

**PASSATO
PRESENTE
e
FUTURO**

DELLE MACCHINE-GIOCO

LA TECNOLOGIA DELLE MACCHINE-GIOCO

Questo libretto non solo vi illustrerà le prospettive future delle macchine-gioco ed i problemi insiti nel produrle. Vi fornirà anche le specifiche tecniche elaborate dai nostri esperti che dovranno essere, secondo il loro parere, alla base delle macchine-gioco degli anni '90. Pronti?...

- IL GRANDE FRATELLO**Pag. 2
Come è cominciata - i principi di progettazione ed i problemi dei primi computer dedicati ai giochi...
- LARGO È BELLO**Pag. 5
I 16-bit sono veramente la risposta giusta all'aumento della potenza dei giochi?
- LA MACCHINA-COCKTAIL**Pag. 7
Un nuovo modo di affrontare il problema: macchine che dividono il lavoro su più processori.
- PROCESSORI FAI DA TE**Pag. 11
Se un 68000 non basta, perché non progettarsi un processore fatto su misura?
- ELABORAZIONE PARALLELA**Pag. 14
I processori fatti su misura possono essere molto potenti, ma chi vorrà programmarli?
- VIDEO INTELLIGENTE**Pag. 16
Come presentare mondi immaginari sullo schermo - e come farli scorrere
- GLI SPRITE DEGLI ANNI '90**Pag. 18
L'Amiga può essere un mago nella gestione degli sprite, ma che utilità può avere ciò nei giochi dell'ultima generazione?
- UNA NUOVA ERA VIDEO**Pag. 20
Come è possibile fare in modo che i giochi per computer abbiano una definizione delle immagini pari a quella della TV.
- IL FUTURO**Pag. 24
Il nostro progetto di un computer per gli anni '90
- LE SPECIFICHE**Ill di copertina
Dieci punti per un computer degli anni '90

IL GRANDE FRATELLO

Nel corso degli ultimi cinque anni c'è stata una rivoluzione nella progettazione dei computer per videogiochi. Prima di guardare avanti è necessario, comunque, dare uno sguardo al passato. Inizieremo esaminando i primi home computer per compararli con quelli odierni.

Agli albori della storia, diciamo verso il 1984, il modello dominante di home computer era quello che avrebbe potuto essere chiamato "Grande Fratello" (un nome appropriato, visto l'anno) a causa della sua dipendenza da un unico, potente (o non troppo potente) processore centrale o CPU (Central Processing Unit). Ogni operazione eseguita da uno Spectrum o da un Vic 20 viene in realtà eseguita dalla CPU. Diciamo, per esempio, che volete far apparire uno sprite sullo schermo: la CPU copia i dati dell'immagine nella memoria dello schermo. Volete far suonare una musicchetta al computer? La CPU invia il motivetto al processore audio, una nota alla volta. Volete eseguire dei calcoli complessi, diciamo, di geometria tridimensionale? La CPU esegue i calcoli.

Dal momento che in questo genere di computer la CPU esegue (più o meno) tutto, la sua potenza è il fattore fondamentale per le prestazioni della macchina. L'unico modo di aumentarne la potenza consiste nel renderlo più rapido, aumentando il numero di operazioni che la CPU è in grado di eseguire in un secondo, o, alternativamente, nell'aumentare la "lunghezza" degli operandi, cioè delle combinazioni di bit, facendo in modo che il computer esegua più calcoli per ciascuna operazione. Di questi due metodi, il secondo era quello che sembrava avere le prospettive migliori nel lontano 1984. Cominciavano infatti ad essere disponibili i primi processori a 16 e perfino a 32-bit, che erano destinati a sostituire l'antiquato processore ad 8-bit che aveva dato inizio, con la sua apparizione, al boom del computer.

PIU' IN FRETTA, PIU' IN FRETTA!

Potete pensare alla velocità e alla lunghezza dei dati come ad un'autostrada. Se volete che essa sia capace di permettere un maggiore flusso di traffico potete aumentare la massima velocità consentita, ma non potreste comunque superare un certo limite: una volta che il limite supera i 150 Km orari le cose iniziano a farsi difficili ed a 200 Km orari l'autostrada non sarebbe più funzionale.

Ciò che è valido per le autostrade, lo è anche per i processori. Esiste

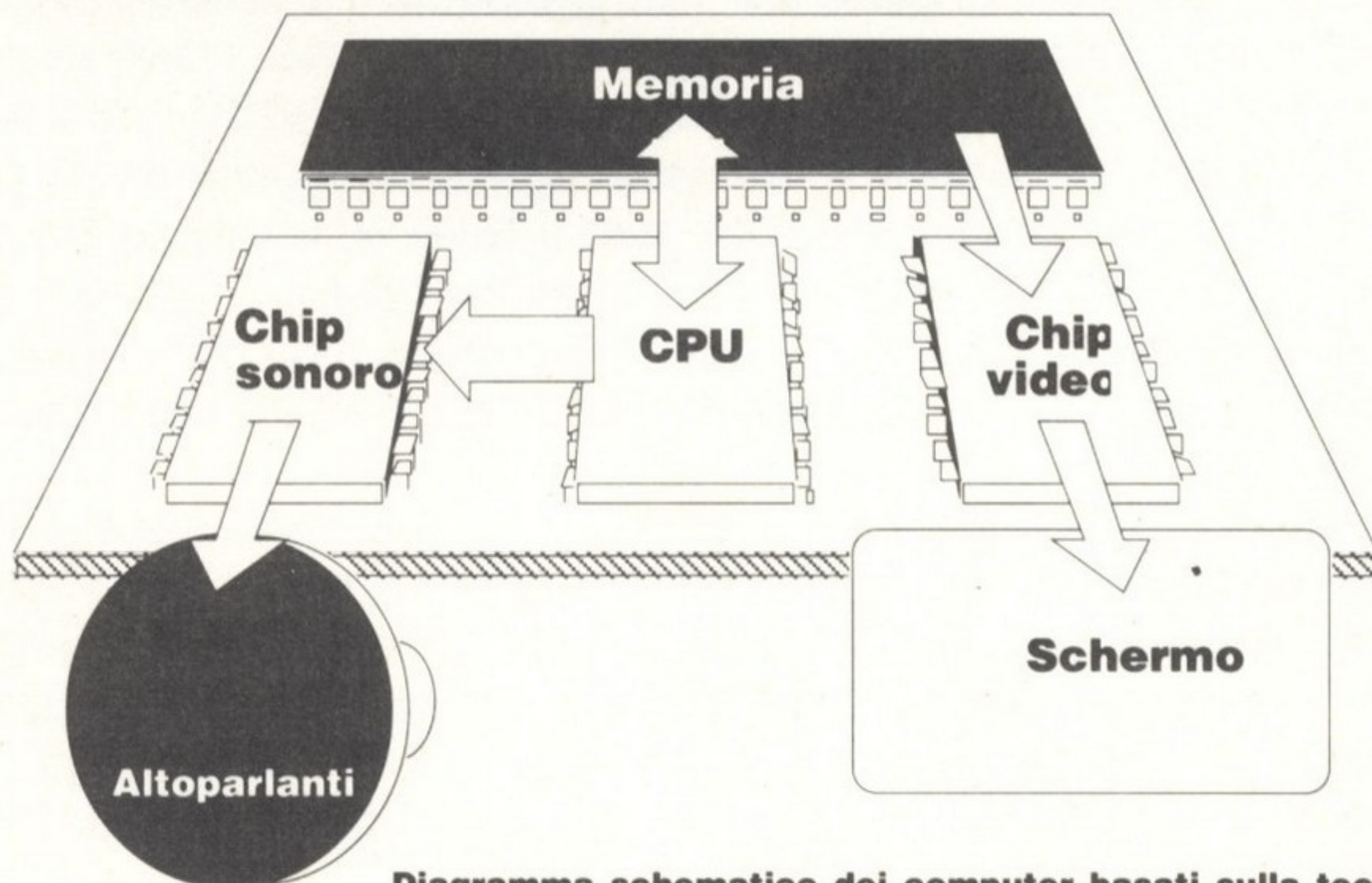


Diagramma schematico dei computer basati sulla teoria del "Grande Fratello", con la CPU che dirige tutte le operazioni. Le principali CPU utilizzate nei primi anni ottanta erano lo Z80 (Spectrum, Amstrad, Enterprise) ed il 6502 (C64, BBC, Oric).

un limite teorico alla velocità operativa di un processore - i dati non possono essere elaborati ad una velocità superiore a quella della luce - ma esiste anche un limite pratico e ben più importante. Un processore può operare al massimo alla velocità consentita dalla memoria disponibile. Ogni volta che il processore ha bisogno di un'istruzione, deve leggere una locazione di memoria. Se le istruzioni gli impongono di leggere o scrivere altre locazioni - ed in un gioco queste istruzioni impegnano gran parte del tempo del processore - allora deve accedere più volte alla memoria.

Se acquistate un veloce PC AT compatibile, la sua CPU - un 80286 - potrebbe operare a 16MHz o più e (a questa velocità) accedere alla memoria 8 milioni di volte al secondo. Una velocità così elevata può provocare dei problemi (vedi riquadro a pag. 6), così il modo solito di procedere consiste nell'utilizzare una memoria cache - un'area di "deposito" temporanea - cioè una memoria statica ad accesso rapido o SRAM (Static Random Access Memory). La memoria SRAM contiene, ad esempio, i 16K di locazioni di memoria ai quali la CPU deve accedere più spesso, così diminuisce la necessità di servirsi della più lenta memoria dinamica o DRAM.

La memoria cache, comunque, non va bene nel caso il programma sia un gioco. Un programma per PC può passare il 90% del suo tempo acce-

dendo sempre alle stesse poche locazioni, ma un gioco necessita in media di accedere regolarmente a 100 o più K di memoria. Tanto per cominciare, ci sono lo schermo visibile (quello che state guardando) e lo schermo nascosto (quello che il computer sta preparandosi a mostrarvi) che portano via circa 30K ciascuno. Quindi ci sono gli sprite, i dati relativi al sonoro e le tabelle che riportano il comportamento di ciascun oggetto. In aggiunta a tutto ciò c'è il codice vero e proprio: nei giochi questo tende ad essere particolarmente prolisso, perché generalmente girano più velocemente. (Una classica tecnica impiegata dall'Amstrad utilizzava 80 identiche istruzioni in fila per eseguire solo parte di un dato compito e ci sono paralleli di ciò in molte macchine).

