

# And, Or, Not

■ *Concluso il breve ciclo sulla teoria della calcolabilità, a partire da questo numero Appunti di Informatica tratterà il tema delle reti logiche: i veri «mattoni» che costituiscono l'hardware di un computer.*

*Trattando di calcolabilità abbiamo mostrato come partendo da semplici strumenti quali i numeri naturali e le operazioni primitive su essi definite era possibile calcolare qualsiasi funzione calcolabile. A partire da questo mese, con le reti logiche, andando a rispolverare i fatidici zeri e uno binari che corrono all'interno di ogni calcolatore digitale, scopriremo insieme fin dove è possibile arrivare.*

*Buon viaggio... ■*

## Algebra della commutazione

Tra le tante cose che «sanno pure le pietre» annoveriamo certamente il fatto che i computer (tradizionali) intendono solo in termini di zeri e di uno, le cosiddette cifre binarie, e non in termini di programmi scritti in un linguaggio ad alto livello o in una qualsiasi forma di linguaggio assembly. Alla fin fine, ciò che la CPU sarà in grado di *masticare* saranno sempre e solo livelli logici 0 e 1, vero-falso, +5 volt - 0 volt, che dire si voglia.

Nel corso di questi articoli, i quali come al solito non hanno neppure il minimo intento di sostituirsi ai testi specifici in materia, assumeremo che all'interno di un computer circolino le cifre binarie 0 e 1. Il fatto che questi non siano veri e propri numeri ma segnali elettrici non lo terremo in considerazione, tanto per cambiare, per non appesantire troppo la faccenda. Ancora una volta, e non poteva essere diversamente, ci comporteremo quanto più informaticamente possibile: ciò che manipoleremo è *pura* informazione. Punto e basta.

Detto questo, prima di entrare nel merito, occorre fare una piccola introduzione riguardo l'algebra della commutazione (tranquilli, è di una banali-

tà unica...) che è alla base della descrizione delle reti logiche che tratteremo. Gli 0 e gli 1 già li abbiamo: diamo per vero il fatto che in qualche modo esistono e dobbiamo solo manipolarli. Per fare questo, si usano essenzialmente tre operazioni: l'And, l'Or e il Not che danno il nome a quest'articolo. Le prime prendono due cifre binarie e ne restituiscono una, la terza prende una cifra binaria e ne restituisce un'altra. Per cifra binaria, qualora non fosse chiaro, si intende uno 0 o un 1. Null'altro.

La più semplice, il Not non fa altro che complementare la cifra binaria: se è 0 da 1, se è 1 da 0, ovvero:

$$\begin{aligned} \text{NOT}(0) &= 1 \\ \text{NOT}(1) &= 0 \end{aligned}$$

L'And di due cifre binarie è uguale a 1 se *entrambe* le cifre sono pari a 1, 0 altrimenti. Per iscritto:

$$\begin{aligned} 0 \text{ AND } 0 &= 0 \\ 0 \text{ AND } 1 &= 0 \\ 1 \text{ AND } 0 &= 0 \\ 1 \text{ AND } 1 &= 1 \end{aligned}$$

Di contro, l'Or di due cifre binarie è uguale a 1 se *almeno* una di queste due è pari a 1:

$$\begin{aligned} 0 \text{ OR } 0 &= 0 \\ 0 \text{ OR } 1 &= 1 \end{aligned}$$

$$\begin{aligned} 1 \text{ OR } 0 &= 1 \\ 1 \text{ OR } 1 &= 1 \end{aligned}$$

Fanno parte dell'algebra della commutazione alcuni importanti teoremi facilmente dimostrabili col metodo della *perfetta induzione*: essendo il dominio e il codominio così ristretto, è sufficiente verificare le relazioni sostituendo una per una (e sono sempre poche) tutte le combinazioni di valori alle variabili indicate. Tali relazioni sono mostrate in figura 1: lì, per compattare la scrittura, l'AND è sostituito da un puntino, l'OR dal simbolo + e il NOT col simbolo di complementazione. A, B, o C sono variabili binarie ovvero possono valere 0 o 1.

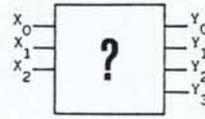
## Reti logiche combinatorie

Nell'introduzione di questo articolo dicevamo che le reti logiche rappresentano i veri e propri mattoni coi quali è formato l'hardware di ogni calcolatore digitale.

