Prof. Mariacarla Staffa
a.a. 2022/2023

Laboratorio di Architettura Degli Elaboratori

Mappe di Karnaugh

Ricapitolando…

Funzioni Booleane

- Ciascuna variabile booleana può assumere uno dei due stati '1' o '0'. Due variabili prese insieme possono individuare $2^2 = 4$ stati. La variabile *A* può essere presente come *A* o !*A*. Una seconda variabile *B* può anch'essa essere presente come *B* o !*B* . Queste due variabili prese insieme possono dare luogo a
- quattro combinazioni: *AB*, *A!B* , *!AB*, *!A!B*. Assegnando 1 alla variabile vera e 0 alla variabile complementata possiamo riscrivere i quattro stati come 11,10,01,00.
- Tre variabili possono essere scritte in 2^3differenti combinazioni, dando luogo a 2^3 stati;
- *n* variabili danno luogo a 2^n stati.

Tabella delle Combinazioni

- La tabella delle combinazioni è un'elencazione sistematica di tutte le combinazioni che un gruppo di variabili binarie può assumere, ordinate secondo la sequenza numerica binaria.
- Date tre variabili *A*, *B*, *C*, sono
- possibili $2^3 = 8$ differenti combinazioni di queste variabili prese insieme.
- Di seguito è mostrata la tabella delle combinazioni delle tre variabili *A*, *B*, *C*

A	*B*	*C*	*Comb*
0	0	0	*!A !B !C*
0	0	1	*!A !B C*
0	1	0	*!A B !C*
0	1	1	*!A B C*
1	0	0	*A !B !C*
1	0	1	*A !B C*
1	1	0	*A B !C*
1	1	1	*ABC*

Tabella delle Combinazioni

- Una funzione booleana viene univocamente definita dalla sua tabella della verità. La tabella della verità è un'estensione della tabella delle combinazioni: a questa viene aggiunta sulla destra una colonna nella quale è indicato lo stato che la funzione assume in corrispondenza di ogni combinazione delle variabili, per tutte le combinazioni.
- Di seguito è mostrata la tabella della verità di una funzione *f* di tre variabili:

ABC	*f*
000	0
001	1
010	0
011	1
100	0
101	1
110	1
111	0

Funzioni Booleane

E' immediato scrivere l'espressione booleana della funzione f partendo dalla tabella della verità. La funzione è data dalla OR delle combinazioni (somma canonica) per le quali $f = 1$:

$$f = !A\ !B\ C + !A\ B\ C + A\ !B\ C + A\ B\ !C.$$

Il metodo appena illustrato mostra come si giunge alla sintesi delle funzioni booleane. L'espressione appena scritta è l'espressione primitiva della funzione f. Applicando i teoremi dell'algebra booleana è possibile scrivere l'espressione più semplice della funzione f.

I mintermini di una funzione vengono spesso identificati con il numero in base 10 corrispondente al valore binario del mintermine. (Es., a!b!c = m_4) e la funzione indicata come somma dei mintermini $f=\Sigma\ m_i$

Mappe di Karnaugh

Una più eloquente ed utile rappresentazione di una funzione booleana è data dalla mappa di Karnaugh.

Tutte le possibili combinazioni che un Gruppo di variabili può assumere sono rappresentate in forma di matrice nella mappa.

Mappe di Karnaugh

- La mappa di Karnaugh può essere usata per definire una funzione.
- In ogni cella corrispondente alla combinazione delle variabili per cui la funzione è vera si pone '1'; dove la funzione è falsa si pone '0'.
- Essa da una definizione del tutto equivalente a quella data dalla tabella della verità.

Minimizzazione di funzioni booleane

- Si può effettuare la minimizzazione delle funzioni booleane mediante manipolazioni algebriche, utilizzando i teoremi dell'algebra di Boole, oppure mediante l'elaborazione delle mappe di Karnaugh.

Minimizzazione delle funzioni booleane

- **La mappa di Karnaugh** è una disposizione ordinata di celle, che contengono le combinazioni delle variabili in modo che nel passare da una cella ad una contigua cambi lo stato di una sola variabile.
- La mappa contiene una cella per ogni combinazione delle variabili, in modo da esaurire tutte le combinazioni possibili.
- Una mappa di 2 variabili deve contenere 4 celle, perché vi sono 2^2 combinazioni differenti delle due variabili. Una mappa di tre variabili deve contenere 2^3 celle; una mappa di *n* variabili deve contenere 2^n celle.
- Raffiguriamo la mappa di Karnaugh di tre variabili.

