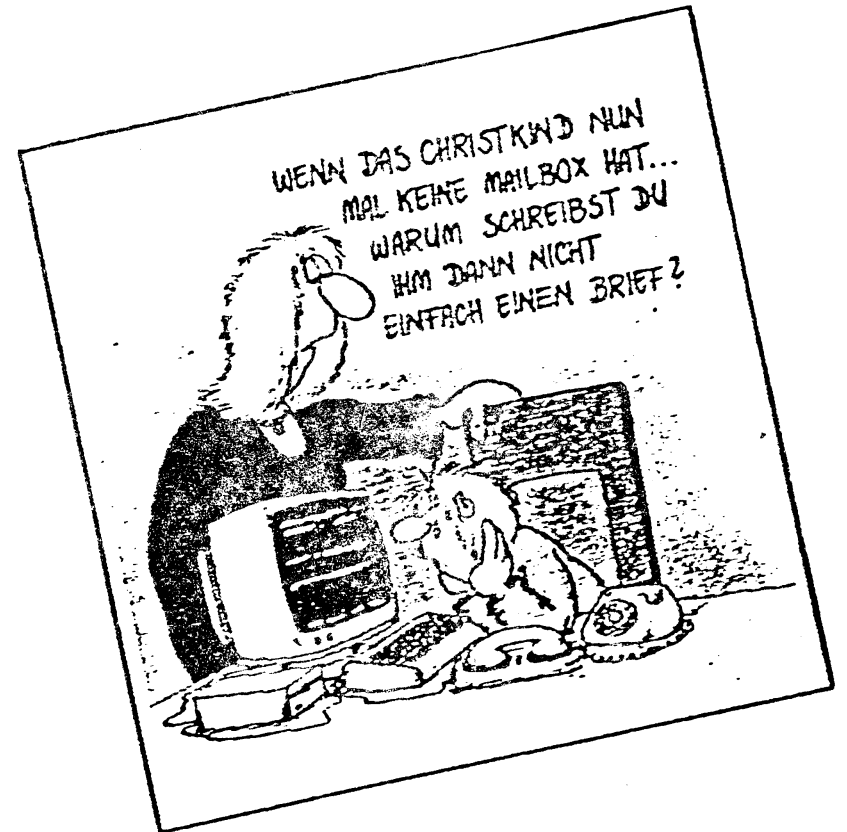


CLUB 80

CLUBINFO
DER

TANDY -
GENIE -
UND KOMITEE -
ANWENDER



22. AUSGABE

- I N H A L T S U E R Z E I C H N I S -

	Seite: und Autor:
<u>Clubinternes</u>	
Neues vom Vorstand	1 - 2 Hartmut Obermann
Geburtstagskinder	3 Jutta Obermann
Moin, Moin	4 Hartmut, Kurt, Gerald
Club-Emblem	5 - 6 Arnulf, Heinrich, Klaus
Vorstellungen	7 - 9 Hartmut, Alexander, Uli
Termine / Messen	10 Redaktion

<u>Software</u>	
Der schnellste Zahnarzt	11 - 12
DA CAPO	13 - 16
Schwere Kavallerie	17 - 19 Klaus-Jürgen Mühlenbein
Von 8 auf 24 Pins	20 - 22
TSCRIPS und die 24 Nadeln	23 - 25 Arnulf Sopp
Noch ein Bildschirmschoner	26 - 27 Alexander Schmid
Ein weiterer Bildschirmschoner	28 - 30 Arnulf Sopp
Etwas für Leute, die nie Zeit haben	31
Es geht noch schneller	32 - 33 Alexander Schmid
Ein Fehler in SYS28/SYS	33 - 34
The Times, They Are A-Changing	35 - 36 Arnulf Sopp
BOBS - Basic Datenbanksystem	37 - 39 Klaus Hermann
PASCAL - Teil 2	40 - 43 Rüdiger Sörensen
RPNL	44 - 52 Kurt Müller
Vergleich zweier Disk-Dateien	53 - 56
Vergleich zweier Programme	57 - 59 Arnulf Sopp

	Seite: und Autor:
<u>Hardware</u>	
Die Maus am TRS 80	60 - 62 Hartmut Obermann
Das 80-Zeichen-Karten-Problem ...	63 - 66 Helmut Bernhardt
Schwimmende Bilder	66 Alexander Schmid

<u>Börse</u>	
Wer hat was -- wer will was	67 - 70

<u>Sonstiges</u>	
Prüfzählfehler beim Lesen	71 Helmut Bernhardt
Fraktur	72 Klaus-Jürgen Mühlenbein
Sicherer Platz für Daten	73 - 77 Artikel aus TEST
Weihnachtsbummel (u. a.)	78 Spatzel vom Mausebär
32 Bit zum Niedrigpreis	79 - 80 Artikel aus ...

<u>Programm-Bibliothek</u>	
Neues von der Diskothek	81 - 82 Werner Förster

	Seite: und Autor:
<u>Club Bibliothek</u>	
Club-80 - Bicherei	83 Hartmut Obermann
Buchvorstellungen	84 - 86 .. Verschiedene

<u>Die letzten Seiten</u>	
Impressum	87
Schluß	88
Clubmitgliederadressen	am INFO-Ende Redaktion
Sonderausgaben	am INFO-Ende Hans-Martin Stephan, Redaktion

Neues vom Vorstand

Hallo Freunde,

ich hoffe, daß euch dieses Info noch vor der Jahreswende 1987 -> 1988 erreicht und wünsche euch deshalb einen guten Rutsch und ein ebensolches neues Jahr! Sollte sich das Info doch verspäten, gelten diese Wünsche natürlich genau so nachträglich, wie die Weihnachtgrüße, die ich hiermit aussprechen möchte.

Zum Jahreswechsel habe ich eine gute und eine schlechte Nachricht für euch.

Zunächst die schlechte: bis spätestens zum 15. Februar muß jeder seinen Jahresbeitrag 1988 bezahlt haben. Diejenigen, die diesen Termin versäumen, bekommen das nächste Info nicht mehr und werden, wenn sie auch die dann folgende Mahnung außer acht lassen, aus dem Club ausgeschlossen.

Und nun die gute: da der CLUB 80 im letzten Jahr sehr sparsam gewirtschaftet hat, bekommen alle Mitglieder, die vor dem 30.06.1987 dem Club beigetreten sind, eine Beitragsrückerstattung von 10,- DM! Bei den Mitgliedern, die zwischen dem 01.07 und dem 31.12.1987 beigetreten sind, reduziert sich die Rückzahlung auf 5,- DM. Um den Verwaltungsaufwand möglichst gering zu halten, wird der jeweilige Betrag vom zu zahlenden Jahresbeitrag 1988 abgezogen (beachtet dazu die den Infos beigelegten Überweisungsformulare). Mitglieder, die den Club zum Jahreswechsel verlassen, kommen nicht mehr in den Genuß der Rückzahlung!

Ich habe noch eine kleine Bitte an diejenigen Mitglieder, die den CLUB 80 zum Jahreswechsel, aus welchen Gründen auch immer, verlassen wollen. Bitte verabschiedet euch nicht sang- und klanglos durch Nichtbezahlen des Jahresbeitrages, sondern schickt mir eine Abmeldung. Eine Postkarte mit ein paar Zeilen reicht schon und erspart mir sehr viel Arbeit. Ich bedanke mich im voraus und wünsche denen, die uns wegen eines Systemwechsels verlassen viel Spaß und Erfolg mit der neuen Maschine!

Regionaltreffen

Wie ihr an anderer Stelle in diesem Info lesen könnt, fand am 21./22.11 ein Regionaltreffen "Nord" statt. Bezüglich dieses Treffens sprachen mich einige Mitglieder mit dem Wunsch an, auch für ihren Bereich Regionaltreffs zu veranstalten. Dazu möchte ich bemerken, daß nicht ich Veranstalter des "Nordlichtertreffens" war und ich mich auch nicht dazu bereitfinden werde, für andere Bereiche Regionaltreffen zu organisieren.

Meiner Meinung nach ist die Organisation und Durchführung Sache der Leute, die sich eine Zusammenkunft in ihrem Bereich wünschen. Dabei sollten sie wie folgt vorgehen:

1. eine kleine (3-4) Gruppe Interessierter finden und einen Termin absprechen (telefonisch);
2. den gefundenen Termin mittels eines Rundschreibens incl. Anmeldeformular etc. an die Mitglieder des CLUB 80, die in angemessener Entfernung zum Treffpunkt wohnen, verschicken (Portokosten können vom Club getragen werden!);

3. wenn zeitlich machbar, Information aller Mitglieder

durch Bekanntgabe des Termins im Info;

4. last but not least Information des Vorstandes!

Ein ähnliches Schema hat sich bei dem durch Gerald Schröder organisierten Treffen in der Nähe von Hamburg bewährt.

Zu diesem Thema möchte ich abschließend noch bemerken, daß ich sehr gerne bei den Vorbereitungen zu Regionaltreffen behilflich bin, mir aber nicht die Arbeit aufhalsen will und kann (schließlich gehe ich noch einer kleinen Nebenbeschäftigung nach), mehrere Regionaltreffen pro Jahr zu veranstalten!

Abwesenheit

Aus beruflichen Gründen werde ich mich (wieder einmal) für längere Zeit auf Reisen befinden. So werde ich mich für die Monate Jan., Feb. und März in Rentsburg (Schleswig-Holstein) aufhalten. Dabei wird es mir sicher nicht möglich sein, jedes Wochenende "in die Heimat" zu fahren. Unter diesem Umstand wird natürlich auch die Erledigung meiner Korrespondenz leiden; konkret: die Beantwortungszeit für Briefe wird wohl auf zwei bis drei Wochen steigen!

Ich möchte euch bitten, mir diese Verzögerungen schon im voraus nachzusehen und dringende Anfragen an die entsprechenden Vorstandsmitglieder (Gerald Schröder, Eckehard Kuhn usw.) zu richten. Die meisten administrativen Angelegenheiten (u.a. Verwaltung der Kasse, Versenden von Probeinfos usw.) wird meine bessere Hälfte erledigen, so daß der Club in keiner Beziehung führungslos ist!

Bücherei

Wie schon im letzten Info ausführlich verkündet, erweitert der Club demnächst seine Bücherei. Auf dem Nordlichtertreff haben Gerald und ich folgende Bücher ausgewählt, die angeschafft werden: -Turbo Pascal Wegweiser für Microcomputer (Grund- und Aufbaukurs);
- Pascal und Pascal-Systeme
- Programmierung in Modula 2
- CP/M - Anatomie eines Betriebssystems
- Das Z80 - Buch

Damit ist die Anschaffungswelle aber bei weitem noch nicht vorbei; im Frühjahr 1988 sollen weitere Bücher angeschafft werden. Wer noch Vorschläge/Buchbesprechungen hat, möge diese bitte an Gerald richten. Buchbesprechungen sollten zusätzlich direkt Jens Neueder zur Veröffentlichung zugesandt werden!

Sonderhefte

Eine weniger gute Nachricht muß ich euch leider noch eröffnen. Wie inzwischen sicher jeder mitbekommen hat, ist Jens Neueder durch den Bau seines eigenen Hauses bis über beide Ohren mit Arbeit eingedeckt. Dadurch wird sich der Neudruck vergriffener und die Erstellung neuer Sonderhefte bis in die Mitte des Jahres 1988 verzögern. Sollte sich jemand finden, der Jens diese Arbeit abnehmen kann, soll er sich bitte bei ihm melden.

Damit bin ich mal wieder am Ende angelangt. Ich wünsche euch viel Spaß bei der Lektüre des Infos, euer

Karimut Obermann

HEFT
22
Dezember
1987

02

01

Geburtstagskinder

Hallo, ihr lieben Clubfreunde,
 heute wende ich mich einmal direkt an euch (normalerweise ist ja mein Mann mein Sprachrohr zu euch).
 Ich möchte eine Idee des Bremerhavener Computerclubs aufgreifen und den Mitgliedern, die in den Monaten November, Dezember und Januar Geburtstag haben oder hatten, recht herzlich gratulieren.
 Im voraus möchte ich noch bemerken, daß ich nicht von allen Mitgliedern das Geburtsdatum vorliegen habe. All diejenigen, die in der genannten Zeitspanne Geburtstag haben und nicht genannt sind, bitte ich, mir das Datum mitzuteilen. DANKE.

Hier also nun unsere "Geburtstagskinder":

Im Monat November hatten Geburtstag:

Krispin Michael am 2.
 Hentz Werner am 3.
 Mühlenbein Klaus-Jürgen am 7.
 Schäfer Walter am 8.
 Neueder Jens am 28.

Im Monat Dezember hat Geburtstag:
 Böcker Dieter am 4.

Im Monat Januar haben Geburtstag:

Seelmann-Eggebert Jörg am 9.
 Betz Heinrich am 17.
 Drowäler Bernd am 29.

Noch einmal herzlichen Glückwunsch an alle genannten und ungenannten vom Vorstand und von mir.

Jutta Obermann

Moin, Moin!

Während die anderen noch beim Kaffeetrinken sind, dürfen wir (Hartmut und Gerald) einen kurzen Bericht über das Nordländertreffen verfassen.

Eins läßt sich jetzt schon sagen: die Sache war ein voller Erfolg! Schon am Freitag, den 20.11.87, war am Abend viel los, nämlich ein großes Trinkgelage der standfesten "Ältesten" des Clubs, Helmut Bernhardt, Arnulf Sopp, 1. und 2. Vorsitzender. Erst gegen 2.00 Uhr am Sonnabend war das Bett in Sichtweite.

Ein paar Stunden später fing der aktive Teil des Treffens sehr erfolgreich an, indem unser Assembler-Papst Arnulf sein IIIs durch einen 220 V-Reset ins Nirwana schickte. Dadurch war sein Geist aller Sorgen entledigt und er konnte uns so seine Geistesblitze unbehindert zuwerfen (noch ist nicht klar, ob er damit auch jemanden getroffen hat).

Nachdem die Stromversorgung wiederhergestellt war, konnte die Arbeit mit den Rechnern (zwei Genie I, ein Genie II, drei Model IVp) endlich beginnen. Es wurde kopiert, gehackt und eine Menge Informationen ausgetauscht. Notizblöcke wurden mit Ideen gefüllt, die später aufgegriffen und im Info veröffentlicht werden sollen. Als Beispiel wäre hier der von Hartmut angeleierte Vergleich zwischen drei Pascal-Dialekten, Z80-Assembler und RPNL zu nennen.

Eine der Hauptinteressen aller Anwesenden (ausgenommen Arnulf) war die Implementation und Anwendung von CP/M. Ein Problem bei der Anwendung war die Bedienung der Hilfsprogramme (Tastendefinition etc.). Wer zu diesen Thema und anderen Benutzungsproblemen etwas zu sagen hat: bitte mit kleineren Kapiteln im Clubinfo an die Öffentlichkeit treten!

Wie es sich für ein Treffen gehört, kam auch die unvermeidliche Hardware auf ihre Kosten. Der Versuch, den Drucker vom Hans-Joachim Eilers an den Rechner von Harald Mand anzuschließen, endete mit einem generellen "Lichtaus" für den Drucker. Eine Sicherung hatte sich zur Befehlsverweigerung entschlossen und war auf Nimmerwiedersehen durchgebrannt. Unser Beileid an den stolzen Besitzer!!!

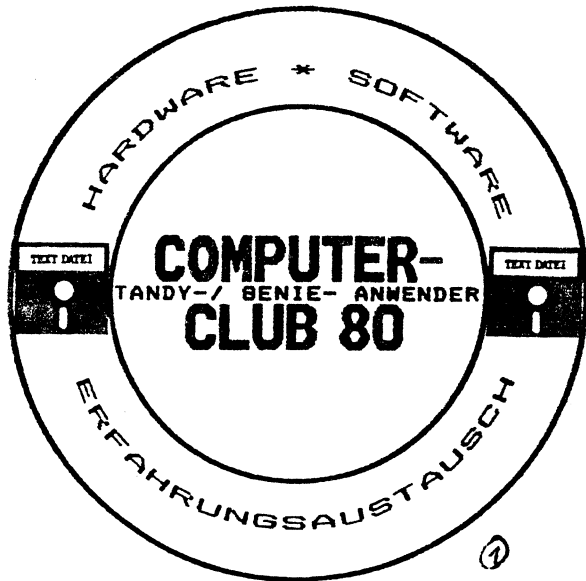
Neben den im Text schon genannten Software-Desperados und Hardware-Rambos nahmen am Treffen noch teil:

Kurt Müller mit Frau
 Michael Krispin auch mit Frau
 Matthias Homann (Gast mit Tendenz zum Clubeintritt)

In der Hoffnung, daß auch das nächste Regionaltreffen, egal ob Nord, Süd oder Mitte, ein solcher Erfolg wird, verbleiben wir (die Verfasser)

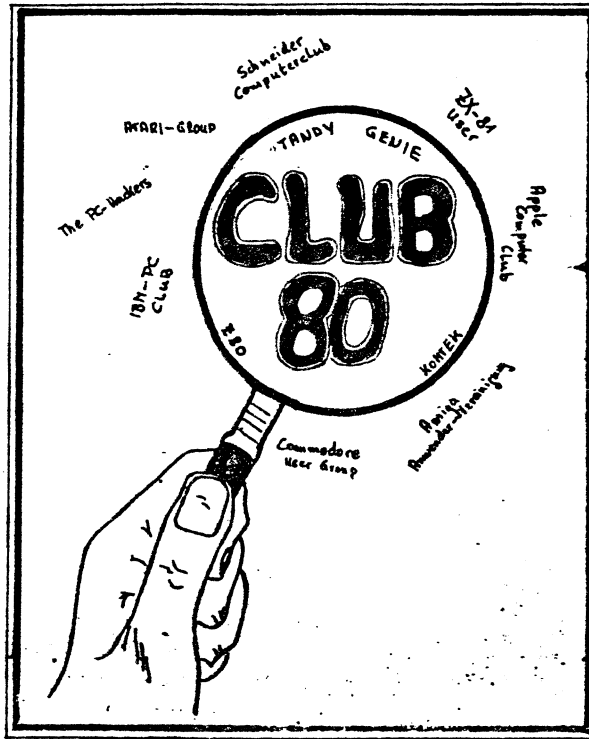
Hartmut Kurt und Gerald

Das Clubemblem - unser Aushängeschild

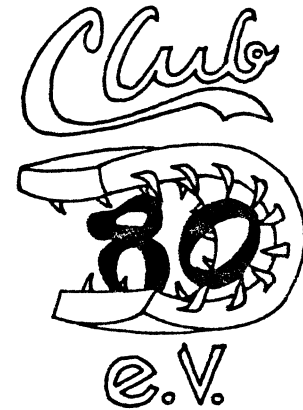
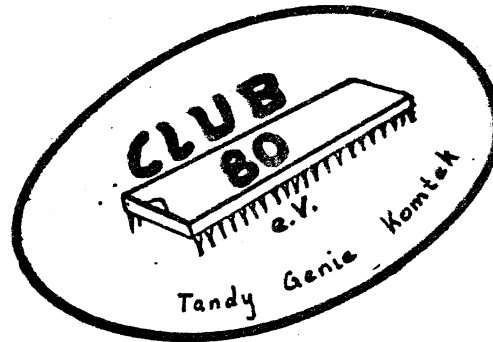


Angeregt durch Gerald Schröders Bericht im letzten Info über die bis jetzt noch magere Ausbeute bei den Entwürfen für ein Club-Emblem habe ich den Computer, Drucker, Zirkel, Schere und Leim zur Hand genommen und mit eigenen Versuchen begonnen. Im Laufe von zwei Stunden war der Papierkorb mit vielen Schnipseln gefüllt, schließlich lag aber doch eine brauchbare Vorlage auf dem Tisch.

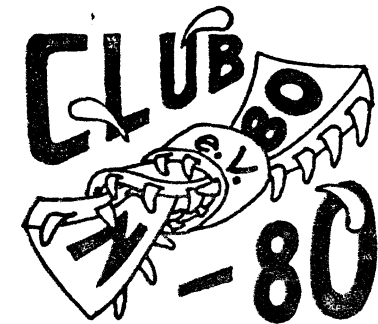
Man kann die Qualität des Entwurfes mit etwas mehr Aufwand noch verbessern:



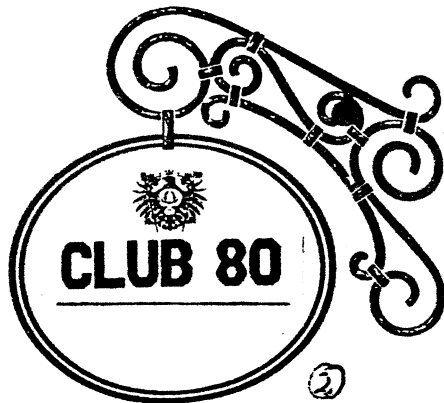
Klaus Hermann



Arnulf Sopp



Mit einer schwungvollen, im Lichtsatz hergestellten Schrift, statt des Nadeldrucks zwischen den beiden Kreisbögen, würde das Emblem lockerer und lebhafter aussehen. Aber mit der jetzt laufenden Aktion soll ja erst einmal Material gesammelt werden für ein "Aushängeschild" unseres Clubs. Unsere Vorstandschaft kann dann entscheiden, ob einer der Entwürfe noch verbessert werden soll, oder - was auch denkbar wäre - daß aus Teilen mehrerer Entwürfe die endgültige Form unseres Emblems entsteht.



Vorstellung - die zweite!

07 Es ist schon einige Zeit her, daß eine kurze Vorstellung meiner Person im Info erschienen ist. Inzwischen hat sich einiges geändert, und das nicht nur bei mir. Dadurch, daß der CLUB 80 sich ständig über neue Mitglieder freuen kann, werden sehr oft gleichlautende Fragen zu meiner Person, meinen Interessen und vor allem zu meiner Gerätekonfiguration gestellt. All Dies möchte ich hier nochmals für alle Mitglieder offenlegen!

Zur Person:

Ich bin Jahrgang 1958 (gehe also stark auf die 30 zu) und am 24. August geboren. Nach der Hauptschule und anschließendem Abschluß der Mittleren Reife im Fachbereich Elektrotechnik erlernte ich den Beruf des Informationselektronikers. Kurz nach Abschluß der Lehre bemühte sich zuerst die Bundeswehr und dann meine heutige Frau meiner. Von beidem konnte und wollte ich mich bis heute nicht mehr trennen und so gebe ich nun in Fragebogen an, Beruf: Soldat, Familienstand: verheiratet, Kinder: eine Tochter (8 Jahre alt)!

Zu den Interessen:

Natürlich liegen meine Interessen, neben den Hauptinteressen Familie und Beruf, im Bereich der Computerei. Auf dieses Hobby kam ich durch die Notwendigkeit, mich während der Zeit, in der ich nur "Soldat auf Zeit" war, auf einen späteren Zivilberuf vorbereiten zu müssen. Da ich die größten Chancen im Bereich der EDV sah und sehe, habe ich versucht, mich intensiv in dieses Gebiet einzuarbeiten. Dabei erfasste mich das Computereifer, das mich auch bis heute nicht mehr losgelassen hat!

Zur Zeit liegen meine Computerinteressen vor allem im Softwarebereich wobei allerdings auch die Hardwareseite nie ganz in den Hintergrund tritt (siehe "Maus am TRS 80"). Wenn mir die Arbeit als Clubvorstand Zeit dazu läßt, beschäftige ich mich unter anderem mit Turbo-Pascal (Grafiktreiber, Umschreiben von PC-Programmen), Assemblerprogrammierung (Patches in Betriebssystemen) und dem Ausprobieren von Standardanwenderprogrammen (WordStar, dBASE, Multiplan usw.) und Sprachen. Eventuell werde ich mich demnächst auch mit MODULA beschäftigen. Ein Hobby im Hobby ist das Knacken von CP/M-Fremdformaten, was mir immer sehr viel Spaß macht.

Zum System:

Anfangen habe ich einmal mit einem 16k Model 1, welches ich langsam auf 48k und drei Laufwerke ausgebaut habe. Dabei haben vor allem die Diskdrives ständig gewechselt, so daß ich glaube, mit Anschluß und Konfigurierung dieser Massenspeicher einige Erfahrung zu haben.

Vor einiger Zeit bin ich dann auf ein Model 4p umgestiegen, welches natürlich sofort zwei 80/DS/DD-Laufwerke und einen Anschluß für externe Drives (da hängt eine 40/DS/DD-Floppy dran) erhielt. Kurz darauf folgten die 128k-Erweiterung und die Installation der micro-Lab-HRG (640*240 Punkte). Als weitere Zusatzgeräte gibt es eine Maus und einen Epson RX80F/T-Drucker. Demnächst soll der Speicher auf 320k aufgebohrt werden, und in der Schublade schlummert auch noch der Plan für einen ECB-Busanschluß (der ruht dort aber schon fast ein Jahr lang!). Auch der Anschluß eines 8"-Laufwerks ist schon lange geplant.

Das 4p ist zwar, Zitat: "Der schönste Portable der Welt!", tragbar ist er mit ca. 14kg Gewicht aber nur bedingt. Aus diesem Grund habe ich mir zusätzlich noch ein Model 100 zugelegt. Das Ding ist nicht größer als ein DIN A4-Blatt, 5 cm hoch, ca. 1.5 kg schwer und damit tatsächlich tragbar! Den 100'er benutze ich hauptsächlich als Speicherschreibmaschine für unterwegs, da sich Dank der RS 232-Schnittstelle die Übertragung der Texte auf das 4p sehr einfach gestaltet.

Damit wäre wohl alles gesagt, was von Interesse sein könnte und mehr wird außerdem nicht verraten!

Kontakt übermann

Ich würde mich freuen, wenn noch mehr Mitglieder, die sich bisher noch nicht im Info vorgestellt haben, dies nachholen würden. Ich glaube, daß es sicher für alle Leser interessant ist, mehr über die anderen Clubmitglieder zu erfahren und so bestimmt noch mehr Kontakte zwischen den Mitgliedern entstehen!

Hallo Leute,
auch wenn es schwerfällt, aber Ihr habt wieder ein neues Clubmitglied zu beklagen. Ich habe bis jetzt zwei Jahre mein Unwesen im Bremerhavener und im Münchner Club getrieben. Der dritte Beitrag für den CLUB 80 war mir immer etwas zuviel des Guten, aber die sinkende Aktivität läßt einen fast keine andere Wahl.

Zur Person: Ich bin 22 Jahre alt und wurde durch eine Reihe grausamer Schicksalsschläge nach München verschlagen (ich bin ein Hamburger ...nein, keiner von Mc Donalds!). Momentan bin ich mitten im Studium (E-Technik) und hoffe einigermaßen heil durchzukommen. Zu meinen Hobbies zählt, wie man sieht, auch die Computerei. Mein Fuhrpark besteht aus einem Genie IIs das mit 8MHz läuft und mit einer HiRes (480x192), einer 80-Zeichen-Karte und 512k-RAM (mit Helmut Bernhardt's Banker) ausgerüstet ist und einem Epson RX-80 F/T. Am liebsten beschäftige ich mich mit den kleinen Hintertürchen und Stolpersteinen im DOS und wie man die Library sinnvoll erweitert (hat jemand mehr als 65 Befehle untergebracht?)

Auf gute Zusammenarbeit

Alexander Schmid

0A

Guten Tag, Freunde!

Der guten Sitte des Club 80 folgend, sich als neues Mitglied kurz vorzustellen, möchte ich einige Zeilen zum Besten geben:

Einige von Euch mögen hier und dort schon den einen oder anderen Artikel von mir gelesen haben, sodaß diese sich schon ein Bild davon machen können, was sich der Club 80 da mit dem Herrn Heidenreich angelacht hat. Mein "EDV-Lebenslauf" dürfte diese Bild weiter abrunden:

So Ende 1979 wurde ich erstmals mit der Datenverarbeitung konfrontiert, als im Rahmen eines E-Technik-Studiums auch zwei Semester Datenverarbeitung nebst Praktikum angesagt waren. Nach anfänglich wenig Begeisterung ob dieser zusätzlichen Belastung neben "wichtigeren" Studienfächern schlug dieselbe bald in Richtung EDV um. Mit dem Erfolg, daß die ehemals als wichtiger angesehenen anderen Studienfächer bald zur Nebensache wurden und Ihr jetzt keinen Dipl. Ing. FH im Club begrüßen dürft. Als zudem noch mich störte, daß man den Rechner eigentlich nur via Lochkarten bedienen durfte, den Rechenzentrums-Assistenten dagegen, daß ich den Rechner nicht nur via Lochkarten bediente, folgte einem Hausverbot im Rechenzentrum die Anschaffung eines eigenen Computers.

Hätte ich damals das preiswerteste Modell erworben - manch einer mag sich noch an das "Superboard" erinnern - wäre ich wohl der Prozessorfamilie weiter gefolgt und hätte heute einen Commodore auf dem Schreibtisch stehen. Schon damals der Hardwareseite weniger zugeneigt entschied ich mich, - zum Glück? - gegen den Superboard-Bausatz für einen kompletten EG 3003 (später hieß er auch Genie I) und gesellte mich damit zu den 80er Freunden.

Diesem EG 3003 mit sagenhaften 16 KByte Hauptspeicher, Keiner Groß/Kleinschrift und ohne deutsche Umlaute folgte, sobald es meine Brieftasche zuließ, das Genie III günstig zum Händler-Einkaufspreis für nur (!) 5182,- DM. Irgendwann dazwischen lernte ich auch Arnulf kennen, über den ich letztendlich auf den Club 80 aufmerksam wurde.

Nachdem mich neulich Vater Staat für 15 Monate aus dem bürgerlichen Leben entführte, opferte ich gesparten Wehrosold, Entlassungsgeld und ein wenig vom Notgroschen für einen AT-Kompatiblen. Aber keine Angst, Freunde: dem 80er werde ich schon allein aus dem Grunde treu bleiben, weil man mit NEWDOS/80 ein Entwicklungssystem und mit MS-DOS eine Anwenderprogrammabfahrkrücke unter den Fingern hat!

Für den Club 80 habe ich mich letztendlich auch deshalb entschlossen, weil in unserem 80er Hobby noch soviel Interessantes steckt, daß es einfach zu schade wäre, seine Werke mangels Interesse "echter" Fachzeitschriften in den Mülleimer wandern zu lassen.

Ach, so! Wen's neben der technischen Hobbyseite interessiert: Ich bin Jahrgang '58, 183 cm groß, rund 100 kg schwer und derzeit arbeitslos.



„Frau Müller, Sie haben ein äußerst ungewöhnliches Baby!“

--- Termine --- Termine --- Termine ---
Nächster Redaktionsschluß 28. Februar 1988
Jahresbeitragstermin 15. Februar 1988 (spätestens !!)

--- Messen '88 ---
Hobby-tronic & COMPUTERSCHAU Dortmund 3. - 7. Februar 1988
IMPRINTA Düsseldorf 18. - 24. Februar 1988

HEFT
22
Dezember
1987

09 Euer Uli

* * Der schnellste Zahnarzt * *

 (Unmed.-techn.Assistent: KaJot Mb)

Seit es bei den Zahnärzten die amerikanischen Drehstühle gibt, auf denen der erwartungsvolle Patient mit 26400 UpM (= Kammerton!) gedreht wird, während der Bohrer stillsteht, macht das Wurzelziehen richtigen Spaß! Denn das mühsam unterdrückte "Aaaahhh..." des geehrten Gedrehten hat jetzt wenigstens die offizielle, genormte Tonhöhe, die die Anwesenden - Arzt, Assistentin und ihr Opfer - im musikalischen Genusse vereint und somit harmonische Stimmung aufkommen läßt. Kein Unwohlsein kann diese mehr beeinträchtigen. Und es geht flott von der Hand...

Warum erst in diesem Jahrhundert?

Schon vor 2250 Jahren lehrte uns ARCHIMEDES, wie man Wurzeln angenehm schnell und schmerzlos zieht. Und das mit beliebig vielen Stellen!

Als ich im letzten INFO u.a. ein Programm zum Wurzelziehen mitteilte, habe ich euch diese Tatsache unterschlagen, (um eure Geduld zu prüfen). Diejenigen, die es ausprobierten, haben das Programm (oder sich selbst) in der Luft zerrissen. Mit Recht!

Ein Fünfzeiler tut's!

In Sekundenschnelle, je nach Güte des Schätzwertes, den jeder sofort ansetzen kann, der die Quadrate der Zahlen von 1 bis 10 kennt; also die Zahlen: 1, 4, 9, 16, 25, 36, 49, 64, 81, 100. Als Schätzwert nimmt man diejenige Zahl von 1 bis 10, deren Quadrat den ersten beiden Stellen des Radikanden (das ist der "Patient") am nächsten kommt, wenn er eine gerade Anzahl Stellen hat bzw. der ersten Ziffer am nächsten kommt, wenn seine Stellenzahl ungerade ist; an diese vorläufige Zahl werden noch halb soviel Nullen angehängt, wie der Patient Stellen außerdem noch hat. Beispiel:

Um die Wurzel aus 4711 zu ziehen, beginne ich mit dem groben Schätzwert 70, denn 7 zum Quadrat (=49) kommt den beiden ersten Ziffern (47) am nächsten; dazu kommt eine Null, denn außer 47 hat dieser "Patient noch zwei Ziffern. Oder:

Um die Wurzel aus 8123456 zu ziehen, beginne ich mit dem Provisorium 3000: Die Stellenzahl ist ungerade; ich schaue mir also nur die erste Ziffer (8) an; das Quadrat von 3 kommt ihr am nächsten; dahinter gehören noch 3 Nullen, von wegen der 6 Ziffern nach der 8. - Capéz? - Capé!

Mehr bedarf's nicht. Das Programm errechnet hieraus in Sekundenschnelle die - na was schon? Und das mit einer Genauigkeit, die kein Taschenrechner bietet - nämlich mit soviel Stellen hinter dem Komma, wie übrig bleiben, wenn man die Anzahl Stellen vor dem Komma von 16 abzieht. Mehr kann die "Variable mit doppelter Genauigkeit" bekanntlich nicht leisten.

Wer gibt einen Trick an, die mit 16 Stellen begrenzte "Genauigkeit" unseres 8-Bitters zu überbieten? Ich meine, nicht nur beim Zahnarzt, sondern bei jeder beliebigen Berechnung? (Man führe die Rechnung mit String-Transformationen durch!)

Hier nun ARCHIMEDES' Meisterleistung:

Wurzelberechnung nach ARCHIMEDES

```

1  *** Wurzelberechnung nach ARCHIMEDES ***
2  *** <C> K.-J.Mühlenbein, Weinheim 1987 ***
3
10 CLS
   DEFINT L
   DEFDBL R,S,W
   PRINT "          Wurzelberechnung"
20 PRINT
   INPUT "Radikand ";R
30 INPUT "Schätzwert für die Wurzel ";S
   L = LEN ( STR$ (S) ) - 1
40 W = R / S
   IF ABS ( S - W ) < 1 / 10^(15 - L)
   THEN PRINT "Wurzel aus"R"="W
   END
50 S = ( S + W ) / 2  ← Archi's Trickzange!
   GOTO 40
  
```



(oder: Wie man Töne erzeugt)

Meiste Töne

Nein? - Vielleicht möchtest du welche von dir geben? Aber wie??

Ich meine natürlich: Wie mit dem Computer?

Nun, ein paar physikalische Grundlagen der Tonerzeugung mögen dazu verhelfen; die Übertragung auf den Computer ist dann - wie ich inzwischen erlebte - vom jeweiligen Computer abhängig; selbst bei gleichem Prozessor. Also meist keine "kompatible Angelegenheit". Aber wenn du deinen Computer kennst - das sei vorausgesetzt - wird es dir leicht fallen, das hier Beschriebene auf dein Gerät zu übertragen. (Hierzu gehört nicht einmal Musikalität.) Und zwar natürlich in Maschinensprache! Denn wenn sich erst noch der Interpret zwischen Tonanforderung und -ausführung einschalten muß, hörst du vor lauter "Denkpausen" - na, was wohl? Eben Pausen und kaum Töne...

Machen wir's so kurz wie bündig. Denn es soll keine Oper mit drei Akten werden. Ja nicht einmal eine Ouverture.

Schlimmstenfalls eine Tonleiter.

Und die bringt dich dann schon heiter

um mehr als einen Ton weiter...

Jeder reine Ton ist im wesentlichen eine Folge von Sinuswellen. Die zugehörigen "Obertöne" wollen wir verschweigen. Und was eine Sinuswelle ist, weiß jeder Techniker. Eben eine schöne Welle. Die hat einen Berg und ein Tal. Beide haben eine bestimmte Höhe bzw. Tiefe; die wir als absolut-gleich groß annehmen wollen. Glaube nur nicht, dies sei die "Tonhöhe". Diese Höhe (Tiefe) bestimmt vielmehr die Intensität, mit der die Welle dich umschmeißt, wenn du darin badest. Man nennt sie die "Amplitude": (Physikalisch ist die Schallintensität zum Amplitudenquadrat proportional.)

Die akustische Tonhöhe jedoch - die nicht unsere Muskeln, sondern unsere Ohren etwas angeht - wird durch die Häufigkeit bestimmt, mit der die einzelnen Wellen dich anstoßen. Sie ist um so höher, je schneller die einzelnen Wellenstöße aufeinander folgen. Brummst Du noch wohligh "Ahhh", wenn dich alle zwei Sekunden ein Wellengang hin- und herschaukelt, so kreischst du bald in hellen (hohen) Tönen, wenn dich ein wilder Wellengang mit kurzen schnellen Stößen durcheinander schüttelt! Oder nicht?

Lautstärke ist also eine Sache der Amplitude - Tonhöhe eine Sache des "Wellenganges", der langsam oder schnell aufeinander folgt; sagen wir: der "Schrittlänge" der Wellen, kurz: Wellenlänge. In der Akustik bevorzugt man zur Kennzeichnung der Tonhöhe das Gegenstück der Wellenlänge: die schon erwähnte Häufigkeit, mit der die Wellen in jeder Sekunde eintr@ff@n; oder vornehmer: die Frequenz (proportional zur reziproken Wellenlänge), gemessen in Hertz (Hz) = Schwingungen pro Sekunde.

Frequenz mal Wellenlänge ergibt die Wellengeschwindigkeit. Im hier behandelten Fall: die Schallgeschwindigkeit (i.d.R. in Luft so um die 330 m/s).

Trifft die so erschütterte Luft auf deines Ohres Trommelfell, so hörst du was (hoffentlich). Und umgekehrt: Schlägst du auf ein Trommelfell (bitte, nicht des Ohres, nimm eine Pauke), so wird die Luft von ihm, nein: von den von ihm ausgehenden "Schwingungen" (Stoßwellen in der Luft) erschüttert. Wem sag' ich das?

Und genau so ein tonerzeugendes "Trommelfell" ist die Membran des Lautsprechers in oder an deinem Computer (so du einen Verstärker hast), den du, wie du weißt, durch "Stromstöße" - oder vielmehr Spannungsschwankungen in Schwingungen versetzen kannst.

So, damit haben wir schon alles, was an Voraussetzungen gebraucht wird: Wir wissen nun, daß die Membran des Lautsprechers in mechanische Schwingungen versetzt werden muß; dies vollbringt der Verstärker, in welchem elektrische Spannungen erzeugt werden müssen, die mit der gewünschten Frequenz zwischen Ausschlägen nach der einen Seite ("Wellenberg") und solchen nach der anderen Seite ("Wellentalern") gleichmäßig abwechseln! Die Aufgabe des Computers ist es nun, diese elektrischen Signale im "richtigen", d.h. melodiegerechten Rhythmus zu erzeugen. Da die Taktfrequenz der CPU aber festliegt und 200 bis 300 mal höher ist als alle hörbaren Töne, muß die CPU stets während der Dauer einer Halbwelle vor sich "hintrudeln", bevor der nächste Anstoß nach der anderen Seite (Halbwelle) erfolgt. Dies macht die Programmierung zugegebenermaßen etwas komplex und ein Musikprogramm entsprechend unübersichtlich (finde ich).

Wir können auch die Stärke der Ausschläge (ihre Amplitude) beeinflussen, um dadurch verschiedene Lautstärken hervorzurufen; also das, was man in der Musik "Dynamik" nennt. Aber das soll uns jetzt nicht beschäftigen, denn mit unseren "billigen" Computern haben wir hierfür nur sehr begrenzte Möglichkeiten. Daher klingt einfache Computer-"Musik" auch meist so "eintönig" (=monoton? - Ein untreffendes Wort! Denn es werden ja viele Töne erzeugt, nur jedoch mit gleicher Lautstärke; also eher: "mono-phon".)

Außer der Schwingungsdauer, die der reziproken Frequenz und damit Tonhöhe entspricht, müssen wir bei einem Lied natürlich auch noch die Dauer der einzelnen Töne selbst, eben die Tondauer berücksichtigen. Das bedeutet, daß der Wechsel zwischen höherer und niedriger Spannung im Verstärker nicht nur mit einer der gewünschten Frequenz (= Tonhöhe) entsprechenden Geschwindigkeit erfolgen muß, sondern auch eine gewünschte Zeit lang andauern muß (= Tondauer). Bei der Realisierung im Programmablauf bedeutet das also, wie jedermann sofort erkennt, die Ineinanderschachtelung zweier Programmschleifen. Damit ist nun, meine ich, wirklich alles gesagt, was zu den Grundlagen Computer-unterstützter Tonerzeugung (CAT = Computer Aided Tuning <Copyright 1987 KaJot, zur Weitergabe wie immer freigegeben>) zu sagen ist.

Bleibt nur noch die Frage: Wie kommt der Wille zum Ton von der programmgesteuerten CPU bzw. vom Speicher zum spannungsgesteuerten Verstärker?!

Dies ist offenbar von Computer zu ebendiesem verschieden. Beim GENIE IIIs erzeugt die Speicherstelle 3860Hex einen "Knacks", wenn sie angesprochen wird (weiß ich von Arnulf); beim TRS-80 wird die Spannung über den Kassettenport 255dez = FFhex ausgeliefert, und zwar mit höherem Potential ("positiver Wellenberg"), wenn Bit 0, und mit niedrigerem Potential ("negativer Wellenberg" = "Wellental"), wenn Bit 1 gesetzt ist; dazwischen liegt in der Mitte die Ruhelage, um die die Welle schwingt, mit rückgesetzten Bits 0 und 1:

- | | | | | | |
|----|------------|---|-------------------------|---|-------------------|
| a) | Wellenberg | { | => Bit 0 gesetzt | } | = 01 binär = 1dez |
| | | | => Bit 1 nicht gesetzt | | |
| b) | Ruhe | | => Bit 0 und 1 rückges. | | = 00 binär = 0dez |
| c) | Wellental | { | => Bit 0 nicht gesetzt | } | = 10 binär = 2dez |
| | | | => Bit 1 gesetzt | | |

Lassen wir die Spannung zwischen a) und b) schwanken, so erhalten wir also einen Ton. Das wollen wir gleich mal "in BASIC" ausprobieren:

FOR I = 0 TO 100 : OUT 255, 1 : OUT 255, 0 : NEXT I
Wir erhalten einen Brumnton, dessen Tiefe durch die CPU-Frequenz und die Interpretationsgeschwindigkeit bedingt ist.
Wenn wir statt 1 und 0 im Wechsel 2 und 0 auf den Port 255 ausgeben (also Zeilen b und c abwechseln), erhalten wir das Gleiche (gleicher Spannungsunterschied).
Lassen wir 1 und 2 miteinander wechseln (Zeilen a und c), erhalten wir (etwa) den gleichen Ton, aber viermal so stark (doppelter Spannungsunterschied = doppelte Amplitude = vierfache Intensität; die Hör-Intensität wird allerdings wegen des WEBER-FECHNER-schen Gesetzes nicht viermal so laut empfunden.) Mehr nicht zum Thema "Lautstärke" (s.o.)
Die Tonhöhe jedoch erzeugen wir in Maschinensprache, indem wir den ungleich schnelleren CPU-Takt ohne dazwischen geschalteten Interpreter so oft walten lassen, wie die Frequenz des jeweils gewünschten Tons es vorschreibt. Wie bereits gesagt.

Um dies alles etwas zu erfassen, schauen wir uns einmal das Listing des "CLUBBO-Liedes" in diesem Heft an! Es handelt sich um eine revidierte Fassung, nachdem mich Arnulf auf einige Umständlichkeiten in meiner ersten Fassung hingewiesen und mir Tips gegeben hat, wie man den Code eleganter gestalten kann. (Die Tonwiedergabe selbst ist dadurch allerdings nicht schöner geworden - "Mann bleibt Mann" und Computer dto.)
Zeilen 52 bis 100 enthalten die Melodie "Wohlauf, Kameraden,...". Hierzu später.
In Zeile 10 wird die Adresse des Anfangs der Melodie ins Doppelregister HL geladen.
Zeilen 11 bis 15 sorgen dafür, daß die Töne nicht ineinander verschwimmen, sondern schön rhythmisch aufeinander folgen.
In Zeile 16 wird der Akku mit der Adresse des "LSB" (niederer Teil des 2-Byte-Wertes der Tonhöhe) geladen, der dann in Zeile 21 zum Register C weitergeschickt wird, falls das Wiederholungszeichen "DACAPO" noch nicht erreicht wurde. Wurde dieses erreicht (Zeile 76), wird das DACAPO-Flag zuerst unkenntlich gemacht (sonst gäbe es eine "Schallplatte mit Sprung") und dann dieser erste Teil der Melodie wiederholt. (Anmerkung: Nur die mit dem Asteriken * gekennzeichneten Zeilen stammen von mir; alles übrige von Arnulf.)
Zeilen 22-23 schicken das MSB, das höherwertige Byte der Tonhöhe, zum Register B, so daß jetzt das Doppelregister BC die die Frequenz bestimmende "Wellenlänge" 150 des ersten Tones enthält.
Zeilen 24-27 machen das Gleiche mit der Tondauer (hier: die "90"), die in den nächsten beiden inzwischen erreichten Adressen (hier 5249h und 524Ah) steht.
Nachdem der "Notenzeiger" HL in Zeile 28 auf die Adresse der nächsten Note gesprungen ist, wird in Zeile 29 die erste Schleife zur Kontrolle der Tondauer betreten. Solange sie läuft (Kontrolle in Zeilen 30-32), wird nun ab Zeile 33 bis Zeile 39 der Spannungspegel am Port 255 abwechselnd hoch und niedrig gesetzt und dazwischen das Unterprogramm "Knack" (Zeilen 41 bis 46) gefahren, welches die Membran schubst bzw. zerrt (push/pull). Nach jeder Volloschwingung = 2 Halbschwingungen = 1 Periode ... (Forts. nächste Seite) ->

Ann.: Der Techniker weiß, daß so keine reinen Sinuswellen erzeugt werden können; auf das hierfür notwendige Superpositionsprinzip mehrerer - theoretisch unendlich vieler - Frequenzen soll hier nicht eingegangen werden. Es genügt der Hinweis, daß dieser Mangel für den aräseligen Ton dieser Computer "musik" mitverantwortlich ist. Die von Arnulf stammende Label-Bezeichnung "Knack" ist deshalb durchaus treffend.

... fragt das Programm in Zeilen 43/44, ob noch Schwingungen gewünscht werden; denn pro Sekunde sind ja je nach Tonhöhe viele Schwingungen auszuführen. Ist dies der Fall, geht's zurück nach Zeile 42 zum nächsten Schwingungs "knack", andernfalls zurück zu Zeile 39, die sich zunächst die soeben durch die dauernde Dekrementierung abgebaute Tonhöhe aus dem Stack zurückholt, um sie wieder zur Verfügung zu haben, falls sich nach dem Rücksprung nach Label "loop2" (Zeile 40) ergibt, daß - entsprechend dem jeweiligen Stand des Tondauerzählers, welcher nunmehr in Zeilen 30 bis 32 geprüft wird - der Ton noch länger andauern soll. Ist der Ton noch nicht zünde - Verzeihung: zuende - geht das Spielchen mit Spannungszuteilung usw. ab Zeile 33 von vorne los. Ist aber DE=0000000Q00000000bin, so beschäftigt sich unser fleißiger Tonerzeuger auf eben die gleiche Weise mit dem nächsten Ton, der ihm in Zeile 16 vorgelegt wird - nicht ohne vorher das eingangs bereits gelobte (unmerkliche!) "Päuschen" (Zeilen 11-15) eingehalten zu haben...

Ann.: Ich weiß zwar, daß die Gleichung DE=0 genügt hätte; doch weiß ich auch um die didaktische Unvollkommenheit von Computer-Lehrbüchern; 16 Nullen prägen dem Anfängerhirn optisch ein, daß es sich um ein Doppelregister mit 16 Bits handelt. [2 * 8 Bit = 16 Byte]

War dieses bisher der "programmatische" Teil des Programms, sozusagen das Instrument oder Handwerkszeug, so folgt jetzt der nicht weniger wichtige, aber wesentlich flexiblere Teil, der das zu bearbeitende Material enthält: die Melodie; die Folge der Töne. Dieser Teil ist flexibel, weil hierfür jederzeit andere Melodien eingesetzt werden können.
Statt einer ausführlichen Ableitung der Formeln für Frequenz- und Tondauervorgabe, die zu weit führen würde (das INFO ist kein Mathebuch), sei es gestattet, nur diese Formeln selbst anzugeben. Anzuwerker ist lediglich, daß die Basiszahl, auf der die Berechnungen für die Tonhöhen beruhen, vom CPU-Takt abhängig ist. Was ich auch erst bei der Übertragung vom GIII's auf meinen TRS-80 herausfand.
Drum:
Probieren ergänzt das Studieren!

Kajol

ANHANG: Datenformeln.

Kammerton a' -> 440 Hz

übrige Frequenzen: f = 440 * (sqrt(2))^n / Hz

mit n > 0 Anzahl der Halbtonabstände vom Kammerton nach oben bzw. n < 0 " " " " " " " " unten.

Höhenzähler (=jeweils 1. Zahl in der Tönetabelle):

H = k/f

worin die "Zeitkonstante" k experimentell zu ermitteln ist. Sie ist vom CPU-Takt sowie der Programmstruktur abhängig und beträgt bei meinem Programm k=66000. [Basiszahl für TRS80/1]

Jede Tonhöhe hat ferner einen eigenen "Zähler" für die Notendauern. Als Basiswert für die Viertelnote mit dem Kammerton a' (=440Hz) wurde L=90 gewählt. Es gilt:

L = c * f (2. Zahl i.d. Tönetabelle)

Der Längenzähler ist also eine lineare Funktion der Frequenz. Die Proportionalitätskonstante c hat in unserem Musikbeispiel den Wert c = 0,2045... / Hz^-1
Die anderen Zähler für 1/16, 1/8- und 3/16-Noten sind entsprechende Bruchteile von L.

T O N E T A B E L L E

* * * Schwere Kavallerie * * *
(von Sopp, nicht Suppé)

Es wurde geritten! Der DERBY-Sekt ist vergeben! Mein Preisausschreiben aus INFO 20, Seite 9 ("Noch ein Leserbrief") ist gelaufen. Kaum war das INFO raus, kam das Roß aus Lübeck angesprengt - sein Reiter nahm die Trophäe unter hellem, klarem Trompetenklang entgegen: Arnulf war der Erste (und Letzte). Er dokumentierte mir seine Lösung nicht nur mit einem sauberen Listing, ndern demonstrierte es auch gleich ad auditum von Tonband-Kassette:

"Wohlauf, Kameraden, auf's Pferd, auf's Pferd..." erklang's aus meiner Stereoanlage.

