

Nome e Cognome \_\_\_\_\_ Matricola \_\_\_\_\_

**Esercizio 1**

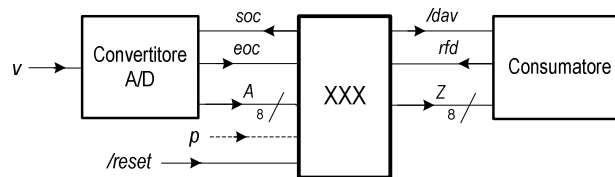
- 1) Sintetizzare una rete combinatoria *RRR* senza far uso di sommatori/sottrattori che funziona da *rivelatore di parità*. La rete riceve in ingresso una configurazione  $X$  di  $n$  bit ed un bit di parità  $p$  e genera in uscita un bit  $e$  che segnala un eventuale errore. In particolare se  $p$  vale 1 e  $X$  ha un numero *dispari* di bit a 1 ovvero se  $p$  vale 0 e  $X$  ha un numero *pari* di bit a 1, allora non c'è errore di parità e *RRR* mette  $e$  a 0, altrimenti c'è errore di parità e *RRR* mette  $e$  a 1.  
**NB:** la domanda verrà considerato *nulla* se non viene rispettato il vincolo sottolineato.

- 2) Progettare e sintetizzare una rete sequenziale sincronizzata *SSS* che riceve in ingresso l'uscita  $e$  di *RRR* ed ha a sua volta un'unica variabile di uscita  $z$ . Tale rete tiene normalmente  $z$  a zero e la mette ad 1 al terzo errore di parità (**anche non consecutivo**) rilevato, dopo di che si blocca fino al successivo reset.

**Esercizio 2**

Descrivere e sintetizzare l'unità *XXX* in modo che risponda alle seguenti specifiche:

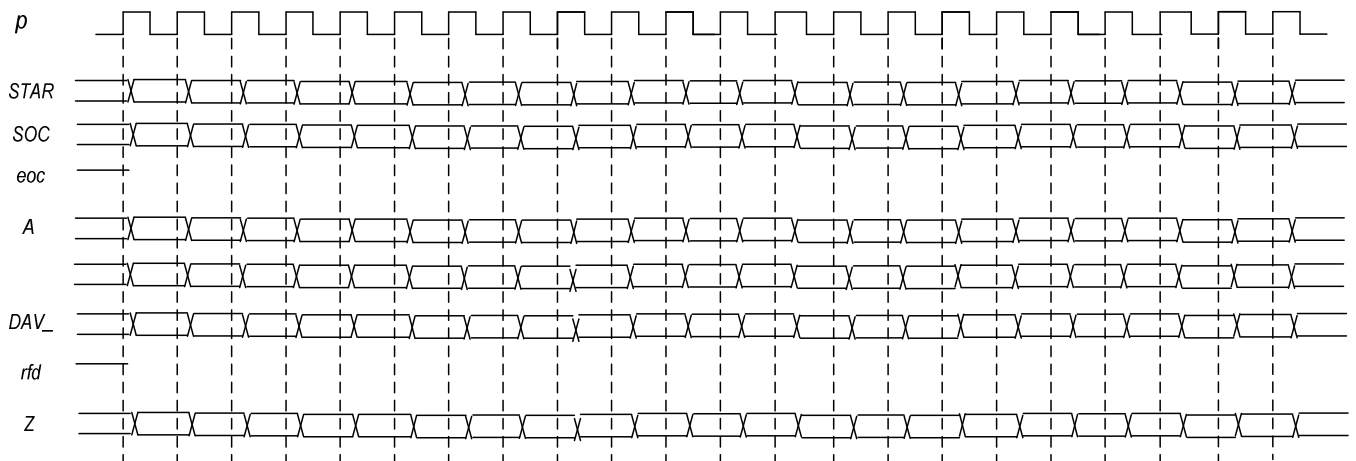
- 1) Preleva un byte  $A$  da un *convertitore A/D* e lo interpreta come la rappresentazione in complemento a due di un numero intero  $a$ ;
- 2) Invia al Consumatore la rappresentazione  $Z$  del numero intero  $z = -(a/2)$ ;
- 3) Torna la punto 1.



Disegnare il diagramma di temporizzazione nell'ipotesi che il Convertitore invii a XXX il byte  $A=10010001$ . Per non fare un diagramma troppo lungo, si supponga che i tempi di risposta negli handshake siano piuttosto brevi (ad esempio un clock e mezzo)

**Data** la banalità dell'esercizio, sbagliare l'aritmetica e/o gli handshake significa sbagliare l'esercizio.

**Ricordare** che, fin dalla scuola elementare, si sa che in base dieci il quoziente di  $125/10$  è 12 e che per calcolarlo non serve un divisore.



Nome e Cognome \_\_\_\_\_ Matricola \_\_\_\_\_

**Esercizio 1 - Soluzione**

1) La rete *RRR* è lo XOR di tutti i fili di ingresso, come è immediato verificare.

Utilizzando la codifica S0=00, S1=01, S2=11, S3=10, si ottiene:

2) La tabella della rete *SSS* è la seguente:

e	0	1	z
S0	S0	S1	0
S1	S1	S2	0
S2	S2	S3	0
S3	S3	S3	1

y1y0	e	0	1	z
00		00	01	0
01		01	11	0
11		11	10	0
10		10	10	1

Con il che abbiamo:

$$a_1 = y_1 + y_0 \cdot e, \quad a_0 = \overline{y_1} \cdot e + y_0 \cdot \overline{e}, \quad z = y_1 \cdot \overline{y_0}.$$

**Esercizio 2 – Soluzione**

```

module XXX(soc,eoc, A, dav_,rfd,Z, p,reset_);
  input      p, reset_;
  output     soc;
  input      eoc;
  input [7:0] A;
  output     dav_;
  input      rfd;
  output [7:0] Z;

  reg        SOC,DAV_;      assign soc=SOC, dav_=DAV_;
  reg [7:0]  OUT;           assign Z=OUT;
  reg [1:0]  STAR;          parameter S0=0,S1=1,S2=2,S3=3;

  always @(posedge p or negedge reset_)
    if (reset_==0) begin SOC<=0; DAV_<=1; STAR=S0; end else #3
      casex(STAR)
        // Prelievo di un byte dal Convertitore A/D
        S0: begin SOC<=1; STAR<=(eoc==1)?S0:S1; end
        S1: begin SOC<=0; OUT<=(~{A[7],A[7:1]})+1; STAR<=(eoc==0)?S1:S2; end
        // Invio del byte al Consumatore
        S2: begin DAV_<=0; STAR<=(rfd==1)?S2:S3; end
        S3: begin DAV_<=1; STAR<=(rfd==0)?S3:S0; end
      endcase
endmodule

```

