

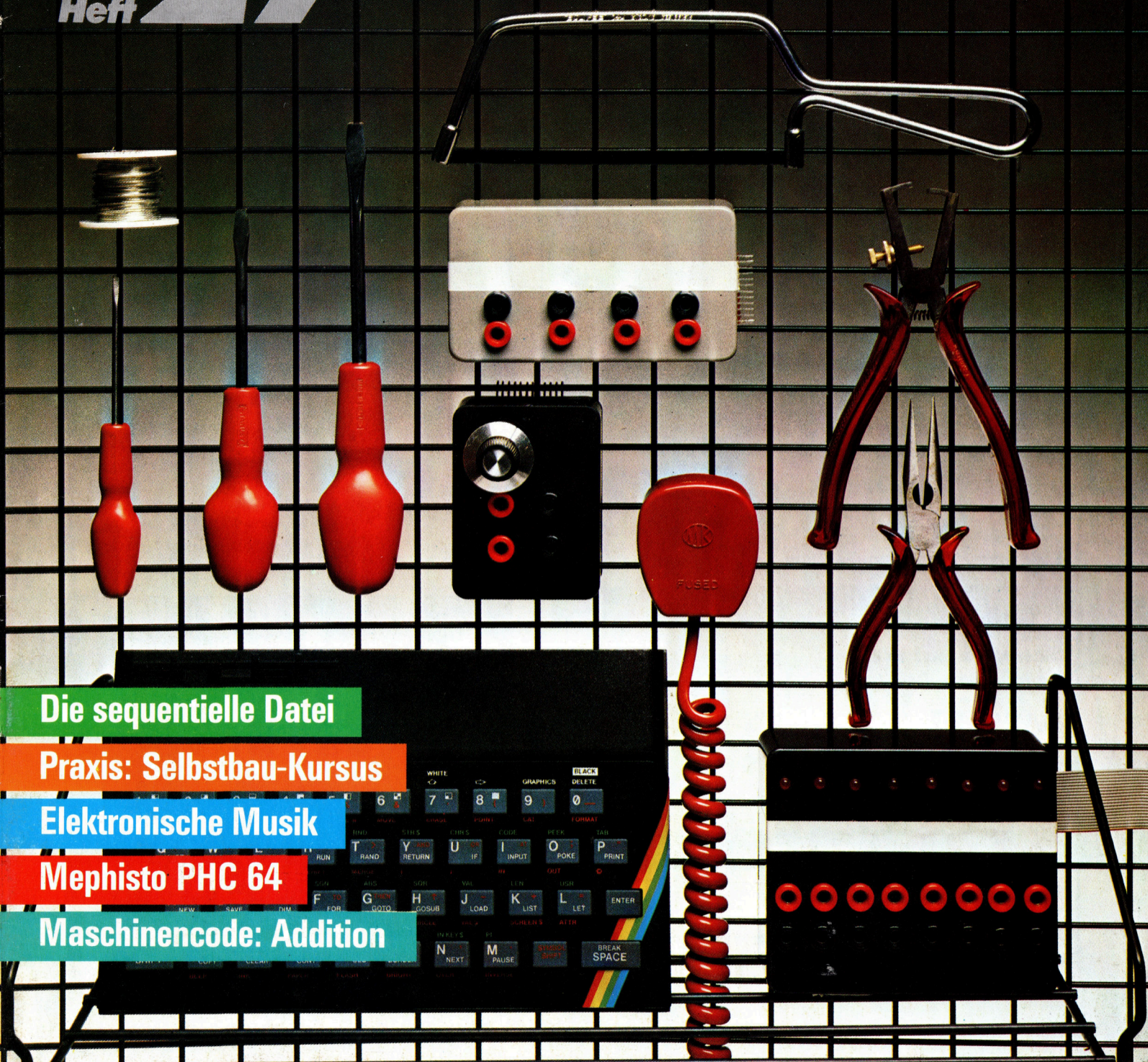
Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Ein wöchentliches Sammelwerk

Heft **27**



Die sequentielle Datei

Praxis: Selbstbau-Kursus

Elektronische Musik

Mephisto PHC 64

Maschinencode: Addition

computer kurs

Heft 27

Inhalt

Computer Welt



Großer Schlußakkord 729

Modernere Systeme für elektronische Musik

Über die Sprache 747

Roboter verstehen und formulieren selbst

Spiel um Millionen 755

Aufstieg und Fall von Atari

Software



Serieller Zugriff 733

Der Einsatz sequentieller Dateien

Sommerspiele 754

„Summer Games“ aus dem Hause Epyx

Hardware



Der Mephisto PHC 64 735

Ein extrem erweiterbarer Heimcomputer

LOGO 27



LOGO-Geometrie 738

Das Zeichnen von Zykloiden

Tips für die Praxis



Wiederholung 740

Werkzeug, Techniken, Bauteile

Peripherie



Schwarz auf Weiß 742

Anschlagfreie Drucker

BASIC 27



Das ABC des Acorn B 745

Einfaches Aufrufen der Unterroutinen

Bits und Bytes



Flagge zeigen 750

Die Statusregister bei der Addition

Fachwörter von A—Z

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 wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

Abonnement

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

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

SAMMELORDNER

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

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

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

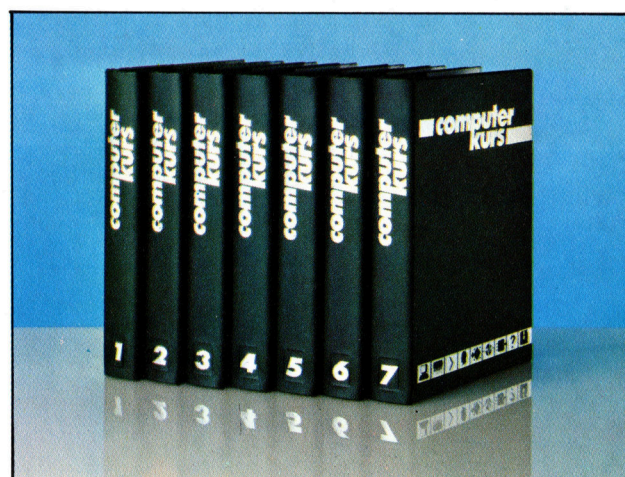
Schweiz: Der Sammelordner kostet sfr 15. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

INHALTSVERZEICHNIS

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

Redaktion: Winfried Schmidt (verantw. f. d. Inhalt), Elke Leibinger, Susanne Brandt, Uta Brandl (Layout), Sammelwerk RedaktionsService GmbH, Paulstraße 3, 2000 Hamburg 1

Vertrieb: Marshall Cavendish International Ltd., Heidenkampsweg 74, 2000 Hamburg 1, Tel.: 040/23 40 85



© APSIF, Copenhagen, 1982, 1983; © Orbis Publishing Ltd., 1982, 1983; © Marshall Cavendish Ltd., 1984, 1985; **Druck:** E. Schwend GmbH, Schmolterstraße 31, 7170 Schwäbisch Hall



Großer Schlußakkord

Im letzten Teil dieser Serie beschäftigen wir uns mit einigen der fortgeschrittenen Systeme, die seit Einzug der digitalen Technik in die elektronische Musik auf den Markt gelangt sind.

Die wohl bedeutendste jüngere Entwicklung elektronischer Musik fand im Bereich der digitalen Aufnahmetechnik statt. Nicht nur, daß die Tonqualität enorm verbessert wurde, sondern das Wort „Aufnahme“ hat aufgrund der verbesserten Technik eine völlig neue Bedeutung erfahren.

Seit dem Zweiten Weltkrieg waren Tonaufnahmen auf Magnetband die Norm, wobei diese von der winzigsten Microcassette bis hin zum großspuligen 24-Spur-Band in professionellen Aufnahmestudios reichte. Bei der Magnet-Tonaufnahme werden die winzigen Partikel des Metalloxyds auf der Bandoberfläche analog zur Klangwellenform, die sie darstellen, in komplexen Mustern geordnet. Beim Vorbeilaufen des Magnetbandes am Wiedergabekopf werden diese Muster in Folgen elektrischer Spannungen umgewandelt. Die jeweilige Spannung wird dann an die Lautsprecher weitergeleitet, wo sie den aufgezeichneten Klang reproduziert.

Bei der digitalen Aufnahme wird das Klangeignis numerisch aufs Band codiert. Der Wiedergabekopf ist ein Digital/Analog-Umwandler. Die Lautsprecher werden auf dieselbe Art wie zuvor betrieben, doch die in sie geleiteten Spannungen werden vom D/A-Wandler erzeugt. Genügend Daten für diese Umwandlung vorausgesetzt, sind mit dem digitalen Band Klangreproduktionen möglich, die die auf Magnetband bei weitem übertreffen. Das Band kann hundertfach ohne Qualitätsverlust digital kopiert werden. Beim Magnetband dagegen leidet die Qualität mit jedem neuen Kopiervorgang.

Dieses Problem der Störgeräusche ist Ton-technikern seit langem bekannt. Seit dessen Lösung findet man in den großen Aufnahmestudios 24-Spur-Digital-Tonbandmaschinen. Durch ihre Verwendung ist eine so perfekte Klangreproduktion möglich, daß selbst ein Toningenieur nicht mehr feststellen kann, ob das Klangeignis von einem Musiker im Studio erzeugt wurde oder ob es sich um eine Wiedergabe einer digitalen Bandaufnahme handelt. Doch diese Technik brachte neue Pro-

Von New England Digital entwickelt, gilt dieses Instrument als eines der fortgeschrittensten der Welt. Neben den üblichen Synthesizer-Funktionen verfügt der Rechner über eine Klangspeicherkapazität von bis zu 10 Megabyte.

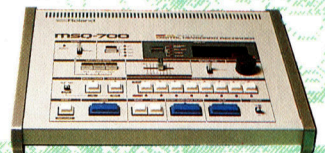


Der Fairlight CMI war eines der ersten Computer-Musiksysteme. Das menügesteuerte Betriebssystem erlaubt eine Fülle von Möglichkeiten, die von der Tastatursteuerungskontrolle bis zur Wellenform-Darstellung reichen. Ferner sind Ausdrücke der komponierten Stücke möglich.

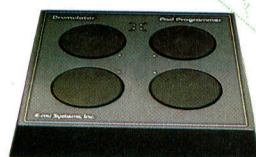


Der Roland MSQ-700 gilt als weiterer MIDI-kompatibler Sequenzer. Er verfügt über sämtliche MIDI-Eigenschaften und kann bis zu 6500 Noten speichern.

Das Yamaha KX5-MIDI-Interface stellt eine Verbindung zwischen Synthesizern und dem Yamaha CX5 her. Seit Einführung der MSX-Computer dient dieses Gerät auch als musikalische Schnittstelle zu Heimcomputern.



Die Drumulator-Schlagzeugmaschine von Emu verfügt über eine Speicherkapazität von 10 088 Noten bei 64 Stücken. Ferner gibt es Erweiterungsmöglichkeiten durch Einstekken von ROMs, mit denen z. B. Effekte wie lateinamerikanische oder afrikanische Rhythmen erzielt werden können.





bleme: Es ist nicht mehr möglich, zu „sehen“, an welcher Stelle des digitalen Bandes sich Klänge befinden. Ein anderes Problem stellen die „Studiogeräusche“ dar. „Studio-Noise“ ist ein unerwünschter und gewöhnlich auch nicht wahrnehmbarer Effekt mancher im Studio verwendeter Geräte. Magnetbänder waren zu unempfindlich, um das registrieren zu können.

Doch bei digitaler Aufnahme werden diese Geräusche hörbar.

Wenngleich die 24-Spur-Aufnahme in Digitaltechnik derzeit nur ganz großen Studios vorbehalten ist, kann eine einspurige digitale Aufzeichnung derselben Qualität von jedem Besitzer eines Betamax-Heimvideorecorders erzielt werden. Videoband ist ein digitales Trägermedium und kann folglich zur Aufzeichnung jeder Art von Daten eingesetzt werden. Mit dem Sony PCM (Pulse Code Modulator) läßt sich jeder Betamax-Videospieler in ein Audio-Aufnahmegerät umwandeln. Dieses Gerät bietet Möglichkeiten, die die eines analogen Aufnahmegerätes übertreffen.

Digitale Soundcodierung, auch „sampling“ genannt, ist Herzstück und Basis des Fairlight CMI (Computer Musical Instrument), einem der besten neuen Systeme. Der Fairlight kann jeden beliebigen Klang von bis zu zwei Sekunden Dauer erfassen und ihn dann über eine Breite von sechs Oktaven reproduzieren. „Sampling“ ist der eigentliche Durchbruch in der elektronischen Musik. Jahrelang haben Toningenieure wie Musiker versucht, den Klang von Streich- oder Holzblasinstrumenten mit Hilfe von Synthesizern zu simulieren. In einigen Fällen wurden mit dieser Technik hervorragende Ergebnisse erzielt. Doch mit dem „Sampling“ ist nicht nur eine perfekte Reproduktion des Klangs von „Streichern“ möglich, sondern auch die Klangreproduktion einer ganz bestimmten Geige. Und mehr: In einigen Fällen läßt sich der Klang eines bestimmten Spielers in einem bestimmten Raum realistisch wiedergeben.

Ein „Sampling“-Instrument wie der Fairlight übertrifft die natürlichen Möglichkeiten eines Musikinstrumentes bei weitem. Für einen Flötisten ist es leicht, einen warmen, sanften Ton in der unteren Stimmlage der Flöte zu erzielen. Doch selbst dem erfahrensten Spieler ist es unmöglich, denselben Ton zwei Oktaven höher zu blasen, da die Konstruktion der Flöte dies nicht erlaubt. Ein Mann am Fairlight wird diesen warmen, sanften Ton einfach erfassen und ihn dann mittels Keyboard über zwei Oktaven transponieren. Das Ergebnis klingt noch immer wie eine Flöte, doch es handelt sich um eine Flöte, die in der „realen“ Welt überhaupt nicht existieren kann.

Der Fairlight stellt die erfaßten Klänge, die auf Disketten gespeichert sind, auf dem Bildschirm dar. Unterschiedliche Charakteristiken eines individuellen Klangs können nacheinander überprüft werden. Oft ist es leichter festzustellen, ob ein Klang „falsch“ ist, wenn man ihn sieht statt ihn zu hören. Durch Betrachten und Erkennen der Verwandtschaft der unterschiedlichen Wellenformen innerhalb eines Klangs kann ein Punkt ausgewählt werden, von dem aus der Klang in eine Schleife (loop) gebracht, also wiederholt wird. Ist der richtige Punkt gewählt, schafft man die Illusion eines

From Francis Bacon's THE NEW ATLANTIS, published in 1624:

Wee have also Sound-Houses, wher wee practise and demonstrate all Sounds, and their Generation. Wee have Harmonies which you have not, of Quarter-Sounds, and lesser Slides of Sounds. Diverse Instruments of Musick likewise to you unknowne, some sweeter than any you have: Together with Bells and Rings that are dainty and sweet. We represent Small Sounds as Great and Deepe: Likewise Great Sounds, Extenuate and Sharpe; Wee make diverse Tremblings and Warblings of Sounds, which in their Originall are Entire. We represent and imitate all Articulate Sounds and Letters and the Voices and Notes of Beasts and Birds. We have certaine Helps, which sett to the Eare do further the Hearing greatly. We have also diverse Strange and Artificiall Echo's, Reflecting the Voice many times, and as it were Tossing it: And some that give back the Voice Lowder then it come, some Shriller, and some Deeper; Yea some rendring the Voice, Differing in the Letters or Articulate Sound, from that they receyve, Wee have also meanes to convey Sounds in Trunks and Pipes, in strange Lines, and Distances.

Dieses Dokument ist der utopischen Vision „Neu Atlantis“ entnommen, geschrieben von Francis Bacon (1561–1626). Seine Klangbeschreibungen waren eine Voraussage der ungewöhnlichen Kraft und Vielseitigkeit der elektronischen Musik von heute.

Aus Francis Bacons „Neu Atlantis“, veröffentlicht 1624

Wir haben auch akustische Werkstätten, wo wir alle Töne und ihre Erzeugungsarten untersuchen und erforschen. Wir kennen Harmonien, die bei Euch nicht üblich sind, indem wir nicht nur jenes harte und weiche Beta wie Ihr mischen, sondern auch die Viertel der Töne und gewisse sehr zarte Tremolos. Wir besitzen Musikinstrumente, die Euch noch nicht bekannt sind; einige davon bringen eine zarte Musik hervor, schöner als die Eure. Auch Glocken und Schellen von angenehmstem Klange haben wir. Wir erzeugen dünne Töne so wie starke und volle, volle auf gleiche Weise wie dünne und scharfe. Ferner rufen wir viele Schwankungen aus Tönen hervor, die ursprünglich einfach sind. Wir erzeugen alle artikulierten Laute und Buchstaben und ahmen sie künstlich nach, ebenso alle Stimmen und Laute der Säugetiere und Vögel. Wir haben ferner Hilfsmittel für das Gehör, die, an die Ohren gebracht, den Sinn selbst wie die lautliche Übertragung unterstützen. Wir haben auch viele wunderbare und kunstvolle Schallreflektoren, die Ihr Echo nennt und die die Stimme nicht nur vielfältig zurückwerfen, sondern sie einerseits auch verstärken, andererseits aber schwächen, ferner einige, die den artikulierten Laut, anders als er üblich ist, wiedergeben. Wir haben schließlich Mittel, Töne durch Rohre und andere Hohlräume, sogar auf gewundenen Wegen, zu übertragen.

Entnommen aus: Rowohlts Klassiker der Literatur und der Wissenschaft
Reihe: Philosophie des Humanismus und der Renaissance,
Band 3, 1966.



Dauertones. Beim analogen Klang ist das nicht möglich. Mit dieser Schleifenbildung sind also völlig neue, ungewöhnliche Dimensionen in der Musik möglich.

Dem Fairlight-Anwender stehen zwei Eingabemöglichkeiten neben der Echtzeit auf der Tastatur zur Verfügung. Bei der einen, als „Page R“ bezeichnet, wird ein traditionelles Notenliniensystem auf dem Bildschirm dargestellt, in das der Anwender über Keyboard die Noten eingibt. Taktfehler werden automatisch vom Computer „ausgeglichen“, da Metrum bzw. Takt durch den Anwender vorher eingegeben wurden.

Die zweite Möglichkeit besteht in der Verwendung von MCL (Music Composition Language). Bei dem Fairlight MCL wird vorausgesetzt, daß jedes Notenereignis mittels Keyboard eingegeben wird. Takt und Akzentuierung können Note um Note modifiziert werden. Da der Fairlight achttimmig ist, kann der Benutzer acht verschiedene Sequenzen eingeben, die in acht verschiedenen Klangarten oder „Stimmen“ spielbar sind. Diese lassen sich im Taktablauf minimal zeitlich versetzen – eine Sache von Millisekunden –, und das Gesamtergebnis wird durch die interne Uhr des Fairlight gesteuert.

Leistungsfähige Musiksysteme

Es stellt sich die Frage, wozu diese Art musikalischer Darbietung dienen soll, zumal sie von einem Computer ausgeführt wird. Die Antwort lautet, daß Menschen nie exakt im Takt spielen. Eines der wesentlichen Aufführungselemente – das gilt besonders für Jazz-Musiker, aber auch für Künstler der klassischen Musik – ist die Art, wie ein Musiker Takte „beugt“. Mit einem System wie dem Fairlight können bestimmte Interpretationsformen simuliert werden. Diese Simulationen lassen sich wiederum bei experimenteller Arbeit verwenden, genauso wie eine Computersimulation für das Design von Autokarosserien, Flugzeugflügeln und Hitzeschilden für Raumschiffe Anwendung findet.

Viele Musiker befürchten, daß Instrumente wie der Fairlight den Menschen ersetzen könnten, nicht zuletzt wegen der Simulationsfähigkeit der Systeme, die stets weiterentwickelt wird. Gruppen wie Wang Chung, Duran Duran und Culture Club arbeiten mit dem Fairlight. Oft ist nicht mehr zu unterscheiden, was tatsächlich „live“ gespielt und was Computer-generiert wurde. Eines steht fest: Hat ein Anwender erst einmal das Potential des Fairlight begriffen, ist er für ihn mehr als lediglich ein teures Spielzeug. Er ist ein einzigartiges neues Musikinstrument, dessen Möglichkeiten erst einmal ausgelotet werden müssen.

Der Fairlight ist, wenngleich sehr bekannt, nicht das einzige Instrument seiner Art. Das doppelt so teure Synclavier-System bietet in

größerem Umfang ähnliche Möglichkeiten. Die Daten werden dabei auf einer Winchester-Festplatte gespeichert. Mit dem Synclavier-System könnte eine komplette Schallplatte produziert und aufgezeichnet werden und würde somit einen 24spurigen Digitalrecorder überflüssig machen.

