Architetture dei Calcolatori

Introduzione alle Reti Logiche

Prof. Francesco Lo Presti

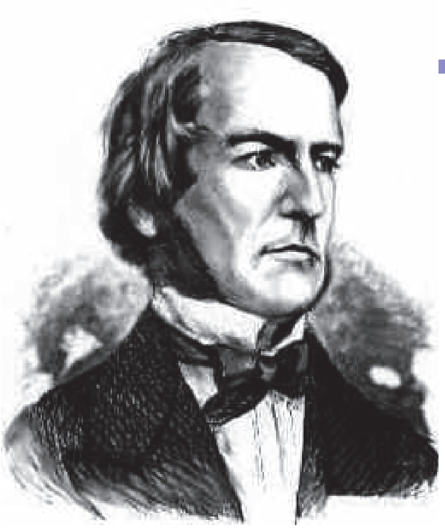
Logica Digitale

- Circuiti Logici
  - 2 livelli logici (logica binaria): 0, 1
  - Voltaggio alto/basso
- Porte logiche di base
  - AND, OR, NOT
- Tipologie di Circuiti Logici
  - Combinatori (senza memoria, senza stato)
    - ✓ Rete combinatoria
    - ✓ Implementa funzione booleana
  - Sequenziali (con memoria, comportamento dipendente dallo stato)
    - ✓ Rete sequenziale
    - ✓ Implementa un automa a stati finiti

Algebra Booleana

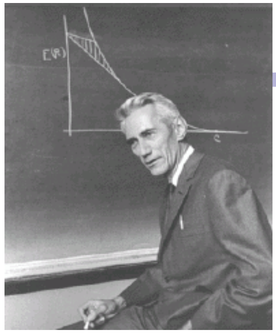
- Variabili: A, B, …
- Dominio delle variabili: 2 valori
  - 1 / 0; Y / N; true / false; …
- Operazioni Fondamentali
  - AND, OR, NOT
- …e loro significato
  - AND: entrambi gli ingressi sono veri
  - OR: almeno un ingress è vero
  - NOT: nega l'ingresso
- Prende il nome da George Boole

George Boole (1815-1864)



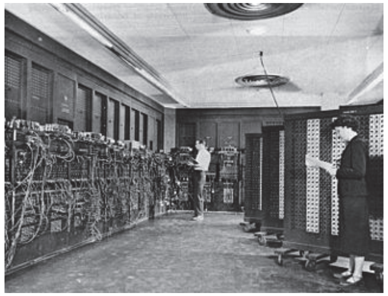
An Investigation of the Laws of Thought, on Which are founded the Mathematical Theories of Logic and Probabilities (1854)

Claude Shannon (1916-2001)

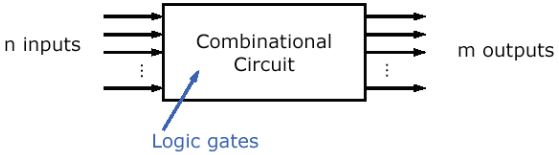


A Symbolic Analysis of Relay and Switching Circuits (1938)

ENIAC (Electronic Numerical Integrator And Calculator) (1946)



Rete Combinatoria



- Ognuna delle m uscite può essere epsressa come funzione degli n ingressi

Definizione di Funzione Booleana (1)

- Tabelle della verità "truth tables"
  - Rappresentazione esplicita del valore della funzione (uscita/output) per ogni possible configurazione degli ingressi

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

A	NOT
0	1
1	0

Esempio di tabella della verità per una funzione a 3 variabili

f(A,B,C)= 1 se e solo se almeno due ingressi sono uguali a 1

A	B	C	f(A,B,C)
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

Definizione di Funzione Booleana (2)

- Tramite espressioni booleane
- Un espressione booleana consiste di:
  - variabili
  - costanti 0 and 1
  - Operatori booleani (AND, OR, NOT)
  - parentesi

M = OR(AND(NOT(A),NOT(B)), AND(A,B))

Funzioni Booleane

- Convenzioni
  - NOT (negazione): NOT(A) = A' = $\overline{A}$
  - AND (congiunzione): AND(A,B) = AB = A•B
  - OR (disgiunzione): OR(A,B) = A+B

M = OR(AND(NOT(A),NOT(B)), AND(A,B))

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

Precedenza tra Operatori Booleani

- Ordine di valutazione delle espressioni booleane:
  1. Parentesi
  2. NOT
  3. AND
  4. OR
- Esempio: $F=A(B+C)(C+\overline{D})$

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

$M = \overline{A}\,\overline{B} + AB$

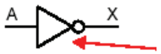
Identità Booleane

Principio di Dualità: uguaglianza algebrica rimane vera se gli operatori OR e AND, e gli elementi 0 o 1 sono scambiati tra loro

$1A = A$	$0+A = A$	Identità
$0A = 0$	$1+A = 1$	Elemento Nullo
$AA = A$	$A+A = A$	Idempotente
$A\overline{A} = 0$	$A+\overline{A} = 1$	Inverso
$AB = BA$	$A+B = B+A$	Commutativa
$(AB)C = A(BC)$	$(A+B)+C = A+(B+C)$	Associativa
$A+BC = (A+B)(A+C)$	$A(B+C) = AB+AC$	Distributiva
$A(A+B) = A$	$A+AB = A$	Assorbimento
$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A+B} = \overline{A}\,\overline{B}$	De Morgan

