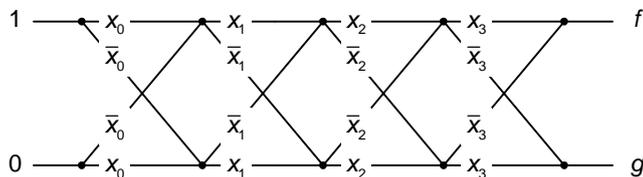


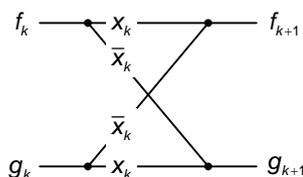
Reti Logiche

Appello del 18 dicembre 2006 – Seconde prove

(D2) Analizzare il comportamento della seguente rete a contatti:



Il circuito illustrato è una rete iterativa, la cui cella elementare



ha funzioni di trasmissione

$$\begin{aligned} f_{k+1} &= x_k f_k + \bar{x}_k g_k \\ g_{k+1} &= x_k g_k + \bar{x}_k f_k \end{aligned}$$

con $f_0 = 1$, $g_0 = 0$, $f_4 = f$, $g_4 = g$. Cominciamo con l'osservare che, se $f_k = \bar{g}_k$, allora è anche $f_{k+1} = \bar{g}_{k+1}$; infatti:

$$\begin{aligned} f_{k+1} &= x_k f_k + \bar{x}_k g_k = x_k f_k + \bar{x}_k \bar{f}_k = \overline{x_k \oplus f_k} \\ g_{k+1} &= x_k g_k + \bar{x}_k f_k = x_k \bar{f}_k + \bar{x}_k f_k = x_k \oplus f_k \end{aligned}$$

Dal momento che è anche $f_0 = \bar{g}_0$, questa osservazione consente di analizzare una sola delle due funzioni, ottenendo l'altra solo alla fine. Applicando iterativamente la formula per f_k , abbiamo:

$$\begin{aligned} f_1 &= \overline{x_0 \oplus f_0} = \overline{x_0 \oplus 1} = x_0 \\ f_2 &= \overline{x_1 \oplus f_1} = \overline{x_1 \oplus x_0} \\ f_3 &= \overline{x_2 \oplus f_2} = \overline{x_2 \oplus x_1 \oplus x_0} = \overline{x_2 \oplus x_1 \oplus x_0} \\ f_4 &= f = \overline{x_3 \oplus f_3} = \overline{x_3 \oplus x_2 \oplus x_1 \oplus x_0} \end{aligned}$$

e di conseguenza

$$g_4 = g = x_3 \oplus x_2 \oplus x_1 \oplus x_0$$

In definitiva, f e g sono rispettivamente la *funzione di parità* e la *funzione di disparità* delle quattro variabili x_0, x_1, x_2, x_3 .

In generale, in una rete costituita da n celle con variabili x_0, x_1, \dots, x_{n-1} , avremo:

$$f = \begin{cases} x_0 \oplus x_1 \oplus \dots \oplus x_{n-1} & \text{se } n \text{ è pari} \\ \overline{x_0 \oplus x_1 \oplus \dots \oplus x_{n-1}} & \text{se } n \text{ è dispari} \end{cases}$$
$$g = \begin{cases} \overline{x_0 \oplus x_1 \oplus \dots \oplus x_{n-1}} & \text{se } n \text{ è pari} \\ x_0 \oplus x_1 \oplus \dots \oplus x_{n-1} & \text{se } n \text{ è dispari} \end{cases}$$

(D3) Mostrare che la funzione

$$f(x_3x_2x_1x_0) = \sum(3, 7, 11, 12, 13, 14)$$

è sintetizzabile come $(x_3x_2) \oplus (x_1x_0)$.

La mappa di Karnaugh per la funzione f

		x_3x_2			
		00	01	11	10
x_1x_0	00			1	
	01		1		
	11	1	1		1
	10		1		

fornisce l'espressione in somma di prodotti

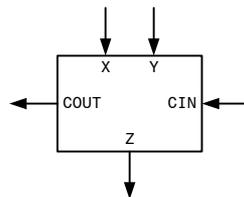
$$f = x_3x_2\bar{x}_1 + x_3x_2\bar{x}_0 + \bar{x}_3x_1x_0 + \bar{x}_2x_1x_0$$

che può essere manipolata come

$$\begin{aligned} f &= x_3x_2(\bar{x}_1 + \bar{x}_0) + x_1x_0(\bar{x}_3 + \bar{x}_2) = \\ &= x_3x_2\bar{x}_1x_0 + x_3x_2\bar{x}_1\bar{x}_0 = \\ &= (x_3x_2) \oplus (x_1x_0) \end{aligned}$$

(D4) Un addizionatore iterativo a N bit è costituito da sole porte logiche NAND con tempo di propagazione t_G ; descrivere la struttura della cella e valutare i tempi minimo e massimo impiegati dalla rete per calcolare la somma di due operandi a N bit.

