Progettazione mediante mappe di Karnaugh

Progettazione circuiti

Finora abbiamo visto come progettare circuiti utilizzando:

1. **Espansione mintermini o maxtermini**
2. **Semplificazione mediante algebra di Boole**

Svantaggi semplificazione mediante algebra di Boole

1. Difficile da applicare in *maniera* ***sistematica***
2. Difficile ***capire*** *se abbiamo ottenuto o meno una* ***soluzione minimale***

Alternativa semplice e veloce:
mappe di Karnaugh (o K-maps)

Minimizzazione mediante mappe di Karnaugh

- Metodo ***visuale*** (*alternativo* al metodo analitico)
- Semplice da utilizzare
- Limitazione:
  - Adatto a ***funzioni di 2,3,4 variabili***
  - Ancora *applicabile con 5, in casi estremi con 6 variabili*
- Per funzioni con più variabili occorre utilizzare metodi di *minimizzazione algoritmici*

Mappa a 2 variabili

Simile a una tabella di verità, specifica il valore della funzione per determinate combinazioni di valori

#	A	B	mintermine
0	0	0	m_0
1	0	1	m_1
2	1	0	m_2
3	1	1	m_3

B \ A	0	1
0	m_0	m_2
1	m_1	m_3

Esempio

Partiamo da una tabella di verità

#	A	B	F
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	0

Esempio

Costruiamo la K-map riportando i valori dalla tabella di verità

#	A	B	F
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	0

B \ A	0	1
0	**1**	**0**
1	**1**	**0**

Esempio

I valori 1 corrispondono a mintermini

#	A	B	F
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	0

B \ A	0	1
0	**1**	**0**
1	**1**	**0**

B \ A	0	1
0	**1** (A'B')	**0**
1	**1** (A'B)	**0**

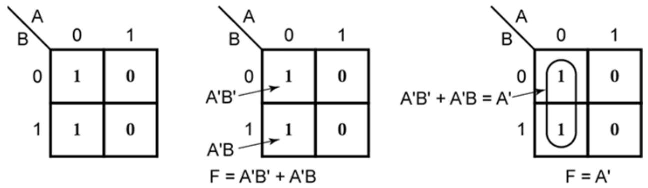
F = A'B' + A'B

Esempio

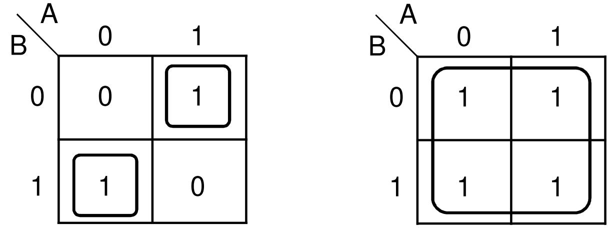
Identifichiamo **rettangoli massimali costituiti da valori 1 adiacenti**

Il numero di celle coinvolte **deve** essere **potenza di 2**

#	A	B	F
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	0



Altri esempi

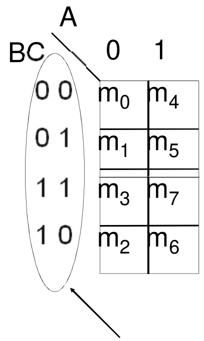


F=A'B+AB'

F=1

K-map a 3 variabili

#	A	B	C	mintermine
0	0	0	0	m_0
1	0	0	1	m_1
2	0	1	0	m_2
3	0	1	1	m_3
4	1	0	0	m_4
5	1	0	1	m_5
6	1	1	0	m_6
7	1	1	1	m_7



notare l'ordinamento

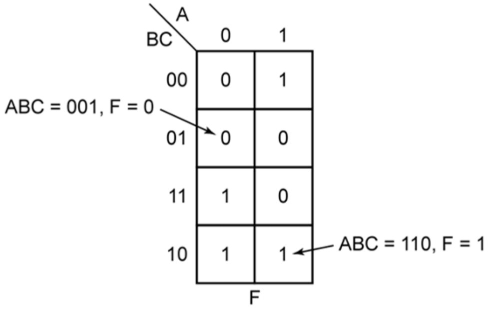
Gray Code

- ***Se su un lato della tabella ci sono due variabili***, i valori non sono ordinati in maniera crescente o decrescente
- Invece, la sequenza segue la regola del **gray code**
- Tra una stringa di bit e la successiva può **cambiare un solo bit**
- Ideato da Frank Gray nel 1947, utilizzato in telecomunicazioni

Decimale	Binario	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

K-map a 3 variabili: Esempio

#	A	B	C	**F**
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0



Progettazione mediante SOP o POS

È possibile usare le K-maps per ottenere SOP o POS

Come?

