

Componenti combinatori speciali

Architettura degli elaboratori

M. Favalli



Engineering Department in Ferrara

Decoder

Multiplexer

- Si é descritto un approccio top-down al progetto di reti combinatorie
- Alcune funzioni sono di utilizzo talmente comune da essere state inserite in librerie di progetto utilizzabili in maniera bottom-up
- Vedremo alcuni fra i piú rilevanti di questi componenti

Decoder

Multiplexer

Componente con n ingressi $\{x_{n-1}, x_{n-2}, \dots, x_0\}$ e 2^n uscite $\{y_{2^n-1}, y_{2^n-2}, \dots, y_0\}$

Si ha $y_k = 1$ se $k = \sum_{i=0}^{n-1} x_i 2^i$ e 0 altrimenti (qui la somma é quella aritmetica)

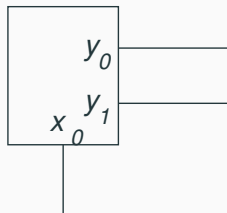
In pratica l'uscita il cui indice é codificato come numero binario dalla configurazione in ingresso si porta a 1

Quindi viene prodotto in uscita un codice detto del tipo 1-out-of- n , ovvero le uscite sono tutte a 0 salvo una

Decoder

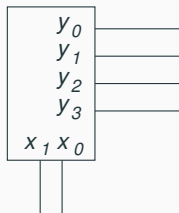
$n = 1$

x_0	y_1	y_0	Equazioni
0	0	1	$y_0 = x'_0$
1	1	0	$y_1 = x_0$



$n = 2$

x_1, x_0	y_3	y_2	y_1	y_0	Equazioni
00	0	0	0	1	$y_0 = x'_1 x'_0$
01	0	0	1	0	$y_1 = x'_1 x_0$
10	0	1	0	0	$y_2 = x_1 x'_0$
11	1	0	0	0	$y_3 = x_1 x_0$

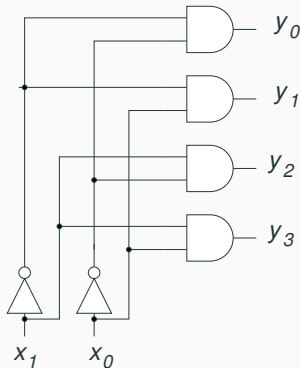


Struttura a livello gate

Si tratta di una rete a un livello

Il costo di un decoder non é trascurabile: $l = n2^n$

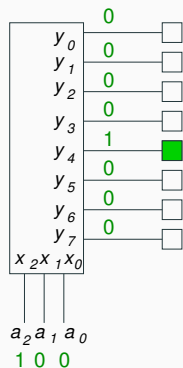
Si vedranno in seguito alcuni accorgimenti per ridurlo



Applicazioni

I decoder possono essere utilizzati per produrre, sulla base del valore di x , un segnale di abilitazione per uno di 2^n oggetti diversi (ad esempio celle di memoria)

In questi casi x assume il significato di un indirizzo

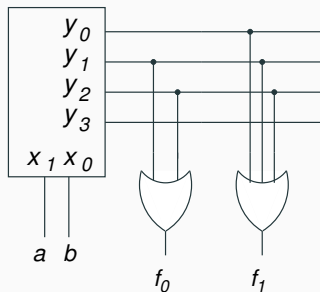


Sintesi di forme canoniche SP mediante decoder

- Un decoder mette a disposizione tutti i possibili termini prodotto corrispondenti alle configurazioni di $\{x_0, x_1, \dots, x_{n-1}\}$
- Lo si può quindi utilizzare per realizzare la forma canonica SP qualsiasi funzione f_i di n variabili
- Questo può essere fatto semplicemente connettendo a una porta logica OR le uscite del decoder corrispondenti a mintermini di f_i

Sintesi di forme canoniche SP mediante decoder: esempio

Esempio: $f_0 = ab' + a'b$ e $f_1 = (ab)' = a'b' + ab' + a'b$



Sintesi di forme canoniche SP mediante decoder: ROM

- La memoria a sola lettura può essere vista come modello computazionale di una rete combinatoria
 - data $f : \{0, 1\}^n \Rightarrow \{0, 1\}^m$, si possono interpretare gli ingressi come indirizzi di una memoria che contiene parole di dimensione m
 - per una configurazione i degli ingressi, la configurazione delle uscite $[f_0(i), f_1(i), \dots, f_{m-1}(i)]$ può essere interpretata come $mem[i]$
- Le memorie ROM (Read Only Memory) sono realizzate (dal punto di vista logico) proprio con la sintesi di forme canoniche SP mediante decoder

Decoder con segnale di abilitazione

- Un segnale di *enable* (en) può essere messo in prodotto logico con ciascuna uscita

