

Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Heft **8**

Im Spielfieber

Lokale Netzwerke

Interpreter und Compiler

Microdrive

Sinclair ZX 81

Software-Piraten

Programmierkurse
BASIC und LOGO

Ein wöchentliches Sammelwerk

computer kurs

Heft 8

Inhalt

Computer Welt

Im Spielfieber 197

Die faszinierenden Arcadespiele

Analoge Systeme 216

Rechner, die direkt reagieren

Hardware

Lokale Netzwerke 200

Ein System – viele Anwender

Sinclair ZX 81 202

Tips für die Praxis

Der „Mini“ mit den vielen Extras 204

ZX 81 – handlich und ausbaufähig

Software

Dolmetscher für Maschinensprache 206

Compiler und Interpreter

Software-Piraten 208

Millionenschaden durch Raubkopien

Peripherie

Klein, aber oho! 210

Speichern mit dem Microdrive

BASIC 8

Arithmetische und String-Funktionen 212

Fragen und Antworten

Können Computer Sprache verstehen? 218

LOGO 8

Bauen Sie den „Turm von Hanoi“! 219

Bits und Bytes

Die Schaltzentrale 222

Wie die CPU arbeitet

Multiplizieren im Binärsystem 224

Fachwörter auf einen Blick

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 lesbar 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

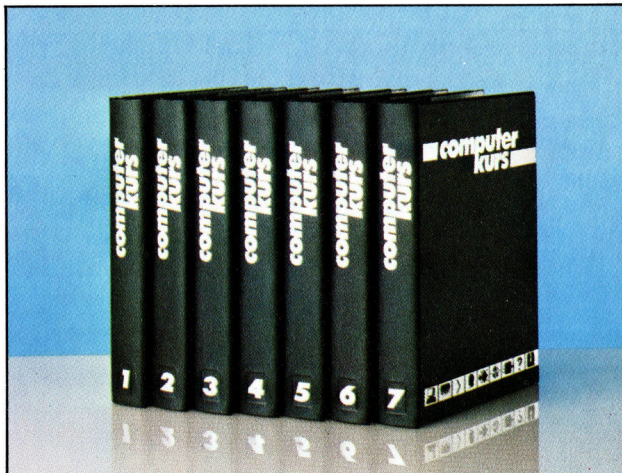
Schweiz: Der Sammelordner kostet sfr 15. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Sammelordner Computer Kurs, und notieren Sie Ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

INHALTSVERZEICHNIS

Alle 12 Hefte erscheint ein Teilindex; Heft 84 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





Im Spielfieber

Faszinierende Arcadespiele wie „Star Wars“ haben neue Dimensionen im Bereich der modernen Computergames eröffnet.



„Schnapp ihn, bevor er den Boden erreicht, oder er verdoppelt sich ... paß auf!! Das Mutterschiff nähert sich ... nimm die Sensorbombe ...“ Das ist kein Dialog aus einem Weltraumfilm, sondern eine Unterhaltung, wie man sie heutzutage in einer Spielhalle hören kann.

Im Jahre 1971 versuchte ein junger Mann namens Nolan Bushnell die Cafe- und Barbesitzer seiner Heimatstadt Sunnyvale davon zu überzeugen, einen neuen, von ihm erfundenen Spielautomaten aufzustellen. Der Apparat ließ sich wie andere Automaten durch Münzeinwurf in Gang setzen. Neu an der ganzen Sache war das Konzept, das man heute unter dem Namen „Interaktives Telespiel“ kennt.

Schließlich erklärte sich ein Gastwirt bereit, das Gerät auszuprobieren. Schon zwei Tage später aber rief er Bushnell ärgerlich an und beschwerte sich, daß das Gerät nicht mehr funktioniere. Als Bushnell sein Gerät daraufhin untersuchte, fand er die einfache Ursache dafür: Der Geldkasten war voll, und es ließen sich keine Münzen mehr einwerfen. Bushnell löste das Problem, indem er einen größeren Behälter einbaute.

Pong – eine Variante des Tischtennis (oder Ping-Pong) – war der Urvater all der spannenden und interessanten Arcadespiele, die auf

der Technologie des Microcomputers aufgebaut sind und sich heute weltweiter Beliebtheit erfreuen. Bushnell versuchte als erster, ein Spiel wie Tischtennis per Computer zu simulieren. Bei seinen Spielen waren jedoch zwei Mitspieler nötig, die gegeneinander und nicht einzeln gegen die Maschine antraten.

„Space Invaders“

Erstaunlicherweise dauerte die Entwicklung der nächsten Generation von Arcadespielen sehr lange. Erst 1977 brachte die japanische Firma Taito das äußerst erfolgreiche „Space Invaders“ auf den Markt.

In der Fachsprache fällt Space Invaders in die Spielkategorie „Feindabschuß mit Deckungsmöglichkeiten“. Hierbei bewegt der Spieler seine Kampfstation entlang des unteren Bildschirmrandes, versteckt sich hinter einer Deckung, wenn er bedroht wird, und feuert auf Feinde, die sich reihenweise in seine



Richtung bewegen. Der Angriffsrhythmus der Feinde ist vorbestimmt und wird von elektronisch erzeugten Tönen begleitet, die mit der Geschwindigkeit des Angriffs synchron laufen. Die Beschreibung des Spiels hört sich simpel an, der Gesamteindruck ist jedoch faszinierend.

Die ursprünglichen Space Invaders verwendeten einen monochromen Bildschirm und einfache Rastergrafik. Weder in der Hardware noch in den Programmen wurden neue Techniken angewendet – trotzdem löste das Spiel geradezu eine Revolution aus, wie man sie seit der Einführung des Kinos nicht mehr erlebt hatte. Da Space Invaders auf der gleichen Hardware basierte wie die heutigen Heimcomputer, bemerkten die Micro-Hersteller schnell, daß sich hier ein völlig neuer Markt auftat. Die Marketingstrategien konzentrierten sich jetzt nicht mehr auf das Selbst-Programmieren als Kaufanreiz, sondern auf den Computer als Unterhaltungsmedium. Schon bald schlug sich

computern profitierten von der Entwicklung dieser neuen Spielkonzepte. So wurde zum Beispiel Sprite-(Phantom) Grafik ursprünglich für Arcadespiele entwickelt und später auf Heimcomputer übertragen. Spezielle Grafik- und Tonerzeugungschips wie der Video-Interface- und Sound-Interface-Chip von Commodore entstammen der gleichen Quelle wie die drei Chips, die Atari für ähnliche Zwecke in seine Heimcomputerserie eingebaut hatte.

Erste Atari-Spielcomputer

Atari, dessen rapider Aufstieg auf dem ersten Spiel von Nolan Bushnell basierte, ist ein gutes Beispiel für den Technologieaustausch zwischen Arcadespielen und Heimcomputern. Lange bevor jemand daran dachte, den Heimcomputer als Unterhaltungsmedium anzusehen, erweiterte Atari sein Angebot an Microcomputern um spezielle Spielcomputer des Typs VCS (Video Cartridge System), mit denen arcadeähnliche Spiele in die Haushalte gebracht wurden. Da Atari selbst einer der größten Produzenten von Arcademaschinen ist, hat die Firma hier natürlich erhebliche Vorteile. Es gibt aber noch andere Firmen der Unterhaltungsbranche – wie Columbia Broadcasting System –, die einen großen Anteil dieses Marktes kontrollieren.

Alle Arcadespiele beruhen entweder auf Abläufen des realen Lebens wie z. B. Tischten-

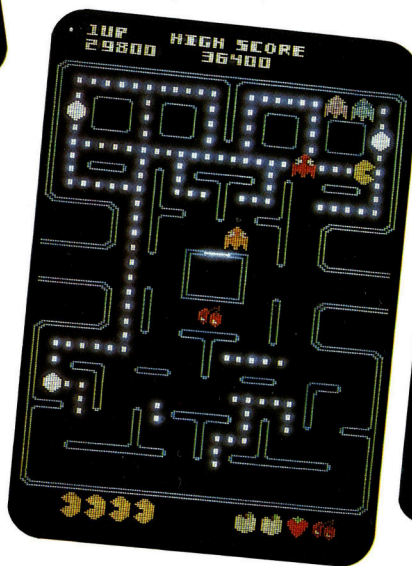
Die bewährten Labyrinthspiele wie PacMan sind inzwischen beliebter als Arcadespiele wie Space Invaders, in denen der Spieler sich nur auf einer festgelegten Linie bewegen kann. Beide Spieltypen verwenden Sprite-(Phantom) Grafik und haben daher nur mäßige Darstellungsqualität. Seit einiger Zeit gibt es aber Spiele wie „Battlezone“ oder „Tempest“, in denen der Spieler gegen futuristische Panzer und Raketen kämpft, und deren grafische Darstellung eine Illusion von Tiefe schafft.



diese Tendenz im Design der Geräte nieder. Sieben Jahre später werden die Geräte immer noch über ihre Spielmöglichkeiten vermarktet, obwohl auch heute selbst die besten Heimcomputer nicht an die der Arcadespiele heranreichen. Spielautomaten verfügen oft über Speicherkapazitäten von bis zu einem Megabyte und Grafikmöglichkeiten, die man sonst nur in den Spezialterminals der Groß-EDV für CAM und CAD findet.

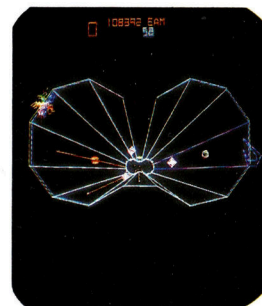
Viele Neuerungen im Heimcomputerbereich haben ihren Ursprung in den Arcadespielen. So wurde die Tendenz zu den 16- oder 32-Bit-Prozessoren dadurch ausgelöst, daß Anwender mehr als nur 64 KByte Speicher zur Verfügung haben wollten und schnellere Rechenzeiten benötigten. Die Hersteller von Arcadespielen gehörten zu den ersten, die 16-Bit-Prozessoren wie den 68000 von Motorola in ihre Maschinen integrierten, und ihr Bedarf an Verarbeitungsgeschwindigkeit und Speicherkapazität wuchs schneller als in der Branche der Bürocomputer.

Aber auch die Besitzer von einfachen Heim-



nis und Autorennen oder auf frei erfundenen Situationen wie „Space Invaders“, „PacMan“ und „Frogger“. In den letzten zwölf Jahren haben diese beiden Grundlagen sich zu den zwei Hauptrichtungen entwickelt, wobei aber weitaus mehr Phantasie- und Weltraumspiele entwickelt wurden als Programme zur Simulation von Realabläufen.

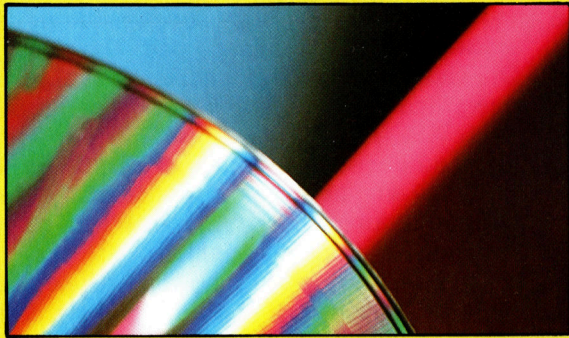
Die zwei „Ur-Spiele“ Pong und Space Invaders wurden bereits beschrieben. Wenn man





Raumstation

„Astron Belt“ repräsentiert eine neue Generation von Arcadespielen, in denen mit Bildplatten (siehe unten) ein beweglicher Hintergrund geschaffen wird, vor dem das Spiel abläuft. Da auf Bildplatten direkt (random) zugegriffen werden kann, ist es möglich, von einer Szene blitzartig auf eine andere umzuschalten. Die Darstellungen können Filmausschnitte oder Computerbilder sein. Astron Belt bietet Stereosound und einen vibrierenden Sitz, der über die niedrigen Frequenzen der Tonerzeugung gesteuert wird.



sich die Weiterentwicklung dieser beiden Hauptrichtungen betrachtet, erhält man einen Einblick in ihre Funktionsweisen. „Breakout“ war das erste Spiel, bei dem der Mensch gegen eine Maschine antrat. Der Spieler schlug einen Ball gegen eine Ziegelwand, aus der jeder Ziegel, auf den der Ball traf, vom Bildschirm verschwand. Ziel war es, alle Ziegel verschwinden zu lassen, ohne einen einzigen Ball zu verlieren. Auf diesem Prinzip aufbauend gab es bald Spiele, die Golf, Billard und Flipperautomaten simulierten. Die Spiele waren um so erfolgreicher, je perfekter es dem Programmierer gelang, die Eigenschaften der realen Welt (z. B. Wind, Schwerkraft, Oberflächenwiderstand oder fehlerhaften Abschlag) zu reproduzieren.

Spieler kontra Programmierer

Kriterien dieser Art lassen sich auf die Phantasiespiele nicht anwenden, denn hier tritt der Spieler eigentlich gegen den Programmierer an, dessen Regeln er sich in einem künstlich geschaffenen Universum unterwerfen muß. Auf die erste Version von Space Invaders folgten Spiele, in denen mehrere Arten von Gegnern nach unterschiedlichen Methoden zufallsgesteuert angriffen. Später konnte der Spieler seine Figur über den gesamten Bildschirm bewegen und es entstanden daraus Spiele wie „Defender“, die als die besten Ihrer Art betrachtet werden.

Auch Spiele, in denen sich der Spieler wie bei Space Invaders nur auf einer Linie bewegen konnte, entwickelten sich weiter. So muß z. B. bei „Missile Command“ und dessen Varianten eine Raketenbasis gegen die Angriffe

von Interkontinentalraketen verteidigt werden. Labyrinth- und Verfolgungsspiele wie PacMan entwickelten sich aus Autorennen, in denen Rennwagen in einer begrenzten Zeit so oft wie möglich unfallfrei über eine Bahn gesteuert werden mußten. Da in den Autorennen aber kein echter Wettbewerb der Maschinen möglich war – abgesehen vielleicht von Ölschleuren, die auf der Rennbahn plötzlich auftraten – wurden diese Zeitspiele bald in Verfolgungsjagden umgewandelt. Die Rennbahn verwandelte sich in Labyrinth, und aus Autos wurden Früchte, Glühbirnen und Münzen.

Die Autorennen selbst wurden durch die Simulation eines dreidimensionalen Hintergrundes immer realistischer gestaltet, so daß man aus der Sicht des Fahrers den Eindruck erhielt, daß die Straße sich auf das Auto hinbewegt. Ähnliche Darstellungsmethoden wurden auch in Flugsimulatoren verwendet.

Schließlich gibt es noch die traditionellen Brettspiele wie Dame, Schach und Backgammon. Diese Spiele existieren ausschließlich auf Heimcomputern, da sie viel Spielzeit erfordern und die Darstellung vor den eigentlichen Spielabläufen in den Hintergrund tritt.

Der wesentliche technische Unterschied zwischen Heim- und Arcadecomputern besteht heute hauptsächlich in den Methoden der Grafikerzeugung. Alle früheren und viele aktuelle Spiele verwenden Rastergrafik zur Darstellung, einige aber funktionieren nach der Vektormethode (z. B. Asteroids), bei der nur die Figuren auf dem Bildschirm – und nicht der Hintergrund – von dem Elektronenstrahl abgetastet wird.



Lokale Netzwerke

Der Zusammenschluß mehrerer Computer zu einem Netzwerk ermöglicht die gemeinsame wirtschaftliche Nutzung von Peripheriegeräten, Programmen und Daten.

Der Einsatz von mehreren Computern, beispielsweise in Großraumbüros, ist ein Beispiel für ein Netzwerk. Teure Peripheriegeräte sind allen angeschlossenen Rechnern zugänglich; ein Vorteil, der sich aus der Kopplung mehrerer Computer ergibt. Eine solche Vernetzung bietet darüber hinaus jedem Bediener die Möglichkeit, auf gemeinsame Datenbanken zurückzugreifen.

In Büros sind Microcomputer heute schon so weit verbreitet, daß sich in vielen Gebäuden – und auch schon in vielen Räumen – eine Anzahl Geräte gleicher Bauart befinden. Hier lohnt es sich, die Maschinen untereinander zu verbinden, um dadurch teure Peripheriegeräte wie zum Beispiel Diskettenstationen oder hochwertige Drucker allen zur Verfügung zu stellen und damit wirtschaftlicher zu nutzen.

Die Kosten für Computer sind heute ständig im Sinken begriffen. Technisch hochentwickelte Peripheriegeräte können inzwischen schon teurer sein als der Microcomputer, der sie steuert. So bietet zum Beispiel das Winchester-Laufwerk (eine Festplatte, die in einem luftdicht abgeschlossenen Kasten arbeitet) Zugriffszeiten und Speicherkapazitäten, die mit denen der Groß-EDV vergleichbar sind. Winchesterplatten erhielten ihren seltsamen Namen von dem 30/30-Repetiergewehr, da sie wegen ihrer Kapazität große Datenmengen schnell verarbeiten und außerdem mehrere Terminals gleichzeitig bedienen können. In

ähnlicher Weise mögen Matrixdrucker für die meisten Arbeiten völlig ausreichen, für professionelle Textverarbeitung wird aber oft ein Drucker mit hoher Druckqualität benötigt. Ein entsprechender Typenraddrucker ist jedoch teuer. Da ein einziger Anwender das Gerät meist nur für einen Teil der zur Verfügung stehenden Zeit nutzt, ist es wirtschaftlicher, Drucker dieser Art mehreren Arbeitsplätzen gleichzeitig zur Verfügung zu stellen.

Die gemeinsame Nutzung von Peripheriegeräten ist jedoch nicht der einzige Vorteil, der sich aus einem Zusammenschluß der Geräte untereinander ergeben kann. Innerhalb eines „Netzwerks“ können Microcomputer auch Daten austauschen. Werden Maschinen oder „Arbeitsplätze“ innerhalb eines Gebäudes zusammengeschlossen, so spricht man von einem lokalen Netzwerk (englisch: LAN – Local Area Network).

Eine Vernetzung bietet ihren Teilnehmern die Möglichkeit, auf gemeinsame Datenbanken zugreifen zu können. Es muß jedoch si-





chergestellt sein, daß diese Informationen nicht ohne Wissen der Teilnehmer verändert werden können. Macht zum Beispiel eine Fabrik über ein Netzwerk die Verfügbarkeit von Bauteilen und Rohmaterialien einer Reihe von Terminals zugänglich, muß jeder Teilnehmer mit aktuellen Informationen versorgt werden, wenn nicht schon nach kurzer Zeit ein Chaos entstehen soll.

