

CADU

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

48 CORSO PRATICO COL COMPUTER

422006

è una iniziativa
FABBRI EDITORI

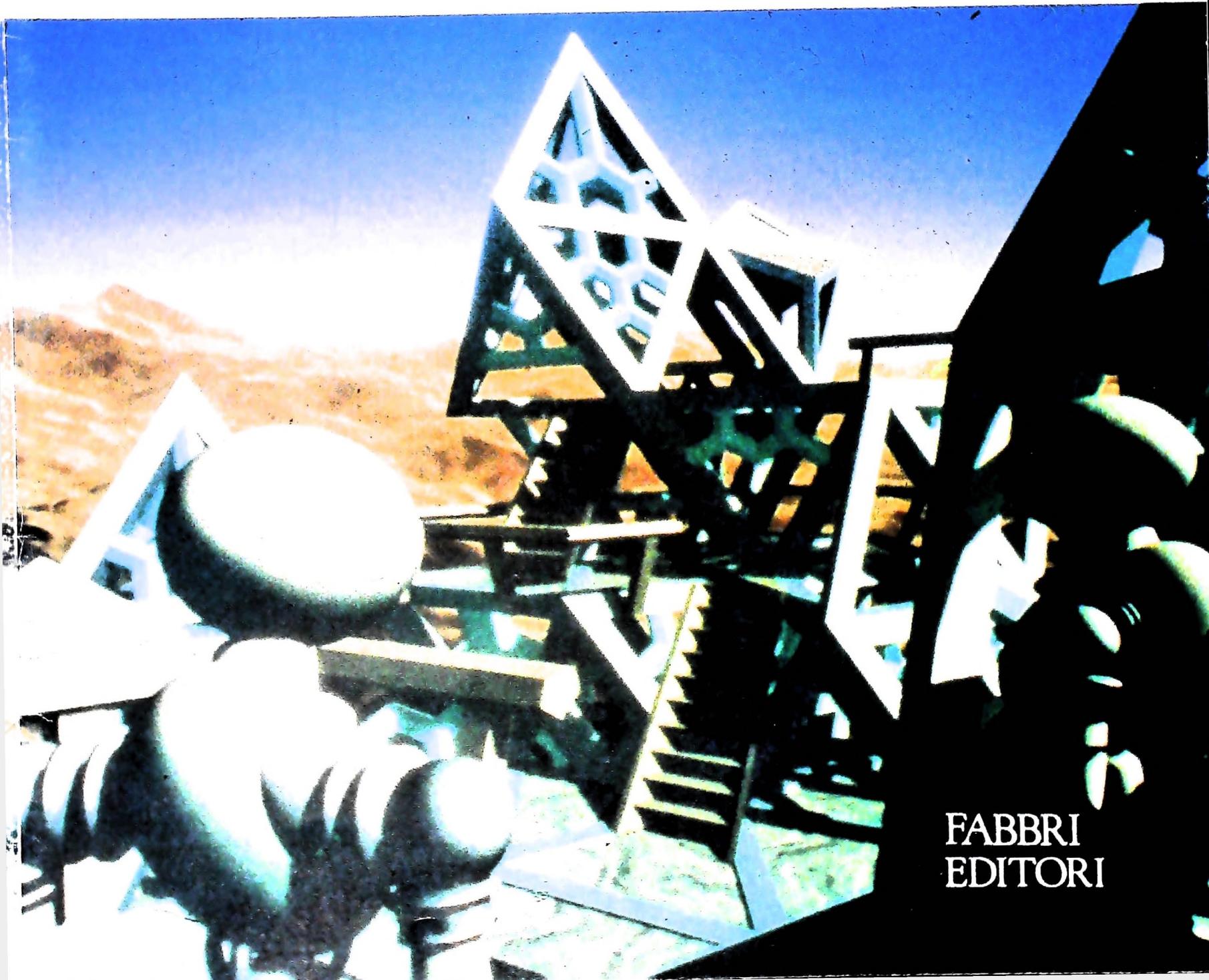
in collaborazione con
BANCO DI ROMA

e **OLIVETTI**



Banco di Roma

diretto da **GIANNI DEGLI ANTONI**



**FABBRI
EDITORI**

IL BANCO DI ROMA

FINANZIA IL VOSTRO ACQUISTO

DI M 10 e M 20

Acquisto per contanti

È la formula di acquisto tradizionale. Non vi sono particolari commenti da fare, se non sottolineare che troverete ampia disponibilità presso i punti di vendita Olivetti, poiché, grazie al "Corso pratico col computer", godrete di un rapporto di privilegio.

Il servizio di finanziamento bancario

Le seguenti norme descrivono dettagliatamente il servizio di finanziamento offerto dal Banco di Roma e dagli Istituti bancari a esso collegati:

Banca Centro Sud
Banco di Perugia

Le agenzie e/o sportelli di questi istituti sono presenti in 216 località italiane.

Come si accede al credito e come si entra in possesso del computer

- 1) Il Banco di Roma produce una modulistica che è stata distribuita a tutti i punti di vendita dei computer M 10 e M 20 caratterizzati dalla vetrofania M 10.
- 2) L'accesso al servizio bancario è limitato solo a coloro che si presenteranno al punto di vendita Olivetti.
- 3) Il punto di vendita Olivetti provvederà a istruire la pratica con la più vicina agenzia del Banco di Roma, a comunicare al cliente entro pochi giorni l'avvenuta concessione del credito e a consegnare il computer.

I valori del credito

Le convenzioni messe a punto con il Banco di Roma, valide anche per le banche collegate, prevedono:

- 1) Il credito non ha un limite minimo, purché tra le parti acquistate vi sia l'unità computer base.
- 2) Il valore massimo unitario per il credito è fissato nei seguenti termini:
 - valore massimo unitario per M 10 = L. 3.000.000
 - valore massimo unitario per M 20 = L. 15.000.000
- 3) Il tasso passivo applicato al cliente è pari

al "prime rate ABI (Associazione Bancaria Italiana) + 1,5 punti percentuali".

- 4) La convenzione prevede anche l'adeguamento del tasso passivo applicato al cliente a ogni variazione del "prime rate ABI"; tale adeguamento avverrà fin dal mese successivo a quello a cui è avvenuta la variazione.
- 5) La capitalizzazione degli interessi è annuale con rate di rimborso costanti, mensili, posticipate; il periodo del prestito è fissato in 18 mesi.
- 6) Al cliente è richiesto, a titolo di impegno, un deposito cauzionale pari al 10% del valore del prodotto acquistato, IVA inclusa; di tale 10% L. 50.000 saranno trattentate dal Banco di Roma a titolo di rimborso spese per l'istruttoria, il rimanente valore sarà vincolato come deposito fruttifero a un tasso annuo pari all'11%, per tutta la durata del prestito e verrà utilizzato quale rimborso delle ultime rate.
- 7) Nel caso in cui il cliente acquisti in un momento successivo altre parti del computer (esempio, stampante) con la formula del finanziamento bancario, tale nuovo prestito attiverà un nuovo contratto con gli stessi termini temporali e finanziari del precedente.

Le diverse forme di pagamento del finanziamento bancario

Il pagamento potrà avvenire:

- presso l'agenzia del Banco di Roma, o Istituti bancari a esso collegati, più vicina al punto di vendita Olivetti;
- presso qualsiasi altra agenzia del Banco di Roma, o Istituto a esso collegati;
- presso qualsiasi sportello di qualsiasi Istituto bancario, tramite ordine di bonifico (che potrà essere fatto una volta e avrà valore per tutte le rate);
- presso qualsiasi Ufficio Postale, tramite vaglia o conto corrente postale. Il numero di conto corrente postale sul quale effettuare il versamento verrà fornito dall'agenzia del Banco di Roma, o da Istituti a esso collegati.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.

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
Ricercatore universitario presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

Curatori di rubriche
MARCO ANELLI, DIEGO BIASI, ANDREA GRANELLI, ALDO GRASSO, MARCO MAIOCCHI, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARMELLI

Testi
CLAUDIO PARMELLI, ANDREA GRANELLI, ALDO GRASSO, MASSIMO MASSIRONI
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

Redazione
CARLA VERGANI
LOGICAL STUDIO COMMUNICATION

Ari Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÈ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer - Copyright © sul fascicolo 1985 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 10 marzo 1984 - Iscrizione al Registro Nazionale della Stampa n. 00262, vol. 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico del Gruppo Editoriale Fabbri S.p.A., Milano - Diffusione - Distribuzione per l'Italia Gruppo Editoriale Fabbri S.p.A., via Mecenate, 91 - Milano - tel. 02/50951 - Pubblicazione periodica settimanale - Anno II - n. 48 - esce il giovedì - Spedizione 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

PROTOCOLLI DI RETE

Governano il flusso dei dati in una rete, assicurandone un trasferimento veloce e corretto.

Capacità delle linee di trasmissione

È frequente, nei libri o negli articoli inerenti le trasmissioni dati, l'utilizzo come sinonimi dei due termini "bit al secondo" e "baud". Questo non è corretto, vedremo ora qual è la differenza.

Si può esprimere la capacità di trasportare dati di una linea di comunicazione indicando la massima quantità di informazioni che può essere trasmessa lungo la linea in condizioni ideali. L'unità di informazione più piccola che si può impiegare è il bit (che ormai conosciamo), e risulta conveniente quindi esprimere la capacità del canale con il numero di bit al secondo che possono essere trasportati dal canale stesso. Se trasmettiamo dati lungo una linea telefonica, moduliamo quei dati sull'onda sinusoidale della portante.

La frequenza dell'onda sinusoidale che si può inviare (ottenendo risultati soddisfacenti) lungo un circuito telefonico normale è di circa 3 KHz. I primi modem erano costruiti secondo tecniche di modulazione che imponevano effettivamente di modulare un bit per ciclo della frequenza fondamentale della portante: un modem a 300 bit al secondo (bps) impiegava quindi una portante a 300 Hz (cicli al secondo).

Successivamente, lo studio di migliori tecniche di modulazione, ha reso possibile la modulazione di più bit per ciclo della portante. Esistono quindi modem a 2400 bps che impiegano una portante a 1200 Hz (cps) modulando due bit per ciclo, modem da 4800 bps che modulano 3 bit per ciclo su una portante da 1600 cps, modem da 9600 bps con una portante a 2400 cps e 4 bit per ciclo.

A rigore, la velocità in baud esprime la frequenza della portante, si dovrà quindi dire che il modem a 2400 bps impiega una portante a 1200 baud, mentre spesso il modem viene indicato avente una velocità di 2400 baud.

Livelli di protocollo

I protocolli di rete sono insiemi di regole che governano il flusso dei dati in una rete con lo scopo di assicurare che i dati siano trasferiti velocemente e correttamente da un punto a un altro. Tali protocolli quindi controllano l'individuazione e la correzione automatica dell'errore e prevedono procedure di ripristino (*Recovery procedures*) in modo da consentire l'ordinata gestione dei casi che possono presentarsi.

Come si può vedere dalla figura delle pagine seguenti, in una rete ci possono essere parecchi livelli di protocollo. Nel diagramma, è rappresentata una rete con due concentratori: l'elaboratore che sta trasmettendo una serie di messaggi multi-segmento a un terminale remoto (dall'inglese *Remote job entry* o abbreviato RJE, sigla che spesso verrà utilizzata), cioè un terminale impiegato per l'introduzione remota dei lavori, collegato al concentratore n. 2.

Il messaggio trasmesso può consistere di un certo numero di blocchi di dati e ogni blocco deve viaggiare su tre spezzoni di linea (o DATA LINKS) e attraverso due concentratori per raggiungere il terminale RJE. In questo esempio, si dovranno avere almeno tre livelli di protocollo.

Il protocollo al livello più basso è quello che definisce le operazioni di hardware e che indica un insieme di regole sul modo di inviare e di ricevere i singoli bit di dati sulle linee di trasmissione. Questo protocollo definirà il tipo di interfaccia hardware che verrà utilizzata e potrà essere l'interfaccia V24 tra l'elaboratore e il modem e l'interfaccia raccomandata X.21 se si sta usando una rete digitale.

