



Progettazione Digitale:

Modellazione dei ritardi e mapping
tecnologico

Prof. Antonino Mazzeo

A cura di:

Ph.D., Ing. **Alessandra De Benedictis**, alessandra.debenedictis@unina.it

Testi di riferimento:

Franco Fummi, Mariagiovanna Sami, Cristina Silvano - Progettazione digitale - McGraw-Hill
G.De Micheli - Synthesis and optimization of digital circuits - McGraw-Hill – *Par. 8.6.1*

*CDL Magistrale Ing. Informatica - Prof. Antonino Mazzeo
Architettura dei Sistemi di Elaborazione*

Delay modeling

- Una rete logica può essere rappresentata mediante un grafo orientato, in cui ogni vertice è associato a una porta o, più in generale, ad una sottorete.
- A ciascun vertice può essere associato un peso pari al ritardo proprio della porta logica o sottorete che rappresenta.
- Il modello più semplice prevede che si assegni un ritardo unitario per ogni livello logico, ma si possono considerare modelli più complessi in cui si tiene conto della famiglia tecnologica di appartenenza.

Propagazione del ritardo

□ Per la valutazione del ritardo in una rete si considera in generale il *worst case*, in modo tale da tener conto di eventuali errori nella stima della propagazione dei ritardi, dovuti a fattori esterni (temperatura, alimentazione ecc.) o a parametri di fabbricazione.

□ A ciascun vertice viene inoltre assegnata una stima del tempo necessario per il set dei segnali in uscita, detto *data-ready time*.

I data-ready time degli input primari denotano quando sono stabili, sono il punto di partenza per il calcolo del ritardo nel circuito e sono spesso nulli;

Calcolo del ritardo

□ Il calcolo del *data-ready time* può essere fatto in vari modi; considerando esclusivamente le dipendenze sul grafo e trascurando la possibilità che alcuni percorsi potrebbero non verificarsi mai in relazione alle specifiche funzioni booleane locali, possiamo assumere che il data-ready time in ogni vertice interno e di uscita della rete sia la somma del ritardo di propagazione nel vertice più il data-ready time del più lento degli ingressi.

$$t_i = d_i + \max_j t_j \quad \forall j \text{ entrante in } i$$

Con d_i = ritardo di propagazione nel vertice

t_j = data ready time per il nodo j

Percorso critico

□ Il massimo data-ready time si ha presso un vertice di output, e viene detto ***ritardo critico topologico della rete.***

