

**Künstliche Intelligenz:
Lisp-Variante PC-Scheme**

Die Mikrocomputer-Zeitschrift

7 DM · 60 öS · 7 sfr. · Januar 1987

1

Leistungsexplosion:

Supermikro 80386

EPROM-Übersicht

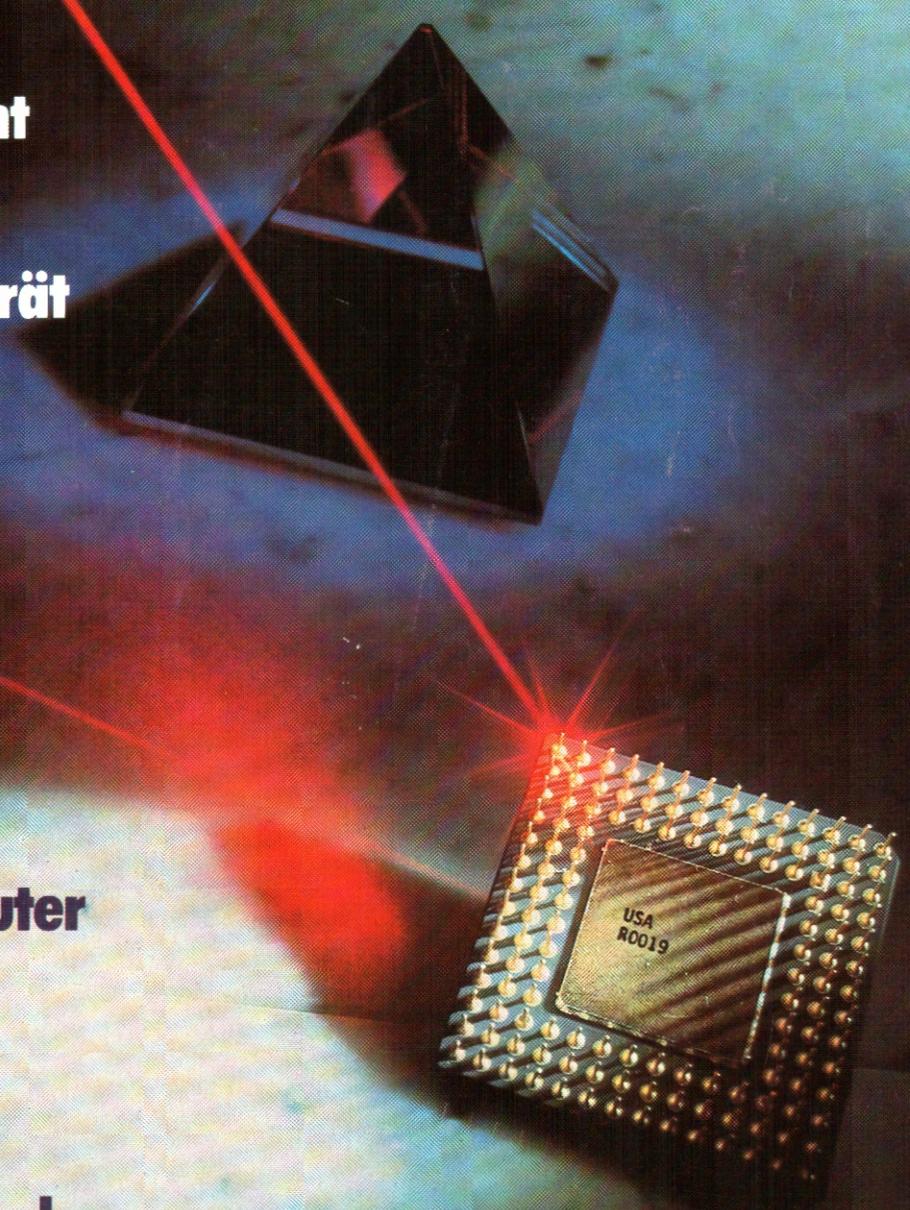
**EPROM-
Programmiergerät**

**PC-Grafik:
EGA im Detail**

**Atari ST:
Desktop-
Informationen**

**68000 im
mc-CP/M-Computer**

**Im Test:
Open Access II
Epson EX 1000
Turbo-Pascal-Tools**



mc-kolumne

Typen gibt's 3

mc-briefe 6

mc-info 14

Spruch des Monats 41

Zitat des Monats 104

mc-bücher 38

mc-grundlagen

Supermikro 80386 39

Überzeugende Architektur und gewaltige Beschleunigung bei Anwendungssoftware machen die CPU 80386 für jedermann interessant

