Corso di Calcolatori Elettronici I
A.A. 2010-2011

Minimizzazione delle funzioni booleane tramite: prima parte

Lezione 9

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica (allievi A-DE+Q-Z)
Corso di Laurea in Ingegneria dell'Automazione

Funzioni incompletamente specificate

- Nei problemi di progetto, è possibile, in alcune circostanze, che il valore di una funzione booleana per alcune n-uple di valori delle sue variabili possa essere indifferentemente 0 o 1
  - Il valore può essere irrilevante ai fini del funzionamento del sistema descritto dalla funzione
  - Può esserci una dipendenza tra le variabili che esclude alcune combinazioni

Funzioni incompletamente specificate

- Si parla pertanto di "**punti di non specificazione**" o ***don't care***
- Due funzioni si dicono **compatibili** se assumono gli stessi valori, eccetto al più nei punti di non specificazione
- Se i punti di non specificazione sono k le funzioni compatibili sono 2^k
- Due funzioni compatibili "speciali"
  - f_0 = vale 0 **in tutti** i k punti di non specificazione
  - f_1 = vale 1 **in tutti** i k punti di specificazione

Funzione incompletamente specificata: esempio

A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	–	–	–	–
1	0	1	1	–	–	–	–
1	1	0	0	–	–	–	–
1	1	0	1	–	–	–	–
1	1	1	0	–	–	–	–
1	1	1	1	–	–	–	–

Tabella 3.3 - Tabella di decodifica da codice BCD a Eccesso 3. I trattini indicano condizioni di indifferenza.

da: G. Bucci. Calcolatori Elettronici – Architettura e organizzazione. © McGraw-Hill, 2009

Definizione di on-set , don't care set, off-set

- Sia $f(x_0, x_1, \ldots, x_{n-1})=f(X)$ una generica funzione di n variabili. Si definiscono i seguenti insiemi
  - On-set $\Sigma = \{X_i|f(X_i) = 1\}$
  - Don't care-set $\Delta=\{X_i|f(X_i)= - \}$
  - Off-set $\phi=\{X_i|f(X_i) = 0\}$

per cui valgono le relazioni

$\Sigma \cup \Delta \cup \phi = B^n$; $\Sigma \cap \Delta = \emptyset$; $\Sigma \cap \phi = \emptyset$; $\Delta \cap \phi = \emptyset$

- due dei tre insiemi sono sufficienti a definire in modo completo e univoco una generica funzione

Specifica formale con tabelle di verità

Tabella delle verità di una funzione: $f(B^4) \longrightarrow B$

x	y	z	v	o
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

x	y	z	v	o
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

- $o=f(x,y,z,v)=\Sigma\{1, 4, 5, 6, 7, 9, 11, 14, 15)$

Rappresentazione in forma canonica

PRIMA FORMA CANONICA (forma normale di tipo P o somma di prodotti)

- Una generica funzione a n variabili f(x1, …, xn) può essere espansa nella forma di disgiunzione di tutti i suoi mintermini.
- Un **mintermine** è un termine prodotto in cui compaiono tutte le variabili di ingresso corrispondenti alle combinazioni per cui la funzione booleana assume valore 1. In particolare il mintermine è composto dalle variabili che assumono valore 1 prese in forma naturale e da quelle che assumono valore 0 prese in forma complementata.

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

Forma P f=!x!yz + x!y!z + x!yz + xy!z + xyz

Rappresentazione in forma canonica

SECONDA FORMA CANONICA (forma normale di tipo S o prodotto di somme)

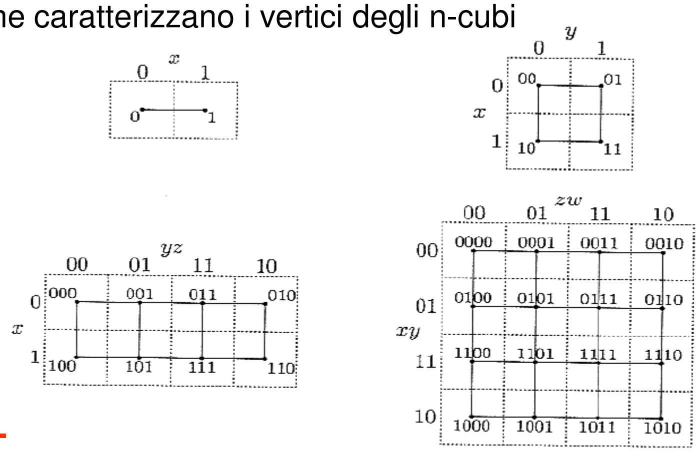
- Una generica funzione a n variabili f(x1,…,xn) può essere espansa nella forma di congiunzione di tutti i suoi maxtermini.
- Un **maxtermine** è un termine somma in cui compaiono tutte le variabili di ingresso corrispondenti alle combinazioni per cui la funzione booleana assume valore 0. In particolare il maxtermine è composto dalle variabili che assumono valore 0 prese in forma naturale e da quelle che assumono valore 1 prese in forma complementata.

