LABORATORIO DI ARCHITETTURA DEI CALCOLATORI
lezione n° 5

Prof. Rosario Cerbone

rosario.cerbone@libero.it

a.a. 2005-2006

Ottimizzazione di circuiti combinatori

- In questa lezione vengono riassunti i concetti fondamentali dell'ottimizzazione esatta a 2 livelli.
- Inoltre viene mostrato come utilizzare SIS per effettuare l'ottimizzazione di un dispositivo digitale combinatorio specificato in formato blif.

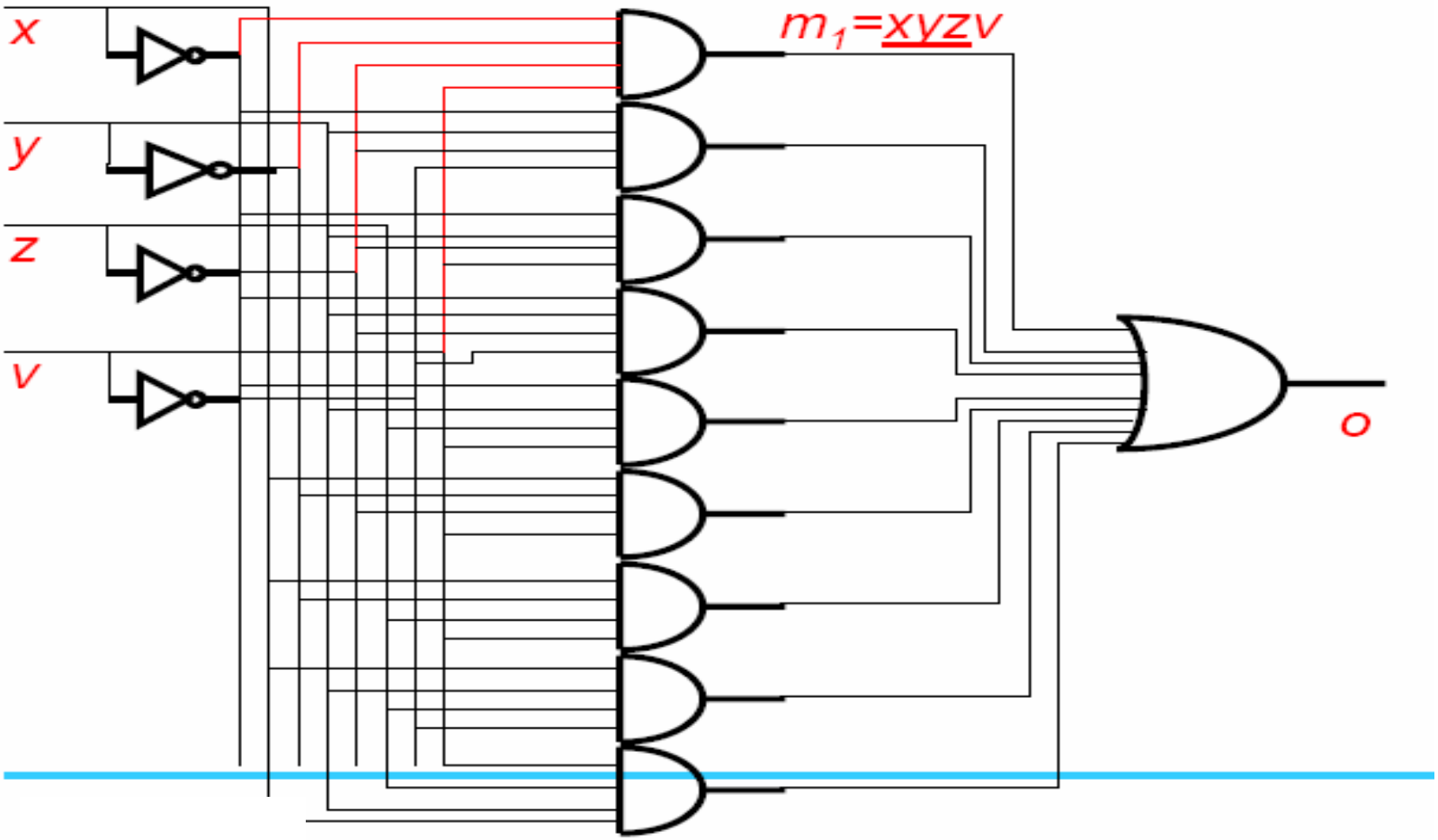
Ottimizzazione esatta a 2 livelli

- Tra le caratteristiche principali di un circuito digitale, un particolare importanza rivestono:
- l'***area*** (misurabile come il numero di porte logiche a 2 ingressi necessarie per la realizzazione del circuito)
- il ***ritardo di propagazione*** con cui i segnali applicati in ingresso forniscono il corrispondente risultato in uscita (misurabile come il massimo numero di porte logiche che un segnale applicato agli ingressi deve attraversare per raggiungere l'uscita).
- E' pertanto necessario studiare tecniche che permettano di ottimizzare l'area e il ritardo in modo da avere circuiti "piccoli" e "veloci".

