

*REH*DESIGN

# CPU280

Software-Manual

## 1. Allgemeines

Die vorliegenden Versionen des Bootladers (V1.02), des Betriebssystems (V1.02snc) und des Format-Managers (V1.01) sind zwar voll funktionsfähig, aber nicht als endgültiger Stand zu betrachten. Insbesondere beim Bootlader und beim Betriebssystem sind bereits einige Details für die nächste Version vorge-merkt. Deswegen bitte ich, beim Auftreten irgendwelcher Fehler diese genau zu lokalisieren und zu beschreiben, damit sie in späteren Versionen behoben werden können. Auch konkrete Verbesserungsvorschläge und Ideen sind willkommen.

Das Betriebssystem ist modular aufgebaut, so daß Anpassungsarbeiten an die jeweilige Umgebung normalerweise nur wenige Module betreffen dürften (insbesondere Schnittstellen und Laufwerke). Mit Rücksicht auf spätere Updates meinerseits empfehle ich jedem Anwender, in den von mir gelieferten Quelltexten möglichst wenige Änderungen durchzuführen und diese genau zu protokollieren. Erweiterungen meiner Programme sollten in eigenen Modulen bzw. als Include-Dateien realisiert werden. Globale Deklarationen sind in zwei LIB-Dateien enthalten, getrennt nach Prozessor (Z280EQU.LIB) und CPU280-Karte bzw. -System (CPU280.LIB). Diese beiden Dateien sind in alle BIOS-Module eingebunden.

Die Software ist auf die Umgebung innerhalb meines Systems angepaßt, d.h. sie unterstützt außer den Funktionen der CPU-Karte auch die dort vorhandenen Schnittstellenkarten. Anhand der Treiberrouitinen kann deren Ansteuerung ersehen werden; u.U. lassen sich durch wenige Änderungen andere Karten verwenden. Die Unterstützung von optionalen Karten sollte grundsätzlich so realisiert werden, daß durch Setzen eines Flags die Option aktiviert/deaktiviert werden kann (bedingte Assemblierung). Dies ist zwar bei einigen Optionen derzeit noch nicht der Fall, wird aber ab der nächsten Systemversion so gehandhabt werden.

Bei Fragen über den aktuellen Stand irgendwelcher Features sollte man immer zunächst die entsprechenden History-Dateien betrachten. Derartige Dateien existieren für den Lader, das System-BIOS und den Format-Manager.

## 2. Bootlader

Der Bootlader besteht aus drei Modulen: Eins davon bildet die Basis für den Kaltstart (Geräteinitialisierungen, RAM-Test) und den eigentlichen Bootvorgang, die anderen enthalten die Disk-I/O-Unterprogramme und das Setup.

Nach einem Reset wird zunächst anhand der Echtzeituhr die Taktfrequenz des Prozessors ermittelt (in Vielfachen von 614 kHz). Danach wird die Konsolenschnittstelle initialisiert (entsprechend den Werten im Setup; bei Inbetriebnahme 8+n+1 Bit, 9600 Baud, ohne Handshake) und eine Startmeldung ausgegeben. Daran schließt sich der RAM-Test an, der den gesamten adressierbaren Speicher auf Beschreibbarkeit untersucht (zerstörungsfrei). Hierbei wird gleichzeitig die vorhandene Kapazität ermittelt. Verlieft der RAM-Test fehlerfrei, so wird der Bootlader ins RAM kopiert und dort gestartet. Weitere, ausführliche Hardware-Tests sind vorgesehen, aber derzeit noch nicht implementiert.

Während dieser Operationen zeigen die LED's den aktuellen Zustand an. Der Hardware-Reset schaltet alle drei LED's an. Unmittelbar nach dem Reset wird die erste LED gelöscht. Nach erfolgreicher Erkennung der Taktrate folgt die zweite LED und nach Abschluß des RAM-Tests die dritte. Bei leerer Batterie in der Echtzeituhr wird nur die zweite LED ausgeschaltet.

