

Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Heft 39

Plotsub-Programm

Digital/Analogwandler

Strategiespiel

Neu von Atari: der 130 XE

Die tragbaren Computer



**Ein wöchentliches
Sammelwerk**

computer kurs

Heft 39

Inhalt

Hardware

Comeback-Versuch

Ataris neuer 130 XE

1065

Software

Überblick durch Diagramme

Kalkulationsprogramm mit Grafik

1068

Elementare Kräfte

„Archon“ für Schach- und Action-Freunde

1082

BASIC 39

Magische Zahlen

Felder mit positiven Integer-Zahlen

1070

Dreidimensionale Kurven

Grafik mit mathematischen Formeln

1086

Tips für die Praxis

Digital/Analog-Wandler

Eine Erweiterung des User-Ports

1072

PASCAL

Der Record-Befehl

Zusammenfassung strukturierter Daten

1074

Computer Welt

Leichtgewichte

Die tragbaren Computer

1077

Llamasoft

Humorvolle Spiele

1088

Computer-Logik

Prüfroutinen

Vorrangschaltung und Prüfbit-Generator

1080

Peripherie

Kleine Scheibe

Schneiders Diskettensystem

1083

Bits und Bytes

Starke Steigung

Verwendung des Plotsub-Programms

1089

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 leserlich enthalten.

SAMMELORDNER

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

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

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

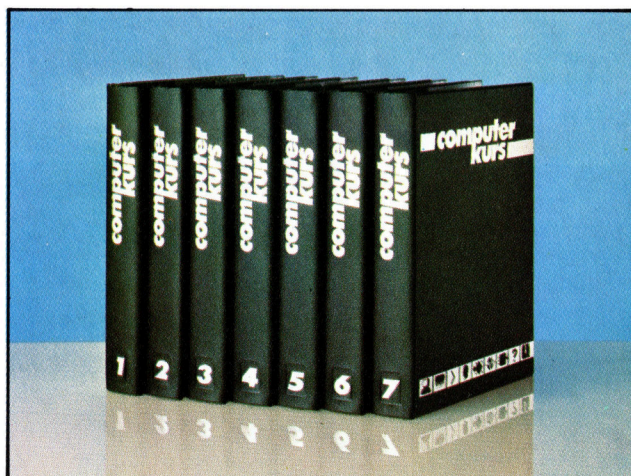
Schweiz: Der Sammelordner kostet sfr 15. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

INHALTSVERZEICHNIS

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

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

Vertrieb: Marshall Cavendish International Ltd., Heidenkampsweg 74, 2000 Hamburg 1





Der Atari 130 XE ist das erste Produkt des Unternehmens seit der Übernahme durch Jack Tramiel. Atari erhofft sich mit diesem Rechner, der 128 KByte RAM aufzuweisen hat und für einen günstigen Preis angeboten wird, einen großen Marktanteil zu sichern.

Comeback-Versuch

Der Atari 130 XE bietet die gleichen guten Grafik- und Soundfähigkeiten wie seine XL-Vorgänger. Die eigentliche Herausforderung aber sind das Design und die 128 KByte zu einem ungewöhnlichen Preis.

Obwohl Atari einer der Pioniere im Heimcomputer-Markt war, brachten die frühen achtziger Jahre dem Unternehmen wenig Gutes. 1984 – nach schweren Umsatzeinbußen – wurde das Unternehmen von Jack Tramiel, dem ehemaligen Commodore-Boss, übernommen. Er versuchte, das Ruder des angeschlagenen Computer-Giganten herumzureißen. Seine Marketingpolitik in Sachen Computertechnologie schlug sich im Handel darin nieder, daß die Preise für den 600 XL und 800 XL drastisch gesenkt wurden – gerade rechtzeitig zum Weihnachtsgeschäft.

Dieser Schachzug allein reichte aber nicht aus, um die Situation zu ändern. Atari sah sich vor dasselbe Problem gestellt wie viele andere Computerhersteller. Rückläufige Verkäufe ließen die Software-Unterstützung schrumpfen, womit der Abverkauf noch weiter gebremst wurde. Und das bedeutete: Die für die Entwicklung neuer Rechner erforderlichen Geldmittel wurden nicht erwirtschaftet.

Paradoxerweise war es Jack Tramiel, der Atari überhaupt erst in diese Situation gebracht hatte. Sein aggressives Marketing für den Commodore 64 führte beinahe dazu, daß Atari vom Markt gefegt wurde. Anfang 1985 än-

derte sich die Situation völlig. Commodore sah sich mit sinkenden Verkaufszahlen konfrontiert und landete mit dem Plus-4 einen Flop, wogegen Atari neue Produkte ankündigte. Erster dieser neuen Rechner ist der Atari 130 XE, ein Computer, der technisch auf dem 6502C-Prozessor aufgebaut ist.

Prinzipiell entspricht der 130 XE dem früheren Atari Acht-Biter, den es in dieser Form seit 1980 gibt. Der wesentliche Unterschied besteht im eleganten neuen Design und der erweiterten Speicherkapazität: Runde 128 K RAM kann der 130 XE vorweisen.

Modernes Styling

Das Gehäuse des Computers unterscheidet sich wesentlich von dem seiner Vorläufer. Der hellgraue Kunststoff wurde in die elegante Form gebracht, die der Verbraucher von einem modernen Computer erwartet, wozu Rundungen gehören und geformte, leichtgängige Tasten, die die Eingabe erleichtern.

Wie die anderen Rechner ist auch der Atari 130 XE mit fünf vorprogrammierten Funktionstasten ausgestattet. Anders aber ist die Platzierung derselben: Sie befinden sich jetzt über



dem eigentlichen Keyboard. Ihre Form ähnelt der von Parallelogrammen.

Die Schnittstellen auf der Rückseite und rechtsseitig des neuen Atari warten mit Überraschungen auf. Seitlich befindet sich die übliche neunpolige Joystick-Buchse, die Atari als erster verwendete und dann von fast allen übernommen wurde. Auf der Rückseite wurde der 13-polige serielle Controll-Anschluß installiert, über den Atari-eigene Peripheriegeräte wie Cassettenrecorder, Disketten-Stationen und Drucker angeschlossen werden. Rechts befinden sich die Modulsteckschächte sowie die Erweiterungsschnittstellen. In diese passen auch die zahlreichen, ausgezeichneten Cartridges wie beispielsweise „Pacman“ und „Galaxians“.

„Bank Switching“

Der Erweiterungsport indes unterscheidet sich vom bisher üblichen Standard. Vorläufermodelle verfügten über einen 50-poligen Platinenstecker, der als Erweiterungs-Bus diente. Der neue Bus ist als 14-Weg-Stecker angelegt, in Form eines Modul-Steckers. Ferner verfügt der 130 XE über Monitor-Anschluß, RF-Buchse für Fernseher und den standardisierten Atari-Netzstecker.

Design und Kompatibilität, wenngleich gut gemacht, bescherten Atari bei der XL-Serie wenig Erfolg. Was aber den eigentlichen Unterschied zu den Vorgängermodellen des 130 XE ausmacht und als Verkaufsargument gilt, ist die hohe Speicherkapazität in Relation zum Preis. Natürlich kann ein Acht-Bit Prozessor nur 64 KByte RAM gleichzeitig adressieren. Um diesen Wert zu verdoppeln, bedient man sich des sogenannten „Bank Switching“. Dank dieser Technik sieht der Computer ein „Fenster“ von 64 KByte in der Gesamtmenge von 128 KByte. Dieses Verfahren ist nicht perfekt, reduziert die Verarbeitungsgeschwindigkeit, und die Programmierung ist aufwendiger. Auf der anderen Seite stehen dem Benutzer dadurch zusätzliche 64 KByte zur Verfügung.

Genaugenommen ist das „Bank Switching“-Prinzip verbreiteter, als man vermutet. Sowohl der Oric Atmos als auch der C 64 verfügen über mehr Speicherkapazität als nur 64 KByte. Bei beiden Rechnern findet das genannte Prinzip zur Nutzung der Speicherkapazität ebenfalls Anwendung.

Ungewöhnlich ist der Preis für den 130 XE. Um einen 128 KByte-Computer zu einem relativ geringen Preis anbieten zu können, mußte Atari die Produktionskosten drastisch senken. Da der 130 XE eine überarbeitete Version der älteren Atari-Modelle ist, fielen kaum noch Entwicklungskosten an. Die eigentlichen Einsparungen fanden rechnerintern statt.

Der Speicherbereich ist mit 16 acht-KByte RAM Chips gefüllt. Die Herstellungskosten dieser Chips, die längst nicht mehr als „tech-

Erweiterungs-Port

Dieser Anschluß war bei den älteren Modellen nicht vorhanden.

RF-Buchse

Hier wird das Fernsehgerät angeschlossen.

RAM-Chips

Die 128 KByte RAM befinden sich in diesen beiden Gruppen von 8-KByte-Chips.

Speichersteuerungs-Chip

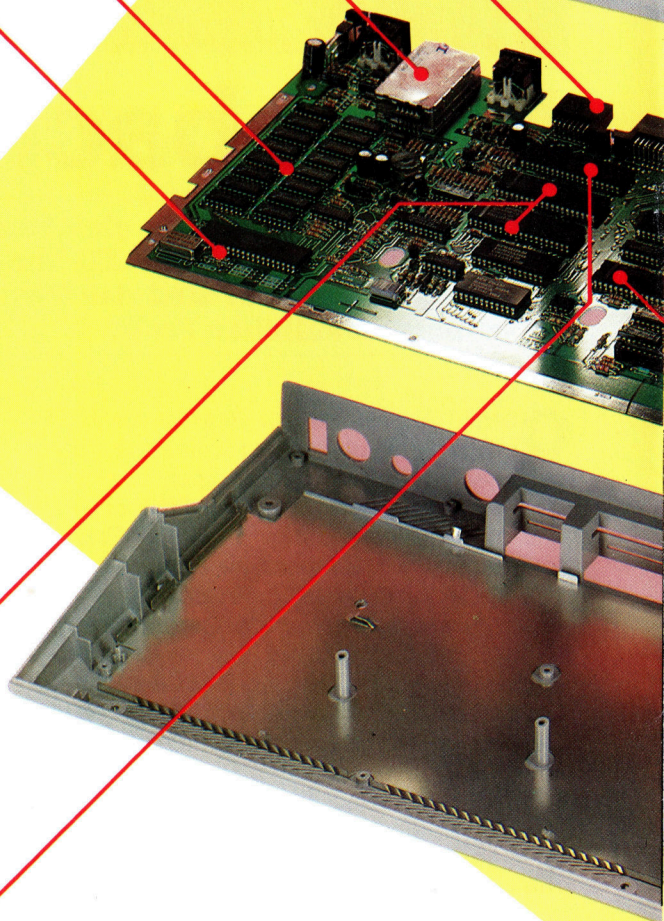
Die Speichersteuerung und die „Bank Switching“-Routinen laufen über diesen neuen Chip, der „Freddy“ getauft wurde.

Grafik-Chips

ANTIC- und GTIA-Chip steuern die Bildschirmgrafik.

CPU

Auch beim 130 XE dient der 6502 als Zentraleinheit.





Steckmodul-Schacht
Damit ist der „Neue“ mit der Modul-Software aus dem großen vorhandenen Angebot kompatibel.

Peripherie-Anschluß
Atari-eigene Peripherien wie Diskettenstation und Drucker können über diese serielle Schnittstelle angeschlossen werden.

Joystick-Anschlüsse
Der Rechner ist mit zwei Joystick-Ports ausgestattet.

PIA-Chip
Die Input/Output-Kontrolle erfolgt über einen 6520-Chip.

Sound-Chip
Der „POKEY“ bietet Soundmöglichkeiten über vier Oktaven.

nologisch führend“ betrachtet werden, sind in den vergangenen Jahren beachtlich gesunken. Das schlägt sich im Preis nieder. Weiter wurden Kosten gespart, indem man die Baugruppen auf der Platine mengenmäßig reduzierte. Um die Kompatibilität zu gewährleisten, wurden beim 130 XE viele Chips der XL-Serie verwendet. Die Platine des Rechners ist gut gestaltet und übersichtlicher als viele Rechner mit der Hälfte der Speicherkapazität. Atari hat zudem große Summen in automatisierte Fabriken investiert, deren erstes Produkt der 130 XE ist. Sämtliche Baugruppen werden maschinell auf die Platine gelötet.

Da weder Sound-, Grafik- noch BASIC-ROM-Chip grundsätzlich verändert wurden, entspricht der Rechner in den Augen des Benutzers dem, was Ataris guten Namen begründet hat: hervorragender Sound und beste Grafik. Ferner kommt dem Benutzer die Umgestaltung des Handbuchs zugute. Die Dokumentationen zu früheren Modellen waren unzureichend und enthielten nur wenige Informationen. Die neue BASIC-Einführung hat man erheblich verbessert, und im Anhang werden technische Details dargelegt. Um diese Version jedoch ganz verstehen zu können, ist immer noch das „Atari BASIC Reference Manual“ erforderlich.

Rätselhafte Einführung

Die Atari-Computer-Angebotspalette bedurfte zwar der Verbesserung, doch irgendwie ist die Einführung des 130 XE rätselhaft. Die zusätzlichen 64 K-RAM-Speicherkapazität bieten dem Programmierer zwar mehr Platz, doch bis heute gibt es keine Programme, die diesen wirklich nutzen. Üblicherweise wäre die Einführung eines solchen Rechners der Einstieg in den „Small Business“-Markt. Atari indes wehrt solche Absichten prinzipiell ab. Der eigentliche Grund für die Herstellung des Atari 130 XE mag letztlich darin liegen, daß man ein Konkurrenzprodukt für den Commodore 128 schaffen wollte – einen C 64-kompatiblen Rechner, der ebenfalls mit einer höheren Speicherkapazität ausgestattet ist. Das Gerät ist aber weitaus teurer als der neue Atari.

Atari 130 XE

ABMESSUNGEN

350 x 233 x 63 mm

ZENTRALEINHEIT

6502C, 1,79 MHz Taktgeschwindigkeit

SPEICHERKAPAZITÄT

128 KByte RAM, 24 KByte ROM

BILDSCHIRMDARSTELLUNG

40 x 24 Textdarstellung, 320 x 192 Punkte (hochauflösend) mit 256 Farben

SCHNITTSTELLEN

Modulschacht, TV-Buchse, Composite Monitor-Stecker, zwei Joystick-Buchsen, serielle Input/Output-Schnittstelle, Erweiterungs-Schnittstelle

PROGRAMMIERSPRACHEN

Atari-BASIC, LOGO, FORTH, PILOT

TASTATUR

62 Tasten, einschließlich fünf vorprogrammierter Funktionstasten

HANDBÜCHER

Im Handbuch wird Atari-BASIC erläutert. Die Darstellung erfolgt aber noch immer sehr vereinfacht. Zusätzliche Erläuterungen, beispielsweise wie die zusätzlichen 64 K-RAM zu „verwalten“ sind, werden im Anhang gegeben.

STÄRKEN

Der 130 XE verfügt über dieselben Vorteile wie seine Vorgänger, ist aber mit zusätzlicher Speicherkapazität von 64 K-RAM ausgestattet.

SCHWÄCHEN

Beim 130 XE handelt es sich um einen überarbeiteten Computer, der seit Jahren auf dem Markt ist und dem Verbraucher letztlich nichts wesentlich Neues bietet.



Der neue Atari bietet Kompatibilität mit älteren Peripheriegeräten der Firma, etwa dem hier gezeigten „Touchtablet“. Das gleiche gilt ebenso für Diskettenstationen, Drucker und natürlich Software-Steckmodule.



Überblick durch Diagramme

Trotz vieler Ähnlichkeiten im Aufbau haben alle Kalkulationssysteme ihren eigenen Charakter und ihre speziellen Eigenheiten. Dies trifft auch auf Graph Plan zu – ein Kalkulationspaket mit Grafiksystem –, das auf den Acorn-B-Computern läuft, die mit einem Z80 als Zweitprozessor ausgerüstet sind.

Im Gegensatz zu den in dieser Serie vorgestellten Programmpaketen arbeitet Graph Plan disketten-orientiert. Es ist Teil des Softwarepaketes, das Acorn beim Kauf eines Z80-Zweitprozessors kostenlos mitliefert. Wie die anderen Programme dieses Paketes ist auch Graph Plan außerordentlich hilfreich und zuverlässig.

Wenn das System auch nicht der Qualität des Abacus von Psion entspricht, so gleichen die schnellen Lade- und Speicherzeiten des Diskettensystems diesen Nachteil spielend wieder aus. (Wenn Sie an Cassettensoftware gewöhnt sind, mag die Geschwindigkeit der QL Microdrives durchaus ausreichen, haben Sie jedoch einmal mit Disketten gearbeitet, dann können Microdrives entnervend langsam erscheinen.)

Umfangreiche Befehlsliste

In diesem Artikel werden wir uns hauptsächlich mit der grafischen Darstellung von Kalkulationswerten beschäftigen. Graph Plan bietet – wie schon der Name sagt – außer vielen eingebauten mathematischen und kaufmännischen Formeln auch eine breite Palette von grafischen Darstellungsmöglichkeiten.

Die Bedienung von Graph Plan ist sehr eigenwillig, da alle Befehle (insgesamt 144) über Nummern aufgerufen werden. Nach dem Laden des Programms erscheint auf dem Bildschirm die gewohnte Kalkulationstabelle mit Reihen und Spalten, wobei rechts die zwanzig Grundbefehle mit ihren Nummern dargestellt sind. Alle Befehle werden am oberen Bildschirmrand auf der dritten Statuszeile hinter „ENTER COMMAND“ eingegeben.

Obwohl die Nummern der Befehle leicht aus dem Menü oder der Befehlsliste des ausgezeichneten Handbuches herausgesucht werden können, ist die Bedienung im Vergleich zu den meisten anderen Kalkulationssystemen – speziell zu Rennern wie Lotus 1-2-3 – sehr umständlich. Auf Systemen dieser Art braucht nämlich nur der Anfangsbuchstabe des betreffenden Befehls eingegeben oder der Cursor mit den Pfeiltasten auf den gewünschten Be-

fehl geführt zu werden.

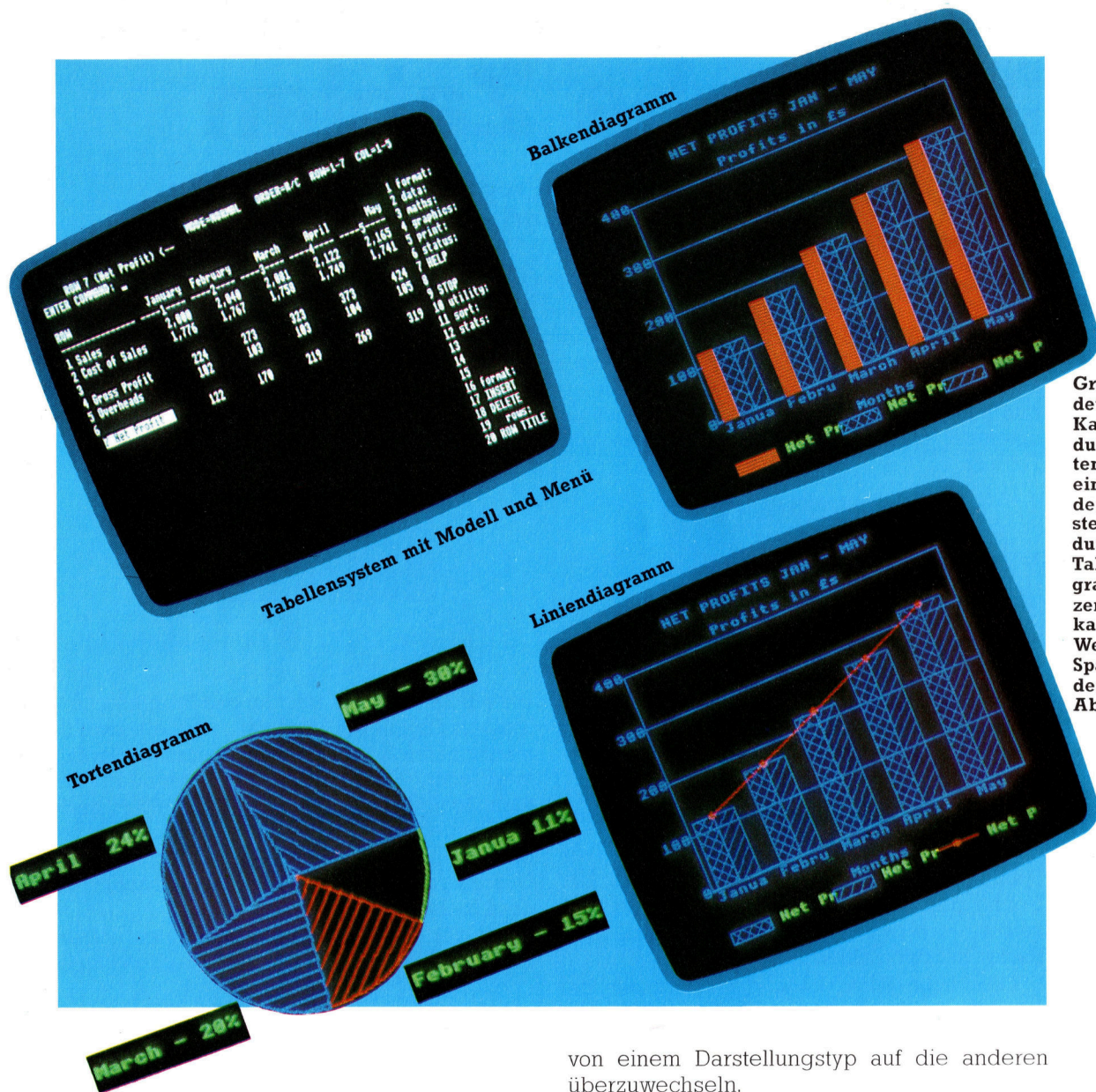
Hochentwickelte Pakete wie Lotus 1-2-3 stellen außerdem automatisch Informationen über die Funktionen der einzelnen Befehle dar – ein Komfort, den Graph Plan nicht bietet. Das Programm verändert jedoch die rechts unten angezeigte Befehlsliste je nach aufgerufenem Kommando. Wenn Sie beispielsweise Befehl 2 – „data“ – wählen, dann werden an dieser Stelle die Befehle 29 bis 48 (Eingabe- und Bearbeitung von Daten) angezeigt. Mit der HELP-Funktion (Befehl 7) lassen sich außerdem Erklärungen einzelner Befehle abrufen.

