

Einsteigen - Verstehen - Beherrschen

DM 3,80 65 30 sfr 3,80

computer kurs

Heft **32**

Ein wöchentliches Sammelwerk

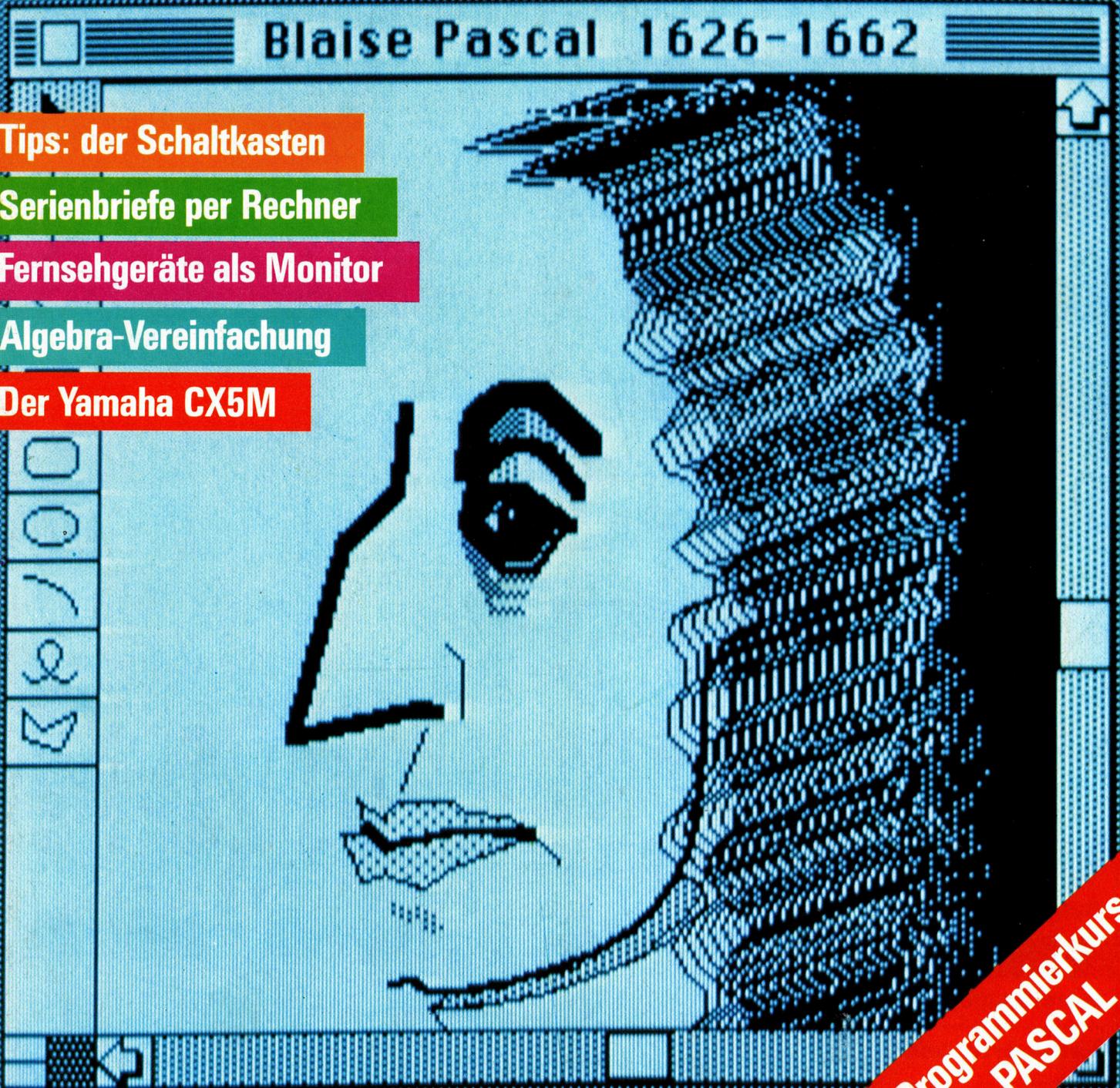
Tips: der Schaltkasten

Serienbriefe per Rechner

Fernsehgeräte als Monitor

Algebra-Vereinfachung

Der Yamaha CX5M



Programmierkurs
PASCAL

computer kurs

Heft 32

Inhalt

Computer Welt

Financial-Micro-Times 869
Spiele mit Geschäftscharakter

Perfekte Zukunft? 894
Künftige Entwicklungen der Robotik

PASCAL

Disziplin ist alles 872
Eine gut strukturierte Programmiersprache

Software

Vier Minuten Vorwarnzeit 874
„Missile Command“ von Atari

Schnelle Post 884
Serienbriefe per Rechner

Hardware

Gut gestimmt 875
Ein musikalischer Computer: Yamaha CX5M

Computer-Logik

De Morgans Gesetz 878

Peripherie

Bildwiedergabe via Videobuchse 881
Fernseher mit Monitorqualität

BASIC 3 2

Grafik-Entwürfe 886

Tips für die Praxis

Schaltkasten 888

Bits und Bytes

Adreßregister 890

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 Überweiskopie Ihre vollständige Anschrift gut lesbar enthalten.

SAMMELORDNER

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

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

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

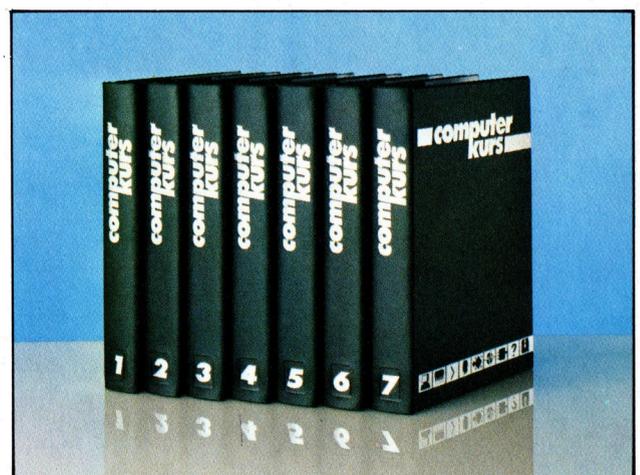
Schweiz: Der Sammelordner kostet sfr 15. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

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 Brandt (Layout), Sammelwerk RedaktionsService GmbH, Paulstraße 3, 2000 Hamburg 1

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





FINANCIAL MICROTIMES



Große Geschäfte

Viele Menschen haben sich mit dem Gedanken beschäftigt, ein eigenes Geschäft zu betreiben. Doch selbst wenn es nur um einen kleinen Laden geht, können die damit verbundenen Probleme beträchtlich sein. Das gilt besonders für die Bewältigung finanzieller Fragen. Dieser Beitrag beschäftigt sich nicht mit Business-Software, sondern hat Simulationen zum Inhalt, die Probleme und Herausforderungen des „Big Business“ einschließen. Das wohl bekannteste Spiel dieser Art heißt „Monopoly“ und wurde von dem gleichnamigen Brettspiel adaptiert, bei dem es um Erwerb und Verkauf von Immobilien geht. Spiele mit Geschäftscharakter sind für Computer ideal.

Die Branchenbreite der zu betreibenden Unter-

nehmen ist umfangreich: Brauereien, Luftfahrtgesellschaften und Autohandlungen sind nur einige Beispiele. Jede Geschäftsart hat ihre spezifischen Probleme.

Doch für welche Art von Geschäft man sich auch entscheiden mag — Geschäftssimulationsprogramme werden nach einem Grundmuster geschrieben. Man hat die Funktion des Managers eines Unternehmens, und bei Spielbeginn steht einem ein bestimmter Geldbetrag zur Verfügung. Dieser beträgt beim Programm „Corn Cropper“, in dem man Farmer ist, lediglich 50 000 Pfund, doch als Leiter einer Ölgesellschaft, wie etwa bei „Dallas“, benötigt man allein zum Start 100 Millionen Dollar.

Teile dieses Kapitals

werden dann in sogenannte „Sachwerte“ wie Flugzeuge, Autofabriken oder Land investiert. Danach erst kann man die entsprechende Dienstleistung anbieten oder Waren produzieren und Erträge erwirtschaften. Aber natürlich ist das nicht so einfach: Erfolg oder Mißerfolg eines Unternehmens hängen von vielen anderen Faktoren ab.

Die Realitätsnähe der einzelnen Programme ist unterschiedlich. „Dallas“ dient vorwiegend der Unterhaltung. Andere dagegen sind sehr realistisch: Bei „Corn Cropper“ werden Daten über Sonnenscheindauer und Regenmenge ebenso eingesetzt wie ideale Bedingungen für das Wachstum des Weizens, Arbeitslöhne und laufende Kosten für Traktoren.

Sicher wird es einige Leute geben, die sich mit dem bloßen Betreiben eines Unternehmens nicht

begnügen, nicht einmal mit einer Ölgesellschaft, die 100 Millionen Dollar „schwer“ ist. Die Möglichkeit, die sich diesen Spielern bietet, ist, einen Staat zu lenken. So wird in „1984“ das Wirtschaftsgefüge von Großbritannien simuliert, und der Spieler fungiert diesmal als Premierminister.

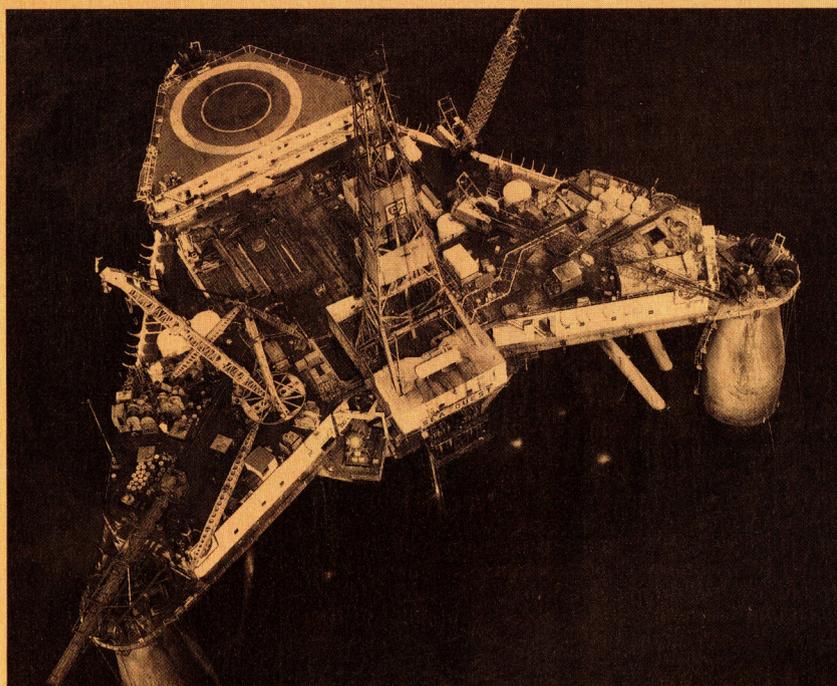
Für die Simulation der Staatswirtschaft bedienen sich Ökonomen eines „Modells“, — nicht eines physischen Modells, sondern eines umfangreichen Diagramms, in dem dargestellt wird, wie die einzelnen Wirtschaftsbereiche ineinander übergreifen und sich beeinflussen. Das in „1984“ dargestellte Modell ist sehr einfach, da es nur fünf Teilbereiche umfaßt: Regierung, Bankwesen, Bevölkerung, Industrie und den Rest der Welt. Doch selbst unter diesen Voraussetzungen gibt es 30 Querverbindungen innerhalb des Mo-

dells: Die Regierung macht der Industrie Zugeständnisse, die Bevölkerung zahlt Steuern an die Regierung und so weiter.

Auch bei „Treasury“ gibt es ein Wirtschaftsmodell, das die britische Regierung zur Vorhersage für die wirtschaftliche Entwicklung des Landes benutzt. Es ist klar, daß dieses Modell komplizierter gestaltet werden mußte. Das Simulationsprogramm, das dieses Modell beinhaltet, heißt „AMODEL“ und läuft auf einem Großrechner von Sperry, dem 1100. Für die Erzeugung der Vorhersagen sind 1100 verschiedene Variablen erforderlich, beispielsweise Inflationsrate, Arbeitslosen-zahl der letzten zehn Jahre. Allein diese Daten belegen über 250KByte an Speicherkapazität. Zum Betrieb des Programmes sind insgesamt acht MByte erforderlich.

Das Modell vermag Arbeitslosenzahlen vorherzusagen, Umfang und Menge von Im- bzw. Exporten, den Dollar-Wechselkurs und viele andere wirtschaftliche Indikatoren. Wie umfangreich und geschlossen dieses Modell ist, verdeutlicht die Tatsache, daß allein zur Vorhersage der Inflationsrate des nächsten Jahres eine Rechenzeit von zwei bis drei Minuten erforderlich ist. Das ist oberflächlich betrachtet nicht viel, doch sollte dabei nicht vergessen werden, daß der Rechner in nur einer Sekunde etwa zehn Millionen maschinencodierte Anweisungen ausführt! Der Ausdruck der Vorhersage dauert weitere 15 bis 20 Minuten.

Computersimulations-spiele bieten die Möglichkeit, ein eigenes Geschäft zu betreiben, ohne das Risiko, Ersparnisse zu verlieren, wenn das Unternehmen scheitert. Leider hat man auch nichts von den erwirtschafteten Gewinnen!



Ölsuche

„Dallas“ ist ein Geschäftsspiel, das Sie zum Manager eines Ölunternehmens macht. Mit 100 Millionen Dollar Startkapital soll ein Ergebnis von 200 Millionen Dollar erreicht werden, um so den größten Konkurrenten — Euing Oil — übernehmen zu können. (Die falsche Schreibweise des aus der bekannten Fernsehserie übernommenen Namens ist Absicht!)

Auf dem Bildschirm sieht man eine Landkarte von Texas. Im Spielverlauf hat man die Möglichkeit, Konzessionen in den verschiedenen Bereichen zu bekommen. Voraussetzung dafür ist ein erfolgreiches Gebot. Nur dann

dürfen Sie eine Probebohrung nach Öl vornehmen. Entdeckt man in dem betreffenden Feld Öl, kann man einen Bohrturm errichten, ferner eine Raffinerie und eine Pipeline.

Bei zu großem Wagemut — beispielsweise, wenn man zu viele Konzessionen erwirbt, ohne daß man auf Öl stößt — muß man zur Aufnahme von Geld zur Bank gehen und dort Zinsen zahlen. Übersteigt die Anleihe 20 Millionen Dollar, geht das Unternehmen in den Besitz der Euing Associates über.

Cases Computer Software liefert das Programm, das für mehrere Systeme erhältlich ist.

Hochfinanz

In diesem Spiel ist man Präsident der L-AIR, einer privaten Luftfahrtgesellschaft, die bei Spielbeginn über drei Millionen Pfund Grundkapital verfügt. Die Anlagewerte müssen innerhalb von sieben Jahren auf 30 Millionen Pfund erhöht werden. Damit ist die Übernahme der British Airways möglich und das Spiel gewonnen.

Zu Beginn jedes Rechnungsjahres ist zu entscheiden, wie viele Flugzeuge man einsetzen will. Dabei bedient man sich der Vorhersagen für das Passagieraufkommen. Anfangs hat man nicht genügend Geld, um ein Flugzeug — Einstandspreis zehn Millionen Pfund pro Maschine — zu kaufen, also muß man Kapital leihen. In ertragreicheren Jahren muß der Spieler entscheiden, ob er Flugzeuge mieten oder kaufen will, was gleichermaßen von den Charraten und vom Zinssatz abhängt.

Umfang der Besatzung und Wartung der Flugzeuge sind ebenfalls zu berücksichtigen. Hat man zu wenig Besatzung oder

Flugzeuge, müssen Flüge gestrichen werden. Im umgekehrten Fall vergeudet man Geld.

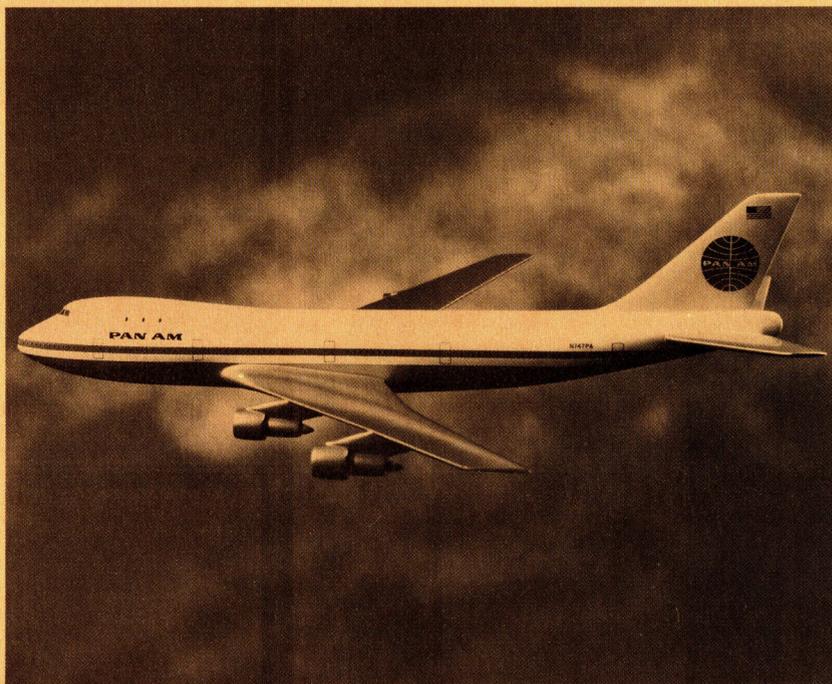
Zwischendurch taucht ein „Telex-Band“ auf dem Bildschirm auf, das Nachrichten übermittelt. So kann zum Beispiel die

OPEC die Ölpreise — damit auch die Preise für Kerosin — erhöht haben, oder man hat die Erlaubnis für den Betrieb einer neuen Luftfahrtstrecke bekommen.

Am Ende eines jeden Jahres zeigt eine Bilanz

das Gesamtergebnis der L-AIR, und der Jahresbericht des Präsidenten erläutert den Erfolg des Unternehmens.

„Airline“, von Cases Computer Simulations, gibt es für den Spectrum, Oric und Acorn.





Erste Gelegenheit



In diesem Spiel ist man der Premierminister von Großbritannien im Jahre 1984. Ziel ist es, so lange wie möglich im Amt zu bleiben. Die Popularität hängt dabei nur vom wirtschaftlichen Erfolg der Nation ab.

Bei Spielbeginn sind unterschiedliche finanzielle Entscheidungen zu treffen. Als Hilfestellung werden acht wirtschaftliche Indikatoren aufgezeigt, darunter die Inflationsrate und der Prozentsatz der Arbeitslosigkeit. Bei Beginn jedes Jahres wird grafisch dargestellt, wie sich die Indikatoren im Laufe eines Amtsjahres verändert haben.

Die erste zu fällende Entscheidung ist die Festsetzung des Leitzinnsatzes, mit dem der Zinssatz für das laufende Jahr bestimmt wird. Darauf müs-

sen die Einkommensverbesserungen für Beamte, Industrie und andere Arbeitnehmer berechnet werden.

Nach Festsetzung der Kosten muß über die Ausgabenverteilung der verschiedenen Ressorts und Ministerien befunden werden. Abschließend wird der Staatshaushalt verkündet, der Gelegenheit zur Steuererhöhung zwecks Verbesserung des staatlichen Einkommens gibt. Am Ende jedes Jahres der fünfjährigen Regierungszeit wird durch Meinungsumfrage der Popularitätsgrad der Maßnahmen ermittelt. Ziel des Spieles ist es, nach Ablauf der Amtsperiode wiedergewählt zu werden.

„1984“ gibt es für den Acorn B und den Spectrum von Incentive Software.

Goldener Herbst

„Corn Cropper“ für den Spectrum und den Acorn B von Cases Computer Simulations simuliert den Betrieb einer Farm, die Weizen anbaut. Bei Spielbeginn stehen 50 000 Pfund zur Verfügung, die innerhalb von fünf Jahren auf 250 000 Pfund anwachsen sollen.

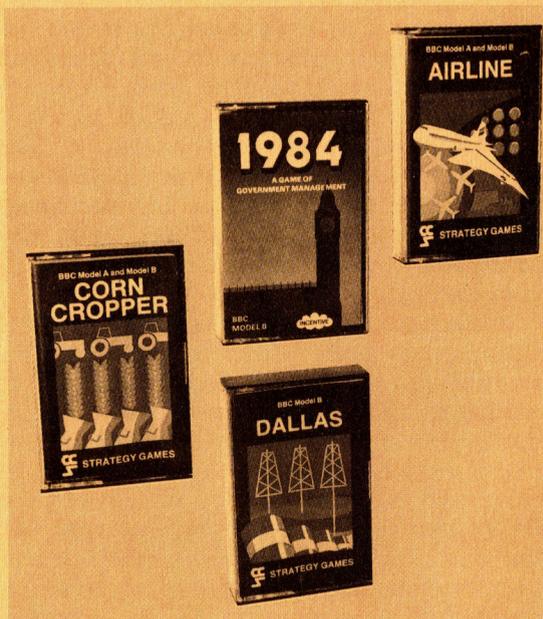
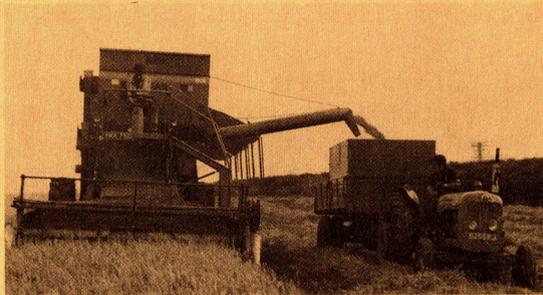
Zunächst müssen 30 Morgen Land bebaut werden. Dafür sind der Kauf von Saatgut, die Anmietung von Traktoren, die Einstellung von Arbeitskräften und Bewässerung — falls es nicht genug regnet — erforderlich. Zu Beginn eines jeden Monats werden Vorhersagen über Regenmenge und Temperaturen aufgezeigt, woraus abzuleiten ist, ob man anpflanzen sollte oder nicht.