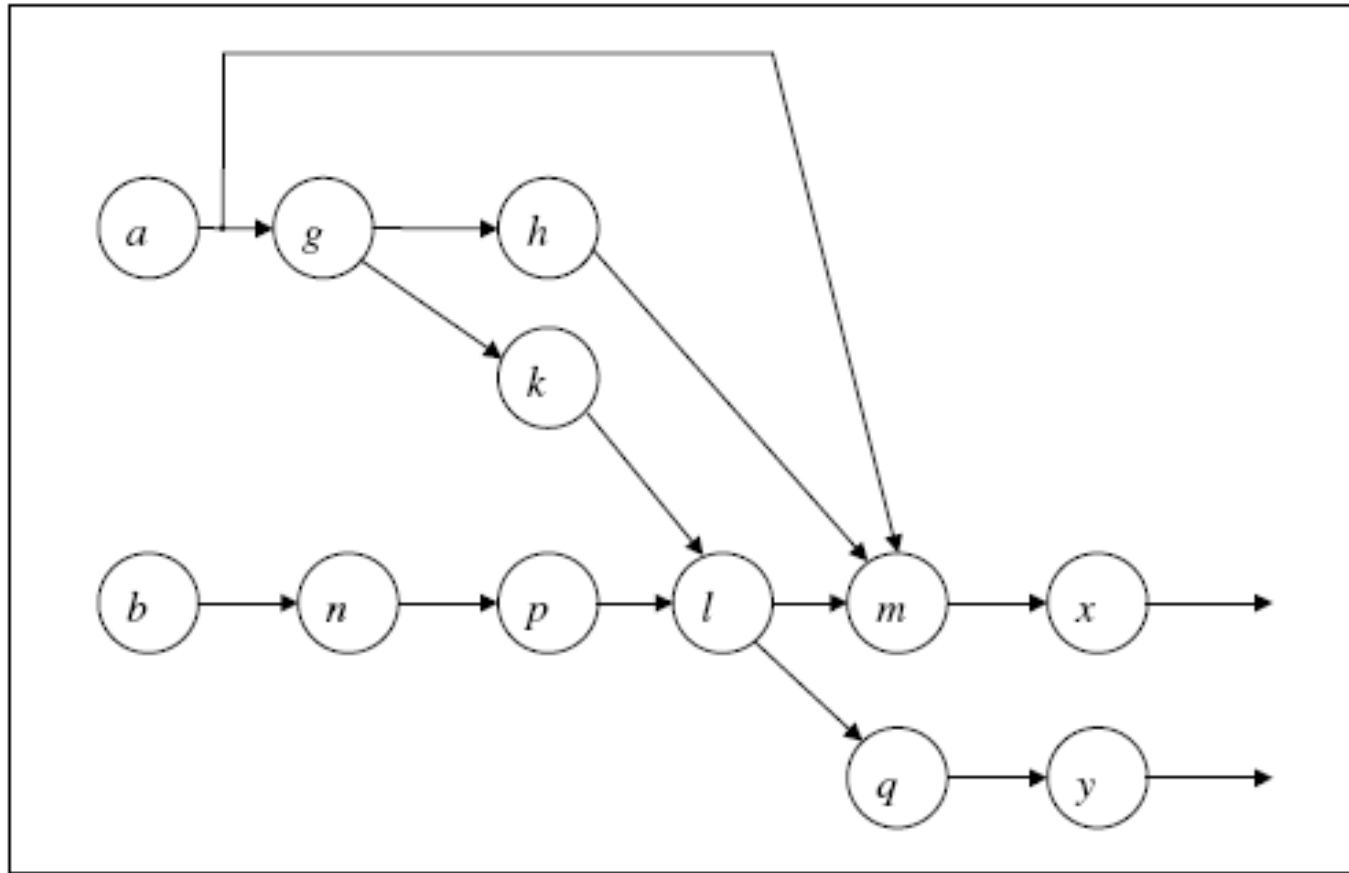
□ Esso corrisponde al peso del percorso più lungo nella rete, dove i pesi sono i ritardi di propagazione associati ai vertici. Tale percorso è detto **percorso critico.**

Si noti che tale percorso potrebbe non essere unico.