Distinguiamo tra reti combinatorie e reti sequenziali: di quest'ultime ci occuperemo in seguito. Una rete logica combinatoria, vista dall'*esterno* appare come una scatola chiusa dotata di morsetti di ingresso e morsetti di uscita. Se volete, pensate pure ad un

- 1)  $A + \bar{A} = 1$
- 2)  $\bar{\bar{A}} = A$
- 3)  $A + A = A$
- 4)  $A + 0 = A$   
 $A + 1 = 1$
- 5)  $A + B = B + A$
- 6)  $A + (B + C) = (A + B) + C$
- 7)  $A \cdot (B + C) = A \cdot B + A \cdot C$
- 8)  $\bar{A} \cdot \bar{B} = \overline{A + B}$
- $A \cdot \bar{A} = 0$
- $A \cdot A = A$
- $A \cdot 1 = A$   
 $A \cdot 0 = 0$
- $A \cdot B = B \cdot A$
- $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- $A \cdot (B \cdot C) = (A \cdot B) \cdot (A + C)$
- $\bar{A} \cdot \bar{B} = \overline{A + B}$

Figura 1



(a)

$X_0$	$X_1$	$X_2$	$Y_0$	$Y_1$	$Y_2$	$Y_3$
0	0	0	1	1	0	0
0	0	1	0	1	1	0
0	1	0	1	1	1	1
0	1	1	0	1	1	0
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	1	1	0	1

(b)

Figura 2

integrato con tanto di piedini. Applicando in ingresso una qualsiasi combinazione di valori binari (uno per morsetto di ingresso) otterremo ai morsetti di uscita un determinato valore binario dipendente da questo. Né più né meno che una funzione.

Da sottolineare la staticità di una rete combinatoria: a differenza delle reti sequenziali, applicando più volte lo stesso dato in ingresso otterremo sempre lo stesso risultato in uscita. Ovvero presa la nostra misteriosa scatola figura 2A, la prima cosa che possiamo fare è costruire la corrispondente tabella di verità (figura 2B): applichiamo agli ingressi tutte le combinazioni di 0 e di 1 e annotiamo cosa accade in uscita, ai morsetti  $Y_0...Y_3$ .

Le reti logiche, come intuibile, possono avere un numero qualsiasi di ingressi e un numero qualsiasi di uscite, il nostro scopo sarà quello di realizzare la rete partendo solo dal comportamento esterno: un po' come risolvere un gioco, dimmi come ti comporti... ti dirò come sei.

**All'interno**

Cominciamo la nostra avventura all'interno delle reti logiche svelandovi il primo segreto: le funzioni logiche elementari. Non sono una novità in quanto ne abbiamo appena parlato nell'algebra della commutazione: ci riferiamo all'AND, OR e NOT, questa

volta sottoforma di componenti più che funzioni binarie. In figura 3 sono mostrate rispettivamente una porta AND, una porta OR e una porta NOT tutte con relativa tabellina di verità. Come nel caso della scatola di cui sopra, applicando ad esempio ad una porta AND uno 0 e un 1, troveremo in uscita uno 0; applicando due 1 otterremo un 1 in uscita. Discorso analogo per le altre porte, riferendovi sempre (finché non imparate!) alle relative tabelline di verità. È inoltre possibile estendere le porte AND e OR di cui sopra a più di due ingressi, lasciando ovviamente inalterato il loro significato: una porta AND ad n ingressi presenta in uscita il valore 1 se *tutti* i suoi ingressi sono ad 1; una porta OR ad n ingressi restituisce 1 se *almeno* uno dei suoi ingressi è ad 1. Altrimenti 0.

Le reti logiche combinatorie sono formate da elementi di questo tipo. Progettare una rete logica si traduce nel combinare opportunamente i componenti di figura 3 (possibilmente con minor spreco possibile) fino ad ottenere una rete rispondente alle caratteristiche richieste. Le caratteristiche richieste sono semplicemente di avere una relativa tabella di verità uguale a quella di partenza tutto qui.

**Un primo esempio**

Secondo segreto: una rete logica ad n ingressi ed m uscite non è altro che

un insieme di m reti ad n ingressi ed una sola uscita che forniscono ognuna uno dei valori  $Y_1...Y_m$ . Da questo, la risoluzione di una rete siffatta si riduce a risolvere un certo insieme di reti ad una sola uscita. Ovvero come primo esempio vedremo la risoluzione di una rete a tre ingressi ed una uscita, la cui tabella di verità è mostrata in figura 4A.

La prima operazione che compiremo sarà di ricavare l'espressione logica equivalente alla tabella: una espressione fatta di Not, And e Or che valutata (sostituendo i valori per  $X_0, X_1$  e  $X_2$ ) ha un comportamento analogo a quello della rete da risolvere.

È molto facile: dalla tabella si evidenziano tutte le righe la cui Y vale 1 e si costruisce la *forma canonica* dell'espressione come somma di termini prodotto, ognuno di questi ottenuto complementando o meno le variabili se in quella riga valevano 0 o 1. Come prima, somma sta per Or e prodotto per And. Procediamo: prendiamo la prima riga in cui la Y vale 1, la terza. Le variabili valgono 0 1 1, quindi il primo termine è:

$$\bar{X}_2 \cdot X_1 \cdot X_0$$

$X_2$  è complementato dato che il suo valore era 0,  $X_1$  e  $X_0$  no in quanto valevano 1.

