Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Praxis: Sound und Grafik

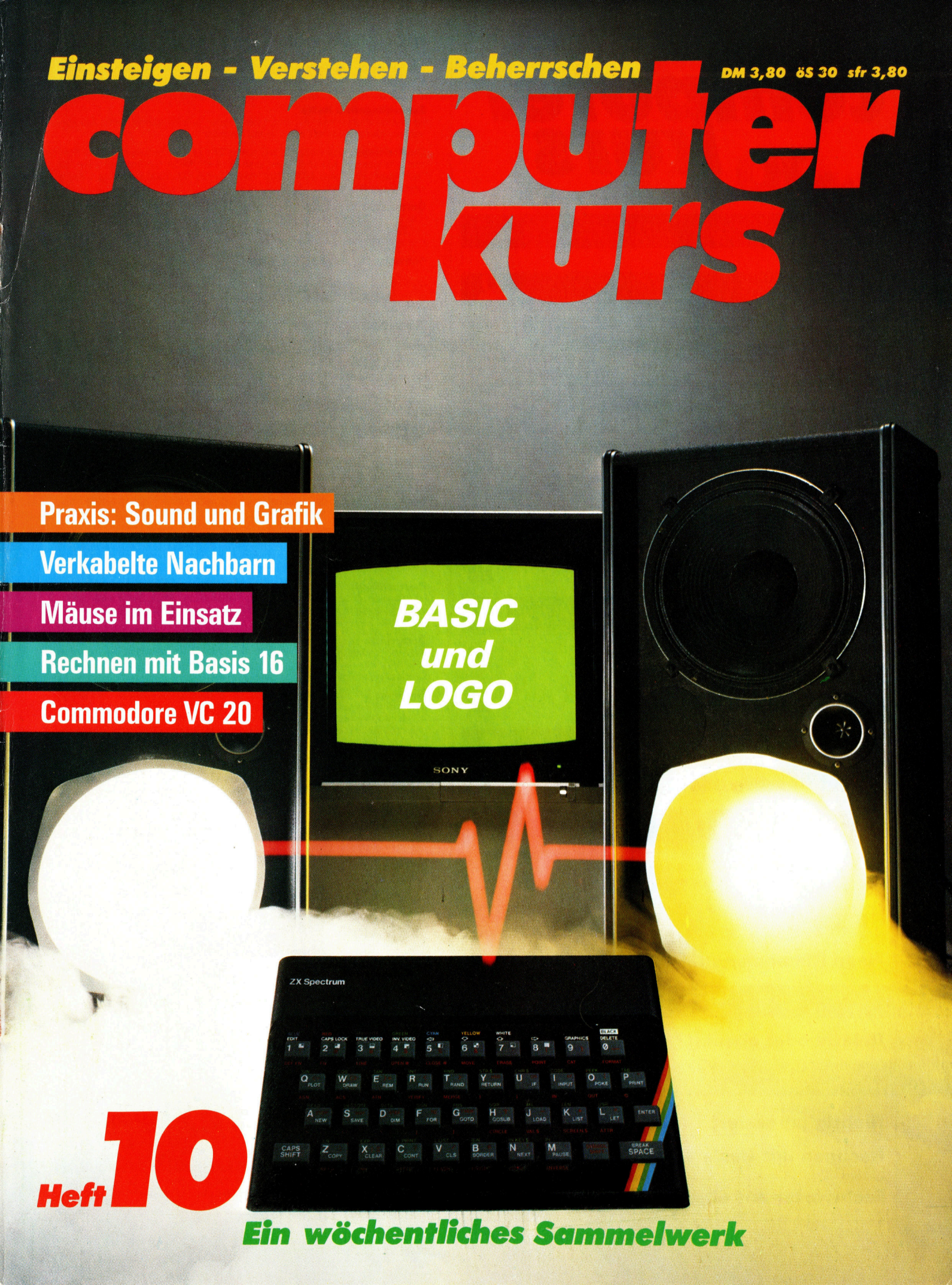
Verkabelte Nachbarn

Mäuse im Einsatz

Rechnen mit Basis 16

Commodore VC 20

BASIC und LOGO



Heft 10

Ein wöchentliches Sammelwerk

computer kurs

Heft 10

Inhalt

WIE SIE JEDE WOCHE IHR HEFT BEKOMMEN

Computer Kurs ist ein wöchentlich erscheinendes Sammelwerk. Die Gesamtzahl der Hefte ergibt ein vollständiges Computer-Nachschlagewerk. Damit Sie jede Woche Ihr Heft erhalten, bitten Sie Ihren Zeitschriftenhändler, Computer Kurs für Sie zu reservieren.

Zurückliegende Hefte
Ihr Zeitschriftenhändler besorgt Ihnen gerne zurückliegende Hefte. Sie können sie aber auch direkt beim Verlag bestellen.

Deutschland: Das einzelne Heft kostet DM 3,80. Bitte füllen Sie eine Postzahlkarte aus an: Marshall Cavendish Int. Ltd. (MCI), Sammelwerk-Service, Postgiroamt Hamburg 48064-202, Postfach 105703, 2000 Hamburg 1, Kennwort: Computer Kurs.

Österreich: Das einzelne Heft kostet öS 30. Bitte füllen Sie eine Zahlkarte aus an: Computer Kurs, Wollzeile 11, 1011 Wien, Postscheckkonto Wien 7857201 oder legen Sie Ihrer Bestellung einen Verrechnungsscheck bei. Kennwort: Computer Kurs.

Schweiz: Das einzelne Heft kostet sfr 3,80. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Computer Kurs, und notieren Sie ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

Abonnement

Sie können Computer Kurs auch alle 2 Wochen (je 2 Ausgaben) per Post zum gleichen Preis im Abonnement beziehen. Der Abopreis für 12 Ausgaben beträgt DM 45,60 inkl. Mwst., den wir Ihnen nach Eingang der Bestellung berechnen. Bitte senden Sie Ihre Bestellung an: Marshall Cavendish Int. Ltd. (MCI), Sammelwerk Service, Postgiroamt Hamburg 86853-201, Postfach 105703, 2000 Hamburg 1, Kennwort: Abo Computer Kurs. Bitte geben Sie an, ab welcher Nummer das Abo beginnen soll und ob Sie regelmäßig für jeweils 12 Folgen einen Sammelordner wünschen. Bei Bestellungen aus Österreich oder Schweiz senden Sie Ihren Auftrag bitte auch an die Hamburger Adresse. Berechnung und Zahlung erfolgen in Landeswährung zum Ladenpreis.

WICHTIG: Bei Ihren Bestellungen muß der linke Abschnitt der Zahlkarte Ihre vollständige Adresse enthalten, damit Sie die Hefte schnell und sicher erhalten. Überweisen Sie durch Ihre Bank, so muß die Überweisungskopie Ihre vollständige Anschrift gut leserlich enthalten.

SAMMELORDNER
Sie können die Sammelordner entweder direkt bei Ihrem Zeitschriftenhändler kaufen (falls nicht vorrätig, bestellt er sie gerne für Sie) oder aber Sie bestellen die Sammelordner für den gleichen Preis beim Verlag wie folgt:

Deutschland: Der Sammelordner kostet DM 12. Bitte füllen Sie eine Zahlkarte aus an: Marshall Cavendish International Ltd. (MCI), Sammelwerk-Service, Postgiroamt Hamburg 48064-202, Postfach 105703, 2000 Hamburg 1, Kennwort: Sammelordner Computer Kurs.

Österreich: Der Sammelordner kostet öS 98. Bitte füllen Sie eine Zahlkarte aus an: Computer Kurs Wollzeile 11, 1011 Wien, Postscheckkonto Wien 7857201 oder legen Sie Ihrer Bestellung einen Verrechnungsscheck bei. Kennwort: Sammelordner Computer Kurs.

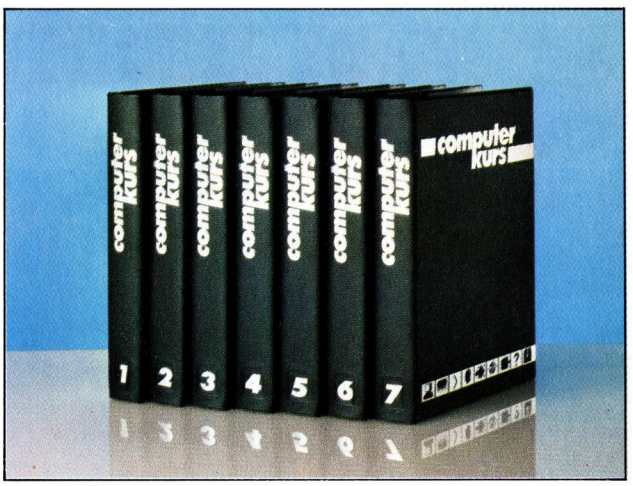
Schweiz: Der Sammelordner kostet sfr 15. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Sammelordner Computer Kurs, und notieren Sie Ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

INHALTSVERZEICHNIS

Alle 12 Hefte erscheint ein Teilindex. Die letzte Ausgabe von Computer Kurs enthält den Gesamtindex — darin einbezogen sind Kreuzverweise auf die Artikel, die mit dem gesuchten Stichwort in Verbindung stehen.

Redaktion: Winfried Schmidt (verantw. f. d. Inhalt), Joachim Seidel, Elke Leibinger, Susanne Brandt, Sammelwerk Redaktions-Service GmbH, Paulstraße 3, 2000 Hamburg 1

Vertrieb: Marshall Cavendish International Ltd., Heidenkampsweg 74, 2000 Hamburg 1, Tel.: 040/23 40 80



Druck: E. Schwend GmbH, Schmollerstraße 31, 7170 Schwäbisch Hall

Verkabelte Nachbarn

Die überregionale Einführung des Kabelfernsehens ermöglicht Besitzern von Heimcomputern den Zugang zu neuen Kommunikationskanälen. In England beispielsweise hat sich ‚Cable TV' längst durchgesetzt.

Kabelfernsehen ermöglicht nicht nur den Empfang von 20 bis 30 verschiedenen Kanälen, sondern bietet Heimcomputerbenutzern die Möglichkeit, miteinander zu kommunizieren. Voraussetzung wäre allerdings, daß dafür ein Kanal zur Verfügung gestellt wird.

Das ‚Viewdata System', das den Zugang zu großen Datenbanken ermöglicht, und das ‚Local Area Network' (Lokales Netzwerk) wurden bereits vorgestellt. Mit der ‚Prestel-Mailbox' (Computer-Briefkasten) des Viewdata Systems in England können Nachrichten zwischen den Teilnehmern ausgetauscht werden, und es ist möglich, allgemeine Informationen über eine Zentrale anzufordern. Der Wirkungsbereich ist jedoch begrenzt.

In ähnlicher Weise stößt auch ein ‚Local Area Network' sehr bald auf Grenzen. Man kann zwar mit jeder angeschlossenen Sende- und Empfangsstation in Verbindung treten, aber auch die besten Microcomputersysteme können ein lokales Netzwerk nur über zwei bis drei Kilometer aufrechterhalten; außerdem sind Banken, Handel und Versicherungsagenturen mit großer Wahrscheinlichkeit nicht angeschlossen. Es ist natürlich möglich, sowohl ein Prestel-Teilnehmer zu sein, als auch einen Netzwerkanschluß zu betreiben, aber auch diese Lösung ist noch weit davon entfernt, all die Möglichkeiten zu eröffnen, die eine perfekte elektronische Verbindung bieten müßte.

Ein technischer Faktor verhindert im Augenblick noch den Aufbau von Netzwerken beispielsweise einer ganzen Stadt. Bei der Ausweitung eines Netzes werden die Signalverluste schon bei geringen Entfernungen zu hoch, ganz gleich ob normale Kabel, Koaxialkabel (wie z. B. für die Verbindung zwischen Fernseher und Antenne), oder Glasfaserkabel eingesetzt werden. Dieser Signalverlust über größere Entfernungen begrenzt die Ausdehnung der lokalen Netzwerke, und nur durch den aufwendigen Einsatz von ‚Boostern' oder Zwischenverstärkern können die Wirkungsbereiche erweitert werden.

Weiterhin muß die Signaldichte berücksichtigt werden – die Menge der Daten, die übermittelt wird. Diese Komponente hat Einfluß auf die Bandbreite, d. h. den Bereich der Übertragungsfrequenzen, der benötigt wird. Es gibt

Die Neubaustadt Milton Keynes wurde von Anfang an mit Kabelfernsehen ausgerüstet. In den 22 000 Wohnungen können sieben TV-Kanäle (sechs aktuelle, ein Kanal nur für Filme) und sechs VHF-Radiokanäle empfangen werden. Nächster Schritt in Richtung auf ein umfassendes Netzwerk ist die zentrale Erfassung von Gas- und Elektrizitätsverbrauch, anstatt jede Gebäudeeinheit einzeln zu messen. In öffentlichen Gebäuden werden bald lokale Netzwerke und Anschlüsse an die allgemeinen Viewdatasysteme zur Verfügung stehen.

eine einfache Regel, nach der für jedes Bit pro Sekunde, das zu übertragen ist, eine Bandbreite von zwei Hertz benötigt wird. Eine Übertragung von 300 BAUD (300 Bit pro Sekunde) erfordert 600 Hz; eine Übertragung von 1200 BAUD dagegen 2,4 kHz. Die normale Sprache über eine Telefonleitung benötigt ein Minimum von 2,4 kHz, für die Übermittlung eines Fernsehbildes in Farbe sind jedoch 8 MHz erforderlich – dreitausendmal mehr als bei normaler Spache. Das heißt: Mit der Bandbreite, die für die Übertragung eines einzigen Fernsehbildes benötigt wird, könnten 3000 Telefongespräche gleichzeitig übermittelt werden.

Die Kapazität eines Übertragungsmediums kann vielleicht am anschaulichsten anhand der Anzahl von Telefongesprächen dargestellt werden, die gleichzeitig darüber geführt werden können. Das Kabel mit der größten Übertragungskapazität, das ‚British Telecom' im Augenblick verwendet, kann 20 000 Gespräche gleichzeitig weiterleiten. Experimente ergaben für Koaxialkabel aber eine maximale Kapazität von bis zu 500 MHz – es könnten demnach bis zu 167 000 Telefongespräche gleichzeitig übertragen werden. Das Kabel selbst hat einen Durchmesser von einem Zentimeter. Bei einem Glasfaserkabel können jedoch über eine Faser, die dünner ist als ein Menschenhaar, schon bis zu 10 000 Telefonate gleichzeitig stattfinden.

In den Vereinigten Staaten verwendet die Firma Music Television (MTV) einen Satelliten, um ihre Programme an 1650 Verteilerstationen zu übermitteln. Diese erreichen dann per Kabel die Wohnungen der 13,5 Millionen Teilnehmer. In jeder Music Television-Sendestunde sind acht Werbeminuten enthalten. Obwohl die Programme nur empfangen werden können, bietet die Firma die Möglichkeit, über gebührenfreie Telefonnummern daran mitzuwirken.

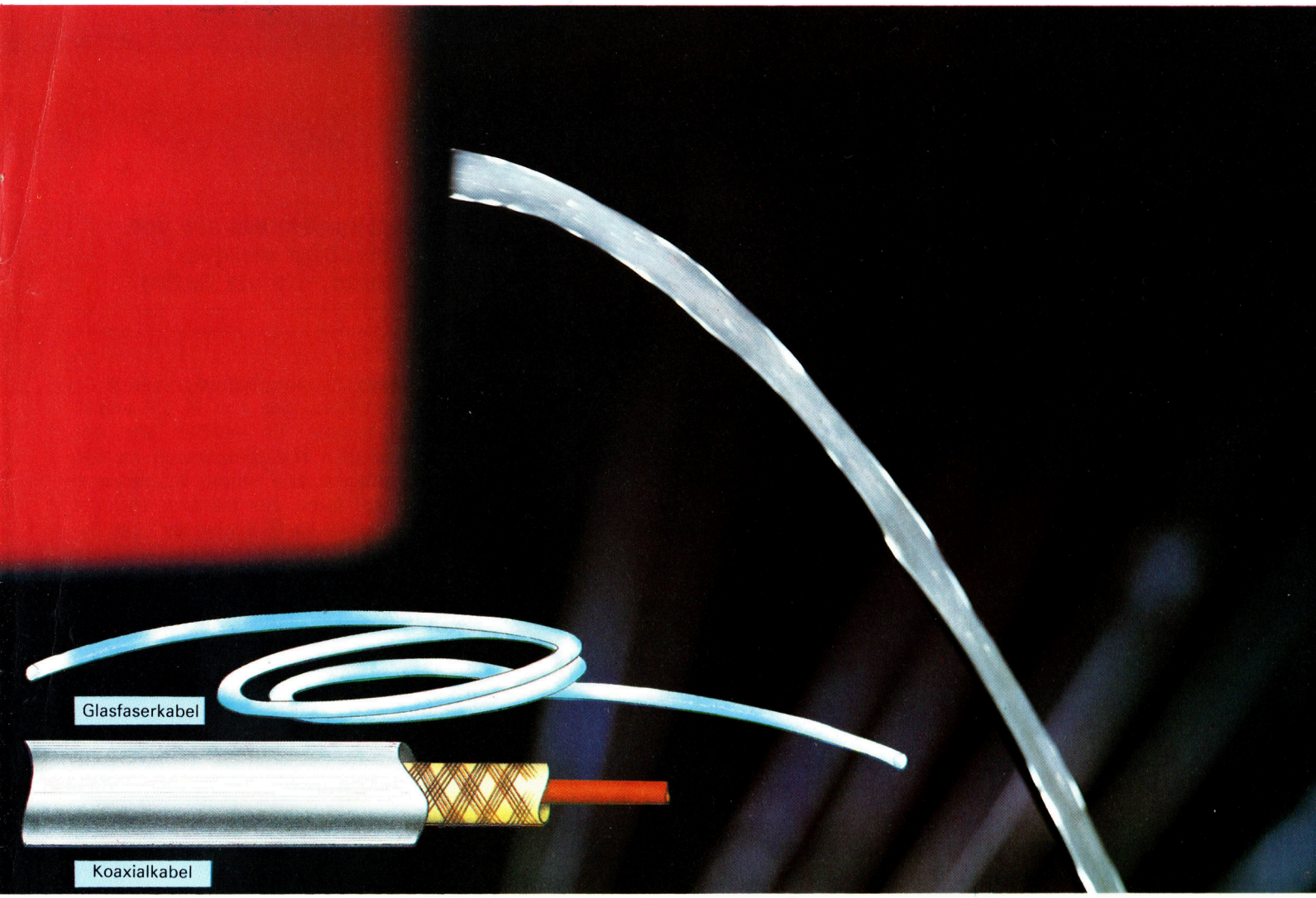
Stadt mit Netzwerken

Schon heute wäre es nicht schwer, eine ganze Stadt mit Netzwerken (Networks) zu versehen, da die benötigten Kommunikationsgeräte bereits auf dem Markt erhältlich sind. Nehmen wir einmal den Acorn B, der mit dem lokalen Acorn-eigenen Netzwerk Econet ausgestattet ist, und stellen damit ein Kommunikationssystem aus einzelnen Networks zusammen. Bei einem lokalen Netzwerk kann eine Station maximal nur 500 Meter von der Zentrale entfernt sein. Über Econet können 254 Teilnehmer zusammengeschlossen werden, wobei eine Station als Zentrale den Zugang zu dem gemeinsamen Speichermedium steuert (File Server). Eine zweite Station übernimmt die Druckersteuerung. Würde jetzt aber noch eine weitere Station dafür abgestellt, die Verbindung zu einem zweiten Netzwerk herzustellen, dann könnten beide Netze miteinander kommunizieren. Die Verbindung läßt sich leicht über die parallelen Schnittstellen der beiden Maschinen realisieren. Je mehr Maschinen für die Kommunikation nach außen abgestellt werden, desto mehr Netzwerke könnte man in diesem Fall ansprechen.

Würde eine solche Idee eine breite Basis in der Bevölkerung finden, könnte dafür speziell ein Netzwerk mit größerer Kapazität konstruiert werden. Es ist jedoch unwahrscheinlich, daß diese Lösung gewählt wird, da sie voraussetzt, daß alle Teilnehmer den gleichen Heimcomputer benutzen.

In Großbritannien hat sich gezeigt, daß Geldmittel für ein derartig großes Projekt nur unter großen Schwierigkeiten zu beschaffen sind. Als von der britischen Regierung die ersten Vorschläge für die Einrichtung eines Kabelfernsehens kamen, wurde argumentiert, daß besonders die Informationsindustrie von einem derartigen System profitieren könnte. Folglich führten mehrere große Firmen, die sich einen zukünftigen Marktanteil sichern wollten, Testprojekte durch. Eine über ganz England verbreitete Kette von Supermärkten richtete sogar ein eigenes System für „Teleshopping" ein, um die Reaktion der Öffentlichkeit zu testen. Der Test mußte jedoch vorzeitig abgebrochen werden – mit der Erkenntnis, daß Dienste dieser Art bei den Käufern nicht ankommen würden.

In den Vereinigten Staaten machte man ähnliche Erfahrungen. Kabelfernsehen ist zwar schon seit längerer Zeit existent, wird aber

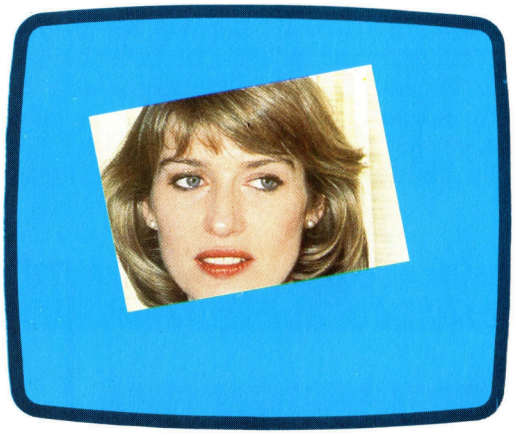


kaum für den digitalen Datenaustausch genutzt.

In Skandinavien und Holland tendiert man dahin, die Heimcomputer auf kommunaler Ebene miteinander zu verbinden. So kann dort die Post schon auf elektronischem Wege versandt werden, es gibt Computersysteme für Babysitting und sogar computerisierte Abstimmungen über Entscheidungen der Verwaltung. Diese sozialen Vorteile sind von den Entscheidungsträgern des britischen Kabelfernsehens nicht berücksichtigt worden.

Neue Möglichkeiten

Auch diejenigen, die im Augenblick nicht daran denken, sich einen Microcomputer anzuschaffen, könnten von den Vorteilen dieser neuen Art der sozialen Kommunikation angeregt werden, sich intensiver mit den modernen Kommunikationsmedien zu beschäftigen. Sind die Kabel erst einmal installiert, dann haben Dienstleistungsfirmen wie z. B. Banken oder Fernmeldedienste unter Umständen sogar selbst Interesse daran, Terminals für eine geringe Kostenbeteiligung oder sogar kostenlos zu installieren. In einigen Gebieten Frankreichs hat man bereits damit begonnen, Telefonbücher abzuschaffen und die Fernmelde-Auskunft durch Abfragen über Viewdata Terminals zu ersetzen.