(Drum: "Auf's Wohl, auf's Wohl, AS-semblihierer!") [Gewischer-] So muß ich's wohl glauben, daß es auf dem Gills gut gelungen war - denn von Diskette hätte ich sein Programm nicht abspielen können: Das GENIE IIIs erzeugt die "Musik" direkt aus der Speicherzelle 3860h mittels eines Knacklautes, während der TRS-80 sie über seinen Kassettenport 255 (OFFH) hergibt (siehe mein "3.Assemblerprogramm" im INFO 15, Seite 15).

Daß diese beiden als "Musikinstrumente" nicht kompatibel sind, merkte ich aber erst, als ich unbekümmert Arnulfs Listing eingegeben und (mit ZEUS) assembliert hatte und dann dem Verstärker meines TRS-80 die zackige Melodie abverlangte - es kam zu einem Hänger... Erneuter Briefwechsel (siehe "Das TSCRIPS-Drama"; aber dies- nur einmal); darauf erst erfolgte besagte Aufklärung (man muß die Esoteriker beim Gewissen nehmen).

Suppés - pardon: Sopp's "Leichte" wurde bei der Anpassung an den TRS-80 unter meinen Händen zur "Schweren Kavallerie". Der gelungene Effekt (auch aus seinem Verstärker ertönt jetzt das "Reiterlied") beweist mir zwar, daß mein Tun nicht falsch war. Selbst ZEUS hat mir "00000 Fehler" bescheinigt. Doch glaube ich, daß es eine elegantere Lösung für meine Anpassung geben kann. Ebenso für meine "da capo"-Erfindung: Arnulf hatte die Noten für die Wiederholung des ersten Teils explizit hingeschrieben. Musikanten setzen stattdessen nur die beiden Wiederholungssymbole (senkrechte Balken durch die Notenlinien). So wollte ich es auch haben. Ich baute also ein "Da capo" ein (16 Bytes) und sparte mir dadurch die nochmalige Eingabe der gleichen 48 Integers = 96 Bytes. Es klappt zwar; aber vielleichtginge auch das eleganter zu lösen und der Originalcode um mehr als diese 80 Bytes zu kürzen. Die kleine Verlängerung der Pause durch diese Subroutine ist für scharfe Ohren unüberhörbar. (Das nachfolgende Listing gibt Arnulfs Lösung wieder. Die von mir eingebrachten Teile sind mit einem '*' versehen. Für Verbesserungsvorschläge bin ich stets dankbar.)

Unser ältestes Clubmitglied, TACITUS, der Große Schweiger, hatte die Preisaufgabe natürlich auch gelöst. Und die Lösung natürlich nicht eingesandt. Und deshalb natürlich auch keinen Sekt bekommen. * Wir Drei haben nun beschlossen, den Oberen des CLUB 80 und seinen Mitgliedern einen nahe-liegenden Vorschlag zu unterbreiten. Dieser sei - Würde, wenn sie gebührt - hier pathet-poet-isch vorgebracht:

Tacitus / Arnulf / KajoT - kurz: TAKT - erklären als unumstößlichen Fakt: Der oben beschriebene Reiter-Hit, geschrieben und assembliert Bit für Bit, sei künftig als zünftiges Hacker-Symbol das Lied des CLUB 80!

* Wohlauf & Auf's Wohl! *

```

00001 ; #####
00002 ; #           Das CLUB 80-Lied
00003 ; # für GILLS in Toene gesetzt von Arnulf Sopp
00004 ; # *für TRS80 in Szene gesetzt von Ka-Jot-bein *
00005 ; #           (Nachahmung empfohlen)
00006 ; #           = no COPYRIGHT 1987 =
00007 ; #####
00008 ;
00009 ; ORG      5200h
00010 start LD      HL,tab ;Tab.f.Tonhöhen & -längen
00011 loop0 LD      BC,512 ;Zähler f.Päuschen zw.zwei Tönen
00012 loop1 DEC     BC      ;Zähler erniedrigen
00013 LD      A,B      ;prüfen, ob er
00014 OR      C        ;schon abgelaufen ist
00015 JR      NZ,loop1 ;weiter warten auf Godot
00016 LD      A,(HL)   ;LSB der Tonhöhe laden
00017 CP      50      ;DA CAPO-Flag erreicht? *
00018 JR      Z,DACAPO ;falls ja, wiederhole! *
00019 OR      A        ;00 als Endekennung?
00020 RET     Z        ;wenn ja, zuruck ins DOS
00021 LD      C,A      ;sonst LSB nach C
00022 INC     HL      ;nächstes Byte der Tabelle
00023 LD      B,(HL)   ;als MSB der Tonhöhe nach B
00024 INC     HL
00025 LD      E,(HL)  ;LSB der Tondauer
00026 INC     HL
00027 LD      D,(HL)  ;MSB "
00028 INC     HL      ;Tabellenstelle für nächsten Ton
00029 loop2 DEC     DE  ;Tondauertzähler erniedrigen
00030 LD      A,D      ;prüfen, ob er
00031 OR      E        ;schon abgelaufen ist
00032 JR      Z,loop0 ;falls ja, nächster Ton
00033 LD      A,I      ;Spannungspegel für Port 255 *
00034 PUSH   BC       ;Tonhöhe retten
00035 CALL   knack    ;zur Tonausgabe (1.Halbwelle)
00036 POP    BC       ;Tonhöhe zurückholen *
00037 PUSH   BC       ;aber auch nochmal retten *
00038 CALL   knack    ;zur Tonausgabe (2.Halbwelle)
00039 POP    BC       ;Tonhöhe zurückholen *
00040 JR      loop2  ;weiter in der Tondauerschleife
00041 knock OUT    (OFFH),A ;Halbwelle ausgeben
00042 frequ DEC     BC  ;Frequenzzähler erniedrigen
00043 LD      A,B      ;checken, ob schon abgelaufen
00044 OR      C        ;dann ist A=00
00045 RET     Z        ;zurück, falls dem so ist
00046 JR      frequ  ;sonst Frequenz weiter halten
00047 DACAPO LD     BC,1000 ;Ausschaltung des *
00048 LD      IX,HALBIT ; DA CAPO-Flags
00049 LD      (HALBIT),BC ; mittels eines *
00050 LD      (IX+2),2 ; Index-Registers *
00051 JP      start  ; Wiederholung des 1.Teils *
00052 tab DW     150,90 ;Töne: erst Höhe, dann Dauer
00053 DW     112,120
00054 DW     112,90
00055 DW     112,30
00056 DW     112,60
00057 DW     150,45
00058 DW     178,36
00059 DW     224,30
00060 DW     150,90

```

HEFT
22
Dezember
1987

18

526B	9600	00061	DW	150,67
526F	9600	00062	DW	150,22
5273	9600	00063	DW	150,90
5277	7000	00064	DW	112,90
527B	6400	00065	DW	100,33
527F	5900	00066	DW	89,151
5283	6400	00067	DW	100,101
5287	5900	00068	DW	89,37
528B	5400	00069	DW	84,160
528F	5900	00070	DW	89,113
5293	5900	00071	DW	89,37
5297	5900	00072	DW	89,151
529B	6400	00073	DW	100,134
529F	983A	00074	DW	15000,2
		00075		
52A3	3200	00076 HALBZT	DW	50,150
52A7	9600	00077	DW	150,67
52AB	7700	00078	DW	119,28
52AF	6400	00079	DW	100,134
52B3	6400	00080	DW	100,134
52B7	6400	00081	DW	100,134
52BB	7000	00082	DW	112,90
52BF	6400	00083	DW	100,33
52C3	5900	00084	DW	89,151
52C7	6400	00085	DW	100,67
52CB	6400	00086	DW	100,67
52CF	7000	00087	DW	112,120
52D3	7000	00088	DW	112,60
52D7	5900	00089	DW	89,75
52DB	4800	00090	DW	72,180
52DF	9600	00091	DW	150,67
52E3	9600	00092	DW	150,22
52E7	9600	00093	DW	150,90
52EB	8500	00094	DW	133,50
52EF	7700	00095	DW	119,56
52F3	7000	00096	DW	112,60
52F7	5900	00097	DW	89,75
52FB	6400	00098	DW	100,67
52FF	7700	00099	DW	119,56
5303	7000	00100	DW	112,120
5307	00	00101	DB	0
5200		00102	END	start

; kurze Verschnaufpause vor
; der nächsten Paraphrase
; ← DA CAPO-Flag *

; Endekennung des Liedes

00000 Fehler

00000 Fehler

DACAPO	5235	HALBZT	52A3	frequ	522F	knack	522D
loop0	5203	loop1	5206	loop2	521A	start	5200
tab	5247						

Handwritten mark

Von 8 auf 24 Pins

Computerfreaks sind von der Seuche infiziert, immer mehr und bessere Geräte haben zu müssen. Gott sei Dank! Wer sich in diesen Zeiten einen neuen Drucker kauft, wird wohl je nach Geldbeutel auf einen mit 24 Nadeln umsteigen. Wohl getan! Bei Listings und Texten ist der Unterschied nur in den schöneren Zeichen zu erkennen, die eben von dreimal so vielen Pins gemalt werden. In der Graphik aber wird man seinen teuren Drucker gelegentlich verfluchen. Wenn man sie wie gewohnt mit einem Byte pro Punktspalte erzeugt, kommt ein Streifenmuster heraus, denn es wird dabei nur jeder dritte Pin abgefeuert. Hier ein Vergleich zwischen 8- und 24-Nadel-Modus:

Das ist 8-Pin-Graphik eines Phantasiemusters: \$
und das ist 24-Pin-Graphik desselben Musters: \$

Wie man sieht (falls man das obere Zeichen in der Reproduktion überhaupt sieht), machen Querstreifen nicht immer dick; hier machen sie reichlich dünn. Es kommt deshalb darauf an, den restlichen 16 Nadeln etwas zu tun zu geben. Dabei genügt es nicht, den einleitenden Codestring zur Definition von 24-Pin-Graphik auszugeben und dann das Datenbyte dreimal zum Drucker zu schicken. Das sähe nämlich so aus:

Dies ist 24-Pin-Graphik von Quark: \$

Da stehen nur drei eng gequetschte Graphik-Dollars übereinander. Statt dessen muß jedes Bit des auszugebenden Bytes verdreifacht werden. Dazu genügt der Akku nicht mehr. Man braucht drei Register, die hilfsweise je eines der drei senkrecht übereinanderstehenden Graphikbytes beinhalten.

Das im Anschluß an diesen Text gelistete Programm rettet daher zunächst alles, was im Programmverlauf verändert wird, außer dem Akku (das wäre überflüssig). Die Hauptsache passiert in der inneren Schleife loop2. Dreimal wird das jeweilige Bit 7 des Akkus ins Carry-Flag rotiert. Jedesmal wird dann dieses Bit nach E geschoben, dessen Bit 7 ins Carry. Von dort gelangt es ins Bit 0 von D, wobei das Bit 7 von D ins Carry geht. Und das wiederum wird zum Bit 0 von H. Simpler, aber weniger genau ausgedrückt: Der Akku wird in dreifacher Spreizung in die drei Hilfsregister rotiert.

In der äußeren Schleife loop1 werden die 8 Bits des ursprünglichen Graphikbytes abgezählt und in der beschriebenen Weise behandelt. Am Ende steht in den drei Registern H, D und E das verdreifachte Graphikbyte. Die drei Register werden nacheinander ausgedruckt, die alten Registerinhalte sodann restauriert.

Dies ist nur eine Hilfsroutine, die in ein eigentliches Arbeitsprogramm eingefügt werden muß, z. B. in TSCRIPS, das seine Sonderzeichensätze über die 8-Nadel-Graphik doppelter Dichte erzeugt. Es versteht sich, daß der einleitende Codestring zur Definition der Graphikart (Dichte, Geschwindigkeit, Anzahl der Nadeln) auf die 24 Pins angepaßt werden muß. Je nach Drucker, je nach Rahmenprogramm kann das sehr unterschiedlich aussehen.

Um überhaupt auszuprobieren, wie eine Graphik von 8 auf 24 Nadeln umgerechnet werden kann, dient das am Ende gelistete Testprogramm. Beim Label gra2 stehen die Codes, die sich zu dem oben ausgedruckten Dollar zusammensetzen. Der Rest sind SteuerCodes (abgesehen von den Texten), die je nach Drucker variieren können (hier für den NEC P6).

Man darf nun nicht erwarten, daß die 8-Pin-Graphik mit 24 Nadeln wie gemalt aussieht. Wenn die einzelnen Bits verdreifacht werden, verschwinden nur die Streifen. Das Graphikzeichen sieht ein wenig klötzchenartig aus, denn noch immer sind es nur 8 verschiedene, bloß eben verdreifachte Informationen, die eine Punktspalte ausmachen.


```

00001 ; Kernprogramm für die Konversion von 8- auf 24-Pin-Graphik
00002
0000 ES 00003 PUSH HL ;alle Register retten
0001 DS 00004 PUSH DE
0002 CS 00005 PUSH BC
0003 4F 00006 LD C,A ;Ursprungsbyte (für 8 Pins) retten
0004 0608 00007 LD B,8 ;Zähler für 8 Bits des Ursprungsbytes
0006 68 00008 loop1 LD L,B ;retten
0007 0603 00009 LD B,3 ;Zähler für Verdreifachung der Bits
0009 79 00010 loop2 LD A,C ;Datenbyte laden
000A 07 00011 RLCA ;Bit 7 -> Cy
000B CB13 00012 RL E ;Cy -> Bit 0 v. E (unteres 24-Pin-Byte)
000D CB12 00013 RL D ;dto. D (mittleres)
000F CB14 00014 RL H ;dto. H (oberes)
0011 10F6 00015 DJNZ loop2 ;bis 1 Bit verdreifacht ist
0013 45 00016 LD B,L ;Bitzähler
0014 4F 00017 LD C,A ;rotiertes Byte als neuer Ausgangswert
0015 10EF 00018 DJNZ loop1 ;bis 8 Bits verdreifacht sind
0017 7C 00019 LD A,H ;oberes Datenbyte (für 24 Pins)
0018 CDB405 00020 CALL 05b4h ;ausdrucken
001B 7A 00021 LD A,D ;mittleres Datenbyte
001C CDB405 00022 CALL 05b4h ;unteres Datenbyte
001F 7B 00023 LD A,E
0020 CDB405 00024 CALL 05b4h
0023 C1 00025 POP BC ;Register restaurieren
0024 D1 00026 POP DE
0025 E1 00027 POP HL
0026 C9 00028 RET ;falls es ein Unterprogramm war
0000 00029 END
00030

```

00000 Fehler

loop1 0006 loop2 0009

```

00001 ; Testprogramm zur Konversion von 8- auf 24-Pin-Graphik
00002
5200 00003 ORG 5200h
00004
5200 213052 00005 start LD HL,gra1 ;Codestring für ein Graphikmuster, 8 Pins
5203 CD6A44 00006 CALL 446ah ;den String auf den Drucker ausgeben
5206 217852 00007 LD HL,gra3 ;Einleit. Gr. doppelter Dichte, 24 Pins
5209 CD6A44 00008 CALL 446ah ;ausgeben
520C 216F52 00009 LD HL,gra2 ;Anfang der 8-Pin-Datenbytes
520F 0608 00010 LD B,8 ;Zähler der Datenbytes für 8 Pins
5211 C5 00011 loop0 PUSH BC ;retten
5212 E5 00012 PUSH HL ;dto. Datenzeiger
5213 4E 00013 LD C,(HL) ;Ursprungsbyte (für 8 Pins) laden
5214 0608 00014 LD B,8 ;Zähler für 8 Bits des Ursprungsbytes
5216 68 00015 loop1 LD L,B ;retten
5217 0603 00016 LD B,3 ;Zähler für Verdreifachung der Bits
5219 79 00017 loop2 LD A,C ;Datenbyte laden
521A 07 00018 RLCA ;Bit 7 -> Cy
521B CB13 00019 RL E ;Cy -> Bit 0 v. E (unteres 24-Pin-Byte)
521D CB12 00020 RL D ;dto. D (mittleres)
521F CB14 00021 RL H ;dto. H (oberes)
5221 10F6 00022 DJNZ loop2 ;bis 1 Bit verdreifacht ist
5223 45 00023 LD B,L ;Bitzähler
5224 4F 00024 LD C,A ;rotiertes Byte als neuer Ausgangswert
5225 10EF 00025 DJNZ loop1 ;bis 8 Bits verdreifacht sind

```

```

5227 7C 00026 LD A,H ;oberes Datenbyte (für 24 Pins)
5228 CDB405 00027 CALL 05b4h ;ausdrucken
5228 7A 00028 LD A,D ;mittleres Datenbyte
522C CDB405 00029 CALL 05b4h
522F 7B 00030 LD A,E ;unteres Datenbyte
5230 CDB405 00031 CALL 05b4h
5233 E1 00032 POP HL ;Datenzeiger der 8-Pin-Graphik
5234 23 00033 INC HL ;erhöhen
5235 C1 00034 POP BC ;Bytezähler
5236 10D9 00035 DJNZ loop0 ;bis alle Datenbytes verdreifacht sind
5238 3E0D 00036 LD A,0dh ;String mit CR abschließen
523A C3B405 00037 JP 05b4h ;CR drucken und zurück ins DOS
00038
523D 44 00039 gra1 DM 'Das ist 8-Pin-Graphik eines Phantasiemusters:',09h
526B 18 00040 DB 1bh,'L',08h,00h ;Definition für doppelte Dichte, 8 Pins
526F 32 00041 gra2 DB 32h,32h,49h,0ffh,0ffh,49h,26h,26h,0dh ;Datenbytes
5278 75 00042 gra3 DM 'und das ist 24-Pin-Graphik desselben Musters:',09h
52A6 1B 00043 DB 1bh,'" ',21h,08h,00h,03h ;Def. für dopp. Dichte, 24 Pins
00044
5200 00045 END start ;dort Einsprung

```

00000 Fehler

gra1 523D gra2 526F gra3 5278 loop0 5211 loop1 5216 loop2 5219 start 5200



Raumschiffel... Sie haben meine Kommandozentrale zerstört.

Im letzten Beitrag zu diesem Thema wurde gezeigt, wie eine Graphik für den Drucker von 8 auf 24 Nadeln umgerechnet werden kann. Ein für den Vielschreiber alltäglicher Anwendungsfall ist TSCRIPS. Dieses Textprogramm erzeugt Sonderschriften über die Drucker-HRG. Es geht von einem Printer mit 8 oder 9 Nadeln aus. Wenn einer mit 24 Pins angeschlossen ist, entsteht ein streifiges Muster, weil nur 8 Nadeln benutzt werden. In der Folge soll gezeigt werden, wie alle 24 Pins eine selbst definierte (oder in einem IMAGEx/DAT-File ergatterte) Schrift drucken. Die Steuerodes gelten für den NEC P6. Da er Epson-kompatibel ist, kann das Programm wohl ohne Änderungen von den meisten übernommen werden, die einen 24-Nadel-Drucker haben.

Dennoch empfiehlt es sich, es zunächst nur mit einer Kopie von TSCRIPS auszuprobieren. In meinem Textprogramm habe ich nämlich schon dermaßen herumgewühlt, daß ich selbst nicht mehr weiß, ob die zu ändernden Adressen der Originalversion entsprechen. Wenn überhaupt, dann ist es Version 5.4. Bei Problemen will ich versuchen, dem verzweifelten Leser hilfreich in die Seite zu treten.

Den Kern des Programms will ich nicht mehr erläutern. Er ist in dem Artikel "Von 8 auf 24 Nadeln" beschrieben. Bevor er aber erreicht wird, setzen diese neuen Überlegungen an. Das Zeichen "*" leitet den Ausdruck einer Sonderschrift ein. Dies ist der einzige Fall von Graphik, den TSCRIPS unterstützt, wenn man den Drucker nicht direkt mit &-Codes programmieren möchte. So lag es nahe, die #-Routine zu ändern.

TSCRIPS kennt als einzigen Modus die doppelt dichte Graphik mit 8 Nadeln, die mit ESC-'L'-n-m eingeleitet wird. Es wäre möglich, aber recht aufwendig gewesen, diese vier Bytes zu erkennen, abzufangen und zu ändern. Statt dessen bedient sich dieses Programm der Möglichkeit des NEC P6, jeden Graphikmodus einem jeden anderen zuzuordnen. Deshalb wird der Codefolge ESC-'L' (doppelte Dichte, 8 Pins) dieselbe Dichte, jedoch mit 24 Pins angedichtet (ESC-'?'-'L'-33). So kann die einleitende Graphik-Definition einfach weiterhin unbeachtet auf den Drucker ausgegeben werden. Erst die nachfolgenden Graphikdaten müssen umgerechnet werden.

Bis das erste davon ankommt, muß vier Bytes lang gewartet werden: ESC-'L'-n-m. Ein Zähler wird dafür aufgesetzt. Wenn der abgelaufen ist, passiert das, was im letzten Artikel zu diesem Thema beschrieben wurde: Der Inhalt des Akkus wird um das Dreifache gespreizt ausgegeben. Das Resultat ist dieselbe Sonderschrift (oder was auch immer), jedoch ohne die Querstreifen.

Zur Demonstration hier mein (zugegebenermaßen nicht eben bescheidener) Briefkopf, oben mit 8 Pins nach alter Art, darunter mit 24 Nadeln gedruckt:



A. SOPP

Wakenitzstr. 8
2400 Lübeck 1
Telefon:
0451-79 19 26

A. SOPP

Wakenitzstr. 8
2400 Lübeck 1
Telefon:
0451-79 19 26

Angelpunkt ist das Erkennen des Zeichens "*" im Text. Es leitet einen Graphikstring sowohl ein als es ihn auch beendet. Daher kann das Ende, das Restaurieren der alten Programmlogik, nur in einem Austausch des Originals mit der Umleitung bestehen. Ein direktes Setzen oder Löschen verbietet sich. Gottlob, denn platzsparender geht es nicht.

Im hier wiedergegebenen Listing wird vorausgesetzt, daß an der Stelle 86E3 ein auszugebendes Byte an den Drucker geschickt wird (CALL 06B4h). Wer meine bisherigen TSCRIPS-Erweiterungen benutzt, muß hier CALL check7f oder eine andere CALL-Adresse einsetzen, je nach den Anpassungen auf die eigene Hardware. Und wer eine andere TSCRIPS-Version fährt, sollte sich über den passenden Eingriffspunkt vergewissern. Es ist die Stelle, wo mit OUT (FDh),A oder LD (37E8h),A ein Byte auf den Drucker ausgegeben wird.

So ganz nebenbei werden die erwähnten TSCRIPS-Erweiterungen dabei kürzer. Nur mit "*" wird Graphik eingeleitet. Da die #-Routine Umleitungen patcht, ist in allen anderen Fällen (7Fh für Randausgleich bei Proportionalsschrift, 40h als @, nicht als \$) eine Überprüfung überflüssig, ob das soeben an den Drucker gesandte Byte etwa zu einer Graphik gehört. Zukünftige Erweiterungen von TSCRIPS dürfen sich weiterhin gerne auf das Graphik-Flag 8941h beziehen. Es ist immer noch =1, wenn gerade Graphik aktiv ist, =0, wenn nicht.

Arnulf Sopp

```

00001 ;Patch in TSCRIPS, um Graphik (Sonderzeichensätze) m. 24 Pins zu erzeugen
8942 00002 ORG 8942h ;Beginn der #-Routine
8942 C00030 00003 CALL graph ;in die neue Graphik-Routine verzweigen
00004
00005 ;hier wird normalerweise ein Byte an den Drucker gesandt.
86E3 00006 ORG 86E3h
86E3 C0B405 00007 CALL 05B4h ;Byte auf den Drucker ausgeben
86E4 00008 calladr EQU 8-2 ;die Adresse ändert sich bei Graphik
00009
00010 ;ein #-Zeichen wurde erkannt: Änderungen vornehmen
3000 00011 ORG 3000h ;oder in Höhe oder so
3000 E5 00012 graph PUSH HL
3001 05 00013 PUSH DE
3002 3E05 00014 LD A,5 ;4 Zeichen: ESC-L-n-m, dann (5.) Graphik
3004 326030 00015 LD (gract),A ;als Zähler setzen
3007 215E30 00016 LD HL,gracode ;Zuordnung von ESC-L zu einem Graphikmod.
300A 3E20 00017 LD A,20h ;XOR-Operand für 1 (-) 33 (8 (-) 24 Pins)
300C AE 00018 XOR (HL) ;Modus umschalten
300D 77 00019 LD (HL),A ;neu setzen
300E 2AE486 00020 LD HL,(calladr) ;aktuelle Adresse der Druckroutine
3011 112730 00021 LD DE,cntdown ;mögliche Umleitung
3012 00022 grdevia EQU 8-2
3014 ED53E486 00023 LD (calladr),DE ;beide vertauschen
3018 221230 00024 LD (grdevia),HL
301B 215E30 00025 LD HL,grastro ;Codefolge zur Graphik mit 24 Pins
301E C06A44 00026 CALL 446ah ;auf den Drucker ausgeben
3021 01 00027 POP DE
3022 E1 00028 POP HL
3023 3A4189 00029 LD A,(8941h) ;Überschriebenen Befehl nachholen
3026 C9 00030 RET
00031
00032 ;ESC-'L'-n-m wird ausgegeben; mitZählen, bis 1. Graphikbyte kommt
3027 E5 00033 cntdown PUSH HL
3028 216030 00034 LD HL,gract ;Zähler für 4 Codes: ESC-'L'-n-m
302B 35 00035 DEC (HL) ;herunterzählen
302C E1 00036 POP HL
302D C2F405 00037 LD A,05 ;Byte ausdr., falls noch kein Gr.-Datum

```

```

00038
00039 ;ein Graphikdatum kaa em; von 8 auf 24 Nadeln umrechnen
3030 E5 00040 PUSH HL ;alle Register retten
3031 D5 00041 PUSH DE
3032 C5 00042 PUSH BC
3033 216030 00043 LD HL,gract ;Zeiger auf Zähler der Steuercodes
3036 34 00044 INC (HL) ;wurde dekrementiert: wieder auf 0 setzen
3037 4F 00045 LD C,A ;Ursprungsbyte (für 8 Pins) retten
3038 0608 00046 LD B,E ;Zähler für 8 Bits des Ursprungsbytes
303A 68 00047 grabyte LD L,B ;retten
303B 0603 00048 LD B,3 ;Zähler für Verdreifachung der Bits
303D 79 00049 grabit LD A,C ;Datenbyte laden
303E 07 00050 RLCA ;Bit 7 -> Cy
303F CB13 00051 RL E ;Cy -> Bit 0 v. E (unteres 24-Pin-Byte)
3041 CB12 00052 RL D ;dto. D (mittleres)
3043 CB14 00053 RL H ;dto. H (oberes)
3045 10F6 00054 DJNZ grabit ;bis 1 Bit verdreifacht ist
3047 45 00055 LD B,L ;Bitzähler
3048 4F 00056 LD C,A ;rotiertes Byte als neuer Ausgangswert
3049 10EF 00057 DJNZ grabyte ;bis 8 Bits verdreifacht sind
304B 7C 00058 LD A,H ;oberes Datenbyte (für 24 Pins)
304C C0B405 00059 CALL 05B4h ;ausdrucken
304F 7A 00060 LD A,D ;mittleres Datenbyte
3050 C0B405 00061 CALL 05B4h
3053 7B 00062 LD A,E ;unteres Datenbyte
3054 C0B405 00063 CALL 05B4h
3057 C1 00064 POP BC ;Register restaurieren
3058 D1 00065 POP DE
3059 E1 00066 POP HL
305A C9 00067 RET
00068
00069 ;Steuercodes für den Drucker, Zählbyte für die Steuercodes
305B 18 00070 grastrg DB 1bh,'?',',',01h,03h ;definiert ESC-'L' für 8/24 Pins
305E 00071 gracode EQU 8-2 ;1 oder 33 -> 8 oder 24 Nadeln
3060 00 00072 gract DB 00h ;Zähler für Graphik-Steuercodes
00073
0000 00074 END

```

00000 Fehler

```

calladr 86E4 cntdown 3027 grabit 303D grabyte 303A gract 3060 gracode 305E
graph 3000 grastrg 305B grdevia 3012

```

Noch ein Bildschirmschoner

Nachdem Arnulf sein Süppchen für seine Maschine gekocht hat, möchte ich hier eine Möglichkeit für die Minderbemittelten unter uns zeigen, die sich zwar kein Gills, aber wenigstens eine 80-Zeichen-Karte leisten haben. Da ist nämlich der von Arnulf vermißte Video-Schalter drauf. Wenn man an den Steuerport der Karte per OUT eine 4 ausgibt, werden die 80x24 Zeichen dargestellt. Mangels Masse (Inhalt) wird die Mattscheibe auch schön schwarz. Wer will, kann die zwei OUT's (ein- und ausschalten) in Arnulfs Listing einbauen, ich habe aber noch ein anderes, nämlich das wohl bekannte 'BLANK' etwas umgestrickt. Da jetzt kein Buffer mehr gebraucht wird, ist das Ding auch ziemlich kurz und es reagiert im Gegensatz zu Arnulfs Produkt auch, wenn ein Programm etwas ausgeben will und nicht nur auf Knopfdruck.

Alexander Schmid

```

00001 ;*****
00002 ;*
00003 ;*          'BLANK/CMD'
00004 ;*
00005 ;*          ROUTINE TO BLANK SCREEN AFTER SET PERIOD
00006 ;*
00007 ;* CARL FORD, 256 N VIRGINIA AVE, PENNS GROVE, NJ 08069 *
00008 ;*          (609)-299-0200
00009 ;*
00010 ;*          LAST REVISED - 11/12/84
00011 ;*
00012 ;* Diese Version arbeitet zusammen mit der 80-Zeichen
00013 ;* Karte von Schaidtke (es darf sonst kein Treiber
00014 ;* dafür aktiv sein!)
00015 ;*****
00016 ;
00017          ORG      2bf5h ;CSAVE + CLOAD oder sonstwo
00018
00019 enqint EQU      4410h ;INTERRUPT ENQUE ROUTINE
00020 deqint EQU      4413h ;INTERRUPT DEQUE ROUTINE
00021 time EQU        10 ;# OF 6 SECOND PERIODS
00022 keydcb EQU      4015h ;KEYBOARD DEVICE CONTROL BLOCK
00023 viddcb EQU      401dh ;VIDEO DEVICE CONTROL BLOCK
00024 keytst EQU      38ffh ;KEY-PRESSED TEST LOCATION
00025
00026
00027 ;          *** INITIALIZE ROUTINES
00028
00029 ;*** INSERT KEYBOARD & PRINT MONITORS INTO KEYBOARD QUE
00030 entry LD        HL,(keydcb+1) ;GET CURRENT KEYBD VECTOR
00031 LD        (keyjmp+1),HL ;PUT AT END OF ROUTINE
00032 LD        HL,keys ;KEYBD PROCESSING ADDRESS
00033 LD        (keydcb+1),HL ;PUT INTO DCB VECTOR
00034 LD        HL,(viddcb+1) ;GET CURRENT VIDED VECTOR
00035 LD        (vidjmp+1),HL ;PUT AT CONTINUATION
00036 LD        HL,print ;RESTORE SCREEN VECTOR
00037 LD        (viddcb+1),HL ;PUT INTO DCB VECTOR
00038 ;          *** PUT "BLANK" INTO INTERRUPT QUE
00039 LD        DE,intrtn ;INTERRUPT RTN ADDRESS
00040 CALL     enqint ;PUT INTO INTERRUPT QUE
00041 ;          *** INTERRUPT PROCESSING ROUTINE

```

HEFT
22
Dezember
1987

26

```

00042
00043 ;
00044 intrtn DB 0 ;WORK AREA
00045 DB 0 ;WORK AREA
00046 DB 255 ;# OF 6 SEC INTERVALS
00047 DB 255 ;LENGTH OF 1ST INTERVAL
00048 PUSH AF ;SAVE REGISTER
00049 LD A,(timer) ;GET CURRENT TIMER COUNT
00050 DEC A ;SHORTEN IT BY 1
00051 LD (timer),A ;SAVE IT
00052 JR NZ,exit ;TIMER NOT 0 YET
00053 ;
00054 LD A,4 ;Umschaltbyte laden
00055 OUT (0d2h),A ;Steuerport der 80Z-Karte
00056 ;
00057 CALL deact ;DISABLE 'BLANK'
00058 exit POP AF ;RESTORE REGISTER
00059 RET
00060
00061 ;
00062 ;
00063 deact LD DE,intrtn ;VECTOR TO ROM ROUTINE
00064 CALL deqint ;REMOVE FROM QUE
00065 RET ;BACK TO CALLING PROGRAM
00066
00067 ;
00068 keys PUSH AF ;SAVE REGISTER
00069 LD A,(keyst) ;GET TEST LOCATION
00070 CP 0 ;KEY PRESSED
00071 CALL NZ,restor ;RESTORE SCREEN, ETC
00072 POP AF ;RESTORE REGISTER
00073 keyjmp JP 0 ;TO KEYBOARD DRIVER
00074
00075 ;
00076 print PUSH AF ;SAVE REGISTER
00077 CALL restor ;RESTORE SCREEN
00078 POP AF ;RESTORE REGISTER
00079 vidjmp JP 0 ;TO PRINT DRIVER
00080
00081 ;
00082 restor EXX ;SAVE REGISTERS
00083 LD A,(timer) ;CHECK TIME LEFT
00084 CP 0 ;SCREEN ALREADY BLANKED?
00085 JR NZ,reset ;IF NOT, GO
00086 LD A,0 ;Umschaltbyte
00087 OUT (0d2h),A ;Steuerport der 80Z-Karte
00088 ;
00089 LD DE,intrtn ;ROUTINES ADDRESS
00090 CALL enqint ;RESTORE TO INTERRUPT QUE
00091 ;
00092 reset LD A,time ;GET DELAY
00093 LD (timer),A ;RESET DELAY COUNTER
00094 EXX ;RESTORE REGISTERS
00095 LD A,0 ;CANCEL KEYPRESS
00096 RET ;BACK TO CALLING ROUTINE
00097
00098 ;
00099 timer DEF B time ;TIMING COUNTER
00100
00101 END entry ;AUTO EXECUTE FROM "CMD"

```

Ein weiterer Bildschirmschoner

Seiten kommt Resonanz auf etwas, das man für den Club schreibt. In der 38. Ausgabe der Zeitung des Münchner Clubs war es der Fall. Alexander Schmid stellte ein Programm vor, das nach einer gewissen Zeit den Bildschirm ausschaltet. Er konkretisierte damit meinen Vorschlag, der freilich unverändert nur auf dem Genie 3s läuft. Alexander ist auch bei uns Mitglied, deshalb vermute ich, daß sein Beitrag auch in dieser Nummer des Infos steht. Seine Routine ist meiner insoweit überlegen, als nicht nur der User den Screen wieder einschalten kann. Wenn ein Programm meint, es müsse etwas anzeigen, sieht man den Bildschirm ebenfalls wieder. Das hat Sinn und Verstand.

Deshalb war ich mit meinem Programm überhaupt nicht mehr zufrieden und schrieb es einfach neu. Die Umleitung für die Interrupts ist dieselbe. Nach einer guten Minute knipsen sie den Bildschirm aus. Angemacht wird er in der neuen Version nicht mehr nur durch einen Tastendruck, sondern wann immer irgendwer irgendwas von der Konsole will. Dabei wird jetzt übrigens auch die Graphik berücksichtigt, die in der ersten Version nicht mit abgeschaltet wurde.

Schönheitsfehler: Es werden *nur* die I/O-Routinen beachtet. Wenn etwa eine Graphik eingeblendet wird, kriegt das Programm nichts davon mit. Noch einer: Der Fernseher geht auch dann an, wenn etwas auf den Drucker ausgegeben wird. Ihn auszuschließen, hätte mehr Aufwand mit entsprechendem RAM-Verschleiß bedeutet. Der Drucker läuft so selten, daß man das in Kauf nehmen kann.

Ohnehin bleibt die Abschaltung des Bildschirms aktiv, auch wenn eine Ein- oder Ausgabe (Tastatur, Bildschirm, Drucker) aktiviert wird. Rd. 100 Sekunden später ist der Bildschirm wieder dunkel.

Zusätzlich kann dieses Feature auch wieder ausgeschaltet werden. Wenn das Programm mit irgendeinem Argument nach Komma oder Blank aufgerufen wird, wird es deaktiviert. Es erschien mir übertrieben, extra zu checken, ob das Zeichen nach dem Programmnamen Y für Yes oder J für Ja oder N für Nein bzw. No lautet. Auch ein A für Abschalten schaltet ab, allerdings auch ein E für (erfolglos) Einschalten. Der Programmname allein schaltet jedenfalls ein.

Die diversen Ladeadressen (Jewalls nach ORG) mögen verwirren. Es sind Speicherplätze, die z. Zt. in meinem DOS gerade noch frei sind. Morgen vielleicht nicht mehr. Die ORG-Adressen können beliebig geändert werden, wenn man an den bezeichneten Stellen keinen freien Speicher hat.

Der Teil, der zunächst im Sektorpuffer liegt, um von dort in das RAM im Tastatur-Adressbereich verladen zu werden, konnte nicht einfach mit ORG 3B80h für diese Adressen definiert werden. Er wäre an der beim Laden noch selektierten Tastatur zerschellt (Fehlermeldung "Ladeversuch auf ROM-Speicherplatz").

Nach Alexanders Beitrag und meiner früheren Version braucht zur Programmlogik nichts mehr erklärt zu werden. Erwähnenswert sind nur die beiden Kuckuckseier in 4601h und 03CAh. An 4600h (innerhalb der INT-Serviceroutine) steht ein CALL nach 45BEh, der umgeleitet wird (Adress-Operand in 4601/2h). An dieser Stelle sind alle Register gerettet, so daß man frei darüber verfügen kann.

Ähnlich verhält es sich für 03CAh. An 03C9h wird HL mit der Adresse 03DDh geladen (Operand in 03CA/Bh). Sie kommt als gemeinsame RET-Adresse aller I/O-Routinen auf den Stack. Wenn dort nun die Adresse der Umleitung steht, wird eben statt dessen diese aufgerufen. Sie endet dann mit einem JP nach 03DDh. In beiden Fällen wird HL mit der Fortsetzungsadresse geladen. Dorthin gelangt die jeweilige Umleitung mit JP (HL) am Ende der Prozedur.

Arnulf Sopp

```

00001 ; .....
00002 ;   Bildschirm- und -ausknipser zur Schonung der Bildröhre
00003 ;   erweiterte Version für das Genie III
00004 ;
00005 ;   nach einer Anregung von Alexander Schaid
00006 ;
00007 ;   Arnulf Sopp
00008 ; .....
00009
00010 ; jegliche I/O soll den Bildschirm und ggf. die Graphik wieder anschalten
03CA 00011   ORG   03cah   ; Rückkehradresse aller I/O-Routinen
03CA BF05 00012   DW    switch   ; Umleitung setzen
00013
00014 ; ein Interrupt soll ggf. den Bildschirm und die Graphik ausschalten
4601 00015   ORG   4601h   ; CALL-Adresse in der INT-Routine
4601 BF05 00016   DW    switch   ; Umleitung setzen
00017
00018 ; Switch-Routine, des das RAM "hinter" der Tastatur zugänglich macht
05BF 00019   ORG   05bfh   ; in der geänderten Drucker-Routine
05BF F5 00020   switch  PUSH  AF    ; Akku und Flags retten
05C0 C1 00021   POP    BC      ; aber den Stack sauberhalten
05C1 57 00022   LD     D,A      ; und für die Umleitung puffern
05C2 DBFA 00023   IM    A,(0fah) ; Systembyte 1
05C4 5F 00024   LD     E,A      ; retten
05C5 F601 00025   OR    01h      ; RAM im mm-I/O-Adreßbereich selektieren
05C7 F3 00026   DI          ; Interrupts maskieren
05C8 D3FA 00027   OUT   (0fah),A
05CA C3803B 00028   JP    int_io-offset ; UP bei gemeinsamem Einsprung aufrufen
05CD 78 00029   exit  LD     A,E      ; evtl. geändertes Systembyte 1
05CE D3FA 00030   OUT   (0fah),A      ; neu ausgeben
05DD C5 00031   PUSH  BC      ; ehem. Akku und Flags
05D1 F1 00032   POP   AF      ; restaurieren
05D2 FB 00033   EI          ; Interrupts wieder zulassen
05D3 E9 00034   JP    (HL)      ; weiter im Programm
00035
00036 ; Umleitung der Interrupt-Service-Routine zum Verwalten des Zählers
4200 00037   ORG   4200h   ; im Skt.puffer, wo es nicht dumm auffällt
0680 00038   offset EQU  $-3b80h ; Abstand zur Arbeitsadresse des Programms
4200 E3 00039   int_io EX    (SP),HL ; HL (- RET-Adresse (hinter dem Caller))
4201 7C 00040   LD     A,H      ; MSB der RET-Adresse
4202 FE46 00041   CP    46h      ; kommt der CALL aus der INT-Routine?
4204 E3 00042   EX    (SP),HL ; Stack restaurieren
4205 2021 00043   JR    NZ,io    ; falls nein (dann aus der I/O-Routine)
00044
00045 ; Interrupt-Umleitung: ggf. den Bildschirm und die Graphik ausschalten
4207 210000 00046   LD     HL,0000h ; INT-Zähler laden
4208 00047   counter EQU  $-2 ; verändert sich
420A 23 00048   INC   HL      ; erhöhen
420B CB64 00049   BIT   4,H      ; 1000h schon erreicht?
420D C8A4 00050   RES   4,H      ; nur bis 0FFFh zulassen
420F 22883B 00051   LD     (counter-offset),HL ; Zähler neu laden
4212 21BE45 00052   LD     HL,45beh ; Fortsetzungsadresse der INT-Routine
4215 78 00053   LD     A,E      ; ja, Input von Systembyte 1 (Port FA)
4216 2809 00054   JR    Z,ret_int ; falls 1000h noch nicht erreicht
4218 E682 00055   AND   82h      ; Bildschirm- und Graphik-Bit isolieren
421A 32AE38 00056   LD     (vid_bit-offset),A ; zur Erinnerung dort patchen
421D 78 00057   LD     A,E      ; Systembyte 1
421E E67D 00058   AND   7dh      ; Bildschirm- und Graphikbit rückerlösen
4220 5F 00059   return LD  E,A ; neues Systembyte 1
4221 F601 00060   ret_int OR  01h ; mm-I/O bleibt selektiert
4223 D3FA 00061   OUT   (0fah),A ; Systembyte neu ausgeben
4225 C3C005 00062   JF    exit     ; dort weitermachen
00063
00064 ; Umleitung für alle I/O-Routinen: ggf. den Bildschirm wieder einschalten

```

```

4228 AF 00065   XOR   A      ; A (- 00
4229 B2 00066   OR    D      ; ist etwas eingelesen/ausgegeben?
422A 78 00067   LD    A,E      ; Systembyte 1
422B 2808 00068   JR    Z,ret_io ; falls nein (I/O-Operation beenden)
422D F600 00069   CR    0Ch      ; evtl. Bildschirm und Graphik einschalten
422E 00070   vid_bit EQU  $-1 ; Bits 1 und 7 hier gepuffert
422F 210000 00071   LD     HL,0000h ; Zähler fängt neu an
4232 22883B 00072   LD     (counter-offset),HL ; frischen Zähler aufsetzen
4235 21D003 00073   ret_io LD  HL,03ddh ; Fortsetzungsadr. aller I/O-Routinen
4238 1BE6 00074   JR    return   ; I/O Routine normal fortsetzen
00075   endprog
00076
00077 ; Initialisierung: Feature en- oder disablen
423A C0D54C 00078   start CALL 4cd5h ; Befehlszeiger auf Trennzeichen stellen
423D 2812 00079   JR    Z,initial ; falls nur CR folgt
423F 7E 00080   LD    A,(HL) ; noch ein Zeichen: welches?
4240 FE4A 00081   CP    'J'      ; J für JA?
4242 280D 00082   JR    Z,initial ; falls ja
00083
4244 21BE45 00084   LD     HL,45beh ; alte Fortsetzungsadresse der INT-Routine
4247 220146 00085   LD     (4601h),HL ; restaurieren
424A 21D003 00086   LD     HL,03ddh ; Fortsetzungsadresse aller I/O-Routinen
424D 22CA03 00087   LD     (03cah),HL ; restaurieren
4250 C9 00088   RET          ; weiter im DOS oder in BASIC
00089
4251 210042 00090   initial LD  HL,int_io ; Beginn des Umleitungsprogramms
4254 118038 00091   LD     DE,3b80h ; Zieladresse "hinter" der Tastatur
4257 013A00 00092   LD     BC,endprog-int_io ; Länge des Programms
425A CDA006 00093   CALL 06e0h ; RAM "hinter" der Tastatur selektieren
425D E080 00094   LDIR   ; Programm dorthin übertragen
425F C3A806 00095   JP    06abh ; wieder mm-I/O selektieren und raus
00096
423A 00097   END   start ; dort Einsprung

```

00000 Fehler

```

counter 4208   endprog 423A   exit 05CD   initial 4251   int_io 4200   io 4228
offset 0680   ret_int 4221   ret_io 4235   return 4220   start 423A   switch 05BF
vid_bit 422E

```

Etwas für die Leute, die nie Zeit haben

31

Wenn man seine BASIC-Programme schneller machen will, benutzt man meistens Integers. Wenn man 'logisch' vorgeht, kann man dabei noch etwas mehr rausholen. Der Ausdruck $X=X+1$ kann z.B. durch $X=-NOT X$ und $X=X-1$ durch $X=NOT-X$ ersetzt werden.

Die Sache ist ganz einfach:

```
5 = 00000101b
-5 = 11111011b (Zweierkomplement: NOT 5 plus 1)
NOT-5 = 00000100b = 4
```

Wer jetzt noch (fast) ganz genau nachmessen will, um wieviel seine Programme jetzt schneller sind, kann das mit folgendem Trick tun:

In den Speicherstellen 407CH - 407EH steht der maximale Wert für die Stunden, Minuten und Sekunden der Echtzeituhr, in 44CBH steht, nach wievielen 25ms-Interrupts die Uhr weitergestellt werden soll. Wenn man nun in 407CH - 407EH jeweils 99 und in 44CBH eine 1 schreibt, erhält man eine Uhr, die im 40stel Sekunden-Takt bis 999999 zählt (wem die knapp 7 Stunden nicht reichen, der soll sich eine CRAY ins Wohnzimmer stellen).

```
100 ' Genaue Laufzeitmessung
110 '
120 ' 407CH - 407EH: Tabelle zum Erhöhen der Uhrzeit
130 ' : (59 Sekunden, 59 Minuten, 23 Stunden)
140 '
150 ' 44CBH : Anzahl der Aufrufe
160 ' : (40 entspricht einer Erhöhung pro Sekunde)
163 '
165 ' 44CCH : Interruptzähler für Uhr
170 '
180 ' 4041H - 4043H: Speicher für Uhrzeit
190 '
200 CMD"t"
210 N=16508:POKE N,99:POKE N+1,99:POKE N+2,99 ' Tabelle
220 N=17611:POKE N,1:POKE N+1,1 ' Interrupts
230 N=16449:POKE N,0:POKE N+1,0:POKE N+2,0 ' Uhr stellen
240 '
250 CMD"r":FOR N=1 TO 10000:NEXT:CMD"t" ' oder sonstwas
260 '
270 N=16449:A=PEEK(N+2)*10000+PEEK(N+1)*100+PEEK(N)
280 PRINT A/40"Sekunden"
290 CMD"r"
300 END
```

Wichtig: Wenn die Messung genau sein soll, müssen die Laufwerke stehen! (Solange sich die Floppys drehen sind die Interrupts blockiert und die Uhr läuft nicht).

Vor dem eigentlichen Programm also eine FOR-NEXT-Schleife o.ä. einbauen, oder wie oben die Zeit mit POKE stellen.

Für ' $X=X+1$ ' in der Schleife in Zeile 250 habe ich 18.3 Sekunden gemessen, für ' $X=-NOTX$ ' aber nur 17.375 Sekunden!

Das ist ungefähr 5% schneller (bei 1.78 MHz wird's wohl genauso aussehen, auch wenn es länger dauert).

Alexander Schmid

Es geht noch schneller

32

Als ich etwas im GDOS rumgestöbert habe, habe ich einen Trick entdeckt, mit dem man seinen lahmen (Basic-) Programmen nochmal ins Kreuz treten kann:

Beim GDOS 2.4 ist u.a. ein Programm namens NEW dabei, welches eine Basic-Erweiterung darstellt. Unter den zusätzlichen Befehlen findet sich auch &KEYOFF, mit dem man die Tastatur abschalten kann. Alter Hut dachte ich zuerst, mit POKE 16405,0 hat man doch dasselbe. Denkste! &KEYOFF ist noch schneller und außerdem funktionieren INKEY\$ und die BREAK-Taste!

Das tolle an diesem Nachbrenner ist, daß er nur 9 Bytes lang ist (Tatsache). Um das Ding einzubauen zu können braucht man allerdings einen Banker, da hierfür ein CALL im ROM ungeleitet wird. Bei 1D1Eh steht ein CALL 0358h zur Tastaturabfrageroutine, die ja die große Bremse ist. Hierhin wird nun die Adresse des kleinen Zusatzprogramms geschrieben (bei NEW 3352h, kann aber beliebig sein), welches folgendermaßen aussieht:

```
3352 3A4038 LD A,(3840h) ; ENTER, BREAK usw. abfragen
3355 E604 AND 04 ; BREAK ?
3357 CB RET Z ; nein, weiter im Programm
3358 C35803 JP 0358h ; ja, BREAK auswerten
```

Einfach, aber genial, was sich die Leute bei Phoenix da ausgedacht haben. Wenn die BREAK-Taste nicht gedrückt wird, wird die übrige Tastatur erst gar nicht abgefragt; nur bei INKEY\$ oder INPUT wird der Rest auch noch abgeklappert, da dafür andere Routinen zuständig sind.

In Basic könnte die Sache so aussehen:

```
100 ' &KEYOFF
110 '
120 PRINT TIME$:FOR X=1 TO 10000:NEXT:PRINT TIME$:PRINT
130 '
140 WOHIN=&H2BF5 ' beliebig
150 FOR X=0 TO 8:READ A:POKE WOHIN+X,A:NEXT
160 DATA 58,64,56,230,4,200,195,88,3
170 USX(0)=8448:USX(2)=8704:USX(4)=201 ' INTEGER !!
180 USX(1)=WOHIN:USX(3)=&H1D1F ' 7455 dez.
190 DEFUSR=VARPTR(USX(0)) ' Disk-Version
190 A=VARPTR(USX(0)):POKE 16526,A-INT(A/256):POKE 16527,A/256
200 A=USR(0)
210 '
220 PRINT TIME$:FOR X=1 TO 10000:NEXT:PRINT TIME$
230 END
```

Die etwas umständliche Methode, die zwei Bytes nach 1D1Fh/1D20h zu bringen ist notwendig, weil die Bytes auf einmal dorthin müssen. Mit POKE &H1D1F,XX:POKE &H1D20,YY hängt sich der Rechner leider auf und zwar aus folgendem Grund: Ursprünglich steht dort 58 03, unsere Routine steht z.B. bei FFF0h. Nach dem ersten POKE steht dort F0 03. Wenn der Interpreter jetzt die Tastatur abfragt landet er im Acker bei 03F0h und rührt sich nicht mehr, statt auch den 2. Wert zu POKEn und nach FFF0h zu springen.