Porta NOT

- Porta NOT – inverte il segnale

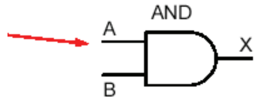


Se A è 0, X è 1
Se A è 1, X è 0

- Detta anche inverter

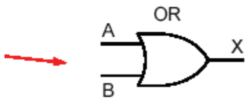
Porta AND

- Uscita è 1 se tutti gli ingressi sono pari a 1
- Porta AND a due ingressi



Porta OR

- Uscita è 1 se almeno uno degli ingressi è pari a 1
- Porta OR a due ingressi



Porta NAND - NOT AND

- L'uscita è 0 se entrambi gli ingressi sono pari a 1

- Diagramma Logico

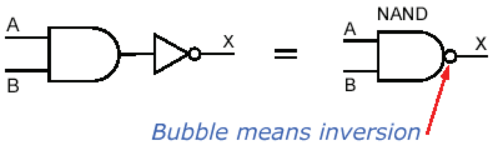


Tabella della verità

A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Porta NOR - NOT OR

- L'uscita è 0 se almeno un ingresso è pari a 1
- Diagramma Logico

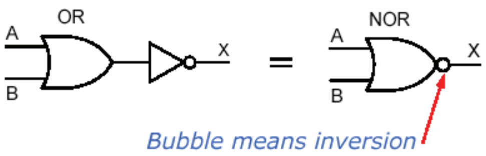


Tabella della verità

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Porta XOR – OR esclusivo

- 2 ingressi: L'uscita è 1 se esattamente un ingresso è pari a 1
- Diagramma Logico

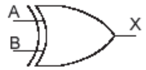


Tabella della verità

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

- Per più di 2 ingressi: l'uscita è 1 se un numero dispari di ingressi è pari a 1

Porte (Gate) universali

- Quante funzioni logiche esistono con n ingressi?
- Con n ingressi, ci sono $2^{(2^n)}$ funzioni logiche diverse
- Caso n=2

A	B	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
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

Porte Universali (2)

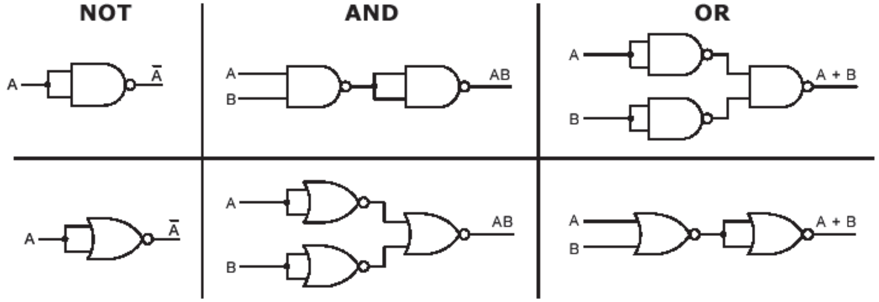
- AND, OR, NOT possono realizzare tutte le possibili funzioni booleane (discusso in seguito)
- E' possibile usare un numero inferiore di tipi di porte?
- Porta universale: una porta logica che può implementare ogni funzione booleana

Porte universali(3)

- Le porte AND, NOT sono sufficienti!
- Le porte OR, NOT sono sufficienti!
- Le sole porte NAND sono sufficienti!
- Le sole porte NOR sono sufficienti!

Equivalenza fra porte

- Le porte AND, OR, NOT possono essere implementate usando solo porte NAND o solo porte NOR



…tornando all'espressione precedente…

- Circuito a due ingressi
- Uscita è 1 se i due ingressi sono uguali
  - (i.e., entrambi 0 o entrambi 1)
- Funzione Booleana:

  $M = \bar{A}\,\bar{B} + AB$

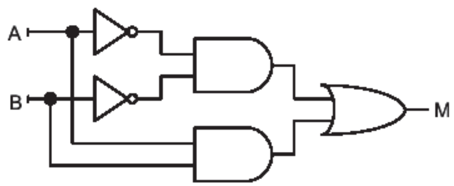
Tabella di Verità

A	B	M
0	0	1
0	1	0
1	0	0
1	1	1

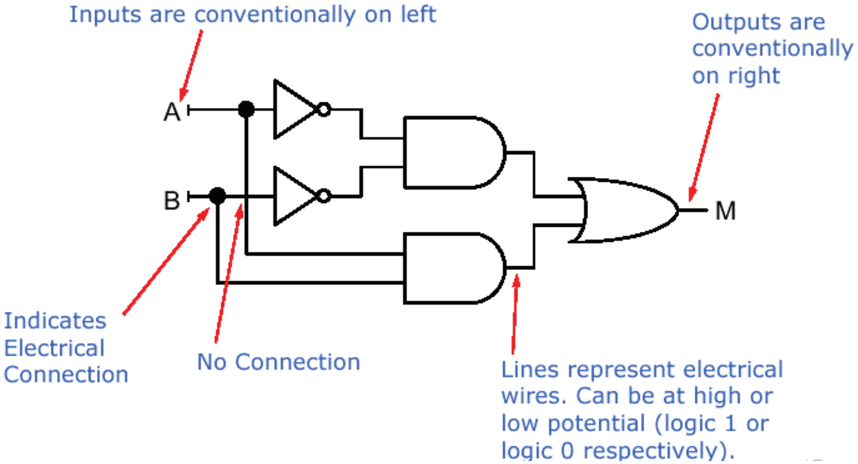
Definizione (e realizzazione) di Funzione Booleana (3)

- Tramite circuito logici
  - Combinazione di porte logiche tra loro collegate

A	B	M
0	0	1
0	1	0
1	0	0
1	1	1



Convenzioni per i circuiti logici



Esercizio

- Scrivere la tabella di verità e il circuito logico che implementa la seguente espressione booleana:

$F = X + \overline{Y}Z$

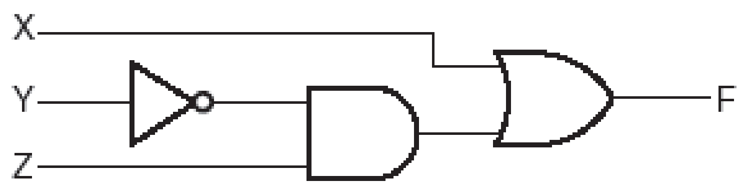
Esercizio – Tabella di Verità

$F = X + \overline{Y}Z$

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Esercizio – Circuito Logico

$F = X + \overline{Y}Z$



Conversioni …

- Circuito Logico ->
  - -> Espressione Boolena (ispezione visiva da sinistra a destra)
  - -> Tabella di verità (calcolo esplicito caso per caso)
- Espressione booleana->
  - -> Circuito (costruzione bottom-up)
  - -> Tabella di verità (calcolo esplicito caso per caso)
- Tabella di verità ->
  - -> Circuito Logico (tramite espressione booleana)
  - -> Espressione booleana (tramite le forme canoniche)