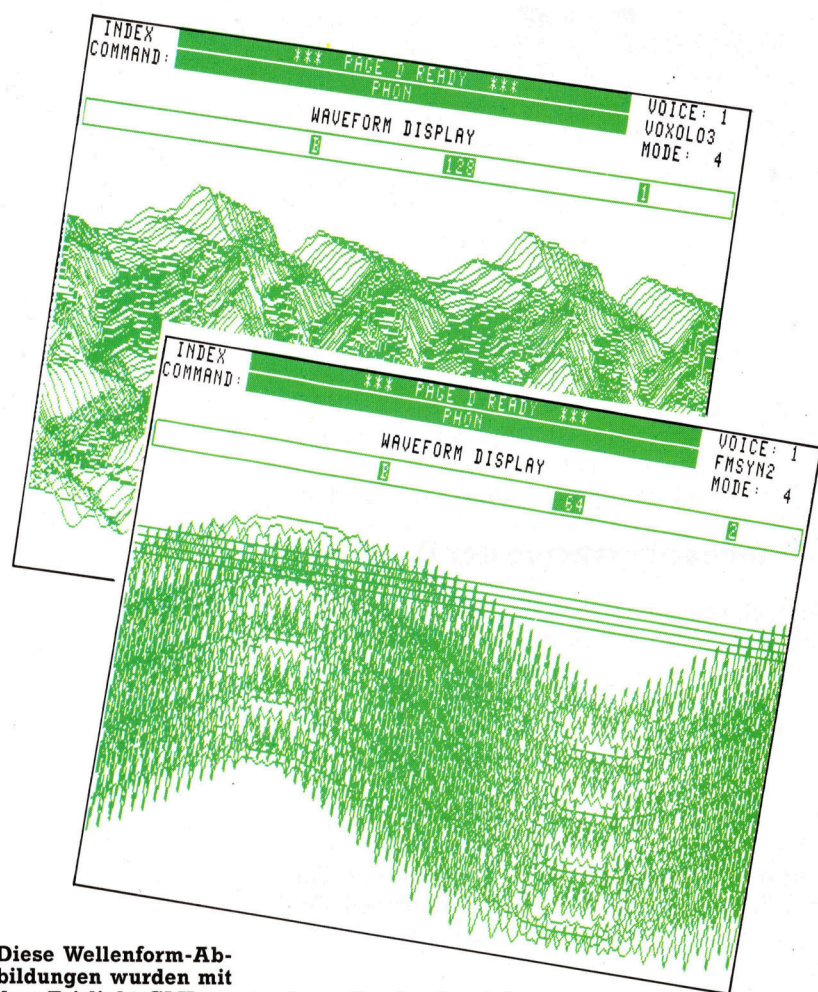
„Kurzweil“, bisher noch ein Prototyp, verarbeitet ein Programm auf der Basis der Mustererkennung. Das bedeutet praktisch: Wenn auf dem Keyboard eine Note gespielt wird, werden verschiedene Muster abgetastet und Charakteristiken eines Klangs miteinander kombiniert, um einen speziellen Klang zu erzeugen. Die gewählten Charakteristiken sollten die Art, in der die Musik gespielt wurde, wiedergeben. Damit kämen Charakter und „Feeling“ der Darbietung dem auf einem normalen Instrument Möglichen gleich. Der einzige Unterschied besteht darin, daß Klaviere und Flügel als Einzelinstrumente nicht genau gleich sind. Dagegen klingen alle Kurzweil-CMI-„Pianos“ prinzipiell gleich, außer, es gelänge dem Spieler, eine ganz spezifische Software auf der vorgenannten Basis zu entwickeln.

Ein dem Fairlight, Synclavier und Kurzweil weit überlegenes System, das alle Vorteile der drei vereint, soll bei Lucasfilm in Entwicklung sein. Das ist das Unternehmen, das für den „Krieg der Sterne“ verantwortlich zeichnet. ASP (Audio Signal Processor) soll alle nur vorstellbaren Arten digitaler Klangerzeugung bieten, die sonst nur in Studios von gigantischem Ausmaß möglich wären. Hatten wir es in den siebziger Jahren mit großen Aufnahmestudios, vollgepackt mit Computern und Synthesizern, zu tun, die auf mehrere Räume verteilt wurden, und heute bereits mit Geräten, die auf Schreibtischgröße verkleinert wurden, so wird das Studio der Zukunft im wahrsten Sinne des Wortes „tragbar“ sein.

Der Durchbruch der digitalen Technik ist nicht allein auf klangerzeugende Systeme und Geräte beschränkt. Bestandteil moderner Aufnahmestudios ist zumeist eine Reihe von Klangbearbeitungseinheiten. Das Hallgerät ist wohl eines der bekanntesten dieser Art. Die



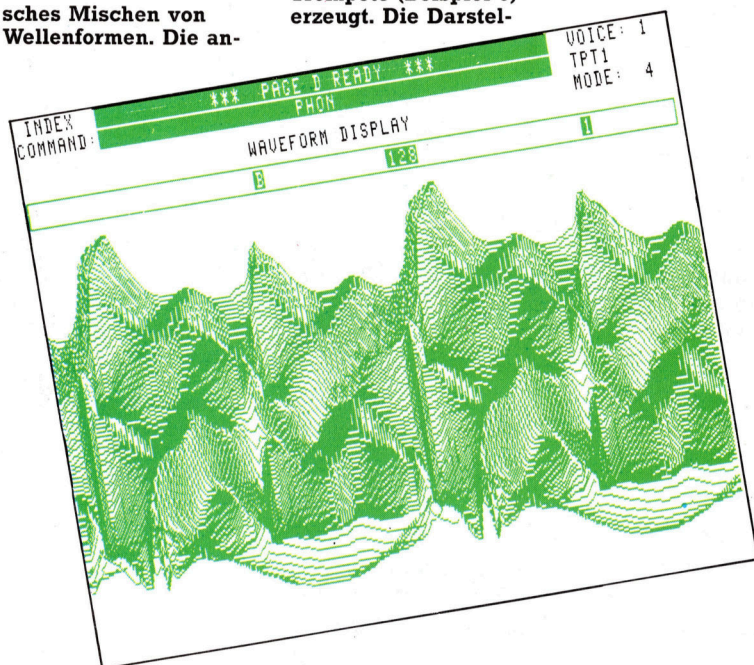
Die beim Fairlight CMI verwendete Kompositionssprache ist menügeführt. Die Wahl wird einfach über Anweisungen auf dem Bildschirm getroffen. Die dargestellten Wellenformen werden mit einer Reihe von Klang-Parametern in Datenform generiert, danach dargestellt und schließlich ausgedruckt.



Diese Wellenform-Abbildungen wurden mit dem Fairlight-CMI-Musik-System geschaffen und ausgedruckt. Die erste zeigt ein Sinuswellen-Muster, das unter Verwendung der FM-Synthese erzeugt wurde. Der Klang entstand durch elektronisches Mischen von Wellenformen. Die an-

deren Ausdrücke stellen Beispiele von „Sample Sounds“ dar. Die Wellenformen werden durch Digitalisieren des realen Klangs der menschlichen Stimme (Beispiel 2) bzw. einer Trompete (Beispiel 3) erzeugt. Die Darstel-

lung erfolgt „dreidimensional“ oder topografisch und zeigt die zeitliche Veränderung jedes in der Komposition verwendeten Klanges.



Musik wird durch die Einheit geleitet, um so ein Echo oder einen „Hall“ zu erzeugen. Viele Musiker bedienen sich dieses Geräts, um ihrer Musik einen ganz speziellen Klang zu verleihen, sei er nun ein Rockabilly-Gitarrist oder ein Reggae-Musiker.

Quantec z. B. ist ein Digitalgerät, das nicht nur Echo erzeugt, sondern imstande ist, die Akustik unterschiedlich großer Räume zu simulieren. Der kleinste „akustisch darstellbare Raum“ ist eine Schachtel von einem Kubikmeter Rauminhalt. Die vorprogrammierten Simulationen beinhalten Räume wie Wohnzimmer, Auditorien, Flugzeughallen und Kathedralen. Die wohl interessanteste Möglichkeit des Quantec besteht darin, den Hall über die natürlichen akustischen und physikalischen Grenzen hinaus zu verlängern. Findet beispielsweise ein Klangereignis in einem eine Kathedrale simulierenden Raum statt –, bei gleichzeitiger Hall-Maximierung –, dauert dies entsprechend lang. Das Ergebnis: Es ist, als höre man ein Echo derart, als seien zehn Grand Canyons miteinander verbunden!

Digitale Musik beim Film

Man erzählt, daß es um 1950 im amerikanischen Mittelwesten ein kleines Aufnahmestudio in der Nähe eines Getreidesilos gegeben habe. Dieser gewaltige Raum war Ursprung für den ganz speziellen Rockabilly-Sound des Studios. Digitale Verfahren wie Quantec sind weit aus realitätsbezogener. Sie finden in den unterschiedlichsten Produktionsstadien bei der Video- und Filmherstellung Anwendung, wobei die Schauspieler in akustisch „toten“ Räumen aufgenommen und die Filme anschließend mit Akustik ausgestattet werden.

Viele Menschen, Musiker wie Musikliebhaber, sind davon überzeugt, daß die Digitaltechnik zerstörerisch auf die Musik wirkt. Sie glauben, daß Musik, die auf künstliche Weise geschaffen und ausgeführt wird, die „menschliche“ Komponente, also Spontaneität und Ausdruckskraft, durch die technischen Möglichkeiten überlappt. Ein sicherlich überzeugendes Argument, das wir am Kinofilm überprüfen wollen.

Über Jahrzehnte war es filmisch möglich, Szenen aus beliebigen Winkeln zu fotografieren, zu zoomen und über größere Bereiche zu schwenken. Man konnte Sequenzen von Bildern wiederholen, sie verlangsamen, beschleunigen, rückwärts laufen lassen und den Film so bearbeiten, daß innerhalb weniger Sekundenbruchteile höchst ungewöhnliche Nebeneinanderstellungen möglich waren. Im Musikbereich ist eine vergleichbare Manipulationsmöglichkeit erst seit kurzer Zeit gegeben. Das Gros des Publikums akzeptiert Kino als ausdrucksvolles Medium. So ist naheliegend, daß auch digitale Musik derart betrachtet werden kann.



Serieller Zugriff

In dieser Folge wird untersucht, wie sich sequentielle Dateien für Programme einsetzen lassen und wie einige Beschränkungen dieser Dateiart umgangen werden können.

In sequentiellen Dateien sind alle Daten als Block auf Diskette oder Band gespeichert. Das Speicherformat bestimmt dabei auch die Zugriffsmethoden. So müssen beim Suchen bestimmter Informationen normalerweise alle in der Datei vorhandenen Daten gelesen werden, bis die gewünschte Information gefunden ist. Für die Änderung einzelner Datensätze wird oft erst die Ursprungsdatei bis zu der gewünschten Information kopiert, dann die Änderung eingefügt bzw. angehängt und schließlich der Rest der Ursprungsdatei hinter die Veränderungen geschrieben.

Bei diesen Vorgängen ist die Anordnung der Informationen innerhalb der Datei ein wichtiger Faktor. Normale Texte bestehen dabei fast immer aus Folgen von ASCII-Codes, die jeweils mit einer Endmarkierung abgeschlossen sind. Für die Datenbasis beispielsweise einer Bücherkartei muß die Information jedoch stärker strukturiert sein. Normalerweise werden die Daten dabei in Datensätze und Felder unterteilt, wobei jedes Buch seinen eigenen Eintrag beziehungsweise Datensatz erhält. Die Datensätze sind wiederum in Felder unterteilt, die Buchtitel, Autor, Herausgeber etc. enthalten. Spezialzeichen trennen die Struktureinheiten voneinander.

Trennung der Datensätze

Das Zeichen für den Wagenrücklauf oder RETURN (ASCII-Code 13) wird oft für die Trennung von Feldern und Datensätzen eingesetzt. Da alle Datensätze der Datei die gleiche Anzahl Felder enthalten, kann das Programm leicht herausfinden, wo ein Datensatz endet und ein neuer beginnt.

Sequentielle Dateien müssen folgende Anforderungen erfüllen: Es muß möglich sein, Datensätze zu editieren, zu löschen, abzurufen und hinzuzufügen. In den Diagrammen auf der nächsten Seite haben wir die Abläufe dieser verschiedenen Zugriffsarten dargestellt. Bei fast allen dargestellten Abläufen werden die Daten automatisch in eine neue (aktuelle) Datei kopiert. Die aktualisierten Informationen sind in der neuen Datei dabei an den korrekten Positionen gespeichert. Nach Abschluß des Kopiervorgangs wird die neue Datei als aktuelle Datei angesehen und die alte entweder gelöscht oder als „Sicherungskopie“ aufbewahrt.

Diese einfachen Techniken bilden die Grundlagen der sequentiellen Dateiverwaltung. Sie gehen jedoch von einer wichtigen Voraussetzung aus: Das Betriebssystem muß zwei Dateien zur gleichen Zeit offenhalten können, damit von einer Datei gelesen und in eine andere Datei geschrieben werden kann.

Sicherungskopie der Datei

Nicht alle Diskettensysteme erfüllen diese Anforderung. Und auf Heimcomputern, die ausschließlich mit Cassettenlaufwerken arbeiten, ist dies oft nur durch den Einsatz zweier Cassettenrecorder möglich. Einige Geräte sind daher auch mit zwei Recorder-Schnittstellen ausgerüstet.

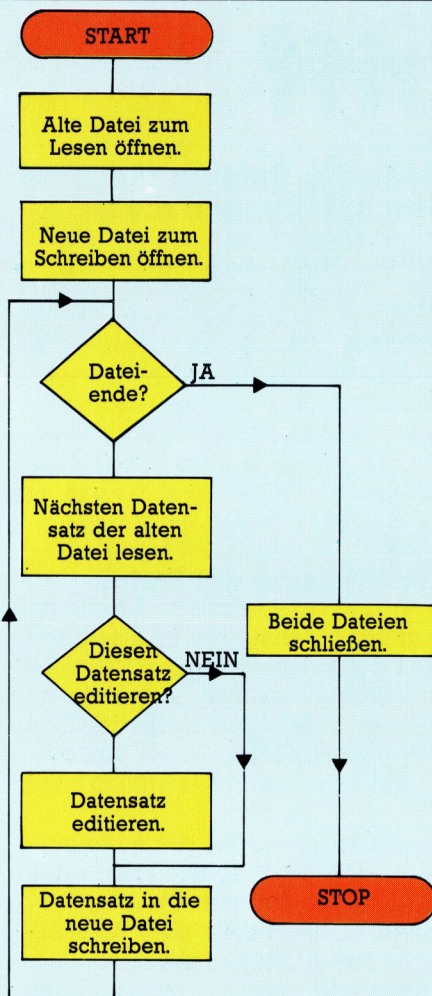
Die Speichertechnik, bei der die ursprüngliche und die modifizierte Datei aufbewahrt werden, findet hauptsächlich bei Dateien Einsatz, die sich wegen ihrer Größe nicht vollständig in den Arbeitsspeicher laden lassen und stückweise bearbeitet werden müssen. Bei kleineren Dateien kann eine hohe Bearbeitungsgeschwindigkeit erreicht werden, wenn die gesamte Datei in eine Speichermatrix geladen und dort bearbeitet wird. Bei dieser Methode lassen sich alle Verwaltungsvorgänge im Arbeitsspeicher ausführen, wobei die gesamte neuerstellte Datei nach Abschluß der Arbeiten auf Band oder Diskette übertragen und gespeichert wird.

Diese Arbeitsweise enthält eine Gefahr: Änderungen in der Datei sind nur dann permanent fixiert, wenn die Information auf das Speichermedium geschrieben ist. Bricht das Programm während eines Editiervorgangs ab, sind alle bis zu dem Zeitpunkt geänderten Daten verloren. Wenn Sie diese Methode für Ihre Programme einsetzen, sollten Sie Ihre Dateien sooft wie möglich speichern, damit Sie immer eine möglichst aktuelle Sicherungskopie zur Verfügung haben. Auch sollte beim Beenden des Programms automatisch eine endgültige Kopie erstellt werden.

Viele kleinere Systeme bieten als Speichermöglichkeit nur sequentielle Dateien an. Deren Einsatz ist zwar etwas umständlich, aber dennoch leicht zu verstehen. In der nächsten Folge werden wir die schnellen und einfachen Zugriffsmöglichkeiten der Dateien mit wahlfreiem Zugriff (Random Access) ausführlich erläutern.

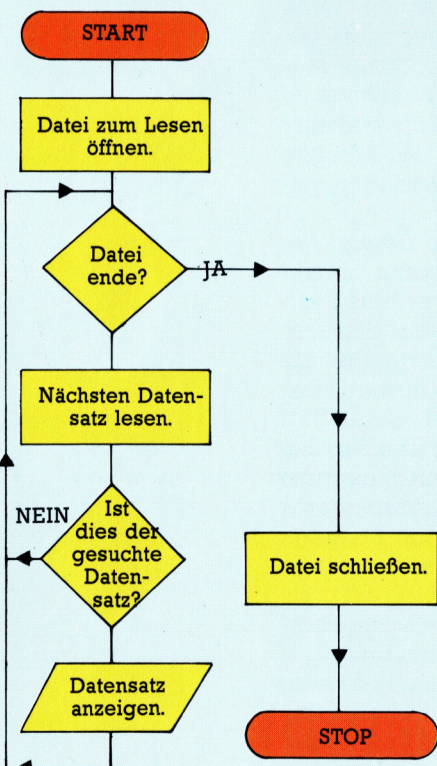
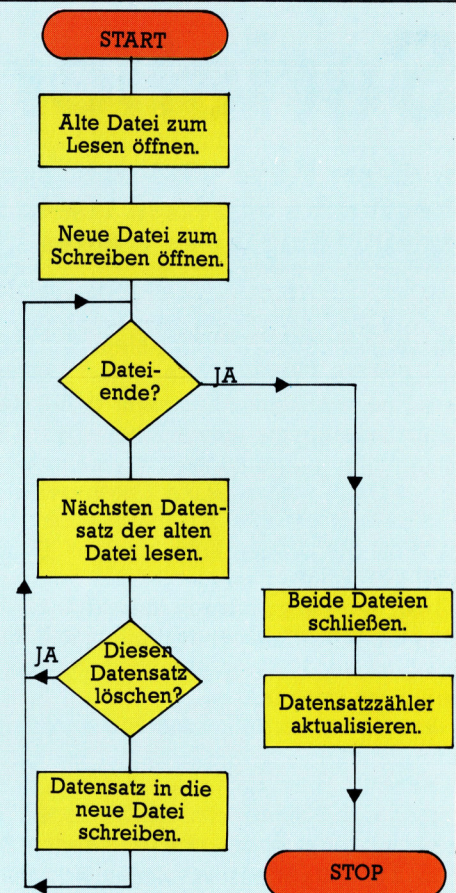
Datensätze editieren

Beim Editieren eines Datensatzes werden zunächst alle Informationen vor dem zu verändernden Datensatz in eine neue Datei kopiert. Der Datensatz wird dann in den RAM-Bereich geladen und dort vom Programm editiert. Schließlich wird der veränderte Datensatz in der neuen Datei gespeichert. Und alle weiteren Datensätze der alten Datei werden in die neue Datei kopiert. Mit einem Schreibvorgang läßt sich eine beliebige Anzahl Datensätze nacheinander bearbeiten.



