Ottimizzazione di funzioni combinatorie

Per *ottimizzazione* di una funzione si intende la sua trasformazione, attraverso passi successivi, con lo scopo di ottenere un'espressione equivalente ma migliore rispetto ad una data metrica di valutazione (area occupata, tempo necessario a produrre un dato risultato, potenza o energia assorbita ecc.).

Possibili metriche di area:

- Numero di porte logiche generiche
- Numero di porte logiche a due ingressi
- Numero di implicanti o di implicati
- Numero di letterali

Costo di una espressione algebrica (1)

- Il costo in termini di *letterali* C_L è pari al numero delle variabili indipendenti della funzione, ciascuna moltiplicata per il numero di volte che essa compare nella forma.

- Il costo in termini di *funzioni* o *porte* C_P è pari al numero delle funzioni elementari f_i che la compongono, che per reti unilaterali è uguale al numero complessivo di porte adoperate.

Costo di una espressione algebrica (2)

- Il costo in termini di *ingressi* C_I è pari al numero delle funzioni f_i che la compongono, ciascuna moltiplicata per le variabili (dipendenti o indipendenti) di cui è funzione. Per reti unilaterali tale costo equivale al numero complessivo di porte adoperate, ciascuna pesata per il numero di ingressi (fan-in).

Esempio

- f=b(a'+c+d) $\mathbf{C_L}$=4 $\mathbf{C_P}$=2 $\mathbf{C_I}$=5
- f=bc(a'd+b'+c)+c'(d+a')(b+c) $\mathbf{C_L}$=11 $\mathbf{C_P}$=7 $\mathbf{C_I}$=17

Costo di una espressione algebrica (3)

F3=AB+C(D+E)
F3=AB+CD+CE

costo ingressi=8 per (a) e 9 per (b)

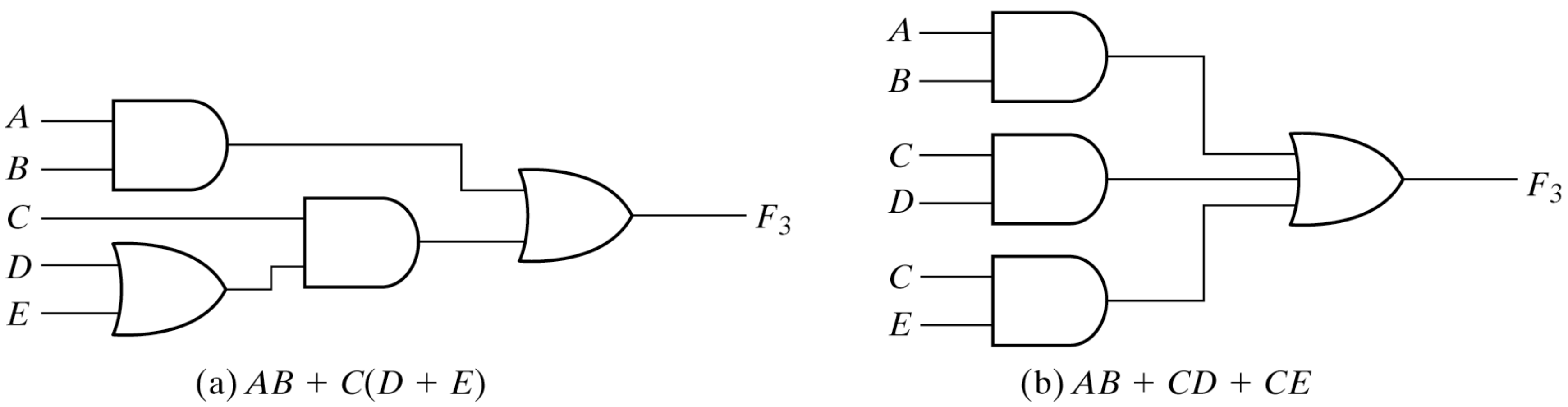


Fig. 2-4 Three- and Two-Level implementation

Costo di una espressione algebrica (4)

$G = ABCD + A'B'C'D$ (i)

$G = (A'+B)(B'+C)(C'+D)(D'+A)$ (ii)

- Costo letterali (i) e (ii) è pari a 8 ma la forma (ii) occupa una maggiore area
- Costo ingressi è pari a 8+2=10 per (i)
- Costo ingressi è pari a 8+4=12 per (ii)

Funzioni di costo: esempio

$$f = bc(a\overline{d} + \overline{b} + c) + \overline{c}(d + \overline{a})(b + c)$$

Costo di letterali (CL)	**11**
Costo di funzioni o di porte (CP)	**7**
Costo di ingressi (CI)	**17**

Ottimizzazione di funzioni combinatorie: espansione (1/3)

I metodi di ottimizzazione delle funzioni combinatorie sono basati sull'applicazione delle proprietà dell' Algebra di Boole:

- Assorbimento:

  P1+P2 = xP + x'P = (x+x')P = P

in tal caso si dice che P1 e P2 generano ***consenso***

- Idempotenza:

  P+P=P

Esse consentono di semplificare l'espressione di una funzione a partire dalla sua rappresentazione in forma canonica, che ne assicura la copertura in termini di somma di mintermini (o prodotto di maxtermini).