- Esattamente come fatto in precedenza per l'espansione in mintermini o maxtermini
- Identificando **gruppi di celle adiacenti contenenti 1 nel caso di SOP o 0 nel caso di POS**
- *Le celle 1 sono da interpretare come mintermini, le celle 0 come maxtermini*

Procedura

- Identificare gruppi di celle adiacenti contenenti lo stesso valore logico (1 per SOP, 0 per POS)

I gruppi di celle:

- Devono essere *adiacenti*
- Devono essere un *numero potenza di 2* (es. 1, 2, 4, 8, …)

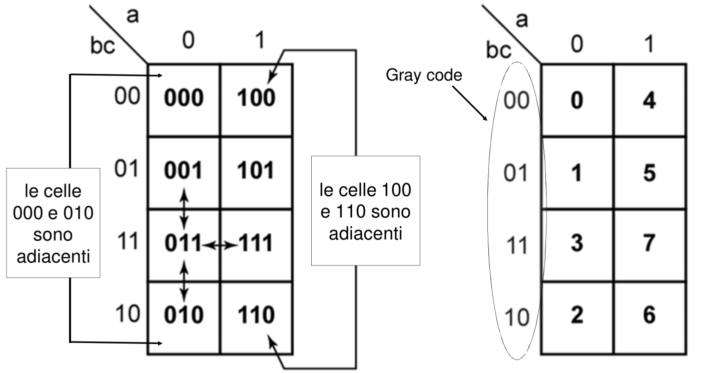
Celle adiacenti:

- *Differiscono per il valore logico di una sola delle variabili*
- Quindi, una cella sul *bordo* della K-map è adiacente alla cella corrispondente sul bordo opposto

Quanto semplifico?

- *Maggiore il numero di celle raggruppate, **migliore il livello di semplificazione***
  - Ovvero, variabili che semplifico
- **Minore il numero di gruppi utilizzati, minore il numero di termini nella somma**

Adiacenze nelle K-map a 3 variabili



Esempio

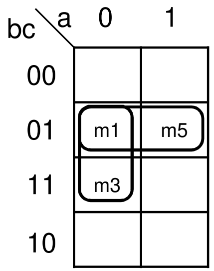
Minimizzare la seguente funzione:

$F(a, b, c) = \Sigma m(1, 3, 5) = \prod M(0, 2, 4, 6, 7)$

Forma SOP

$F(a, b, c) = \Sigma m(1, 3, 5)$

#	A	B	C	mintermine
0	0	0	0	m_0
1	0	0	1	m_1
2	0	1	0	m_2
3	0	1	1	m_3
4	1	0	0	m_4
5	1	0	1	m_5
6	1	1	0	m_6
7	1	1	1	m_7



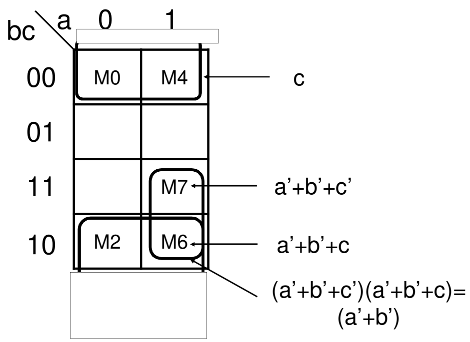
F=B'C+A'C

Beh.. avremmo potuto semplificare! F=C(A'+B')

Minimizzazione: maxtermini

$F(a, b, c) = \prod M(0, 2, 4, 6, 7)$

#	A	B	C	maxtermine
0	0	0	0	M_0
1	0	0	1	M_1
2	0	1	0	M_2
3	0	1	1	M_3
4	1	0	0	M_4
5	1	0	1	M_5
6	1	1	0	M_6
7	1	1	1	M_7



F=c(a'+b')

Esercizio

Derivare la forma SOP minima per la seguente espressione:

$F(A,B,C) = \Sigma m(1, 3, 4, 6)$

Esercizio

Determinare la forma POS minima per la seguente espressione:

$F(A,B,C) = \Pi\, M(0, 2, 6)$

Esercizio

Determinare, data la seguente tabella di verità

1. Espansione in mintermini
2. Espansione in maxtermini
3. Forma SOP minimizzata
4. Forma POS minimizzata

#	A	B	C		F
0	0	0	0		0
1	0	0	1		1
2	0	1	0		0
3	0	1	1		1
4	1	0	0		1
5	1	0	1		0
6	1	1	0		0
7	1	1	1		1

Esempio

- Usiamo le K-map per determinare la forma SOP minima per la seguente espressione:
- $F(a,b,c) = \Sigma m(0, 1, 2, 5, 6, 7)$

Costruiamo la K-map

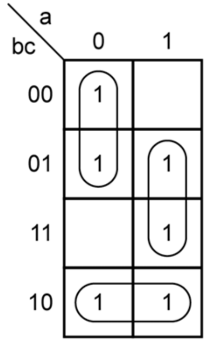
$F(a,b,c) = \Sigma m(0, 1, 2, 5, 6, 7)$

#	A	B	C		F
0	0	0	0		1
1	0	0	1		1
2	0	1	0		1
3	0	1	1		0
4	1	0	0		0
5	1	0	1		1
6	1	1	0		1
7	1	1	1		1

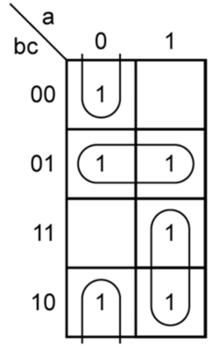
bc \ a	0	1
00	m0	
01	m1	m5
11		m7
10	m2	m6

Come raggruppiamo i mintermini?

Due forme minimali alternative



F = a'b' + bc' + ac

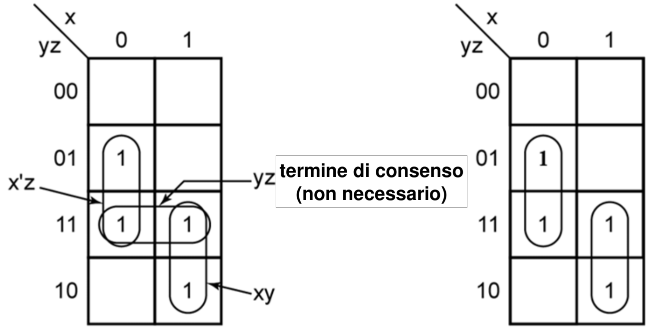


F = a'c' + b'c + ab

Teorema del consenso

- XY + X'Z + YZ = XY+X'Z
- Come lo dimostriamo con le K-Map?

Teorema del consenso



xy + x'z + yz = xy + x'z

K-Maps: Aumentiamo il numero di variabili…

K-map a 4 variabili

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

Le celle corrispondono alle 16 righe di una tabella di verità a 4 variabili

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Identifichiamo le adiacenze

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Identifichiamo le adiacenze

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

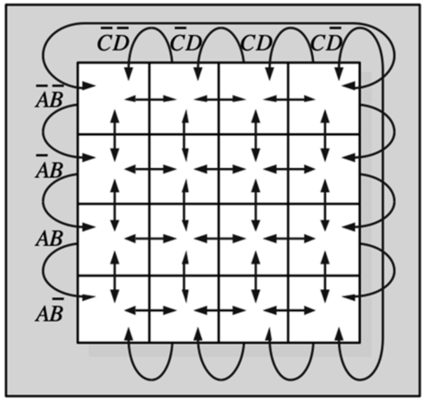
Identifichiamo le adiacenze

CD \ AB	00	01	11	10
00	$\boxed{0}$	4	12	8
01	$\boxed{1}$	$\boxed{5}$	13	$\boxed{9}$
11	$\boxed{3}$	7	15	11
10	2	6	14	10

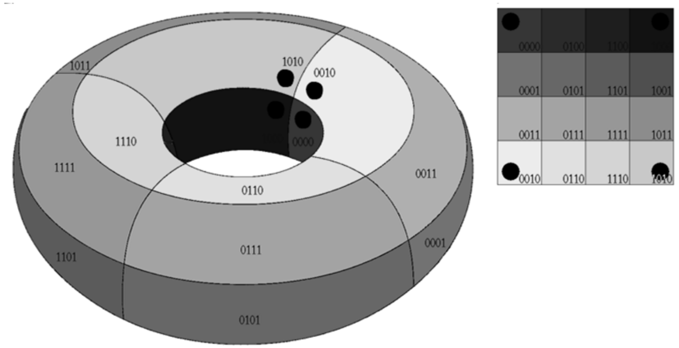
Identifichiamo le adiacenze

CD \ AB	00	01	11	10
00	0	$\boxed{4}$	12	8
01	1	5	13	9
11	3	$\boxed{7}$	15	11
10	$\boxed{2}$	$\boxed{6}$	$\boxed{14}$	10