Wie schon gesagt kann WOHIN irgendwo im RAM liegen. Die neun Bytes dürfen aber nie irgendwie überschrieben werden, da man sich sonst nur noch an den Reset-Knopf halten kann. Für Disk-Besitzer bietet sich z.B. ZBF5h (CSAVE-Routine) hervorragend an.

Noch was für die Statistiker:

FOR X=1 TO 50000:NEXT : Disk-Basic 40.125s
 (REAL-Zahlen) POKE 16405,0 26.95s
 &KEYOFF 18.65s.

Alexander Schmid



Ein Fehler in SYS28/SYS

Als ausgesprochener Fan von ruinösen Spontankäufen habe ich mir gestern einen bidirektionalen Traktor für den NEC P6 zugelegt. Nicht daß ich etwa allzu viel zu plotten hätte. Es gibt jedoch ein paar andere Features des P6, die ohne Rückwärtstransport des Papiers unschöne Folgen haben können.

Der unidirektionale Traktor ist jetzt nicht mehr nötig. Braucht ihn jemand? Wer sich den NEC P6 kaufen möchte, müßte ihn extra gegen Aufpreis bestellen. Meiner ist billiger. Aber zur Sache:

SYS28 von G-DOS bietet einige mehr oder weniger nützliche Routinen an, die sich auf den Drucker beziehen. Sie müssen je nach Modell gepatcht werden. Das wollte ich gleich gründlich tun und disassemblierte daher SYS28 zunächst. Auffällig waren einige unmotivierte NOPs, die wohl auf früherem Gewühle by Trommelschläger bzw. Phönix beruhen. Bei näherem Hinsehen (wofern das bei ob solch dilettantischer Programmieretechnik verweintem Auge noch drin war) fanden sich massenweise katastrophale Klümmzüge, die nach Kürzung schreien. SYS28 ist jetzt ca. 300 Bytes kürzer, wobei zu bedenken ist, daß der reine Programmcode ohne Tabellen überhaupt nur ca. 600 Bytes betrug.

Am gravierendsten war ein Fehler, der ausgerechnet beim Library-Befehl DR (Ausgabe von Text aus der DOS-ready-Ebene) auftritt: ESC (CHR\$(27)) und Codes kleiner als 20h (Leerzeichen), die nach '#' eingegeben werden können, werden nur als 1. Zeichen im Textstring erkannt. Sofort danach wird der Text ausgedruckt, auch wenn nochmals '\$' (für ESC) oder Pound folgen sollte. Das werden dann eben gewöhnliche Dollars oder Pound-Zeichen.

Im Listing ist eine DR-Routine wiedergegeben, die den Vorzug hat, zu funktionieren (ab Label dr). Sie ruft zur Feststellung der Druckerbereitschaft das UP testprt auf. Auch dort habe ich einiges geändert. Jetzt werden hier alle denkbaren Drucker-Fehler berücksichtigt. Übrigens auch mit der zutreffenden Meldung, was vorher auch nicht der Fall war.

Da SYS28 jetzt total umgebaut ist, darf der geneigte Leser sich nicht durch die anderen Ladeadressen beirren lassen. Was nicht für den Befehl DR bzw. die Feststellung des Druckerstatus von Belang ist, ist mit LIST OFF ausgespart, was an den springenden Zellennummern zu erkennen ist.

Der Rest von SYS28 wurde einfach nur kürzer und schneller, zu korrigieren war nichts. Deshalb möchte ich nicht das komplette neue File vorstellen. Wer Interesse daran hat, kann mich antriggern.

4028	00D105	00044	testprt	CALL	m05d1	;Drucker bereit?
4028	C8	00045	RET	Z		;falls ja
402C	E5	00046	PUSH	HL		;Befehlszeiger retten
402D	B7	00047	OR	A		;Drucker ausgeschaltet?
402E	0EAF	00048	LD	C,teznopr&0ffh		;Text 'Drucker aus'
4030	2811	00049	JR	Z,display		;falls ja
4032	07	00050	RLCA			;Drucker busy?
4033	0E91	00051	LD	C,tezbussy&0ffh		;Text dazu
4035	380C	00052	JR	C,display		;falls ja
4037	07	00053	RLCA			;out of paper?
4038	0E96	00054	LD	C,teznopap&0ffh		;Text dazu
403A	3807	00055	JR	C,display		;falls ja
403C	07	00056	RLCA			;ready?
403D	0EB3	00057	LD	C,teznrdy&0ffh		;Text dazu
403F	3002	00058	JR	NC,display		;falls nein
4041	0EA2	00059	LD	C,tezdesl&0ffh		;sonst ist der Drucker deselektiert
4043	21884D	00060	display	LD	HL,tezdrc	;Text 'Drucker'
4046	CD6744	00061	CALL	m4467		;anzeigen
4049	69	00062	LD	L,C		;HL (- Text je nach den
404A	CD6744	00063	CALL	m4467		;Fehlertext anzeigen
404B	2ECO	00064	LD	L,beep&0ffh		;abschließender Piepser, CR
404F	CD6744	00065	CALL	m4467		;ausgeben
4052	E1	00066	POP	HL		;Befehlszeiger
4053	CD4900	00067	CALL	m0049		;auf Tastendruck warten
4056	FE0D	00068	CP	0dh		;ENTER gedrückt?
4058	28CE	00069	JR	Z,testprt		;falls ja
405A	3E39	00070	LD	A,39h		;sonst erzwungene Beendigung der Funktion
405C	C9	00071	RET			
		00072				
405D	CD284D	00073	dr	CALL	testprt	;Druckerbereitsch. testen, evtl. Meldung
4060	C0	00074	RET	NZ		;falls erzwungene Beendigung der Funktion
4061	CD054C	00075	CALL	m4cd5		;auf nächsten Befehlsparameter stellen
4064	7E	00076	drloop	LD	A,(HL)	;nächstes Zeichen
4065	FE24	00077	CP	'\$'		; '\$' für ESC?
4067	2002	00078	JR	NZ,nodoll		;falls nein
4069	3618	00079	LD	(HL),1bh		; '\$' durch ESC ersetzen
406B	FE23	00080	nodoll	CP	'\$'	; '\$' stellvertretend für CTRL?
406D	2003	00081	JR	NZ,nopound		;falls nein
406F	23	00082	INC	HL		;ja, nächstes Zeichen
4070	C866	00083	RES	6,(HL)		;CTRL-Zeichen erzeugen
4072	7E	00084	nopound	LD	A,(HL)	;evtl. geändertes Byte laden
4073	23	00085	INC	HL		;nächste Stelle im Befehlsstring
4074	CD8405	00086	CALL	m05b4		;Byte ausdrucken
4077	FE0D	00087	CP	0dh		;CR?
4079	C8	00088	RET	Z		;Ende, falls ja
407A	18E8	00089	JR	drloop		;oben weiter
		00090				
4088	44	00098	tezdrc	DM	'Drucker',03h	
4091	62	00099	tezbussy	DM	'busy',03h	
4096	6F	00100	teznopap	DM	'ohne Papier',03h	
40A2	64	00101	teznrdy	DM	'deselektiert',03h	
40AF	61	00102	teznopr	DM	'aus',03h	
40B3	6E	00103	teznrdy	DM	'nicht bereit',03h	
40C0	1E	00104	beep	DB	'ah,07h,0dh	;Rest der Zeile löschen, Beep, CR

00000 Fehler

HEFT
22
Dezember
1987

33

34

Wer keine Echtzeituhr eingebaut hat, kann diese Seiten überschlagen. Wer zwar eine hat, aber am Wochentag nicht interessiert ist, auch (CP/M und CALVA-DOS, das ehemalige H-DOS, berücksichtigen ihn aber). Wer jedoch seine eingebaute Kalenderuhr mit einem Befehl vollständig programmieren will, könnte es mit meinem Programm versuchen:

Es hat eine ähnliche Funktion wie die Library-Befehle ZEIT und DATUM (TIME und DATE in NewDOS-80) und eignet sich daher zum Einbau in ein SYS-File, beispielsweise als Erweiterung des ZEIT-Befehls. Die Echtzeituhr ist jedoch batterlegepuffert und muß recht selten nachgestellt werden. Für die wenigen Male sind mir die Systemprogramme zu wertvoll.

Die Eingabesyntax entspricht den Befehlen ZEIT und DATUM. Zusätzlich kann der Wochentag eingegeben werden, und zwar abgekürzt als MO, DI, MI ... Es müssen nicht alle Zeitarten programmiert werden, die Reihenfolge ihrer Eingabe ist gleichgültig. Kleinbuchstaben beim Wochentag sind zulässig. So könnte die Programmierung der RTC aussehen:

```
xyz,FR,23.10.87,17:59:00<ENTER>
```

Dabei bedeutet xyz den Namen, mit dem das Programm natürlich zunächst aufgerufen werden muß.

In einer Endlosschleife wird der Befehlsstring, der dem Programmnamen folgt, nach zulässigen Zeitangaben durchforstet. Sie werden zur Bearbeitung dem System (Uhrzeit und Datum) bzw. dem Segment für den Wochentag übergeben. Eine Überprüfung der Anzahl solcher Angaben findet nicht statt. Daher hat man die Möglichkeit, dem obigen Beispiel vor <ENTER> noch DI für Dienstag anzuhängen, falls man etwa den falschen Tag eintippte (heute ist aber Freitag). Es gilt jeweils die letzte Eingabe.

Je nach Computer und Betriebssystem können Änderungen erforderlich sein. Die Ports 5Ah (Datenport der RTC) und 5Bh (Adressport) können sich unterscheiden. Auch die Requestcodes für die Befehle ZEIT und DATUM müssen nicht mit diesem Programm übereinstimmen. Hier sollte der Leser seine Unterlagen zu Rate ziehen.

Arnulf Sopp

```

00001 ; Utility zum vollständigen Programmieren der Echtzeituhr
00002
5200 00003 ORG 5200h ;oder wo auch immer Platz ist
00004
5200 CDD54C 00005 start CALL 4cd5h ;Trennzeichen lesen, Befehlszeiger weiter
5203 200C 00006 JR NZ,nextpar ;falls dem Befehl etwas folgt
00007
00008 ;das Folgende ist eine Spezialität des CALVA-DOS
00009 ;für andere DOSes: weglassen und statt dessen oben RET Z programmieren
5205 3E8B 00010 LD A,0bh ;Requestcode für DOS-Befehl SYS,CL
5207 EF 00011 RST 28h ;Zeitanzeige durchführen und raus
00012
00013 ;Unterprogramme zur Ausführung der DOS-Befehle ZEIT und DATUM
5208 0E0A 00014 zeit LD C,0ah ;Code für ZEIT-Befehl in SYS7/SYS
520A 1802 00015 JR jp_sys7 ;dort weiter
520C 0E0B 00016 datum LD C,0bh ;Code für DATUM-Befehl in SYS7/SYS
520E 3EE9 00017 jp_sys7 LD A,0e9h ;Requestcode für SYS7/SYS
5210 EF 00018 RST 28h ;DOS-Befehl ZEIT oder DATUM durchführen
00019
00020 ;prüfen, ob Uhrzeit, Datum oder Wochentag eingegeben wurde

```

```

5211 E5 00021 nextpar PUSH HL ;Befehlszeiger retten
5212 7E 00022 LD A,(HL) ;nächstes Zeichen in Befehlsstring
5213 FE3A 00023 CP '9'+1 ;folgt im Befehl eine Ziffer?
5215 3829 00024 JR C,numeric ;falls ja (Uhrzeit oder Datum)
00025
00026 ;Wochentag wurde eingegeben: den Tag in der Tabelle aufsuchen
5217 4F 00027 LD C,A ;BC (- die beiden nächsten Zeichen)
5218 23 00028 INC HL ;nächstes Zeichen in Befehlsstring
5219 46 00029 LD B,(HL)
521A 215E52 00030 LD HL,daytab ;Tabelle der Tagesnamen
521D 3E07 00031 LD A,7 ;Zähler für 7 Tage
521F 5E 00032 seekday LD E,(HL) ;DE (- Name eines Tages)
5220 23 00033 INC HL ;Zeiger nachstellen
5221 56 00034 LD D,(HL)
5222 23 00035 INC HL ;Zeiger auf den nächsten Tagesnamen
5223 EB 00036 EX DE,HL ;HL (- Tagesname)
5224 ED42 00037 SBC HL,BC ;Tagesnamen mit dem in der Tabelle vergl.
5226 2807 00038 JR Z,founday ;falls den richtigen gefunden
5228 EB 00039 EX DE,HL ;Tabellenzeiger restaurieren
5229 3D 00040 DEC A ;Zähler erniedrigen
522A 20F3 00041 JR NZ,seekday ;falls noch ein paar Versuche frei
00042
00043 ;Aufruf von DATUM mit falschen Parametern -> Fehlermeldung und raus
522C E1 00044 error POP HL ;Stack bereinigen
522D 18DD 00045 JR datum ;DATUM mit falschen Paras: Fehlermeldung
00046
00047 ;Wochentag existiert in der Tabelle: Tag auf die Echtzeituhr ausgeben
522F 015B6A 00048 founday LD BC,6a5bh ;RTC-Reg. 6, WR-Flag, Adressport 5B
5232 ED41 00049 OUT (C),B ;Reg. 6 (Wochentage) für WR adressieren
5234 3D 00050 DEC A ;Tage werden ab 00 gezählt
5235 035A 00051 OUT (5ah),A ;Wochentag auf RTC ausgeben
5237 ED69 00052 OUT (C),L ;L=00, beendet die Programmierung
5239 010200 00053 LD BC,0002h ;2 Zeichen bis zum nächsten Teil des Bef.
00054
00055 ;wennsane Fortsetzung: Befehlszeiger nachstellen und das Ganze von vorn
523C E1 00056 goread POP HL ;Befehlszeiger restaurieren
523D 09 00057 ADD HL,BC ;Befehlszeiger auf den nächsten Teil
523E 18CD 00058 JR start ;Befehlsstring weiter interpretieren
00059
00060 ;Uhrzeit oder Datum wurde eingegeben: entsprechenden DOS-Befehl aufrufen
5240 FE30 00061 numeric CP '0' ;noch eine erlaubte Ziffer?
5242 38E8 00062 JR C,error ;falls unsinniges Zeichen: Fehlermeldung
5244 23 00063 INC HL ;Zeiger auf Trennzeichen v. Zeit o. Datum
5245 23 00064 INC HL
5246 7E 00065 LD A,(HL) ;Trennzeichen der Zeiteinheiten laden
5247 2B 00066 DEC HL ;zurück auf 1. Zeichen
5248 2B 00067 DEC HL
5249 FE2E 00068 CP '.' ;Datum? (Trennzeichen '.')
524B 2809 00069 JR Z,setdate ;falls ja
524D FE3A 00070 CP ':' ;Uhrzeit? (Trennzeichen ':')
524F 20DB 00071 JR NZ,error ;Fehleranzeige, falls nein
5251 CD0852 00072 CALL zeit ;DOS-Befehl ZEIT durchführen
5254 1803 00073 JR adjptr ;nächsten Teil des Befehls bearbeiten
5256 CD0C52 00074 setdate CALL datum ;DOS-Befehl DATUM durchführen
5259 010800 00075 adjptr LD BC,0008h ;8 Zeichen bis zum nächsten Befehlssteil
525C 18DE 00076 JR goread ;Befehlsstring weiter interpretieren
00077
00078 ;Tabelle der Wochentagsnamen
525E 53 00079 daytab DR 'SO','SA','FR','DO','MI','DI','NO'
00080
5200 00081 END start ;hier Einsprung

```


BDBS - Basic Datenbanksystem

Seit kurzem befindet sich in unserer Programmbibliothek ein Datenbanksystem für das Model 4/4P. Das Programmpaket umfaßt insgesamt 7 Programme, 1 Hauptprogramm und 6 Overlay-Programme. Es wurde von Hardin Brothers geschrieben und im März 1986 in der 80 MICRO veröffentlicht. Die Programme sind in einer übersichtlichen und strukturierten Form geschrieben und lassen sich dadurch leicht an eigene Bedürfnisse anpassen bzw. abändern, was durch die Verwendung von verständlichen Variablennamen und Kommentaren noch zusätzlich unterstützt wird.

Nachfolgend nun eine kurze Beschreibung anhand des Programm-Menüs:

1. Definition einer Datenbank

Innerhalb einer Datenbank können bis zu 500 Datensätze bearbeitet werden. Es ist jedoch nicht weiters schwierig, diese Zahl zu erhöhen. Insgesamt kann man 40 Datenfelder mit je 30 Zeichen definieren.

2. Eröffnung einer Datenbank

Eingabe des Namens und des Laufwerks der Datenbank

3. Neue Datensätze zur Datenbank hinzufügen

4. Datensätze lesen, suchen, ändern und löschen

Hier hat man die Möglichkeit, einen Datensatz entweder direkt unter der lfd. Nummer aufzurufen oder ihn nach einem einzugebenden Suchkriterium innerhalb der definierten Datenfelder suchen zu lassen. Beim Suchen ist zu beachten, daß keine Buchstabenkonvertierung vorgenommen wird. Es bietet sich bereits bei der Eingabe an, alle Worte z.B. in Großbuchstaben zu schreiben. Oder man baut sich eine entsprechende Konvertierungsroutine ein.

4. Auswahldateien bilden

Es lassen sich insgesamt sechs Datenfelder miteinander vergleichen (Vergleichsmöglichkeiten eines Datenfelds: <; <=; =; >=; >; <>) entweder mit einem anderen Datenfeld oder einer Konstanten). Zudem besteht die Möglichkeit diese sechs Datenfelder untereinander durch die logischen Operatoren AND und OR miteinander zu verknüpfen. Die Auswahl erfolgt entweder von der gesamten Datenbank, der sortierten Indexdatei oder der alten Auswahldatei. Die Auswahldatei wird auf der Diskette abgespeichert.

5. Sortierte Indexdateien bilden

Mit diesem Menüpunkt lassen sich sortierte Dateien entweder von der gesamten Datenbank oder der Auswahldatei erstellen. Nach Auswahl eines Datenfeldes wird die Datei sortiert und in einer Indexdatei abgespeichert.

6. Ausdrucke erstellen

Man kann hier entweder von allen Datensätzen, von der Auswahldatei oder der sortierten Indexdatei Ausdrucke erstellen. Diese kann man sich entweder über Bildschirm oder Drucker anschauen oder für späteren Bedarf in einer Datei abspeichern lassen.

Sicherlich kann das Programm nicht mit einer kommerziellen Datenbank konkurrieren, zumal der Basic-Interpreter bekannterweise ja nicht das Schnellste ist was es gibt. Aber für den Hausgebrauch finde ich das Programm recht gut. Vor allem braucht man keine umfangreiche Befehlssyntax kennen, um mit dem Programm arbeiten zu können. Außerdem läßt es sich leicht ergänzen bzw. ändern.

Beim Einsatz auf einem Model 4 mit 128 kB Speicher läßt sich die RAM-Disk gut als schneller Datenspeicher oder (wenn man innerhalb des Menüs öfters wechselt) zum schnellen Laden der Programm-Module einsetzen.

Vor der erstmaligen Benutzung sollte man im Programm DEFINE/OVL am Schluß der Zeile 2140 folgende Ergänzung vornehmen:

```
      DIM FIELD.NUMBER(FIELDS), LABEL$(FIELDS), EOL(FIELDS)
```

Eine weitere Änderung ist erforderlich, wenn man das Datum im Format TT/MM/JJ eingeben möchte. Hierzu müssen in MAIN/BAS in der Zeile 180 die "/" durch "." ersetzt werden. In Zeile 190 ist dieses nochmals notwendig; außerdem müssen die Vergleichszahlen "12" und "31" miteinander vertauscht werden.

Wie bereits oben erwähnt kann man das Programm für eigene Bedürfnisse ergänzen. Mir gefiel z.B. nicht, daß normalerweise nur ein Ausdruckformat und eine Auswahldatei zur Verfügung steht. Möchte man einen anderen Ausdruck oder andere Auswahlkriterien, bleibt einem nichts anderes übrig, als die bereits erstellten Dateien zu überschreiben. Auf Dauer ist dies jedoch keine brauchbare Lösung.

Ergänzung des Basic Datenbanksystems

Zur Erweiterung der Benutzungsmöglichkeiten werden die Ausdruck(/RPT) und Auswahl(/SLC) - Dateien durch Anhängen einer Nummer an den Dateinamen erweitert. Somit lassen sich bis zu je 99 versch. Dateien benutzen.

Benötigt wird die Nummer zur Erstellung von

1. versch. Ausdruckformaten (.../RPT)
2. versch. Auswahldateien (.../SLC)

Nicht erforderlich ist sie dagegen für die .../SEL-Datei. In dieser Datei werden die durch die .../SLC-Datei ausgewählten Datensätze abgespeichert. Da sich die .../SEL-Datei bei einer Ergänzung der Datensätze sowieso ändern würde, ist es nicht erforderlich diese für längere Zeit abzuspeichern. Gleiches gilt für die .../IND-Datei.

Der Datenbanknamen wird auf max. 6 Stellen begrenzt. Die restlichen zwei Stellen werden für die Nummer benötigt.

HEFT
22
Dezember
1987

3A

Erforderlich Änderungen der Programme:

Eingabe der Nummer:
in MAIN/BAS:

```
1120 ' Eingabe der Datelnummer (2stellig)
1122 NUMMER$="":
      PRINT "Geben Sie die Nummer der Datei ein (2stellig) ":
      INPUT " ==> ".NUMMER$
1124 IF LEN(NUMMER$) <> 2 THEN 1122
1126 IF VAL(NUMMER$) < 1 OR VAL(NUMMER$) > 99 THEN 1122
1128 RETURN
```

Datenbanknamen auf 6 Stellen begrenzen:

in MAIN/BAS:
Z 504: 6

in DEFINE/OVL:
Z 2002:6

Änderung der OVL-Programme/Ergänzung der Dateinamen mit Nummer

in DEFINE/OVL

```
Z 2124: GOSUB 1122: TEST$ = FILE$ + NUMMER$ + ....
Z 2190: .....: OPEN "O",1,FILE$ + NUMMER$ + ....
```

in SELECT/OVL

```
Z 2014: GOSUB 1122: TEST$ = FILE$ + NUMMER$ + ....
Z 2032: ... THEN GOSUB 1122: OPEN "O",3,FILE$ + NUMMER$ + ....
```

in REPORT/OVL

```
Z 2002: GOSUB 1122: TEST$ = FILE$ + NUMMER$ + ....
Z 2030: .... OPEN "I",1,FILE$ + NUMMER$ + ....
```

So und nun wünsche ich viel Spaß beim Arbeiten mit BDS. Eine kurze Bedienungsanleitung befindet sich beim Programmpaket. Eine ausführliche Beschreibung findet sich in 80 MICRO, März 1986.

Klaus Hermann

Noch Interesse ?

Teil zwei der Vorstellung von PASCAL.

Beim letzten Mal haben wir uns schon mit einigen Datentypen und Kontrollstrukturen beschäftigt. Heute wird das noch ein wenig vertieft. Zu den vordefinierten Funktionen, die bereits im Sprachumfang enthalten sind, gehören neben denen aus der letzten Folge auch noch :

SUCC(X)

liefert den nächst höheren Wert der Variable X zurück, die ein Typ mit Ordnungsfunktion sein muß.

PRED(X)

liefert entsprechend den nächst niedrigeren Wert.

Die Funktion SUCC() wird oft benutzt, um INTEGER-Variablen zu inkrementieren, d.h. um 1 zu erhöhen. Das ist etwas schneller als das übliche Verfahren : I := I + 1;

Das in der letzten Folge vorgestellte Programm litt am Mangel von Hilfsmitteln zur Ein/Ausgabe von Daten. Dieses Manko soll nun behoben werden. Prinzipiell geht jedes PASCAL I/O über Files vonstatten. Zwei davon sind bereits vordefiniert und bedürfen keinerlei Deklaration : die Files *INPUT* und *OUTPUT*. Das sind die Bezeichner für Terminal-I/O. Schreiben kann man nach *OUTPUT*, lesen von *INPUT*. Dabei sind diese Bezeichner physikalisch mit den Terminal I/O-Devices identisch. Weil I/O via Terminal so häufig vorkommt, kann man die Bezeichner *INPUT* und *OUTPUT* einfach weglassen. Die Prozeduren, die in PASCAL das I/O übernehmen, heißen *READ* und *WRITE*. Beide Prozeduren haben Parameterlisten, das ganze geht ähnlich über die Bühne wie man das von den BASIC-Anweisungen *INPUT* und *PRINT* kennt. Ein kleines Beispiel :

```
WRITE( Dies ist eine Textkonstante );
```

gibt eben diese Konstante am Bildschirm aus, völlig gleichwertig ist die Anweisung :

```
WRITE(OUTPUT, Dies ist eine Textkonstante );
```

Konstanten einlesen kann man natürlich nicht, da ein Einlesen ja einer Zuweisung entspricht. Als Beispiel dafür ein Programmfragment :

```
VAR x : REAL;
```

```
....
```

```
READ(X);
```

READ und *WRITE* sind sogenannte generische Prozeduren, das bedeutet, daß die Anzahl der Parameter nicht feststeht, sondern erst während der Übersetzung bestimmt wird. Man kann damit alle vordefinierten Typen behandeln, allerdings nicht alle selbst definierten (das geht nämlich auch in Pascal, Datentypen selbst basteln !). Aber zuerst wollen wir doch mal sehen, was es sonst noch an Datentypen gibt. Da wäre erst mal, unverzichtbar, das Feld oder der Vektor oder das *ARRAY*. Felder kann man von allem definieren, was einem Spaß macht, auch von Feldern oder von Files oder von Steckrüben...

Allerdings hat Pascal eine gravierende Einschränkung : die Grenzen eines Feldes müssen Konstanten sein. Dynamische Felder gibt es nicht, aber was viel feineres...

Kommen wir zur praktischen Anwendung, diesmal ein kleines Beispiel aus der Mathematik. Seien A und B Vektoren im dreidimensionalen Raum, also darstellbar durch jeweils ein Feld mit 3, sagen wir REAL, Elementen. Dann gibt es eine Verknüpfung dieser beiden Vektoren, das sogenannte Skalarprodukt, welches wie folgt definiert ist :

$$A \circ B := \text{SUMME}(A_i \cdot B_i), i = 1..3$$

(gerade jetzt wünsche ich mir ein Textsystem, das mathematische Formeln darstellen kann). Das kleine i ist als Index zu verstehen.

In Pascal sieht das Programm vielleicht so aus wie das kleine Beispiel. Dazu ist noch etwas zu bemerken : KOMMENTARE werden in Klammern eingeschlossen, und zwar entweder in die Kombination ('...') oder in die geschweiften Klammern. Die Prozeduren READLN und WRITELN sind identisch mit READ bzw. WRITE, erzeugen aber nach Eingabe bzw. Ausgabe einen Zeilenvorschub.

Im Programm ist außerdem noch eine FOR-Schleife zu sehen. Der Schleifenkörper ist dabei immer eine Anweisung. Das kann auch eine zusammengesetzte Anweisung sein. Eine zusammengesetzte Anweisung besteht aus einer Anzahl von einfachen Anweisungen, welche in die Schlüsselwörter BEGIN und END eingeklammert sind. Hört sich komisch an... Vielleicht tippt ihr das Beispiel mal ab und experimentiert ein wenig damit. Man könnte die Anzahl der ARRAY-Elemente vergrößern und rausfinden, was ein Skalarprodukt im fünfdimensionalen Raum ist.

Felder stellen ein außerordentlich nützliches Hilfsmittel dar, aber wem erzähle ich das. Bis jetzt gab es, z.B. gegenüber BASIC, noch wenig neues zu vermeiden. Betrachten wir nun eine der großen Pascal-Stärken ; Selbstdefinierte Typen. Formal dient dazu das Schlüsselwort TYPE in strengen Implementierungen muß dieses Wort vor VAR stehen, in TURBO nicht. Hinter TYPE folgt der Name des zu definierenden Typs, ein Zeichen '=' und dann die Beschreibung des Typs, beendet wird die Deklaration mit dem Semikolon.

Nehmen wir ein Beispiel. Der Typ INTEGER enthält bekanntlich sowohl positive wie auch negative Zahlen. Stellen wir uns eine Anwendung vor, bei der eine Variable nur ganzzahlige Werte zwischen 1 und 12 annehmen kann, sagen wir, sie repräsentiere einen Monat. Es ist nun sinnvoll, einen Typ für die Variable zu erfinden. Also :

```
TYPE MONAT = 1..12;
VAR M : MONAT;
```

Nun werdet ihr sagen, alles schön und gut, und wozu soll das gut sein ? Ganz einfach, falls im Programmlauf ein Fehler auftritt und die Variable M, sagen wir, einen Wert von 25 zugewiesen bekommt, hält das Programm mit einem Laufzeitfehler. Damit kann man Fehler entdecken, die einem sonst durch die Lappen gehen, und man wundert sich. Übrigens nennt man den oben definierten Typ Unterbereichstyp oder Subrange-typ. Darüber hinaus gibt es noch den sog. Aufzähltyp. Wieder ein Beispiel :

```
TYPE MONAT =
(JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC);
VAR M : MONAT;
```

Man kann dann tatsächlich schreiben : IF M = OCT THEN WRITE(OKTOBER);

An diesem Punkt kommen wir wieder zur Diskussion der Ordnungsrelationen zurück. Jeder Wert eines Aufzähltyps entspricht der Stelle, an der er steht, z.B. ist der Wert vom MAR intern 2. Deshalb ist MAR kleiner als MAY, der durch 4 repräsentiert wird (der erste Wert ist Null). Wenn man sich durch einen solchen Aufzähltyp bewegen will, kann man natürlich nicht schreiben

```
M := M + 1;
```

sondern man muß die Funktionen PRED und ORD benutzen. Hoffentlich wird der Zusammenhang klarer, wenn ihr euch das zweite Beispielprogramm anschaut.

Zum Experimentieren lädt euch herzlich ein,

Rüdiger

```
program Skalar(input, output) ;
```

```
(* Dieses Programm fordert den Benutzer auf, einige Zahlen einzugeben und berechnet daraus das Skalarprodukt. *)
```

```
(* Dargestellt werden :
```

```
FOR - Schleife
ARRAY - Typ
READ/WRITE
Kommentare
```

```
*)
```

```
CONST
  upper_bound = 3;
```

```
TYPE
  Vektor = array [ 1..upper_bound ] of real;
```

```
VAR
  a,b : vektor;
  i : integer;
  Skalar_Produkt : Real;
```

```
begin (* Hauptprogramm *)
```

```
  writeln; (* Leerzeile ausgeben *)
  writeln('Gib dreimal zwei Zahlen ein. '); (* Prompt *)
  writeln('Mehrere Zahlen in einer Zeile werden');
  writeln('durch LEERZEICHEN getrennt !');
```

```
  for i := 1 to upper_bound do (* FOR - Konstrukt. *) HEFT
    readln(a[i], b[i]); (* 2 Parameter, *) 22
```

```
  Skalar_Produkt := 0; (* dann CR/LF *)
  for i := 1 to upper_bound do (* Initialisieren ! *) Dezember
    skalar_produkt := skalar_produkt + a[i] * b[i]; (* Rechnung *) 1987
```

```
  writeln; (* Leerzeile *)
  writeln(' Das Skalarprodukt ist : ', Skalar_Produkt);
  (* beachte die 2 Parameter ! *)
  readln; (* warte auf RETURN *)
```

```
end. (* Fertig *)
```

```
program monate;
```

```
(* Dieses Programm demonstriert Unterbereichs- und
Aufzaehltypen.
*)
```

```
TYPE
```

```
MONAT = (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC);
```

```
VAR
```

```
M : MONAT;
I, Z : INTEGER;
```

```
begin (* Hauptprogramm *)
```

```
  repeat (* primitive Eingabepuefung *)
    ClrScr; (* Bildschirm loeschen *)
    Write('Gib eine ganze positive Zahl bis 100 ein : ');
    ReadLn(i);
  until (i > 0) and (i <= 100);
  z := 1; m := JAN; (* Monat auf Januar setzen *)
  while z <= i do (* Abbruch wenn z > i *)
    begin
      writeln('Monat : ', ord(m)+1); (* Wert von M ausgeben *)
      m := succ(m); (* naechster Monat *)
      z := z + 1; (* Zaehler erhoehen *)
      if ord(m) > 12 then m := JAN; (* von vorne anfangen *)
    end; (* while *)
  end. (* Program *)
```



Klar, sein Computer ist toll!
Aber unsere Hardware wird damit fertig.

"Wer kennt sie nicht, Die Vierte - FORTH. Eine feine Sache, aber mal ehrlich, irgendwie ist das doch wie eine große Kiste randvoll gefüllt mit Lego-Steinen. Solange man nichts sucht ist alles OK, aber wenn ..."

Tja, an dieser Stelle verlassen wir besser das Geschehen und blenden um zu RPNL. Als das letzte Mal etwas zu diesem Thema gesagt wurde, schrieb man das Jahr 1986. Mittlerweile sind mehr als 12 Monate vergangen. Ein Lebenszeichen ist also angebracht. Mir ist nicht bekannt ob irgend jemand auf die Idee gekommen ist, den Compiler von unserem Diskothekar abzufordern, dort liegt er nämlich für Interessierte bereit. Meine Absicht ist es, die von mir eingebauten Arithmetik-Routinen vorzustellen.

Weil es mit den vier Grundrechenarten allein nicht getan ist, sind zur Unterstützung noch einige zusätzliche Dinge von Nöten. Vorweg deshalb die Liste mit den Befehlen und deren Bedeutung:

\$PEEK	(->)	Unterprogramm: DE := (HL)
\$POKE	(->)	Unterprogramm: (HL) := DE
:= (D)	(d1 adr ->)		Lädt 32Bit Datum nach angegebene Adresse
?(D)	(adr ->	d1)	Holt 32Bit Datum von angegebener Adresse
DOUBLE	(n1 ->	d1)	Wandelt einfachgenaue Zahl in eine doppelgenaue Zahl um (vorzeichenrichtig)
DSIGN	(d1 d2 ->	sign(d2)*d1)	Überträgt das Vorzeichen von d2 auf d1
OK	(n1 ->	flag)	Prüft n1 ob kleiner Null
PRINTD	(adr ->)		Gibt den Inhalt einer doppelgenauen Variablen auf dem Monitor aus
BINASC	(d1 ->	0 a1 .. an cnt)		Wandelt d1 in einen ASCII-String um. 'cnt' ist die Anzahl der erzeugten Ziffern ohne Minuszeichen!
D+	(d1 d2 ->	d3)	Addiert zwei doppelgenaue Zahlen
D-	(d1 d2 ->	d3)	Subtrahiert zwei doppelgenaue Zahlen
D*	(d1 n1 ->	d2)	Multipliziert eine doppelgenaue Zahl mit einer einfachgenauen Zahl
MDIV	(d1 n1 ->	d2)	Dividiert eine doppelgenaue durch eine einfachgenaue Zahl

Ein weiterer Bestandteil der Arithmetik-Routinen sind die Variablen 'BASE' und 'DPL'. BASE hält die aktuelle Basis in der alle Zahlenausgaben erfolgen sollen. DPL bestimmt, an welcher Stelle innerhalb des Ziffernstrings der Dezimalpunkt eingesetzt wird. Die Zählung erfolgt von der niederwertigsten Ziffer ab in Rich-

tung der höchstwertigsten. Ein Beispiel:

```
DPL := 3    N := 123456789
```

Die Ausgabe erfolgt in der Form

```
123456.789
```

Nun aber zur Sache. Beginnen wir mit den Wort-Definitionen '\$PEEK' und '\$POKE'. Das vorangestellte Dollarzeichen soll ein Unterprogramm kennzeichnen, das mit einem Z80 CALL-Befehl aufgerufen werden kann. Ich erinnere in diesem Zusammenhang an meine Ausführungen im Club-Info No. 15, S. 57...64 zu diesem Thema. Den Abschluß bildet deshalb auch ein ganz ordinärer RET-Befehl. Zweck dieser beiden Worte ist es, die Übernahme von Daten aus Variablen, die mit DECLARE vereinbart wurden, zu vereinfachen und zu vereinheitlichen. Auf High-Level-Ebene ist die vorgehensweise folgendermaßen:

```
      :  
      :  
<variablen-name> ?  
<variablen-name> :=  
      :  
      :
```

Nicht so jedoch, wenn die gleich Aktion in Assembler erfolgen soll. RPNL unterstützt auf dieser Ebene eigentlich nur das Einbinden von Adressen auf brauchbare Weise. Alles Andere muß vom Anwender kommen. Soll nun aus einer Variablen etwas gelesen werden, muß man wissen wie diese angelegt ist. Die Definition mit 'VAR' erzeugt zunächst einmal eine Adresse, die auf das Wort <variable> im Runtime-Modul zeigt. Das ist notwendig, da ebend dieses Wort den Zeiger generiert und auf den Stack legt, mit dessen Hilfe dann '?' auf den Inhalt der Variablen zugreift. Zur Verdeutlichung ein Beispiel:

```
      :  
      :  
<variable-name> +----> <variable>  
<variable-adr> -----+ < data >  
      :  
      :
```

Das dürfte verdeutlichen was gemeint ist, wenn ich in Assembler schreibe: LD HL,<variablen-name>. Der Compiler sucht den Namenseintrag in seiner Liste. Dort findet er die Adresse und setzt diese in die Assmbleranweisung ein. Wie das Beispiel zeigt, ist das aber nicht der Zeiger auf die Daten. Den erhält man erst nach zweimaligem Inkrementieren von HL. Entsprechendes gilt fuer Konstanten, deren Aufbau identisch ist. Den Rest erledigt nun '\$PEEK' oder '\$POKE' für den Anwender. Beide Unterprogramme erwarten im HL-Register den Zeiger auf das Datenwort bzw. den zu beschreibenden Speicherplatz. Das Register DE enthält das geholte bzw. das abzuspeichernde Datenwort.

```
DECLARE  
  VAR BASE          ;hält die aktuell Zahlenbasis  
  VAR DPL           ;hält die Lage des Dezimalpunktes  
  ARRAY NUMBER 2   ;die 32Bit Variable  
DEND
```

```
10 BASE :=
```

```
CALL $PEEK  
  5E          ;$PEEK: LD      E,(HL)  
  23          ;          INC    HL  
  56          ;          LD      D,(HL)  
  23          ;          INC    HL  
  C9          ;          RET  
CEND
```

```
CALL $POKE  
  73          ;$POKE: LD      (HL),E  
  23          ;          INC    HL  
  72          ;          LD      (HL),D  
  23          ;          INC    HL  
  C9          ;          RET  
CEND
```

Die Belegung der Register ist so getroffen, daß ein einheitliches Parameterhandling auf Assemblerebene zustande kommt. Zu den nächsten beiden Worten gibt es nicht viel zu sagen, außer daß sie sich ausgiebig der bereits erwähnten Worte bedienen und so gleichzeitig als Anwendungsbeispiel dienen können.

```
CODE :=(D)          ;(d1 adr -> )  
  E1          ;PEEKD: POP    HL  
  CD $PEEK    ;          CALL  $PEEK  
  D5          ;          PUSH  DE  
  CD $PEEK    ;          CALL  $PEEK  
  D5          ;          PUSH  DE  
CEND
```

```
CODE ?(D)          ;(adr -> d1 )  
  E1          ;PEEKD: POP    HL      ;ADR  
  C1          ;          POP    BC      ;DATA-H  
  D1          ;          POP    DE      ;DATA-L  
  CD $POKE    ;          CALL  $POKE  
  50          ;          LD      D,B  
  59          ;          LD      E,C  
  CD $POKE    ;          CALL  $POKE  
CEND
```

Als nächstes folgen weitere nützliche Dinge. Hier gibt es ebenfalls nicht viel zu erklären, alle Einzelheiten sind leicht zu erkennen. Wer eine Funktion zur Absolutwertbildung vermisst, dem kann geholfen werden:

HEFT
22
Dezember
1987

46

```

PROGRAM DABS      ;(d1 -> abs(d1) )
  DUP DUP
  DSIGN
END
    
```

Ein Mangel den RPNL bisher auszeichnete, war das Fehlen des Wortes '0<'. Die Abfrage einer Zahl auf < 0 war dadurch nur sehr umständlich möglich. Die Definition sei hier gleich mitgeliefert.

```

CODE DOUBLE      ;(n1 -> d1 )
  E1      ;DOUBLE: POP   HL
  E5      ;      PUSH   HL
  11 00 00 ;      LD    DE,0000
  CB 7C    ;      BIT   7,H
  28 01    ;      JR    Z,DBL
  1B      ;      DEC   DE
  D5      ;DBL:   PUSH  DE
CEND
    
```

```

CODE DSIGN      ;(d1 d2 -> sign(d2) * d1 )
  C1      ;DSIGN: POP   BC
  CB 78    ;      BIT   7,B
  C1      ;      POP   BC
  28 0F    ;      JR    Z,DSIG
  D1      ;      POP   DE
  C1      ;      POP   BC
  21 00 00 ;      LD    HL,0000
  AF      ;      XOR   A
  ED 42    ;      SBC  HL,BC
  E5      ;      PUSH  HL
  21 00 00 ;      LD    HL,0000
  ED 52    ;      SBC  HL,DE
  E5      ;      PUSH  HL
CEND
    
```

```

CODE 0<        ;(n1 -> flag)
  E1      ;ZLT:   POP   HL
  11 00 00 ;      LD    DE,0000
  CB 7C    ;      BIT   7,H
  28 01    ;      JR    Z,ZLT1
  1B      ;      DEC   DE
  D5      ;ZLT1:  PUSH  DE
CEND
    
```

Wie schon weiter oben gesagt, ist eine Nutzung von Systemressourcen auf Assemblerebene bisher kaum möglich. Aus diesem Grund sind die Divisions- und Multiplikations-Routinen als Unterprogramme angelegt. Es kann so beim Entwurf anderer Programme ohne Probleme darauf zugegriffen werden. Ausgenommen hiervon sind Addition und Subtraktion, da diese unkompliziert sind und sich mit den bestehenden Z80-Befehlen schnell erstellen lassen.

Nun aber ein paar Worte zu '\$DIV' und '\$MULT'. Zuerst zu \$DIV. Damit der Algorithmus bei gesetztem Bit 15 im Divisor nicht aussteigt und Müll abliefern, ist eine Erweiterung des Dividenten auf 33 Bit unumgänglich. Hierfür wird der zweite Akkumulator AF benutzt. Durch diesen Trick kann der volle Wertebereich von 0...65535 für den Divisor zugänglich gemacht werden. Eine vorzeichenbehaltene Division ist aber trotzdem nur bei vorheriger Auswertung der Vorzeichen und entsprechender Korrektur des Ergebnisses möglich.

Die Rechnung erfolgt nach althergebrachter Schulmanier: Divisor subtrahieren, wenn Zwischenergebnis positiv dann weiter, sonst Rückaddition, den Dividenten eine Stelle nach Links schieben und wiederholen. Nach 32 Runden steht dann das Ergebnis fest. Während der Rechnung besteht folgende Registerkonfiguration:

```

<A'HLDE(SP)F'>
- <BC>
-----
<DE(SP)>      := Ergebnis
<HL>          := Rest
    
```

Rückspeichern der Register:

```

<BC> <-- <HL>
<HL> <-- <SP>
    
```

```

Ergebnis: <DEHL>
REST      : <BC>
    
```

Der Divisor geht verloren.

Die Anordnung der Register ist so dargestellt, wie sie vom Algorithmus benutzt werden. Mangels genügend 16bit-Register werden die niederwertigen 16 Bit des Ergebnisses während der Rechnung auf dem Stack gehalten

```

CALL $DIV      ;<DEHL>,R<BC> := <DEHL> / <BC>
  E5      ;$DIV:  PUSH  HL
  AF      ;DIV1:  XOR   A
  08      ;      EX   AF',AF
  AF      ;      XOR   A
  67      ;      LD   H,A
  6F      ;      LD   L,A
  3E 20    ;      LD   A,32 ;32 Runden sollens sein
  E3      ;DLOOP: EX   (SP),HL ;4. Register holen und
  08      ;      EX   AF',AF
  ED 6A    ;      ADC  HL,HL ;nach links schieben
  E3      ;      EX   (SP),HL
  CB 13    ;      RL   E
  CB 12    ;      RL   D
  ED 6A    ;      ADC  HL,HL
  CE 00    ;      ADC  A,00
  ED 42    ;      SBC  HL,BC
  DE 00    ;      SBC  A,00
  30 03    ;      JR   NC,DIFOK
    
```

```

09      ;      ADD      HL, BC
CE 00   ;      ADC      A, 00
3F      ;DIFOK: CCF
08      ;      EX       AF', AF
3D      ;      DEC      A
20 E5   ;      JR       NZ, DLOOP
44      ;DDONE: LD      B, H
4D      ;      LD      C, L
E1      ;      POP      HL
08      ;      EX       AF', AF
ED 6A   ;      ADC      HL, HL ;Ergebnis nachbehandeln
CB 13   ;      RL      E
CB 12   ;      RL      D
C9      ;      RET
CEND

```

Zwei Zahlen zu multiplizieren ist weit unkomplizierter als die Durchführung einer Division. Es werden einfach weniger arithmetische Funktionen benötigt: Addition mit und ohne Carry genügen. Dieser Umstand hilft die doch recht zeitfressende Operation mit dem 4. Register auf dem Stack, zu umgehen.

Die Belegung der Register während der Rechnung:

```

<DEHLIX>
+ <BC>

```

Das Endergebnis ergibt sich aus dem Inhalt von IX (Low Part) und dem Inhalt von HL (High Part). Damit die Schnittstelle am Ende wieder stimmt, sind noch einige Registerinhalte umzuspeichern:

```

DE <-- HL
HL <-- IX

```

Der Multiplikator in <BC> bleibt unverändert erhalten. Das ist besonders bei Kettenrechnungen etc. von Vorteil. Der alte Inhalt von IX geht nicht verloren.

```

CODE $MULT ; <DEHL> := <DEHL> * <BC>
3E 20     ;MULT: LD      A, 32
DD E5     ;      PUSH   IX
DD 21 00 00 ;      LD      IX, 0000
DD 29     ;NLOOP: ADD   IX, IX
ED 6A     ;      ADC      HL, HL
CB 13     ;      RL      E
CB 12     ;      RL      D
30 05     ;      JR       NC, NOSUM
DD 09     ;      ADD     IX, BC
30 01     ;      JR       NC, NOSUM
23       ;      INC     HL
3D       ;NOSUM: DEC   A
20 E8     ;      JR       NZ, NLOOP
DD E5     ;MDONE: PUSH  IX

```

```

D1      ;      POP      DE
EB      ;      EX       DE, HL
DD E1    ;      POP      IX
CEND

```

```

CODE D+   ; (d1 d2 -> d3 )
C1       ;DADD: POP    BC
D1       ;      POP    DE
E1       ;      POP    HL
E3       ;      EX     (SP), HL
19       ;      ADD    HL, DE
E3       ;      EX     (SP), HL
ED 4A    ;      ADC    HL, BC
E5       ;      PUSH   HL
CEND

```

```

CODE D-   ; (d1 d2 -> d3 )
C1       ;DSUB: POP    BC
D1       ;      POP    DE
E1       ;      POP    HL
E3       ;      EX     (SP), HL
AF       ;      XOR    A
ED 52    ;      SBC    HL, DE
E3       ;      EX     (SP), HL
ED 42    ;      SBC    HL, BC
E5       ;      PUSH   HL
CEND

```

```

CODE M*   ; (d1 n2 -> d3 )
C1       ;MULTI: POP    BC ; Multiplikator
D1       ;      POP    DE ; Multiplikant HIGH
E1       ;      POP    HL ; Multiplikant LOW
CD $MULT ;      CALL  $MULT ; Produziere große Zahl
E5       ;      PUSH  HL ; Produkt, LOW
D5       ;      PUSH  DE ; Produkt, HIGH
CEND

```

```

CODE MDIV ; (d1 n1 -> d2 )
C1       ;MDIV: POP    BC ; Divisor
D1       ;      POP    DE ; Dividend HIGH
E1       ;      POP    HL ; Dividend LOW
CD $DIV  ;      CALL  $DIV
E5       ;      PUSH  HL ; Quotient LOW
D5       ;      PUSH  DE ; Quotient HIGH
CEND

```

Wenn schon mit 9-stelligen Zahlen gerechnet werden kann, wäre es schön, sie auch auf dem Bildschirm ausgeben zu können. Einen entsprechenden Ausgabebefehl gibt es als nächstes. Er ist zweiteilig, um eine höhere Flexibilität in der Anwendung zu erhalten. Die Binärzahl wird mit dem Wort 'BINASC' in einen ASCII-String umgewandelt und Ziffer für Ziffer auf dem Stack abgelegt. Die Wandlung erfolgt unter Berücksichtigung der Systemvariablen 'BASE' und 'DPL'. Das ist besonders angenehm bei

HEFT
22
Dezember
1987

der Ausgabe von DMarks. Bei DPL = 0 fehlt der Dezimalpunkt ganz, bei DPL <> 0 erscheint er links von der angegebenen Stelle. Damit eventuelle Sonderformate (Uhrzeit etc.) einfacher erzeugt werden können, liegt zuoberst auf de Stack noch ein Ziffern-Count, der angibt, wieviele Ziffern der gesamte String lang ist (inkl. eventuellem Dezimalpunkt !). Das Vorzeichen der Zahl wird an dieser Stelle noch nicht eingesetzt (BINASC kennt nur positive Zahlen !!!). In der Regel wird dieser Zähler nicht benötigt und deshalb vor einsetzen des Vorzeichens vom Stack entfernt. Das Stringende ist erreicht, wenn die '0' hinter dem letzten ASCII-Zeichen erkannt wird.

```

CODE BINASC      ;(d1 -> 0 a1 a2 ... an cnt)
D1              ;CONV: POP DE
E1              ;CONV: POP HL
DD 21 BASE      ;CONV1: LD IX, BASE
DD 36 03 00     ; LD (IX+03), 00
FD 21 DPL       ; LD IX, DPL
FD 7E 02        ; LD A, (IX+02)
FD 77 03        ; LD (IX+03), A
01 00 00        ; LD BC, 0000
C5              ; PUSH BC
06 00          ;ALOOP: LD B, 00
DD 4E 02        ; LD C, (IX+02)
CD $DIV         ; CALL $DIV
79              ; LD A, C
FE 0A          ; CP 10 ; A > 9 ?
38 02          ; JR C, LT10
C6 07          ; ADD A, 'A'-'9'+1
C6 30          ;LT10: ADD A, '0'
4F             ; LD C, A
06 00          ; LD B, 00
C5              ; PUSH BC
FD 7E 03       ;QPNT: LD A, (IX+03)
3D             ; DEC A
FD 77 03       ; LD (IX+03), A
B7             ; OR A
20 03         ; JR NZ, NOPNT
0E 2E         ; LD C, '.'
C5             ; PUSH BC
DD 34 03       ;NOPNT: INC (IX+03) ;STELLEN INCR.
7C             ; LD A, H
B5             ; OR L
B2             ; SR D
B3             ; OR E
20 D5         ; JR NZ, ALOOP
FD 7E 03       ; LD A, (IX+03)
CB 17         ; RL A
30 CE         ; JR NC, ALOOP
06 00         ; LD B, 00
DD 4E 03       ; LD C, (IX+03)
C5             ; PUSH BC
DD 36 03 00    ; LD (IX+03), 00
FD 36 03 00    ; LD (IX+03), 00
CEND

```

Die vorzeichenrichtige Behandlung einer Zahl wird von 'PRINTD'

erledigt. Auf den Stack wird genau wie bei 'PRINT' die Adresse einer Variablen auf den Speicherplatz der Zahl erwartet, deren Inhalt auf den Stack geholt und das Vorzeichen festgehalten. Mit 'DSIGN' erfolgt dann eine Absolutwertbildung und anschließender Wandlung durch 'BINASC'. Als letztes wird geprüft, ob ein Minuszeichen ('-') eingefügt werden muß. Damit kann die Zahl endlich ausgegeben werden.

```

PROGRAM PRINTD      ;(adr -> )
? (D) DUP >R
OVER OVER DSIGN
BINASC DROP        ;Stellenzähler 'cnt' vernichten
R> 0<
IF 45 ENDIF        ; '-' := 45 !
32                 ;in jedem Fall ein führendes Blank
REPEAT
  DUP 0=
UNTIL
  OUTCHAR
LOOP
BASE ? 16 =
IF
  WRITE 'H'
ENDIF
DROP
END

```

Mit '>R' und 'R>' ist es ermöglichen, Daten vom Parameter-Stack auf den Return-Stack und wieder zurück zu transportieren. Werte, die momentan stören, können so aus dem Weg geschafft werden, ohne daß extra eine Variable dafür eingerichtet werden muß. Die Benutzung beider Worte unterliegt jedoch Regeln, deren Nichtbeachtung garantiert zum System- und Programmabsturz führen:

1. Daten, die auf den Return-Stack ausgelagert werden, müssen in jedem Fall innerhalb einer PROGRAM-Definition von dort wieder zurückgeholt werden.
2. Die FOR...LOOP Konstruktion macht ebenfalls gebrauch vom Return-Stack und legt dort die Schleifenparameter ab. Eine Überschneidung von mit '>R', 'R>' ausgelagerten Daten und den Schleifenparametern ist deshalb zu vermeiden.