- Esempio ($n = 2$):

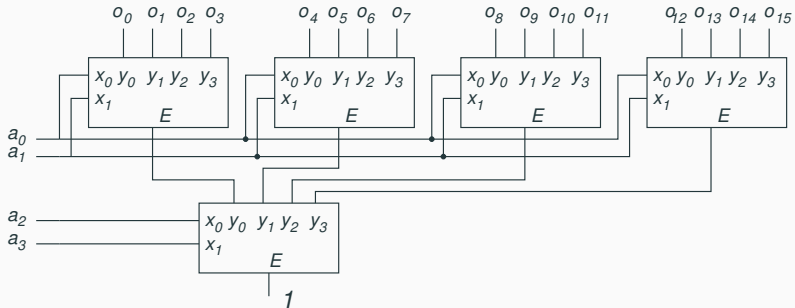
$$y_0 = x_1' x_0' en, \quad y_1 = x_1' x_0 en, \quad y_2 = x_1 x_0' en, \quad y_3 = x_1 x_0 en$$

- Consente di mettere tutte le uscite a 0
- Questo consente la connessione gerarchica di più decoder con un numero di ingressi minore di n per formare un unico decoder a n ingressi

- Consideriamo per semplicità il caso a 2 livelli e si supponga che siano disponibili decoder con k e j ingressi ($k + j = n$)
- Il decoder può essere formato con:
 - un decoder a k ingressi le cui uscite forniscono i segnali di enable a 2^k decoder a j ingressi
 - il primo decoder riceve in ingresso i primi k ingressi e i secondi i rimanenti $n - k = j$
 - costo $l = (k + 1)2^k + 2^k(j + 1)2^j = 2^k(k + 1 + (j + 1)2^k)$
 - può anche mettere al primo livello un decoder da j ingressi e 2^j decoder a k ingressi al secondo livello

Decodifica multilivello

Esempio: $n = 4$, $k = j = 2$



Il costo di questa realizzazione é pari a $l = 12 + 4 * 12 = 60$, il costo di un singolo decoder a 4 ingressi é $l = 64$

Il guadagno aumenta al crescere di n

1. **Si realizzi un decoder a 5 ingressi disponendo di decoder a 2 e a 3 ingressi, si confronti il costo delle due alternative possibili**

Decoder

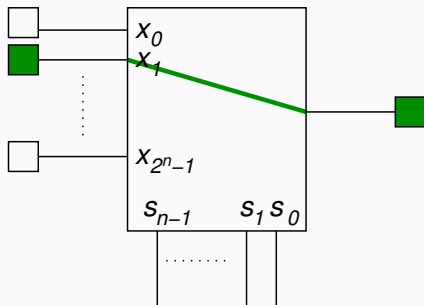
Multiplexer

Multiplexer

- Il multiplexer é un componente che ha $2^n + n$ ingressi partizionati fra:
 - ingressi dati $\{x_0, x_1, \dots, x_{2^n-1}\}$
 - ingressi di selezione $\{s_0, s_1, \dots, s_{n-1}\}$
- L'uscita é data da $y = x_i \mid i = \sum_{j=0}^{n-1} s_j 2^j$ (la somma é quella aritmetica)
- Tradotto in un espressione booleana si ha $y = \sum_{i=0}^{2^n-1} p_i x_i$ (ove p_i é il termine prodotto corrispondente alla configurazione i e la somma é quella logica)

Il multiplexer come blocco di selezione

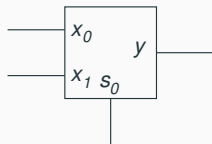
Il multiplexer può essere visto come un componente che riporta in uscita il valore dell'ingresso dati selezionato dagli ingressi di selezione



