Corso di
Architettura degli Elaboratori
a.a. 2021/2022

Il livello logico digitale:
Algebra Booleana e
Circuiti logici digitali di base

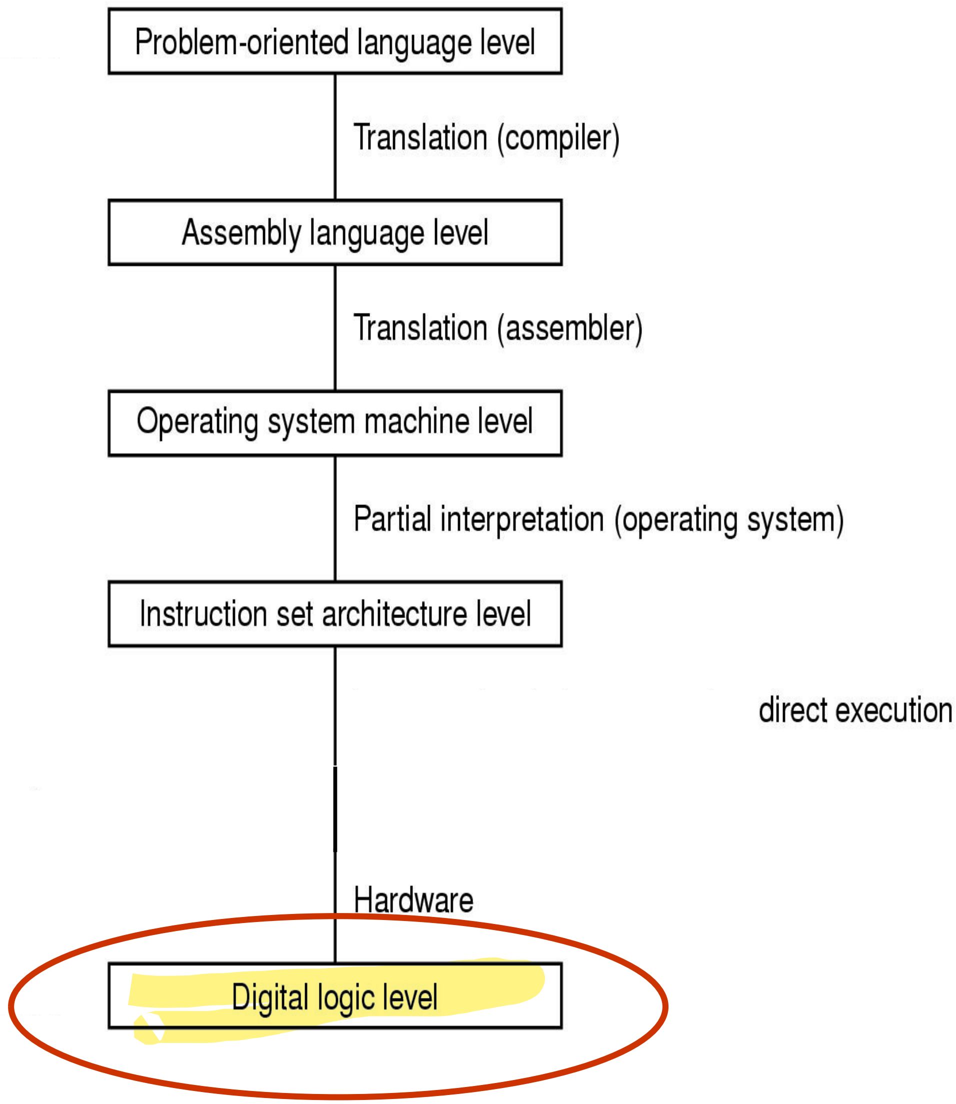
Livello della logica digitale

Livello della *Logico-Digitale*

Costituenti di base del computer:

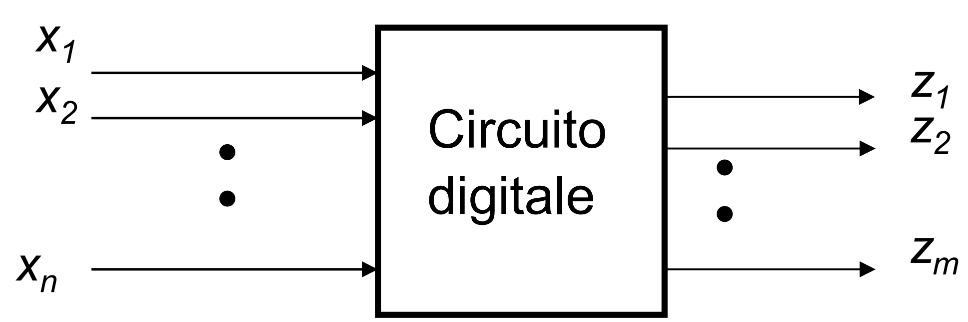
- porte
- registri
- memoria

Sotto questo livello ci sono i dispositivi (funzionamento interno delle porte: transistor)



Circuiti digitali

- Gli elementi di base con cui si sono costruiti i calcolatori si chiamano **circuiti digitali** (o reti digitali)
- Sono dispositivi che utilizzano **solo due valori logici**: 0 (segnale tra 0 e 1 volt) e 1 (segnale tra 2 e 5 volt).
  - I valori di tensione possono anche essere altri.
- Un circuito digitale trasforma segnali (binari) di **ingresso** $x_1, x_2, \ldots, x_n$ nei segnali (binari) di **uscita** $z_1, z_2, \ldots, z_m$.