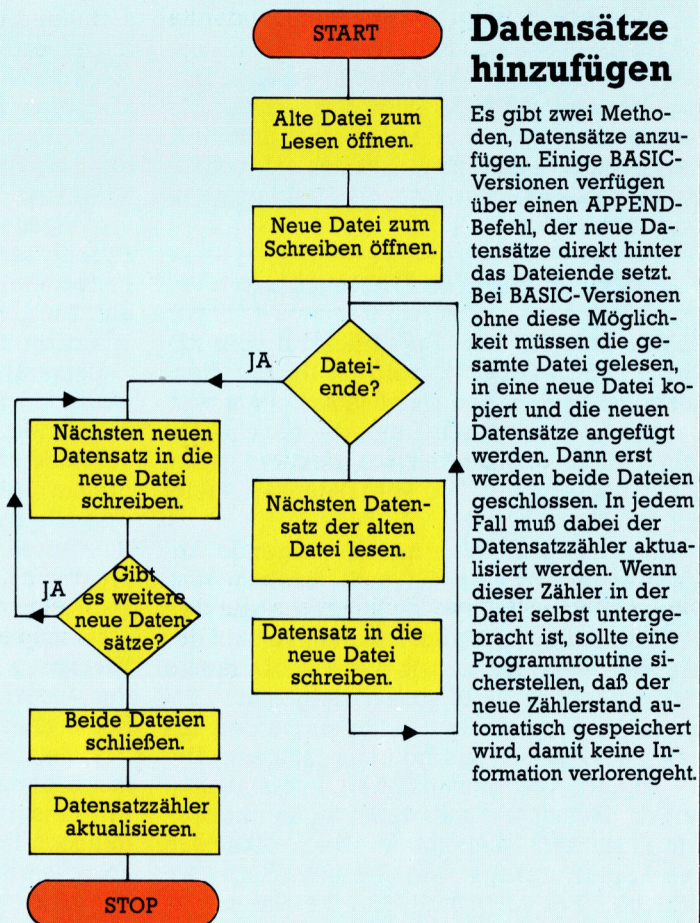
Datensätze löschen

Beim Löschen eines Datensatzes wird die Datei bis zu dem Datensatz, der gelöscht werden soll, gelesen und kopiert. Der betreffende Datensatz wird dann zwar gelesen, aber nicht in die neue Datei übernommen. Schließlich werden die restlichen Datensätze der alten Datei ebenfalls in die neue Datei kopiert. Bei einem einzigen Durchgang können mehrere Datensätze gelöscht werden. Wie beim Hinzufügen von neuen Datensätzen muß auch hier der Stand des Datensatzzählers sofort aktualisiert werden.



Datensätze lesen

Sequentielle Dateien eignen sich nicht für Programme, die einzelne Datensätze schnell nacheinander lesen müssen, da für einen Lesevorgang die gesamte Datei von Anfang an durchsucht werden muß. Soll ein einziger Lesevorgang mehrere Datensätze finden, dann müssen die Suchkriterien in der gleichen Reihenfolge stehen, in der die Datei organisiert ist. Aus diesem Grund werden sequentielle Dateien vor dem Speichern oft sortiert (z. B. alphabetisch).



Datensätze hinzufügen

Es gibt zwei Methoden, Datensätze anzufügen. Einige BASIC-Versionen verfügen über einen APPEND-Befehl, der neue Datensätze direkt hinter das Dateiende setzt. Bei BASIC-Versionen ohne diese Möglichkeit müssen die gesamte Datei gelesen, in eine neue Datei kopiert und die neuen Datensätze angefügt werden. Dann erst werden beide Dateien geschlossen. In jedem Fall muß dabei der Datensatzzähler aktualisiert werden. Wenn dieser Zähler in der Datei selbst untergebracht ist, sollte eine Programmroutine sicherstellen, daß der neue Zählerstand automatisch gespeichert wird, damit keine Information verlorengeht.



Der Mephisto PHC 64

Neben dem futuristischen Design und der joystickgesteuerten Cursorführung steht als besonderes Merkmal die Möglichkeit, das Gerät auf knapp vier Megabyte zu erweitern.

Der Mephisto PHC 64 (ursprünglich Elan bzw. Enterprise 64 genannt) brach schon vor seiner Einführung in den Markt eine Art Rekord: Zwischen Ankündigung des Gerätes und Auslieferung an den Handel liegen nunmehr zwei lange Jahre.

Wie bei dem Sinclair QL hatte der Hersteller des Mephisto Schwierigkeiten, all die angekündigten beeindruckenden Eigenschaften in Spezialchips unterzubringen. Jetzt sind die Probleme beseitigt, und es stellt sich die Frage, ob sich das Warten lohnte.

Das ungewöhnliche, flache Design des Mephisto mutet futuristisch an. Statt des üblichen kastenähnlichen Gehäuses hat die Vorderseite abgerundete Kanten, deren Linien bis zum eckigen Hauptkasten leicht ansteigen.

Das Gerät verfügt über eine Standardtastatur, deren Steuertasten (Control, Escape und Enter) bequem zu bedienen sind. Ungewöhnlich sind die Funktionen der zwei Tasten Erase und Delete, da sie vorwärts und rückwärts löschen können. Mit der Hold-Taste lassen sich laufende Programme anhalten.

Zu beiden Seiten der Tastatur befinden sich grüne Steuertasten. Über dem Tastenfeld liegen acht blaue Funktionstasten, die doppelt belegt werden können. Sie sind mit oft eingesetzten Befehlen wie LIST und AUTO definiert, lassen sich jedoch mit zusätzlichen Funktionen programmieren.

Integrierter Joystick

Die rote Stop-Taste hält ein laufendes Programm an, das dann mit CONT oder RUN neu gestartet werden kann. Normalerweise führt die Escape-Taste diese Funktion aus. Auf dem Mephisto wird diese Taste jedoch für die Umschaltung von einer Programmebene auf die nächste eingesetzt.

Die Tasten sind erhöht in das Gehäuse eingebaut. Obwohl die Tastatur besser aussieht als die des Sinclair QL, macht sie auf den ersten Blick einen unzuverlässigen Eindruck. Dieser schwindet jedoch beim praktischen Einsatz, und es stellt sich heraus, daß sie zuverlässiger arbeitet als bei vielen Maschinen mit einem professionell anmutenden Tastenfeld. Der eingebaute Joystick hat eine Länge



von drei Zentimetern und ist, wie das übrige Keyboard, auf eine Membran montiert.

In der linken Gehäusesseite befindet sich eine Steckleiste für Cartridges. Hier wird – ungewöhnlich für eine moderne Maschine – das BASIC als Cartridge eingesteckt. Zweifellos werden in Zukunft auch andere Sprachen für den Mephisto verfügbar sein. Obwohl die Steckleiste tief in das Gehäuse eingelassen ist, läßt sich die Cartridge problemlos einstecken. Auf der rechten Seite der Maschine liegt eine parallele Steckleiste, die ebenfalls für zukünftige Erweiterungen gedacht ist.

Auf der Rückseite des Gerätes ist der Reset-Knopf angebracht, über den bei einmaligem Drücken der Computer zu einem Warmstart veranlaßt wird. Wird er zweimal gedrückt, läßt die Maschine die Cartridge. Die zweite Funktion ist notwendig, da der Computer nur bei einem Kaltstart prüft, ob eine Cartridge eingesetzt ist.

Ferner befinden sich auf der Gehäuserückseite eine ganze Reihe Schnittstellen, darunter ein Buchsenpaar für Joysticks und eine parallele Centronics-Schnittstelle für den Anschluß eines Druckers. Über eine RS232/432-kompatible serielle/Netzwerk-Schnittstelle kann der Mephisto mit einem lokalen Netzwerk verbunden werden. Auch lassen sich daran serielle Peripheriegeräte anschließen oder über ein externes Modem Datenübertragung mit anderen Computern durchführen. RGB- und TV-Ausgang sind ebenfalls vorhanden.

Alle Schnittstellenbuchsen sind als Platinenstecker ausgelegt. Bei einem Erfolg der Ma-

Obwohl der von der britischen Firma Enterprise gebaute Mephisto schon 1983 angekündigt war, dauerte es zwei Jahre, bis das Gerät auf dem Markt erschien. Die Verzögerung hat die Firma mit Sicherheit viel Geld gekostet; denn wenn die Maschine 1983 noch viel Neues geboten haben mag, so hat die Konkurrenz diesen Vorsprung heute doch längst aufgeholt.

schine dürften genügend Hersteller Zusatzgeräte für den Anschluß von beispielsweise Atari-Joysticks anbieten. Platinensteckverbindungen ermöglichen auch den Anschluß von Steuergeräten und leistungsfähigen Touch Tablets. Zum gegenwärtigen Zeitpunkt sind jedoch noch keine weiteren Peripheriegeräte auf dem Markt erhältlich.

Auf der Rückseite der Maschine befindet sich außerdem eine Cassettenrecorder-schnittstelle mit zwei REMote-Steckern, die zwei Recorder gleichzeitig ansteuern kann – je einen für Eingabe und Ausgabe.

Beim Anschalten des Gerätes zeigt der Bildschirm das Hersteller-Logo. Wenn die BASIC-Cartridge nicht angeschlossen ist, erscheint nach dem Drücken der Enter-Taste automatisch das integrierte Textsystem. Mit den Funktionstasten kann nun eine Reihe von Bearbeitungsmöglichkeiten für die Eingabe und Formatierung von Texten aufgerufen werden. Die wichtigsten Funktionen werden am oberen Bildschirmrand dargestellt, während die restlichen Bearbeitungsmöglichkeiten sich mit der Funktionstaste F5 sehr einfach über ein Menü aufrufen lassen.

Praktisch ist dabei die Möglichkeit, zwischen einer Darstellung von 40 (Default) und 80 Zeichen pro Zeile wählen zu können. Da der PHC 64 mit einem Z80-Prozessor und dem 80-Zeichen-Format ausgestattet ist, wird es bei Verfügbarkeit der Diskettenstation möglich sein, CP/M einzusetzen.

Leistungsfähiges BASIC

Der Cursor wird mit dem Joystick und nicht mit Steuertasten über den Bildschirm bewegt. Viele Anwender werden sich nur schwer an diese Methode gewöhnen können, doch mit ein wenig Übung läßt sich der Cursor präzise steuern.

Ist beim Anschalten des Gerätes die BASIC-Cartridge eingesetzt, ruft das Drücken der Enter-Taste diese Programmiersprache automatisch auf. Hier zeigt sich der Vorteil der Cartridge gegenüber einer Speicherung des BASIC im RAM: Dem Anwender stehen 49 KByte Arbeitsspeicher zur Verfügung. Das von Intelligent Software erstellte BASIC ist außergewöhnlich vielseitig und weckte schon bei seiner Ankündigung hohe Erwartungen.

Der Mephisto ist mit vier Klanggeneratoren ausgestattet, die sich über den Befehl SOUND ansprechen lassen. Die Tonsteuerungsbefehle PITCH, DURATION und ENVELOPE können Klänge individuell verändern. Weiterhin erzeugt der Mephisto Stereosignale, die sich mit den Befehlen LEFT und RIGHT auf je einen der beiden Lautsprecher leiten lassen.

Wenn auch der Klang des Mephisto ausgezeichnet ist, so enttäuscht die Qualität des eingebauten Lautsprechers, der kaum besser ist als der des Sinclair Spectrum. Der Computer

Cassettenrecorder-schnittstelle

Die Schnittstelle enthält vier Buchsen – je eine für Ein- und Ausgabe und zwei REMote-Anschlüsse.

Wärmeableitung

Diese Kühlrippen leiten überschüssige Wärme ab.

Cartridge-Steckleiste

Die BASIC-Cartridge wird hier eingesetzt.

Lautsprecher

Der Mephisto verfügt über einen eingebauten Lautsprecher, kann aber auch über die Monitorschnittstelle an ein HiFi-System angeschlossen werden.

RAM-Chips

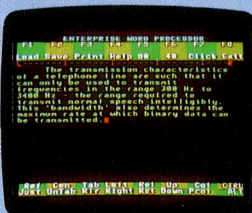
In diesen Chips ist der Arbeitsspeicher von 64 KByte enthalten.

Zentraleinheit

CPU des Mephisto ist der weit verbreitete Z80A-Prozessor.

Eingebaute Software

40-Zeichen-Darstellung (Textverarbeitung)



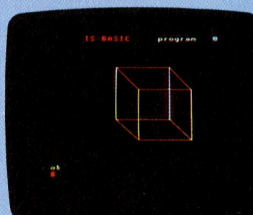
80-Zeichen-Darstellung (Textverarbeitung)



Grafikprogramm



Grafikdarstellung



Im Gegensatz zu anderen Microcomputern verfügt der Mephisto über ein eingebautes Textsystem. Ist keine Cartridge eingesetzt, wird das Textsystem automatisch aufgerufen. Es kann 40 oder 80 Zeichen pro Zeile darstellen. Auch im 80-Zeichen-Format ist die Schrift erstaunlich gut lesbar.

Eine der großen Überraschungen des Mephisto ist der Einsatz der Turtle-Grafik. Über die Angabe von Richtung und Länge, die der „Zeichenstift“ zurücklegen soll, lassen sich komplizierte Zeichnungen anfertigen.



Joystick

Der Mephisto setzt für die Cursorsteuerung des Bildschirmditors einen Joystick statt Steuertasten ein.

Reset-Knopf

Das Drücken dieses Knopfes löst einen „Warmstart“ aus.

Erweiterungssteckleiste

Dieser parallele Bus läßt sich für zukünftige Erweiterungen einsetzen.

Chip für die Klangsteuerung

Dies ist „Dave“, einer der beiden Spezialchips des Mephisto. Er steuert vier Klangkanäle mit acht Oktaven.

Chip für die Bildschirmsteuerung

Dieser als „Nick“ (der Name des Designers) bezeichnete Spezialchip macht die ausgezeichnete Grafik des Mephisto möglich.

läßt sich jedoch ohne weiteres an ein HiFi-System anschließen.

Ebenfalls beeindruckend sind die Grafikfähigkeiten. Es gibt acht Grafikarten, bei denen sich 2, 4, 14 oder 256 verschiedene Farben in hoher (HIRES) oder niedriger (LORES) Auflösung darstellen lassen. LORES hat nur die Hälfte der horizontalen Auflösung von HIRES. Die Anzahl der gleichzeitig darstellbaren Farben variiert je nach Auflösung. HIRES 2 kann zwei Farben mit der Auflösung 640×180 anzeigen, während HIRES 256 alle 256 Farben auf den Schirm bringt, die Auflösung dann aber nur 80×180 Pixel beträgt.

Spezielle Farbbefehle

Die Farben lassen sich entweder über Zahlen von 0 bis 255 anwählen oder – bei acht vorprogrammierten Farben – durch einen Namen, zum Beispiel WHITE oder RED. Interessanterweise können die Farben über die RGB-Befehle gemischt werden. Rot entspricht dabei $RGB(1,0,0)$ und Grün $RGB(0,1,0)$. Die Addition erzeugt $RGB(1,1,0)$ – Gelb.

Wäre der Mephisto wie angekündigt 1983 verfügbar gewesen, hätte er den Markt im Nu erobert; schon die Qualität der eingebauten Software und die Erweiterungsmöglichkeit auf vier Megabyte hätten seinen Erfolg sichergestellt. Heute ist jedoch die Konkurrenz weitaus größer. Die neuen Geräte von Atari, Commodore und Sinclair, um nur einige Firmen zu nennen, bieten inzwischen ähnliche Fähigkeiten. Dazu kommt, daß der Mephisto PHC 64 für einen relativ hohen Preis von circa 1200 Mark angeboten wird. Bleibt zu hoffen, daß die angekündigten Peripheriegeräte und das 3,5-Zoll-Diskettenlaufwerk bald fertiggestellt sein werden. Das englische Softwarehaus Intelligent arbeitet derzeit daran, das Programmangebot für den HPC 64 zu erweitern.

Mephisto PHC 64

ABMESSUNGEN

$60 \times 260 \times 405$ mm

ZENTRALEINHEIT

Z80A mit 4 MHz

SPICHERKAPAZITÄT

64 K RAM, von denen 49K für BASIC-Programme zur Verfügung stehen (32K für das Betriebssystem und das Textsystem und weitere 16 K über die BASIC-Cartridge). Erweiterungsfähig auf 4 Megabyte.

BILDSCHIRMDARSTELLUNG

Textsystem: maximal 50 Zeilen mit je 84 Zeichen. Acht Grafikarten, bei höchster Auflösung 672×512 Pixel. Maximal 256 Farben gleichzeitig darstellbar.

SNITTSTELLEN

Centronicskompatible Parallelschnittstelle, zwei Schnittstellen für Joysticks, RS232-kompatible serielle Schnittstelle, RGB- und TV-Anschluß, Erweiterungssteckleiste.

PROGRAMMIERSPRACHEN

BASIC, weitere in Vorbereitung.

TASTATUR

69 Schreibmaschinentasten mit acht Funktionstasten. Eingebauter Joystick.

HANDBÜCHER

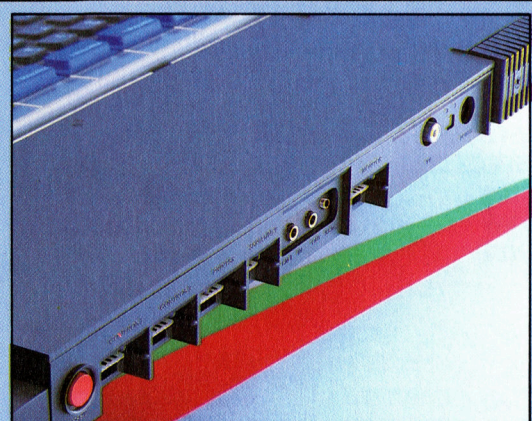
Der Hersteller liefert ein kurzes Einführungshandbuch über die Aufstellung des Computers. Darin enthalten sind auch Diagramme zur Erläuterung von Tastatur und Schnittstellen. Ebenfalls enthalten ist ein Lehrbuch für BASIC.

STÄRKEN

Der Computer verfügt über ein gutes BASIC und besitzt ausgezeichnete Grafik- und Klangmöglichkeiten. Ein Textsystem ist eingebaut.

SCHWÄCHEN

An die außerhalb des Standards liegenden Schnittstellenverbindungen sind zur Zeit nur wenige Peripheriegeräte anschließbar. Das derzeitige Softwareangebot ist gering.



Schnittstellen zur Peripherie

Mit dem Einsatz von Steckverbindungen statt DIN-Buchsen weicht der Mephisto von der Norm ab. Peripheriegeräte von anderen Herstellerfirmen lassen sich nur mittels Spezialkabel anschließen. Mit Hilfe des Netzwerkanschlusses lassen sich 32 HPC 64 miteinander verbinden.



BASIC-Cartridge

Äußerst ungewöhnlich für eine moderne Maschine ist der Einsatz der 16-K-ROM-Cartridge mit dem BASIC von Intelligent Software. Durch die Verwendung von externen Programmodulen läßt sich das BASIC jedoch leicht gegen andere Programmiersprachen austauschen.



LOGO-Geometrie

In dieser Folge werden Prozeduren entwickelt, die Zykloiden auf den Bildschirm zeichnen. Welche Überlegungen dafür notwendig sind, erfahren Sie nachstehend.

In einer früheren Folge des Kurses wurde gezeigt, wie man in LOGO einen einfachen Kreis zeichnet:

```
TO KREIS
  REPEAT 360 [FORWARD 1 RIGHT 1]
END
```

Das Ergebnis ist ein in etwa kreisförmiges Gebilde, dessen Fertigstellung allerdings reichlich lange dauert.

Die Prozedur zeichnet keinen glatten Kurvenverlauf, sondern ein 360-seitiges Polygon, das jedoch für die meisten Zwecke ausreichend ist. Für kleinere Grafikexperimente genügen auch „Kreise“ mit 60 oder 30 Seiten, die entsprechend schneller gezeichnet werden. Die folgenden Programme verwenden entweder 60- oder 120-seitige Polygone, die Sie je nach Wunsch modifizieren können.

Zunächst aber ist es notwendig zu wissen, was eine Zykloide ist. Stellen Sie sich einen Kreis vor, der auf einer Geraden entlangläuft. Markieren Sie nun einen Punkt im Kreisfeld Ihres imaginären Kreises und verfolgen Sie dann den Weg, den dieser Punkt beim Rollen des Kreises beschreibt. Die daraus resultierende Kurve wird „Zykloide“ genannt. Wir versuchen, auf dieser Basis ein Programm zu erstellen, das derartige Kurven erzeugt.

Zur Darstellung der Zykloide machen wir „Schnappschüsse“ nach jeder Drehung des Kreises um sechs Grad auf der Linie. Bei dieser Drehung bewegt sich der Kreis um $(2 \cdot \pi \cdot \text{Radius} / 60)$ Einheiten vorwärts. Die X-Koordinate des Kreismittelpunktes nimmt um diesen Wert zu (die Y-Koordinate bleibt natürlich unverändert). Gleichzeitig hat die Linienrichtung, die zum Markierungspunkt führt, um sechs Grad zugenommen.

Das Programm stellt zur Lösung vier einfache Aufgaben:

1. Bewegung des Kreismittelpunktes,
 2. Setzen der Turtle in den Mittelpunkt,
 3. Vorgabe der richtigen Richtung und
 4. Vorwärtsbewegung auf dem Radius.
- Damit wird die Turtle in die nächste zu zeichnende Position gebracht. Wir zeichnen einen Punkt auf dem Bildschirm und wiederholen das Ganze.