L'unico modo di superare la strettoia della memoria disponibile è di dotare la macchina di un processore SRAM, ma ai prezzi odierni ciò sarebbe incredibilmente costoso. Con un pizzico di fortuna i prezzi dovrebbero iniziare a crollare verso il 1992 o il 1993, grazie ai miglioramenti nei processi di produzione. Il risultato, però, sarà semplicemente di avere memoria più a buon mercato, perché la velocità sarà la stessa dei processori centrali di oggi.

DI COSA È FATTA LA MEMORIA...

Al momento esiste un solo tipo di memoria sufficientemente a buon mercato da poter essere usata in grandi quantità. È la Memoria Dinamica ad Accesso Casuale o memoria DRAM (Dynamic Random Access Memory); purtroppo non è molto veloce. In una memoria DRAM il limite massimo di lettura o scrittura, oltre il quale si possono avere risultati inaffidabili, è 5 milioni di volte al secondo. A prima vista potrà sembrare un limite elevato, ma in pratica le cose sono ben diverse: un 68000 con una frequenza di 8MHz come quello dell'Atari ST può accedere alla memoria fino a 2 milioni di volte al secondo, quindi è chiaro che il limite massimo non è così elevato. Si può raddoppiare la velocità del processore, ma non oltre, poiché possono nascere dei problemi. Non serve a nulla avere un processore capace di accedere alla memoria 8 milioni di volte al secondo se la memoria non può stargli dietro. (N.B. Questa "velocità d'accesso alla memoria" non è da confondere con la velocità del processore. La maggior parte dei processori accede alla memoria solo ogni due operazioni o, come il 68000, ogni quattro).

LARGO È BELLO

Aumentare la velocità non è l'unico modo per far eseguire ad un processore una mole maggiore di lavoro. Si può anche provare ad aumentare la lunghezza degli operandi (per restare in tema di autostrade, da ora in avanti useremo larghezza invece di lunghezza). Torniamo per un momento alla nostra analogia con il traffico di un'autostrada e vediamo come funziona questa seconda soluzione.

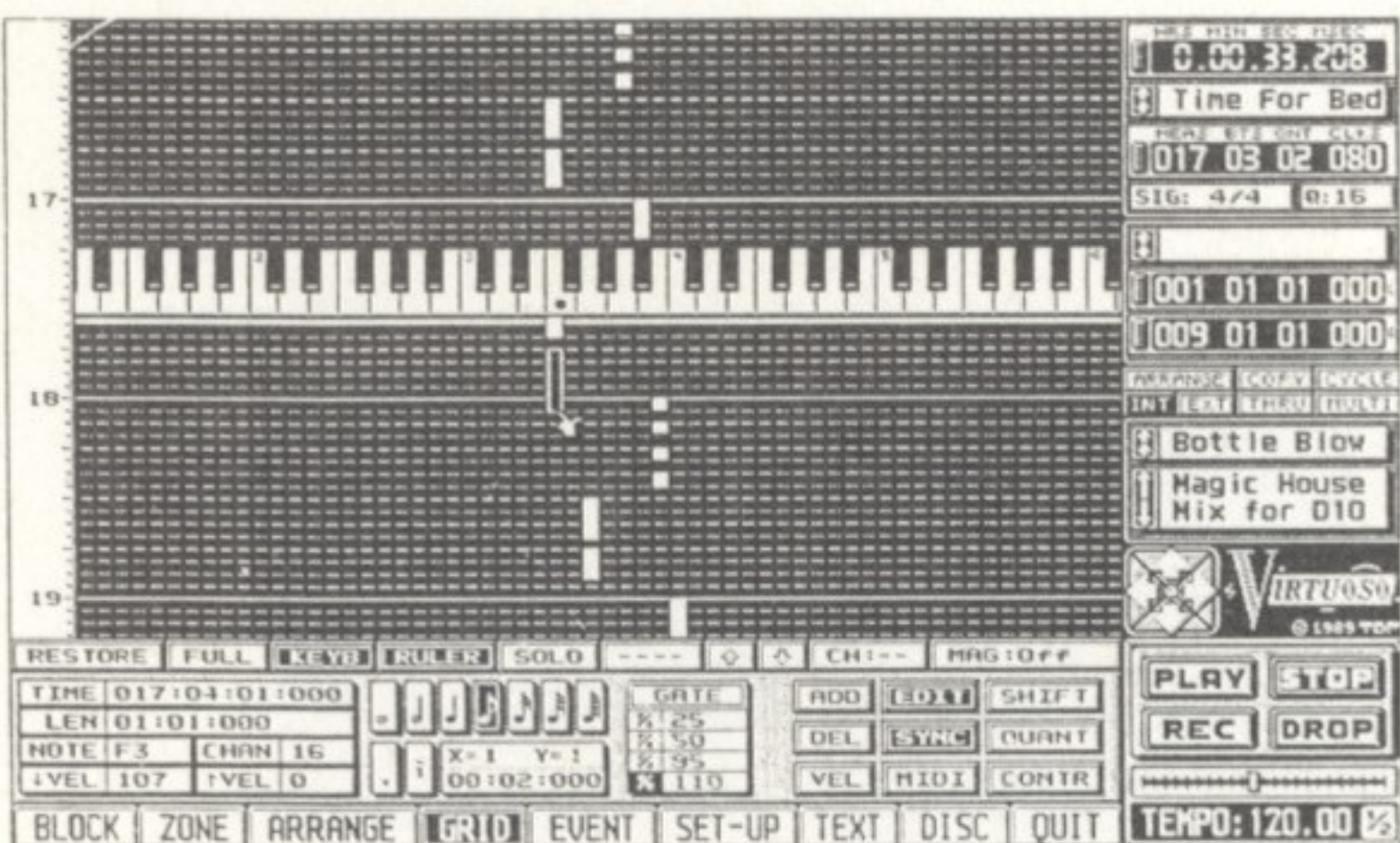
Se una strada non riesce a far fronte alla mole di traffico ed è già stato aumentato il limite di velocità al livello massimo di sicurezza, una buona alternativa è quella di aggiungere qualche corsia extra, trasformandola in un'autostrada a cinque o magari sei corsie. In questo caso non è detto che le automobili si muoveranno più velocemente, ma è certo che un numero maggiore di esse potrà percorrere ogni tratto dell'autostrada, cosicché questa permetterà il passaggio di una maggiore mole di traffico.

Come non potete aumentare all'infinito la velocità di un computer, così non potete aumentare all'infinito la sua "larghezza". Aumentarla oltre un certo limite, in verità, non è di grande utilità.

Un'autostrada a 100 corsie è difficile e costosa da costruire e non è detto che svolga il suo compito meglio di un'autostrada a dieci corsie. Le probabilità che un giorno cento macchine debbano percorrere l'autostrada fianco a fianco sono quasi nulle, quindi perché perdere tempo e denaro a costruirla?

Lo stesso vale per i processori. I processori degli home computer avevano originariamente una larghezza di 8-bit - potevano cioè elaborare i dati solo a 8-bit alla volta. Come potete intuire, 8-bit non sono in realtà una grande mole di dati.

Il MIDI è uno standard di comunicazione tra computer e strumenti musicali e tutti i codici in standard MIDI sono a 8-bit (numeri binari ad otto cifre variabili tra valori decimali di 0 e 255). Per questa ragione, come nel caso degli scacchi (vedi pag. 8), i programmi MIDI possono non sfruttare al massimo le opportunità offerte da un sistema a 16-bit.



Non bastano per immagazzinare i puntatori interni di cui ha bisogno un programma né possono contenere le numerose coordinate necessarie al funzionamento dei giochi di oggi. A causa di ciò, i processori a 8-bit devono trattare queste quantità di dati a pezzi, necessitando di numerosi accessi alla memoria per leggere o scrivere ciascuna di esse. L'avvento dei primi processori a 16-bit, come l'8086 (quello degli attuali PC compatibili) ed il 68000 (Amiga e ST) fu, quindi, un grande passo avanti.

BIT SPRECATI?

Dal momento che nei programmi-gioco, molte quantità di dati avevano già una larghezza di 16-bit, questi nuovi processori potevano, da subito, elaborare effettivamente i dati a velocità doppia nella maggior parte dei compiti a cui erano preposti, e questo prima ancora di considerare l'aumento della velocità operativa reso possibile dai nuovi processori. Vale la pena notare che l'espressione "nella maggior parte dei compiti" è approssimativa. Alcuni generi di giochi non si avvalgono molto di dati a 16-bit. Un'esempio classico sono i programmi di scacchi, che hanno una larghezza naturale di 8-bit (a causa della forma della scacchiera). I programmi di scacchi trascorrono gran parte del loro tempo trattando dati a 8-bit, così la larghezza extra è ampiamente sprecata.

Tali esempi di spreco sono però rari su macchine a 16-bit, ma diventano di gran lunga più comuni quando si passa a 32-bit. Prendete l'Archimedes, per esempio: il suo processore ARM può leggere o

IL COSTO DELLA LARGHEZZA...

Costa aumentare la lunghezza degli operandi di un processore. Se raddoppiate il numero di bit che esso può elaborare, raddoppiate il numero di "piedini" di metallo di cui ha bisogno ed il numero di "linee" di rame che dovete stampare sulla scheda. Attualmente i processori a 32-bit ed i sistemi ad essi connessi sono i più ampi tra quelli che è possibile produrre in un'economia di mercato. Nonostante sia possibile sviluppare tecniche di manifattura per la produzione di processori a 64 o 128-bit, è difficile credere che qualcuno si preoccuperà di farlo. Il salto di qualità tra 16 e 32-bit è già molto basso ed andare oltre porterebbe benefici ancora meno sensibili.

scrivere numeri fino a 4 miliardi con un solo accesso alla memoria. Nonostante ci siano impieghi importanti per larghezze di queste dimensioni - specialmente nella gestione di sprite o nel riempimento di aree in immagini in 3D solido - ugualmente ci sono un mucchio di occasioni in cui questi bit supplementari non sono affatto necessari.

