

Reti logiche sequenziali

■ Dopo la scherzosa pausa del numero scorso, riprendiamo a parlare di informatica, nella fattispecie di reti logiche. Questo mese sarà la volta delle reti sequenziali, diverse da quelle trattate due mesi orsono essendo oggetti di natura non statica. Ogni output non dipende funzionalmente solo dall'input corrente, ma dalla «storia» di input via via elaborati. ■

Riassunto

Una rete logica combinatoria, vista dall'esterno, appare come una scatola dotata di terminali di ingresso e terminali di uscita. Applicando ai terminali di ingresso combinazioni di valori binari, otterremo ai terminali di uscita altri valori binari. Come già ribadito due numeri fa, né più né meno che una funzione binaria: applicando in ingresso cento volte lo stesso input otterremo cento volte lo stesso output: da questo la staticità delle reti logiche combinatorie. La prima fase della realizzazione di una rete, è la costruzione della cosiddetta tabella di verità: nient'altro che una tabella in cui annotiamo per ogni input quale sarà il risultato. Da questa direttamente, o tramite le mappe di Karnaugh, si ricostruisce la funzione logica corrispondente, co-

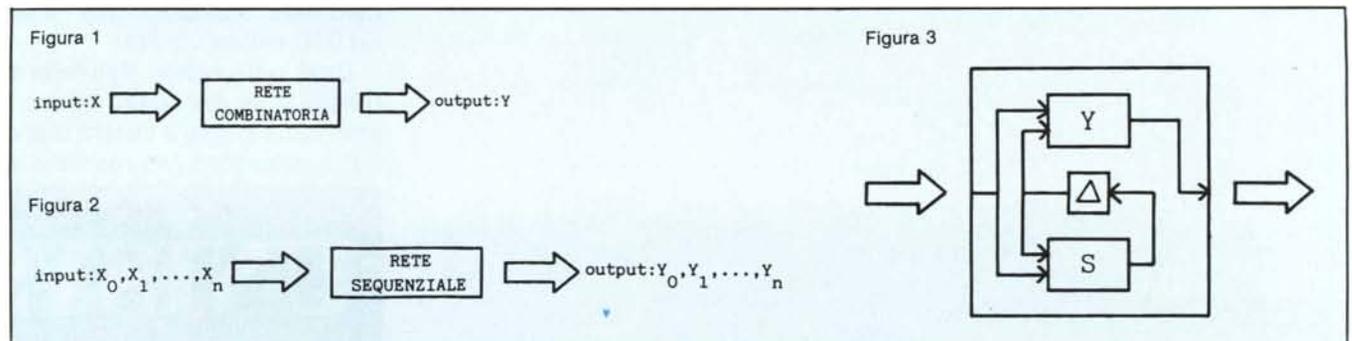
stituita da somme di prodotti o prodotti di somme (logiche) a seconda del metodo usato.

Per la realizzazione fisica della rete logica, una volta ottenuta la funzione corrispondente, si utilizzano componenti elettronici AND, OR, NOT. Ricordiamo che il componente AND esegue il prodotto logico dei suoi due input, l'OR la somma logica, il NOT la complementazione. A questo punto, il passaggio da funzione a rete è più che mai immediato: ove nell'espressione troviamo un prodotto sostituiremo ad esso una porta AND collegando ad essa gli ingressi da «moltiplicare». L'uscita della porta sarà il risultato del prodotto da cui siamo partiti. Discorso analogo per le somme, utilizzando una porta OR. Tutte le uscite delle porte, eventualmente rielaborate tramite altri componenti, costituiranno le

uscite della rete. Questo solo per rinfrescarvi un po' le idee: se avete dubbi a riguardo, vi rimandiamo all'articolo pubblicato in questa stessa rubrica sul numero 63 di MC.