Ottimizzazione di funzioni combinatorie: espansione (2/3)

- L’applicazione delle proprietà di assorbimento ed idempotenza è alla base del processo di espansione, volto a trasformare l’espressione algebrica di una funzione in modo da costruire termini prodotto (o somma) costituiti dal minor numero possibile di letterali.

- Si introduce così il concetto di implicante, ossia un prodotto (o una somma) di letterali risultante dal processo di espansione, che assorbe più mintermini (maxtermini) semplificando la copertura di una funzione.

Ottimizzazione di funzioni combinatorie: espansione (3/3)

- Applicando ripetutamente il processo di espansione ad una funzione è possibile determinare un insieme di implicanti primi, ossia non ulteriormente semplificabili, candidati a far parte della copertura ottima della funzione.

- Fra gli implicanti primi è possibile estrarre un set di implicanti primi essenziali, necessari alla copertura poiché sono gli unici a coprire qualche "uno" della funzione; la copertura ottima conterrà dunque tali implicanti essenziali più un sottoinsieme degli implicanti primi rimanenti, scelti secondo un criterio di costo.

Tre tipologie di ottimizzazione dei circuiti combinatori

- Circuiti a 2 livelli e 1 uscita: metodo esatto per identificare i primi implicanti essenziali e un metodo esatto o approssimato (branch & bound) per identificare una copertura ottima
  - **In questo corso tratteremo solo questo problema**
- Circuiti a 2 livelli e più uscite: esistono metodi approssimati per trovare la copertura ottima basati sull'identificazione di implicanti primi essenziali di ogni singola uscita.
- Circuiti a più livelli e più uscite: numerosi metodi approssimati per esplorare diverse alternative di area e ritardo (i più efficaci: sintesi ottima a 2 livelli di porzioni del circuito a 1 uscita.)

Costo Forma Minima: esempio

$$f = bc(a\overline{d} + \overline{b} + c) + \overline{c}(d + \overline{a})(b + c)$$

$$\Updownarrow$$

CL=11; CP=7; CI=17

$$f = b(\overline{a} + c + d)$$

CL=4; CP=2; CI=5

Alcuni presupposti teorici (1/3)

- Una funzione $y = f(x_1, x_2, \ldots, x_n)$ può essere espressa come una somma contenente soltanto i suoi implicanti primi (forma *PI*)
  - Infatti, un implicante non primo A_i presente in una forma somma-di-prodotti della *f* può essere sempre eliminato osservando che si può aggiungere alla *f* un implicante primo P_i a sua volta implicato da A_i senza alterare la *f*
  - A_i è poi assorbito da P_i

Dimostrazione

- *Una funzione $y = f(x_1, x_2, ..., x_n)$ può essere espressa come somma di soli suoi primi implicanti (o - per brevità - in forma PI).*

Dimostrazione

Se la funzione in forma elementare $y = \sum A_i$ contiene un termine A_k non PI si ha allora $A_k \Rightarrow P_j \Rightarrow y$ con P_j primo implicante;

P_j può essere aggiunto alla *y* e si ottiene $y = \sum A_k + P_j$;

essendo $A_k + P_j = P_j$, A_k può essere eliminato dalla forma di y, dove viene sostituito da P_j.

Il ragionamento va ripetuto per tutte le clausole non PI.

Alcuni presupposti teorici (2/3)

- Una forma elementare che minimizzi i valori dei costi CL e CI è una forma PI
- Fra le forme minime a 2 livelli che minimizzano CP ne esiste almeno una PI
- Sotto il vincolo di rete a 2 livelli, la forma minima va allora cercata tra le forme PI

DIM.: Fra tutte le forme elementari non PI, si scelga quella minima; se questa viene trasformata in una forma PI nella quale P_j sostituisce A_k, poiché P_j è una clausola di ordine inferiore ad A_k, diminuisce CL (diminuendo i letterali) e diminuisce CI (diminuendo il numero di variabili indipendenti della funzione componente) mentre CP non aumenta (diminuisce se era già nella sommatoria, resta inalterato altrimenti).

Alcuni presupposti teorici (3/3)