UN NUOVO APPROCCIO

Ciò di cui si ha realmente bisogno oggi è un nuovo approccio al problema della produzione di una macchina potente - e questo è quanto cercheremo di esaminare ora. Vedete, il vero problema del tipico computer stile "1984" è che ogni cosa era centralizzata. È come se ogni cosa che accade in Italia debba accadere a Milano. Volete mangiare? Volete bere? Volete lavorare o andare a vedere un film? Andate a Milano. Vista la questione in questo modo, la prossima volta che siete bloccati dal traffico alla periferia di Milano potreste pensare che l'errore di base in realtà non ha nulla a che vedere con il traffico. Il problema è che se una città è costretta a soddisfare i bisogni di quasi 60 milioni di persone, inevitabilmente le cose non funzioneranno. Nel microcosmo del computer, proprio come nella vita reale, abbiamo bisogno di un sistema articolato su più "città" - più luoghi in cui viene svolto il lavoro e succedono cose. Invece di un unico processore "Grande Fratello" che sovrintende su tutto, potremmo tentare di dividere equamente il lavoro. Ed è a questo punto che entrano in scena i computer di oggi ...

LA MACCHINA-COCKTAIL

Nonostante la loro apparente potenza, macchine come l'ST, e l'Archimedes sono ancora basate in gran parte sul vecchio principio del "Grande Fratello", con un processore centrale che si occupa di tutto. Queste macchine sono però un ramo della famiglia dei computer destinato presto ad estinguersi. L'ST tiene duro grazie alla potenza del 68000 e all'attuale stato della programmazione dei giochi, che sta ancora trovando i modi per superare i limiti imposti dagli 8-bit. L'Archimedes impiega la tecnologia RISC (vedi il prossimo ca-

pitolo) per spingere al massimo il concetto di "Grande Fratello", grazie ad una CPU ultra-potente, ma le macchine del futuro dovrebbero essere caratterizzate da una CPU meno potente, accompagnata da diversi processori specializzati, ai quali verranno affidati i compiti più gravosi.

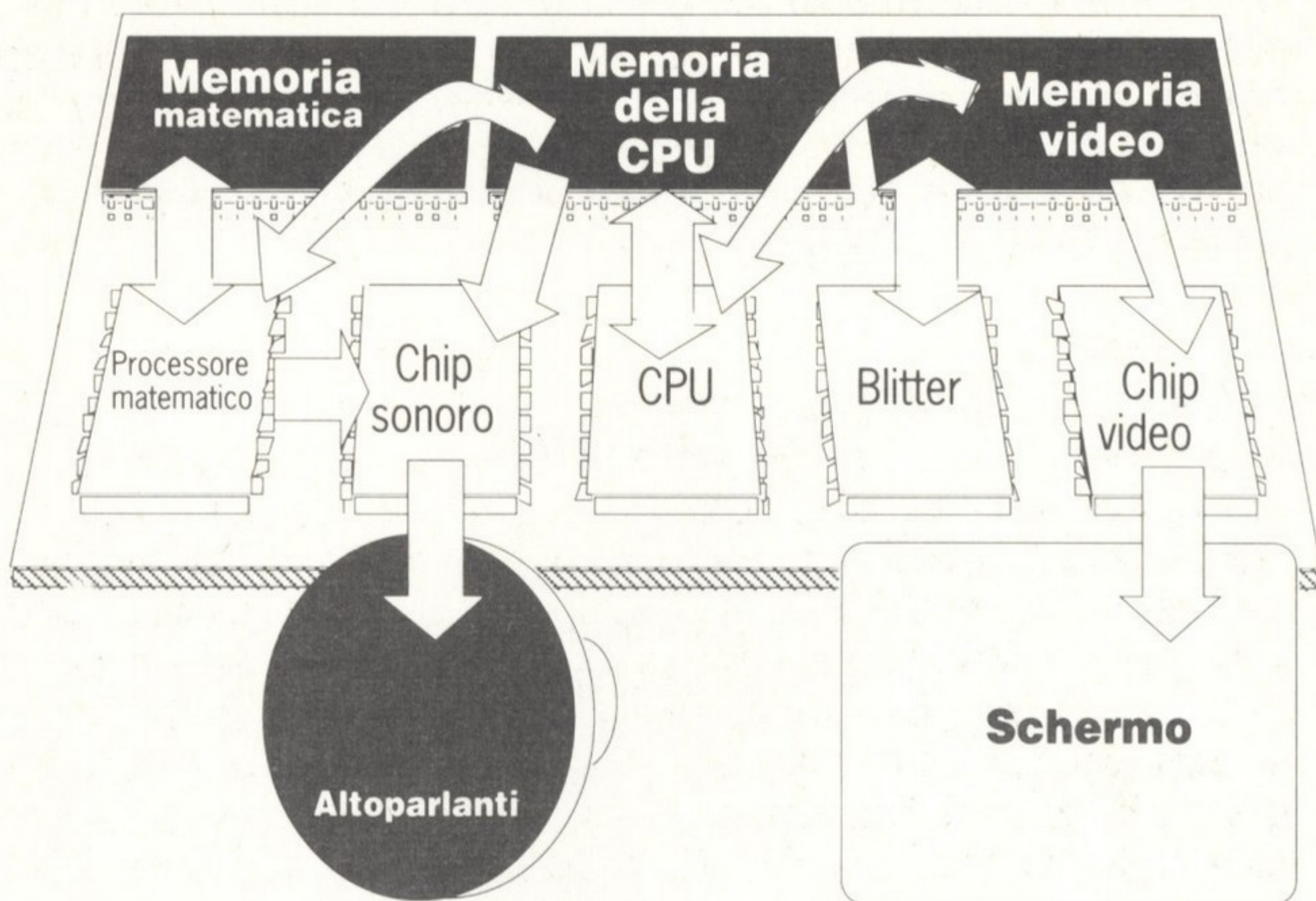
Le moderne macchine-gioco rappresentano un grosso passo avanti per diverse ragioni, particolarmente per i tre processori separati (CPU, *blitter* e processore matematico) che eseguono il lavoro di elaborazione. Il *blitter* ed il processore matematico sono specializzati nell'eseguire quei compiti che la CPU esegue lentamente. Dal momento che sono specializzati, possono eseguire il lavoro più velocemente, e a causa del modo in cui è strutturata la memoria, possono lavorare mentre la CPU è impegnata a fare tutt'altro.

Notate anche come il processore audio abbia finalmente ciò che il processore video ha sempre avuto: cioè la possibilità di leggere i propri dati direttamente dalla memoria, piuttosto che essere "imboccato" dalla CPU. Nonostante non ci sia una particolare "area di memoria audio", il processore audio può leggere melodie o forme d'onda preparate in precedenza, così come il processore video visualizza immagini pre-definite. Questo significa che la musica può essere indipendente dalle azioni che si svolgono sullo schermo e che la CPU può attivare con una singola istruzione effetti sonori molto complessi.

Il *blitter* si incontra sempre più comunemente nelle schede circuitali ed oggi, grazie anche alla pubblicità dovuta all'Amiga, è un termine sulla bocca di tutti. Il nome è una contrazione dell'espressione *Block Image Transferrer* (Trasferitore di Immagini a Blocchi) e ciò è esattamente il suo compito - trasferire blocchi di immagini, rettangoli di dati grafici, da una locazione all'altra. Un *blitter* di media capacità può ritagliare via i bordi superflui, così da trasferire anche immagini non rettangolari, e può eseguire semplici compiti "artistici" come riempire rettangoli con colori pieni o tracciare linee rette. Qualsiasi CPU potrebbe svolgere questo genere di compiti, ma il *blitter* ha un grosso vantaggio: è molto più veloce.

CAVALLI DA CORSA

La verità è che la maggior parte delle CPU sono progettate da ricercatori che lavorano nel campo dell'elettronica digitale o da loro equivalenti nel campo della scienza informatica, non da programmatori di giochi. Non c'è quindi molto da sorprendersi se una normale CPU è



Nel campo dei moderni home computer, l'Amiga rappresenta una tipica "macchina-cocktail" della prima generazione, con i suoi PAD (tre processori specializzati soprannominati Paula, Agnes e Denise) che sovrintendono alle operazioni di memoria e di schermo. Qui sopra potete vedere un diagramma che rappresenta una tipica "macchina-cocktail" - notate come il diagramma sia molto più complesso, con due nuovi processori che aggiungono al sistema la loro potenza di elaborazione.

molto rapida a manipolare gli stack (non molto utile in uno sparattutto) o a commutare tra un processo e l'altro (idem), ma si trova col fiato corto quando deve spostare dei pixel da una parte all'altra dello schermo. I *blitter* sono invece l'ideale, perché non hanno altro compito che muovere i pixel, quindi possono specializzarsi a farlo.

Nel loro campo, i processori matematici sono l'ideale esattamente per la stessa ragione. Questi processori specializzati nelle operazioni matematiche sono in realtà molto più vecchi dei *blitter*. Il processore matematico Intel 8087 è servito a questo scopo nei PC fin dai primi anni ottanta - ma per anni i processori matematici sono stati impegnati esclusivamente per software scientifico o finanziario. Tutto ciò sta per cambiare, grazie all'innovativo progetto della Flare

per il Konix. Il Processore di Segnale Digitale o DSP (Digital Signal Processor) del Konix è infatti un potente elaboratore matematico capace di elaborare un flusso impressionante di dati al secondo. Essendo specializzato in calcoli matematici, questo processore può eseguire calcoli di geometria tridimensionale o di generazione dei frattali più velocemente di qualsiasi altra CPU oggi presente sul mercato. C'è da scommettere che, come il *blitter* dell'Amiga, questo piccolo congegno darà origine nei prossimi anni a una vasta progenie di imitatori.

CHE BESTIONI!

La Psygnosis è stata la prima casa di software del Regno Unito a commercializzare prodotti per Atari ST, ma col passare del tempo si sono concentrati sempre più sull'Amiga. Attualmente si può dire che producono esclusivamente per il computer della Commodore, mentre per ST realizzano solo materiale su commissione o conversioni di giochi. Ciò è dovuto alla natura di "macchina-cocktail" dell'Amiga, che permette un grande controllo sugli aspetti grafici e sonori dei giochi della Psygnosis. Nella foto potete vedere una schermata del loro gioco *Shadow of the Beast* (recensito su K11, pag. 44-45), che si avvale di numerosi effetti grafici e sonori ottenuti con processi intensivi di elaborazione, tra cui 13 livelli di scorrimento parallattico nelle scene all'aperto che si muovono alla velocità di 50 quadri al secondo, mentre grossi sprite sono manipolati sullo schermo col sottofondo di una colonna sonora campionata. Nonostante l'Atari ST e l'Amiga abbiano la stessa CPU, un Motorola 68000, quella dell'Atari sarebbe travolta se tentasse di svolgere un simile compito, mentre il 68000 dell'Amiga può chiamare in suo aiuto i processori *custom* della macchina.