Ist ein Bild tausend Worte wert?
Es ist schwer, die Anzahl Bits genau zu bestimmen, die für ein farbiges Fernsehbild benötigt werden. Da die dafür verwandte Bandbreite von 9 MHz aber eine Übertragungsrate von 4,5 Millionen BAUD ermöglicht und jedes Bild 25mal pro Sekunde wiederholt wird, erhalten wir nach einer Rechnung die Zahl von 180 000 Bit pro Bild.

Glasfasertechnologie nutzt die vollständige Reflektion innerhalb des Glases. Da es von den „Wänden" der Faser fast vollständig reflektiert wird, durchmißt Licht, das an einem Ende der Faser eingebracht wird, die gesamte Länge der Faser mit erstaunlich wenig Verlust an Leuchtkraft. Für Datenübermittlung wird Licht eines infraroten Lasers verwendet, da das Signal pulsieren muß, d. h. mit großer Geschwindigkeit an- und ausgeschaltet wird. Koaxialkabel sind uns als Verbindungskabel zwischen Fernseher und Antenne vertraut. Sie bestehen aus einem einzigen Strang stark belastbaren Kupferkabels, das von einem Netz dünner Kabel umgeben ist. Beide Leitungen sind voneinander isoliert.

Daten im Warteraum

Computer übertragen Informationen sehr viel schneller als sie von mechanischen Geräten, z. B. einem Drucker, bewältigt werden können. Dieses Zeitproblem lösen Kurzzeitspeicher, die als Buffer bezeichnet werden.

Die als Buffer bekannten Kurzzeitspeicher haben die Aufgabe, „Stöße" aufzufangen. Sie machen es möglich, daß die verschiedenartigen Bestandteile eines Computersystems im „Gleichklang" arbeiten. Das Wort „Buffer" steht beim Computer für zwei ganz verschiedene Dinge – ein Programmierer versteht darunter den gezielten Gebrauch von Speicherbereichen, ein Konstrukteur von Stromkreisen dagegen meint damit eine Art elektrischen Signalverstärker.

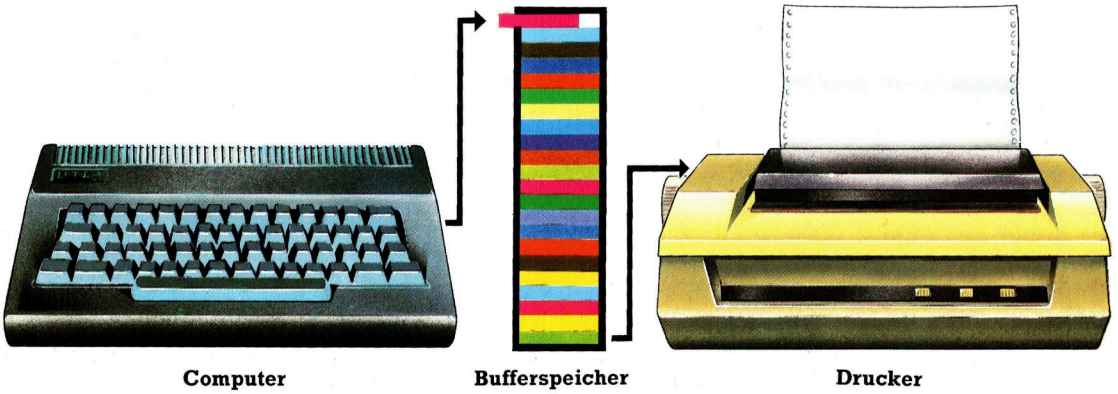
Speicherbuffer

Wie das mit dem Buffer als „Warteraum" funktioniert, macht ein Beispiel aus der Textverarbeitung deutlich. Ein Textverarbeitungssystem kann, neben vielen anderen Dingen, einen Textblock von einer Stelle eines gespeicherten Textes an eine andere verschieben. Der Text besteht aus einer Folge von Zeichen und Zwischenräumen, die ausgedruckt werden, und anderen Zeichen, die nicht ausgedruckt werden (z. B. dem Symbol für den Wagenrücklauf). All diese Zeichen sind im Computerspeicher als ASCII-Code in Binärform dargestellt, wobei für jedes Zeichen ein Byte im Speicher benötigt wird. Um diesen Textblock von einem Speicherbereich zum nächsten zu schieben, muß ein gewisser Teil des Computerspeichers dazu abgestellt werden, vorübergehend den Text aufzunehmen. Ein solcher Speicherbereich wird als Buffer bezeichnet.

Ein anderes Beispiel ist das Ausdrucken eines von einem Textverarbeitungssystem erzeugten Schriftstücks, das aus 15 000 einzelnen Zeichen besteht. Es leuchtet ein, daß diese Zeichen nicht alle gleichzeitig an den Drucker übertragen und sofort ausgedruckt werden können, denn die meisten Drucker schaffen kaum mehr als 80 Buchstaben pro Sekunde. Hier hilft wieder der Buffer. Gesteuert von der Software des Textverarbeitungssystems wird ein Drucker-Buffer zur vorübergehenden Aufnahme des Textes geschaffen, der den Text in einer dem Drucker angemessenen Geschwindigkeit überträgt und zum Ausdruck bereitstellt.

Ein solcher Drucker-Buffer braucht nicht sehr groß zu sein, es genügen schon 128 oder 256 Bytes. Zunächst wird ein Block von ASCII-Zeichen in den Buffer geschrieben, dann werden diese Zeichen einzeln herausgelesen. Was zuerst in den Buffer hineingeschrieben wird, wird auch als erstes wieder herausgelesen (ein Brief soll natürlich in der gleichen Reihenfolge ausgedruckt werden, wie er eingetippt wurde). Buffer dieser Art werden deshalb FIFO-Buffer genannt (first in/first out), was soviel heißt wie „zuerst rein/zuerst raus". Ist schließlich alles aus dem Buffer herausgelesen, füllt ihn die Software mit einem neuen Textblock zum Ausdrucken.

FIFOs werden überall dort eingesetzt, wo Unterschiede in der Verarbeitungsgeschwindigkeit bestehen, sei es zwischen Computern und Druckern, zwischen Computern und Floppy-Laufwerken oder zwischen Computern



Computer Bufferspeicher Drucker

Ein Drucker kann nicht so schnell drucken, wie ihm der Computer die Daten sendet. Darum werden die Daten zunächst in den Buffer „hineingeschrieben". Ist der Buffer gefüllt, veranlaßt er durch ein Besetzt-Signal den Computer, die Datenübertragung abzubrechen. Nun gibt er den Bufferinhalt in einer dem Drucker verträglichen Geschwindigkeit weiter, und zwar in der gleichen Reihenfolge, wie die Daten in den Buffer gelangten. Ist der Buffer leer, beginnt der Computer wieder mit dem Füllen des Buffers. Dieser Vorgang wiederholt sich so lange, bis der gesamte Text ausgedruckt ist.

Kontrollierter Durchfluß

Wenn in einen Tank eimerweise Wasser gefüllt wird, erhält man am Auslaufrohr einen konstanten Wasserstrahl. Auf gleiche Weise „schüttet" der Computer Daten in den Buffer, der sie „tröpfchenweise" in einer angepaßten Geschwindigkeit weitergibt.

und Tastaturen. Computer können wegen ihrer hohen Verarbeitungsgeschwindigkeit zwar eine gedrückte Taste schneller identifizieren als man sie betätigen kann, doch gibt es Situationen, in denen der Computer einen Tastendruck nicht schnell genug erkennen und das zugehörige Symbol anzeigen kann. Dies ist der Fall, wenn der Computer gerade andere Funktionen ausführt (zum Beispiel eine Datei abfragt). Abhilfe schafft auch dabei der Buffer, in dem die Daten der gedrückten Tasten abgelegt werden, bis der Computer sie wieder darstellen kann.

Die meisten Rechnerbetriebssysteme sind nicht in der Lage, andere Funktionen auszuführen, solange ein angeschlossener Drucker läuft. Beim Ausdrucken umfangreicher Programme oder langer Texte ist der Bediener gezwungen, so lange untätig vor dem Bildschirm zu sitzen und zu warten, bis der Ausdruck beendet ist.

Hardware-Buffer

Viele Hersteller bieten deshalb jetzt einen zusätzlichen Drucker-Buffer an, der zwischen Computer und Drucker geschaltet wird. Dieser Kasten enthält einen eigenen Speicher (manchmal bis zu 16 KByte) mit eingebauter Software. Wenn ein Computer Daten ausdrucken soll, sendet er Bytes an den Drucker, bis er ein „Besetzt"-Signal erhält, das ihm sagt, daß der Drucker kein weiteres Byte mehr annehmen kann. Dann muß der Computer so lange warten, bis das Besetzt-Signal aufgehoben und ihm damit angezeigt wird, daß der Drucker wieder Bytes annehmen kann. Obwohl Drukker gewöhnlich einen kleinen Bufferspeicher eingebaut haben (oft nicht mehr als 2 KByte), ist dessen Kapazität für größere Datenmengen zu klein. Hier hilft der zusätzliche Hardware-Buffer mit einer größeren Kapazität, die ausreichend sein kann, alle auszudruckenden Daten aufzunehmen. Der Drucker arbeitet zwar nicht schneller als bisher, aber der Computer wird schneller, weil er andere Aufgaben erledigen kann, während der Drucker arbeitet.

Speicher werden oft ähnlich wie ein großer Topf oder ein Schubfach benutzt, um Programme und Daten zu speichern. Man kann sie aber auch zu Stapelspeichern (Stacks) und Buffern ordnen. Stapelspeicher arbeiten nach dem LIFO-Prinzip (last in/first out); d. h., was zuerst hineinkommt, wird erst zum Schluß herausgelesen. Stacks kann man mit einem Stapel Tabletts vergleichen. Das zuletzt aufgelegte Tablett wird als erstes wieder abgenommen und eine Feder schiebt den restlichen Stapel nach. Stacks und Buffer sind Kurzzeitspeicher. Ihr Unterschied liegt lediglich in der Reihenfolge, wie Daten hineingeschrieben und herausgelesen werden. Stacks werden „intern" in höher entwickelten Sprachen (z. B. in BASIC-Übersetzern) verwendet, wenn Informationen vorübergehend bis zu einem späteren Abruf zu speichern sind. Hier ein BASIC-Beispiel mit verschachtelten FOR-NEXT-Schleifen:

```
FOR X = 1 TO 10
PRINT "X = ";X
FOR Y = 1 TO 10
GOSUB SCAN
NEXT Y
PRINT "C$ = ";C$
NEXT X
```

Wenn der BASIC-Übersetzer (Interpreter) zur zweiten FOR-Anweisung kommt, muß er sich daran erinnern, welche Variable die vorhergehende FOR-Anweisung verwendet (in diesem Fall X). Er legt deshalb die erste FOR-Information in einem Stack ab. Ist die innere Schleife durchlaufen, holt er sich die im Stack obenauf liegende Information und weiß, daß die gegenwärtige FOR-Anweisung die Variable X verwendet. Bei mehrmals verschachtelten FOR-NEXT-Schleifen werden Informationen für mehrere FORs im Stack abgelegt. Wird die Information vom Stack „heruntergeholt", geschieht dies natürlich in der umgekehrten Reihenfolge.

Buffer dagegen arbeiten im „Durchlauf". Sie werden oft in Eingabe-/Ausgaberoutinen als Schnittstelle zwischen den Routinen oder den verschiedenen mit unterschiedlicher Übertragungsart oder Geschwindigkeit arbeitenden Geräten verwendet. Eine Eingaberoutine in BASIC beispielsweise arbeitet in Zeilen, deren Länge durch das RETURN-Zeichen begrenzt ist. Manche Interpreter arbeiten dagegen mit Einheiten, die mit einem Zeichen pro Zeile rechnen. Buffer benötigen gewöhnlich einen Zeiger, der anzeigt, wo im Buffer das nächste Zeichen geschrieben werden soll. Dieser Zeiger kann ein Byte sein oder aus mehreren Bytes bestehen.

Signalverstärkung

Computer arbeiten mit der Transistor-Transistor-Logik, kurz TTL. Diese Schaltung bezeichnet die Binärzahl 1 mit fünf Volt und die Binärzahl 0 mit Null Volt. Die Zentraleinheit eines Computers kann zwar diese Spannungen liefern, aber nicht in einer Stärke, die ausreicht, um alle anderen anliegenden Chips ebenfalls zu aktivieren. Aus diesem Grund sind in die Ausgangsleitungen der Zentraleinheit Signalbuffer geschaltet, um den verfügbaren Strom zu verstärken. Signalbuffer selbst sind kleine Chips, die jeweils als Buffer für sechs Signale arbeiten.

Commodore VC-20

Commodores kleinster Heimcomputer bietet raffinierte Eigenschaften zu einem vernünftigen Preis.

Commodore Business Machines war für einen der ersten Heimcomputer verantwortlich – den Personal Electronic Transactor (PET). Er war 1977 erstmals im Handel erhältlich. Im Jahre 1981 wurde der Commodore VC-20 auf den Markt gebracht, der viele Eigenschaften des PET in sich vereinigt. Nicht nur, daß der VC-20 den gleichen 6502-Microprozessor benutzt, auch das gleiche BASIC befindet sich im ROM, jedoch leider nicht Commodores leistungsfähigste Version.

Der offensichtlichste Unterschied zwischen beiden Maschinen liegt in den zusätzlichen Grafikmöglichkeiten des VC-20. Bis zu 16 Farben sind für das Fenster verfügbar. Allerdings können der Rahmen des Bildschirms und die einzelnen Zeichen und Symbole nur eine von acht Farben annehmen. Der Bildschirm besteht aus 32 384 Bildpunkten.

Der Zeichensatz selbst ist überraschend groß. Er bietet Groß- und Kleinschrift und zwei Sätze von Grafikzeichen auf 62 Tasten. Weitere vier Tasten können dazu benutzt werden, acht programmierbare Funktionen zu unterstützen. Das Design der Tastatur ist technisch wie ergonomisch gut.

Der Hauptnachteil des VC-20 ist die geringe Speicherkapazität von nur fünf KByte, die sich aus 3,6 KByte RAM und 1,4 KByte für Grafik zusammensetzt. Durch zusätzliche Erweiterungsmodule kann die Kapazität jedoch auf 32 KByte erhöht werden.

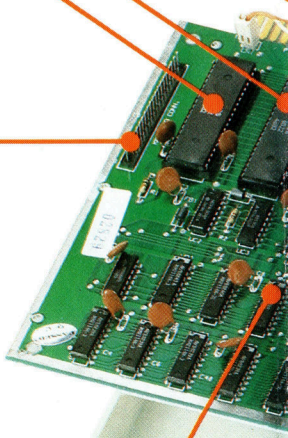
Interfaces für Paddles/Joysticks/Lichtgriffel, Spielmodule/Speichererweiterungen, Drucker/Diskettenstation, Cassettenrecorder und Fernseher sind eingebaut. Eine serielle Schnittstelle mit RS232-Standard kann zum Anschluß eines Modems oder eines Druckers anderen Fabrikates benutzt werden. Außerdem ist ein recht umfassendes Spektrum an Hardware-Erweiterungen und Handbüchern im Fachhandel erhältlich.

Cassettenrecorderanschluß
Aufgrund des speziellen Übertragungsformates muß bei allen Commodore-Computern – also auch beim VC-20 – ein Cassettenrecorder derselben Marke angeschlossen werden.

Benutzer-Ein-/Ausgang
An diese serielle Schnittstelle können mehrere zusätzliche Peripheriegeräte angeschlossen werden.

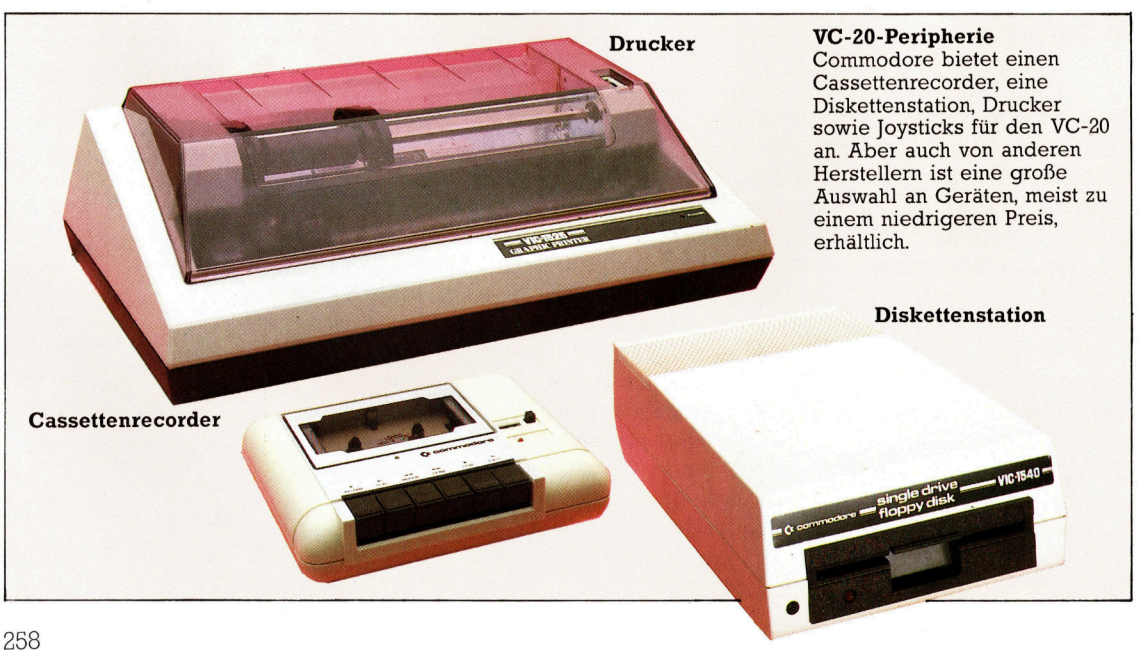
Peripheral Interface Adaptor
Die Coprozessoren kontrollieren sämtliche Ein-/Ausgabeoperationen. Sie sind beispielsweise dazu fähig, parallele in serielle Formate umzuwandeln.

Tastaturstecker
Hier wird die Tastatur mit dem eigentlichen Rechner verbunden.

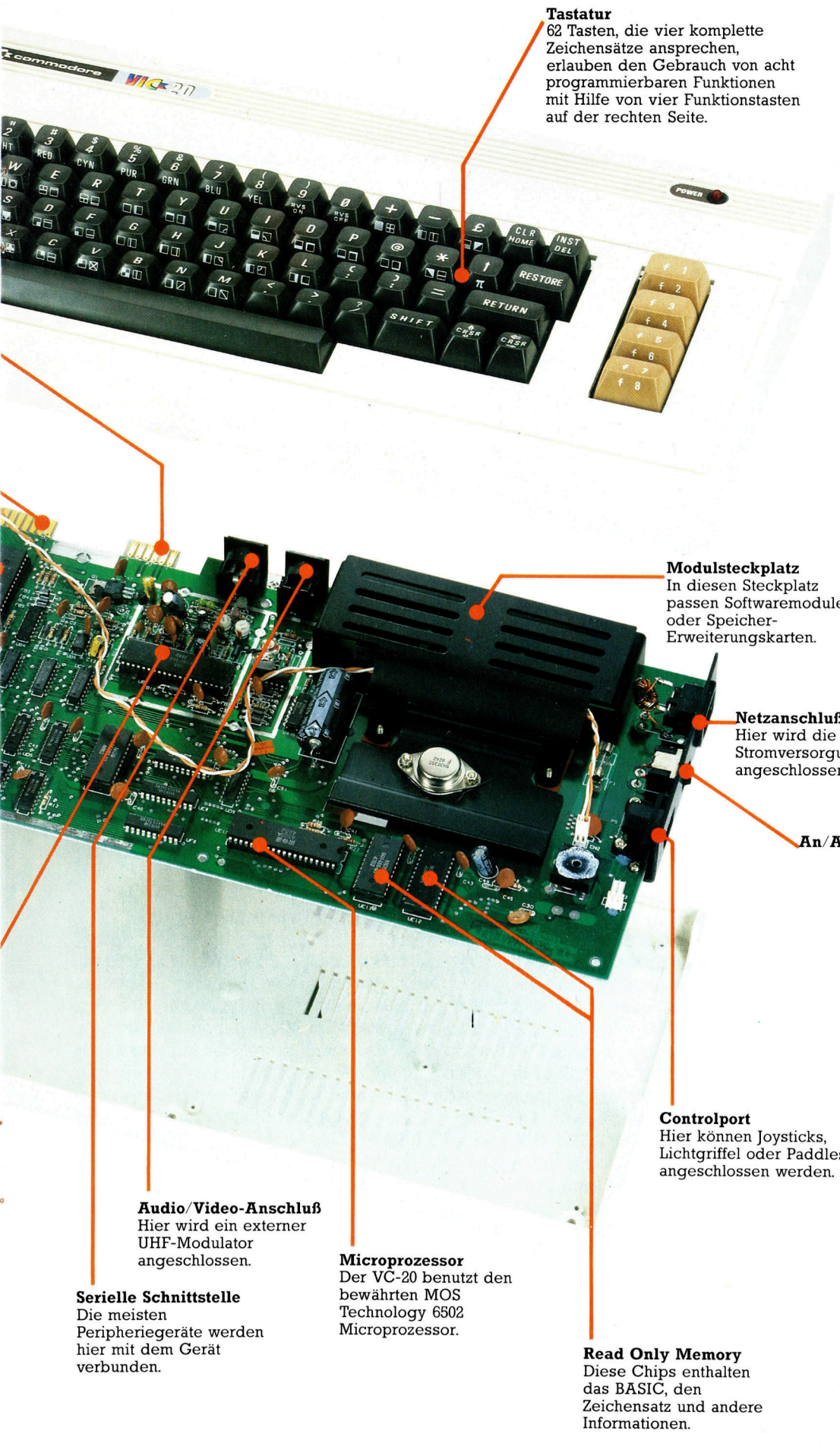


Video-Interface-Chip
Dieser Coprozessor kontrolliert den gesamten Bildschirmaufbau und den Sound-Generator.

Random Access Memory
Der VC-20 ist mit 5 KByte RAM ausgestattet, doch der Speicher kann extern auf 32 KByte erweitert werden.



VC-20-Peripherie
Commodore bietet einen Cassettenrecorder, eine Diskettenstation, Drucker sowie Joysticks für den VC-20 an. Aber auch von anderen Herstellern ist eine große Auswahl an Geräten, meist zu einem niedrigeren Preis, erhältlich.



Tastatur
62 Tasten, die vier komplette Zeichensätze ansprechen, erlauben den Gebrauch von acht programmierbaren Funktionen mit Hilfe von vier Funktionstasten auf der rechten Seite.

Modulsteckplatz
In diesen Steckplatz passen Softwaremodule oder Speicher-Erweiterungskarten.

Netzanschluß
Hier wird die externe Stromversorgung angeschlossen.

An/Aus-Schalter

Controlport
Hier können Joysticks, Lichtgriffel oder Paddles angeschlossen werden.

Audio/Video-Anschluß
Hier wird ein externer UHF-Modulator angeschlossen.

Serielle Schnittstelle
Die meisten Peripheriegeräte werden hier mit dem Gerät verbunden.

Microprozessor
Der VC-20 benutzt den bewährten MOS Technology 6502 Microprozessor.

Read Only Memory
Diese Chips enthalten das BASIC, den Zeichensatz und andere Informationen.

Commodore VC-20

PREIS
350 DM inklusive Cassettenrecorder

GRÖSSE
404 × 216 × 75 mm

GEWICHT
1820 g

TAKTFREQUENZ
1 MHz

MASCHINEN-SPEICHER
Der Computer ist in der Standardausführung mit 5 KByte RAM ausgestattet. Erweiterbar bis zu 32 K.

BILDSCHIRM-DARSTELLUNG
23 Zeilen à 22 Zeichen. Hochauflösende Grafik mit 184 × 176 Punkten. Maximal können 16 Farben angesprochen werden.

SCHNITTSTELLEN
Zwei serielle Anschlüsse, Audio/Video-buchse, Cassetten-anschluß, Modulsteck-platz und Controlport.

PROGRAMMIER-SPRACHE
BASIC

WEITERE PROGRAM-MIERSPRACHEN
Assembler und BASIC-Erweiterungen können dazugekauft werden.

ZUBEHÖR
Handbuch, Netzteil und Antennenkabel

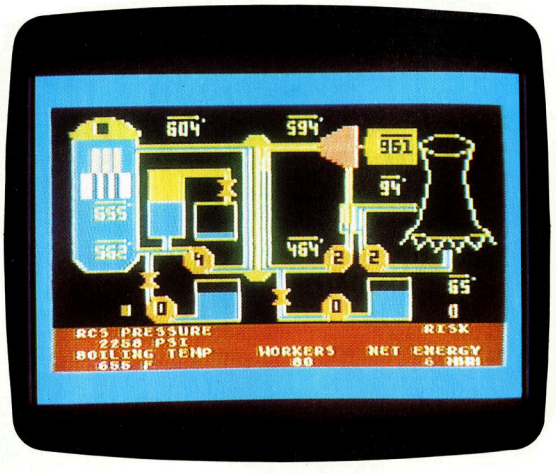
TASTATUR
Schreibmaschinen-ähnliche Tastatur mit 62 Tasten plus vier speziellen Funktionstasten.

DOKUMENTATION
Eines der Gebiete, in denen Commodore schlecht abschneidet, ist die Dokumentation ihrer Heimcomputer. Das Handbuch ist einfach geschrieben und gewährt nur einen sehr oberflächlichen Einblick in die Fähigkeiten dieses Computers. Glücklicherweise gibt es eine große Anzahl erweiternder Literatur.

Modellverhalten

Simulation ist eine hilfreiche Technik zum Testen und Durchspielen von Situationen, die sonst zu gefährlich oder zu kostspielig wären. Auch auf dem Heimcomputer können Simulationen sehr lehrreich sein.

Die Simulation ist eines der wichtigsten Anwendungsgebiete von Computern. Sie ist eine Methode zur Vorhersage von Auswirkungen, die entstehen, wenn eine Situation bestimmten Einflüssen ausgesetzt ist. Das „Modell", das dabei auf dem Rechner durchgespielt wird, ist eine vereinfachte Darstellung dieser Situation, wobei die wesentlichen Gesichtspunkte der Problemstellung beibehalten und die kaum beeinflussenden Einzelheiten weggelassen werden.



Ein wichtiger Anwendungsbereich für die Simulation ist die Ausbildung an Modellen existierender Einrichtungen. Ein Beispiel ist die Atari-Simulation eines Kernkraftwerkes. Der Bediener hat hier die verschiedenen Kühlsysteme zu steuern, um eine Überhitzung des Reaktors zu vermeiden. Die Dokumentation erklärt ausführlich die Steuer- und Kontroll-Funktionen innerhalb eines solchen Kraftwerkes.

Es gibt drei verschiedene Bedeutungen des Wortes „Modell": Das bildhafte Modell, z. B. eine Fotografie oder eine Landkarte, zeigt die räumlichen Verhältnisse und Beziehungen zwischen den einzelnen Bildelementen. Eine andere Art besteht aus Komponenten, die sich zueinander in ähnlicher Weise verhalten wie die das Problem darstellenden wirklichen Elemente. Dazu gehören solche Probleme, wie sie von Analogrechnern gelöst werden. Die dritte Form sind symbolische Modelle, die mit abstrakten Symbolen und mathematischen Verhältnissen zur Darstellung der Modellsituation arbeiten. Diese dritte Modellart wird auf Digitalrechnern simuliert.

Zur Simulation stehen vier grundlegende Situationen zur Verfügung. Die erste ist gegeben, wenn die Situation zu gefährlich wäre, um in Wirklichkeit durchgeführt zu werden. Ein Beispiel dafür wäre die Ermittlung der Radioaktivität, die in der Umgebung eines Kernreaktors noch als unbedenklich angesehen werden kann.

Die zweite Situation mag durch ein Beispiel aus der Volkswirtschaft erklärt werden, wo es nahezu unmöglich ist, die Vielfalt der mit einem Problem zusammenhängenden Einflüsse in einer einzigen mathematischen Formel zusammenzufassen. In diesem Fall ist es einfacher, einzelne Modellgleichungen auf dem Computer zu rechnen und die Auswirkung der Ergebnisse auf verschiedene Geschehnisse und Vorgänge zu beurteilen.

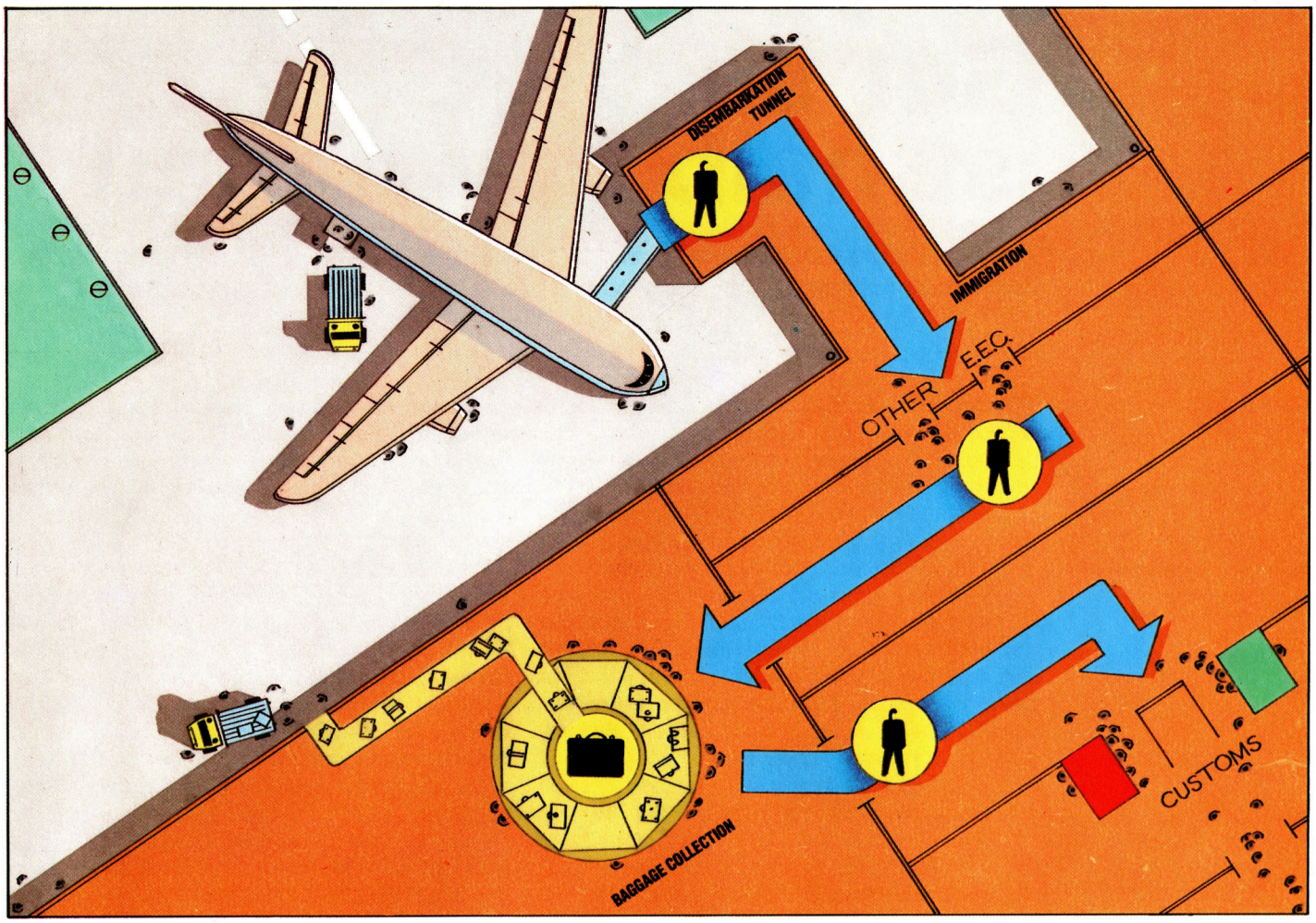
Die dritte Situation wird gekennzeichnet durch Problemstellungen, die mit Objekten so riesigen Ausmaßes verbunden sind, daß sich ein „Experiment im Sandkasten" (der in diesem Fall ein Computer ist) nicht vermeiden läßt. Für einen geplanten Flughafen zum Beispiel werden die technischen und organisatorischen Probleme in umfangreichen Modellrechnungen untersucht. Hierzu gehören Lärmentwicklung und andere Umwelteinflüsse genauso wie das zu erwartende Personen- und Verkehrsaufkommen in der Umgebung des vorgeschlagenen Geländes.

Die vierte Situation, die für Modellrechnungen von Bedeutung ist, betrifft ein Problem, das völlig theoretischer Natur ist und keinerlei physikalische Experimente zuläßt oder ermöglicht. Astrophysiker zum Beispiel stellen Spekulationen darüber an, auf welche Weise Sterne entstehen.

Geschlossene Systeme

Systeme und deren Modelle lassen sich in offene und geschlossene Systeme einteilen. Wenn z. B. eine Familie festlegt, wie das verfügbare Einkommen ausgegeben werden soll, so können die Ausgabenposten auf verschiedene Art und Weise hin- und hergeschoben werden. Am Ende jedoch müssen sich Ausgaben und Einnahmen die Waage halten. In diesem Fall haben wir es mit einem geschlossenen System zu tun. Hätte diese Familie jedoch die Absicht, das Geld nach Lust und Laune auszugeben, ohne auf das verfügbare Einkommen Rücksicht zu nehmen, wäre dies ein offenes System.

Bei theoretischen Simulationen kann man nicht absolut sicher sein, das richtige Modell gewählt zu haben. Die Menschheit glaubte fest daran, daß die Erde Mittelpunkt des Universums sei, bis Kopernikus ein weitaus einfacheres mathematisches Modell mit der Sonne als Mittelpunkt präsentierte. Heute stellen Astronomen fest, daß sich die Sterne mit zunehmen-



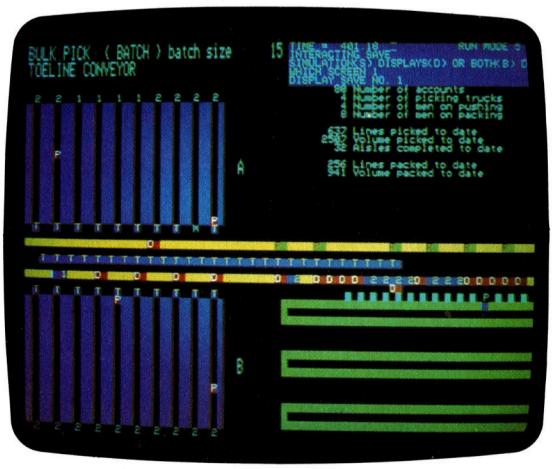
der Geschwindigkeit von uns entfernen, woraus der Schluß gezogen werden kann, daß wir uns im Zentrum des Universums befinden. Bei dieser Annahme wird uns folgendes Modellbeispiel enttäuschen: Wenn man kleine Tintenkleckse auf einen Luftballon spritzt, um damit die Sterne darzustellen, und den Luftballon aufbläst, so wird sich jeder Punkt von den anderen wegentfernen. Dabei stellt kein „Stern" den Mittelpunkt des Ballons dar.

Trotzdem haben Modellrechnungen viele Vorteile. Sie helfen uns, Theorien besser zu formulieren, sie beschleunigen Untersuchungen, ermöglichen das Durchspielen von Abänderungen und, was am meisten zählt, sie sind billiger als reale Experimente.

Ein typisches Problem, für dessen Lösung sich die Simulation geradezu anbietet, ist das Warteverfahren in der Luftfahrt. Die bekannte Situation auf Flughäfen: Die Wetterverhältnisse verschlechtern sich plötzlich, oder es steht nur noch eine Landebahn zur Verfügung. Die ankommenden Flugzeuge müssen „Holding Pattons" fliegen, d. h. so lange einen festgelegten Rundkurs halten, bis sie zur Landung eingewiesen werden können. Da die Treibstoffreserven der Flugzeuge begrenzt sind und der Landeanflug selbst auch einiges an Kerosin verbraucht, muß man den Zeitpunkt genau kennen, zu dem ein Flugzeug spätestens zur Landung ansetzen muß. Das Simulationssystem ermittelt nicht nur die Reihenfolge, in der die Flugzeuge „heruntergeholt" werden, sondern auch Zufallsereignisse wie die außerplanmäßige Ankunft eines Flugzeuges.

Die Simulation hat auch zur Entwicklung spezieller Programmiersprachen, wie GASP, SIMCRISPS und GPSS, geführt. Mit der zunehmenden Kompliziertheit unserer Welt wird die Simulation als Mittel zur Lösung von Problemen immer mehr an Bedeutung gewinnen.

Bei der Planung des neuen Terminals 4 des Flughafens in Heathrow bei London wurden umfangreiche Computersimulationen durchgeführt. So auch die Ankunftssituation, wenn die Flugzeuge mit unterschiedlichen Fluggastzahlen und Gepäckstücken ankommen. Die Ergebnisse waren dann optimale Planungsmaßstäbe.



Die Firma BL-Systems entwickelte ein Microcomputer-Modellpaket für die Erstellung ihrer neuen Werke. Das Modellpaket wird seitdem kommerziell vertrieben. Der Schwerpunkt des Programms liegt zwar auf der modellhaften Darstellung von Abläufen, es liefert jedoch auch Grafiken in Form von Diagrammen. Die Zahlen jeder Prozeßstufe zeigen den Arbeitsablauf an und weisen auf Probleme hin.

Gut sortiert ist halb gewonnen

Daten in eine bestimmte Ordnung zu bringen ist für viele Programme eine unbedingte Notwendigkeit, aber – viele Wege führen zum Ziel.

Bubble Sort
Hier ein Beispiel für den Bubble Sort mit neun Karten (Z ist die Zehn). Die Ordnung nimmt nach jedem Durchlauf (Pass) von rechts nach links zu. Die 1 und 2 unterhalb bezeichnen die Karten, die gerade miteinander verglichen werden.

```
Sortierbeginn
2 8 9 3 Z 5 K 6 7 Beginn Pass 1
1 2
8 2 9 3 Z 5 K 6 7
  1 2
8 9 2 3 Z 5 K 6 7
    1 2
8 9 3 2 Z 5 K 6 7
      1 2
8 9 3 Z 2 5 K 6 7
        1 2
8 9 3 Z 5 2 K 6 7
          1 2
8 9 3 Z 5 K 2 6 7
            1 2
8 9 3 Z 5 K 6 2 7
              1 2
8 9 3 Z 5 K 6 7 2 Ende Pass 1
9 8 Z 5 K 6 7 3 2 Ende Pass 2
9 Z 8 K 6 7 5 3 2 Ende Pass 3
Z 9 K 8 7 6 5 3 2 Ende Pass 4
Z K 9 8 7 6 5 3 2 Ende Pass 5
K Z 9 8 7 6 5 3 2 Ende Pass 6
Sortierende
```

Einsatzmethode
Bei der Einsatzmethode nimmt die Ordnung von links nach rechts zu. Hierbei werden die Karten gleich auf die richtige Position gebracht.

```
Sortierbeginn
2 8 9 3 Z 5 K 6 7
2 1
8 2 9 3 Z 5 K 6 7
2   1
9 8 2 3 Z 5 K 6 7
    2 1
9 8 3 2 Z 5 K 6 7
2       1
Z 9 8 3 2 5 K 6 7
      2   1
Z 9 8 5 3 2 K 6 7
2           1
K Z 9 8 5 3 2 6 7
        2     1
K Z 9 8 6 5 3 2 7
          2       1
K Z 9 8 7 6 5 3 2
Sortierende
```

Programmierer haben sehr viel Zeit für die Entwicklung von Sortieralgorithmen (Methoden der Problemlösung) aufgewendet. Hochentwickelte Sortiersysteme sind schwer zu analysieren und noch schwerer darzustellen. Die einfachen Methoden lassen sich aber leicht mit Hilfe von Spielkarten erklären.

Legen Sie dreizehn Karten der gleichen Farbe in einer Reihe auf einen Tisch aus. Die Ordnung ist dabei nicht wichtig, nur das As und die Zwei sollten nicht am rechten Rand der Reihe liegen. Die Aufgabe besteht darin, jetzt von links nach rechts in absteigender Reihenfolge zu sortieren (König, Dame, Bube . . . As). Diese Aufgabe erscheint sehr einfach. Stellen wir jedoch die Bedingung, daß immer nur eine Karte bewegt werden darf, daß Karten nicht aufeinander gelegt werden dürfen und daß nur der geringstmögliche Platz auf dem Tisch belegt werden darf, dann wird die Aufgabe auf einmal alles andere als einfach, und man muß schon nachdenken, um eine effektive Methode dafür bestimmen zu können. In diesem Beispiel stellen die Karten die einzelnen Daten dar, die zu sortieren sind, der begrenzte Platz auf dem Tisch entspricht dem Speicherbedarf des Computers, und Sie selbst simulieren den Programmablauf. Wie läßt sich das Problem nun lösen?

1) Nehmen Sie eine Münze und legen Sie sie als Markierung unter die Karte am linken Ende der Reihe. Vergleichen Sie jetzt die markierte Karte mit der Karte rechts davon. Sind beide in der richtigen Reihenfolge? Wenn nicht, dann vertauschen Sie die Karten, lassen aber die Münze an ihrem Platz. Beobachten Sie dabei, was zu tun ist, um die beiden Karten zu vertauschen, wenn Sie nur eine Karte bewegen und keine Karte auf eine andere legen dürfen.

2) Sind beide Karten in der richtigen Reihenfolge, bewegen Sie die Münze eine Position nach rechts und wiederholen Sie Schritt 1. Jetzt befinden Sie sich in einer „Schleife", die beendet ist, wenn die Münze unterhalb der Karte am rechten Ende der Reihe liegt. Damit haben Sie einen Durchlauf (englisch: Pass) ausgeführt.

3) Sehen Sie sich am Ende des ersten Durchlaufs die Karten einmal an. Das As hat seinen korrekten Platz am rechten Ende der Reihe gefunden. Wenn Sie jetzt die Karten wie in Schritt 1 und 2 nochmals durchgehen, liegt die Zweierkarte auf ihrem richtigen Platz. Die Schritte 1 und 2 werden so lange wiederholt, bis alle Karten in absteigender Reihenfolge sortiert sind.

Sie haben sicher bemerkt, daß diese Methode einige Nachteile hat: Sie ist umständlich, nicht ökonomisch, da der Austausch zweier Karten drei Vorgänge erfordert und – als wichtigstes – viele der Vergleiche zwischen den Karten sind überflüssig. Liegt z. B. das As nach dem ersten Durchgang an der richtigen Stelle, braucht man die Münze nicht mehr auf die Position 13 (an der sowieso kein Vergleich mehr erfolgen kann) zu legen. Nach dem zweiten Durchgang hat es keinen Zweck, die Münze auf Platz 12 zu legen, da jetzt auch diese Karte auf der richtigen Position liegt und so weiter.

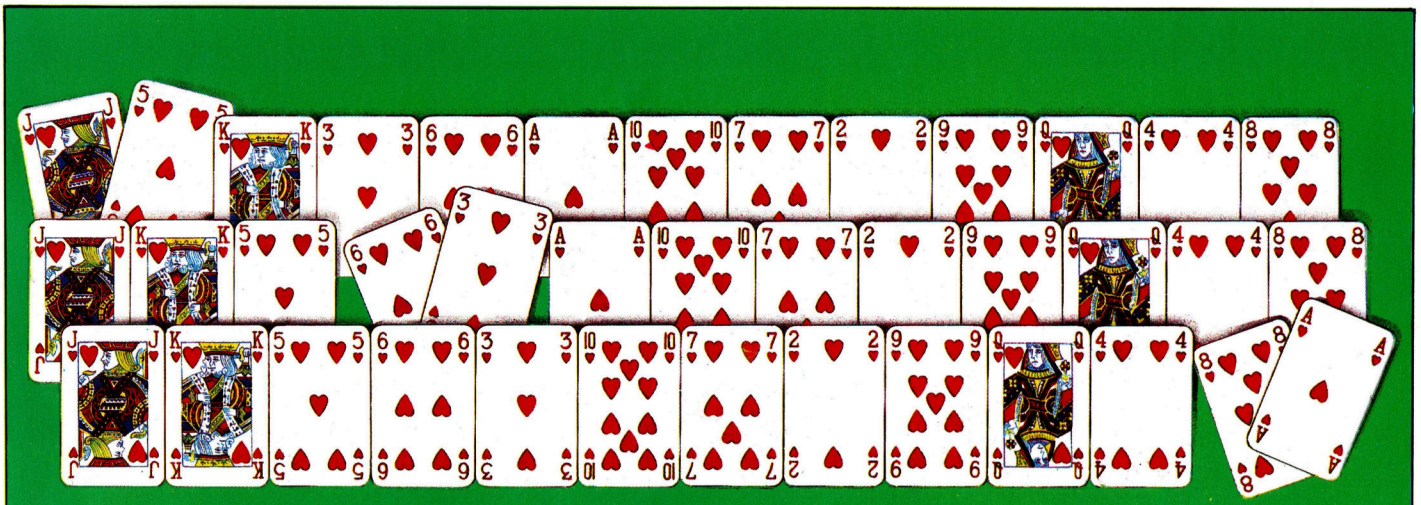
Ein weiteres Problem ist die Beendigung des Sortierlaufes. Ein Computer kann bis in die Ewigkeit fortfahren, Karten miteinander zu vergleichen, wenn man ihm nicht sagt, wann er damit aufhören soll. Die einzige sichere Regel dafür aber ist: Werden keine Karten mehr ausgetauscht, ist die Sortierung beendet. Diese Sortiermethode für kleine Datenmengen wird „Bubble Sort" genannt. Zu ihren Vorteilen zählen die leichte Programmierbarkeit und der geringe Speicherbedarf.

