

22.08.90 //

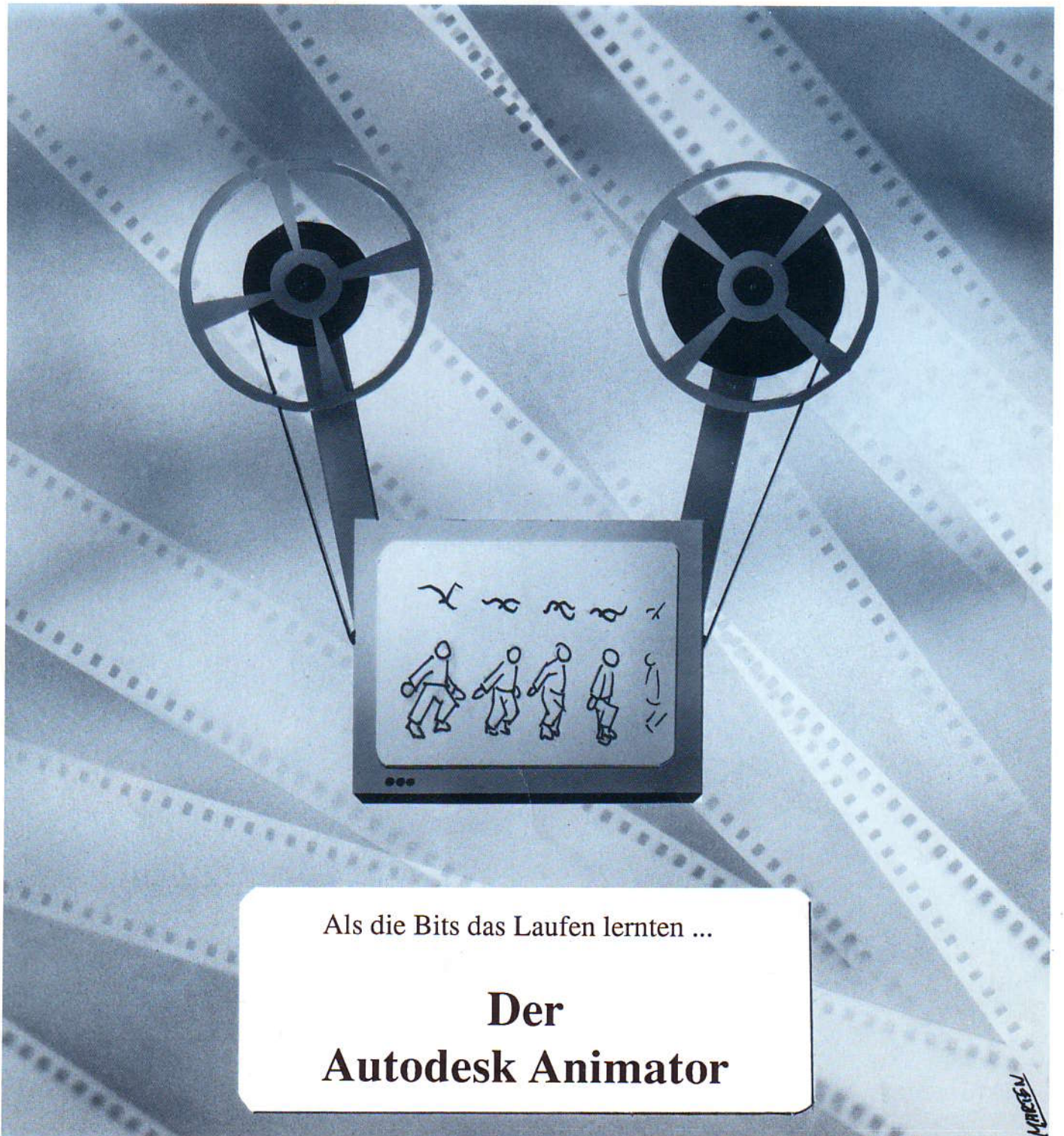
LOOP

25

August 1990

Zeitung für Computer-Bauer, -Anwender, -Programmierer und -Starter

DM 3,50



Als die Bits das Laufen lernten ...

Der Autodesk Animator

Autodesk

Leitartikel

Bericht von der CeBIT

Einige kurze Eindrücke von der CeBIT
25/4

CPU Z80

DSP zum 'reinschnuppern mit dem NDR-Computer

Im 3. Teil gehts nun konkret in die diskrete Fouriertransformation
25/5

Einsatz von FLOMONCG

Fenstertechnik beim NDR-Computer
25/7

Eisenbahnsteuerung mit dem Z80-Einsteigerpaket

Eine REL-Karte steuert 24 Magnete
25/12



Eindruck von der CeBIT: Eine "kleine" Anzahl von vernetzten PCs am Stand der Fa. Novell

CPU 680XX

Computer zum Anfassen

25/14

Ausgabe von Texten und Zeichen über die serielle Schnittstelle

25/18

Schaltungskorrekturen für NDR-Systeme mit CPU 68020

Über einige "Schönheitsfehler" in der Kombination CPU68020 und RAM256
25/19

Alles null und nichtig

Patchwork Teil 7: Löschen und Neuanlegen von Dateien auf DOS-Disketten
25/20

mc-modular AT

Filmreif(e)

Ein Animationsprogramm läßt Trickfilmer- Herzen höher schlagen: Autodesk Animator
25/23

CPU 8088

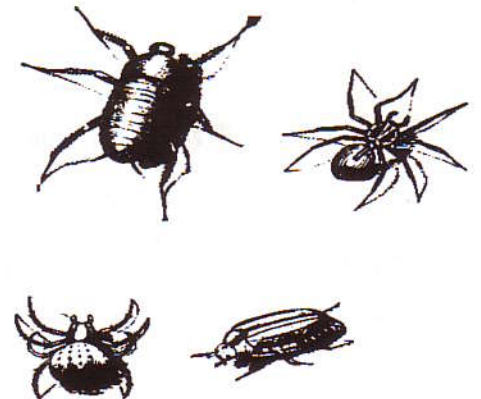
CPU 8088 - MS-DOS

Teil 4: CONFIG.SYS
"... wie ich mein System konfiguriere..."
25/24

Grundlagen

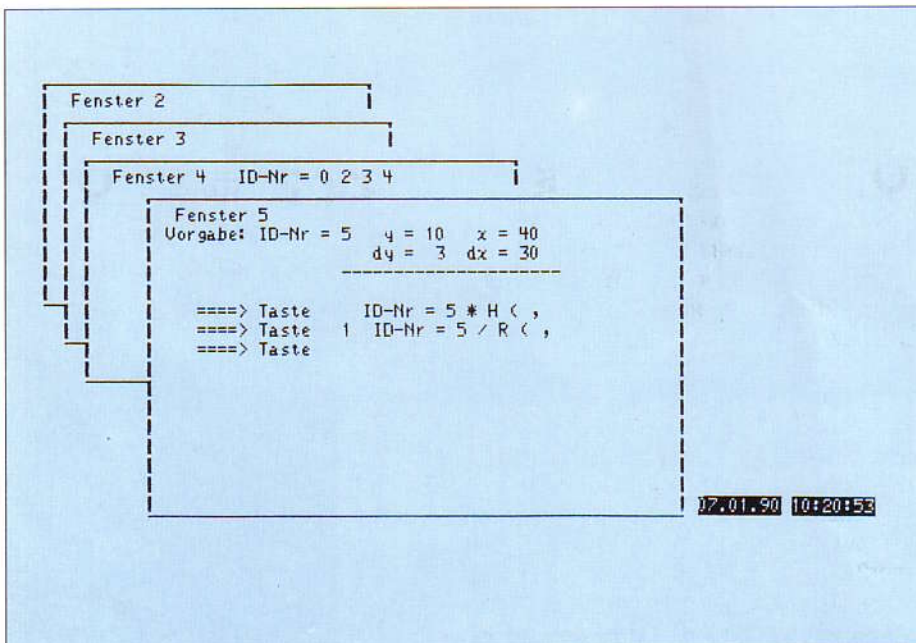
Der Computer ist krank

Teil 2 über Computer-Viren beschäftigt sich mit der Situation einer Verseuchung des Rechners
25/25



New Technologies

Teil 2: Ein Einblick in die Welt der digitalen Signalprozessoren - die verschiedenen DSP-Gruppen und Typen
25/26



Einsatz von FLOMONCG: Hardcopy des Programmes DEMO-AS2.COM

Rubriken

Editorial

Der NDR-Computer-Katalog

25/3

Comic - Strip

Load

25/30

Kleinanzeigen

25/18

Jetzt lieferbar

25/11

Impressum

LOOP Zeitung für Computerbauer

Herausgeber:

Gerd Graf

Druck:

Karl-Heinz Rieder, Kempten

Redaktion:

Gerd Graf, Ulrich Kracker, Nikolaus Bischof

Herstellung und Anzeigenverwaltung:

GES GmbH, Magnusstraße 13, 8960 Kempten

Gestaltung:

BBS Computer-Systeme, Elisabeth Mayr, MARTEN

Anzeigenpreisliste: 6/89

Editorial

In eigener Sache

Der NDR-Computer-Katalog ist in der zweiten Auflage ausverkauft. Achttausend Kataloge gingen über den Versand.

Natürlich gibt es einen neuen Katalog - allerdings, Sie werden sich freuen, dies zu hören, kostenlos.

Warum? Nun, viele Anwender des NDR-Systems wollen ganz einfach wissen, ob und was es Neues gibt. Sie arbeiten vielleicht mit dem Einsteigerpaket oder mit dem Grundausbau und wissen gar nicht, wie man das modulare System erweitern kann. Hier kann der neue Katalog helfen.

Ein weiteres sehr wichtiges Marktsegment ist die DDR. Dort sind seit Jahren vorrangig 8080 bis Z80-basierende Systeme im Einsatz. Das NDR-System mit seinen modularen CPUs vom Z80 über die 68000-Linie bis zum 8088 bietet einerseits den Schutz der bisherigen Software-Investitionen und die Möglichkeit, hochmoderne Zentraleinheiten kennenzulernen. Der "Export" ist nun ja kein Problem mehr - noch vor wenigen Monaten wäre es undenkbar gewesen, eine 68020-Zentraleinheit auszuführen!



Da der neue Katalog kostenlos verteilt wird, muß er auch kostengünstig sein. Dies erreichten wir durch eine Straffung des Programmes um unwichtigere Artikel, durch Verlagerung einiger Artikel hin zum 19"-Ausbildungssystem m-i-c sowie durch einen kompakten Druck. Der neue Katalog wird ab September verfügbar sein; Bestellungen nehmen wir gerne schon heute entgegen. Verwenden Sie dazu die Karte am Ende der LOOP, oder rufen Sie einfach an.

Im Software-Bereich hat sich einiges bewegt - so ist die erweiterte Version des 680xx-Grundprogrammes (V 6.21) nunmehr lieferbar. Dieses Grundprogramm unterstützt nun alles, was neu ist - von der GDP mit Hardscroll bis zum neuen Programmer2.

Im Preis hat sich einiges getan - es ist billiger als die alte Version. Dafür gibts auch

kein Update. Also einfach neu bestellen.

Auf dieses Grundprogramm setzt die neue Version von JADOS 3.5 auf. Herausragendes Merkmal dieser Version: Unterstützung einer Festplatte mit SCSI-Controller. Mit weiteren Möglichkeiten versehen, wuchs JADOS nun von "einfachen" Betriebssystem bis zum hervorragenden Werkzeug für Programm-Entwickler und Anwender.

Im OS/9-Bereich wird sich einiges tun - hier werden vom OS/9 "Papst", Volker Wiegand, neue Software- und auch Hardware-Elemente entwickelt, die das System zu einem Profi- Multi-User und Multi-Tasking System erweitern.

Aber auch der "einfache" Anwender kommt nicht zu kurz - so findet die neue Multi-IO-Baugruppe, die alle wichtigen Ein-Ausgabe-Funktionen beinhaltet, viele Freunde.

Der NDR-Computer lebt - natürlich nur mit Ihnen, Ihrer Unterstützung, Ihren Anwerberberichten und nicht zuletzt Ihren Bestellungen!

Gerd Graf

Gerd Graf

Bericht von der CeBIT

Natürlich haben Sie es schon erraten: In Halle 7 A03 stellten wir selbst aus! Wie immer auf einer Messe, zeigten wir die Neuigkeiten des Hauses.

Obwohl für uns natürlich der Stand in Halle 7 A03 der absolut interessanteste der Hannover Messe CeBIT war, hier einige Eindrücke vom Messegeschehen - ganz subjektiver Art.

ter Software, die in Netzwerken eingesetzt werden können. So ist

Allem voran wurde das neue Multi-IO-Interface präsentiert. Nicht nur das Interface, sondern auch die damit gezeigten Anwendungsbeispiele fanden rege Begeisterung - so wurde das Puste-Windrad, aufgebaut aus LEGO-Technik-Systemen, kräftig beblasen. Der Tageshöchststand und der aktuelle Stand wurden am Bildschirm angezeigt. Über das PC-Interface wurde die Information einer Gabellichtschranke eingelesen, die die Umdrehungszahl des Rades darstellte. Das entsprechende Programm dazu in Turbo-Pascal verfasste Herr Ehrensperger.

Als weitere Applikation zeigten wir die Regelung eines Gewächshauses - ebenfalls mit der Multi-IO.

Reges Interesse fanden die Software für produzierende Betriebe "A.L.F." für Auftrag-Lager-Fertigung sowie unsere neuen 19"-Systeme; der 19"-AT, der auf dem modular-AT basiert sowie das 19" m-i-c-Schulungssystem, das wiederum auf dem NDR-Computer basiert. Es ist sozusagen der NDR-Computer im industriellen Gehäuse mit ECB-Bus.

Doch nun zu den weiteren Eindrücken der Messe. Enttäuscht hat mich - eigentlich wie jedes Jahr - die Halle 1. Großartige Neuigkeiten waren nicht zu sehen, und der hundertste PC oder die portable Schreibmaschine halten meine Begeisterung in Grenzen.

Spannender wurde es dann schon in der Netzwerk-Halle - hier schoß NOVELL mit seinem Stand und dem Massenaufgebot an PCs (siehe Bild) sicher den Vogel ab. Über 200 PCs (!) waren hier unter NetWare /386 mit einem (!) Server verbunden - ein Beweis für die Qualität dieses Multi-User-Systemes.

NOVELL-Anwendungen und Baugruppen, die diese Software unterstützen, gab es in



Eindruck von der CeBit: Eine "kleine" Anzahl von vernetzten PCs am Stand der Fa. Novell.

reicher Zahl. So hat sich wieder ein Standard durchgesetzt. IBM und Microsoft konkurrieren mit dem LAN-Manager, der mittlerweile auch ein Bestandteil des OS/2-Betriebssystemes (V 1.2) ist. Interessante Anwendung bei Microsoft in Halle 7: Hier wurden ein uralter PC, verschiedene ATs und ein Apple Macintosh unter diesem LAN-Manager vernetzt.

An bekannten Programmen gab es viel neues - so die neue Version des StarWriters 5.01, der nun problemlose Tabellen erstellt und auch mit einer Maus-Steuerung und Pop-Up-Menüs versehen ist.

Aldus zeigte seine neueste Version des PageMakers: 4.0. Herausragende Eigenschaft neben vielen Detailverbesserungen: Eine integrierte, ordentliche Textverarbeitung.

Ganz neu auf der Messe waren FAX-Karten - unter anderem Karten mit überlager-

mit relativ bescheidenem Aufwand ein Fax-Gerät sozusagen am Arbeitsplatz eines jeden Netzwerkers vorhanden.

Portables, LapTops und andere waren in großer Zahl da - der heißeste war sicher der extrem flache und leichte LapTop von Sharp, versehen mit DOS auf Eproms und einer 20 MB-Platte.

Floppylaufwerk gabs aus Platzgründen keines mehr - das Laden erfolgt über LapLink, das ebenfalls mit zum Lieferumfang gehört.

Dies die kurzen Eindrücke eines sicherlich sehr knappen und subjektiven Rundganges.

Vielleicht schreiben Sie uns mal, was IH-NEN auf der CeBIT besonders gefallen hat - es muß ja nicht immer nur unser Stand gewesen sein!

DSP zum 'reinschnuppern' mit dem NDR-Computer

Fourier und Euler

Mit Hilfe der Gleichungen (19) und (20) aus LOOP 24 lassen sich die Gleichungen

(6), sowie (9) und (10) (ebenfalls in LOOP 24) auch so formulieren:

Bisher wurden im Rahmen dieser Artikelserie unter anderem die Grundlagen vorbereitet, um nun konkret in die Diskrete Fouriertransformation einzusteigen.

Mathematisch kann der Vorgang der Abtastung mit der Multiplikation des Zeitsigna-

zur Verfügung steht, sondern lediglich einzelne Stützwerte $f(k \cdot T_A)$ mit einem zeitlichen Abstand T_A , redu-

ziert sich das Integral zu einer Summe:

$$f(t) = \sum_{n=-\infty}^{\infty} \underline{c}_n e^{(j n f_1 2\pi t)} \quad (21)$$

$$\underline{c}_n = \frac{1}{T} \int_{t=0}^T f(t) e^{-j n f_1 2\pi t} dt \quad (22)$$

$$\underline{c}_k = \frac{1}{T} \sum_{n=0}^{N-1} f(nT_A) e^{-j n f_1 k 2\pi t} T_A \quad (23)$$

$0 \leq k < \infty$

Diese Schreibweise der Gleichungen (21) und (22) stellt die Ausgangsbasis für die Diskrete Fouriertransformation (DFT) und deren Spezialfall, die Fast Fouriertransformation (FFT) dar.

Die Abtastung eines kontinuierlichen Signals $f(t)$ mit dem Rechner hat zur Folge, daß nur eine begrenzte Menge an Stützwerten von $f(t)$ abgespeichert werden können. Ob mit dieser Folge von Werten exakt eine ganze Periode der Grundwelle des Zeitsignals $f(t)$ erfaßt werden kann oder nicht, ist reine Glückssache.

Signale gehen Fensterln

Auf jeden Fall aber wird aus $f(t)$ ein abgetastetes und periodisches Eingangssignal gewonnen, worauf die Fouriertransformation angewendet werden kann. Das real existierende Zeitsignal $f(t)$ kann selbstverständlich kontinuierlich und vielleicht mit einer beliebigen Zeit T^* periodisch auftreten. Nur der Rechner kann dies unter Umständen nicht voll erfassen!

Man spricht in diesem Zusammenhang von einer Fensterung des Signals. Dieser Effekt bringt meistens eine Verfälschung der Spektralwerte mit sich, kann jedoch oftmals gering gehalten werden, wenn T (Aufzeichnungsdauer im Rechner) groß gegenüber T^* (Periode des wirklichen Signals $f(t)$) ist.

Zeitungspapier und Frequenzen, beides läßt sich falten

les $f(t)$ mit einer DIRAC-Impulsreihe beschrieben werden (DIRAC-Impulse sind ganz kurze Nadelimpulse). Dieser Multiplikation im Zeitbereich entspricht die Faltung im Frequenzbereich. Das bedeutet, daß sich die Frequenzlinien, die dem Zeitsignal $f(t)$ zugeordnet werden, um Frequenzmarken herum legen, die einen Abstand von $1/T_A$ auf der Frequenzachse haben.

Der Index k gibt hierbei die Ordnung (Position auf der Frequenzachse) der Oberschwingung an.

Aufgrund der Periodizität des Spektrums von \underline{c}_k (siehe Abb. 10) erscheint es sinnvoll, den Index k nur bis maximal $k = N$ hochzuzählen, da danach keine neuen Funktionswerte für \underline{c}_k zu erwarten sind. In der Tat genügt es sogar, den Index k in Gleichung (23) lediglich bis $k=1/2 N$ zu zäh-

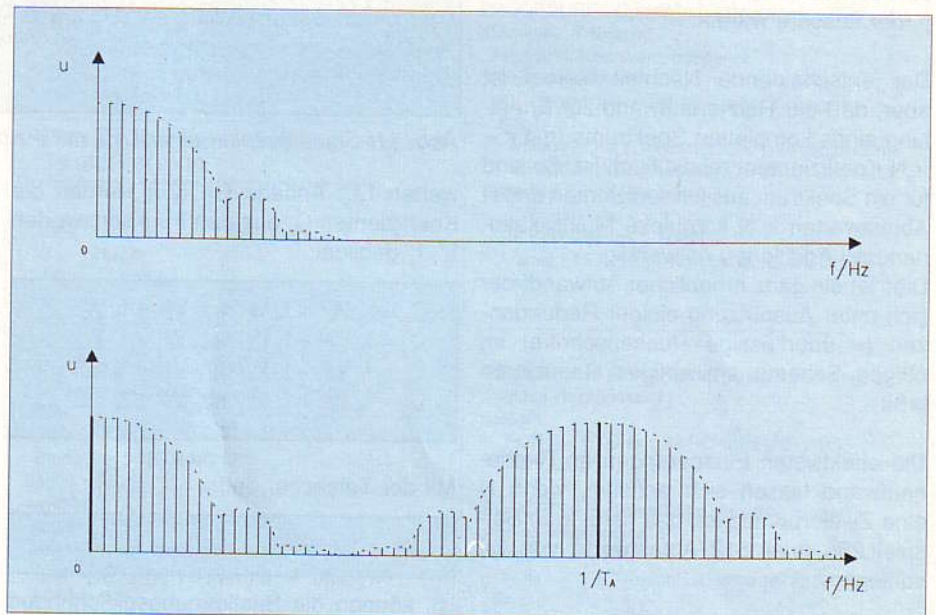


Abb. 10: Gegenüberstellung: nichtgefaltetes und gefaltetes Spektrum (mit DIRAC-Impulsen) eines willkürlichen Zeitsignales.

Weg vom Integral, hin zur Summe

Da zur Bildung der komplexen Koeffizienten \underline{c}_n (siehe Gleichung (22)) nunmehr keine geschlossene Funktion $f(t)$

len. Ab hier wiederholen sich die Koeffizienten \underline{c}_k spiegelbildlich.

Sie wiederholen sich eben deshalb, weil die Eingabewerte $f(n)$ über die DIRAC-Im-

pulsreihe abgetastet werden (Multiplikation) und sich das Spektrum von $f(t)$ um Frequenzlinien herum faltet, die genau um den Abstand $1/T_A$ auseinanderliegen.

Läßt man aber die Koeffizienten bewußt ab $k = 1/2 \cdot N$ enden, ließe sich aus diesem verkürzten Spektrum nurmehr das (periodische) Signal $f(t)$, ohne die multiplikativer verknüpfte DIRAC-Impulsreihe rekonstruieren. Dieser Umstand ist jedoch keineswegs bedauerlich, da die DIRAC-Impulsreihe lediglich zu Hilfszwecken eingeführt wurde.

