

Reti logiche - Prova scritta del 9 Settembre 2016

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 Novembre.

☐

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 Novembre. 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).

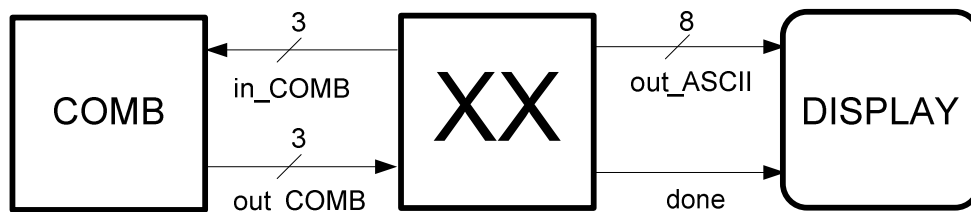
☐

Esercizio 1

Descrivere la rete sequenziale asincrona con due variabili di ingresso x_1 e x_0 ed una variabile di uscita z , che si evolve come segue: se $x_1 = x_0$, allora $z = x_1 = x_0$; altrimenti, se $x_1 \neq x_0$, allora z commuta. Sintetizzare la rete secondo il modello ad elementi neutri di ritardo, con la rete combinatoria RC1 in forma PS. Detto T_a il tempo di attraversamento della RC1, stabilire il tempo minimo di permanenza di un ingresso affinché la RSA sia pilotata correttamente.

Esercizio 2

Descrivere e sintetizzare l'unità XX avente le specifiche che seguono.



COMB è una rete combinatoria da ispezionare. Il Display accetta e visualizza i caratteri di cui riceve le codifiche ASCII su *out_ASCII* purché notificate da un fronte in discesa su *done* (normalmente a 0). L'unità XX deve far comparire sul Display la tabella di verità di COMB con il formato che si desume dall'esempio che segue. Terminata l'ispezione di COMB, l'unità XX si blocca fino a nuovo reset asincrono. Il tempo di risposta di COMB è inferiore al periodo del clock.

Esempio. Se COMB avesse la seguente tabella di verità:

in COMB	out COMB
000	010
001	100
010	011
011	110
100	001
101	111
110	000
111	101

sul Display dovrebbe comparire:

0:2
1:4
2:3
3:6
4:1
5:7
6:0
7:5

Le codifiche ASCII dei caratteri che interessano sono le seguenti:

0011 0000	0	0011 1010	:
0011 0001	1	0000 1010	Cursore su una nuova linea
0011 0010	2	0000 1101	Cursore all'inizio della nuova linea
0011 0011	3		
0011 0100	4		
0011 0101	5		
0011 0110	6		
0011 0111	7		

Soluzione esercizio 1

La rete sequenziale asincrona è descritta dalla seguente tabella di flusso:

x_1x_0						z
		00	01	11	10	
S_0	S_0	S_1	—	S_1	0	
S_1	S_0	S_1	S_2	S_1	1	
S_2	—	S_3	S_2	S_3	1	
S_3	S_0	S_3	S_2	S_3	0	

Gli stati S_0 ed S_2 sono gli stati in cui la rete si stabilizza quando i valori delle variabili di ingresso sono entrambi uguali rispettivamente a 0 e 1, mentre gli stati S_1 ed S_3 sono stati stabili raggiunti quando l'uscita commuta rispettivamente da 0 a 1, e da 1 a 0.

Adottando le codifiche $S_0 = 00$, $S_1 = 01$, $S_2 = 11$, e $S_3 = 10$, la rete è esente da corse critiche. Con riferimento al modello strutturale con elementi neutri di ritardo (non necessari perché la rete è priva di alee essenziali), le mappe di Karnaugh relative alle uscite della rete CN1 sono:

$y_1y_0 \backslash x_1x_0$						z
		00	01	11	10	
y_1y_0	00	00	01	--	01	0
	01	00	01	11	01	1
	11	--	10	11	10	1
	10	00	10	11	10	0
		a_1a_0				

Una possibile coppia di forme PS corrispondenti (esenti da alee statiche) è:

$$a_1 = (x_1 + x_0) \cdot (x_0 + y_1) \cdot (x_1 + y_1),$$

$$a_0 = (x_1 + x_0) \cdot (x_0 + \bar{y}_1) \cdot (x_1 + \bar{y}_1).$$

Per la rete CN2, è immediato verificare che:

$$z = y_0.$$

Il tempo minimo di permanenza di uno stato di ingresso è pari a $2T_a$, visto che non è necessario un ritardo di marcatura e la RSA è normale.

Soluzione esercizio 2

```
//=====
module XX(in_COMB,out_ASCII,done,out_COMB,clock,reset_);
  input      clock,reset_;
  output [2:0] in_COMB;
  input  [2:0] out_COMB;
  output      done;
  output [7:0] out_ASCII;
  reg  [2:0] IN_COMB;    assign in_COMB=IN_COMB;
  reg      DONE;        assign done=DONE;
  reg  [7:0] OUT_ASCII;  assign out_ASCII=OUT_ASCII;
  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;

  always @(reset_==0) #1 begin DONE<=0; IN_COMB<='B000; STAR=S0; end
  always @(posedge clock) if (reset_==1) #3
  casex(STAR)
    S0: begin OUT_ASCII<={'B00110,IN_COMB}; DONE<=1; STAR<=S1; end
    S1: begin DONE<=0; STAR<=S2; end
    S2: begin OUT_ASCII<='B00111010; DONE<=1; STAR<=S3; end
    S3: begin DONE<=0; STAR<=S4; end
    S4: begin OUT_ASCII<={'B00110,out_COMB}; DONE<=1; STAR<=S5; end
    S5: begin DONE<=0; STAR<=S6; end
    S6: begin OUT_ASCII<={'B00001010}; DONE<=1; STAR<=S7; end
    S7: begin DONE<=0; STAR<=S8; end
    S8: begin OUT_ASCII<={'B00001101}; DONE<=1; STAR<=S9; end
    S9: begin DONE<=0; IN_COMB<=IN_COMB+1; STAR<=(IN_COMB=='B111)?S10:S0; end
    S10: begin STAR<=S10; end
  endcase
endmodule
//=====
```