određeni vrednostima promenljivih stanja i stanja ulaza na kraju generisanja impulsa CLK , dakle po prelazu mreže u novo stanje (može biti i isto stanje) prema datom dijagramu stanja sekvencijalne mreže.

```
module state_machine2
```

```

title 'zadatak 8.2_a'

    U1 device 'P16R4';
"Definisanje koriscenih pinova komponente
    clk, oe, clr    pin  1,11,19;
    u1,u0,iz1,iz0  pin  2,3,12,13;
    Q1,Q0          pin  14,15;

"Definisanje konstanti
    H,L,X,Z,C = 1,0, .X.,.Z.,.C.;

"Definisanje stanja mreze
    S0 = ^b00; S1 = ^b01; S2 = ^b11; S3 = ^b10;

"Vektori
    Ulaz      = [clk, oe, clr, u1, u0];
    Izlaz     = [iz1, iz0, Q1, Q0];
    Stanje_mreze = [Q1, Q0];

equations
Q1      := !clr&(Q1&Q0#Q0&u1&!u0#Q1&u1&!u0);
Q0      := !clr&(Q1&!u1&!u0#Q1&Q0&!u1#!Q1&!Q0&u0#!Q1&u1&!u0);
iz1     = Q1&!Q0&u0;
iz0     = Q1&!Q0#Q1&u1;

test_vectors
    'Testiranje datog vremenskog dijagrama'
    (Ulaz      -> Izlaz)
"([clk, oe, clr, u1, u0] -> [iz1, iz0, Q1, Q0])
[ C , L , H , L , L ] -> [ L , L , L , L ];
[ C , L , L , H , H ] -> [ L , L , L , H ];
[ C , L , L , H , L ] -> [ L , H , H , H ];
[ C , L , L , H , L ] -> [ L , H , H , L ];
[ C , L , L , H , H ] -> [ L , L , L , L ];
[ L , L , L , H , L ] -> [ L , L , L , L ];
[ H , L , L , H , L ] -> [ L , L , L , H ];
[ C , L , L , H , L ] -> [ L , H , H , H ];
[ L , L , L , L , H ] -> [ L , L , H , H ];
[ C , L , L , L , H ] -> [ L , L , H , H ];
[ L , L , L , H , L ] -> [ L , H , H , H ];
[ C , L , L , H , L ] -> [ L , H , H , L ];
[ L , L , L , L , L ] -> [ L , H , H , L ];
[ L , L , L , H , H ] -> [ H , H , H , L ];
[ C , L , L , L , L ] -> [ X , X , L , H ];
[ C , L , L , H , H ] -> [ L , L , L , L ];
[ C , L , L , H , H ] -> [ L , L , L , H ];
[ C , L , H , L , L ] -> [ L , L , L , L ];
[ C , H , H , L , L ] -> [ L , L , Z , Z ];
[ C , L , H , L , L ] -> [ L , L , L , L ];

```

```
end state_machine2
```

b)

```

module state_machine2
title 'zadatak 8.2_b'

    U1 device 'P16R4';
"Definisanje koriscenih pinova komponente
    clk, oe, clr    pin  1,11,19;
    u1,u0,iz1,iz0  pin  2,3,12,13;
    Q1,Q0          pin  14,15;

"Definisanje konstanti