Unsere zweite Sortiermethode ähnelt daher der Methode, mit der wir normalerweise Karten sortieren würden:

1) Legen Sie die gemischten Karten ein zweites Mal in einer Reihe aus, und setzen Sie die Münze unter die zweite Karte von links. Die Karte, unter der sich die Münze befindet, nennen wir die „erste markierte Karte".

2) Nehmen Sie jetzt die erste markierte Karte aus der Reihe heraus, so daß eine Lücke entsteht. Legen Sie eine andere Münze unter die Karte links davon. Die Karte, unter der sich die zweite Münze befindet, nennen wir die „zweite markierte Karte".

3) Vergleichen Sie die erste markierte Karte mit der zweiten markierten Karte. Sind beide Karten in der richtigen Reihenfolge, dann legen Sie die erste markierte Karte wieder an ihren Platz zurück und springen Sie auf Schritt 4. Sind die Karten nicht in der richtigen Reihen-



Royal Flush

Der Bubble Sort läßt sich mit Hilfe eines Kartenspiels darstellen. Dabei soll als Ergebnis der König links außen liegen und das As rechts außen. Zuerst werden die beiden Karten am linken Ende der Reihe miteinander verglichen. Da sie nicht in der richtigen Reihenfolge liegen, werden sie miteinander vertauscht, dann werden die zweite und die dritte Karte verglichen und wiederum vertauscht. Bei dem fünften Vergleich findet diese Methode das As und versetzt es nach jedem weiteren Test um eine Karte nach links. Der zweite Durchlauf wiederholt dies, und die Zwei liegt jetzt neben dem As an der richtigen Stelle.

folge, legen Sie die zweite markierte Karte in die Lücke und rücken die zweite Münze um eine Position nach links und markieren damit eine neue Karte (befindet sich die zweite markierte Karte am linken Ende der Reihe, ist dies nicht möglich, legen Sie in diesem Falle die erste markierte Karte in die Lücke und springen Sie auf Schritt 4).

Vergleichen Sie jetzt die zweite markierte Karte mit der ersten markierten Karte (die nicht in der richtigen Reihenfolge lag), und wiederholen Sie Schritt 3, bis die korrekte Position der ersten markierten Karte gefunden ist.

4) Rücken Sie die erste Münze um eine Position nach rechts und wiederholen Sie Schritt 2 und 3, bis die Münze am rechten Ende der Reihe angekommen ist. Die Karten befinden sich jetzt in der richtigen Reihenfolge.

Dieses System wird Einsatzmethode (englisch: Insertion Sort) genannt und ähnelt der Methode, mit der die meisten Menschen ein Blatt sortieren. Obwohl diese Methode nicht so einfach zu programmieren ist wie der Bubble Sort, ist sie doch weitaus effektiver. Im weiteren Verlauf des Kurses werden wir auf einige komplexere Sortieralgorithmen eingehen.

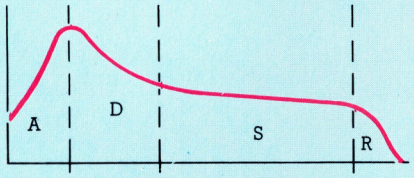
```
9  REM*****************************
10 REM* SORTIERALGORITHMEN *
11 REM*****************************
100 INPUT "WIEVIELE ZAHLEN SOLLEN SORTIERT
    WERDEN? ";LT
150 IF LT<3 THEN LET LT=3
200 LET LT=INT(LT)
250 DIM R(LT), C(LT)
300 LET Z=0:LET Q=0:LET P=0
350 LET I=1:LET O=0:LET II=2:LET TH=2
400 INPUT "WIEVIELE TESTS? ";N
450 FOR CT=I TO N
500 GOSUB 4000
550 FOR SR=I TO TH
600 GOSUB 5000
650 PRINT:PRINT:PRINT:PRINT
700 PRINT "TEST #";CT+SR/10
750 INPUT "START SORTIERUNG = RETURN";A$
800 PRINT "DIE UNSORTIERTE LISTE:"
850 GOSUB 3000
900 ON SR GOSUB 6000,7000
950 PRINT "DIE SORTIERTE LISTE:"
1000 GOSUB 3000
1050 NEXT SR
1100 NEXT CT
1150 END
2999 REM*********************
3000 REM* LISTE DRUCKEN *
3001 REM*********************
3100 FOR K=I TO LT
3200 PRINT R(K);
3300 NEXT K
3400 PRINT
3500 RETURN
3999 REM***********************
4000 REM* RND GENERATOR *
4001 REM***********************
4100 RANDOMIZE
4200 FOR K=I TO LT
4300 LET C(K)=INT(100*RND)
4400 NEXT K
4500 RETURN
4999 REM*************************
5000 REM* RND REGENERATOR *
5001 REM*************************
5100 FOR K=I TO LT
5200 LET R(K)=C(K)
5300 NEXT K
5400 PRINT:PRINT
5500 RETURN
5999 REM************
6000 REM* BUBBLE *
6001 REM************
6050 PRINT "BUBBLE SORT — ANFANG"
6100 FOR P=LT—I TO I STEP—I
6150 LET F= —I
6200 FOR Q=I TO P
6250 LET Z=Q+I
6300 IF R(Q)<R(Z) THEN LET D=R(Q):LET R(Q)=R(Z):LET
     R(Z)=D:LET F=O
6350 NEXT Q
6400 IF F= —I THEN LET P=I
6450 NEXT P
6500 PRINT "BUBBLE SORT — ENDE"
6550 RETURN
6999 REM***********************
7000 REM* EINSATZMETHODE *
7001 REM***********************
7050 PRINT "EINSATZMETHODE — ANFANG"
7100 FOR P=II TO LT
7200 LET D=R(P)
7300 FOR Q=P TO II STEP—I
7400 LET R(Q)=R(Q—I)
7500 IF D<=R(Q) THEN LET R(Q)=D:LET Q=II
7600 NEXT Q
7700 IF D>R(I) THEN LET R(I)=D
7800 NEXT P
7850 PRINT "EINSATZMETHODE — ENDE"
7900 RETURN
```

Dieses BASIC-Programm zeigt die Unterschiede zwischen dem Bubble Sort und der Einsatzmethode. Das Programm wurde auf Geschwindigkeit hin geschrieben und deshalb nicht dokumentiert. Es müßte auf fast allen Maschinen funktionieren; vergleichen Sie es aber mit den ON . . . GOSUB Routinen und den RND- und RANDOMIZE-Funktionen.

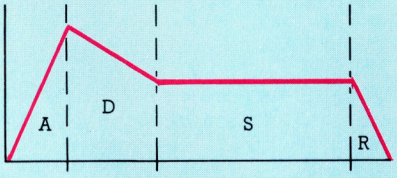
Sound-Begriffe

Hüllkurvengenerator, Schwingungsform, Rauschen und Tonausgabe

Unter Hüllkurve versteht man die Veränderungen, die sich in der Intensität eines Klanges während seiner Dauer vollziehen. In den Diagrammen unten finden Sie die Hüllkurven von Tönen mehrerer Instrumente aufgezeichnet. Hüllkurven unterteilen sich normalerweise in vier Teile: Attack, Decay, Sustain und Release (ADSR). Beim Anschlagen einer Taste auf dem Klavier steigt die Lautstärke des Tones sehr schnell bis auf ihren Höhepunkt an (Attack), um dann langsam wieder abzunehmen (Decay). Für einige Zeit behält der Ton eine fast konstante Lautstärke (Sustain-Ebene) und klingt relativ schnell aus (Release), wenn die Taste losgelassen wird. Das Sustain bezeichnet dabei eine bestimmte Lautstärke, während Attack, Decay und Release Zeitabschnitte sind. Ein Gerät, das alle vier Komponenten eines Tones steuern kann, wird ADSR-Generator genannt. Im Prinzip ist jede Maschine, die einen Klang an- und abstellen kann, eine Art Hüllkurvengenerator.



Natürlicher Klavierklang



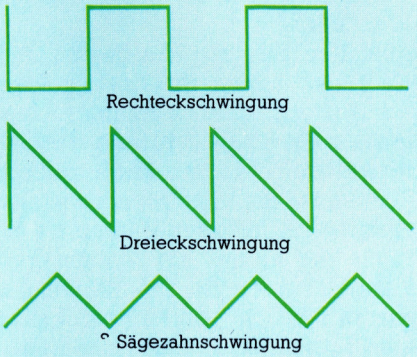
Synthetischer Klavierklang

Die meisten Computer verfügen über einen Tongenerator, der nur ein „Biep" erzeugen kann. Das heißt, sie können einen Klang nur für eine bestimmte Zeit und eine vorgegebene Lautstärke anschalten. In diesem Fall sind die A-, D-, und R-Komponenten gleich Null.

Den sich wiederholenden Umriß eines Signals, das durch einen Oszillator sichtbar gemacht wird, bezeichnet man als Schwingungsform. Die Form der Schwingung gibt dem Klang seinen Charakter (Klangfarbe). Zwei Instrumente können nie identisch klingen, selbst wenn beide den gleichen Ton spielen. Die Ursache dafür liegt in den unterschiedlichen Schwingungsformen, die sie erzeugen. Die bekanntesten Schwingungsformen sind die Rechteck- (auch Impuls genannt), die Dreiecks- und die Sägezahnschwingung – siehe Abbildung.

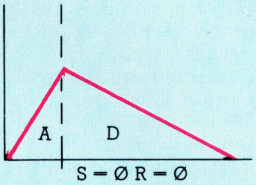
Auf den meisten Heimcomputern kann nur eine Schwingungsform – fast immer die Rechteckschwingung – erzeugt werden. Aus diesem Grund haben sie einen harten synthetischen Klang.

Vom Musikalischen her betrachtet ist der Commodore 64 im Augenblick der interessanteste Computer, da man jede der drei Schwingungsformen auf allen drei Oszillatoren ansprechen kann. Die Schwingungsformen können mit Hilfe von Filtern verändert werden, die ähnlich wie die Höhen- und Tiefenregulierung in einer HIFI-Anlage funktionieren und die Töne weicher klingen lassen. Sehr nützlich ist dabei, daß die Filtereinstellung während der Dauer eines Tones verändert werden kann. Damit können natürliche Töne täuschend ähnlich nachgeahmt und synthetische Töne interessanter gestaltet werden.

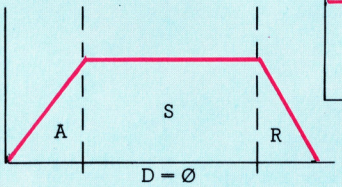


Rauschen ist eine komplexe Art von Klang und wird von Zufallsschwingungen erzeugt. Da das Ohr im Rauschen sich wiederholende Muster nicht erkennen kann, hört es auch keine bestimmte Tonhöhe. Beispiele dafür sind Regen, Wind und Donner. Rauschen klingt nicht immer gleich, da es aus einer zufälligen Zusammensetzung reines (weißen) Rauschens besteht, in das einige dominante Töne gemischt sind. Auf den meisten Microcomputern mit Rauscherzeugung kann man daher den Charakter des Rauschens verändern oder reine Klänge einmischen. Das Ergebnis: der Bereich vom Wind bis zu Explosionen.

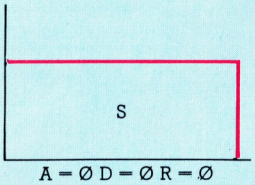
Normalerweise wird der Ton über den Lautsprecher des Monitors ausgegeben, er läßt sich aber auch über eine HIFI-Anlage wiedergeben.



S = Ø R = Ø
Xylophon



D = Ø
Violine



A = Ø D = Ø R = Ø
Orgel

Grafik-Spiele

Grafikspiele leicht gemacht mit den Befehlen PEEK und POKE.

Die Grundlagen der Computergrafik wurden bereits dargestellt. Nun folgt die Erzeugung auf dem Bildschirm. Zwei Hauptmethoden bieten sich dafür an: der Befehl POKE und der PRINT-Befehl mit dem ganzen Bereich seiner Steuerfunktionen.

Der POKE-Befehl lädt einen Wert in eine bestimmte Speicherstelle. Im Arbeitsspeicher des Computers ist ein Bereich für den Bildschirmaufbau reserviert. Für jede Position auf dem Schirm existiert dabei eine eigene Speicherstelle. Ein Standardbildschirm von 25 Zeilen mit je 40 Zeichen hat dafür 1000 Speicherstellen, von denen jede eine Zahl enthält, die einem bestimmten Zeichen des Zeichensatzes der Maschine entspricht. Dieser kann entweder auf dem ASCII-Code oder einem speziell vom Hersteller des Gerätes entwickelten Code basieren. Zusätzlich zu diesem Speicherbereich existiert oft ein weiterer, der die Farbe jeder einzelnen Position kennzeichnet.

Auf dem Commodore 64 zum Beispiel gibt es im BASIC nur wenige Befehle, mit denen eine Grafik gesteuert werden kann, daher werden hier die POKE-Kommandos für den Bildschirmaufbau verwendet. Die Speicheradressen für die auf dem Bildschirm dargestellten Zeichen befinden sich zwischen Adresse 1024 und 2023, während die Farbinformationen im Bereich von 55296 bis 56295 gespeichert sind. Auf dem Commodore hat der Buchstabe A den Bildschirmcode 1, die Farbe Schwarz ist durch eine 0 gekennzeichnet. Die Befehle, die ein schwarzes A in der obersten linken Ecke des Bildschirms darstellen, sind:

```
10 POKE 1024,1
20 POKE 55926,0
30 END
```

Durch eine einfache Änderung kann man auf der obersten Bildschirmzeile eine ganze Reihe von A's darstellen:

```
10 FOR X = 0 TO 39
20 POKE 1024+X,1
30 POKE 55926+X,0
40 NEXT X
50 END
```

Die A's werden durch eine FOR-NEXT-Schleife auf den Schirm gebracht, die die Speicherstelle für das Zeichen und die Farbe bei jedem Durchlauf um 1 erhöht. Bauen Sie nun einen Befehl ein, der die alten A's jedesmal löscht, wenn ein neues A auf dem Schirm erscheint, dann scheinen die A's über den Bildschirm zu laufen – eine einfache Form, Bewegung auf den Schirm zu bringen.

Der Leerzeichencode des Commodore ist 32. Man muß dieses Zeichen nur an die richtige Stelle bringen: exakt ein Zeichen hinter der Position, an der das neue A erscheinen wird. Fügen Sie diese Programmzeile ein:

```
35 POKE 1024+X,32
```

Oder auch:

```
15 POKE 1024+(X—1),32
```

Die Speicherstellen mit POKE zu laden ist eine mühselige Methode für die Grafik-Erzeugung. Hintergrundbilder programmiert man daher besser mit den Befehlen READ und DATA. Die meisten Heimcomputer verfügen über eine große Anzahl von Grafikbefehlen und machen es möglich, mit einigen wenigen Kommandos interessante farbige Bilder auf dem Schirm zu produzieren, die vollständige geometrische Formen zeichnen. Mit diesen Anweisungen können einzelne Punkte auf dem Schirm definiert werden, die sich dann zu Linien verbinden lassen. Auch Kreise, Bögen und Quadrate werden als Ganzes gezeichnet und mit Farbe ausgefüllt.

Eine schnelle Methode, bestimmte Formen zu löschen, ist das einfache Nachzeichnen der alten Form in der Hintergrundfarbe. Das schnelle Zeichnen, Löschen und Nachzeichnen ist auch die Grundlage für interessante Grafikprogramme. Je schneller sich die Elemente einer Grafik bewegen können, desto realistischer sehen ihre Bewegungen aus.

Wegen ihrer Geschwindigkeit bietet sich dafür die Verwendung von Sprites an, deren Darstellung auf der vorhergehenden Position nicht gelöscht werden muß, bevor sie bewegt werden.

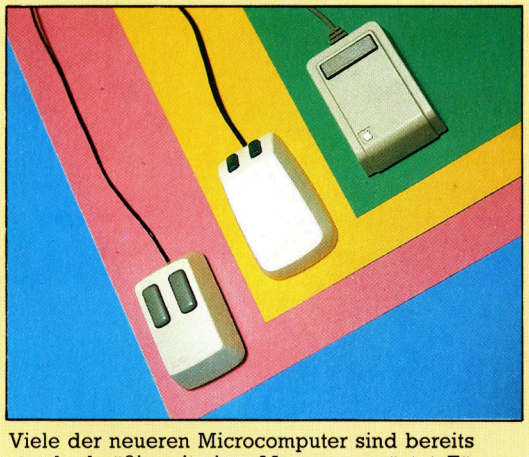
Sternenbilder
Hier ein kurzes PRINT-Programm für den Dragon. Die Variable X ist die Bildschirmposition, auf die ein Stern gedruckt werden soll. Zeile 40 löscht den alten Stern, sobald der neue gedruckt wurde.

```
10 CLS: REM BILDSCHIRM LOESCHEN
20 FOR X = 160 TO 191
30 PRINT @X, "*";
40 PRINT @X—1," ";
50 NEXT X
60 END
```

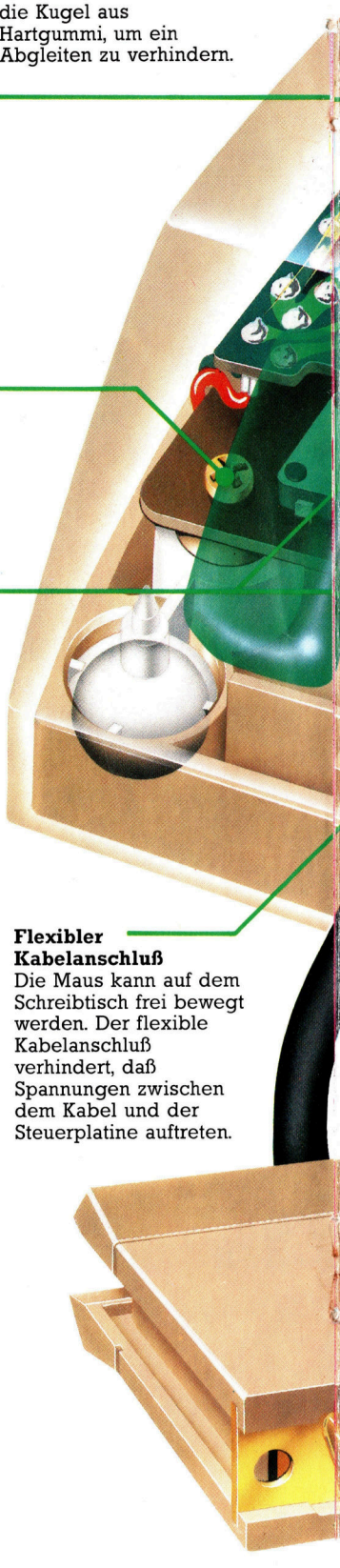
Mäuse im Einsatz

Mit der Maustechnik gehören komplizierte Eingaben über die Tastatur der Vergangenheit an.

Noch vor wenigen Jahren konnten Computer Informationen nur über sperrige elektromechanische Schreibmaschinen empfangen, die „Teletype-Maschinen" genannt wurden. Diese Geräte waren laut, unzuverlässig und umständlich zu bedienen. Inzwischen sind fast nur noch Bildschirmterminals (VDU – englisch: Visual Display Unit) mit Tastatur im Einsatz, die weitaus schneller und leiser arbeiten. Diese Gerätekombination beseitigte viele der Probleme, die die Teletype-Maschinen mit sich brachten. So entfielen beispielsweise die Unmengen von Lochstreifenpapier, die nach dem Eingeben von Programmen oder Daten nicht mehr zu verwenden waren. Bei beiden Geräten – der alten Teletype und dem modernen Bildschirmgerät mit Tastatur – müssen jedoch immer noch alle Eingaben Zeichen für Zeichen oder Zeile für Zeile erfolgen. Es ist nicht möglich, den Cursor schnell über den gesamten Bildschirm zu bewegen, an einer Stelle des Bildschirms ein Programm aus dem Menü auszuwählen und dann direkt an eine andere Stelle zu gehen, um Daten zu verändern oder eine neue Datei zu eröffnen. Man ist an das umständliche Format der Tastatur und an den Cursor gebunden. Unabhängig von der Tastatur ist man nur auf speziellen Grafikterminals und in Computerspielen, bei denen die Steuerung mit Joysticks oder Track Balls erfolgt. Mit Geräten dieser Art kann ein kommerzieller Anwender jedoch nichts anfangen.

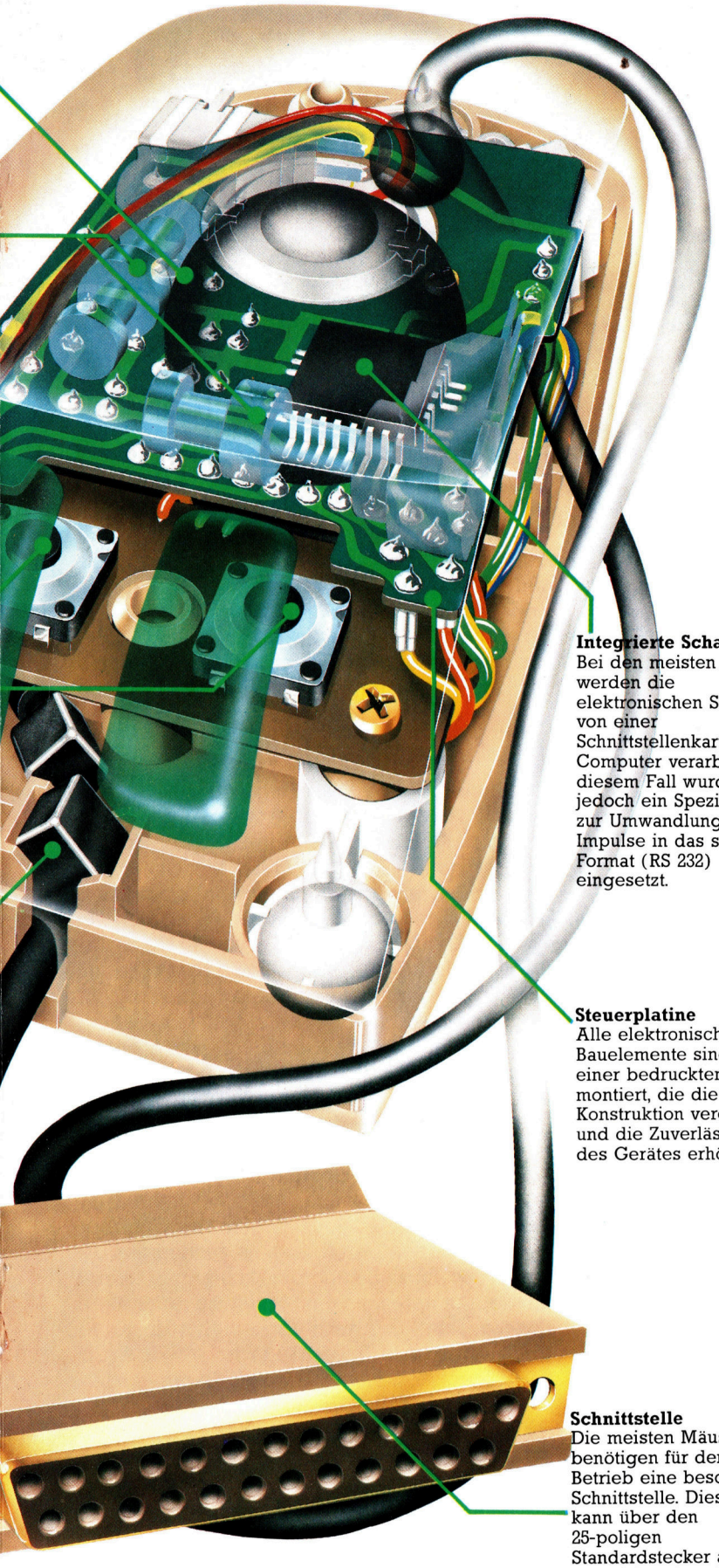


Viele der neueren Microcomputer sind bereits standardmäßig mit einer Maus ausgerüstet. Für ältere Maschinen gibt es Erweiterungsmodule. Die meisten Mäuse arbeiten mit einer rotierenden Kugel auf der Unterseite, während sich oben bis zu drei Eingabetasten befinden.



