

---

# Minimizzazione di Reti Logiche Combinatorie Multi-livello

Maurizio Palesi

## Introduzione

---

- I circuiti logici combinatori sono molto spesso realizzati come **reti multi-livello** di porte logiche
  - Aumento dei gradi di libertà per l'ottimizzazione
    - ✓ Sfruttamento del trade-off area/ritardo
    - ✓ Soddisfare i vincoli tecnologici
  - Difficoltà di modeling e ottimizzazione
    - ✓ Metodi esatti: praticamente non attuabili
    - ✓ Euristiche (2 passi)
      - Ottimizzazione trascurando i vincoli (semplici modelli per area e prestazioni)
      - I vincoli sono presi in considerazione (library binding)
  - Fattorizzazione

## Fattorizzazione

$z\bar{v}$ \ $xy$	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

$$f = \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}}$$

- Corrispondente ad un circuito costituito da **8 porte AND a 4 ingressi** e **1 porta OR a 8 ingressi**
  - Raramente disponibili in una libreria
  - Caratterizzati da ritardi elevati

## Fattorizzazione

$$f = \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}} + \underline{xy}\underline{z\bar{v}}$$

- Applicando la proprietà distributiva del prodotto rispetto alla somma

$$f = \underline{xy}(\underline{z\bar{v}} + \underline{z\bar{v}}) + \underline{xy}(\underline{z\bar{v}} + \underline{z\bar{v}}) + \underline{xy}(\underline{z\bar{v}} + \underline{z\bar{v}}) + \underline{xy}(\underline{z\bar{v}} + \underline{z\bar{v}})$$

- Riapplicando nuovamente la stessa proprietà

$$f = (\underline{xy} + \underline{xy})(\underline{z\bar{v}} + \underline{z\bar{v}}) + (\underline{xy} + \underline{xy})(\underline{z\bar{v}} + \underline{z\bar{v}})$$

- Ricordando che  $(\underline{ab} + \underline{ab})' = (\underline{ab} + \underline{ab})$

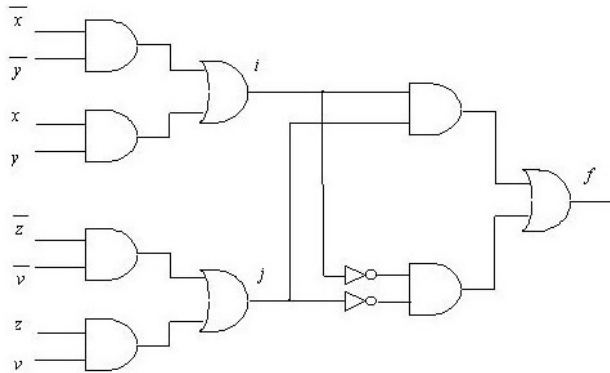
$$i = (\underline{xy} + \underline{xy})$$

$$j = (\underline{z\bar{v}} + \underline{z\bar{v}})$$

$$f = \underline{ij} + \underline{ij}$$

## Fattorizzazione

- Costo della rete ancora di 9 porte logiche
  - Ma tutte le porte sono a 2 ingressi
- Numero di letterali da 32 a 12



Maurizio Palesi

5

## Fattorizzazione

- La tecnica di fattorizzazione, se applicata manualmente, implica una certa misura di **intuito** (o di fortuna) da parte del progettista
  - Deve sapere **scegliere** nel modo migliore i termini **rispetto a cui fattorizzare** e l'ordine
  - Spesso occorre effettuare una fase di **espansione** (Teorema di Shannon) prima di fattorizzare
- Utilizzo di strumenti di progettazione automatica

Maurizio Palesi

6

## Modelli di Reti Logiche

- Il comportamento di un circuito combinatorio a  $n$  ingressi ed  $m$  uscite può essere espresso da un vettore di funzioni Booleane:

$$f_i: B^n \rightarrow \{0, 1, *\}, i=1, 2, \dots, m$$

- Tale funzione, che può essere *non* completamente specificata, rappresenta una corrispondenza esplicita tra lo spazio degli ingressi primari e lo spazio delle uscite primarie
- La *struttura* di un circuito combinatorio multi-livello, in termini di interconnessione di porte logiche, può essere descritto da una *rete logica*
- Una *rete logica* è una struttura che collega dei *moduli* (porte di I/O e porte logiche) attraverso reti di interconnessione

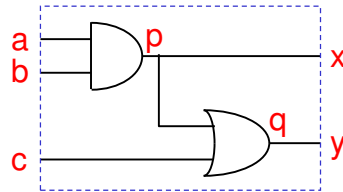
## Modelli di Reti Logiche (cont.)

