

Cadell

Spediz. in abbonamento postale GR. II/70 L. 2.000
(...)

23 CORSO PRATICO COL COMPUTER

421750

F4

F5

F6

F7

F8

diretto da **GIANNI DEGLI ANTONI**

Cadell

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**

BARTON

**IN OMAGGIO
IL DECIMO POSTER
"LA STORIA
DELL'INFORMATICA"**

FABBRI
EDITORI

Nel Pocket Computer PC-1500A c'è tutta la potenza di un Home Computer



PC-1500 A
dimensioni: 195 x 25,5 x 86 mm.
429.000 + IVA

**Il Pocket Computer
con 8,5 kbyte di RAM espandibili
e una serie di interessanti accessori.
Per il commerciale, il manager, l'ingegnere e l'hobbista.**

CPU - C-MOS da 8 bit
Permette un'alta velocità d'elaborazione con poco consumo d'energia.

Mini-visore grafico
È in grado di visualizzare fino a 26 caratteri o qualsiasi tipo di rappresentazione grafica grazie ai suoi 1092 punti.

Superiorità operativa
La tastiera tipo macchina da scrivere ne facilita l'uso. Inserito nell'interfaccia opzionale CE-150, il PC-1500A può essere addirittura utilizzato come una piccola macchina da scrivere!

Sei tasti software possono essere utilizzati come tasti funzione, o comandi, o come tasti per la definizione di programmi. Il sistema di rotazione a tre livelli fa svolgere con 6 tasti il lavoro di 18 tasti software. La **funzione di blocco del programma** blocca il tasto MODE; il PC-1500A conserva così il programma lasciando libera solo la funzione RUN. Effettuerete così il programma senza rischiare cancellazioni involontarie.

Linguaggio BASIC più ricco
Gli **statement aggiuntivi**

del BASIC del PC-1500A forniscono variabili che possono essere liberamente definite usando uno o due caratteri, disposte secondo schemi a due dimensioni per il calcolo matriciale, stringhe di variabili, comandi per la grafica.

Funzione orologio
Indica mese, data, ora, minuti e secondi. Ha anche una funzione acustica. Alle scadenze programmate il computer emette dei BIP sonori e visualizza il messaggio.

Altre caratteristiche

- Si può regolare il tono ed il numero delle ripetizioni dei BIP.
- Si possono programmare ed effettuare dei giochi.
- Abbreviazioni e dieci comandi diretti rendono la programmazione più semplice.
- Il dispositivo di spegnimento automatico evita sprechi di energia.
- Premendo i tasti SHIFT o SML si introducono lettere minuscole.
- I programmi e gli accessori per il PC-1500 sono compatibili col PC-1500A e viceversa

Accessori opzionali

L'interfaccia registratore/stampante grafica CE-150

- Come stampante realizza l'hardcopy di programmi, dati e calcoli e offre una funzione grafica a quattro colori: rosso, nero, verde, blu. Stampa i caratteri in nove corpi diversi. Ogni riga ne può contenere da 4 a 36. Questa stampante può essere controllata dall'operatore orientando il senso di stampa verso l'alto, il basso, a destra e a sinistra.
- Come interfaccia si può collegare ad uno o due registratori (uno per la memorizzazione dei dati e programmi, l'altro per richiamarli).

CE-151 - CE-155 (8 e 16 kbyte)

Moduli d'espansione memoria RAM
La RAM può essere ampliata di 4 o 8 kbyte inserendo uno dei due moduli opzionali, ottenendo una capacità totale di 12,5 o 16,5 kbyte.

CE-159 - CE-161 (8 e 16 kbyte)

Moduli di RAM programmabile
I moduli CE 159 e CE 161 possono essere programmati e quindi essere rimossi. La memoria è mantenuta per due anni se il modulo è staccato e per 5 se mantenuto nel PC-1500A. La capacità totale, con uno di questi moduli, sarà di 16,5 o 24,5 kbyte.

RS-232C ed interfaccia parallela CE-158

Questa interfaccia permette la connessione del PC-1500 A con MODEM, accoppiatori acustici, lettori di codici a barre, stampanti seriali o parallele, o con altri computer.

CE-152 Registratore a cassette

Il CE-152 registra programmi o dati. Si possono utilizzare due CE-152 contemporaneamente (uno per leggere, l'altro per registrare).

CE-153 Tavolettina software

La CE-153 ha 140 tasti definiti per facilitare l'elaborazione dei dati.

Direttore dell'opera
GIANNI DEGLI ANTONI

Comitato Scientifico
GIANNI DEGLI ANTONI
Docente di Teoria dell'Informazione, Direttore dell'Istituto di Cibernetica dell'Università degli Studi di Milano

UMBERTO ECO
Ordinario di Semiotica presso l'Università di Bologna

MARIO ITALIANI
Ordinario di Teoria e Applicazione delle Macchine Calcolatrici presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

MARCO MAIOCCHI
Professore Incaricato di Teoria e Applicazione delle Macchine Calcolatrici presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

DANIELE MARINI
Riceratore universitario presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

Curatori di rubriche
TULLIO CHERSI, ADRIANO DE LUCA (Professore di Architettura del Calcolatori all'Università Autonoma Metropolitana di Città del Messico), GOFFREDO HAUS, MARCO MAIOCCHI, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARPELLI, ENNIO PROVERA

Testi
ADRIANO DE LUCA, CLAUDIO PARPELLI, GOFFREDO HAUS, CLAUDIO MORDÀ, Etnoteam (ADRIANA BICEGO)

Tavole
Logical Studio Communication
Il Corso di Programmazione e BASIC è stato realizzato da Etnoteam S.p.A., Milano
Computergrafica è stato realizzato da Eidos, S.c.r.l., Milano
Usare il Computer è stato realizzato in collaborazione con PARSEC S.N.C. - Milano

Direttore Editoriale
ORSOLA FENGLI

Coordinatore settore scientifico
UGO SCAIONI

Redazione
MARINA GIORGETTI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGE

Segretaria di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer - Copyright ©, sul fascicolo 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A. - Milano - Copyright ©, sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A. - Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 135 del 1° marzo 1984 - Iscrizione al Registro Nazionale della Stampa n. 00262, vo. 3, Foglio 429 del 20/9/1982 - Stampato presso lo Stabilimento Grafico di Gruppo Editoriale Fabbri S.p.A. - Milano - Diffusione Gruppo Editoriale Fabbri S.p.A. via Mecenate, 91 - tel. 50951 - Milano - Distribuzione per l'Italia A & G Marco s.a.s. via Forzezza 27 - tel. 2526 - Milano - Pubblicazione periodica settimanale - Anno I - n. 23 - esce il giovedì - Spedizioni in abb. postale - Gruppo II/70 - L'Editore si riserva la facoltà di modificare il prezzo nel corso della pubblicazione, se costretto da mutate condizioni di mercato.

concessionaria
per l'Italia

MELCHIONI

**TUTTA LA POTENZA DI UN COMPUTER
NEL PALMO DELLA TUA MANO**

Per ulteriori informazioni scrivete a
MELCHIONI - Divisione Pocket Computer - 20135 MILANO - Via P. Colletta 37

SHARP

“OPERATE” E CONTROLLO

Istruzioni che coinvolgono registri interni e un primo sguardo al circuito di controllo.

L'istruzione SBB XX (figura 1) è un'istruzione di sottrazione. Come la ADD XX, vista in precedenza, è una istruzione composta e si comporta nello stesso modo fatta eccezione per la fase T_6 dove, in questo caso, abbiamo $S_U = 1$.

Le istruzioni “Operate”

Vediamo adesso le istruzioni di tipo “Operate”; questo tipo di istruzioni va acquistando sempre più importanza a mano a mano che migliora l'architettura dei microprocessori. Se analizziamo un po' attentamente quali sono le cause che ritardano l'esecuzione di un programma, ci rendiamo conto che il ritardo generato dall'accesso alla memoria è senz'altro importante, dovuto, naturalmente, a tutta una serie di problemi tecnici intrinseci agli elementi stessi di memoria. Nelle istruzioni “Operate”, che coinvolgono registri interni degli stessi microprocessori, la velocità di esecuzione risulta più alta: è

conveniente, quindi, che un microprocessore abbia a disposizione molti registri interni. Questa è la tendenza nella realizzazione dell'architettura dei microprocessori moderni.

Istruzioni ADA e ADB

L'istruzione ADA somma i contenuti degli accumulatori A e B (figura 2) e deposita il risultato in A. Questa istruzione si esaurisce nella fase T_2 . L'istruzione ADB è uguale (figura 3), fatta eccezione per il fatto che il risultato è depositato in B.

Istruzioni SUA e SUB

La SUA è un'istruzione di sottrazione tra A e B (figura 4) con deposito del risultato in A; la SUB (figura 5) è identica, ma deposita il risultato in B.

①

ISTRUZIONE SBB XX			
FASE	CON		
T_2	E_{PC}	L_M	
T_3	I_{PC}	L_{AR}	VMA
T_4	E_{AR}	L_M	
T_5	L_B	VMA	
T_6	$E_U S_U L_A$		

Come cambiano i comandi, nelle varie fasi, per l'esecuzione dell'istruzione SBB XX.

②

ISTRUZIONE ADA		
FASE	CON	
T_2	E_U	L_A
T_3		
T_4		
T_5		
T_6		

Come cambiano i comandi per l'istruzione ADA.

③

ISTRUZIONE ADB		
FASE	CON	
T_2	E_U	L_B
T_3		
T_4		
T_5		
T_6		

Come cambiano i comandi per l'istruzione ADB.

④

ISTRUZIONE SUA			
FASE	CON		
T_2	E_U	S_U	L_A
T_3			
T_4			
T_5			
T_6			

Come cambiano i comandi per l'istruzione SUA.

⑤

ISTRUZIONE SUB			
FASE	CON		
T_2	E_U	S_U	L_B
T_3			
T_4			
T_5			
T_6			

Come cambiano i comandi per l'istruzione SUB.

Istruzione HLT

Questa istruzione non richiede nessuna fase di esecuzione, perché la sua presenza nel circuito di controllo attiva un meccanismo di bloccaggio che ferma tutto il processo, come vedremo meglio nello schema a blocchi del controllo, che daremo più avanti.

