Giovanni Schgör (g.schgor)

LOGICA COMBINATORIA

18 November 2005

Simulazione dei circuiti di logica combinatoria

Guida all'uso del programmadidattico di simulazione (logcomb.exe) dedicato alla logica combinatoria.

Prerequisito per la comprensione di tali funzioni, è la conoscenza della Logica Booleana (variabili e relative operazioni). In mancanza di questa, si consiglia di apprenderne prima le basi, seguendo ad esempio l'apposito corso (in Java).

Il programmi proviene da esperienze di formazione aziendale, è scritto in VisualBasic3, ed è scaricabile via Internet (con un click del mouse sul nome).

L'esecuzione richiede la presenza nella directory System di Windows del programma interprete VBRUN300.dll (normalmente presente nella libreria standard di Windows)

Gli argomenti trattati sono

Di ciascun argomento viene dato, nelle pagine richiamate, una breve spiegazione di utilizzo del rispettivo programma di simulazione. Per un più approfondito commento audiovisivo, si rimanda al già citato corso in Java.

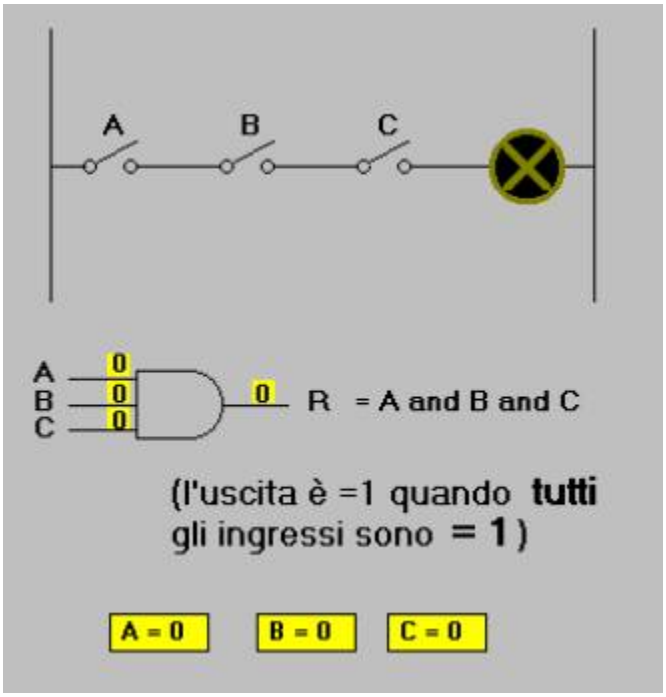
Elementi logici

Questo programma mostra il funzionamento dei principali elementi logici. A differenza dei normali testi di logica, viene però qui mostrato visivamente l'effetto che una combinazione di segnali d'ingresso ha sullo stato dell'uscita di ciascun tipo di elemento.

Una tabella di “verità” è infatti difficilmente memorizzabile, mentre si ricorda più facilmente l’associazione con il corrispondente circuito a contatti (ed il funzionamento di questo è banalmente evidente).

Col programma di simulazione è infatti possibile (con un click nel rispettivo campo di ciascuna variabile) commutare lo stato della variabile stessa, quindi cambiare la combinazione degli ingressi, ottenendo l’indicazione dello stato d’uscita corrispondente.

A titolo di esempio, si riporta la parte principale della funzione **And**.



In modo simile vengono trattati gli elementi **Or, Not, Nand, Nor**.

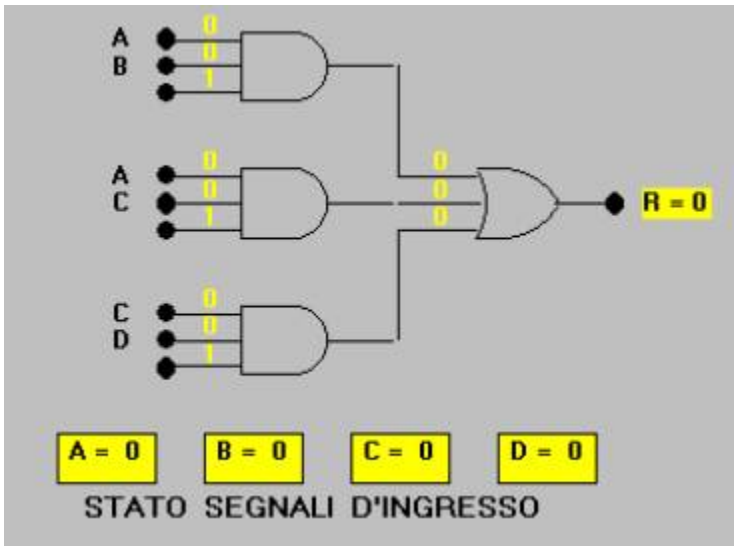
Questo dovrebbe dare la base per l’interpretazione degli schemi logici a più elementi, perché permette di associare il simbolo dell’elemento stesso con la funzione svolta.

