Università di Salerno
Dipartimento di Ingegneria Industriale

UNIVERSITÀ DEGLI STUDI DI SALERNO

# Fondamenti di Informatica

## Algebra di Boole e Circuiti Logici

Prof. Arcangelo Castiglione

A.A. 2016/17

# L’Algebra di Boole – 1/3

- **Un po’ di storia**
  - Il matematico inglese George Boole nel 1847 fondò un campo della matematica e della filosofia chiamato **logica simbolica**
  - Shannon per primo applicò la logica simbolica ai circuiti nel 1939
- **L’algebra di Boole è caratterizzata da**
  - **Variabili booleane (o binarie):** variabili i cui valori possono essere 0 oppure 1
    - Ma anche: vero/falso, on/off, si/no
  - **Operazioni (o funzioni) booleane:** funzioni i cui input ed output sono variabili booleane

# L'Algebra di Boole – 2/3

- **Relazione con i circuiti logici**
  - Si studia l'algebra booleana poiché le sue funzioni sono isomorfe ai circuiti digitali
  - Un circuito digitale può essere espresso tramite un'espressione booleana e viceversa
    - Le variabili booleane corrispondono a segnali
    - Le funzioni booleane corrispondono a circuiti

# L’Algebra di Boole – 3/3

- Un **operando** può essere
  - Variabile booleana
  - Valore booleano (1 o 0)
- Sulle variabili ed i valori booleani si definiscono gli **operatori OR**, **AND** e **NOT**
  - Ed altri definiti a partire da essi
- Gli operatori **OR** e **AND** sono **operatori binari**: agiscono su **due** operandi
- L’operatore **NOT** è un **operatore unario**: agisce su **un solo** operando
- Nella valutazione delle espressioni booleane esiste una **relazione di precedenza fra gli operatori** **NOT**, **AND** e **OR**, nell’ordine in cui sono stati elencati
  - Per alterare tale relazione bisogna usare le parentesi

# Gli Operatori (o Funzioni)

- Gli **operatori (o funzioni)**
  - **OR**
  - **AND**
  - **NOT**

# Gli Operatori (o Funzioni)

- Gli **operatori (o funzioni)**
  - **OR**
  - **AND**
  - **NOT**

# OR – Somma Logica

- Il *risultato* dell'operatore (o funzione) **OR** è **1** se **almeno uno degli operandi** vale **1**. Il risultato è **0** negli altri casi

Tavola di verità

| $x_1$ | $x_2$ | $F(x_1, x_2) = x_1 + x_2$ |
|---|---|---|
| 0 | 0 | **0** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **1** |

- Date *n* variabili binarie, la loro somma logica (OR) è data da

$$x_1 + x_2 + \ldots + x_n = \begin{cases} \mathbf{1} \text{ se almeno una } x_i \text{ vale } \mathbf{1},\ con\ 1 \le i \le n \\ \mathbf{0} \text{ se } x_1 = x_2 = \ldots = x_n = \mathbf{0} \end{cases}$$

# OR: Possibili Rappresentazioni

- **x | y** **<- Usata in MATLAB**
- **or(x, y)** **<- Usata in MATLAB**
- **x # y**
- **x or y**
- **x + y**
- **x** $\cup \boldsymbol{y}$
- **x** $\vee \boldsymbol{y}$

# Gli Operatori (o Funzioni)

- Gli **operatori (o funzioni)**
  - **AND**
  - **OR**
  - **NOT**

# AND – Prodotto Logico

- Il *risultato* dell'operatore (o funzione) **AND** è **1** se il valore di **entrambi gli operandi è 1**. Il risultato è **0** negli altri casi

**Tavola di verità**

| $x_1$ | $x_2$ | $F(x_1, x_2) = x_1 \times x_2$ |
|---|---|---|
| 0 | 0 | **0** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **1** |

- Date *n* variabili binarie indipendenti, il loro prodotto logico (AND) è dato da

$$x_1 \times x_2 \times \ldots \times x_n = \begin{cases} \mathbf{0} \text{ se almeno una } x_i \text{ vale } \mathbf{0},\ con\ 1 \leq i \leq n \\ \mathbf{1} \text{ se } x_1 = x_2 = \ldots = x_n = \mathbf{1} \end{cases}$$

# AND: Possibili Rappresentazioni

- **x & y <- Usata in MATLAB**
- **and(x, y) <- Usata in MATLAB**
- **x and y**
- $\mathbf{x} \wedge \boldsymbol{y}$
- $\mathbf{x} \cap \boldsymbol{y}$
- $\mathbf{x} \times \boldsymbol{y}$
- **x * y**
- **xy**

# Gli Operatori (o Funzioni)

- Gli **operatori (o funzioni)**
  - **AND**
  - **OR**
  - **NOT**

# NOT – Negazione

- L'operatore (o funzione) **NOT**, inverte il valore dell'operando su cui opera, per questo talvolta viene detto *inverter*

| $x_1$ | $F(x_1) = \overline{x_1}$ |
|---|---|
| 0 | **1** |
| 1 | **0** |

- Doppia negazione

$$\overline{\overline{0}} = 0$$
$$\overline{\overline{1}} = 1$$

- L'elemento $\bar{x}$ viene detto complemento di x

# NOT: Possibili Rappresentazioni

- **y = ~x <- Usato in MATLAB**
- **y = not(x) <- Usato in MATLAB**
- **y = !x**
- **y = not x**
- **y = x’**
- **y =** $\neg x$
- **y =** $\overline{x}$

# Algebra di Boole: Alcune Identità

| Funzione AND | Funzione OR | Funzione NOT |
|---|---|---|
| $0 \times 0 = 0$ | $0 + 0 = 0$ | $x + \bar{x} = 1$ |
| $0 \times 1 = 0$ | $0 + 1 = 1$ | $x \times \bar{x} = 0$ |
| $1 \times 0 = 0$ | $1 + 0 = 1$ | $\bar{\bar{x}} = x$ |
| $1 \times 1 = 1$ | $1 + 1 = 1$ | |
| $x \times 0 = 0$ | $x + 0 = x$ | |
| $0 \times x = 0$ | $0 + x = x$ | |
| $x \times 1 = x$ | $x + 1 = 1$ | |
| $1 \times x = x$ | $1 + x = 1$ | |
| $x \times x = x$ | $x + x = x$ | |
| | | |

# Algebra di Boole: Alcune Identità

| Funzione AND | Funzione OR | Funzione NOT |
|---|---|---|
| $\mathbf{0 \times 0 = 0}$ | $0 + 0 = 0$ | $x + \bar{x} = 1$ |
| $\mathbf{0 \times 1 = 0}$ | $0 + 1 = 1$ | $x \times \bar{x} = 0$ |
| $\mathbf{1 \times 0 = 0}$ | $1 + 0 = 1$ | $\bar{\bar{x}} = x$ |
| $\mathbf{1 \times 1 = 1}$ | $1 + 1 = 1$ | |
| $x \times 0 = 0$ | $x$ | |
| $0 \times x = 0$ | $0 + x$ | |
| $x \times 1 = x$ | $x + 1$ | |
| $1 \times x = x$ | $1 + x$ | |
| $x \times x = x$ | $x + x = x$ | |
| | | |

**Prodotto Logico**

# Algebra di Boole: Alcune Identità

| Funzione AND | Funzione OR | Funzione NOT |
|---|---|---|
| $0 \times 0 = 0$ | $\boxed{\mathbf{0 + 0 = 0}}$ | $x + \bar{x} = 1$ |
| $0 \times 1 = 0$ | $\boxed{\mathbf{0 + 1 = 1}}$ | $x \times \bar{x} = 0$ |
| $1 \times 0 = 0$ | $\boxed{\mathbf{1 + 0 = 1}}$ | $\bar{\bar{x}} = x$ |
| $1 \times 1 = 1$ | $\boxed{\mathbf{1 + 1 = 1}}$ | |
| $x \times 0 = 0$ | $x + 0 = x$ | |
| $0 \times x = 0$ | $0 + x = x$ | |
| | $x + 1 = 1$ | |
| | $1 + x = 1$ | |
| | $x + x = x$ | |
| | | |

