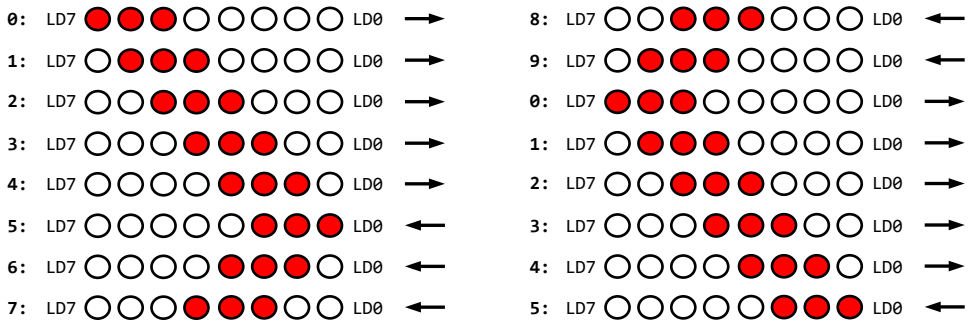
Szinkron sorrendi funkcionális elemek használata
Knight Rider futófény

A gyakorlaton a Knight Rider futófényt tervezzük meg kétféleképpen a laboron használt Logsys Spartan-3E FPGA kártyára, melyen 8 darab LED található. Kezdetben a bal szélső 3 LED világít és minden órajel felfutó él hatására a minta eggyel jobbra lép. A végállapotok elérésekor irányváltás történik.

F1. A futófényt ***egyirányú bináris számlálóra*** alapozva tervezzük meg, amelynek kimenetét egy kombinációs hálózat kódolja át a LED-eken megjelenő mintára.

F1.a A léptetés irányát is figyelembe véve hány állapota van a futófénynek? Az állapotok száma alapján milyen tulajdonságokkal (bitszám, vezérlő jelek) rendelkező bináris számlálót kell használnunk?

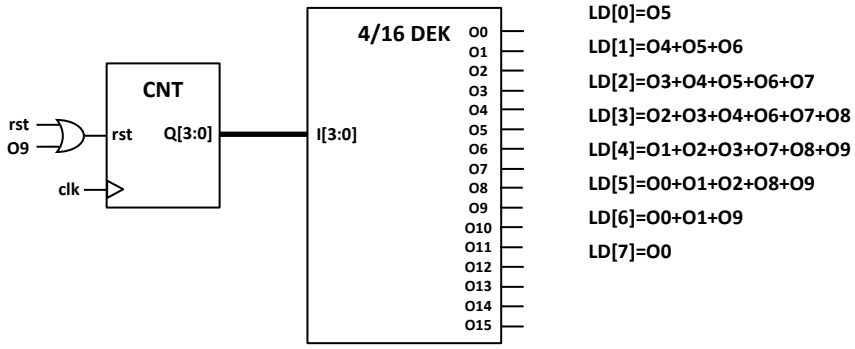


Az ábra alapján látható, hogy a léptetés irányát is figyelembe véve a futófénynek 10 állapota van. Ennek megfelelően ***4 bites felfele számláló bináris számlálóra*** van szükség. Mivel egy ilyen számlálónak 16 állapota van, ezért az állapotszám csökkentése szükséges a 9-es állapotban történő ***szinkron törléssel***.

F1.b Tegyük fel, hogy a megvalósításhoz diszkrét alkatrészeket (logikai kapuk, funkcionális egységek) használunk fel. Az átkódoló hálózathoz milyen alkatrészeket válasszunk, ha azok közül minél kevesebb félét szeretnénk felhasználni?

Az átkódoló hálózathoz szükséges egyedi kombinációs logika tervezését elkerülhetjük, ha szisztematikus tervezési módszerben gondolkodunk. Egy 4/16-os bináris dekóder segítségével dekódoljuk a számláló állapotait, minden állapothoz tartozik egy-egy dekóder kimenet (O0 – O15). A LED-eket vezérlő jeleket a megfelelő dekóder kimenetek VAGY kapcsolatával elő tudjuk állítani. Azokat a dekóder kimeneteket kell VAGY kapcsolatba hozni, amely állapotban az adott LED be van kapcsolva. A számlálót a dekóder O9 kimenete törli.

Tehát az átkódoló hálózat összesen kétféle alkatrésszel, egy 4/16 bináris dekóderrel és VAGY kapukkal megvalósítható.



LD[0]=O5
LD[1]=O4+O5+O6
LD[2]=O3+O4+O5+O6+O7
LD[3]=O2+O3+O4+O6+O7+O8
LD[4]=O1+O2+O3+O7+O8+O9
LD[5]=O0+O1+O2+O8+O9
LD[6]=O0+O1+O9
LD[7]=O0

F1.c Tegyük fel, hogy a megvalósításhoz az FPGA kártyát használjuk. Adjuk meg a futófény Verilog leírását! A magasszintű Verilog leírás esetén miben célszerű eltérni az előző ponthoz képest?