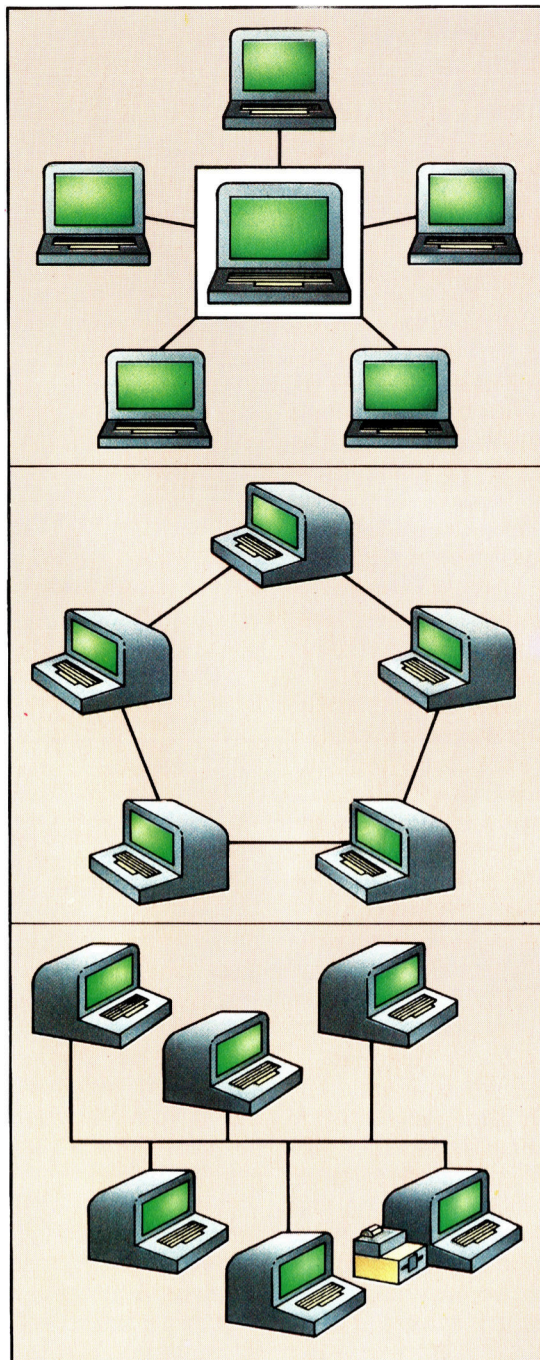
Werden in einem Netzwerk Microcomputer mit 64 KByte RAM oder mehr verwendet, so bietet es sich an, einen Teil der Datenbank in den RAM-Bereich zu laden und sich um die „Master-Datei“ so lange nicht mehr zu kümmern, bis ein anderer Teil davon benötigt wird. Wenn das Steuerprogramm Veränderungen der zurückgeladenen Dateisektion nicht entdecken kann, um die „Master-Datei“ auf den neuesten Stand zu bringen, besteht die Gefahr, daß Daten, die vom Anwender empfangen werden, bereits überholt sind.

Netzwerk Zeitungsredaktion

Netzwerke bieten aber noch weitere vielfältige Möglichkeiten. So können Dokumente von mehreren Personen nacheinander bearbeitet werden, ohne daß sie je auf Papier erscheinen müssen. Diese Technik wird heute schon von modern ausgerüsteten Zeitungs- und Zeitschriftenredaktionen verwendet. Dabei schreibt der Autor die Originalfassung eines Artikels an seinem Terminal und übermittelt sie über das Netzwerk dem Redakteur zur Korrektur. Als nächstes überprüft ein Lektor Rechtschreibung, Wortgebrauch und Grammatik und gibt den fertigen Artikel zur Druckvorbereitung an den Schriftsetzer weiter. Vor Einrichtung der lokalen Netzwerke mußten Artikel oft zwei- bis dreimal geschrieben werden, bis sie druckreif waren.

Netzwerkssysteme gibt es für viele Typen von Microcomputern, am weitesten verbreitet ist jedoch das Econet. Es benötigt einen Microcomputer als „Master“, um den zentralen Speicher und die Zugriffe der einzelnen angeschlossenen Terminals intern zu verwalten. Dieses Gerät kann entweder als „Master“ verwendet oder als normaler Microcomputer laufen, wenn die Netzwerkfunktionen nicht benötigt werden. Wird von mehreren Terminals jedoch ein einzelner Drucker zentral betrieben, muß dafür ständig eine Maschine zur Steuerung abgestellt sein.

Econet ermöglicht den Anschluß von bis zu 254 Arbeitsplätzen und zwei „Master“-Stationen. Die Grenzen des Netzes werden von der maximalen Entfernung eines Arbeitsplatzes von dem internen Taktgeber (der „Clock“) bestimmt (500 Meter). Der interne Taktgeber ist ein separater Kasten, der die Datenübertragungsrate innerhalb des Netzwerks synchronisiert. Der Aufbau des Econet-Systems ist relativ einfach, da wie bei dem Telefonsystem nur



Netzwerk-Strukturen

Stern

Bei einem sternförmigen Netzwerk ist jeder Arbeitsplatz mit der Zentrale verbunden, die wiederum alle Peripheriegeräte steuert.

Ring

Manche Netzwerke werden aufgrund spezieller Anforderungen in einem Ring zusammengeschlossen. Diese Methode ist weniger gebräuchlich, da Daten häufig erst die Hälfte aller Arbeitsplätze passieren müssen, bevor sie ihre Bestimmungsort erreichen.

Bussystem

Der Aufbau hochentwickelter Netzwerke wie Econet ist dem moderner Microcomputer sehr ähnlich: Daten und Steuerimpulse werden von dem Ursprungsterminal direkt an ihren Bestimmungsort gesendet.

zwei Kabelpaare benötigt werden. Ein Kabelpaar transferiert Daten und das andere die Taktimpulse, mit denen die Übertragung synchronisiert wird.

Die hochentwickelten Kommunikationsprogramme des Econet befinden sich auf einem EPROM, das in jedes Arbeitsplatzgerät eingebaut ist. Schwierigste Aufgabe des Systems ist die Vermeidung von „Zusammenstößen“, das heißt, es muß sichergestellt sein, daß zu jeder Zeit immer nur ein Teilnehmer des Netzes Informationen übermittelt. Andere Netzwerke, die mit Heimcomputern arbeiten, funktionieren auf ähnliche Weise, obwohl sie im allgemeinen nicht über die umfassenden Möglichkeiten des Econet verfügen.



Sinclair ZX 81

Der ZX 81 ist der „Volkswagen“ unter den Heimcomputern. Er ist einfach aufgebaut, robust und für Anfänger geeignet.

Der ZX 81 von Sinclair ist eine Weiterentwicklung des ZX 80. Von allen Computern, die Ergebnisse auf dem Bildschirm ausgeben können, kostet der ZX 81 am wenigsten. Die Zahl der Chips wird durch Verwendung eines ULA reduziert. Ein ULA (Uncommitted Logic Array) ist ein Baustein des Computers mit einer großen Anzahl festgelegter logischer Grundfunktionen. Der ZX 81 ist der „Volkswagen“ unter den Heimcomputern: einfache Ausführung ohne unnötige Knöpfe, Tasten oder Schalter.

Das externe Netzgerät wird seitlich angeschlossen. Weiterhin gibt es Buchsen für einen Cassettenrecorder, einen normalen Fernseher und weitere Peripheriegeräte. Der ZX-Drucker ermöglicht den preiswerten Ausdruck von Listings oder Ergebnissen. Allerdings ist das benötigte metallisierte Papier relativ teuer. Der ZX 81-Bildschirm stellt, untypisch für einen monochromen Schirm, schwarze Zeichen auf weißem Hintergrund dar. Er gibt nur Großbuchstaben wieder. Das Sinclair-BASIC unterscheidet sich zwar vom Microsoft-Standard, ist dafür aber besonders gut für Anfänger geeignet.

Auffallendes Merkmal ist die bedruckte Folientastatur. Unter den hellen rechteckigen Flächen befinden sich Sensortasten. Für die Textverarbeitung ist diese Folientastatur weniger geeignet, für das einfache Eingeben von Programmen ist sie durchaus brauchbar. Nach dem Einschalten des Computers erscheint ein K auf dem Bildschirm. Dies bedeutet, daß sich Ihr Gerät im Betriebszustand „keyword“ befindet: Wenn Sie die Taste L drücken, sehen Sie auf dem Bildschirm den BASIC-Befehl (keyword) LET, bei P erscheint PRINT. Über jeder

ULA

Die logischen Funktionen vieler kleiner Bausteine sind in einem ULA vereinigt. Man nennt diesen Baustein auch „Leimchip“: In ihm sind sämtliche wichtigen Computerfunktionen zusammen-„geleimt“. Durch diesen Baustein unterscheiden sich ZX 80 und ZX 81. Die Funktionen des ULA haben im ZX 80 noch zwölf kleine „Leimchips“ ausführen müssen, was seinen höheren Preis verursachte.

Taste finden Sie einen solchen Befehl. Schalten Sie in den Betriebszustand „function“ um, erscheint der Cursor als F. Auf dem Bildschirm wird eine der Funktionen dargestellt, die unter den Tasten aufgedruckt sind. Wollen Sie normalen Text eingeben, müssen Sie auf die Betriebsart „letters“ umschalten. Der Cursor erscheint als L.

Der ZX 81 ist schon lange genug auf dem Markt, so daß umfangreiche Programmsammlungen erhältlich sind. Einige seiner enthusiastischen Anhänger haben die Grundausstattung mit nur 1 KByte-Arbeitsspeicher als Herausforderung angesehen, unter dieser Voraussetzung Abenteuerspiele und sogar Schach im ZX 81 unterzubringen. Es gab einige Zeit lang eine Speichererweiterung von 16 KByte im Handel. Leider sorgte das steckbare Plastikmodul für großen Ärger, weil sein Stecker sehr locker saß. Die kleinste Erschütterung genügte, um den Kontakt zu lösen und mühsam eingegebene Programme verschwinden zu lassen. Sinclair hat den Versager mittlerweile durch ein zuverlässigeres RAM-Modul ersetzt.

Hunderte kleiner Firmen bieten Erweiterungen und Extras für den ZX 81 an, beispielsweise umfangreichere und zuverlässigere RAM-Module, Anschlußkabel für Drucker und Farbmonitore. Es werden sogar Gehäuse mit besseren Tastaturen und mehr Raum angeboten, in denen die meisten Erweiterungen und Extras geschützt Platz finden.

Tastatur

Unbefriedigend an dieser Folientastatur ist, daß sie keine „fühlbare“ Rückmeldung liefert. Um die vorhandene Fläche des kleinen Gehäuses optimal nutzen zu können, wurden die Tastenfelder mit mehreren Funktionen belegt. So können Sie beispielsweise mit der Taste L den Buchstaben L erzeugen oder den BASIC-Befehl LET, die BASIC-FunktionUSR oder das Symbol =.

Hf-Modulator

Erzeugt ein Hochfrequenzsignal, das ein normales Fernsehgerät empfangen kann.

Anschlußbuchsen

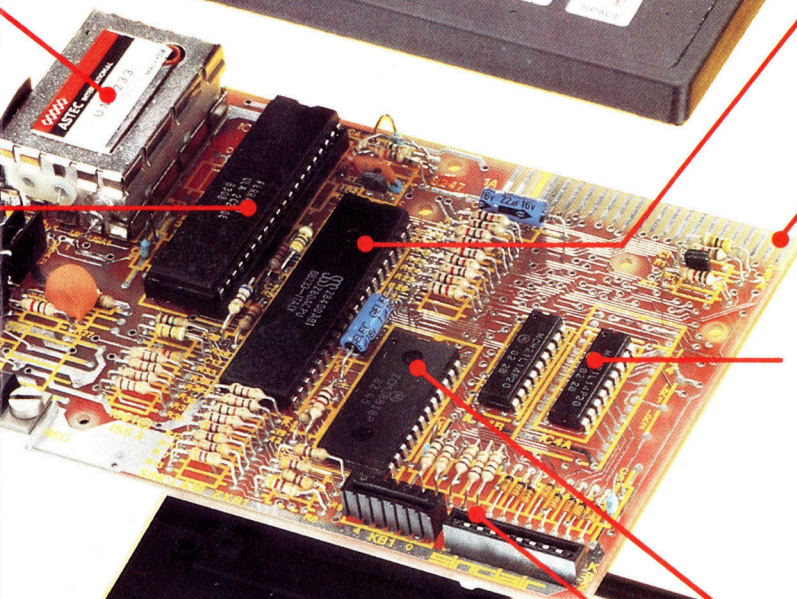
Drei Miniatur-Buchsen stehen für Versorgung und die Ein- und Ausgabe über einen Cassettenrecorder zu Verfügung.

Quantität gegen Qualität

Als Sir Clive Sinclair einen preiswerten Heimcomputer auf den Markt bringen wollte, wurde schnell deutlich, welche Kompromisse er eingehen mußte. Die Tastatur beispielsweise ist besonders schwach ausgefallen. Dieser Computer wird deshalb mehr für die Nutzung fertig gekaufter Programme eingesetzt und weniger zur Eingabe eigener Programme. Es besteht ein großes Angebot an Spielprogrammen für den ZX 81. Varianten von Breakout, Space Invaders und den sich gegenseitig verspeisenden Caterpillars waren nach kurzer Zeit auf dem Markt. Dienstprogramme für die Fehlersuche im Maschinencode und verschiedene betriebswirtschaftliche Programme folgten.

**CPU**

Der ZX 81 verwendet als CPU einen Standard-Z80-Prozessor.

**Steckleiste**

Hier werden Speichererweiterungen und weitere Peripheriegeräte aufgesteckt. Die Leiste ist von der Rückseite des Gehäuses her zugänglich.

RAM

Die Leiterplatte ist so ausgelegt, daß sich verschiedene Speicherbausteine einsetzen lassen. Der Arbeitsspeicher kann aus zwei kleinen RAM-Bausteinen bestehen.

ROM

Der Festspeicher enthält das Betriebsprogramm mit BASIC, dem Zeichengenerator und die Routinen zum Laden und Speichern der Cassetten. Ein anderer Computer mit einer Z80-CPU könnte den Inhalt des Sinclair-ROMs nicht lesen.

Tastatur-Anschluß

Die Tastatur wird an die ersten beiden Reihen der Kontaktstifte vorn rechts angeschlossen.

ZX 81**PREIS**

Etwa 135 DM, mit 16KByte RAM 85 DM

ABMESSUNGEN

175 x 168 x 43 mm

GEWICHT

300 g

TAKTFREQUENZ

1 MHz

MASCHINEN-SPEICHER

8 KByte ROM für BASIC-Interpreter und Betriebssystem. 1 KByte RAM, davon 123 Bytes für Systemdaten und die restlichen für BASIC-Programme und Bildschirmdarstellung. Eine Erweiterung auf 16 KByte ist für weniger als 100 DM möglich.

BILDSCHIRM-DARSTELLUNG

Schwarzweiß-Darstellung von 24 Zeilen zu je 32 Zeichen Text oder Grafik mit 64 x 44 Blöcken. Die unteren zwei Zeilen sind für die Darstellung der Eingabe reserviert.

SCHNITTSTELLEN

Standard-Buchsen für Stromversorgung, Cassettenrecorder und Fernsehgerät. Steckleiste für Speichererweiterung und weitere Peripheriegeräte.

PROGRAMMIERSPRACHE

Sinclair-BASIC

WEITERE PROGRAMMIERSPRACHEN

FORTH, ASSEMBLER für Z 80-Maschinen-code

ZUBEHÖR

Netzgerät, TV-Anschlußkabel, Anschlußkabel für Cassettenrecorder, Bedienungsanleitung

TASTATUR

Folientastatur mit 40 Sensortasten.

DOKUMENTATION

Die Bedienungsanleitung des ZX 81 setzt keine Computerkenntnisse voraus und hat am Ende jedes Kapitels viele Beispiele und Übungen.



Der „Mini“ mit den vielen Extras

Durch das riesige Angebot an Software und Zusatzgeräten läßt sich der ZX 81 zu einem leistungsfähigen Heimcomputersystem ausbauen.

Der ZX 81 ist zwar einer der kleinsten und billigsten Computer auf dem Markt, aber es gibt ein fast unerschöpfliches Software-Angebot für diesen Rechner. Obwohl der ZX 81 ab Werk keinen Ton, keine Farbe und keine hochauflösende Grafik besitzt, gibt es sehr ansprechende Programme. Wer mit den Grafikzeichen etwas experimentiert, kann relativ gute Bilder erstellen.

Wem die Grafikzeichen nicht ausreichen, der kann auf ein HRG-Modul zurückgreifen. Allerdings ist der Preis für dieses Modul um einiges höher als der Preis für den Rechner selbst. Zum Glück gibt es mittlerweile verschiedene Programme, mit deren Hilfe man hochauflösende Grafik gestalten kann. Natürlich muß man bei solchen Programmen Kompromisse eingehen. Sie sind nicht mit dem Modul zu vergleichen, aber angesichts der vorhandenen Grafikzeichen des ZX 81 bieten sie doch eine gute Alternative zum Modul. Zumal der Preis dieser Programme zwischen 15 und 30 Mark liegt und so für jeden erschwinglich sein dürfte.

Wem das noch nicht ausreicht, der kann sich

eine Soundbox anschaffen. Sie wird von verschiedenen Herstellern angeboten und kann an die Rückseite des ZX 81 angeschlossen werden. Die Tonmöglichkeiten sind, gemessen am Preis (etwa 100 Mark), recht gut. Es gibt auf dem Markt auch verschiedene Programme zur Tonerzeugung, aber die Qualität läßt sehr zu wünschen übrig. Bei diesen Programmen wird der Ton über die SAVE-Buchse abgegeben. Man braucht also noch einen Verstärker, um den Ton hören zu können.