Da in der Zeit T eine Anzahl von N Abtastungen erfaßt wurden, kann Gleichung (23) auch alternativ geschrieben werden:

$$C_k = \sum_{n=0}^{N-1} f(n) \exp(-j 2 \pi \frac{k n}{N}) \quad (24)$$

Der Ausdruck $\exp(\dots)$ steht für $e(\dots)$ und ist mit den begrenzten Darstellungsmöglichkeiten der Textverarbeitung auf dem Computer zu erklären.

Gleichung (24) stellt somit die Diskrete Fouriertransformation dar, wie sie unmittelbar in ein Rechenprogramm umgesetzt werden könnte. Im wesentlichen wären das zwei ineinander verschachtelte Multiplikationsschleifen. Der innere Zähler wäre n , der äussere wäre k .

Der entscheidende Nachteil hierbei ist aber, daß der Rechenaufwand zur Ermittlung eines kompletten Spektrums (mit $k = 1/2 \cdot N$ Koeffizienten) relativ hoch ist. So sind für ein Spektrum aus k Koeffizienten und N Abtastwerten $k \cdot N$ komplexe Multiplikationen und Additionen notwendig.

Dies ist ein ganz erheblicher Aufwand, der sich unter Ausnützung einiger Redundanzen (= überflüssige Rechenschritte) im obigen Schema um einiges Reduzieren läßt!

Die effektivsten Einsparungen an Rechenaufwand lassen sich erzielen, wenn N eine Zweierpotenz ist, das heißt zum Beispiel 256 oder 512 Abtastwerte von $f(t)$ vorliegen.

Hier wird zuviel berechnet...

Um die Redundanzen in Gleichung (24) besser erkennen zu können, wird der Ausdruck:

$$W = e^{-j 2 \pi / N} \quad (25)$$

eingeführt. Damit verkürzt sich die Gleichung (24) zu:

$$C_k = \sum_{n=0}^{N-1} f(n) W^{kn} \quad (26)$$

Man könnte den komplexen Faktor W^{kn} in Gleichung (26) als einen Gewichtungsfaktor ansehen, dessen Betrag 1 ist und der lediglich eine Drehung des komplexen Zeigers C_k herbeiführt. Diese Zusammenhänge lassen sich am einfachsten anhand eines Beispiels erläutern:

Beispiel:

Es soll die Bildung einer Vier-Punkte-DFT ($N=4$) verfolgt werden, mit den Funktionen

Durch Umstellen der einzelnen Terme wird nun klar, daß einige Terme nur einmal berechnet werden müssen, da deren Zwischenergebnis für mehrere Koeffizienten weiterverwendet werden können:

$$\begin{aligned} C_0 &= f_0 + f_2 W^0 + W^0 (f_1 + f_3 W^0) \\ C_1 &= f_0 + f_2 W^2 + W^1 (f_1 + f_3 W^0) \\ C_2 &= f_0 + f_2 W^0 + W^2 (f_1 + f_3 W^0) \\ C_3 &= f_0 + f_2 W^2 + W^3 (f_1 + f_3 W^0) \end{aligned}$$

Formeln werden zu Schmetterlingen

Dieser zuletzt aufgeschriebene Gleichungsblock ist nun die Grundlage für die bekannte graphische Schmetterlings-Darstellung. Dabei handelt es sich um eine Art von Signalfußdarstellung:

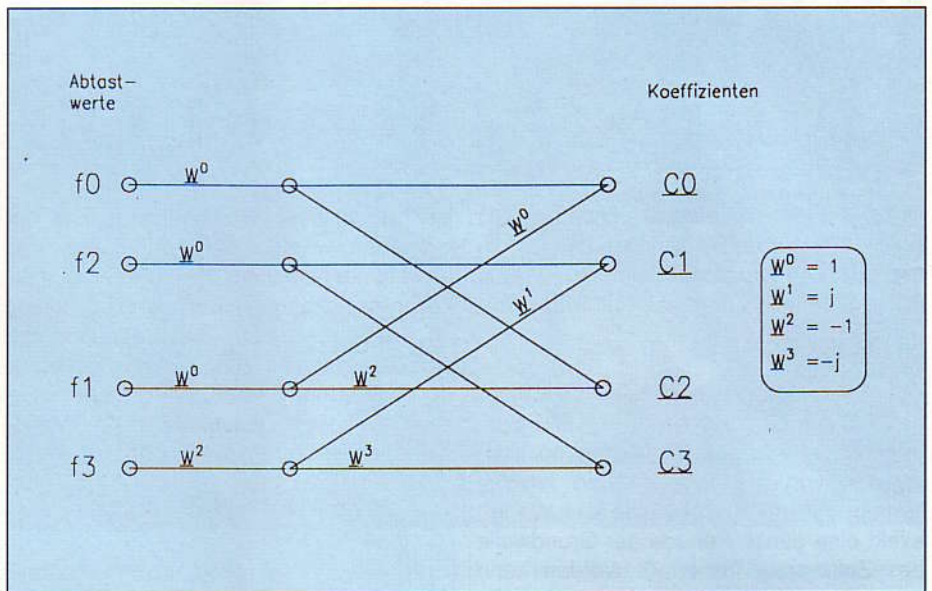


Abb. 11: Signalfußplan einer DFT mit 4 Koeffizienten

werten $f_0 \dots f_3$: Anhand Gl. (26) werden die Koeffizienten C_k aus den Funktionswerten $f_0 \dots f_3$ gebildet:

$$\begin{aligned} C_0 &= f_0 W^0 + f_1 W^0 + f_2 W^0 + f_3 W^0 \\ C_1 &= f_0 W^0 + f_1 W^1 + f_2 W^2 + f_3 W^3 \\ C_2 &= f_0 W^0 + f_1 W^2 + f_2 W^4 + f_3 W^6 \\ C_3 &= f_0 W^0 + f_1 W^3 + f_2 W^6 + f_3 W^9 \end{aligned}$$

Mit der Tatsache, daß:

$$W^n = W^{n \text{ mod } N} \quad (27)$$

ist, können die Bestimmungsgleichungen für $C_0 \dots C_3$ folgendermaßen umgeformt werden:

$$\begin{aligned} C_0 &= f_0 W^0 + f_1 W^0 + f_2 W^0 + f_3 W^0 \\ C_1 &= f_0 W^0 + f_1 W^1 + f_2 W^2 + f_3 W^2 W^1 \\ C_2 &= f_0 W^0 + f_1 W^2 + f_2 W^0 + f_3 W^2 \\ C_3 &= f_0 W^0 + f_1 W^3 + f_2 W^2 + f_3 W^1 \end{aligned}$$

Man beachte, daß die Reihenfolge der Indizes der Eingangswerte nicht in aufsteigender Reihenfolge aufgetragen wurden! Dies hat seinen Grund in der übersichtlicheren Darstellung des Signalfußschemas.

Die Gesetzmässigkeit dieser Reihenfolge läßt sich erkennen, wenn die Indizes in binärer Darstellung in normaler und in der verdrehten Reihenfolge gegenübergestellt werden:

Index dez.	Index binär	Index bitreversed	Index, bitrev., dez
0	00	00	0
1	01	10	2
2	10	01	1
3	11	11	3

Tabelle 1: Bitreversshuffling, zur Umsortierung der Eingabewerte in einen FFT-Algorithmus

Die Ermittlung der FFT gliedert sich also in folgende Teilschritte:

1. Stelle sicher, daß $N = 2^x$ ist
2. Lege je ein reelles und ein imaginäres Feld der Länge N an. Trage Meßwerte in reelles Feld ein, imaginäres Feld zu Null setzen
3. Sortiere die Daten (reell) nach dem Bitreversedverfahren um.
4. Verknüpfe Daten stufenweise mit Faktor W^{kn} , wie in Abb. 11 gezeigt.

Tabelle 2: Grobes Vorgehensmuster bei der Berechnung der FFT

Jetzt darf der Z80 'ran'...

Dies war doch ein gehöriges Maß an Theorie, die nun in die Praxis umgesetzt werden will.

Für den nächsten (und letzten) Teil dieser Artikelserie bleibt es reserviert, eine FFT-Routine für den NDR-Rechner mit dem Z80-Prozessor vorzustellen.

Der Leser mag nun denken: Das geht doch nicht; ein 8 Bit Prozessor und FFT, das sind zwei grundverschiedene Welten! Lassen Sie sich überraschen, so langsam ist ein Z80 auch wieder nicht...

Literatur:

Bartsch:
Taschenbuch Mathematischer Formeln, Verlag Harri Deutsch, 1984

Norbert Schäfer, Manfred Bertuch:
'Butterfly-Algorithmus', erschienen in c't 8/86, Verlag Heise

Dr. R. Best:
'Digitale Meßwertverarbeitung' erschienen in der Reihe: tm - Gastvorlesung, Jahrgänge 1988...90. Verlag Oldenbourg

Jost-Reimer Hoof

Einsatz von FLOMONCG

2. Folge

Fenstertechnik beim NDR-Computer ist seit gut 2 Jahren möglich, nachdem Rüdiger Nahm erst das Programmpaket RNWINDOW und dann die Integration dieses Programmes mit einem überarbeiteten FLOMON herausgebracht hat. An Beschreibungen fehlt es nicht, aber ich suchte vergeblich nach veröffentlichten, leicht verständlichen Demonstrations-Programmen.

In der letzten LOOP veröffentlichte ich vier Programme, die sich mit Datum/Zeit und Statuszeile befaßten. Heute geht es um die Fenstertechnik. Die Beispiele sind in Turbo Pascal und Assembler geschrieben, weil mir Basic nicht so liegt, und die Beispiele im Handbuch im Ansatz in Basic nachzulesen sind.

1. Fenster öffnen und verschieben

Sicher hat jeder RNWINDOW- und FLOMONCG-Besitzer Fensterbeispiele ausprobiert und dann Erfahrungen gesammelt. So sollte man in einem Programm nicht nur ein Fenster öffnen, sondern es am Ende auch wieder schließen, sonst ist der Griff zur Reset-Taste vorprogrammiert, denn der weitere Dialog mit dem Rechner spielt sich immer im aktiven Fenster ab, wie klein es auch ist!

Das erste Demo-Programm DEMOFE1.PAS (Bild 1.1) ist in Turbo Pascal geschrieben und öffnet ein Fenster, das nach einem Menü durch Tastendruck verschoben, verkleinert und vergrößert werden kann. Dabei ist für jede Funktion eine eigene Prozedur geschrieben, um die Übersichtlichkeit zu verbessern.

```
PROGRAM DemoFenster1;
(*.....*)
* Testprogramm fuer WINDOW-Technik von FLO-
* MONCG hier: Fenster-Verschieben / Verkleinern /
* Vergroessern
*.....*
* J.-R. Hoof, Heikendorf * LastUpdate 29.12.1989
* Tel 0431 - 24 20 70
*.....*)
```

```
VAR
  i : integer;
  cha : Char;
```

```
PROCEDURE Hintergrund;
{ Beschreibt den Hintergrund mit Schrift }
VAR
  Str1 : STRING [80];
  StringGross : STRING [80];
Begin
  writeln;
  write ('          ', chr(27), ');');
  write ('Demo - Programm fuer');
  writeln(' Window - Gestaltung ', chr(27), '(');
  { Invers aus }
  Writeln;
  Str1 := 'NDR - Computer * JoHo - Software (C) *';
  FOR i := 1 TO 20 DO
  Begin
    StringGross := Copy (Str1, i, 42-i)+Str1 + Copy (Str1,
1, i);
    writeln (' ', StringGross);
  End;
End;
```

```
PROCEDURE OpenWindow(Nummer, YGroeesse,
XGroeesse, YAbstand,
XAbstand, Kennwert : integer);
{ Fenster oeffnen }
{ Kennwert 1 = unsichtbar, sonst sichtbar }
Begin
  write (#27, '$O', Nummer, Chr (YGroeesse+32));
  write (Chr (XGroeesse+32), Chr (YAbstand+32));
  write ( Chr (XAbstand+32));
  { Window wird unsichtbar eroeffnet }
  IF Kennwert <> 1 THEN
    write ( #27, '$E');{ Window sichtbar machen }
End;
```

```
PROCEDURE GrossRechts (Anzahl : integer);
{ rechts vergroessern }
Begin
  FOR i := 1 TO Anzahl DO write (#27, '$r');
End;
```

```
PROCEDURE GrossUnten (Anzahl : integer);
{ unten vergroessern }
Begin
  FOR i := 1 TO Anzahl DO write (#27, '$d');
End;
```

```
PROCEDURE KleinRechts (Anzahl : integer);
{ rechts verkleinern }
Begin
  FOR i := 1 TO Anzahl DO write (#27, '$l');
End;
```

```
PROCEDURE KleinUnten (Anzahl : integer);
{unten verkleinern }
```



```

Begin
  FOR i := 1 TO Anzahl DO write (#27, '$u');
End;

}

PROCEDURE VerschiebeRechts (Anzahl : integer);
{ nach rechts verschieben }
Begin
  FOR i := 1 TO Anzahl DO write (#27, '$R');
End;

PROCEDURE VerschiebeLinks (Anzahl : integer);
{ nach links verschieben }
Begin
  FOR i := 1 TO Anzahl DO write (#27, '$L');
End;

PROCEDURE VerschiebeOben (Anzahl : integer);
{ nach oben verschieben }
Begin
  FOR i := 1 TO Anzahl DO write (#27, '$U');
End;

PROCEDURE VerschiebeUnten (Anzahl : integer);
{ nach unten verschieben }
Begin
  FOR i := 1 TO Anzahl DO write (#27, '$D');
End;

}

PROCEDURE Menu;
Begin
  Write (chr (27, '*'));{ Bildschirm loeschen }
  Write (chr (30)); { Cursor HOME }
  Writeln;
  Writeln (' 1 = Fenster nach links ');
  Writeln (' 2 = Fenster nach rechts ');
  Writeln (' 3 = Fenster nach oben ');
  Writeln (' 4 = Fenster nach unten ');
  Writeln (' 5 = Fenster rechts verkleinern ');
  Writeln (' 6 = Fenster rechts vergruessern ');
  Writeln (' 7 = Fenster unten verkleinern ');
  Writeln (' 8 = Fenster unten vergruessern ');
  Write (' 9 = Ende ');
End;

{=Hauptprogramm=====}

Begin
write (#27, '*');
{ Bildschirm loeschen }
write (#27, '$@');
{ alle Windows schliessen + neu init. }
Hintergrund;
{ Hintergrund in Window 0 schreiben }
OpenWindow (1, 15, 40, 5, 20, 0);
{ Window 1 oeffnen }
REPEAT
  Menu; { Menue ausgeben }
  Read (KBD, Cha); { Menuwert einlesen }
CASE Cha OF
  '1': VerschiebeLinks (1);
  '2': VerschiebeRechts (1);
  '3': VerschiebeOben (1);
  '4': VerschiebeUnten (1);
  '5': KleinRechts (1);
  '6': GrossRechts (1);
  '7': KleinUnten (1);
  '8': GrossUnten (1);
End;
UNTIL (Cha = '9');
write (#27, '$C'); { Fenster schliessen }
write (' ==> Taste ');
REPEAT { Warten }
UNTIL (KeyPressed);
End.

```

Bild 1.1. Anlegen eines Windows in Turbo-PASCAL

Das eigentliche Hauptprogramm hat daher nur wenige Zeilen. Die Prozedur HINTERGRUND ist eine kleine Spielerei meinerseits und gibt eine lauftext-ähnliche Zeilenausgabe über den ganzen Bildschirm.

Die wichtigste Prozedur ist OPENWINDOW, mit der ein Fenster geöffnet wird. Parameter sind: ID-Nummer, Größe in y- und x- Richtung in Zeichengröße (nicht Pixelgröße), sowie die Koordinaten der linken oberen Ecke. Zusätzlich verlangt die Prozedur einen Kennwert, der angibt, ob das Fenster unsichtbar (Zahl = 1) oder sichtbar (jede andere Zahl) sein soll. Es lassen sich z.B. Fenster unsichtbar öffnen und mit Information beschreiben, um sie dann später im Programm anzuwählen und sichtbar zu machen.

In das geöffnete und sichtbare Fenster wird ein Menü geschrieben, mit dessen Zahlen das Fenster nach allen Seiten verschoben, vergrößert und verkleinert werden kann. Die Zuordnung erfolgt in der Case-Anweisung. Die Verschiebe-Prozeduren erwarten auch einen Parameter, der angibt, um wieviel mal verschoben werden soll. Hier ist der Parameter gleich 1 gesetzt, was vom Prinzip eine einfachere Routine ermöglicht hätte, nämlich:

Begin Write (#27, '\$', Buchstabe) End;
Für Freunde des Z80-Codes habe ich dieses Programm auch in Assembler geschrieben, nämlich DEMO-AS1.ASM (Bild_1.2).

```

TITLE FensterVerschieben 30.12.1989

;Dieses Programm entspricht weitgehend dem DEMO-FE1.PAS,
;und gibt die Routinen in Assembler wieder:
;Fenster nach allen Seiten verschieben, vergrößern
;und verkleinern.

;
;Jost-Reimer Hoof, Heikendorf, LastUpdate:
;30.12.1989
;
System equ 0005 ; System-Einsprung

ORG 100h ; Hauptprogramm
CALL Window ; Fenster eröffnen

Loop:
LD DE, Menu ; Menü ins Fenster
; schreiben
CALL TextAus
LD C, 1 ; Buchstabeneingabe-
; Funktion
CALL System
CP '1'
JP NZ, L2
CALL ESC_Aus ; ESC '$'
LD A, 'L' ; Verschieben nach links
CALL BuAus
L2:
CP '2'
JP NZ, L3
CALL ESC_Aus ; ESC '$'
LD A, 'R' ; Verschieben nach

```

```

;rechts
CALL BuAus
L3:
CP '3'
JP NZ, L4
CALL ESC_Aus ; ESC '$'
LD A, 'U' ; Verschieben nach
; oben
CALL BuAus
L4:
CP '4'
JP NZ, L5
CALL ESC_Aus ; ESC '$'
LD A, 'D' ; Verschieben nach un-
; ten
CALL BuAus
L5:
CP '5'
JP NZ, L6
CALL ESC_Aus ; ESC '$'
LD A, 'I' ; Verkleinern rechts
CALL BuAus
L6:
CP '6'
JP NZ, L7
CALL ESC_Aus ; ESC '$'
LD A, 'r' ; Vergrößern rechts
CALL BuAus
L7:
CP '7'
JP NZ, L8
CALL ESC_Aus ; ESC '$'
LD A, 'u' ; Verkleinern unten
CALL BuAus
L8:
CP '8'
JP NZ, L9
CALL ESC_Aus ; ESC '$'
LD A, 'd' ; Vergrößern unten
CALL BuAus
L9:
CP '9' ; = Ende
JP NZ, Loop

LD DE, Weiter; warten bis Taste ge-
; drückt
CALL TextAus
LD C, 1 ; Buchstabeneingabe-
; Funktion
CALL System
CALL ESC_Aus ; ESC '$'
LD A, 'C' ; aktives Fenster schlie-
; ßen
CALL BuAus
RET

;===== Unterprogramme
WindowAus:
LD A, 27 ; ESC '*' = Bildschirm lö-
; schen
CALL BuAus
LD A, '*'
CALL BuAus
RET

;
;
Window:: Fenster-öffne-String
CALL ESC_Aus
LD DE, Wind_String ; Fenster öff-
; nen
CALL TextAus
CALL ESC_Aus
LD A, 'E' ; Fenstersicht-
; barmachen
CALL BuAus
RET
Wind_String:
DB 'O', 2, 32+15, 32+40, 32+5, 32+20,
; '$'
; ^ ID, y, x, dy, dx
; ^ = O wie Otto
;
TextAus: ; Textausgabe
LD C, 9 ; Stringausgabe-Funk-
; tion
CALL System
RET

;
ESC_Aus: ; Ausgabe von ESC und "$"
LD A, 27 ; ESC ausgeben

```



```

CALL BuAus
LD A, '$' ; "$" ausgeben
BuAus: ; Buchstabenausgabe
LD E, A
LD C, 2 ; Buchstabenausgabe-
Funktion
CALL System
RET

;
Weiter: DB 27, ')' ; invers ein
DB ' =====> Taste '
DB 27, '(' ; invers aus
DB '$'

;===== T e x t
Menu:
DB 1Ah, ' '
DB 27, 'I' ; invers ein
DB ' Menue ', 13, 10
DB 27, 'k' ; invers aus
DB ' 1 = Fenster nach links ', 13, 10
DB ' 2 = Fenster nach rechts ', 13, 10
DB ' 3 = Fenster nach oben ', 13, 10
DB ' 4 = Fenster nach unten ', 13, 10
DB ' 5 = Fenster rechts verkleinern ', 13, 10
DB ' 6 = Fenster rechts vergrössern ', 13, 10
DB ' 7 = Fenster unten verkleinern ', 13, 10
DB ' 8 = Fenster unten vergrössern ', 13, 10
DB ' 9 = Ende $'

```

Bild 1.2. Arbeiten mit Fenstern, diesmal in Assembler

Kardinalproblem ist die Stringausgabe, die sich unter CPM mit der Funktion 9 lösen ließe, gäbe es nicht das "\$"-Zeichen. Dieses Zeichen (Hex-Code = 24 hex) beendet bei CPM die Stringausgabe, und dieses Zeichen ist leider in allen Fenster-Kommandos enthalten. Ich biete gedanklich folgende Lösungen an:

1. Man gibt jedes Zeichen einzeln mit der CPM-Funktion 2 aus.

2. Man schreibt ein eigenes Unterprogramm, dem nach UP-Aufruf der auszugebende String folgt, abgeschlossen mit einer "0". Ich fand in der mc 6/1984 einen guten Vorschlag von Mathias Neuhaus unter "Z80 - Kniffe".

3. Man gibt die beiden Zeichen "ESC" und "\$" mit der CPM-Funktion 2 aus und den Rest des Kommandos entweder genauso oder mit der Funktion 9.

4. Man schreibt sich sein eigenes Unterprogramm mit der CPM-Funktion 2.

Mir gefallen Lösungen 2 und 4 am besten. Doch habe ich die Lösung 2 nicht weiter verfolgt, weil die Technik zu unübersichtlichen Programmen führt. DEMO-AS1.ASM habe ich nach Vorschlag 3, das nächste Programm nach Lösung 4 programmiert.