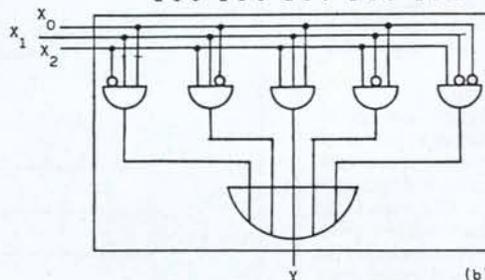
La quinta riga (la seconda con Y uguale a 1) vale 1 1 0. Il secondo ter-

METODO TABELLA DI VERITA'

$X_2$	$X_1$	$X_0$	Y
0	0	0	0
0	0	1	0
0	1	1	1
0	1	0	0
1	1	0	1
1	1	1	1
1	0	1	1
1	0	0	1

Figura 4

$$Y = \bar{X}_2 X_1 X_0 + X_2 X_1 \bar{X}_0 + X_2 X_1 X_0 + X_2 \bar{X}_1 X_0 + X_2 \bar{X}_1 \bar{X}_0$$



COMPONENTE AND

$X_0$	$X_1$	Y
0	0	0
0	1	0
1	1	1
1	0	0

(a)

COMPONENTE OR

$X_0$	$X_1$	Y
0	0	0
0	1	1
1	1	1
1	0	1

(b)

COMPONENTE NOT

X	Y
0	1
1	0

(c)

Figura 3

COMPONENTE NAND

$X_0$	$X_1$	Y
0	0	1
0	1	1
1	1	0
1	0	1

Figura 5

COMPONENTE NOR

$X_0$	$X_1$	Y
0	0	1
0	1	0
1	1	0
1	0	0

mine, di conseguenza, sarà:

$$X_2 \cdot X_1 \cdot \bar{X}_0$$

e così via sino all'ultimo termine.

L'espressione completa è mostrata sempre in figura 4. Il passaggio da espressione a rete è quantomai immediato, trattandosi semplicemente di sostituire ogni termine con l'opportuno elemento AND (con elementi NOT all'ingresso, i pallini, dove necessario) e mettere in OR tutte le uscite di questi, come mostrato in figura 4B. Signore e signori ecco a voi la rete corrispondente.

Si poteva anche procedere in maniera complementare in prodotti di somme invece che di somme di prodotti, nel qual caso avremmo evidenziato le righe in cui la Y valeva 0. In tal caso la rete sarebbe stata formata da tre elementi OR le cui uscite finivano in una porta AND: a dire il vero avremmo anche risparmiato elementi, ma non importa, il primo è solo un esempio... e per risparmiare si utilizzano ben altri metodi. Prima di svelarvi altri segreti, una simpatica curiosità.

**NAND & NOR**

Esistono altri due importanti elementi logici, le porte NAND e NOR, mostrati in figura 5, sottoforma di componente e di relativa tabellina di verità. Non sono altro che delle normali porte AND e OR suffissate da un NOT: dove le prime valevano 0 queste valgono 1 e viceversa. Perché tanta importanza a questi due nuovi signori?

Semplice, usando solo elementi NAND o solo elementi NOR è possibile costruire qualsiasi rete combinato-