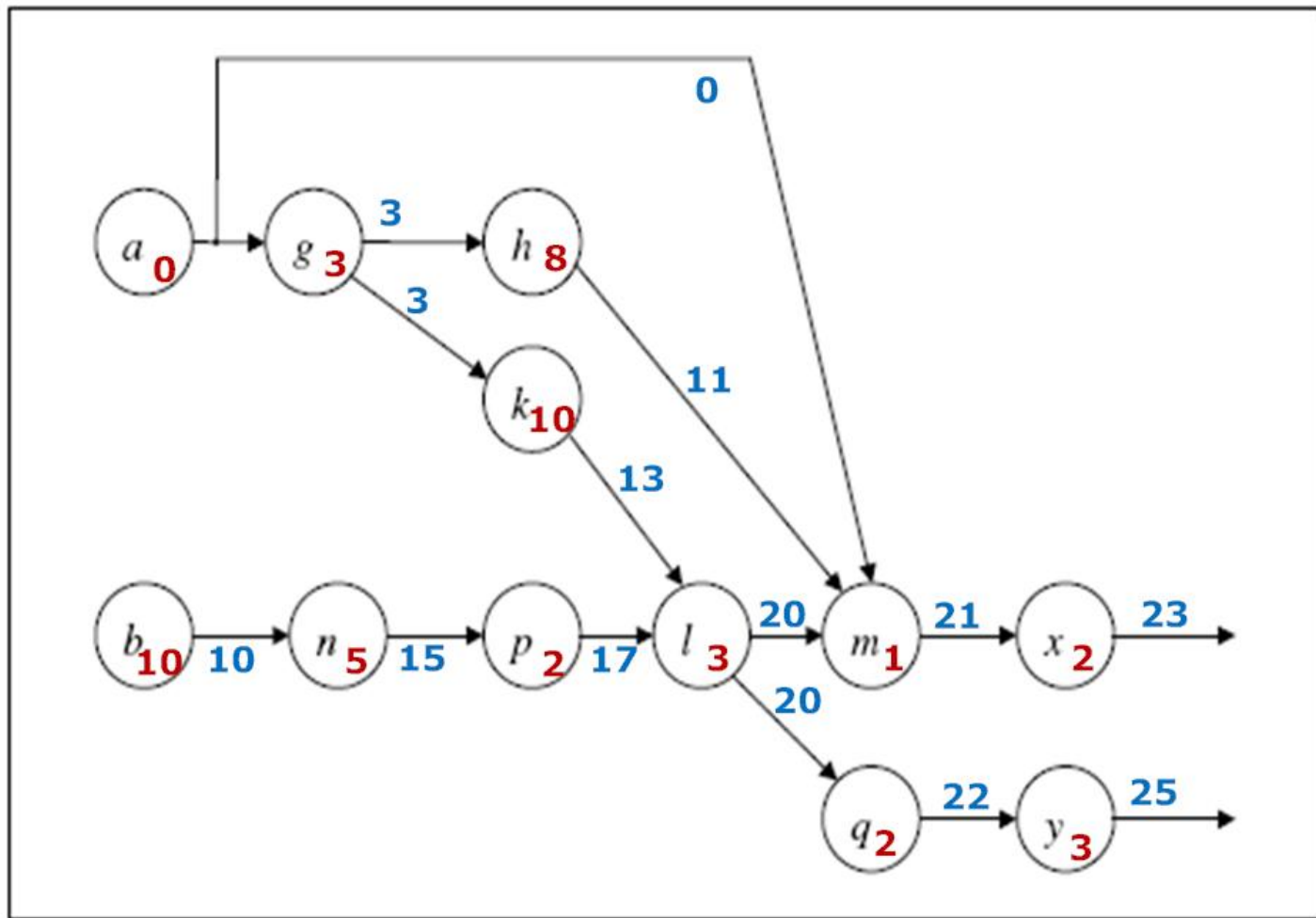
Slack

- ❑ A ciascun vertice si associa usualmente la differenza tra il data-ready time richiesto e quello reale, detta **slack**.
- ❑ Uno slack positivo indica che la rete rispetta i vincoli di tempificazione, mentre uno slack negativo comporta che vengano avviate procedure di ottimizzazione per ristrutturare la rete in modo che soddisfi i vincoli posti.