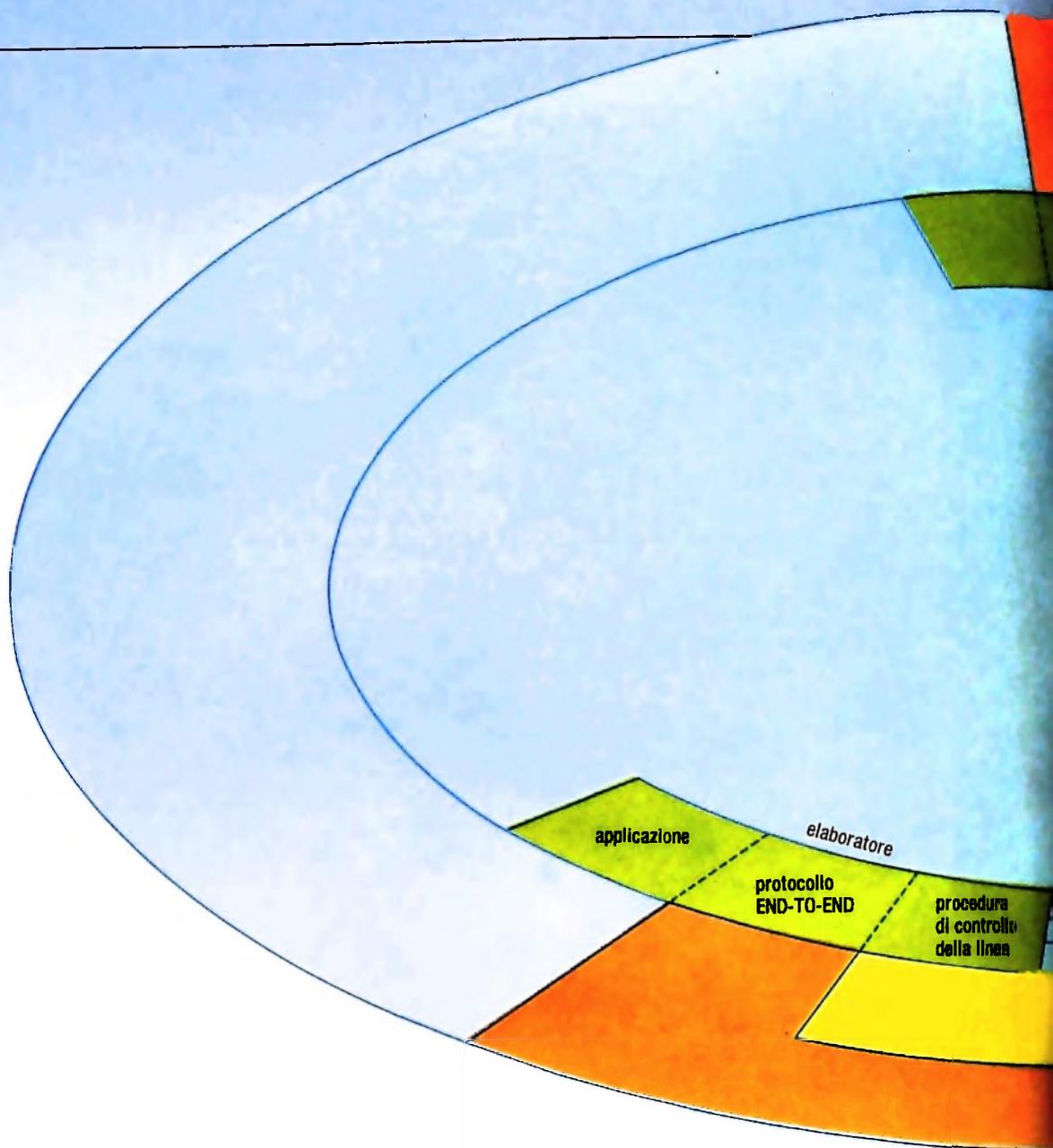
Il protocollo prevede quindi una serie di operazioni dette di *Handshaking* che si devono svolgere per permettere al mittente e al destinatario di dialogare. Quindi, quando si vuole trasmettere si alza il segnale RTS e si aspetta il segnale RFS prima di cominciare a trasmettere.

Nella maggior parte dei sistemi, questi protocolli di basso livello sono trasparenti per l'utente al quale basta conoscerne l'esistenza e i ritardi che di solito ne conseguono, senza altri dettagli. In breve i protocolli di hardware a più basso livello assicurano che i bit possano essere inviati sulla linea di trasmissione e che possano essere rilevati all'altro estremo.

Questi protocolli non offrono però alcuna protezione contro gli errori di trasmissione. L'eventuale bit errato verrebbe gestito dai protocolli di basso livello come tutti gli altri. Per far fronte a questa e altre eventualità, si ricorre a un protocollo a più alto livello chiamato *Procedura di controllo di linea* (o *Protocollo di linea* o *Disciplina di linea*), cioè un insieme di regole che assicurano che il blocco dei dati vada da un estremo all'altro della linea correttamente.

Ciò richiede l'individuazione e la correzione automatica di eventuali errori. Per raggiungere questo obiettivo occorre innanzitutto specificare il formato dei blocchi dei dati e il metodo impiegato per individuare gli errori. Il meccanismo di individuazione degli errori è già insito nei dispositivi collocati a ogni estremo del collegamento e viene avviato con una

L'illustrazione ci mostra una rete a più tronchi con diversi livelli di protocollo. È importante ricordare che i protocolli di rete sono insiemi di regole che governano il flusso dei dati in una rete. Essi assicurano il corretto trasferimento dei dati da un punto a un altro. L.C.P. sta per Procedura di Controllo di Linea. R.J.E. sta per Remote Job Entry. In particolare, nel diagramma è rappresentata una rete con due concentratori.



serie di messaggi di supervisione che informano il dispositivo trasmettente del modo con cui si svolge la trasmissione. Se scopre un errore, il dispositivo ricevente può chiedere la trasmissione del blocco.

La procedura di controllo della linea definisce rigorosamente il formato e il modo con cui devono essere impiegati tutti i messaggi di supervisione per poter gestire qualsiasi eventualità che si possa verificare lungo la linea.

I termini di *Procedura di controllo della linea* e *Procedura di controllo del collegamento* vengono di solito impiegati indifferentemente, ma forse il secondo è più corretto.

Controllo di linea

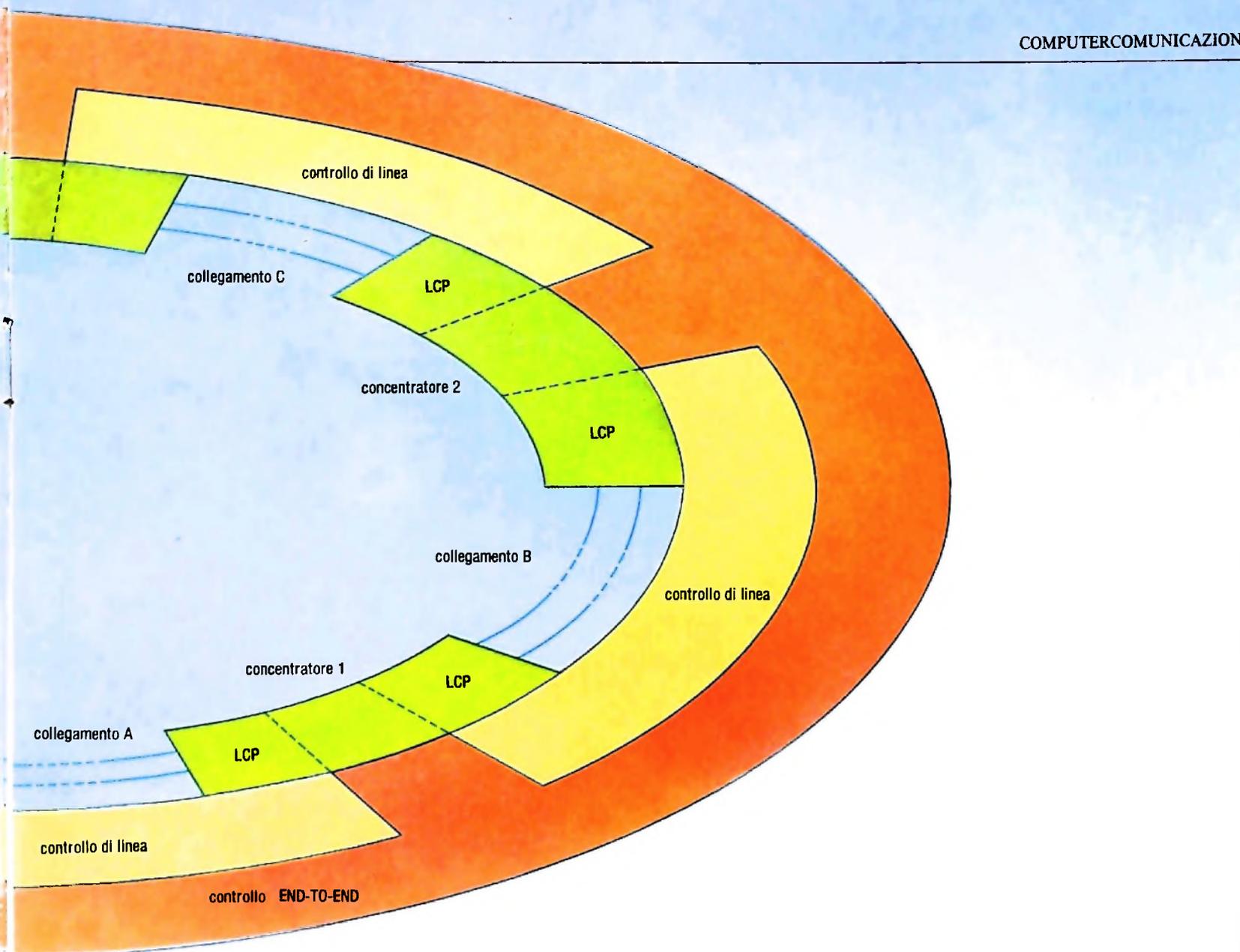
Con questo termine si indica di solito il controllo del flusso di dati lungo la linea di trasmissione, ma in realtà la proce-

dura ha significato più vasto. Infatti con essa si controlla il flusso di dati lungo un collegamento logico punto-a-punto tra due terminali (o elaboratori) costituito dai seguenti elementi: linee di comunicazione, modem, interfacce hardware di comunicazione sul terminale (o elaboratore) e parte di hardware del terminale (o elaboratore) e/o di software impiegato per ospitare le procedure di controllo che operano sui collegamenti.

Secondo questo criterio, il blocco di dati viene immesso sulla linea di trasmissione a un estremo tramite la procedura di controllo del collegamento, e verrà restituito correttamente tramite la stessa procedura di controllo del collegamento all'altro estremo della linea.

Se il blocco si perde completamente lungo il cammino, la procedura di controllo del collegamento darà inizio a una idonea azione di ripristino.

Si può trasferire il concetto di collegamento dati come con-



nessione logica punto a punto alle linee multidrop perché in ogni singola operazione di trasferimento dei dati si invieranno dati da un punto a un altro anche se ci possono essere molti terminali sulla linea.

Nella rete che compare nella figura riprodotta sopra, si trovano procedure di controllo che operano sui collegamenti A, B e C: ognuna di queste procedure assicura che i singoli blocchi di dati attraversino in modo corretto i collegamenti da esse controllati.

Per essere sicuri che un messaggio attraversi la serie di collegamenti e i concentratori e raggiunga la sua destinazione avremo bisogno di un protocollo a più alto livello, indicato nella figura come *Protocollo end-to-end* (o protocollo da un estremo all'altro).