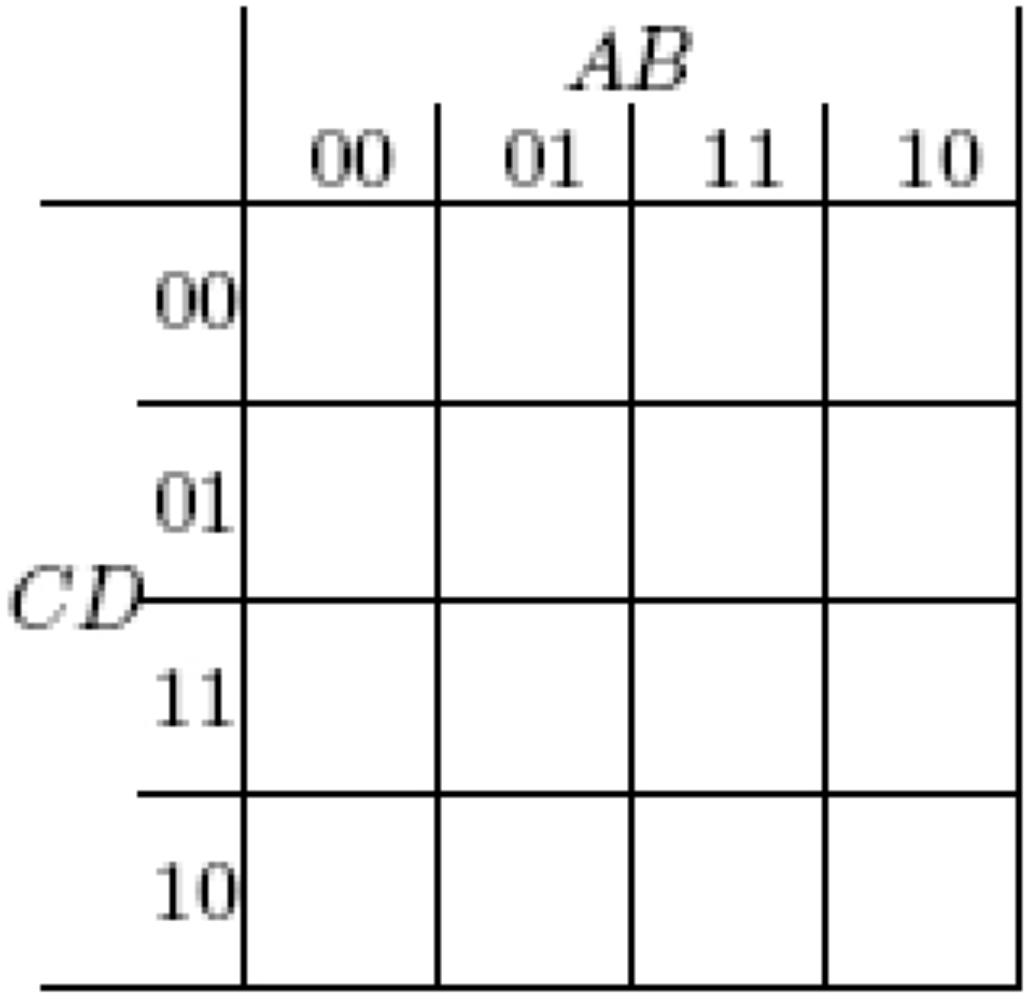
Minimizzazione delle funzioni booleane

C \ AB	00	01	11	10
0				
1				

Minimizzazione delle funzioni booleane

- Raffiguriamo il diagramma della mappa di Karnaugh per quattro variabili.
- Le variabili sono identificate sopra e a lato del diagramma.
- Le combinazioni delle variabili *A* e *B,* sopra in orizzontale, e delle variabili *C* e *D,* lateralmente in verticale, sono disposte secondo il codice di Gray di due variabili.

Minimizzazione delle funzioni booleane



Minimizzazione delle funzioni booleane

- Consideriamo ora la struttura delle mappe di quattro variabili. Avendo ordinato le combinazioni, contenute nelle celle, secondo il codice di Gray, che è ciclico, i due bordi superiore ed inferiore della tabella risultano adiacenti e una sola variabile cambia stato nell'attraversamento del bordo. Pertanto a questo punto possiamo considerare la mappa come cilindrica, con i due bordi superiore ed inferiore coincidenti.
- Una considerazione analoga vale per i bordi destro e sinistro: anch'essi sono coincidenti. Ecco allora che la mappa di Karnaugh, che disegnamo in forma di matrice piana, è in realtà in forma di una superficie toroidale, senza bordi. Di conseguenza, qualsiasi cella ha una cella contigua su ognuno dei suoi quattro lati.