PROCESSORI FAI DA TE

La cosa più importante dei processori come il *blitter* dell'Amiga o il DSP della console Konix è che sono fatti su misura (o all'inglese *custom-made*). I progettisti di oggi non sono limitati ai soli processori disponibili in commercio, ma possono crearsi i processori che gli servono in base ai loro bisogni. Un *blitter* è piuttosto facile da produrre, perché i compiti che esso deve eseguire sono piuttosto semplici: basta un po' di "forza bruta". I processori matematici non sono così facili da produrre, ma oggi è possibile produrre intere CPU *custom*. L'unico problema diventa, a questo punto, il costo-orario del progettista, dato che le moderne tecniche elettroniche permettono di produrre il prototipo di un processore con qualche milione di lire. Un costo davvero economico.

Nei prossimi anni il più grosso compito per i progettisti di home computer sarà quello di produrre potenti processori *custom* sempre più adattati alle necessità dei giochi. Per fare ciò è necessario prestare maggiore attenzione a quello che dicono i programmatori. Al momento, la necessità più pressante è quella di un *blitter* più potente: il *blitter* dell'Amiga richiede troppo tempo per "scaldarsi" - e il gioco non vale la candela quando si tratta di piccoli sprite - e, come il *blitter* del Konix, è di scarsa utilità nei moderni giochi in 3D. I programmatori hanno disperatamente bisogno di un processore che possa disegnare le figure impiegate in giochi come *Carrier Command* e *Virus*. I triangoli andrebbero benissimo - perché grazie ad essi si possono costruire sagome più complesse - ma per il momento non si sa di nessuno che stia progettando un tale processore, che potremmo chiamare *tritter*. L'ideale sarebbe un *tritter* ad alta velocità, combinato con un processore



Il Multi-System della Konix: in questa "macchina-cocktail" dell'ultima generazione, la CPU serve solo per le funzioni fondamentali. Originariamente era stato scelto uno Z80, ma successivamente, per esigenze commerciali, è stato preferito impiegare un 8086 a 16-bit. La morale è che in una "macchina-cocktail" la scelta della CPU non è di fondamentale importanza.

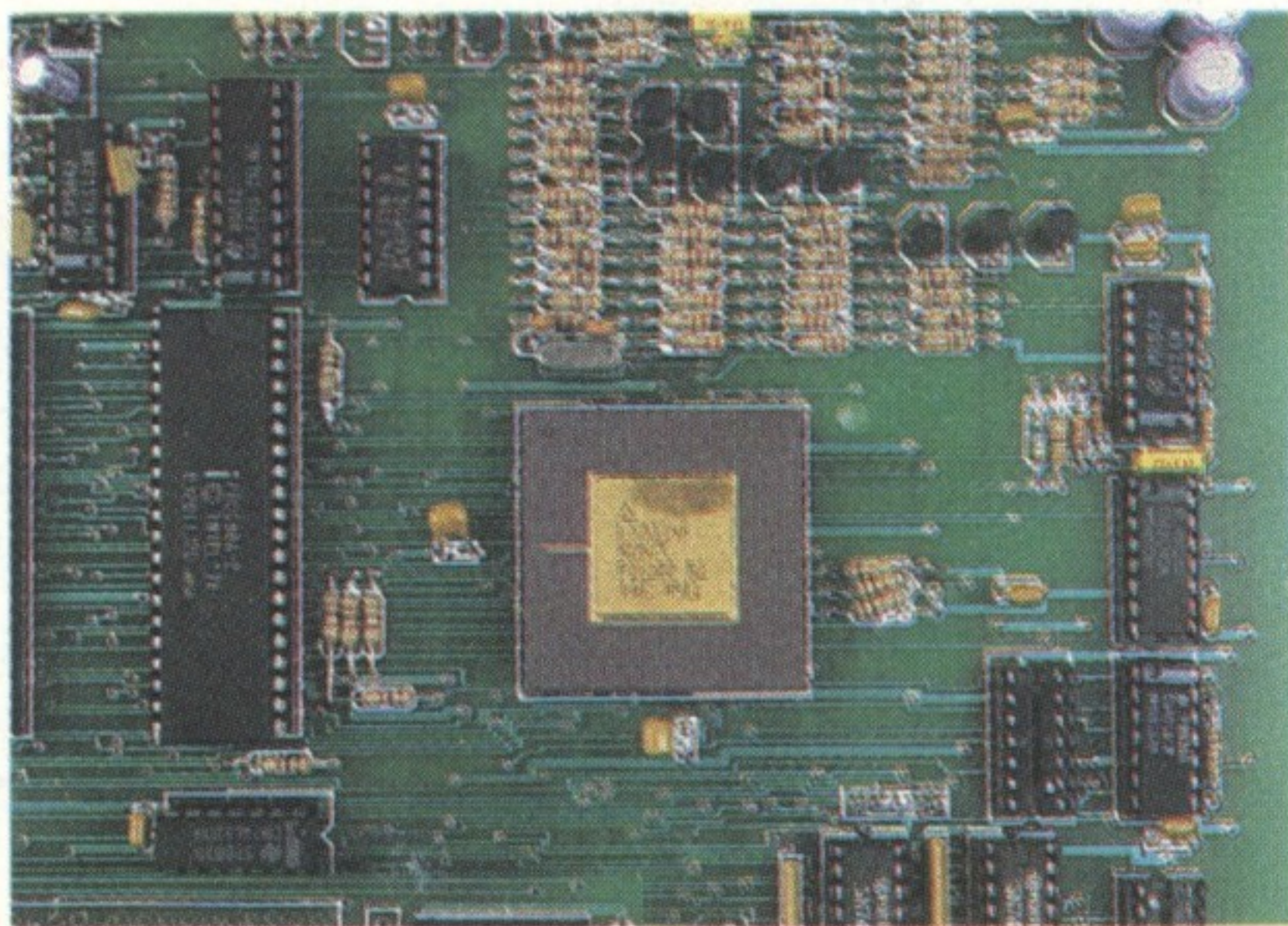
matematico "specializzato" in geometria. Ruotare e spostare le coordinate dei giochi 3D prospettici è un lavoro che porta via molto tempo e, peggio ancora, incredibilmente difficile da programmare. Il triste fatto della vita è che le leggi della trigonometria tridimensionale sono universali - tutti questi giochi devono eseguire gli stessi calcoli - così sarebbe perfettamente logico e possibile costruire un processore *custom* da mettere a disposizione di tutti.

SEPARAZIONE DEI POTERI

Ancor più importante è il modo in cui questi processori sono collegati alla memoria della macchina. Assegnando ad ognuno dei tre processori del sistema una diversa area di memoria con cui lavorare, sarebbe possibile farli lavorare contemporaneamente per la maggior parte del tempo. In questo modo, per esempio, il processore matematico può elaborare un nuovo set di coordinate mentre il *blitter* prepara la nuova schermata ed la CPU controlla eventuali collisioni con gli oggetti. Non solo il *blitter* ed il processore matematico eseguono i loro compiti più rapidamente di quanto potrebbe fare la CPU, ma lo fanno senza interrompere i compiti in cui è, contemporaneamente, impegnata la CPU.

L'idea di dividere la memoria, a dire il vero, risale a qualche anno prima dei *blitter* e dei processori matematici. Fu implementata sul Sinclair Spectrum per risolvere i problemi di "contesa". In un moderno computer multiprocessore viene definita "contesa" la situazione in cui due processori - la

L'ASIC (Application Specific Integrated Circuit) della Konix è tipico dell'integrazione su larga scala dei processori *custom* dell'ultima generazione, caratterizzati da trattamento di grafica e sonoro, controller dell'accesso al disco, *blitter*, ROM, RAM veloce, unità logica e aritmetica e porte di controllo. Questo mostro garantisce gran parte della potenza che caratterizza questa console.



CPU ed il *blitter*, per esempio - cercano di accedere contemporaneamente alla stessa area di memoria. Quando ciò accade, il processore meno importante viene temporaneamente "congelato" (il termine tecnico è *locked out* ovvero "chiuso fuori").

La soluzione proposta da Sinclair sullo Spectrum era di dividere la memoria, così che il processore video potesse solo accedere ai primi 16K (dedicati alla gestione dello schermo). La CPU aveva totale controllo dei rimanenti 32K, così che i programmi in essa allocati potessero essere eseguiti a velocità maggiore. Ciò significava: a) che la maggior parte dei programmi dello Spectrum erano scritti nei detti 32K di memoria e b) che lo Spectrum a 16K risultava perciò quasi totalmente inutile.

È bizzarro notare come l'Amiga sia progettato in modo da avere una divisione della memoria molto simile, ma la maggior parte degli Amiga presenti in Italia - gli Amiga 500 - non possono beneficiarne. L'Amiga si basa sul concetto di "memoria-chip" e "memoria-fast": il processore video ed il *blitter* hanno accesso solo alla memoria-chip, così che i programmi allocati nella memoria-fast possono funzionare a piena velocità anche quando il *blitter* sta lavorando. Sfortunatamente fanno parte della memoria-chip i primi 512K di memoria, così un'Amiga 500 privo di espansione in realtà non ha alcuna memoria-fast. C'è, ad ogni modo, un lato positivo. Dal momento che difficilmente un gioco per Amiga sfrutta il *blitter*, la CPU ha comunque a sua disposizione tutta la memoria-chip (anche se questo non sembra molto positivo).

MOMENTI DI VUOTO

Lo Spectrum non aveva un *blitter*, naturalmente, ma la sua CPU perdeva comunque tempo prezioso mentre attendeva che il processore video accedesse alla memoria. Come in tutti i normali computer, il processore video doveva leggere la parte di memoria riguardante lo schermo per conoscere le immagini che doveva mostrare: questo genere di accesso alla memoria aveva la priorità su qualunque altro compito la CPU stesse svolgendo in quel momento, perché il lavoro svolto dal processore video ha un'importanza critica. Se il processore video interrompesse il suo lavoro anche per un istante, l'immagine presentata sarebbe rovinata. Sui primi ZX80 e ZX81, l'accesso al video non aveva la priorità. Ovviamente, quando la CPU era al lavoro le immagini sparivano!

