



**UNIVERSITÀ DEGLI STUDI DI PARMA
FACOLTÀ DI INGEGNERIA**

Corso di Reti Logiche A

anno accademico 2005-2006
prof. Stefano CASELLI
prof. William FORNACIARI

I compitino del 11 novembre 2005

Bozza soluzioni del 21.11.2005

Cognome (LEGGIBILE).....Nome (LEGGIBILE)	
Matricola Firma	
Corso di Laurea: <input type="checkbox"/> ing informatica; <input type="checkbox"/> telecom; <input type="checkbox"/> elettronica; <input type="checkbox"/> altro (specif)	
<input type="checkbox"/> teledidattico	
Professore di riferimento: <input type="checkbox"/> CASELLI; <input type="checkbox"/> FORNACIARI	

Riportare i dati personali in modo che siano **LEGGIBILI**

D1	D2	D3	D4	D5	TOT
					33

NOTE PER LO SVOLGIMENTO

Si raccomanda di essere sintetici (al fine di evitare inutili perdite di tempo in trattazioni generiche e poco significative ai fini della valutazione finale) e ordinati allo scopo di migliorare l'interpretazione da parte dei correttori.

I temi proposti debbono essere risolti utilizzando unicamente lo spazio lasciato al termine del testo di ogni quesito, il retro delle pagine o, eventualmente, utilizzando lo spazio finale. Verranno corretti SOLO i fogli componenti il tema d'esame.

È vietato consultare testi o appunti di qualunque genere così come interagire con i vicini. Chiunque sia trovato in possesso di materiale relativo al corso, anche se non strettamente attinente al tema d'esame, vedrà annullata la prova.

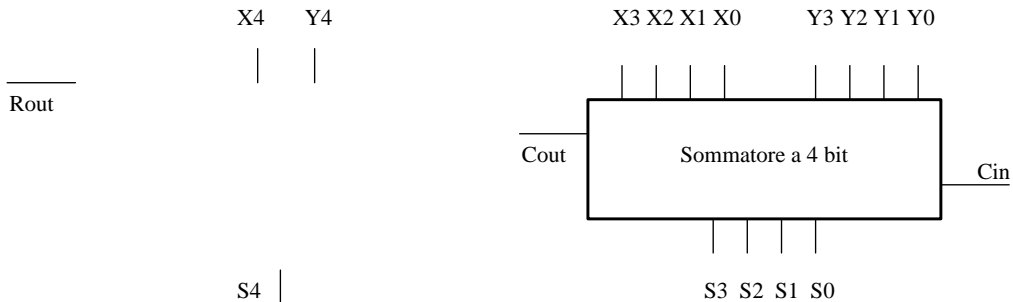
Quesito D1

È dato il blocco funzionale combinatorio, riportato in figura, che realizza la somma di due operandi a 4 bit e di un eventuale riporto in ingresso.

Si realizzi un sommatore di operandi a 5 bit facendo uso del **blocco** sommatore assegnato e di SOLE porte logiche **NAND** a due ingressi. Le uscite del circuito risultante saranno pertanto **S4,S3,S2,S1,S0** e il riporto in uscita **Rout**, mentre gli ingressi saranno i due operandi a 5 bit (**X4,X3,X2,X1,X0; Y4,Y3,Y2,Y1,Y0**) e il riporto in ingresso **Cin**.

Saranno privilegiate soluzioni anche su più **livelli** ovvero in grado di riusare sottocircuiti comuni.

Si riportino i **passaggi**, le **espressioni logiche intermedie** e **finali** e si completi opportunamente il disegno del circuito.



Soluzione

In pratica si deve aggiungere una cella Full-Adder da 1 bit in cascata al blocco sommatore.