(torna all'indice)

Circuito di prova

Come immediata esperienza del funzionamento di un circuito logico, questa simulazione permette di “sperimentare” direttamente il comportamento di una configurazione di **And** e **Or**.

Un'espressione logica del tipo **AB+AC+CD**, può essere quindi facilmente tradotta in schema equivalente e, attraverso questo, vederne lo stato dell'uscita in corrispondenza ad ogni combinazione delle variabili.



(lo stato dei segnali d'ingresso può essere variato con un click del mouse nel rispettivo rettangolo)

Per l'attribuzione dei segnali ai singoli ingressi è previsto una tabella (non mostrata in figura) da cui si può impostare la variabile (ad es. A) o il suo inverso (notA), da applicare.

Con tale configurazione, si può quindi costruire la "tabella della verità" relativa, in funzione di tutte la combinazioni delle variabili in gioco (2^N, se N è il numero di variabili considerate).

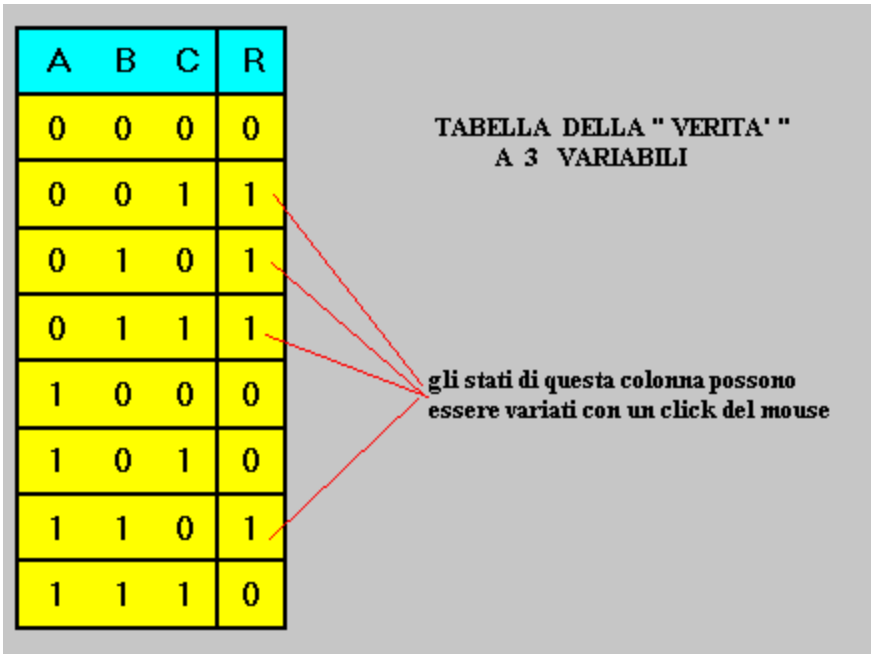
(torna all'indice)

Minimizzazione da tabella della verità

Per realizzare un singolo circuito logico, occorre conoscere a quali combinazioni delle variabili d'ingresso deve corrispondere l'uscita "vera" (cioè lo stato "1") dell'uscita.