Rappresentazioni SOP e POS

- E' conveniente specificare un espressione booleana come …
- Somma di prodotti (SOP – Sum Of Products)
  - Combinazione OR di combinazioni AND degli ingressi
  - $F_1 = \overline{A}\,B + BC$ è una rappresentazione SOP
  - $F_2 = \overline{A}\,B + \overline{BC}$ non è una rappresentazione SOP
- Prodotti di somme (POS – Products Of Sums)
  - Combinazione AND di combinazioni OR degli ingressi
  - $F_1 = \overline{A}\,(B + \overline{C})(B+A)$ è una rappresentazione POS
  - $F_2 = \overline{A}\,B + BC$ non è una rappresentazione POS

Forma canonica delle funzioni booleane

- E' un formato "standard" di rappresentare SOP o POS
  - SOP: somma di mintermini
  - POS: prodotto di maxtermini
- Un mintermine è un prodotto in cui appaiono tutte le variabili, in forma affermata o negata
- Un maxtermine è una somma in cui appaiono tutte le variabili, in forma affermata o negata
- Esempi (con variabili A, B e C):

  $F = (\overline{A} + B + C)\,(A + \overline{B} + C)$ è in forma canonica POS

  $M = AB + \overline{A}\,BC$ non è in forma canonica SOP

Mintermini

- Dato che ogni variabile appare affermata (e.g. X) o negata (e.g. $\overline{X}$), ci sono 2^n possibili mintermini per *n* variabili
- Esempio: A due variabili (X e Y) corrispondono i seguenti 4 mintermini:
  - XY
  - $\overline{X}Y$
  - $X\overline{Y}$
  - $\overline{X}\,\overline{Y}$

Maxtermini

- Dato che ogni variabile appare affermata (e.g. X) o negata (e.g. $\overline{X}$), ci sono 2^n possibili maxtermini per *n* variabili
- Esempio: A due variabili (X e Y) corrispondono i seguenti 4 maxtermini:
  - $X + Y$
  - $X + \overline{Y}$
  - $\overline{X} + Y$
  - $\overline{X} + \overline{Y}$

Maxtermini e Mintermini: Indice

- Esempio: mintermini e maxtermini con 2 variabili

Indice	Mintermine	Maxtermine
0	$\bar{X}\bar{Y}$	X+Y
1	$\bar{X}$Y	X+$\bar{Y}$
2	X$\bar{Y}$	$\bar{X}$+Y
3	XY	$\bar{X}$+$\bar{Y}$

- L'indice di un mintermine (maxtermine) viene usato per descrivere quali variabili appaiono affermate o negate nel mintermine (maxtermine)

Ruolo dell'indice

- L'indice di un mintermine o maxtermine, espresso come nunero binario, è usato per determinare se le variabili, secondo un'ordine standard (e.g., alfabetico), appaiono affermate o negate nel rispettivo mintermine o maxtermine
- Mintermini:
  - "1" indica che la variabile appare affermata
  - "0" indica che la variabile appare negata
- Maxtermini:
  - "0" indica che la variabile appare affermata
  - "1" indica che la variabile appare negata

Mintermini e indici

Indice	Codifica indice			Mintermine	Simbolo
	X	Y	Z		
0	0	0	0	$\bar{X}\bar{Y}\bar{Z}$	m_0
1	0	0	1	$\bar{X}\bar{Y}Z$	m_1
2	0	1	0	$\bar{X}Y\bar{Z}$	m_2
3	0	1	1	$\bar{X}YZ$	m_3
4	1	0	0	$X\bar{Y}\bar{Z}$	m_4
5	1	0	1	$X\bar{Y}Z$	m_5
6	1	1	0	$XY\bar{Z}$	m_6
7	1	1	1	XYZ	m_7

Maxtermini e indici

Indice	Codifica indice			Maxtermine	Simbolo
	X	Y	Z		
0	0	0	0	X+Y+Z	M_0
1	0	0	1	$X+Y+\bar{Z}$	M_1
2	0	1	0	$X+\bar{Y}+Z$	M_2
3	0	1	1	$X+\bar{Y}+\bar{Z}$	M_3
4	1	0	0	$\bar{X}+Y+Z$	M_4
5	1	0	1	$\bar{X}+Y+\bar{Z}$	M_5
6	1	1	0	$\bar{X}+\bar{Y}+Z$	M_6
7	1	1	1	$\bar{X}+\bar{Y}+\bar{Z}$	M_7

Relazione fra mintermni e maxtermini

- Teorema di DeMorgan $\overline{x \cdot y} = \bar{x} + \bar{y}$ $\overline{x + y} = \bar{x} \cdot \bar{y}$
- Esempio a 2 variabili: $M_2 = \bar{x} + y$ $m_2 = x \cdot \bar{y}$
- M_2 è il complemento di m_2 e viceversa.
- Dato che il teorema di DeMorgan's vale per n variabili, la proprietà vale in generale:

$$M_i = \overline{m_i} \qquad m_i = \overline{M_i}$$

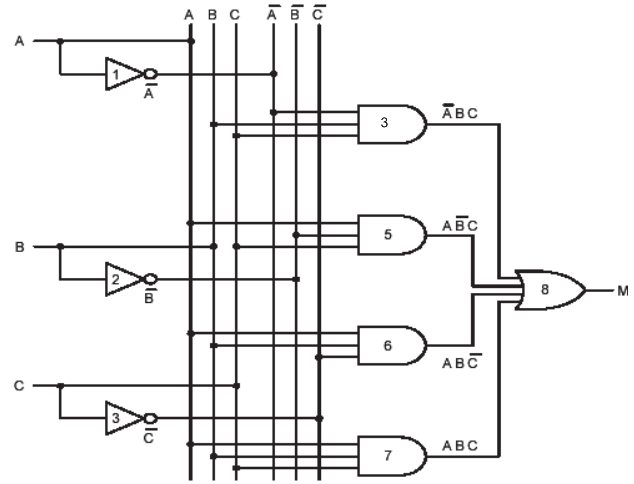
- M_i è il complemento di m_i viceversa.

