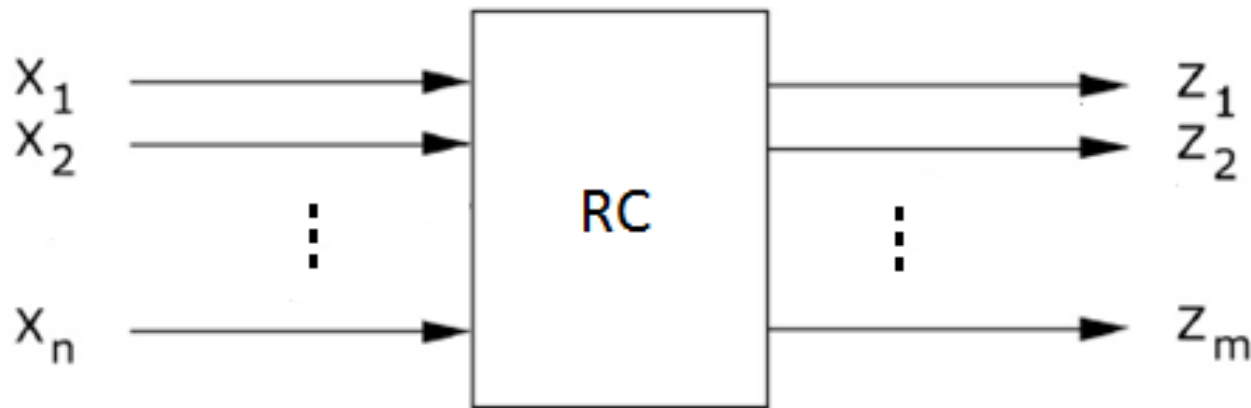




# Algebra booleana e Reti logiche

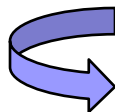
Reti combinatorie

## Rete combinatoria



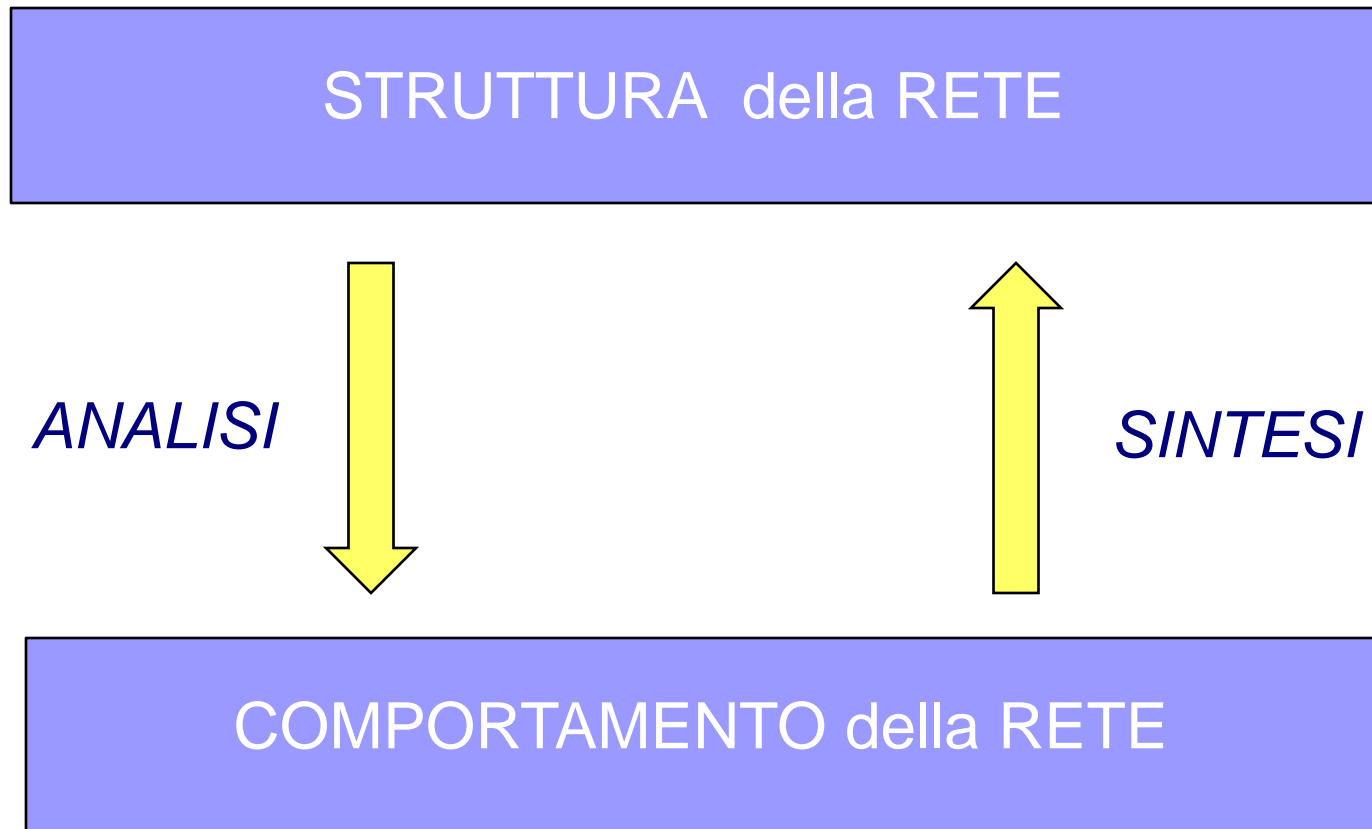
$input = \{x_1, x_2, \dots, x_n\}$

$output = \{z_1, z_2, \dots, z_m\}$



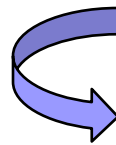
L'uscita è esclusivamente  
funzione dell'ingresso