Die Monate verstreichen, und nun ist der „Reifestatus“ zu beachten — eine Art Kalender, in dem aufgezeigt wird, ob

der Weizen bereits geerntet werden kann. Im günstigen Fall sind eventuell weitere Arbeitskräfte und Erntemaschinen zu beschaffen.

Es gibt allerdings auch Probleme: So können beispielsweise Ratten das Saatgut fressen. Ist das Korn nicht geschützt, besteht die Gefahr eines Schädlingsbefalls. Frost kann ebenfalls zur Vernichtung der Pflanzen beitragen. Zur Verbesserung des Ertrages benötigt man Düngemittel. Doch auch diese müssen rechtzeitig ausgebracht werden.

Ziel ist es, durch Verkauf der Erträge wachsende Einkünfte zu bekommen, um so mehr investieren zu können. Das Bankkonto darf nicht überzogen werden. Letzter Ausweg in diesem Fall ist der Verkauf von Grundbesitz, um an Bargeld zu kommen.



Diese Geschäftsprogramme für Heimcomputer basieren auf Simulationstechniken und erlauben dem Spieler, eine Luftfahrtgesellschaft, eine Weizenfarm, eine Ölgesellschaft oder gar die Geschehnisse der britischen Wirtschaft zu leiten. Der Realitätsgrad hängt von der Anzahl der Kontrollfaktoren des Spiels ab.

Disziplin ist alles

Mit diesem Artikel beginnt eine Serie über PASCAL. Wir werden uns mit den Gedanken und Prinzipien, die zur Entwicklung führten, beschäftigen. Oft als „streng diszipliniert“ bezeichnet, wurde PASCAL entworfen, um gute Programmier-Techniken zu fördern.



Als Mathematiker, Wissenschaftler und Philosoph entwickelte Blaise Pascal (1623–1662) im Jahre 1642 die erste mechanische Rechenmaschine. Die Programmiersprache PASCAL wurde zu Ehren dieses Beitrags zur Computerwissenschaft nach ihm benannt.

Die Programmiersprache PASCAL wurde im Jahre 1970 von Professor Niklaus Wirth in Zürich entwickelt. Die Sprache erhielt den Namen eines französischen Mathematikers und Philosophen des 17. Jahrhunderts – Blaise Pascal –, der die erste mechanische Rechenmaschine mit den vier Grundfunktionen erfunden hatte. Die Sprache PASCAL wurde durch ALGOL 60 stark beeinflusst und war eine direkte Antwort von Wirth auf die „überzüchtete“ und komplexe Version ALGOL 68. Er verkündete, PASCAL würde:

- Programmstrukturen und Konzepte präzise auszudrücken erlauben;
- demonstrieren, daß eine kleine, maschinenunabhängige Sprache mit einem flexiblen Satz an Daten, Anweisungen und Programmstrukturierungs-Details als allgemeines Problemlösungswerkzeug verwendet werden kann;
- bei der Organisation von großen Programmen und komplexen Software-Projekten größere Sicherheit bieten;
- über umfassende Fehlerprüfroutinen verfügen (speziell während der Compilierung) und dadurch ideal als Lehrsprache für Anfänger verwendbar sein;
- sinnvoll auf Microcomputern eingesetzt werden können.

Alle diese Vorsätze sind mit großem Erfolg realisiert worden. Ein kleiner PASCAL-Compiler belegt in der Regel etwa 25 KByte. Obwohl PASCAL nur über ein kleines Vokabular verfügt, das leicht erlernbar ist, ist es trotzdem viel leistungsstärker als manche andere Computersprache. Viel wichtiger ist jedoch, daß PASCAL im Ausdruck erheblich stärker ist. Dies gilt zum einen für die Art, in der Algorithmen geschrieben werden können, und zum anderen für die Einfachheit, mit der Daten umschrieben werden können, egal, wie kompliziert sie sind.

Vielleicht sogar der größte Vorzug von PASCAL ist die eben erwähnte einfache Ausdrucksweise für Algorithmen. Sie können so programmiert werden, wie man über sie nachdenkt. Diese Freiheit beim Programmieren macht PASCAL zu einem idealen Problemlösungswerkzeug und nicht zu einem weiteren Bestandteil des betreffenden Problems. Die Sprache hat viele weitere Vorteile. PASCAL ist eine Compilersprache, was nicht nur bedeutet, daß die Programmausführung viel schneller

wird, sondern auch, daß kein Speicherplatz unnötig mit Interpretern und Programmtexten belegt wird. – Alles was Sie brauchen, ist der compilierte „Object“-Code.

In der Tat ist es gerade bei langen Programmen viel einfacher, in PASCAL statt in BASIC zu programmieren. PASCAL läßt eine „schlampige“ Programmierung gar nicht erst zu. Betrachtet man die Sache etwas näher, so ist die aufgezwungene Disziplin nicht strenger als die, die zur effektiven und sicheren Organisation eines Programms ohnehin notwendig ist. Die eingesparte Zeit beim späteren „Debuggen“ eines Programms wiegt dies bei weitem auf. Ferner bedeutet dies, daß ein Grundwissen über PASCAL ein enormer Vorteil für das Verständnis der Sprachen der nächsten Dekade ist – beispielsweise bei Modula-2, OC-CAM oder ADA.

Ein Programmierer, der von BASIC auf PASCAL umsteigen will, wird bei einem ersten Blick auf ein PASCAL-Programm den krassensten Unterschied in den sehr umfangreichen und eigenartig aussehenden Definitionen (die manchmal fast sinnlos erscheinen) sehen. Der gesamte Anfang eines längeren PASCAL-Programms scheint absolut nichts zu „tun“. Dies liegt zum Teil daran, daß Sie eigene Worte innerhalb der Programmiersprache verwenden können. Eine entsprechende Definition muß selbstverständlich vor dem eigentlichen Programmstart erfolgen, damit PASCAL die Worte als Befehle interpretieren kann. PASCAL ermöglicht Ihnen so, am Anfang eines Programms neue Befehle festzulegen, wie ClearScreen oder Pause (Anzahl der Sekunden), und diese dann im eigentlichen Programm zu verwenden. Das sieht dann so aus:

```
begin
  ClearScreen;
  Write ("Hallo!");
  Pause (3);
  ... usw.
```

Überraschend einfach

PASCAL überrascht den Programmierer mit einer ausgezeichneten Betrachtung des Konzepts eines Computersystems, so daß Daten und Berechnungen in einer natürlichen, logischen Form definiert und ausgedrückt werden können.

Die wesentlichen Details von Pascal

- Vollständig freies Format und freie Gestaltung des Programmtextes
- Flexible Bedingungen bei der Benennung von Programmobjekten
- Die Möglichkeit zur Definition neuer Befehlswörter
- Einfache und einheitliche Syntax
- Modulare Programmstruktur
- Flexible Kontrolle von Daten und Berechnungen
- Sehr schnelle Compilierung sowie Fehlerdiagnose
- Ein kleiner, hocheffizienter Compiler

Das Diagramm, das den „Stammbaum“ höherer Programmiersprachen zeigt, beinhaltet nur die Haupteinflüsse der wichtigsten befehlsorientierten Compilersprachen. Daher werden funktionelle Sprachen, wie LISP oder PROLOG, nicht berücksichtigt. Auch FORTH wird nicht aufgeführt, da es kaum klassifizierbar ist. Der Hauptstamm beginnt mit ALGOL 60. Danach kommt kaum eine moderne Sprache, die nicht direkt oder indirekt von PASCAL beeinflusst wurde.

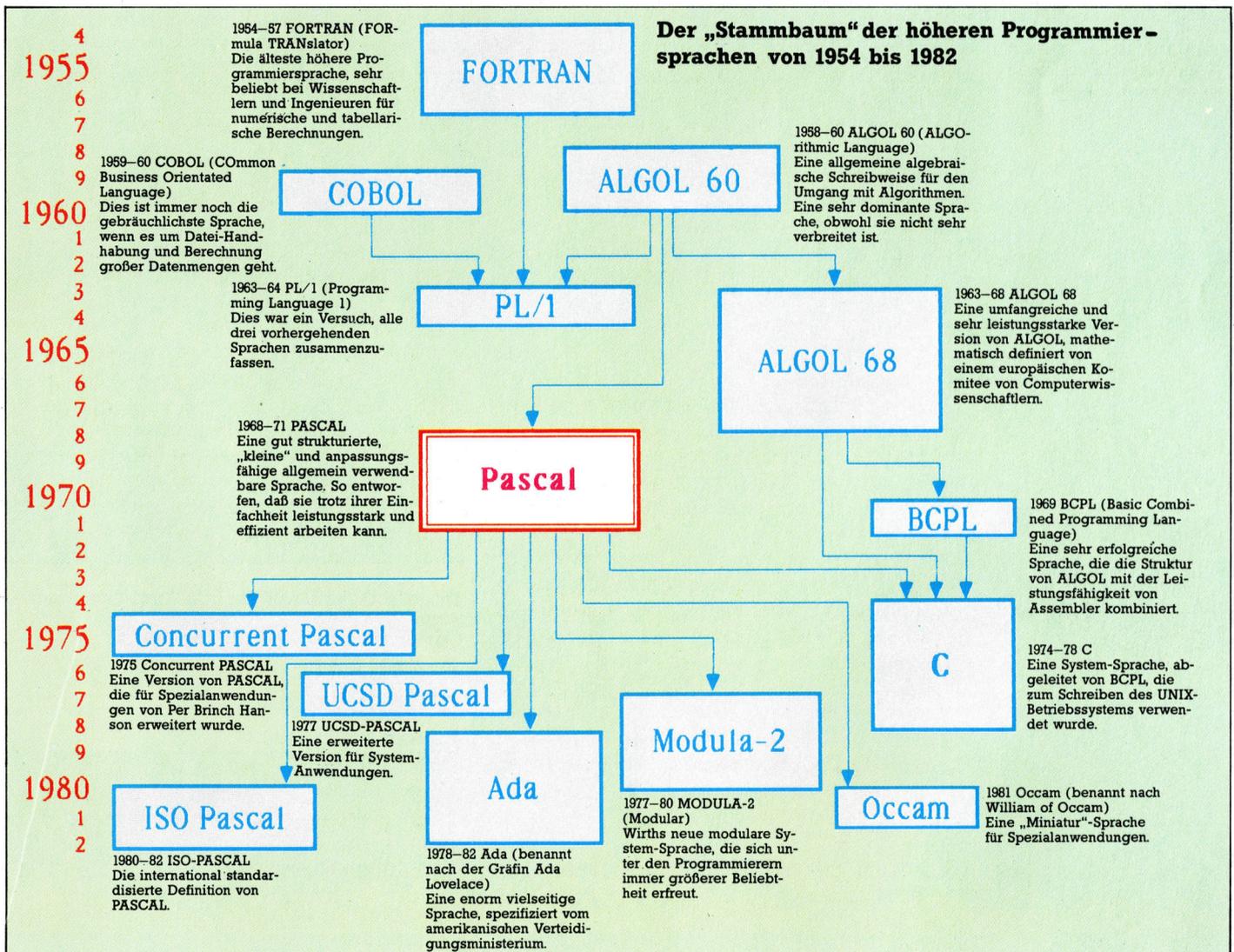
Neben dem Einsatz als Lehrsprache eignet sich PASCAL auch für kommerzielle und systemspezifische Anwendungen. PASCAL ist für die Entwicklung von Finanzpaketen und Compilern verwendet worden. Das „p-system“, das an der UCSD (Universität von Kalifornien in San Diego) gegen Ende der siebziger Jahre entworfen wurde, ist sowohl in PASCAL entwickelt als auch geschrieben worden. Die Software für Lisa und Macintosh von Apple wurde, einschließlich der Betriebssysteme, hauptsächlich in PASCAL bzw. der abgeleiteten Sprache CLASCAL geschrieben. Viele tausend professionelle Programmierer haben sich

auf PASCAL umgestellt, entweder an den Universitäten auf DEC oder VAX oder eventuell mit Hilfe eines Apple II mit einer Version des „p-systems“, bekannt als Apple-PASCAL.

Programmentwicklungen

Mitglieder des ursprünglichen UCSD-Teams findet man heute bei der Arbeit an MODULA-2-Compilern oder anderen Systemen sowie beim Schreiben von Anwenderprogrammen für viele Software-Häuser. – Und alle arbeiten mit PASCAL oder einer leicht abgewandelten Version dieser Sprache.

Ein beachtenswerter Punkt ist, daß all dies ohne die finanzielle Unterstützung kommerzieller Firmen geschah. PASCAL war allein durch seine Vorzüge so erfolgreich, daß es keiner verkaufsfördernden Werbung von Software-Produzenten bedurfte. Unsere Serie über PASCAL konzentriert sich auf die Standardfassung (ISO-PASCAL). Bei Themen wie Grafik oder Routinen behandeln wir auch einige maschinenspezifische Besonderheiten, die einzeln erläutert werden.





Vier Minuten Vorwarnzeit

Eines der aggressivsten (manche würden gar sagen der geschmacklosesten) Spiele, die je programmiert wurden, ist „Missile Command“. Hier soll eine Zivilisation gegen Atomraketen mit Mehrfachsprengköpfen verteidigt werden.

Wenngleich Missile Command schon längst den Gipfel seiner Attraktivität überschritten hat, so gilt es doch als eines der erfolgreichsten überhaupt. Für Atari-Computer ist eine Heimversion erhältlich.

Der Spieler übernimmt die Rolle eines Befehlshabers einer Raketenabwehrstation während eines Atomkrieges und muß sechs Städte vor der atomaren Vernichtung bewahren. Hierbei geht es um Reaktions- und Kombinationsvermögen, die bei allen „Ballerspielen“ bestimmend sind.

Auf dem Bildschirm wird eine abstrakte „Landschaft“ gezeigt, in der die sechs Städte liegen. In der Mitte befindet sich ein pyramidenähnliches Gebilde, auf dem die startbereiten Abwehrraketen positioniert sind. Die Spuren der angreifenden Raketen werden am oberen Bildschirmrand sichtbar. Der Spieler bewegt mittels Joystick ein Zielkreuz über den Schirm. Dieses wird auf dem Weg der ankommenden Raketen plaziert. Durch Betätigung des Feuerknopfes wird eine Abwehrrakete abgefeuert, die auf den durch das Kreuz bestimmten Koordinaten explodiert und dabei die im Bereich befindlichen feindlichen Raketen zerstört.

Eine Reihe anfliegender Raketen ist jedoch mit Mehrfachsprengköpfen ausgestattet, die verschiedene Flugrichtungen einnehmen und eine ganze Stadt zerstören können. Noch komplizierter wird das Spiel dadurch, daß tieffliegende gegnerische Flugzeuge und Satelliten auftauchen, die ebenfalls Raketen salven abfeuern. Am Ende jeder Angriffswelle wird die Zahl der erfolgreich verteidigten Städte gezeigt, sowie die Anzahl der übriggebliebenen Abwehrraketen.

Mit fortschreitendem Spielverlauf bewegen sich die Raketen schneller, und die Zahl der Mehrfachsprengköpfe nimmt zu. In dieser Phase ist die Entwicklung einer Strategie wich-

tiger als die Abwehr einzelner Raketen durch gezielte Schüsse. So kann der Spieler beispielsweise eine „Barriere“ von Abwehrraketen plazieren, die in einer Reihe explodieren und – mit Glück – die angreifenden Raketen auf einmal sprengen.

In den höheren Spielerebenen kommen mit Fallschirmen abgeworfene Sprengköpfe hinzu, die sehr schwer zu zerstören sind. Explodiert die Abwehrrakete neben dem Ziel, wird der betreffende Sprengkopf abgelenkt. Deshalb muß die Abwehrrakete genau plaziert sein.

Im ganzen Spiel ist entscheidend, daß nur eine begrenzte Anzahl von Abwehrraketen zur Verfügung steht (30 in der ersten Ebene). Sind diese verbraucht, muß man hilflos zusehen, wie die Städte zerstört werden.

Pro Schwierigkeitsgrad finden je zwei Angriffswellen statt. Danach wird der Punktestand gezeigt. Wie auch bei anderen Atari-Programmen kann man auf eine höhere Ebene „umschalten“. Die Schwierigkeitsgrade sind an unterschiedlichen Vorder- und Hintergrundfarben zu erkennen. Sind alle sechs Städte zerstört, ist das Spiel beendet, und auf dem Bildschirm erfolgt eine heftige Explosion, aus der die Worte „THE END“ gebildet werden.

Für Atari-Computer ist das Programm als Steckmodul erhältlich. Es wird mit einer umfangreichen Bedienungsanleitung geliefert. Die Broschüre gibt eine detaillierte Beschreibung für Spielvorbereitung, Punktesammeln und Spieltips und ist reich illustriert.

Missile Command: Für alle Atari-Computer

Hersteller: Atari Corporation

Autoren: Atari

Joysticks: Erforderlich

Format: Steckmodul



Gut gestimmt

Yamaha hat als Hersteller hochwertiger Musikinstrumente, besonders von Synthesizern, seit langem einen guten Ruf. Der unlängst vorgestellte MSX-Rechner CX5M basiert auf den Erfahrungen des Unternehmens und ist der erste Heimcomputer, der speziell zum Musizieren entwickelt wurde.

Obwohl der Yamaha CX5M mit dem MSX-Standard ausgestattet ist und gleichermaßen fürs Spielen wie für Textverarbeitung verwendet werden kann, wurde der Rechner doch vor allem für Freunde elektronischer Musik entwickelt. Eine externe YK-10- oder YK-01-Klaviertastatur kann in die seitlich am Rechner befindliche Buchse gesteckt werden. Dort sind ebenfalls die MIDI-Ports zu finden, durch die jeder Synthesizer oder jeder andere elektronische Klangerzeuger mit entsprechendem MIDI-Interface über den Rechner gesteuert werden kann. Ferner steht ein Paar von Audiobuchsen zur Verfügung, die den Anschluß externer Lautsprecher erlauben.

Der Computer ist anderen MSX-Rechnern ähnlich. 48 Tasten, ergänzt um zehn Kontrolltasten, über die die MSX-Standard-Funktionen aktiviert werden, wie Grafiktaste und Codetaste zur Erzeugung von Spezial- und Sonderzeichen, bilden das Tastenfeld. Darüber sind fünf Funktionstasten angebracht, die mit zehn frei programmierbaren Befehlen belegt werden können. Unten rechts befindet sich das aus vier Tasten bestehende Cursor-Steuerungsfeld. Darüber sind fünf weitere Tasten angebracht, die mit MSX-Standard-Befehlen wie INSert, DELete und STOP belegt sind. Am oberen Rand der Tastatur befindet sich ein Steckmodulschacht.

An der rechten Seite des CX5M liegen zwei Atari-Standard-Joystick-Ports. Auf der Rückseite findet man einen Parallelanschluß, das Centronics-kompatible Drucker-Interface, eine Cassettenbuchse, einen Audioanschluß für externe Mono-Wiedergabe, einen Videomonitor-Anschluß, eine Buchse für Fernseheranschluß sowie eine für die Stromversorgung. Die MIDI-Verbindungen und das Keyboard-Interface, die dem CX5M seine einzigartigen musikalischen Möglichkeiten geben, sind in einem Modul untergebracht.

Das dreieinhalb Oktaven umfassende YK-01-Keyboard – im Lieferumfang des Rechners enthalten – macht einen sehr professionellen



Eindruck, wenngleich „ernsthafte“ Keyboardspieler die Tasten für zu schmal halten mögen. Durch integrierte Software kann die Tastatur „geteilt“ werden. Das bedeutet: Eine programmierte Stimme läßt sich auf dem unteren Tastaturteil spielen und gleichzeitig eine andere Stimme auf den restlichen Tasten. So können beispielsweise „Streicher“ auf den oberen Tasten erzeugt werden, die im unteren Tastaturteil von „Bläsern“ begleitet werden. Auf dem Keyboard kann sowohl monophon als auch polyphon (oder in beiden Modi) gespielt werden. Gleichzeitig sind acht Noten spielbar.

Der Yamaha CX5M, so erwartet man, wird besser in Musikfachgeschäften als in Computershops verkauft werden. Zwecks Unterstreich-ung seiner musikalischen Möglichkeiten wird der Rechner als Paket mit dem YK-01-bzw. dem YK-10-Klavier-Keyboard angeboten. Diese Tastatur paßt in die kleine Schnittstelle, die an der Unterseite des Computers befestigt ist.