Im Allgemeinen (d.h. ohne Eingriff in den Bootvorgang) wird das gesamte Betriebssystem aus dem EPROM heraus gebootet, wobei die Einstellungen aus dem NVRAM verwendet werden. Hat sich die Prüfsumme des EPROM's seit dem letzten Booten geändert (System-Update), oder wurde während des RAM-Test die DEL-Taste betätigt, so wird vor dem Bootvorgang das Setup aufgerufen, in dem alle Einstellungen des NVRAM geändert werden können. Hierzu gehören z.B. die angeschlossenen Laufwerke und -typen, die Einstellungen der beiden Schnittstellen, das Bootlaufwerk und die 'Drive Search Chain'.

Betätigen von ESC während des RAM-Tests führt dazu, daß statt aus dem EPROM von Diskette gebootet wird (das Boot-

laufwerk wurde im Setup angegeben).

Im auf der CPU280 reichlich vorhandenen Speicher, der nicht für das System selbst genutzt wird, legt der Bootlader eine RAM-Disk an. Diese RAM-Disk wird dabei mit einem Directory Label versehen. Ist das Label nach einem Reset noch unbeschädigt, wird auch die RAM-Disk nicht gelöscht. Bei kleinen Beschädigungen wird der Benutzer gefragt, ob sie gelöscht werden soll. Nur bei vielen Fehlern wird die RAM-Disk ohne Frage gelöscht.

Darüberhinaus erkennt der Bootlader noch einige externe Pseudo-Drives, d.h. RAM- oder EPROM-Karten, die von CP/M wie Laufwerke angesprochen werden. Dies sind die c't 1-MB-RAM-Disk und die c't 256-KB-Solid-State-Floppy. Deren Kapazitäten werden anhand von Jumperkombinationen ermittelt. Details können im Quelltext nachgesehen oder beim Autor erfragt werden.

Da verschiedene Konsolenschnittstellen zur Verwendung kommen können und während der ersten Meldungen des Bootladers (und der Initialisierung) noch kein RAM verwendet werden darf (es ist zu diesem Zeitpunkt noch nicht getestet!), sind die benötigten Funktionen Initialisieren, Zeichenein- und -ausgabe als Macros in einer eigenen Include-Datei (LDRIO.LIB) abgelegt. Die von mir gelieferten Macros benutzen die MPU-interne serielle Schnittstelle. Zum Ansteuern anderer Schnittstellen sind entsprechende Macros selbst zu erstellen, wobei nur die im Quelltext ausdrücklich freigegebenen Register verwendet werden dürfen.

## 3. Bedienung des Setup-Programms

Ein Modul des Bootladers ist das Setup-Programm. Damit können verschiedene Einstellungen der CPU280 festgelegt werden. Das Setup-Programm ist hierarchisch aufgebaut und bietet derzeit drei Einstellungsbereiche an:

- |                   |  |
|-------------------|--|
| 1. Laufwerke      | Hier werden die angeschlossenen Diskettenlaufwerke deklariert.   |
| 2. Schnittstellen | Hier erfolgen alle Einstellungen für die beiden seriellen Schnittstellen der CPU280 sowie die Device-Zuordnungen nach Reset. |
| 3. Sonstiges      | Alles, was nicht unter 1. und 2. paßt.   |

Grundsätzlich wird das Setup-Programm über einzelne Tastenbetätigungen bedient (Single-Button). In der Regel werden nur numerische Eingaben (jeweils eine Ziffer als Auswahl aus den angebotenen Möglichkeiten) benötigt. Die meisten Menüpunkte erklären sich von selbst, so daß im folgenden nur die vielleicht unklaren Punkte beschrieben werden.

Im Menü 'Schnittstellen' kann man außer den Einstellungen der seriellen Schnittstellen auch die Zuordnungen nach dem Booten festlegen. Darunter sind die Zuweisungen zu verstehen, die man auch mit dem CP/M-Plus-Kommando DEVICE einsehen bzw. ändern kann. Im Setup-Programm werden dazu Code-Nummern verwendet, die gleich der Nummer der jeweiligen Schnittstelle in der Device-Tabelle sind (beginnend bei 0). Ein Wert von 0 repräsentiert also die Schnittstelle CRT1, der Wert 1 die CRT2, und der Wert 5 die Parallelschnittstelle SPOOL (bei meiner System-Konfiguration). Werden bei diesem Menüpunkt falsche Werte eingetragen, so kann z.B. nach der Meldung '62k TPA' nichts mehr ausgegeben werden!