**Somma Logica**

**NOTA**: 1+1=1

# Algebra di Boole: Alcune Identità

| Funzione AND | Funzione OR | Funzione NOT |
|---|---|---|
| $0 \times 0 = 0$ | $0 + 0 = 0$ | $x + \bar{x} = 1$ |
| $0 \times 1 = 0$ | $0 + 1 = 1$ | $x \times \bar{x} = 0$ |
| $1 \times 0 = 0$ | $1 + 0 = 1$ | $\bar{\bar{x}} = x$ |
| $1 \times 1 = 1$ | $1 + 1 = 1$ | |
| $x \times 0 = 0$ | $x + 0 = x$ | |
| $0 \times x = 0$ | $0 + x = x$ | |
| $x \times 1 = x$ | $x + 1 = 1$ | |
| $1 \times x = x$ | $1 + x = 1$ | |
| $x \times x = x$ | $x + x = x$ | |
| | | |

**Legge dell'idempotenza**

# Algebra di Boole: Proprietà e Leggi

| | |
|---|---|
| **Proprietà Commutativa**<br>$x_1x_2 = x_2x_1$<br>$x_1 + x_2 = x_2 + x_1$ | **Leggi di Assorbimento**<br>$x_1 + x_1x_2 = x_1$<br>$x_1(x_1 + x_2) = x_1$ |
| **Proprietà Distributiva**<br>$x_1(x_2 + x_3) = x_1x_2 + x_1x_3$<br>$x_1 + (x_2x_3) = (x_1 + x_2) \times (x_1 + x_3)$ | **Leggi di De Morgan**<br>$\overline{x_1 + x_2} = \bar{x}_1 \times \bar{x}_2$<br>$\overline{x_1 \times x_2} = \bar{x}_1 + \bar{x}_2$ |
| **Proprietà Associativa**<br>$x_1(x_2x_3) = (x_1x_2)x_3$<br>$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$ | **Altre Note**<br>$x_1 + \bar{x}_1x_2 = x_1 + x_2$<br>$x_1(\bar{x}_1 + x_2) = x_1x_2$ |

# Leggi di De Morgan – 1/4

- Il complemento di una somma di variabili è uguale al prodotto dei complementi delle variabili
  - Il complemento di due o più variabili poste in OR è uguale all'AND dei complementi delle singole variabili

**Proprietà Commutativa**

$$x_1 x_2 = x_2 x_1$$
$$x_1 + x_2 = x_2 + x_1$$

**Leggi di Assorbimento**

$$x_1 + x_1 x_2 = x_1$$
$$x_1(x_1 + x_2) = x_1$$

**Proprietà Distributiva**

$$x_1(x_2 + x_3) = x_1 x_2 + x_1 x_3$$
$$x_1 + (x_2 x_3) = (x_1 + x_2) + (x_1 + x_3)$$

**Leggi di De Morgan**

$$\boxed{\overline{x_1 + x_2} = \overline{x_1} \times \overline{x_2}}$$
$$\overline{x_1 \times x_2} = \overline{x_1} + \overline{x_2}$$

**Proprietà Associativa**

$$x_1(x_2 x_3) = (x_1 x_2) x_3$$
$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$

**Altre Note**

$$x_1 + \overline{x_1} x_2 = x_1 + x_2$$
$$x_1(\overline{x_1} + x_2) = x_1 x_2$$

# Leggi di De Morgan – 2/4

- Il complemento di un prodotto di variabili è uguale alla somma dei complementi delle variabili
  - Il complemento di due o più variabili poste in AND è equivalente all'OR dei complementi delle singole variabili

**Proprietà Commutativa**

$$x_1x_2 = x_2x_1$$
$$x_1 + x_2 = x_2 + x_1$$

**Leggi di Assorbimento**

$$x_1 + x_1x_2 = x_1$$
$$x_1(x_1 + x_2) = x_1$$

**Proprietà Distributiva**

$$x_1(x_2 + x_3) = x_1x_2 + x_2x_3$$
$$x_1 + (x_2x_3) = (x_1 + x_2) + (x_1 + x_3)$$

**Leggi di De Morgan**

$$\overline{x_1 + x_2} = \overline{x_1} \times \overline{x_2}$$
$$\boxed{\overline{x_1 \times x_2} = \overline{x_1} + \overline{x_2}}$$

**Proprietà Associativa**

$$x_1(x_2x_3) = (x_1x_2)x_3$$
$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$

**Altre Note**

$$x_1 + \overline{x_1}x_2 = x_1 + x_2$$
$$x_1(\overline{x_1} + x_2) = x_1x_2$$

# Leggi di De Morgan – 3/4

- **Osservazione:** $\bar{\bar{x}} = x$ (Eq. 1)
- **Legge 1 di De Morgan:** $\overline{x_1 + x_2} = \overline{x_1} \times \overline{x_2}$ (Eq. 2)
- Utilizzando (Eq. 1) posso scrivere (Eq. 2) come segue: $\overline{\overline{x_1 + x_2}} = \overline{\overline{x_1} \times \overline{x_2}}$
- Utilizzando ancora (Eq. 1) ottengo che $x_1 + x_2 = \overline{\overline{x_1} \times \overline{x_2}}$
- L'OR fra $x_1$ e $x_2$ può essere espresso in termini delle sole operazioni AND e NOT
  - Ogni volta che in un'espressione booleana troviamo un OR, lo possiamo sostituire con la appropriata combinazione di AND e NOT
    - Ogni espressione può essere espressa in termini delle sole due operazioni logiche AND e NOT

# Leggi di De Morgan – 4/4

- **Osservazione:** $\bar{\bar{x}} = x$ (Eq. 1)
- **Legge 2 di De Morgan:** $\overline{x_1 \times x_2} = \overline{x_1} + \overline{x_2}$ (Eq. 3)
- Utilizzando (Eq. 1) posso scrivere (Eq. 3) come segue: $\overline{\overline{x_1 \times x_2}} = \overline{\overline{x_1} + \overline{x_2}}$
- Utilizzando ancora (Eq. 1) ottengo che $x_1 \times x_2 = \overline{\overline{x_1} + \overline{x_2}}$
- L'AND fra $x_1$ e $x_2$ può essere espresso in termini delle sole operazioni OR e NOT
  - Ogni volta che in un'espressione booleana troviamo un AND, lo possiamo sostituire con la appropriata combinazione di OR e NOT
    - Ogni espressione può essere espressa in termini delle sole due operazioni logiche OR e NOT

# Alcune Osservazioni

- Identità, proprietà e leggi dell'algebra booleana sono generalmente applicate nelle trasformazioni di funzioni booleane in altre equivalenti, ma di più facile realizzazione circuitale
- Dalle leggi di **De Morgan** si evince che la scelta delle funzioni OR, AND e NOT, come funzioni primitive, è ridondante

# Funzioni Logiche (o Booleane) – 1/5

- Date ***n* variabili** booleane indipendenti ***$x_1$, $x_2$,…, $x_n$***, queste possono assumere **$2^n$ configurazioni** distinte
  - Ad esempio, per *n = 3* si hanno *8* configurazioni
- **Configurazione:** AND di tutte le variabili, dove quelle corrispondenti ai valori 0 compaiono negate
  - **Prodotto fondamentale o prodotto minimo (minterm)**
- Ogni riga (in rosso) mostra il valore restituito a partire da una particolare configurazione dell'input

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

$$010 \longleftrightarrow \overline{x}_1 x_2 \overline{x}_3$$

# Funzioni Logiche (o Booleane) – 2/5

| Configurazioni | $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|---|
| $\overline{x_1}\,\overline{x_2}\,\overline{x_3}$ | 0 | 0 | 0 | **0** |
| $\overline{x_1}\,\overline{x_2}\,x_3$ | 0 | 0 | 1 | **0** |
| $\overline{x_1}\,x_2\,\overline{x_3}$ | 0 | 1 | 0 | **0** |
| $\overline{x_1}\,x_2\,x_3$ | 0 | 1 | 1 | **1** |
| $x_1\,\overline{x_2}\,\overline{x_3}$ | 1 | 0 | 0 | **0** |
| $x_1\,\overline{x_2}\,x_3$ | 1 | 0 | 1 | **1** |
| $x_1\,x_2\,\overline{x_3}$ | 1 | 1 | 0 | **1** |
| $x_1\,x_2\,x_3$ | 1 | 1 | 1 | **1** |

- 011 indica tra le $2^3$=8 configurazioni possibili, quella in cui
  - $x_1$ = 0
  - $x_2$ = 1
  - $x_3$ = 1
- Questa configurazione si scrive semplicemente con il prodotto $\overline{x_1}\,x_2\,x_3$

