

Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Heft **75**

Ein wöchentliches Sammelwerk



Computer und Beruf

Research Machines 380Z

Drum-Computer C 64

Gehobene Textverarbeitung

computer kurs

Heft 75

Inhalt

Computer Welt

Jobs für Hacker? 2073

Aufstiegschancen im Computer-Business

Schnelle Schritte 2083

Einstieg in die rasante Computertechnologie

Bits und Bytes

Erhöhte Befehlsgewalt 2076

Selbstdefinierte Basic-Befehle für den Spectrum

Trommelwirbel 2092

Der C 64 wird zum Drum-Computer

Tips für die Praxis

Punkt für Punkt 2078

Wirewrap-Technik für das Digitalvoltmeter

Unter Spannung 2098

Der Aufbau der LED-Anzeige

Software

Spitzenreiter 2081

Eine Textverarbeitung für gehobene Ansprüche

Wurzelbehandlung 2096

MS-DOS-Inhaltsverzeichnisse und Dienstmodule

Hardware

380Z, der Robuste 2086

Spitzenreiter in Sachen Haltbarkeit

Sprache

In der Bibliothek 2088

Kooperation zwischen C und Unix

BASIC 75

Letzte Feinheiten 2090

Strategie und Taktik für Go-Experten

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.

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

SAMMELORDNER

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

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

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

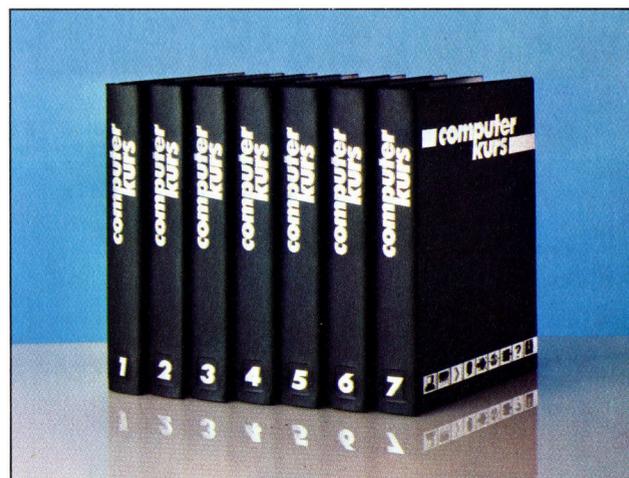
Schweiz: Der Sammelordner kostet sfr 15. Bitte wenden Sie sich an Ihren Kiosk; dort werden Sie jederzeit die gewünschten Exemplare erhalten.

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), Peter Aldick, Holger Neuhaus, Uta Brandl (Layout), Sammelwerk Redaktions-Service GmbH, Paulstraße 3, 2000 Hamburg 1

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



© APSIF, Copenhagen, 1982, 1983; © Orbis Publishing Ltd., 1982, 1983; © Marshall Cavendish Ltd., 1984, 1985, 1986; **Druck:** E. Schwend GmbH, Schmollerstraße 31, 7170 Schwäbisch Hall



Jobs für Hacker?

Ihre Computer-Kenntnisse bringen Sie vielleicht auf die Idee, auch Ihren Broterwerb im Computer-Bereich zu suchen. Die Rechner-Industrie bietet mehr Stellen an als andere Berufszweige.

An erster Stelle bei einer Entscheidung für das Computer-Business muß immer die Frage stehen, welchen Teilbereich aus diesem komplexen Arbeitsfeld Sie wählen möchten: Superjobs liegen auch hier nicht auf der Straße. Karrieremöglichkeiten gibt es zwar, die einzelnen Laufbahnen unterscheiden sich aber sehr stark. Der eine Beruf bietet mehr Gehalt, der zweite interessante Technologieerfahrungen, ein dritter vielleicht gute Zukunftschancen oder ein besonders fruchtbares Arbeitsklima...

Bedenken Sie, daß eine Computer-Karriere auch zur Einbahnstraße werden kann. Der Einstieg legt Ihre künftigen Entwicklungsmöglichkeiten fest, besonders dann, wenn der Arbeitgeber in Ihre Ausbildung investiert oder Sie ein ganz spezielles Fachgebiet auswählen.

Die Arbeit mit Computern ist außerordentlich vielschichtig, aber auch hier kommen die üblichen Abstufungen vor. Jeder Job ist mit unterschiedlichen Anforderungen verbunden, wird unterschiedlich bezahlt und hat unterschiedlich gute Karrierechancen. In der Produktion von elektronischen Geräten oder Computern sind verschiedene Kategorien von Angestellten mit der Hardware beschäftigt: Fließbandarbeit gibt es hier genauso wie Tätigkeiten als Produktionsingenieur, Entwickler oder gar als Forscher.

Fachmann gesucht

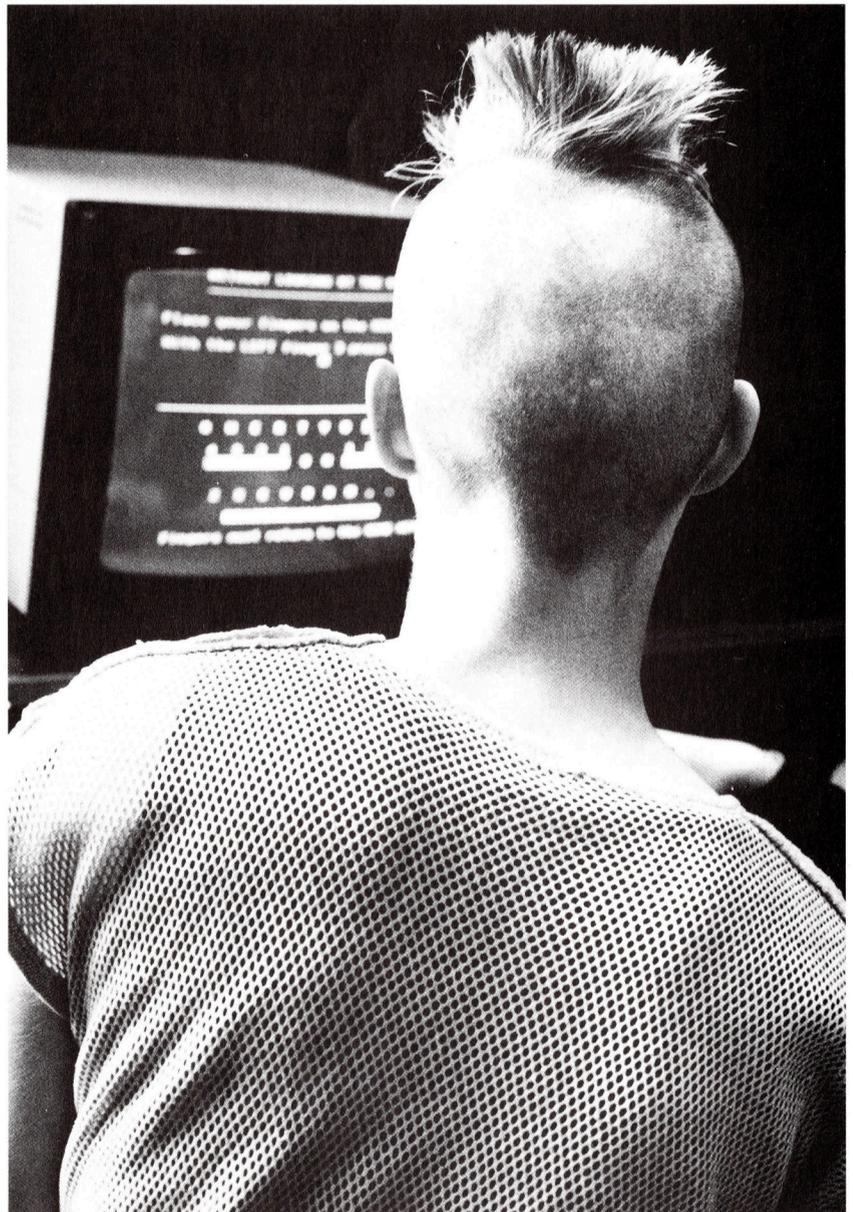
Systementwicklung – ob Microprozessor oder Großrechner – leistet nur der erfahrene oder überaus begabte Fachmann, wobei die Ingenieurlaufbahn auch mit eher durchschnittlichem Talent gewählt werden darf. Erfolgreiche Firmen stellen zunehmend mehr Personal für die Entwicklung neuer Produktgenerationen ein, um Anschluß zu halten.

Jobs in der Entwicklung werden ausschließlich an Studienabsolventen vergeben, die gute Bezahlung, Sicherheit und Prestige erwarten können. Akademiker überwiegen gegenüber Nichtakademikern im industriellen Bereich zahlenmäßig um ein Vielfaches. Als Studienfach empfehlen sich für dieses Gebiet die Elektrotechnik oder das Informatikstudium an einer Fachhochschule oder Universität. Die drei- bis fünfjährige Ausbildung vermittelt die theoretischen Grundlagen und fortgeschrittene praktische Anwendung.

Als Alternative dazu gibt es auch noch den Berufsweg des staatlich geprüften Technikers, der im allgemeinen als Weiterbildung nach einer Lehre eingeschlagen wird. Gute Schulzeugnisse allein genügen selten. Im allgemeinen wird erwartet, daß ein neuer Mitarbeiter Schaltungen verstehen kann und die Grundprinzipien eines Microprozessors kennt.

Auch den findigsten Programmierer oder Hobby-Elektroniker wird man nicht gleich zu Anfang an den Entwurf von PCs oder Chips

Im Computer-Bereich finden sich noch offene Stellen. Heimcomputer-Erfahrung reicht hier jedoch selten aus. Vorsicht ist bei „Schmalspur“-Ausbildungen geboten – gleichgültig, ob sie von einer Schule oder als Pseudo-Ausbildung in einem Betrieb offeriert werden.





Computer-Einzelhandel

Eine große Zahl jüngerer Leute drängt in eine Tätigkeit als Verkäufer im Computer-Fachhandel. Der Verkauf über den Ladentisch an den Endverbraucher kann schnell seinen Glanz verlieren. Oft ist die Bezahlung schlecht, die Umsätze sind unregelmäßig. Es ist fraglich, ob der Markt für den kleinen Händler auf die Dauer tragfähig bleibt. Allerdings kommt es auch vor, daß sich Mini-Firmen zu leistungsfähigen Handelsunternehmen entwickeln.

Für den Neuling bietet der Verkauf die Möglichkeit, viel über PCs zu erfahren. Gründliche Einarbeitung in verschiedene Systeme und Programmieretechniken ist aber die Ausnahme – dafür fehlt fast immer die Zeit. Trotzdem kann der Verkauf als Vorbereitung für die Bewerbung bei größeren Firmen genutzt werden.

setzen. Davor steht das Training anderer wichtiger Fähigkeiten. Ein Entwicklungsprojekt wird normalerweise auf ein Team von Dutzenden von Angestellten verteilt.

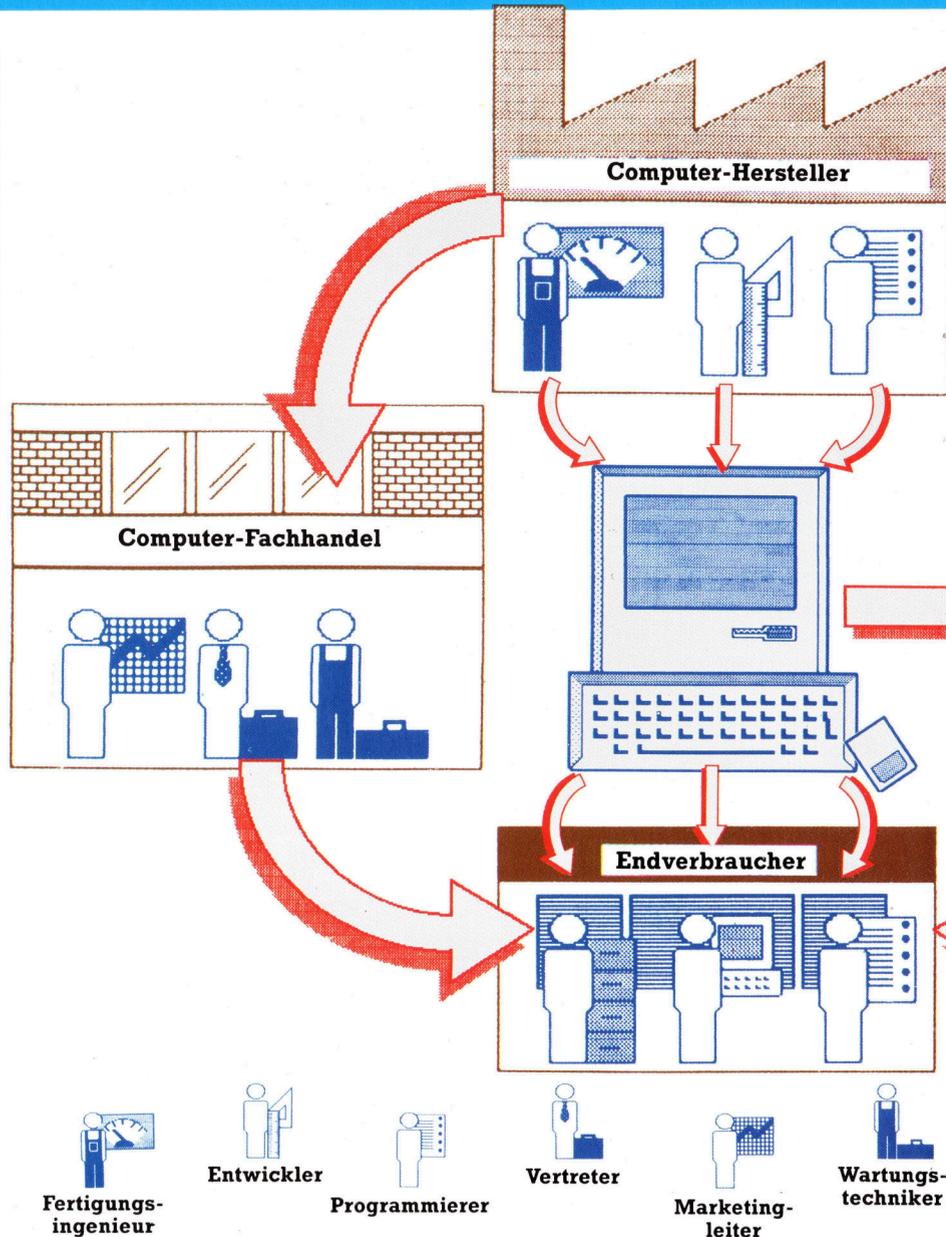
Man neigt dazu, gerade im Computer-Business und in der Elektrotechnik der Fähigkeit zum wissenschaftlichen Arbeiten einen hohen Stellenwert einzuräumen. Die schnellen Fortschritte dieses Arbeitsbereiches machen Fortbildung und ein analytisches Vorgehen bei der Problemlösung unabdingbar. Leider hält sich bis heute die Meinung, daß Frauen sich schnellem Umdenken verschließen. Der prozentuale Anteil weiblicher Mitarbeiter in der Elektronik ist nach wie vor verschwindend gering, beginnt allerdings langsam zu wachsen.

Auch heute noch kann es gelegentlich vorkommen, daß Begeisterung eine formale Ausbildung ersetzt. Findiges Wartungspersonal wird immer gebraucht, aber die Aufstiegsmöglichkeiten und auch die Bezahlung sind wenig reizvoll. Offene Stellen finden sich meist nur bei kleineren Microcomputer-Firmen.

Die meisten Arbeitsplätze gibt es in der Software-Entwicklung und in der Wartung. Programmierer stellen fast die Hälfte aller Angestellten im Computerbereich. Ein weiteres Drittel entfällt auf den Vertrieb.

Die meisten Programmierer arbeiten noch in der reinen Datenverarbeitung (DV), der mehr und mehr dem Bereich „Management Information Service“ (MIS) zur Konkurrenz wird. Viele Programmierer sind auch für Computerhersteller, Softwarehäuser oder Computer-Dienstleistungsunternehmen tätig.

Eine Tätigkeit in der Datenverarbeitung ist sehr oft der erste Schritt zum Geschäft mit dem Computer. Allerdings nimmt die Wichtigkeit dieses Bereiches ab, weil Computer immer leichter zu bedienen sind und zunehmend selbsttätig arbeiten können. Eine Stellung als Operator, Programmierer oder System-Analytiker kann jedoch Sprungbrett sein.



Die Programmierung ist heute ein hochspezialisiertes Geschäft. Ein Fachmann in den Sprachen FORTH oder PASCAL wird nicht sofort auf die Feinheiten von COBOL oder FORTRAN umschalten können. Zusätzlich verkompliziert sich die Situation durch das Übergewicht von PASCAL und COBOL im kommerziellen Bereich gegenüber FORTH und FORTRAN

Die Veränderungen in der Gehaltsstruktur sind im Computer-Bereich immer wieder für eine Überraschung gut: Mehr als in anderen Industriezweigen schlagen hier Markterschiebungen sofort auf das Gehaltsniveau durch. Die nebenstehende Statistik stammt aus Großbritannien und zeigt den durchschnittlichen Gehaltszuwachs für zwei ausgewählte Monate. Die Statistik ist allerdings nicht inflationsbereinigt – sie zeigt, daß auch im aufstrebenden Computer-geschäft bestimmte Berufsgruppen, etwa die Programmierer, die Möglichkeit eines Reallohnabbaus einkalkulieren müssen.

Sprünge beim Gehalt

Tätigkeit	September	März
Computer-Manager	6,6	8,9
Leiter DV	9,6	8,6
Leit. System-Analytiker	9,0	10,5
Leitender Analytiker	7,3	7,9
Leitender Programmierer	4,8	8,6
Programmierer (Analytik.)	6,5	7,8
Programmierer	5,7	1,6
Alle Beschäftigten	6,6	7,6



bei wissenschaftlich/technischer Anwendung.

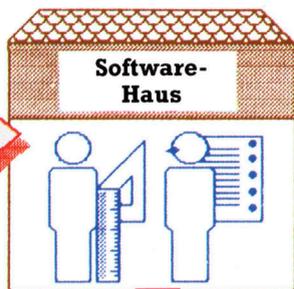
Ein nicht unerheblicher Teil – etwa ein Drittel – aller im Computerbereich Tätigen arbeitet für Herstellerfirmen. Außer in der Produktion selbst, der Geräteentwicklung und Wartung beschäftigen Computerfirmen eine große Zahl von Vertriebsleuten. Der Vertrieb wird entweder vom Hersteller selbst, von einer unabhängigen Vertriebsfirma oder vom Fachhandel übernommen; gelegentlich beteiligen sich auch Softwarehäuser daran.

Wie in anderen Industriezweigen werden auch Computer-Vertriebsleute in drei Kategorien eingeteilt. Zum einen gibt es Direktvertrieber, obwohl der „freie“ Verkauf in diesem Bereich zu den Ausnahmen zählt. Im allgemeinen wird dem Vertriebsmann ein gewisser Kundenstamm zugewiesen.

Eine zweite Kategorie sind die Marketing-Fachleute. Sie sollen (speziell im Großrechner-Bereich) die Beziehungen zu Anwendern pflegen, die sich bereits für ein System entschieden haben. Gelegentlich werden auch Neukunden mitbetreut. Großfirmen wie etwa IBM bilden ihre Verkäufer als Gesprächspartner und Helfer der DV-Leiter in den Kundenfirmen aus. In kleineren Firmen hat der Vertriebsdirektor die Aufgabe, dafür zu sorgen, daß Fach- und Großhändler die Produkte der eigenen Firmen bestellen und verfügbar halten.

Die dritte Vertriebsgruppe ist für das Zusammenwirken aller Komponenten – Hardware, Software und System – zuständig. In dieser Position findet man viele ehemalige Programmierer oder Ingenieure.

Die Wahl einer geeigneten Ausbildung ist nicht ganz einfach. Bevor Sie mehr oder weniger zufällig in irgendeinen Kursus hineinstolpern, müssen Sie sich unbedingt genau infor-



System-analytiker



Operator

Zwar ist die Hardware-Produktion immer noch das Herz der Computer-Industrie, bei der Suche nach einer interessanten Beschäftigung gewinnen die Spezialgebiete aber zunehmend an Wichtigkeit. Marketing und Vertrieb haben große Zuwachsraten, was die Beschäftigtenzahl angeht. Die Zeichnung stellt die Vernetzung zwischen Herstellern, Händlern, Drittanbietern und dem Endverbraucher dar.

Wartungsingenieure

Die Arbeit des Wartetechnikers hat noch Ähnlichkeit mit handwerklicher Tätigkeit – die meiste Zeit wird tatsächlich noch „im“ Computer verbracht, also bei der Fehlersuche. Dazu muß der Servicetechniker Kenntnisse sowohl aus der Elektronik und Elektrik haben als auch die Grundzüge des Programmierens kennen.

Bezahlung und Arbeitsbedingungen sind meist gut – dagegen steht allerdings auch, daß Techniker viel herumreisen müssen und nicht selten 24 Stunden täglich auf Abruf bereitstehen sollen. Der Angestellte in der Produktion hat es in dieser Beziehung besser.

Mit der gesteigerten Zuverlässigkeit moderner Rechner verliert der Wartetechnik zunehmend seine Arbeitsgrundlage – Selbstdiagnosesysteme u. ä. übernehmen die Fehlersuche. In naher Zukunft ist damit zu rechnen, daß der Wartungsfachmann nur noch Ersatzteile ausliefert und sie dort einsetzt, wo der Computer im Selbsttest einen Fehler angezeigt hat.

Computerspiel-Autoren

Wenn es sie je gegeben hat, dann ist spätestens heute die Zeit vorbei, in der Teenager mit guten Ideen durch einen Renner auf dem Computerspiel-Markt über Nacht reich werden konnten.