Der Menüpunkt 'Speichergröße' in 'Sonstiges' dient nur zu Testzwecken. Die Speichergröße wird bei jedem Reset neu auf den tatsächlich ermittelten Wert gesetzt.

Die Umschaltung zwischen 'Normalzeit' und 'Sommerzeit' hat keinen direkten Einfluß auf die aktuelle Zeit. Man setzt damit vielmehr ein Flag in der Echtzeituhr, so daß diese (bei Einstellung Sommerzeit) am ersten Aprilsonntag automatisch von 1:59 Uhr auf 3:00 Uhr weiterschaltet und äquivalent am letzten Sonntag im Oktober wieder zurück. Da das Wochentags-Register durch die aktuelle Software jedoch nicht gesetzt wird, kann die Umschaltung durchaus an einem anderen Tag erfolgen. Da auch die Umschaltzeitpunkte nicht mit denen zwischen MEZ und MESZ übereinstimmen, sollte man zur Sommerzeiteinstellung besser die Uhr einfach eine Stunde vorstellen (wie jede andere Uhr auch).

Zur Eingabe der 'Drive Search Chain' gibt man einfach die

vier Kennbuchstaben hintereinander ein (ohne Doppelpunkte). Will man weniger als vier Laufwerke angeben, kann man durch die Eingabe von 'x' vorzeitig beenden.

Die Drive Search Chain wird (ebenso wie das Temporary Drive und die Search Order) automatisch bei jedem Systemstart gestellt. Dadurch wird der gesonderte Aufruf von SETDEF gespart.

Das 'Bootlaufwerk', welches im Menü 'Sonstiges' angegeben werden kann, ist nicht mit dem Default-Laufwerk von CP/M zu verwechseln, das bei GENCPM festgelegt werden muß (das Laufwerk, auf dem sich der Anwender nach dem Systemstart vorfindet). Hier legt man einfach nur fest, von welchem Laufwerk beim Booten von Diskette (z.B. für Tests neuer Systemversionen) die Systemdatei CPM3.SYS geladen werden soll.

#### 4. AutoFormat-System

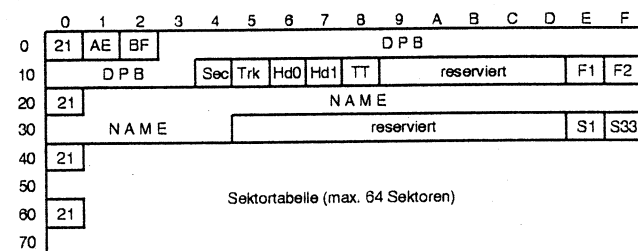
Sowohl Bootlader als auch System-BIOS verfügen über eine AutoFormat-Funktion, d.h. sie passen sich an das aktuelle Diskettenformat an. Hierzu befindet sich auf den Disketten in Spur 0, Sektor 1 ein 128 Byte großer Parameterblock (Format Parameter Block, FPB), der das physikalische und logische Diskettenformat vollständig beschreibt.

Wird ein Diskettenwechsel festgestellt (ein weiteres Feature des System-BIOS), so wird beim nächsten Zugriff zunächst versucht, diesen Parameterblock zu lesen. Treten dabei Fehler auf, oder enthält die Diskette keinen Parameterblock, so werden die Werte vom zuletzt bearbeiteten Format weiterverwendet. Durch den Format-Manager ist es natürlich auch möglich, das System auf Fremdformate ohne Parameterblock einzustellen.

Das Default-Format wird im Disk-Modul des Lader-BIOS bzw. im Tabellen-Modul des System-BIOS festgelegt und muß das größtmögliche zu bearbeitende Format beschreiben, da die Größe der Directory-Puffer und der Allocation-Vektoren beim Generieren eines Systems durch GENCPM automatisch anhand der Diskettenparameter eingestellt werden. In der von mir ausgelieferten Version ist hier ein Phantasie-Format mit 880 Blocks, 512 Directory-Einträgen und 1024 Byte großen Sektoren eingetragen. (D.h. man kann nur von parametrisierten Disketten booten!)

Derzeit wird während der AutoFormat-Bearbeitung nicht überprüft, ob die angelegten Puffer groß genug für das gewählte Format sind; deswegen werden Fehler hier unweigerlich zu Abstürzen führen.