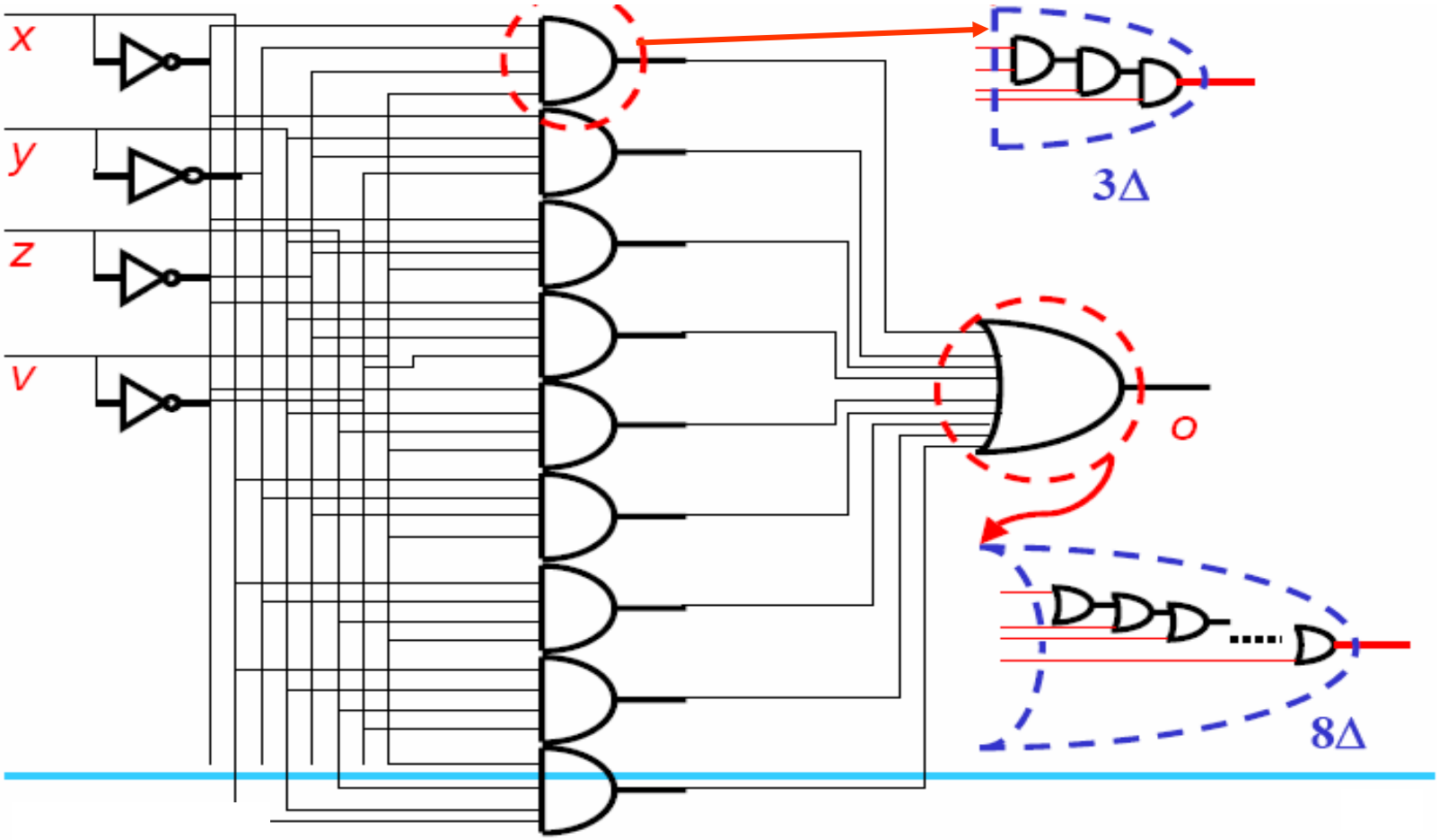
Ottimizzazione esatta a 2 livelli

- L'ottimizzazione di un circuito comporta
- normalmente un compromesso tra:
- Prestazioni (ritardo di propagazione)
- Area (o costo)
- Potenza dissipata
- Testabilità
- ...

