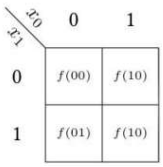
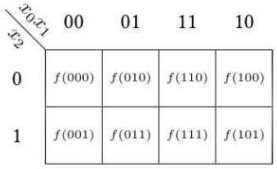
Lezione 9 Algebra Booleana 3 + Circuiti Combinatori

martedì 17 ottobre 2023 14:50

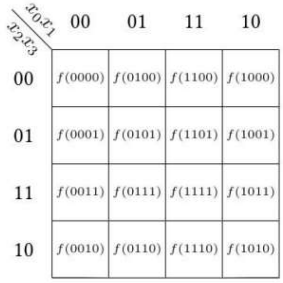
Andiamo a vedere ora in dettaglio il funzionamento delle mappe Karnaugh (mappe k): innanzitutto diciamo che è un'ulteriore rappresentazione delle tabelle di verità , le quali sono organizzate in rettangoli o quadrati a seconda del numero dei variabili che compaiono nella tabella. Da notare che le celle della tabella contigue sono caratterizzate da Codice di Gray , quindi da distanza di Humming pari ad 1 : tra celle contigue cambia al massimo 1 singolo bit .
Se tra due celle contigue il valore di una variabile cambia, viene ignorata in base alla legge della somma tra una variabile ed il suo negato che vale 1 . Con questo metodo è possibile semplificare funzioni fino a 6 variabili . Andiamo a vedere in generale :



2 variabili

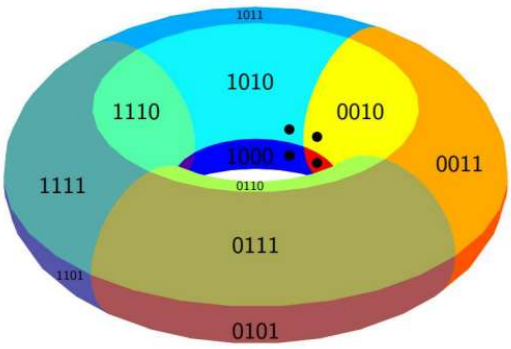


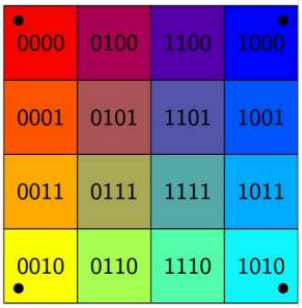
3 variabili



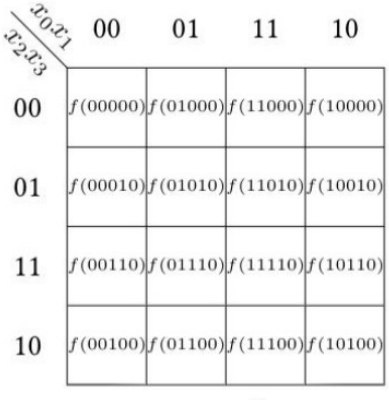
4 variabili

Ricordiamo della distanza di humming pari ad 1 .
Piccola osservazione : nel caso di tabella k a 4 variabili (16 combinazioni) , la si può pensare come uno sviluppo toroidale :

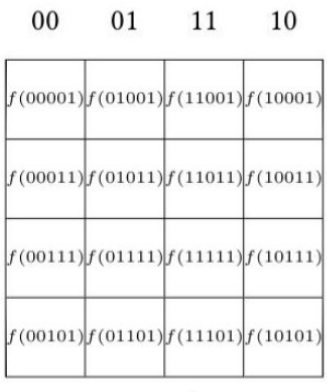




Nel caso invece di funzione a 5 variabili : si usano 2 tabelle 4*4 , ma con una particolarità : nella prima tabella si considera la quinta variabile = 0 , mentre nella seconda tabella la si considera = 1 : in questo caso l'adiacenza si vede anche facendo la sovrapposizione delle tabelle :

