



Giovanni Schgör (g.schgor)

LOGICA SEQUENZIALE

24 November 2005

Simulazione dei circuiti logici sequenziali

Questo articolo è una guida all'uso di un programma didattico dedicato all'apprendimento del funzionamento degli elementi fondamentali e dei circuiti che, con questi elementi, svolgono funzioni logiche sequenziali.

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

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

Il programma ([logseq.exe](#)) tratta i seguenti argomenti:

1. - [S-R Flip Flop](#)
2. - Esempio di applicazione ([serbatoio](#))
3. - [Temporizzatore](#)
4. - [Sequenze a fasi](#)
5. - Esempio di [automatismo sequenziale](#)
6. - Simulazione [comandi ascensore](#) (eseguibile da: [simasc.exe](#))

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.

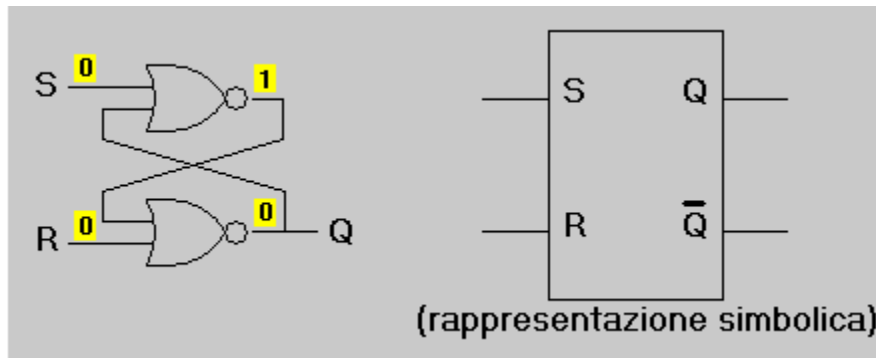
SR Flip Flop

Mentre nella **logica combinatoria** lo stato delle uscite è stabilito univocamente dalla combinazione degli stati d'ingresso, quindi si riproduce invariato ogni volta che si verificano quelle condizioni, ciò non succede nei circuiti di **logica sequenziale**.

In questi ultimi infatti lo stato delle uscite dipende anche dalla sequenza temporale degli eventi, cioè da come si sono susseguiti nel tempo gli stati degli ingressi. Quindi, a parità di combinazione degli stati d'ingresso possono verificarsi stati diversi delle uscite.

Nella logica sequenziale vi sono infatti elementi che "memorizzano" una certa combinazione di stati, condizionando così l'evolversi delle successive uscite.

Elemento principale per questa funzione di memorizzazione è il cosiddetto Set-Reset Flip Flop, combinazione di elementi logici collegati in modo da condizionare lo stato d'uscita Q, in funzione di come si sono precedentemente presentati gli stati degli ingressi S (Set) ed R(Reset)



Il programma di simulazione mostra infatti che se, con R allo stato "0", all'ingresso S si presenta lo stato "1", l'uscita Q diventa "1" e permane in questo stato anche quando S ritorna allo stato "0". (Il ripristino di Q=0 avviene solo quando R=1).

La simulazione avviene semplicemente battendo sulla tastiera del calcolatore rispettivamente i tasti S ed R.

