FONDAMENTI DI INFORMATICA

Lezione n. 2

- **ALGEBRA BOOLEANA**
- **CIRCUITI LOGICI**
- **ELEMENTI PRIMITIVI**
- **QUALCHE ESERCIZIO CON ELEMENTI LOGICI**

In questa lezione sono ripresi i concetti principali di base dell'algebra booleana e introdotta la relazione tra quest'ultima e i circuiti logici.

ALGEBRA di BOOLE II

Un po' di storia:

- ***La logica greca*: le formule logiche sono espresse nel linguaggio comune e soddisfano le normali regole sintattiche.**
- ***La logica matematica*: si usa un linguaggio artificiale in cui parole e simboli hanno funzioni semantiche rigidamente definite.**
- **G.Boole: Mathematical Analysis of Logic (1847)**
  - **La logica va collegata alla matematica e non alla metafisica.**
  - **Si costruisce un sistema formale e solo successivamente si assegna ad esso una interpretazione nel linguaggio comune.**

ALGEBRA BOOLEANA

La teoria dei circuiti logici considera come modello matematico di supporto la *Algebra Booleana*.

George Boole
matematico inglese (1815-1864)

L'algebra Booleana consiste di:

- **un insieme di elementi K**
- **due funzioni {+, · } che fanno corrispondere a una qualsiasi coppia di elementi di K un elemento di K**
- **Una funzione {¯}**

Tali per cui valgono un insieme di assiomi:

(segue)

ASSIOMI DELL'ALGEBRA BOOLEANA

1. **K contiene al minimo due elementi *a* e *b* tali che *a* ¹ *b*.**
2. ***Chiusura*: Per ogni *a* e *b* in K: *a*+*b* Ì K e *a*• *b* Ì K .**
3. ***Proprietà commutativa:*** $a + b = b + a$ **e** $a \bullet b = b \bullet a$
4. ***Proprietà associativa:*** $(a + b) + c = a + (b+c) = a + b + c$
   e $a \bullet (b \bullet c) = (a \bullet b) \bullet c = a \bullet b \bullet c$
5. ***Indentità:***
   - **Esiste un elemento identità rispetto a {+}, tale che *a* + 0 = *a* per ogni *a* Ì K**
   - **Esiste un elemento identità rispetto a {• }, tale che *a* • 1 = *a* per ogni *a* Ì K**

ASSIOMI DELL’ALGEBRA BOOLEANA

6. *Proprietà distributiva:*

- $a + (b \bullet c) = (a + b) \bullet (a + c)$
- $a \bullet (b + c) = (a \bullet b) + (a \bullet c)$

7. *Complemento*: Per ogni *a* Ì K esiste un elemento

$\overline{a}$ Ì K tale che:

- $(a + \overline{a}) = 1$
- $(a \bullet \overline{a}) = 0$

ALGEBRA BOOLEANA

E’ possibile verificare che:

- **l’insieme K = { 0, 1 }**
- **Le operazioni logiche OR, AND, NOT come definite dalle tabelle sottostanti**

soddisfano tutti gli assiomi dell’algebra booleana.

A	$f(A)=\overline{A}$
1	0
0	1

Inversione

A	B	f(A,B)=A+B
0	0	0
0	1	1
1	0	1
1	1	1

OR
somma logica

A	B	f(A,B)=A• B
0	0	0
0	1	0
1	0	0
1	1	1

AND
prodotto logico

ALTRI OPERATORI ELEMENTARI

E' possibile definire altre funzioni od operatori elementari.

A	B	f(A,B)=AÅ B
0	0	0
0	1	1
1	0	1
1	1	0

EX-OR

A	B	$f(A,B)=\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

NAND

A	B	$f(A,B)=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

NOR

FUNZIONI BOOLEANA

Oltre alle funzioni OR e AND esistono altre funzioni a due variabili.
Le funzioni a due variabili sono complessivamente 16

X	Y	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
			AND		X			+ aritm o XOR	OR	NOR						NAND	

ALGEBRA BOOLEANA

L'algebra booleana opera su variabili che possono solo assumere valore 0 o 1.
Una funzione logica:

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

È una legge che fa corrispondere a ogni combinazione di valori delle variabili indipendenti

$$x_1, x_2, \ldots, x_n$$

uno e un solo valore binario della variabile *z*.