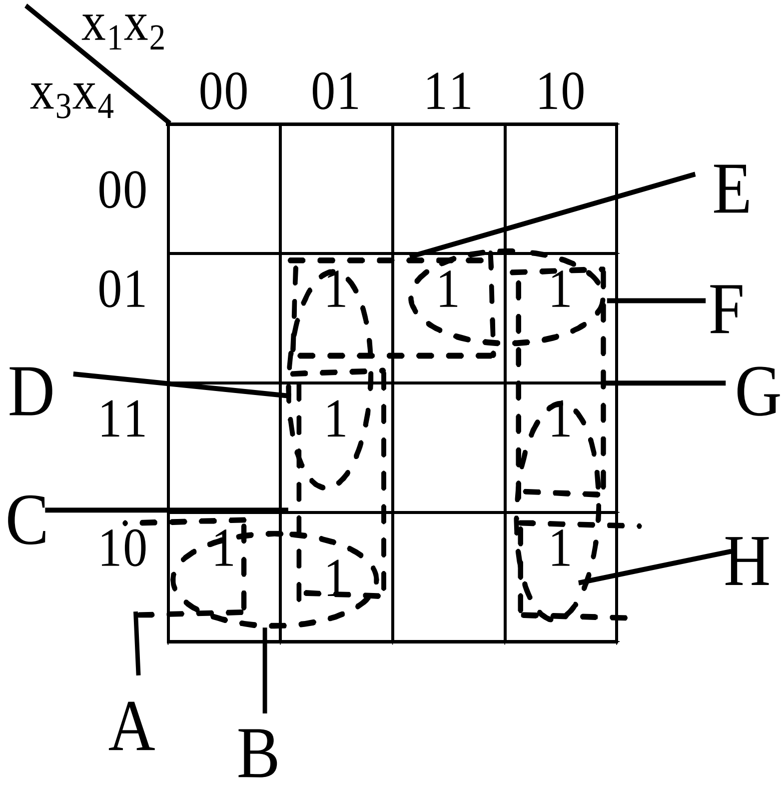
- Un implicante primo E_i di una funzione *f* è detto ***essenziale*** se è l'unico ad essere implicato da un mintermine di *f*
- In altri termini, E_i è l'unico a "*coprire*" un determinato mintermine della funzione
- Il ***nucleo N*** della funzione è la somma dei suoi implicanti primi essenziali

$$\boxed{N = \sum_{i=1}^{k} E_i}$$

- Ogni forma minima di f è del tipo ***f* = *N*+*R***

  con **N** ed **R** eventualmente nulli

Funzioni cicliche (N=0)



N = 0

Metodi di Ottimizzazione

I Metodi di Ottimizzazione possono essere classificati in due macrocategorie:

- Metodi Esatti
  - Karnaugh
  - Quine-McCluskey
- Metodi Euristici
  - Utilizzano algoritmi euristici che applicano trasformazioni alla rete e valutano l’ottimo ad ogni passo

Il metodo delle Mappe di Karnaugh

Si articola in due fasi:

1) **Espansione**: consiste nella ricerca degli implicanti primi, costituiti dai sottocubi di area massima sulle mappe

2) **Copertura**: consiste nel determinare il sottoinsieme minimo di implicanti primi della funzione in grado di coprire tutti i suoi mintermini

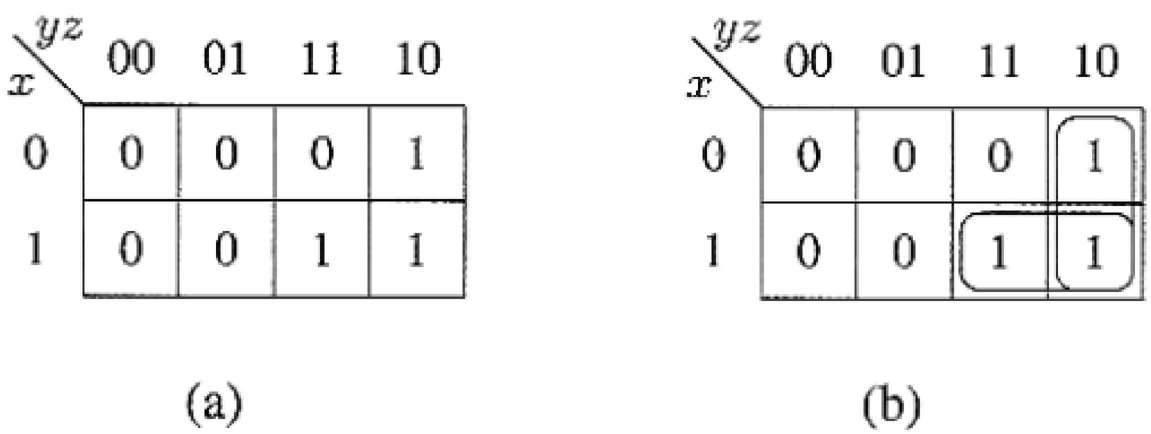
Procedura per la minimizzazione

1. Ricerca di tutti gli implicanti primi della funzione *f* da minimizzare;
2. Selezione degli implicanti primi essenziali, per individuare il **nucleo *N***;
3. Copertura della funzione, cioè determinazione della forma minima di *R*, da aggiungere ad *N* per minimizzare *f*

I passi 1) e 2) possono essere svolti con l’ausilio delle mappe di Karnaugh per funzioni fino a 5 variabili

