Logica e Reti Logiche
(Episodio 13: Blocchi funzionali combinatori)

Francesco Pasquale

5 dicembre 2024

Nell'Episodio 11 abbiamo visto come costruire un piccolo circuito, il Full-Adder, che esegue una semplice *funzione*: prende in input tre bit (due bit più un bit di riporto "in entrata") e restituisce in output due bit: la *somma* e il *riporto* "in uscita". Abbiamo visto come questo semplice *blocchetto funzionale* può essere poi messo in serie per ottenere, per esempio, un circuito che esegue la somma di numeri a 32 bit.

In questo episodio costruiamo due *blocchi funzionali*, Decoder e Multiplexer, che sono così generali da poter essere usati in diverse circostanze. Per esempio, se presi della "taglia" opportuna, possono essere usati per implementare qualunque funzione Booleana.

1 Decoder

Definizione 1.1 (Decoder)**.** Un Decoder $n{:}2^n$ è un circuito con n input, diciamo $x_0, x_1, \ldots, x_{n-1}$, e 2^n output, diciamo $y_0, y_1, \ldots, y_{2^n-1}$, tale che per ogni $i = 0, 1, \ldots, 2^n - 1$

$$y_i = \begin{cases} 1 & \text{se } (i)_{10} = (x_{n-1} \cdots x_1 x_0)_2 \\ 0 & \text{altrimenti} \end{cases}$$

In altre parole il circuito *decodifica* la sequenza di bit in input ponendo a 1 l'output y_i il cui indice $i \in \{0, 1, \ldots, 2^n - 1\}$ è il numero intero rappresentato in binario dagli n bit in input. Per esempio, se i due bit in input di un Decoder 2:4 sono $(x_1, x_0) = (1, 0)$, i quattro bit in output devono essere $(y_3, y_2, y_1, y_0) = (0, 1, 0, 0)$.

Si noti che in un decoder, qualunque sia la sequenza di bit in input, uno e uno solo degli ouput avrà valore 1.

Come costruiamo un tale circuito? Vediamo.

Se $n = 1$ abbiamo un solo input x_0 e due output y_1, y_0; y_0 deve essere 1 quando $x_0 = 0$ mentre y_1 deve essere 1 quando $x_0 = 1$:

x_0	y_1	y_0
0	0	1
1	1	0

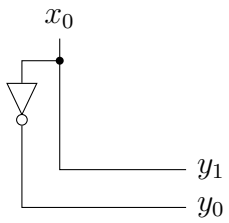


Figura 1: DECODER 1:2

Se $n = 2$ abbiamo due input x_0, x_1 e quattro output y_3, y_2, y_1, y_0

x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

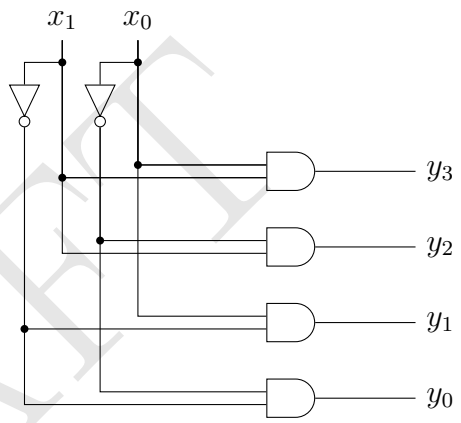


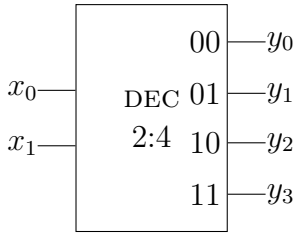
Figura 2: DECODER 2:4

A questo punto dovrebbe essere chiaro come la costruzione si generalizza a un DECODER n:2^n per qualunque n.

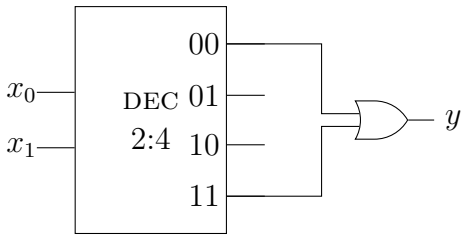
Esercizio 1. Progettare un DECODER 3:8 e disegnarne lo schema.