- Una rete logica può essere rappresentata da un *DAG* (*Directed Acyclic Graph*) nel quale i vertici corrispondono ai moduli e i lati rappresentano reti a due terminali, nelle quali le reti originali a terminale multiplo sono state ridotte
- Una rete logica i cui moduli interni corrispondano a porte logiche appartenenti ad una libreria viene chiamata *rete logica mappata* (*bounded or mapped logic network*)
- Il comportamento di un circuito può essere rappresentato attraverso strutture equivalenti. Al contrario, un unico comportamento può essere derivato dalla struttura di un circuito

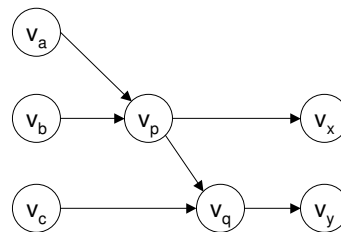
## Esempio di Rete Logica

Comportamento logico di I/O  
 $x = ab$   
 $y = c + ab$

Rete logica mappata



Grafo della rete logica



Maurizio Palesi

9

## Modelli di Reti Logiche

- Una rete logica *non* gerarchica rappresentata dal grafo  $G_n(V,E)$  è costituita da:
  - Un insieme di vertici  $V$  partizionato in 3 sotto-insiemi
    - ✓  $V^I$  vertici relativi a ingressi primari e  $n_i = |V^I|$  numero degli ingressi primari
    - ✓  $V^O$  vertici relativi a uscite primarie e  $n_o = |V^O|$  numero delle uscite primarie
    - ✓  $V^G$  vertici interni e  $n_g = |V^G|$  numero dei vertici interni
    - ✓ Ogni vertice è etichettato da una variabile
  - Un insieme di funzioni booleane combinatorie scalari associate ai vertici interni
- Gli *invertitori* sono impliciti nel modello e non sono rappresentati. In pratica, ogni vertice può fornire segnali di entrambe le polarità (*rete logica a doppia polarità*)

Maurizio Palesi

10

## Modelli di Reti Logiche

### Esempio

- Si consideri la rete logica con variabili di ingresso primarie  $\{a,b,c,d,e\}$ , variabili di uscita primarie  $\{w,x,y,z\}$  descritta dalle seguenti equazioni

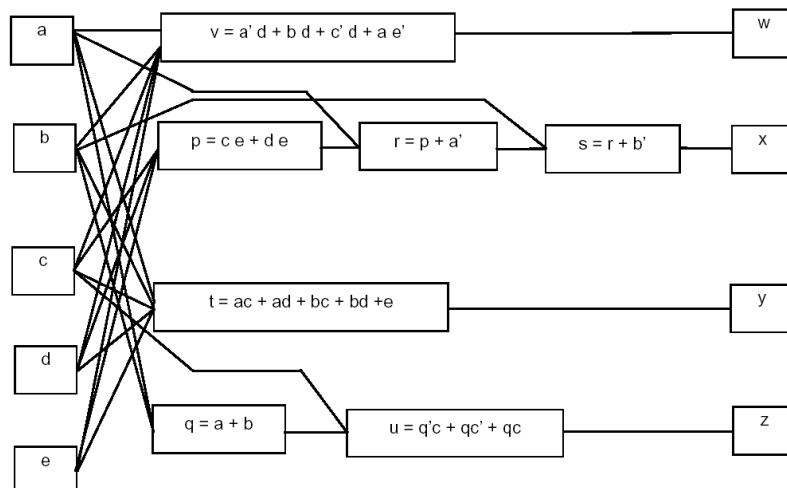
$$\begin{aligned}p &= ce + de \\q &= a + b \\r &= p + a' \\s &= r + b' \\t &= ac + ad + bc + bd + e \\u &= q'c + qc' + qc \\v &= a'd + bd + c'd + ae' \\w &= v \\x &= s \\y &= t \\z &= u\end{aligned}$$

