LABORATORIO DI ARCHITETTURA DEI CALCOLATORI

lezione n° 7

Prof. Rosario Cerbone

rosario.cerbone@libero.it

http://digilander.libero.it/rosario.cerbone

a.a. 2006-2007

Minimizzazione di circuiti combinatori multilivello

- In questa lezione vengono riassunti i concetti fondamentali della minimizzazione approssimata multi-livello. In particolare viene mostrato come utilizzare SIS per effettuare tale operazione.

Minimizzazione approssimata multi-livello

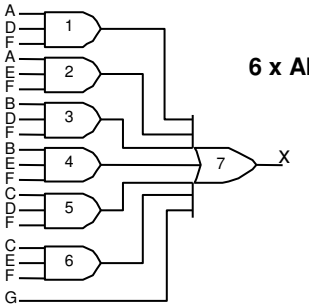
- La minimizzazione multi-livello consente al progettista di bilanciare area e ritardo di un circuito con un maggior grado di libertà rispetto alla minimizzazione a 2 livelli.
- Tuttavia, non esistono tecniche esatte efficienti che portino alla realizzazione di configurazioni ottime usando la minimizzazione multi-livello, pertanto si ricorre a tecniche euristiche che garantiscono buone soluzioni in tempi ragionevoli.

Minimizzazione approssimata multi-livello

Esempio 1

- ***Consideriamo la seguente funzione scritta come somme di prodotti minimizzata:***

- ***x* = A D F + A E F + B D F + B E F + C D F + C E F + G**

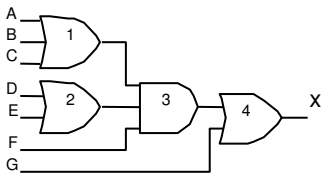


6 x AND a 3 ingressi + 1 x OR a 7 ingressi

Minimizzazione approssimata multi-livello

- ***Realizziamo una fattorizzazione della funzione (successive applicazioni della proprietà distributiva)***

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



1 x OR a 3 ingressi, 2 x OR a 2 ingressi, 1 x AND a 3 ingressi

Minimizzazione approssimata multi-livello

Esempio 2.

- ***Consideriamo la funzione seguente:***

$$F(x,y,z,k) = \sum 0,3,5,6,9,10,12,15$$

Applicando il metodo delle mappe di Karnaugh si arriva alla espressione:

F = !X!Y!Z!K + !X!YZK + !XY!ZK + !XYZ!K + X!Y!ZK + X!YZ!K + XY!Z!K + XYZK

Applicando la fattorizzazione:

F = !X!Y(!Z!K + ZK) + !XY(ZK + Z!K) + X!Y(!ZK + Z!K) + XY(!Z!K + ZK)

F = (!Z!K + ZK)(!X!Y + XY) + (ZK + Z!K)(!XY + X!Y)

Ricordando che: !(!A!B + AB) = (!AB + A!B) si può scrivere:

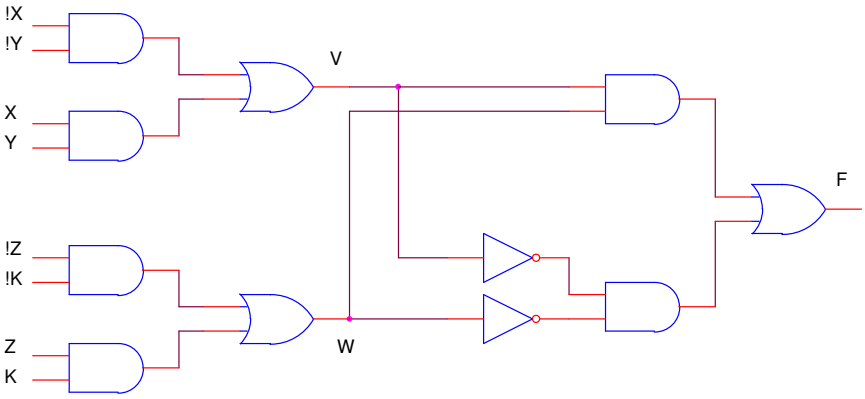
V = (!X!Y + XY); W = (!Z!K + ZK)

F = VW + (!V!W)

Minimizzazione approssimata multi-livello

Esempio 2.