## ***Analisi e sintesi***



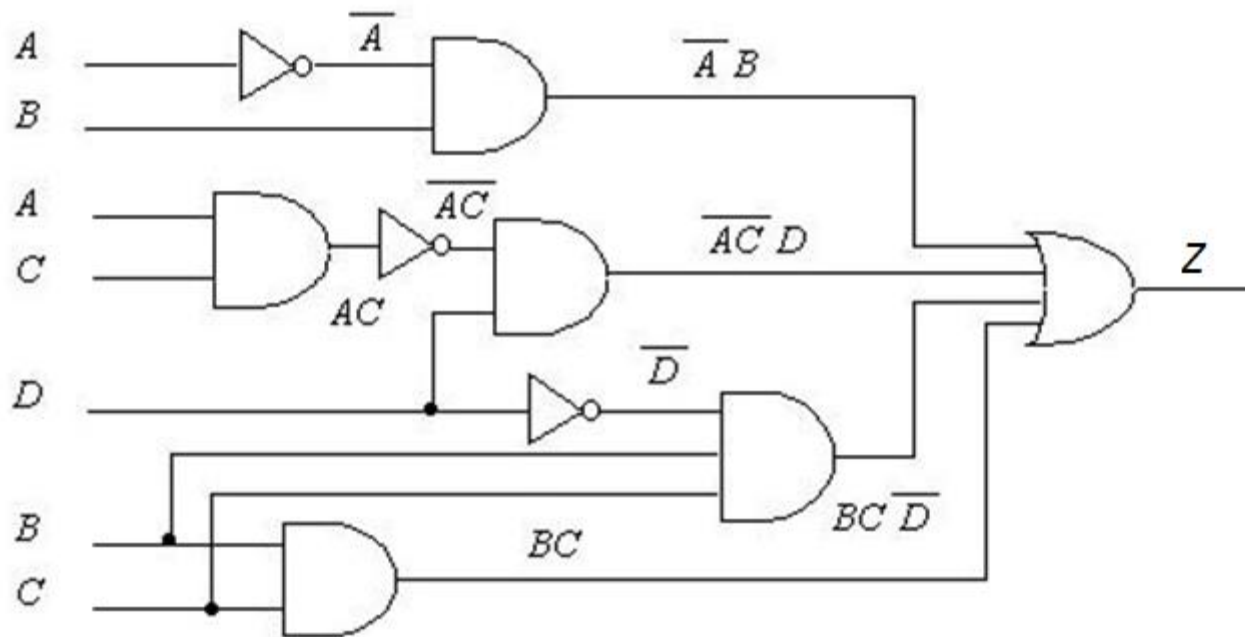
## ***Analisi della rete***

- Procedimento mediante il quale, dato lo schema logico della rete, se ne identifica il comportamento (per verificare se la rete è costruita in modo corretto o al fine di apportare modifiche)



*Dal circuito  
all'espressione  
booleana*

## Esempio



$$Z = \bar{A}B + \bar{A}\bar{C}D + BC + BC\bar{D}$$

## ***Sintesi della rete***

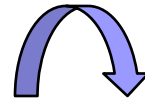
- Procedimento che conduce dalla specifica funzionale, ovvero dalla descrizione del funzionamento della rete, alla sua realizzazione circuitale:
  - *sintesi dalle tabelle di verità*
  - *scomposizione in sotto-reti*

## ***Sintesi dalle tabelle di verità***

- Si costruisce la tabella di verità che definisce la relazione tra le variabili di output e quelle di input (enumerando tutti i casi possibili).
- Per ogni variabile di output, si scrive la corrispondente espressione logica in forma canonica.
- Si semplificano le espressioni logiche (minimizzazione).
- A partire dalle espressioni logiche ottenute, si disegna lo schema circuitale.

## ***Scomposizione in sotto-reti***

- Una rete complessa può essere progettata attraverso la composizione di reti più semplici.
- In questo modo si rinuncia alla soluzione teoricamente ottima a favore di una maggiore comprensibilità e gestibilità del progetto.



Moduli combinatori  
di uso ricorrente



# Decodificatore

- Rete combinatoria con:

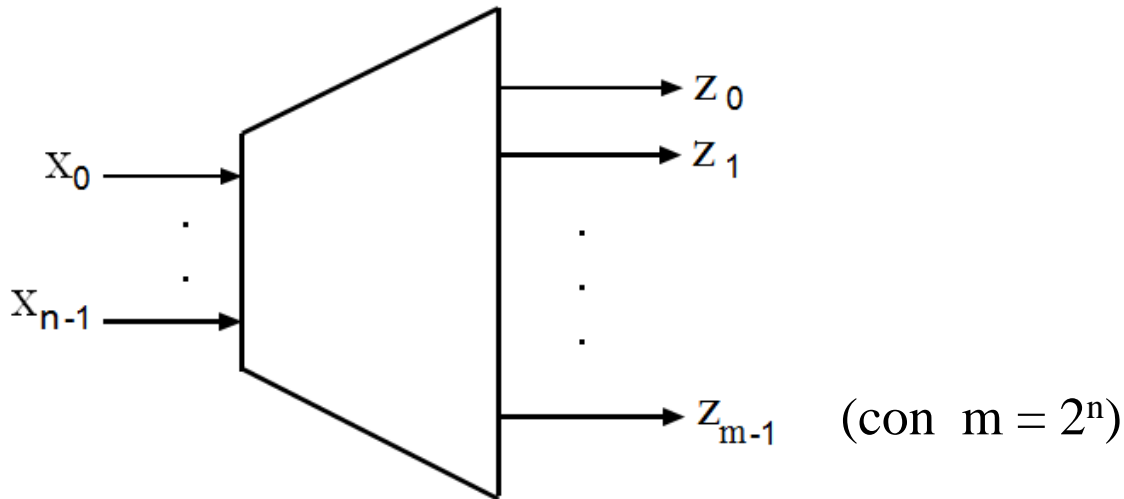
- ✓  $n$  ingressi

- ✓  $m = 2^n$  uscite



In corrispondenza a ogni  
ennupla di valori in input  
un solo terminale di uscita  
risulta attivato (assume cioè il valore 1)