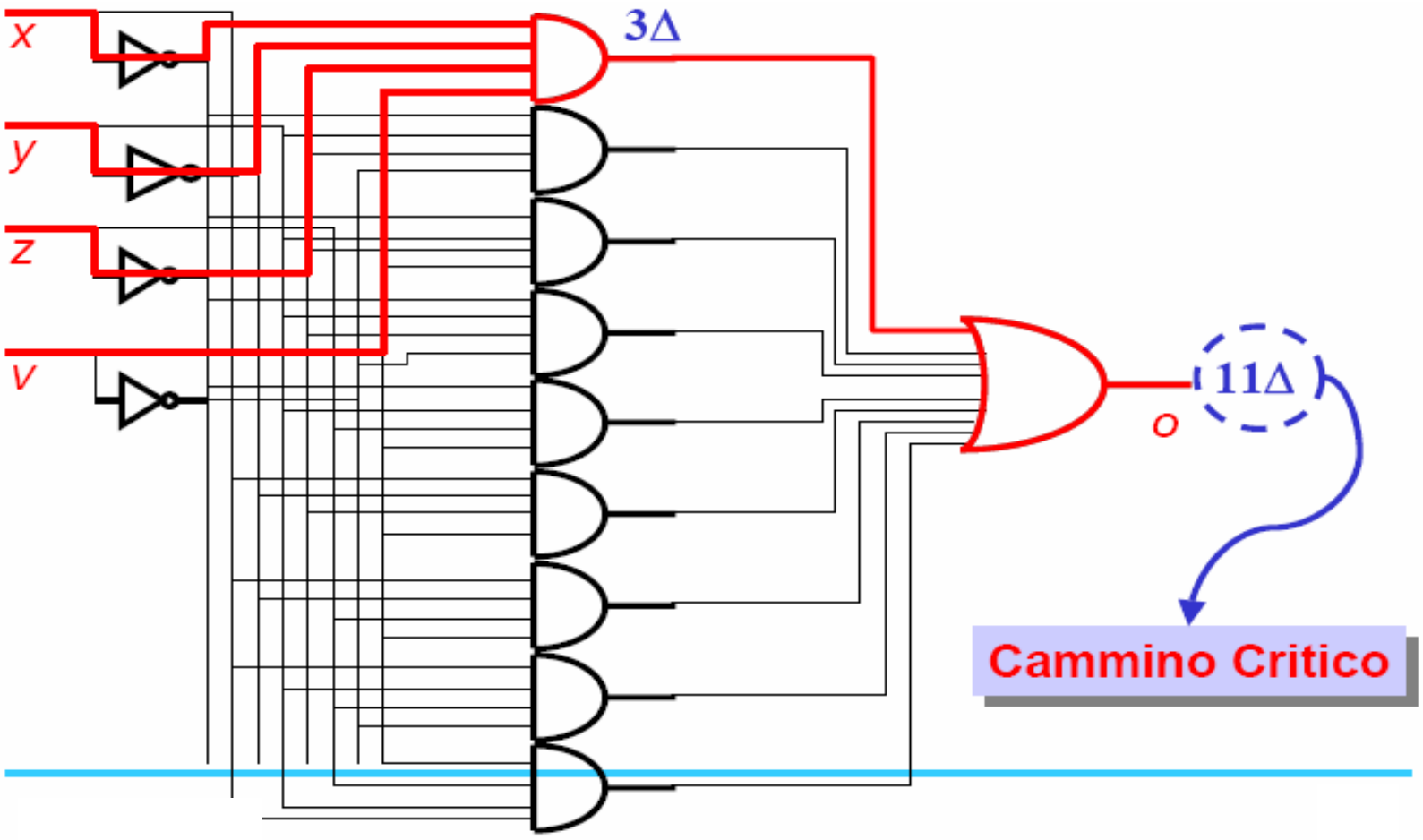
Ottimizzazione esatta a 2 livelli



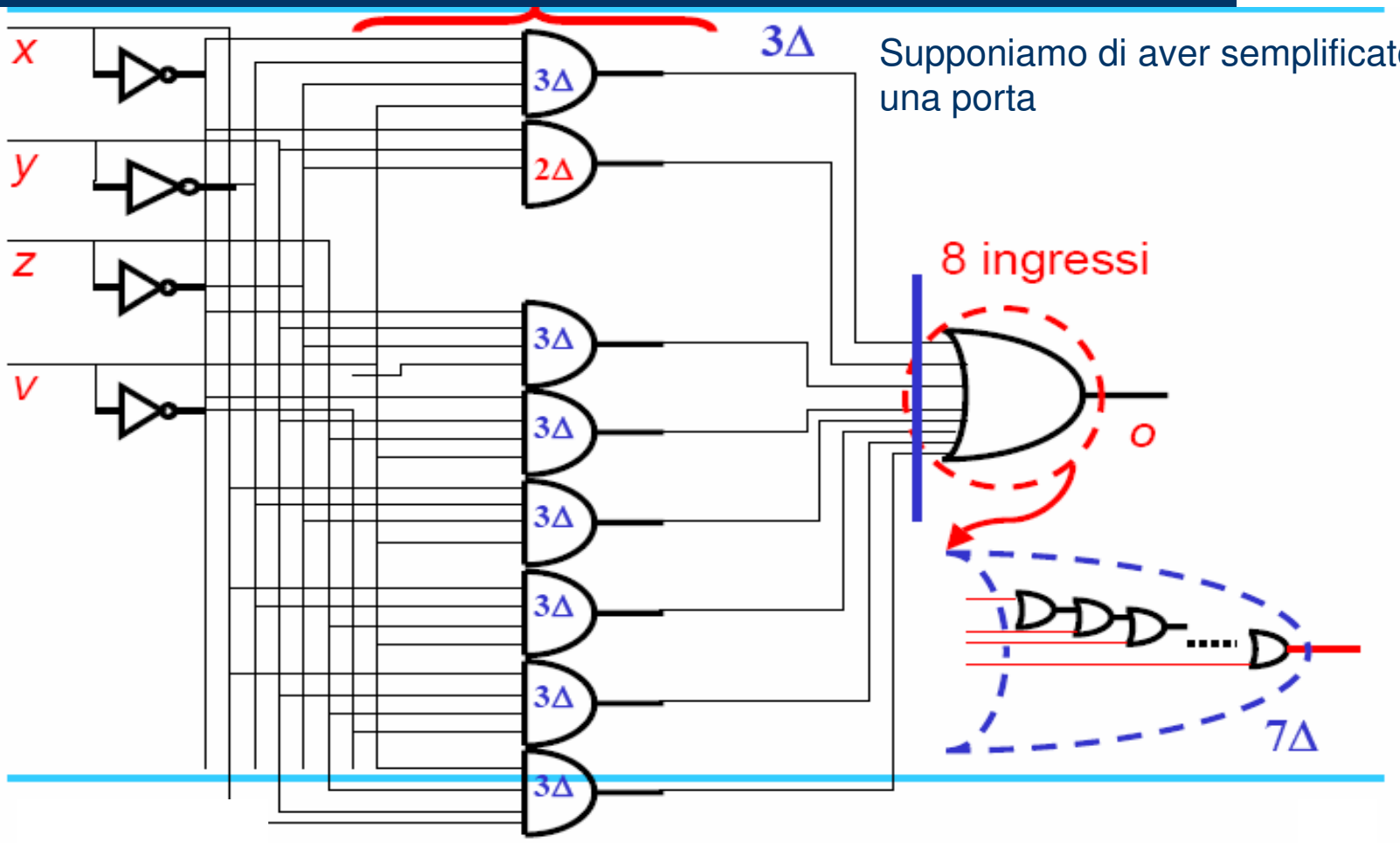
Ottimizzazione esatta a 2 livelli



Ottimizzazione esatta a 2 livelli



Ottimizzazione esatta a 2 livelli



Supponiamo di aver semplificato una porta

Ottimizzazione esatta a 2 livelli

- Riepilogando:
- In un circuito a due livelli (somma di prodotti) la riduzione del numero di prodotti riduce sia l'area che il ritardo.
- L'ottimizzazione dei circuiti a due livelli segue, quindi, il criterio molto semplice di identificare la *copertura minima*
- Alla copertura minima corrisponde il circuito a due livelli di *area minima* e *ritardo minimo*

Ottimizzazione esatta a 2 livelli

- Per comprendere appieno le tecniche di ottimizzazione è necessario fornire le seguenti definizioni:
- - Un ***implicante*** si dice ***primo*** se non esiste nessun altro implicante di dimensioni maggiori (ovvero formato da un minor numero di letterali) che lo contenga interamente.
- - Un ***implicante*** si dice ***essenziale*** se esiste almeno un mintermine coperto dall'implicante che non è coperto da nessun altro implicante della funzione

Ottimizzazione esatta a 2 livelli

- Esistono varie tecniche per ottimizzazione area e ritardo di propagazione.
- La minimizzazione di circuiti a 2 livelli avviene come segue:
- 1. Si identificano tutti gli implicanti primi essenziali.
- 2. Si identifica un insieme minimo di implicanti che coprano tutti i mintermini non coperti dagli implicanti primi essenziali.
- 3. La funzione di copertura ottima è data dalla somma degli implicanti di cui ai punti 1 e 2.