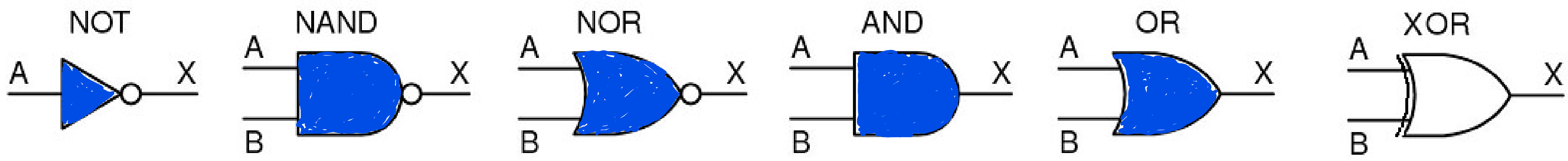
Porte logiche

- I circuiti sono detti ***combinatori*** quando l'uscita è funzione esclusivamente dell'ingresso; sono detti ***sequenziali*** quando l'uscita è funzione oltre che dell'ingresso anche di uno ***stato***.

- Gli elementi primitivi dei circuiti digitali sono chiamati ***porte logiche*** e calcolano alcune funzioni di questi segnali a due valori.

- Questi dispositivi si basano sul fatto che si può far funzionare un ***transistor*** come un interruttore binario molto veloce.

Porte logiche



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

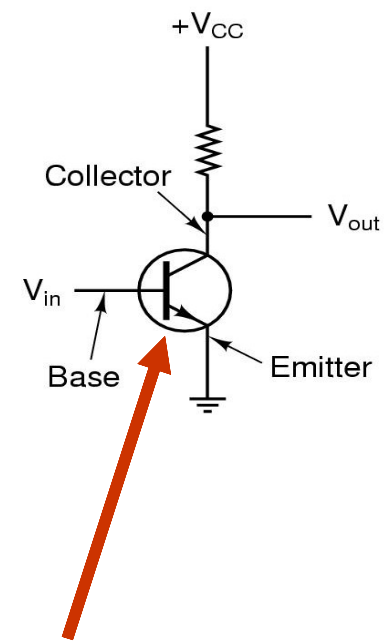
(f)

Ovvi

Transistor

Il transistor si comporta come un interruttore binario molto veloce:

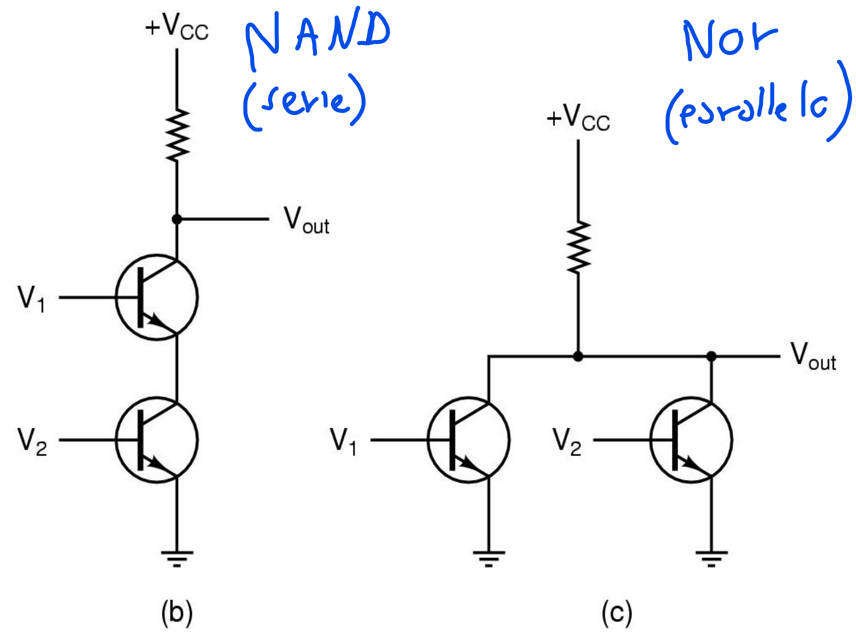
- quando V_{in} è basso, il transistor si disabilita (circuito aperto) e si comporta come “*una resistenza infinita*”, quindi V_{out} è alto; viceversa quando V_{in} è alto, il transistor si attiva e si comporta come “*un filo*” mettendo a terra V_{out}
- pochi ***nanosecondi*** per passare da uno stato all'altro: “alto” (tensione V_{CC}) 1 logico, “basso” (terra) 0 logico



Transistor bipolare

Porte logiche

- ***Logica positiva* (0 = bassa tensione; 1= alta tensione)**
  - (b) una ***porta* NAND**: due transistor collegati in serie
  - (c) una ***porta* NOR**: due transistor collegati in parallelo



