



Volfango Furgani

OPERAZIONI LOGICHE E ALGEBRA DI BOOLE

23 August 2002

global \$id,\$section; ?>

Operazioni logiche e algebra di Boole:

- [Definizioni a parole delle operazioni logiche](#)
- [Condizioni di indifferenza](#)
- [Fasi del progetto di un circuito combinatorio](#)
- [Forma canonica](#)
- [Lista della spesa \(per i compiti in classe\)](#)
- [Proprietà di dualità](#)
- [Espressione unica dei due teoremi di DeMorgan](#)
- [I due modi equivalenti per ottenere un circuito a soli nand o soli nor](#)
- [Regole per le mappe di Karnaugh](#)
- [La stessa struttura circuitale dà luogo ad applicazioni diverse](#)

Definizioni a parole delle operazioni logiche

[\(Dom. 15\)](#) L'operazione logica **and** dà uno se e solo se tutti gli ingressi sono ad uno. Se almeno un ingresso è zero, l'uscita è zero.

L'espressione algebrica è: $Y = AB$.

L'operazione logica **or** dà zero se e solo se tutti gli ingressi sono a zero. Se almeno un ingresso è uno, l'uscita è uno.

L'espressione algebrica è: $Y = A + B$.

L'operazione logica **not** è unaria perché agisce su un solo ingresso. Dà uno se l'ingresso è a zero. Se l'ingresso è uno, l'uscita è zero. L'uscita dà, quindi, il valore logico opposto all'ingresso. Questa operazione si dice complementazione o complementare l'ingresso.

L'espressione algebrica è: $Y = A$ e si legge A negato.

[\(Dom. 18\)](#) L'operazione logica **nand** dà zero se e solo se tutti gli ingressi sono ad uno. Se almeno un ingresso è zero, l'uscita è uno. L'espressione algebrica è: $Y = AB$.

L'operazione logica **nor** dà uno se e solo se tutti gli ingressi sono a zero. Se almeno un ingresso è uno, l'uscita è zero.

L'espressione algebrica è: $Y = A + B$.

L'operazione logica **orex** dà zero se e solo se tutti gli ingressi sono uguali tra loro. Se gli ingressi sono diversi, l'uscita è uno.

L'espressione algebrica è: $Y = AB + AB$

L'operazione logica **norex** dà uno se e solo se tutti gli ingressi sono uguali tra loro. Se gli ingressi sono diversi, l'uscita è zero.

L'espressione algebrica è: $Y = AB + AB$

Condizioni di indifferenza ([Dom. 23](#))

Un circuito presenta in uscita una condizione di indifferenza se o 1) si richiede agli ingressi una prestazione fisicamente impossibile (premere il tasto n° 12 in una tastiera a 10 tasti) o 2) si presenta agli ingressi una condizione logicamente impossibile quale serbatoio del carburante contemporaneamente pieno e vuoto. Le condizioni di indifferenza si denotano con una x e servono per minimizzare il circuito e/o per segnalare un possibile guasto dei sensori. (serbatoio colmo e vuoto).

Fasi del progetto di un circuito combinatorio ([Dom. 20](#))

- 1) Lettura della descrizione a parole ed identificazione degli ingressi ed uscite del sistema
- 2) Assegnazione delle variabili.
- 3) Assegnazione dei valori logici alle variabili.
- 4) Esecuzione della tabella di verità
- 5) Estrazione della forma canonica scelta e relativa minimizzazione algebrica od esecuzione delle k-mappe.
- 6) Disegno del circuito minimo utilizzando tutti i tipi di porte.
- 7) Disegno del circuito minimo utilizzando l'adatto tipo di porta logica universale (nand per la prima forma e nor per la seconda).
- 8) Redazione della lista della spesa ed identificazione del circuito più conveniente.

Forma canonica ([Dom. 24](#))

Si definisce forma canonica una espressione algebrica nei singoli termini della quale compaiono tutte le variabili o vere o negate.

La prima forma canonica è composta dalla somma di alcuni dei possibili prodotti di tutte le variabili. Questa forma viene chiamata somma di prodotti o forma **SP**. Ogni termine della forma canonica in esame si chiama minterm.

La seconda forma canonica è composta dal prodotto di alcune delle possibili somme di tutte le variabili. Questa forma viene chiamata prodotto di somme o forma **PS**. Ogni termine della forma canonica in esame si chiama maxterm.

Noi chiameremo arbitrariamente prima forma quella minima derivante dalla prima forma canonica e seconda forma quella minima che deriva dalla seconda forma canonica.