POLY und MONO

Das Musikprogramm wird durch den Befehl CALL MUSIC aufgerufen, worauf ein aus fünf Blöcken bestehendes Menü erscheint. Durch Betätigung der Return-Taste kann der Anwender die Optionsliste absuchen und die Parameter jeder Option durch Betätigung der Cursor-Tasten ändern. Die als POLY und MONO bezeichneten Blöcke erlauben dem Spieler, zwischen 46 vorprogrammierten Stimmen und zwischen den beiden Modi zu wählen. Die Stimmen umfassen konventionelle Musikinstrumente wie Orgel, Gitarre und Perkussionsinstrumente – etwa Vibraphon oder auch Kuhglocke – ebenso wie „Alltagsgeräusche“. Dazu gehören eine Krankenwagensirene oder das Fallen von Regentropfen. Aufgrund des besonderen Sounds einiger Stimmen können diese bei Verwendung von „Sustain“ nicht abgerufen werden. Der Anwender kann durch



Tastatur
Die Tastatur umfaßt alle Funktionstasten, die zum MSX-Standard gehören, einschließlich der frei programmierbaren Funktionstasten und dem Cursor-Steuersfeld.

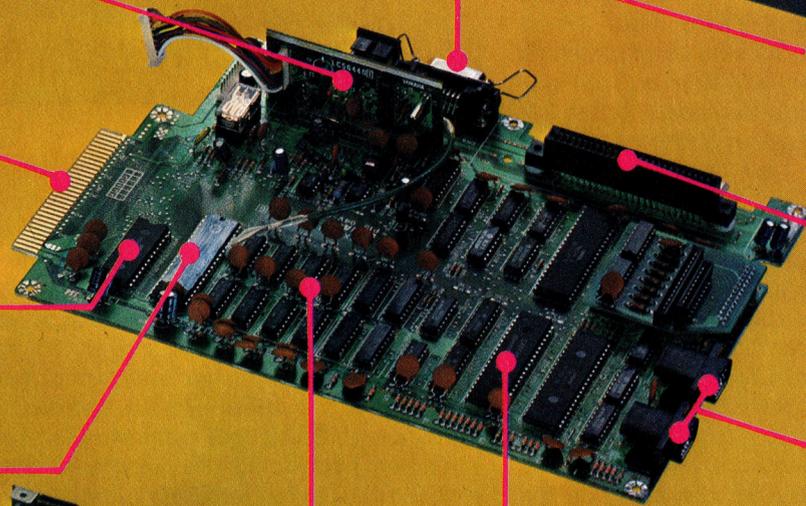
Stromversorgung
Die Stromversorgung wurde seitlich an der Hauptplatine untergebracht.

Steckleiste
Die Box mit dem Musikinterface wird auf diese Steckleiste aufgeschoben.

BASIC-ROM
Das 32 K-MSX-BASIC befindet sich in diesem Chip.

Video-Chip
Der Computer ist mit dem TMP8265-Videochip ausgestattet, und bietet dem Programmierer 32 Sprites und 16 verschiedene Farben.

RF-Modulator
Hiermit kann der Computer an ein normales Fernsehgerät angeschlossen werden.



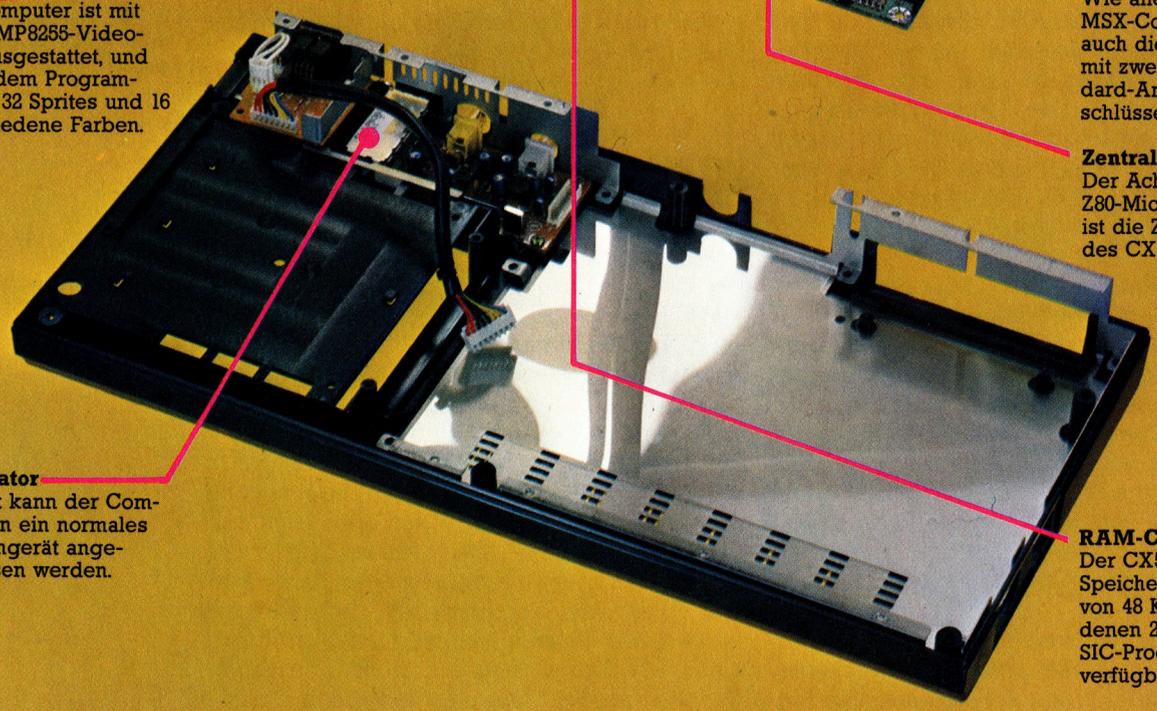
Drucker-Schnittstelle
Der CX5M ist mit einem Centronics-kompatiblen Parallel-Drucker-Port ausgestattet.

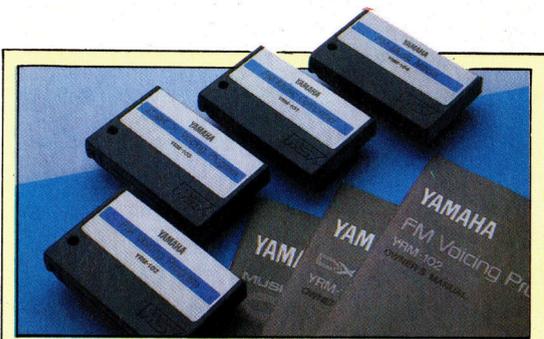
Modulschacht
Der parallele Modulschacht nimmt sowohl weitere Musikprogramme in Steckmodulform auf als auch Standard-MSX-Module.

Joystick-Anschlüsse
Wie alle anderen MSX-Computer ist auch dieser Rechner mit zwei Atari-Standard-Analoganschlüssen versehen.

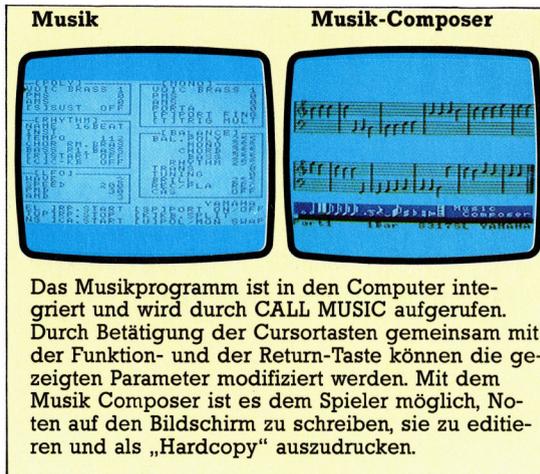
Zentraleinheit
Der Achtbit-Z80-Microprozessor ist die Zentraleinheit des CX5M.

RAM-Chips
Der CX5M hat eine Speicherkapazität von 48 K RAM, von denen 28 K für BASIC-Programme frei verfügbar sind.





Dies sind einige der derzeit für den CX5M zur Verfügung stehenden Module. Jedes der gezeigten Programme erlaubt dem Computer die Ausführung einer bestimmten Funktion: Vom Programmieren des DX-7-Synthesizers bis hin zur Komposition von Musikprogrammen. Mit den Modulen werden detaillierte Dokumentationen geliefert.



Das Musikprogramm ist in den Computer integriert und wird durch CALL MUSIC aufgerufen. Durch Betätigung der Cursor-Tasten gemeinsam mit der Funktion- und der Return-Taste können die gezeigten Parameter modifiziert werden. Mit dem Musik Composer ist es dem Spieler möglich, Noten auf den Bildschirm zu schreiben, sie zu editieren und als „Hardcopy“ auszudrucken.

Veränderung der Stimmen in den Modi POLY und MONO gegensätzlich klingende Töne erzeugen, also auf einem Teil monophon, auf dem anderen polyphon spielen. Natürlich ist eine Verwendung der Tastatur komplett ebenfalls möglich. Dagegen läßt sich die Aufteilung des Keyboards in zwei POLY-Bereiche nicht durchführen.

Die erzeugbaren Klänge sind bemerkenswert, selbst bei Wiedergabe über einen gewöhnlichen Fernseherlautsprecher. Allerdings haben einige Klänge nur entfernte Ähnlichkeit mit dem, was sie laut Namen sein sollen. So klingt beispielsweise „Zug“ wie eine ganz gewöhnliche elektronische Orgel. Doch das ist ein Problem aller Hersteller von Synthesizern. Da die meisten Anwender wohl aber nicht eine exakte Reproduktion solcher „Alltagsgeräusche“ erwarten, werden die meisten Spieler mit den erreichbaren Klangergebnissen voll auf zufrieden sein.

Zahlreiche Klangvariationen

Der Bereich RHYTHM sorgt für die rhythmische Begleitung einer Melodie. Sechs verschiedene Rhythmen, die aus Akkorden, Bass und Schlagzeug zusammengesetzt werden können, stehen zur Verfügung. Bass- und Akkord-Töne können vom Spieler selbst definiert werden. Das Rhythmustempo läßt sich ändern. Ferner gibt es eine Option, die die Änderung der Rhythmustonhöhe erlaubt. Durch Modifizierung der Klangwelle über LFO (Low Frequency Oscillator) sind weitere Klangvarianten möglich. Der Bereich BALANCE erlaubt das Setzen verschiedener Lautstärken der einzelnen Rhythmus-elemente, erweitert um MONO und POLY. Damit können die Intensität der Rhythmusbegleitung reduziert und die Lautstärke des POLY-Modus zur besseren Darstellung der Melodie erhöht werden. Sind die gewünschten Klänge einmal erzeugt, können sie durch SAVE auf Cassette gespeichert und mit LOAD später wieder geladen werden.

Neben der integrierten Software bringt Ya-

maha eine Reihe von Steckmodulen für den CX5M heraus. Dazu gehört unter anderem das FM-Stimmprogramm, eine erweiterte Version der Standardsoftware. Sie erlaubt dem Spieler detailliertere Stimmgestaltung durch Veränderung der Frequenzen und Algorithmen. Mit einem zweiten Stimmprogramm auf Modul ist das Programmieren des Yamaha-DX7-Synthesizers möglich. Mittels „FM Music Macro“ kann der Computer mit zusätzlichen BASIC-Befehlen gefüttert werden, wogegen nach Laden des „FM Music Composer“ lediglich Notenlinien zu sehen sind, die auf herkömmliche Art „beschrieben“ werden können. Die Ausgabe über einen Drucker ist ebenfalls möglich.

Obwohl Yamaha ein umfangreiches Softwarepaket für den CX5M anbietet, hat man den Eindruck, daß die Möglichkeiten des Rechners nicht richtig ausgeschöpft werden. Besonders das Modulationsprogramm scheint ein wenig begrenzt. In Teilen sind Klangunterschiede überhaupt nicht festzustellen. Das ist verwunderlich, da Yamaha das CX5M-Betriebssystem auf der Basis des DX7-Synthesizers entwickelt hat. Die Art der Parameteränderung, für die Yamaha sich entschieden hat, wirkt etwas unständig. So müssen beispielsweise die Cursor-Tasten betätigt und zugleich der Cursor durch die Return-Taste gesteuert werden. Laut Auskunft von Yamaha wird aber demnächst bessere Software lieferbar sein.

Der CX5M ist eindeutig für Leute konzipiert, die sowohl einen Computer haben wollen, sich aber auch für elektronisch erzeugte Musik interessieren. Für den Preis von etwa 2000 Mark ist der CX5M hochinteressant. Wer sich jedoch nicht für elektronische Musik interessiert, wird kaum von den Vorteilen des Rechners überzeugt sein, da vergleichbare MSX-Rechner für die Hälfte des Preises zu haben sind. In wie weit Yamaha mit diesem Konzept Erfolg haben wird und den Rechner als Computer für Musikbegeisterte verkaufen kann, bleibt abzuwarten und hängt von der zukünftigen Entwicklung der Software ab.

Yamaha CX5M

ABMESSUNGEN

413 x 216 x 64 mm

ZENTRALEINHEIT

Z80A, 3,58 MHz

SPICHERKAPAZITÄT

48 K RAM, davon 28 K für BASIC-Programme frei verfügbar.

BILDSCHIRM-DARSTELLUNG

40 x 24 Zeichen Text; 256 x 192 Punkte grafische Darstellung, 16 Farben und bis zu 32 Sprites.

SCHNITTSTELLEN

Centronics-Drucker, Fernseher, Monitor, Audioanschluß, Stereo-Anschluß, 2 Joystickanschlüsse, Cassettenrecorderanschluß, ROM-Steckmodulschacht, Erweiterungsbus, Tastatur/Keyboard-Schnittstelle, MIDI-Ein- und Ausgang.

PROGRAMMIERSPRACHEN

BASIC, PASCAL, Assembler

TASTATUR

67 Tasten umfassende Schreibmaschinentastatur mit Cursor-Steuerungsfeld; fünf frei programmierbare Funktionstasten. Die Klaviertastatur umfaßt acht Noten (polyphon) und hat einen Umfang von dreieinhalb Oktaven.

HANDBÜCHER

Ein sehr merkwürdiges Handbuch: Zwar werden einige Details zur Verwendung der Musik-Software gegeben, eine generelle Einführung erfolgt aber nicht. Das MSX-BASIC wird kaum erwähnt.

STÄRKEN

Dieser Rechner ist für Elektronikmusiker bestens geeignet. Unter diesem Aspekt betrachtet gibt es zum CX5M in dieser Preisklasse keine Alternative.

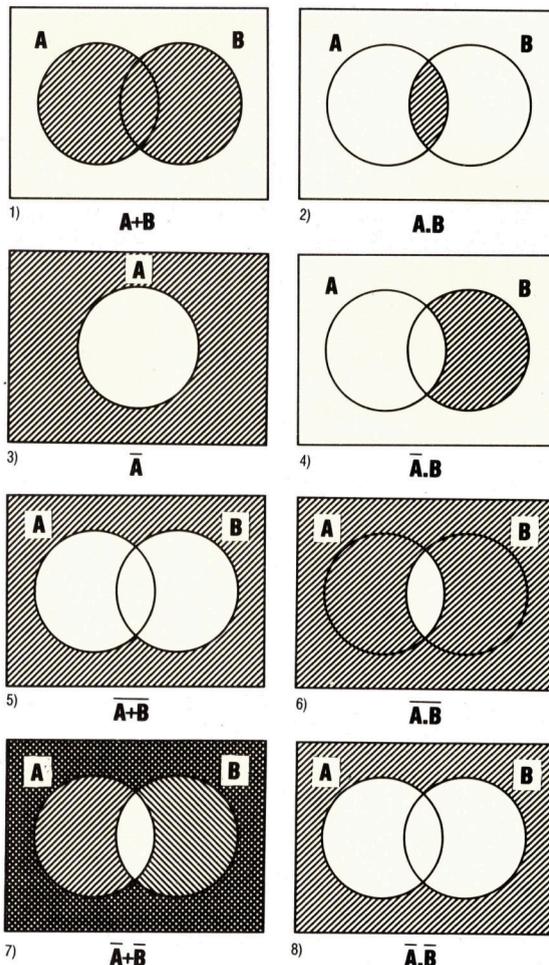
SCHWÄCHEN

Soll der Rechner auch langfristig Erfolg haben, ist eine erhebliche Verbesserung der Software dringend erforderlich. Das Handbuch ist mangelhaft. Für Anwender, die nicht an elektronischer Musik interessiert sind, ist der Rechner zu teuer.

De Morgans Gesetz

Die Vereinfachung Boolescher Algebra führt zu logischen Ausdrücken, die mit einer kleinen Zahl von Operationen (AND, OR und NOT) auskommen. Durch klar definierte Regeln können Gatterbauteile eingespart werden – die gesamte Schaltung wird kompakter und ist preiswerter herzustellen.

Die Ausdrücke der Booleschen Algebra lassen sich grafisch sehr anschaulich durch Schnittmengen-Diagramme darstellen. Die Fläche innerhalb eines Rechteckes symbolisiert dabei alle Kombinationen der Wahrheitswerte, die eingegeben werden können. Die Kreise im Rechteck stehen für bestimmte Kombinationen von Eingabewerten. Hier sehen Sie einige Schnittmengen-Diagramme:



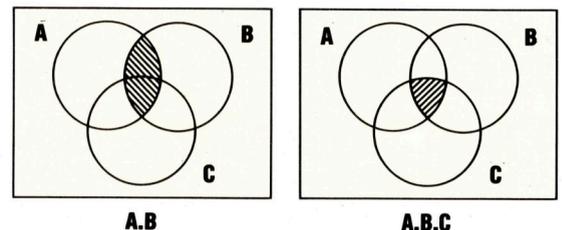
Beim Vergleich von Bild 5 mit Bild 7 sehen Sie, daß NOT (A OR B) etwas anderes ist als NOT (A) OR NOT (B). Ähnlich verdeutlichen Bild 6 und 8 den Unterschied zwischen NOT (A AND B) und NOT (A) AND NOT (B).

Um sich die AND- bzw. OR-Funktion als Diagramm vorzustellen, können Sie sich A.B als das Gebiet denken, wo die Fläche A die Fläche B überlappt – A + B ist dagegen die Vereinigung der Flächen von A und B. In der Booleschen Algebra gibt es viele Ausdrücke, die beim Anblick der dazugehörigen Grafik leichter zu verstehen sind und deren Richtigkeit sich optisch sofort erfassen läßt: Zeichnen Sie die Diagramme folgender Beziehungen: (0 steht dabei für „leere Menge“)

- 1) $A.A = A$
- 2) $A.\bar{A} = 0$
- 3) $A.0 = 0$
- 4) $A.1 = A$
- 5) $A.(A + B) = A$
- 6) $A.(\bar{A} + B) = A.B$

Boolesche Gesetze

Die Regel von der Dualität ist bei der Vereinfachung Boolescher Ausdrücke eine große Hilfe. Sie beruht auf der Symmetrie der Funktionen AND und OR. Um das Gegenstück einer wahren Funktion zu finden, werden alle AND durch OR, alle Nullen durch Einsen ersetzt (natürlich auch umgekehrt). Aus dem fünften Ausdruck der obigen Liste würde so $A + A.B = A$. Da auch dieser Ausdruck richtig ist, zeigt das Schnittmengen-Diagramm: A.B wird vollständig von A umschlossen. Diese Methode läßt sich auch in Fällen mit drei Variablen anwenden, etwa bei $A.B + A.B.C = A.B$. Die beiden Diagramme bestätigen die Richtigkeit:



Versuchen Sie, auch die Gegenstücke der anderen fünf Funktionen zu finden, und kontrollieren sie die Gültigkeit mit entsprechenden Schnittmengen-Diagrammen.