Soundbox-Klänge

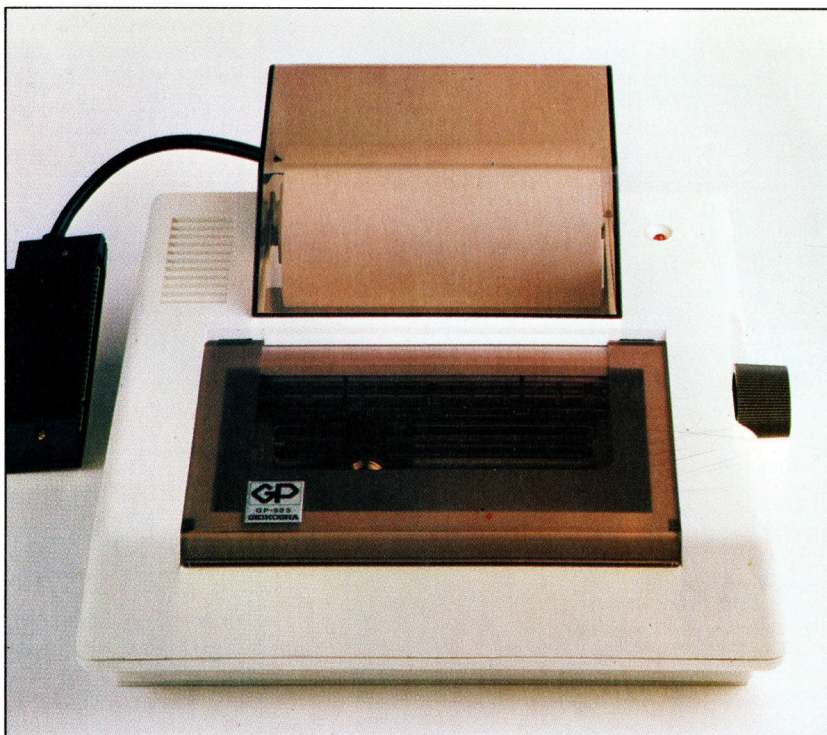
Bei einer Soundbox wird der Ton über einen eingebauten Lautsprecher abgegeben. Auf jeden Fall sollte man eine 16K-Erweiterung an die Steckleiste des ZX 81 anschließen, denn die meisten Programme laufen auf 16K. Neuerdings wird der ZX 81 komplett mit einer 16K-Erweiterung angeboten. Anscheinend sind einige Händler dahintergekommen, daß die 1K-Grundversion auf Dauer nicht ausreicht. Man kann den ZX 81 auch bis auf 64K erweitern. Dies ist allerdings nur dann sinnvoll und praktisch, wenn man den ZX 81 beispielsweise gewerblich verwenden will.

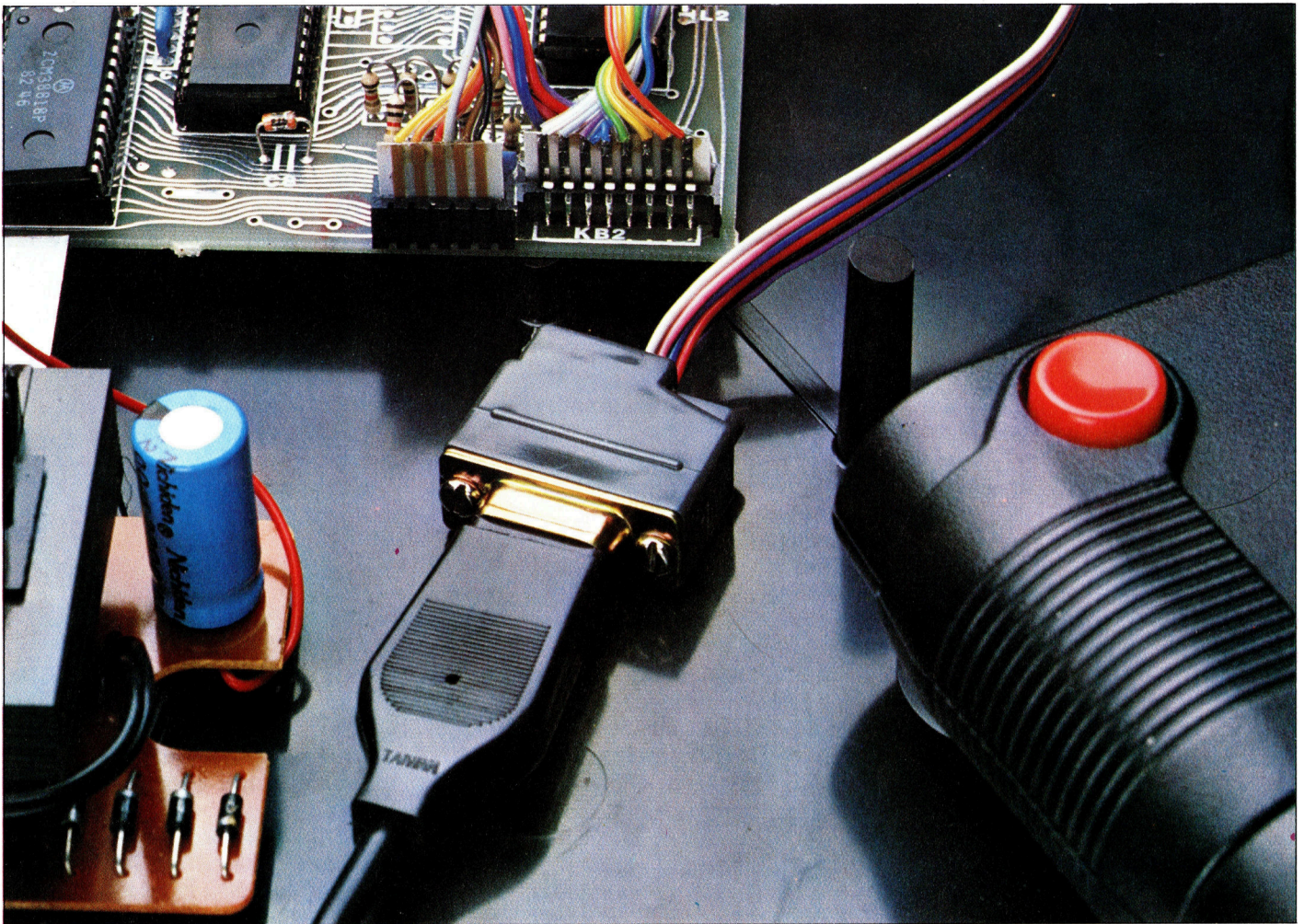
Zum Spielen gibt es alle bekannten Programme wie SPACE INVADERS, PAC MAN, DONKEY KONG. Da die meisten in Maschinensprache geschrieben sind, ist an der Geschwindigkeit nichts auszusetzen. Doch spätestens jetzt bemerkt man die größte Schwachstelle an dem ZX 81: seine Tastatur. Von der Folientastatur kann man kaum befriedigende Ergebnisse erwarten, weder beim Programmieren noch beim Spielen.

Will man mit dem ZX 81 hauptsächlich spielen, empfiehlt es sich, ein Joystick-Interface anzuschließen. Diese gibt es in verschiedenen Versionen zu kaufen (von 100 bis 200 Mark). Bei einigen gibt es eine Joystickbuchse, die der Tastenbelegung von Spielen einiger großer Softwareanbieter entspricht. Dies ist natürlich eine große Einschränkung, aber es gibt auch Interfaces, die frei programmierbar sind. Allerdings sind sie entsprechend teurer.

Für alle, die nicht so viel Geld haben oder ausgeben wollen, bieten einige Computerzeitschriften Bauanleitungen an. Es wurden schon öfters Bauanleitungen für das Joystickinterface

Dieser Matrixdrucker für den Sinclair Heimcomputer ZX 81 wird von der Firma Seikosha hergestellt. Das Gerät hat einen entscheidenden Vorteil gegenüber dem Original-Sinclairprinter: Es arbeitet mit Normalpapier und nicht mit teurem Metallpapier.





des ZX 81 veröffentlicht. Sie bieten in jedem Falle eine günstige Alternative. Will man mit dem ZX 81 selbst programmieren, ist auf die Dauer eine Zusatz tastatur unumgänglich. Es gibt schon sehr preisgünstige Tastaturen ohne Gehäuse zu kaufen. Mit ein wenig handwerklicher Begabung kann man ein Gehäuse selbst bauen. Empfehlenswert ist eine Tastatur mit Gehäuse, in das der ZX 81 eingebaut werden kann. Meistens sind diese Gehäuse so groß, daß man auch das Netzgerät und die Speichererweiterung darin unterbringen kann. Wer in Maschinensprache programmiert, sollte sich eine Tastatur mit einem Zehnerblock anschaffen. In diesem Falle sollte auch ein Assembler nicht fehlen.

Tempo durch M-Coder

Aber auch diejenigen, die von Maschinencode nichts verstehen, brauchen auf die Geschwindigkeit nicht zu verzichten. Es gibt verschiedene M-Coder zu kaufen, die selbstgeschriebene BASIC-Programme in Maschinenprogramme umwandeln. Zwar gibt es einige Befehle und Funktionen, die nicht in Maschinensprache umgewandelt werden, aber diese sind leicht zu umgehen, wenn man das Programm neu strukturiert. Für alle professionel-

len Maschinencode-Programmierer gibt es die Möglichkeit, ein Assembler-Betriebssystem in den ZX 81 einzubauen. Dadurch fallen alle BASIC-Befehle weg.

Dieser Chip bietet eine ungeahnte Vielfalt an Möglichkeiten. Beispielsweise hat er eine Repeat-Funktion zur Cursorsteuerung, scrollt auf Wunsch das Listing und vieles andere mehr. Man kann auch noch auf die Programmiersprache FORTH umsteigen. Zu diesem Zweck werden Pakete angeboten, die die entsprechende Software und Anleitungsbücher beinhalten. Von dieser Möglichkeit wird allerdings nur wenig Gebrauch gemacht. Das merkt man daran, das es nur sehr wenige, meist private Anbieter für Software in dieser Sprache gibt.

Natürlich kann man an den ZX 81 auch einen Drucker anschließen. Der Sinclair-Printer ist allerdings nicht zu empfehlen, da er mit Metallpapier betrieben wird. Er ist zwar in der Anschaffung billig, aber das Papier ist sehr teuer.

Der ZX 81 ist zwar ein kleiner und sehr billiger Computer, aber durch das große Angebot von Soft- und Hardware kann er zu einem leistungsfähigen System ausgebaut werden. Dies gilt gleichermaßen für Leute, die mit dem ZX 81 nur spielen, und für diejenigen, die ernsthaft programmieren wollen.

Mit diesem Adapter lassen sich sogar zwei Joysticks an den Sinclair ZX 81 anschließen. Für anspruchsvolle Spiele ist allerdings die 16 KByte-RAM-Erweiterung unabdinglich.



Dolmetscher für Maschinensprache

Zwei unterschiedliche Umwandlungsprogramme, Interpreter und Compiler, sorgen für die Übersetzung von Eingaben und Befehlen in die für Computer verständliche Maschinensprache.

Die ersten Computer besaßen noch keine Tastaturen. Programmbefehle mußten mühsam einzeln über eine Leiste mit acht Schaltern eingegeben werden, die entweder auf „an“ oder „aus“ gestellt werden konnten und damit einen bestimmten Befehl darstellten. Die Schaltmuster waren die ersten Entsprechungen des Maschinencodes.

In der logischen Weiterentwicklung ersetzte eine Schreibmaschinentastatur die Schaltleiste und englische Begriffe die ursprünglichen Schaltmuster. Am Ende entwickelten sich Hochsprachen wie BASIC, die das Programmieren in Maschinensprache verdrängten.

Der Computer veränderte sein Funktionsprinzip jedoch nicht, und noch heute arbeiten Prozessoren auf der Grundlage der alten Schaltmuster. Damit der Prozessor aber Programmbefehle einer Hochsprache ausführen kann, mußten in Maschinencode Übersetzungsprogramme geschrieben werden, die die Hochsprachen in eine für den Prozessor verständliche Form umwandelten. Diese Umwandlungsprogramme werden je nach Übersetzungsmethode Compiler oder Interpreter genannt.

Interpreter werden in Heimcomputern normalerweise gleich mitgeliefert und sind eine Methode, Hochsprachen in einen kostengünstigen Code zu übersetzen, den der Computer verstehen kann. Sie benötigen wenig Speicherplatz und stellen daher viel Speicher für Programme und Daten zur Verfügung.

Microcomputer, die weniger als 1600 Mark kosten, besitzen fast immer einen BASIC-Interpreter. Man gibt ein BASIC-Programm ein, tippt RUN und das Programm funktioniert – oder bricht mit einer Fehlermeldung des Systems ab wie z. B.:

```
SYNTAX ERROR ON LINE 123
```

Wenn man dann LIST eingibt, den Fehler korrigiert und wiederum RUN tippt, läuft das Programm oder stoppt bei dem nächsten Fehler usw. Inzwischen gibt es BASIC-Interpreter, die Fehler bereits bei der Eingabe von Programmzeilen anzeigen.

Zwar ist der Interpreter leicht anzuwenden, er weist aber einige Nachteile auf: Jedesmal,

wenn Sie RUN eingeben, muß er das Programm im Speicher finden, es Zeile für Zeile übersetzen und dann ausführen. Befindet sich in Ihrem Programm folgende Schleife

```
400 LET N=0
500 PRINT N
600 LET N=N+1
700 IF N<100 THEN GOTO 500
```

so muß der Interpreter die Zeilen 500 bis 700 hundertmal von neuem übersetzen und ausführen, als ob er nie vorher auf sie gestoßen wäre.

Compiler funktionieren nach einem völlig anderen Prinzip. Sie sind teuer, recht kompliziert und benötigen für sich und ihre Programme viel Speicherplatz. Um arbeiten zu können, brauchen sie fast immer Diskettenlaufwerke als Zwischenspeicher.

Compiler bieten auf der anderen Seite enorme Flexibilität und Geschwindigkeit. Unsere vier BASIC-Zeilen oben würden nur ein einziges Mal übersetzt werden, der entsprechende Code jedoch einhundertmal ausgeführt.

Dieser Zeitgewinn hat seinen Preis. Nehmen wir an, Sie haben einen BASIC-Compiler und möchten ein BASIC-Programm ausführen. Dafür müssen Sie zunächst den Editor laden und starten. Mit diesem Editor schreiben Sie das Programm und speichern es als sogenannten „Quellcode“ auf eine Diskette.

Programme müssen Namen haben, damit man sie im Speicher wiederfinden kann. Der Editor fragt daher zunächst nach dem Name des Quellprogramms. Programmnamen bestehen oft aus zwei Teilen: Der erste Teil ist die Bezeichnung wie z. B. TESTPROG – und der zweite Teil ist normalerweise ein Code aus drei Zeichen, der den Programm- oder Dateityp bezeichnet. Dieser Teil des Namens wird „Extension“ oder „Kürzel“ genannt. Ein BASIC-Programm mag den Code BAS als Kürzel haben, und Ihr Quellcode mag daher auf der Diskette unter dem Namen TESTPROG.BAS abgelegt sein. Jetzt geben Sie ein:

```
COMPILE TESTPROG.BAS
```

und der Computer lädt den Compiler, startet



Ein Programm compilieren

Ein Programm selbst zu compilieren ist weitaus schwieriger als die Verwendung eines Interpreters. Ein solches Programm läuft jedoch schneller ab. Dies sind die wesentlichen Schritte zur Erstellung eines compilierten Programms:

Der Editor wird geladen.

Das Programm wird in BASIC eingegeben

Das Programm wird als „Quellcode“ auf Diskette gespeichert.

Der Compiler wird geladen. Der Name des Quellcodes wird eingegeben.

Der Compiler produziert den „Objektcode“ – die Übersetzung des Programmes in Maschinensprache.

Fehler im Programm werden angezeigt und müssen im Quellcode korrigiert werden.

Sind keine Fehler vorhanden, kann der Objektcode direkt in die Maschine geladen und ausgeführt werden.

ihn und läßt ihn den Quellcode TESTPROG. BAS in Maschinencode übersetzen.

Nach kurzer Zeit hat der Computer Ihr Programm in einen „Object Code“ umgewandelt, den er unter dem Namen TESTPROG. OBJ auf der Diskette speichert. Das Kürzel OBJ zeigt an, daß sich hier ein in den Maschinencode übersetztes Quellprogramm befindet.

Während der Compiler das Quellprogramm übersetzt, überprüft er es auf Eingabefehler. Findet er Fehler, zeigt er sie in folgender Form an:

```
100 REED X:IF X=3(N+2) LET P=Q
      1           2           3
```

FATAL ERROR:—

- 1) //REED// UNRECOGNISED COMMAND
- 2) ///ILLEGAL OPERATOR HERE
- 3) ??'THEN' OR 'GOTO' EXPECTED HERE

Für jede Programmzeile, die einen Fehler enthält, wird eine derartige Nachricht auf den Bildschirm gebracht. Die Fehlermeldungen sind weitaus ausführlicher als bei einem BASIC-Interpreter. Ist ein Fehler aufgetreten, muß der Editor neu geladen und gestartet, der Quellcode neu von der Diskette eingelesen und korrigiert werden. Dann beginnt der Vorgang der Compilierung von neuem.

Sind keine weiteren Eingabefehler vorhanden, können Sie

RUN TESTPROG

eingeben, und das Programm funktioniert entweder, wie Sie es sich gedacht haben, oder es müssen noch weitere Änderungen vorgenommen werden.

Die Vorteile eines Compilers treten im Entwicklungsstadium eines Programmes nicht besonders zutage, obwohl die ausführlichen Fehlermeldungen eine große Programmierhilfe sein können. Compiler zeigen ihre Stärke erst, wenn ein Programm fertig ist und Sie RUN eingegeben haben.

Compilierte Programme sind bis zu 50 mal schneller als interpretierte Programme. Die Schnelligkeit eines compilierten Programmes wird jedoch durch mühselige und langsame Programmentwicklung erkauft.

Ist ein Programm erst einmal compiliert, so wird das Quellprogramm nur noch als Referenz benötigt und kann daher lesbar und mit ausführlichen Kommentaren versehen programmiert werden. Das compilierte Programm kann dabei weitaus kürzer werden als das Quellprogramm, da Kommentare nicht mit in den Maschinencode übersetzt werden.

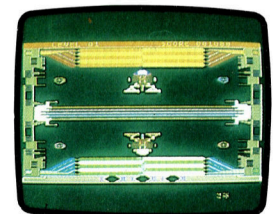
Die Tatsache, daß Objektcode aus unverständlichem Maschinencode besteht, kann auch von Vorteil sein. Softwarehäuser verkaufen allgemein nur Programme im Objectcode, der weitaus schwerer zu ändern ist und dessen Sicherungen kaum gebrochen werden können.



Für ein Abenteuerspiel ist Geschwindigkeit nicht wichtig, da der Großteil des Programms aus Texten besteht. Sie sind oft in BASIC geschrieben und werden während des Ablaufs übersetzt.



Weil sie sehr viele Berechnungen enthalten, können zahlreiche kommerzielle Programme (besonders Kalkulationsprogramme) nur unter großen Schwierigkeiten in Maschinencode programmiert werden.



Für schnelle Weltraumspiele mit sich ständig verändernder Grafik sind sogar compilierte Programme nicht schnell genug. Sie müssen direkt in Maschinencode programmiert werden.



Software-Piraten

Raubkopien verursachen den Software-Herstellern schmerzhafteste Verluste. Verständlicherweise ergreifen sie jede Möglichkeit, um ihre Programme vor unbefugtem Zugriff zu schützen.



Kopieren von Programmen ist Diebstahl. Leider sind die Produzenten oft im Nachteil, weil der Diebstahl schwer zu verhindern oder nachzuweisen ist und das Gesetz sie wenig vor Software-Piraten schützt. Einige Software-Produzenten behaupten, daß von jedem verkauften Programm bis zu hundert Raubkopien gemacht werden.

