Tema 2 – Transmisie paralela seriala.

Neciu Laurentiu Florin
Digori Gheorghe
331CC

Protocol

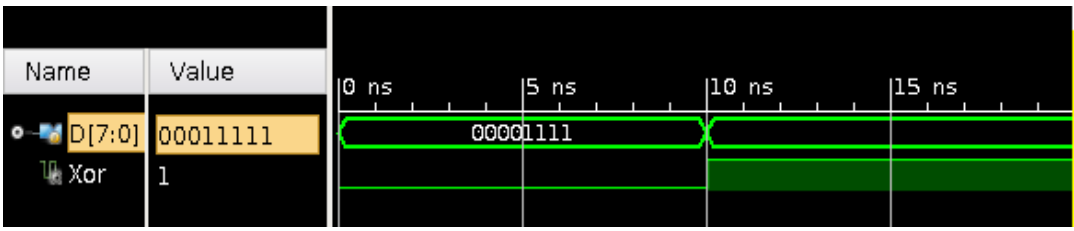
Transmitatorul prezinta date de intrare pe 8 biti, un bit de reset, de start si un ceas. Datele sunt scoase ca semnale de iesire pe un bit. Ele sunt precedate de un bit de start (0) si terminate cu unu sau doi biti de stop (1). Cand transmitatorul este inactiv, transmite 1 incontinuu. Semnalul de reset este sincron pe frontul crescator al ceasului. Bitii sunt transmisi de la biti de rang mai mic la rang mai mare.

Module folosie

- H1

Circuit pentru determinarea bitului de paritate.
Semnal de intrare: datele D pe 8 biti.
Semnal de iesire: suma modulo 2 dintre toti bitii.
Out = D[0] ^ D[1] ^ D[2] ^ D[3] ^ D[4] ^ D[5] ^ D[6] ^ D[7]

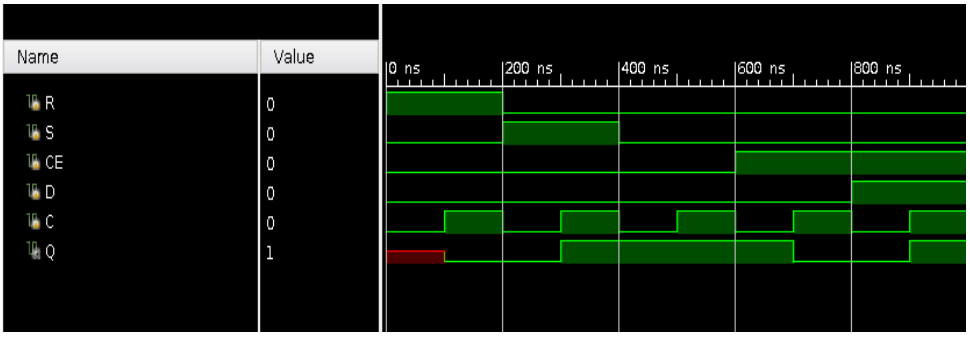
```
H1 uut(D, Xor);

initial begin
D = 8'b00001111;
#10;
D = 8'b00011111;
#10;

end
```