Ottimizzazione esatta a 2 livelli

- La seguente tabella evidenzia i principali risultati ottenuti in termini di algoritmi di minimizzazione a 2 livelli per circuiti a una o più uscite.

	Circuiti a 2 livelli
Circuiti a una uscita	Esiste un metodo esatto per trovare gli implicanti primi essenziali Esiste un metodo esatto o approssimato (dipende dal circuito) per ottenere la funzione di copertura ottima
Circuiti a più uscite	Esiste un metodo approssimato per trovare la copertura ottima che si basa sul metodo esatto di identificazione degli implicanti primi essenziali di ogni singola uscita

Metodo di Quine-McCluskey per funzioni completamente specificate

1. Estrarre dalla tabella delle verità i mintermini della funzione.

2. Ordinare i mintermini in base al numero di 1 (letterali positivi) che contengono.
   Raggruppare i mintermini con le stesso numero di 1.

3. Per ogni mintermine *m* il cui numero di 1 è pari a *i*, cercare tra gli eventuali mintermini il cui numero di letterali 1 è *i*+1, quelli a distanza di Hamming 1 da *m*.
   Per ogni mintermine a distanza di Hamming 1 da *m*, riportare in una nuova tabella una riga che "collassi" i letterali di *m* e di *n* riportando il simbolo di *don't care* (–) nella posizione corrispondente all'unico letterale in cui *m* e *n* differiscono.
   Ad esempio, dati i mintermini *m* = 0101 e *n* = 0111, nella nuova tabella verrà riportata la riga 01–0.

Metodo di Quine-McCluskey per funzioni completamente specificate

4. Considerare la tabella ottenuta al passo 3 e reiterare i passi 2 e 3 finché non sarà più possibile collassare mintermini. Durante i passi 2 e 3 marcare i mintermini (o le righe della tabella corrispondenti al collassamento di 2 mintermini) che non è stato possibile collassare.

5. Le righe non collassate delle varie tabelle create reiterando i passi 2 e 3 corrispondono agli implicanti primi della funzione.

6. Creare una tabella che riporta in colonna i mintermini e in riga gli implicanti primi. Inserire nell'elemento (*i, j*) il valore 1 se e solo se il mintermine *j* è coperto dall'implicante *i*.

Metodo di Quine-McCluskey per funzioni completamente specificate

7. Nella tabella del punto 6 marcare i mintermini coperti da un unico implicante primo. Tali implicanti sono implicanti primi essenziali e faranno sicuramente parte della copertura ottima.
8. Creare una nuova tabella a partire dalla tabella del punto 6 ottenuta eliminando le righe degli implicanti essenziali e le colonne corrispondenti ai mintermini coperti dagli implicanti corrispondenti.
9. Nella tabella creata al punto 8 eliminare le eventuali righe dominate (ovvero le righe in cui gli 1 sono un sottoinsieme degli 1 contenuti in un'altra riga).

Metodo di Quine-McCluskey per funzioni completamente specificate

10. Ripetere i passi 7,8 e 9 nella tabella ottenuta al punto 9 finché tutti i mintermini risultano coperti, oppure si ottiene una tabella in cui non esistono righe dominanti.
    Nel primo caso sono stati identificati tutti gli implicanti necessari per ottenere la copertura ottima. Nel secondo caso procedere come indicato al punto 11.
11. Nella tabella ottenuta al punto 10 cercare ed eliminare le righe dominanti (ovvero le righe in cui gli 1 sono una sovra-insieme degli 1 contenuti in un'altra riga) e ripetere i passi a partire dal 9. Qualora non esistano righe dominanti si deve procedere usando una tecnica di branch and bound, ovvero si sceglie un implicante (di solito quello che copre il maggior numero di mintermini), si considera come se fosse essenziale, e si riparte dal passo 8.

