Corso di Calcolatori Elettronici I

Minimizzazione

Dipartimento di Informatica e Sistemistica
Università Degli Studi di Napoli Federico II

1

Forme Ridotte

- Vantaggi di una forma “ridotta”
  - Maggiore compattezza dal punto di vista dell’algebra della logica
  - Realizzazione più economica dal punto di vista dei circuiti

2

Funzioni di costo

- ***Costo di letterali*** **(*CL*)*:*** numero delle variabili indipendenti della funzione, contate sul numero di volte in cui esse compaiono nella forma. Equivale, per reti bilaterali, al numero di contatti adoperati.
- **Costo di funzioni o di porte (*CP*)*:*** numero delle funzioni elementari f_i che la compongono. Equivale, per le reti unilaterali, al numero complessivo di porte adoperate.
- **Costo di ingressi (*CI*):** numero delle funzioni f_i che la compongono, ciascuna moltiplicata per le variabili (dipendenti o indipendenti) di cui è funzione. Equivale, per le reti unilaterali, al numero complessivo di porte adoperate, ciascuna "pesata" per il numero di ingressi che possiede.

3

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**

4

Costo Forma Minima: esempio

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

CL=11; CP=7; CI=17

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

CL=4; CP=2; CI=5

Alcuni presupposti teorici (1/2)

- 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

7

Dimostrazione (2/2)

- *Una funzione $y = f(x_1, x_2, \ldots, x_n)$ 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

- 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).

9

Nucleo e Residuo

- 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

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

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

  con N e **R (insieme degli implicanti primi non essenziali)** eventualmente nulli.

10

Metodo di Quine-McCluskey

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

11

Ricerca degli implicanti

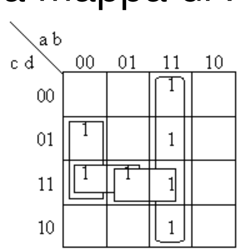
- Tre alternative:
  - Algebrica (Quine)
  - Tabellare (McCluskey)
  - Grafica (Karnaugh)

12

Metodo di Karnaugh

$$f = abcd + \bar{a}bcd + \bar{a}\bar{b}cd + \bar{a}\bar{b}\bar{c}d + ab\bar{c}d + abc\bar{d} + ab\bar{c}\bar{d}$$

- Gli implicanti primi sono individuati graficamente come sottocubi di area massima nella mappa di Karnaugh



Primi implicanti

$bcd,\ \bar{a}cd,\ \bar{a}\bar{b}d,\ ab$

20

Matrice di Copertura e Copertura minima

- Individuati gli implicanti primi, occorre scegliere tra di essi un insieme minimo che consenta di “coprire” ancora la funzione

	abcd	0001	0011	1100	0111	1101	1110	1111
A	00-1	1	1					
B	0-11		1		1			
C	-111				1			1
D	11--			1		1	1	1

Copertura minima

- Un modo generale per definire il problema della copertura è il seguente:
  - data una matrice di *N* righe e *M* colonne, i cui elementi siano $a_{ij} = 1$ oppure $a_{ij} = 0$, si dice che una riga *i copre* una colonna *j* se $a_{ij} = 1$. Si selezioni il numero minimo di righe che coprono tutte le colonne.
- Il problema della copertura è di interesse generale in molti settori differenti (ad esempio, in applicazioni di testing)