```

```

H,L,X,Z,C = 1,0, .X.,.Z.,.C.;

"Definisanje stanja mreze
S0 = ^b00; S1 = ^b01; S2 = ^b11; S3 = ^b10;

"Vektori
  Ulaz      = [clk, oe, clr, u1, u0];
  Izlaz     = [iz1, iz0, Q1, Q0];
  Stanje_mreze= [Q1, Q0];
equations
  ![Q1,Q0]  :=      clr;

state_diagram Stanje_mreze

  State S0: iz1=0;
           iz0=0;
           IF (u1#u0)      THEN S1 ELSE S0;

  State S1: iz1=0;
           iz0=0;
           IF (u1&!u0)    THEN S2 ELSE S0;

  State S2: iz1=0;
           IF (!u1)       THEN S2 ELSE iz0=0;
           ELSE S3       WITH iz0=1;

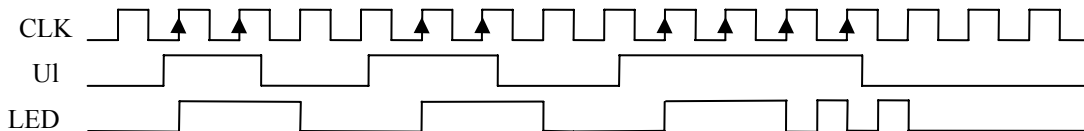
  State S3: iz1=1;
           IF (u1&!u0)    THEN S3 WITH iz1=0; iz0=1;
           ELSE IF (!u1&!u0) THEN S1 WITH iz1=0; iz0=1;
           ELSE S0       WITH iz1=1; iz0=1;

test_vectors
  'Testiranje datog vremenskog dijagrama'
  (Ulaz      ->  Izlaz)
" ([clk, oe, clr, u1, u0] -> [iz1, iz0, Q1, Q0])
[ C , L, H, L, L ] -> [ L, L, L, L ];
[ C , L, L, H, H ] -> [ L, L, L, H ];
[ C , L, L, H, L ] -> [ L, H, H, H ];
[ C , L, L, H, L ] -> [ L, H, H, L ];
[ C , L, L, H, H ] -> [ L, L, L, L ];
[ L , L, L, H, L ] -> [ L, L, L, L ];
[ H , L, L, H, L ] -> [ L, L, L, H ];
[ C , L, L, H, L ] -> [ L, H, H, H ];
[ L , L, L, L, H ] -> [ L, L, H, H ];
[ C , L, L, L, H ] -> [ L, L, H, H ];
[ L , L, L, H, L ] -> [ L, H, H, H ];
[ C , L, L, H, L ] -> [ L, H, H, L ];
[ L , L, L, L, L ] -> [ L, H, H, L ];
[ L , L, L, H, H ] -> [ H, H, H, L ];
[ C , L, L, L, L ] -> [ X, X, L, H ];
[ C , L, L, H, H ] -> [ L, L, L, L ];
[ C , L, L, H, H ] -> [ L, L, L, H ];
[ C , L, H, L, L ] -> [ L, L, L, L ];
[ C , H, H, L, L ] -> [ L, L, Z, Z ];
[ C , L, H, L, L ] -> [ L, L, L, L ];

```

Zadatak 8.3.

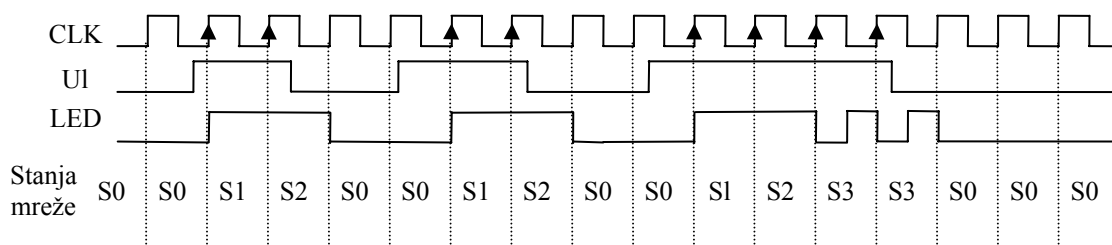
Napisati *ABEL* program, kojim se u komponenti *PAL P16R4* realizuje sekvencijalna mreža data vremenskim dijagramima na slici 8.3.1. Definirati stanja sekvencijalne mreže. Signalom $CLR=1$ mreža se inicijalizuje u početno stanje S_0 . Dijagram stanja realizovati pomoću *STATE_DIAGRAM* i *CASE_ENDCASE* konstrukcija *ABEL* jezika. Test vektore generisati prema datom vremenskom dijagramu signala sa slike 8.3.1.