V_1	V_2	V_{out}
0	0	1
0	1	1
1	0	1
1	1	0

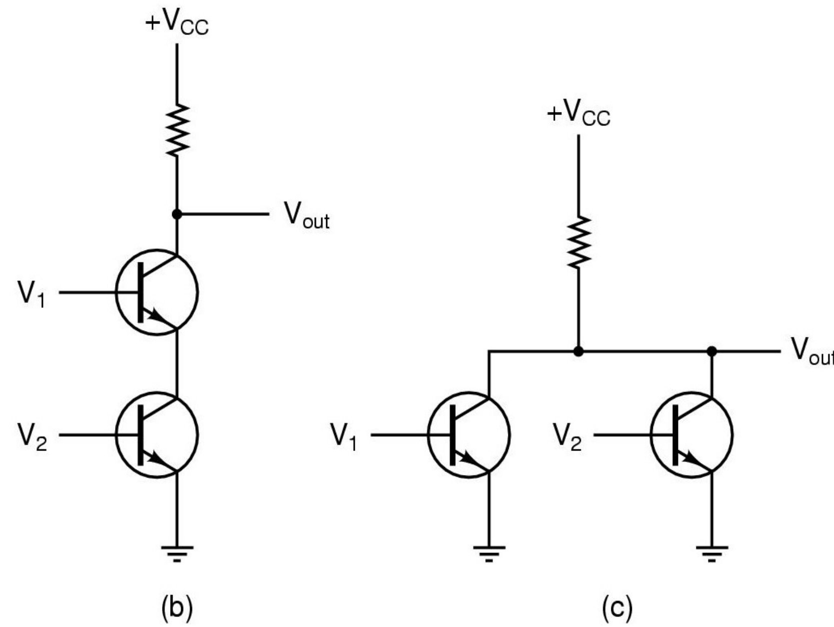
(b)

V_1	V_2	V_{out}
0	0	1
0	1	0
1	0	0
1	1	0

(c)

Porte logiche

- ***Logica negativa* (0 = alta tensione; 1 = bassa tensione)**
  - (b) una ***porta NOR***: due transistor collegati in serie
  - (c) una ***porta NAND***: due transistor collegati in parallelo



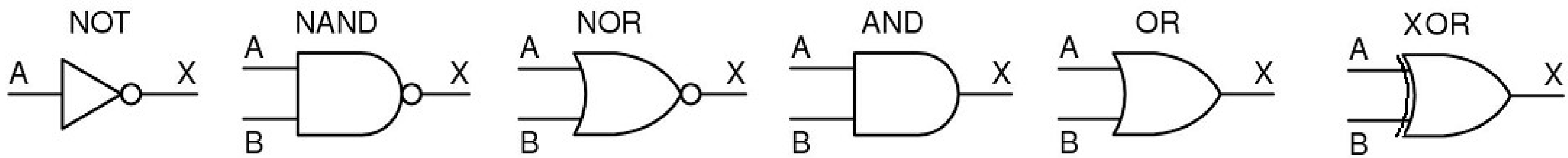
V_1	V_2	V_{out}
1	1	0
1	0	0
0	1	0
0	0	1

(b)

V_1	V_2	V_{out}
1	1	0
1	0	1
0	1	1
0	0	1

(c)

Porte logiche



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

(f)

- NOT: un transistor (invertitore)
- NAND e NOR: due transistor
- AND e OR: tre transistor
  (quelli del NAND e NOR, rispettivamente, più un invertitore
- XOR: 8 transistor (A XOR B) = (A OR B) AND (A NAND B)

Algebra Booleana (George Boole, 1815-1864)

- L’analisi e la progettazione del comportamento dei circuiti digitali si fonda sull’Algebra di Boole
  - Analisi: modo sintetico di descrivere le funzioni dei circuiti digitali
  - Progettazione: data una funzione del circuito digitale sviluppo di implementazione semplificata (o ottimizzata)
- Come ogni altra algebra si fa uso di variabili e di operazioni
  - Variabili logiche: possono assumere solo il valore 0 (FALSO) o 1 (VERO)
  - Operazioni logiche: AND, OR e NOT

Algebra Booleana (George Boole, 1815-1864)

- Due valori costanti 0 e 1
- Operatore unario "**NOT**":
  - $\overline{A}$: "not" A ($\neg$ A)
- Operatori binari "**AND**" e "**OR**":
  - AB: A "and" B
  - A + B: A "or" B
- Una qualunque combinazione di variabili o costanti booleane legate tra loro dagli operatori fondamentali è una ***espressione logica***
- *Esempio*: $A\overline{B} + B\overline{C}$ (è vera solo quando A = 1 e B = 0 *oppure* B = 1 e C = 0)
- In assenza di parentesi AND ha precedenza su OR

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\overline{B}$

Algebra Booleana

- Dimostriamola:

$A+BC=(A+B)(A+C)$ TESI

$(A+B)(A+C)=$

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Diversa dall'algebra ordinaria

Algebra Booleana

- Dimostriamola:

$A+BC=(A+B)(A+C)$ TESI

$(A+B)(A+C)=$
$= AA+AC+BA+BC=$
$= A+AC+AB+BC=$
$= A(1+C+B)+BC=$
$= A+BC$

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Diversa dall'algebra ordinaria

Algebra Booleana

- Dimostriamola:

$\overline{(A+B)}=\bar{A}\bar{B}$ TESI

$\overline{(A+B)}(A+B)=0$

se vale la tesi allora

$(\bar{A}\bar{B})(A+B)=0$

$\bar{A}\bar{B}A+\bar{A}\bar{B}B=0\bar{B}+\bar{A}0=0$

CVD

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Una forma della legge di De Morgan

Algebra Booleana

- Dimostriamola:

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

$\overline{(AB)}(AB)=0$

se vale la tesi allora

$(\overline{A}+\overline{B})(AB)=0$

$\overline{A}BA+A\overline{B}B=0B+A0=0$

CVD

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Una forma della legge di De Morgan

Algebra Booleana

- Dimostrate le due equivalenti forme per lo XOR e calcolate numero **di porte** e di **transistor**:

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

$$(A+B)(\overline{A}+\overline{B}) =$$

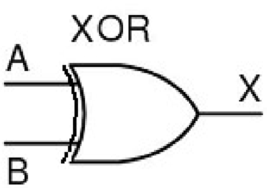
$$A\cdot(\overline{A}+\overline{B})+B(\overline{A}+\overline{B})=$$

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

$$0+A\overline{B}+B\overline{A}+0$$

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

cvd



A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

(f)

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Algebra Booleana

- Dimostrate le due equivalenti forme per lo XOR e calcolate numero di porte e di transistor:

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

partiamo da

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

De Morgan

De Morgan's law	$\overline{AB}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}\,\overline{B}$

Algebra Booleana

- Dimostrate le due equivalenti forme per lo XOR e calcolate numero di porte e di transistor:

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

Sempre 1, per cui posso metterli in AND

partiamo da

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

Algebra Booleana

- Dimostrate le due equivalenti forme per lo XOR e calcolate numero di porte e di transistor:

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

partiamo da

Idempotenza ed AND con propria negazione

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

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

Algebra Booleana

- Dimostrate le due equivalenti forme per lo XOR e calcolate numero di porte e di transistor:

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

partiamo da

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

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

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

Algebra Booleana

- Dimostrate le due equivalenti forme per lo XOR e calcolate numero di porte e di transistor:

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

Sempre uguali a 1

partiamo da

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

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

$=(AB+A\overline{B}+B)(\overline{A}+A\overline{B}+\overline{A}\overline{B})=(A\overline{B}+B(A+1))(A\overline{B}+\overline{A}(1+\overline{B}))=$

Algebra Booleana

- Dimostrate le due equivalenti forme per lo XOR e calcolate numero di porte e di transistor:

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

AND con propria negazione: sempre 0

partiamo da

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

$=(AB+A\bar{B}+BB+B\bar{B})(\bar{A}A+\bar{A}\bar{A}+A\bar{B}+\bar{A}\bar{B})=$

$=(AB+A\bar{B}+B)(\bar{A}+A\bar{B}+\bar{A}\bar{B})=(A\bar{B}+B(A+1))(A\bar{B}+\bar{A}(1+\bar{B}))=$

$=(A\bar{B}+B)(A\bar{B}+\bar{A})=A\bar{B}A\bar{B}+A\bar{B}\bar{A}+BA\bar{B}+\bar{A}B=A\bar{B}+\bar{A}B$

Funzioni Booleane

Una ***funzione booleana*** di n variabili $x_1, x_2, \ldots, x_n$ è una *relazione* che associa un valore booleano a ciascuna delle 2^n configurazioni possibili delle n variabili:

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

Una funzione booleana può essere espressa in varie forme:

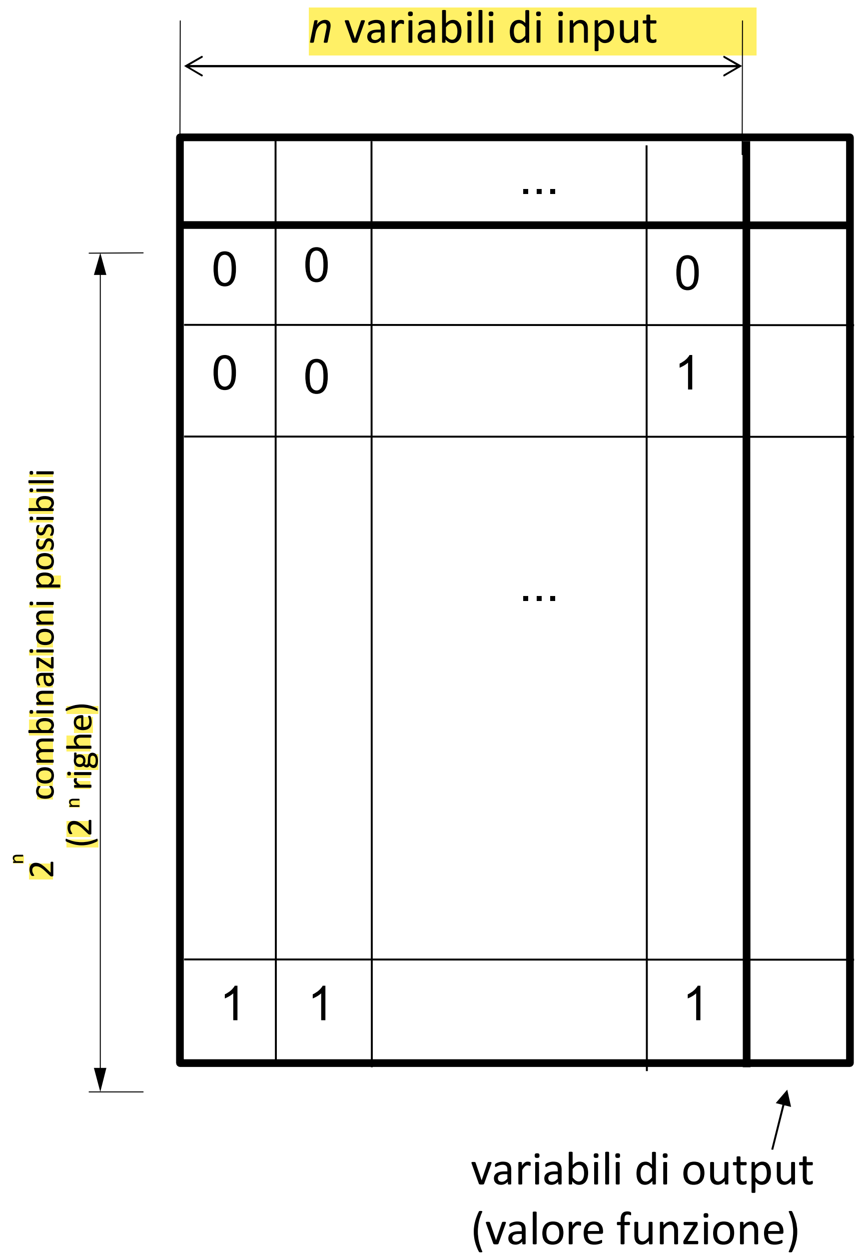
- Tabelle di verità
- Formule algebriche
- Mappe di Karnaugh
- Binary Decision Diagrams (BDDs)

Algebra Booleana

- ***Tabelle di verità:*** descrivono completamente il valore di una funzione Booleana attraverso tutte le combinazioni di input; *n* input corrispondono a 2^n combinazioni (righe)

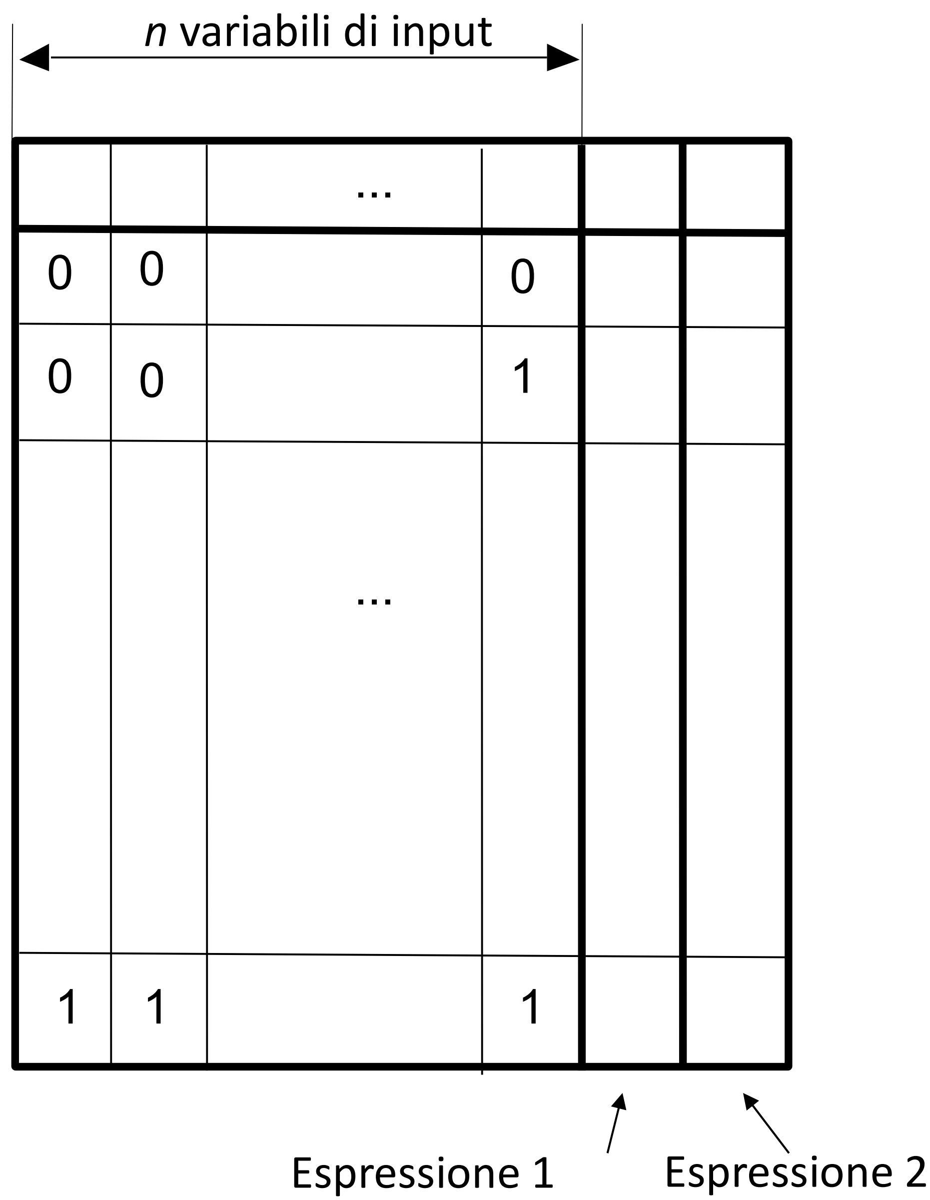
- È ***finito*** l'insieme delle funzioni Booleane di *n* input: 2^{2^n} funzioni
(es., se *n* = 2 allora 16 funzioni diverse)