Implementazione funzioni booleane

- Ogni funzione booleana può essere implementata come OR di combinazioni di AND degli ingressi
  - *SOP canonici*
- Dalla tabella di verità
  - *Per ogni valore 1 della funzione scrivere gli ingressi in AND*
    - ✓ Affermati se la variabile vale 1
    - ✓ Negati se la variabile vale 0
  - *Combinare questi in OR*
- $M=\bar{A}BC+A\bar{B}C+AB\bar{C}+ABC$
  $= m_3+m_5+m_6+m_7$
- M=combinazione OR di mintermini

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

Circuito Logico Equivalente



Teorema di Shannon (1)

- Data una funzione booleana $f(X_1,X_2,\ldots,X_n)$ vale la seguente uguaglianza:

$$f(X_1,X_2,\ldots,X_n) = X_1 \cdot f(1,X_2,\ldots,X_n) + \overline{X_1} \cdot f(0,X_2,\ldots,X_n)$$

Teorema di Shannon: caso a due variabili

- Dato $f(X_1,X_2)$ si ha:

$$\begin{aligned} f(X_1,X_2) &= X_1 \cdot f(1,X_2) + \overline{X_1} \cdot f(0,X_2) = \\ &= X_1 \cdot [X_2 \cdot f(1,1) + \overline{X_2} \cdot f(1,0)] + \\ &\quad \overline{X_1} \cdot [X_2 \cdot f(0,1) + \overline{X_2} \cdot f(0,0)] = \\ &= X_1 X_2 \cdot f(1,1) + X_1 \overline{X_2} \cdot f(1,0) + \\ &\quad \overline{X_1} X_2 \cdot f(0,1) + \overline{X_1}\,\overline{X_2} \cdot f(0,0) \end{aligned}$$

  - $f(1,X_2)=g(X_2)$
  - $f(0,X_2)=h(X_2)$

Teorema di Shannon e implementazioni funzioni booleane – SOP canonici

- …
- Dalla tabella di verità
  - Per ogni valore 1 della funzione scrivere gli ingressi in AND
    - ✓ Affermati se la variabile vale 1
    - ✓ Negati se la variabile vale 0
  - Combinare questi in OR

X_1	X_2	$F(X_1,X_2)$
0	0	1
0	1	0
1	0	0
1	1	1

$$\begin{aligned} f(X_1,X_2) &= X_1 X_2 \cdot f(1,1) + X_1 \overline{X_2} \cdot f(1,0) + \\ &\quad \overline{X_1} X_2 \cdot f(0,1) + \overline{X_1}\,\overline{X_2} \cdot f(0,0) = \\ &= X_1 X_2 \cdot 1 + X_1 \overline{X_2} \cdot 0 + \\ &\quad \overline{X_1} X_2 \cdot 0 + \overline{X_1}\,\overline{X_2} \cdot 1 = \\ &= X_1 X_2 + \overline{X_1}\,\overline{X_2} \end{aligned}$$

Implementazione funzioni booleane (2)

- Ogni funzione booleana può essere implementata come l'AND di combinazioni in OR degli ingressi
  - POS canonici
- Dalla tabella di verità
  - Per ogni valore 0 della funzione scrivere gli ingressi in OR,
    - ✓ Affermati se la variabile vale 0
    - ✓ Negati se la variabile vale 1
  - Combinare questi in AND
- $F=(A+B+C)\cdot(A+B+\overline{C})\cdot(\overline{A}+B+C)\cdot(\overline{A}+B+\overline{C})\cdot(\overline{A}+\overline{B}+\overline{C})$
  $=M_0\cdot M_1\cdot M_4\cdot M_5\cdot M_7$
- F = una combinazione AND di maxtermini

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Teorema di Shannon (2)

- Data una funzione booleana $f(X_1,X_2,\dots,X_n)$ vale la seguente uguaglianza (duale di quella precedente):

$$f(X_1,X_2,\dots,X_n) = (X_1 + f(0,X_2,\dots,X_n)) \cdot (\overline{X_1} + f(1,X_2,\dots,X_n))$$

Teorema di Shannon – caso a due variabili

- Dato $f(X_1,X_2)$ si ha:

$$\begin{aligned} f(X_1,X_2) &= (X_1 + f(0,X_2)) \cdot (\overline{X_1} + f(1,X_2)) = \\ &= \{X_1 + [(X_2 + f(0,0)) \cdot (\overline{X_2} + f(0,1)]\} \cdot \\ &\quad \{\overline{X_1} + [(X_2 + f(1,0)) \cdot (\overline{X_2} + f(1,1)]\} = \\ &= (X_1 + X_2 + f(0,0)) \cdot (X_1 + \overline{X_2} + f(0,1)) \cdot \\ &\quad (\overline{X_1} + X_2 + f(1,0)) \cdot (\overline{X_1} + \overline{X_2} + f(1,1)) \end{aligned}$$

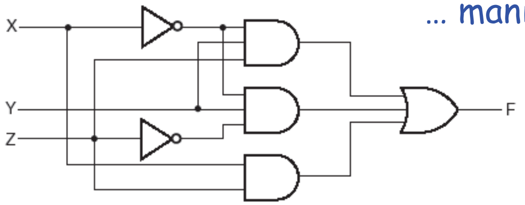
  - $f(1,X_2)=g(X_2)$
  - $f(0,X_2)=h(X_2)$

Algoritmi di Minimizzazione

- Mappe di Karnaugh
  - Impiegabili per funzioni boolane di 2-4 variabili
- Algortimo di Quine-McKluskey
- …

Ottimizzazione dei Circuiti

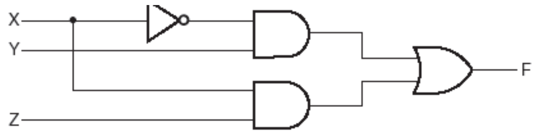
- Obiettivo: ottenere l'implementazione più semplice di una data funzione booleana
- Ottimizzazione è un approccio formale alla semplificazione, realizzata tramite un algoritmo
- Ottimizzazione richiede la definizione di una funzione costo per valutare la "semplicità" di un circuito
- L'ottimizzazione di circuiti a due livelli (SOP e POS) :
  - SOP
    - ✓ Minimizzare il numero di prodotti
    - ✓ Minimizzare il numero di variabili in ciascun prodotto
  - …e in modo simile nel caso POS



… manipolazione algebrica…

$$
\begin{aligned}
F &= \bar{X}YZ + \bar{X}Y\bar{Z} + XZ \\
&= \bar{X}Y(Z + \bar{Z}) + XZ \\
&= \bar{X}Y1 + XZ \\
&= \bar{X}Y + XZ
\end{aligned}
$$

…una espressione SOP più semplice da luogo ad un circuito più semplice



Mappe di Karnaugh (KM)

- Rappresentazione di una funzione booleana tramite una griglia/matrice
- Ogni cella rappresenta una combinazione dei valori in ingresso/mintermine
  - Può essere vista come un modo diverso di rappresentare la tabella di verità
- Combinazioni ingressi/Mintermini con una sola variabile diversa occupano celle adiacenti
- E' sufficiente considerare solo combinazioni per cui la funzione assume valore 1 (caso SOP)

Mappa di Karnaugh per funzioni a 2 variabili

- Mappa generica

X \ Y	0	1
0	$\bar{X}\bar{Y}$	$\bar{X}Y$
1	$X\bar{Y}$	XY

- Funzione F = XY

X \ Y	0	1
0		
1		1

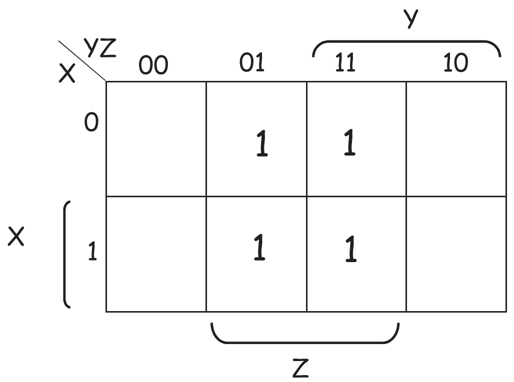
Funzione F = X + Y

X \ Y	0	1
0		1
1	1	1

Mappa di Karnaugh per funzioni a 3 variabili

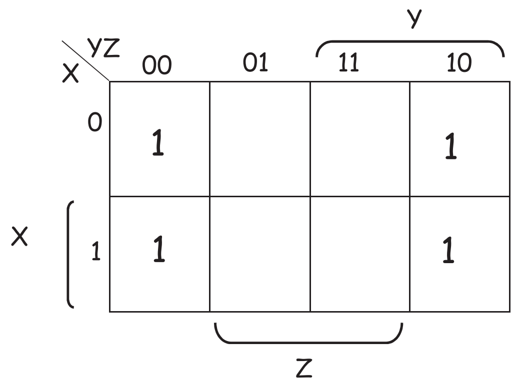
	X \ YZ	00	01	11 (Y)	10 (Y)
	0	$\bar{X}\bar{Y}\bar{Z}$	$\bar{X}\bar{Y}Z$	$\bar{X}YZ$	$\bar{X}Y\bar{Z}$
X	1	$X\bar{Y}\bar{Z}$	$X\bar{Y}Z$	XYZ	$XY\bar{Z}$
			Z	Z	

Esempio



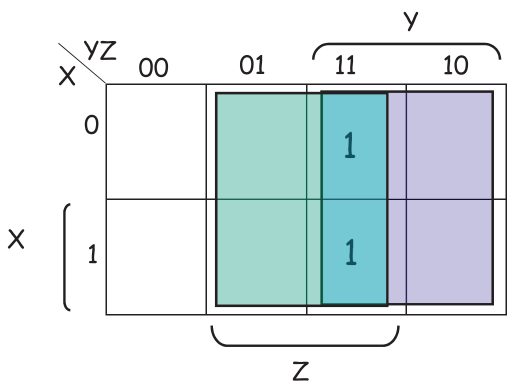
X	Y	Z	F=Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Esempio



X	Y	Z	F=$\bar{Z}$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

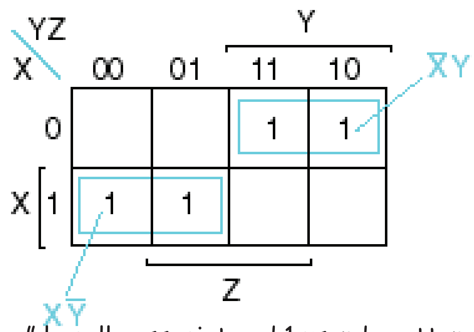
Esempio



X	Y	Z	F=YZ
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Esempio

- $F = \bar{X}YZ + \bar{X}Y\bar{Z} + X\bar{Y}Z + X\bar{Y}\bar{Z}$
  $= \bar{X}Y(Z + \bar{Z}) + X\bar{Y}(Z + \bar{Z})$
  $= \bar{X}Y + X\bar{Y}$