Descrizione generale

A differenza delle reti combinatorie che hanno un comportamento puramente funzionale, le reti sequenziali dispongono di un proprio stato interno. L'output di una rete sequenziale dipende sia dal dato di ingresso che dallo stato interno. Quest'ultimo dallo stato interno precedente e dal dato in ingresso. In figura 2 è mostrata una rete sequenziale, come abbiamo fatto per quelle combinatorie, «vista dall'esterno». Immaginiamo che all'istante iniziale t_0 (come vedete c'è di mezzo il tempo) la rete sia in uno stato iniziale



S0 e riceva in input il dato X0. L'output corrispondente Y0 dipende, come detto, dall'input e dallo stato interno, quindi sarà funzione di X0 e S0. Generato l'output, lo stato interno cambia essendo anch'esso funzione di X0 e S0: trasla in S1, ovvero stato interno al tempo t1.

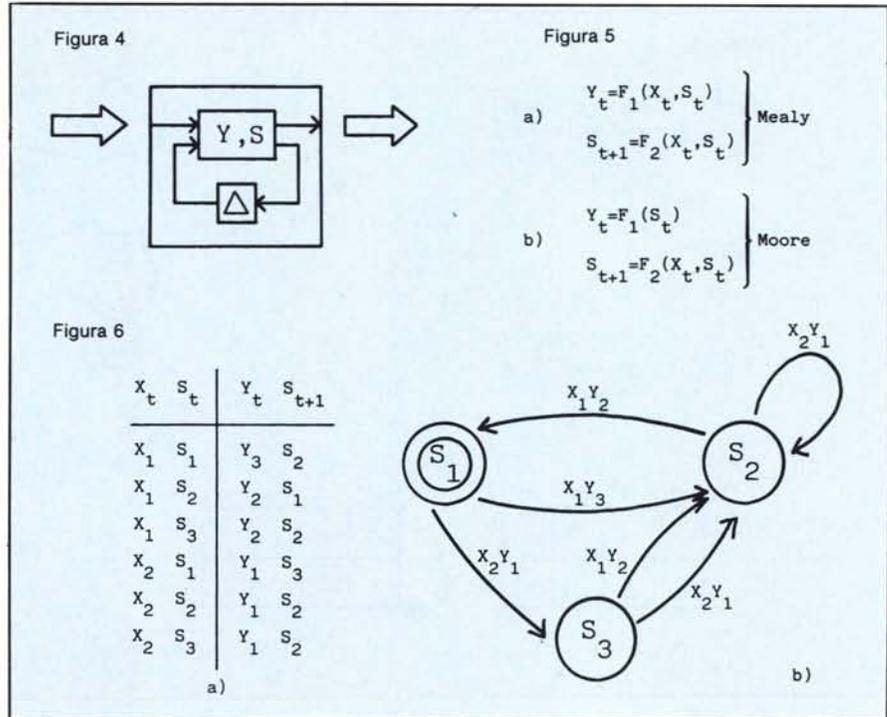
Al colpo successivo, istante t1, l'input X1 e lo stato S1 generano Y1 e S2... all'istante t2, S2 e X2 generano Y2 e S3... e così via, sequenzialmente. Da notare che lo stato interno, variando in funzione della storia degli input, non fa altro che tenere traccia di tutti gli input finora elaborati. Per chi ci ha seguiti nella teoria della computabilità aggiungiamo che una rete sequenziale non è altro che un automa a stati finiti deterministico. Chi non ha letto l'articolo in questione ignori la frase precedente. Riassumendo, una rete sequenziale fornisce sequenze di output in funzione di sequenze di input: per la precisione ogni output è funzione di tutti gli input finora elaborati più naturalmente lo stato iniziale (quello con cui facciamo partire la rete).

Gli input come al solito saranno parole binarie formate da n bit, così come gli stati interni (non dimentichiamo che stiamo parlando dei mattoni di un calcolatore) in generale di lunghezza diversa. L'insieme degli input in numero di 2 alla n costituiranno il cosiddetto alfabeto di ingresso, mentre per gli stati parleremo semplicemente di insieme di stati. Ad esempio la nostra rete potrebbe avere 3 morsetti di ingresso (alfabeto di ingresso formato da 8 elementi) e 4 stati interni codificati con parole di due bit l'uno (come noto con due bit codifichiamo 4 valori).