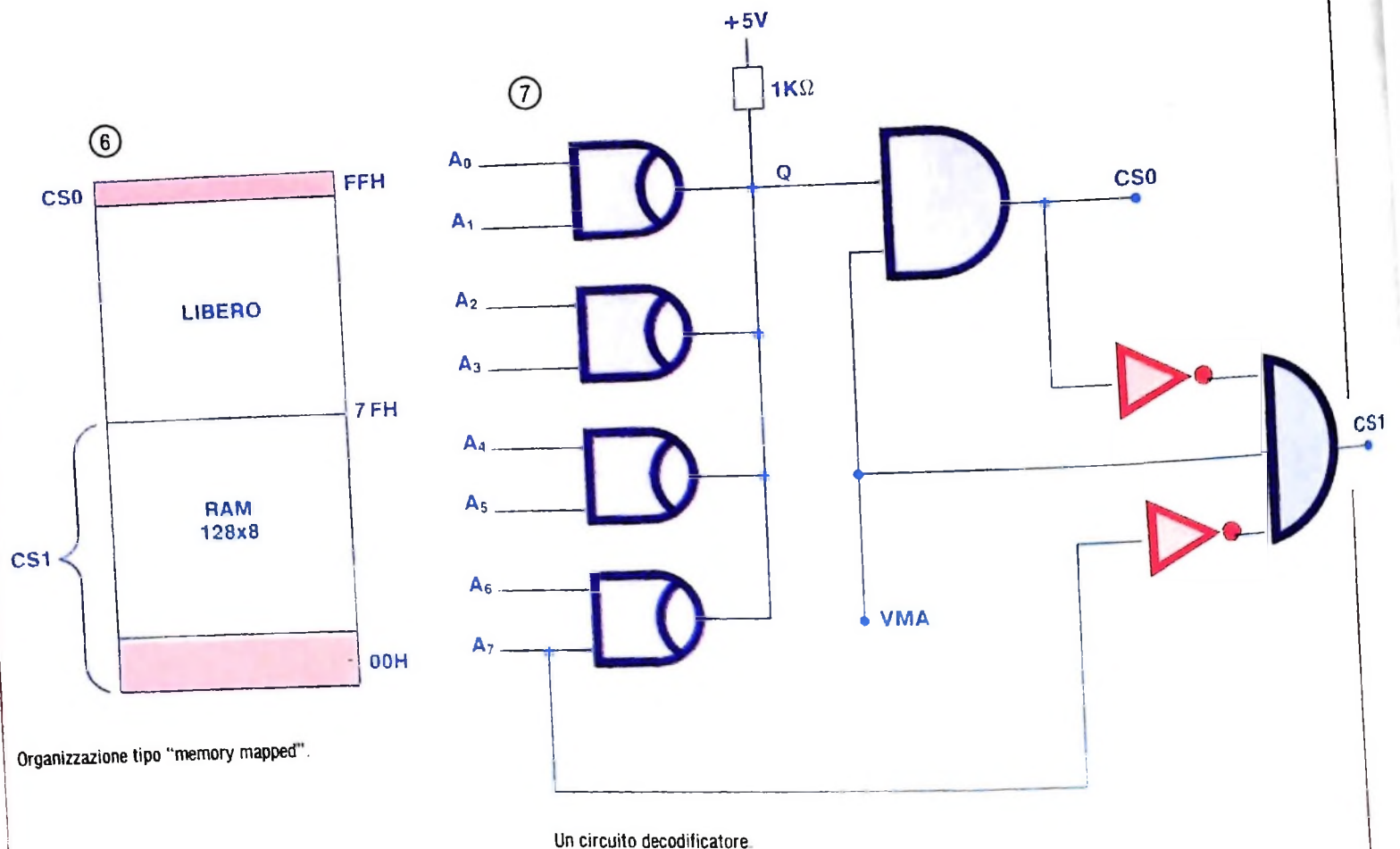
Prima di passare a trattare il complesso circuito di controllo, ci sembra che sia interessante vedere com'è stata realizzata l'organizzazione tipo "memory mapped" nel nostro piccolo sistema. Abbiamo detto che, in definitiva, tale organizzazione consiste nel disporre tutti i registri esterni come celle di memoria con appropriati indirizzi (figura 6). Nel nostro sistema abbiamo a disposizione 256 indirizzi; di questi, 128, a partire da 00H fino a 7FH, sono stati assegnati alla memoria RAM; da 8FH (esadecimale!) fino a FEH abbiamo spazio libero per altri usi come, per esempio, aumentare la memoria, mentre l'ultimo posto, FFH, è stato riservato al registro O. Per poter fare in modo che siano attivi solo i registri giusti, cioè quelli in cui noi vogliamo operare è necessario avere a disposizione un decodificatore.

Il decodificatore è un circuito che viene progettato dopo che si è decisa la configurazione degli elementi esterni, cioè una volta che, secondo le nostre esigenze, abbiamo sistemato gli

elementi a disposizione nello spazio degli indirizzi, dobbiamo individuare un sistema che sia capace di abilitare opportunamente gli spazi prescelti.

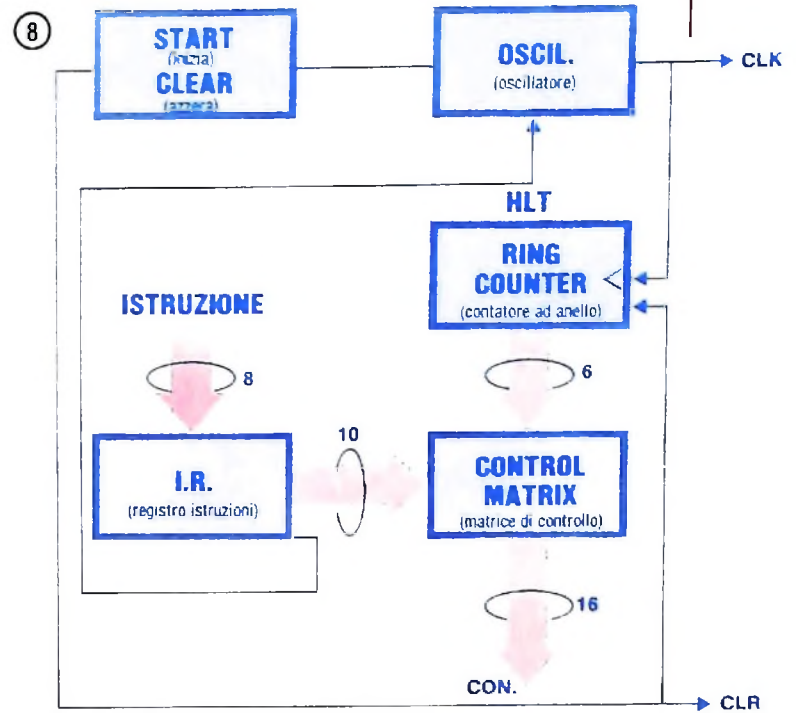
In figura 7 c'è il nostro decodificatore, che funziona nella seguente maniera: poiché per selezionare il registro O abbiamo bisogno che l'indirizzo sia FFH, se questo indirizzo entra nelle entrate da $A_0 \rightarrow A_7$ allora nell'uscita Q abbiamo un 1, che assieme all'1 di VMA permette la fuoriuscita di un 1 in CS0, che abilita appunto il registro O. Naturalmente gli AND di entrata degli indirizzi devono essere AND a collettori aperti, cioè dobbiamo collegarli con una resistenza all'alimentatore. Quando invece $A_7=0$ (indipendentemente dal valore delle altre entrate) avremo $Q=0$; di conseguenza CS0, che diventa 1, entra nell'AND di CS1, lo stesso A_7 , invertito a sua volta, diventa 1 all'entrata dello stesso AND; così, quando VMA convalida l'indirizzo, CS1 diventa 1 e abilita la memoria, come si voleva.

Dopo aver condotto quest'analisi risulta decisamente chiaro che quando, per esempio, vogliamo mostrare sul display un dato presente negli accumulatori non facciamo altro che usare l'istruzione STA FFH e STB FFH (ricordiamo ancora che gli indirizzi degli elementi esterni devono sempre essere in notazione esadecimale) e il nostro dato apparirà sullo schermo del display.

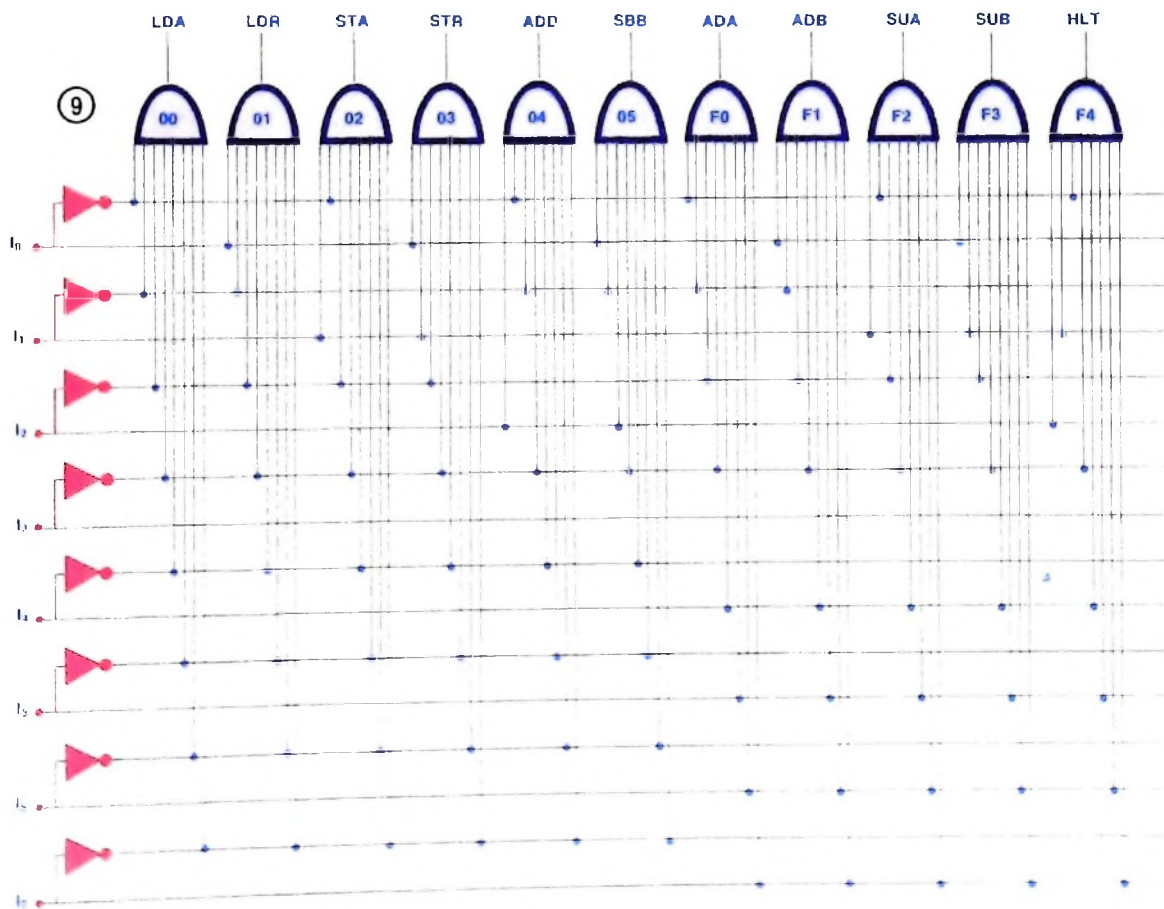


Controllo

Lo schema a blocchi del controllo è dato dalla figura 8. In alto a sinistra abbiamo un circuito chiamato *start clear* (inizia azzerata): in realtà si tratta di due pulsanti che emettono impulsi elettronicamente trattati (senza oscillazioni), per cui quello generato dallo start entra nel circuito oscillatore e inizia l'oscillazione, mentre il secondo non è altro che la linea del bus di controllo chiamata CLR (cancellazione oppure azzeramento). L'oscillatore è un oscillatore a onda quadra, nel nostro caso con una frequenza di 1 MHz, che emette a sua volta una linea di comando, la CLK. Questa linea entra nel *ring counter* (contatore ad anello), che abbiamo già visto, il quale provvede a scandire le fasi di ricerca ed esecuzione di ogni istruzione; per questo emette un piccolo bus interno di 6 linee che entrano nel *control matrix* (matrice di controllo). Il registro IR (registro delle istruzioni) (figura 9) è un circuito combinatorio, in cui, da una parte (in alto), entrano le istruzioni; il valore di ogni bit resta invariato da un lato e invertito dall'altro, cosicché abbiamo i due valori a disposizione per ogni bit. Questi segnali vengono collegati opportunamente alle otto entrate di ogni AND e solo una di queste uscite può essere attiva in un dato istante. Per esempio, prendiamo la istruzione SBB che, ricordate, ha il codice 05H:



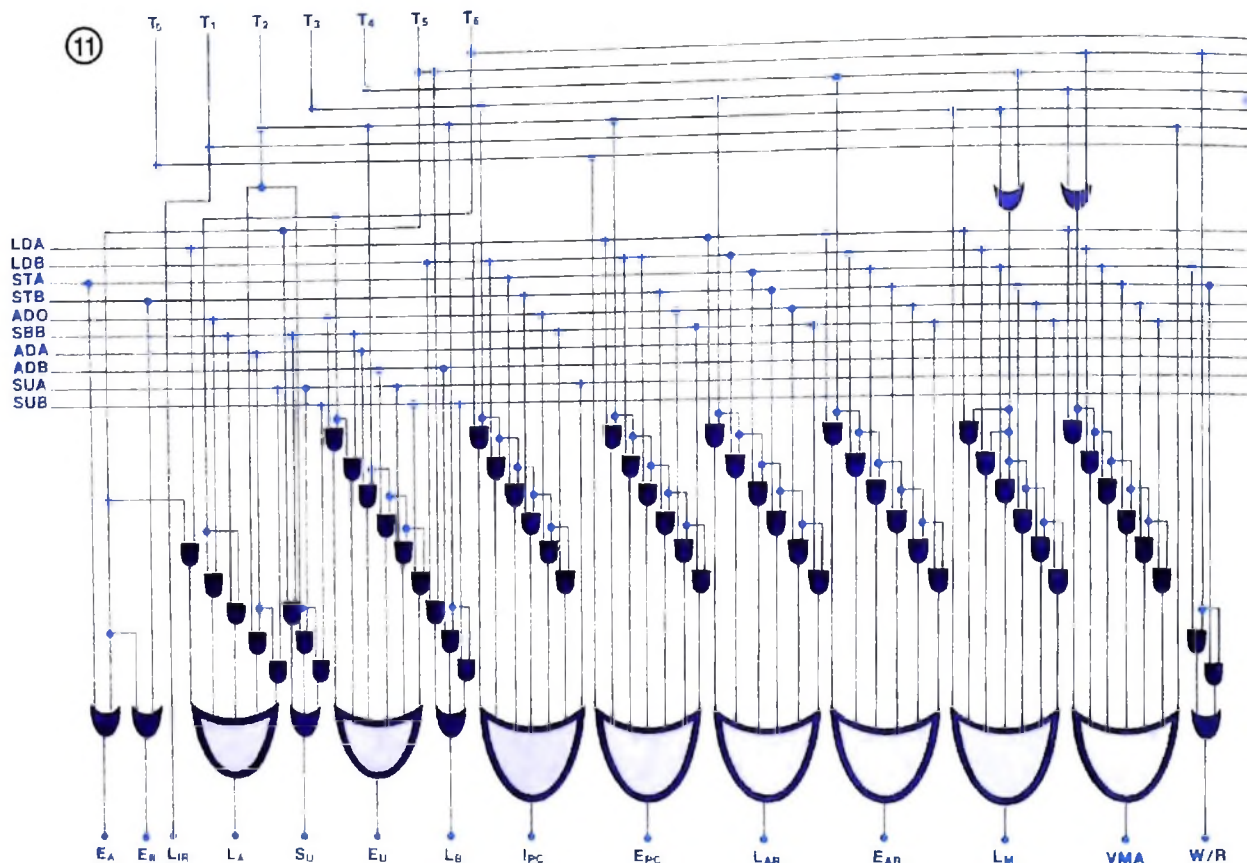
Schema a blocchi del circuito di controllo.



Il circuito decodificatore delle istruzioni.

10	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
	0	0	0	0	0	1	0	1

Il codice dell'istruzione SBB.



Come è fatta la matrice a due entrate del circuito di controllo.

vuol dire che i valori delle linee $I_0 \rightarrow I_7$ sono come in figura 10 e pertanto, se per I_7 (che è uno zero) prendiamo il segnale invertito, quindi un 1 e facciamo lo stesso con I_6, I_5, I_4, I_3 e I_1 , mentre per I_2 e I_0 prendiamo il segnale normale, in questo modo, all'entrata dell'AND corrispondente ci sono soltanto 1 e, quindi, all'uscita avremo un 1; in qualsiasi altro AND non c'è coincidenza di 1 all'entrata e le uscite saranno sempre 0. Questo registro, come possiamo vedere dalla figura 10, ha 11 linee di uscita: 10 entrano nel registro control matrix e una, la linea HLT, entra invece nel registro oscillatore; pertanto, quando questa linea è vera, cioè uguale a 1, va a bloccare l'oscillatore che smette di oscillare e tutto viene bloccato. L'ultimo registro che resta da analizzare è il control matrix (figura 11). Questo registro è una matrice a due entrate: a sinistra entrano le linee attive delle istruzioni (sappiamo che solo una può essere attiva in un dato istante); da sopra, inve-

ce, entrano le linee delle fasi di esecuzione che, in poche parole, marcano il tempo, in basso esce il BUS di Controllo con tutte le sue linee che cambiano adeguatamente ad ogni CLK. Riprendiamo, per esempio, la stessa istruzione SBB XX, però incominciamo dall'inizio, cioè dalla fase T_0 per vedere anche il ciclo di accesso (fetch). Se eseguiamo i comandi collegati alle varie fasi avremo:
 $T_0 = E_{PC}, L_M$
 $T_1 = L_{IR}, VMA$
 Questi infatti sono i comandi che si ripetono ogni volta che vogliamo prelevare un'istruzione dalla memoria. A questo punto il circuito IR decodificherà l'istruzione e attiverà la linea corrispondente. Abbiamo scelto come esempio l'istruzione SBB XX; provate a verificare voi stessi se alle varie fasi corrispondano i comandi appropriati per realizzare tale istruzione.

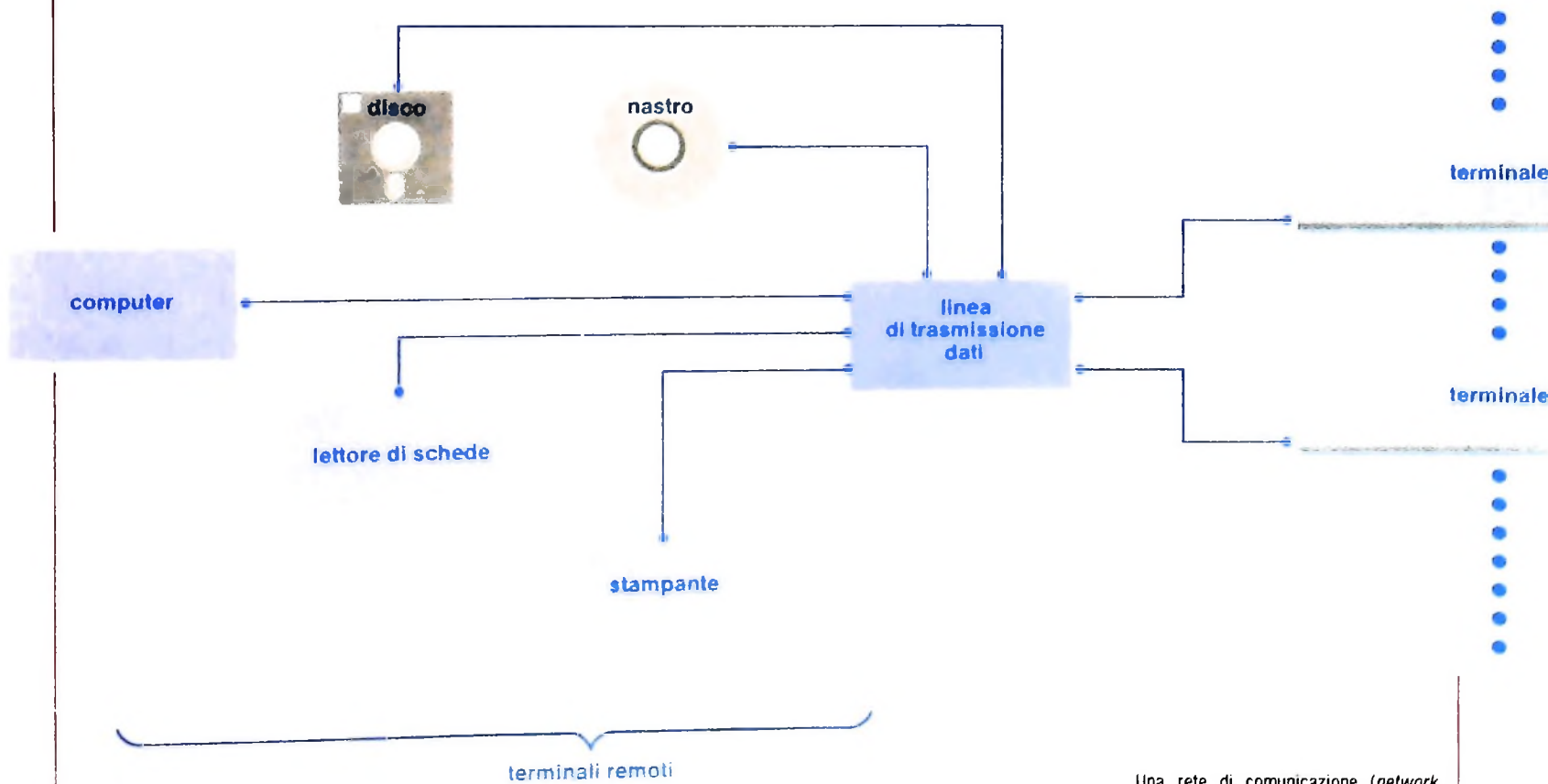
LA TRASMISSIONE DATI

Come vengono trasferite informazioni tra elaboratore e terminali o tra terminali.



Con il termine "trasmissione dati" si definisce il trasferimento di informazioni tramite un *mezzo trasmissivo*, da un punto denominato *origine*, che genera l'informazione, ad un altro chiamato *destinatario*. Nei sistemi di comunicazione, queste informazioni prendono il nome di *dati* o *messaggi*, mentre il mezzo trasmissivo viene identificato dal termine *rete di comunicazione* (*network*, con vocabolo inglese) (si vedano le il-

lustrazioni in questa pagina) ed è costituito da un elaboratore con uno o più terminali collegati con *linee di trasmissione*. Queste linee permettono il flusso di messaggi tra l'elaboratore ed i terminali ad esso collegati o tra i terminali stessi. I terminali, poi, possono essere di diverso tipo: telescriventi (TTY), stampanti, lettori di schede, terminali video (CRT), elaboratori, dischi magnetici, nastri magnetici.



Una rete di comunicazione (*network*, con termine inglese) è costituita da un elaboratore con uno o più terminali collegati con linee di trasmissione dati.

Codici di trasmissione e strutture

Utilizzando un sistema di trasmissione di solito si desidera trasmettere un flusso di caratteri che possa essere ricostruito in un altro punto in una forma comprensibile all'operatore o comunque in forma tale da poter essere manipolata; tali caratteri sono, come in un sistema computerizzato, rappresentati da gruppi di bit.

I vari gruppi di bit che rappresentano i caratteri formanti l'alfabeto di un dato sistema di trasmissione dati, sono chiamati *codici*.

Un byte è un gruppo di bit consecutivi che sono trattati come una unità; un byte è composto da 8 bit e normalmente rappresenta un carattere.

Nella trasmissione dati vengono utilizzati codici che necessitano di 5, 6, 7, 8, 10 o 11 bit per rappresentare un carattere e la differenza sorge dal fatto che alcuni codici prevedono un differente numero di bit per formare un efficiente sistema di rivelazione degli errori.

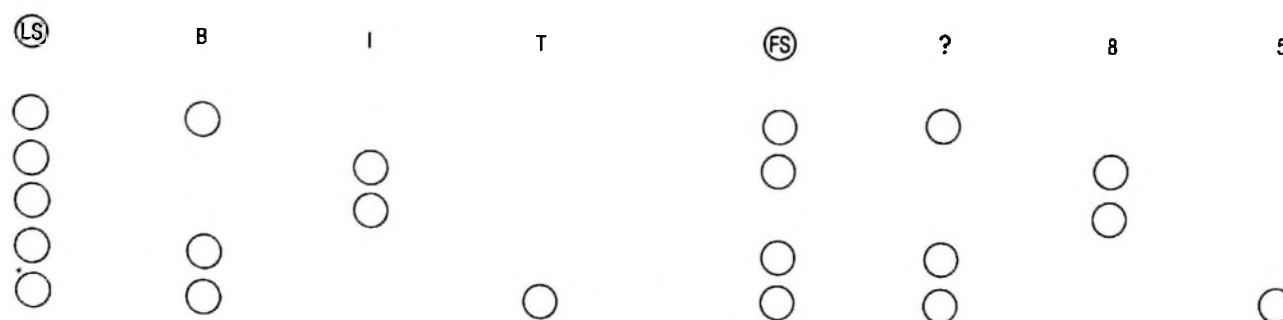
Per esempio, un codice di un solo bit può rappresentare ed esprimere due soli caratteri — per esempio la lettera A e la B — associando allo stato 0 la lettera A e allo stato 1 la B.

