

I sistemi di elaborazione

di Anna Pugliese

Mi riesce del tutto spontaneo immaginare la gran parte dei lettori di queste righe mossi dalla curiosità di comprendere quale sia il vero argomento di questo articolo; curiosità che nasce dalla troppa genericità della denominazione «sistema di elaborazione»

Cominciamo subito col dire che il termine, nella sua più larga accezione, può essere applicato ad una vastissima gamma di strumenti che vanno dal palottoliere al supercomputer. Fra i tanti possibili sistemi di elaborazione emerge senza dubbio quello definito da Von Neumann come il "sistema calcolatore": una definizione che corrisponde a quella del generico e diffuso computer. Sebbene da un paio di decenni a questa parte siano nati dei computer con architetture speciali, cui la definizione del sistema calcolatore non può essere applicata, sono veramente pochi coloro che hanno avuto l'occasione di vedere dal vero macchine di questo genere. È dunque al modello proposto da Von Neumann che l'attenzione dovrà essere rivolta ancora per molto tempo. Dopo aver fornito queste doverose precisazioni è giunto il momento di «scoprire le carte in tavola» e dichiarare quello che è l'argomento del quale ci occuperemo. Nei due precedenti numeri della rubrica la nostra attenzione è stata rivolta alle

cosiddette reti logiche dei particolari sistemi intesi come strutture mediante cui vengono eseguite sequenze di operazioni appartenenti ad un certo insieme detto «insieme delle operazioni del sistema». È di tali sistemi che vogliamo occuparci, mediante una serie di esempi che ci porteranno dai sistemi elementari a veri e propri processor. Sistemi di questo genere non comprendono il sistema calcolatore, ne rappresentano invece una sua parte detta «CPU» (Central Processing Unit) cioè unità di elaborazione centrale.

Componenti logici di un sistema

Presenteremo adesso alcune reti logiche che possono essere considerate come componenti standard di un sistema di elaborazione, componenti dal cui assemblaggio è possibile ottenere pressoché qualsiasi unità di elaborazione.

Addizzatore (figura 1)

Scopo di questo strumento è quello di ottenere la somma S di due numeri A e B di n bit ciascuno. Esso è composto da n elementi ognuno dei quali è capace di sommare tre cifre binarie indicate genericamente con A_i , B_i ed r_{i+1} , dove r_{i+1} è il riporto della somma effettuata dall'elemento precedente. La figura 1a mostra questi n elementi ed i collegamenti tra loro esistenti. Ogni elemento di addizzatore è una rete combinatoria con tre ingressi e due uscite, la cui tabella di verità (lo strumento cioè che riporta le uscite della rete corrispondenti ad ogni possibile combinazione degli ingressi) è illustrata in figura 1b. La figura 1c riporta invece UNA schematica rappresentazione dell'intero addizzatore. Al di là delle semplificazioni da cui esce fuori la figura 1c, è evidente che un addizzatore dev'essere visto come un'unica rete combinatoria avente $2n+1$ ingressi ed $n+1$ uscite.

Decodificatore (figura 2)

È una rete logica combinatoria con n ingressi e 2^n uscite. Ha lo scopo di decodificare la combinazione degli n bit d'ingresso, in modo da ottenere un'uscita in cui uno solo dei bit che la