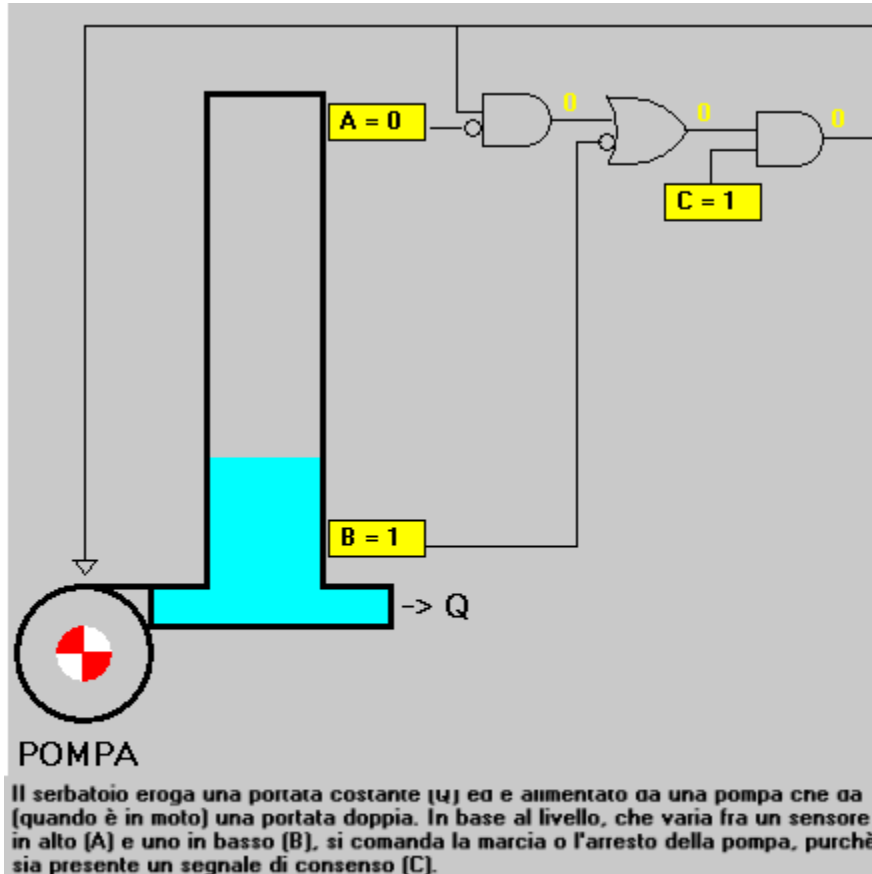
(torna all'[indice](#))

Esempio di applicazione (serbatoio)

Come immediato esempio di applicazione della logica sequenziale, nel programma di simulazione viene presentato il caso del controllo automatico del riempimento di un serbatoio idraulico.

IL problema è abbastanza semplice: quando il livello del serbatoio scende alla posizione del rilevatore B (basso), si deve mettere in funzione la pompa che fa crescere il livello del liquido fino al rilevatore A (alto). A questo punto la pompa si ferma ed il livello ricomincia a scendere.

La parte interessante della simulazione sta nel seguire lo stato delle uscite degli elementi logici costituenti il circuito di controllo, funzione dei segnali provenienti dai rilevatori A e B.



E' evidente che la pompa si deve avviare quando B=0 e che fermarsi quando A=1, ma si osservi che normalmente è A=0 e B=1 (sia che il serbatoio in fase di svuotamento che di riempimento), quindi senza "memorizzazione" la pompa non potrebbe rimanere in moto.

L'esempio, pur nella sua semplicità, è indicativo dei fondamenti della logica sequenziale che, vale la pena di ricordarlo, è la base di tutti gli automatismi industriali.

(torna all'[indice](#))

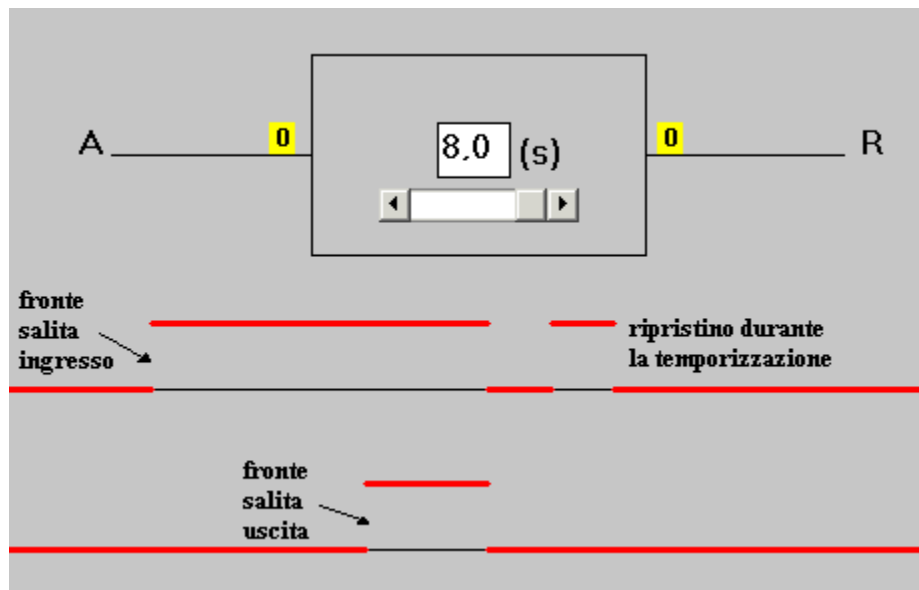
Temporizzatore

L'altro elemento fondamentale negli automatismi industriali oltre il S-R FF, è il temporizzatore.