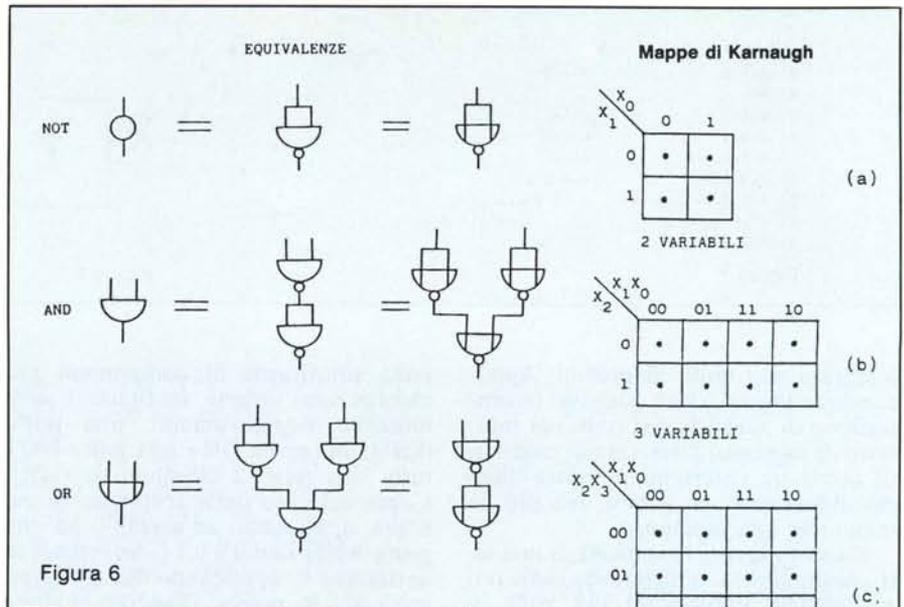


Figura 6

ria. Per dimostrare questo fatto, assunto che OR, AND e NOT di prima sono sufficienti (credeteci), è sufficiente mostrare come è possibile costruire ognuna di queste tre porte con soli elementi NAND e NOR. Il tutto è mostrato in figura 6. Ad esempio per ottenere un NOT è sufficiente cortocircuitare gli ingressi di una di queste nuove porte per ottenere il voluto. Infatti dal teorema 3 di figura 1 abbiamo che  $A \cdot A = A$  e che  $A + A = A$ , se dopo di questo troviamo un «pallino» si ha che:

A NAND A = NOT(A)  
 A NOR A = NOT(A)

Soprattutto, a differenza di altri/e dipendenti, non urla, non piange, non è isterico, non è nevrastenico, esaurito o schizofrenico o... peggio.

Pallometro in mano, comincio a disegnare le prime porte and/or. Matteo mi guarda e dice:

«Toh!, un bel servizio di bicchieri...»

Lieve sorriso comprensivo da parte del sottoscritto, mentre continuo a disegnare...

Pochi attimi ancora poi Matteo replica: «Ah!, no, sono delle porte logiche... toh! anche le mappe di Karnaugh...»

La matita mi scivola tra le dita. Rino ed io ci guardiamo nelle palle degli occhi per poi esclamare in coro:

«Azzo!... Che bella battuta!»

Poi scoprimmo che Matteo (da oggi ribattezzato professor Matteo) prima di prendere il posto di lavoro in questione, studiava fisica all'università e dopo aver lasciato gli studi ufficiali, alla sera legge libri di informatica di vario taglio e peso. Propongo una targa di riconoscimento... più una bella promozione a docente ordinario. Tanti auguri.

Figura 7

**Esempi di implicanti**

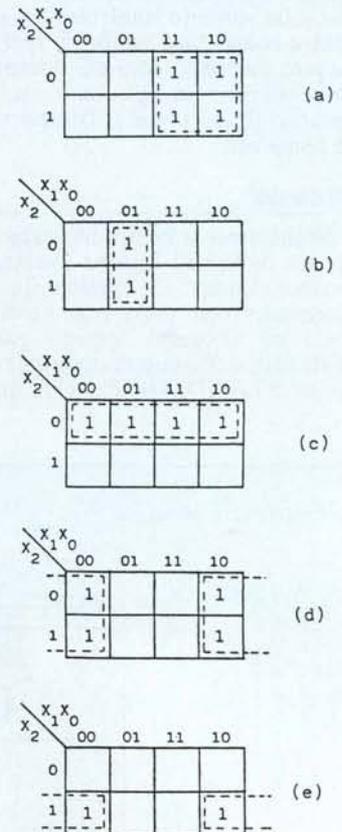


Figura 8

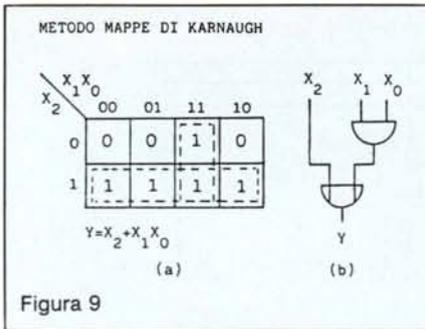
**Matteo SuperStar**

Che bella battuta!

Fatto sta che il sottoscritto (erano da poco passate le 17 di un grigio pomeriggio romano, in compagnia del piccolo Rino), armato dei potenti mezzi tecnologicamente avanzati della mia valigetta tuttofare, si accingeva a sedere al tavolo dei grafici (i noti fratelli Saltarelli) per eseguire i disegni che trovate ora in queste pagine.

Il buon Matteo, approfittando del momentaneo attimo di quiete (ebbene sì, strano a dirlo, ma in quel momento in redazione c'era proprio silenzio) ronzava anch'egli tra le stanze e i corridoi redazionali di MCmicrocomputer.

Dovete sapere che il buon Matteo, assunto dalla Technimedia in qualità di magazziniere addetto anche (a quanto pare) alla spedizione di programmi su disco e nastro, è sicuramente il più amato da tutti proprio per le sue alte doti di signorilità, savoir faire, feeling, public relation (negate al sottoscritto da molti) e tant'altro.