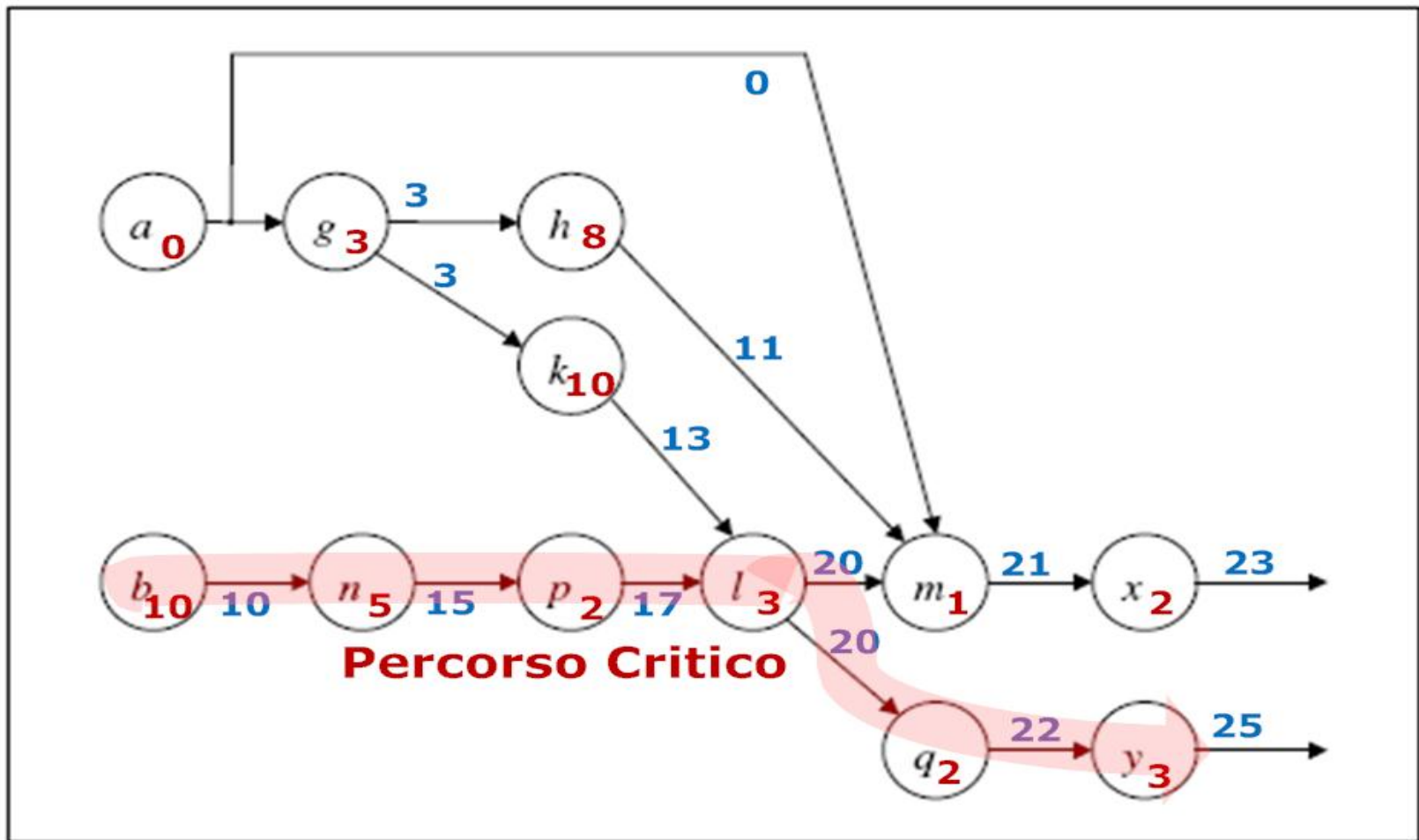
Esempio di rete per l'analisi dei ritardi



Esempio di rete per l'analisi dei ritardi



Esempio di rete per l'analisi dei ritardi



Prestazioni di una rete

- ❑ L'obiettivo della sintesi logica è l'ottimizzazione delle cifre di merito **area** e **prestazioni**
- ❑ Le **prestazioni** di una rete logica sono valutate di norma come *il ritardo di propagazione lungo il percorso critico*
 - *Reti combinatorie a due livelli: si riducono contemporaneamente area e ritardo*
 - *Reti combinatorie a più livelli: area e ritardo non procedono nella stessa direzione*



Osservazioni

- ❑ Il processo di ottimizzazione di reti logiche viene effettuato:
 - applicando delle euristiche e trascurando i vincoli (semplici modelli per area e prestazioni)
 - prendendo successivamente in considerazione i vincoli tecnologici (library binding)

Esempio 1

$z \backslash y$	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

$$f = x'y'z'v' + xyz'v' + x'yz'v + xy'z'v + x'y'zv + xyzv + x'yzv' + xy'zv'$$

9 porte
2 livelli

Si supponga di avere a disposizione solo porte a 2 ingressi; la funzione diventa:

$$f = \{ [((x'y)'z)'v' + ((xy)z')v'] + [((x'y)z')v + ((xy')z')v] \} + \{ [((x'y')z)v + ((xy)z)v] + [((x'y)z)v' + ((xy')z)v'] \}$$

