Progettazione di Circuiti Combinatori

Sommario

- Mintermini e maxtermini
- Espansione in mintermini e maxtermini
- Procedure di progettazione e analisi
- Funzioni non completamente specificate

Specifica del circuito

La progettazione del circuito parte da una sua specifica, ovvero:

- Una descrizione testuale
- Una tabella di verità

Esempio

- Un full adder riceve in ingresso due cifre binarie **A**, **B**, e un riporto **Cin**
- Il full adder produce in uscita una somma **S** e un riporto **Cout**

Tabella di verità

#	A	B	Cin		S	Cout
0	0	0	0		0	0
1	0	0	1		1	0
2	0	1	0		1	0
3	0	1	1		0	1
4	1	0	0		1	0
5	1	0	1		0	1
6	1	1	0		0	1
7	1	1	1		1	1

Mintermini e Maxtermini

Mintermine

- Un *mintermine di **n** variabili è un prodotto (AND) di **n literal*** nelle quali ***ciascuna*** delle **n** variabili appare *esattamente **una volta***, nella forma *"true" (es. A) o complemento (es. A')*
  - Una literal è una variabile o il suo complemento
- Data una riga in una tabella di verità, un mintermine si ottiene:
  - Includendo la *forma true delle variabile* se il valore di tale variabile è ***1***
  - Includendo la *forma complementata* se il valore della variabile è ***0***

Mintermini

#	A	B	C		Mintermine
0	0	0	0		m_0=A'B'C'
1	0	0	1		m_1=A'B'C
2	0	1	0		m_2=A'BC'
3	0	1	1		m_3=A'BC
4	1	0	0		m_4=AB'C'
5	1	0	1		m_5=AB'C
6	1	1	0		m_6=ABC'
7	1	1	1		m_7=ABC

Espansione in mintermini

- Una funzione scritta come somma (OR) di mintermini prende il nome di **espansione in mintermini** o **forma canonica SOP** o **forma normale disgiuntiva**
- Data una tabella di verità, ***includiamo nell'OR tutti i mintermini che corrispondono alle righe in cui la funzione f=1***
- Data una funzione, la corrispondente espansione in mintermini è **unica**
  - Ma *non è necessariamente la soluzione a costo minimo*

Espansione mintermini

Determinare l'espansione in mintermini per la seguente tabella di verità

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

Soluzione

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

F=A'B'C'+A'BC+ABC'

Torniamo all'adder…

Produciamo l'espansione in mintermini per le uscite dell'adder…

#	A	B	Cin	S	Cout
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

S=A'B'Cin+A'BCin'+AB'Cin'+ABCin

Cout=A'BCin+AB'Cin+ABCin'+ABCin

Esercizio

Determinare l'espansione in mintermini per le seguenti funzioni:

- F_1 = AB + BC + A'C'
- F_2 = A·(B + C) + (A' + B)·(B + C')

Esercizio

Determinare la tabella di verità per le seguenti espressioni:

- F_1 = ABC' + AB'C + A'BC + A'BC' + AB'C'
- F_2 = ABCD + ABC'D' + A'BC'D + AB'C'D

Generalizzazione

$$F = a_0 m_0 + a_1 m_1 + a_2 m_2 + \cdots + a_7 m_7 = \sum_{i=0}^{7} a_i m_i$$

A B C	*F*
0 0 0	a_0
0 0 1	a_1
0 1 0	a_2
0 1 1	a_3
1 0 0	a_4
1 0 1	a_5
1 1 0	a_6
1 1 1	a_7

a_i vale 0 o 1

Notazione

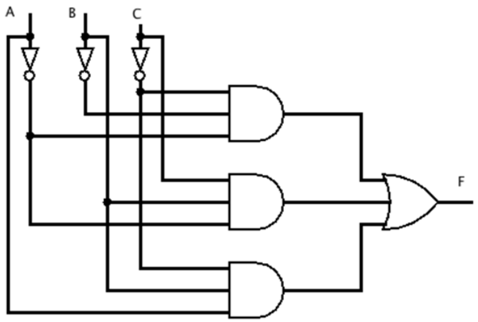
Usata per riferirsi all'espansione in mintermini

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

$F=\sum m(0,3,6)$

Realizzazione circuito

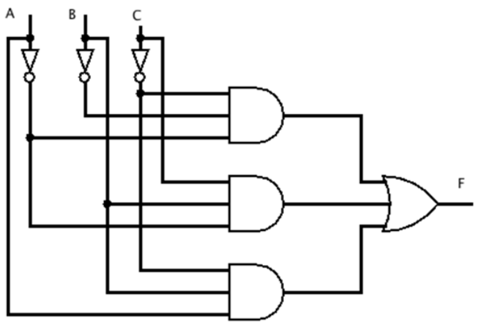
F=A’B’C’+A’BC+ABC’



Come potremmo realizzarlo con sole porte NAND?

