

Sintesi di reti logiche multilivello

M. Favalli

Engineering Department in Ferrara



Sommario

- 1 Introduzione
- 2 Algoritmi euristici
- 3 Aspetti tecnologici

Sommario

- 1 Introduzione
- 2 Algoritmi euristici
- 3 Aspetti tecnologici

Motivazioni

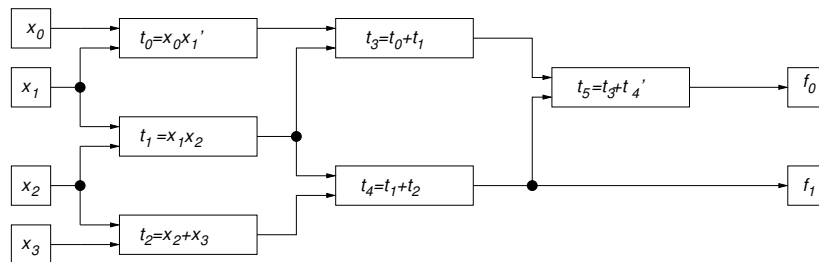
- Esistono numerose funzioni che non possono essere sintetizzate in maniera conveniente come reti a 2 livelli
- Di più, nonostante il ridotto numero di livelli dovrebbe corrispondere a un buon ritardo nell'elaborazione degli ingressi, tali reti presentano porte logiche con un fan-in molto elevato e quindi lente
- Si può verificare che le reti multilivello possono presentare costi decisamente inferiori
- Si consideri ad esempio l'espressione SP
$$f = abc + abd + bcd + acd$$
 che ha $I = 12$ o $G = 16$
- Fattorizzando si ottiene $f = ab(c + d) + cd(a + b)$ che ha costo $I = 8$ e $G = 12$

Approccio al problema

Per sintetizzare le reti multilivello é necessario disporre di:

- una metodologia per descrivere la rete
- una metrica per il costo (letterali)
- algoritmi per l'ottimizzazione

Esempio di modello di rete multilivello



Che equivale alle seguenti equazioni:

$$\begin{aligned}
 t_0 &= x_0 x_1' & t_1 &= x_1 x_2 & t_2 &= x_2 + x_3 \\
 t_3 &= t_0 t_1 & t_4 &= t_1 + t_2 & t_5 &= t_3 + t_4' \\
 f_0 &= t_5 & f_1 &= t_4
 \end{aligned}$$

Il costo é $l = 12$

Topologia

Grafo aciclico orientato (DAG) $G(V, E)$

- $V = \{v_1, v_2, \dots, v_n\}$ é l'insieme dei nodi che é partizionato in 3 sottoinsiemi
 - nodi di ingresso
 - nodi di uscita
 - nodi interni
- $E = \{e_1, e_2, \dots, e_k\}$ é l'insieme degli archi orientati
- Si suppone che ciascun nodo v_i sia rappresentato da un espressione a 2 livelli (SP) che utilizza come variabili i segnali dei nodi che presentano un arco verso v_i (supporto locale)

Ottimizzazione di reti multilivello

- Le reti multilivello possono essere ottenute per costruzione (funzioni aritmetiche) o da reti a 2 livelli tramite opportune trasformazioni
- Dal punto di vista dell'ottimizzazione, lo spazio delle possibili reti multilivello equivalenti a una data funzione é molto piú grande di quello delle reti a 2 livelli e pertanto sono possibili solo metodi di tipo euristico

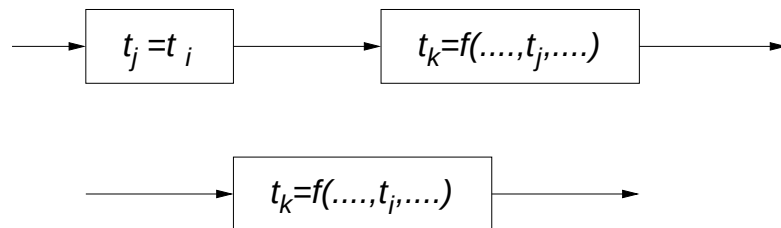
1 Introduzione

2 Algoritmi euristici

3 Aspetti tecnologici

Sweep

Si tratta di un'operazione con la quale si eliminano i nodi interni la cui espressione è semplicemente uguale a una variabile