Außer diesem eigenwilligen Befehlssystem hat Graph Plan noch andere Besonderheiten. Während die meisten Kalkulationssysteme auf dem Konzept des Feldes (das heißt dem Kreuzungspunkt von Zeile und Spalte) aufbauen, behandelt Graph Plan Zeilen und Spalten als getrennte Einheiten. Dabei zeigt der „Data-Pointer“ in der zweiten Statuszeile außer der aktuellen Feldnummer an, ob der Zeilen- oder Spaltenmodus eingeschaltet ist.

Diese Unterscheidung hätte in einem System mit einzelnen Feldadressen als Bezugspunkt keine Bedeutung. Graph Plan benötigt diese Angabe jedoch für seine Diagrammdarstellung, da der Bezugspunkt hier nur eine Zeile oder eine Spalte, nicht aber beides gleichzeitig sein kann. Der Darstellungsmodus läßt sich festlegen, indem der Cursor mit den Steuerpfeilen auf den Data-Pointer „Zeile“ oder „Spalte“ gesetzt wird.

Wir werden zur Anschauung ein einfaches Modell aufbauen, das fünf Spalten „Januar“ bis „Mai“ und fünf Zeilen mit den Titeln „Verkauf“, „Einkauf“, „Bruttogewinn“, „Kosten“ und „Nettogewinn“ enthält. Anhand dieser Angaben läßt sich bereits eine einfache Datentabelle aufbauen. Das Diagramm dieses Modells hat dabei im Zeilenmodus eine andere Bedeutung als das im Spaltenmodus.

Auf Zeilenbasis sieht die Kurve der Verkaufszahlen von Januar bis Mai folgendermaßen aus: Die Spaltentitel („Januar“ – „Mai“) werden an der X-Achse angezeigt, während die Zeilenwerte von Balken dargestellt werden. Wenn Sie den Data-Pointer jedoch auf



Graph Plan unterscheidet sich von anderen Kalkulationssystemen durch seine numerierten Befehle, die in einem Menü rechts auf dem Bildschirm dargestellt werden, sowie durch seine Fähigkeit, Tabellendaten sofort in grafischer Form umsetzen zu können. Dabei kann Graph Plan die Werte einer Zeile oder Spalte in drei verschiedenen Formaten (siehe Abbildung) anzeigen.

Spaltenmodus setzen, entsteht ein völlig anderes Diagramm mit den Zeilenüberschriften („Verkauf“, „Einkauf“, etc.) an der X-Achse, während die Balken die Werte der Spalte 1 („Januar“) anzeigen.

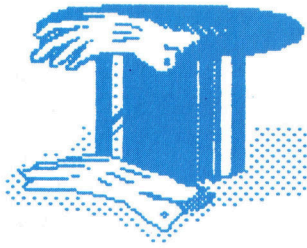
Verschiedene Diagrammtypen

Mit entsprechend strukturierten Daten kann Graph Plan sofort eine ganze Reihe unterschiedlicher Diagramme erzeugen und auf dem Bildschirm anzeigen. Nach Eingabe des Befehls „62 SELECT“ hinter dem „ENTER COMMAND“- Prompt erscheint in der dritten Statuszeile eine weitere Eingabeaufforderung. Im Zeilenmodus werden Sie nach der Nummer der Zeile gefragt, deren Werte grafisch dargestellt werden sollen. Danach können Sie den Diagrammtyp wählen (Balkendiagramm = 1, Liniendiagramm = 2 oder Tortendiagramm = 3) und mit dem Befehl „61 DISPLAY“ auf dem Bildschirm anzeigen. Es ist ebenfalls möglich,

von einem Darstellungstyp auf die anderen überzuwechseln.

Im Handbuch gibt es eine Entscheidungshilfe in Diagrammform, die die verschiedenen Gestaltungsmöglichkeiten aufführt. Die angezeigten Diagramme lassen sich mit dem Befehl „63 OPTIONS“ verändern. Dabei erscheint auf dem Bildschirm zunächst ein Menü mit sechs Optionen: Display Chart, Define Chart Options, Define Axes Options, Define Pie Options, Print Chart und Plot Chart. Im Anhang des Graph-Plan-Handbuches befindet sich ein Führer durch diese Untermenüs, der bei der Einarbeitung in das System sorgfältig gelesen werden sollte. Darin wird erklärt, wie sich Überschriften einfügen lassen, Farbe und Struktur der Balken festgelegt werden und welche Maßstäbe es gibt (auch logarithmische Darstellung ist möglich).

Mit seinen grafischen Fähigkeiten und den eingebauten mathematischen und statistischen Funktionen eignet sich Graph Plan gut für einfache wissenschaftliche und technische Anwendungen, aber auch für den kommerziellen Bereich.



Magische Zahlen

Wir setzen unsere Serie der BASIC-Programmierung mit einem mathematischen Gedächtnispuzzle fort.

Ein magisches Quadrat ist ein Raster von Feldern, in denen sich positive Integer-Zahlen (1,2,3,4,5, usw.) befinden. Das Ziel der Aufgabe ist, die Zahlen so anzuordnen, daß jede Reihe und jede Spalte bei der Addition das gleiche Resultat ergeben. Das einfachste magische Quadrat besteht aus drei-mal-drei-Feldern.

7	6	2	15
3	8	4	15
5	1	9	15
15	15	15	

In diesem Beispiel ergeben alle Reihen und Spalten bei der Addition das Ergebnis 15. Zur Übung probieren Sie das Puzzle doch einmal selbst – verwenden Sie ein fünf-mal-fünf-Raster mit den Zahlen 1 bis 25.

Glücklicherweise kann der Computer bei der Lösung des Problems helfen. Man kann zum Beispiel ein Programm schreiben, das die Felder mit Zahlen belegt und dann jeweils die Reihen und Spalten auf ihre Summe hin überprüft. Diese Methode ist jedoch sehr langwierig. Die folgenden Punkte zeigen einen besseren Weg.

Muster mit Methode

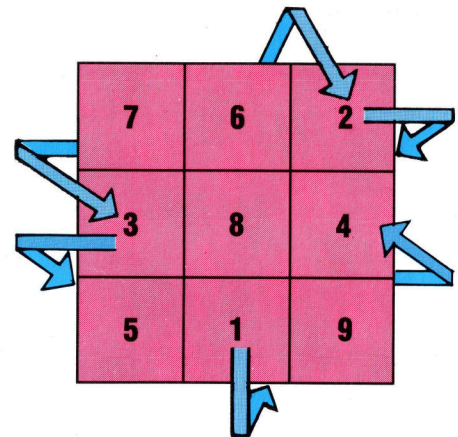
1) Beginnen Sie mit der Zahl 1 im mittleren Feld der untersten Reihe.

2) Danach gehen Sie mit der Zelle eine Reihe nach unten und ein Feld nach rechts. Sollte sich rechts kein Feld mehr befinden, verwenden Sie statt dessen das entsprechende Feld in der ersten Spalte. Ähnlich verfahren Sie, wenn sich unten kein Feld mehr befindet. Verwenden Sie dann das entsprechende Feld in der ersten Reihe.

3) Ist das nächste Feld bereits belegt, fahren Sie mit dem Feld links neben dem zuletzt belegten Feld fort.

4) Setzen Sie dieses Verfahren fort, bis alle Felder belegt sind.

Die folgende Abbildung zeigt die beschriebene Methode:



Dieses Zahlenspiel soll nun das Programm erledigen. Definieren Sie dafür zuerst ein zweidimensionales Array in der Größe des geplanten magischen Quadrates, und verwenden Sie eine Schleife zur Belegung der Felder nach den angegebenen Regeln. Beachten Sie jedoch, daß diese Methode nur bei magischen Quadraten mit einer ungeraden Anzahl Feldern funktioniert.

Bei der Darstellung des Quadrates sollten Sie darauf achten, daß alle Zahlen ordentlich in

Magische Quadrate

Diese Quadrate wurden mit dem gezeigten Programm generiert und überprüft. In dem 15 × 15-Quadrat kann ein diagonales Muster drei- und zweistelliger Zahlen untergebracht werden.

52	42	32	22	12	2	73	72	62
63	53	43	33	23	13	3	74	64
65	55	54	44	34	24	14	4	75
76	66	56	46	45	35	25	15	5
6	77	67	57	47	37	36	26	16
17	7	78	68	58	48	38	28	27
19	18	8	79	69	59	49	39	29
30	20	10	9	80	70	60	50	40
41	31	21	11	1	81	71	61	51

Reihen und Spalten angeordnet werden. Dies erreicht man am schnellsten mit der PRINT USING-Anweisung. Verfügt Ihr Computer nicht über diesen Befehl, sollten alle zu druckenden Zahlen in einen String umgewandelt werden. In diesem String können dann entsprechende Leerzeichen eingefügt werden, so daß alle Zahlen gleichlang sind. Eine entsprechende Unteroutine sieht wie folgt aus:

```
1000 REM Wandle A in A$ um
1010 A$ = STR$(A)
1020 IF LEN(A$) < 3 THEN A$ = " " + A$
      GOTO 1020
1030 RETURN
```

Die exakte Umwandlungsmethode ist vom verwendeten Computer abhängig (siehe BASIC-Dialekte).

Darstellung der Quadrate

Das nächste Problem ist die Bildschirmgröße. Die meisten Computer sind nicht in der Lage, größere magische Quadrate auf einmal auf dem Bildschirm darzustellen. Eine 40-Zeichen-Darstellung bietet Platz für 13 zweistellige Zahlenreihen. Ein 13-mal-13-Quadrat dagegen enthält bereits einige dreistellige Zahlen, so daß ein 9-mal-9-Quadrat das größte ist, das auf einmal dargestellt werden kann. Mit einem Drucker können erheblich größere Quadrate erzeugt werden. Die meisten Drucker haben eine maximale Druckbreite von 80 oder 132 Spalten. Größere Quadrate können in mehreren Abschnitten gedruckt werden, die dann nachträglich zusammengefügt werden.

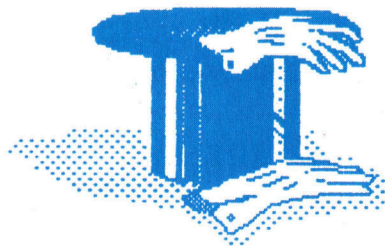
```
10 REM*****
15 REM***MAGIC SQUARES*****
20 REM***SET-UP*****
30 M=19:DIM A(M,M)
40 PRINT:PRINT"Magic Squares"
50 PRINT:PRINT"How many rows (1 to 19)";
  :INPUT S
60 IF S<0 OR S>INT(S) THEN PRINT"ERROR":GOTO 50
70 IF S>M THEN PRINT"ERROR":GOTO 50
80 IF S/2=INT(S/2) THEN PRINT"ERROR - Odd
  Numbers Only":GOTO 50
90 REM***GENERATE SQUARE*****
100 X=INT(S/2)+1:Y=S:C=1
110 A(X,Y)=C
120 C=C+1:IF C>S*2 THEN GOTO 200
130 X=X+1:IF X>S THEN X=1
140 Y=Y+1:IF Y>S THEN Y=1
150 IF A(X,Y)<>0 THEN X=X-2:Y=Y-1
160 IF Y=0 THEN Y=S
170 IF X=0 THEN X=S
180 IF X=-1 THEN X=S-1
190 GOTO 110
200 REM***PRINT SQUARE*****
210 PRINT:PRINT
220 FOR Y=1 TO S:FOR X=1 TO S
230 A=A(X,Y):GOSUB 380:PRINT " ";A$;" ";
240 NEXT X:PRINT:NEXT Y

250 REM***CHECK ROWS & COLS****
260 F=0
270 FOR Y=1 TO S:T=0
280 FOR X=1 TO S:T=T+A(X,Y):NEXT X
290 IF F=0 THEN U=T:F=1
300 IF T(>)U THEN PRINT"ERROR - Row 1 &
  Row";Y;" Do Not Match":STOP
310 U=T:NEXT Y
320 FOR X=1 TO S:T=0
330 FOR Y=1 TO S:T=T+A(X,Y):NEXT Y
340 IF T(>)U THEN PRINT"ERROR - Row 1 &
  Col";X;" Do Not Match":STOP
350 U=T:NEXT X
360 PRINT:PRINT"All rows and cols add to ";T
370 STOP
380 REM*****NUM-STRING CONV*****
390 A$=STR$(A)
400 IF LEN(A$)<3 THEN A$=" " + A$:GOTO 400
410 RETURN
```



BASIC-Dialekte

Dieses Programm ist in Microsoft-BASIC geschrieben und sollte auf den meisten Computern un geändert funktionieren. Spectrum-Besitzer müssen bei allen Zuordnungsanweisungen LET einfügen. Das Programm fragt nach der Anzahl der Reihen (und Spalten) und prüft, ob es eine positive gerade Anzahl von Feldern enthält. Danach berechnet es die Zahlen und stellt das magische Quadrat dar. Anschließend, ab Zeile 250, überprüft es seine eigene Ausgabe.



130	114	98	82	66	50	34	18	2	211	210	194	178	162	146
147	131	115	99	83	67	51	35	19	3	212	196	180	164	148
164	148	132	116	100	84	68	52	36	20	4	213	197	181	165
166	165	149	133	117	101	85	69	53	37	21	5	214	198	182
183	167	151	135	119	103	87	71	55	39	23	6	215	199	183
200	184	168	152	136	120	104	88	72	56	40	7	216	200	184
217	201	185	169	153	137	121	105	89	73	57	8	217	201	185
9	218	202	186	170	154	138	122	106	90	74	9	218	202	186
26	10	219	203	187	171	155	139	123	107	91	26	10	219	203
43	27	11	220	204	188	172	156	140	124	108	43	27	11	220
60	44	28	12	221	205	189	173	157	141	125	60	44	28	12
62	46	45	29	13	222	206	190	174	158	142	62	46	45	29
79	63	47	31	30	14	223	207	191	175	159	79	63	47	31
96	80	64	48	32	15	224	208	192	176	160	96	80	64	48
113	97	81	65	49	33	17	1	225	209	193	113	97	81	65



D/A-Wandler

In diesem Abschnitt soll das User-Port-System um einen Digital/Analog-Wandler erweitert werden. Der Wandler erlaubt die Ansteuerung analoger Geräte, macht aber auch digital erzeugte „Sphärenklänge“ hörbar.

Ein Digital/Analog-Wandler kann zwar aus einzelnen Bauelementen hergestellt werden, wir greifen zur Vereinfachung jedoch auf eine fertige integrierte Schaltung zurück.

Der Analogausgang des DAC-Chips (DAC=Digital/Analogue-Converter) wird mit einem ebenfalls integrierten Verstärkerbaustein gebuffert.

Erster Schritt: Das Zuschneiden des Gehäuses, das zu den anderen Bus-Erweiterungen passen sollte. Durch einen Minicon-Anschluß kann auch dieses Zusatzmodul später nochmals erweitert werden.

Zweiter Schritt: Die Platine zuschneiden (16 Streifen à 30 Löcher) und die Leiterbahn-Unterbrechungen nach der Zeichnung vornehmen. IC-Sockel und Drahtbrücken, danach die beiden Kondensatoren einlöten. Daraufhin folgt (falls gewünscht) der Erweiterungsanschluß und das Flachkabel.

Dritter Schritt: Die vier Anschlußbuchsen und das Potentiometer montieren und mit verzinnter Litze verdrahten sowie die Zuleitungen der Platine festlöten.

Vierter Schritt: Zum Schluß werden die ICs eingesetzt. Beachten Sie, daß die Chips in verschiedener Richtung eingesteckt werden müssen – beim D/A-Wandler liegt die Markierung links, wenn der Minicon-Stecker nach oben zeigt, die Markierung des anderen ICs ist auf der rechten Seite.

Nach dem Zusammenbau und einer gründlichen Überprüfung aller Lötstellen können Sie den Digital/Analog-Wandler testen. Der Wandler erzeugt zu jedem Acht-Bit-Wert im Datenregister des User Ports eine entsprechende Spannung. Der Gleichstromausgang des Moduls stellt entsprechend den Datenregister-Werten von 0 bis 255 eine Spannung zwischen 0 und 2,5 Volt zur Verfügung. Der zweite Ausgang simuliert eine Wechselstromquelle und läßt sich mit dem Potentiometer auf die für das jeweils angeschlossene Gerät passende Spannung einstellen.

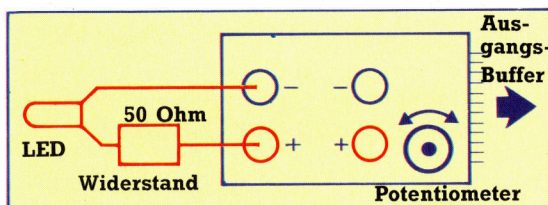
Mit einem einfachen Experiment können Sie die Helligkeit einer Leuchtdiode verändern.

Erster Schritt: Schließen Sie die LED (vom gleichen Typ wie die LEDs im Ausgangsbuffer) mit einem 50 Ohm-Widerstand zusammen.

Zweiter Schritt: Der D/A-Wandler wird direkt mit dem Buffer verbunden.

Dritter Schritt: Die Leuchtdioden/Wider-

standsschaltung wird mit dem Gleichstromausgang des D/A-Wandlermoduls verbunden. Starten Sie das untenstehende Programm:



```
10 REM **** CBM64 D-TO-A TEST PROGRAM ****
20 DDR=$6579:DATREG=$6577
30 VL=127
40 POKE DDR,255:REM ALL OUTPUT
50 POKE DATREG,VL
60 PRINT VL
70 GET A$
80 IF A$<>"Z" AND A$<>"X" THEN 70
90 IF A$="X" THEN DV=1
100 IF A$="Z" THEN DV=-1
110 VL=VL+DV
120 IF VL<256 AND VL>=0 THEN 50
```

```
10 REM **** BBC D-TO-A TEST PROGRAM ****
20 DDR=$FE62:DATREG=$FE60
25 value=127
30 ?DDR=255:REM ALL OUTPUT
40 REPEAT
55 ?DATREG=value
57 PRINTvalue
60 A$=GET$
62 IF A$<>"Z" AND A$<>"X" THEN 60
65 IF A$="X" THEN dv=1 ELSE dv=-1
67 value=value+dv
70 UNTIL (value>255 OR value<0)
```

Das Programm setzt das Daten-Richtungs-Register auf 255 (also 11111111) und schaltet damit alle User Port-Anschlüsse auf Ausgabe. Der User Port wird danach auf einen Anfangswert von 127 gestellt. Mit den Tasten Z oder X läßt sich dieser Wert nun erhöhen oder vermindern. Bei Werten über 255 bzw. kleiner als Null wird das Programm beendet.

Aus der Erhöhung eines Digitalwertes im Datenregister erzeugt unser Wandler eine steigende Versorgungsspannung für die LED. Sobald eine ausreichende Spannung erreicht ist, leuchtet die LED erst schwach, dann immer heller auf. Die größte Helligkeit erreicht die Diode beim Datenregister-Wert 255.