Questo protocollo raccoglie un altro insieme di regole relative al flusso di messaggi. Un messaggio può essere costituito da un certo numero di blocchi e l'elaboratore deve sapere se

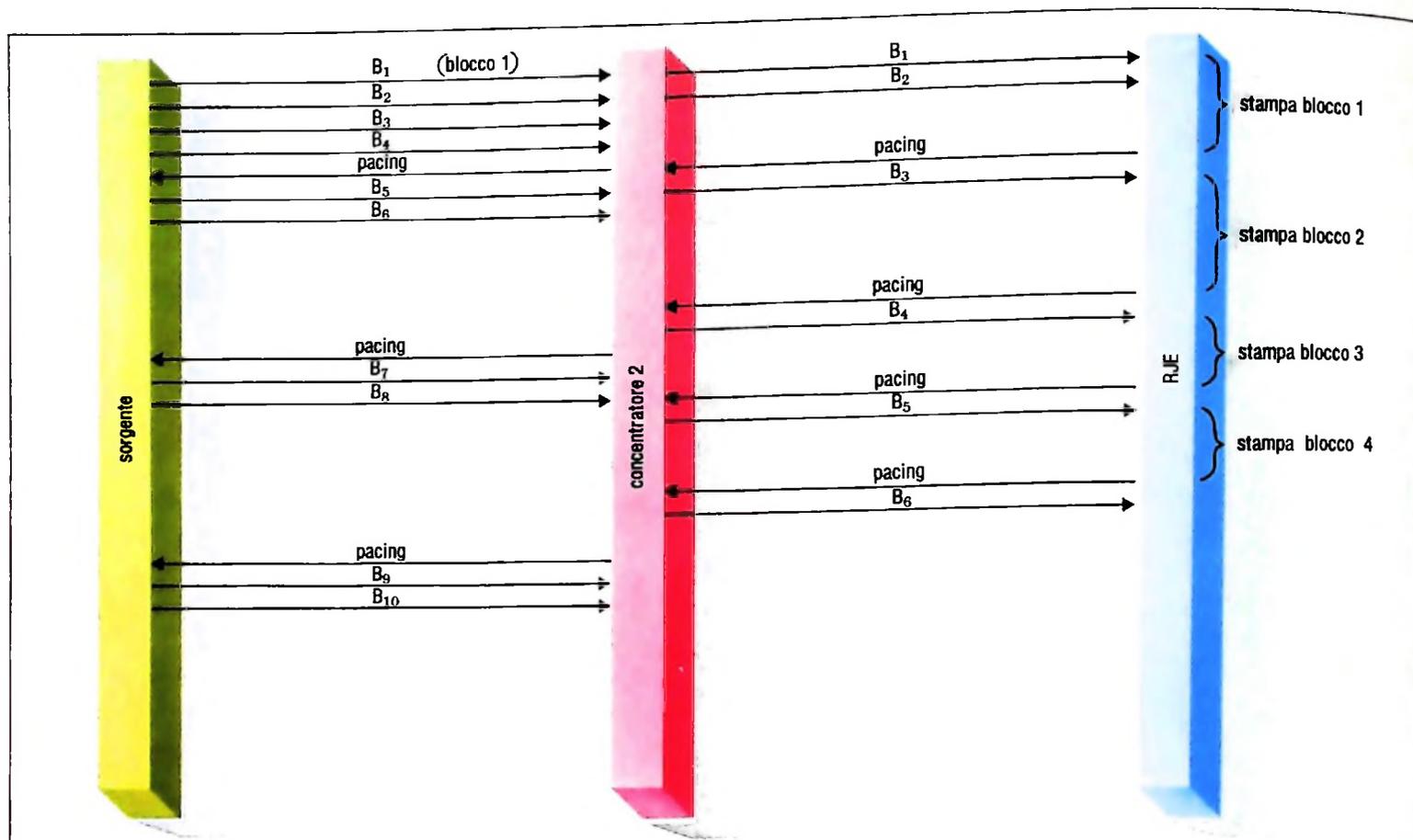
il messaggio è stato ricevuto completamente.

Il protocollo end-to-end, dopo aver ceduto tutti i segmenti del messaggio alla prima procedura di controllo di linea, può inviare un messaggio speciale per richiedere al terminale destinatario se ha davvero ricevuto l'intero messaggio.

Se il messaggio non è stato ricevuto completamente, il protocollo end-to-end può dare inizio a un'azione di ripristino.

Controllo del flusso

Il protocollo di livello più alto ha anche un'altra funzione: il *Controllo del flusso*. Ciò vuol dire che tale protocollo deve assicurare che i dati scorrano senza intoppi lungo la rete, senza sovraccaricare terminali e concentratori. Il protocollo di collegamento di solito tenta di consegnare i dati quanto più velocemente gli è possibile, ma il dispositivo destinatario può



Flusso di controllo del terminale RJE nella rete a più tronchi della precedente illustrazione.

imporre dei limiti sulla velocità con la quale può accettare i dati in arrivo.

Un terminale RJE, per esempio, può avere due memorie di transito in modo da essere in grado di ricevere un blocco di dati mentre sta stampando il blocco ricevuto in precedenza.

Se la linea di comunicazione può consegnare i dati a una velocità superiore a quella alla quale il terminale RJE è in grado di stamparli, si può avere un sovraccarico del terminale, a meno che non sia stato previsto qualche controllo che consenta di evitare questa eventualità.

Per stabilire questo controllo si può seguire il metodo secondo il quale si impiega un messaggio detto di *Pacing* che interviene segnando il ritmo o il passo dei dati che vanno dal destinatario fino alla sorgente.

In tal modo si possono inviare al terminale RJE i primi due blocchi alla massima velocità che può essere gestita dal sistema, il quale però non sarà in grado di ricevere il terzo blocco fino a quando non avrà finito di stampare il primo: così non manderà una risposta di *pacing* fino a quando non avrà finito di stampare il primo blocco.

Ricevuto il messaggio di *pacing* che dà il via libera, la sorgente invierà il successivo blocco di dati e così via. Il protocollo end-to-end alla sorgente deve conoscere le caratteristiche del terminale per capire cosa fare, come in questo caso, in cui deve sapere che può inviare rapidamente i primi due

blocchi ma che poi deve attendere una risposta di *pacing* prima di inviare il blocco successivo.

Come indicato nella figura riportata sopra, i livelli ai quali si scandisce il tempo in una rete possono essere diversi: il diagramma si limita a mostrare il modo con cui si segna il ritmo dei messaggi senza precisare tuttavia tutta la sequenza a livello di collegamento di scambio dei messaggi.

Il concentratore 2, per esempio, può assegnare quattro buffer al terminale RJE permettendo di far funzionare alla massima velocità il terminale stesso. Si può quindi procedere a controllare il flusso tra il concentratore 2 ed il terminale RJE nel modo già descritto, ma si può applicare un diverso insieme di regole tra sorgente e concentratore 2.

Poiché il concentratore ha quattro buffer, la sorgente può inviare i primi quattro blocchi il più velocemente possibile aspettando poi dal concentratore il messaggio che scandisce il tempo.

Il concentratore, dopo aver consegnato i primi due blocchi al terminale RJE, invia un messaggio che scandisce il tempo all'elaboratore centrale (HOST) che a sua volta provvede a inviare i due blocchi successivi.

Dopo aver inviato quei primi due blocchi al terminale RJE, il concentratore è pronto per riceverne altri due e per segnalare questo invia un messaggio di scansione iniziando la ripetizione della sequenza.

LINGUAGGI E GRAMMATICHE

Ogni linguaggio è basato su regole che definiscono formalmente la struttura delle frasi.

Sintassi e semantica

Ogni *linguaggio* è basato su un "vocabolario" di parole o simboli. È caratteristica dei linguaggi che alcune sequenze di simboli siano riconosciute come frasi corrette del linguaggio, mentre altre sono considerate illegali.

Ciò che determina se una sequenza di parole o una frase è corretta e quindi valida è la *grammatica* oppure la *sintassi* o struttura del linguaggio.

Infatti intuitivamente possiamo considerare la sintassi come l'insieme delle regole che definiscono le frasi formalmente corrette del linguaggio.

Ancora più importante è il fatto che tali regole conferiscono alle frasi una struttura che è strumentale nel riconoscimento del significato o *semantica* delle stesse.

Esaminando per esempio la frase del tipo

" cane mangia "

riconosciamo una struttura sintattica fatta di un soggetto e un verbo, dove il soggetto è in grado di compiere l'azione descritta dal verbo.

Se però noi modifichiamo la frase in

" palla mangia "

avvertiamo subito che c'è qualcosa che non va nella semantica, nel contenuto informativo della frase, pur rispondendo a tutte le caratteristiche grammaticali di quella precedente.

Ne consegue che la sintassi e la semantica sono intimamente connesse. Le definizioni strutturali devono pertanto sempre essere considerate come ausiliarie a un più alto scopo.

Di seguito ci limiteremo tuttavia esclusivamente agli aspetti strutturali dei linguaggi formali, ignorando gli aspetti di significato e interpretazione.

- una frase è formata da un soggetto seguito da un verbo,
- un soggetto consiste o del simbolo "cane" o del simbolo "palla",
- un verbo è sostituibile dal solo simbolo "mangia"

Definizione di grammatica e linguaggio

Per descrivere un linguaggio e la sua sintassi abbiamo bisogno di un altro linguaggio, un *metalinguaggio*, non confondibile o mescolabile con l'oggetto del nostro studio.

Il metalinguaggio più diffuso è la notazione Bockus-Nour-Form (BNF), che prende il nome da coloro che la idearono e applicarono per la prima volta alla definizione del linguaggio algoritmico ALGOL60.

Le due frasi precedenti appartengono al linguaggio che può essere descritto in modo formale dalla seguente sintassi espressa in BNF:

```
<frase> ::= <soggetto> <verbo>
<soggetto> ::= cane | palla
<verbo> ::= mangia
```

In questo formalismo distinguiamo:

- i *simboli terminali*, cioè i simboli che, opportunamente giustapposti, formano le frasi del linguaggio

"cane palla mangia"

- i *simboli non terminali* o *metasimboli*, che servono per descrivere la struttura delle frasi del linguaggio e rappresentano le sue categorie sintattiche rilevanti.

("<frase>", "<soggetto>", "<verbo>").

Le tre *regole grammaticali* o *di produzione* indicano le modalità con cui è possibile sostituire un simbolo non terminale o metasimbolo.

Si leggono rispettivamente così:

L'idea di base è che una frase del linguaggio può essere derivata partendo da un metasimbolo iniziale per applicazione successiva di regole di produzione.

Nel nostro caso, partendo dal metasimbolo <frase> otteniamo per sostituzione le due frasi precedenti: le regole grammaticali determinano come sono generate o prodotte le frasi legali del linguaggio.

Sintetizzando e formalizzando meglio, una grammatica è definita come una quadrupla

$$G = \langle T, N, P, S \rangle$$

dove:

- T è l'insieme o ALFABETO dei simboli terminali,
- N è l'insieme dei simboli non terminali,
- P è l'insieme delle regole di produzione,
- S è il SIMBOLO DISTINTO o SCOPO della grammatica: esso è il metasimbolo dal quale partire nella costruzione delle frasi.

Allora il linguaggio descritto dalla grammatica G è l'insieme delle sequenze di simboli terminali, le *sentenze*, che possono essere generate a partire dal simbolo distinto applicando le regole di produzione specificate.