Maurizio Palesi

11

## Modelli di Reti Logiche

### Esempio - Rappresentazione



Costo associato alla rete logica =  $(8 + 8 + 9 + 8)$  letterali = **33 letterali**

Maurizio Palesi

12

## Ottimizzazioni di Reti Logiche

- Gli obiettivi dell'ottimizzazione logica a due livelli e multi-livello sono differenti
  - Per **logica a due livelli** rappresentata come somma di prodotti
    - ✓ Area e ritardo sono proporzionali alla dimensione della copertura
    - ✓ Ottenere una copertura minima corrisponde ad ottimizzare **sia area sia ritardo**
  - Per **logica multi-livello**, implementazione ad area minima non corrisponde a implementazione a ritardo minimo e viceversa (es. sommatori)
    - ✓ Compromesso tra area e ritardo



Maurizio Palesi

13

## Ottimizzazione a più Livelli

- **Obiettivo**: ridurre **area** e **ritardo**
  - Problema di ottimizzazione multi-obiettivo
  - Nella pratica si identificano punti di bilancio
    - ✓ Minimizzare area (imponendo vincoli su ritardo)
    - ✓ Minimizzare ritardo (imponendo vincoli su area)

Maurizio Palesi

14

## Stima dell'Area

---

- L'area occupata da una rete logica multi-livello è proporzionale al numero di porte logiche e alle interconnessioni (wiring)
  - L'area delle porte logiche è definibile una volta che si conosca la libreria tecnologica
    - ✓ Valutabile parametricamente in base al numero di ingressi
    - ✓ In base al numero di porte logiche equivalenti (*NAND2*) che implementano la corrispondente funzionalità logica e al numero di letterali
  - L'area dovuta ai collegamenti è molto più difficile da stimare
    - ✓ Proporzionale al numero di letterali

## Stima del Ritardo

---

- Ritardo proporzionale al numero di livelli logici e alle interconnessioni
  - Nel caso di *bounded network* (reti mappate su una libreria tecnologica), il ritardo di ogni singola porta logica è specificato
  - Altrimenti il ritardo è stimato in base al ritardo associato ad ogni vertice (es. ritardo unitario per ogni vertice)
- Modelli di ritardo più sofisticati tengono conto del fan-out e delle interconnessioni associati ai vertici
- Ottimizzazione in *timing* = Ridurre il ritardo associato al percorso più lungo detto *percorso critico*

## Ottimizzazione Multi-livello: Metodi

---

### ■ Metodi esatti

- Elevata complessità computazionale
- Non applicabili ai casi reali

### ■ Metodi approssimati

- Metodi euristici basati sull'applicazione iterativa di *trasformazioni* che preservano il comportamento di I/O
- L'esecuzione di trasformazioni in qualunque sequenza salvaguarda l'*equivalenza* della rete logica
- Metodi che differiscono per
  - ✓ Tipo delle trasformazioni
  - ✓ Selezione e ordine delle trasformazioni

## Ottimizzazione Multi-livello

---

### ■ Problema della sintesi multi-livello

- Trovare un'appropriata sequenza di trasformazioni da applicare alla rete logica
  - ✓ Una rete logica viene dichiarata *ottima* in area e ritardo rispetto ad un insieme di trasformazioni **quando l'applicazione di queste non può più migliorare la funzione di costo**