Meccanismo della semplificazione

- Si parte scrivendo la tabella della verità della funzione. Da essa si ricava quali sono le combinazioni vere e si pone un '1' nelle celle della mappa corrispondenti alle combinazioni vere. Ogni '1' collocato nella mappa corrisponde ad una combinazione presente nella somma canonica, espressione della funzione.
- Per come è stata costruita la mappa, a due '1' collocati in celle contigue corrispondono combinazioni che differiscono soltanto in una variabile: le rispettive combinazioni nella somma canonica si sommano secondo il teorema (in forma generalizzata)

$$term*Y + term*\ !Y = term\ *(Y + !Y\ ) = term.$$

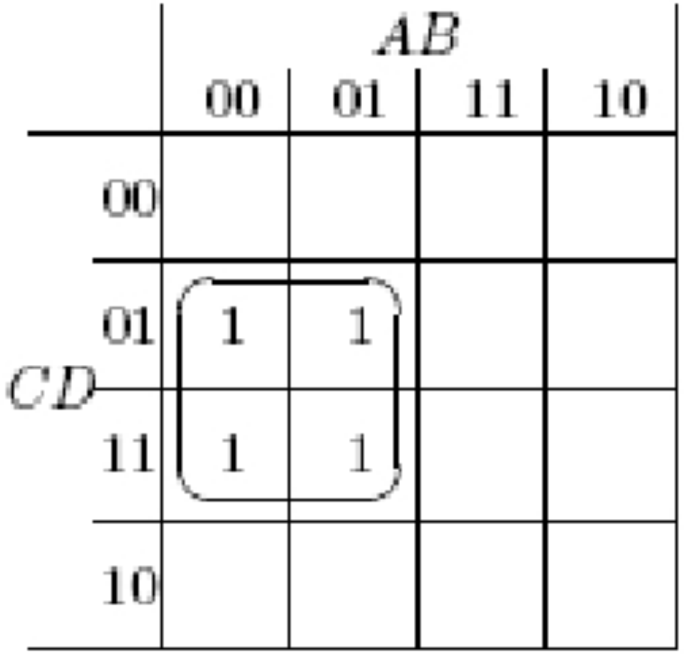
- La semplificazione delle funzioni avviene attraverso l'applicazione ripetuta del suddetto teorema. Inoltre le ridondanze sono automaticamente eliminate.

Meccanismo della semplificazione

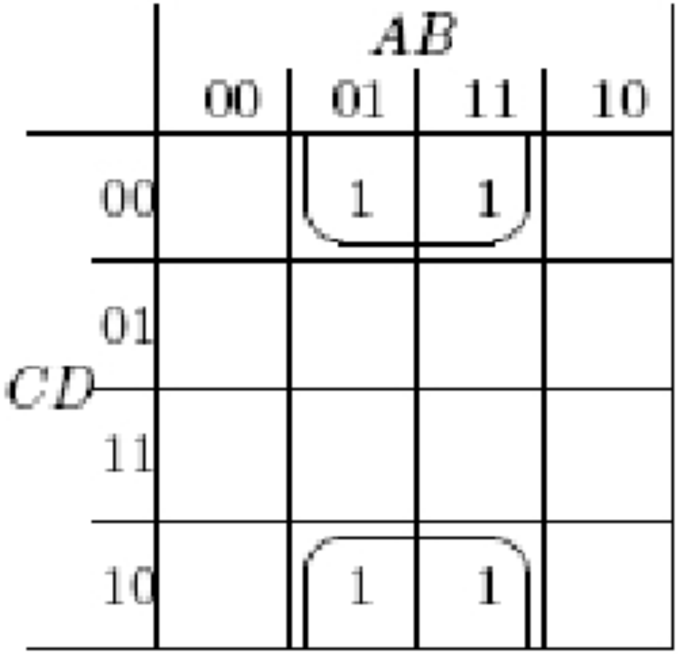
- Elenchiamo qui di seguito le regole da seguire per individuare i gruppi di celle rilevanti per costruire l'espressione semplificata di una funzione.
  - i gruppi possono contenere 1, 2, 4, 8 o in generale 2_n celle
  - i gruppi non possono includere celle contenenti uno '0'
  - i gruppi possono essere orizzontali o verticali, ma non diagonali: i gruppi sono quindi in forma di rettangoli o di quadrati
  - ogni gruppo deve essere il più largo possibile, cioè deve contenere quanti più '1' possibile
  - ogni cella contenente un '1' deve appartenere ad almeno un gruppo
  - i gruppi si possono sovrapporre
  - le celle che si trovano sui bordi possono venir raggruppate con quelle corrispondenti dal lato opposto (ricordiamoci del "toroide")
  - i gruppi devono essere nel minor numero possibile senza contraddire alcuna delle regole elencate precedentemente.

Meccanismo della semplificazione

- Seguono alcuni esempi di semplificazione con mappe di quattro variabili.