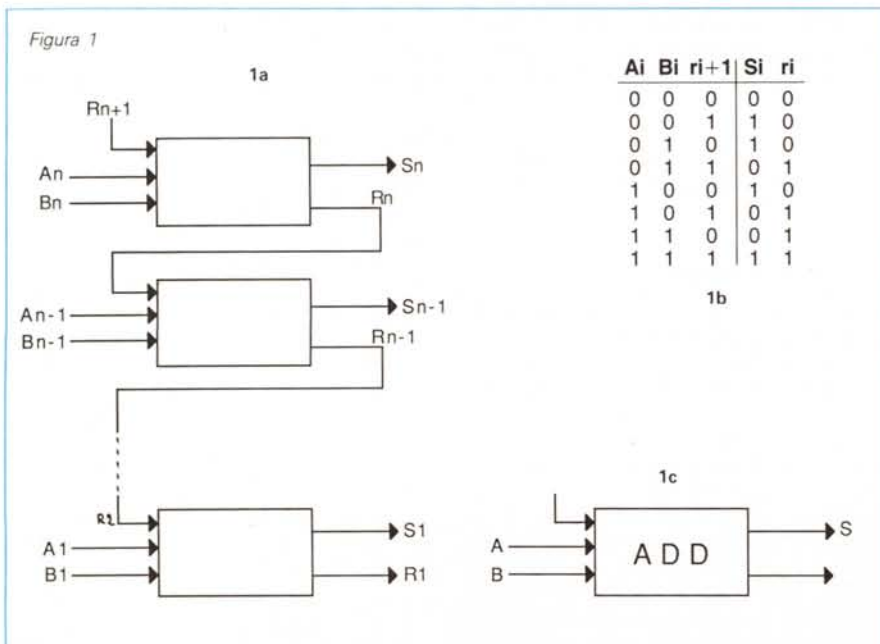
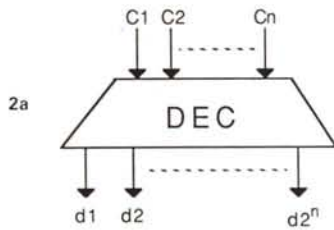


Figura 2



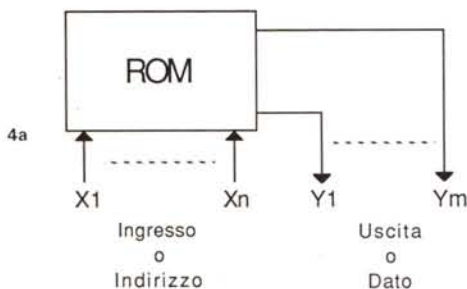
2b

C ₁	C ₂	d ₁	d ₂	d ₃	d ₄
0	0	0	0	0	1
0	1	0	0	1	0
1	1	0	1	0	0
1	0	1	0	0	0

compongono assume il valore 1, mentre gli altri dovranno valere 0. Esistono ovviamente molti modi di associare gli ingressi a corrispondenti uscite in modo da ottenere la funzione del decodificatore; quello mostrato nella tabella di verità della figura 2b è uno dei modi in cui è possibile realizzare un decodificatore di due bit, avente cioè 2 ingressi e $2^2 = 4$ uscite.

Commutatore (figura 3)

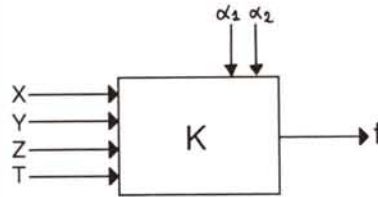
È una rete combinatoria che ha lo scopo di commutare sull'uscita una sola fra le k diverse informazioni presenti al suo ingresso; la commutazione è guidata dal valore assunto da un certo numero di variabili di controllo, anch'esse fornite come input alla rete stessa. La



4b

X ₁	X ₂	Y ₁	Y ₂	Y ₃	Y ₄
0	0	0	1	1	1
0	1	0	1	0	1
1	0	1	1	0	0
1	1	0	1	0	1

Figura 4 - Una memoria di sola lettura; lo schema generale (4a) ed una possibile tabella di verità (4b) con n=2 ed m=4.