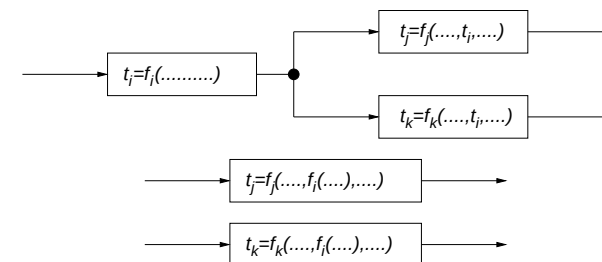
Si tratta di una trasformazione globale

Algoritmi

- Gli algoritmi euristici che operano su una rete multilivello compiono una serie di trasformazioni (con eventuali iterazioni)
- Trasformazioni locali: modificano le espressioni algebriche dei vari nodi e non la topologia della rete
- Trasformazioni globali: modificano sia le espressioni che la topologia della rete
- Similmente al caso delle reti a due livelli, alcune trasformazioni possono aumentare il costo (numero di letterali) della rete
- In particolare, esiste un compromesso fra il numero di nodi e la complessità della funzione corrispondente a ciascun nodo

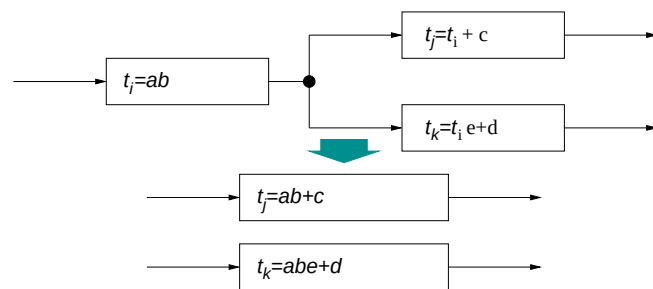
Eliminate

Si sostituisce l'espressione di un nodo in uno o piú nodi in cui essa compare e poi si elimina il nodo stesso



In generale questa trasformazione porta a un aumento del numero di letterali dell'espressione equivalente alla rete di partenza (viene attuata solo al di sotto di una certa soglia)

Esempio



Factor

- Fattorizzazione di un'espressione a due livelli corrispondente a un nodo (trasformazione locale)
- Si utilizzano euristici per individuare le variabili da raccogliere e l'ordine delle fattorizzazioni
- L'idea è di raccogliere le variabili che compaiono nel maggior numero di implicant nell'espressione SP
- Lo strumento è una tabella avente come righe gli implicant e come colonne i letterali
- Per ciascuna coppia impicante-letterale la tabella contiene un 1 se il letterale compare nel termine prodotto e 0 altrimenti
- L'ultima riga contiene la somma di ciascuna colonna

Simplify

- Nel caso in cui un nodo interno sia caratterizzato da un'espressione a più di due livelli, la si trasforma in un'espressione a 2 livelli
- Si applica un algoritmo esatto (ad es. Quine-McCluskey) o un euristico per semplificare tale espressione
- La rete a monte e quella a valle possono creare delle indifferenze sugli ingressi del supporto di tale funzione (se si tiene conto di tali indifferenze, la trasformazione viene definita full simplify)

Factor: esempio

Si consideri:

$$f = abd' + a'bd + a'b'd' + a'cd + b'cd'$$

f	a	a'	b	b'	c	c'	d	d'
abd'	1	0	1	0	0	0	0	1
$a'bd$	0	1	1	0	0	0	1	0
$a'b'd'$	0	1	0	1	0	0	0	1
$a'cd$	0	1	0	0	1	0	1	0
$b'cd'$	0	0	0	1	1	0	0	1
somma	1	3	2	2	2	0	2	3