Das Software-Geschäft wird heute zunehmend von den gleichen Problemen geplagt wie die Musikindustrie. Unrechtmäßiges Kopieren von Programmen ist das mehr oder minder kriminelle Geschäft von Software-Piraten. Es beginnt schon damit, daß Heimcomputer-Anwender sich von einem Freund ein Programm leihen und kopieren. Ohne allzu große Schwierigkeiten lassen sich auch von Programmen Raubkopien anfertigen, die in Maschinencode geschrieben wurden und deshalb mit dem BASIC-Befehl SAVE nicht kopierbar sind. Zwei Cassettengeräte sind ausreichend, um die Programme von einer Cassette auf die andere überspielen zu können. Ein Computer ist dazu nicht erforderlich.

Nicht wenige Produzenten von Computerspielen behaupten, daß von jedem legal verkauften Programm bis zu 100 Raubkopien angefertigt werden. Man könnte natürlich argu-

mentieren, daß dies durchaus zu verkraften sei. Doch die Mehrzahl der Programmierer fährt keinen Rolls Royce, sondern verdient mit Programmeschreiben den Lebensunterhalt.

Je raffinierter Software-Piraten vorgehen, desto schwieriger wird es, ihnen den Raub nachzuweisen. Sie wenden ihre Tricks weniger bei Spielprogrammen, dafür um so häufiger bei kommerzieller Software an. Einige verändern die Originalprogramme, bevor sie sie als eigene Schöpfung weiterverkaufen. Diese Veränderungen können durchaus echte Verbesserungen der Leistungsfähigkeit oder Anwenderfreundlichkeit sein. Andere beschrän-

Cassetten mit Computerprogrammen werden wie Musik-Cassetten kopiert. Gewöhnlich benutzt man ein Kopiergerät, daß ein Haupt- und mehrere parallel aufzeichnende Nebenaufwerke besitzt. In das Hauptlaufwerk wird das Original gesteckt. Das Kopieren von beiden Seiten dauert nur einige Sekunden. Disketten müssen einzeln mit normalen Disketten-Laufwerken kopiert werden.



ken sich auf die bloße Änderung der Urheberangaben, die zu Beginn auf dem Bildschirm erscheinen, oder maskieren das Original durch veränderte Gestaltung des Schirmbilds.

Weil Uneinigkeit darüber besteht, ob veränderte Programme, genau wie die einfach kopierten, Raubkopien sind, gelingt es vielen „Räubern“, ungeschoren davonzukommen. Software-Produzenten werden von den Copyright-Bestimmungen wenig geschützt. Der Schutz beschränkt sich auf Gedrucktes, einschließlich Noten. Computerprogramme, die auf Cassette oder im RAM gespeichert sind, bleiben vogelfrei. Damit sich dies ändert, müssen entsprechende Gerichtsurteile vorliegen, und das braucht seine Zeit.

Ungenügender Ideenschutz

Sehr undurchsichtig wird die Rechtslage, wenn es um Ideendiebstahl geht: nicht das Programm, sondern Schirmbildgestaltung und Reaktionen des Programms auf Anwendereingaben kopiert der Plagiator. Das Programm schreibt er von Grund auf neu. Selten merkt der Anwender den Unterschied zwischen Original und Plagiat. Ein Beispiel für besonders eifrigen Ideendiebstahl ist PacMan, ein Computerspiel, das in Spielhallen berühmt wurde. Atari ist der Urheber einer Version dieses populären Spiels für Heimcomputer und Videocassetten-Systeme aus eigener Fertigung. Doch in kürzester Zeit wurden viele Varianten angeboten. Trotz leichter Unterschiede war allen die Grundidee des herumrausenden, alles verschlingenden, kleinen Vielfraßes gemeinsam. Die meisten Plagiatoren konnte Atari durch Gerichtsentscheid oder, wenn es kleinere Firmen waren, durch bloße Drohung mit dem Gericht ausschalten. Doch dies dauerte mehrere Monate.

Um Programme besser zu schützen, müssen Software-Autoren Mittel und Wege außerhalb der Gerichte finden. Einige vertreten die lobenswerte Meinung, daß Kopieren durch niedrige Verkaufspreise verhindert werden kann. Bei komplexen Programmen bietet ein gutes Anwenderhandbuch und attraktive Verpackung einen gewissen Schutz.

Die Registrierung der Käufer ist eine Möglichkeit, teure kommerzielle Programme zu schützen. Der Käufer kommt nur in den Genuß der Unterstützung durch den Hersteller, wenn er eine spezielle Karte aus dem Handbuch zurückgeschickt hat.

Zu den sogenannten „harten“ Methoden des Kopierschutzes gehören „Dongles“. Dies sind Module in Streichholzschachtelgröße, die in entsprechende Anschlüsse am Computer gesteckt werden müssen, damit das Programm läuft. Gewöhnlich enthält das Dongle ein ROM mit einem besonderen Code aus einer Reihe von Einsen und Nullen. Das Programm liest diesen Code in bestimmten Abständen. Sollte

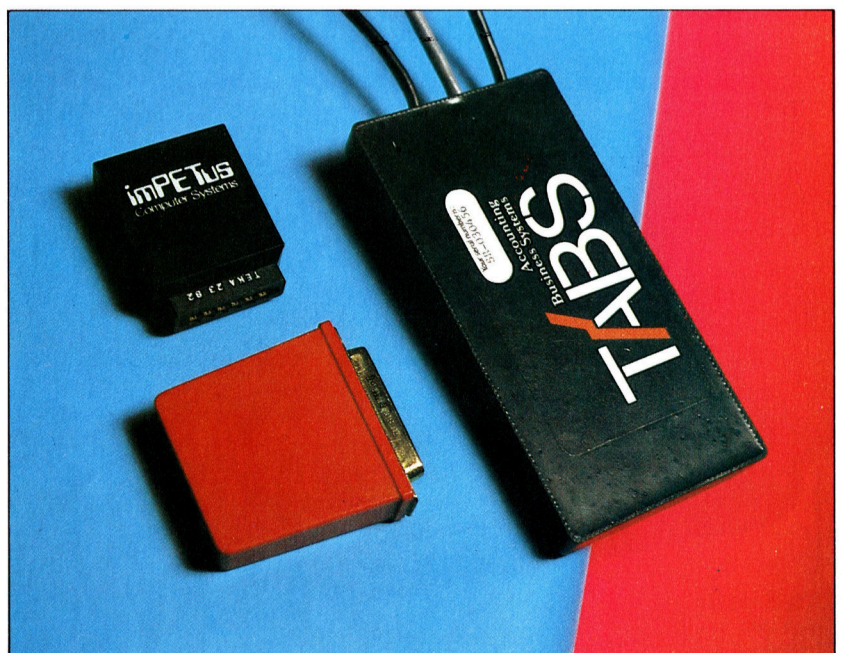
die gelesene Bitfolge den Erwartungen des Programms nicht entsprechen, bricht es einfach ab. Code im Dongle und der Kopie des gekauften Programms müssen aufeinander abgestimmt sein und können durchaus einmalig sein. Raubkopien sind daher nur möglich, wenn entweder das Dongle gefälscht wird, oder die Programmpassagen, die dessen Code lesen und vergleichen, gelöscht werden. Dies überschreitet die Fähigkeiten der meisten Heimcomputer-Anwender. Unmöglich ist es jedoch nicht.

Viele Untersuchungen sind mit dem Ziel unternommen worden, auch ohne Dongles den gleichen Kopierschutz zu erreichen. Die Idee wurde bekannt als „watermarking“, dem Wasserzeichen von Geldscheinen vergleichbar. Diese Wasserzeichen sind dem normalen Programm überlagerte Codes, die beim Kopieren nicht auf die andere Cassette oder Diskette übertragen werden. Die Raubkopie ist dadurch wertlos.

Für Produzenten von Computerspielen ist ein kleines ROM-Modul der einzig vernünftige Weg, ihre Ware zu schützen. Dieses Modul erzielt zudem einen höheren Preis, weil lange Ladezeiten, wie z. B. bei Cassetten, entfallen. Hundertprozentig sicher ist dieser Weg aber auch nicht, denn es gibt inzwischen Geräte, die das Kopieren von ROM-Modul auf Cassette oder ein anderes lösch- und wieder programmierbares ROM-Modul ermöglichen.

Software-Piraten und Urheber versuchen in einer Art Räuber- und Gendarm-Spiel, sich gegenseitig durch besonderen Erfindungsreichtum auszutricksen. Ganz verschwinden wird diese Form des geistigen Diebstahls sicher nie, doch lassen sich die Hürden so hoch stellen, das mit Raubkopien kein großes Geschäft mehr zu machen ist.

Dongles bieten Kopierschutz in Form von kleinen Hardware-Modulen. Dem Computer bleibt der Zugang zum Programm solange verschlossen, bis das zugehörige Dongle eingesteckt ist. Die Module sind häufig vergossen, um Eingriffe in die Elektronik zu verhindern.



Klein, aber oho!

Das Sinclair-Microdrive stellt gegenüber den üblichen Massenspeichern eine beachtenswerte Alternative dar: Ein Endlos-Magnetband ersetzt Cassette und Diskette.

Die üblichen Cassetten sind für Anwender von Heimcomputern ein erschwingliches und im allgemeinen auch zuverlässiges Mittel, Daten zu speichern oder gekaufte Programme zu laden. Doch auch das Cassettsystem hat seine Schwächen. Selbst schnelle Geräte übertragen nicht mehr als 1200 Bits pro Sekunde. Beim Laden umfangreicher Programme oder beim Suchen bestimmter Programmteile muß man mit einigen Minuten Wartezeit rechnen. Ein weiterer Nachteil ist, daß Computer den schnellen Vor- und Rücklauf des Cassettengeräts oft nicht steuern können. Befindet sich das gesuchte Programm am Ende des Bandes, so muß man lange spulen.

Diskettenlaufwerke weisen diese Mängel nicht auf, sie kosten aber auch entsprechend

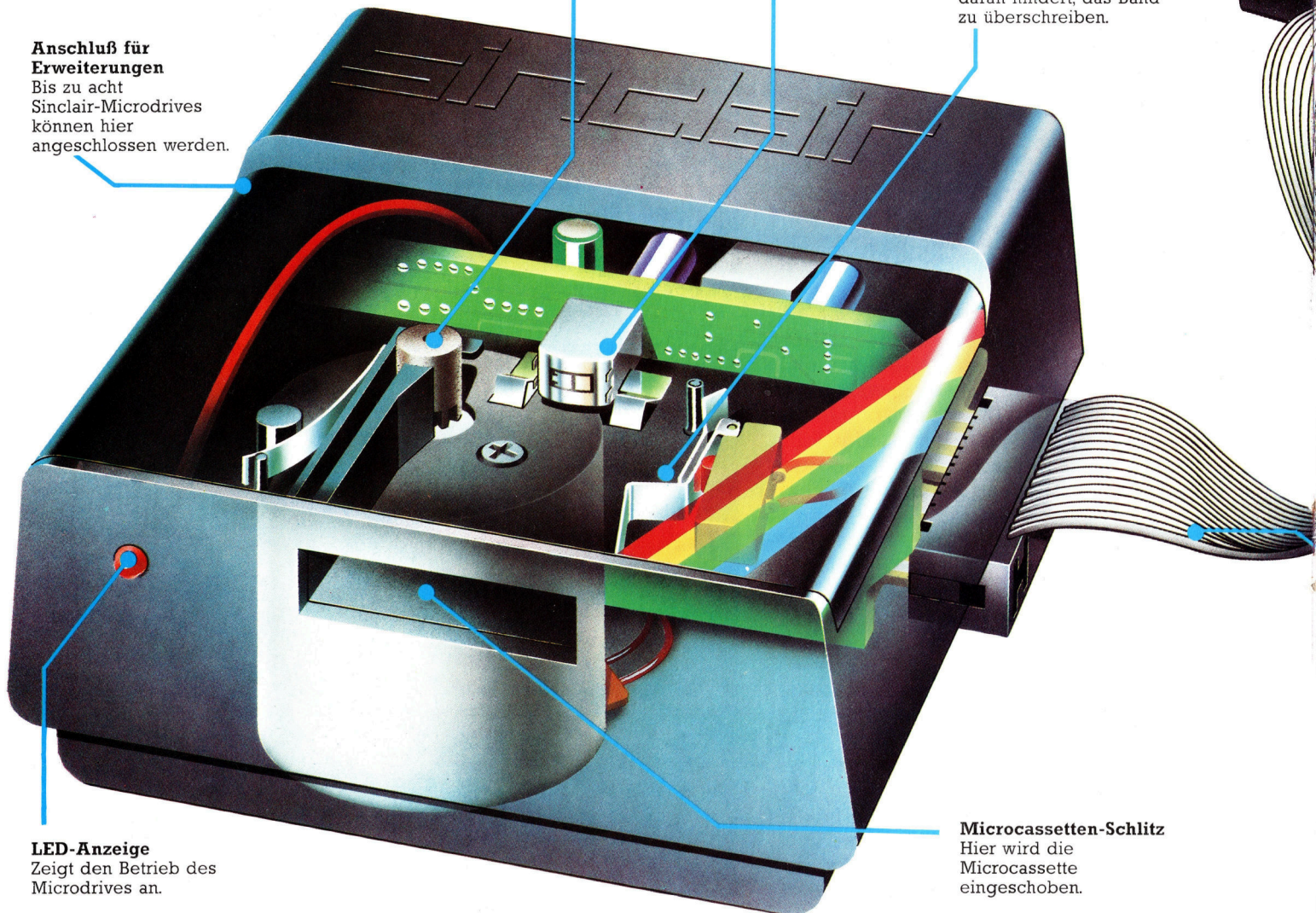
Band-Antriebsrolle
Die Bandschleife wird von dieser Rolle am Schreib- und Lesekopf vorbeigezogen. Die Rolle ist mit der Antriebsrolle in normalen Cassettengeräten vergleichbar.

Schreib- und Lesekopf
Dieser miniaturisierte Schreib- und Lesekopf ist dem Aufnahme- und Wiedergabekopf normaler Cassettengeräte ähnlich.

Schnittstelle
Über diese Schnittstelle werden die Microcassettengeräte angeschlossen. Gleichzeitig ist dies ein serieller Anschluß für Drucker und ein Netzwerk, um bis zu 64 ZX-Spectrums zusammenzuschalten.

Bandschutz-Microschalter
Microcassetten, bei denen der Schreibschutz entfernt wurde, betätigen den Microschalter, der seinerseits das Gerät daran hindert, das Band zu überschreiben.

Anschluß für Erweiterungen
Bis zu acht Sinclair-Microdrives können hier angeschlossen werden.



LED-Anzeige
Zeigt den Betrieb des Microdrives an.

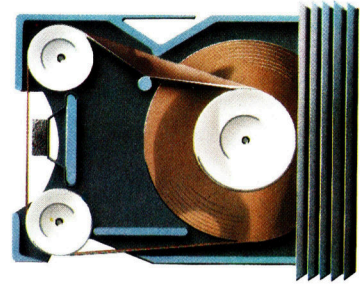
Microcassetten-Schlitz
Hier wird die Microcassette eingeschoben.



Schnittstellen-Kabel
Mit diesem Flachbandkabel wird das erste Microcassetten-Gerät an die Schnittstelle angeschlossen.

mehr. Was sich Heimcomputer-Anwender wünschen, ist ein Massenspeicher, dessen Datenübertragung um ein Vielfaches schneller als beim Cassettengerät ist, dessen Preis aber unter dem eines Diskettenlaufwerks liegt. Diese Massenspeicher gibt es unter der englischen Bezeichnung „floppy tapes“ oder „stringy floppies“, was auf die dünne Beschaffenheit des Bandes hindeutet. Die Firma Exatron in USA hat diese Art Massenspeicher für das Tandy TRS-80-Modell I entwickelt. Dieses Gerät verwendete Cassetten mit endloser Bandschleife. Die Grundidee stammte von den Audiocassetten mit acht Spuren, die einige Jahre zuvor Aufsehen erregt hatten. Das Prinzip ist einfach: Das Band bewegt sich mit konstanter Geschwindigkeit; gesuchte Programmteile werden in wesentlich kürzerer Zeit gefunden. Ein Verzeichnis aller Programme und Dateien (files) ist ebenfalls auf dem Band gespeichert. Es ist mit dem „directory“ von Diskettenlaufwerken vergleichbar.

Die Datenübertragungsraten sind bei diesen Massenspeichern mindestens fünfmal schneller als bei Audiogeräten, weil die Aufzeichnung nicht in analoger, sondern in digitaler Form erfolgt. Auch die Schnittstelle ist recht einfach: Sie hat acht parallele Datenleitungen.



Microcassette mit endloser Bandschleife

Die Microcassette von Sinclair verwendet ein 2 mm breites Video-Magnetband in Form einer geschlossenen Endlosschleife. Die Konstrukteure entschieden sich für Videoband wegen seiner langen Lebensdauer und seiner hohen Festigkeit. Ungefähr sieben Sekunden benötigte die Bandschleife der Microcassette für einen Durchlauf. Die Übertragungsrate liegt bei sechs KByte pro Sekunde. Dies ist eine erhebliche Verbesserung gegenüber den 1,5 KByte der Cassettschnittstelle beim Spectrum. Egal an welcher Stelle auf dem Band sich ein Programm befindet, in spätestens 15 Sekunden ist es im Arbeitsspeicher. Jede Bandschleife kann Datenmengen von bis zu 100 KByte speichern.

Das erforderliche Betriebsprogramm ist im Cassettengerät gespeichert oder befindet sich in einem ROM, das in einen freien Sockel des Computers gesteckt wird.

Auch das Microdrive von Sinclair nutzt das Prinzip der endlosen Bandschleife, die kontinuierlich am Schreib- und Lesekopf vorbeigezogen wird.

Ein äußerst zuverlässiger Massenspeicher mit einer hohen Übertragungsrate ist die Digitalcassette. Entsprechende Recorder gibt es seit längerem für professionelle Anwender. Aber sie sind teuer. Preiswerte Cassettengeräte wie der Hobbit erschienen, nachdem Philips seine Micro-Digitalcassetten auf den Markt brachte. Diese Geräte erzielten bemerkenswerte Übertragungsraten, obgleich sie die Grundmerkmale gewöhnlicher Audiocassetten beibehalten und das Prinzip der endlosen Bandschleife nicht angewendet wurde. Das Verzeichnis der aufgezeichneten Dateien oder Programme wird bei Micro-Digitalcassetten in der Mitte des Bandes gespeichert. Das Betriebssystem steuert alle Bewegungen des Bandes. Microcassettenrecorder gibt es inzwischen im PC-2151 von Sharp und im portablen HX-20 von Epson.

