Circuiti Combinatori

Capitolo 3

Introduzione

- Circuiti combinatori
  - Gli ingressi e le uscite prevedono solo 2 stati logici
    - Logica positiva
      - Livello di tensione basso → 0 logico
      - Livello di tensione alto → 1 logico
    - Logica negativa
      - Livello di tensione basso → 1 logico
      - Livello di tensione alto → 0 logico
  - Le uscite dipendono **esclusivamente** dagli ingressi
  - Circuito elementare o porta logica

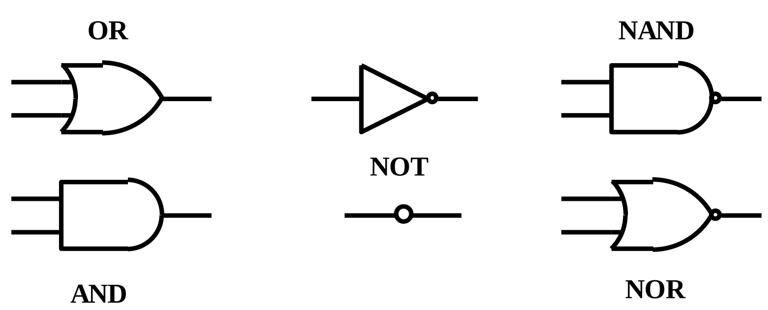


figura 3.1.1

Definizioni

- Itinerari e livelli
  - **Elementi**: porte logiche
  - **Ingressi** ed **Uscite**
  - **Itinerario**: percorso che collega 2 elementi
  - **Livello** di un elemento **X** rispetto un'uscita **U** e secondo un determinato itinerario **I:** il numero di elementi compreso **X …**
  - **Livello** di una variabile rispetto un'uscita **U** e secondo un determinato itinerario **I** : il numero di elementi …

A_i: ingressi
B_i: uscite

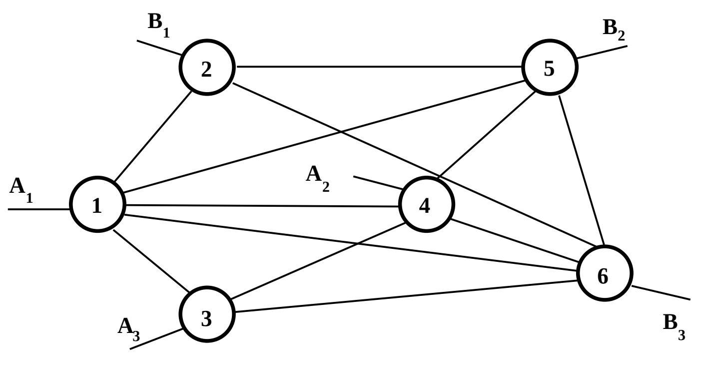


figura 3.2.1

Esempi

Nota: lo stesso elemento puo' possedere diversi livelli secondo diversi itinerari

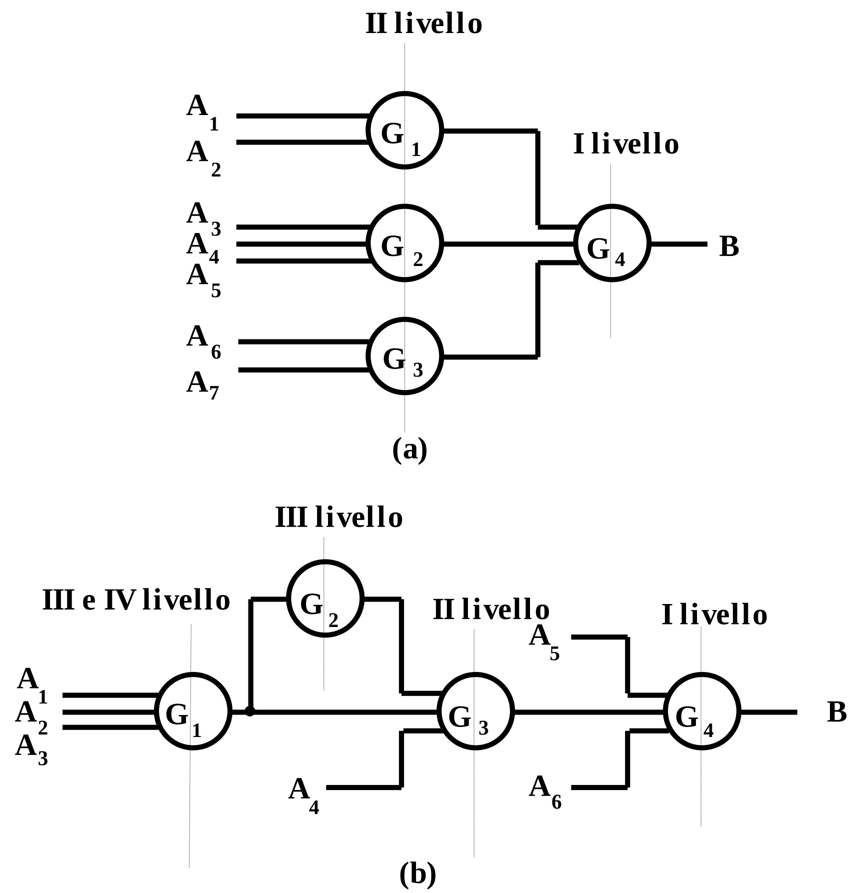


figura 3.2.2

Analisi di circuiti AND-OR-NOT

- Dato il circuito come ottenere la funzione
  - Partendo dagli ingressi
  - Si analizzano le funzioni svolte dai vari elementi
  - Secondo tutti gli itinerari possibili

$$y_A = x_1 . x_2 . \overline{x_3}$$

$$y_{A1} = \overline{x_1 . x_2 . \overline{x_3}} = \overline{x_1} + \overline{x_2} + x_3$$

$$y_{A2} = \overline{x_4}$$

$$y_B = x_4 . y_A = x_1 . x_2 . \overline{x_3} . x_4$$

$$y_C = y_{A1} . y_{A2} = \overline{x_4} . \left(\overline{x_1} + \overline{x_2} + x_3 \right)$$

$$y = y_B + y_C + x_5 = x_1 . x_2 . \overline{x_3} . x_4 + \overline{x_4} . \left(\overline{x_1} + \overline{x_2} + x_3 \right) + x_5$$

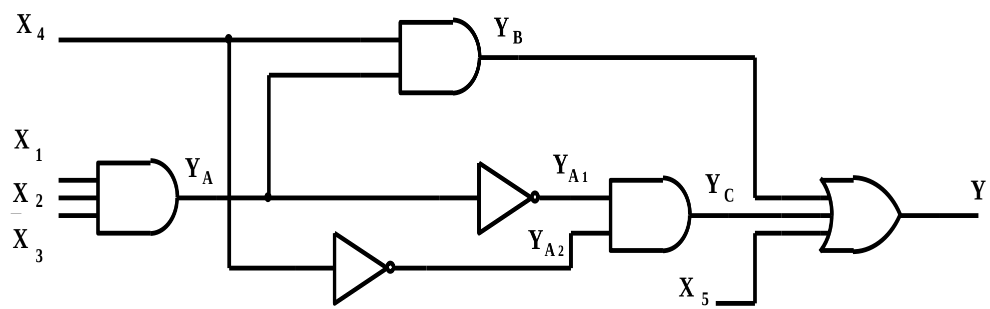


figura 3.3.1

Analisi di circuiti NAND

- NAND: operatore universale
- Le espressioni ricavate con l'analisi dei percorsi possono risultare complesse o quantomento poco comprensibili

$$y = [(x_1/x_2)/x_3/x_4]/(x_4/x_5)/x_6$$

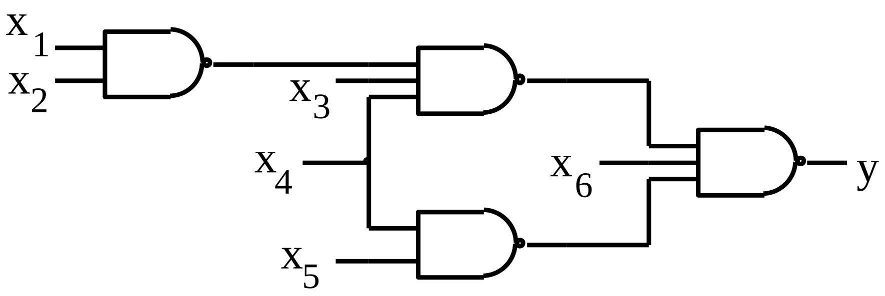


figura 3.4.1

$$y = \overline{\overline{\overline{x_1 . x_2} . x_3 . x_4} . \overline{x_4 . x_5} . x_6} = \overline{\overline{(\overline{x_1} + \overline{x_2}) . x_3 . x_4} . (\overline{x_4} + \overline{x_5}) . x_6} =$$

$$= (\overline{x_1} + \overline{x_2}) . x_3 . x_4 + x_4 . x_5 + \overline{x_6}$$

Analisi di circuiti NAND

- Si puo' agevolmente pervenire ad un risultato in forma di somme di prodotti:

Nota:

• Le variabili ai livelli dispari sono negate, ai livelli pari sono dirette

• Ai livelli dispari vi e' la somma, ai livelli pari il prodotto

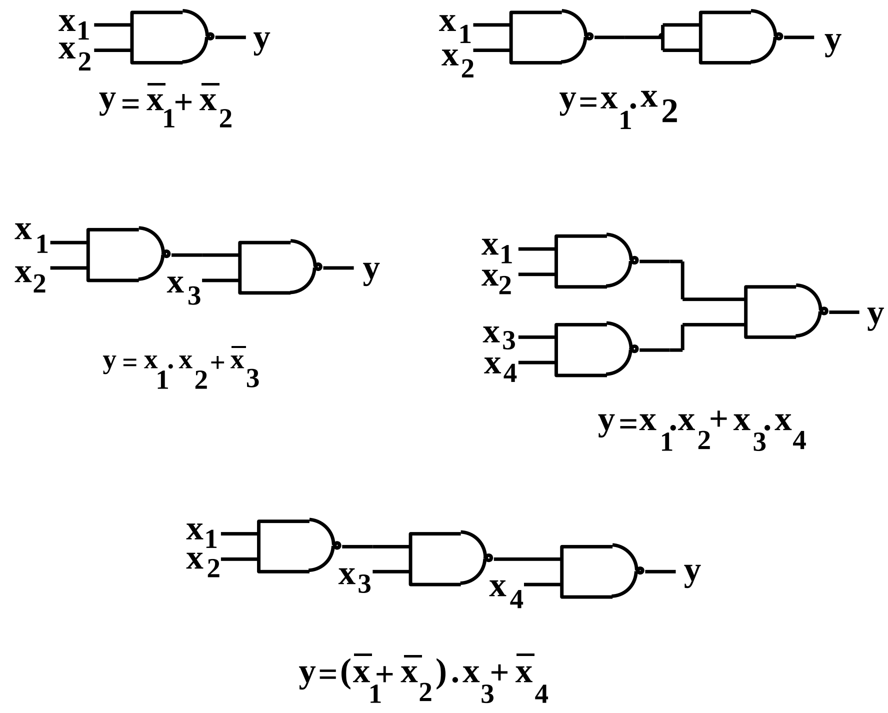


figura 3.4.2

Analisi di circuiti NAND

Esempio:

- Le variabili ai livelli dispari sono negate, ai livelli pari sono dirette
- Ai livelli dispari vi e' la somma, ai livelli pari il prodotto

