

Architettura delle reti sequenziali

di Anna Pugliese

Le reti sequenziali differiscono dalle reti combinatorie per il solo fatto che alcuni, fra i valori prodotti come uscite della rete, vengono ripresentati agli ingressi, per concorrere a determinare un nuovo input. Detto questo, è evidentemente necessario fermarsi un attimo per richiamare quei concetti delle reti combinatorie, indispensabili per la trattazione di quelle sequenziali. La cosa non richiederà più di tanto, per il fatto di aver voluto dare alla presente trattazione, un taglio che ci permetta di giungere fino in fondo all'argomento senza dover rinunciare a quegli aspetti che, pur essendo da «addetti ai lavori», sono pur sempre la parte più interessante; un taglio, per dirla in altri termini, che trascura, laddove è possibile, il problema della sintesi della rete, lasciando maggiore spazio alla trattazione delle possibili architetture di rete

La figura 1 illustra l'architettura esterna di una rete combinatoria avente m ingressi ed n uscite. Essa è già nota ai lettori del precedente numero della rubrica, cui rimandiamo per una più ampia trattazione sulle reti combinatorie. Scopo di una rete combinatoria è quello di produrre, sui morsetti d'uscita, n valori binari (sotto forma di opportuni valori fisici) che esprimono il risultato di una ben precisa elaborazione degli m valori binari presenti sui morsetti d'ingresso. Se in un qualsiasi istante cambiasse l'input della rete, dopo un tempo Δ l'output della rete presenterà il corrispondente valore d'uscita, dove Δ è un intervallo temporale detto «tempo di attraversamento della rete». L'interno della rete è composto da circuiti AND, OR, e NOT e da collegamenti fra i morsetti d'ingresso, i vari circuiti posti opportunamente in cascata fra loro, e chiaramente i morsetti d'uscita. Caratteristica intrinseca di ogni rete combinatoria è che la funzione logica da essa calcolata (vale a dire: la legge mediante la quale gli output sono prodotti a partire dagli input) è indipendente dal tempo: ogniquale volta si presenti all'ingresso della rete una stessa configurazione di valori binari, verrà presentato in uscita lo stesso risultato.

Reti sequenziali

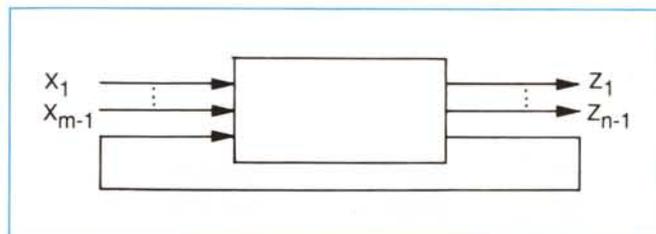
Riconsideriamo la figura 1. Proviamo a prendere una delle uscite binarie, ad esempio Z_n e colleghiamola con uno degli ingressi, diciamo X_m . Cosa succe-

derà? Avremo ottenuto una rete sequenziale che, con ogni probabilità, si comporterà in modo «pazzo». Vediamo perché. Chiamiamo X la configurazione di m valori binari all'ingresso della rete. Trascorso un tempo Δ a partire dall'istante t_0 in cui X è stata applicata ai morsetti, avremo che la configurazione Z si sarà presentata agli n morsetti d'uscita della rete, dove $Z=f(X)$ ed f è la funzione calcolata dalla rete (combinatoria). Al tempo $t_0+\Delta$ il valore di Z_n sarà diventato un nuovo valore per X_m ; se questo nuovo valore è diverso dal vecchio, avremo che la nuova configurazione in ingresso X' sarà diversa dalla precedente e, presumibilmente, otterremo successivamente un $Z'=f(X')\neq Z$, con conseguente possibile variazione di Z_n . Non è difficile rendersi conto del fatto che potremmo finire col trovarci davanti ad una rete che produce ciclicamente una serie di uscite, senza mai stabilizzarsi. Ciò malgrado, supponiamo che la dea bendata sia dalla nostra e che la rete sia sempre in grado di stabilizzarsi, vale a dire di produrre una uscita in cui $Z_n=X_m$, per cui fino a che non cambierà l'input della rete, avremo un output invariato sull'uscita. Osserviamo la figura 2. In essa è mostrata la rete sequenziale della quale stiamo parlando. Notiamo che tale rete, ha $m-1$ ingressi ed $n-1$ uscite dal momento che Z_n ed X_m sono diventati dei morsetti interni al sistema, e non più di input/output. Osserviamo il diagramma temporale della figura 3, esprime un esempio di cosa potrebbe succedere