In den sechziger Jahren arbeitete das Stanford Research Institute in Kalifornien als erstes an einer Lösung dieses Problems. 1970 wurde die erste ‚Maus' – so wurde der neue Steuermechanismus getauft – zum Patent angemeldet. Seinen Namen erhielt das Steuergerät aufgrund seines Aussehens: Die Maus war klein genug, um in eine Hand zu passen; sie hatte einen „Schwanz" (das Kabel); und die ersten Geräte besaßen außerdem noch zwei „Ohren" (die Steuerknöpfe).

Die Maus arbeitet nach folgendem Prinzip: Bewegt man sie über eine glatte Oberfläche, analysiert ein Mechanismus in ihrem Innern die Bewegung nach links/rechts oder oben/unten und die Kombination dieser vier Richtungen. Diese Bewegungen werden direkt als Steuersignale des Cursors auf dem Bildschirm umgesetzt. Hauptsächlich werden zwei Methoden angewandt, um die Bewegungen der Maus in elektrische Signale umzusetzen. In beiden Fällen befindet sich auf der Unterseite der Maus eine große Kugel, die mit der Oberfläche Kontakt hat, während die Maus bewegt wird. Die Bewegung der Kugel überträgt sich auf Kontaktrollen. Ein System arbeitet mit codierten Rädern, die mit einem Ende der Rollen verbunden sind und mit Streifen von leitendem und nicht-leitendem Material beschichtet wurden. Bewegt sich die Kugel, werden die auf diese Weise generierten Impulse von dem



Integrierte Schaltung
Bei den meisten Mäusen werden die elektronischen Signale von einer Schnittstellenkarte im Computer verarbeitet. In diesem Fall wurde jedoch ein Spezialchip zur Umwandlung der Impulse in das serielle Format (RS 232) eingesetzt.

Steuerplatine
Alle elektronischen Bauelemente sind auf einer bedruckten Platine montiert, die die Konstruktion vereinfacht und die Zuverlässigkeit des Gerätes erhöht.

Schnittstelle
Die meisten Mäuse benötigen für den Betrieb eine besondere Schnittstelle. Diese Maus kann über den 25-poligen Standardstecker an jede RS-232-Schnittstelle angeschlossen werden.

Steuerprogramm der Maus gezählt und damit die Position des Cursors auf dem Bildschirm bestimmt. Das andere System verwendet für diesen Zweck zwei Scheiben, die mit Schlitzen versehen und auf die Kontaktrollen montiert sind. Auf der einen Seite der Scheiben befindet sich eine Lichtquelle und auf der anderen eine Photozelle. Die durch die Schlitze fallenden Lichtimpulse werden in elektrische Signale umgewandelt, die jetzt wie bei dem mechanischen System den Cursor steuern.

Ist der Cursor auf die gewünschte Stelle des Bildschirms geführt worden, kann seine Position über eine der Eingabetasten an den Computer weitergegeben werden. Diese Anzahl der Eingabetasten ist von Hersteller zu Hersteller verschieden. Manche Systeme verwenden drei Tasten, die Firma Microsoft entschied sich für zwei, und Apple kommt mit einem Druckknopf aus. Die Tasten können für die Programmwahl aus einem Menü – wie zum Beispiel bei dem Programm ‚MultiTool Word' von Microsoft – und für die Steuerung des Cursors benutzt werden.

Schnelle Korrekturen

Die Mäuse mit ihren speziell dafür geschriebenen Programmen haben den Vorteil, daß sie leicht bedient werden können, auch wenn man mit Tastaturen und Computern nicht vertraut ist. Statt einen Programmnamen einzugeben oder eine Folge von Tasten zu drücken, bewegt der Anwender die Maus einfach an die Stelle des Bildschirms, an dem eine bestimmte Funktion aufgeführt ist, und drückt eine einzige Taste, um diese Funktion zu aktivieren.

Leider läßt sich auch mit der Maus nicht vollständig auf die Tastatur verzichten – neue Texte und Zahlen müssen auch weiterhin per Tastatur in den Computer eingegeben werden. Korrekturen und Umstellungen bereits vorhandener Daten lassen sich jedoch sehr vereinfachen. Die bei der Entwicklung des ‚Lisa' von Apple durchgeführten Tests ergaben, daß ein Anwender, der mit dem Computer nicht vertraut ist, schon nach 15 Minuten mit den mausgesteuerten Programmen und Funktionen umgehen kann. Vergleichbare Programme auf herkömmlichen Systemen benötigen dafür eine Einarbeitungszeit bis zu 20 Stunden, da besonders bei der Bedienung der Tastatur Probleme auftreten und umständliche Befehle erlernt werden müssen. Ältere Heimcomputer-Modelle, die nicht für den Maus-Betrieb eingerichtet sind, können (zumindest bei einigen Herstellern) mit einem Erweiterungsmodul ausgestattet werden.

Elektronische Mäuse werden bald zur Standardausrüstung der Heimcomputer gehören. Sie vereinfachen die Arbeit, sind leicht zu bedienen und schrecken den Anfänger nicht so sehr wie eine konventionelle Tastatur, die oft verwirrend wirkt.

Neue Dimensionen

Wie wir bereits gesehen haben, können in eindimensionalen Feldern Daten gespeichert werden, die alle etwas gemeinsam haben. Zweidimensionale Felder werden dagegen für Tabellen und Listen verwendet, da hier mehrere Komponenten Berücksichtigung finden.

Bisher haben wir zwei verschiedene Variablentypen kennengelernt, einfache und verschachtelte Variablen. Einfache Variablen kann man mit Speicherstellen vergleichen, in denen man Zahlen (oder Zeichenketten) speichern kann. Um wieder an die gespeicherten Daten heranzukommen, muß man den Variablennamen (z. B. LABEL) verwenden. Mit einfachen Variablen kann man nur einen Wert oder String speichern.

Außerdem haben sie auch nur „einfache" Variablen-Namen, wie zum Beispiel N, B2, X oder Y3. Mit verschachtelten Variablen, auch als eindimensionale Felder bezeichnet, kann man ganze Listen an Werten und Strings speichern. Die Anzahl an Werten oder Strings, die man maximal speichern kann, wird am Anfang des Programmes durch die DIM-Anweisung festgelegt.

Zum Beispiel bewirkt die Anweisung DIM A (16), daß das Feld mit dem Namen A 16 verschiedene Werte aufnehmen kann. Beachten Sie, daß viele BASIC-Versionen den durch A(0) bezeichneten Teilbereich bereits als erstes Element akzeptieren. Bei einer solchen BASIC-Version würden durch die Anweisung DIM A(16) nicht nur 16, sondern sogar 17 Speicherstellen reserviert. Die einzelnen Speicherstellen werden über die bereits erwähnten Untervariablennamen angesprochen.

Die Anweisung PRINT A(1) gibt den Inhalt der ersten Speicherstelle des Feldes aus. LET B=A(12) weist der Variablen B den Inhalt der zwölften Speicherstelle des Feldes zu. LET A(3) = A(5) weist den Inhalt der fünften Speicherstelle der dritten Speicherstelle des Feldes zu. Betrachten Sie das Beispiel einer Tabelle der Haushaltsausgaben über den Zeitraum von einem Jahr:

	MIETE	TELEFON	STROM	ESSEN	AUTO
JAN	500.00	54.50	89.40	423.00	145.00
FEB	500.00	47.30	94.30	444.35	155.20
MÄR	500.00	59.20	110.50	410.20	132.30
APR	500.00	56.90	78.30	389.40	167.70
MAI	500.00	67.30	92.80	445.70	145.80
JUN	500.00	62.70	97.30	432.45	151.20
JUL	500.00	45.50	86.40	478.75	149.30
AUG	500.00	52.00	94.20	430.20	150.40
SEP	500.00	59.40	99.10	453.55	143.00
OKT	500.00	79.20	114.30	472.15	158.40
NOV	500.00	72.70	105.20	435.25	154.30
DEZ	500.00	84.50	132.70	568.30	185.40

Zweidimensionale Datenfelder werden hauptsächlich für Tabellen verwendet. Vor der Dateneingabe muß der benötigte Speicherplatz mit der DIM-Anweisung z. B. DIM X(30,40) reserviert werden.

Erfaßt man solche Informationen in dieser Form, so kann man sie relativ leicht auf verschiedene Art und Weise verarbeiten. Es ist zum Beispiel sehr einfach, die Gesamtausgaben des Monats März herauszufinden, indem man alle Einzelwerte der Reihe März addiert. Genauso einfach ist es, wenn man die Gesamtausgaben für Telefon oder Auto wissen möchte. Man braucht nur die einzelnen Werte der entsprechenden Spalte zu addieren. Eine weitere Möglichkeit besteht darin, monatliche oder jährliche Durchschnittswerte zu ermitteln. Eine solche Tabelle wird als zweidimensionales Feld bezeichnet. Es hat 12 Reihen und fünf Spalten.

Zweidimensionale Felder

Zweidimensionale Felder, wie dieses, können in BASIC ähnlich entwickelt werden wie eindimensionale Felder. Der einzige Unterschied besteht darin, daß die Variable jetzt jeweils zwei Angaben zur Auffindung einer bestimmten Speicherstelle benötigt. Wenn wir ein BASIC-Programm schreiben, in dem wir diese Tabelle mit Informationen verwenden wollen, ist es am einfachsten, die gesamte Tabelle als ein einziges zweidimensionales Feld zu behandeln. Genau wie einem normalen eindimensionalen Feld geben wir ihm einen Variablen-Namen. Nennen wir es A (für „Array"=Feld). Jetzt müssen wir es, wieder genau wie bei einem eindimensionalen Feld, DIMensionieren. Da wir 12 Reihen und 5 Spalten zu verarbeiten haben, muß die Anweisung wie folgt lauten: DIM A(12,5). Die Reihenfolge, in der die Unterbezeichnung angegeben wird, ist wichtig. Im allgemeinen ist man übereingekommen, zuerst die Reihen und dann die Spalten zu definieren. Unsere Tabelle enthält 12 Reihen (für jeden Monat eine) und 5 Spalten (eine für jede der fünf Kategorien an Ausgaben). Daraus ergibt sich, daß es ein 12-mal-5-Feld ist.

Die DIM-Anweisung beinhaltet zwei wesentliche Funktionen. Zum einen reserviert sie die für das Feld notwendigen Speicherstellen im Speicher des Computers. Zum anderen gestattet sie, jede der einzelnen Speicherstellen innerhalb des Feldes anzusprechen. Hierzu muß der Variablen-Name gefolgt von der Spalten- und Reihen-Position in Klammern angegeben

werden. Die DIM-Anweisung DIM X(3,5), zum Beispiel, würde ein Variablenfeld mit Namen X und drei Spalten sowie fünf Reihen zur Verfügung stellen.

Betrachten Sie die Tabelle und nehmen Sie an, daß die Informationen als Elemente eines zweidimensionalen Feldes mit dem Namen A eingegeben wurden. Finden Sie die Werte, die in A(1,1), A(1,5), A(2,1), A(3,3) und A(12,3) abgelegt sind. Man kann die Tabelleneintragungen mittels LET-Anweisungen innerhalb eines Programms eingeben. Betrachten Sie unser Beispiel:

```
30 LET A(1,2) = 54.5
40 LET A(1,3) = 89.4
50 LET A(1,4) = 423
:
:
610 LET A(12,5) = 185.4
```

Eine weitaus einfachere Zuordnungsmethode ist die Verwendung von READ und DATA-Anweisungen oder die INPUT-Anweisung in Verbindung mit verschachtelten FOR . . . NEXT-Schleifen.

```
10 DIM A(12,5)
20 FOR R=1 TO 12
30 FOR C=1 TO 5
40 READ A(R,C)
50 NEXT C
60 NEXT R
70 DATA 500.00, 54.50, 89.40, 423.00,
   145.00, 500.00, 47.30, 94.30
80 DATA 444.35, 155.20, 500.00, 59.20,
   110.50, 410.20, 132.30, 500.00
90 DATA 56.90, 78.30, 389.40, 167.70,
   500.00, 67.30, 92.80, 445.70
100 DATA 145.80, 500.00, 62.70, 97.30,
    432.45, 151.20, 500.00, 45.50
110 DATA 86.40, 478.75, 149.30, 500.00,
    52.00, 94.20, 430.20, 150.40
120 DATA 500.00, 59.40, 99.10, 453.55,
    143.00, 500.00, 79.20, 114.30
130 DATA 472.15, 158.40, 500.00, 72.70,
    105.20, 435.25, 154.30, 500.00
140 DATA 84.50, 132.70, 568.30, 185.40
150 END
```

Bei diesem Programm sind einige Punkte zu beachten, z. B. die DIM-Anweisung gleich in der ersten Programmzeile des Programms. Eine DIM-Anweisung sollte innerhalb eines Programmes nur einmal ausgeführt werden. Daher ist es üblich, sie möglichst ganz am Anfang zu plazieren, bevor irgendwelche Schleifen ausgeführt werden. Der zweite Punkt, der zu beachten ist, besteht darin, daß wir es in diesem Programm mit zwei FOR . . .NEXT-Schleifen zu tun haben. Die eine dient für die Zuordnung der Reihen und die andere für die Spalten. Diese beiden Schleifen werden nicht nacheinander aktiviert, sondern gleichzeitig. Sie sind ineinander verschachtelt. Beachten Sie die gewählten Schleifenbereiche. FOR R=1 TO 12 zählt den Wert für die Reihen von 1 bis 12. FOR C=1 TO 5 zählt den Wert der Spalten von 1 bis 5.

Beim ersten Durchlauf der beiden Schleifen, also nachdem die Programmzeilen 20 und 30 ausgeführt wurden, haben R und C beide den Wert 1. Somit lautet die READ-Anweisung in Zeile 40 READ A(1,1). Der erste Datenwert der DATA-Anweisung ist 500. Gemäß dem Programm wird dieser Wert der ersten Spalte und der ersten Reihe des Feldes zugeordnet.

Nachdem der eben beschriebene Vorgang beendet ist, wird der Programmlauf durch die NEXT C-Anweisung zur Zeile 30 verzweigt, und der Wert von C wird auf zwei erhöht. Zeile 40 entspricht nun READ A(1,2), und der nächste Datenwert (54.50) wird der ersten Reihe und der zweiten Spalte unseres Feldes zugeordnet. Dieser Vorgang wird jetzt solange durchgeführt, bis C auf den Wert 5 erhöht wird. Wenn dies passiert, wird der Programmlauf durch die NEXT R-Anweisung in Zeile 60 zu Zeile 20 verzweigt, und der Wert von R wird auf 2 erhöht. In Zeile 30 wird der Wert von C wieder auf 1 gesetzt. Jetzt entspricht Zeile 40 der Anweisung READ A(2,1).

Betrachten wir nun die DATA-Anweisung. Beachten Sie, daß die Kommas zur Trennung der einzelnen Informationen verwendet werden. Vor der ersten und nach der letzten Information darf allerdings kein Komma stehen. Zwischen jede Information haben wir Leerstellen gesetzt, was jedoch nicht nötig ist. Seien Sie bei der Dateneingabe aufmerksam, denn es kann leicht zu Fehleingaben kommen, die dann später nur noch sehr schwer wiederzufinden sind. Die Anzahl an DATA-Anweisungen ist nicht limitiert. Verwenden Sie so viele, wie Sie für das Programm brauchen. Die einzelnen Informationen werden nacheinander gelesen, beginnend bei der ersten DATA-Anweisung bis zur letzten, bis alle eingelesen wurden. Achten Sie darauf, daß die Anzahl der DATA-Werte korrekt ist, da Sie sonst beim Programmlauf eine Fehlermeldung erhalten.

Gesamtwerte

Damit man mit dem Programm etwas Nützliches anfangen kann, müssen wir es erweitern. Der erste Schritt bei der Entwicklung eines Programmes zur Verwaltung der Haushaltsfinanzen sollte eine zusammenfassende Beschreibung der notwendigen Eigenschaften sein. So müssen wir uns zum Beispiel entscheiden, ob wir Gesamtwerte und Durchschnitte für monatliche Ausgaben einer Kategorie (z. B. Stromausgaben) berechnen wollen oder nicht. Und wir können ausarbeiten, wie wir die Ergebnisse zu einem späteren Zeitpunkt weiterverarbeiten wollen.

Eine weitere Ausarbeitung wird zeigen, daß

Welche Werte werden bei den folgenden Befehlen ausgegeben?
PRINT A(12,1)
PRINT A(1,5)
PRINT A(5,1)
PRINT A(5,5)

wir Unterroutinen zur Berechnung der Gesamtausgaben eines Monats oder für Kategorien (MONATTOTAL und KATTOTAL), durchschnittliche monatliche Ausgaben (MDURCHSCHN) und durchschnittliche jährliche Ausgaben nach Kategorien (KDURCHSCHN), benötigen. Wir bezeichnen diese Unterroutinen mit aus einem Wort bestehenden Namen, damit wir das Programm einfacher planen können. Bei weiteren Überlegungen könnten wir entscheiden, sogar den Programmteil des Hauptmenüs wie eine Unterroutine zu behandeln. So wird gewährleistet, daß der Hauptteil des Programms ein eigenständiges Modul bleibt. Jetzt sieht unsere Programmübersicht schon etwas anders aus:

```
HAUPTPROGRAMM (DATENEINGABE)
  MENUE (RUFT UNTERROUTINEN)
ENDE

** UNTERROUTINEN **

1 MENUE
2 GESAMTAUSGABEN
3 DURCHSCHNITTSWERTE

(2) GESAMTAUSGABEN
4 MONATTOTAL
5 KATTOTAL

(3) DURCHSCHNITTSWERTE
6 MDURCHSCHN
7 KDURCHSCHN
```

Diese Übersicht über die einzelnen Programmteile zeigt uns, daß die Unterroutine MENUE die Auswahl zwischen GESAMTAUSGABEN und DURCHSCHNITTSWERTEN gestattet. Diese beiden Programmteile sind ebenfalls Unterroutinen. Die GESAMTAUSGABEN-Unterroutine ermöglicht eine weitere Wahlmöglichkeit zwischen MONATTOTAL und KATTOTAL. Diese beiden Unterroutinen sind diejenigen, die die aktuellen Berechnungen ausführen.

Die DURCHSCHNITTSWERTE-Unterroutine gibt uns die Wahlmöglichkeit zwischen MDURCHSCHN und KDURCHSCHN. Auch hier sind dies die Routinen, die die Berechnungen ausführen. Nun können wir mit der Programmierung der Module (Unterroutinen) beginnen. Die einzige Änderung, die am Hauptprogramm durchgeführt werden muß, ist der Aufruf einer Unterroutine vor der END-Anweisung. Das sieht dann so aus:

```
145 GOSUB ** MENUE **
```

Beachten Sie, daß wir immer noch Namen statt Zahlen als Kennzeichnung der Unterroutinen verwenden. Viele Programmiersprachen, wie zum Beispiel Pascal, gestatten den Aufruf von Unterroutinen über Namen. Bei den meisten BASIC-Versionen ist dies leider nicht möglich. Es werden direkte Zahlenangaben, also Zeilennummern, benötigt. Wir bleiben trotzdem bei den Namen, da diese Details auch nachträglich geändert werden können.

Überlegen Sie, wie man die MENUE-Unterroutine schreiben kann.

```
REM DIE ** MENUE ** UNTERROUTINE
PRINT "WUENSCHEN SIE G(ESAMTAUSGA-
  BEN)  ODER D(URCHSCHNITTSWERTE)?"
PRINT "WAEHLEN SIE G ODER D"
INPUT L$
IF L$="G" THEN GOSUB * GESAMTAUSGA-
  BEN *
IF L$="D" THEN GOSUB * DURCHSCHNITTS-
  WERTE *
RETURN
```

ANMERKUNG: Wir haben die Unterroutinen jeweils durch Sterne gekennzeichnet. Sie müssen statt der Namen Zahlen verwenden.

Nehmen wir einmal an, Sie geben G für GESAMTAUSGABEN ein. Das Programm ruft dann die Unterroutine GESAMTAUSGABEN auf. Dadurch würde ein anderes Menue dargestellt:

```
REM DIE ** GESAMTAUSGABEN ** UNTER-
  ROUTINE
PRINT "WUENSCHEN SIE GESAMTAUSGA-
  BEN"
PRINT "NACH M(ONAT) ODER K(ATEGORIE)"
PRINT "TIPPEN SIE M ODER K"
INPUT L$
IF L$="M" THEN GOSUB * MONATTOTAL *
IF L$="K" THEN GOSUB * KATTOTAL *
RETURN
```

Nehmen wir an, Sie haben M für MONATTOTAL gewählt. Lassen Sie uns versuchen, eine Routine zu schreiben, die die Gesamtausgaben eines jeden Monats eines Jahres berechnen kann.

```
REM DIE ** MONATTOTAL ** UNTERROU-
  TINE
REM SIE BERECHNET DIE GESAMTAUSGA-
  BEN
REM FUER JEDEN BELIEBIGEN MONAT
PRINT "WAEHLEN SIE DEN MONAT"
PRINT "1-JAN 2-FEB 3-MAERZ 4-APR 5-MAI"
PRINT "6-JUN 7-JUL 8-AUG 9-SEP"
PRINT "10-OKT 11-NOV 12-DEZ"
PRINT "GEBEN SIE DIE MONATSNUMMER
  EIN"
LET T=O
INPUT M
FOR C=1 TO 5
LET T=T+A(M,C)
NEXT C
PRINT "DIE GESAMTAUSGABEN FUER DEN
  MONAT"
PRINT "NUMMER ";M;" BETRAGEN ";T
RETURN
```

Antworten zu den Übungen der letzten Lektion:

RND-Funktion

```
40 IF R > 6 THEN LET R=1
```

Schleife und Durchschnittszahl

```
5 FOR L=1 TO 100
:
80 LET T=T+R
90 NEXT L
100 LET A=T/100
110 PRINT A
120 END
```

INKEY$

```
10 PRINT "DRUECKEN SIE EINE TASTE"
20 LET A$=INKEY$
30 IF A$=" " THEN GOTO 20
40 PRINT "DIE TASTE, DIE SIE GEDRUECKT HABEN,
   WAR "; A$
50 END
```

(Auf einigen Computern muß zusätzlich die Zeile:

```
15 INKEY$ <>" " THEN GOTO 15
```

eingefügt werden.)

Taktschleife

```
5 PRINT "DRUECKEN SIE DIE LEERTASTE NACH 10 SE-
  KUNDEN"
10 FOR L=O TO 1
20 LET R=R+1
30 IF INKEY$=" " THEN GOTO 60
40 LET L=O
50 NEXT L
60 PRINT "DER WERT VON R IST NACH
   10 SEKUNDEN ";R
70 END
```

IF-THEN-Test

```
10 GOSUB 1000
20 PRINT "DENKEN SIE SICH EINE ZAHL"
30 FOR G=1 TO 5
40 INPUT N
50 IF N > R THEN GOTO 110
60 IF N < R THEN GOTO 130
70 IF N=R THEN GOTO 150
80 NEXT G
90 PRINT "KEIN WEITERER VERSUCH,
   SIE HABEN VERLOREN!"
100 GOTO 500
110 PRINT "SIE LIEGEN ZU HOCH"
120 GOTO 80
130 PRINT "SIE LIEGEN ZU NIEDRIG"
140 GOTO 80
150 PRINT "SIE HABEN RICHTIG GERATEN, GRATU-
    LIERE"
500 END
1000 REM ** RND-UNTERPROGRAMM **
```

(Setzen Sie hier die Unterroutine ein.)

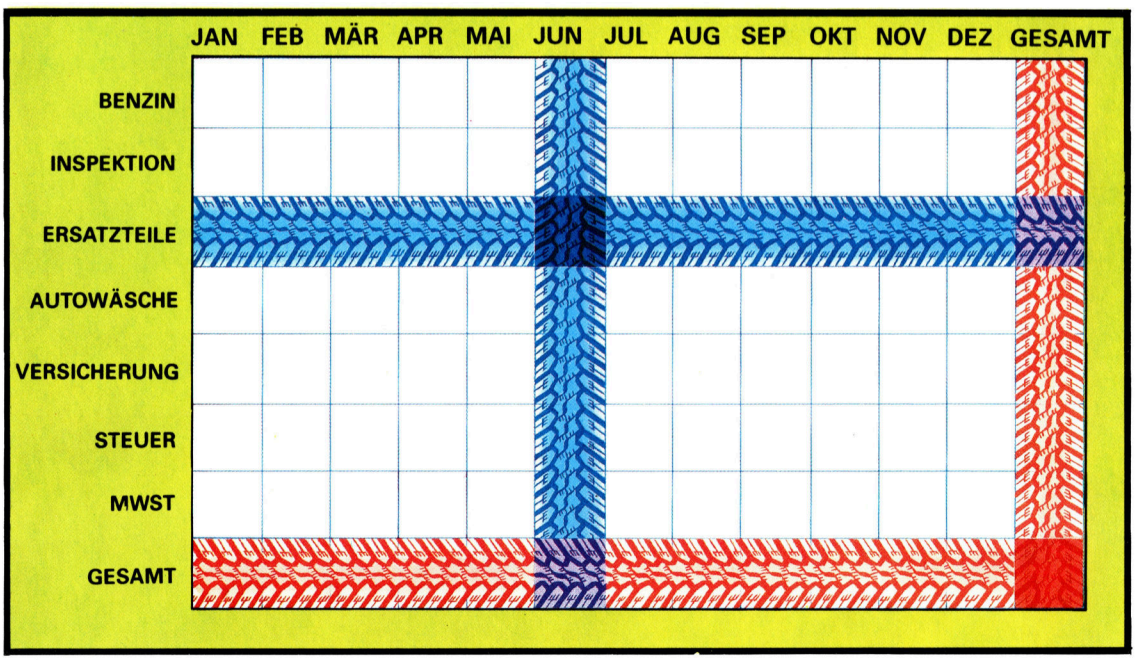
```
1020 RETURN
```

Die Nummer für den gewünschten Monat wird eingegeben. Durch die INPUT-Anweisung wird die Nummer der Variablen M (MONAT) zugeordnet. M wird verwendet, um die Reihe des zweidimensionalen Feldes A zu bestimmen. Die FOR-NEXT-Schleife erhöht den Wert von C (für die Spalte) von eins bis fünf. Nehmen wir an, wir hätten die Nummer 3 für März eingegeben. Beim ersten Durchlauf der Schleife entspricht die LET-Anweisung LET T=T+A(3,1). Beim zweiten Durchlauf lautet sie dann LET T=T+A(3,2) und so weiter.

Übungen

- Zuordnen von Werten: Schreiben Sie ein Programm, das den Elementen (BENZIN, STEUER usw.) der Matrix Werte zuordnet. Als nächstes schreiben Sie eine Unterroutine, die nach einem Monat und einer Ausgabenkategorie fragt. Zusätzlich soll sie den Inhalt der spezifizierten Stelle ausgeben. Als letztes schreiben Sie eine Unterroutine, die das Ergebnis jeder Spalte ermittelt und dann dieses Ergebnis in der ganz unten befindlichen Box ablegt. Verfahren Sie genauso mit den Reihen und ermitteln Sie dann das Gesamtergebnis. Legen Sie es in der ganz rechts unten befindlichen Box ab.
- FEHLER: Das folgende Programm würde nicht einwandfrei funktionieren. Korrigieren Sie die zwei Fehler.

```
10 DIM A(3,4)
20 FOR R = 1 TO 3
30 FOR C = 1 TO 4
40 READ A(R,C)
50 NEXT C
60 NEXT R
70 FOR X = 1 TO 3
90 FOR Y = 1 TO 4
100 PRINT A(Y,X)
110 NEXT Y
120 NEXT X
130 DATA 2, 4, 6, 8,
      10, 12, 14, 16, 18,
      20, 22
140 END
```



Das Bild zeigt eine Matrix (Feld), bestehend aus 8 mal 13 Einzelfeldern. Die Reihen repräsentieren die verschiedenen Elemente der Kosten für den Unterhalt eines Autos. Die Spalten repräsentieren die verschiedenen Monate des Jahres. Befolgen Sie die Angaben in der Übung „Zuordnen von Werten", um die jährlichen Kosten zum Unterhalt eines Autos zu berechnen.

Schöne Aussichten

Seit superschnelle Rechner Satellitenbilder auswerten und Tendenzen analysieren, sind auch die Wetterberichte um einiges genauer geworden.

Ergebnisse umfangreicher Computerarbeit begleiten uns durch den Alltag, ohne daß wir uns dessen immer bewußt sind. Zu den fortschrittlichsten Computeranwendungen gehört der tägliche Wetterbericht, der eine extrem hohe Datenverarbeitungskapazität voraussetzt. Bedenkt man die vielen Einflüsse, die bei einer Wetterprognose zu berücksichtigen sind, überrascht es, wie oft die Voraussage zutrifft. Dies ist ein Erfolg, den der Computer für sich verbuchen kann.

Die klimatischen Faktoren, die das Wetter über den britischen Inseln und auch über dem atlantischen Küstengebiet der europäischen Landmasse beeinflussen, sind sehr komplexer Natur. Sie werden hauptsächlich durch die Nähe des Nordpols und des Atlantischen Ozeans bestimmt. Die Gebiete an der Ostseite des Atlantiks sind für die klimatischen Auswirkungen innerhalb des 2500-Meilen-Gürtels dieser Zone recht anfällig. Ursache ist der sogenannte Coriolis-Effekt, der in der Ost-West-Drehung der Erde begründet ist. Er wird verständlich, wenn man sich vor Augen hält, daß sich ein Gegenstand am Äquator mit einer Geschwindigkeit von mehr als 1600 km/h fortbewegt. Diese kräftige Drehbewegung in Verbindung mit der normalen Pol-Äquator-Luftströmung erzeugt in der nördlichen Hemisphäre eine vorherrschend westlich gerichtete Wetterlage (Winde entstehen im Westen). Dieser ständige Ansturm feuchter Luft (er schwankt je nach den örtlichen Temperaturunterschieden) bestimmt das Wetter in Mitteleuropa.

Die „Wetterfrösche" stützen sich hauptsächlich auf Beobachtungen von Wetterschiffen, Bojen, Ballons und Flugzeugen, die Daten über die Wetterentwicklung von strategisch wichtigen Punkten liefern. Die Meteorologen „sehen" dann voraus, was geschieht, wenn dieses Atlantikwetter den Kontinent erreicht. Dabei

Der im Juni 1981 gestartete Wettersatellit Meteosat 2 befindet sich auf einer geostationären Umlaufbahn etwa 35 880 km über dem Äquator und sammelt Informationen von vielen Erdstationen.

hilft ihnen die Kenntnis früherer Wetterentwicklungen.

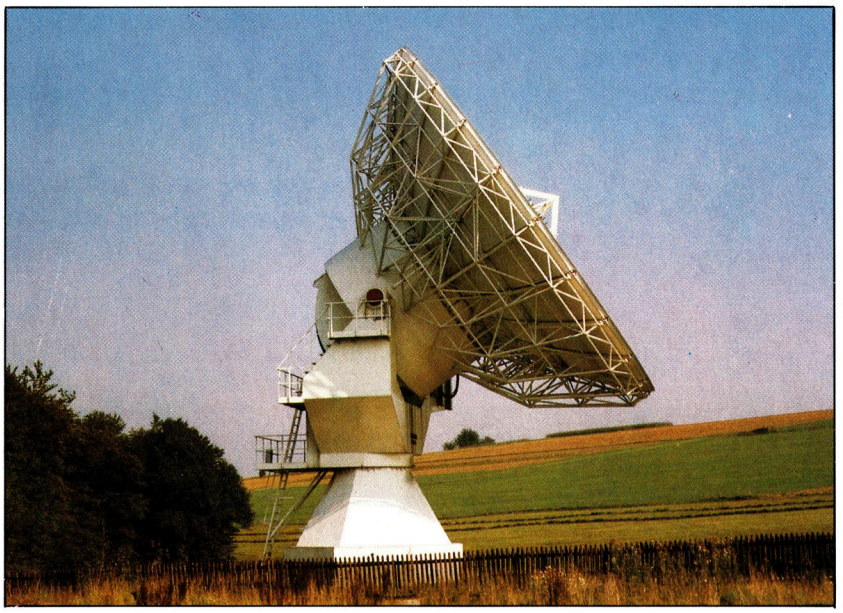
Vor dem März 1979, als der Wettersatellit Meteosat 1 in seine Umlaufbahn geschossen wurde, waren die Isobaren die einzige Grundlage einer Wetterprognose. Isobaren sind Linien, die Punkte gleichen Luftdrucks auf einer Karte verbinden. Aus ihnen kann man Schlüsse über Geschwindigkeit und Richtung von Kalt- und Warmluftfronten und der sie begleitenden Tief- und Hochdruckgebiete ziehen.

Isobarenkarten sind zwar die bekannteste Art von Wetterkarten, aber nicht die einzige. Aufgrund der riesigen Datenmengen, die Computer speichern können, sind die Wetterämter in der Lage, Karten herauszugeben, aus denen die durchschnittliche Temperatur, Regenmenge, Dauer der täglichen Sonneneinstrahlung usw. zu ersehen sind.

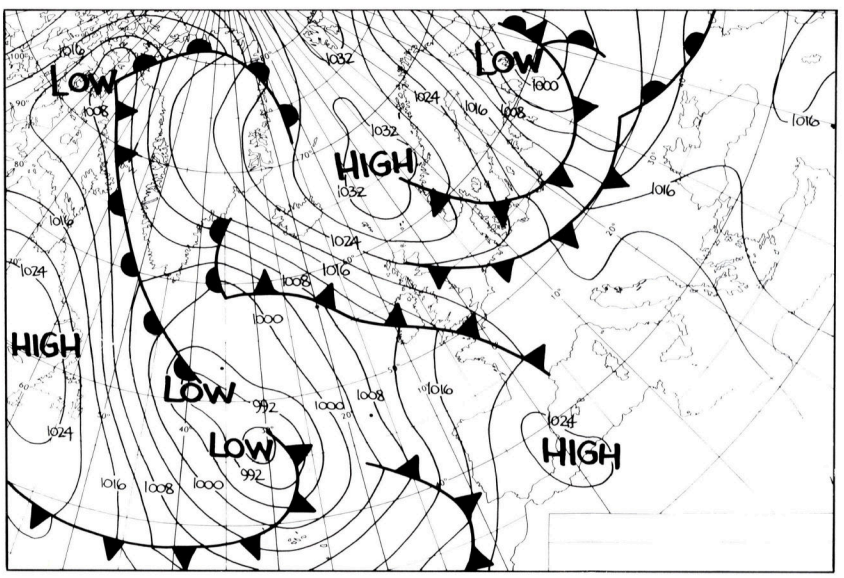
Karten über die aktuelle Wetterlage bringen die Wetterämter nach wie vor heraus. Doch jetzt ist es möglich, die Wetterentwicklung so darzustellen, wie sie der Satellit Meteosat sieht. Er liefert Analogsignale, die die Herstellung farbiger Karten und Bildschirmanzeigen möglich machen. Diese Darstellungen, die etwa alle vier Minuten neu erzeugt werden, geben ein lebendiges Bild von der Wetterlage und versetzen die Meteorologen in die Lage, für ihre Voraussage immer auf die momentane Wetterlage zurückgreifen zu können.

Im Juni 1981 wurde der erste Wettersatellit durch Meteosat 2 ersetzt. Er läuft auf einer geostationären Umlaufbahn etwa 35 880 km über dem Äquator, sammelt Daten von vielen über den Globus verstreuten Erdstationen und liefert diese Daten an jeden, der an dieses Informationssystem angeschlossen ist.

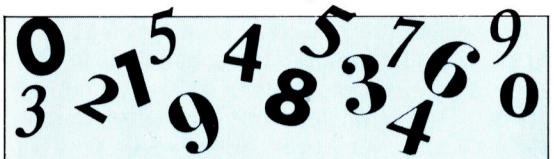
Theoretisch ist es möglich, diese Informationen sogar auf einem Heimcomputer zu analysieren und zu interpretieren, allerdings nicht in Echtzeit. Man braucht die empfangenen Daten lediglich bei Empfang aufzuzeichnen. Da die Daten aber analog sind, kann ihre Umwandlung zu erheblichen Schwierigkeiten führen. Außerdem müßte die Antennenanlage genau auf den Satelliten ausgerichtet sein. Die Verarbeitung und Auswertung von Satelliteninformationen ist nur ein kleiner Bestandteil der Arbeit der Wetterämter. Neben ähnlichen Institutionen in anderen Teilen der Welt unterhalten die Wetterämter ein globales Wettermodell und leiten von diesem eine Menge statistischer Daten ab. Daraus wird eine Datenbank historischer Informationen gebildet, von der Trendentwicklungen des globalen und lokalen Klimas abgeleitet werden. Dazu zählen nicht nur barometrische Daten, sondern auch genaue Angaben über Windgeschwindigkeit und Windrichtung, Regenmengen und Temperatur. Das Sammeln dieser Daten ist wichtig für historische Analysen. Sie sind für die Landwirtschaft ebenso lebensnotwendig wie für viele Industriezweige.



Erdstationen, die Satellitensignale empfangen, können sehr unterschiedlich sein. Die im Bild gezeigte Station kann sowohl empfangen als auch senden und ist nicht an stationäre Satelliten gebunden.



Wetterkarten sind eigentlich nichts weiter als Verzeichnisse von Luftdruckwerten. Die konzentrischen Linien verbinden Punkte gleichen barometrischen Drucks. Die Windrichtung um ein Tief ist entgegen dem Uhrzeigersinn, die Windrichtung um ein Hoch dagegen im Uhrzeigersinn. Im Süden ist es genau umgekehrt.

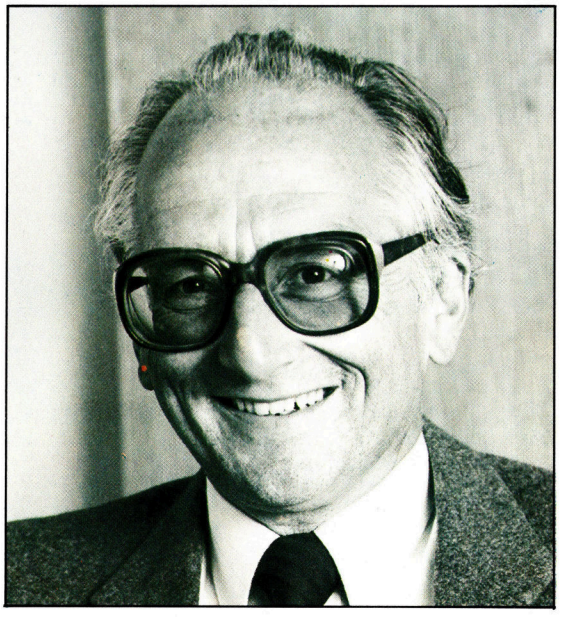


Zahlenfresser

Große Computer werden im Bereich der Wissenschaft hauptsächlich zur Verarbeitung rein numerischer Informationen in Form sehr langer und komplexer Gleichungen eingesetzt. Gleiches gilt auch für die Meteorologie. Solche Rechnungen lassen sich auch auf einem Heimcomputer ausführen. Da die Zahlen jedoch oft bis zu 30 und mehr Stellen hinter dem Komma aufweisen, ist es unmöglich, mit Kleinrechnern akzeptable Verarbeitungszeiten zu erreichen.

Chuck Peddle

Der Computerpionier entwickelte den Microprozessor 6502 und einen der ersten Personalcomputer.



Chuck Peddle machte sich daran, einen Personalcomputer zu entwickeln. Der sollte ein in sich abgeschlossenes Gerät sein, das nach dem Einstecken in die Steckdose sofort einsatzbereit ist. Das Ergebnis war der PET. Fast zur selben Zeit erschien der Apple II. PET zeichnete sich aus durch: eingebauten Monitor, integriertes Cassettengerät und den bekannten BASIC-Interpreter von Microsoft. Der PET hat viele Änderungen überstanden und ist weiterhin populär. Dazu hat besonders die Umstellung auf eine vollwertige und somit verständliche Schreibmaschinentastatur beigetragen.

Er ist eine Generation älter als die Wunderkinder unter den Computerkonstrukteuren, Stephen Wozniak und Steve Jobs. 1973 ging Chuck Peddle zu Motorola, um unter anderem bei der Entwicklung des 6800 Microprozessors mitzuarbeiten.

Der 6800, einer der ersten Microprozessoren auf dem Markt, war mit einem Stückpreis von 200 Dollar entsprechend teuer. Chuck Peddle fand, daß dieser Preis eine breite Markteinführung verhindern würde, und verließ Motorola, um bei der Firma MOS Technology neu zu beginnen.

Was er in diesem relativ kleinen Unternehmen entwickelte, sollte bald der erfolgreichste Microprozessor des ersten Microcomputer-Jahrzehnts werden – die 6502 MPU. Niemand konnte erahnen, daß dies der Grundstein einer ganzen Industrie war, die eine soziale Umwälzung von enormen Ausmaßen in Gang setzte.

Einer der schon sehr früh die weitreichenden Auswirkungen von Microprozessoren und speziell des 6502 von MOS Technology erkannte, war Jack Tramiel, Ex-Präsident von Commodore. Bis dahin waren Commodores Umsätze mit einer Reihe von Büroprodukten und den verschiedensten Taschenrechnern recht bescheiden gewesen.

Es ist nicht schwer zu verstehen, daß Tramiel, Hauptkunde von 4-Funktionen-Chips für Taschenrechner von MOS Technology, diese Firma eines Tages aufkaufte, obwohl Commodore selbst alle Mühe hatte, den Kopf über Wasser zu behalten. Wichtigster Bestandteil des Kaufvertrages aber war für Tramiel, neben dem 6502, der Entwicklungsingenieur Chuck Peddle.

Chuck Peddle hatte sicher auch mit seinem Konzept eines Personalcomputers Eindruck gemacht. Ähnliche Ideen verfolgten, unabhängig von Peddle, die Apple-Gründer Wozniak und Jobs. Chuck Peddle war so sehr um seine Idee besorgt, daß er sich mit Bill Gates, dem Gründer von Microsoft und Vater des berühmten BASIC-Interpreters, zusammentat, um Apple aufzukaufen. Dies geschah fast zur gleichen Zeit, als Commodore MOS Technology erwarb. Wozniak und Jobs forderten damals 150 000 Dollar für Apple, aber Peddle und Gates konnten nur zwei Drittel davon aufbringen.

So blieb Chuck Peddle bei Commodore und übernahm 1977 die Entwicklung des PET (Personal Electronic Transactor). Zur selben Zeit bauten Wozniak und Jobs gerade den Apple II. Der PET unterschied sich davon durch eingebauten Monitor, integriertes Cassettengerät sowie eine Tastatur, deren Qualität allerdings mehr mit einem Taschenrechner als einer Schreibmaschine vergleichbar war. Ungeachtet dieses Nachteils hatte Commodore im Handumdrehen 1000 PETs zum Preis von etwa 1500 Dollar das Stück verkauft. Damit war die erste Generation von Microcomputern geboren, die speziell für den privaten Hausgebrauch entwickelt wurde.

Drei Jahre später entwickelte Chuck Peddle ein neues Talent – das des Firmenchefs. Zusammen mit Chris Fish, einem ehemaligen Finanzfachmann von Commodore, gründete er Sirius Systems Technology als Tochterunternehmen der Walter Kidde Corporation.

Die Entwicklung im Bereich der Personalcomputer konzentrierte sich damals auf die 16-Bit-Chips wie den 8088 von Intel. Auch bei IBM bastelte man fieberhaft am PC. Sirius brachte den Sirius I jedoch einige Wochen früher auf den Markt. Der Sirius I war der erste preiswerte Microcomputer der 16-Bit-Generation, der in großer Auflage produziert wurde und weite Beachtung fand.

Die Bedienung war durch eine frei bewegliche Tastatur und den hochauflösbaren Grafikmonitor mit blendfreiem Bildschirm angenehm einfach. Der Sirius I setzte einen bis dahin nicht gekannten Standard für Microcomputer-Bürosysteme.

Relativ oder absolut?

Der Unterschied zwischen Turtle- und Koordinaten-Geometrie. Grafikbeispiele Spirale, Würfel und Vieleck mit dem Commodore-LOGO.