## Ottimizzazione Multi-livello

---

### ■ Le trasformazioni

- Si valutano utilizzando delle cifre di merito
  - ✓ In modo da scartare le trasformazioni non convenienti
- Si applicano in modo iterativo
  - ✓ Il procedimento termina quando nessuna ulteriore applicazione di queste la migliora
- Per ogni trasformazione è definito un algoritmo
  - ✓ Dove la trasformazione può essere applicata?
  - ✓ Termina quando nessuna trasformazione dello stesso tipo può essere applicabile
  - ✓ Gli algoritmi legati a trasformazioni diverse vengono applicati in sequenza
- Sequenze di applicazione diversa portano a risultati diversi
  - ✓ Script di sintesi

## Trasformazioni Algebriche

---

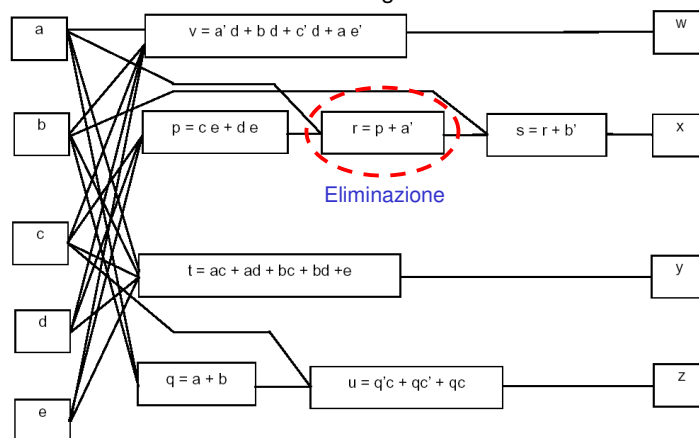
- Sweep
- Eliminazione
- Decomposizione
- Estrazione
- Semplificazione
- Sostituzione

## Sweep

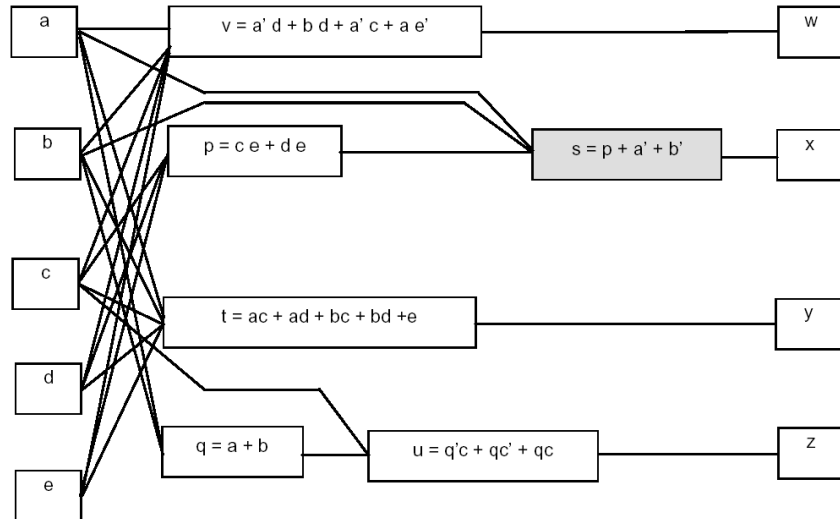
- Elimina dalla rete
  - I nodi con un solo ingresso
  - I nodi le cui funzioni danno valore costante
- Viene richiamata a valle di altre trasformazioni

## Eliminazione

- L'eliminazione di un vertice interno è la sua rimozione dalla rete. La variabile corrispondente al vertice è rimpiazzata dalla corrispondente espressione in tutte le sue occorrenze nella rete logica