Configurazione del codice USASCII (ASCII)

				b ₇ → 0 0 0 0 1 1 1 1								
				b ₆ → 0 0 1 1 0 0 1 1								
				b ₅ → 0 1 0 1 0 1 0 1								
				b ₄ b ₃ b ₂ b ₁								
				↓ ↓ ↓ ↓								
				COL	0	1	2	3	4	5	6	7
				RIGA	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	'	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	9	HT	EM)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	;
1	1	0	1	13	CR	GS	-	=	M]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Configurazione del codice USASCII, un codice a 8 bit, di cui sette sono significativi, mentre l'ottavo è un bit per il controllo di parità. La tabella è impostata in modo che le colonne diano i tre bit più significativi e le righe i quattro

meno significativi: il codice corrispondente al carattere posto in una certa casella ha quindi come primi tre bit quelli della colonna in cui si trova quella casella e come bit successivi i quattro della riga su cui si trova la casella.



Un codice Baudot su banda perforata: si possono vedere i caratteri "estensione del codice" LS e FS che svolgono una funzione analoga a quella dei tasti per le minuscole e le maiuscole di una comune macchina per scrivere. Il codice Baudot, che è un codice a 5 bit, è usato soprattutto sulle reti telex.

Un codice a due bit ci permetterebbe di poter contare su un alfabeto composto da quattro caratteri associando, per esempio:

00 con la lettera A
01 con la lettera B
10 con la lettera C
11 con la lettera D

Continuando potremmo formare un codice a 3 bit composto di otto caratteri e così via, seguendo la oramai nota regola delle potenze di 2. Un codice è una convenzione e nulla, in linea di principio, vieta di scegliere una convenzione piuttosto che un'altra, ma, per comunicare, due unità (quali che siano) devono avere un codice comune. Per questo sono stati standardizzati internazionalmente dei codici di trasmissione dati.

1) Il codice ASCII

Lo United States of America Standard Code for Information Interchange (USASCII o più comunemente ASCII) è un codice a 8 bit, uno dei quali per il controllo di parità, che permette di rappresentare 128 caratteri (infatti i sette bit significativi consentono $2^7 = 128$ configurazioni diverse). Il codice ASCII è uno dei codici più diffusi per la trasmissione dati ed è il risultato di 5 anni di lavoro dal 1963 (data in cui apparve il primo codice denominato ASCII63) al 1968.

I sette livelli di informazione permettono di codificare un intero insieme di caratteri maiuscoli e minuscoli, più altri caratteri grafici e di controllo.

Nella pagina a fronte è riportato lo schema tramite il quale viene generalmente rappresentato il codice ASCII, formato da 8 colonne, numerate da 0 a 7, con in testa la configurazione dei tre bit più significativi tra i sette costituenti il carattere e 16 righe numerate da 0 a 15 con a fianco la configurazione binaria dei 4 bit meno significativi.

Per identificare un qualsiasi carattere sulla tabella di conversione possiamo utilizzare due vie: la prima tramite la sua rappresentazione binaria (ad esempio, il carattere C corri-

sponde alla sequenza binaria 1000011); la seconda tramite intersezione dei numeri di riga e colonna (ad esempio, lo stesso carattere C è rappresentato dalla coppia 4/3).

2) Il codice BAUDOT

È un codice a 5 bit, che risale alla fine dell'Ottocento e trae il nome da un ingegnere francese. È utilizzato in prevalenza sulle reti internazionali — e private — telex (assume anche il nome di *codice telex*).

Tramite il codice Baudot, si possono formare fino a $2^5 = 32$ configurazioni diverse e ciò non è sufficiente per rappresentare l'intero insieme dei caratteri alfanumerici: per raggiungere lo scopo il codice Baudot utilizza due caratteri (chiamati estensione del codice) detti LETTER SHIFT (LS noto anche come "tasto lettera") e FIGURE SHIFT (FS noto anche come "tasto cifre") e sono praticamente simili ai tasti "minuscole" e "maiuscole" di una normale macchina per scrivere.

Quando uno di questi caratteri è utilizzato, esso viene applicato a tutti i caratteri seguenti fino ad una nuova utilizzazione di un LS o di un FS. In pratica, si ha l'effetto di un sesto bit in ciascuno dei rimanenti 30 gruppi di bit.

Nella figura in alto, viene mostrata come esempio una tipica banda perforata per telex, ove appare evidente come l'utilizzo di un carattere LS o FS causi una differente interpretazione degli stessi gruppi di 5 bit; la figura identifica come 1 i "buchi" e come 0 i "non buchi". L'aggiunta dei due caratteri speciali provoca una estensione del codice da 32 a 62 elementi (in realtà si ottengono 57 differenti caratteri, dato che LS, FS, RITORNO CARRELLO, SPAZIO, SALTO RIGA hanno un solo valore qualunque sia il carattere SHIFT utilizzato). Nella pagina seguente è data la configurazione del codice Baudot.

3) Il codice BCD

Il codice BCD (Binary Coded Decimal) è un codice a 6 bit, impiegato come codice interno da alcuni elaboratori. I sei livelli forniscono 64 possibili combinazioni e nella trasmissio-

Configurazione del codice Baudot						
MINUSCOLE (LS)	MAIUSCOLE (FS)	1	2	3	4	5
A	—	•	•			
B	?	•			•	•
C	:		•	•	•	
D	\$	•			•	
E	3	•				
F	!	•		•	•	
G	&		•		•	•
H	£			•		•
I	8		•	•		
J	.	•	•		•	
K	(•	•	•	•	
L)		•			•
M	.			•	•	•
N	.			•	•	
O	9				•	•
P	0		•	•		•
Q	1	•	•	•		•
R	4		•		•	
S	Bell	•		•		
T	5					•
U	7	•	•	•		
V	:		•	•	•	•
W	2	•	•			•
X	/	•		•	•	•
Y	6	•		•		•
Z	"	•				•
LS (tasto lettere)		•	•	•	•	•
FS (tasto cifre)		•	•		•	•
SPAZIO				•		
ritorno carrello (CR)					•	
LF (salto riga)			•			
blank						
• Rappresenta un 1 Blank rappresenta uno zero						

Configurazione del codice Baudot. Con cinque bit consente la rappresentazione in tutto di 57 caratteri diversi (i caratteri LS, FS, RITORNO CARRELLO, SPAZIO e SALTO RIGA hanno un solo valore, indipendentemente dal carattere SHIFT utilizzato).

ne dati a volte viene utilizzato come codice a 7 bit aggiungendo un settimo bit, di parità.

4) Il codice EBCDIC

L'Extended Binary Coded Decimal Interchange Code è un codice IBM Sistema 360/370 ed ha una combinazione di 256 caratteri (in questo caso, a differenza del codice ASCII dove l'ottavo bit è di parità, tutti e otto i bit sono utilizzati per la rappresentazione di caratteri).

I caratteri di controllo

Per quanto riguarda il codice ASCII, l'alfabeto, o *character set* comprende anche 32 "caratteri di controllo" che sono suddivisi in differenti categorie: separatori di informazione; controlli di formato, controlli di dispositivi.

a) Separatori di informazione

US (Unit Separator), che delimita l'unità di informazione;
RS (Record Separator), che delimita il record d'informazione qualora si abbiano record costituiti da unità di informazione;
GS (Group Separator), utilizzato per delimitare un certo gruppo di dati formato da record;
FS (Files Separator) impiegato per separare un certo numero di gruppi di dati.

b) Controlli di formato

BS (Back Space), fa arretrare la testina di stampa (o il cursore di un videoterminale) verso sinistra di uno spazio;
HT (Horizontal Tabulation), permette l'avanzamento del cursore (o testina di stampa) in direzione orizzontale;
LF (Line Feed), fa avanzare il cursore (o testina di stampa) sulla successiva riga mantenendo la stessa posizione rispetto al margine sinistro;
VT (Vertical Tabulation), sposta la testina di stampa (o il cursore) di un numero predeterminato di righe mantenendo inalterata la posizione rispetto ai margini;
FF (Form Feed), esegue la stessa funzione di VT ma sposta il cursore (o testina) su di un altro foglio o pagina video;
CR (Carriage Return), sposta il cursore (o testina) alla prima posizione della stessa riga.

c) Controlli dei dispositivi

Questi codici sono utilizzati per controllare i dispositivi ausiliari di un terminale: ad esempio, con DC1 si può comandare l'accensione di un registratore collegato al terminale (per registrare una comunicazione), con DC2 lo si può spegnere, con DC3 si può inviare su una stampante collegata a un videoterminale il contenuto dello schermo per permetterne la stampa, con DC4 si può effettuare la disabilitazione della tastiera di un terminale per impedire l'invio di dati.

Lezione 22

I tipi strutturati: il costruttore di tipo

Abbiamo visto fin qui che ad ogni variabile corrisponde in ogni istante dell'esecuzione, un valore. Così una variabile intera assume, volta per volta, un valore intero, una variabile stringa assume un valore che corrisponde alla sequenza di caratteri assegnatale e così via. Diciamo che tali variabili sono di tipo *semplice*, col che intendiamo appunto che possono assumere un solo valore alla volta.

Diverso è il caso degli array che costituiscono una sola variabile caratterizzata da più valori. Un array infatti è composto di più elementi ciascuno dei quali assume un proprio valore. Poiché i dati raccolti in un array sono logicamente correlati, consideriamo di trattare con un'unica variabile (a cui infatti attribuiamo un unico nome), che è però composta di un insieme di informazioni. Parliamo allora di variabili di *tipo strutturato*. Se proviamo a dare una definizione di array diremo che è un insieme di componenti tutti dello stesso tipo a cui facciamo riferimento tramite un indice. Quando parliamo, cioè, di una struttura, la descriviamo specificando le caratteristiche dei componenti, il modo con cui sono aggregati e il modo con cui vi facciamo riferimento. In sintesi una struttura è definita da:

- un costruttore;
- i componenti.

Per costruttore intendiamo l'insieme di regole che specificano:

- le modalità di aggregazione dei componenti;
- le modalità di accesso agli stessi;
- le relative modalità di riferimento.

L'array è un esempio di costruttore di tipo, con le seguenti caratteristiche:

- regola di aggregazione: un array è un insieme finito di componenti omogenei, cioè dello stesso tipo;
- modalità di accesso: si può fare riferimento ai componenti in qualunque ordine;
- modalità di riferimento: il riferimento ai componenti è effettuato con un indice.

I costruttori di tipo in Pascal: il costruttore array

Nei linguaggi evoluti quali il PASCAL i costruttori di tipo consentono di dichiarare al compilatore variabili strutturate come nell'esempio seguente:

```
VAR LISTINO: ARRAY [1..10] OF 500..10000;
```

che chiede al compilatore di allocare una tabella di nome LISTINO di dieci elementi ciascuno dei quali può assumere i valori da 500 a 10000; ma possono essere usati anche per definire nuovi tipi. Così:

```
TYPE TABELLA = ARRAY [1..10] OF 500..10000;
```

fornisce al compilatore la descrizione di un *nuovo tipo*, senza peraltro provocare un'allocazione di spazio in memoria. In questo modo non si è fatto altro che attribuire un nome a una descrizione dell'insieme di attributi di una variabile. Pertanto, qualora il tipo TABELLA sia stato precedentemente definito, la dichiarazione della variabile LISTINO si ridurrà alla seguente:

VAR LISTINO: TABELLA;

che ha l'effetto di richiamare al compilatore la descrizione introdotta dalla parola chiave TYPE. In questo modo il tipo TABELLA potrà essere ripetutamente richiamato in differenti dichiarazioni di variabili essendo a tutti gli effetti noto al compilatore come i tipi messi a disposizione dallo stesso (detti "tipi predefiniti").