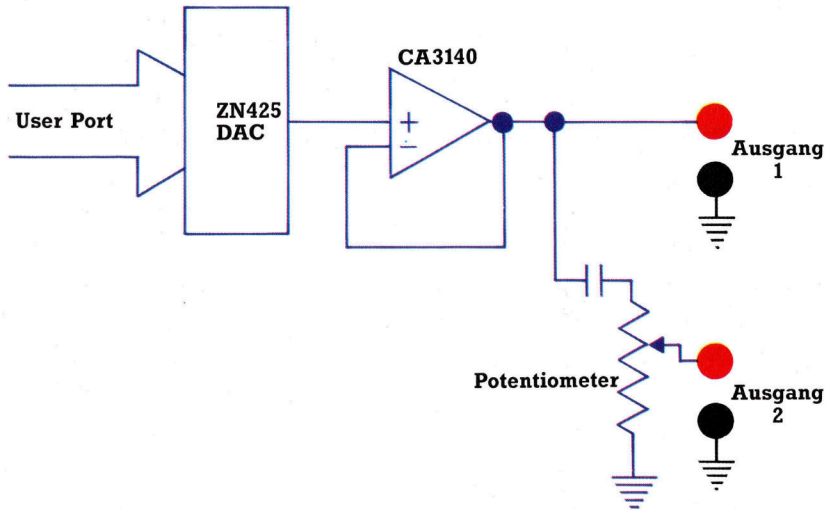
Sollte die LED nicht ansprechen, vertauschen Sie ihre Anschlüsse am D/A-Wandlermodul. Im Gegensatz zu normalen Lampen arbeiten LEDs nämlich nur, wenn der Gleichstrom in einer bestimmten Richtung fließt – bei verpoltem Anschluß bleiben sie dunkel.

Bei dem hier abgebildeten Digital/Analog-Wandler-Modul ist die Achse des Potentiometers viel zu lang – sie sollte so weit verkürzt werden, daß sich ein Drehknopf daran montieren läßt. Wahrscheinlich kennen Sie die Schwierigkeit, genau passende Bauteile zu erwerben – meist muß man die Teile vor der Montage erst noch verändern, damit alles richtig sitzt.





Schaltplan



Wie bei jedem Schaltungsaufbau müssen Sie auch diesmal Ihre Platine sehr sorgfältig bearbeiten – speziell bei den Leiterbahn-Unterbrechungen darf Ihnen kein Fehler unterlaufen. Am besten ständig mit dem Multimeter nachprüfen! Passive Bauteile und Drahtbrücken

werden zuerst eingelötet, wobei man das Lötzinn möglichst sparsam verwenden sollte. Vorsicht auch bei den ICs – die Markierung muß auf der richtigen Seite sitzen!

Bauteil-Liste

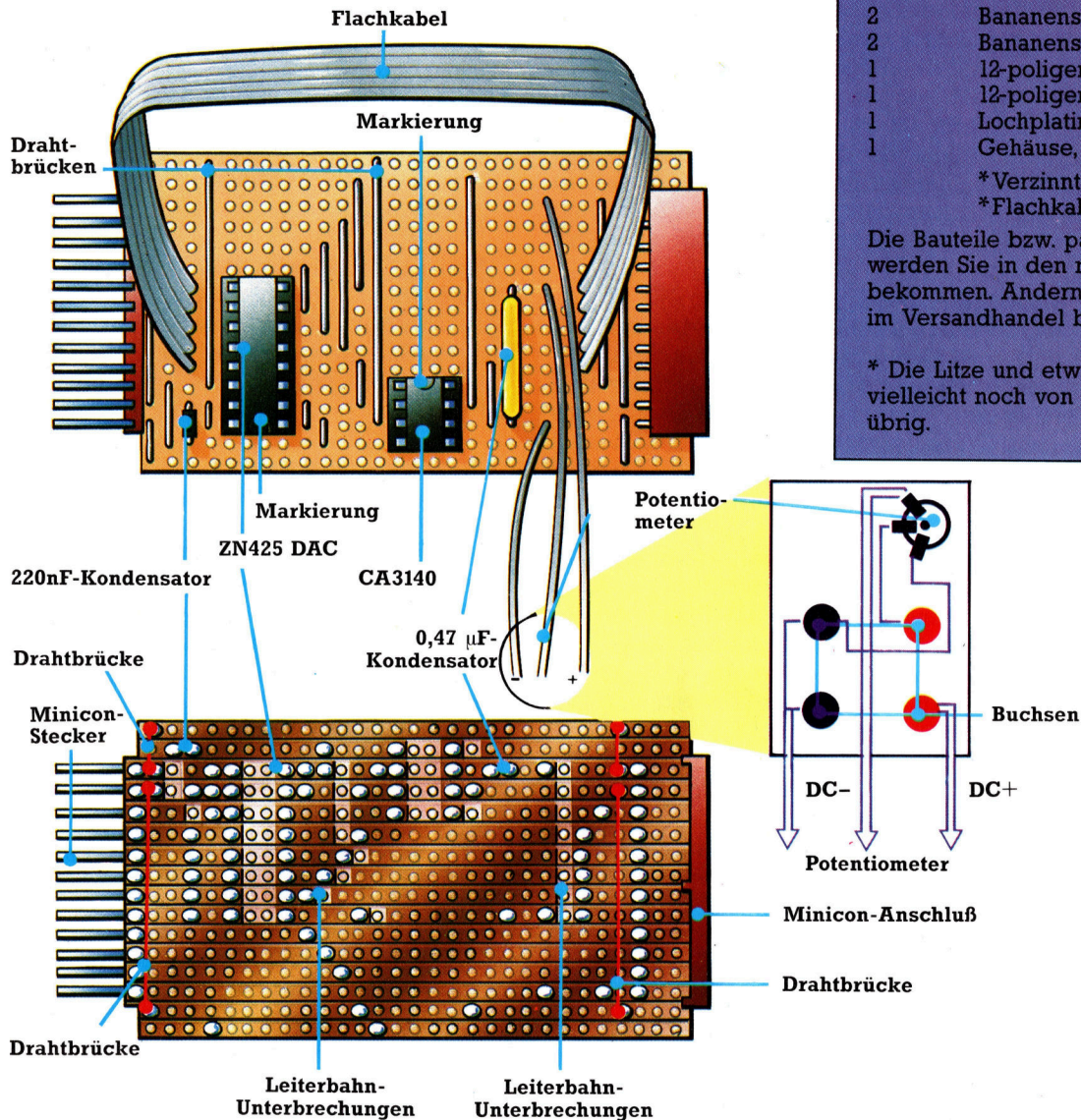
Anzahl	Bauteil
1	ZN 425 DAC Digital/Analog-Wandler
1	CA 3140 Operationsverstärker
1	16-poliger IC-Sockel
1	8-poliger IC-Sockel
1	Kondensator, 220 nF
1	Kondensator, 0,47 μ F
1	Drehpotentiometer, 10 k Ohm
2	Telefonbuchsen, rot, 4 mm
2	Telefonbuchsen, schwarz, 4 mm
2	Bananenstecker, rot, 4 mm
2	Bananenstecker, schwarz, 4 mm
1	12-poliger Minicon-Anschluß
1	12-poliger Minicon-Stecker
1	Lochplatine, 24 Streifen à 50 Löcher
1	Gehäuse, 80 x 61 x 41 mm

*Verzinnte Litze

*Flachkabel, fünfadrig

Die Bauteile bzw. passende Vergleichtypen werden Sie in den meisten Elektronik-Läden bekommen. Andernfalls kann man sie auch im Versandhandel bestellen.

* Die Litze und etwas Flachkabel haben Sie vielleicht noch von früheren Bastelarbeiten übrig.



Der Record-Befehl

In diesem Artikel gehen wir nochmals auf Sets ein und untersuchen dann den Datentyp „Record“. Mit Record lassen sich unterschiedlich strukturierte Daten bequem zusammenfassen.

In der letzten Folge wurde gezeigt, wie sich Programmabläufe durch den Einsatz von Sets vereinfachen lassen. Element eines Sets kann jeder echte Skalartyp (kein Real) sein. Außer dem Operator IN, der testet, ob ein Objekt Element eines Sets ist, werden für Sets die normalen Operatoren eingesetzt. Zwar gibt es keinen Mechanismus, der ein bestimmtes Element aus einem Set herausnimmt, doch läßt sich mit den Operatoren \leq und \geq feststellen, ob ein Set Teilmenge eines anderen Sets ist, während $=$ und $\langle \rangle$ die Sets auf Mengengleichheit testet.

Für mathematische Anwendungen dürfen Mengen nicht nur unbegrenzte Objekte enthalten, sondern können theoretisch als unendlich definiert werden. (Das ist der Vorteil, wenn Probleme nur auf dem Papier gelöst und nicht in funktionierende und physisch begrenzte Hardware umgesetzt werden müssen.) In einem Computer sind Sets daher in Bezug auf Größe und Geltungsbereich Grenzen gesetzt. Typen wie:

RiesenSet = SET OF Ganzzahlen;
sollten deshalb nicht definiert werden. Ein Set-Element an der Untergrenze eines Sets muß immer einen Ordinalwert von Null oder darüber haben, während die absolute Obergrenze je nach Computer zwischen 255 und 4095 liegt. Beachten Sie, daß die Leermenge (durch [] dargestellt) Element aller nur möglichen Sets ist, unabhängig von deren Typ. Obwohl dies nicht der exakten Typendefinition von PASCAL zu entsprechen scheint, wird in der Praxis der Typ der Leermenge immer von den Typen der anderen Sets eines Ausdrucks bestimmt.

Sets als Testabfragen

Es ist nicht möglich, Untersets in Sets des gleichen Typs direkt einzuschließen. Außerdem sind die Operatoren $<$ und $>$ für Sets nicht definiert. Der Grund dafür liegt in der praktischen Anwendung – die meisten Einschränkungen von PASCAL ergeben Sinn, wenn sie vom Standpunkt des praktischen Einsatzes oder der reinen Logik her betrachtet werden. Die Beantwortung der Frage, ob ein Set echte Teilmenge eines anderen Sets ist, muß daher mit einem doppelten Test erfolgen:

$(A \geq B) \text{ AND } (A \langle B)$

Bei dem Test von „Haus“ im Bingo-Programm der letzten Folge war dieser spezielle Test nicht nötig, da der Set der aufgerufenen Zahlen

nur größer oder gleich dem Set der in der Karte enthaltenen Nummern sein konnte, wobei der erste Fall weitaus wahrscheinlicher ist. Der folgende boolesche Ausdruck

Haus := Aufruf \geq Karte
ergibt „true“, wenn alle Elemente von „Karte“ in dem Set „Aufruf“ enthalten sind. Durch diese Eigenschaften und Funktionen eignen sich Sets ideal als Datenstrukturen für individuelle Problemlösungen. Eins der praktischsten Einsatzgebiete von Sets sind Tests von Untersets des Typs „char“. Mit dem folgenden Programm lassen sich Spiele steuern:

```
Nein := ['N','n'];
Ja := ['J','j'];
```

REPEAT

(* Spielablauf und Anzeige *)

(* der Gewinnpunkte *)

write ('Noch ein Spiel ? ');

ReadLn (Antwort);

WHILE NOT (Antwort IN Ja + Nein) DO
BEGIN

WriteLn ('J(a) oder N(ein) : Spalte ');

write ('Noch ein Spiel ? ');

ReadLn (Antwort);

End

UNTIL Antwort IN Nein

Wenn Sie auf die Elemente einer Datenstruktur einzeln zugreifen möchten, haben Sie die Wahl zwischen Arrays, Files und Records. Besonders die Records lassen sich für praktische Anwendungen sehr gut einsetzen, da sie unterschiedliche Datentypen speichern können.

In kommerziellen Programmen gibt es oft Datensätze, deren Felder Namen, Adressen, Telefonnummern, Rechnungsnummern etc. enthalten. Die eingesetzte Datenstruktur muß sich dabei als Einheit verarbeiten lassen, aber auch den Zugriff auf die einzelnen Felder erlauben. In PASCAL können Informationen dieser Art als Einheit zusammengefaßt und verarbeitet werden, wobei die einzelnen Felder individuell abrufbar sind und gemäß ihrem Datentyp bearbeitet werden können. Gemischte Records lassen sich leicht anlegen:

TYPE

Raum = RECORD

Nummer : 1..999;

Richtung : (Nord, Ost, Sued, West);

Besetzt : boolean

END; (* Raum *)

VAR

Buero : Raum;

Ebenso wie CASE durch den Einsatz des reservierten Wortes END eine Sonderstellung einnimmt, so ist auch die Definition eines Records im Deklarationsteil die Ausnahme zu der Regel, daß BEGIN und END nur paarweise eingesetzt werden dürfen. END sollte daher zur leichteren Identifizierung immer den Namen des Records als Kommentar enthalten. Jede Variable des Typs „Raum“ ist nun aus drei verschiedenen Feldern zusammengesetzt. In diesem Fall ist jedes Feld ein anderer Skalartyp. Es könnte aber auch der gleiche Typ – einfach oder strukturiert – sein. Es gibt keine Einschränkung, welche Typen in einem Record untergebracht werden dürfen. Ein Feld kann daher durchaus ein Array von Files enthalten, in denen Records untergebracht sind. Diese Records können wiederum aus Sets bestehen.

Die WITH-Anweisung

Zwischen den Ausdrücken RECORD und END gleicht die Syntax exakt der VAR-Deklaration. Dabei werden jedoch Feldnamen deklariert, die Teil der Datenstruktur sind. Die Wörter „Nummer“, „Richtung“ und „besetzt“ gibt es daher außerhalb des „Bereichs“ des Record-Namens nicht. Sie sind lokale Variablen und können daher in weiteren Teilen des Programms in anderen Zusammenhängen eingesetzt werden. Es gibt zwei Möglichkeiten, auf die Felder zuzugreifen: mit dem Punkt (.) und der Anweisung WITH.

Zur Wahl eines bestimmten Feldes wird der Name des gesamten Records mit einem Punkt von dem Feldnamen getrennt.

Buero. Nummer

bezieht sich nur auf die Ganzzahl des Feldes „Nummer“. Der Inhalt von Record kann folgendermaßen initialisiert werden:

```
read ( Buero.Nummer );
' Buero.Richtung := Ost;
' Buero.besetzt := Bewohner <> [ ]
```

Beachten Sie, daß im letzten Beispiel die Leermenge „Bewohner“ eingesetzt wurde, deren Typ nicht deklariert ist.

Wenn auf alle oder fast alle Felder eines Records zugegriffen werden soll, kann eine Zuweisung mit Punkten recht umständlich werden. Mit WITH ist dies weitaus einfacher. WITH bedeutet: „Ich möchte diesen Record bearbeiten und gebe daher nur die Feldnamen an“. Der Aufbau der WITH-Anweisung entspricht dem der WHILE-Schleife. Die oben aufgeführte Initialisierung eines Records sieht mit WITH weitaus klarer aus:

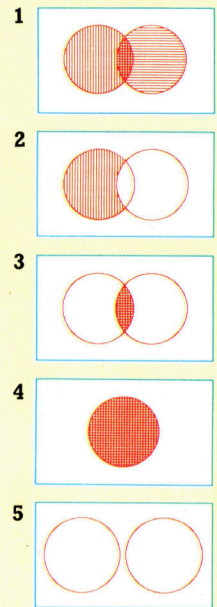
WITH Buero DO

```
BEGIN
    Nummer := 123;
    Richtung := Ost;
    Besetzt := true;
END
```

PASCAL-Operatoren für Sets

Diese Liste enthält alle Set-Operationen, die in PASCAL möglich sind (mit ihren Venn-Diagrammen).

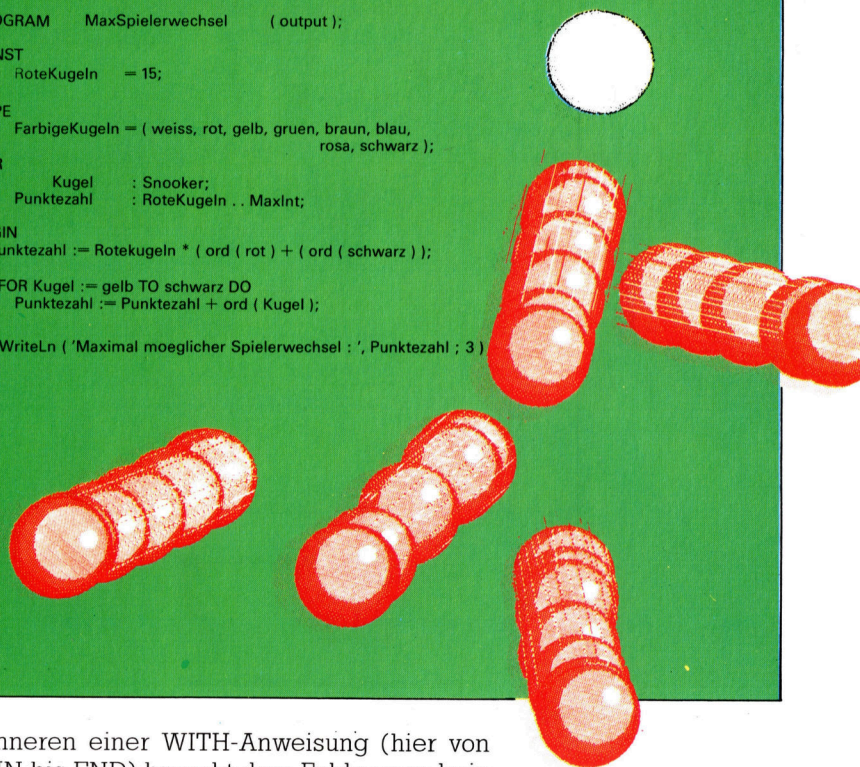
- 1) Vereinigungsmenge ($S1 + S2$): Die Vereinigungsmenge zweier Mengen enthält alle Elemente, die in $S1$ und in $S2$ enthalten sind. (Diagramm 1)
- 2) Mengendifferenz ($S1 - S2$): Die Mengendifferenz ist die Untermenge aller Elemente von $S1$, die nicht zu $S2$ gehören. (Diagramm 2)
- 3) Durchschnittsmenge ($S1 * S2$): Die Durchschnittsmenge enthält alle Elemente, die in beiden Sets gleichzeitig vorkommen. (Diagramm 3)
- 4) Mengengleichheit ($S1 = S2$): Mengengleichheit entsteht, wenn die Menge $S1$ mit der Menge $S2$ identisch ist. (Diagramm 4)
- 5) Mengenungleichheit ($S1 <> S2$): Ergibt „true“, wenn nicht jedes Element von $S1$ auch Element von $S2$ ist. (Diagramm 5)
- 6) Teilmenge ($S1 \leq S2$): Ergibt „true“, wenn jedes Element von $S1$ ebenfalls Element von $S2$ ist.
- 7) Teilmenge ($S1 \geq S2$): Ergibt „true“, wenn jedes Element von $S2$ ebenfalls Element von $S1$ ist.
- 8) Enthalten in ($E \text{ IN } S1$): Ergibt „true“, wenn der Set mit einem Element (E) ein Unter-set von $S1$ ist.



Billard

Hier ist die Auflösung für die Aufgabe der letzten Folge. Es sollte ein Programm geschrieben werden, das die maximal möglichen Spielerwechsel in einem Billardspiel berechnet.

```
PROGRAM    MaxSpielerwechsel    ( output );
CONST
    RoteKugeln    = 15;
TYPE
    FarbigeKugeln = ( weiss, rot, gelb, gruen, braun, blau,
                    rosa, schwarz );
VAR
    Kugel    : Snooker;
    Punktezahl : RoteKugeln .. MaxInt;
BEGIN
    Punktezahl := RoteKugeln * ( ord ( rot ) + ( ord ( schwarz ) );
    FOR Kugel := gelb TO schwarz DO
        Punktezahl := Punktezahl + ord ( Kugel );
    WriteLn ( 'Maximal moeglicher Spielerwechsel : ', Punktezahl ; 3 )
END.
```



Im Inneren einer WITH-Anweisung (hier von BEGIN bis END) braucht dem Feldnamen kein Record-Name und Punkt voranzustehen. Auch wenn eine andere Variable des Programms ebenfalls den Namen „Nummer“ hat, entsteht kein Konflikt, da lokale Variablen immer Vorrang haben. Die Feldbezeichnung mit Punkt ist nicht an diese Grenzen gebunden:

Buero.Nummer := nummer

ordnet beispielsweise die externe Variable (Nummer) dem Feld (nummer) des Datensatzes zu. Da PASCAL keine Unterscheidung zwischen Groß- und Kleinschreibung vornimmt, muß sich der nicht näher definierte Variablenname „nummer“ auf eine externe Variable beziehen (sofern er nicht in eine WITH-Anweisung eingeschlossen ist). Eine Zuordnung zwischen Variablen des gleichen Record-Typs muß jedoch mit der Punktanschreibung erfolgen:

Fassade.Richtung := Aussicht.Richtung
Über WITH könnte hier nur das Feld eines

Records angesprochen werden, da sonst eine Mehrdeutigkeit auftritt, die der Compiler nicht zuläßt. Möglich wäre aber auch:

WITH Fassade DO

Richtung := Aussicht.Richtung

Beide Schreibformen haben ihren speziellen Anwendungsbereich, der sich aus den Programmanforderungen ergibt.