'R>' und '>R' können leider nicht auf User-Ebene definiert werden, da die Startadresse des Return-Stack im Dickeyary nicht verzeichnet ist. Hier kommt nur eine Definition im Assembler-Programm des Compilers in Frage. Als Behelfslösung ist aber folgendes brauchbar:

```

PROGRAM >R
  ZEICHEN :=
END

PROGRAM R>
  ZEICHEN ?
END

```

'ZEICHEN' ist eine Systemvariable, in die mit 'INCHAR' eingelesene Zeichen abgespeichert werden! Sollte ZEICHEN nicht mehr frei sein, hilft nur eine eigene Variable.

Damit bin ich mit der Beschreibung des Arithmetik-Paketes am Ende und es kann an's experimentieren gehen.

Vergleich zweier Disk-Dateien

Normalerweise hat man keine zwei gleichen Programme mit unterschiedlichen Namen auf einer Scheibe. Doppelt genäht kostet Diskettenplatz. Es gibt jedoch Fälle, in denen das durchaus so ist. Wenn man beispielsweise nicht mehr so recht weiß, ob TS/CMD das gute alte TSCRIPS/CMD ist oder eine inzwischen erweiterte Version. Vielleicht sind sie ja identisch, so daß ein Programm gelöscht werden kann.

Ein anderer Fall ist mir heute begegnet und veranlaßte mich, die hier vorgestellte Routine zu schreiben. Die Tüftler unter euch werden gelegentlich vor demselben Problem stehen. Ich war dabei, MEMDISK/CMD zu analysieren, um es gewissen Hardware-Änderungen an meinem Computer anpassen zu können. Da wurden weite Teile erst mal in irgendwelche Banks an verschwiegene Adressen geschmuggelt, um später dort abgerufen werden zu können. Der Übersichtlichkeit halber war es sinnvoll, den Offset der jeweiligen Lade- zur Arbeitsadresse eines Programmsegments in die Labels der Sprungziele einzubeziehen.

Wer schon mit Offsets gearbeitet hat, wird darob so manchen Fluch getan haben. Oft vergißt man, dem Label den Zusatz "+offset" anzuhängen. Die Folgen sind in der Regel katastrophal. Es mußte daher eine bequeme Möglichkeit gefunden werden, festzustellen, ob die neu erzeugte Source noch gewisse Ähnlichkeiten mit dem untersuchten Programm aufweist.

Dazu wird sie assembliert und der neue Objektcode mit Hilfe dieser Utility mit dem alten verglichen. Verschiedene Assembler erzeugen u. U. verschieden lange Records. Die Zählbytes im Header unterscheiden sich dann, was die Utility als Abweichung anzeigen würde. Deshalb ist es im Zweifel ratsam, die noch unveränderte Source, wie sie beispielsweise DSMBLR/CMD vom Target-Programm ableitert, sofort wieder zu assemblieren.

Wenn die Source gestreamlined ist, wenn also beispielsweise Offsets berücksichtigt sind, und wenn sie unter anderem Namen assembliert wurde, geht's nun ab:

Die Utility wird wie gewohnt mit ihrem Namen aufgerufen, gefolgt von den beiden Namen der zu vergleichenden Files. Sie können alle Zusätze enthalten, die das DOS so für Filespecifications zuläßt. Die drei Dateinamen (die beiden letzten ggf. mit Extension!) werden wie üblich durch Blank oder Komma voneinander getrennt.

Das Programm fragt beim Einsprung zunächst, ob die Ausgabe der Abweichungen auf den Drucker erfolgen soll. Wenn man ziemlich sicher ist, daß nur wenige Bytes unterschiedlich sein dürften, genügt der Bildschirm. Wer sich selber nicht über den Weg traut, sollte die Abweichungen lieber ausdrucken lassen; es könnte eine längere Latte werden.

Für beide Programme werden sodann ein FCB eröffnet und ein Sektorpuffer reserviert. Über die zuständigen DOS-Routinen wird von beiden Programmen ein Sektor nach dem anderen geladen und verglichen. Sobald eine Abweichung auftritt, wird sie angezeigt:

rel. Skt. / rel. Byte : Abweichung ==> 0004/B7:A9C6
Diese Zahlen sind willkürlich gewählt und dienen nur als Beispiel: Im file-relativen Sektor 4 unterscheidet sich das relative Byte B7h in beiden Dateien. Das Byte B8h wird ebenfalls angezeigt, denn oft - zumindest im oben skizzierten Anwendungsfall - sind es 16-Bit-Adressen, bei denen man nicht aufgepaßt hat.

Die abweichenden Bytes werden nur aus einem File angegeben. Je nach Reihenfolge der Eingabe der Programmnamen können es die richtigen oder die falschen Bytes sein. Zur Anzeige kommen diejenigen aus dem im Befehlsstring zuerst benannten Programm.

Diese Detektiv-Utility dient einem weiteren gemeinnützigen Zweck. DSMBLR zerstückt ASCII-Strings, Folgen von DEFBS und DEFWS usw., weil seine Zeilenlänge begrenzt ist. Da das Auge mithackt, wird man die Zeilenaufteilung in aller Regel neu ordnen. Dabei geht leicht einmal ein Blank oder ein unscheinbares anderes Byte über'n Jordan. Danach verschiebt sich der ganze Rest entsprechend nach vorne. Führt man nun das Vergleichsprogramm, ergibt sich eine verächtlich hohe Anzahl von Abweichungen an fortlaufenden Sektoradressen. Am Beginn dieser Katastrophe kann man nun nach dem verschwundenen Byte fahnden. Wenn es zufällig im letzten Sektor des Files der Fall war, wird auch das kleinere Zählbyte im Record-Header als Abweichung angezeigt.

Auf die Programmlogik möchte ich hier nicht weiter eingehen. Sie erklärt sich aus den üppigen Kommentaren. Was so an den aufgerufenen DOS-Routinen passiert, kann Hartmut Grosser in seinem DOS-Buch ohnehin besser erklären als ich.

Auf wackeres Debuggen!

Arnulf Sopp

```

00001 ; Utility zum Vergleich zweier Dateien
00002 ;
00003 ; bei Abweichung werden zwei Bytes angezeigt,
00004 ; weil es sich oft um eine fehlerhafte Adresse handelt
00005 ;
00006 ; Copyright geschunken
00007
00008
00009 ORG 5200h
00010
00011 ; je nach Userwunsch Ausgabe der Abweichungen auf Bildschirm oder Drucker
00012 start PUSH HL ; Zeiger auf den Befehlsstring retten
5201 219C52 00013 LD HL,device ; Text "Drucker Ausgabe? (Y/N)"
5204 CD6744 00014 CALL 4467h ; anzeigen
5207 CD4900 00015 CALL 0049h ; Tastatur abfragen
520A CD8545 00016 CALL 45b5h ; Eingabe in Großbuchstaben unwandeln
520D FE4A 00017 CP 'J' ; Ausgabe auf den Drucker?
520F 2005 00018 JW NZ,open ; falls nein
5211 3E6A 00019 LD A,6ah ; Adress-LSB der Druckerroutine
5213 328A52 00020 LD (device),A ; Ausgabeadresse auf Drucker umleiten
00021
00022 ; für beide Files einen FCB eröffnen
00023 ; Fehler, wenn im Aufrufbefehl nicht die Namen
00024 ; zweier existierender Dateien enthalten sind
5216 E1 00025 open FOP HL ; Zeiger auf den Befehlsstring
5217 CDD54C 00026 CALL 4cd5h ; Trennzeichen Überspringen
521A 11EA52 00027 LD DE,fcbl ; FCB für das erste genannte File
521D CD1C44 00028 CALL 441ch ; den Dateinamen in den FCB übertragen
5220 3E2F 00029 error1 LD A,2fh ; Fehlercode "schlechte Parameter"
5222 C20944 00030 error2 JP NZ,44B9h ; falls kein gültiger Filename (Fehler)
5225 CDD54C 00031 CALL 4cd5h ; nächstes Trennzeichen Überspringen
5228 110A53 00032 LD DE,fcbl ; FCB für das zweite Programm
522E CD1C44 00033 CALL 441ch ; Programmnamen in den FCB übertragen
522E 20F0 00034 JR NZ,error1 ; falls kein gültiger Filename
5230 21D055 00035 LD HL,buffer2 ; Sektorpuffer für das zweite Programm
5233 45 00036 LD B,L ; logische Recordlänge = 256 Bytes (L=DD)
5234 CD2444 00037 CALL 4424h ; FCB für das zweite Programm eröffnen
5237 20E9 00038 JR NZ,error2 ; falls ein Fehler aufgetreten ist
5239 11EA52 00039 LD DE,fcbl ; FCB für das erste Programm
523C 210054 00040 LD HL,buffer1 ; dessen Sektorpuffer
523F CD2444 00041 CALL 4424h ; diesen FCB ebenfalls eröffnen

```

HEFT
22
Dezember
1987

54

```

5242 20DE 00042 JR NZ,error2 ;falls ein Fehler aufgetreten ist
00043
00044 ;Sektoren beider Dateien einlesen
00045 ;Programmende, falls die Dateien vollständig untersucht sind
5244 48 00046 LD C,B ;BC (- 0000 (ab Sektor 0 zählen)
5245 CD3644 00047 loadlop CALL 4436h ;reinen Sektor von Datei 1 einlesen
5248 F5 00048 PUSH AF ;Fehlercode und Z/NZ-Flag retten
5249 FE1C 00049 CP 1ch ;Fehlercode "Ende der Datei angetroffen"?
524B CA204D 00050 dosexit JP Z,402dh ;zurück nach DOS-Ready, falls erledigt
524E FE1D 00051 CP 1dh ;Fehlercode "hinter Ende der Datei"?
5250 28F9 00052 JR Z,dosexit ;dann auch fertig
5252 F1 00053 POP AF ;anderer Fehlercode mit Fehlerflag
5253 20CD 00054 JR NZ,error2 ;evtl. anderen Fehler anzeigen und raus
5255 110A53 00055 LD DE,fc2 ;Zeiger auf den FCB der Datei 2
5258 210055 00056 LD HL,buffer2 ;Zeiger auf deren Sektorpuffer
525B CD3644 00057 CALL 4436h ;reinen Sektor von Datei 2 einlesen
525E 20C2 00058 JR NZ,error2 ;falls ein Fehler aufgetreten ist
00059
00060 ;je einen Sektor beider Dateien vergleichen
5260 110054 00061 LD DE,buffer1 ;Sektorpuffer des 1. Programms
5263 1A 00062 complop LD A,(DE) ;1 Byte des Programms 1 laden
5264 BE 00063 CP (HL) ;dasselbe wie in Programm 2?
5265 2828 00064 JR Z,match ;falls Übereinstimmung
00065
00066 ;es wurde ein abweichendes Byte angetroffen: anzeigen
5267 E5 00067 PUSH HL ;Zeiger retten
5268 210D52 00068 LD HL,numbuff ;Puffer für Hex-Daten
526B D5 00069 PUSH DE ;Zeiger retten
526C 50 00070 LD D,B ;DE (- Sektornummer
526D 59 00071 LD E,C
526E CD634D 00072 CALL 4063h ;DE in Hex in den Puffer schreiben
5271 23 00073 INC HL ;Trennzeichen '/' aussparen
5272 01 00074 POP DE ;Pufferzeiger restaurieren
5273 78 00075 LD A,E ;relatives Byte in Sektor
5274 CP684D 00076 CALL 4068h ;in den Hexpuffer schreiben
5277 23 00077 INC HL ;Trennzeichen ':' aussparen
5278 1A 00078 LD A,(DE) ;abweichendes Byte
5279 CD684D 00079 CALL 4068h ;in den Puffer
527C 1C 00080 INC E ;Zeiger erhöhen - war es das Sektorende?
527D 2405 00081 JR Z,onlyone ;nur 1 Byte anzeigen, falls ja
527F 1A 00082 LD A,(DE) ;sonst nächstes Byte laden
5280 1C 00083 INC E ;Zeiger weiterstellen
5281 CD684D 00084 CALL 4068h ;Byte in Hex in den Puffer schreiben
5284 360D 00085 onlyone LD (HL),0dh ;Anzeigestring mit CR abschließen
5286 21B452 00086 LD HL,txbuff ;Anfang des kompletten Anzeigestrings
5289 CD6744 00087 CALL 4467h ;Abweichung auf Video/Drucker ausgeben
528A 00088 device EQU $-2 ;Ausgabeadresse für Video oder Drucker
528C E1 00089 POP HL ;Zeiger restaurieren
528D 1D 00090 DEC E ;wegen später INC E
528E 68 00091 LD L,E ;beide Zeiger auf dieselbe Stelle setzen
00092
00093 ;gegf. nach Anzeige der Abweichung weiter prüfen
528F 1C 00094 match INC E ;beide Pufferzeiger erhöhen
5290 2C 00095 INC L
5291 20DD 00096 JR NZ,complop ;falls noch nicht 256 Bytes bearbeitet
5293 11EA52 00097 LD DE,fc1 ;sonst FCB-Zeiger für Programm 1 laden
5296 210C54 00098 LD HL,buffer1 ;dto. Pufferzeiger
5299 03 00099 INC BC ;Sektorzähler erhöhen
529A 12A9 00100 JR loadlop ;nächsten Sektor bearbeiten
00101
00102 ;versch. Texte, Puffer usw.
529C 44 00103 dvcrea DM 'Druckerausgabe? (J/N) ',0dh,0dh ;höfliche Frage
5284 72 00104 txbuff DM 'rel. Skt. / rel. Byte : Abweichung ===' ;Text
52DD 78 00105 numbuff DM 'xxxx/yy:zzzz',0dh ;Puffer für numerische Angaben

```

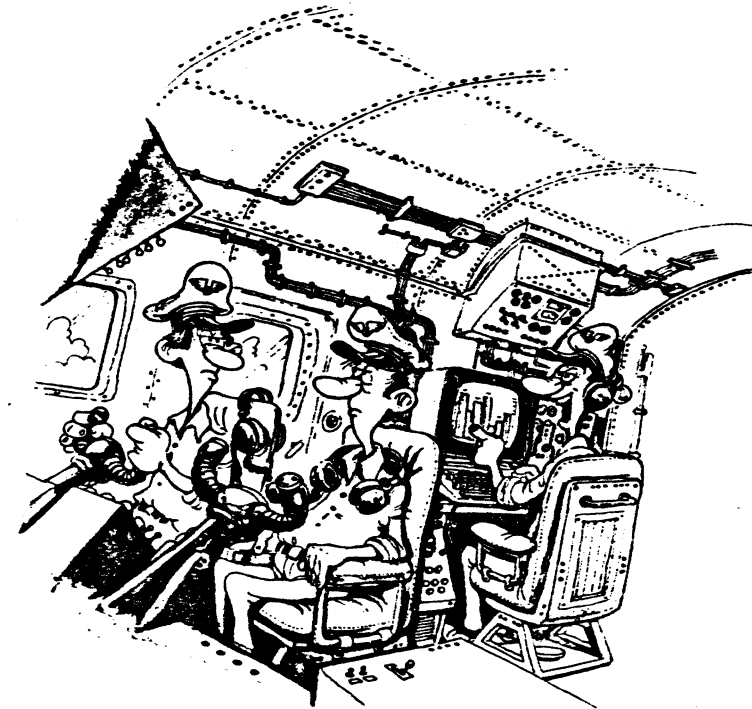
```

00106
0020 00107 fc1 DS 20h ;32 Bytes Platz für den 1. FCB
0020 00108 fc2 DS 20h ;dto. für den 2. FCB
00109
5400 00110 buffer1 EQU $+0100h$0ff100h ;beide Sektorpuffer an einer
5500 00111 buffer2 EQU $+0200h$0ff100h ;"glatten" Adresse hinter dem Programm
00112
5200 00113 END start

00000 Fehler

buffer1 5400 buffer2 5500 complop 5263 device 528A dosexit 524B dvcrea 529C
error1 5220 error2 5222 fc1 52EA fc2 530A loadlop 5245 match 528F
numbuff 52DD onlyone 5284 open 5216 start 5200 txbuff 5284

```



*Keine Ahnung, wie unsere Position ist,
aber die Aktien werden demnächst steigen.*

Vergleich zweier Programme: Noch'n Tool

In meinem Artikel "Vergleich zweier Disk-Dateien", vermutlich auch in dieser Nummer, wird eine Möglichkeit aufgezeigt, zu prüfen, ob man bei der Rekonstruktion eines Fremdprogramms einen Bock geschossen hat. Lt. Ed Murphy hat man auf jeden Fall. Bei der Weiterarbeit an meinem Problem (MEMDISK sollte geknackt werden) erwies es sich als sinnvoll, zunächst OVL4/SYS den Schneid abzukaufen. Das ist ein Overlay, das aus dem ziemlich universellen G-DOS 2.4 das Lieblings-DOS des Genie 3s macht. OVL4/SYS ist so chaotisch programmiert, daß einem die Tränen kommen. Also ein Einschub, bevor es konstruktiv weitergeht: OVL4/SYS erst mal auf Vordermann bringen.

In diesem File gibt es 47 (in Worten: siebenundvierzig!!!) ORG-Statements. Ihre Reihenfolge entspricht ungefähr der Ordnung auf meinem Schreib-tisch. Eher schlimmer. Wer mich kennt, weiß, was das bedeutet! Also erst mal OVL4/SYS umbauen, damit ein mäßig begabter rothaariger Durchschnitts-Schleswig-Holsteiner halbwegs Durchblick bekommt. Gesagt, getan - geweiht: Einen lausigeren Taschenrechner als mein G3s nach dieser Operation habe ich selten erlebt.

Irgendwo mußte ich mich bei den zahllosen Move-Anweisungen an ZEUS vertippt haben. Der Vergleich der Disk-Programme mußte versagen, denn OVL4/SYS war völlig umgebaut. Also konnte nur ein Vergleich beider geladenen Objektfiles noch etwas bringen. Wie das? Die werkeln unten im Adreßbereich des Interpreters und des DOS. Die beiden würde ich mir durch meinen OVL4-Fehltritt ungern verhunzen lassen. Also mußten sie mit einem Versatz geladen werden, so daß das Betriebssystem nichts zu motzen hatte.

Die im Anschluß gelistete Utility bringt das. Sie nullt zunächst einen Puffer aus, von dem jedes der beiden zu vergleichenden Programme eine Hälfte belegen soll. Die Nullen sind wichtig, damit bei nicht zusammenhängenden ORG-Flicken in den Lücken gleiche Voraussetzungen herrschen. Anschließend wird für beide Programme in der gewohnten Weise ein FCB eingerichtet.

Nun muß das DOS zu wissen kriegen, daß es möglicherweise Harakiri begeht, wenn es beim Laden der Programme die im Record-Header vorgesehene Adresse glaubt. Sie muß um den Offset erhöht werden. Das zweite der beiden Programme muß doppelt so hoch gepuffert werden, denn es ist erforderlich, daß beide gleichzeitig im Speicher stehen. Ein Patch in LOADP (so nennt Hartmut Grosser in seinem DOS-Buch die Routine, die Disk-Programme im CMD-Format in den Speicher schlürft) sorgt für den Versatz. Der Rest ist Handwerk: Beide Puffer werden miteinander verglichen. Bei Nichtübereinstimmung eines Bytes werden die (tatsächliche, ohne Offset) Ladeadresse und dann das geheimnisvolle Byte angezeigt.

Gemessen am früher vorgestellten Vergleichsprogramm ist dieses hier sehr spartanisch. Der einzige Komfort sind die vier Labels am Beginn. Die Ladeadresse des Tools, der Offset zur Arbeitsadresse der zu untersuchenden Programme, ihre Pufferadresse und die Einsprungsadresse des Ausgabekanals (Bildschirm oder Drucker) können damit von Fall zu Fall den Erfordernissen angepaßt werden. Die beiden Namen der (hoffentlich - das walte Murphy!) eineligen Programmzwillinge müssen ebenfalls jeweils geändert werden.

Da es sich um eine Utility für die Hand des Assembler-Cracks handelt, ist diese kleine Mühe weit weniger lästig als ein Programm für mich gewesen wäre, das jeder Schimpanse hätte bedienen können. Der Grund ist ebenso erfreulich wie simpel: Jedwedes Programm, wofern es nicht mehr als höchstens die Hälfte des ab 5200h verfügbaren Speichers belegt, kann damit überprüft werden.

Wo im Einzelfall resid liegt (die Arbeitsadresse des Tools), wo der Pufferanfang namens buffer steht, wie hoch der Offset ist (Label offset), das muß immer wieder neu ausgetüftelt werden. Aber die Kunde vom Addieren

und Subtrahieren soll sich einem Ondit zufolge in Clubkreisen schon herum-gesprochen haben. Keine Hürde also.

Das Programm von neulich kann einfach abgefahren werden wie SUPERZAP/CMD. Dieses hier ist weniger gefügig, dafür aber in seinen Angaben aussagekräftiger. Man hat die Wahl.

Arnulf Sopp

```

5200      00001 buffer EQU      5200h      ;Anfang des Programmuffers
5200      00002 offset EQU     5200h      ;Versatz zur tatsächlichen Ladeadresse
3000      00003 resid EQU      3000h      ;oder wo auch inner Platz sein mag
446A      00004 display EQU    446Ah      ;Anzeige durch den Drucker
          00005                ;(für den Bildschirm: 4467)
          00006
3000      00007 ORG      resid
          00008
3000 210052 00009 start LD      HL,buffer ;Beginn des benutzten Puffers
3003 E5      00010 PUSH     HL           ;brauchen wir noch
3004 110152 00011 LD      DE,buffer+1 ;nächste Stelle
3007 01FFAD 00012 LD      BC,-buffer-1 ;Länge des Puffers
300A -75    00013 LD      (HL),L ;die 1. Stelle ausnullen (L = 00)
300B E080   00014 LDIR                ;den ganzen Puffer alle machen
300D 117830 00015 LD      DE,fcbl ;FCB für das 1. der zu vergleichenden Pr.
3010 D5     00016 PUSH     DE           ;brauchen wir auch noch
3011 210042 00017 LD      HL,4200h ;Sektorpuffer des DOS wird benutzt
3014 45     00018 LD      B,L ;LRL = 256 (L = 00)
3015 CD2444 00019 CALL    4424h ;1. FCB eröffnen
3016 119F30 00020 LD      DE,fcbl ;FCB des 2. der zu vergleichenden Progr.
3018 D5     00021 PUSH     DE           ;wird auch noch gebraucht
301C CD2444 00022 CALL    4424h ;den 2. FCB eröffnen
301F 216930 00023 LD      HL,devia ;Umleitung der LOADP-Routine des DOS
3022 224E4C 00024 LD      (4c4eh),HL ;Umleitung an geeigneter Stelle patchen
3025 E1     00025 POP      HL           ;Adresse des 2. FCB
3026 CD2E4C 00026 CALL    4c28h ;das 2. Programm adressenversetzt laden
3029 3EA4   00027 LD      A,offset/36*2 ;offset für das 1. Programm (MSB genüt)
302B 326D30 00028 LD      (base),A ;in die Umleitung patchen
302E E1     00029 POP      HL           ;Adresse des 1. FCB
302F CD284C 00030 CALL    4c28h ;das 1. Programm adressenversetzt laden
3032 21654C 00031 LD      HL,4c65h ;ursprüngliche CALL-Adresse in LOADP
3035 224E4C 00032 LD      (4c4eh),HL ;restaurieren
3038 E1     00033 POP      HL           ;Anfang des Puffers
3039 110CA4 00034 LD      DE,buffer*2 ;Anfang der oberen Hälfte des Puffers
303C D10052 00035 LD      BC,buffer*2-offset ;Länge der Prüfstrecke = 1 Hälfte
303F 1A     00036 loop LD      A,(DE) ;ein Byte des 1. Programms laden
3040 BE     00037 CP      (HL) ;mit dem passenden des 2. Pr. vergleichen
3041 C44C30 00038 CALL    WZ,print ;fehler ausdrucken, falls ungleich
3044 23     00039 INC      HL           ;nächste Stelle unten
3045 13     00040 INC      DE           ;und oben im Puffer
3046 0B     00041 DEC      BC           ;Zähler erniedrigen
3047 78     00042 LD      A,B ;prüfen, ob er schon
3048 B1     00043 OR      C ;auf 0000 gesunken ist
3049 CE     00044 RET      Z ;zurück ins DOS, falls ja
304A 18F3   00045 JR      loop ;sonst weiter vergleichen
          00046
304C E5     00047 print PUSH   HL           ;retten, was verändert wird
304D 05     00048 PUSH   DE           ;
304E 116E52 00049 LD      DE,offset ;Offset zur tatsächlichen Ladeadresse
3051 B7     00050 OR      A ;Cy löschen wegen SBC
3052 ED32   00051 SBC      HL,DE ;tatsächliche Ladeadresse des Bytes
3054 EB     00052 EX      DE,HL ;nach DE

```

HEFT
22
Dezember
1987

```

3055 21eF30 00053 LD      NL.numbuff      ;Puffer für numerische Werte in ASCII
305E E5      00054 PUSH     HL              ;00h und trocken aufbewahren
3059 FS      00055 PUSH     AP              ;Akkumulator auch verbündet
305A CD634D 00056 CALL    4063h           ;die Ladeadresse in den Puffer schreiben
305D 23      00057 INC      HL              ;das Trennzeichen überspringen
305E F1      00058 PGP      AF              ;das nicht Übereinstimmende Byte
305F CD684D 00059 CALL    4068h           ;auch in den Puffer
3062 E1      00060 POP      HL              ;Adresse des Pufferanfangs
3063 CD6A44 00061 CALL    display         ;den Pufferinhalt ausdrucken
3066 D1      00062 POP      DE              ;Register restaurieren
3067 E1      00063 POP      HL              ;
3068 C9      00064 RET                          ;gelaufen
          00065
3069 CD654C 00066 devin CALL    4c65h           ;das 4. Byte des Record-Headers einlesen
          00067 ;(= MSB der Ladeadresse)
306C C652    00068 ADD     A,offset/256    ;um den Offset erhöhen (MSB lang)
306D          00069 base EQU     $-1              ;(der kann auch anders lauten)
306E C9      00070 RET                          ;
          00071
306F 78      00072 numbuff DN   'xxxx/yy',09h,03h ;Puffer für die Anzeige
          00073
3075 4F      00074 fcb1  DN   'OVL4/CMD',0dh ;FCB für das Originalprogramm
001E          00075 DS     3D              ;reichlich Platz dafür
309F 4F      00076 fcb2  DN   'OVL4/SYS',0dh ;FCB für seinen Zwilling
          00077
3E00          00078 END     start          ;dort Entry

```

00000 Fehler

```

base 306D  buffer 5200  devin 3069  display 446A  fcb1 3078  fcb2 309F
loop 303F  numbuff 306F  offset 5200  print 304C  resid 3000  start 3000

```



CLUB 80 HARD

Die Maus am TRS 80!

Kaum zu glauben aber wahr sind die Preise, die man heutzutage für eines dieser kleinen Rolltierchen auf den Tisch legen muß. Als ich mir vor knapp einem Jahr ein Mäuslein zulegte, habe ich 80,— Märker gelöhnt und das, obwohl mir mein Finanzminister nur eine Gebrauchtaurus zugestand! Heute bekommt man die kleinen Grautiere neu schon für 69,— DM (gesehen in der CHIP 10/87) und so wird die Überlegung interessant, ob und wofür man eine Maus braucht und wie man sie an den TRS 80 anschließen kann.

Die erste Frage kann ich nur für mich persönlich beantworten, die Frage nach dem Anschluß glaube ich aber für alle Rechner der Tandyfamilie (und noch einige andere mehr) allgemeingültig gelöst zu haben. Ich benutze die Maus ausschließlich in Grafikprogrammen und dort ist sie sehr nützlich. Die Cursorpositionierung, die gerade beim 4p sehr problematisch ist, da die entsprechenden Tasten in der rechten unteren Ecke des Hackbretts kreuzartig zusammengefasst sind, wird erheblich vereinfacht und genauer. Die verschiedenen Zeichenprogramme sind mit Maus einfach leichter zu bedienen!

Was nun den Anschluß betrifft, müssen wir zunächst ein paar Sachen klären, da die Sache ansonsten mit Sicherheit in die Hose geht, denn "Maus ist nicht gleich Maus"! Wenn man sich einmal die Anzeigen in verschiedenen Zeitschriften genau ansieht, wird man feststellen, daß es mehrere verschiedene Mausarten gibt. Als Unterscheidungsmerkmal dient im allgemeinen die Art des Anschlusses an den Computer. Hier gibt es die sog. IBM-, RS 232- und last but not least die ATARI/Commodore-Mäuse. Obwohl mir alles, was den Vornamen Commodore trägt (ob es nun Computer, Floppys oder sonstwas sind) auf Anhieb erst einmal unsympatisch ist, sind es die Mäuse dieser Firma die, die sich am leichtesten zur Zusammenarbeit mit den Tandy's bewegen lassen.

Selbstverständlich wird jetzt so mancher Besitzer einer RS 232-Schnittstelle sagen, daß diese Behauptung nicht stimmt. An seine serielle Schnittstelle kann er jederzeit eine RS 232-Maus anschließen und benötigt keinerlei zusätzliches Interface! Doch damit hat er nur teilweise recht. Zwar ist der hardwaremäßige Anschluß einer solchen Maus tatsächlich einfacher, die Einbindung in die Programme ist dagegen mit erheblichen Schwierigkeiten verbunden. Entweder müßte man für jedes Programm einen eigenen Maustreiber konstruieren oder einen solchen in das verwendete Betriebssystem einbauen.

Die von mir entwickelte Schaltung enthebt den Anwender von der zusätzlichen Softwareeinbindung indem er die Maus praktisch parallel zur Tastatur schaltet. Sie ähnelt damit dem schon mehrfach gezeigten Anschluß eines Joystick an die TRS 80-Tastatur. Der Vergleich mit einem Joystick liegt schon deswegen nahe, da der Anschluß an den Computer über den gleichen Stecker erfolgt, und sich die Maus im Prinzip auch so verhält wie ein solcher Freudenstängel.

Drückt man den Hebel eines Joysticks in eine Richtung, wird ein Schalter geschlossen, der die Masseleitung mit dem entsprechenden Ausgang verbindet. Eine Maus gibt für jede Umdrehung in eine Richtung, an entsprechenden Ausgang eine Anzahl negativer Impulse ab. Diese Impulse kann man dazu nutzen, einen elektronischen Schalter die Pfeiltasten "drücken" zu lassen.



Und schicke uns Dein Manna vom Himmel, o Herr!

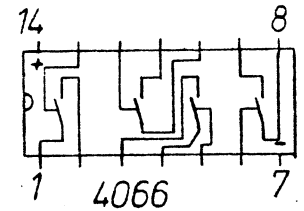
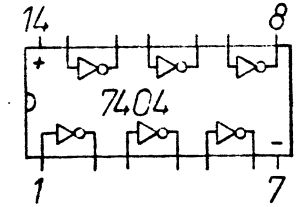
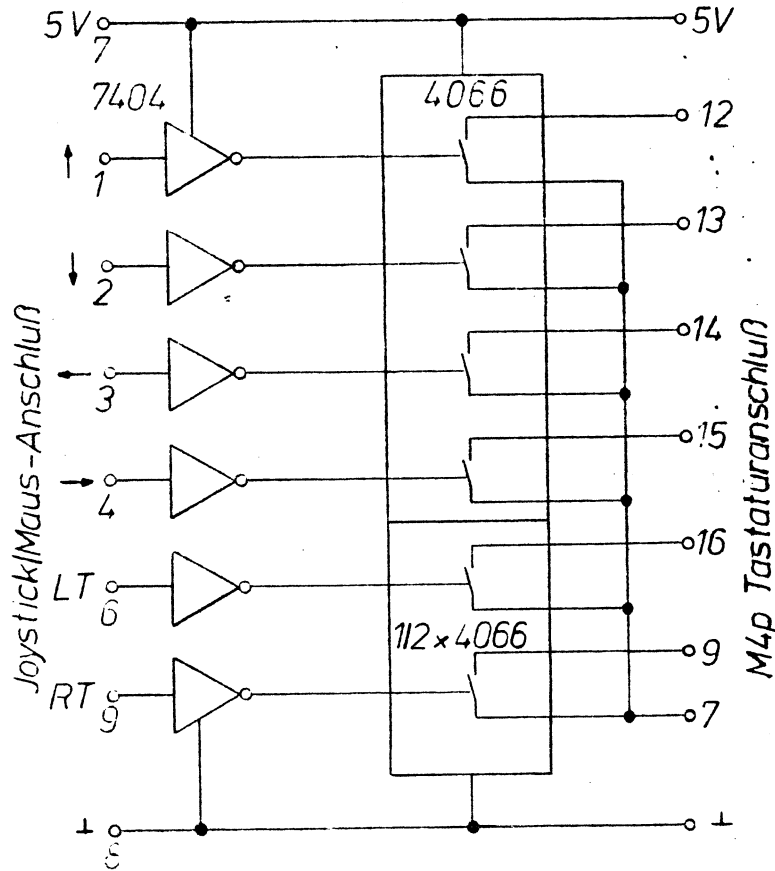
Der erwähnte elektronische Schalter ist in meiner Schaltung ein 4066. Dieses CMOS-IC enthält vier Analogschalter, die jeweils einzeln über einen Eingang betätigt werden können. Die prinzipielle Innenschaltung des IC's könnt ihr auf nebenstehender Zeichnung sehen. Einzig störender Umstand ist der, daß die Schalter mit positiver Spannung geschaltet werden müssen, während unser Mäuschen, wie oben schon erwähnt, negative Impulse abgibt. Aus diesem Grund ist noch ein Inverterbaustein 7404 und nicht etwa ein CMOS-Inverter 4049 eingesetzt wurde, hat folgenden Grund. Während der Testphase, in der ich unter anderem mit einem solchen CMOS-IC arbeitete, traten immer wieder zunächst unerklärliche Cursorbewegungen auf, obwohl die Maus nicht bewegt wurde. Einige Messungen mit dem Oszilloskop zeigten sehr schnell, daß die CMOS-Inverter bei unbewegter Maus praktisch mit offenen Eingängen arbeiteten und dadurch keinen definierten Ausgangszustand annehmen konnten, was zu den erwähnten Cursorbewegungen führte. Der Einsatz der TTL-Inverter behob dieses Problem sehr schnell!

Wie man der Schaltskizze entnehmen kann, braucht man zum Anschluß der Maus also nur drei IC's (1 7404 und 1 1/2 4066), sowie eine Joystickbuchse. Der tastaturseitige Anschluß ist auf der Zeichnung für das Model 4p angegeben, kann aber ohne Probleme auf jede TRS 80- und im Prinzip auch auf jede andere Matrixtastatur umgesetzt werden.

Selbstverständlich kann an dem Mausanschluß auch jeder Joystick, auch die mit Dauerfeuereinrichtung, angeschlossen werden, worüber sich wohl vor allem die Spielfreaks freuen werden!

Damit genug gelötet! Viel Spaß beim (jetzt hätte ich doch beinahe "mausen" geschrieben) Basteln und Computern,

Karsten Obermann



61 *Die Begründung für die ständige Verschlechterung des Wetters und die Verschiebung von Jahreszeiten und Klimazonen sehen Computerfachleute darin, daß Freund Petrus seinen Laden auf Computer umgestellt hat. Leider hat er dabei einen Fehler gemacht und sich einen Commodore angeschafft!

Vor dem Untergang der offenen und entwicklungs-fähigen Computertechnik durch Verdrängung durch den "Industrie-Standard" wurde von einer Aachener Firma für TRS-80 und GENIE eine 80-Zeichen-Karte angeboten. Wenn diese Karte auch in den meisten Fällen nur Probleme bereitete, war sie doch eine Option, die durch entsprechende Verbesserung durchaus eine Erweiterung für unsere Computer hätte werden können.

Diese Karte entsprach der TRS-80-Philosophie, daß der Bildwiederhol-speicher in den Adreßraum der CPU eingeblendet ist und die CPU die Bildschirmverwaltung selbst erledigen muß, wie das bei unseren Computern für den 64x16-Bildschirm ohnehin gemacht wird und vom Level II-ROM und vom DOS auch unterstützt wird. Die 80-Zeichen-Karte wird allerdings nicht unterstützt. Nur bei neueren TRS-80- und GENIE-Rechnern ist bei der dort schon eingebauten 80-Zeichen-Option ein manchmal funktionierender Treiber im DOS enthalten. Aber all die Programme für den TRS-80, die vom 80x24-Bildschirm nichts wissen, und die Videoausgabe nicht über das DOS abwickeln, merken nichts davon, so daß bei Umschalten auf 80 Zeichen dann nur Zeichensalat entsteht.

Glücklicherweise gibt es immer Leute, die aus Begeisterung für ein bestimmtes Programm in mühevoller Arbeit dort die Unterstützung solcher Hardware-Erweiterungen einbauen. Dadurch wird ein 80x24-Zeichen-Bildschirm aber immer noch nicht TRS-80-Kompatibel.

Es sollte noch erwähnt werden, daß im GENIE III und IIIs ausschließlich mit dem CRT6845 der Bildschirm verwaltet wird. Durch Programmierung des 6845 kann dieser im 64x16- und im 80x24-Format arbeiten. Im GIIIs ist außerdem der Zeichengenerator ein RAM-Speicher, in den beliebige Zeichensätze geladen werden können, wobei dann aber immer nur ein Zeichensatz zur Zeit benutzt werden kann. Außerdem verfügt der GIIIs noch über eine flexibel HRG, die mit dem Textbildschirm XOR-verknüpft angezeigt werden kann. Trotzdem bleibt es aber Aufgabe der CPU, im Textmodus die Bildschirmadressen zu berechnen und im HRG-Modus beim Zeichnen einer Linie alle Punkte der Linie zu errechnen.

Der eigentliche Sinn eines 80x24-Zeichen-Bildschirms ist auch nicht die Anpassung des NEWDOS/80 daran sondern die Nutzung unter CP/M. Für CP/M muß ohnehin das BIOS der Hardware angepaßt werden und da kommt es dann nicht mehr darauf an, ob ein 80x24-Bildschirm ähnlich unserem 64x16-Bildschirm bedient wird oder ob da völlig andere Verhältnisse vorliegen.

Wenn also eine noch so sehr an die TRS-80-Verhältnisse angenäherte 80-Zeichen-Karte dann doch nicht Kompatibel ist, dann kann es auch eine beliebig angeschlossene Terminal-Karte sein. Solche Karten bedeuten zudem noch einen geringeren Programmieraufwand, weil das Anwenderprogramm bei einfacher Zeichenausgabe ohne absolute Cursor-Adressierung sich keine Gedanken um Bildschirmposition, Zeilen- und Seitenende und Scrolling machen muß. Diese Leistungen erbringt selbst das dümmste Terminal selbst.

Grundsätzlich ist ein Terminal ein Interface zwischen Computer und Anwender und umfaßt daher zur Ausgabe an den Anwender die Bildschirmausgabe (und manchmal zusätzlich auch die Druckerausgabe) und die Tastatureingabe. Die Eingabe erfolgt hauptsächlich über ASCII-Tastaturen, die mit einem Strobe-Signal dem Terminal ein ASCII-Zeichen übergeben. Das wird zum Teil parallel oder auch seriell gelöst.

Das Terminal gibt das Zeichen unbesehen an den Computer weiter und dieser echot es nach Übernahme an das Terminal zurück. Erst dann wird das Zeichen vom Terminal auf den Bildschirm ausgegeben.

In einigen Fällen werden dabei Ausnahmen gemacht. So verstehen Terminals meistens ein bestimmtes Steuerzeichen als Befehl 'Local' und ein anderes (oder das gleiche ein zweites mal) als Befehl 'On Line'. Damit kann man umschalten zwischen Ausgabe nur der Zeichen vom Computer und sofortige Ausgabe der Zeichen von der Tastatur, ohne diese an den Computer weiterzureichen.

Manche Terminals können auch mit einem Steuerzeichen in einen Setup-Modus versetzt werden, von wo aus gewisse Parameter des Terminals über die Tastatur geändert werden können. Dazu kann gehören: die Umschaltung des Bildschirmformates, die Wahl nationaler Zeichensätze, die Änderung der Baudrate für die Kommunikation mit dem Computer u.s.w.

Die Verbindung mit dem Computer kann verschiedene Formen annehmen. Die verbreitetste Variante ist ein über eine serielle Schnittstelle angeschlossenes Terminal. Damit ist es möglich, daß das Terminal mit Bildschirm und Tastatur ein eigenständiges Gerät ist und der Computer weiter entfernt stehen kann. Da bei den heutigen Tischgeräten dieser Gesichtspunkt nicht mehr ausschlaggebend ist, werden auch schon viele 'Video-Karten' für diverse Bussysteme als Steckkarten im Computer über den Systembus angeschlossen. Einige Varianten vereinigen beide Gesichtspunkte und verfügen über z.B. einen ECB-Bus-Stecker, der nur der Zuführung der Versorgungsspannungen dient, und eine RS232C-Verbindung mit dem Computer. Prinzipiell ist der Anschluß über den Systembus wegen des parallelen Datentransfers schneller. Bei Baudraten von 9600 oder 19200 ist aber auch der serielle Anschluß schnell genug. Auf alle Fälle kann er mit der Ausgabe eines memory mapped Bildschirms, der unter CP/M für die Ausgabe jedes Zeichens ständig ein- und ausgeblendet werden muß, mithalten.

Das hat alles Vor- und Nachteile: Ein Terminal verlangt eine sehr viel leistungsfähigere Tastatur, die schon selbst die Umwandlung von ausgelesenen gedrückten Tasten in ASCII-Codes leisten muß. Bei unseren Rechnern muß das DOS bzw. der Z80 diese Umwandlung erledigen, was entsprechend umfangreiche Eingaberoutinen und auch CPU-Rechenzeit kostet. Da ein Programm aber bei Eingaben ohnehin nur auf die Tastatur wartet, ist das ohne Belang. Andererseits ist es bei einer ASCII-Tastatur auch nicht möglich, Mehrstasten-Befehle einzubauen (123, JKL, DFG). Ähnliches kann aber mit programmierbaren Funktionstasten oder einer Interrupt-getriebenen Terminaleingabe auch realisiert werden. Das wird aber in den meisten Fällen aus Angst vor Interrupt-Programmierung unterlassen.

Programmierbare Funktionstasten von ASCII-Tastaturen werden meistens durch Ausgabe von Codes > 7FH beim Drücken einer 'F'-Taste realisiert. Die Eingaberoutine des Betriebssystemes lügt sich dann bei einem solchen eingegebenen Zeichen einen diesem Zeichen zugeordneten String als eingegeben vor.

Als reines Textausgabegerät kann also eine Terminal-Karte in TRS-80-Kompatiblen Computern insbesondere unter CP/M als leistungsfähiger Ersatz für eine von der CPU selbst verwaltete, memory mapped 80-Zeichen-Karte gute Dienste leisten. Der Anschluß einer ASCII-Tastatur ist möglich aber nicht zwingend. Gegenüber der Minimal-Tastatur des TRS-80 und des GENIE ist eine ASCII-Tastatur aber ein zusätzlicher Komfort, der die übliche Fingerakrobatik bei der Eingabe von Umlauten endlich durch dafür vorgesehene Tasten an der richtigen Stelle ersetzt.

Bei Terminal-Karten existiert ein eigener Prozessor, der den Bildwiederhol-speicher verwaltet und die von der CPU übergebenen Zeichen entgegennimmt. Außer den darstellbaren Zeichen (20H-7FH) versteht er die meisten Steuerzeichen (00H-1FH) und setzt diese in entsprechendes Wohlverhalten um. Meistens reichen diese Befehle aber nicht für alle Wünsche der Bildschirmverwaltung aus, so daß

außerdem noch ESC-Sequenzen weitere Steuerfunktionen bereitstellen. In der Festlegung solcher ESC-Sequenzen und in deren Menge unterscheiden sich die verschiedenen Terminals voneinander. Die Bedienung eines Terminals ähnelt also mehr der eines Druckers als der des herkömmlichen TRS-80-Bildschirms. Und genauso wie bei den Druckern gibt es mehr oder weniger leistungsfähige Terminals.

So gibt es analog den Typenrad-Druckern auch reine Text-Terminals, die nur ASCII-Zeichen (natürlich auch deutsche Umlaute) darstellen können, und andere, die wie ein Matrixdrucker auch grafikfähig sind. Eine solche grafikfähige Terminalkarte, die GRIP, hat uns Manfred Held im letzten Info schon vorgestellt und auch gezeigt, daß sowas bei uns durchaus gute Dienste leisten kann. Die neueste Version der GRIP (Vers.5.53) ist in der Lage, HRG mit 768x567 Punkten darzustellen und Text in bis zu 96x35 Zeichen auszugeben. Durch den HD6345-CRTC, der aufwärtskompatibel zum MC6845 ist, wurde dort auch ein augenschonendes Smooth-Scrolling (Hochschieben des Bildschirminhaltes nicht textzeilenweise sondern dotzeilenweise) erreicht. Gegenüber der etwas lahmen Urform GRIP1 ist die GRIP5.53 eine der schnellsten Terminal-Karten für CP/M-Computer überhaupt.

Eine ähnlich leistungsfähige Karte ist die Video 1 von Janich & Claas, die ebenfalls HRG und Textdarstellung beherrscht.

Weniger empfehlenswert ist die TERM 1 des mc-CP/M-Computers, die zwar wegen des EF9366-Video-Controllers in der Graphik einige interessante Aspekte bietet, die aber wegen der äußerst langsamen Textdarstellung überhaupt nicht zu gebrauchen ist.

Die GDP64 des NDR-Klein-Computers ist eine stark abgemagerte TERM 1 ohne eigenen Terminal-Processor. Hier muß die CPU selbst den EF9366 programmieren. Das bringt zwar Geschwindigkeitsvorteile, macht die normale Textausgabe aber recht umständlich. Diese Karte ist auch im eigentlichen Sinn kein Terminal, ihre Bedeutung ist auch einzig in der schnellen Vektorgraphik des EF9366 zu sehen.

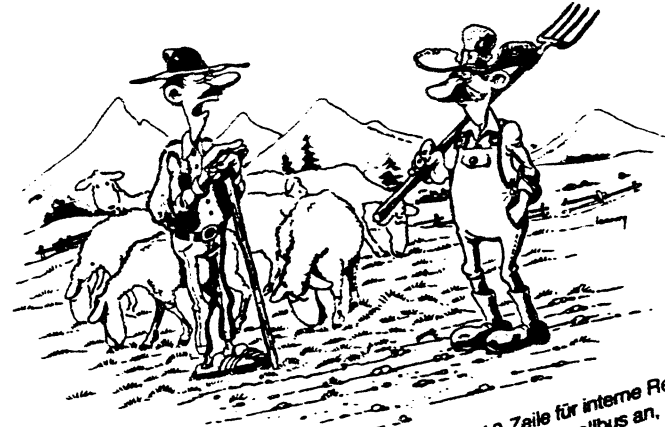
Beiden Karten mit dem EF9366 ist gemeinsam, daß sie die vereinnahmten HRG-Inhalte nicht wieder hergeben (es sei denn, man rüstet diese Eigenschaft in Hardware nach).

In den meisten Fällen genügt es aber, wenn man für die Implementation von CP/M eine reine 80x24-Text-Terminal-Karte benutzt. CP/M fordert zwar keine 80x24-Zeichen-Darstellung und flexiblere Programme lassen sich auch für 64x16-Bildschirme installieren; die meisten unter CP/M laufenden Programme gehen aber von dieser Bildschirmgröße aus, so daß man beim 64x16-Zeichen-Format einen großen Teil der Informationen nur vorbeihuschen sieht und nicht lesen kann. Besonders ärgerlich ist dies bei den in vielen Utilities aus dem Programm heraus aufrufbaren HELP-Texten.

Eine sehr elegante Lösung besteht in der Verwendung des c't-Text-Terminals. Neben einer mit maximal 19200 Baud Übertragungsrates sehr schnellen Textausgabe überzeugt diese Karte besonders durch ihr günstiges Preis/Leistungs-Verhältnis. Die Karte läßt sich für ca. 300 DM aufbauen und bietet dabei eine Fülle von Optionen. So lassen sich Zeichen mit bis zu 8 gleichzeitig wirksamen Attributen darstellen und es können gleichzeitig verschiedene Zeichen oder Zeichengruppen auf dem Bildschirm mit verschiedenen Attributen dargestellt werden. Die Attribute umfassen: Normaldarstellung, doppelt breit, doppelt hoch, halb hell, invers, blinkend und unsichtbar. Die Attribute werden durch entsprechende ESC-Sequenzen gesetzt und gelten dann für alle folgenden ausgegebenen Zeichen, bis andere Attribut-Befehle ausgegeben werden.

Literatur

Low-Cost: c't Text-Terminal; Tilman Reh; c't 9/86, 65
Das mc-Terminal; R.Jäger, H.-J. Regge, G. Tobergte, mc 9/83, 70
Das mc-Grafik-Terminal; R.-D. Klein; mc 8/83, 68; mc 9/83, 70
Grafik-Interface-Processor GRIP-1; J.C. Lotter, c't 6/84, 52; c't 7/84, 92



... dann benutze die AO-Zeile für interne Registerwahl, schließ die anderen drei Signale an den Kontrollbus an, und vergiß nicht, den Mist von Deinen Stiefeln zu kratzen.