Die meisten Spiele werden von einer recht großen Schar von Freiberuflern geschrieben, die sich ihre Position gleich beim Auftauchen der ersten preiswerten und leistungsfähigen Heimcomputer erkämpft haben.

Nur wenige Programmierer arbeiten allein. Die meisten haben Verträge mit kleinen Software-Firmen oder verkaufen ihre Autorenrechte an Unternehmen, die den Vertrieb für sie übernehmen.

Eine Laufbahn als Freiberufler hat ihre Probleme. Um hinreichend zu verdienen, muß täglich gearbeitet werden – ob man nun gerade Lust dazu hat oder nicht. Freiberufliche Programmierer haben nicht selten einen normalen Beruf und schaffen ihre Produkte in der zweiten Schicht – nach Feierabend. Zudem gab es in den letzten Jahren einige Zusammenbrüche gerade bei kleinen Software-Häusern. Bei der Auswahl des Produktes ist äußerste Vorsicht geboten – Programme für einen Rechner, der nur in geringen Stückzahlen abgesetzt wurde, bringen nichts ein. Schließlich ist das Programmieren von Spielen auch keine echte Laufbahn – man kann es nicht im klassischen Sinne lernen.

mieren. In den letzten Jahren schießen – auch als Folge der hohen Arbeitslosigkeit – Computerschulen wie Pilze aus dem Boden. Aber es ist Vorsicht angebracht: Nicht jede Kurzausbildung ist seriös und vermittelt wirklich fundiertes Wissen.

Eine weitere Möglichkeit der Ausbildung neben dem Beruf wird von Fachhochschulen angeboten, etwa das Studium der Elektrotechnik oder Informatik in Abendkursen. Man darf die Doppelbelastung aber nicht unterschätzen. Besonders Kandidaten mit Familie werden hierbei auf eine harte Probe gestellt. Das erworbene Diplom ist allerdings dem normalen Fachhochschuldiplom gleichwertig und deutet auf ein großes Standvermögen hin.

Die Ausbildung an Universitäten und Fachhochschulen wird häufig auf der Industrieseite mit Mißtrauen betrachtet. Den staatlichen Einrichtungen sagt man mangelnde Flexibilität nach. Viele Lehrinhalte seien bereits überholt, das Computerzeitalter an den Bildungseinrichtungen vorbeigegangen. Auch das ist zu berücksichtigen: Es gibt gerade an den Fachhochschulen und Universitäten gravierende Unterschiede in der Qualität und Aktualität. Aber auch überbewerten darf man diesen Faktor nicht. Die Fähigkeit zum wissenschaftlichen Arbeiten, zum Umgang mit Fachliteratur und ähnliches sind vom konkret vermittelten Stoff unabhängig und können in der Praxis die wertvollere Hilfe sein.



Erhöhte Befehlsgewalt

Mit dem Interface 1 können Sie eigene Befehle in das BASIC des Spectrum einbauen. Diese Möglichkeit ist außerordentlich nützlich, da das Interface 1 die Microdrives und andere Geräte nur mit neuen Befehlen steuern kann. Wir untersuchen die Einbautechnik.

Um neue BASIC-Befehle einsetzen zu können, müssen wir zunächst einmal die Fehlerbehandlungsroutinen des Spectrum „täuschen“, damit sie die neu definierten Befehle nicht zurückweisen. Wir beschäftigen uns daher zuerst ausführlich mit den Fehlerrou-tinen des Sinclair Spectrum.

Sofort nach einem Druck auf die ENTER-Taste wird jede neu eingegebene BASIC-Zeile vom BASIC-Interpreter auf Fehler überprüft. Diese Prüfung fängt fehlende Variablen, falsch eingesetzte Parameter etc. ab, damit der Spectrum nicht von groben Eingabefehlern „belästigt“ wird. Zeilen ohne offensichtliche Fehler werden entweder an die angegebene Programmposition gesetzt oder, sofern es sich um direkte Befehle handelt, sofort ausgeführt.

Enthält die Zeile Fehler, wird der Befehl RST#8 ausgeführt, gefolgt von DEFB, der dem Betriebssystem die Art des Fehlers anzeigt. Das OS zeigt nun das blinkende Fragezeichen.

Nach dem ersten Test wird die Syntax der Zeile nochmals überprüft. Dieser zweite Test bewertet beispielsweise die angegebenen numerischen Ausdrücke und ruft dann die entsprechenden ROM-Routinen auf. In diesem Stadium werden Fehler wie „Variable existiert nicht“ entdeckt.

Der erste Test (auch Syntaxprüfung genannt) stellt sicher, daß alle Anweisungen richtig geschrieben wurden, führt den Befehl aber nicht aus. Erst der zweite Test – Ablaufprüfung genannt – aktiviert den Befehl. Mit dem Einbau des Interface 1 (IF1) ändert sich dieser Ablauf jedoch. Wir hatten bereits gesehen,

daß das IF1 beim Ansprechen der Adressen #08 oder #1708 das Haupt-ROM ausblendet und statt dessen das Schatten-ROM aktiviert. Der Spectrum arbeitet nun solange mit dem Schatten-ROM, bis das Programm die Adresse #700 des IF1 anspricht und damit das Haupt-ROM wieder anschaltet.

Ein Fehler aktiviert also sofort das Schatten-ROM, das eine weitere Fehlerprüfung durchführt. Wenn der Fehler durch einen der legalen Interface-1-Befehle wie CAT oder FORMAT verursacht wurde, führt das ROM des Interface 1 die entsprechenden Routinen aus und übergibt die Steuerung ans Haupt-ROM.

Wenn die Fehlerbedingung jedoch nicht von einem der Interface-1-Befehle verursacht wurde, verzweigt die Routine zu der in der Systemvariablen VECTOR (im IF1-ROM bei 23735 und 23736) gespeicherten Adresse. VECTOR enthält die Anfangsadresse der Fehlerbearbeitungsroutine des Schatten-ROM. Beachten Sie, daß diese Adresse bei den unterschiedlichen ROM-Versionen des IF1 an einer anderen Position liegt. In der nächsten Folge gehen wir auf die wichtigsten Unterschiede dieser Versionen genauer ein und erklären, wie man die Versionsnummer feststellt.

Die erste ROM-Version des Interface 1 arbeitet mit VECTOR = #1F0. Dabei führt ein Syntaxfehler direkt zurück zum Editor. Wenn Sie nun die in VECTOR gespeicherte Adresse ändern, können Sie die Fehlerbehandlung auf Ihre eigene Routine leiten.

#0010

Dieses Modul ist die einzige Möglichkeit, vom Schatten-ROM aus Routinen des Haupt-ROMs aufzurufen. Das Modul führt das An- und Abschalten des Haupt-ROMs automatisch aus. Es wird folgendermaßen eingesetzt:

```
RST #0010
DEFW adresse ;Adresse der aufgerufenen Routine
```

#0020

Diese Routine des Schatten-ROM entspricht dem RST #8 des Haupt-ROM. RST #0020, ge-

Schattenspiele

Das Schatten-ROM des Interface 1 besitzt außer der Steuerung der Microdrives, der RS232-Schnittstelle und der Netzwerkeigenschaften auch eine Reihe nützlicher Routinen, darunter ein Dezimal-Hexadezimalwandler.

Das Buch „Shadow ROM Disassembly“ von Melbourne House (ISBN 0-86161-191-8) enthält die ROM-Routinen in allen Einzelheiten. Zusätzlich zeigt es einige praktische Dienstprogramme.





folgt von einem Byte mit dem passenden Fehlercode, aktiviert das Modul. Unsere Tabelle zeigt die möglichen Fehlermeldungen.

#0028

Wird zur Erzeugung einer „normalen“ Fehlermeldung über das Haupt-ROM, nicht mit dem Schatten-ROM, eingesetzt. Vor Aufruf dieser Routine muß die Adresse 23610 mit dem entsprechenden Fehlercode geladen werden.

#01F0

Stellt während der Syntaxprüfung einen blinkenden Cursor dar.

#05B7

Der Aufruf dieser Routine zeigt an, daß die Syntaxprüfung abgeschlossen ist.

#05C1

Der Befehl beendet die Befehlsausführung des Schatten-ROM und gibt die Steuerung an das Haupt-ROM zurück.

#0700

Veranlaßt die Rückkehr zum Haupt-ROM. Es ist eine Möglichkeit, das Schatten-ROM zu verlassen und das Haupt-ROM zu aktivieren. Zum Schluß noch den entsprechenden Hakencode:

● **Hakencode 50**

sollte – wie alle anderen Hakencodes – nicht in Verbindung mit dem Schatten-ROM eingesetzt werden. Der Code ermöglicht, vom Haupt-ROM aus Routinen des Schatten-ROM anzusprechen, und sollte nur eingesetzt werden, wenn die Adressen des Schatten-ROM genau bekannt sind. Beim Aufruf des Moduls muß die Adresse der gewünschten Schatten-ROM-Routine im Register HL gespeichert sein.

```
LD HK,adresse; Adresse der Routine
LD (23789),HL
RST #8
DEFB 50
```

Bei einem aktiven Schatten-ROM sind auch die „Restart“-Befehle des Z80 umgestellt. Außerdem wird die Tastatur nicht mehr abgefragt und die Systemvariable FRAMES nicht inkrementiert. Eine längere Aktivierung des Schatten-ROM stört daher auch die Zeitmessung, die diese Variable steuert.

Gelegentlich gibt es auch Probleme beim Einsatz der Fließkommaberechnung vom Schatten-ROM aus. Da die Routine bei #0010 Schwierigkeiten mit dem Aufruf dieses Moduls zu haben scheint, ist es besser, dazu das Schatten-ROM zu verlassen und mit dem Haupt-ROM zu arbeiten. Das Schatten-ROM können Sie mit dem Hakencode 50 aufrufen.

Byte	Fehlermeldung	Byte	Fehlermeldung
00	Ungültiger BASIC-Befehl	11	Strom bereits eröffnet
01	Ungültige Stromnummer	12	Schreiben in Lesedatei
02	Ungültige Gerätebezeichnung	13	Lesen aus Schreibdatei
03	Ungültiger Name	14	Laufwerk schreibgeschützt
04	Ungültige Laufwerksnummer	15	Microdrive voll
05	Ungültige Stationsnummer	16	Microdrive nicht angeschlossen
06	Fehlender Name	17	Datei nicht gefunden
07	Fehlende Stationsnummer	18	Fehler im Hakencode
08	Fehlende Laufwerksnummer	19	CODE-Fehler
09	Fehlende Baudrate	20	MERGE-Fehler
10	Kopfsatz stimmt nicht überein	21	Verifikation nicht erfolgreich
		22	Falscher Dateityp
		255	Programm beendet

Einblick mit PEEK

Mit diesem Programm können Sie die Routinen des Schatten-ROM selbst untersuchen. Aus der Disassemblierung fremder Programme kann man einiges lernen – unter anderem den optimalen Einsatz der untersuchten Routine. Leider können Sie das Schatten-ROM nicht direkt mit PEEK ansprechen. Das folgende kurze BASIC-Programm lädt daher einen beliebigen Abschnitt des Schatten-ROM in den Arbeitsspeicher, wo Sie es bequem untersuchen können.

```
10 GO SUB 200
20 INPUT "How many bytes do you want to CLEAR?";b
30 CLEAR r-(b+24)
40 GO SUB 200
50 FOR n=1 TO 23
60 READ d: POKE r+n,d
70 NEXT n
80 INPUT "Start address in Shadow ROM?";s
90 INPUT "Number of bytes to copy?";b
100 LET z=s: GO SUB 300
110 POKE r+12,l: POKE r+13,h
120 LET z=b: GO SUB 300
130 POKE r+18,l: POKE r+19,h
140 LET z=r+9: GO SUB 300
150 POKE r+2,l: POKE r+3,h
160 LET z=r+24: GO SUB 300
170 POKE r+15,l: POKE r+16,h
180 RANDOMIZE USR (r+1)
190 PRINT "Data stored at" :r+24
195 STOP
200 LET r=PEEK 23730+256*PEEK 23731: RETURN
300 LET h=INT (z/256): LET l=z-256*h: RETURN
400 DATA 33,0,0,34,237,92,207,50,225,225,33,0,0,17,0,0,1,0,0,237,176,199,201,
```



Punkt für Punkt

Das DVM-Modul, der Kern unseres Multimeters, kann Spannungen bis zu 1,9999 Volt messen. Für den sinnvollen Einsatz braucht es eine Abschwächer-Schaltung, damit sich auch höhere Spannungen bestimmen lassen. Weitere Zusatzschaltungen sind für die Messung von Strömen und Widerständen nötig.

Zum Aufbau der Schaltung sollten Sie diesmal besser keine Platine mit Kupferstreifen verwenden, sondern eine Lochplatine mit Lötunkten im 2,54mm-Raster: Leiterbahnen legen den Schaltungsaufbau relativ starr fest, bei einzelnen Lötunkten lassen sich dagegen kleine Abwandlungen der Schaltung sehr einfach durchführen.

Im Gegensatz zu normalen Digitalschaltungen hat der A/D-Wandler empfindliche Analogeingänge mit hoher Eingangsimpedanz. Diese Eingänge müssen sehr sorgfältig vor der Einstrahlung von Störsignalen geschützt werden. Wir haben die Schaltung so aufgebaut, daß der Analogteil auf der einen Seite des Wandler-ICs liegt (Pin 1 bis 10), der Digitalteil auf der anderen Chipseite. Beim Aufbau des Prototyps stellte sich heraus, daß die Leistungsfähigkeit der Schaltung bereits durch kleine Änderungen an der Position der Verbindungsdrähte merklich beeinflußt wird. Im Idealfall würde man für diese Schaltung eine

optimierte Leiterplatte verwenden – für die meisten Bastler ist jedoch der Aufwand für das Belichten und richtige Ätzen einer Platine zu aufwendig. Wir haben uns daher für eine normale Verdrahtung entschieden.

Das Bild zeigt die Platine von der Bauteilseite, die Verbindungsleitungen liegen auf der Unterseite. Zur Vereinfachung ist die Verdrahtung schematisiert dargestellt, Sie brauchen die Drähte natürlich nicht rechtwinklig zu führen. Verdrahten Sie besser nicht stramm von Punkt zu Punkt – wenn die Leitungen etwas länger als nötig sind, lassen sie sich später noch so lange bewegen, bis die Schaltung optimal arbeitet.

Unsere Testplatine haben wir in Wire-wrap-Technik mit Wrap-Sockeln und Wire-wrap-Litze hergestellt. Das hat viel für sich: Falsch angeschlossene Leitungen lassen sich leicht wieder abwickeln und zu einem anderen Kontakt führen. Sie können natürlich auch löten.

Bei den meisten ICs liegen die Anschlußbeinchen im Abstand von 2,54 mm ($\frac{1}{10}$ Zoll). Die dafür geeigneten Platinen haben ein Lochraster im gleichen Abstand, wobei jedes der Löcher mit einem dünnen Kupferferring umgeben ist, an dem die Bauteile dann bequem angelötet werden können.

Auf die Größe kommt es an

Damit die Bauteile genug Platz haben, sollte Ihre Platine mindestens 50 Reihen à 45 Löcher haben. Setzen Sie zuerst die drei IC-Sockel ein – den 28-poligen Sockel für den A/D-Wandler (IC1), den 8-poligen Sockel für das Timer-IC (IC2) und schließlich den 16-poligen Sockel des Siebensegment-Treibers (IC3). Mit ganz wenig Lötzinn auf je zwei Pins können Sie die Sockel auf ihrem Platz fixieren.

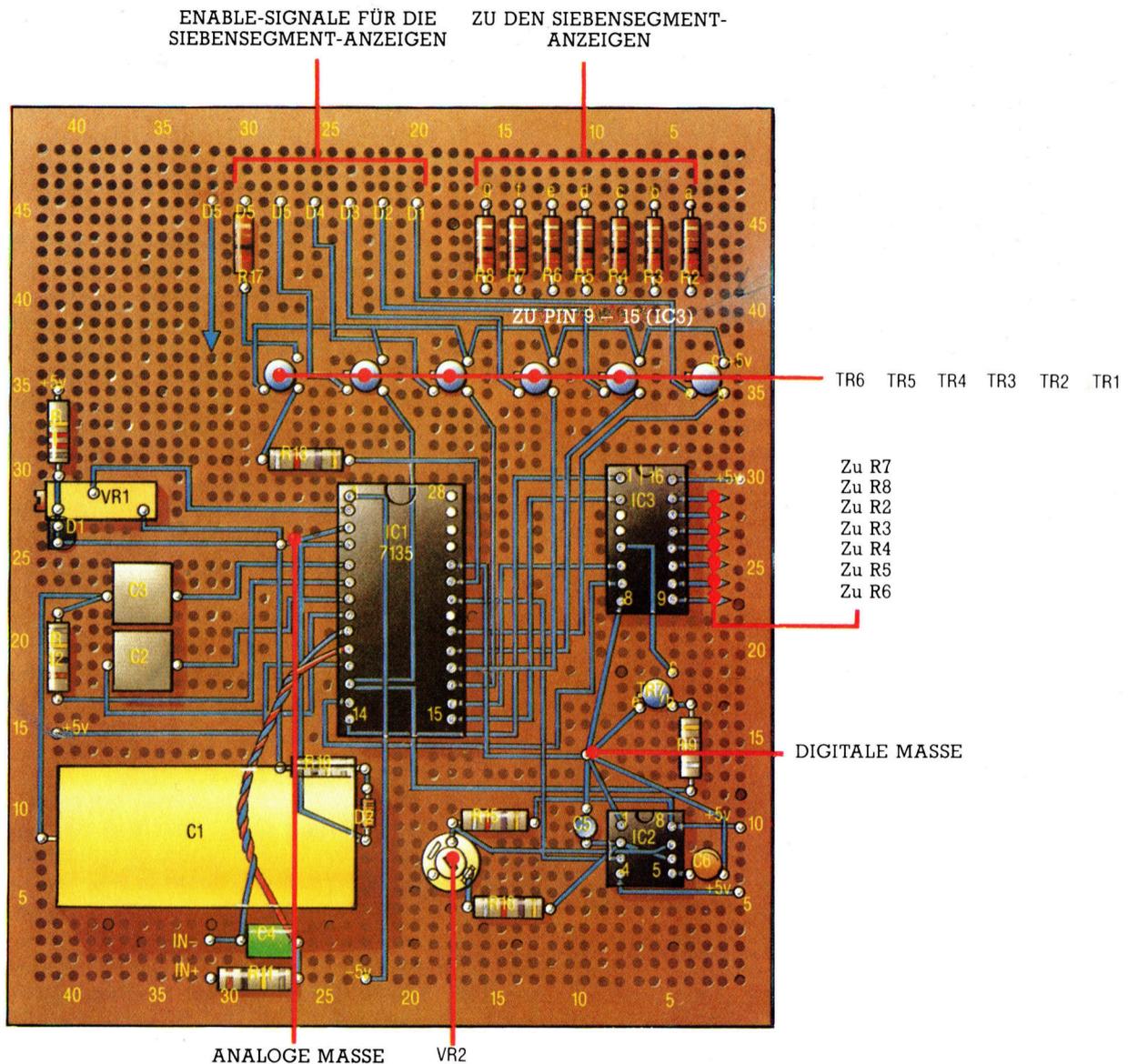
Bevor Sie weitere Bauteile einsetzen, löten Sie zunächst die +5-Volt-Leitungen an die drei entsprechenden Sockelpunkte. Wir haben nicht die gesamte Verdrahtung für die Versorgungsspannung abgebildet, um das Schaltbild einfacher zu halten. Es sind lediglich alle +5-Volt-Kontakte mit dem Netzteil zu verbinden. Nur ein IC, das 7135, braucht auch -5 Volt. Dazu legen Sie eine zusätzliche Leitung von Pin 1 des IC1 zu einem geeigneten Punkt am Rande der Platine.

Liste der Bauteile

Anzahl	Bez. im Plan	Bauteil
1	C1	Polypropylen-Kondensator; 0,47 μ F
2	C2–C3	Polycarbonat-Kondensator; 1,00 μ F
1	C4	Polycarbonat-Kondensator; 0,10 μ F
1	C5	Polystyrene-Kondensator; 470 pF
5	C6	Keramischer Scheibenkondensator, 22 000 pF
1	D1	Referenz-Zenerdiode 9491, Z-Spannung 1,22 Volt
1	D2	Diode 1N4148
6	TR1–5, TR7	NPN-Transistor 2N2219
1	TR6	PNP-Transistor 2N2905
1	VR1	10-gängiges Trimmerpotentiometer, 10 kOhm
1	VR2	Trimmerpotentiometer, 10 kOhm
1	IC1	Analog/Digitalwandler-IC Typ 7135
1	IC2	Low Power Timer-IC Typ ICM7555
1	IC3	Siebensegment Treiber-IC Typ 7447A
1	–	28-poliger IC-Sockel
1	–	16-poliger IC-Sockel
1	–	8-poliger IC-Sockel
1	R1	Widerstand 8,2 kOhm
7	R2–R8, R14, R17	Widerstand 180,0 Ohm
1	R9	Widerstand 47,0 kOhm
3	R10–R12	Widerstand 100,0 kOhm
1	R13, R15, R16	Widerstand 4,7 kOhm



Layout der Multimeter-Platine



Montage und Verdrahtung

Unser Platinenlayout ist für die Montage auf einer Lochplatte im 2,54-mm-Raster gedacht. Die Leitungen sind nur der Deutlichkeit halber rechtwinklig geführt. Die Verbindungen zwischen analoger und digitaler Masse und von D5 zur analogen Masse sowie alle +5-Volt-Versorgungsleitungen haben wir auf der Zeichnung weggelassen.