## Eliminazione (cont.)

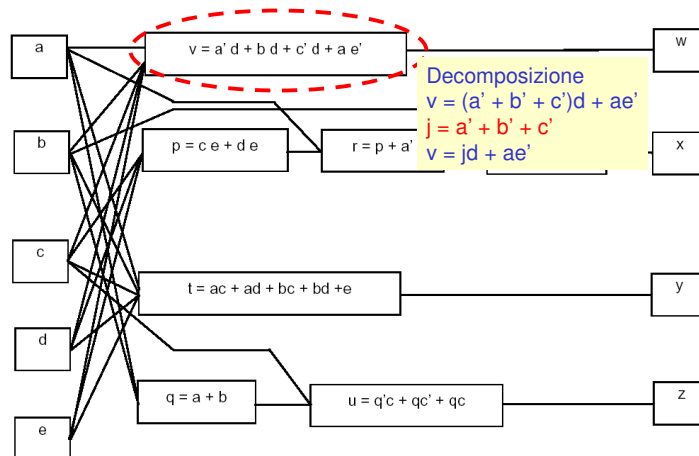


Maurizio Palesi

23

## Decomposizione

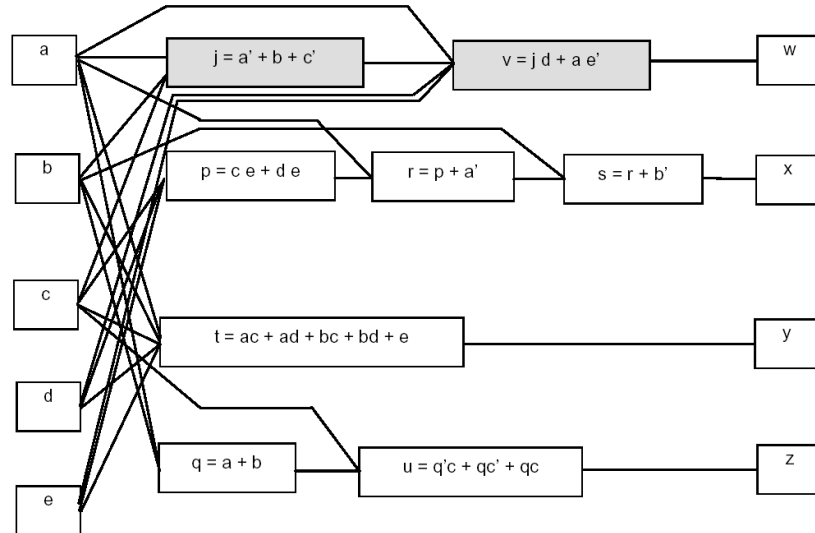
- La decomposizione di un vertice interno è la sostituzione del vertice con due (o più) vertici che formano una sottorete equivalente al vertice originale