Die beschriebenen Geräte haben jedoch einen entscheidenden Nachteil: Sie verwenden unterschiedliche Aufzeichnungsformate, was den Programmaustausch und die Anzahl verfügbarer Programme stark einschränkt. Computer wie der PC-1251 und der HX-20 können Programme von einem normalen Cassettengerät laden und es dann über das eingebaute Gerät auf Microcassette speichern.

Arithmetische und String-Funktionen

In BASIC sind zahlreiche Funktionen bereits fest definiert. Der Programmaufwand kann erheblich reduziert werden, wenn man den Einsatz dieser Funktionen beherrscht.

Angenommen, Sie wollen in einem Ihrer Programme auch die Quadratwurzel einer Zahl errechnen. Um dies zu erreichen, kann man verschiedene Wege einschlagen. Der unkomplizierteste, jedoch zeitaufwendige wäre der, eine Tabelle mit den Quadratwurzelzahlen aufzustellen und aus dieser Tabelle dann die Quadratwurzel für eine bestimmte Zahl zu entnehmen – ein Weg, den Sie sicher von der Schule her kennen. Eine bessere Methode bietet die Programmiersprache BASIC mit der Quadratwurzel-Funktion, die in den meisten BASIC-Versionen eingebaut ist. Der Rechenvorgang wird automatisch von der Sprache erledigt. Das ganze sieht so aus:

```
10 REM DIESES PROGRAMM ERMITTELT DIE
20 REM QUADRATWURZEL EINER ZAHL
30 PRINT "GEBEN SIE DIE GEWUENSCHTE
   ZAHL EIN."
40 PRINT "QUADRATWURZEL VON"
50 INPUT N
60 LET A = SQR(N)
70 PRINT "DIE QUADRATWURZEL VON ";N;
   "IST ";A
80 END
```

Tippen Sie dieses kurze Programm ein. Sie werden feststellen, daß Sie die Quadratwurzel der Zahl erhalten, die eingegeben wird. Die Gesetzmäßigkeiten, nach denen diese Quadratwurzel-Funktion angewendet wird, sind folgende:

In BASIC ist eine „Funktion“ allgemein ein Befehlswort (in diesem Fall SQR), dem der zu berechnende Wert unmittelbar in Klammern folgt. Im obigen Programm ist N die über die Tastatur einzugebende Zahl, von der die Quadratwurzel ermittelt werden soll. Zeile 60 besagt: „Ordne die Quadratwurzel von N der Variablen A zu“, und Zeile 70 druckt diesen Wert aus.

Der in Klammern stehende Ausdruck wird als „Argument“ oder Parameter der Funktion bezeichnet. Er kann in Form einer Variablen oder einer Zahl eingegeben werden. Dazu ein Beispiel:

```
10 PRINT SQR(25)
20 END
```

Nach dem gleichen Schema lassen sich auch Operationen als Argument einsetzen:

```
10 LET A = 10
20 LET B = 90
30 LET C = SQR(A+B)
40 PRINT C
50 END
```

Kombiniert man die Zeilen 30 und 40, indem man den Umweg über die zusätzliche Variable C ausläßt, wird das Programm noch kürzer:

```
10 LET A = 10
20 LET B = 90
30 PRINT SQR(A+B)
40 END
```

Die meisten BASIC-Versionen beinhalten eine ganze Reihe von Funktionen und bieten dem Programmierer auch die Möglichkeit, neue Funktionen innerhalb eines Programms zu definieren.

Die gebräuchlichsten Funktionen kann man in zwei Kategorien einteilen: numerische Funktionen, in denen das „Argument“ (der in Klammern stehende Teil) eine Zahl, eine numerische Variable oder ein numerischer Ausdruck ist, und „String“-Funktionen, zu deutsch „Zeichenfolgenfunktionen“, in denen das Argument aus einem einzelnen oder mehreren alphanumerischen Zeichen besteht. Zunächst zu den numerischen Funktionen.

An früherer Stelle wurde ein Programm behandelt, das die Anzahl der für das Auskleiden eines Raumes notwendigen Kacheln errechnet. Die Berechnung in diesem Programm war jedoch nicht optimal, da das Ergebnis Bruchstücke einer Kachel enthalten konnte; ein mögliches Ergebnis konnte zum Beispiel die Zahl 988,24 sein. Heute wird ein Weg gesucht, der ein Ergebnis liefert, das auf die nächstliegende ganze Zahl gerundet ist.

```
10 PRINT "GEBEN SIE EINE ZAHL MIT
   DEZIMALBRUCH EIN"
20 INPUT N
30 PRINT "DER GANZZAHLIGE WERT IST ";
40 PRINT INT(N)
50 END
```



Wird beispielsweise die Zahl 3,14 eingegeben, so erscheint auf dem Schirm:

```
DER GANZZAHLIGE WERT IST 3
```

Da man Kacheln nur als Ganzes bekommt, muß eine 1 zu dem auf dem Bildschirm dargestellten Wert hinzugefügt werden, um ein korrektes Ergebnis zu erhalten.

Für andere Berechnungen ist es wichtig festzustellen, ob das Vorzeichen einer Zahl negativ oder positiv ist. Zum Lösen dieser Aufgabe benutzt man die SGN-Funktion.

```
10 PRINT "GEBEN SIE EINE ZAHL EIN."
20 INPUT N
30 LET S = SGN(N)
40 IF S = -1 THEN GOTO 100
50 IF S = 0 THEN GOTO 120
60 IF S = 1 THEN GOTO 140
100 PRINT "DIE ZAHL WAR NEGATIV."
110 GOTO 999
120 PRINT "DIE ZAHL WAR NULL."
130 GOTO 999
140 PRINT "DIE ZAHL WAR POSITIV."
150 GOTO 999
999 END
```

Die in Zeile 30 von der SGN-Funktion an S „zurückgegebenen“ Werte werden in den Zeilen 40, 50 und 60 überprüft. Das Programm unterscheidet zwischen drei möglichen Werten: -1, wenn der in Klammern stehende Parameter eine negative Zahl war; 0, wenn der Parameter Null war; 1, wenn der Parameter eine positive Zahl war. Durch die Verwendung der SGN-Funktion in Zeile 30 lassen sich Programmierzeilen wie die folgenden einsparen.

```
IF N<0 THEN LET S = -1
IF N = 0 THEN LET S = 0
IF N>0 THEN LET S = 1
```

Nun zu anderen numerischen Funktionen: ABS liefert den „absoluten“ Wert einer Zahl ohne Vorzeichen. Der absolute Wert von -6 ist beispielsweise 6. Versuchen Sie folgendes:

```
10 LET X = -9
20 LET Y = ABS(X)
30 PRINT Y
40 END
```

Mit Hilfe der Funktion MAX erhält man die höherwertige von zwei Zahlen:

```
10 LET X = 9
20 LET Y = 7
30 LET Z = X MAX Y
40 PRINT Z
50 END
```

MIN vergleicht ebenso wie MAX zwei Zahlen, stellt jedoch den kleineren Wert dar:

BASIC-Dialekte

TAB

Auf einigen Computern können Funktionen wie TAB, LEN, CHR\$ ohne Klammern verwendet werden; beispielsweise kann TAB(30) auch so geschrieben werden: TAB 30

LEFT\$

In manchen BASIC-Versionen sind diese Begriffe nicht vorhanden. Durch Ändern der Programmzeichen läßt sich jedoch der gleiche Effekt erzielen:

RIGHT\$

LEFT\$(Z\$,N) wird zu Z\$(TO N)
RIGHT\$(Z\$,N) wird zu
Z\$(LEN(Z\$)-N+1 TO)
MID\$(Z\$,N) wird zu Z\$(P TO P+N-1)

MID\$

Auch diese Funktion ist bei einigen BASIC-Dialekten nicht definiert. Durch kleine Subroutinen ist jedoch der gleiche Arbeitsablauf gewährleistet. Angenommen, eine Programmzeile lautet:

INSTR

```
20 LET P=INSTR(A$, "FALL")
Sie kann ersetzt werden durch:
20 LET X$=A$:LET Z$="FALL":GOSUB
9930:LET P=U
9929 STOP
9930 LET U=0:LET X=LEN(X$):LET
Z=LEN(Z$)
9940 FOR W=1 TO X-Z+1
9950 IF MID$(X$,W,Z)=Z$ THEN LET
U=W
9960 IF U=W THEN LET W=X-Z+1
9970 NEXT W
9980 RETURN
Auf dem Spectrum und dem ZX81 ist
Zeile 9950 so zu ersetzen:
9950 IF X$(W TO W+Z-1)=Z$ THEN
LET U=W
```

```
10 PRINT "GEBEN SIE EINE ZAHL EIN."
20 INPUT X
30 PRINT "GEBEN SIE EINE WEITERE ZAHL
EIN."
40 INPUT Y
50 LET Z = X MIN Y
60 PRINT Z
70 END
```

Beachten Sie bitte, daß diese beiden letzten Funktionen zwei Argumente anstelle eines einzelnen aufweisen und daß diese nicht in Klammern zu setzen sind. Bei den meisten BASIC-Versionen sind auch weitere numerische Funktionen wie LOG zum Auffinden des Logarithmus einer Zahl, TAN für den Tangens, COS für den Cosinus und SIN für den Sinus bereits definiert.

Nun zu den „String“-Funktionen. Einige dieser eingebauten alphanumerischen Funktionen haben Sie bereits im Namen-Sortierprogramm kennengelernt. Eine der gebräuchlichsten String-Funktionen ist LEN. Sie zählt die Anzahl der zwischen zwei Anführungszeichen stehenden Zeichen oder die Anzahl der einer String-Variablen zugeordneten Zeichen. Ein Beispiel:

```
10 LET A$ = "COMPUTER"
20 LET N = LEN(A$)
30 PRINT "DIE ANZAHL DER ZEICHEN IM
  STRING IST ";N
40 END
```

Das folgende Programm dient dazu, eine „Namen-Pyramide“ aufzubauen. Zuerst wird der erste Buchstabe des Wortes gedruckt, dann der erste und der zweite Buchstabe und so weiter, bis das ganze Wort gedruckt ist.

```
5 REM DIESES PROGRAMM DRUCKT EINE
  NAMEN-PYRAMIDE
10 LET A$ = "MEIER"
20 FOR L = 1 TO 5
30 LET B$ = LEFT$(A$,L)
40 PRINT B$
50 NEXT L
60 END
```

Man erhält folgenden Ausdruck:

```
M
ME
MEI
MEIE
MEIER
```

In diesem kurzen Programm wird die LEFT\$-Funktion dazu verwendet, Zeichen aus einem String herauszuziehen. Die LEFT\$-Funktion hat zwei Parameter; der erste spezifiziert die Zeichenfolge, und der zweite (der nach dem Komma folgt) steht für die Anzahl der aus dem String zu separierenden Zeichen und zwar von links beginnend. Der String „MEIER“ wurde der Variablen A\$ zugeordnet, also liefert LEFT\$(A\$,1) den Buchstaben M. LEFT\$(A\$,2) gibt die Buchstaben ME aus. Das obige Programm verwendet einen Index L, der den Bereich von 1 bis 5 umfaßt, so daß der zweite Parameter in der LEFT\$-Funktion bei jedem Schleifendurchgang um 1 erhöht wird. Da die Anzahl der im auszudruckenden Wort „MEIER“ enthaltenen Zeichen genau bekannt ist, ergibt sich, daß die 5 bei diesem Beispiel die obere Grenze in der FOR-NEXT-Schleife sein soll. Wie aber sieht die Eingabe aus, wenn die Anzahl der Zeichen in der Schleife nicht im voraus bekannt ist?

Dieses Problem läßt sich mit der LEN-Funktion lösen. LEN verwendet einen String (in Anführungszeichen) oder eine Stringvariable als Argument. Hier einige Beispiele:

```
10 REM DIESES PROGRAMM PRUEFT DIE
  LEN-FUNKTION
20 PRINT LEN("COMPUTER")
30 END
```

Läßt man das Programm laufen, erscheint eine 8 auf dem Bildschirm. Die Funktion hat die Anzahl der Buchstaben im Wort COMPUTER ge-

zählt und das Ergebnis dargestellt. Man kann dies auch auf andere Weise erreichen:

```
10 REM BESTIMMEN DER LAENGE EINES
  STRINGS
20 LET A$ = "COMPUTER KURS"
30 LET L = LEN(A$)
40 PRINT L
50 END
```

Der Wert der Variablen L ist in diesem Fall 13, da die Leerstelle zwischen den beiden Worten auch als Zeichen gerechnet wird. Nun wird die LEN-Funktion in einer Abwandlung des vorigen Programms zum Bau einer Buchstabenpyramide verwendet:

```
10 REM DIESES PROGRAMM DRUCKT EINE
  BUCHSTABEN-PYRAMIDE AUS
20 PRINT "GEBEN SIE EINEN NAMEN EIN."
30 INPUT A$
40 LET N = LEN(A$)
50 FOR L = 1 TO N
60 LET B$ = LEFT$(A$,L)
70 PRINT B$
80 NEXT L
90 END
```

Bei jedem Durchgang durch diese Schleife steigt der Wert von L schrittweise um 1, bis der Wert N, der die Länge des Namen-Strings enthält, erreicht ist. Wenn Sie den Namen SCHMIDT eingeben, wird Zeile 40 das Äquivalent zu LET N = LEN("SCHMIDT"), und für N wird der Wert 7 gesetzt. Beim ersten Schleifendurchgang stellt die Variable L in Zeile 50 den Wert 1 dar. Zeile 60 hat die gleiche Bedeutung wie LET B\$ = LEFT\$("SCHMIDT",1)

B\$ wird ein Zeichen aus dem String zugeordnet, und zwar der Reihenfolge nach von links nach rechts. Das erste Zeichen ist der Buchstabe S.

Beim zweiten Schleifendurchgang wird L die 2 zugeordnet. Damit werden die ersten beiden Zeichen aus dem String genommen und der String-Variablen B\$ zugeordnet, so daß B\$ nunmehr zwei Buchstaben, nämlich SC enthält. Nach dem letzten Schleifendurchgang enthält B\$ alle sieben Zeichen des ursprünglichen Eingabe-Strings. Danach wird B\$ ausgedruckt.

Die RIGHT\$-Funktion arbeitet ähnlich wie LEFT\$, jedoch werden Zeichen hierbei von rechts nach links definiert.

Eine weitere String-Funktion im vorliegenden Sortierprogramm ist INSTR. Sie dient dazu, die Stelle des ersten Auftretens eines bestimmten Teils eines Strings (den man mit „Subroutine“ bezeichnen kann) innerhalb eines Strings aufzufinden. In unserem Sortierprogramm wurde INSTR dazu verwendet, die Position der Leerstelle zwischen Vorname und Nachname zu lokalisieren. Dazu ein Beispiel:

```
10 LET A$ = "WASSERFALL"
```



```
20 LET P = INSTR(A$, "FALL")
30 PRINT P
40 END
```

Zur Arbeitsweise der INSTR-Funktion: INSTR lokalisiert das erste Auftreten des „Sub-Strings“. Ist der vollständige A\$-String WASSERFALL, dann ist die Startposition des Sub-Strings FALL die 7, weil das F der siebte Buchstabe im Wort WASSERFALL ist. Einige BASIC-Versionen arbeiten nicht mit INSTR, sondern mit der Funktion INDEX. Auf welche Weise INSTR (oder INDEX) dazu verwendet wird, die Leerstelle innerhalb eines Strings zu lokalisieren, sehen Sie hier:

```
10 REM AUFFINDEN DER POSITION EINER
LEERSTELLE IN EINEM STRING
20 LET A$ = "HEIMCOMPUTER"
30 LET P = INSTR(A$, " ")
40 PRINT P
50 END
```

Hier ist zu beachten, daß der zweite Parameter in der INSTR-Funktion (Zeile 30) " " ist. Die Anführungszeichen schließen eine Leerstelle ein, nämlich das „Zeichen“, das gesucht wird. Das Programm gibt die 5 als Wert für P aus, weil die Leerstelle die fünfte Position im String belegt. Welcher Wert wird zum Beispiel ausgegeben, wenn die Zeile 30 wie folgt geändert wird?

```
LET P = INSTR(A$, "C")
```

Versuchen Sie nun, die erläuterten Funktionen in das Sortierprogramm einzufügen.