Kurz ein paar Worte zu diesem Assembler-Programm: Mit dem Unterprogramm WIN-

DOW wird das Fenster geöffnet, indem "ESC" und "\$" mit dem UP "ESC_Aus" und der Rest des Kommandos als String "Wind_String" an den Rechner gegeben werden. Nach der Menü-Ausgabe und Einlesen der Kennzahl wird die entsprechende Funktion durch Vergleichen ausgewählt, wobei nach dem UP "ESC_Aus" der fehlende Buchstabe mit der CPM-Funktion 2 ausgegeben wird. Mit der Eingabe von "9" wird das Programm beendet, nach Drücken einer Taste wird das aktive Fenster gelöscht.

2. Fenster positionieren und Parameter abfragen

Beim Demo-Programm DEMO-FE2 (Bild 2.1) werden mehrere Fenster geöffnet, im 4. Fenster werden die ID-Nummern aller bis dahin vereinbarten Fenster abgefragt und ausgegeben. Dabei wird auffallen, daß es kein Fenster mit der ID-Nummer 1 gibt, sondern nur 0, 2, 3 und 4. Dafür zeigt der 5. Wert "-35". Wie ist das zu erklären? Im Puffer steht sicher hinter den 4 Werten ein Carriage-Return = 13 dez, von diesem Wert wird per Programm 48 dez = '0' abgezogen, was dann -35 ergibt.

```

PROGRAM DemoFenster2;
(*
* Testprogramm fuer WINDOW-Technik von FLO
* MONCG hier: Position des aktiven Fensters, Posi-
* tion und Groesse neu setzen, ID abfragen
*)
* J.-R. Hoof, Heikendorf, LastUpdate 29.12.1989
* Tel 0431 - 24 20 70
*)

VAR
wid, wy, wx, wdy, wdx : integer;
Cha : Char;

PROCEDURE OpenWindow (* Fenster oeffnen *)
(Nummer, YGroesse, XGroesse, YAbstand,
XAbstand, Kennwert : integer);
(* Kennwert 1 = unsichtbar, sonst sichtbar *)
Begin
write(#27, '$O', Nummer, Chr(YGroesse+32));
write(Chr(XGroesse+32), Chr(YAbstand+32));
write(Chr(XAbstand+32));
(* Window unsichtbar eroeffnen *)
IF Kennwert <> 1 THEN
write (#27, '$E');
(* Window sichtbar machen *)
End;

PROCEDURE Weiter;
Begin
Write (' =====> RETURN - Taste');
Read;
Writeln;
End;

PROCEDURE FensterID;
(* Fenster-ID holen *)
VAR id1, id2, id3, id4, id5 : Char;
Begin
Write (chr(27), '$I');
(* Fenster-ID holen *)

```

```

Read (KBD, id1);
Read (KBD, id2);
Read (KBD, id3);
Read (KBD, id4);
Read (KBD, id5);
(* Cursor positionieren *)
Write (chr(27), '=', chr(32 + 0));
write(chr(32 + 2));
Writeln ('FensterID = ');
Writeln ((Ord (id1) - 48):3);
Writeln ((Ord (id2) - 48):3);
Writeln ((Ord (id3) - 48):3);
Writeln ((Ord (id4) - 48):3);
Writeln ((Ord (id5) - 48):3);
End;

```

```

PROCEDURE FensterParameter;
(* Parameter des aktiven Fensters *)
VAR ID, y, x, dy, dx : Char;
(* abfragen *)
Begin
Write (chr(27), '$?');
Read (KBD, ID);
Read (KBD, y);
Read (KBD, x);
Read (KBD, dy);
Read (KBD, dx);
Writeln (' FensterID - Nr.: ', Ord(ID) - 48);
Write (' y = ', Ord (y) - 32, ' x = ');
writeln(Ord (x) - 32);
Write(' dy = ');
writeln(Ord (dy) - 32, ' dx = ', Ord (dx) - 32);
End;

```

```

PROCEDURE FensterNeuSetzen;
(* Position neu festsetzen *)
VAR ID, y, x, dy, dx : Char;
Begin
Writeln (' Vorgabe: ID = ', wid:2, ' y = ', wy:3, ' x = ',
wx:3);
Writeln (' dy = ', wdy:3, ' dx = ', wdx:3);
Weiter;
Write (chr(27), '$I', Chr (32 + 8), Chr (32 + 12));
End; (* Fenster neu setzen *)

```

```

PROCEDURE FensterDatenNeu;
(* Fenstergroesse neu setzen *)
Begin (* mit Abfrage der Durchfuehrung *)
Writeln (' Fenster wird vergroessert ');
Weiter;
Write (chr (27), '$A'); (* Antwort ein *)
Write (chr (27), '$i', Chr (32 + 15), Chr (32 + 50));
Read (KBD, Cha);
(* ^ hier mal zum Test "20" einsetzen *)
IF Cha = '1'
THEN Writeln (#27, ' '); +++++ Ausführung
o.k. ' ;
#27, '('
ELSE Writeln (#27, ' '); ***** Ausführung nicht
durchgefuehrt, #27, '(';
Write (chr (27), '$B'); (* Antwort aus *)
End;

```

```

(=== Hauptprogramm =====)

Begin
write (#27, '*'); (* Bildschirm loeschen *)
write (#27, '$');
(* alle Windows schliessen + neu initialisieren *)
OpenWindow (2, 10, 30, 2, 2, 0);
OpenWindow (3, 10, 30, 4, 4, 0);
OpenWindow (4, 10, 40, 6, 6, 0);
FensterID;
wid := 5; (* Vorgaben fuer Fenster 5 *)
wy := 10;
wx := 40;
wdy := 10;
wdx := 30;
OpenWindow (wid, wy, wx, wdy, wdx, 0);
FensterNeuSetzen;
(* aktives Fenster neu positionieren *)

```



```
FensterParameter;
(* Parameter des aktiven Fensters abfragen *)
FensterDatenNeu;
(* aktives Fenster mit neuen Daten *)
FensterParameter;
(* Parameter des aktiven Fensters abfragen *)
Weiter;
write (#27, '$');
(* alle Windows schliessen + neu initialisieren *)
End.
```

Bild 2.1. Öffnen von mehreren Fenstern unter PASCAL.

Weiter im Programm werden im 5. Fenster die Parameter ausgegeben, mit denen das Fenster geöffnet wurde. Dann wird das Fenster umgesetzt und schließlich das Fenster neu gesetzt mit jeweiliger Ausgabe der abgefragten Parameter.

Bei der Prozedur "FensterDatenNeu" ist noch die Möglichkeit eingebaut, die korrekte Ausführung der Operation abzufragen. Dies wird durch eine "1" bzw. durch einen o.k.-String angezeigt. Ändern Sie mal versuchsweise in dieser Prozedur an der gekennzeichneten Stelle 15 in 20, dann wird die Operation nicht ausgeführt und dies auch angezeigt durch Ausgabe einer 0" bzw. eines entsprechenden Textes. In Bild 2.2 ist das Programm als Assembler-Listing wiedergegeben. Hier ist für die Stringausgabe das Unterprogramm "TextAus" nach einzelnen Kommandos für das Fenster-Handling übersichtlich zusammengefaßt sind.

```
TITLE Fenster-Position30.12.1989

;Dieses Programm setzt ein Fenster um, fragt die
;Fenster-Parameter ab und setzt das aktive Fenster
;mit neuen Daten.entpricht DEMO.FE2.PAS

;Jost-Reimer Hoof, Heikendorf
;*LastUpdate: 30.12.1989

System equ 0005
Cl equ 0F003h ; Zeichen von Tastatur
; tur

ORG 100h ;Hauptprogramm

;-----
LD HL, Init ;Bildschirm löschen
CALL TextAus
LD HL, Fenster2; 2. Fenster öffnen
CALL TextAus
LD HL, Fenster3; 3. Fenster öffnen
CALL TextAus
LD HL, Fenster4; 4. Fenster öffnen
CALL TextAus
CALL ID_Abfrage; ID-Abfrage
LD HL, Fenster5; 5. Fenster öffnen
CALL TextAus
CALL Weiter ; Warten bis Taste gedrückt
; drückt
LD HL, NeuSetzen; Fenster auf neue
; Position ;setzen
CALL TextAus
CALL AbfrageParameter; Parameter ab-
```

```
; fragen
CALL Weiter ; Warten bis Taste gedrückt

LD HL, AbfrageEin; Abfrage nach Erfolg
; ein
CALL TextAus
LD HL, DatenNeu; Fenstergröße neu
; setzen
CALL TextAus
LD C, 1 ; Erfolg abfragen ohne
; Ausgabe
CALL System
LD HL, AbfrageAus; Abfrage nach Er-
; folg aus
CALL TextAus
CALL AbfrageParameter; Parameter ab-
; fragen
CALL Weiter

LD HL, EndeString; Alle 4 Fenster lö-
; schen
CALL TextAus
RET

; Unterprogramme
ID_Abfrage: ; ID-Abfrage
LD HL, ID_String
CALL TextAus
LD A, 5 ; 5 Abfragen
CALL TastaturAbf
LD A, 4 ; 4 Ausgaben
CALL ID_BuffAusgabe
RET

AbfrageParameter: ; Parameter des Fen-
; sters abfragen
LD HL, Parameter
CALL TextAus
LD A, 5 ; 5 Abfragen
CALL TastaturAbf
LD A, 5 ; 5 Ausgaben
CALL BuffAusgabe
RET

TastaturAbf: ; direkte Tastaturabfrage
; und Ergebnisse in "Buffer"
LD HL, Buffer
LD B, A
Ta: PUSH BC
PUSH HL
CALL Cl ; Tastaturabfrage
POP HL
LD (HL), A
INC HL
POP BC
DJNZ Ta
RET

BuffAusgabe: ; Buffer ausgeben
LD B, A
PUSH BC
LD HL, ID_Text ; ID-Text ausgeben
CALL TextAus
LD HL, Buffer
LD A, (HL) ; 1. Wert aus Buffer ho-
; len
LD C, 30h ; und 30 hinzuaddieren,
ADD A, C ; damit ASC-Zeichen
; (05 + 30h)
DEC B
INC HL
PUSH HL
CALL BuchAus ; und ausgeben
CALL Blank ; Blank ausgeben
POP HL
POP BC
Ab: PUSH BC
LD A, (HL)
INC HL
PUSH HL ; andere Werte des Buf-
; fers aus-

CALL BuchAus ; geben, dazu ein ' '
CALL Blank ; Blank ausgeben
POP HL
POP BC
DJNZ IDAb
RET

Blank: ; Blank ausgeben
LD A, ' '
BuchAus: ; Buchstaben ausgeben
LD E, A
LD C, 2
CALL System
RET

TextAus: ; String ausgeben
LD A, (HL) ; Adresse des Strings
CP 0
RET Z
PUSH HL
CALL BuchAus ; Buchstaben ausge-
; ben
POP HL
INC HL
JP TextAus

Weiter: ; Warten bis Taste gedrückt
LD DE, WeiterString
LD C, 9 ; Stringausgabe-Funk-
; tion
CALL System
LD C, 1 ; Konsoleingabe-Funk-
; tion
CALL System
RET

WeiterString:
DB 13, 10, ' =====> Taste $'
; Fenster-Kommando-Strings

Init:
DB 27, '*', 0 ; Bildschirm löschen

Fenster2:
DB 27, '$O', 2, 32+10, 32+30, 32+2
DB 32+2
; Fenster öffnen
DB 27, '$E' ; Fenster sichtbar ma-
; chen
DB ' Fenster 2 ', 0

Fenster3:
DB 27, '$O', 3, 32+10, 32+30, 32+4
DB 32+4
; Fenster öffnen
```



```

DB 27, '$E' ; Fenster sichtbarmachen
DB ' Fenster 3 ', 0
Fenster4:
DB 27, '$O', 4, 32+10, 32+40, 32+6
DB 32+6
; Fenster öffnen
DB 27, '$E' ; Fenster sichtbar-
; machen
DB ' Fenster 4 ', 0
Fenster5:
DB 27, '$O', 5, 32+10, 32+40, 32+10
DB 32+30
; Fenster öffnen
DB 27, '$E' ; Fenster sichtbar
; machen
DB ' Fenster 5 ', 13, 10
DB ' Vorgabe: ID-Nr = 5 y = 10 x = '
DB ' 40 ', 13, 10
DB ' dy = 3 dx = 30 ', 13, 10
DB ' _____ ', 0
;
ID_String:
DB 27, '$I', 0 ; Fenster-ID holen
NeuSetzen:
DB 27, '$I', 32+8, 32+12, 0 ; Fenster
; neu setzen
Parameter:
DB 27, '$?', 0 ; Parameter des akt.
; Fensters abfragen
DatenNeu:
DB 27, '$I', 32+15, 32+50, 0 ; Fenster
; gröÙe neusetzen
AbfrageEin:
DB 27, '$A', 0 ; Durchführ-Bestäti-
; gung ein
AbfrageAus:
DB 27, '$B', 0 ; Durchführ-Bestäti-
; gung aus
;
EndeString: ; FensterschlieÙe-
; String
DB 27, '$C' ; 5. Fenster schlie-
; ßen
DB 27, '$C' ; 4. Fenster schlie-
; ßen
DB 27, '$C' ; 3. Fenster schlie-
; ßen
DB 27, '$C', 0 ; 2. Fenster schlie-
; ßen
;..... Ende .....

```

Bild 2.2. Das Öffnen von Windows unter Assembler

Es war nicht einfach, das Abfragen der ID-Nummern im Fenster 4 und der Parameter im Fenster 5 zu realisieren. Der Versuch,

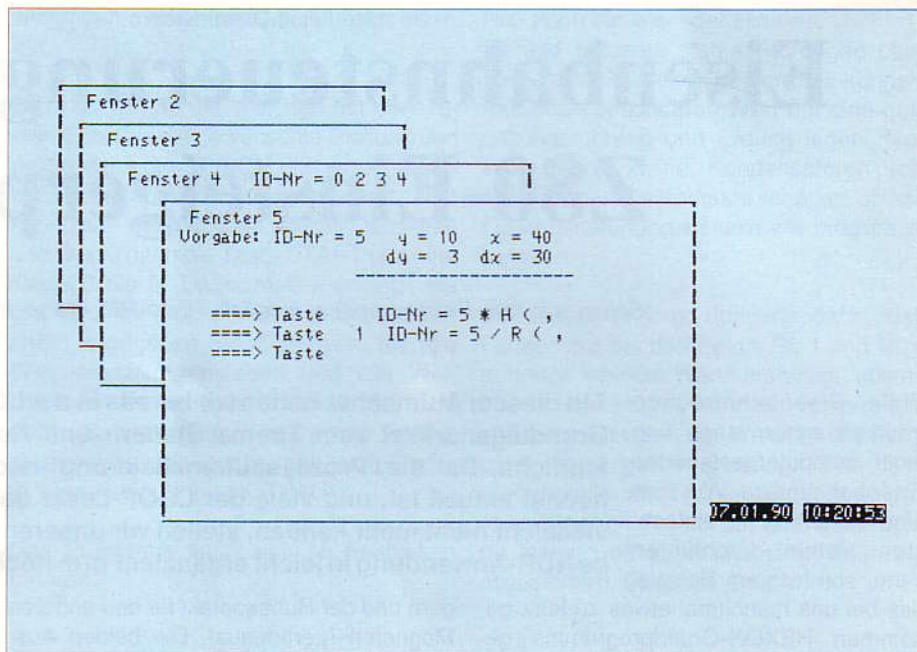


Bild 2.3: Die Behlohnung der Tipparbeit: Übersicht. 'Fenster unter FlomonCG

den Tastaturstatus CSTS abzufragen, brachte kein Ergebnis, da das Einschleusen der Daten anders realisiert wird. Nur direkte Abfrage der Tastatur über den Sprung F003h = CI brachte Erfolg.

Bild 2.3 zeigt eine Hardcopy dieses Programmes DEMO-AS2.ASM. Im Fenster 4 sind die bis dahin vereinbarten ID-Nummern zu sehen, im Fenster 5 wird zuerst die Vorgabe, d.h. die Parameter für die Fensteröffnung direkt ausgegeben. Nach dem Umsetzen werden die neuen Parameter in Form von "5 * H (, " ausgegeben. Während die ID-Nummer noch leicht in ein ASC-Zeichen umgewandelt werden konnte, steht "*" für 2A hex = 42 dez. Zieht man hiervon vereinbarungsgemäß 32 dez ab, so erhält man 10 dez, was dem y-Wert entspricht. Entsprechend ist "H" - 32 dez = 40 dez der x-Wert. Die beiden nächsten Werte stehen für dy und dx. Durch eine Umwandlungs-Routine HEX nach ASC ließe sich der Wert besser ausgeben, das

hätte aber dieses Demo-Programm nur unnötig verlängert.

Entsprechend lassen sich die nächsten Werte deuten: Vor dem Zeichen "ID - Wert" steht eine "1". Dieser Wert gibt die Bestätigung, daß die Fensteroperation erfolgreich war, da vor dieser Operation der Abfrage-Modus eingeschaltet wurde. Ändert man im String "DatenNeu" an der gekennzeichneten Stelle den y- Wert von 32 + 15 in 32 + 20, so kann das Fenster 5 nicht auf diese Größe erweitert werden. Die Operation wird nicht ausgeführt, und der Erfolgscode "0" = "nicht durchgeführt" wird ausgegeben. Eine solche Abfrage läßt sich gut für gewisse Rettungs- oder Modifikationsmöglichkeiten nutzen.

3. Fortsetzung

In der nächsten LOOP werde ich Demo-Programme anbieten über Fadenkreuz-Umdefinieren, Notizbuch-Möglichkeiten, Lokal-Modus und Monitor-Modus.

In eigener Sache

Jetzt lieferbar

Grundprogramm V 6.21:

Schon in der LOOP 24 wurden die Merkmale der neuen GP-Version für die Prozessoren 680xx hervorgehoben. Diese Version V 6.21 ist für alle Prozessoren die

aktuellste und steht zur Auslieferung bereit.

- Das Grundprogramm ist im Durchschnitt billiger geworden (ca. 20%).

- Das GP V 6.2 wird ohne Dokumentation geliefert, ist aber mit der bisherigen Doku betreibbar, da 100% aufwärtskompatibel

- Das komplett neuerstellte, 280 Seiten starke Handbuch V 6.2 ist für alle GP Varianten (ob Disk oder EPROM) gültig.

Kleiner Wermutstropfen:

Dadurch daß der Preis des GP gesenkt werden konnte, entfällt leider auch die Möglichkeit von Updates älterer GP-Versionen.

Rolf Dieter Klein

Eisenbahnsteuerung mit dem Z80-Einsteigerpaket

„Schon mit dem Einsteigerpaket möglich - Eine REL-Karte steuert 24 Magnete

Viele Eisenbahnfreunde träumen schon lange von einer computergesteuerten Eisenbahnanlage. Wie man eine Steuerung mit einfachsten Mitteln durchführen kann, soll hier am Beispiel des bei uns manchmal etwas zu kurz gekommen HEXON-Grundprogramms gezeigt werden. Das Programm kann natürlich auch für alle anderen Versionen des NDR-Klein-Computers umgeschrieben werden, daher existiert auch eine Programmversion in PASCAL.

Die Aufgabe:

Gesteuert werden soll eine Märklin-Eisenbahn. Märklin arbeitet mit Wechselstrom. Nichtsdestotrotz wird die Märklin-Lösung natürlich auch für Gleichstrom-Eisenbahnen gültig sein.

Eine Weiche besitzt bei Märklin z.B. zwei Magnete. Ein Magnet schaltet auf Geradeausfahrt, ein zweiter auf Abbiegen. Die Magnete werden dazu nur für kurze Zeit unter Strom gesetzt und sind im Normalfall stromlos.

Aus 4 mach 12

Die REL-Baugruppe besitzt 8 Relais, mit je zwei Umschaltkontakten. Folglich könnte man mit einer REL-Baugruppe bei oberflächlicher Betrachtung eigentlich nur vier Weichen schalten, je ein Relais für einen Elektro-Magneten.

Doch es gibt einen Trick, um sogar 12 Weichen, also 24 Magnete schalten zu können: es wird dazu die sogenannte Multiplextechnik verwendet. Zwei Relais versorgen je eine Gruppe von 6 Relaiskontakten mit Strom (6 x 1 Umschaltkontakte). Zwei Kontaktpaare werden jeweils gleichzeitig von einem Relais (2 x Um) geschaltet. Da immer nur eine von beiden Gruppen aktiv ist, kommt man mit 6 Relais für die beiden Gruppen aus. Nur wird für jeden Montageartikel ein Kontakt verwendet: der Arbeitskontakt für den einen (z.B. abbie-

Mit diesem Aufmacher hatten wir bereits in der LOOP 7 einen Grundlagenartikel zum Thema Steuern und Regeln veröffentlicht. Da die Prozessautomatisierung nach wie vor höchst aktuell ist, und viele der LOOP-Leser das Heft Nr. 7 vielleicht nicht mehr kennen, stellen wir unseren Lesern diese NDR-Anwendung in leicht ergänzter Form noch einmal vor

gen) und der Ruhekontakt für den anderen Magneten (geradeaus). Die beiden Auswahlrelais geben die Schaltspannung an die Kontakte der Umschaltrelais nur kurzzeitig weiter (Schaltimpuls), dadurch ergibt sich genau das gewünschte Verhalten. Bild 1 zeigt den Verdrahtungsplan.

Einen kleinen Nachteil hat das Verfahren auch. Jeweils 6 Magnetartikel werden

durch den Stromimpuls gleichzeitig aktiviert. Der Computer muß sich also die Stellung aller Artikel merken (was ihm aber nicht schwerfällt).

Beispiel für den Programmablauf:

Weiche 5 soll gerade gestellt werden. Dazu wird RL5 in die entsprechende Stellung gebracht (Relais an oder aus), alle Relais, RL3, RL4, RL6 und RL7 müssen jetzt auch in die aktuelle Stellung springen, denn die restlichen Weichen werden ja auch nochmals gestellt. Nun wird R1 kurz eingeschaltet. Alle Weichen, 1, 3, 5, 7, 9 und 11 werden nun gestellt.