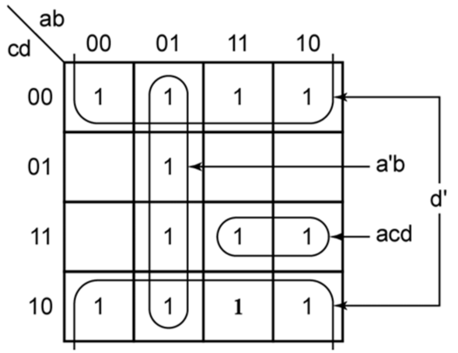
Adiacenze Sommario



Adiacenze (rappresentazione toroidale)



Esempio



Esempio

Minimizziamo la funzione:

$F = \Sigma m(1, 3, 4, 5, 10, 12, 13)$

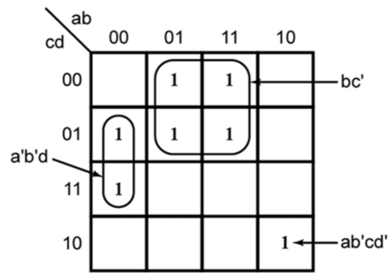
Esempio

#	a	b	c	d
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

$F = \Sigma m(1, 3, 4, 5, 10, 12, 13)$

cd \ ab	00	01	11	10
00		1	1	
01	1	1	1	
11	1			
10				1

Minimizzazione



$f_1 = \Sigma m(1, 3, 4, 5, 10, 12, 13)$
$= bc' + a'b'd + ab'cd'$

Esempio

Minimizziamo la funzione:

$F = \Sigma m(0, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15)$

Esempio

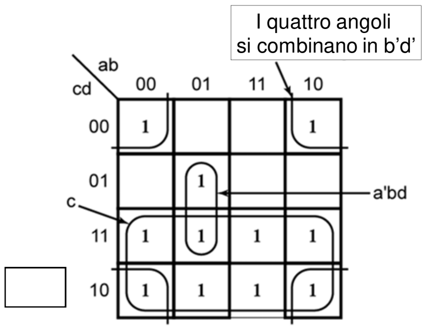
#	a	b	c	d
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

$F = \Sigma m(0, 2, 3, 5, 6,$

$7, 8, 10, 11, 14, 15)$

cd \ ab	00	01	11	10
00	1			1
01		1		
11	1	1	1	1
10	1	1	1	1

Esempio: soluzione



$f_2 = \Sigma m(0, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15)$
$= c + b'd' + a'bd$

Minimizzazione POS

Minimizziamo la funzione:

$F(a,b,c,d) = \Pi\ M(1, 3, 9, 12)$

Esempio (POS)

#	a	b	c	d
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

F(a,b,c,d) = Π M(1, 3, 9, 12)

cd \ ab	00	01	11	10
00			0	
01	0			0
11	0			
10				

Minimizzazione POS

cd \ ab	00	01	11	10
00			0	
01	0			0
11	0			
10				

F=(a+b+d')(b+c+d')(a'+b'+c+d)

Simulatore di circuiti

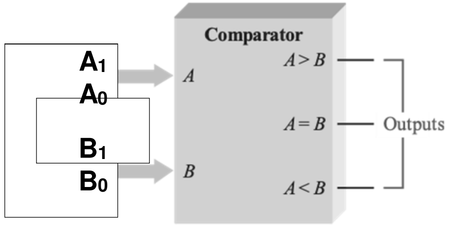
Logisim

http://sourceforge.net/projects/circuit/

- Consente di tracciare circuiti combinatori e sequenziali
- Packaging di sottocircuiti in componenti
- Simulazione e analisi circuiti
- Generazione automatica di circuiti a partire da tabelle di verità
- Semplificazione mediante Mappe di Karnaugh

Esercizio

- Ricordate il comparatore visto nella prima lezione?
- Bene, ora realizziamolo per il caso di ingressi a 2 bit



Esercizio

Determinare la forma SOP minima per la seguente espressione:

F(A,B,C,D) = Σm(1, 5, 6, 8, 9, 12, 13, 14)

Quindi, implementarla sia mediante AND/OR/NOT che mediante NAND

Esercizio

Determinare la forma POS minima per la seguente funzione

F(A,B,C,D) = Σm(1, 5, 6, 8, 9, 12, 13, 14)

Infine, produrre un'implementazione mediante porte NOR

Esercizio

Determinare l'espressione minima per la funzione:

F(A,B,C,D) = Π M(0, 2, 3, 7, 9, 10, 11, 14)

Nota: provare a realizzare la funzione sia in forma SOP che POS

Qual è la più conveniente?