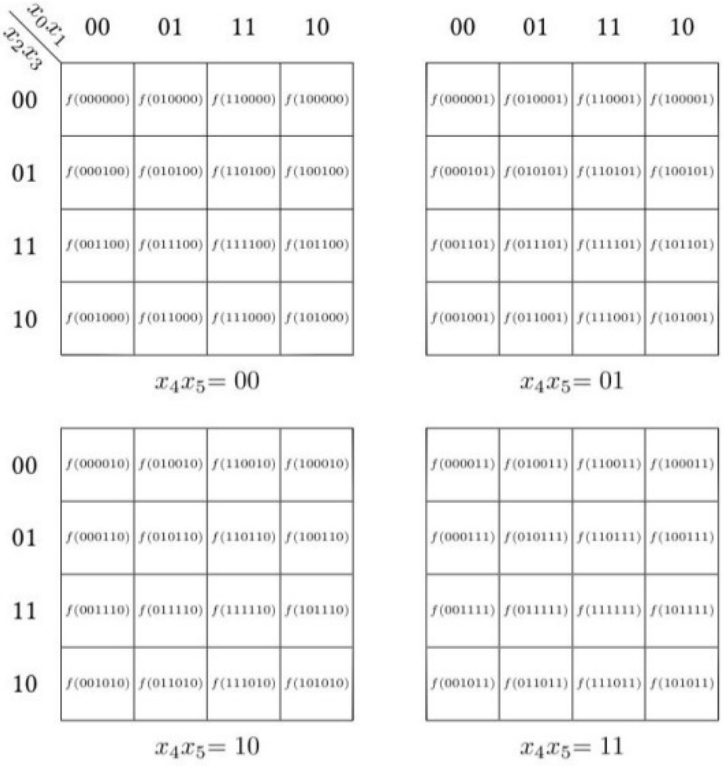


$x_4 = 0$



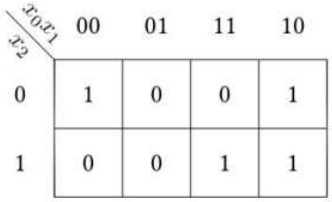
$x_4 = 1$

Nel caso di funzione a 6 variabili , il procedimento è analogo : solo che le tabelle ora diventano 4 in più : per ognuna delle due variabili in più, si usano le opportune configurazioni (00,01,11,10) ed oltre alla sovrapposizione delle tabelle , si usa anche la sovrapposizione ad "L" :

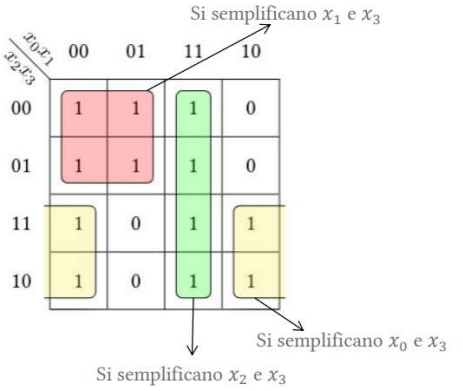


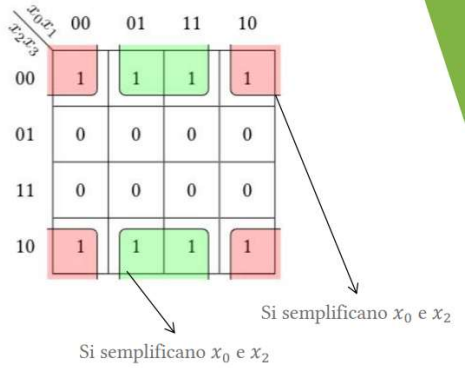
Vediamo ora un esempio :

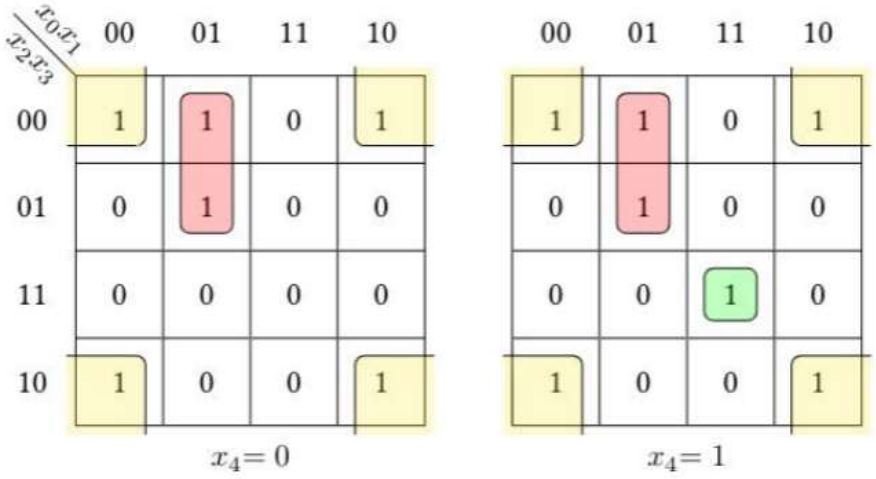
k	x_1	x_2	x_3	$f(k)$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1