Übungen

- Schleifen 1: Welchen Ausdruck liefert dieses Programm?

```
10 LET A = 500
20 FOR L = 1 TO 50
30 LET A = A - 1
40 NEXT L
50 PRINT "DER WERT VON A IST ";A
```

- Schleifen 2: Was erscheint auf dem Bildschirm, wenn dieses Programm läuft?

```
10 REM
20 REM DIES IST EINE ZEITSCHLEIFE
30 REM
40 REM
50 PRINT "START"
60 FOR X = 1 TO 5000
70 NEXT X
80 PRINT "STOP"
90 END
```

- Schleifen 3: Welches Ergebnis wird ausgedruckt, wenn Sie nach Aufforderung die Zahl 60 eintippen?

```
10 PRINT "GEBEN SIE EINE ZAHL EIN."
20 INPUT N
30 LET A = 100
40 FOR L = 1 TO N
50 LET A = A + 1
60 NEXT L
70 PRINT "DER WERT VON A IST JETZT ";A
80 END
```

- Schleifen 4: Was sehen Sie auf dem Bildschirm, wenn dieses Programm läuft?

```
10 FOR Q = 1 TO 15
20 PRINT "SCHLEIFEN OHNE ENDE"
30 NEXT Q
40 END
```

- READ 1: Welches Ergebnis erscheint?

```
10 READ X
20 READ Y
30 READ Z
40 PRINT "ICH TESTE DIE
READ-ANWEISUNG."
50 DATA 50, 100, 20
60 PRINT X + Y + Z
```

- READ 2: Welchen Ausdruck zeigt der Bildschirm, wenn dieses Programm läuft?

```
100 FOR L = 1 TO 10
110 READ X
120 PRINT "X=";X
130 NEXT L
140 DATA 1,2,3,5,7,11,13,17,19,23
```

Antworten zu den vorhergehenden Übungen

Variablen

A B6 X D\$ X\$ A12 D9 Q81 Q5 X H\$

Arithmetik 1

```
10 LET B = 6
20 PRINT B
```

Arithmetik 2

```
10 LET A = 5
20 LET B = 7
30 LET C = 9
40 LET D = A+B+C
50 PRINT D
```

Arithmetik 3

17

Vergleich 1

5

Vergleich 2

601 (Zahlen sind angenommen)

Vergleich 3

10000

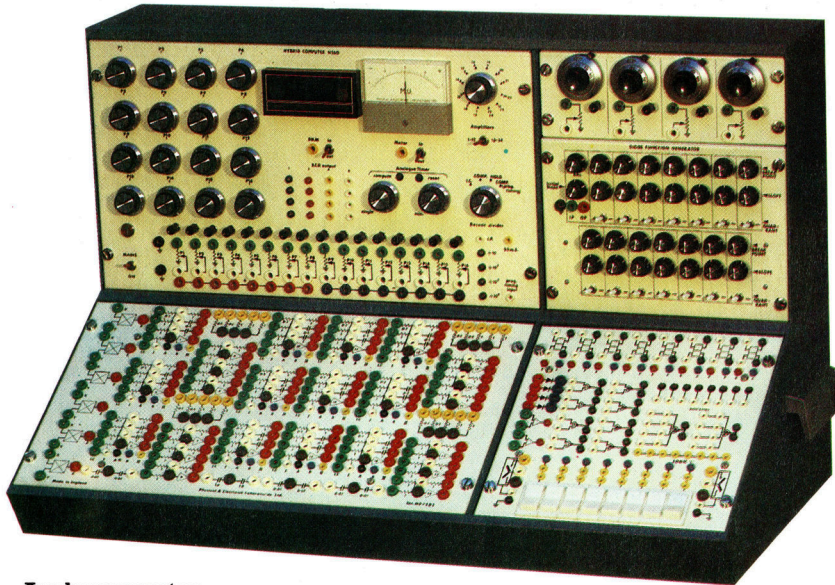
PRINT-Übung

Print "DER WERT VON T IST ";T



Analoge Systeme

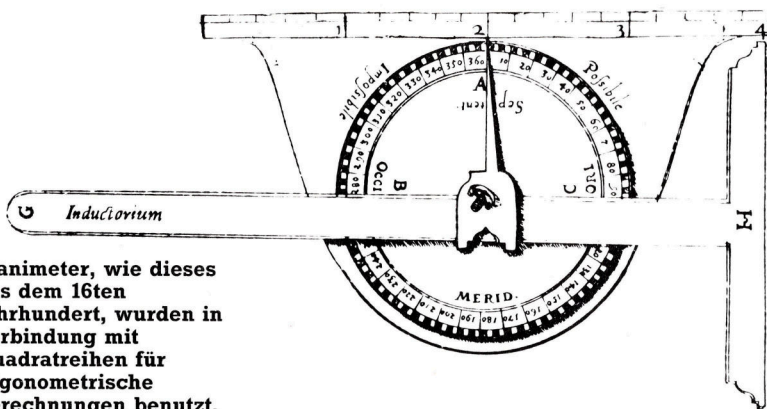
Im Gegensatz zu digitalen Rechnern reagieren analoge Computer direkt proportional auf Veränderungen, ohne eine Umsetzung in digitale Werte zu benötigen.



Analogcomputer benutzen keine Sprachen wie zum Beispiel BASIC. Sie werden „programmiert“, indem man die einzelnen elektrischen Einheiten, die sich auf der Rückseite befinden, mit Kabeln verbindet.

Nicht alle Computersysteme verarbeiten Daten digital, also über Binärziffern (digits). Es gibt auch Systeme, die Daten und Informationen analog umsetzen. Der Tachometer beispielsweise ist ein einfacher analoger Computer, denn die Geschwindigkeit des Fahrzeuges ist direkt proportional zur Position

Planimeter, wie dieses aus dem 16ten Jahrhundert, wurden in Verbindung mit Quadratreihen für trigonometrische Berechnungen benutzt. Die Geometer mußten Messungen durchführen und später mathematische Berechnungen, um dadurch die lange Dreiecksseite zu bestimmen. Dieses Analoginstrument vereinfachte die Arbeit ganz erheblich, und führte zu exakteren Ergebnissen.



der Tachonadel.

In der Entwicklung der Elektronik entdeckte man schon früh die Parallelität zwischen elektronischen Bauelementen, wie zum Beispiel einer Spule, und mechanisch arbeitenden Geräten. So fanden Ingenieure heraus, daß die Schwingungen in einem Stromkreis, der aus einer Spule und einem Kondensator besteht, den gleichen Gesetzen unterliegen wie die Schwingungen eines Körpers, der an einer

Stahlfeder hängt. Beide physikalischen Vorgänge haben ein und dieselbe mathematische Grundlage – die Basis für die Entwicklung analoger Computer war geschaffen.

Moderne analoge Computer können die vielfältigsten Aufgaben übernehmen. Ihre elektronischen Bauelemente, wie Transistoren, Kondensatoren oder Widerstände, findet man in vielen Haushaltsgeräten wieder. Einige dieser Analoggeräte sind den Originalen fast wirklichkeitsgetreu nachgebildet. So zum Beispiel Flugzeugmodelle für Windkanaltests. Diese Modelle sind exakte, verkleinerte Kopien des Flugzeugrumpfes. Es gibt aber auch analoge Darstellungen, die völlig abstrakt erscheinen. Die Darstellung einer bestimmten Lebenssituation kann – auf analoge Weise – in einer Liste mathematischer Formeln umgesetzt sein, und ein Schaltkreis kann beispielsweise den Wasserfluß eines Stauwehres symbolisieren.

In der Praxis wurden analoge Prinzipien bereits Anfang des 17. Jahrhunderts angewendet, und zwar bei dem 1630 von William Oughtred entwickelten Rechenschieber. Beim Rechenschieber werden Zahlen auf zwei beweglichen Schienen so verteilt, daß bei Verschiebung der Schienen multipliziert werden kann. Das Ergebnis kann dann direkt auf einer der Schienen abgelesen werden. Ende des 19. Jahrhunderts entwickelte Lord Kelvin eine Handkurbelmaschine, mit der man den Tidenhub, also die Wasserstandsveränderung von Ebbe zu Flut, in einem Hafen berechnen konnte.

Addieren durch Stromstärke

Der erste elektronisch gesteuerte analoge Computer wurde 1947, kurz nach der Entwicklung der digitalen Rechner, in Betrieb genommen. Wie bereits erläutert, führen Digitalrechner Rechenoperationen mit Hilfe sogenannter logischer Gatter durch. Analoge Computer hingegen nutzen bei ihren Kalkulationen die speziellen Eigenschaften der Elektrizität aus. Dazu ein Beispiel: Durch einen Draht fließt Strom von fünf Ampere, durch einen zweiten Draht Strom von vier Ampere. Werden nun beide Drähte miteinander verbunden, dann beträgt die Stromstärke neun Ampere, also genau die Summe aus beiden. Allein durch die Eigenschaften des Stroms entsteht ein „Addierer“.

So lassen sich oft schwierige mathematische



Probleme durch einfach aufgebaute Schaltkreise lösen. Ein Beispiel hierfür ist die Integralrechnung, die Berechnung einer Fläche unter einer Kurve. Da analoge Computer anders „programmiert“ sind als digitale Rechner, muß zur Lösung des Problems ein Schaltkreis konstruiert werden, der das Problem auf elektronische Weise repräsentiert. Dazu werden alle Bauelemente hinter ein Steckbrett montiert, auf dessen Vorderseite die Verbindungen zwischen den einzelnen Bauteilen über Stecker und Kabel hergestellt werden. Eine solche Anlage ähnelt sehr stark einem altmodischen Telefonschaltbrett.

In einem analogen Computer werden die veränderlichen Ströme und Spannungen zur Darstellung verschiedener realer physikalischer Größen, wie beispielsweise Kräfte oder Geschwindigkeiten genutzt. In Digitalrechnern dagegen werden alle Daten durch eine Reihe elektronischer Impulse dargestellt, die binäre Eins durch fünf Volt, die Null durch null Volt. Ein weiterer Unterschied zwischen einem analogen und einem digitalen Computer liegt in der Art der Speicherung der Daten. In Analoggeräten werden Informationen durch sich verändernde Werte gespeichert, in Digitalgeräten durch einzelne, „diskrete“ Einheiten.

Fehlerlose Datenübertragung

Der große Vorteil der Digitalrechner liegt in ihrer Fähigkeit, Daten ohne Verluste und fehlerlos übertragen und bearbeiten zu können. Wird ein Stromstoß von zehn Volt durch einen Schaltkreis geschickt, so kann er durch Materialeigenschaften abgeschwächt werden und dann nur noch 9,9 Volt betragen. In einem analogen System, in dem Schwankungen der Stromspannung eine Änderung der Information bedeuten, würde der Verlust von 0,1 Volt zu einer neuen Information führen. Stellen beispielsweise zehn Volt den Ton A einer Singstimme dar, so verändert sich der Ton bei einem Spannungsabfall auf 9,9 Volt unter Umständen schon auf Gis.

In einem digitalen System hingegen gibt es nur zwei mögliche Signale, entweder 5 oder 0 Volt, und jede Spannung in der Nähe von 5 Volt wird automatisch als 5 Volt erkannt und genau auf diesen Wert gebracht. So werden Abweichungen korrigiert und Fehler können sich nicht gegenseitig verstärken. Dies ist der Nachteil der analogen Computer, daß Signale beim Durchlaufen mehrerer Schaltkreise immer mehr verfälscht werden können und der Fehler dadurch zunehmend verstärkt wird.

Trotzdem erzielen analoge Computer einen Vorteil in der Darstellung von Größen durch verschiedene Stromstärken oder -spannungen. Denn das System reagiert sofort, sowie sich die Eingangswerte ändern. Dies gilt auch für sehr schnelle Änderungen, da keine Zeitverluste durch die Codierung der Größen in digitale

Analoger Regelkreis

Stoßdämpfer bestehen aus einer Feder und einem Dämpfer. Die Feder absorbiert plötzliche Stöße und die entstehenden Schwingungen werden durch den Dämpfer unterdrückt. Ingenieure müssen das beste Verhältnis von Feder und Dämpfer wählen, um den optimalen Fahrkomfort jederzeit zu gewährleisten. Mit einem elektrischen Schaltkreis kann man das System aus Feder, Dämpfer, Auto und Rad nachahmen. Man simuliert die Stöße durch eine Wechselspannung und verändert die elektrischen Werte, bis die Ausgangsschwingung so klein wie irgend möglich ist.

Werte, deren Bearbeitung und die Decodierung für die Ausgabe auftreten. Diese Eigenschaft ist für Anwendungen, die schnellste Reaktionen erfordern, unentbehrlich. Beispielsweise muß ein Autopilot im Verkehrsflugzeug beim Landeanflug auf eine Windböe mit einer Geschwindigkeit reagieren, die über längere Berechnungen selbst eines modernen Digitalcomputers nicht zu erzielen ist. Sensoren registrieren die Windböe und generieren eine kleine Spannung. Der Schaltkreis des Autopiloten reagiert unverzüglich mit einer relativ größeren Änderung der Ausgangsspannung, die automatisch die Querruder steuert, um das Flugzeug in seiner alten Fluglage zu halten.

Aber auch wenn die analogen Rechner weiterhin Anwendung finden, so werden sie doch durch die technische Fortentwicklung der digitalen Systeme, vor allem hinsichtlich einer größeren Verarbeitungsgeschwindigkeit, immer stärker vom Markt verdrängt werden.

Falsch verstanden!

Die Art, in der ein solcher Satz verändert wird, ähnelt sehr stark den akkumulierenden Fehlern in einem analogen Schaltkreis.

In einem digitalen System können nur zwei Meldungen übertragen werden: Null und Eins. Falls also eine Veränderung auftritt, kann sie leicht erkannt, und in der nächsten Stufe korrigiert werden.

Können Computer Sprache erkennen?

?

■ Gibt es schon Heimcomputer, die die menschliche Sprache verstehen können?

Der Bereich der menschlichen Spracherkennung befindet sich noch im Anfangsstadium der technologischen Entwicklung. Das größte Problem liegt in der unterschiedlichen Betonung und Geschwindigkeit, mit der Menschen sprechen. Aufgrund der Vielzahl aller möglichen Silben- und Wortkombinationen sind zudem riesige Speicherkapazitäten notwendig, damit der Rechner aufgrund von vorher eingegebenen Sprachmustern Vergleiche anstellen kann. Einen interessanten Weg beschritt die Firma Texas Instruments. Sie bietet einen Personal-Computer an, der in gewissem Umfang menschliche Worte erkennt. Dafür müssen verschiedene Worte, die auf dem Bildschirm erscheinen, öfters vom Anwender nachgesprochen werden, damit der Rechner sich das Klangmuster merken kann. Mit einem auf die Spracherkennung besonders abgestimmten Prozessor gehen die Fähigkeiten des Computers immerhin so weit, daß kleine BASIC-Programme eingesprochen werden können, ohne eine Taste zu berühren. Heimcomputer bieten derartigen Komfort bislang

nicht. Aber es gibt schon Software, die einige in ein Mikrofon gesprochene Befehle erkennt und in Anweisungen für Computerspiele umsetzt.

?

■ Wie gut eignen sich Joysticks für Grafik-Software?

Das Zeichnen mit dem Computer wird durch die Verwendung eines Joysticks erheblich vereinfacht, da der Cursor viel schneller auf dem Bildschirm an jede gewünschte Position gebracht werden kann. Mit der Tastatur sind solche Bewegungen nur schrittweise zu bewerkstelligen. Alternativen zum Joystick bieten die Maus, der Lightpen, das Grafiktablett oder das „Koala Pad“. Mit diesen Geräten läßt sich der Cursor noch genauer steuern. Das Grafiktablett hat zudem den Vorteil, daß der Cursor nicht nur der Bewegung folgt, sondern auf der Eingabefläche schon im voraus mit dem Zeichenstift genau bestimmt werden kann, wo der Cursor auf dem Bildschirm erscheint. Demgegenüber wird die Maus über eine beliebige Tischfläche geführt, die der Bildschirmfläche entsprechenden Koordinaten stehen nicht von vornherein fest.

?

■ Ist das angekündigte MSX ein neues Betriebssystem?

MSX steht für „Micro-Soft Extended BASIC“ und leitet unter japanischer Federführung einen neuen Standard für Heimcomputer ein. Nicht nur Betriebssystem, BASIC und Bus sind identisch, sondern alle MSX-Rechner arbeiten mit demselben Mikroprozessor Z80A, der mit 3,6MHz getaktet wird. So können alle Peripheriegeräte untereinander ausgetauscht werden, Programme laufen auf allen Computern gleich gut, und viele Hardware-Erweiterungen passen auch an MSX-Rechner anderer Hersteller. Die ersten Heimcomputer dieses Typs werden in Deutschland unter anderem von den Firmen Sony und Spektravideo angeboten.

?

■ Was ist eine Hardcopy-Routine?

Zu einer sorgfältigen Dokumentation von Programmen gehören Unterlagen, die zeigen, wie der Bildschirm während des Programmablaufs aufgebaut ist, und in welcher Form die Eingaben vom Benutzer abgefordert werden. Hardcopies drucken nach Belieben den aktuellen Bildschirminhalt aus, doch diese Programmroutinen sind nicht gerade einfach aufgebaut. Ein besonderes Problem besteht in der Notwendigkeit, den üblichen Programmablauf für den Aufruf der Hardcopy-Routine zu unterbrechen. Dadurch wird der ursprüngliche Bildschirminhalt nach oben gerollt und unter Umständen das zu dokumentierende Programm zerstört. Eine Lösung besteht für BASIC in einem Hardcopy-Programm, das in Maschinensprache geschrieben ist und zusammen mit dem BASIC-Programm im Hauptspeicher steht. Gute Hardcopy-Routinen lassen sich während des BASIC-Programmlaufes mit einer Kontroll-Taste starten, ohne den Lauf zu unterbrechen.

Hardcopy-Routinen sind in der Lage, den jeweiligen Inhalt des Bildschirms auszudrucken. In diesem Fall das Spiel „Pedro“. Doch die Programmroutinen sind nicht gerade einfach aufgebaut.



Bauen Sie den „Turm von Hanoi“!

Mehr über recursive Prozeduren, eigene Hilfsprogramme und die Bedingungsüberprüfung auf TRUE oder FALSE.

Ein gutes Beispiel für recursive Prozeduren ist ein Sortierprogramm, das Gegenstände nach unterschiedlichen vorgegebenen Kriterien ordnet. Die Bestandteile dieses „Puzzles“ setzen sich aus mehreren Scheiben verschiedener Größe zusammen. Diese sind zu Beginn auf einem der drei zur Verfügung stehenden Holzstifte aufgesetzt, und zwar so, daß die größte Scheibe unten liegt, darauf befindet sich die nächstkleinere usw. Die kleinste Scheibe liegt obenauf.

Das Programm, wir nennen es „Turm von Hanoi“, soll die Scheiben von einem Stift auf einen anderen „umschichten“. Um die korrekte Reihenfolge zu gewährleisten, können die Scheiben nur einzeln transportiert werden.

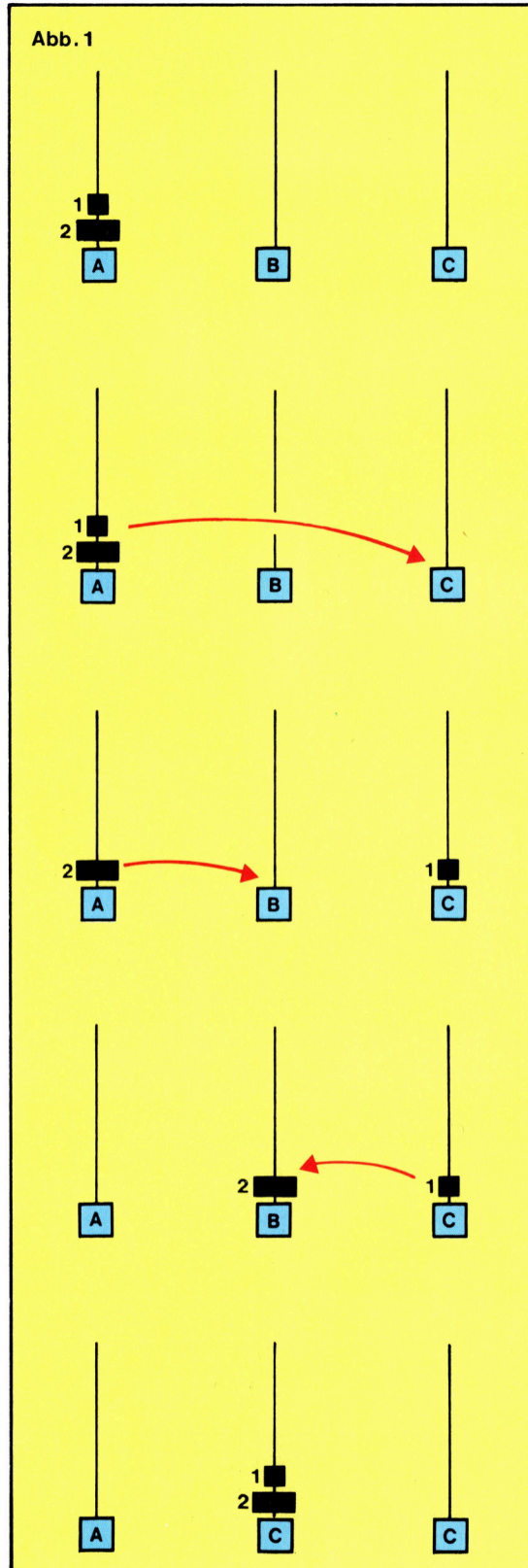
Bevor die Prozedur geschrieben wird, sollten die jeweiligen Schritte sorgfältig durchdacht und zur Veranschaulichung auf Papier gezeichnet werden. Die erste Abbildung zeigt den Sortiervorgang mit zwei Scheiben. Das erscheint noch relativ einfach. Die Aufgabenstellung sieht folgendermaßen aus: Die beiden Scheiben (1 und 2) sollen von Stift A auf B gesetzt werden, wobei die ursprüngliche Reihenfolge beibehalten werden muß. Um nun die größere Scheibe (2) als erstes Element auf Stift B legen zu können, muß zunächst Scheibe 1 entfernt, das heißt, auf Stift C gelegt werden. Danach wird zunächst Scheibe 2 und anschließend Scheibe 1 auf Stift B gesetzt. Fertig!