Schwimmende Bilder

Wer denkt, hier gibt's wieder was grafisches, hat leider Pech gehabt. Es geht vielmehr um einen kleinen, aber sehr störenden Fehler, der in der Hardware auftauchen kann.

Als ich für einen Freund mein altes Benie II wieder aus dem Schrank geholt habe und es auf Vordermann bringen wollte habe ich eine böse Überraschung erlebt: Booten ging noch, aber das Monitorbild war absolut unmöglich. Die Ränder wirkten, als ob Wasser die Mattscheibe herunterlaufen würde. Da bei mir kein Schlangenbeschwörer wohnt ging die große Sucherei los. Als Ursache stellte sich schließlich eine kaputte Diode im Netzteil heraus, die aus irgend einem Grund in Durchlaßrichtung hochohmig geworden war. Dadurch ist die Spannung für die TTLs von 5V auf ca. 4.3V gesunken. Warum das Ding überhaupt noch lief ist mir ein Rätsel. Nach dem Austausch war jedenfalls alles wieder in Ordnung und das Bild ruhig und stabil. Wenn Euer Kasten auch den Freischwimmer machen will, meßt erst mal die Spannungen nach. So schnell geht ein Rechner schließlich nicht kaputt. Meistens sind es nur solche lächerlichen Kleinigkeiten, die einem das Leben versauern.

Alexander Schöid

BÜRSE -- BÜRSE -- BÜRSE

Es war mein schönstes!

GROSSER MARKTTAG !!!

VERSCHENKE (tatsächlich!)

EIN TOSHIBA-LAUFWERK

Es sei nicht verschwiegen, daß es reparaturbedürftig ist (infolge menschlichen Versagens versehentlich mit 220 V „bestroat“ (aber nur ganz kurz, die Lichtsicherung flog sofort raus!). Für künftigen Hardwarefreak sicher eine Kleinigkeit. Abholung oder (als Bestellung) für Porto DM 3,— auf Postgirokonto München, Nr. 3847 37-804, Frau Ulrike Schneider; Kennwort: LAUFWERK.

Des weiteren habe ich mich entschlossen, mich von einem meiner beiden GENIE III zu trennen, wenn sich ein Interessent bei fairem Wertausgleich (ich dachte so an etwa 1270,— DM) findet.

Für VG I/II - TRS 80 Model I-Besitzer wären alle Probleme einer 80-Zeichenkarte, freies RAM für CP/M, Kabelsalat usw. komfortabel gelöst! Außerdem mit (selbstverständlich mitgeliefertem) NEWDOS/6DOS-System volle Kompatibilität zu diesen Maschinen (aber etwa doppelt so schnell!).

Bildschirm umschaltbar 64x16 oder 80x25 Z./Z. Freie Tastatur, deutsch mit Umlauten, Ziffernblock, 8 Funktionstasten frei belegbar (bis 16 mit SHIFT), zwei eingebaute 80-Track-Laufwerke DS (1,4 MByte formatiert), zwei weitere LW extern anschließbar. Aufrüstung bis 1 MByte RAM möglich (vgl. Clubinfo 14, Juli 86, S. 91 - 100, an meinem anderen GENIE III erfolgreich für lässige 199,- bis 249,- DM + Clubfreundhilfe praktiziert: RAM-Disk, SYS-RAM, vier nahezu unabhängige Computer mit verschiedenen Programmen gleichzeitig auf Knopfdruck usw., und nochmal 2 - 10 mal schneller....!)

Bei Interesse Unmassen von Software jeder Art (original), „Geniothek“ mit GENIETEXT + Kalkulation, GenieCalc, Lohn + Gehalt, rel. Datenbank DIS (80x25 Z.-Bildschirm). Nachkalkulation (Orgelbau, aber leicht für jede Branche zu ändern, da komfortabel-professionelle BASIC-Programmierung meines Sohnes), Buchführungen (freiberuflich und andere), Zeichen-generator HEADLINE, ASEM (blerkurs) 4 und OPAL 68000 Crossassembler (alles von Milke/IDA), SuperCross (ca. 180 Formate einschl. MS-DOS lesen) mit Basic-Programmkonvertierung etc. usw. und so fort (2 x die Hälfte vergessen!). Dazu CP/M mit Shell, mit Nordstar, Turbopascal u.a. Außerdem ca. 1/2 t einschlägige Literatur.

Darüber süßte gesondert verhandelt werden.

Alles befindet sich in gutem Zustand. Die genannten Preise gelten bei Abholung, andernfalls + Porto.

Richard Renact, Bahnhofstraße 100, 7128 LAUFFEN am Neckar

Unser ehemaliges Mitglied Uwe Gromottka hat sich, nachdem er sich auch beruflich mit MS-DOS beschäftigen muß, nun auch für private Zwecke einen solchen Rechner angeschafft. Dadurch wird bei ihm natürlich eine ganze Menge Hardware frei, die er nun veräußern möchte. Es steht demnach zum Verkauf:

- Video Genie II mit eingebautem Disk-Controller 1771/1791
 - CMC Monitor grün 12 Zoll
 - 1 Diskettenlaufwerk 40/DS/DD
 - Schmidtke 80 Zeichen-Karte mit Banking für CP/M.
 - Auch Paperware gibt es eine ganze Menge:
 - Trommelschläger's Service Manual
 - TRS-80 Disk und andere Geheimnisse in deutsch
 - Betriebssystem CP/M (Plate / Franzis-Verlag)
 - Programmierung des Z80 (Zaks / Sybex)
 - Kopieren von Programmanleitungen und anderen Schriften.
- Die Preise sind Verhandlungssache und direkt zu erfragen
bei: Uwe Gromottka
Lange Reihe 40
2803 Weyhe
☎ 0421 / 88496

*läßt man bei P. Spiels
brennen! Auch C-MOS*

DDR und Computer

Seit kurzem habe ich Kontakt mit einem jungen DDR-Bürger, der zur Zeit eine Lehre im Elektronikbereich durchmacht. Dieser junge Mann wünscht sich nichts sehnlicher (ist vielleicht etwas in der "anderen Republik" zum Privatgebrauch nicht gibt, bat er mich um Hilfe.

Ich gebe diese Bitte an euch weiter. Wenn irgendwo, bei irgendwem ein Computer (vom programmierbaren Taschenrechner bis zum TRS 80/Genie) unbenutzt in der Ecke steht und sowieso auf den Müll soll, wäre dies eine sinnvolle Verwendung für das Gerät! Einzige Bedingung, die erfüllt sein sollte: das Gerät sollte funktionsfähig sein, da Ersatzteile in der DDR nicht zu beschaffen sind. Auch eine Anleitung wäre von vorteil, da begleitende Literatur mit Sicherheit fehlt!

Wer also seine Hardware vom TI 57 bis zum MS-DOS-Computer (kostenlos) los werden möchte, kann sich an mich wenden. Ich gebe auch gerne die Adresse zur weiteren Kontaktaufnahme weiter!

Kontakt Obermann

VERKAUFE:

- TRS 80, Model 1, L 2, 16 K
- 2 TANDON-Laufwerke, 40 Tr., SS, DD
beides gegen entsprechende Gebote an
Klaus Hermann

Eintr DM 75,- von Arnulf Sopp, Tel. 0451-7191298
Unidirektionaler Traktor für den NEC 6 (im Lieferumfang des Neugeräts nicht
enthalten), Komplet mit höherem Gehäusedeckel und Papierführung.
Zustand wie neu. Neupreis Ende 1986: DM 130,-

.....HILFERUF von Ka-Jott in Not(!):
? Wer kennt das ?
TRS-80 MUSIKPROGRAMM

BÜRSE -- BÜRSE -- BÜRSE

Suche für TRS-80/M I mit 48 k RAM
das NEWBAS-Programm auf Kassette.
W. Hentz
Am Tränkgarten 20
Tel. (0 61 09) 6 66 25 ab 19 Uhr

Disk Version von W. Müller, Ratingen
und kann mir die Anleitung zum Kopieren geben ?
Seit TANDY's Hochburg in Ratingen (b. Düsseldorf) aufgelöst ist, gibt's von dort keine Hilfe mehr -
(wem sag' ich das?) [Jetzt sind wir Kundendiesel doch gelüht.]
Dieses Programm bringt die saubersten Töne hervor - aber es gelingt (mir) nicht, mehr als eine Bildschirmseite zu schreiben: THERE IS NO "SCROLLING"!

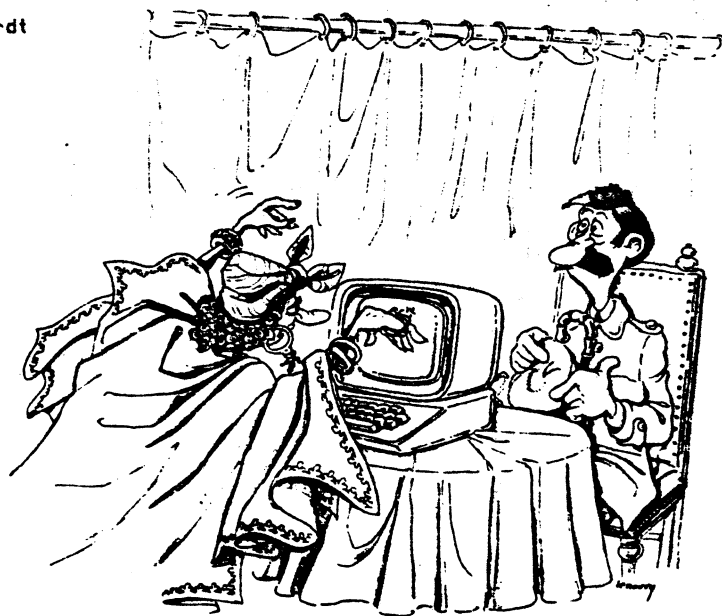
"Prüfzahlfehler beim Lesen"

Diese Horrormeldung stelle man sich vor, wenn man nach dem Überarbeiten eines 3 Files umfassenden Assemblerprogramms, das beim letzten Assemblieren 7 K Byte Objektcode ergab, einen der Source-Files laden will. Wenn sich dieses Erlebnis dann auch noch bei den anderen beiden Files einstellt, reicht das dann erst mal für den Rest des Tages. Wer etwas härter im Nehmen ist, macht dann noch einige zaghafte Versuche mit SUPERZAP, das bei etwas Glück mit seiner Vermutung, daß einige oder alle Daten des defekten Sektors im Puffer enthalten sind, Recht hat. Leider weigert es sich, diese nach eventuell nötigen Korrekturen wieder auf die Diskette zurückzuschreiben. Der Umfang des Schadens läßt sich zwar feststellen aber nicht beheben.

So ganz wollte ich mich aber nicht mit der Vorstellung anfreunden, 32 DIN A4 Seiten gut kommentierten Programms neu einzutippen. Also erst mal versuchen, ob sich der Rest der Files auf eine andere Diskette kopieren läßt. Wenn dann da einige Sektoren mit ESH gefüllt sind, würde ich da mit SUPERZAP schon was reinbringen, das ich hinterher mit ZEUS wieder in sinnvollen Text verwandeln könnte. Hauptsache, es tritt kein CRC-Fehler mehr auf. Beim Kopieren mit COPY,0,1 traten dann auch die erwarteten Meldungen auf, die ich mit einem <F>ortfahren beantwortet habe. Insgesamt 5 Sektoren hat das DOS beanstandet. Ich sollte offensichtlich 1,25 KByte von ESH in von ZEUS verwertbares Material umzapen. Na gut, an die Arbeit, mit SUPERZAP den ersten beim Kopieren beanstandeten diskrelativen Sektor geholt und - eitel Freude - es war keine einzige ESH zu sehen sondern mein Originaltext, der ohne CRC-Fehler geladen wurde.

Von den anderen 4 Sektoren war dann noch einer geringfügig zu korrigieren und nach einer halben Stunde waren meine Source-Files wieder heil. Das mag für manchen von Euch der Schmeiß von gestern sein, wenn man aber unter der Vorstellung lebt, bei solchen Katastrophen nur das Handtuch werfen zu können, dann mag diese Erfahrung recht hilfreich sein. Trotzdem hoffe ich, das möglichst wenige von Euch drauf zurückgreifen müssen.

Helmut Bernhardt



- mit jemand reden, heißt: "Echig" mit ihm reden, sozusagen "gerade heraus"!

Fractura = lat. Bruch. - Frakturschrift (Frakturdruck) = "gebrochene Schrift".

"Mit jemand Struktur reden" kann also zum Bruch führen! Darin renne ich mich besser aus als in der Assemblersprache. (Manch einer weiß das!)

Nicht nur, sie zu reden - auch sie zu lesen macht Spaß.

Denn dann ist sie viel schöner gedruckt, als meine hier, die ich mit TSCRIPS' Zeichenbau-Utility verbrochen ("frakturiert") habe.

Recht echig ist sie, im Gegensatz zur runden lat. ANTIQUA, die man fast nur noch vorgefetzt bekommt.

Die Redensart stammt übrigens aus dem 17. Jahrhundert. (Das ist das Jahrhundert, dessen Jahreszahlen mit 16.. anfangen. Nicht 17... wie neulich in einer "Sachzeitschrift" zu lesen stand...)

Hier eine Kostprobe meines Alphabetes, da bisher noch nicht alle Buchstaben und Ziffern vorkamen:

ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890
abcdefghijklmnopqrstuvwxyz

Manchem gefällt's nicht (u.a. meiner Frau.) "Geschmacksache". - Geschmacksache? Über den Geschmack "läßt sich (nicht) streiten". (Dieses unklare Sprichwort sollte besser lauten: "Über Geschmack läßt sich diskutieren".)

Diejenigen, die weder mit mir über Geschmacksfragen noch mit TSCRIPS' über Fingerfertigkeit und HRG-Gültigkeit "streiten" wollen, können gern meinen Zeichensatz auf Disk (oder Diskette) haben (formatiert zwischen 0 bis 80 Tracks, von undicht bis doppelt dicht). Aufschrift hinten.

Übrigens: Wer Angst hat vor dem bösen "j", dessen richtiger Einsatz heute weitgehend unbekannt ist, braucht sich darum keine Sorgen zu machen und kann getrost stets "s" schreiben. * Wer aber die unfassende Regel wissen möchte - nun gut, sie lautet:

Das "lange" (jog. stumme oder scharfe) "j" steht am Silbenanfang und das "s" am Silbenende!

Wie schreibt sich also: Das Beste?

Das Beste? - Mitnichten! Sondern: Das Beste! - Denn beim Trennen ist es das be=

ste, man läßt das "st" beisammen; dieses wird nie zerrissen! Das "j" steht hier also am Silbenanfang.

Aber wie so schreibt man dann das Wort "besser" beim Trennen: bej=

ser, also auch mit zwei langen "j", wo doch das j der ersten Silbe ein "Schluß-Es" ist???

Nun, ganz einfach:

Keine Regel ohne - ach so, ihr wißt schon.

Und die Ausnahmen sind es doch, die die Regeln und das Leben so interessant machen.

xlves / - xalot

Sicherer Platz für Daten?

Der größte Unsicherheitsfaktor beim EDV-Betrieb? Vermutlich der Mensch. Die Fehlerrate von Computerbedienern konnten und wollten wir nicht testen, dafür aber die Zuverlässigkeit des gebräuchlichsten Speichermediums für Personal Computer: 2 verkaufshäufige Disketten offenbarten uns ihr magnetisches Gedächtnis.

Was man schwarz auf weiß besitzt, kann man getrost nach Hause tragen. Mit dieser Meinung steht der bellensene Schüler aus Goethes »Faust« auch im Computer-Zeitalter keineswegs alleine da. So ließe sich heute des Dichters Meisterwerk, Vers für Vers in Bits übersetzt, auf einem Dutzend Disketten unterbringen, aber man hätte den »Faust« damit gleichzeitig den Unwägbarkeiten der Elektronik auf Gedächtnis und Vorrat ausgeliefert. Was zum Beispiel, wenn aufgrund eines Materialfehlers die Grotchenfrage von der Magnetschicht verschwände oder der Osterspaziergang einem auf der Diskette haftenden Schmutzpartikel zum Opfer fiel? Sicher, auch Bücher, Zeitschriften, Aktenordner sind nicht unzerstörbar oder können verlorengehen.

Aber der Gedanke, wichtige Daten einer unscheinbaren Plasticscheibe anzuvertrauen, widerstrebt zuweilen selbst aufgeklärten Zeitgenossen. Wer mit Computern arbeitet, muß auch speichern. Bei Daten für den sofortigen Gebrauch genügt der Arbeitsspeicher, der in jedem Rechner eingebaut ist. Für die dauerhafte Aufbewahrung von Informa-

tionen braucht man dagegen sogenannte Massenspeicher. In der Regel sind das magnetische Datenträger. Nicht wenige Homecomputer-Besitzer betreiben dazu Audio-Kassetten – vor allem wegen der billigen Aufnahmegeräte. Diese Speicherung ist freilich ebenso umständlich wie zeitraubend, da man mitunter die ganze Kassette umspulen muß, um an die gewünschte Stelle heranzukommen. Bei Personal Computern (PC) verwendet man deshalb nur noch Disketten oder Festplatten. Beide gewährleisten den direkten Zugriff auf alle Daten. Festplatten haben darüber hinaus unbestreitbare Vorzüge (große Kapazität bei geringer Fehlerquote und sehr kurzen Zugriffszeiten), dafür bleiben die Daten auf Disketten mobil, geeignet also für Transport oder Versand.

Disketten gibt es in drei verschiedenen Größen, die nach der Kantenlänge der Hülle benannt sind:

■ 8-Zoll- oder Maxidisketten (Kantenlänge etwa 20 Zentimeter) haben für den PC-Bereich keine Bedeutung.

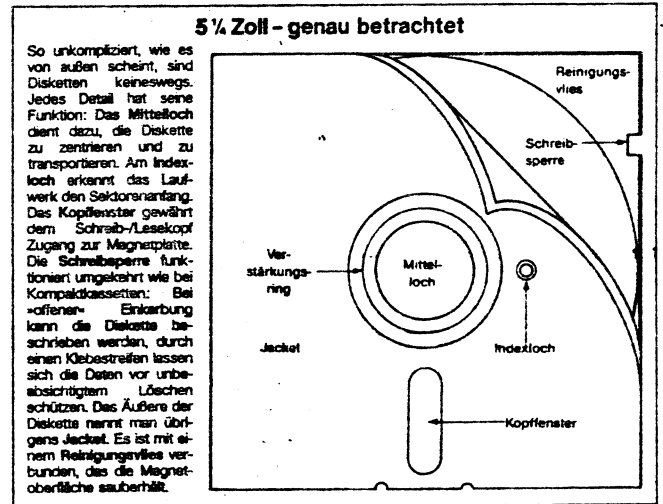
■ 5 1/2-Zoll- oder Minidisketten (Kantenlänge 13 Zentimeter) haben den Löwenanteil am Diskettenmarkt. Kein Wunder, passen sie doch in die serienmäßigen Laufwerke von IBM-kompatiblen PCs. Wir haben diese Größe für unseren Test ausgewählt.

■ 3 1/2-Zoll- oder Mikrodisketten – sie messen neun Zentimeter an den Kanten – gehört nach Meinung mancher Experten die Zukunft. Die steil ansteigende Verkaufskurve spricht dafür. Neben der geringen Größe gilt auch ihr stabiles Hartplastikgehäuse als Pluspunkt. Besitzer von Apple- und Atari-PCs profitieren schon jetzt davon.

Obwohl wir uns bei unserem Test bereits auf eine Größe festlegten, konnten wir unmöglich das gesamte Angebot in diesem Segment berücksichtigen. So verzichteten wir zum Beispiel auf die No-name-Fabrikate. Auch mußten wir uns auf Disketten mit folgenden Kenndaten beschränken: zweiseitig beschreibbar, doppelte Speicherdichte, Spurdichte: 48 tpi, softsektoriert. Wenn Sie nun daraus folgern, daß es auch hardsektorierte oder einseitige Disketten mit einfacher Speicher- und anderer Spurdichte gibt, liegen Sie genau richtig. Weder das Äußere einer Diskette noch ihre Bezeichnung läßt indes erkennen, wie die Signale auf der Magnetschicht angeordnet sind.

Beschreibbarkeit: Da es Floppy-Laufwerke mit einem oder zwei Schreib-/Leseköpfen gibt, bekommt man auch Disketten, die sich von einer Seite und solche, die sich beidseitig beschreiben lassen. Der Unterschied? Im Prinzip keiner, nur sind zweiseitig beschreibbare Disketten vom Hersteller auch für diese Anwendung geprüft, während bei den anderen die Kehrrseite durchaus fehlerhaft sein kann. Man könnte also probieren, eine Diskette, die für Einkopf-Laufwerke vorgesehen ist, beidseitig zu beschreiben – muß sich dann allerdings über einen Datenverlust infolge einer schadhafte Stelle nicht wundern. Deshalb unser dringender Rat: Verwenden Sie für Doppelkopf-Laufwerke nur die dafür vorgesehenen Fabrikate.

Die Signalaufzeichnung und -abtastung erfolgt bei Disketten spurweise, und zwar von außen nach innen in abgeschlossenen, konzentrischen Kreisen. Der Magnetkopf zieht also erst ganz außen seine Bahn, um dann nach jeweils einer Umdrehung in einen Kreis mit etwas geringerem Durchmesser zu »springen«. Je mehr Spuren der Magnetkopf auf der Diskette zieht, desto größer ist die Informationsfülle, die sich unterbringen läßt. Die Anzahl dieser Spuren wird in tracks per inch (tpi), also Spuren pro Zoll angegeben. Die gängigen Maßzahlen für die Mini-Diskette sind 48 und 96, wobei 48 tpi lediglich 40 Spuren und 96 tpi 80 Spuren auf der Diskette entspricht. Grundsätzlich könnte eine 48-tpi-Diskette auch in einem Laufwerk mit 96 tpi – wie es der IBM-AT-Computer besitzt – gelesen werden. Das Beschreiben klappt freilich nicht, denn wegen der höheren Signaldichte benötigen die IBM-



AT-Laufwerke Disketten mit einer anderen Magnetfolie, sogenannte High-Density-Fabrikate.

Freilich ist die Anzahl der Spuren nicht der einzige Anhaltspunkt für das Aufnahmevermögen der Magnetscheiben, denn auch die Speicherdichte längs der Spuren kann unterschiedlich sein. Da es aber kaum jemanden interessiert, wieviel Bit auf einer Diskettenspur unterzubringen sind, geben die Hersteller auf ihren Produkten meist nur an, ob diese einfache oder doppelte Speicherdichte haben. Der Normalfall ist heute die doppelte Dichte, abgekürzt zum Beispiel mit dem Buchstaben D. Disketten einfacher Dichte sind nur wenig gebräuchlich.

Eine Normung, wie sie sich zum Beispiel bei Audiokassetten bewährt, gibt es zwar auch für Disketten, aber durchgesetzt hat sie sich leider noch nicht. Der Käufer muß also bis auf weiteres die technischen Grunddaten mehr oder wenig mühsam aus der Artikelbezeichnung entschlüsseln. In einigen Fällen fehlen die Informationen auf der Verpackung oder Diskettenhülle ganz und gar. Disketten sind übrigens abwärtskompatibel, man kann also eine für 96 tpi geprüfte Diskette durchaus mit 48 tpi oder eine zweiseitige nur auf einer Seite bespielen.

Alle für Personal Computer gebräuchlichen Disketten sind softsektoriert. Das bedeutet: Außer dem Indexloch, das dem Schreib-/Lesekopf anzeigt, wo er mit der Arbeit beginnen soll (siehe auch Schnittzeichnung oben), haben sie keine weiteren sichtbaren Markierungen in der Magnetplatte. Die Sektoreneinteilung übernehmen magnetische Informationen bei der

Formatierung. Bei den ersten – hardsektorierten – Disketten besorgten hingegen richtige Löcher die Sektoreneinteilung – die somit auch sichtbar war.

Funktionsprüfung

Die bei unseren PC-Systemen notwendige Formatierung war auch Gegenstand einer Prüfung, die wir an jeweils 90 Exemplaren jeder Diskettenmarke durchführten. Eigentlich dürfte es dabei keine Probleme geben, aber bei drei Fabrikaten widersetzten sich einzelne Disketten beharrlich dieser Sektoreneinteilung. Wenn aber auf einer Diskette einige tausend Bytes gesperrt sind oder sie sogar mechanische Fehler aufweist, kann sie als zuverlässiger Informationsträger kaum noch gelten. Dabei trägt die Boeder-Hi-Tec-Diskette, die in der Formatierprüfung versagte, das Prädikat 100% error free. An diesem hohen Anspruch muß sie sich dann auch messen lassen.

Im Gegensatz zu Video- und Audiobändern, bei denen ja auch magnetisch aufgezeichnet und wiedergegeben wird, dürften Disketten im Grunde genommen keine Fehler aufweisen. Während sich beispielsweise ein sogenannter Drop-out, also ein Signalausfall infolge Verschmutzung oder Beschichtungsfehler, dort lediglich als Tonstörung oder Blitzen im Bild bemerkbar macht, hat er hier mitunter verheerende Folgen. Wenn nämlich einige Daten auf der Diskette fehlen, geht das Programm nach dieser Fehlerstelle nicht einfach zur Tagesordnung über, sondern



COMPUTER-ZEITUNG
Disketten (5 1/4 Zoll)

75

78



Fabrikat (Anbieter-Adressen siehe Seite 26)	BASF Flexy Disk Quellmetric 2D ¹⁾				BASF Flexy Disk Science 2D ¹⁾		Dysan Flexible Diskette 104/2D		Maxell MD 2-D		Nashua MD2D WP-R Art.-Nr. 2725-970		Quelle Privileg Art.-Nr. 555.511		RPS High Focus Floppydisk MN2DD-O-G	
Preis für 10 Disketten nach Markterhebung in DM	35,- bis 70,-				110,-		70,-		45,50,-		30,-		14,95,-		65,-	
Mittlerer Preis für 10 Disketten in DM	35,-				110,-		70,-		45,50,-		30,-		14,95,-		65,-	
Verpackung	Stülpfachteil				Stülpfachteil		Stülpfachteil		Stülpfachteil		Stülpfachteil		Stülpfachteil		Stülpfachteil	
Material der Einstecktasche	Faserfloss				Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss	
test-Qualitätsurteil	GUT				GUT		GUT		GUT		GUT		GUT		GUT	
TECHNISCHE PRÜFUNG	sehr gut		sehr gut		sehr gut		sehr gut		zufriedenstellend		zufriedenstellend		gut		gut	
Prüfung der Abmessungen	gut		+		sehr gut		++		sehr gut		++		sehr gut		++	
Dauerprüfung	sehr gut		++		sehr gut		++		sehr gut		++		sehr gut		++	
Temperaturprüfung bis 50 °C	sehr gut		++		sehr gut		++		sehr gut		++		mangelhaft*) Hülle verformt		-	
Verarbeitung der Hülle	sehr gut		++		sehr gut		++		gut		+		sehr gut		++	
Verarbeitung der Magnetplatte	sehr gut		++		sehr gut		++		sehr gut		++		zufriedenstellend*) Aufzeichnungsbereich unvollständig poliert		-	
FUNKTIONSPRÜFUNG	sehr gut		sehr gut		sehr gut		gut		gut		sehr gut		sehr gut		gut	
Fehler beim Formatieren (90 Prüfmuster)	keine		++		keine		++		keine		++		keine		++	
Elektromagnetische Eigenschaften	sehr gut		++		sehr gut		++		sehr gut		++		sehr gut		++	



Fabrikat (Anbieter-Adressen siehe Seite 26)	Sentinel Diskettes		Verbatim Vertriebsdiskette MD 550-01		Xvi2		Boeder Hi-Tec 2D Art.-Nr. 37540		3M Sicherheits-Diskette DS, DD Art.-Nr. 785-07M		Scotch Mini-Diskette 25 2D 48 tpi		IBM P/N 602 34 50		Kaufhof als Minidiskette Art.-Nr. 605 190		Memorex 25/2D Flexible Disk Art.-Nr. 3203 5220		Verbatim Datafile Minidisk MD 550-01 Art.-Nr. 10188	
Preis für 10 Disketten nach Markterhebung in DM	35,- bis 70,-		35,- bis 70,-		35,- bis 70,-		42,50,-		42,50,-		42,50,-		30,-		17,95,-		54,80,-		39,50,-	
Mittlerer Preis für 10 Disketten in DM	35,-		35,-		35,-		42,50,-		42,50,-		42,50,-		30,-		17,95,-		54,80,-		39,50,-	
Verpackung	Stülpfachteil		Stülpfachteil		Stülpfachteil		Stülpfachteil		Stülpfachteil		Stülpfachteil		Stülpfachteil		Faltfachteil		Faltfachteil		Stülpfachteil	
Material der Einstecktasche	Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss		Faserfloss	
test-Qualitätsurteil	GUT		GUT		GUT		ZUFRIEDENSTELLEND		ZUFRIEDENSTELLEND		ZUFRIEDENSTELLEND		ZUFRIEDENSTELLEND		ZUFRIEDENSTELLEND		ZUFRIEDENSTELLEND		ZUFRIEDENSTELLEND	
TECHNISCHE PRÜFUNG	sehr gut		gut		sehr gut		sehr gut		zufriedenstellend		zufriedenstellend		gut		zufriedenstellend		zufriedenstellend		zufriedenstellend	
Prüfung der Abmessungen	sehr gut		++		sehr gut		++		gut		+		sehr gut		++		mangelhaft*)		-	
Dauerprüfung	sehr gut		++		sehr gut		++		sehr gut		++		sehr gut		++		sehr gut		++	
Temperaturprüfung bis 50 °C	sehr gut		++		sehr gut		++		sehr gut		++		sehr gut		++		mangelhaft*) Hülle leicht-verformt		-	
Verarbeitung der Hülle	gut		+		sehr gut		++		gut		+		gut		+		sehr gut		++	
Verarbeitung der Magnetplatte	gut		+		zufriedenstellend*) Aufzeichnungsbereich etwas unvollständig poliert		++		sehr gut		++		mangelhaft*) Aufzeichnungsbereich unvollständig poliert		-		zufriedenstellend		+	
FUNKTIONSPRÜFUNG	gut		gut		gut		zufriedenstellend*)		zufriedenstellend		zufriedenstellend		zufriedenstellend		zufriedenstellend		zufriedenstellend		zufriedenstellend	
Fehler beim Formatieren (90 Prüfmuster)	gut		+		keine		++		keine		++		keine		++		mangelhaft bei einem Exemplar 5120 Bytes gesperrt		-	
Elektromagnetische Eigenschaften	gut		+		gut		+		gut		+		zufriedenstellend		o		zufriedenstellend		o	

*) Führt zur Abwertung (siehe -Ausgewählt - auf Seite 26) 1) Lt. Anbieter ab August '87 Nachfolgeprodukte mit geänderter Bezeichnung im Handel test 9/87 Bd. 5, 826 test 9/87 Bd. 5, 827

Test Disketten (5 1/4 Zoll)

Fortsetzung von Seite 25

kommt im ungünstigsten Fall völlig durcheinander. Ein Absturz mit Datenverlust ist dann leicht möglich.

Ähnlich können sich sogenannte Drop-ins auswirken. Das sind Störsignale, die das Laufwerk daran hindern können, die gespeicherten Daten fehlerfrei zu lesen. Daß es in diesem Prüfpunkt, der ebenso wie die Drop-outs, die Überschreibbarkeit, Modulation und mittlere Lesespannung unter unserer Überschrift »Elektronische Eigenschaften« läuft, keine Ausfälle gab, spricht für die Qualität der Disketten.

Technische Prüfung

Kein Ruhmesblatt für die Endkontrolle sind hingegen jene von uns entdeckten Datenräuber, bei denen schon die Verarbeitung zu wünschen übrig ließ. Ein Beispiel: Einige Memorex-Exemplare wichen in den Abmessungen so deutlich von der Norm ab, daß sie manchen Laufwerken mechanische Schwierigkeiten bereiten könnten. Ärgerlich ist es allemal, wenn eine Diskette klemmt. Unvollständig polierte Oberflächen auf der Magnetscheibe fielen uns gleich ein paarmal auf, besonders unangenehm bei Maxell und Scotch/3M. Weitgehend unempfindlich indes reagierten die meisten Test-Disketten auf hohe Temperaturen: 50 Grad Celsius überstanden fast alle ohne erkennbare Qualitätseinbußen. Bei einigen hatte sich allerdings nach dieser Prüfung die Hülle verformt.

Die Dauerprüfung meisterten alle Disketten ohne Probleme. Leider ist es dennoch nicht möglich, Aussagen über ihre Haltbarkeit auf Jahre hinaus zu machen. Unsere begrenzten Prüfzeiten lassen die dazu notwendigen Langzeitversuche einfach nicht zu. Wer allerdings von wichtigen Datenträgern Sicherheitskopien anfertigt, braucht erfahrungsgemäß nichts zu befürchten.

Trotz der beachtlichen prüftechnischen Unterschiede rühnen wohl die meisten Diskettenfehler vom falschen Benutzerverhalten her. So darf man die Magnetschicht keinesfalls mit Flüssigkeiten oder Fremdkörpern in Berührung bringen oder anfassen. Verschmutzte Disketten gehören in den Abfall, Reinigungsversuche mit Verdünnern oder speziellen Mittelchen sind nicht zu empfehlen. Die Magnetschicht wird beim Betrieb im Laufwerk durch ein Reinigungsvlies, das innen an der Hülle befestigt ist, ohnehin ständig gesäubert.

Disketten darf man niemals falten, knicken oder mit schweren Gegenständen belasten. Starke Hitze oder pralle Sonneneinstrahlung kann die Funktion ebenso

beeinflussen wie die Einwirkung äußerer Magnetfelder. Vorsicht vor allem bei Bürostrahlern mit Magnetverschlüssen! Daß ein Bekleben oder Beschreiben der Informationsschicht »Gift« ist, bedarf eigentlich gar keiner besonderen Betonung. Und zur Aufbewahrung von Disketten eignen sich staubdichte Plastikboxen be-

sonders gut. Einige Testfabrikate kann man bereits in ziemlich stabilen Aufstellboxen kaufen. Auch die oftmals mitgelieferten Stülpschachteln lassen sich recht gut wiederverwenden – und sei es nur für Transport- oder Versandzwecke.

Unser Rat

Datensicherheit ist (meist) keine Frage des Preises. Schon für knapp 15 Mark bekommt man eine Zehnerpackung »guter« Markendisketten ohne technische oder elektromagnetische Schwächen: Quelle Privileg macht's möglich.

Noch mehr Qualität für wenig Geld bietet die Boeder Disky, die ein »sehr gutes« Gesamturteil erreichte. Auch die Diskette von Fuji und die Verbatim Op-

tima glänzten mit der Bestnote. Gewaltig sind mitunter die Preisspannen beim gleichen Produkt. Zehn Exemplare der BASF Flexy Disk zum Beispiel gibt es schon für 35 Mark, sie können aber auch mehr als das Doppelte (76 Mark) kosten.

Zwar bieten die meisten Disketten gute Qualität, aber vor einem Hard- oder Software-Defekt kann man nie sicher sein. Deshalb: Speichern Sie wichtige Daten auf verschiedenen Disketten und lagern Sie diese getrennt!

Ausgewählt, geprüft, bewertet

Im Test: 20 Disketten für Personal Computer (5 1/4 Zoll, zweiseitig beschreibbar, doppelte Speicherdichte, 48 tpi, softsektoriert), darunter eine Produktleiche. Einkauf der Prüfmuster: Februar 1987.

■ **Preis**
Überregionale Handelserhebung im Juni 1987. Beim mittleren Preis handelt es sich um den Median, bei weniger als 10% Verkaufsstellen um das arithmetische Mittel. Katalogpreise, einheitliche Preise bei Eigenmarken des Händlers, Festpreise und Preise laut Anbieter werden ohne Preispanne angegeben.

■ **Bewertung**
Technische Prüfung 40 %
Funktionsprüfung 60 %

■ **Abwertung**
Das Test-Qualitätsurteil konnte nicht besser lauten als das Ergebnis der Funktionsprüfung. Das Urteil Funktionsprüfung konnte nicht besser sein als die Elektromagnetischen Eigenschaften. Bei einem »Mangelhaft« in der Formatierprüfung wurde die Funktionsprüfung um eine Stufe abgewertet. »Mangelhafte« Urteile in der Abmessungsprüfung, bei der Verarbeitung der Magnetplatte oder in der Temperaturprüfung hatten eine Abwertung der Technischen Prüfung um eine Stufe zur Folge. Bei einer »zufriedenstellenden« Verarbeitung der Magnetplatte konnte die Technische Prüfung bestenfalls »gut« sein.

■ **Technische Prüfung**
Die Außenabmessungen wurden mit Hilfe einer Fallehre nach ISO 7487/1-1985 entsprechend DIN 66247 Teil 1/1.85 geprüft. Die Dauerprüfung umfaßte 3 Millionen Umläufe mit fest eingestelltem Schreib-Lesepuls auf Spur 10. Vergleich der Lesespannung vor und nach dem Test – auch auf den Nachborspuren. Temperaturprüfung: Lagerung der Disketten 72 Stunden lang bei +50/-3 °C und 80 % relativer Feuchte, Akklimatisierung bei +23/-2 °C und ca. 50 % relativer Feuchte. Anschließend wurden die Lesespannung und die Außenabmessungen überprüft. Die Verarbeitung der Hülle ließen wir nach folgenden Kriterien untersuchen: Oberflächen-

struktur (Bildung von Fingerabdrücken), Ausformung der Felzanten, Ausführung des Verschlusses, Befestigung der inneren Gleitschicht (Vlieseinlage). In einer Sichtprüfung wurde festgestellt, ob sich am Schreib/Leseschlitz Flusen von der Vlieseinlage zeigen. Unter der Rubrik Verarbeitung der Magnetplatte haben wir folgende Teilprüfungen zusammengefaßt: Ausführung der Poltur (Besetzungsspuren), Grade des Verstärkungsringes, Reibstreifen (Streifen auf der Folle vom Herstellungsprozeß).

■ **Funktionsprüfung**
Zu Beginn des Tests wurden sämtliche zur Verfügung stehenden Prüfmuster (50 Stück pro Fabrikat) im MS-DOS Format (40 Spuren, 16 Sektoren) formatiert. Formatierfehler wurden protokolliert und im Testbericht festgehalten. Die elektromagnetischen Eigenschaften wurden mit Hilfe eines speziellen Disketten testers (Media Logic 2000) ermittelt und mit einer Bezugsdiskette der Physikalisch-Technischen Bundesanstalt (PTB) verglichen. Die folgenden Eigenschaften wurden geprüft: Signalausfall (drop out), Störsignal (drop in), Überschreibbarkeit (overwrite), Modulation, mittlere Lesespannungsamplitude (average signal amplitude) auf den Spuren 00 und 39 sowie die Auflösung (resolution) auf der inneren Spur. Die Meßwerte wurden mit den Mindestanforderungen der European Computer Manufacturers Association (ECMA) verglichen und darüber hinaus vergleichend beurteilt.

Anbieter-Adressen

BASF Aktiengesellschaft, Daten-Technik-VID/AMI, Postfach 5148, 6800 Mannheim 1 – Döbball & Boeder GmbH, Weiker Str. 50, 6035 Forchheim am Main – 3M Deutschland GmbH, Postfach 10 04 92, 4060 Hees 1 – Dymec, siehe 30-dcc – Fuji Photo Film (Europe) GmbH, Postfach 32 04, 4000 Düsseldorf – IBM Deutschland GmbH, Postfach 100, 7000 Stuttgart 80 – Kaufhof AG, Leonhard-Platz Str. 1, 5000 Köln 1 – Masell Europe GmbH, Postfach 11 07 48, 4000 Düsseldorf 11 – Memorex GmbH, Postfach 71 05 53, 6000 Frankfurt/AM 71 – Neuhus Copygraph GmbH, Postfach 43 40, 3000 Hannover 1 – Quelle International, 8510 Fürth 800 – RPH-Print-Produktions-Systeme GmbH, Rhein-von-Stein-Str. 7-8, 6057 Dattelnbach 1 – Sinterlab, siehe 3M – Sinterlab Computer Products Europe n. V., Bodenseestraße 12, B-3830 Welken-Belgien – Verbatim GmbH, Frankfurt Str. 63-69, 6036 Eschborn – Xerox GmbH, Frankfurt Str. 100, 6036 Eschborn.

Weihnachtsbummel (u. a.)

Zwischen Dienst und Feierabend- wie Wochenendhäuslesebauern ist es uns Provinzlerern doch tatsächlich möglich, sich zu einen samstäglichem Einkaufstag in der Großstadt aufzuffahren.

Er unterscheidet sich m.E. von Üblichen, weil unsere Computerfreake auch außer Haus nicht aufhören können Ihre Sprache zu sprechen, zum Bleistift:

Wir stehen vor einem Tisch mit irgendwelchen Skiunterhemden, mit Reizverschluß, die verschiedene Farben und Muster aufweisen. Nach begeisterten Gefühl meinerseits vernimmt man plötzlich "Du Spatzel vorne hängt ein Demo", sprich ein ausgepacktes Hemd liebevoll auf einen Bügel gehängt.

Nach diversen Einkäufen von weltlichen Dingen, wie emaillierter Pfannen, Bratpfännchen usw. landet man unausweichlich in der Computerabteilung.

Man benötigt, wenn schon sonst Nichts, Disketten, sind solche nicht noname und günstig im Angebot greift man natürlich gleich doppelt zu, in Form von singel seided dabel dänzidi (was nichts Anderes heißt als 2-seitig doppelte Dichte, aber daß dürfte den Wenigsten unter dem weiblen Geschlecht verborgen geblieben sein).

Super findets Mausebär dann noch wenn PD (Pablk Domein) zur Verfügung steht, wie in unserem Fall. Geduldlich wähle ich in der Diskettenbox und suche für Ihn jene welche heraus, die er nach literarischen Übersicht für gut befunden hat. Nun denn- nichts wie ran an die Kiste und zupf-wird Diskette für Diskette kopiert. Aber damit nicht genug, werden die gezupften wieder in die Box einsortiert so findet sich welch Jubel noch manches überaus interessante Disketten (3,5). Diese können ungehindert durch vorsorglichen Hamsterkauf kopiert werden.

Spätestens jetzt verabschiede ich mich, in der Hoffnung Abwechslung zu finden. Während ich das Schreibwarengeschäft durchkäme und all den Dingen mehr oder weniger Beachtung schenke, fällt meine Wahl auf zwei schnucklige Tüten (Verpackungsmaterial) die ich dezent wieder zurücklege, weil es die Menschentraube vor der Kasse es sicherlich nicht zuläßt meinen Geldbeutel vor einen halben Stunde zu zücken.

Irgendwann besinnt sich jeder seiner besseren Hälfte wieder, welche wohl in der Compterabteilung kleben bleibt. Nichts wie durchgewühlt zwischen Gesundheitsbewußten umhüllt von Knoblauchduft, glitzernden Kugeln u. baumelnden Engelchen, rauf zu Ihm.

Welch Wunder er stand nicht mehr an "seiner" Kiste, dafür ein bißchen weiter hinten. Mausebär hat Gleichgesinnten entdeckt, welcher großzügiger Weise das Neueste zupfen läßt.

Ich beschränke mich erquickender Weise aufs Beobachten von Leuten, da er mit verklärtem Blick u. innerlicher Wonne, ja doch nicht gänzlich von meiner Anwesenheit beglückt ist.

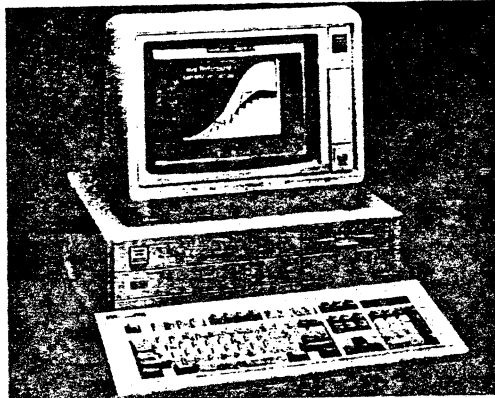
Meine Augen fixieren einen Softie, wohl volljährig, begleitet von Mama. Er begnügt sich etwas unsicher mit der Begutachtung von elektrischen Schreibmaschinen. Mit gespreiztem Fingerchen wird später sogar ausprobiert, während seine Mama als Auswahlkriterium Nr. 1 die Anschlaglautstärke auserkoren hat.

Später - viel später bekomme ich ein Küsschen auf die linke Wange, und Mausebär fragt an, was Spatzel noch kaufen will. Immerhin ein schnuckliges Ende von einem von vielen Besuchern in ähnlichen Compterabteilungen.

Drucker ohne Laser - mit Adler Vorkriegsmodell - Spatzel von Mausebär

HEFT
22
Dezember
1987

78



32-bit-Technik von Tandy: das Modell 4000; wenn die Zahlen stimmen - ein Preisknüller

So ganz vorn steht der amerikanische PC-Hersteller Tandy nicht in der öffentlichen Aufmerksamkeit. In seinem Heimatland USA behauptet er sich jedoch seit den Pioniertagen der Branche beständig auf einen der vorderen Plätze.

Dazu verhalfen ihm zunächst einmal seine eigene Ladenkette. Hinzu kommt noch, daß die Tandy-Produkte auch stets relativ kostengünstig, wenn auch nicht zu absoluten Tiefpreisen zu haben waren.

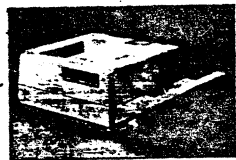
Jetzt hat sich das Unternehmen offenbar entschlossen, eine echte Preisbombe platzen zu lassen. Und das geschah zugleich mit dem Eintritt in den 32-bit-Club - mit dem Modell 4000 auf der Basis des Prozessors Intel 80386.

Bei Redaktionsschluss war ein deutscher Preis für die Maschine noch nicht bekannt. In den USA kostet das Gerät mit einer 20-MByte-Festplatte 3500 Dollar. Zum derzeitigen Umrechnungskurs entspricht das etwa 7500 Mark. Der Hersteller ist bekannt dafür, daß er seine amerikanischen Preise einigermaßen getreulich umsetzt, so daß hierzulande mit dem erwähnten Betrag zu rechnen ist.

Außerdem sieht der neue Computer recht herkömmlich aus. Er hat die Tandy-typischen runden For-

men und ist auch nicht extrem kompakt - allerdings auch lange nicht so wuchtig wie der originale PC/AT.

Bei der Diskette hat sich der Hersteller von vornherein für das modernere



HP-kompatibel: Der Laserdrucker LP 1000



Erster MS-DOS-Handheld von Tandy: Der 1400 LT

3 1/2-Zoll-Format entscheiden. Es lassen sich aber auch andere Laufwerke installieren - etwa ein 5 1/4-Zoll-Laufwerk, aber auch ein Bandlaufwerk. Dafür gibt es praktischerweise drei Montage-Positionen.

Bei der internen Konstruktion verzichtete der Hersteller ebenfalls darauf, einem aktuellen technischen Trend zu folgen: An-

ders als bei manchen neuen Rechnern sind beim Tandy 4000 die unverzichtbaren Schnittstellen (je eine parallel und seriell) und auch die Videofunktionen nicht auf der Hauptplatine integriert.

Allerdings tun hier auch schon ASICs (application specific integrated Circuits, anwendungsspezifische integrierte Schaltkreise) ihren Dienst. Acht dieser maßgeschneiderten Chips ersetzen eine Vielzahl herkömmlicher Bauteile und verbessern gleichzeitig die Zuverlässigkeit der Maschine.

Der Arbeitsspeicher des Rechners läßt sich mit 1-Megabit-RAM-Chips auf der Hauptplatine bis acht Megabyte ausbauen. Mit einer Zusatzplatine in dem speziellen Steckplatz erreicht sie bis zu 16 MByte. Verwendet man die derzeit noch eher gebräuchlichen 256-Kbit-Chips, sind es zwei beziehungsweise vier MByte. Leider gab es vorerst noch keine Informationen darüber, wie der Arbeitsspeicher organisiert ist, so daß man über seine Leistungen im Vergleich mit anderen 80386-Maschinen noch nichts sagen kann. Die übrigen Merkmale des Tandy 4000 sind eher konventionell zu nennen: Die Taktfrequenz beträgt 16 MHz, und es gibt vorerst wahlweise zwei Festplatten mit 20 und 40 MByte bei 65 beziehungsweise 28 Millisekunden mittlerer Zugriffszeit.

Bei dem optionalen Arithmetik-Prozessor für rechenintensive Anwendungen entschied man sich für den 80287. Er leistet zwar nicht so viel wie der 80387, ist aber eher verfügbar und nicht zuletzt auch billiger.

Überhaupt fällt der Tandy 4000 nicht durch außergewöhnliche Leistungen auf. Er rückt aber die Technik der leistungsstarken 32-bit-Personal-Computer für viele erstmals in erreichbare Nähe - zu seinem Preis gab es vor nicht allzulanger Zeit gerade einen normalen IBM PC/XT mit 8088-Prozessor.

Im übrigen hat Tandy zugleich mit der Vorstellung

des starken PC-Modells seine Produktpalette auch an anderen Stellen kräftig erweitert. Neu bei Tandy ist ein zum IBM PC kompatibler Laptop-Computer. Das Modell 1400 LT bietet zwei oberhalb der Tastatur eingebaute Diskettenlaufwerke (3 1/2 Zoll, 720 KByte) und einen beleuchteten Flüssigkristall-Bildschirm mit 640 mal 200 Bildpunkten und Graustufen.

Als Prozessor ist der (8088-kompatible) V20 von NEC eingebaut, der mit 7,16 oder 4,77 MHz läuft. Der Arbeitsspeicher ist 768 KByte groß. Davon dienen 128 KByte als RAM-Diskette oder als Drucker-Puffer. Das Gerät kostet in den USA rund 1600 Dollar, entsprechend knapp 3400 Mark. Es kann laut Hersteller mit einer Batterieladung rund vier Stunden lang arbeiten.

Zu den neuen Produkten zählt neben zwei Nadel-

Technische Daten

Tandy 4000

- Prozessor: Intel 80386
- 16 MHz Taktfrequenz, wahlweise Arithmetik-Prozessor 80287 (8 MHz)
- Arbeitsspeicher: 1 MByte, ausbaubar bis 16 MByte
- Massenspeicher: eine 3 1/2-Zoll-Diskette, 1,44 MByte, wahlweise 20- oder 40-MByte-Festplatte
- Bildschirm: 12 Zoll (30,8 cm) monochrom oder 12 Zoll (30,5 cm) farbig
- Schnittstellen: 1 x seriell, 1 x parallel (Steckplatine)
- Steckplätze: 8 im PC/AT-Format, 2 im PC/XT-Format (davon 1 belegt)
- 1,23-MHz-Steckplatine
- Abmessungen (Länge x Breite x Höhe): 40,1 cm x 48,3 cm x 16,5 cm
- Software: Betriebssysteme MS-DOS 3.2, XENIX System V, künftig auch OS/2

drucker-Modellen auch der Laserdrucker LP 1000. Dieses Gerät bietet die für die untere Leistungsklasse üblichen Merkmale: eine Auflösung von 300 mal 300 Punkten pro Zoll und ein (mechanisches) Drucktempo von sechs Seiten je Sekunde.

Das Gerät kann wie ein Laserjet Plus von Hewlett-Packard arbeiten. Der interne Arbeitsspeicher von 1,5 MByte erlaubt den Druck von Grafik.

