

A hálózat egy 8 bites Johnson-számláló (shift regiszter, melynek soros kimenete invertálva van visszavezetve a soros bemenetre). A számláló a BTN2 gombbal vihető alapállapotba, az alapállapot 8'b0000_0000. A hálózat órajele az FPGA panel 100 MHz-es órajele. A számláló léptetését egy másodpercenként érkező, 1 órajel szélességű engedélyezőjel végzi. Ezt a jelet is a hálózat állítja elő. A számláló kimeneteinek állapotát a 8 LED jelzi. A számláló az SW0 kapcsoló 1 állapotában számol, 0 állapotában „pause” módban van.

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    input sw0,
    output[7:0] led
);

    /* 1 Hz-es engedelyezojel */
    reg[26:0] q;
    wire en=(q==99_999_999);
    always@(posedge clk)
        q<=(rst|en)?0:(q+1);

    /* szamlalo */
    reg[7:0] johnson;
    always@(posedge clk)
        if(rst)
            johnson<=0;
        else if(en&sw0)
            johnson<={~johnson[0],johnson[7:1]};

    /* LED-ek meghajtasa */
    assign led=johnson;

endmodule
```

A hálózat egy 8 bites, bináris számlálót valósít meg, amely az SW1 kapcsoló 0 állásában a páros, 1 állásában a páratlan számokon lépdel végig. A generált számokat a LED-eken kell megjeleníteni, a számolást pedig a 100 MHz-es órajelből generált, másodperc frekvenciájú engedélyezőjel ütemezi. Az SW0 kapcsoló 1 állása a rendszert alapállapotba viszi (az SW1 kapcsoló állásától függően 0 vagy 1).

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    input sw1,
    output[7:0] led
);

    /* 1 Hz-es engedelyezojel */
    reg[26:0] q;
    wire en=(q==99_999_999);
    always@(posedge clk)
        q<=(rst|en)?0:(q+1);

    /* szamlalo */
    reg[7:0] cntr;
    always@(posedge clk)
        if(rst)
            cntr<={7'b0,sw1};
        else if(en)
            cntr<={cntr[7:1]+1,sw1};

    /* LED-ek meghajtasa */
    assign led=cntr;

endmodule
```

A hálózat egy futófényt („knight rider”) valósít meg. A BTN0 gomb hatására (reset jel) csak a jobb oldali LDO LED világít, majd a gomb elengedésekor a fény másodperces ütemezéssel balra shiftelődik. A ledsor két végére érve a futási irány megfordul. A hálózat órajele az FPGA panel 100 MHz-es órajele. A shiftelést egy másodpercenként érkező, 1 órajel szélességű engedélyezőjel ütemezi. Ezt a jelet is a hálózat állítja elő. A generálandó minta tehát:

0000_0001 → 0000_0010 → 0000_0100 → 0000_1000 → 0001_0000 → 0010_0000 → 0100_0000 →
1000_0000 → 0100_0000 → 0010_0000 → 0001_0000 → 0000_1000 → 0000_0100 → 0000_0010 →
0000_0001 → 0000_0010 →...

Ügyeljen rá, hogy minden állapot ugyanennyi ideig tartson (egy másodperc).

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    output[7:0] led
);

    /* 1 Hz-es engedelyezojel */
    reg[26:0] q;
    wire en=(q==99_999_999);
    always@(posedge clk)
        q<=(rst|en)?0:(q+1);

    reg[7:0] shr;

    /* iranyvaltas */
    reg dir;
    always@(posedge clk)
        if(rst|(shr==8'b00000001))
            dir<=0;
        else if(shr==8'b10000000)
            dir<=1;

    /* leptetes */
    always@(posedge clk)
        if(rst)
            shr<=8'b00000001;
        else if(en)
            shr<=dir?{1'b0,shr[7:1]}:{shr[6:0],1'b0};