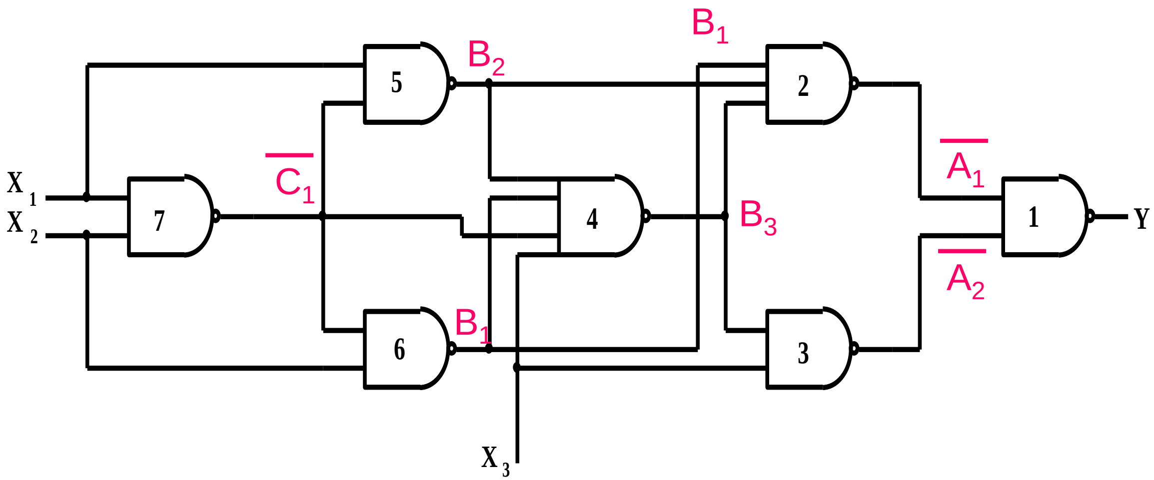


figura 3.4.3

$$y = A_1 + A_2$$

$$A_1 = B_1 B_2 B_3 \qquad A_2 = B_3 x_3$$

$$B_1 = C_1 + \overline{x_2} \qquad B_2 = C_1 + \overline{x_1} \qquad B_3 = \overline{B_2} + \overline{B_1} + C_1 + \overline{x_3}$$

$$C_1 = x_1 \cdot x_2$$

$$y = \left(x_1 . x_2 + \overline{x_2}\right)\left(x_1 . x_2 + \overline{x_1}\right)\left(\left(\overline{x_1} + \overline{x_2}\right) . x_1 + \left(\overline{x_1} + \overline{x_2}\right) . x_2 + x_1 . x_2 + \overline{x_3}\right) +$$
$$+ \left(x_1 . \overline{x_2} + \overline{x_1} . x_2 + x_1 . x_2 + \overline{x_3}\right) x_3 =$$

Analisi di circuiti NAND

Esempio:

- Le variabili ai livelli dispari sono negate, ai livelli pari sono dirette
- Ai livelli dispari vi e' la somma, ai livelli pari il prodotto

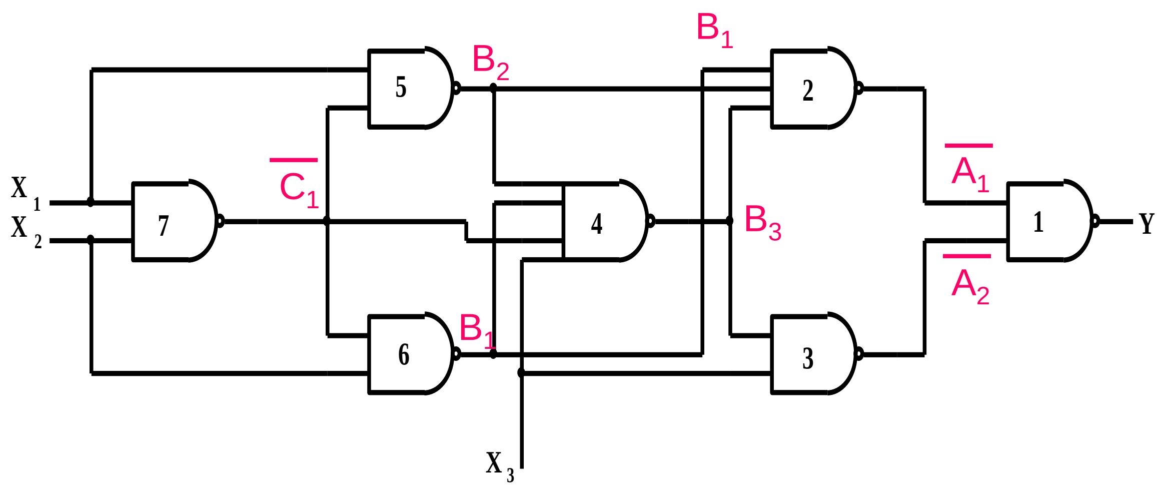


figura 3.4.3

$$y = (\quad) + (\quad)$$

$$y = [(\quad)(\quad)(\quad)] + [(\quad)x_3]$$

$$y = [((\quad) + \overline{x_2})((\quad) + \overline{x_1})(. + . + . + \overline{x_3})] + [(. + . + . + \overline{x_3})x_3]$$

$$y = [((x_1x_2) + \overline{x_2})((x_1x_2) + \overline{x_1})(()() + ()() + ()() + \overline{x_3})] + [(()() + ()() + ()() + \overline{x_3})x_3]$$

$$y = [((x_1x_2) + \overline{x_2})((x_1x_2) + \overline{x_1})((x_2)(\overline{x_1} + \overline{x_2}) + (x_1)(\overline{x_1} + \overline{x_2}) + (x_1)(x_2) + \overline{x_3})] +$$

$$+ [((x_2)(\overline{x_1} + \overline{x_2}) + (x_1)(\overline{x_1} + \overline{x_2}) + (x_1)(x_2) + \overline{x_3})x_3]$$

Analisi di circuiti NOR

Analogamente:

- Le variabili ai livelli dispari sono negate, ai livelli pari sono dirette
- Ai livelli dispari vi e' il prodotto, ai livelli pari la somma

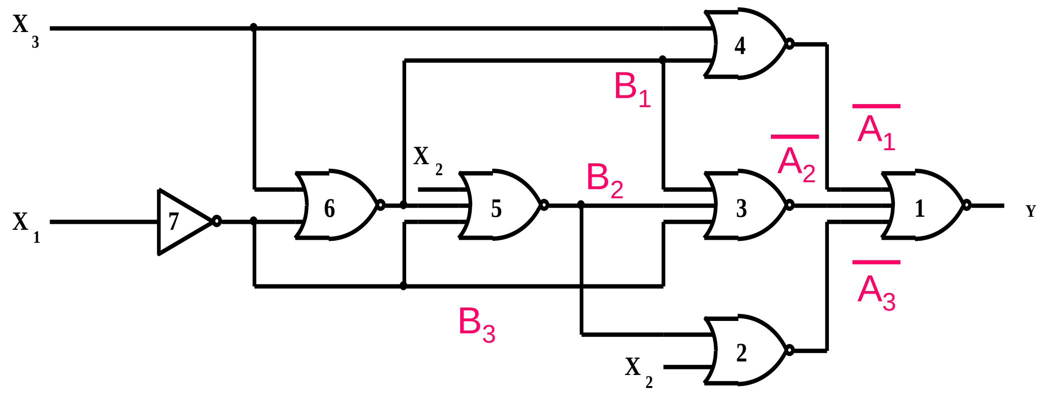


figura 3.5.1

$$y = A_1 A_2 A_3$$

$$A_1 = x_3 + B_1 \qquad A_2 = B_1 + B_2 + B_3 \qquad A_3 = B_2 + x_2$$

$$B_1 = x_1 \overline{x_3} \qquad B_2 = \overline{x_2}\,\overline{B_1}\,\overline{B_3} \qquad B_3 = \overline{x_1}$$

$$y = \left(x_3 + x_1 . \overline{x_3}\right) . \left(x_1 . \bar{x}_3 + \overline{x_2}\left(\overline{x}_1 + x_3\right) x_1 + \bar{x}_1\right)\left(\overline{x_2}\left(\overline{x_1} + x_3\right) x_1 . + x_2\right)$$

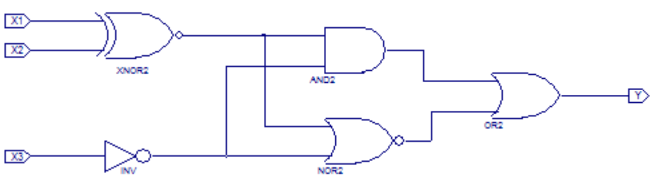
Sintesi di circuiti combinatori

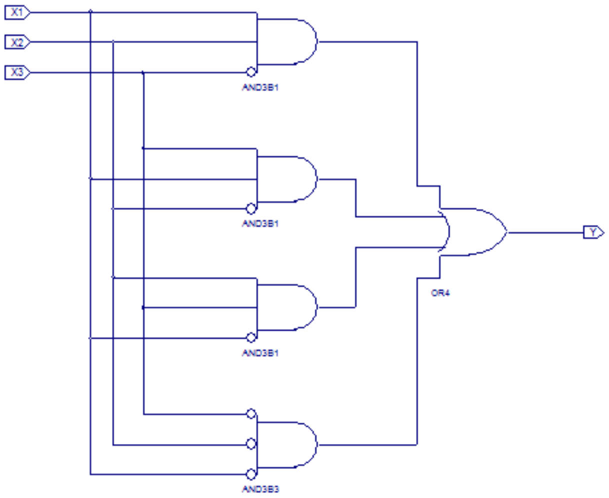
Progettare il circuito che realizza una certa funzione logica

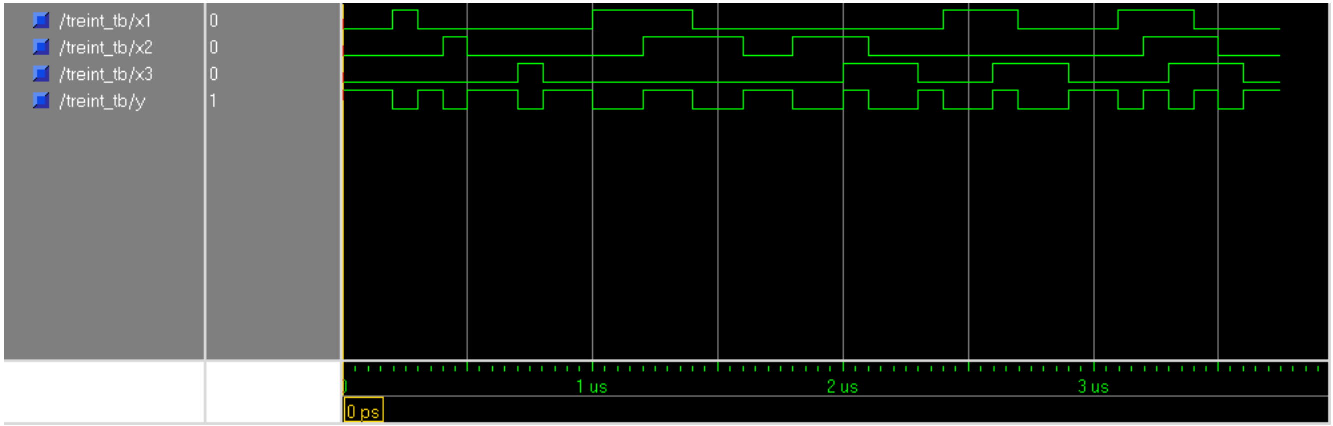
- Vi possono essere diversi modi di esprimere la funzione
  - Descrizione comportamentale o “verbale” (Behavioural)
  - Tabella di verita’
  - Espressione analitica (Dataflow)
  - Schema logico (Schematic)
- Vi sono diversi modi per realizzare una funzione
  - Compromesso Tempo ↔ Area
  - Vincoli (temporali, di occupazione, potenza dissipata, implementativi)
  - La forma minima puo’ non essere ottima oppure puo’ non essere realizzabile (e’ pero’ quella con minor ritardo)
  - La forma piu’ conveniente va’ determinata caso per caso
  - Vi possono essere piu’ uscite e parti del circuito condivise
- La sintesi parte dalla tabella di verita’ ed applica le semplificazioni piu’ **opportune** (definite caso per caso)

Esempio

Due sintesi diverse per un controllore di parita' a tre bit







Sintesi di circuiti AND OR NOT

- Dalla tabella di verita’
- Si ricava la forma minima a 2 livelli
- Individuare la forma minima piu’ conveniente
- Realizzazione il circuito