31 porte
5 livelli

Esempio 1

$$f = x'y'z'v' + xyz'v' + x'yz'v + xy'z'v + x'y'zv + xyzv + x'yzv' + xy'zv'$$

Applicando la fattorizzazione:

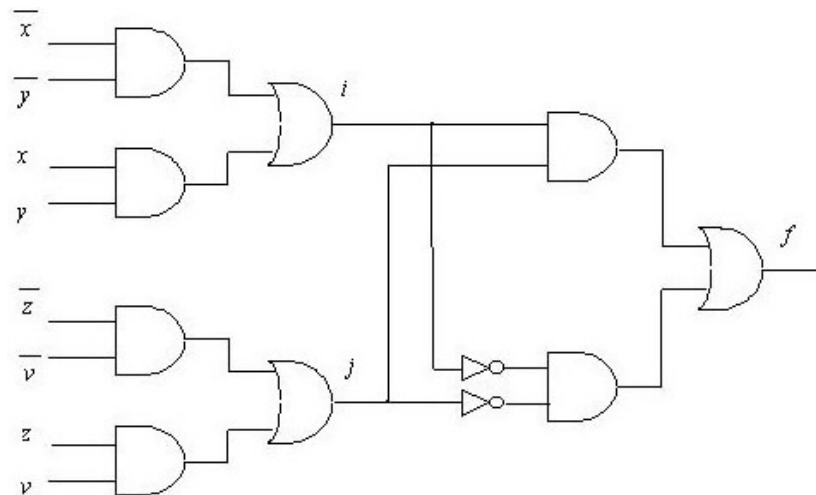
$$\begin{aligned} f &= xy(z'v' + zv) + x'y(z'v + zv') + xy'(z'v + zv') + x'y'(z'v' + zv) = \\ &= (\mathbf{xy + x'y'}) (\mathbf{z'v' + zv}) + (x'y + xy')(z'v + zv') \end{aligned}$$

Notando che $(a'b' + ab)' = (a'b + ab')$ si ha:

$$i = (xy + x'y')$$

$$j = (zv + z'v')$$

$$\Rightarrow f = ij + i'j'$$



**9 porte
4 livelli**

Esempio 2

Si supponga di disporre di porte con un massimo di 3 ingressi (ritardo uniforme τ)

$$f = l' + c'gh' + ab'k' + gk' + \textcolor{red}{a'b'c'd'e'} + ad'e'f' + e'g'i' + e'j'$$

- *La porta AND a 5 ingressi è realizzata come cascata di 2 AND a 3 ingressi (dà il ritardo maggiore)*
- *L'OR a 8 ingressi è realizzato con 3 OR in parallelo seguiti da 1 OR finale*

23 letterali
5 livelli

Esempio 2

Si proceda ora a fattorizzare k' fra il 3° e il 4° termine

$$\begin{aligned} f &= l' + c'gh' + ab'k' + gk' + a'b'c'd'e' + ad'e'f' + e'g'i' + e'j' = \\ &= l' + c'gh' + k'(ab' + g) + a'b'c'd'e' + ad'e'f' + e'g'i' + e'j'; \end{aligned}$$

22 letterali

5 livelli

Si applichi ancora la fattorizzazione – questa volta rispetto a e' , per i termini dal 4° all'ultimo

