

Cognome e Nome: _____ Matricola _____

Prima della consegna barrare una delle due caselle sottostanti. L'opzione scelta non può essere modificata dopo la consegna.

Chiedo che la mia prova scritta sia corretta e valutata subito, perché intendo sostenere la prova orale in questo appello. Prendo atto che, a seguito della mia decisione, la mia prova scritta cesserà di essere valida al termine di questo appello e non potrà essere usata per l'appello straordinario di Aprile.

☐

Chiedo che la mia prova scritta sia corretta e valutata dopo la fine dell'appello in corso, perché ho diritto a ed intenzione di rimandare la prova orale all'appello straordinario di Aprile. Prendo atto che il mio diritto a rimandare la prova orale sarà oggetto di verifica, e che dovrò ripetere l'intero esame da capo se la verifica darà esiti negativi (per qualunque motivo).

☐

Esercizio 1

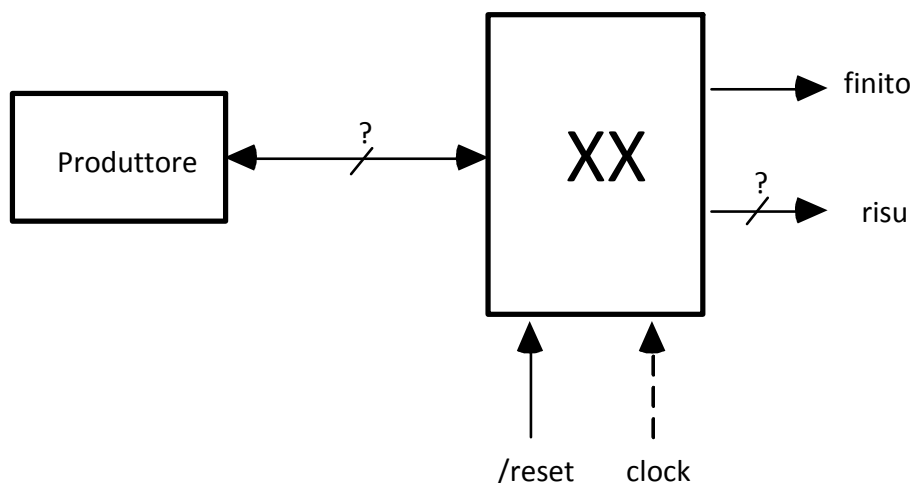
Sia x un intero in base $\beta = \text{dieci}$ su $n = 1$ cifra e sia X la sua rappresentazione secondo la codifica BCD (8421).

- 1) Descrivere e sintetizzare in forma PS la rete combinatoria che ha in ingresso la rappresentazione X di x e produce in uscita la rappresentazione Y di $-x$.
- 2) Calcolare le uscite della rete (istanziando le espressioni algebriche della sintesi) quando x assume il *minimo* ed il *massimo* valore rappresentabile.

Esercizio 2

L'unità **XX** che ciclicamente preleva un byte da un Produttore del tipo visto a lezione, lo elabora calcolando quante coppie di bit 01 sono contenute nel byte (qualunque sia la posizione che occupano) ed emette il risultato del conteggio tramite la variabile di uscita *risu*. Per notificare ad un eventuale consumatore che un nuovo conteggio è terminato, pone poi ad 1 la variabile di uscita *finito* per un periodo di clock, dopodiché ricomincia da capo.

- 1) Specificare i collegamenti della figura
- 2) Descrivere e sintetizzare l'unità **XX**. Nella sintesi evidenziare le equazioni delle reti per le variabili di condizionamento e disegnare la porzione di parte operativa relativa al registro in cui viene via via memorizzato il numero delle coppie di bit 01



Esercizio 1 - soluzione

Per i numeri in oggetto, l'essere $a \leftrightarrow A$ su una cifra in base dieci, significa $-5 \leq a < +5$ e

$$A = \begin{cases} a & \text{se } a \geq 0 \\ 10 + a & \text{se } a < 0 \end{cases}$$

cioè

a	A
-5	5
-4	6
-3	7
-2	8
-1	9
0	0
1	1
2	2
3	3
4	4

Il risultato $y = -x$ è rappresentabile se e solo se $x > -5$. In accordo alla codifica 8421, si ha $X = (x_3 x_2 x_1 x_0)_2$ e $Y = (y_3 y_2 y_1 y_0)_2$, e la rete è descritta dalla seguente tabella di verità:

x	$y = -x$	X	Y	$x_3 x_2 x_1 x_0$	$y_3 y_2 y_1 y_0$
0	0	0	0000	0000	0000
1	-1	1	9	0001	1001
2	-2	2	8	0010	1000
3	-3	3	7	0011	0111
4	-4	4	6	0100	0110
-4	4	6	4	0110	0100
-3	3	7	3	0111	0011
-2	2	7	2	1000	0010
-1	1	9	1	1001	0001
		others	----	others	----

cui corrispondono le mappe di Karnaugh:

$x_1 x_0$	$x_3 x_2$				$x_1 x_0$	$x_3 x_2$			
	00	01	11	10		00	01	11	10
00	00	01	--	00	00	00	10	--	10
01	10	--	--	00	01	01	--	--	01
11	01	00	--	--	11	11	11	--	--
10	10	01	--	--	10	00	00	--	--
$y_3 y_2$					$y_1 y_0$				

Una possibile realizzazione a costo minimo di tipo PS è la seguente:

$$y_3 = \bar{x}_2 \cdot \bar{x}_3 \cdot (x_1 + x_0) \cdot (\bar{x}_1 + \bar{x}_0),$$

$$y_2 = (x_2 + x_0) \cdot (x_1 + \bar{x}_0) \cdot (\bar{x}_2 + \bar{x}_0),$$

$$y_1 = (x_1 + x_2 + x_3) \cdot (\bar{x}_0 + x_1) \cdot (x_0 + \bar{x}_1),$$

$$y_0 = x_0.$$

La rete ha un'uscita di overflow, che va ad 1 quando $X = 5 = (0101)_2$. Dalla sintesi PS si ottiene immediatamente: $ov = x_1 + \bar{x}_0 + \bar{x}_3$.

Soluzione Esercizio 2

```
//-----
module XX(byte_in,dav_,rfd, risu,finito, p,reset_);
    input          p,reset_;

    input [7:0] byte_in;
    input      dav_;
    output     rfd;
    output[2:0] risu;
    output     finito;

    reg [7:0] APP;
    reg [2:0] RISU;
    reg      RFD;
    reg      FINITO;

    reg [2:0] STAR;
    parameter[2:0] S0=0, S1=1, S2=2, S3=3;

    assign rfd=RFD;
    assign finito=FINITO;
    assign risu=RISU;

    always @(reset_==0) begin STAR=S0; end
    always @(posedge p) if (reset_==1) #3
        casex(STAR)
            S0: begin FINITO<=0; APP<=byte_in; RISU=0; RFD<=1;
                  STAR<=(dav_==0)?S1:S0; end
            S1: begin RFD<=0; STAR<=(dav_==0)?S1:S2; end
            S2: begin RFD<=1; RISU<=(APP[7:6]=='B01)?(RISU+1): RISU;
                  APP<={APP[6:0],1'B0}; STAR<=(APP=='H00)? S3 : S2; end
            S3: begin FINITO<=1; STAR<=S0; end
        endcase
endmodule
//-----
```