X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

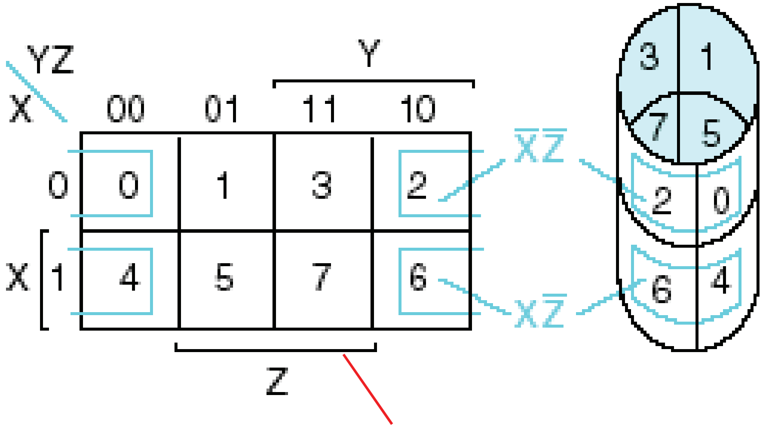
Idea:

- "Coprire" le celle associate ad 1 usando rettangoli (insiemi di celle adiacenti)
  - Ogni retangolo di 2^k celle adiacenti (k=0,1,2,…) rappresenta un termine prodotto
  - Più grandi i rettangoli più semplici i prodotti a loro associati

Combinare celle

- Combinando celle si riducono i le variabili che appaiono nel risultante termine
- Nelle Mappe a 3 variabili:
  - Una cella rappresente un mintermine a 3 variabili
  - Due celle "adiacenti" rappresentano un termine prodotto con due variabili
  - Quattro celle "adiacenti" rappresentano un termine prodotto con una sola variabile
  - Otto celle "adiacenti" rappresentano l'espressione sempre vera = 1.

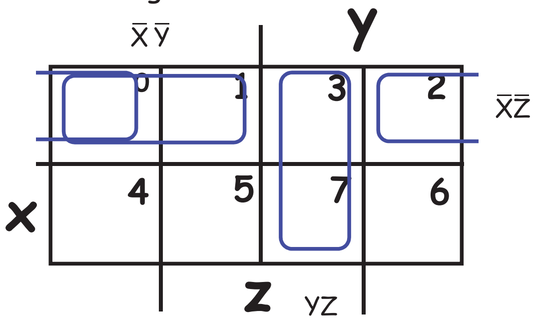
Adiacenze "Circolari" nella mappe a 3 variabili



Le etichette sono utili a determinare l'espressione associata ad un rettangolo

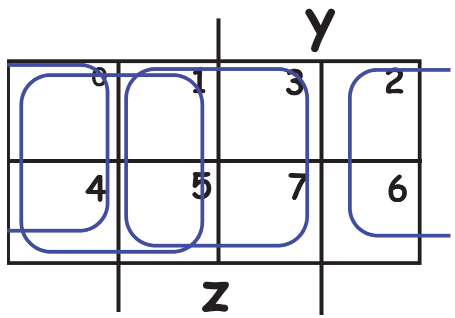
Mappe a 3 variabili

- Esempi di rettangoli di 2 celle:



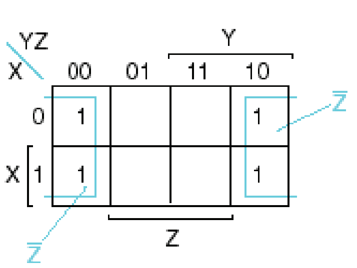
Mappe a 3 variabili

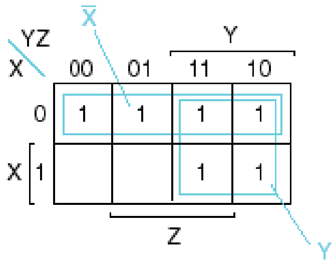
- Esempi di rettangoli di 4 celle:



Mappe a 3 variabili

- Esempi di rettangoli di 4 celle:

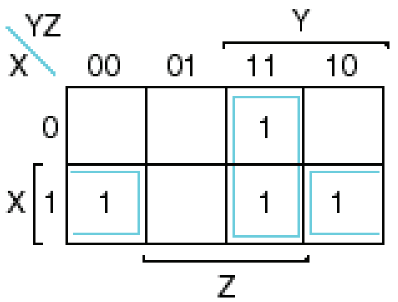




Esercizio (1)

- Qual'è l'espressione SOP minima per la funzione booleana $F_1=m_3+m_4+m_6+m_7$?

Soluzione

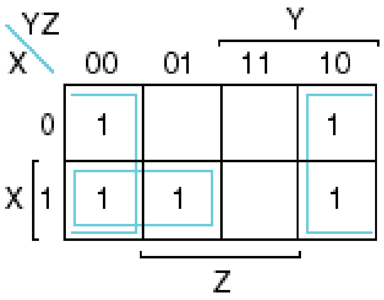


- $F_1 = YZ + X\,\overline{Z}$

Esercizio (2)

- Qual'è l'espressione SOP minima per la funzione booleana $F_2=m_0+m_2+m_4+m_5+m_6$?

Soluzione



- $F_2 = \overline{Z} + X\,\overline{Y}$

k-cubo

- E' un insieme di 2^k celle adiacenti
- 0-cubo, 1 cella, un mintermine
- 1-cubo, 2 celle adiacenti
- 2-cubo, 4 celle adiacenti
- 3-cubo, 8 celle adiacenti
- ....