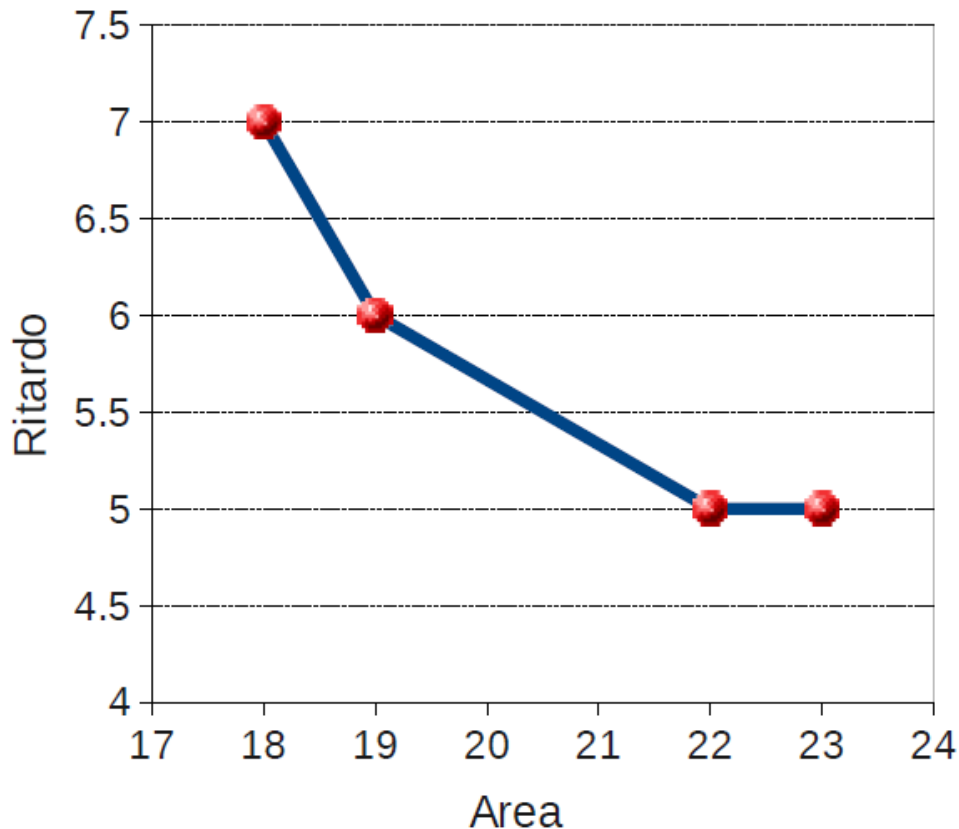
$$f = l' + c'gh' + k'(ab' + g) + e'(a'b'c'd' + ad'f' + g'i' + j')$$

19 letterali

6 livelli

Esempio 2

Infine, si applichi iterativamente la fattorizzazione dentro la seconda parentesi, questa volta rispetto a d

$$f = l' + c'gh' + k'(ab' + g) + e'(d'(a'b'c' + af') + g'i' + j')$$


18 letterali
7 livelli

Sintesi Vs Ottimizzazione

- Nella realizzazione di reti combinatorie a multilivello, più che ricercare un ottimo (che non è sempre definibile in maniera univoca), si cerca una soluzione accettabile in termini di area e ritardo
- Sarebbe più corretto parlare di **sintesi** invece che di ottimizzazione. La sintesi può prevedere:
 - Minimizzazione dell'area (con vincolo sul ritardo)
 - Minimizzazione del ritardo (con vincolo sull'area)

Mapping tecnologico

- Nella fase di sintesi si fa riferimento
 - a librerie di celle standard messe a punto e ottimizzate da un produttore di silicio, oppure
 - a componenti detti *semi-custom*, componenti in cui celle con struttura determinata sono organizzate secondo una topologia regolare, normalmente ad *array (matrice)*.
 - La specializzazione sul progetto avviene solo mediante la definizione della struttura di interconnessione.
- Tale fase è definita **library binding** o **mapping tecnologico**



Osservazioni

- ❑ Contemplare la libreria finale già nella fase di ottimizzazione del progetto può portare ad ulteriori riduzioni di costo , ma
- ❑ Tipicamente le fasi di progettazione sono separate

Perche?

