

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 25 másodperces visszaszámláló időzítőt valósít meg, amely a hátralévő időt BCD formátumban jelzi ki az FPGA panel LED-jein. A reset gomb megnyomásakor a kijelzett érték 25 (azaz 8'b00100101), majd a gomb elengedésekor elkezdődik a másodpercenkénti visszaszámlálás. Az idő letelte után a kijelzett érték nulla marad, az időzítő csak a reset gomb ismételt megnyomásával indítható újra. A hálózat órajele az FPGA panel 100 MHz-es órajele.

```
`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);

    /* szamlalo */
    reg[1:0] d1;
    reg[3:0] d0;
    wire done=~|{d1,d0};

    always@(posedge clk)
        if(rst)
            d0<=4'b0101;
        else if(en&~done)
            d0<=(d0==0)?9:(d0-1);

    always@(posedge clk)
        if(rst)
            d1<=2'b10;
        else if(en&(d0==0)&~done)
            d1<=d1-1;

    /* LED-ek meghajtasa */
    assign led={2'b00, d1, d0};

endmodule
```

A hálózat egy futófényt („knight rider”) valósít meg. A BTN0 gomb hatására (reset jel) csak a jobb oldali LD0 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 ugyannyi 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'bxxxxxxx;
        endcase
endmodule
```

A hálózat egy programozható stroboszkópot valósít meg: az FPGA panel nyolc LED-je egyszerre villog a kapcsolósor alsó három bitjén beállított frekvenciával. A villogás kitöltési tényezője 50%. A reset gomb megnyomásakor az összes LED sötét. A hálózat órajele az FPGA panel 100 MHz-es órajele.

```
`timescale 1ns / 1ps

module toplevel(
    input clk, //100 MHz
    input rst,
    input[2:0] sw,
    output reg[7:0] led
);

    /* frekvenciaavlaszo multiplexer */
    reg[26:0] div;
    always@(*)
        case(sw)
            0: div<=70_000_000; //0 Hz (ezt sosem eri el a szamlalo --> nem villog)
            1: div<=49_999_999; //1 Hz
            2: div<=24_999_999; //2 Hz
            3: div<=16_666_666; //3 Hz
            4: div<=12_499_999; //4 Hz
            5: div<= 9_999_999; //5 Hz
            6: div<= 8_333_332; //6 Hz
            7: div<= 7_142_856; //7 Hz
        endcase

    /* engedelyezojel */
    reg[25:0] 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 reset gomb megnyomásakor a jobb oldali hétszegmenses kijelzőn egy A karakter látható, majd a gomb felengedését követően másodpercenként eggyel balra lép. A kijelzősor széleit elérve a lépkedési irány megfordul. A hálózat órajele az FPGA panel 100 MHz-es órajele.

xxxA → xxAx → xAxx → Axxx → xAxx → xxAx → xxxA → ...

```
`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);

    /* iranyvaltas */
    reg dir;
    always@(posedge clk)
        if(rst|(an==4'b1110))
            dir<=0;
        else if(an==4'b0111)
            dir<=1;

    /* leptetes */
    always@(posedge clk)
        if(rst)
            an<=4'b1110;
        else if(en)
            an<=dir?{an[0],an[3:1]}:{an[2:0],an[3]};

    /* A karakter */
    assign seg=8'b10001000;

endmodule
```


Másik megoldás:

```
`timescale 1ns / 1ps

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

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

    /* szamlalo az utemezeshez */
    reg[2:0] cntr;
    always@(posedge clk)
        if(rst)
            cntr<=0;
        else if(en)
            cntr<=(cntr==5)?0:(cntr+1);

    /* multiplexer */
    always@(*)
        case(cntr)
            0:      an<=4'b1110;
            1:      an<=4'b1101;
            2:      an<=4'b1011;
            3:      an<=4'b0111;
            4:      an<=4'b1011;
            5:      an<=4'b1101;
            default: an<=4'bxxxx;
        endcase

    /* A karakter */
    assign seg=8'b10001000;

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 őrzi. 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
```

A hálózat egy hexadecimális, 0–F értéktartományú számlálót valósít meg az FPGA panel hétszegmenses kijelzőin. A számláló értékét a jobb oldali kijelző jeleníti meg, a többi kijelző sötét. A reset gomb megnyomásakor a kijelzett érték a kapcsolósor alsó négy bitjén beállított szám, majd a gomb elengedésekor ez az érték másodpercenként eggyel növekszik (túlszorduláskor F után 0 következik). A hálózat órajele az FPGA panel 100 MHz-es órajele.

```
`timescale 1ns / 1ps

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

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

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

    /* szegmensdekoder */
    always@(*)
        case(cntr)
            0: seg<=8'b11000000;
            1: seg<=8'b11111001;
            2: seg<=8'b10100100;
            3: seg<=8'b10110000;
            4: seg<=8'b10011001;
            5: seg<=8'b10010010;
            6: seg<=8'b10000010;
            7: seg<=8'b11111000;
            8: seg<=8'b10000000;
            9: seg<=8'b10010000;
            10: seg<=8'b10001000;
            11: seg<=8'b10000011;
            12: seg<=8'b11000110;
            13: seg<=8'b10100001;
            14: seg<=8'b10000110;
            15: seg<=8'b10001110;
        endcase

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
```