A Verilog leírásban nem kell külön-külön megadni a dekódert és annak kimeneteinek VAGY kapcsolatait, hanem egy case szerkezetben a LED-eken megjelenő minta egyszerűen megadható.

```
module knight_rider_cnt(
   input  wire       clk,
   input  wire       rst,
   output wire [7:0] ld
);

//4 bites bináris számláló. A futófénynek 10 állapota van,
//így a 9-es érték (1001) elérésekor töröljük a számlálót.
reg [3:0] cnt;

always @(posedge clk)
begin
   if (rst || (cnt == 4'd9))
      cnt <= 4'd0;
   else
      cnt <= cnt + 4'd1;
end

//A LED-eken megjelenő mintát előállító kombinációs logika.
reg [7:0] led_pattern;

always @(*)
begin
   case (cnt)
      4'd0   : led_pattern <= 8'b11100000;
      4'd1   : led_pattern <= 8'b01110000;
      4'd2   : led_pattern <= 8'b00111000;
      4'd3   : led_pattern <= 8'b00011100;
      4'd4   : led_pattern <= 8'b00001110;
      4'd5   : led_pattern <= 8'b00000111;
      4'd6   : led_pattern <= 8'b00001110;
      4'd7   : led_pattern <= 8'b00011100;
      4'd8   : led_pattern <= 8'b00111000;
      4'd9   : led_pattern <= 8'b01110000;
      default: led_pattern <= 8'b00000000;
   endcase
end

assign ld = led_pattern;

endmodule
```

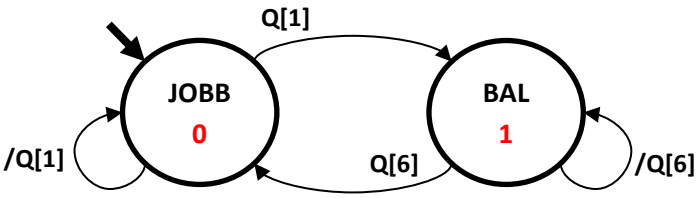
F2. A futófényt ***shiftregiszterre*** alapozva tervezzük meg, amelynek kimenete közvetlenül adja a LED-eken megjelenő mintát.

F2.a Milyen tulajdonságokkal (bitszám, vezérlő jelek) rendelkező shiftregisztert kell használnunk?

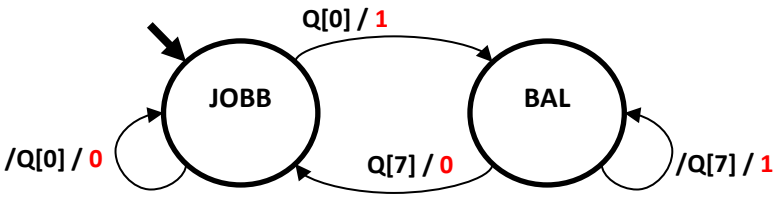
8 bites, tölthető (az 11100000 kezdőállapot beállításához), kétirányú shiftregiszter szükséges.

F2.b Gondoljuk át, hogy mikor szükséges irányt váltani és ez alapján rajzoljuk fel a léptetés irányát meghatározó állapotgép állapotgráfját és a futófény működésének idődiagramját!

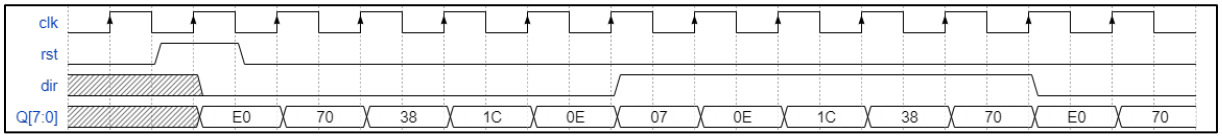
A futófény állapotait szemléltető ábra alapján látható, hogy az adott iránynak megfelelő végállapotban a léptetés iránya már meg kell forduljon, hogy a következő órajel felfutó élre már az új léptetési irány jusson érvényre. Tehát a végállapotok előtti állapotokat kell dekódolni az irányváltáshoz, amennyiben ***Moore típusú vezérlőt*** használunk (a Verilog kód ezt tartalmazza). Ha a shiftregiszter kimenete Q[7:0], akkor jobbra léptetés esetén a Q[1] bit, balra léptetés esetén pedig a Q[6] bit jelzi először ezeket az állapotokat.



Mealy típusú vezérlő esetén a végállapotokat kell dekódolni, melyeket jobbra léptetés esetén a Q[0] bit, balra léptetés esetén pedig a Q[7] bit jelez először.



A futófény működésének idődiagramja az alábbi ábrán látható. A tömörebb leírás végett a LED-ek állapota hexadecimálisan van megadva.



F2.c Tegyük fel, hogy a megvalósításhoz az FPGA kártyát használjuk. Adjuk meg a futófény Verilog leírását!

```
module knight_rider_shr(
   input  wire       clk,
   input  wire       rst,
   output wire [7:0] ld
);

//A LED-eken megjelenő mintát előállító 8 bites kétirányú shiftregiszter.
//dir=0: jobbra léptet, dir=1: balra léptet.
reg [7:0] shr;
reg       dir;

always @(posedge clk)
begin
   if (rst)
      shr <= 8'b11100000;
   else
      if (dir)
         shr <= {shr[6:0], 1'b0};
      else
         shr <= {1'b0, shr[7:1]};
end

assign ld = shr;

//A léptetési irányt meghatározó "állapotgép".
wire change_to_l = shr[1];
wire change_to_r = shr[6];

always @(posedge clk)
begin
   if (rst || change_to_r)
      dir <= 1'b0;
   else
      if (change_to_l)
         dir <= 1'b1;
end

endmodule
```