Questo elemento, che ha innumerevoli versioni, introduce negli automatismi logici ben definiti tempi di ritardo, in modo da condizionare il verificarsi di eventi.

Nella sua versione più semplice, il temporizzatore ritarda il tempo di salita (dallo stato "0" allo stato "1") del segnale della sua uscita, rispetto alla salita del segnale applicato al suo ingresso.

Nel programma di simulazione si vede come, una volta stabilito il tempo di ritardo, il fronte del segnale A sia "dilazionato" nel segnale R (sempreché A non torni a "0" durante lo scorrere del ritardo, il che ripristina daccapo il tempo). Il cambiamento di stato del segnale A è ottenuto semplicemente battendo il tasto A sulla tastiera del calcolatore.



Oltre la ovvia funzione già citata di comando temporizzato, il dispositivo ha importanti applicazioni nella diagnostica dei moderni impianti. Se infatti una data azione non avviene entro un tempo massimo stabilito, la relativa segnalazione può contenere anche la causa che ne ha impedito lo svolgimento, rendendo così facile ed immediato l'intervento di manutenzione.

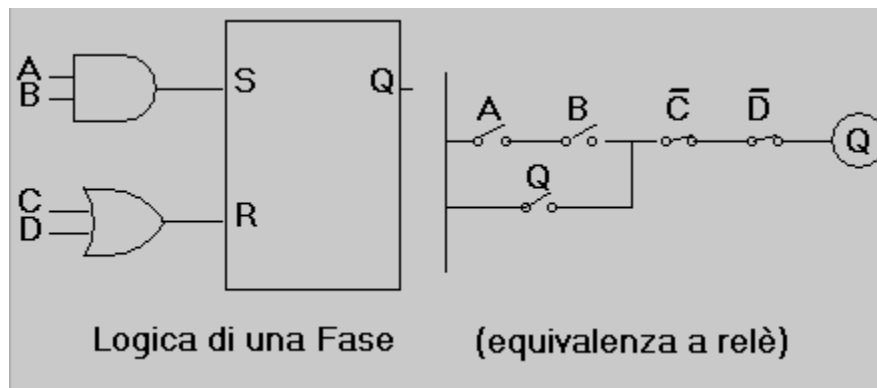
(torna all'[indice](#))

Sequenza a fasi

Si è già detto dell'importanza del S_R Flip Flop negli automatismi industriali. Poiché ogni ciclo di un processo automatizzato è composto da successioni di azioni elementari, ognuna delle quali deve avvenire solo in presenza di ben stabilite condizioni, è fondamentale il concetto di "fase" in una sequenza.

Nel progetto di una sequenza automatizzata si deve infatti stabilire prima la successione delle singole azioni, poi tutte le condizioni che devono permetterne il singolo svolgimento e da ultimo le condizioni che determinano la fine di ciascuna azione. Ad ogni azione si associa quindi il numero di fase che stabilisce l'ordine in cui queste devono avvenire.

La "logica" di una fase è concettualmente semplice, ed è illustrata nella figura seguente:



Come si può vedere, l'attivazione della fase (Set) avviene quando si verificano tutte le condizioni stabilite (qui per semplicità rappresentate dai soli 2 segnali A e B), il che porta ad "1" il segnale Q, associato al comando dell'azione che deve essere svolta in quella fase.

La fase termina (Reset) quando si verifica una qualsiasi delle condizioni previste per il suo annullamento (qui indicato dai segnali C e D, che porrebbero essere l'una l'avvenuto completamento dell'azione, col passaggio alla prossima fase, l'altro un arresto comandato, ad es. per ragioni di emergenza).

Per rendere più evidente la funzione della fase, in figura è stata rappresentata anche l'equivalente schema a relè, che rende immediata la comprensione della logica relativa.

L'applicazione sistematica di questo concetto, semplifica quindi la realizzazione e la manutenzione degli impianti automatizzati. Per quanto complessi essi siano, si possono in questo modo ridurre ad un insieme di schemi elementari (anche se poi il loro numero può essere molto grande).