- Doua module FDRSE

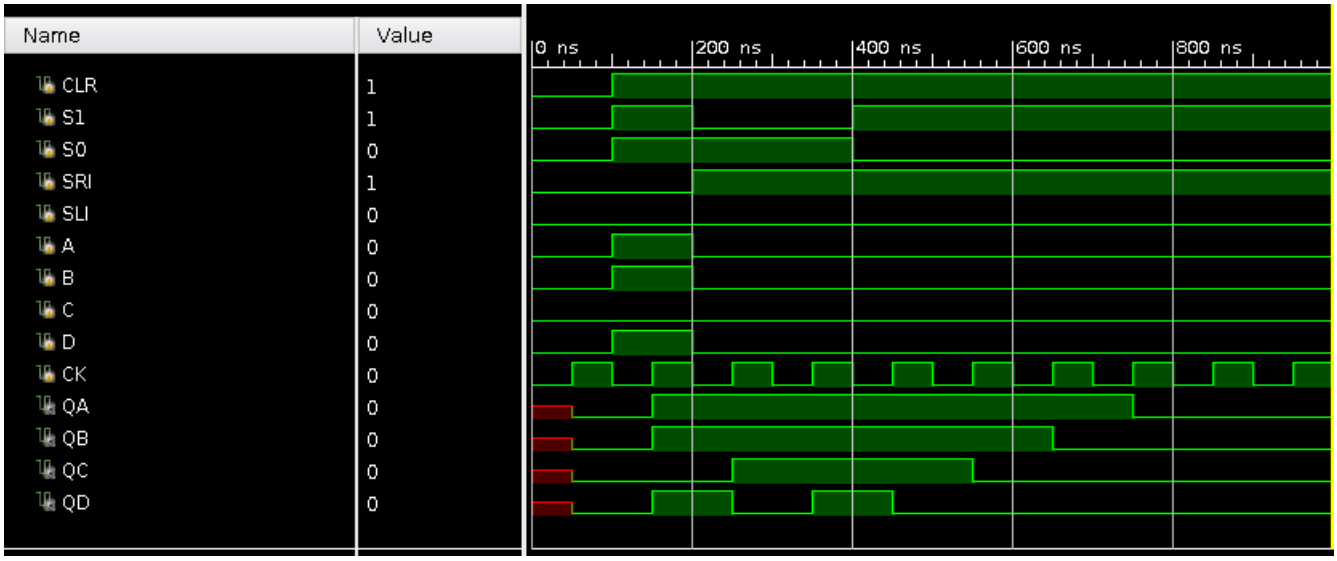
Intrare					Iesire		
S			R	CE	D	C	Qn
1			X	X	X	posedge	1
0			1	X	X	Posedge	0
0			0	0	X	X	Q(n-1)
0		0	1	1	posedge	1	
0	0	1	0	Posedge	0		

```
module FDRSE(R, S, CE, D, C, Q);
input R, S, CE, D, C;
reg Q;
output Q;
always @(posedge C)
    if(R)
        Q = 0;
    else if(S)
        Q = 1;
    else if(CE)
        Q = D;
endmodule
```



Doua module X74_194

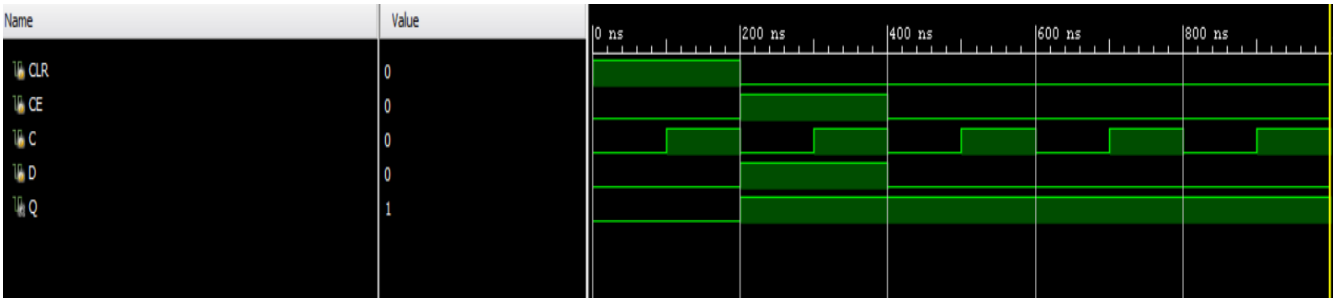
Intrare							Iesire			
CLR	S1	S0	SRI	SLI	A-D	CK	QA	QB	QC	QD
0	X	X	X	X	X	X	0	0	0	0
1	0	0	X	X	X	X	QA	QB	QC	QD
1	1	1	X	X	A-D	Pos	A	B	C	D
1	0	1	SRI	X	X	Pos	SRI	QA	QB	QC
1	1	0	X	SLI	X	Pos	QB	QC	QD	SLI




```
module X74_194(CLR, S1, S0, SRI, SLI, A, B, C, D, CK, QA, QB, QC, QD);

    input CLR, S1, S0, SRI, SLI, A, B, C, D, CK;
    reg QA, QB, QC, QD;
    output QA, QB, QC, QD;

    always @(posedge CK) begin
        if(CLR == 0)
            begin
             QA = 0;
             QB = 0;
             QC = 0;
             QD = 0;
            end
        else if(S1 == 1 && S0 == 1)
            begin
             QA = A;
             QB = B;
             QC = C;
             QD = D;
            end
        else if(S1 == 1)
            begin
             QA = QB;
             QB = QC;
             QC = QD;
             QD = SLI;
            end
        else if(S0 == 1)
            begin
             QD = QC;
             QC = QB;
             QB = QA;
             QA = SRI;
            end


    end

endmodule
```