L'interno

In figura 3 potete ammirare una rete sequenziale vista dall'interno. Si riconoscono tre elementi: due reti combinatorie (colpo di scena!) e un elemento di ritardo contrassegnato con una lettera delta. L'input è trasmesso sia alla rete S (o dello stato interno) che alla rete Y (rete delle uscite). L'elemento di ritardo non fa altro che conservare lo stato interno, per proporlo al colpo successivo alle reti S e Y, per un ciclo: si badi bene che tale modello è solo ideale e proprio per via della sincronizzazione non funzionerebbe mai. Comunque per quello che dobbiamo fare noi (niente) funziona pure troppo bene.

All'istante iniziale nell'elemento delta è conservato lo stato iniziale e quindi questo è presente anche agli ingressi di S e Y. Non appena arriva il primo input le due reti combinatorie (tenete sempre sott'occhio la figura 3) elaborano i loro risultati fornendo Y



l'uscita vera e propria ed S il nuovo stato interno (inviandolo a delta).

Arrivato il secondo input il ciclo si ripete finché ci sono dati da elaborare. Notare che se gli ingressi erano ad n bit e gli stati interni codificati con m bit, le due reti combinatorie interne saranno tutt'e due ad n + m ingressi.

In figura 4 è mostrata una visione meno «esplosa» di una rete sequenziale: le due reti da n e m bit di ingresso possono essere viste come un'unica rete in cui m linee di uscita sono riproposte in ingresso attraverso l'elemento di ritardo delta. È comunque esattamente la stessa cosa.

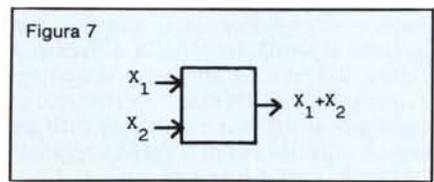
Il modello di figura 3 è detto Modello di Mealy dal nome del suo autore: in figura 5a) sono mostrate le relazioni esistenti tra uscite, ingressi e stati interni di una generica rete sequenziale secondo tale modello. In figura 5b) abbiamo le relazioni relative al modello detto di Moore: in questo caso le uscite sono funzione del solo stato interno, il quale però è in ogni caso funzione sia dello stato interno precedente che del dato in ingresso. Ciò implica essenzialmente che la sequenza delle uscite è traslata temporalmente di un ciclo rispetto alla sequenza di ingresso. Ovvero al primo colpo l'uscita