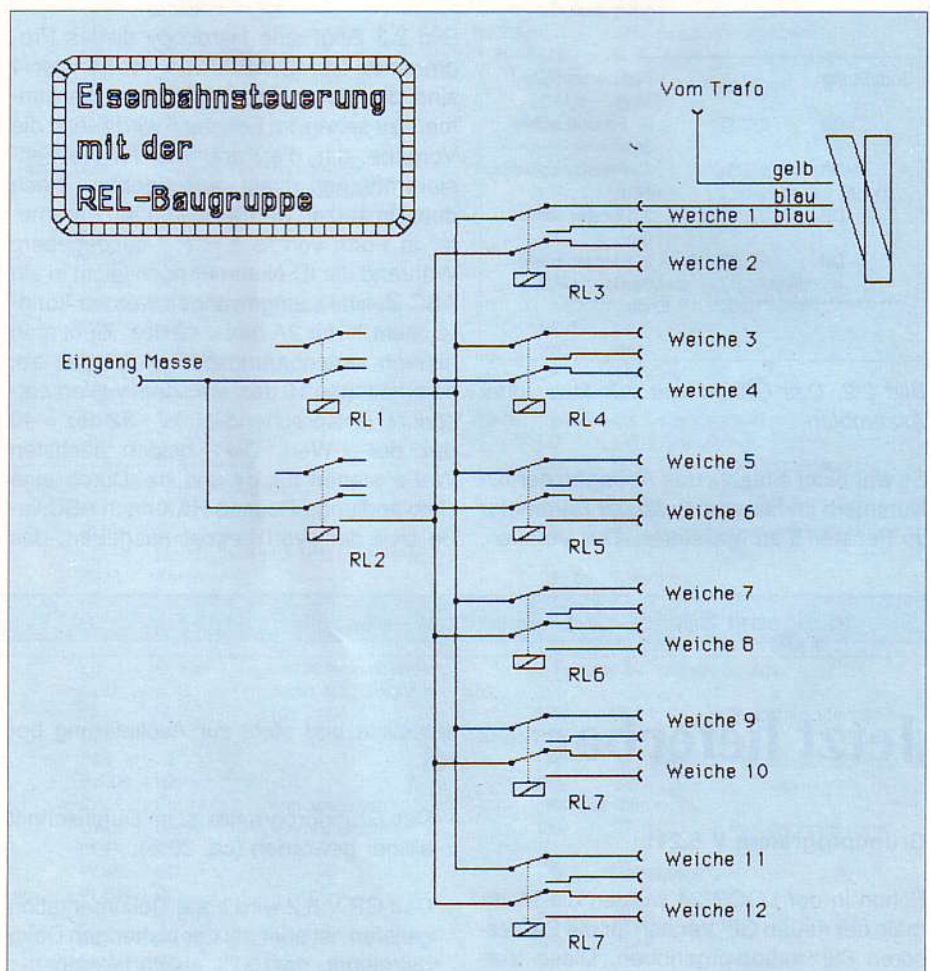


Bild 1: Verdrahtung zwischen REL-Baugruppe und Weichenantrieb

Die "Eisenbahn-Software"

Bild 2 zeigt das Programmlisting für den HEXMON, die REL-Baugruppe wird auf Adresse 50h eingestellt (01010000; 0 = Schalter an).

Die HEX-Tastatur wird nach dem Programmstart zum Stellen der Weichen verwendet. Übrigens kann man natürlich auch Signale o.ä. mit der Baugruppe betreiben. Die Tasten 0 bis C steuern die 12 Weichen. Jeder Tastendruck stellt die Weiche in die entgegengesetzte Richtung. Wenn man die Taste BEF drückt, so werden alle Weichen in eine Vorzugslage gestellt.

Weichen stellen mit Grips

Wozu dann überhaupt einen Computer? Nun - man kann sich bestimmte Weichen-

stellungen merken und auf Wunsch abrufen.

Dazu stellt man die Weichen mit den Tasten 0 bis C in die gewünschte Stellung und drückt dann die SPEICH und eine Taste 0 bis F. Die Position ist gespeichert. Nun kann man die Weichen wieder verstellen und drückt dann die Taste START und eine Taste 0 bis F. Dabei muß man jetzt die gleiche Ziffer drücken, die man beim Speichern angegeben hat. Dann wird die alte Weichenstellung geladen und alle Weichen dementsprechend gestellt. So kann man z.B. bestimmte Fahrstrecken vorprogrammieren, oder, wenn man auch Signale verwendet, einen NOTHALT.

Das Programm kann sich 16 Positionen merken.

Hier noch ein wichtiger Hinweis: Da alle Magnete starke Störungen beim Ein- und Ausschalten hervorrufen können, muß man besonderen Wert auf eine gute Leitungsführung und Erdung legen. Man sollte auch kleine Kondensatoren (ca. 100nF) über alle Kontakte schalten, um die Funkenerzeugung so klein wie möglich zu halten.

Die Software sorgt übrigens dafür, daß Funken nur bei den Relais RL 1 und RL 2 auftreten können. Wenn man nicht ausreichend entstört hat, kann es vorkommen, daß beim Schalten der Weichen der Computer auch gleich aussteigt. Das Programm sollte man daher zunächst einmal ohne Weichen, nur mit der REL-Baugruppe testen und dann unbedingt irgendwo abspeichern, bevor es ernst wird.

```

;*****
; Eisenbahnsteuerung mit der REL-Baugruppe
; mit Hilfe von HEXMON und HE110
;
; Rolf-Dieter Klein B60107 1.1
;
; Mit dem Programm kann man 12 Magnet-
; artikel (Weichen oder Signale mit je
; zwei Magneten) bedienen.
;*****
aseq ; absoluter Code.

0000
0004 anzeige equ 9 ; las Verzögerung und Anzeige
000C holetaste equ 0ch ; Einlesen einer Taste.
000F tonus equ 0fh ; In Zahl umwandeln.
;
0050 rel equ 50h ; Mit DIL-Schalter einstellen.

org B100h ; Start bei HEXID mit HEXMON

; Hauptprogramm - Start.
start:
call clear ; alle Weichen neutral stellen
call schalte ; und ausführen.
schleife:
call holetaste ; Menue und Auswahl.
cp 47h ; BEFEH, dann Weichen neutral
jr z,start ; stellen.
cp 4bh ; START (in) ;dann eigene Tabelle laden
jr c,ladetab ;
cp 2bh ; SPEICH (in) ;drucken, dann Positionen merK
jr z,meripos ;
call tonus ; Taste 0..9,A..F
jr c,schleife ; keine Num. Taste, dann löschen alles.
ld hl,weicht ; Adresse der Tabelle laden
cp 0ch ; 0..B erlaubt, Rest ungueltig.
jr nc,schleife ; nicht ausführen.
and 0fh ; Index darauf addieren
ld b,0 ;
ld c,a ;
add hl,bc ; hi=Zeiger auf Weichenstellung
ld a,(hl) ; dann komplementieren
cpl ; und Wert ablegen
ld hl,a ; Weiche kann man nun umschalten.
call schalte ;
jr schleife ; weiter in Schleife.

ladetab: ; Positionen von einer Tabelle
call holetaste ; Dann 0 bis F drucken
call tonus ; als Positionsmarker
jr c,schleife ; nein, dann zurueck.
call null2 ; hl ist dann Startadresse
ld de,weicht ; und in die aktuelle
ld b,12 ; Tabelle uebertragen
ld hl,ladetab ;
ld a,(hl) ; neuen Wert von Merke-Tabelle holen
ld (de),a ; und in aktuelle Tabelle speichern
inc hl ; neue Position bilden
inc de ;
djnz lad ; Wiederholen bis alle 12 Weichen gestellt.
call schalte ; Weichen stellen.
jr schleife ; dann zurueck.

meripos:
call holetaste ; Dann 0 bis F drucken
call tonus ; als Positionsmarker
jr c,schleife ; nein, dann zurueck.
call null2 ; hl ist dann Startadresse
ld de,weicht ; und in die aktuelle
ld b,12 ; Tabelle uebertragen
ld hl,meripos ;
ld a,(hl) ; Wert aus aktueller Tabelle laden
ld (hl),a ; und in Merke-Tabelle ablegen.
inc hl ;
inc de ;
djnz merip ;
jr schleife ; dann zurueck.

null2:
mul12: ; aktu*12 + Tabellensadresse
ld c,a ;
ld b,0 ; Index * 12
push bc
pop hl
add hl,hl ; *2
add hl,bc ; *2 * x
add hl,hl ; (*2 * x) * 2
;
; Unterprogramm, dass die Weichen stellt
; Dabei steht in Akku der Wert i oder 3
; je nachdem, welche Weichengruppe
; geschaltet werden soll.
;
; A=1 oder 3, je nach Schaltgruppe.
push'af
call del100 ; warten bis andere Relais ein.
pop af
cutlrel'a ;
call del100 ; 100 ms warten
ld a,0 ; RL 0 ausschalten
out (rel),a ;
ld a,2 ; RL 1 auch
out (rel),a ;
call del100 ; warten bis aus.
ret ; Unterprogramm Ende

del100:
ld b,100 ; ca. 100 Millisekunden
warte:
call anzeige ; verzoeget um las
djnz warte ; 100 Mal durchlaufen
ret

; Unterprogramm stellt die Relais gemess
; der Tabelle.
; hl=Adresse von 6 Speicherzellen
; die die Stellung (0,0fff) beinhalten.
einsteil:
ld b,a ; Anzahl der Relais
ld c,4 ; Stellindex RL 3.. RL 8
einlp:
ld a,(hl) ; Stellung prüfen
or a ; 0 dann weiter
jr nz,ein1 ; < 0 dann
ld a,c ; Relais aus
jr ein2 ; und weiter
ein1:
ld a,c ; <0, also
add a,1 ; Relais an
ein2: ; hier weiter.
out (rel),a ; und ausführen
inc c ; neuer Code
inc c ; in 2er Schritten.
inc hl ; neue Position.
djnz einlp ; bis alle Positionen aktuell
ret

; Unterprogramm stellt alle Weichen
; auf aktuellen Stand gemess der Speichertabelle.

schalte:
ld hl,weicht ; Relais Gruppe 1 vorbereiten
call einsteil ;
ld a,1 ; Gruppe 1 schalten
call pulse ;
ld hl,weicht2 ; Relais Gruppe 2 vorbereiten
call einsteil ;
ld a,3 ; Gruppe 2 schalten
call pulse ;
ret ;
weicht1:
defb 0,0,0,0,0,0 ; Gruppe 1
weicht2:
defb 0,0,0,0,0,0 ; Gruppe 2

tabelle: ; Stellungen, die bei Druecken
; von START eingenommen werden soll.
; 16 verschiedene Felder moeglich
end

```

Bild 2: Z80-Einsteigerpaket steuert Eisenbahnweichen über HEXMON und REL-Baugruppen

Herb

Computer zum Anfassen.

Unseren besonderen Dank und Anerkennung richten wir einmal an eine Bank, die auch Ihnen bekannt ist, und zwar die Bayerische Vereinsbank. Sie fragen sich sicher, wo hier die Verbindung besteht! Dies versuchen wir Ihnen im folgenden klarzulegen, damit Sie eine Bank auch einmal aus einer anderen Perspektive sehen und kennenlernen.

Die BV und GES

So wie die Überschrift dieses Artikels, so lautet auch eine Ausstellungsserie der Bayerischen Vereinsbank (BV). Es ist eine Wanderausstellung, die nun schon im 3. Jahr bundesweit durchgeführt wird. Jeden Monat findet sie in einer anderen Filiale der BV und somit auch in einer anderen Stadt. Diese Ausstellung ist für das Jahr 1990 wiederum komplett ausgebucht. "Computer Zum Anfassen" bedeutet zunächst einmal, daß für die einzelnen Filialen zusätzliche Arbeit, Engagement und persönlicher Einsatz gefordert wird. Hierzu einiges im Einzelnen später.

Das Motto dieser Ausstellung kann man ruhig wörtlich nehmen, denn für die breite

Jeder hat seine Ideen, Vorstellungen und Wünsche, die Technik "COMPUTER", sinnvoll einzusetzen, um dadurch eventuell mehr Zeit im privaten Bereich zur Verfügung zu haben. Es ist der Drang zur Bequemlichkeit; oder "der Zahn der Zeit" der uns alle zur Eile zwingt und uns mit Hektik segnet, sodaß oft ein Computer hilfreich erscheint. Trotzdem nehmen sich viele, die sich damit beschäftigen die Zeit, Artikel zu schreiben, die wir dann zum Beispiel veröffentlichen. An dieser Stelle möchten wir unseren aufrichtigen Dank und ein großes Lob an all diejenigen aussprechen, die hier privates Engagement zur Entfaltung bringen.

Öffentlichkeit ist damit die Möglichkeit gegeben, einen Computer aus der Nähe zu sehen, sein Innenleben und die technischen Details anzusehen und auch einmal *anzufassen*.

Der ausschließliche Zweck dieser Ausstellung von NDR-Computersystemen wird darin von der BV natürlich nicht gesehen. Sie verfolgt damit noch andere, idealere Ziele. Die BV hat es sich zur Aufgabe gemacht, die Funktionsweise und das Innenleben eines Computers an *jedermann* heranzutragen, um ein gewisses Grundwissen zu vermitteln.

So wird im Einzelnen erklärt was eine CPU ist, was der Unterschied zwischen ROM und RAM ist, was es auf sich hat mit Takt, Quarz und so weiter. Natürlich wird auch

der Aufbau von Massenspeichern erläutert: Begriffe wie interner- und externer Speicher oder der Aufbau einer Diskette und deren Format, sowie auch die Funktion einer Festplatte werden direkt am Objekt verdeutlicht.

Daß hier tatsächlich grosser Nachholbedarf besteht, können die Vortragenden täglich spüren. Eine Ausstellung wird im Mittel auch von 20 bis 30 Schulklassen besucht, es stellt sich dann sehr bald heraus, daß es immer wieder die selben grundlegenden Fragen sind, die einer Beantwortung bedürfen...

Die Firma Graf-Elektronik-Systeme hat sich auf dem Gebiet der Aus- und Weiterbildung bekanntermassen eine langjährige Erfahrung erarbeitet. Deshalb liegt hier die enge Zusammenarbeit mit der Bayerischen Vereinsbank nahe

Wie sieht die Organisation der BV-Ausstellung aus

Wir von GES übernehmen den Aufbau, die Einarbeitung eines Mitarbeiters der BV und die Führung von Schulklassen an zwei Tagen.

Oft ist auch noch am 1. Abend der Ausstellung ein Zusammentreffen geladener Gäste vorgesehen, die aus verschiedenen Bereichen der Klein- und Mittelständischen Betriebe sowie aus dem Schulbereich (Lehrer) kommen. Dieser Personenkreis nimmt einerseits natürlich an der allgemeinen Führung teil, zeigt andererseits besonders großes Interesse, und sind in der abendlichen, privaten Atmosphäre gerne bereit, über ihre Eindrücke, Erfahrungen und Probleme zu sprechen und zu diskutieren.

Für den Organisator der jeweiligen BV-Filiale mag es oft sehr schwer sein, das geeignete Publikum zusammen zu stellen, um eine rege Diskussion zu erwirken. Hier hat sich bisher das Fingerspitzengefühl der Bankier gegenüber ihren Kunden bezahlt



gemacht, denn mit Übersicht, Menschenkenntnis und Erfahrung waren diese Abende immer ein großer Erfolg. Die Leistungen der einzelnen BV-Filialen wollen wir an dieser Stelle hervorheben, denn es ist schliesslich *kein Muß* von Oben, also von der Hauptzentrale in München, sondern eigenes Interesse der einzelnen Niederlassungen der Bayerischen Vereinsbank. Diese müssen sich um die Ausstellung "Computer zum Anfassen" bewerben. Wenn dann die Zusage von München erfolgt, so kann in feinfühleriger Weise organisiert werden.

Wo werden die Computer aufgestellt, denn der tägliche Ablauf des öffentlichen Parteiverkehrs darf nicht gestört werden? Es müssen oft vertrauliche Gespräche geführt werden. Informations- und Kassenschalter sollen nicht beeinträchtigt werden. Der Standort wird mit dem Mitarbeiter der Firma Graf abgesprochen, denn es ist manchmal schwer, die Euphorie und Begeisterung der einzelnen Schüler unter Kontrolle zu bringen.

Die nächste Frage wird lauten, welche Schulen, welche Klassen und welche Altersstufen sind daran interessiert und können angesprochen werden. Schulen anrufen, Termine für die DEMO koordinieren, Einladungen, Anschreiben absenden. Es gilt zu versuchen, die zuständigen Ausbildungsleiter in Betrieben zu erreichen, um mit ihnen und deren Schützlingen eine Teilnahme an der Ausstellung zu vereinbaren.

Es gilt die Verantwortlichen im Schul- und Ausbildungsbereich für den ersten Abend als Gäste zu laden. Bei allen Bemühungen



zur Computerwissens-Vermittlung sollen auch die bankspezifischen Interessen damit verbunden werden, indem den Gästen Neuigkeiten und Aktivitäten aus dem Finanzbereich angekündigt und weitergegeben werden. Dies Alles ist mit enorm viel Zeitaufwand und natürlich auch mit Unkosten verbunden. Hier hat die Bayerische Vereinsbank keine Mühen und Kosten gescheut, denn die verschiedenen NDR-Computer-Systeme sind Eigentum der BV.

Alle mittel- und unmittelbar damit betroffenen Personen von GES können ob der Zusammenarbeit mit der BV nur Lobesworte aussprechen, und die hervorragende Organisation erwähnen. Die letzten 3 Ausstellungen sind unserem Vertriebsleiter Herrn Herb noch in bester Erinnerung.

Es war dies die BV in Lauf a.d. Pegnitz, Hof und Aschaffenburg. An diesen Orten entstanden Fotos zu diesem Artikel.

Zusammenfassend möchten wir hiermit feststellen, daß diese Ausstellung nicht dazu gedacht ist, viele Kunden in die einzelnen Geschäftsbereiche der Bayerischen Vereinsbanken zu bekommen, sondern die Vorstellung und fehlendes Wissen über Computer für Ausbildung und Beruf weiter zu vermitteln. Es sollen die Verantwortlichen auf die Möglichkeiten zur sinnvollen Investitionen in Computer- Soft- und Hardware im Ausbildungsbereich aufmerksam gemacht werden.

Hier nochmals unsere Anerkennung an die BV und weiterhin viel Erfolg!

Lagerräumung - Aktuelle Sonderangebote

NDR-Computer, Bausätze

70224 NP_CPU280	2.Wahl CPU280 Platine mit kleinen, meist markierten Fehlern; für Bastler klein Problem, mit Lötstopp und Bestückungsdruck	10,00
70155 NP_EPF	Eprom-Floppy Platine - 2	20,00
70180 NP_FLO2	II.Wahl FLO2 Platine mit kleinen, meist markierten Fehlern; für Bastler kein Problem, mit Lötstopp und Bestückungsdruck	10,00
70230 NP_GDP64K	GDP64K-Leiterplatte R4 neu komplett Bestückungsdruck und Lötstopp	20,00
70181 NP_HEX10	II.Wahl HEX10 Platine mit kleinen, meist markierten Fehlern; für Bastler kein Problem, mit Lötstopp und Bestückungsdruck	10,00
70159 NP_POW26/22P	POW26/22 Platine	2,00
70153 NP_RAM16	RAM16 Platine	10,00
70233 NP_RAM256T	RAM256 Platine, teilbestückt mit Sockeln und passiven Bauelementen, Platine neuwertig	50,00
70189 NP_ROA16	ROA 16 Platine	25,00
70223 NP_ROA64	2.Wahl ROA64 Platine m.Kl.Fehlern	10,00
70190 NP_ROB	ROB Platine	15,00
70097 NP_SBC3	SBC3 Platine Rev. 2	20,00
70096 NP_SER	SER Platine Rev. 3	20,00
70193 NP_SPRACHE	SPRACHE Platine	25,00
70222 NS_UPDATE138	Update nur Modula-Compiler 38	100,00
70194 NX_VIDEOBETA1	Video BETA Teil 1+2	206,34
70195 NX_VIDEOVCC	Video VCC Teil 1+2	173,28

Sirius-Computer

70084 SS_FLBU2000	Finanzbuchhaltung Markowitsch für Victor Sirius Festplatten-Rechner auf 51/4", voll lizenzierte Version, ohne Handbuch	570,00
70046 SZ_SIRIUS_DISC	Victor Sirius m. 2 51/4" Laufwerken Diskettenlaufwerke mit hoher Kapazität, komplett mit Tastatur und Bildschirm	1596,00
70045 SZ_SIRIUS_PLATTE	Victor Sirius m. Festpl.u.l Laufw. 51/4" Diskettenlaufwerk mit hoher Kapazität, komplett mit Bildschirm und Tastatur	2166,00
70019 SZ_VICTOR_PORTABL	Victor Sirius Portable	2052,00
Diverse Artikel		
70216 SA850	Shugart SA851, DS/DD, 8" gebraucht	489,99
70191 SCHATPLAN	Schaltpläne und Unterlagen	10,00
70147 SO24POL	24 poliger Sockel	0,82
70010 TX_TA_SCHREIBMASC	TA Typenradschreibmaschine mit einzeliliges Korrekturdisplay SE1020C, komplett mit Originalunterlagen und PC-Interface	3306,00
70007 TZ_P3	Alphatronic P3 CP/M-Computer mit 2 Laufwerken hohe Kapazität, Komplett mit Tastatur	570,00
70201 XB_FLO1B	FLO1 Bausatz	268,99
70154 XF_CRT1F	CRT1 Fertiggerät	150,00
70202 XF_FLO1F	FLO1 Fertiggerät	359,00