# Decodificatore



L'insieme degli ingressi può essere interpretato come la codifica binaria di un numero  $i$  :

la corrispondente uscita  $z_i$   
è posta a 1, tutte le altre a 0

# Decodificatore

- Ciascuna uscita  $z_i$  ( $i = 0, \dots, m-1$ ) coincide con un prodotto fondamentale (*mintermine*):

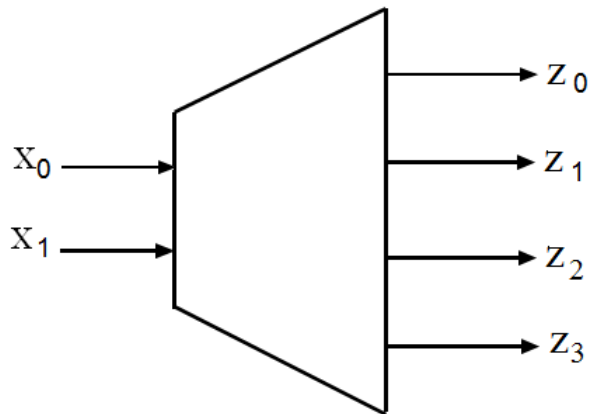
$$z_0 = \bar{x}_{n-1} \bar{x}_{n-2} \dots \bar{x}_1 \bar{x}_0$$

$$z_1 = \bar{x}_{n-1} \bar{x}_{n-2} \dots \bar{x}_1 x_0$$

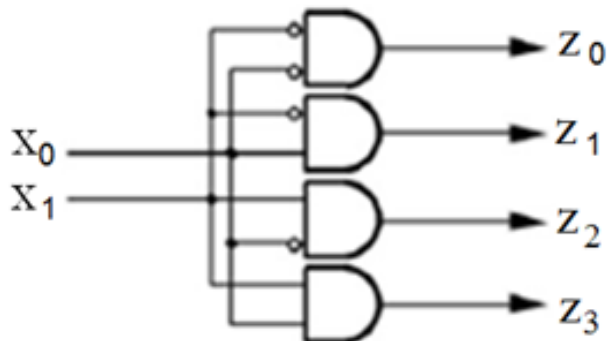
·  
·  
·

$$z_{m-1} = x_{n-1} x_{n-2} \dots x_1 x_0$$

## *Esempio di decodificatore ( $n = 2, m = 4$ )*



$x_1$	$x_0$	$z_3$	$z_2$	$z_1$	$z_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



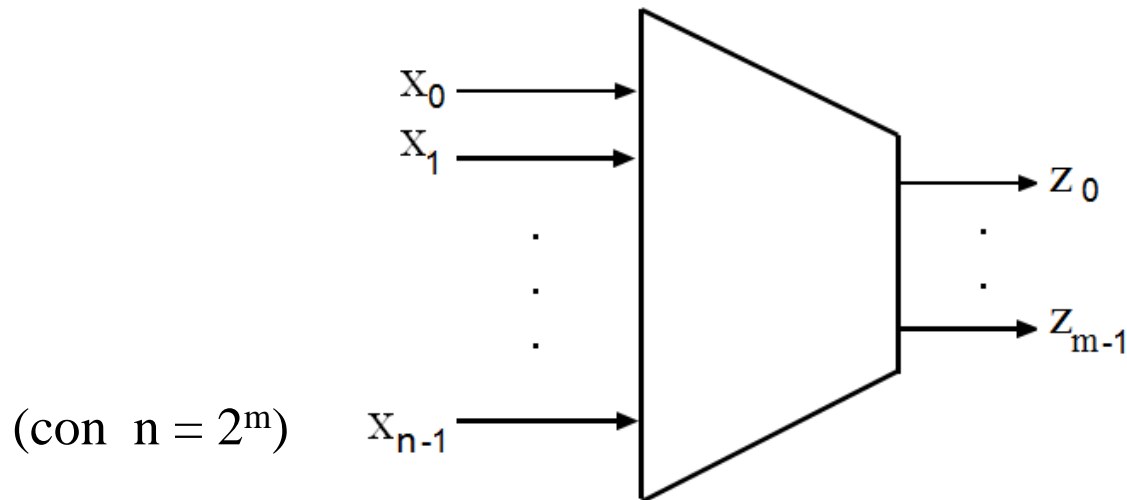
$$z_0 = \bar{x}_1 \bar{x}_0 \quad z_1 = \bar{x}_1 x_0$$

$$z_2 = x_1 \bar{x}_0 \quad z_3 = x_1 x_0$$

# Codificatore

- Svolge la funzione inversa di un decodificatore e prevede quindi:
  - ✓  $n = 2^m$  ingressi
  - ✓  $m$  uscite
- Il comportamento della rete è specificato soltanto per le configurazioni in ingresso che contengono un solo 1 (le altre danno luogo a condizioni di indifferenza).

