3^ Esercitazione : Mappe di Karnaugh

A) LE MAPPE DI KARNAUGH

Si sa che per ogni FB esistono infinite EB equivalenti. Se ne vuol trovare la *minima* rispetto ad una qualche misura (tipicam. il numero di operatori). Infatti, tradotta l'EB in un circuito, si ha che :

- meno porte → minor costo (benché attualmente il costo di un componente elementare sia pressoché nullo)
- meno porte → minor tempo di attraversamento (di solito)

M. di K. metodo semplice ma buono solo fino a funzioni di 4 variabili. Trova una minima FND

Si basano sul fatto che $x \cdot y \cdot z \;+\; x \cdot y \cdot z \;=\; x \cdot y$

Le mappe consentono di identificare velocemente gli addendi della FCD così semplificabili.

Mappe :

$f(x, y)$

x \ y	0	1
0	p_{00}	p_{01}
1	p_{10}	p_{11}

$f(x, y, z)$

x \ y z	00	01	11	10
0	p_{000}	p_{001}	p_{011}	p_{010}
1	p_{100}	p_{101}	p_{111}	p_{110}

$f(x, y, z, t)$

x y \ z t	00	01	11	10
00	p_{0000}	p_{0001}	p_{0011}	p_{0010}
01	p_{0100}	p_{0101}	p_{0111}	p_{0110}
11	p_{1100}	p_{1101}	p_{1111}	p_{1110}
10	p_{1000}	p_{1001}	p_{1011}	p_{1010}

Algoritmo:

- trasforma f in FCD.

- $p_i = \begin{cases} 1 & \text{se } minterm(p_i) \quad \text{FCD}(f) \\ 0 & \text{altrimenti.} \end{cases}$

  (dove, per esempio, per $f(x,y,z)$ si ha che $P_2 = minterm(p_2) = minterm(p_{010}) = \bar{x} \cdot y \cdot \bar{z}$)

- raggruppa tutti gli 1 adiacenti con un insieme minimale di blocchi.
  (N.B.: le mappe sono da pensare curvate su di sé (*toro*) e quindi ogni elemento della mappa ha 4 elementi adiacenti [1])

- considera la FND che ha per ogni blocco del ricoprimento un addendo ottenuto come il prodotto delle variabili che assumono lo stesso valore su tutti gli elementi del blocco. Le variabili saranno affermate se assumono 1 , negate altrimenti.[2]

Esempio 1: si trovi la minima FND per la funzione seguente

x	*y*	*z*	*t*	*f*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	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	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Anzitutto osserviamo che

$$\text{FCD}(f) = P_0 + P_1 + P_8 + P_9 + P_{10} + P_{13}$$

che richiede $4_{\text{porte NOT}} + 18_{\text{porte AND}} + 5_{\text{porte OR}} = 27_{\text{porte}}$

[1] Per esempio, per $f(x,y,z,t)$, l'elemento p_{0101} è adiacente a p_{0001} , p_{0100} , p_{0111} e p_{1101} ; così come p_{0000} è adiacente a p_{0001} , p_{0100} , p_{1000} e p_{0010} .

[2] Per esempio, per $f(x,y,z,t)$, se il blocco copre due 1 adiacenti allora l'addendo corrispondente sarà il prodotto di 3 letterali, se il blocco copre quattro 1 adiacenti allora si avrà il prodotto di 2 letterali, se il blocco copre otto 1 allora si avrà un solo letterale

xy \ zt	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	0	1	0	0
10	1	1	0	1

da cui

$$\text{FND}_{min} = \bar{y} \cdot \bar{z} + x \cdot \bar{z} \cdot t + x \cdot \bar{y} \cdot \bar{t}$$

che richiede solo $3_{\text{porte NOT}} + 5_{\text{porte AND}} + 2_{\text{porte OR}} = 10_{\text{porte}}$

Il metodo funziona anche per funzioni booleane parziali (N.B.: una FB parziale è una funzione che non è definita su tutte i 2^n possibili input [3]). Con le mappe di Karnaugh i valori indefiniti possono venir considerati 0 o 1 a seconda di quale assegnamento dà il ricoprimento migliore.