Un modul FDCE

Intrare				Iesire
CLR	CE	D	C	Q
1	X	X	X	0
0	0	X	X	Q
0	1	D	Posedge	D



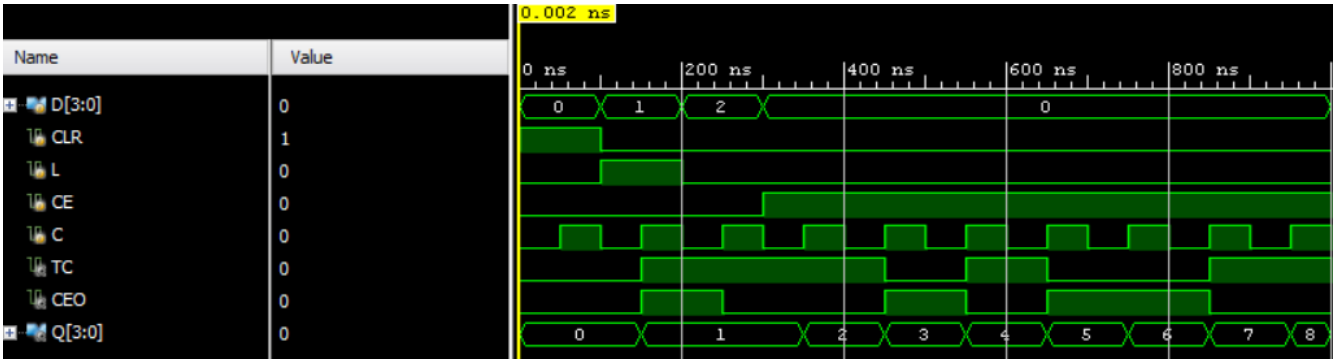

```
module FDCE(CLR, CE, C, D, Q);

    input CLR, CE, C, D;
    reg Q;
    output Q;

    always @(posedge C or CLR)
    begin
        if(CLR)
            Q <= 0;
        else if (CE)
            Q <= D;
    end


endmodule
```

Un modul CB4CLE

Intrare					Iesire		
CLR	L	CE	C	D3-D0	Q3-Q0	TC	CEO
1	X	X	X	X	0	0	0
0	1	X	Posedge	D3-D0	D3-D0	TC	CEO
0	0	0	X	X	Q3-Q0	TC	0
0	0	1	Posedge	X	Inc	TC	CEO

Unde TC = Q0 ^ Q1 ^ Q2 ^ Q3

CEO = TC ^ CE




```
module CB4CLE(CLR, L, CE, C, D, Q, TC, CEO);

input [3:0] D;
reg [3:0] Q;
output [3:0] Q;

input CLR, L, CE, C;
reg  TC, CEO;
output TC, CEO;

always @(posedge C | CLR)
    begin
        if(CLR)
           begin
            Q = 0;
            TC = 0;
            CEO = 0;
           end
        else if(L)
            begin
               Q[3:0] = D[3:0];
               TC = Q[0] ^ Q[1] ^ Q[2] ^ Q[3];
               CEO = TC ^ CE;
            end
        else if(CE)
            begin
               Q[3:0] = Q[3:0] + 4'b0001;
               TC = Q[0] ^ Q[1] ^ Q[2] ^ Q[3];
               CEO = TC ^ CE;
            end
        else
            begin
               CEO = 0;
            end
    end

endmodule
```

- Un modul H2 de comanda

Modulul reprezinta un automat de stari cu semnale sincrone si asincrone.
Automatul trece din starea curenta in urmatoarea pe frontul crescator al ceasului daca trece sincron, insa la semnale asincrone trecerea in starea urmatoare este imediata. Semnalul de reset este sincron pe frontul crescator al ceasului.
Automatul are 6 stari:

Din starea 0 se da semnalul rdy la iesire si se asteapta semnalul start pentru trecere la starea 1. Cand automatul este rdy, circuitul emite 0 continuu la iesire (bitul de start).