Funzioni non completamente specificate

Funzioni non completamente specificate

- Come visto in precedenza, il valore assunto dalla funzione F è "**don't care**" ovvero una **X**
- Nell'espansione in mintermini o maxtermini, **potevamo assegnare alla X il valore 1 o 0**
- Nelle mappe di Karnaugh, **posso includere o meno le celle con la X**
  - Lo faccio **se ciò mi aiuta a minimizzare i termini o le literal in ciascun termine**

Esempio

Minimizzare la seguente funzione non completamente specificata:

$F = \Sigma m(1, 3, 5, 7, 9) + \Sigma d(6, 12, 13)$

Esempio

#	a	b	c	d
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

$F = \Sigma m(1, 3, 5, 7, 9) + \Sigma d(6, 12, 13)$

cd \ ab	00	01	11	10
00			X	
01	1	1	X	1
11	1	1		
10		X		

Minimizzazione SOP

cd \ ab	00	01	11	10
00			X	
01	1	1	X	1
11	1	1		
10		X		

f = Σm(1, 3, 5, 7, 9) + Σd(6, 12, 13)
 = a'd + c'd

Esercizio

Determinare la forma minima SOP per la seguente funzione non completamente specificata:

F(A,B,C,D) = Σm(1, 5, 9, 13, 14) + Σd(4, 7, 8, 15)

Esercizio

Determinare la forma minima POS per la seguente funzione non completamente specificata:

F(A,B,C,D) = Π M(1, 3, 4, 9, 10, 12) · Π D(2, 6, 11, 14)

Implicanti, Implicanti primi, Implicanti primi essenziali

Implicanti e implicanti primi

Literal

- Ciascuna variabile nella forma vera o complementata

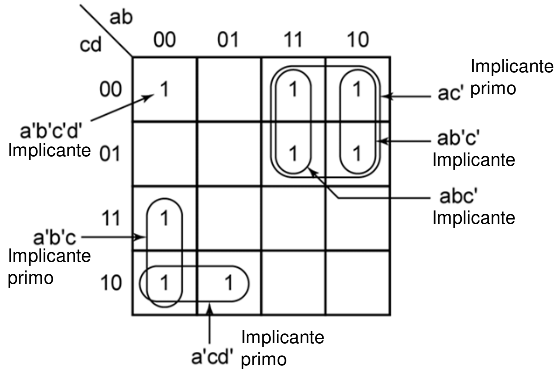
Implicante (SOP)

- Gruppo di 1 (o singolo 1) che può essere combinato seguendo le regole di adiacenza delle K-map
- Corrisponde a un **prodotto di termini**

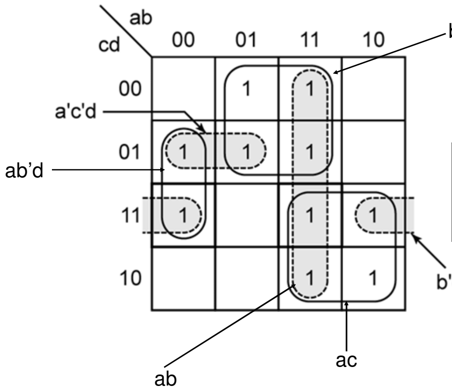
Implicante primo (SOP)

- Un prodotto di termini **che NON può essere combinato con un altro per eliminare una literal**

Esempio



Identificazione Implicanti Primi

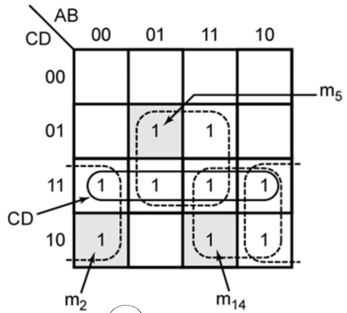


Soluzione minima:
F=a'b'd+bc'+ac

Implicanti primi:
a'b'd', bc', ac, a'c'd, ab, b'cd

Non tutti gli implicanti primi sono necessari per la rappresentazione minima…quelli in azzurro sono già coperti da altri implicanti primi. La forma minima contiene solo un sottoinsieme degli implicanti primi -> quelli essenziali