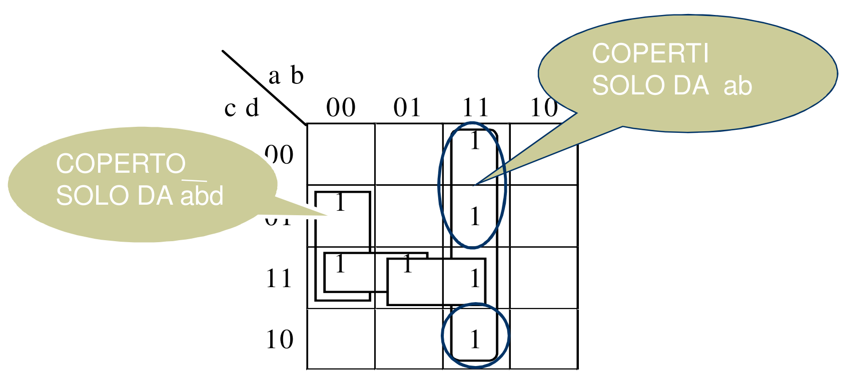
Individuazione del nucleo

- Per quanto visto prima, il primo passo da fare per individuare la copertura è selezionare l'insieme degli implicanti primi essenziali (il ***nucleo***).
- Tali implicanti devono infatti essere sempre presenti e la loro selezione è univoca
- Nella matrice di copertura vengono individuati in corrispondenza di colonne con un unico '1'
- Nell'esempio corrente, N = A + D

23

Individuazione del nucleo con le mappe di Karnaugh

- Sulle mappe di Karnaugh la selezione del nucleo è immediata:
  - sono essenziali gli implicanti che sono gli unici a ricoprire un "1"



Metodi di Copertura minima

- Trovare la copertura minima vuol dire trovare la forma minima di R, cioè quegli implicanti che, pur non essendo essenziali, devono essere eventualmente aggiunti al nucleo per trovare una forma che "copra" tutti i mintermini.
- La selezione degli implicanti primi non essenziali è arbitraria.
- La scelta deve essere:
  - Ottima secondo le prefissate funzioni di costo
  - Ottenibile in maniera computazionalmente efficiente
- Due fondamentali alternative
  - Algebrica (Petrick)
  - Tabellare (righe/colonne dominanti)

25

Metodo di Petrick

- Esprime la condizione algebrica secondo la quale *TUTTE* le colonne devono essere coperte da *ALMENO UN* implicante.
- Può allora essere espressa come una AND di OR.
- Tale forma può essere trasformata in una OR di AND che esprime tutte le possibili alternative per la soluzione del problema di copertura.

26

Metodo di Petrick: esempio

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9
A	1		1		1		1	1	1
B	1	1		1			1	1	
C		1				1			1
D			1				1		
E				1		1			
F					1			1	
G						1			1

(A+B)(B+C)(A+D)(B+E)(A+F)(C+E+G)(A+B+D)(A+B+F)(A+C+G)=

=ACE + ABC + ABG + ABE + BCDF + BDFG

per risolvere il problema di copertura possono essere usati {A,C,E}, oppure {A,B,C}, oppure…

27

Righe/Colonne dominanti

- Chiamiamo "***linea***" indifferentemente una riga o una colonna
- Una linea *L* domina la linea *K* se la "include", ovvero se contiene tutti i suoi *1*

La colonna di destra domina quella di sinistra

1	1	0	1	0
0	1	0	1	0

La riga in alto domina l'altra

28

Righe/Colonne dominanti

- Se si eliminano le **righe dominate** e le colonne **dominanti**, da una matrice di copertura, se ne trae una equivalente (che rappresenta, cioè, il medesimo problema di copertura)

29

Righe/Colonne dominanti

- Il metodo tabellare per righe/colonne dominanti 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.

30

Esempio

	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

37

Esempio

	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

38

Esempio

	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

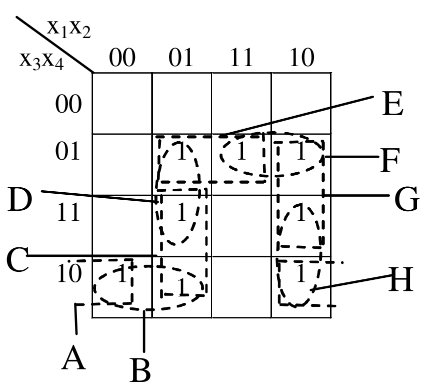
riga D dominata dalla F

Colonna P_9 dominante

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

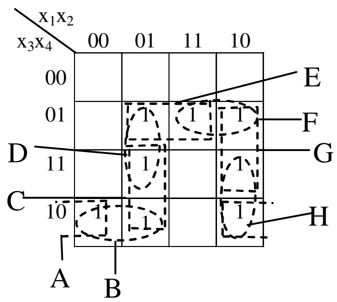
Righe A e B dominate dalla **C**

Funzione ciclica



N = ?