Das Prinzip dieses Sortiervorgangs sollte nun klar sein. Schwieriger wird es jedoch, wenn sechs Scheiben umgeschichtet und sortiert werden müssen (Abbildung 2). Dazu muß wiederum Scheibe 6 (die größte) von Stift A auf B gelegt werden. Was bedeutet, daß die ersten fünf Scheiben von Stift A auf C zu stecken sind. Diese Aufgabe wird später ein Unterprogramm in der Prozedur lösen. Es entsteht also ein neuer „Turm von Hanoi“ aus fünf Elementen. Mit der nachfolgenden recursive Prozedur lassen sich die Scheiben korrekt ordnen:

```

TO HANOI :NUMMER :VON :NACH :TEMP
  IF :NUMMER = 0 [STOP]
  HANOI :NUMMER - 1 :VON :TEMP :NACH

```



Die beiden Scheiben befinden sich zunächst auf Stift A. Es soll erreicht werden, daß die Scheiben wiederum nach Größe geordnet auf Stift B gesetzt werden.



```

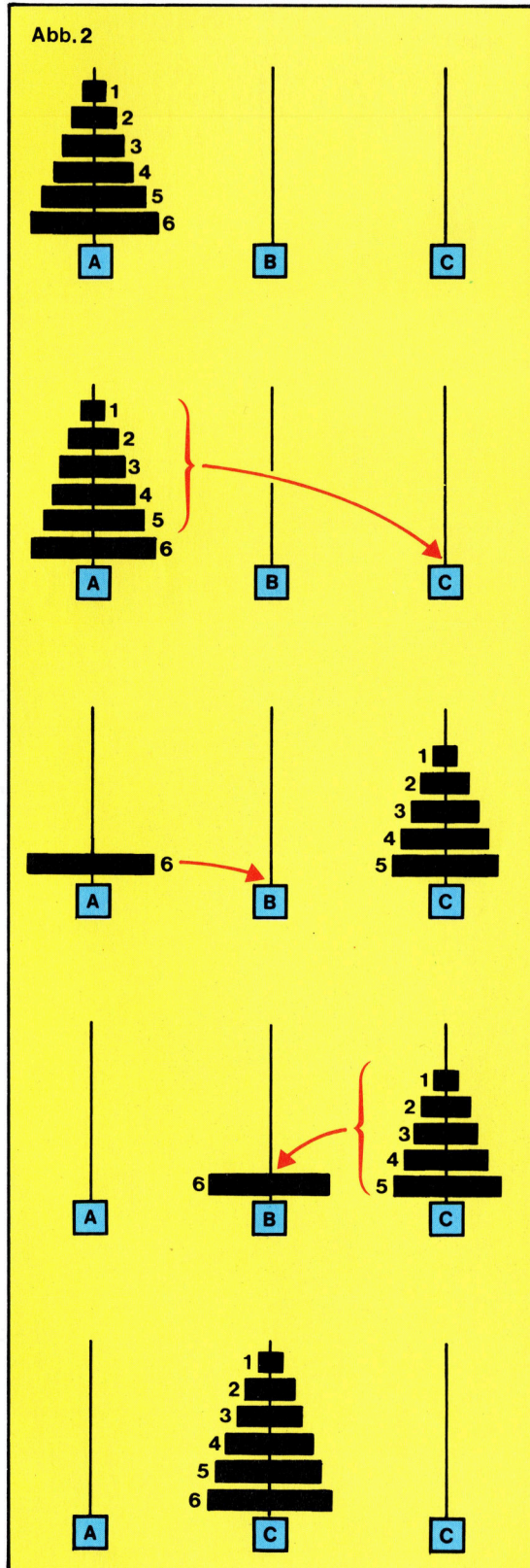
PR (SENTENCE [MOVE DISK] :NUMMER [VON
STIFT] :VON [NACH STIFT] :NACH)
HANOI :NUMMER — 1 :TEMP :NACH :VON
END

HANOI 6 "A "B "C

```

Bei diesem Programm müssen vier Eingaben eingetippt werden. Mit der Variablen :NUMMER wird festgelegt, wieviel Elemente die Prozedur sortieren soll. Die Angaben "A, "B und "C bezeichnen die drei Holzstifte, auf die die geordneten Scheiben aufgesteckt werden.

Jetzt wird's schon schwieriger. In diesem Beispiel sind sechs Scheiben geordnet auf einen anderen Stift zu legen.



Drei Variationen

Zur Übung sollten Sie nun versuchen, diese Problemstellung in einer anderen Programmiersprache zu lösen. Sie werden sehen, daß das Programm erheblich länger wird, als die LOGO-Version. Das nächste Beispiel für recursive Prozeduren entstand aus der Zusammenarbeit einiger amerikanischer High-School-Studenten und LOGO-User-Gruppen. Die Überlegung dabei war, Geometrie anhand von selbständigen Experimenten und Entwicklungen zu vermitteln. Dabei sollten natürlich auch die Vorurteile derjenigen abgebaut werden, die in Turtle Graphics nichts anderes als ein nettes Spielzeug sehen. Die folgende Prozedur gibt eine vorgegebene Zeichnung in drei verschiedenen Variationen aus:

```

TO VERZWEIGEN :LAENGE :STUFE
IF :STUFE = 0 [STOP]
FORWARD :LAENGE
LEFT 45
VERZWEIGEN :LAENGE / 2 :STUFE — 1
RIGHT 90
VERZWEIGEN :LAENGE / 2 :STUFE — 1
LEFT 45
BACK :LAENGE
END

```

Die Befehle, die in diesem Programm verwendet werden, sind bereits vorher ausführlich besprochen worden. Zur Erinnerung noch einmal der Programmablauf in Kurzform. Die Variable :LAENGE steht für die Strecke, die bei jedem Programmdurchgang von der Turtle zurückgelegt wird, und :STUFE repräsentiert den Wert, der angibt, wie oft die recursive Prozedur abgerufen werden soll.

Ändern Sie nun das Programm so, daß auch verschiedene Variablenwerte und andere Bewegungsabläufe als Eingaben möglich sind. Oder Sie versuchen, eine Prozedur – unter Verwendung der oben gezeigten – zu schreiben, die mehrere Bäume auf den Bildschirm zeichnet. Abbildung 3 zeigt ein Beispiel, wie dieser Baum aussehen könnte.

Hilfsprogramme

Wie bereits erwähnt, bietet LOGO die Möglichkeit, eigene, vielseitig einsetzbare Unterprogramme zu definieren, die je nach Wunsch in die Prozeduren einfügt oder von ihnen aufgerufen werden können. Auf diese Weise lassen sich auch die Möglichkeiten, die Befehle wie IF, THEN oder ELSE bieten, erheblich er-



weitem. Die folgenden Zeilen zeigen Beispiele für selbstdefinierte Hilfsprogramme, in denen ebenfalls Bedingungen abgefragt werden.

Die folgende recursive Prozedur überprüft eine Bedingung und ruft sich selbst solange auf, bis das Ergebnis des Testes „FALSE“ ergibt.

```
TO WAHREND :BEDINGUNG :AKTION
  IF RUN :BEDINGUNG [RUN :AKTION] [STOP]
  WAHREND :BEDINGUNG :AKTION
END
```

Durch den Befehl RUN kann man die Anweisungen, die in der Liste stehen, aufrufen. Wie sich diese Prozedur in Verbindung mit einer neuen anwenden läßt, demonstrieren die nächsten Zeilen:

```
TO ZAEHLEN
  MAKE "N 1
  WAHREND [ :N < 20] [PRINT :N MAKE
    "N :N + 1]
END
```

Diese Prozedur „zählt“ von 1 bis 20 und gibt die jeweiligen Zahlen auf dem Bildschirm aus.

Die nächste selbstdefinierte Bedingungsüberprüfung nennen wir „VON . . . BIS“. Die Prozedur ruft sich auf, bis das Ergebnis der Prüfung, das je nach Vorgabe FALSE oder TRUE lauten soll, korrekt ist.

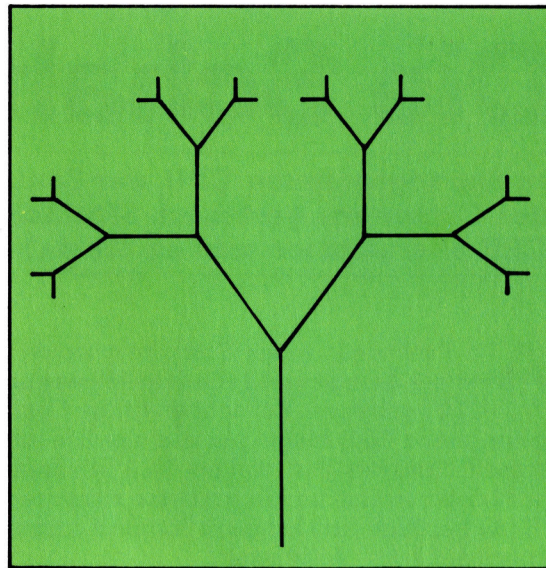
```
TO VON :AKTION :BIS :BEDINGUNG
  RUN :AKTION
  IF BIS :BEDINGUNG [STOP]
  VON :AKTION :BIS :BEDINGUNG
END
```

```
TO BIS :BEDINGUNG
  IF RUN :BEDINGUNG [OP "TRUE] [OP "FALSE]
END
```

Die Prozedur VON beinhaltet als Parameter drei eigenständige Listen. Geben Sie nun das folgende Programm ein:

```
TO VERSUCH
  MAKE "N 1
  VON [PRINT :N MAKE "N :N + 1] [BIS]
    [ :N > - 1]
END
```

Die Variable :N wird mit einem Wert initialisiert, der größer ist als derjenige, auf den die Variable überprüft werden soll. Die Prozedur :AKTION wird vor der :BEDINGUNG aufgerufen. Das Programm :BIS dagegen beinhaltet nur eine Liste von Parametern, die die Eingaben auf TRUE oder FALSE überprüfen. Versuchen Sie nun, diese Prozeduren in eigene, von Ihnen selbst entwickelte Programme einzubauen und die Werte entsprechend Ihren Vorgaben zu verändern.



Schreiben Sie eine Prozedur, die unter Verwendung der Recursion mehrere Bäume auf den Bildschirm zeichnet.

Spiel mit Linien

Zur Entspannung hier noch ein kleines Programm, in dem Sie Ihre Kenntnisse der Turtle Graphics erproben können:

```
TO BILD1 :A :B :X1 :Y1 :X2 :Y2
  IF :A = 0 THEN STOP PENUP
  SETXY :X1 :Y1 PENDOWN
  SETXY :X2 :Y2
  MAKE "X1 :X1 + :B
  MAKE "Y2 :Y2 + :B
  BILD1 :A - 1 :B :X1 :Y1 :X2 :Y2
END
```

```
TO BILD2 :A :B :X1 :Y1 :X2 :Y2
  IF :A = 0 THEN STOP PENUP
  SETXY :X1 :Y1 PENDOWN
  SETXY :X2 :Y2
  MAKE "X1 :X1 - :B
  MAKE "Y2 :Y2 + :B
  BILD2 :A - 1 :B :X1 :Y1 :X2 :Y2
END
```

```
TO DEMO1
  BILD1 25 10 (-158) (-120) 158 (-120)
END
```

```
TO DEMO2
  BILD2 25 10 158 (-120) (-158) (-120)
END
```

```
TO ZEICHNUNG
  FULLSCREEN DEMO1 DEMO2
END
```

Bei dieser Prozedur stellen X1 und Y1 die Anfangs-, X2 und Y2 die Endkoordinaten der einzelnen Linien dar. Der Abstand zwischen den Linien ist in der Variablen B und die Anzahl der zu zeichnenden Linien in der Variablen A abgelegt. Versuchen Sie nun durch Änderung der Werte neue Zeichnungen zu erstellen.



Die Schaltzentrale

In der sogenannten CPU, der Zentraleinheit, werden alle Funktionen des Computers gesteuert. Sämtliche Arbeitsprozesse werden von der CPU übernommen und an die dafür vorgesehenen Chips verteilt.

Alle Funktionen eines Computers werden von der Zentraleinheit (Central Processing Unit—CPU) gesteuert. Ein sehr einfacher Computer (siehe Bild) müßte nur aus der Zentraleinheit, Speicher und einigen E/A-Schaltungen bestehen. Ein- und Ausgabemechanismen (E/A) benötigt der Computer für die Kommunikation mit der „Außenwelt“.

Ein Speicher wird für die Bereitstellung von Instruktionen und Daten für die Zentraleinheit gebraucht. Die zu bearbeitenden Daten können dabei sowohl Zahlen als auch binären Code enthalten, der eine Vielfalt von Zeichen darstellen kann.

Somit enthalten einige der Speicherstellen Befehle, andere dagegen Daten, die die Zentraleinheit bearbeiten soll. Wie kann die Zentraleinheit zwischen diesen beiden verschiedenen Speicherinhalten unterscheiden?

Zentraleinheiten der 8-Bit-Prozessoren (die Steuereinheit fast aller Heimcomputer) werden normalerweise in Form eines einzigen Chips gebaut, bei dem sich an beiden Längsseiten 20 Kontaktstifte befinden. Über jeden dieser Kontaktstifte (abgesehen von den zwei Stiften für die Stromversorgung von 0 und +5 Volt) laufen Impulse, die von Peripheriegeräten wie E/A-Platinen oder von dem Speicher übermittelt oder dorthin gesendet werden.

Typische Zentraleinheiten verfügen über 16 Adreßkontakte, die mit dem Adreß-Bus ver-

bunden sind. Jeder dieser Kontaktstifte übermittelt ein Ausgabesignal, das eine Null oder eine Eins darstellen kann. Die sechzehn Adreßleitungen können also 65536 Kombinationen dieser beiden Zeichen darstellen und damit die gleiche Anzahl von Speicherstellen individuell ansprechen.

Weiterhin gibt es acht Datenkontakte, die mit dem Daten-Bus verbunden sind. Diese Kontakte übermitteln von einer Ein- und Ausgabeeinheit oder von dem Speicher Daten an die Zentraleinheit oder empfangen sie von dort. Weitere Kontakte übermitteln Kontrollsignale, die entweder von der Zentraleinheit erzeugt oder dorthin gesendet werden.

Universalregister

Innerhalb der Zentraleinheit gibt es einige Spezielspeicher mit einem oder zwei Byte Kapazität, sogenannte Register. Die anderen Speicher innerhalb der Zentraleinheit werden Universalregister genannt und für die Zwischenspeicherung von Informationen verwendet. Weiterhin gibt es noch zwei Funktionsmodule: die ALU und den Kontroll-Block.

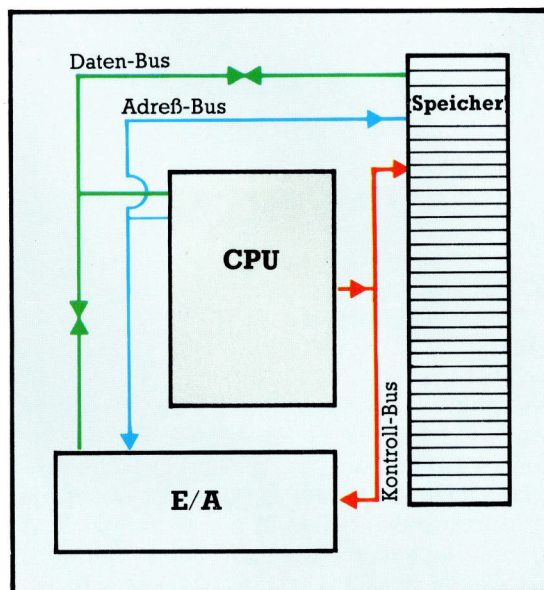
Die Abkürzung ALU steht für Arithmetik-Logik-Einheit (englisch: Arithmetic and Logic Unit). Dieser Teil der Zentraleinheit führt die mathematischen und logischen Vorgänge aus, wie Additionen, das Rotieren von Bits innerhalb eines Bytes und die Abläufe bei den logischen Operatoren AND und OR.

Der Kontroll-Block ist eine spezielle Schaltung, die die Impulsfolgen des Systems koordiniert und synchronisiert. Die Funktion wird an dem Beispiel der weitverbreiteten Z80-Zentraleinheit erläutert.

Wird von der Zentraleinheit der Befehl 11000110 aus dem Speicher empfangen, so addiert sie den Inhalt des nächsten Bytes im Speicher auf den Inhalt eines ihrer internen Register. Wenn das Resultat an einer bestimmten Speicherstelle abgelegt werden soll, muß der nächste Befehl an die Zentraleinheit die Zahlenfolge 00110010 sein, gefolgt von zwei Bytes, die die Speicherstelle bezeichnen.

Angenommen, das Berechnungsergebnis ist die Dezimalzahl 37 und die zwei Bytes hinter dem Befehl bezeichnen die Speicherstelle 33126 (dezimal). Der Befehl veranlaßt jetzt den Kontroll-Block, die Adreßkontakte auf die binäre Entsprechung von 33126 (d. h.

Ein sehr einfacher Computer (siehe Bild) braucht eigentlich nur aus einer Zentraleinheit, Speicher und einer E/A-Schaltung zu bestehen. In dem Speicher sind Befehlsfolgen enthalten, die über die Zentraleinheit bestimmte Abläufe auslösen. Der Speicher kann ebenfalls Daten unterbringen, die von der Zentraleinheit später benötigt werden. Die E/A-Schaltung wird zur Kommunikation mit der Außenwelt benötigt. Steuert ein Computer beispielsweise eine Waschmaschine, dann erhält er seine Eingabesignale über die Knöpfe der Frontplatte und seine Ausgabesignale schalten Motoren und Heizung aus und ein. Die internen Befehlsfolgen der Zentraleinheit sind binär codiert. Die verschiedenen Typen von Zentraleinheiten arbeiten alle mit unterschiedlichen Befehlssätzen.