Die mc-EPROM-Übersicht 42

Anschlußbelegung und technische Hinweise zu allen EPROMs vom 2716 bis zum 27210

Zenith mit Intel 80386 98

Eintritt in die 80386-Welt ab 14 000 DM

mc-hard

mc-EPROM-Brenner 46

Über die Centronics-Schnittstelle kann unser Selbstbaugerät die meisten EPROMs lesen und programmieren. Komplett mit Schaltplan und Software

ECB und 68000: mc-68-ECB 59

Der 68000 sorgt für frischen Wind im System und arbeitet prächtig mit dem Z80 zusammen

mc-soft

Zeichengenerator für Nadeldrucker 43

Sonderzeichen drucken mit dem Computer

Neues vom Software-Service 58

Bewährte Programme wurden verbessert, neue Programme für den Profi in den Vertrieb aufgenommen

Druckeranpassung unter MS-DOS 69

Einmal geladen – immer parat: Die universelle Routine zur Einstellung von Matrixdruckern

Austausch illegaler Z80-Befehle 74

Moderne CMOS-Z80-CPU's verweigern nichtlegale Opcodes



Intel 80386

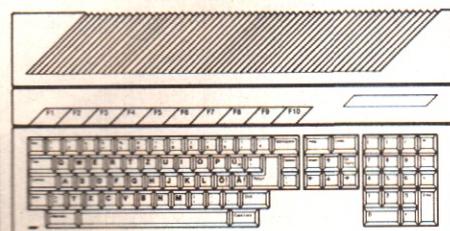
Die Supermikroprozessoren kommen. Zwar gibt es noch nicht viel Software dafür, aber die 32-Bit-Prozessoren sind überall im Gespräch. Intel hat sich mit dem 80386 hervorgetan und die ersten Personal Computer mit diesem Prozessor gibt es zu kaufen. Nicht gleich von IBM, sondern zunächst von Compaq, Rair und Zenith.

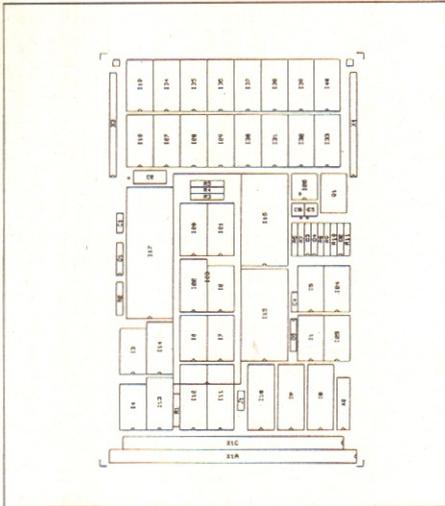
Seite 39

Tests

Data Becker ist einer der Hechte im Karpfenteich der Software-Anbieter für die modernen 16-Bit-Rechner. Sympathisch ist, daß die Produkte aus diesem Haus meistens auch für das schmale Budget tragbar sind. Textomat ST ist ein gutes Beispiel dafür. Es macht den Atari ST zu einem leistungsfähigen Schreibsystem.

Seite 99



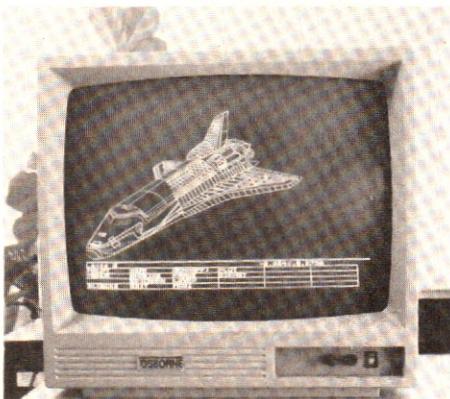


mc-68-ECB

Ausgewogenheit muß sein. Wenn den Intel-Prozessoren und den Systemen drum herum in dieser Ausgabe viel Platz eingeräumt wird, dann darf auch die Linie der 68000er nicht fehlen. mc führt vor, wie man einen 68000 in ein CP/M-ECB-System integriert. Die Platine bietet bis zu 1 MByte RAM, CP/M-68k wird laufen und – man kann alles selbst bauen. **Seite 59**

