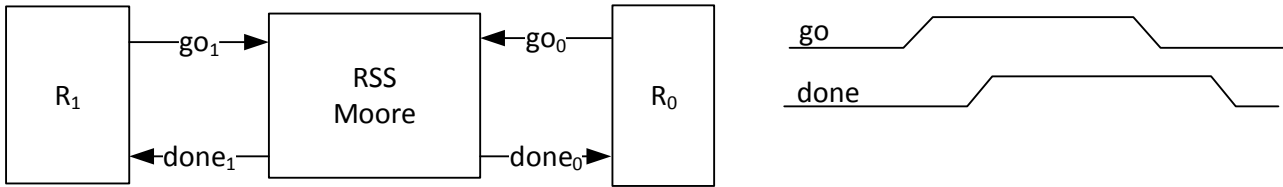


**Esercizio 1**

Descrivere e sintetizzare una rete sequenziale sincronizzata di Moore che ha due variabili di ingresso  $go_1, go_0$ , e due variabili di uscita  $done_1, done_0$ . Tali variabili supportano due handshake con altrettante reti  $R_1$  ed  $R_0$ , e gli handshake sono disegnati in figura. Le due reti  $R_1$  ed  $R_0$  si attengono al protocollo di handshake.



Si supponga che al reset tutti gli handshake siano a riposo. La RSS gestisce **un handshake alla volta**, e le due uscite  $done_1, done_0$  non sono mai settate contemporaneamente. Se  $R_1$  inizia un handshake, la RSS risponde portando  $done_1$  ad 1, ma se dopo  $R_0$  inizia a sua volta un handshake, la RSS non setta  $done_0$  finché non ha resettato  $done_1$ . Viceversa, se  $R_0$  inizia per prima. Se  $R_1$  ed  $R_0$  iniziano l'handshake contemporaneamente, la rete gestisce prima l'handshake con  $R_0$ , poi quello con  $R_1$ .

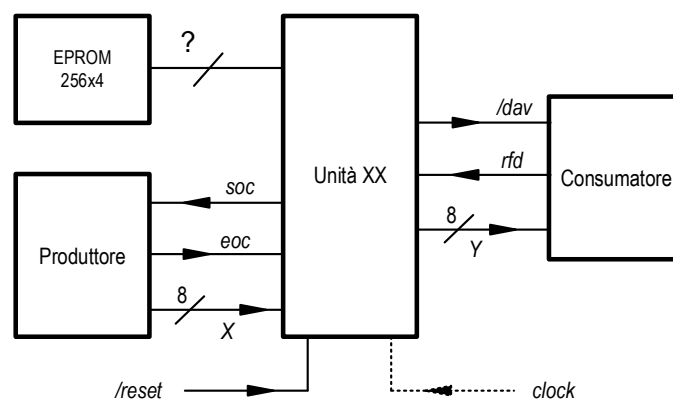
Usare un modello di sintesi con D-FF come elemento di marcatura, e sintetizzare a porte NOR la rete combinatoria che ha come uscita le variabili di stato.

**Esercizio 2**

Descrivere l'Unità XX, definita funzionalmente come segue.

- 1) Riceve numeri naturali a 8 bit dal Produttore ed invia numeri naturali a 8 bit al Consumatore, instaurando con il Produttore un handshake del tipo *soc, eoc* e con il Consumatore un handshake del tipo */dav, rfd*.
- 2) Ogni volta che riceve dal Produttore un nuovo numero naturale  $X$ , lo interpreta come la rappresentazione in complemento a 2 di un integer  $x$  ed utilizza  $ABS(x)$  come indirizzo per accedere (in lettura) alla EPROM da 256x4 bit.
- 3) Se i quattro bit ritornati dalla EPROM esprimono un numero naturale primo, allora il numero naturale  $X$  viene trasmesso al Consumatore, altrimenti viene ignorato e viene iniziato un nuovo ciclo di acquisizione di un nuovo numero naturale dal Produttore, e così via all'infinito.

**Nota:** Porre al reset iniziale OUT a 0; sintetizzare la parte operativa relativa al registro OUT; ricordare che sono numeri primi i numeri 2, 3, ....



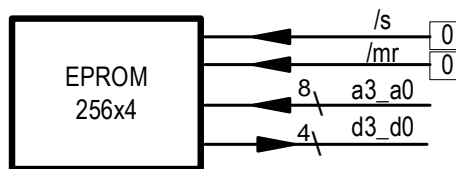
**Soluzione esercizio 1**

Si chiami  $S_R$  lo stato interno iniziale, ed  $S_i$  lo stato interno in cui si gestisce l'handshake con la rete  $R_i$ . La tabella di flusso della rete è la seguente:

$g_{o1}, g_{o0}$		$done_1, done_0$			
	00	01	11	10	
SR	SR	S0	S0	S1	00
S0	SR	S0	S0	S1	01
S1	SR	S0	S1	S1	10

Scegliendo la codifica (ovvia)  $S_R = 00$ ,  $S_0 = 01$ ,  $S_1 = 10$ , la rete combinatoria che genera le uscite è un cortocircuito,  $done_i = y_i$ , e la rete combinatoria a porte NOR che genera gli ingressi ai DFF è la seguente:

$$a_1 = \overline{g_{o1}} + (\overline{g_{o0}} + y_1), a_0 = \overline{g_{o0}} + (\overline{g_{o1}} + y_1)$$

**Soluzione esercizio 2**

```

module XXX(eoc,soc,X, dav_,rfd,Y, d3_d0,a7_a0, clock, reset_);
  input      clock, reset_;
  input      eoc, rfd;
  output     soc, dav_;
  input[7:0] X;
  output[7:0] Y;
  input[3:0] d3_d0;
  output[7:0] a7_a0;
  reg        SOC,DAV_; assign soc=SOC; assign dav_=DAV_;
  reg[7:0] OUT;      assign Y=OUT;
  reg[7:0] A7_A0;    assign a7_a0=A7_A0;
  reg[2:0] STAR;     parameter S0=0,S1=1,S2=2,S3=3,S4=4;
  always @(reset_==0) begin SOC=0; DAV_=1; OUT<=0; STAR=S0; end
  always @(posedge clock) if (reset_==1) #3
    casex(STAR)
      S0: begin SOC<=1; STAR<=(eoc==1)?S0:S1; end
      S1: begin SOC<=0; A7_A0<=(X[7]==0)?X:(~(X)+1); STAR<=(eoc==0)?S1:S2; end
      S2: begin OUT<=(primo(d3_d0)==1)?X:OUT; STAR<=(primo(d3_d0)==1)?S3:S0; end
      S3: begin DAV_<=0; STAR<=(rfd==1)?S3:S4; end
      S4: begin DAV_<=1; STAR<=(rfd==0)?S4:S0; end
    endcase

  function primo;
    input[3:0] d3_d0;
    casex(d3_d0)
      2: primo=1;
      3: primo=1;
      5: primo=1;
      7: primo=1;
      11: primo=1;
      13: primo=1;
    default: primo=0;
    endcase
  endfunction
endmodule

```