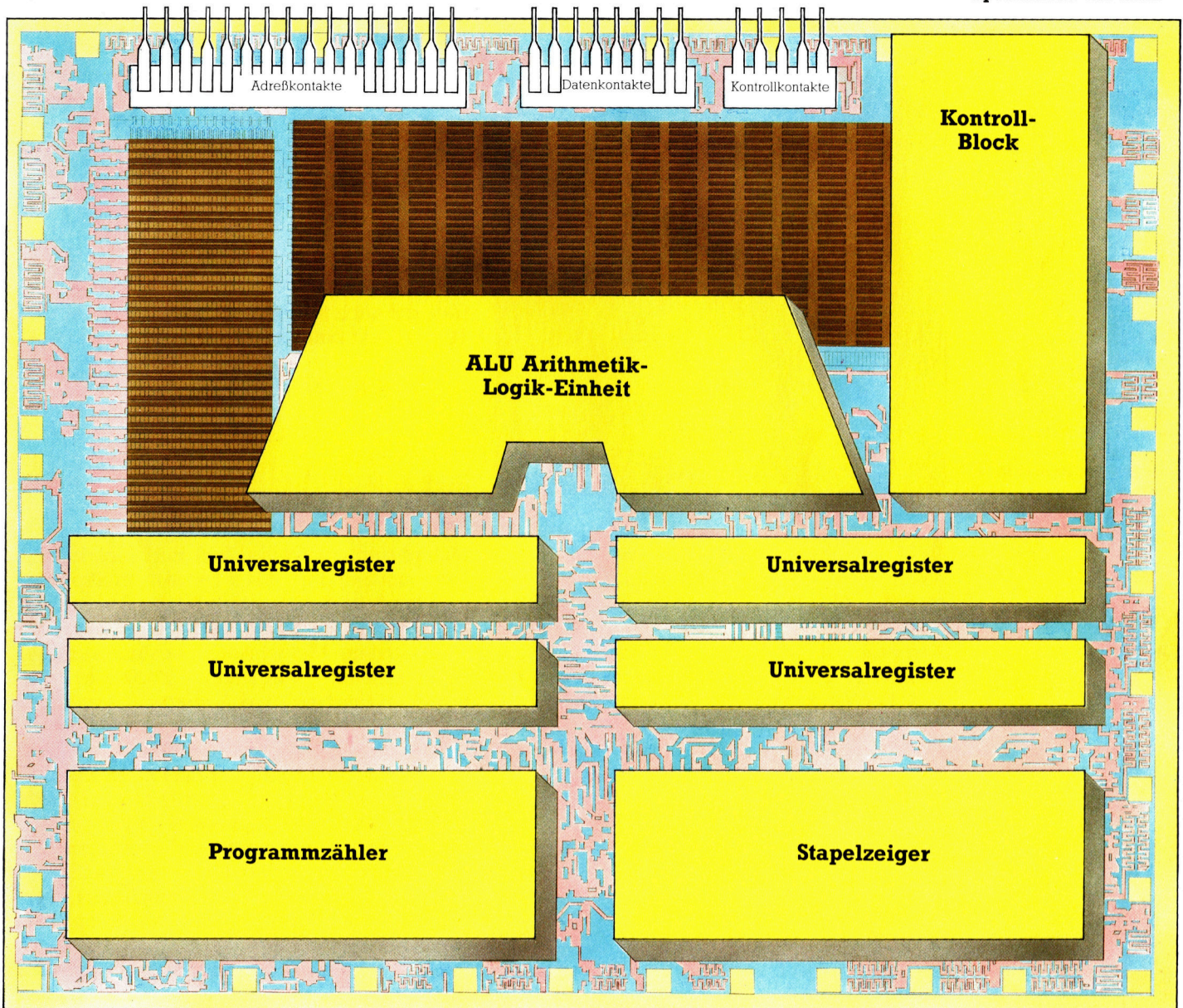


1000000101100110) zu setzen. Weiterhin wird dem Speicher über den Kontroll-Bus das Signal gesendet, daß in Kürze Daten ankommen werden, die gespeichert werden sollen. Über den Daten-Bus werden die Daten dann an den Speicher gesandt und auf der Speicherstelle abgelegt, die von dem Adreß-Bus übermittelt wurde. Werden diese Daten zu einem späteren Zeitpunkt von der Zentraleinheit (z. B. für

haben normalerweise Namen wie Stack Pointer, Program Counter oder Akkumulator. Universalregister werden als X-Register, Y-Register oder C-Register bezeichnet.

Eines der wichtigsten Spezialregister ist der Program Counter, der Programmzähler. Diese interne Speicherzelle enthält (in Binärzahlen) immer die Adresse des nächsten Befehls, der im Speicher zur Ausführung bereitliegt. Will

Die Abbildung zeigt eine Zentraleinheit mit ihren Registern, einer Arithmetik-Logik-Einheit, die aus Hunderten von logischen Gattern besteht (mit denen Rechenoperationen wie Addition, logische Operationen wie AND



ihre Darstellung auf dem Bildschirm) benötigt, müßte ein anderer Befehl an die Zentraleinheit übermittelt werden. In Kurzform würde der Kontroll-Block diesen Befehl als „von der Speicherstelle 33126 ein Byte holen und in einem der internen Register zwischenspeichern“ verstehen.

Die Anzahl der internen Register oder Zwischenspeicher einer Zentraleinheit hängt von ihrem Typ ab. Es gibt 8-Bit- (ein Byte) oder 16-Bit- (zwei Byte) Register. Die Spezialregister

die Zentraleinheit den nächsten Befehl ausführen, wird der Inhalt des Programzählers über den Adreß-Bus an den Speicher ausgegeben und das an der entsprechenden Stelle gespeicherte Byte über den Daten-Bus der Zentraleinheit übermittelt. Ein wichtiger Teil der 8-Bit-Register ist der Akkumulator. Der Akkumulator ist ein Zwischenspeicher für von der ALU erzeugte Ergebnisse oder für Daten, die vom Speicher oder den E/A-Bausteinen übermittelt wurden oder dorthin gesendet werden sollen.

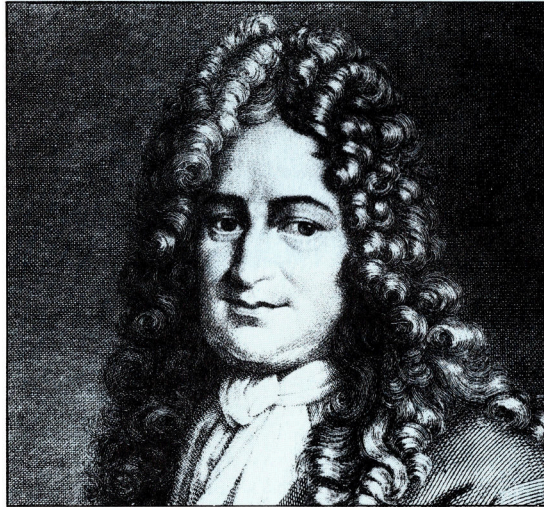
und die Herstellung des Komplements von Binärzahlen ausgeführt werden) und dem Kontroll-Block. Der Kontroll-Block nimmt die (binär) codierten Befehle entgegen, interpretiert sie und steuert dann die Abläufe innerhalb der Zentraleinheit.



Multiplizieren im Binärsystem

Computer multiplizieren auch die größten Zahlen durch einfaches Zusammenzählen von Binärziffern.

Gottfried Wilhelm Leibnitz (1646–1716) war ein Zeitgenosse von Isaac Newton und befaßte sich mit Philosophie, Mathematik und Technik. Er untersuchte die Möglichkeiten der Anwendung binärer Arithmetik in Vorrichtungen zum Rechnen und konstruierte eine Maschine zum Multiplizieren und Dividieren. In späteren Jahren diskutierte Leibnitz mit Newton über Themen der höheren Mathematik.



Genau dieses Verfahren wird auch im binären Zahlensystem angewendet:

$$\begin{array}{r} 101 \times 11 \\ 101 \\ + 101 \\ \hline 1111 \end{array}$$

Dies gilt auch für größere Zahlen. Das Beispiel 14×12 sieht binär so aus:

$$\begin{array}{r} 1110 \times 1100 \quad (14 \times 12) \\ 1110 \\ 1110 \\ 0000 \\ + 0000 \\ \hline 10101000 \quad (168) \end{array}$$

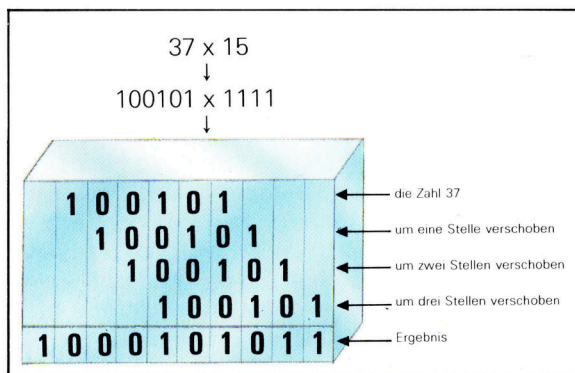
Will man 14 mit 12 multiplizieren, wäre der einfachste Weg, die 14 zwölfmal untereinander zu schreiben und dann zu addieren, also $14 + 14 + 14 + \dots$ (zwölfmal). Da die Multiplikation in gewisser Weise ein wiederholtes Addieren ist, würde man mit dieser Vorgehensweise sicher zum Ziel kommen. Die ersten Rechenmaschinen haben tatsächlich auf diese Weise multipliziert. Dieses Verfahren ist jedoch mühsam und zeitaufwendig. Deshalb wurde eine bessere Methode entwickelt.

Das Multiplizieren zweier Zahlen schreibt man üblicherweise so:

$$\begin{array}{r} 14 \times 12 \\ 14 \\ + 28 \\ \hline 168 \end{array}$$

(Oft wird eine freie Stelle auch mit einer 0 aufgefüllt.)

In der Darstellung wird 100101 (37) mit 1111 (15) multipliziert – die Zahl 100101 erscheint viermal als „Kopie“, wobei jede „Kopie“ entsprechend der Position der multiplizierenden 1 nach rechts verschoben ist. Zum Schluß werden alle „Kopien“ addiert und ergeben als Resultat 1000101011, dezimal die Zahl 555.



Multipliziert man eine Zahl mit 1, bleibt die Zahl unverändert ($14 \times 1 = 14$), multipliziert man mit 0, ist das Ergebnis 0 ($14 \times 0 = 0$). Dies gilt in allen Zahlensystemen.

Das binäre Multiplizieren besteht aus drei schematischen Vorgängen: Kopieren, Verschieben (shifting) und Addieren. Genau dies ist auch die Arbeitsweise des Computers. Zuerst wird die zu multiplizierende Zahl entsprechend der Anzahl der Einsen im Multiplikator kopiert. Steht im Multiplikator eine 0, entsteht anstelle einer Kopie eine Reihe von Nullen. Im nächsten Schritt werden Kopien und Null-Reihen untereinander gesetzt und entsprechend der Stellen des Multiplikators verschoben. Dann werden die Zahlen spaltenweise addiert.

Ein Computer braucht zum Multiplizieren ein riesiges Aufnahmevermögen für die Stellen. Allgemein kann gesagt werden, daß das Ergebnis einer Multiplikation die doppelte Stellenzahl der größeren Zahl haben kann.

Multiplikationsergebnisse von Computern können aber auch falsch sein – eine Feststellung, die manchen überraschen mag. Die Ursache dafür ist in fast allen Fällen auf eine unzureichende Stellenzahl für die Ergebnisspeicherung zurückzuführen. Sie wird von der Hardware bestimmt. Bei zu wenigen Stellen tritt ein sogenannter „Überlauf“ (overflow) auf. Die letzten Stellen gehen verloren, und das Ergebnis wird falsch.

Fachwörter auf einen Blick

CAM

(Computer Aided Manufacturing) – Programme zur Steuerung numerisch gestützter Werkzeugmaschinen

DMA

(Direct Memory Access) – direkter Speicherzugriff auf Daten mittels eines besonderen Steuerelements ohne Einschaltung der CPU

dongle

Modul zum Schutz gegen unbefugtes Kopieren von Programmen

festverdrahtet

Logische Funktionen werden durch festverdrahtete Schaltungen verwirklicht (zum Beispiel Gatter)

Helligkeit

Kann vom Computer-Bediener am Bildschirm verändert werden; beeinflusst die Lesbarkeit der dargestellten Zeichen

Hertz

(Hz) Maßeinheit für die Frequenz; wurde nach dem Physiker Heinrich Hertz benannt

höchstwertiges Bit

Stelle mit dem höchsten Wert eines Bits; zum Beispiel bei binär 1000 ist 1 das Bit mit dem höchsten Wert

Hostrechner

Bezeichnung für eine große Datenverarbeitungsanlage, mit der auch kleinere Computersysteme verbunden werden können

Informatik

Naturwissenschaftlicher Studiengang; Informatik bezeichnet die Lehre der Datenverarbeitung und deren Anwendung

Interpreter

Programm des Betriebssystems zur Abarbeitung von Programmen höherer Programmiersprachen

LAN

(Local Area Network) – Netzwerk von Computersystemen in einem begrenzten lokalen Bereich

Makrobefehl

Unter einem bestimmten Namen wird eine Gruppe von Befehlen aufgerufen; ein sogenannter Makroassemblierer sorgt für die Übersetzung der Namen in Befehle

manuell

mit der Hand betätigt

Multiplex

Datenübertragungsform, wobei verschiedene Sender einen gemeinsamen Übertragungsweg benutzen; anschließend werden die Daten entsprechend ihrer Zugehörigkeit wieder aufgeteilt

Netzwerk

(network) – Zusammenschluß von Rechnern zu einem Datenkommunikationsnetz

Objektcode

Durch Compiler in Maschinencode übersetzte Programme

Optischer Markierungsleser

Peripheriegeräte, die Markierungen (Bleistift, Drucke oder Lochungen) abtasten und diese in Eingabeimpulse umwandeln



Quellcode

(Source code) – in einer Programmiersprache abgefaßtes Programm

Quellprogramm

(Source program) – ursprüngliche Fassung von Programmen, die für den Gebrauch in Maschinensprache kompiliert werden müssen

Sonderzeichen

Zeichen und Symbole, die neben Buchstaben und Zahlen im Zeichensatz vorhanden sind

Stopplänge

Bezeichnet die Länge eines Bandes, das am Schreib/Lese-Kopf nach dem abschließenden Schreib- oder Lesevorgang bis zum endgültigen Stopp entlangläuft

Terminal-Interface

Anschlußgerät für Terminals zur Anpassung von Geschwindigkeit und Spannungspegel zum Rechner beim Datentransfer

Transferrate

Übertragungsgeschwindigkeit für Daten

Überschreiben

Eingabe von Daten in den Speicher, wobei die bereits abgespeicherten Daten gelöscht und zerstört werden

verschieben

(shift) – Verschieben binärer Daten in einem Datenregister

Winchesterplatte

Festplatten-Speicher mit hoher Informationsdichte und kurzer Zugriffszeit

Zustandsbit

Bit, das vom Computer auf 0 oder 1 gesetzt wird, um einen bestimmten Zustand anzuzeigen; zum Beispiel um festzuhalten, ob ein Zusatzgerät angeschlossen ist (busy) oder nicht

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

computer kurs Heft 9

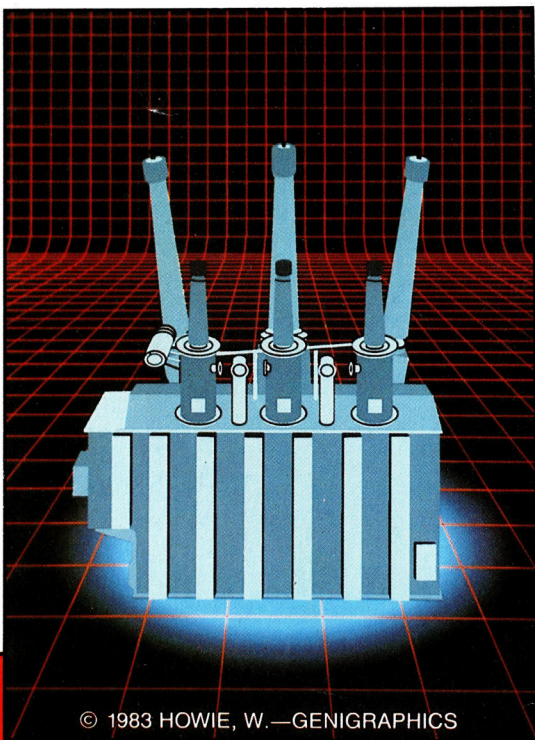
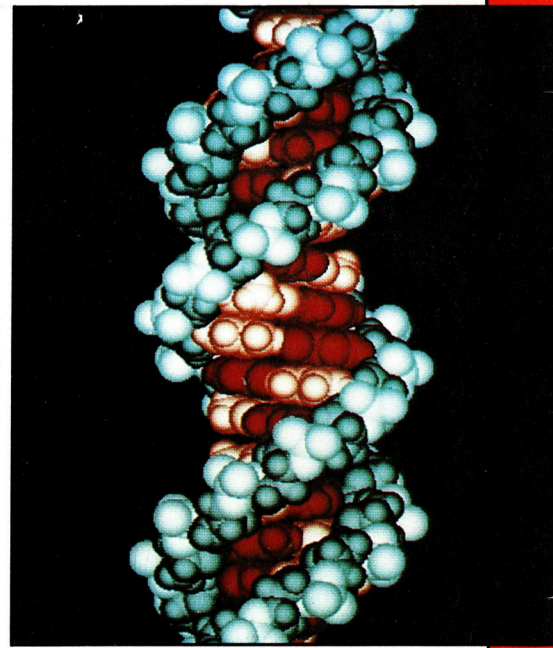


Sord M 5
Grafiken sind die Stärke
dieses japanischen
Heimcomputers.



Speichertechnik
Von der Elektronenröhre
bis zum Biospeicher
von morgen.

Bewegte Bilder
Computer eröffnen
neue Dimensionen
in der
Trickfilmtechnik.



+++ **Grafiktablets** +++ **3-D-Effekte mit dem Heimcomputer** +++ **Dateien im Griff** +++ **Tips und Tricks für Grafik- und Sound-erzeugung** +++ **Mathematische Grundlagen der Computerlogik** +++ **PEEK und POKE**