Il metodo delle Mappe di Karnaugh - fase di espansione

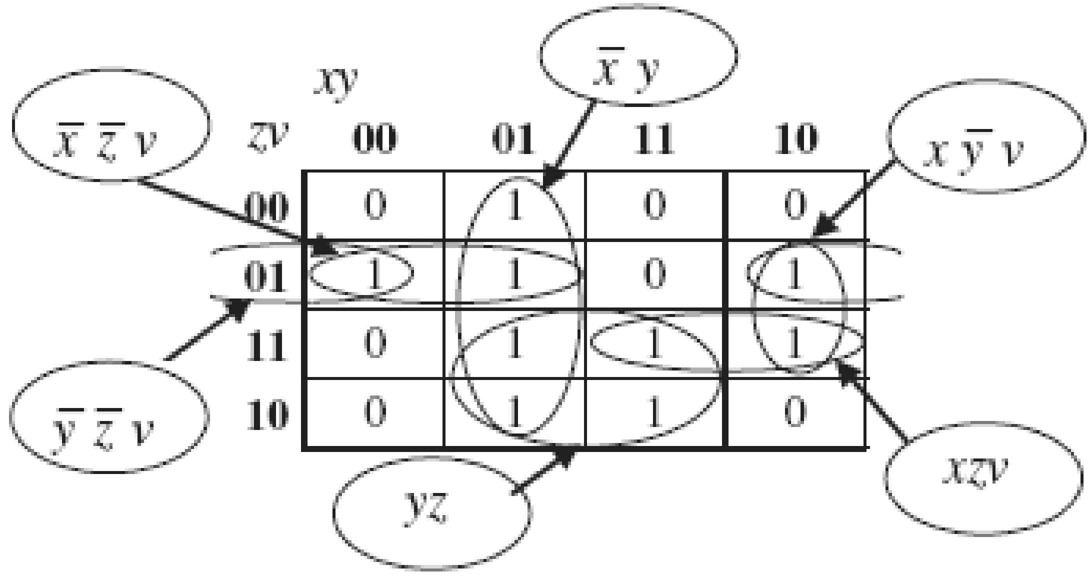
I mintermini semplificabili sono rappresentati da celle adiacenti sulle mappe: l' operazione di espansione viene effettuata a partire da ogni mintermine in tutte le direzioni per raggrupparne un numero equivalente a una potenza di 2. Ciascun mintermine può appartenere a più raggruppamenti.



Mappa di Karnaugh della funzione $f(x, y, z) = xyz + xyz' + x'yz'$

Il metodo delle Mappe di Karnaugh – fase di espansione (esempio)

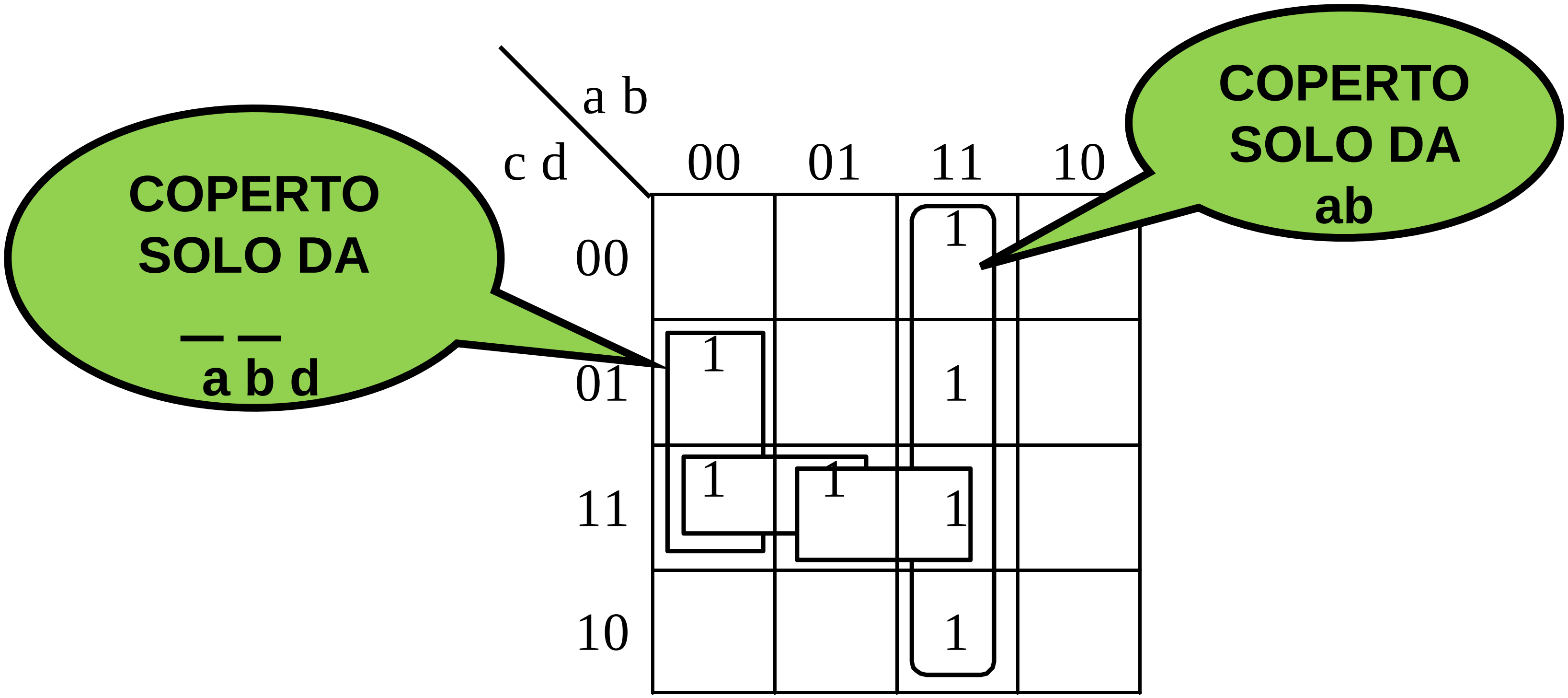
Mappa di Karnaugh per la funzione
f(*x*,*y*,*z*,*v*)=Σ{1, 4, 5, 6, 7, 9, 11, 14, 15}