I costruttori di tipo in Pascal: il costruttore record

Oltre al costruttore ARRAY sopra presentato, il PASCAL mette a disposizione alcuni altri costruttori particolarmente adatti a risolvere problemi tipici della programmazione. Ne vediamo, in questa lezione e nelle successive, una breve descrizione. Abbiamo già visto esempi di programmi BASIC che elaborano una data, che è sempre stata trattata come composta da tre variabili rispettivamente per il giorno, il mese e l'anno. Queste tre informazioni sono però logicamente correlate anche se corrispondono a tre valori e quindi a tre componenti differenti. In PASCAL possiamo dichiarare una sola variabile nel modo seguente:

```
VAR DATA: RECORD
    GIORNO: 1..31;
    MESE: 1..12;
    ANNO: 0..90
END;
```

In tal modo, abbiamo richiesto al compilatore di mettere a disposizione una variabile di nome DATA costituita di tre componenti: un componente di nome GIORNO che può assumere i valori da 1 a 31; un componente di nome MESE che può assumere i valori da 1 a 12; un componente di nome ANNO che può assumere i valori da 0 a 90 (supposto che date le caratteristiche del problema interessino gli anni compresi tra il 1900 e il 1990). La variabile così dichiarata può essere trattata (letta o modificata) nella sua globalità, o componente per componente. In questo caso, però, per non dimenticare che stiamo parlando, per esempio, del componente MESE del record DATA, useremo la seguente dicitura:

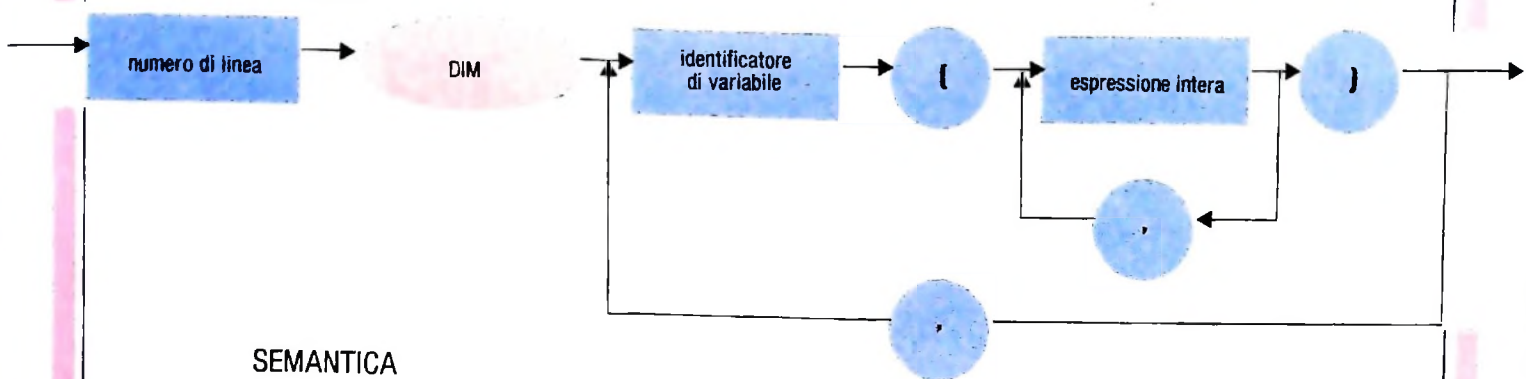
DATA.MESE

Nello stesso programma infatti potrebbero essere usate altre variabili semplici di nome MESE o altre variabili strutturate dotate di un componente di questo nome. Nei problemi risolti con calcolatore, sono molto frequenti situazioni in cui ci si trova di fronte a informazioni che, pur avendo una propria individualità, sono tra loro correlate. Così è per esempio il caso delle informazioni anagrafiche che pur avendo ciascuna caratteristiche proprie sono correlate dall'appartenenza a un unico individuo. In PASCAL si descrivono così:

```
TYPE ANAGRAFICA: RECORD
    NOME, COGNOME: STRINGA;
    ETÀ: 0..100;
    SESSO: (F, M);
    CONIUGATO: BOOLEAN;
    TITOLOSTUDIO: (ELEMENTARE, MEDIA, SUPERIORE, LAUREATO)
END;
```

Istruzione DIM

SINTASSI



SEMANTICA

L'istruzione DIM predispone una o più variabili "strutturate", costituite cioè da aggregati di variabili semplici. Le variabili così allocate hanno un numero di elementi che dipende dagli indici dichiarati. Tali elementi vengono individuati specificando il valore corrispondente di ciascuno degli indici per ciascuna delle dimensioni dell'array.

L'istruzione:

```
10 DIM A(10)
```

indica una variabile di 11 elementi, con l'indice che varia da 0 a 10.

L'istruzione:

```
10 DIM B(6,7)
```

alloca una variabile di 56 elementi, identificabili con una coppia di indici, che assumono valori rispettivamente da 0 a 6 e da 0 a 7.

```
10 DIM C(3,3,3)
```

È l'immagine di un "casellario" "cubico" di lato 4.

Non ci sono limiti al numero di indici né al numero di elementi di un array a eccezione della quantità di memoria disponibile. Qualora non sia disponibile la memoria sufficiente per l'allocatione dell'array richiesto viene fornita un'indicazione di errore di tipo OM, che indica proprio che si è verificata una trasgressione dei limiti di memoria (overflow).

L'istruzione DIM azzerà le variabili numeriche e pone una stringa vuota in quelle alfanumeriche.

Se un array viene usato prima che venga fornita una corrispondente istruzione DIM, le dimensioni di ciascun indice sono automaticamente assunte uguali a 10.

Se in un'espressione il valore di un indice supera la dimensione dichiarata con l'istruzione DIM, viene fornita un'indicazione d'errore di tipo BS.

Si noti che, essendo il linguaggio BASIC su M10 interpretato (e cioè memorizzato come testo e quindi eseguito istruzione per istruzione, senza che ne venga fatta precedentemente una manipolazione complessiva per una traduzione nel linguaggio della macchina), l'istruzione DIM può essere eseguita con valori determinati durante l'esecuzione del programma come nell'esempio seguente:

```
10 INPUT N
20 DIM A(N)
```

che predispone un insieme di $N+1$ elementi, congruentemente con il valore letto durante l'esecuzione.

che descrive un insieme di informazioni autonome ma tutte riferite a una stessa persona.

Riassumiamo dunque le caratteristiche del costruttore **RECORD**:

— modalità di aggregazione: una struttura record è costituita di un insieme finito di elementi disomogenei, cioè non necessariamente dello stesso tipo, detti **CAMPI**, a ciascuno dei quali è attribuito un nome;

— l'accesso a ciascun campo può avvenire in un ordine qualsiasi, non è cioè vincolante l'ordine con cui sono dichiarati: pertanto nell'esempio precedente possiamo fare direttamente riferimento al campo **TITOLOSTUDIO** senza prendere in considerazione i precedenti;

— modalità di accesso ai componenti: il riferimento ai componenti è fatto tramite il nome della variabile seguito da "." e dal nome del campo. Così, se il programma usa due variabili di tipo **ANAGRAFICA** così dichiarate:

```
VAR CITTADINITALIANI, CITTADINISTRANIERI: ANAGRAFICA;
```

per accedere ai corrispondenti campi **NOME** diremo rispettivamente:

```
CITTADINITALIANI.NOME
```

e

```
CITTADINISTRANIERI.NOME
```

I record in Basic

Il concetto di costruttore di tipo corrisponde all'esigenza di modellare le strutture di dati in modo più possibile conforme alle caratteristiche del problema da trattare. In quest'ottica i costruttori di tipo sono "mattoni" che consentono di disegnare strutture anche molto complesse, come vedremo in seguito, e sono il risultato di una maturità nella definizione di linguaggi di programmazione ancora lontana quando il **BASIC** è nato. Abbiamo così visto che è possibile usare array in **BASIC**, ma per quanto riguarda la struttura record il linguaggio non ci mette a disposizione alcuno strumento. Nel caso pertanto di informazioni correlate non ci resta che utilizzare variabili distinte, come già abbiamo visto nei programmi che elaborano una data.

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- Il concetto di costruttore di tipo e di struttura di dati;
- la differenza tra dichiarazione di variabile e definizione di tipo;
- i costruttori **ARRAY** e **RECORD** in **PASCAL**;
- la realizzazione **BASIC** di strutture record.

ARMONIA

Come si organizzano sequenze parallele di eventi musicali.

Abbiamo già visto come i fenomeni musicali siano costituiti da sequenze di eventi musicali nel tempo e dalla *sovrapposizione* di sequenze, ovvero dal *parallelismo* (polifonia); le sequenze possono essere costituite da eventi musicali astratti (temi, melodie, movimenti ecc.) o anche da eventi musicali più di dettaglio come sequenze di parametri acustici (frequenze, ampiezze, campioni numerici ecc.).

Se guardiamo una partitura, osserviamo sequenze di note sovrapposte (o sullo stesso rigo o su rigi diversi). Le relazioni tra gli eventi musicali di una *parte* con quelli di un'altra non sono descritte esplicitamente; per conoscere la struttura di queste relazioni possiamo:

a) avere una cultura musicale sufficiente per dedurre dal testo la sua struttura;



Disegni di Valentine Hugo ispirati alla "Danza sacra dell'eterna" da *La sagra della primavera* (Parigi, collezione André Meyer).

b) conoscere una forma di descrizione più astratta (*strutturale*, appunto).

La descrizione dei fenomeni musicali per un elaboratore è bene sia data in una forma che preveda l'esplicitazione delle relazioni tra gli eventi costitutivi dei diversi processi.

Vedremo qui alcune strutture armoniche classiche (la *cadenza*, il *basso numerato*, il *contrappunto*) e discuteremo un particolare problema di sovrapposizione di ritmi.

Cadenza

La conclusione di una frase musicale viene generalmente realizzata mediante una concatenazione degli accordi più importanti della tonalità: la tonica (I), la dominante (V) e la sottodominante (IV).

Queste sequenze di accordi sono chiamate *cadenze* e, nella

musica, hanno una funzione molto simile a quella della *punteggiatura* nella scrittura.

Sono quattro i tipi di cadenza: la cadenza *perfetta*, costituita dalla sequenza della triade sulla dominante e della tonica (V - I); la cadenza *plagale* (molto usata nella musica liturgica), costituita dalla sequenza sottodominante-tonica (IV - I); la cadenza *evitata*, che è il collegamento dalla dominante alla sopradominante, invece che alla tonica (V - VI); ha una funzione più sospensiva che conclusiva; la cadenza *sospesa*, costituita dalla sequenza di un accordo qualsiasi e della triade sulla dominante; in genere, o II-V o IV-V o VI-V o I-V.

Proseguendo nella metafora della punteggiatura, la cadenza perfetta e quella plagale sono come il *punto* (anche se con sfumature differenti), la cadenza evitata è come una *virgola* e quella sospesa è come un *punto e virgola*.

Una chiusura molto usata è la sequenza tonica-sottodominante-dominante-tonica (I-IV-V-I).

Strutture ritmiche

Proviamo ora a considerare un esercizio in cui sovrapponiamo frammenti ritmici diseguali in durata a una sequenza ritmica di base omogenea. I frammenti ritmici che consideriamo sono tratti dalla partitura della *Sagra della primavera* di Igor Stravinsky (ai numeri 91 e 92, inizio dei *Cercles Mysterieux des Adolescentes*).