Esercizio 2. Quante porte logiche servono per costruire un DECODER n:2^n?

Una volta costruito il nostro *blocco funzionale* DECODER, possiamo immaginare di averlo a disposizione come le porte logiche AND, OR, e NOT e usarlo per costruire circuiti più complessi. A questo scopo, possiamo disegnare un DECODER 2:4 in questo modo.



Si osservi che se abbiamo un DECODER n:2^n e una porta OR (con abbastanza ingressi) possiamo implementare qualunque funzione Booleana di n variabili. Per esempio, $y = x_0x_1 + \bar{x}_0\bar{x}_1$ si può implementare così



Esercizio 3. Costruire un circuito che implementi la seguente tabella di verità

p	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
q	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
r	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
s	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
?	0	0	0	1	1	0	1	1	0	0	0	0	1	0	1	0

Esercizio 4. Progettare un circuito che prenda in input 4 bit, che rappresentano un numero in codifica binaria, e restituisca 1 se il numero in input è divisibile per *tre* e 0 altrimenti.

Esercizio 5. Progettare un "*encoder*", ossia un circuito che esegue l'operazione inversa del decoder: un ENCODER 2^n:n prende in input 2^n bit, $x_0, \ldots, x_{2^n-1}$, di cui assumiamo che uno solo sia 1 e tutti gli altri siano 0, e restituisce in output n bit, $y_0, \ldots y_{n-1}$, che rappresentano la codifica binaria dell'indice $i \in \{0, \ldots, 2^n - 1\}$ tale che $x_i = 1$.

2 Multiplexer

Definizione 2.1 (Multiplexer)**.** Un MULTIPLEXER 2^n:1 è un circuito con $2^n + n$ input, diciamo $x_0, x_1, \ldots, x_{2^n} - 1$ e $s_0, \ldots, s_{n-1}$, e un output, y, tale che

$$y = x_{(s_{n-1} \ldots s_1, s_0)_2}$$

In altre parole, fra le variabili in input $x_0, \ldots, x_{2^n-1}$, il circuito dà in output il bit contenuto nella variabile il cui indice è codificato in binario dalle variabili di *selezione* $s_0, \ldots s_{n-1}$. Per esempio, in un MULTIPLEXER 4:1, se gli input sono $(x_3, x_2, x_1, x_0) = (0, 1, 1, 0)$ e $(s_1, s_0) = (1, 0)$, il circuito darà in output $y = 1$, perche $(s_1 s_0)_2 = (10)_2 = (2)_{10}$ e $x_2 = 1$. Come costruiamo un circuito che risponda a queste specifiche?

Se $n = 1$, abbiamo soltanto due variabili in input x_1, x_0 e una variabile di selezione s_0. Chiaramente possiamo procedere come abbiamo fatto per il decoder e scrivere la tabella di verità con i tre input e un output e poi costruire il circuito.

Esercizio 6. Scrivere la tabella di verità di un MULTIPLEXER 2:1 e progettare il circuito.

Possiamo anche ragionare più ad altro livello e osservare che, quando $s_0 = 0$ dobbiamo avere $y = x_0$ e quando $s_0 = 1$ dobbiamo avere $y = x_1$. Quindi possiamo costruire il circuito così

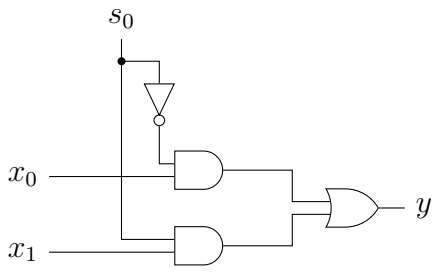


Figura 3: MULTIPLEXER 2:1

Si noti che il multiplexer implementa un "if": `if` s_0 `then` x_1 `else` x_0.

Esercizio 7. Costruire un MULTIPLEXER 2:1, usando un DECODER 1:2 due porte AND e una porta OR.

Esercizio 8. Costruire un MULTIPLEXER 4:1 usando solo porte logiche elementari, ragionando come nel caso 2:1. Quante porte AND usate?