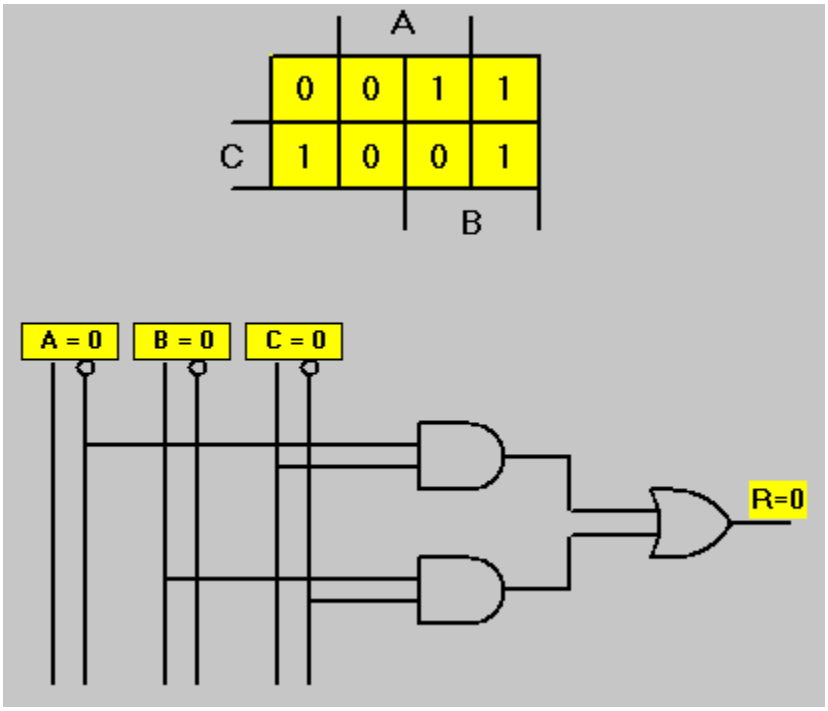
Questo viene normalmente rappresentato dalla "tabella della verità", che contiene tutte le possibili condizioni delle variabili d'ingresso e il corrispondente stato dell'uscita ("0" oppure "1"), e costituisce la "specifica" inequivocabile del funzionamento desiderato.

Considerando qui il caso con 3 variabili (A,B,C), si hanno 8 combinazioni di queste, a ciascuna delle quali deve essere assegnato lo stato dell'uscita (R). Nel programma di simulazione, questo è possibile con un semplice click del mouse.



Nella realizzazione di un circuito logico è poi importante minimizzare il numero di elementi impiegati (ovviamente a parità di funzionamento delle relative condizioni logiche), e per tale funzione è utile ricorrere alla "Mappa di Karnaugh".

Il programma ricava automaticamente questa mappa, traducendola poi in circuito minimizzato.



(nella figura si osservi che invece di 4 And a 3 ingressi, per realizzare la tabella precedente sono sufficienti 2 And a 2 ingressi).

Il programma permette infine la simulazione del funzionamento del circuito ricavato (click nei riquadri delle variabili), per verificarne la corrispondenza alla "specifica" stabilita, cioè alla tabella della verità compilata inizialmente.

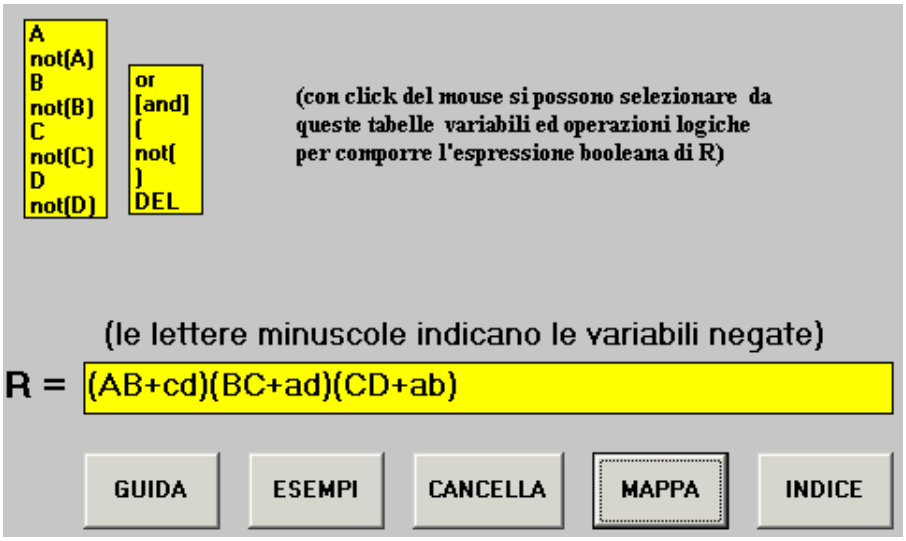
(torna all'indice)

Minimizzazione da espressione logica

Una forma più concisa della specifica di un circuito logico, rispetto alla tabella di verità, è rappresentata dall'espressione booleana equivalente. L'uscita R del circuito deve quindi soddisfare le condizioni logiche contenute nell'espressione.

Il programma di simulazione permette la "scrittura" dell'espressione di R attraverso la scelta dei termini da due tabelle: la prima delle variabili (4 possibili, A,B,C,D, e naturalmente i loro inversi), la seconda delle operazioni logiche fra queste.