Figura 1

Figura 2 - Un prototipo di rete sequenziale ottenuto da quella combinatoria della figura 1, con una semplice variazione.



col trascorrere del tempo, sulla nostra rete. Al tempo t_0 , la configurazione $X = x_1 x_2 \dots x_{m-1}$ si presenta ai morsetti d'ingresso; dopo un tempo Δ l'uscita $Z = z_1 z_2 \dots z_{n-1}$ sarà comparsa sui morsetti d'uscita, ma sarà necessario attendere ancora un intervallo di tempo t_s (tempo di stabilizzazione) perché l'uscita Z_n riflettendosi sull'ingresso X_m provochi una successione di variazioni sull'uscita Z che terminerà solo quando otterremo un'uscita in cui $z_n = x_m$, vale a dire una uscita stabile. t_s è, per ipotesi, il massimo tempo necessario perché ciò avvenga. Ne consegue che a partire dall'istante $t_0 + \Delta + t_s$ la rete è pronta a ricevere un nuovo input, il che avverrà all'istante t_1 , e così via.

Due considerazioni. La prima, banale, è che una eventuale variazione dell'input della rete, prima che questa si sia stabilizzata, porta inevitabilmente ad un non corretto funzionamento della rete. La seconda è fondamentale. L'input iniziale della rete, è costituito non solo da $X = x_1 x_2 \dots x_{m-1}$ ma da x_m che chiameremo «stato interno della rete», in questo caso: stato interno iniziale; l'uscita Z della rete dipenderà ovviamente anche da tale stato interno, non solo, ma ogni uscita della rete dipende dallo stato interno attuale il quale a sua volta, essendo una parte dell'uscita complessiva precedente, è funzione dell'input precedente e dello stato interno precedente. In altri termini, detto $S(t)$ lo stato interno della rete al tempo t , avremo che:

$$z(t + \Delta + t_s) = f_z(x(t), s(t)) \quad (1)$$

$$s(t + \Delta + t_s) = f_s(x(t), s(t)) \quad (2)$$

Le equazioni 1 e 2 definiscono il modello matematico delle reti sequenziali. Da questo punto di vista, scopriamo quindi che le reti sequenziali sono macchine mediante le quali è possibile implementare il calcolo di coppie di funzioni, una funzione f_z per il calcolo dell'uscita della rete ed una funzione f_s per il calcolo del nuovo stato interno della rete.

La figura 4 mostra molto chiaramente quanto abbiamo appena detto.

Giunti a questo punto il discorso sulle reti sequenziali sarebbe finito, se ci dimenticassimo dell'ipotesi semplificativa nella quale ci siamo posti; in altri termini: se ci mettiamo nell'ipotesi che la benda cada giù dagli occhi della dea di

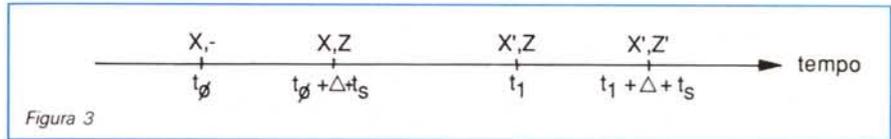


Figura 3

cui sopra è doveroso concludere che ella non possa più favorirci una volta resasi conto di chi gli sta davanti, per cui: addio alle speranze di stabilizzazione della rete in un tempo massimo t_s .