ELABORAZIONE PARALLELA

Non appena iniziate a dividere in questo modo la memoria, assegnandone le varie parti ai diversi processori, ecco che non ci sono più limiti: potete creare un sistema potente quanto volete, con le sole limitazioni dovute al costo. Se un processore matematico non basta, perché non mettercene un secondo? Se un solo *blitter* è troppo lento a generare gli sprite, perché non farlo fare a più di uno?

Questo modo di assegnare un singolo compito a diversi processori si chiama "elaborazione parallela". Negli ambienti informatici seri, si parla di elaborazione parallela già da diverso tempo. Lì, al centro dell'attenzione, troviamo i Transputer Inmos, una famiglia di processori progettata appunto per compiti d'elaborazione parallela. L'anno scorso, l'Atari ha presentato la sua ATW, la Transputer Workstation (già nota come ABAQ), basata sul Transputer T800, il più potente della serie. Per diverse ragioni, questo è un processore anomalo: ha una memoria *cache* veloce preinserita, può svolgere internamente calcoli in virgola mobile ed ha costi di produzione spaventosi.

Se il prezzo del T800 scenderà in modo ragionevole nei prossimi cinque anni, potrebbe diventare, verso la meta degli anni '90, la base per delle macchine-gioco eccezionali. Purtroppo, l'attuale altissimo costo è legato alla sua estrema complessità. Tutte le possibilità che offre non sono gratis ed è presumibile che siano gestibili in altro modo.

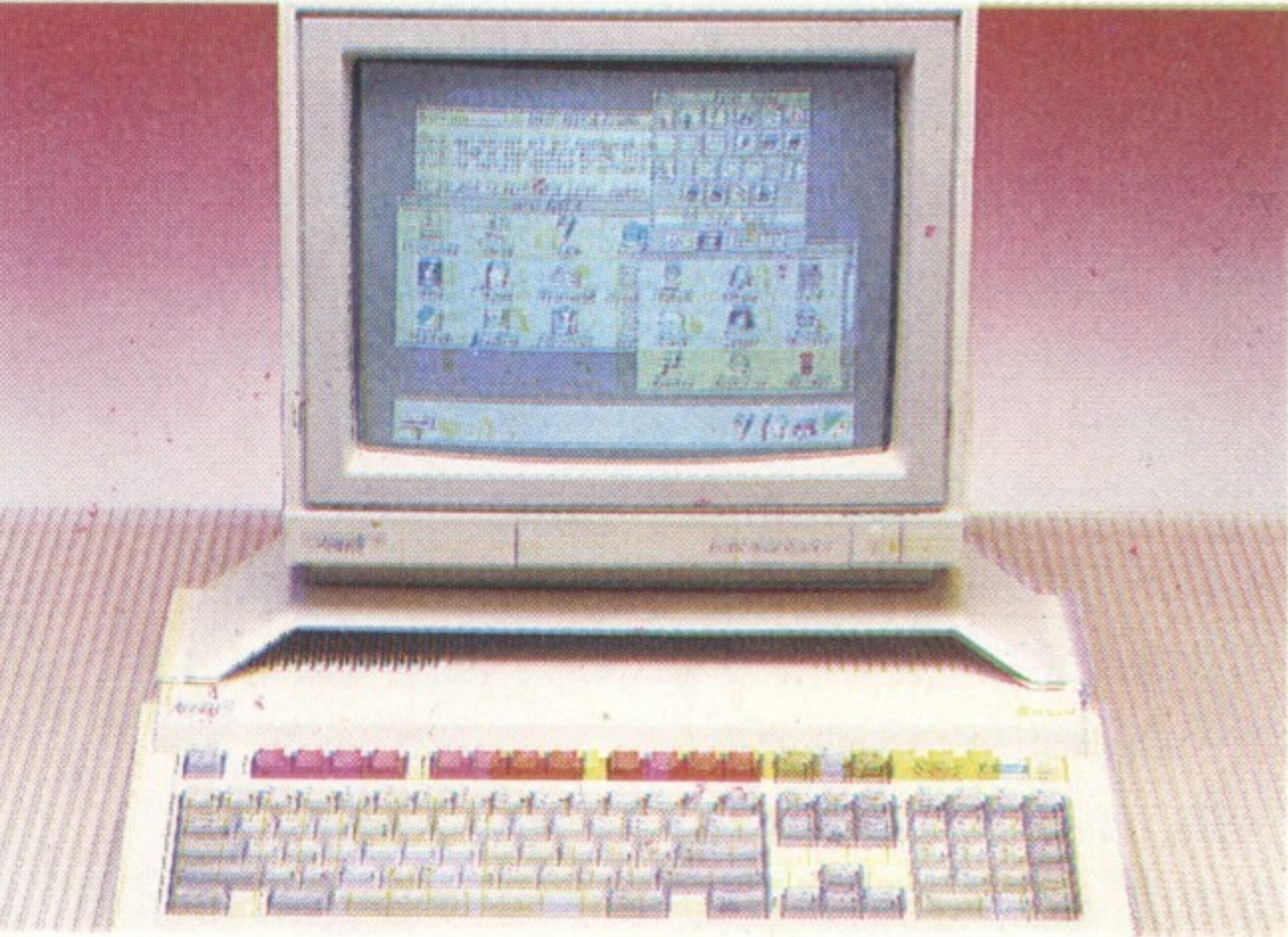
Attualmente, comunque, se volete un processore progettato appositamente per lavorare in gruppo, la famiglia dei Transputer è l'unica esistente. Questo, a meno che qualche gruppo di progettazione intraprendente non decida di creare in proprio un nuovo processore destinato a compiti di elaborazione parallela. Un'impresa per nulla facile, ma che può essere redditizia: i tecnici della Acorn hanno prodotto l'ARM per la serie di computer Archimedes, un processore notevole da tutti i punti di vista. Grazie all'avvento della tecnologia RISC (Reduced Instruction Set Computer), è possibile ottenere dei microprocessori eccezionali con pochissimi componenti.

TRA IL DIRE E IL FARE

Ovviamente, produrre una CPU dedicata presenta un grosso svantaggio: nessuno saprà come programmarla. Quando furono lanciati sul mercato, lo Spectrum e il C64 usavano CPU che erano già ben conosciute da molti appassionati. Anche l'ST e l'Amiga ebbero il vantaggio di essere stati anticipati dal QL della Sinclair, che adottava lo stesso processore 68000. Ma una macchina-gioco con un processore dedicato (o più di uno) è tutto un'altro affare. Perché un programmatore dovrebbe prendersi la briga di studiare un processore

sconosciuto, se nessun altro produttore lo userà mai? Non è vantaggiosamente economico dedicare molto tempo ad un processore senza storia e con poco futuro.

Ma anche a questo può esserci un rimedio: se un produttore ha le idee ben chiare su cosa vogliono i programmatori, potrebbe creare un processore tanto interessante da non lasciare loro altra scelta. I programmatori adorano l'hardware ben progettato e sono disposti a tutto pur di lavorarci sopra. Se la Acorn, alcuni



La gamma degli Archimedes fa uso dell'ARM, un processore progettato appositamente e molto popolare presso i programmatori per il suo set di istruzioni ridotto e per l'altissima velocità. Ma, sfortunatamente, il prezzo del sistema non è alla portata di tutti...

anni fa, avesse inserito l'ARM in una macchina-gioco della stessa fascia di costo dell'ST, avrebbe trovato un numero incredibile di potenziali programmatori. Questo perché, per merito dei progettisti, l'ARM è sia facile che divertente da programmare. La curva d'apprendimento non è neanche troppo elevata, perché ha una struttura ben congegnata. In confronto, il processore 8086 della console Konix è un vero mostro, anche se lo si conosce bene. Il tipo di approccio della Acorn è azzardato, è vero, ma il processore giusto nella macchina giusta ha tutte le probabilità di attirare i programmatori.

VIDEO INTELLIGENTE

Fino ad ora, abbiamo discusso di potenza di elaborazione, cioè della frequenza con la quale un computer può alterare aree di memoria o eseguire calcoli. Avere una notevole potenza di elaborazione va benissimo, ma non è l'unica cosa che serve - ne è essenziale - in una buona macchina-gioco. Un buon hardware video può assicurare ottimi effetti su schermo anche con una CPU mediocre.

Prendete, ad esempio, il CPC della Amstrad e il Commodore 64. In termini di pura potenza di elaborazione vince senz'altro il CPC, il suo processore Z80A a 4MHz è approssimativamente due volte più potente del 6502 a 1MHz del C64, ma per la maggior parte dei giochi è meglio il C64. Perché? Non è una questione di tavolozza di colori o risoluzione dello schermo - vince il CPC anche in questi due casi - ma piuttosto nella maniera più intelligente con la quale l'hardware del C64 indirizza l'uscita video. Tra la memoria del C64 ed il vostro televisore c'è un sacco di altra roba, mentre il processore video del CPC è montato così come arriva dal produttore.