V = (!X!Y + XY) ; W = (!Z!K + ZK); F = VW + (!V!W)



Minimizzazione approssimata multi-livello

- La tecnica di fattorizzazione, se applicata manualmente, implica una certa misura di intuito (o di fortuna) da parte del progettista, che deve saper scegliere i termini rispetto a cui fattorizzare e l'ordine di applicazione della fattorizzazione.
- Spesso si deve procedere ad una fase di espansione della funzione (ricorrendo al teorema di Shannon) prima di fattorizzare.
- Il procedimento risulta meglio applicato mediante strumenti di progettazione automatica (es. il SIS).
- Con il comando DECOMP si può ottenere una prima fattorizzazione della funzione, sulla quale applicare le tecniche di ottimizzazione indicate di seguito, dopo aver formalizzato alcuni concetti base.

Minimizzazione approssimata multi-livello

- Una rete logica può essere rappresentata da un *DAG (Directed Acyclic Graph)* nel quale i vertici corrispondono ai moduli (porte di I/O) 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

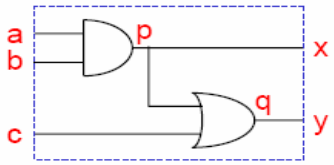
Minimizzazione approssimata multi-livello. Esempio di rete logica

Comportamento logico di I/O

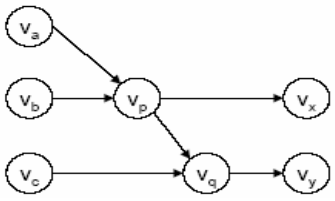
$x = ab$

$y = c + ab$

Rete logica mappata



Grafo della rete logica



Minimizzazione approssimata multi-livello.

- Una rete logica *non* gerarchica rappresentata dal grafo *Gn*(*V,E*) è costituita da:

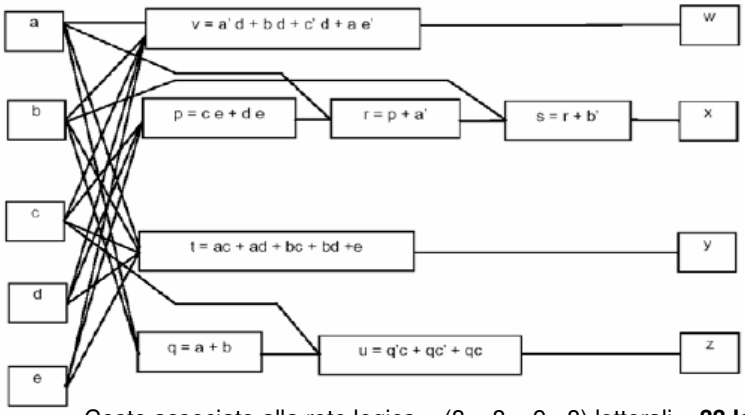
1. Un insieme di vertici *V* partizionato in *3* sotto-insiemi
    1. *VI* vertici relativi a ingressi primari e *ni* = |*VI*| numero degli ingressi primari
    2. *VO* vertici relativi a uscite primarie e *no* = |*Vo*| numero delle uscite primarie
    3. *VG* vertici interni e *ng* = |*VG*| numero dei vertici interni
    Ogni vertice è etichettato da una variabile
2. 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*)

Minimizzazione approssimata multi-livello. Esempio

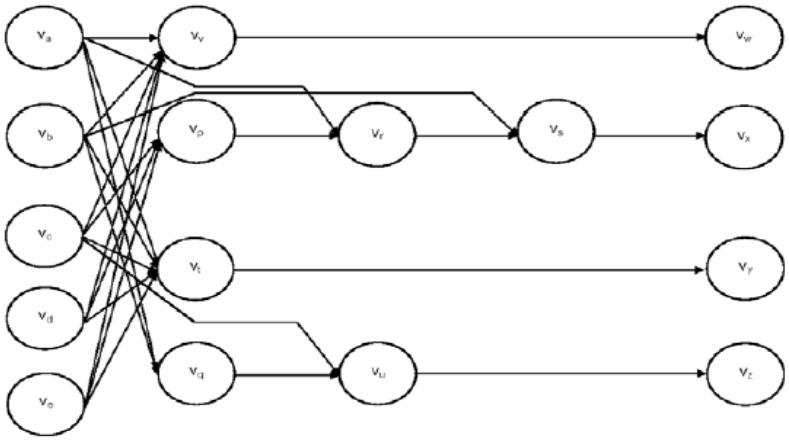
- Si consideri la rete logica con variabili di ingresso primarie {*a*,*b*,*c*,*d*,*e*},
- variabili di uscita primarie {*w*,*x*,*y*,*z*} descritta dalle seguenti equazioni
- *p = ce + d e*
- *q = a + b*
- *r = p + a'*
- *s = r + b'*
- *t = ac + ad + bc + bd + e*
- *u = q'c + qc' + qc*
- *v = a'd + bd + c'd + ae'*
- *w = v*
- *x = s*
- *y = t*
- *z = u*