Esempi

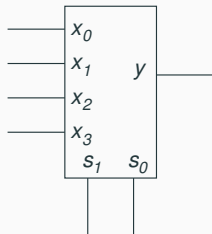
$$n = 1$$

$$y = x_0 s'_0 + x_1 s_0$$



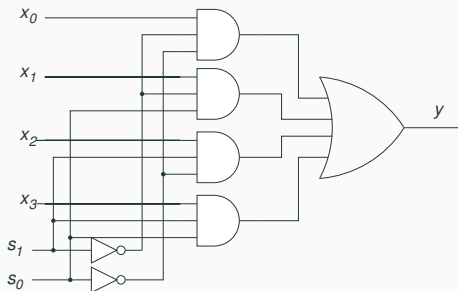
$$n = 2$$

$$y = x_0 s'_1 s'_0 + x_1 s'_1 s_0 + x_2 s_1 s'_0 + x_3 s_1 s_0$$

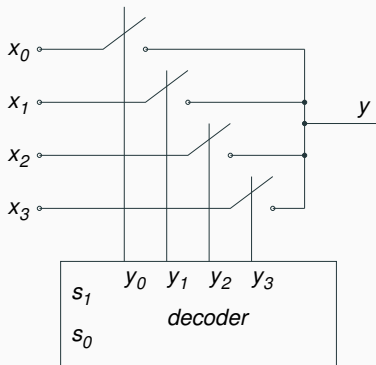


Realizzazioni al livello gate e al livello switch

Gate



Switch



Il ruolo del MPX nella sintesi

- Si supponga di considerare una generica funzione in cui nella tabella della verità non sono riportati zeri e uni, ma il valore $f(i)$ di ciascuna riga i
- Possiamo generalizzare quanto visto per la forma canonica SP e scrivere

$$f(a_0, a_1, a_2, \dots, a_{n-1}) = \sum_{i=0}^{2^n-1} p_i f(i)$$

ove p_i é il termine prodotto di n variabili corrispondente alla configurazione i

- **proposizione:** f vale 1 se sono nella configurazione i -ma e $f(i) = 1$
- Come si può osservare, questa é l'equazione di un MPX con i valori di $f(i)$ come ingressi

Il ruolo del MPX nella sintesi

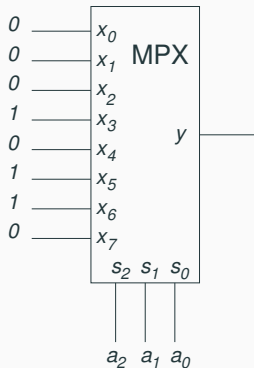
- f può essere implementata con un MPX in cui gli ingressi di selezione sono connessi ai segnali (a_i) corrispondenti alle variabili della funzione e gli ingressi dati sono connessi ai valori di $f(i)$
- Il MPX é quindi in grado di realizzare una qualsiasi funzione di n variabili
- É in effetti una rete programmabile che viene usata nelle FPGA per avere celle programmabili, ovvero celle che sono in grado di realizzare una qualsiasi funzione di 4 o 5 variabili

Esempio

Si realizzi con un MPX una funzione che vale 1 quando 2 dei suoi 3 ingressi sono a 1

Tabella di verità

i	$a_2 a_1 a_0$	f
0	000	0
1	001	0
2	010	0
3	011	1
4	100	0
5	101	1
6	110	1
7	111	0



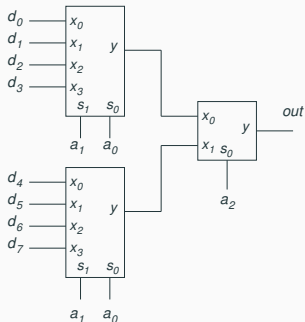
Realizzazione gerarchica di multiplexer

- É possibile realizzare MPX con n bit di selezione (a_{n-1}, \dots, a_0) e 2^n ingressi dati (d_0, \dots, d_{2^n-1}) utilizzando MPX con un numero di bit di selezione $k < n$
- Supponiamo di disporre di MPX a k e j ingressi di selezione ($k + j = n$)
- Si può realizzare un MPX a due livelli nel seguente modo:
 - si dispone un primo livello di 2^k MPX a j bit di selezione ciascuno
 - agli ingressi dati di questi MPX si connettono ordinatamente i 2^n ingressi dati ($2^k 2^j = 2^{k+j} = 2^n$)
 - i j bit di minor peso degli ingressi di selezione (a_{j-1}, \dots, a_0) vanno invece connessi agli ingressi di selezione di tali MPX
 - le uscite di tali MPX vengono connesse ai 2^k ingressi dati del MPX che realizza l'uscita i cui bit di selezione (locali) sono connessi ordinatamente ai rimanenti k bit di selezione (a_{n-1}, \dots, a_j)

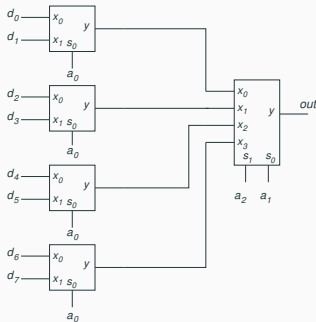
Esempio

Si realizzi un MPX a 8 ingressi dati e 3 bit di selezione disponendo di MPX a 2 ingressi dati e 1 bit di selezione e di MPX a 4 ingressi dati e 2 bit di selezione

$$j = 2 \quad k = 1$$



$$j = 1 \quad k = 2$$



- **Si sono visti due componenti, il decoder e il multiplexer, che sono ampiamente utilizzati nel progetto dei sistemi digitali**
- **Si sono viste alcune applicazioni rilevanti come la ROM o le celle usate negli FPGA**