Factor: esempio

Si scelga ad esempio d' :

$$f = d'(ab + a'b' + b'c) + a'bd + a'cd = d'f_1 + f_2$$

Il procedimento può essere iterato per le due nuove sottoespressioni

	f_1	a	a'	b	b'	c	c'
$f_1 = ab + a'b' + b'c$	ab	1	0	1	0	0	0
	$a'b'$	0	1	0	1	0	0
	$b'c$	0	0	0	1	1	0
	somma	1	1	1	2	1	0

$$f_1 = ab + a'b' + b'c$$

f_2	a	a'	b	b'	c	c'	d	d'
$a'bd$	0	1	1	0	0	0	0	1
$a'cd$	0	1	0	0	1	0	1	0
somma	0	2	1	0	1	0	2	0

$$f_2 = a'bd + a'cd$$

Substitute

- Si utilizza un nodo presente nella rete $t_j = f_d(\dots)$ per semplificare un altro nodo $t_i = f_i(\dots)$
- In particolare, si utilizza la divisione algebrica (di f_i per f_d) fra polinomi per ottenere: $f_i = f_q(\dots)f_d(\dots) + f_r(\dots)$
- Esempio: sia $f_i = ac + bc + ad + bd + e$ e $f_d = c + d$, si ha: $f_q = a + b$ e $f_r = e \Rightarrow f_i = (a + b)(c + d) + e$
- Se $f_r = 0$, f_d é un divisore di f_i . Se $f_q = 0$, la trasformazione é inutile

Factor: esempio

Nel primo caso si sceglie b' : $f_1 = b'(a' + c) + ab = b'f_3 + ab$, ove $f_3 = (a' + c)$

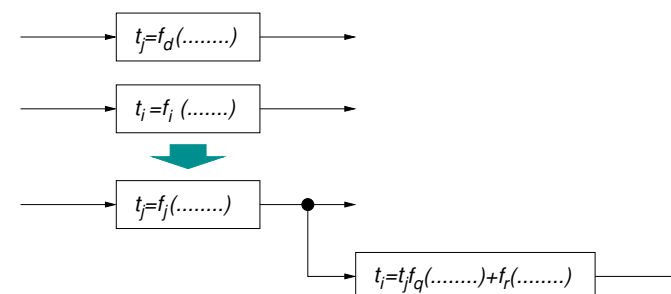
Nel secondo caso, a' : $f_2 = a'(bd + cd) = a'f_4$, ove $f_4 = bd + cd$.
In quest'ultima sottoespressione é evidente che si puó raccogliere d
da cui: $f_4 = d(b + c) = df_5$, ove $f_5 = b + c$

Infine si ottiene:

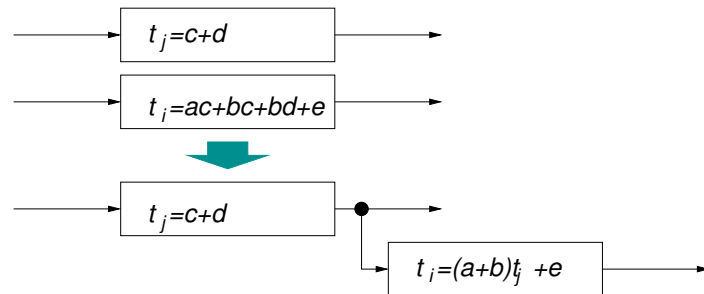
$$\begin{aligned} f &= d'f_1 + f_2 = \\ &= d'(ab + b'f_3) + a'f_4 = d'(ab + b'f_3) + a'(df_5)) \\ &= d'(ab + b'(a' + c)) + a'd(b + c) \end{aligned}$$

Oltre a ridurre i letterali, possono emergere sottoespressioni comuni a nodi già presenti nella rete

Substitute



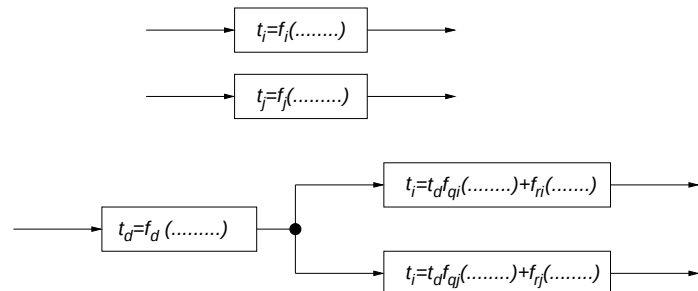
Substitute (esempio)



(ENDIF)

Reti logiche 21 / 36

Extract



(ENDIF)