Der Parameterblock (FPB) auf Diskette hat den folgenden Aufbau:



Die verwendeten Kürzel haben folgende Bedeutungen:

- 21,AE,BF CP/M-verträgliche FPB-Kennung und ID
- DPB Standard-CP/M-Plus Disk Parameter Block
- Sec,Trk Physikalische Sektor- und Spurzahl
- Hd0,Hd1 Logische Kopfnummern Vorder-/Rückseite
- TT Spurübersetzungs-Kennzahl (s.u.)
- F1,F2 Bitflags
- NAME Name des Formates, 20 Zeichen ASCII
- S1,S33 Sektoren Nr.1 und Nr.33 (fehlen in Tabelle)
- Sektortabelle Tabelle mit logischer Sektorfolge (Skewtable)

Die Spurübersetzungsverfahren sind dabei wie folgt definiert (v=Vorderseite, r=Rückseite):

- 0 = keine Spurübersetzung (Modus im Bit-Flag gültig)
- 1 = v0..vmax, rmax..r0
- 2 = vmax/2..vmax, vmax/2-1..v0, rmax/2..rmax, rmax/2-1..r0
- 3 = v0, v1, r0, r1, v2, r2, v3, r3, ..., vmax, rmax
- 4 = v0, r0, v1, r1, ..., vmax, rmax (Halbspur mit fortlaufenden Sektornummern, z.B. bei Kaypro)

Aufbau des Bit-Flags F1:

- Bit 7,6 : Density (00 = FM, 01 = MFM, 11 = HD)
- Bit 5,4 : reserviert
- Bit 3 : Daten invertiert
- Bit 2 : Multi-Sector-I/O möglich
- Bit 1,0 : Überlauf (00 = keiner (einseitig), 01 = Sektor, 10 = Spur, 11 = Halbspur)

Das zweite Bit-Flag (F2) ist bislang reserviert.

#### 5. System-Generierung

Sowohl für den Bootlader als auch für das System-BIOS empfiehlt sich die Verwendung von Make-Dateien, damit man die teilweise umständlichen Programmaufrufe bei der Systemprogrammierung automatisieren kann.

Wir verwenden dazu das Make-Utility von Wolfgang Mues, welches unter CP/M-Plus läuft, die normalen Zeiteinträge der Dateien verwendet und Freeware ist. Das Utility und entsprechende Make-Files liegen der Systemsoftware bei. Dadurch braucht nach beliebigen Änderungen nur Make zu starten.

Da kein echter Z280-Assembler frei verfügbar ist, bedienen wir uns hier eines weiteren Freeware-Utilities. Axel 'Fifi' Zinser hat den PRE280 geschrieben, der die Quelltexte nach Z280-Anweisungen durchsucht und diese durch DB/DW-Pseudobefehle ersetzt. Dadurch kann trotz Verwendung echter Z280-Mnemonics (keine Macro-Aufrufe!) ein normaler Z80-Assembler verwendet werden (Empfehlung: SLR180).

Zum Linken des System-BIOS kann nur der DRI-Linker LINK verwendet werden. Es ist der einzige Linker, der fehlerfreien SPR-Code (System Page Relocatable) erzeugt. In genau diesem muß das System-BIOS vorliegen, um von GENCPM mit dem BDOS zusammengefügt zu werden. LINK ist übrigens Bestandteil von CP/M-Plus, läuft aber auch unter CP/M-2.2.

GENCPM richtet sich bei der Konfiguration des Systems nach den Werten in der Steuerdatei GENCPM.DAT. Diese Datei enthält Angaben über verfügbare Speicherbereiche, Hash- und Cache-Einstellungen und dergleichen. Bei größeren Änderungen am Betriebssystem (z.B. Einfügen einer Festplatte) sollte man grundsätzlich GENCPM noch einmal 'manuell' durchgehen und die Einstellungen prüfen bzw. ändern. Dabei kann folgende allgemeine Empfehlung gegeben werden: Für alle vorhandenen physikalischen Diskettenlaufwerke sollte man genug Directory-Buffer vorsehen, daß die gesamte Directory des üblicherweise verwendeten Formates in die Puffer paßt. Nicht angeschlossene Laufwerke bekommen keinen Puffer (man gibt dann die gemeinsame Verwendung mit einem der vorhandenen Drives an). Pseudo-Laufwerke (RAM- oder EPROM-Disks) benötigen i.A. nur einen Directory-Buffer. Bei den Datenpuffern verhält sich die Sache grundsätzlich anders. Um einen gemeinsam genutzten Disk-Cache zu erhalten, empfiehlt sich die Deklaration eines größtmöglichen Puffers für Laufwerk A: und gemeinsame Nutzung durch alle anderen Laufwerke.