Esempio : realizzare il prodotto x3
per un numero 0≤ x ≤5

x_1	x_2	x_3	y_1	y_2	y_3	y_4
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	1	1	0	0
1	0	1	1	1	1	1
1	1	0	-	-	-	-
1	1	1	-	-	-	-

figura 3.7.1

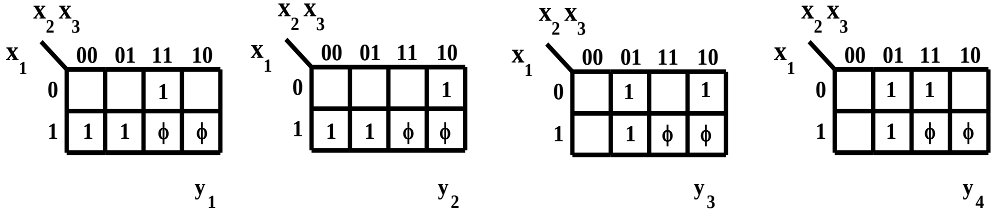


figura 3.7.2

$$y_1 = x_1 + x_2 . x_3$$
$$y_2 = x_1 + x_2 . \overline{x_3}$$
$$y_3 = \overline{x_2} . x_3 + x_2 . \overline{x_3}$$
$$y_4 = x_3$$

Sintesi di circuiti AND OR NOT

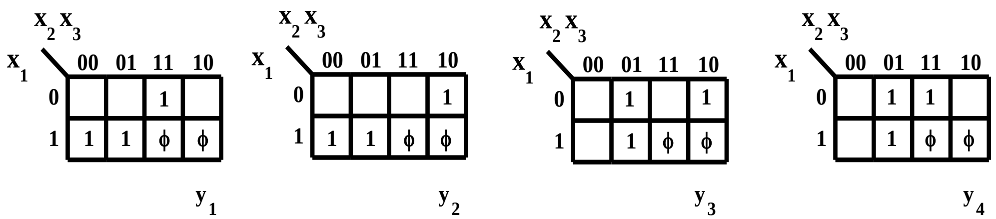


figura 3.7.2

$$y_1 = x_1 + x_2 \cdot x_3$$
$$y_2 = x_1 + x_2 \cdot \overline{x_3}$$
$$y_3 = \overline{x_2} \cdot x_3 + x_2 \cdot \overline{x_3}$$
$$y_4 = x_3$$

si puo' inividuare un fattore a comune tra y_2 e y_3

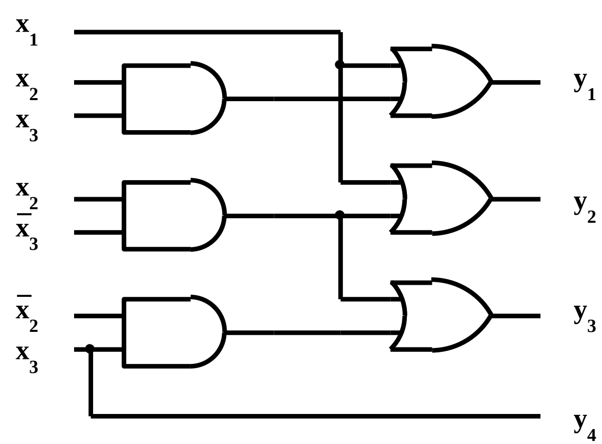


figura 3.7.3

Decomposizione in sconnessione semplice

- E’ un metodo piu’ sistematico
- Consente di “spezzare” la funzione in piu’ funzioni semplici
- Sia $x_1, x_2, x_3, x_4, \ldots, x_n$ un insieme di variabili e siano A e B siano due sottoinsiemi disgiunti di questo insieme X

$$A \cup B = X \qquad \text{e} \qquad A \cap B = \Phi$$

- Sia $f(x_1, x_2, x_3, x_4, \ldots, x_n)$ una funzione booleana
- Se e’ possibile individuare due funzioni t.c.

$$\mathrm{f}(\mathrm{X}) = \mathrm{F}[\varphi(\mathrm{A}), \mathrm{B}]$$

- Allora le funzioni F e φ formano una decomposizione in sconnessione semplice della funzione f
- L’insieme A sono le **variabili al contorno**
- Le variabili B sino **variabili indipendenti**

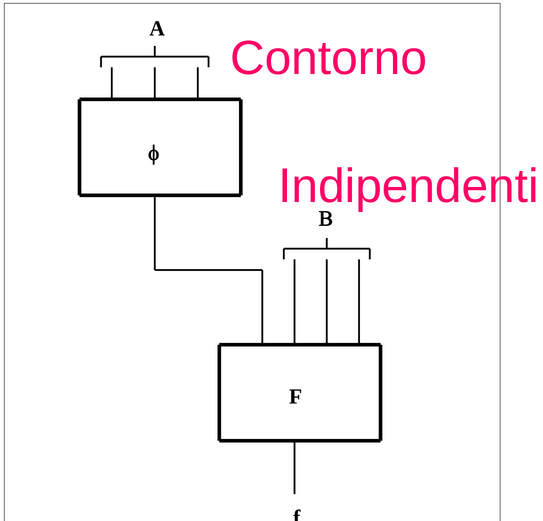


figura 3.8.1

Decomposizione in sconnessione semplice

- Qualunque funzione puo' essere partizionata
  - ad esempio :

$$f(x_1,x_2,x_3,x_4)=\overline{x_2}.\overline{x_3}.\alpha(x_1,x_4)+\overline{x_2}.x_3.\beta(x_1,x_4)$$
$$+x_2.\overline{x_3}.\gamma(x_1,x_4)+x_2x_3.\delta(x_1,x_4)$$

  - Esiste una partizione semplice se e solo se le funzioni α,β,γ,δ sono tutte derivabili dalla stessa funzione φ ovvero se sono o la costante 0, oppure la costante 1, oppure φ o not(φ)
  - Questo puo' venir facilmente evidenziato in particolari mappe di partizione (paricolari mappe di Karnaught)
  - Ovviamente bisogna analizzare TUTTE le possibili partizioni
  - Si analizzano le molteplicita' delle colonne

Decomposizione in sconnessione semplice

- Esempio

$$f(x_1, x_2, x_3, x_4) = \sum(4,5,6,7,8,13,14,15)_m$$

$x_2x_3 \backslash x_1x_4$	00	01	11	10	
00	[0] 0	[1] 0	[9] 0	[8] 1	α
01	[2] 0	[3] 0	[11] 0	[10] 0	β
11	[6] 1	[7] 1	[15] 1	[14] 1	δ
10	[4] 1	[5] 1	[13] 1	[12] 0	γ

figura 3.8.2

$$f(x_1, x_2, x_3, x_4) = \overline{x_2}.\overline{x_3}.\varphi(x_1, x_4) + x_2 .\overline{x_3}.\overline{\varphi}(x_1, x_4) + x_2 . x_3 =$$

$$= F(\varphi(x_1, x_4), x_2, x_3)$$

con:

$$\varphi(x_1, x_4) = x_1 .\overline{x_4}$$

Decomposizione in sconnessione semplice

- Teorema: Una mappa di partizione corrisponde ad una decomposizione in sconnessione semplice se e solo se la molteplicita' di colonna e' minore o uguale a 2.

  Ovvero: avere una molteplicita' di colonna inferiore a 2 e' condizione necessaria e sufficiente:
  - Necessaria: se la molteplicita' e' minore di 2 la funzione puo' essere scritta nella forma $f(X) = F[\varphi(A), B]$ e pertanto la funzione risulta decomponibile
  - Sufficiente: Se la funzione e' decomponibile puo' essere scritta nella forma $f(X) = F[\varphi(A), B]$ e pertanto nella mappa di partizione la molteplicita' di colonna sara' inferiore a 2.

Decomposizione in sconnessione semplice

- Per 4 variabili le mappe sono:
  - 4 tabelle di dimensione 2x8 (4 partizioni 1+3)
  - 6 (3) tabelle di dimensione 4x4 (6 partizioni 2+2)
- Per 5 variabili le mappe sono:
  - 5 tabelle di dimensione 2x16 (5 partizioni 1+4)
  - 10 (5) tabelle di dimensione 4x8 (10 partizioni 3+2)
- Per 6 variabili le mappe sono:
  - 6 tabelle di dimensione 2x32 (6 partizioni 1+5)
  - 15 tabelle di dimensione 4x16 (15 partizioni 2+4)
  - 20 (10) tabelle di dimensione 8x8 (20 partizioni 3+3)
- Per 7 variabili le mappe sono:
  - 7 …. 21 … 35

Decomposizione in sconnessione semplice

$$f(x_1, x_2, x_3, x_4) =$$
$$= \sum (4,5,6,7,8,13,14,15)_m$$

Possono esistere varie partizioni

$$f(x_2/x_1,x_3,x_4) = \bar{x}_2.(x_1.\bar{x}_3.\bar{x}_4) + x_2.(\overline{x_1.\bar{x}_3.\bar{x}_4})$$

$$f(x_2,x_3/x_1,x_4) = x_2.x_3 + \bar{x}_2.\bar{x}_3.(\overline{\bar{x}_1 + x_4}) + x_2.\bar{x}_3.(\bar{x}_1 + x_4)$$

$$f(x_1,x_2/x_3,x_4) = \bar{x}_1.x_2 + x_1.\bar{x}_2.(\overline{x_3 + x_4}) + x_1.x_2.(x_3 + x_4)$$

$$f(x_2,x_4/x_1,x_3) = x_2.x_4 + x_2.\bar{x}_4.(\bar{x}_1 + x_3) + \bar{x}_2.\bar{x}_4.(\overline{\bar{x}_1 + x_3})$$

CARTA DI DECOMPOSIZIONE PER QUATTRO VARIABILI

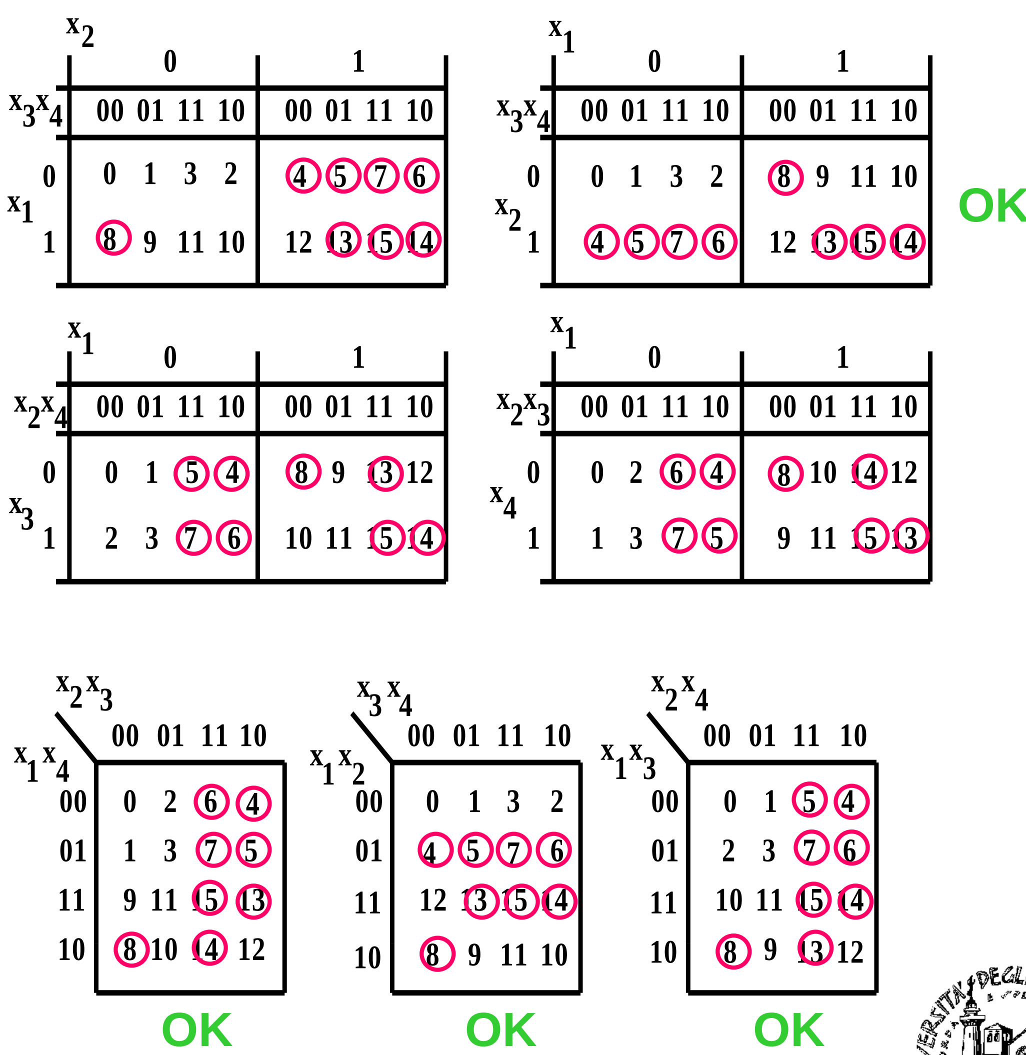


figura 3.8.3

Altre decomposizioni Disgiuntive

- Decomposizione in sconnessione multipla
  - Siano A_1,A_2, … sottinsiemi disgiunti di variabili