Mit besseren Leistungsmerkmalen hat Tandy unter anderem sein PC/AT-kompatibles Modell 3000 ausgestattet. Die Maschine läuft jetzt mit 12 MHz (umschaltbar auf 6 MHz) und bietet als Grundausstattung 640 statt 512-KByte-RAM. Außerdem gibt es neuerdings ebenfalls drei Laufwerk-Plätze für Diskettenstationen verschiedenen Formats, Festplatte und

Bandlaufwerk. Der Grundpreis der Maschine ist geringfügig gesunken (in den USA von 2200 auf 2000 Dollar).

Tandy 4000

32 BIT ZUM NIEDRIGPREIS

Einen Personal-Computer mit dem Intel 80386 hat jetzt auch Tandy vorgestellt. Das Gerät ist ein echter Preisknüller.



Neues von der Diskothek

Wie auf dem Vorstandstreffen vereinbart, hat Hartmut 3 Public Domain Disketten für das Model3 und 4 angeschafft. Da diese Disketten auf meiner Anlage nicht lesbar sind kann ich hier nur eine kurze Übersicht und keine Programmbeschreibungen geben, vielleicht holt dies einer von Euch mal nach.

Nr.	Name	Titel	Nr.	Name	Titel	Nr.	Name	Titel
<u>UTILITY-Programme für das Model4</u>			<u>The Best of Model3</u>			<u>The Best of Model4</u>		
Disk22	M3TOM4/CMD	Convert		SINSTEP/CMD	Stepping Through Basic		TLOAD/CMD	A Happy Medium
	PIXEL/ASC			WORDCHKR/BAS	Letter Perfect		SETUP/JCL	Extra Strength DOS
	PXLDEMO/BAS	Pixel		WORD/BAS	Hinrichs' Word Processor		ENHANCE/OVL	ScripAid
	SORT4/ASC	Sortierprogramm		EXEC/CMD	Brick by Brick		MAKEDO/BAS	Command Performance
	SRT4DEMO/BAS			SPOOL/CMD	April Spool		GRAPHICS/CMD	Restored Art
	MAD/CMD	Memory Adresses		MPH/CMD	Smart Talk		M3TOM4/CMD	Convert
	PF/FLT	Programmable Filter Keys		DRAW/CMD	Drawing Boards		RUN4/BAS	Autorun
	RESTORE/CMD			TIMER3/BAS	Keeping Time		LIFE/CMD	Life
	BASREF/CMD	Crosscheck		PROJMIND/BAS	Time Keeper		WD/CMD	Windows
	RUN4/BAS	Autorun		EASYDATA/BAS	Little Wonder		MICROTAB/BAS	MicroTab
	HELPER/BAS			NOVACALC/BAS	Novacalc		PROJMIND/BAS	Project Minder
	TIMER4/CMD	Disk Timer		GRAPH/CMD	Grade-A Graphics		MAIN/BAS	Basic Data Base System
	FIND10/CMD	Finder		PATCHER/BAS	Patch Maker			
	FIND11/CMD			BLANKER/CMD	Blank Expressions			
	ZAP4/CMD	Disk Zapper		COMMANDO/CMD	Macro Economics			
	ECI/CMD	ECI Command						
	ECI/JCL							
	LOCATOR/BAS							
	SPACEMAP/CMD							

Das war's von der Club-Bibliothek,

Werner Förster

HEFT
22
Dezember
1987

82

83

Gutes ist zu berichten von der Clubbücherei! Zwar ist die Ausleihrate immer noch sehr niedrig, dafür wächst die Bibliothek aber munter weiter.

Das ist nicht zuletzt auf unser ehemaliges Mitglied Josef Konrad zurückzuführen, der dem Club nun auch noch das Buch "Einführung in 'C'" geschenkt hat. Ich möchte mich an dieser Stelle nochmals sehr herzlich für diese Spende bedanken!

Auch zwei sehr wichtige Bücher aus der Reihe "and other Mysteries", die uns beim Clubtreffen 86 abhanden gekommen waren, konnten zu einem recht günstigen Preis wiederbeschafft werden. Thomas Buskowiak, der sein Model 4p verkauft hat, war so freundlich uns hierbei zu helfen! Aus der gleichen Quelle stammt auch das Handbuch des CP/M 2.2-Betriebssystems, welches neu in unserer Sammlung ist.

Mit Gruß an alle Leseratten, Euer

Karntaut Obermann

Nr. 0018: Basic Disk I/O Faster And Better Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips und Tricks zum DOS, zur DISK, zu Files etc. zeigt (mit Programm-Diskette zum Buch).

Nr. 0019: Basic Faster And Better Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips, Tricks und Routinen in BASIC bietet.

Nr. 49 : Some Common BASIC Programs

Poole, Borchers, Koessel --- Osborne/McGraw-Hill

Mehr als 70 BASIC-Programme aus den verschiedensten Bereichen!

Nr. 50 : Das Handbuch des CP/M 2.2-Betriebssystems

--- Markt Technik

Originalhandbuch von Digital Research in englischer Sprache!

Nr. 51 : Der Einstieg in C

Paul Chirilan --- Markt Technik

Einführung in die Programmiersprache C anhand von über 70 praxisnahen Beispielen!

84

Buchbesprechung

Dr. Eberhard Zehendner: "Das Z80-Buch"
Markt & Technik Verlag, DM 59.--
ISBN 3-89090-219-7

Das Werk präsentiert sich mit stolzen 682 Seiten und einem nicht weniger umfangreichen wie gewichtigen Inhalt. Das Kapitelverzeichnis am Anfang enthält satte 29 Punkte. Es beginnt mit einer kurzen Beschreibung des Z80 und geht dann zielstrebig ins Eingemachte wie Listenverarbeitung, Bitfelder, Integer- und Realarithmetik, Bearbeitung von Zeichenketten, Mengen und Verbunde etc. Alle Themen sind in Assembler am Ende aufgeführt und soweit Übungsbeispiele, deren Lösungen erläutert sind. Es wird endlich einmal nicht auf das Buch von Herrn XYZ verwiesen, wo man näheres nachlesen kann und deshalb an dieser Stelle nur das Prinzip gezeigt wird. Die Assembler-Routinen sind keine 0815 Standardlösungen. Der Autor hat sich zu den beschriebenen Problemen durchaus seine Gedanken gemacht und recht raffinierte Lösungen gefunden. Aus meiner Sicht eine sehr gute Bereicherung der eigenen oder auch Club-Bibliothek, trotz des hohen Preises.

Tschüß *Juch*
Joegen, Turbo-Pascal,
Programmierpraxis
(Hanser Verlag, DM 58)

Das Kompendium für die

Ein ganz hervorragendes Werk für fortgeschrittene Turbo-Pascal-Freunde. Das Kompendium wird hauptsächlich angesprochen, trainiert werden aber hauptsächlich die Anwendungen von Turbo-Pascal, die sich mit anderen Programmiersprachen meines Wissens gar nicht oder nur schwer verwirklichen lassen wie z.B. Manipulationen mit Texten. Jede Anwendung ist mit einem Übungsprogramm versehen, sodaß das Gelernte sofort ausprobiert werden kann. Alle Programme sind auf drei Disketten zu haben, nur der Kostenpunkt von je DM 96 bzw. DM 258 für alle scheint mir persönlich viel zu hoch.

Musterprogramme, Utilities und Modulbibliotheken für raffinierte Lösungen in Turbo-Pascal, (Interest-Verlag DM 92)

Bei diesem Werk handelt es sich um eine Loseblattsammlung in einem DIN A 4 Ordner. Neben einer Einführung in das Turbo-Pascal System finden wir Hinweise zur Softwareerstellung, den ersten Teil eines Pascal-Kurses zum Selbststudium und eine Reihe von fertigen Programmen zur Wissenschaft, Technik, Finanzen etc., sowie Module für Utilities. Leider ist alles auf MS-DOS angestimmt. Ca. alle drei Monate gibt's Ergänzungslieferungen. Allerdings scheint mir der Preis von DM 92 sowie der Seitenpreis von 35 Pfg. für die Nachträge jenseits der Schmerzgrenze.

Hans-Martin

Gerald stellt in der 21. Ausgabe "Die Werkzeuge eines Assembler-Programmierers" vor. Wenn man Training als ein Werkzeug verstehen will, dann hat er eins vergessen. Der Z80 frißt dem Crack nur dann aus der Hand, wenn er am Ball bleibt, wann immer sich die Gelegenheit bietet.

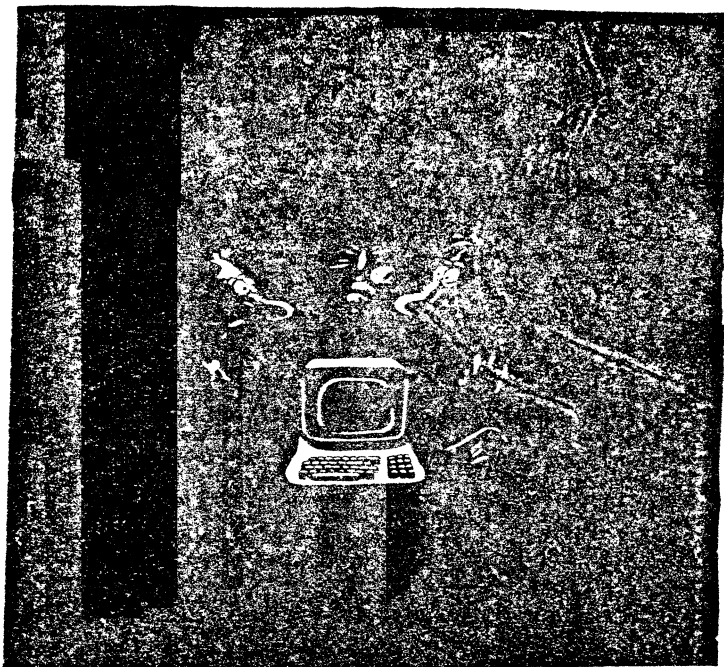
Aber aus einem anderen Grund beziehe ich mich auf Gerald's Beitrag. Da gibt es nämlich durchaus ein Buch, das nahezu alle Forderungen erfüllt, die er an die Paperware stellt. Es ist "Z80 - Einführung und Programmierung" von Nichols, Nichols, Ray, erschienen im Eletor-Verlag. Es hat nur drei Nachteile:

1. Es kostet DM 35,- obwohl man es vom Umfang her nur als Taschenbuch bezeichnen kann.
2. Es enthält alle Angaben zu den Z80-Befehlen nur in Hex, obgleich für den fortgeschrittenen Programmierer die oktale Darstellung wünschenswert wäre.
3. Die wirklich sinnvolle ASCII-Tabelle enthält es auch nicht. Man kann sie sich selber erstellen, wenn mal eine Stunde frei ist. (Dem wirklichen Crack ist auch diese Stunde kaum je vergönnt.)

Das Buch ist als Anleitung für den Nanocomputer NBZ80 geschrieben, bezieht sich jedoch nur in wenigen Passagen auf dieses Gerät. Der ganze Rest ist allgemeine Z80-Knowledge vom Feinsten, angereichert mit Arbeitsaufgaben an den Lernenden. Im Anhang finden sich üppige Tabellen mit allem, was auch der Profi nicht auswendig weiß. Bloß eben nicht oktal.

Arnulf Sopp

Schließ doch den Computer an und frag ihn, wo der Lichtschalter ist ...

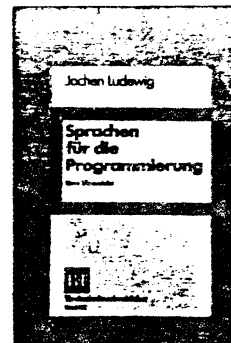


Der sehr aufmerksame Leser wird vielleicht bemerken, daß das folgende Buch in PASCAL 1/87 bereits reviewt wurde. Dennoch und vor allem weil der Strom der Anfragen nach einem Buch für »Einsteiger« nicht abreißt, hier nochmals das wirklich für jeden geeignete Werk:

Sprachen für die Programmierung: Eine Übersicht
von Jochen Ludewig
Bibliographisches Institut, Mannheim, 1985,
[Reihe B.I. Hochschultaschenbücher, Band 622],
194 Seiten,
Preis: DM 22,80
ISBN 3-411-00622-6

Der Autor ist Software-Spezialist und hat es unternommen, für eine Unterrichtsreihe der

Erwachsenenbildung, einen Überblick über die verschiedenen gängigen Programmiersprachen zusammenzustellen.



Ein weit reichender Blick von oben

Daraus ist das vorliegende Buch entstanden. Es sollte damit kein umfassendes Lehrwerk für Informatiker geschaffen werden, sondern ein populäres »Aufklärungsbuch« für alle, die mit primitiven Sprachen (Zitat) arbeiten, um diese zur Beschäftigung mit höheren Programmiersprachen anzuregen.

Behandelt werden - mehr oder weniger ausführlich - die Sprachen Assembler, FORTRAN, COBOL, PL/I, die ALGOL-Familie, Pascal, Modula, C, BASIC, LISP, LOGO, Smalltalk und Prolog, sowie am Ende eine Klassifikation von Programmiersprachen, die an eine Einteilung nach Chomsky angelehnt ist.

Betont wird durchgängig der Einfluß, den die Wahl der verwendeten Sprache auf die eingesetzten Programmiertechniken und Programmierfehler hat.

Der Stil der Darstellung ist unverkrampft und eingängig. Der Band enthält zudem zahlreiche Abbildungen, die sowohl informativ sind, als auch den Text auflockern.

Als besonderen Witz stellt der Autor jeder Sprache als Allegorie ein bestimmtes Transportmittel gegenüber: der Sprache C z.B. einen Land-Rover »viel

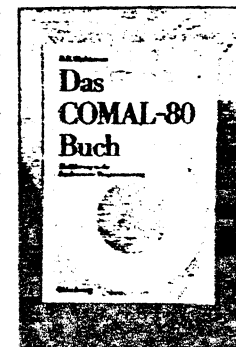
Flexibilität speziell für bodennahe Aufgaben«, der Sprache LISP einen Helikopter »schwer zu handhaben, aber auf speziellen Gebieten unschlagbar«.

Das Erscheinen dieses Buches in der Reihe der Hochschultaschenbücher könnte den allgemein interessierten Leser vielleicht von der Lektüre abhalten, weil er ein akademisch ausgerichtetes Buch erwartet.

Dies ist aber ganz und gar nicht der Fall! Der Band sei allen Programmierern im Hobby und im Beruf warm empfohlen: der Lohn der Lektüre ist ein Überblick der wichtigsten eingesetzten Sprachen heute und eine Entscheidungshilfe für die Beurteilung neuer Sprachen morgen.

(Wolfgang J. Weber)

Das COMAL-80 Buch
von B. R. Christensen
Oldenbourg Verlag,
München 1986,
269 Seiten,
Preis: DM 29,80
ISBN 3-486-28268-2



Ein Buch für Einsteiger

Das Buch enthält eine einführende Darstellung der Programmiersprache COMAL-80, wobei auf Seiten des Lesers keinerlei Vorkenntnisse über den Umgang mit Computern und das Programmieren vorausgesetzt werden.

Die Erläuterungen des Autors sind extrem verständlich, detailliert und genau: er führt Schritt für Schritt in den Gebrauch der Sprache auf dem Computer C-64 (Commodore) ein.

Es handelt sich um ein Lehrbuch für den Anfänger und nicht, wie es der Titel vielleicht vermuten ließe, um eine umfassende Sprachbeschreibung von COMAL-80.

Der Aufbau des Buches ist gut durchdacht.

Alle grundlegenden Merkmale und Befehle der Sprache werden behandelt, jedoch folgt die Darstellung keiner strengen technischen Systematik, sondern der Logik des Lernens. Das heißt: Informationen werden dort gegeben, wo sie gebraucht werden, und auch vor Wiederholungen scheut sich der Autor nicht.

Die Struktur der einzelnen Kapitel überzeugt durch die durchgängige Berücksichtigung der Bedürfnisse der Lernenden: am Anfang steht jeweils ein Überblick, in den Text sind zahlreiche Übungen eingebettet, und jedes Kapitel endet mit Aufgaben.

Die grafische Gestaltung des Typoskripts ist ansprechend und aufgelockert.

Da durchgängig auf das Cerit C-64 Bezug genommen wird, konnten auch Besonderheiten dieses Systems wie z.B. Sprites angesprochen werden.

Wer COMAL-80 auf anderen Geräten einsetzt, z.B. Schneider CPC-464 oder IBM PC), muß den Text an einigen Stellen (meist geringfügig) abändern.

COMAL ist eine ideale Sprache für Programmieranfänger, weil sie einen erstaunlich hohen Komfort bietet (Einfachheit, deutsche Systemmeldungen, hohe Geschwindigkeit bei der Programmausführung und vieles mehr) und dabei nicht den Weg zu höheren Sprachen wie Pascal oder C verbaut.

Das vorliegende liebevoll geschriebene Buch ist gerade für Programmieranfänger sehr gut geeignet: Leser mit Vorkenntnissen wären hingegen mit weniger breit angelegten Darstellungen besser bedient.

(Wolfgang J. Weber)

- I M P R E S S U M -

<u>1. Vorsitzende</u>	Hartmut OBERMANN Schwalbacher Straße 6 6209 Heidenrod 1 ☎ 06124 /3913
<u>2. Vorsitzende</u>	Gerald SCHRÖDER Am Schützenplatz 14 2185 Seevetal 1 ☎ 04185 /2682
<u>Hardwarekoordinator</u>	Eckehard KUHN Im Dorf 14 7443 Frickenhausen 1 ☎ 07022 /45417
<u>Diskotheke</u>	Werner FÖRSTER Christoph-Krebs-Straße 9 8720 Schweinfurt ☎ 09721 /21841
<u>Redaktion</u>	Jens NEUEDER Panoramastraße 21 7178 Michelbach /Bilz ☎ 0791 /42877
<u>Autoren</u>	Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.
<u>Druck</u>	Peter Spieß Trugenhofenstraße 27 8859 Rennertshofen 1 ☎ 08434 /454
<u>Bankverbindung</u>	des CLUB 80 Postgirokonto Peter STEVENS Sonderkonto CLUB 80 Konto-Nummer 285 491 - 465 Postgiroamt Dortmund BLZ 440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle
der jeweiligen eingeschickten Infobeiträge
durch die Redaktion.

Hallo Club-80er,

nun habe ich es doch noch geschafft, vor Weihnachten, mit dem neuesten INFO fertig zu werden, wenn es auch erst 1988 in Euren Briefkästen landen wird.

Somit kann ich mich schon jetzt recht herzlich für Eure rege Mitarbeit an den 87'er CLUBINFO's bedanken. Ohne Euren Fleiß beim Erstellen der Artikel, ausdenken und erproben der Hard- und Software, wären wir sicher nicht in der Lage so ein umfangreiches INFO zu bieten.

Also nochmals Dank, weiter so, und bitte nicht nachlassen.

Allen unseren CLUB-80-Mitgliedern wünsche ich (im nachhinein) ein frohes Weihnachtsfest (mit vielen brauchbaren Geschenken), und einen gelungenen Rutsch 1988 mit einer Wahnsinns-Silvester-Orgie.

Nun noch zu der aktuellen vor Euch liegenden INFO. Im Anhang daran findet Ihr eine Referenzkarte mir diversen Druckersteuerzeichen der gängigsten Drucker. Sicher ist sie eine Hilfe für Euch beim Umschreiben von Programmen mit Druckroutinen. Weiterhin habe ich mal wieder eine Sonderpublikation beigelegt. Unter dem Namen "Mikroelektronik" werden Euch wichtige und häufig gebrauchte Ausdrücke von A bis Z erklärt. Die Sonderpublikation kann sicher einigen von Euch als kleines Nachschlagewerk dienen.

Ich wünsch Euch ein Frohes Fest und Guten Rutsch ins neue Jahr !
Bis zur neuesten INFO grüßt Euch

Jens

P.S.: Da ich aus den bekannten Zeitgründen nicht dazu komme mich persönlich, für die von Euch erhaltenen und noch kommenden Weihnachts- und Neujahrsgrüße, zu bedanken, möchte ich dies hiermit tun.

Stand: Dezember 1987
 Bitte überprüft Eure Daten auf Richtigkeit
 und teilt mir Unregelmäßigkeiten mit.
 Die Redaktion

Name	Vorname	Straße	PLZ	Stadt	Telefon	privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Mistedt	04182	/8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204	/62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431	/241907	// 0431 /74047
Betz	Heinrich	St. Wolfgangstraße 13	8551	Hausen	09191	/31698	// 09191 /611108
Brandel	Hermann	Cäciliastraße 30	8439	Postbauer-Heng	09180/	493	// 0911 /219-245
Braun	Harald	Postfach 8011	2300	Kiel 17	-		// -
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451	/7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261	/69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121	/508218	// -
Drowälder	Bernd	Hügel 1	4441	Mettringen	05233	/4320	// 02557 /1236
Eilers	Hans-Joachim	Thomas-Dehler-Straße 6	2900	Oldenburg	0441	/ 53239	// 0441 / 7983945
Emmerich	Helmut	Waldstraße 5	6682	Ottweiler	06824	/4114	// -
Förster	Werner	Christoph-Krebs-Straße 7	8720	Schweinfurt	09721	/21841	// 09721 /51256
Grömottka	Uwe	Lange Reihe 40	2803	Meyne	0421	/ 88496	// 0421 /300-68/0
Hartmann	Hans-Günter	Mowenstraße 7	2876	Berne 2	04406	/ 1911	// 0421 / 2483405
Heidenreich	Ulrich	Herderstraße 35	4300	Essen 1	0201	/ 2821/0	// -
Held	Mantred	Stirnerstraße 22	8635	Pleinfeld	09144	/6563	// 0911 /2195245
Hentz	Werner	Am Irankgarten 20	6457	Maintal 2	06109	/66625	// 0691 /583639
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127	/70024	//
Hill	Peter	Bergstraße 65	6754	Otterberg	-		// -
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948	/542	// 02921 /70431
Krispin	Michael	Schwanzstraße 8	4130	Moers 3	02841	/73690	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022	/45417	// 09171 /832665
Lachmann	Wolfgang	Am Ringofen 11	6407	Schlitz	06642	/6940	// 069 /832051
Loose	Gerhard	Viefhaushof 42	4306	Essen 13	0201	/212608	// -
Mand	Harald	Kleinflintbeker Straße 7	2302	Flintbek bei Kiel	04347	/3629	// 0431 /3013500
May	Holger	Marienstr. 9	5768	Sundern 2	02935	/1668	// -
Misioch	Maldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911	/506051	// 0911 /107945
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201	/55052	// -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040	/7246083	// 04151 /8891-37
Neueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791	/42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124	/3913	// -
Obscherningkat	Helmut	1 RUE DES BRUYERES	F-68360	Soultz	0033089/762690		// -
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221	/872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01	/7847418	//
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141	/603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8000	Fürstenfeldbruck	08141	/43791	//
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133	/4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122	/43551	// 04103 /605310
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203	/331383	// 0203 /331383
Schmid	Alexander	St. Cajetan-Straße 38/VII	8000	München 80	089	/495326	// -
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	-		// -
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221	/707044	// -
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203	/740897	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105	/2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025	/1631	// 08025 /41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	5305	Alfter	0228	/643853	// -
Sopp	Arnulf	Makenitzstr. 8	2400	Lübeck 1	0451	/791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859	Rennertshofen 1	08434	/454	// 08431 /7041684
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a. TN.	05401	/99585	// 05401 /30096
Stevens	Peter	Danziger Straße 11	5042	Erfstadt	02235	/42350	// -
Stober	Reiner	Nelkenstraße 12	3216	Salzhemmendorf 4	05153	/1564	// -
Sörensen	Rüdiger	Thomas-Mann-Straße 3A	6500	Mainz 1	06131	/32860	// 06131 /395268
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362	/42497	// 02362 /23127
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711	/731205	//
Wacker	Fred	Postfach 2246	7550	Rasstatt	07222	/52574	// -
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035	/3361	// -
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531	/54686	// -

Das DOS-Sier ist eine Extra-Leistung von DOS International. Hier finden Sie in übersichtlicher tabellarischer Form wissenswerte Informationen rund um Ihren PC und seine Peripherie. Die Seiten sind zum Ausschneiden oder Fotokopieren gedacht. So erhalten Sie mit der Zeit ein einzigartiges Archiv von geballten Informationen zu Ihrem Computersystem.

Drucker- steuer-codes auf einen Blick

Verschiedene Drucker benutzen unterschiedliche Steuer-codes. Gleiche Funktionen funktionieren somit auf dem einen Drucker problemlos, während sie auf dem nächsten nicht zu finden sind. Das heutige DOS-Sier gibt Ihnen den Überblick in diesem Gewirr.

(Nils Teller/hg)

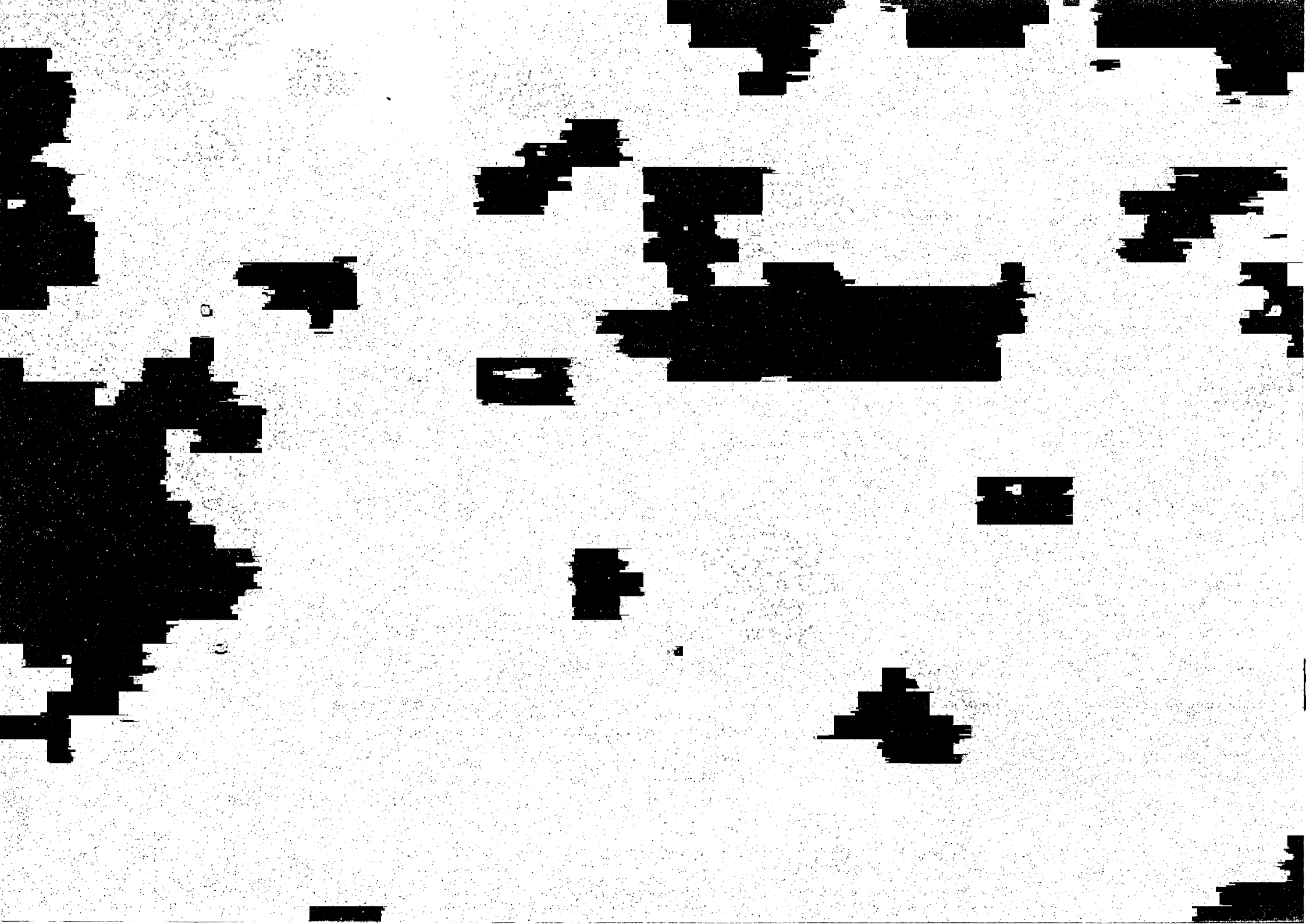
Druckermode / Steuerung	ESC-P-Norm	Jejimal	Epson FX 80	Fujitsu DX 2100	Citizen 120 D	Star SG 10	SG 10 IBM-Mode
Summer (Bell)	BEL	7	BEL	BEL	BEL	BEL	BEL
Rückschritt (Backspace)	BS	8	BS	BS	BS	BS	BS
Drucker initialisieren	ESC @	27, 84	ESC @	ESC @	ESC @	ESC @	ESC @
Papierendeckennung aus	ESC 8	27, 86	ESC 8	ESC 8	ESC 8	ESC 8	ESC 8
Papierendeckennung ein	ESC 9	27, 87	ESC 9	ESC 9	ESC 9	ESC 9	ESC 9
Druckkopf an den Zeilenanfang	ESC <	27, 60	ESC <	ESC <	ESC <	ESC <	ESC <
Druckwegoptimierung (Toggle)	ESC U	27, 85	ESC U n	ESC U n	ESC U n	ESC U n	ESC U n
Druckerpuffer löschen	CAN	24	CAN	CAN	CAN	CAN	CAN
letztes Zeichen im Puffer löschen	DEL	127	DEL	DEL	DEL	DEL	DEL
Drucker selektieren	DC 1	17	DC 1	DC 1	DC 1	DC 1	DC 1
Drucker inaktivieren	DC 3	19	DC 3	DC 3	DC 3	DC 3	DC 3
höchstwertige Bit (MSB) auf 1 setzen	ESC >	27, 62	ESC >	ESC >	ESC >	ESC >	ESC >
höchstwertige Bit (MSB) auf 0 setzen	ESC -	27, 61	ESC -	ESC -	ESC -	ESC -	ESC -
MSB-Modus löschen	ESC #	27, 35	ESC #	ESC #	ESC #	ESC #	ESC #
Druckqualität							
Draft und NLQ (Toggle)	ESC x	27, 120	n.v.	ESC % (2) (0)	ESC x n	ESC B 4	ESC 4 n
Schriftart							
Pica (Normal)	ESC P	27, 80	ESC P	ESC P	ESC P	ESC B 1	ESC P
Elite	ESC M	27, 77	ESC M	ESC M	ESC M	ESC B 2	ESC M
Proportional (Toggle)	ESC p	27, 112	ESC p	ESC p n	ESC p n	ESC p n	ESC p n
Schriftformen							
Breitschrift für die gegenwärtige Zeile	SO	14	SO	SO	SO	SO	SO
Breitschrift (SO) löschen	DC 4	20	DC 4	DC 4	DC 4	DC 4	DC 4
Breitschrift, unbegrenzt	ESC SO	27, 14	ESC SO	ESC SO	ESC SO	ESC SO	ESC SO
Breitschrift (Toggle)	ESC W	27, 87	ESC W n	ESC W n	ESC W n	ESC W n	ESC W n
Schmalschrift (Condensed) für eine Zeile	SI	15	SI	SI	SI	SI	SI
Schmalschrift, unbegrenzt	ESC SI	27, 15	ESC SI	ESC SI	ESC SI	ESC SI	ESC SI
Schmalschrift löschen	DC 2	18	DC 2	DC 2	DC 2	DC 2	DC 2
Fettschrift	ESC E	27, 89	ESC E	ESC E	ESC E	ESC E	ESC E
Fettschrift löschen	ESC F	27, 70	ESC F	ESC F	ESC F	ESC F	ESC F
doppelter Anschlag	ESC G	27, 71	ESC G	ESC G	ESC G	ESC G	ESC G
doppelter Anschlag löschen	ESC H	27, 72	ESC H	ESC H	ESC H	ESC H	ESC H
Kursivschrift	ESC 4	27, 52	ESC 4	ESC 4	ESC 4	ESC 4	ESC 4
Kursivschrift löschen	ESC 5	27, 53	ESC 5	ESC 5	ESC 5	ESC 5	ESC 5
Unterstreichen (Toggle)	ESC -	27, 45	ESC - n	ESC - n	ESC - n	ESC - n	ESC - n
Hochstellen	ESC S 0	27, 83, 0	ESC S 0	ESC S 0	ESC S 0	ESC S 0	ESC S 0
Tiefstellen	ESC S 1	27, 83, 1	ESC S 1	ESC S 1	ESC S 1	ESC S 1	ESC S 1
Hoch- / Tiefstellen löschen	ESC T	27, 84	ESC T	ESC T	ESC T	ESC T	ESC T
Wahl mehrerer Druckarten (Makro)	ESC !	27, 33	ESC ! n	ESC ! n	ESC - 3 n	ESC ? n	ESC ! n
Zeichensätze							
internationale Zeichensätze wählen	ESC R	27, 82	ESC R n	ESC R n	ESC R n	ESC 7 n	ESC R n
IBM-Zeichensatz 1	ESC 7	27, 55	ESC 7	ESC 7	ESC 7	n.v.	ESC 7
IBM-Zeichensatz 2	ESC 6	27, 54	ESC 6	ESC 6	ESC 6	n.v.	ESC 6
ladbarer Zeichensatz (Toggle)	ESC %	27, 37	ESC %	ESC %	ESC %	ESC %	ESC % n
benutzerdefinierte Zeichen	ESC &	27, 38	ESC & n1..nm	ESC & n1..nm	ESC & n1..nm	ESC * n1..nm	ESC & n1..nm
ROM-Zeichensatz in Zeichengenerator übertragen	ESC : 0	27, 58, 48	ESC : 0	ESC : 0	ESC : 0	ESC * 0	ESC : 0
Grafik							
Standarddicke	ESC K	27, 75	ESC K	ESC K	ESC K	ESC K	ESC K
doppelte Dicke	ESC L	27, 76	ESC L	ESC L	ESC L	ESC L	ESC L
doppelte Dicke und doppelte Geschwindigkeit	ESC Y	27, 89	ESC Y	ESC Y	ESC Y	ESC Y	ESC Y
vierfache Dicke	ESC Z	27, 90	ESC Z	ESC Z	ESC Z	ESC Z	ESC Z
Makro-Modus	ESC *	27, 42	ESC *	ESC *	ESC *	ESC *	ESC g
9-Nadel-Modus	ESC ^	27, 94	ESC ^	ESC ^	ESC ^	n.v.	n.v.
Zeilenverschiebung (einmalig)							
Wagenrücklauf (Carriage Return)	CR	13	CR	CR	CR	CR	CR
einfacher Zeilenverschiebung (Line Feed)	LF	10	LF	LF	LF	LF	LF
Zeilenverschiebung um n/216° (9 Nadeln)	ESC j	27, 106	ESC j n	ESC j n	n.v.	n.v.	n.v.
Zeilenverschiebung um n/180° (24 Nadeln)	ESC j	27, 74	n.v.	ESC j n	n.v.	n.v.	ESC j n
absolute Steuerung (vertikal)							
1/8° Zeilenabstand	ESC 0	27, 48	ESC 0	ESC 0	ESC 0	ESC 0	ESC 0
7/72° Zeilenabstand	ESC 1	27, 49	ESC 1	ESC 1	ESC 1	ESC 1	ESC 1
1/6° Zeilenabstand	ESC 2	27, 50	ESC 2	ESC 2	ESC 2	ESC 2	ESC 2
n/216° Zeilenabstand	ESC 3	27, 51	ESC 3 n	ESC 3 n	ESC 3 n	n.v.	ESC 3 n
n/72° Zeilenabstand	ESC A	27, 65	ESC A n	ESC A n	ESC A n	ESC A n	ESC A n
Seltenverschiebung (Form Feed)	FF	12	FF	FF	FF	FF	FF
Seitenlänge (Zeilen)	ESC C	27, 67	ESC C n	ESC C n	ESC C n	ESC C n	ESC C n
Seitenlänge (Zoll)	ESC C 0	27, 67, 0	ESC C 0	ESC C 0	ESC C 0	ESC C 0	ESC C 0
Überspringen der Perforation	ESC N	27, 78	ESC N n	ESC N n	ESC N n	ESC N n	ESC N n
Löschen von ESC N	ESC O	27, 79	ESC O	ESC O	ESC O	ESC O	ESC O
Vertikaler Tabulator	VT	11	VT	VT	VT	VT	VT
VT-Stops	ESC B	27, 66	ESC B n1..nm	ESC B n1..nm	ESC B n1..nm	ESC B O n1..nm	ESC B O n1..nm
VFU-Kanal wählen (Vertical Format Unit)	ESC /	27, 47	ESC / n	ESC / n	ESC / n	n.v.	n.v.
VFU-Stop setzen	ESC b	27, 98	ESC b n	ESC b n	ESC b n	ESC b n	ESC b n
absolute Steuerung (horizontal)							
rechten Rand setzen	ESC Q	27, 81	ESC Q n	ESC Q n	ESC Q n	ESC Q n	ESC Q n
linken Rand setzen	ESC 1	27, 49	ESC 1 n	ESC 1 n	ESC 1 n	ESC M n	ESC 1 n
Horizontaler Tabulator	HT	9	HT	HT	HT	HT	HT
HT-Stops	ESC D	27, 68	ESC D n1..nm	ESC D n1..nm	n.v.	ESC D n1..nm	ESC D n1..nm

Begriffe aus der

Mikroelektronik

Sonderheft

Redaktion EXUB 80
Dezember 1987



Begriffe aus der Mikroelektronik

Die Mikroelektronik ist heute zu einem Gegenstand des täglichen Lebens geworden. Begriffe wie *digital*, *Mikroprozessor* oder *Chip* (meist allerdings fälschlich für verkapselte integrierte Bauelemente gebraucht) gehören inzwischen zur Umgangssprache. In der folgenden Zusammenstellung werden die wichtigsten Begriffe aus "Technologien" und Anwendungen der Mikroelektronik in alphabetischer Reihenfolge aufgelistet und erläutert. Der Leser hat damit die Möglichkeit, einen Begriff, dessen Bedeutung ihm unbekannt oder unklar ist, schnell zu finden und sich in wenigen Sätzen über ihn zu informieren. Tauchen innerhalb einer Erklärung Begriffe auf, die an anderer Stelle behandelt sind, so wird auf sie durch einen Pfeil (→) hingewiesen.

Die Zusammenstellung erhebt keinen Anspruch auf Vollständigkeit, umfaßt aber alle wichtigen und häufig gebrauchten Ausdrücke.

A

Abfallzeit (fall time)

Maß für die Steilheit eines Impulsabfalls. Es ist die Zeit, in der ein Signal von 90% auf 10% seines Dachwertes abgefallen ist (Bild 1).

Äquivalenzschaltung, EXNOR-Schaltung (exclusive NOR-gate)

Logische → Digitalschaltung, welche die Äquivalenz-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn beide Eingangssignale gleich sind, gleichgültig, ob sie beide 1 oder 0 sind (s. Bild 13). Mathematische Beschreibung: $X = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$ (gelesen: X = A UND B ODER A NICHT UND B NICHT).

Addierer, Addierschaltung (adder)

Schaltung zum Addieren zweier → Binärzahlen. im einfachsten Fall ein → Halbaddierer, der Summe und Übertrag der Addition zweier einstelliger Zahlen bildet. Ein → Volladdierer, der aus zwei Halbaddierern zusammengeschaltet werden kann, berücksichtigt bei der Addition zweier einstelliger Zahlen noch den Übertrag von der Addition der nächstniedrigen Stelle. Zum Addieren n-stelliger Zähler müssen entsprechend n Volladdierer verwendet werden.

Analog-Digital-Wandler, A/D-Wandler (analog-to-digital-converter)

Schaltung zur Umformung eines analogen Eingangssignals in ein digitales Ausgangssignal. Die Zahl der Ausgangsbits bestimmt

das Auflösungsvermögen, ein n-bit-Wandler kann 2^n unterschiedliche Signalwerte erfassen.

Anstiegsgeschwindigkeit (slew rate)

Maß für die Ausgangsspannungsänderung eines Verstärkers, gemessen in Spannung pro Zeit (V/µs). Häufig als maximale (höchste) Anstiegsgeschwindigkeit angegeben, so daß aus diesem Wert berechnet werden kann, wieviel Zeit für eine bestimmte Spannungsänderung erforderlich ist oder welche Spannungsänderung höchstens in einer bestimmten Zeit erwartet werden kann.

Anstiegszeit (rise time)

Maß für die Steilheit eines Impulsanstiegs. Es ist die Zeit, in der ein Signal von 10% auf 90% seines Dachwertes angestiegen ist (s. Bild 1).

Antivalenzschaltung, EXOR-Schaltung (exclusive-OR gate)

Logische → Digitalschaltung, welche die Antivalenz-Funktion (Exklusiv-ODER) darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn beide Eingangssignale unterschiedliche Werte haben (s. Bild 13). Mathematische Beschreibung: $X = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$ (gelesen: X = A UND B NICHT ODER A NICHT UND B).

Instabile Schaltung, astabiler Multivibrator (astable circuit)

Schaltung zum Erzeugen eines freischwingenden Rechtecksignals, dessen Frequenz durch RC-Glieder** bestimmt wird. Sie enthält im allgemeinen zwei kapazitiv gekoppelte → Inverter mit einer zusätzlichen kapazitiven Rückkopplung vom Ausgang des zweiten auf den Eingang des ersten Inverters.

Ausgangsfächer, Ausgangslastfaktor (fan-out)

Innerhalb einer → logischen Digitalfamilie das Maß dafür, wie viele Eingänge von einem Ausgang parallel angesteuert werden können. Ein Ausgang mit dem Ausgangsfächer 15 kann demnach 15 Eingänge mit jeweils dem → Eingangsfächer 1 ansteuern.

* Technologie Gesamtheit der technischen Arbeitsvorgänge in einem Fertigungsbereich.
 ** R: Abkürzung für Widerstand.
 C: Abkürzung für Kondensator, Kapazität.

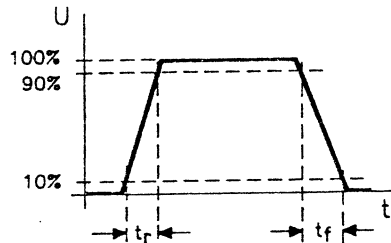
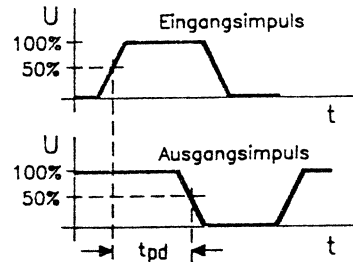


Bild 1: Definition von Anstiegszeit t_r , Abfallzeit t_f und Verzögerungszeit t_{pd}



B

BCD, binär codiertes Dezimalsystem (BCD, binary coded decimal)

Ein → Binärcode, bei dem jede Ziffer einer Dezimalzahl in ein binäres 4-bit-Wort umgeformt wird. Z.B. wird die Dezimalzahl 30 zu 00110000. Der Code ist bewertet mit 1-2-4-8 (von rechts nach links) und wird deshalb auch als 1-2-4-8-Code bezeichnet.

BCD-Zähler (BCD counter)

Ein → Binärzähler, der aus hintereinandergeschalteten Gruppen von jeweils vier Zählerstufen besteht. Jede dieser Gruppen liefert an vier Ausgängen eine Dekade des BCD-Codes und gibt nach dem zehnten Eingangsimpuls einen Übertragsimpuls an die nächste Gruppe ab.

BFL (BFL, buffered FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → DFETs und → Schottkydioden besteht (s. Bild 2). Die Transistoren T_1 und T_2 bilden die logische Eingangsverknüpfung, die Transistoren T_3 und T_4 sind als ohmsche Widerstände geschaltet, der Transistor T_5 ist der Ausgangstransistor in Sourceschaltung, und die beiden Dioden D_1 und D_2 sind Pegelschiebedioden, die den Ausgangspegel an den Eingangspegel angleichen. Die BFL ist sehr schnell, hat eine hohe Ausgangsbelastbarkeit, aber auch eine relativ hohe Verlustleistung.

Binärcode (binary code)

Ein Code, bei dem jedes Codeelement nur zwei verschiedene Werte, z.B. 0 und 1, annehmen kann.

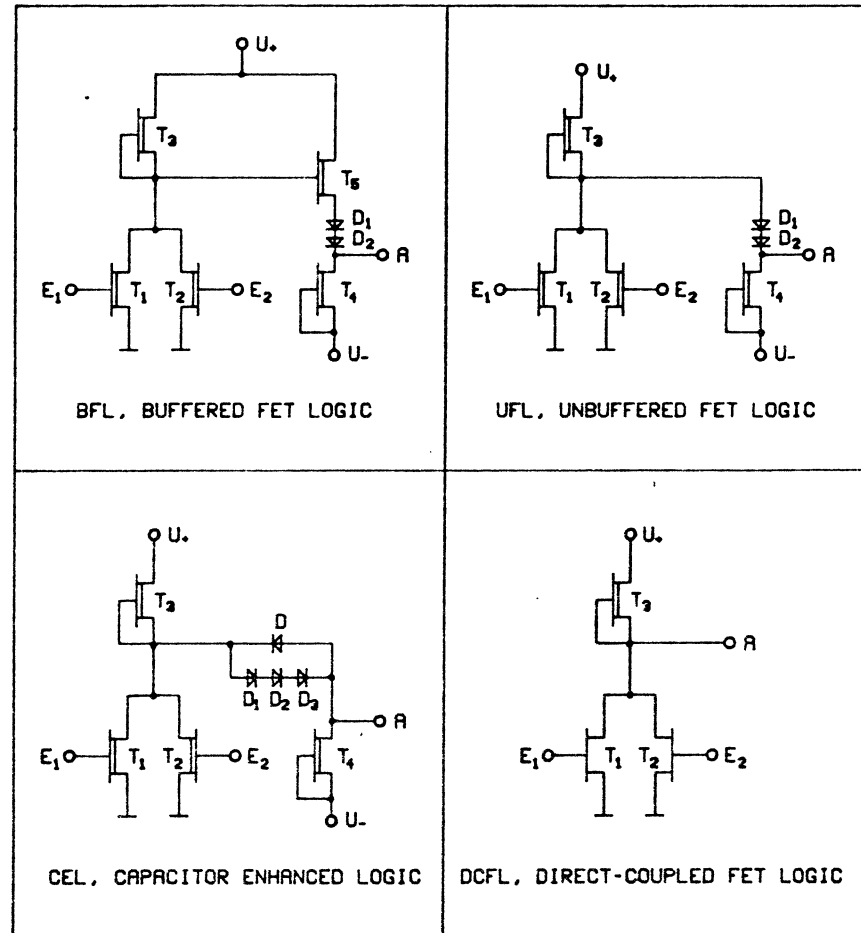


Bild 2: Grundsaltungen von GaAs-Digitalschaltungsfamilien

Binär-Dezimal-Umwandlung
(binary-to-decimal conversion)

Umwandlung des Binär- in den Dezimal-Code, z. B. der Binärzahl 1101 in die entsprechende Dezimalzahl 13. Dies kann elektrisch in einem Binär-Dezimal-Wandler geschehen, der üblicherweise 4 Eingänge und 10 Ausgänge hat (BCD-Dezimal-Wandler). Jeder der 10 Ausgänge entspricht einer der Ziffern von 0 bis 9 und führt dann Ausgangssignal, wenn die zugehörige binäre Signalkombination an den Eingängen anliegt.

Binäre Speicherzelle
(binary cell)

Die kleinste Einheit eines binären Speichers. Sie kann die logischen Werte 0 oder 1 annehmen und somit 1 bit speichern.

Binärzähler
(binary counter)

Schaltung, die nach jeweils zwei Eingangssignalen (dies können Impulse oder Impulsflanken sein) den Ausgangszustand ändert und somit durch 2 teilt. Der Zählbereich kann durch Hintereinanderschaltung gleichartiger Stufen beliebig erweitert werden. Bei n Stufen sind 2ⁿ Zählerzustände gegeben, so daß die Ziffern von 0 bis 2ⁿ-1 im → Binärcode dargestellt werden.

Bipolare Bauelemente
(bipolar devices)

Halbleiterbauelemente mit abwechselnd aufeinanderfolgenden p- und n-dotierten Schichten, z. B. npn-Transistor, Pin-Diode. Im Gegensatz dazu → Unipolare Bauelemente.

Bipolare Speicher
(bipolar memories)

Speicher (z. B. → RAMs oder → ROMs), die in Bipolartechnik ausgeführt sind. Im Vergleich mit → CMOS-Speichern sind höhere Arbeitsgeschwindigkeiten erreichbar, jedoch auf Kosten höherer Leistungsaufnahme und geringerer → Packungsdichte.

Bipolartechnologie
(bipolar fabrication)

Zur Herstellung von → bipolaren Bauelementen ist eine Reihe von → Maskierungs- und → Diffusionsschritten erforderlich. Ausgangspunkt ist ein einkristallines Halbleiter-Substrat, in das von der Oberfläche her Fremdatome (→ Dotiermittel) eindiffundiert werden, wodurch entweder n- oder p-leitende Schichten erzeugt werden. Die Dotierung geschieht örtlich begrenzt über Masken, die durch → Fotolithografie aufgebracht werden. Heute wird ausschließlich der → Planarprozeß in Verbindung mit → Epitaxie angewendet.

Bistabile Schaltung, Flip-Flop
(bistable circuit)

Logische Schaltung, die zwei stabile Ausgangszustände hat, zwischen denen sie umgeschaltet werden kann. Der jeweils eingeschaltete stabile Zustand bleibt dabei auch nach Wegnahme der Eingangssignale erhalten. Bistabile Schaltungen können deshalb zur Informationsspeicherung verwendet werden.

Bit
(bit)

Abkürzung für *binary digit* (binäres Zeichen). Es ist die kleinste Informationseinheit; das binäre Zeichen kann nur zwei unterschiedliche Werte, z. B. 0 oder 1, annehmen.

Boolesche Algebra, Schaltalgebra
(Boolean algebra)

Auf den englischen Mathematiker Boole zurückgehendes Verfahren, um logische Zusammenhänge mathematisch beschreiben zu können. Grundlage für die Berechnung von Systemen, die mit

→ logischen Schaltungen aufgebaut werden. Da eine logische Entscheidung nur entweder richtig oder falsch sein kann (1 oder 0), ist eine technische Darstellung mit binären digitalen Schaltungen, die ebenfalls nur zwei diskrete Signalzustände aufweisen, möglich.

Bus
(bus)

Im allgemeinen mehrpolige Verbindung (8polig, 16polig) zwischen Datensendern und Datenempfängern, häufig bidirektional, d. h. in beiden Richtungen verwendbar.

C

CFL
(CFL, capacitor enhanced FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → DFETs und → Schottkydioden besteht (s. Bild 2). Die Transistoren T₁ und T₂ bilden die logische Eingangsverknüpfung, die Transistoren T₃ und T₄ sind als ohmsche Widerstände geschaltet. Die Dioden D₁-D₃ sind Pegelschlebedioden, die den Ausgangspegel an den Eingangspegel angleichen. D₄ ist in Sperrichtung gepolt und wirkt als Kondensator, der parallel zu D₁-D₃ geschaltet ist, wodurch die Schaltgeschwindigkeit gegenüber der → UFL erhöht wird.