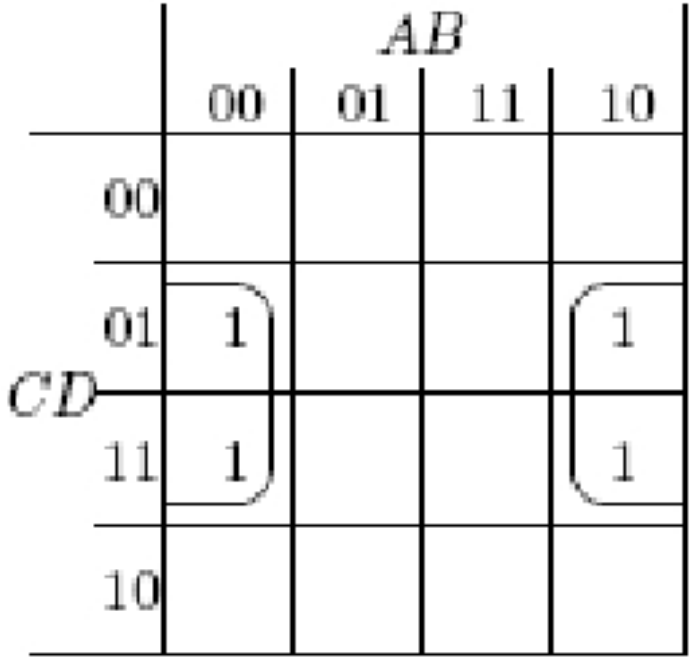
$\bar{A}D$



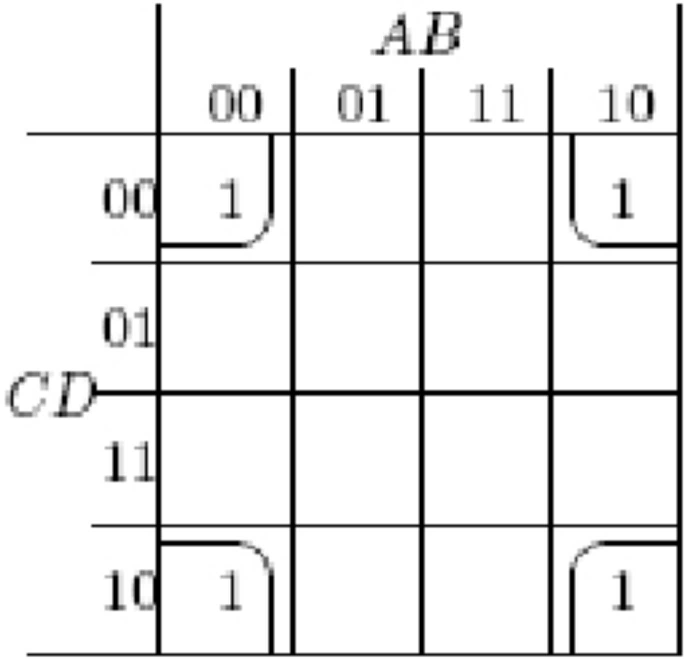
$B\bar{D}$

Meccanismo della semplificazione

- Seguono alcuni esempi di semplificazione con mappe di quattro variabili.