Funzione ciclica



	P_3	P_6	P_7	P_8	P_9	P_{10}	P_{13}	P_{14}
A				1		1		
B				1	1			
C					1		1	
D	1						1	
E	1		1					
F		1	1					
G		1						1
H						1		1

Funzione ciclica

- Metodo di Petrick:

  (A+B)(B+C)(C+D)(D+E)(E+F)(F+G)(G+H)(H+A)=
  =ABCDEFGH+BCDEFGH+…+BDFH+…+ACEG+…=
  = BDFH+ACEG

  e quindi si individuano due coperture minime:
  {B,D,F,H} e {A,C,E,G}.

Ricapitolando

Specifica della funzione
(tabella di verità)

↓

Ricerca di tutti gli implicanti primi

↓

Estrazione degli implicanti primi essenziali

↓

Selezione di un sottoinsieme dei restanti implicanti che determina una copertura minima

metodo algebrico (Quine), tabellare (McKlusky), mappe di Karnaugh

metodo algebrico, righe/colonne dominanti, metodo di Petrick

44

Presenza di don’t care

- I don’t care possono essere sfruttati per minimizzare ulteriormente la struttura di una funzione logica

  → *si può cercare tra tutte le funzioni compatibili quella che ha costo minimo*

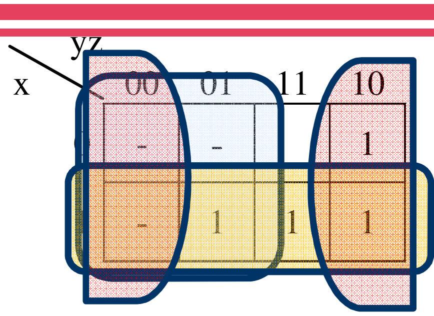
45

Presenza di don’t care

- Notate che ‘1’ nella tabella di verità consentono di ottenere implicanti più “ampi”
- D’altro canto, un maggior numero di ‘0’ nella tabella di verità riduce il numero di mintermini da coprire
  - → conviene considerare i don’t care come ‘1’ quando si cercano gli implicanti, e come ‘0’ quando si ricerca la copertura

46

Esempio



implicanti primi ottenuti considerando i don’t care pari a 1

	P_2	P_5	P_6	P_7
x		1	1	1
$\overline{y}$		1		
$\overline{z}$	1		1	

matrice di copertura considerando i don’t care pari a 0

$$N = x + \overline{z} \;\Rightarrow\; y_{\min} = x + \overline{z}$$

47

XOR ED EQ

51

Funzioni XOR ed EQ

- Le funzioni XOR (⊕) ed EQ (≡) assumono una notevole rilevanza dal punto di vista pratico
  - XOR: rappresenta la cifra-somma delle cifre-addendi x ed y, *somma modulo 2.*
    - Assume particolare importanza in aritmetica
  - EQ: consente di generare una variabile che indica se due bit sono uguali o diversi.

$$x \oplus y = \overline{x \equiv y} = x\overline{y} + \overline{x}y = (x + y)(\overline{x} + \overline{y})$$

$$x \equiv y = \overline{x \oplus y} = xy + \overline{x}\,\overline{y} = (x + \overline{y})(\overline{x} + y)$$

Funzioni XOR e EQ

- Applicando le regole generali di trasformazione in forma NAND e NOR, si ha:

XOR IN FORMA NAND (2 liv)	$x \oplus y = (x \uparrow \overline{y}) \uparrow (\overline{x} \uparrow y)$
XOR IN FORMA NAND (3 liv)	$x \oplus y = x(\overline{x} + \overline{y}) + y(\overline{x} + \overline{y}) =$ $= (x \uparrow (x \uparrow y)) \uparrow (y \uparrow (x \uparrow y))$
XOR IN FORMA NOR	$(x \downarrow y) \downarrow (\overline{x} \downarrow \overline{y})$
EQ IN FORMA NAND	$(x \uparrow y) \uparrow (\overline{x} \uparrow \overline{y})$
EQ IN FORMA NOR	$(x \downarrow \overline{y}) \downarrow (\overline{x} \downarrow y)$