3b

alpha	x	y	f
0	0	1	0
0	1	0	1
0	1	1	1
0	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1
1	0	0	0

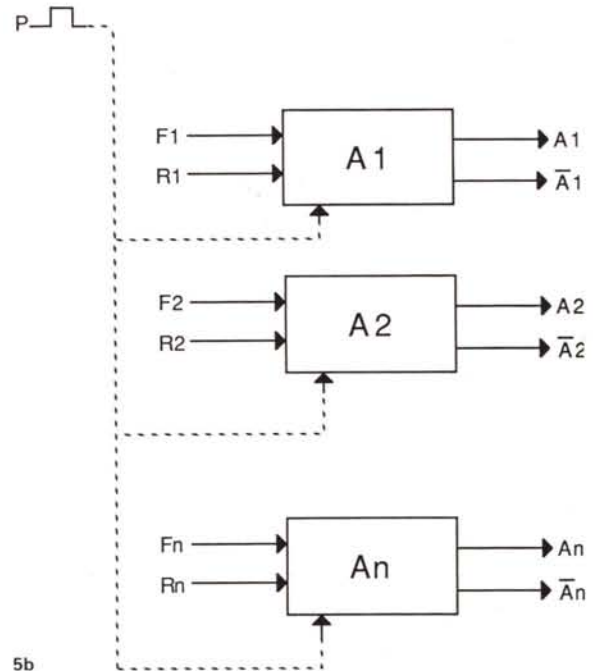
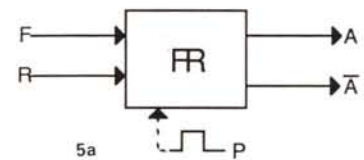
Figura 3 - Lo schema di un commutatore avente in ingresso quattro informazioni più due variabili di condizione (figura 3a) e la tabella di verità di un commutatore con due informazioni ed una variabile di controllo.

cosa può essere chiarita osservando la figura 3a. Supponiamo per il momento che gli ingressi della rete siano tutti costituiti da un solo bit. La funzione della rete sarà allora quella di porre:

f=X	se	$\alpha_1\alpha_2=00$
f=Y	se	$\alpha_1\alpha_2=01$
f=Z	se	$\alpha_1\alpha_2=10$
f=T	se	$\alpha_1\alpha_2=11$

dove ovviamente l'associazione tra il valore di $\alpha_1\alpha_2$ e l'uscita f può essere scelta anche in altri modi. Se le informazioni da commutare sono composte da n bit ciascuna, sarà allora necessario concatenare n elementi di commutatore ognuno dei quali provvede alla commutazione di un singolo bit, e tutti collegati alle stesse variabili di controllo, in modo da fornire un'uscita f composta da n bit, corrispondenti agli n bit dell'informazione commutata. Lo schema di figura 3a può essere considerato come lo schema semplificato di un generico com-

Figura 5



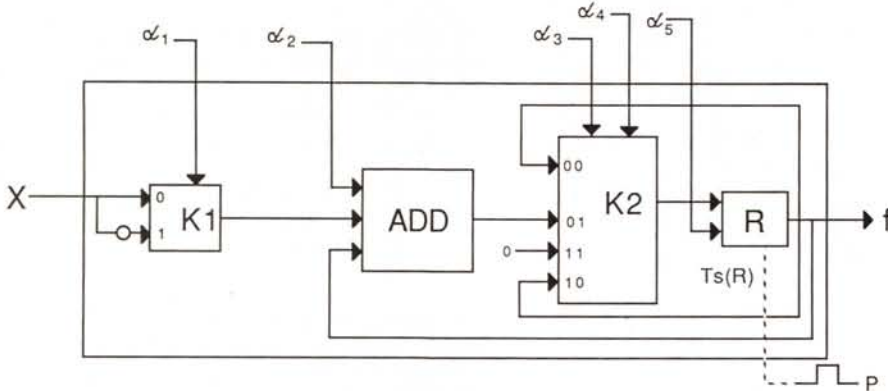


Figura 6 - La parte operativa (PO) del sistema accumulatore completo.

non lo mostra, sono generalmente collegati ad un'unica variabile binaria detta variabile di controllo, il cui scopo è quello di inibire la possibilità di modifica del registro, se vale 0, o viceversa abilitarla se vale 1.