In den nächsten Folgen werden wir Arrays und Files genauer untersuchen und an Beispielen zeigen, wie die Möglichkeiten dieser flexiblen Strukturen genutzt werden können.

Von Yards, Feet und Inches

Das Programm „Strecken“ verarbeitet zwei Längeneinheiten, die per Tastatur in Yards, Feet und Inches eingegeben werden. Über die Definition eines Record-Typs wird jedem Wert der drei unterschiedlichen Maßeinheiten ein eigenes Feld zugeordnet. Das Programm addiert dann die beiden Strecken, wobei mit DIV und MOD festgestellt wird, ob ein Übertrag vom Feld „Inches“ in Feld „Feet“ und von „Feet“ in „Yards“ nötig ist. Die Ergebnisse werden den Feldern von „Gesamt“ zugeordnet. Die Zuordnung ganzer Records kann direkt ausgeführt werden, die Bearbeitung muß jedoch für jedes Feld einzeln erfolgen. So ist „Gesamt := AStrecke + BStrecke“ nicht möglich. Beachten Sie den Gebrauch des Punktes und der WITH-Anweisung beim Lesen der Werte von AStrecke und BStrecke.

```
PROGRAM Strecken (input, output);
CONST
    MaxByte = 255;
TYPE
    byte = 0 .. MaxByte;
    Strecke = RECORD
        ins : 0 .. 11;
        ft  : 0 .. 2;
        yds : byte;
    END; (* Strecke *)
VAR
    Inches,
    Feet : byte;
    AStrecke,
    BStrecke,
    Gesamt : Strecke;
BEGIN
    WriteLn ( 'Strecke in Yards, Feet und Inches getrennt, ');
    WriteLn ( 'durch Leerzeichen oder RETURN eingeben.' );
    WriteLn;
    write ( 'Strecke A : ' : 15 );
    read ( AStrecke.yds, AStrecke.ft, AStrecke.ins );
    write ( 'Strecke B : ' : 15 );
    WITH BStrecke DO
        read (yds, ft, ins);
    Inches := AStrecke.ins + BStrecke.ins;
    Feet := AStrecke.ft + BStrecke.ft + Inches DIV 12;
    WriteLn;
    WITH Gesamt DO
        BEGIN
            ins := Inches MOD 12;
            ft  := Feet MOD 3;
            yds := AStrecke.yds + BStrecke.yds
                + Feet DIV 3;
            WriteLn ( 'Die Gesamtstrecke ist : ' );
            WriteLn ( yds : 25, 'yds.', ft : 1,
                'ft. und ', ins : 1, 'ins.' )
        END
    END.
```

Datenschutz

In unserem Bingo-Programm sollte mit Hilfe von Schleifen verhindert werden, daß bereits aufgerufene Nummern wiederholt werden. Außerdem sollte die Eingabe von illegalen Nummern ausgeschlossen werden. Die beste Methode, das Programm vor der Eingabe falscher Daten zu schützen, ist die Verwendung von WHILE-Schleifen, die auf die beiden ReadLn-Anweisungen folgen. In beiden Fällen wird die gleiche Struktur eingesetzt:

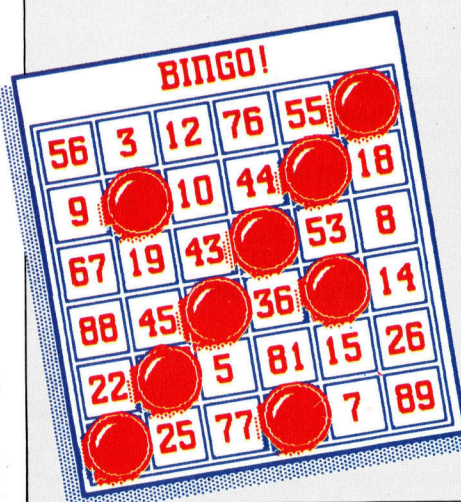
```
WHILE illegale Nummer DO
    eine Fehlermeldung ausgegeben
    einen neuen Prompt anzeigen
    die Daten nochmals einlesen (read)
```

Beim Festlegen der Kartennummern muß sichergestellt werden, daß die eingegebenen Zahlen nicht außerhalb des möglichen Bereiches (Eins bis 90) liegen und daß eine Wiederholung bereits eingegebener Nummern ausgeschlossen ist:

```
WHILE NOT ( Nummer IN moeglich ) OR
    ( Nummer IN Karte ) DO
BEGIN
    WriteLn ( Nummer : 20, ' ist nicht legal' );
    write ( 'Neueingabe : ' : 14 );
    ReadLn ( Nummer )
END;
(* etc. *)
```

Beim Aufruf der Nummern wird die gleiche Methode eingesetzt, die Eintrittsbedingung für die Prüfschleife ist jedoch geändert:

```
WHILE NOT ( Nummer IN moeglich ) OR
    ( Nummer IN Aufruf ) DO
    (* etc. *)
```





Leichtgewichte

Enthusiastisch wurde Adam Osbornes erster tragbarer Micro-computer im Jahre 1981 gefeiert. Gemessen an diesem Rechner und den Nachbauten seinerzeit ist die neueste Portable-Generation wirklich „tragbar“.

Seit es „Hand-held“-Computer gibt, mußte der Begriff „portable“ (also tragbar) neu definiert werden. Und tatsächlich werden die vor wenigen Jahren noch als „portable“ Micros eingeführten Rechner heute als „transportabel“ bezeichnet. Wirkliche Tragbarkeit bieten heute Computer, die mit eigener Stromversorgung, Bildschirm und integriertem Speicher ausgestattet und dabei nicht größer als ein Telefonbuch sind.

Der Epson HX-20 war der erste Computer dieser Art. Heute aber ist die Flüssigkristallanzeige mit ihren winzigen zwanzig Zeichen in vier Reihen Indiz für das Alter des Rechners. Neue Portables wie der Tandy 100, NEC PC-8210A und Olivetti M10 kosten etwa ebensoviel, stellen aber die vierfache Menge Zeichen dar.

Ständig griffbereit

Was können diese Computer überhaupt? Was sind ihre Vor- und Nachteile gegenüber konventionellen Bürorechnern? Hauptkaufargument dürfte wohl die ständige Verfügbarkeit von Computerpotenz sein. Manche Leute haben ihren PC häufig nicht in Greifweite und verbringen viel Zeit in Hotels, auf Flughäfen oder in Zügen. Mit dem Portable oder Hand-held kann diese Zeit genutzt werden.

Die jüngste Portable-Generation ermöglicht den Einsatz bei wissenschaftlichen oder ingenieurtechnischen Aufgaben ebenso wie bei der Buchführung, finanziellen Verwaltung oder Textverarbeitung.

Die meisten Hand-held-Computer verfügen über mindestens drei integrierte Programme. Das ist einmal der BASIC-Interpreter, eine Textverarbeitung und die Telekommunikations-Software. Darüber hinaus bieten der Tandy 100 und der Olivetti dem Anwender integrierte Adreß- und Termin-Programme, mit denen man Adressen, Telefonnummern und wichtige Termine festhalten und abrufen kann.

Das Kommunikationsprogramm ist besonders wichtig, da damit die Kommunikation über Telefon mit anderen Rechnern und Datenbanken erst möglich wird. Zugleich fungiert der Portable als Telex-Anschluß oder als Sen-



Computern auf Reisen wird immer populärer, vor allem bei Geschäftsleuten, die so auch die Zeit, die sie außerhalb ihres Büros verbringen, sinnvoll nutzen können. Vertreter nehmen Computer zu ihren Kunden mit, um Berechnungen durchzuführen, die sonst erst Tage später ausgeführt werden könnten.

Manager haben unterwegs Zugriff zu allen wichtigen Informationen durch Modem über die normale Telefonleitung und können am Ende eines Tages nach Rückkehr ins Büro die Daten direkt an einen Großrechner senden.

Computer im Flugzeug

Geschäftsleute, die ihre Computer im Flugzeug benutzen wollen, sollten sich vorher bei der betreffenden Airline erkundigen, ob die Inbetriebnahme an Bord überhaupt zulässig ist. Die Internationale Luftfahrtbehörde (CAA) hat festgestellt, daß manche batteriebetriebenen Geräte Störungen in der



Elektronik des Flugzeuges verursachen. Diese Empfehlung wird unterschiedlich behandelt. Lufthansa und Qantas verbieten generell den Betrieb elektronischer Geräte an Bord. Japan Airlines macht keine Einschränkungen. British Airways untersagt die Computerbenutzung an Bord.

der und Empfänger für elektronische Post. Voraussetzung dafür ist natürlich ein Modem oder ein Akustik-Koppler. Geschäftsleute können somit jederzeit Verbindung zu ihrem Büro aufnehmen. Ein Reporter beispielsweise kann am Ort des Geschehens seine Geschichte auf einem Portable schreiben und sie an den Redaktionsrechner übermitteln.

Integrierte Software

Teurere Portables wie der Sharp PC-5000 und der Epson PX-8 arbeiten mit MS-DOS und CP/M-Betriebssystemen, die denen der üblichen PCs entsprechen. Deshalb ist auf ihnen fast die gesamte Business-Software lauffähig.

Der Epson PX-8 ist sogar mit dem populären Textverarbeitungsprogramm „Wordstar“ ausgestattet. Beim Sharp verschafft ein Blasen-



speicher (als Modul) zusätzliche 128 KByte Speicherplatz pro Einheit. Diese Module verarbeiten Daten mit einer höheren Geschwindigkeit als Diskettenstationen.

Bei preiswerteren Portables werden die Programme von Cassetten geladen. Dieser Prozeß ist natürlich langsamer. Der NEC PC-8201A wird mit einer Cassette geliefert, auf der mehrere Programme enthalten sind: ein programmierbarer Rechner, ein Textformatierprogramm, ein Börsen- bzw. Aktienprogramm und ein Darlehnsberechnungsprogramm. Das Rechenprogramm speichert bis zu 99 Eingaben. Der Textformatierer besorgt die Druckvorbereitung für die über das Textverarbeitungsprogramm eingegebenen Dokumente, indem er die Zeichenbreite ermittelt, die Seitenaufteilung vornimmt, Seitenziffern hinzufügt usw. Das Aktienprogramm ist für Leute gedacht, die über Wertpapiere verfügen und ihre Gewinne und Verluste ermitteln wollen.

Wie andere Computer, so können auch die Portables mit Peripherien verbunden werden, so etwa Druckern, Cassettenrecordern und Modems. Neben der Größe zeichnet sich ein echter Portable durch Batteriebetrieb, integrierten Bildschirm, Textverarbeitung und Kommunikationsprogramme in ROM-Form aus.

Computer wie der Apple IIc oder der Apricot werden zwar als Portables beworben, können aber nicht überall verwendet werden, da sie an eine Stromversorgung angeschlossen, mit einem Monitor verbunden und die Programme von Diskette ins RAM geladen werden müssen. Sie sind zwar kleiner und leichter, haben ansonsten aber mehr mit den üblichen PCs als mit Portables gemein.

Neben der eigentlichen Batterie-Stromversorgung sind Portables mit einer Notstromversorgung auf der Basis von kleinen Nickel-Kadmium-Batterien ausgerüstet.

Dazu verfügen die meisten Portables über einen Strichcode-Leser, der unter anderem für die Kontrolle der Lagerhaltung wichtig ist. In diesem Code sind Preis- und Datumsinformationen enthalten, die über Computer verarbeitet werden können. Von den hier gezeigten Rechnern verfügen bis auf den Casio FP-200 alle Computer über die genannte Lesemöglichkeit. Der Tandy 100, der NEC PC-8201A und

Epson HX-20

Er hat zwar nur eine kleine LCD-Anzeige, ist aber serienmäßig mit Cassettenrecorder, Textverarbeitung und Drucker ausgestattet.

Casio FP-200

Er ist der preiswerteste unter den Hand-helds, dafür aber auch nicht mit integrierter Textverarbeitung, sondern nur mit einer Art Spreadsheet ausgestattet.



der Olivetti M10 sind sich auf vielerlei Art ähnlich, da sie alle in derselben japanischen Fabrik gefertigt werden. Trotzdem gibt es einige Unterschiede: Der Bildschirm des Olivetti kann im Winkel verändert werden, und der NEC verfügt über weniger integrierte Software. Die Speicher des Tandy 100 und des Olivetti lassen sich nicht über 32 KByte hinaus erweitern, wogegen beim NEC 64 KByte möglich sind. Darüber hinaus lassen sich beim NEC austauschbare 32-KByte-Module verwenden, die die Daten selbst dann behalten, wenn sie sich nicht im Computer befinden.

So wie Reiseschreibmaschinen kein Ersatz für Büroschreibmaschinen, sondern lediglich eine Ergänzung sein sollen, verhält es sich

auch bei den Portables. Die kleinen LCD-Schirme lassen ohnehin keine längere Arbeit zu, da die Augen aufgrund der Zeichenanzeige schnell ermüden. Dazu kommt, daß die LCD-Darstellung auf Eingaben langsamer reagiert als eine Kathodenstrahl-Röhre.

Im Gegensatz zu den Tastaturen der professionellen Geräte erweisen sich die der Portables bei längerer Verwendung als sehr anstrengend. Preisgünstige Portables können zudem die meiste Business-Software nicht verarbeiten. Trotzdem sind die Portables vor allem für häufig reisende Geschäftsleute sehr nützlich.

Epson PX-8

CP/M-Business-Software ist darauf lauffähig. Das bekannte Textverarbeitungsprogramm Wordstar ist im Lieferumfang enthalten.

Tandy TRS-80

Modell 100/NEC

PC-8201A/Olivetti M10

Es handelt sich um denselben Rechner in verschiedenen Gehäusen. Die Textverarbeitung ist gut, und er verfügt über mehrere Schnittstellen.

Die Abbildung zeigt außerdem ein batteriebetriebenes Modem von Olivetti.


Modell	Standard-Speicher	Erweiterbar auf	Bildschirm-darstellung	Gewicht
Casio FP-200	8 K	32 K	8 x 20	1,4 kg
Epson HX-20	16 K	32 K	4 x 20	1,8 kg
Epson PX-8	64 K	64 K + 120 K*	8 x 80	2,3 kg
NEC PC-8201A	16 K	64 K + 32 K*	8 x 40	1,8 kg
Olivetti M10	8 K	32 K	8 x 40	1,8 kg
Tandy TRS-80 Modell 100	8 K	32 K	8 x 40	1,8 kg

*Der NEC faßt ein 32 K-RAM-Steckmodul, der Epson PX-8 wird durch 120 K-RAM erweitert.

Prüfroutinen

In diesem Abschnitt wollen wir den Weg zweier Baugruppen nachvollziehen – von der Planung eines Prüfbit-Generators und einer Vorrangschaltung führt er über die Wahrheitstabelle und deren Vereinfachung mit Boolescher Algebra bis zum kompletten Schaltplan.

Bevor wir uns mit der Erstellung der Baugruppen befassen, müssen wir ein weiteres Logikgatter genauer kennenlernen – das ausschließende OR-(XOR)-Gatter. Dieses Bauteil ist zwar schon früher erwähnt worden, seine Boolesche Algebra und das Schaltungssymbol wurden jedoch noch nicht erklärt:

Wahrheitstabelle			Schaltungssymbol
A	B	C	
0	0	0	
0	1	1	
1	0	1	
1	1	0	

Die Wahrheitstabelle zeigt, daß sich der Ausgang C auf zwei Arten darstellen läßt:

- a) $C = A \oplus B = \bar{A}.B + A.\bar{B}$
 b) $\bar{C} = \bar{A} \oplus \bar{B} = \bar{A}.\bar{B} + A.B$

Der zweite Ausdruck ergibt sich aus den Fällen, in denen C nicht Eins ist. Das XOR-Gatter kommt im ersten Beispiel zum Einsatz:

umgebaut werden. Die Wahrheitstabelle befindet sich am linken Rand dieser Seite. Hier die entsprechende k-Tafel:

	A		\bar{A}		
B	B	1		1	C
			1		
	1		1		
		1		1	
		D			

Der symmetrische Aufbau der Tafel macht die Vereinfachung unmöglich, weil sich aus den Werten keine Gruppen bilden lassen. Der Ausdruck für P lautet:

$$P = \bar{A}.\bar{B}.\bar{C}.D + \bar{A}.\bar{B}.C.\bar{D} + \bar{A}.B.\bar{C}.\bar{D} + \bar{A}.B.C.D + A.\bar{B}.\bar{C}.\bar{D} + A.\bar{B}.C.D + A.B.\bar{C}.\bar{D} + A.B.C.D$$

Die rot- und die blaugedruckten Ausdrücke können zusammengefaßt werden:

$$P = (\bar{A}.\bar{B} + A.B).(\bar{C}.D + C.\bar{D}) + (\bar{A}.B + A.\bar{B}).(\bar{C}.\bar{D} + C.D)$$

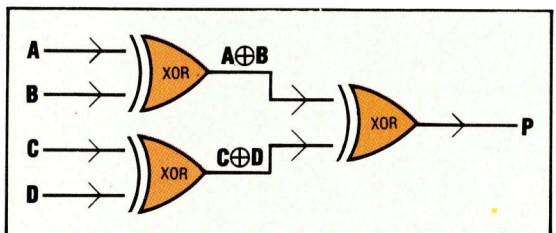
Hier läßt sich der Ausdruck für das XOR-Gatter (siehe oben) zur weiteren Vereinfachung einsetzen:

$$P = (\bar{A} \oplus \bar{B}).(\bar{C} \oplus D) + (A \oplus B).(C \oplus \bar{D})$$

Wenn der Inhalt jeder Klammer als Eingabe zum XOR-Gatter benutzt wird, reduziert sich der Ausdruck noch weiter:

$$P = (A \oplus B) \oplus (C \oplus D)$$

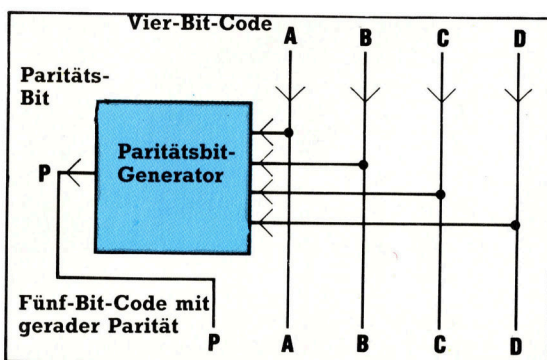
Das entspricht einer Kaskadenschaltung aus XOR-Gattern:



Durch ein weiteres XOR-Gatter wird die Schaltung zum Paritätsprüfer: Das Gatter vergleicht

Wahrheitstabelle

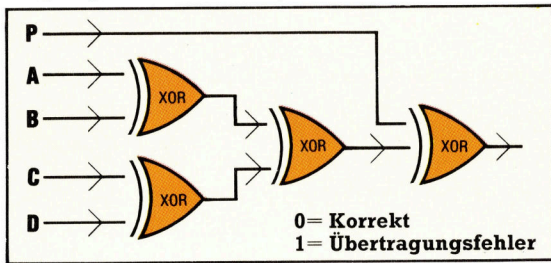
A	B	C	D	P
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



Paritäts- oder Prüfbits sind ein wichtiger Faktor beim Entwurf von Datenübermittlungs-Systemen. Prüfbits werden einem Datencode hinzugefügt, so daß jede einzelne Codegruppe stets eine gerade Anzahl (bzw. nach Vereinbarung auch ungerade Zahl) von Einsen enthält. (Gerade Anzahl = „Even Parity“, ungerade Anzahl = „Odd Parity“.) Das Paritätsbit erlaubt die Prüfung der korrekten Datenübermittlung. Unsere Schaltung erzeugt das Prüfbit für eine Vier-Bit-Eingabe. Mit kleinen Änderungen kann sie auch als Prüfschaltung für eingehende Daten

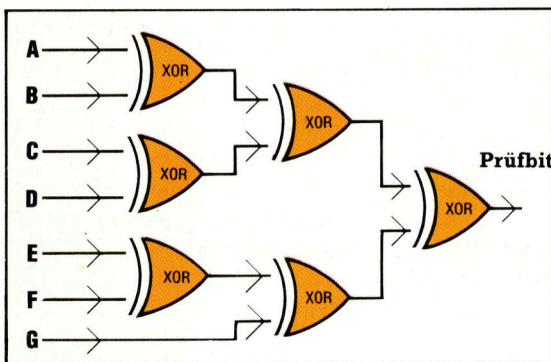
dazu das eingehende Prüfbit mit einem im Empfangsgerät erzeugten Paritätsbit.