    /* LED-ek meghajtasa */
    assign led=shr;

endmodule
```

Másik megoldás:

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    output reg[7:0] led
);

    /* 1 Hz-es engedelyezojel */
    reg[26:0] q;
    wire en=(q==99_999_999);
    always@(posedge clk)
        q<=(rst|en)?0:(q+1);

    /* szamlalo az utemezeshez */
    reg[3:0] cntr;
    always@(posedge clk)
        if(rst)
            cntr<=0;
        else if(en)
            cntr<=(cntr==13)?0:(cntr+1);

    /* multiplexer */
    always@(*)
        case(cntr)
            0:    led<=8'b00000001;
            1:    led<=8'b00000010;
            2:    led<=8'b00000100;
            3:    led<=8'b00001000;
            4:    led<=8'b00010000;
            5:    led<=8'b00100000;
            6:    led<=8'b01000000;
            7:    led<=8'b10000000;
            8:    led<=8'b01000000;
            9:    led<=8'b00100000;
            10:   led<=8'b00010000;
            11:   led<=8'b00001000;
            12:   led<=8'b00000100;
            13:   led<=8'b00000010;
            default: led<=8'bxxxxxxxx;
        endcase

endmodule
```

A hálózat adott ütemben villogó 8 darab LED-et valósít meg, amelynek az ütemezését a SW1 és SW0 kapcsolók állása határozza meg. A LED-ek {SW1, SW0}==0 esetben 0,25 mp-enként, {SW1, SW0}==1 esetben 0,5 mp-enként, {SW1, SW0}==2 esetben 1 mp-enként, {SW1, SW0}==3 esetben pedig 2 mp-enként váltanak állapotot (mind a 8 LED egyszerre). A hálózat órajele az FPGA panel 100 MHz-es órajele.

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    input[1:0] sw,
    output reg[7:0] led
);

    /* frekvenciaavlaszo multiplexer */
    reg[27:0] div;
    always@(*)
        case(sw)
            0: div<= 24_999_999; //4 Hz
            1: div<= 49_999_999; //2 Hz
            2: div<= 99_999_999; //1 Hz
            3: div<=199_999_999; //0.5 Hz
        endcase

    /* engedelyezojel */
    reg[27:0] q;
    wire en=(q==div);
    always@(posedge clk)
        q<=(rst|(q>=div))?0:(q+1);

    /* LED-ek invertalasa */
    always@(posedge clk)
        if(rst)
            led<=0;
        else if(en)
            led<=~led;

endmodule
```

A hálózat bekapcsolás után egy 0 karaktert jelenít meg a jobb oldali hétszegmenses kijelzőn, majd ezt követően másodperces időzítéssel a 0 karakter jobbról balra halad a hétszegmenses kijelzőkön. A bal oldali kijelzőt elérve a karakter ismét jobbról lép be. A hálózat órajele az FPGA panel 100 MHz-es órajele.

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    output reg[3:0] an,
    output[7:0] seg
);

    /* 1 Hz-es engedelyezojel */
    reg[26:0] q;
    wire en=(q==99_999_999);
    always@(posedge clk)
        q<=(rst|en)?0:(q+1);

    /* anodjel */
    always@(posedge clk)
        if(rst)
            an<=4'b1110;
        else if(en)
            an<={an[2:0],an[3]};

    /* a kijelzett ertek konstans 0 */
    assign seg=8'b11000000;

endmodule
```

A hálózat egy, az SW0 kapcsolóval változtatható számlálási irányú, 0-9 értéktartományú másodpercszámláló értékét mutatja a jobb oldali hétszegmenses kijelzőn, az LD0 LED egy paritásbitet jelenít meg, az LD1 LED pedig a szám páros jellegét azonosítja. A hálózat órajele az FPGA panel 100 MHz-es órajele. A számlálást egy másodpercenként érkező, 1 órajel szélességű engedélyezőjel uturezi. Ezt a jelet is a hálózat állítja elő.