Un esempio di sistema: l'accumulatore completo

Definito il funzionamento dei vari pezzi è giunto il momento di metterli assieme. Il semplice sistema che ci proponiamo di progettare è un sistema capace di eseguire 5 operazioni molto semplici, ma fondamentali per l'unità di calcolo di un computer. Il sistema è dotato di un registro R, di n bit, e di un ingresso X, sempre di n bit, sul quale è possibile ricevere un dato da sommare o sottrarre al contenuto di R. Le altre tre

mutatore avente come effettivi morsetti d'ingresso, nel caso di k informazioni di n bit ciascuna, un totale di $(k \cdot n) + \log_2 k$ bit, dove $\log_2 k$ (che andrà eventualmente arrotondato per eccesso) rappresenta il numero di variabili di controllo necessarie. La figura 3b infine, mostra una possibile tabella di verità per un elemento di commutatore con $k=2$.

R.O.M. (figura 4)

Una memoria ROM (Read Only Memory) è una rete combinatoria avente n ingressi ed m uscite. Essa può essere considerata come un insieme di 2^n celle di memoria contenenti ognuna m bit di informazione non modificabile. La ROM fornirà in uscita il contenuto della cella il cui indirizzo è codificato dalla configurazione di n bit presente all'ingresso.

Registro (figura 5)

Osserviamo dapprima la figura 5a. Essa illustra un cosiddetto «flip-flop» detto flip-flop FR. Le sue due uscite (come quelle di ogni flip-flop) corrispondono al valore memorizzato nel flip-flop ed al suo complemento. Il valore memorizzato nel flip-flop (e di conseguenza anche le sue uscite) viene sostituito dal valore presente sul morsetto binario d'ingresso F, se e solo se un segnale impulsivo è applicato al morsetto p, e contemporaneamente si ha che il valore di R è uguale ad 1.

La figura 5b mostra invece un vero e proprio registro. Esso è una rete sequenziale avente $2n$ ingressi normali ed un ingresso impulsivo p. Come si vede dalla figura, l'impulso p viene distribuito a tutti gli n flip-flop FR da cui il registro è composto.

Gli n bit R_1, \dots, R_n anche se la figura

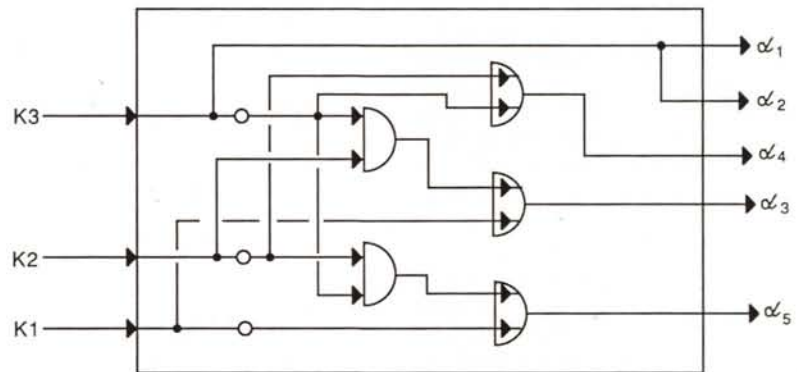


Figura 7 - La parte controllo (PC) da associare alla PO della figura 6 per completare un sistema accumulatore completo.

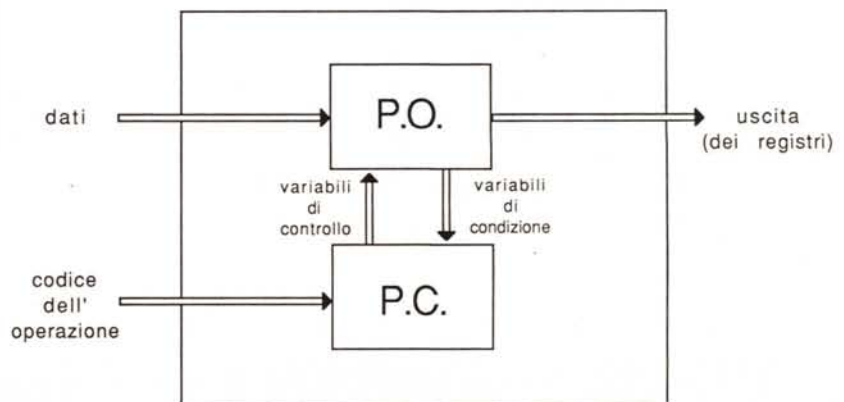


Figura 8 - Lo schema generale di un sistema di elaborazione.