EGA

Eine neue Dimension in Sachen Grafik hat der Enhanced Grafik Adapter von IBM eröffnet. Seine Eigenschaften sind überall bekannt, aber nicht jeder Anwender kann alle Fähigkeiten nutzen. mc schildert die logische Struktur von EGA und zeigt, wie man ihn richtig programmiert. **Seite 80**



Die Desktop-Informationen des Atari ST 78
Durch Ändern eines Byte erscheint eine andere grafische Bedieneroberfläche

Grafik mit der IBM-EGA 80
Nach einem Blick auf jedes Byte des Grafik-Adapters bleibt keine Frage offen

Multitasking unter MS-DOS 86
Mit den Interrupt-Service-Routinen von MS-DOS sachverständig umgehen

Pascal-Tracer 90
Ein wertvolles Programm in Turbo-Pascal, mit dem Laufzeitoptimierungen einfach durchgeführt werden

Variablen sichtbar gemacht 94
Unter Kontrolle dieser Routine wird keine Variable mehr falsch belegt

Joyce-Disketten mit CPC-Systemen kopiert 95
Das Textverarbeitungssystem Joyce wird durch ein Basic-Programm zur universellen Kopierstation

mc-test

Preiswerte Textverarbeitung 99
Textomat für den Atari ST: Für Programmierer geeignet

Open Access II 102
Das richtige Software-Paket für alle, die ein leistungsstarkes, modular ausbaubares System benötigen

Neuer Turbo-Tutor 105
Mit diesem Lernprogramm können Profis und solche, die es werden wollen einiges dazulernen

Neue Generation 106
Die aktuelle Druckerserie EX von Epson zeichnet sich durch einfache Bedienung aus

Neue Tools für Turbo-Pascal 108
Damit nicht jeder Programmierer das Rad neu erfindet, gibt es jetzt die wichtigsten Routinen zu kaufen

PC-Scheme 110
Ein Lisp für weniger als vierhundert Mark von Texas Instrument

Netzwerke sind begehrt 112
Ein Blick auf Arcnet von Novell

mc-markt 113

mc-vorschau 142

Impressum 140

```

Program a;

(* aus Platzgründen wurde in diesem Listing der Includefile
'JRTINC.PAS'
weggelassen *)

var
  q : integer;

Procedure b;
var
  w : integer;
begin jrtracer_begin(2, 'Procedure b');
  for w:=1 to 20000 do ;
  ; jrtracer_end; End;

begin jrtracer_init; jrtracer_begin(1, 'Program a');
  b;
  for q:=1 to 10000 do ;
  ; jrtracer_end; jrtracer_output(2); End.
    
```

Bild 5. Der Tracer hat das Programm aus Bild 4 um einige Prozeduraufrufe ergänzt

Program/Procedure/Function	Aufruf	Netto	Brutto
Program a	1	4	11
Procedure b	1	7	7

Bild 6. In dieser Form erhält man die Analysedaten

den Stack und den Speicher, der vom Tracer benötigt wird.

Die Prozedur BEGIN wird zu Beginn jeder Prozedur aufgerufen. Sie speichert die Zeit des Aufrufs und die Prozedurnummer auf dem Stack. Gegebenenfalls wird auch der Name der Prozedur im Speicher vermerkt. Die Prozedur END wird am Ende jeder Prozedur aufgerufen. Sie vermerkt die verbrauchte Zeit (Brutto/Netto) im Speicher und setzt den Timer zurück. Um Kollisionen mit dem Timer zu vermeiden, wird er durch die Inline-Statements unterdrückt bzw. reaktiviert.

Die Prozedur OUTPUT wird vor dem letzten END aufgerufen und gibt die Zeiten auf dem Drucker aus. Die Konstanten MAXPROCS und STACKSIZE sind nor-

malerweise ausreichend. Bei rekursiven Programmen muß eventuell der Stack erweitert werden (STACKSIZE).

Erweiterungsmöglichkeiten

In die Prozedur BEGIN kann bei Bedarf ein Monitor eingestrickt werden, der den Pfad zur Prozedur ausgibt. Der Pfad ist auf dem Stack gespeichert. Die Prozedurnummern sind in der Stackkomponente CALLER vermerkt, sie dienen als Index auf den Array-Speicher, der die dazugehörigen Namen beinhaltet.

Ebenso ist ein prozedurweiser Einzelschrittmodus möglich. Auch kann bei rekursiven Programmen das Debugging vereinfacht werden, indem die aktuelle und die maximale Stacktiefe ausgegeben werden.