PROPRIETA'

N.	PROPRIETA'	NOME
1	**Esistono gli elementi 0,1 Î K tali che :** **A + 0 = A** **A · 1 = A**	**Esistenza elementi identità**
2	**A + B = B + A** **A · B = B · A**	**Proprietà commutativa**
3	**A · (B + C) = A · B + A · C** **A + (B · C) = (A + B) · (A + C)**	**Proprietà distributiva**
4	**Per ogni A Î K, esiste $\overline{A}$ tale che:** **A · $\overline{A}$ = 0 e A + $\overline{A}$ = 1**	**Esistenza dell'inverso**

segue

PROPRIETA'

N.	PROPRIETA'	NOME
5	**$A + (B + C) = (A + B) + C$** **$A (BC) = (AB) C$**	**Proprietà associativa**
6	**$A + A = A$** **$A \cdot A = A$**	**Proprietà dell'idempotenza**
7	**$\overline{A + B} = \overline{A} \cdot \overline{B}$** **$\overline{A \cdot B} = \overline{A} + \overline{B}$**	**Legge di DeMorgan**
8	**$\overline{\overline{A}} = A$**	**Involuzione**

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

PROPRIETA' III

Le tabelle precedenti riportano le proprietà principali dell'algebra booleana. La verifica di tali proprietà può avvenire in tutti i casi mediante:

- **analisi esaustiva**
- **utilizzando le proprietà riportate in precedenza.**

PRINCIPIO DI DUALITA'

- **Tutte le proprietà precedenti sono espresse a coppie.**
- **Una proprietà è derivabile dall'altra con lo scambio di:**
  - **0 « 1**
  - **+ « •**

TEOREMA DI DE MORGAN*

E' possibile dimostrare il seguente teorema:

$$\overline{X_1 + X_2 + \dots X_n} = \overline{X}_1 \bullet \overline{X}_2 \bullet \dots \bullet \overline{X}_n$$

Dimostrazione per induzione :

- $\overline{X_1 + X_2} = \overline{X}_1 \bullet \overline{X}_2$ **verifica tabellare**
- $\overline{(X_1 + X_2 + \dots X_n) + X_{n+1}} = \overline{(X_1 + X_2 + \dots X_n)} \bullet \overline{X}_{n+1}$

$$= (\overline{X}_1 \bullet \overline{X}_2 \bullet \dots \overline{X}_n) \bullet \overline{X}_{n+1}$$

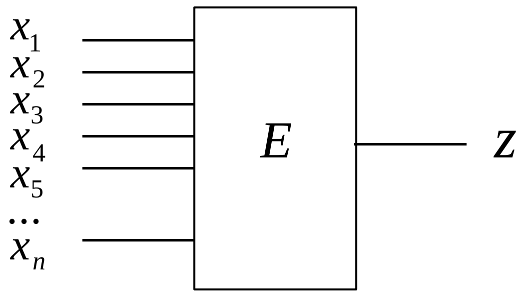
- **Versione duale** $\overline{X_1 \bullet X_2 \bullet \dots \bullet X_n} = \overline{X}_1 + \overline{X}_2 + \dots \overline{X}_n$

DeMorgan (1806 - 1871)

PORTE LOGICHE

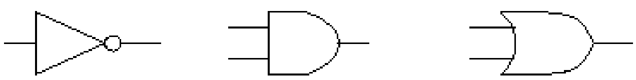
- **Espressione logica generale**

$$z = E(x_1, x_2, ..., x_n)$$

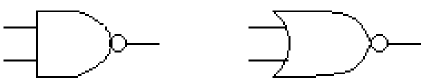


PORTE LOGICHE

- **Una porta logica è un blocco elementare in cui la relazione fra variabili di ingresso e variabili di uscita è descritta da funzioni booleane.**



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



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

ELEMENTI PRIMITIVI

- **Una qualunque funzione logica può essere realizzata utilizzando un numero limitato di funzioni elementari (insieme completo).**
- **Sono insiemi completi:**
  NAND, NOR, (AND,NOT), (OR,NOT)
- **Questa proprietà garantisce che l'uso di un limitato numero di funzioni consente di rappresentare una qualunque funzione logica.**
- **Utilizzando un numero limitato di tipi di porte logiche è possibile realizzare un circuito che realizza una qualsiasi funzione logica.**

FONDAMENTI DI INFORMATICA

Esercizio n. 1

La polizza n. 22 può essere stipulata se l'assicurando:

1. **Ha stipulato la polizza n. 19 ed è un maschio sposato**

oppure 2. **Ha stipulato la polizza n. 19 ed è sposato e ha meno di 25 anni.**

oppure 3. **Non ha stipulato la polizza n. 19 ed è una donna sposata.**

oppure 4. **E' un maschio con meno di 25 anni.**

oppure 5. **È sposato con 25 o più anni.**

E' possibile semplificare questa sequenza di regole ?