Den Kontakt zwischen dem analogen und dem digitalen Massepunkt sollten Sie mit möglichst kräftigem Draht herstellen. Eine +5-Volt-Versorgungsleitung können Sie in der Nähe der Kanten einmal ganz um die Platine herumführen – dann ist jeder Punkt leicht erreichbar. Wenn Sie Spannungen nur gegen Masse bestimmen wollen, muß

zusätzlich eine Verbindung zwischen Pin 9 des 7135-ICs und dem analogen Massepunkt hergestellt werden. Löten Sie zuerst nur die passiven Bauteile ein – auf die Behandlung der empfindlichen CMOS-ICs gehen wir später noch genauer ein, damit Ihnen kein Unglück passiert.

Zwischen Masse und die Versorgungsleitungen (+ 5 V) werden Scheibenkondensatoren gesetzt. IC1 sollte über je einen Kondensator an Pin 1 und Pin 11 mit der analogen Masse verbunden werden. Bei IC2 liegt ein Kondensator zwischen Pin 8 und der digitalen Masse, ein weiterer Kondensator verbindet Pin 16 von IC3 mit der digitalen Masse. Der 47 µF-Elko liegt mit dem Minuspol an Pin 1 von IC1, der Pluspol wird mit der Masse verbunden. Achten Sie auf richtige Polung!



Wir brauchen zwei verschiedene Massekontakte: Einen für die analoge Masse (in der Nähe von Pin 4 des IC1) und einen für die digitale Masse. Wie bereits mehrfach betont, müssen alle Analog-Leitungen auf Massepotential mit auf einen einzigen Kontakt gelegt werden. Ebenso werden alle digitalen Masseleitungen in einem Punkt zusammengeführt. Die analogen Massekontakte von IC1 (Pin 3), IC2 (Pin 1) und IC3 (Pin 8) werden also zusammengefaßt. Pin 24 von IC1 schließen Sie am digitalen Massepunkt an. Die beiden Massepunkte verbinden Sie zum Schluß mit möglichst kräftiger Litze. (Die Verbindung fehlt im Bild, darf aber keinesfalls vergessen werden!)

Vor dem Einbau weiterer Komponenten sollten alle Verbindungen mit einem Ohmmeter oder einem einfachen Durchgangsprüfer auf guten Kontakt getestet werden.

Nach den IC-Sockeln setzen Sie die weiteren Bauteile in die angegebenen Positionen. Wenn die Größe der von Ihnen verwendeten Komponenten von unserem Beispiel abweicht, müssen Sie die Verdrahtung anpassen. Die

meisten Bauteile können durch Vergleichstypen ersetzt werden. Am schwierigsten ist es, den Integrations-Kondensator C1 (0,47 μ F) zu bekommen.

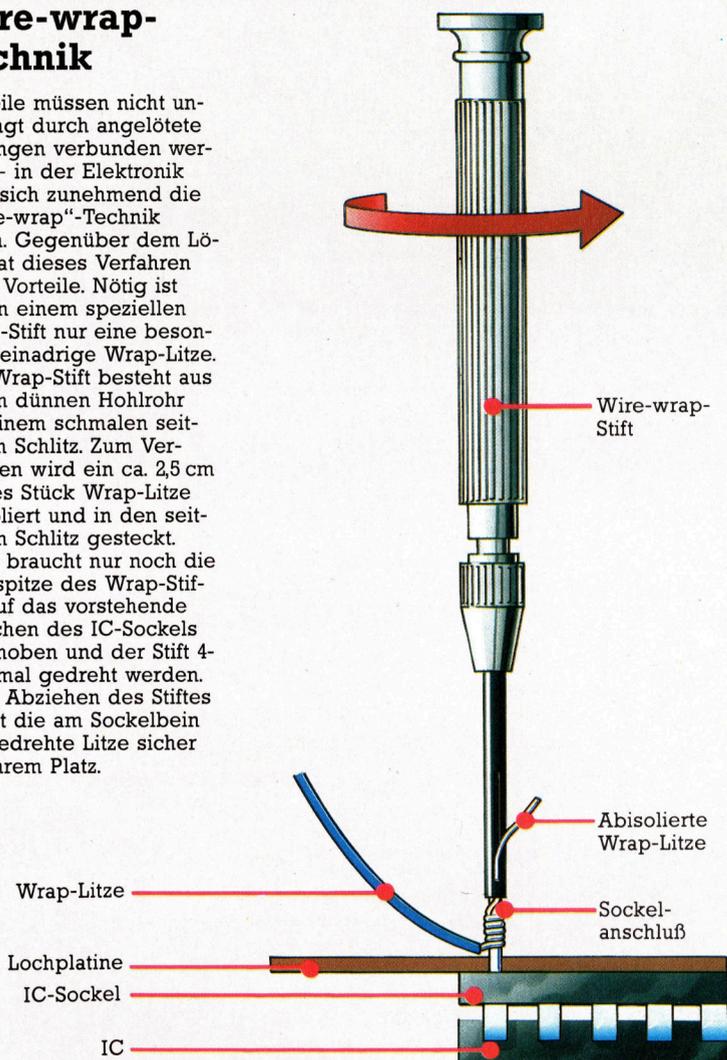
Für möglichst genaue Messungen sollten Sie hier einen Polypropylen-Kondensator verwenden. Wenn es diesen Bautyp nirgends zu kaufen gibt, müssen Sie auf einen Polycarbonat-Kondensator ausweichen.

Vielleicht haben Sie auch bei der Beschaffung geeigneter Siebensegment-LEDs Schwierigkeiten. Es ist nicht immer einfach, ein Anzeigefeld für die 1 mit dem Plus- oder Minuszeichen davor (für die erste Ziffer der Anzeige) zu bekommen. Wir haben die Schaltung so abgewandelt, daß für alle Ziffern das gleiche Siebensegment-Display verwendet werden kann. Als erste Ziffer erscheint entweder nichts oder die Eins. Negative Meßwerte können durch den mittleren Querbalken des Zifferndisplays kenntlich gemacht werden. Positive Werte haben kein Vorzeichen.

VR1 sollte ein mehrgängiges Trimmerpoti von guter Qualität sein, für VR2 können Sie eine preiswertere Bauform wählen. Im Idealfall ist D1 eine Referenz-Zenerdiode (Typ 9491). Falls Sie diese nicht bekommen, kann auch eine normale 2,7-Volt-Zenerdiode eingesetzt werden. Allerdings leidet die Meßgenauigkeit des Multimeters ein wenig darunter.

Wire-wrap-Technik

Bauteile müssen nicht unbedingt durch angelötete Leitungen verbunden werden – in der Elektronik setzt sich zunehmend die „Wire-wrap“-Technik durch. Gegenüber dem Löten hat dieses Verfahren viele Vorteile. Nötig ist neben einem speziellen Wrap-Stift nur eine besondere einadrige Wrap-Litze. Der Wrap-Stift besteht aus einem dünnen Hohlrohr mit einem schmalen seitlichen Schlitz. Zum Verdrahten wird ein ca. 2,5 cm langes Stück Wrap-Litze abisoliert und in den seitlichen Schlitz gesteckt. Dann braucht nur noch die Hohlspitze des Wrap-Stiftes auf das vorstehende Beinchen des IC-Sockels geschoben und der Stift 4- bis 5mal gedreht werden. Beim Abziehen des Stiftes bleibt die am Sockelbein festgedrehte Litze sicher auf ihrem Platz.



Transistortypen

Die Transistoren (außer TR7) dienen zum Einschalten der LEDs. Beachten Sie, daß Transistor 1 bis 5 NPN-Typen, TR6 jedoch ein PNP-Transistor ist. PNP deshalb, weil TR6 bei einem logischen Low schalten muß, während die anderen Transistoren auf ein logisches High reagieren sollen. TR7 wird für die Unterdrückung führender Nullen gebraucht. D2 kann jede kleine Siliziumdiode sein – den angegebenen Typ werden Sie wahrscheinlich ohne Schwierigkeiten im Bastelgeschäft bekommen.

Die Eingangssignale werden mit einem kurzen Stück verdrehter Leitung oder über ein dünnes Koaxialkabel an die Pins 9 und 10 von IC1 geführt (Abschirmung an Pin 9, Innenleiter an Pin 10). Bei unseren Versuchen mit der Testschaltung erhielten wir eine wesentliche Steigerung der Meßgenauigkeit, wenn Pin 9 von IC1 mit der analogen Masse verbunden war. Diesen Anschluß können Sie vornehmen, wenn Sie keine potentialfreien Messungen machen wollen.

Bevor Sie jetzt die ICs einsetzen, sollten Sie die Schaltung noch einmal sehr gründlich mit dem Plan vergleichen. Achten Sie auf korrekte Positionierung der Kollektor- und Emitteranschlüsse – bei TR6 liegen sie andersherum als bei TR1 und TR5.

Im nächsten Abschnitt behandeln wir die Verdrahtung des LED-Feldes und den Umgang mit den ICs.



Spitzenreiter

Das kommerzielle Textverarbeitungspaket Samna Word III gehört aufgrund des hohen Verkaufspreises in die „Rolls-Royce“-Klasse der Textsysteme. Wir stellen fest, welche Eigenschaften diesen hohen Preis rechtfertigen.

Wie fast alle kostspieligen Textsysteme läuft auch Samna Word III auf allen IBM-Kompatiblen, die mit MS-DOS-Versionen 2.0 und darüber arbeiten, da erst in DOS 2.0 die hierarchischen Inhaltsverzeichnisse mit den „Pfadern“ eingeführt wurden. Das Paket besteht aus einem Einführungshandbuch, Tastaturaufgaben für eine Reihe von Sprachen und einem detaillierteren Anwenderhandbuch mit Erläuterungen der Befehle und fünf Programmdisketten. Eine Menge Stoff!

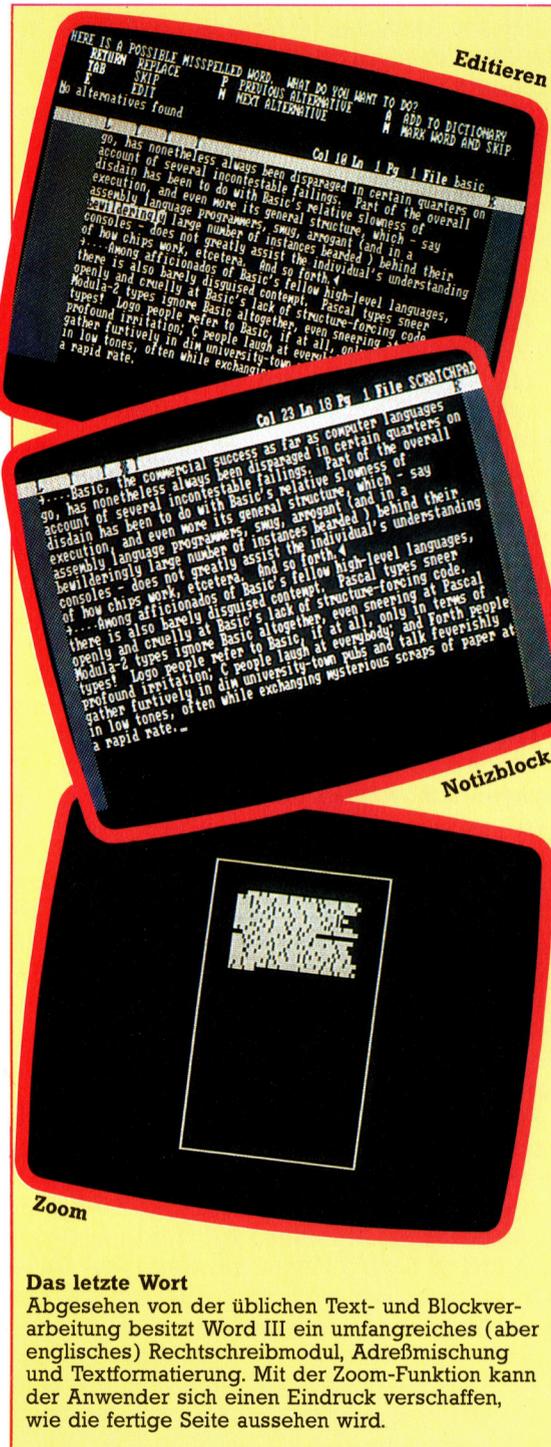
Das Hauptprogramm belegt zwei Disketten, auf zwei weiteren sind Dienstprogramme, Druckeroutinen und das Directory untergebracht. Die fünfte Diskette gehört eigentlich nicht zum Programm, sondern enthält eine Reihe kurzer Lernprogramme, die den Anwender mit dem Programm vertraut machen.

Nach der Installation nehmen die 45 Anwenderdateien des Programms (auf mehreren Disketten) insgesamt 700 KBytes ein – recht viel für ein einziges Programm. Beim Aufruf des Programms erscheint zunächst einen „Notizblock“, der anzeigt, daß im Augenblick keine Datei geladen ist. Farbige Balken geben die Randpositionen an, und ein Querbalken zeigt die normalen Randgrenzen und die Stellungen der Tabs. Der obere Bildschirmrand enthält einen Zeiger auf die aktuelle Cursorposition, den Dateinamen und die READY!-Anzeige.

Einer der größten Vorteile des IBM PC (und kompatibler Geräte) sind die vielen frei programmierbaren Funktionstasten. Die Programmierer von Word III haben diesen Vorteil voll ausgenutzt und nicht nur eine Reihe von Befehlen auf die Funktionstasten gelegt, sondern auch andere Tasten für ihre Zwecke höchst sinnvoll eingesetzt.

So wird der Zehnerblock des IBM PC zwar auch standardmäßig als Cursorsteuerung eingesetzt, doch bewegen in Word III einige Tasten (7, 9 und 1) den Cursor wortweise, satzweise und absatzweise. Die Taste „Scroll Lock“ kennzeichnet Blöcke zur speziellen Bearbeitung, und die Funktionstasten rufen oft eingesetzte Befehle wie Unterstreichung und Formatierung zur Ausführung auf.

Word III ist umfassender als alle anderen in



Das letzte Wort

Abgesehen von der üblichen Text- und Blockverarbeitung besitzt Word III ein umfangreiches (aber englisches) Rechtschreibmodul, Adreßmischung und Textformatierung. Mit der Zoom-Funktion kann der Anwender sich einen Eindruck verschaffen, wie die fertige Seite aussehen wird.

dieser Serie untersuchten Programme. Nach Aufruf einer Funktion erscheint oft die Frage, welcher Textbereich bearbeitet werden soll – das gesamte Schriftstück, der Absatz oder der Text hinter dem Cursor.

Die Samna-Programmierer haben versucht, größtmögliche Flexibilität und maximale Bedienerfreundlichkeit miteinander zu verbinden. So besitzt Word III beispielsweise kein



Samna Word III

WORTUMBRUCH

Word III bietet automatischen Wortumbruch und Zeilenjustierung.

BLOCKVERARBEITUNG

Es gibt alle üblichen Blockbefehle, darunter auch die Bearbeitung von Spaltenblöcken.

HILFSFUNKTIONEN

Das Programm bietet ausführliche Hilfsinformationen.

80 ZEICHEN PRO ZEILE

Die Standardeinstellung liegt bei 72 Spalten pro Zeile, läßt sich aber leicht auf 80 Spalten umstellen.

WORTZÄHLER

Nicht vorhanden

SUCHEN/TAUSCHEN

Zeichenketten bis zu 30 Zeichen sind möglich; es gibt eine Reihe von Wahlmöglichkeiten.

SPEZIALFUNKTIONEN

Obwohl Word III praktische Fähigkeiten wie Zoom besitzt, fehlen doch etliche Standardfunktionen (z. B. die Darstellung einer Unterstreichung).

MISCHEN MIT ADRESSEN

Word III besitzt eine Datenbank und Routinen zur Mischung von Text und Adressen.

RECHTSCHREIBPRÜFUNG

Der Prüfungsvorgang ist schnell und einfach, das englische Wörterbuch enthält jedoch viele amerikanische Schreibweisen.

SCHRIFTARTEN

Word III unterstützt Subscripts und Fettdruck, sieht aber leider keine weiteren Schrifttypen vor.

DATEIVERKETTUNG

Nicht vorhanden

automatisches Einfügen. Wenn Text in die Mitte eines Schriftstückes eingesetzt werden soll, muß erst mit der Insert-Taste Platz geschaffen werden. Tasword II beispielsweise verlangt nach dem Einfügen eine Neuformatierung des Absatzes. In Word III genügt es, „Insert“ nochmals zu drücken, und der Absatz formatiert sich automatisch.

Auch die Hilfsfunktion von Word III ist praktisch. Sie wird mit der Help-Taste links oben im alphanumerischen Tastenfeld aufgerufen und zeigt die verfügbaren Optionen an. Die Hilfsinformation erscheint aber auch automatisch, wenn der Anwender vor Aufruf einer Funktion lange zögert.

Einige Funktionstasten sind mit einem breiten Spektrum verschiedenartiger Abläufe programmiert, darunter auch die Do-Taste. Unter den Dienstmodulen, die damit aufgerufen werden können, befinden sich zwar normale Dateifunktionen wie „delete“ und „copy“, doch gibt diese Taste auch Zugang zu Fähigkeiten, über die andere Programme nicht verfügen.

Bei umfangreichen Schriftstücken kann der Bildschirm nur einen kleinen Ausschnitt zeigen. Wenn Sie aber wissen wollen, wie die ganze Seite aussieht, kann Ihnen die Zoom-Funktion mit Balken anzeigen, welches Gesicht die Titel und Absätze der Seite geben. Auf diese Weise läßt sich die Seite wunschgemäß gestalten, ohne daß der Text mehrmals ausgedruckt werden muß.

Der richtige Kniff

Die Do-Taste beseitigt aber noch einen weiteren Nachteil der Textfenster. Wenn Sie in einem sehr breiten Schriftstück auf eine Spalte außerhalb des angezeigten Fensters springen wollen, können Sie den Text mit dem Befehl Fold elektronisch „falten“ und damit weit entfernte Teile der Seite gleichzeitig darstellen.

Kommerzielle Programme müssen heute nicht nur mit dem IBM PC kompatibel sein, sondern auch mit anderer Software, speziell mit dem weit verbreiteten Lotus 1-2-3. Word III kann daher ASCII-Dateien von der Diskette lesen und im Samna-Format anzeigen. Auf diese Weise lassen sich auch von Lotus erzeugte Dateien von Word III verarbeiten.

Jede kommerzielle Textverarbeitung, die beim User ankommen will, muß weiterhin Text und Adressen mischen und Rechtschreibung prüfen können. Word III verfügt über beide Möglichkeiten. Das Prüfmodul für Rechtschreibung – proof – ist sogar eine der umfassendsten Routinen dieser Art.

Wie die anderen Word-III-Module besitzt das Rechtschreibsystem eine Reihe von Optionen, mit denen Sie festlegen können, was geprüft werden soll. Danach vergleicht der Computer jedes Wort mit seinem internen „Wörterbuch“. Bei einem unbekanntem Wort zeigt er eine Reihe Wahlmöglichkeiten an, darunter

auch andere Schreibweisen.

Das Word-III-Modul zum Mischen von Adressen und Texten heißt „Automatic Merge“ und enthält eine kleine Datenbank. Darin lassen sich eine Reihe Felder – Vorname, Name und Adresse etwa – unterbringen, die im Schriftstück an der gewünschten Position eingesetzt und gedruckt werden. Das Programm kann auch mit sogenannten „Wildcards“ einzelne Datensätze heraussuchen. Samna Word III ist eine außerordentlich flexible Textverarbeitung, die im Vergleich mit anderen Systemen nicht schlecht abschneidet. Die Frage bleibt jedoch, ob das Paket seinen Preis wert ist. Wenn Sie ein umfassendes Programm mit ungewöhnlichen Fähigkeiten brauchen, kann hier die Antwort durchaus „Ja“ lauten.

Das Kleingedruckte

Am anderen Ende der Skala von Samna Word III liegt Mini Office von Database Publications. Das Programm zum Taschengeldpreis wird auf Cassette geliefert. Es enthält nicht nur eine Textverarbeitung, sondern auch eine Datenbank, Dienstprogramme für Grafik und ein Kalkulationssystem. Das Paket läuft auf dem Acorn B, Electron, Commodore 64 und den Schneider-Computern. Doch obwohl die Textverarbeitung nur wenige der hochentwickelten Funktionen von Word III besitzt, enthält es eine Reihe von Fähigkeiten, die man auch in teureren Programmen recht häufig vergeblich sucht.

Mini Office deckt zwar nur die Grundfunktionen ab, doch setzen viele Anwender beispielsweise die komplexe Druckformatierung höher entwickelter Programme nie ein. Das Programm ist menügesteuert und kann über die Funktionstasten eine Reihe Funktionen aufrufen, darunter das Laden, Speichern und Drucken von Texten.

Das Textsystem unterstützt automatischen Wortumbruch und Einsetzen/Löschen, aber keine Formatierung. Am oberen Bildschirmrand sind Informationen aufgeblendet, darunter ein Wortzähler, die Zahl der verfügbaren Zeichen und eine Uhr. Text läßt sich auch von einem Teil des Schriftstückes in einen anderen kopieren, jedoch nicht blockweise, sondern durch einfaches Übertragen der Zeichen von der vorigen auf die aktuelle Cursorposition.