# Funzioni Logiche (o Booleane) – 3/5

- Una variabile $y$ è **funzione** delle $n$ variabili indipendenti $x_1, x_2, \ldots, x_n$, se esiste un criterio che fa corrispondere in modo univoco ad ognuna delle $2^n$ configurazioni delle variabili $x_i$ $(i=1,\ldots,n)$ un valore di $y$ (ovviamente 0 oppure 1)

$$y = F(x_1, x_2, \ldots, x_n)$$

- Una rappresentazione esplicita di una funzione è la **tavola (o tabella) di verità**, in cui si elencano tutte le possibili combinazioni di $x_1$, $x_2$, …, $x_n$, con associato il valore di $y$

$$\boxed{y = x_1 + x_2} \Rightarrow$$

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 0 | 0 | **0** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **1** |

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

$\overline{x_1}\, x_2\, x_3$

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

$\overline{x_1}\, x_2\, x_3$ $\quad$ $x_1\, \overline{x_2}\, x_3$

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

$\overline{x_1}\, x_2\, x_3$ $\quad x_1\, \overline{x_2}\, x_3$ $\quad x_1\, x_2\, \overline{x_3}$ $\quad x_1\, x_2\, x_3$

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

$\overline{x_1}\, x_2\, x_3 \qquad x_1\, \overline{x_2}\, x_3 \qquad x_1\, x_2\, \overline{x_3} \qquad x_1\, x_2\, x_3$

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

$$\overline{x_1}\, x_2\, x_3 + x_1\, \overline{x_2}\, x_3 + x_1\, x_2\, \overline{x_3} + x_1\, x_2\, x_3$$

# Funzioni Logiche (o Booleane) – 4/5

- Oltre che mediante tavola di verità, ogni **funzione booleana** può essere rappresentata tramite la sua **espressione booleana (forma canonica)**

- Per passare dalla rappresentazione mediante **tavola di verità** alla notazione tramite **espressione booleana** è necessario
  1. **Identificare** tutte le righe della tavola di verità che danno 1 in output
  2. Per ogni riga con un 1 in output, **scrivere il *minterm* della configurazione delle variabili** che la definiscono
  3. **Collegare tramite OR** tutti i minterm ottenuti

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

$F(x_1, x_2, x_3)$

$$\overline{x_1}\, x_2\, x_3 + x_1\, \overline{x_2}\, x_3 + x_1\, x_2\, \overline{x_3} + x_1\, x_2\, x_3$$

# Funzioni Logiche (o Booleane) – 5/5

- $F(x_1, x_2, x_3) = \overline{x_1}\, x_2\, x_3 + x_1\, \overline{x_2}\, x_3 + x_1\, x_2\, \overline{x_3} + x_1\, x_2\, x_3$

- Usando i minterm possiamo determinare l'espressione booleana di una funzione booleana a partire dalla tavola di verità
- L'espressione booleana trovata si chiama **forma canonica** della funzione
- Tutte le funzioni logiche possono essere riportate in forma canonica
- Se un minterm assume valore 1 anche la funzione $F$ assume il valore 1

| $x_1$ | $x_2$ | $x_3$ | $F(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **1** |

# Esempio 1: dalla Tavola di Verità alla Funzione

| $x$ | $y$ | $z$ | $F$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **0** |

- **Problema:** date tre variabili booleane $(x, y, z)$, si scriva la funzione $F$ che vale 1 quando solo due di esse hanno valore 1

**Si può scrivere la funzione come somma logica (OR) delle configurazioni corrispondenti agli 1**

$$F(x, y, z) = \bar{x}yz + x\bar{y}z + xy\bar{z}$$

Forma canonica: somma di prodotti (OR di AND)

**N.B. tutte le funzioni logiche si possono scrivere in questa forma**

# Esempio 2: dalla Tavola di Verità alla Funzione

| $x$ | $y$ | $z$ | $F$ |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **1** |
| 0 | 1 | 0 | **1** |
| 0 | 1 | 1 | **0** |
| 1 | 0 | 0 | **1** |
| 1 | 0 | 1 | **0** |
| 1 | 1 | 0 | **0** |
| 1 | 1 | 1 | **1** |

- **Problema:** date tre variabili booleane $(x, y, z)$, si scriva la funzione $F$ che vale 1 quando il numero di 1 è dispari

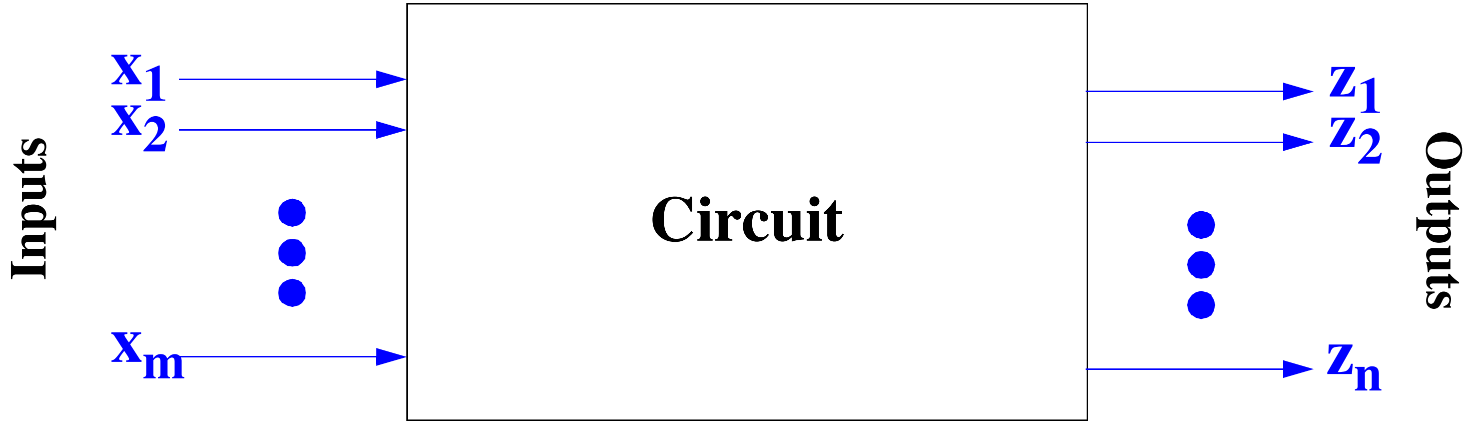
**Si può scrivere la funzione come somma logica (OR) delle configurazioni corrispondenti agli 1**

$$F(x, y, z) = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$

Forma canonica: somma di prodotti (OR di AND)

**N.B. tutte le funzioni logiche si possono scrivere in questa forma**

# Circuito Logico




- Il **cuore** di un **sistema digitale** è il **circuito logico** digitale
  - Progettato a partire da **porte logiche**
  - **Porte** collegate tra loro per formare **circuiti**
  - **Circuiti** che combinati tra loro rivestono grande importanza pratica nell'**architettura** del **computer**

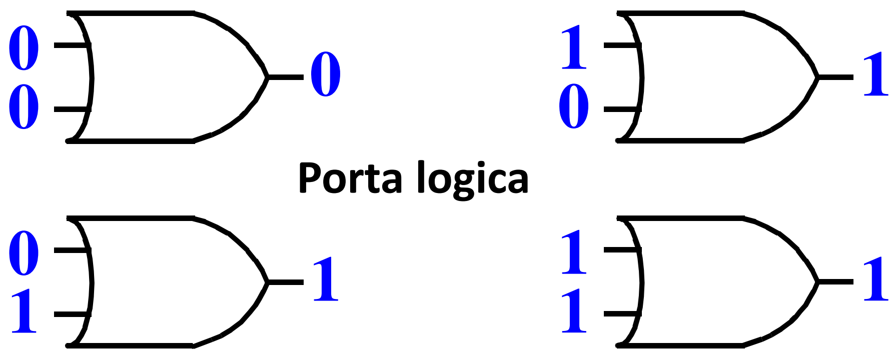
# Porte Logiche

- Elementi base utilizzati per creare circuiti logici
  - Qualsiasi circuito può essere implementato usando solo porte logiche
    - AND, OR e NOT
- Dispositivi elettronici che implementano semplici funzioni booleane
- Ciascuna porta ha il proprio **simbolo logico** che permette a funzioni complesse di essere rappresentate mediante un diagramma logico
- La **funzione di ciascuna porta** può essere rappresentata da una **tavola di verità o** utilizzando la **notazione booleana**