- ***Forma canonica:*** righe ordinate per valori crescenti degli ingressi interpretando i valori delle variabili di ingresso come cifre di una codifica binaria



Equivalenza con tabelle di verità

Due, o più, diverse espressioni algebriche sono equivalenti se hanno identiche tabelle di verità



Algebra Booleana: forme canoniche

- ***Formula normale disgiuntiva (FND):***
  - Sommatoria (OR) di termini ciascuno dei quali è una produttoria (AND) di ***letterali*** costituiti da nomi di variabili di ingresso o da negazioni dei nomi di variabili di ingresso.
  - È ***minimale*** quando, applicando le proprietà algebriche di equivalenza non è possibile ottenere una FND equivalente contenente un numero di letterali inferiore
- ***Formula normale congiuntiva (FNC)***
  - Concetto "duale" del precedente ossia è una produttoria (AND) di termini ciascuno dei quali è una sommatoria (OR) di ***letterali*** costituiti da nomi di variabili di ingresso o da negazioni di nomi di variabili di ingresso

Algebra Booleana

- *Funzione di maggioranza su tre input*: restituisce 1 se la maggioranza degli input è 1, 0 altrimenti
- La funzione produce 1 nella quarta, sesta, settima e ottava riga
- La funzione M è 1 nelle righe: $\overline{A}BC$, $A\overline{B}C$, $AB\overline{C}$, ABC
- $M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

  iscrive casi, in cui vale 1
- Forma normale disgiuntiva

A	B	C	M
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

Se mettessimo in OR anche i casi dove M vale 0 non cambierebbe nulla

Algebra Booleana

- La funzione produce 0 nella prima, seconda, terza e quinta riga
- La funzione M è 0 nelle righe:

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

↑ scrivo casi in cui vale 0

- Forma normale congiuntiva

A	B	C	M
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

Se mettessimo in AND anche i casi dove M vale 1 non cambierebbe nulla

Da formula algebrica a tabella di verità

- Elencare tutte le possibili configurazioni delle *n* variabili di ingresso
- Per ogni configurazione, valutare i valori di uscita delle funzioni elementari NOT, AND e OR che compongono l'espressione
- Assumendo l'espressione iniziale una FND (FNC), l'uscita della funzione OR (AND) rappresenta il valore da inserire nella corrispondente riga della tabella che si sta costruendo