Variablen sichtbar gemacht

Bei der Programmentwicklung mit Apple-II-FP-Basic ergibt sich im Laufe der Zeit ein unüberschaubarer Wust von Variablen und Variablennamen. Tippfehler führen hier schnell zu undefinierten, „merkwürdigen“ Erscheinungen, auch mehrfaches Umbenennen einer Variablen erfordert das ständige Suchen nach veralteten Namen. Hinzu

kommt, daß der Interpreter bei undeklarierten Variablen diese ohne eine Fehlermeldung einreihet.

Das Programm VARLIST (Bild) liefert während der Laufzeit, nach Abbruch oder Ende eines Programms eine Liste der definierten Strings, Real- und Integer-Variablen und Funktionen. Es erfüllt

damit in etwa die Funktion des 6502-BRK-Befehls des Monitors und ergänzt so den Trace-Befehl des FP-Basic.

Felder werden nicht gemeldet. Numerische Variablen werden inklusive ihres Wertes ausgegeben. Der Typ String bzw. Integer wird durch \$ bzw. % gekennzeichnet, Funktionen mit &. Sind mehr als 19 Variablen definiert, so wird der Bildschirm seitenweise geblättert.

9500-	A9	27	8D	F6	03	A9	95	8D	+=0421
9508-	F7	03	A9	C6	A0	95	20	3A	+=03F8
9510-	DB	A9	00	85	50	A9	95	85	+=041C
9518-	51	20	8C	F2	AD	D2	03	29	+=039A
9520-	7F	F0	03	4C	D0	03	60	A5	+=0396
9528-	69	85	9B	A5	6A	85	9C	A9	+=0462
9530-	8C	20	5C	DB	20	58	FC	A0	+=03F7
9538-	13	84	17	A9	E6	A0	95	20	+=0392
9540-	3A	DE	A5	6C	C5	9C	90	08	+=041F
9548-	D0	13	A5	9B	C5	6B	90	0D	+=03F0
9550-	A9	DC	A0	95	20	3A	DB	20	+=040F
9558-	0C	FD	4C	BE	FD	A0	00	B1	+=0431
9560-	9B	85	06	20	5C	DB	C8	B1	+=03F6
9568-	9B	85	07	29	7F	48	F0	03	+=030A
9570-	20	5C	DB	24	06	30	01	88	+=023A
9578-	24	07	10	02	C8	C8	B9	F4	+=037A
9580-	95	20	5C	DB	A2	02	68	F0	+=03EB
9588-	01	CA	20	4A	F9	A5	07	45	+=031F
9590-	06	30	1B	A9	F8	A0	95	20	+=0347
9598-	3A	DB	20	DE	E0	20	F9	EA	+=04F6
95A0-	A5	06	10	07	A5	9D	A4	A2	+=034A
95A8-	20	F2	E2	20	2E	ED	20	8E	+=03DD
95B0-	FD	A0	07	E6	9B	D0	02	E6	+=04DD
95B8-	9C	88	D0	F7	C6	17	D0	82	+=051A
95C0-	20	50	95	4C	2F	95	8D	8D	+=032F
95C8-	D5	D3	C5	A0	A6	A0	C6	CF	+=05E8
95D0-	D2	A0	D6	C1	D2	AD	CC	C9	+=061D
95D8-	D3	D4	BD	00	8D	AD	D7	C5	+=050A
95E0-	C9	D4	C5	D2	AD	00	CE	C1	+=0570
95E8-	CD	C5	BA	A0	D7	C5	D2	D4	+=062E
95F0-	BA	8D	BD	00	A6	A4	A5	+=0463	
95F8-	BD	A0	00	+=015D					

Einen Überblick über die verwendeten Variablen gibt dieses Programm

Da alle Variablennamen intern immer nur zweistellig abgespeichert sind, kann es schwer sein, eine gesuchte Variable zu identifizieren, genauso wie der Interpreter „PA1=“ und „PA2=“ als identische Zuweisungen auf eine einzige Variable „PA“ ansieht.

Das Programm steht nach BRUN VARLIST mit „&“ zur Verfügung, es werden keine Parameter übergeben. Ohne DOS kann das Programm mit 9500G vom Monitor oder CALL 38144 von Basic aus installiert werden.

VARLIST schützt sich selbsttätig gegen Überschreiben; nach HIMEM, FP oder MAXFILES muß es erneut geladen werden. Das Programm ist für 48 KByte Speicher ausgelegt. Das Abspeichern auf Diskette geschieht mit BSAVE VARLIST,A\$9500,L\$FB.

Werner K. G. Münchheimer