Reti Logiche
Appello dell'8 luglio 2008

(D2) Risolvere il sistema di equazioni booleane:

$$\begin{cases} z(\overline{x} \oplus y) + \overline{x}(y \oplus z) = \overline{x}(y \oplus z) + x\overline{y}z \\ \overline{y}(\overline{x} + z) + \overline{x}(\overline{y} + \overline{z}) = y(\overline{x} + z) + z(x + y) \\ \overline{x} \oplus y = y(x + z) + \overline{y}(x + \overline{z}) \end{cases}$$

Indicando con

$$\begin{aligned} L_1 &= z(\overline{x} \oplus y) + \overline{x}(y \oplus z) & R_1 &= \overline{x}(y \oplus z) + x\overline{y}z \\ L_2 &= \overline{y}(\overline{x} + z) + \overline{x}(\overline{y} + \overline{z}) & R_2 &= y(\overline{x} + z) + z(x + y) \\ L_3 &= \overline{x} \oplus y & R_3 &= y(x + z) + \overline{y}(x + \overline{z}) \end{aligned}$$

i membri delle singole equazioni, con

$$E_1 = \overline{L_1 \oplus R_1}$$
$$E_2 = \overline{L_2 \oplus R_2}$$
$$E_3 = \overline{L_3 \oplus R_3}$$

le funzioni corrispondenti a ciascuna equazione, e con

$$S = E_1 E_2 E_3$$

l'intero sistema di equazioni, le tavole di verità corrispondenti sono:

xyz	L_1	R_1	E_1	L_2	R_2	E_2	L_3	R_3	E_3	S
0 0 0	0	0	1	1	0	0	1	1	1	0
0 0 1	1	1	1	1	0	0	1	0	0	0
0 1 0	1	1	1	1	1	1	0	0	1	1
0 1 1	0	0	1	0	1	0	0	1	0	0
1 0 0	0	0	1	0	0	1	0	1	0	0
1 0 1	0	1	0	1	1	1	0	1	0	0
1 1 0	0	0	1	0	0	1	1	1	1	1
1 1 1	1	0	0	0	1	0	1	1	1	0

Le (due) soluzioni sono pertanto date da $y = 1,\ z = 0$.

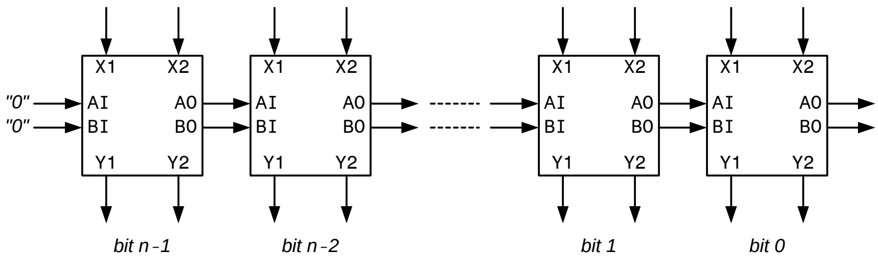
(D3) Un modulo combinatorio **S** (*sort*) accetta in ingresso due operandi interi assoluti X_1, X_2 con numero predeterminato di bit, e riemette sulle uscite Y_1, Y_2 gli stessi due operandi in modo che $Y_1 \geq Y_2$.

(a) Progettare il modulo **S** come rete combinatoria iterativa.

(b) Definire una rete combinatoria in grado di ordinare *tre* operandi, utilizzando soltanto moduli **S**.

Progetto del modulo

La rete iterativa per il sort ha la struttura:



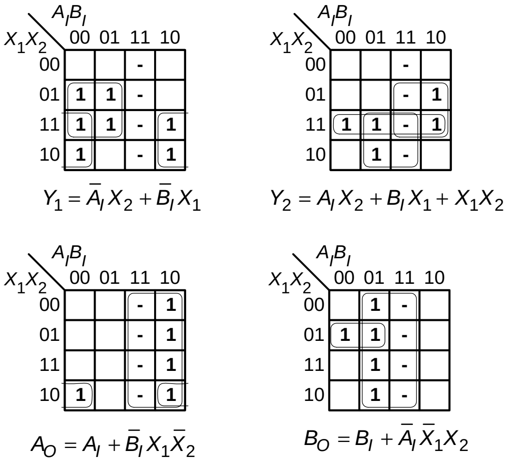
dove il significato dei segnali di stato A, B è descritto nella tavola:

A B	
0 0	*Operandi uguali, o non ordinati*
0 1	*X2 > X1 (Y1 = X2, Y2 = X1)*
1 0	*X1 > X2 (Y1 = X1, Y2 = X2)*
1 1	*non usato*

La tavola di verità del generico modulo è:

$A_I B_I X_1 X_2$	$Y_1 Y_2$	$A_O B_O$
0 0 0 0	00	00
0 0 0 1	10	01
0 0 1 0	10	10
0 0 1 1	11	00
0 1 0 0	00	01
0 1 0 1	10	01
0 1 1 0	01	01
0 1 1 1	11	01
1 0 0 0	00	10
1 0 0 1	01	10
1 0 1 0	10	10
1 0 1 1	11	10
1 1 0 0	- -	- -
1 1 0 1	- -	- -
1 1 1 0	- -	- -
1 1 1 1	- -	- -

Le mappe di Karnaugh per le quattro funzioni di uscita sono:



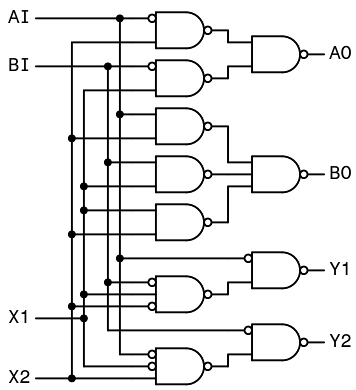
$Y_1 = \overline{A}_I X_2 + \overline{B}_I X_1$

$Y_2 = A_I X_2 + B_I X_1 + X_1 X_2$

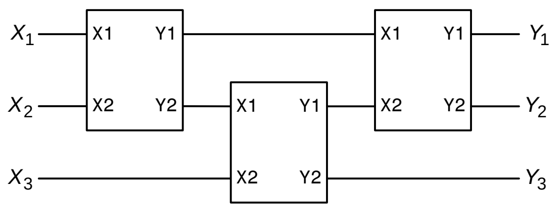
$A_O = A_I + \overline{B}_I X_1 \overline{X}_2$

$B_O = B_I + \overline{A}_I \overline{X}_1 X_2$

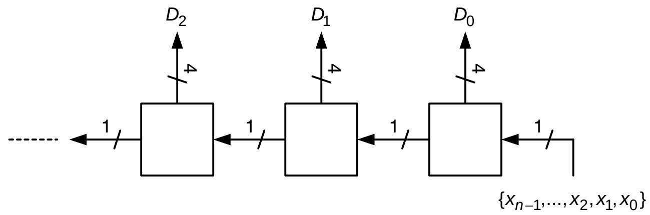
Una possibile realizzazione con Nand è:



Rete di sort



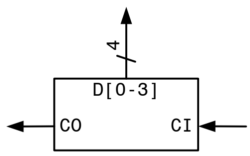
(D4) Una rete sequenziale sincrona accetta in ingresso, a partire dal più significativo, i bit $x_{n-1}...x_2x_1x_0$ di un numero intero assoluto *N*, e produce in uscita la rappresentazione BCD di *N*. La rete ha struttura modulare (cfr. figura a fianco) tale che ciascun modulo genera una singola cifra BCD del risultato. Progettare il circuito relativo al singolo modulo, e determinare il numero di moduli necessari in funzione del numero *n* di bit dell'ingresso *N*.