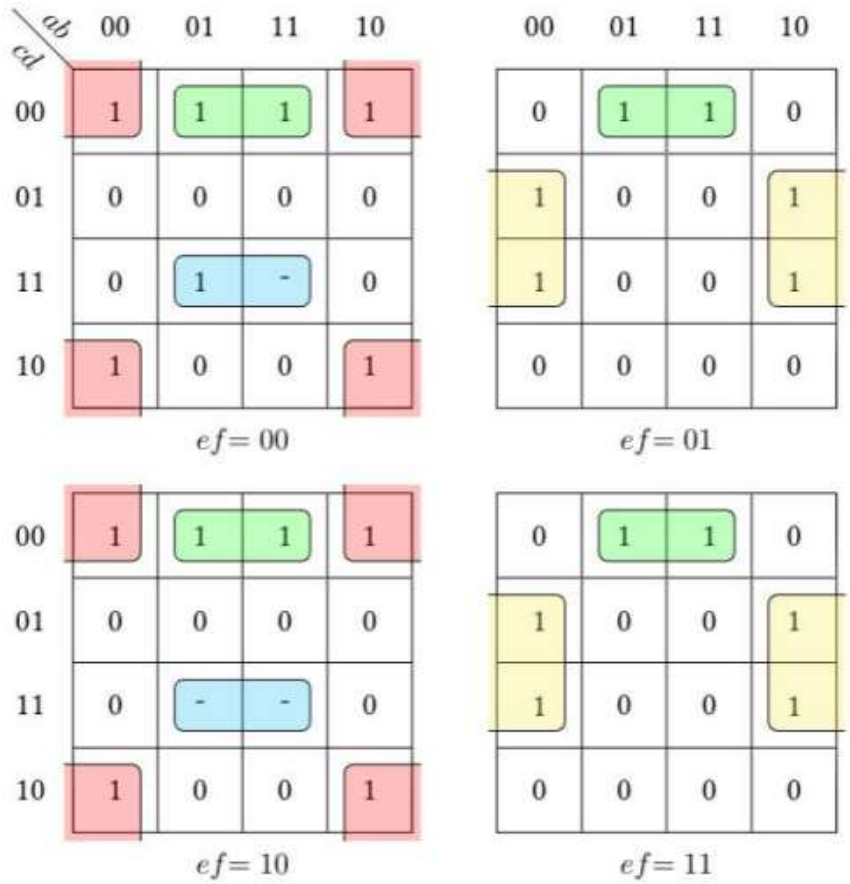
Per costruire gli insiemi di copertura (maxtermini =0 e mintermini=1) si raggruppa a seconda della rappresentazione in gruppi di potenze di 2 (2,4,8,16, …) , avendo cosi' due possibili rappresentazioni : SOP o POS . **Da notare che il raggruppamento dei termini non è univoco.**
Andiamo a vedere degli esempi a 4,5,6 variabili :







Da notare che qui il mintermine in verde deriva dalla sovrapposizione delle tabelle .



Idem per quanto la sovrapposizione , inoltre da notare che l'insieme in blu ha due trattini , ovvero don't care conditions : configurazioni nelle quali la funzione non si portà mai trovare , quindi a seconda del caso vengono considerati mintermini o maxtermini .