Um bootfähige EPROMs zu erstellen, muß man die von GENCPM erzeugte Systemdatei CPM3.SYS mit dem Bootlader (LOADER.CIM) und dem Kommandoprocessor (CCP.COM) zusammenfügen. Diese Funktion erfüllt das Programm GEN-EPR, welches die Datei SYSTEM.EPR anlegt. Diese Datei muß nur noch mit SPLIT16 in zwei Hälften (EVN gerade, ODD ungerade) aufgeteilt werden, bevor man diese in EPROMs programmieren kann.

#### 6. Fehlerbehandlung

Treten bei Diskettenoperationen Fehler auf (Time-Out, Schreib-/Lesefehler oder CRC-Fehler), so wird ähnlich wie bei MS-DOS eine Fehlermeldung ausgegeben und eine Eingabe vom Benutzer abgewartet. Dieser kann zwischen Abbruch, Ignorieren und Wiederholen wählen (durch Eingabe von A, I oder R). Eine Eingabe von ^C an dieser Stelle wirkt wie 'Abbruch', übergibt jedoch dieses Zeichen auch an das laufende Programm.

Bei der aktuellen Systemversion (V1.02snc) besteht noch das Problem, daß nach einem Time-Out beim Zugriff auf ein Laufwerk spätere Zugriffe auf andere Laufwerke nicht mehr möglich sind (d.h. Fehlermeldungen produzieren). Abhilfe kann

die Eingabe von ^C auf der CP/M-Plus-Kommandozeile schaffen (Ausloggen aller Laufwerke). Die Fehlerursache ist bekannt, mit der nächsten Systemversion wird dieser Fehler behoben.

Treten 'Fehler' der CPU auf (z.B. Traps bei versuchter Division durch Null oder wegen eines privilegierten Befehls), wird eine entsprechende Meldung ausgegeben und der Kommandoprozessor gestartet. Ebenso löst ein NMI (Non-Maskable Interrupt) einen Neustart des CCP aus, allerdings ohne Ausgabe einer Meldung.

## **7. Format-Manager**

Der Format-Manager dient zur Verwaltung der verschiedenen Formate im Zusammenhang mit dem AutoFormat-System. Er ist ebenfalls modular aufgebaut, aber in Turbo-Pascal programmiert. Die Bedienung erfolgt menügesteuert, so daß keine ausführliche Anleitung erforderlich ist.

Sämtliche Informationen über die Formate werden in der Datei FORMAT.DAT abgelegt. Diese wird beim Programmstart zunächst auf dem aktuellen Laufwerk, bei Mißerfolg dann auf Laufwerk A: gesucht.

Die Verwaltung der Formatdefinitionen beschränkt sich auf ein sinnvolles Minimum. Formate können nicht beliebig umsortiert oder gelöscht werden, sondern man kann nur neue Definitionen hinzufügen oder bestehende ändern. Dies ist insofern auch sinnvoll, da sich die Nummern oft benutzter Formate einprägen und daher nicht geändert werden sollten. Zu Tests bei der Untersuchung fremder Disketten sollte man immer dieselbe Formatnummer verwenden, um unnötigen Daten-Ballast zu vermeiden.

Vom Format-Manager kann eine formatierte Liste aller definierten Formate ausgedruckt werden. Es werden etwa 120 Zeichen pro Zeile gedruckt, weswegen der Drucker vorher auf Schmalschrift eingestellt werden muß. Da die Ansteuerung von Druckern nicht einheitlich ist, kann diese Umschaltung nicht vom Format-Manager vorgenommen werden.