Implicanti Primi

- $F = P_1 + P_2 + P_3 + \ldots$
- Il termine che corrisponde a un k-cubo è detto implicante
  - Infatti è un termine P_n che implica la funzione F,
  - i.e. se P_n è vero allora F è vero
- Un implicante è un implicante primo di F se non implica un altro implicante di F
- Un implicante primo corrisponde a un k-cubo di dimensione massima, ovvero un k-cubo nella mappa di Karnaugh che non è contenuto in un h-cubo di dimensioni maggiori (h>k)

Rappresentazioni Minime

$F = P_1 + P_2 + P_3 + \ldots$

è una rappresentazione minima SOP se:

1. Ogni P_i è un implicante primo
2. il numero di termini è minimo

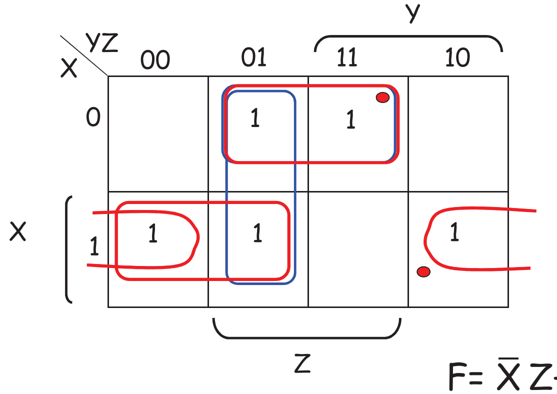
Procedura per il calcolo dell'espressione minima

1. Determinare i k-cubi di dimensione massima (implicanti primi)
2. Se un 1 è coperto da un solo k-cubo di dimensione massima, selezionare quell k-cubo
   - Il corrispondente implicante è detto essenziale
3. Selezionare il minimo numero di rimanti k-cubi che coprono gli 1 non coperti da implicanti primi essenziali

Esercizio (3)

- Calcolare l'espressione minima SOP per la funzione booleana

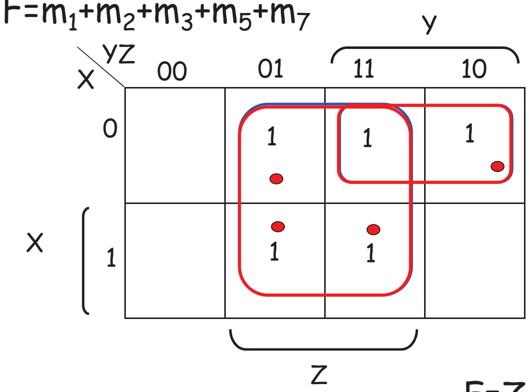
$$F=m_1+m_3+m_4+m_5+m_6$$



$F=\bar{X}Z+X\bar{Z}+X\bar{Y}$

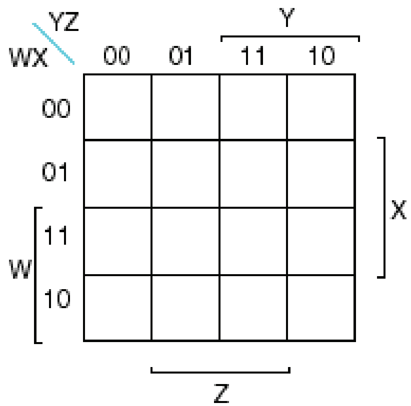
Esercizio (4)

- Calcolare l'espressione minima SOP
  $F=m_1+m_2+m_3+m_5+m_7$

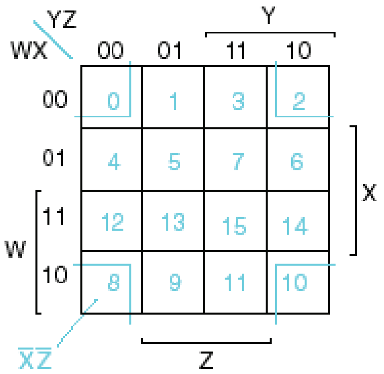


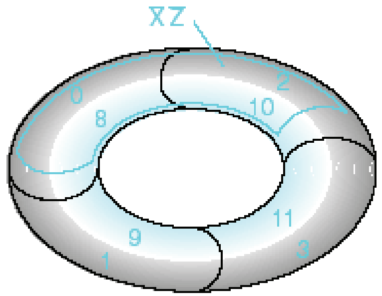
$F=Z+\bar{X}Y$

Mappe di Karnaugh a 4 variabili



Adiacenze circolari per le 4 variabili



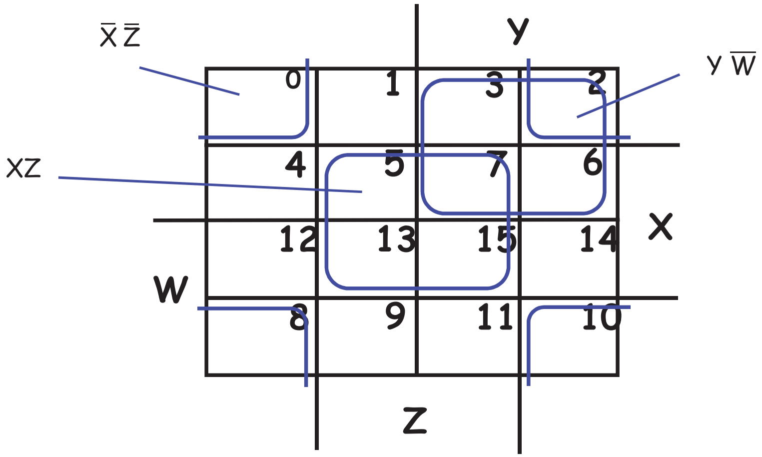


Termini a 4 variabili

- Mappe a 4 variabili sono caratterizzate da rettangoli/k-cubi corrispondenti a :
  - un singolo 1 = 4 variabili (un Mintermine)
  - due 1 = 3 variabili,
  - quattro 1 = 2 variabili
  - otto 1 = 1 variabili,
  - sedici 1 = 0 variabili (funzione costante "1")