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

Il metodo delle Mappe di Karnaugh – individuazione del nucleo

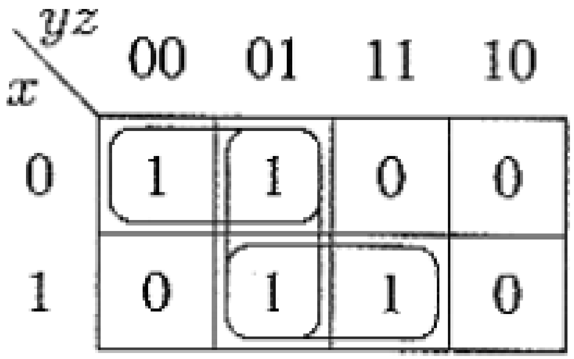
- Sono essenziali gli implicanti primi che **<u>da soli</u>** ricoprono un 1



Il metodo delle Mappe di Karnaugh – copertura

- Individuati gli implicanti primi essenziali (nucleo), occorre scegliere, fra gli implicanti primi non ancora selezionati, quelli che consentono di coprire tutta la funzione con un costo totale minimo
- In casi semplici, la copertura minima è facilmente individuabile sulla mappa di Karnaugh
- In generale, per la ricerca della copertura minima (compreso il nucleo) è possibile utilizzare metodi tabellari

Il metodo delle Mappe di Karnaugh-Esempio



Considerando tutti gli implicanti primi individuati si ottiene l'espressione:

f(xyz)= x'y'+ y'z+ xz

che non risulta minima.

Esaminando la mappa si nota che gli implicanti x'y' e xz sono sufficienti a coprire tutti gli 1 della funzione, e quindi fanno parte della copertura minima mentre l'implicante y'z può essere trascurato. In definitiva quindi:

f(xyz)= x'y'+ xz

La fase di copertura: matrice di copertura

- La fase di copertura può essere eseguita con l'ausilio di una matrice di copertura;
- La **matrice di copertura** si costruisce ponendo sulle righe gli implicanti primi di f determinati nella fase di espansione e sulle colonne tutti i mintermini per cui la funzione vale 1;
- La casella in posizione (i,j) viene marcata con una "x" o un "1" se l'implicante i copre il mintermine j
  - in caso di funzioni non completamente specificate, nella tabella di copertura vengono indicati solo i mintermini per cui la funzione vale 1, poiché non è necessario coprire le condizioni di indifferenza

Individuazione del nucleo sulla matrice di copertura

- Gli implicanti essenziali (nucleo) sono facilmente individuabili nella matrice di copertura, poiché sono i soli a coprire un dato mintermine (una sola “x” in una colonna).
- Una volta individuati tali implicanti, è possibile generare una nuova tabella eliminando la riga corrispondente all’implicante essenziale e tutte le colonne dei mintermini da esso coperti nella tabella di partenza.
- Si riesamina la nuova tabella prodotta finché non è più possibile individuare implicanti essenziali.

Copertura e criteri di dominanza

- Una volta ottenuto il nucleo, per determinare la copertura completa della funzione è possibile applicare i **criteri di dominanza di riga e di colonna**
  - La ***riga i domina la riga j*** se l'implicante Pi copre tutti i mintermini che copre l'implicante Pj più almeno uno
  - La ***colonna i domina la colonna j*** se il mintermine mj è coperto da un sottoinsieme degli implicanti che coprono mi

- I criteri di dominanza prescrivono di **eliminare dalla matrice di copertura le righe dominate e le colonne dominanti**

Criteri di dominanza: spiegazione

- La ***riga i domina la riga j:***

i mintermini coperti dall'implicante dominato sono un sottoinsieme dei mintermini coperti dall'implicante dominante: scegliendo di eliminare l'implicante dominato e mantere il dominante avremmo la certezza di coprire un insieme maggiore di mintermini, con un costo totale di copertura di sicuro non maggiore di quello che si avrebbe facendo la scelta opposta

- La ***colonna i domina la colonna j:***

qualsiasi implicante copra mj copre anche mi: scegliendo di eliminare il mintermine mi e mantenere mj avremmo la certezza che gli implicanti selezionati per coprire quest'ultimo coprono anche il primo, con un costo totale di copertura non maggiore di quello che si avrebbe facendo la scelta opposta

Fase di copertura completa

- Il metodo tabellare per la copertura che fa uso della matrice di copertura procede allora come segue:
    1. Si ricercano gli implicanti primi (*PI*) e si individuano quelli essenziali;
    2. Si includono nella forma minima i *PI* essenziali, eliminandoli dalla matrice, unitamente con i mintermini ricoperti;
    3. Si eliminano le righe dominate e le colonne dominanti;
    4. Si individuano i *PI* essenziali “secondari” della matrice così ridotta;
    5. Si ripetono i passi 2, 3, 4 finché è possibile.