In unserem LOGO-Kurs haben wir bislang hauptsächlich die Möglichkeiten der Atari-Version vorgestellt. Die nachfolgenden Beispiele beziehen sich auf das Commodore-LOGO.

Wie bereits erwähnt, unterscheidet man bei LOGO zwischen Befehlen im direkten Mode und solchen, die in Prozeduren zusammengefaßt und mit einem Namen versehen sind. Im Gegensatz zu den LOGO-Befehlen, die immer in Großbuchstaben geschrieben werden müssen, kann der Name einer Prozedur Groß- und Kleinbuchstaben enthalten. Die Eingabe für die erste Prozedur mit dem Namen ‚WUERFEL' sieht so aus:

```
EDIT WUERFEL
```

Nach Betätigen der RETURN-Taste wird der Bildschirm gelöscht, und es erscheint die Meldung ‚TO WUERFEL' sowie am unteren Rand EDIT CTRL-C TO DEFINE, CTRL–G TO ABORT. Das Wort EDIT zeigt hier an, daß Sie sich nicht mehr im direkten Mode befinden, sondern nun die Befehle für die Prozedur WUERFEL eingeben sollen, die erst später, nach Aufruf dieser Prozedur, ausgeführt werden. Haben Sie Ihre Anweisungen eingetippt, drücken Sie CTRL-C, um die Prozedur zu definieren bzw. abzulegen. Durch diesen Befehl wird die Prozedur WUERFEL in ein internes „Wörterbuch" aufgenommen. Um die Prozedur zu vervollständigen, geben Sie die folgenden Anweisungen ein:

```
TO WUERFEL
  REPEAT 4 [FD 50 RT 90]
END
```

Nachdem CTRL-C gedrückt wurde, gibt LOGO als Antwort WUERFEL DEFINED aus. Wollen Sie jetzt eventuelle Fehler korrigieren, geben Sie CTRL–G ein, wobei alle Editierfunktionen oder die Cursorsteuerungstasten wie gewohnt benutzt werden können. Außer diesen gibt es bei LOGO noch zahlreiche andere Möglichkeiten, lange Prozeduren zu korrigieren. Auf diese wird im weiteren Verlauf des Kurses noch ausführlich eingegangen.

Um zu sehen, wie das Ergebnis des Programms auf dem Bildschirm aussieht, geben Sie DRAW ein (dadurch erscheint der Grafikschirm) und anschließend WUERFEL. Die Turtle führt nun die Befehle der Prozedur aus. Und zwar exakt in der Reihenfolge, in der die Anweisungen eingegeben wurden.

Der normale LOGO-Befehlssatz, bestehend aus sogenannten ‚Primitives', steht mit dem Einladen von LOGO zur Verfügung. Sobald nun aber eine Prozedur definiert wurde, kann deren Name, genauso wie die festgelegten Befehle, in Prozeduren eingebaut werden.

Das bedeutet in diesem Fall, daß sich der Befehlssatz beliebig nach den jeweiligen Erfordernissen erweitern läßt. Bei dem aufgezeigten Beispiel entsteht ein Rechteck mit einer Seitenlänge von 50 Einheiten. Um die Prozedur zu ändern, etwa um die Länge von 30 Einheiten festzulegen, wird

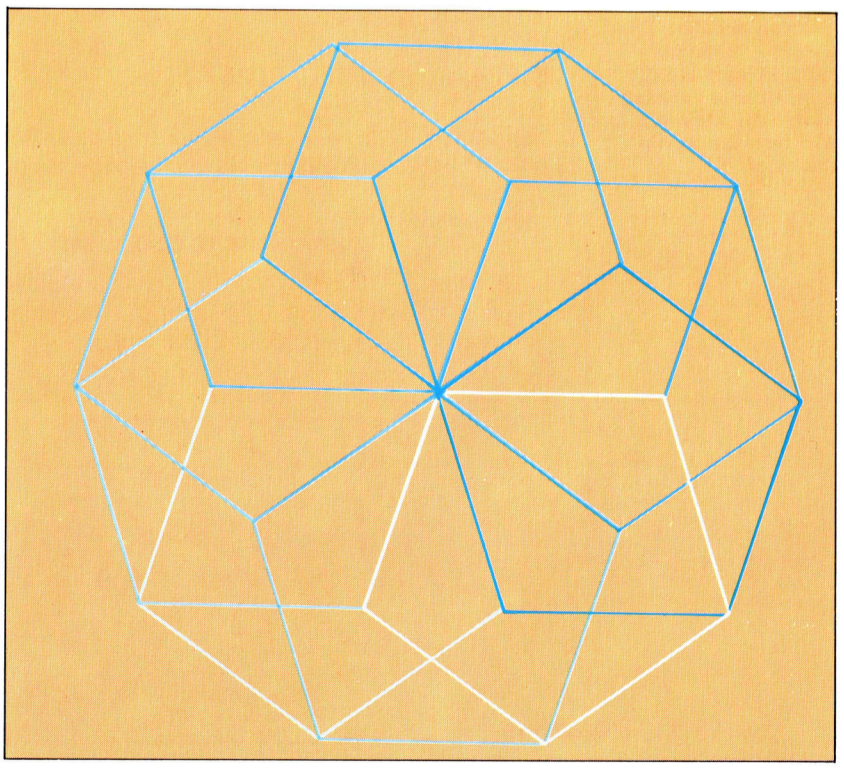
```
EDIT WUERFEL
```

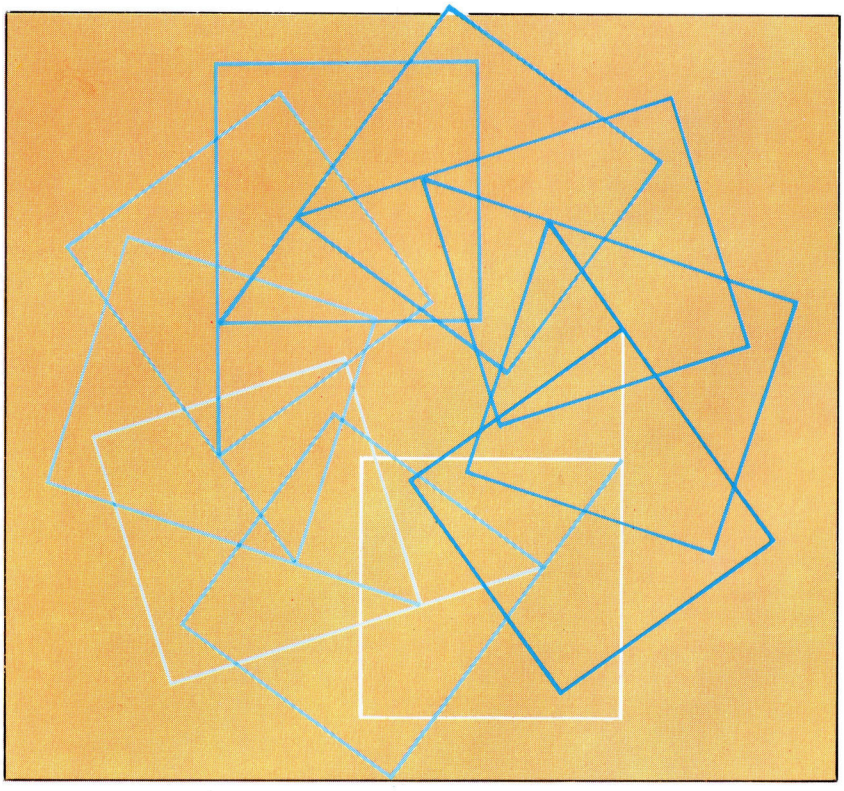
eingetippt. Sie befinden sich jetzt wieder im Editier-Mode und auf dem Bildschirm erscheint der komplette Inhalt der WUERFEL-Prozedur. Mit Hilfe der Steuerungstasten kann man nun beispielsweise 50 in 30 ändern. Um diese Prozedur mit den neuen Werten zu definieren, geben Sie wieder CTRL-C ein. Versuchen Sie nun, mit den erlernten Befehlen andere geometrische Figuren wie Dreiecke, Sterne usw. zu erstellen. Berücksichtigen Sie

Abkürzungen

EDIT	ED
HIDETURTLE	HT
SHOWTURTLE	ST

VIELECK





MUSTER

dabei auch die REPEAT-Anweisung. Dazu wieder ein Beispiel:

REPEAT 5 [FD 50 RT 72]

und danach:

REPEAT 10 [PENT RT 36]

Man kann mit diesen wenigen Befehlen die unterschiedlichsten Figuren bauen. Versuchen Sie nun folgendes:

REPEAT 10 [WUERFEL RT 36 FD 25]

Sieht das Ergebnis so aus, wie Sie es erwartet haben? Sicherlich sind bei Ihren Eingabeübungen manchmal etwas seltsame Figuren entstanden. Um ein Dreieck zu zeichnen, haben Sie wahrscheinlich diese Zeile eingetippt

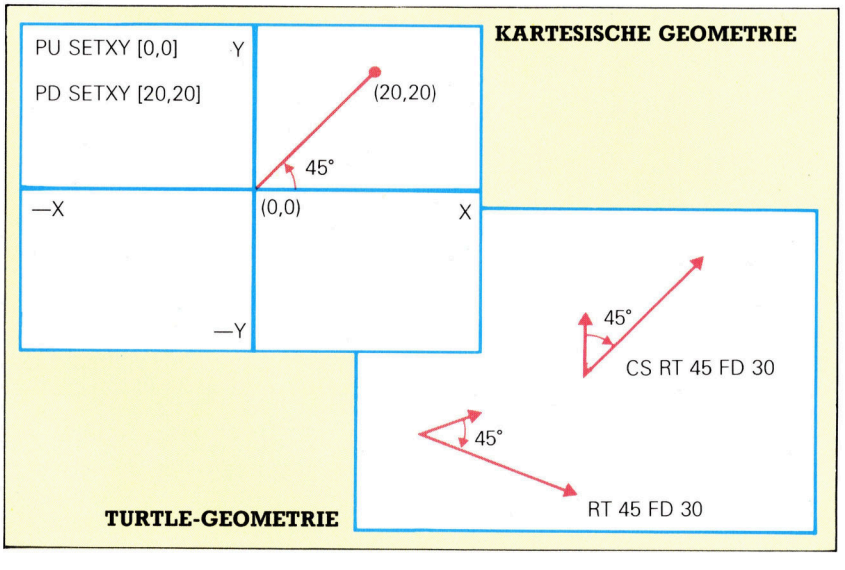
REPEAT 3 [FD 50 RT 60]

Was dabei auf dem Schirm erscheint, ist jedoch nur ein halbes Sechseck. Es ist eben doch nicht so einfach, wie es scheint. Deshalb sollten Sie bei der Eingabe der Anweisungen selber Turtle „spielen" und sich vorstellen, was die einzelnen Befehle bewirken. Überlegen Sie nun, wie eine in sich geschlossene Figur zu erstellen ist. Die Turtle muß einmal auf dem Bildschirm herumwandern, wobei die Ausgangs- und Endposition übereinstimmen müssen. Das heißt, es sind innerhalb einer Figur Drehungen von insgesamt 360 Grad auszuführen. Nehmen wir als Beispiel wieder das Dreieck. Bei diesem muß sich die Turtle dreimal drehen. Die Berechnung sieht so aus: 360/3=120 Grad. Jetzt ist es leicht, das Ergebnis in eine korrekte Programmeingabe zu fassen.

REPEAT 3 [FD 50 RT 120]

Dieses Berechnungsprinzip bleibt bei allen „geschlossenen" Figuren gleich – der einzige Unterschied besteht in der Anzahl und Größe der Winkel, in der die Turtle zu drehen ist. Komplizierter wird die Berechnung, wenn man beispielsweise einen sechszackigen Stern zeichnen möchte, da die Winkel nicht alle gleich sind.

Die Bewegungen der Turtle werden durch „relative" Werte festgelegt. Im Gegensatz dazu steht die auf Koordinaten beruhende Geometrie, die mit absoluten Werten arbeitet. Die Turtle-Steuerung bezieht sich auf die momentane Position sowie die Stellung, in der sich die Turtle befindet.



Die Turtle im Koordinatensystem

Die Bewegungen der Turtle gehen, anders als bei der auf Koordinaten bezogenen Geometrie, von der momentanen Position innerhalb der Figur aus. Die Werte der Turtle-Geometrie sind also relativ. Die Koordinaten-Geometrie dagegen arbeitet nur mit absoluten Werten, wobei man sich den Bildschirm als ein Raster vorstellen kann, dessen horizontale und vertikale Punkte, ausgehend vom Schirmmittelpunkt, mit festgelegten Zahlen gekennzeichnet und ansteuerbar sind. Neben den bereits erwähnten Möglichkeiten läßt sich die Turtle auch durch Koordinatenangaben steuern.

Der Befehl SETXY 20 30 zum Beispiel führt die Turtle von der gegenwärtigen Position auf die Koordinate 20, 30. Wurde zuvor die Anweisung PENDOWN gegeben, zeichnet die Turtle eine Linie von der vorhergehenden Position zur jetzigen; PENUP bewirkt logischerweise, daß diese Linie nicht zu sehen ist. Die Ausgangsposition (0,0) liegt auf dem Schirmmittelpunkt.

Die folgenden Prozeduren demonstrieren den Unterschied zwischen der Turtle- und der Koordinaten-Geometrie:

TO GEOMETRIE1

```
   REPEAT 4 [FD 50 RT]
END

TO GEOMETRIE2
   SETXY 0 50
   SETXY 50 50
   SETXY 50 0
   SETXY 0 0
END
```

Beide Programme erzeugen das gleiche Resultat auf dem Bildschirm. Was geschieht jedoch, wenn Sie die zwei Quadrate etwa um 30 Grad drehen wollen? Mit RT 30 GEOMETRIE1 läßt sich die Drehung problemlos bewerkstelligen; die zweite Prozedur dagegen muß für diesen Zweck komplett neu geschrieben und berechnet werden, da ja hier absolute Bildschirm-Koordinaten eingegeben wurden. Es gibt aber auch Fälle, bei denen es ratsam ist, auf den SETXY-Befehl zurückzugreifen, da dieser schneller ausgeführt wird als die FORWARD-Anweisung.

Ein anderer Faktor, der bei LOGO zu berücksichtigen ist, liegt in der „Kurzsichtigkeit" der Turtle. Sie kann immer nur einen Schritt nach dem anderen ausführen – parallele Bewegungsabläufe sind nicht möglich. Soll etwa ein Kreis gezeichnet werden, so bewegt sich die Turtle ein Stück vorwärts, dreht sich, geht wieder vorwärts etc., und zwar so lange, bis der Kreis geschlossen ist. In LOGO übersetzt sieht das so aus:

```
TO KREIS
   REPEAT 360 [FD 1 RT 1]
END
```

Unser Beispiel zeigt ein 360seitiges Vieleck; doch die Linien, die die einzelnen Punkte verbinden, sind so kurz, daß die Zeichnung als eine perfekte Kurvenform erscheint. Um die lange „Zeichnungszeit" zu verkürzen, kann man statt dessen ebenso eine 36seitige Figur zeichnen, da der Unterschied in der Darstellung unerheblich ist.

```
TO KREIS
   REPEAT 36 [FD 10 RT 10]
END
```

LOGO-Dialekte

Die verschiedenen LOGO-Versionen unterscheiden sich nicht maßgeblich voneinander. Einige Befehle variieren jedoch aufgrund Hardware-spezifischer Vorgaben. Der vollständige Befehlssatz für Ihren Computer ist im LOGO-Handbuch aufgeführt. Bei einigen LOGOs muß der EDIT-Befehl mit Anführungsstrichen geschrieben werden: EDIT "WUERFEL.
Die andere Abweichung ist bei der SETXY-Anweisung zu beachten. Sie wird von einigen Computern nur in Form von SETPOS verarbeitet.

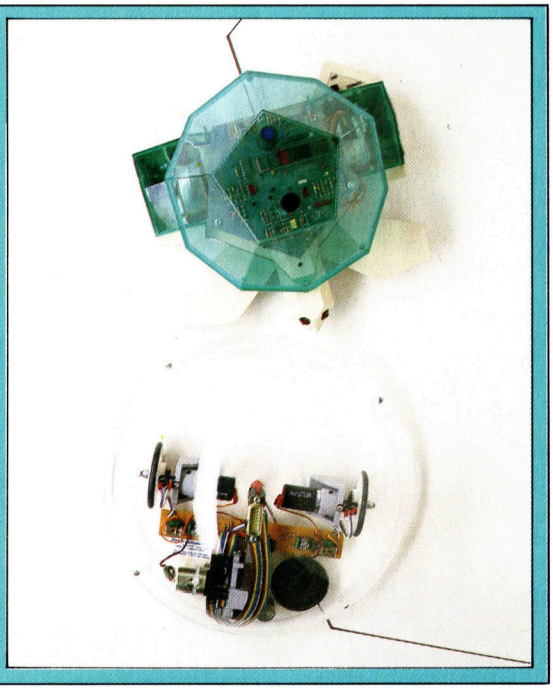
Mit der über den Computer steuerbaren Boden-Turtle werden den Kindern eventuelle Ängste vor dem Computer genommen. Die Aufgabe dieses Spiels besteht darin, einen Gegenstand mit der Turtle, die durch Befehlseingaben bewegt wird, anzusteuern.

Verändern Sie diese Zeilen so, daß daraus ein Halbkreis oder andere geometrische Formen entstehen.

Mit dem nächsten Programm wird eine Spirale auf den Schirm gezeichnet. Diese Prozedur ist sicherlich relativ umständlich programmiert, verwendet jedoch nur Befehle, die bereits erklärt wurden.

```
TO SPIRALE
   FD 10 RT 10 FD 10 RT 20 FD 10 RT 30
   FD 10 RT 40 FD 10 RT 50 FD 10 RT 60
   FD 10 RT 70 FD 10 RT 80 FD 10 RT 90
END
```

Diese mechanischen Zeichenroboter werden direkt von LOGO gesteuert. Die obere „Schildkröte" soll Kindern die mögliche Angst vor dem Computer nehmen, das untere Gerät ist ein handelsüblicher Zeichenroboter, der vom LOGO-Programm gesteuert wird.



Richtig adressiert

Die Zentraleinheit CPU ordnet Transport und Manipulation von Daten und Befehlen, die in Tausenden von Speicherplätzen des Computers abgelegt sind.

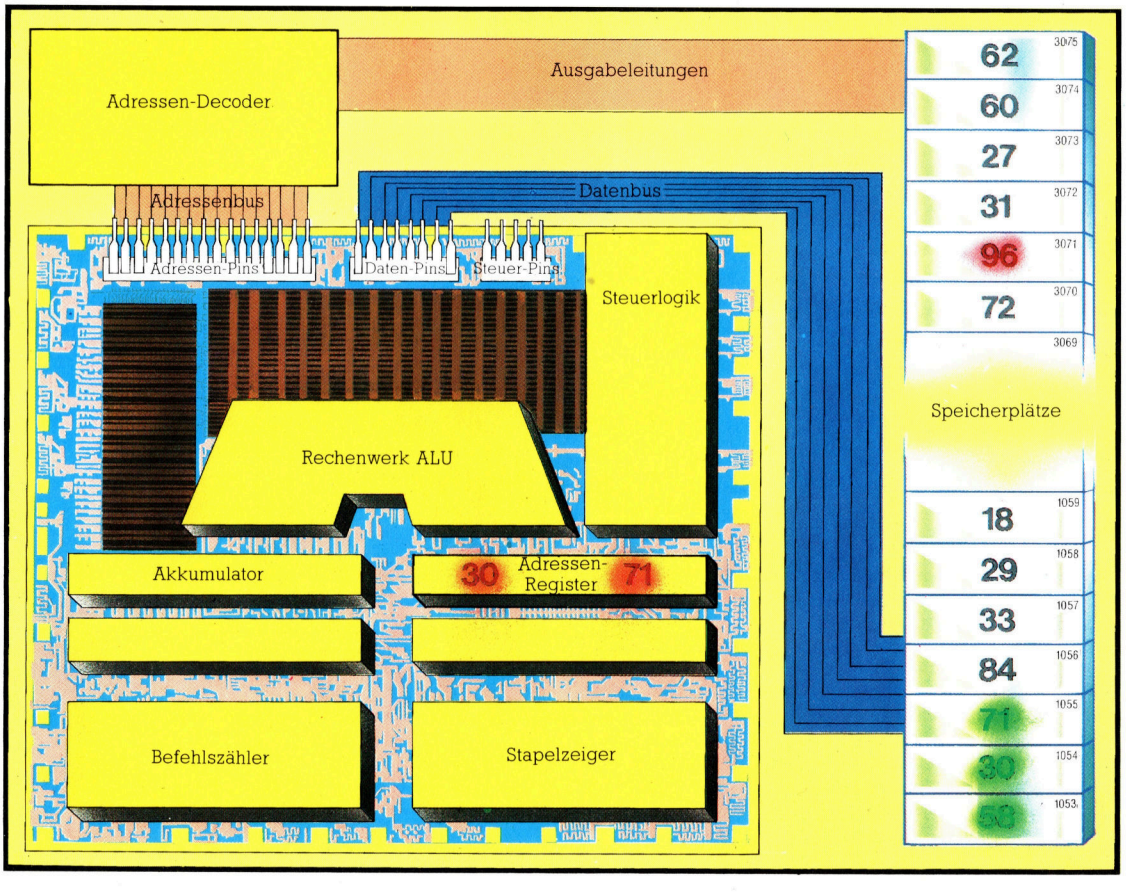
Die CPU verschafft sich durch Adressierung des gewünschten Speicherplatzes und Laden seines Inhaltes über den Datenbus Zugang zu gespeicherten Befehlen oder Daten. Dieser Vorgang ist reichlich kompliziert, weil sich hinter den gespeicherten Bytes (8 Bit umfassende binäre Codes) sowohl Daten, die vom Prozessor nach unterschiedlichen Vorgaben manipuliert werden müssen, als auch Befehle verbergen können.

Die vielen Tausend Speicherplätze des Computerspeichers enthalten schließlich beides: Befehle, die der CPU sagen, was zu tun ist, und Daten, an denen sie diese Manipulationen auszuführen hat. Wie aber erkennt die Zentraleinheit den Unterschied zwischen Befehlen und Daten?

Befehle sind codierte binäre Ausdrücke, die die CPU anweisen, bestimmte Operationen in vorgegebener Reihenfolge auszuführen. Erkennt die CPU im binären Ausdruck 00111010 z. B. einen Befehl und nicht nur ein Datenbyte, so könnte sie die nächsten beiden Speicherplätze adressieren, deren Inhalt in ein besonderes Adressen-Register laden, die Adressenleitungen dem Inhalt dieses Registers entsprechend setzen und dann den Inhalt der neu adressierten Speicherplätze über den Datenbus in den Akkumulator laden.