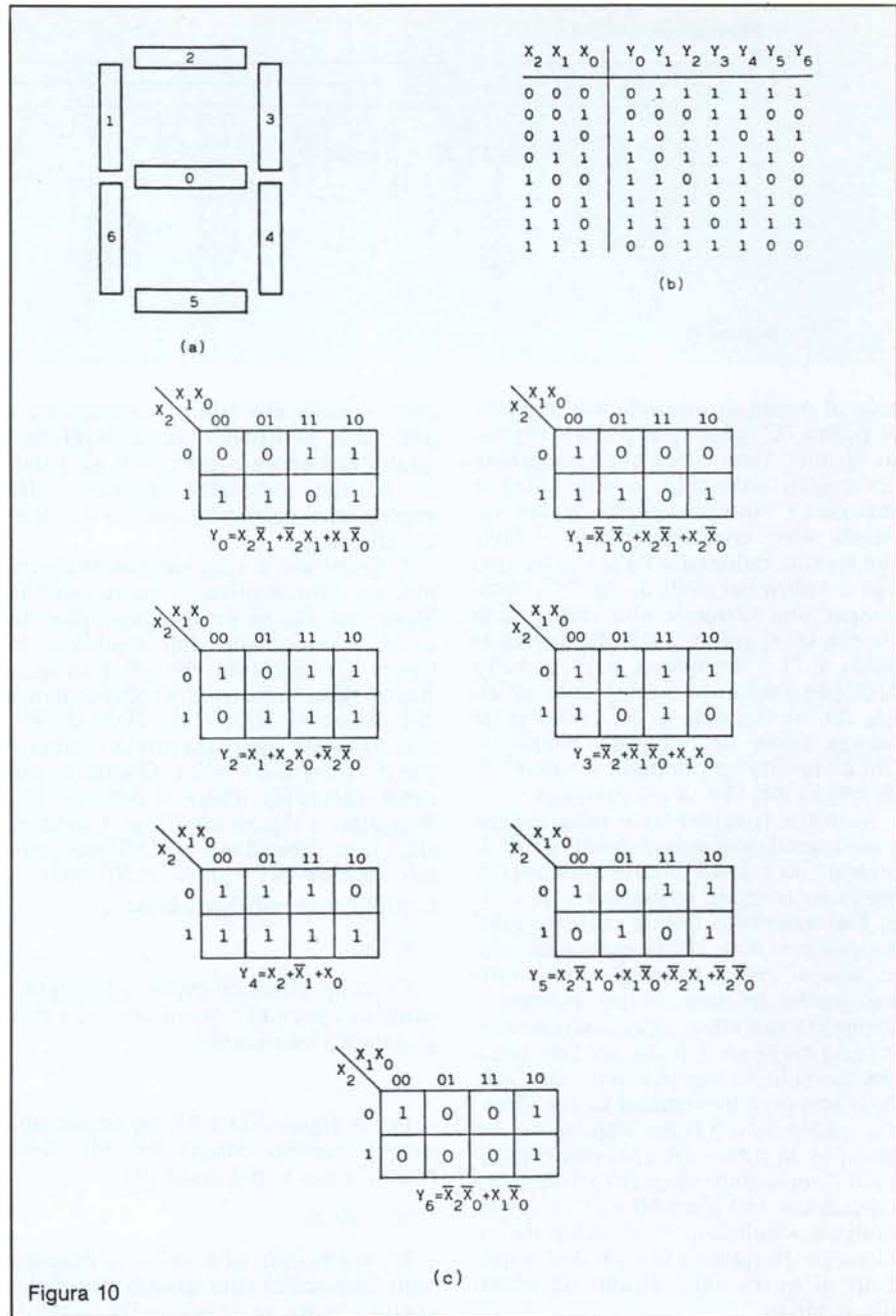


Per ottenere porte AND e OR useremo il teorema 8 (le cosiddette formule di De Morgan, per chi non le avesse riconosciute) se dobbiamo costruire una porta AND da porte NOR o porte OR da porte NAND, o semplicemente ne useremo due in cascata, dove la seconda funziona da NOT (caso in cui dobbiamo passare da NOR a OR o da NAND ad AND). Bell'è dimostrato: con le sole porte NOR o con le sole porte NAND possiamo costruire AND, OR e NOT e, conseguentemente, qualsiasi rete combinatoria. Tutto questo è «cosa buona» quando si è interessati più alla *standardizzazione circuitale* (uso di un solo tipo di porta) che ad un vero e proprio risparmio di elementi. Non è nuovo il fatto che costi meno la realizzazione e costruzione di un integrato fatto da molti pezzi «facili» piuttosto che da pochi pezzi «difficili».