# Codificatore

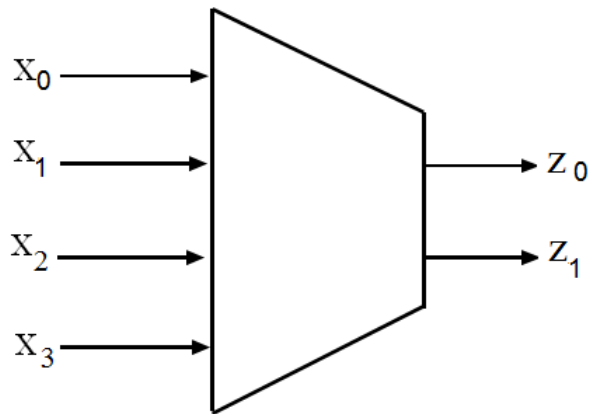


Data la configurazione di ingresso

$$x_i = 1 \text{ e } x_j = 0 \text{ (per ogni } j \neq i),$$

la corrispondente configurazione in uscita  $z_0, z_1, \dots, z_{m-1}$   
corrisponde alla codifica binaria del numero  $i$

## Esempio di codificatore ( $n = 4, m = 2$ )



$x_3$	$x_2$	$x_1$	$x_0$	$z_1$	$z_0$
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
<i>others</i>				-	-

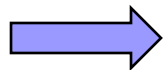
$$z_0 = x_1 + x_3 \quad z_1 = x_2 + x_3$$



In generale, le funzioni di uscita si semplificano sfruttando le molte condizioni di indifferenza presenti nella tabella di verità.

# Selettori

- *Selettore di ingresso o multiplexer :*  
permette di selezionare uno degli  $N$  ingressi  
e presentarlo sull'unica uscita
- *Selettore di uscita o demultiplexer :*  
permette di dirottare l'unico ingresso  
su una delle possibili  $N$  uscite



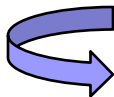
Linee di comando per la selezione



# Selettori

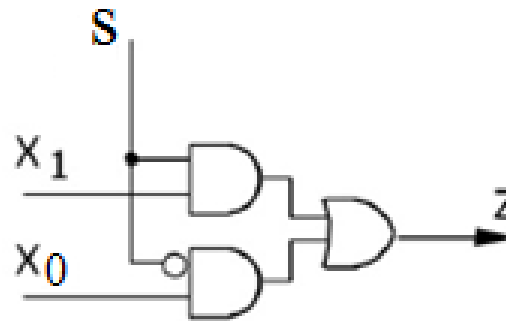
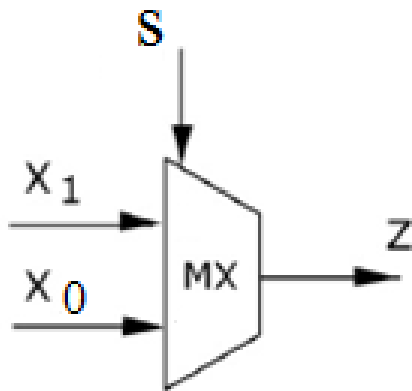
- Per la selezione dell'ingresso (uscita) è richiesto un numero  $n$  di linee di comando pari all'intero uguale o superiore a  $\log_2 N$ :

la configurazione delle  $n$  variabili di comando, interpretata secondo la codifica binaria, determina l'indice  $i$  dell'ingresso  $x_i$  (o dell'uscita  $z_i$ ) da selezionare

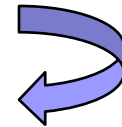


Di fatto, le variabili di comando possono essere considerate come ingressi aggiuntivi della rete.

## Selettore di ingresso a due vie

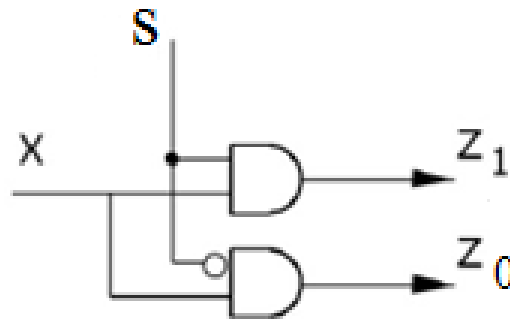
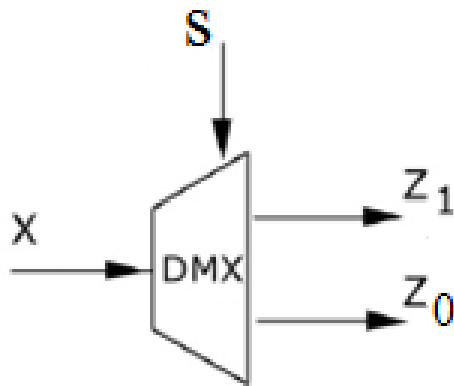


$$z = x_0 \bar{s} + x_1 s$$

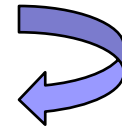


Il valore assunto dalla variabile di comando  $s$  determina quale tra gli ingressi,  $x_0$  e  $x_1$ , deve essere trasferito sull'uscita  $z$ .

## Selettore di uscita a due vie



$$z_0 = x\bar{S} \quad z_1 = xS$$



Il valore assunto dalla variabile di comando  $s$  determina su quale uscita,  $z_0$  o  $z_1$ , deve essere dirottato l'ingresso  $x$  (l'altra uscita è posta a 0).