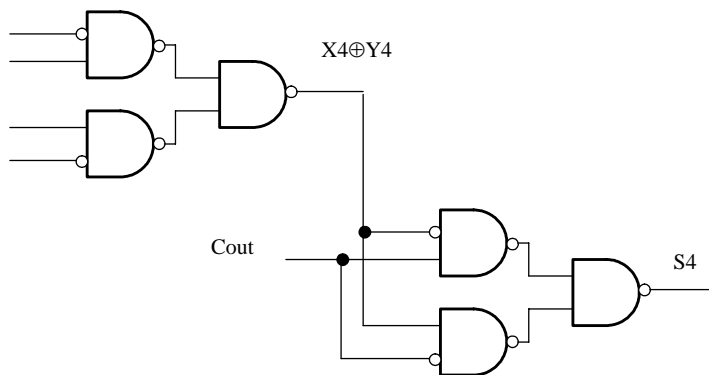
$$S4 = (X4 \oplus Y4) \oplus Cout$$

$$Rout = Cout'X4Y4 + Cout (X4+Y4)$$

Per la sintesi si può procedere nel modo classico, oppure si può osservare che l'espressione di S4 può scriversi come:

$$(X4'Y4+X4Y4') Cout' + (X4'Y4+X4Y4')' Cout,$$

dove $(X4'Y4+X4Y4') = ((X4'Y4)' (X4Y4)')'$ che corrisponde al circuito



Per quanto concerne Rout, si ha:

$$Rout = Cout'X4Y4 + Cout (X4+Y4) = ((Cout'X4Y4)' ((Cout (X4+Y4))')' = ((Cout'(X4Y4)')' (Cout ((X4' Y4')'))'$$

Il disegno del circuito è lasciato per esercizio allo studente

Quesito D2

Data la seguente tabella di verità, riportare (trascrivendo tutti i passaggi):

- Il risultato della **prima fase** del metodo di Quine McCluskey, ovvero la ricerca degli implicanti primi. Suggerimento: servirsi delle mappe di Karnaugh per verificare il risultato.
- La **seconda fase** del metodo, ovvero la ricerca della copertura ottima, commentando i vari passaggi.

mintermine	A	B	C	D	Uscita
0	0	0	0	0	x
1	0	0	0	1	
2	0	0	1	0	x
3	0	0	1	1	
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	x
7	0	1	1	1	1
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	1
12	1	1	0	0	
13	1	1	0	1	1
14	1	1	1	0	
15	1	1	1	1	1

M0	0000	* v	0,2	00-0	* v	0,2,4,6	0-0	P2=a'd'
			0,4	0-00	v			
M2	0010	* v				4,5,6,7	01--	P3=a'b
M4	0100	v	2,6	0-10	* v			
			4,5	010-	v	5,7,13,15	-1-1	P4=bd
M5	0101	v	4,6	01-0	v			
M6	0110	* v						
			5,7	01-1	v			
M7	0111	v	5,13	-101	v			
M11	1011	v	6,7	011-	v			
M13	1101	v						
			7,15	-111	v			
M15	1111	v	11,15	1-11	P1=acd			
			13,15	11-1	v			

copertura

	4	5	7	11	13	15
P1				x		x
P2	x					
P3	x	x				
P4		x	x		x	x

P1 e P4 essenziali (m11 e m13), poi P2 o P3 sono indifferenti

AB \ CD	00	01	11	10
00	X			X
01	1	1	1	X
11		1	1	
10			1	

Diagram showing prime implicants P1, P2, P3, and P4 circled on the Karnaugh map. P1 covers (11,1) and (11,3). P4 covers (01,1), (11,1), and (10,3). P2 covers (00,1) and (00,3). P3 covers (00,1) and (01,1).

Quesito D3

Si deve sintetizzare una rete combinatoria con 5 ingressi $F(A,B,C,D,S)$ descritta dalla mappa che segue.

- Per la sintesi sono disponibili e devono essere utilizzati un MUX ad un ingresso di selezione e porte logiche con al massimo tre ingressi. Si descriva il procedimento adottato per ricavare le espressioni logiche e si disegni il circuito corrispondente.
- Si discuta il problema delle alee statiche con riferimento al circuito ricavato. Le reti combinatorie collegate agli ingressi di programmazione del MUX sono immuni da fenomeni di alea statica? La funzione complessiva F osservata all'uscita del MUX è immune da alee statiche? Le risposte fornite devono essere adeguatamente motivate.