Osservazioni

- ❑ Le prestazioni non sono l'unico parametro da valutare in un progetto di largo respiro!
- ❑ Dividere le due fasi permette di riadattare un dato progetto a librerie diverse e quindi anche all'evoluzione della tecnologia
 - garantendo una vita più lunga al progetto iniziale e
 - riducendo i costi totali di progettazione su una sequenza di “generazioni” tecnologiche

Librerie e associazione

- Le librerie disponibili sono progettate a vari livelli di astrazione;
 - librerie di livello logico includono l'insieme di primitive logiche volute.
 - nella fase di associazione si cerca l'implementazione migliore possibile del circuito dato, in modo da sfruttare le caratteristiche delle celle di libreria così da minimizzare area e ritardi
- La libreria fornisce, per ogni cella,
 - l'area e
 - il ritardo di propagazione

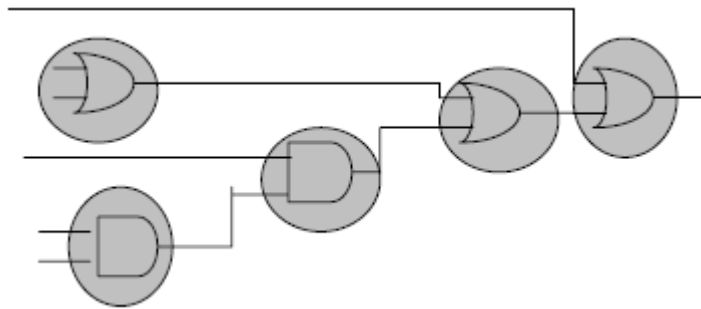
Tecniche di associazione

□ Il MappingTecnologico si può ricondurre ad un problema di copertura di un grafo

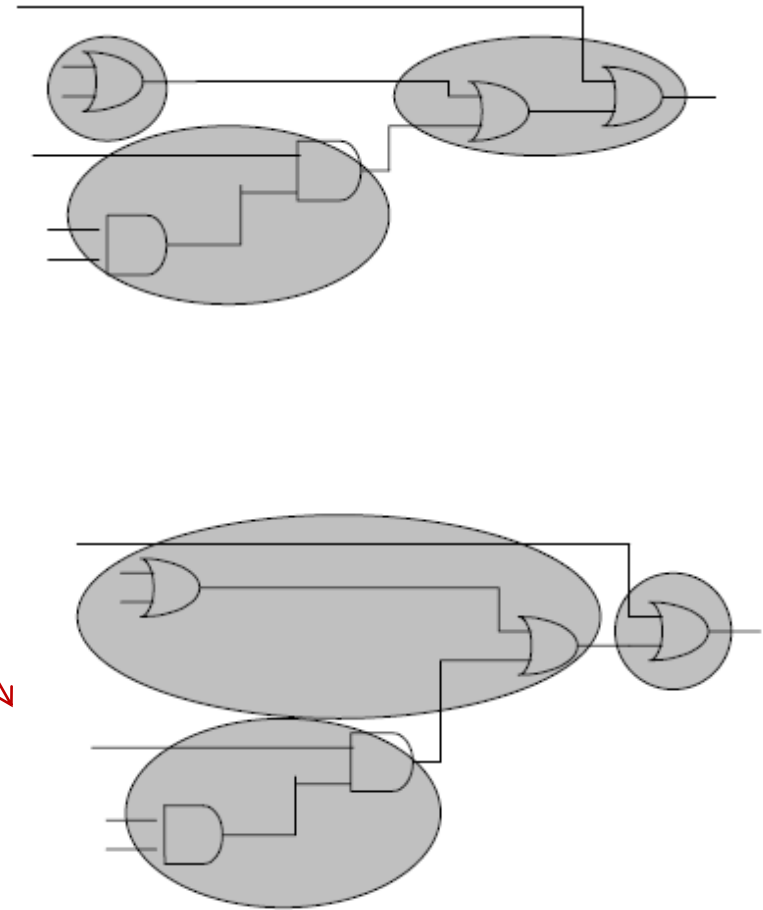
■ identificazione di un insieme di corrispondenze *che coprano tutti i vertici della rete logica da sintetizzare* (ogni nodo della rete logica deve trovare almeno un abbinamento con le celle)