Bei Betrachtung der Bilder 5 und 8 aus der ersten Diagramm-Reihe sehen Sie eine Beziehung, die immer wahr ist: $\bar{A} + \bar{B} = \overline{A.B}$. Bild 6 und 7 zeigen: $\bar{A}.\bar{B} = \overline{A + B}$. Diese beiden als „Gesetz von de Morgan“ bezeichneten Beziehungen gelten auch bei mehr als zwei Variablen ($\overline{A+B+C} = \bar{A}.\bar{B}.\bar{C}$ und $\overline{A.B.C} = \bar{A} + \bar{B} + \bar{C}$). Dabei

wird in mehreren Schritten verfahren:

$$\begin{aligned} & (\overline{A+B}) \cdot \overline{C} \\ &= \overline{A} \cdot \overline{B} \cdot \overline{C} \quad (\text{de Morgans Gesetz auf den Klammerausdruck angewandt}) \\ &= \overline{A+B+C} \quad (\text{mit de Morgans Gesetz neu zusammengestellt}) \end{aligned}$$

Entsprechend der normalen Algebra gibt es auch in der Booleschen Algebra drei weitere gültige Gesetze. Das „Assoziativgesetz“ erlaubt die Verschiebung von Klammern:

$$\begin{aligned} (A \cdot B) \cdot C &= A \cdot (B \cdot C) = A \cdot B \cdot C \\ (A+B)+C &= A+(B+C) = A+B+C \end{aligned}$$

Die Reihenfolge der Buchstaben kann mit dem „Kommutativgesetz“ verändert werden:

$$\begin{aligned} A \cdot B &= B \cdot A \\ A+B &= B+A \end{aligned}$$

Das „Distributivgesetz“ erlaubt das Ausmultiplizieren von Klammern:

$$A \cdot (B+C) = A \cdot B + A \cdot C$$

Einige Beispiele für Vereinfachungen mit Hilfe der genannten Gesetze:

$$\begin{aligned} 1) \text{ Vereinfache } (\overline{A+B} + \overline{A} \cdot B) \cdot B \\ &= (\overline{A} \cdot \overline{B} + \overline{A} \cdot B) \cdot B \quad (\text{de Morgan}) \\ &= \overline{A} \cdot \overline{B} \cdot B + \overline{A} \cdot B \cdot B \quad (\text{Distributivgesetz}) \\ &= 0 + \overline{A} \cdot B \quad (\overline{B} \cdot B = 0, B \cdot B = B) \\ &= \overline{A} \cdot B \end{aligned}$$

$$\begin{aligned} 2) \text{ Vereinfache } \overline{A} \cdot \overline{B} + \overline{A} \cdot B + A \cdot B \\ &= \overline{A} \cdot (\overline{B} + B) + A \cdot B \quad (\text{Distributivgesetz}) \\ &= \overline{A} + A \cdot B \quad (\overline{B} + B = 1) \\ &= \overline{A} + B \quad (\text{Gegenstück der 6. Beziehung}) \end{aligned}$$

$$\begin{aligned} 3) \text{ Vereinfache } \overline{\overline{A} + B} + \overline{\overline{A} + B} + \overline{A} \cdot B \\ &= \overline{\overline{A} \cdot \overline{B}} + \overline{\overline{A} \cdot \overline{B}} + \overline{A} \cdot B \\ &= \overline{\overline{A} \cdot \overline{B}} + \overline{\overline{A} \cdot \overline{B}} + \overline{A} \cdot B \quad (\text{de Morgan}) \\ &= A \cdot B + A \cdot B + \overline{A} \cdot B \\ &= A \cdot B + A \cdot B + \overline{A} \cdot B \quad (\overline{\overline{A}} = A) \\ &= A \cdot (\overline{B} + B) + \overline{A} \cdot B \quad (\text{Distributivgesetz}) \\ &= A + \overline{A} \cdot B \quad (\overline{B} + B = 1) \\ &= A + B \quad (\text{Gegenstück der 6. Beziehung}) \end{aligned}$$

Im letzten Abschnitt des Kurses haben wir die Schaltung des Exklusiv-OR-Gatters kennengelernt. Nun können Sie auch die Schaltung des Gatters ohne Beeinträchtigung der Funktion vereinfachen. Die Wahrheitstabelle des Gatters sieht so aus:

Eingänge		Ausgang
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Aus der Tabelle haben wir bereits ermittelt, daß $C = \overline{A} \cdot B + A \cdot \overline{B}$ ist. Durch eine kleine Vereinfachung würden fünf Logik-Gatter für diesen Ausdruck genügen. – Es gibt jedoch noch eine zweite Möglichkeit: C kann dann 1 sein, wenn A und B beide nicht 1 oder beide nicht 0 sind. C läßt sich also auch so ausdrücken:

$$C = \overline{\overline{A} \cdot B + \overline{A} \cdot \overline{B}}$$

Dies läßt sich mit de Morgans Gesetz zu

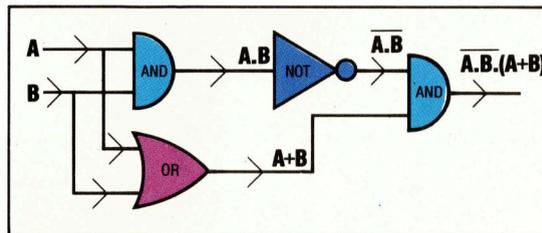
$$C = (\overline{\overline{A} \cdot B}) \cdot (\overline{\overline{A} \cdot \overline{B}})$$

und in einem weiteren Schritt zu

$$C = \overline{\overline{A} \cdot B} \cdot (\overline{A + B})$$

vereinfachen.

Dafür aber reichen vier Gatter:

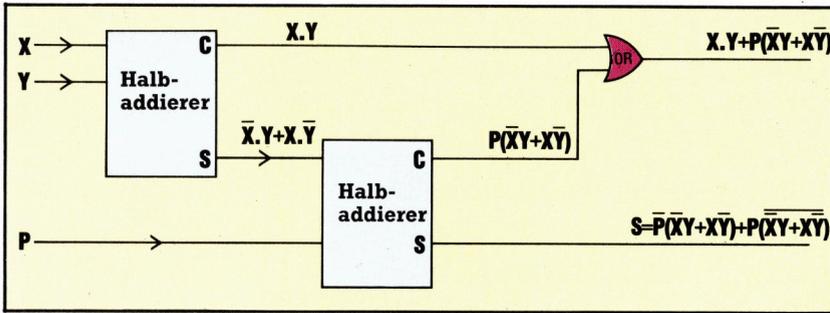


Einen einfachen Addierer zum Zusammenzählen von zwei Bits mit zwei Ausgängen für die Summe und den Übertrag haben Sie schon konstruiert. Dieser „Halbaddierer“ hatte mit „X“ und „Y“ bezeichnete Eingänge. Das Verhalten des Ausgangs S ließ sich mit dem Booleschen Ausdruck $S = \overline{X} \cdot Y + X \cdot \overline{Y}$ beschreiben. De Morgans Gesetz kann auch diesen Ausdruck vereinfachen, nämlich zu $S = \overline{X \cdot Y} \cdot (X + Y)$. Der Übertrags-Ausgang ist einfach $C = X \cdot Y$.

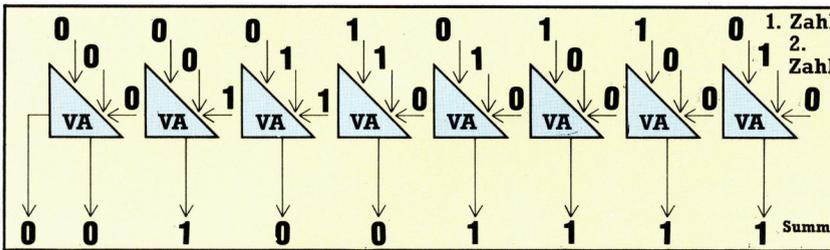
Bau eines Volladdierers

In jeder binären Rechnung müssen pro Spalte des Ergebnisses insgesamt drei Ziffern addiert werden: zwei Summanden und der Übertrag der vorhergehenden Spalte. Um diesen Vorgang zu simulieren, brauchen wir eine Schaltung mit drei Eingängen und zwei Ausgängen. Wird der Übertrag aus der vorhergehenden Zeile mit P bezeichnet, ergibt sich für den Volladdierer diese Wertetafel:

Eingänge			Ausgänge	
P	X	Y	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Hier die Schaltung eines Addierers für Acht-Bit-Zahlen, der aus acht Volladdierern aufgebaut ist:



Für die Fälle, in denen $S = 1$ ist, ergibt sich aus der Tabelle für S der Ausdruck:

$$S = \bar{P}\bar{X}.Y + \bar{P}X.\bar{Y} + P.\bar{X}.\bar{Y} + P.X.Y$$

Nach den neuen Regeln vereinfacht wird daraus

$$S = \bar{P}(\bar{X}.Y + X.\bar{Y}) + P(\bar{X}.\bar{Y} + X.Y)$$

(Distributivgesetz)

$$S = \bar{P}(\bar{X}.Y + X.\bar{Y}) + P(\bar{X}.\bar{Y} + X.Y)$$

(de Morgan)

Ähnlich kann aus der Wertetafel ein Ausdruck für C ermittelt werden:

$$C = \bar{P}X.Y + P\bar{X}.Y + P.X.\bar{Y} + P.X.Y$$

$$C = X.Y(\bar{P} + P) + P(\bar{X}.Y + X.\bar{Y})$$

(Distributivgesetz)

$$C = X.Y + P(\bar{X}.Y + X.\bar{Y})$$

($\bar{P} + P = 1$)

Beachten Sie, daß $\bar{X}.Y + X.\bar{Y}$ der Summenausgang eines Halbaddierers ist. Ein Volladdierer läßt sich also aus zwei Halbaddierern aufbauen.

ÜBUNG 3

1) Vereinfachen Sie die folgenden Ausdrücke:

- a) $A.(\bar{A} + \bar{B})$
- b) $X + Y.(X + Y) + X.(\bar{X} + Y)$
- c) $P.Q + \bar{P}.Q + \bar{P}.\bar{Q}$
- d) $\bar{X} + \bar{Y}.\bar{Z} + \bar{Z}.Y$

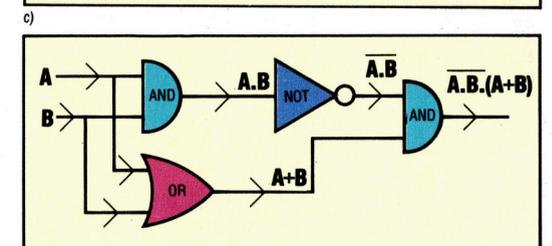
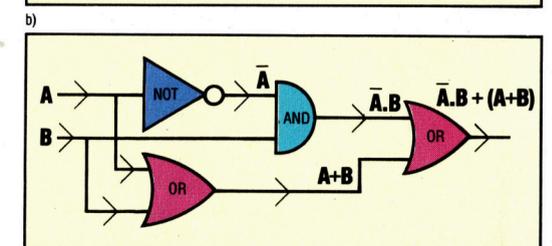
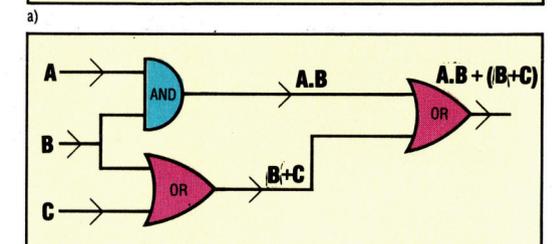
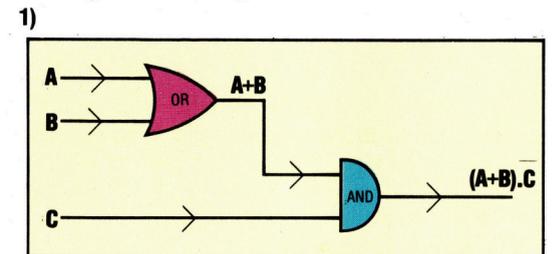
2) Eine Auto-Alarmanlage hat einen Ein/Ausschalter sowie Schalter an den vorderen Türen. Bei eingeschalteter Anlage und einer oder zwei offenen Türen gibt die Anlage Alarm. Zeichnen Sie eine Wahrheitstabelle mit drei Eingängen (Tür A, Tür B, Ein/Ausschalter) und einem (Alarm)-Ausgang. Schreiben Sie den Booleschen Ausdruck für die Funktion auf und zeichnen Sie die Logikschaltung dazu.

3) Eine Lampe im Treppenhaus soll mit drei Schaltern bedient werden, die an der Eingangstür sowie unten und oben an der Treppe liegen. Wie sieht die Logikschaltung aus?

4) Sie sind zu dritt auf einer einsamen Insel. Einer Ihrer Leidensgenossen spricht immer die Wahrheit, während der andere immer lügt. Wie können Sie herausfinden, wer die Wahrheit spricht? Die Wahrheitstabelle hilft Ihnen bei der Bewertung der Antworten. Beginnen Sie mit der Frage: „Sagst du immer die Wahrheit?“

		Mögliche Antworten	
		Ja	Nein
Was tut der Angesprochene	Lügt	1	0
	sagt die Wahrheit	1	0

Lösungen zu Übung 2



- 2a) $C = A.B$
- b) $S = A.B.(A + B)$
- 3a) $X = (A + B).(B.C + C)$
- b) $X = A.B + B$

Bildwiedergabe via Videobuchse

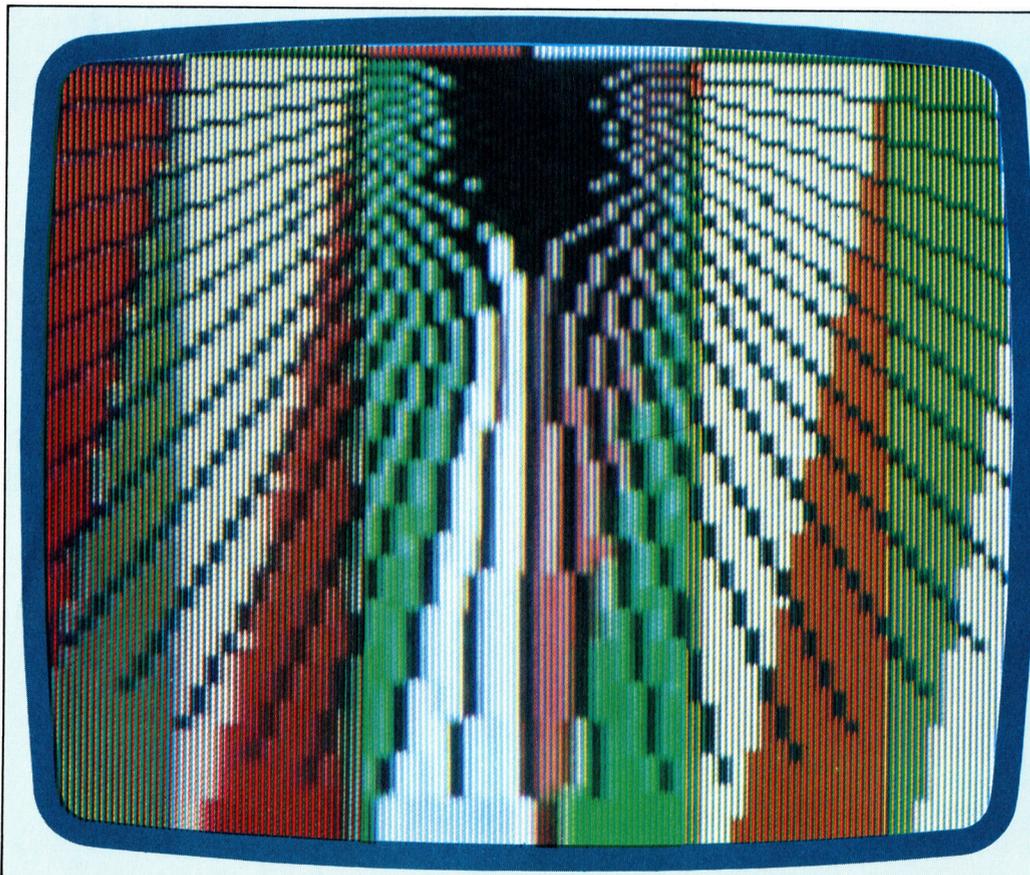
Sie können zwar jeden Heimcomputer an einen Fernsehapparat anschließen, aber ein Monitor liefert doch ein wesentlich besseres Bild. Als Alternative werden Kombinationsgeräte vorgestellt, die Monitor-Bildqualität und Programmempfang vereinen.

Die meisten Heimcomputerbesitzer gewöhnen sich schnell an das Geflimmer und die verschwommenen Farben, die ihr Rechner in Verbindung mit einem normalen Fernsehgerät erzeugt. Wer aber einen Monitor zur Verfügung hat, erlebt eine Art Offenbarung: brillante Farbwiedergabe und ein ruhiges Bild ohne Schattenbildung. Aber die höhere Bildqualität hat ihren Preis – Monitore sind teuer und ohne Zusatzgeräte für den Programmempfang unbrauchbar.

Es gibt allerdings eine Möglichkeit, nämlich die Fernsehempfänger/Monitor-Kombination, die beider Vorteile vereint. Dabei handelt es sich um ein Empfangsgerät mit einer Zusatzbuchse, über die bei Anschluß eines Rechners durchaus Monitorqualität erzielt werden kann. Vielleicht besitzen Sie ein solches Gerät, ohne

es zu wissen: Viele neuere Fernsehgeräte verfügen über eine Videorecorder-Buchse, an die ein Rechner angeschlossen werden kann.

Die Schwierigkeiten bei der Verwendung eines normalen Farbfernsehers als Ausgabegerät für den Computer beruhen zu einem erheblichen Teil auf der Art der Signalaufbereitung. Fernsehprogramme kommen über die Antenne als hochfrequentes Rundfunksignal an, aus dem der Bildinhalt (das Videosignal) im Empfänger durch Demodulation „herausgezogen“ werden muß. In einem Computer wird umgekehrt das Bildsignal durch einen Modulator (mit der Fernseher-Ausgangsbuchse verbunden) zunächst einem hochfrequenten Träger „aufmoduliert“, der vom Empfänger wie ein Antennensignal verarbeitet werden kann, um das Bild zu erzeugen. Bei dieser doppelten



Dreierlei Maß

Das Bildsignal kann vom Computer in drei Formen geliefert werden. Alle Rechner haben einen Fernseher-Antennenanschluss, der aber meist ein schlechtes Bild ergibt. Viele haben auch eine Buchse für einen Monitor. Fernseher/Monitor-Kombinationen bieten neben mehr Bildschärfe die Möglichkeit des normalen Programmempfangs. Verantwortlich für die unterschiedliche Bildqualität bei Fernsehern und Monitoren ist das jeweilige Bildsignal. Die nebenstehenden Bilder wurden auf demselben Empfänger/Monitor mit dreierlei Eingangssignalen fotografiert: Das TV-Signal (mittlerer Ausschnitt) liefert das schlechteste Bild; das Composite-Video-Signal ergibt schon eine deutliche Verbesserung (links), und das beste Bild (rechts) erhalten Sie mit dem RGB-Signal.

Umsetzung leidet die Wiedergabequalität erheblich. Ein Monitor verarbeitet dagegen das unmodulierte Bildsignal unmittelbar. Außerdem ist seine Bandbreite von meist 20 MHz wesentlich größer als die eines Fernsehgeräts, was eine höhere Detailauflösung bedeutet.

Beim Kauf eines Monitors ist zu beachten, mit welchem Bildsignal-Ausgang Ihr Rechner ausgestattet ist. Es kann sich um eine RGB-(Rot/Grün/Blau) oder um eine Composite-Video-Buchse handeln. Das RGB-System ist zwar vorzuziehen, aber auch das Composite-Video-Signal ist der Bildqualität des Fernsehers weit überlegen.

Auch bei Fernseher-Monitor-Kombinationen gibt es zwei Versionen – normale Fernseher mit nachgerüstetem Monitor-Eingang und Spezialentwicklungen. Letztere sind empfehlenswert, weil der Umbau der Standardmodelle oft ohne Wissen des Herstellers erfolgt und damit die Garantie erloschen sein kann. Die Spezialgeräte sind vorwiegend für den Betrieb mit Videorecordern ausgelegt und haben durchweg Composite-Video-Eingänge, die mit „Video“ oder „AV“ (Audio-Video) bezeichnet sind. Die Steckerskizzen auf der nächsten Seite zeigen, wie Ihr Rechner jeweils anzuschließen ist.

Vorteil: Tonwiedergabe

Ein großer Vorteil der kombinierten Fernseher/Monitore gegenüber den reinen Monitoren ist die Tonwiedergabemöglichkeit. Die Heimcomputer vieler Hersteller – vor allem von Atari, Commodore und Dragon – sind für die Tonausgabe auf einen Fernsehapparat angewiesen. Ein Standardmonitor hat kein Tonenteil, während bei den Fernseher/Monitor-Kombinationen Verstärker und Lautsprecher eingebaut sind.

Ist Ihr Rechner mit einem RGB-Ausgang ausgestattet, haben Sie weniger Auswahlmöglichkeiten. Drei der bekanntesten Systemgruppen sind: das „Profeel“-System von Sony, die Geräte mit Euro-AV-Buchse (beispielsweise von Nordmende) und das ITT-Modell. Der Sony-Profeel verarbeitet sowohl RGB- wie Composite-Video-Signale, hat aber keine Normbuchse. Das ITT-Gerät weist eine RGB-Buchse auf. Die Nordmende-Modelle sind bei Heimcomputerbesitzern wegen der Euro-AV-Buchse besonders verbreitet. Dabei handelt es sich um ein international genormtes Peripherie-Steckersystem, das sowohl für RGB- wie für Composite-Video-Signal eingerichtet ist.

Auch einige andere Fernsehgeräte sind mit Euro-AV-Buchse ausgerüstet – vielleicht gehört Ihrer dazu. Dabei muß allerdings eventuell zur Umschaltung von Fernseher- auf Monitorbetrieb der Euro-AV-Eingang umgesteckt werden. Andererseits ist das Euro-AV-System zukunftssicher, weil es RGB- und Composite-Video-Signale verarbeitet. Man sollte sich beim Kauf schon genau informieren.

Ferguson TX (mit RGB)

Ein Empfänger-Monitor mit zwei DIN-Buchsen für RGB- und Composite-Video-Signal. Die Bedienungsknöpfe für die Wahl der Betriebsart (Programmempfang und Monitorbetrieb) sind an der Frontseite angebracht.

Nordmende 1534

Eins der vielen Nordmende-Geräte mit Euro-AV-Buchse für Composite-Video oder RGB-Signal. Lieferbar sind sieben verschiedene Bildschirmformate, wobei zumeist noch nach Wunsch Hand- oder Fernbedienung erhältlich ist. Das abgebildete Gerät hat einen 33cm-Bildschirm.



Fidelity CM14

Sowohl als Empfänger/Monitor wie auch ohne Empfangsteil mit Euro-AV-Buchse für RGB- und Composite-Video-Signal erhältlich.



ITT RL2315
Für RGB- und Composite-Video-Signal sind getrennte DIN-Buchsen vorgesehen. Die Wahl des Eingangs erfolgt über einen Umschalter an der Rückseite, an den man nicht ganz leicht herankommt.