Esempio (1/6)

$$f(x_1, x_2, x_3, x_4, x_5) = \sum (0,1,2,8,9,15,17,21,24,25,27,28,31) =$$

$$= \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4\overline{x}_5 + \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 x_5 + \overline{x}_1\overline{x}_2\overline{x}_3 x_4 \overline{x}_5 + \ldots\ldots$$

Esempio (2/6)

Implicanti	Mintermini coperti
A = -100-	8,9,24,25
B = --001	1,9,17,25
C = 0-00-	0,1,8,9
D = 11-11	27,31
E = -1111	15,31
F = 110-1	25,27
G = 10-01	17,21
H = 11-00	24,28
J = 000-0	0,2

Esempio (3/6)

	P_0	P_1	P_2	P_8	P_9	P_{15}	P_{17}	P_{21}	P_{24}	P_{25}	P_{27}	P_{28}	P_{31}
A				1	1				1	1			
B		1			1		1			1			
C	1	1		1	1								
D											1		1
E						1							1
F										1	1		
G							1	1					
H									1			1	
J	1		1										

Primi implicanti essenziali: N (nucleo) = J, E, G, H

Esempio (4/6)

	P_0	P_1	P_2	P_8	P_9	P_{15}	P_{17}	P_{21}	P_{24}	P_{25}	P_{27}	P_{28}	P_{31}
A				1	1				1	1			
B		1			1		1			1			
C	1	1		1	1								
D											1		1
E						1							1
F										1	1		
G							1	1					
H									1			1	
J	1		1										

Gli implicanti primi essenziali J,E,G,H coprono i mintermini: 0, 2, 15, 17, 21, 24, 28, 31

Esempio (5/6)

	P_1	P_8	P_9	P_{25}	P_{27}
A		1	1	1	
B	1		1	1	
C	1	1	1		
D					1
F				1	1

	P_1	P_8	P_{25}	P_{27}
A		1	1	
B	1		1	
C	1	1		
F			1	1

	P_1	P_8
A		1
B	1	
C	1	1

riga D dominata dalla F
Colonna P_9 dominante

F implicante primo essenziale secondario:
copre P_{25} e P_{27}

Righe A e B dominate dalla **C**

Esempio (6/6)

$$f(x_1, x_2, x_3, x_4, x_5) = \sum (0,1,2,8,9,15,17,21,24,25,27,28,31) =$$

$$= \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5} + \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,x_5 + \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,x_4\,\overline{x_5} + \ldots\ldots =$$

$$= E + G + H + J + F + C =$$

$$= x_2 x_3 x_4 x_5 + x_1 \overline{x_2}\,\overline{x_4}\,x_5 + x_1 x_2 \overline{x_4}\,\overline{x_5} + \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_5}$$

$$+ x_1 x_2 \overline{x_3}\,x_5 + \overline{x_1}\,\overline{x_3}\,\overline{x_4}$$

Metodo Quine-McCluskey

- Metodo esatto per la sintesi di reti a 2 livelli
- Fattibile fino a circa 20 ingressi
- In grado di considerare funzioni a piú uscite
- Puó minimizzare sia il costo degli implicanti che quello dei letterali

L’algoritmo (facilmente implementabile) opera in due fasi distinte

1) **Espansione**
2) **Copertura**

Il metodo di Quine-McCluskey – I Fase

1. Si considerano i mintermini appartenenti all'ON-Set e al DC-Set della funzione, espressi mediante i valori dei corrispondenti letterali, e li si ordina in senso crescente in base al numero di "1" contenuti, dividendoli in classi.
2. Ogni elemento di ciascuna classe viene confrontato con tutti gli elementi della classe immediatamente successiva allo scopo di individuare consensi: la variabile eventualmente eliminata in caso di consenso viene segnata con il simbolo di don't care nel nuovo implicante generato dal processo di espansione.

   - in caso di confronto fra implicanti contenenti don't care è possibile generare espansione solo se il simbolo di don't care si trova nella stessa posizione nei due implicanti di partenza ed essi differiscono solo per un letterale.

Il metodo di Quine-McCluskey – I Fase

3. Ogni volta che due implicanti partecipano ad un raccoglimento devono essere marcati poiché non rappresentano implicanti primi, e non devono quindi essere considerati nella seconda fase.

   - Nel caso di funzioni non completamente specificate se due mintermini entrambi appartenenti al DC-Set generano espansione, il nuovo implicante viene introdotto nella successiva tabella ma viene marcato a priori, poiché non sarà necessario coprirlo nella successiva fase.