SETSCREEN ist die erste Prozedur, die von ZYKLOIDE aufgerufen wird. Hier geht es um Details, die zur Darstellung nötig sind. Die eigentliche Aufgabe von SETSCREEN ist die

Festlegung des Seitenverhältnisses und der Aufruf von NOWRAP, der das Programm beendet, wenn die Kurve den Bildrand erreicht.

```
TO BEWEGMITTE
  MAKE "XMITT :XMITT + :STEP
END
TO PUNKT
  PD
  FORWARD 1
  BACK 1
  PU
END
```

Wählt man statt eines Punktes auf dem Kreisbogen einen Punkt im Innern des Kreises, ist das Resultat eine „verkürzte Zykloide“. Legen wir einen Punkt außerhalb des Kreises fest, entsteht eine „verlängerte Zykloide“. Um diese Auswirkungen beobachten zu können, wird ZYKLOIDE durch Eingaben modifiziert, die die Entfernung des Markierungspunktes vom Kreisbogen darstellen. Positive Werte erzeugen verkürzte, negative Werte verlängerte Zykloide.

```
TO ZYKLOIDE
  SETSCREEN
  MAKE "WINKELSCHRITT 6
  MAKE "PI 3.14
  MAKE "RADIUS 15
  MAKE "KREISBOGEN 2 * :PI * :RADIUS
  MAKE "STEP :KREISBOGEN / (360 /
    :WINKELSCHRITT)
  MAKE "XMITT (-150)
  ZYK 0
END
TO SETSCREEN
  .ASPECT 0.93
  NOWRAP
  DRAW
  PENUP
  HT
END
TO ZYK :WINK
  BEWEGMITTE
  SETXY :XMITT 0
  SETH :WINK
  FORWARD :RADIUS
  PUNKT
  ZYK :WINK + :WINKELSCHRITT
END
TO ZYKLOIDE :AUSSEN
  SETSCREEN
```




```

MAKE "WINKELSCHRITT 6
MAKE "PI 3.14
MAKE "RADIUS 15
MAKE "KREISBOGEN 2 * :PI * :RADIUS
MAKE "STEP :KREISBOGEN / (360 /
:WINKELSCHRITT)
MAKE "XMITT (-150)
MAKE "ENTFERNUNG :RADIUS -
:AUSSEN
ZYK 0
END
TO ZYK :WINK
BEWEGMITTE
SETXY :XMITT 0
SETH :WINK
FORWARD :ENTFERNUNG
PUNKT
ZYK :WINK + :WINKELSCHRITT
END

```

Durch das Markieren der Punkte läßt sich leicht verfolgen, was geschieht. Die Prozedur VERBINDEN fügt die einzelnen Punkte zu einer Kurve zusammen:

```

TO VERBINDEN :A :B
SETPOS :A
PD
SETPOS :B
PU
END
TO SETPOS :POS
SETXY FIRST :POS LAST :POS
END

```

Die Prozedur verbindet die Koordinaten der beiden eingegebenen Punkte. Zum Beispiel: VERBINDEN [12 34] [67 89]. In diesem Programm muß die alte Position des Punktes festgehalten und dann mit der gegenwärtigen verbunden werden. Ergebnis des verbesserten Programms:

```

TO ZYKLOIDE :AUSSEN
SETSCREEN
MAKE "WINKELSCHRITT 6
MAKE "PI 3.14
MAKE "RADIUS 15
MAKE "KREISBOGEN 2 * :PI * :RADIUS
MAKE "STEP :KREISBOGEN / (360 /
:WINKELSCHRITT)
MAKE "XMITT (-150)
MAKE "ENTFERNUNG :RADIUS -
:AUSSEN
MAKE "ALTPOS LIST :XMITT
:ENTFERNUNG
ZYK 0
END
TO ZYK :WINK
BEWEGMITTE
SETXY :XMITT 0
SETH :WINK
FORWARD :ENTFERNUNG

```

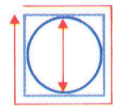
```

MAKE "NEUPOS POS
VERBINDEN :ALTPOS :NEUPOS
MAKE "ALTPOS :NEUPOS
ZYK :WINK + :WINKELSCHRITT
END
TO POS
OUTPUT LIST XCOR YCOR
END

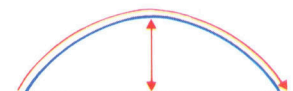
```

Experimentieren Sie doch einmal mit diesen Prozeduren. In Mathematiklehrbüchern wird behauptet, daß die Bogenlänge einer Zykloide gleich dem Umkreis des Quadrates ist, das den erzeugenden Kreis umschließt. Durch Modifizierung der Prozeduren läßt sich dieser Lehrsatz überprüfen.

Beinhaltet Ihre LOGO-Version Sprites, haben Sie eine andere und sogar bessere Möglichkeit, die Punkte darzustellen. Ein Vorteil dabei wäre, daß Sie durch Verwendung von TELL, YCOR und XCOR feststellen könnten, wo sich der Punkt befindet.



Quadrat mit Umkreis



Der Zykloidenbogen

LOGO-Dialekte

Beachten Sie die unterschiedliche Schreibweise des IF-Befehls, z. B.: IF :A = 120 [STOP].

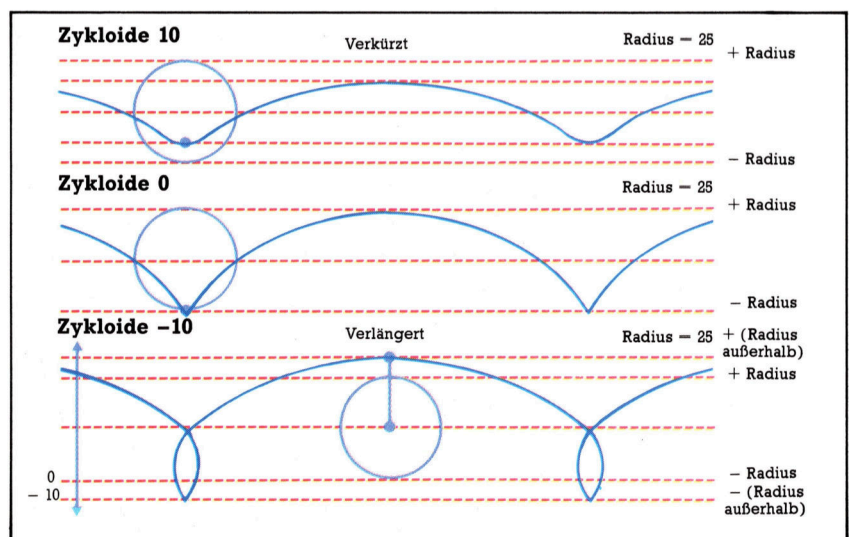
SETPOS und POS sind als Befehle definiert. In einigen Versionen muß SETXY durch SETPOS ersetzt werden (die Eingabe dafür ist eine Liste) und DRAW durch CS.

Ersetzen Sie gegebenenfalls NOWRAP durch FENCE. Da es beim Atari-LOGO kein FENCE gibt, wird WINDOW verwendet und anschließend «BREAK», um die Prozedur zu stoppen.

Für die Definition des Seitenverhältnisses gilt:

.SETCR beim Atari,
SETSCRUNCH beim Apple,
SETSCRUNCH, gefolgt von einer Liste, beim
Sinclair Spectrum.

Ein Zykloide ist eine Kurve, die ein Punkt auf dem (verlängerten) Radius eines auf einer Geraden rollenden Kreises beschreibt. Der Kurvenverlauf hängt davon ab, ob der Markierungspunkt im Kreiskörper, außerhalb des Kreises oder auf der Kreislinie liegt.





Wiederholung

Bevor es im elektronischen Selbstbau-Kursus weitergeht, sollten Sie sich die Zeit für einen kleinen Rückblick auf die behandelten Themenkreise nehmen.

Am Anfang des Kurses kam die Bedeutung geeigneter Werkzeuge zur Sprache. Für Verbesserungen und Reparaturen sollte man dabei unbedingt eine gewisse Minimal-Ausrüstung bereithalten. Zwar läßt sich eine Kreuzschlitzschraube auch mit einem flachen Schraubendreher, eine sehr kleine Mutter auch mit der Pinzette lösen – ohne Schaden für Gerät oder Material geht es so aber auf die Dauer nicht. Auf lange Sicht lohnt die Anschaffung einiger Spezial-Werkzeuge.

Ähnlich ist es auch beim Verdrahten – bei einer mit geeignetem Werkzeug sauber gelöteten Verbindung ist das Fehlerrisiko geringer als bei einer hastigen „Pfuscharbeit“. Lötzinn sollte zum Beispiel immer nur auf den Draht, nie auf die Lötkolbenspitze gegeben werden. Die Spitze wird vom Flußmittel des Zinns leicht angegriffen. Bei zwei zu verbindenden Drähten sollten Sie darauf achten, daß die gesamte Kontaktstelle mit Zinn abgedeckt ist. Erst dann nimmt man den LötKolben beiseite und kühlt durch leichtes Anpusten. Dieses Verfahren verhindert das Entstehen von „kalten“ Lötstellen.

Die Grundlagen der digitalen Elektronik haben wir bereits erläutert. Dabei wurden die Funktionen der wichtigsten Bauelemente erklärt, von denen der Widerstand das unkomplizierteste ist.

Kondensatoren dienen im Computer zum Ausfiltern unerwünschter Signalanteile. Jedes Signal wird bei der Übertragung, Verstärkung oder Veränderung mehr oder weniger verzerrt – der Kondensator hilft, das Signal wieder zu entzerren. Erst dadurch kann ein ungestörter Signalfuß erreicht werden.

Trotz der nützlichen Funktionen von Widerständen und Kondensatoren stand der Transistor in der Mitte unserer Bauteil-Kunde. Er hat im Computer die Aufgabe, Signale zu verarbeiten. Je nach Betriebsart arbeitet der Transistor als Verstärker oder als Schalter. Wichtig ist, daß ein Transistor jedes Signal in Abhängigkeit von anderen Signalen ein- oder ausschalten kann. Damit erhält er die Funktion eines logischen Gatters.

Im Selbstbau-Kurs haben Sie auch gesehen, daß man nur wenige Bauteile neben dem Transistor braucht, um die einfachsten Logikschaltungen (NOT-, OR-, AND-Gatter) aufzubauen.

Für ein einzelnes Logik-Gatter gibt es zwar kaum eine sinnvolle Einsatzmöglichkeit, mehrere zusammen können aber Daten ver- und

bearbeiten. Im letzten Kursabschnitt haben wir mit den logischen Gattern von zwei ICs einen Halbaddierer hergestellt – eine Schaltung, die zwei Binär-Bits addieren kann.

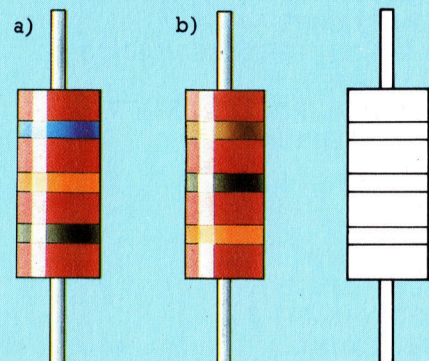
Integrierte Schaltkreise (ICs) sind die kompliziertesten Bauteile der Elektronik. Die von uns bisher verwendeten Transistor-Transistor-Logik-ICs (TTL) gehören zu den SSI-(= Small Scale Integration) Elementen, die auf einem Chip nur relativ wenige Transistoren vereinigen. Früher wurden noch ganze Computer aus SSI-Chips aufgebaut. Heute hat die Anzahl der Transistoren auf einem einzigen Chip gewaltig zugenommen: Schon bei MSI-(Medium Scale Integration) Elementen findet man komplette Logikschaltungen in einem IC.

Inzwischen ist die Technik bei LSI- und sogar VLSI-Chips angelangt (Large bzw. Very Large Scale Integration). Damit läßt sich ein kompletter Prozessor auf einem Chip unterbringen – eine Schaltung aus Tausenden von Einzeltransistoren, deren logische Verknüpfung kaum noch vorstellbar ist. Solche Bauteile sind extrem leistungsfähig und lassen sich dadurch vielfältig einsetzen.

Nachdem wir die Grundlagen kennen, dürfen wir uns an komplexere Aufgaben herantrauen, insbesondere an die Konstruktion nützlicher Zusatzschaltungen für den Heimcomputer. Überprüfen Sie mit den Beispielen auf der nächsten Seite noch einmal Ihre Kenntnisse. Und scheuen Sie sich nicht, in Zweifelsfällen noch einmal nachzuschlagen!

1) Widerstände

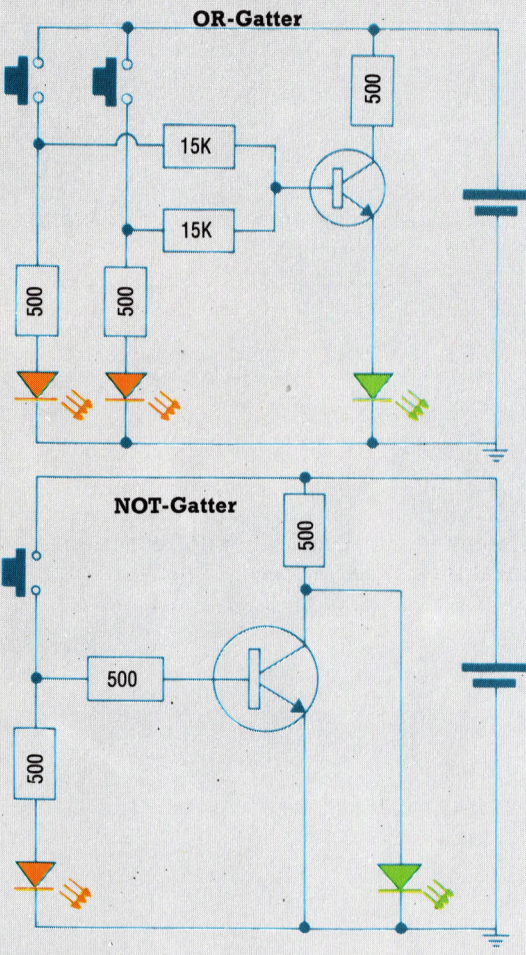
An den farbigen Ringen auf einem Widerstand können Sie seinen Wert ablesen. Wieviel Ohm haben die beiden hier abgebildeten Widerstände? Welche Farben würden die Ringe auf einem 150-Ohm-Widerstand haben?





2) NOR-Gatter

An früherer Stelle haben wir NOT-, OR- und AND-Gatter mit Transistoren aufgebaut. In dieser ersten Übung soll auf ähnliche Weise ein NOR-Gatter (NOT+OR) hergestellt werden. Als kleine Hilfe sind noch einmal die Schaltungen des OR- und des NOT-Gatters unten abgedruckt. Das Problem kann auf zwei verschiedene Arten gelöst werden. Die eine Lösung ist, ein OR- und ein NOT-Gatter zu kombinieren.



3) Binär-Dezimal-Wandler

Versuchen Sie, eine Schaltung zu bauen, die Dezimalziffern in Binärzahlen umwandelt. Damit es nicht zu schwierig wird, beschränken wir uns auf zweistellige Binärzahlen, also die Dezimalzahlen von 0 bis 3. Ihre Schaltung sollte vier Eingangstaster haben, die mit 0, 1, 2 und 3 bezeichnet sind. Beim Drücken eines Schalters sollten die entsprechenden binären Bits durch das Aufleuchten von LEDs angezeigt werden. Die Wahrheitstabelle für den Wandler sieht so aus:

NULL Taster	EINS Taster	ZWEI Taster	DREI Taster	Hohes Bit	Niedriges Bit
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

4) BCD oder nicht?

Der BCD-Code (Binär Codiertes Dezimalsystem) steht zwischen den reinen Binärzahlen und dem Dezimalsystem – jede einzelne Dezimalziffer wird in ihr binäres Äquivalent umgewandelt. Eine BCD-Zahl setzt sich dementsprechend aus Vier-Bit-Gruppen zusammen. Die Zahl 53 wird zu 01010011, wobei 0101 für die 5 und 0011 für die 3 steht. Jede zulässige BCD-Ziffer ist also eine Bit-Gruppe zwischen 0000 (dezimal 0) und 1001 (dezimal 9). Bit-Gruppen zwischen 1010 und 1111 sind im BCD-Code nicht zugelassen.

Bauen Sie eine Schaltung auf, die eine Eingabe auf BCD-Gültigkeit prüft. Die Schaltung braucht vier Eingangstaster von B0 bis B3. Damit wird die BCD-Zahl eingegeben. Zwei Ausgänge (eine grüne und eine rote LED) zeigen an, ob die Zahl dem BCD-Code entspricht (grün) oder nicht (rot). Hier die Wahrheitstabelle:

Entsprechende Dezimalzahl	B3	B2	B1	B0	Im BCD-Code gültig	Im BCD-Code ungültig
0	0	0	0	0	1	0
1	0	0	0	1	1	0
2	0	0	1	0	1	0
3	0	0	1	1	1	0
4	0	1	0	0	1	0
5	0	1	0	1	1	0
6	0	1	1	0	1	0
7	0	1	1	1	1	0
8	1	0	0	0	1	0
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	0	1
12	1	1	0	0	0	1
13	1	1	0	1	0	1
14	1	1	1	0	0	1
15	1	1	1	1	0	1

Aus dieser Tabelle können Sie ersehen, daß ein Signal für gültige BCD-Zahlen aus $B3 + B2 \cdot B1$ entsteht. Das Signal für eine ungültige Eingabe ergibt sich aus dem NOT von:

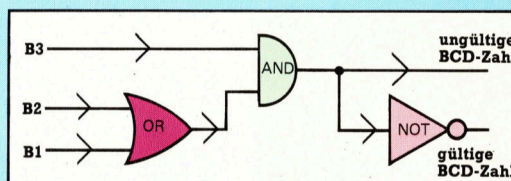
$$\overline{B3 + B2 \cdot B1}$$

das sich so vereinfachen läßt:

$$\overline{B3} \cdot \overline{B2 \cdot B1}$$

$$B3 \cdot (B2 + B1)$$

Die Schaltung zur Prüfung auf BCD-Gültigkeit ist recht einfach. Es werden nur drei Eingänge benutzt. Unten ein Vorschlag für den Logik-Aufbau des BCD-Testers:



Lösungen finden Sie im nächsten Abschnitt des Selbstbau-Kurses.

Schwarz auf Weiß

Fast jeder Computerbesitzer arbeitet mit einem Drucker, beispielsweise um Listings besser überprüfen zu können. Im Bürobereich ist dafür ein Schönschreibdrucker, zum Beispiel eine Typenradmaschine, unumgänglich. Für den Computerbenutzer, der nur sein Programmprotokoll haben will, gibt es preiswertere Geräte.

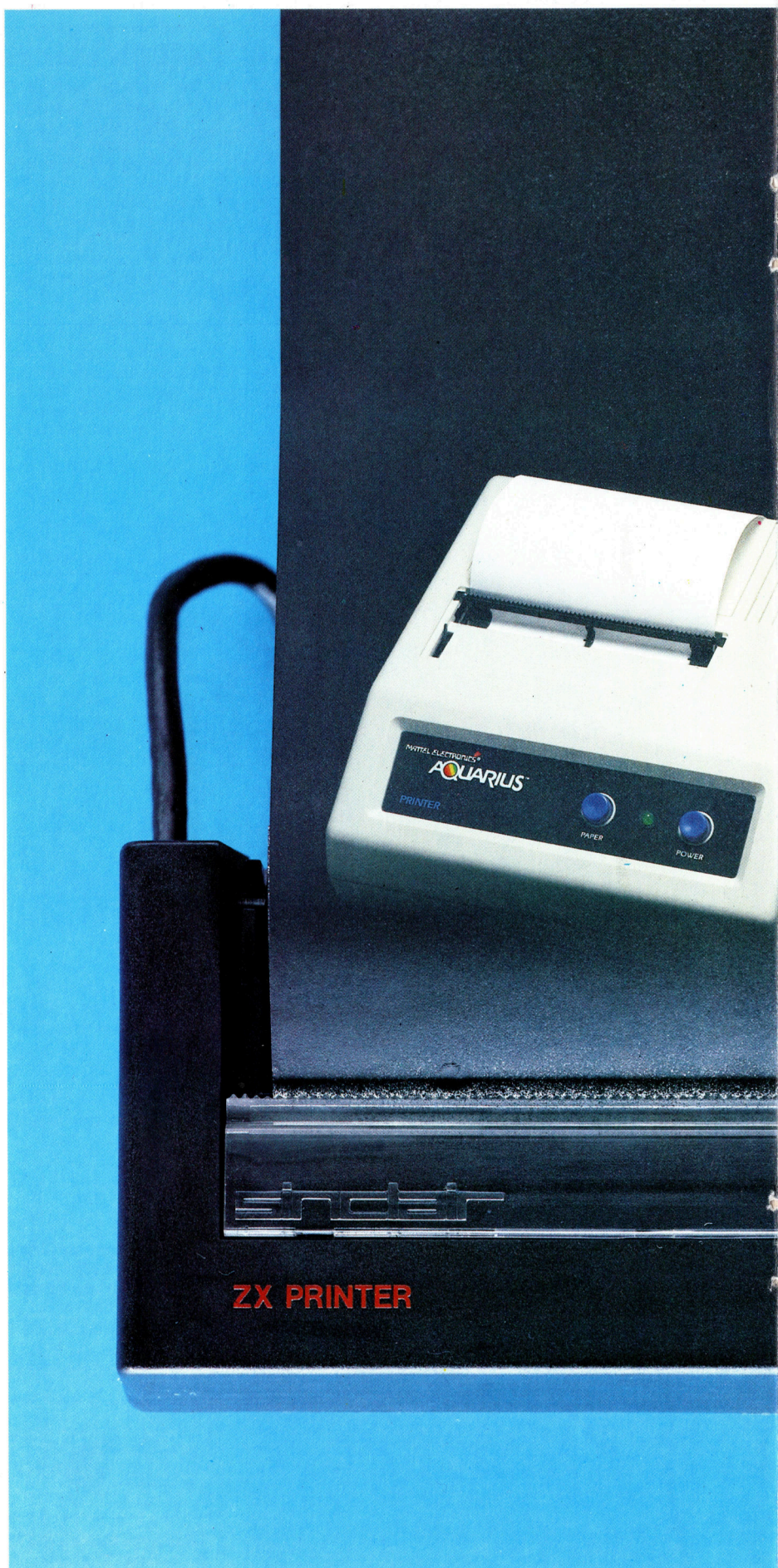
Geräte zum Ausdruck von Programmen oder anderen Texten stehen bei den meisten Computerbesitzern weit oben auf dem Wunschzettel. Leider können gute Matrixdrucker zwei- bis dreimal soviel wie der Rechner selbst kosten. Eine billige Alternative ist da ein anschlagfreier Drucker.