Algoritmo di conversione binario-BCD

$A \leftarrow 0_{10}$
for $k = n-1$ **to** 0 **step** -1
$A \leftarrow (2A + x_k)_{10}$
end for

La singola cella ha la struttura:



L'algoritmo da implementare nella cella è il seguente:

$$T = 2D + C_i$$
$$D' = T \bmod 10$$
$$C_O = \lfloor T/10 \rfloor$$

La tavola di transizione è la seguente:

D_{3210}	$C_I = 0$		$C_I = 1$	
	D'_{3210}	C_O	D'_{3210}	C_O
0000	0000	0	0001	0
0001	0010	0	0011	0
0010	0100	0	0101	0
0011	0110	0	0111	0
0100	1000	0	1001	0
0101	0000	1	0001	1
0110	0010	1	0011	1
0111	0100	1	0101	1
1000	0110	1	0111	1
1001	1000	1	1001	1
1010	- - - -	-	- - - -	-
1011	- - - -	-	- - - -	-
1100	- - - -	-	- - - -	-
1101	- - - -	-	- - - -	-
1110	- - - -	-	- - - -	-
1111	- - - -	-	- - - -	-

Si noti come C_O non dipenda da C_I ma solo dallo stato interno (D_{3210}), dunque la macchina sequenziale all'interno della singola cella può essere implementata come macchina di Moore. Inoltre, solo D_0' dipende da C_I, essendo addirittura $D_0' = C_I$.

Tavole di eccitazione (realizzazione con flip-flop D):

D_1D_0 \ D_3D_2	00	01	11	10
00		1	-	
01			-	1
11			-	-
10			-	-

$$D_3' = D_2\overline{D}_1\overline{D}_0 + D_3D_0$$

D_1D_0 \ D_3D_2	00	01	11	10
00			-	1
01			-	
11	1	1	-	-
10	1		-	-

$$D_2' = D_3\overline{D}_0 + D_1D_0 + \overline{D}_2D_1$$

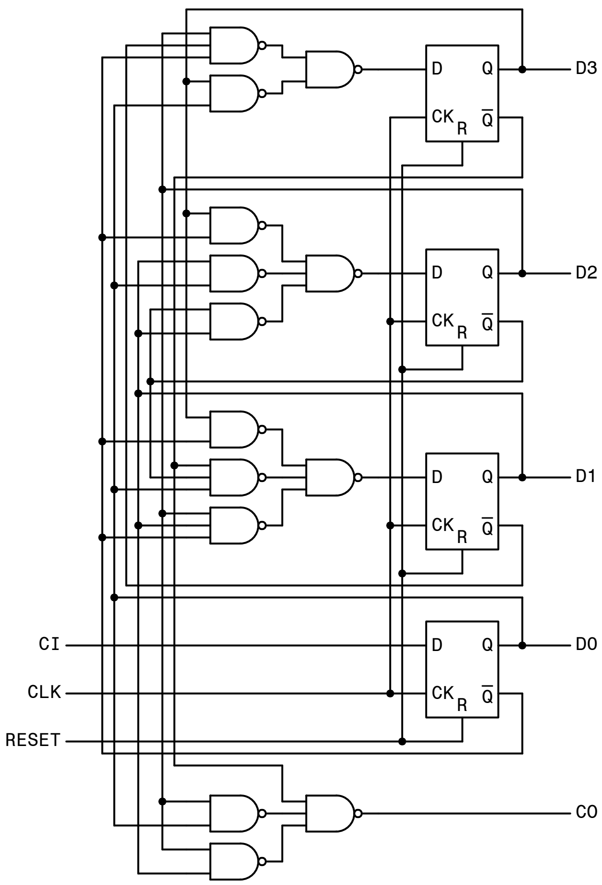
D_1D_0 \ D_3D_2	00	01	11	10
00			-	1
01	1		-	
11	1		-	-
10		1	-	-

$$D_1' = D_3\overline{D}_0 + \overline{D}_3\overline{D}_2D_0 + D_2D_1\overline{D}_0$$

D_1D_0 \ D_3D_2	00	01	11	10
00			-	1
01		1	-	1
11		1	-	-
10		1	-	-

$$C_O = D_3 + D_2D_0 + D_2D_1$$

Realizzazione:



Dimensionamento della rete

Il massimo numero esprimibile in binario su *n* bit è $2^n - 1$, mentre il massimo numero esprimibile su *m* cifre decimali è $10^m - 1$; dovrà dunque essere $2^n - 1 \leq 10^m - 1$, ossia $2^n \leq 10^m$, da cui, detto $\lceil x \rceil$ il più piccolo intero non minore di *x*, si deduce

$$m = \left\lceil \log_{10} 2^n \right\rceil = \lceil n \log_{10} 2 \rceil$$