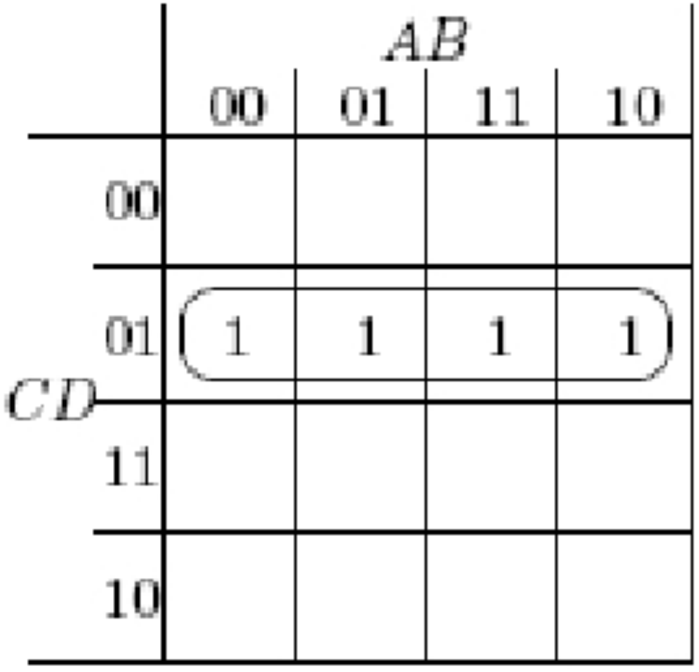
$\bar{B}D$



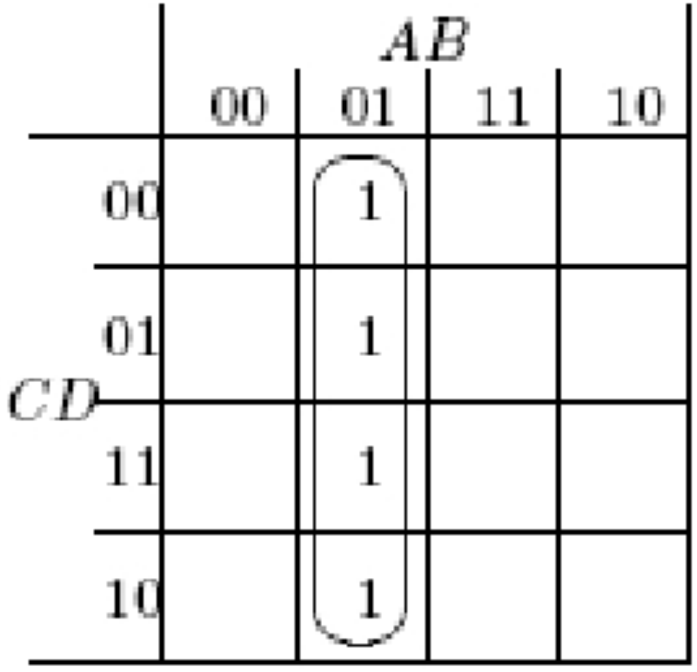
$\bar{B}\bar{D}$

Meccanismo della semplificazione

- Seguono alcuni esempi di semplificazione con mappe di quattro variabili.



$\bar{C}D$



$\bar{A}B$

Meccanismo della semplificazione

- Seguono alcuni esempi di semplificazione con mappe di quattro variabili.

$CD \backslash AB$	00	01	11	10
00			1	1
01			1	1
11			1	1
10			1	1

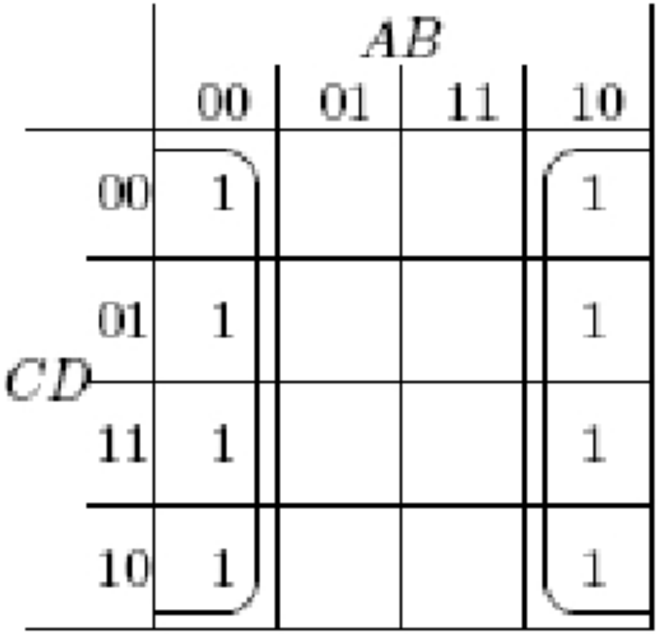
A

$CD \backslash AB$	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

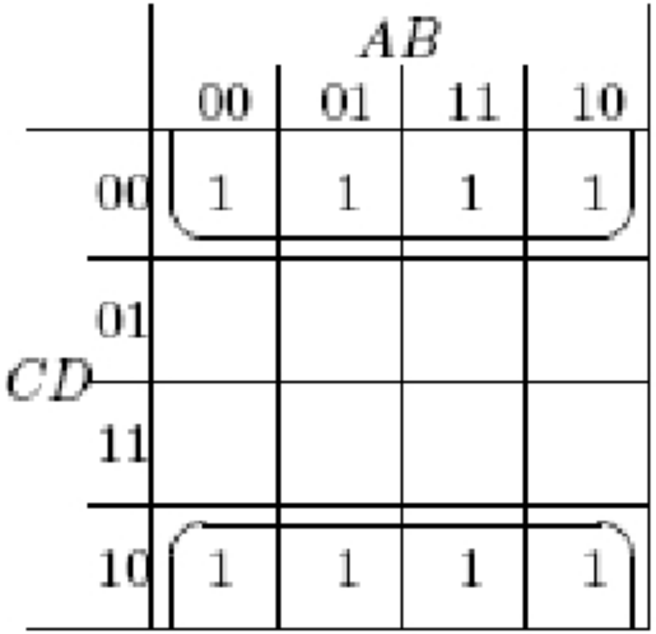
C

Meccanismo della semplificazione

- Seguono alcuni esempi di semplificazione con mappe di quattro variabili.