Minimizzazione approssimata multi-livello. Esempio. Rappresentazione.



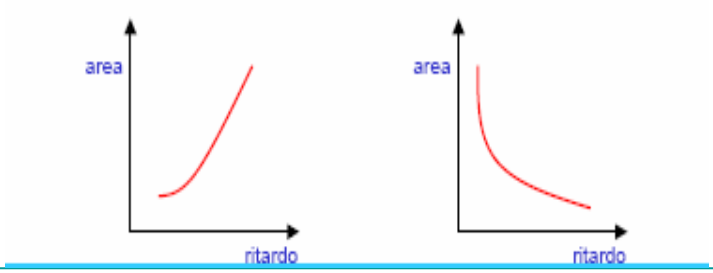
Costo associato alla rete logica = (8 + 8 + 9 +8) letterali = **33 letterali**

Minimizzazione approssimata multi-livello. Esempio. Grafo associato.



Minimizzazione approssimata multi-livello.

- Gli obiettivi dell'ottimizzazione logica a due livelli e multilivello 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
  - Compromesso tra area e ritardo



Minimizzazione approssimata multi-livello

- Riassumiamo di seguito i concetti principali relativi alle tecniche di sintesi applicate durante la minimizzazione multi-livello.
- Si consideri che il circuito viene rappresentato come un insieme di nodi interconnessi tra loro, e che ad ogni nodo corrisponde una funzione booleana a una sola uscita.
- Ogni nodo, pertanto, rappresenta una piccola porzione dell'intero design.
- Le tecniche descritte nei seguenti punti vengono solitamente ripetute secondo un determinato ordine fino al raggiungimento di una configurazione che soddisfi le aspettative del progettista.

Minimizzazione approssimata multi-livello. Trasformazioni.

- Sweep
- Eliminazione
- Scomposizione
- Estrazione
- Semplificazione
- Sostituzione

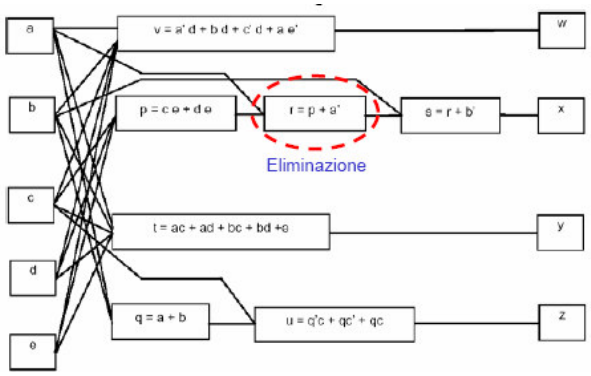
Minimizzazione approssimata multi-livello

1. ***Sweep:*** eliminazione dei nodi con un un'unica linea di ingresso e di nodi con valore costante.

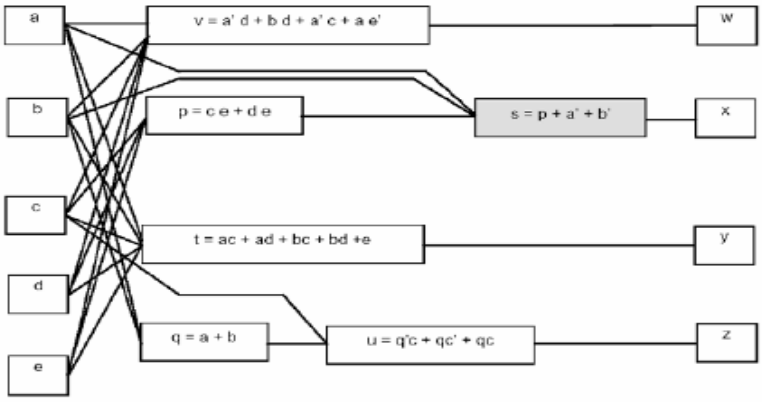
2. ***Eliminazione:*** eliminazione di un nodo interno alla rete.