Slika 8.3.1. Vremenski dijagrami ulaznih i izlaznih signala sekvencijalne mreže

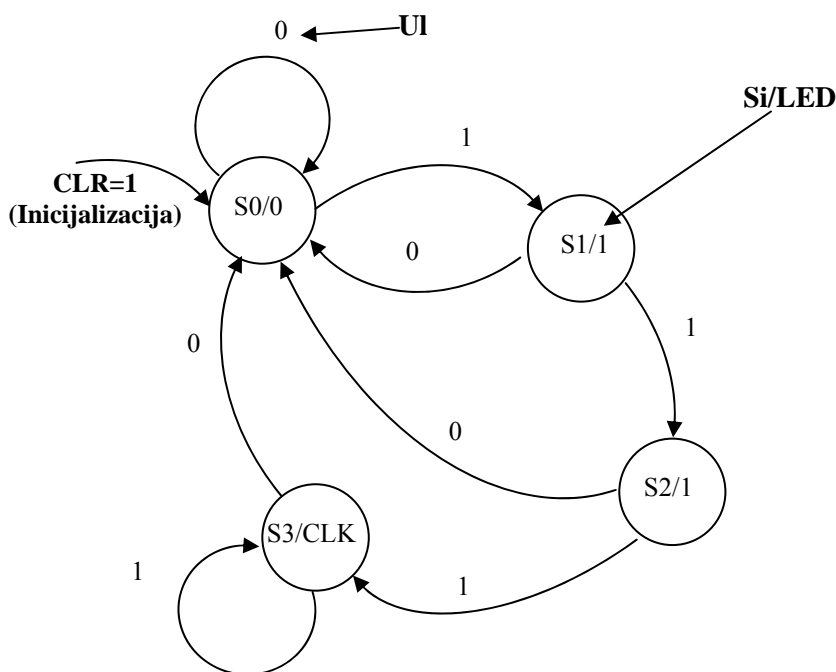
REŠENJE:

Na osnovu datih vremenskih dijagrama određujemo stanja sekvencijalne mreže (slika 8.3.2). Uočavamo da izlazi ne zavise od trenutne vrednosti ulaza UI , tako da je mreža *Moore*-ovog tipa.



Slika 8.3.2. Stanja mreže i vremenski dijagrami ulaznih i izlaznih signala

Na osnovu definisanih stanja određujemo dijagram stanja sekvencijalne mreže dat na slici 8.3.3.



Slika 8.3.3. Dijagram stanja sekvencijalne mreže

```

module state_machine3
title 'zadatak 8.3'

    U1 device 'P16R4';
"Definisanje koriscenih pinova komponente
    clk, oe, clr      pin  1,11,19;
    ul,clk_in, LED    pin  2,3,12;
    Q1,Q0             pin  14,15;

"Definisanje konstanti
    H,L,X,Z,C = 1,0, .X.,.Z.,.C.;

"Definisanje stanja mreze
    S0 = ^b00; S1 = ^b01; S2 = ^b10; S3 = ^b11;

"Vektori
    Ulaz          = [clk, clk_in, oe, clr, ul];
    Izlaz         = [LED, Q1, Q0];
    Stanje_mreze = [Q1, Q0];
equations
    ![Q1,Q0]      :=      clr;

state_diagram Stanje_mreze

    State S0: LED = L;
        CASE ul : S1;
            !ul : S0;
        ENDCASE;
    State S1: LED = H;
        CASE ul : S2;
            !ul : S0;
        ENDCASE;
    State S2: LED = H;
        CASE ul : S3;
            !ul : S0;
        ENDCASE;
    State S3: LED = clk_in;
        CASE ul : S3;
            !ul : S0;
        ENDCASE;

test_vectors
    'Testiranje datog vremenskog dijagrama'
    (Ulaz          -> Izlaz)
"([clk, clk_in, oe, clr, ul]      -> [LED, Q1, Q0])
    [ C ,   C ,   L ,   H ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   H ] -> [ H ,   L ,   H ];
    [ C ,   C ,   L ,   L ,   H ] -> [ H ,   H ,   L ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   H ] -> [ H ,   L ,   H ];
    [ C ,   C ,   L ,   L ,   H ] -> [ H ,   H ,   L ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   H ] -> [ H ,   L ,   H ];
    [ C ,   C ,   L ,   L ,   H ] -> [ H ,   H ,   L ];
    [ C ,   C ,   L ,   L ,   H ] -> [ C ,   H ,   H ];
    [ C ,   C ,   L ,   L ,   H ] -> [ C ,   H ,   H ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];
    [ C ,   C ,   L ,   L ,   L ] -> [ L ,   L ,   L ];

end state_machine3

```


Napomena: Potrebno je povezati pinove 1 i 3 komponente P16R4, da bi se generisali potrebni vremenski dijagrami. Zato je i formiran ulaz CLK_IN koji se koristi kao običan kombinacioni ulaz, obzirom da se CLK signal na pinu 1, vodi samo na ulaz CLK ivičnih D flipflova u datoj PAL komponenti.