4. Il procedimento viene ripetuto finché non è più possibile determinare consensi; gli implicanti che risulteranno non marcati alla fine della prima fase sono gli implicanti primi della funzione.

Quine-McCluskey – I Fase - ESEMPIO

x	y	z	v	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

m_i	x	y	z	v	
1	0	0	0	1	✓
4	0	1	0	0	✓
5	0	1	0	1	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
7	0	1	1	1	✓
11	1	0	1	1	✓
14	1	1	1	0	✓
15	1	1	1	1	✓

$\{m_1 \ldots m_n\}$	x	y	z	v	
1, 5	0	–	0	1	A
1, 9	–	0	0	1	B
4, 5	0	1	0	–	✓
4, 6	0	1	–	0	✓
5, 7	0	1	–	1	✓
6, 7	0	1	1	–	✓
6, 14	–	1	1	0	✓
9, 11	1	0	–	1	C
7, 15	–	1	1	1	✓
11, 15	1	–	1	1	D
14, 15	1	1	1	–	✓

$\{m_1 \ldots m_n\}$	x	y	z	v	
4, 5, 6, 7	0	1	–	–	E
6, 7, 14, 15	–	1	1	–	F

A B C D E F sono tutti gli **implicanti primi** della funzione (non sono stati marcati durante il processo di espansione)

Il metodo di Quine-McCluskey – II Fase: copertura

1. Si costruisce la **matrice di copertura** come visto in precedenza, con l'unica accortezza che gli eventuali mintermini corrispondenti a **don't care** nella funzione non devono essere inclusi nella matrice: non è necessario coprire condizioni di indifferenza!

2. Si applica il metodo di copertura tabellare visto finché non si assicura la copertura di tutti i mintermini della funzione.

Quine-McCluskey – II Fase – ESEMPIO (1/2)

	m1	m4	m5	m6	m7	m9	m11	m14	m15
A	X		X						
B	X					X			
C						X	X		
D							X		X
E		X	X	X	X				
F				X	X			X	X

Il mintermine m4 risulta coperto solo da E e il mintermine m14 solo da F:

E ed F pertanto sono implicanti essenziali e possono essere cancellati dalla tabella insieme a tutti i mintermini che coprono.

C(F)= {E,F}

Quine-McCluskey – II Fase – ESEMPIO (2/2)

	m1	m9	m11
~~A~~	~~X~~		
B	X	X	
C		X	X
~~D~~			~~X~~

Nella tabella risultante ogni mintermine è coperto almeno da due implicanti: non ci sono più implicanti essenziali e si può procedere col metodo della dominanza:

La riga C domina la riga D e la B domina la A: cancello le righe A e D.

	m1	m9	m11
B	X	X	
C		X	X

A questo punto B e C sono diventati pseudo-essenziali perché coprono mintermini non coperti da altri implicanti: essi sono allora etichettati come implicanti essenziali *secondari* e aggiunti alla **copertura minima di f**:

C(F)= {E,F,B,C}

Esercizio 1 (1/4)

- Minimizzare con il metodo di Quine-McCluskey, la rete con quattro ingressi ed una uscita specificata come segue:

  ONSet={0,2,4,5,6,7,8,9,13,15}; DCSet=Ø

Soluzione:

- Si considerino i valori degli ingressi delle configurazioni che costituiscono l'ONSet e si ricava:

ONSet={0000,0010,0100,0101,0110,0111,1000,1001,1101,1111}

- che dà origine alla seguente partizione:

ONSet={{0000}{0010,0100,1000}{0101,0110,1001}{0111,1101}{1111}}

Esercizio 1 – I fase (2/4)

Passo 0	Passo 1	Passo 2
0000 (0) ~	00-0 (0,2) ~	0--0 (0,4,2,6)
0010 (2) ~	0-00 (0,4) ~	01-- (4,5,6,7)
0100 (4) ~	-000 (0,8)	-1-1 (5,7,13,15)
1000 (8) ~	0-10 (2,6) ~	
0101 (5) ~	010- (4,5) ~	
0110 (6) ~	100- (8,9)	
1001 (9) ~	01-0 (4,6) ~	
0111 (7) ~	01-1 (5,7) ~	
1101 (13) ~	-101 (5,13) ~	
1111 (15) ~	011- (6,7) ~	
	1-01 (9,13)	
	-111 (7,15) ~	
	11-1 (13,15) ~	

Tutte le configurazioni che non sono state marcate con il simbolo "~" sono implicanti primi. Si determina così l'elenco completo degli implicanti primi da considerare:

P1	P2	P3	P4	P5	P6
!a!d	!ab	bd	!b!c!d	a!b!c	a!cd
(0,2,4,6)	(4,5,6,7)	(5,7,13,15)	(0,8)	(8,9)	(9,13)

Seconda fase: si considera la tabella implicanti/mintermini e, applicando i tre consueti criteri, viene semplificata.

Esercizio 1 – II fase (3/4)

	0	2	4	5	6	7	8	9	13	15
P1	X	X	X		X					
P2			X	X	X	X				
P3				X		X			X	X
P4	X						X			
P5							X	X		
P6								X	X	

P1 e P3 sono essenziali: cancello le righe corrispondenti e i mintermini da essi coperti.