Preso atto, per l'infinitesima volta, della nostra «jella nera», rimbocchiamoci le maniche e vediamo come è possibile forzare la rete ad una stabilizzazione. Esistono 2 metodi per farlo. Il primo consiste nell'apportare una vera e propria modifica all'architettura della rete, mentre il secondo, cui accenneremo solamente, è basato su tecniche di trasformazione della tabella di flusso della rete. Esamineremo entrambi i metodi nei due successivi paragrafi.

Reti sequenziali sincrone

In un certo istante t_0 , la configurazione X si presenta all'ingresso della rete, concorrentemente ad una configurazione SIP, rappresentante lo stato interno presente della rete, e codificata mediante un certo numero di morsetti d'ingresso provenienti, mediante opportuni «anelli di retroazione» da un ugual numero di morsetti d'uscita. Osserviamo la figura 5. Trascorso un certo tempo Δ , come al solito necessario a garantire l'attraversamento della parte combinatoria della rete, l'uscita $Z = f_z(X, S)$ è disponibile sugli appositi morsetti, mentre p valori binari vengono presentati all'in-

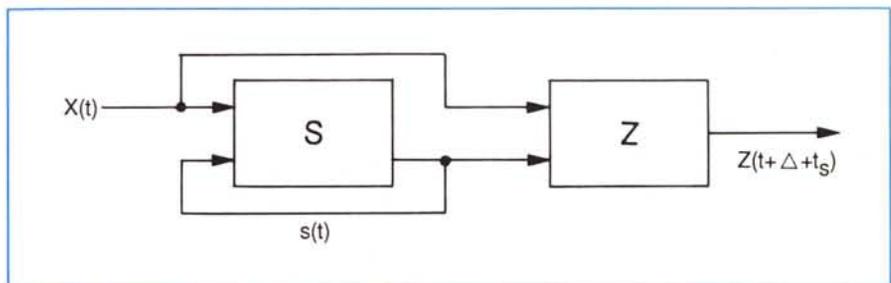


Figura 4 - Una rete sequenziale con evidenziate le parti combinatorie per il calcolo delle funzioni f_s ed f_z .

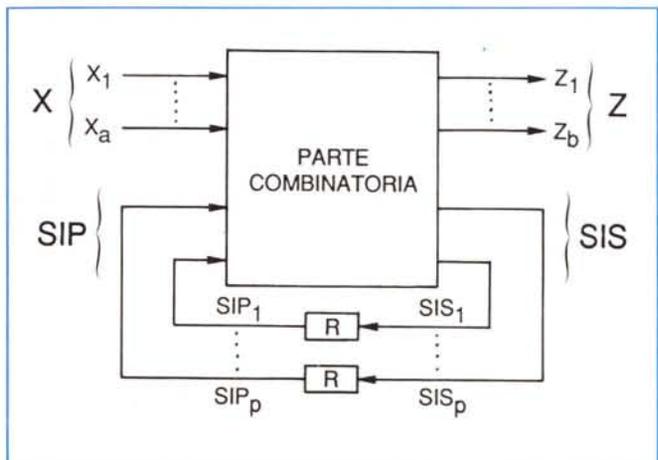


Figura 5 - Una rete sequenziale composta da "a" ingressi binari, "b" uscite binarie e "p" elementi binari che compongono lo stato interno della rete, ognuno dei quali è ritardato mediante gli elementi di ritardo "R".

gresso degli elementi di ritardo R. Questi p valori binari codificano lo stato interno successivo (SIS) della rete, vale a dire quello che diventerà lo stato interno presente (SIP) della rete all'istante t_1 , dopo che il ritardo procurato dagli elementi R si sarà esaurito. In tale istante t_1 , sarà necessario inviare un nuovo input alla rete, diciamo X' , onde evitare che la variazione dei soli morsetti SIP procuri variazioni sull'uscita della rete. In parole semplici, questa soluzione al problema della stabilizzazione della rete, consiste nel sincronizzare perfettamente l'invio di nuovi input con la produzione, che fa la rete, di nuovi SIP. Per usare un gioco di parole, quello che si è fatto è di ottenere una stabilizzazione instabile della rete, che ne garantisce un corretto funzionamento solo nel caso di una perfetta sincronizzazione degli input.