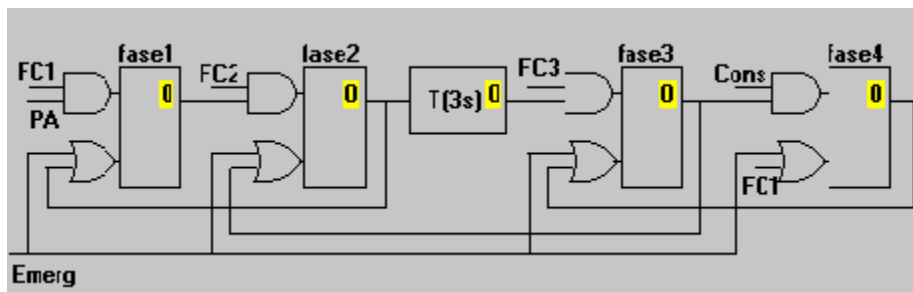
Si sottolinea che questi concetti sono validi anche nell'applicazione degli attuali "Controllori Logici Programmabili" (PLC), servendosi delle specifiche istruzioni proprie del particolare linguaggio adottato.

(torna all'[indice](#))

Esempio di sequenza a fasi

Come esempio applicativo del concetto di fase di una sequenza, nel programma di simulazione viene riportato un semplice automatismo di movimentazione.

Le funzioni da svolgere vengono illustrate all'attivazione del pulsante "Testo" e lo schema funzionale è quello qui sotto riportato.



E' subito evidente che vi sono 4 fasi, e dovrebbero anche apparire chiare le condizioni per l'attivazione di ciascuna di loro. E' ad es. facile riconoscere che si entra nella fase1 (inizio della sequenza automatica) con il pulsante di avviamento PA, ma solo se c'è anche la presenza del segnale del finecorsa FC1 (che indica che tutto è a posto per iniziare).

Si noti che la la fase successiva (fase2) sarà attivata quando sarà raggiunto il finecorsa FC2, e che questa attivazione comporta il Reset della fase1. Nella fase2 viene attivato un temporizzatore che darà il comando di movimento solo dopo 3 s, per raggiungere la posizione del finecorsa FC3. Raggiunto questo si entra in una fase di attesa (fase3) di un "Consenso a proseguire", per attivare la fase4, che fa tornare al punto di partenza (FC1).

Come si vede la sequenza è elementare, ma rappresentativa delle reali situazioni impiantistiche. La simulazione ne mostra lo svolgimento, che può quindi essere osservato associandolo allo stato delle varie fasi (attenzione all'indicazione i nei riquadri gialli).

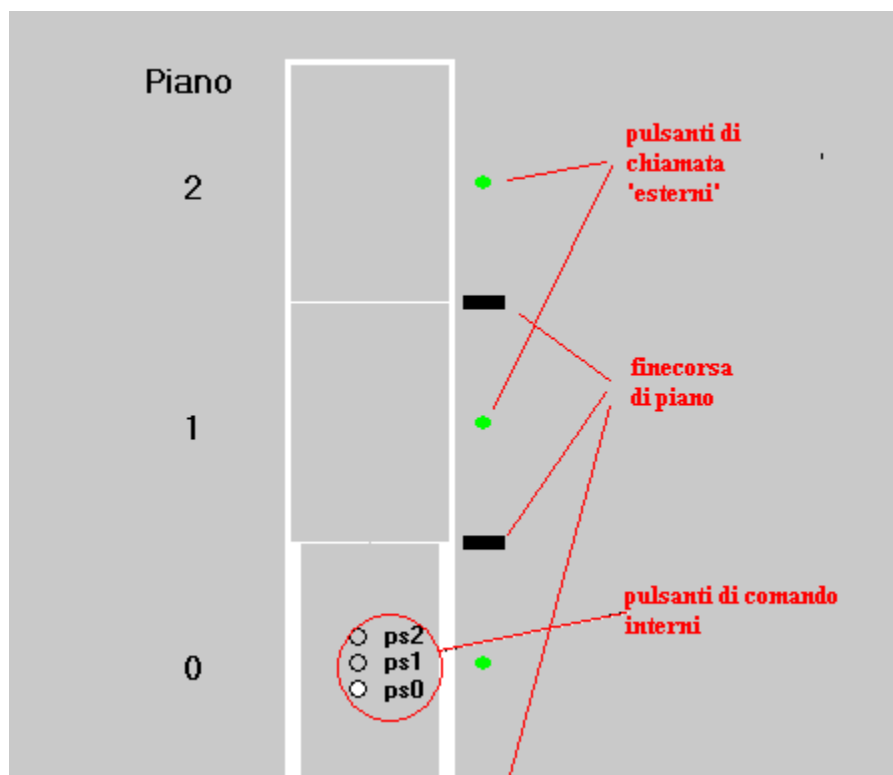
Completa la simulazione la possibilità di arresto del ciclo automatico in "Emergenza" (si osservi però che per il ripristino in automatico nella realtà, occorre riportare il sistema in posizione FC1)