Chip
(chip)

Unverkapseltes Halbleiterbauelement (Transistor, Integrierte Schaltung), das entweder zum Einbau in ein Gehäuse vorgesehen ist oder direkt in → Hybridschaltungen eingesetzt wird. Die Verbindungen mit der Schaltung werden durch gebondete* Drähtchen hergestellt.

CML, Logik mit Stromsteuerung
(CML, current mode logic)

Familie von Logikschaltungen, bei denen die Schalttransistoren im ungesättigten Bereich arbeiten, wodurch hohe Schaltgeschwindigkeiten, symmetrische Übertragungskennlinien und geringes Überspringen erreicht werden, → ECL.

CMOS, Komplementäre MOS-Technik
(CMOS, complementary metal oxide semiconductor)

Eine → MOS-Technik, bei der im Gegensatz zur → NMOS- oder → PMOS-Technik beide Transistortypen gemeinsam verwendet werden. Der einfache → Inverter besteht aus der Reihenschaltung eines N-Kanal- und eines P-Kanal-Transistors, so daß im Ruhezustand und unabhängig vom logischen Eingangssignal immer einer der beiden Transistoren gesperrt ist. Strom fließt nur während des Umschaltvorganges (Bild 3). CMOS-Schaltungen haben deshalb geringe Verlustleistungen; sie haben hohe Störicherheit und sind unempfindlich gegenüber Spannungs- und Temperaturschwankungen. CMOS-Schaltungen, die durch → Ionenimplantation hergestellt werden, arbeiten noch mit Betriebsspannungen von 1,5 V.

D

DCFL
(DCFL, direct-coupled FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → EFETs, → DFETs und → Schottkydioden besteht (s. Bild 2). Die Transistoren T₁ und T₂ (EFETs) bilden die logische Eingangsverknüpfung, der Transistor T₃ (DFET) ist als ohmscher Widerstand geschaltet. Eine Pegelangleichung wie bei → BFL, → UFL oder → CFL ist hier nicht erforderlich. Die Leistungsaufnahme ist gering, die Ausgangsbelastbarkeit ist vergleichbar wie bei der UFL.

* Bonden: Verbinden von Metallen durch Druck, Temperaturerhöhung und/oder Ultraschallenergie.

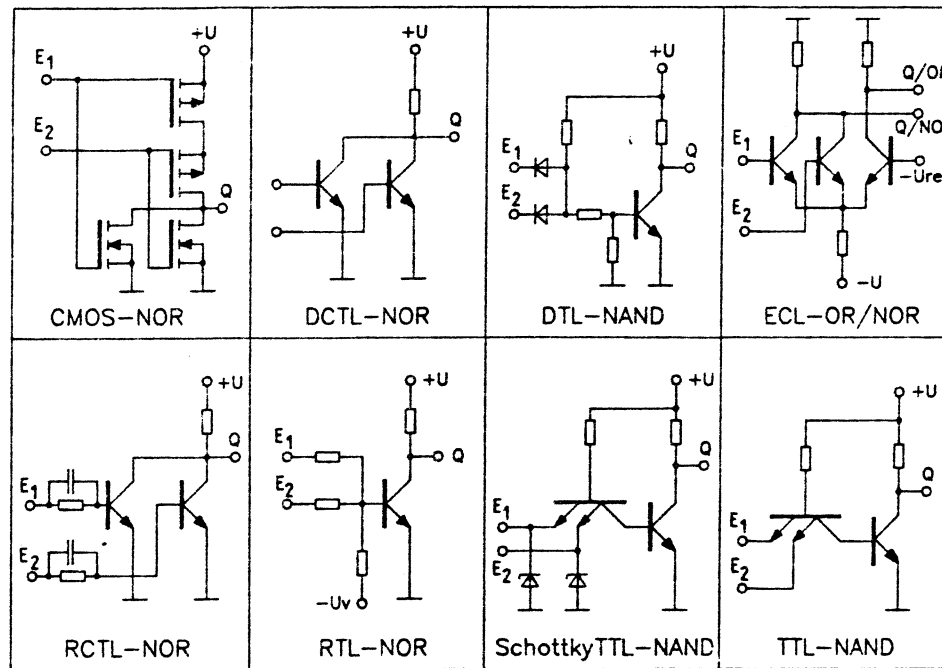


Bild 3: Grundschaltungen der wichtigsten Logikfamilien in vereinfachter Darstellung

U Betriebsspannung
+U Versorgung
Uref Referenzspannung
E₁, E₂ Eingänge
Q Ausgang

DCTL, Direkt gekoppelte Transistorlogik
(DCTL, direct-coupled transistor logic)

Nicht mehr gebräuchliche Logikfamilie, bei der parallel- oder hintereinandergeschaltete Transistoren die logische Verknüpfung bilden, wobei die Basisanschlüsse direkt von der vorhergehenden Schaltung angesteuert werden (s. Bild 3).

Decoder, Decodierschaltung
(decoder)

Schaltung, die einen Code in einen anderen Code umwandelt, z. B. BCD in dezimal, dezimal in binär, BCD in 7-Segment.

DFET, FET vom Verarmungstyp
(DFET, depletion mode FET)

Eine Art des → Feldeffekttransistors, bei dem schon dann ein leitfähiger Kanal zwischen Drain und Source besteht, wenn das Gate noch keine Steuerspannung aufweist. Je nach Polarität der Gatespannung nimmt die Leitfähigkeit weiter zu (Anreicherungsbetrieb) oder ab (Verarmungsbetrieb). Bei direkter Verbindung von Gate und Source (Gatespannung Null) wirkt der DFET wie ein ohmscher Widerstand.

D-Flipflop
(D-type flip-flop)

Digitale → Folgeschaltung mit mindestens einem D-(Daten-)Eingang und einem C-(Takt-, Clock-)Eingang und häufig zwei komplementären* Ausgängen. Der logische Wert des Datensignals

wird mit dem Taktsignal auf den Ausgang übertragen und gespeichert (s. Bild 10). D-Flipflops werden als Zähler, Frequenzteiler, Speicher und Schieberegisterstufen verwendet.

Dickfilmschaltungen, Siebdruckschaltungen
(thick-film circuits)

Eine Art der → Hybridschaltungen, die üblicherweise im → Siebdruckverfahren hergestellt werden. Dabei können auf einem → Substrat, meist Keramik, alle passiven Bauelemente, wie Leiterbahnen, Streifenleitungen, Leitungskreuzungen, Widerstände und Kondensatoren, als verhältnismäßig dicke Schichten (5 bis 40 µm) aufgedruckt und in Brennprozessen mit dem Substrat und untereinander fest verbunden werden. Dickfilmschaltungen werden sowohl für die Konsumelektronik** als auch in hochwertigen Ausführungen für schnelle Digitalschaltungen (Bild 4) und für Mikrowellenschaltungen hergestellt. In Verbindung mit Halbleiterchips lassen sich auch umfangreiche Systeme auf geringem Raum integrieren. Das abwechselnde Drucken von Leiterbahnebenen und dielektrischen Isoliererebenen ergibt Vielschichtschaltungen (Multilayerschaltungen) für komplizierte Verdrahtungsstrukturen.

* Komplementäre Ausgänge: Ausgangsbar, welches das logische Signal und das invertierte Signal gleichzeitig abgibt.
** Konsumelektronik: Überbegriff für elektrische Bauelemente oder Geräte die für den allgemeinen Gebrauch bestimmt sind. Im Gegensatz dazu: *Kommerzielle Elektronik*, die zum Einsatz in Industriebetrieben, an Hochschulen oder an Forschungsanstalten bestimmt ist.

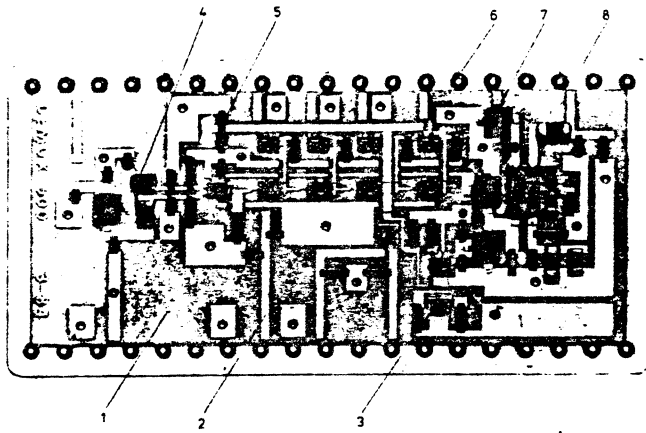


Bild 4: Hybrid integrierte Dickfilmschaltung. Beispiel eines schnellen digitalen Multiplaziers (700 Mbit/s)
Die Siebdruckschaltung ist mit verkapselten Einzeltransistoren, Chipkondensatoren und monolithisch integrierten Schaltungen in Chipform bestückt

1 Keramiksubstrat	5 gedruckter Widerstand
2 50- Ω -Leitung	6 Anschlußleift
3 integrierte Schaltung	7 Einzeltransistor
4 Chipkondensator	8 Gehäuseboden

Dielektrische Isolation
(dielectric isolation)

Üblicherweise sind die Bauelemente bipolarer Integrierter Schaltungen durch Sperrschichten voneinander isoliert, was zu störenden Sperrschichtkapazitäten und Leckströmen führt. Bestimmte Herstellungsverfahren benutzen deshalb Siliziumdioxid (SiO₂, Quarz), das durch Oxidation der Siliziumoberfläche entsteht, als isolierendes Dielektrikum. Dadurch werden die Streukapazitäten wesentlich verkleinert und die Leckströme beseitigt.

Differenzverstärker
(differential amplifier)

Symmetrische Schaltung aus mindestens zwei, möglichst identischen Transistoren mit Konstant-Stromspeisung (Bild 5). Die Differenz der Ausgangsspannungen ist dabei proportional (verhältnismäßig) der Differenz der Eingangsspannungen, während gleiche Eingangsspannungen, unabhängig von ihrer Höhe, keine Änderung der Ausgangsspannung zur Folge haben. Der Differenzverstärker ist wesentliches Bauteil aller \rightarrow Operationsverstärker und der \rightarrow Emitter-gekoppelten Logik.

Diffusion
(diffusion)

Verfahren zum \rightarrow Dotieren eines Halbleiterkristalls mit Fremdatomen, um eine bestimmte Art (P oder N) und Höhe der Leitfähigkeit zu erreichen. Der Diffusionsvorgang geschieht bei hohen Temperaturen. Die Eindringtiefe des \rightarrow Dotiermittels ist zeitabhängig.

digital
(digital)

Ziffernmäßig, von engl. digit Ziffer. Digitale Signale beschränken sich auf bestimmte diskrete Werte oder Wertebereiche eines Signals. Das einfachste und gebräuchlichste Digitalsystem ist das duale, zweiwertige, bei dem nur 2 Signalzustände möglich sind, die allgemein mit 0 und 1 bezeichnet werden, wobei z.B.

0 eine niedrige und 1 eine hohe Spannung sein kann. Daneben werden für besondere Anwendungen noch Digitalssysteme mit mehr als zwei Ziffern verwendet, z. B. das Ternärsystem mit drei Ziffern.

Digital-Analog-Wandler, D/A-Wandler
(digital-to-analog converter)

Schaltung zur Umformung eines digitalen Eingangssignals in ein analoges Ausgangssignal. Die Zahl der Eingangsbits bestimmt die Stufung des Ausgangssignals. Ein n-bit-Wandler erzeugt ein Ausgangssignal mit bis zu 2ⁿ diskreten Signalwerten. Durch ein Tiefpaßfilter mit geeigneter Grenzfrequenz wird dieses treppenförmige Signal geglättet.

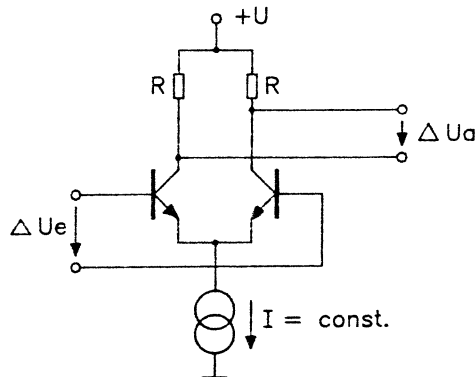


Bild 5: Prinzipschaltung des Differenzverstärkers. Die Differenz der Ausgangsspannungen ist proportional zur Differenz der Eingangsspannungen.

Digitale Anzeigen
(digital displays, displays)

Sie werden zur Darstellung von Dezimalziffern oder auch als alphanumerische Anzeigen zur Darstellung von Buchstaben, Ziffern und Zeichen verwendet. Aufgebaut sind sie als Leuchtdioden (\rightarrow LED), Flüssigkristalle (\rightarrow LCD) oder Gasentladungsröhren. Am bekanntesten sind die 7-Segment-Anzeigen (Bild 6), die aus 7 einzeln ansteuerbaren Leuchtsegmenten die Ziffern von 0 bis 9 nachbilden und z. B. über einen BCD-7-Segment- \rightarrow Decoder an eine digitale Schaltung angeschlossen werden.

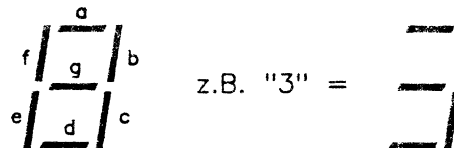


Bild 6: 7-Segmentanzeige
Die Segmente sind mit a bis g bezeichnet und werden einzeln angesteuert, bei „3“ z. B. die Segmente a, b, c, d und g

Digitalisaltungen
(digital circuit)

Schaltungen, die mit \rightarrow digitalen Signalen arbeiten, deren Eingangs- und Ausgangssignale demnach auf bestimmte Signalewerte oder Wertebereiche beschränkt sind. Im normalen Sprachgebrauch werden darunter ausschließlich binäre digitale Schaltungen verstanden, die nur zwei unterschiedliche Signalzustände, 0 und 1 oder \rightarrow Low und \rightarrow High, aufweisen.

DIP, Dual-in-Line-Gehäuse
(DIP, dual in-line package)

Gehäuse aus Kunststoff oder Keramik mit zwei parallelen Anschlußstiftreihen zur Aufnahme integrierter Schaltungen. Die Stifte sind im 1/10-Zoll-Raster angeordnet. Das Standard-DIP hat 14 Anschlüsse und ist 6,5 mm x 19 mm groß. Es gibt aber auch z. B. DIPs mit nur 8 und mit bis zu 60 Anschlüssen.

Disjunktion
(disjunction)

Logische Verknüpfung, die dann erfüllt ist, wenn mindestens eine Eingangsvariable den Wert 1 hat. Sie stellt die ODER-Funktion dar und kann mittels \rightarrow ODER-Schaltungen realisiert werden. Mathematisch lautet die ODER-Funktion zweier Variabler $X = A \vee B$ (gesprochen: A ODER B).

Displays
(displays)

Anderer Ausdruck für \rightarrow (digitale) Anzeigen.

Dotieren
(doping)

Einbau von Fremdatomen (\rightarrow Dotiermittel) in einen Halbleiterkristall zur Erhöhung der Leitfähigkeit durch Erzeugung von Überschubladungen (Elektronen oder Löcher).

Dotiermittel
(dopant)

Fremdatome, mit denen ein Halbleiterkristall gezielt verunreinigt wird zur Beeinflussung der Art und Höhe der Leitfähigkeit, z. B. durch \rightarrow Diffusion oder \rightarrow Ionenimplantation.

Dünnschichtschaltungen
(thin-film circuits)

Eine Art der \rightarrow Hybridschaltungen, die durch Aufdampfen oder Aufputtern* in Verbindung mit \rightarrow Fotolithografie und Ätztechniken hergestellt werden. Dabei können auf einem \rightarrow Substrat, meist Keramik oder Glas, alle passiven Bauelemente, wie Leiterbahnen, Streifenleitungen, Leitungskreuzungen, Widerstände und in Grenzen auch Kondensatoren, als sehr dünne (0,01 bis 1 μ m) Schichten aufgebracht werden. Leiterbahnen werden häufig noch galvanisch verstärkt (bis auf etwa 10 μ m). Dünnschichtschaltungen werden als Analogschaltungen für Frequenzen bis zum MHz-Bereich (Tantaltechnik), hauptsächlich aber als hybrid integrierte Mikrowellenschaltungen (\rightarrow MIC) bis zu höchsten Frequenzen eingesetzt. Wie bei den \rightarrow Dickfilmschaltungen werden auch hier die Halbleiterbauelemente als \rightarrow Chips nachträglich in die Schaltungen eingebaut (Bild 7).

Dynamische MOS-Schaltungen
(dynamic MOS circuits)

Im Gegensatz zu den üblichen Speichern, die statisch arbeiten, indem sie die Speicherwirkung von \rightarrow bistabilen Schaltungen ausnutzen, wird in dynamischen MOS-Schaltungen die Information in Form einer Kondensatorladung gespeichert. Dieses Verfahren läßt sehr hohe \rightarrow Packungsdichten zu, hat aber den Nachteil, daß die Kondensatorladungen über nicht zu vermeidende Leckströme abfließen. Dynamische Speicher müssen deshalb periodisch nachgeladen (aufgefrischt, refreshed) werden. Dies geschieht durch automatisches Auslesen und Wiedereinschreiben des Speicherinhalts.

Dynamisches RAM
(dynamic RAM)

Bestimmte \rightarrow dynamische MOS-Schaltung (Speicher mit wahlfreiem Zugriff, \rightarrow RAM), bei der die Information in Form von Kondensatorladungen gespeichert wird.

DTL, Dioden-Transistorlogik
(DTL, diode-transistor-logic)

Logikfamilie, bei der ein Transistor als \rightarrow Inverter dient und die logische Verknüpfung durch Diodennetze geschieht (s. Bild 2).

E

EAROM, Elektrisch veränderbarer Festwertspeicher
(EAROM, electrically alterable ROM)

Sonderform eines Festwertspeichers (\rightarrow ROM), der elektrisch programmiert werden kann und dessen Speicherinhalt nichtflüchtig ist, d. h. dessen Speicherinhalt auch bei Wegfall der Betriebsspannung erhalten bleibt. Grundsätzlich handelt es sich um einen Schreib-Lesespeicher (\rightarrow RAM), jedoch geht der Schreibvorgang sehr viel langsamer vor sich als der Lesevorgang, so daß diese Speicherart ausschließlich als ROM verwendet wird.

ECL, Emittergekoppelte Logik
(ECL, emitter-coupled logic)

Logikfamilie, deren Schalttransistoren im nichtgesättigten Bereich arbeiten (\rightarrow CML) und deren Grundprinzip auf den \rightarrow Differenzverstärker zurückzuführen ist (s. Bild 2). Es ist die schnellste zur Zeit verwendete Logikfamilie auf Siliziumbasis mit Schaltzeiten von bis unter 1 ns. Die Schaltungen haben symmetrische Übertragungskennlinien, deshalb ist die Störsicherheit in beiden logischen Zuständen etwa gleich groß. Der \rightarrow logische

* Sputtern: Dünnschichtprozess, bei dem das aufzubringende Material durch Beschuss mit Edelgasionen aus der Materialoberfläche herausgelöst wird und sich als Film auf der Substratoberfläche niederschlägt.

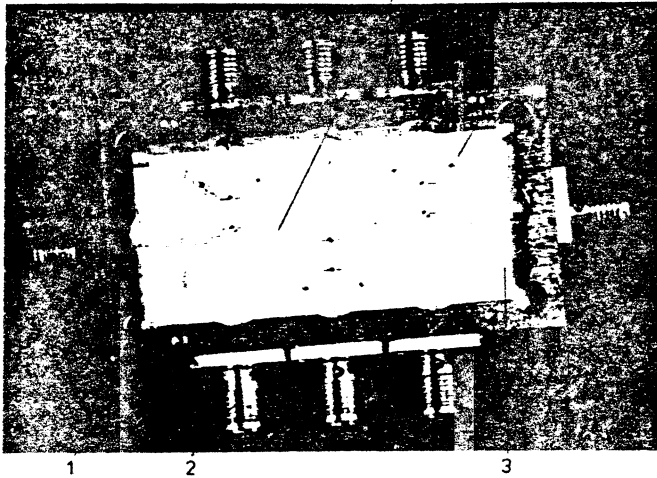


Bild 7: Hybrid integrierte Dünnschichtschaltung. Auf einem Keramiksubstrat (2,5 cm x 5 cm) sind Leitungsstrukturen aufgesputtert. Chipdioden sind leitfähig angeklebt und gebondet.

- 1 Gehäuse
- 2 Koaxialbuchse
- 3 Keramiksubstrat
- 4 Diodenchip
- 5 Leiterbahn

Hub liegt bei etwa 1 V, der Ausgangsfächer ist wegen der Ausgangsemittlerfolger hoch, es können 50-Ω-Leitungen direkt angeschlossen werden. Auf Grund des Differenzverstärkerprinzips kann die Referenzspannung gegen Temperatur- und Betriebsspannungsänderungen kompensiert werden. Außerdem steht stets ein komplementäres Ausgangspaar zur Verfügung.

EEROM, Elektrisch löschbarer Festwertspeicher
(EEROM, electrically erasable ROM)

Sonderform eines Festwertspeichers (→ ROM), der elektrisch programmiert werden kann und dessen Speicherinhalt nichtflüchtig ist. Im Gegensatz zum → EAROM muß dieser Speicher vor dem Neuprogrammieren durch eine angelegte Spannung vollständig gelöscht werden. Löschen- und Programmiervorgang können fast beliebig oft wiederholt werden.

EFET, FET vom Anreicherungstyp
(EFET, enhancement mode FET)

Eine Art des → Feldeffekttransistors, der bei Spannung Null zwischen Gate und Source nichtleitend ist. Erst ab einer bestimmten Gatespannung, deren Polarität umgekehrt zur Drain-Source-Spannung sein muß, setzt die Leitfähigkeit des Drain-Sourcekanals ein und nimmt mit wachsender Gatespannung zu.

Eingangsfächer, Fan-in
(fan-in)

Der Eingangsfächer stellt in Form einer dimensionslosen Zahl die Belastung dar, die ein bestimmter Eingang einer logischen Schaltung innerhalb einer logischen Familie für einen ihn ansteuernden Ausgang bildet. Normale Eingänge haben den Eingangsfächer 1, es gibt jedoch auch Eingänge (z. B. Takteingänge

bei mehrstufigen Schieberegistern) mit höheren Werten. Die Summe der Eingangsfächer aller parallel an einen Ausgang geschalteten Eingänge darf den Ausgangsfächer dieses Ausgangs nicht überschreiten.

Elektronenstrahlolithografie
(electron beam lithography)

→ Maskenherstellung im Elektronenstrahlverfahren.

Epitaxie
(epitaxy)

Epitaxie ist ein technologischer Vorgang im → Planarprozeß, bei dem man auf der Oberfläche eines Einkristalls aus der Gas- oder Flüssigphase eine Schicht aus dem gleichen Material aufwachsen läßt, wobei diese Schicht die Kristallstruktur fortsetzt. Durch gleichzeitiges Zusetzen von → Dotiermitteln wird die neue Schicht dotiert. Es entstehen dadurch beliebig dünne und gleichmäßiger dotierte Schichten, als dies bei der → Diffusion möglich ist.

EPROM, Elektrisch programmierbarer Festwertspeicher
(EPROM, electrically programmable ROM)

Sonderform eines Festwertspeichers (→ ROM), der elektrisch programmiert werden kann und dessen Speicherinhalt nichtflüchtig ist. Wie ein → EEROM muß auch dieser Speicher vor dem Neuprogrammieren vollständig gelöscht werden, jedoch geschieht dies hier durch Belichten der Halbleiteroberfläche mit UV-Licht*. Im Gehäuse dieser Speicher ist deshalb ein Quarzfenster eingebaut, durch welches das UV-Licht eindringen kann. Der Löschen- und Programmiervorgang kann nahezu beliebig oft wiederholt werden.

* UV-Licht: Licht im nicht sichtbaren Spektralbereich, oberhalb des Violetten.

Exklusions-Schaltung, Ausschließungs-Schaltung
(exclusion circuit)

Logische Schaltung mit zwei Eingängen, welche die → UND-NICHT-Verknüpfung bildet. Das Ausgangsergebnis ist nur dann wahr, wenn die erste Eingangsangabe wahr, die zweite dagegen falsch ist. Mathematischer Ausdruck: $X = A \wedge B$ (gelesen A UND B NICHT).

Exklusiv-NICHT-ODER-Schaltung
(exclusive NOR)

→ Äquivalenzschaltung

Exklusiv-ODER-Schaltung
(exclusive OR)

→ Antivalenzschaltung

F

FAMOS

(FAMOS, floating-gate avalanche-injection MOS)

Eine Ausführungsform des → EPROMs. Als Speicherzelle dient ein → MOS-Transistor, dessen → Gate aus Silizium besteht und durch völlige Einbettung in hochisolierendes SiO₂ potentialmäßig schwimmend (floating) angeordnet ist (Bild 8). Bei der Programmierung wird durch Anlegen einer hohen Spannung ein Lawinendurchbruch (avalanche-injection) hervorgerufen, der Ladungsträger zum Gate transportiert. Diese Ladung kann nach Wegnahme der Programmierspannung von dem isolierten Gate nicht mehr abfließen und wirkt für den betreffenden Transistor wie eine dauernd anliegende Gatespannung. Die Abfrage des Speicherinhalts geschieht durch Messung des Drain-Source-Widerstandes jedes einzelnen Transistors. Gelöscht wird der gesamte Speicherinhalt mit UV-Licht.

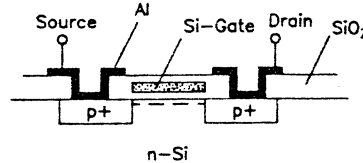


Bild 8: FAMOS-Transistor. Die Gateelektrode ist völlig in SiO₂ eingebettet; eingespeicherte Ladungen können deshalb nicht abfließen.

FAST-Logik
(FAST logic)

Eine Weiterentwicklung der → TTL-Technik. Dabei werden die Bauelemente gegeneinander nicht durch Sperrschichten, sondern durch Oxid isoliert, wodurch die parasitären Kapazitäten verringert und die Bauelemente wesentlich kleiner gemacht werden können. Die Schaltschwelle liegt höher als bei der TTL und auch höher als bei der Schottky-TTL, die Signalverzögerungszeiten sind mit 2,7 ns um ca. 10 % niedriger, die Verlustleistung beträgt nur ein Funftel (4 mW). FAST-Schaltungen sind zu den übrigen TTL-Familien kompatibel.

FET, Feldeffekttransistor
(FET, field effect transistor)

Feldeffekttransistoren sind → unipolare Halbleiterbauelemente mit Steuerung des Stromes in einem leitenden Kanal durch ein elektrisches Feld. Es wird nur eine Art von Ladungsträgern ver-

wendet (n-Typ oder p-Typ). Im Gegensatz zum bipolaren Transistor geschieht die Steuerung strom- und damit leistungslos. Deshalb weisen FETs sehr hohe Eingangswiderstände auf (Gigaohm-Bereich). Zwei Grundformen sind zu unterscheiden: der Sperrschicht-FET (pn-FET) und der Isolierschicht-FET (IGFET, insulated gate FET). FETs haben meistens drei Elektroden: Source (Emitter), Drain (Kollektor) und Gate (Steuerelektrode), es gibt jedoch auch Ausführungen mit zwei Gates (Dual-gate FET). Zwischen Drain und Source wird die Betriebsspannung angelegt, zwischen Gate und Source die Steuerspannung.

Sperrschicht-FETs verwenden einen p- oder n-dotierten Kanal mit metallischen Anschlüssen für Drain und Source, in dem ein n- oder p-dotierter Bereich mit dem metallischen Gate-Anschluß eindiffundiert ist (Bild 9). Liegt keine Gatespannung an, so ist der Kanal leitend, und zwar in beiden Richtungen (ohmsches Verhalten). Durch eine an das Gate angelegte Sperrspannung wird der Stromfluß zwischen Source und Drain mit wachsender Sperrspannung laufend verringert (der Kanal wird eingeschnürt), bis schließlich der Strom Null wird (Abschnür-, pinch-off-Spannung).

Bei Isolierschicht-FETs, zu denen die → MOS-FETs (metal-oxide-semiconductor) und die MNS-FETs (metal-nitride-semiconductor) gehören, allgemein spricht man auch von MIS-FETs (metal-insulator-semiconductor), ist die Gateelektrode durch eine Oxid- oder Nitrid-Schicht vom Kanal isoliert (s. Bild 9). Da bei diesen Transistoren auch der geringe Sperrstrom, den pn-FETs aufweisen, entfällt, haben sie sehr hohe Eingangswiderstände. IG-FETs werden als p- oder n-Kanal-Typen hergestellt, jedoch muß noch zwischen solchen vom Anreicherungs- (enhancement type) und Verarmungstyp (depletion type) unterschieden werden. Anreicherungstypen sind ohne Gatespannung gesperrt und werden mit wachsender Gatespannung leitend. Verarmungstypen sind auch bei Gatespannung Null bereits leitend, die Leitfähigkeit des Kanals nimmt je nach Polarität der Gatespannung entweder zu oder bis zur völligen Sperrung ab.

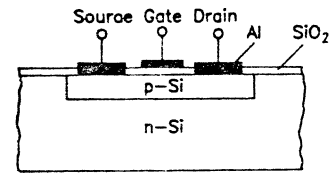
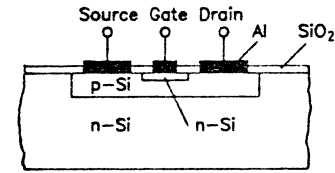


Bild 9: Bau eines planaren pn-FETs und eines planaren MOSFETs im Querschnitt

FIFO-Speicher

(FIFO memory, first in/first out memory)

Der FIFO-Speicher ist ein Schreib-Lesespeicher ohne Adressierung, bei dem die ankommenden Daten der Reihe nach, beginnend mit dem letzten freien Speicherplatz, eingeschrieben und in der gleichen Reihenfolge auch wieder ausgelesen werden, wobei nach jedem ausgelesenen Bit der gesamte Speicherinhalt nachrückt, so daß nur ein Lesevorgang möglich ist. Schreiben

und Lesen kann mit unterschiedlichen Taktfrequenzen geschehen (asynchroner Betrieb). FIFOs werden hauptsächlich als Zwischenspeicher zur Anpassung der Datenrate des Eingangssignals an die Datenrate des Verarbeitungssystems verwendet.

Flatpack, Flachgehäuse
(flatpack)

Quadratische oder rechteckige Gehäuseform für integrierte Schaltungen, meist aus Keramik, mit seitlich herausgeführten Anschlußbahnen. Kleiner, aber teurer, und schwieriger zu montieren als → DIP-Gehäuse.

Flipflop
(flip-flop)

Eine → bistabile Schaltung, die als Zähler, Frequenzteiler, Speicher oder Schieberegisterstufe verwendet werden kann. Die wichtigsten Flipflops sind → D-, → JK-, → RS-, → RST- und → T-Flipflop (s. Bild 10).

Folgeschaltung, Sequentielle Schaltung
(sequential circuit)

Logische Schaltungen, deren Ausgangszustände nicht nur von den augenblicklich anliegenden Eingangssignalen abhängen,

sondern auch von zeitlich vorhergehenden. Hierzu gehören z. B. → Flipflops, → Speicherschaltungen und → Schieberegister.

Fotolack
(photoresist)

Lichtempfindlicher Lack, der als dünne Schicht aufgetragen wird und zur → Maskierung bei der → Fotolithografie dient. Es gibt positive und negative Fotolacke; sie werden je nachdem verwendet, ob nach der Entwicklung die belichteten oder die unbelichteten Bereiche zurückbleiben sollen.

Fotolithografie
(photolithographic process)

Bei der Herstellung von Halbleiterbauelementen wird eine ganze Reihe von Masken benötigt, die bestimmte Bereiche der Oberfläche bei der → Diffusion oder beim Ätzen abdecken. Diese Masken werden fotografisch angefertigt, indem eine dünne (unter 1 µm) → Fotolackschicht auf der Oberfläche des Halbleiterkristalls angebracht ist, die über eine entsprechende Fotovorlage belichtet wird. Nach der Entwicklung bleibt das positive oder negative Muster der Vorlage zurück. Die dann noch verbliebenen und vom Entwicklungsvorgang gehärteten Lackbereiche schützen die Oberfläche gegen das Diffusions- oder Ätzmittel. Es können

Linienbreiten bis etwa 1 µm erreicht werden. Feinere Auflösungen erfordern andere Verfahren, z. B. Belichtung mit Elektronenstrahlen.

Frequenzteiler
(frequency divider)

Digitale Schaltungen, welche die Frequenz einer Eingangsimpulsfolge auf einen ganzzahligen Bruchteil herabsetzen. Sie können beliebig hintereinandergeschaltet werden. Binäre Frequenzteiler teilen durch Potenzen von 2 (1:2, 1:4, 1:8 usw.), Dezimalteiler teilen durch Potenzen von 10 (1:10, 1:100 usw.). Außerdem gibt es programmierbare Teiler, bei denen durch Steuersignale das Teilungsverhältnis gewählt werden kann. Zur Frequenzteilung können → Flipflops verwendet werden.

FROM, Festwertspeicher mit Schmelzbrücken
(FROM, fusible ROM)

Eine spezielle Art des → PROMs, bei dem die Programmierung durch Abschmelzen von dünnen Metallbrücken geschieht. Das Programmieren kann nur einmal vorgenommen werden.

G

GaAs-Digitalschaltung
(GaAs digital circuit)

Ähnlich wie in der Si-Technik verschiedene Logikfamilien (→ TTL, → ECL) hergestellt werden, haben sich auch in der GaAs-Technik bestimmte Schaltungsfamilien ergeben. Verwendet werden in größerem Maßstab zur Zeit die Buffered-FET-Logik (→ BFL), die Unbuffered-FET-Logik (→ UFL) und die Capacitor-Enhanced-Logik (→ CEL), alle mit → DFETs, sowie die Direct-Coupled-FET-Logik (→ DCFL) mit → EFETs. GaAs-Digitalschaltungen haben Signalverarbeitungsgeschwindigkeiten von einigen Gbit/s und Signalanstiegs- und Abfallzeiten von teilweise unter 100 ps. Bild 11 zeigt als Beispiel eine OR/NOR-Schaltung mit fünf Eingängen (Harris HMD 11101).

Galliumarsenid, GaAs
(gallium arsenide, GaAs)

Ein Halbleiter aus der Gruppe der III-V-Verbindungen, bestehend aus Gallium (Gruppe III des Periodischen Systems) und Arsen (Gruppe V), der gegenüber Silizium zwei prinzipielle Vorteile hat; zum einen ist die Elektronenbeweglichkeit wesentlich höher, so daß Bauelemente mit sehr hoher Grenzfrequenz hergestellt werden können, zum anderen ist der undotierte Halbleiter semisierend, weswegen das GaAs-Substrat praktisch als Isolator betrachtet werden kann, in den alle zu integrierenden Bauelemente ohne weitere gegenseitige Isolationsvorkehrungen eingebaut werden können. Aus GaAs werden z. B. → FETs, → Bipolartransistoren, → HEMTs, → monolithisch integrierte Schaltungen, → MMICs und optoelektronische Bauelemente, wie → LEDs und Laserdioden hergestellt.

Gate
(gate)

Steuerelektrode eines → Feldeffekttransistors. Der englische Ausdruck Gate wird außerdem an Stelle von → Verknüpfungsschaltung oder → Gatter benutzt (z. B. UND-Gate, OR Gate).

Gate Array, Gatterfeld
(gate array)

Integrierte Digitalschaltung mit einer großen Anzahl von → Verknüpfungsschaltungen, die nach Kundenangaben (Kundenspezifikation) vom Hersteller zusammengeschaltet werden. Dadurch können kundenspezifizierte umfangreichere Schaltungen, für die sich wegen kleiner Stückzahlen die Entwicklung einer speziellen

integrierten Schaltung nicht lohnt, preisgünstiger hergestellt werden.

Gatter, Gatterschaltung
(gate circuit)

Häufig benutzter Laborausdruck für → Verknüpfungsschaltung.

Geätzte Schaltung
(printed circuit)

Wird häufig als gedruckte Schaltung bezeichnet. Es handelt sich um metallische Leiterbahnstrukturen, die mittels → Fotolithografie durch Ätzen oder elektrolytisches Abtragen auf ein- oder beidseitig metallbeschichteten Trägerplatten erzeugt werden. Die geätzten Schaltungen werden anschließend mit weiteren Bauelementen (Widerstände, Transistoren, integrierte Schaltungen, Stecker usw.) bestückt. Die Bauteile werden entweder von Hand oder maschinell verlötet.

Gleichstromverstärker
(direct current amplifier, DC-amplifier)

Verstärker, dessen einzelne Stufen direkt (gleichstrommäßig, galvanisch) miteinander verbunden sind, der demnach keine untere Grenzfrequenz besitzt. Fast alle → Operationsverstärker sind Gleichstromverstärker. Bei der Schaltungsauslegung müssen besondere Vorkehrungen gegen Betriebsspannungs- Bauelementdaten- und Temperaturänderungen getroffen werden.

Golddotierte TTL-Schaltungen
(gold-doped TTL circuits)

Schaltungen aus der → TTL-Familie, bei denen durch → Dotieren mit Goldatomen die Erholzeit der Transistoren im Sättigungsbereich herabgesetzt wird. Sie sind etwas langsamer als die → Schottky-TTL-Schaltungen, dafür aber in einem größeren Temperaturbereich verwendbar.

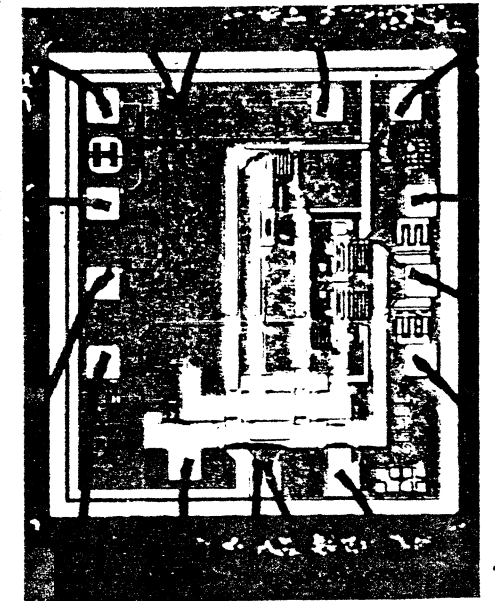
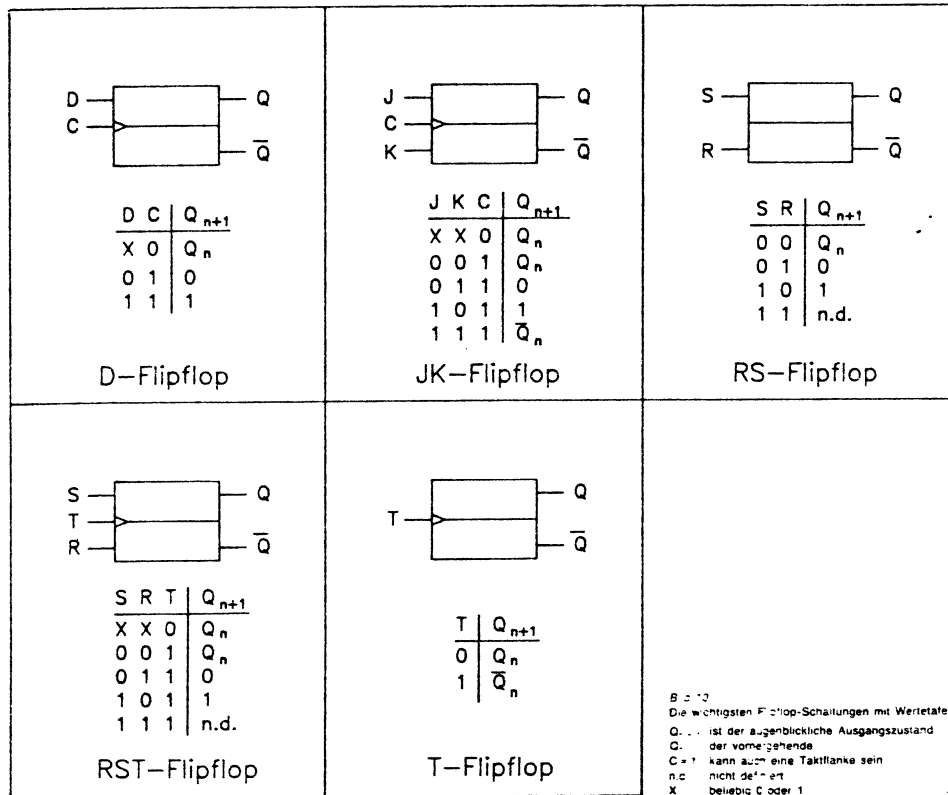


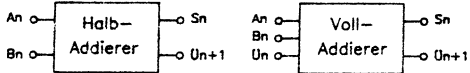
Bild 11: Monolithisch integrierte OR/NOR-Schaltung mit fünf Eingängen in GaAs-Technik (Harris HMD 11101)



B = 10
Die wichtigsten Flipflop-Schaltungen mit Wertetafeln
Q... ist der augenblickliche Ausgangszustand
C... der vorhergehende
C = 1 kann auch eine Taktfanke sein
n.d. nicht definiert
X = beliebig 0 oder 1

Halbaddierer
(half adder)

Die einfachste Form eines binären → Addierers, der zwei einstellige Binärzahlen addiert und Summen- und Übertragungssignal bildet. Er hat zwei Eingänge und zwei Ausgänge (Bild 12). Zwei Halbaddierer können zu einem → Volladdierer zusammengesetzt werden.



An	Bn	Sn	Un+1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Un	An	Bn	Sn	Un+1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Bild 12: Schaltsymbole und Wertetabellen des Halb- und Volladdierers

Halbleiter
(semiconductor)

Material, dessen elektrische Leitfähigkeit zwischen derjenigen von Isolatoren und derjenigen von Metallen liegt. Die Leitfähigkeit ist stark von der Temperatur (sie steigt mit wachsender Temperatur) und von Verunreinigungen abhängig. Die zweite Abhängigkeit benutzt man, um durch gezielte Verunreinigungen (→ Dotieren) die Leitfähigkeit nach Typ und Stärke zu beeinflussen. Die bekanntesten Halbleiter sind Germanium (Ge), Silizium (Si) und Gallium-Arsenid (GaAs). Die Leitfähigkeit von Halbleitern kann teilweise auch durch Licht (Fotohalbleiter), elektrische Felder oder magnetische Felder (Halleffekt) beeinflusst werden.

Halbleiterspeicher
(semiconductor memories)

Schaltungen mit einer Vielzahl von → Speicherzellen, die in einer bestimmten Weise angeordnet (organisiert) sind und von außen adressiert werden können. Die Zahl der Speicherzellen ist meistens eine Potenz von 2, z. B. 256, 512, bei Speicherkapazitäten über 1000 bit wird nur das Vielfache des 1024-bit-Speichers mit einem angehängten k angegeben, z. B. 1 k = 1024, 2 k = 2048, 4 k = 4096, 64 k = 65536 usw. Außer den Speicherzellen enthält ein Halbleiterspeicher noch Hilfsschaltungen, wie Schreib- und Leseverstärker und Adressendecoder.

Die wichtigsten Halbleiterspeicher sind Speicher mit wahlfreiem Zugriff (→ RAM), Festwertspeicher (→ ROM) und programmierbare Festwertspeicher (→ PROM). Sie können als → bipolare Speicher oder in → MOS-Technik ausgeführt sein. Bipolare Speicher sind im allgemeinen schneller als MOS-Speicher, die dafür weniger Leistungsaufnahme und höhere → Packungsdichten haben. Halbleiter drängen von ihrer Speicherkapazität her immer mehr in den Bereich der Massenspeicher (Magnetband, Magnetblasenspeicher, Magnetplatte) vor.*

HEMT
(HEMT, high electron mobility transistor)

Der HEMT besteht aus einer Schichtenfolge von GaAs und GaAlAs (Gallium-Aluminium-Arsenid) und ist ein Transistor mit sehr hoher Grenzfrequenz, hoher Verstärkung und niedrigem Rauschen. Er ist sowohl für → monolithisch integrierte Mikrowellenschaltungen als auch für schnelle → Digitalschaltungen

geeignet. Bild 13 zeigt den Aufbau: Auf einem undotierten und damit semiisolierendem GaAs-Substrat ist zunächst eine sehr dünne (ca. 70 Å) Schicht aus undotiertem GaAlAs aufgebracht. Darüber befindet sich eine Schicht aus n-dotiertem GaAlAs, die im Bereich der Gateelektrode teilweise abgeätzt wird. Unter den metallischen Source- und Drainelektroden ist eine n-dotierte GaAs-Schicht zur Verbesserung der Leitfähigkeit. Die Schichtenfolge wird üblicherweise durch → Molekularstrahlepitaxie erzeugt. Den Stromtransport übernehmen Elektronen, die in der Grenzschicht zwischen GaAlAs und GaAs in Form einer sehr dünnen („zweidimensionales Elektronengas“) Schicht entstehen. Deshalb wird der HEMT auch als TEGFET oder 2DEGFET (two-dimensional electron gas FET) bezeichnet.

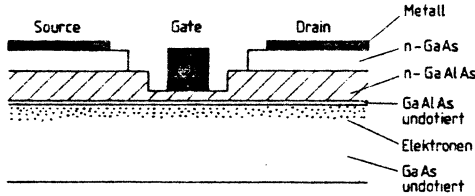


Bild 13: Aufbau eines HEMT im Querschnitt

Hexadezimalsystem
(hexadecimal system)

Zahlensystem, das in der Rechnertechnik häufig benutzt wird. Es baut sich aus Potenzen zur Basis 16 auf. Als Ziffern werden die Dezimalziffern 0 bis 9 und die Buchstaben A bis F verwendet (A ≙ 10, F ≙ 15). Die Hexadezimalzahl A3C z. B. bedeutet $10 \cdot 16^2 + 3 \cdot 16^1 + 12 \cdot 16^0 = 2560 + 48 + 12 = 2620$.

High, H, hoch
(high)

Bezeichnung für denjenigen der beiden logischen Werte oder Wertebereiche einer binären Digitalschaltung, der das höhere Potential aufweist. Im Gegensatz dazu → Low. High kann der logischen 1 (positive Logik) oder der logischen 0 (negative Logik) entsprechen. Bei → TTL-Schaltungen ist High z. B. + 5 V. (Low = 0 V), bei → ECL-Schaltungen -0,9 V (Low = -1,8 V).

NINIL, NNIL, Störsichere Logik
(high noise immunity logic)

Eine → Logikfamilie, die sich durch eine besonders hohe Störsicherheit auszeichnet. Sie wird oft als Eingangsschaltung vor → TTL- und → CMOS-Schaltungen eingesetzt.

HTL
(high threshold logic)

Bezeichnung für Logikschaltungen mit hoher Störschwelle.

Hybridschaltung
(hybrid circuit)

Schaltung, die sich aus verschiedenen Arten von Bauteilen zusammensetzt, speziell als *hybrid integrierte Schaltung*, z. B. eine → monolithisch integrierte Schaltung mit aufgedampften Metallschichtwiderständen oder, und dies umfaßt den größten Teil aller Hybridschaltungen, die Kombination von passiven → Dickfilm- oder → Dünnfilmschaltungen mit Halbleiterbauelementen.

* Siehe auch den Beitrag „Digitale Datenspeicher“, Unterrichtblätter Jg. 37/1984, Nr. 12, S. 438-467 oder Nachdruck Ausgabe Januar 1986.

IC, integrierte Schaltung
(IC, integrated circuit)

Englische Abkürzung für → integrierte Schaltung.

Integrierte Schaltung, IC, IS
(integrated circuit, IC)

Verwirklichung einer elektrischen Schaltung durch Integration, d. h. gemeinsames, gleichzeitiges und untrennbares Herstellen und Zusammenfügen aller Bauelemente einschließlich aller Zwischenverbindungen auf einem gemeinsamen Träger (→ Substrat). Erst die Integration ermöglicht hochgradige Miniaturisierungen. Man unterscheidet → monolithisch und → hybrid integrierte Schaltungen, wobei zumindest die erstgenannten nachträglich weder veränderbar noch reparierbar sind. Im allgemeinen Sprachgebrauch werden integrierte Schaltungen abgekürzt auch als IC oder IS bezeichnet.

I²L, Ionenimplantationslogik
(I²L, ion implantation logic)
(Sprach: „I-Quadrat eI“)

Eine Logikfamilie, hauptsächlich für großintegrierte Schaltungen (→ LSI). Bei der Ionenimplantationstechnologie wird der Halbleiterkristall nicht durch → Diffusion, sondern durch Beschuß mit hochenergetischen ionisierten Dotierungsatomen dotiert. Zur → Maskierung werden Aluminium- oder Siliziumdioxidmasken verwendet. Die Ionen haben nur eine sehr geringe Eindringtiefe (ca. 1 nm), sie kann über die Strahlenergie gesteuert werden. Der Dotierungsgrad des Halbleiters läßt sich sehr genau einstellen.

Inverter, Inverterschaltung, NICHT-Schaltung
(inverter, negation gate, NOT-gate)

Der Inverter ist ein Bauelement, das ein Eingangssignal umkehrt, z. B. aus einer Sinusschwingung die gegenphasige Sinusschwingung erzeugt, wobei im allgemeinen die absoluten Spannungswerte erhalten bleiben. Bei digitalen Schaltungen stellt er die NICHT-Funktion dar, wandelt also 1-Signale in 0-Signale um und umgekehrt. Mathematische Beschreibung: $X = \bar{A}$ (gelesen: X = A NICHT).

Ionenimplantation
(ion implantation)

Technologie zur Herstellung von Halbleiterbauelementen, bei der die → Dotierung durch Beschuß der Halbleiteroberfläche mit hochenergetischen ionisierten Fremdatomen geschieht, → I²L.

IS, Integrierte Schaltung
(IC, integrated circuit)

Abkürzung für integrierte Schaltung.

Isolierung integrierter Schaltungen
(IC isolation)

Da alle Bauelemente bei → integrierten Schaltungen in das gleiche → Substrat eingebettet sind, müssen sie gegeneinander isoliert werden. Dies geschieht entweder durch → pn-Übergänge, die in Sperrichtung gepolt sind, oder durch → dielektrische Isolierschichten, meist SiO₂. Sperrschichtisolierungen haben den Nachteil, daß sie benachbarte Bauelemente kapazitiv miteinander koppeln, die Erdkapazität erhöhen und Leckströme aufweisen. Dielektrische Isolierschichten verringern die Kapazitäten beträchtlich und sind sehr hochohmig, allerdings ist ihre Herstellung schwieriger und teurer.

JK-Flipflop
(J-K flip-flop)

Digitale → Folgeschaltung mit mindestens je einem J-, K- und C- (Takt-, Clock-)Eingang und häufig zwei komplementären Ausgängen. Das Taktsignal ruft je nach Kombination der beiden Datensignale an J und K eine Änderung des Ausgangszustandes hervor (s. Bild 5). Haben J und K 0-Stellung, so tritt keine Änderung am Ausgang auf. Haben J und K 1-Signale, so wird der Ausgangszustand invertiert (Frequenzteiler): 1 an J und 0 an K bewirkt 1 am Ausgang, 0 an J und 1 an K bewirkt 0 am Ausgang. JK-Flipflops werden als Frequenzteiler, Zähler, Speicher und als Schieberegisterstufen verwendet.

