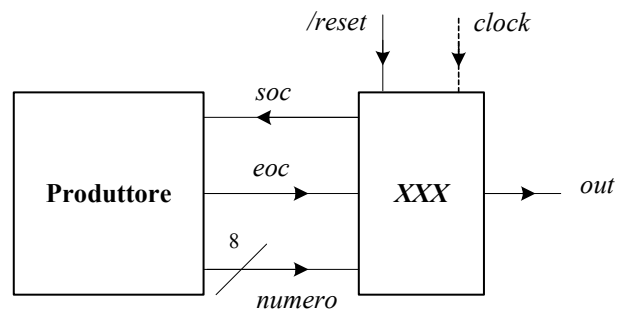


Esercizio 1

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

```