operazioni sono: l'azzeramento di R, la moltiplicazione di R per 2 (o, più precisamente la sua traslazione logica a sinistra), e la divisione di R per 2 (o, più precisamente, la sua traslazione logica a destra).

La seguente tabella mostra le 5 operazioni del sistema con dei codici operativi che ad esse sono stati associati.

K ₁	K ₂	K ₃		operazione:
0	0	0	OP1:	R+X → R
0	0	1	OP2:	R-X → R cioè R+ \bar{X} +1 → R
0	1	0	OP3:	0 → R
0	1	1	OP4:	R/2 → R cioè Td(R) → R
1	0	0	OP5:	R*2 → R cioè Ts(R) → R

Il codice operativo di ogni operazione è formato da una combinazione di 3 bit, essendo 2 bit insufficienti a produrre 5 diverse configurazioni. Le tre configurazioni che ci restano libere, vale a dire K₁K₂K₃ = 101, 110 e 111 saranno associate ad una cosiddetta «operazione nulla» OP0 che sarà implementata ponendo a zero l'ingresso di controllo del registro R e quindi inibendo la modifica del registro stesso.

Vediamo ora di riflettere sulle varie parti del sistema, osservando la figura 6 dove esso è mostrato nella sua totalità. Osserviamo la figura 6 partendo dalla parte destra e cioè dal registro R. All'ingresso di R è collegata l'uscita del commutatore K₂ e la variabile di controllo α₅. Se α₅ = 0 il contenuto di R, anche in presenza dell'impulso, non viene modificato, permettendo in tal modo la realizzazione dell'operazione nulla OP0. ponendo α₅ = 1, R assumerà invece il valore proveniente dall'uscita di K₂ il quale commuta, secondo il valore di α₃α₄ uno dei suoi ingressi, e precisamente:

- Td(R) se α₃α₄ = 00
- l'uscita di ADD se α₃α₄=01
- 0 se α₃α₄ = 11
- Ts(R) se α₃α₄=10

Le traslazioni logiche destre e sinistre del registro R, non sono riportate esplicitamente per non complicare la leggibilità dello schema. È facile tuttavia immaginare come esse possano essere ottenute a partire dal valore presente all'uscita del registro R, se si pensa che R stesso è in realtà formato da n flip-flop: basterà collegare l'uscita di ogni flip-flop al flip-flop precedente (bit meno significativo) o successivo (bit più significativo) per ottenere rispettivamente la traslazione destra o sinistra, e porre a zero il primo o l'ultimo, rispettivamente, dei flip-flop.

Continuando l'esame della figura 6, è facile comprendere che l'uscita di ADD consiste del risultato di X+R se α₁α₂=00

o del risultato di R-X se α₁α₂=11.

Il fatto che R-X coincida con il valore di R+ \bar{X} +1, suppone ovviamente che i numeri in questione siano rappresentati in complemento a due, dove per ottenere l'opposto di un numero è sufficiente farne il complemento (tutti i bit 1 a 0 e tutti i bit 0 ad 1) ed aggiungere 1 al risultato.