# Funzione OR: Tavola di Verità e Porta Logica

Tavola di verità

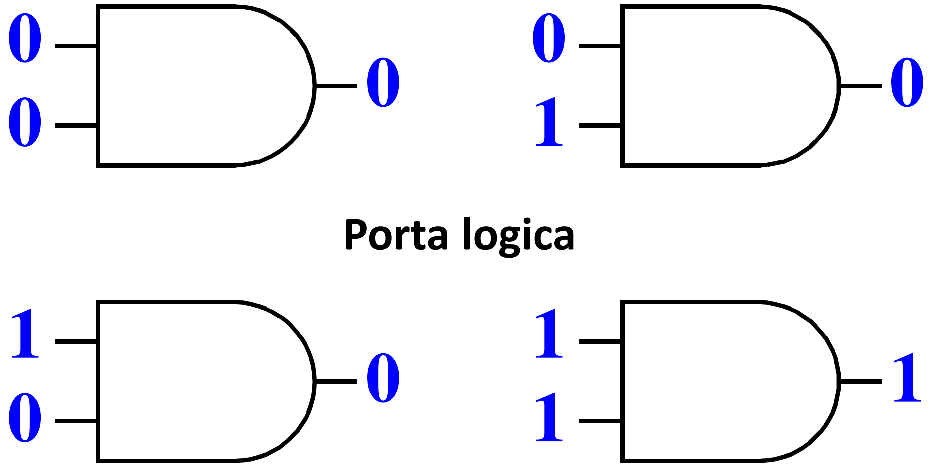
| $x_1$ | $x_2$ | $x_1\ OR\ x_2$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



# Funzione AND: Tavola di Verità e Porta Logica

Tavola di verità

| $x_1$ | $x_2$ | $x_1\ AND\ x_2$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