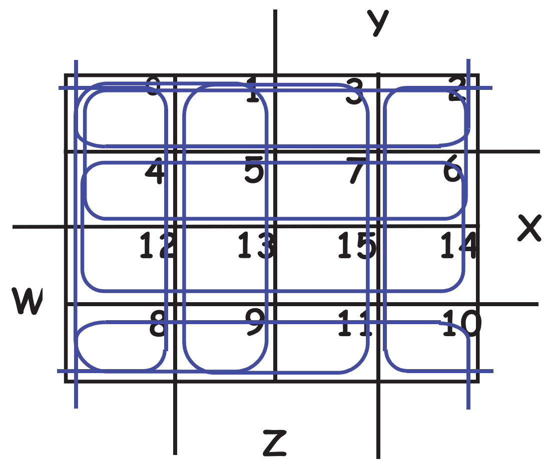
Mappe a 4 variabili

- Esempi di k-cubi/rettangoli



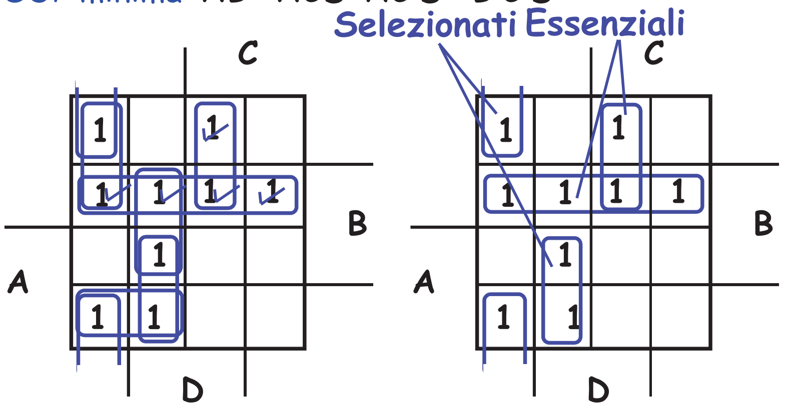
Mappe a 4 variabili

- Esempi di k-cubi/rettangoli



Esempio: Semplificare F(A, B, C, D) dato la mappa di Karnaugh

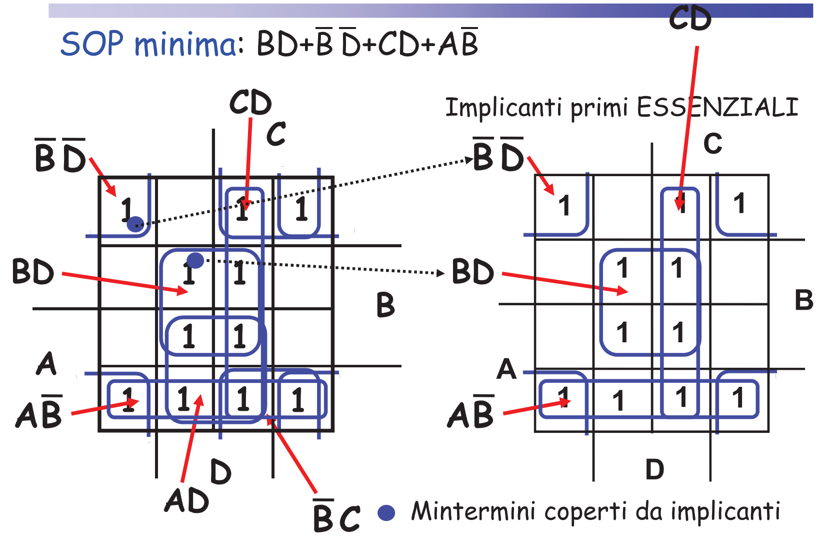
SOP minima: $\bar{A}B + \bar{A}CD + A\bar{C}D + \bar{B}\,\bar{C}\,\bar{D}$



✓ Mintermini coperti da implicanti primi essenziali

Esempio

SOP minima: $BD+\bar{B}\,\bar{D}+CD+A\bar{B}$



● Mintermini coperti da implicanti primi essenziali

"Don't Care" nelle Mappe di Karnaugh

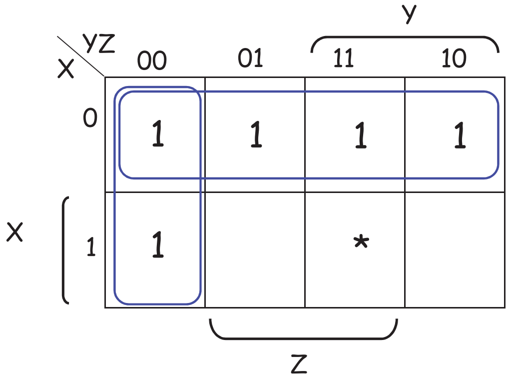
- Ci sono dei casi dove una tabela di verità o mappa di Karnaugh contiene righe/celle per le quali:
  - Le corrispondenti configurazioni in ingresso non occorrono
  - Il valore in uscita per quella particolare configurazione non sarà utilizzato
- In questi casi, il valore il valore in uscita corrispondente non ha bisogno di essere definito
  - Il valore in uscita è indicato come "don't care" (valore "x" o "*")
- Questo può essere sfruttato per minimizzare il circuito logico sostituendo a "x" il valore 0 o 1 in modo opportuno

Esempio

- Si consideri la seguente codifica a 3 bit X,Y, Z dei giorni della settimana:
  - Lunedì 000
  - Martedì 001
  - Mercoledì 010
  - Giovedì 011
  - Venerdì 100
  - Sabato 101
  - Domenica 110
- E la funzione booleana F(X,Y,Z)= X Y Z è un giorno feriale

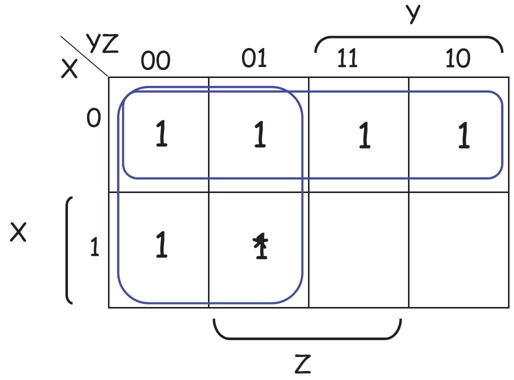
X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	*

Esempio



$$F = \overline{X} + \overline{Y}\,\overline{Z}$$

Esempio



$$F = \overline{X} + \overline{Y}$$