Maurizio Palesi

24

## Decomposizione (cont.)

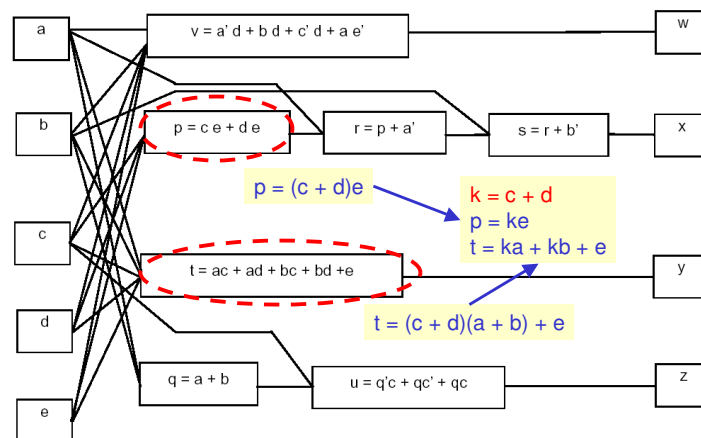


Maurizio Palesi

25

## Estrazione

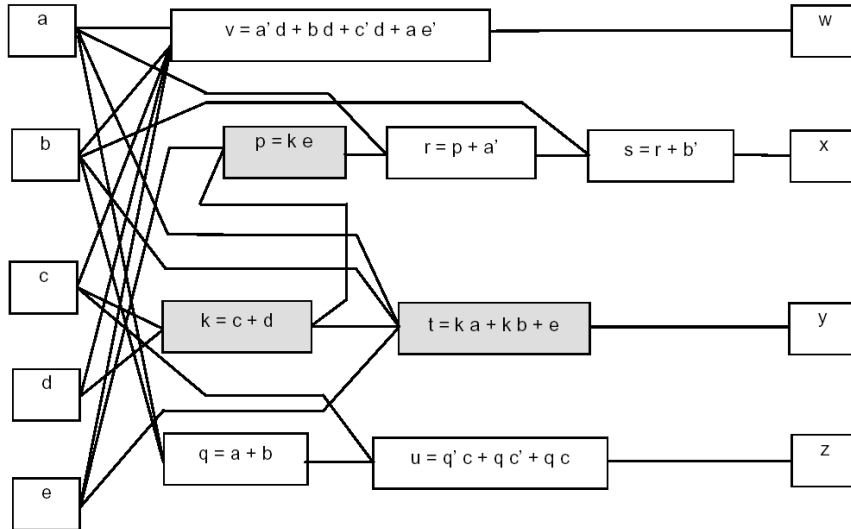
- Una sotto-espressione comune a due funzioni associate a due vertici può essere estratta creando un nuovo vertice associato alla sottoespressione



Maurizio Palesi

26

## Estrazione (cont.)

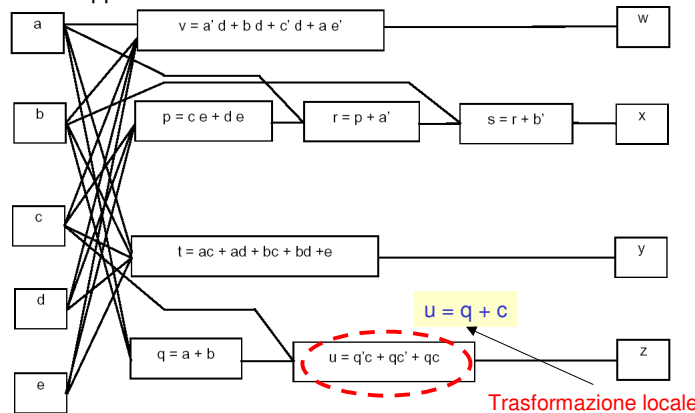


Maurizio Palesi

27

## Semplificazione

- Una funzione è ridotta in complessità sfruttando le proprietà della sua rappresentazione. Se la funzione è rappresentata nella forma a due livelli allora le tecniche di ottimizzazione a due livelli possono essere utilizzate. Se l'insieme di supporto non cambia allora la trasformazione si dice locale