Zadatak 8.4.

a) Napisati ABEL program, kojim se u komponenti PAL P16R4 realizuje brojač sa sekvencom brojanja 1→3→5→2→6→1→3... Po signalu RESET aktivnom na logičkoj jedinici brojač prelazi u stanje 4 nakon čega nastavlja brojanje 5→2→6→1→3→5...

b) Šta treba prepraviti u programu tako da sekvenca brojanja brojača bude 0→1→2→3→4→0→1..., i da nakon RESET signala brojač pređe u stanje 0.

REŠENJE:

a,b)

```

module state_machine4
title 'zadatak 8.4'

    U1 device 'P16R4';
"Definisanje korisćenih pinova komponente
    clk, oe, reset pin 1,11,19;
    Q2,Q1,Q0      pin 14,15,16;

"Definisanje konstanti
    H,L,X,Z,C = 1,0, .X,..Z,..C.;

"Definisanje stanja mreze

" Rešenje zadatka pod a)
    S0 = 1; S1 = 3; S2 = 5; S3 = 2; S4 = 6;
    INIT = 4; INIT_STATE = [1,0,0];

" Rešenje zadatka pod b)
    S0 = 0; S1 = 1; S2 = 2; S3 = 3; S4 = 4;
    INIT = 0; INIT_STATE = [0,0,0];

"Vektori
    Ulaz      = [clk, oe, reset];
    Stanje_mreze = [Q2, Q1, Q0];
    Izlaz     = [Q2, Q1, Q0];
equations
    ![Q2,Q1,Q0] := reset& !INIT_STATE;

state_diagram Stanje_mreze

    State INIT: goto S2; 'Pod b) State INIT: goto S1;'
    State S0: goto S1;
    State S1: goto S2;
    State S2: goto S3;
    State S3: goto S4;
    State S4: goto S0;

test_vectors
    'Testiranje datog vremenskog dijagrama'
    (Ulaz      -> Izlaz)
    " ([clk, oe, reset] -> [Q2, Q1, Q0])

```

```

"Pod b)
[ C , L , H ] -> 0;
[ C , L , L ] -> 1;
[ C , L , L ] -> 2;
[ C , L , L ] -> 3;
[ C , L , L ] -> 4;
[ C , L , L ] -> 0;
[ C , L , L ] -> 1;
[ C , L , L ] -> 2;
[ C , H , L ] -> Z;
[ C , L , L ] -> 4;

```

```

"Pod a)
[ C , L , H ] -> 4;
[ C , L , L ] -> 1;
[ C , L , L ] -> 3;
[ C , L , L ] -> 5;
[ C , L , L ] -> 2;
[ C , L , L ] -> 6;
[ C , L , L ] -> 1;
[ C , L , L ] -> 3;
[ C , H , L ] -> Z;
[ C , L , L ] -> 2;
end state_machine4

```

Zadatak 8.5.

Napisati *ABEL* program, kojim se u komponenti *PAL P16R4* realizuje 3-bitni brojač sa modovima rada ($I_1I_0=00$ -brojač na gore, $I_1I_0=01$ -brojač na dole, $I_1I_0=10$ u brojač se upisuje sadržaj A_1A_0 , $I_1I_0=11$ u brojač se upisuje sadržaj B_1B_0 . Po signalu *RESET* aktivnom na logičkoj jedinici brojač prelazi u stanje 000.

REŠENJE:

```

module state_machine5 flag '-r3'
title 'zadatak 8.5'

    U1 device 'P16R4';
"Definisanje koriscenih pinova komponente
    clk, oe, resetpin 1,11,19;
    I1,I0             pin 13,18;
    A1,A2,A0         pin 3,4,5;
    B1,B2,B0         pin 7,8,9;
    Q2,Q1,Q0         pin 15,16,17;

"Definisanje konstanti
    H,L,X,Z,C = 1,0, .X.,.Z.,.C.;
" Modovi rada brojaca
    Gore,Dole,Ucitaj_A,Ucitaj_B = 0,1,2,3;

"Definisanje stanja mreze
    S0 = ^b0000;  S4 = ^b0100;
    S1 = ^b0001;  S5 = ^b0101;
    S2 = ^b0010;  S6 = ^b0110;
    S3 = ^b0011;  S7 = ^b0111;