Die Bezeichnung „anschlagfrei“ rührt daher, daß diese Geräte ohne mechanischen Anschlag von Typen arbeiten. Beim Thermodruckverfahren erfolgt die Zeichendarstellung durch Wärmeeinwirkung auf Spezialpapier. Solche Systeme wurden zuerst für Registrierkassen und tragbare Terminals wie die der Texas-Silent-700-Serie entwickelt.

Der zeilenweise über das Papier bewegte Schreibkopf eines Thermodruckers enthält eine Anzahl von Heizstiften, die wie die Nadeln des Matrixdruckers vertikal untereinander angeordnet sind. Wo ein Punkt erscheinen soll, wird der entsprechende Stift rasch erhitzt, und die Papierstelle darunter verändert die Farbe. Der so erreichbare Kontrast genügt für den Ausdruck von Listings, aber bei Briefen stellt man im allgemeinen höhere Ansprüche. Weit mehr beeindruckt bei diesen Geräten der nahezu geräuschlose Betrieb.

Außer von Texas Instruments wurde das Thermodruckverfahren auch von Apple (für den kleinen „Leisedrucker“ Silentype) und in ähnlicher Form von Mattel als Zubehör für den Aquarius-Heimcomputer übernommen.

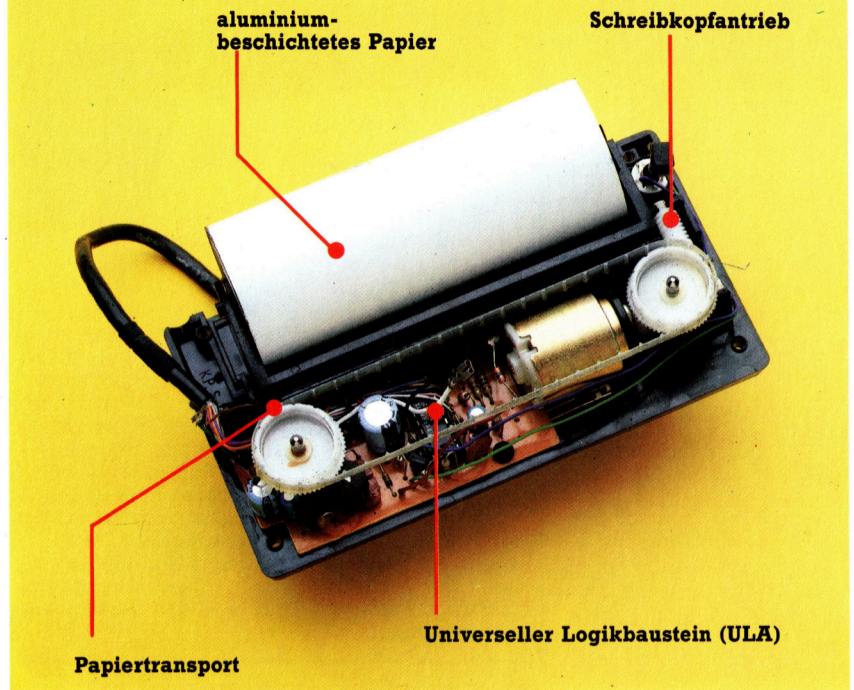
Ein anderes anschlagfreies Druckverfahren, nämlich das elektrostatische, arbeitet vergleichsweise laut. Es konnte sich, abgesehen von einem frühen Centronics-Produkt (Microprinter P1), kaum durchsetzen, bis Sir Clive Sinclair den elektrostatischen ZX-Drucker auf den Markt brachte. Dieser wird nun in beachtlichen Stückzahlen als Spezialzubehör für den ZX 81 und den Spectrum verkauft.



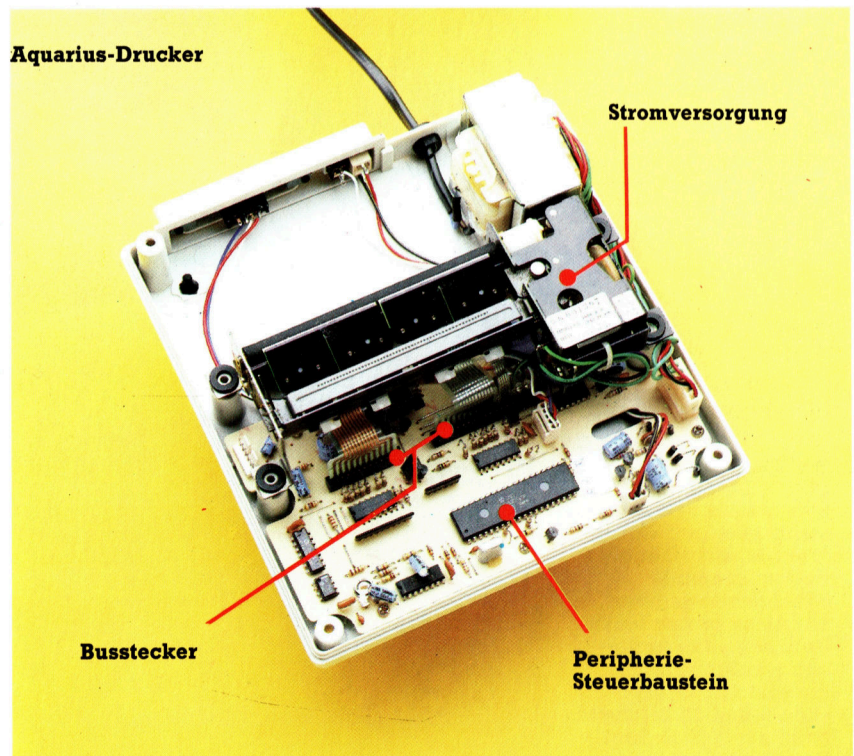
Obwohl für den ZX81 entwickelt, ist der ZX-Drucker Spectrum-kompatibel. Das Gerät stellt eine Billiglösung für den Heimcomputer-Benutzer dar, der Programmprotokolle ausdrucken will. Dem niedrigen Anschaffungspreis steht allerdings entgegen, daß Elektrostatik-Papier benutzt werden muß.

Beim Aquarius-Heimcomputer wurde das zweite preisgünstige Druckverfahren gewählt, nämlich der Thermodruck. Auch hierbei ist leider die Verwendung von Spezialpapier unumgänglich. Für den Aquarius gibt es außerdem einen Vierfarb-Printer, der mit Kugelschreiberpatronen arbeitet.

ZX-Drucker



Aquarius-Drucker





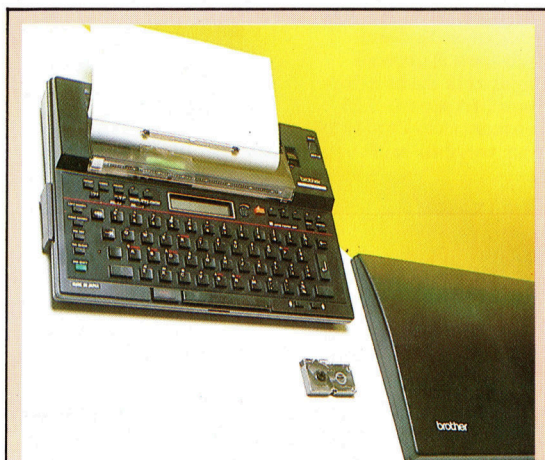
Bei elektrostatischen Systemen wird ein Kopf mit nur einem Schreibstift über ein spezialbeschichtetes Papier geführt. Für jeden Punkt eines Zeichens wird ein Funkenüberschlag ausgelöst, der ein Loch in die dünne

Metallbeschichtung einer Unterlage aus Schwarzpapier brennt. Sinclair verbesserte das Verfahren durch Verwendung von zwei Schreibköpfen auf einem Endlosband – dennoch bleiben für jede Textzeile acht Kopfdurchläufe erforderlich.

Der Hauptnachteil beider Druckverfahren liegt darin, daß teures Spezialpapier erforderlich ist. Bei Thermodruckern müssen Sie genau die richtige Papiersorte kaufen, sonst fehlt es an Kontrast. Die Schrift bleicht zudem mit der Zeit aus, besonders unter Wärmeeinwirkung. Elektrostatik-Papier ist noch empfindlicher. In Verbindung mit Feuchtigkeit löst sich die Beschichtung, und der Text wird unleserlich. Am besten machen Sie sich in jedem Fall gleich eine Fotokopie, wenn Sie eine zuverlässige und dauerhafte Dokumentation brauchen. Das silbrige Papier läßt sich gut kopieren.

Ungeachtet dieser Nachteile sind die anschlagfreien Systeme für die Hersteller preiswerter Heimcomputer der einzige Weg, um einigermaßen günstige Drucker anbieten zu können. Das Matrixverfahren ermöglicht außerdem eine punktgetreue Kopie der Bildschirmdarstellung, so daß die Grafikwiedergabe keinen zusätzlichen Aufwand erfordert.

In der Zwischenzeit haben die beschriebenen Systeme Konkurrenz in Gestalt der Vierfarbplotter von Tandy und Sharp bekommen, die präzise Textzeichen und hervorragende Liniengrafik auf normalem Rollenpapier liefern. Die anschlagfreien Geräte mit ihrer einfachen Mechanik sind zur Zeit trotz eingeschränkter Möglichkeiten eine gute Lösung für die Computer der unteren Preisklasse.



Brother EP-22

Dieses Gerät ist wohl die eleganteste Version eines Thermodruckers. Für einen günstigen Preis erhalten Sie gleichzeitig einen Drucker mit 75 Zeichen pro Zeile und eine Reiseschreibmaschine. Außer Spezial-Thermopapier kann in Verbindung mit einem Farbband auch Normalpapier verwendet werden. Das Gerät enthält einen batteriegepufferten Speicher für 2000 Zeichen. Außerdem ist eine LCD-Anzeige eingebaut, mit deren Hilfe innerhalb der letzten 16 Anschläge vor dem Ausdrucken Korrekturen möglich sind. Die Schriftqualität entspricht nicht der eines Schönschreibdruckers – u.a. fehlen echte Unterlängen – aber als tragbare Schreibmaschine mit Drucker ist das Gerät vielseitig einsetzbar.

Der Thermodrucker TP-10 von Tandy wird, anders als die beiden umseitig gezeigten Billigdrucker, mit einer seriellen Standard-schnittstelle geliefert, so daß er an fast jedem Heimcomputer betrieben werden kann. Er arbeitet mit 30 Zeichen/s und 32 Zeichen/Zeile auf 10,5 cm breitem Thermopapier.



Das ABC des Acorn B

Wir setzen unsere Betrachtungen über die verschiedenen BASIC-Versionen mit der des Acorn B fort.

Bei den meisten BASIC-Dialekten ist zu kritisieren, daß sie eine unstrukturierte Arbeitsweise unterstützen. Dies gilt besonders für schlecht durchdachte Problemlösungen, die beispielsweise zum unnötigen Einsatz von GOTO führen.

Durch die Verwendung von ELSE in Verbindung mit der IF...THEN-Anweisung läßt sich der Einsatz von GOTOs auf ein Minimum reduzieren, indem mit einem zutreffenden und einem nicht zutreffenden Vergleich in derselben Anweisung gearbeitet werden kann. Betrachten Sie die folgenden Programmzeilen:

```
1500 IF TEST>0 THEN GOTO 1800
1600 PRINT "WERT AUSSERHALB
      BEREICH"
1700 GOTO 1900
1800 PRINT "KEIN PROBLEM"
1900 NEXT L
```

Sie können ersetzt werden durch:

```
1500 IF TEST>0 THEN PRINT "KEIN
      PROBLEM" ELSE PRINT "WERT
      AUSSERHALB BEREICH"
1900 NEXT L
```

Für GOSUB muß normalerweise eine Zeilennummer als Argument verwendet werden, was zwei Nachteile mit sich bringt. Als erstes gibt beispielsweise GOSUB 1000 keinen weiteren Hinweis zum Inhalt der aufzurufenden Unteroutine. Zudem sind Änderungen aufgrund der Zeilennummerangabe sehr schwierig. GOSUB ist wie GOTO recht langsam in der Ausführung, da die spezifizierte Zeilennummer jedesmal im Programm neu gesucht werden muß. Auch beim Acorn-BASIC werden Unter Routinen eingesetzt, doch sie werden statt von Zeilennummern über Namen aufgerufen, so daß man erkennen kann, wofür welche Routine verwendet wird.

Routinen (procedures) und Funktionen beginnen mit DEF PROC oder DEF FN, gefolgt von einem Namen und gewöhnlich mit einer Parameter-Liste. Zum Beispiel:

```
1200 DEF FNcalc(a,b,c)=(a-b)*c/100 und
2500 DEF PROCoperate(w,x$,y$,z)
```

Die Definition verwendet diese Parameter so, als wären es Programmvariablen. Wenn das Programm die Funktion oder Routine aufruft,

können die Parameter trotzdem noch gegen andere Variablen oder Ausdrücke desselben Datentyps ausgetauscht werden. Beispiel:

```
250 ergebnis=FNcalc (preis,kosten,12)
```

oder

```
545 PROCoperate (6,name$,"meier",
      array(12))
```

Die Werte der Parameter werden dann anstelle der Variablen in der Definition verwendet. Beachten Sie, daß eine Funktion in einem Ausdruck verwendet werden kann, als wäre sie eine Variable oder ein arithmetischer Ausdruck. Eine Prozedur dagegen muß wie eine BASIC-Anweisung gehandhabt werden. Der Befehl LOCAL, der Variablen ausschließlich zur Verwendung innerhalb eines bestimmten Programmblocks definiert, verhindert die sonst üblichen Unter Routinen-Fehler. Betrachten Sie hierzu den folgenden Programmcode:

```
100 FOR K=1 TO 10:GOSUB 500:NEXT
      K:END
500 FOR K=1 TO 5:PRINT "*****":NEXT:
      RETURN
```

Hier wird die Variable K als Schleifenzähler in der Hauptprogrammzeile 100 verwendet und dann in der Unter routine in Zeile 500 noch einmal. Es ist wahrscheinlich, daß hierdurch die Programmausführung beeinträchtigt wird. Beim Acorn B ist dies jedoch vermeidbar:

```
100 FOR K=1 TO 10:PROCstars:NEXT:END
500 DEF PROCstars
520 LOCAL K
540 FOR K=1 TO 5:PRINT "*****":NEXT
560 ENDPROC
```

Der LOCAL-Befehl bewirkt, daß zwischen den Zeilen 500 und 560 die Variable K eine neue Variable ist, unabhängig von der Variablen K an einer anderen Stelle des Programms und ohne den Wert der anderen Variablen K zu beeinflussen. (Beachten Sie, daß PROCstars eine Procedure ohne Parameterangabe ist.)

REPEAT...UNTIL ist eine Schleifenstruktur, in der die entsprechenden Anweisungen so lange wiederholt werden, bis die Bedingung, die dem Wort UNTIL folgt, erfüllt ist.

```
200 DATA 12,234,31,45,65,0,76,81
250 REPEAT
300 READ nummer:summe=
      summe+nummer
350 UNTIL nummer=0
400 PRINT "Die Summe ist ";summe
```

Diese Form ist erheblich einfacher zu lesen

und viel weniger fehleranfällig als eine GOTO- oder FOR...NEXT-Schleife.

Das Acorn-BASIC beinhaltet die Kommandos TRACE, ON ERROR... und ERL, die bei der Fehlersuche sehr hilfreich sind. TRACE bewirkt, daß die gerade auszuführenden Programmzeilen auf dem Bildschirm dargestellt werden. Mit ON ERROR...GOTO (oder GO-SUB) kann man die Programmausführung bei jedem normalen Fehler (einschließlich Drück-

ken der ESC-Taste) zu einer von Ihnen vorgegebenen Routine verzweigen (z. B. eine Routine zum Ausdrucken aller Variableninhalte). ERL ist eine Systemvariable, die die Nummer der Zeile enthält, in der ein Fehler aufgetreten ist. Das BASIC des Acorn B enthält noch einige zusätzliche Funktionen, wie die Möglichkeit, Routinen des Betriebssystems aufzurufen, den Befehl VDU, die zahlreichen Systemvariablen und den Assembler.

Acorn-B-Grafik

Die Befehle des Acorn-B-BASIC, die sich direkt auf seine Grafik beziehen, sind:

MODE

Mit MODE N wird der Darstellungsmodus des Computers ausgewählt, wobei N die Werte von 0 bis 7 haben kann:

Modus	Grafik	Farben	Text
0	640 × 256	2	80 × 32
1	320 × 256	4	40 × 32
2	160 × 256	16	20 × 32
3	nur Text	2	80 × 25
4	320 × 256	2	40 × 32
5	160 × 256	4	20 × 32
6	nur Text	2	40 × 25
7	Teletext		40 × 25

In den Modi 0 und 6 kann der Zeichensatz vom Anwender mit dem Befehl VDU geändert werden. Die Teletext-Zeichen des Modus 7 sind nicht zu ändern und entsprechen nicht dem Standard-ASCII-Code.

COLOUR

Dieser Befehl setzt eine von 16 Farben für Text und Hintergrund, abhängig vom gewählten Modus.

COLOUR N

Dabei kann N Werte von 0 bis 15 für Textfarben und Werte von 128 bis 143 für die Hintergrundfarben annehmen. Die durch den Wert von N gesetzten Farben sind von Modus zu Modus verschieden.

VDU

VDU A entspricht der Anweisung PRINT CHR\$(A). Ebenso hat VDU A,B,C denselben Effekt wie PRINT CHR\$(A);CHR\$(B);CHR\$(C). So können die vielen komplexen Text- und Grafikroutinen mit Kontrolle über die 32 CHR\$-Codes, die die meisten auf Grafik bezogenen BASIC-Befehle simulieren, mit einer kleinen Anzahl von VDU-Befehlen konstruiert werden.

CLG

Mit diesem Befehl wird der Grafikbereich des Bildschirms gelöscht, und der Cursor wird in seine „Home“-Position in der unteren linken Ecke des Bildschirms gesetzt.

CLS

Dieser Befehl löscht den Textbereich des Bildschirms und setzt den Cursor in seine

„Home“-Position in der oberen linken Ecke. Die Grafik auf dem Bildschirm wird ebenfalls gelöscht.

DRAW

Hiermit werden in den Modi 0, 1, 2, 4 und 5 Linien auf den Bildschirm gezeichnet. Der Befehl hat folgende Form:

DRAW X,Y

Der Punkt, der durch die X- und Y-Koordinaten definiert wird, ist der Endpunkt der Linie. Der Startpunkt kann entweder der Endpunkt der zuletzt gezeichneten Linie sein oder aber ein durch den MOVE-Befehl definierter Punkt.