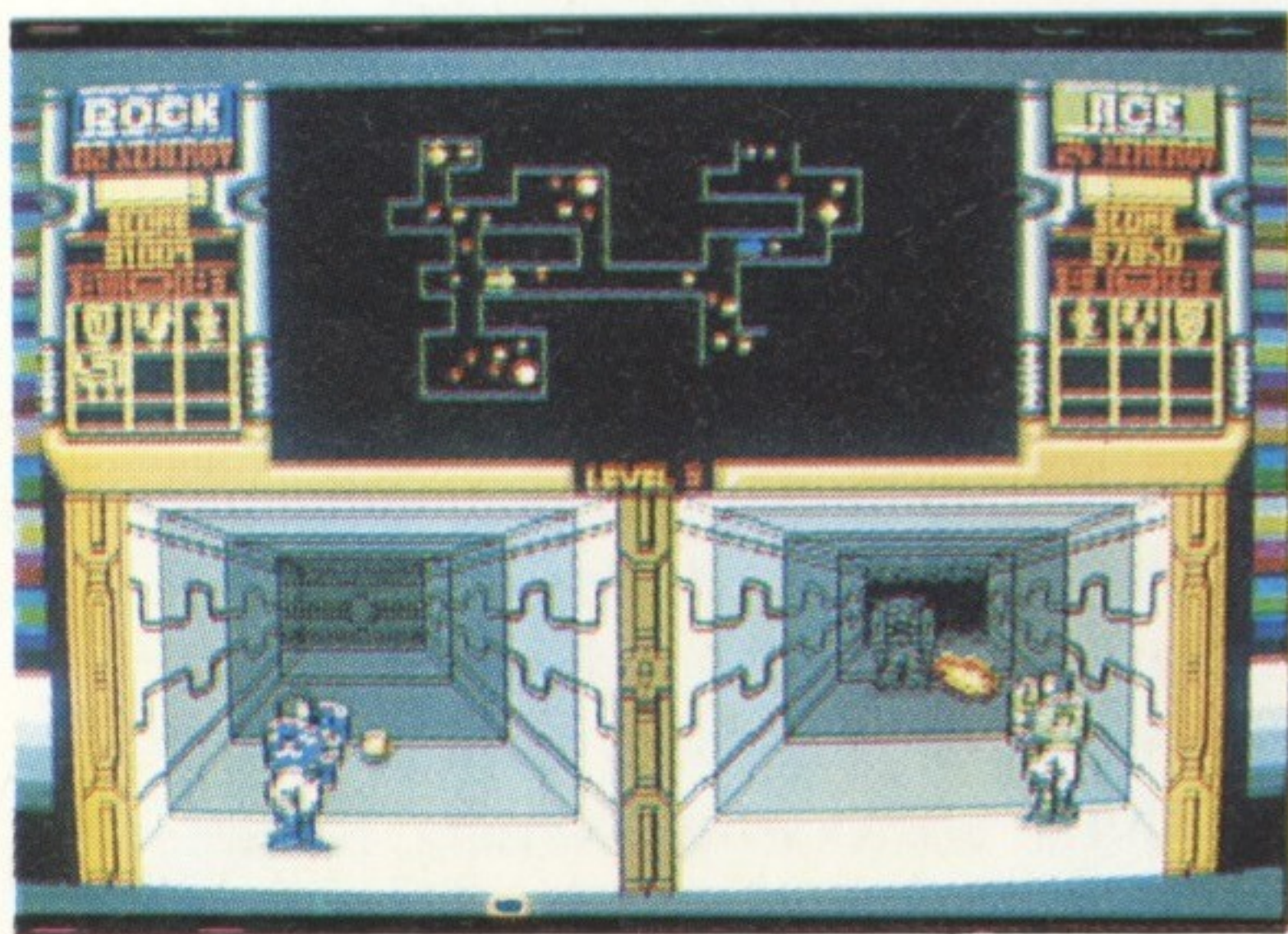
Come la maggior parte delle macchine prodotte nel 1984, il CPC ha una gestione dello schermo "a mappa di memoria": una grossa parte della RAM viene usata per descrivere come deve apparire ogni schermata e in essa ogni bit definisce la posizione ed il colore di un singolo pixel (cioè di un puntino luminoso). Il processore video del CPC scansiona la memoria 50 volte al secondo per tradurre la descrizione dei pixel nei segnali luminosi necessari per raffigurarli sullo schermo. Se un determinato bit significa "il prossimo pixel è blu", il processore video dice semplicemente al monitor che "il prossimo pixel è blu". Questo è tutto: legge una descrizione, la traduce in segnale video e la manda al monitor.

In un sistema a mappa di memoria, il programma modifica l'immagine su schermo alterando le descrizioni dei pixel immagazzinate in una data locazione di memoria video. Se lo schermo deve scorrere orizzontalmente, il programma deve modificare i pixel uno per volta, cinquanta volte al secondo, per restare sincronizzato con il processo di scansione del processore video. A schermo pieno, la cosa

è impossibile; lo Z80 del CPC non è sufficientemente veloce. Se vi siete mai chiesti perché i giochi a scorrimento orizzontale per CPC sono così scadenti, ora ne conoscete la ragione (anche l'ST, pur con tutta la potenza del suo 68000 ha problemi, in questo settore: ancora una volta, il responsabile è lo schermo a mappa di memoria).

Il C64 non ha questi problemi di scorrimento, perché il suo schermo non è così strettamente legato al contenuto della mappa di memoria video. Se si desidera che lo schermo scorra lateralmente, non è necessario spostare tutte le descrizioni dei pixel; basta fare sapere al processore video che l'immagine va spostata di lato e questi ne terrà conto traducendo le descrizioni in segnali. Per passare questo tipo di istruzioni al processore video, il C64 impiega solo qualche milionesimo di secondo, circa diecimila volte meno del metodo brutale usato dal CPC.

Lo stesso, anche se in misura minore, avviene con gli sprite. Su una macchina a lettura diretta della mappa di memoria, animare uno



Nei primi anni ottanta, il progetto hardware del C64 permise lo sviluppo di molti giochi basati sull'uso di sprite.

sprite significa cancellare la vecchia immagine e sostituirla con una nuova. Su quelle tipo C64, ciò non è necessario: basta mandare al processore video l'istruzione di muovere un dato sprite o di sostituirlo con un altro. Le figure degli sprite non sono parte della mappa di memoria, ma sono, piuttosto, disegnate "sopra" di essa dal processore video durante la traduzione del segnale. Ancora una volta, un processo

che avrebbe fatto sprecare molto tempo in un sistema a lettura diretta della mappa di memoria, prende solo qualche milionesimo di secondo sul C64, grazie al suo hardware video.

GLI SPRITE DEGLI ANNI '90

Eccovi un bel rompicapo. Come il C64, l'Amiga ha un processore video dedicato per lo scorrimento del video e degli sprite. Possiede anche diverse altre possibilità grafiche che possono fare comodo per i giochi. Tuttavia, nulla di tutto ciò, di solito, è utilizzato dai giochi in commercio. Salvo poche eccezioni, tutti usano il processore centrale, il 68000, per fare gran parte del lavoro. Perché le caratteristiche che hanno fatto il successo del C64 vengono regolarmente ignorate sull'Amiga?

La risposta di massima è che la maggior parte dei giochi per Amiga sono stati fino ad ora trasferiti direttamente dall'ST e, come abbiamo visto, i giochi per ST si affidano completamente al 68000 (vista la mancanza di un processore video intelligente che possa essere d'aiuto) e così devono fare anche i giochi per Amiga. Ma questa è solo la prima metà della risposta. I programmatori si limitano semplicemente a trasferire i giochi dall'ST all'Amiga perché possono farla franca. E possono farla franca perché l'Amiga non ha mai realmente avuto bisogno di tutto quel ben di dio di hardware video.

Davanti ad una macchina-gioco che implementa via hardware lo scorrimento dello schermo e lo spostamento degli sprite, viene da chiedersi: "A cosa serve?". Rispetto al C64, la risposta è ovvia: il 6502 a 1 MHz. non ha la potenza sufficiente per ottenere giochi decenti. Senza l'hardware aggiuntivo, il C64 sarebbe stato, come il suo predecessore Vic 20, un computer sottodimensionato. Rispetto all'Amiga, invece, la risposta non è così ovvia. Con gli sprite, il 68000 se la cava brillantemente da solo, senza creare problemi di apprendimento o restrizioni tecniche ai programmatori.

Più in generale, l'Amiga evidenzia un altro fatto relativo all'hardware video, e cioè che ha dei grossi limiti. Un processore video progettato per giochi con scorrimento dei pixel e degli sprite, non serve a nulla in un'avventura isometrica tridimensionale alla *Knight Lore*. Non serve neppure per velocizzare degli oggetti in 3D solido, né per gestire le vaste porzioni di paesaggio e di fogliame usate nei giochi frontali tipo *Out-Run*. In altre parole, benché l'Amiga sia la punta avanzata della macchina-cocktail, il suo processore video è

stato progettato per un tipo di giochi ormai in via d'estinzione. Per le conversioni dei coin-op dei giorni nostri, anche l'Amiga si deve basare sul 68000 oltre che sul *blitter*. Come abbiamo già visto, anche il *blitter* non è esente da critiche, essendo poco utile per i giochi in 3D solido, quindi nel giro di qualche anno anche l'Amiga sarà sullo stesso livello dell'Atari per quanto riguarda la grafica. Avrà sempre parecchio hardware in più, ma nulla che valga la pena usare per i nuovi tipi di giochi.

CAMBIARE LE REGOLE...

Con lo sviluppo degli stili di gioco, cambiano anche le esigenze di hardware. Questo gioco in 3D isometrico fu in origine scritto per lo Spectrum dalla Ultimate. Gli sfondi cambiavano semplicemente al cambio di stanza. D'altro canto, *OutRun*, ha bisogno o di un processore molto veloce o di un *blitter* per modificare il paesaggio che scorre ai lati (e sullo sfondo) della strada.



Sopra, OutRun
Sotto, Knight Lore

UNA NUOVA ERA VIDEO

Se i *blitter* e i processori matematici sono destinati a diventare "*tritter*" e *trigger*, come dovranno evolvere i processori che gestiscono lo scorrimento del video e gli sprite? Tanto per cominciare, il concetto di scorrimento dei pixel è destinato a scomparire.

Lo scorrimento dei pixel è stata un'innovazione importante per i computer con mappa di memoria video. Negli anni passati, su sistemi come i PC con scheda CGA, gli Spectrum e così via, ciascun byte in memoria doveva rappresentare più pixel. Muovere la mappa di memoria a un byte per volta, il minimo spostamento possibile, comportava lo spostamento di tutta l'immagine. Lo stesso problema, lo incontriamo anche ora sui PC EGA, sull'ST e sull'Amiga: la modifica di un byte in memoria provoca lo spostamento di otto pixel sul video.

Ma questo sistema di mappe di memoria video comincia ad essere sorpassato e, negli anni '90, dovrebbe cessare di essere usato. Il sistema di rappresentazione usato sulla console Konix e sull'Archimedes è molto superiore: lo spostamento di un byte in memoria equivale a quello di un pixel sullo schermo, quindi diventa facile far scorrere lo schermo di un pixel per volta anche senza l'aiuto del processore video. I progettisti potranno anche incorporare in una nuova macchina lo scorrimento via hardware, ma questo cesserà di essere un plus di vendita: con schermi di un byte per pixel, lo scorrimento via hardware sarà automaticamente morbidissimo.

