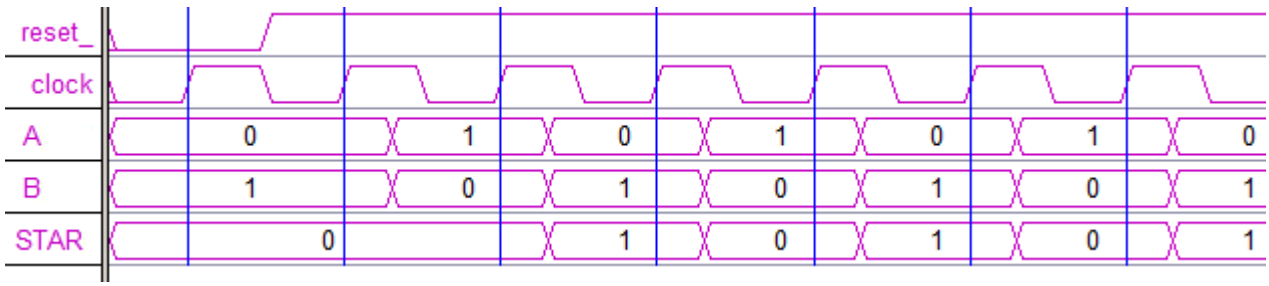
Per entrare nell'ottica delle temporizzazioni
delle reti con registri operativi e registro di stato

```
module XXX(clock,reset_);
 input        clock, reset_;

 //Registri Operativi
 reg A,B;

 //Registro di Stato
 reg STAR;
 parameter S0=0,S1=1;

always @(reset_==0) begin A<=0; B<=1; STAR<=S0; end
 always @(posedge clock)if (reset_==1) #3
  casex(STAR)
   S0:  begin A<=A+1; B<=A; STAR<=(A==0)?S0:S1; end
   S1:  begin B<=A; A<=B; STAR<=S0; end
  endcase
endmodule
```



Descrizioni di un Contatore, modulo 4, delle sequenze in ingresso del tipo 11,01,10

Modello di Mealy Ritardato

```
module Riconoscitore_e_Contatore_Modulo_4(z1_z0,x1_x0,clock,reset_);
 input        clock,reset_;
 input  [1:0] x1_x0;
 output [1:0] z1_z0;
 reg    [1:0] COUNT;   assign z1_z0=COUNT;
 reg    [3:0] STAR;
 parameter    S0=0,S1=1,S2=2,S3=3,S4=4,S5=5,S6=6,
              S7=7,S8=8,S9=9, S10=10,S11=11;
 always @(reset_==0) begin COUNT<=0; STAR<=S0;
 always @(posedge clock) if (reset_==1) #3
  casex(STAR)
   S0 : begin COUNT<=0; STAR<=(x1_x0=='B11)?S1:S0; end
   S1 : begin COUNT<=0; STAR<=(x1_x0=='B01)?S2:(x1_x0=='B11)?S1:S0; end
   S2 : begin COUNT<=(x1_x0=='B10)?1:0;
              STAR<=(x1_x0=='B11)?S1:(x1_x0=='B10)?S3:S0; end
   S3 : begin COUNT<=1; STAR<=(x1_x0=='B11)?S4:S3; end
   S4 : begin COUNT<=1; STAR<=(x1_x0=='B01)?S5:(x1_x0=='B11)?S4:S3; end
   S5 : begin COUNT<=(x1_x0=='B10)?2:1;
              STAR<=(x1_x0=='B11)?S4:(x1_x0=='B10)?S6:S3; end
   S6 : begin COUNT<=2; STAR<=(x1_x0=='B11)?S7:S6; end
   S7 : begin COUNT<=2; STAR<=(x1_x0=='B01)?S8:(x1_x0=='B11)?S7:S6; end
   S8 : begin COUNT<=(x1_x0=='B10)?3:2;
              STAR<=(x1_x0=='B11)?S7:(x1_x0=='B10)?S9:S6; end
   S9 : begin COUNT<=3; STAR<=(x1_x0=='B11)?S10:S9; end
   S10: begin COUNT<=3; STAR<=(x1_x0=='B01)?S11:(x1_x0=='B11)?S10:S9; end
   S11: begin COUNT<=(x1_x0=='B10)?0:3;
              STAR<=(x1_x0=='B11)?S10:(x1_x0=='B10)?S0:S9; end
  endcase
endmodule
```

Modello con Registri Operativi e Registro di Stato

```
module Riconoscitore_e_Contatore_Modulo_4(z1_z0,x1_x0,clock,reset_);
 input        clock,reset_;
 input  [1:0] x1_x0;
 output [1:0] z1_z0;
 //Registro Operativo
 reg    [1:0] COUNT;   assign z1_z0=COUNT;
 //Registro di Stato
 reg    [1:0] STAR;    parameter S0=0,S1=1,S2=2;
 always @(reset_==0) begin COUNT<=0; STAR<=S0;
 always @(posedge clock) if (reset_==1) #3
  casex(STAR)
   S0: begin COUNT<=COUNT; STAR<=(x1_x0=='B11)?S1:S0; end
   S1: begin COUNT<=COUNT; STAR<=(x1_x0=='B01)?S2:(x1_x0=='B11)?S1:S0; end
   S2: begin COUNT<=(x1_x0=='B10)?(COUNT+1):COUNT;
             STAR<=(x1_x0=='B11)?S1:S0; end
  endcase
endmodule
```