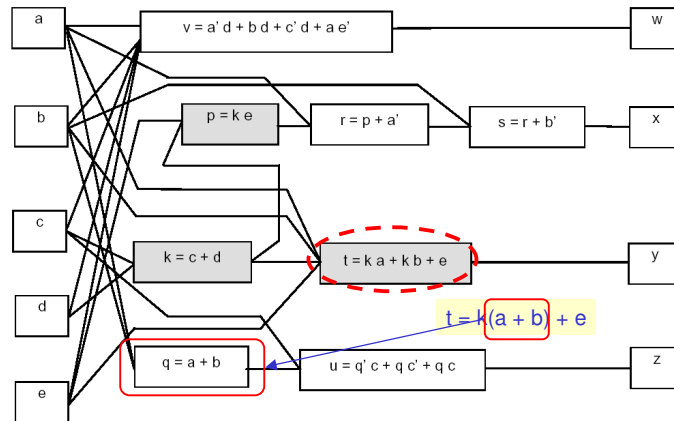


Maurizio Palesi

28

## Sostituzione

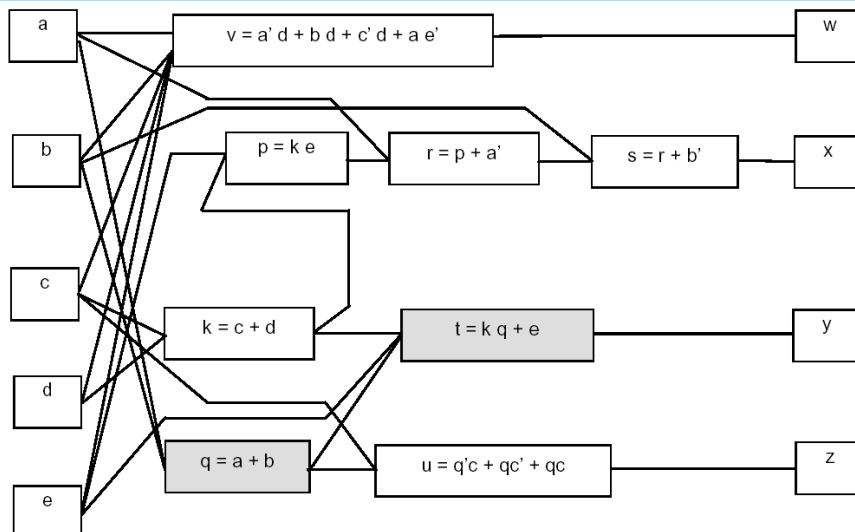
- Una funzione è ridotta in complessità utilizzando un ingresso aggiuntivo che non appartiene all'insieme di supporto. La trasformazione richiede la creazione di una dipendenza ma può anche portare ad eliminarne altre



Maurizio Palesi

29

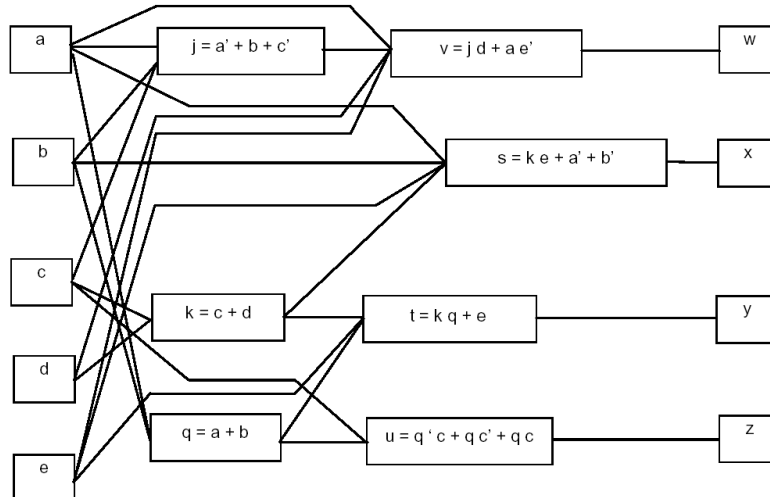
## Sostituzione (cont.)



Maurizio Palesi

30

## Risultato delle Trasformazioni



Costo associato alla rete logica trasformata =  $(7 + 4 + 5 + 8)$  letterali = **24 letterali**

Maurizio Palesi

31

## Risultato delle Trasformazioni

$$\begin{aligned}
 k &= c + d \\
 q &= a + b \\
 s &= ke + a' + b' \\
 t &= kq + e \\
 u &= q'c + qc' + qc \\
 v &= jd + ae' \\
 w &= v \\
 x &= s \\
 y &= t \\
 z &= u
 \end{aligned}$$

- Rispetto alla rete logica di riferimento il numero totale dei letterali è stato ridotto da 33 a 24

Maurizio Palesi

32

## Trasformazioni Booleane

### ■ Idea di base

→ Associare ad ogni nodo della rete

✓ Non solo la funzione booleana locale

✓ ...ma anche un insieme di *condizioni di indifferenza locali*

– Si considerano le relazioni tra il singolo nodo e l'intera rete

### ■ Condizioni di indifferenza esterne

→ Di controllabilità di ingresso

→ Di osservabilità di uscita

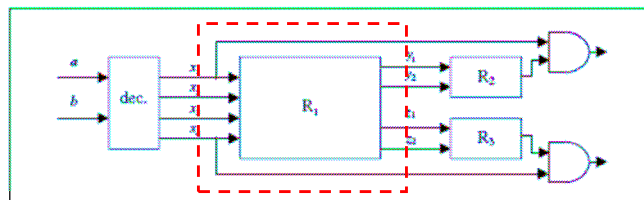
## Condizioni di Indifferenza Esterne

### ■ Di controllabilità di ingresso

→ Controllability don't care ( $CDC_{in}$ )