Un esempio

Per essere più chiari sarà bene scendere in ulteriori dettagli. Consideriamo la seguente grammatica:

```
T = < IL, LA, BIMBO, MELA, GATTO, LATTE, MANGIA, AMA >
N = < <frase>, <soggetto>, <verbo>, <complemento>, <articolo>, <nome> >
P = < <frase>      ::= <soggetto> <verbo> <complemento>,
      <soggetto>   ::= <articolo> <nome>,
      <complemento> ::= <articolo> <nome>,
      <articolo>    ::= IL | LA,
      <nome>        ::= BIMBO | MELA | GATTO | LATTE,
      <verbo>       ::= MANGIA | AMA >
S = <frase>
```

Allora possiamo, partendo dal simbolo distinto, operare sostituzioni indicate dalle regole dell'insieme P, fino a ottenere

una sequenza di soli simboli terminali. Così applicando successivamente le regole indicate

```
<frase>
(regola 1) <soggetto> <verbo> <complemento>
(regola 2) <articolo> <nome> <verbo> <complemento>
(regola 3) <articolo> <nome> <verbo> <articolo> <nome>
(regola 4) IL <nome> <verbo> <articolo> <nome>
(regola 5) IL BIMBO <verbo> <articolo> <nome>
(regola 6) IL BIMBO MANGIA <articolo> <nome>
(regola 4) IL BIMBO MANGIA LA <nome>
(regola 5) IL BIMBO MANGIA LA MELA
```

si ottiene una sentenza legale del linguaggio.

La suddetta grammatica permette anche di generare frasi

che, rispetto alla lingua italiana, sono semanticamente prive di significato; del tipo:

IL MELA MANGIA LA BIMBO

Frasi come questa oltre ad essere semanticamente prive di significato non rispettano le regole di concordanza tra articoli e nomi.

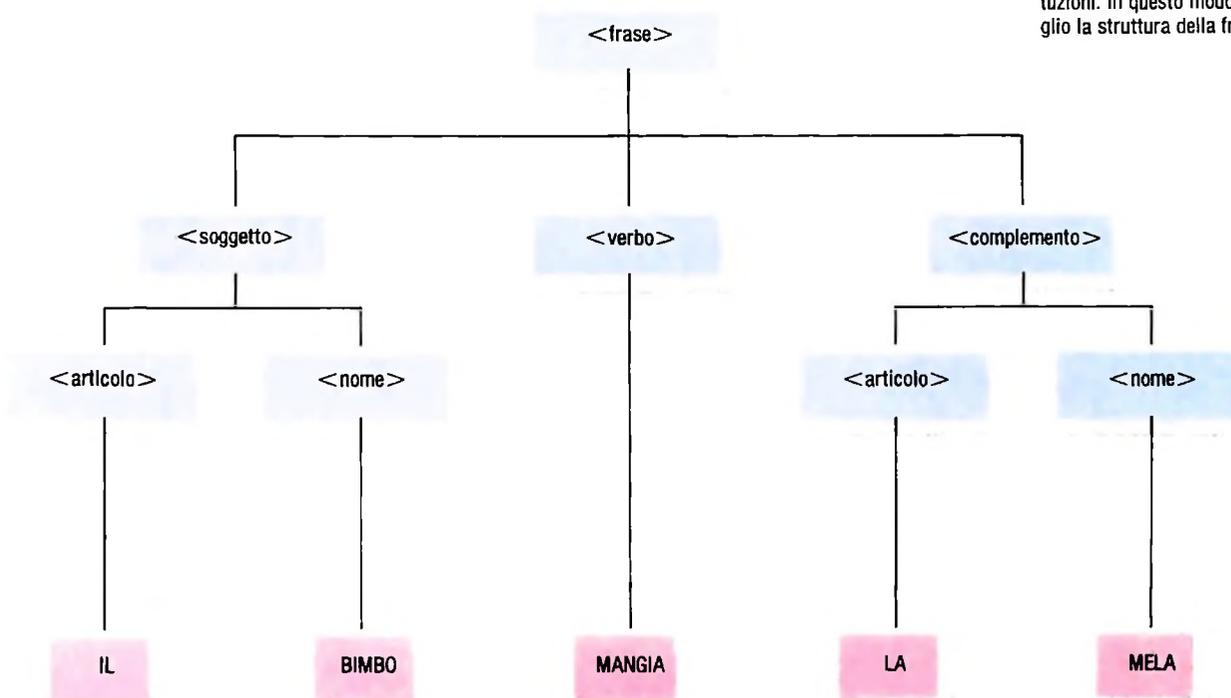
Frasi di questo tipo sono comunque da considerarsi corrette in quanto rispettano le regole di produzione della grammatica. Per quanto riguarda le regole di concordanza è possibile ricorrere a grammatiche che vincolino l'occorrenza di un

simbolo, sia terminale sia non terminale, a un particolare contesto.

Così per esempio si potrebbe vincolare la presenza dell'articolo IL ai nomi maschili della grammatica considerata. Grammatiche di questo tipo vengono denominate "dipendenti dal contesto".

Spesso, per mostrare meglio la struttura della frase, si rappre-

Nello schema detto 'albero di derivazione' vengono così rappresentate le sostituzioni. In questo modo osserviamo meglio la struttura della frase.



sentano le sostituzioni con uno schema come quello della figura sopra che viene detto ALBERO DI DERIVAZIONE.

Sostituiamo ora la produzione sul complemento con la seguente:

`<complemento> ::= <articolo> <nome> | <articolo> <nome> , <complemento>`

avendo introdotto anche un nuovo simbolo terminale " , ".

Una simile produzione, che usa il metasimbolo <complemento> come elemento di sostituzione dello stesso è detta RICORSIVA e permette di estendere all'infinito il numero di frasi del linguaggio.

Possiamo infatti ora generare frasi come quella illustrata nella figura che possiamo osservare nella pagina seguente. Si nota infatti che, in questo tipo di albero di derivazione, possiamo prolungare a nostro piacimento la lista degli elementi che costituiscono il complemento, arrestandola con la scelta della prima alternativa della produzione.

Una grammatica diventa così uno strumento finito atto a descrivere un insieme di frasi infinito.

Una grammatica come quella illustrata permette di sostituire

un metasimbolo con una adeguata sequenza. Questo è possibile anche indipendentemente dal contesto in cui il metasimbolo si colloca.

Il linguaggio descritto è detto per tale motivo LIBERO DA CONTESTO. Più formalmente si vede che in ogni produzione della grammatica il lato sinistro è costituito da un solo simbolo non terminale.

È possibile tuttavia osservare che esistono linguaggi più complessi da descrivere e, al contrario, linguaggi che risultano assai più semplici.

Tra i primi, quelli DIPENDENTI DA CONTESTO: nella grammatica di tali linguaggi si esprime la possibilità di sostituire un metasimbolo con altri solo se questo si trova in un adeguato contesto.

Lezione 47

Ancora sui grafici delle funzioni

Nell'ultima lezione abbiamo tracciato il grafico della funzione

$$y = \sin(x)$$

in modo molto povero e veramente poco elegante da un punto di vista estetico. Cercheremo in questa lezione di rimediare a ciò che abbiamo tralasciato, cogliendo l'occasione per esaminare nuovi comandi per il microplotter.

Ci proponiamo infatti di:

- tracciare la funzione seno in rosso
- evidenziare gli assi x e y con la loro scala
- scrivere il nome della funzione sul foglio.

Tracciare gli assi

Cominciamo a porci il problema di tracciare gli assi: ciò è molto facile con il nostro microplotter, in quanto abbiamo a disposizione un comando apposito, e precisamente il comando X.

Esso permette di tracciare contemporaneamente l'asse e i segmenti che ne individuano la scala, e ha la seguente forma:

$$Xp, s, n$$

ove:

p può valere

0 se vogliamo tracciare l'asse verticale delle y

1 se vogliamo tracciare l'asse orizzontale delle x

s rappresenta il numero di colonne che deve essere lasciato tra un tratto e l'altro dell'indicazione della scala sull'asse

n indica quanti tratti di suddivisione vogliamo che compaiano sull'asse tracciato per rappresentare la scala.

Il comando X deve essere usato in modo grafico, usando la solita istruzione LPRINT.

Esso traccia l'asse rispetto al punto in cui il pennino è posizionato nell'istante in cui il comando viene eseguito.

Così, volendo tracciare l'asse x in modo da avere 4 suddivisioni, ciascuna di $\pi/2$, potremo usare le istruzioni:

```
10 LPRINT CHR$(18)
15 LPRINT "X1,120,4" 'Asse x
```

Analogamente, per tracciare l'asse y con 6 tratti di 50 righe ciascuno, tre sopra e tre sotto l'asse x, potremo aggiungere le istruzioni:

```
16 LPRINT "M0,-150" 'Asse y
17 LPRINT "X0,50,6"
```

ove la linea 16 ha lo scopo di posizionarsi in basso, sotto l'asse x, per tracciare la parte negativa dell'asse y.

Possiamo ora aggiungere la parte di programma che traccia il seno di x, chiedendo di tracciare il segno in rosso:

```
19 LPRINT "C3" 'Rosso
20 FOR I=0 TO 6.28 STEP .1
30 LET P$="D"+STR$(I*480/6.28)+"", "+STR$(
SIN(I)*100)
40 LPRINT P$
50 NEXT I
```

Poniamoci ora l'obiettivo di scrivere, a fianco delle varie "tacche" sugli assi, i valori relativi. Scriveremo queste informazioni in blu, e useremo semplicemente un insieme di comandi "M" per posizionarci nei punti in cui scrivere, e "P" per scrivere. Traceremo le seguenti scritte:

- "0", vicino all'origine
- "Pi/2", "Pi", "3/2 Pi", "2 Pi" sui tratti di separazione dell'asse delle x, ove con "Pi" indichiamo il valore del pi greco
- "+1" e "-1" sull'asse delle y
- "x" e "y" agli estremi dei due assi rispettivi.

Aggiungeremo quindi al programma le seguenti istruzioni:

```
60 REM Scala
65 LPRINT "C1" 'Blu
70 LPRINT "M4,-18" 'Origine
80 LPRINT "P0"
90 LPRINT "M106,-18" 'Pi mezzi
100 LPRINT "PPi/2"
110 LPRINT "M230,-18" 'Pi
120 LPRINT "PPi"
130 LPRINT "M340,-18" '3/2 Pi
140 LPRINT "P3/2 Pi"
150 LPRINT "M438,5" '2 Pi
155 LPRINT "P2 Pi"
```

```

160 LPRINT "M5,95"      'Asse y
170 LPRINT "P+1"
180 LPRINT "M5,-105"
190 LPRINT "P-1"
200 LPRINT "M3,150"    'Coordinata y
210 LPRINT "Py"
220 LPRINT "M470,-20"  'Coordinata x
230 LPRINT "Px"

```

I valori inseriti nei comandi "M" sono stati ottenuti per tentativi: a fronte di una prima costruzione del programma si sono osservati problemi, difficoltà di lettura, sovrapposizioni dei segni; dopo qualche tentativo è stato chiaro quali valori dovessero essere usati.

Ora, aggiungiamo il nome della funzione, in verde, selezionando opportunamente un valore per la dimensione dei caratteri, con il comando "S". Come al solito, useremo il comando "M" per posizionarci opportunamente.

Aggiungeremo al programma le seguenti istruzioni:

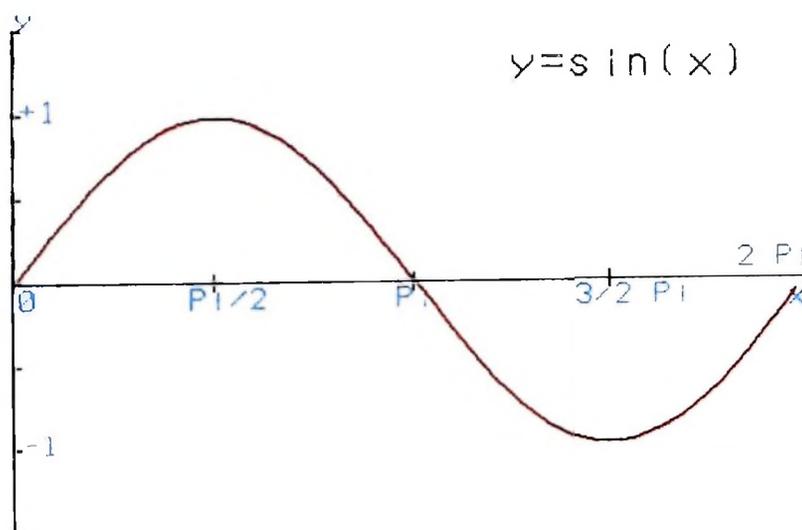
```

240 LPRINT "M300,120"  'Nome funzione
250 LPRINT "S2"        ' Cambia misura
260 LPRINT "C2"        ' In verde
270 LPRINT "Py=s in(x)"

```

Finalmente possiamo considerare completo il nostro programma e passare quindi all'esecuzione.