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

Forma S f=(x+y+z)(x+!y+z)(x+!y+!z)

Rappresentazione mediante mappe di Karnaugh (1/2)

Una mappa di Karnaugh è una trasposizione di un n-cubo su due dimensioni che mantiene le proprietà di adiacenza che caratterizzano i vertici degli n-cubi



Rappresentazione mediante mappe di Karnaugh - esempi (2/2)

c \ a b	00	01	11	10
0	0	1	1	1
1	1	1	1	1

Mappa di Karnaugh della funzione OR(a, b, c)

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

Mappa di Karnaugh della funzione f(x,y,z)=xyz+xy!z+!xy!z

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 funzione (1/4)

- 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 funzione (2/4)

- 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) **C_L**=4 **C_P**=2 **C_I**=5
- f=bc(a!d+!b+c)+!c(d+!a)(b+c) **C_L**=11 **C_P**=7 **C_I**=17

Costo di una funzione (3/4)

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

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

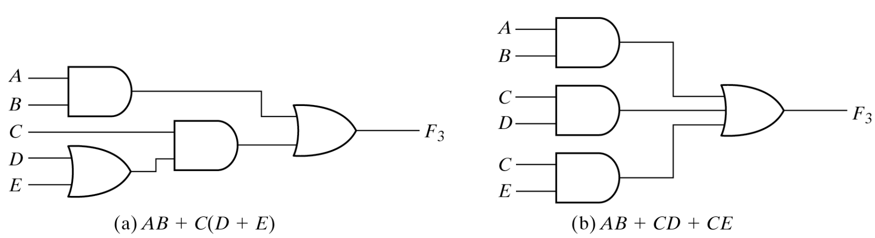


Fig. 2-4 Three- and Two-Level implementation

Costo di una funzione (4/4)

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)

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 + !xP = (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.
- Circuiti a 2 livelli e più uscite: come prima e metodo approssimato per trovare la copertura ottima basato 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 liv. di porzioni del circuito a 1 uscita.)

Metodi di Ottimizzazione

I Metodi di Ottimizzazione possono essere classificati in due macrocategorie:

- Metodi Esatti
  - Karnaugh
  - Quine-McCluskey
- Euristiche

Applicabili entrambi a reti a due o più livelli

Metodi esatti
Metodo delle Mappe di Karnaugh

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

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.

$x \backslash yz$	00	01	11	10
0	0	0	0	1
1	0	0	1	1

(a)

$x \backslash yz$	00	01	11	10
0	0	0	0	1
1	0	0	1	1

(b)

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

Il metodo delle Mappe di Karnaugh- Esempio 1

$x \backslash yz$	00	01	11	10
0	1	1	0	0
1	0	1	1	0

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

f(xyz)= !x!y+ !yz+ xz

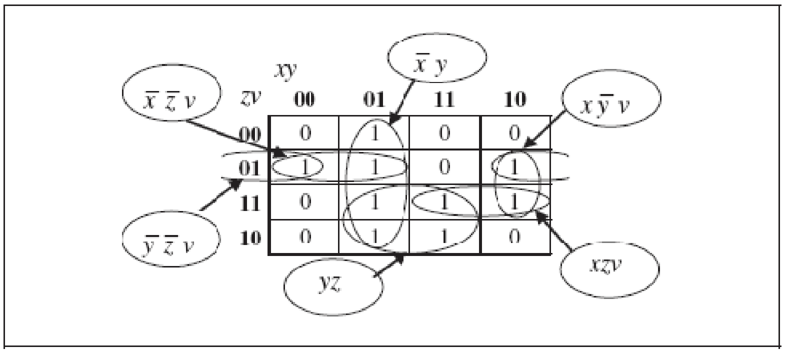
che non risulta minima.

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 !yz può essere trascurato. In definitiva quindi:

f(xyz)= !x!y+ xz

Il metodo delle Mappe di Karnaugh – Esempio 2

Mappa di Karnaugh per la funzione
f(x,y,z,v)=Σ{1, 4, 5, 6, 7, 9, 11, 14, 15)
I Primi Implicanti PI sono i sottocubi di area massima



Matrice di copertura (1/3)

1. Si costruisce la **matrice (o tabella) di copertura:** si conviene di porre sulle righe gli implicanti primi di f determinati nella prima fase 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 appartenenti al’ON-Set della funzione poiché non è necessario coprire le condizioni di indifferenza
   - **NB**: nel caso in cui si adopera la convenzione con mintermini sulle righe e implicanti sulle colonne vanno rivisti opportunamente i criteri di essenzialità e dominanza mostrati di seguito

Matrice di copertura (2/3)

2. Si individuano gli implicanti **essenziali**, ovvero quelli che sono i soli a coprire un dato mintermine (una sola “x” in una colonna): tali implicanti vengono inseriti nella copertura minima della funzione.

   Si genera una nuova tabella eliminando la riga corrispondente all’implicante essenziale e tutte le colonne dei mintermini da esso coperti nella tabella di partenza.

3. Si riesamina la nuova tabella prodotta finché non è più possibile individuare implicanti essenziali. A questo punto si procede col metodo della dominanza per determinare gli implicanti **essenziali secondari** (vedi dopo).

Matrice di copertura: criteri di dominanza

- La ***riga i domina la riga j*** se l’implicante Pi copre tutti i mintermini che copre l’implicante Pj più almeno uno
  - ✓ 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*** se il mintermine mj è coperto da un sottoinsieme degli implicanti che coprono mi
  - ✓ 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

Matrice di copertura (2/3)

4. Si eliminano dalla tabella le **righe dominate** e le **colonne dominanti** e si cercano eventuali implicanti essenziali secondari, ovvero quelli che verificano la condizione di essenzialità nella tabella ridotta: tali implicanti vengono inseriti nella copertura della funzione

5. Si ripete il passo 4 finchè non si assicura la copertura di tutti i mintermini della funzione

Matrice di copertura: 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}

Matrice di copertura: 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 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}