



Giovanni Schgör (g.schgor)

PROBLEMI DI LOGICA E VISSIM3

26 January 2009

Premessa

Ho l'impressione che la **logica booleana** non sia molto popolare fra gli studenti italiani.

Sembra una delle tante cose che si devono imparare (come certe poesie imparate a memoria) e che poi svaniscono nel tempo.

E' un peccato, perché potrebbe rivelarsi più utile del previsto nella progettazione di apparati elettronici ed anche nello sviluppo di software, laddove vi siano da considerare precise condizioni di funzionamento.

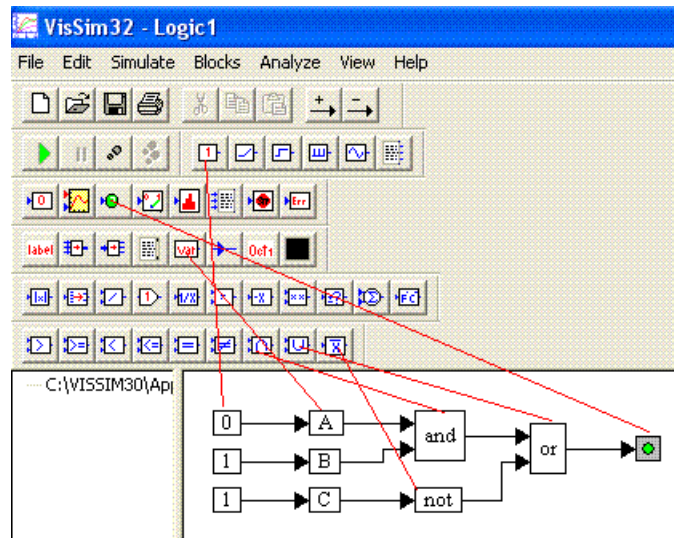
Quello che intendo dire è che spesso manca il lato applicativo dell'apprendimento: magari uno sa che esiste la teoria, ma in quanto poi a metterla in pratica...

Il presente articolo, vuole essere dunque uno stimolo ad un più concreto utilizzo della logica simbolica, tenendo conto delle recenti possibilità di **simulazione** offerta dai programmi disponibili in Rete.

Il VisSim3

A seguito degli articoli [\[1\]](#) [\[2\]](#) di [RenzoDE](#), che segnalano la disponibilità del programma di simulazione gratuito **VisSim3** (per soli scopi didattici, scaricabile da [qui](#)), possiamo utilizzarne le semplici procedure di simulazione applicate agli elementi booleani.

Per entrare subito in argomento, vediamo come osservare il comportamento un piccolo circuito logico.



La "costruzione" del circuito è immediata : clickando col mouse sulle singole icone, è possibile sistamarle nella pagina nella posizione e quindi nella sequenza voluta.

Ciascun blocco deve essere eventualmente completato con i dati relativi (ad es. ad una variabile deve essere dato un nome, attivando prima il pulsante destro del mouse).

Sempre con il mouse è poi possibile collegare i vari blocchi puntando sull'ingresso e (col pulsante sinistro premuto) andare verso l'uscita da cui il segnale deve pervenire.

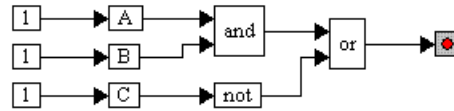
La caratteristica che dovrebbe favorire l'uso di questo simulatore da parte di non-esperti è l'identificazione dei vari elementi (non simbolici, come in altri programmi ma con chiaramente indicata la funzione logica svolta).

Non occorre poi un'approfondita conoscenza della Logica Booleana, per seguire lo svolgimento dei segnali fra i vari elementi e determinarne per ciascuno lo stato dell'uscita.

Normalmente interessa soprattutto stabilire lo stato finale (rappresentato nel circuito dalla lampada: verde per stato "0" e che diventa rossa per stato "1"), in base alla combinazione degli stati dei segnali agli ingressi (A,B e C).

Nel caso specifico, che l'uscita deve essere attivata solo se A **e** B sono attivati ("1") **o anche se** C è disattivato ("0"). L'espressione booleana che corrisponde a questo è $(A \text{ And } B) \text{ Or } (\text{Not } C)$, quindi il circuito non fa che riprodurre tale espressione.

Si vede allora che per attivare l'uscita, dati gli stati degli ingressi come nella precedente figura, basta commutare A da "0" ad "1"



(stesso risultato si sarebbe ottenuto anche commutando C da "1" a "0")

Una volta completato lo schema ed assegnati i valori delle singole variabili, la simulazione avviene su semplice comando con Simulate/Go.

Minimizzazione dei circuiti

Spesso nella stesura di un circuito logico si è portati, magari dallo stesso enunciato del problema, ad utilizzare elementi che risultano "ridondanti" (cioè superflui ai fini delle relazioni logiche svolte).