## Costruzione di funzioni logiche tramite selettori

- Una qualunque funzione logica di  $n$  variabili può essere realizzata attraverso un selettore d'ingresso a  $N = 2^n$  vie.
- In generale, infatti, una funzione è esprimibile come somma di mintermini:

$$z = \sum_i (c_i \cdot C_i)$$

dove i coefficienti  $c_i$  valgono:

- ✓ 1 in corrispondenza dei mintermini  $C_i$  per i quali la funzione vale 1
- ✓ 0 altrimenti

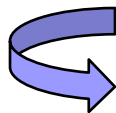
## Esempio

$x_1$	$x_2$	$x_3$	$z$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$\begin{aligned}z &= f(x_1, x_2, x_3) \\&= 1 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \\&\quad + 0 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \\&\quad + 0 \cdot \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \\&\quad + 1 \cdot \bar{x}_1 \cdot x_2 \cdot x_3 \\&\quad + 0 \cdot x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \\&\quad + 0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \\&\quad + 1 \cdot x_1 \cdot x_2 \cdot \bar{x}_3 \\&\quad + 1 \cdot x_1 \cdot x_2 \cdot x_3\end{aligned}$$

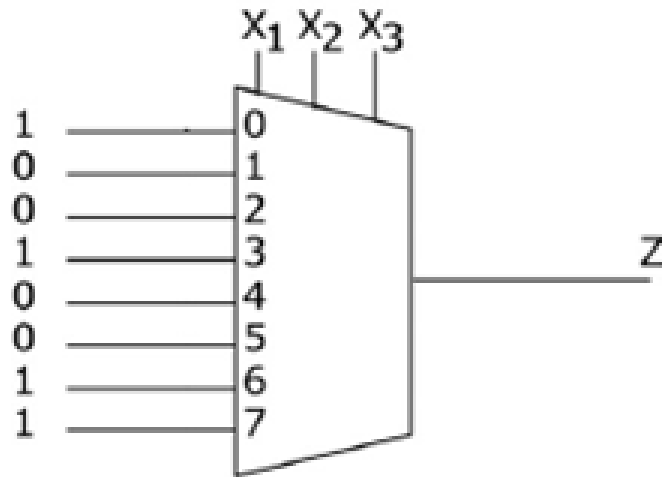
$$z = \sum_i (c_i \cdot C_i)$$

$$z = \sum (0, 3, 6, 7)$$



La funzione può essere realizzata  
con un selettore a otto vie

## Esempio



$$z = \sum_i (c_i \cdot C_i)$$

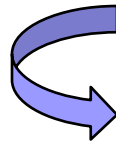
$$z = \sum (0,3,6,7)$$

- Le variabili  $x_1$ ,  $x_2$  e  $x_3$  sono impiegate come linee di selezione
- Le otto linee di ingresso sono poste a 0 o a 1 a seconda del corrispondente  $c_i$

# *Unità aritmetiche e logiche*

- Aritmetica binaria:

le tabelline delle operazioni  
contengono i due simboli 0 e 1

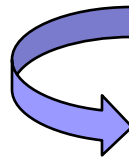


possono essere interpretate  
come le tabelle di verità  
delle funzioni logiche  
corrispondenti alle  
operazioni aritmetiche

## Somma di due bit

- Tabellina aritmetica della somma ( $S$ ) e del relativo riporto ( $R$ ):

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



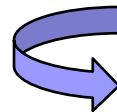
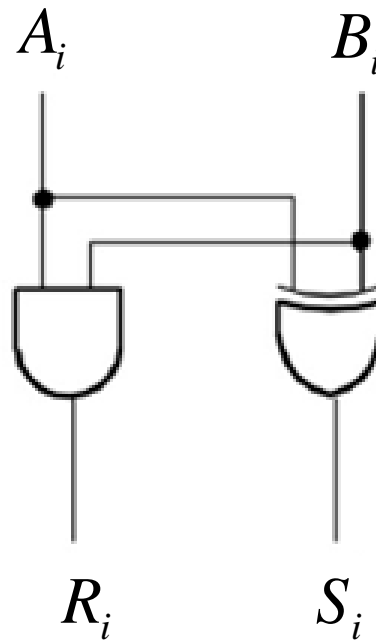
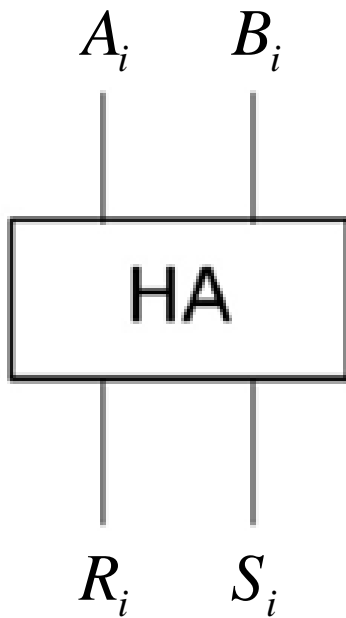
*Tabella di verità delle  
funzioni logiche:*

$$S_i(A_i, B_i)$$

$$R_i(A_i, B_i)$$



## Semisommatore (Half-Adder)



$$R_i = A_i B_i$$

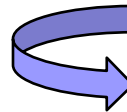
$$S_i = \bar{A}_i B_i + A_i \bar{B}_i = A_i \oplus B_i$$