Lagerräumung - Aktuelle Sonderangebote

NOR-Computer, Bausätze	70089 NF_KEY	51,30	70145 NF_S20	999,00
70148 NB_BANKSEL 70094 NB_BUS4	30,00 198,00	Tastaturentroller Anschluß Parallel-Tastatur, komplett bestückt, betriebsbereit Rev. 3	70145 NF_S20	999,00
70143 NB_C64ADAPT 70196 NB_EPROM 70148 NB_F0W26/22B1 70181 NB_F0W26/22B2 70151 NB_RAM16 70187 NB_ROM16 70192 NB_SPRACHE	40,00 109,00 39,00 55,01 20,00 40,00 259,00	70119 NF_KEY2 70256 NF_MUSIK 70157 NF_F0W26/22F1 70160 NF_F0W26/22F2 70247 NF_PROMERK 70150 NF_RAM16 70111 NF_RAM256 70095 NF_RAM64 70249 NF_REL 70188 NF_ROM16 70237 NF_ROM64	129,00 29,00 68,00 97,00 97,99 30,00 198,00	ohne Netzteil aber komplett verdrahtet mit CPU68K, BUS3, IOE+GEN, 4xROM64, KEY, GDR64K, 8x18, EASS0-1 Version 4.3, EMOGIDOS, EFRAGE, FILM, EHARDCOPY, ED10, EQUICKFROM, EFRAGEV, EDROM, EFILM1, ECELIST, EMOSSICOMP, EDATED DEMO-System aus Hamburg 280/CP/M GENI, NEZ, 2xBUS2 gekoppelt, 2xTEACI 80 Spuren, IOE+GEN, FLO2, CPU280, 2xRAM256, BANKBOOT, KEY, GDR64k, EFLORONGS, 2x88, komplett verdrahtet und betriebsbereit Festplattenschnittsteller SASI/SGSI Laborgerät CPU SB22 Festiggr., teilw.m.Eprom
NOR-Computer, Eproms	29,00	RAM 64 komplett m. 64Kb Speicher Relais-Baugruppe geprüft, Master ROM16 Fertigerüst ROM64 Fertigerüst ohne Speicher Laborgerät, fast neu Gehäuse 2 mit eingeb. Netzteil NEZ mit BUS4 - komplett verkabelt GEH4 (vom mc-modular-AT) Netzteil, BU5E088, komplett verdrahtet - Ideal f. ECB-System GEH2 mit eingebautem BUS4 komplett verkabelt (kein Netzteil) GEH4 komplett m. eingebautem NEZ ohne BUS GEH3 NOR-Gehäuse mit Klappdeckel mechanisch nicht ganz vollständig, leer	70226 NF_SBC2 70227 NF_SBC2A	99,00 129,00
NOR-Computer, Fertigerüste und Gehäuse	49,00 79,00 69,00	70126 NF_S01 70127 NF_S02 70128 NF_S03 70129 NF_S04 70130 NF_S05	398,00 398,00 298,00 298,00 97,99	CPU SB22 Fertigerüst mit aufgebauter Verkehrsampel und IOE CPU SB22 Fertigerüst m.aufgeb.Musik SBC3 Fertigerüst R2, Laborgerät Serielle Schnittstelle SOUND Labormaster-Fertigerüst Tastaturteil oh.Gehäuse-Vorführmodell Tastaturteil komplett m. Gehäuse und Kabel - Vorführmodell TAST1 komplett mit Gehäuse u. Kabel umgewandelt werden TAST2 Ausstattungsstück geprüft Uhrenbaugruppe R2 kompl.m.Akku Laborgerät C64ADAPT Handbuch Eprom-Floppy Handbuch FLO2 Handbuch RAM16 Handbuch 280 Aufbauanleitung II.Wahl Bankboot Platine II.Wahl MOT3 Platine II. Wahl RAM 16 Platine für 16K RAM-Baugruppe mit RAM-IC's 2114 auf SNP-BUS Bankboot Platine R2, neu BANKSEL Platine Neue BUS2 Platine R1 Lehrbuche 5V einlöten falls Stecker 2 benutzt wird NOR BUS3 Platine eventuell mit Kurzschlüssen BUS4 Platine Rev. 1 NOR BUS4 Platine eventuell mit Kurzschlüssen C64ADAPT Platine Platinensatz COL256 3PI-neuertig II.Wahl CPU68K Platine mit kleinem, meist markierten Fehler; für Basler kein Problem, mit Lötlott und Beutungsdruck
70235 NF_COL256A 70115 NF_CPU64180 70234 NF_CPU6800X 70092 NF_CPU68K 70252 NF_CPU8088 70118 NF_CPU280 70257 NF_DA 70250 NF_ETNAUST 70090 NF_FLO2 70113 NF_FLO20 70091 NF_GDR64K 70239 NF_HCOFY 70115 NF_HEX10 70242 NF_HEX102 70229 NF_HEX10K 70244 NF_IBMGP 70253 NF_IOE2 70116 NF_IOEGER	398,00 149,00 150,00 147,99 198,00 89,00 49,00 159,00 171,00 149,00 198,00 69,00 139,00 149,00 149,00 99,00 99,00 89,00	70131 NF_S06 70132 NF_S07 70133 NF_S08 70134 NF_S09 70135 NF_S10 70136 NF_S11 70137 NF_S12 70138 NF_S13 70139 NF_S14 70140 NF_S15 70141 NF_S16 70142 NF_S17 70143 NF_S18 70144 NF_S19	398,00 398,00 398,00 379,00 398,00 298,00 398,00 398,00 398,00 398,00 398,00 348,00 198,00 249,00 449,00	CPU SB22 Fertigerüst Netzteil, BU5E088, komplett verdrahtet - Ideal f. ECB-System GEH2 mit eingebautem BUS4 komplett verkabelt (kein Netzteil) GEH4 komplett m. eingebautem NEZ ohne BUS GEH3 NOR-Gehäuse mit Klappdeckel mechanisch nicht ganz vollständig, leer GEH2 NDR, eingebaut BUS4, NEZ, Lüfter komplett verdrahtet GEH2 NDR mit eingebautem Leistungsstarkem Grelagelat-Netzteil, BUS4, komplett verdrahtet. GEH2, eingeb. NEZ, BUS4, kompl. verdrahtet. GEH2, 2xBUS3 nebeneinander, NEZ nicht verdrahtet GEH2 NDR, NEZ eingebaut, verdrahtet GEH2, eingebaut NEZ und BUS3 GEH2, eingebaut NEZ, BUS4, Lüfter komplett verkabelt GEH2, eingebaut NEZ, BUS4, Lüfter komplett verkabelt GEH2, eing. NEZ, BUS4, kompl. verkabelt GEH2, vorbereitet f. 68020-Ausbau mit BUS4, BUS3, ohne Netzteil, Verkabelung für Netzteil vorgehen. eingebautes BUS4 und Leistungsstarkes Grelagelat-Netzteil, jedoch ohne Deckel Gehäuse 1 mit Netzteil 2 Ideal für ECB-System, da Grelagelaten eingebaut. GEH1, tragbar m. Bügel, eingeb. 3xBUS2 Gehäuse 1 komplett verdrahtet mit eingebauten Buchsen, BUS3, 2xTEACI 80 Spuren Laufwerk 51/4"
70244 NF_IBMGP 70253 NF_IOE2 70116 NF_IOEGER	99,00 99,00 89,00	70143 NF_S18 70144 NF_S19	249,00 449,00	zwei Baugruppen mit Halbleitern IOE2 Labormaster-Fertigerüst fertig aufgebaut, ohne Lötlott E4 IOE-Fertigerüst mit aufgebauter Centronics Schnittstelle - also komplette Druckeransteuerung

Die neuen Labtops von TopLink

Artikel	Technische Daten	Preis incl. MWSt
TL 3140	<p>Netzunabhängig, Akku wechselbar, CPU 80286 / 10-12 MHz, VGA-LCD-Display (640 x 480 x 16), 1 MB RAM, HD: 40 MB / 27 ms, LW: 3.5" / 1.44/720, 2 x SER, 1 x PAR, Tastatur herausnehmbar. Extern anschließbar: VGA-Monitor, 5 1/4" - LW, MF2-Tastatur, Expansionsbox. Gehäuse weiß, Gewicht 7.5 kg. Best. Nr. 41231.....</p>	DM 6726,—
TL 3240 N	<p>Netzabhängig, CPU 80286 NEAT / 16 MHz, EGA-Plasma-Display, 16 Graustufen, 2MB RAM, HD: 40 MB / 27 ms, LW: 3.5" / 1.44/720, 2 Steckplätze frei (1 x AT - 1 x PC), 2 x SER, 1 x PAR. Extern anschließbar: EGA-Monitor, 5 1/4"-LW, MF2-Tastatur. Gehäuse grau, Gewicht 8.5 kg. Best. Nr. 41070.....</p>	DM 8800,—
TL 3240 V	<p>Netzabhängig, CPU 80286 / 20 MHz, VGA- Plasma-Display (640 x 480 x 16), 2MB RAM, HD: 40 MB / 27 ms, LW: 3.5" / 1.44/720, 2 Steckplätze frei (1 x AT - 1 x PC), 2 x SER, 1 x PAR. Extern anschließbar: VGA-Monitor, 5 1/4"-LW, MF2-Tastatur. Gehäuse grau, Gewicht 8.5 kg. Best. Nr. 41232.....</p>	DM 8900,—
TL 5600-40	<p>Netzabhängig, CPU 80386 / 25 MHz, VGA/EGA- Plasma-Display (640 x 480 x 16), 2MB RAM, HD: 40 MB / 27 ms, LW: 3.5" / 1.44/720, 2 Steckplätze frei (1 x AT - 1 x PC), 2 x SER, 1 x PAR. Extern anschließbar: VGA/EGA-Monitor, 5 1/4"-LW, MF2-Tastatur. Gehäuse grau, Gewicht 8.5 kg. Best. Nr. 41233.....</p>	DM 10900,—
TL 5600-100	<p>Netzabhängig, CPU 80386 / 25 MHz, VGA/EGA- Plasma-Display (640 x 480 x 16), 2MB RAM, HD: 100 MB, LW: 3.5" / 1.44/720, 2 Steckplätze frei (1 x AT - 1 x PC), 2 x SER, 1 x PAR. Extern anschließbar: VGA/EGA-Monitor, 5 1/4"-LW, MF2-Tastatur. Gehäuse grau, Gewicht 8.5 kg. Best. Nr. 41201.....</p>	DM 17500,—

GRAF
computer

Alle Preise sind freibleibend ab Lager Kempten
Graf Elektronik Systeme GmbH, Magnusstr. 13, 8960 Kempten
Tel.: 0831-6211 Fax: 0831-61086

Ralph Dombrowski

Ausgabe von Texten und Zeichen über die serielle Schnittstelle

Da man die CO2-Routine, die für die Ausgabe über viele Schnittstellen vorgesehen ist, auch umlenken kann, ist es möglich, sie so umzulenken, daß die Textausgabe auch über einen seriellen Drucker funktioniert.

In der LOOP 8/9 wurde schon erklärt, wie man die CO2-Routine umlenken kann; deshalb möchte ich nicht näher darauf eingehen. Wenn sie umgelenkt ist, benötigt

man nur ein passendes Programm, das die Daten an die serielle Schnittstelle leitet. Außerdem muß diese vorher eingestellt werden. Das Listing des Programms er-

klärt sich selbst, weshalb weiterer Kommentar eigentlich überflüssig ist. Ich wünsche viel Spaß beim Betrieb des Druckers.

<pre> userinit: move.b #\$1e,d0 move.b #\$0b,d1 move #!siinit,d7 </pre>	<pre> * Lenkt die CO2-Routine auf die * serielle Schnittstelle um und * initialisiert diese. * Stellt die Schnittstelle auf: * 9600 Baud, 8 Bit, keine Parität, * 1 Stop ein * Freigabe * Stellt die gewünschten Werte * ein, falls andere Einstellungen * gewünscht sind, so kann im * Handbuch der SER-Karte oder * im Handbuch zum Grundpro- * gramm nachgelesen werden. Es * sind viele Einstellungen mög- * lich. </pre>	<pre> trap #1 move.l #so,d0 move #4ef9,\$24(a5) move.l d0,\$24+2(a5) move #!usr,d7 trap #1 rts </pre>	<pre> * Holt die Adresse des * Unterprogramms für die serielle * Schnittstelle nach d0 * Befehl JMP.l * Adresse SO. * Schaltet die CO2-Routine auf USER- * Berteib um. Beim Aufruf von CO2 * wird jetzt der Sprungvektor bei * Adresse \$24(a5) abgesprungen, * welcher die Routine für die * Zeichenausgabe anspricht. * Der Befehl CRT schaltet wieder auf * Bildschirmausgabe um. </pre>
---	---	---	--

Kleinanzeigen

Verkaufe preisgünstig: NDR-Klein-Computer 68008, betriebsbereit mit Preh-Tastatur, bernst.-farb. Monitor, 2 x 5 1/4"-Laufwerke, 1 SER, 5 ROA und 1 RAM64/256 (alle bestückt), GDP64k, PROMER r3, Maus/Hardcopy, BUS2, 2 IOE, Netzteil 60W, Modula-2 Compiler u. andere Software sowie Zeitschriften und Handbücher. Helmut Steinecke, Graudenzer Str. 5/2, 7250 Leonberg, Tel.: 07152/47505.

Anfänger: Verkf. Monitor, Key, SBC2, GDP64k, Key-Print, BUS3A, NE3, Softw. Preis VHS, mögl. komplett (alles neuwertig). Horst Pries, Lindenstr.4, 24 Lübeck 1, Tel.: 0451/81511

Verkaufe gegen Gebot: SBC3, PRIMER, CAS, RAM64/256, POWER5V, SBC3. Platinen teilbestückt: Seriell, AD10x1, CPU, ROA64, BankBoot, FLO3, Eprom-Löschgerät, Druckerpuffer 64 kB, Flachb. Plotter HPX84. Karlheinz Hahn, Kreuzlach 19, 8806 Neudettelsau

Verkaufe: NDR-Comp. Platinen vollbestückt, Tastatur, Drucker usw... Info bei Werner Link, Franz-Beer-Str. 19, 7987 Weingarten Tel.: 0751/52914

Verkaufe: CPU8088 200,—, ROA256/1M bestückt mit 256 kB 500,—, BUSKOPP 150,—, FLO3 teilbestückt 100,— alles neuwertig. Hinrich Garrels, Zembergweg 20, 7730 Villingen 24

Verkaufe wegen Systemwechsels: SBC3, GDP64k, IOE, IOE2, KEY, CAS, POW5V, BUS3 A, ROA64 voll best., ROA64/256 voll best., TAST2, Monitor (bernst.), Datasette, EFLOMON, EBASIC, EZEAT, EGRUND2, sowie Literatur kompl. DM 400,- Rudolf Detzer, Laerchenstr. 2, 8261 Teising, Tel.: 08633/1507

Kontaktanzeige:

Suche Kontakt zu Z-80 Anwendern im Kreis Bergheim (Erftkreis) zwecks Erfahrungsaustausches. Suche ebenfalls ein Fractalprogramm. Bitte melden bei Dietmar Simons, Gladbacher Str. 261, 5013 Elsdorf, Tel.: 02274/4641 (ab 18:00 Uhr)

Verkaufe: NDR-Computer, 68008 Processor, mit SER, PAR. und Maus, 2xFloppy 5,25 und Schaltnetz., 640 KRam und div., komplett aufgeb. im Gehäuse mit Software und alle Unterlagen u. Datenbücher. Tel.:05821/3629

Verkaufe: NDR-Comp. in PC-Gehäuse, FLO3 mit zwei 5 1/4" FD55 FR; ROA64 incl. GRUND 6.0; ROA256/1M, 192k bestückt; GDP64HS; KEY3 mit ATARI-Maus; KEY1 mit TAST3; SOUND; IOE; IOE2; SER; 18 BUS-Plätze; S-Netzteil 150W; FBAS-Monitor grün; mit JADOS V3.01, DRAW V1.4, RIP V3.4, DATEI2 V2.2, SOUD2, u.a. VB 1200,- DM.

Funktionsfähige Baugruppen: CPU8088 incl. BIOS/PALS 90,-; CPUZ80 25,-; Bank-Boot mit FLOMONCG V1.4 40,-; Ohne ICs: GDP64k 30,-; CAS 20,-; COL256 Leerplatinen 40,-; Jeweils incl. Handbuch, ICs in Präzisionssockel, Preise VB. CP/M 2.2-Software: Wordstar/dBase/HEBAS/ u.a. incl. Dokumentation komplett für 50,- Srefan Uhlmann, Fuchsstraße 35, 8510 Fürth, Tel.: 0911/722300

Klaus Rumrich

Schaltungskorrekturen für NDR-Systeme mit CPU68020

CPU68020

Mit Pull-Up Widerständen wurde hier sehr großzügig umgegangen: Jeder unbenutzte Gateingang hat seinen eigenen Widerstand. Allerdings ist R20 gar nicht mit VCC verbunden. Außerdem fehlt bei Jumper 1 am rechten Anschluß die Verbindung zu VCC.

Signalleitungen, die die 680xx-Prozessoren nicht verwenden (z.B. -M1 oder -RFSH) sollten über einen Widerstand mit VCC verbunden werden, da z.B. die COL256 diese Signale benutzt.

Schließlich sind im Handbuch bei der Bedeutung des Jumpers 4 die Wartezyklen für Speicher- und I/O-Zugriffe vertauscht. Richtig ist, daß die obere Reihe für I/O und die untere Reihe für Speicher zuständig ist.

RAM64/256 r4, r5

Bei dieser Baugruppe wird, anders als im Handbuch beschrieben, immer ein Wait-Signal erzeugt und nicht nur bei angesprochenen Karten. Auch das zweite -RAS-Signal wird immer erzeugt.

Ich möchte hier beschreiben, wie der Fehler bei Verwendung der 256 KBit-Bausteine behoben werden kann. Werden 64 KBit-Bausteine verwandt, so können die Gatter in J21 etwas anders verknüpft werden (siehe Bild 1).

Die Korrektur erfolgt mittels der vorverdrahteten Jumper unter J18 und J19. Die Brücken werden unterbrochen, Stifteleisten eingelötet und die Steckbrücken nach rechts gesetzt. Dies alleine genügt allerdings noch nicht, da nun ein Wait bereits beim Anlegen der richtigen Adresse auch ohne das -MREQ generiert wird.

Dies ist besonders gefährlich bei den CPUs 68000 und CPU68020 mit getrennten Bussen. Bei einem Byte-Zugriff wird die Adresse an alle Bushälften oder Busviertel ausgegeben, aber nur ein Busteil erhält auch ein -MREQ. Dies führt dazu, daß die eigentlich nicht selektierte(n) Karte(n) einen Wait erzeugen, der nicht wieder zu-

Herr Klaus Rumrich ist ein 'altgedienter' Softwarepartner von GES und insofern auch recht kompetent, was die Hardware des NDR-Computers angeht. Da er vorwiegend ein 68020er-System im Einsatz hat, sind ihm auch prompt ein paar schaltungstechnische 'Schönheitsfehler' in der Kombination CPU68020 und RAM256 aufgefallen, wobei er so freundlich war, diese für einen LOOP-Beitrag aufzunotieren: **Beginnen wir mit dem Chef im NDR-System, der...**

rückgenommen wird, weil das Schieberegister J19 wegen des fehlenden -MREQ-Signals nicht laufen kann.

muß nach rechts gesteckt werden. Dann wird die Karte nur noch dann ausgewählt, wenn sowohl die Adresse stimmt, als auch ein -MREQ vorhanden ist.

Der Betrieb am 68020 bleibt gefährlich, wenn der Cache eingeschaltet ist. Wenn die CPU den Cache gefüllt hat und beispielsweise in einer Warteschleife keine Speicherzugriffe mehr

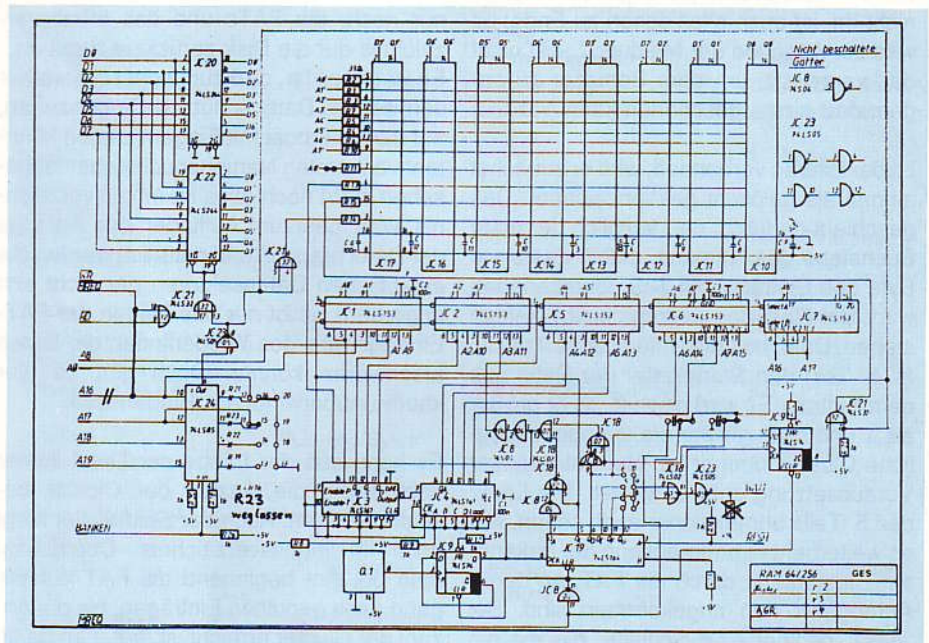


Bild 1. Schaltplan der RAM256 r4 mit sinnvollen Modifikationen zum Einsatz in 68020er-Systemen, nach Rumrich.

Leider sind kaum noch Gatter auf der Karte frei, sodaß ich den Adressvergleicher J24 mitbenutzt habe. Bei Bestückung mit 256er-Bausteinen ist nämlich der Vergleich der Adressleitungen A16 und A17 überflüssig. Man kann also die Verbindung von A16 zu Pin 10 des J24 unterbrechen (Vorsicht: Es sind zwei Unterbrechungen auszuführen und eine Drahtbrücke einzusetzen, da das Signal A16 von J24 weitergeleitet wird!) und statt dessen ein Stück Draht von -MREQ an J24 zu legen. An Jumper 3 darf dann die obere Brücke nicht mehr für 256K eingestellt sein, sondern

benötigt, werden keine -MREQ-Signale mehr erzeugt, die auf der RAM64/256 einen Refresh auslösen sollen.

Besser wäre es, die Refresh-Erzeugung ganz unabhängig von Speicherzugriffen zu machen, z.B. einen auf der Karte erzeugten Takt zu verwenden. Dies bedeutet allerdings einen größeren Aufwand der nur durch zusätzliche Gatter bewältigt werden kann. Genügend Platz wäre aber noch auf der Platine r5, so daß bei einer neuen Revision hier Abhilfe geschaffen werden könnte.

Bei Bestückung mit 256KBit-Bausteinen sollten die Widerstände R21 und R23 nicht bestückt werden, um die Adressleitungen A16 und A17 nicht unnötig zu belasten.

Beim Einsatz von vier RAM64/256 kommt sonst bereits ein Strom 20mA zustande.

Zusammen mit den Eingangsströmen anderer Baugruppen wird dann leicht der maximale Ausgangsstrom der Bustreiber von 24mA überschritten.

Ähnliches gilt für R24, den Pull-Up Widerstand für die Wait-Leitung. Da bereits auf

den CPU-Baugruppen ein Pull-Up Widerstand vorgesehen ist, sollten auf allen anderen Baugruppen die Widerstände weggelassen werden. Der maximale Ausgangsstrom des verwendeten 74LS05 beträgt nämlich nur 8 mA und wird sonst um ein Vielfaches überschritten.

Günter Renner

Alles null und nichtig ?

Patchwork - 7. Teil

Will man eine Datei löschen, beginnt alles mit denselben Vorgängen wie beim Lesen einer Datei (s. 5. Teil): man muß zunächst einen Namen eingeben. Dieser Name wird nach den bereits bekannten Vorarbeiten im Inhaltsverzeichnis gesucht. Ist er nicht da, ist auch alles schon zu Ende; es wird zur Ausgabe der Meldung '... ist nicht da!' verzweigt, um dem Benutzer zu sagen, daß nichts geschehen kann.