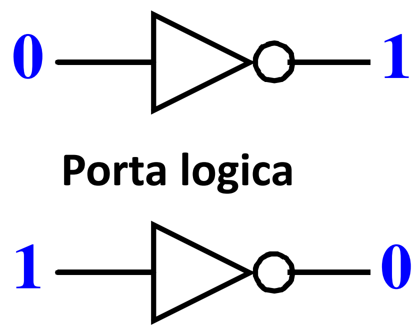



**Porta logica**

# Funzione NOT: Tavola di Verità e Porta Logica

Tavola di verità

| $x$ | $NOT\ x$ |
|---|---|
| 0 | 1 |
| 1 | 1 |



# Altre Porte Logiche: Porta NAND

**NAND = NOT AND**


(a) Circuit symbol

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

(b) Truth table

$$C = \overline{A \cdot B}$$

(c) Boolean expression

# Altre Porte Logiche: Porta NOR

**NOR = NOT OR**


(a) Circuit symbol

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

(b) Truth table

$$C = \overline{A + B}$$

(c) Boolean expression

# Altre Porte Logiche: Porta XOR


(a) Circuit symbol

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

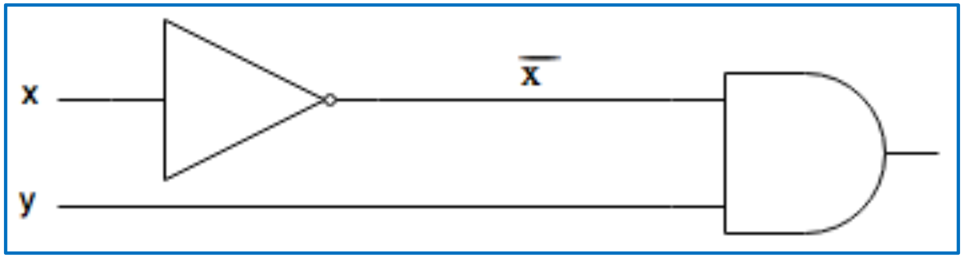
(b) Truth table

$$C = A \oplus B$$

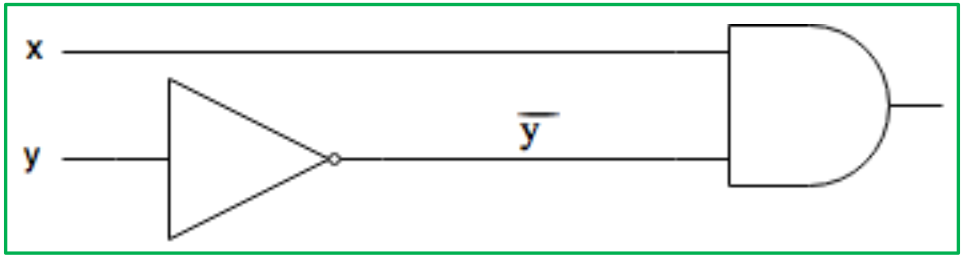
(c) Boolean expression

# Esempio 3: dalla Funzione al Circuito

- Circuito per la funzione $\bar{x}$y

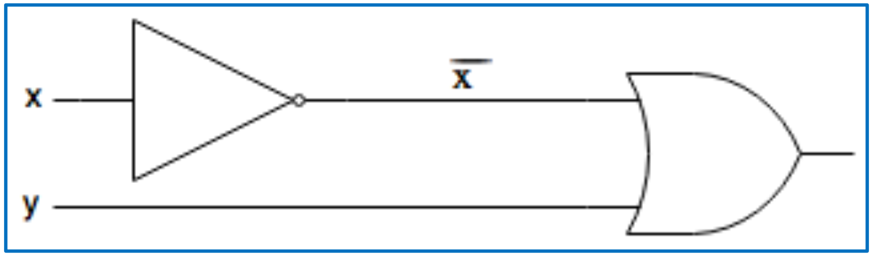



- Circuito per la funzione $x\bar{y}$

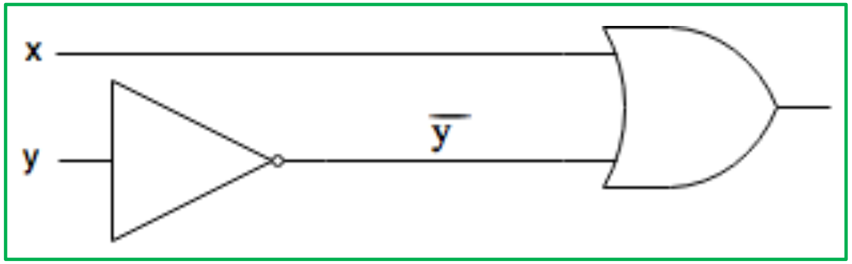


# Esempio 4: dalla Funzione al Circuito

- Circuito per la funzione $\bar{x} + y$

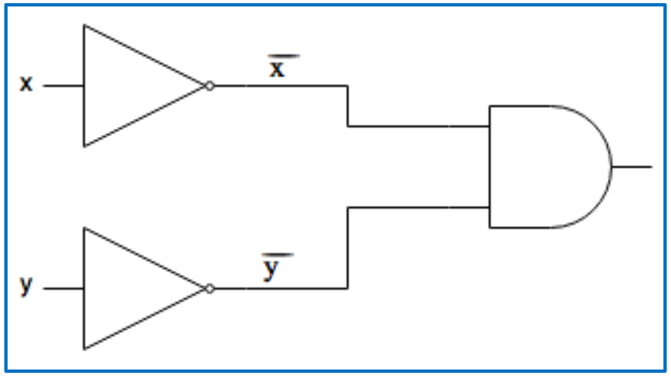



- Circuito per la funzione $x + \bar{y}$

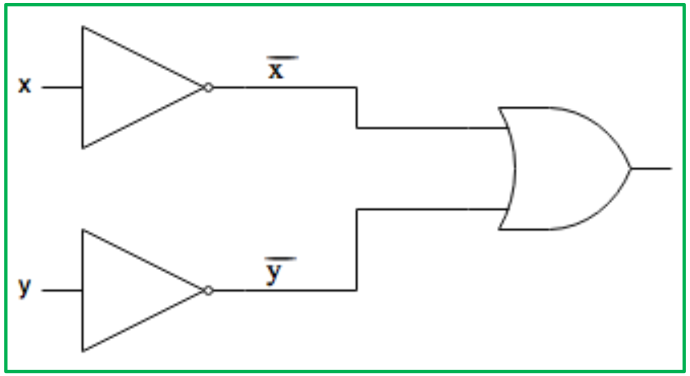


# Esempio 5: dalla Funzione al Circuito

- Circuito per la funzione $\bar{x}\bar{y}$

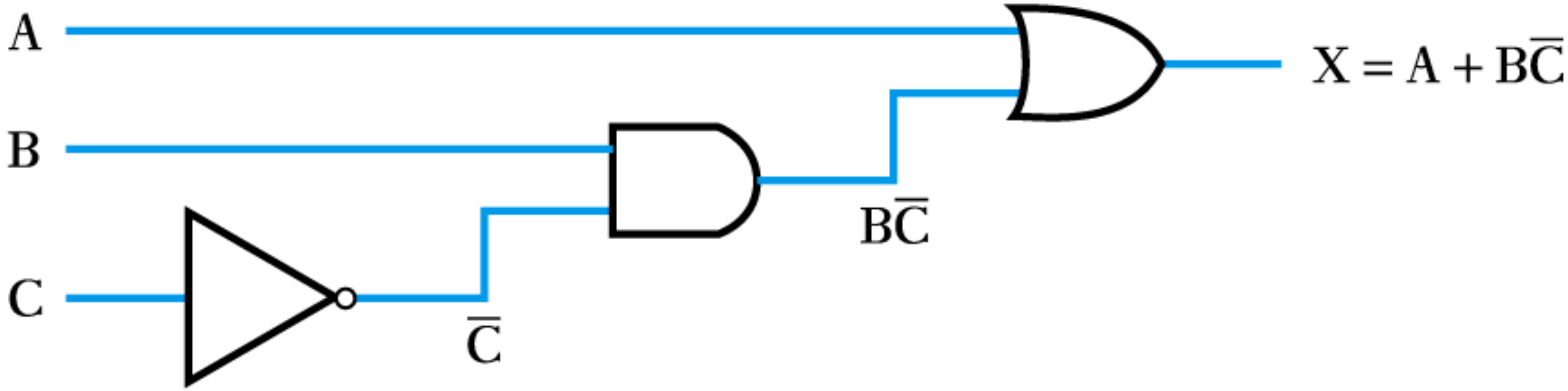



- Circuito per la funzione $\bar{x} + \bar{y}$



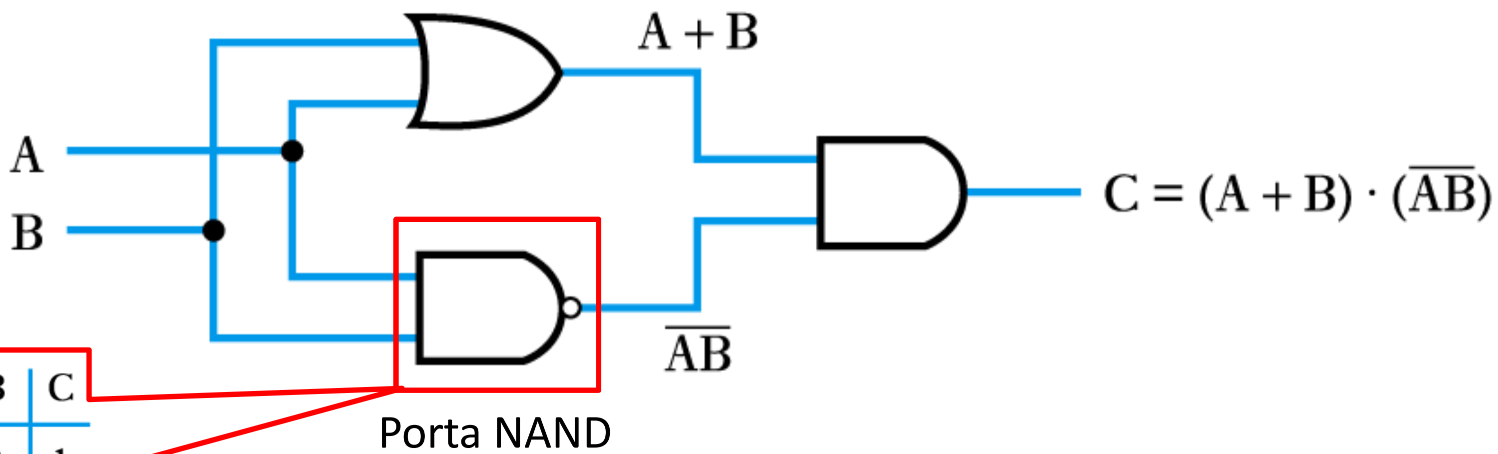
# Esempio 6: dalla Funzione al Circuito

$$X = A + B\overline{C}$$



# Esempio 7: dalla Funzione al Circuito

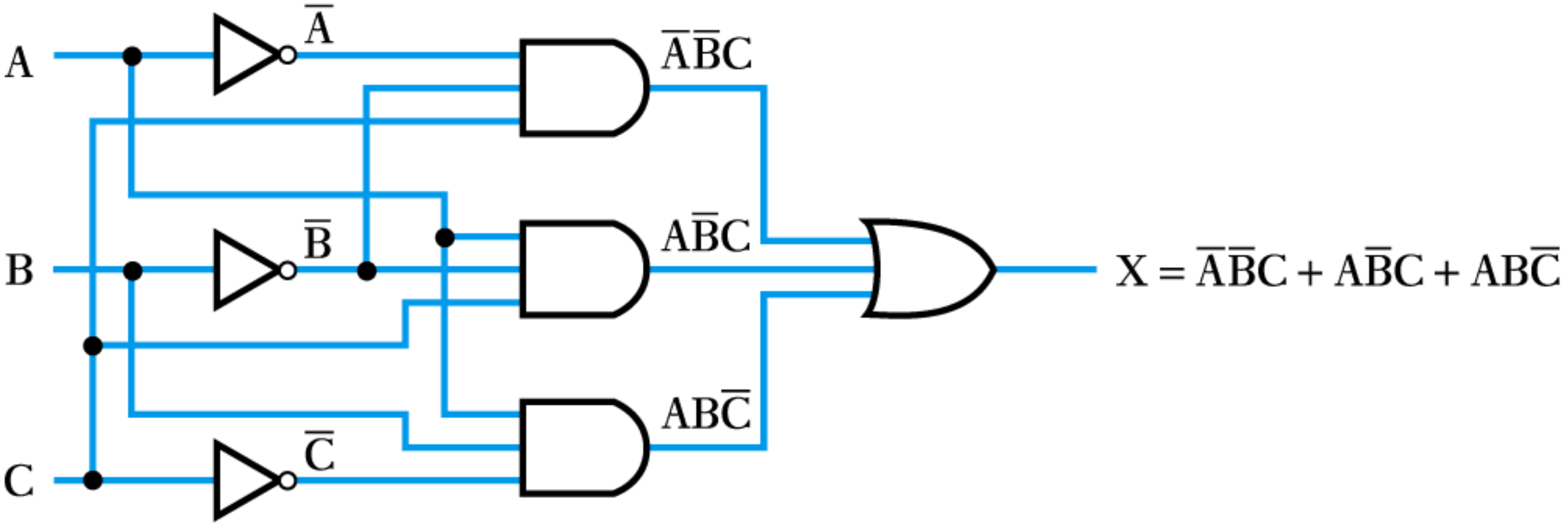
$$C = (A + B) \cdot \overline{(AB)}$$




| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Esempio 8: dalla Funzione al Circuito