Il risultato è il seguente:



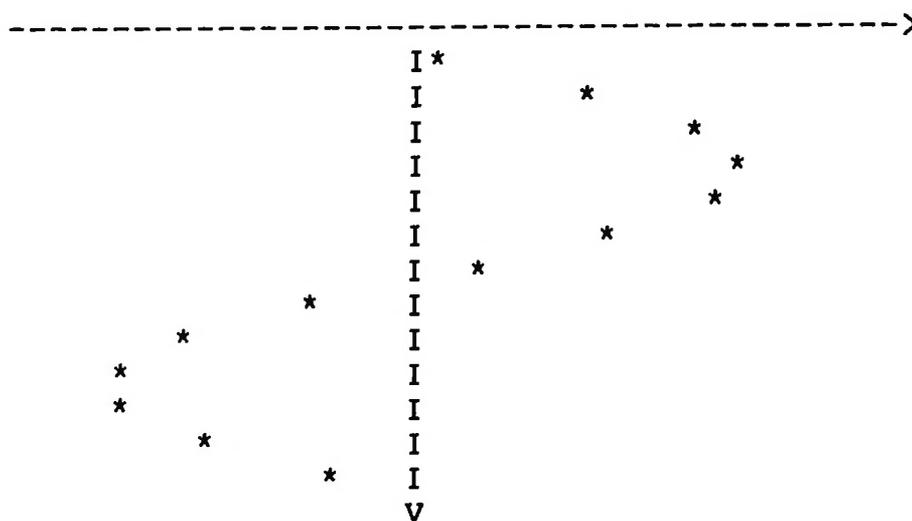
Il microplotter non è l'unico strumento per poter tracciare grafici, ovviamente. Già abbiamo visto in una precedente lezione un grafico tracciato sullo schermo dell'M10 azionando un pixel per volta. Inoltre, anche con una stampante tradizionale è

possibile ottenere grafici.
 Per renderci conto, però, della scarsa qualità grafica rispetto a quella che possiamo ottenere con il microplotter, esaminiamo il seguente programma:

```

10 REM Grafico con stampante
20 LPRINT "----->"
30 FOR I=0 TO 6.28 STEP .5
35 IF SIN(I)>=0 THEN 55
40 LPRINT TAB(SIN(I)*15+20);"*";TAB(19);"I"
50 GOTO 60
55 LPRINT TAB(19);"I";TAB(SIN(I)*15+20);"*"
60 NEXT I
65 LPRINT TAB(19);"V"
  
```

che fornisce il seguente risultato:



Si noti l'istruzione LPRINT per dichiarare che la visualizzazione deve essere fatta su stampante, e la funzione TAB; dire PRINT TAB (10); "*" è come dire: "sposta la testina di stampa alla posizione 10, e poi stampa un asterisco".

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- il comando X, che permette di tracciare gli assi di un grafico, con le relative suddivisioni di scala;
- l'istruzione LPRINT usata con la stampante;
- la funzione TAB che permette di posizionare il punto di stampa.

I GIOCHI ELETTRONICI

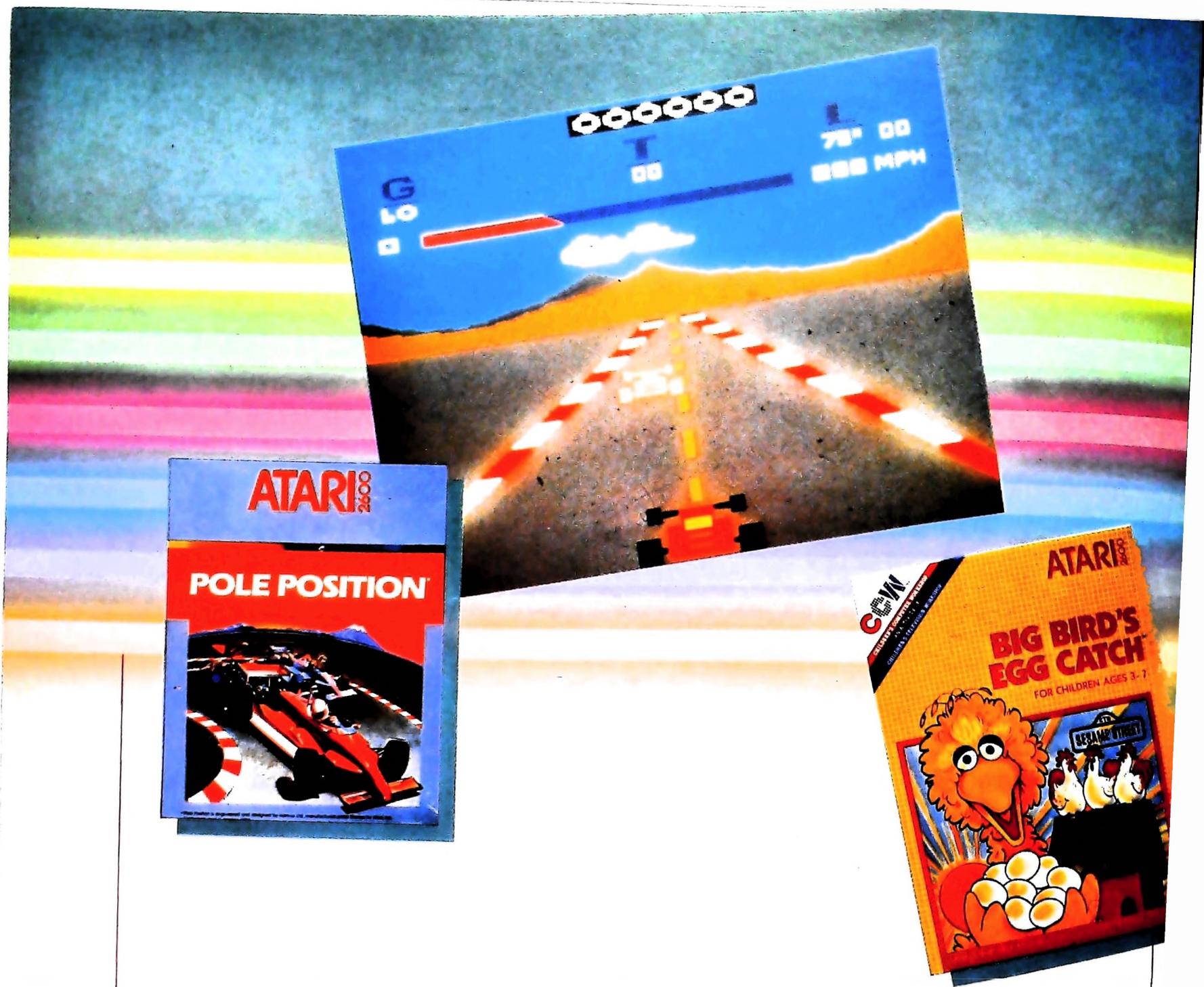
Le meraviglie della tecnologia al servizio del gioco.

La strategia con cui l'elettronica ha deciso di entrare in tutte le case è soffice e avvolgente: i computer & videogame hanno così ricevuto l'incarico di porgere il catalogo dei programmi con i quali i computer intendono riplasmare il mondo. Con un piccolo, ma decisivo cambiamento, sta finendo l'epoca in cui i giocattoli prefiguravano rigorosamente l'universo delle funzioni adulte, significavano sempre qualcosa, si compiacevano di essere in qualche modo dei "formati ridotti" di veri oggetti umani.

I giochi elettronici sembrano abbandonare questa meccanica delle uguaglianze e delle similitudini "realistiche": invece di rappresentare il mondo degli adulti, preferiscono esercitarsi sul loro immaginario (film, Tv, libri, fumetti...). Il giocattolo si avvia a stabilire equivalenze più complesse: rifacendo il mondo degli adulti, il suo compito principale era quello di renderlo innocente, di purificarlo. Adesso, basato com'è sull'immagine, conosce solo un rapporto simulativo mettendo spesso in causa la differenza tra "vero" e "falso". Il giocatto-

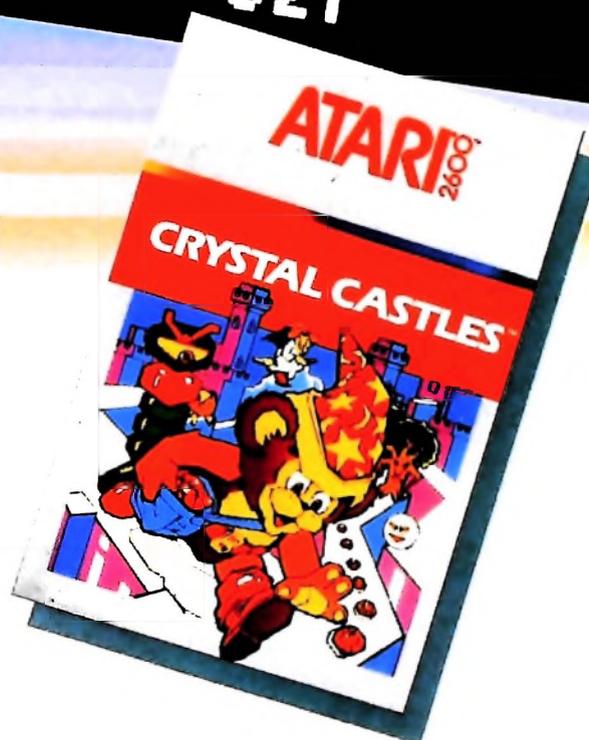
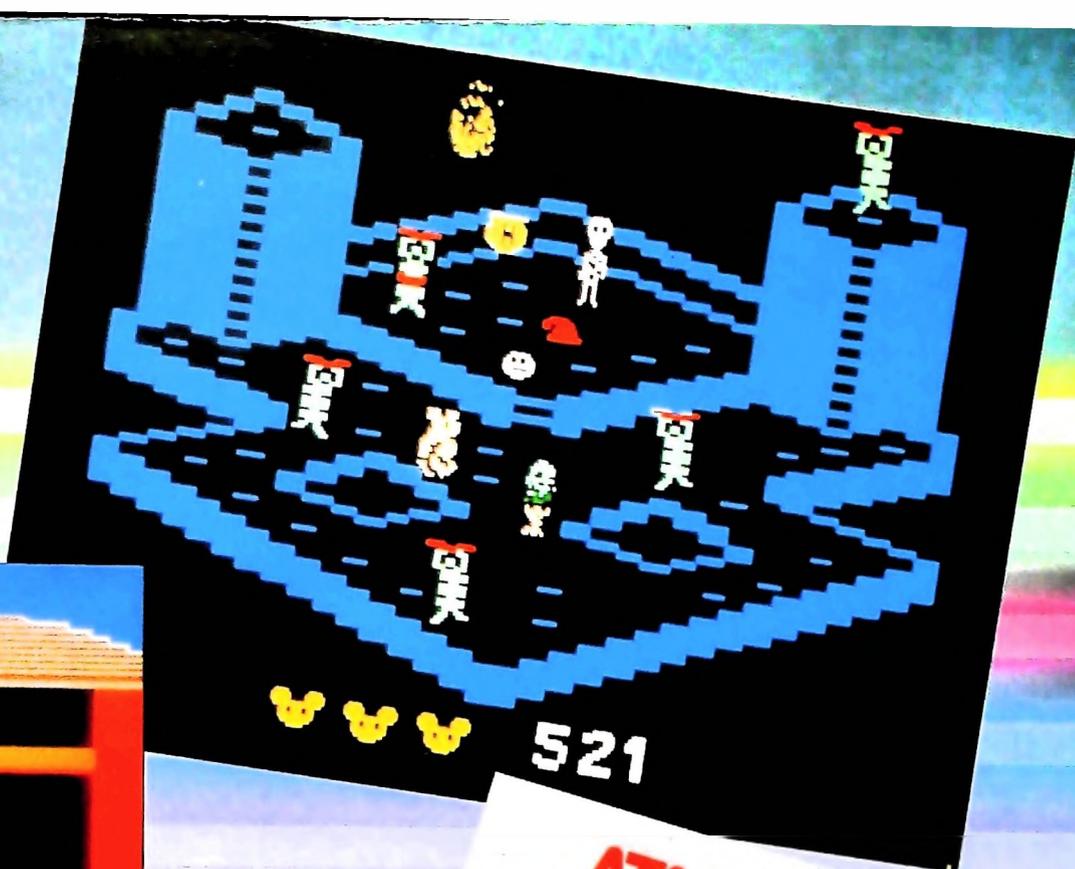
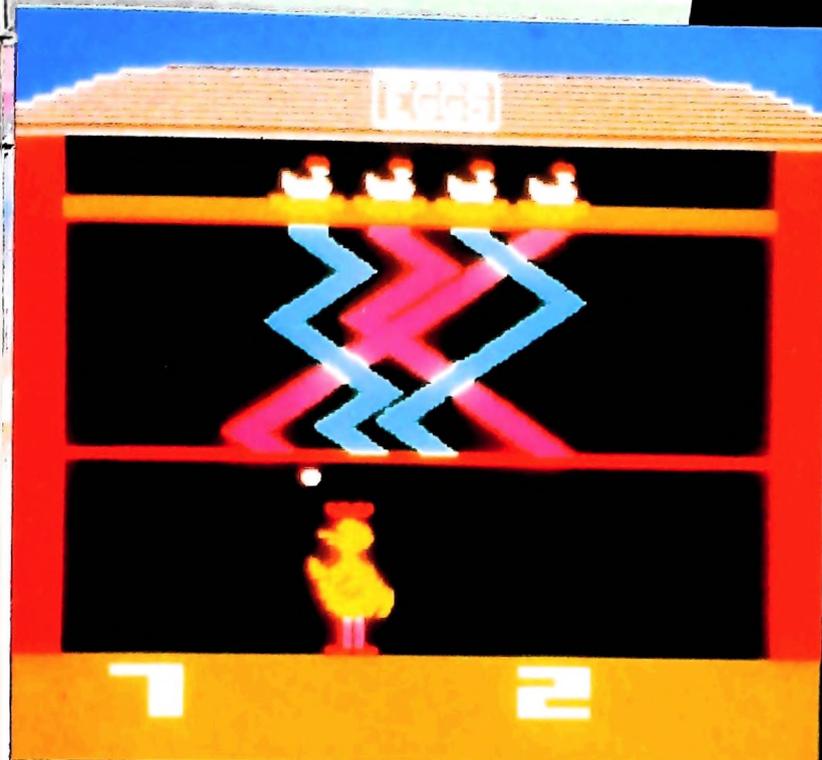