Funzione XOR

- Forme alternative:

$$x \oplus y = (x \uparrow \overline{y}) \uparrow (\overline{x} \uparrow y)$$

forma a 2 livelli (costo di porte **5** includendo anche la NOT)

$$x \oplus y = x(\overline{x} + \overline{y}) + y(\overline{x} + \overline{y}) =$$
$$= (x \uparrow (x \uparrow y)) \uparrow (y \uparrow (x \uparrow y))$$

forma a 3 livelli (costo di porte **4** riutilizzando un fattore comune)

Proprietà della XOR e di EQ

- A differenza di NAND e NOR, la XOR e la EQ non costituiscono un insieme funzionalmente completo
- Permettono però di ottenere una NOT

$$x \oplus 1 = x \cdot 0 + \overline{x} \cdot 1 = \overline{x}$$

$$x \equiv 0 = x \cdot 0 + \overline{x} \cdot 1 = \overline{x}$$

57

Funzioni parità e disparità (1/9)

Dato un insieme X di m variabili,

- *funzione parità di X, p(X)*

$$p(X) = p(x_1, x_2, \ldots, x_m)$$

  - vale “1" se è pari il numero di valori "1" di X, "0" altrimenti.
- *funzione disparità di X, d(X)*
  - vale “1" se è dispari il numero di valori "1" di X, "0" altrimenti.
  - il complemento della funzione di parità $d(X) = \overline{p(X)}$

Funzioni parità e disparità (2/9)

- Per n=2

$$d(x_1,x_2) = x_1 \oplus x_2$$
$$p(x_1,x_2) = x_1 \equiv x_2$$

- In generale si ha

$$d(X) = \sum_{i \in D} P_i$$

  dove D è l'insieme dei numeri interi che, espressi in aritmetica binaria, contengono un numero "dispari" di cifre "1";
- *Mintermine dispari*

$$P_i \text{ con } i \in D$$

Quindi:

d(X) è la sommatoria di tutti i mintermini dispari.

Funzioni parità e disparità (3/9)

Dualmente….

- *Mintermine pari* $P_i \text{ con } i \in P$

p(X) è la sommatoria di tutti i mintermini pari.

- Dei 2^n mintermini totali, metà sono mintermini pari e metà dispari, dunque:

  CP = $2^{(n-1)}$+1   CI = (n + 1) $2^{(n-1)}$

  sia per p(X) sia per d(X).

- p(X) e d(X) hanno forma minima coincidente con la forma canonica dal momento che non hanno mintermini adiacenti (Mappa di Karnaugh “a scacchiera”).

Funzioni parità e disparità (4/9)

- Per n=3:

$$d\left(x_1,x_2,x_3\right)= x_1\overline{x_2}\,\overline{x_3} + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\underline{x_2}x_3 + x_1x_2x_3$$

$$p\left(x_1,x_2,x_3\right)= \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + \overline{x_1}\,\overline{x_2}\,\overline{x_3}$$

- CP = 5
- CI = 16

Funzioni parità e disparità (4/9)

- Per n=3:

$$d\left(x_1,x_2,x_3\right)= x_1\overline{x_2}\,\overline{x_3} + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\underline{x_2}x_3 + x_1x_2x_3$$

$$p\left(x_1,x_2,x_3\right)= \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + \overline{x_1}\,\overline{x_2}\,\overline{x_3}$$

- Proprietà notevole:

$$d\left(\overline{X}\right)=\begin{cases} d\left(X\right) & \text{per n pari} \\ p\left(X\right) & \text{per n dispari} \end{cases}$$

infatti

  - se il numero di "1" è pari, per n pari lo è anche quello degli "0“ mentre per n dispari il numero degli "0" è dispari.