Tabella di verità $\longleftrightarrow$ Formula algebrica

Da tabella di verità a formula algebrica

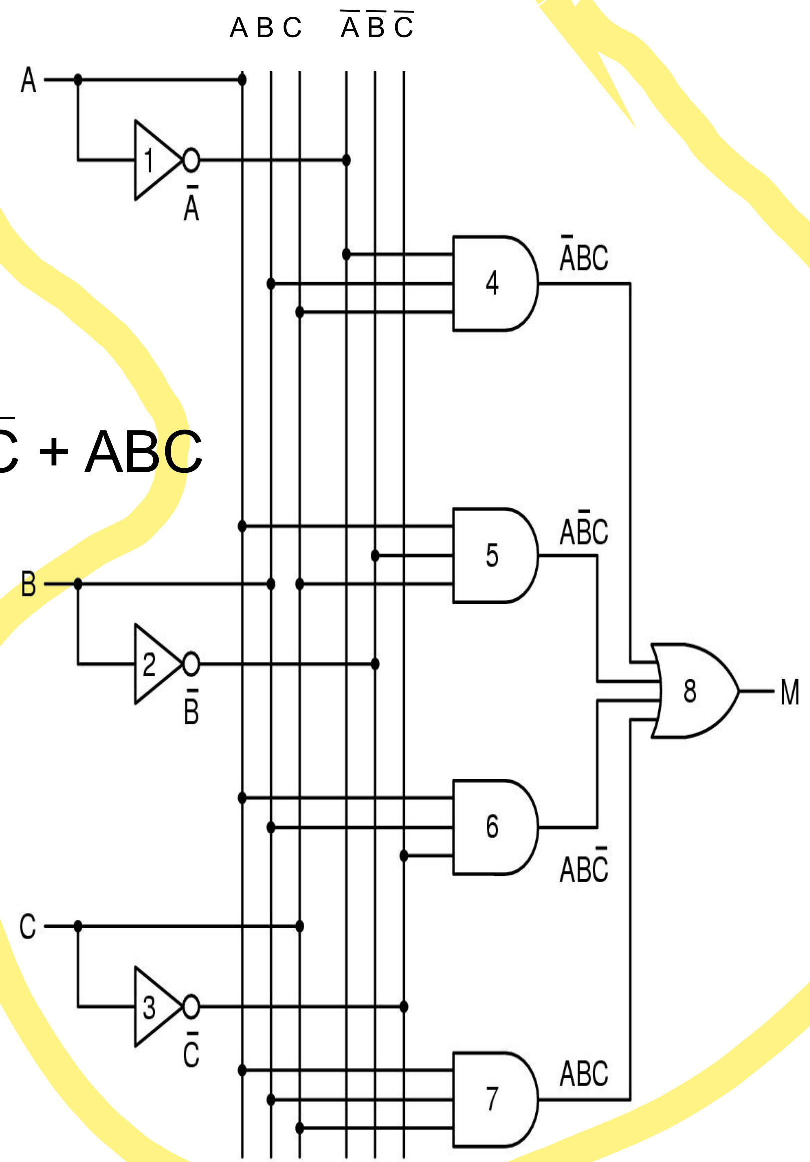
1. Scrivere la tabella di verità per la funzione
2. Disporre gli invertitori per generare il complemento di ogni input
3. Introdurre una porta AND per ogni termine con un 1 nella colonna dei risultati
4. Collegare le porte AND agli input appropriati
5. Inviare l'output di tutte le porte AND in una porta OR

Convenzione: l'incrocio tra due linee non implica alcuna connessione a meno che non sia presente il simbolo • nel punto di intersezione

$$\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

A	B	C	M
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

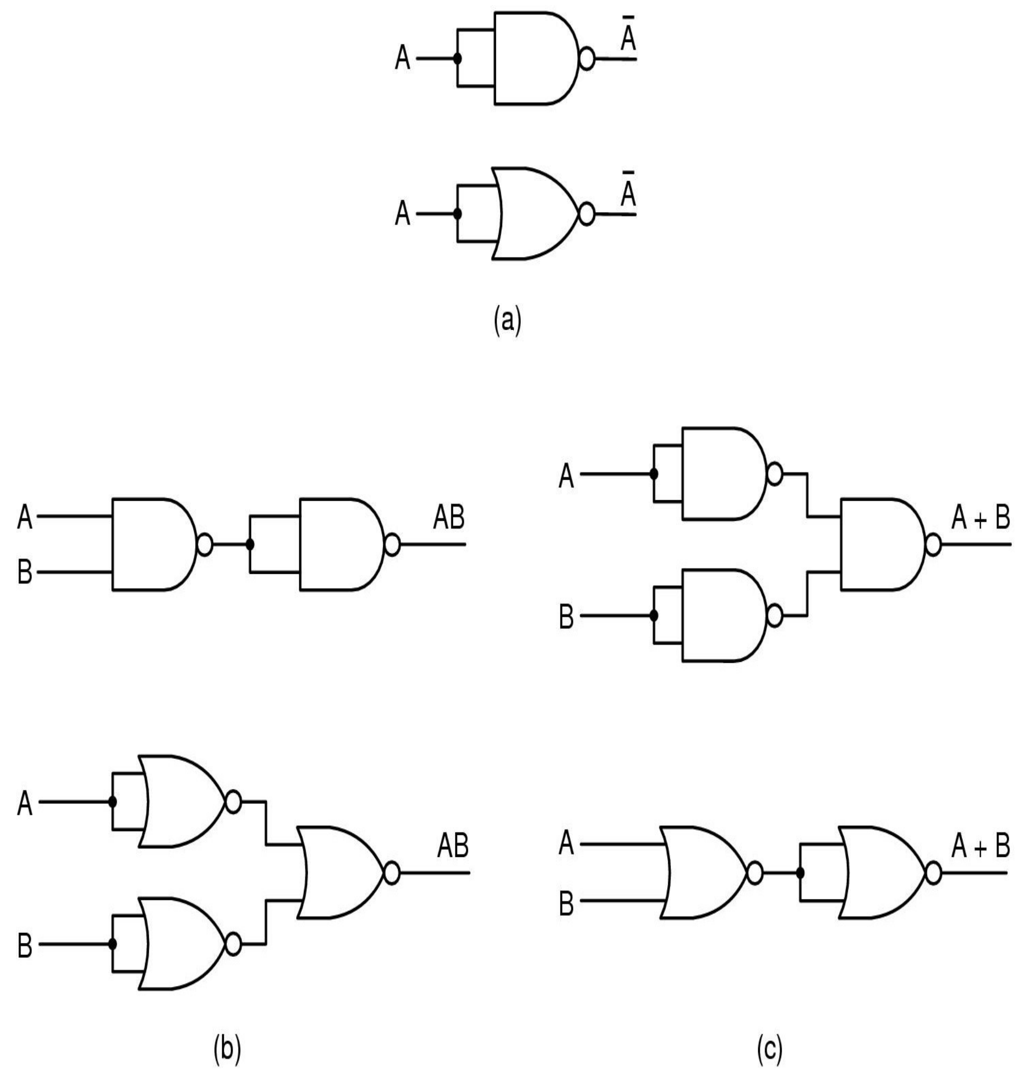
(a)