Il Barone Rosso può ora volare nei cieli dell'elettronica. I personaggi più amati dai ragazzini di tutto il mondo vivono oggi le loro avventure grazie ai programmi di gioco più o meno sofisticati.



lo tradizionale si colloca in un microcosmo della *produzione*, il gioco elettronico in quello della *riproduzione*. Tutto cominciò nel 1962, quando uno studente del prestigioso MIT (il Massachusetts Institute of Technology) realizzò un primo, rudimentale videogioco: "Space War" (Guerra spaziale) era il suo nome e lo firmava Steve Russell. Per giocare però c'era bisogno del grosso computer dell'Università, di un'approfondita conoscenza tecnologica e di molta pazienza. Ma un altro studente, Nolan Bushnell, cominciava intanto a elaborare per il proprio computer una serie di programmi passatempo: "A quell'epoca — sostiene oggi Bushnell — non avevo certo le idee chiare. I miei giochi erano per iniziati e soprattutto richiedevano una conoscenza elevata dei linguaggi del calcolatore". Il rischio insomma era quello che il videogioco rimanesse un piacevole diversivo per professori annotati.

Dopo molti, inutili tentativi ecco finalmente il colpo di genio. Nel 1972 Bushnell realizzò in pochi mesi "Pong", un game in cui il giocatore doveva cercare di respingere un quadratino luminoso ruotando una manopola che comandava una rudimentale paletta, sfidando il computer o un altro avversario. Era in pratica la trasposizione del tennis da tavolo in codice elettronico, un meccanismo che non richiedeva nessuna tastiera per essere manovrato ma due normali manopole. Bushnell cercò di vendere l'idea alla Midway, consociata della Bally e una delle maggiori società americane di apparecchiature per il divertimento automatico. La Midway rifiutò la proposta considerando quel rudimentale gioco elettronico un'idea senza futuro. E così Bushnell decise di fondare lui stesso una società che chiamò *Atari*, prendendo a prestito un termine del gioco orientale del "GO", che significa più o meno "scacco matto". Nel maggio dello stesso anno fu la



Magnovox a scrivere un'altra pagina della storia dei videogiochi, presentando il primo videogame da casa, "Odyssey", nato da un altro grande "creativo", Ralph Baer. Il gioco però non ebbe successo e l'idea venne parzialmente accantonata. In capo a pochi mesi, altre società, che avevano subito fiutato l'affare, scesero in campo e si organizzarono per la conquista di quel nuovo territorio.

"I videogiochi, allora, erano ingombranti e 'stupidi' — ricorda ancora Bushnell — e funzionavano soltanto a gettone nei locali specializzati (le Arcades). Ben presto mi resi conto che avremmo perso la partita se l'Atari non fosse riuscita in fretta a produrre qualcosa di diverso". È il 1975 quando Bushnell crea la versione casalinga del "Pong", una scatola di plastica nera che si collega al televisore di casa mediante lo stesso ingresso dell'antenna. Piccolo, facile da usare, relativamente economico (costava sulle trentamila lire), il "Pong"

domestico è subito un grande successo: in neppure dodici mesi la Atari ne vende 100 mila pezzi.

Il 1976 è l'anno della svolta: mentre Bushnell si fa assorbire da una multinazionale dello spettacolo, altre aziende si inseriscono sul mercato e soprattutto molti ingegneri che lavoravano per la NASA nel settore della simulazione dei voli, a seguito delle riduzioni volute da Carter dei programmi spaziali, dovettero commercializzare il loro lavoro, i loro sofisticatissimi programmi. Inizia così l'invasione dei computer e dei videogame.

Adesso, passati dieci anni circa, i videogiochi — intesi come cartucce, come giochi per console — stanno attraversando un brutto momento. Alcune ditte prestigiose hanno dovuto chiudere i battenti. Com'è possibile che i videogiochi siano già entrati in crisi? Molti i fattori che hanno determinato tale situazione: saturazione del mercato (rispetto, naturalmente,

alle consolle vendute), costo elevato della cartuccia, errori di valutazione, ripetitività nelle proposte, politiche distributive all'insegna dell'improvvisazione. Fra le varie cause, una è stata determinante. I videogame si rivolgono a una fascia d'età molto giovane (diciamo le scuole elementari); dopo l'entusiasmo iniziale il ragazzino fa presto ad annoiarsi, vuol cominciare a fare qualcosa lui. Ecco perché non conoscono crisi gli home computer, i quali offrono la possibilità di giocare usando la tastiera. Più correttamente, allora, va specificato che nel settore giochi elettronici sono in crisi i videogame, mentre incontrano ancora il grande favore del pubblico i cosiddetti computer game.

I laserdisc game

Per quel che riguarda i giochi elettronici nei locali pubblici, la grande novità si chiama laserdisc game. Basato sul principio del compact disc a lettura laser, questo nuovo sistema ha segnato il più grande salto tecnologico nel campo dei giochi. Codificando centinaia di immagini con precisi dettagli a colori (i "solchi" di un compact disc corrispondono a segnali di frequenza modulata che possono essere trasformati in suoni e immagini quando sono colpiti da un raggio laser a basso potere di neon-elio) e programmando il microprocessore a trovare ciascuna di esse nell'attimo stesso in cui si chiede l'"informazione", il game offre ai giocatori la netta impressione di essere nel vivo di un cartone animato o di una scena filmata. Anzi il giocatore ha come l'impressione, e per questo si parla di gioco interattivo, di essere un regista che modifica a suo piacimento le sequenze del filmato-gioco. I discolaser sono stati trasformati in giochi interattivi grazie alla possibilità di poter saltare quasi immediatamente da un punto all'altro del disco. Così in un gioco al laser si può salvare da morte sicura il protagonista con una mossa ben calcolata del joystick: il laser salta su una banda diversa del disco e propone al giocatore una nuova situazione. Questo sarebbe impossibile da ottenere con un supporto tradizionale, cartuccia o nastro che sia.

Visto che in seguito parleremo soprattutto dei giochi che si possono ottenere anche con la tastiera, in particolare con quella di M10, soffermiamoci ancora un poco sui laserdisc game, segnalando i tre maggiori successi. "Dragon's Lair", il primo gioco su discolaser, ha decretato la fine delle immagini composte da trattini e puntini e ha aperto la strada a splendidi disegni animati, degni delle migliori produzioni cinematografiche. Ecco perché Don Bluth, il suo creatore, è stato soprannominato a buon diritto il Walt Disney degli anni Ottanta. "Dragon's Lair" è una favola di magia e stregoneria che presenta ben 29 situazioni diverse: giocando bene, senza fare errori, si vedono sei minuti di animazione dall'inizio alla fine; ma tutte le possibilità di gioco riunite danno un totale di ventidue minuti. "M.A.C.H.3" è un gioco di simulazione di volo, estremamente affascinante. Ci sono bersagli da colpire, nemici da abbattere e, se si vuole sopravvivere, ci vuole molta abilità nel tenere la rotta: ma soprattutto si prova la sensazione di volare davvero tanto è perfetta l'integrazione

tra grafica e tema. "Firefox" si basa sulle scene originali dei film di Clint Eastwood, combinando azioni cinematografiche con immagini digitali. Il giocatore si cala nei panni del maggiore Gant e ruba il progetto Firefox all'Unione Sovietica. Segue una battaglia contro una flotta di Mig sovietici. Giustamente Atari ha equipaggiato "Firefox" con gli stessi strumenti di controllo per il volo già felicemente sperimentati con "Star wars".

Storie senza scampo né riscatto

Quelle storie senza scampo né riscatto, quei combattimenti privi di inibizione, dubbi o ripensamenti, che hanno fatto la fortuna iniziale dei videogame sembrano guardare al passato più che al futuro e, paradossalmente, rappresentano l'atto di fedeltà all'infanzia (di coloro cui si rivolgono e dell'elettronica stessa) che i nuovi giochi hanno voluto coscientemente sottoscrivere. Distruzioni totali, annientamenti, eliminazioni — che tanto hanno fatto trepidare le buone coscienze dei pedagoghi — rafforzano quel gusto della crudeltà che è uno dei tratti tipici del comportamento infantile e saldano il vecchio al nuovo. Curiosa questa strada che l'elettronica si è scelta per farsi desiderare: le meraviglie della tecnologia al servizio della ben collaudata spietatezza delle fiabe.

Tre figure archetipe dominano sovrane nella costituzione dei computer & videogame: la ricerca, la nutrizione e la soppressione. La continua reversibilità della prima, il suo movimento inesausto, ruota intorno a due gesti irreversibili: mangiare e uccidere.

La ricerca, il viaggio, il distacco: da sempre la composizione di qualsiasi racconto è costruita sullo spostarsi dell'eroe nello spazio: "Odissea" e "Ombre rosse", "Il gatto con gli stivali" e "Tex Willer". La narrazione nasce dall'incanto del viaggiare, dal movimento fra scene sempre mutevoli; così spesso l'opera assume come forma mitica quella del viaggio: processo conoscitivo, metafora di un processo critico, percorso di iniziazione. Una messe copiosa di game, come vedremo, ha per schema di base la figura del labirinto.

