FONDAMENTI DI INFORMATICA

Lezione n. 3

- **FORME CANONICHE.**
- **TRASFORMAZIONI.**
- **ESERCIZI.**

In questa lezione verranno considerate le proprietà dell'algebra booleana che saranno poi utili per l'analisi e la progettazione di circuiti a livello logico.

Introdurremo poi le tecniche per trasformare la rappresentazione algebrica di un problema nella rappresentazione circuitale.

UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione
COMPUTER ENGINEERING

FORME CANONICHE - I

Un *minterm* è una espressione prodotto che contiene in modo affermato o negato tutte le variabili della funzione.

- **In una funzione di 3 variabili** $\{x_1, x_2, x_3\}$ **sono minterm:**

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

non sono minterm:

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

UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione
COMPUTER ENGINEERING

FORME CANONICHE - II

Un *maxterm* è una espressione somma che contiene in modo affermato o negato tutte le variabili della funzione.

- **In una funzione di 3 variabili** $\{x_1, x_2, x_3\}$ **sono maxterm:**

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

non sono maxterm:

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

UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione
COMPUTER ENGINEERING

FORME CANONICHE - III

Le somme di minterm o i prodotti di maxterm sono detti forme canoniche.

- **Esempi di forme canoniche:**

•SdP: $(\overline{x_1} \bullet \overline{x_2} \bullet x_3) + (x_1 \bullet \overline{x_2} \bullet x_3)$

•PdS: $(\overline{x_1} + \overline{x_2} + x_3)(x_1 + \overline{x_2} + x_3)$

UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione
COMPUTER ENGINEERING

LA SOMMA DI NUMERI BINARI

Riporto	*1 1 1 0*
Addendo	1 0 1 1 +
Addendo	0 1 1 1 =
Somma	1 0 0 1 0

X_0	Y_0	C_0	S_0	C_1
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

UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione
COMPUTER ENGINEERING

FORME CANONICHE - IV

$C_1 = 1$ se

x_0	y_0	C_0	
0	1	1	$\overline{x_0}\, y_0 C_0 +$
1	0	1	$x_0 \overline{y_0} C_0 +$
1	1	0	$x_0 y_0 \overline{C_0} +$
1	1	1	$x_0 y_0 C_0$

Una somma di *minterm* rappresenta direttamente tutti gli 1 di una funzione.

FORME CANONICHE - V

$C_1 = 0$ se	x_0	y_0	C_0	
	0	0	0	$(x_0 + y_0 + C_0)$
	0	0	1	$(x_0 + y_0 + \overline{C}_0)$
	0	1	0	$(x_0 + \overline{y}_0 + C_0)$
	1	0	0	$(\overline{x}_0 + y_0 + C_0)$

Un prodotto di *maxterm* rappresenta direttamente tutti gli 0 di una funzione.

FORME CANONICHE - V

Una qualunque funzione $f(x_1, x_2, x_3, \ldots x_n)$

è espressa univocamente come:

- **Somma di minterm:**

$$\sum_i (\dot{x}_{i1} \cdot \dot{x}_{i2} \cdot \ldots \cdot \dot{x}_{in})$$

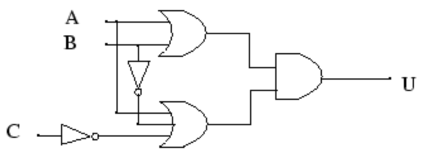
- **Prodotto di maxterm:**

$$\prod_i (x_{i1} + \dot{x}_{i2} + \ldots + \dot{x}_{in})$$

LOGICA A DUE LIVELLI - I

Una qualunque espressione Prodotto di Somme (Pds) può essere implementata da un circuito a due livelli di logica.

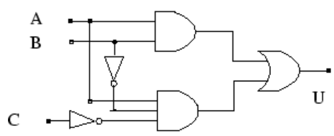
$U = (A+B)(A+\overline{B}+\overline{C})$



LOGICA A DUE LIVELLI - II

Una qualunque espressione Somma di Prodotti (SdP) può essere implementata da un circuito a due livelli di logica.

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



LA SOMMA DI NUMERI BINARI

X_0	Y_0	C_0	S_0	C_1
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

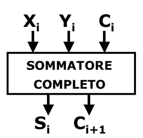
$C_{1(SP)} = \overline{X}_0Y_0C_0 + X_0\overline{Y}_0C_0 + X_0Y_0\overline{C}_0 + X_0Y_0C_0$

$C_{1(PS)} = (X_0+Y_0+C_0)\ (X_0+Y_0+\overline{C}_0)$
$(X_0+\overline{Y}_0+C_0)\ (\overline{X}_0+Y_0+C_0)$

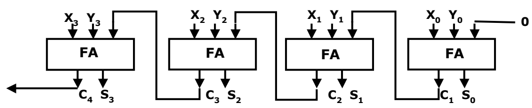
$S_{0(SP)} = \overline{X}_0\overline{Y}_0C_0 + \overline{X}_0Y_0\overline{C}_0 + X_0\overline{Y}_0\overline{C}_0 + X_0Y_0C_0$

$S_{0(PS)} = (X_0+Y_0+C_0)\ (X_0+\overline{Y}_0+\overline{C}_0)$
$(\overline{X}_0+Y_0+\overline{C}_0)\ (\overline{X}_0+\overline{Y}_0+C_0)$

LA SOMMA DI NUMERI BINARI

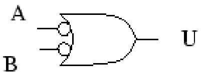


Utilizzando il modulo elementare (*Sommatore completo* o *Full Adder* o *FA*) riportato nella figura è possibile realizzare un circuito che esegue la somma di numeri binari di lunghezza qualsiasi.