Minimizzazione approssimata multi-livello. 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



Minimizzazione approssimata multi-livello. Eliminazione.

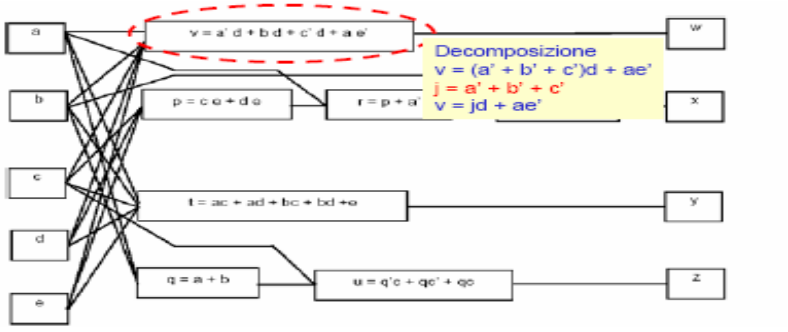


Minimizzazione approssimata multi-livello. Scomposizione

3. ***Scomposizione:*** sostituzione di un nodo interno con un'insieme di nodi la cui funzionalità sia equivalente a quella del nodo sostituito. L'operazione viene effettuate per diminuire la complessità di un nodo.

Minimizzazione approssimata multi-livello. Scomposizione

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

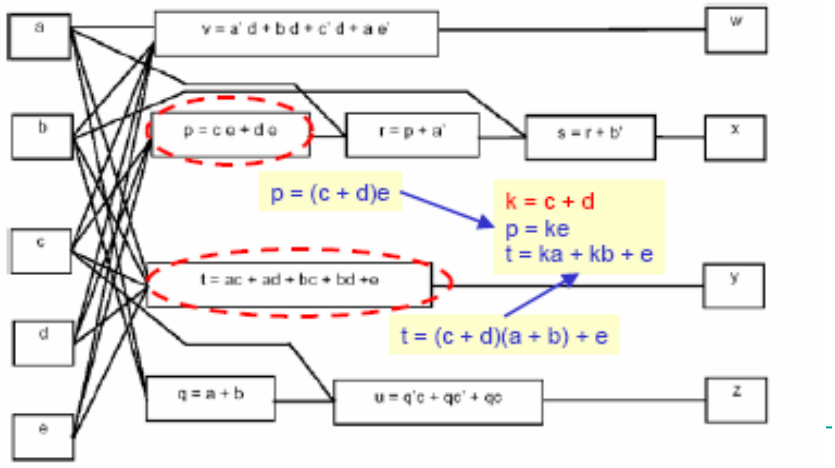


Minimizzazione approssimata multi-livello. Estrazione.

4. ***Estrazione:*** estrazione di una sottoespressione comune a più nodi che viene rappresentata con un nuovo nodo.

Minimizzazione approssimata multi-livello. Estrazione.

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

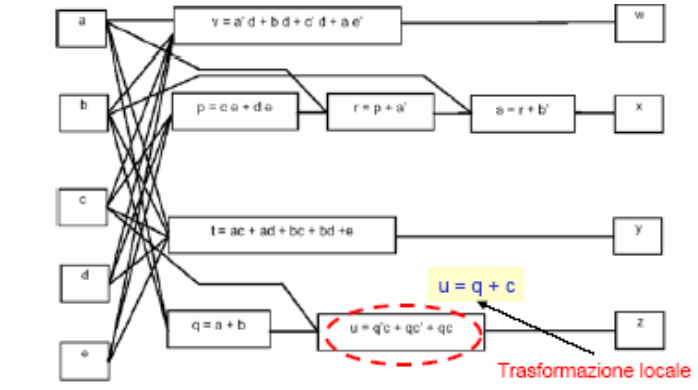


Minimizzazione approssimata multi-livello. Semplificazione

5. ***Semplificazione:*** riduzione della complessità di una funzione sfruttando le proprietà della sua rappresentazione, localmente o globalmente

Minimizzazione approssimata multi-livello. Semplificazione.