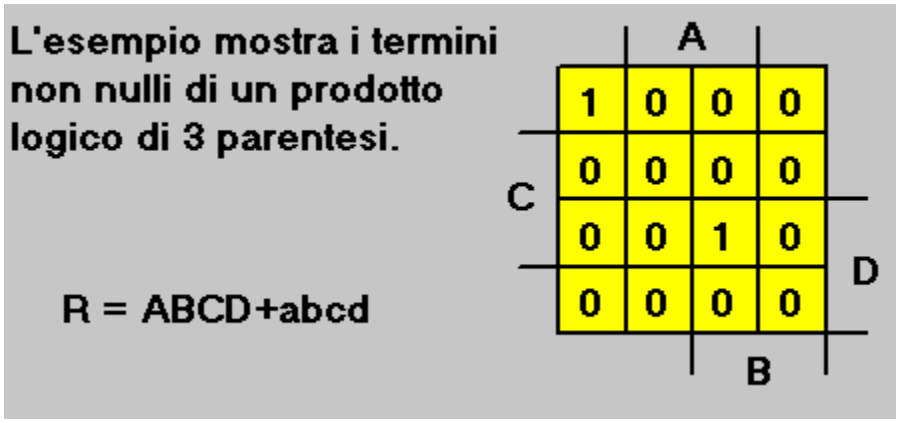
Occorre notare che, data la complessità di interpretazione dell'espressione, vi sono limitazioni nella sua struttura. Si consiglia pertanto di esaminare gli "esempi" previsti nel programma, che riportano anche queste limitazioni nei vari casi.



(la figura mostra la struttura tratta da un esempio. Si noti la convenzione di rappresentare le variabili negate con la corrispondente lettera minuscola : a = Not A)

Dopo che è stata scritta l'espressione, con il tasto "Mappa" è possibile vedere la mappa di Karnaugh corrispondente, da cui, con i soliti criteri dei massimi

raggruppamenti, è possibili ricavare il circuito minimizzato (non indicato dal programma).



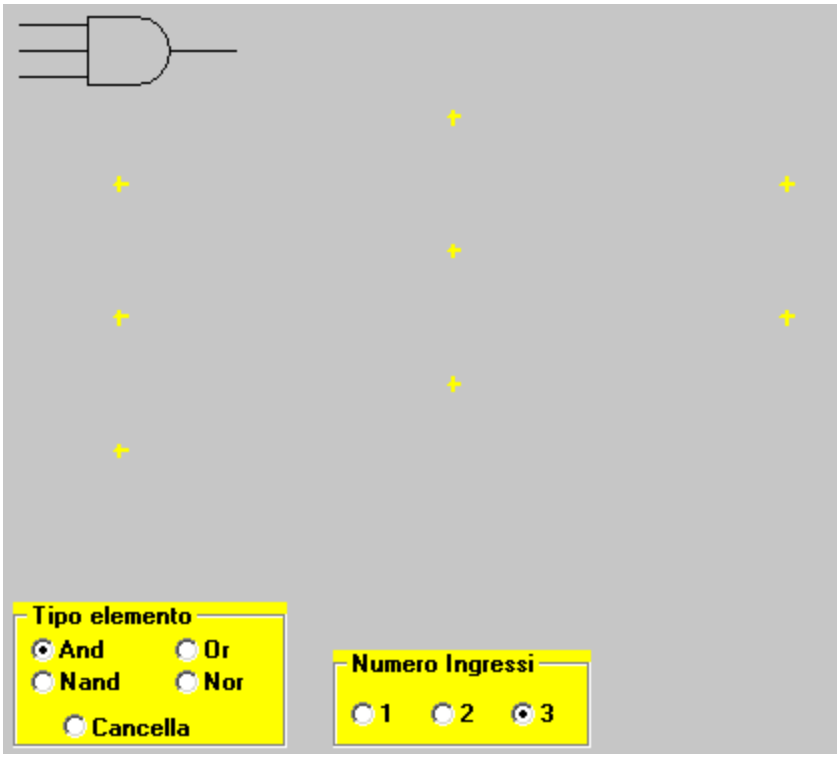
(la figura mostra la costruzione della mappa dell'esempio precedente e la sua conclusione con l'indicazione dell'espressione equivalente minimizzata).

