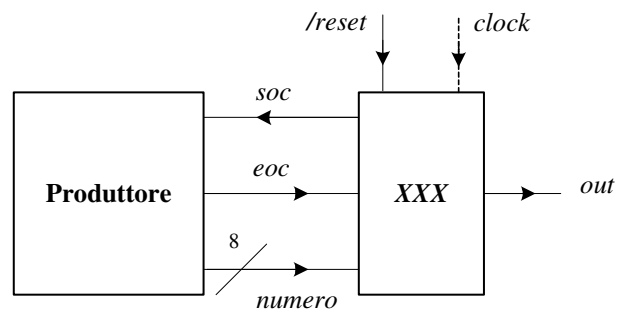


Esercizio 1

Si consideri un flip-flop D che è allo stesso tempo positive-edge-triggered e negative-edge-triggered, ovvero entra nella fase di campionamento sia in corrispondenza della transizione da 0 a 1 della variabile p , che in corrispondenza della transizione opposta di p da 1 a 0. Con riferimento ad una implementazione di tale flip-flop come coppia di reti campionatore-ritardatore, **descrivere** – completando opportunamente la tabella di flusso sotto riportata – e **sintetizzare** la rete campionatore sui fronti in salita e discesa.

d \ p	p = 0		p = 1		$z_1 z_0$
	0	1	0	1	
S_0	S_0	S_0			01
S_1	S_1	S_1			10
S_2					
S_3					

**Esercizio 2**

Descrivere e sintetizzare l'unità **XXX** in modo che emetta, tramite l'uscita *out*, una sequenza di 0 e di 1, mantenendo ogni livello per un numero di cicli di clock pari al *numero* naturale **richiesto e fornito** dal Produttore durante l'emissione del livello precedente

NOTA: Il Produttore emette numeri così grandi ed è così veloce da concludere un nuovo handshake in un tempo inferiore al tempo di permanenza del livello logico presente su *out*

Suggerimento per inizio descrizione di XXX, con condizioni al reset iniziale

```

module XXX(soc,eoc,numero,out,clock,reset_);
  input      clock,reset_;
  input      eoc;
  output     soc;
  input [7:0] numero;
  output     out;
  reg        SOC,OUT; assign soc=SOC; assign out=OUT;
  reg [7:0] DURATA_ZERO,DURATA_UNO;
  reg [...:0] STAR; parameter S0=0, S1=1, ... ;
  always @(reset_==0) begin OUT<=0; DURATA_ZERO<=8; ...; end
  always @(posedge clock) if (reset_==1) #3
    casex(STAR)
      ...
      ...
      ...
    endcase

```

Soluzione Esercizio 1

Una possibile descrizione della rete campionatore è data dalla seguente tabella di flusso:

d \ p	p = 0		p = 1		$z_1 z_0$
	0	1	0	1	
S_0	S_0	S_0	S_2	S_3	01
S_1	S_1	S_1	S_2	S_3	10
S_2	S_0	S_1	S_2	S_2	01
S_3	S_0	S_1	S_3	S_3	10

Adottando le codifiche $S_0 = 00$, $S_1 = 11$, $S_2 = 01$, $S_3 = 10$, la rete risulta esente da corse critiche delle variabili di stato. Con riferimento al modello strutturale con elementi di ritardo, la tabella delle transizioni corrispondente risulta essere

$y_1 y_0$ \ pd	pd				$z_1 z_0$
	00	01	11	10	
00	00	00	10	01	01
01	00	11	01	01	01
11	11	11	10	01	10
10	00	11	10	10	10

$a_1 a_0$

Per sintetizzare la rete CN1 si può procedere come segue:

$y_1 y_0$ \ pd	pd				a_1
	00	01	11	10	
00	0	0	1	0	
01	0	1	0	0	
11	1	1	1	0	
10	0	1	1	1	

$y_1 y_0$ \ pd	pd				a_0
	00	01	11	10	
00	0	0	0	1	
01	0	1	1	1	
11	1	1	0	1	
10	0	1	0	0	

Le liste di copertura sono opportunamente ridondate per assicurare l'assenza di alee statiche del primo ordine:

$$a_1 = \bar{p} \cdot y_1 \cdot y_0 + \bar{p} \cdot d \cdot y_0 + d \cdot y_1 + p \cdot y_1 \cdot \bar{y}_0 + p \cdot d \cdot \bar{y}_0$$

$$a_0 = \bar{p} \cdot d \cdot y_1 + p \cdot \bar{d} \cdot \bar{y}_1 + \bar{d} \cdot y_1 \cdot y_0 + d \cdot \bar{y}_1 \cdot y_0 + \bar{p} \cdot y_1 \cdot y_0 + \bar{p} \cdot d \cdot y_0 + p \cdot \bar{y}_1 \cdot y_0 + p \cdot \bar{d} \cdot y_0$$

Per la rete CN2, è immediato verificare che:

$$z_1 = y_1, \quad z_0 = \bar{y}_1.$$

Soluzione esercizio 2

```
module XXX(soc,eoc,numero,out,clock,reset_);
  input      clock,reset_;
  input      eoc;
  output     soc;
  input [7:0] numero;
  output     out;
  reg        SOC,OUT; assign soc=SOC; assign out=OUT;
  reg [7:0]   DURATA_ZERO, DURATA_UNO;
  reg [1:0]   STAR; parameter S0=0, S1=1, S2=2, S3=3;
  always @(reset_==0) begin DURATA_ZERO<=6; OUT<=0; SOC<=0; STAR<=S0; end
  always @(posedge clock) if (reset_==1) #3
    casex(STAR)
      S0: begin OUT<=0; DURATA_ZERO<=DURATA_ZERO-1;
              SOC<=1; STAR<=(eoc==1)?S0:S1; end
      S1: begin DURATA_ZERO<=DURATA_ZERO-1;
              SOC<=0; DURATA_UNO<=numero; STAR<=(DURATA_ZERO==1)?S2:S1; end
      S2: begin OUT<=1; DURATA_UNO<=DURATA_UNO-1;
              SOC<=1; STAR<=(eoc==1)?S2:S3; end
      S3: begin DURATA_UNO<=DURATA_UNO-1;
              SOC<=0; DURATA_ZERO<=numero; STAR<=(DURATA_UNO==1)?S0:S3; end
    endcase
endmodule
```