Das Einstellen eines Formates auf einem Diskettenlaufwerk, in dem sich eine Diskette ohne gültigen FPB befindet, ist sowohl durch den Menüpunkt 'Fremdformat bearbeiten', als auch durch Kommandozeilenparameter beim Programmaufruf möglich. Für die letztere Methode gibt man als Parameter die Nummer des gewählten Formates und den Kennbuchstaben des Laufwerkes an, z.B. 'FORMAT 20 C'. In diesem Fall wird der Format-Manager sofort nach Ausführung der Funktion wieder verlassen.

## **8. Standard-Formate**

Die zu empfehlenden Standardformate (Reh CP/M-3 V2.1, Reh 5.25" HD und Reh 3.5" HD) zeichnen sich durch Ausnutzen der maximalen Kapazität des jeweiligen Datenträgers sowie durch extreme Geradlinigkeit aus. Grundsätzlich werden Sektoren von 1 KB Größe verwendet, die zweite Seite wird per Sektorüberlauf angesteuert und der Skew beträgt 1 (physikalisch) und 2 (logisch). Die Formate haben keine Offsetsuren.

Diese Grundsätze sollten auch bei allen zukünftigen Formatdefinitionen angewandt werden. Diese Formate zeichnen sich durch hohe Leistungsfähigkeit und gleichzeitig gute Portierbarkeit aus.

Die logischen Parameter (Blockgröße, Directory-Größe) sind weitgehend dem persönlichen Geschmack überlassen. Trotzdem empfiehlt sich auch hier die Verwendung einheitlicher Werte. Die Blockgröße sollte 2 KB betragen, um unnötige Blocking-Verluste zu vermeiden. Bei großen Kapazitäten (z.B. 3.5" HD) sind auch 4-KB-Blocks zu vertreten.

## **9. Vorgesehene Verbesserungen**

Bei der Arbeit mit den beschriebenen Software-Versionen sind bereits einige Verbesserungsmöglichkeiten deutlich geworden. Die im folgenden aufgeführten Punkte sind für die Realisierung in einer der nächsten Systemversionen vorgesehen:

- a) Global:
  - Erweiterungen grundsätzlich durch bedingte Assemblierung abschaltbar. Dazu neue globale Steuerdatei OPTIONS.LIB.
  - Alle Texte (Setup, Format-Manager, Fehlermeldungen) werden durch bedingte Assemblierung aus mehreren Sprachen ausgewählt.
- b) Bootlader:
  - Möglichst vollständiger Hardwaretest mit klaren Fehlermeldungen über LED's und/oder Klartextausgabe.
  - Evtl. Disk-Lademöglichkeit auch für CCP. Vorschläge für eine mögliche Realisierung sind willkommen.
  - Erweiterung des Setup um Einstellungen für Laufwerks-Anlauf- und Nachlaufzeiten sowie Anzahl der Wiederholungen bei Read, Write und Verify.
  - Erweiterung des Setup um Festplattenparameter.
- c) System-BIOS:
  - Ein neues residentes BDOS ersetzt das bisherige RESBDOS mit deutlichen Vorteilen. Dadurch Aufruf des gebankten Systems durch System Calls (SC) mit höherer Sicherheit und Geschwindigkeit.
  - Neue User-Funktionen 'Connect To Interrupt' und 'Disconnect Interrupt' machen externe Interrupts vom ECB-Bus zugänglich fuer Anwenderprogramme.
  - Nach DI-Instruktionen läuft ein Time-Out. Wenn Programme die Interrupts unzulässig lange sperren (gedacht ist an 100ms), werden diese einfach wieder freigegeben.
  - Evtl. Änderung des BIOS-eigenen XON/XOFF-Protokolls, welches derzeit einige Schwächen aufweist. Evtl. auch völlige Entfernung aus dem BIOS, da Hardware-Handshake effektiver ist und für die Konsole das BDOS ein leistungsfähiges Softwarehandshake durchführt.
  - Bei 40-spurigen Diskettenlaufwerken wird die Steprate gegenüber 80-spurigen Laufwerken halbiert (entsprechend den Spezifikationen).
  - Nach einem FDC-Reset (bei Time-Outs nach versuchten Zugriffen) werden alle eingeloggtten Laufwerke auf Spur 0 gefahren, dadurch nach Time-Out eines Laufwerkes keine Fehler mehr auf allen anderen Laufwerken.
  - Evtl. Abwicklung der Character-I/O-Funktionen über Interrupts (aber immer noch tabellengesteuert).
- d) Format-Manager:
  - Die Datei FORMAT.DAT wird entlang der 'Drive Search Chain' gesucht, nicht stur auf dem Default-Drive und auf A:.