Lista della spesa (per i compiti in classe)

Per identificare il numero di porte e quindi di integrati da utilizzare si tenga conto che si trovano facilmente sul mercato porte a 2, 3, 4 ed 8 ingressi. In un normale integrato a 14 pin, di cui due riservati all'alimentazione, si trovano, tenendo conto che una singola porta occupa un numero di pin pari a quello degli ingressi più uno riservato all'uscita:

- 1) Integrati per qualunque funzione logica (anche nel seguito) con quattro porte a due ingressi.
- 2) Integrati con tre porte a tre ingressi.
- 3) Integrati con due porte a quattro ingressi e due pin liberi talora occupati da un not.
- 4) Integrati con una porta ad otto ingressi e quattro pin liberi..
- 5) Integrati con sei porte not.

Proprietà di dualità ([Dom. 27](#))

Data una espressione algebrica che risulti verificata, scambiando and con or, or con and, 0 con 1 ed 1 con 0 si ottiene una seconda diversa espressione algebrica che risulta pure verificata.

Espressione unica dei due teoremi di DeMorgan ([Dom. 35](#))

1) Identificare sotto il segno di negazione complessiva i vari termini della funzione algebrica che debbono essere legati da un'unica operazione logica e precisamente o da solì or o da solì and.

2) Cambiare l'operazione logica che lega i singoli termini (and con or oppure or con and) e spezzare la negazione complessiva in più negazioni sui singoli termini.

I due modi equivalenti per ottenere un circuito a soli nand o soli nor ([Dom. 33](#)) ([Dom. 34](#))

Metodo algebrico. Consiste nel negare due volte l'intera espressione algebrica ottenuta dalle minimizzazione ed applicare una sola volta il teorema di DeMorgan. Disegnare poi il circuito così ottenuto.

Metodo circuitale. Consiste nel disegnare il circuito logico con porte and, or e not. Poi si sostituisce ad ogni porta il circuito equivalente a soli nand per la I forma o soli nor per la II forma. Il risultato è mostruoso. Se ricordate che due negazioni affermano e quindi sono inutili, potete semplificare le coppie di not posti uno dopo l'altro. In tal modo vedrete che spesso il circuito così ottenuto richiede meno integrati di quello di partenza con and, or e not. In futuro capirete, dopo alcuni esercizi che esiste una via brevis: si può sostituire ogni porta del circuito nella I forma con nand e nella seconda con nor. Meraviglia delle meraviglie, il risultato è quello voluto.

Regole per le mappe di Karnaugh ([Dom. 38](#))

Ogni k-mappa ha 2^n caselle ciascuna contraddistinta da un insieme delle n variabili (vere o negate) che chiameremo arbitrariamente indirizzo della casella.

Per minimizzare una funzione espressa come tabella di verità con l'uso delle k-mappe si seguano le regole prescritte:

- 1) Inserire ogni uno (od ogni zero per la seconda forma che è meno usata) nella casella con l'opportuno indirizzo. Inserire anche le x (condizioni di indifferenza). In seguito elemento stà per uno o per zero.
- 2) Raggruppare gli uno o gli zero che risultano adiacenti in gruppi di uno, due, quattro, otto o sedici (2^n) elementi. Tenere conto che due caselle si dicono **adiacenti** se, a colpo d'occhio, sono confinanti per un lato, (non per un angolo in diagonale) o sono poste agli estremi di una riga o di una colonna. La definizione matematica recita che due caselle sono adiacenti quando nei loro indirizzi cambia una sola variabile. Elegante, ma non molto usata negli esercizi.
- 3) I raggruppamenti debbono avere forma quadrata o rettangolare (non da sei elementi, assassini!).
- 4) Ogni elemento deve essere partecipe di almeno un raggruppamento. Un elemento (od una x) può essere compreso in più raggruppamenti diversi. Attenzione: non è obbligatorio raggruppare tutte le x. Raggruppare solo quelle che servono per rispettare il successivo punto 5).
- 5) Importante fonte di errori: i raggruppamenti debbono essere di **estensione massima** ed in **numero minimo**. Tale obiettivo va raggiunto utilizzando anche alcune o tutte le x.
- 6) Per identificare l'indirizzo di ogni raggruppamento scrivere le variabili che non cambiano valore dentro al raggruppamento. Ignorare quello che lo cambiano.
- 7) Se abbiamo raggruppato gli uno scriveremo la prima forma, altrimenti la seconda.

La stessa struttura circuitale dà luogo ad applicazioni diverse ([Dom. 52](#))

I esempio. Il mux può fungere, come abbiamo visto, da convertitore parallelo-serie, da generatore di minterm e da selettore di dati con la stessa struttura circuitale, cioè lo stesso circuito fa tre cose diverse se si cambia nome ai piedini ovvero si pensano in modo diverso gli ingressi e le uscite.

II esempio. Allo stesso modo il decoder può servire per identificare un dato od un indirizzo (decodifica), da generatore di minterm e da demux, cioè da convertitore serie-parallelo e da distributore di dati.