Ist der Eintrag vorhanden, wird er zunächst einmal als gelöscht gekennzeichnet. Das geschieht dadurch, daß lediglich der erste Buchstabe des Namens, mithin das erste Byte des Eintrags, mit \$e5 überschrieben wird. Der Rest des Eintrags bleibt einfach stehen. Dann holt die Unteroutine 'firstclu' (s. 5. Teil) den Startcluster der Datei aus dem Eintrag. Es wird geprüft, ob er größer als 1 und nicht größer als die höchstmögliche Clusternummer ist. Nur unter dieser Voraussetzung geht es weiter. Die Leser des 5. Teils ahnen sicher auch schon, wie es weitergeht - immer entlang der bekannten Clusterkette durch die FAT, an deren Anfang wir nun angekommen sind. Bei 'loes1' beginnt eine Schleife, die die ganze Kette bis zu ihrem Ende bearbeitet.

Mit 'nextclu' wird ein Wort aus der FAT geholt, das den gesuchten Eintrag, nämlich die nächste Clusternummer enthält, ins Register d3 geladen und nach d0 gerettet. Die untersten 12 bits werden in d3 mittels UND-Verknüpfung auf \$000 gesetzt und mittels 'putnew' in die FAT zurückgeschrieben, wobei die überschüssigen oberen 4 bits natürlich unverändert bleiben müssen, weil sie zu einem anderen Eintrag gehören! Aus d0 gewinnt man die Nummer des nächsten Clusters, mithin auch die nächste Position in der FAT, die genauso genullt wird. Das Ganze hat ein Ende, sobald ein Wert größer \$ff7 (Endemarke) oder kleiner als 2 auftritt.

Zwei zentrale Funktionen stehen nun noch aus, um DOS-Disketten bearbeiten zu können: das Löschen und das Neuanlegen von Dateien. Die Programme, die diese Aufgaben in unserem einfachen Filemonitor wahrnehmen, heißen 'loesch' und 'speich' und sind diesmal im Listing enthalten.

Damit ist bereits alles geschehen, was zum Löschen einer Datei nötig ist. Es bleibt nur noch, die FATs und das Inhaltsverzeichnis auf die Disk zurückzuschreiben. Es ist ganz klar, daß zumindest bis hierher verbissene Datenschützer Dinge sehen, auf die sie erbost mit Fingern zeigen könnten: vom ersten Namensbuchstaben abgesehen steht noch alles im Inhaltsverzeichnis, was man zum Auffinden des Anfangs der Datei braucht, und natürlich wurden die eigentlichen Datensektoren gar nicht erst angerührt! Es ist nur das Fehlen der FAT-Einträge, das das Wiederfinden der Daten erschweren könnte. Doch gibt es hier kaum unüberwindbare Hindernisse.

Es kann aus der Länge der Datei immer noch leicht die Anzahl der Cluster bestimmt werden. Auch der Startcluster steht noch im Inhaltsverzeichnis. Durchsucht man bei ihm beginnend die FAT aufsteigend nach genullten Einträgen, bis die Anzahl der Cluster erreicht ist, hat man beste Chancen, daß die gelöschten Daten bereits wiederentdeckt sind. Es soll Leute geben, die von dieser Möglichkeit häufiger Gebrauch machen. Sie gehen sehr leichtfertig mit ihren Dateien um, weil sie auf ihre 'second chance' und auf die Utilities eines gewissen Mr. Norton blind vertrauen...

Das Abspeichern neuer Daten geschieht prinzipiell nach genau demselben Schema, nur unter umgekehrtem Vorzeichen. Auch hier wird zunächst einmal das Inhaltsverzeichnis nach einer Datei mit dem eingegebenen Namen durchsucht. Ist eine solche da, so gibt der hier vorgestellte Montitor ziemlich lakonisch die Meldung '... ist schon da!' aus und überläßt es dem Bediener, sie erst zu löschen oder aber einen

anderen Namen zu wählen. Es ist Ansichtssache, hier das automatische und vor allem vorwarnungslose Löschen bzw. Überschreiben einer Datei vorzusehen.

Man muß allerdings unbedingt vermeiden, daß zwei Dateien desselben Namens in einem Inhaltsverzeichnis stehen, und dafür gibt es eben diese beiden verschiedenen Wege.

Teil 1: Der verrückte Bootsektor Loop 17
Teil 2: Eine gefährliche Operation Loop 19
Teil 3: Log. physikalischer Verwirrspiel Loop 20
Teil 4: Sage mir, was Du hast Loop 22
Teil 5: Jetzt wird gelesen Loop 23
Teil 6: Ärger mit Ä Loop 24
Teil 7: Alles null und nichtig LOOP 25
Teil 8: Das Benutzerinterface LOOP 26

Dann gilt es, sich in den Besitz einer freien Position im Inhaltsverzeichnis zu bringen, in die der Name, der Startcluster, die Länge in Bytes und das Attribut der Datei eingetragen wird. Dies sind die absoluten Mindestanforderungen an einen Dateieintrag. Den Namen übernimmt man einfach aus dem Eingabepuffer. Weil nur die FAT über die Belegung der Disk Auskunft gibt, muß

der Startcluster und auch alle weiteren, die benötigt werden, aus ihr entnommen werden; dies besorgt die Routine 'newclu', die den erstmöglichen freien Cluster ausfindig macht. Zum Anlegen der Clusterkette gibt es noch das Unterprogramm 'putnew', das neue Einträge in die FAT schreibt. Und die Länge der Datei? Weil keine andere Quelle

zur Verfügung steht, muß der Benutzer sie von Hand eingeben. Das hat den Nachteil, daß man erst einmal die Länge in Kilobytes umrechnen muß, bietet aber den Vorteil, daß man Dateien beliebiger Art bearbeiten kann. Natürlich wäre es möglich, etwa die Länge von Texten automatisch zu bestimmen - sicher eine sinnvolle Erweiterung der

im 6. Teil beschriebenen Programme zur Textverarbeitung.

Damit wären die Kernroutinen in groben Zügen beschrieben. Was noch fehlt wäre das, was man gemeinhin Benutzerinterface nennt: Ein- und Ausgaben über die Konsole. Sie sind Gegenstand der letzten Folge.

Patchwork-Listing Teil 7		bmi spe6		moveq #2,d1 subq #1,d5	
****Menuefunktionen				*schreiben *fuer DBRA	
loesch:		bsr crlfkon	*neue Zeile	spe5:	*jetzt erst
bsr getname		bsr lookfor	*Datei dieses	bsr cluster	*Daten schreiben
move.b namebuf(pc),d0	*Name eingeben	beq schonda	*Namens schon	dbra d5,spe7	*Laengenzaeher
cmp.b #'.',d0	*darf nicht mit	da?		bsr putfat1	*wenn fertig
ble menue	*<.' beginnen	lea dirbuf(pc),a4	*wenn nicht	bsr copyfat	*alles zurueck
bsr testdis	*Disk ansehen	moveq #112-1,d4		bsr testfat	*alles ok?
bne.s loes4		spe1:		spe6:	*am Ende
loes0:		cmp.b #0,(a4)		bra testflg	*Floppy-Fehler?
bsr lookfor	*ist sie da?	beq.s spe2		spe7:	
bmi nichtda	*wenn ja	cmp.b #5,(a4)	*freien Eintrag	bsr newclu	*naechsten freien
move.b #5,(a4)	*Entry loeschen	beq.s spe2	*im Inhaltsver-	cmp #maxclu,d2	*Cluster suchen
bsr firstclu	*Startcluster	adda.l #20,a4	*zeichnis suchen	ble.s spe5	*weiterrschreiben
cmp #2,d2	*nach d2 holen	dbra d4,spe1		spe4a:	
blt fatfehl	*und pruefen	bra istvoll		lea meld5(pc),a0	*Disk voll?
cmp #5ff7,d2	*Ende?	spe2:		bsr meldung	*dann abbrechen
bgt.s loes3		movem.l a4,-(a7)		bra loes0	*alles loeschen
cmp #maxclu,d2	*moeglich?	lea namebuf(pc),a2	*Eintrag anlegen		
bgt fatfehl		moveq #11-1,d4	*11 Zeichen		
loes1:		spe3:		**** Unterprogramme	
bsr nextclu	*FAT-Eintrag	move.b (a2)+,(a4)+	*Name	putnew:	*neue Clusternr.
move d3,d0	*nach d3 holen	dbra d4,spe3		btst #0,d2	*in die FAT
and #5f00,d3	*retten/nullen	move.b #20,(a4)+	*Attribut	beq.s putne1	*schreiben
bsr putnew	*zurueck damit	moveq #20-1,d4	*Rest	rol #4,d3	*ggf. rotieren
and #5ff,d0	*naechsten	spe3a:		putne1:	
cmp #2,d0	*genauso	clr.b (a4)+	*mit \$00 belegen	move.b d3,0(a4,d4.w)	*lsb
blt fatfehl	*behandeln	dbra d4,spe3a		ror #8,d3	
move d0,d2		movem.l (a7)+,a4		move.b d3,1(a4,d4.w)	*msb
cmp #maxclu,d2	*bis Ende	lsl #2,d5	*Laenge/Bytes	rol #8,d3	
ble.s loes1		move.b d5,\$1d(a4)	*nur msb/lsw	rts	
cmp #5ff7,d2		rol #8,d5		newclu:	
ble fatfehl		move.b d5,\$1e(a4)	*und lsb/msw	move d2,d0	*freien Cl. suchen
loes3:		ror #8,d5	*rotieren wegen		*alte Nr. retten
bsr putfat1	*alles auf Disk	lsl #2,d5	*INTEL-Folge!	newcl1:	
bsr copyfat	*zurueckschrei-	moveq #1,d2		addq #1,d2	*inkrementieren
ben		movem.l a4,-(a7)	*ab FAT-Anfang	cmp #maxclu,d2	*bis Diskende
bsr putdir		spe4:	*entry retten	bgt.s newcl3	*oder
bsr testfat	*alles ok?	addq #1,d2		bsr nextclu	*bis FAT-Eintrag
loes4:		bsr nextclu	*in der FAT	and #5ff,d3	
bra testflg		and #5ff,d3	*ersten freien	and #5ff,d3	
speich:		bne.s spe4	*Cluster suchen	bne.s newcl1	*null wird!
lea meld7(pc),a0	**wieviel kB'	movem.l (a7)+,a4	*DIR-entry zu-	bsr nextclu	*Eintrag holen
bsr meldung		rueck		or #5ff,d3	*neues Ende
bsr getwert	*Eingabe	ror #8,d2	*Cluster-Nr.	bsr putnew	*eintragen
ble menue	*nur > 0!	move d2,\$1a(a4)	*ins DIR eintr.	exg d2,d0	*alten nach d2
cmp #maxclu,d0	*nur < maxclu!	rol #8,d2		bsr nextclu	*dessen Eintrag
bge menue		cmp #maxclu,d2	*moeglich?	and #5f00,d3	*updaten
move d0,d5	*Laenge/kB retten	bgt.s spe4a	*nur dann	and #5ff,d0	
bsr getname	*welche Datei?	bsr putdir	*zurueckschreiben	or d0,d3	
move.b namebuf(pc),d0	*steht ein Name	bsr nextclu	*vorlaeufiges	bsr putnew	
cmp.b #'.',d0	*drin?	or #5ff,d3	*Ende in der	move d0,d2	*ist neuer Cl.
ble menue		bsr putnew	*FAT eintragen	newcl3:	
bsr testdis	*ansehen	movea.l filebuf(pc),a0	*auf die Floppy	rts	

Listing: Routinen zum Erstellen und Löschen von Dateien (Fortsetz. Seite 22)

*Platz suchen beq.s spe2 adda.l #20,a3 dbra d4,spe1 bra istvoll	moveq#11-1,d4 spe3: move.b (a2)+,(a3)+ dbra d4,spe3 move.b #20,(a3)+ moveq#20-1,d4	*Name *Attribut *Rest	dbra d4,spe3a movem.l (a7)+,a3	
spe2: movem.la3,-(a7) lea namebuf(pc),a2	spe3a: clr.b (a3)+	*auf \$00	lsl #2,d5 move.b d5,\$1d(a3) rol #8,d5 move.b d5,\$1e(a3) ror #8,d5 lsr #2,d5	*Laenge/Bytes *nur msb/lsw *und lsb/msw
	*neuer Eintrag			

Listing: Routinen zum Arbeiten mit Dateien (Ende)

Filmreif(e)

Ein Animationsprogramm läßt Trickfilmer-Herzen höher schlagen: Autodesk Animator

Doch schnell kann sich die Lust auf PC-Kino in Frust verwandeln: Scheinbar höchst komplizierte Menüs und neuartiges Vokabular wie "Gel" oder "Drizzle" schrecken zunächst einmal ab. Um die eigenen Werke professionell darzustellen, sollte eine Woche des Trainings vorerst genügen. Das Tutorial-Handbuch hilft dabei kräftig.

Naserümpfen vielleicht auch bei den Besitzern von erweiterten VGA-Grafikkarten: Der Animator läuft ausschließlich im MCGA-Modus mit 320 x 200 Punkten und 256 Farben. Mittlerweile lassen sich jedoch dank der Farbenvielfalt sehr ansprechende Bilder in diesem Modus erzeugen. Ausserdem braucht ein nur 16farbiges Motiv mit 640 x 480 Pixel doppelt soviel Bits wie ein MCGA-Bild und würde den Animator buchstäblich zur Schnecke machen. Da wir schon bei den Hardware-Voraussetzungen sind: Als Mindestvoraussetzung führt das 500seitige Handbuch (Referenz und Tutorial) einen XT mit 8 MHz und VGA-Karte, 640 kByte RAM und eine 10-MByte-Festplatte an. Der Hersteller Autodesk empfiehlt dazu eine RAM-Disk oder Expanded Memory mit 2 oder 3 MByte, um die Effektivität zu steigern.

Die ganze Kino-Maschine bietet neben der Möglichkeit Bilder ablaufen zu lassen, eine Untertitelung mit 18 verschiedenen Schriften, sowie ein komplettes Malprogramm an, dessen Fähigkeiten durchaus an Deluxe Paint PC heranreichen. Bei der Verwendung von Grafik-Formaten kennt die Software nur "GIF"-Dateien.

Wie arbeitet man mit dem Animator?

Die Angst vor dem Menü-Dickicht schlägt bald in Begeisterung um, sobald der Anwender das Grundsystem verstanden hat.

Die erste Begegnung mit dem Autodesk Animator läßt über die professionellen Trickszenen staunen. Kein Wunder, denn im Gegensatz zu Kino oder Fernsehen jagt der Animator 70 Bilder pro Sekunde über den Bildschirm (das Fernsehen dagegen nur 25 Bilder /s). Beim Aufruf der Demo-Datei "Tigercat" läuft zum Beispiel eine Raubkatze mit geschmeidigen Bewegungen über den Bildschirm.

Direkt nach dem Start befindet man sich im "Home-Panel", der Kommandozentrale im Animator. Von hier aus greift der Filmregisseur auf 22 grafische Werkzeuge und 26 Maltechniken sowie auf zahlreiche weitere Funktionen zu. Dieses Hauptmenü bietet schnellen Zugriff auf jedes Bild des gesamten Filmes. Eine Play-Taste startet die Vorführung. Zoom-Funktionen, Farb-Palette und Untermenüs stehen ebenfalls im "Home-Panel" zur Verfügung.

Auf der Bildschirmfläche oberhalb des Home-Panels kann unmittelbar mit dem Zeichnen begonnen werden, falls man es nicht vorzieht bereits abgespeicherte Motive (im GIF-Format) zu laden. Auf der Diskette befinden sich zu diesem Zweck bereits zwei Beipieldateien. Über die Menü-Leiste "PIC" können GIF-Bilder gespeichert, gelöscht und geladen werden.

Nach dem Laden zeigt sich nun das erste Bild auf dem Monitor. Jetzt kann der Anwender in diesem Bild noch "Verschönerungen" nach eigenem Gusto vornehmen und dabei diverse Mal-Techniken austesten. Veränderungen können mit "UNDO" wieder zurückgenommen werden, bzw. mit "RESTORE" der Ausgangspunkt erreicht werden. Jetzt muß der Benutzer Filmlänge und Spielgeschwindigkeit regeln. Im Menüteil "FRAMES" bestimmt er die Anzahl der benötigten Einzelbilder, deren obere Grenze bei 4000 liegt. Natürlich lassen sich zwischen bestehenden Bildern neue einfügen oder bestehende löschen. "PLAY SPEED" regelt die Pause zwischen zwei Einzel-Szenen von 1/70 s

bis 121/70 s.

In diesen Menüs wird die Anzahl der Einzel-Bilder (sog. "FRAMES") für den Anfang auf 100 eingestellt. Automatisch setzt der Animator das zuvor geladene Bild in alle 100 FRAMES.

Damit liegt natürlich noch keine Animation vor. Es bieten sich nun aber drei Möglichkeiten an, dem Clip dennoch Leben einzuhauchen: "OPTICS", "CELS" und "TRACE".

Unter "OPTICS" definiert der Anwender auf drei Achsen die Richtung der Animation oder greift auf vorgefertigte Bewegungsabläufe zurück (Zurückziehen, Drehungen um verschiedene Achsen, Dehnen oder Stauchen usw.).

Nachdem der Benutzer hier seine Wahl getroffen hat, schaltet er auf "Wire-Frame", um die Bewegungsabläufe in einer Preview-Funktion abchecken zu können.

Der Befehl "RENDER" dreht von alleine unseren 100teiligen Film mit den entsprechenden Vorgaben. Jetzt kann sich der Anwender zurücklehnen und je nach Umfang rund 10 Minuten warten, bis der fertige Film vorliegt. An dieser Stelle lüftet sich ein großes Geheimnis des Animators: Der Benutzer muß nur ein einziges Bild laden und einige wenige Angaben zur Bewegung machen, schon läuft das Computer-Kino.

Etwas mehr Eigeninitiative verlangt die zweite Art der Film-Produktion mit der Funktion "CEL". CEL beschreibt den Ausschnitt von Bildschirmen, die der Anwender beliebig verändern kann.

Für den dritten Weg der Film-Produktion mit der Funktion "TRACE" braucht der Benutzer nicht nur etwas mehr Zeit, sondern auch etwas zeichnerisches Talent. Ohne Hilfe von Clips oder vordefinierten

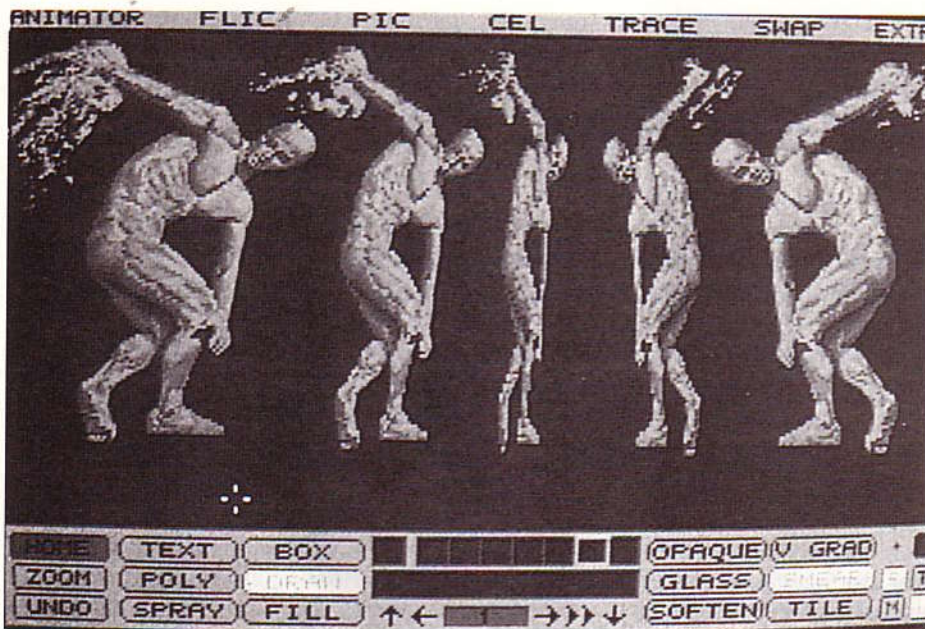


Bild 1: Optics; wie darf die Bewegung denn aussehen?

Bewegungen kann beispielsweise er die 100 Bilder einzeln zeichnen - so richtig mit Stift und Palette. Aber auch hier bietet der Animator auf Wunsch eine Hilfestellung, indem er für ein neues Bild die Silhouette des vorhergehenden einblendet, so daß Bewegungsabläufe auch von Hand ruckfrei gestaltet werden können. Hier kann die gesamte Funktionsvielfalt des eingebauten Malprogramms zum Tragen kommen: "Drizzle" (Je schneller die Maus, desto dünner der Strich), "GEL" (Weichzeichner), "PETAL" (Struktur einer Blume), "CLOSE" (füllt Pixel-Lücken) und viele weitere. Zu einem guten Film gehört auch ein Titel. Dieser kann mit der entsprechenden Textverarbeitungs-Funktion eingegeben oder von Diskette geladen werden. Wie der Titel im Film behandelt werden soll kann der Anwender in weiten Grenzen festlegen: Scrollen (auf oder ab), imaginäres Eintippen oder Stillstand.

Was kann Animator sonst noch?

Gegenüber konventionellen Animationsprogrammen, die fast immer die langwierige Eingabe von Koordinaten und die Programmierung von Bewegungsabläufen voraussetzen, muß sich der Anwender hier um diese Dinge nicht mehr explizit kümmern. Dies trifft auch zu, wenn Bewegungsabläufe manuell definiert werden sollen. Fertig abgelegte Filme können bequem durchgeblättert werden, sowie miteinander verknüpft werden. Hierbei gibt es eine Reihe von interessanten Möglichkeiten. Filme lassen sich überlagern, unterlagern sowie die Überblendung von einem Film zum anderen. Der Übergang kann hart, weich oder mit den Stilmitteln wachsende Streifen oder wachsende Ringe erfolgen. Wenn Bilder zweier Filme miteinander gemischt werden sollen, übernimmt

Animator automatisch die Farbpaletten-Anpassung, sowie die evtl. erforderliche Anpassung der Filmlängen.



Bild 2: Korrekte Farbanpassung zweier Filme für die Hintergrundkulisse

Der Menüpunkt "SPECIAL EFFECTS" bietet die doppelte Vergrößerung oder Verkleinerung eines Bildes, das Löschen mit Ausnahme eines definierten Ausschnittes, das absichtliche Verwischen einer Bewegung, eine immer größer werdende Quadratrasterung von 320 x 200 bis 1 x 1, oder die Umwandlung eines Fotos in zwei oder mehrere Halbtöne (gut für die Ausgabe auf Druckern). Bleibt noch zu erwähnen, daß der Animator die Filme natürlich auch rückwärts, in Zeitlupe oder in Zeitraffer spielt und einzelne Passagen in einer LOOP-Funktion wiederholen kann.