Funzioni parità e disparità (5/9)

La funzione duale della disparità d di n variabili è d stessa per n dispari, è la parità per n pari:

$$\delta_{d(X)}(X) = \begin{cases} p(X) & \text{per n pari} \\ d(X) & \text{per n dispari} \end{cases}$$

Dimostrazione

Applicando il teorema di Shannon e considerando la proprietà precedente:

$$\delta_{d(X)}(X) = \overline{d\left(\overline{X}\right)} = \begin{cases} \overline{d(X)} = p(X) & \text{per n pari} \\ \overline{p(X)} = d(X) & \text{per n dispari} \end{cases}$$

Funzioni parità e disparità (6/9)

- La funzione disparità è associativa

  d(X) = d(d(X1), d(X2).... d(Xn))

  Infatti se X è suddiviso in sottoinsiemi disgiunti, il numero di "1" di X è dispari se è dispari il numero di sottoinsiemi con un numero dispari di "1".

- La funzione di parità non è sempre associativa
  - *Applicando il teorema di Shannon*

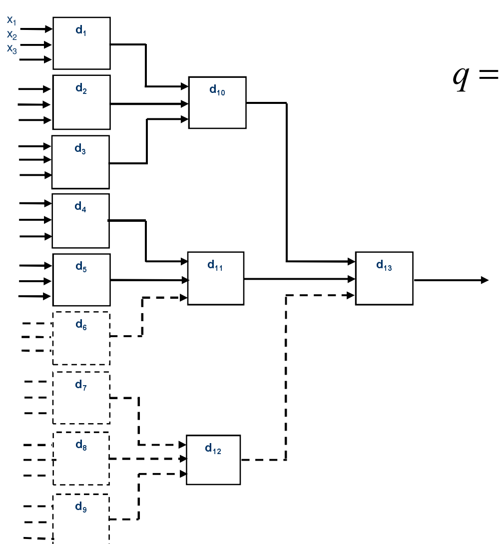
$$p(X) = \overline{d(X)} = \delta_{d(X)}\left(\overline{d(X_1)},\ldots,\overline{d(X_n)}\right) = \delta_{d(X)}\left(p(X_1),\ldots,p(X_n)\right)$$

È ancora la funzione di parità solo per n pari

Funzioni parità e disparità (7/9)

- L'associatività di d(X) consente spesso un risparmio nella realizzazione circuitale:
  $d(a, b, c) = d(d(a, b), c) = (a \oplus b) \oplus c = a \oplus b \oplus c$
- Per funzioni di *n* variabili si può adoperare una "struttura ad albero" (**l** livelli)
  - Se $n=k^l$ → $l=\log_k n$
    - La funzione d può essere realizzata adoperando esclusivamente componenti a *k* ingressi;
    - Ad ogni livello il numero delle variabili da trattare si divide per k quindi il numero di livelli è **l**
    - Ogni componente effettua una riduzione k→1 riducendo di k-1 il numero delle variabili da trattare

Funzioni parità e disparità (8/9)



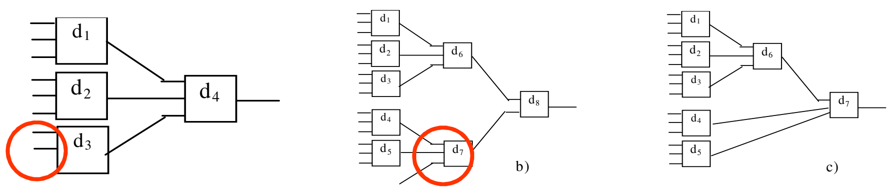
$$q = \left\{ \left\lceil \frac{n-1}{k-1} \right\rceil \right\} \qquad l = \{\lceil \log_k n \rceil\}$$

- *n*: numero di ingressi complessivi
- *k*: numero di ingressi delle funzioni componenti
- *l*: numero di livelli
- *q*: numero di porte componenti

70

Funzioni parità e disparità (9/9)

- Se $k^{\ell-1} < n < k^{\ell}$
  - si ha ancora una rete ad **l** livelli, ma l'albero completo di ingressi è "potato" nei suoi nodi e rami ed inoltre una componente potrebbe avere meno di k ingressi



8 variabili, l=2, q=4 16 variabili, l=3, q=8 15 variabili, l=3 q=7