La rete sequenziale mostrata in figura

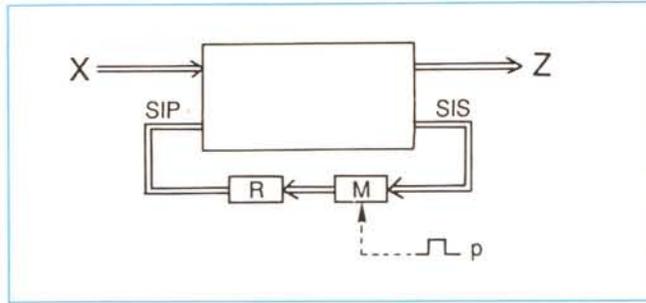


Figura 6 - Architettura reale di una rete sequenziale sincrona.

stato interno successivo, anche un segnale «p» detto impulso. Questo segnale impulsivo è di natura diversa rispetto agli altri segnali finora utilizzati, che sono invece detti segnali a livelli. La differenza tra questi due tipi di segnali, come illustra la figura 7, sta nel fatto che un segnale a livelli può occupare sia il livello 0 che il livello 1 (dove questi valori sono ovviamente solo logici) e transire in un qualsiasi istante da un livello all'altro, mentre un segnale impulsivo occupa di norma il livello 0, transisce da 0 a 1 solo quando un

cuito» provocato dagli anelli di retroazione sui segnali costituenti lo stato interno della rete. Tale cortocircuito (lungi dal preoccupare per il verificarsi di un esasperato «effetto Joule») provoca una successione di transizioni dello stato interno della rete, oltre che delle uscite corrispondenti. Le reti sequenziali sincrone, forniscono una soluzione a tale problema che consiste nello «sbarare» gli anelli di retroazione impedendo così il ciclare dei segnali all'interno della rete. La soluzione fornita dalle reti sequenziali asincrone è più mite, e consiste in modifiche apportate al funzionamento della rete. Vediamo come. Sia a lo stato interno presente della rete. Se ai morsetti d'ingresso si presenta la configurazione X_n , avremo un'uscita $f_z(X_n, a) = Z_k$ ed un nuovo stato interno $f_s(X_n, a) = b$. Poiché il nuovo stato interno b può diventare lo stato interno presente prima che l'ingresso X_n sia variato, la rete si troverà a generare una nuova uscita $f_z(X_n, b)$ ed un nuovo SIS $f_s(X_n, b)$. Per ottenere un corretto funzionamento della rete, ed un'immediata stabilizzazione, basterà assicurare che $f_z(X_n, b) = Z_k$ (in modo che non vi siano uscite non significative) ed $f_s(X_n, b) = b$ (in modo che SIS=SIP).

A partire dalla tabella di definizione di una rete sequenziale, esistono dei procedimenti formali che permettono di trasformare tale tabella in una equivalente in cui, per ogni ingresso e per ogni stato interno, la regola di cui sopra venga garantita. La tabella così ottenuta viene detta «tabella normale» della rete, ed a partire da essa è possibile realizzare una rete sequenziale asincrona. La sintesi di tali reti, tuttavia, richiede un certo numero di precauzioni aggiuntive quali: l'eliminazione delle alee statiche influenti, quella delle alee essenziali e delle corse critiche che costituiscono possibilità di malfunzionamenti della rete. A causa della maggiore difficoltà di implementazione delle reti asincrone rispetto a quelle sincrone, queste ultime sono da preferirsi rispetto alle prime. Esistono tuttavia casi in cui gli input della rete possono variare ad istanti imprevedibili, impedendo un funzionamento sincrono delle reti; per tali tipologie di applicazioni, non esiste alternativa all'uso delle reti sequenziali asincrone.