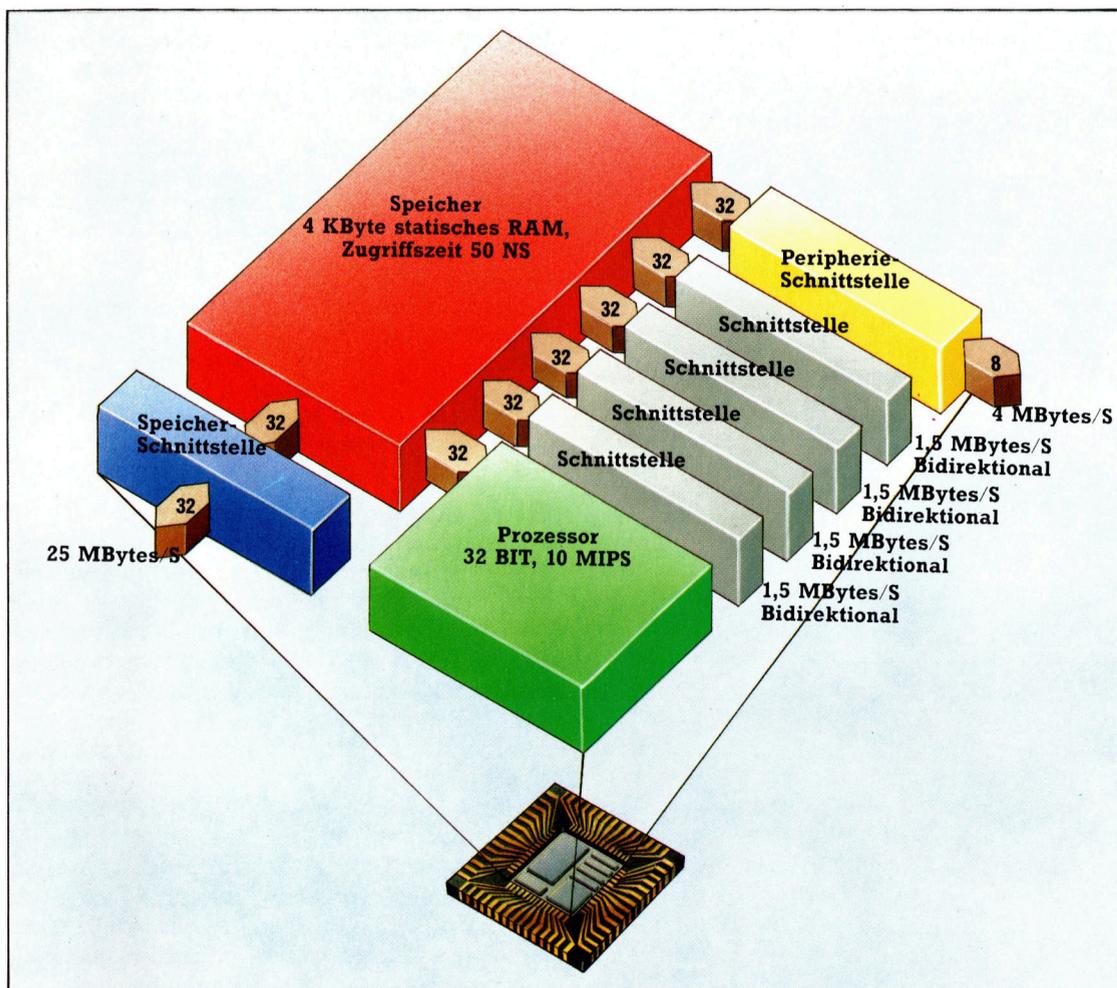
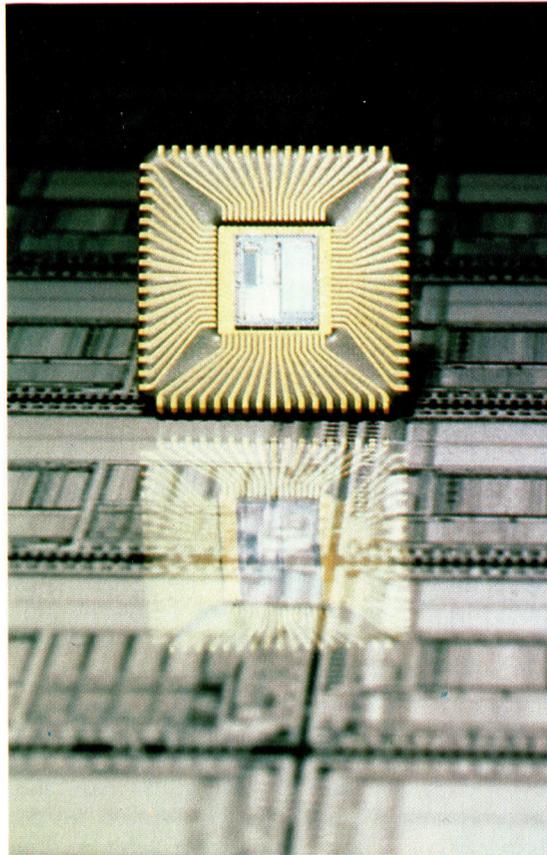
START
A word processor is ideal for writing letters and reports instead of using a typewriter or pen and paper. When you make a typing error or change your mind, the word processor enables you to edit the text with ease.
END



Schnelle Schritte

Der Einfluß, den die Entwicklung von Transputern auf die Computertechnik hat, läßt sich nur mit der Entdeckung des Transistors in den fünfziger Jahren vergleichen. Sie erlauben höchste Verarbeitungsgeschwindigkeiten.

Die Erfindung des Transistors zu Anfang der fünfziger Jahre bereite den Weg zur Serienproduktion von Computern. Ein Microprozessor besteht heute aus zahlreichen identischen Transistoren, die auf einem einzelnen Chip untergebracht sind. Der Inmos Transputer ist ein Prozessor, der ähnlich wie ein Transistor angewandt werden soll – als Komponente oder Grundbaustein für größere Systeme. Der Name ergibt sich aus TRANSistor und ComPUTER. Eine Verbindung mehrerer



Von der Firma Inmos stammt ein neuer Stern am IC-Himmel: Der Transputer ist ein kompletter 32-Bit-Computer auf einem einzigen Chip. Neben dem RAM-Speicher sind auch vier serielle Schnittstellen integriert. Transputer sind für den Einsatz in größeren Parallelrechnern vorgesehen. Durch die seriellen Schnittstellen lassen sich viele Transputer zu einem Netzwerk zusammenfassen. Der Aufbau eines solchen Netzwerkes kann die unterschiedlichsten Strukturen aufweisen – allein die Problemstellung determiniert die Rechnerarchitektur.

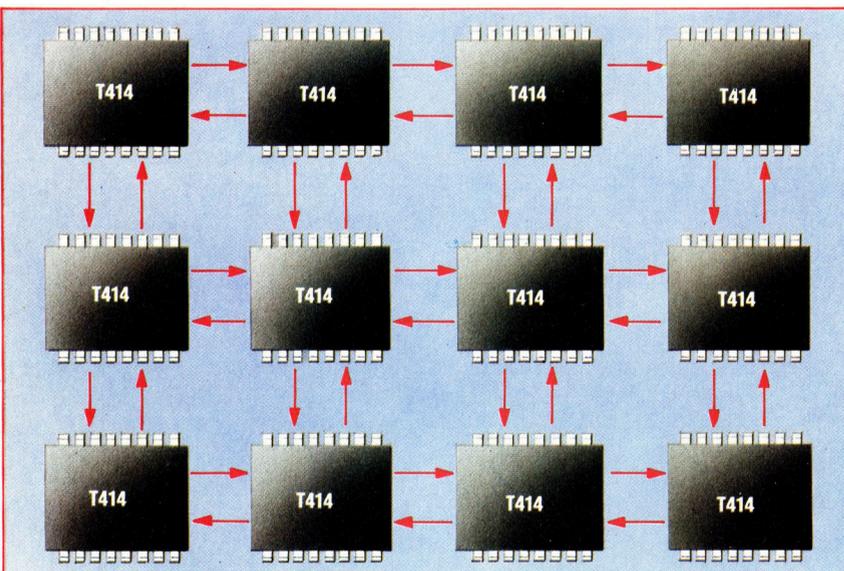
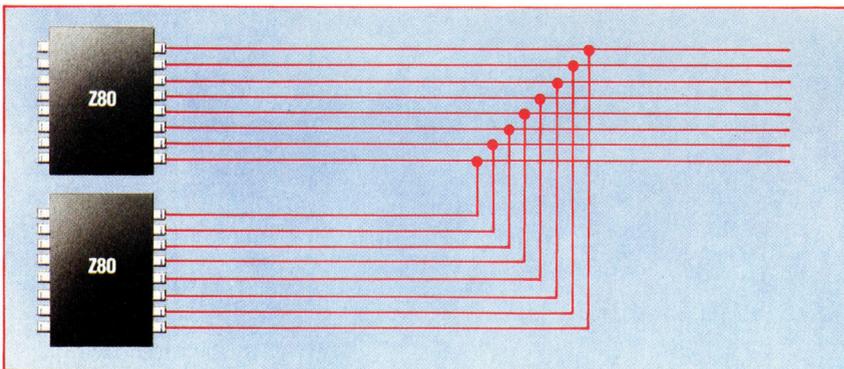


Transputer macht Parallelverarbeitung von Daten möglich, wobei jeder Transputer eine getrennt programmierbare Einheit darstellt.

Die Zusammenschaltung funktioniert, weil Transputer alle Komponenten eines vollständigen Computers vereinigen. Jeder Transputer hat eine CPU, Speicherplatz und serielle Datenschnittstellen. Dadurch kann er Programme selbstständig ausführen, wobei als externe Unterstützung nicht mehr als Strom und Clock-Pulse nötig sind. Transputer können sich zudem über ihre seriellen Datenverbindungen einfacher verständigen als die üblichen CPUs.

Ein herkömmlicher Microprozessor wie etwa der Z80 „beherrscht“ jede Schaltung, in der er eingesetzt wird. Die Verständigung mit der Außenwelt erfolgt über den prozessorgesteuerten Parallelbus. Zwei Z80 zu einer Unterhaltung zu bewegen, ist aus zwei Gründen schwierig: Zunächst gibt es Platzprobleme auf der Karte, wenn viele Prozessoren denselben Bus benutzen sollen – jeder Anschluß braucht eben acht Leitungen. Bei den modernen 16- oder 32-Bit-CPU's wird dieses Problem natürlich noch entsprechend größer.

Eine weitere Schwierigkeit ist, daß jeder Z80 den Bus allein kontrollieren möchte. Es muß also durch Software verhindert werden, daß beide Prozessoren gleichzeitig senden. Außerdem ist ein Z80 beim Datenaustausch über den



Von Neumann-Architektur

Alle modernen Computer – bescheidene Homecomputer genau wie Super-Rechner – funktionieren nach den Neumannschen Prinzipien. Ein Computer nach dem Neumann-Prinzip verfügt über einen zentralen Prozessor (CPU), der mit einem Speicher verbunden ist. Im Speicher stehen sowohl die auszuführenden Befehle als auch Daten, die mit diesen Befehlen verarbeitet werden sollen. Die Befehle werden nacheinander (sequentiell) abgerufen und von der CPU ausgeführt.

Dieser Aufbau, der von Neumanns Namen trägt, ermöglichte die Konstruktion von Multifunktionsanlagen. Wichtig ist dabei, daß die Zahlen im Speicher sowohl Programme als auch Daten sein können. Dadurch bekamen Computerkonstruktoren einen zusätzlichen Freiraum. Bisher mußten sie nämlich ihre Apparate für jede Spezialaufgabe wie etwa Entschlüsselung oder Maschinensteuerung neu konstruieren.

Das Problem bei einem von-Neumann-Rechner liegt in der sequentiellen Befehlsverarbeitung. Wie schnell man die CPU auch macht, sie muß ihre Befehle und Daten aus dem Speicher holen. Hierbei gibt es physikalische Grenzen – die Geschwindigkeit, mit der ein einzelner Microprozessor Daten mit dem Speicher austauschen kann, wurde zum Engpaß.

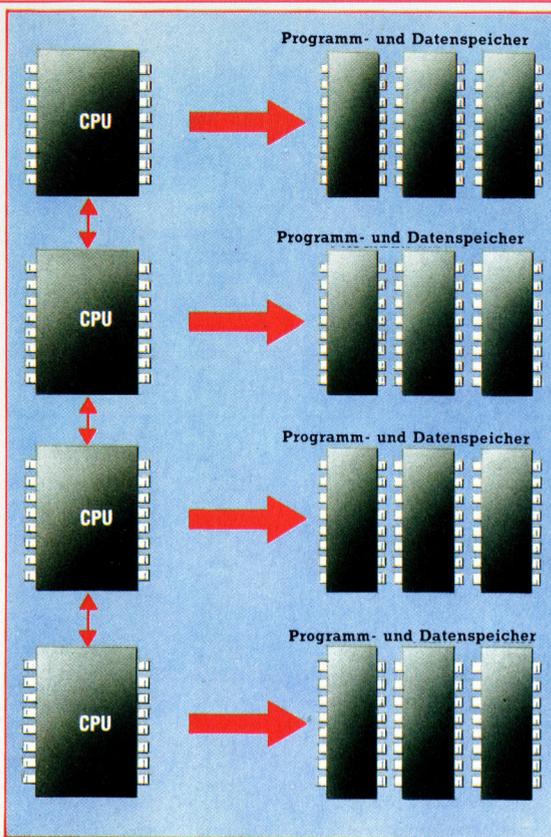
Dieser Engpaß kann durch eine Vielzahl von gleichzeitig arbeitenden CPUs umgangen werden. Die Technik dahinter wird als „Parallelverarbeitung“ bezeichnet. Trotz beschränkter Lesegeschwindigkeit können 100 CPUs mit 100 Speichereinheiten eben auch hundertmal schneller arbeiten als ein einzelner Prozessor. Parallele Rechner werden zusätzlich dadurch beschleunigt, daß jede einzelne CPU weniger mit dem Speicher kommunizieren muß.

Parallelverarbeitung hat natürlich auch ihre spezifischen Schwierigkeiten. Um ein Programm auf mehreren CPUs laufen zu lassen, muß jeder Prozessor wissen, was er zu tun hat. Er muß seinen Teil abarbeiten, ohne mit der Tätigkeit seiner Kollegen zu kollidieren

Bus für weitere Aufgaben nicht verfügbar.

Beim Transputer ist dieser Vorgang einfacher – er braucht nur zwei Leitungen, um mit einem „Kollegen“ in Kontakt zu treten. Jeder Transputer hat zudem vier serielle Schnittstellen, er kann also mit vier anderen Chips gleichzeitig kommunizieren. Damit sind zwei- oder gar dreidimensionale Felder von Transputern bei minimalem Verdrahtungsaufwand realisierbar.

Transputer brauchen bei ihrer Kommunikation keine spezielle Software, um Prioritätsfragen zu klären. Ein einzelner Transputer kann vieles gleichzeitig erledigen – etwa Daten von allen vier Schnittstellen empfangen, ohne seine anderen Aufgaben deshalb nur einen Sekundenbruchteil zu unterbrechen.



oder sie zu überholen.

Wenn wir zum Beispiel den Ausdruck
 $(3+4) \cdot (9-7+3)$

von zwei Prozessoren auswerten lassen wollen, könnte jeder einen Klammerausdruck berechnen, und einer von beiden die abschließende Multiplikation vornehmen.

Natürlich muß vorher festgelegt sein, welche CPU den letzten Schritt ausführt. Wenn das CPU1 ist, darf sie ihre letzte Berechnung nicht ausführen, bevor CPU2 fertig ist. Die CPUs müssen sich also verständigen, also auch Teilergebnisse austauschen können. Solche Probleme lassen sich mit herkömmlichen Programmiersprachen nur ausgesprochen mühsam lösen.

Diese Arbeitsweise erlaubt es, große Programme in Einzelteile zu zerlegen, die dann auf mehreren Transputern simultan abgearbeitet werden. Jeder Transputer kann seinen Teil des Programms im eigenen Speicher ablegen und die „Kollegen“ zum gegebenen Zeitpunkt über ein Interface informieren. Muß ein spezieller Programmteil besonders schnell ausgeführt werden, ist nur ein vergrößerter Einsatz von Transputer-Arbeitskräften notwendig.

Das Geheimnis, das Transputer zu all diesem befähigt, liegt in der Sprache, in der sie programmiert werden. Anders als ein traditioneller Microprozessor wird ein Transputer nicht in Assembler, sondern in einer speziell entwickelten Sprache namens OCCAM programmiert. Sie ist auf Parallelverarbeitung zu-

Einsame Spitze

Technologie: Der T414 wird in 1.5 Mikron CMOS-Technik hergestellt. Dadurch ist die Packungsdichte größer als bisher üblich.

Prozessor: 32-Bit-Microprozessor mit etwa zehn Millionen Befehlen pro Sekunde (etwa 10mal schneller als etwa der Motorola 68000). Arbeitet mit 70 Befehlen, die OCCAM direkt unterstützen.

Speicherplatz auf dem Chip: Statisches RAM mit 50 ns Zugriffszeit. Wird benutzt, um kürzere OCCAM-Prozesse durchzuführen; ersetzt die Register konventioneller Microprozessoren. Adressierbarer Speicher: bis zu vier Gigabytes. Damit kann ein Transputer als Einzelrechner etwa in PCs eingesetzt werden. Der Prozessor unterscheidet nicht zwischen internem und externem Speicher. Einzige Ausnahme ist die Zugriffszeit.

Serielles Interface: Jedes Interface kann bidirektional zehn Megabits pro Sekunde verarbeiten. Damit haben die vier Schnittstellen eine Kapazität von 80 Megabit pro Sekunde. Der Transfer verläuft asynchron mit Hardware-Handshake; Clocksignale müssen daher nicht berücksichtigt werden.

Gleichzeitigkeit: Ein einzelner T414 kann mehrere Aufgaben gleichzeitig wahrnehmen. Er wird dabei von einem Prozeßsteuer-Modul überwacht. Alle Bereiche des Chips arbeiten unabhängig voneinander. Der Prozessor kann also auch dann Rechnungen durchführen, wenn Daten über das Interface ein- oder ausgehen.

Weiterentwicklung: Der T414 ist der erste aus einer Familie kompatibler Transputer, bei denen einzelne Speichergruppen oder Schnittstellen durch spezielle Funktionsgruppen ersetzt werden sollen. Der Grafik-Prozessor G213 und das Controller-IC M212 stehen kurz vor der Fertigstellung.

Die Firma: Inmos wurde 1978 mit Unterstützung der englischen Regierung gegründet und gehört heute zur Thorn EMI Gruppe. Entwurf und Entwicklung der Transputer und der Programmiersprache OCCAM wurde größtenteils in Bristol durchgeführt, produziert wird in Newport, Wales.

geschnitten. Ein OCCAM-Programm setzt sich aus einzelnen „Prozessen“ zusammen, die etwa einer Subroutine oder Prozedur in BASIC oder PASCAL entsprechen. Im Unterschied zu BASIC oder PASCAL können mehrere Prozesse gleichzeitig ausgeführt werden.

OCCAM kann Daten nicht nur über Variablen, sondern auch über „Kanäle“ austauschen. Damit stehen Ergebnisse aus verschiedenen, gleichzeitig ablaufenden Prozessen allen angeschlossenen Transputern sofort zur Verfügung. Die Kanäle sind so intelligent aufgebaut, daß keine Information ausgetauscht werden kann, ohne daß beide Partner dafür bereit sind. Dabei spielt es für OCCAM keine Rolle, ob ein Prozeß auf demselben Transputer abläuft oder auf einem anderen.



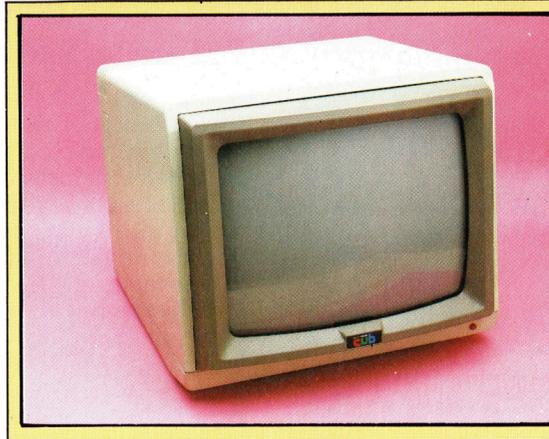
380Z, der Robuste

Robuster Aufbau und hohe Grafikauflösung haben diesem Microcomputer vor allem in englischen Schulen und auch beim Militär zu einer beträchtlichen Verbreitung verholfen.

Die Rechner der britischen Firma Research Machines Ltd. heben sich durch eine bemerkenswerte Langlebigkeit vom übrigen Microcomputer-Angebot ab. Dabei liegen diese Maschinen weder technisch an der Spitze, noch sind sie besonders preisgünstig, dafür aber extrem stabil und zuverlässig. Der bekannteste Rechner dieses Herstellers, der RML 380Z ist, verglichen mit andern Micros, geradezu monströs: Die Zentraleinheit steckt in einem 19-Zoll-Gehäuse mit massiven Haltegriffen, wie es sonst für Laborelektronik verwendet wird. Nach Abnehmen des Deckels wird der Grund deutlich: Fast ein Viertel des Volumens verschlingt das Netzteil. Schon an dessen Gewicht ist zu erkennen, daß es sich nicht um ein modernes Schaltnetzteil, sondern um einen konservativen Aufbau mit großem Transformator und mächtigen Kondensatoren handelt. Dieses althergebrachte Konzept hat den nicht zu unterschätzenden Vorteil, extrem belastungsfähig und fast unzerstörbar zu sein.

Die hohe Zuverlässigkeit war wohl mit ausschlaggebend dafür, daß der 380Z vom britischen Verteidigungsministerium in großen Stückzahlen für die Lagerverwaltung und ähnliche Zwecke angeschafft wurde. In Schulen mit Leistungskursen in Mathematik und Naturwissenschaften fand der Rechner dagegen vor allem wegen der hochauflösenden Grafik Eingang, die sich in Lehrveranstaltungen für visuelle Demonstrationen sehr bewährte.