Empfänger Fünf-Bit-Code



In der Praxis wird für die Datenübertragung meist der ASCII-Code verwendet, der aus sieben Datenbits und einem Paritätsbit besteht. Ein Prüfbit-Generator für ASCII hätte diese Schaltung:

Sieben-Bit-Daten



Die meisten Computer regeln ihren Datenfluß über ein System von Vorrangschaltungen. Die CPU kann durch das Signal eines Peripheriegerätes aufgerufen werden (Interrupt). Manchmal geschieht das von mehreren Peripheriegeräten gleichzeitig. Unsere Vorrangschaltung verbindet vier Peripheriegeräte und schaltet nur den Aufruf des Gerätes mit der höchsten Priorität an den Computer durch.

Für den Aufruf des Peripheriegerätes sind zwei Leitungen nötig. Eine dritte Ausgangsleitung gibt an, ob gerade ein Aufruf stattfindet. Die Geräte sollen P, Q, R und S heißen, wobei P die höchste und S die niedrigste Priorität hat. Die Ausgangsleitungen A und B melden das betreffende Peripheriegerät, Z ist die Rufleitung. Bedingungen, die unsere Vorrangschaltung ignorieren soll, sind in der Wahrheitstabelle jeweils mit X bezeichnet.

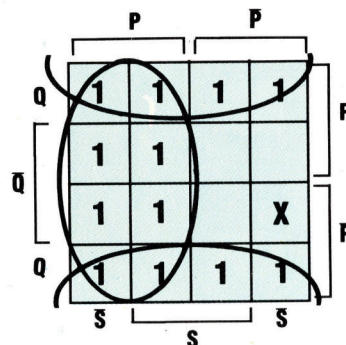
P	Q	R	S	A	B	Z
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Zum besseren Verständnis hilft die Betrachtung der letzten Zeile: Hier ruft P, das Gerät mit höchster Priorität, den Computer auf – alle Si-

gnale der niederwertigeren Geräte werden infolgedessen ignoriert.

Die drei Ausgangsleitungen müssen einzeln analysiert werden. Beginnen wir mit A, deren k-Tafel so aussieht:

Für A

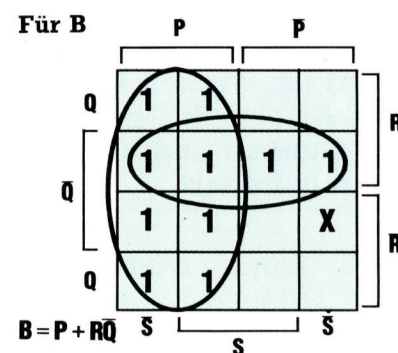


Der zu ignorierende Fall für die Ausgangsseite von A ist in der k-Tafel mit X bezeichnet. Anders sieht es auf der Eingangsseite aus: Wenn P = Eins und Q, R und S zu ignorierende Bedingungen sind, müssen alle acht Felder der k-Tafel ausgefüllt werden, in denen P = Eins ist. Mit der k-Tafel finden wir die Vereinfachung zu diesem Ausdruck:

$$A = P + Q$$

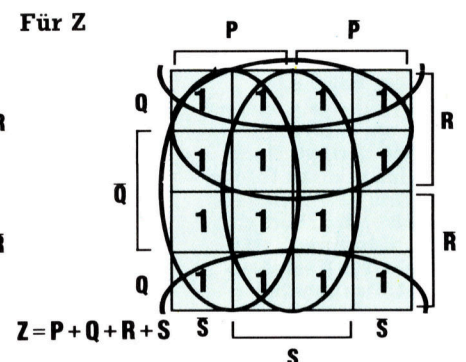
Hier die k-Tafeln für Leitungen B und Z:

Für B



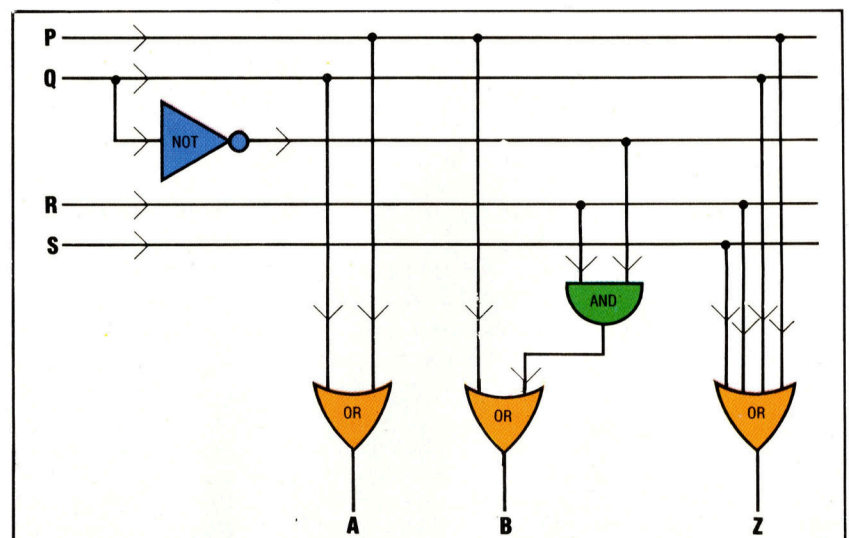
$$B = P + RQ$$

Für Z



$$Z = P + Q + R + S$$

Aus der Verbindung der drei Ausdrücke ergibt sich dieser Schaltungsaufbau:





Elementare Kräfte



„Archon“ ist dem Schachspiel verwandt. Wird eine Figur „aufgefordert“, ein anderes Feld zu betreten, marschiert, hüpfet oder fliegt sie – je nachdem, über welche Fähigkeiten sie verfügt. Nach Erreichen eines Feldes, das von einem Gegner besetzt ist, ändert sich die Bildschirmdarstellung – der Kampf beginnt. Das Ergebnis hängt vom Geschick des Spielers und der relativen Stärke der Gegner ab.

Durch Kombination von Schach-elementen und schnell bewegter Grafik, wie sie aus den Spielhallen bekannt ist, werden bei „Archon“ von Ariolasoft Geschick, strategisches Denken und Reaktionsvermögen gefordert.

Das Programm „Archon“ fasziniert den Schachspieler, begeistert aber auch den geübten Action-Spieler. Spielgrundlage ist der Kampf zwischen „Licht“ und „Dunkel“. Bei Spielbeginn wird eine Art Schlachtfeld gezeigt, auf dem die gegnerischen Parteien in Reihen wie beim Schach aufgestellt sind. Die Figuren, die über unterschiedliche Fähigkeiten verfügen, tragen Namen wie „Phoenix“ und „Ritter“ auf der „Licht“-Seite, wogegen „Zombies“, „Kobolde“ und „Drachen“ das „Dunkel“ vertreten. Einige Figuren können „fliegen“, was bedeutet, daß sie vor ihnen befindliche Figuren überspringen dürfen, wogegen andere lediglich auf dem „Boden“ beweglich sind.

Das Spielfeld selbst ist, ähnlich wie ein Schachbrett, in schwarze und weiße Felder unterteilt. Im Spielverlauf aber verändern sich die Farben mancher Quadrate. Grund dafür ist, daß die Lichtkräfte auf weißen Quadraten stärker sind und die Kräfte des Dunkels auf schwarzen Feldern.

Ziel bei „Archon“ ist es, die fünf „Kraftpunkte“ zu besetzen, von denen sich vier kreuzförmig angeordnet an den Rändern des Feldes befinden. Bei der Eröffnungsstrategie

müssen die Figuren von der entgegengesetzten Farbe entfernt werden, da sie dort besonders empfindlich für Angriffe sind. Die eigene Farbe verschafft mehr „Kraft“. Dadurch wird auch die zweite Reihe geöffnet, womit „bodengebundene“ Figuren gezogen werden können und der Aufbau einer geschlossenen Verteidigungslinie möglich ist, die der Gegner nicht durchbrechen kann.

Bis zu dieser Phase entspricht das Programm einem auf Computer übertragenen Schachspiel. Der eigentliche Unterschied aber wird deutlich, wenn man ein Quadrat zu besetzen versucht, das bereits von einem gegnerischen Piktogramm belegt ist. Führt man seine Figur auf dieses Feld, wird die gegnerische nicht geschlagen, sondern es erscheint eine „Nahaufnahme“ auf dem Bildschirm. Die Kontrahenten werden zu beiden Seiten des Feldes positioniert. Dort findet sich auch eine Maßeinheit, die die Stärke der eingesetzten Figur anzeigt. Nun beginnt eine Auseinandersetzung im Spielhallen-Stil. Mit jedem Treffer schwindet die Stärke der entsprechenden Figur, was an einem Meßbalken ablesbar ist. Ist die Kraft völlig erloschen, kann der Gegner das Feld einnehmen. Da sich die Spielfiguren in Stärke und Fähigkeiten voneinander unterscheiden, sind diese Kämpfe unterschiedlich schwer. So kann beispielsweise ein Drache Flammen auf einen Gegner fauchen – und zwar über die ganze Feldbreite. Ein Ritter dagegen muß dicht an seinem Feind sein, um treffen zu können. Der Kampf wird durch im Spielverlauf variierte Barrieren weiter erschwert.

Auf beiden Seiten steht ein Piktogramm zur Verfügung, mit dem man zaubern kann. Die Zaubersprüche können aber nur einmal für das Piktogramm verwendet werden. So ist es möglich, eine Figur „wiedezubeleben“, seinen Spielraum zu erweitern oder eine „Elementar-Figur“ herbeizurufen. Sie wird herbeigezaubert, wenn der Feind stärker als die eigene Spielfigur ist.

Man kann „Archon“ allein (gegen den Computer) oder mit einem Partner spielen. Obwohl der Computer nur ein durchschnittlicher Stratege ist, erweist er sich durchaus als ein sehr fähiger Gegner.

Archon: Für Apple, Atari und C64

Herausgeber: Ariolasoft (Electronic Arts)

Autoren: Anne Westfall, Jon Freeman, Paul Reiche III

Joystick: Erforderlich

Programm: Diskette

Kleine Scheibe

Das modulare Konzept des Schneider CPC464 und sein hervorragendes BASIC verschafften dem Rechner einen guten Ruf. Die Diskettenstation eröffnet dem Anwender zusätzliche interessante Möglichkeiten.

Trotz aller Beliebtheit wies der Schneider CPC464 bisher ein erhebliches Manko auf: Es fehlte ein schneller Massenspeicher – das eingebaute Cassettenlaufwerk war kein ausreichender Ersatz. Die Diskettenstation wurde schon bei der Vorstellung des Rechners angekündigt, und viele Käufer hatten auf eine baldige Lieferbarkeit gesetzt. Sie mußten jedoch über ein Jahr warten, bis das System endlich in den Handel kam.

Das Diskettenpaket besteht aus dem Laufwerk selbst, einer Steuereinheit mit Flachbandkabel für den Anschluß an den rückwärtigen Erweiterungsstecker des Rechners und der Anleitung. Zusätzlich werden das CP/M-Betriebssystem sowie Dr. LOGO mitgeliefert.

Das Laufwerk arbeitet mit 3-Zoll-Disketten nach Hitachi-Norm. Diese Entscheidung ist etwas ungewöhnlich, da sich eher die 3½-Zoll-Disketten von Sony durchzusetzen scheinen. Zahlreiche Hersteller – unter anderem Apple und Apricot – haben sich bereits auf die 3½-Zoll-Norm festgelegt, und außer Schneider (der englische Hersteller ist Amstrad) hat bisher keine größere Firma die Hitachi-Norm aufgegriffen.

Beim Öffnen des Geräts sieht man die beiden Motoren für den Spindelantrieb und die Steuerung des Schreib/Lese-Kopfes. Dahinter liegt das Netzteil in einem Blechgehäuse, das zur Abschirmung von Wärme und Streufeldern dient. Der Netzschalter befindet sich an der Rückseite über dem Interface-Anschluß.

Die verwendeten Disketten sehen zwar anders aus als die Sony-Konkurrenten, aber das Funktionsprinzip ist das gleiche. Das Gehäuse mit den Abmessungen 100 x 80 x 4 mm trägt über dem Schreib/Lese-Fenster für den Magnetkopf eine Metallabdeckung als Schutz gegen Berühren. Bei den Schneider-Disketten ist die Abdeckung auf der Innenseite des Gehäuses angebracht, bei Sony dagegen außen.

Das Laufwerk arbeitete beim Test schnell und zuverlässig; ohne Schwierigkeiten ließen sich in Sekundenschnelle Dateien auffinden und laden. Der Geräuschpegel schien deutlich niedriger als bei den meisten 5¼-Zoll-Floppies.

Die Systemdiskette enthält drei Programme; das Amstrad-eigene Betriebssystem AmsDOS, das bekannte CP/M-DOS von Digital Research und Dr. LOGO aus dem gleichen Hause. Dr. LOGO ist eine Variante der Programmier-

sprache LOGO, die für Grafikerstellung und im Unterricht zunehmend Verbreitung findet.

Das AmsDOS ist wohl das schwächste von den drei Systemen. Wenn Sie damit arbeiten wollen, müssen Sie zunächst das CP/M laden. Im Unterschied zu dem selbständigen Betriebssystem CP/M stellt AmsDOS nur Diskettenbefehle bereit, die in Verbindung mit dem Strichsymbol **■** (Shift @) aufzurufen sind, zum Beispiel **■DRIVE** und **■DIR**.

Umständliches AmsDOS

Die Handhabung der AmsDOS-Dateien erfolgt reichlich umständlich. Um etwa mit ERASE eine Datei zu löschen, müssen Sie den Dateinamen zunächst als String definieren – erst dann können Sie die Datei durch Löschen des Strings loswerden. Für die Datei KUNDEN.\$\$\$ beispielsweise sieht die Löschprozedur dann zum Beispiel so aus:

```
A$= "KUNDEN.$$$"
■ERA, aA$
```

Noch komplizierter wird das Verfahren beim Umbenennen einer Datei (RENAME), denn dazu müssen dem alten und dem neuen Namen erst Strings zugewiesen werden, die man dann ändern kann. Vielen Benutzern mag es

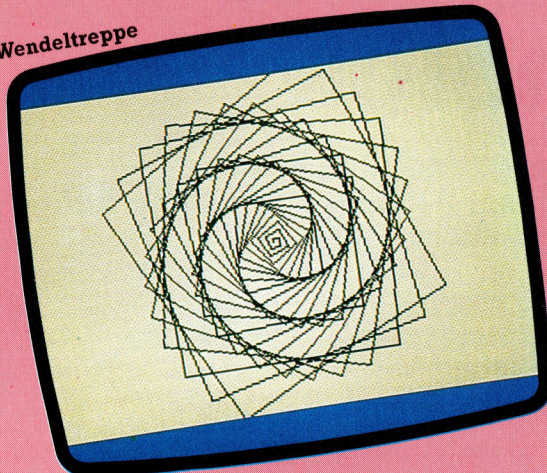
Mit dem Diskettenlaufwerk DDI-1 von Amstrad (in Deutschland von Schneider vertrieben) können CPC464-Besitzer ihr System so ausbauen, daß es den Anforderungen an einen Rechner für „ernsthafte“ Anwendungen genügt. In Verbindung mit den drei Programmpaketen CP/M, AmsDOS und Dr. LOGO, die zum Lieferumfang gehören, ergibt sich eine erhebliche Erweiterung der Einsatzmöglichkeiten.



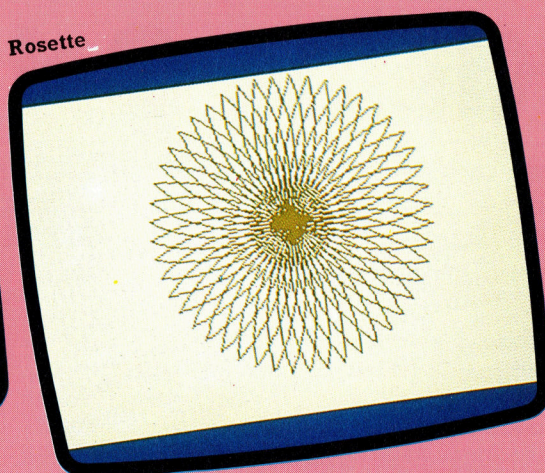
Turtle-Grafik

Diese Grafiken wurden mit Dr. LOGO produziert. Obgleich bei Dr. LOGO die Veränderung von Vorder- und Hintergrundfarben vorgesehen ist, gibt es keine Möglichkeit, die Form der Turtle zu verändern oder die gezeichneten Figuren farbig auszufüllen. Dafür wird von Dr. LOGO die Tonerzeugung unterstützt.

Wendeltreppe



Rosette



einfacher erscheinen, für solche Fälle CP/M zu laden, wobei eine einzige Befehlszeile dasselbe leistet. Aber auch das CP/M ist nicht problemlos: Das Schneider-BASIC läuft nicht unter CP/M, so daß Sie für den Diskettenbetrieb mit BASIC ausschließlich auf das Amstrad angewiesen sind.

Die Systemdiskette enthält die CP/M-Version 2.2, die in den letzten Jahren auf zahllosen 8-Bit-Rechnern implementiert worden ist. Mit dem CP/M-System werden auch die üblichen nicht-residenten Kommandoroutinen bereitgestellt. Diese „transienten“ Segmente befinden sich auf der Diskette, werden nur bei Bedarf ins RAM geladen und anschließend vom System wieder gelöscht. Problematisch ist dabei, daß Sie eigentlich zwei Laufwerke am Rechner brauchen, wenn Sie unter CP/M mit vielen transienten Routinen arbeiten wollen (etwa mit PIP zum Überspielen von Dateien zwischen Peripheriegeräten): Sie benötigen eins für die Systemdiskette, damit die transienten Kommandos leicht und schnell erreichbar sind, und eins für die Floppy mit den Dateien.

Abgesehen davon, daß der ständige Wechsel zwischen Daten- und Systemdiskette beim Einzellaufwerk ziemlich lästig ist, bietet CP/M bei vielen Befehlen keine Ausweichmöglichkeit, wenn das zweite Laufwerk fehlt und Sie die Dateien nicht direkt von Diskette zu Diskette überspielen können. Um dem zu begegnen, hat Amstrad besondere Kommandos für den Einzellaufwerk-Betrieb vorgesehen. Über FILECOPY erhält man Anweisungen für den Diskettenwechsel beim Kopieren einzelner Dateien und über DISCCOPY für das Kopieren ganzer Disketten.

Rückgriff auf CP/M

Schneider möchte mit seinen Rechnern nicht unbedingt die technologische Spitze bieten, sondern verläßt sich lieber auf ein erprobtes und bewährtes Konzept für vielseitigen Gebrauch. Deshalb entschied sich die Firma auch noch für den Z80-Prozessor, obwohl sich die

Konkurrenz schon auf 16-Bit-Rechner umstellt. Dazu paßt auch der Rückgriff auf das universelle, jedoch schon etwas betagte CP/M-System, für das eine umfangreiche Programm-bibliothek zur Verfügung steht.

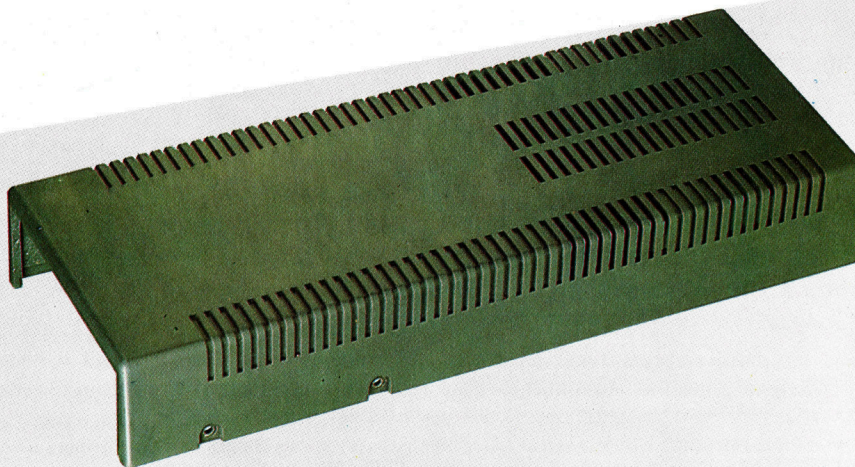
Trotz der Schwierigkeiten beim CP/M-Betrieb mit nur einem Laufwerk gebührt Amstrad ein großes Lob für die Implementierung dieses leistungsfähigen Betriebssystems auf einem Heimcomputer. Noch vor wenigen Jahren kostete ein CP/M-Rechner über siebentaussend Mark, und jetzt bietet Schneider ein solches System für weniger als die Hälfte an.