## *Somma di due numeri interi*

- Consideriamo due numeri interi rappresentati in forma binaria attraverso due parole di  $n$  bit:

$$A = [A_{n-1} \dots A_0]$$

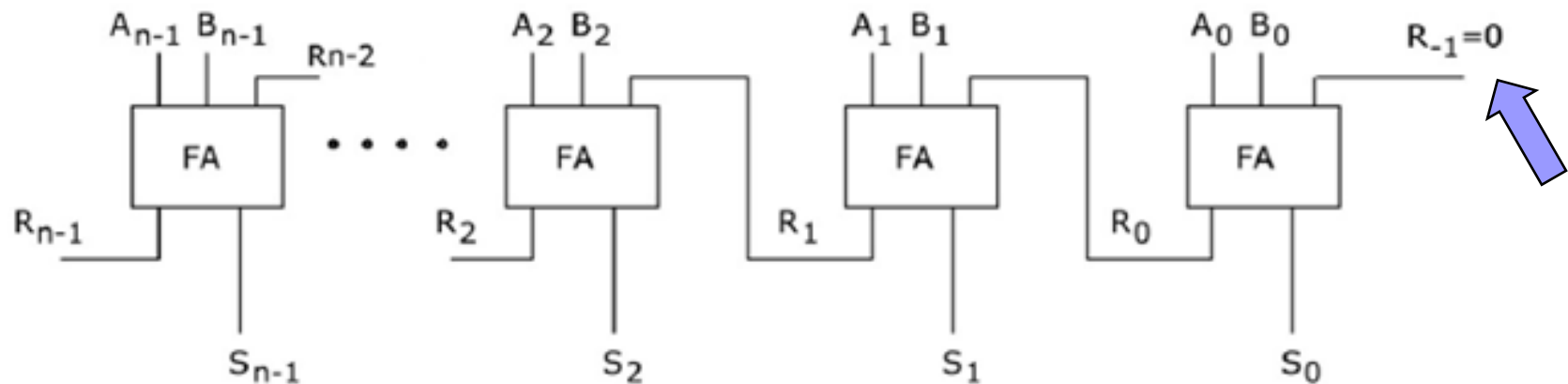
$$B = [B_{n-1} \dots B_0]$$



*Somma:*

$$S = [S_{n-1} \dots S_0]$$

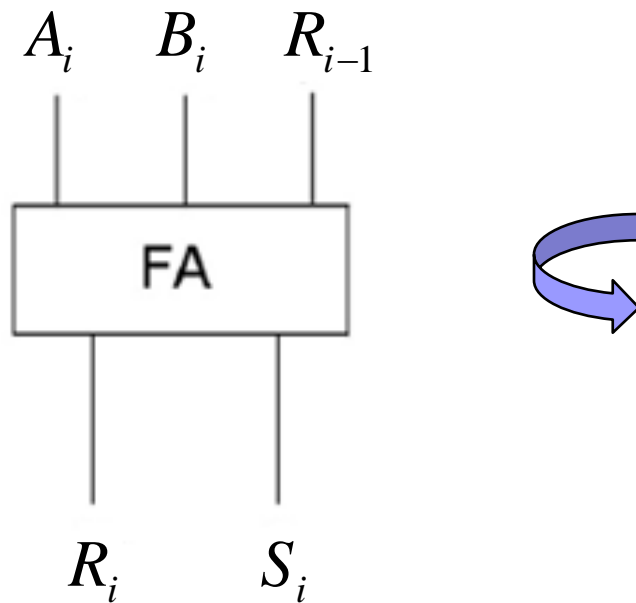
## Sommatore di parole di $n$ bit



La somma è effettuata a partire dai bit meno significativi procedendo verso sinistra e tenendo conto del riporto.

La cella elementare della rete è detta *Full Adder (FA)*.

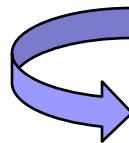
## Sommatore completo (Full Adder)



$A_i$	$B_i$	$R_{i-1}$	$S_i$	$R_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

## Sommatore completo (*Full Adder*)

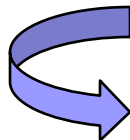
- La somma  $S_i$  dei tre bit  $A_i$ ,  $B_i$  e  $R_{i-1}$   
dà risultato 1 solo se è dispari  
il numero di bit a 1.



$$\begin{aligned} S_i &= A_i \oplus B_i \oplus R_{i-1} \\ &= (A_i \oplus B_i) \oplus R_{i-1} \end{aligned}$$

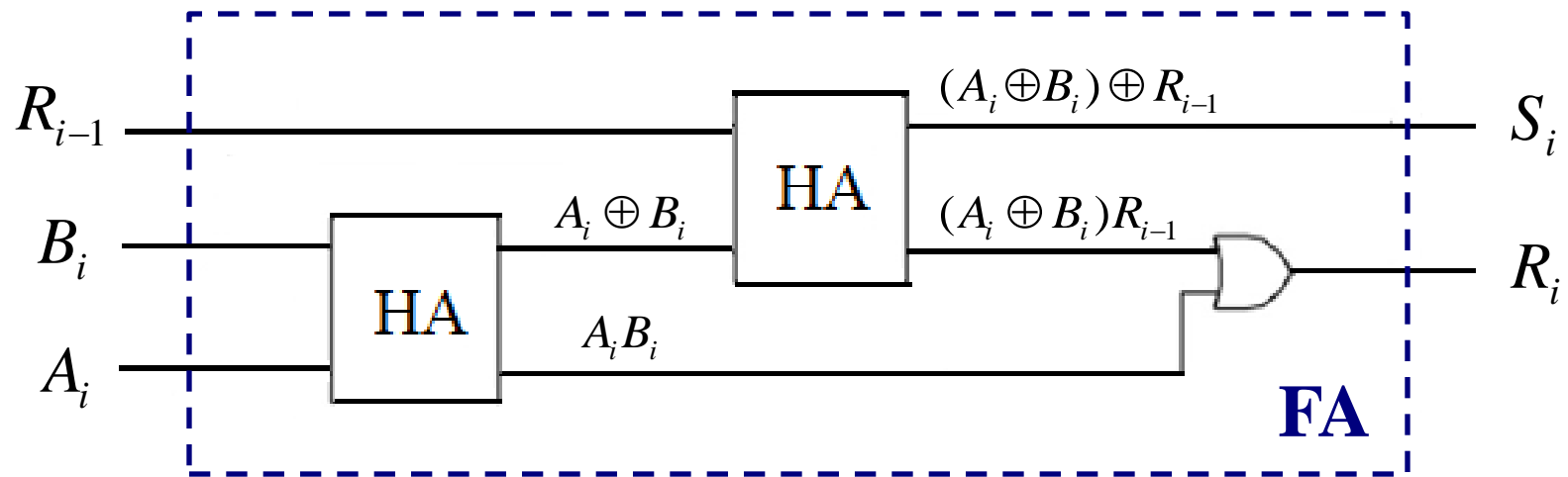
## Sommatore completo (*Full Adder*)