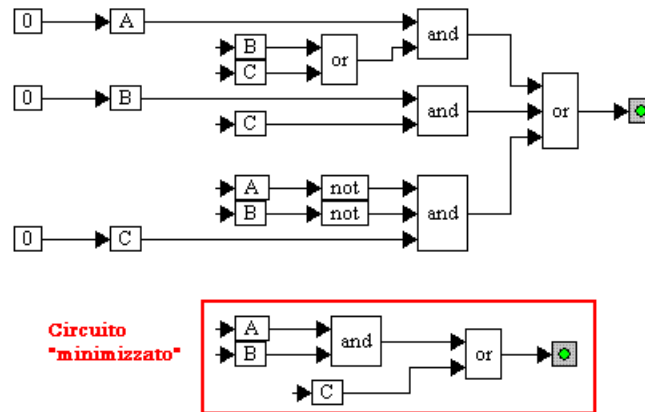
E' ovvio che un buon progettista non dovrebbe trascurare la verifica di una possibile riduzione degli elementi utilizzati, con evidenti vantaggi sia economici che funzionali.

Eccone un esempio.

Un segnale R dipende dallo stato di altri 3 (A,B e C), e deve essere attivo ("1") nelle seguenti condizioni:

- con A=1, ma con B o C =1
- sia B che C =1
- sia A che B =0, ma C=1

Un progettista potrebbe, prendendo alla lettera l'enunciato, procedere alla stesura del circuito in questo modo:



E' infatti evidente la corrispondenza dello schema logico rispetto alle condizioni poste, ma questo, pur funzionando correttamente, non sarebbe la soluzione migliore.

Si può infatti "minimizzare" l'impiego di elementi, applicando i metodi dell'algebra booleana (in particolare le mappe di Karnaugh), ottenendo ad esempio il circuito sottostante (incorniciato in rosso), che riproduce tutte le condizioni poste con meno della meta' degli elementi logici utilizzati nella prima stesura.

Quello che si vuole qui mettere in risalto è la possibilità, con VisSim3, di un'immediata verifica dell'effettiva corrispondenza delle due configurazioni per tutte le possibili combinazioni degli stati degli ingressi.

Soluzioni particolari

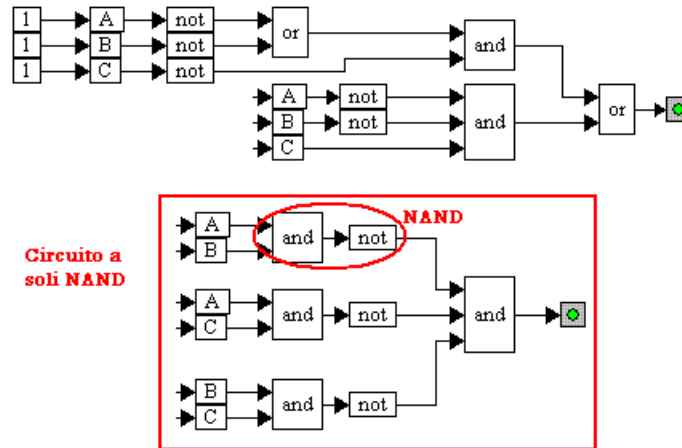
Spesso il progettista si trova costretto all'uso esclusivo di determinati elementi e quindi la stesura dello schema logico deve essere fatta tenendo conto di tali vincoli.

Tipico ad esempio il caso di progetto utilizzando solo elementi di tipo NAND.

Ecco anche qui un esempio applicativo:

Una lampada d'allarme deve accendersi in funzione dello stato di 3 variabili (A,B e C), nelle seguenti condizioni:

- quando $C=0$ e almeno uno degli altri 2 segnali A o B è $=0$
- quando $C=1$ ed entrambi i segnali A e B sono $=0$



Lo schema superiore mostra la soluzione ricavata dalla semplice "traduzione" dell'enunciato, mentre quello incorniciato in rosso rappresenta la soluzione con l'utilizzo di soli elementi NAND (anche quello di uscita, indicato come AND, perché nella realtà il comando effettivo della lampada avviene con segnale "0").

Si noti, per inciso, che la prima stesura richiederebbe 9 elementi logici, mentre nella seconda se ne utilizzano 4.

Conclusioni

Si sono mostrate le possibili semplificazioni circuitali senza volutamente entrare nei dettagli applicativi dei metodi booleani su cui si basano queste semplificazioni (il Forum di ElectroYou è sempre aperto per questo).

Si è invece voluto sottolineare la possibilità di simulazioni in modo da poter verificare rapidamente la rispondenza di un circuito ai requisiti voluti (cosa non sempre scontata), prima di procedere a realizzazioni di prototipi, guadagnando così tempo e denaro.

Estratto da ["http://www.electroyou.it/mediawiki/index.php?title=UsersPages:G.schgor:articolo12"](http://www.electroyou.it/mediawiki/index.php?title=UsersPages:G.schgor:articolo12)