Identificazione termini essenziali

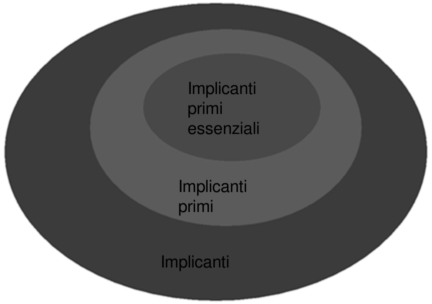


f = CD + BD + B'C + AC

Potremmo fare a meno di qualche termine?

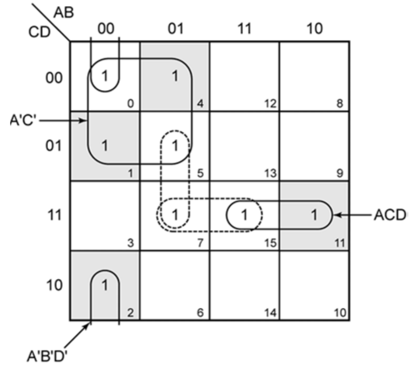
Implicanti primi essenziali

Se un mintermine **<u>è coperto da un solo implicante primo, allora tale implicante primo è detto implicante primo essenziale</u>**, e quindi deve essere incluso nella SOP minimale



Implicanti primi essenziali

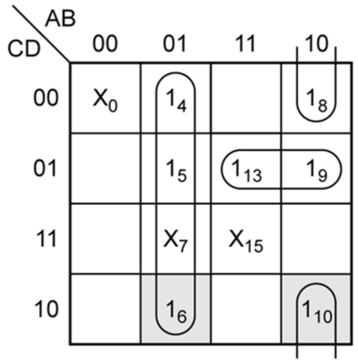
I *<u>mintermini in blu sono coperti da un solo implicante primo</u>*. Negli altri casi è possibile trovare almeno 2 implicanti primi che coprono quel mintermine



Identificazione copertura minima

- Identificare **<u>implicanti primi</u>**
- Tra questi, identificare gli **implicanti primi essenziali**
- Quindi, cercare di **<u>coprire quanto non ancora coperto dagli implicanti primi essenziali</u>**
  - A volta, la scelta non è ovvia…
- L'espressione Booleana **<u>risultante potrebbe non essere unica</u>**

Determinare la copertura minima



Gli 1 in blu sono coperti solo da un implicante primo. Implicanti primi essenziali: A′B, AB′D′

AC′D Copre gli 1 restanti

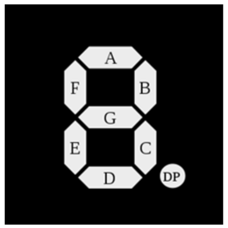
Copertura minima

Una copertura minima consiste **nel numero minore possibile di termini prodotto** (per un'espressione SOP**) o termini somma** (espressione POS) e, per ciascun termine, nel **numero minimo possibile di literal**

Applicazione

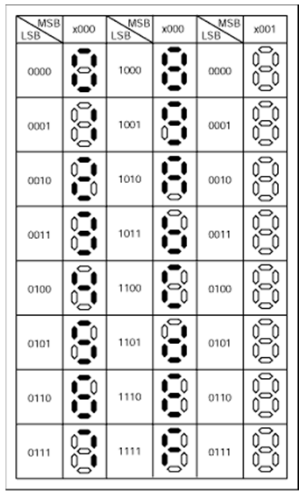
Decoder per display a 7 segmenti

Display a 7 segmenti

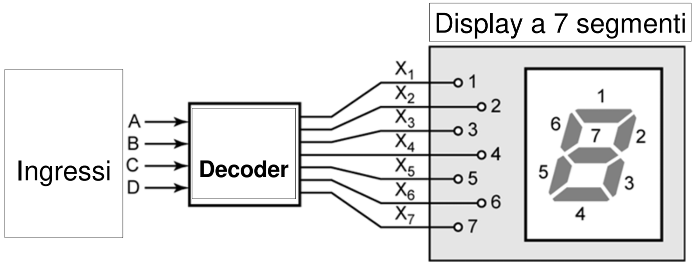


Codifica

Come noterete il display sarebbe in grado di pilotare tutte le cifre esadecimali (0-9 e A-F)