Sull'esempio di "Pac man" il video appare come un frequentatissimo fast food popolato da strani personaggi che sembrano provenire direttamente da "Gargantua e Pantagruele" di Rabelais: tutto ciò che viene mangiato è oggetto di piacere e di vittoria.

Una sventura qualsiasi, simulata come nei racconti di fate, è la forma fondamentale dell'intreccio. Dalla sventura e dalla reazione a essa nasce il soggetto, preferibilmente fantascientifico per una sorta di complicità o simbiosi tecnologica. I giochi contro gli invasori hanno come meta l'annientamento totale e l'unico potere che il giocatore riesce a esercitare è la pura sopravvivenza: ogni partita è sempre un "the game after". Ma non ci si lasci ingannare da scenari di demoniaca sottigliezza, dall'affollamento di cadaveri, dall'enfasi di una vocazione apocalittica.

Già il cinema e la Tv hanno dimostrato che spesso si mette in scena un disagio (catastrofi, invasioni, guerre stellari) per esorcizzare la paura di quel disagio.

SIMULATORI DI CIRCUITI DIGITALI

Sistemi di simulazione in grado di seguire praticamente tutte le fasi di sviluppo di un progetto elettronico.

Anche se sviluppati dapprima per aiutare la progettazione e il testing dei circuiti integrati su larga scala (i cosiddetti VLSI), esistono ormai sistemi di simulazione logica e circuitale in grado di seguire praticamente tutte le fasi di sviluppo di un progetto elettronico.

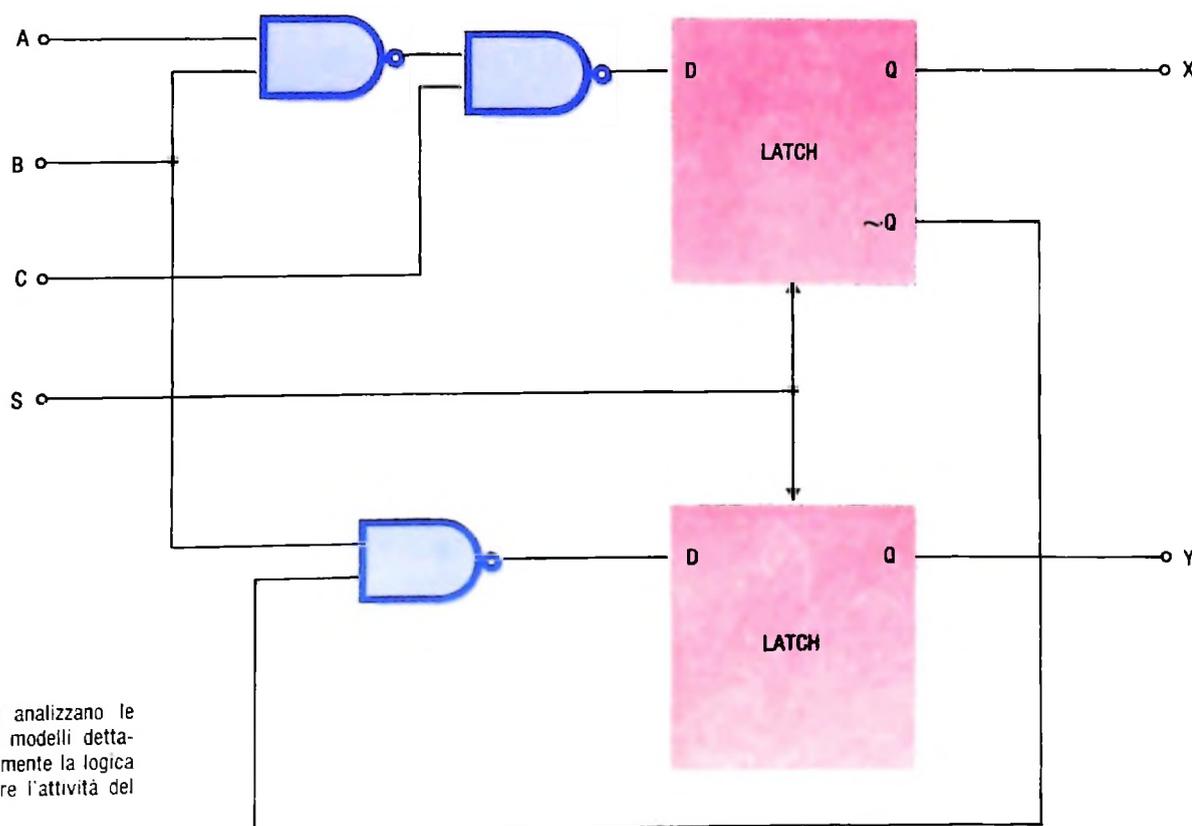
Anzi, con la disponibilità di più semplici interfacce utente, possono incominciare a rimpiazzare in alcuni casi il vero e proprio prototipo per quanto riguarda la verifica funzionale e delle effettive prestazioni.

Inoltre, con il miglioramento degli algoritmi di simulazione e dell'efficienza dell'hardware, anche l'analisi dei circuiti più complessi — opportunamente scomposta in fasi successive — può essere svolta con stazioni di lavoro CAE (Computer Aided Engineering), piuttosto che con calcolatori di grandi dimensioni.

Cambiano i metodi di progetto

È chiaro che la diffusione di questi simulatori porterà importanti innovazioni al classico metodo di lavoro dell'ingegnere elettronico. In primo luogo le nuove idee sono più facili da provare perché si può evitare di costruire il vero e proprio prototipo, almeno in tutti gli stadi intermedi del progetto. Inoltre, diversi ingegneri possono lavorare contemporaneamente sullo stesso progetto, o data base di progetti, cosa che non è possibile su un singolo prototipo (e che presenta anche vantaggi di maggiore uniformità aziendale).

È anche possibile, mentre non lo è con un prototipo, considerare i *casi peggiori*, cioè per esempio simulare il circuito usando i limiti più sfavorevoli specificati dal costruttore per i parametri dei singoli componenti.



I simulatori strutturali analizzano le grosse reti. Richiedono modelli dettagliati, e usano essenzialmente la logica booleana per determinare l'attività del circuito.

I simulatori consentono anche di "entrare" negli elementi di circuito, per analizzarne il comportamento, se questi sono stati modellati adeguatamente. È il caso dei circuiti con microprocessori, in cui è possibile, per esempio, ispezionare lo stato dei registri interni del singolo microprocessore, mentre viene simulato il circuito che lo utilizza.

Ma ancora più importante è il fatto che i simulatori, in qualche modo, riavvicinano gli ingegneri responsabili del testing con i progettisti. Usando i data base di simulazione è possibile infatti incominciare a sviluppare strategie di testing quando ancora il progetto non è terminato. Inoltre sfruttando la capacità del simulatore di analizzare anche circuiti non funzionanti, è più facile progettare efficaci punti di test sui circuiti stampati, nonché migliori test-pattern per la verifica automatica degli stessi in fase di produzione.

Classi di simulatori

Esistono simulatori circuitali, strutturali e comportamentali. Forse il più noto simulatore *circuitale* è SPICE, progettato all'Università di Berkeley (California). SPICE analizza l'attività circuitale di reti lineari di componenti discreti; questo livello di dettaglio generalmente non è richiesto dalle più grandi reti digitali, tuttavia è indispensabile nel progetto dei singoli elementi di circuito (da inserire poi in libreria).

I simulatori *strutturali* producono dati meno dettagliati rispetto ai simulatori circuitali, ma analizzano le grosse reti. Normalmente espandono tutti i modelli ad alto livello in rappresentazioni a livello di porte logiche e poi analizzano l'attività circuitale usando l'algebra booleana (confronta con la figura riportata nella pagina precedente).

Più complessi sono i simulatori *comportamentali* che non usano modelli funzionali basati su porte logiche, ma linguaggi di modellazione ad alto livello mediante i quali è possibile descrivere il comportamento di ampi "blocchi" circuitali co-

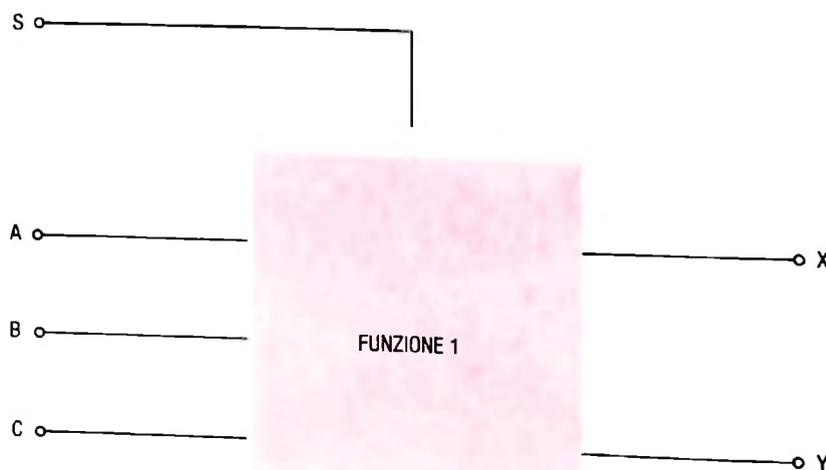
me si può vedere nella figura in basso. Quindi anche i circuiti microprogrammati diventano analizzabili, nonché i microprocessori stessi, troppo complessi per essere simulati a livello di porte logiche.

Con i simulatori comportamentali inoltre si può aggirare un ostacolo piuttosto comune, rappresentato dal fatto che i fabbricanti di circuiti integrati spesso non rendono disponibili tutti i dati necessari per arrivare al modello strutturale dei loro prodotti, e questo per evidenti ragioni di competitività. Sfortunatamente i simulatori comportamentali sono ancora poco diffusi in quanto molto complessi. Uno dei problemi è quello dell'interfacciamento con i sistemi strutturali, necessario perché, nella pratica, i grossi circuiti digitali sono costituiti da alcuni componenti principali (che possono essere descritti solo comportamentalmente) interconnessi però da reti di circuiti più semplici che devono essere descritte strutturalmente (figura della pagina a lato).

Gerarchia dei simulatori

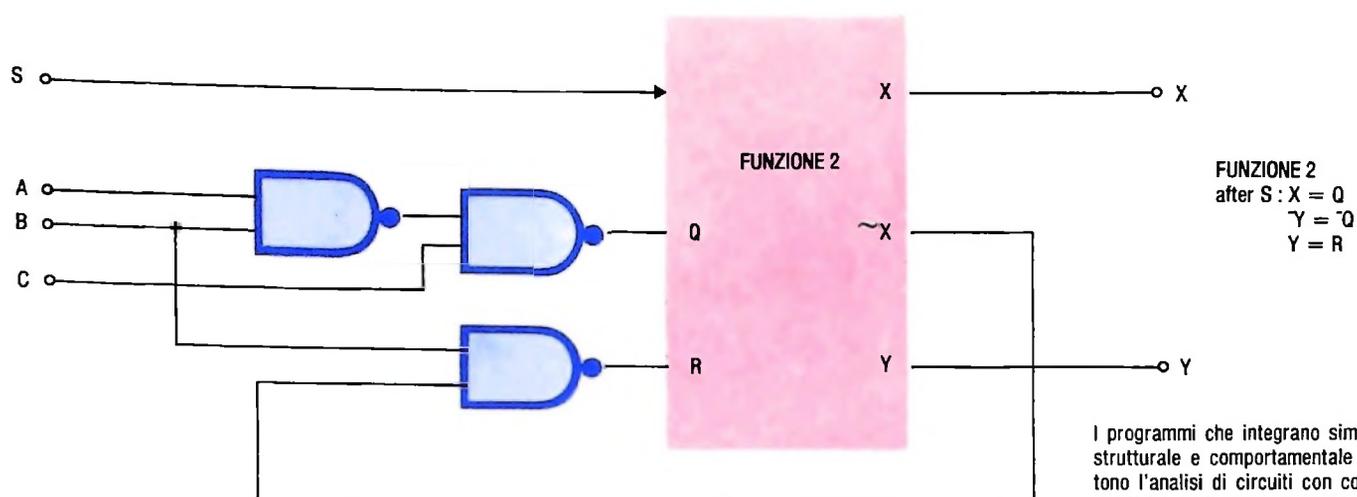
All'inizio di un progetto il simulatore comportamentale aiuta a definire le funzioni dei principali elementi del sistema. Queste funzioni a loro volta costituiscono l'insieme di caratteristiche cui le rappresentazioni a più basso livello devono conformarsi. Mentre si progetta il singolo blocco, i risultati della sua simulazione devono concordare con quanto specificato a livello comportamentale generale. Lo stesso vale successivamente per il livello circuitale.