Qui sotto sono riportati i sei frammenti ritmici in esame: A è costituito da una struttura di quattro quarti seguita da due quarti (4 + 2), B è come A con l'aggiunta di un quarto nella seconda parte (4 + 3), C è una struttura di quattro quarti (4 = 2 + 2), D di cinque quarti

(5 = 2 + 3), E di sei quarti (6 = 3 + 3); la sequenza di questi frammenti dura in totale 28 quarti ed è sovrapposta alla ripetizione per 14 volte della struttura F costituita da quattro ottavi (2); la struttura di base che consideriamo è quindi:

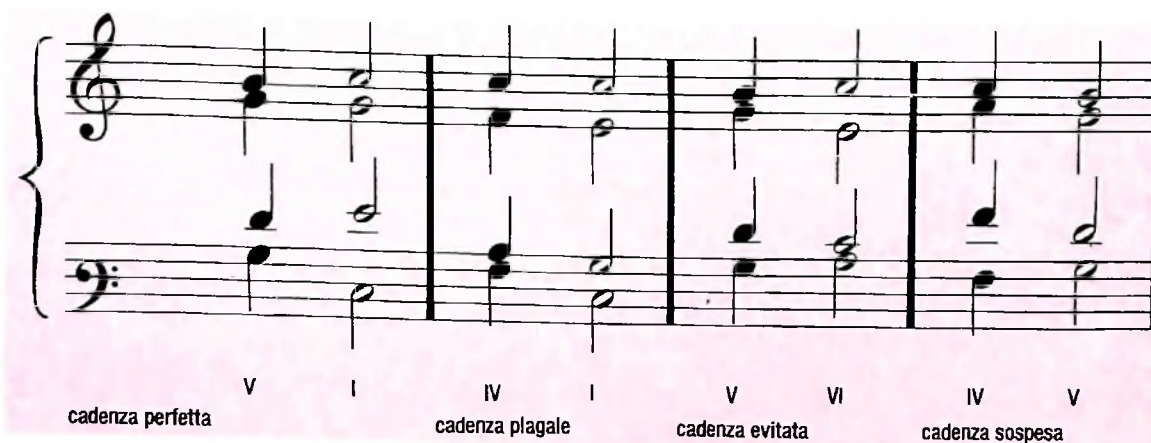
prima voce: A B C D E

seconda voce: F F F F F F F F F F F F F F F F

Proviamo allora a scrivere un programma per M10 che permetta di realizzare strutture musicali basate su frammenti melodici anche differenti da quello della partitura che abbiamo qui analizzato, ma con



Strutture ritmiche estratte dalla partitura della "Sagra della primavera" di Igor Stravinsky; le strutture A, B, C, D, E, si succedono nel tempo sovrapposte alla ripetizione della struttura F (dai "Cercles mysterieux des adolescentes", riferimento 91-92 della partitura originale).



La conclusione di una frase musicale viene generalmente realizzata mediante la concatenazione degli accordi più importanti della tonalità. Queste sequenze

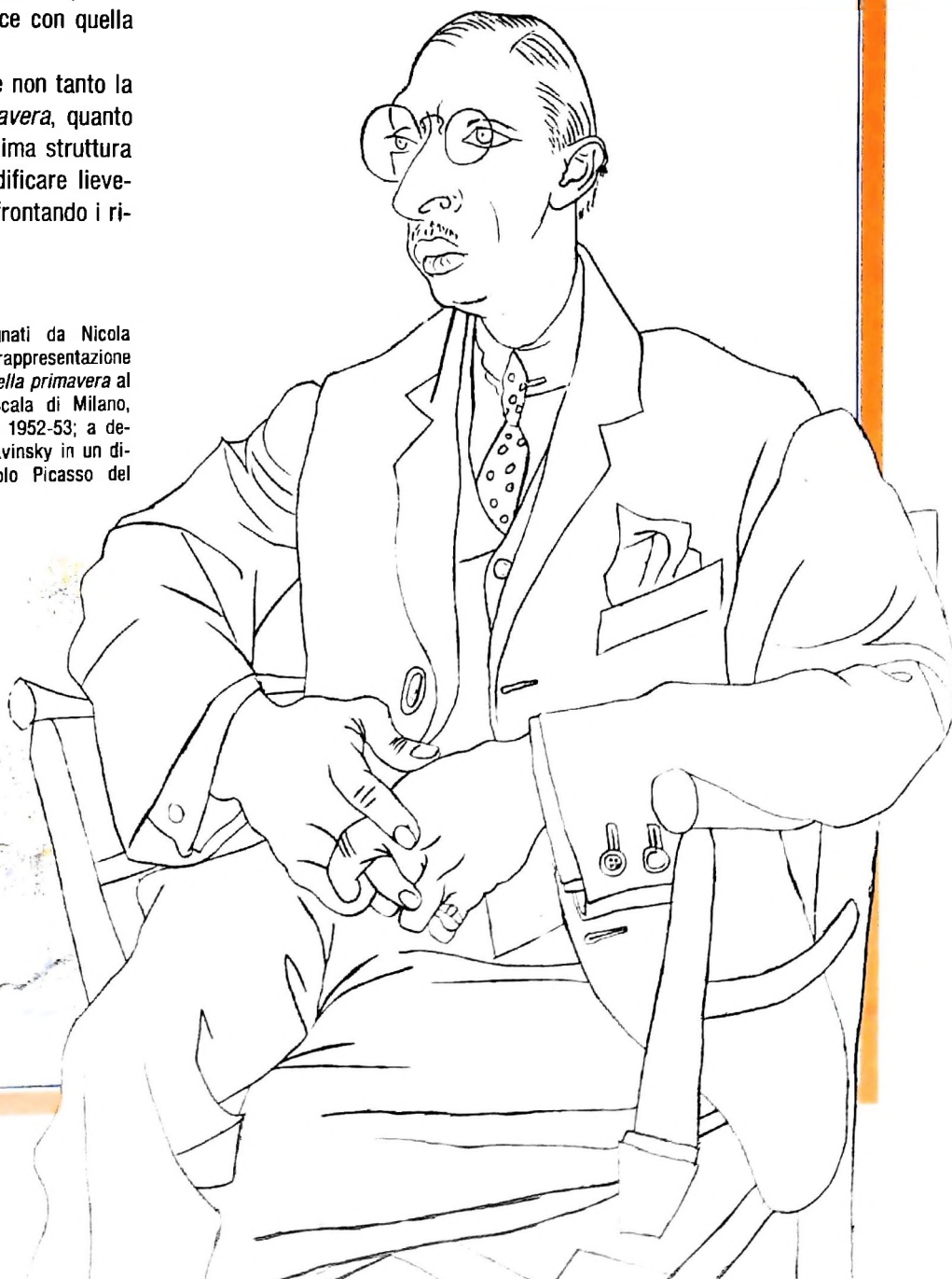
di accordi sono chiamate cadenze. Sono qui riportati i quattro tipi tradizionali di cadenza: la perfetta, la plagale, l'evitata e la sospesa.

la medesima struttura ritmica; poiché M10 non permette la polifonia, invece di sovrapporre la struttura della prima voce con quella della seconda, le porremo in sequenza, una dopo l'altra.

Avremo quindi un programma che permette di eseguire non tanto la struttura originale considerata della *Sagra della primavera*, quanto tante altre strutture musicali, tutte basate sulla medesima struttura ritmica complessa; un esercizio interessante sarà modificare lievemente i frammenti melodici delle strutture ritmiche confrontando i risultati con l'originale di Stravinsky.



Figurini disegnati da Nicola Benois per la rappresentazione de *La sagra della primavera* al Teatro alla Scala di Milano, nella stagione 1952-53; a destra, Igor Stravinsky in un disegno di Pablo Picasso del 1920.



Basso numerato

Per basso numerato (o continuo) si intende una forma di notazione abbreviata molto in uso durante l'epoca barocca per descrivere l'armonia base (di *accompagnamento*) di un certo brano musicale.

La scrittura del basso numerato è realizzata mediante numeri che indicano la struttura dell'accordo da costruire sulla nota indicata esplicitamente nel pentagramma.

The image shows two musical staves. The top staff, labeled 'realizzazione degli accordi', shows a sequence of chords in a treble clef. The bottom staff, labeled 'basso numerato', shows a sequence of notes in a bass clef with numbers below them: 6, 4, 6, 6, 5, 4, 3, 2, 9, and 'ecc.'. The numbers indicate intervals from the bass note.

Esempi di basso numerato: i numeri sono usati per stabilire gli intervalli fra le altezze dell'accordo, calcolati a partire dalla nota del basso; generalmente vengono omesse le indicazioni relative agli intervalli più ovvi, cioè le terze e le quinte.

Contrappunto

Le forme di struttura musicale (da un punto di vista armonico) più usate sono:

- a) *omofonia*: sequenza melodica senza accompagnamento;
- b) *monodia*: sequenza melodica con accompagnamento;
- c) *polifonia*: più melodie che si sviluppano contemporaneamente (in relazione, o non, l'una con le altre).

La musica di quest'ultimo genere viene detta anche contrappuntistica. Per realizzare una composizione contrappuntistica possiamo basarci su tre principi fondamentali:

- 1) produrre un interesse melodico (mediante l'affermazione di frammenti melodici o motivici o mediante imitazione);
- 2) produrre un interesse ritmico (mediante una individualità delle strutture ritmiche delle differenti *parti* polifoniche);
- 3) costruire la parte più grave con funzione di basso fondamentale. Per imitazione intendiamo una trasformazione di un frammento melodico che ne mantenga la riconoscibilità (il caso più banale è la trasposizione).

Per esempio, consideriamo la ben nota canzoncina *Fra' Martino campanaro*. Si tratta in realtà di un *canone*, una delle forme contrappuntistiche più rigorose e di cui quindi è più facile fornire una descrizione formale adatta per comunicare con un elaboratore. In un canone, la prima voce espone un tema (*antecedente*), la seconda lo ripete (uguale o trasformato per trasposizione o inversione speculare o altro) con uno spostamento temporale rispetto alla prima voce; la seconda voce è detta *conseguente*; la seconda diventa poi antecedente della

terza che è invece conseguente della seconda, e così via. Possiamo realizzare un esempio di programmazione contrappuntistica di *Fra' Martino* seguendo lo schema seguente:

procedura *Fra' Martino*;

begin

P1; P1;

Fra' Martino;

P2; P2;

P3; P3;

P4; P4

end

Nell'ipotesi di disporre di un sistema multiprocessore, questo modello di descrizione produce una struttura *ricorsiva* di canone infinito come segue:

```

P1 P1 P2 P2 P3 P3 P4 P4
    P1 P1 P2 P2 P3 P3 P4 P4
        P1 P1 P2 P2 P3 P3 P4 P4
            P1 P1 P2 P2 P3 P3 P4 P4
                P1 P1 ...
    
```

The image shows two musical staves. The first staff has four measures of music, with the first two measures labeled 'P1' and the last two labeled 'P2'. The second staff has four measures of music, with the first two labeled 'P3' and the last two labeled 'P4'. This illustrates the recursive structure of the canon.

Il testo di *Fra' Martino*: le sequenze di ogni battuta sono codificate come P1, P2, P3, P4, a due a due.

IL COMPUTER IN BANCA

L'automazione dei servizi bancari, il self-service di agenzia e lo home-banking.

Nella parte precedente sono state analizzate le motivazioni che hanno condotto all'introduzione di tecnologie informatiche e telematiche nell'ambiente bancario, che risulta oggi uno tra i settori maggiormente automatizzati dell'apparato economico produttivo. Se l'applicazione di quelle tecnologie è stata finora principalmente orientata a risolvere problemi interni al sistema banca, la tendenza attuale è invece quella verso lo sviluppo di strumenti orientati a modificare radicalmente il rapporto tradizionale con l'utenza. La base di questo nuovo orientamento sta nell'applicazione intensiva delle tecniche di trasferimento elettronico di fondi e nella progressiva verifica della loro affidabilità.

L'automazione dei servizi bancari si sta quindi indirizzando principalmente verso il cosiddetto self-service di agenzia e verso lo home-banking.

Prossimamente su quegli schermi

I servizi di self-service di agenzia vengono resi possibili dall'installazione di terminali in aree appositamente allestite all'interno delle agenzie stesse. Lo scopo è di mettere in grado l'utente di poter eseguire autonomamente alcune operazioni che tradizionalmente vengono eseguite da un operatore umano con cui l'utente deve interagire. I programmi che permettono di utilizzare questi terminali hanno la caratteristica di presentare le operazioni possibili in una forma "amichevole" e semplice all'utente, e quindi si avvalgono dei risultati della ricerca nel campo delle interfacce uomo/macchina: linguaggio iconico, realizzato attraverso immagini in scala ridotta (le *icone*), di oggetti con cui l'utente deve interagire; programmi a menù; comprensione del linguaggio naturale.