Metodo di Quine-McCluskey per funzioni non completamente specificate

- Nel caso di funzioni non completamente specificate, esistono alcune combinazioni delle variabili di input per cui non è specificato un corrispondente valore per le variabili di output. Le righe della tabella delle verità che corrispondono a tali combinazione costituiscono il ***don't care set*** della funzione. I valori per le variabili in output che corrispondono alle righe del don't care set possono assumere indifferentemente il valore 0 o il valore 1.
- A seconda di tale scelta è possibile ottenere circuiti più o meno ottimizzati.

Metodo di Quine-McCluskey per funzioni non completamente specificate

- Il metodo pratico per ottimizzare funzioni non completamente specificate è esattamente quello descritto precedentemente per le funzioni completamente specificate, considerando le righe della tabella delle verità corrispondenti al don't care set come se fossero mintermini (ovvero appartenenti all' on-set).
- E' necessario tuttavia ricordare che tali "finti mintermini" non devono essere necessariamente coperti dagli implicanti che costituiranno la copertura minima. Pertanto le righe corrispondenti al don't care set devono essere marcate come "collassabili" a priori durante i passi 2 e 3 dell'algoritmo.

Minimizzazione di circuiti combinatori a 2 livelli tramite SIS

- Per minimizzare una rappresentazione blif con SIS è possibile utilizzare il comando **full_simplify** dopo aver caricato il modello da ottimizzare.
- E' possibile osservare l'effetto della minimizzazione usando il comando **write_eqn** prima e dopo aver utilizzato il comando full_simplify. Il comando write_eqn visualizza l'espressione booleana corrispondente al modello rappresentato in blif.
- Inoltre il comando print_stats visualizza le seguenti informazioni sul circuito: numero variabili in input (PI), numero variabili in output (PO), numero di porte logiche (nodes), numero di elementi di memoria (LATCHES), numero di letterali (SOPS).

Minimizzazione di circuiti combinatori a 2 livelli tramite SIS

- Per descrivere il ***dont'care set*** di una funzione è necessario usare la keyword **.exdc**.
- Ad esempio, data la seguente tabella di verità parzialmente specificata

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	-
0	1	1	0
1	0	0	1
1	0	1	-
1	1	0	-
1	1	1	1

Minimizzazione di circuiti combinatori a 2 livelli tramite SIS

- il formato blif che la rappresenta è il seguente:
- .model esempio
- .inputs a b c
- .outputs z
- .names a b c z
- 001 1
- 100 1
- 111 1
- .exdc
- .names a b c z
- 010 1
- 101 1
- 110 1
- .end

Minimizzazione di circuiti combinatori a 2 livelli tramite SIS

- Volendo analizzare ed ottimizzare il circuito precedente è possibile eseguire i seguenti comandi dal prompt dei comandi (si supponga che i il modello blif sia stato salvato in un file di nome esempio.blif):

> sis	Avvia il programma SIS
> read_blif esempio.blif	Carica il modello in memoria
> write_eqn	Scrive l'espressione corrispondente al modello
> print_stats	Stampa info sul modello
> full_simplify	Ottimizza il modello
> write_eqn	scrive l'espressione del modello dopo ottimizzazione
> print_stats	Stampa info sul modello dopo ottimizzazione

Esercizio 1

- Scrivere la tabella delle verità e rappresentare nel formato blif il circuito digitale corrispondente alla seguente espressione booleana:

  f(x,y,z,v)=!xyz+!x!y+y!zv+xv!y+yz.