Grafik mit Dr. LOGO

Dr. LOGO ist ein Dialekt der Sprache LOGO, die inzwischen sehr verbreitet ist und in der Ausbildung benutzt wird. Dr. LOGO läuft unter CP/M, das zuerst geladen werden muß. Der komfortable CP/M-Diskettenbetrieb kommt also auch Dr. LOGO zugute. Zum Beispiel gestattet CP/M die Verwendung der freizügigen „Wildcards“: LOGO-Files können dabei ohne die vollständige Namenseingabe aufgerufen werden, wobei nach Eintippen eines Namens-teils alle Dateien angesprochen sind, in deren Namen die eingegebene Zeichenfolge enthalten ist.

Über Dr. LOGO sind beim CPC464 Grafikbetrieb und Tonerzeugung in der Weise möglich, die auch in unserem LOGO-Kurs behandelt wurde. Auch diese Abfrage des Joysticks ist möglich, so daß Sie mit Dr. LOGO Spiele für Joystick-Steuerung schreiben können.

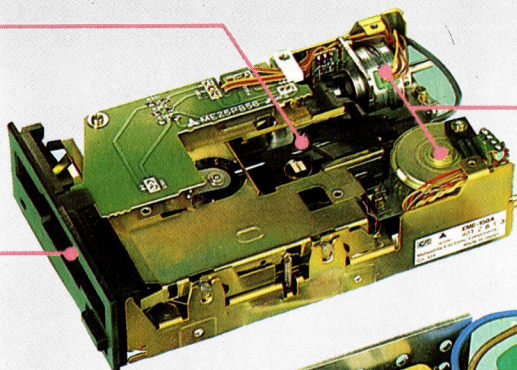
Obwohl Dr. LOGO sehr gut implementiert ist und beim Vergleich mit vielen anderen LOGO-Versionen gut wegkommt, ist die Frage der Effizienz unter den Hardware-Voraussetzungen des CPC464 nicht ganz vom Tisch. So führte das LOGO-Mosaikprogramm, das durch vielfache Wiederholung des gleichen Musters den Bildschirm füllte und auf dem Commodore 64 einwandfrei lief, beim CPC464 zum Überlaufen des Speichers – vielleicht ein Zeichen, daß die Anpassung nicht ganz optimal ist. Davon abge-


Schreib-/Lese-Kopf

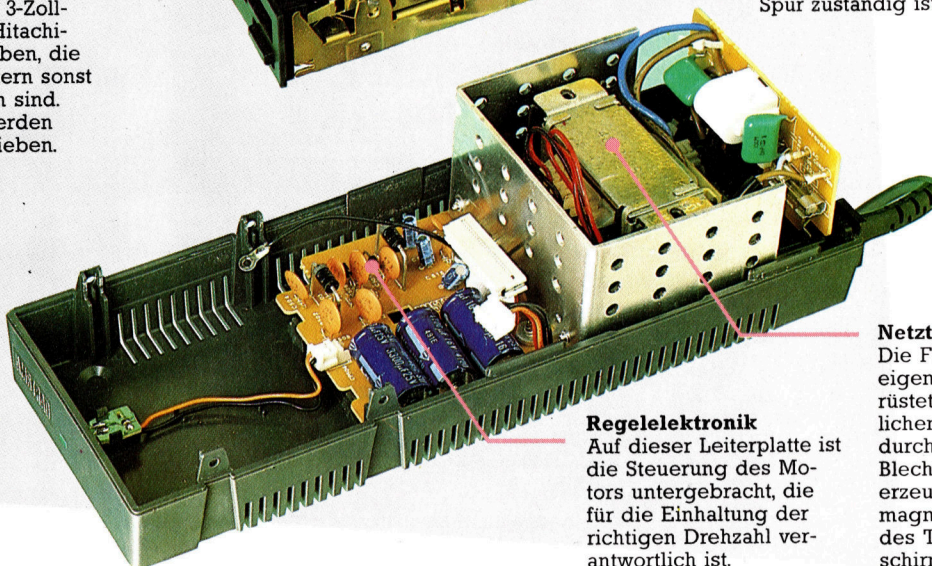
Der Schreib-/Lese-Kopf registriert die Änderungen des Magnetfelds auf der Diskette und setzt sie in elektrische Signale um.

Diskettenschlitz

Hier werden die 3-Zoll-Disketten nach Hitachi-Norm eingeschoben, die bei Heimcomputern sonst kaum anzutreffen sind. Die Disketten werden einseitig beschrieben.


Motoren

Der Motor unten im Bild treibt die Diskettenspinde an, während der obere für die Einstellung des Schreib-/Lese-Kopfes auf die gewünschte Spur zuständig ist.


Regelelektronik

Auf dieser Leiterplatte ist die Steuerung des Motors untergebracht, die für die Einhaltung der richtigen Drehzahl verantwortlich ist.

Netzteil

Die Floppy ist mit einem eigenen Netzteil ausgerüstet. Vom empfindlichen Laufwerk ist es durch ein gelochtes Blech getrennt, das die erzeugte Wärme und das magnetische Streufeld des Transformators abschirmen soll.

Schneider DDI-1
ABMESSUNGEN

290 x 105 x 75 mm

SPEICHERKAPAZITÄT

Systemdiskette: 169

KByte

Datendiskette: 178

KByte

SCHNITTSTELLEN

Parallelschnittstelle mit 34adrigem Kabel, das zwei Laufwerke versorgen kann.

DOKUMENTATION

Die mitgelieferte Anleitung ist unsystematisch aufgebaut; bestimmte Informationen sind manchmal schwer zu finden.

STÄRKEN

Die Verfügbarkeit von CP/M und Dr. LOGO macht den CPC464 zu einem leistungsfähigen Rechner.

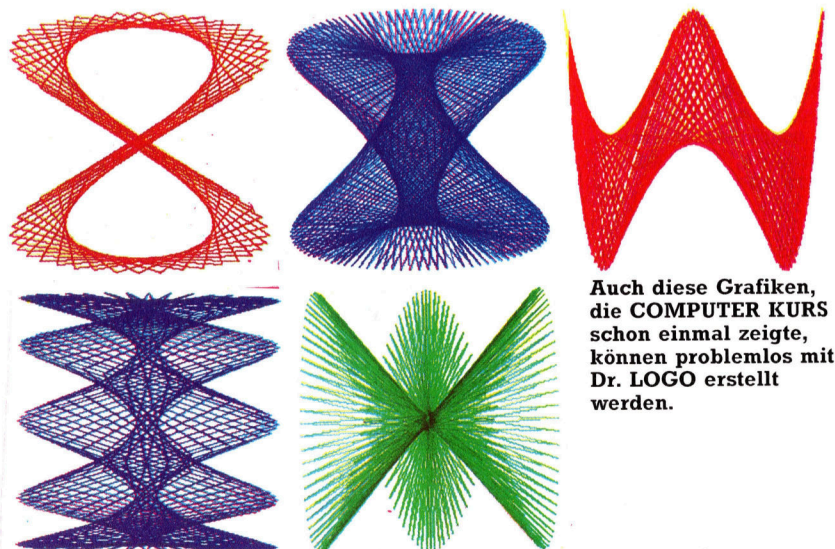
SCHWÄCHEN

Das AmsDOS ist nicht sehr überzeugend, und bei Verwendung nur eines Laufwerks können weder AmsDOS noch CP/M die Erwartungen anspruchsvoller Benutzer erfüllen.

sehen ist Dr. LOGO schnell und benutzerfreundlich und für Anfänger ideal.

Wie bei jeder Diskettenstation für Heimcomputer wird auch hier der Erfolg an der Software-Unterstützung hängen. Amstrad hat schon eine Reihe von Programmen für das Laufwerk entwickelt, die sich größtenteils eher an den „ernsthaften“ Anwender wenden. Dazu gehören ein Datenbanksystem, ein Textverarbeitungspaket und Sprachen wie PASCAL.

Mit dem Vertrieb der Diskettenstation hat Schneider einen aussichtsreichen Schritt unternommen, den CPC464 zu einem Mittelklasse-Rechner aufzuwerten. Die Ergänzung durch das Laufwerk erhöht die Konkurrenzfähigkeit des Rechners und zeigt, daß Schneider den CPC464 aus dem abflauenden „Spiel“-Computermarkt herausheben möchte.



Auch diese Grafiken, die COMPUTER KURS schon einmal zeigte, können problemlos mit Dr. LOGO erstellt werden.

3-D-Kurven

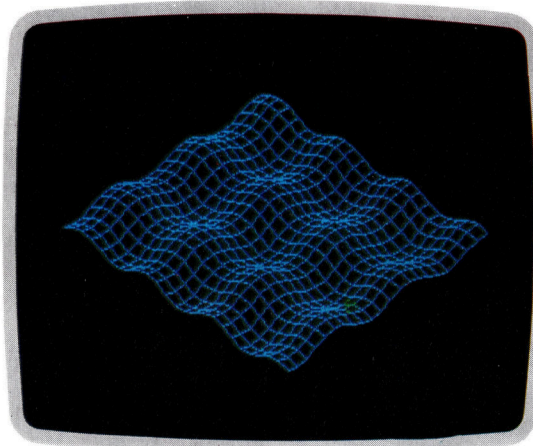
Mathematische Formeln sind zur Erzeugung von Computer-Grafiken sehr wichtig. In diesem Artikel wird gezeigt, wie man dreidimensionale Figuren erstellt, indem man verschiedene mathematische Funktionen in ein Programm integriert.

Die Grafik-Fähigkeiten der meisten Computer eignen sich hervorragend für die Umsetzung von mathematischen Gleichungen.

Die dargestellten Grafiken wurden unter Verwendung der anschließend gezeigten Programme produziert. Sie sind alle als Graphen in drei Dimensionen berechnet worden. Ein 3-D-Graph besteht aus vielen zweidimensionalen Graphen, die mit kleinen Unterschieden zur gleichen Zeit dargestellt werden. Da ein Computer Bilder nur in zwei Dimensionen darstellen kann, ist das Ergebnis nicht wirklich dreidimensional. Die Art der Darstellung bewirkt jedoch den optischen Tiefeneffekt.

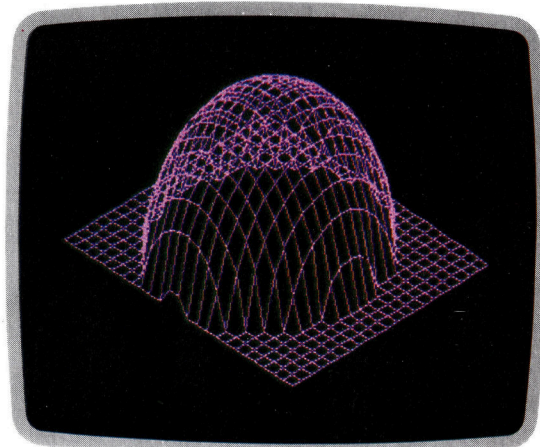
Die gezeigten Programme berechnen die Werte einer Gleichung mit den zwei Variablen X und Z. Das Ergebnis (Y) wird für mehrere Werte von X und Z berechnet. Jeder Wert von Y wird zur Darstellung eines Punktes auf dem Bildschirm verwendet, wobei die Werte von Y mit Punkten auf der vertikalen Achse in Bezug gebracht werden – das bedeutet, je höher der Wert von Y, desto näher wird der Punkt am oberen Bildschirmrand dargestellt. Benachbarte Punkte werden durch gerade Linien miteinander verbunden, wodurch der Eindruck einer Kurve entsteht. Die Kurven einer Richtung repräsentieren Graphen von X und Y, wobei Z konstant ist. Die Kurven, die diese Graphen durchschneiden, sind Graphen von Y und Z mit X als Konstante (in diesem Fall werden sie auf einer Ebene mit Y- und Z-Achsen gezeichnet, also eine Ebene, die genau um 90° zur normalen X- und Y-Achse eines zweidimensionalen Graphen gedreht ist).

Modifizieren Sie das Programm, indem Sie die unter den Bildern gezeigten Programmzeilen einfügen.



$$170 \ Y = (\sin(X) + \cos(Z)) / 60$$

Diese Darstellungen sind auch für diejenigen interessant, die sich sonst nicht für Mathematik interessieren. Es ist unterhaltsam (und ein wenig schwierig) mit einer Gleichung ein bestimmtes Muster zu erzeugen. Um den hier gezeigten Graphen zu verändern, muß die Funktion in Zeile 170 des BASIC-Programms modifiziert werden. Einige Funktionen sind etwas komplizierter, so daß sie sich über mehrere Programmzeilen erstrecken. Benutzen Sie dafür die Zeilennummern von 151 bis 179.



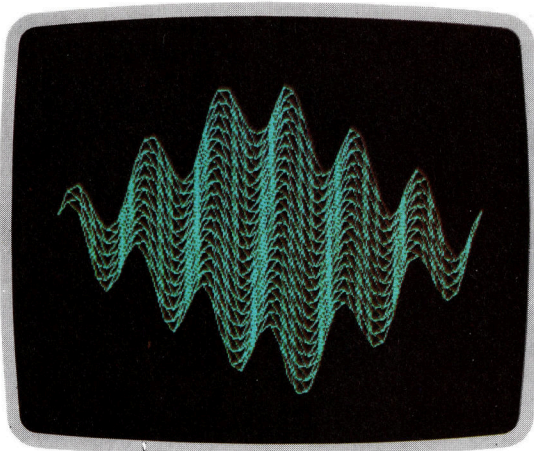
$$165 \ C = 60 - X^2 - Z^2$$

$$170 \ Y = \text{SQR}(C * (\text{SGN}(C) + 1)) / 45$$

Bei der Auswahl einer Funktion müssen Sie darauf achten, daß die errechneten Werte nicht zu groß werden und der Graph die Bildschirmgrenzen nicht überschreitet. Damit die Darstellung im vorgegebenen Rahmen bleibt, müssen die Werte eventuell durch eine entsprechende Zahl dividiert werden.

Dieses Programm kann mit leichten Änderungen auf den meisten Computern verwendet werden. Der erste Programmabschnitt ist für den Bildschirmaufbau zuständig. Der zweite Teil des Programms wird zur Speicherung der errechneten Werte verwendet, mit denen der Graph gezeichnet wird. Diese Ergebnisse werden in einem Array aufbewahrt. Die Dauer der Berechnungen ist abhängig von der gewählten Funktion. Während dieser Zeit hat es den Anschein, als mache der Computer nichts. Würden die Berechnungen jedoch während des Zeichnens der Linien durchgeführt, wäre der Zeitbedarf doppelt so groß.

Damit Sie ein wenig experimentieren kön-

170 $Y = \sin(X+Z)/12$

nen, haben wir mehrere Funktionen zur Auswahl dargestellt. Sie sollten jedoch auch eigene Funktionen in das Programm eingeben und so Ihre eigenen Graphen entwickeln. Achten Sie jedoch darauf, daß der Graph innerhalb der Bildschirmgrenzen dargestellt wird und daß keine illegalen mathematischen Berechnungen (etwa Divisionen durch Null) durchgeführt werden.

Fehlerquellen

Um eine derartige Division zu vermeiden, addieren Sie am besten eine sehr kleine Konstante (z. B. 0.00001) zu jeder Variablen, deren Wert sich auf 0 reduzieren könnte. Die einzige Sicherheitsmaßnahme gegen das Wurzelziehen aus negativen Zahlen ist die Verwendung der ABS-Funktion. Dadurch wird sichergestellt, daß die Zahlen positiv sind.

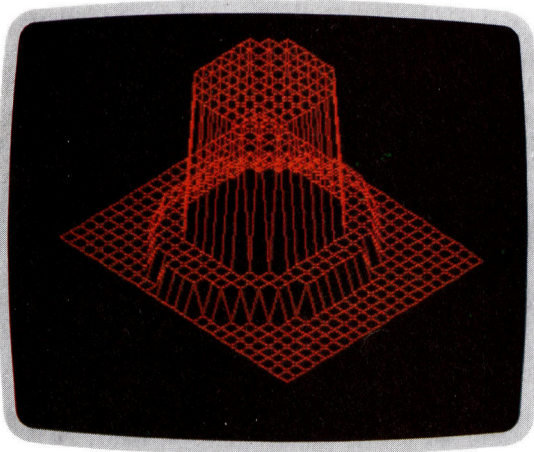
Einige interessante Darstellungen können durch die allgemeinen mathematischen Funktionen wie SIN, COS, LOG usw. produziert werden. Andere Möglichkeiten bieten sich durch Funktionen, die es nur auf Computern gibt, wie beispielsweise INT, SGN und ABS.

Das hier gezeigte Programm kann sicherlich in einigen Punkten verbessert werden. Sie

können das Listing auch dahingehend ändern, daß es automatisch jede Funktion so anpaßt, daß sie innerhalb der Bildschirmgrenzen dargestellt wird. Oder versuchen Sie doch einmal, Punkte in einer dritten Richtung zu zeichnen, also Kurven für X und Z, wobei Y konstant bleibt. Doch selbst wenn Sie das Programm nur in der hier gezeigten Form verwenden, ist es sehr interessant zu beobachten, was aus den einfachen Gleichungen entsteht. Die Ergebnisse werden Sie überraschen.

Programmversion für den Acorn B

```
10 REM * GRAPH PLOTTING
20:
30 REM * SET UP SCREEN
40 ACROSS=1280:TALL=1024 :UP=-1
50 XGAP=25:ZGAP=15
60 WIDE=INT(ACROSS/XGAP/2)
70 DEPTH=INT(TALL/ZGAP/3)
80 MODE4:CLS:PRINTTAB(12)"CALCULATING"
90:
100 REM * CALCULATE GRAPH
110 START=20
120 DIM G(WIDE,DEPTH)
130 FOR A=-DEPTH/2 TO DEPTH/2
140 FOR B=-WIDE/2 TO WIDE/2
150 X=A*20/WIDE:Z=B*20/DEPTH
160 REM * INSERT FUNCTION BELOW HERE
170 Y=(SIN(X)+COS(Z))/60
180 G(B+WIDE/2,A+DEPTH/2)=Y*UP*TALL
190 NEXT B:NEXT A:CLS
200:
210 REM * DRAW GRAPH ; X-Y PLANE
220 FOR Z=1 TO DEPTH
230 XBASE=XGAP*Z
240 ZBASE=TALL/2+Z*ZGAP+START*UP
250 XOLD=XBASE+XGAP
260 ZOLD=ZBASE-ZGAP-G(1,Z)
270 FOR X=1 TO WIDE
280 XNEW=XBASE+X*XGAP
290 ZNEW=ZBASE-X*ZGAP-G(X,Z)
300 PLOT 4,XOLD,ZOLD:PLOT 5,XNEW,ZNEW
310 XOLD=XNEW:ZOLD=ZNEW
320 NEXT X:NEXT Z
330:
340 REM * DRAW GRAPH ; Z-Y PLANE
350 FOR X=1 TO WIDE
360 XBASE=XGAP*X+DEPTH*XGAP
370 ZBASE=TALL/2-X*ZGAP+DEPTH*ZGAP+START*UP
380 ZOLD=ZBASE-ZGAP-G(X,DEPTH-1)
390 XOLD=XBASE-XGAP
400 FOR Z=0 TO DEPTH-1
410 XNEW=XBASE-Z*XGAP
420 ZNEW=ZBASE-Z*ZGAP-G(X,DEPTH-Z)
430 PLOT 4,XOLD,ZOLD:PLOT 5,XNEW,ZNEW
440 XOLD=XNEW:ZOLD=ZNEW
450 NEXT Z:NEXT X
460:
470 REM * HOLD DISPLAY
480 GOTO 470
```



165 $C = X * X + Z * Z + 0.00001$
170 $Y = \text{SGN}(\text{INT}(23/C))/3 + \text{SGN}(\text{INT}(55/C))/15$

BASIC-Dialekte

Spectrum

Setzen Sie die LET-Anweisungen und folgende Zeilen ein:

```
40 LET ACROSS=256:LET TALL=176:LET UP=-1
50 LET XGAP=5:LET ZGAP=3
80 CLS
290 PLOT XOLD,ZOLD : DRAW XNEW-XOLD,ZNEW-ZOLD
410 PLOT XOLD,ZOLD : DRAW XNEW-XOLD,ZNEW-ZOLD
```

Oric-1/Atmos

Fügen Sie die folgenden Zeilen ein:

```
40 ACROSS=256:TALL=176:UP=-1
50 XGAP=5:ZGAP=3
80 HIRE
300 CURSET XOLD,ZOLD,1:DRAW XNEW-XOLD,ZNEW-ZOLD,1
430 CURSET XOLD,ZOLD,1:DRAW XNEW-XOLD,ZNEW-ZOLD,1
```




LLAMASOFT

Llamasoft ist ein Unternehmen, das auf den Fähigkeiten nur eines Mannes basiert – Jeff Minter. Wegen seiner witzigen Spiele für Commodore- und Atari-Rechner wurde Minter zum Star der Software-Welt.