Tornando agli operatori universali : sono necessari tutti e tre ? AND , OR , NOT?? assolutamente no, visto che grazie a De Morgan ed una doppia negazione uno si leva ! :

$$\text{Esempio: } f(x,y,z) = x + yz + \overline{x}\,\overline{z} = \overline{\overline{f}}(x,y,z) = \overline{\overline{x} \cdot \overline{yz} \cdot \overline{\overline{y}\,\overline{z}}}$$

Riducendo ulteriormente : usiamo una sola porta logica per esprimere tutta l'algebra booleana **: la porta NAND :**

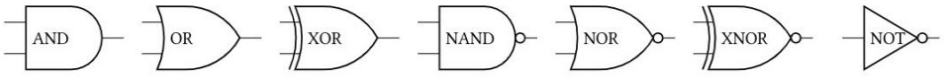
- $a|a = \overline{a}$
- $(a|a)|(b|b) = \overline{\overline{a} \cdot \overline{b}} = a + b$
- $(a|b)|(a|b) = \overline{\overline{(a \cdot b)} \cdot \overline{(a \cdot b)}} = (a \cdot b) + (a \cdot b) = a \cdot b$
- $a|(a|a) = \overline{a \cdot \overline{a}} = a + \overline{a} = 1$
- $(a|(a|a))|(a|(a|a)) = 1|1 = 0$

Dualmente alla NAND , vi è la NOR :

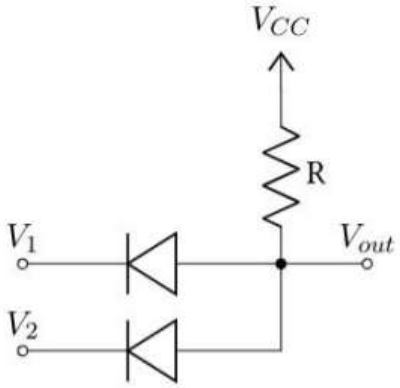
- $a \downarrow a = \overline{a}$
- $(a \downarrow a) \downarrow (b \downarrow b) = \overline{\overline{a} + \overline{b}} = a \cdot b$
- $(a \downarrow b) \downarrow (a \downarrow b) = \overline{\overline{(a + b)}} = a + b$
- $a \downarrow (a \downarrow a) = 0$
- $(a \downarrow (a \downarrow a)) \downarrow (a \downarrow (a \downarrow a)) = 1$

Di solito si usano questi due operatori per via del numero in costo di componenti, rispetto alle porte non negate (OR / AND) .

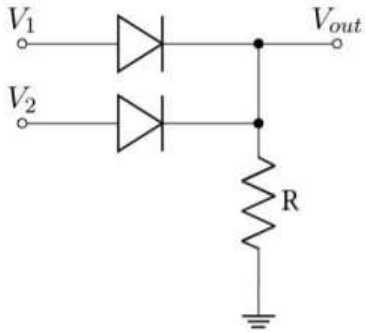
Quindi si parla di reti che accettano input variabile booleane e restituiscono variabili booleane , le quali reti vengono sintetizzate dalle solite regole dell'algebra booleana : in genere si usano delle porte logiche per fare questa sintesi :



Vediamo un pò di storia : queste porte inizialmente furono create ed implementate mediante diodi : in base alla configurazione dei diodi e della resistenza , si hanno due implementazioni duali : AND (primo schema) e OR (secondo schema):