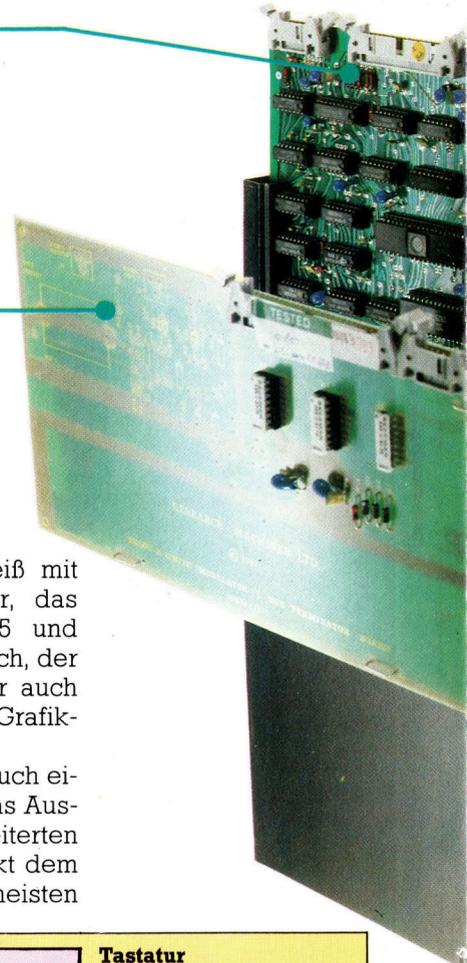
Das Programmpaket für hochauflösende Grafik (High Resolution Graphics = HRG) besteht aus einer Anzahl von Maschinencode-Routinen, die in Benutzerprogrammen aufrufbar sind, und mit denen sich die über die HRG-Karte erzeugte Bildschirmausgabe beeinflussen läßt. Diese Karte ermöglicht überhaupt erst den Grafikbetrieb und ist auch für heutige Verhältnisse noch recht leistungsfähig, obwohl sie schon etliche Jahre alt ist. Durch Verändern des Inhalts bestimmter Speicherplätze lassen sich unterschiedliche Grafikauflösungen zwischen 160 x 96 und 320 x 192 Pixel wählen. Dabei sind die üblichen Farben Rot, Grün,



Monitor
Um die Möglichkeiten des 380Z sinnvoll nutzen zu können, ist ein Farbmonitor mit RGB-Anschluß unerlässlich. Der Rechner ist wahlweise mit 40- oder 80-Zeichen-Karte erhältlich, und das bestimmt auch die Anforderungen, die bei der Anschaffung des Monitors entsprechend zu berücksichtigen sind.

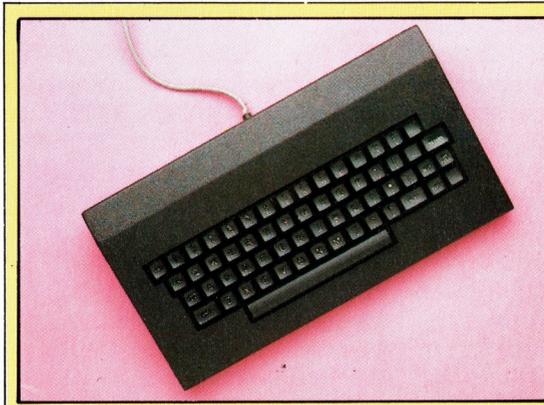
Disketten-Steuerkarte
Diese Leiterplatte enthält einen speziellen Steuerbaustein für die Diskettensteuerung, außerdem einen Z80-Timer-Chip sowie einen Interface-Baustein 8521 für die serielle Ein/Ausgabe.

Bus-Abschlußkarte
Der „Bus Terminator“ hat die Aufgabe, den internen Bus am CPU-fernen Ende elektrisch reflexionsfrei abzuschließen, um das Auftreten von möglichen Störpulsen zu verhindern.



Blau, Gelb, Magenta, Cyan sowie Weiß mit max. 255 Helligkeitsstufen darstellbar, das heißt bis zu 1786 Farbtöne (7 x 255 und Schwarz). Man kann den Speicherbereich, der für die Farbintensität gedacht ist, aber auch zum gleichzeitigen Ablegen mehrerer Grafikseiten verwenden.

Die zugehörigen Steuerbefehle wie auch einige verwandte Kommandos (z. B. für das Ausdrucken von Grafiken) sind im erweiterten RML-BASIC enthalten. Da dieser Dialekt dem Microsoft-BASIC nahesteht, haben die meisten



Tastatur
Die Tastatur des 380Z ist in einem kleinen, aber gewichtigen Metallgehäuse untergebracht. Die Tasten selbst sind weitgehend normgerecht angeordnet und qualitativ hochwertig – leichtgängig, doch mit solidem Anschlaggefühl. Die Tastatur ist offensichtlich für harte Beanspruchung gedacht und hervorragend für den Gebrauch in Schulen geeignet.



RAM-Platine

Auf der CPU-Leiterplatte ist nicht genügend Platz für eine brauchbare Arbeitsspeicherkapazität. Deshalb ist eine zusätzliche RAM-Karte vorgesehen, mit der sich der Adreßraum des Z80 von 64 KByte voll ausnutzen läßt.

Video-Karte

Die Chips für die Erzeugung der Bildschirm-signale befinden sich auf einer separaten Karte, die vom Prozessor angesteuert wird. Es gibt Videokarten mit 40 und 80 Zeichen pro Zeile.

Z80-Busverbindung

CPU-Karte

Hier sind der Z80-Prozessor, das ROM, ein Teil des RAM und die meisten anderen für den Rechnerbetrieb unerlässlichen Komponenten untergebracht.

HRG-Karte

Mit dieser Karte wird der 380Z voll farbgrafikfähig.

Netzteil

Die Stromversorgung ist ungewöhnlich groß und schwer ausgelegt. Sie läßt sich praktisch nicht überlasten.

Reset-Taste

Schaltenschloß

Der Rechner kann nach außen verriegelt werden, um jeden unbefugten Eingriff zu unterbinden.

Ventilator

Da das Netzteil dafür eingerichtet ist, ohne Modifikation noch mehrere Zusatzkarten zu versorgen, ist ein Ventilator zur Unterstützung der Kühlung nötig.

Zwei Diskettenlaufwerke

Research Machines 380Z

ABMESSUNGEN:

595 × 425 × 215 mm

ZENTRALEINHEIT:

Z80

TAKTFREQUENZ:

4 MHz

SPEICHER:

Bis zu 6 KByte ROM und 64 KByte RAM

BILDSCHIRM-DARSTELLUNG:

24 Zeilen mit 40 oder 80 Zeichen, sieben Farben in max. 255 Helligkeitsstufen, Grafikauflösung 160 × 96 und 320 × 192 Punkte

SCHNITTSTELLEN:

RS 232 seriell, Cassettenrecorder, Paralleldrucker

MITGELIEFERTE SPRACHE:

Erweitertes Research Machines-BASIC

WEITERE SPRACHEN:

ALGOL, FORTRAN und CP/M-Standardsprachen

LIEFERUMFANG:

Handbuch für Inbetriebnahme, CP/M, Diskettenlaufwerke und Cassettenbetrieb sowie BASIC-Dienstprogramme auf Diskette

TASTATUR:

60 Schreibmaschinentasten, für Textverarbeitung geeignet

DOKUMENTATION:

Hervorragend, wenn auch reichlich trocken.

Schlüsselwörter wohlbekannte Funktionen.

Der Basic-Interpreter und das HRG-Paket vereinnahmen gemeinsam so viel RAM-Kapazität, daß für größere Benutzerprogramme zu wenig Raum bleibt. Deshalb kann der User zwischen drei verschiedenen BASIC-Versionen wählen – ganz ohne Grafik, mit einem Teil der HRG-Software oder alles komplett.

Der Einsatz der HRG-Routinen ist nicht auf BASIC beschränkt, sondern sie sind als normale Maschinencode-Files auch in jeder anderen Sprache aufrufbar. Da der 380Z unter CP/M läuft, ist eine ganze Reihe von Programmiersprachen verfügbar. Höchst ungewöhnlich für einen Microcomputer ist, daß es auch einen ALGOL-Compiler gibt.

ALGOL war in der Vergangenheit die bevorzugte Sprache im europäischen Hochschulbereich, während in den USA vorwiegend FORTRAN verwendet wurde. ALGOL zeichnet sich ähnlich wie PASCAL durch eine strenge logische Struktur aus und eignet sich besonders für die numerische Mathematik.

In der Bibliothek

Wir beenden unsere C-Serie mit einem kurzen Blick auf die Schnittstelle zwischen C und dem Betriebssystem Unix.

Die Sprache C wird hauptsächlich zur Entwicklung von Systemsoftware, Betriebssystemen, Dienstprogrammen und schnellen Anwendungsprogrammen eingesetzt. Für diesen Zweck muß zwischen der Sprache und dem zugehörigen Betriebssystem eine Schnittstelle existieren, über die auch Systemfunktionen programmierbar sind.

Speziell die Funktion `sizeof(objekt)` unterstützt die Entwicklung von Code, der auf unterschiedlichen Systemen laufen soll. `sizeof(objekt)` wird mit einem Variablennamen oder einem Typennamen geladen und liefert die Anzahl Bytes, die zur Speicherung der Variablen nötig sind. Mit Hilfe von `sizeof(objekt)` kann der Programmcode unabhängig davon funktionieren, ob `int` 16 oder 32 Bits lang ist. `sizeof(objekt)` wird häufig mit den Speicherfunktionen `calloc()`, `malloc()`, `realloc()` und `free()` eingesetzt.

● `calloc()`:

```
char *calloc(anzahl,laenge)
int anzahl,laenge
```

`calloc()` reserviert für `anzahl` Variablen einen Speicherbereich (`laenge` gibt die Bytezahl der Variablen an), initialisiert ihn auf Null und liefert einen Pointer auf das erste Datenelement (oder Null, wenn zu wenig Speicher vorhanden ist). Der folgende Befehl reserviert Platz für 50 Ganzzahlen:

```
p=calloc(50,sizeof(int));
```

● `malloc`

```
char *malloc(bytes)
unsigned bytes;
```

`malloc()` ähnelt `calloc()`, initialisiert die Bytes jedoch nicht. Beide Funktionen liefern einen Pointerwert auf `char`, da `char` nur ein Byte lang ist. Wenn er auf andere Datentypen zeigen soll, muß er umgewandelt werden.

● `realloc`

```
char *realloc(p,bytes)
char *p;
int bytes;
```

`realloc()` ändert die Größe des Speicherbereichs (auf den `p` zeigt) in die mit `bytes` angegebene Bytezahl und kopiert den Inhalt des alten Bereichs in den neuen Bereich. Da der Speicherbereich sich dabei verschieben kann, sind Pointer auf den alten Bereich nicht mehr gültig.

● `free()`:

```
free(p)
char *p;
```

Diese Funktion gibt einen Speicherbereich frei, der mit `calloc`, `malloc` oder `realloc` reserviert wurde. Der Einsatz anderer Pointerwerte kann unerwünschte Folgen haben.

Eine weitere Kategorie von C-Funktionen steuert die Fehlerbehandlung. In C-Programmen treten die meisten Fehler bei der Ein- und Ausgabe oder beim Aufruf von Bibliotheksfunktionen auf. Die Bibliothek enthält die Standardvariable `errno` vom Typ `int` und eine Liste von Fehlermeldungen namens `sys_err_list[]`, die vom Programm aus direkt angesprochen werden kann. Wenn beim Ablauf von Bibliotheksfunktionen Fehler eintreten, wird `errno` mit einer Zahl geladen. Der Befehl

```
perror(s)
```

```
char *s="Fehlermeldung";
```

gibt auf den Datenstrom `stderr` je nach Inhalt von `errno` die Fehlermeldung des Systems und der Bibliothek aus.

Da bei Ein- und Ausgabebefehlen nicht immer eine normale Fehlermeldung des Systems ausgegeben wird, testet `int ferror(file_pointer)`, ob der Fehler beim Schreiben oder Lesen der angegebenen Datei aufgetreten ist. Die Funktion liefert Null (Fehler) oder ungleich Null (kein Fehler). Da Fehler den Dateizugang behindern können, läßt sich die Fehleranzeige auch mit der Funktion `clearerr(file_pointer)` löschen.

Eine Reihe weiterer Funktionen verwalten Diskettendateien und Directories. Das Handbuch des C-Compilers liefert hier alle nötigen Einzelheiten. Darunter gibt es mit Sicherheit den Befehl `access()`, der den Zugriffsmodus der angegebenen Datei oder des Directories prüft; `chmod()`, der den Zugriffsmodus ändert und `chdir()`, der ein anderes Directory aufruft.

Unser Programmbeispiel enthält alle Fähigkeiten von C, die wir bisher besprochen haben. Einige davon wurden zwar nur kurz erwähnt (z. B. rekursive Funktionsaufrufe), andere jedoch (z. B. `malloc`) ausführlich behandelt. Das Programm erstellt den Index einer langen Textdatei – das heißt eine Liste aller Wörter mit den entsprechenden Seitennummern. Die Wörter sind in einer verketteten Liste gespeichert, die mit jedem neuen Wort wächst. Jedem Wort ist ein Pointer zugeordnet, der auf eine weitere verkettete Liste mit den Seitennummern zeigt.

C ist eine der wenigen Sprachen, in der auf kleinen Micros entwickelte Programme auch auf großen Micros, Minis und sogar den Geräten der Groß-EDV laufen.

Indexaufbau

```

/* In der Datei index.h */
#define NULL 0
#define MAXWORDSIZE 20
typedef char ENTRY [MAXWORDSIZE];
/* Jedes Element des Indizes enthält den direkten
Eintrag, einen Pointer auf die Liste der Seiten-
nummern und einen Pointer auf das nächste
Datenelement */
struct page__number
{
    int pn;
    struct page__number *pNext;
}
typedef struct page__number page;
typedef page *plink;
struct index__element
{
    ENTRY entry;
    plink pages;
    struct index__element *exit;
}
typedef struct index__element element;
typedef element *link;
/* Wir können uns nun mit 'element' auf ein
Listenelement beziehen. Ein Pointer auf ein Ele-
ment wird 'link' genannt. */

/* In der Datei index.c */
#include <stdio.h>
#include <string.h>
#include <index.h>
#define LPP = 66; /* lines per page */
main(argc,argv)
int argc;
char *argv[];
link head;
{
    FILE infile;
    int lc = 0, pc = 1, inword = 0;
    ENTRY nextword;
    char *nw;

/* Liste der Einfachheit halber mit einem ersten
Dummyeintrag füllen */
    head = new__entry ("",NULL, 0);
/* Dateinamen in den Index eintragen */
    if(argc != 2)
    {
        fprintf(stderr, "\nusage is %s filename\n",
            *argv);
        exit(1);
    }
    if (infile = fopen (*++argv, "r") == NULL)
        fprintf(stderr, "\nfile not found %s\n",
            *argv);
        exit(1);
    }
    nw = nextword;
    while (c = getc(infile) != EOF)
    {
        if(c == '\n')
            /* Eins zu Zähler addieren, prüfen, ob Seitenende */
            {
                ic += 1;

```

```

                if (lc > LPP)
                {
                    pc += 1;
                    lc = 0;
                }
            }
        else if (inword)
        {
            if (isalpha(c))
                *nw++ = c;
            else
            {
                inword = 0;
                *nw = '\0';
                insert (nextword, head, pc);
                nw = nextword;
            }
        }
        else
        {
            if (isalpha(c))
            {
                inword = 1;
                *nw++ = c;
            }
        }
    }
    if(inword)
    {
        *nw = '\0';
        insert (nextword, head, pc);
    }
    display__index;
}
insert(e)
ENTRY e;
{
    insert(e,l,pnum)
    ENTRY e;
    link l;
    int pnum;
    static link lastl;
    {
        lastl = head;
        if(l == NULL)
        {
            lastl->next = new__entry (e,l,pnum);
            return;
        }
        else
        {
            int s;
            s = strcmp(e,l->entry);
            if(s == 0)
                /* Wort ist vorhanden, Seitennummer hinzufügen */
                {
                    add__page__number (pnum,
                        l->pages);
                    return;
                }
            else if(s > 0)

```

```

                ein neues einsetzen */
                {
                    lastl->next = new__entry (e,l,pnum);
                    return;
                }
            }
        }
    }
    /* noch nicht gefunden, daher mit einem rekur-
siven Modul Liste für das Einsetzen durchsuchen */
    lastl = l;
    insert(e,l->next, pnum);
    return;
}
}
link new__entry(e,l,pnum)
ENTRY e;
link l;
int pnum;
{
    /* mit malloc Platz für Eintrag schaffen */
    link newl;
    newl = (link) malloc (sizeof (element));
    /* Umwandlung des von malloc gelieferten char
Pointers */
    newl->entry = e;
    newl->next = l;
    newl->pages = (plink) malloc (sizeof (page));
    newl->pages->pn = pnum;
    newl->pages->pnext = NULL;
    return(newl);
}
add__page__number (pnum,pl)
int pnum;
plink pl;
{
    /* Ende der Liste der Seitennummern finden */
    while (pl->pnext) != NULL)
        pl = pl->pnext;
    pl->pnext = (plink) malloc (sizeof (page));
    pl->pnext->pnext = NULL;
    pl->pnext->pn = pnum;
    return;
}
display__index ;
link l;
plink pl;
{
    l = head->next;
    while(l != NULL)
    {
        printf ("%s\t", l->entry) ;
        pl = l->pages;
        while (pl->next != NULL)
        {
            printf ("%d4,", pl->pn);
            pl = pl->next;
        }
        printf ("%d\n", pl->pn);
    }
    return;
}
}
/* zu weit, daher nach dem letzten Datenelement

```

Letzte Feinheiten

Wir beenden unser Go-Programmprojekt mit einigen Überlegungen zu Ergänzungen, mit denen die Antwortzeit des Computers verkürzt werden kann, und fügen Speicherroutinen an.

Spectrum-, Commodore 64- und Schneider-Besitzer haben sicher bemerkt, daß ihre Programme Modifikationen der Acorn-B-Version sind. Eine Überarbeitung, die Verwendung kurzer Variablenamen und das Ausnutzen der Vorteile des jeweiligen Computers könnten das Spiel erheblich verbessern.

Die Spectrum- und C64-Versionen können wegen ihres langsameren BASIC natürlich nicht so schnell sein wie die Acorn B-Version. Allerdings können einige Routinen in Maschinencode umgewandelt werden, um die Ausführung zu beschleunigen. Die Häufigkeit, in der Routinen ausgeführt werden, kann durch Setzen einer Zählvariablen am Anfang jeder Routine ermittelt werden. Eine der kritischsten Routinen ist PROCsearch. Sie wird nicht nur während der „PROCgroup evaluation“ verwendet, sondern auch mindestens einmal pro Aufruf von FNlegality. Da das Spielbrett (board%) als Bytesequenz vorliegt, ist eine Konvertierung in Maschinensprache sehr einfach.

Es gibt noch viele andere Taktiken bei Go, die die Spielstärke des Programms verbessern könnten. Hierzu zählt beispielsweise Joseki (Eröffnungsspiel). Ein wichtiges Detail, das implementiert werden könnte, ist das von „Leben und Tod“. Das Programm ermittelt bereits einen „unkonditionalen Tod“: Es überprüft, ob eine bestimmte Steingruppe keine Freifelder mehr hat. Geht man hier einen Schritt weiter, kann man bestimmen, ob eine Gruppe „unkonditionales Leben“ hat. Zu diesem Zweck sind zwei Überprüfungen vorzunehmen.

Geschickte Tests

1. Alle Freifelder der Gruppe sind an den Stellen zur Probe mit gegnerischen Steinen zu besetzen, wo es legal ist.
2. Mit PROCsearch sind die verbleibenden Freifelder der Gruppe zu zählen. Beträgt ihre Zahl (clib%) zwei oder mehr, kann die Gruppe nicht eingenommen werden.

Es sollte beachtet werden, daß beim Füllen der Freifelder alle Möglichkeiten bedacht werden müssen. Füllt man sie der Reihe nach, werden eventuell falsche Augen nicht besetzt. Eine Gruppe kann zum Beispiel ein internes Freifeld haben, das nur besetzt werden kann, wenn alle externen Felder belegt sind. Anschließend können vielleicht wieder andere Felder besetzt werden.

Funktioniert der Vorgang für eine Gruppe, so kann auch das ganze Brett untersucht werden. Dadurch wird möglich, daß sich Gruppen Augen teilen. Ein Beispiel hierzu finden Sie im ersten Teil dieser Artikelreihe. Für die Implementierung müssen Sie zuerst die obenstehende Prozedur nachvollziehen. Wird ein internes Freifeld gefunden, das auf mindestens einer Seite von einer gleichfarbigen anderen Gruppe umschlossen wird, muß sich die Routine rekursiv selbst aufrufen, um die Sicherheit der angrenzenden Gruppe zu prüfen, usw. Nur wenn die angrenzenden Gruppen vor einem Angriff sicher sind, ist das interne Auge der ersten Gruppe außer Gefahr.

Das Spielbrett wirft aufgrund seiner Größe Probleme auf. Ein Brett mit 19×19 Schnittpunkten bietet 3³⁶¹ Konfigurationsmöglichkeiten freier schwarzer oder weißer Felder auf dem Brett, was in Zehnerpotenzen ausge-

Gut gespeichert

Um ein Spiel auf Cassette oder Diskette zu speichern und es zu einem späteren Zeitpunkt fortzusetzen, ist eine neue Routine PROCsave game erforderlich, die nach Eingabe von QUIT aufgerufen werden muß. Die Save-Prozedur sollte alle Status-Variablen und Byte-Tabellen speichern. Hierzu zählen:

board% bis board%+255	Hauptbrett-Bytes
weight% bis weight%+255	Werte der Wichtungstabelle, die durch PROCremove geändert werden können
move%	Gibt nicht nur die Zugnummer an, sondern auch, welcher Spieler an der Reihe ist
capture%(2)	Anzahl der von Schwarz oder Weiß eingenommenen Steine
ko%	Gibt die Position eines ko-Punktes an oder aber Null

Zusätzlich können Sie speichern:

ataril\$ und atari2\$	Gibt eine „ATARI“-Warnung von Schwarz oder Weiß an
T\$	Wahlmethode des letzten Zugs von Schwarz

drückt etwa 10^{172} entspricht. Eine grobe Schätzung der durchschnittlichen Anzahl möglicher Züge eines Spielers ergibt 250. Wird die traditionelle Spieltechnik der Baumstruktur in Go implementiert, würde eine intensive Berechnung über drei Züge voraus bereits das Generieren und Vergleichen von mehr als acht Mio. Brettpositionen erfordern. Es ist bekannt, daß aber sogar Amateure ca. 30 Spielzüge vorausdenken können. Daher wird ein Go-Programm stets in der Spielstärke beschränkt sein.