Possiamo costruire un MULTIPLEXER 4:1 anche usando un DECODER 2:4, quattro porte AND e una porta OR, come in Figura 4

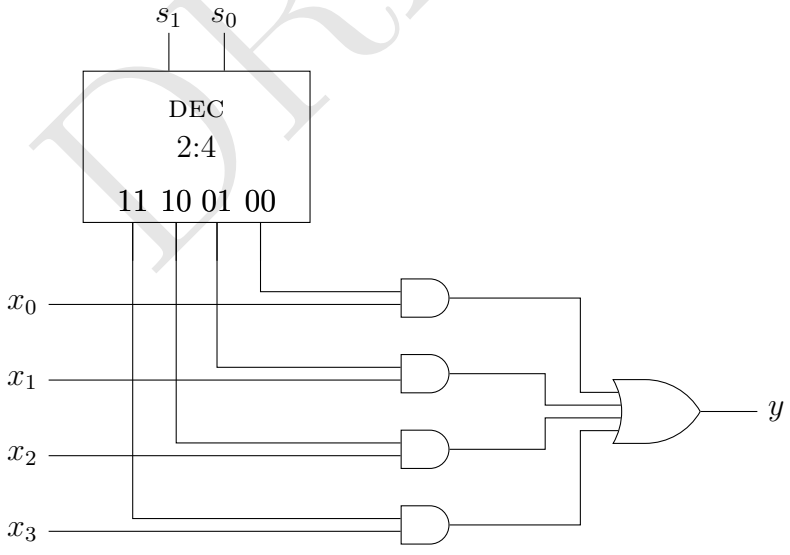
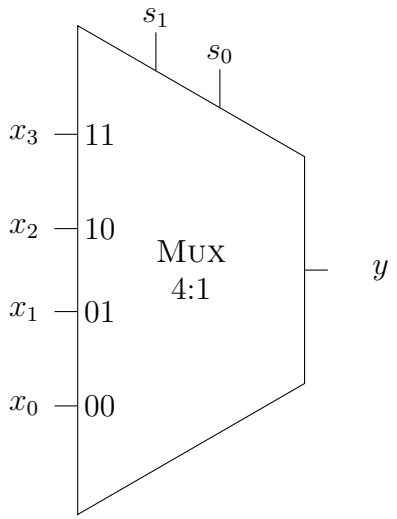


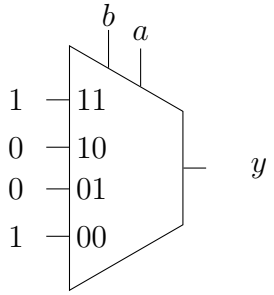
Figura 4: MULTIPLEXER 4:1

Il simbolo con cui in genere si indica un multiplexer è questo qui



Con un MULTIPLEXER 2^n:1 si può implementare una qualunque funzione con n variabili, associando agli input di selezione $s_0, \ldots, s_{n-1}$ del multiplexer le variabili della funzione, e agli input $x_0, \ldots, x_{2^n-1}$ del multiplexer i valori Booleani che la funzione assume nelle righe corrispondenti della tabella di verità. Per esempio, la funzione $y = ab + \bar{a}\bar{b}$ si può implementare così

a	b	y
0	0	1
0	1	0
1	0	0
1	1	1



Esercizio 9. Implementare con un multiplexer la tabella di verità dell'Esercizio 3.

Esercizio 10. Costruire un circuito che implementi la seguente funzione Booleana $f : \{0,1\}^3 \to \{0,1\}$

$$f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$$

Esercizio 11. Costruire un MULTIPLEXER 8:1 usando due MULTIPLEXER 4:1 e un MULTIPLEXER 2:1.

Esercizio 12. Si consideri la seguente funzione Booleana:

$$y = bc + \overline{a}\overline{b}\overline{c} + b\overline{c} \tag{1}$$

1. Implementare la funzione in (1) usando soltanto un MULTIPLEXER 4:1
2. Implementare la funzione in (1) usando un MULTIPLEXER 2:1, una porta OR e una porta NOT.