(torna all'[indice](#))

Simulazione ascensore

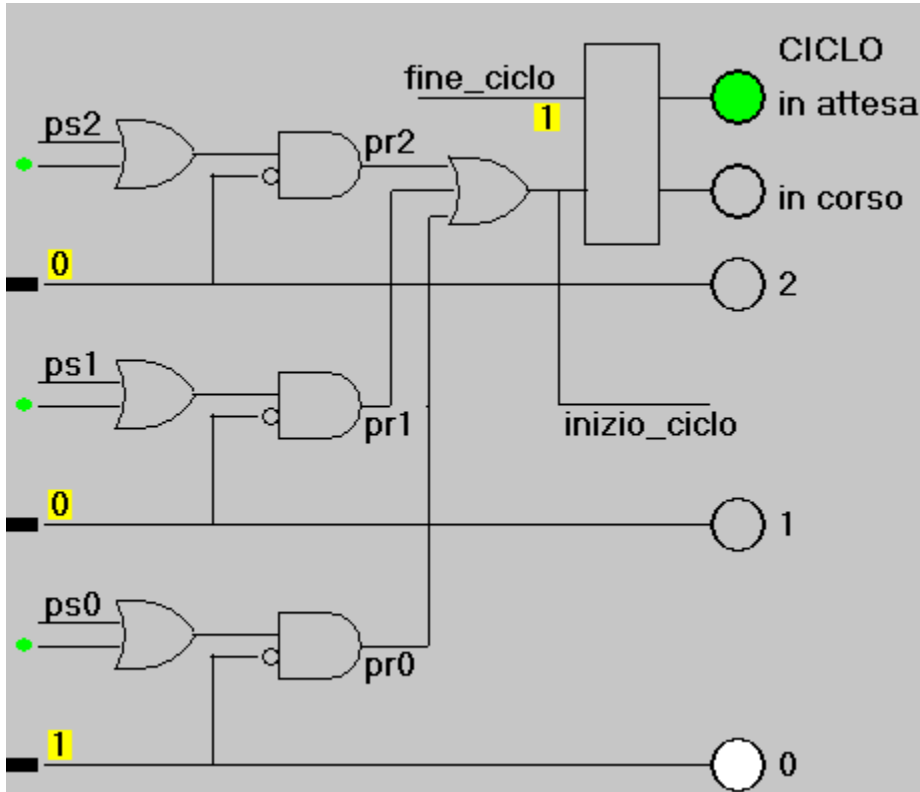
Come esempio concreto di automatismo sequenziale, viene simulato il controllo di un ascensore (la scelta è dovuta al fatto che il funzionamento di un ascensore è noto a tutti, quindi le sequenze sono facili da seguire intuitivamente).

Naturalmente, per una più facile comprensione, i circuiti di controllo sono stati semplificati e ridotti all'essenziale, ma comunque possono dare un'idea abbastanza esatta dell'automatismo.