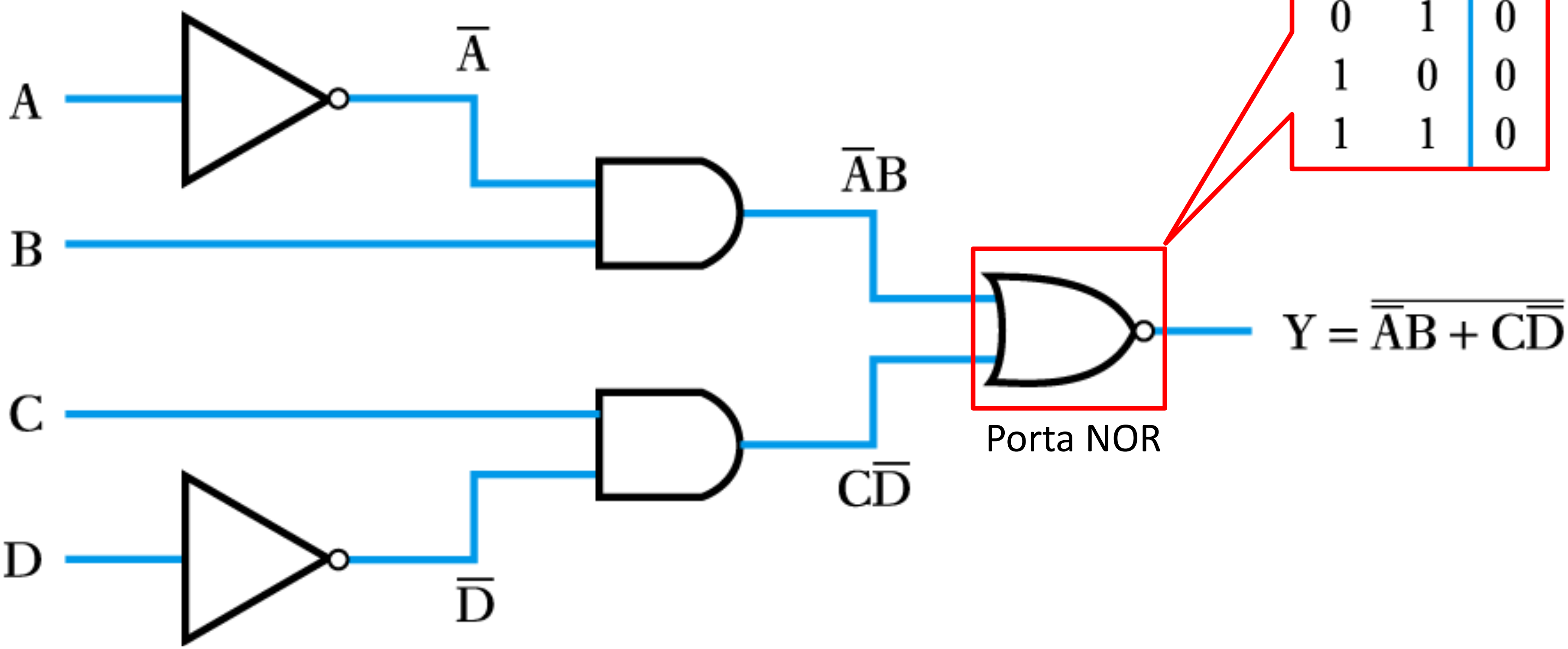
$$X = \overline{A}\,\overline{B}C + A\overline{B}C + AB\overline{C}$$



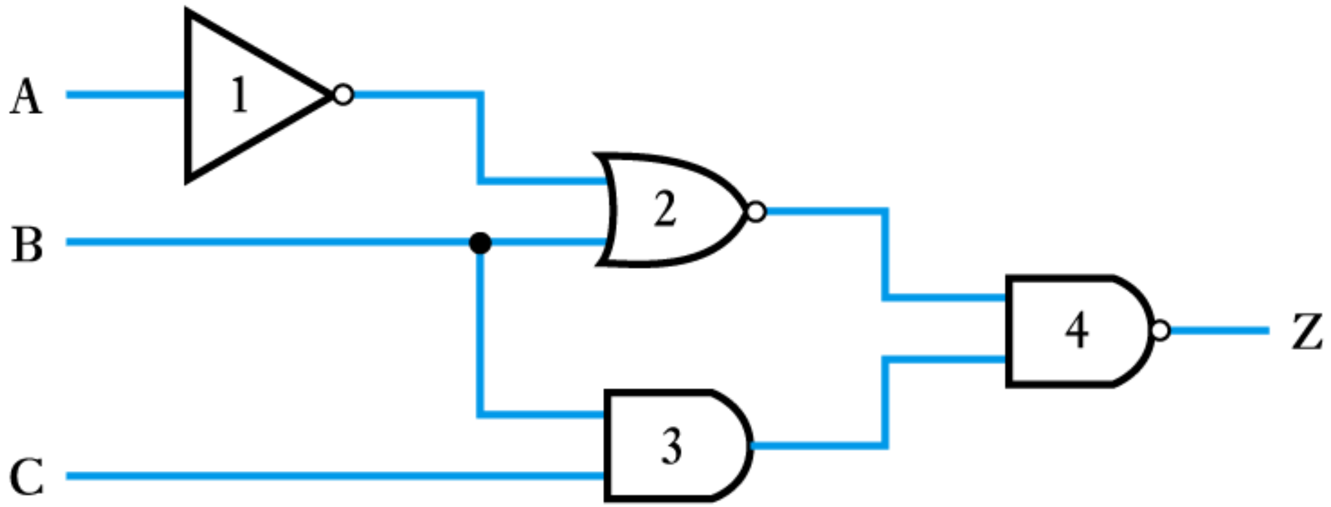
# Esempio 9: dalla Funzione al Circuito

$$Y = \overline{\overline{A}B + C\overline{D}}$$

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

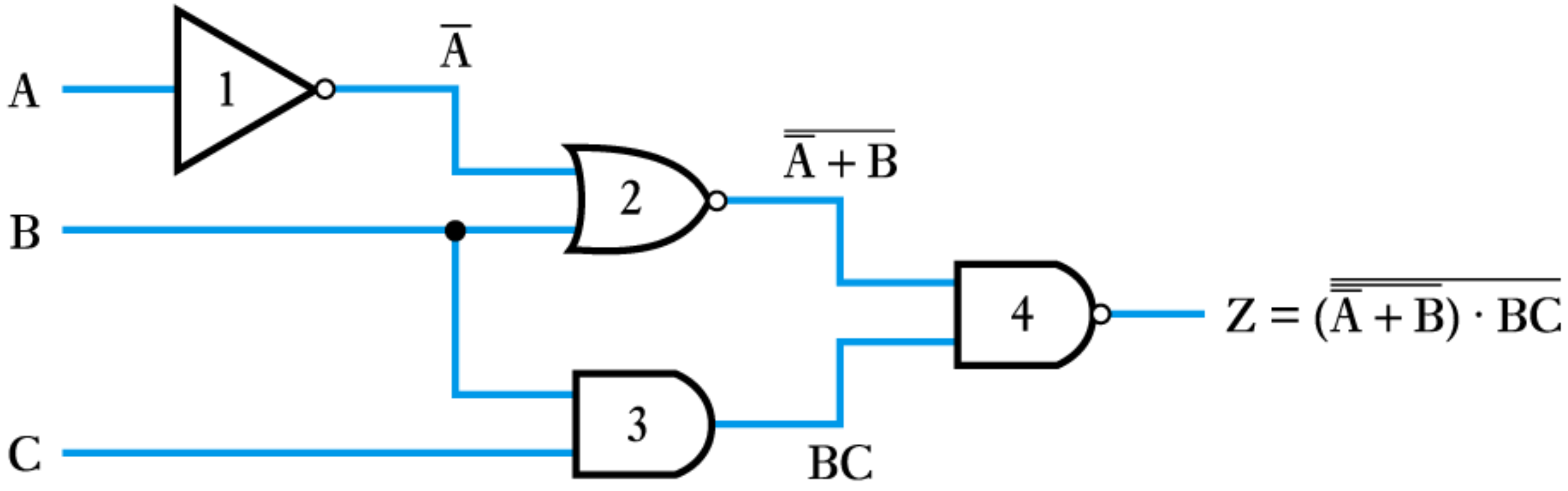


# Esempio 10: dal Circuito alla Funzione – 1/2



# Esempio 10: dal Circuito alla Funzione – 2/2

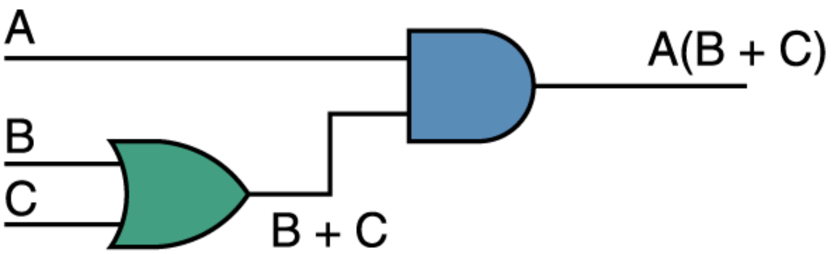
- Procedere progressivamente dagli input verso l'output aggiungendo a turno le espressioni logiche all'output di ciascuna porta logica



# Esempio 11: Funzione => Tavola di Verità => Circuito

- Si consideri la seguente funzione: $A(B + C)$

| A | B | C | B + C | A(B+C) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |



# Ricapitolando…

- Abbiamo visto che una **funzione logica** (ma anche un **circuito logico**) può essere **espressa in due modi**
  - **Tavola di Verità**
  - **Porte Logiche**
- Perché abbiamo bisogno di tutte queste diverse rappresentazioni?
  - Alcune sono più facili di altre per cominciare a progettare un circuito
  - Di solito si comincia con la tavola di verità
  - Si deriva un’espressione booleana da essa (magari esemplificata)
  - Si trasforma l’espressione booleana in un circuito