B



D

Condizioni di indifferenza e ridondanze

- Può accadere che nella tabella della verità di una funzione compaiano condizioni in cui per una qualche combinazione il valore della funzione può essere indifferentemente '1' o '0'. Questa condizione viene indicata nella tabella con il simbolo Φ. Tale simbolo appare anche nella mappa di Karnaugh.
- Può accadere inoltre che certe combinazioni non si verifichino mai (ad es., quando usiamo il codice BCD sei delle sedici combinazioni non si verificano mai): queste combinazioni sono dette ridondanze.
- Nella tabella della verità in corrispondenza della ridondanza viene posto il simbolo X. Tale simbolo appare anche nella mappa di Karnaugh.

Condizioni di indifferenza e ridondanze

- Sia la condizione di indifferenza sia la ridondanza sono utili nella minimizzazione delle espressioni delle funzioni. Basta scegliere uguali a '1' le condizioni di indifferenza che consentono una semplificazione.
- Le ridondanze non si verificano mai, quindi possiamo attribuire loro, dove è utile, il valore fittizio '1' unicamente per consentire la semplificazione.

Condizioni di indifferenza e ridondanze

- La tabella della verità seguente illustra le considerazioni fatte.

$A\ B\ C$	f_1	f_2
000	0	Φ
001	1	Φ
010	Φ	0
011	Φ	1
100	0	Φ
101	0	Φ
110	X	X
111	X	X

Condizioni di indifferenza e ridondanze

- Disegniamo le mappe di Karnaugh per f_1 e f_2 e ricaviamo le funzioni semplificate.

$C \backslash AB$	00	01	11	10
0		ϕ	$\times$	
1	1	ϕ	$\times$	

f_1

$C \backslash AB$	00	01	11	10
0	ϕ		$\times$	ϕ
1	ϕ	1	$\times$	ϕ

f_2

- Scrivere le forme canoniche con e senza indifferenze e ridondanze

Condizioni di indifferenza e ridondanze

- Risultano:

$$f_1 = !A\ C; \quad f_2 = C.$$

- Non considerando le condizioni di indifferenza e le ridondanze si avrebbe:

$$f_1 = !A\ !BC; \quad f_2 = !ABC.$$

Logica Combinatoria

- una rete combinatoria è un circuito logico avente n ingressi ($x_1,x_2,\ldots,x_n$) ed m uscite ($y_1,y_2, \ldots,y_m$), ciascuno dei quali assume valori binari (0/1), tale che a ciascuna combinazione degli ingressi corrisponde un'unica combinazione delle uscite.
- da un punto di vista logico, ogni uscita può essere definita come una funzione booleana degli ingressi $y_i = y_i(x_1,x_2,\ldots,x_n)$.
- ad ogni istante, il valore delle uscite dipende unicamente dal valore assunto dagli ingressi nello stesso istante.

Logica Combinatoria

- La procedura per progettare una rete logica combinatoria passa attraverso i seguenti stadi:
- 1: definizione completa e univoca del problema da risolvere
- 2: analisi del problema, con individuazione delle variabili d'ingresso e delle funzioni di uscita
- 3: scrittura della tabella della verità di ogni funzione
- 4: sintesi delle funzioni e loro semplificazione con le mappe di Karnaugh
- 5: disegno della schema logico della rete.

Tabelle delle verità

- Una funzione booleana può essere descritta per mezzo di una ***tabella delle verità*** che assegna ad ogni combinazione dei valori di input i corrispondenti valori agli output.
- Per ogni funzione esiste un'unica tabella delle verità che la rappresenta e viceversa.
- Tuttavia per ogni funzione esistono infinite espressioni per rappresentarla.

Esercizio 1

- Data la seguente funzione booleana avente 4 ingressi ed una uscita:
  - On-set = [1,4,9,12,13,15]
  - Dc-set = [0,3,10,11]
- Utilizzando il metodo delle mappe di Karnaugh sintetizzarla in forma minima come somma di prodotti.

Esercizio 1

A	B	C	D	
0	0	0	0	X
0	0	0	1	1
0	0	1	0	0
0	0	1	1	X
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

On-set = [1,4,9,12,13,15]
Dc-set = [0,3,10,11]

CD \ AB	00	01	11	10
00	X	1	1	
01	1		1	1
11	X		1	X
10				X

$$f = B!C!D + !BD + AD$$

Esercizio 2

- Data la seguente funzione booleana avente 4 ingressi ed una uscita:
  - On-set = [0,2,4,7,9,12,14]
  - Dc-set = [1,6,8,11]
- Utilizzando il metodo delle mappe di Karnaugh sintetizzarla in forma minima come somma di prodotti.