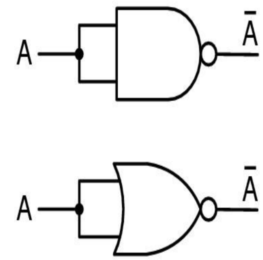
(b)

Implementazione di circuiti

- Sostituire le porte con più input con dei circuiti equivalenti che usano porte a due input
- Convertire il circuito in un solo tipo di porta (per convenienza)
- NAND e NOR sono porte ***complete***
- ***Usate le identità per dimostrare l'equivalenza dei circuiti fatti solo con NAND e NOR***
- Nota: in generale non si ottiene il circuito ottimale (per numero di porte impiegate)

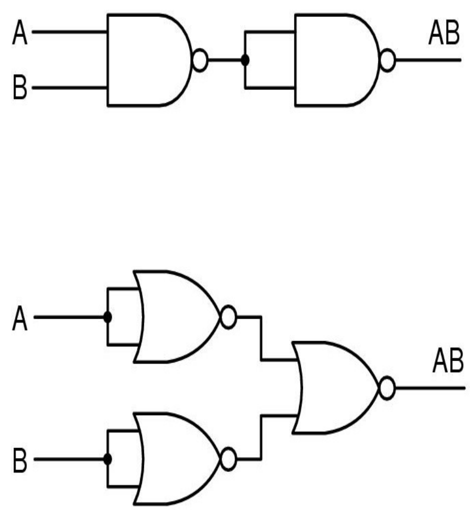


Implementazione di circuiti

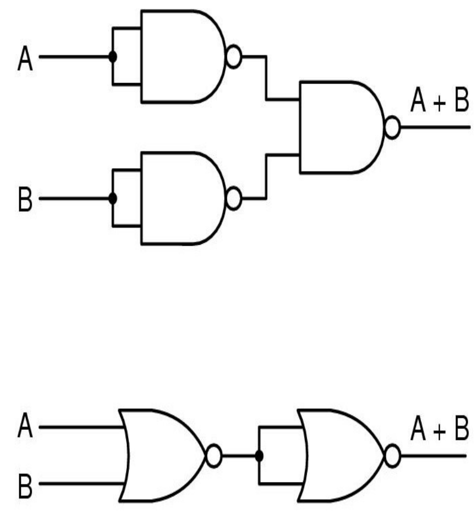


(a)

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$



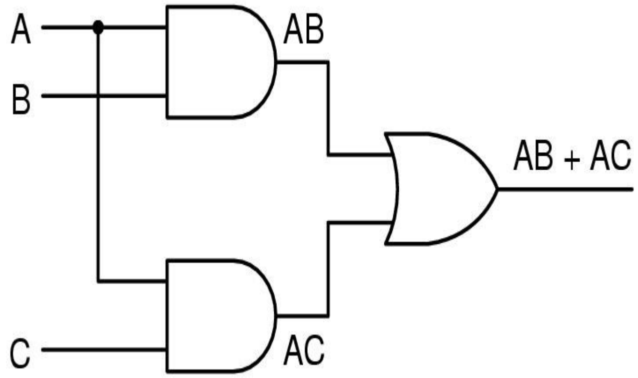
(b)



(c)

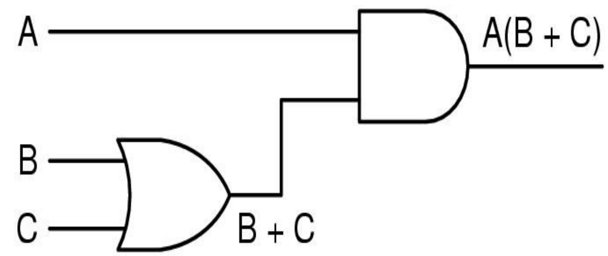
Implementazione di circuiti

- Equivalenza di circuiti: esistono più circuiti che realizzano la stessa funzione booleana
- È importante trovare quella più semplice nel senso del minor numero di porte
- Usare a tal fine le proprietà dell'algebra Booleana



A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

(a)



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

(b)

Circuiti di base

- Circuiti integrati (IC) o ***chip***
- I chip si possono classificare in:
  - SSI (*Small Scale Integrated*): da 1 a 10 transistor
  - MSI (*Medium Scale Integrated*): da 10 a 100 transistor
  - LSI (*Large Scale Integrated*): da 100 a 100.000 transistor
  - VLSI (*Very Large Scale Integrated*): più di 100.000 transistor
  - ULSI (*Ultra Large Scale Integrated*): fino a 10 milioni di transistor