Circuito per la somma di numeri binari a 4 bit con propagazione del riporto

LA NEGAZIONE



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

Un cerchio all'ingresso (o all'uscita) di una porta logica ha significato di negazione.

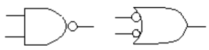
I nuovi simboli così costruiti possono essere utilizzati per rappresentare funzioni booleane anche complesse.

EQUIVALENZE

Il teorema di De Morgan afferma che:

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

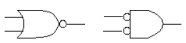
che corrisponde all'equivalenza circuitale:



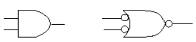
Le relazioni di equivalenza dell'algebra booleana sono interpretate a livello circuitale come relazioni di equivalenza fra moduli logici.

EQUIVALENZE

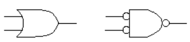
La possibilità di rappresentare in modo diverso le stesse funzioni logiche consente di effettuare trasformazioni circuitali basandosi su proprietà algebriche.



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



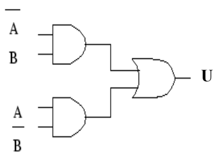
$$A \bullet B = \overline{\overline{A} + \overline{B}}$$



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

TRASFORMAZIONI CIRCUITALI

La funzione EX-OR realizzata con un circuito a due livelli di logica.



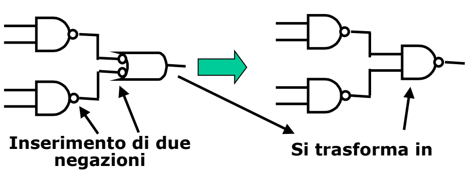
EX-OR

A	B	U
0	0	0
0	1	1
1	0	1
1	1	0

TRASFORMAZIONI CIRCUITALI

- **Aggiungendo due pallini di negazione in serie sulle uscite delle porte NAND.**
- **Trasformando la porta di uscita in porta NAND**

circuito con sole porte NAND .



TRASFORMAZIONI CIRCUITALI

Le trasformazioni precedenti possono essere generalizzate:

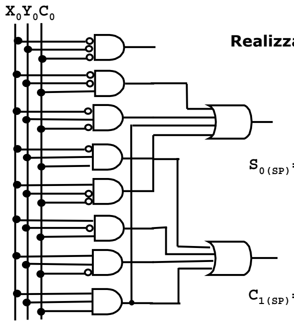
$$A \bullet B \bullet C + D \bullet E \bullet F + \ldots = \overline{\overline{A \bullet B \bullet C} \bullet \overline{D \bullet E \bullet F}}$$

$$(A + B + C) \bullet (D + E + F) \bullet \ldots = \overline{\overline{A + B + C} + \overline{D + E + F}}$$

Ogni funzione (rappresentata come SP o PS) può essere realizzata utilizzando solo porte NAND o porte NOR.

Le porte NAND o NOR sono insiemi completi.

LA SOMMA DI NUMERI BINARI



Realizzazione con i decodificatori

$S_{0(SP)} = \overline{X}_0\overline{Y}_0C_0+\overline{X}_0Y_0\overline{C}_0+X_0\overline{Y}_0\overline{C}_0+X_0Y_0C_0$

$C_{1(SP)} = \overline{X}_0Y_0C_0+X_0\overline{Y}_0C_0+X_0Y_0\overline{C}_0+X_0Y_0C_0$

ESERCIZI

Verificare la seguente identità:

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

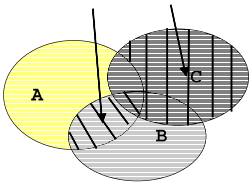
Altre tecniche:

- **confrontare la tabella delle verità,**
- **diagrammi di Venn**
- **confrontare le forme canoniche.**

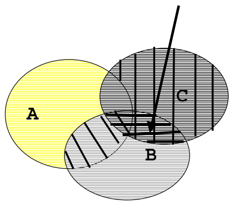
ESERCIZI

Diagrammi di Venn:

$\boldsymbol{A\bullet B+\overline{A}\bullet C}$

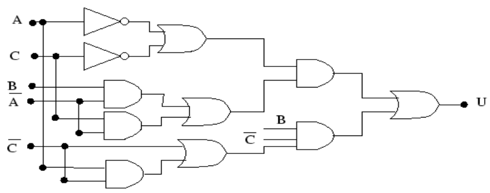


$\boldsymbol{A\bullet B+\overline{A}\bullet C+B\bullet C}$



ESERCIZI

Ottimizzare il seguente circuito logico:



Equivalente a:

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

ESERCIZI

Occorre semplificare la relazione:

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