Reti logiche 23 / 36

Extract

- Questa trasformazione esegue la sostituzione in più nodi di una funzione f_d che può anche non essere presente nella rete
- Questo viene fatto dall'operazione extract che utilizza tecniche piuttosto complesse per individuare divisori comuni a più nodi della rete
- Sia f_d uno di questi divisori comuni e siano f_i e f_j due candidati

$$f_i = f_d(\dots)f_{q_i}(\dots) + f_{r_i}(\dots)$$

$$f_j = f_d(\dots)f_{q_j}(\dots) + f_{r_j}(\dots)$$
- È una trasformazione globale che aumenta il numero di nodi riducendo i letterali

Reti logiche 22 / 36

Applicazione dell'extract

In pratica gli algoritmi che sfruttano l'operazione di extract considerano tutte le possibili coppie di nodi in cerca di divisori comuni:

- costituiti da un solo cubo con la factor
- costituiti dalla somma di due o più cubi

Le operazioni di divisione possono essere fatte in due modi:

- algebrico (divisione fra polinomi), ad esempio
 $t_i = ab + ac + bd + bc + e$ è divisibile per $b + c$ e quindi
 $t_i = (a + d)(b + c) + e$
- booleano (sfruttando proprietà come $a.a' = 0$ o $a.a = a$), ad esempio
 $t_i = a + bc + f$ è divisibile per $a + c$ e quindi
 $t_i = (a + b)(a + c) + f$

La divisione algebrica è molto più efficiente di quella booleana

(ENDIF)

Reti logiche 24 / 36

Divisione algebrica

- Data un'espressione x (forma normale SP), la si vuole dividere per $y = \sum c_i$ dove c_i è un implicante
- In pratica si divide $x \forall c_i$ ottenendo $q_i = x/c_i$ dove le q_i sono ancora forme normali SP. Quindi, $q_i \sum c_{i,j}$
- Il quoziente $q = x/y$ è dato dalla somma logica degli implicanti $c_{i,j}$ che compaiono in tutti i q_i
- Esempio: $x = ad + aef + bcd + bcef + ag$ e sia $y = a + bc$, quindi:

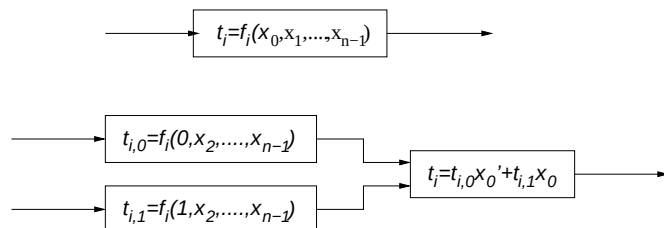
$$c_0 = a \Rightarrow q_1 = d + ef + g$$

$$c_1 = bc \Rightarrow q_2 = d + ef$$
- Quindi $q = d + ef \Rightarrow x = (d + ef)(a + bc) + ag$

Decompose

In questo caso si applica semplicemente il teorema di Shannon per estrarre da un singolo nodo 2^k nuovi nodi (ove k è il numero di variabili rispetto a cui si espande)

Produce nuovi nodi da utilizzare in altre trasformazioni



Esempio

Si verifichi se $a + c$ è estraibile algebricamente da $t_i = ac + ad + bd + bc + e$ e da $t_j = ab + ac + bf + cf + g$ e si tracci il grafo della rete dopo l'estrazione.

Ordine di applicazione delle trasformazioni

- La simplify viene tipicamente applicata dopo la eliminate, infatti quest'ultima operazione incrementa la complessità delle espressioni dei nodi
- Le trasformazioni di substitute ed extract possono essere eseguite in teoria dopo ogni passo anche se chiaramente possono sfruttare forme fattorizzate dei singoli nodi
- Esistono procedure più o meno standard per semplificare le reti multilivello

Esempio

Si considera una rete multilivello con gli ingressi a, b, c, d che implementa le funzioni f e g