GCOL

Dieser Befehl setzt die gegenwärtigen Farben für den Grafikvorder- und -hintergrund.

GCOL N,M

Hierbei bestimmt N, wie die Farbe verwendet werden soll (Werte 0 bis 4), und M definiert die logische Farbe, wobei dieselben Prinzipien gelten wie für COLOUR. Die fünf Möglichkeiten von N sind:

- 0 — Verwende durch M spezifizierte Farbe
- 1 — OR M-Farbe mit gegenwärtiger Farbe
- 2 — AND M-Farbe mit gegenwärtiger Farbe
- 3 — EXCLUSIVE-OR M-Farbe mit gegenwärtiger Farbe
- 4 — Invertiere gegenwärtige Farbe

MOVE

Dieser Befehl positioniert den Grafik-Cursor an einem spezifizierten Punkt.

MOVE X,Y

Der Befehl hat denselben Effekt wie DRAW, nur daß keine Linie gezeichnet wird.

PLOT

Dieser Befehl kann für viele Grafik-Funktionen verwendet werden, einschließlich Punkt-, Linien- und Dreieckzeichnen.

PLOT K,X,Y

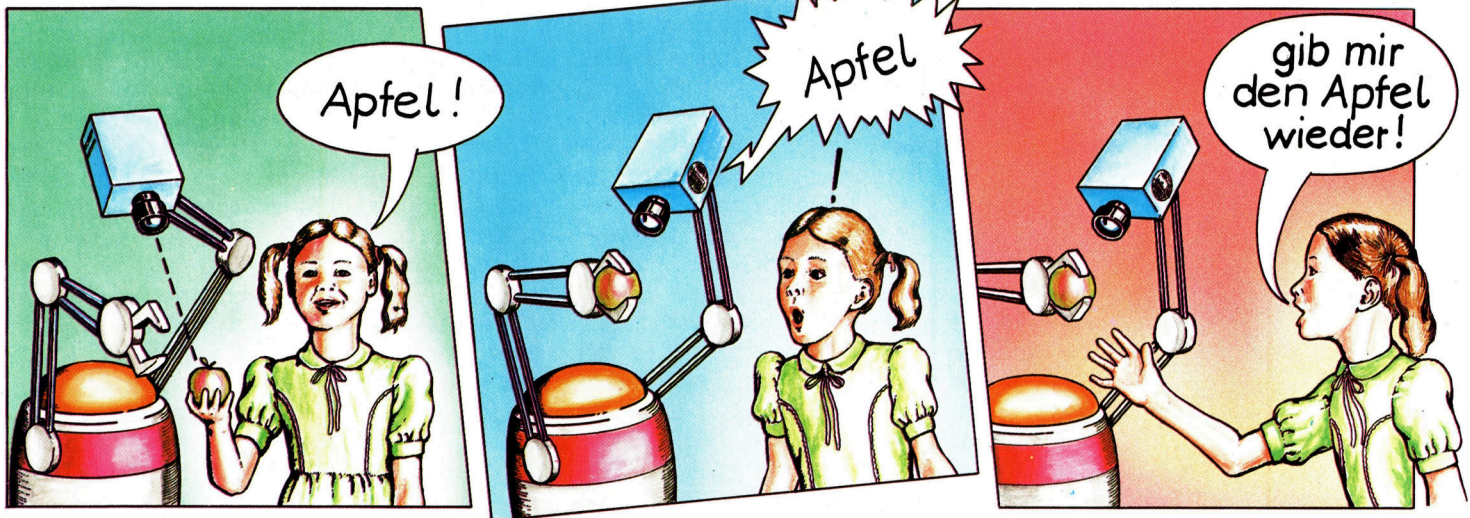
Hierbei definiert K den Typ der zu PLOTten- den Grafik. K kann Werte in einem Bereich von 0 bis 225 annehmen, womit der Typ der zu zeichnenden Linien und die entsprechenden Farben spezifiziert werden.

POINT

Dieser Befehl ermöglicht die Abfrage der logischen Bildschirmfarbe einer bestimmten Koordinate. Dies geschieht in der Form:

NUMVAR = POINT(X,Y)

NUMVAR vertritt eine numerische Variable.



Über die Sprache

Die Sprache ist eine der kompliziertesten Anforderungen, die an Roboter gestellt werden. Um einige der Probleme zu verstehen, die damit zusammenhängen, werden zunächst die wichtigen Theorien des Sprachlernprozesses dargelegt.

Aus dem Studium der menschlichen Sprache haben sich zwei Theorien herausgebildet. Die Vertreter der einen sind der Ansicht, daß Sprachfähigkeit angeboren sei. Die andere Gruppe ist der Auffassung, daß Sprechen eine erlernbare Fähigkeit ist. Die Psychologen der ersten Gruppe argumentieren, daß der Mensch das einzige Wesen ist, das mittels Sprache kommunizieren kann. Demgegenüber vertreten die Befürworter der anderen Gruppe ihre Auffassung aufgrund von Experimenten mit Tieren, daß diese durch eine erlernte Zeichensprache mit Menschen kommunizieren können.

Man hat bereits versucht, Computer zum Sprechen zu bringen. Dies geschah, indem man grammatikalische Satzstrukturen eingab. Bei anderen Experimenten ließ man Roboter neue Worte auf der Basis von Morphemen (kleinsten Sprachelementen) beliebiger Sprache lernen, indem man sie ihnen zeigte. Doch bis heute gibt es kein wirklich erfolgreiches Verfahren, einen Roboter sprechen zu lehren.

Daraus resultiert, daß die Sprechfähigkeit eines Roboters anders zu bewerkstelligen sein muß, da die Fähigkeit nun einmal angeboren sein muß und nicht erlernt werden kann. Folglich sind die Grundlagen in Programmform zu erarbeiten und im Roboter dauerhaft zu speichern, als sei dieser tatsächlich damit ge-

boren. Das beinhaltet zwei wichtige Elemente: syntaktische und semantische Analyse.

Die syntaktische Analyse betrifft die Grammatik des Gesagten. Sie decodiert die Struktur einer Nachricht oder bringt diese in eine grammatikalische Form, die der Roboter übersetzen kann. Die am weitesten verbreitete Methode ist die der „Verzweigung“. Das bedeutet: Ein Satz wird aus verschiedenen Einzelteilen aufgebaut bzw. in diese zergliedert. Keine leichte Aufgabe – doch immerhin ein Weg, der zuweilen erfolgreich ist.

Weit komplizierter ist die semantische Analyse, die voraussetzt, daß der Roboter den Sinn einer Nachricht versteht. Das Problem bei dieser Methode ist, daß es keine zusammenhängende Sprache gibt. Die Bedeutung hängt also immer vom gesprochenen Gesamtzusammenhang ab. Dieser Kontext schließt Wissen um die Welt an sich beim Sprechen ebenso ein wie die Kenntnisse darum, welches Wissen die Gesprächspartner über- und voneinander haben.

Dieses Phänomen hat der Computerwissenschaftler Terry Winograd experimentell nachzuvollziehen versucht. Er schrieb ein Programm, das einen Roboter befähigte, zu verstehen, was er sagte, und den Anweisungen zu folgen. Allerdings experimentierte Winograd mit der Computersimulation eines Roboters, der nur in einem sehr klein definierten Umfeld operieren konnte. Diese „Welt“ bestand aus einer Reihe von Bauklötzen, die umgestellt werden konnten. Das als SHRDLU bekannte Programm Winograds ließ zwar eine gute semantische Analyse zu, doch die Gestaltung der Umgebung durfte nur sehr einfach sein. Ein Roboter, der im Chaos der echten Umwelt agieren müßte, hätte ungeheure Schwierigkeiten, das ihm Gesagte zu verstehen.

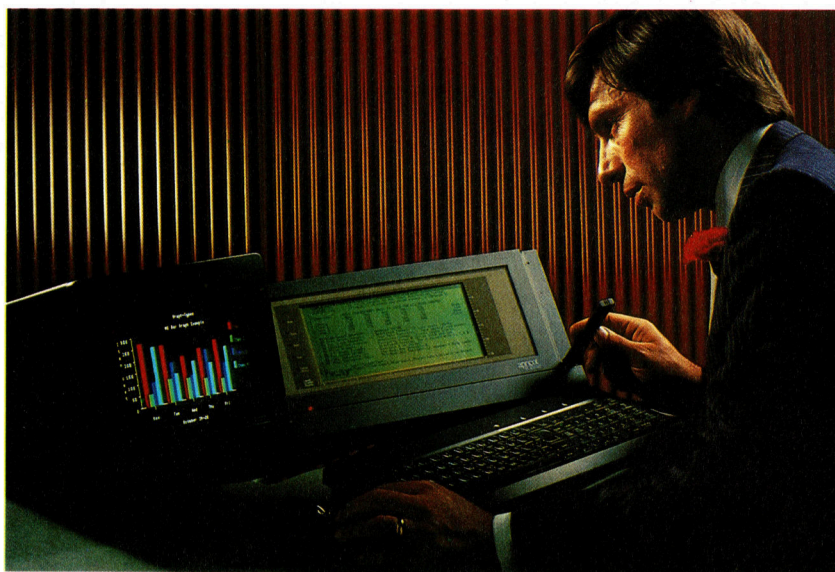
Eine Zeit lang glaubte man, daß die Sprach-

Sieht ein Mensch einen Gegenstand, etwa einen Apfel, und ordnet ihm einen Namen zu, so ist die Bedeutung des Begriffes „Apfel“ festgelegt. Der Roboter kann einen Gegenstand visuell erkennen, wenn er ihn mit seinem internen Vorrat an aufgezeichneten Mustern vergleicht und zudem das Klangmuster in Übereinstimmung bringt. Der Roboter weiß nicht, daß der Apfel eine essbare Frucht ist und, was noch wichtiger sein mag, daß der Apfel dem Menschen „gehört“.



eingabe bei Robotern durch Syntax-Analyse der Eingabe allein „verstanden“ werden könne und so der eigentliche Sinn der Nachricht verständlich sei. Neuere Forschungen aber verdeutlichten die Wichtigkeit des Wissens um das Umfeld und den Kontext der gesprochenen Nachricht. Daraus leitete man Experimente ab, bei denen zunächst eine Syntax-Analyse des sprachlichen Signals zur Erkennung zentraler Inhalte durchgeführt wurde. Im Rahmen dessen, was der Roboter über die Welt weiß, und der wenigen Dinge, die in dieser Welt gesagt werden könnten, überarbeitet der Roboter die erste Analyse, um das Gesagte nach und nach korrekt zu verstehen. Doch dies ist weit mehr, als einer der gegenwärtig erhältlichen Industrieroboter tun kann.

Apricot F1



Roboter- wie Computersprache sind vergleichsweise einfach zu erzeugen. Sprach-Synthese-Elemente wie etwa das hier von „Currah“ gibt es sogar für kleine Heimcomputer. Weitaus schwieriger ist die Spracherkennung, da der Mensch Klänge und Silben unterschiedlich betont. Für diese Erkennung wäre viel Speicherplatz erforderlich, schon um nur wenige Worte eindeutig erkennen zu können. Die Anbieter „Big Ears“ und „Apricot“ haben für ihre Systeme entsprechende Module geschaffen. Diese erlauben aber nur wenige Anwendungsmöglichkeiten.

Sehen wir uns daher einmal an, was ein „normales“ Robotersystem verstehen und sprechen kann.

Sprachsynthese

Bei der einfachsten Form der Spracherzeugung verwendet man ein Tonbandgerät. Die von einem Menschen gesprochene Nachricht ist auf Band aufgezeichnet und wird vom Roboter bei gegebenem Anlaß einfach abgespielt. Das mag zwar wenig mit dem zu tun haben, was man sich unter Sprachsynthese vorstellt, ist aber Grundlage für alle Sprachsynthese-Systeme. Betrachten wir die Grenzen dieses Verfahrens, und versuchen wir dann, es weitgehend zu verbessern.

Offensichtlich ist, daß ein Tonbandgerät Nachteile wegen seiner Mechanik, seines Preises und seiner Zerbrechlichkeit hat. Der erste Schritt wäre, die Nachricht in digitale Form zu bringen und auf einem Chip im Speicher des Roboters abzulegen. Dies erfolgt mittels eines Analog/Digital-Umwandlers, wobei für die Darstellung der sich ständig verändernden Wellenform der Sprache Zahlen verwen-

det werden. Genau diese Methode findet bei der digitalen Aufzeichnung von Musik, etwa bei Compact-Disc-Systemen, Anwendung.

Aber sie hat auch Nachteile. Eines der Hauptprobleme ist, daß das digitalisierte Signal viel Speicherplatz benötigt. Bei der Aufnahme von Compact Discs hat man 44 000 akustische Signale pro Sekunde mit einer Auflösung von etwa 16 Bits (Das bedeutet: Die Amplitude der Welle wird als 16-Bit-Zahl gespeichert, womit 2^{16} Ebenen erkannt werden können. $2^{16}=65\,536$). Bei Anwendung dieses Verfahrens würden pro Aufnahmesekunde 88 000 Bytes des Speichers belegt werden. Eine gesprochene Nachricht würde somit die Spei-



Big Ears

Currah

cherkapazität jedes Microcomputers bei weitem überschreiten. Dieser Wert gilt allerdings nur für die Klangwiedergabe in High Fidelity. Ein einfacheres Sprach-System könnte auf einer Acht-Bit-Basis bei einer Aufzeichnungsquote von 3000 Signalen pro Sekunde betrieben werden. Dafür wären nur drei KByte an Speicher erforderlich.

Um den Speicher weiter zu entlasten, sind ergänzende Einsparungen erforderlich. Sprachforscher stellten fest, daß die gesprochene Sprache in einzelne Elemente auflösbar ist. Insgesamt hat man 40 verschiedene dieser Phoneme für fast alle Sprachen herausgefunden. So wurde es möglich, die genaue akustische Information zur Beschreibung dieser vierzig Phoneme als Grundlage der Robotersprache einzusetzen. Üblicherweise wird die Phonem-Information auf einem „Speech-Synthesizer-Chip“ festgehalten. Der Roboter hat nichts weiter zu tun, als diese Phoneme für die gewünschte Nachricht miteinander zu verbinden. Die Nachricht ist als Phonem-String im Speicher des Computers abgelegt.



Die existierenden Sprach-Synthesizer lassen sich so programmieren, daß die vom Roboter zu sprechende Nachricht in einer phonetischen Englisch (bzw. Deutsch)-Version eingegeben wird. So würde etwa die Nachricht „Können Sie herkommen?“ als „Koinnaen See haircuomben?“ eingegeben. Das ermöglicht dem Synthesizer-Chip, den entsprechenden Klang zu erzeugen. Sprachwissenschaftler begnügen sich mit dieser Beschreibung der Phoneme nicht – sie haben ein eigenes Alphabet dafür –, doch ein Roboter ist damit zufrieden.

Hier sollte noch einmal betont werden, daß der Roboter nicht mehr mit einer vorher aufgezeichneten Nachricht arbeitet. Er ist in der Lage, eigene Nachrichten zu erzeugen. Da das möglich ist, kann der Roboter alles Gewünschte sagen, ohne daß die gesamte Botschaft vorher aufgezeichnet werden muß. Wir könnten jetzt also versuchen, einige Grammatikregeln einzugeben, um den Roboter selbständig etwas sagen zu lassen. Doch wie bereits erwähnt: Da die Menge der von einem Roboter gewünschten Äußerungen limitiert ist, gibt es keinen Grund für ein umfangreiches Vokabular, es sei denn, wir wollten wissenschaftlich experimentieren.

Wer je den Speech-Synthesizer eines Roboters gehört hat, weiß, daß diese Sprache nicht perfekt ist. Das hängt von zwei Faktoren ab. Die Form der von einem Menschen gesprochenen Phoneme steht in unmittelbarem Zusammenhang zwischen den vorangehenden und darauffolgenden Phonemen. Der Gesamtklang der menschlichen Stimme wird von der Bedeutung des Gesagten geprägt. „Setzen Sie sich!“ und „Setzen Sie sich!“ sind zwei gleich geschriebene Nachrichten, klingen aber völlig unterschiedlich, wenn die erste von einem höflichen Gastgeber gesprochen, die zweite von einem aufgebrachten Lehrer ausgestoßen wird. Man hat versucht, die Betonung bei der Sprachsynthese zu integrieren. Da der Roboter aber nicht weiß, was er wie sprechen soll, ist das problematisch.

Spracherkennung

Eines der grundlegenden Probleme bei der Entwicklung eines Spracherkennungssystems ist die Vielschichtigkeit der Dinge, die einem Roboter gesagt werden, und die Vielzahl der Möglichkeiten, sie auszudrücken. Eine Problemlösung bestünde darin, alles, was wir sagen können und der Roboter verstehen soll, auf Band aufzuzeichnen. Sobald wir sprächen, würde er intern das Band abhören und nach Ähnlichkeiten zum gesprochenen Wort suchen. Auf diesem Verfahren beruht das Spracherkennungsvermögen vieler Roboter. Dieses Wortgerüst-Verfahren ist sehr einfach. Beim Roboter-Unterricht wird ein Wort oder ein Satz mehrfach wiederholt, bis der lernende Roboter eine Art „Mittel“ aus dem Gesagten ge-

zogen hat. Soll ein Roboter nur wenige Dinge sagen und sie immer ähnlich aussprechen, empfiehlt sich diese Methode. Sie findet bei Robotern Anwendung, die auf einfache Befehle wie „vorwärts“, „rückwärts“ etc. zu reagieren haben.

Die Lösung ist aber vergleichsweise einfach. Das Verfahren wird als „diskrete (getrennte) Spracherkennung“ bezeichnet, da die einzelnen Worte voneinander unterschieden werden. Das heißt, daß zwischen den einzelnen Nachrichten Pausen liegen, in denen man nichts sagt.

Probleme durch Verschmelzung

Die eigentlichen Probleme kommen, wenn ein Roboter eine fortlaufende Unterhaltung analysieren soll. Versuchen Sie einmal, „Das ist ein schöner Sommertag“ zu sagen und dabei genau auf die Abfolge zu achten. Der Satz klingt durch das Verschmelzen der Silben und Worte wie „Dasistain schöner Somertag“.

Der Mensch löst das Problem dadurch, daß er beim Zuhören errahnt, was sein Gesprächspartner sagen will. Er kann die Nachricht ohne Probleme decodieren. Der Roboter steht vor dem Problem zu erkennen, was gesagt und was gemeint ist.

Sprachfähigkeit von Robotern setzt sich zunehmend durch, trotz der noch vorhandenen Probleme. Weitaus schwieriger ist die Spracherkennung. Der derzeitige Status ist etwa dem vergleichbar, was bei einem gut dressierten Hund stattfindet: Er „hört“ und „versteht“ einzelne Befehle. Ganze Sätze bleiben ihm jedoch vollständig unverständlich. Doch eine Lösung dieser Probleme wird zweifelsfrei schon bald gefunden sein.

Die „Voicemate“ ist ein stimmgesteuerter Roboterarm, der im Newcastle Polytechnic College für die Anwendung in Laboratorien sowie industrielle Zwecke entwickelt wurde.





Flagge zeigen

In diesem Artikel wird untersucht, wie mathematische Abläufe von der Systemarchitektur gesteuert werden. Außerdem sehen wir uns die Rolle der Statusregister bei der Addition genauer an.

In der Assemblersprache des Z80 und des 6502 ist ADC ein wichtiger Befehl. Das mnemotische Kürzel bedeutet „Addiere mit Übertrag“ (oder „Add with Carry“) und stellt eine der Hauptinstruktionen der Assemblerprogrammierung dar. Das Übertragsbit („carry bit“) erfüllt dabei eine wesentliche Funktion. Sehen wir uns die Addition zweier Zahlen im Akkumulator genauer an:

A7	=	10100111
+ 3E	=	+ 00111110
E5	=	11100101

Da der Akkumulator ein Acht-Bit-Register ist, dürfen die geladenen Zahlen und die Ergebnissumme nur Zahlen von je 8 Bit zwischen \$00 und \$FF sein. Das bedeutet jedoch nicht, daß Additionsvorgänge nur Ergebnisse unter \$100 erzeugen dürfen. Die folgende Addition überschreitet die Grenze des Akkumulators:

FF	=	11111111
+ FF	=	+ 11111111
1FE	=	11111110

Die Addition der beiden höchsten Zahlen, die ein Byte darstellen kann, ergibt eine Summe, die nur ein Neun-Bit-Akkumulator fassen könnte. Mit einem zusätzlichen Bit ließe sich also die größte Zahl, die im Akkumulator entstehen kann, problemlos anzeigen. Dieses zusätzliche Bit wird nur benötigt, wenn ein Übertrag vom höchstwertigen Bit des Akkumulators entsteht.