Esercizio n. 1

Le frasi precedenti possono essere trasformate:

La polizza n. 22 può essere stipulata se l'assicurando:

1. **Ha stipulato la polizza n. 19 AND è un maschio AND è sposato**

OR 2. **Ha stipulato la polizza n. 19 AND è sposato AND ha meno di 25 anni.**

OR 3. **Non ha stipulato la polizza n. 19 AND è una donna AND è sposata.**

OR 4. **E' un maschio AND ha meno di 25 anni.**

OR 5. **È sposato AND ha 25 o più anni.**

SOLUZIONE ESRCIZIO N. 1

A = l'assicurando è coperto dalla polizza n. 19

B = l'assicurando è sposato

C = l'assicurando è maschio

D = l'assicurando ha meno di 25 anni.

1. $A \bullet B \bullet C$ 2. $A \bullet B \bullet D$
3. $\overline{A} \bullet B \bullet \overline{C}$ 4. $C \bullet D$
5. $B \bullet \overline{D}$

SOLUZIONE ESRCIZIO N. 1

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

$$A \cdot B \cdot C + A \cdot B \cdot D + B \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} + C \cdot D = \qquad \cdot (A+1)$$

$$A \cdot \cancel{B} \cdot C + A \cdot B + B \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} + C \cdot D =$$

$$A \cdot B + B \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} + C \cdot D = \qquad \cdot (\overline{C}+1)$$

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

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

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

$$A \cdot B + B + C \cdot D = B + C \cdot D$$

SOLUZIONE ESRCIZIO N. 1

A = l'assicurando ha stipulato la polizza n. 19

B = l'assicurando è sposato

C = l'assicurando è maschio

D = l'assicurando ha meno di 25 anni.

Soluzione $\Longrightarrow$ $B + C \cdot D$

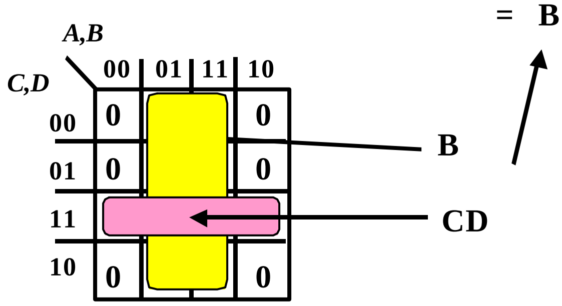
La polizza n. 22 può essere stipulata se l'assicurando:

1. E' sposato

oppure 2. E' un maschio con meno di 25 anni.

SOLUZIONE ESRCIZIO N. 1

$$A \bullet B \bullet C + A \bullet B \bullet D + \overline{A} \bullet B \bullet \overline{C} + C \bullet D + B \bullet \overline{D} =$$
$$= B + CD$$



L'uso delle mappe consente di trovare facilmente la soluzione ottima.

Esercizio n. 2

Cinque astronauti A,B,C,D,E sono pronti per una missione spaziale. Per definire la lista di coloro che partiranno è necessario soddisfare tutte le seguenti relazioni:

a) A o B devono necessariamente essere inclusi nella lista, ma non contemporaneamente;

b) C o E devono essere inclusi nella lista, anche contemporaneamente;

c) qualora D sia incluso nella lista anche B deve essere incluso nella lista;

d) A e C possono essere entrambi inclusi o entrambi esclusi dalla lista;

e) qualora E sia incluso anche C e D lo devono essere.

SOLUZIONE ESERCIZIO N. 2

A, B, C, D, E sono variabili indipendenti che assumono valore 1 se l'astronauta corrispondente viene prescelto e valore 0 in caso opposto.