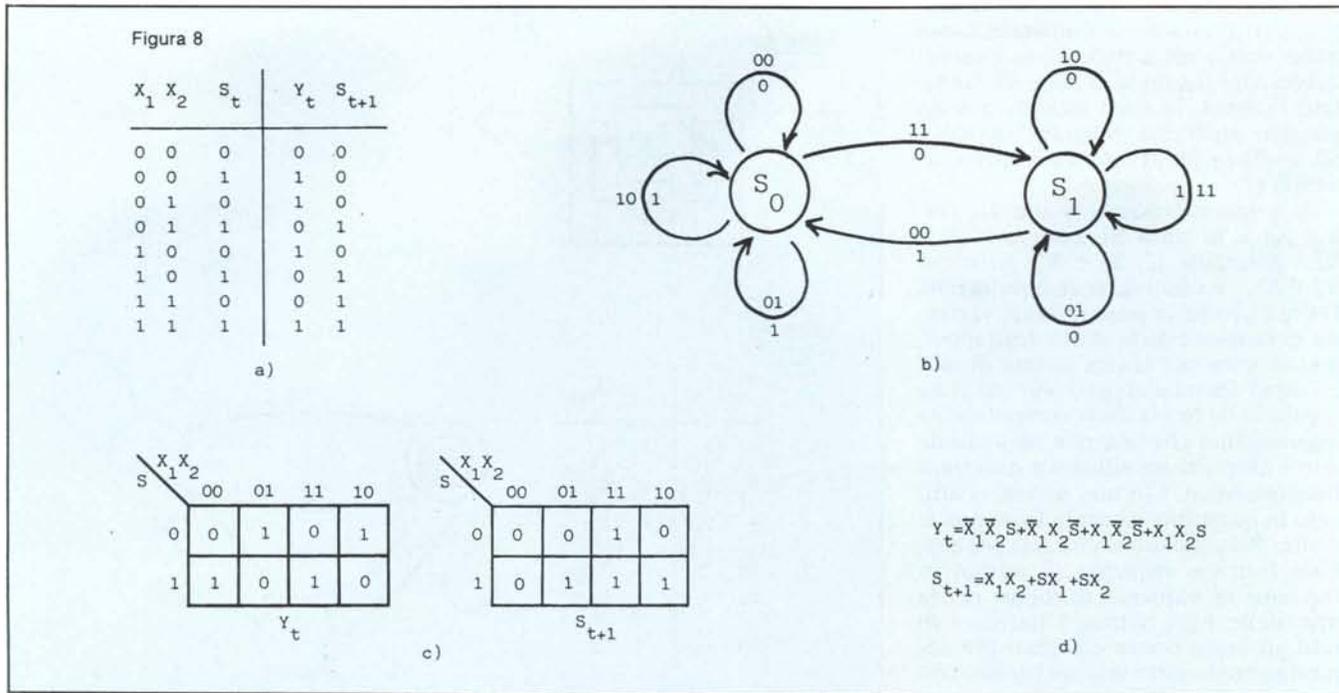
non dipende dal primo ingresso: questo modificherà il solo stato interno che produrrà i suoi effetti sulle uscite al ciclo successivo, quando cioè arriva il secondo ingresso.

Noi continueremo a riferirci al modello di Mealy.

Tabelle di flusso

Se per descrivere le reti combinatorie era sufficiente scrivere la corrispondente tabella di verità, per le reti sequenziali adopereremo un metodo assai simile detto della tabella di flusso. Come era da immaginare, per la descrizione, non potremo non tenere conto della struttura interna della rete ovvero che esiste una parte S, una parte Y e c'è di mezzo il tempo. In figura 6a) abbiamo messo la tabella di flusso di una semplice rete ad un ingresso, tre stati interni e due uscite. L'alfabeto di ingresso sarà dunque formato da due simboli, quello di uscita da 4 di cui solo 3 (per noi) significativi. Per stendere una tabella di flusso dovremo indicare per ogni simbolo di ingresso e stato interno al tempo t quale sarà l'uscita (sempre al tempo t) e lo stato interno al tempo t + 1. Il comportamento della rete, una volta stabilito qual è lo stato iniziale, è così univocamente determinato. Proviamo a fare un «giro» nella tabella di figura 6a). Diciamo che lo stato iniziale è S1 (ma potrebbe essere uno qualsiasi). Se in ingresso inviamo X2 (guardate la quarta riga) l'uscita sarà Y1 e il nuovo stato interno S3. Ora inviamo X1: dalla terza riga troviamo che l'uscita corrispondente è





Y2 e che il nuovo stato interno è S2. Della serie: ...e così via.

Grafi

Un altro modo per descrivere una rete sequenziale, molto più intuitivo, immediato da leggere e da «percorrere», è quello di rappresentare un bel grafo del comportamento. I nodi rappresenteranno gli stati interni, gli archi le transizioni di stato, etichettate dal dato in ingresso e dall'output generato. In figura 6b) abbiamo rappresentato il grafo della stessa rete di figura 6a). Lo stato iniziale è stato cerchiato due volte per poterlo riconoscere. Proviamo a compiere la stessa storia di input dell'esempio precedente. Da S1 (questa volta occhio alla figura 6b) con input X1, seguendo la freccia corrispondente, «approdiamo» in S3 con output Y1: l'arco uscente da S1 etichettato X2Y1 dice proprio questo. Dal pallottolo S3 con input X1 (stiamo effettuando lo stesso percorso di prima) finiamo in S2 rilasciando l'output Y2. Eccetera, eccetera.