```

```

"Vektori
  Ulaz      = [clk, oe, reset, I1, I0, A, B];
  Stanje_mreze= [Q2,Q1,Q0];
  Izlaz     = [Q2,Q1,Q0];
  A         = [A2,A1,A0];
  B         = [B2,B1,B0];
  mod_brojaca = [I1,I0];

equations
  ![Q2,Q1,Q0] :=      reset& ![0,0,0];
  ![Q2,Q1,Q0] :=      (mod_brojaca == Ucitaj_A)& ![A2,A1,A0];
  ![Q2,Q1,Q0] :=      (mod_brojaca == Ucitaj_B)& ![B2,B1,B0];

state_diagram Stanje_mreze
  State S0:  CASE (mod_brojaca == Gore)      : S1;
             (mod_brojaca == Dole)         : S7;
             ENDCASE;
  State S1:  CASE (mod_brojaca == Gore)      : S2;
             (mod_brojaca == Dole)         : S0;
             ENDCASE;
  State S2:  CASE (mod_brojaca == Gore)      : S3;
             (mod_brojaca == Dole)         : S1;
             ENDCASE;
  State S3:  CASE (mod_brojaca == Gore)      : S4;
             (mod_brojaca == Dole)         : S2;
             ENDCASE;
  State S4:  CASE (mod_brojaca == Gore)      : S5;
             (mod_brojaca == Dole)         : S3;
             ENDCASE;
  State S5:  CASE (mod_brojaca == Gore)      : S6;
             (mod_brojaca == Dole)         : S4;
             ENDCASE;
  State S6:  CASE (mod_brojaca == Gore)      : S7;
             (mod_brojaca == Dole)         : S5;
             ENDCASE;
  State S7:  CASE (mod_brojaca == Gore)      : S0;
             (mod_brojaca == Dole)         : S6;
             ENDCASE;

test_vectors
  'Testiranje datog vremenskog dijagrama'
  (Ulaz      ->  Izlaz)
" ([clk, oe, reset, I1, I0, A, B]-> [Q2, Q1, Q0])
[ C , L ,  H , 0 , 0 , X , X] -> 0;
[ C , L ,  L , 0 , 0 , X , X] -> 1;
[ C , L ,  L , 0 , 0 , X , X] -> 2;
[ C , L ,  L , 0 , 0 , X , X] -> 3;
[ C , L ,  L , 0 , 0 , X , X] -> 4;
[ C , L ,  L , 0 , 0 , X , X] -> 5;
[ C , L ,  L , 0 , 0 , X , X] -> 6;
[ C , L ,  L , 0 , 0 , X , X] -> 7;
[ C , L ,  L , 0 , 0 , X , X] -> 0;
[ C , L ,  L , 0 , 0 , X , X] -> 1;
[ C , L ,  L , 0 , 1 , X , X] -> 0;
[ C , L ,  L , 0 , 1 , X , X] -> 7;
[ C , L ,  L , 0 , 1 , X , X] -> 6;
[ C , L ,  L , 0 , 1 , X , X] -> 5;
[ C , L ,  L , 0 , 1 , X , X] -> 4;
[ C , L ,  L , 0 , 1 , X , X] -> 3;
[ C , L ,  L , 0 , 1 , X , X] -> 2;
[ C , L ,  L , 0 , 1 , X , X] -> 1;
[ C , L ,  L , 0 , 1 , X , X] -> 0;
[ C , L ,  L , 0 , 1 , X , X] -> 7;
[ C , L ,  L , 1 , 0 , 6 , X] -> 6;

```

```
[ C , L , L , 0 , 1 , X , X] -> 5;  
[ C , L , L , 1 , 0 , 2 , X] -> 2;  
[ C , L , L , 1 , 1 , X , 7] -> 7;  
[ C , L , L , 0 , 0 , X , X] -> 0;  
[ C , L , L , 1 , 1 , X , 4] -> 4;  
[ C , L , H , 0 , 0 , X , X] -> 0;  
end state_machine5
```