$$f(X) = F[\psi_1(A_1), \psi_2(A_2), \dots, \psi_{m-1}(A_{m-1}), \psi_m(A_m)]$$

oppure

$$f(X) = F[\psi_1(A_1), \psi_2(A_2), \dots, \psi_{m-1}(A_{m-1}), A_m]$$

- Decomposizione in sconnessione iterativa

$$f(X) = F[\psi_2(\psi_1[A_1], A_2), A_3]$$

- Decomposizione in sconnessione complessa

$$f(X) = F[\gamma(\psi[A_1], A_2), \lambda(A_3), A_4]$$

Teoremi sulle decomposizioni disgiuntive

- Decomposizione iterativa

Se per una funzione esistono 2 decomposizioni in sconnessione semplice

$$f(X) = F[\lambda(A, B), C] = G[\psi(A), B, C]$$

Esiste allora la decomposizione in sconnessione iterativa

$$f(X) = F[\rho(\psi[A], B), C]$$

con:

$$\rho(\psi[A], B) = \lambda(A, B)$$

Teoremi sulle decomposizioni disgiuntive

- ESEMPIO Decomposizione iterativa

$$f(x_1, x_2, x_3, x_4, x_5) = \sum(5,10,11,14,17,21,26,30)_m$$

$$f(X) = F[\psi(x_1, x_3, x_5), x_2, x_4]$$

$$f(X) = G[\lambda(x_1, x_3), x_2, x_4, x_5]$$

$$A = (x_1, x_3) \qquad B = (x_5)$$

$$C = (x_2, x_4)$$

x_1	0				1			
$x_2 x_4$ \ $x_3 x_5$	00	01	11	10	00	01	11	10
00	0	1	5	4	16	17	21	20
01	2	3	7	6	18	19	23	22
11	10	11	15	14	26	27	31	30
10	8	9	13	12	24	25	29	28

x_2	0				1			
$x_1 x_3$ \ $x_4 x_5$	00	01	11	10	00	01	11	10
00	0	1	3	2	8	9	11	10
01	4	5	7	6	12	13	15	14
11	20	21	23	22	28	29	31	30
10	16	17	19	18	24	25	27	26

figura 3.9.1

$$f(X) = \overline{x_2}.\overline{x_4}.\psi(x_1, x_3, x_5) + x_2 . x_4 .\overline{\psi}(x_1, x_3, x_5)$$

con (riunendo i termini adiacenti secondo Karnaugh):

$$\psi(x_1, x_3, x_5) = x_1 . x_5 + x_3 . x_5$$

che puo' essere scritto come:

$$\psi(x_1, x_3, x_5) = x_5 .(x_1 + x_3) \qquad \text{C.V.D.}$$

Teoremi sulle decomposizioni disgiuntive

- Decomposizione multipla

Se per una funzione esistono 2 decomposizioni in sconnessione semplice

$$\mathrm{f}(\mathrm{X}) = F[\lambda(A), B] = G[\psi(B), A]$$

Esiste allora la decomposizione in sconnessione multipla

$$\mathrm{f}(\mathrm{X}) = H[\lambda(A), \psi(B)]$$

NB: la condizione appena posta si puo' intendere verificata quando sulla carta di partizione vi e' una molteplicita' inferiore a 2 sia per le righe che per le colonne ovvero tanto per la carta diretta che per la trasposta

Teoremi sulle decomposizioni disgiuntive

- ## ESEMPIO Decomposizione multipla

Controllore di parita’

Vale sia la condizione:

$$f(x_1,x_2/x_3,x_4)=\left(\overline{x_1}.\overline{x_2}+x_1.x_2\right).\overline{\psi}(x_3,x_4)+\left(\overline{x_1}.x_2+x_1.\overline{x_2}\right).\psi(x_3,x_4)$$

$$\psi(x_3,x_4)=\overline{x_3}.x_4+x_3.\overline{x_4}$$

quanto la condizione:

$$f(x_3,x_4/x_1,x_2)=\left(\overline{x_3}.x_4+x_3.\overline{x_4}\right).\lambda(x_1,x_2)+\left(\overline{x_3}.\overline{x_4}+x_3.x_4\right).\overline{\lambda}(x_1,x_2)$$

$$\lambda(x_1,x_2)=\overline{x_1}.x_2+x_1.\overline{x_2}$$

x_3x_4 \ x_1x_2	00	01	11	10
00	1		1	
01		1		1
11	1		1	
10		1		1

figura 3.9.2

Il teorema enunciato indica l'esistenza di una decomposizione multipla:

$$f(X)=H\left[\psi(x_3,x_4),\lambda(x_1,x_2)\right]$$

Per trovare H

$$F(\psi,x_1,x_2)$$

$$H(\psi,\lambda)=\overline{\psi}.\overline{\lambda}+\psi.\lambda$$

ψ \ x_1x_2	00	01	11	10
0	1		1	
1		1		1
	$\lambda=0$	$\lambda=1$	$\lambda=0$	$\lambda=1$

ψ \ λ	0	1
0	1	
1		1

figura 3.9.4

Teoremi sulle decomposizioni disgiuntive

- Decomposizione multipla (secondo caso)

Se per una funzione esistono 2 decomposizioni in sconnessione semplice

$$\mathrm{f}(\mathrm{X}) = F[\lambda(A), B, C] = G[\psi(B), A, C]$$

Esiste allora la decomposizione in sconnessione multipla

$$\mathrm{f}(\mathrm{X}) = H[\lambda(A), \psi(B), C]$$

Teoremi sulle decomposizioni disgiuntive

- ESEMPIO Decomposizione multipla (secondo caso)

x_1x_2 \ x_3 / x_4x_5	0: 00	0: 01	0: 11	0: 10	1: 00	1: 01	1: 11	1: 10
00	0	1	3	2	4	5	7	6
01	8	9	11	10	12	13	15	14
11	24	25	27	26	28	29	31	30
10	16	17	19	18	20	21	23	22

x_3x_4 \ x_1 / x_2x_5	0: 00	0: 01	0: 11	0: 10	1: 00	1: 01	1: 11	1: 10
00	0	1	9	8	16	17	25	24
01	2	3	11	10	18	19	27	26
11	6	7	15	14	22	23	31	30
10	4	5	13	12	20	21	29	28

figura 3.9.5

$$f(x_3,x_4,x_5/x_1,x_2)=\left(\overline{x_3}.\overline{x_4}.\overline{x_5}+\overline{x_3}.x_4.x_5+x_3.\overline{x_4}.x_5+x_3.x_4.\overline{x_5}\right).\lambda(x_1,x_2)+$$
$$+\left(\overline{x_3}.\overline{x_4}.x_5+\overline{x_3}.x_4.\overline{x_5}+x_3.\overline{x_4}.\overline{x_5}+x_3.x_4.x_5\right).\overline{\lambda}(x_1,x_2)$$

$$\lambda(x_1,x_2)=\overline{x_1}.x_2+x_1.\overline{x_2}$$

$$f(x_1,x_2,x_5/x_3,x_4)=\left(\overline{x_1}.\overline{x_2}.\overline{x_5}+\overline{x_1}.x_2.x_5+x_1.\overline{x_2}.x_5+x_1.x_2.\overline{x_5}\right).\varphi(x_3,x_4)+$$
$$+\left(\overline{x_1}.x_2.\overline{x_5}+\overline{x_1}.\overline{x_2}.x_5+x_1.\overline{x_2}.\overline{x_5}+x_1.x_2.x_5\right).\overline{\varphi}(x_3,x_4)$$

$$\varphi(x_3,x_4)=\overline{x_3}.x_4+x_3.\overline{x_4}$$

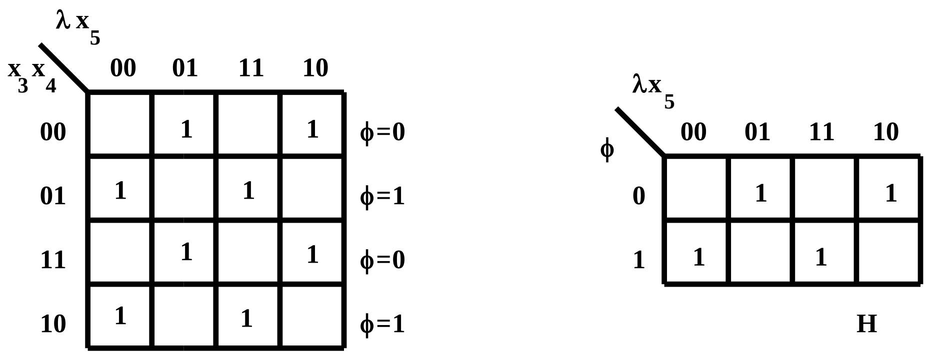
Teoremi sulle decomposizioni disgiuntive

- ESEMPIO Decomposizione multipla (secondo caso)

$$A = (x_1, x_2)$$

$$B = (x_3, x_4)$$

$$C = (x_5)$$



(a) (b)

figura 3.9.6

$$H(\lambda, \varphi, x_5) = (\bar{\lambda}.\overline{x_5} + \lambda.x_5).\varphi + (\bar{\lambda}.x_5 + \lambda.\overline{x_5}).\bar{\varphi}$$

$$\rho(\lambda, x_5) = \bar{\lambda}.x_5 + \lambda.\overline{x_5}$$

si ottiene in definitiva:

$$f(X) = \bar{\rho}.\varphi + \rho.\bar{\varphi}$$

Sintesi di circuiti NAND

- NAND: operator universale
- La sintesi tende ad ottimizzare (minimizzare) il numero di gate
  - La sintesi e’ diversa se
    - sono presenti sia le variabili dirette che negate
    - sono presenti solo le variabili dirette
      (si devono impiegare dei NAND come invertitori)

Sintesi circuiti NAND

- In caso di presenza di variabili dirette e negate
- si possono sintetizzare eq. sotto forma di somme di prodotti
  - livelli dispari : somme
  - livelli pari prodotti
  - livelli dispari: variabili negate
  - livelli pari variabili dirette
- Si possono anche sintetizzare eq. sotto forma di prodotto di somme aggiungendo un invertitore

$$f(X) = \prod \varphi_i (x_1, ..., x_n) = \left[\prod \varphi_i (x_1, ..., x_n)\right] + 0$$

$$x / 1 = \overline{x . 1} = \overline{x}$$

Sintesi circuiti NAND

- Esempio

$$y = x_1.(x_2.x_3.x_4 + x_5) + \left[\left(\overline{x_2} + \overline{x_3} + \overline{x_4}\right).\overline{x_5} + x_6\right].x_7$$