Realizzazione

Una volta «chiarito il concetto», la realizzazione di una rete sequenziale è assai banale. Si tratta infatti di realizzare semplicemente le due reti combinatorie S e Y e il gioco è fatto. Dalla tabella di flusso possiamo facilmente ricavare le due tabelle di verità corrispondenti che non sono altro che la prima e la seconda colonna combinate una volta con la terza, una volta con la quarta. Per quanto riguarda la loro

realizzazione, mappe di Karnaugh e implicanti stanno lì ad aspettare di essere usate... e poi ormai siamo diventati bravissimi...

Facciamo un esempio

Come ben sapete 1 + 1 fa 10. In binario, naturalmente! Per l'esattezza fa 0 col riporto di 1. 1 + 1 + 1 fa 11, o meglio, 1 col riporto di 1. Una volta imparate queste due semplici regolette saremo in grado di fare la somma binaria, cifra per cifra, di due qualsiasi numeri binari. Vediamo se siamo in grado di far eseguire la somma ad una rete sequenziale. Ovviamente numeri di lunghezza qualsiasi. Provate a pensarci un po' prima di continuare a leggere.

Semplicemente banale. La rete risultante avrà due ingressi ed una uscita (figura 7). Sui due ingressi invieremo via via una cifra di un numero e una cifra dell'altro, dalle meno significative alle più significative. In uscita avremo le cifre somma corrispondenti, ovviamente tenendo costantemente conto anche del riporto delle cifre precedenti. In figura 8a) troviamo la tabella di flusso della rete da realizzare: vediamo come è fatta. Lo stato interno della rete coincide col valore del riporto delle cifre appena sommate. Lo stato iniziale sarà ovviamente 0 (al momento di iniziare una somma non avremo riporti). La tabella di flusso di figura 8a) non fa altro che descrivere l'operazione di somma con riporto: ad esempio, 0 + 0 con riporto di 0 fa ancora 0 con riporto di 0 (prima riga della tabella)... 1 + 0 con riporto di 1 fa 0

con riporto di 1 (sesta riga in tabella)... ecc. ecc.

In figura 8b) troviamo il grafico corrispondente: notare la simmetria dell'insieme... poi dicono che la matematica è disordine! Ad ogni modo, i due stati S0 e S1 sono proprio «riporto di zero» e «riporto di uno», mentre le etichette sugli archi riportano i dati in ingresso e il valore in uscita. Ad esempio da S0 (nessun riporto nell'ultima addizione) con ingressi 11 (ovvero stiamo eseguendo 1 + 1) rilasciamo un 0 e trasliamo in S1 ovvero c'è un riporto per la prossima somma di cifre.

Da notare che per iniziare una nuova somma, ed essere sicuri che lo stato interno sia S0 (la somma precedente potrebbe aver sporcato tale stato) è sufficiente eseguire una somma 0 + 0 ed ignorare il risultato: in tutti i casi, infatti, non si genererà riporto lasciando dunque lo stato interno in S0.

In figura 8c) abbiamo riportato le due mappe di Karnaugh per l'uscita Y e per lo stato interno successivo: la costruzione di queste discende direttamente dalla tabella di flusso di figura 8a). I più attenti avranno notato che per la Y non ci poteva andare peggio: sia gli uno che gli zeri sono così sparpagliati che non riusciamo ad evidenziare un solo implicante da più di 1 o uno 0: in casi come questi, Karnaugh ci aiuta ben poco. Nel caso della S, di contro, possiamo evidenziare ben tre implicanti «da due» quindi l'espressione equivalente appare abbastanza ridotta: sia questa che quella della Y potete trovarla in figura 8d). Per questo mese è tutto.