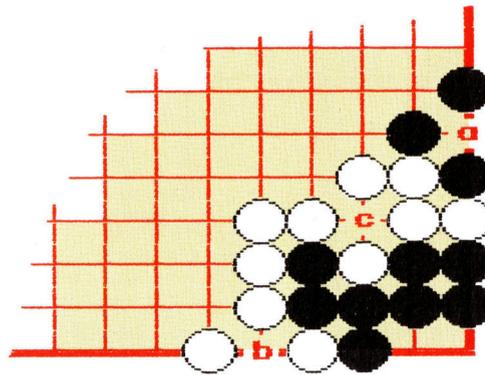
Brennpunkte erkennen

Die Lösung des Problems großer Baumstrukturen ist die Verwendung einer „Brennpunkt“-Funktion. Damit kann zuerst ein grobes Berechnen des gesamten Brettes erfolgen, wobei die kritischen Situationen erkannt werden. Nach Auswählen von einem oder zwei „Schlachtfeldern“ können für sie tiefere Verzweigungen erstellt werden. Die Verzweigungen können durch die Verwendung des Alpha-beta-Algorithmus, sowie über Routinen, die kritische Situationen, Augen-Positionen usw. herausgreifen, nochmals erheblich gekürzt werden.

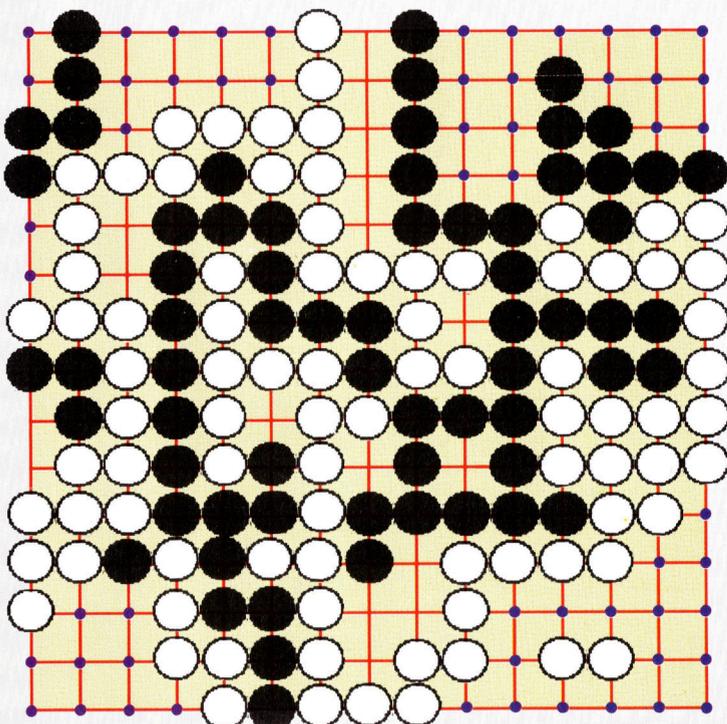
Ein weiteres Problem bei Go ist die Flexibilität einiger Regeln. So zählt als Spielende, wenn kein Spieler mehr einen Vorteil erzielen kann. Dabei haben Go-Programme große Schwierigkeiten, diesen vagen Umstand zu erkennen. Ein weiteres Problem bezieht sich auf die speziellen ko-Bedingungen. Unser Diagramm zeigt eine „dreifache ko-Position“.

Schwarz spielt auf „b“ und besiegt den weißen Stein rechts. Weiß nimmt nun den schwarzen Stein direkt unter „a“ ein, worauf Schwarz Weiß bei „c“ überlistet. Weiß kann nun an die Position des ersten Verlustes zurückgehen und den schwarzen Stein auf „a“ entfernen. Schwarz entfernt den weißen Stein auf „a“ usw. Während eines normalen Spiels wird, falls keiner der Spieler diesen „Ortskampf“ aufgibt, das Spiel beendet. Zudem können noch erheblich komplexere ko-Situationen eintreten. Um sie zu erkennen, muß das Go-Programm alle vorhergegangenen Spielpositionen speichern und sie mit der aktuellen Position vergleichen.

Derzeit ist kein Go-Programm perfekt. Vor einigen Jahren schien es jedoch auch noch unvorstellbar, daß es Computer zu akzeptablen Leistungen im Schach bringen könnten. In Japan jedenfalls wird Go derzeit eingesetzt, um die Leistungsfähigkeit der 5. Computergeneration zu testen.



Abschließende Bewertung des Spieles



Die weiße Gruppe rechts unten hat 17 Punkte des Gebietes eingeschlossen. Die beiden weißen Steine innerhalb des Gebietes sind nicht mitgerechnet – es werden nur die freien Stellen gezählt.

Die schwarze Gruppe oben rechts besteht tatsächlich aus zwei Gruppen, da nur eine diagonale Verbindung zwischen den beiden Hälften besteht. Trotzdem sind alle Ausgänge des Gebietes durch schwarze Steine besetzt, so daß Schwarz 17 Punkte erhält.

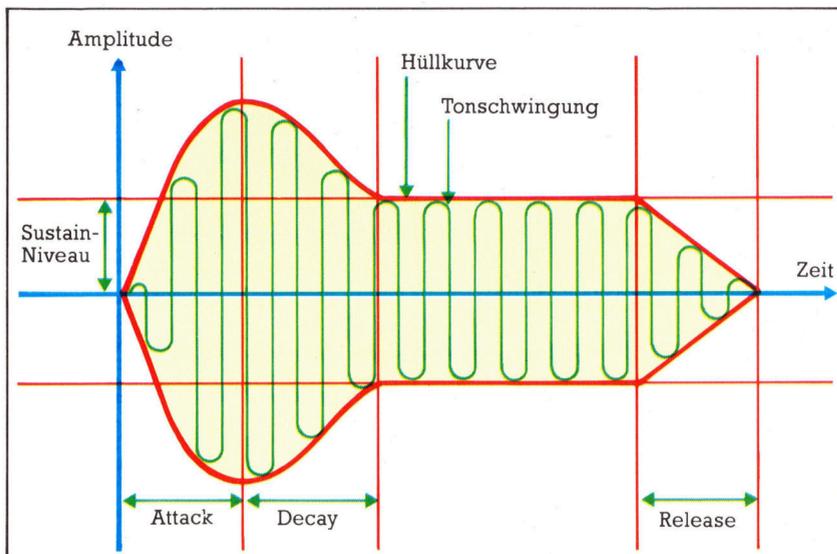
Weiß hat die untere linke Ecke eingeschlossen, doch befindet sich dort noch ein vereinsamer schwarzer Stein. Die Go-Regeln besagen, daß dieser Stein während des Spieles nicht unbedingt eingenommen werden muß. Dafür wird er jetzt entfernt und hinterläßt zehn Felder für Weiß. Außerdem erhält Weiß noch einen Extra-Punkt für den auf diese Art entfernten schwarzen Stein.

Schwarz scheint die obere linke Ecke umzingelt zu haben, ist jedoch selber durch eine größere weiße Gruppe eingeschlossen. Daher kann die schwarze Gruppe vom Brett genommen werden, und Weiß erntet 18 Gebietspunkte sowie weitere fünf Punkte für die besiegten schwarzen Steine. Alle anderen freien Felder sind neutral, da sie keine Gebiete repräsentieren, die von nur einer Farbe eingeschlossen werden.



Trommelwirbel

Der SID-Chip (Sound Interface Device) erzeugt die Klänge des Commodore 64 und ist eigentlich ein eigener Synthesizer, der mit entsprechender Software sogar roboterähnliche Sprache erzeugen kann und über enorme Klangfähigkeiten verfügt.



Der Charakter eines Tones – das heißt die Gruppe von Merkmalen, mit der wir beispielsweise zwischen einem Klavier und einer Violine unterscheiden – hängt von der Form der Hüllkurve ab. In der elektronischen Klangsynthese besteht eine Hüllkurve aus vier klar voneinander getrennten Abschnitten – Attack, Decay, Sustain und Release (ADSR). Die Dauer der einzelnen ADSR-Abschnitte wird mit POKE-Werten in die Register des SID-Chips geladen.

Die vom Commodore 64 erzeugten Töne werden normalerweise über die RF-Buchse direkt zum Fernseher geleitet. Die Klangugabe läßt sich jedoch sehr verbessern, wenn Sie an den Audio/Video-Ausgang ein HiFi-System anschließen. In dieser Folge untersuchen wir, wie sich der Commodore 64 per Software in eine Rhythmusmaschine verwandeln läßt. Bei Anschluß eines HiFi-Systems können dabei erstaunlich realistische Töne entstehen.

Der SID-Chip kann jedoch nicht nur Töne erzeugen, sondern auch Audiosignale entgegennehmen, die von externen elektronischen Geräten oder von Musikinstrumenten (z. B. elektrischen Gitarren) stammen. Das externe Signal läßt sich mit der Audio-Ausgabe des SID-Chips mischen. Seien Sie beim Einsatz von E/A-Schnittstellen jedoch vorsichtig, da falsch angeschlossene Leitungen den Computer beschädigen können. Schlagen Sie vor dem Anschließen im Hardwarehandbuch die korrekten Verbindungen und Voltzahlen nach.

Wir untersuchen zunächst, wie sich Musikklänge aus einzelnen, reinen Tönen zusammensetzen. Danach beschreiben wir eine andere Methode, die die gleiche Wirkung erzielt: Über die Hüllkurve eines Tonablaufs lassen sich Klänge mit unterschiedlichem Charakter erzeugen. Das bedeutet, daß man durch Ändern der Steuerwerte fast jeden Klang hervorrufen kann. Die Steuerwerte heißen auch „ADSR“ (Attack, Decay, Sustain und Release).

Leider eignet sich das BASIC des Commodore 64 nicht besonders zur Steuerung von Klängen. Eine wirksame Tonerzeugung kann daher nur mit vielen PEEK- und POKE-Befehlen erreicht werden.

Klänge erreichen unser Ohr in Form von periodischen Luftschwingungen, wobei die Zahl der Schwingungen pro Sekunde „Tonhöhe“ oder „Tonfrequenz“ genannt wird. Die untere Hörschwelle des menschlichen Ohrs liegt zwar bei 15 Zyklen pro Sekunde (15 Hertz), doch empfinden wir einen reinen Ton von 100 Hertz noch als tief; der Kammerton A über dem mittleren C liegt bei 440 Hz. Durch Frequenzverdoppelung steigt die Tonhöhe um eine Oktave. Der Mensch kann normalerweise etwa 10,5 Oktaven wahrnehmen, von denen der dreistimmige SID-Oszillator etwa acht erzeugen kann – von 0 bis 4000 Hz.

Der französische Physiker Jean Fourier (1768 bis 1830) entdeckte als erster, daß jede periodische Wellenform aus einer reinen Grundschwingung und Obertönen besteht. Die Obertonfrequenzen sind ganzzahlige Vielfache der Grundschwingung und werden „Harmonische“ genannt. Der charakteristische Klang eines Instrumentes wird vom Lautstärkenverhältnis der Obertöne (oder „Harmonischen“) zueinander bestimmt.

Reine Töne entsprechen reinen Sinusschwingungen, die sich – als Analogsignal – nicht leicht auf digitalen Geräten mit nur 0 oder 5 Volt erzeugen lassen. Statt reine Frequenzen zu generieren und sie zu periodischen Klängen zusammenzumischen, arbeiten Heimcomputer daher mit einer anderen Technik.

Schwingungsformen

Jede der drei Stimmen des SID-Chips kann je nach Einstellung eine von vier periodischen Schwingungsformen erzeugen:

- 1) Die **Sägezahn-schwingung** enthält alle Harmonischen. Die Intensität der Nten Harmonischen entspricht proportional $1/N$.
- 2) Die **Dreieck-schwingung** enthält nur die ungeraden Harmonischen. Die Intensität der Harmonischen N entspricht hier proportional $1/N^2$.
- 3) Eine **Rechteck-schwingung** enthält ungerade Harmonische proportional zu $1/N$. Durch Änderung der „Impulsbreite“ lassen sich unterschiedliche Rechteckwellen mit eigener har-



monischer Mischung erzeugen.

4) **Das weiße Rauschen** ist eine zufällige Frequenzmischung, die oft für Spezialeffekte eingesetzt wird.

Der harmonische Inhalt kann weiterhin noch durch die drei Filter des SID-Chip (Low-Pass, Band-Pass und High-Pass) verändert werden. Ein Low-Pass-Filter läßt beispielsweise alle Frequenzen unter einem bestimmten Wert passieren, schwächt aber alle höheren Frequenzen ab. Mit diesen Möglichkeiten und der Steuerung der ADSR-Hüllkurve läßt sich fast jeder Klang erzeugen.

Auch die Sprachsynthese kann mittels mehrerer Techniken erzeugt werden. Dabei werden die Grundelemente der Sprache – sogenannte „Phoneme“ – in einem Array von ADSR-Werten gespeichert. Da die deutsche Sprache nur etwa 52 verschiedene Phoneme besitzt, ist die Codierung nicht zu umfangreich. Mit einem Maschinencodeprogramm werden die ASC-Texte (die Commodoreversion von ASCII) in eine Folge von Phonemcodes übersetzt und dann über das ADSR-Array an den SID-Chip gesandt. Das hört sich einfacher an, als es ist, da die Regeln zur Übersetzung von Text in Phoneme recht kompliziert sind. Die Sprachqualität hängt wesentlich von der Arbeit dieses Programmteils ab. Da sich die Sprachsynthese auf dem Commodore 64 gut einsetzen läßt, gibt es schon eine Reihe kommerzieller Produkte, die mit dieser Technik arbeiten.

Klangentwicklung

Unser ADSR-Diagramm zeigt die allgemeine Form eines Klanges und betont dabei die Teile, die der SID-Chip über ein Programm steuern kann. Die vier ADSR-Faktoren sind:

- 1) **Attack:** Die Anschlagslautstärke
- 2) **Decay:** Das Abklingen eines Tons auf ein konstantes Niveau
- 3) **Sustain:** Die Lautstärke auf konstantem Niveau

Wert	Attack-Rate (Zeit/Zyklus)	Decay/Release-Rate (Zeit/Zyklus)
0	2 ms	6 ms
1	8 ms	24 ms
2	16 ms	48 ms
3	24 ms	72 ms
4	38 ms	114 ms
5	56 ms	168 ms
6	68 ms	204 ms
7	80 ms	240 ms
8	100 ms	300 ms
9	250 ms	750 ms
10	500 ms	1.5 secs
11	800 ms	2.4 secs
12	1 sec	3 secs
13	3 secs	9 secs
14	5 secs	15 secs
15	8 secs	24 secs

4) **Release:** Das Ausklingen des Tones vom Sustainniveau auf Null.

Die Abschnitte von Attack, Decay und Release werden mit Nybbles (vier Bits) der SID-Register gesteuert. Jeder Parameter kann daher Werte von 0 bis 15 annehmen, die mit POKE in die SID-Register gestellt werden. Unsere Tabelle links unten veranschaulicht die Dauer der Abläufe.

Die Dauer des Sustain-Abschnitts wird durch eine Verzögerungsschleife abhängig vom Notenwert bestimmt. In unserer zweiten Tabelle haben wir die für eine Violine typischen ADR-Werte aufgeführt. Der Commodore 64 erzeugt seine Klänge in sechs Schritten.

ADSR	Zeit	POKE-Wert
A	500ms	10
D	300ms	8
S	—	—
R	750ms	9

Hier ein Beispiel, bei dem SID=54272 die Basisadresse des Chips ist.

- Schritt 1: Lautstärke einschalten
POKE SID+24,15
 - Schritt 2: ADSR auswählen, z. B.:
POKE SID+5,9 :REM ATTACK/DECAY
STIMME #1
POKE SID+6,0 :REM SUSTAIN/RELEASE
STIMME #1
 - Schritt 3: Für jeden Oszillator die Frequenz wählen, z. B.:
POKE SID+1,25 :REM HI-BYTE FREQUENZ
STIMME #1
POKE SID,0 :REM LO-BYTE FREQUENZ
STIMME #1
 - Schritt 4: Wellenform wählen, z. B.:
POKE SID+4,33 :REM SÄGEZAHN FÜR
STIMME #1
- An diesem Punkt fängt der Ton an zu klingen – im Fachjargon: „Es öffnet sich das Gatter“.
- Schritt 5: Eine Verzögerungsschleife für die Dauer des Tons auf der Sustain-Ebene.
 - Schritt 6: Die Wellenform muß freigegeben werden, z. B.:
POKE SID+4,32 :REM SÄGEZAHN FREI-
GEBEN

Eine Melodie können Sie auf dem Commodore 64 am einfachsten spielen, wenn Sie die Anfangswerte von ADSR setzen und mit einer FOR...NEXT-Schleife die hi/lo-Bytes der Frequenzen mit READ aus DATA-Anweisungen lesen. Wenn Sie in die DATA-Befehle Nullen einsetzen, verändert sich das Tempo der verschiedenen Stimmen, während Delay hingegen konstant bleibt.



Quellentext in Assembler

```

;*****
;*****
; **
; **  CBM DRUM MACHINE **
; **  SOURCE CODE    **
; **
;*****
;*****
;
VOL      = $D418 ;SID VOLUME
ATT1     = $D405 ;SID ATTACK VOICE 1
SUS1     = $D406 ;SID SUSTAIN VOICE 1
PULSE    = $D402 ;SID PULSE RATE VOICE 1
WAVE1    = $D404 ;SID WAVEFORM VOICE 1
BASS     = $D401 ;SID HI-BYTE FREQ VOICE 1
ATT2     = $D40C ;SID ATTACK VOICE 1
SUS2     = $D40D ;SID SUSTAIN VOICE 2
WAVE2    = $D40B ;SID WAVEFORM VOICE 2
SNARE    = $D408 ;SID HI-BYTE FREQ VOICE 2
ATT3     = $D413 ;SID ATTACK VOICE 3
SUS3     = $D414 ;SID SUSTAIN VOICE 3
WAVE3    = $D412 ;SID WAVEFORM VOICE 3
BELL     = $D40F ;SID HI-BYTE FREQ VOICE 3
ROW1     = $C350 ;STORAGE FOR VOICE 1
ROW2     = $C360 ;STORAGE FOR VOICE 2
ROW3     = $C370 ;STORAGE FOR VOICE 3
TEMPO    = $02A7 ;TEMPORARY STORE FOR DELAY
XCOUNT   = $02A8 ;TEMPORARY STORE FOR X REGISTER
YCOUNT  = $02A9 ;TEMPORARY STORE FOR Y REGISTER
LOVEC    = $FB   ;STORE FOR LO-BYTE VECTOR
HIVEC    = $FC   ;STORE FOR HI-BYTE VECTOR
PLAY     = $FD   ;TOGGLE FOR PLAY DRUM (1=YES)
DELAY    = $FE   ;STORE FOR CURRENT DELAY STATUS
KEY      = $C5   ;CURRENT KEY PRESSED
;
*        = $C000 ;ASSEMBLE FROM 49152 (DECIMAL)
;-----
;SET UP WEDGE
;-----
SEI      ;DISABLE INTERRUPT REQUEST
LDA $0314 ;GET CONTENTS OF VECTOR LO-BYTE
STA LOVEC ;STORE IN LOVEC
LDA $0315 ;GET CONTENTS OF VECTOR HI-BYTE
STA HIVEC ;STORE IN HIVEC
LDA #WEDGE ;GET LO-BYTE WEDGE START ADDR.
STA $0314 ;STORE IN IRQ VECTOR LO-BYTE
LDA #>WEDGE ;GET HI-BYTE OF WEDGE START
STA $0315 ;STORE IN IRQ VECTOR HI-BYTE
CLI      ;RESUME INTERRUPT REQUEST
RTS     ;RETURN
;-----
;REMOVE WEDGE
;-----
SEI      ;DISABLE INTERRUPT REQUEST
LDA LOVEC ;GET ORIGINAL VALUE FROM LOVEC
STA $0314 ;STORE IN IRQ VECTOR LO-BYTE
LDA HIVEC ;GET ORIGINAL VALUE FROM HIVEC
STA $0315 ;STORE IN IRQ VECTOR HI-BYTE
CLI      ;RESUME INTERRUPT REQUEST
RTS     ;RETURN
;-----
;MAIN LOOP
;-----
WEDGE JSR REG ;GOSUB REG
LDA KEY ;WHICH KEY PRESSED ?
CMP #$03 ;IS IT FUNCTION KEY #1 ?
BNE CONT1 ;BRANCH IF NOT TRUE
JSR FLAG0 ;GOSUB FLAG0
CONT1 LDA KEY ;WHICH KEY PRESSED ?
CMP #$04 ;IS IT FUNCTION KEY #7 ?
BNE CONT2 ;BRANCH IF NOT TRUE
JSR FLAG1 ;GOSUB FLAG1
CONT2 LDA KEY ;WHICH KEY PRESSED ?
CMP #$05 ;IS IT FUNCTION KEY #3 ?
BNE CONT3 ;BRANCH IF NOT TRUE
JSR ADD ;GOSUB ADD
CONT3 LDA KEY ;WHICH KEY PRESSED ?
CMP #$06 ;IS IT FUNCTION KEY #5 ?
BNE CONT4 ;BRANCH IF NOT TRUE
JSR MINUS ;GOSUB MINUS
CONT4 JSR REST ;GOSUB REST
STX XCOUNT ;STORE VALUE OF X REGISTER
STY YCOUNT ;STORE VALUE OF Y REGISTER
JMP $EA31 ;JUMP BACK TO INTERRUPT
;-----
;CHECK ROUTINE
;-----
REST LDA PLAY ;GET TOGGLE VALUE
BNE BEGIN ;IF EQUAL TO 1 THEN BRANCH