■ soluzione più diffusa: sostituire sottoreti con celle della libreria: si identificano sottoreti che possono essere adeguatamente coperte con istanze di celle date ottimizzando area e/o ritardo

Tecniche di associazione



Supponiamo di avere
porte AND e OR a 3
ingressi



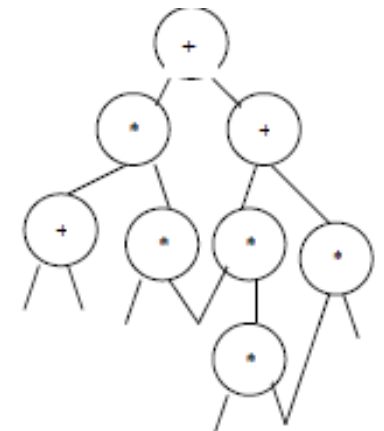
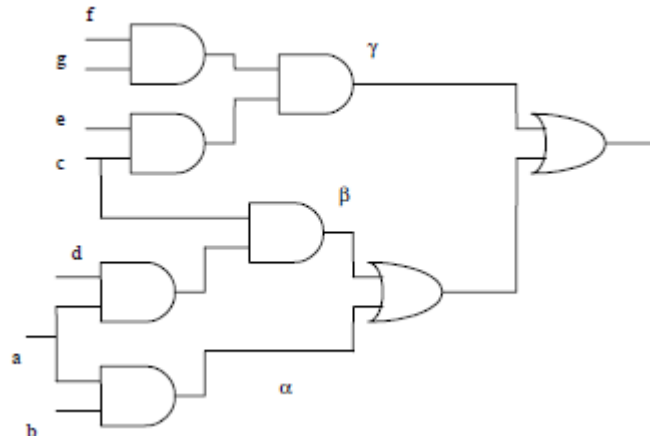
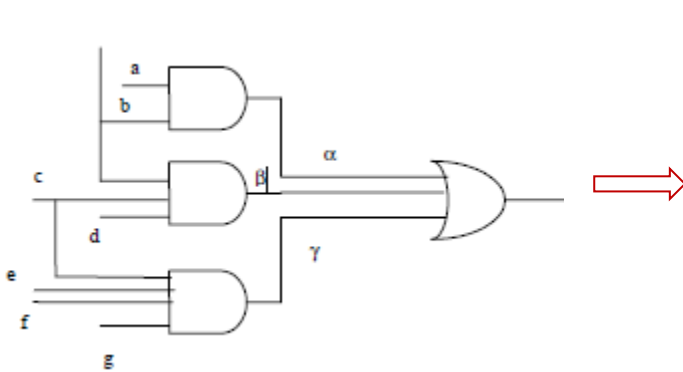
Tecniche di associazione

- La copertura esatta è un problema computazionalmente intrattabile; si ricorre ad algoritmi euristici, che cercano una soluzione sub-ottima.
- Sono state proposte due classi di algoritmi:
 - **booleani**: sia le celle di libreria, sia le parti della rete logica per cui si cercano abbinamento e copertura sono rappresentate da funzioni booleane, e
 - **strutturali**: usano grafi che rappresentano la scomposizione algebrica della rete booleana
 - ... differiscono nella fase di abbinamento

Approccio strutturale

□ In questo caso, l'abbinamento viene fatto ricercando nel grafo che rappresenta la rete logica sottografi isomorfi ad altri che rappresentino celle di libreria.

■ a tal fine, rete logica e celle devono essere tradotti in grafi confrontabili



■ Supponiamo di avere solo porte AND e OR a 2 ingressi

Approccio strutturale

- I grafi corrispondenti alle celle di libreria costituiscono dei *modelli*
- il problema dell'abbinamento diventa quello di ricercare nel grafo della rete logica dei sottografi isomorfi ai modelli, e di completare tale ricerca in modo da garantire la *copertura della rete logica* data, possibilmente in modo *ottimo (rispetto all'area e/o ai ritardi)*.