(torna all'indice)

Schema logico libero

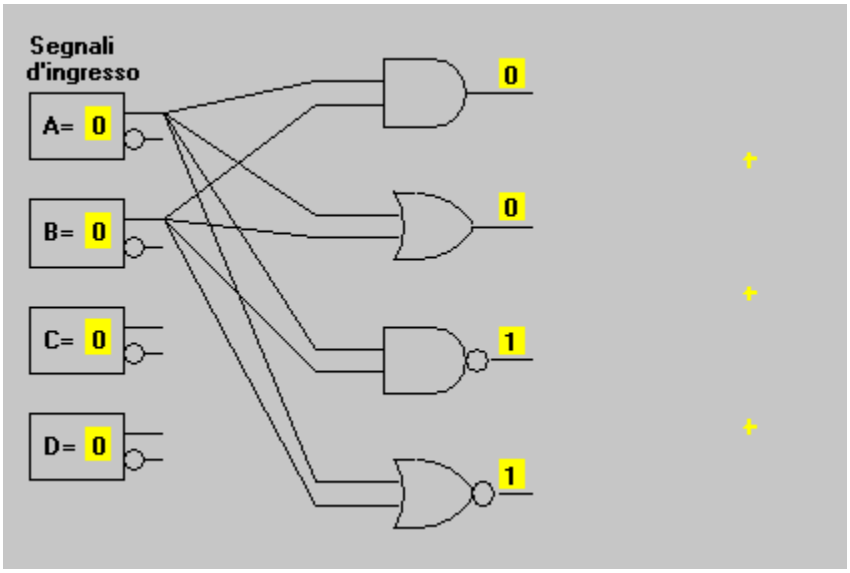
L'ultimo programma di questa serie è il più complesso da utilizzare, ma permette una simulazione completa di un circuito logico costruito liberamente.

Nel campo destinato allo schema, alcune crocette fissano la posizione in cui si può "disegnare" un elemento logico a scelta. Due tabelle determinano questa scelta: con la prima si sceglie il tipo, con la seconda il numero di ingressi. Con un click nella posizione voluta si materializza quindi l'elemento.



(la figura presenta un elemento And a 3 ingressi, posizionato sulla prima posizione)

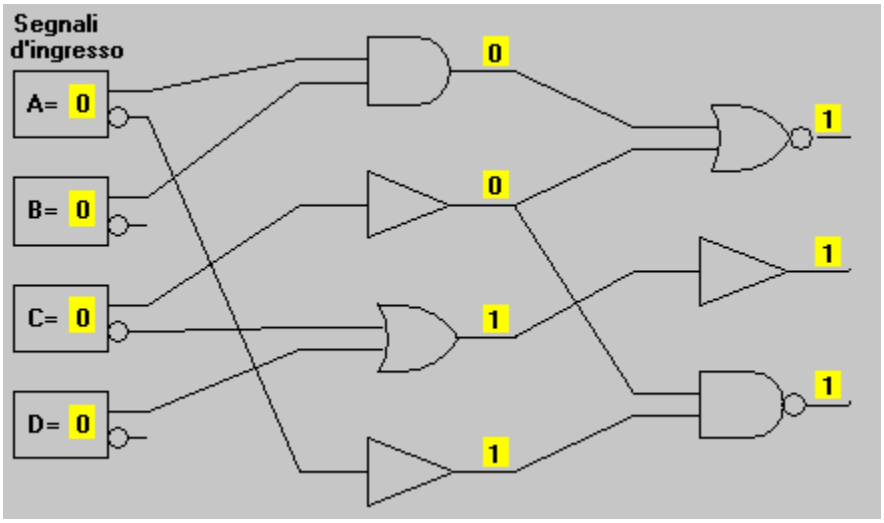
Quando tutti gli elementi voluti sono disegnati, occorre passare alla fase di "collegamento", per collegare ciascun elemento della prima colonna alle variabili di ingresso. (nel caso di elementi su più colonne, occorre poi collegarli fra loro). I collegamenti si realizzano con click prima sul terminale di partenza, poi su quello di arrivo



(la figura mostra la costruzione di 4 elementi collegati ciascuno agli stessi ingressi A e B)>

Durante la fase di funzionamento poi, variando lo stato di A e B, compaiono anche gli stati delle uscite dei singoli elementi. E' dunque possibile, come nell'esempio indicato, confrontare il comportamento dei diversi elementi.

E' però evidente che la maggior utilità sta nel simulare circuiti composti da più elementi su più colonne, come nella figura seguente.



(torna all'indice)