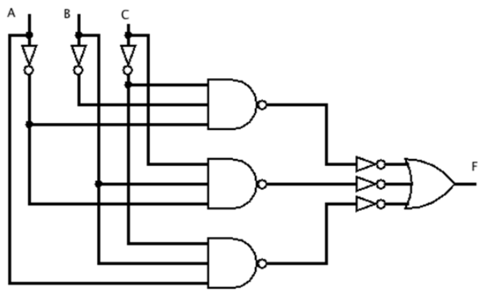
Realizzazione circuito

F=A’B’C’+A’BC+ABC’



Realizzazione mediante NAND

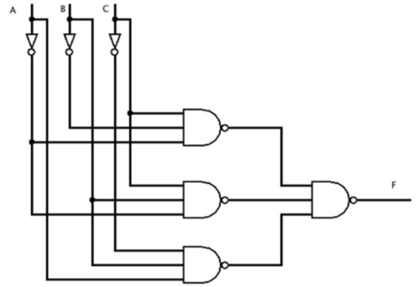
F=A’B’C’+A’BC+ABC’



Inserisco dei NOT davanti a tutti gli ingressi dell’OR che negano le uscite NOT dei NAND

Realizzazione mediante NAND

Sfruttiamo la proprietà X'+Y'+Z'= (XYZ)' e trasformiamo in NAND l'OR con ingressi negati



I restanti NOT si realizzano cortocircuitando gli ingressi di porte NAND

Realizzazione mediante NAND: sommario

- Sostituiamo gli AND con NAND
- Aggiungiamo un NOT in ingresso all'OR, in modo da annullare il NOT dei NAND
- Infine, possiamo realizzare i NOT e l'OR mediante NAND
- visto che ho negato gli ingressi della OR per annullare il NOT dei NAND**, ogni OR sara' del tipo A'+B', che corrisponde a (AB)' e quindi a una NAND**

Maxtermine

- Un maxtermine di **n** variabili è una *somma (OR)* di **n literal** in cui ciascuna variabile *appare esattamente una volta*, nella forma true o complemento (non entrambe)
- Data una tabella di verità, il maxtermine si ottiene
  - Includendo la **forma true** (es. A) **se il valore è 0**
  - Includendo la **forma complementata** (es. A') se il valore è **1**

Maxtermini

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

Espansione in maxtermini

- Una funzione scritta come prodotto (AND) di maxtermini è detta **espansione in maxtermini** o **forma canonica POS** o **forma normale congiuntiva**
- Data una tabella di verità, includiamo nell'AND tutti i *maxtermini che corrispondono alle righe in cui la funzione* ***f=0***
- Data una funzione, la corrispondente espansione in maxtermini **è unica**
- Ma non è necessariamente la soluzione a costo minimo

Espansione in maxtermini

Determinare l'espansione in maxtermini della funzione corrispondente alla seguente tabella di verità

A	B	C		F
0	0	0		1
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		0

Soluzione

A	B	C		F
0	0	0		1
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		0

F=(A+B+C')(A+B'+C)(A'+B+C)(A'+B'+C')

Perché funziona?

A	B	C		F
0	0	0		1
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		0

- Equivale a:
  (A'B'C)' · (A'BC')' · (AB'C')' · (ABC)'

Applichiamo De Morgan

$(A'B'C)' \cdot (A'BC')' \cdot (AB'C')' \cdot (ABC)' =$

$=(A''+B''+C') \cdot (A''+ B'+C'') \cdot (A'+B''+C'') \cdot (A'+B'+C')=$

$= (A+B+C') \cdot (A+ B'+C) \cdot (A'+B+C) \cdot (A'+B'+C')$

Corrisponde all'AND dei maxtermini delle righe in cui la funzione F vale zero

Torniamo all'adder…

Produciamo l'espansione in maxtermini per le uscite dell'adder…

#	A	B	Cin	S	Cout
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

S=(A+B+Cin)(A+B'+Cin')(A'+B+Cin')(A'+B'+Cin)

Cout=(A+B+Cin)(A+B+Cin')(A+B'+Cin)(A'+B+Cin)

Esercizio

Determinare la tabella di verità per le seguenti funzioni:

- $F_1=(A+B'+C)(A'+B'+C)$
- $F_2=(A+B'+C')(A'+B'+C')(A'+B+C')$

Espansione in maxtermini

Determinare l'espansione in maxtermini delle seguenti funzioni:

- $F_1 = (A + B')(A + C)(B + C')$
- $F_2 = A'B' + A'C' + B'C$

Generalizzazione

$$F = (a_0 + M_0)(a_1 + M_1)(a_2 + M_2) \cdots (a_7 + M_7) = \prod_{i=0}^{7}(a_i + M_i)$$