Mit Hilfe des mitgelieferten Format-Konvertierers lassen sich sowohl ganze Filme aus der AMIGA- und ATARI-Welt in das "FLIC"-Format des Animators umwandeln, als auch einzelne Bilder im McIntosh- oder PCX-Format verarbeiten. Über ein weiteres Konvertierungsprogramm kann der Animator auch Dokumente aus AutoCad, AutoSketch und AutoShade in Animator-Filme umwandeln.

Der Haupteinsatz des Autodesk-Animators liegt in Präsentationen auf Messen, Schulungen und in der Werbebranche. Über einen im Handbuch erwähnten Adapter (allerdings NTSC!) können die Filme auch auf Video-Band aufgezeichnet werden. Apropos Filmezeigen: Ein mit Animator erstelltes Werk kann selbstverständlich auch ohne das Animator-Paket selbst vorgeführt werden. Dazu liefert Autodesk quasi einen Filmprojektor mit, der FLI-Dateien abspielt, und seine Steuerungsanweisungen aus einer ASCII-Datei bezieht.

Unter dem Schlußstrich bleibt die Begeisterung für ein mächtiges Programm, das im Preis mächtig untertreibt. Nur etwas über DM 800 kostet das Paket. Die erste Version, in der kein einziger Fehler auftrat, läßt kaum Wünsche offen...

Mit freundlicher Genehmigung der Zeitschrift PC-Plus Magazin, Verlag Markt & Technik. Im genannten Magazin erschien im Heft 2/90, Seite 132 der Original-Bericht von H. Stefan Schneider. In der LOOP dagegen erscheint ein (modifizierter) Auszug dieses Artikels.

Volker Stahl.

CPU8088 - MS/DOS

Teil 4: CONFIG.SYS

Wenn Sie schon einmal in Ihrem DOS-Handbuch geblättert haben (ist Ihres auch in Englisch?), sind Sie sicherlich schon auf die Stelle gestoßen wo es heißt: 'How to Configure Your System', also: Wie ich mein System konfiguriere. Da stellt sich uns zunächst die Frage, was bedeutet: 'Sein System konfigurieren' ?

Um Ihnen Zeit zu sparen, habe ich selbst schon mal in meinem Fremdwörterlexikon nachgeschlagen, und erhielt die Wortdeutung 'gestalten, bilden und verformen'. Ich kann Sie beruhigen, dies hat nur sehr wenig mit der Konfigurations-Datei CONFIG.SYS zu tun. Rufen Sie doch einfach mal die CONFIG.SYS Datei auf (falls sie überhaupt auf der DOS-Diskette vorhanden ist!) - diese Datei kann man also nicht direkt ausführen, es ist also keine Programmdatei, sondern eine Textdatei. Das Kürzel '.SYS' verrät uns, daß diese Datei vom Betriebssystem benützt wird.

Nun aber zur Bedeutung; ich erkläre es am besten mit einem praktischen Beispiel:

Nehmen wir an, Ihr System bestehe aus einer CPU, RAM, 1 Laufwerk und Monitor. Dann ist Ihr System so konfiguriert (==> also voreingestellt): Prozessor 8088, 512KB RAM, 1 x 5,25" Floppy, Monochrom-Grafik. Nun wollen Sie Ihr System um ein zweites Laufwerk und um einen Drucker erweitern. Das Betriebssystem weiß aber nichts von einem zweiten Laufwerk und einem angeschlossenen Drucker. DOS muß also neu konfiguriert werden; und dies geht mit der Konfigurationsdatei CONFIG.SYS und mit (den bei DOS mitgelieferten) Standard-Treiber-Programmen.

==> mit der CONFIG.SYS Datei stellen Sie das Betriebssystem vorein (engl. setup)

Diese Konfigurationsdatei kann eine begrenzte Anzahl von speziellen Anweisungen, wie zum Beispiel BREAK oder DEVICE, verarbeiten. Am Ende dieses 4. Teiles werden Sie Ihre eigene CONFIG.SYS

Hallo NDR-PC User! Sicherlich haben Sie schon gespannt auf diese neue LOOP gewartet (nicht zuletzt wegen 'Teil 4: CONFIG.SYS, oder?').

Haben Sie auch kräftig die Kommandos aus der letzten LOOP (Teil 3: EDLIN.COM) geübt? Gut, dann können wir heute richtig 'in die Materie eindringen', und den EDLIN erstmals für etwas Sinnvolles einsetzen (Sie wissen nicht mehr was der EDLIN ist? - bitte in Teil 3 nachlesen, diesmal aber aufmerksamer!

Datei für Ihren NDR-PC erstellen können! Hier die wichtigsten Befehle (Anweisungen) in der Übersicht:

BREAK
BUFFERS
COUNTRY
DEVICE

BREAK: Damit können Sie bestimmen, ob die Abbruch-Sequenz <CTRL> C oder <CTRL> Abbr von DOS überprüft wird.
Syntax: BREAK=ON oder BREAK=OFF

BUFFERS: Damit geben Sie die Größe der Speicherbereiche an, welche DOS benützt um Daten beim Schreiben/Lesen von Disketten in diese Puffer speichert.
Syntax: BUFFERS=xxx (xxx = 2 bis 255)

COUNTRY: Damit geben Sie die landesspezifischen Angaben, wie zum Beispiel Datums- und Zeitformate oder Währungssymbole an.
Syntax: COUNTRY=xxx (xxx siehe Handbuch)

DEVICE: Damit können Sie Treiber-Programme (wie zum Beispiel Bildschirm- oder Maustreiber) installieren. Auf der DOS-Diskette sind bereits folgende Treiber vorhanden:

ANSI	.SYS	Installieren der ESC-Sequenzen
DISPLAY	.SYS	Installieren der Bildschirm-Console
DRIVER	.SYS	Installieren eines externen Laufwerkes
PRINTER	.SYS	Installieren der Drucker-Console
RAMDRIVE	.SYS	Installieren einer RAM-Disk (virtuelles Laufwerk)

Syntax: DEVICE=treiber

Interessant scheint mir der RAMDRIVE Treiber zu sein! Darum gehe ich hier näher dazu ein:

Eine RAM-Disk (==> Speicherlaufwerk) oder ein virtuelles Laufwerk existiert nur im Speicher (also nicht als Mechanik), aber nur solange der Strom angeschaltet ist! Der RAMDRIVE Treiber simuliert den Speicherbereich als zweites, oder drittes Laufwerk.

Syntax:
DEVICE=RAMDRIVE.SYS rg sg dg

rg = RAM-Größe in KB (16 bis vorhandener Speicher)
sg = Sektor-Größe (128, 512, 1024)
dg = Directory-Größe (4 bis 1024)

Teil 1: NDR PC - IBM PC

Teil 2: PC-Basiswissen

Teil 3: EDLIN.COM

Teil 4: CONFIG.SYS

Teil 5: AUTOEXEC.BAT

Daraus ergibt sich der Standardbefehl, welchen auch ich benutze:

DEVICE=RAMDRIVE.SYS 64 128 64

Standardmässig sollte Ihr NDR-PC folgend konfiguriert sein (legen Sie dazu mit EDLIN die neue Datei CONFIG.SYS an, und benützen Sie folgende Anweisungen):

BREAK=ON
BUFFERS=20
COUNTRY=049
DEVICE=ANSI.SYS
FILES=15

Und nun wünsche ich Ihnen noch viel Spaß beim Konfigurieren und 'RAM-Driven', bis zur nächsten LOOP und Teil 5: AUTOEXEC.BAT

Der Computer ist krank

Der Virus hat zugeschlagen

Stellen Sie sich vor, Sie entwickeln gerade ein wichtiges Programm, vielleicht sogar in Nacharbeit. Sämtliche Tools kommen zum Einsatz, Backups werden gemacht und und...Endlich haben Sie es geschafft: Das Programm ist fertig. Noch einmal abspeichern und dann Feierabend! Jedoch - wie dem Autor bereits passiert - es gibt nichts abzuspeichern! Statt dessen bemerkt man das vertraute Formatiergeräusch des Stepmotors des Laufwerks. Wer ähnliches erlebt hat, weiß wie einem dann zumute ist. Doch was nun ?

1.Tip: KÜHLEN KOPF BEWAHREN !

Auch wenn einem nicht der Hitchhiker-Virus in beruhigenden großen Lettern "Don't Panic!" auf den Bildschirm geschrieben hat. Gleich den Rechner ausschalten (RESET-feste Viren!). Die Festplatte vorher nicht parken. Den Rechner solange nicht einschalten, bis man sich der weiteren Vorgehensweise sicher ist.

Das klingt zwar überflüssig, wer aber einmal einen Computer-Besitzer gesehen hat, der soeben die Formatierung seiner Festplatte beobachten mußte, wird den Tip ernst nehmen. Besser eine Nacht drüber schlafen als unnötigen Schaden durch überhastete Aktionen verursachen. Freunde, Firma, Computertreff usw. informieren.

2.Tip: ALLE DISKETTEN SCHREIBSCHÜTZEN !

Dazu bei 5 1/4" Klebestreifen anbringen und bei 3 1/2" Fensterchen öffnen. Ist eine Festplatte vorhanden, dann die Schreib-Lese-Leitung auftrennen und einen Schalter einbauen, oder Festplatte ausbauen bzw. abstecken (Achtung: Festplatte ist in dem Moment meist nicht geparkt! Nicht nachträglich parken!). Sollte man sich entschließen einen Schalter zum Auftrennen der Schreib/Lese-leitung einzubauen, so darf man den geeigneten Pull UP/DOWN-Widerstand nicht vergessen, der die offene Leitung auf den logischen Pegel für 'Lesen' zieht.

3.Tip: DEN VIRUS-KATALOG (s.u.) ZUR HAND NEHMEN UND VERGLEICHEN !

Schon die Art der Schädigung kann vielleicht den Virus identifizieren. Mit Hilfe von

Der 2. Teil des Berichts über Computer-Viren beschäftigt sich mit der Situation einer Verseuchung des Rechners. Es wird versucht Hinweise zu geben, wie man als Betroffener den Schadensumfang feststellt und den Schaden eingrenzt. Außerdem werden Adressen angegeben, wo einem geholfen werden kann.

Diskettenmonitoren (z.B. Norton-Utility unter MS/DOS, DiskEdit unter JADOS) verseuchte Disketten untersuchen. Ist der Virus identifiziert, so kann man sich meist das entsprechende Anti-Virus-Programm besorgen. Ist kein Gegenmittel vorhanden oder bleibt der Virus unbekannt, dann muß man die jüngste, noch nicht verseuchte Generation von Disketten bzw. Backups finden. Dies ist umso leichter, je mehr man vorbeugende Maßnahmen getroffen hat (-> Teil 3).

4. Tip: WENN MAN SICH NICHT 100% SICHER IST; DAB ES UNVERSEUCHT IST: ALLES FORMATIEREN !

So schwer es fällt: Alles! Insbesondere die Festplatte. Denn: Ein einziger übriggebliebener Virus und der Ärger beginnt bald von vorne!

Diagnose: Virus positiv

Eine Infektion mit einem Virus läßt sich aber schon erkennen, bevor eine Schädigung auftritt, da Viren auch in der Inkubationszeit (d.h. in der Zeit von der ersten Infektion bis zum Ausbruch der "Krankheit") zum Teil deutliche Spuren hinterlassen.

Mögliche Anzeichen sind (Punkte 1 bis 7 aus [1]):

- häufige, unerklärliche Fehlermeldungen vom Betriebssystem
- längere Zugriffszeiten als üblich
- unerwartet gelöschter Bildschirm und andere, bössartige oder komische Effekte, die sich kaum durch Programmfehler erklären lassen
- unsinnige Zeichen in Files
- veränderte Filelänge von Programmen
- geändertes Erstellungsdatum von Programmen
- wiederholtes, unerklärliches Verschwinden von Files
- Rechner ist allgemein deutlich langsamer
- markante Strings, die typische Meldungen von Viren sind, lassen sich in Dateien finden

- starke Zunahme der Bad Sectors auf Disketten und/oder Festplatte

Diese Liste ist sicherlich noch nicht vollständig. Tref-fen aber ein oder mehrere

Punkte zu, so muß nicht unbedingt eine Infektion vorliegen, die Wahrscheinlichkeit ist aber sehr groß. Auch hier empfiehlt sich die Beachtung von Tip 1 - 4 .

Kann ein Viren-Detektor helfen ?

Leider muß man sagen: Begrenzt! Es wird nie einen Allround-Detektor geben, der entscheiden kann, ob ein Programm mit irgendeinem beliebigen Virus infiziert ist oder nicht. Dies läßt sich an Hand des sog. Halteproblems von Turing (-> Busy Beaver) zeigen.

Die Detektoren, die es gibt, reagieren nur auf ganz bestimmte Merkmale eines einzigen Virus (manche können auch mehrere Viren bekämpfen). Schon eine andere Generation von diesem Virus muß nicht mehr erkannt werden (z.B. gibt es 7 Viren, die alle vom ISRAELI #1 abstammen; so etwas wird 'virus strain' = Virus-Familie genannt). Eine Kontrolle auf alle bekannten Viren ist möglich, wenn auch, bei ca. 150 (Stand: Sommer '89, Frühjahr '90 vermutlich ca. 200) weltweit bekannten Viren, ein sehr komplexes Problem. Ein Lösungsversuch ist Dr. Solomons Antiviren Toolkit Version 2.4 (deutsch), das mit 3-4 Upgrades (Programm plus Handbuch) pro Jahr zur Bekämpfung auch der neuesten Viren fähig sein soll.

Dieses recht umfangreiche Programm ist zu einem Preis von 350,- DM (incl. 2 Upgrades und MWSt.) erhältlich. Interessenten wenden sich bitte an:

perComp - Verlag GmbH
Holzmühlenstr. 84
2000 Hamburg 70 .

Der Virus-Katalog

Ein anderer erfolgversprechender Ansatz ist der inzwischen auch international verwendete Viren-Katalog des Virus Test Center Hamburg unter der Führung von Prof. Brunnstein. Informationen für weit über 100 Viren (Stand: Sommer '89) sind dort incl. Bekämpfungsstrategien registriert. Der Katalog beschreibt im wesentlichen drei Aspekte[2]:

1. Bezeichnung, Art und Residenz eines Computer-Virus,
2. wesentliche Merkmale eines Computer-Virus, sowie
3. getestete Gegenmaßnahmen."

Ein Abdruck des Katalog würde leider den Rahmen des Artikels sprengen. LOOP-Leser, die sich für das Formular und die Beschreibungen der Viren interessieren, wenden sich bitte an:

Virus Test Centrum
Universität Hamburg
Fachbereich Informatik
Schlüterstr. 70
2000 Hamburg 13
(frankierter Rückumschlag !).

Das VTC hat sich bereit erklärt, im Rahmen seiner knappen Möglichkeiten betroffene PC/AT/XT-Benutzer zu helfen und zugesandte Disketten zu prüfen und auch Anti-Viren zur Verfügung zu stellen.

Den Besitzern des NDR-Klein-Computer kann dort leider nicht geholfen werden. Meines Wissens ist aber hier noch kein Virus aufgetaucht. Wer sich allerdings allgemein an Informationen über Viren auf 68000-Rechnern interessiert, schreibe an:

Prof. David Ferbrache
Heriot - Watt - University
Edinburgh
Scotland / UK .

Eine Warnung an alle, die jetzt immer noch einen Virus für den NKC in Umlauf bringen wollen: noch ist der Kreis derjenigen, die das Können hätten einen Virus zu produzieren, nicht unüberschaubar ! Es wird nochmals auf §303 b (Computersabotage) des StGB verwiesen, da das in Umlauf bringen von Viren kein Kavaliersdelikt sondern ein Fall für den Staatsanwalt ist.

Zum Schluß ein Ausblick auf Teil 3. Dort werden vorbeugende Maßnahmen erör-

tert, die ohne großen Aufwand realisierbar sind; u.a. wird auch ein sog. Signaturprogramm vorgestellt.

Literatur:

[1]Bauknecht, Kurt, Strauss, Christine
Universität Zürich "Virenprophylaxe im Hochschulbereich" Proceedings der 19. GI-Jahrestagung 1989

[2]Brunnstein, Klaus Universität Hamburg
"Zur Klassifikation von Computer-Viren: Der 'Computer Virus Katalog' " Proceedings der 19. GI-Jahrestagung 1989

Anmerkung der Redaktion:

Versehentlich wurde in der LOOP 24, im ersten Teil dieser Artikelserie: 'Freitag der 13te' der Autor nicht genannt. Natürlich stammt auch dieser Teil von Georg Neumann aus München.

Christian Czech

Teil 2

New-Technologies

Ein Einblick in die Welt der digitalen Signalprozessoren

Stand der Technik

Primär läßt sich feststellen, daß sich Signalprozessoren grundsätzlich in drei Gruppen einteilen lassen. Diese sind folgendermaßen differenziert:

Gruppe 1.

Signalprozessoren mit werkseitig integrieren Algorithmen (OnChipROM)

Gruppe 2.

Signalprozessoren mit eingeschränktem Programmierkomfort aber optimalem Konzept für spezifische signalverarbeitende Algorithmen

Gruppe 3.

Frei programmierbare Signalprozessoren mit umfangreichem Befehlssatz und weitreichenden Einsatzmöglichkeiten.

Die erste Gruppe wurde speziell für festgelegte Anwendungen konzipiert. Hierbei handelt es sich meist um programmierbare digitale Filterbausteine mit integrierten spezifischen Filteralgorithmen vom Typ

Nachdem wir im ersten Teil allgemein auf die digitale Signalverarbeitung und die Architektur von DSPs eingegangen sind, wenden wir uns heute den verschiedenen DSP-Gruppen und Typen zu.

FIR (Finite Impulse Response) oder IIR (Infinite Impulse Response). Die Filterkoeffizienten müssen natürlich von einem externen Baustein (Hostprozessor, Speicher) geladen werden, danach führt der DSP den Filtervorgang einschließlich der I/O-Vorgänge selbstständig aus. Der Vorteil gegenüber fest verdrahteten Filterbausteinen liegt in der großen Flexibilität der Filtercharakteristik, die über programmtechnische Mittel variiert werden kann. Die Funktionsweise digitaler Filter wurde an anderer Stelle in einer der letzten Loop-Ausgaben bereits ausführlich besprochen.

Ein typischer Vertreter ist der DSP 56200 von Motorola, dessen ROM bereits verschiedene Versionen des FIR-Algorithmus (einfache, duale und adaptive FIR-Version) enthält.

Die nächste Gruppe ist den oft teureren, für bestimmte Systemlösungen vorbehaltenen

Signalprozessoren zuzuordnen.

Letztere zeichnen sich durch einen kargen Befehlssatz und relativ umständlichen

Handhabung aus - würden jedoch für spezielle Signalverarbeitungen optimiert und erreichen deshalb Höchstleistungen.

In diesem Zusammenhang möchte ich neben dem VEKTOR SIGNAL PROZESSOR von Zoran v.a. die neue Digital Array Signalprozessorfamilie HDSP66 von Signal Processing Technologies erwähnen. Sie besteht aus dem eigentlichen DASP (mit Rechenwerk) und einem zusätzlichen PAC (Programmable Array Controller), der für das Systemmanagement (z.B. Adressberechnungen) verantwortlich ist.

Da der DASP bis zu acht komplexe Zahlen parallel verarbeiten kann, erreicht er eine Rechengeschwindigkeit von 500 Millionen Arithmetikoperationen in der Sekunde ! Er wurde in erster Linie für FFT-nahe Anwendungen mit extremen Geschwindigkeitsanforderungen entwickelt.

Die letzte Gruppe der frei programmierbaren Signalprozessoren bildet auch die umfangreichste. Zugehörige DSPs sind aufgrund ihres flexiblen Aufbaus und Befehlsatzes für weitreichende Anwendungen geeignet.

Neben den Kriterien der Rechengeschwindigkeit und der optimalen Eingliederungsmöglichkeit in ein System (durch integrierte Schnittstellen) spielt hier v.a. auch die Wortbreite in Verbindung mit der Darstellungsart eine wesentliche Rolle. Die Festkomma-Arithmetik ist bei 16 oder 24 Bit Wortbreite für die meisten Anwendungen ausreichend und deshalb weit verbreitet. Dagegen verfügen alle 32 Bit-DSPs über ein Gleitkomma-Rechenwerk, was den darstellbaren Wertebereich und die Rechengenauigkeit erheblich erweitert; dies ist z.B. bei Grafikanwendungen (Ray-Tracing) eine notwendige Voraussetzung.

Fast alle großen Hersteller bieten bereits vollständige frei programmierbare DSP-Familien an. Die TMS 320XX-Reihe von Texas Instruments war eine der ersten DSP-Familien auf dem Markt. Motorola zog mit ihrer DSP 5600X/9600X-Reihe erst relativ spät nach. Diese scheint dafür jedoch ausgereifter und leistungsfähiger zu sein.

Zur Unterstützung von Software- und Systementwicklern wurden von Herstellerseite aus sog. Entwicklungshilfsmittel konzipiert. Dabei handelt es sich zum einen um Softwarepakete (bestehend aus Simulator, Cross-Assembler, C-Compiler etc.), die auf bereits vorhandenen und weitverbreiteten Rechnersystemen (wie z.B. PC/AT, Mac II, VAX- oder Sun-Workstation) lauffähig sind und es ermöglichen, Applikationen bereits zu entwerfen und zu testen ohne vorher eine vollständige DSP-Karte aufbauen zu müssen.

Zum anderen sind Hardwarepakete verfügbar (bestehend aus In Circuit-Emulator, Entwicklungsboard), die es zulassen zeitkritische Soft- und Hardwareprobleme in Echtzeit zu untersuchen. Letztere sind angesichts des hohen Datendurchsatzes keine Seltenheit.

Die interne Zykluszeit der gebräuchlichsten DSPs liegt zwischen 50 und 200 ns - eine Geschwindigkeit, die relativ viele Einzyklus-Befehle (zwischen 100 und 400) bei einer Abtastperiode von 20 μ s (wir erinnern uns an das anfängliche Beispiel: Audio-Processing mit 50 kHz Sampling-Rate) zuläßt.

Tabelle 1 zeigt eine kleine Auswahl häufig eingesetzter bzw. leistungsfähiger DSPs der dritten Gruppe (frei programmierbare Signalprozessoren).