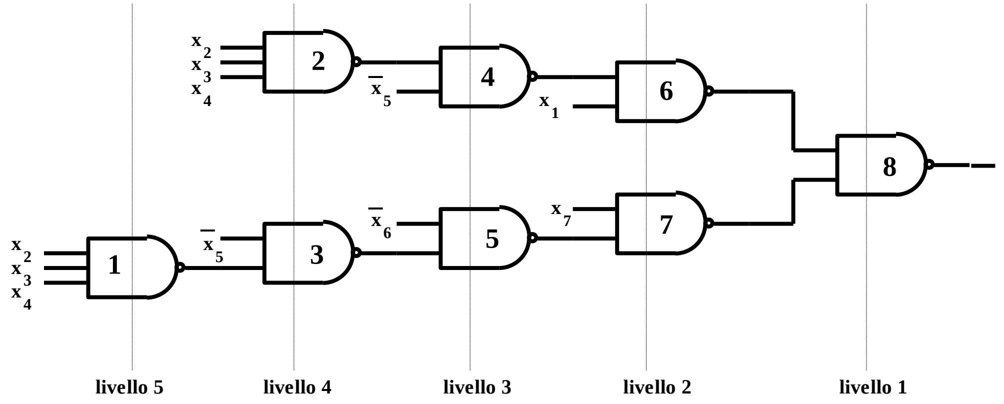


figura 3.10.1

Sintesi circuiti NAND

- Nota: Se una porta NAND presentasse troppi ingressi
  - per ripartire l'operazione su piu' porte si deve aggiungere un invertitore

$$x_1/x_2/x_3/..../x_k = \left(\overline{x_1/x_2/x_3}\right)/..../x_k$$

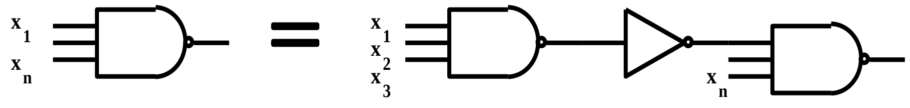


figura 3.10.2

Sintesi di circuiti NAND

- Se sono presenti solo variabili dirette
  - Bisogna cercare di far entrare le variabili negate su livelli dispari
  - Aggiungendo al limite termini ridondanti
  - La soluzione richiede comunque una certa pratica
- Esempio

$$y = x_1 . \left(\overline{x_2} + \overline{x_3} + x_4 \right) + \overline{x_2} . x_3 . x_4$$

  - richiede 4 NAND e 2 invertitori

$$y = x_1 . \left(\overline{x_2} + \overline{x_3} \right) + x_1 . x_4 + \left(\overline{x_2} + \overline{x_3} \right) . x_3 . x_4$$

  - richiederebbe 6 NAND ma una analisi del circuito dimostra che un componente e' replicato

Tecnica del Bundling

- Per eliminare invertitori (non agli ingressi)
  - Aumentano pero’ gli ingressi (fan-in) delle porte logiche

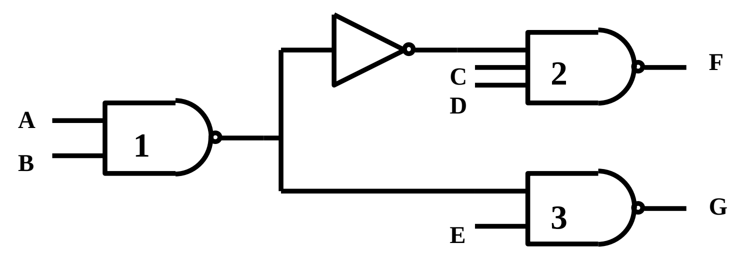


figura 3.11.1

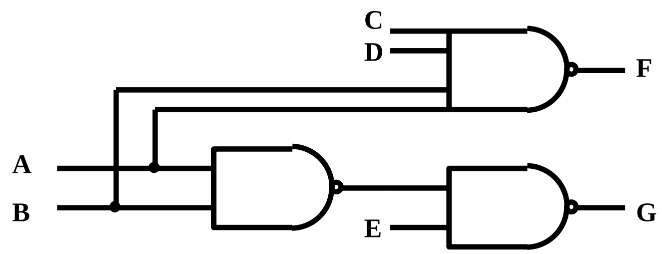


figura 3.11.2

Sintesi circuiti NOR

- E’ simile al caso della sintesi con NAND
- Si possono sintetizzare eq. sotto forma di prodotto di somme
  - livelli dispari : prodotto
  - livelli pari: somme
  - livelli dispari: variabili negate
  - livelli pari: variabili dirette
- Si possono anche sintetizzare eq. sotto forma di somme di prodotti aggiungendo un invertitore

$$y = \sum \varphi_i\left(x_1,\dots,x_n\right) = \left[\sum \varphi_i\left(x_1,\dots,x_n\right)\right] \cdot 1$$

$$x \downarrow 0 = \overline{x + 0} = \overline{x}$$

Sommatori

- Semisommatore, sommatore senza carry in
  Half Adder

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \overline{A}.B + A.\overline{B}$$

$$R = A.B$$

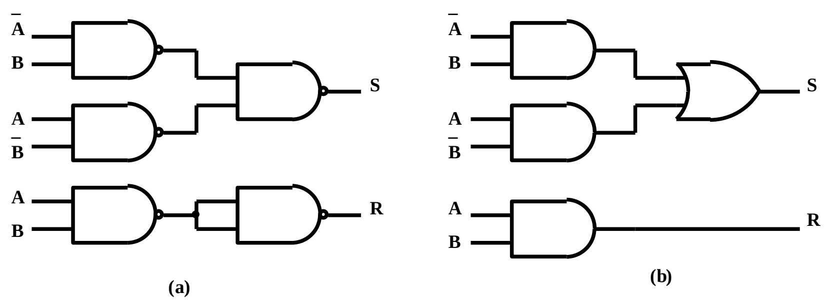


figura 3.13.1

$$S = \left(\overline{A} + \overline{B}\right).\left(A + B\right)$$

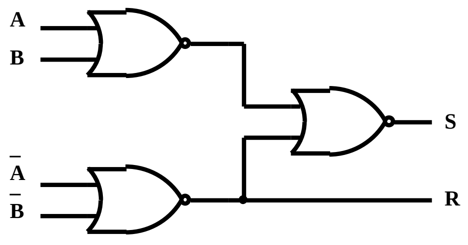


figura 3.13.2

Sommatori

- Full Adder
  - Si puo' sintetizzare la funzione a 3 ingressi
  - Si possono iterare due Half-Adder

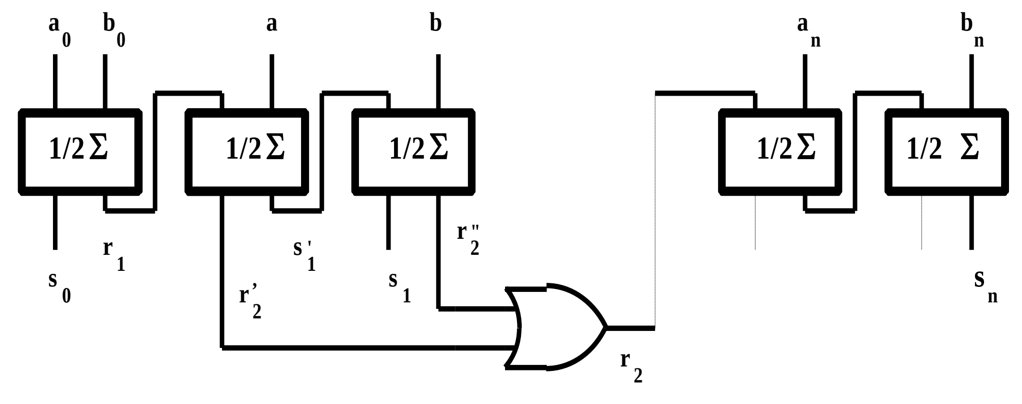
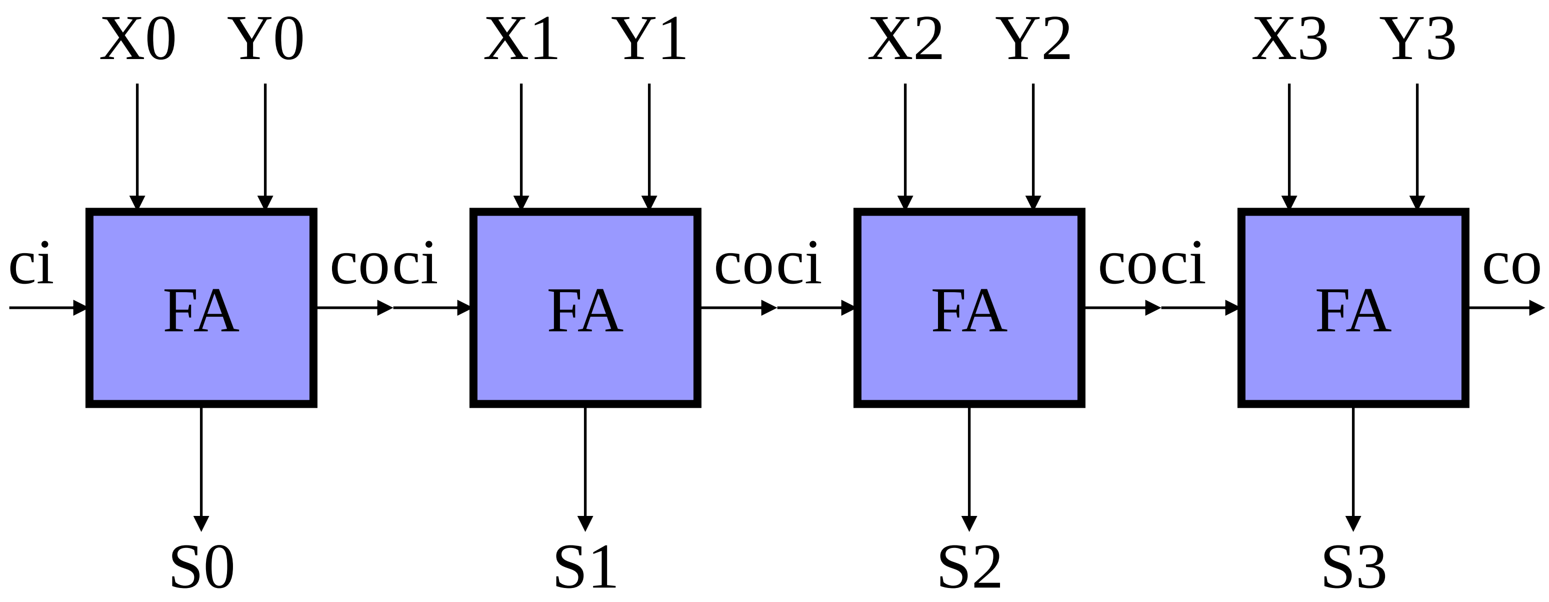


figura 3.13.3

SOMMATORE - Ripple Carry (RC)

- Riporto seriale del Bit Carry dal LSB al MSB
- E’ lento ma semplice



SOMMATORE - Carry Look Ahead

- Si puo' definire la presenza o meno del Carry in posizione *i-esima* dall'analisi dei bit precedenti
- E' molto veloce
- Può risultare piuttosto complesso
- La logica che definisce la presenza del Carry passa attraverso il calcolo di due parametri:
  - Gi (Generate)
  - Pi (Propagate)

$$\begin{cases} P_i = X_i + Y_i \\ G_i = X_i \cdot Y_i \\ C_i = C_{i-1} \cdot P_i + G_i \end{cases}$$