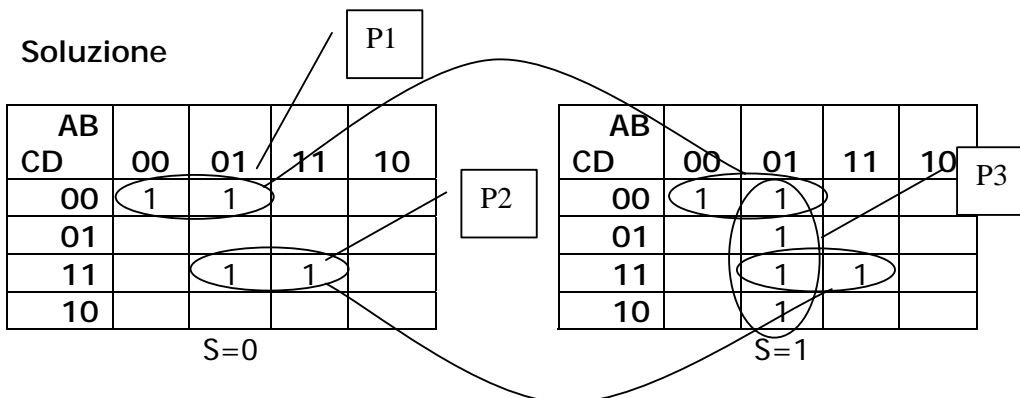
AB				
CD	00	01	11	10
00	1	1		
01				
11		1	1	
10				

S=0

AB				
CD	00	01	11	10
00	1	1		
01		1		
11		1	1	
10		1		

S=1

Soluzione



Vi sono due termini che possono essere condivisi, P1 e P2, entrambi a tre ingressi

$$F(S=0) = P1 + P2$$

$$F(S=1) = P1 + P2 + P3$$

La rete finale prevede l'uso di S per pilotare il MUX e la creazione di due reti, una per ogni ingresso del MUX, tutte composte da porte al più a tre ingressi.

Le reti combinatorie $F(S=0)$ e $F(S=1)$ non presentano alee statiche perché le espressioni individuate non hanno mintermini adiacenti coperti da implicanti diversi (si osservino le sottomappe qui sopra). La funzione F complessiva presa all'uscita del MUX ha invece alee statiche quando il MUX commuta tra i due canali. In particolare, esistono 4 configurazioni di (A,B,C,D) in cui sia $F(S=0)$ che $F(S=1)$ valgono 1, ma nella commutazione tra i due canali (variazione di S) si potrebbe avere un transitorio a 0 dell'uscita. Questo ovviamente nell'ipotesi che il funzionamento interno del MUX sia equivalente alla selezione di mintermini isolati, diversamente non è possibile asserire nulla.

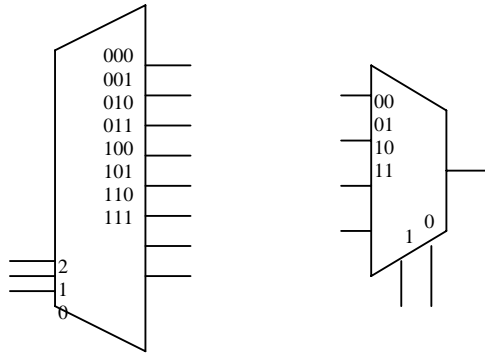
Quesito D4

Siano dati un **decoder** con 3 ingressi e 8 uscite e un **multiplexer** con due ingressi di selezione, come mostrato in figura.

Facendo uso di tali componenti e delle comuni porte logiche con **al massimo 4 ingressi**, si deve realizzare un circuito dotato di 5 ingressi da un bit (A, B, C, S_0, S_1) e di una uscita da un bit (Out), con la seguente funzionalità:

1. Quando $S_0, S_1 = 00$ allora $Out = A + B + C$
2. Quando $S_0, S_1 = 01$ allora $Out = 1$ se $A > (B + C)$
3. Quando $S_0, S_1 = 10$ allora Out è il riporto della somma di $A + B + C$
4. Quando $S_0, S_1 = 11$ allora $Out = 1$ se $A \leq (B + C)$

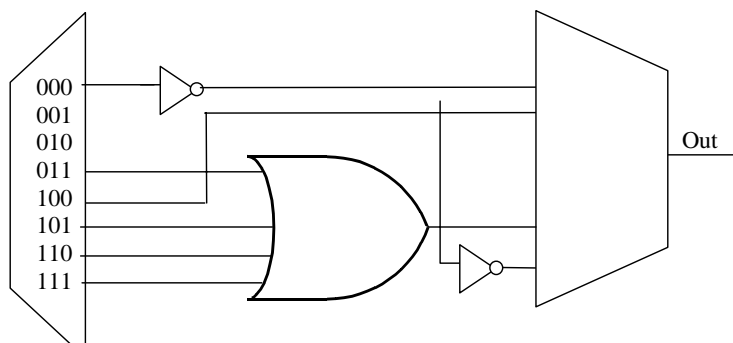
Si descriva come si perviene al risultato e si riporti il circuito corrispondente.