Das Extrabit wird Übertragsbit genannt. Es befindet sich in einem Acht-Bit-Register, das als „prozessor status register“ (PSR) bekannt ist. Dieses ist mit dem Akkumulator und der ALU so verbunden, daß die einzelnen Bits des PSR nach jeder Akkumulatoraktivität je nach Ergebnis gesetzt oder gelöscht werden. Der Inhalt der Statusregister kann zwar als einfache Zahl angesehen werden, wird gewöhnlich aber als eine Anordnung von acht binären Flags verstanden, deren individueller Zustand die Auswirkungen des letzten Vorgangs im Akkumulator anzeigt. (Ein Flag ist eine Variable, deren Inhalt den Zustand oder Wahrheitswert einer Bedingung wiedergibt und nicht als absoluter Zahlenwert angesehen wird. Ein Flag

zeigt normalerweise nur zwei Zustände an: gesetzt oder nicht gesetzt; an oder aus; 0 oder 1.)

Wenn im Akkumulator ein Vorgang stattfindet, der vom achten Bit aus einen Übertrag erzeugt, dann wird das Übertragsflag des PSR automatisch auf 1 gesetzt. Ein Vorgang, der keinen Übertrag erzeugt, setzt das Flag zurück auf 0. Das Übertragsflag wird jedoch nur angesprochen, wenn der Befehl einen Übertrag ermöglicht. Abläufe, bei denen der Akkumulator als Zwischenspeicher eines Lade- oder Speichervorgangs eingesetzt wird, wirken sich auf das Übertragsflag nicht aus.

Addition und Übertrag

Bei der Addition zweier Zahlen im Ein-Byte-Format weiß man normalerweise nicht im voraus, wie das Ergebnis aussehen wird. Da die Summe größer als \$FF sein kann, ist es ratsam, für das Ergebnis zwei Bytes bereitzustellen. Sehen wir uns die Beispiele nochmals an:

Hex-Zahlen	Übertrags-Flag	Binär-Zahlen
A7 =		10100111
+ 3E =		+ 00111110
00E5 =	0	11100101
Kein Übertrag		
FF =		11111111
+ FF =		+ 11111111
01FF =	1	11111110
Übertrag		

Das Additionsergebnis ist in beiden Fällen als Zwei-Byte-Zahl dargestellt. Im ersten Fall wurde das Übertragsflag auf 0 zurückgesetzt, da vom achten Bit der Summe aus kein Übertrag durchgeführt wurde. (Das Zwei-Byte-Ergebnis ist \$00E5, wobei das höherwertige Byte \$00 ist.) Im zweiten Fall ergab sich beim achten Bit jedoch ein Übertrag, und das Übertragsflag wurde gesetzt. Das höherwertige Byte des Ergebnisses beträgt daher \$01.

Um das richtige Ergebnis einer Addition zu erhalten, müssen wir den Inhalt des Akkumulators im niederwertigen Byte des Zwei-Byte-Speichers ablegen und das Übertragsflag als das höherwertige Byte des Ergebnisses speichern. Es gibt zwar keine direkte Instruktion für die Speicherung des Übertragsflags, jedoch berücksichtigt der Befehlsaufbau von



ADC diesen Vorgang automatisch. ADC bedeutet: „Addiere den Operanden des Befehls zum Inhalt des Akkumulators plus dem Wert des Übertragsbits, stelle das Ergebnis in den Akkumulator und den Übertrag in das Übertragsregister. Eine Addition läuft daher immer in zwei Stufen ab. In der ersten Stufe wird der aktuelle Status des Übertragsflags verarbeitet und in der zweiten Stufe wird das Flag neu gesetzt.

Da der Stand des Übertragsflags direkten Einfluß auf das Additionsergebnis hat, müssen wir schon vor Ablauf des Befehls den aktuellen Status des Übertragsflags berücksichtigen. Aus diesem Grund haben wir in der vorigen Folge die beiden Befehle CLC und AND A eingesetzt. CLC ist eine Instruktion des 6502 und bedeutet „Übertragsflag löschen“, während AND A auf dem Z80 eine logische AND-Verknüpfung des Akkumulators auslöst.

Nachdem die Berechnung abgeschlossen ist, muß der Inhalt des Flags gespeichert werden. Dies läßt sich durch die Addition der Konstanten \$00 auf das höherwertige Byte des Ergebnisses erreichen. Ist das Übertragsflag nicht gesetzt, hat dies keine Auswirkung. Ist es gesetzt, erscheint als Übertrag die 1.

Hier der schematische Ablauf unserer ersten Methode der Ein-Byte-Mathematik:

- 1) Übertragsflag löschen.
- 2) Akkumulator mit einer Zahl laden.
- 3) Die zweite Zahl addieren.
- 4) Den Inhalt des Akkumulators im niederwertigen Byte (LOBYTE) der Zwei-Byte-Adresse speichern.
- 5) Den Akkumulator mit dem Inhalt des höherwertigen Bytes (HIBYTE) laden.
- 6) Die Konstante \$00 addieren.
- 7) Den Inhalt des Akkumulators im höherwertigen Byte speichern.

In der Assemblersprache sieht die Routine folgendermaßen aus:

Beide Prozessoren		
Bezeichnung	Befehl	Operand
BYTE1	EQU	\$FF
BYTE2	EQU	\$FF
LOBYTE	EQU	\$A000
HIBYTE	EQU	\$A001
	ORG	\$A020

Z80		6502	
Op-code	Operand	Op-code	Operand
LD	A,\$00	LDA	#\$00
LD	(HIBYTE),A	STA	HIBYTE
AND	A	CLC	
LD	A,BYTE1	LDA	#BYTE1
ADC	A,BYTE2	ADC	#BYTE2
LD	(LOBYTE),A	STA	LOBYTE
LD	A,(HIBYTE)	LDA	HIBYTE
ADC	A,\$00	ADC	#\$00
LD	(HIBYTE),A	STA	HIBYTE
RET		RTS	

LOBYTE, HIBYTE und ORG enthalten nur Beispielwerte. Die ersten beiden Befehle laden \$00 in HIBYTE, um eventuell in dieser Adresse enthaltene Daten zu löschen. LOBYTE braucht nicht auf diese Weise behandelt zu werden, da dessen Inhalt automatisch von dem niederwertigen Byte des Ergebnisses überschrieben wird.

Übertragsflag und andere

Auch in diesem Beispiel kommen die unterschiedlichen Strukturen des Z80 und des 6502 zum Vorschein. Die mnemotischen Kürzel und die Kennzeichnung von Konstanten durch „#“ lassen die Bedeutung eines Befehls leicht erkennen. Die Version des Z80 ist nicht so einfach zu durchschauen, da das Kürzel LD für alle Lade- und Speichervorgänge des Akkumulators eingesetzt wird. Auch wird eine Konstante nicht durch „#“, sondern durch das Fehlen von Klammern um den Operanden angezeigt. LDA,BYTE1 bedeutet „Lade den Akkumulator mit der Konstanten BYTE1“, während LDA,(HIBYTE) „Lade den Akkumulator mit dem Inhalt der Speicherstelle HIBYTE“ ausführt. Da in der Assemblersprache alle Befehle im Hexcode eindeutig definiert sind, mag die Frage entstehen, wie der Assembler (oder Programmierer) entscheiden kann, welcher Opcode den gleichlautenden mnemotischen Kürzeln zugeordnet werden muß. Die Antwort dafür liegt in der Art der Adressierung, auf die wir in der nächsten Folge eingehen werden.

Außer dem Übertragsflag enthält das Statusregister des Prozessors noch andere Flags. Hier ein Überblick:

Z80 PSR:	S	Z		H		P/V	N	C
Bitnummer	7	6	5	4	3	2	1	0
MSB								LSB
6502 PSR:	S	V		B	D	I	Z	C

PSR BIT	Z80	6502	PSR BIT
7	(S)– Vorzeichen	(S)– Vorzeichen	7
6	(Z)– Null	(V)– Überlauf	6
5	ungenutzt	ungenutzt	5
4	(H)– Halbübertrag	(B)– Break	4
3	ungenutzt	(D)– BCD-Modus	3
2	(P/V)– Parität/Überlauf	(I)– Interrupt	2
1	(N)– Subtraktion	(Z)– Null	1
0	(C)– Übertrag	(C)– Übertrag	0

Im Augenblick sind für uns nur die Flags Übertrag, Vorzeichen und Null wichtig. Wir haben gesehen, daß das Übertragsflag nach einer Addition den Wert des Übertrags aus dem achten Bit des Akkumulators enthält. Das Vorzeichenflag ist immer eine Kopie des achten Bits (Bit 7 des Akkumulators). Das Nullflag wird auf 1 gesetzt, wenn der Inhalt des Akkumulators Null ist und auf 0 zurückgesetzt, wenn der Inhalt nicht Null ist.



Lösung der Assembleraufgaben

1) Die fertig assemblierten Programme werden im nebenstehenden Kasten wiedergegeben. Die Symbole BYTE1 und BYTE2 werden für Konstante und als symbolische Adressen eingesetzt. Als Adressen müssen sie bei der Assemblierung allerdings im Zwei-Byte-Format stehen.

2) Der Befehl „Rückkehren aus einem Unterprogramm“ fehlt am Ende beider Programme. Der vollständige Code des 6502 muß folgende Zeile enthalten:

A00D 60 RTS

und die Z80 Version:

A00D C9 RET

3) Der Wert \$45 wird zunächst als Konstante in den Akkumulator geladen, dann wird \$45 dazu addiert, so daß der Akkumulator den Wert \$8A enthält. Das Ergebnis wird in der RAM-Adresse \$0045 gespeichert. Der Wert \$38 wird dann als Konstante zum Akkumulator addiert, so daß dort der Wert \$C2 (\$45 + \$45 + \$38) enthalten ist. Schließlich wird dieses Gesamtergebnis in der RAM-Adresse \$0038 untergebracht.

4) „Konstante“ sind Daten, die in einem Befehl selbst gespeichert sind. In den Instruktionen der Beispielpprogramme (z. B. LDA #\$9C und LD A,\$E4) sind die Werte \$9C und \$E4 Daten, die in den Akkumulator geladen werden. Sie sind als Operanden in den Befehlen gespeichert und stellen den Inhalt der Bytes dar, die dem Op-code jeweils unmittelbar folgen. Daten, die nicht als Konstante in den Op-code integriert werden können, werden in anderen Bereichen des Speichers untergebracht.

5) Der Wert von BYTE1 ist \$45. In der korrekten Schreibweise zeigt es die Speicheradresse \$0045 an. Die Adresse befindet sich also auf der Speicherseite 0.

Übung

Wenn wir den Inhalt des Prozessor Statusregisters (PSR) untersuchen wollen, ist es ratsam, diese Zahl im Binärformat und nicht als Hexbyte anzuzeigen. Die folgenden Programmzeilen für den Spectrum sind eine Unteroutine zur Umwandlung von dezimalen in binäre Zahlen. Setzen Sie diese Zeilen in das Monitorprogramm ein.

```
7000 RGM ***** BINAER BYTE U/R *****
7001 REM * UMWANDLUNG DER ZAHL N (<256) *
7002 REM * IN EINE 8-ZEICHEN-BINAER- *
7003 REM * DARSTELLUNG IN B$
7010 B$=""
7020 FOR D=8 TO 1 STEP-1
7030 LET N1=INT(N/2)
7040 LET R=N-2*N1
7050 LET B$=STR$(R)+B$
7060 LET N=N1
7070 NEXT D
7080 RETURN
```

BASIC-Dialekte

Auf dem Commodore 64 muß die Zeile 7050 folgendermaßen geändert werden:
7050 B\$=MID\$(STR\$(R),2)+B\$

Speicher- adresse	Maschinen- code	Assembler- befehle		
6502				
0000		START	EQU	\$A000
0000		BYTE1	EQU	\$45
0000		BYTE2	EQU	\$38
0000			ORG	START
A000	A9 45		LDA	#BYTE1
A002	18		CLC	
A003	69 45		ADC	#BYTE1
A005	8D 45 00		STA	BYTE1
A008	69 38		ADC	#BYTE2
A00A	8D 38 00		STA	BYTE2
Z80				
0000		START	EQU	\$A000
0000		BYTE1	EQU	\$45
0000		BYTE2	EQU	\$38
0000			ORG	START
A000	3E 45		LD	A,BYTE1
A002	A7		AND	A
A003	CE 45		ADC	A,BYTE1
A005	32 45 00		LD	(BYTE1),A
A008	CE 38		ADC	A,BYTE2
A00A	32 38 00		LD	(BYTE2),A

Befehlsstruktur

Mnemotisches
Kürzel

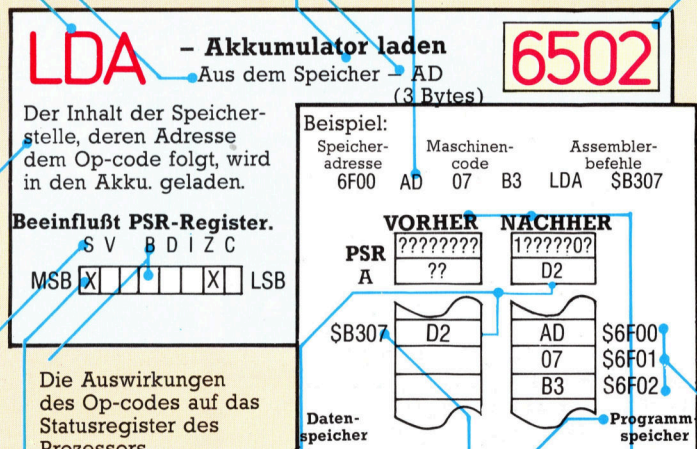
Mnemotische Kürzel
werden oft unter-
schiedlich angewandt.

Op-code in Hexdarstellung

Anzahl der Bytes des
vollständigen Befehls.

X=gesetzt/
rückgesetzt
?=nicht definiert
Prozessor Z80
oder 6502.

Ein assembliertes Bei-
spiel für den Ge-
brauch des Op-codes.





LDA - Akkumulator laden 6502

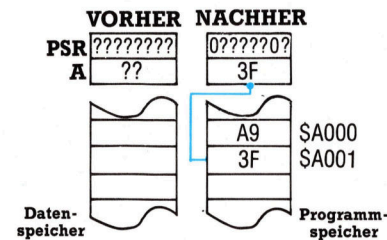
Konstante - A9 (2 Bytes)

Der Inhalt des Bytes, das dem Op-code folgt, wird in den Akku. geladen.

Bewirkt beim PSR:

SV BDI ZC
MSB [X] [] [] [] [] [] [X] LSB

Beispiel:
Speicher-
adresse A000
Maschinen-
code A9 3F
Assemblerbefehle LDA #3F



LD A, - Akkumulator laden Z80

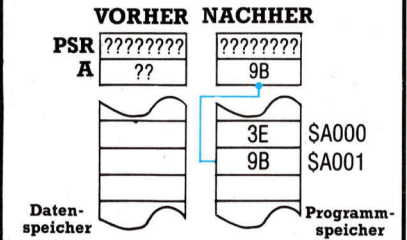
Konstante - 3E (2Bytes)

Der Inhalt des Bytes, das dem Op-code folgt, wird in den Akku. geladen.

Bewirkt beim PSR:

SZ H V NC
MSB [] [] [] [] [] [] [] [] LSB
Keine Wirkung

Beispiel:
Speicher-
adresse A000
Maschinen-
code 3E 9B
Assemblerbefehle LD A,9B



LDA Akkumulator laden 6502

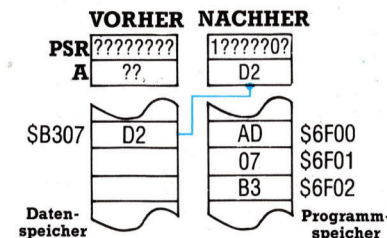
vom Speicher - AD (3 Bytes)

Der Inhalt der Speicher-
stelle, deren Adresse
dem Op-code folgt, wird
in den Akku. geladen.

Bewirkt beim PSR:

SV BDI ZC
MSB [X] [] [] [] [] [] [X] LSB

Beispiel:
Speicher-
adresse 6F00
Maschinen-
code AD 07 B3
Assemblerbefehle LDA \$B307



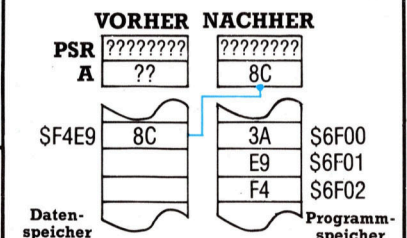
LD A, - Akkumulator laden Z80

vom Speicher - 3A (3 Bytes)

Der Inhalt der Spei-
cherstelle, deren
Adresse dem Op-code
folgt, wird in den
Akkumulator geladen.

SZ H V NC
MSB [] [] [] [] [] [] [] [] LSB
Keine Wirkung

Beispiel:
Speicher-
adresse 6F00
Maschinen-
code 3A E9 F4
Assemblerbefehle LD A,(\$F4E9)



STA - Akkumulator speichern 6502

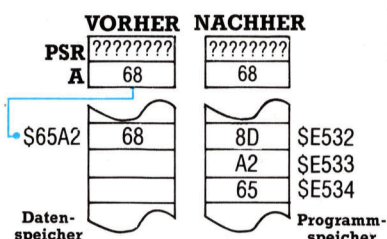
in den Speicher - 8D (3 Bytes)

Der Inhalt des Akku. wird
in die Speicherstelle
geladen, deren Adresse
dem Op-code folgt.

Bewirkt beim PSR

SV BDI ZC
MSB [] [] [] [] [] [] [] [] LSB
Keine Wirkung

Beispiel:
Speicher-
adresse E532
Maschinen-
code 8D A2 65
Assemblerbefehle STA \$65A2



LD(),A Akkumulator laden Z80

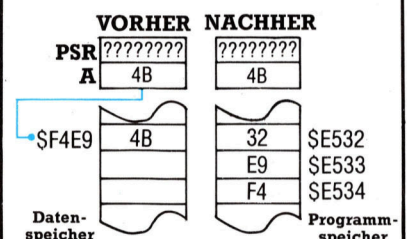
in den Speicher - 32 (3 Bytes)

Der Inhalt des Akku.
wird in die Speicherstelle
geladen, deren Adresse
dem Op-code folgt.

Bewirkt beim PSR:

SZ H V NC
MSB [] [] [] [] [] [] [] [] LSB
Keine Wirkung

Beispiel:
Speicher-
adresse E532
Maschinen-
code 32 E9 F4
Assemblerbefehle LD (\$F4E9),A



ADC - Addieren mit Übertrag 6502

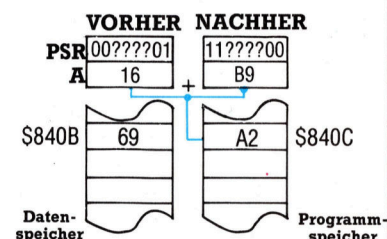
Konstante - 69 (2 Bytes)

Das Übertragsflag und
der Inhalt des Bytes, das
dem Op-code folgt, wer-
den zu dem Inhalt des
Akk. addiert.

Bewirkt beim PSR

SV BDI ZC
MSB [X] [X] [] [] [] [] [X] [X] LSB

Beispiel:
Speicher-
adresse 840B
Maschinen-
code 69 A2
Assemblerbefehle ADC #A2



ADC A, - Addieren mit Übertrag Z80

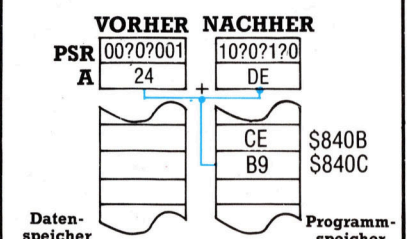
Konstante - 69 (2 Bytes)

Das Übertragsflag und
der Inhalt des Bytes, das
dem Op-code folgt, wer-
den zu dem Inhalt des
Akk. addiert.

Bewirkt beim PSR:

SZ H V NC
MSB [X] [X] [X] [X] [X] [X] [X] [X] LSB

Beispiel:
Speicher-
adresse 840B
Maschinen-
code CE B9
Assemblerbefehle ADC A,B9





Sommerspiele

Seit der Olympiade 1984 in Los Angeles sind viele Programme aufgetaucht, die sich mit dem Leistungssport beschäftigen. Nur wenige dieser Programme stammen aus Europa, der Löwenanteil hat amerikanische „Väter“. Auch beim hier vorgestellten „Summer Games“ handelt es sich um ein USA-Produkt.

Die „Summer Games“ sind eine Entwicklung des Softwarehauses Epyx. Acht verschiedene sportliche Wettkämpfe können mit diesem Programm ausgetragen werden. An jedem Wettbewerb können bis zu acht „Sportler“ teilnehmen, die die Wahl haben, für welches der 17 Länder sie starten wollen.