Anschlußsuche

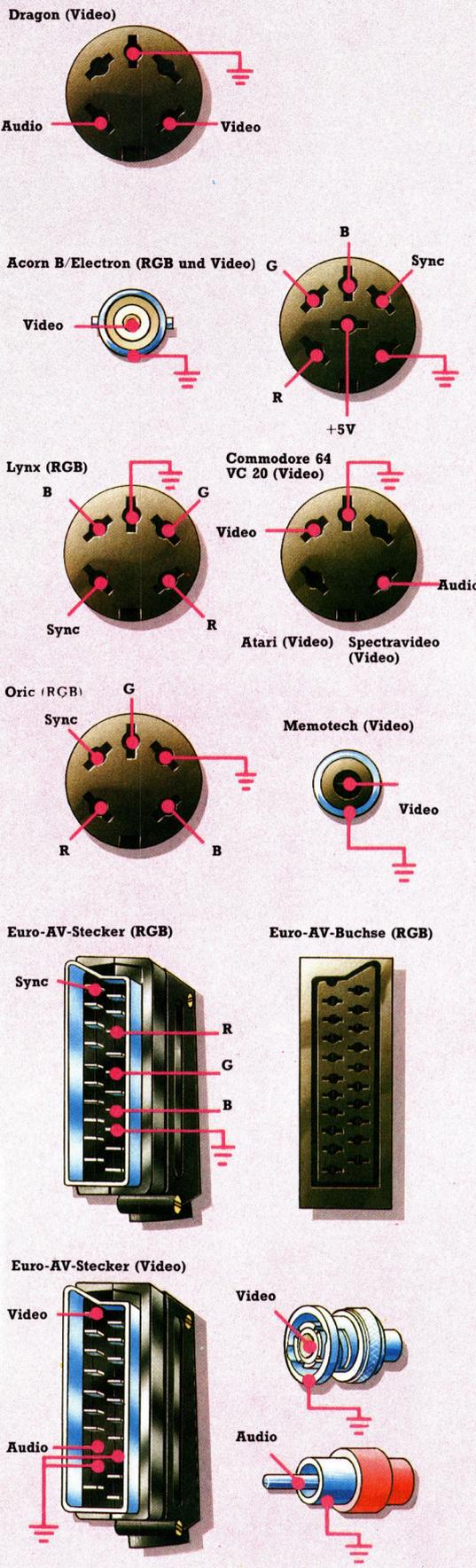
Monitore können mit RGB- oder mit Composite-Video-Signalen versorgt werden. Beim RGB-Signal kommen die Farbanteile Rot, Grün und Blau und das Synchronsignal auf vier getrennten Adern, und zwar meist über einen mehrpoligen DIN-Stecker. Beim Composite-Video-Signal sind alle Farbsignale und das Synchronsignal zu einem Mischsignal vereinigt, das im allgemeinen über Phono- oder BNC-Stecker zugeführt wird. Bei einigen Rechnern ist dagegen für Video- und Tonsignal ein gemeinsamer mehrpoliger Anschlußstecker vorgesehen.

Die Skizzen links zeigen die Stiftbelegung bei einigen Heimcomputern. Wenn bei Ihrem Empfänger/Monitor eine der unten abgebildeten Steckverbindungen vorgesehen ist, brauchen Sie nur ein Kabel mit den richtigen Steckern und entsprechenden Aderanschlüssen zu kaufen oder selbst zu löten, so daß beispielsweise R mit R verbunden ist.

Empfänger/Monitore mit Euro-AV-Buchse müssen über den Stecker von Monitorbetrieb auf Programmempfang umgeschaltet werden. Dazu sind im allgemeinen ein 5-Volt-Ausgang am Rechner und eine Schaltmöglichkeit erforderlich.

Der Sinclair Spectrum und der ZX81 sind hier nicht aufgeführt, weil sie keinen Monitor-Ausgang haben. Das Composite-Video-Signal läßt sich allerdings durch eine kleine Schaltungsänderung erzeugen. Für den Spectrum wird außerdem ein Steck-Adapter angeboten, der ein RGB-Signal liefert.

Der VC20 und die ersten 64er-Versionen sowie die Atari- und Spectravideo-Rechner haben alle die gleiche Monitorbuchse, während die neuen Commodore-Modelle mit einer achtpoligen DIN-Buchse ausgestattet sind. Dabei wird Pin 2 für die Masse, Pin 3 für den Ton und Pin 4 für das Video-Signal verwendet.





Schnelle Post

Einen Berg von Briefumschlägen mit Adressen versehen zu müssen, ist lästig genug. Mit einem Heimcomputer, einem Textprogramm und einer Datenbank läßt sich zumindest bei der Erstellung von Serienbriefen viel Zeit sparen.

Serienbriefprogramme können weit mehr als nur Adressen drucken, wobei eine reine Beschriftung von Aufklebern sich schon von einer Datenbank – ohne Textprogramm – ausführen läßt. Bei allen Systemen müssen jedoch zunächst Namen und Adressen Feld für Feld eingegeben werden. Einmal erstellte Daten lassen sich dann beliebig oft ausdrucken.

Komfortablere Datenbanken können ihre Listen nach unterschiedlichen Kriterien zusammenstellen und drucken: zum Beispiel alle Adressen mit der Ortsangabe „Hamburg“. Doch selbst wenn eine Datenbank die Adressierung der Briefumschläge übernimmt, wäre damit nur ein Teil der Aufgabe erledigt. Standardbriefe, die zum Beispiel einen bestimmten Kundenkreis erreichen sollen, müßten außerdem noch mit Adressen auf den Briefbögen versehen werden.

Doch trotz einer individuellen Adresse bleibt die Anonymität und das Unpersönliche des Standardtextes. Es hat wenig Sinn, im Text Lücken für den Namen des Kunden freizulassen, da alle Lücken so lang sein müssen wie der längste Name und ein „persönlicher“ Brief dieser Art ein seltsames Aussehen hätte. Wenn Sie also Briefe verschicken wollen, die persönlich ansprechen, müssen Sie ebenso viele Briefe tippen, wie Adressen in Ihrer Liste vorhanden sind.

„Persönliche“ Briefe vom Computer

Die ideale Lösung wäre ein System, das den Standardbrief über ein Textprogramm schreibt und aus einer Datenbank automatisch alle individuellen Einzelheiten aufruft. Mit einem derartigen System ließen sich problemlos jede Menge persönlicher Briefe schreiben.

Es werden eine ganze Reihe Systeme dieser Art angeboten. Darunter einfache Versionen wie das Textsystem „Memoplan“ und die Datenbank „Fileplan“ von Acorn. Beide Programmpakete arbeiten mit zwei Diskettenstationen und sind Teil einer Softwareauswahl, die Acorn beim Kauf der Z80-Erweiterung für den Acorn B kostenlos zur Verfügung stellt.

Zur Herstellung von Serienbriefen mit individuellen Daten müssen beide Programme allerdings gemeinsam eingesetzt werden. Über Fileplan werden dabei die Namen und Adressen erfaßt und mit Memoplan der Standardbrief entworfen. Da jedes Feld der Namens- und Adreßdatei numeriert ist, braucht der Anwender nur die entsprechende Feldnummer in den Standardbrief einzusetzen, um den Feldinhalt in den Brief mit einzubeziehen. Der Computer arbeitet sich sequentiell durch die Adreßliste und fügt die Feldinhalte der ausgewählten Datensätze in den Brief ein.

Da das Textsystem den Standardtext bei jedem Druck neu formatiert, sehen alle Briefe aus, als wären sie individuell geschrieben. Mit Memoplan können auch Meldungen in den Standardbrief eingesetzt werden, die den Inhalt eines bestimmten Feldes kennzeichnen.

Spezielle Programmpakete für Serienbriefe wie „Mailmerge“ von Micropro oder der „Mailing List Manager“ von Peachtree sind auf kommerzielle Computersysteme wie den IBM PC und den Sirius ausgerichtet. Sie verfügen

Tagesdatum
Wird von dem Programm automatisch gesetzt.

**Korrespondenz-
adresse**
Aus der Datenbank

Bezug
Aus der Datenbank

Name
Aus der Datenbank

Brief
Aus der Datenbank

Mazzini Associates Inc.
Hauptstrasse 13
Luzern 1
Schweiz
Betrifft Anfrage Nr.: WM/3258
Sehr geehrter Herr Mazzini,
wir danken für Ihren Brief vom 15. 8. 1985 und Ihre Einladung, uns an der Ausschreibung für den Bau des Gruetzi-Einkaufszentrums zu beteiligen.
Unsere Planungsabteilung ist dabei, ein Angebot zu erstellen, das wir innerhalb eines Monats an Ihre Geschäftsstelle in Luzern senden werden.

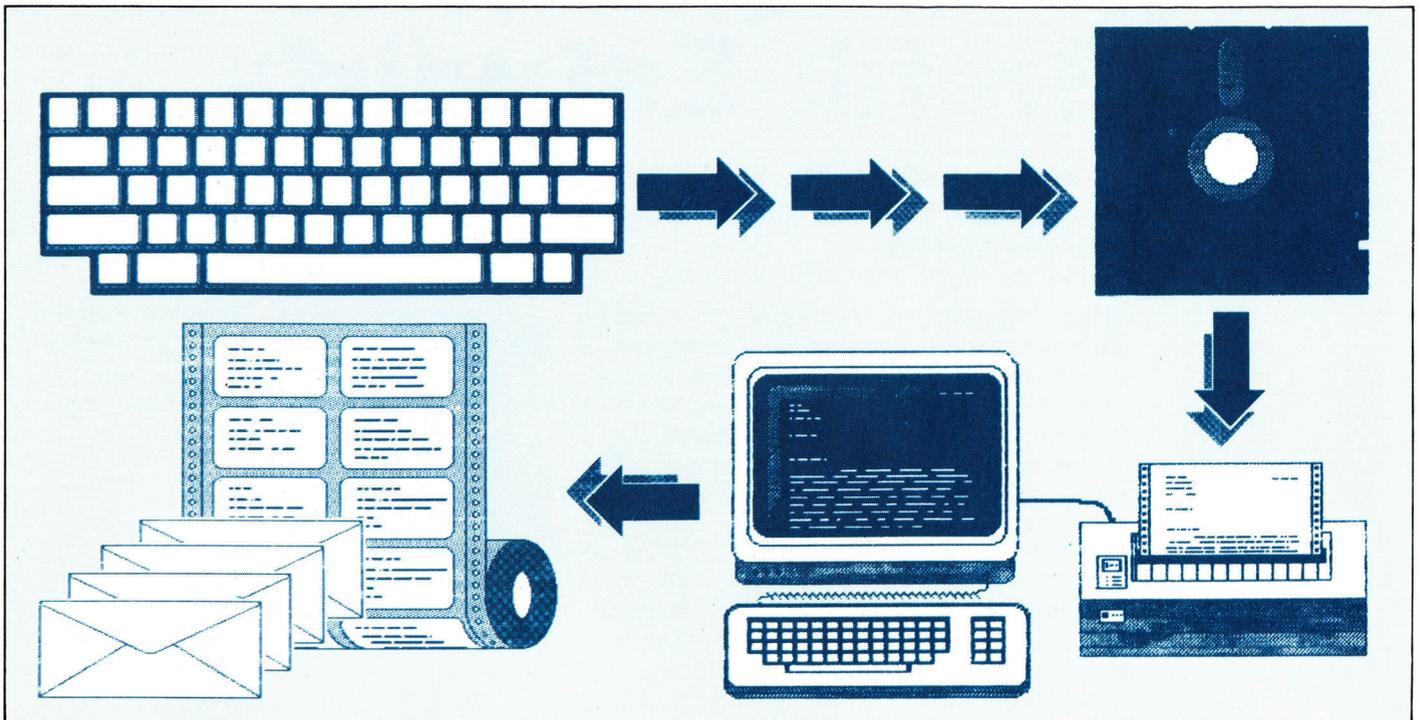
Mit freundlichen Grüßen

Manfred Terra
Geschäftsführer

Mönckebergstr. 555
2000 Hamburg 1
Hamburg, den 18. 08. 1985

**Korrespondenz-
datum**
Aus der Datenbank

Geschäftsstelle
Aus der Datenbank



über eine ganze Reihe ausgereifter, praktischer Fähigkeiten und können zum Beispiel nicht nur eine große Anzahl unterschiedlicher Adreßdateien aufbauen, sondern auch Spezialadressen schnell aus einer Hauptdatei herausfiltern.

In diesen Systemen werden mit den Standardtechniken der Datenbanken Schlüsselfelder festgelegt und Indizes aufgebaut, während logische Feldabfragen die Such- und Auswahl-funktionen übernehmen. Dabei kann der Anwender die Auswahlkriterien für die Adressen bestimmen.

Hochentwickelte Programmpakete können oft auch Spezialformate drucken. Diese Fähigkeit ist besonders interessant, wenn Namen und Adressen auf bestimmte Aufkleberformate ausgerichtet werden sollen. Auf dem „Peachtree“-System stellt zum Beispiel die Option LABEL FORMAT des Hauptmenüs einen Kasten und eine Aufstellung aller Felder der Adreßdatei dar, um dem Anwender einen Eindruck von dem späteren Druckformat des Aufklebers zu geben.

Nur in großen Mengen sinnvoll

Serienbriefprogramme sind nur bei größeren Dateien sinnvoll. Wenn Adressen nicht bereits im Computer gespeichert sind, läßt sich eine kleine Anzahl von Aufklebern oft schneller per Schreibmaschine beschriften. Alle kommerziellen Systeme und auch viele, die als ROM für Heimcomputer geliefert werden, speichern ihre Daten auf Disketten – nur wenige sind für Cassettenbetrieb und kleine Datenmengen ausgelegt.

Die in Cambridge ansässige Softwarefirma GCC liefert zum Beispiel ein Programmpaket

für den Acorn-B. GCC hat zuvor eine eigene Datenbank – „Starbase“ – angeboten, mit der sich Adreßlisten im Datenbankformat aufbauen ließen. Die Firma brachte anschließend eine 16K-ROM-Version von Starbase auf den Markt, die zusammen mit dem ROM-Textsystem „Wordwise“ ein Serienbriefsystem darstellt.

Individuelle Standardbriefe

Im Lieferumfang von Starbase sind außer dem Programm noch ein Handbuch und eine Diskette mit Hilfsmodulen enthalten. In Verbindung mit Wordwise lassen sich damit Standardbriefe individuell gestalten und Adreßaufkleber unterschiedlicher Formate beschriften. Die Hilfsprogramme können außerdem während der Adreßformatierung Befehle an den Drucker ausgeben und so eine Anzahl unterschiedlicher Schriftbilder erzeugen.

Da Starbase mit den Feldern der Adreßdateien mathematische Operationen ausführen kann, lassen sich damit auch Kontoauszüge und Rechnungen serienweise drucken. Dabei erledigt der Computer automatisch alle Berechnungen.

Die Besitzer eines Commodore 64 haben eine ganze Reihe von Briefsystemen zur Auswahl. Ein Beispiel ist „Visawrite“ von Visa Software, das nicht auf Disketten beschränkt ist, sondern auch mit Cassetten arbeiten kann. Das Programm läßt sich mit allen Datenbanken einsetzen, die sequentielle Dateien erstellen können, verfügt aber auch über ein eigenes Modul, das bis zu 500 Namen und Adressen speichern kann. Zur Einrichtung der Datei wird dabei jeder Datensatz als Textseite definiert und in den Standardbrief integriert.

Das Textsystem liefert einen Standardbrief, in dem an gekennzeichneten Positionen individuelle Informationen, wie etwa Datum, Name, Adresse, aus den Datensätzen der Datenbank eingefügt werden. Nach Fertigstellung der Briefe werden die Adressen auf selbstklebende Etiketten gedruckt.

Grafik-Entwürfe

In diesem Teil des Kurses erfahren Sie, wie man mathematische Diagramme mit einfachen Anweisungen in faszinierende Grafiken umsetzen kann.

Das Zeichnen des Diagramms eines mathematischen Ausdruckes ist ausgesprochen einfach. Man nimmt einfach einen Bereich an Werten für eine der Variablen des Ausdruckes und arbeitet dann die entsprechenden Werte der anderen Variable aus. Mit einer einfachen Formel, wie beispielsweise $Y=X^2$, kann man folgende Tabelle erstellen:

X	-5	-4	-3	-2	-1	0	1	2	3	4	5
Y	25	16	9	4	1	0	1	4	9	16	25

Zeichnet man die Punkte auf Millimeterpapier und verbindet sie dann mit einer durchgezogenen Kurvenlinie, entsteht die wohlbekannte Parabel. Diese Kurve repräsentiert die grafische Umsetzung der Funktion $Y=X^2$. Man könnte sie auch als Weg betrachten, der von einem bestimmten Punkt aus entlang der Funktion $Y=X^2$ verläuft. Dieser Weg wird auch „geometrischer Ort“ genannt. Durch Manipulation und Kombination verschiedener Orte kann man mit geringem Aufwand viele interessante Formen generieren.

Nehmen wir als erstes Beispiel einen Kreis. Die Formel dafür ist allerdings etwas komplizierter als die der Parabel. Sowohl der X- als auch der Y-Teil der Gleichung werden durch eine dritte Variable bzw. einen dritten Parameter definiert. Der Kreis entsteht durch folgende Gleichung:

$$X = R \sin(I)$$

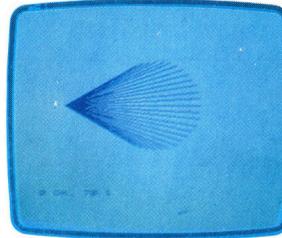
$$Y = R \cos(I)$$

Wenn wir eine Reihe von Werten für X und Y für den Fall ausarbeiten, daß der Winkel I einen kompletten Kreis umschreibt (Werte von 0° bis 360°), erhalten wir den geometrischen Ort des Kreises. Das folgende kurze Programm für den Spectrum erstellt einen Kreis.

```

10 REM ZEICHNEN EINES KREISES
20 LET xm=256: LET ym=176: LET xc=INT(xm/2): LET yc=INT(ym/2)
30 LET r=50
40 LET s=PI/20
50 FOR i=0 TO 2*PI STEP s
60 INK 2: PLOT xc+r*SIN(i), yc+r*COS(i)
70 NEXT i
    
```

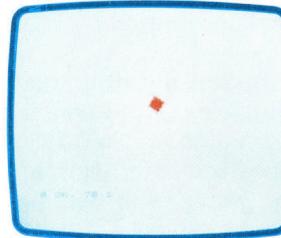
Durch Ändern der Schrittweite (STEP) zwischen den zu zeichnenden Punkten lassen sich die Definition des Kreises und die Geschwindigkeit, mit der er gezeichnet wird, variieren. Sie können diese Formel übrigens auch zum Zeichnen von Bögen und Ellipsen verwenden. Für Bögen wählen Sie Werte von I. Für Ellipsen geben Sie R in der X-Formel einen anderen Wert als in der Y-Formel.



Unser erstes Muster entsteht dadurch, daß man jeden Punkt der Kreisfigur mit einem festen Punkt verbindet.

```

10 REM flower
20 LET xm=256: LET ym=176: LET xc=INT(xm/2): LET yc=INT(ym/2)
30 LET r=50
40 LET s=PI/20
50 FOR i=0 TO 2*PI STEP s
60 INK 2: PLOT xc+r*SIN(i), yc+r*COS(i)
70 NEXT i
    
```



Die folgenden Programme positionieren den Punkt im Zentrum, beziehungsweise auf der linken Seite des Kreises.

```

10 REM Circle & Fixed point
20 LET xm=256: LET ym=176: LET xc=INT(xm/2): LET yc=INT(ym/2)
30 LET r=50
40 LET s=PI/20
50 FOR i=0 TO 2*PI STEP s
60 INK 2: PLOT xc+r*SIN(i), yc+r*COS(i)
70 NEXT i
    
```



Der nächste Schritt ist, einen beweglichen Punkt anstelle eines festen Punktes zu verwenden. Wir werden zwei Kreisfiguren simultan zeichnen und jeweils die beiden zusammengehörigen Punkte mit Linien verbinden.

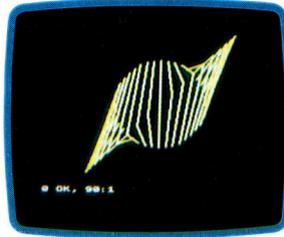
```

10 REM Nested circles
20 LET xm=256: LET ym=176: LET xc=INT(xm/2): LET yc=INT(ym/2)
30 LET r=50
40 LET s=PI/20
50 FOR i=0 TO 2*PI STEP s
60 LET x=xc+(r+30)*SIN(i): LET y=yc+(r+30)*COS(i)
70 LET p=xc+r*SIN(i): LET q=yc+r*COS(i)
80 INK 2: PLOT x,y: DRAW p-x,q-y
90 NEXT i
    
```



Das Programm, das wir jetzt entwickelt haben, enthält genug Informationen, um Hunderte von verschiedenen Mustern zu zeichnen. Eine einfache Variation ist, SIN und COS in einer

der Formeln auszutauschen. Alternativ kann man auch die Werte von SIN und COS im Quadrat verwenden und so die unterschiedlichsten Effekte bewirken.



herumexperimentieren. Einige einfache Modifikationen des Programms ermöglichen sogar, daß der Computer selbständig mit verschiedenen Mustern experimentiert.

In einem normalen Mathematik-Lehrbuch finden Sie zahlreiche Formeln zur Generierung verschiedener Kurven. Andererseits werden Sie sicher mehr Spaß daran haben, wenn Sie mit Ihren eigenen Ideen

BASIC-Dialekte

Die hier gezeigten Programme laufen auf einem 16-KByte- und 48-KByte-Spectrum. Trotzdem ist eine Verwendung der Ideen auf einem anderen Computer sehr einfach. Die einzigen Voraussetzungen sind, daß Ihr Computer über hochauflösende Grafik (mindestens 286x176), Fließkomma-BASIC mit den Funktionen SIN und COS und einen Befehl zum Zeichnen von Punkten und geraden Linien verfügt.