Come il decoder pilota il display



Esercizio

- Progettare un decoder per display a 7 segmenti
- **Ingressi:** 4 bit: ABCD (A è il bit più significativo)
- **Uscite:** X_1, X_2, X_3, X_4, X_5, X_6, X_7
  - pilotano i segmenti del display
- Nota: supponiamo che gli ingressi siano in BCD (Binary-Coded-Decimal), quindi ignoriamo l'uscita per ingressi >1001

Conversione POS-SOP

Conversione POS-SOP

Effettuare conversioni POS-SOP e viceversa mediante mappe di Karnaugh è molto semplice

Procedimento:

1. Data l'espressione di partenza, tracciare i corrispondenti zeri (POS) o uni (SOP) sulla mappa
2. Quindi, riempire le restanti celle con uni (passaggio a SOP) o zeri (passaggio a POS)
3. Infine, identificare gruppi sugli uni o zeri appena tracciati

Esempio

(A’+B’+C+D)(A+B’+C+D)(A+B+C+D’)(A+B+C’+D’)(A’+
B+C+D)(A+B’+C’+D)

Tracciamo gli zeri

(A’+B’+C+D)(A+B’+C+D)(A+B+C+D’)(A+B+C’+D’)(A’+B+C+D)(A+B’+C’+D)

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

Aggiungiamo gli uni…

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

Raggruppiamo

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

F=AD+AC+BD+A’B’D’

Esercizio

Convertire l'espressione

(W+X'+Y+Z')(W'+X+Y'+Z')(W'+X+Y'+Z)(W'+X'+Z')

in forma SOP minima

Design di circuiti con più output

Circuiti a più output

- In teoria, simile alla progettazione di circuiti a input singolo
- Progettiamo **il circuito di ogni singolo output**
- Successivamente si cerca di **sfruttare al massimo i sotto-circuiti in comune**

Esempio

$F_1(A,B,C,D)=\sum m(11,12,13,14,15)$

$F_2(A,B,C,D)=\sum m(3,7,11,12,13,15)$

$F_3(A,B,C,D)=\sum m(3,7,12,13,14,15)$

Mappe…

F$_1$

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

F$_2$

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

F$_3$

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

$F_1(A,B,C,D)=\sum m(11,12,13,14,15)$

$F_2(A,B,C,D)=\sum m(3,7,11,12,13,15)$

$F_3(A,B,C,D)=\sum m(3,7,12,13,14,15)$

Scelta ovvia…

F$_1$

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

F$_2$

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

F$_3$

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

$F_1(A,B,C,D)=AB+ACD$

$F_2(A,B,C,D)=CD+ABC'$

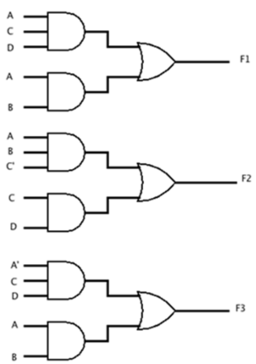
$F_3(A,B,C,D)=AB+A'CD$

Realizzazione "banale"

$F_1(A,B,C,D)=AB+ACD$

$F_2(A,B,C,D)=CD+ABC'$

$F_3(A,B,C,D)=AB+A'CD$



2 NOT, 6 AND, 3 OR

Alternativa

$F_1(A,B,C,D)=AB+ACD$

$F_2(A,B,C,D)=CD+ABC'$

$F_3(A,B,C,D)=AB+A'CD$

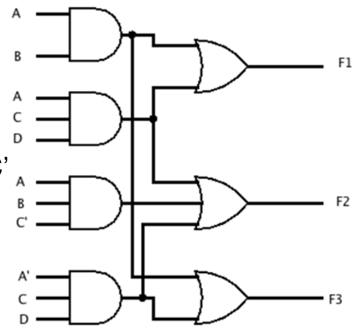
- **<u>AB è in comune tra F1 e F3</u>**
- In F_2 potrei scrivere CD= ACD+A'CD = CD (A+A')
  - A questo punto il termine CD non mi serve più