Speicherplatz-Adresse

Dies hört sich zwar etwas verwirrend an, doch wir haben gerade eine Adressierung von Speicherplätzen beschrieben, wie sie in der weit verbreiteten CPU Z80 vorkommt. Die Illustration verdeutlicht dies (alle verwendeten Zahlen sind Dezimalzahlen). Nehmen wir also an, die CPU weiß, daß das nächste Byte mit Adresse 1053 kein Datenbyte sondern ein Befehl ist. Dann wird sie über den Adressenbus 1053 oder binär 0000010000011101 senden, und ihre Adressenleitungen dem binären Ausdruck entsprechend auf Null oder Eins setzen.



Selbst die einfachste CPU besteht aus vielen Funktionsblöcken. Befehle, im Englischen kurz ‚opcodes' genannt, müssen vom Speicher in die CPU geladen werden. Nach der Decodierung durch die Steuerlogik werden die Befehle ausgeführt. Im illustrierten Beispiel löst der vom Speicherplatz 1053 geladene Befehl 58 eine Kette von Aktionen aus: Der Inhalt von Speicherplatz 1054 wird in eine Hälfte des 16-Bit-Adressen-Registers geladen, der Inhalt von 1055 in die zweite Hälfte. Nachdem beide Bytes dieses Registers auf den Adressenbus gegeben und decodiert sind, wird 96, der Inhalt des so adressierten Speicherplatzes 3071, von der CPU über den Datenbus in den Akkumulator geladen. Dann wird der Adressenbus auf 1056 gesetzt. Das ist die Adresse vor dem Sprung nach 3071, aber um Eins erhöht. Durch die bei der Konstruktion des Chips festgelegten Verkettungen von Daten und Befehlen erkennt die CPU den Inhalt als Befehl. In unserem Beispiel bewirkt 84 deshalb eine Invertierung des Akkumulator-Inhaltes.

Der Adressen-Decoder ändert daraufhin nur eine Ausgabe-Leitung, die Speicherplatz 1053 adressiert.

Als nächstes wird der adressierte Inhalt, 58 oder 00111010 binär, über den Datenbus in die CPU geladen. Da die CPU einen Befehl erwartet, bestimmt die Steuerlogik, welche Operationen ablaufen sollen. In unserem Beispiel hat der Befehl folgenden Inhalt: Benutze die nächsten 16 Bit als Adresse und lade den Inhalt des adressierten Speicherplatzes in den Akkumulator. Sobald die CPU diesen Befehl über die Steuerlogik erkennt, weiß sie, daß die nächsten beiden Bytes eine Adresse sein müssen, deren Inhalt sie in den Akkumulator laden muß. Daraus folgt, daß die CPU erst nach Ausführung dieser Operationen den nächsten Befehl erwarten kann und der nächste Befehl vom Speicherplatz 1056 kommen muß.

Die CPU ändert in unserem Beispiel also zuerst den Adressenbus von 1053 auf 1054 und lädt den Inhalt dieses Speicherplatzes über den Datenbus in eine Hälfte des Adressen-Registers. Dann ändert sie den Adressenbus auf 1055 und lädt das so adressierte Byte in die zweite Hälfte des Adressen-Registers. Der Transport der Daten zur CPU geschieht also immer über den Datenbus, einer Art Einbahnstraße, die von den Speicherplätzen zur CPU führt.

Die CPU überträgt nun den Inhalt des Adressen-Registers auf den Adressenbus: Die neue Adresse lautet dann 3071 oder binär 0000101111111111. Nach Aktivierung des Speicherplatzes 3071 durch den Adressen-Decoder lädt die CPU 96 oder binär 01100000 über den Datenbus in den Akkumulator und gibt 1056, die um Eins erhöhte Adresse vor dem Sprung nach 3071, auf den Adressenbus. Die CPU erwartet unter dieser Adresse einen Befehl. Erinnern Sie sich bitte, daß der bis jetzt geschilderte Vorgang durch **einen** Befehl ausgelöst wurde und automatisch abläuft.

CPU-Befehlsvorrat

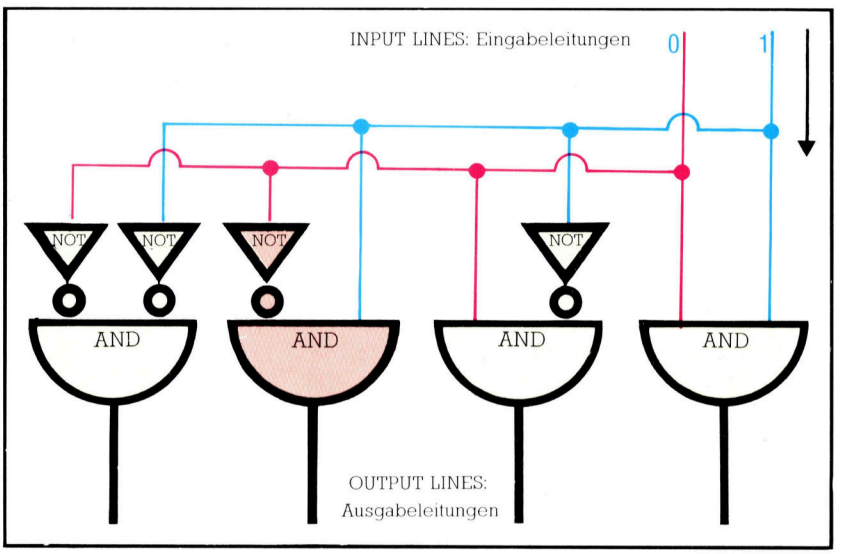
Welcher Art könnte der unter Adresse 1056 gespeicherte Befehl sein? Die Antwort ist nicht einfach, weil der Befehlsvorrat der CPU, abhängig vom Typ, mehrere Dutzend oder einige Hundert umfassen kann. Nehmen wir an, daß die Daten im Akkumulator invertiert werden sollen (Nullen werden in Einsen, Einsen in Nullen verwandelt). Nehmen wir weiter an, der Inhalt des Speicherplatzes 1056 wäre ein Befehl, der diese Operation ausführt, und hieße 84. Die CPU wird dann, sobald sie 84 über den Datenbus geladen hat, den Akkumulatorinhalt 96 oder binär 01100000 in 159 oder binär 10011111 verwandeln. Nach diesem Befehl zur Invertierung erwartet die CPU einen weiteren Befehl, denn im Akkumulator befindet sich ja noch das invertierte Datenbyte.

Die beschriebene Art, Speicherplätze zu adressieren, ist nur eine von vielen Möglichkeiten, die der Programmierer zur Verfügung hat. Auch sind die verwendeten Befehle, 58 für das Laden des Akkumulators und 84 für das Invertieren seines Inhalts, nur für die modellhafte CPU unseres Beispiels gedacht. Das vorgeführte Prinzip gilt jedoch generell, auch wenn die verschiedenen Hersteller vom Microprozessoren gleiche Befehle unterschiedlich codieren und die Befehlsätze nicht immer gleichen Umfang haben.

Für die Eingabe/Ausgabe (E/A) von Daten müssen ebenfalls feste Speicherplätze reserviert werden, die von der CPU genau wie Daten oder Befehle adressiert werden. Gewöhnlich hat ein 8-Bit-Microprozessor nur acht Leitungen (Pins) für die Adressierung von E/A-Speicherplätzen. Dies bedeutet, daß maximal nur 256 Eingabe/Ausgabe-Adressen belegt werden können. Für die meisten der möglichen Anwendungen von Microcomputern ist dies jedoch mehr als genug.

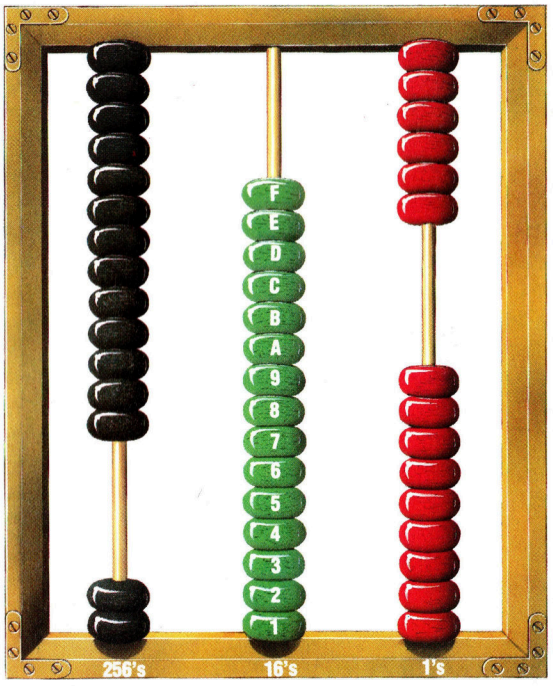
Bei der Adressierung ist besonders darauf zu achten, daß nur die Adresse angesprochen wird (entweder eine Speicherplatz- oder eine Ein/Ausgabeadresse), die beim jeweiligen Programmablauf benötigt wird. Die anderen Adressen dürfen dabei nicht aktiviert werden. Jede Adressierung erfordert auch die Fähigkeit, Adressen zu decodieren, damit nur der eine gewünschte Speicherplatz freigegeben wird. Diese Art der Freigabe ist mit einfachen Logikbausteinen zu bewältigen, wenn nur wenige Adressenleitungen zu decodieren sind. Das Prinzip eines 2-zu-4-Decoders zeigt die Illustration. Häufig geschieht die Auswahl von E/A-Geräten auf diese einfache Weise. Mit steigender Anzahl von Adressenleitungen steigt jedoch der Aufwand für die Decodierung überproportional. Muß beispielsweise von 65 536 verschiedenen Speicherplätzen jeder für sich eindeutig adressierbar sein, so geschieht dies meistens direkt im Speicherbaustein selbst.

Der 16 Leitungen umfassende Adressenbus kann jeden einzelnen von 65 535 möglichen Speicherplätzen eindeutig adressieren. Adressen-Decoder setzen dazu die Nullen und Einsen auf dem Adressenbus in ein Signal um, das nur einen Speicherplatz freigibt. Der größte Teil der Decodierung wird durch logische Schaltungen vorgenommen, die sich auf dem Speicherchip befinden. Die Illustration zeigt, wie mit zwei Eingabeleitungen immer nur eine von vier möglichen Ausgabeleitungen ausgewählt werden kann.



Rechnen mit Basis 16

Programmierer benutzen hexadezimale Zahlen, obwohl Computer nur binäre Zahlen, also nur zwei logische Zustände, erkennen und verarbeiten können; denn Hexzahlen lassen sich sehr einfach in Binärzahlen umrechnen.



Ein Abakus für hexadezimale Zahlen müßte auf jeder Stange 15 Perlen haben. Das hexadezimale Zahlensystem hat die Basis 16 und fünfzehn Ziffern: von 0 bis 9 und A bis F. Die mittlere Stange des Abakus läßt erkennen, daß A für dezimal 10 steht, B für 11, C für 12, D für 13, E für 14 und F für 15. Hat die Zählung der Einer-Perlen F erreicht, wird die erste Sechszehner-Perle gezählt und die Zählung der Einer-Perlen beginnt von vorn, bis alle Sechszehner-Perlen gezählt sind. Der abgebildete Abakus stellt die Dezimalzahl 761 dar. Sie wird gebildet aus 2×256 + 15×16 + 9×1. Ihre hexadezimale Form ist 2F9.

Es ist offensichtlich, warum für Computer das binäre Zahlensystem so vorteilhaft ist: Ein/Aus-Zustände elektrischer Signale lassen sich als Einsen und Nullen darstellen oder codieren. Wir haben uns dagegen an das Dezimalsystem mit der Basis Zehn gewöhnt, und schon Kleinkinder können leichte Rechenaufgaben lösen, weil sie ihre zehn Finger zu Hilfe nehmen können.

Warum nun aber gerade hexadezimale Zahlen mit der Basis 16? Nun, sie lassen sich viel leichter als Dezimalzahlen in binäre Zahlen verwandeln, und umgekehrt gilt das gleiche. Trotzdem stellt sich die Frage, warum man Programme nicht gleich in binärer Form schreibt, wenn Computer binäre Zahlen so gut verstehen. Zwei Gründe sind dafür ausschlaggebend: Erstens sind binäre Ausdrücke viel umfangreicher als ihre dezimalen oder hexadezimalen Äquivalente, und zweitens macht man, wegen der vielen Einsen und Nullen, sehr häufig Fehler. Nehmen wir als Beispiel 356 dezimal. Ihre vielstellige binäre Form, 101100100, ist viel weniger anschaulich.

Das hexadezimale System arbeitet mit 16 verschiedenen Ziffern einschließlich der Null. Die größte einstellige hexadezimale Zahl entspricht dezimal 15. Da mit diesem System nur zehn verschiedene Ziffern darstellbar sind, nimmt man für die hexadezimalen Zahlen, die dezimal 10 bis 15 entsprechen, die ersten sechs Buchstaben des Alphabets zu Hilfe. Wir zählen hexadezimal also wie gewöhnlich von 0 bis 9 und dann A, B, C, D, E, F. F entspricht also dezimal 15. Eine mehrstellige hexadezimale Zahl kann demnach aus Ziffern und Buchstaben bestehen. Die Tabelle zeigt Dezimalzahlen bis 21 und ihre binäre und hexadezimale Schreibweise. Hex 15 besteht aus 5 Einern und einem Sechszehner, was zusammen in dezimaler Schreibweise 21 ergibt.

FFFF ist die hexadezimale Schreibweise für 65 535. Der Vorteil liegt auf der Hand: Die Dezimalzahl hat fünf Stellen, die entsprechende Binärzahl 16 Stellen, und die gleichwertige „Hexzahl" hat nur vier Stellen. Ein zusätzlicher Vorteil ist, daß eine hexadezimale Stelle oder Ziffer genau vier binäre Stellen wiedergeben kann. Dies vereinfacht, wie wir gleich sehen werden, die Umwandlung von binär in hexadezimal und zurück.

Wir beginnen von rechts und teilen die binäre Zahl in Gruppen zu je vier Ziffern. Für jede binäre Vierergruppe ist einfach die entsprechende hexadezimale Ziffer einzutragen. Hier sind einige Beispiele:

		1110	1001	binär
		=E	9	hex
		=233		dezimal
	1111	1000	1100	binär
	=F	8	C	hex
	=3980			dezimal
0111	1110	0010	1101	binär
=7	E	2	D	hex
=32301				dezimal

Wofür benötigt man nun eigentlich hexadezimale Zahlen? Denn um Programme in BASIC zu schreiben, sind weder hexadezimale noch binäre Zahlen nötig. Richtig, aber wer Programme in Maschinencode oder ASSEMBLER schreiben will, tut dies sehr bequem mit hexadezimalen Zahlen.

Zur Unterscheidung von Dezimalzahlen schreibt man hinter hexadezimalen Zahlen ein H. Dezimal 256 erscheint dann als 100H und wird gelesen: Eins, Null, Null, Hex.

Binär	Dezimal	Hex
00000001	1	1
00000010	2	2
00000011	3	3
00000100	4	4
00000101	5	5
00000110	6	6
00000111	7	7
00001000	8	8
00001001	9	9
00001010	10	A
00001011	11	B
00001100	12	C
00001101	13	D
00001110	14	E
00001111	15	F
00010000	16	10
00010001	17	11
00010010	18	12
00010011	19	13
00010100	20	14
00010101	21	15

Fachwörter auf einen Blick

Abakus
Ursprünglich war dies ein mit Sand bestreutes Brett, auf das Zahlzeichen geschrieben wurden, um auf diese Art und Weise zu rechnen; später wurde daraus ein Rechenbrett, auf dem mit Hilfe frei beweglicher Steine Additionen und Subtraktionen durchgeführt wurden; als Weiterentwicklung entstand der heute noch bekannte Holzrahmen mit Drahtstäben, auf die Perlen aufgefädelt wurden; durch Verschieben der Perlen wird das Zusammenzählen und Abziehen verdeutlicht

ADSR
Kurzform für die schnell aufeinanderfolgenden Veränderungen eines Tones beim „Anschlag": Attack, Decay, Sustain und Release

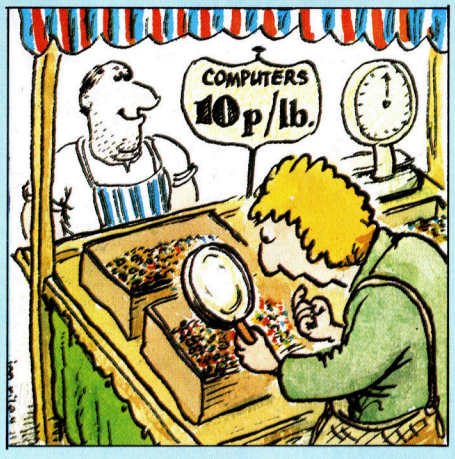
Akkumulator
akkumulatives Register; dieser Speicher des Microprozessors ist bei fast allen Ein/Ausgaben beteiligt

Coprozessor
Um die Leistungsfähigkeit zu verbessern, haben einige Rechner einen zweiten Prozessor, der der CPU Aufgaben abnimmt

Editor
Hilfsprogramm, das Eingabe, Korrektur, Speicherung und Ausgabe von Programmen unterstützt

EEROM
Electrically Erasable ROM; Festwertspeicher, der durch das Anlegen eines Löschimpulses gelöscht und dann neu „beschrieben" werden kann

Eindimensionales Feld
Daten mit gemeinsamen Eigenschaften werden entweder nur untereinander (Spalte) oder nur nebeneinander (Zeile) geschrieben



ELAN
Um 1975 entwickelte höhere Programmiersprache, die auf Erfahrungen mit PASCAL und ALGOL basiert

Fehlersuchprogramm
Dienstprogramm, das die Fehlersuche unterstützt; es stellt eine Reihe von Funktionen zur Verfügung, wie beispielsweise das Setzen von Programmunterbrechungs-Punkten sowie Kontrolle und Änderung der verwendeten Variablen und Register während des Programmablaufs

FIFO-Speicher
First-In-/First-Out-Speicher, dessen Daten nicht durch die Angabe einer Adresse, sondern durch die Reihenfolge der Lesevorgänge spezifiziert werden; jeder Lesebefehl holt das Datum aus dem Speicher, das sich am längsten darin befindet

Hexzahl
Die hexadezimalen Zahlen basieren auf den Potenzen der Zahl 16; die verschiedenen Stellenwerte werden durch die Ziffern 0 bis 9 und die Buchstaben A bis F dargestellt

Hohe Auflösung
Die Anzahl der Bildpunkte einer Flächeneinheit am Bildschirm entscheidet maßgeblich über die Schärfe und Deutlichkeit der Grafiken; um so mehr Punkte darstellbar sind, desto besser das Bild

LIFO-Speicher
Last-In-/First-Out-Speicher; die zuletzt gespeicherten Daten werden zuerst ausgegeben

UHF-Modulator
Eine große Anzahl von Heimcomputern kann an gewöhnliche Fernsehgeräte angeschlossen werden; dies ist jedoch nur durch sogenannte UHF-Modulatoren möglich; diese bereiten die elektronischen Signale auf, so daß sie in das TV-Gerät über die Antennenbuchse eingegeben werden können

Vorzeichensymbol
(sign) Symbol, das zur Trennung und Unterscheidung von negativen und positiven Werten dient

Zugriffsart
Der Zugriff auf gespeicherte Daten ist jeweils abhängig von dem verwendeten Speichermedium (Cassettenrecorder oder Diskettenstation); RAM-Halbleiterspeicher erlauben einen direkten Zugriff, während Magnetbandspeicher nur einen sequentiellen Zugriff erlauben

Bildnachweise:
253: Ian McKinnel
254: MTV
255: Gary Marsh, BBC Stills
259: Chris Stevens
260: Ian McKinnel/Pilot Software
261: BL Systems, Roy Ingram
263: Tony Lodge
266: Ian McKinnel
267: Trevor Hill
274: Sirius
275, 276, 279: Liz Dixon
277: Alan Coode
280: Mark Watkinson

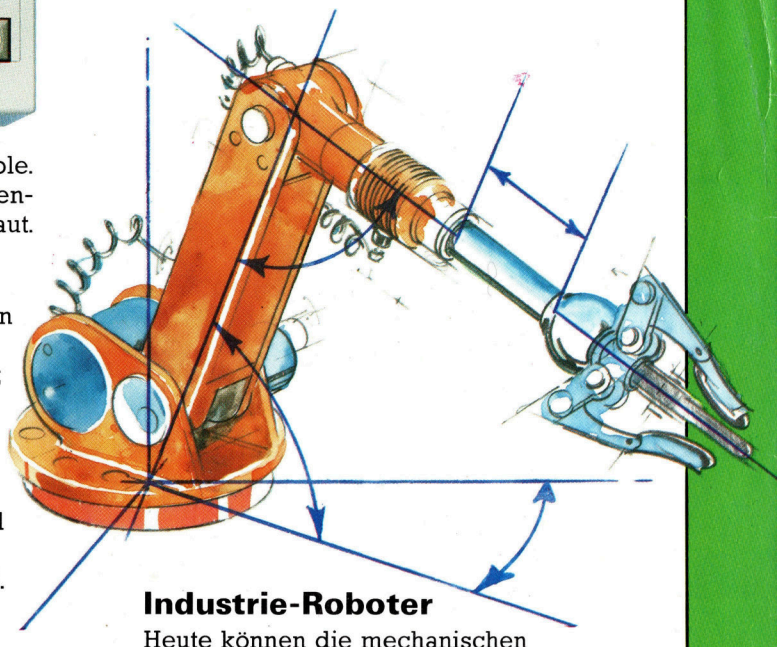
++ Vorschau +++ Vorschau +++ Vorschau +++

computer kurs Heft 11

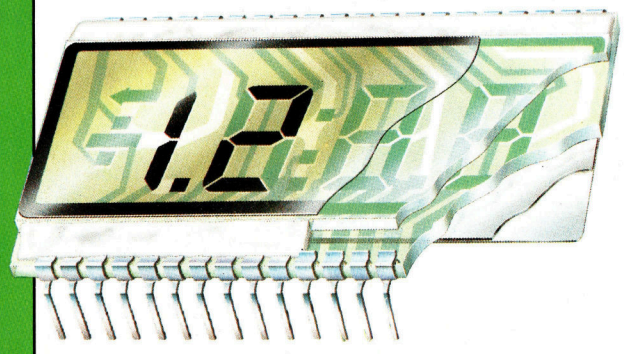
Portable Der Epson HX 20 ist ein echter Portable. LCD-Anzeige, Drucker und Cassetten-Laufwerk sind eingebaut.

Labyrinth

Irrgärten übten von je her Faszination auf Menschen aus; das ist bei Labyrinth-Computer-Spielen nicht anders. Im nächsten Heft wird beschrieben, wie sie aufgebaut sind.



Industrie-Roboter
Heute können die mechanischen Arbeiter bereits hochspezialisierte Montagearbeiten durchführen.



LCD-Anzeigen Flüssigkristalle finden im Computerbau häufig Verwendung.

+++ Das „Programm-Genie“ +++ Tips für die Praxis +++ Alles über Lichtgriffel +++ Fortsetzung der Logo- und Basic-Kurse +++ Flipper auf dem Bildschirm +++