Il comportamento della singola cella



è dato dalle equazioni:

$$Z = X \oplus Y \oplus C_{IN}$$

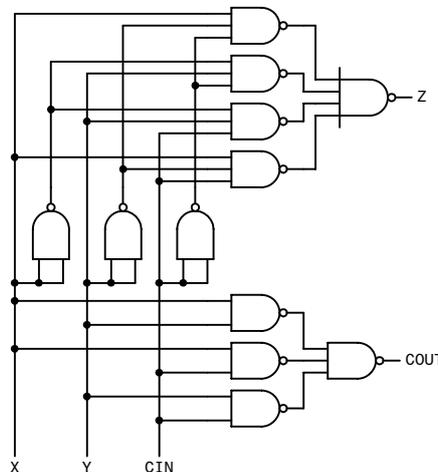
$$C_{OUT} = XY + C_{IN}(X + Y)$$

che, trasformate in somma di prodotti, diventano:

$$Z = XY\bar{C}_{IN} + \bar{X}Y\bar{C}_{IN} + \bar{X}\bar{Y}C_{IN} + XYC_{IN}$$

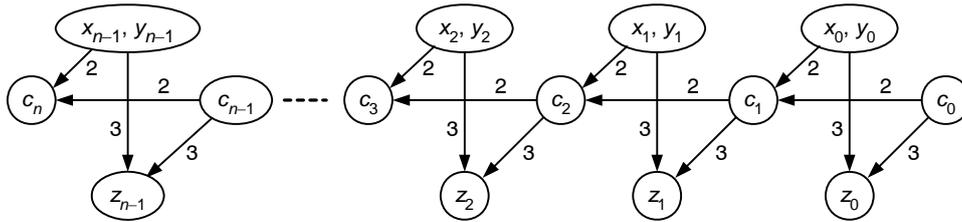
$$C_{OUT} = XY + XC_{IN} + YC_{IN}$$

La loro realizzazione in soli NAND è la seguente:

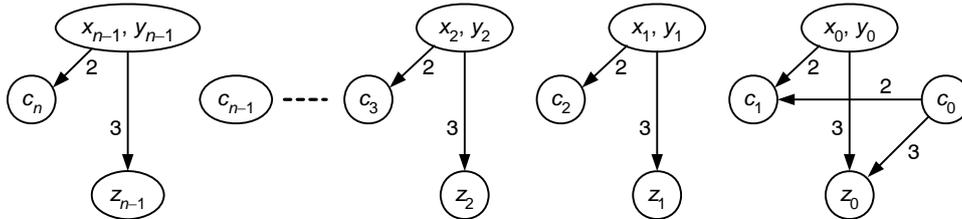


Si noti che il tempo massimo di propagazione da qualunque ingresso all'uscita Z è pari a $3t_G$, mentre il tempo di propagazione da qualunque ingresso all'uscita $COUT$ è pari a $2t_G$.

I tempi minimi e massimi per l'esecuzione di una somma vanno calcolati prendendo in considerazione tutte le possibili variazioni degli ingressi; l'analisi può comodamente essere fatta utilizzando il *grafo di propagazione*, dove le etichette sui rami orientati indicano il tempo di propagazione (in unità t_G) tra nodo e nodo:



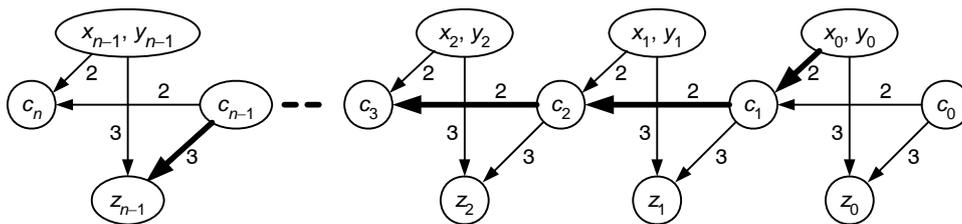
Si può osservare che, se i carry intermedi c_1, c_2, \dots, c_{n-1} non variano al variare degli ingressi, tutti i rami da essi uscenti possono essere eliminati dal grafo:



Il minimo tempo di calcolo della somma è allora dato dal minimo percorso tra ingresso ed uscita, e vale evidentemente

$$T_{\min} = 3t_G$$

Il tempo massimo, invece, è dato dal più lungo percorso tra ingresso e uscita (uno di tali percorsi è evidenziato nella figura che segue):



e richiede

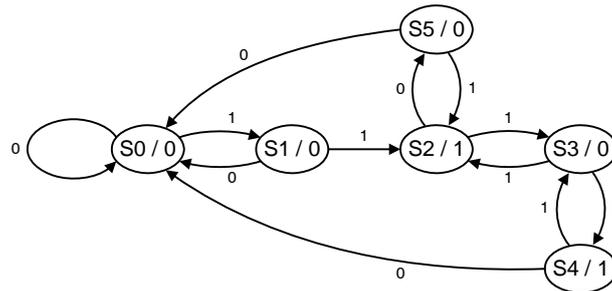
- la generazione del carry c_1 in un tempo $2t_G$
- la propagazione del carry da ciascuna cella alla successiva fino a c_{n-1} in un tempo totale pari a $2(n-2)t_G$
- la generazione del bit di somma z_{n-1} in un tempo $3t_G$

per un totale di

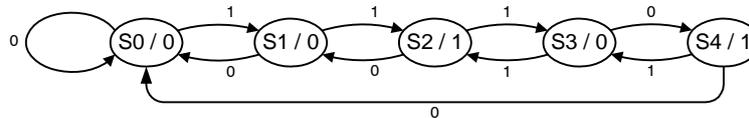
$$T_{\max} = 2t_G + 2(n-2)t_G + 3t_G = (2n+1)t_G$$

(D5) Progettare una rete sequenziale LLC dotata di un ingresso X e di un'uscita Y , asservita a un clock di periodo T , che reagisca alla sequenza di ingresso 11 con la sequenza di uscita $(10)^*$, (ossia con un'oscillazione con periodo $2T$) e torni a riposo ($Y = 0$) alla ricezione della sequenza 00.

Una macchina di Moore il cui comportamento soddisfa alle specifiche del problema è la seguente:



Si osservi tuttavia che gli stati S1 e S5, oltre ad avere identiche uscite, hanno anche identici stati successivi a parità di ingresso; dunque S1 ed S5 sono equivalenti e possono essere fusi in un unico stato. La macchina che ne deriva è allora la seguente:



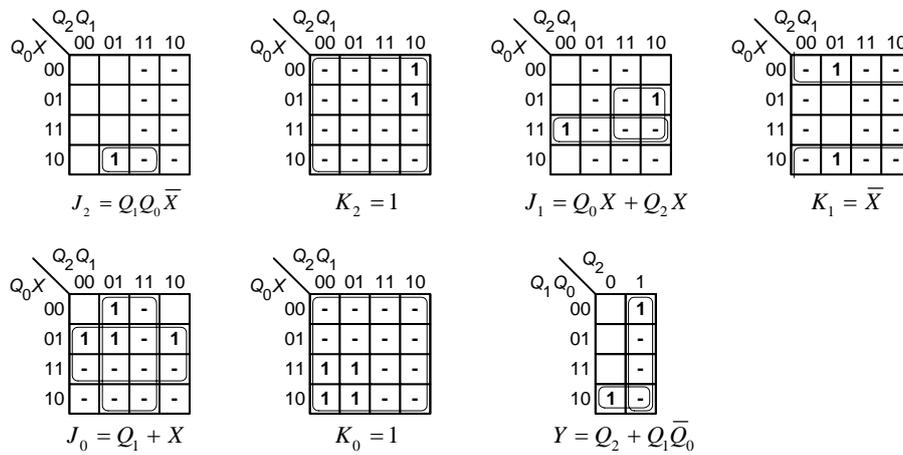
Assegnando una codifica arbitraria agli stati, costruiamo la tavola di transizione:

S	$Q_2Q_1Q_0$	X	S'	$Q'_2Q'_1Q'_0$	Y
S0	000	0	S0	000	0
		1	S1	001	0
S1	001	0	S0	000	0
		1	S2	010	1
S2	010	0	S1	001	1
		1	S3	011	0
S3	011	0	S4	100	0
		1	S2	010	1
S4	100	0	S0	000	1
		1	S3	011	0

Volendo realizzare la macchina con flip-flop JK, dalla tavola di transizione ricaviamo la tavola di eccitazione, comprensiva delle condizioni non specificate:

$Q_2Q_1Q_0$	X	$Q'_2Q'_1Q'_0$	J_2K_2	J_1K_1	J_0K_0
000	0	000	0-	0-	0-
	1	001	0-	0-	1-
001	0	000	0-	0-	-1
	1	010	0-	1-	-1
010	0	001	0-	-1	1-
	1	011	0-	-0	1-
011	0	100	1-	-1	-1
	1	010	0-	-0	-1
100	0	000	-1	0-	0-
	1	011	-1	1-	1-
101	-	---	--	--	--
110	-	---	--	--	--
111	-	---	--	--	--

Le mappe di Karnaugh per gli ingressi J, K dei flip-flop e per l'uscita Y risultano le seguenti:



Il circuito finale è il seguente:

