coordinamento di Andrea de Prisco

Reti logiche combinatorie

di Anna Pugliese

I sistemi di elaborazione

Con il termine «sistema di elaborazione» è possibile riferirsi ad una vasta gamma di macchine logiche composte da elementi circuitali opportunamente connessi fra loro al fine di elaborare sequenze di informazioni producendo, a partire da esse, ben definite sequenze di risultati. Un approccio realistico dell'argomento deve comunque essere subordinato ad una serie di semplificazioni, che rendano possibile la trattazione di tali sistemi, con il ben noto metodo delle «approssimazioni successive». È questo lo spirito mediante il quale «Appunti di Informatica» si propone di trattare questa materia, sicuramente oscura per gran parte degli utenti di computer nonché per coloro ai quali l'appellativo di «utente» sta, e a ragione, abbastanza stretto.

Il fine, che è quello di saperne di più su cosa si celi sotto un austero chip, è onesto dirlo, è di scarsa rilevanza pratica, ma il gusto di saperne di più, si sa, è quello che spinge a comprare il primo PC e a firmare cambiali per l'acquisto del secondo.

La figura 1 può essere d'aiuto per avere un'idea di cosa sia un sistema d'elaborazione; in essa è schematizzata la parte centrale di un sistema.

Un sistema è caratterizzato dall'insieme delle operazioni che esso è capace di eseguire.

gni operazione del sistema è associata ad un codice operativo che presentandosi in input alla struttura centrale del sistema, assieme ad eventuali dati, sui quali l'operazione deve essere effettuata, attiva l'esecuzione dell'operazione nella struttura centrale, che provoca l'uscita di eventuali risultati in output.

Il passo successivo è quello di entrare nella scatola, che in figura 1 è stata chiamata «struttura centrale» del sistema, per osservare come l'esecuzione dell'operazione possa prendere atto. Ma all'interno della scatola, come i più pessimisti avranno già previsto, altre scatole sono presenti, opportunamente connesse fra loro e con gli input e gli output della scatola esterna. Fra i componenti della struttura centrale dei sistemi di elaborazione, è possibile riconoscere le «reti logiche», componenti altresì di molte altre apparecchiature elettroniche.

Tali reti rientrano in due grandi categorie: le reti logiche combinatorie e le reti logiche sequenziali; sebbene le seconde siano, da un punto di vista strutturale, delle semplici varianti delle prime, alla base di ognuna delle due tipologie di reti sta un'apposita teoria.

È alla teoria, nonché alla pratica, delle reti logiche combinatorie che è il caso di rivolgere lo sguardo in questa sede.

X	Y	Z
0	0	1
0	1	0
1	1	1
1	0	0

Figura 3 - Una tabella di verità di una rete combinatoria con due ingressi ed una uscita.

Una rete combinatoria è composta da un insieme di circuiti elementari, mediante i quali essa è capace di manipolare i segnali presenti in input e di presentarli opportunamente elaborati in output.

Con riferimento alla figura 2, utilizzeremo la seguente terminologia ed i seguenti simboli.

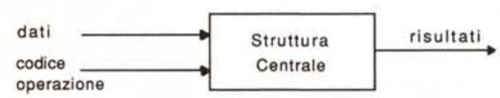


Figura 1

Figura 2

Chiameremo «morsetti» di una rete combinatoria, gli ingressi e le uscite della stessa. A questi morsetti è possibile applicare dei segnali in configurazione binaria; vale a dire che ad ogni singolo morsetto è possibile applicare o meno della corrente elettrica al fine di implementare fisicamente quello che in teoria è il «valore» del morsetto: i possibili valori di un morsetto sono due, e si indicano con i simboli «0» e «1».

Per fare un esempio consideriamo una rete combinatoria con due ingressi ed una uscita.

Chiamiamo X ed Y i due morsetti d'ingresso, e Z quello d'uscita. Dire che X=0 ed Y=1, significherà allora, dal punto di vista fisico, che al morsetto Y è applicata una tensione di 5 volt mentre al morsetto X no. O, se volete, il contrario: tutto è relativo. Una volta stabilito il modo di implementare fisicamente la rete, è ovviamente necessario attenersi

AND	OR	NOT
$0 * 0 = 0$	$0 + 0 = 0$	$\overline{0} = 1$
$0 * 1 = 0$	$0 + 1 = 1$	$\overline{1} = 0$
$1 * 0 = 0$	$1 + 0 = 1$	
$1 * 1 = 1$	$1 + 1 = 1$	

Figura 4 - I postulati dell'algebra della commutazione.

Figura 5 - Le tabelle di verità delle reti elementari.

X	Y	Z
0	0	0
0	1	1
1	1	1
1	0	1

(a)

X	Y	Z
0	0	0
0	1	0
1	1	1
1	0	0

(b)

X	Z
0	1
1	0

(c)

alle convenzioni, ma inizialmente la scelta è libera.

Ciò che da un punto di vista logico caratterizza fortemente le reti combinatorie, è il fatto che esse implementano delle vere e proprie funzioni logiche che trasformano i valori in ingresso nei valori in uscita. In altri termini, è possibile definire completamente la rete combinatoria dell'esempio precedente, dichiarando quali sono le uscite corrispondenti ad ogni possibile combinazione degli ingressi. La cosa è fattibile mediante una tabella come quella della figura 3, detta: «Tabella di verità».

Una rete combinatoria, il cui funzionamento è definito dalla tabella di verità in figura 3, è una rete che implementa una funzione logica ben precisa.

Se indichiamo con «Vero» e «Falso» i segnali «1» e «0» rispettivamente, la funzione logica di prima può essere enunciata come: «Il risultato è vero se, e solo se, i due dati di ingresso sono uguali tra loro».

Mediante le tabelle di verità è possibile definire funzioni logiche di ogni genere e tipo. Vedremo come.

L'algebra della commutazione

Con il termine «algebra della commutazione» ci si riferisce ad un sistema algebrico in cui gli unici valori esistenti sono «0» ed «1», e le operazioni eseguibili sono la somma logica, il prodotto logico, e la complementazione logica. Queste tre operazioni sono indicate rispettivamente con «OR», «AND», e «NOT», e sono definite come illustrato nelle tabelle di figura 4.

A partire da questi postulati di base è possibile ricavare un certo numero di proprietà dell'algebra della commutazione.

Siano X ed Y due variabili. È evidente dall'osservazione della figura 4 che:

$$X + Y = Y + X$$

dal momento che, qualunque sia il valore (0 oppure 1) di X e qualunque sia il valore (0 oppure 1) di Y, il risultato della loro somma logica non cambia se si cambia l'ordine degli addendi.

Analogamente è possibile dimostrare che:

		$X * Y = Y * X$
$X + \overline{X} = 1$	;	$X * \overline{X} = 0$
$X + 0 = X$	;	$X * 1 = X$
$X + 1 = 1$	;	$X * 0 = 0$
$\overline{\overline{X}} = X$		e così via.

Torniamo alle nostre reti combinatorie.

Reti elementari

Consideriamo una rete combinatoria la cui tabella di verità corrisponde a quella in figura 5a.

È facile convincersi del fatto che tale rete combinatoria implementa la funzione logica OR, vale a dire che l'uscita Z della rete è sempre uguale alla somma logica degli ingressi X ed Y, qualsiasi siano i valori degli ingressi stessi. Analogamente, in figura 5b è mostrata la tabella di verità di una rete combinatoria che implementa la funzione logica AND, ed in figura 5c quella della funzione logica NOT. Queste tre reti combinatorie rivestono grossa importanza, dal momento che costituiscono le «reti elementari»: ogni possibile rete combinatoria può essere ottenuta dalla connessione di reti elementari.

Dopo aver dato un'occhiata alla figura 6, che mostra le tre reti elementari in una rappresentazione meno astratta di quella della figura 2 o di quella delle corrispondenti tabelle di verità, andiamo a dimostrare come sia possibile realizzare la rete dell'esempio di figura 3, mediante la connessione di reti elementari.

La rete produceva in uscita il valore Z=1 se X ed Y erano uguali (entrambi 0 od entrambi 1) e Z=0 altrimenti.

Un'espressione logica che fornisce una simile Z è la seguente:

$$Z = \overline{X} * \overline{Y} + (X * Y)$$

La dimostrazioned la diamo mediante il cosiddetto metodo della «perfetta induzione» che consiste nel provare la veridicità della tesi per ogni possibile caso particolare. Per farlo, utilizziamo una specie di tabellone, nelle prime colonne del quale inseriamo le variabili X ed Y, da cui l'espressione dipende, nelle colonne intermedie inseriamo dei risultati parziali cioè delle sottoespressioni quali: $\overline{X}$, $\overline{Y}$, $\overline{X}\ \overline{Y}$, $X*Y$ e, nell'ultima colonna, l'espressione intera.

A questo punto basta cancellare mentalmente le colonne intermedie e verificare se la tabella così ottenuta corrisponde alla tabella di verità di figura 3, che è quella della rete che vogliamo realizzare. Dimostrata la cosa, trascuriamo per il momento come sia stato

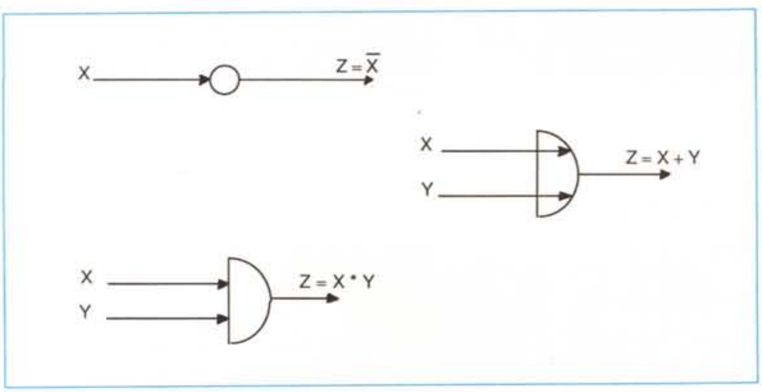


Figura 6 - Gli schemi logici delle reti elementari.

possibile ottenere l'espressione logica che descrive Z, e diamo invece un'occhiata alla figura 8, dove gli elementi logici OR, AND e NOT, sono interconnessi per produrre, a partire dai morsetti d'ingresso X ed Y, il morsetto d'uscita Z il cui valore corrisponde a $Z = (\overline{X}*\overline{Y}) + (X*Y)$.

Confrontando fra loro la figura 8 con la figura 2, possiamo facilmente renderci conto del fatto che la tabella di verità è una descrizione generale del funzionamento della rete, detta: «descrizione ai morsetti»; nel momento in cui passiamo all'implementazione effettiva della rete, allora dobbiamo andare a vedere quale dev'essere il contenuto della «scatola nera» cui i morsetti della rete sono applicati. Questa scatola nera conterrà sempre altre scatole nere, che nel caso di figura 8 sono le reti elementari AND, OR e NOT i cui morsetti sono opportunamente connessi fra loro. Vedremo, in successive puntate della rubrica, come più reti combinatorie possano essere connesse tra loro, e con altri elementi detti «registri», per formare dei cosiddetti sistemi di reti sequenziali, categoria alla quale appartengono i processori dei computer.

In un certo senso possiamo dire che in questa trattazione dei processori, stiamo seguendo il metodo «bottom-up» (=dal basso in alto) nel senso che, a partire dalle reti elementari, che nella scienza dell'informazione costituiscono dei componenti atomici, non ulteriormente divisibili (una loro trattazione richiederebbe infatti nozioni squisitamente elettroniche), costruiamo oggetti più complessi sulla base dell'integrazione di quelli più semplici. Prima di continuare questo nostro processo d'astrazione, è necessario soffermarci ancora sulle reti combinatorie e precisamente sulle tecniche di progettazione delle stesse.

Ribadendo che la tabella di verità di una rete, ne costituisce una sua descrizione funzionale, chiameremo il processo di costruzione vero e proprio della rete, a partire dalla sua descrizione funzionale: processo di «sintesi» della rete, mentre il processo inverso, che consiste nell'ottenere la tabella di verità della rete a partire dalle connessioni esistenti fra le reti elementari che la compongono, lo chiameremo: processo di «analisi» della rete. In questo contesto dovrebbe essere chiaro che il processo di sintesi di una rete consiste nel suo progetto, e poiché è proprio questo quello che ci riguarda più da vicino, tratteremo del progetto delle reti combinatorie mediante un vero e proprio esempio di sintesi.

Alla base della sintesi delle reti combinatorie, si trovano dei concetti espressamente matematici che riguardano le funzioni dell'algebra della commutazione. Per ovvi, ed immagino condivisi, motivi questa sede non ci permette di divagare liberamente su tali problematiche, che verranno perciò esposte nella maniera più elementare possibile.

X Y	$\overline{X}$	$\overline{Y}$	$\overline{X}\cdot\overline{Y}$	$X\cdot Y$	$\overline{X}\cdot\overline{Y}+X\cdot Y$ (=Z)
0 0	1	1	1	0	1
0 1	1	0	0	0	0
1 1	0	0	0	1	1
1 0	0	1	0	0	0

Figura 7

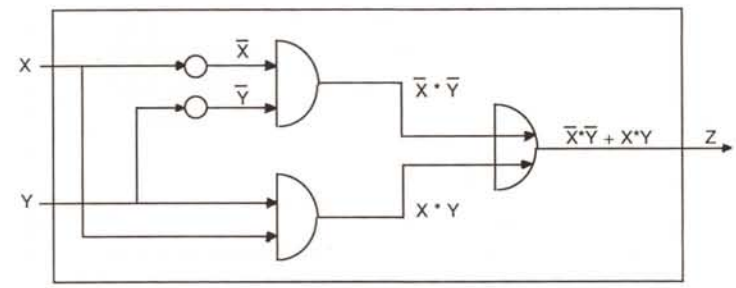


Figura 8

X Y Z	f
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	1

(a)

Z \ XY	0 0	0 1	1 1	1 0
0	0	0	1	0
1	0	1	1	1

(b)

Figura 9

Le mappe di Karnaugh

Una mappa di Karnaugh (dal nome dell'inventore) è fondamentalmente un modo diverso di rappresentare quelle stesse informazioni contenute nella tabella di verità di una rete combinatoria.

Osserviamo la figura 9. In essa è riportata la descrizione funzionale di una rete combinatoria, sia con il metodo della tabella di verità che con quello della corrispondente mappa di Karnaugh.

La tabella di verità della figura 9a, descrive la funzione logica calcolata da una rete combinatoria con tre morsetti d'ingresso (x, y, e z) ed un morsetto d'uscita (f). L'insieme di tutte le possibili configurazioni di ingresso è dato dalle combinazioni di 3 variabili binarie, ed è quindi grande $2^3=8$. Nella prima colonna della tabella di verità, queste 8 configurazioni possono essere inserite in qualsiasi ordine; in particolare noi abbiamo scelto come ordinamento, dall'alto in basso, quello di elencare tali combinazioni come se fossero i primi 8 numeri binari (da 0 a 7); così, dopo la configurazione 000 si presenta la 001, poi la 010 e così via.

Qualsiasi altro ordine sarebbe andato bene lo stesso, poiché lo scopo è quello di elencare tutte le configurazioni d'ingresso che potrebbero presentarsi in un generico istante sulla rete, al fine di specificare il valore dell'uscita f corrispondente ad ognuna di esse.

Lo scopo di una mappa di Karnaugh è leggermente diverso. Anche mediante la mappa viene descritta la funzione logica calcolata dalla rete: il valore di f corrispondente all'input xyz=011 è inserito nella casella che si trova all'incrocio tra la colonna xy=01 e la riga z=1, cioè f=1. Una volta compreso il meccanismo, non è difficile accertarsi che la mappa e la tabella di figura 9 sono equivalenti. La differenza consiste nel fatto che, nella mappa, le caselle corrispondenti a combinazioni dei valori d'ingresso che differiscono per il valore di una sola variabile, occupano posizioni adiacenti.

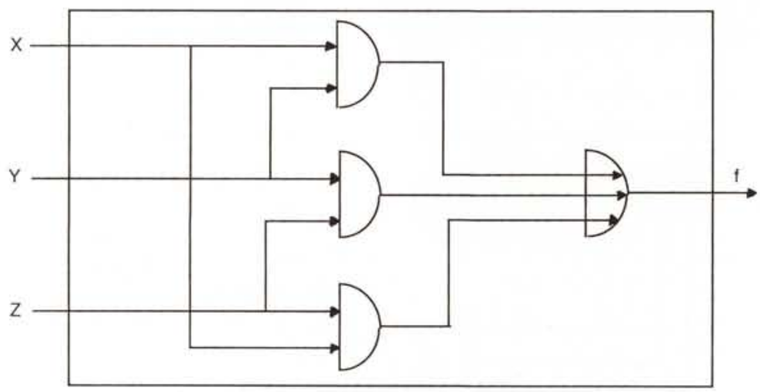


Figura 10

Ad esempio, le configurazioni d'ingresso 000 e 010, differendo per il solo valore della variabile Y, sono dette «combinazioni adiacenti»; le corrispondenti caselle nella mappa, sono rispettivamente: quella in alto a sinistra e quella immediatamente alla destra di quest'ultima, cioè: sono adiacenti. Adiacenti possono considerarsi anche le due caselle in terza colonna: esse sono 110 e 111 e differiscono quindi per la sola variabile Z. Adiacenti, infine, sono anche le due caselle corrispondenti alle combinazioni 000 e 100, sebbene sulla mappa «sembrino» non esserlo: il trucco sta nel considerare la mappa come se fosse una rappresentazione, sulla carta, di una sfera. Riassumendo, ogni casella è adiacente alla casella che sta alla sua destra, a quella alla sua sinistra, a quella che gli sta sopra ed a quella che gli sta sotto, con l'intesa che se, ad esempio, una casella sta all'estrema destra, va considerata come casella adiacente destra ad essa, quella che sta dalla parte opposta (estrema sinistra) sulla stessa riga.

Vediamo adesso come la faccenda delle combinazioni adiacenti, possa rivelarsi utile ai fini del nostro processo di sintesi della rete. Procediamo, come sempre, esemplificando.

Se volessimo indicare che la funzione f vale 1 per xyz=110, noi potremmo dire che $f=x*y*\overline{z}$. L'espressione $x*y*\overline{z}$ infatti, vale 1 se: x=1, y=1, e z=0, in quanto se z=0 allora $\overline{z}=1$, ed il prodotto logico di tre variabili che valgono 1, dà come risultato appunto 1. Ma l'uscita f della nostra rete, vale 1 non solo per xyz=110, ma anche per xyz=111, xyz=011 e xyz=101. Potremmo allora dire che:

$f=x*y*\overline{z} + x*y*z + \overline{x}*y*z + x*\overline{y}*z$.

Confrontando attentamente questa espressione con la mappa di figura 9, ci accorgeremo che l'espressione è composta dalla somma logica di 4 termini ognuno dei quali è il prodotto logico delle 3 variabili d'ingresso, complementate oppure non complementate, in modo tale che ogni termine identifichi una delle caselle della mappa, il cui contenuto è 1.

In altre parole: f vale 1 se ci troviamo davanti ad un input che cade in una delle caselle contenenti 1, vale 0 altrimenti. La cosa, fin qui, è più facile da capire che da spiegare.

Procediamo.

L'espressione trovata per F è di per sé un'esatta formulazione della funzione calcolata dalla rete, e perciò stesso è implementabile mediante 4 porte AND a tre ingressi ciascuna, 1 porta OR a quattro ingressi e 3 porte NOT.

Ci viene in aiuto il concetto di adiacenza. I primi due termini di f, sono: xyz ed $xy\overline{z}$, i quali designano le due caselle della terza colonna della mappa di Karnaugh; tali caselle sono adiacenti, cioè differiscono per il solo valore della variabile z; qualunque sia z, infatti, f varrà 1. Questo significa che possiamo riscrivere l'espressione di f nel seguente modo:

$f=x*y + \overline{x}*y*z + x*\overline{y}*z$. Per quanto ne sappiamo, in questa nuova (ma equivalente) espressione di f, non esistono altre combinazioni adiacenti. La cosa è invece falsa, in quanto il termine x*y designa ora una coppia di caselle, cioè: x*y*z ed $x*y*\overline{z}$. Orbene la prima di queste due caselle è adiacente sia ad $\overline{x}*y*z$ che ad $x*\overline{y}*z$, per cui l'espressione di f può essere ulteriormente semplificata ottenendo: f= x*y + y*z + x*z, il che ci permette di implementare la rete con un notevole risparmio di porte AND, OR e NOT.

Prima di concludere, vediamo come tutto questo discorso poteva essere fatto direttamente sulla mappa di Karnaugh. A partire da tale mappa, avremmo potuto raggruppare le caselle adiacenti il cui valore è 1, a due a due, essendo sempre possibile trovare un termine per designare due caselle adiacenti (basta non inserire nel termine, la variabile che differisce nelle due caselle).

Con tale metodo avremmo ottenuto i tre seguenti raggruppamenti:

1) le due caselle della terza colonna, mediante il termine x*y (escludendo la variabile cangiante z);
2) la casella nella riga 2 colonna 2 con quella della riga 2 colonna 3, mediante il termine y*z (escludendo x);
3) la casella di riga 2 colonna 3 con quella di riga 2 colonna 4, mediante x*z (escludendo y).