A B C	*F*
0 0 0	a_0
0 0 1	a_1
0 1 0	a_2
0 1 1	a_3
1 0 0	a_4
1 0 1	a_5
1 1 0	a_6
1 1 1	a_7

a_i vale 0 o 1

Notazione

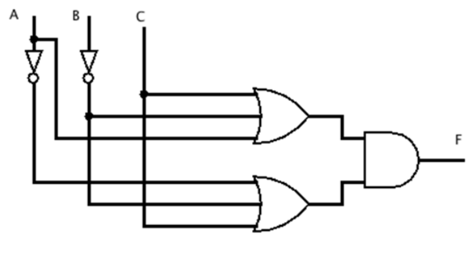
Per riferirsi all'espansione in maxtermini

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

$F=\prod M(1,2,4,7)$

Realizzazione circuito

F=(A+B'+C)(A'+B'+C)

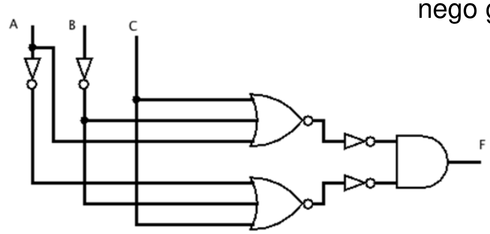


Realizzazione mediante NOR

Stesso principio utilizzato per la realizzazione tramite NAND delle SOP

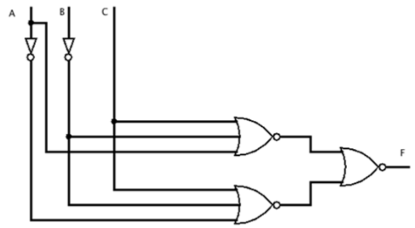
F=(A+B'+C)(A'+B'+C)

Inserisco dei NOR e poi nego gli ingressi delle AND



Realizzazione mediante NOR

Sostituiamo l'AND con ingressi negati con un NOR sfruttando la proprietà X'Y'=(X+Y)'



Realizziamo i restanti NOT cortocircuitando gli ingressi di porte NOR

Conversione SOP ➞ POS

1. Valutare ciascun mintermine identificando i corrispondenti numeri binari
2. Identificare i numeri binari non inclusi nella somma precedente
3. Per tali numeri binari, determinare i maxtermini e realizzare l'espansione in maxtermini

Esempio

- F=A'B'C'+A'BC+AB'C+ABC
- Numeri binari corrispondenti:
  - 000+011+101+111
- Quali combinazioni di numeri binari mancano?
  - 001, 010, 100, 110
- Realizziamo la corrispondente SOP:
  - (A+B+C')(A+B'+C)(A'+B+C)(A'+B'+C)

Conversione POS ➞ SOP

Analoga alla conversione SOP ➞ POS

1. Valutare ciascun maxtermine, identificando i corrispondenti numeri binari
2. Identificare i numeri binari non inclusi nella somma precedente
3. Per tali numeri binari, determinare i mintermini e realizzare l'espansione in mintermini

Esercizio

Convertire in SOP la seguente espressione POS:

F=(A+B+C)(A'+B+C')(A'+B+C)

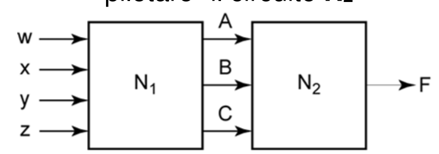
Funzioni non completamente specificate

Funzioni non completamente specificate

- Una funzione è completamente specificata se il suo valore (0 o 1*) è definito per tutte le combinazioni delle variabili d'ingresso*
- Potrebbero esserci casi in cui *non è necessario definire un valore di uscita* per alcune combinazioni di ingressi
- Tali combinazioni sono definite come "**don't care**"

Funzioni non completamente specificate

Consideriamo il circuito **N_1** che ha lo scopo di "pilotare" il circuito **N_2**



Supponiamo che il circuito **N_1** non generi tutte le combinazioni di **A B C**
Supponiamo che non vi siano combinazioni di **w, x, y, z** tali che **ABC** assumano i valori **001** o **110**

Tabelle di verità

Simbolo **X** o **d** al posto di **0** o **1** nelle righe per le quali la funzione non è specificata

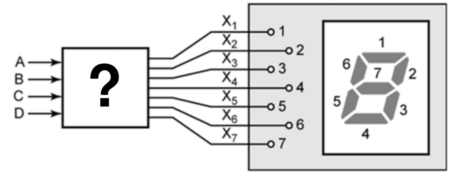
“don't care” per ABC = 001

“don't care” per ABC = 110

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

Esempio di funzione non completamente specificata

Circuito per pilotare un display a 7 segmenti
7-segment decoder



- 4 bit di ingresso
- Funzione di uscita non specificata per ingresso>9 (ABCD>1001)