Contemporaneamente, possiamo aspettarci di vedere spuntare sui computer da casa un nuovo genere di sprite. Invece di avere forma e dimensioni fisse, i "super-sprite" degli anni '90 saranno totalmente flessibili. Il processore video non si limiterà a disegnarli sopra uno sfondo, ma potrà anche ingrandirli, ridurli, ruotarli o distorcerli nel modo appropriato. Quando gli oggetti si avvicineranno al giocatore diventeranno più grandi, fino a riempire lo schermo se necessario, grazie ad una semplice istruzione passata al processore, senza che il programmatore sia costretto a prevedere più sprite diversi per forma o dimensioni.

Super-sprite come questi sono già comuni nelle sale giochi (giochi come ThunderBlade della Sega ne sono zeppi), ma un equivalente per i computer da casa è ancora di là da venire.

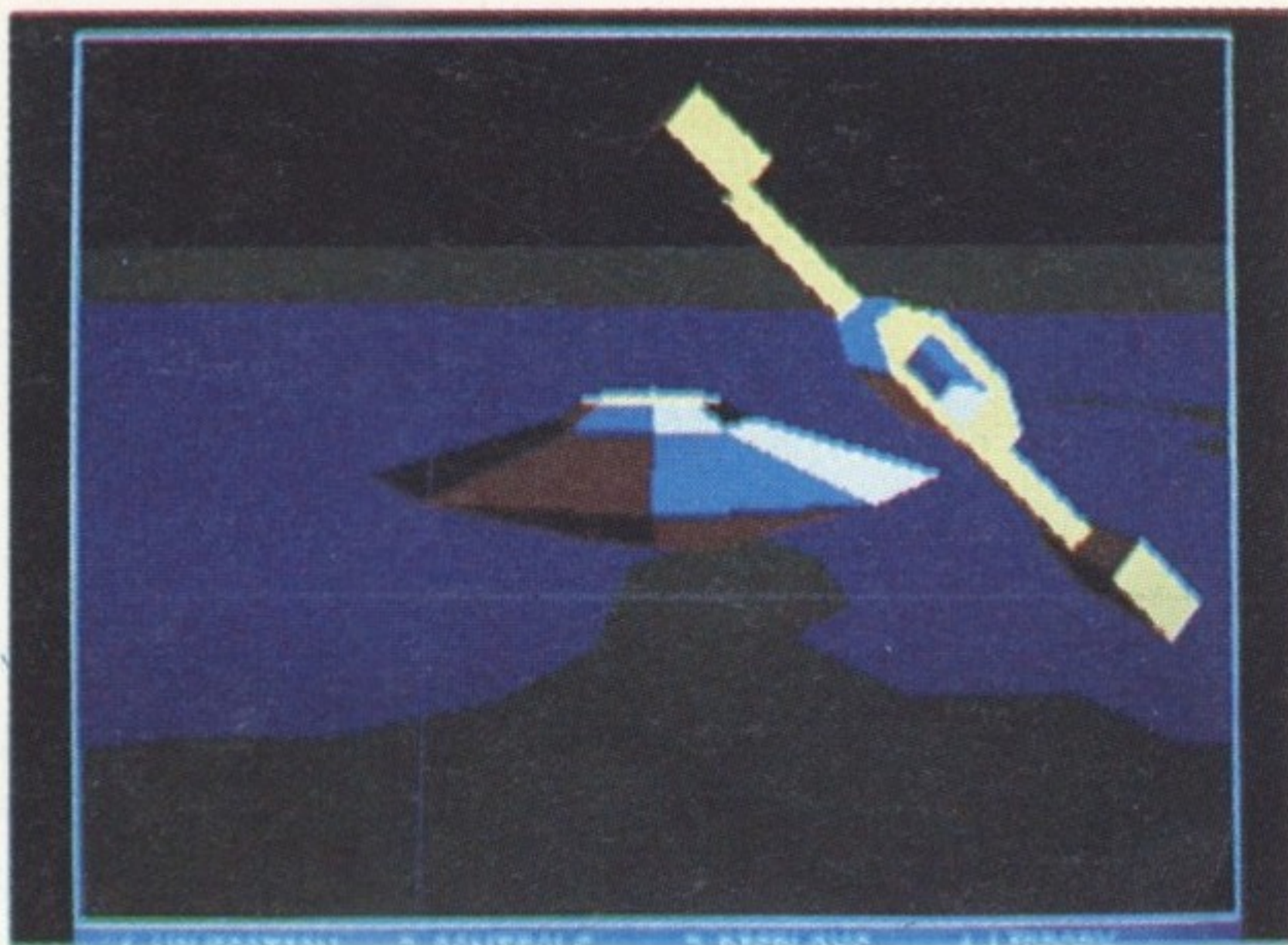
Tuttavia, la sfida del futuro per l'hardware video non si limita al

semplice scorrimento e agli sprite, anzi potrebbe modificare radicalmente l'aspetto dei giochi. Riguarda la tecnologia video nel suo complesso e l'uso che ne è finora stato fatto nei computer.

Per i giochi, il tipico *display* casalingo è un televisore a colori, che di per sé ha una risoluzione limitata, solitamente di 320x200 pixel. Negli anni '90 potremo aspettarci l'ascesa di un nuovo standard televisivo, con una definizione molto maggiore. Ma anche se il salto sarà grande, diciamo fino 800x600 pixel, non sarà in grado di competere con l'aumento della potenza dei computer. Già adesso, l'Atari Transputer Workstation può gestire 16 milioni di colori per pixel alle velocità tipiche dei videogiochi; e questo prima ancora che dei programmatori seri ci abbiano messo su le mani. Le macchine-gioco della metà del prossimo decennio saranno ancora più potenti. Mappe di memoria di schermo da 2000x1500 pixel non dovrebbero essere una cosa insolita, ma sarà difficile trovare un televisore o un monitor in grado di visualizzare queste immagini (almeno a prezzi abbordabili).

Il problema è che i computer producono immagini più dettagliate con meno colori - l'ideale è 256 - mentre i televisori producono immagini meno dettagliate, ma con molti più colori. Il metodo televisivo sembra migliore: le forme si fondono una con l'altra, cosicché, a meno di essere vicinissimi, non è possibile vedere la distinzione tra i pixel. Sullo schermo di un computer, però, non è possibile fondere le figure e le linee diagonali hanno un andamento seghettato che uccide ogni realismo.

C'è una tecnica di computer graphics chiamata *anti-aliasing* (o anche *de-jagging*), che permette di fondere morbidamente tra di loro le figure e che consente di ottenere un



Le raffigurazioni in 3D solido come questa necessitano di un "tritter" - un blitter in grado di generare forme triangolari - per disegnare le forme sulle quali è basata la figura. Già che ci sono, i progettisti potrebbero metterci anche un trigger, un processore che gestisca la trigonometria tridimensionale degli oggetti del gioco.



Non è la risoluzione a rendere tanto attraenti i giochi del Multi-System della Konix, ma la possibilità di visualizzare contemporaneamente 256 colori sullo schermo.

Il tipico schermo televisivo ha una risoluzione di soli 320x200 pixel, ma l'enorme gamma di colori dà l'illusione di un'immagine molto più definita.

processore video intelligente che traduca la mappa di memoria in un'immagine di 320x200 pixel per visualizzarla su un normale televisore, cioè un'immagine con un quinto della risoluzione originale. Tradurrebbe il colore di ciascun pixel televisivo tenendo in considerazione la corrispondente area di 5x5 pixel e facendo la media del loro valore cromatico (si veda il diagramma per capire il procedimento).

Con un processore *de-jagger*, gli altri processori (la CPU, il *tritter*, il processore per i super-sprite, ecc.) potrebbero funzionare normalmente - in alta risoluzione e con relativamente pochi colori - mentre il vostro TV a colori da 625 linee di risoluzione continuerebbe a ricevere immagini in bassa risoluzione. Il *de-jagging* provvederebbe ad ammorbidire i bordi delle figure generate dal *tritter* e, se usato per l'elaborazione video finale, potrebbe eliminare alcuni degli effetti più sgraditi della rotazione e del ridimensionamento dei super-sprite.

realismo simil-televisivo molto gradevole. Ma la tecnica anti-aliasing è difficile da programmare e tiene occupata la CPU troppo a lungo per venire usata in un normale gioco. Ci vuole una soluzione hardware, e qui, di nuovo, entrano in ballo i processori video. Immaginate che la vostra macchina abbia una mappa di memoria video con, diciamo, la descrizione di un'immagine di 1600x1000 pixel a 256 colori. Invece di cercare un monitor in grado di raffigurare tutti i 1600x1000 pixel, potreste installare un