Das Leistungsspektrum reicht von Bauteilen der ersten Generation (TMS 32010) bis zu den Fließkommaprozessoren der 4. und 5. Generation (TMS 320C30, DSP 9600x).

Die auf der nächsten Seite aufgeführte Tabelle wurde der Übersichtlichkeit halber auf einige wenige Prozessorangaben reduziert. Diese sollen einen kleinen Einblick in die Quantität des DSP-Marktes gewähren, lassen jedoch keine differenzierten Leistungsvergleiche zu.

Pipelining

Der Grad des Pipelinings im Prozessor stellt normalerweise ein Qualitätsmerkmal dar, da er angibt, wieviel Teilbereiche im Prozessor maximal simultan ausgelastet werden können.

So laufen bei einer 5-stufigen Pipeline beispielsweise folgende Schritte gleichzeitig ab:

1. Instruktion holen (fetch)
2. Instruktion decodieren (decode)
3. Operanden vom Speicher holen
4. Multiplikation durchführen
5. Addition/Subtraktion durchführen, Ergebnis runden und abspeichern (3-5: execute)

Es versteht sich von selbst, daß nicht alle 5 Stufen gleichzeitig ein und denselben Befehl ausführen. Vielmehr durchlaufen aufeinanderfolgende Befehle hintereinander alle Stufen (Fließbandprinzip). So werden beispielsweise die in Stufe 3 transportierten Operanden erst im nächsten Taktzyklus multipliziert und im übernächsten addiert.

Damit wird auch eine Problematik allzuvieler Pipelinestufen deutlich. Der Datenfluß muß linear/gleichmäßig erfolgen, damit die Pipeline zu jedem Zeitpunkt vollständig ausgelastet ist!

Ist dies - wie bei den meisten signalverarbeitenden Algorithmen - der Fall, so wird in jedem Taktzyklus durchschnittlich genau eine Instruktion ausgeführt.

Würde ein derartiger Befehl allein ausgeführt, so bräuchte der Chip dazu insgesamt 5 Taktzyklen.

Datenabhängigkeit aufeinanderfolgender Befehle unterbricht ebenfalls den Datenfluß, da das Ergebnis einer Multiplikation & Addition (in obigem Beispiel) erst nach einem weiteren Taktzyklus für den nachfolgenden Befehl bereitsteht.

Arbeitet die MAC-Unit ohne Pipelining (wie bei den Motorola Chips) besteht dieses Problem nicht mehr. Das Ergebnis einer MAC-Verknüpfung kann hier mit dem

nächsten Befehl sofort weiterverarbeitet werden.

Des Weiteren entstehen durch häufige Interruptanforderungen an den Prozessor Zeitverluste - in erster Linie durch die Notwendigkeit die Pipeline nach jedem Rücksprung ins Hauptprogramm neu aufzufüllen. Dies muß in Echtzeitsystemen gebührend berücksichtigt werden.

DSP im Echtzeitbetrieb

Echtzeitsysteme - die das primäre Anwendungsfeld der Signalprozessoren darstellen - haben den entscheidenden Nachteil zeitkritisch zu sein. Dies bedeutet, daß alle partiären Rechenvorgänge innerhalb eines gegebenen, geringen Zeitintervalls (in unserem anfänglichen Beispiel etwa 20 μ s) abgeschlossen sein müssen. Um dies einhalten zu können sollte der Entwickler die Leistungsanforderungen seines zukünftigen Systems genau kennen.

Erst jetzt kann er sich aus dem großen Marktangebot einige Exemplare 'herauspicken' und diese genauer auf ihre Verwendbarkeit hin überprüfen (d.h. ihre Leistung im System abschätzen).

Nach welchen Kriterien er dabei vorgeht wirkt sich allerdings maßgeblich auf den Erfolg seines Unternehmens aus, denn Signalprozessor ist nicht gleich Signalprozessor - wie folgendes Beispiel zeigt:

Der TMS 320C25 benötigt bei einer internen Zykluszeit von 80 ns etwa 8.8 μ s für eine komplexe 1024 Punkte FFT-Berechnung (eine in der Signalverarbeitung typische Anwendung), der DSP 56001 von Motorola beendet diesen Algorithmus - bei etwa gleicher Zykluszeit (75 ns) - schon nach 2.6 μ s! (Herstellerangaben).

Ein Blick auf die Busarchitektur der beiden Prozessoren gibt Aufschluß über diese Differenz - der Motorola-DSP arbeitet intern mit 4 Bussystemen - sein Gegenspieler dagegen nur mit Zwei (einfache Harvard-Architektur).

Wie steht es nun aber mit dem leistungsstarken Hitachi-Prozessor (HD 81831, int. Zykluszeit 50 ns, 5 Bussysteme!)?

Trotz seiner geringen Zykluszeit und der höheren Integration benötigt dieser für obengenannten Algorithmus 3.8 μ s - ganze 1.2 μ s mehr als der scheinbar langsamere 56001-Chip.

Wie erklärt sich dieses 'Paradoxon'?

Ein (sehr) genauer Blick in das Datenbuch des Prozessors läßt auf folgendes schließen:

Der Hitachi-Chip entwickelt seine volle Rechengeschwindigkeit ausschließlich bei Verwendung von Daten aus den internen

Typ:	HD 81831	DSP 54000/1	DSP 96001/2	MB 8764	TMS 320C10	TMS 320C20/25	TMS 320C30	ADSP 2100A	DSP 16A	DSP 32C	LPD 77C25	LPD 77C30
Hersteller:	Hitachi	Motorola	Motorola	Fujitsu	Texas Instruments	Texas Instruments	Texas Instruments	Analog Devices	AT&T	AT&T	NEC	NEC
Min. Taktfreq./ext. Takt:	50ns/20MHz	74ns/27MHz	74ns/27MHz	100ns/10MHz	200 ns/20 MHz	200 ns/20 MHz	60 ns/33 MHz	80ns/12.5MHz	25ns/20MHz	80ns/50MHz	122 ns/8.2 MHz	150 ns/13.3 MHz
Wortbreite:	16 Bit	24 Bit	32 Bit	16 Bit	16 Bit	16 Bit	32 Bit	16 Bit	16 Bit	32 Bit	16 Bit	32 Bit
Zahlenformat:	Fest-komma	Fest-komma	Gleit-komma	Fest-komma	Fest-komma	Fest-komma	Gleit-komma	Fest-komma	Fest-komma	Gleit-komma	Fest-komma	Gleit-komma
Rechenbreite: (MAC-Unit)	16 x 16 - 32	32 x 24 - 56	24EB x 24EB - 96(*)	16 x 16 - 26	16 x 16 - 32	16 x 16 - 32	24EB x 24EB - 32EB	16 x 16 - 40	16 x 16 - 32	24EB x 24EB - 32EB	16 x 16 - 32	24EB x 24EB - 67EB
MAC-Zeit (P-Pipe-Lining):	50 ns (P)	74 ns	74 ns	100 ns (P)	400 ns (P)	200 ns (P)	80 ns (P)	80 ns (P)	25 ns (P)	80 ns (P)	122 ns (P)	150 ns (P)
Interne Busstruktur:	Harvard, stark erh.	Harvard, stark erh.	Harvard, stark erh.	Harvard, erweitert	Harvard	Harvard	Harvard, stark erh.	Harvard, modifiziert	Harvard	von Neumann	Harvard, modifiziert	Harvard
Adressbus:	4	3	3	3	2	2	5	2	2	1	2	k. A.
Datenbus:	5	4	5	3	2	2	4	2	2	1	2	k. A.
Ext. Bus-systeme:	2	1	1 / 2	2	1	1	2	2	1	1	keines	1
Schnittstellen:	parallel	parallel, SSI, SCI	parallel, SSI, SCI/DMA	keine	keine	SIO	2 x SIO, DMA	keine	P10, S10	DMA, S10	P10, S10	parallel, S10
Technologie:	1.3 um, CMOS	1.2 um, CMOS	1.2 um, CMOS	7, CMOS	NMOS, CMOS	NMOS, 1.6 um CMOS	1.0 um NMOS	1.0 um, CMOS	1.0 um, CMOS	0.75, DMOS	1.5 um, CMOS	7 um, CMOS
Bemerkung:	Lädt paral. zum internen Rechenvergang Daten aus externem Speich. nach	MAC-Unit kommt ohne Pipelining aus. Grosser Adressraum	spezielles Datenformat '96002 hochleistungsgrafikproz.	1. CMOS-DSP am Markt	Kostensteiger DSP der ersten Generation	Timer	2 int. Timer großer ext. Adressraum	Kein On-Chip Ram	Für Telekommunikation optimiert	kein Harvard, jed. Buszyklus von 20 ns!	Kein ext. Bussystem, daher nur Peripheriebaustein	Verfügt nur über Ein-Zyklus-Befehle. Bis zu 33 MFLPps!

Tabelle 1. Eine Auswahl von frei programmierbaren DSPs

schon eine Vielzahl von Anwendungsgebieten ab:

- Meßtechnik

Erst durch den Einsatz digitaler Signalprozessoren wurde es möglich, kostengünstige(re) digitale Meßgeräte zu entwickeln, um hochfrequente Analogsignale in Echtzeit zu analysieren (FFT, Autokorrelation etc.).

- Steuerungstechnik

Hier stehen Einsatzgebiete wie Robotik aber auch die Kfz-Elektronik im Vordergrund. Bauteile der 1. u. 2. Generation sind hier aufgrund der geringen Kosten besonders gefragt.

- Weltraumtechnik

Für die Kommunikation mit Satelliten werden große Rechenleistungen benötigt. Die Nutzung unserer Chip-Klasse wird sich in nächster Zeit allerdings nur auf die Bodenstationen beschränken, da das Risiko eines Ausfalls im Orbit zu hoch ist.

- Radaranlagen

Eine Verbesserung kommt neben der militärischen v.a. der zivilen Luftfahrt zu gute. Kontrastverbesserungen und Mustererkennung durch intelligente Algorithmen sind hier angestrebte Ziele.

- Digitale Audiotechnik

Der Bereich der Musikelektronik wurde in den letzten Jahren revolutioniert. Effektgeräte, digitale Mischpulte sowie Worksta-

tions basieren u.a. auf Signalprozessoren. Digitale Massenspeicher (DAT-Recorder) profitieren ebenfalls von der neuen Technik.

- Digitale Bildverarbeitung

Rechenaufwendige Bildalgorithmen (Ray-Tracing) können nun in angemessener Geschwindigkeit ausgeführt werden. Ein immenses Anwendungsfeld stellt die Einführung des digitalen Fernsehens dar.

- Telekommunikation

DSPs finden sich in schnellen Modems, als 'Number-Cruncher' oder Echounterdrücker wieder.

etc.

Zukünftig wird in obigen Bereichen die Bewältigung noch größerer Datenmengen erforderlich. Dazu sind auch leistungsfähigere Signalprozessoren unumgänglich.

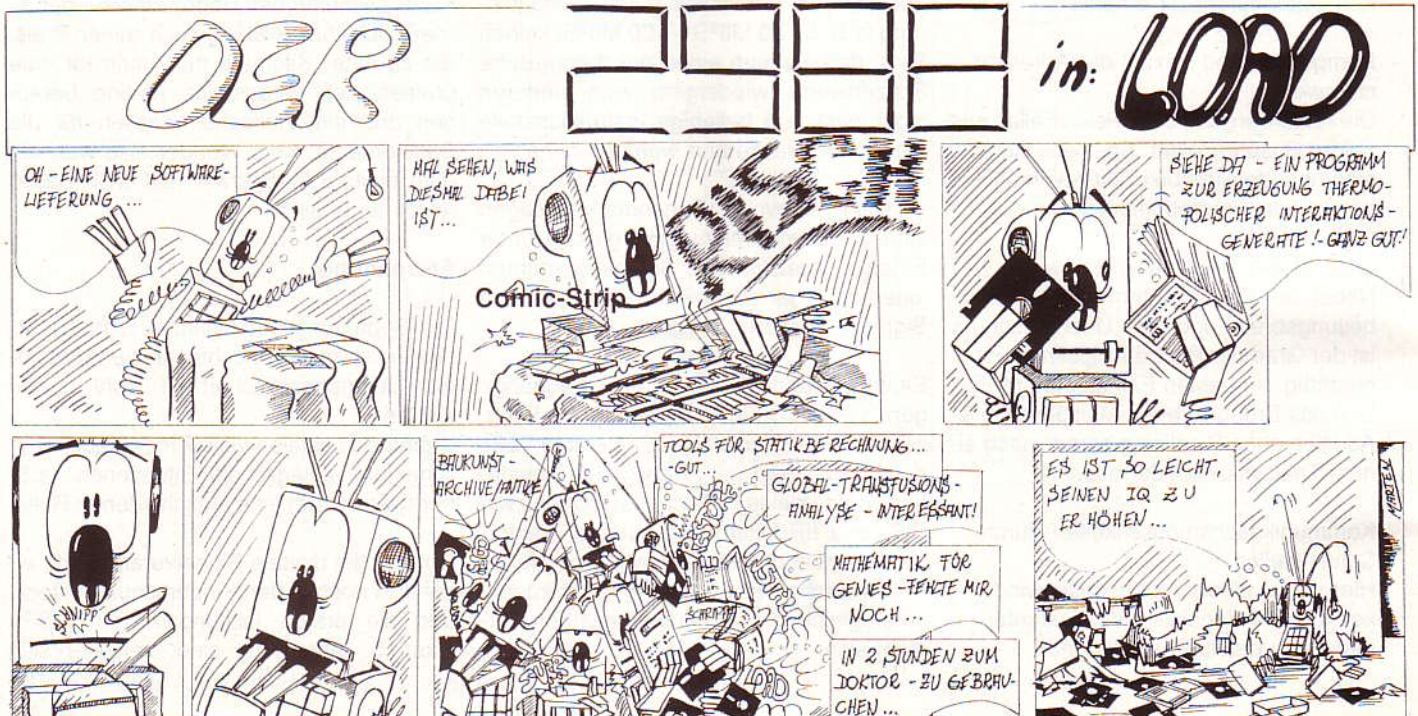
Die Integrationsgrenze von Silizium-Chips dürfte jedoch bald erreicht sein. Heutige Strukturen unterschreiten bereits die 1 µm Marke. Eine Leistungssteigerung läßt sich bei gleichem Halbleitermaterial nur noch durch globale Parallelisierung einzelner Hochleistungsprozessoren erreichen (DSP-Netzwerke ähnlich dem Tranputerprinzip) oder/und durch den Einsatz von VLIW-Strukturen in Verbindung mit speziellen vektorisierenden Compilern. VLIW-Chips verfügen über viele Rechenwerke, die parallel arbeiten und mit nur einem langen Befehlswort (Very Large Instruction Word) angesprochen werden können. In

kleinem Rahmen wird dies bereits bei den heutigen Signalprozessoren praktiziert.

Nicht weniger Interessant ist die Erforschung und Verarbeitung neuer Halbleitermaterialien, die aufgrund gewisser physikalischer Phänomene höhere Taktraten zulassen. Bestes Beispiel ist die Halbleiterverbindung Gallium-Arsenid (GaAs), auf dessen Basis bereits Chips mit Taktraten weit über 200 MHz realisiert wurden (bisher leider nur für militärische Nutzungen). Zur Zeit sind logischerweise Herstellungskosten und -aufwand von GaAs-Mikroprozessoren überdimensional hoch - die Methoden der Integration noch sehr unausgereift. Allerdings kündigten einige namhafte Halbleiterhersteller die ersten serienmäßigen GaAs RISC-Chips mit Taktfrequenzen über 100 MHz schon für Ende dieses Jahrtausends an. Man darf auf den ersten GaAs-Signalprozessor gespannt sein!

Literatur:

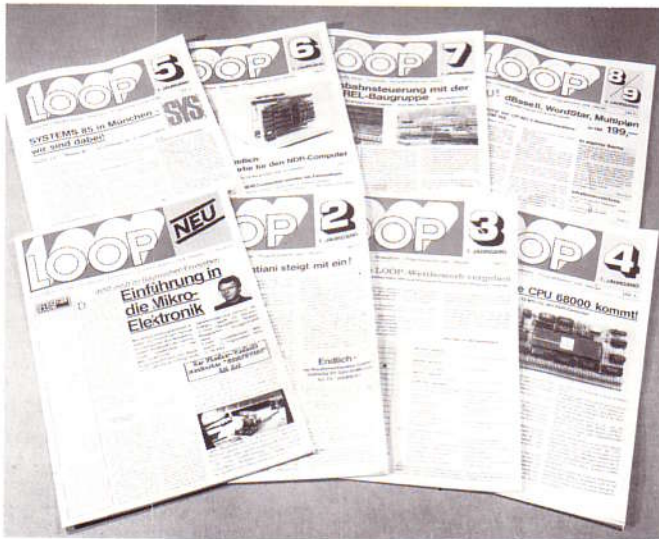
- M. Fabig: Analoge Welt A/D, c't 3/89, S. 198-210, Heise-Verlag
 - A. Bode: VLIW: Sanfter Übergang zur Parallelverarbeitung, c't 3/90, S. 232 ff., Heise-Verlag
 - D. Shear: EDN's DSP Benchmarks, Special Report in EDN Sep.29, 1988
 - Signalprozessoren 2 und 3, Oldenburg-Verlag
 - Elektronik-Sonderheft Nr. 244: Digitale Signalprozessoren, 1987
- sowie diverse Datenbücher und Artikel in Design & Elektronik, Jahrgang 1989, Markt & Technik Verlag



Auch nach dem Kauf:

Die Computerzeitschrift *LOOP* ist die Brücke zum Kunden – Programme, Infos, Tips + Tricks!

Jahres-Abo DM 25,-, Probeexemplar kostenlos!



Umfassend informiert Sie unser Katalog.

Schutzgebühr: DM 10,- incl. MWSt.



Bitte bestellen Sie mit anhängender Postkarte!

BESTELLKARTE

Ich / Wir bestelle(n) unter Anerkennung Ihrer Geschäfts- und Lieferungsbedingungen folgende Artikel:

Stück	Bestell-Nr.	Bezeichnung	Einzelpreis
	10 834	GES-Katalog	10,-
	10 061	LOOP-Abo	25,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift

Bei Minderjährigen die des gesetzl. Vertreters

BESTELLKARTE

Ich / Wir bestelle(n) unter Anerkennung Ihrer Geschäfts- und Lieferungsbedingungen folgende Artikel:

Stück	Bestell-Nr.	Bezeichnung	Einzelpreis
	10 834	GES-Katalog	10,-
	10 061	LOOP-Abo	25,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift

Bei Minderjährigen die des gesetzl. Vertreters

Neue Produkte - Neue Preise

Software			Software		
Best.-Nr.	Produkt	Preis DM	Best.-Nr.	Produkt	Preis DM
11513	RESIDEMO58 Demoverision des Regelkreissimulators RESI für MS/DOS 5 1/4"-Disk, selbstab- laufend, mit Handbuch	30.-		68020 auf 4 x 32 k-EPROMs, ohne HB	
11514	RESIDEMO38 dto. auf 3 1/2"-Disk	30.-	11221	EGRUND08QUELLE38 Grundprogramm und Quelle für 68008, Version 6.2 auf 3 1/2"-Disk, ohne HB.	89.-
11515	RESI58 RESI-Vollversion für MS/DOS 5 1/4"-Disk Grafischer Regelkreis-Editor und Simulator, analoge und digitale Regler	699.-	11263	EGRUND00QUELLE38 wie unter 11221, für 68000.	89.-
11516	RESI38 dto. auf 3 1/2"-Disk	699.-	11333	EGRUND20QUELLE38 wie unter 11221, für 68020.	89.-
11519	RESISTUDENT58 wie 11515, gegen Vorlage von Imatrikulationsbesch. oder Schülerausweis	299.-	11220	EGRUND08QUELLE58 Grundprogramm und Quelle für 68008, Version 6.2 auf 5 1/4"-Disk, ohne HB.	89.-
11520	RESISTUDENT38 dto. auf 3 1/2"-Disk	299.-	11262	EGRUND00QUELLE58 wie unter 11220, für 68000.	89.-
11218	EGRUND08KD Das <u>neue</u> Grundprogramm V 6.2 für 68008 auf 7 x 8 k-EPROMs, ohne HB	149.-	11332	EGRUND20QUELLE58 wie unter 11220, für 68020.	89.-
11219	EGRUND00KD wie unter 11218, für 68000.	149.-	11539	EGRUNDH Neues Handbuch für alle GP-Varianten der Version 6.2. Umfang: 280 Seiten!	69.-
11335	EGRUND20KD wie unter 11218, für 68020.	149.-		Hardware	
11258	EGRUND08KD256 Das <u>neue</u> Grundprogramm V 6.2 für 68008 auf 2 x 32 k-EPROMs, ohne HB	129.-	11345	MULTIIOF MULTI I/O Fertigergerät für PC- und NDR-Systeme.	398.-
11259	EGRUND00KD256 wie unter 11258, für 68000.	139.-	10365	RELF 8fach Relais-Baugruppe für NDR-Systeme. Je 2 x Um. Solange Vorrat reicht!	259.-
11334	EGRUND20KD256 Das <u>neue</u> Grundprogramm V 6.2 für	139.-	10364	RELB wie unter 10365 in Bausatzform. Solange Vorrat reicht!	179.-

Bitte
Porto
nicht
vergessen

ANTWORT

GRAF
computer

Graf Elektronik Systeme GmbH
Postfach 1610

8960 Kempten

Bitte
Porto
nicht
vergessen

ANTWORT

GRAF
computer

Graf Elektronik Systeme GmbH
Postfach 1610

8960 Kempten

Anschrift:

Lieferform: Nachnahme Vorkasse
 Bankeinzug

Bankeinzug: Hiermit ermächtige ich die Firma GES
GmbH, den Rechnungsbetrag für die auf dieser Karte
angegebenen Bestellungen von meinem Konto:

BLZ: _____ Konto-Nr. _____

Bank: _____
abzubuchen. Falls mein Konto die erforderliche
Deckung nicht aufweist, besteht seitens des konto-
führenden Kreditinstitutes keine Verpflichtung zur
Einzahlung.

Datum: _____ Unterschrift _____

Anschrift:

Lieferform: Nachnahme Vorkasse
 Bankeinzug

Bankeinzug: Hiermit ermächtige ich die Firma GES
GmbH, den Rechnungsbetrag für die auf dieser Karte
angegebenen Bestellungen von meinem Konto:

BLZ: _____ Konto-Nr. _____

Bank: _____
abzubuchen. Falls mein Konto die erforderliche
Deckung nicht aufweist, besteht seitens des konto-
führenden Kreditinstitutes keine Verpflichtung zur
Einzahlung.

Datum: _____ Unterschrift _____