Die ersten notwendigen Änderungen dürften bei XM und YM auftreten. Dies sind die Variablen, die die maximalen X- und Y-Werte enthalten, die auf Ihrem Computer ansteuerbar sind. Abhängig von den von Ihnen verwendeten Funktionen kann es sein, daß Sie auch andere Konstanten, wie beispielsweise R und S, ändern müssen. Als nächstes müssen Sie sicherstellen, daß Ihr Computer sich in einem entsprechenden Grafik-Modus befindet. Außerdem müssen Sie eine Farbe zum Zeichnen auswählen. Abschließend brauchen Sie noch einen Befehl zum Zeichnen der Linien zwischen den Koordinaten X und Y sowie P und Q. Beim Spectrum wird dies durch PLOT gefolgt von DRAW bewerkstelligt. Der DRAW-Befehl ist beim Spectrum etwas kompliziert, da er sich jeweils auf den ersten mit PLOT gezeichneten Punkt bezieht, wogegen wir in unserem Beispiel die Linie zu einer bestimmten absoluten Position zeichnen müssen. Die meisten Computer verfügen über eine absolute Funktion zum Zeichnen von Linien, wodurch diese Aufgabe erheblich leichter zu lösen ist.

DRAGON 32/64 Mit dem Modus PMODE 4 des Dragon erhalten Sie ein Raster von 286x192 Punkten, das für die hier gezeigten Programme gut verwendbar ist. Benutzen Sie SCREEN 1,0 oder SCREEN 1,1 für eine grüne oder blaßgelbe Hintergrundfarbe. Der LINE-Befehl kann zum Zeichnen der Linien verwendet werden (LINE(X,Y)-(P,Q),PSET).

COMMODORE 64/VC 20 Diese Computer verfügen zwar über eine akzeptable hochauflösende Grafik, jedoch leider nicht über entsprechende Befehle zur Handhabung. Um die Programme trotzdem verwenden zu können, müssen Sie entweder Ihre eigenen Befehle zum Zeichnen von Punkten und Linien entwickeln, oder eine BASIC-Erweiterungs-Cartridge, wie beispielsweise Simon's BASIC, verwenden.

EINIGE IDEEN

- 1) Erinnern Sie sich an das einfache Beispiel, mit einer FOR...NEXT-Schleife einen Kreis zu zeichnen. Wir zeigten Ihnen, wie man mit demselben Programm Ellipsen und Bögen entwickeln kann. Machen Sie sich jetzt einmal Gedanken, wie man das Programm modifizieren muß, um Spiralen zu zeichnen.
- 2) Verwenden Sie zum Zeichnen weiterer Figuren andere Funktionen wie zum Beispiel SQR und TAN.
- 3) Entwickeln Sie animierte Versionen der hier gezeigten Programme, indem Sie Bereiche zum Speichern der letzten fünf gezeichneten Linien DIMensionieren. Diese Linien sollten sich um zwei geometrische Orte bewegen.
- 4) Wie wäre es mit Bildern, die auf drei geometrischen Orten basieren? Gestalten Sie zwei davon sehr einfach (vielleicht einen Kreis und eine gerade Linie), damit das Bild nicht durch zu viele Linien „überladen“ wird.

```
10 REM Twisted circles
20 LET xm=256: LET ym=176: LET xc=INT
(xm/2): LET yc=INT (ym/2)
30 LET r=50
40 LET s=PI/20
50 FOR i=0 TO 2*PI STEP s
60 LET x=xc+(r+30)*COS (i): LET y=yc+
(r+30)*SIN (i)
70 LET p=xc+r*SIN (i): LET q=yc+r*COS
(i)
80 INK 2: PLOT x,y: DRAW p-x,q-y
90 NEXT i
```



```
10 REM Circle & SIN
20 LET xm=256: LET ym=176: LET xc=INT
(xm/2): LET yc=INT (ym/2)
30 LET r=50
40 LET s=PI/20
50 FOR i=0 TO 2*PI STEP s
60 LET x=xc+SIN (i)*80: LET y=yc+(r+3
0)*SIN (i)
70 LET p=xc+r*SIN (i): LET q=yc+r*COS
(i)
80 INK 2: PLOT x,y: DRAW p-x,q-y
90 NEXT i
```



```
10 REM Circle & SIN*COS
20 LET xm=256: LET ym=176: LET xc=INT
(xm/2): LET yc=INT (ym/2)
30 LET r=50
40 LET s=PI/20
50 FOR i=0 TO 2*PI STEP s
60 LET x=xc+r*SIN (i)*COS (i): LET y=
yc+r*SIN (i)*COS (i)
70 LET p=xc+r*SIN (i): LET q=yc+r*COS
(i)
80 INK 2: PLOT x,y: DRAW p-x,q-y
90 NEXT i
```



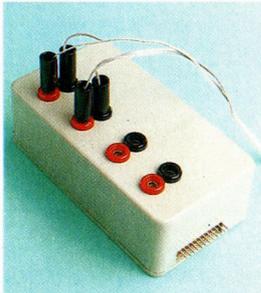
```
10 REM Random circle patterns
15 RANDOMIZE
20 LET xm=256: LET ym=176: LET xc=INT
(xm/2): LET yc=INT (ym/2)
30 LET r=60: LET s=PI/20
35 CLS
40 LET c=INT (RND*4)+1: LET d=INT (RN
D*4)+1
50 FOR i=0 TO 2*PI STEP s
60 LET x=xc+r*SIN (i/c)*COS (i*d): LE
T y=yc+r*SIN (i/e)*COS (i*f)
70 LET p=xc+r*SIN (i): LET q=yc+r*COS
(i)
80 PLOT x,y: DRAW p-x,q-y
90 NEXT i
100 IF INKEY$="" THEN GO TO 100
110 GO TO 35
```





Schaltkasten

Wir setzen den Selbstbau-Kurs mit einer Bauanleitung fort: Das neue Interface ermöglicht die Computer-Steuerung von Glühlampen und kleinen Elektromotoren.



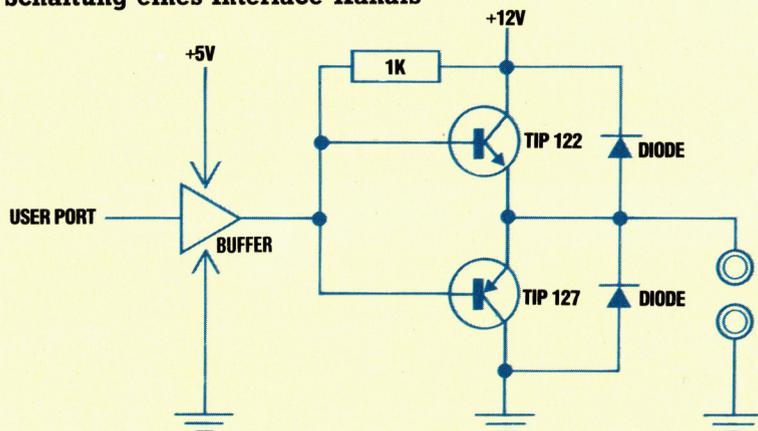
Dies ist das fertige Interface mit Minicon-Anschluß und Zuleitungen. Die Platine muß sorgfältig zugeschnitten werden, damit sie sich beim Einsetzen des Steckers nicht im Gehäuse verschiebt.

Durch das Zusatzgerät kann Ihr Heimcomputer Niedervolt-Anlagen mit geringer Stromaufnahme steuern. Dazu werden alle vier Bit des User Ports mit dem Buffer-IC gesichert, das Sie schon von der Konstruktion des Ausgangs-Buffers her kennen. Der Ausgangsstrom des IC wird durch zusätzliche Schalttransistoren vergrößert. Die Komplementärschaltung der Transistoren macht sowohl Stromaufnahme als auch -abgabe möglich. Durch zwei Schutzdioden im Ausgang ist sichergestellt, daß die Transistoren nicht von der induktiven Last eines Relais oder Motors zerstört werden.

Das neue Interface kann unter anderem auch als bidirektionale Motorsteuerung verwendet werden. Der Motor wird zwischen Masse und Ausgang geschaltet und läuft, wenn der Portanschluß ein „High“ (5 Volt) ausgibt. Bei „Low“ (0 Volt) bleibt er wieder stehen.

Durch Anschluß des Motors zwischen zwei Interface-Ausgängen läßt sich die Drehrichtung ändern: Der Motor steht, solange der Rechner für beide Anschlüsse das gleiche ausgibt (High bzw. Low) – es fließt kein Strom. Erst bei einer 1 für den einen Ausgang und einer 0 für den anderen ergibt sich ein Spannungsunterschied, der nun den Motor in Betrieb setzt.

Schaltung eines Interface-Kanals



Interface – Montage

Als erstes wird das Interface-Gehäuse hergestellt. Dazu müssen Sie Löcher für die Buchsen und den Bus-Anschluß (evtl. auch für den Minicon-Anschluß) bohren.

Nach dem Einbau der Anschlußbuchsen werden diese verdrahtet. Mit einem Stück verzinnter Litze werden zuerst die Masse-Anschlüsse (schwarz) verbunden. Daran wird die gekennzeichnete Leitung eines 9-adrigen Flachkabels festgelötet, und die anderen Leitungen werden jeweils paarig an den vier roten Buchsen befestigt.

Schneiden Sie sich die Lochplatine (Veroboard) auf die richtige Größe zu (45 Löcher \times 16 Streifen) und entfernen Sie einseitig eine halbe Lochreihe. (Abschnitt aufbewahren, er wird später noch gebraucht.) Einzelne Leiterbahnen müssen nach Abbildung „A“ unterbrochen werden.

Zuerst die passiven Bauteile – IC-Sockel, Bus-Steckanschluß und Drähte – einlöten. Wenn Sie auch die Bus-Erweiterung einbauen, schließen Sie deren Leitung jetzt noch nicht an.

Als nächstes werden Widerstände und Dioden eingelötet – achten Sie dabei auf die Dioden-Durchlafrichtung! Jetzt fehlen noch die Transistoren – auch dabei ist die Richtung einzuhalten. Zuletzt werden die Leitungen festgelötet, das IC eingesetzt (Bild „B“) und das Gehäuse verschraubt – fertig!

Liste der Bauteile

Anzahl Bauteile

1	Lochplatine, 50 Löcher \times 24 Streifen (Veroboard)
4	Widerstände, 1 kOhm
8	Dioden Typ 1N4001
4	Transistoren TIP 122
4	Transistoren TIP 127
1	IC 7407
1	IC-Sockel, 14-polig
4	Telefonbuchsen, rot, 4 mm
4	Telefonbuchsen, schwarz, 4 mm
4	Bananenstecker, rot, 4 mm
4	Bananenstecker, schwarz, 4 mm
1	Minicon-Stecker, 12-polig
1	Gehäuse, 120 \times 65 \times 40 mm
1	Minicon-Anschluß, 12-polig*

Einen Rest Flachkabel und Litze haben Sie sicher noch von früheren Bastelarbeiten.
* Das letzte Bauteil wird jetzt noch nicht unbedingt gebraucht. Es dient zum späteren Anschluß von Interface-Erweiterungen.



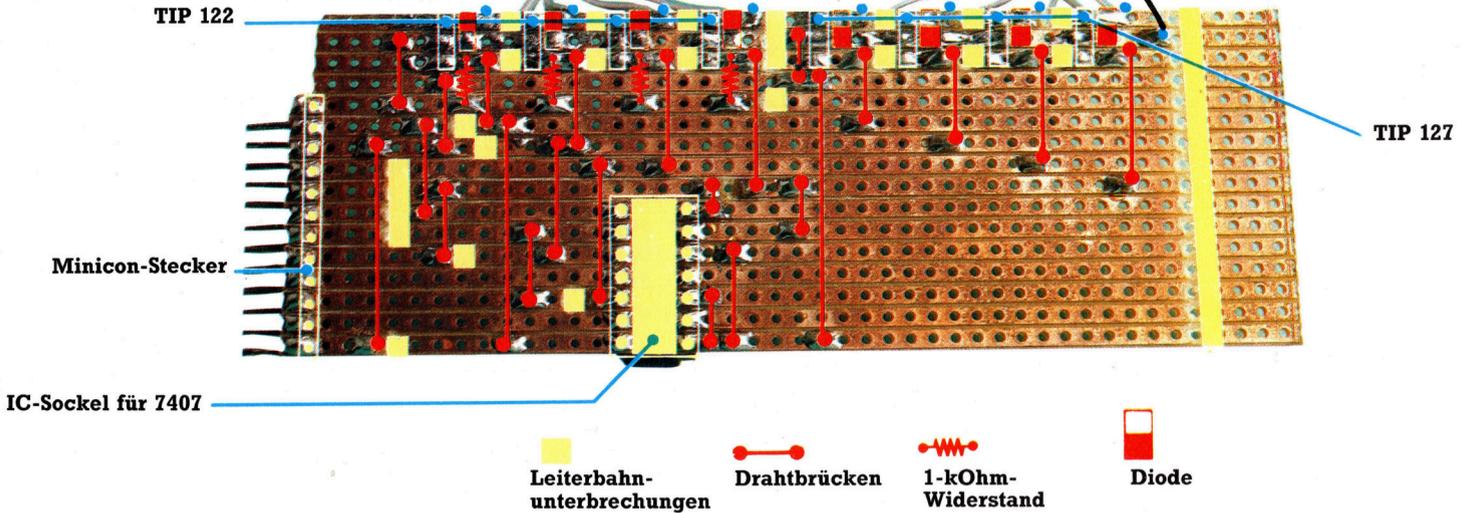
Leiterbahnseite

Achtung bei den Leiterbahnunterbrechungen! Besonders bei den Transistor-Anschlußpunkten dürfen Sie keine Unterbrechung vergessen!

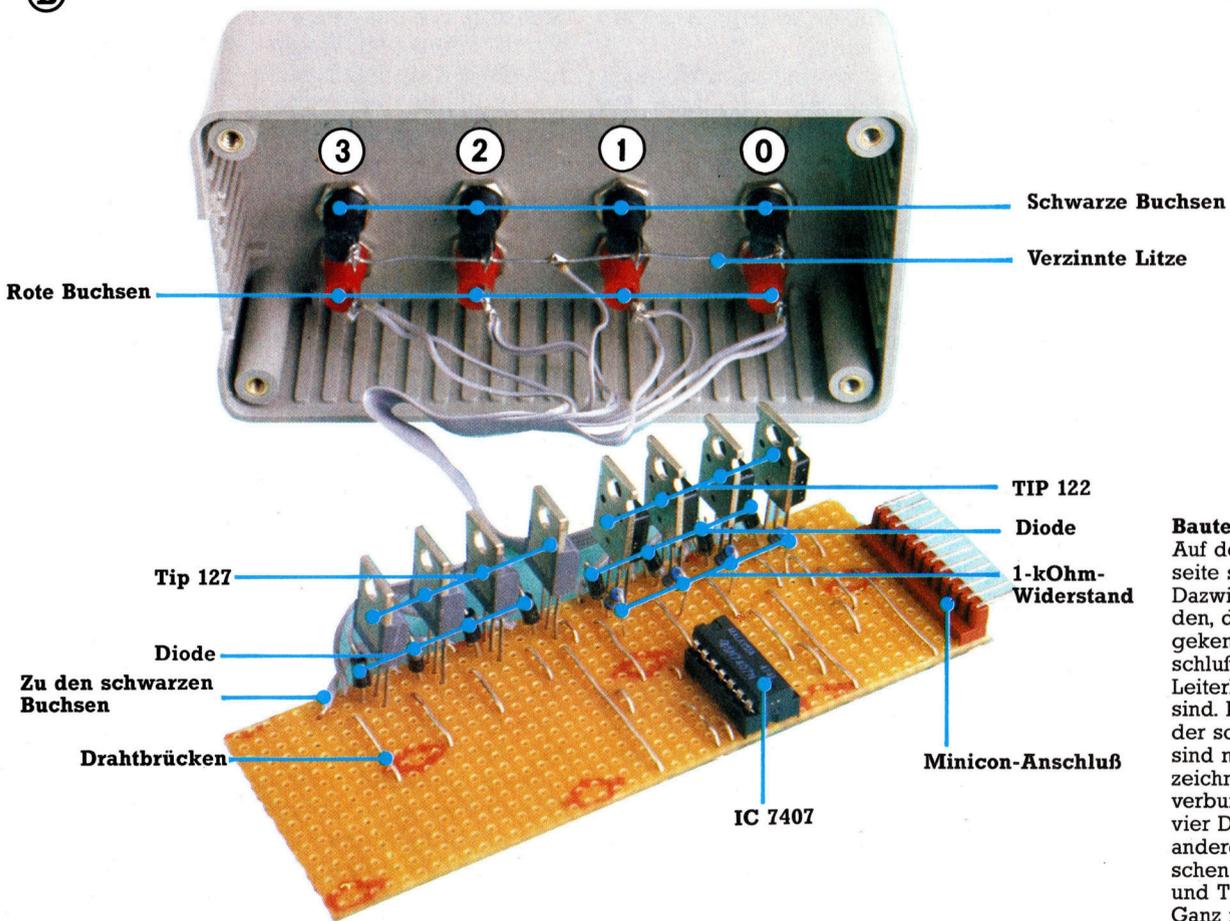
Zu den roten Buchsen

Zu den schwarzen Buchsen

Ⓐ



Ⓑ



Bauteilseite

Auf der rechten Platinenseite sitzen die TIP 122. Dazwischen die vier Dioden, die mit dem nicht gekennzeichneten Anschluß an der seitlichen Leiterbahn festgelötet sind. Die Anschlußdrähte der schwarzen Buchsen sind mit der gekennzeichneten Diodenseite verbunden. Die restlichen vier Dioden liegen in der anderen Richtung zwischen Anschlußleitungen und TIP 127-Transistoren. Ganz rechts an der Platine befindet sich der Minicon-Stecker.



Adreßregister

Bisher wurde untersucht, wie die CPU den Speicher über Register wie den Akkumulator und die ALU anspricht. Diesmal gehen wir auf die grundlegenden mathematischen Operationen ein.

Die Unterschiede zwischen den Microprozessoren Z80 und 6502 zeigen sich am deutlichsten in der völlig andersartigen Ausführung einfacher mathematischer Vorgänge. Die vielen Register und der hochentwickelte Satz von Befehlen kennzeichnen den Z80 als komplex und flexibel, während der 6502 durch die Einfachheit seiner Struktur und den unkomplizierten Befehlssatz eher ein simpler Prozessor zu sein scheint, der zwar robust und praktisch ist, aber nicht die Klasse des Z80 besitzt. Obwohl dieser Eindruck teilweise durchaus berechtigt sein kann, besitzt der 6502 mit seinen vielen Adressierungsmodi und seinem Einsatz der Zero-Page als zusätzlichem Index-Register doch eine Flexibilität, mit der er auf dem Markt für Computer noch lange Zeit eine wichtige Rolle spielen wird.

Der Z80 besitzt einen wesentlichen Vorteil in der Anpassungsfähigkeit seiner Register. Sie lassen sich wahlweise als Zwei-Byte- und als Ein-Byte-Register einsetzen und verfügen damit über einen enormen Adreßraum. Doch obwohl der 6502 keine Zwei-Byte-Register besitzt, kann er über bestimmte Adressierungsarten die Zero-Page als eine Matrix von Ein-Byte- und Zwei-Byte-Registern einsetzen.

Wir haben gesehen, daß die CPU-Register viele Möglichkeiten bieten, den Speicher zu adressieren. Für die Bearbeitung des Speichers sind normalerweise jedoch andere Vorgänge wichtig als das Laden, Speichern und Vergleichen einzelner Bytes. So spielt die Ausführung der vier Grundrechenarten in einem Computersystem eine wesentliche Rolle. Da die Befehlssätze des Z80 und des 6502 jedoch nur Addition und Subtraktion unterstützen, müssen Multiplikation und Division sowie die Addition und Subtraktion von Zahlen größer als \$FF programmiert werden. Diese Begrenzung beider CPUs und die damit verbundene Notwendigkeit, Algorithmen für Multiplikation und Division erarbeiten zu müssen, kann für den Programmierer großen Lernwert haben.

Mit den Befehlen ADC und INC hatten wir auf beiden CPUs eine Ein-Byte-Arithmetik ausgeführt. Die folgenden Programme addieren den Inhalt zweier Zwei-Byte-Speicherstellen auf zwei unterschiedliche Weisen:

6502	Z80
ADDR1 DW \$7E60	ADDR1 DW \$7E60
ADDR2 DW \$4A51	ADDR2 DW \$4A51
SUM DS \$03	SUM DS \$03
BEGIN CLC	BEGIN LD A,\$00
LDA ADDR1	AND A
ADC ADDR2	LD HL,(ADDR1)
STA SUM	LD DE,(ADDR2)
LDA ADDR1+1	ADD HL,DE
ADC ADDR2+1	LD (SUM),HL
STA SUM+1	ADC A,\$00
LDA \$00	LD (SUM+2),A
ADC \$00	RET
STA SUM+2	
RTS	

Während die Ein-Byte-Methode des 6502 auch auf dem Z80 funktioniert, läßt sich die Methode des Z80 nicht auf dem 6502 einsetzen, da Registerpaare verwendet werden. Beachten Sie die Strategien, mit denen ein möglicher Übertrag abgefangen wird. Zwischen CLC (6502) und AND A (Z80), die das Übertragsflag vor der Addition auf 0 setzen, und der Veränderung des dritten Bytes von SUM befindet sich eine ganze Reihe von Vorgängen, die das größtmögliche Ergebnis berücksichtigen.