Se la spiegazione, come mi auguro, è stata sufficiente, non ci resta che capire chi e come fornirà i valori necessari alle 5 variabili di controllo. Tali valori, se opportunamente forniti, permettono al sistema di eseguire una sola fra le 5 operazioni eseguibili, ad ogni ciclo di clock, dove il clock è lo strumento cui è affidato l'invio degli impulsi p alla ricezione di ognuno dei quali il contenuto del registro R diviene passibile di cambiamento. Il calcolo dei valori da assegnare alle variabili di controllo dovrà ovviamente dipendere dal codice operativo che verrà fornito al sistema. Tale codice non è considerato nella figura 6 in quanto essa illustra una parte soltanto del sistema, precisamente la così detta «parte operativa» (PO). L'altra parte del sistema, detta «parte controllo» (PC), si occuperà di produrre gli opportuni valori per le variabili di controllo, sulla base dei valori che essa avrà in ingresso specificanti il codice operativo dell'operazione prescelta. Nel nostro caso, la PC del sistema sarà una rete combinatoria la cui tabella di verità è riportata di seguito.

	K ₁	K ₂	K ₃	α ₁	α ₂	α ₃	α ₄	α ₅
OP1	0	0	0	0	0	0	1	1
OP2	0	0	1	1	1	0	1	1
OP4	0	1	1	=	=	0	0	1
OP3	0	1	0	=	=	1	1	1
OP5	1	0	0	=	=	1	0	1
OP0	1	0	1	=	=	=	=	0
OP0	1	1	0	=	=	=	=	0
OP0	1	1	1	=	=	=	=	0

La tabella di verità è non completamente specificata. Per comprendere questo fatto, si pensi ad esempio al caso in cui α₅ = 0; osservando la figura 6 si capisce che qualsiasi sia il valore delle altre variabili di controllo, il contenuto di R non cambierà, per cui il valore scelto per tali variabili sarà scelto in modo da semplificare la complessità della PC del sistema. In particolare otterremo le seguenti relazioni:

$$\begin{aligned} \alpha_1 &= \alpha_2 = K_3 \\ \alpha_3 &= K_1 + K_2 \bar{K}_3 \\ \alpha_4 &= \bar{K}_2 + \bar{K}_3 \\ \alpha_5 &= \bar{K}_1 + \bar{K}_2 \bar{K}_3 \end{aligned}$$

a partire dalle quali è stata ottenuta la rete logica combinatoria mostrata in figura 7, costituente la parte controllo del nostro sistema. La rete della figura 7 potrebbe essere realizzata anche mediante una ROM avente 8 celle di memoria contenenti ognuna una configurazione di 5 bit; in tal caso il codice operativo costituente l'ingresso della nostra PC corrisponderebbe all'indirizzo di una cella della ROM contenente la sequenza dei 5 bit da assegnare alle 5 variabili di controllo della PO.

La struttura generale dei sistemi

Il sistema di elaborazione che abbiamo visto nell'esempio, è costituito da due reti logiche: la rete sequenziale di figura 6 e la rete combinatoria di figura 7. In realtà, la suddivisione del sistema in una PO ed una PC è una caratteristica di ogni sistema di elaborazione, dettata da varie esigenze. In generale però, entrambe le reti logiche costituenti il sistema sono reti sequenziali. Il fatto di aver fatto eccezione nel caso del nostro esempio, a causa dell'estrema semplicità del nostro sistema, conferma comunque in pieno la regola essendo possibile vedere una rete combinatoria come un caso particolare di una rete sequenziale avente un solo strato interno.

Se si pensa alla possibilità di dover realizzare operazioni condizionali, operazioni cioè che dipendono dai risultati lasciati da operazioni precedenti, ed alla crescente difficoltà di controllare le varie operazioni al crescere del loro numero, si comprende la necessità di dover

ricorrere a speciali tecniche di progettazione della Parte Controllo dei sistemi. Non ci resta sufficiente spazio, in questa sede, per tentare un seppur breve accenno a tali tecniche. Esse comunque permettono di progettare qualsiasi sistema di elaborazione, ed in particolare la CPU dei computer, realizzando il sistema mediante un ciclo di due reti sequenziali, secondo lo schema della figura 8.