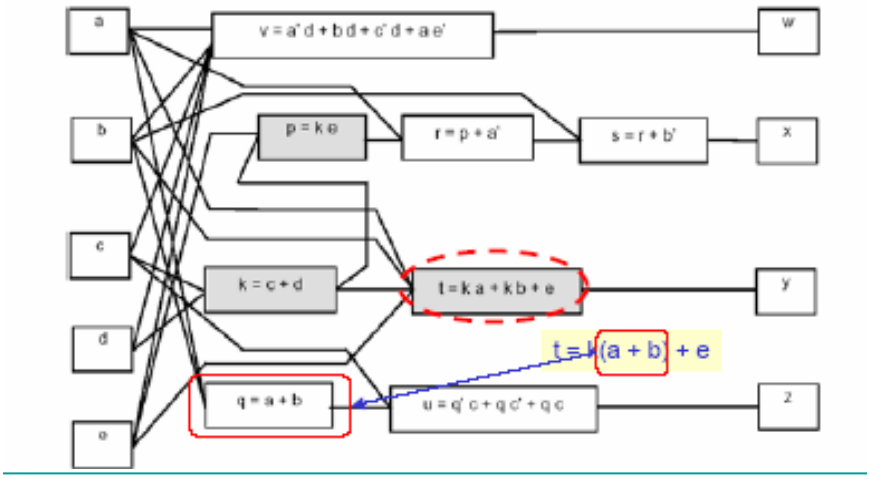
- Una funzione è ridotta in complessità sfruttando le proprietà della sua rappresentazione. Se la funzione è rappresentata nella forma a due livelli allora le tecniche di ottimizzazione a due livelli possono essere utilizzate. Se l'insieme di supporto non cambia allora la trasformazione si dice locale



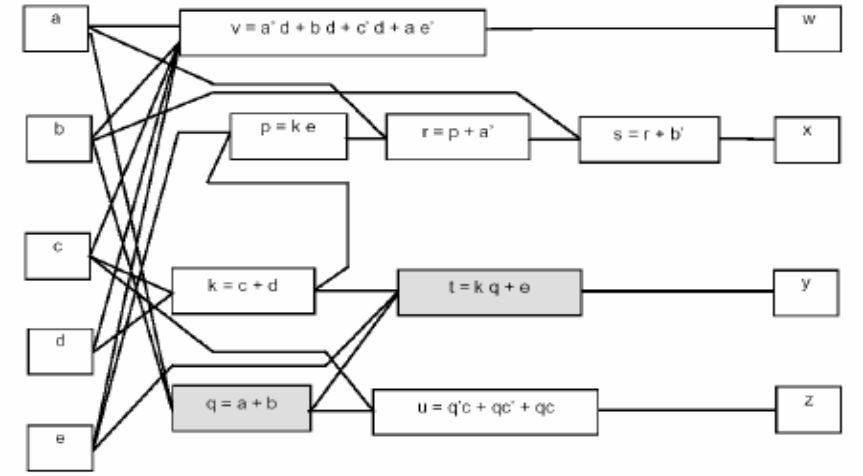
Minimizzazione approssimata multi-livello. Sostituzione.

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

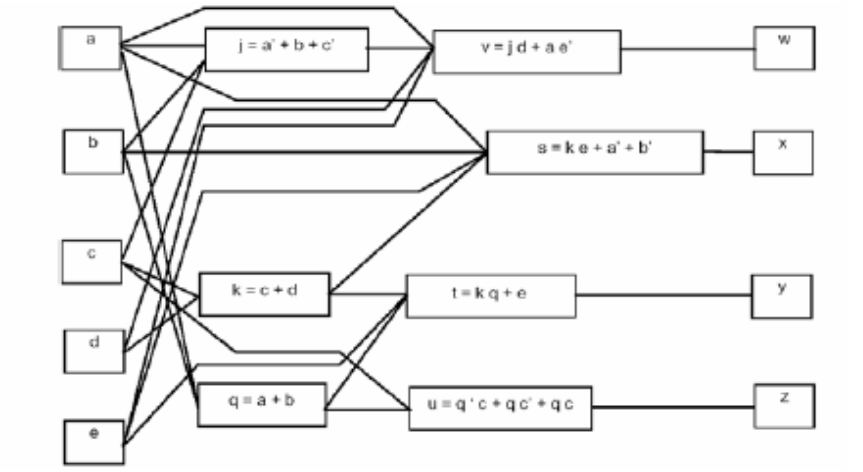
Minimizzazione approssimata multi-livello. Sostituzione.