Esempio 2: si trovi la minima FND per la funzione seguente

x	*y*	*z*	*f*
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	-
1	0	1	0
1	1	0	1
1	1	1	-

La mappa di K. è

x \ yz	00	01	11	10
0	*1*	*0*	*0*	*1*
1	-	*0*	-	*1*

da cui $E_{min} = \bar{z}$

[3] Una funzione parziale può in realtà essere vista come una funzione totale $f : \{0, 1\}^n \to \{0, 1, -\}$ dove - è detto *don't care* ed esprime il fatto che la funzione non è definita sull'input corrispondente.

N.B.: è risultato migliore assegnare al primo ‘-‘ il valore 1 mentre al secondo il valore 0; ciò origina una EB t.c. $E_{min}(100) = 1$ e $E_{min}(111) = 0$. L’arbitrarietà di tale scelta è giustificata dal fatto che la funzione non è definita su tali input e quindi possiamo assumere che non si presenteranno mai alla funzione (e pertanto il comportamento della funzione su di essi è irrilevante).

B) ANALISI & SINTESI DI DUE CIRCUITI COMBINATORI

Esempio 3: funzione di overflow per la somma di due naturali a 2 bit.

$$f:\ \{0,1\}^2 \times \{0,1\}^2 \longrightarrow \{0,1\}$$

t.c. $f(x,y,z,t) = 1$ sse $xy_2 + zt_2$ (cioè i numeri binari ottenuti concatenando i primi due input e gli ultimi due) dà luogo ad overflow (cioè il risultato della somma non è esprimibile come numero di 2 bit).

x	***y***	***z***	***t***	***f***
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

la cui mappa di K. è

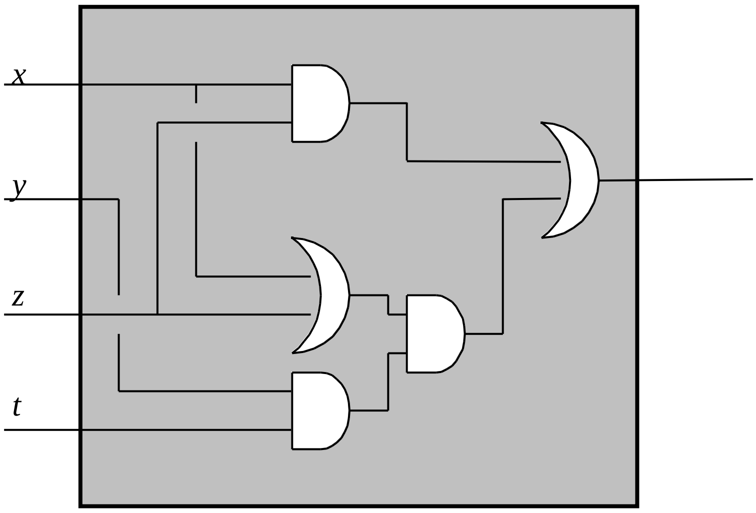
z t / ***x y***	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

da cui $E_{min} = x\cdot z + x\cdot y\cdot t + y\cdot z\cdot t$
N.B.: osservare che l'EB così ottenuta è la minima FND, che non è detto in generale che sia la minima EB in assoluto. Infatti l'EB dell'esempio precedente è equivalente alla seguente

$$E_{min}' = x\cdot z + y\cdot t\cdot (x + z)$$

che usa due porte AND in meno.

Pertanto il circuito minimo che realizzeremo è quello associato a quest'ultima EB e sarà



Esempio 4: funzione di primalità[4] per interi di 4 bit in complemento a 2.

$$f: \{0,1\}^4 \longrightarrow \{0,1\}$$

t.c. $f(x, y, z, t) = 1$ sse $xyzt_2$ (cioè il numero binario in complemento a 2 ottenuto concatenando quattro input) è primo.

x	*y*	*z*	*t*	DEC	*f*
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	2	1
0	0	1	1	3	1
0	1	0	0	4	0
0	1	0	1	5	1
0	1	1	0	6	0
0	1	1	1	7	1
1	0	0	0	Non usata	-
1	0	0	1	–7	1
1	0	1	0	–6	0
1	0	1	1	–5	1
1	1	0	0	–4	0
1	1	0	1	–3	1
1	1	1	0	–2	1

[4] **Def.:** dati due interi p e q, si dice che <u>p è divisibile per q</u> se p *DIV* q è un numero intero.
Def.: un numero intero z si dice <u>primo</u> se $z \neq 1, -1$ e z è divisibile solo per $1, -1, z, -z$.