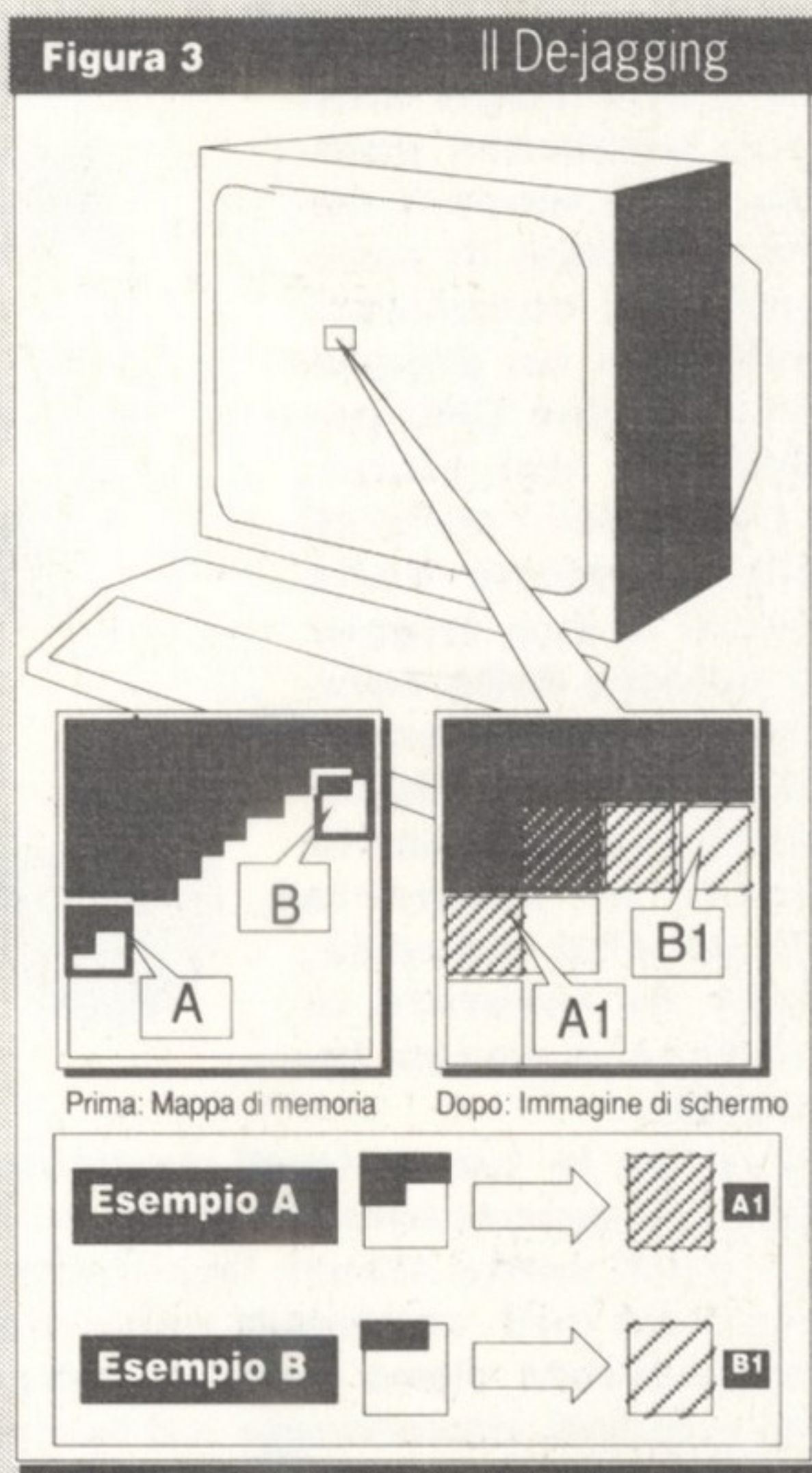
PER UNO SCHERMO MIGLIORE

Per capire come funziona un processore *de-jagger*, prendiamo una microscopica sezione della mappa di memoria video di descrizione dei 1600x1000 pixel. Per semplificare, supponiamo che si tratti del bordo di un triangolo nero su sfondo bianco.

Il bordo descritto nella memoria è troppo dettagliato per essere rappresentato sul televisore, ma se ne può occupare il *de-jagger*. Esso divide l'area considerata in porzioni di 5 x 5 (qui ce ne sono 16) e calcola la media dei valori di descrizione dei pixel di ciascuna porzione. È questa media che viene usata nel pixel che apparirà sullo schermo.

Se tutte le 25 descrizioni sono nere, il pixel sullo schermo sarà nero; se sono tutte bianche, il pixel che apparirà sarà bianco. Se c'è un misto dei due, il pixel sul TV sarà grigio (la tonalità dipenderà dall'equilibrio tra le descrizioni di bianco o di nero).

Nel caso della porzione A, per esempio, ci sono otto descrizioni di nero, il 32% del totale, che daranno un valore di grigio del 32% per il corrispondente pixel A1. La porzione B ha un numero inferiore di descrizioni di nero, solo 3, cioè il 12% del totale, cosicché il corrispondente pixel B1 sarà grigio al 12%.



IL FUTURO

Noi di K crediamo che il divertimento informatico sia IL medium del futuro. E anche se immaginiamo che tecnologie come il CDI o il Video Interattivo possano eventualmente diventare dei rivali per i computer e per le console, siamo comunque convinti che ci sarà sempre spazio per i computer tradizionali con tastiera e mouse. In queste pagine abbiamo appunto delineato alcune delle promesse, delle possibilità e dei problemi che ci riserveranno le macchine-gioco del futuro. Ora sta all'industria del settore concretizzarle!

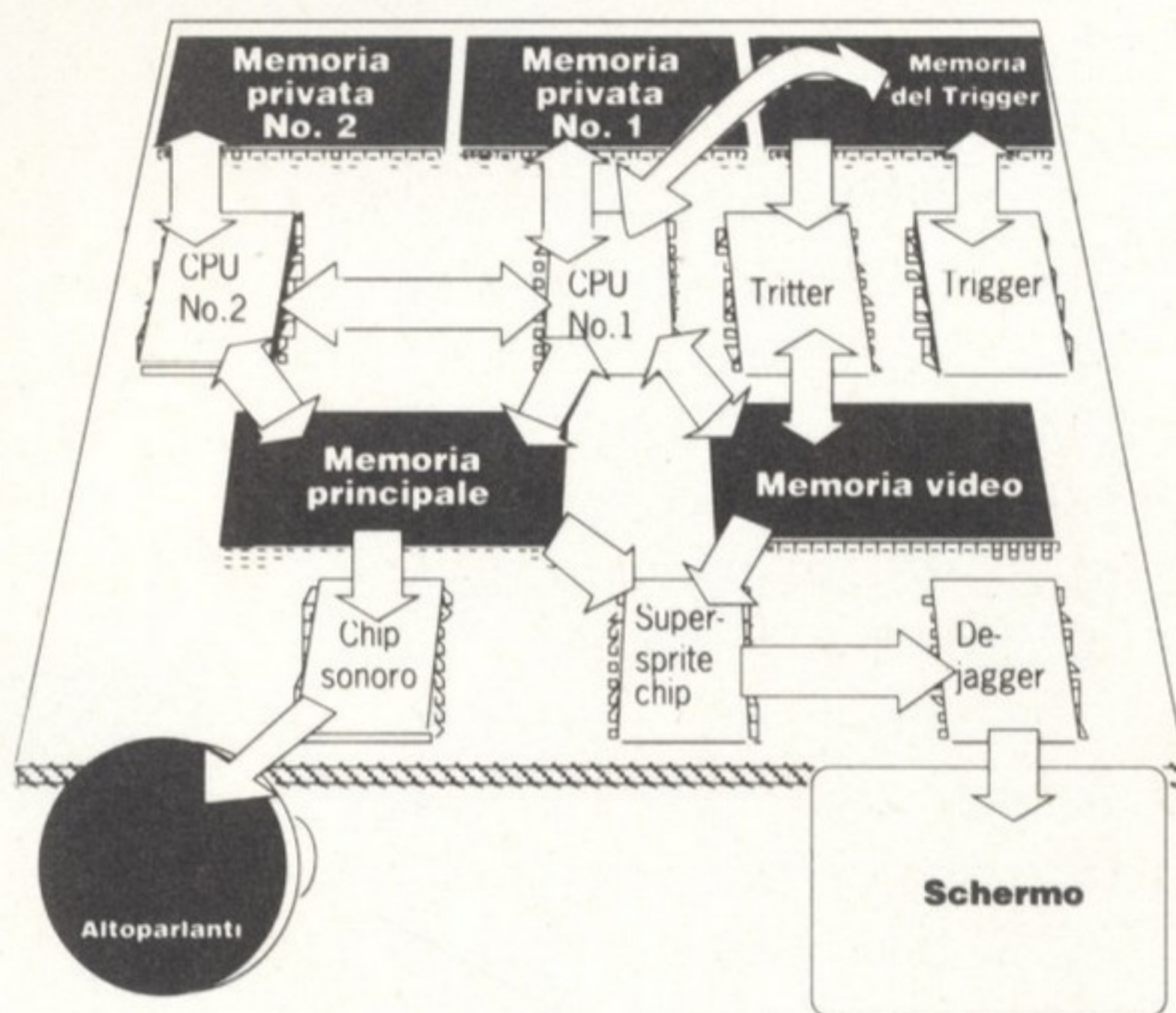
Questa ipotetica macchina-gioco degli anni '90 può sembrare complessa, ma i modelli definitivi rischiano di essere ancor più complicati. Qui abbiamo un doppio di tutto, due CPU per il controllo del gioco, due tritter per i poligoni e l'elaborazione delle immagini e due trigger per il calcolo delle coordinate tridimensionali. Sia le CPU che i trigger dispongono di banche dedicate di memoria SRAM ad alta velocità, in modo da lavorare in simultanea e senza interferenze.

Quando le tre serie di processori hanno creato il fondale dello schermo, il processore dedicato ai super-sprite gli sovrappone le immagini degli sprite stessi, ingrandite come in *ThunderBlade*, o ruotate a partire dai dati contenuti nella memoria principale del computer. La figura risultante viene quindi ridotta e ridefinita dal *de-jagger* (vedi la figura) e quindi visualizzata sul televisore.

Notate che in questo progetto, il processore matematico è sparito. Con i *trigger*, dedicati alla gestione della geometria del gioco, il processore può essere sostituito da un generatore di forme d'onda, che per semplicità, è fuso con il processore audio.

Figura 4

La Macchina Gioco degli anni 90



LE SPECIFICHE

Ovviamente, ci sono molti fattori da tenere in considerazione nel formulare le specifiche di una macchina-gioco per gli anni '90, ma crediamo che qualunque macchina che segua almeno questi dieci punti possa avere successo!

1. Un *tritter* capace di ridisegnare l'intero schermo due volte ogni 50esimo di secondo.
2. Un *trigger* che possa ruotare, traslare o ridurre/ingrandire in scala figure con 10000 vertici per ogni quadro televisivo.
3. Due identiche CPU, ciascuna a 32-bit, capaci di accedere alla memoria a 10Mhz (10 milioni di volte al secondo). Da non confondere con la frequenza d'orologio, che dovrebbe ovviamente essere superiore (vedi pag. 6).
4. 1 milione di pixel interni e 256 colori su-schermo da una tavolozza a 18-bit (250 mila colori).
5. Un *de-jagger* - deve produrre un segnale TV-compatibile ed essere in grado di ridurre una griglia quadrata in un pixel TV.
6. 16K di memoria SRAM privata per ciascun processore.
7. Almeno 1Mb di memoria principale.
8. Un processore per produrre super-sprite in grado di disegnare 1000 immagini di sprite per quadro, ciascuno dei quali deve poter essere modificato in scala, ruotato e duplicato come richiesto.
9. I sistemi di sviluppo devono essere consegnati alle case di software almeno sei mesi prima del lancio, consentendo di produrre una ragionevole base di software per il computer.
10. La produzione dovrà essere organizzata in modo da soddisfare la domanda e rispettare i termini di consegna.