MC

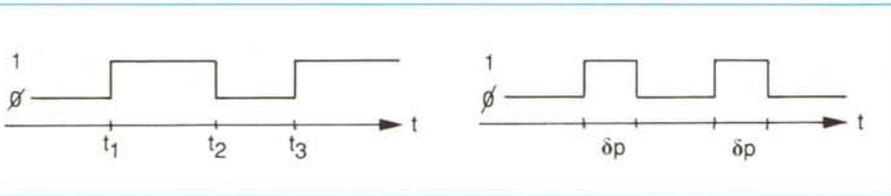


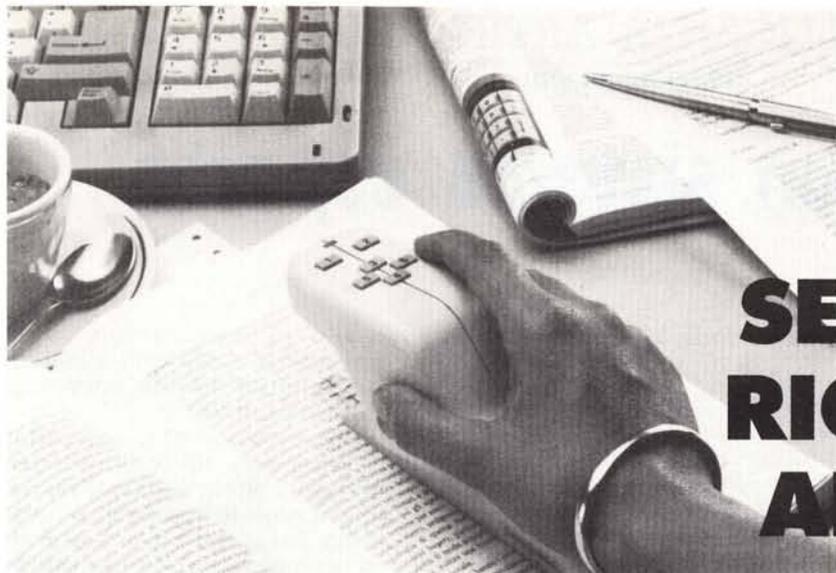
Figura 7 - Segnali a livelli e segnali impulsivi.

5, è un modello generale di rete sequenziale sincrona. Tale modello è, però, ideale, ed esso non potrebbe funzionare nella realtà, a causa della diversità nei tempi di propagazione dei segnali fisici sui vari morsetti della rete. In altri termini, se una delle variabili binarie costituenti lo stato interno successivo della rete, giunge all'ingresso del suo elemento di ritardo, sfalsata rispetto alle altre di un tempo δ , tale variabile andrà ad influenzare la corrispondente variabile SIP, sempre sfalsata del tempo δ , la qual cosa può causare la transizione della rete su stati erronei. Per risolvere questo problema è necessario che i vari elementi di ritardo R, siano sostituiti con degli elementi capaci di memorizzare l'informazione per un tempo qualsiasi. L'architettura della rete sequenziale conseguente a questa modifica, è detta architettura regale, ed è mostrata in figura 6. Il fatto di aver utilizzato delle frecce singole per indicare il flusso delle varie configurazioni sulla rete, invece di mostrare tutti i morsetti, è solo una scelta convenzionale per dare maggiore leggibilità alla figura. Quello che la figura 6 mette invece in evidenza, rispetto all'architettura ideale di figura 5 è l'aggiunta, negli anelli di retroazione, dell'elemento di memoria M cui sono applicati, oltre che i p segnali codificanti lo