V_{out}	V_1	V_2
V_L	V_L	V_L
V_L	V_L	V_H
V_L	V_H	V_L
V_H	V_H	V_H

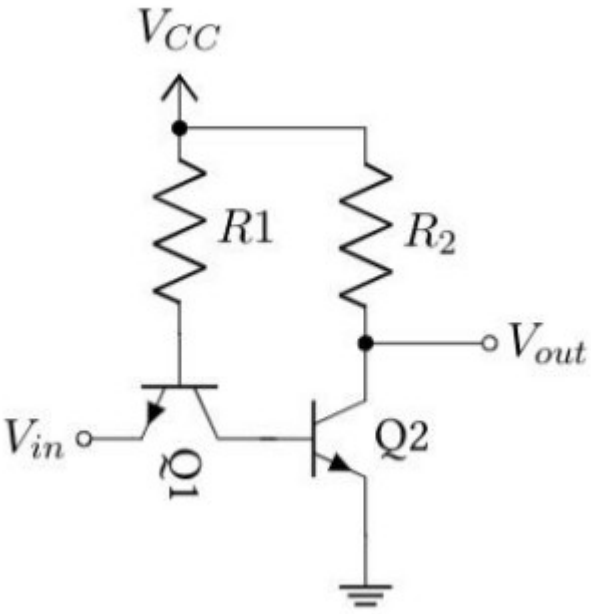


V_{out}	V_1	V_2
V_L	V_L	V_L
V_H	V_L	V_H
V_H	V_H	V_L
V_H	V_H	V_H

Queste porte con questa implementazione avevano un problema : visto che per far funzionare il diodo fa diminuire la tensione applicata al componente , si ha che per più componenti connessi in cascata , si arriva ad una grande attenuazione del segnale :

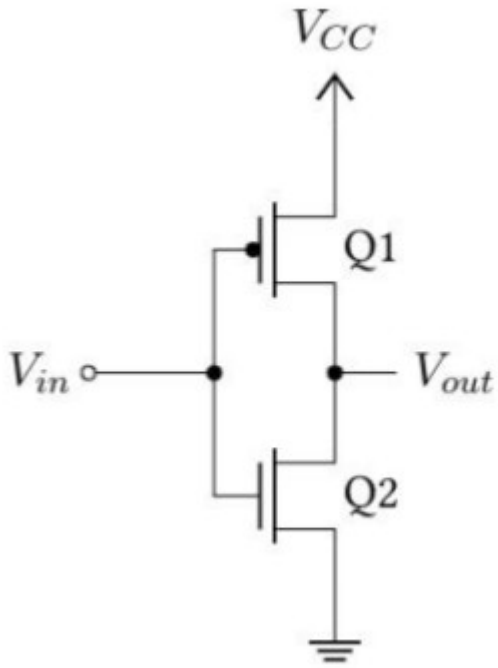
$$V_{out} = \max(V_1, V_2) - V_d$$

Vista questo consumo di energia , a partire dagli anni 60 si è optato per usare una tecnologia a transistor : alcuni servono per commutare il segnale, mentre altri per amplificarlo :

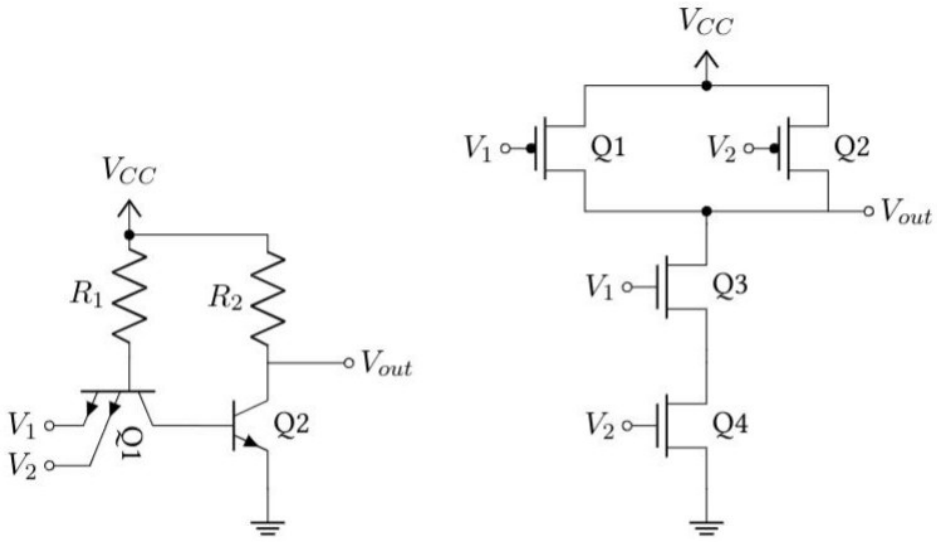


Questa è una porta not!