```

```

RTS ;RETURN
;-----
BEGIN DEC DELAY ;DECREMENT DELAY
BEQ START ;BRANCH IF ZERO
RTS ;RETURN
;-----
START JSR COUNT ;GOSUB COUNT
CPX #$10 ;END OF LOOP ?
BNE CHECK ;BRANCH IF NOT 0
JSR RESET ;GOSUB RESET
;-----
CHECK LDA ROW1,Y ;GET VALUE OF ROW1 OFFSET BY Y
BEQ NEXT1 ;BRANCH IF ZERO
JSR DRUM1 ;GOSUB DRUM1
NEXT1 LDA ROW2,Y ;GET VALUE OF ROW2 OFFSET BY Y
BEQ NEXT2 ;BRANCH IF ZERO
JSR DRUM2 ;GOSUB DRUM2
NEXT2 LDA ROW3,Y ;GET VALUE OF ROW3 OFFSET BY Y
BEQ NEXT3 ;BRANCH IF ZERO
JSR DRUM3 ;GOSUB DRUM3
NEXT3 INY ;INCREMENT OFFSET
INX ;INCREMENT LOOP COUNTER
RTS ;RETURN
;-----
;SUBROUTINES
;-----
FLAG0 LDA #$00 ;STORE ZERO -
STA PLAY ;IN PLAY
RTS ;RETURN
;-----
FLAG1 LDA #$01 ;STORE 1 -
STA PLAY ;IN PLAY
RESET LDX #$00 ;RESET X REGISTER
LDY #$00 ;RESET Y REGISTER
RTS ;RETURN
;-----
COUNT LDA TEMPO ;GET VALUE OF TEMPO
STA DELAY ;STORE IN DELAY
RTS ;RETURN
;-----
ADD LDA TEMPO ;GET VALUE OF TEMPO
CMP #$FF ;COMPARE RESULT WITH 255
BEQ CONT5 ;BRANCH IF TRUE
INC TEMPO ;INCREMENT TEMPO
CONT5 RTS ;RETURN
;-----
MINUS LDA TEMPO ;GET VALUE OF TEMPO
CMP #$01 ;COMPARE RESULT WITH 1
BEQ CONT6 ;BRANCH IF TRUE
DEC TEMPO ;DECREMENT TEMPO
CONT6 RTS ;RETURN
;-----
REG LDX XCOUNT ;STORE VALUE OF XCOUNT IN X REG
LDY YCOUNT ;STORE VALUE OF YCOUNT IN Y REG
RTS ;RETURN
;-----
;DRUM SOUND ROUTINES
;-----
DRUM1 LDA #$0E ;SET
STA SUS1 ;
LDA #$20 ;UP
STA PULSE ;
LDA #$42 ;BASS
STA WAVE1 ;
LDA #$03 ;DRUM
STA BASS ;
LDA #$41 ;AND PLAY
STA WAVE1 ;
RTS ;RETURN
;-----
DRUM2 LDA #$07 ;SET
STA ATT2 ;
LDA #$0C ;UP
STA SUS2 ;
LDA #$80 ;SNARE
STA WAVE2 ;
LDA #$41 ;DRUM
STA SNARE ;
LDA #$81 ;AND PLAY
STA WAVE2 ;
RTS ;RETURN
;-----
DRUM3 LDA #$02 ;SET
STA ATT3 ;
LDA #$0D ;UP
STA SUS3 ;
LDA #$12 ;COW
STA WAVE3 ;
LDA #$64 ;BELL
STA BELL ;
LDA #$11 ;AND PLAY
STA WAVE3 ;
RTS ;RETURN
.END

```



Wurzelbehandlung

Hierarchische Inhaltsverzeichnisse sind besonders bei großen Dateimengen sehr praktisch. Wir untersuchen die MS-DOS-Kommandos für Inhaltsverzeichnisse und außerdem den Aufbau weiterer, sehr nützlicher Dienstmodule.

Bisher wurden die integrierten Befehle behandelt, die in allen MS-DOS-Versionen zur Verfügung stehen. Ab Version 2.0 besitzt MS-DOS jedoch eine Reihe neuer Fähigkeiten, die unter anderem auf Unix zurückgehen. Die hierarchische Struktur der Inhaltsverzeichnisse ist dabei mit Abstand die wichtigste Neuerung. Schon auf den modernen Disketten mit hoher Datendichte zeigen sich die Vorteile; Festplatten können ohne diese Struktur kaum noch verwaltet werden.

Nach dem Booten eines DOS-2-Systems „sieht“ der Anwender auf der aktuellen Diskette eine Reihe von Dateien – zumeist Programme (.COM oder .EXE) oder Datendateien. Microdisketten im Format 3½ Zoll können heute aber durchaus bis zu 720 KBytes enthalten und Festplatten Dutzende von Megabytes. Damit sind große Dateizahlen möglich, die in Sekunden am Auge des Anwenders vorbeiziehen. Da für ihn die meisten Daten uninteressant sind, ist eine Unterteilung des Speicherbereichs in überschaubare „Sub-Directories“ sehr praktisch.

Das Standard-Inhaltsverzeichnis wird „Root“ (Wurzel) genannt. Mit dem Befehl md (Make Directory – Inhaltsverzeichnis anlegen) wird innerhalb der „Root“ ein Sub-Directory eingerichtet.

```
md work
```

baut aus der Root-Directory eine „Datei“ (work) mit den Namen anderer Dateien auf.

Wenn Sie nun in diesem neuen Sub-Directory Dateien oder Programme anlegen wollen, können Sie mit chdir (CHange DIRectory) oder cd work

das Inhaltsverzeichnis wechseln. dir sollte jetzt eigentlich ein leeres Inhaltsverzeichnis von work anzeigen, doch hat MS-DOS dort schon zwei Sub-Directories mit den (etwas seltsamen) Namen . und .. angelegt. Sie zeigen das aktuelle Directory (work) und das Mutterdirectory an. Bei Eingabe von

```
dir..
```

erscheint die Liste aller Dateien des Root-Directories, darunter auch der Eintrag

```
WORK <DIR> 17-05-86 11.21a
```

Nach der Eingabe

```
mkdir today
```

```
chdir today
```

befinden Sie sich in einem Sub-Directory mit dem Mutterdirectory work und der „Großmut-

ter“, in diesem Fall der Root-Directory.

Eine Datei, die nicht im aktuellen Directory liegt, muß mit dem „Pfadnamen“ aufgerufen werden. Programme des Root-Directory würden beispielsweise mit .. \. oder einfach mit \prog bezeichnet werden (der Schrägstrich trennt die Elemente eines Pfadnamens oder dient am Anfang eines Pfadnamens als Kürzel für „Root-Directory“). Die nächsthöhere Ebene (des Sub-Directory work) erreichen Sie mit chdir.. oder – in diesem speziellen Fall – mit cd\work. Wenn Sie jetzt md source eingeben, erhält today das „Schwesterdirectory“ source. Beide sind sozusagen „Töchter“ des aktuellen Directory work.

Der „Directory-Baum“

Wegen seiner Baumstruktur wird dieser Aufbau auch als „Directory-Baum“ bezeichnet. Mit einigen weiteren Zweigen entsteht die Struktur in unserer Abbildung.

Der vollständige Pfadname einer Datei MEMO.DOC im Directory text lautet:

```
\work\source\text memo.doc
```

Mit einem Pfadnamen, der bei der Root (\) anfängt, können Sie von einer beliebigen Position des Baumes aus jede Datei ansprechen. Im Sub-Directory today genügt jedoch:

```
\source\text memo.doc
```

Der Befehl dir zeigt zwar nur das aktuelle Directory an, kann aber auch mit einem anderen Pfad eingesetzt werden:

```
dir b:\pascal\source\newprogs
```

Hier ist der letzte Name keine Datei, sondern ein Directory. Auch die Laufwerksbezeichnung kann verändert werden (in diesem Fall Laufwerk b). Solange die Diskette nicht aus einem Laufwerk genommen wird, „erinnert“ sich das System an alle Sub-Directories, die bisher ausgewählt wurden.

```
copy \*. *b:
```

kopiert daher alle Dateien im Root-Directory der aktuellen Diskette in das Directory der Diskette in Laufwerk b. Jede Diskette kann beim Formatieren oder über LABEL.EXE mit einem Namen versehen werden.

LABEL.EXE und FORMAT.EXE sind zwei Module einer Reihe von MS-DOS-Dienstprogrammen. Sie werden als separate Programme (normalerweise mit der Erweiterung .EXE, manchmal aber auch mit .COM) über ihre Namen auf-



gerufen. Die Formatierung einer Diskette geschieht durch

format b:

Der Befehl **FORMAT** kann zwei durch Schrägstriche getrennte Zusatzangaben enthalten:

s kopiert nach dem Formatieren auch das MS-DOS System

p fragt, mit welchem Label die Diskette versehen werden soll. Es darf nur eine (Root-)Datei dieses Namens geben.

Hierbei wird vorausgesetzt, daß das aktuelle Directory **FORMAT.EXE** enthält. Ist das nicht der Fall, läßt sich mit dem Befehl **path** ein Pfadname definieren:

path A system A utils

Bei einem unbekanntem Befehl sucht das System diese Datei nun auf dem definierten Pfad (oder einer Reihe von Pfaden, die durch Semikolons voneinander getrennt sind).

path

zeigt den aktuellen Suchpfad an. Ist kein Pfad definiert, wird **No path** angezeigt.

Viele MS-DOS-Versionen enthalten externe Befehle, die auf das OEM-System abgestimmt sind. Einige dieser Befehle sind jedoch in fast allen Versionen enthalten. So ermöglicht **PRINT.EXE** das Drucken einer Datei im „Hintergrund“, während der Anwender am Computer weiterarbeitet. Mit dem Zeileneditor **EDLIN.EXE** lassen sich Textdateien ohne zusätzliche Textverarbeitung anlegen. **DISKCOPY.COM** (oder **DCOPY**) kopiert den gesamten Inhalt einer Diskette, und **DISKCOMP** (oder **COMP**) stellt fest, ob die Übertragung fehlerlos abgelaufen ist.

Alle MS-DOS-Dienstmodule, die normalerweise mit zwei Laufwerken arbeiten, sind „intelligent“ genug, um auch mit nur einem Laufwerk zu funktionieren. So lädt **DISKCOPY** die Daten der Quelldiskette in den Speicher und meldet sich, wenn die Disketten getauscht werden müssen. Nachdem die Daten auf die Bestimmungsdiskette geschrieben sind, zeigt eine weitere Meldung an, daß nun wieder die Quelldiskette eingelegt werden soll. Auf diese Weise wird die gesamte Diskette kopiert.

„Versteckte“ Dateien

CHKDSK.EXE prüft Disketten, liefert umfassende Informationen über deren Status und zeigt auch die Existenz von „versteckten“ Dateien an. Der Befehl

chkdsk

testet das aktuelle Laufwerk und stellt einen Bericht mit folgendem Format dar:

```
720666 Bytes Diskettenkapazität
47528 Bytes in zwei versteckten Dateien
2048 Bytes in vier Directories
526386 Bytes in 39 Anwenderdateien
144704 Bytes auf der Diskette frei
```

```
262144 Bytes Speicherkapazität
198726 Bytes frei
```

Praktischerweise zeigt **CHKDSK** auch gleich den Status des Arbeitsspeichers an (die letzten beiden Einträge) und prüft die allgemeine Diskettenkapazität und die „File Allocation Table“ (FAT) der angeschlossenen Geräte. Beim Diskettenwechsel holt sich MS-DOS Systeminformationen von der neuen Diskette und baut eine neue FAT auf.

Vorsicht beim Umstellen

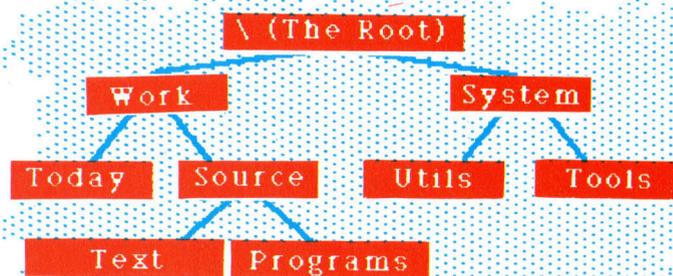
Noch ein wichtiger Hinweis: Prüfen Sie Ihre DOS-Disketten nie mit dem **CHKDSK**-Modul einer anderen MS-DOS-Version: die Diskettensektoren und Directorydaten können dort ein völlig anderes Format haben. Speziell beim parallelen Arbeiten mit DOS 1.x und DOS 2.x ist das gefährlich, weil die Sektoren unterschiedlich angelegt sind. Es kann Stunden dauern, bis Sie eine mit Maschinencodesektoren durchsetzte Textdatei wieder bereinigt haben. Wenn Sie mit der **CHKDSK**-Version des DOS 1 eine unter DOS 2 formatierte Festplatte mit 10 oder 20 MBytes prüfen, sind die gespeicherten Daten kaum mehr zu retten.

Bei der Umstellung eines alten IBM PC oder eines Sanyo von DOS 1 auf DOS 2 sollten Sie daher alle Daten auf formatierten Disketten (ohne System) sichern und vor der Installation des DOS 2 alle Speichermedien neu formatieren. Obwohl die Diskettenmodule der neueren DOS-Versionen meistens prüfen, ob sie mit dem aktuellen System arbeiten können, sollten Sie hier sehr vorsichtig sein.

Die nächste Folge wird sich noch eingehender mit den „versteckten“ Dateien befassen.

Arbeitsteilung

Im Directorysystem von MS-DOS lassen sich die Dateien in Sub-Directories aufteilen. Dabei können die Directories nicht nur Dateinamen, sondern auch weitere Dateiverzeichnisse enthalten. Unser Beispiel zeigt eine Struktur, in der die Datei „Text“ Teil des Directory „Source“ ist, das wiederum zum Directory „Work“ gehört. Im Root-Directory sind alle Dateigruppen der Baumstruktur verzeichnet.





Unter Spannung

Zur Vollständigkeit fehlt unserem Digitalvoltmeter noch die Spannungsversorgung und die Sieben-segment-Anzeige.

Die Anzeige unseres Multimeters setzt sich aus fünf gleichartigen LED-Displays zusammen. Die LEDs müssen nicht gesockelt werden, allerdings ist das Löten an den kleinen Anschlüssen etwas schwierig. In der Bauteilliste sind keine Sockel angegeben, wenn Sie dennoch welche verwenden möchten, müssen Sie daran denken, daß nur 24-polige, 28-polige und 40-polige Sockel den richtigen Pinabstand haben (sieben Löcher bei einem Lochraster von 2,54 mm).

Sie können die LEDs auf eine kleine Extra-Platine montieren und über ein Flachkabel mit der Hauptschaltung verbinden. Das Display

Die LED-Anzeige

Die fünfstellige LED-Anzeige des Multimeters wird auf einer kleinen Extra-Platine aufgebaut. Die sieben Signalleitungen zum Einschalten der Leuchtbalken (a bis f) versorgen die vier rechten Ziffern gemeinsam, da sie im Multiplex-Verfahren angesteuert werden. Die Verbindungen zwischen den LEDs können Sie mit Wire-wrap-Litze herstellen. Mit einem kurzen Stück Flachkabel wird die Anzeigeplatine mit der Hauptschaltung gekoppelt. Die richtigen Anschlußpunkte sind auf Seite 2081 abgebildet.

Die Ziffern-enable-Signale von der Hauptplatine werden jeweils zu Pin 3 der einzelnen LEDs geführt. Wenn Sie die Zeichnung von rechts nach links betrachten, verbinden Sie D1(3) mit D1, D2(3) mit D2, D3(3) mit D3, D4(3) mit D4, schließlich D5(3) mit dem am weitesten rechts liegenden Anschlußpunkt, der mit D5 bezeichnet ist. Die Dezimalpunkt-Kontakte Dn(DP) und TR6(e) erläutern wir in der letzten Folge.

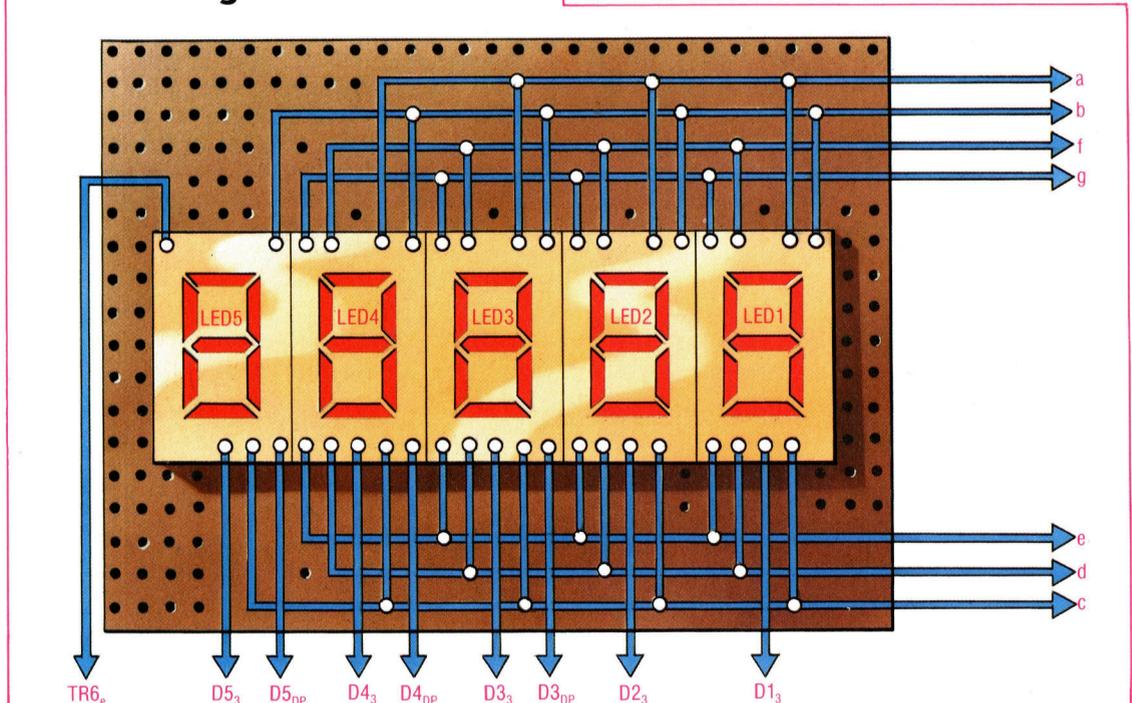
Liste der Bauteile

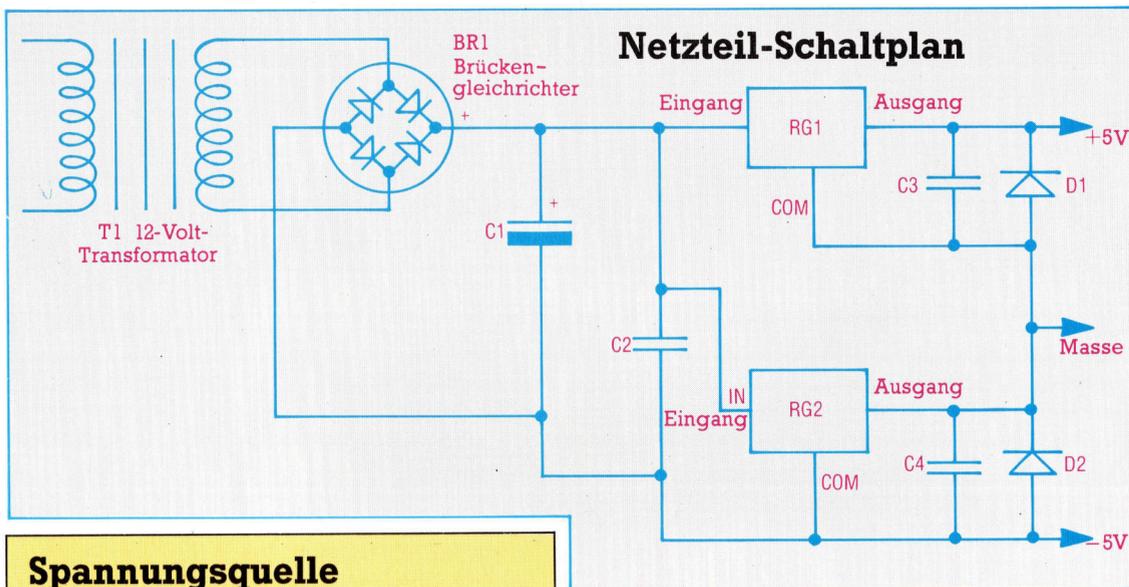
Anzahl	Bezeichnung im Schaltplan	Bauteil
5	LED 1-5	7-Segment LED-Anzeige, gemeinsame Anode
1	T1	Transformator, 220/12 Volt
1	BR1	Brückengleichrichter
1	C1	Elektrolytkondensator, 680 µF, 40 Volt
1	C2	Tantalkondensator, 0,33 µF
2	C3, C4	Tantalkondensator, 0,1 µF
2	RG1, RG2	Festspannungsregler +5 Volt
1	-	Kühlblech
2	D1, D2	Gleichrichterdiode, Typ 1N4001

Kleinmaterial

- 1,5 m Wire-wrap-Litze
- 0,5 m 12-poliges Flachkabel

LED-Anzeige





Spannungsquelle

Das Netzteil für das Multimeter ist recht einfach aufgebaut. Neben einem Brückengleichrichter und zwei Festspannungsreglern zum Umwandeln der 12-Volt Wechselspannung in Gleichspannungen von +5 und -5 Volt sind als Bauteile lediglich Kondensatoren und Dioden vorgesehen.

kann dann an einer beliebigen Stelle im Gehäuse installiert werden. An den fünften Pin der Ziffern displays 3, 4 und 5 werden Leitungen zur Darstellung des Dezimalpunktes angeschlossen.