Passando da un livello all'altro aumenta naturalmente il livello di dettaglio nell'analisi temporale e di propagazione dei ritardi: per esempio, l'analizzatore strutturale deve identificare le temporizzazioni incerte e gli stati logici sconosciuti, mentre l'analizzatore circuitale deve anche preoccuparsi dell'influenza dei carichi e delle capacità spurie sulla velocità delle singole porte. Contemporaneamente le dimensioni del-



I simulatori comportamentali usano linguaggi ad alto livello per rappresentare la funzione di rete. Le descrizioni a questo livello non richiedono modelli basati su porte logiche. Quindi anche i circuiti microprogrammati, nonché i microprocessori stessi, diventano analizzabili.

FUNZIONE 1
after S: $X = (A \cdot B) + C$
 $Y = X + B$



I programmi che integrano simulazione strutturale e comportamentale permettono l'analisi di circuiti con comportamenti integrati su piccola scala (SSI) e grande scala (LSI). Inoltre permettono la simulazione di sistemi in cui alcuni blocchi, ancora in progettazione, sono provvisoriamente descritti solo al livello comportamentale.

la rete da analizzare però diminuiscono e quindi la complessità totale della simulazione rimane pressoché costante e alla portata di calcolatori di medie dimensioni, come dicevamo all'inizio dell'articolo.

Il ruolo dei modelli

La capacità di un simulatore di rivelare i più sottili problemi di temporizzazione e strutturali dipende in larga misura dalle caratteristiche del modello dei componenti che utilizza. I simulatori elaborano solo astrazioni dei circuiti fisici, cosicché analisi dettagliate richiedono trasformazioni precise dei parametri di circuito in funzioni matematiche.

I simulatori circuitali, come SPICE, usano modelli molto astratti e quindi permettono semplici rappresentazioni dei componenti. Per esempio, un amplificatore operazionale (di per sé già una generalizzazione degli amplificatori usati nei circuiti analogici) può essere modellato facendo uso solo di funzioni matematiche che rappresentano resistenze, capacità e sorgenti di corrente.

Al livello successivo i modelli sono basati su caratteristiche circuitali meno facili da definire. Con il crescere della complessità circuitale il numero delle diverse interazioni di condizioni fisiche che influenzano l'attività della rete aumenta, e quindi i modelli diventano più involuti. Entrano in gioco poi anche le limitazioni dovute alla potenza di calcolo di cui si dispone.

I modelli di temporizzazione non solo devono rappresentare accuratamente la funzione dei componenti, ma devono anche essere abbastanza dettagliati da permettere al simulatore che li usa di controllare che gli elementi di rete siano utilizzati nel rispetto di certi limiti critici.

Gli stessi modelli devono anche fornire abbastanza dati comportamentali da consentire il riconoscimento e la propaga-

zione degli stati instabili e sconosciuti.

Infine, devono tenere conto delle "porte fantasma" che si creano, per esempio, collegando insieme circuiti tristabili o a *collettore aperto* (se si collegano più segnali a collettore aperto in un singolo nodo si crea, elettricamente parlando, una porta AND del tutto equivalente ad una normale altra porta AND attiva).

Scald

Per modellare reti complesse è utile che il simulatore supporti un linguaggio di progettazione "strutturalmente simile" al progetto da realizzare. Scald (Structured Computer Aided Logic Design) è un simulatore che appunto è basato su un linguaggio di descrizione delle reti di tipo gerarchico, che riconosce strutture ad alto livello come i *bus* (le strutture di interconnessione delle varie schede che compongono un sistema digitale), o i registri. Scald è stato sviluppato ai laboratori Lawrence Livermore e usato per la progettazione del supercomputer della marina americana S1. Si serve del parallelismo inerente al progetto per accelerare l'analisi delle reti e permette la modellazione diretta di ampi *data path*. Per esempio in un tipico mainframe in media ci sono dieci processi simultanei su ogni *data path*, e un programma come Scald è quindi in media dieci volte più rapido di un normale simulatore.

Per accelerare ulteriormente l'analisi, il simulatore Scald si preoccupa separatamente della funzione di rete e delle caratteristiche di temporizzazione, mantenendo l'analisi di queste ultime (analisi del caso peggiore) indipendente da specifici stimoli d'entrata.

I simulatori che controllano il comportamento della rete usando un insieme predefinito di valori sulle entrate producono solo la temporizzazione corrispondente agli stati creati

da quei valori. Al contrario, un analizzatore di temporizzazione algoritmico e indipendente dalle entrate può verificare tutti i possibili stati, e quindi assicurare la correttezza della rete in ogni condizione.

Helix

Helix è forse il più sofisticato dei simulatori commerciali attualmente disponibili.

Anzi, più che di simulatore in questo caso si deve parlare di uno strumento per generare simulatori.

Permette la modellazione di elementi in modo gerarchico, e usa un linguaggio per la descrizione del comportamento circuitale basato sul Pascal. Helix non usa però primitive funzionali per descrivere i modelli a più alto livello come avviene con gli altri simulatori, ma solo costrutti ottimizzati per descrivere funzioni circuitali concorrenti (parallele) e relazioni temporali.

In questo modo, per esempio, non esistono limitazioni ai soliti due o tre livelli logici per i componenti, perché se ne possono definire in numero arbitrario. E anche se Helix è stato progettato per la simulazione discreta, non analogica, è chiaro che è possibile approssimare anche componenti analogiche del circuito, magari con l'aiuto di Spice per gli algoritmi rilevanti.

Perché un simulatore possa essere usato efficientemente è necessario che abbia una buona interfaccia utente, cioè che permetta all'ingegnere di esprimere i dati e di leggere i risultati in modo per lui naturale. Helix usa un driver grafico che accetta descrizioni di reti a livello di schema (quindi non richiede la lista dei componenti e delle loro interconnessioni), e produce inoltre rappresentazioni grafiche ad ogni livello dell'analisi.

Accuratezza della simulazione

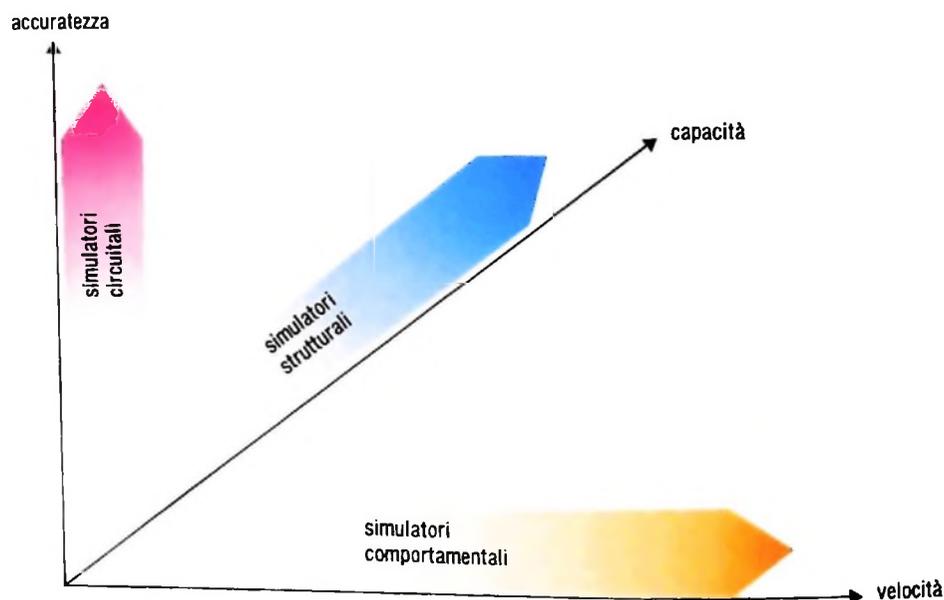
È chiaro che il fattore principale di accuratezza della simulazione è rappresentato dalla precisione e completezza del modello dei componenti circuitali utilizzati. Tuttavia maggiore è la complessità e precisione dei modelli, maggiore deve essere l'efficienza degli algoritmi di simulazione impiegati, così come maggiore deve essere la velocità dell'hardware. Si rende quindi generalmente necessario un compromesso fra accuratezza, velocità e dimensioni della macchina. Anche le dimensioni dei data base circuitali devono per forza essere limitate. I simulatori comportamentali normalmente preferiscono massimizzare la velocità di esecuzione a spese della risoluzione dei risultati: agli stadi iniziali del progetto è più importante avere velocemente un quadro globale piuttosto che i dettagli dell'analisi.

I simulatori strutturali, che sono cruciali specialmente in vista della progettazione della fase di testing del sistema, hanno particolarmente bisogno dei dettagli di temporizzazione del circuito e quindi sono più lenti e richiedono maggiori capacità di calcolo, ma fortunatamente non devono normalmente essere usati interattivamente come i simulatori comportamentali.

Infine i simulatori circuitali richiedono capaci data base di elementi circuitali, ma lavorando di solito su reti piuttosto semplici (di solito meno di 500 transistor), non hanno bisogno di computer molto veloci (figura in basso).

Un discorso a parte meritano i simulatori come Helix che, essendo strutturati gerarchicamente, possono avere più modelli per ogni singolo elemento circuitale, a diverso livello di dettaglio. Il progettista può quindi scegliere la libreria di modelli da utilizzare in funzione della velocità di risposta richiesta, delle dimensioni delle reti da analizzare e della capacità di calcolo a disposizione.

I simulatori circuitali, lavorando di solito su reti piuttosto semplici, non richiedono computer molto veloci. L'accuratezza di simulazione dipende dalla velocità degli algoritmi e dalla capacità di calcolo dell'hardware. I simulatori ottimizzano una delle caratteristiche a scapito delle altre.



— UN NUOVO MODO DI USARE LA BANCA. —

Conto corrente più

TANTI PENSIERI
IN MENO CON IL CONTO
CORRENTE "PIÙ"
DEL BANCO DI ROMA.

Essere cliente del Banco di Roma vuol dire anche essere titolari del conto corrente "più". Un conto corrente più rapido: perché già nella maggior parte delle nostre filiali trovate gli operatori di sportello che vi evitano le doppie file.

Più comodo, perché potete delegare a noi tutti i vostri pagamenti ricorrenti: dai mutui all'affitto, dalle utenze alle imposte.

Più pratico, perché consente l'utilizzo del sistema di prelievo automatico Bancomat e l'ottenimento della carta di credito.

Inoltre un servizio utilissimo, soprattutto per imprenditori e commercianti denominato "esito incassi", consente di avere comunicazione dell'eventuale insolvenza entro solo cinque giorni dalla scadenza. Una opportunità veramente speciale.

Più sicuro, perché con una minima spesa potrete assicurarvi contro furti e scippi mentre vi recate in banca o ne uscite.

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.



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 micro-plotter 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

Olivetti Personal Computer è un marchio registrato di Olivetti Personal Computer S.p.A. - Via Mecenate, 72
 Nome e Cognome _____
 Via _____
 CAP/CITÀ _____
 TELEFONO _____