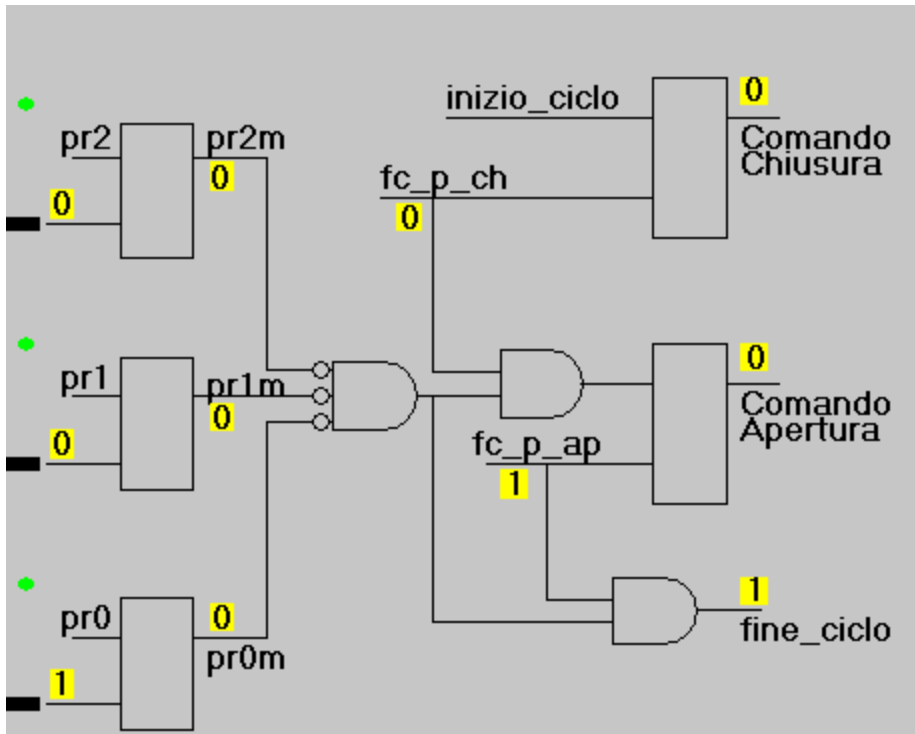


Come si può vedere dalla figura, l'ascensore simulato si muove fra 3 piani, comandato dai "pulsanti" indicati (per comandarli, fare click con il mouse). All'interno si "accende" (bianco) il pulsante relativo al piano in cui si trova l'ascensore.

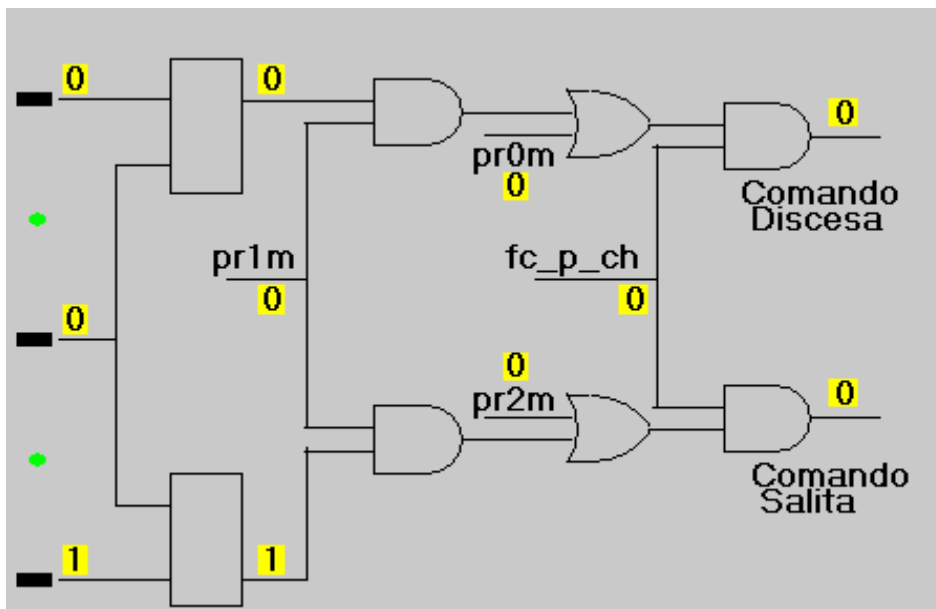
Il programma riporta tutte le informazioni utili a seguire l'evolversi dei comandi, che si riducono a 3 schemi fondamentali: il comando luci, comandi porta, comandi motore. Appositi pulsanti permettono la scelta dello schema da osservare. Qui sotto è riportato quello delle luci



Quando si attiva un pulsante di comando (interno o esterno) si "prenota" il piano desiderato (pr..) e si dà inizio al ciclo. Evidentemente questo inizia con la chiusura delle porte.



La figura rappresenta lo schema "comando porte" in cui si memorizza la richiesta (pr.m) e si comanda la chiusura. Ovviamente questo comando termina con il segnale di porta chiusa (fc_p_ch), ma ciò dà avvio al comando del motore che deve muovere l'ascensore.



La figura rappresenta il "comando motore", in cui si deve decidere se l'ascensore deve salire o scendere. (La semplificazione del caso, con soli 3 piani, rende facile questa scelta).

E' poi ovvio che il movimento deve arrestarsi in corrispondenza del piano prenotato, e qui avviare il comando di apertura porta. Il ciclo termina quando la porta si è aperta completamente (fc_p_ap)..

Quello che deve essere sottolineato è che l'evoluzione temporale dei segnali, propria della logica sequenziale, complica notevolmente la progettazione e la stessa interpretazione dello schema rispetto alla "staticità" della logica combinatoria. Ma, come si è già detto, tutta l'automazione industriale pone problemi sequenziali.

(torna all'[indice](#))