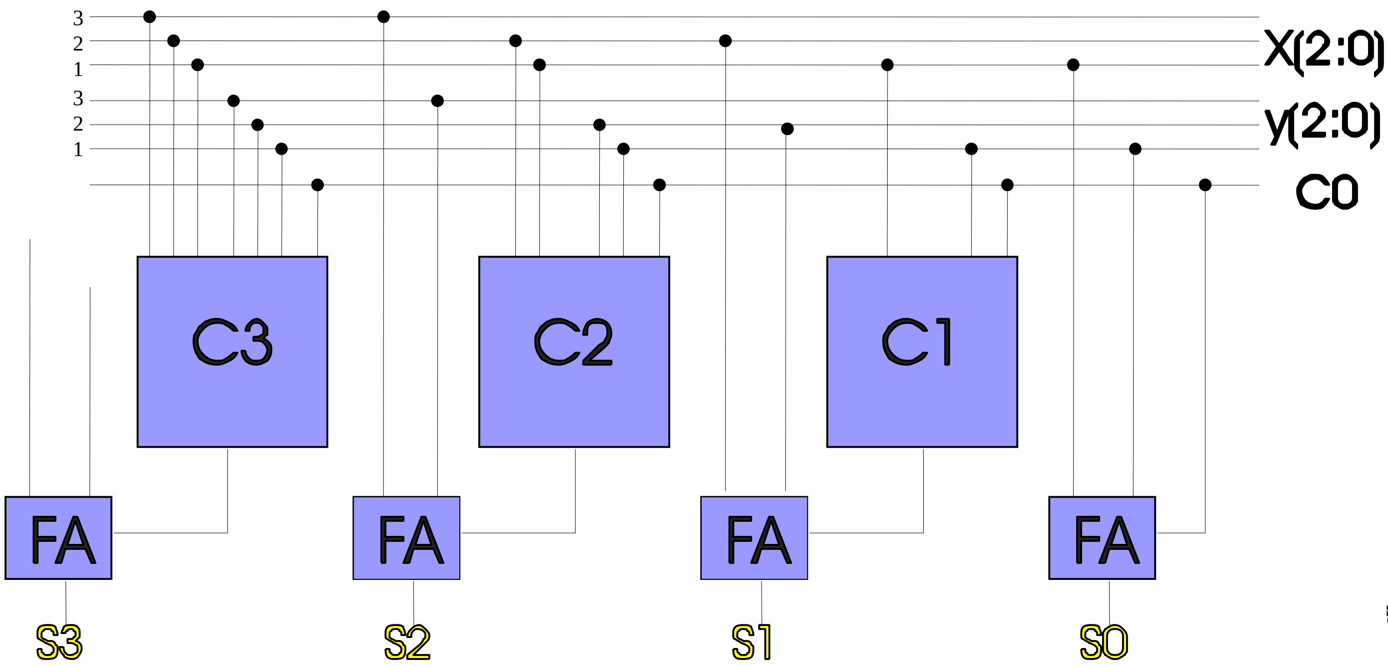
SOMMATORE - Carry Look Ahead

$$C_1 = C_0 \cdot P_1 + G_1$$

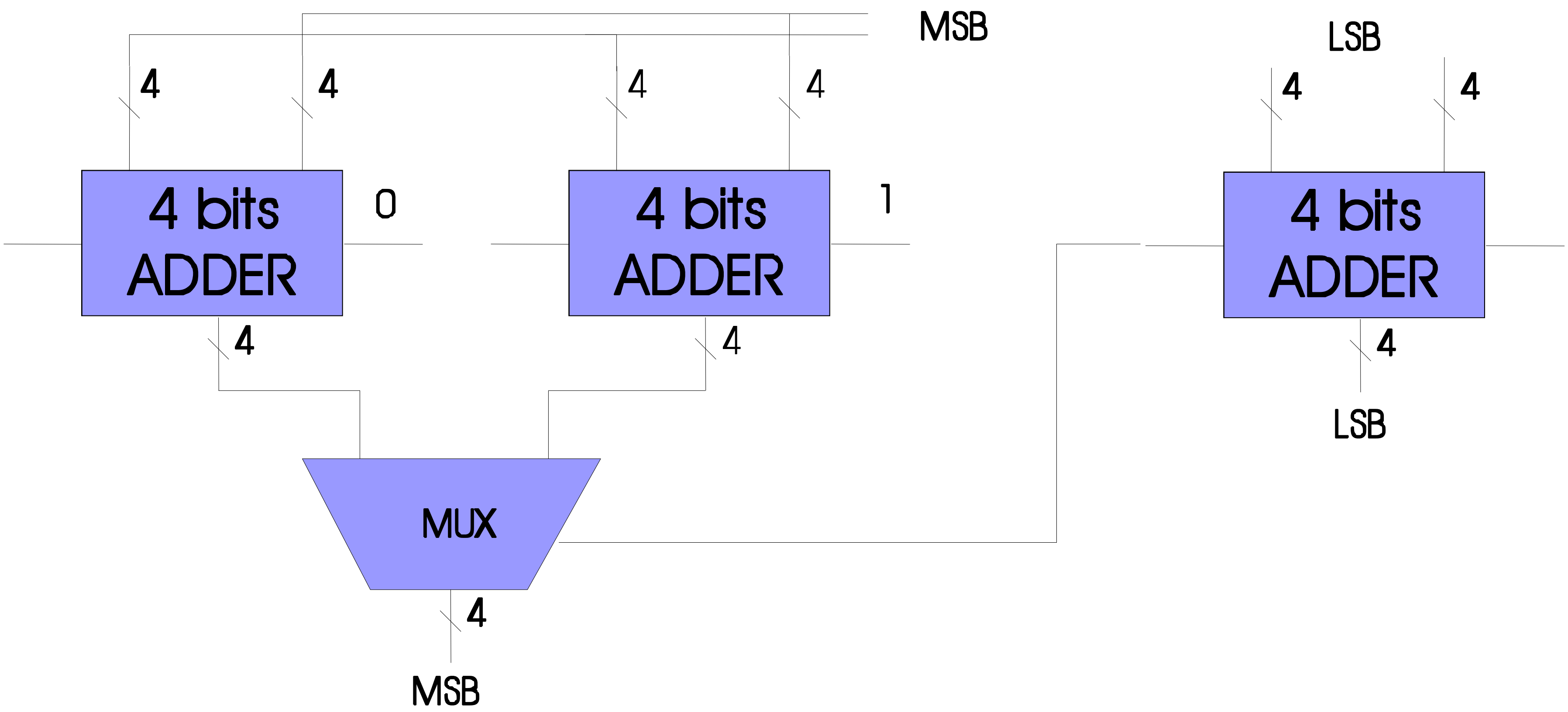
$$C_2 = (C_0 \cdot P_1 + G_1) \cdot P_2 + G_2$$

$$C_3 = ((C_0 \cdot P_1 + G_1) \cdot P_2 + G_2) \cdot P_3 + G_3$$

$$C_4 = \ldots$$



Sommatore - Carry Select



Circuiti multiterminali

- Sono circuiti che presentano piu' uscite
  - Le eventuali semplificazioni debbono essere operate tenendo in considerazioni termini a comune fra le diverse uscite
  - Fino a 5/6 variabili si possono usare le mappe di Karnaught
    - Non e' un metodo del tutto rigoroso
    - Si realizzano le mappe per le varie funzioni di uscita
    - Si evidenzino i sottoinsiemi a comune fra piu' mappe (eventualmente spezzandoli in modo opportuno)
  - Altrimenti ci si rifà ad una modifica del metodo di Quine McCluskey
    - Nelle tabelle, oltre agli implicanti si evidenziano anche le uscite alle quali detti implicanti sono necessari

Mappe di Karnaught

Esempio

$$y_1 = \sum(0,4,6,7,10,14,15)_m$$

$$y_2 = \sum(4,5,6,7,11,12,13,14,15)_m$$

$$y_3 = \sum(0,1,4,5,9,10,11,13,14,15)_m$$

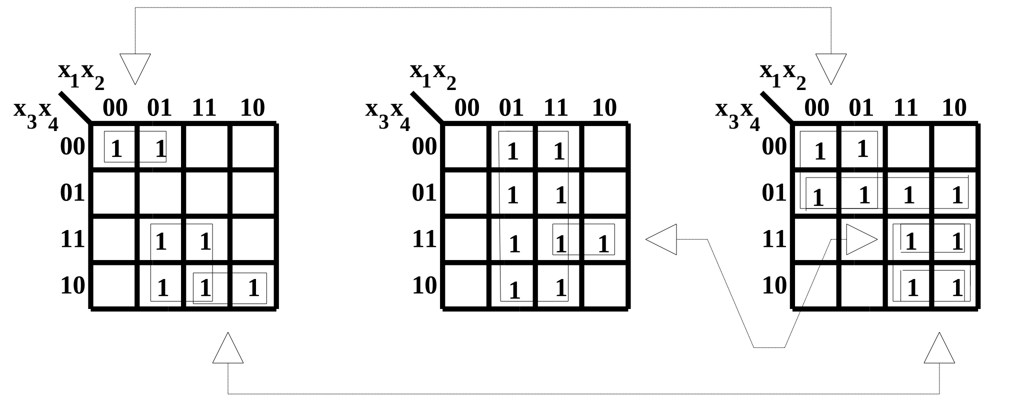


figura 3.14.1

Semplificazione "separata"

$$y_1 = \overline{x_1}.\overline{x_3}.\overline{x_4} + x_2.x_3 + x_1.x_3.\overline{x_4}$$

$$y_2 = x_2 + x_1.x_3.x_4$$

$$y_3 = \overline{x_1}.\overline{x_3} + \overline{x_3}.x_4 + x_1.x_3$$

Richiede 10 gates

Semplificazione "congiunta"

$$y_1 = \overline{x_1}.\overline{x_3}.\overline{x_4} + x_2.x_3 + x_1.x_3.\overline{x_4}$$

$$y_2 = x_2 + x_1.x_3.x_4$$

$$y_3 = \overline{x_3}.x_4 + x_1.x_3.x_4 + x_1.x_3.\overline{x_4} + \overline{x_1}.\overline{x_3}.\overline{x_4}$$

Richiede 8 gates

Metodo di Quine - McCluskey

x_1	x_2	x_3	x_4	y_1	y_2	y_3
0	0	0	0	0	0	0
0	0	0	1	1	0	1
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
0	1	1	1	1	1	0
1	0	0	0	0	0	0
1	0	0	1	1	0	1
1	0	1	0	0	1	1
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

1	0	0	0	1	y_1	-	y_3
4	0	1	0	0	-	y_2	y_3
6	0	1	1	0	y_1	y_2	-
5	0	1	0	1	-	y_2	y_3
12	1	1	0	0	-	y_2	y_3
10	1	0	1	0	-	y_2	y_3
9	1	0	0	1	y_1	-	y_3
7	0	1	1	1	y_1	y_2	-
13	1	1	0	1	-	y_2	y_3
14	1	1	1	0	y_1	y_2	y_3
11	1	0	1	1	y_1	-	y_3
15	1	1	1	1	y_1	y_2	y_3

Metodo di Quine - McCluskey

1	0	0	0	1	y_1	-	y_3
4	0	1	0	0	-	y_2	y_3
6	0	1	1	0	y_1	y_2	-
5	0	1	0	1	-	y_2	y_3
12	1	1	0	0	-	y_2	y_3
10	1	0	1	0	-	y_2	y_3
9	1	0	0	1	y_1	-	y_3
7	0	1	1	1	y_1	y_2	-
13	1	1	0	1	-	y_2	y_3
14	1	1	1	0	y_1	y_2	y_3
11	1	0	1	1	y_1	-	y_3
15	1	1	1	1	y_1	y_2	y_3

	1,5	0	-	0	1	-	-	y_3
A ...	1,9	-	0	0	1	y_1	-	y_3
	4,6	0	1	-	0	-	y_2	-
	4,5	0	1	0	-	-	y_2	y_3
	4,12	-	1	0	0	-	y_2	y_3
	6,7	0	1	1	-	y_1	y_2	-
	6,14	-	1	1	0	y_1	y_2	-
	5,7	0	1	-	1	-	y_2	-
	5,13	-	1	0	1	-	y_2	y_3
	12,13	1	1	0	-	-	y_2	y_3
	12,14	1	1	-	0	-	y_2	y_3
B ...	10,14	1	-	1	0	-	y_2	y_3
	10,11	1	0	1	-	-	-	y_3
	9,13	1	-	0	1	-	-	y_3
C ...	9,11	1	0	-	1	y_1	-	y_3
	13,15	1	1	-	1	-	y_2	y_3
	14,15	1	1	1	-	y_1	y_2	y_3
D ...	11,15	1	-	1	1	y_1	-	y_3
	7,15	-	1	1	1	y_1	y_2	-

E ...	1,5,9,13	-	-	0	1	-	-	y_3
	4,5,6,7	0	1	-	-	-	y_2	-
F ...	4,5,12,13	-	1	0	-	-	y_2	y_3
	4,6,12,14	-	1	-	0	-	y_2	-
G ...	6,7,14,15	-	1	1	-	y_1	y_2	-
	5,7,13,15	-	1	-	1	-	y_2	-
H ...	12,13,14,15	1	1	-	-	-	y_2	y_3
I ...	10,11,14,15	1	-	1	-	-	-	y_3
L ...	9,11,13,15	1	-	-	1	-	-	y_3

M ...	4,5,6,7,12,13,14,15	-	1	-	-	-	y_2	-

Nota: A- Implicante fondamentale per la funzione y_1

Metodo di Quine - McCluskey

Copertura minima

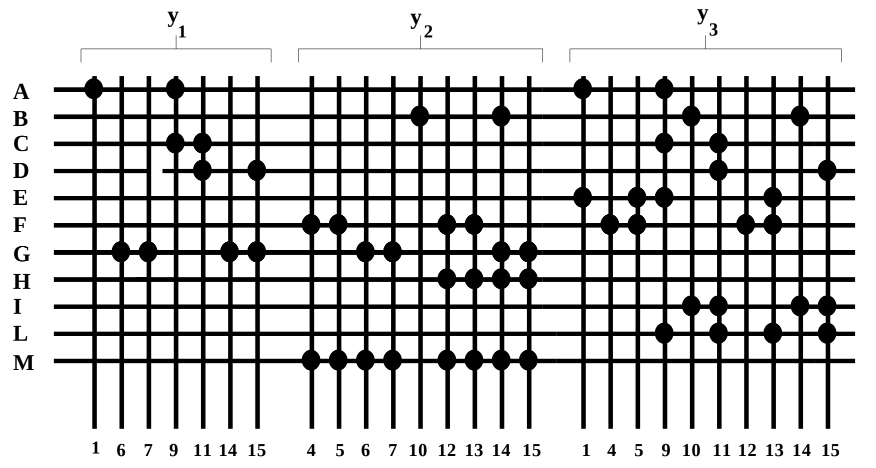


figura 3.14.1

Implicanti essenziali:

A copre i termini minimi 1 e 9 in y1 e y3.
B copre i termini minimi 10 e 14 in y2 e y3.
F copre i termini minimi 4, 5, 12 e 13 in y2 e y3.
G copre i termini minimi 6, 7, 14 e 15 in y1 e y2.