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    output[3:0] an,
    output reg[7:0] seg,
    input sw0,
    output[7:0] led
);

    /* 1 Hz-es engedelyezojel */
    reg[26:0] q;
    wire en=(q==99_999_999);
    always@(posedge clk)
        q<=(rst|en)?0:(q+1);

    /* szamlalo */
    reg[3:0] cntr;
    always@(posedge clk)
        if(rst)
            cntr<=0;
        else if(en&~sw0)
            cntr<=(cntr==9)?0:(cntr+1);
        else if(en&sw0)
            cntr<=(cntr==0)?9:(cntr-1);

    /* csak a jobb oldali kijelzot használjuk */
    assign an=4'b1110;

    /* szegmensdekoder */
    always@(*)
        case(cntr)
            4'd0: seg<=8'b11000000;
            4'd1: seg<=8'b11111001;
            4'd2: seg<=8'b10100100;
            4'd3: seg<=8'b10110000;
            4'd4: seg<=8'b10011001;
            4'd5: seg<=8'b10010010;
            4'd6: seg<=8'b10000010;
            4'd7: seg<=8'b11111000;
            4'd8: seg<=8'b10000000;
            4'd9: seg<=8'b10010000;
            default: seg<=8'bxxxxxxxx;
        endcase

    /* LED-ek meghajtasa: parossag es paros paritasbit */
    assign led={6'd0, ~cntr[0], ^cntr};

endmodule
```

Készítsen 16 bites bináris számlálót, melynek értékét hexadecimális formában jelenítse meg a 4 darab hétszegmenses kijelzőn. A BTN0 (reset jel) gomb hatására a számláló a 0 értéket veszi fel, egyébként fél másodpercenként számol, az aktuális érték megjelenik a hétszegmenses kijelzőn. A hálózat órajele az FPGA panel 100 MHz-es órajele.

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    output reg[3:0] an,
    output reg[7:0] seg
);

    /* 2 Hz-es engedelyezojel a szamlalohoz */
    reg[25:0] q1;
    wire cntr_en=(q1==49_999_999);
    always@(posedge clk)
        q1<=(rst|cntr_en)?0:(q1+1);

    /* 1 kHz-es engedelyezojel a digitek leptetesehez */
    reg[16:0] q2;
    wire an_en=(q2==99_999);
    always@(posedge clk)
        q2<=(rst|an_en)?0:(q2+1);

    /* szamlalo */
    reg[15:0] cntr;
    always@(posedge clk)
        if(rst)
            cntr<=0;
        else if(cntr_en)
            cntr<=cntr+1;

    /* anodjel */
    always@(posedge clk)
        if(rst)
            an<=4'b1110;
        else if(an_en)
            an<={an[2:0],an[3]};

    /* multiplexer a digit kivalasztasahoz */
    reg[3:0] dmux;
    always@(*)
        case(an)
            4'b1110: dmux<=cntr[3:0];
            4'b1101: dmux<=cntr[7:4];
            4'b1011: dmux<=cntr[11:8];
            4'b0111: dmux<=cntr[15:12];
            default: dmux<=4'bxxxx;
        endcase

    /* szegmensdekoder */
    always@(*)
        case(dmux)
            4'h0: seg<=8'b11000000;
            4'h1: seg<=8'b11111001;
            4'h2: seg<=8'b10100100;
            4'h3: seg<=8'b10110000;
            4'h4: seg<=8'b10011001;
            4'h5: seg<=8'b10010010;
            4'h6: seg<=8'b10000010;
            4'h7: seg<=8'b11111000;
            4'h8: seg<=8'b10000000;
            4'h9: seg<=8'b10010000;
            4'hA: seg<=8'b10001000;
            4'hB: seg<=8'b10000011;
            4'hC: seg<=8'b11000110;
            4'hD: seg<=8'b10100001;
            4'hE: seg<=8'b10000110;
            4'hF: seg<=8'b10001110;
        endcase

endmodule
```