### Karnaugh e gli implicanti

In barba a tutto quello che abbiamo detto poche righe fa, mettiamoci nell'ottica di voler risparmiare a tutti i costi elementi. In altre parole ci proponiamo di realizzare reti funzionalmente equivalenti a quelle ricavabili dalle tabelle di verità, con un numero inferiori di componenti. Per fare questo adopereremo le note (?) mappe di Karnaugh, e il metodo degli implicanti. Procediamo con ordine.

Le mappe di Karnaugh costituiscono semplicemente un modo diverso dalle tabelle di verità per descrivere come si comporta per i vari input ogni terminale di uscita di una rete. Il caso più semplice è ovviamente quello di una rete con due soli ingressi (reti a un solo ingresso possono solo essere il NOT o l'identità): in questo caso la mappa ha dimensione  $2 \times 2$  dove le righe sono etichettate dai valori di un ingresso le colonne dai valori dell'altro ingresso (figura 7A). Nelle caselle, in figura 7, sono indicati dei generici puntini, metteremo 0 o 1 a seconda del valore della funzione sugli ingressi di riga e colonna corrispondenti. In figura 7B troviamo la mappa per un termi-

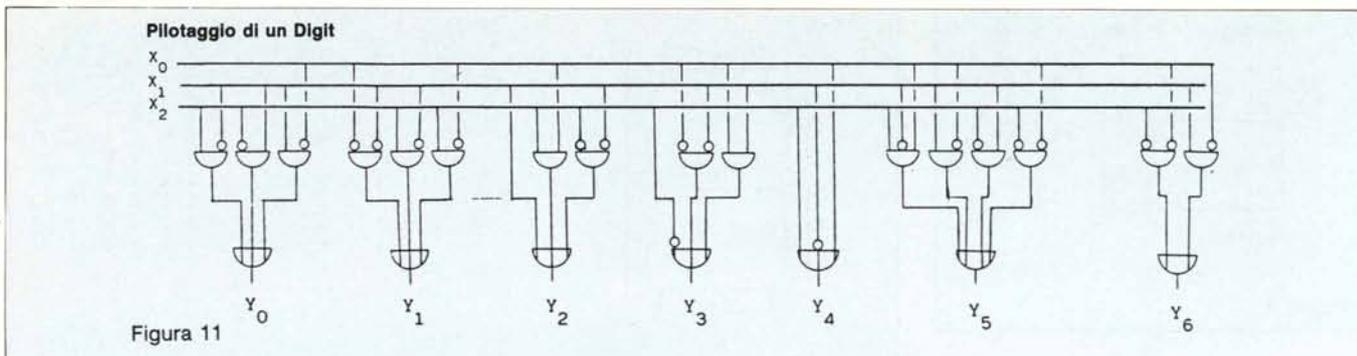


### Quanti lettori?

Proprio in questi giorni sono arrivate in redazione alcune migliaia (si fa per dire) di lettere. Praticamente tutte iniziavano con... non c'è due senza tre!!! Pare infatti che a leggere Appunti di informatica siano in 6 o 7 e non 2 come evidenziato sul numero 61. Matrice comune di tutte queste lettere, oltre ai complimenti per la rivista (qualcuno, poveraccio, s'è anche complimentato per la rubrica in questione), dicevamo matrice comune la soluzione del quesito proposto dal lettore di Monfalcone, con tanto di dimostrazioni del

fatto che il 4 non compare mai ecc. ecc. Sapete che vi dico?

Bravi! Però nessuno ha detto di mandare a noi le soluzioni. Considerato però che tra le lettere giunte ce n'è una disprezzata di un lettore che non è riuscito a scoprire l'enigma, quasi quasi sul prossimo numero faremo una bella selezione delle soluzioni più interessanti. A causa dei tempi tecnici della rivista non è stato possibile farlo questo mese (a stento siamo riusciti a infilare questo riquadretto) indiper-cui-pocchia il mese prossimo faremo una pausa circa le reti logiche. Appuntamento dunque tra trenta giorni.



nale di uscita di una rete a 3 ingressi, in figura 7C quella per reti a 4 ingressi. Si noti come siano state accorpate le variabili: sulle righe o sulle colonne abbiamo i valori congiunti di due variabili. Altra cosa importante, il fatto che tra una colonna e l'altra (e tra una riga e l'altra nel caso di fig. 7C) varia sempre una variabile alla volta: dalla coppia 00 si passa a 01, da questa si passa a 11 e da questa a 10: il tutto funziona anche da una sponda all'altra, da 10 si passa a 00, come se la mappa fosse un «doppio cilindro». Ciò è importante per poter evidenziare gli implicanti che ora tratteremo.