Wir verwenden ein einfaches Netzteil mit zwei Festspannungsreglern. Dadurch kommen Sie mit einer einzigen Trafowicklung aus und brauchen auch nur einen Brückengleichrichter. Die beiden Dioden zur Entkopplung der beiden Regelschaltungen müssen nicht unbedingt vom angegebenen Typ sein – jede andere Gleichrichterdiode erfüllt den Zweck.

Für C1 läßt sich jeder Elko angemessener Kapazität einsetzen, dessen Grenzspannung über 35 Volt liegt. Die Tantal-Kondensatoren C2, C3 und C4 sind durch Verpolung leicht zu zerstören. Die seitliche Markierung (+) bezeichnet die Plus-Seite der Kondensatoren.

Die Bauelemente des Netzteiles können sehr gut auf einer kleinen Platine mit Löffbahnen aufgebaut werden. RG2 wird nur mit wenigen Milliampere belastet und braucht kein Kühlblech. Anders RG1: Der Regler wird stärker beansprucht und verlangt demnach einen kleinen Kühlkörper. Die Anschlußfahne der Regler ist mit dem Mittelanschluß (Masse) verbunden. Sie sollten darauf achten, daß der (Aluminium-)Kühlkörper nicht mit ungeschützten Leitungen in Berührung kommt.

Wenn das Netzteil zusammengebaut ist, können Sie den Netztrafo an eine Steckdose anschließen (Vorsicht! Alle Leitungen sollten gut isoliert sein). Zwischen dem +5 Volt-Ausgang der Schaltung und Masse wird ein 100-Ohm-Widerstand eingesetzt. Wenn Sie ein

Voltmeter besitzen, messen Sie nun die Spannung über diesem Widerstand. Sie sollte nur wenig von 5 Volt abweichen. Falls Sie ein Oszilloskop haben, stellen Sie die Empfindlichkeit auf 10V und den Wahlschalter auf DC (Gleichspannung). Beim Anschluß der Prüflitungen über dem Widerstand muß sich ein stabiler Strahl zeigen. Auch geringe Unregelmäßigkeiten in der Stromversorgung stören die Funktion des Voltmeters nachhaltig.

Nach dem Test des Netzteiles können Sie es mit der Hauptplatine des Voltmeters verbinden. Die ICs dürfen in diesem Stadium aber noch nicht eingesetzt sein. Mit einem Meßgerät wird geprüft, ob an Pin 11 (IC1), Pins 4 und 8 (IC2) und an Pin 16 (IC3) auch wirklich +5 Volt anliegen. Als nächstes schalten Sie das Meßgerät in den Ohmbereich und verbinden eine der Meßleitungen mit der analogen oder digitalen Masse. Wenn Sie den Widerstand zwischen Masse und Pin 3 (IC1), Pin 1 (IC2) und Pin 8 (IC3) messen, sollte ein sehr niedriger Wert (weniger als 1 Ohm) angezeigt werden.

Richtige Positionierung

IC3 ist ein normaler TTL-Baustein und kann ohne besondere Vorsichtsmaßnahmen behandelt werden. Nur auf die richtige Positionierung (Markierung oben) muß geachtet werden. Bei IC1 und IC2 müssen Sie anders vorgehen. IC2 ist die CMOS-Version des Timer-ICs NE555 – es heißt zwar, daß dieses Bauteil gegen elektrostatische Entladungen geschützt ist, man sollte diese Aussage aber nicht unbedingt praktisch überprüfen... IC1 läßt sich jedenfalls mit Sicherheit durch Überspannung zerstören – falls Sie also bisher noch nicht mit CMOS-Bauteilen zu tun hatten, sollten Sie folgendes genau beachten:

Decken Sie zuerst Ihren Arbeitsplatz mit Aluminiumfolie ab. Als nächstes das Netzteil ausschalten und in die Mitte der Alufolie le-



gen. Fassen Sie nun unter die Folie und drücken Sie das Aluminium vorsichtig von unten gegen die Pinseite der Platine. IC1 und IC2 sind meist in eine leitfähige Umhüllung verpackt, oft in einen schwarzen Schaumstoff oder eine speziell behandelte Plastikfolie.

Legen Sie zunächst das Bauteil mit der Verpackung auf die vorbereitete Alufolie. Stützen Sie sich mit

den unbedeckten Ellenbogen auf der Folie ab und packen Sie das IC aus, ohne die Arme hochzuheben. Mit weiter aufgestützten Armen wird der Chip nun in seinen Sockel gesteckt und die Folie von der Platinenunterseite abgezogen. Wichtig: Vor Einschalten des Netzteiles die Folie entfernen!

Sind Netzteil und Verbindung zur Platine fertiggestellt und die ICs wie beschrieben eingesetzt, können Sie die Platine in Betrieb nehmen. Wenn Sie Pin 9 von IC1 nicht an die analoge Masse angeschlossen haben und mit einer erdfreien Meßschaltung arbeiten wollen, werden Sie nach dem Einschalten wild hin- und herspringende Ziffern auf LED1 und LED2 sehen. Sollten die Ziffern nach einem festen Muster wechseln, müssen Sie die Leitungen auf der Platine anders positionieren.

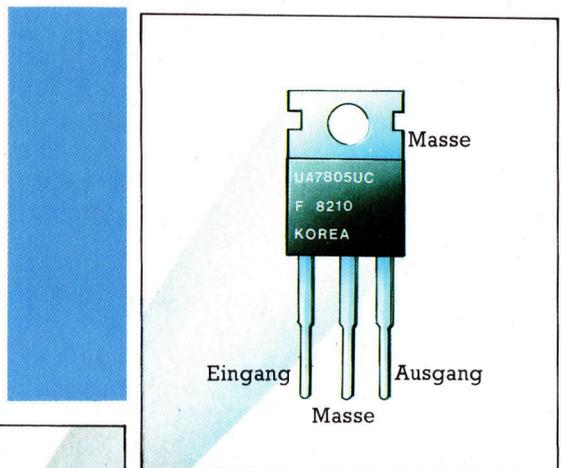
Liegt Pin 9 (IC1) auf Masse, und am Eingang ist nichts angeschlossen, zeigt das Display entweder 00000 oder vielleicht eine 1 oder 2 auf LED1. Wenn Sie jetzt IN- und IN+ kurzschließen, sollte die Anzeige stabil und ohne Flakern bei 00000 und 00001 liegen. Nicht ganz so perfekt wird das Ergebnis sein, wenn Sie bei Referenzdiode D1 oder beim Integrationskondensator C1 Kompromisse machen mußten.

Achtung!
Netzspannung kann gefährlich werden. Bevor Sie an der Platine arbeiten, müssen Sie unbedingt immer vorher den Netzstecker ziehen!

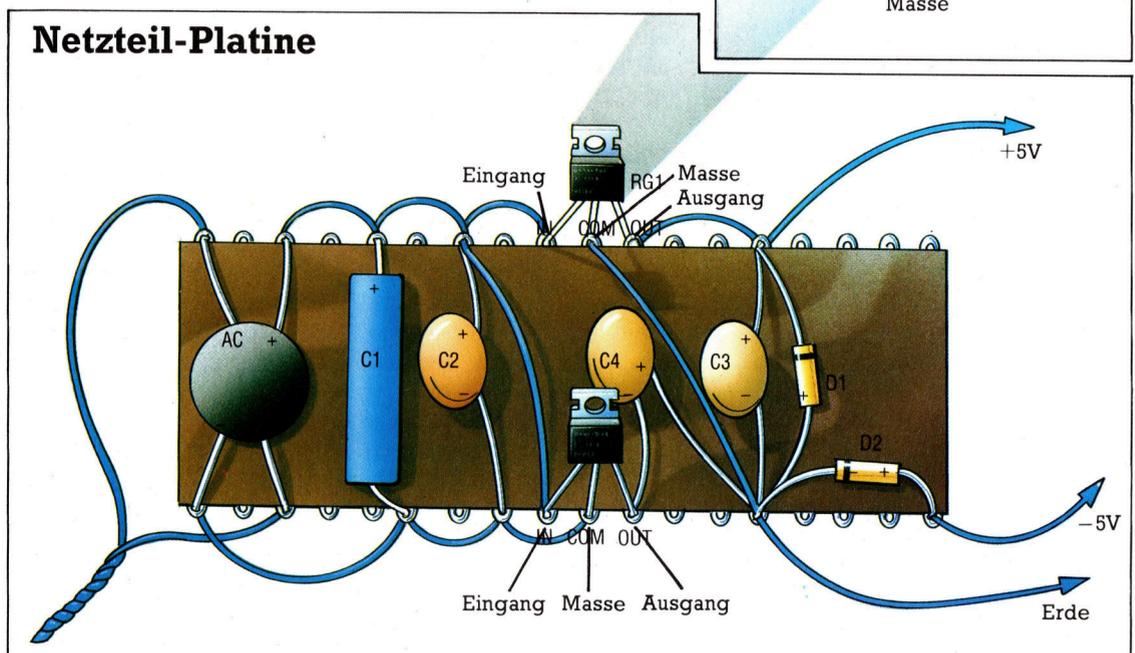
Schaltungsaufbau des Netzteiles

Die Bauteile des Netzteiles und der Spannungsregelung lassen sich gut auf einer kleinen Platine mit Lötflächen montieren. Ein Platinenstück mit 36 Lötstützpunkten reicht dazu aus. Die Anschlüsse der einzelnen Bauteile werden an den Lötflächen befestigt, die man nach Bedarf mit isolierter Litze untereinander verbindet. Die Wechselspannungsleitungen vom Trafo werden am Brückengleichrichter angeschlossen. Die -5-Volt-Leitung liegt am Ende der Platine und führt zum Kontakt unten auf der Multimeter-Schaltung. Die Schaltungsmasse verbinden Sie mit dem analogen Massekontakt.

Es ist sehr wichtig, die Polung von Kondensatoren, Dioden, Brückengleichrichter und Spannungsreglern genau zu beachten. Bei den Festspannungsreglern finden Sie die richtigen Anschlüsse auf der Rückseite. Von links nach rechts liegen die Anschlüsse dann grundsätzlich in der Reihenfolge Eingang, Masse und Ausgang.



Netzteil-Platine



Fachwörter von A bis Z

Permutation = Permutation

Die Elemente einer Menge können unterschiedlich arrangiert werden; jede mögliche Abfolge bildet dabei eine „Permutation“. Für die Buchstaben a, b und c etwa sind sechs Anordnungen denkbar: abc, acb, bac, bca, cab und cba. Allgemein ist die Anzahl der Permutationen von n Elementen durch $n!$ (gesprochen „n Fakultät“) gegeben. Die Definition der Fakultät lautet $n! = n * (n-1) * (n-2) * \dots * 1$. Mit fünf Elementen lassen sich demnach $5! = 5 * 4 * 3 * 2 * 1 = 120$ Permutationen erzeugen.

Picture Processing = Bildverarbeitung

Die Bildverarbeitung beschäftigt sich mit der (meist digitalen) Analyse und Umformung der Signale von bildgebenden Systemen, beispielsweise von Fernsehkameras; die Wiedergabe kann über einen Monitor oder einen geeigneten Drucker erfolgen. Der Informationsgehalt eines Bildes hängt einerseits von der geometrischen Auflösung (Ortsauflösung) ab, das heißt von der Anzahl der Bildpunkte. Ebenso wichtig ist aber zudem die Kontrastauflösung.

Das Bild steht im Speicher als zweidimensionales Array, wobei jedem Pixel ein oder mehrere Bits für die Darstellung von Helligkeit und Farbe zugeordnet sein können. Im einfachsten Fall gibt es nur zwei Helligkeitswerte, nämlich Pixel „ein“ oder „aus“, je nachdem, ob im Bild eine bestimmte Schwellintensität überschritten ist oder nicht. Mehr Information vermittelt eine Graustufen-darstellung, bei der jedem Pixel differenzierte Helligkeitswerte zugewiesen werden können.

Die Bildverarbeitung führt zu einer gezielten Veränderung des Bildcharakters, typisch durch digitale „Filterung“ der Bildinformation. Ein Beispiel dafür ist die Verbesserung der Detaildarstellung bei den Raumsonden-Aufnahmen vom Halleyschen Kometen, aber ebenso die Erzeugung verblüffender Effekte bei Videofilmen.

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.

Plasma Display = Plasma-Anzeige

Plasma-Anzeigefelder finden bei Computern zunehmend Verwendung. Sie arbeiten mit einer ähnlichen Zeichengeometrie wie Flüssigkristall-Anzeigen und emittieren rotes oder oranges Licht. Es entsteht durch eine elektrische Gasentladung innerhalb eines Elektrodenrasters, das zwischen Glasplatten eingebracht ist. Das Raster wird von dünnen Drähten gebildet, die in Matrixform angeordnet sind. Zwischen ihnen zündet beim Anlegen von Spannung die elektrische Entladung, und die Gasmoleküle werden durch Stoßionisation zum Leuchten angeregt. Durch entsprechende Ansteuerung der Matrixdrähte lassen sich beliebige Mosaikmuster erzeugen.

Plotter = Plotter

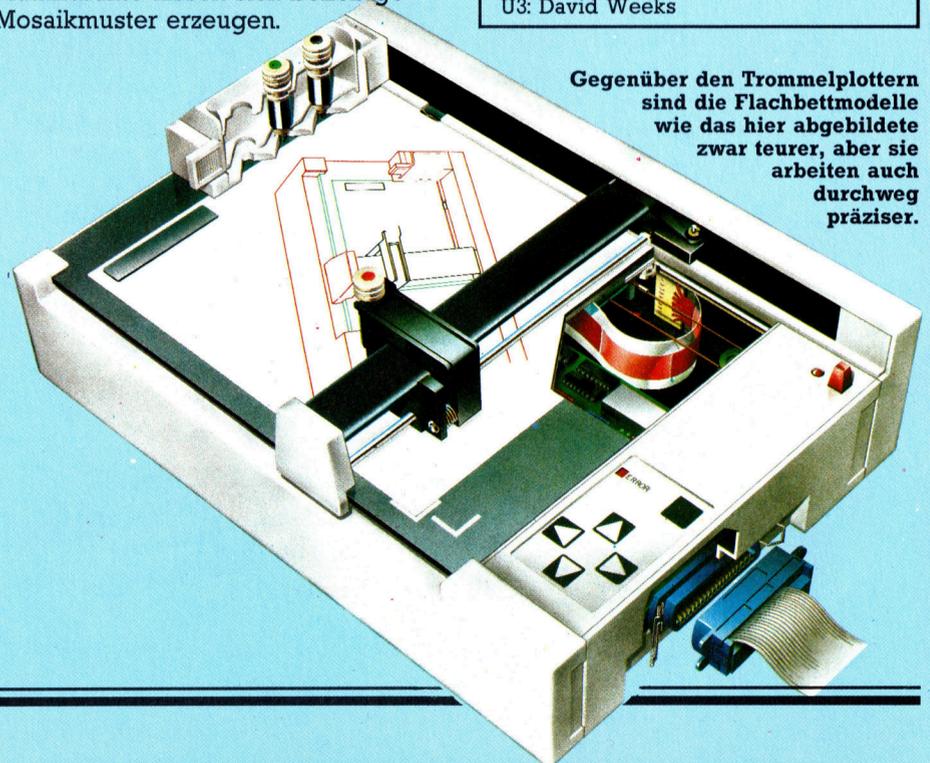
Ein Plotter ist ein Zeichengerät, das Rechnerdaten als Grafik zu Papier bringt – im allgemeinen mit Hilfe eines Schreibstifts, der in x- und y-Richtung über das Papier fährt und gehoben oder gesenkt werden kann. Es gibt zwei Bauformen: Flachbettplotter und Trommelplotter. Bei der Flachbett-Ausführung liegt das Papier als Einzelblatt auf einem ebenen Papiertisch, über den sich in x-Richtung ein Zeichenarm bewegt. Auf dem Arm ist in y-Richtung die Schreibstifthalterung verschiebbar angebracht. Außer diesem Kreuzschlittensystem gibt es sogenannte „Zeichenroboter“ nach dem Turtle-Prinzip.

Im Heimcomputerbereich sind jedoch die Trommelplotter am stärksten verbreitet. Dort wird ähnlich wie bei einer Schreibmaschine das Papier durch eine Walzenführung in y-Richtung transportiert, während davor in x-Richtung ein Wagen mit dem Schreibstift hin- und herfährt.

Bildnachweise

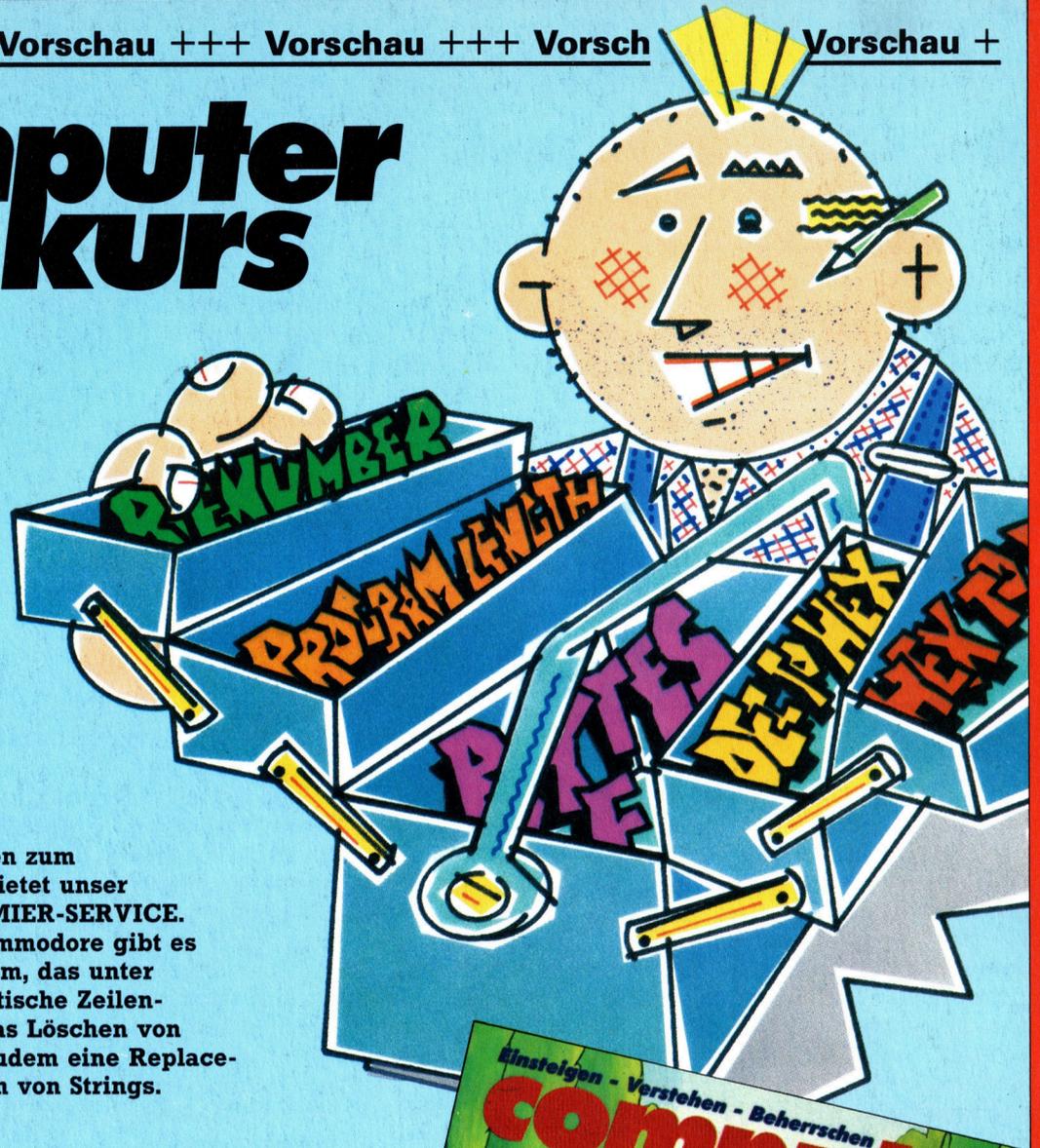
2073: Tony Sleep
2074, 2075, 2089, 2091: Caroline Clayton
2092: Dimension Graphics
2078, 2079: Liz Heaney
2081, 2093, 2094, 2098, 2100: Kevin Jones
2084: Liz Dixon
2086, 2087: Chris Stevens
U3: David Weeks

Gegenüber den Trommelplottern sind die Flachbettmodelle wie das hier abgebildete zwar teurer, aber sie arbeiten auch durchweg präziser.

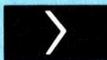


computer kurs

Heft **76**



Neue BASIC-Befehle, die das Programmieren zum Vergnügen machen, bietet unser nächster PROGRAMMIER-SERVICE. Für Spectrum und Commodore gibt es ein „Toolkit“-Programm, das unter anderem eine automatische Zeilennumerierung sowie das Löschen von Blöcken beinhaltet. Zudem eine Replace-Option zum Austausch von Strings.



Gut gemischt

Kartenspielen mit dem Computer – mit unserem BASIC-Programm für das Spiel 17 + 4. Der Beginn einer neuen Serie. Zunächst: Das Mischen.



Das Multitalent

Weitere Schaltungen verleihen unserem selbstgebauten Multimeter zusätzliche Fähigkeiten.



Durch die Röhre

Unter MS-DOS lassen sich Software und Daten logisch organisieren.



Mehr lernen

Bis heute ist noch nicht endgültig entschieden, ob Computer einen Dauerplatz im Schulleben erhalten.