1 1 1 1 | –1 0

la cui mappa di K. è

zt \ xy	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	0	1	0	1
10	-	1	1	0

da cui $E_{min} = \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot t + x \cdot \bar{z} \cdot t + x \cdot \bar{y} \cdot t + x \cdot y \cdot z \cdot \bar{t}$

Fattorizzando si ottiene $E_{min}' = z \cdot (\bar{x} \cdot \bar{y} + x \cdot y \cdot \bar{t}) + t \cdot ((x \textbf{ XOR } y) + x \cdot \bar{z})$

che è realizzabile con un circuito combinatorio di 14 porte.

COMPARATORE ARITMETICO

Problema: date due stringhe binarie di *n* bit *A* e *B* rappresentanti due numeri naturali, dire se

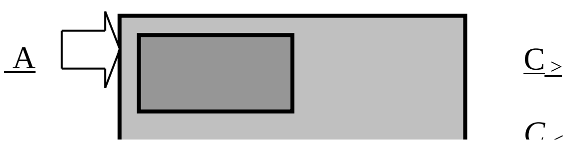
$A > B$, $A = B$, $A < B$

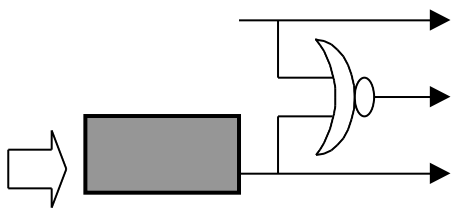
OSS.1 : è possibile solo una delle condizioni appena presentate!!

Vogliamo un circuito del tipo

tale che :
- $C_> = 1$ sse $A > B$
- $C_< = 1$ sse $A < B$
- $C_= = 1$ sse $A = B$

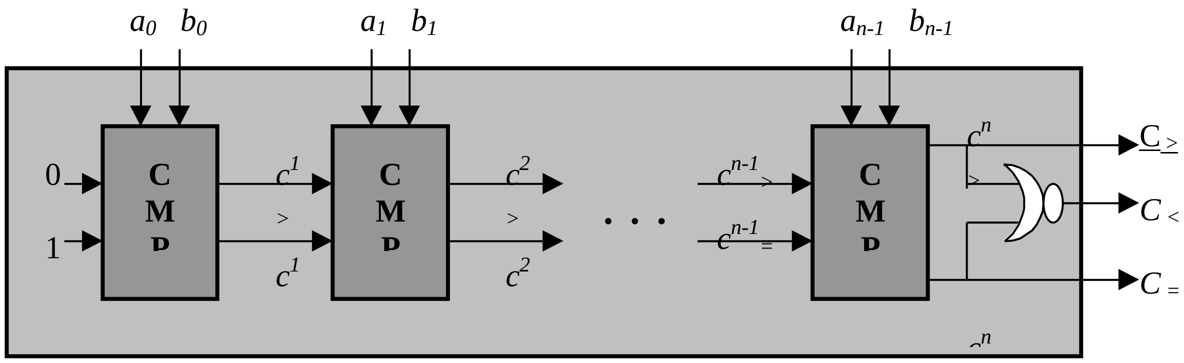
OSS.2: per l'OSS.1 si ha che $C_< = C_> \textbf{ NOR } C_=$ e pertanto basterà trovare i circuiti per calcolare $C_>$ e $C_=$, da cui



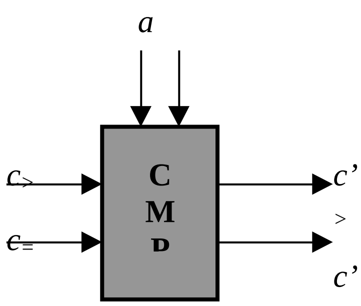


L’idea è quella di costruire un circuito incrementale, cioè un circuito formato da *n* celle circuitali elementari di confronto messe in cascata; per fare ciò usiamo dei risultati parziali così definiti:

Per ogni $i = 1, \dots, n$ · $c^i_> = 1$ sse $a_{i-1} \dots a_0 > b_{i-1} \dots b_0$
· $c^i_= = 1$ sse $a_{i-1} \dots a_0 = b_{i-1} \dots b_0$



A) REALIZZAZIONE della CELLA ELEMENTARE di COMPARAZIONE (CMP)



La tabella di verità per **CMP** è

a	***b***	***c*** $_>$	***c*** $_=$	***c’*** $_>$	***c’*** $_=$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	-	-
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	-	-
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	-	-
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0

1	1	1	1	|	-	-

N.B.: in corrispondenza degli input con $c_> = c_= = 1$ sono stati messi due '-' poiché, per OSS.1, tale situazione non si presenterà mai.

