ESERCITAZIONE 2

Sommario

- Circuiti Combinatori

1. Circuiti Combinatori

In un circuito logico combinatorio per ogni istante, il valore delle uscite è funzione del valore degli ingressi in quello stesso istante.

I circuiti combinatori sono *circuiti senza memoria*

In un circuito combinatorio non è possibile che:

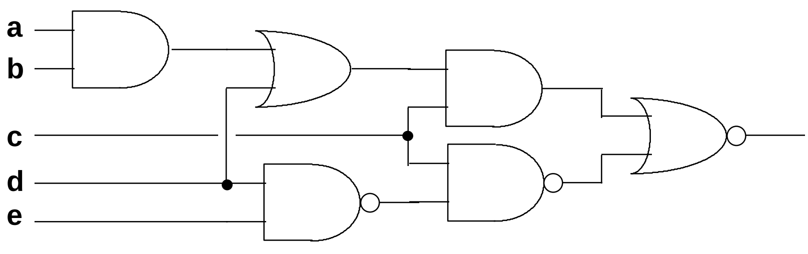
1. siano presenti anelli (o loop);
2. le uscite delle porte siano collegate.

1.1. Esercizi

1. Si disegnino i circuiti corrispondenti alle espressioni seguenti:

- $A * \overline{B}$
- $(A + \overline{B}) * (A + B + \overline{C})$
- $\overline{A} * (B + C * (A * \overline{B} + B))$
- $(A*B + \overline{B}) * \overline{C}$
- $\overline{A} * \overline{B} * \overline{C} + B*C + A*C$

2. Si ricavi l'espressione logica e la tabella della verità del seguente circuito:



2. Transizione da tabella della verità a forma canonica

Alcune definizioni:

1. *minterm* è un'espressione prodotto che contiene in modo affermato o negato tutte le variabili di una funzione;
2. *maxterm* è un'espressione somma che contiene in modo affermato o negato tutte le variabili di una funzione;

3. si definisce *forma canonica* di una funzione booleana la sua espressione in termini di *somma di minterm* oppure di *prodotto di maxterm*.

Una qualunque funzione booleana f(a, b, c, ..., n) è espressa univocamente attraverso le 2 possibili forme canoniche.

Data una tabella della verità per realizzare la rappresentazione della funzione booleana in forma canonica si possono scegliere le due alternative:

1. rappresentare gli 1 della tabella della verità come somme di minterm;
2. rappresentare gli 0 della tabella della verità come prodotto di maxterm.

Esercizio:

Si rappresenti la funzione espressa attraverso la seguente tabella della verità attraverso le 2 forme canoniche.

a	*b*	*c*	*y*
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

1. y (a, b, c) = $\bar{a}*\bar{b}*\bar{c} + \bar{a}*\bar{b}*c + a*\bar{b}*\bar{c} + a*b*\bar{c}$
2. y (a, b, c) = $(a+\bar{b}+c) * (a+\bar{b}+\bar{c}) * (\bar{a}+b+\bar{c}) * (\bar{a}+\bar{b}+\bar{c})$

2.1. Esercizi

2.1.1. Esercizio 1

Disegnare il circuito corrispondente alla seguente tabella della verità:

a	*b*	*c*	*y*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1

2.1.2. Esercizio 2

Ripetere l’esercizio usando i prodotti di somme.

3. Logica a due livelli

Qualunque funzione booleana può essere implementata da un circuito a due livelli di logica. Una logica a due livelli premette di realizzare circuiti aventi un tempo di ritardo minimo: i ritardi sono al massimo la somma dei ritardi introdotti dai due livelli di porte.

Una qualunque espressione *Somma di Prodotti* o *Prodotto di Somme* può essere implementata da un circuito avente 2 livelli di logica.

Utilizzando il teorema di De Morgan è possibile trasformare un circuito a 2 livelli del tipo Somma di Prodotti in circuito NAND-NAND ed un circuito a 2 livelli del tipo Prodotto di Somme in circuito NOR-NOR.

3.1. Esercizi

3.1.1. Esercizio 1

Applicando il teorema di De Morgan trasformare le seguenti somme di prodotti in prodotti negati di prodotti negati e si realizzi il corrispondente circuito del tipo NAND-NAND:

1. $A*\overline{B} + A*B*\overline{C}$
2. $\overline{A}*C + B*\overline{C} + A*B$

3.1.2. Esercizio 2

Applicando il teorema di De Morgan trasformare i seguenti prodotti di somme in somme negate di somme negate e si realizzi il corrispondente circuito del tipo NOR-NOR:

1. $(A+\overline{B}+C) * B$
2. $(A+C) * (A+\overline{B}+C) + (B*\overline{C})$

4. Minimizzazione logica

I parametri per valutare il livello di ottimizzazione di un circuito combinatorio sono:

1. *velocità*;
2. *costo*.

La velocità di un circuito combinatorio si valuta in termini di tempo di ritardo dall'istante in cui si ha una variazione sugli ingressi all'istante in cui tale variazione si è propagata sulle uscite. Per massimizzare la velocità del circuito occorre minimizzare il tempo di ritardo.

Il costo di un circuito combinatorio si valuta nel numero totale:

1. di porte logiche impiegate;
2. di variabili di ingresso per ogni porta logica *fa*(*nin*).

La traduzione diretta in circuito delle forme canoniche non genera un progetto ottimo (o minimizzato).

Una espressione del tipo Somma di Prodotti si dice minima se:

1. contiene il numero minimo di termini prodotto;
2. non è possibile eliminare variabili da alcun termine prodotto.

Una Somma di Prodotti minima corrisponde ad un circuito a minimo costo: numero minimo di porte con fan-in minimo.

Definizioni:

1. un termine prodotto di una funzione è un *implicante* se vale 1 quando la funzione vale 1;
2. un implicante è primo o *principale* se contiene il minor numero di variabili possibili ed eliminando una variabile non è più un implicante;
3. due implicanti si dicono *adiacenti* se differiscono per una sola variabile;
4. un implicante si dice essenziale se è l'unico a coprire un determinato minterm.

Per effettuare la minimizzazione logica occorre:

1. devo calcolare tutti gli implicanti principali;
2. scegliere l'insieme minimo di implicanti principali tale che la somma logica corrisponde alla funzione diminimizzazione.

Torna al Sommario