Esercizio 1

A	B	C	D	F
0	0	0	0	1
0	0	0	1	X
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	X
0	1	1	1	1
1	0	0	0	X
1	0	0	1	1
1	0	1	0	0
1	0	1	1	X
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

On-set = [0,2,4,7,9,12,14]
Dc-set = [1,6,8,11]

CD \ AB	00	01	11	10
00	1	1	1	X
01	x			1
11		1		X
10	1	X	1	

$$f = !A * B * C + B * !D + !A * !D + !B * !C$$

Esercizio 3

- Data la seguente funzione booleana avente 4 ingressi ed una uscita:
  - On-set = [7,8,9,10,11,14,15]
  - Dc-set = [1,3,5]
- Utilizzando il metodo delle mappe di Karnaugh sintetizzarla in forma minima come somma di prodotti.

Esercizio 1

A	B	C	D	F
0	0	0	0	0
0	0	0	1	X
0	0	1	0	0
0	0	1	1	X
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

On-set = [7,8,9,10,11,14,15]
Dc-set = [1,3,5]

CD \ AB	00	01	11	10
00				1
01	X	X		1
11	X	1	1	1
10			1	1

$$f = C * D + \color{blue}{A * !B} + \color{red}{A * C}$$

Esercizio 1

- Data la seguente funzione booleana avente 4 ingressi ed una uscita:
  - On-set = [0,1,4,7,10,12,15]
  - Dc-set = [6,11,14]
- Utilizzando il metodo delle mappe di Karnaugh sintetizzarla in forma minima come somma di prodotti.

Esercizio 1

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	X
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	X
1	1	0	0	1
1	1	0	1	0
1	1	1	0	X
1	1	1	1	1

On-set = [0,1,4,7,10,12,15]
Dc-set = [6,11,14]

AB

CD	00	01	11	10
00	1	1	1	
01	1			
11		1	1	X
10		X	X	1

$$f = !A * !B * !C + B * !D + B * C + A * C$$

…ricapitolando

Mappe di Karnaugh

- Le mappe di Karnaugh sono un metodo per semplificare espressioni booleane in forma SOP
- In realtà non introducono tecniche di semplificazione nuove, sono semplicemente un espediente grafico che consente di rilevare più facilmente implicati che possono essere semplificati
- Quindi alla base delle mappe di Karnaugh c'è il solito principio:

$$PA + P\overline{A} = P$$

Karnaugh Maps (K-Maps)

- Le espressioni booleane sono minimizzate combinando i termini
- Le K-map minimizzano graficamente le equazioni
- $PA + P\overline{A} = P$
- $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C = \overline{A}\,\overline{B}$

A	B	C	Y
0	0	0	**1**
0	0	1	**1**
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y: C \ AB	00	01	11	10
0	**1**	0	0	0
1	**1**	0	0	0

Y: C \ AB	00	01	11	10
0	$\boldsymbol{\bar{A}\bar{B}\bar{C}}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\boldsymbol{\bar{A}\bar{B}C}$	$\bar{A}BC$	ABC	$A\bar{B}C$

K-map

- Il cerchio 1 è nelle caselle adiacenti
- Nell'espressione booleana, includere solo i valori letterali la cui forma true e complemento non sono nel cerchio

A	B	C	Y
0	0	0	**1**
0	0	1	**1**
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y — C \ AB	00	01	11	10
0	1	0	0	0
1	1	0	0	0