→ Configurazioni di ingresso che non vengono mai prodotte dall'ambiente

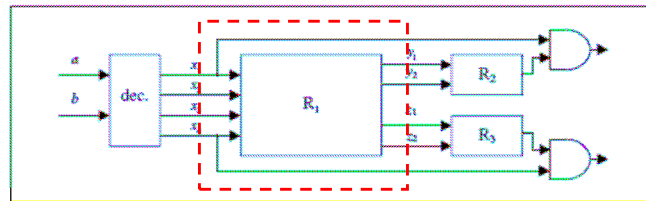
✓ E quindi non vengono mai presentate agli ingressi primari



■  $CDC_{in} = \underline{x_1}\underline{x_2}\underline{x_3}\underline{x_4} + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$

## Condizioni di Indifferenza Esterne

- Di osservabilità in uscita
  - Observability don't care ( $ODC_{out}$ )
  - Configurazioni di ingresso corrispondenti a situazioni in cui l'uscita non verrà osservata



- $ODC_{out} = [\underline{x}_1 \ \underline{x}_1 \ \underline{x}_4 \ \underline{x}_4]^T$

Maurizio Palesi

35

## Condizioni di Indifferenza Esterne

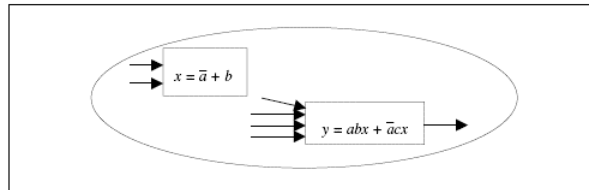
- Insieme complessivo delle condizioni d'indifferenza esterne
  - External don't care ( $DC_{ext}$ )
  - $DC_{ext} = CDC_{in} \cup ODC_{out}$
- $CDC_{in} = \underline{x}_1 \underline{x}_2 \underline{x}_3 \underline{x}_4 + \underline{x}_1 \underline{x}_2 + \underline{x}_1 \underline{x}_3 + \underline{x}_1 \underline{x}_4 + \underline{x}_2 \underline{x}_3 + \underline{x}_2 \underline{x}_4 + \underline{x}_3 \underline{x}_4$
- $ODC_{out} = [\underline{x}_1 \ \underline{x}_1 \ \underline{x}_4 \ \underline{x}_4]^T$

$$DC_{ext} = CDC_{in} + ODC_{out} = \begin{bmatrix} \overline{x}_1 + x_2 + x_3 + x_4 \\ \overline{x}_1 + x_2 + x_3 + x_4 \\ \overline{x}_4 + x_2 + x_3 + x_1 \\ \overline{x}_4 + x_2 + x_3 + x_1 \end{bmatrix}$$

Maurizio Palesi

36

## Insiemi Locali di Condizioni di Indifferenza



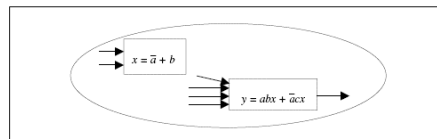
### ■ Mappa di Karnaugh per $y$

$cx$	$ab$ 00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	1	0
10	0	0	0	0

Maurizio Palesi

37

## Insiemi Locali di Condizioni di Indifferenza



- Non puo mai essere  $x \neq \bar{a} + b$ 
  - E' possibile definire le seguenti condizioni di indifferenza di controllabilità

$$CDC = x \oplus (\bar{a} + b) = \bar{x}\bar{a} + \bar{x}b + xab$$

$cx$	$ab$ 00	01	11	10
00	-	-	-	0
01	0	0	1	-
11	1	1	1	-
10	-	-	-	0

Maurizio Palesi

38

## Insieme di Soddisfacibilità

- L'uscita di una funzione non può mai essere diversa dalla valutazione della funzione stessa
- Per l'intera rete  $G(V,E)$  si può calcolare l'*insieme di soddisfacibilità*

$$SDC = \sum_{v_x \in V} (x \oplus f^x)$$

→  $x$  è l'uscita del generico nodo  $v_x$

→  $f^x$  è la funzione che genera  $x$