I servizi di home-banking si basano invece sull'uso di terminali personali come personal computer collocati presso l'abitazione o l'ufficio dell'utente o come i piccoli elaboratori portatili quali lo M10 Olivetti. Questo servizio si basa sostanzialmente su tre tipi di terminali che si diversificano in termini di costo e operazioni possibili:

— terminali tascabili, a bassissimo costo, che inviano messaggi come segnali acustici, usati per rappresentare i codici relativi alla particolare funzione desiderata, mentre le risposte vengono prodotte tramite sintetizzatori vocali e ricevute direttamente via telefono;

— terminali portatili, a basso costo, dotati di tastiera alfanumerica per l'input e di uno schermo per l'output, eventualmente di ministampanti e di modem per la comunicazione;

— terminali per il Videotel e personal computer, dotati di meccanismi di comunicazione.

L'interazione con il sistema banca diventa così indipendente dalla sua collocazione fisica. Alcune applicazioni sono già state realizzate, come per esempio la possibilità di acquisire informazioni sullo stato attuale del proprio conto corrente ricevendo un messaggio vocale prodotto da un sintetizzatore di voce attivo presso il centro di elaborazione della banca con cui il proprio terminale di casa è collegato (il servizio descritto è oggi in fase di sperimentazione per gli utenti del Banco di Roma). I componenti principali per la realizzazione di questo servizio sono una unità di controllo, presso l'elaboratore centrale, e un trasmettitore, presso l'utente. L'unità di controllo riceve le comunicazioni codificate in arrivo interpretandole e concentrandole verso l'elaboratore, trasformando quindi le risposte di quest'ultimo in risposta vocale utilizzando una sequenza di suoni elementari predefiniti con cui vengono costruiti i messaggi. Il trasmettitore è un piccolo sistema tascabile, con cui l'utente da un qualsiasi apparecchio telefonico invia i segnali acustici, relativi al proprio codice e all'operazione che intende eseguire, alla unità di controllo; da essa l'utente riceve risposta, sempre via telefono.

Crittografia

La crittografia è una tecnica con la quale si effettua una trasformazione di dati allo scopo di renderli inutilizzabili a coloro che intendono accedervi senza autorizzazione. Un semplicissimo esempio consisterebbe nel trasformare un messaggio in modo che le vocali presenti si scambino tra loro circolarmente: le "u" in "a", le "a" in "e", le "e" in "i" e così via. Da "crittografia" si otterrebbe "crottugrefoe".

I problemi che vengono risolti sono quelli relativi a:

- riservatezza: si rende impossibile l'estrazione di informazioni da un generico canale di comunicazione;
- autenticazione: si rende impossibile l'inserimento di false informazioni su un generico canale di comunicazione.

Una procedura di crittografia in generale si sviluppa così:

il testo originario in chiaro (senza modifiche) passa attraverso il filtro crittografico, ovvero una funzione di trasformazione invertibile, che produce il testo in codice. Il ricevente autorizzato trasforma il testo codificato applicando a questo la trasformazione inversa. Il testo in codice è utilizzabile soltanto conoscendo la funzione di trasformazione.

Aspetti giuridici, normativi e di sicurezza

Uno dei problemi più rilevanti che vengono posti dall'automazione dei servizi bancari è la relativa insufficienza dell'ordinamento giuridico attuale a regolamentare sia in termini normativi che penali questa nuova situazione. Le attuali "regole del gioco" sono basate su una serie di criteri chiave utilizzati per definire con precisione i confini del campo dei diritti e dei doveri riferiti al caso specifico del rapporto tra banca e cliente; tali regole non hanno più un riscontro così definito nella nuova realtà determinatasi in seguito all'applicazione di tecnologie elettroniche.

Per l'ordinamento giuridico vigente, uno degli aspetti fondamentali di ogni transazione di tipo economico, quindi in particolare di quelle tra banca e clientela, consiste nella documentazione (cartacea) che di queste transazioni deve essere mantenuta a fini probatori dei modi e della entità di quella. Nulla viene specificato circa l'efficacia probatoria delle informazioni registrate su supporto magnetico, come è il caso di quelle di un elaboratore che utilizza nastri o dischi, né dei documenti cartacei eventualmente deducibili da quelle informazioni (per questi deve esistere l'avallo di un operatore umano). Il documento cartaceo è cioè un termine costante di riferimento che l'avvento delle tecnologie elettroniche tenderebbe a eliminare. In particolare, uno degli elementi più importanti in documenti di questo genere è costituito dalla sottoscrizione da parte del suo estensore; la sottoscrizione è una parte essenziale del documento, ai fini delle attribuzioni di responsabilità e necessita, inoltre, della presenza fisica del sottoscrittore, perché la validazione del documento deve essere autografa. Questo aspetto può creare problemi, nel momento in cui si pensi di utilizzare tecniche telematiche per la trasmissione di documenti con valore legale.

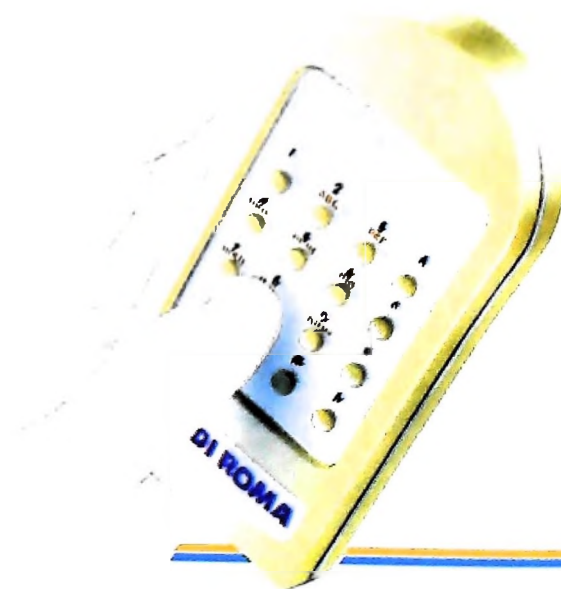
Sempre facendo riferimento al nostro ordinamento, va tenuto presente che l'unico bene legalmente idoneo a estinguere una qualsiasi situazione debitoria è la moneta, che è l'unica a produrre effetto solutorio. Anche nel caso di assegni e titoli di credito, l'effetto solutorio derivante dal loro uso è subordinato all'accertamento della loro copertura effettiva.

Un ulteriore problema posto quindi dal trasferimento elettronico di fondi è costituito dall'attribuzione effettiva del trasferimento stesso, ovvero è relativo al momento in cui questa operazione può effettivamente considerarsi assoluta ed essere quindi, a tutti i fini, un pagamento vero e proprio, anche se non vi è stata movimentazione reale di denaro, ma soltanto uno scambio di messaggi tra banche. Nel caso di trasferimento elettronico il concetto di pagamento come atto unico e istantaneo viene a essere modificato. Quell'evento è sostituito da una procedura, quindi da una successione di eventi che si verificano entro un certo periodo di tempo. Il problema sta nel decidere a che punto di questo periodo di tempo si considera effettuato il pagamento, e quindi il trasferimento, tenuto conto della complessità delle operazioni di trasmissione e verifica della copertura, e soprattutto in relazione alla tendenza all'aumento della quantità di operazioni di questo genere. Stabilire il momento effettivo in cui il trasferimento si considera eseguito è importante, per esempio,

nel caso in cui si desideri revocare il trasferimento stesso, così come per la determinazione dell'appartenenza dei fondi in funzione di possibili azioni da parte di creditori e dell'accertamento dell'entità del patrimonio di chi autorizza il trasferimento e di chi ne è il beneficiario. Se, per esempio, si adottasse il criterio di considerare effettuato il trasferimento nel momento in cui viene data comunicazione di questo, verrebbe immediatamente eliminata ogni possibilità di revoca di questo atto. Nel caso, invece, di ritenere valida questa operazione solo quando il conto del beneficiario sia stato effettivamente accreditato, si andrebbe contro quella che è ormai una pratica comune da parte delle banche, che consiste nel posticipare le operazioni di verifica delle coperture, per accelerare le pratiche. Il concatenamento di problemi diventa ancora più complesso nel caso di operazioni di trasferimento di fondi tra banche poste in paesi diversi, dotati di ordinamenti giuridici diversi. La situazione attuale si è evoluta verso la definizione a livello locale di strumenti giuridici e di legge at-

Home-banking

"Home-banking" significa, sostanzialmente, "la banca in casa" (o nel proprio ufficio): non è necessario recarsi fisicamente in una agenzia per effettuare le normali operazioni bancarie. Un sistema di home-banking completo è ancora una prospettiva per il futuro, ma esistono già sistemi per l'esecuzione da casa di almeno alcune operazioni più semplici, come la richiesta di informazioni sullo stato del proprio conto corrente. L'interazione con il calcolatore centrale della banca avviene via telefono e mediante un terminale domestico che può essere un piccolo terminale tascabile, "dedicato" e di costo bassissimo, un terminale portatile con tastiera alfanumerica e schermo, oppure personal computer (con appositi programmi) o terminali di servizi telematici del tipo "videotex".



ti a regolamentare questo settore e i nuovi aspetti che emergono; in questo senso, l'atto legislativo finora più completo è costituito dal "Fair Fund Transfer Act", emesso nel 1978 dal Congresso degli USA.

Uno degli aspetti di grande rilevanza non solo nel settore bancario, legato all'applicazione di nuove tecnologie, è rappresentato dalla sicurezza delle informazioni e delle operazioni che su queste vengono espletate. Si pongono per esempio problemi relativi al rispetto del cosiddetto segreto bancario e quindi alla protezione, in generale, da accessi indebiti alle informazioni sui clienti e alle norme che devono regolamentare gli accessi autorizzati. Negli USA, per esempio, esistono società la cui attività consiste essenzialmente nel garantire la sicurezza delle informazioni detenute da una banca e nel prevenire i cosiddetti "computer crimes".

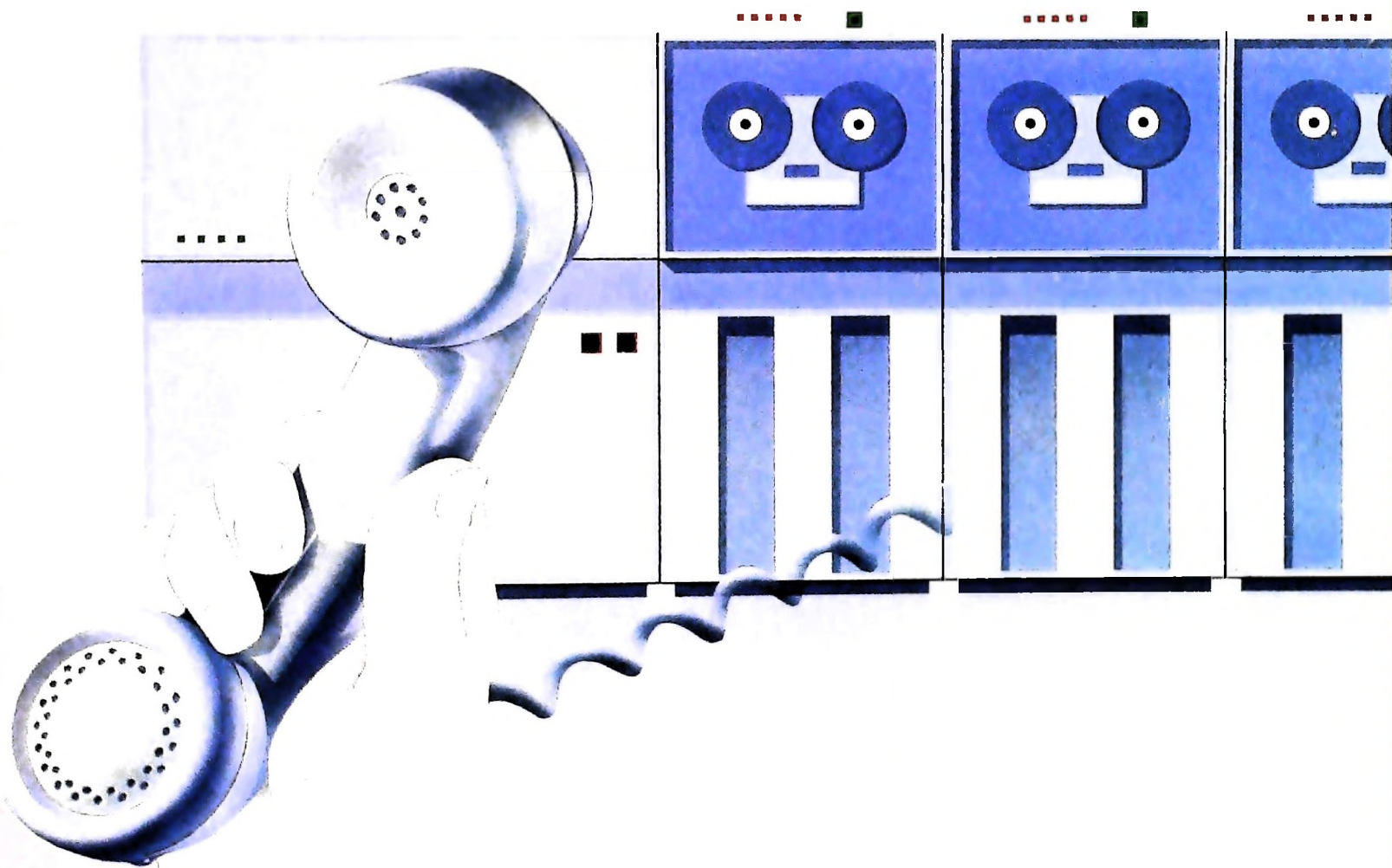
Si tratta di reati perpetrati attraverso l'uso del computer della banca stessa, oppure collegandosi ad esso via cavo telefonico, per esempio con un calcolatore personale. Persone abili