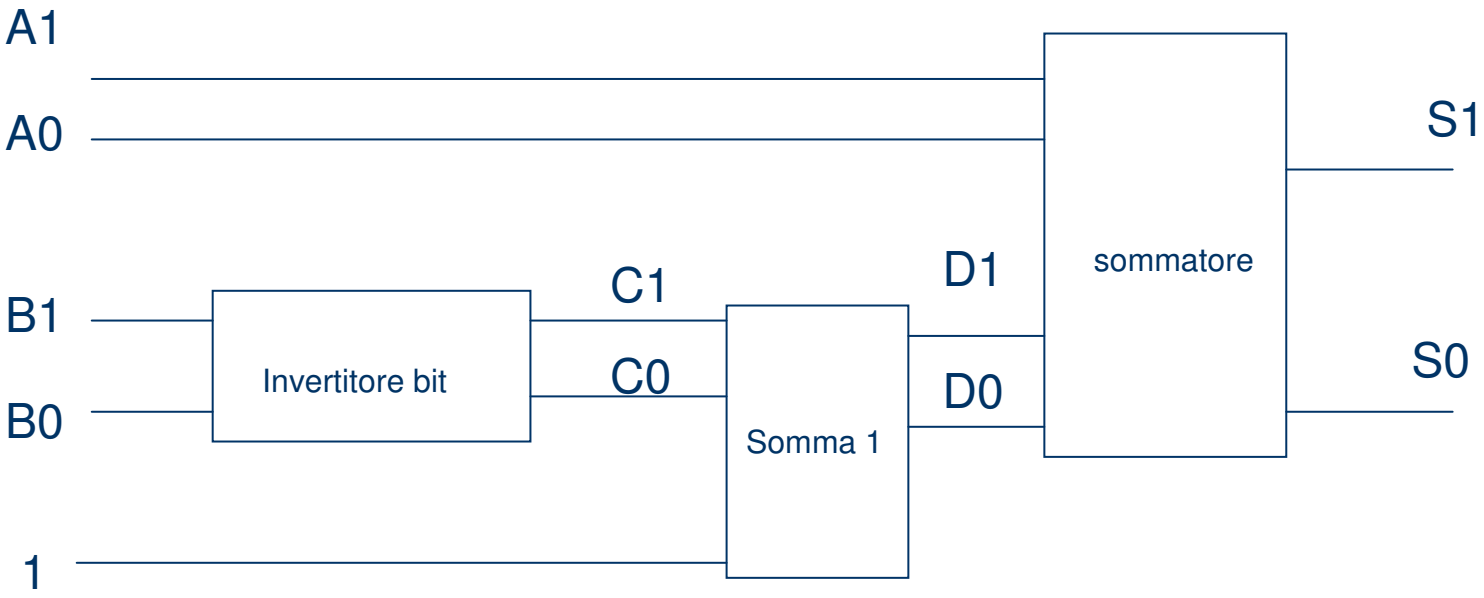
- Eseguire l'ottimizzazione con SIS usando il comando full_symplify.
- Fornire il grado di ottimizzazione confrontando l'area (numero porte logiche) e il ritardo (lunghezza cammino critico) del circuito prima e dopo l'ottimizzazione.
- Visualizzare l'espressione booleana corrispondente al circuito ottimizzato con il comando write_eqn e confrontarla con quella che si ottiene manualmente usando il metodo di Quine-McCluskey.

Esercizio 2

- Descrivere in formato blif il circuito digitale che esegue la sottrazione di 2 numeri binari a 2 bit rappresentati in complemento a 2. Eseguire l'ottimizzazione con SIS. Visualizzare l'espressione booleana corrispondente al circuito prima e dopo l'ottimizzazione e fornire il grado di ottimizzazione.

Esercizio 2

- La schema del circuito è il seguente:



Esercizio 3

- Descrivere nel formato blif un circuito digitale che riceve in input una sequenza di 5 bit che rappresenta la codifica binaria di una lettera dell'alfabeto italiano considerando di associare in ordine crescente i numeri dallo 0 al 21 alle lettere dalla A alla Z (A=0, B=1, …, Z=21).
- Il circuito ha un solo bit in uscita che vale 1 se e solo se la sequenza di input corrisponde ad una vocale.
- La tabella delle verità risulterà parzialmente specificata dal momento che con 5 cifre binarie è possibile rappresentare i numeri dallo 0 al 31.

Esercizio 3

- Descrivere nel formato blif un circuito digitale che riceve in input una sequenza di 5 bit che rappresenta la codifica binaria di una lettera dell'alfabeto italiano considerando di associare in ordine crescente i numeri dallo 0 al 21 alle lettere dalla A alla Z (A=0, B=1, …, Z=21).
- Il circuito ha un solo bit in uscita che vale 1 se e solo se la sequenza di input corrisponde ad una vocale.
- La tabella delle verità risulterà parzialmente specificata dal momento che con 5 cifre binarie è possibile rappresentare i numeri dallo 0 al 31.

Esercizio 3

- Ottimizzare il circuito con SIS nel caso in cui alle righe della tabella delle verità non specificate venga associato:
- - il valore 0,
- - il valore 1,
- - il valore don’t care.
- Quale dei tre circuiti è maggiormente ottimizzato?

Esercizio 4

- Descrivere nel formato blif un circuito che calcola la parte intera della radice quadrata delle 10 cifre decimali rappresentate da numeri binari a 4 bit.
- L'output sarà una sequenza composta da 2 bit.
- La tabella delle verità del circuito risulterà parzialmente specificata dal momento che con 4 bit è possibile rappresentare i numeri dallo 0 al 15.