Nach dem Laden des Spiels erscheinen als erstes Bild die Eröffnungsfeierlichkeiten: Ein Läufer entzündet mit der Fackel das olympische Feuer, Friedenstauben werden bei feierlicher Musikuntermalung freigelassen. Die gute Ton- und Grafikqualität des Spiels wird bereits hier sichtbar: Die fließenden Bewegungen von Läufer und Tauben vor einem mit hochauflösender Grafik fein gezeichneten Hintergrund zeigen ein weiteres Mal die hervorragenden Möglichkeiten der Sprite-Grafik.

Der Spieler hat die Wahl, den Gesamt-Wettbewerb zu bestreiten, nur an einzelnen Ausscheidungen teilzunehmen oder eine Liste der Weltrekorde einzusehen. Zusätzlich können zwei Spieler im Schwimm- und Laufwettkampf direkt gegeneinander antreten, ohne daß der Rechner als Schiedsrichter eingreift.

Erster – und schwierigster – Wettkampf ist der Stabhochsprung. Die Mitspieler versuchen abwechselnd ihr Glück, wobei sie vom Computer Anweisungen erhalten. Zunächst wird die Latte auf die niedrigste Höhe (4 Meter) gelegt. Will der Spieler den Sprung wagen, fragt das Programm, wo er den Stab halten möchte. Erst

danach läuft der Bildschirm-Sportler los.

Beim Absprung wird durch Zurückziehen des Joysticks der Stab aufgesetzt, durch Drücken des Feuer-Knopfes im richtigen Moment löst sich der Springer vom Stab. Wenn es richtig gemacht wird, kippt die Hochsprunghilfe dann auch zur richtigen Seite und läßt die Latte an ihrem Platz ...

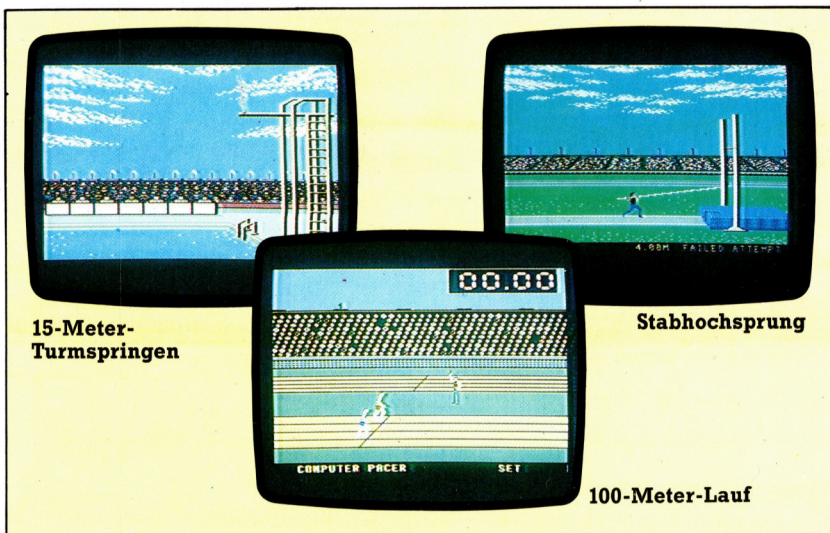
Wirklichkeitsgetreue Abläufe

Der Ablauf des Sprunges verdeutlicht die Vielzahl der vom Rechner zu überwachenden Funktionen: Neben der Steuerung des Bildschirms für einen flüssigen Bewegungsablauf sind auch die Joystick-Eingaben zu berücksichtigen. Zudem sollen sowohl der Spieler als auch der Hochsprung-Stab am richtigen Platz und im korrekten Winkel gezeigt werden. Und all dies soll nicht nur schnell, sondern auch ohne „Denkpausen“ des Computers vor sich gehen.

Nach jedem Wettkampf erscheint eine Anzeigetafel mit den gewonnenen Medaillen – sogar die Nationalhymne des siegreichen Landes wird gespielt. Wegen des großen Speicherbedarfs lädt der Rechner daran anschließend den nächsten Wettkampf – die umfangreichen Programme haben nicht alle gleichzeitig im Speicher Platz.

Die nächsten Disziplinen sind Turmspringen und Bodenturnen. Dabei muß der Spieler mit dem Joystick seinen Sprung vom Turm bzw. seine gymnastischen Kunststücke steuern, wobei er in vorgeschriebener Haltung ins Wasser eintauchen bzw. auf der Matte landen sollte. Der Computer ist die unbeeinflussbare Jury und ahndet jeden Fehler mit einem Punktabzug. Nach der Siegerehrung gibt's nur eine kurze Pause – der nächste Wettkampf folgt.

Diese Fotos der C64-Version zeigen recht deutlich die hohe Qualität von „Summer Games“. Jeder Wettkampf muß einzeln von der Cassette bzw. Diskette geladen werden. Ausgefeilte Programmierungstechnik sorgt für eine hohe Bildauflösung und eine sehr exakte Wiedergabe der Bewegungsabläufe.



Summer Games: Für Apple, Atari und Commodore 64. Auf Diskette und Cassette.

Hersteller: Epyx Software
Vertrieb: Fachhandel

Autoren: Randy Glover, Stephen Landrum, John Leupp, Brian McGhie, Stephen Mundry, Erin Murphy, Scott Nelson

Joysticks: Erforderlich



Spiel um Millionen

Wie so viele andere Heimcomputer-Unternehmen war auch Atari das geistige Kind eines Einzelnen. Nolan Bushnell konnte jedoch wohl kaum voraussehen, welche Konsequenzen es haben würde, als er einen Microprozessor-gesteuerten Regler mit einem Fernseher verband und das Spiel „Pong“ erfand.

Nolan Bushnells Erfolgsrezept gründete sich darauf, daß er dem Zuschauer eine Kontrollmöglichkeit über die Ereignisse auf dem Bildschirm gab. Er setzte somit den Bedarf an aktiver Fernsehfreizeit in die Wirklichkeit um und förderte die Phantasie von Millionen Jugendlicher. (Das Taschengeld „förderte“ er natürlich auch – aus den Taschen.)

Bushnell und seine beiden Partner Ted Dabney und Larry Bryan investierten umgerechnet jeweils etwa 400 Mark, um „Pong“ zu starten. 1972 wurde das Spiel im kalifornischen Sunnyvale vorgestellt und erwies sich bald als profitables Geschäft. Ataris Dominanz im Heimvideospiel-Markt resultierte aus einer wenig später getroffenen Entscheidung, Nutzungsrechte für Bushnells Erfindung zu erwerben.

Bevor sich der Publikumsgeschmack von den Videoautomaten auf die Heimcomputer-spiele verlagerte, war Atari Marktführer in diesem Bereich. Das war in den siebziger Jahren. Das Vermarkten von Spielen ist identisch mit dem von Schallplatten: Die potentiellen Stars müssen im Mittelpunkt stehen und beworben werden. So lag es nahe, daß Atari vom Medien-Multi Warner Communications International, bekannt für seine Schallplatten- und Film-Aktivitäten, übernommen wurde. Mit dem Videoautomaten-Geschäft machte Atari Ende der siebziger Jahre noch satte Gewinne. Doch 1983 verzeichnete das Unternehmen bereits Einbußen von über 25% Umsatz und besicherte der Muttergesellschaft beachtliche Verluste.

Das wohl bekannteste aller Computerspiele, „Space Invaders“ (von Taito), wurde von Atari erworben und erfolgreich vermarktet. Es entwickelte sich zu einem Phänomen und eröffnete ein ganzes Universum intergalaktischer Ballerspiele. In der Zeit des Videoautomaten-booms Ende der siebziger Jahre drehte sich alles um Atari. Ein Renner folgte dem anderen: „Asteroids“, „Battlezone“, „Centipede“, „Lunar Lander“, „Missile Command“ und „Tempest“.

Doch so dramatisch wie der Videoautomatenboom begonnen hatte, endete er auch wieder. Die Kunden wandten sich den Heimcomputern zu, weil sie zwei wesentliche Vorteile boten: Man konnte dieselben Spiele spielen, die es in den Spielhallen gab, doch das Spielen war preiswerter. Und außerdem verfügte man über einen vielseitigen Computer.

Ataris erste Antwort auf diese Marktveränderung bestand darin, die Arcadenhits in Heimcomputer-Variationen herauszubringen. Dabei verwendete man festprogrammierte Steckmodule. Diese Art der Computer-Software erwies sich als sinnvoll, da der Spieler das Programm nicht erst von Diskette oder Cassette laden mußte. Die Festprogrammierung ist jedoch recht teuer. Da die Cartridges nicht frei programmierbar waren, saß das Unternehmen häufig auf Bergen elektronischen Abfalls, bestehend aus unpopulären Spielen.

Sinkendes Glück

Ataris Marketing-Entscheidungen erwiesen sich bald als mangelhaft. Die Verkaufszahlenhochrechnungen basierten auf so erfolgreichen Spielen wie „Pac-Man“. Der Preis für die



Ataris Erfolg war auf die Inspiration eines Mannes zurückzuführen – Nolan Bushnell. Als Bushnell 1971 das erste Computerspiel (Pong) erfand, mag er wohl kaum die Tragweite seiner Schöpfung erkannt haben.

Fehleinschätzung des Marktes war hoch. Anläßlich einer Inventur wurden die nicht verkauften Steckmodule (Verkaufspreis zwischen acht und 25 Dollar) auf 14 Lkw verfrachtet und in ein großes Loch in der Wüste von Nevada gekippt.

Atari verpaßte auch eine andere Gelegenheit, die Besonderheiten von Computerprogrammen zu nutzen: Computer-Codierungen müssen nicht unbedingt auf Datenträgern in den Verkauf gebracht werden. Man kann Pro-



Das Atari Videospiel-System (VCS) einschließlich zweier Joysticks, Netzteil und Spielcassette eignet sich nur zum Spielen. Es kann nicht als Computer verwendet werden. Die Programme sind als Steckmodule erhältlich.



gramme mittels Telefon oder über Kabel, über Rundfunk und Fernsehen übermitteln. Neue Techniken und Produkte standen dafür zur Verfügung. So stellte beispielsweise 1983 die Romox Corporation in den USA eine Maschine mit der Bezeichnung „Romox Programming Terminal“ vor. Diese 15-Megabyte-Hard-Disk-Maschine konnte durchs Telefonnetz Programme überspielen.

Alternativ zu dieser Methode wurde „Gameline“ entwickelt, das Bill van Meister in den USA vorstellte. Gameline bot ein Modem für das Atari VCS an, mit dem Heimcomputer ans Telefonnetz angeschlossen werden konnten. Die Spiele wurden mittels telefonischer Übertragung eingelesen, wobei für 45 Minuten Spielzeit ein Dollar zu zahlen war.

In den siebziger Jahren verdiente Atari Unsummen mit Videoautomaten dieser Art. Nach Auftauchen der Heimcomputer mußte die Marketing-Strategie des Unternehmens jedoch völlig neu überdacht werden.



Zwei der größten amerikanischen Datenbanken, „Compuserve“ und „The Source“, bieten Spielprogramme als Bestandteil ihres Service an. Auch hier erfolgt die Programmübermittlung mittels Modem und Telefonnetz an bzw. in den Computer. „Coleco“, vor Jahresfrist noch Hersteller von Videospielsystemen, hatte ein entsprechendes Abkommen mit AT&T (American Telephone and Telegraph) geschlossen, um einen solchen interaktiven Service bieten zu können. Vor der Übernahme Ataris durch Jack Tramiel hatte das Unternehmen einen Nutzungsvertrag mit Activision vereinbart und außerdem ein vergleichbares System vorbereitet, hausintern als Ataritel bezeichnet, zu dem Computerbesitzer über die Warner Amex Cable Communications Zugriff haben sollten.

Ataris Probleme lagen vornehmlich darin, daß der Videospielmarkt zurückging. Das hat man durch Erweiterung des Heimcomputermarktes auszugleichen versucht. Die Geräte zeichnen sich durch überdurchschnittliche Grafikeigenschaften und leicht anwendbare Software aus. Drei Chips sind Kernstücke der Atari-Rechner: Pokey, Antic und GTIA, die die Ein-/Ausgabe, Grafik und Farbe steuern.

Alle Atari-Rechner der bisherigen Generation basieren auf dem 6502-Microprozessor. Für diese Systeme steht eine große Software-Bibliothek zur Verfügung, darunter VisiCalc, Atarischreiber und eine Finanzverwaltung für den Hausgebrauch. In den USA gibt es außerdem eine Z80-Softcard, womit Digital Researchs CP/M auch auf einem Atari-Computer laufen kann.

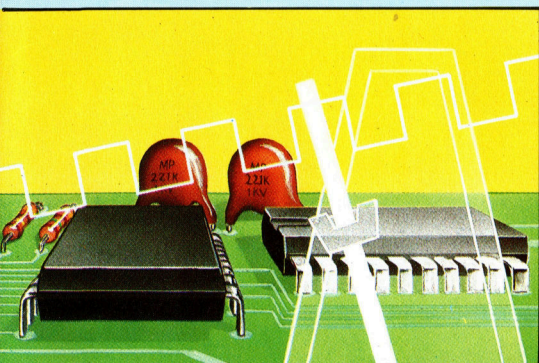
Im vergangenen Jahr setzte das Unternehmen seinen Schwerpunkt auf Software. Für England wurde eigens ein Software-Unternehmen verpflichtet, um die hauseigenen Programme auf andere Systeme umzusetzen und zu vermarkten. Dabei ging es vornehmlich um Commodore-Rechner. Wichtiger aber: Man suchte nach jungen englischen Programmierern, um deren Talente zu fördern und für eigene Entwicklungen zu nutzen. Aus heutiger Sicht betrachtet, könnte Atari den Wiederaufstieg schaffen.

Fachwörter von A bis Z

Clock = Taktgeber

Der Taktgeber im Computer hat die undankbare Aufgabe, den Rechner langsamer zu machen. Die Befehlsausführung erfolgt in der CPU in elementaren Einzelschritten durch Transistor-Schaltungen. Dabei erfordert jede Operation eine gewisse Schaltzeit, um die unterschiedlichen Transistorzustände umzusetzen.

Die Verarbeitungszeiten variieren bei den verschiedenen Gattern, und mit wachsender Anzahl der beteiligten Logikstufen kann der Datenfluß außer Takt geraten. Wenn bei einem Prozessor, der acht Bit parallel verarbeitet, das letzte Bit an einem Ad-dierer-Eingang erst nach der Sum-mation der übrigen erscheint, ist das Ergebnis natürlich wertlos.



So wie ein Metronom dem Musiker gibt die Clock der Zentraleinheit des Rechners den Takt an.

Die ganze Rechnerlogik einschließlich der Speichersteuerung muß daher durch einen zentralen Taktgeber synchronisiert werden. Erst der Taktimpuls veranlaßt die Informationsübernahme durch die nächste Stufe. Die Taktintervalle müssen länger sein als die Ansprechzeit des langsamsten Schaltkreises.

Als Taktgeber dient ein Quarzkristall, der mit einer bestimmten Frequenz von einigen Megahertz „schwingt“. Mikroprozessoren arbeiten vielfach mit Taktfrequenzen von ein, zwei, vier oder sogar acht Megahertz.

Die Taktfrequenz wird von Herstellern häufig als Maß für die Rechengeschwindigkeit hingestellt, was aber allenfalls für den Vergleich von Rechnern mit identischem Pro-

Hier werden einzelne Fachausdrücke eingehend behandelt. Da bei der Kommunikation mit dem Computer meist die englische Sprache verwendet wird, werden hier zunächst die englischen Begriffe genannt, dann die deutsche Übersetzung. In den Gesamtindex werden sowohl deutsche als auch englische Stichwörter aufgenommen, damit Sie es leichter haben, das von Ihnen Gesuchte zu finden.

zessor sinnvoll ist. Ein Rechner mit 6802-Prozessor und 2-MHz-Takt ist nicht unmittelbar einem Rechner mit Z80-Prozessor bei 2 MHz gegenüberzustellen. Abgesehen davon spielen noch andere Einflüsse eine Rolle, so daß die Taktfrequenz höchstens einen vagen Anhaltspunkt für die Rechengeschwindigkeit gibt.

CMOS = CMOS

Bei der Halbleiter-Herstellung werden in das Halbleitermaterial gezielt Fremdatome eingebaut (Dotierung), die je nach Art eine „n-Leitung“ durch negative Elektronen oder eine „p-Leitung“ durch positive Ladungsträger („Löcher“ in der Elektronenbesetzung) bewirken. Die Anordnung von p- und n-leitenden Zonen im einem Halbleiterkristall bestimmt die Funktion des Bauteils: Eine Diode beispielsweise besteht nur aus einer p- und einer angrenzenden n-Schicht, während ein „bipolarer“ Transistor drei Zonen mit der Schichtfolge p-n-p oder auch n-p-n enthält.

Bipolar-Transistoren benötigen einen erheblichen Steuerstrom, weil ihre Steuerelektrode (die „Basis“) niederohmig ist. Beim Feld-Effekt-Transistor (Fet) kann man dagegen die Steuerelektrode (aus Metall) durch eine Oxidschicht vom Halbleiter (Semiconductor) isolieren. Ein solcher „MOS-Fet“ ist fast leistungslos steuerbar. Je nach Dotierung des Halbleitermaterials spricht man dabei von einer PMOS- oder NMOS-Technologie.

Die gleichzeitige Verwendung von PMOS- und NMOS-Fets wird als

CMOS (complementary MOS) bezeichnet, weil dabei jedes Gatter komplementär aus spiegelbildlichen PMOS- und NMOS-Hälften aufgebaut ist. Da immer eine davon gesperrt ist, ergibt sich ein extrem niedriger Energieverbrauch – ideal für batterieversorgte Geräte wie beispielsweise Taschenrechner. Zum Beispiel ist der CMOS-bestückte Tandy Modell 100 netzunabhängig, und der Speicherinhalt bleibt sogar wochenlang erhalten.

Coaxial Cable = Koaxialkabel

Mit schlichten Leitungsdrähten oder Flachbandkabeln lassen sich zwar niederfrequente Signale übertragen, aber für höhere Frequenzen sind sie nicht geeignet. Störsignale aus der Umgebung, Impulsverzerrungen und Dämpfung machen das Signal mit der Leitungslänge zunehmend unbrauchbar. Kabel mit einem koaxial in einem geerdeten Abschirmschlauch aus Metallgeflecht angeordneten isolierten Innenleiter sind dagegen hochfrequenzgeeignet und beispielsweise für die Verbindung von Rechner und Bildschirm unerlässlich. Koaxialkabel werden auch beim Aufbau von Netzwerken zur Verbindung der einzelnen Stationen untereinander verwendet.

COBOL = COBOL

COBOL (COmmon Business Oriented Language=Universalsprache für kaufmännische Anwendungen) war die erste Programmiersprache, die nicht für mathematisch-naturwissenschaftliche Zwecke entwickelt wurde. Sie hat vor allem im kommerziellen Bereich bei Großrechenanlagen und Rechnern der mittleren Datentechnik Verbreitung gefunden. COBOL ist aber auch für Rechner mit CP/M-Betriebssystem verfügbar.

Bildnachweis

729: Ian McKinnell, Syco Systems
730: Mary Evans Picture Library, Bernhard Jennings
736, 744, 754: Ian McKinnell
737, 742, 743: Chris Stevens
739: Liz Dixon
747: Steve Cross
748: William Stuart Systems Ltd.
749: Newcastle Polytechnic
755: Atari



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

computer kurs

Heft 28

Der Osborne Vadem wiegt nur vier Kilo, ist aber ein komplettes System mit zwei integrierten Floppy-Laufwerken.



Sturm auf die Raumfestung

Ursprünglich wurde das Programm ZAXXON für Arcadengeräte in Spielhallen entwickelt, nun ist es auch für den Heimcomputer erhältlich. Die Grafik-Darstellung ist hervorragend.



Prozessor-Revolution

1977 stellte die Firma Zilog den Microprozessor Z80 vor. Dieser Chip sollte für die revolutionäre Entwicklung auf dem Computermarkt sorgen.



Assembler-Adressen

Es gibt mehrere Arten der Adressierung in der Assemblersprache. Wir zeigen die Unterschiede.



Praxis-Tips: Übungen

Dieser Kurs-Teil bringt praktische Anwendungen der bisherigen Theorien: Widerstände, Gatter und Binärwandler.



Roboter fürs Haus

Die „Movits“ sind kleine Bodenroboter, die mit dem Heimcomputer gesteuert werden.