e malintenzionate, a conoscenza delle procedure bancarie, sono riuscite a farsi riconoscere come utenti privilegiati dal sistema di gestione dei fondi e a far accreditare indebitamente cifre più o meno ingenti su propri conti.

Una soluzione ai problemi di riservatezza e autenticazione consiste nelle tecniche di crittografia.

Il cittadino/cliente deve essere inoltre garantito da possibili errori di tipo tecnico dovuti a malfunzionamenti degli impianti o ad errori umani. L'aspetto più rilevante in questo senso è che il supporto magnetico per le informazioni non fornisce ancora le stesse garanzie di stabilità e durata che assicura invece la carta.

Nel nostro paese, non esiste ancora una regolamentazione precisa di questi aspetti legati alla sicurezza delle informazioni e alla riservatezza delle stesse. Solo recentemente è stato presentato in Parlamento un disegno di legge che dovrebbe stabilire i criteri per l'installazione e l'accesso di qualsiasi banca di dati sia pubblica che privata.





Immagini trattate digitalmente

Le immagini generate da un computer si possono in linea di massima dividere in due gruppi:

1. immagini sintetiche, non esistenti nel mondo reale e formate a partire da un modello matematico;
2. immagini trattate digitalmente, generate da un originale esistente nel mondo reale.

In questi esempi considereremo solo questo secondo gruppo. L'immagine di partenza appare in alto e successivamente si hanno due immagini formate con due tipi diversi di input. In particolare possiamo osservare:

- A. immagine prodotta con una tavoletta digitalizzatrice e uno stilo elettromagnetico (*paint system*, ossia disegno manuale);
- B. immagine generata dalla fusione della tecnica esposta in A e della tecnica che si basa sull'immagine generata con una telecamera che esplora automaticamente l'originale, fornendo poi l'informazione al computer.

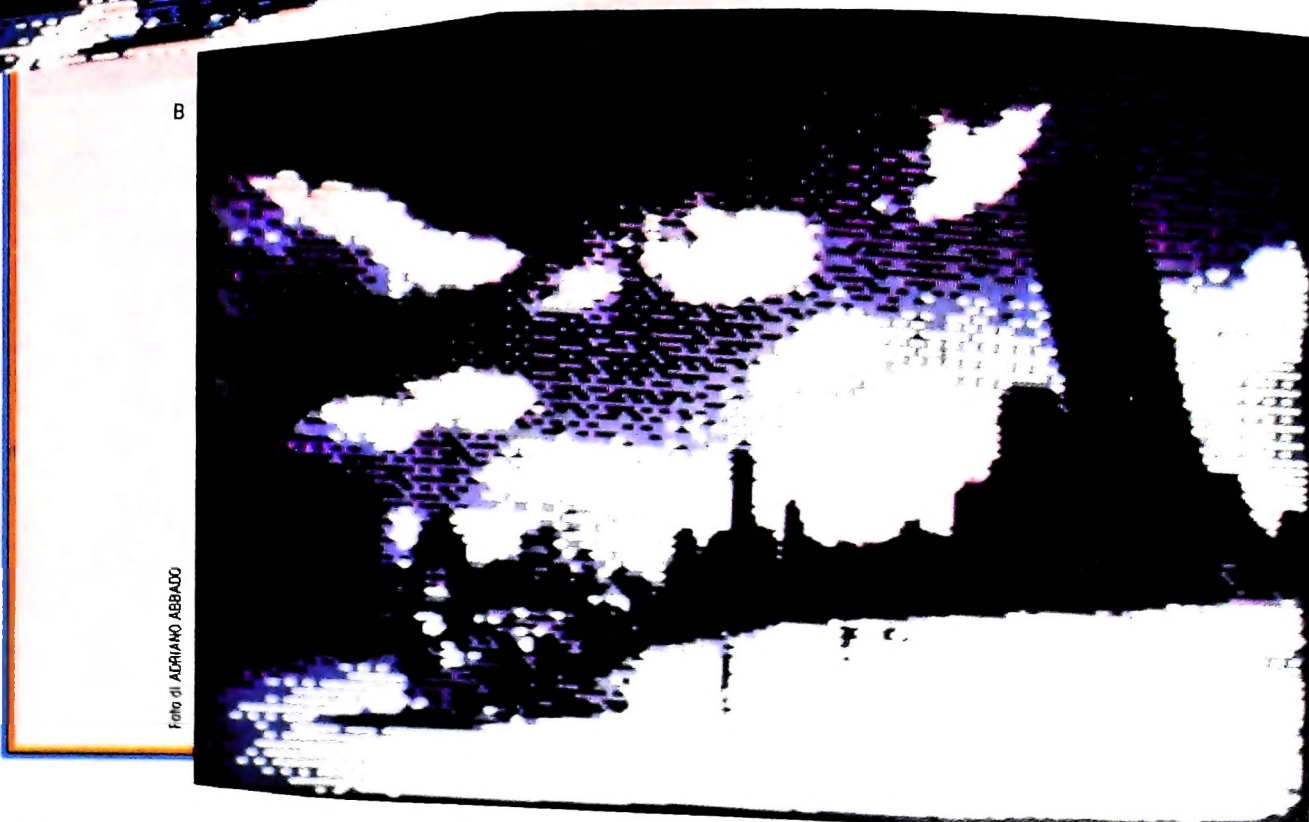
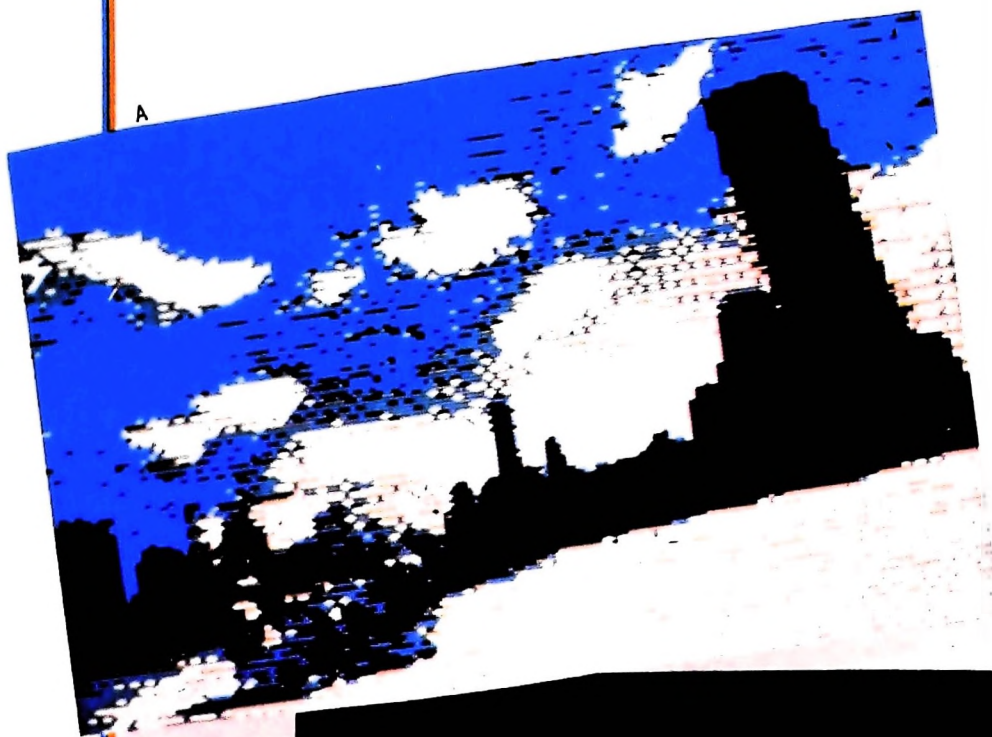


Foto di ADRIANO ABBADO

Olivetti M10 vuol dire disporre del proprio ufficio in una ventiquattrore. Perché M10 non solo produce, elabora, stampa e memorizza dati, testi e disegni, ma è anche capace di comunicare via telefono per spedire e ricevere informazioni. In grado di funzionare a batteria oppure collegato all'impianto elettrico, M10 mette ovunque a disposizione la sua potenza di memoria, il suo display orientabile a cristalli liquidi capace anche di elaborazioni grafiche, la sua tastiera professionale arricchita da 16 tasti funzione.



Ma M10 può utilizzare piccole periferiche portatili che ne ampliano ancora le capacità, come il microplotter per scrivere e disegnare a 4 colori, o il registratore a cassette per registrare dati e testi, o il lettore di codici a barre. E in ufficio può essere collegato con macchine per scrivere elettroniche, con computer, con stampanti. Qualunque professione sia la vostra, M10 è in grado, dovunque vi troviate, di offrirvi delle capacità di soluzione che sono davvero molto grandi. M10: il più piccolo di una grande famiglia di personal.

PERSONAL COMPUTER OLIVETTI M10

L'UFFICIO DA VIAGGIO



Anche in leasing con Olivetti Leasing.

olivetti

UN NUOVO MODO DI USARE LA BANCA.

OGNI VALORE HA UN SUO MODO DI USARE LA BANCA.

GLI INVESTIMENTI CON VOI E PER VOI DEL BANCO DI ROMA.

Il Banco di Roma non si limita a custodire i vostri risparmi. Vi aiuta anche a farli meglio fruttare. Come? Mettendovi a disposizione tecnici e analisti in grado di offrirvi una consulenza di prim'ordine e di consigliarvi le forme di investimento piú giuste. Dai certificati di deposito ai titoli di stato, dalle obbligazioni alle azioni, il Banco di Roma vi propone professionalmente le varie opportunità del mercato finanziario. E grazie ai suoi "borsini", vi permette anche di seguire, su speciali video, l'andamento della Borsa minuto per minuto.

Se desiderate avvalervi di una gestione qualificata per investire sui piú importanti mercati mobiliari del mondo, i fondi comuni del Banco di Roma, per titoli italiani ed esteri, vi garantiscono una ampia diversificazione.

Inoltre le nostre consociate Figeroma e Finroma forniscono consulenze per una gestione personalizzata del portafoglio e per ogni altra esigenza di carattere finanziario.

Veniteci a trovare: ci conosceremo meglio.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.