Esercizio 4

- Ottimizzare il circuito con SIS nel caso in cui alle righe della tabella delle verità non specificate venga associato:
- - il valore 0,
- - il valore 1,
- - il valore don’t care.
- Quale dei tre circuiti è maggiormente ottimizzato?

Metodo di Quine-McCluskey 1

Esempio: consideriamo la seguente funzione logica. Essa può anche essere scritta come

$E(a,b,c,d)=\Sigma_4\ (5,6,7,8,9,12,13,14)$

A	B	C	D	E	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	--> 5
0	1	1	0	1	--> 6
0	1	1	1	1	--> 7
1	0	0	0	1	--> 8
1	0	0	1	1	--> 9
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	1	--> 12
1	1	0	1	1	--> 13
1	1	1	0	1	--> 14
1	1	1	1	0	

Riportiamo le combinazioni che danno uscita “1” in tabella, suddividendole rispetto al PESO, cioè al numero di “1” presenti in ciascuna combinazione.

	A	B	C	D
8	1	0	0	0
5	0	1	0	1
6	0	1	1	0
9	1	0	0	1
12	1	1	0	0
7	0	1	1	1
13	1	1	0	1
14	1	1	1	0

Metodo di Quine-McCluskey 2

Confrontare le configurazioni di una sezione con tutte le combinazioni della sezione successiva.

	A	B	C	D	
8	1	0	0	0	√
5	0	1	0	1	√
6	0	1	1	0	√
9	1	0	0	1	√
12	1	1	0	0	√
7	0	1	1	1	√
13	1	1	0	1	√
14	1	1	1	0	√

	A	B	C	D
8/9	1	0	0	_
8/12	1	_	0	0
5/7	0	1	_	1
5/13	_	1	0	1
6/7	0	1	1	_
6/14	_	1	1	0
9/13	1	_	0	1
12/13	1	1	0	_
12/14	1	1	_	0

Metodo di Quine-McCluskey 3

Reiterare i confronti

	A	B	C	D	
8	1	0	0	0	√
5	0	1	0	1	√
6	0	1	1	0	√
9	1	0	0	1	√
12	1	1	0	0	√
7	0	1	1	1	√
13	1	1	0	1	√
14	1	1	1	0	√

	A	B	C	D	
8/9	1	0	0	_	√
8/12	1	_	0	0	√
5/7	0	1	_	1	
5/13	_	1	0	1	
6/7	0	1	1	_	
6/14	_	1	1	0	
9/13	1	_	0	1	√
12/13	1	1	0	_	√
12/14	1	1	_	0	

	A	B	C	D
8/9/12/13	1	_	0	_

Implicanti primi E= A !C + !A B D + B !C D + !A B C + B C !D + A B !D

Metodo di Quine-McCluskey 4

- costruzione della tabella di copertura

	5	6	7	8	9	12	13	14
A!C				X	X	X	X	
!ABD	X		X					
B!CD	X						X	
!ABC		X	X					
BC!D		X						X
AB!D						X		X

le colonne 8 e 9 sono si possono “coprire” solo usando A!C che quindi diventa un termine indispensabile

Metodo di Quine-McCluskey 5

	5	6	7	8	9	12	13	14
A!C				X	X	X	X	
!ABD	X		X					
B!CD	X						X	
!ABC		X	X					
BC!D		X						X
AB!D						X		X

E= A !C + ???

	5	6	7	14
!ABD	X		X	
B!CD	X			
!ABC		X	X	
BC!D		X		X
AB!D				X

Metodo di Quine-McCluskey 6

	5	6	7	14
!ABD	X		X	
~~B!CD~~	~~X~~			
!ABC		X	X	
BC!D		X		X
~~AB!D~~				~~X~~

E= A !C + ???

le righe relative a B!CD e AB!D sono dominate dalle righe relative a !ABD e BC!D, rispettivamente

Metodo di Quine-McCluskey 7

	5	6	7	14
!ABD	X		X	
~~B!CD~~	~~X~~			
!ABC		X	X	
BC!D		X		X
~~AB!D~~				~~X~~

E= A !C + ???

	5	6	7	14
!ABD	X		X	
!ABC		X	X	
BC!D		X		X

Metodo di Quine-McCluskey 8

	5	6	7	14
!ABD	X		X	
!ABC		X	X	
BC!D		X		X

E= A !C + !A B D + B C !D

	5	6	7	14
~~!ABD~~	X		X	
!ABC		X	X	
~~BC!D~~		X		X