Der Subtraktionsvorgang ähnelt der Addition, da beide Prozessoren über einen SBC („Subtrahieren mit Übertrag“) -Befehl verfügen, wobei die Zwei-Byte-Subtraktion allerdings nur auf dem Z80 unterstützt wird. Da hier ein negatives Ergebnis möglich ist, müssen wir untersuchen, wie ein algebraisches Zeichen binär dargestellt werden kann.

Zunächst brauchen wir über negative Zahlen nur folgende Aussage zu treffen:

$$\text{Wenn } A + B = 0, \text{ dann } A = -B$$

Falls A eine positive Zahl darstellt, dann ist ihre Umkehrung (oder Komplement) die Zahl, die zu A addiert Null ergibt. So ist zum Beispiel das Ein-Byte-Komplement der Ein-Byte-Zahl \$04 die Zahl \$FC:

$$\$04 + \$FC = \$100$$

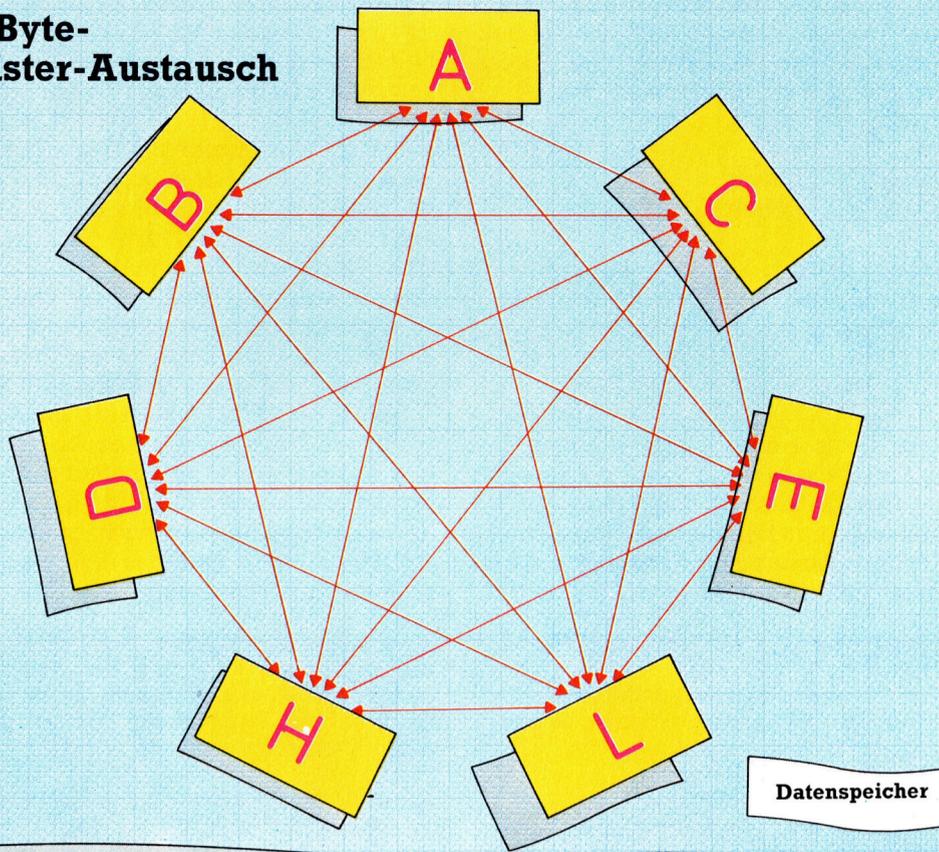
Bedenken Sie hierbei, daß \$100 = \$00 ergibt, falls nur ein Ein-Byte-Register für das Ergebnis zur Verfügung steht. Durch die Komplementdarstellung läßt sich eine Subtraktion auch als eine Addition mit negativem Vorzeichen ansehen, wie unser Beispiel verdeutlicht:

$$A - B \text{ bedeutet dasselbe wie } A + (-B)$$

\$08-\$05 ist daher dasselbe wie \$08 + (-\$05). Da (-\$05) sich aber auch als \$FB schreiben läßt (\$FB + \$05 ergibt \$100), kann unsere ursprüngliche Subtraktion als \$08 + \$FB dargestellt werden. Die Summe ist \$103 und im Ein-Byte-Format \$03.

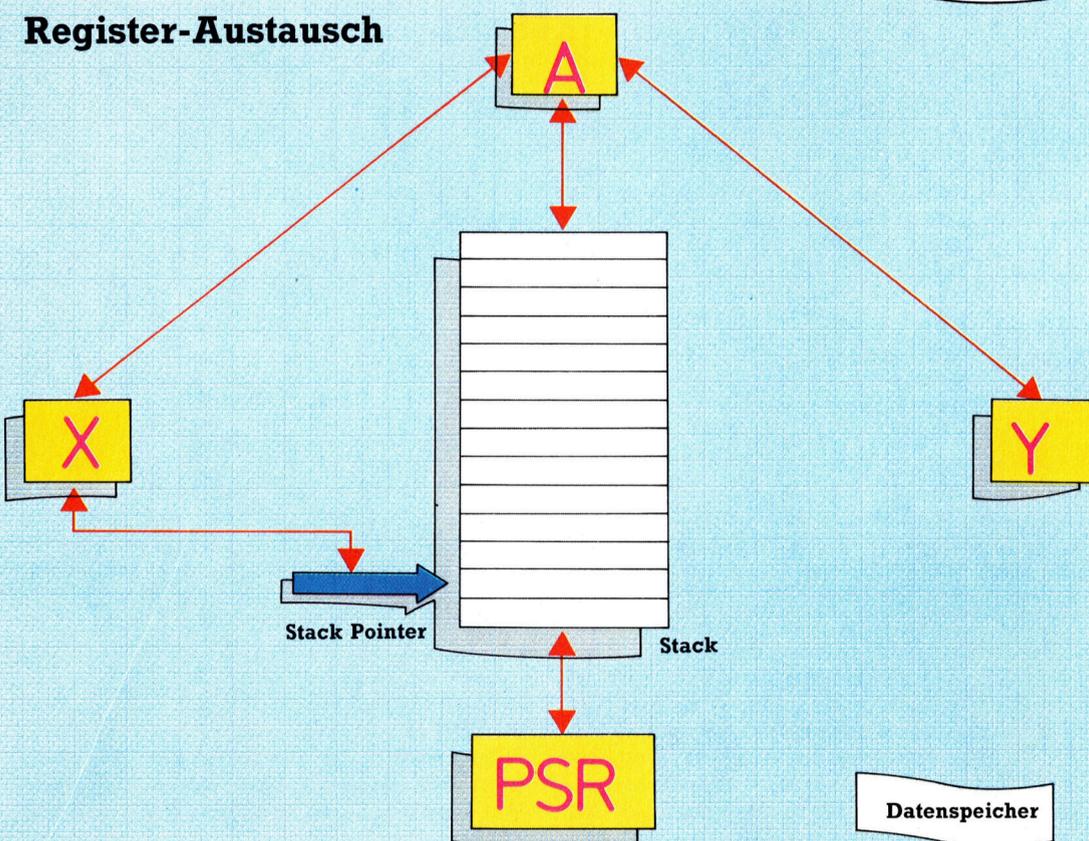


Ein-Byte-Register-Austausch



Die Datenregister des Z80 haben als Ein-Byte-Register die Möglichkeit, mit jedem anderen Ein-Byte-Register Daten auszutauschen. Sie können den Speicher in der direkten, unmittelbaren, indirekten, absoluten und indizierten Adressierung ansprechen. Bei ihrem Einsatz als Zwei-Byte-Registerpaare BC, DE, HL können sie 16-Bit-Daten mit dem Speicher und dem Stack austauschen und auch als 16-Bit-Akkumulatoren für Addition und Subtraktion dienen. Diese vielfältigen Einsatzmöglichkeiten sind der Schlüssel zu dem großen Erfolg des Z80.

Register-Austausch



Die interne Kommunikation des 6502 ist streng linear und beschränkt sich auf die Datenübertragung im Acht-Bit-Format. Nur der Akkumulator kann direkt mit X und Y Daten austauschen. Nur X kann den Stack-Pointer ansprechen, und nur das PSR und der Akkumulator können auf den Stack zugreifen. Der Speicher läßt sich auf absolute, direkte, indirekte, indizierte, unmittelbare Art und über die Zero-Page ansprechen. Der flexible Einsatz der Zero-Page gleicht die geringe Kapazität des Registersatzes aus. Sie stellt 128 CPU-Register im Zwei-Byte-Format zur Verfügung.



Diese Art der Darstellung wird Zweier-Komplement genannt. Das Zweier-Komplement einer Ein-Byte-Zahl entsteht durch die Subtraktion dieser Zahl von \$100. Einer- und Zweier-Komplement stehen in folgender Beziehung zueinander:

$$\begin{array}{r} \$05 = 0000101 \text{ binär} \\ \$FA = 11111010 \text{ Einer-Komplement} \\ \quad +1 \\ \hline \$FB = 11111011 \text{ Zweier-Komplement} \end{array}$$

$$\$05 + \$FA = \$FF$$

$$\$05 + \$FB = \$00$$

Das Einer-Komplement einer Ein-Byte-Zahl entsteht durch Umkehrung jedes binären Bits der Zahl. Wird Eins zu dem Ergebnis addiert, entsteht das Zweier-Komplement. Die Addition einer Zahl und seines Einer-Komplements ergibt immer \$FF, während eine Zahl und ihr

Zweier-Komplement als Ergebnis immer \$00 (eigentlich \$100) haben. In der Mathematik der ganzen Zahlen mit Vorzeichen ist es daher üblich, die Zahlen von \$00 bis \$7F als positiv anzusehen (0 bis 127) und \$80 bis \$FF als negativ (-128 bis -1). In der Binärdarstellung dieser Zahlen werden Sie bemerken, daß alle negativen Ganzzahlen das Bit 7 gesetzt haben, während bei den positiven Zahlen Bit 7 immer auf 0 steht. Bit 7 wird daher auch als Vorzeichen-Bit bezeichnet, wobei das Übertragsflag des Prozessor-Status-Registers als Kopie von Bit 7 gesetzt oder zurückgesetzt wird und damit über das Ergebnis des letzten arithmetischen oder logischen Vorgangs Auskunft gibt.

Dieses recht komplizierte Thema läßt sich leider nicht umgehen, wenn Sie mit positiven und negativen Werten arbeiten wollen. Glücklicherweise können diese Vorgänge mit wenigen einfachen Regeln bewältigt werden, wenn die Grundlagen und das System der Algorithmen einmal verstanden sind.

Lösungen

1) Das folgende Programm kehrt die Reihenfolge der in LABL1 gespeicherten Zeichenkette um:

```

6502
;
ORIGIN ORG $7000
LAST1 EQU $0D
LABL1 DB 'THIS IS A MESSAGE'
TERMN8 DB LAST1
;
BEGIN LDX #$FF
      LDA #LAST1
      PHA
LOOP0 INX
      LDA LABL1,X
      PHA
      CMP #LAST1
ENDLP0 BNE LOOP0
CLRSTK PLA
;
BEGIN1 LDX #$FF
LOOP1 INX
      PLA
      STA LABL1,X
      CMP #LAST1
ENDLP1 BNE LOOP1
      RTS

```

In der Version des 6502 setzt der Code zwischen LOOP0 und ENDLOOP0 in einer Schleife die X-indizierte Adressierung ein, um die Zeichen nacheinander von LABL1 zu laden und auf den Stack zu schieben, nachdem der ASCII-Wert der Endkennung zuvor die Untergrenze des Stacks markiert hatte. Das letzte auf den Stack geschobene Zeichen ist ebenfalls eine Endkennung. Diese wird jedoch durch ihre Position als letztes Zeichen des Strings erzeugt. Damit ist die Schleife abgeschlossen, und das Zeichen mit der Endkennung auf der Spitze des Stacks wird mit CLRSTK gelöscht. In der Z80-Version wird IX in der indirekten Adressierung eingesetzt, um den Akkumulator

von LABL1 an zu laden. Die Zeichen des Strings LABL1 wechseln sich daher auf dem Stack mit den fortlaufenden Werten des Prozessor-Status-Registers ab.

```

Z80
      ORG $C000
LAST1 EQU $0D
LABL1 DB 'THIS IS A MESSAGE'
TERMN8 DB LAST1
;
BEGIN LD IX,LABL1-1
      LD A, LAST1
      PUSH AF
LOOP0 INC IX
      LD A,(IX+0)
      PUSH AF
      CP LAST1
ENDLP0 JR NZ,LOOP0
CLRSTK POP AF
;
BEGIN1 LD IX,LABL1-1
LOOP1 INC IX
      POP AF
      LD (IX+0),A
      CP LAST1
ENDLP1 JR NZ,LOOP1
      RET

```

In beiden Versionen baut der Code zwischen BEGIN1 und ENDLP1 auf Techniken auf, die schon in früheren Programmen eingesetzt wurden. Diesmal wird jedoch der String in umgekehrter Folge von dem Stack gezogen und von LABL1 an aufwärts gespeichert. Die Schleife ist beendet, wenn das Zeichen der Endkennung auf dem Boden des Stacks gefunden wird.

Es ist wichtig, die Befehle für das Laden und Lesen des Stacks gegeneinander auszugleichen. Die Extrembedingungen sind das größte Problem. Wie sollen Anfang und Ende der Schleife aussehen, und welche „Aufräumarbeiten“ (falls überhaupt) sind notwendig?



Der Z80-Befehl bei BEGIN und BEGIN1 (LD IX, LABEL1-1) macht deutlich, wie nützlich ein Assembler-Programm sein kann. Der Ausdruck (LABEL1-1) bedeutet hier „die Adresse des Bytes unmittelbar vor dem Byte, dessen Adresse LABEL1 ist“. Der Assembler übersetzt diese Anweisung automatisch in eine Adresse des Maschinencodes. Die meisten Assembler können Ausdrücke verarbeiten. Normalerweise lassen sich dabei ein oder zwei Operanden durch einfache arithmetische Zeichen („+“ oder „-“) verändern.

2) Dieses Programm kehrt die Reihenfolge der Zeichen um, die in dem bei LABEL1 abgelegten String enthalten sind. Die Anordnung

adresse des Wortes, während X als Index für die Stack-Schleife dient. Danach wird Y als Index der Schleife eingesetzt, die das Herunterziehen vom Stack ausführt, während X die Endadresse des Wortes enthält. Der Begriff „Adresse“ ist hier etwas ungenau, da X und Y Ein-Byte-Register sind und keine vollständige Adresse enthalten können. Statt dessen speichern sie einen Offset zu der Adresse LABEL1.

In der Z80-Version werden statt IX und IY die Registerpaare HL und DE eingesetzt. Wie die X- und Y-Register des 6502 speichern sie die Anfangs- und Endadressen des Wortes, werden aber statt als Indizes über eine Basisadresse als indirekte Adressen eingesetzt

	6502	Z80
;		ORG \$C000
ORIGIN	ORG \$7000	LAST1 EQU \$0D
LAST1	EQU \$0D	SPACE EQU \$20
SPACE	EQU \$20	LABL1 DB 'THIS IS'
LABL1	DB 'THIS IS'	TERMN8 DB LAST1
TERMN8	DB LAST1	;
;		BEGIN LD DE, LABEL1-1
BEGIN	LDX #\$FF	LOOP0 CALL RVSWRD
LOOP0	JSR RVSWRD	CP LAST1
	CMP #LAST1	ENDLP0 JR NZ, LOOP0
ENDLP0	BNE LOOP0	RET
	RTS	;
;		***REVERSE A WORD S/R***
*****REVERSE A WORD S/R*****		LASTCH DB \$00
LASTCH	DB \$00	RVSWRD PUSH DE
LASTX	DB \$00	POP HL
RVSWRD	TXA	INC HL
	TAY	RVSLP0 INC DE
	INY	LD A, (DE)
RVSLP0	INX	PUSH AF
	LDA LABEL1, X	CP SPACE
	PHA	JR Z, CLRSTK
	CMP #SPACE	CP LAST1
	BEQ CLRSTK	ENDRV0 JR NZ, RVSLP0
	CMP #LAST1	CLRSTK POP AF
ENDRV0	BNE RVSLP0	LD (LASTCH), A
CLRSTK	PLA	;
	STA LASTCH	RVSLP1 POP AF
	STX LASTX	LD (HL), A
RVSLP1	PLA	INC HL
	STA LABEL1, Y	LD A, L
	INY	CP E
	CPY LASTX	JR NZ, RVSLP1
ENDLP1	BNE RVSLP1	LD A, H
	LDA LASTCH	CP D
	RTS	ENDRV1 JR NZ, RVSLP1
		LD A, (LASTCH)
		RET

der Wörter bleibt dabei erhalten. Interessant ist hierbei, wie die Befehle JSR und CALL eingesetzt werden. Die Unterroutine RVSWRD hat eine ähnliche Struktur wie das Programm der Übung 1, kehrt jedoch nur die Zeichen eines Wortes und nicht den gesamten String um. In beiden Versionen (6502 und Z80) wird das Index-Register (X und IX) eingesetzt, um die Anfangsadresse des Wortes an die Unterroutine weiterzugeben. Dabei übergibt der Akkumulator den Wert des Zeichens an das Hauptprogramm, das den Ablauf beendet (entweder ein Leerzeichen oder die Endkennung).

Ein weiteres wichtiges Merkmal ist der Gebrauch des Y-Registers in der 6502-Version. Das Register enthält zunächst die Anfangs-

(der Befehl LD A, (DE) bedeutet „Lade den Akkumulator aus dem Byte, dessen Adresse in DE gespeichert ist“). Alle Registerpaare des Z80 lassen sich auf diese Art verwenden. Da dem Befehlssatz jedoch im Zwei-Byte-Format jegliche Vergleichsmöglichkeit fehlt, müssen bei einem Vergleich der Inhalte von DE und HL die Register E mit L und D mit H einzeln verglichen werden. Da der 6502 keinen Befehl für den unmittelbaren Vergleich von X und Y hat, können hier X und Y nur mit Hilfe einer Speicherstelle indirekt miteinander verglichen werden.



Perfekte Zukunft?

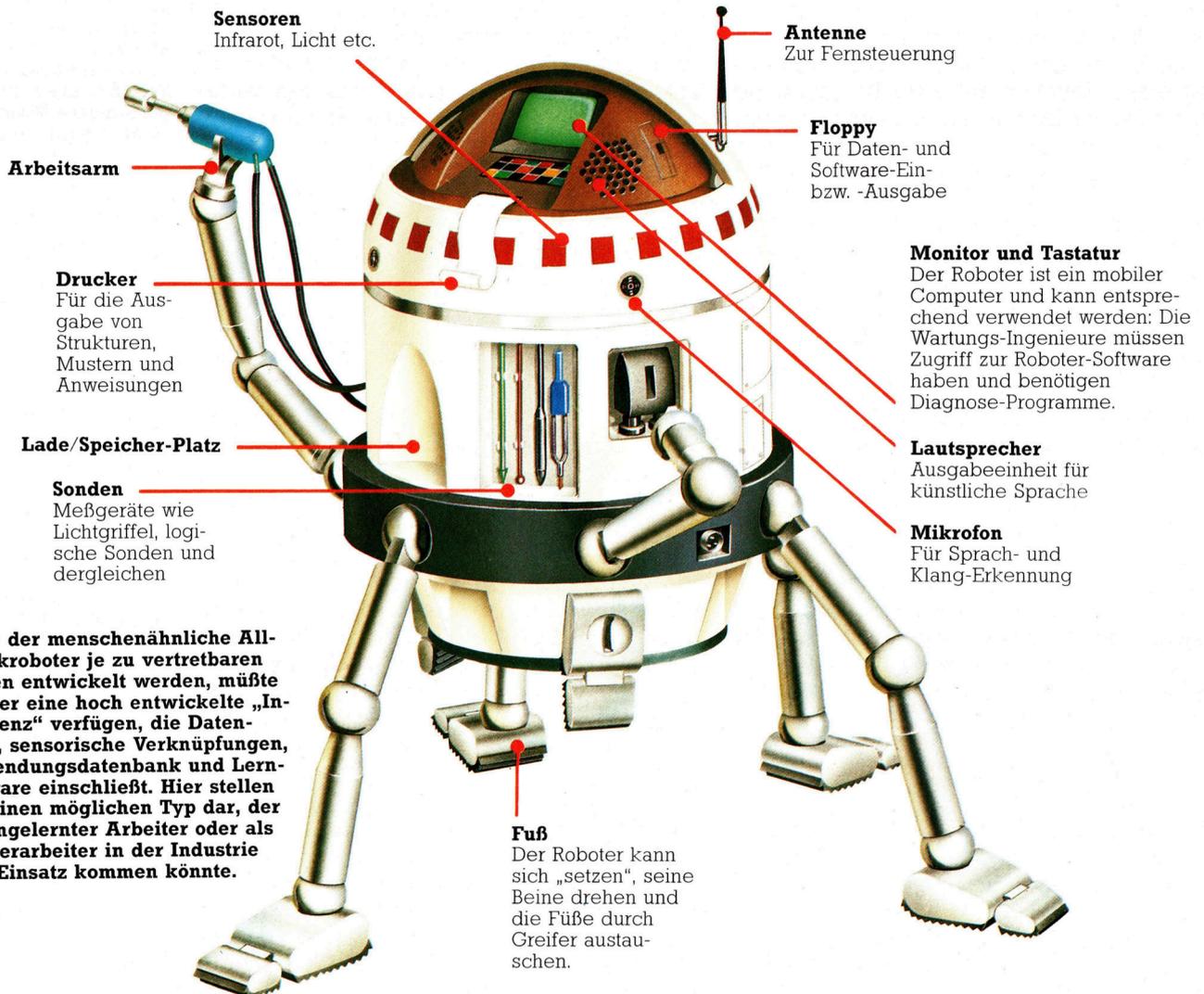
In dieser abschließenden Zusammenfassung gehen wir auf die praktisch bedingten Beschränkungen der heutigen Roboterkonstruktion ein und zeigen mögliche künftige Entwicklungen auf.