Soluzione

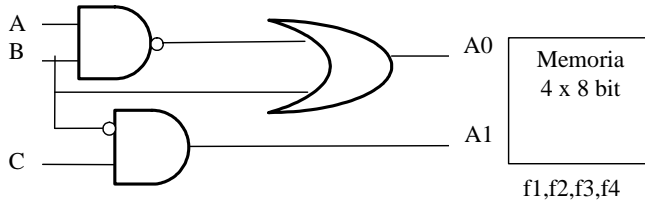
Si pone il decoder sulla sinistra, con in ingresso ABC , e sulla destra il MUX, pilotato da $S_1 S_0$. Il decoder genera tutti i mintermini di ABC , che verranno opportunamente utilizzati per realizzare le funzioni desiderate. Vengono quindi sintetizzate 4 funzioni che vanno ad alimentare gli ingressi di programmazione del MUX.

1. La somma $A + B + C$ vale 0 solo se tutti e tre i segnali sono contemporaneamente a 0, diversamente vale sempre 1. Quindi in questo caso si ha $(A'B'C)'$, risultato ottenibile anche facendo uso delle leggi della doppia negazione e De Morgan.
2. L'unico caso in cui $A > (B + C)$ è quello il cui $A, B, C = 100$.
3. Si ha riporto nella somma di $A + B + C$ nei casi in cui le triplette A, B, C abbiano due o tre uni, ovvero nei quattro casi: 101, 011, 110, 111. La rete corrispondente sarà un OR a 4 ingressi.
4. Questo caso è il complementare del punto 2). Si potrebbe usare un OR ma il numero di ingressi sarebbe (8-1), non compatibile con i vincoli di progetto. In alternativa si può quindi negare la funzione sintetizzata per il caso 2): occorre cioè negare l'uscita 100 del decoder prima di connetterla all'ultimo ingresso del MUX.



Quesito D5

Sia data una rete logica, ottenuta per *estrazione* da un progetto precedente, di dimensioni maggiori. La parte identificata è dotata di tre ingressi (a,b,c) e quattro uscite (f1, f2, f3, f4). La produzione delle uscite prevede di leggere i 4 bit delle varie f_i come i **primi 4 bit più significativi** delle parole di un byte indirizzate in una memoria ROM di 4 parole, selezionabili tramite gli ingressi di indirizzamento A1, A0, come mostrato in figura.



contenuto memoria ROM

indirizzo	Parola di memoria							
A1A0=00	0	1	0	1	0	0	1	1
A1A0=01	1	0	1	0	0	1	0	0
A1A0=10	0	1	0	1	1	0	0	0
A1A0=11	1	0	1	0	1	0	0	0
	F1	F2	F3	F4				

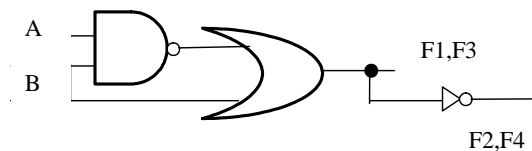
Data la struttura della rete appena esposta e il contenuto della memoria riportato, si determini una rete equivalente, il più possibile semplificata, in grado di generare le quattro uscite f1,f2,f3,f4 a partire dai 3 ingressi a,b,c.

Soluzione:

Dall'analisi del contenuto della memoria (usando per esempio le mappe di Karnaugh oppure per analisi di tipo algebrico) si ricava che:

$f_1 = A_0 = f_3$; $f_2 = A_0' = f_4$, visto che ci interessano solo i primi 4 bit.

La rete diviene quindi molto semplificata e degenera in funzioni costanti



$$F_1 = (ab)' + b = 1 \text{ da cui } F_2 = 0$$