La copertura di f è data da:

C(f)={P1,P3}

	0	2	4	5	6	7	8	9	13	15
P1	X	X	X		X					
P2			X	X	X	X				
P3				X		X			X	X
P4	X						X			
P5							X	X		
P6								X	X	

Esercizio 1 - II fase (4/4)

	8	**9**
~~**P4**~~	~~X~~	
P5	X	X
~~**P6**~~		~~X~~

P5 domina P4 e P6 che vengono cancellate.

La copertura di f è data da:

C(f)={P1,P3,P5}

f = P1+P3+P5= !a!d+bd+a!b!c

Esercizio 2 (1/3)

Minimizzare con il metodo di Quine-McCluskey, la rete con quattro ingressi ed una uscita specificata come segue:

ONSet={4,10,11,13,14,15}; DCSet={3,5,6,7}

Soluzione:

Si considerino i valori degli ingressi delle configurazioni che costituiscono l'ONSet e il DCSet si ricava:

ONSet={0100,1010,1011,1101,1110,1111}

DCSet={0011,0101,0110,0111}

che dà origine alla seguente partizione:

{{0100}{1010,0011,0101,0110}{1011,1101,1110,0111}{1111}}

Esercizio 2 – I fase (2/3)

4	0100	✓
3	*0011*	✓
5	*0101*	✓
6	*0110*	✓
10	1010	✓
7	*0111*	✓
11	1011	✓
13	1101	✓
14	1110	✓
15	1111	✓

(*a*)

4,*5*	010–	✓
4,*6*	01–0	✓
3,7	*0–11*	✓
3,11	–011	✓
5,7	*01–1*	✓
5,13	–101	✓
6,7	*011–*	✓
6,14	–110	✓
10,11	101–	✓
10,14	1–10	✓
7,15	–111	✓
11,15	1–11	✓
13,15	11–1	✓
14,15	111–	✓

(*b*)

4,*5,6,7*	01––	***A***
3,7,11,15	––11	***B***
5,7,13,15	–1–1	***C***
6,7,14,15	–11–	***D***
10,11,14,15	1–1–	***E***

(*c*)

Esercizio 2 – II fase (3/3)

	4	10	11	13	14	15
A	X					
B			X			X
C				X		X
D					X	X
E		X	X		X	X

F = A+C+E=

= !xy+yv+xz

Metodo di Quine-McCluskey: metodi di supporto alla copertura

Sulla tabella non ulteriormente riducibile (non presenta più ne essenzialità ne dominanze di riga o colonna, tabella detta ***ciclica***) occorre fare una scelta di copertura in base a criteri d'ottimo euristici secondo una preassegnata funzione di costo (ad es. metodo Branch&Bound) oppure ricorrere a metodi esatti algebrici, come quello di Petrick.

	x	y	z	w	
P	×			×	1
Q	×	×			3
R		×	×		3
S			×	×	2

Figura 4.30 Tabella di copertura ciclica

Metodo di Petrick

- trasforma il problema della copertura in un problema algebrico:
- si devono coprire tutti i mintermini
- ogni mintermine può essere coperto da più implicanti

Metodo di Petrick

- Metodo sistematico e non esaustivo per identificare coperture di implicanti primi minimi.
- Esprimere tutte le condizioni di copertura che devono essere soddisfatte dagli implicanti primi e dai mintermini in forma di prodotto di somme:
  - Un termine somma per ogni mintermine
  - Il termine somma è composto dagli implicanti che rappresentano una copertura del mintermine
- Convertire il prodotto di somme in somma di prodotti applicando le leggi dell'algebra Booleana.
- Ogni termine prodotto che contiene il minimo numero di letterali specifica una copertura di implicanti primi minima e rappresenta quindi una possibile soluzione di copertura.

Metodo di Petrick - esempio

	A	B	C	D	E
P0	x	x			
P1		x	x		x
P2			x	x	
P3	x			x	x

Il significato della tabella di copertura è il seguente:
per rispettare la funzionalità (*vincolo*)
si deve coprire Il *mintermine* 0, mediante P0 o (OR) mediante P3, e (AND)
si deve coprire Il *mintermine* 3, mediante P0 o (OR) mediante P1, e (AND)
si deve coprire Il *mintermine* 10, mediante P1 o (OR) mediante P2, e ...

Da un *prodotto di somme*

$$(P_0+P_3)*(P_0+P_1)*(P_1+P_2)*(P_2+P_3)*(P_1+P_3)=1$$

$$(P_0+P_3\ P_1)*(P_1P_3+P_2)*(P_1+P_3)=1$$

$$(P_0P_2+P_3\ P_1)*(P_1+P_3)=1$$

Ad una *somma di prodotti*

$$(P_0P_2P_1+P_0P_2P_3+P_3P_1)=1$$

Gruppi di implicanti primi: $P_0P_2P_1$; $P_0P_2P_3$; $\boxed{P_3\ P_1}$