Risultato

$F_1(A,B,C,D)=AB+ACD$

$F_2(A,B,C,D)=ACD+A'CD+ABC'$

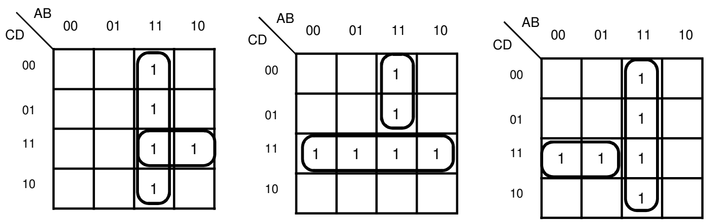
$F_3(A,B,C,D)=AB+A'CD$



2 NOT, 4 AND, 3 OR

Un **<u>numero maggiore di implicanti ci ha permesso di ottenere un circuito più semplice</u>**…

Scelta ovvia…



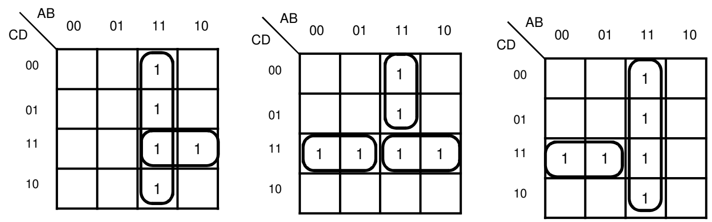
F_1 F_2 F_3

$F_1(A,B,C,D)=AB+ACD$

$F_2(A,B,C,D)=CD+ABC'$

$F_3(A,B,C,D)=AB+A'CD$

Dalle mappe…



F_1 F_2 F_3

Altro esempio…

$F_1(A,B,C,D)=\sum m(2,3,5,7,8,9,10,11,13,15)$

$F_2(A,B,C,D)=\sum m(2,3,5,6,7,10,11,14,15)$

$F_3(A,B,C,D)=\sum m(6,7,8,9,13,14,15)$

Mappe

F_1

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

F_2

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

F_3

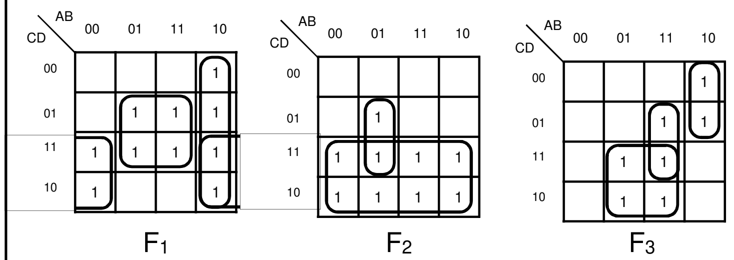
CD \ AB	00	01	11	10
00				1
01			1	1
11		1	1	
10		1	1	

$F_1(A,B,C,D)=\sum m(2,3,5,7,8,9,10,11,13,15)$

$F_2(A,B,C,D)=\sum m(2,3,5,6,7,10,11,14,15)$

$F_3(A,B,C,D)=\sum m(6,7,8,9,13,14,15)$

Minimizziamo separatamente F_1, F_2, F_3



$F_1(A,B,C,D)=BD+B'C+AB'$

$F_2(A,B,C,D)=C+A'BD$

$F_3(A,B,C,D)=BC+AB'C'+ABD$

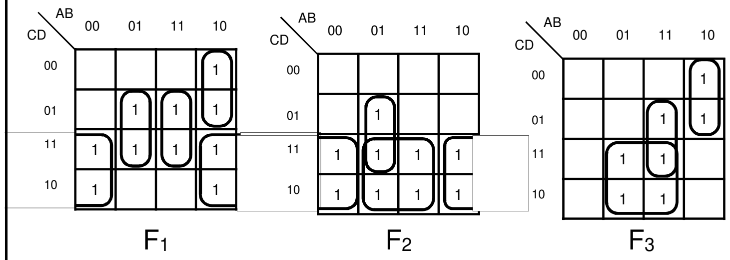
3 NOT
7 AND
3 OR

Come possiamo migliorare?

Non preoccupiamoci di minimizzare la singola funzione

Identifichiamo gli implicanti in comune

Implicanti in comune…



$F_1(A,B,C,D)=A'BD+ABD+AB'C'+B'C$

$F_2(A,B,C,D)=A'BD+B'C+BC$

$F_3(A,B,C,D)=ABD+AB'C'+BC$

3 NOT
5 AND
3 OR

Esercizio

Contate, nei due casi precedenti, il numero totale degli ingressi delle porte considerate

Esercizio

Ridurre il numero di gate da utilizzare per la realizzazione del 7-segment display decoder