

- Esercizio 6.1:** Progettare un circuito combinatorio che a partire da un numero BCD, piloti un display a 7 segmenti per fornirne una rappresentazione numerica. Un display a sette segmenti è caratterizzato come rappresentato in figura. Un segmento è acceso se la linea che la controlla è al valore logico 1 e spento viceversa.

a	b	c	d	e	f	g	Output
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
0	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9

Sono presenti altre 54 possibili combinazioni “don’t care” che producono una visualizzazione errata nel display, cioè hanno un output che non è intelligibile come uno dei valori numerici sopra.

Per esempio, la seguente combinazione non “accende” alcun segmento:

a	b	c	d	e	f	g
0	0	0	0	0	0	0

Mentre la seguente combinazione produce un’uscita non valida

a	b	c	d	e	f	g
1	1	0	0	0	1	1



Per progettare il decoder lavoriamo sull’input fornito dai numeri scritti in binario, per la cui rappresentazione abbiamo dunque bisogno di 4 bit, ma volendo rendere più efficace il decoder in modo che non dia risultati ambigui (come i due sopra) è necessario un “validator”:

w	x	y	z	a	b	c	d	e	f	g	validator
0	0	0	0	1	1	1	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	1
0	0	1	1	1	1	1	1	0	0	1	1
0	1	0	0	0	1	1	0	0	1	1	1
0	1	0	1	1	0	1	1	0	1	1	1
0	1	1	0	0	0	1	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1
1	0	1	0	X	X	X	X	X	X	X	0
1	0	1	1	X	X	X	X	X	X	X	0
1	1	0	0	X	X	X	X	X	X	X	0
1	1	0	1	X	X	X	X	X	X	X	0
1	1	1	0	X	X	X	X	X	X	X	0
1	1	1	1	X	X	X	X	X	X	X	0

Le combinazioni dei 4 bit che rappresentano i numeri decimali da 10 a 15 sono le combinazioni don’t care del nostro decoder.

wx\yz	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	X	X	X	X
10	1	1	X	X

$$a = w + yz + x'y + xz + x'z'$$

wx\yz	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	X	X	X	X
10	1	1	X	X

$$b = x' + yz + y'z'$$

wx\yz	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

$$c = z + x + y'$$

wx\yz	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	X	X	X	X
10	1	1	X	X

$$d = w + yz' + x'y + x'z' + xy'z$$

wx\yz	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	X	X

$$e = yz' + x'z'$$

wx\yz	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

$$f = w + xz' + xy' + y'z'$$

wx\yz	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

$$g = w + yz' + x'y + xz' + xy'$$

wx\yz	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	1	1	0	0

$$\text{validator} = w' + x'y'$$

Alla fine della semplificazione delle mappe di Karnaugh, avremo:

$$\begin{aligned}
 a &= w + yz + x'y + xz + x'z' \\
 b &= x' + yz + y'z' \\
 c &= z + x + y' \\
 d &= w + yz' + x'y + x'z' + xy'z \\
 e &= yz' + x'z' \\
 f &= w + xz' + xy' + y'z' \\
 g &= w + yz' + x'y + xz' + xy' \\
 \text{validator} &= w' + x'y'
 \end{aligned}$$

Il *validator* sarà utilizzato per un buffer tri-state di ciascuna delle uscite a , b , c , d , e ed f in maniera che il decoder non visualizzerà nulla sul Display se non è stata immessa una combinazione BCD valida.

Per semplificare la realizzazione con PSpice del progetto conviene cercare di ottenere più componenti simili nelle varie funzioni.

Sicuramente l'utilizzo dei colori apparirà più che altro confusionale, ma sarebbe utile cercare di non creare doppi e utilizzare il minor numero possibile di porte (anche per risparmiare).

Si potrebbe ancora effettuare qualche modifica algebrica, come sostituire in b $(x+y')$ con $\text{NOT}(x'y)$ in maniera da poter utilizzare $(x'y)$ di g con la semplice aggiunta di un inverter.