Si tratta di evidenziare sulla mappa zone quanto più estese possibile di 1, formate da 1-2-4-8 caselle adiacenti in modo da formare rettangoli o quadrati. Dal momento che le mappe godono, per così dire, di simmetria circolare, una di queste aree (gli implicanti) può anche iniziare da un estremo e terminare sull'altro, «passando da dietro alla mappa». Giusto per fare qualche esempio, in figura 8 sono mostrate delle mappe a tre ingressi in cui abbiamo evidenziato 5 tipici implicanti. Ad esempio, in figura 8A abbiamo evidenziato l'implicante di quattro 1 disposti a quadrato. In figura 8B e 8C, rispettivamente, implicanti rettangolari da 2 a 4 caselle. In figura 8D e 8E due implicanti di quelli «che girano da dietro alla mappa».

Detto questo, la domanda più ovvia è: «a che servono questi benedetti implicanti?». Semplicissimo: un implicante identifica un termine AND del-

l'espressione che stiamo costruendo e una volta evidenziati tanti implicanti quanti bastano a coprire tutti gli 1 della mappa possiamo passare alla espressione corrispondente e da questa alla rete.

Il problema è semmai quello di capire un dato implicante quale termine identifica. Per risolvere questo piccolo arcano, cominciamo dall'implicante di figura 8A: notiamo che gli 1 in questione restano tali indipendentemente dal valore di X2 e dal valore di X0. Ciò significa che basta che X1 valga 1 che il valore della Y è 1. Questo implicante identifica allora il termine X1. Passiamo a figura 8B: lì gli 1 evidenziati non dipendono da X2 ma sono tali quando X1 vale 0 e X0 vale 1. L'implicante corrispondente è:

$$\bar{X}1 \cdot X0$$

Discorso simile al primo per l'implicante di figura 8C: basta che X2 valga 0, quindi l'implicante è:

$$\bar{X}2$$

Per le figure 8D e 8E gli implicanti corrispondenti, sempre per gli stessi motivi, sono rispettivamente:

$$\bar{X}0 \text{ e } \bar{X}0 \cdot X2$$

Riassumendo, una volta evidenziati tanti implicanti (più grandi possibile, anche a costo di sovrapporre parte di questi) quanti bastano a coprire tutti gli 1 della mappa, ricaviamo i corrispondenti termini AND da mettere in OR come facevamo col metodo delle

tabelle. Anche in questo caso possiamo muoverci in senso diametralmente opposto, evidenziando zone di 0 (gli implicati) ed ottenere espressioni di prodotti di somme: guardando la mappa è facile comprendere se è meglio evidenziare gli implicanti o gli implicati.

Facciamo ora un esempio concreto: torniamo alla tabella di figura 4A e, per prima cosa costruiamo la mappa corrispondente, mostrata in figura 9A. Riusciamo ad evidenziare facilmente due implicanti, uno di 2 e uno di 4 caselle. Il primo identifica il termine  $X1 \cdot X0$ , il secondo il termine X2. L'espressione corrispondente:

$$Y = X2 + X1 \cdot X0$$

e la rete corrispondente di figura 9B parlano da sole: esse sono funzionalmente equivalenti a quelle di figura 4B: come risparmi non c'è male!

## Un bel digit

Per concludere questa sessione combinatoria delle reti logiche, una piccola applicazione paleo-computereccia: piloteremo un digit a sette segmenti. Dando in ingresso un numero compreso tra 0 e 7 in binario (per utilizzare tre sole linee) vedremo apparire sul nostro digit il valore decimale dato dall'accensione dei necessari segmenti.

La nostra rete logica combinatoria avrà dunque tre ingressi (X2, X1, X0) e sette uscite (Y0...Y6) che come detto piloteranno i sette segmenti mostrati in figura 10A. In figura 10B troviamo la tabella di verità dell'intera rete. In figura 10C una per una le mappe relative ai 7 morsetti di uscita, già complete di espressioni relative agli implicanti. A proposito di queste, vogliamo farvi notare quella relativa a Y5, dove troviamo un implicante A di dimensione 1 che identifica un termine di tre variabili: proprio lo stesso che avremmo trovato sviluppando la rete secondo il metodo delle tabelle di verità.

La rete completa è mostrata in figura 11, un bel lavorone, ma alla fine funziona...

## Uffa!

Questa volta l'ho sparata grossa. Errata corrige: MC n. 61 pagina 152 dove sta scritto «Se ad esempio il nostro insieme è finito ed è formato da 4 elementi l'insieme delle funzioni da tale insieme in se stesso è 2 alla 4 dunque 16...» non è vero un tubo! Si vede proprio che io i conti non li so proprio fare. Infatti 4 alla 2 sono il numero delle funzioni da 4 elementi in 2 elementi, mentre il numero delle fun-

zioni da 4 in 4 è 4 alla 4 ovvero 256 (almeno credo... che faccia 256). Quindi la profezia di fine paragrafo (cfr. sempre la stessa pagina) pare più una maledizione che altro. Stile legge di Murphy.