Die meisten Software-Häuser werden als „Mini-Unternehmen“ mit einem oder zwei Programmierern gegründet. Haben sie erst einmal Erfolg, werden andere Programmierer engagiert und genau das, was den Erfolg ausmachte, geht verloren – aufgrund des Produktionsdrucks verschwinden die Eigenarten, und die Erzeugnisse des Hauses unterscheiden sich kaum von denen anderer. Llamasoft indes ist anders. „Revenge of the Mutant Camels“ hätte ebenso wenig aus einem anderen Hause kommen können wie „Sheep in Space“.

Diese Besonderheit von Llamasoft wurde von einem Mann geprägt. Zwar ist Jeff Minter gerade erst 22 Jahre alt, doch er zählt bereits zu den Stars der Software-Welt. Seine Vorliebe für Lamas, Kamele und andere behaarte Wesen aus der Tierwelt macht den berühmten „Llama“-Stil aus.

1981 begann Minter Programme zu schreiben, und sein erster Versuch im kommerziel-

len Software-Markt war seine „Defender“-Version für den VC 20, 1982 gründete er Llamasoft. Darauf folgten die beiden sehr erfolgreichen Programme „Traxx“ und „Gridrunner“.

Heute schreibt Minter Programme für den VC 20, den C 64 und Atari-Computer. Die Spectrum-Versionen werden von Salamander Software und Quicksilver produziert.

Die Llamasoft GmbH ist ein „Familienunternehmen“. Geschäftsführer sind Minter selbst, nebst Vater Patrick und Mutter Hazel. Sitz des



Unternehmens ist das Haus der Familie in Tadley, Hampshire. Patrick Minter hilft dem Sohn beim Programmieren, wogegen sich die Mutter um die Verwaltungsarbeit kümmert. Weiter sind zwei Assistenten, zwei Buchhalter und ein Grafiker bei Llamasoft beschäftigt.

Bis heute hat Minter 21 Spiele für Llamasoft programmiert, von denen viele von einem immer wiederkehrenden Thema geprägt sind – dem Kamel. Sein erstes Programm, in dessen Mittelpunkt dieses Tier stand, wurde 1982 für den Atari geschrieben, betitelt „Attack Of The Mutant Camels“, worauf später „The Revenge Of The Mutant Camels“ folgte.

Die Nachfrage nach Llamasoft-Programmen ist groß: Zum Start werden 10 000 Cassetten produziert. Weitere Auflagen in dieser Größenordnung folgen – abhängig vom Verkauf. Minter sieht jedoch keine wesentlichen Veränderungen in der Unternehmensphilosophie. Er meint dazu, daß ihn der geschäftliche Part nicht interessiert. Wie bisher will er sich dem Schreiben neuer Spiele widmen.



Llamasofts surrealistischer Stil wird mit dieser Bildschirmabbildung aus „Revenge Of The Mutant Camels“ grafisch verdeutlicht – dem Nachfolgeprogramm des erfolgreichen „Attack Of The Mutant Camels“

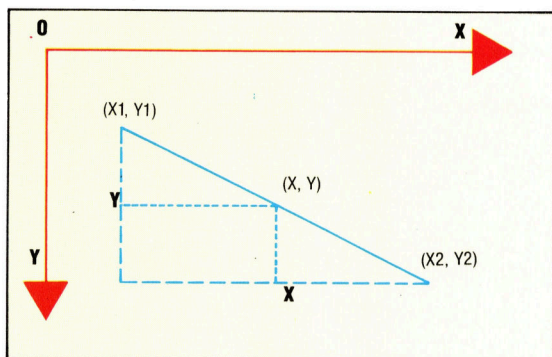


Starke Steigung

In diesem Artikel untersuchen wir eine Routine für den C64, die unter Verwendung des bereits gezeigten Plotsub-Programms Linien mit unterschiedlichen Steigungen zeichnet.

Eine Linie läßt sich auf dem Bildschirm am einfachsten durch die Berechnung des Steigungswinkels zwischen den beiden Endpunktkoordinaten darstellen. Dabei wird vom Anfangspunkt der Linie aus jeder nachfolgende Punkt über eine Inkrementierung der X-Koordinate und die Berechnung der entsprechenden Y-Koordinate aus dem Steigungswinkel so lange geplottet, bis der Endpunkt erreicht ist. Sehen wir uns die mathematischen Vorgänge dieser Technik genauer an.

Wenn wir die Endpunkte der Linie als $(X1, Y1)$ und $(X2, Y2)$ bezeichnen, dann ist der Steigungswinkel $G = (Y2 - Y1) / (X2 - X1)$. Das Bild zeigt eine Linie mit einem Punkt (X, Y) :



Der Steigungswinkel G zwischen dem Anfangspunkt und (X, Y) wird mit der Formel $(Y - Y1) / (X - X1)$ berechnet und gilt natürlich für die ganze Linie. Wenn wir daher G und die Formel gleichsetzen und die Gleichung etwas umformen, können wir damit die Y-Koordinaten für jeden Punkt der Linie berechnen:

$$\begin{aligned} (Y - Y1) / (X - X1) &= G \\ (Y - Y1) &= G(X - X1) \\ Y &= Y1 + G(X - X1) \end{aligned}$$

Besteht der Schritt zwischen X und $X1$ nur aus einer Einheit (d. h. der Wert von X wird jeweils nur um Eins inkrementiert), dann läßt sich die Formel vereinfachen auf:

$$Y = Y1 + G$$

Wenn wir daher am Endpunkt $(X1, Y1)$ anfangen und X inkrementieren, dann kann Y nur durch die wiederholte Addition des Neigungswinkels berechnet werden.

Diese Technik läßt sich auch von BASIC aus mit Plotsub einsetzen, doch müssen in jedem Fall folgende Punkte berücksichtigt werden:

○ Wenn der Wert von $X2$ kleiner ist als $X1$, muß X auf jeden Fall dekrementiert und nicht inkrementiert werden.

○ Wenn der Neigungswinkel größer als Eins wird, ergibt sich keine zusammenhängende Linie, da in diesem Fall Y bei jeder Inkrementierung um mehr als Eins erhöht wird.

○ Negative Steigungswinkel verursachen Probleme: Mit der Arithmetik des Zweierkomplements können zwar negative Veränderungen von X und Y dargestellt werden, doch ist ein „Zwei-Byte“-Zweierkomplement nötig, um die Werte der Y-Koordinaten von 0 bis 199 und die der X-Koordinaten von 0 bis 319 zu berechnen. ○ Vertikale Linien lassen sich nicht zeichnen, da in der Berechnung ihrer Steigungswinkel eine Division durch Null vorkommt.

Das erste Problem läßt sich leicht durch eine Vertauschung der zwei Punkte vor der Berechnung lösen.

Um bei einem Steigungswinkel kleiner als Eins eine zusammenhängende Linie erzeugen zu können, müssen wir den Wert der X-Koordinate inkrementieren und Y berechnen, für Winkel größer als Eins aber Y inkrementieren und X berechnen. Mit Hilfe der Bedingung ($G > 1$?) können zwei Lösungswege vorgesehen werden, mit denen die Division und das doppelte Zweierkomplement vermieden wird.

Das Plotten von Werten kleiner als Eins haben wir bereits beschrieben. Was aber passiert, wenn der Steigungswinkel größer als Eins wird? Definieren wir $DX = X2 - X1$ und $DY = Y2 - Y1$. Über einen Vergleich der Werte DX und DY läßt sich leicht feststellen, ob der Steigungswinkel über Eins liegt: Ist DX kleiner als DY , dann ist der Winkel größer als Eins.

Negative Werte von DX und DY lassen sich nur durch eine kompliziertere Methode ausschalten. Bei einem Steigungswinkel über Eins müssen vier mögliche Fälle berücksichtigt werden (siehe Bild). Unser Programm muß zunächst exakt bestimmen, welcher Fall eintritt, und dann den entsprechenden Ablauf aufrufen:

Fall A: DY muß mit $Y1 - Y2$ neu berechnet werden. Vom Ausgangspunkt $(X1, Y1)$ wird dann Y solange dekrementiert, bis $Y2$ erreicht ist.

Fall B: Vertauschen Sie die Anfangs- und Endpunkte und berechnen Sie diese neu.

Fall C: Fangen Sie bei $(X1, Y1)$ an und inkrementieren Sie Y bis $Y2$ erreicht ist.

Fall D: Vertauschen Sie die Anfangs- und Endpunkte und berechnen Sie diese neu.

Am Anfang des Programms werden DX und DY berechnet. Je nachdem, ob DX oder DY negativ ist, werden die Bits einer bestimmten

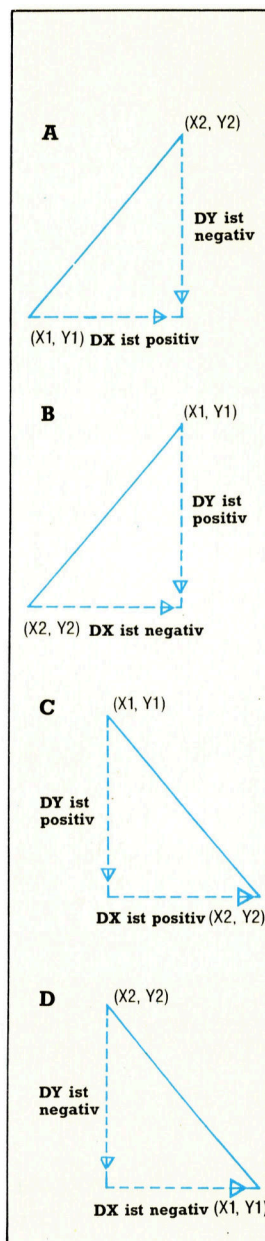
Wenn der absolute Wert des Steigungswinkels $(Y2 - Y1) / (X2 - X1)$ größer als Eins ist, sollte DX auf einen positiven Wert gebracht werden. Zum Beispiel:

A) Berechnen Sie $-(DY)$ neu und dekrementieren Sie Y beginnend bei $(X1, Y1)$, bis $Y2$ erreicht ist.

B) Vertauschen Sie $(X1, Y1)$ mit $(X2, Y2)$ und berechnen Sie neu.

C) Inkrementieren Sie Y beginnend bei $(X1, Y1)$, bis $Y2$ erreicht ist.

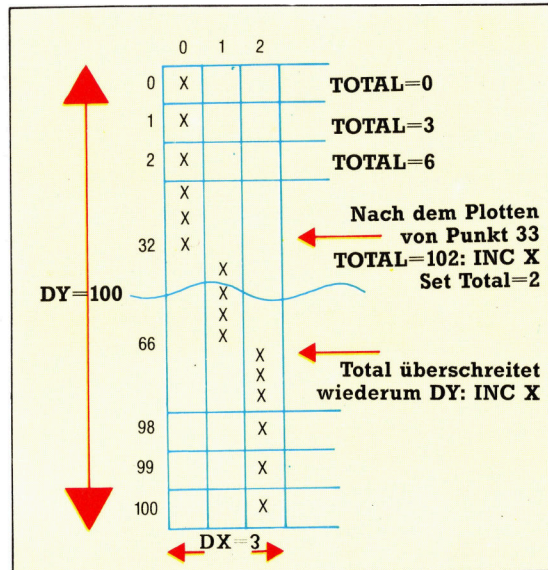
D) Vertauschen Sie $(X1, Y1)$ mit $(X2, Y2)$, und berechnen Sie neu.





Das Bild rechts zeigt, wie eine Linie abschnittsweise aus einzelnen Punkten zusammengesetzt wird. Die hier gezeigte Linie verbindet die Endpunkte (0,0) und (2,100) mit drei gleichlangen vertikalen Strichen.

Das untere Ablaufdiagramm zeigt, wie Line-sub die vier Fälle behandelt, die bei einem Steigungswinkel größer als Eins eintreten. Die Variablen TOTAL und DECFLG werden für die Plotroutine des zweiten Diagramms bereitgestellt. Das Plotten fängt bei dem Punkt (X1,Y1) an, wobei Y je nach Status von DECFLG dekrementiert oder inkrementiert wird.



Speicherstelle gesetzt. So zeigt Bit 0 des Registers NEGREG an, ob DY negativ ist; Bit 1, ob DX negativ ist. Nach der Berechnung kann mit NEGREG getestet werden, welcher der vier Fälle eingetreten ist:

Fall	a	b	c	d
Dezimalwert von NEGREG	1	2	0	3
Binärwert von NEGREG	01	10	00	11

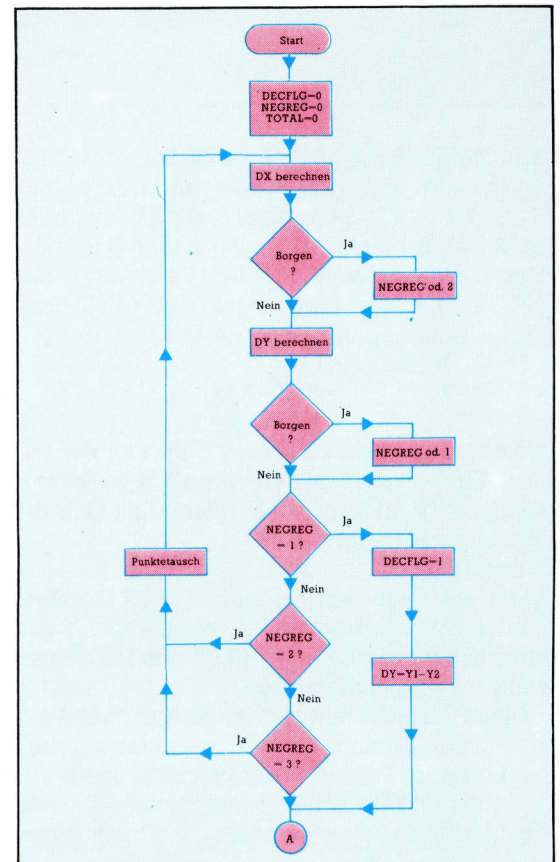
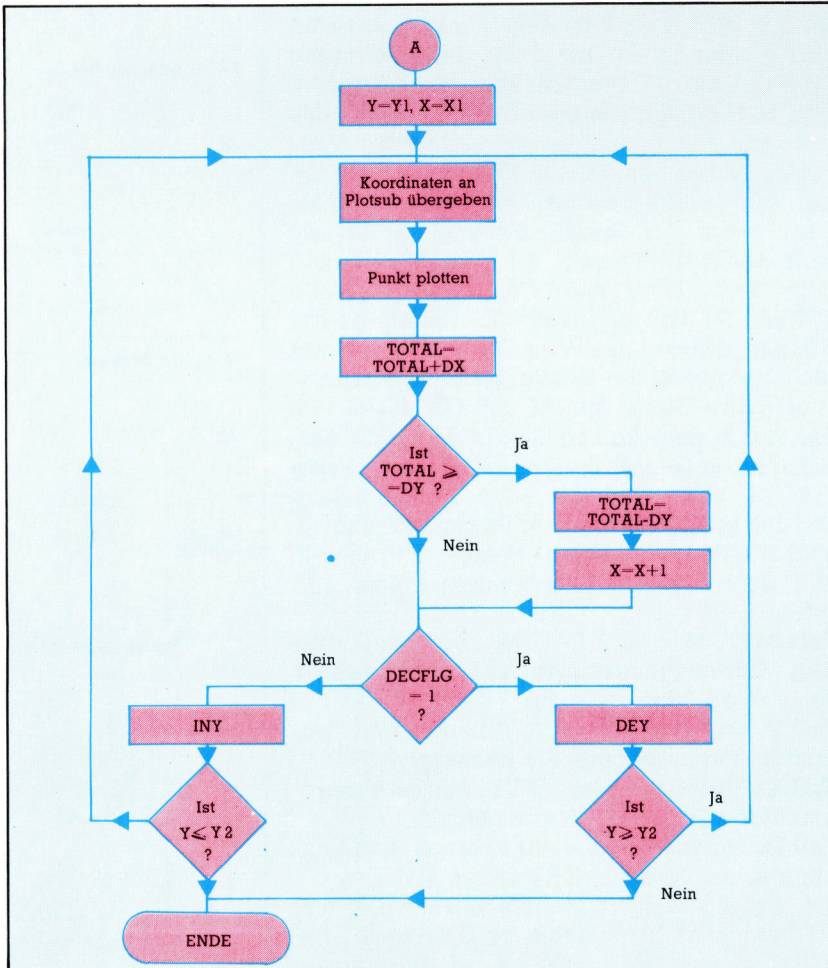
Nach dem Test plotten wir die Linie über eine Schleife, die zunächst vom Anfangspunkt (X1,Y1) den Wert von Y je nach Fall inkrementiert oder dekrementiert, dann den entsprechenden Wert von X berechnet und mit Plotsub den Punkt auf den Schirm bringt. Wenn Y den Wert von Y2 erreicht, ist die Schleife beendet.

Um X aus Y berechnen zu können, müssen wir unsere Formel nochmals umformen:

$$\begin{aligned}(Y-Y_1)/(X-X_1) &= G \\ 1/(X-X_1) &= G/(Y-Y_1) \\ (X-X_1) &= (Y-Y_1)/G \\ X &= X_1 + (Y-Y_1)/G\end{aligned}$$

Für den Fall $(Y-Y_1) = 1$ lautet die Formel daher: $X = X_1 + 1/G$. Da $G = DY/DX$ ist, wird die Formel zu $X = X_1 + DX/DY$. Wenn der Steigungswinkel jedoch größer als Eins wird, ist DY immer größer als DX, wodurch die Division DX/DY immer 0 ergibt mit dem Rest DX. In der Programmschleife addieren wir daher diese verbleibenden Reste und brauchen X nur dann zu inkrementieren, wenn die Gesamtrestsumme größer ist als DY.

Sehen wir uns diesen Ablauf an einem Beispiel genauer an, in dem $(X_1,Y_1)=(0,0)$ und $(X_2,Y_2)=(2,100)$ ist. Da die Linie einen sehr großen Steigungswinkel hat, wird sie mit drei vertikalen Strichen dargestellt. Kleinere Steigungswinkel würden aus einer größeren Anzahl kürzerer Striche bestehen. Bei dem Steigungswinkel von Eins ist jeder einzelne Strich nur ein Pixel lang und es entsteht eine perfekte Diagonale.





Einsatz der Routine unter BASIC

Um die Routine von BASIC aus einsetzen zu können, müssen zunächst Linesub und Plotsub geladen werden. Die entsprechenden Objektcodeprogramme heißen PLOTSUB.HEX und LINESUB.HEX. Nach dem Laden der Maschinencoderroutinen fragt das Beispielprogramm nach zwei Koordinaten, testet den Steigungswinkel und ruft – wenn er nicht kleiner als Eins ist – Plotsub auf, um den HIRES-Bildschirm zu löschen und die Farbe festzulegen. Die Unterroutine in Zeile 2000 wandelt dann die X-Koordinaten in das Format Lobyte – Hbyte, bevor sie per POKE in die

Beispielprogramm in BASIC

```

10 REM *****
12 REM ** LINESUB 64 **
13 REM *****
14 :
15 DN=8:REM FOR CASSETTE SET DN=1
20 IF A=0 THENA=1:LOAD"PLOTSUB.HEX",DN,1
30 IF A=1 THENA=2:LOAD"LINESUB.HEX",DN,1
50 INPUT"FIRST POINT":X1,Y1
60 INPUT"SECOND POINT":X2,Y2
70 GOSUB1000:REM SET HIRES MODE
80 GOSUB2000:REM LINESUB
90 GETA$:IFA$=""THEN 90
95 IF A$="" THEN200
100 REM **** RESET SCREEN ****
110 POKE49408,0:SYS 49422
120 GOSUB3000
125 GETA$:IFA$=""THEN125
127 GOT050
140 :
200 REM **** DRAW TRIANGLE ****
205 XA=30:YA=10:XB=310:YB=90
210 XC=90:YC=180
220 GOSUB1000
230 X1=XA:Y1=YA:X2=XB:Y2=YB:GOSUB2000
240 X1=XC:Y1=YC:GOSUB2000
250 X2=XA:Y2=YA:GOSUB2000
255 GETA$:IFA$="" THEN 255
260 REM **** RESET SCREEN ****
270 POKE49408,0:SYS 49422
275 PRINTCHR$(147)
280 END
290 :
1000 REM **** SET HIRES ****
1010 POKE49408,1:POKE49409,1
1015 POKE49410,1
1020 SYS 49422
1030 RETURN
1040 :
2000 REM **** ENTER LINESUB ****
2010 MHI=INT(X1/256):MLO=X1-256*MHI
2020 NHI=INT(X2/256):NLO=X2-256*NHI
2030 POKE49920,MLO:POKE49921,MHI
2040 POKE49922,NLO:POKE49923,NHI
2050 POKE49924,Y1:POKE49925,Y2
2060 SYS 49934
2070 RETURN
2080 :
3000 REM **** PRINT VALUES ****
3001 RESTORE
3002 PRINTCHR$(147):REM CLEAR SCREEN
3005 FORI=0T013
3010 READA$
3020 PRINTA$,PEEK(49920+I)
3030 NEXT I
3040 DATA X1LO,X1HI,X2LO,X2HI,Y1,Y2,DXLO
3050 DATA DXHI,DY,TEMP,TOTLO,TOTHI,NEGREG,DECFLG
3060 RETURN

```

Speicherstellen gesetzt werden, die Linesub für die Endpunkte der Linie reserviert hat. Die Unterroutine in Zeile 3000 zeigt die Werte aller Speicherstellen an, die mit Linesub-Variablen belegt sind.