Rimangono scoperti il termine minimo 11 in y1 e y3 e quello 15 in y3; l'implicante primo D li copre entrambi.

Metodo di Quine - McCluskey

$$y_1 = G + A + D = x_2 . x_3 + \overline{x_2} . \overline{x_3} . x_4 + x_1 . x_3 . x_4$$

$$y_2 = G + F + B = x_2 . x_3 + x_2 . \overline{x_3} + x_1 . x_3 . \overline{x_4}$$

$$y_3 = D + F + B + A = x_1 . x_3 . x_4 + x_2 . \overline{x_3} + x_1 . x_3 . \overline{x_4} + \overline{x_2} . \overline{x_3} . x_4$$

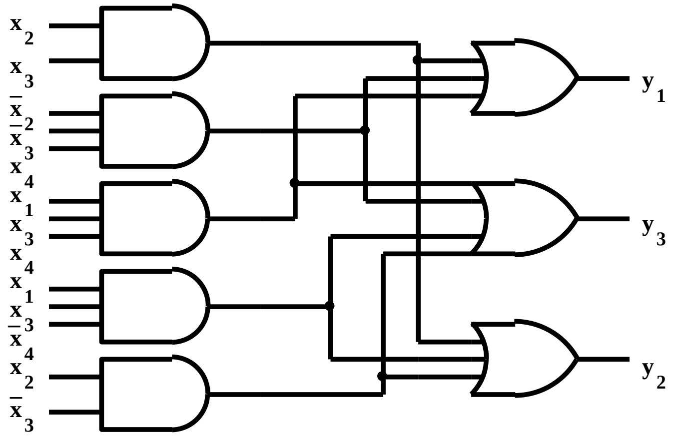


figura 3.14.2

Selettori e matrici

- Realizzano tutti i termini minimi
- Hanno (n ingressi e 2^n uscite)
- Sono anche detti “decodificatori” o “demultiplexer”
- Prendono il nome di “matrici” quando sono realizzati con particolari soluzioni circuitali (diodi)

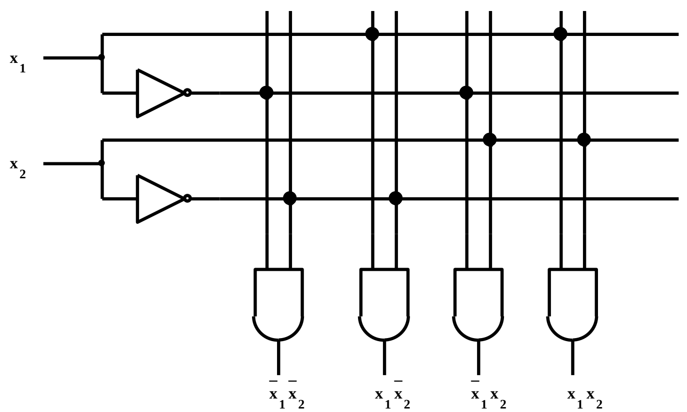


figura 3.15.1

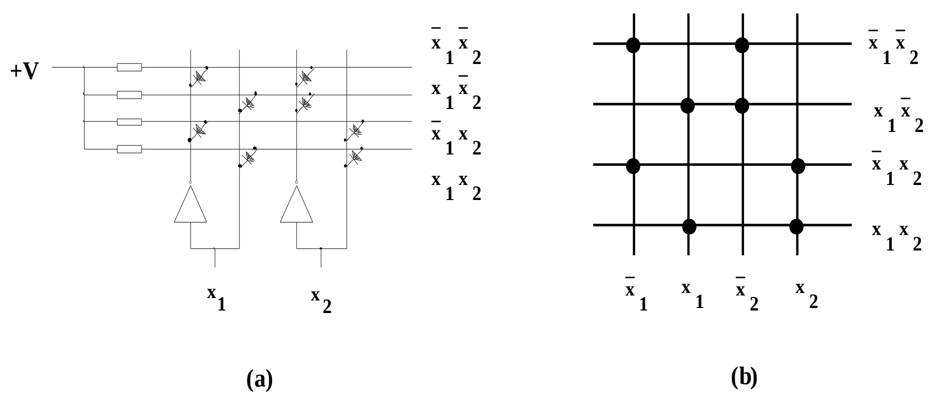


figura 3.15.2

Matrici semplici e multiple

- Le matrici semplici richiedono $n*2^n$ diodi
- Usando matrici multiple si possono ridurre i diodi impiegati
  - La matrice di variabile impiega $n*2^n$ diodi
  - La sottomatrice di combinazione $2*2^n$ (e' presente solo una linea attiva per il primo gruppo ed una sola per il secondo)

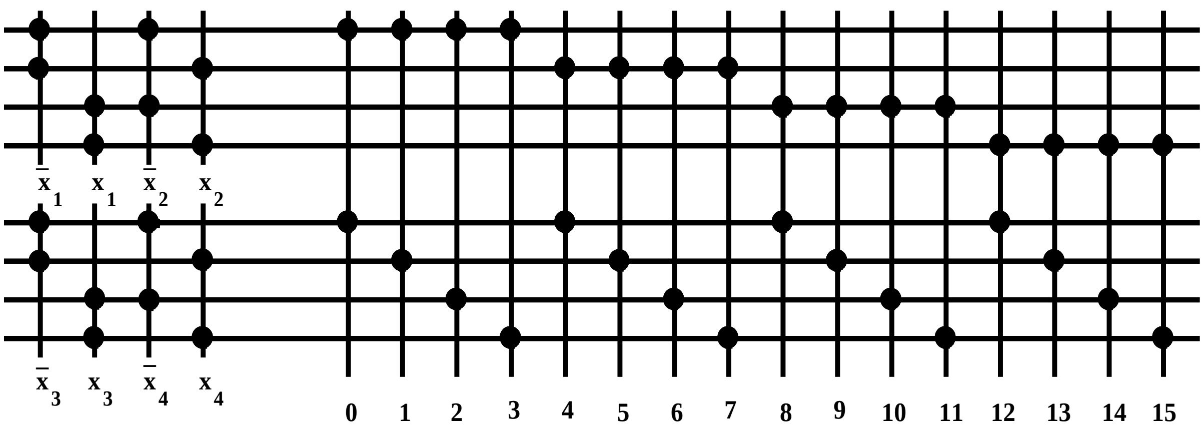


figura 3.15.3

Matrici semplici e multiple

- La suddivisione ottima viene fatta
  - ripartendo in parti quanto piu’ possibile uguali le variabili in sottomatrici (senza scendere sotto a 3)

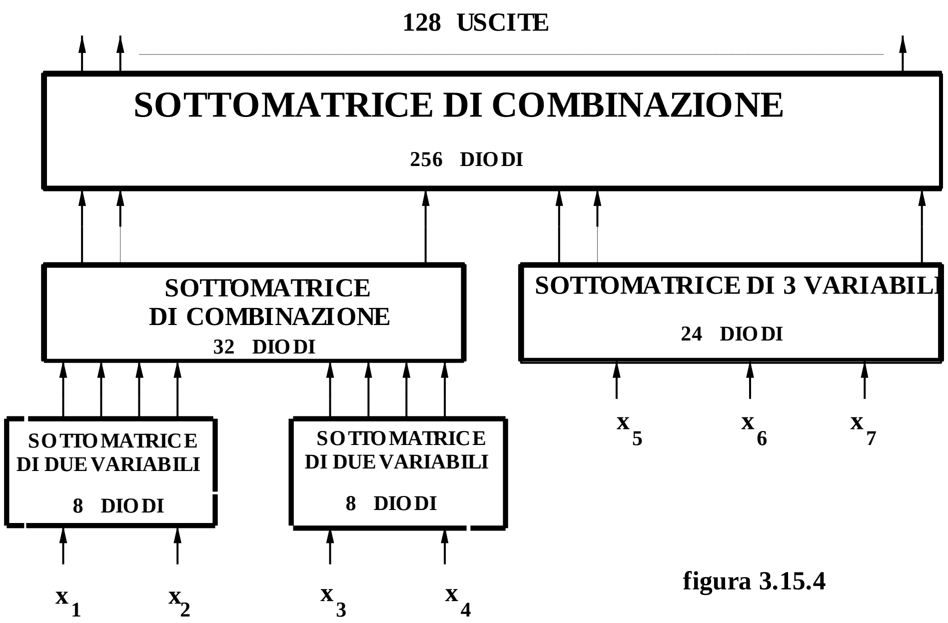


figura 3.15.4

Vengono usati 328 diodi al posto di 896

Matrici incomplete

- Se non compaiono alcuni termini minimi si puo' ottenere un certo riasparmio eliminando le corrispondenti linee di uscita (ed i relativi diodi), non solo sulla matrice d'uscita, ma anche in quelle che eventualmente precedono
- Esempio se non compare il termine $x_1 \overline{x}_2$ (e tutte le sue combianzioni) si possono risparmiare 10 diodi

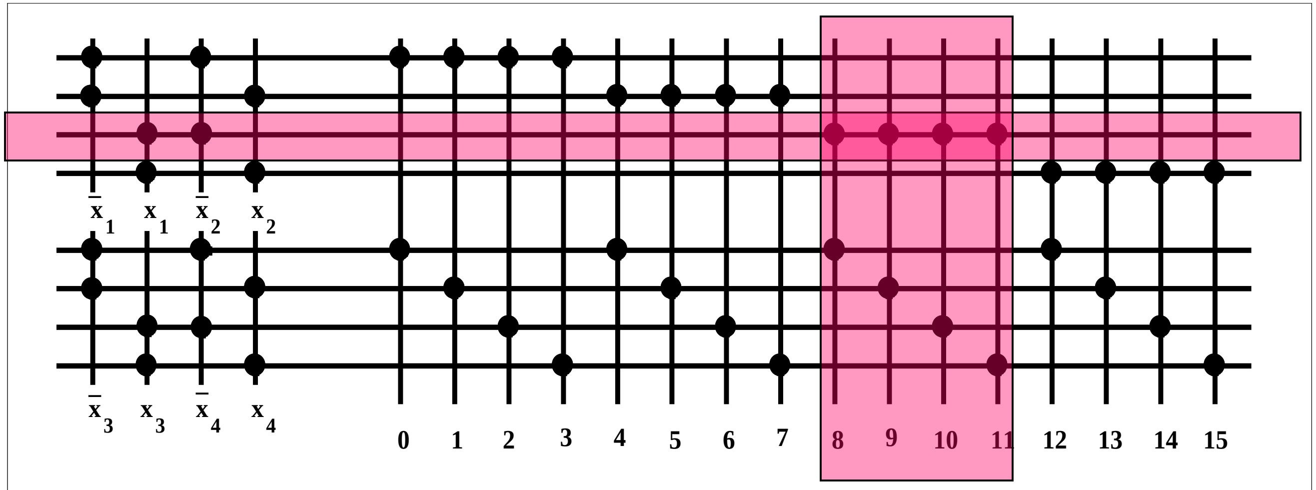


figura 3.15.3

Matrici incomplete

- Generalizzando:
- Esempio: il termine $y = \overline{x_1}.(x_2 . x_5 + x_3)$ non compaia
  - Caso 1
    - M_3 risparmia 2 linee e 6 diodi
    - M_5 risparmia 8+4-2=10 linee e 20 diodi
  - Caso 2
    - M_5 risparmia sempre 20 diodi
    - M_2 ne risparmia solo 2