Ognuna delle 5 condizioni a, b, c, d, e deve essere verificata indipendentemente e assume valore 1 se i valori assegnati alle variabili indipendenti soddisfano ai vincoli imposti e valore 0 se non soddisfano.

a) A o B devono necessariamente essere inclusi nella lista, ma non contemporaneamente;

a = a(A,B) = $\overline{A}B + A\overline{B}$ **(a=1 se A=1 e B=0 oppure A=0 e B=1)**

b) C o E devono essere inclusi nella lista, anche contemporaneamente;

b = b(C,E) = C + E (b=1 se C=1 oppure E =1)

SOLUZIONE ESERCIZIO N. 2

c) qualora D sia incluso nella lista anche B deve essere incluso nella lista;

c = c(D,B)= $DB + \overline{D}$ **(c=1 se D=0 oppure D=1 e B=1)**

d) A e C possono essere entrambi inclusi o entrambi esclusi dalla lista;

d=(A,C)= $AC + \overline{A}\,\overline{C}$ **(d=1 se A=1 e C=1 oppure A=0 e C=0**

e) qualora E sia incluso anche C e D lo devono essere.

e=(E,C,D) = $ECD + \overline{E}$ **(e=1 se E=0 oppure E=1,C=1,D=1)**

SOLUZIONE ESERCIZIO N. 2

Z= (A'B + AB') (C+E) (DB+D') (AC+A'C') (ECD+E') =

(A'BC'+AB'C) (C+E) (DB+D') (ECD+E') =

(A'BC'+AB'C) (CDE+CE') (DB+D') =

(A'BC'D+A'BC'D'+AB'CD') (CDE+CE') =

AB'CD'E'

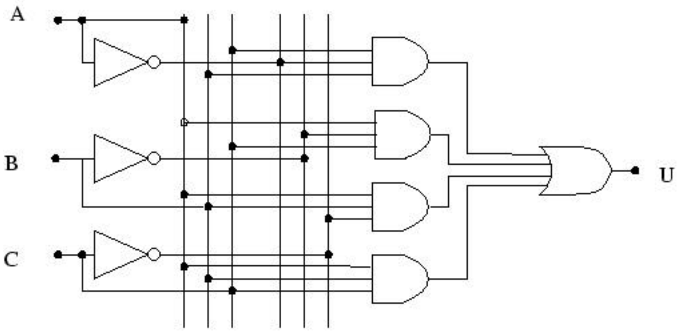
Esercizio n. 3

Progettare un circuito logico a 3 ingressi *(A,B,C)* e una uscita *U* che assuma valore 1 quando, all'ingresso, il numero degli 1 supera il numero degli 0. (Circuito di rilevazione di maggioranza)

A	B	C	U	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC$
1	0	0	0	
1	0	1	1	$A\overline{B}C$
1	1	0	1	$AB\overline{C}$
1	1	1	1	ABC

$$U = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

SOLUZIONE ESERCIZIO 3



$$U = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

SOLUZIONE ESERCIZIO 3

Soluzione alternativa:

A	B	C	U
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

$A + B + C$

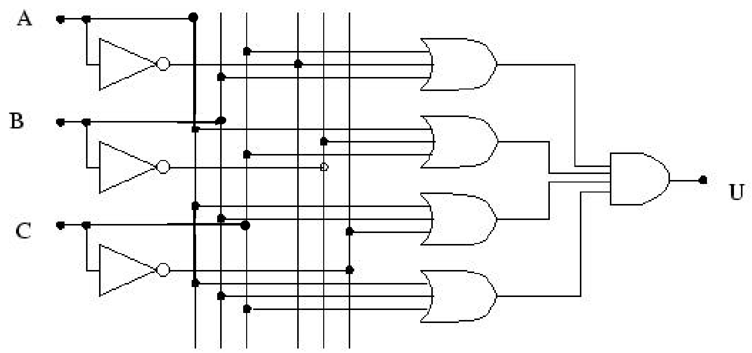
$\longrightarrow A + B + \overline{C}$

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

$\overline{A} + B + C$

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

SOLUZIONE ESERCIZIO 3



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

UNIVERSITA’ DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell’Informazione

COMPUTER ENGINEERING

Esercizio n. 4

Progettare un circuito che accetti in ingresso numeri da 0 a 15.
L'uscita deve assumere valore 1 quando il numero è primo.

Soluzione:
- **Rappresentare i numeri secondo una codifica binaria (4 bit).**
- **Scrivere la tabella della funzione.**
- **Inserire tante porte AND quanti sono gli 1 della funzione.**
- **Inserire una porta OR per generare l'uscita.**