Ovviamente, fermo restando che il resto, Murphy permettendo, dovrebbe essere giusto, in particolare per quel che riguarda gli insiemi infiniti. Non chiedetemi comunque di dimostrarvelo... non ci casco più. Scusate il contrattempo. Grazie. Arrivederci. Punto.

# AVETE MAI PENSATO CHE...



**VELOCI SPEDIZIONI  
IN TUTTA ITALIA**

LA C.D.C. importa direttamente dai costruttori di INTERFACCE, MAIN BOARD, TASTIERE, CASES, ecc. **solo le parti staccate** per garantire il meglio della produzione orientale ed inoltre ASSEMBLA in proprio effettuando un TEST PRELIMINARE DI FUNZIONAMENTO.

LA C.D.C. inserisce sui propri PC/XT/AT\* da SEMPRE solo ed esclusivamente i DRIVE CHINON che sono sinonimo di qualità, silenziosità, ed affidabilità.

LA C.D.C. è organizzata in modo da avere SEMPRE pronto a magazzino quanto Vi occorre e può effettuare spedizioni ANCHE IN GIORNATA (SERVIZIO RAPIDO PER LE ISOLE 24 ORE IN PREPAGATO).

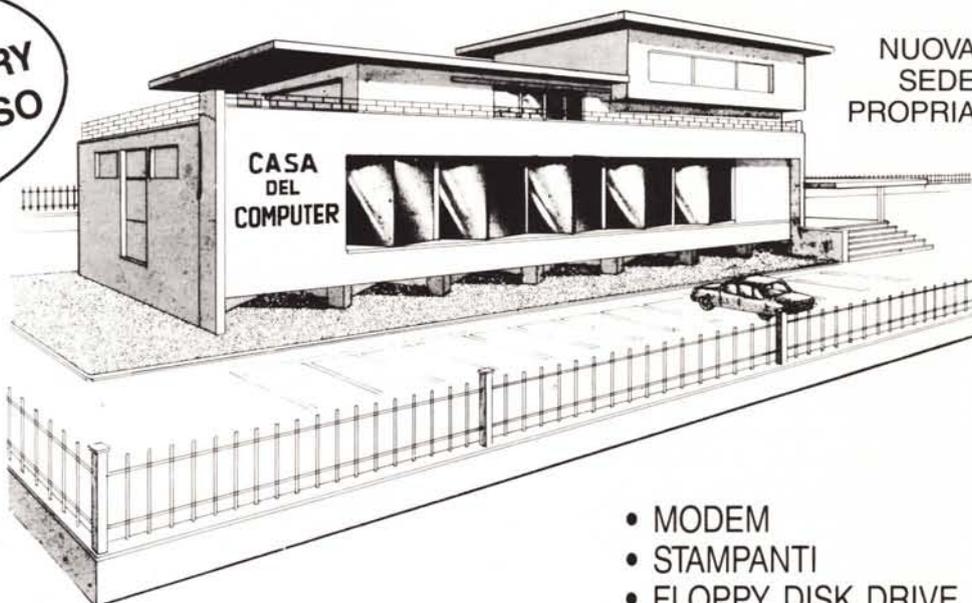
LA C.D.C. GARANTISCE i propri prodotti con la sostituzione immediata o riparazione ANCHE DOPO IL PERIODO DI GARANZIA (servizio HALF COST).

LA C.D.C. ha tutti i pezzi di ricambio a magazzino degli articoli di propria importazione che vengono conservati per minimo 5 ANNI.

**SPESSO È MEGLIO SPENDERE QUALCOSA IN PIÙ PER SPENDERE MENO...**

## ... PENSATECI...!!!

**1°  
CASH & CARRY  
ALL'INGROSSO**



**DEPOSITI:**

**ROMA**

**H2S s.r.l.**

Via Assisi, 80

Tel. 06-7883697

**BOLOGNA**

**TELETEX s.r.l.**

Via Emilia, 51

Anzola Emilia (Bo)

Tel. 051-734485

- PC/XT/AT COMPATIBILI
- PC/XT PORTATILI
- INTERFACCE x APPLE/IBM

- MODEM
- STAMPANTI
- FLOPPY DISK DRIVE
- HARD DISK
- STREAMER
- MONITOR
- DISKETTE

LA CASA DEL COMPUTER - VIA DELLA MISERICORDIA, 84 - 56025 PONTEDERA (PI) - Tel. 0587 - 212.312  
(NUOVA SEDE) - VIA T. ROMAGNOLA, 63 - 56012 FORNACETTE (PI) - Tel. 0587 - 422.022

**RICHIEDETECI IL CATALOGO E PREVENTIVI OGGI STESSO!!**