A.1) REALIZZAZIONE del SOTTO-CIRCUITO per $c'_>$

La mappa di K. è

$a\ b$ \ $c_>\ c_=$	00	01	11	10
00	0	0	-	1
01	0	0	-	0
11	0	0	-	1
10	1	1	-	1

da cui $E_{min} = c_> \cdot a + c_> \cdot \overline{b} + a \cdot \overline{b}$

che fattorizzata porta a $E'_{min} = c_> \cdot (a + \overline{b}) + a \cdot \overline{b}$

A.2) REALIZZAZIONE del SOTTO-CIRCUITO per $c'_=$

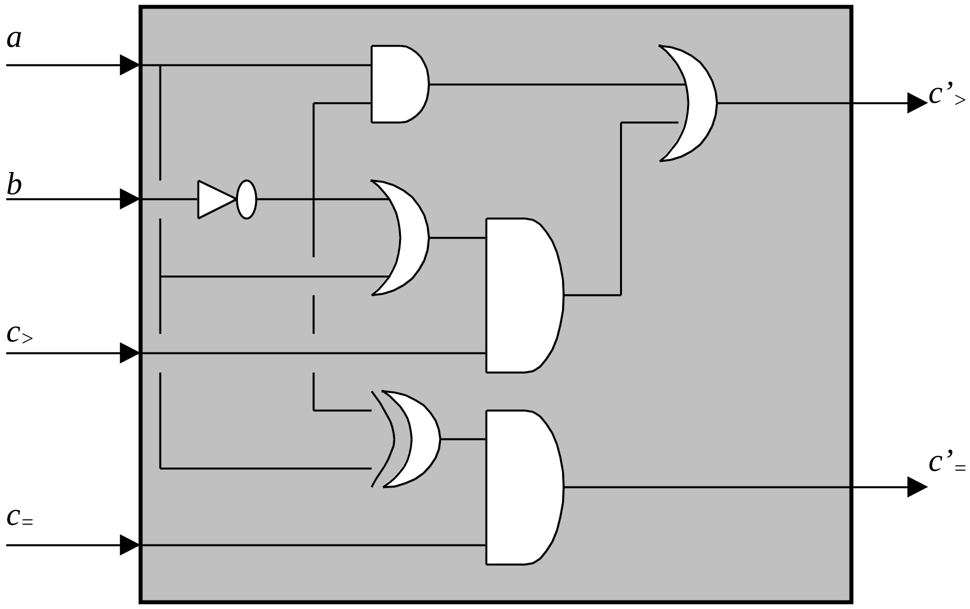
La mappa di K. è

$a\ b$ \ $c_>\ c_=$	00	01	11	10
00	0	1	-	0
01	0	0	-	0
11	0	1	-	0
10	0	0	-	0

da cui $E_{min} = c_= \cdot a \cdot b + c_= \cdot \overline{a} \cdot \overline{b}$

che fattorizzata porta a $E'_{min} = c_= \cdot (a \cdot b + \overline{a} \cdot \overline{b}) = c_= \cdot (a \textbf{ XOR } \overline{b})$

A.3) DISEGNO del CIRCUITO



A.4) COMPARATORE di INTERI

Il circuito concettualmente resta lo stesso; bisogna però notare che, lavorando con interi, bisogna tener conto dei segni e dei valori assoluti. In particolare:

- *se A e B sono entrambe non-negativi :* vedi pagg. precedenti
- *se A e B sono entrambe negativi :* gli output del circuito ottenuto nelle pagg. precedenti vanno complementati
- *se A è non-negativo e B è negativo :* si ha sempre $C_{>} = 1$, $C_{<} = C_{=} = 0$
- *se A è negativo e B è non-negativo :* si ha sempre $C_{>} = 0$, $C_{>} = C_{=} = 0$

Pertanto, detta C l'uscita $C_{>}$ del circuito per i naturali, sia ha che

$$C_{>} = (\overline{a}_{n-1} \cdot \overline{b}_{n-1} \cdot C) + (a_{n-1} \cdot b_{n-1} \cdot \overline{C}) + (\overline{a}_{n-1} \cdot b_{n-1})$$

ricordando che A è non-negativo sse $a_{n-1} = 0$.

In maniera simile si possono ricavare le espressioni per le corrispondenti uscite $C_{<}$ e $C_{=}$ del circuito per interi in complemento a 2.