impulso viene ad esso applicato e, a partire da quell'istante, resta sul livello 1 per il solo tempo δp , dopodiché ritorna sul livello 0. L'applicazione di un impulso sul segnale p dell'elemento di memoria M, provoca la scrittura, nell'elemento M, del valore presente al suo ingresso. Il valore uscente da M coincide sempre con il contenuto di M stesso. Il corretto funzionamento della rete di figura 6 è dunque affidato ad un clock, che genera ad intervalli precisi di tempo (il famoso «tempo di clock») un impulso sul morsetto p. Lo scopo dell'elemento di ritardo R (che non è stato rimosso rispetto all'architettura ideale) è quello di garantire che lo stato interno presente della rete, non vari durante l'intervallo δp di applicazione dell'impulso, onde evitare che, nell'ipotesi in cui il tempo di trasmissione della parte combinatoria sia trascurabile rispetto ai tempi in questione, un nuovo SIS si presenti all'ingresso di M mentre ancora l'impulso di attivazione della scrittura è applicato ad M.

Reti sequenziali asincrone

Il problema della stabilizzazione di una rete come quella di figura 2 e di figura 4, lo richiamiamo, consiste nella necessità di tenere sotto controllo il «cortocir-



**LEGGE.
SELEZIONA.
RICONOSCE.
APPRENDE.**

Ma non dice "mamma".

Potrebbero venirvi strane idee sulle possibilità di TransImage 1000, ma non esageriamo: può solo fare tutto quello che nessun altro scanner manuale o da tavolo riesce a fare.

Solo TransImage 1000 infatti è in grado di leggere, selezionare e riconoscere come caratteri le parti che più vi interessano di qualsiasi documento: lettere, libri e tabulati. Tutto mentre state comodamente lavorando con il vostro programma preferito: word processor, fogli elettronici e data base, dove potrete inserire automaticamente in tempo reale e senza alcun passaggio intermedio le informazioni che state acquisendo.

In poche parole è la soluzione ideale per tutte quelle applicazioni in cui sia richiesta l'introduzione di dati da tastiera, eliminando la necessità di ridigitare le informazioni stampate, accelerandone l'acquisizione e migliorandone l'affidabilità.

TransImage 1000 legge.

TransImage 1000, ideato per i computer IBM PC XT/AT e compatibili, è dotato di una scheda costruita intorno al microprocessore 68000 Motorola e di una telecamera. Il sistema è in grado di leggere con una velocità di scansione di 120 fotogrammi al secondo testi alfanumerici con una risoluzione di 1000 punti per pollice.

TransImage 1000 seleziona.

TransImage 1000 consente la lettura selettiva di testi e dati. Le informazioni, lette riga per riga, possono essere inviate direttamente a tutti i pacchetti software come WordStar,

WordPerfect, Lotus 1-2-3, dBase III ecc., dove vengono riconosciute come caratteri.

Al fine di semplificare gli interventi da tastiera, la camera è provvista di sei tasti funzione programmabili che possono riprodurre uno o più tasti della tastiera. I tasti funzione possono essere usati per simulare i codici dei tasti di movimento cursore, escape, delete, tab, backspace o return. Si possono anche definire sequenze di comandi da associare ad un unico tasto funzione.

TransImage 1000 riconosce.

L'alta risoluzione della sua telecamera, associata ad una avanzatissima tecnica di Image Processing, rende estremamente accurata la cattura ed il riconoscimento delle immagini. Il suo avanzato metodo di classificazione gli permette il riconoscimento di un grandissimo numero di caratteri diversi.

TransImage 1000 apprende.

Il TransImage 1000 può essere anche "istruito" a riconoscere caratteri speciali, simboli e font di caratteri inusuali. Inoltre il driver software è facilmente personalizzabile per utilizzare lo scanner con gran parte dei programmi in ambiente MS DOS. Per il suo tipo di configurazione il TransImage 1000 risulta "trasparente" ai programmi in MS DOS.

Precisione, selettività, velocità e facilità d'uso sono ottime ragioni per apprezzare TransImage 1000. Che poi non dica (ancora) "mamma", per il vostro PC non è poi di fondamentale importanza...

**SMAU
pad. 17, stand C29**



AISEUROPA

AISEUROPA s.r.l. - 00195 Roma, via Giuseppe Ferrari 2 - tel. (06) 3214120/732161- fax (06) 733107