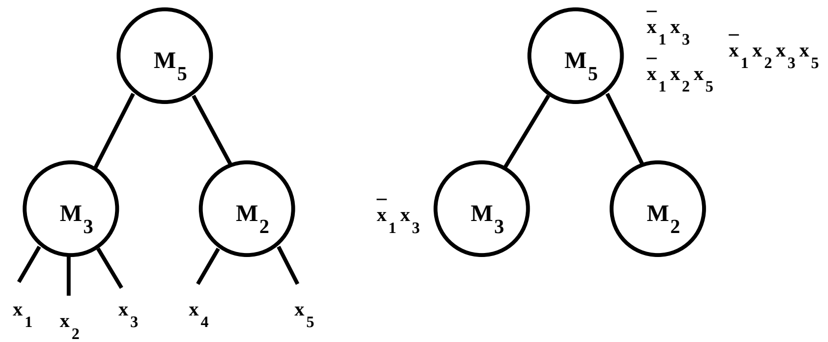


figura 3.15.5

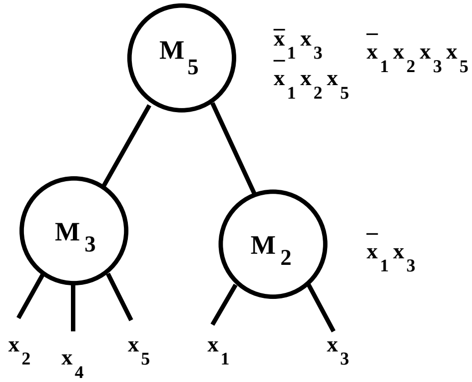


figura 3.15.6

I termini da eliminare facciano sentire la loro influenza il piu' possibile in prossimita' degli ingressi e compaiano nelle sottomatrici di ingresso a massimo numero di variabili.

Funzioni simmetriche

- Se le variabili di simmetria sono usate come ingressi di un sommatore binario
  - Le uscite realizzano le funzioni simmetriche corrispondenti a quell'insieme di variabili

A	B	C	D	S_1	S_2	S_3
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	0	0

$$S_1 = A.B.C.D = S_4^4(A, B, C, D)$$

$$\begin{aligned} S_2 &= \overline{A}.\overline{B}.C.D + \overline{A}.B.\overline{C}.D + \overline{A}.B.C.\overline{D} + \overline{A}.B.C.D + A.\overline{B}.\overline{C}.D + \\ &+ A.\overline{B}.C.\overline{D} + A.\overline{B}.C.D + A.B.\overline{C}.\overline{D} + A.B.\overline{C}.D + A.B.C.\overline{D} = \\ &= S_{(2,3)}^4(A, B, C, D) \end{aligned}$$

$$\begin{aligned} S_3 &= \overline{A}.\overline{B}.\overline{C}.D + \overline{A}.\overline{B}.C.\overline{D} + \overline{A}.B.\overline{C}.\overline{D} + \overline{A}.B.C.D + A.\overline{B}.\overline{C}.\overline{D} + \\ &+ A.\overline{B}.C.D + A.B.\overline{C}.D + A.B.C.\overline{D} = \\ &S_{(1,3)}^4(A, B, C, D) \end{aligned}$$

Funzioni simmetriche

- Usando un semplice sommatore si possono ottenere tutte le funzioni simmetriche

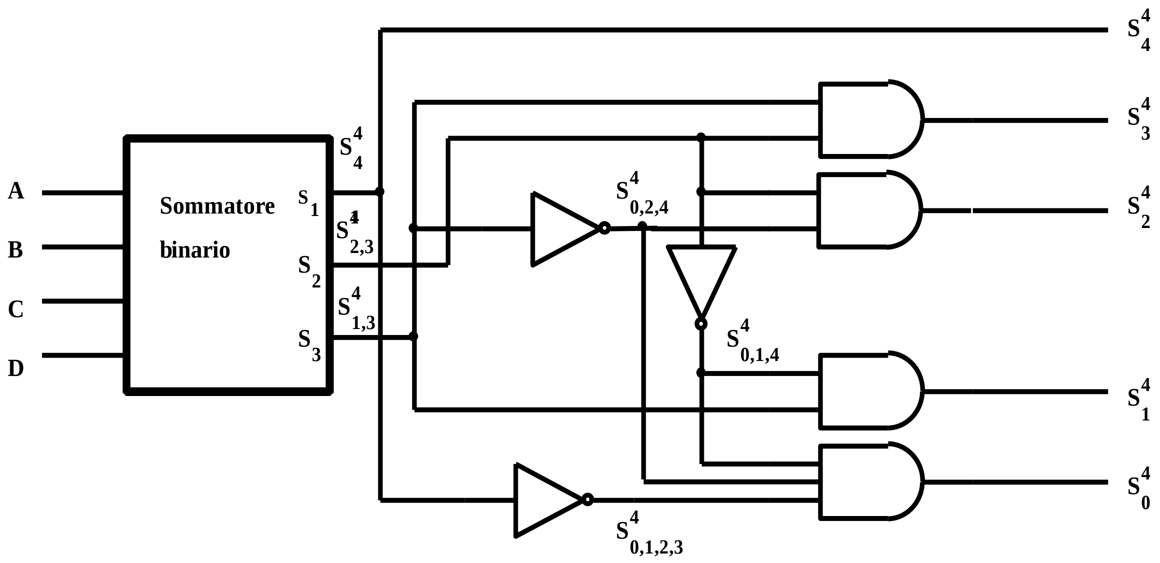
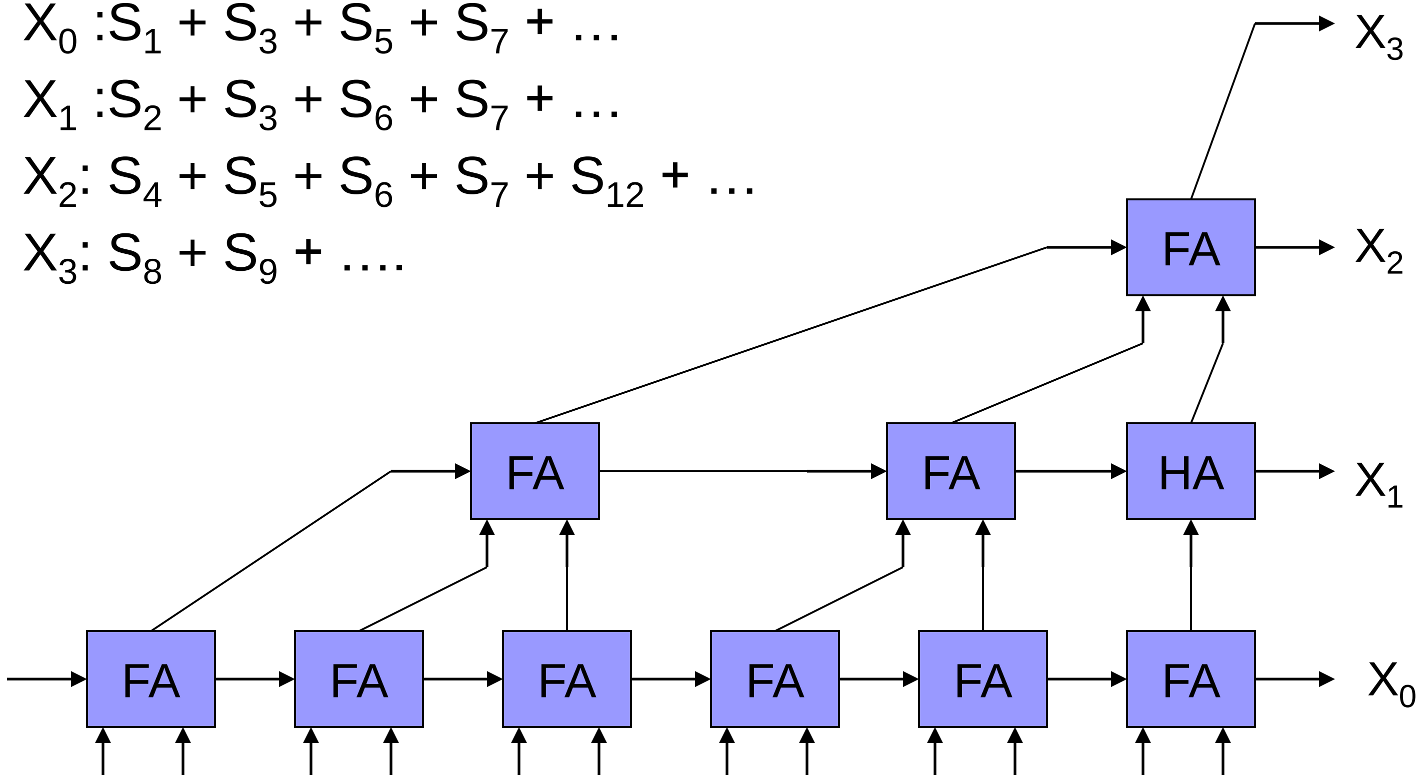


figura 3.16.1

Funzioni simmetriche

- Generalizzando: all'uscita di un sommatore binario (N.B: i bit in ingresso sono tutti di rango 1) escono:
  - X_0 :$S_1 + S_3 + S_5 + S_7 + \dots$
  - X_1 :$S_2 + S_3 + S_6 + S_7 + \dots$
  - X_2: $S_4 + S_5 + S_6 + S_7 + S_{12} + \dots$
  - X_3: $S_8 + S_9 + \dots.$



Funzioni simmetriche

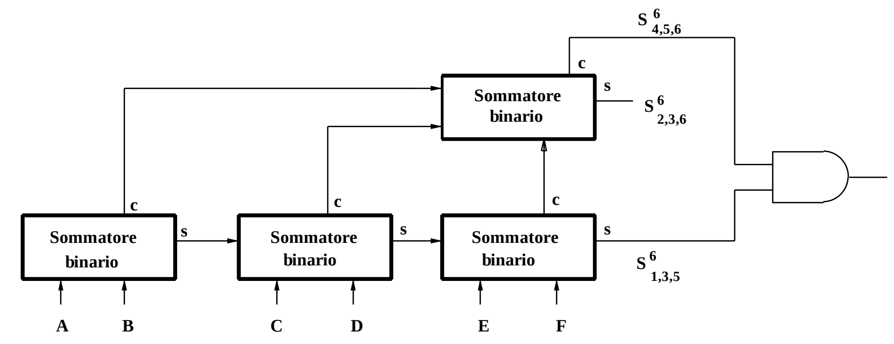


figura 3.16.2

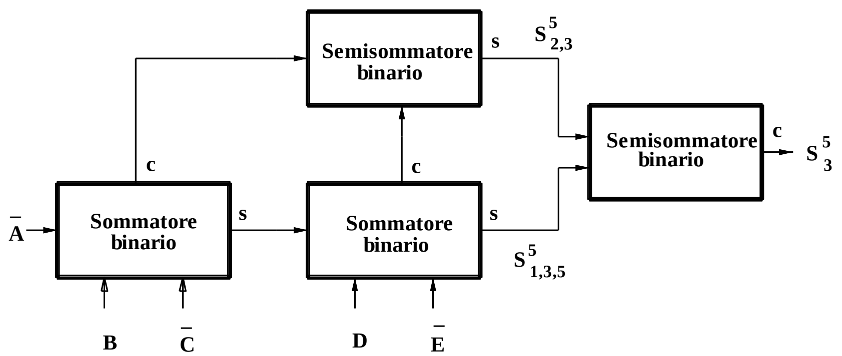


figura 3.16.3