Espansione in mintermini o maxtermini

- Occorre specificare un valore per i casi “don’t care”
- **Possono essere indifferentemente 0 o 1**
- Ovviamente, la **scelta di 0 o 1 influenzerà la complessità della risultante espansione**
- Successivamente vedremo come le funzioni non completamente specificate sono gestite nella minimizzazione mediante *mappe di Karnaugh*

Esempio

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

- Se assegno X=1 in tutti i casi
  A’B’C’+A’B’C+A’BC+ABC’+ABC=
  = A’B’+A’BC+AB=A’B’+B(A’C+A)=
  =A’B’+AB+A’BC
- Se assegno X=0 in tutti i casi:
  A’B’C’+A’BC+ABC=A’B’C’+BC
- Se assegno 1 alla prima X e 0 alla seconda:
  A’B’C’+A’B’C+A’BC+ABC=A’B’+BC

Espansione in mintermini e maxtermini

$$F = \Sigma\, m(0,3,7) + \Sigma d(1,6)$$

mintermini
"don't care"

$$F = \Pi\, M(2,4,5) \bullet \Pi\, D\,(1,6)$$

maxtermini
"don't care"

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

Analisi e design di circuiti combinatori

Circuiti combinatori

- Gli **output sono funzione solo degli input**
- **Nessuna memoria (stato)**
- Possono essere descritti mediante funzioni Booleane e/o tabelle di verità

Design gerarchico

- Se il problema è complesso, decomporlo in **sotto-problemi**
- Successivamente, risolviamo ciascun sottoproblema
- Infine, **combiniamo le soluzioni dei sottoproblemi** per realizzare il circuito che serve per risolvere il problema di partenza
- Infatti, un circuito elettronico è spesso formato da più ***sottocircuiti*** *(a volte su differenti chip) che risolvono sottoproblemi diversi*
- A volte sottoproblemi ricorrenti …

Realizzazione di circuiti

Design dei Circuiti

Modo *più semplice*: rappresentare una funzione mediante *espansione in mintermini o maxtermini*, e realizzarla usando opportuni gate

Dov'è il problema?

Design dei Circuiti

- L'espansione in maxtermini o mintermini *non corrisponde necessariamente al circuito meno costoso* (minor numero di porte, stesso tipo di porte)
- Quindi, il nostro obiettivo ora è **minimizzare il costo del circuito**

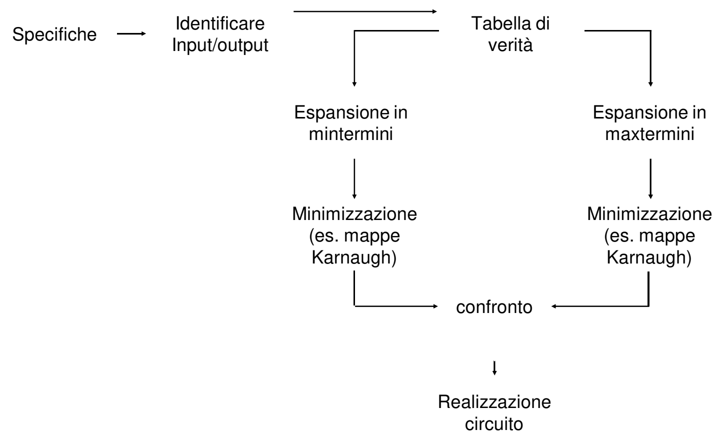
Ma non solo il costo…

- *Velocità*
  - dipende dai tempi di risposta dei circuiti e da quanti **circuiti in cascata** un input deve attraversare per produrre un output
- *Consumo energetico…*
  - vedere data sheet dei circuiti

Processo di sintesi

1. Partire dalle ***specifiche** del circuito*
2. Identificare gli **input** e gli **output** del circuito
3. Produrre la **tabella di verità**
4. Determinare **l'espansione in mintermini e maxtermini**
5. Usare **l'algebra di Boole** o le **mappe di Karnaugh** per identificare **un'espressione equivalente a entrambe**
6. Scegliere la **soluzione a costo minore**
7. **Costruire il circuito**
8. Verifica (es. mediante simulazione)

Sommario



Analisi

Obiettivo: determinare il *comportamento di un circuito a partire dalla sua descrizione* (diagramma del circuito)

Analisi

- Per circuiti semplici (2 livelli) può essere effettuata per ispezione
- Per circuiti a più livelli le cose possono essere più complicate…

Procedura di analisi

1. Identificare input e output
2. Tracciare i *segnali dagli ingressi alle uscite per tutte le combinazioni di ingressi*
3. Determinare *il valore delle uscite*
4. Realizzare la *tabella di verità*
5. Analizzare *aspetti di temporizzazione, consumo*, etc.

- Maggiori dettagli nei corsi di elettronica…