# Esercizio 1: determinare la funzione espressa dalla seguente tavola di verità

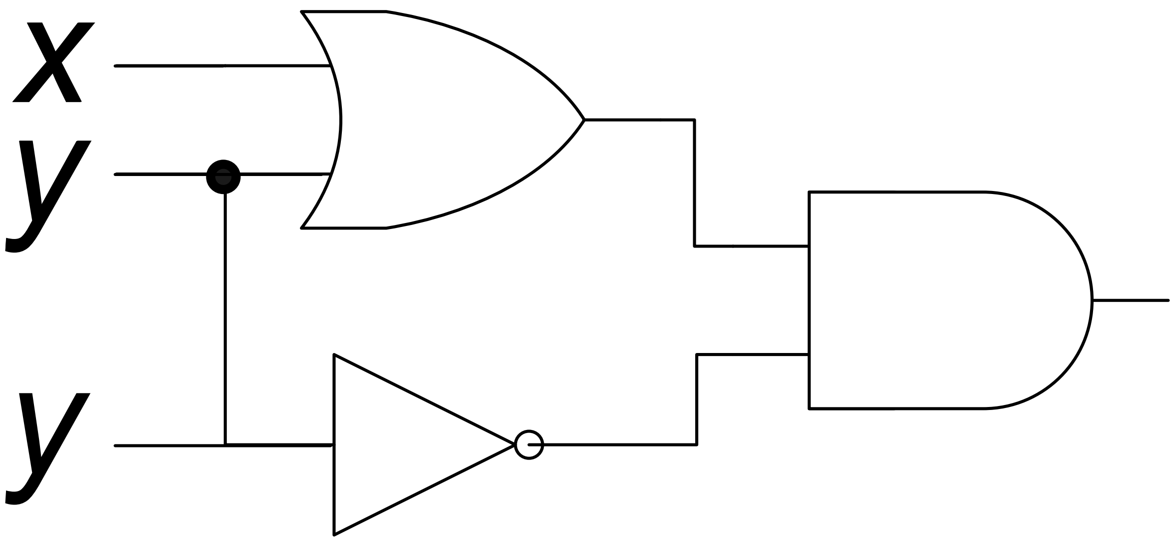
| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

# Esercizio 2: ricavare la tavola di verità relativa alla seguente funzione $\mathbf{F}$

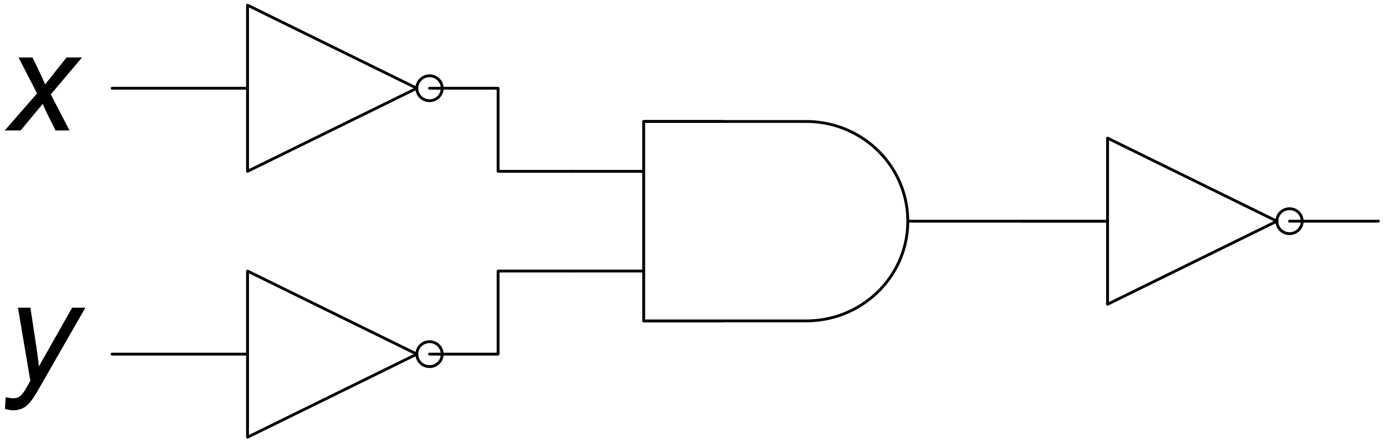
- Vediamo un esempio per la funzione
  - $F = x \times (\overline{y + z})$

| $x$ | $y$ | $z$ | $F$ |
|---|---|---|---|
| 0 | 0 | 0 | ? |
| 0 | 0 | 1 | ? |
| 0 | 1 | 0 | ? |
| 0 | 1 | 1 | ? |
| 1 | 0 | 0 | ? |
| 1 | 0 | 1 | ? |
| 1 | 1 | 0 | ? |
| 1 | 1 | 1 | ? |

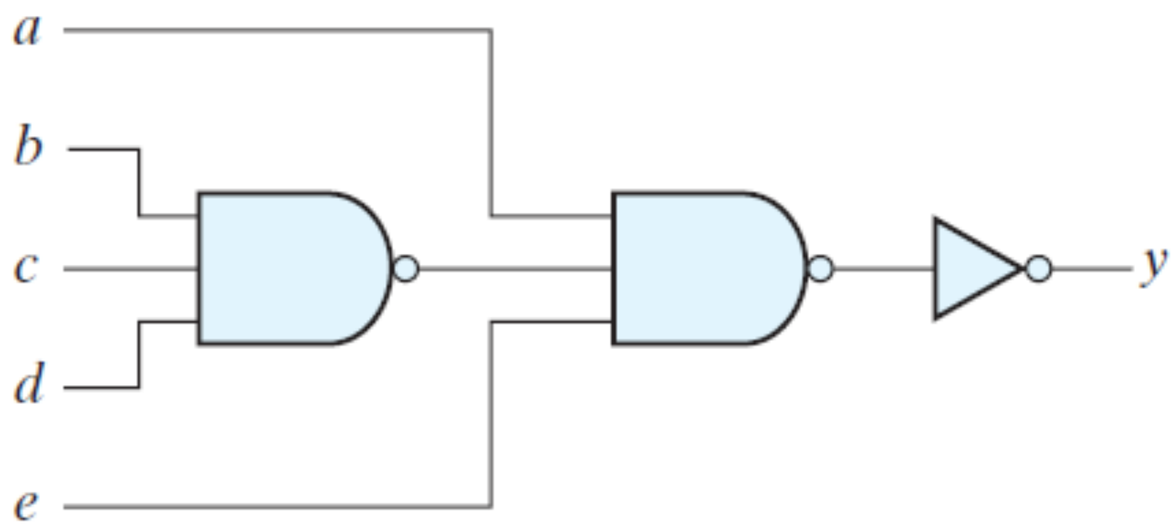
# Esercizio 3: trovare l'output del seguente circuito