Tale metodo (che può essere applicato anche a quadruple ed in generale ad n-uple di caselle, con n multiplo di 2), ci porta all'espressione semplificata già ottenuta per f, cioè: f=xy + yz + xz, che porta alla rete di figura 10.

Tale rete combinatoria, corrisponde, come è facile rendersi conto osservando la tabella della verità, ad un «Votatore» di bit, cioè ad un componente integrato che produce in uscita il risultato di una votazione a maggioranza effettuata sugli ingressi. Un simile componente può essere utilizzato in veri e propri processor per aumentare la correttezza dei sistemi.

DIGITIZER COSMOGRAPH

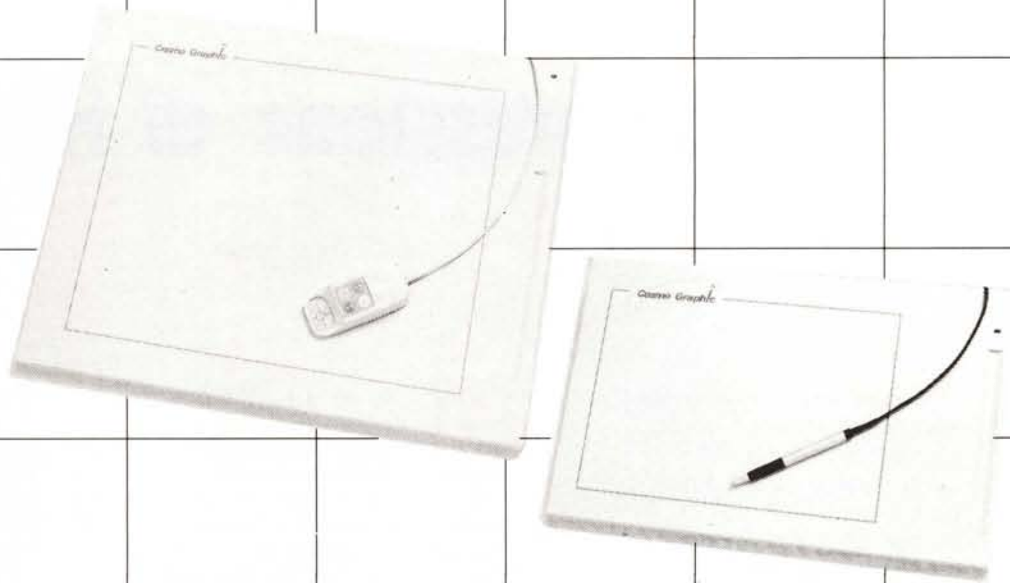
Caratteristiche tecniche
- Input system: Electrostatic Couplig Method
- Input range: 210x300 mm (A4), 420x330 mm (A3)
- Scan rate: maximum 60, 30, 15, 5 points/second (A4) maximum 90, 30, 15, 5 points/second (A3)
- Scaling by meter
- Operating mode: POINT, STREAM, SWITCH STREAM 1, SWITCH STREAM 2 (A4) POINT, STREAM, SWITCH STREAM 1, REMOTE (A3)
- Output: RS 232C (1) Binary format (2) BCD format
- Operating temperature: 5 - 35 C
- Operating moisture: 20 - 80%
- Power supply: DC 13 - 14 V (450-550 mA)
- Measurement: 445x280x29.5 mm (A4) 540x423x50 mm (A3)
- Firmware compatibility: Summagraphic*
- Baud rate: 1200/2400/4800/9600
- Accessories: Cursor (option), Connector cable (standard)

* **SUMMAGRAPHIC è un marchio registrato dalla Summagraphic Corp.**

I COSTI
DIGITIZER COSMOGRAPH
formato A4 Lire 681.000 + IVA
formato A3 Lire 998.000 + IVA

EXECUTIVE

PRESENTI ALLO SMAU
CHIOSCO ESTERNO 2.3 PAD.2



DIGITIZER COSMOGRAPH

Disponibili listini per rivenditore

EXECUTIVE COMPUTER DEALER
Via Bovara, 16 - **LECCO**
Uffici e Magazzino:
Via Buozzi, 23
22053 **LECCO** (CO)
Tel. 0341/282614 r.a.
Fax. 0341/283759

Per informazioni:	INDIRIZZO ______
NOME ______	TEL. ______
DITTA ______	ATTIVITA' ______