Statt den Quellcode für Linesub einzugeben und zu assemblieren, kann das Programm auch über das Ladeprogramm für Maschinencode in den Speicher geladen werden. Dabei werden eine Reihe von DATA-Befehlen eingelesen und in den Speicher gePOKEt. Geben Sie das Ladeprogramm ein, und rufen Sie es mit RUN auf, um Linesub in den Speicher zu laden.

Ladeprogramm für Maschinencode

```

10 FOR I=49934 TO 50371
20 READA:CC=CC+A
30 POKEI,A:NEXT
50 READA:IFCC<>A THEN PRINT"CHECKSUM
ERROR":END
100 DATA72,138,72,152,72,169,0,141,13
110 DATA195,141,12,195,141,10,195,141
120 DATA11,195,173,2,195,56,237,0,195
130 DATA141,6,195,173,3,195,237,1,195
140 DATA141,7,195,16,8,173,12,195,9,2
150 DATA141,12,195,173,5,195,56,237,4
160 DATA195,141,8,195,176,8,173,12,195
170 DATA9,1,141,12,195,173,12,195,201
180 DATA1,240,20,201,2,208,6,32,140
190 DATA196,76,19,195,201,3,208,19,32
200 DATA140,196,76,19,195,173,4,195,56
210 DATA237,5,195,141,8,195,238,13,195
220 DATA173,6,195,24,105,1,141,6,195
230 DATA173,7,195,105,0,141,7,195,238
240 DATA8,195,173,4,195,168,173,7,195
250 DATA201,1,240,115,173,6,195,205,8
260 DATA195,176,107,173,0,195,141,3
270 DATA193,173,1,195,141,4,193,152
280 DATA141,5,193,32,131,193,173,10
290 DATA195,24,109,6,195,176,8,141,10
300 DATA195,205,8,195,144,24,56,237,8
310 DATA195,141,10,195,173,0,195,24
320 DATA105,1,141,0,195,173,1,195,105
330 DATA0,141,1,195,173,13,195,201,1
340 DATA208,31,136,204,5,195,240,3,76
350 DATA161,195,152,141,5,193,173,0
360 DATA195,141,3,193,173,1,195,141,4
370 DATA193,32,131,193,76,134,196,200
380 DATA204,5,195,144,152,76,134,196
390 DATA173,0,195,141,3,193,173,1,195
400 DATA141,4,193,152,141,5,193,32,131
410 DATA193,173,10,195,24,109,8,195
420 DATA141,10,195,173,11,195,105,0
430 DATA141,11,195,173,10,195,56,237,6
440 DATA195,141,10,195,173,11,195,237
450 DATA7,195,141,11,195,48,15,173,13
460 DATA195,201,1,240,4,200,76,104,196
470 DATA136,76,104,196,173,10,195,24
480 DATA109,6,195,141,10,195,173,11
490 DATA195,109,7,195,141,11,195,173,0
500 DATA195,24,105,1,141,0,195,173,1
510 DATA195,105,0,141,1,195,205,3,195
520 DATA208,142,173,0,195,205,2,195
530 DATA208,134,104,168,104,170,104,96
540 DATA173,2,195,141,9,195,173,0,195
550 DATA141,2,195,173,9,195,141,0,195
560 DATA173,3,195,141,9,195,173,1,195
570 DATA141,3,195,173,9,195,141,1,195
580 DATA173,5,195,141,9,195,173,4,195
590 DATA141,5,195,173,9,195,141,4,195
600 DATA96,230
610 DATA50794:REM*CHECKSUM*

```




Assembler-Listing

```
+++++
+++++
++      LINESUB 64  ++
++      ++++++
+++++
PLTSUB = #C183
XLO    = #C183
XHI    = #C184
YLO    = #C185
*      = #C300
+++++ LINESUB VARIABLES +++++
X1LO   ****+1
X1HI   ****+1
X2LO   ****+1
X2HI   ****+1
Y1     ****+1
Y2     ****+1
DXLO   ****+1
DXHI   ****+1
DY     ****+1
TEMP   ****+1
TOTLO  ****+1
TOTHI  ****+1
NEGREG ****+1
DECFLG ****+1
+++++ PUSH REGISTERS ONTO STACK +++++
48      PHA
49      TPA
48      PHA
48      TPA
48      PHA
+++++ INITIALISE VARIABLES +++++
START
A9 00    LDA #000
8D 00 C3 STA DECFLG
8D 0C C3 STA NEGREG
8D 0A C3 STA TOTLO
8D 0B C3 STA TOTHI
+++++ CALCULATE DX +++++
AD 02 C3 LDA X2LO
38      SEC
ED 00 C3 SBC X1LO
8D 06 C3 STA DXLO
AD 03 C3 LDA X2HI
4C 86 C4 JMP FINISH
NODEV
C8      INV
CC 05 C3 CPY V2
90 98    BCC PLOT
4C 86 C4 JMP FINISH
+++++ GRADIENTS LESS THAN ONE +++++
GLESS
AD 00 C3 LDA X1LO
8D 03 C1 STA XLO
AD 01 C3 LDA X1HI
8D 04 C1 STA XHI
90      TPA
8D 05 C1 STA YLO
20 83 C1 JSR PLTSUB
AD 0A C3 LDA TOTLO
18      CLC
ED 08 C3 ADC DY          ; TOTAL=TOTAL+DY
8D 0A C3 STA TOTLO
AD 0B C3 LDA TOTHI
63 00    ADC #000        ; CATCH CARRY
8D 0B C3 STA TOTHI
AD 0A C3 LDA TOTLO
38      SEC
ED 06 C3 SBC DXLO        ; TOTAL=TOTAL-DX
8D 0A C3 STA TOTLO
AD 0B C3 LDA TOTHI
ED 07 C3 SBC DXHI
8D 0B C3 STA TOTHI
30 0F    BHI ADD          ; NEGATIVE RESULT
AD 00 C3 LDA DECFLG
C9 01    CMP #001
F0 04    BEQ DECY
C8      INV
4C 68 C4 JMP PAST
DECY
88      DEY
4C 68 C4 JMP PAST
ADD
AD 0A C3 LDA TOTLO
18      CLC
ED 06 C3 ADC DXLO        ; ADD BACK DX
8D 0A C3 STA TOTLO
AD 0B C3 LDA TOTHI
ED 07 C3 ADC DXHI
8D 0B C3 STA TOTHI
ED 01 C3 SBC X1HI
8D 07 C3 STA DXHI
18 08    BPL NBRANDY
AD 0C C3 LDA NEGREG
09 02    ORA #002        ; FLAG DX NEGATIVE
8D 0C C3 STA NEGREG
+++++ CALCULATE DY +++++
NBRANDY
AD 05 C3 LDA Y2
38      SEC
ED 04 C3 SBC Y1
8D 08 C3 STA DY
B0 08    BCS NBRANDY
AD 0C C3 LDA NEGREG
09 01    ORA #001        ; FLAG DY NEGATIVE
8D 0C C3 STA NEGREG
NBRANDY
+++++ TEST STATE OF NEGREG +++++
AD 0C C3 LDA NEGREG
C9 01    CMP #001
F0 14    BEQ NEG1
C9 02    CMP #002
D0 06    BNE FALSE
20 8C C4 JSR SWOP
4C 13 C3 JMP START
FALSE
C9 03    CMP #003
D0 13    BNE NEG0
20 8C C4 JSR SWOP
4C 13 C3 JMP START
NEG1
AD 04 C3 LDA Y1
38      SEC
ED 05 C3 SBC Y2          ; RECALCULATE DY
8D 08 C3 STA DY
EE 0D C3 INC DECFLG; SET DECFLG
NEG0
AD 06 C3 LDA DXLO
18      CLC
ED 01    ADC #001        ; ADD 1 TO DX
8D 06 C3 STA DXLO
AD 07 C3 LDA DXHI
63 00    ADC #000        ; CATCH CARRY
8D 07 C3 STA DXHI
EE 08 C3 INC DY          ; ADD ONE TO DY
AD 04 C3 LDA Y1
A8      TAY            ; SET Y=Y1
+++++ TEST FOR DX >= DY +++++
AD 07 C3 LDA DXHI
C9 01    CMP #001
F0 73    BEQ GLESS
AD 06 C3 LDA DXLO
CD 08 C3 CMP DY
B0 6B    BCS GLESS
+++++ PLOT POINT +++++
PLOT
AD 00 C3 LDA X1LO
8D 03 C1 STA XLO          ; PASS LOBYTE OF X
AD 01 C3 LDA X1HI
8D 04 C1 STA XHI          ; PASS HIBYTE OF X
90      TPA
8D 05 C1 STA YLO          ; PASS Y
20 83 C1 JSR PLTSUB
AD 0A C3 LDA TOTLO
18      CLC
ED 06 C3 ADC DXLO        ; TOTAL=TOTAL+DX
B0 08    BCS DOINK        ; SUM EXCEEDS 1 BYTE?
8D 0A C3 STA TOTLO
CD 08 C3 CMP DY
90 18    BCC NOINK        ; TOTAL REACHED DY?
DOINK
ED 08 C3 SBC DY          ; TOTAL=TOTAL-DY
8D 0A C3 STA TOTLO
AD 0A C3 LDA X1LO
18      CLC
ED 01    ADC #001        ; INCREMENT X1
8D 00 C3 STA X1LO
AD 01 C3 LDA X1HI
8D 01 C3 STA XHI
NOINK
AD 00 C3 LDA DECFLG
C9 01    CMP #001
D0 1F    BNE NODEV        ; DECFLG=1?
88      DEY
CC 05 C3 CPY V2
F0 03    BEQ FINAL        ; IF Y=V2 PLOT LAST
4C 01 C3 JMP PLOT          ; PLOT LAST POINT
FINAL
90      TPA
8D 05 C1 STA YLO
AD 00 C3 LDA X1LO
8D 03 C1 STA XLO
AD 01 C3 LDA X1HI
8D 04 C1 STA XHI
20 83 C1 JSR PLTSUB
PAST
AD 00 C3 LDA X1LO
18      CLC
ED 01    ADC #001        ; INCREMENT X
8D 00 C3 STA X1LO
AD 01 C3 LDA X1HI
8D 01 C3 STA XHI
63 00    ADC #000        ; CATCH CARRY
8D 01 C3 STA X1HI
CD 03 C3 CMP X2HI
D0 0E    BNE GLESS        ; HAS X REACHED X2?
AD 00 C3 LDA X1LO
CD 02 C3 CMP X2LO
D0 06    BNE GLESS
+++++ PULL REGISTERS OFF STACK +++++
FINISH
68      PLA
A8      TPA
68      PLA
68      PLA
60      RTS
+++++ END OF PROGRAM +++++
+++++ SWOP POINTS SUBROUTINE +++++
SWOP
AD 02 C3 LDA X2LO
8D 09 C3 STA TEMP
AD 00 C3 LDA X1LO
8D 02 C3 STA X2LO
AD 09 C3 LDA TEMP
8D 00 C3 STA X1LO
AD 03 C3 LDA X2HI
8D 09 C3 STA TEMP
AD 01 C3 LDA X1HI
8D 03 C3 STA X2HI
AD 09 C3 LDA TEMP
8D 01 C3 STA X1HI
AD 05 C3 LDA Y2
8D 09 C3 STA TEMP
AD 04 C3 LDA Y1
8D 05 C3 STA Y2
AD 09 C3 LDA TEMP
8D 04 C3 STA Y1
60      RTS
+++++ END OF SUBROUTINE +++++
```

Die Maschinen-coderoutine Linesub

Die Maschinencoderoutine zeichnet Linien mit Steigungswinkeln größer, kleiner oder gleich Eins. Linien mit einem Steigungswinkel unter Eins lassen sich mit der gleichen Methode plotten. Dabei muß nur getestet werden, ob DX größer als DY ist (d. h. ein Steigungswinkel kleiner als Eins) und entsprechend verzweigt werden.

● Für jede X-Koordinate verwendet die Routine zwei Bytes, da X Werte über 255 annehmen kann. Für Y genügt ein einziges Byte, da deren Werte nicht größer als maximal 199 werden.

● Die Speicherstellen, in denen Plotsub seine Koordinaten speichert, werden am Anfang des Assemblerlistings zugeordnet. Dort befindet sich auch die Anfangsadresse von Plotsub.

● Bit 1 und 0 von NEGREG werden je nach Ergebnis der Berechnung von DX und DY gesetzt. Damit sich durch das Ansprechen eines Bits die anderen Bits nicht verändern, wird hier der Logikbefehl ORA verwendet.

● Da die Y-Koordinate in einem einzigen Byte gespeichert werden kann, wird dieser Wert während des gesamten Plotvorganges im Y-Register gehalten, das von Plotsub nicht angesprochen wird. Um Plotsub zusammen mit Linesub verwenden zu können, sollte im Quelltext von Plotsub folgende Änderung ausgeführt werden. Zwischen SBC REMX und TAX muß ein Verzweigungsbefehl eingesetzt werden. Die entsprechende Veränderung kann auch über das Beispielprogramm in BASIC vorgenommen werden, wenn dort die folgenden Zeilen eingefügt werden:

```
5010 FOR I=HRSFLG TO
5380 DATA 169,7,237,8,
193,240,6,170,14,13,193
5410 DATA 38944:
REM*CHECKSUM*
```


Fachwörter von A bis Z

Electrostatic Printer = Elektrostatischer Drucker

Die Bezeichnung „elektrostatisch“ wird oft gleichbedeutend mit „elektrosensitiv“ verwendet. Tatsächlich arbeiten nur die sogenannten Laserdrucker, die bei Großrechnern verwendet werden, nach dem elektrostatischen Verfahren. Wie bei den Fotokopierern wird bei diesen Geräten zunächst eine Trommel, die mit einem Fotohalbleiter beschichtet ist, elektrostatisch homogen aufgeladen. Dann wird ein Laserstrahl dem Schriftbild entsprechend über die Trommel geführt, wobei sich die Halbleiterschicht an den belichteten Stellen durch Freisetzung von Fotoelektronen entgegengesetzt auflädt. Dort setzen sich anschließend beim Vorbeilaufen der Trommel an der „Tonerstation“ feine Farbpartikel fest, die auf Papier übertragen und eingebrannt werden.

Emulator = Emulator

Mit einem Emulator wird auf einem Computer das Verhalten eines Fremdrechners simuliert. Programme für Microcomputer werden oft auf Großrechnern entwickelt, weil dort Speicherkapazität und Testkomfort wesentlich umfangreicher sind. Um beispielsweise Software für den Sinclair Spectrum zu schreiben, braucht man einen Spectrum-Emulator für den Großrechner, der die Eigenschaften des Spectrum simuliert.

ENIAC = ENIAC

ENIAC steht für Electronic Numerical Integrator And Computer – eine universelle Rechanlage, die in den Jahren 1943–46 von John J. Mauchly und J. Presper Eckert jr. an der University of Pennsylvania/USA entwickelt wurde und mit Elektronenröhren als Schaltelementen arbeitete. Der ENIAC war ursprünglich für die Berechnung ballistischer Tabellen im Zweiten Weltkrieg gedacht, wurde aber nicht mehr vor Kriegsende fertig. Dieser Computer füllte einen Riesenraum, der wegen der enormen Wärmeabgabe des Rechners mit einer speziellen Klimaanlage ausgestattet war.

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.

EOF = Ende der Datei

An dem Steuerzeichen „End Of File“ am Ende einer Datei erkennt das Betriebssystem, daß in diesem File keine weiteren Daten zu verarbeiten sind.

Ergonomics = Ergonomie

Die Ergonomie ist jener Teil der Arbeitswissenschaft, der sich mit der Anpassung von Geräten und Arbeitsplätzen an die physischen und psychischen Bedürfnisse des Menschen befaßt. Die Physiologie spielt bei der äußeren Gestaltung eine wichtige Rolle – ergonomische Gesichtspunkte haben zur Entwicklung frei beweglicher Tastaturen, schwenkbarer Bildschirme, hochauf-

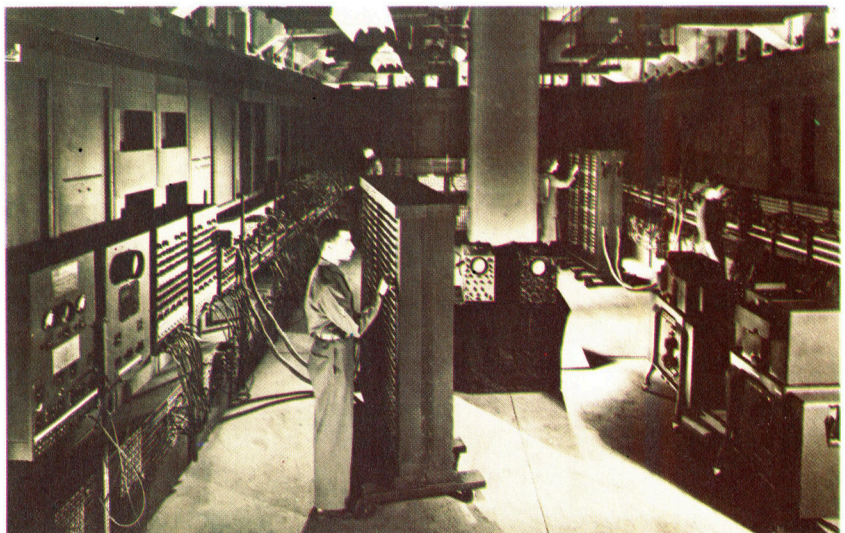
lösender Monitore usw. geführt. Die Psychologie verhilft etwa zu einem Bewußtsein für „Benutzerfreundlichkeit“, das heißt zu einer Gestaltung von Hardware und Software, mit der der Benutzer in allen Situationen problemlos zurechtkommt.

Exclusive-OR = Exklusiv-ODER

Die Exklusiv-OR (XOR)-Operation ist eine elementare Funktion der Booleschen Algebra. Sie verknüpft zwei Ein-Bit-Eingänge A und B zu einer Ausgangsgröße C. Deren Wert ist 1, wenn genau ein Eingang die 1 liefert und 0, wenn beide Eingänge zugleich auf 0 oder 1 liegen. Ein XOR-Gatter läßt sich aus zwei AND-Gattern, zwei Invertern und einem OR-Gatter aufbauen. Das XOR gibt es bei vielen Mikroprozessoren auch als Einzeloperand-Befehl.

Bildnachweise

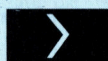
1065, 1067, 1083, 1085: Chris Stevens
1067: Rolf Seiffe
1069, 1070, 1072, 1075, 1076: Ian McKinnell
1075: Liz Dixon
1077: Tony Sleep
1079: Marcus Wilson-Smith
1082: Dimension Graphics
1086: Liz Heaney
1088: Tony Lodge
1088: „Your 64 and VIC-20“
U3: Science Museum



Dieses Ungetüm mit über 17 000 Röhren und einem Gewicht von 30 Tonnen war die Vorstufe der elektronischen Datenverarbeitung.

computer kurs

Heft 40



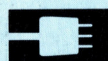
Mondlandung

Auch mit BASIC lassen sich spannende Spiele programmieren, zum Beispiel „Lunar Lander“.



Datenbanken

Im nächsten Heft zeigen wir, welche Datenbanken auf Heimcomputern verwendet werden und wie man sie am besten nutzt.



Erweiterung

Das Interface Plus 1 verwandelt den Acorn Electron in eine Alternative zum Modell B.



Programmiertechnik

Die meisten Leute lernen das Programmieren aus den Handbüchern, die mitgeliefert werden. Wir geben Tips für besseres Arbeiten.



Apricot Portable

Ein wirklich tragbarer PC mit einem speziellen Eingabegerät (Maus/Trackball) und einem Spracherkennungssystem.