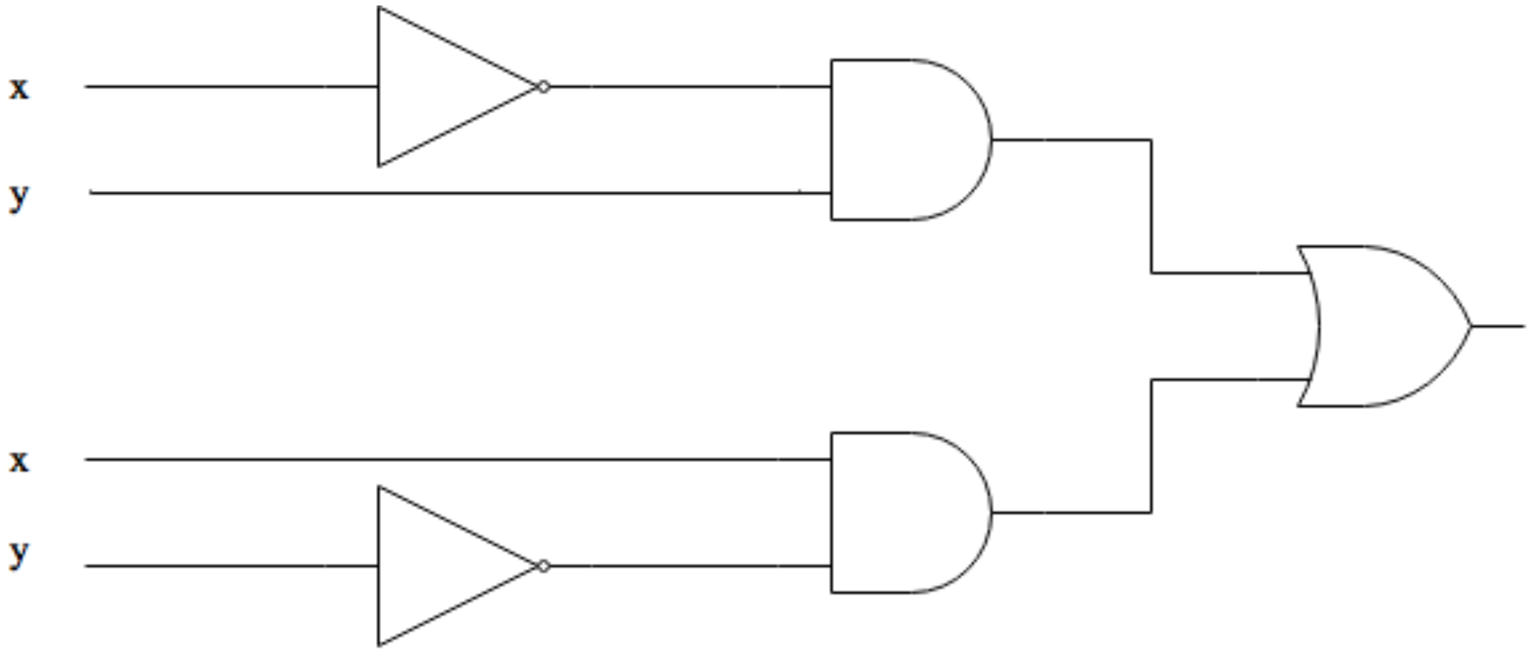
# Esercizio 4: trovare l'output del seguente circuito



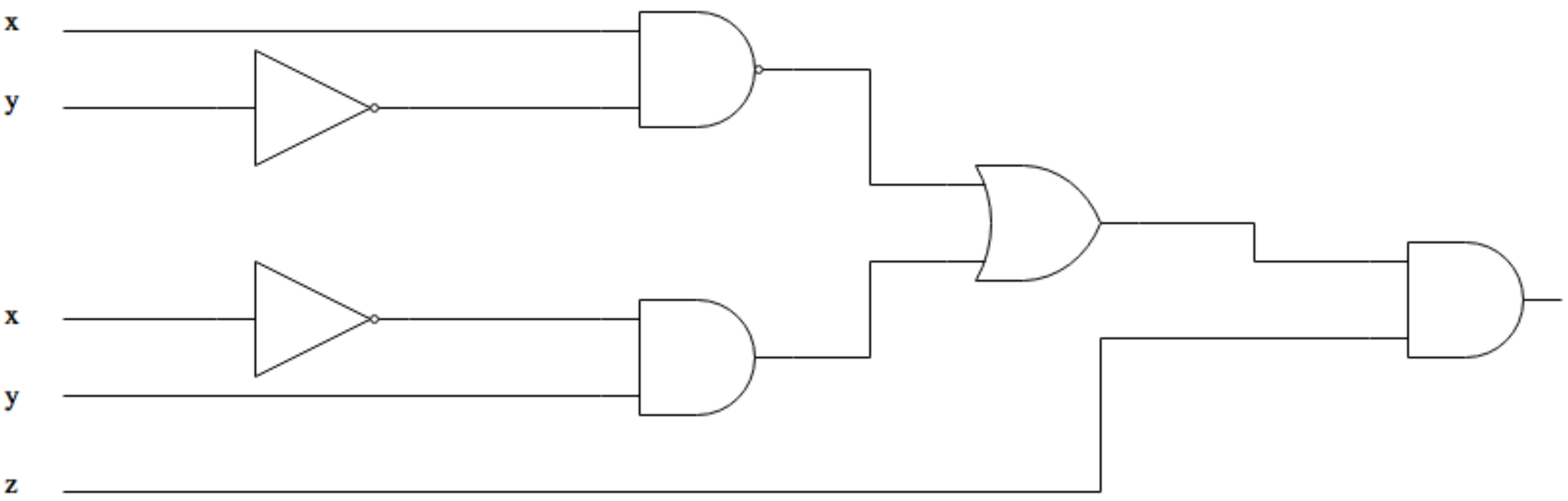
# Esercizio 5: trovare l'output del seguente circuito



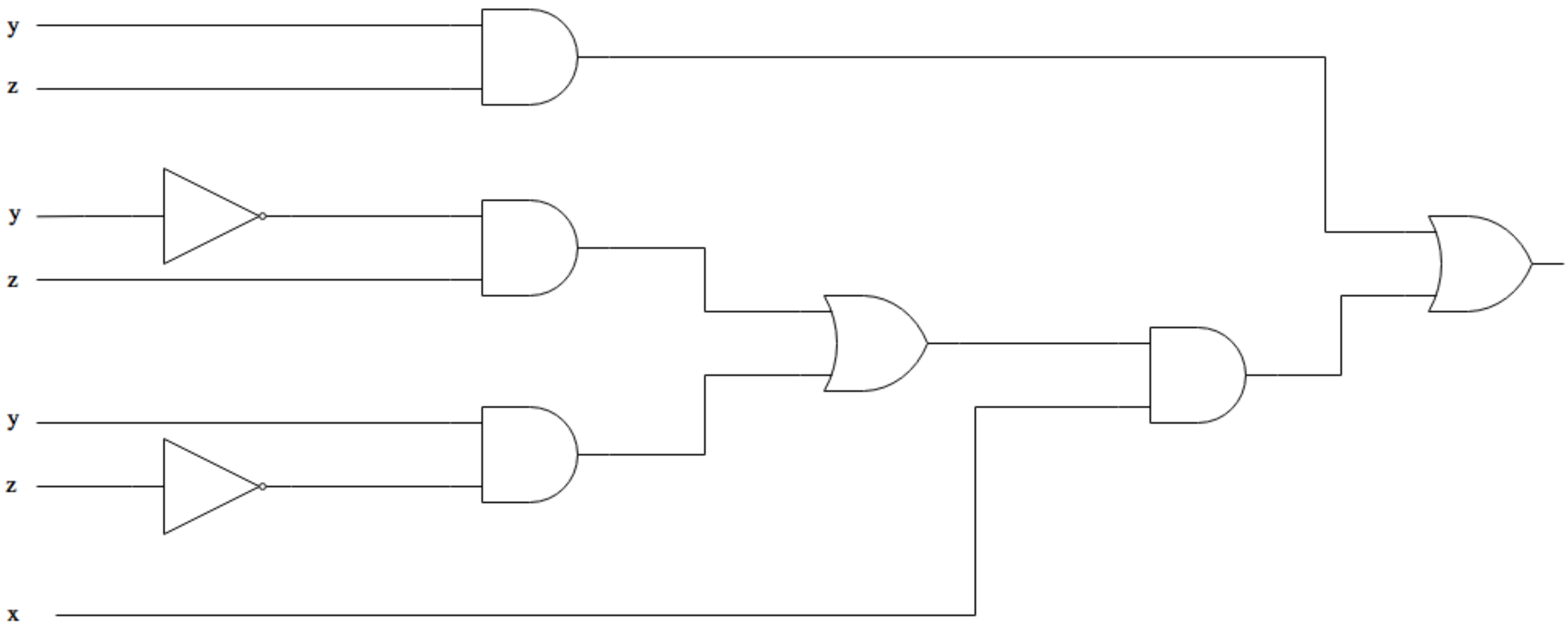
# Esercizio 6: trovare l'output del seguente circuito



# Esercizio 7: trovare l'output del seguente circuito



# Esercizio 8: trovare l'output del seguente circuito



# Esercizio 9: progettare il circuito per ciascuna delle seguenti espressioni

- $\bar{x} + y$
- $\overline{(x + y)}x$
- Scrivere la funzione XOR usando AND, OR e NOT

# Riferimenti

- **Libro di testo**
  - Capitolo 3
    - Paragrafo 4
- **Altri riferimenti**
  - http://www.di.unito.it/~piccolo/teach/AA1516/Lezioni/Lezione2.pdf
  - http://liceocuneo.it/basteris/wp-content/uploads/sites/3/CIRCUITI20DIGITALI1.pdf
  - http://bias.csr.unibo.it/maltoni/arc/Dispense/LogicaDigitale.pdf
  - http://people.unipmn.it/bobbio/DIDATTICA/ARCH1_00/ALDISP_00/varbol00.pdf