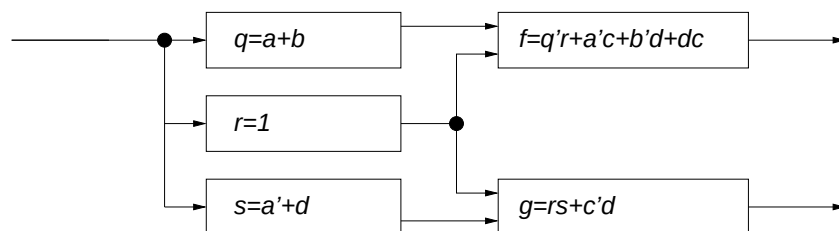
L'ordine di applicazione delle trasformazioni sia il seguente

- ① $\text{eliminate}(p)$; $\text{simplify}(q, r)$;
- ② $\text{eliminate}(q, r)$; $\text{simplify}(f, g)$;
- ③ $\text{factor}(f)$; $\text{substitute}(f)$;

In alcuni casi si può ridurre il numero di letterali o si genera nuove sottoespressioni utilizzabili nella decompose

Esempio - 2

$l=16$

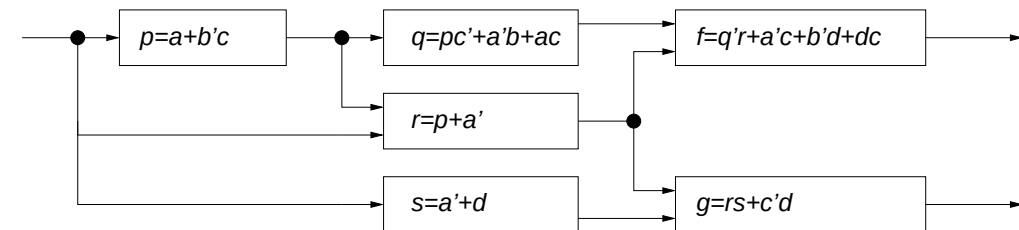


Eliminate: $f = (a + b)' \cdot 1 + a'c + b'd + dc$, $g = s \cdot 1 + c'd$

Simplify: $f = a'b' + a'c + b'd + dc$ e $g = s + c'd$

Esempio - 1

$l=25$

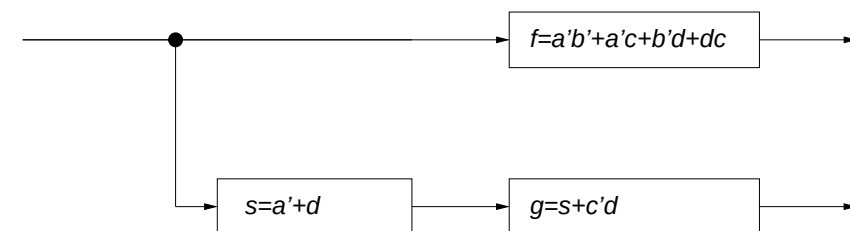


Eliminate: $q = (a + b'c)c' + a'b + ac$, $r = a + b'c + a'$

Simplify: $q = a + b$ e $r = 1$

Esempio - 3

$l=13$

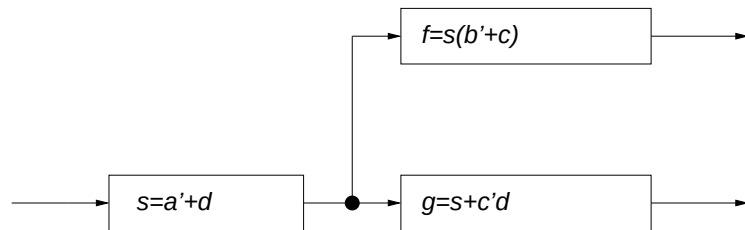


Factor: $f = a'(b' + c) + d(b' + c) = (a' + d)(b' + c)$

Substitute $f = s(b' + c)$

Esempio - 4

I=8



Technology mapping

- Non é detto che la tecnologia metta a disposizione porte AND, OR, NOT con qualsiasi fan-in
- In taluni casi, sono piú convenienti gate come il NAND o il NOR
- Per cui si ha un ultimo passo detto *Technology Mapping* in cui si hanno come ingressi la rete multilivello ottenuta dalla sintesi e la libreria di gate messi a disposizione dalla tecnologia
- Il technology mapping si occupa di "mappare" la rete multilivello sui gate nella libreria (sempre con qualche criterio di costo dato da una stima dell'area)

Sommario

- 1 Introduzione
- 2 Algoritmi euristici
- 3 Aspetti tecnologici