Daca se da semnalul asincron “start” se trece direct in starea 1. Trebuie tinut cont ca semnalul “start” vine de la un alt circuit “receiver” si acesta poate dura o perioada foarte scurta de timp. Nu avem garantia ca va fi un semnal pe ceas.
Din starea 1 se da semnalul incarc si se trece in starea 2 pe ceas.
Din starea 2 se da semnalul de date disponibile (ddisp) si se asteapta cererea de date (dreq).
Se ajunge in starea 3 odata daca cererea de date e activa (asincron). La fel ca la semnalul de start, se are in vedere ca acesta poate sa fie un semnal de o durata foarte scurta de timp. Odata ce am primit semnalul, data date disponibile se face 0, cu inceperea semnalului async dreq. Se porneste transmisia datelor inceput cu transmisia bitului de start (0). Se sta in stare pana s-au trimis ce 12 biti. Numaratorul cb4cle numara 12 biti pana cand se seteaza bitul num12, sincron.
Am pus in bucla always si semnalul num12 dar acest lucru doar ca sa treaca la urmatoarea stare imediat cum bitul num12 e setat. Acest lucru nu face ca semnalul num12 sa fie asincron! Num12 este sincron, fiind iesirea unui circuit secvential CB4CLE sincron, folosit ca numarator, si un circuit combinational.

Se ajunge in starea 4 unde se da semnalul txrdy (transmission ready), se astepata rxrdy (receiver ready) (La fel semnal asincron scurt).
Odata ce acest semnal a fost dat se trece in starea initiala, iar automatul este din nou rdy.

```
module H2(ceas, reset, start, dreq, num12, rxrdy, rdy, incarc, ddisp, depl, txrdy);

    input ceas, reset, start, dreq, num12, rxrdy;
    reg rdy, incarc, ddisp, depl, txrdy;
    output rdy, incarc, ddisp, depl, txrdy;

    reg[2:0] current_state;
    reg[2:0] next_state;

    always @(posedge ceas)
        begin
            if(reset == 1)
                begin
                current_state = 3'b000;
                next_state = 3'b000;
                rdy = 0;
                incarc = 0;
                ddisp = 0;
                depl = 0;
                txrdy = 0;
                end
            else
                current_state = next_state;
         end
```

```
always @(start, dreq, rxrdy, reset, num12, current_state)
   begin
       if(current_state == 3'b000)
           begin
               rdy = 1;
               if(start == 1) begin
                   rdy = 0;
                   current_state = 3'b001;
                   next_state = 3'b001;
                   end
           end
       else if(current_state == 3'b001)
           begin
               rdy = 0;
               incarc = 1;
               next_state = 3'b010;
           end
       else if(current_state == 3'b010)
           begin
               incarc = 0;
               ddisp = 1;
               if(dreq == 1) begin
                   ddisp = 0;
                   current_state = 3'b011;
                   next_state = 3'b011;
                   end
           end
       else if(current_state == 3'b011)
           begin
               depl = 1;
               next_state = 3'b100;
           end
       else if(current_state == 3'b100)
           begin
               if(num12 == 1) begin
                   depl = 0;
                   txrdy = 1;
                   current_state = 3'b101;
                   next_state = 3'b101;
                end
           end
      else if(current_state == 3'b101)
           begin
               if(rxrdy == 1) begin
                   current_state = 3'b000;
                   next_state = 3'b000;
                end
           end
```

Observam simularea finala in poza de mai jos. In acest caz, intrarea este data de bitii 01010110. Observam ca dupa un semnal de reset timp de o perioada, se da un semnal de ready. Odata ce se da semnalul de start (de la un posibil circuit care comunica cu circuitul implementat de noi, semnaul de ready se inchide). La un moment automatul H2 anunta printr-un semnal la iesirea DDISP ca datele sunt disponibile. DREQ este dat de un alt circuit si este preluat asincron. La urmatoarea perioada automatul scoate bitul de 0 apoi intoarce bitii serial, impreuna cu un bit de paritate si doi biti de 1 pentru stop. Dupa ce bitii sunt scosi se observa transmisia semnalului RXRDY. In momnetul acesta automatul arata ca este RDY din nou de transmisie. Bitul Date_OPERATIE arata ca automatul a trecut prin toate cele 12 stari.