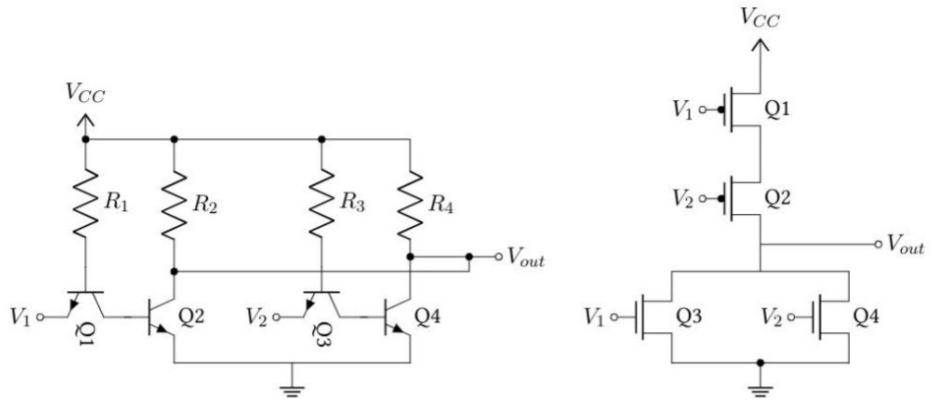
Ma dagli anni 80 si è fatto un miglioramento di questa tecnologia , usando area del dispositvo più piccola e basato su due concetti fondamentali : **rete di pull-up** (collegata direttamente a +Vcc) e **rete di pull-down** (collegata direttamente a massa) :



Inverter



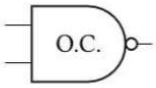
Nand

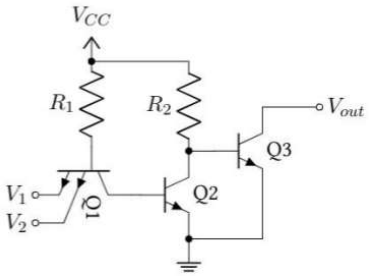


Nor

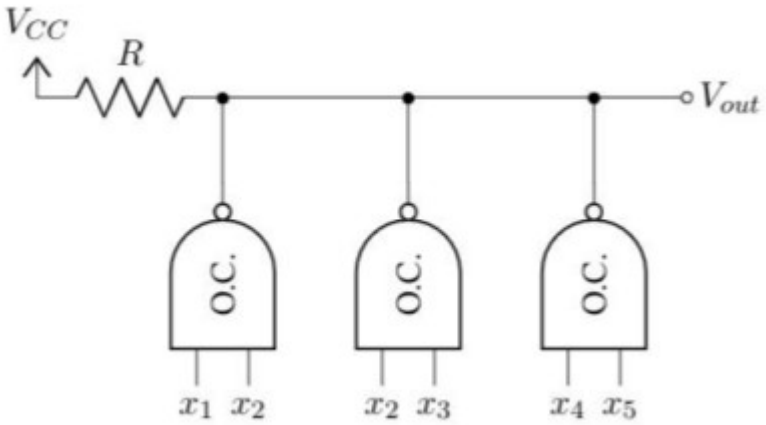
Vediamo ora ultima tecnologia : Open Collector , ovvero tecnologia che aggiunge un transistor il cui collettore è collegato all'uscita della porta : da notare che in questa configurazione si parla di logica negata : la resistenza di pull-up fa salire la tensione quando il circuito è chiuso :

Esempio: NAND in Open Collector

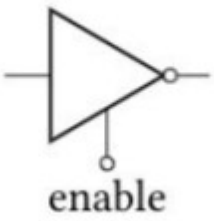


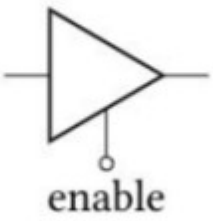


Riassumendo la tecnologia in open collector : si usa quando si ha la necessità di collegare più porte in cascata che possono fornire correnti molto alte, il che potrebbe portare ad un sovraccarico di corrente nel dispositivo (può bruciarsi) . Quindi la corrente che fluisce con questa tecnologia non è direttamente proporzionale al numero di porte usate :



Un primo esempio di circuito combinatorio è il buffer tri-state : fa da ponte tra le diverse parti del circuito , ed è caratterizzato da 3 canali : ingresso, uscita ed enable: In generale l'uscita è uguale ad ingresso se il controllo è attivo , altrimenti stato di alta impedenza (non passa corrente):





IN	ENABLE	OUT
X	0	HiZ
1	1	1
0	1	0