Wie weit die reale Roboterwelt vom fiktiven Konzept denkender mechanischer Wesen entfernt ist, haben wir aufgezeigt. Trotzdem erwarten wir bestimmte Dinge von Robotern. So gehen wir davon aus, daß sie sich völlig frei aus eigener Kraft bewegen, daß sie hören, sehen und fühlen können. Wir unterstellen, daß sie mit uns über Philosophie und Wissenschaft reden, zumindest aber auf intelligente Art kommunizieren. Und daß sie Objekte wie Gedanken so wie wir handelnd bzw. denkend bearbeiten. Mit anderen Worten: Wir

stellen uns Roboter wie menschliche Wesen vor. Betrachten wir aber das Angebot von Industrie- und Experimentierrobotern für den Hobbyisten kritisch und das, was sie können, so sind wir enttäuscht, weil sie nur geringen Ansprüchen genügen.

Da wir über Wesen und Grundlagen der Roboterkonstruktion im Bilde sind, – was kann man nun tatsächlich erwarten? Was werden künftige Roboter können? Fassen wir noch einmal die wesentlichen Konstruktionsanforderungen zusammen.

Metallkragen-Arbeiter



Sollte der menschenähnliche Allzweckroboter je zu vertretbaren Kosten entwickelt werden, müßte er über eine hoch entwickelte „Intelligenz“ verfügen, die Datenbank, sensorische Verknüpfungen, Anwendungsdatenbank und Lernsoftware einschließt. Hier stellen wir einen möglichen Typ dar, der als angelernter Arbeiter oder als Schwerarbeiter in der Industrie zum Einsatz kommen könnte.



Unwahrscheinlich ist, daß Roboter sich auf Beinen bewegen werden, wie es der Mensch tut. Um das Gleichgewicht zu halten, wäre sehr viel Datenverarbeitungszeit und – rein mechanisch gesehen – viel Platz erforderlich. Den Gelenken und der hydraulischen oder elektronischen Muskulatur der Roboter mangelt es an der Flexibilität und Freiheit, die dem Menschen aufgrund des Zusammenspiels zwischen Muskeln, Sehnen und Knorpeln gegeben ist. In manchen Fällen wäre die Fortbewegung eines Roboters auf zwei Beinen sogar eher hinderlich. Dennoch gibt es einige experimentelle Roboter, die – ähnlich wie Insekten – mit vier oder sechs Beinen ausgestattet wurden.

Für militärische Zwecke, Weltraumforschung und Anwendungsbereiche zu Hause scheint die Bewegung auf Rädern am praktischsten zu sein. Und dies wird sich kaum ändern. Die Bewegung des Roboters wird so flüssiger, erreicht aber nie die Eleganz und Anmut menschlicher Bewegung.

Viele Industrieroboter und kleinere Roboterarme sind aus naheliegenden Gründen fest installiert. Ihre Bewegungen beschränken sich auf einen relativ kleinen Aktionsbereich, da sie nur ein oder zwei genau festgelegte Aufgaben auszuführen haben. Sollte sich der Einsatzbereich von Industrierobotern nicht grundlegend ändern, wird sich auch an der Bewegungsart nichts ändern: Sie folgen Bahnen oder Gleisen oder kreisen um feste Achsen. Natürlich mag die Weiterentwicklung der Automation und Robotik bei industriellen Fertigungsmethoden radikale Veränderungen bringen, doch ist es unmöglich vorherzusagen, in welchem Umfang und wie das stattfinden wird.

Roboter können mit hochempfindlichen Sensoren ausgestattet werden, die ihre Wahrnehmungsfähigkeiten in Bereichen, die für den Menschen neu oder nicht vertraut sind, erheblich erweitern. Näherungssensoren, Bewegungsdetektoren, exakt arbeitende Rückmeldungseinrichtungen, die Standortangaben ma-

chen, und auf unterschiedlichsten Frequenzen arbeitende Geräuschdetektoren geben dem Roboter die Möglichkeit, mehr und umfassendere Daten zu sammeln, als der Mensch es kann.

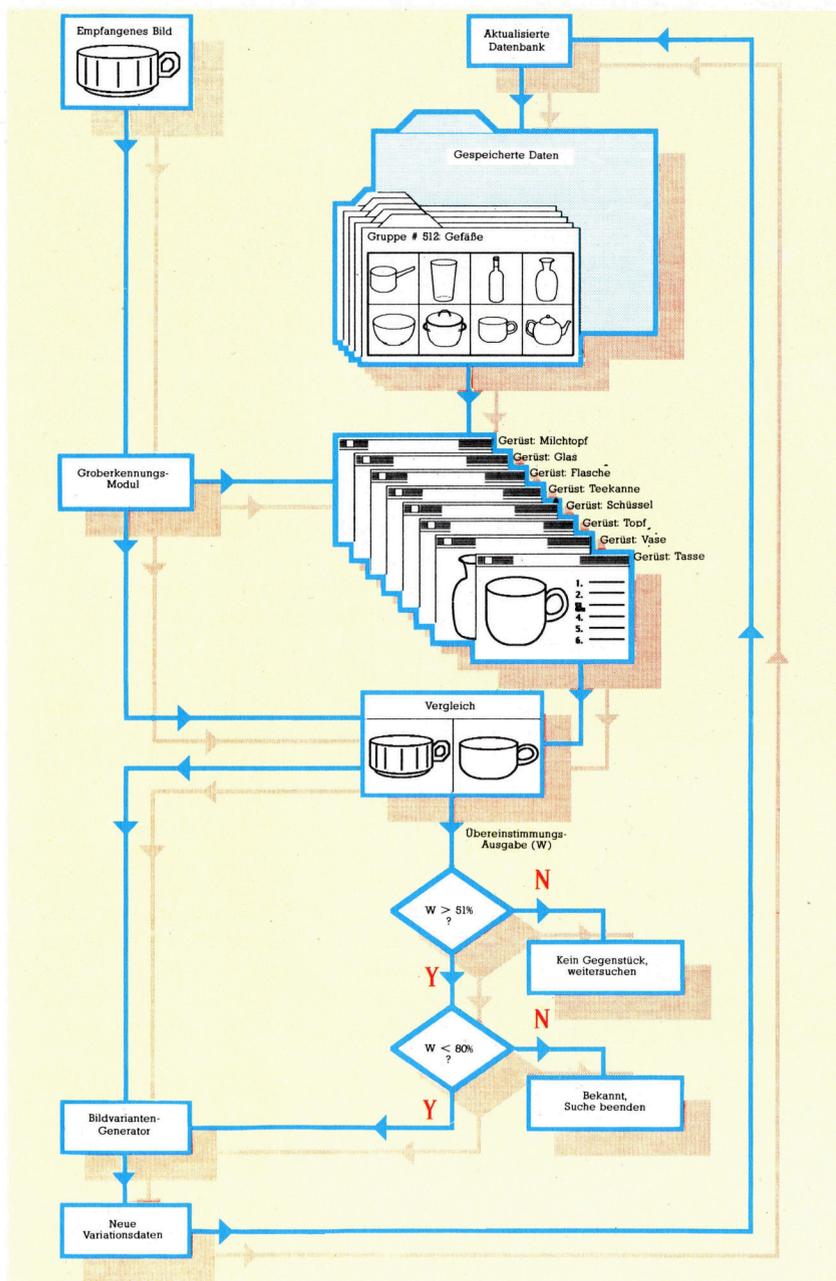
Arbeitsroboter, wie sie vornehmlich in der Industrie Anwendung finden, werden auch künftig nur mit solchen Sensoren ausgestattet sein, die für die Durchführung ihrer spezifischen Aufgaben erforderlich sind. Ein robotischer Schweißarm beispielsweise muß nicht über visuelle Rückmeldungssensoren verfügen. Mit minimaler sensorischer Eingabe kann er seine Aufgabe exakt ausführen.

Kombination aller Rückmeldungen

Allzweck-Roboter, die durch Erfahrung lernen und menschliche Denkprozesse nachvollziehen sollen, müssen dagegen mit so vielen Sensoren wie möglich ausgestattet sein. Andernfalls wäre es für einen Roboter problematisch, seine Umgebung selbständig zu erforschen und die gesammelten Informationen richtig zu verarbeiten. Beim Menschen geschieht dies durch die Kombination visueller und akustischer Rückmeldungen.

Gegenwärtig ist die Datenmenge, die ein Roboter empfangen und verarbeiten kann, eingegeben beispielsweise über Sensoren, durch den Speicherplatz begrenzt. Dazu entstehen weitere Einschränkungen aus der Prozessorstärke und der Datenverarbeitungsgeschwindigkeit. So kann ein Roboter das optische Bild eines Gegenstandes, etwa eines Apfels, speichern und dieses Bild mit einem Namen verbinden. Die Speicherung des Bildes belegt Speicherkapazität, und je besser die Auflösung des entsprechenden Sensors, desto mehr Speicherplatz ist für das Bild erforderlich. Da aber nicht alle Äpfel gleich sind, muß der Roboter entweder über ausreichend Speicherplatz verfügen, um ein charakteristisches „Sampling“ von Äpfeln aufnehmen zu können, oder mit einem Algorithmus programmiert

Die Robotertechnik zeigt sich in ihrer Gesamtbedeutung am nachhaltigsten, wenn sie an Hand von speziellen Einsatzmöglichkeiten demonstriert wird. – So beispielsweise in Form von Kränen, Bulldozern, Transporteinheiten und dergleichen mehr. Wir zeigen hier, wie ein robotischer Bulldozer einen Zug, der aus robotisch gesteuerten Wagen besteht, belädt. Nach der Beladung bewegen sich die halbintelligenten Fahrzeuge zu einem Sammelpunkt und verbinden sich zu einem Wagenzug. Dieser fährt dann auf öffentlichen Straßen, wird aber von einem Menschen, der im Führungsfahrzeug sitzt, gelenkt. Die Mischung menschlicher Gewandtheit und Entscheidungsfähigkeit mit der rein physischen Stärke und der einseitigen Intelligenz der Roboter verdeutlicht die wirtschaftlichste Anwendung der heute zur Verfügung stehenden und auch künftiger Techniken.



Das Wissen über einen Gegenstand kann als „Gerüst“-Bild des Grundgegenstandes gespeichert werden, ergänzt um variierende Daten. Ein von den Sensoren des Roboters erzeugtes Bild wird durch ein Grobanalyse-Modul abgetastet und vermittelt eine erste Klassifizierung des gesehenen Gegenstandes. Anschließend wird jedes Objekt der Klasse in den verschiedenen Variationen mit dem erfaßten Typ solange verglichen, bis ein der Vorlage entsprechendes gefunden ist.

sein, der ihm das Erkennen von Unterschieden ermöglicht und der die Grundform eines Apfels drehen kann, um sie aus jeder Position sehen und erkennen zu können. Selbst bei minimaler Auflösung von etwa 256 Punkten pro Bild reicht die Zahl der Varianten bereits in die Tausende.

Das Problem der Speicherkapazität wird durch Entwicklung leistungsfähiger RAM-Chips gelöst werden (1-MBit-Chips werden gegenwärtig entwickelt). Dazu kommt die Entwicklung spezieller RAM-Chips, die sogenannte „Varianten“-Daten enthalten. Diese Daten können bei Bedarf abgerufen werden, um zwischen verschiedenen Bildern zu unterscheiden.

Ergänzend zu größeren Speicherkapazitäten wäre für einen Roboter, der tatsächlich sensorisch erkennen soll, die Erfassung von Daten aus mehreren Quellen gleichzeitig erforder-

lich. – Und natürlich auch ihre Verarbeitung. Die derzeit zur Verfügung stehenden Prozessoren können diese Datenmengen nicht verarbeiten. Daher würden Datenstaus entstehen. Eine mögliche Lösung wäre der Einsatz von zwei oder mehreren parallel arbeitenden Hochgeschwindigkeitsprozessoren, die zudem über eine entsprechend große Kapazität verfügen. Ein spezieller Steuerprozessor würde dann die entsprechenden Aufgaben innerhalb des Systems verteilen.

Die Fortschritte zur Lösung der Hardwareprobleme gedeihen rasch. Doch ein Roboter benötigt sehr umfangreiche Software, um das Verarbeitete zu verstehen. Anders ausgedrückt: Der Roboter benötigt Verstand, damit er weiß, was er mit seinen Wahrnehmungen zu tun hat.

Einbinden in das Weltverständnis

Wie schon dargelegt, gibt es zwei Hauptrichtungen in der Robotik. Die eine, die wohl auch am fortgeschrittensten ist, ist der Bereich intelligenter Werkzeuge, also der Anwendungsroboter. In der Industrie eingesetzte Arme oder automatische Herstellungssysteme, wie ausgefeilt sie auch sein mögen, müssen lediglich mit entsprechend umfassender Steuersoftware ausgestattet sein. Ein Roboterarm kann mit Koordinaten programmiert werden, um eine Handlungsabfolge auszuführen, ohne daß er „verstehen“ muß, was er tut, und ohne Kenntnis seiner Umgebung. Werden Roboter jedoch für einen breiteren Aufgabenbereich eingesetzt, müssen die einzelnen Arbeiten noch präziser definiert werden. Bewegen sie sich beispielsweise in einem Raum, dessen Inhalt sich täglich ändert, ist nicht nur das Sammeln und Verarbeiten der damit verbundenen Informationen erforderlich, sondern auch die Einbindung ihrer Wahrnehmungen in ihr „Weltverständnis“. Der Roboter muß also sowohl steuernde wie „handelnde“ Software und genügend Speicherplatz haben.

Die Erschaffung künstlichen Verstandes wirft wichtige Fragen auf, die noch der Lösung bedürfen, genauso, wie es ungelöste Probleme bei der Betrachtung menschlicher Denk- und Lernweisen gibt. Was etwa weiß ein Kind tatsächlich bei seiner Geburt? Ist der Erwachsene ein Produkt seiner Umwelt oder seines Erbgutes? Welche Beziehungen bestehen zwischen beiden Faktoren? Ist ein Mensch mit Strukturen ausgestattet, die ihm beim Erlernen von Sprache, Mathematik und Ästhetik helfen? Und wenn ja, wie wirken sie zusammen? Igor Aleksander und Piers Burnet haben in ihrem Buch über Robotik, „Reinventing Man“ (etwa: „Die Wiedererfindung des Menschen“), 1983 im Verlag Kogan Pagel erschienen, aufgezeigt, wie schwer die Beantwortung dieser Fragen ohne direkte Experimente am menschlichen Gehirn ist.

Fachwörter von A bis Z

Daisy Wheel = Typenrad

Für Computerbesitzer, die einen Schönschreibdrucker brauchen, kommt fast nur das Typenradsystem in Frage. Der Name rührt von der Anordnung der Drucktypen her: Sie sitzen auf den Kunststoffspeichen eines Rades (Wheel) von circa 7 cm Durchmesser wie die Blütenblätter eines Gänseblümchens (Daisy). Das Rad dreht sich, bis der gewünschte Buchstabe oben steht, dann schlägt die Type auf Farbband und Papier, wodurch das entsprechende Schriftbild erzeugt wird.

Typenradprinter sind erheblich langsamer als die Matrixdrucker. Ein 1500-Mark-Typenradprinter schafft nur 10-15 Zeichen/s, ein Matrixdrucker dagegen glatt das Zehnfache. Auch die bis zu 8000 Mark teuren Typenrad-Spitzenmodelle erreichen nicht mehr als 120 Zeichen/s, weil das Typenrad vor jedem Anschlag durch den Antriebsmotor einmal beschleunigt und wieder abgebremst werden muß. Bei näherer Betrachtung des Typenrads erscheint die Reihenfolge der Buchstaben zunächst willkürlich – sie ist aber so gewählt, daß unter Berücksichtigung der üblichen Buchstabenkombinationen möglichst kleine Drehwinkel erforderlich sind.

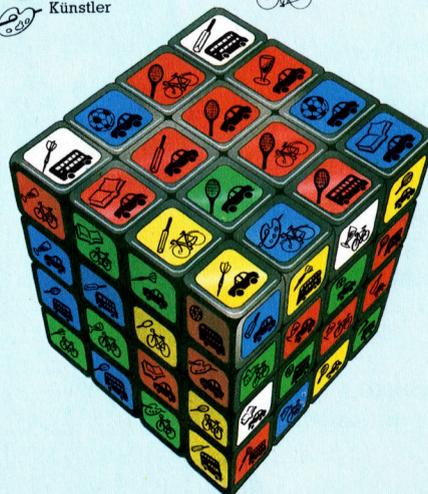
Database = Datenbank

Ein Datenbanksystem ist ein mehr oder weniger universelles Softwarepaket zum Abspeichern und Lesen von Informationen. Die Daten über eine Person (in einer Adreßliste) oder einen Artikel (bei der Lagerverwaltung) sind jeweils in einem Datensatz (record) enthalten, der einer Karteikarte bei konventionellen Karteien entspricht. Jeder Datensatz besteht aus einzelnen namentlich gekennzeichneten Feldern.

Bei einfachen Datenbanksystemen läßt sich nicht mehr als eine Datei auf einmal bearbeiten, während bei Mehrdatei-Systemen ein Datenaustausch zwischen verschiedenen Dateien möglich ist. Ein Abrechnungsprogramm kann auf diese Weise Daten aus einer Kundendatei, in der Kundenadressen und Höchstkredite

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.

vermerkt sind, mit Preisen, Artikelnummern und detaillierten Beschreibungen aus einer Produktdatei zusammenbringen. Ausgefeilte Datenbanksysteme sind programmierbar: Die Software „erlernt“ gängige Befehlssequenzen und führt sie dann auf einen einzigen Tastendruck hin aus. Solche Systeme werden auch als „Anwendungsgenerator“ bezeichnet.



Der eigentliche Sinn eines Datenbanksystems besteht darin, daß man Datensätze nach verschiedenen Gesichtspunkten sortieren kann. Zur Veranschaulichung ist hier der bekannte „Zauberwürfel“ von Rubik gezeigt, der die Datenbank darstellen soll; die einzelnen Facetten der Würfelseiten entsprechen den noch ungeordneten Datensätzen.

Data Compression = Datenverdichtung

Im allgemeinen sieht man als Speicherbedarf je Zeichen ein Byte (= 8 Bit) vor und zusätzlich etwas Platz für die Dateiorganisation in Form von Kopfsätzen und Steuerzeichen. Durch „Datenverdichtung“ oder „Komprimierung“ können Sie den Bedarf jedoch reduzieren. Sie gewinnen damit einerseits Speicherkapazität und verringern andererseits die Datenübertragungszeit.

Jede Verdichtung erscheint zunächst reichlich kompliziert, zumal alle Zeichen und viele Befehle AS-CII-verschlüsselt sind. Aber denken Sie an einen Text, der nur aus Kleinbuchstaben und maximal sechs Satzzeichen besteht – da reichen fünf Bits, um die erforderlichen 32 (= 2⁵) Symbole darzustellen. Statt für jedes Zeichen ein ganzes Byte zu opfern, lassen sich die Daten vor dem Abspeichern auf Diskette durch ein Spezialprogramm komprimieren und nach dem Lesen durch ein anderes Programm wieder auf die normale Länge bringen. Als Nebeneffekt bieten komprimierte Dateien einen gewissen Schutz gegen Mißbrauch, weil sie nicht ganz so leicht von einem fremden System gelesen werden können.

Ein stärkeres Verdichten als im Verhältnis 8:5 ist mit Hilfe von Häufigkeitsanalysen möglich. Bestimmte Buchstabenfolgen sind besonders häufig – ein Durchschnittstext besteht etwa zur Hälfte aus rund hundert Wörtern wie „und“, „der/die/das“ oder „ist“. Stellt man diese Wörter durch je ein Byte (mit einem Kennbit zur Unterscheidung von Einzelzeichen) dar, ergibt sich eine beträchtliche Platzeinsparung.

Bildnachweise

870: Associated Press
871: Associated Press, Ian McKinnell
873, 877, 881, 885, 889, 896:
Ian McKinnell
874, 879, 884: Liz Dixon
875, 882: Chris Stevens
878: Alison Fenton
883, 891, 896: Kevin Jones
888: Ian McKinnell, Liz Dixon
894: Steve Cross



Eine professionelle PASCAL-Ausrüstung kann recht teuer sein. Es gibt aber inzwischen auch viele Compiler, die erschwinglich sind. Wir beschreiben eine Auswahl der erhältlichen Programmpakete.

computerkurs

Heft **33**



Produkt-Image

Erfolg und Mißerfolg eines Produktes hängen weniger von der Herstellung als von der Vermarktung ab. Das gilt besonders für Computer und Software.



Peripherie-Übersicht

Der Heimcomputer ist oft nur der erste Schritt zum kompletten System; Zusatzgeräte erweitern die Hardware. Wir zeigen eine Auswahl.



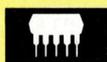
Integrierte Software

Der große Vorteil ist die Möglichkeit, zwischen unterschiedlichen Anwendungen wählen zu können.



Motorsteuerung

Im letzten Teil beschrieben wir den Bau eines Interface zur Steuerung – diesmal stellen wir die Software dazu vor.



Sinclair's Spectrum+

Das Gerät unterscheidet sich in der Leistung kaum von der alten Maschine, hat aber eine brauchbare Tastatur.