Kundenspezifizierte integrierte Schaltung
(custom integrated circuit)

Umfangreiche integrierte Schaltung, z. B. nur mit Verknüpfungsschaltungen (→ Gate array), aber auch mit → Folgeschaltungen, bei der die einzelnen Schaltungseinheiten noch nicht untereinander verbunden sind. Die letzte → Maskierung für die metallischen Zwischenverbindungen geschieht erst nach Angaben des Anwenders, der so auch kleine Serien einer speziellen Schaltung zu wirtschaftlichen Preisen erhält. Es kann und wird dabei oft ein Teil der auf dem → Chip vorhandenen Schaltungseinheiten ungenutzt bleiben.

LCD, Flüssigkristallanzeige
(LCD, liquid crystal display)

Eine Art der → digitalen Anzeigen, die Flüssigkristalle als passives Anzeigemedium verwenden. Der Flüssigkristall befindet sich zwischen den beiden Elektroden eines Plattenkondensators und ist transparent (durchsichtig), solange kein elektrisches Feld auf ihn einwirkt. Unter Feldeinwirkung wird er mehr oder weniger undurchsichtig. Flüssigkristallanzeigen sind passive Anzeigen, da sie selbst kein Licht aussenden. Der Kontrast hängt von den äußeren Beleuchtungsverhältnissen ab. Flüssigkristallanzeigen haben aber gegenüber → LED-Anzeigen den Vorteil eines äußerst geringen Leistungsbedarfs; deshalb werden sie hauptsächlich in batteriebetriebenen Geräten (Digitaluhren, Taschenrechner) verwendet.

LED-Anzeigen
(light emitting diode displays)

Eine Art der → digitalen Anzeigen, die Leuchtdioden als aktives Anzeigemedium verwenden. Leucht- oder Lumineszenzdioden sind Gallium-Arsenid- oder Gallium-Phosphiddioden, die beim Durchfließen eines Stromes Licht bestimmter Wellenlänge aussenden. Sie haben deshalb stets farbiges Licht (rot, gelb, grün) und benötigen je nach Größe Stromstärken von einigen mA bis zu einigen 10 mA. Da die Flußspannung etwa 1 V beträgt, treten (im Gegensatz zu den → LCD-Anzeigen) ziemlich hohe Verlustleistungen auf. Ihr Vorteil liegt in dem hellen und gut erkennbaren Anzeigebild.

Logik
(logic)

Im Sprachgebrauch als Abkürzung für → logische Digitalfamilie gebraucht, z. B. TTL-Logik, ECL-Logik.

Logische Digitalfamilie
(digital logic types)

Gesamtheit aller → logischen Schaltungen, die auf einer gemeinsamen Grundschaltung beruhen und miteinander beliebig kom-

binierbar sind, wobei gleiche Betriebsspannung, gleicher → logischer Hub und meist auch vergleichbare Schaltgeschwindigkeiten gegeben sind. Eine bestimmte Logikfamilie kann zu einer anderen kompatibel sein. Verbreitete Logikfamilien sind: 74xx-TTL, 4xxx-CMOS, 10xxx-ECL (die x stehen für fortlaufende Typenbezeichnungen).

Logische Schaltung
(logic circuit)

Elektrische Schaltung zur Darstellung einer logischen Operation, z. B. → UND-Schaltung Kennzeichnend ist, daß der Zusammenhang zwischen Eingangssignalen und Ausgangssignal eindeutig festgelegt ist und in Form einer → Wertetabelle dargestellt werden kann.

Logische Variable
(logic variable)

Rechengröße der → Booleschen Algebra zur mathematischen Beschreibung logischer Zusammenhänge. Sie kann zwei feste Werte 1 und 0 annehmen, wobei 1 wahr und 0 falsch bedeutet. Logische Variable (Veränderliche) werden als Großbuchstaben geschrieben, z. B. $X = A \wedge B$ (gelesen: X gleich A UND B), dabei ist X die Ausgangsvariable, A und B sind die beiden Eingangsvariablen. Im Falle zweier Eingangsvariablen sind demnach 4 verschiedene Kombinationen möglich: $A = 0$ UND $B = 0$ (ergibt $X = 0$), $A = 1$ UND $B = 0$ ($X = 0$), $A = 0$ UND $B = 1$ ($X = 0$), $A = 1$ UND $B = 1$ ($X = 1$). Für jede dieser Kombinationen gibt es einen zugeordneten eindeutigen Wert für die Ausgangsvariable. Dieser Zusammenhang kann der besseren Übersichtlichkeit wegen in Form einer → Wertetabelle angegeben werden.

Logischer Hub
(logic swing)

Spannungsdifferenz zwischen den beiden logischen Signalzuständen → High und → Low einer Digitalschaltung. Da High und Low stets Spannungsbereiche sind, werden zur Berechnung des Hubs Mittelwerte angenommen.

Low, L, niedrig
(low)

Bezeichnung für denjenigen der beiden logischen Werte oder Wertebereiche einer binären Digitalschaltung, der das niedrigere Potential aufweist. Im Gegensatz dazu → High. Low kann der logischen 0 (positive Logik) oder der logischen 1 (negative Logik) entsprechen. Bei → TTL-Schaltungen ist Low z. B. 0 V (High = +5 V), bei → ECL-Schaltungen -1,8 V (High = -0,9 V).

Low-power-Schottky-TTL, LPSTTL
(low power Schottky TTL)

Sonderform der → Schottky-TTL-Schaltungen mit besonders niedriger Leistungsaufnahme (typisch 2 mW je Verknüpfungsschaltung) und trotzdem hoher Schaltgeschwindigkeit (typisch 8 ns).

LSI, Großintegration
(LSI, large scale integration)

Kennzeichnung für → monolithisch integrierte Schaltungen mit mehr als 1000 Grundschaltungen pro → Chip. Zur Unterscheidung → SSI, → MSI und → VLSI.

M

Maskenherstellung im Elektronenstrahlverfahren
(electron beam microfabrication)

Für extrem feine Strukturen (unter 1 µm) kann die herkömmliche → Fotolithografie nicht mehr verwendet werden, da selbst UV-

Licht zu langwellig ist. Es wird deshalb an Stelle des Lichtstrahls ein Elektronenstrahl verwendet, der einen dünnen polymeren Film auf der Halbleiteroberfläche belichtet. Die Strukturen werden durch rechnergesteuerte Bewegungen des Elektronenstrahls erzeugt.

Maskierung
(masking)

Herstellung von Masken, z. B. im → Fotolithografieverfahren für Diffusions- oder Ätzstrukturen. Als Maskenmaterial wird → Fotolack, Siliziumdioxid und teilweise auch ein aufgedampfter Metallfilm verwendet.

Master/Slave-Schaltung
(master slave circuit)

Logische → Folgeschaltung, die aus zwei hintereinandergeschalteten → Flipflops besteht. Eine Information wird z. B. mit der positiven Flanke des Taktimpulses in die erste Schaltung (Master) eingeschrieben, ohne daß sich das Ausgangssignal der zweiten Schaltung (Slave) ändert. Erst mit der negativen Flanke des Taktsignals erscheint die Information am Ausgang. Master/Slave-Schaltungen werden als → D- und → JK-Flipflops angeboten.

MBE
(MBE)

→ Molekularstrahlepitaxie

MESFET, Metall-Halbleiter-FET
(MESFET, metal-semiconductor FET)

Ein → Feldeffekttransistor, bei dem das Metall der Gateelektrode mit dem Halbleitermaterial des Drain-Source-Kanals einen Schottkyübergang bildet. Der MESFET arbeitet wie ein → PN-FET, ist aber einfacher herzustellen und wird sehr häufig in integrierten GaAs-Schaltungen verwendet.

MIC, integrierte Mikrowellenschaltung
(MIC, microwave integrated circuit)

Eine → Hybridschaltung, meist in → Dünnschichttechnik für den Mikrowellenbereich. Sie besteht im wesentlichen aus Streifenleitungen und Halbleiter- → Chips. Widerstände und Kondensatoren werden als Dünnschichtbauelemente, teilweise ebenfalls in Chipform verwendet (B. 14).

Mikrocomputer
(microcomputer)

Allgemeine Bezeichnung für ein miniaturisiertes Computersystem aus integrierten Schaltungen, bestehend aus → Mikroprozessor, Speicher und Ein- und Ausgabeschaltungen, einschließlich der Betriebssoftware**.

Mikrominiaturisierung
(microminimization)

Verringerung der Bauteilabmessungen einer elektrischen Schaltung auf äußerst kleine Werte durch → monolithische Integration. Dadurch können mehrere tausend Bauelemente auf eine Fläche von 1 mm² konzentriert werden, → LSI, → VLSI.

* polymer, aus großen Molekülen bestehend (Ausdruck aus der organischen Chemie).

** Software: Gesamtheit der Programme für eine Datenverarbeitungsanlage, im Gegensatz dazu Hardware: Gesamtheit der Geräte einer Datenverarbeitungsanlage.

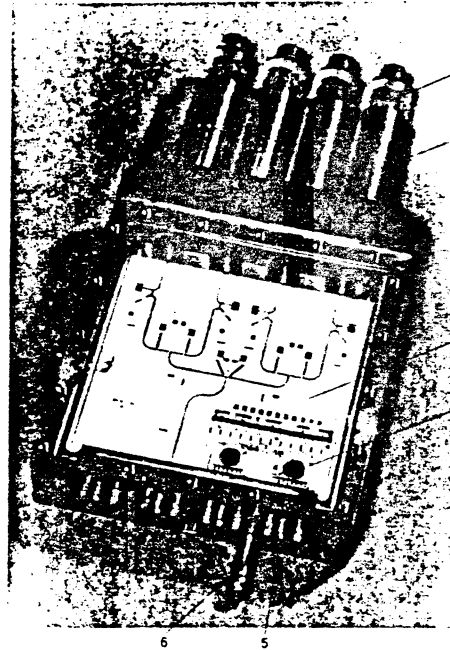


Bild 14: Integrierte Mikrowellenschaltung
Dünnschichtschaltung (Substratgröße 7,5 cm x 7,5 cm) für den 10-GHz-Bereich zur Antennensteuerung mit zusätzlich aufgebrachtener Dickfilmschaltung

- 1 Strahler
- 2 Hohlleiter
- 3 Keramiksubstrat mit Dünnschichtschaltung
- 4 Keramiksubstrat mit Siebdruckschaltung
- 5 Koaxialbuchse
- 6 Durchführung für Steuerleitungen

Mikroprozessor
(microprocessor)

Zentrales Bauelement eines → Mikrocomputers, meist auf einem einzigen → Chip konzentriert, das aus der arithmetisch-logischen Einheit (zur Durchführung von Rechenoperationen und logischen Entscheidungen), dem Steuerblock und dem Registerfeld besteht. Mikroprozessoren können auch für sich betrieben werden und ersetzen dann fest verdrahtete Logikschaltungen.

MISFET
(MISFET, metal-insulator-semiconductor FET)

Allgemein ein → Feldeffekttransistor, bei dem die metallische Gateelektrode durch einen Isolator vom Halbleitermaterial des Drain-Source-Kanals getrennt ist. Hierzu gehören der → MOSFET und der → MNSFET.

MMIC
(MMIC, microwave monolithic integrated circuit)

→ monolithisch integrierte Mikrowellenschaltung.

MNOS, Metallnitrid-FET, MNS
(MNOS, metal-nitride silicon device)

Ein → Feldeffekttransistor, dessen → Gateelektrode zusätzlich zu einer dünnen SiO₂-Schicht noch eine Si₃N₄-Schicht (Silizium-

nitrid) als Isolator zum Drain-Source-Kanal aufweist. Dadurch wird eine bessere Oberflächen- → Passivierung und eine höhere Durchschlagfestigkeit erreicht.

MNSFET
(MNSFET, metal-nitride-semiconductor FET)

Ein → Feldeffekttransistor, bei dem die metallische Gateelektrode durch eine Nitrid-Schicht vom Halbleitermaterial des Drain-Source-Kanals getrennt ist.

Molekularstrahlepitaxie, MBE
(molecular beam epitaxy, MBE)

Verfahren zum Herstellen epitaktischer Schichten, in dem ein gerichteter Strahl nichtionisierter Moleküle von einer Quelle auf Grund ihrer thermischen Energie auf ein Substrat (im allgemeinen ein Halbleiterkristall) gerichtet wird. Mit der MBE lassen sich extrem dünne und sehr genau dotierte Schichten in beliebiger Reihenfolge erzeugen.

Monolithisch integrierte GaAs-Schaltung
(monolithic integrated GaAs-circuit)

Der Halbleiter → GaAs wird hauptsächlich für schnelle → GaAs-Digitalschaltungen, → MMICs und → monolithisch integrierte Millimeterwellenschaltungen verwendet. Da das uncutierte GaAs Isolatoreigenschaften hat, können alle Bauelemente ohne Zwischenisolierungen nebeneinander angeordnet werden, und auch die leitenden Zwischenverbindungen können direkt auf die Substratoberfläche aufgebracht werden. Transistoren werden meistens als → MESFETs, Dioden als → Schottkydioden ausgeführt. Da GaAs für elektrooptische Bauelemente (Laser, LED, Fotodiode) geeignet ist, werden auch elektro-optische Integrationen, wie Treiberverstärker mit Laser oder Fotodiode mit Vorverstärker, realisiert.

Monolithisch integrierte Mikrowellenschaltung, MMIC
(microwave monolithic integrated circuit, MMIC)

Monolithisch integrierte Schaltung für den Mikrowellenbereich größtenteils auf GaAs-, teilweise auch auf Siliziumbasis hergestellt, z. B. Schmal- und Breitbandverstärker, Mischer, Phasenschieber, Umschalter, Oszillatoren, Filterschaltungen oder Kombinationen solcher Baugruppen.

Monolithisch integrierte Millimeterwellenschaltung
(mm-wave monolithic integrated circuit)

Monolithisch integrierte Schaltung für den Millimeterwellenbereich, im allgemeinen auf GaAs-Basis

Monolithisch integrierte Schaltung
(monolithic integrated circuit)

Hauptform der heute gebräuchlichen → integrierten Schaltungen, bei der alle Bauelemente in einen Halbleiterkristall (Silizium, Gallium-Arsenid) durch → Diffusion oder → Ionenimplantation eingebaut sind. Alle aktiven (Dioden, Transistoren) und passiven (Widerstände, teilweise Leitungsverbindungen und Leitungskreuzungen, in geringem Maße auch Kondensatoren) Bauelemente werden in den Kristall eingebaut, die meisten metallischen Zwischenverbindungen und Anschlußflächen werden auf der Kristalloberfläche angebracht.

Monostabile Schaltung, monostabiler Multivibrator
(monostable circuit, single-shot circuit)

Eine → Digitalschaltung mit einem stabilen und einem nichtstabilen Ausgangszustand. Ein Eingangssignal bewirkt die Umschaltung des Ausgangs vom stabilen in den nichtstabilen Zustand.

Die Schaltung kehrt nach einer bestimmten Zeit, die durch die Größe eines RC-Gliedes vorgegeben ist, selbsttätig in den stabilen Zustand zurück. Monostabile Schaltungen werden z. B. benutzt, um aus einem kurzen Eingangs- (Trigger-) Impuls einen Impuls mit festgelegter Dauer zu machen.

MOS, Metalloxid-Halbleiter

(MOS, metal oxide semiconductor)

Ein unipolares Halbleiterbauelement mit einem leitfähigen Kanal zwischen zwei metallischen Elektroden (Drain und Source), dessen Leitfähigkeit von einem elektrischen Feld beeinflusst wird. Dieses Feld entsteht durch Anlegen einer Spannung an die metallische dritte Elektrode (Gate), die vom Kanal durch eine Isolierschicht aus Siliziumdioxid (SiO_2) getrennt ist (s. Bild 9).

MOSFET, Metalloxid-Feldeffekttransistor

(MOSFET, metal oxide semiconductor field effect transistor)

Ein spezieller Feldeffekttransistor mit MOS-Struktur. MOSFETs werden mit p- oder n-leitendem Kanal hergestellt, und zwar jeweils als Anreicherungs- (enhancement type) oder Verarmungstyp (depletion type), so daß insgesamt vier unterschiedliche Transistorarten vorliegen.

MSI, mittlere Integrationsdichte

(MSI medium scale integration)

Kennzeichnung für monolithisch integrierte Schaltungen mit rund 100 bis 1000 Grundsaltungen pro Chip. Zur Unterscheidung → SSI, → LSI, → VLSI.

Multichip-Schaltung

(multichip integrated circuit)

Zusammenschaltung mehrerer Einzel-Chips in einem gemeinsamen Gehäuse. Die Chips werden dabei entweder direkt miteinander und mit den Anschlußstiften des Gehäuses verbunden, oder sie befinden sich auf einer Siebdruck- oder Dünnfilmschaltung.

N

NICHT-Schaltung, Inverter

(NOT gate, negation gate)

Logische → Digitalisaltung, welche die NICHT-Funktion darstellt. Sie bildet als Ausgangssignal stets den Komplementärwert des Eingangssignals, wandelt also 1-Signale in 0-Signale um und umgekehrt. Mathematische Beschreibung: $X = \bar{A}$ (gelesen: $X = A$ NICHT).

NICHT-ODER-Schaltung, NOR-Schaltung

(NOR gate, NOT OR gate)

Logische → Digitalisaltung, welche die NICHT-ODER-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 0, wenn mindestens eines, gleichgültig welches, der Eingangssignale den Wert 1 hat (s. Bild 13). Mathematische Beschreibung: $X = \bar{A} \vee \bar{B}$ (gelesen: $X = A$ ODER B , NICHT). Die NICHT-ODER-Schaltung ist eine Grundsaltung, mit deren Hilfe jede andere logische Schaltung aufgebaut werden kann.

NICHT-UND-Schaltung, NAND-Schaltung

(NAND gate, NOT AND gate)

Logische → Digitalisaltung, welche die NICHT-UND-Funktion darstellt. Ihr Ausgangssignal hat nur dann den Wert 0, wenn Eingangssignal A UND Eingangssignal B den Wert 1 haben (s. Bild 16). Mathematische Beschreibung: $X = A \wedge B$ (gelesen: $X = A$ UND B , NICHT). Die NICHT-UND-Schaltung ist eine Grundsaltung, mit deren Hilfe jede andere logische Schaltung aufgebaut werden kann.

NMOS, n-Kanal-MOS

(NMOS)

Bezeichnung für Schaltungen oder Schaltungsfamilien, die mit n-Kanal-MOS-Bauelementen aufgebaut sind. Dies sind Bauelemente, bei denen der Drain-Source-Kanal n-dotiert ist, also Elektronenüberschuß aufweist. NMOS ist geeignet für VLSI-Schaltungen und hat günstigere elektrische Eigenschaften (Leistungsaufnahme, Schaltgeschwindigkeit) als die PMOS-Technik.

ODER-Schaltung, OR-Schaltung

(OR gate)

Logische → Digitalisaltung, welche die ODER-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn mindestens eines, gleichgültig welches, der Eingangssignale den Wert 1 hat (s. Bild 16). Mathematische Beschreibung: $X = A \vee B$ (gelesen: $X = A$ ODER B).

ODER-NICHT-Schaltung, Inklusionsschaltung

(OR NOT gate, inclusion gate)

Logische → Digitalisaltung, welche die ODER-NICHT-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn entweder das Eingangssignal A den Wert 1 oder das Eingangssignal B den Wert 0 hat oder wenn beides erfüllt ist (s. Bild 16). Mathematische Beschreibung: $X = A \vee \bar{B}$ (gelesen: $X = A$ ODER B -NICHT).

Operationsverstärker, Rechenverstärker

(operational amplifier)

Fast immer ein Gleichstromverstärker, der ursprünglich in Analogrechnern zur Durchführung von Rechenvorgängen (Addition, Multiplikation, Integration, Differentiation) verwendet wurde, heute, besonders als monolithisch integrierter Operationsverstärker, das vielseitigste Analogbauelement überhaupt geworden ist. Operationsverstärker arbeiten nach dem Prinzip des Differenzverstärkers, bestehen aber aus mehreren hintereinandergeschalteten Verstärkerstufen. Die Ausgangsspannung ist gleich der Differenz der Eingangsspannungen, multipliziert mit dem Verstärkerfaktor. Gleich große Eingangsspannungen haben demgegenüber praktisch keinen Einfluß auf die Ausgangsspannung. Operationsverstärker haben Verstärkungsfaktoren bis zu 100 000, hohe Eingangswiderstände (besonders, wenn als Eingangstransistoren → FETs verwendet werden) und einen niedrigen Ausgangswiderstand. Die Verstärkung ist frequenzabhängig, sie ist von 0 Hz bis zur oberen Grenzfrequenz (einige kHz bis einige MHz) konstant und fällt dann mit etwa 6 dB/Oktave ab.

Operationsverstärker werden üblicherweise mit Gegenkopplung betrieben, wodurch die Leerlaufverstärkung (Verstärkung ohne Gegenkopplung) auf eine bestimmte Schleifenverstärkung (Verstärkung mit Gegenkopplung) herabgesetzt wird. Mit abnehmender Schleifenverstärkung nimmt die Bandbreite des Verstärkers zu (das Produkt aus Verstärkung und Bandbreite ist konstant). Wird die Gegenkopplung frequenzabhängig gemacht, erhält man Filterschaltungen. Bei positiver Gegenkopplung (Rückkopplung) können Operationsverstärker als freischwingende Generatoren (Sinus, Rechteck, Dreieck, Sägezahn) verwendet werden.

Optokoppler

(optical coupler, optical insulator)

Optokoppler bestehen aus einer LED und einem Phototransistor (Transistor, der nicht über den Basisstrom aufgrund einer

* Oktave: Frequenzbereich oder Abstand zwischen einer unteren Frequenz und dem doppelten Wert dieser Frequenz.

angelegten Spannung, sondern durch Licht angesteuert wird) Ein Steuerstrom durch die LED läßt diese Licht emittieren, das den Transistor beleuchtet und in diesem einen um den Faktor 100 bis 200 höheren Kollektor-Emitterstrom hervorruft. Ein- und Ausgangsstrom sind wegen der nichtleitenden Lichtstrecke galvanisch voneinander isoliert (bis zu 5 kV Durchschlagfestigkeit bei Dual-in-Line-Gehäusen). Optokoppler können sowohl zur analogen, als auch zur digitalen Signalübertragung verwendet werden, im letzteren Fall sind Datenraten bis zu einigen 10 Mbit/s möglich.

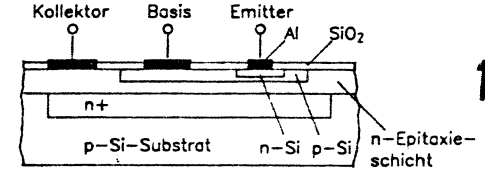


Bild 15 Aufbau eines planaren n-Epitaxialtransistors im Querschnitt

PLL, phasengeregelte Schaltung

(PLL, phase-locked loop circuit)

Schaltung zur Erzeugung einer Sinus- oder Rechteckschwingung, deren Frequenz und Phase auf ein Referenzsignal synchronisiert wird, dessen Frequenz gleich, ein ganzzahliges Vielfaches oder ein ganzzahliger Bruchteil ist. Eine PLL-Schaltung besteht aus einer Phasenvergleichsschaltung, einem Verstärker, einem Tiefpaßfilter, einem spannungsgesteuerten Oszillator (VCO, voltage controlled oscillator) und oft auch aus Frequenzteilerstufen. Sobald Frequenz oder Phase der erzeugten Schwingung mit dem Referenzsignal nicht mehr übereinstimmen, gibt die Phasenvergleichsschaltung eine Korrekturspannung ab, die den Oszillator in Frequenz und/oder Phase entsprechend nachregelt.

PMOS, p-Kanal-MOS

(PMOS)

Die Bauelemente einer integrierten Schaltung sind zwar gegenüber dem Substrat isoliert (durch gesperrte PN-Übergänge oder dielektrische Isolation), aber doch so dicht benachbart, daß zwischen ihnen und dem Substrat unerwünschte (parasitäre) Kapazitäten auftreten, so daß die elektrischen Eigenschaften (obere Grenzfrequenz, Schaltzeiten) negativ beeinflusst werden. Sie sind nicht zu vermeiden, können aber durch eine dielektrische Isolation stark herabgesetzt werden.

PN-Übergang

(p-n junction)

Die Grenze zwischen einem p- und einem n-dotierten Halbleiter, Grundlage für alle bipolaren Halbleiterbauelemente (Dioden, Transistoren, integrierte Schaltungen). Der pn-Übergang hat Gleichrichterverhalten. In Sperrrichtung gepolt ist er hochohmig (es fließt nur ein vernachlässigbar kleiner Reststrom), in Durchlaßrichtung gepolt ist er nach Überschreiten einer Schwellenspannung (Germanium etwa 0,3 V, Silizium etwa 0,7 V) niederohmig. In beiden Richtungen ist eine starke Temperaturabhängigkeit zu beobachten.

PROM, programmierbarer Festwertspeicher

(PROM, programmable read only memory)

Ein Festwertspeicher (→ ROM), der vom Anwender programmiert werden kann. Es gibt PROMs, bei denen nur eine einmalige Programmierung möglich ist, und solche, die beliebig oft neu programmiert werden können (→ EAROM, → EEROM, → EPROM).

Programmierbare Logik

(programmable logic)

Anordnungen von Logikschaltungen oder/und sequentiellen Schaltungen, die vom Anwender nach Maßgabe seiner jeweiligen Systemforderungen programmiert werden können. Hierzu ge-

Packungsdichte

(packaging density)

Gibt die Zahl der Bauelemente pro Volumeneinheit (Raumeinheit) einer integrierten Schaltung an. Je höher die Packungsdichte, um so höher der Miniaturisierungsgrad. Heute werden Packungsdichten von vielen tausend Bauelementen pro mm^3 erreicht. Mit wachsender Packungsdichte muß jedoch gleichzeitig dafür gesorgt werden, daß die Verlustleistung pro Bauelement entsprechend verringert wird, damit die entstehende Verlustwärme nach außen abgeführt werden kann und nicht den Kristall unzulässig aufheizt.

Parasitäre Kapazität

(parasitic capacitance)

Die Bauelemente einer integrierten Schaltung sind zwar gegenüber dem Substrat isoliert (durch gesperrte PN-Übergänge oder dielektrische Isolation), aber doch so dicht benachbart, daß zwischen ihnen und dem Substrat unerwünschte (parasitäre) Kapazitäten auftreten, so daß die elektrischen Eigenschaften (obere Grenzfrequenz, Schaltzeiten) negativ beeinflusst werden. Sie sind nicht zu vermeiden, können aber durch eine dielektrische Isolation stark herabgesetzt werden.

Passivierung, Oberflächenpassivierung

(surface passivation)

Letzter Schritt bei der Herstellung eines Transistors oder einer monolithisch integrierten Schaltung vor dem Aufdampfen der Metallisierung für die Anschlußflächen und Zwischenverbindungen. Auf die Halbleiteroberfläche wird dazu eine Siliziumdioxid- oder Siliziumnitridschicht aufgebracht; sie dient einerseits als Isolator zwischen Bauelementen und Metallisierung, andererseits als Schutz besonders der offenliegenden PN-Übergänge gegen Feuchtigkeit und atmosphärische Verunreinigungen.

PLA, programmierbares logisches Feld

(PLA, programmable logic array)

Eine Anordnung von UND- und ODER-Schaltungen in Form einer Matrix. Die einzelnen Bausteine können entweder vom Hersteller nach Kundenangaben durch eine entsprechende Metallisierung verdrahtet werden, oder der Kunde kann sie selbst programmieren, indem er bestimmte Verbindungen, die in Form von Schmelzbrücken angebracht sind, durchtrennt (ähnlich wie beim → FROM).

Planarprozess

(planar process)

Heute allgemein verwendeter Herstellungsprozess für Transistoren, Dioden und monolithisch integrierte Schaltungen, bei dem die Bauelemente nur in einer dünnen Schicht (einige μm) unter der Kristalloberfläche eingebaut werden und alle Anschlüsse auf der Oberfläche liegen (Bild 15). Der Rest des (einige 100 μm) dicken Kristalls dient als Substrat.

* Matrix: Zeilen- und spaltenweise Anordnung von Buchstaben, Zahlen oder Bauelementen.

hören → PLA, → RAM, → ROM, → PROM und → Mikroprozessor. Im Gegensatz hierzu die festverdrahtete Logik, bei der die Systemforderungen durch eine eigens entworfene und nachträglich nicht mehr wesentlich veränderbare Hardware verwirklicht werden.

R

RAM, Speicher mit wahlfreiem Zugriff, Schreib-/Lese-Speicher (RAM, random access memory)

Speicher, bei dem auf jedes Speicherelement durch eine Adressierung jederzeit zugegriffen werden kann, und zwar sowohl zum Lesen als auch zum Schreiben. Es besteht aus einer bestimmten Zahl von Speicherzellen, die meistens wortweise organisiert sind, den Schreib- und Leseverstärkern und einem oder mehreren Adressendecodern. RAMs werden als Arbeitsspeicher in Datenverarbeitungs-systemen verwendet.

RC-TL, Widerstand-Kondensator-Transistor-Logik (RC-TL, resistor-capacitor-transistor-logic)

Veraltete → Logikfamilie, ähnlich der → RTL, mit zusätzlichen Koppelkondensatoren zur Erhöhung der Schaltgeschwindigkeit (s. Bild 2).

Register (register)

Eine lineare Anordnung von Speicherbauelementen (z. B. → D-Flipflops, → JK-Flipflops) mit gemeinsamer Taktansteuerung, die entweder parallel oder seriell betrieben wird und z. B. ein Datenwort zwischenspeichert (Speicherregister) oder parallel-serienwandelt (→ Schieberegister).

Ringzähler (ring counter)

Ein vom Ausgang auf den Eingang zurückgekoppeltes → Schieberegister, bei dem eine feste Information immer im Kreis herumgeschoben wird. Mit jedem Taktimpuls ändert die Information ihre Position dabei innerhalb der Schaltung um eine Stelle. Der Ringzähler kann zur Frequenzteilung oder zur Impulzzählung verwendet werden.

Ripple-Zähler (ripple counter)

Hintereinanderschaltung von → Frequenzteilerstufen, wobei das Ausgangssignal der ersten Stufe die zweite Stufe ansteuert, das der zweiten Stufe die dritte usw. Wird zur Frequenzteilung und Impulzzählung verwendet und auch als Asynchronzähler oder Asynchronzähler bezeichnet, weil Ausgangssignale der einzelnen Stufen nicht zeitgleich, sondern wegen der → Verzögerungszeiten der Stufen zeitlich gegeneinander versetzt erscheinen.

ROM, Festwertspeicher, Nur-Lese-Speicher (ROM, read only memory)

Im Gegensatz zum → RAM kann beim ROM der Speicherinhalt nicht geändert werden. Der bei der Herstellung fest einprogrammierte Speicherinhalt kann nur beliebig oft ausgelesen werden.

RS-Flipflop (R-S flip-flop)

Grundbaustein aller → Flipflops, bestehend z. B. aus zwei Transistoren in Emitterschaltung oder aus zwei → NICHT-UND-Schaltungen (s. Bild 10) mit zwei Eingängen S (set: setzen) und R (reset: zurücksetzen, löschen) und meist zwei komplementären Ausgängen. Ein 1-Signal am S-Eingang (bei gleichzeitig 0 an R) ruft ein 1-Signal am Ausgang hervor, ein 1-Signal am R-Eingang (bei gleichzeitig 0 an S) ruft ein 0-Signal am Ausgang hervor. Ist

S = 0 und R = 0, so bleibt der Ausgangszustand unverändert (Speicherung der zuletzt eingegebenen Information). 1-Signal sowohl an S als auch an R stellt einen verbotenen Zustand dar, da in diesem Fall kein eindeutiges Ausgangssignal zugeordnet ist. Das RS-Flipflop ist eine → bistabile Schaltung, es wird auch als RS-Flipflop (gelesen: R-NICHT-S-NICHT) hergestellt; hier sind die Eingänge invertiert, d. h. ein 0-Signal an S (bei gleichzeitig 1 an R) ergibt am Ausgang 1, ein 0-Signal an R (bei gleichzeitig 1 an S) ergibt 0. S = 0 und R = 0 ist der verbotene Zustand.

RST-Flipflop (R-S-T flip-flop)

Ein → RS-Flipflop mit einem dritten Eingang T. Die Eingangssignale an R und S haben nur dann Einfluss auf den Ausgang, wenn ein Signal an T anliegt. R und S wirken als Vorbereitungseingänge, T als Takteingang (s. Bild 10). Nicht zu verwechseln mit dem → T-Flipflop.

RTL, Widerstand-Transistor-Logik (RTL, resistor-transistor-logic)

Veraltete → Logikfamilie, die nur aus Transistoren und Widerständen besteht (s. Bild 3). Die logische Verknüpfung wird in Widerstandsnetzwerken durchgeführt, der Transistor dient zur Verstärkung und als → Inverter.

S

Saphir (sapphire)

Ein Edelstein, der als → Substrat für hochwertige integrierte Schaltungen verwendet wird. → SOS-Technik.

Schaltungen mit diskreten Bauelementen (discrete component circuit)

Schaltungen, die im Gegensatz zu → integrierten Schaltungen aus einzelnen (diskreten) Bauelementen (Widerstände, Transistoren, Kondensatoren usw.) aufgebaut werden.

Schieberegister (shift register)

Schaltung aus z. B. → D-Flipflops oder → JK-Flipflops, die gemeinsam getaktet werden und bei der die Information einer Stufe an die benachbarte Stufe übergeben wird. Die Information kann parallel oder seriell ein- und ausgelesen werden, sie kann von links nach rechts oder von rechts nach links durch das Schieberegister geschoben werden. Schieberegister können zur Serien-Parallel- oder Parallel-Serien-Wandlung verwendet werden.

Schmitt-Trigger (Schmitt trigger)

Schaltung aus zwei Transistoren mit interner Rückkopplung, die nur zwei bestimmte Ausgangsspannungen (z. B. 0 V und +5 V) hat und von einer Eingangsspannung angesteuert wird. Je nach Höhe der Eingangsspannung tritt entweder die eine oder die andere Ausgangsspannung auf, wobei die Umschaltung von der niedrigen zur hohen Ausgangsspannung bei einer anderen Eingangsspannung auftritt als die Umschaltung von der hohen zur niedrigen (Schalthysterese). Schmitt-Trigger werden zur Impulsformung oder Impulsregenerierung verwendet, z. B. um ein Sinussignal in ein Rechtecksignal oder einen verwachsenen Impuls in einen Impuls mit steilen Flanken umzuformen.

Schottky-Diode (Schottky diode)

Halbleiterbauelement mit Diodencharakteristik, auch Hot carrier-Diode (Diode mit hoher Ladungsträgerbeweglichkeit) ge-

nannt, das nicht aus einem → PN-Übergang, sondern aus einem Metall-Halbleiterübergang besteht. Der Ladungstransport geschieht nur durch einen Ladungsträgertyp (Elektronen oder Löcher). Schottky-Dioden haben sehr kurze Schaltzeiten und wesentlich geringere Flußspannungen als Dioden mit PN-Übergang. Sie werden in zunehmendem Maße auch in integrierten Digitalschaltungen (→ Schottky-TTL, LPSTTL) verwendet, um deren Schaltgeschwindigkeit zu erhöhen.

Schottky-TTL, Transistor-Transistor-Logik mit Schottkydioden (Schottky TTL)

Die Schaltzeit von logischen Schaltungen, bei denen die Schalttransistoren bis in den Sättigungsbereich durchgesteuert werden (z. B. → DTL, TTL), hängt ganz wesentlich davon ab, wie schnell die in der Basis gespeicherten Ladungsträger wieder abfließen können. Durch Schottky-Klemmdioden parallel zur Basis-Emitterstrecke der Schalttransistoren wird der Sättigungsvorgang vermieden und damit die Schaltgeschwindigkeit gesteigert, allerdings auf Kosten einer höheren Leistungsaufnahme (s. Bild 3). Schottky-geklemmte Schalttransistoren werden z. B. in → TTL-Schaltungen verwendet, die dann als Schottky-TTL bezeichnet werden, s. auch → LPSTTL.

Siebdruck (screen printing)

Verfahren zur Herstellung von gedruckten Strukturen, bei dem ein durch → Fotolithografie maskiertes planes (ebenes) Sieb auf ein → Substrat gelegt und durch das ein Druckmittel (Siebdruckpaste) gestrichen wird. In einem Trocken- und Einbrennprozeß werden die gedruckten Strukturen fest mit dem Substrat verbunden. Siebdruck wird in der Elektrotechnik zur Herstellung von → Dickfilmschaltungen verwendet, indem Pasten mit bestimmten elektrischen Eigenschaften (Leiterpasten, Dielektrikumspasten, Widerstandspasten) in mehreren Ebenen übereinandergedruckt werden.

Siliziumdioxid, SiO₂ (silicon dioxide, SiO₂)

Oxid des Siliziums, das ein ausgezeichneter elektrischer Isolator und gegenüber den meisten Säuren und Ätzmitteln "resistent" ist. Es wird zur → dielektrischen Isolierung und Oberflächen- → Passivierung von Halbleiterbauelementen verwendet.

Siliziumnitrid, Si₃N₄ (silicon nitride, Si₃N₄)

Nitrid des Siliziums, das zum → Passivieren von integrierten Schaltungen verwendet wird und die Halbleiteroberfläche schützt. Meist wird es als erste Passivierung verwendet, indem es direkt auf die Halbleiteroberfläche, noch unterhalb der Metallisierung für die Zwischenverbindungen und Anschlußflächen, gebracht wird. Die Metallisierung wird dann zusätzlich durch eine → Siliziumdioxidschicht geschützt.

SOS, Silizium auf Saphir (SOS, silicon on sapphire)

Verfahren zur Herstellung monolithischer integrierter Schaltungen, bei dem als → Substrat nicht ein Siliziumkristall, sondern → Saphir verwendet wird, der sehr gute Isolationseigenschaften hat. Auf den Saphirkristall wird eine Silizium- → epitaxiale Schicht aufgebracht, in die alle Bauelemente eingebaut werden. Danach wird alles überflüssige Silizium weggeätzt, so daß die Bauelemente als voneinander isolierte Inseln stehen bleiben. Man verringert auf diese Weise die → parasitären Kapazitäten beträchtlich und bekommt Schaltungen mit sehr kurzen Anstiegs- und Abfallzeiten. Die SOS-Technik wird hauptsächlich für → MOS-Schaltungen verwendet.

Speicher, Speicherschaltung (memory)

Grundsätzlich eine Anordnung zur Speicherung von Information (z. B. Kernspeicher, Magnetplattenspeicher); hier: Halbleiterspeicher, eine → Digitalschaltung mit einer mehr oder weniger großen Zahl von → binären Speicherzellen, meist in Form von → Flipflops, und Hilfsschaltungen, wie Schreib- und Leseverstärkern, Adressendecodern. Man unterscheidet flüchtige und nicht-flüchtige Speicher; bei den nichtflüchtigen bleibt die eingespeicherte Information auch nach Abschalten der Betriebsspannung im Speicher erhalten. Speicher können als → RAM, → ROM oder → PROM aufgebaut sein, sie können aus → bipolaren oder → unipolaren Bauelementen bestehen.

SSI, Kleinintegration (SSI, small scale integration)

Kennzeichnung für → monolithisch integrierte Schaltungen mit bis zu einigen 10 Grundsaltungen pro → Chip. Zur Unterscheidung → MSI, → LSI, → VLSI.

Subnanosekunden-ECL (sub-nanosecond ECL)

Eine → ECL-Familie, deren Schaltungen Schaltzeiten von unter 1 ns aufweisen (→ Verzögerungszeit einer ODER-Schaltung z. B. 0,7 ns). Es sind dies die zur Zeit „schnellsten Schaltungen“ auf Siliziumgrundlage.

Substrat, Trägermaterial (substrate)

Allgemein die Unterlage, auf der oder in der sich eine integrierte Schaltung befindet. Bei monolithischen Schaltungen ist dies fast immer der Halbleiterkristall (Ausnahme → SOS), bei → Hybrid-schaltungen Keramik oder Glas. Das Substrat sorgt in erster Linie für die mechanische Festigkeit und Handhabbarkeit einer Schaltung, aber auch für die Wärmeabfuhr und für die elektrische Isolierung.

Superchip (superchip)

Bezeichnung für → Chips mit besonders großer Fläche (einige cm²), auf denen sich Schaltungen (zur Zeit Speicherschaltungen) mit mehreren 100 000 Bauelementen befinden. Beispiel ein RAM mit 64 k (65 536 bit) Speicherkapazität auf einem etwa 6 cm² großen Chip.

T

TEGFET (TEGFET, two-dimensional electron gas FET)

→ HEMT

T-Flipflop (T flip-flop)

Eine → bistabile Schaltung mit einem Eingang und einem Ausgang oder einem komplementären Ausgangspaar (s. Bild 5). Ein Eingangssignal (Spannungswert oder Flanke) bewirkt ein Umschalten des Ausgangssignals. T-Flipflops werden als → Frequenzteiler verwendet.

Transistor (transistor)

Allgemeine Bezeichnung für ein verstärkendes Halbleiterbauelement mit drei Elektroden. Zu unterscheiden sind → bipolare Transistoren (NPN-, PNP-Transistor) und → unipolare Transisto-

* resistent: widerstandsfähig, unangreifbar

ren (→ FET). Beim bipolaren Transistor wird der Strom zwischen Kollektor und Emitter über einen Steuerstrom beeinflusst, der zwischen Basis und Emitter fließt. Beim unipolaren Transistor wird der Widerstand zwischen Drain und Source durch ein elektrisches Feld verändert, das durch eine Gatespannung erzeugt wird.

TTL, T²L, Transistor-Transistor-Logik

(TTL, transistor-transistor logic)

(Sprich: „te-Quadrat el“)

Eine digitale → Logikfamilie, bei der die logische Verknüpfung mit Transistoren (Multiemitter-Transistoren) durchgeführt wird (s. Bild 2). Es ist dies die heute am häufigsten benutzte Logikfamilie, auch in ihren Weiterentwicklungen als → Schottky-TTL und → LPSTTL, und wird in einer großen Zahl von Schaltungstypen angeboten, die teilweise sehr umfangreich sind. Sie hat jedoch den Nachteil eines hohen Leistungsbedarfs und einer geringen Störuneempfindlichkeit, auch ist ihre Schaltgeschwindigkeit für viele Anwendungen zu gering.

T³L, Transistor-Transistor-Transistor-Logik

(T³L, transistor-transistor-transistor logic)

(Sprich: „te hoch drei el“)

Eine spezielle Form der → TTL, bei der ein zusätzlicher Transistor verwendet wird, um die Ausgangsleistung zu erhöhen (höherer → Ausgangsfächer) und die Störeempfindlichkeit zu verringern.

UFL

(UFL, unbuffered FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → DFETs und → Schottkydioden besteht (s. Bild 2). Die Transistoren T₁ und T₂ bilden die logische Eingangsverknüpfung, die Transistoren T₃ und T₄ sind als ohmsche Widerstände geschaltet, die Dioden D₁ und D₂ sind Pegelschiebedioden, die den Ausgangspegel an den Eingangspegel angleichen. Die UFL hat gegenüber der → BFL eine geringere Leistungsaufnahme, aber auch eine niedrigere Ausgangsbelastbarkeit.

UND-Schaltung, AND-Schaltung

(AND gate)

Logische → Digitalschaltung, welche die UND-Funktion darstellt. Ihr Ausgangssignal hat dann und nur dann den Wert 1, wenn beide Eingangssignale den Wert 1 haben (s. Bild 16). Mathematische Beschreibung: $X = A \wedge B$ (gelesen: X = A UND B).

Unipolare Bauelemente

(unipolar devices)

Halbleiterbauelemente, in denen für den Stromtransport nur ein einziger Typ von Ladungsträgern (Elektronen oder Löcher) verwendet wird (z. B. → Feldeffekttransistor). Im Gegensatz dazu → bipolare Bauelemente.

UND-NICHT-Schaltung, Exklusionsschaltung

(AND NOT gate, exclusion gate)

Logische → Digitalschaltung, welche die UND-NICHT-Funktion darstellt. Ihr Ausgangssignal hat nur dann den Wert 1, wenn das Eingangssignal A den Wert 1 und gleichzeitig das Eingangssignal B den Wert 0 hat (s. Bild 16). Mathematische Beschreibung: $X = A \wedge \bar{B}$ (gelesen: X = A UND B NICHT).

V

Vergrabene Schicht

(buried layer)

Ein Bereich mit sehr stark → dotiertem Material unter dem Kollektorgebiet von Planartransistoren zur Verringerung des Kollektorwiderstandes. Dabei werden die betreffenden Bereiche der Oberfläche z. B. durch → Diffusion dotiert, und anschließend wird das eigentliche Kollektormaterial durch → Epitaxie aufgebracht. Es folgen danach die Diffusion für Emitter und Basis

Verknüpfungsschaltung

(gate)

Logische → Digitalschaltung, die eine bestimmte logische Funktion oder Kombination von logischen Funktionen darstellt (z. B. UND-, ODER-NICHT-Schaltung). Je nach Zahl der Eingänge ergibt sich eine feste Zahl von Kombinationen der Eingangssignale (bei 2 Eingängen $2^2 = 4$, bei n Eingängen 2^n), für jede dieser Kombinationen gibt es ein und nur ein Ausgangssignal. Den Zusammenhang stellt man der besseren Übersichtlichkeit wegen in Form einer → Wertetabelle dar. Im Gegensatz zu → Folgeschaltungen ist das Ausgangssignal fest mit den Eingangssignalen verknüpft und ändert sich mit jeder Änderung eines Eingangssignals.

Verzögerungszeit, Laufzeit

(delay time, propagation delay time)

Diejenige Zeit, die ein Signal zum Durchlaufen einer digitalen Schaltung vom Eingang bis zum Ausgang benötigt. Sie wird gemessen zwischen dem Punkt mit 50% der Amplitude einer Flanke des Eingangssignals und dem Punkt mit 50% der Amplitude der zugehörigen Flanke des Ausgangssignals (s. Bild 17).

VLSI, Größtintegration

(VLSI, very large scale integration)

Kennzeichnung für → monolithische integrierte Schaltungen mit über 100000 Bauelementen pro → Chip. Zur Unterscheidung → SSI, → MSI, → LSI.

Volladdierer

(full adder)

Digitalschaltung, bestehend aus zwei → Halbaddierern zur Addition zweier einstelliger Binärzahlen unter Berücksichtigung des

Obertrages der nächstniedrigeren Addition, wobei als Ausgangssignal das Summensignal und das Übertragungssignal zur Verfü- gung stehen (s. Bild 12). Zur Addition mehrstelliger Binärzahlen müssen so viele Volladdierer parallel betrieben werden, wie die größte zu addierende Binärzahl Stellen hat. Ein 10-bit-Addierer z. B. kann zwei 10stellige Binärzahlen addieren, er hat 2×10 Eingänge und 11 Ausgänge (10 Summen- und einen Übertragsausgang).

Vorwärts-/Rückwärtszähler

(up/down counter)

Eine besondere Form des → Binärzählers, bei dem elektrisch zwischen Vorwärts- (Aufwärts-) und Rückwärts- (Abwärts-)Zahlen umgeschaltet werden kann.

W

Wertetabelle, Wahrheitstafel

(truth table)

Aufzistung aller möglichen Kombinationen von Eingangssignalen mit den zugehörigen Ausgangssignalen für eine logische Funktion oder Schaltung. Bei n Eingangssignalen ergeben sich 2^n Kombinationen (vgl. Wertetabellen in Bild 16). Bei → Folgeschaltungen kommt noch als weiteres Eingangssignal das Taktsignal hinzu, auf das man jedoch im allgemeinen verzichtet, weil das Taktsignal Voraussetzung für die Funktion einer Folgeschaltung ist und für Taktsignal 0 keinerlei Änderungen im Ausgangssignal auftreten.

Wired-AND, Verdrahtetes UND

(wired-AND)

Eine spezielle Form der UND-Schaltung, die durch einfaches Verbinden der Ausgänge mehrerer logischer → Digitalschaltungen entsteht (s. Bild 16). An dem Verbindungspunkt kann nur dann 1-Signal vorkommen, wenn alle Ausgänge 1-Signal haben. Das Verdrahtete UND ist z. B. bei manchen → TTL-Schaltungen möglich.

Wired-OR, Verdrahtetes ODER

(wired-OR)

Eine spezielle Form der ODER-Schaltung, die durch einfaches Verbinden der Ausgänge mehrerer logischer → Digitalschaltungen entsteht (s. Bild 16). An dem Verbindungspunkt kommt schon dann ein 1-Signal vor, wenn mindestens einer (gleichgültig welcher) der Ausgänge 1-Signal hat. Das Verdrahtete ODER ist bei fast allen → ECL-Schaltungen möglich.

Dipl.-Ing. Gerhard Manke, Wiss.MA
 (B 1311.03.87 G) Forschungsinstitut der DBP beim FTZ Darmstadt

$X = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <p>Äquivalenz (Exklusiv-NICHT-ODER)</p>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	1	$X = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table> <p>Antivalenz (Exklusiv-ODER)</p>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0	$X = A \vee B$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <p>ODER</p>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1	$X = A \vee \bar{B}$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <p>ODER-NICHT</p>	A	B	X	0	0	1	0	1	0	1	0	1	1	1	1	$X = \bar{A}$ <table border="1"> <tr><th>A</th><th>X</th></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table> <p>NICHT</p>	A	X	0	1	1	0
A	B	X																																																																				
0	0	1																																																																				
0	1	0																																																																				
1	0	0																																																																				
1	1	1																																																																				
A	B	X																																																																				
0	0	0																																																																				
0	1	1																																																																				
1	0	1																																																																				
1	1	0																																																																				
A	B	X																																																																				
0	0	0																																																																				
0	1	1																																																																				
1	0	1																																																																				
1	1	1																																																																				
A	B	X																																																																				
0	0	1																																																																				
0	1	0																																																																				
1	0	1																																																																				
1	1	1																																																																				
A	X																																																																					
0	1																																																																					
1	0																																																																					
$X = A \wedge B$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <p>NICHT-ODER</p>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	$X = \bar{A} \wedge \bar{B}$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table> <p>NICHT-UND</p>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0	$X = A \wedge B$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <p>UND</p>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	$X = A \wedge \bar{B}$ <table border="1"> <tr><th>A</th><th>B</th><th>X</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table> <p>UND-NICHT</p>	A	B	X	0	0	0	0	1	0	1	0	1	1	1	0	$X = X_1 \wedge X_2$ <p>Wired AND</p> $X = X_1 \vee X_2$ <p>Wired OR</p>						
A	B	X																																																																				
0	0	0																																																																				
0	1	0																																																																				
1	0	0																																																																				
1	1	1																																																																				
A	B	X																																																																				
0	0	1																																																																				
0	1	0																																																																				
1	0	0																																																																				
1	1	0																																																																				
A	B	X																																																																				
0	0	0																																																																				
0	1	0																																																																				
1	0	0																																																																				
1	1	1																																																																				
A	B	X																																																																				
0	0	0																																																																				
0	1	0																																																																				
1	0	1																																																																				
1	1	0																																																																				

Bild 16. Die wichtigsten logischen Verknüpfungsschaltungen mit ihren Schaltsymbolen und mit Wertetabellen. A und B sind die Eingänge, X die Ausgänge