Minimizzazione approssimata multi-livello. Sostituzione.



Minimizzazione approssimata multi-livello. Risultato delle trasformazioni.



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

Minimizzazione approssimata multi-livello. Risultato delle trasformazioni.

- $k = c + d$
- $q = a + b$
- $s = ke + a' + b'$
- $t = kq + e$
- $u = q'c + qc' + qc$
- $v = jd + ae'$
- $w = v$
- $x = s$
- $y = t$
- $z = u$
- Rispetto alla rete logica di riferimento il numero totale dei letterali è stato ridotto da 33 a 24

Mapping tecnologico

- Una volta completata l'ottimizzazione è possibile mappare il circuito in una rappresentazione che faccia uso esclusivamente di un insieme di porte logiche appartenenti ad una determinata libreria tecnologica con caratteristiche di area e ritardo diverse per ogni libreria.

Mapping tecnologico

- E' possibile mappare la rappresentazione logica del circuito utilizzando celle standard che rappresentano determinate funzioni decise a priori al momento della loro realizzazione sul silicio, oppure componenti customizzabili (es. FPGA), ovvero componenti organizzati in una topologia la cui funzionalità dipende dalla struttura di interconnessione delle porte che la compongono.
- Tale struttura può essere riprogrammata via software dopo la realizzazione del circuito.

Minimizzazione di circuiti combinatori multi-livello tramite SIS

- Una volta descritto il circuito desiderato nel formato blif, è possibile utilizzare i seguenti comandi per effettuare la minimizzazione.

Minimizzazione di circuiti combinatori multi-livello tramite SIS

1. sweep — Esegue l'operazione di sweep
2. eliminate n — Esegue l'operazione di eliminazione rimuovendo i nodi tali che la loro rimozione non aumenti il numero di letterali di una quantità superiore a n. Per eliminare i nodi che sono utilizzati una sola volta utilizzare il valore -1.

Minimizzazione di circuiti combinatori multi-livello tramite SIS

1. resub lista — Esegue l'operazione di scomposizione dei nodi indicati nella lista. Se la lista non viene specificata, la sostituzione viene eseguita per tutti i nodi della rete.
2. fx — Esegue l'operazione di estrazione.
3. full_simplify — Esegue l'operazione di semplificazione su ogni nodo della rete.

Minimizzazione di circuiti combinatori multi-livello tramite SIS

1. source script Carica lo script ed esegue tutti i comandi contenuti al suo interno. Lo script che fornisce generalmente i risultati migliori è script.rugged
2. set autoexec comando Stampa automaticamente il risultato del comando specificato dopo l'esecuzione di un qualunque altro comando.

Esercizi
lezione 7

- **Esercizio 1:** Eseguire la minimizzazione multi-livello usando lo script script.rugged su tutti i circuiti realizzati durante la lezione numero 5 e 6 (sintesi esatta a 2 livelli). Quale dei dispositivi viene maggiormente ottimizzato?

Esercizi

lezione 7

- **Esercizio 2:** Descrivere nel formato blif il circuito rappresentato dalla funzione
- booleana (x,v,w,z) = f(a,b,c,d,e) descritto dai seguenti nodi:
- *f = !abe + cd + a!cd*
- *g = ed + acb + a!cd*
- *h = ae+ cd +ab*
- *i = g + !h*
- *l = ag + bc!g + ae*
- *m = f + i + cb*
- *n = l + !ia*
- *o = m + a + !be*
- *x = f*
- *v = o*
- *z = l*
- w = n
- Eseguire la minimizzazione multi-livello usando lo script script.rugged.