- Il riporto  $R_i$  vale 1 quando:
  - la somma di  $A_i$  e  $B_i$  dà direttamente riporto  
oppure
  - la somma di  $A_i$  e  $B_i$  vale 1,  
e si ha un riporto in ingresso di 1



$$R_i = A_i B_i + (A_i \oplus B_i) R_{i-1}$$

## Sommatore completo (Full Adder)

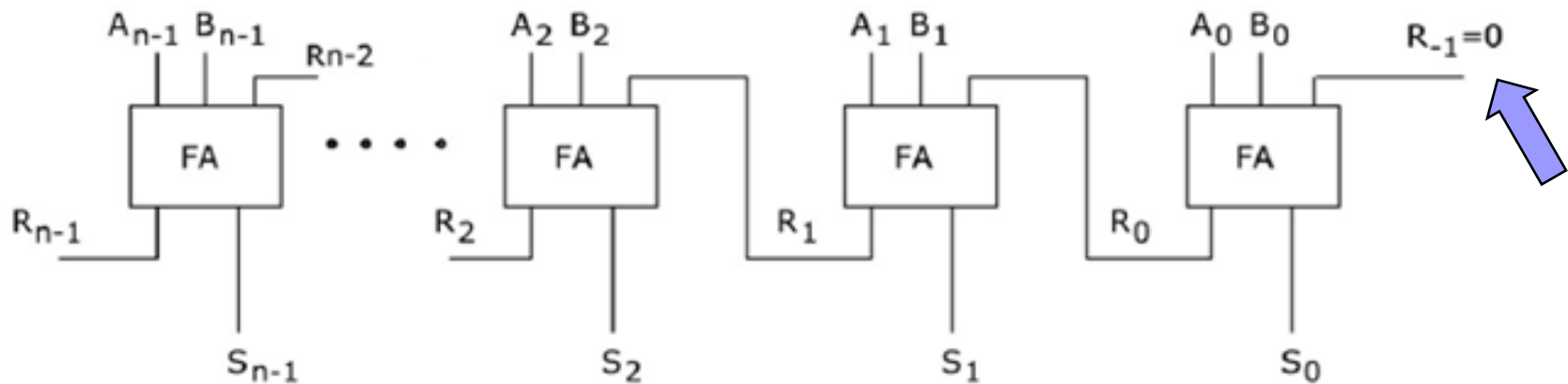


$$S_i = (A_i \oplus B_i) \oplus R_{i-1}$$

$$R_i = A_i B_i + (A_i \oplus B_i) R_{i-1}$$

## Sommatore completo (*Full Adder*)

- Come abbiamo visto in precedenza, a partire dal *Full Adder* (FA) è possibile realizzare un circuito sommatore:

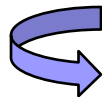




## ***Costruzione di un'unità aritmetica: esempio***

- Al sommatore possiamo aggiungere ulteriori elementi di logica in modo da realizzare un'unità aritmetica in grado di effettuare anche altre operazioni.
- In particolare, la sottrazione è riconducibile alla somma:

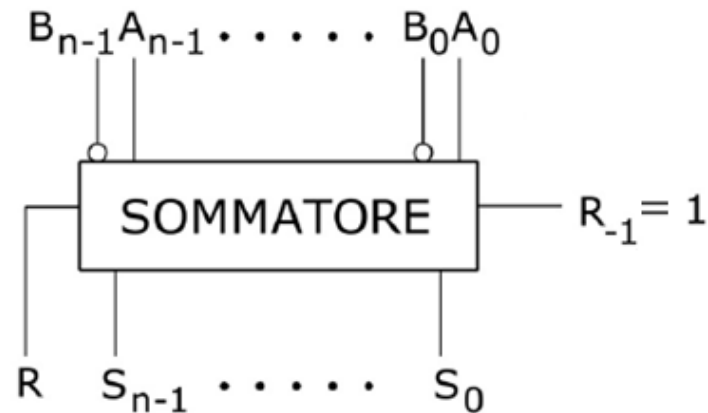
$$A - B = A + (-B)$$



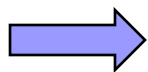
Notazione in complemento a 2 per i numeri negativi: si complementano tutti i bit di B e si aggiunge 1

## Costruzione di un'unità aritmetica: esempio

- Schema della rete che effettua la sottrazione usando un sommatore:

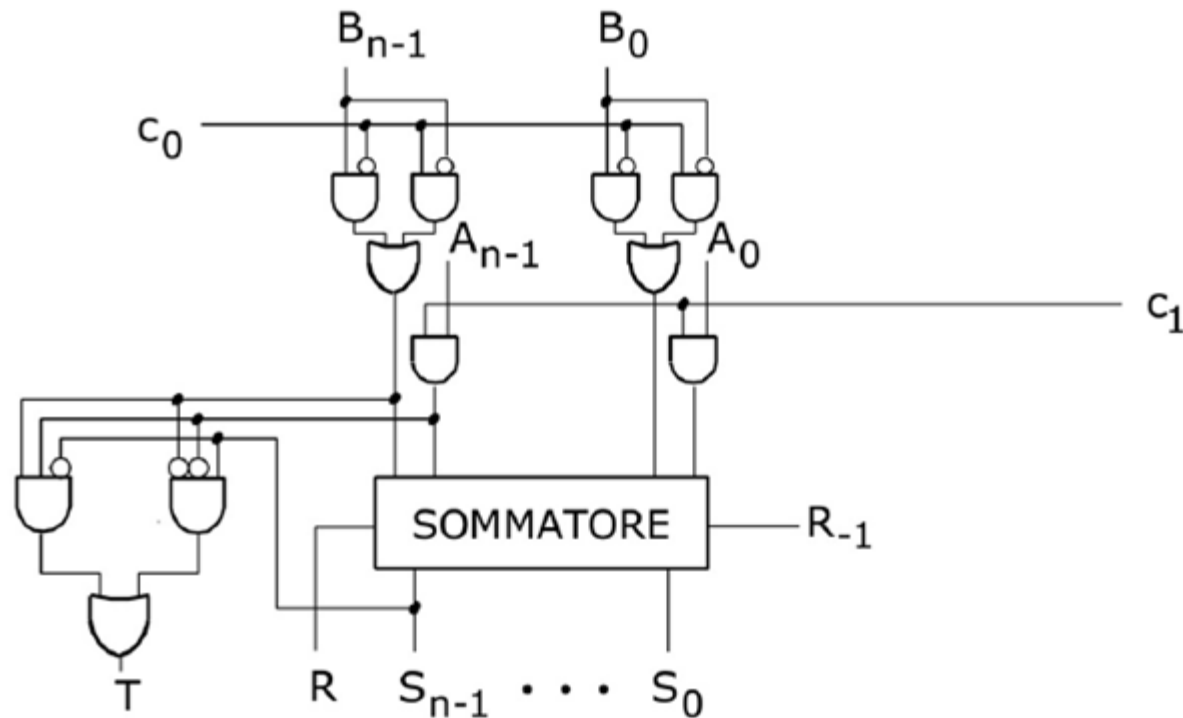


$$A + \bar{B} + 1$$



Prevedendo delle linee di controllo è possibile usare la stessa rete sia per le somme che per le sottrazioni

## Costruzione di un'unità aritmetica: esempio



La linea  $c_0$  seleziona  $B$  oppure il suo complemento, mentre la linea  $c_1$  seleziona come secondo ingresso  $A$  oppure  $0$ . In più, c'è la logica per il controllo del trabocco ( $T$ ).

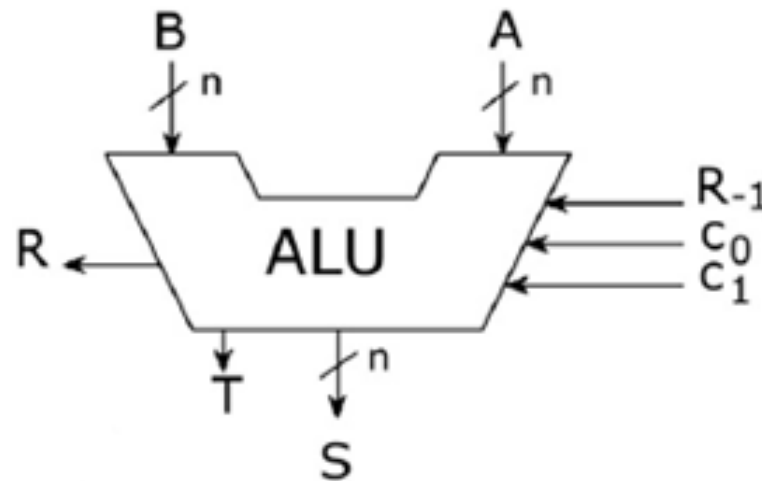
## Costruzione di un'unità aritmetica: esempio

- Comportamento della rete per ogni combinazione dei valori assunti dalle linee di controllo ( $c_1, c_0, R_{-1}$ ):

$c_1$	$c_0$	$R_{-1}$	Risultato	Commento
0	0	0	$S = 0 + B = B$	Selezione di B
0	0	1	$S = 0 + B + 1 = B + 1$	Incremento di B
0	1	0	$S = 0 + \overline{B} = \overline{B}$	Complementazione di B
0	1	1	$S = 0 + \overline{B} + 1 = -B$	Cambio segno di B
1	0	0	$S = A + B$	Somma $A + B$
1	0	1	$S = A + B + 1$	
1	1	0	$S = A + \overline{B} = A - B - 1$	
1	1	1	$S = A + \overline{B} + 1 = A - B$	Differenza $A - B$

## Costruzione di un'unità aritmetica: esempio

- Per quanto semplice, la rete precedente è un esempio di *Unità Aritmetica e Logica* (ALU):



(Figure parzialmente tratte da G. Bucci. *Calcolatori elettronici. Architettura e organizzazione*.  
Copyright © 2009 - The McGraw-Hill Companies)