$$Y = \overline{A}\,\overline{B}$$

3-Input K-Map

Y: C \ AB	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

Truth Table

A	B	C	Y
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	0
1	1	1	1

K-Map

Y: C \ AB	00	01	11	10
0				
1				

K-Map Definitions

- **Complement:** variable with a bar over it

  $\bar{A}, \bar{B}, \bar{C}$

- **Literal:** variable or its complement

  $\bar{A}, A, \bar{B}, B, C, \bar{C}$

- **Implicant:** product of literals

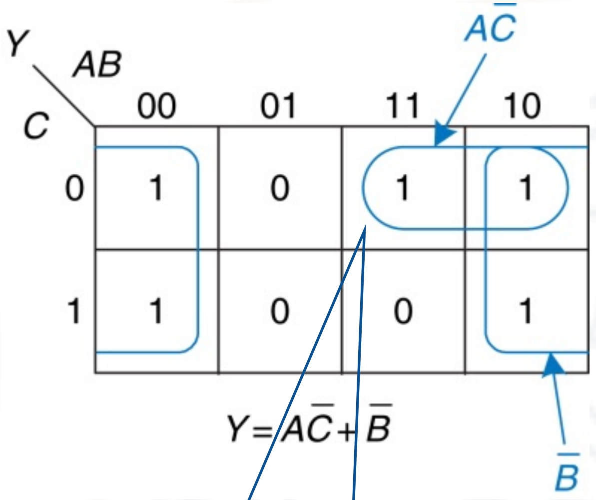
  $A\bar{B}C, \bar{A}C, BC$

- **Prime implicant:** implicant corresponding to the largest circle in a K-map

K-Map Rules

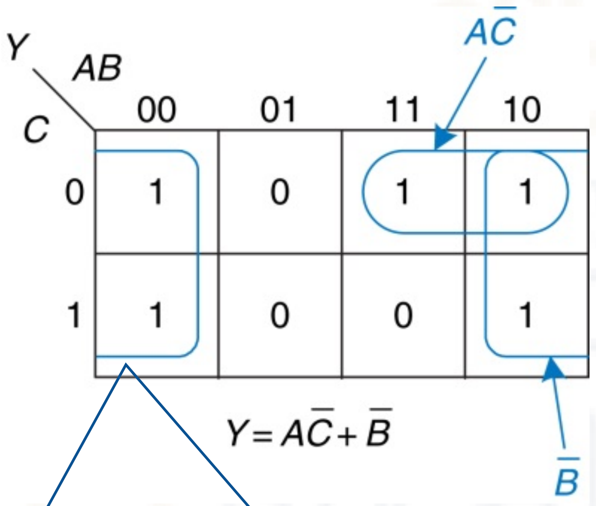
- Every 1 must be circled at least once
- Each circle must span a power of 2 (i.e. 1, 2, 4) squares in each direction
- Each circle must be as large as possible
- A circle may wrap around the edges
- A “don't care” (X) is circled only if it helps minimize the equation

Y (C \ AB)	00	01	11	10
0	1	0	1	1
1	1	0	0	1



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

C \ AB	00	01	11	10
0	1	0	1	1
1	1	0	0	1



$$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C = \bar{A}\bar{B}(\bar{C} + C) + A\bar{B}\,(\bar{C} + C) = \bar{A}\bar{B} + A\bar{B} = (\bar{A} + A)\,\bar{B} = \bar{B}$$

4-Input K-Map

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y

CD \ AB	00	01	11	10
00				
01				
11				
10				

4-Input K-Map

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

4-Input K-Map

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

$$Y = \underbrace{\overline{A}C}_{3} + \underbrace{\overline{A}BD}_{1} + \underbrace{A\overline{B}\,\overline{C}}_{2} + \underbrace{\overline{B}\,\overline{D}}_{4}$$

Esercizio

- Per frequentare un certo corso di elettronica uno studente deve soddisfare le seguenti condizioni:
  1. aver superato almeno 20 esami **ed** essere uno studente di ingegneria in corso, **oppure**
  2. aver superato almeno 20 esami **ed** essere uno studente di ingegneria con il piano di studio approvato, **oppure**
  3. aver superato meno di 20 esami **ed** essere uno studente di ingegneria fuori corso, **oppure**
  4. essere in corso **ed** avere il piano di studio approvato, **oppure**
  5. essere uno studente di ingegneria **ed** avere il piano di studi non ancora approvato.

- Ricavare la funzione logica che minimizza le condizioni precedenti.

Esercizio (definizione variabili)

- Introduciamo le variabili logiche A, B, C, D, Z e definiamole nel seguente modo:
  - A= lo studente ha superato almeno 20 esami
  - B= lo studente è studente di ingegneria
  - C= lo studente è in corso
  - D= lo studente ha il piano di studi approvato
  - Z= lo studente può frequentare il corso

Esercizio (tabella della verità)

A	B	C	D	Z
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**1**
0	1	0	0	**1**
0	1	0	1	**1**
0	1	1	0	**1**
0	1	1	1	**1**
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**1**
1	1	0	1	**1**
1	1	1	0	**1**
1	1	1	1	**1**

1. aver superato almeno 20 esami ed essere uno studente di ingegneria in corso, oppure
2. aver superato almeno 20 esami ed essere uno studente di ingegneria con il piano di studio approvato, oppure
3. aver superato meno di 20 esami ed essere uno studente di ingegneria fuori corso, oppure
4. essere in corso ed avere il piano di studio approvato, oppure
5. essere uno studente di ingegneria con il piano di studi non ancora approvato.

A= lo studente ha superato almeno 20 esami
B= lo studente è studente di ingegneria
C= lo studente è in corso
D= lo studente ha il piano di studi approvato
Z= lo studente può frequentare il corso

Esercizio (mappa di Karnaugh))

A	B	C	D	Z
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

CD \ AB	00	01	11	10
00		1	1	
01		1	1	
11	1	1	1	1
10		1	1	

Esercizio (mappa di Karnaugh))

CD \ AB	00	01	11	10
00		1	1	
01		1	1	
11	1	1	1	1
10		1	1	

$Z = B + CD$

Esercizio (realizzazione)

Z = B + CD