## **10. Prozessor-Bugs**

Leider ist die Z280-MPU alles andere als fehlerfrei. Wir haben (zusätzlich zu den bereits von Zilog bestätigten) einige zum Teil sehr schwerwiegende Bugs gefunden. Diese wirken sich in der Regel nicht auf Anwenderprogramme aus; Programme wie WordStar, Turbo-Pascal, Macro-80, SLR-Tools, ARC und dergleichen laufen völlig fehlerfrei. Schwierigkeiten kann es dagegen bei der Systemsoftware geben, z.B. derart, daß ein einwandfrei programmiertes und bisher perfekt funktionierendes Unterprogramm nach Verschieben um ein paar Byte plötzlich nicht mehr korrekt arbeitet. Aus diesem Grunde ist auch seit einiger Zeit die Weiterentwicklung der Systemsoftware etwas ins Stocken geraten. Inzwischen scheint klar, daß von Zilog keine Unterstützung bezüglich Bugs gegeben wird, so daß stabil laufende Softwareversionen wohl nur durch Ausprobieren ermittelt werden können.

Die beschriebene Software gehört übrigens zu dieser Kategorie der stabil laufenden Systemsoftware.

Es folgt noch eine kurze Liste mit den wesentlichen Bugs:

- Auf einen I/O-Write mit Wait-States darf nicht unmittelbar ein Sprungbefehl (oder auch Call/Restart) folgen. Dies betrifft sowohl die CPU (per OUT-Befehle) als auch die DMA (im Flowthrough-Mode, Single-Byte oder Burst). Angeblich soll der Fehler nur bei wirklich vielen Wait-States auftreten, nicht bei den in der CPU280 grundsätzlich aktivierten vier Wait-States. Das Problem läßt sich dadurch lösen, daß zwischen die beiden kritischen Befehle ein IN-Befehl oder vier NOP-Befehle eingefügt werden. Um Anwenderprogramme auf kritische Sequenzen zu untersuchen, steht das Programm OUTJMP zur Verfügung. Beim DMA-Betrieb ist prinzipbedingt kein Work-Around möglich (!).
- Der DIVUW Befehl liefert unter manchen Umständen ein fehlerhaftes Ergebnis. Als Umstand wird ein gesetztes MSB (Bit 15) des Divisors angegeben. Von anderer Seite verlautet allerdings, es gebe noch weitere Fälle, in denen das Ergebnis falsch ist. Die vorzeichenbehaftete Division mit DIVW ist angeblich fehlerfrei.
- Die DMA gibt manchmal nach einem I/O-Write den Bus nicht wieder frei, sondern verharrt in einem Wartezustand bis zu 20µs lang. Beim Beschreiben von HD-Disketten führt dies zu Fehlern, da der FDC nicht bedient wird. Für diesen Fehler gibt es einen Work-Around, den Stefan Nitschke durch Zufall gefunden hat und von dem nicht einmal Zilog weiß, warum er funktioniert. Er funktioniert aber, und zwar zuverlässig. Die Systemversion 1.02snc unterscheidet sich von der 1.02 nur durch den Einbau dieses Work-Around ('Stefan Nitschke Chaos').
- Die CPU liefert nach bestimmten Operationen ein fehlerhaftes Carry-Flag. Ein durch zwei Befehle EX AF,AF' eingerahmter arithmetischer Befehl ist eine solche Befehlsfolge, nach der das Carry-Flag den falschen Wert hat. Weitere Sequenzen sind derzeit nicht bekannt. Der Fehler tritt adresslagenabhängig auf, wobei diese Abhängigkeit wiederum durch Ein- bzw. Abschalten des Cache beeinflusst werden kann. Es könnte ein Zusammenhang zu einer anderen Adresslagenabhängigkeit im System-BIOS bestehen, bei der ebenfalls das Carry-Flag eine Rolle spielt. Dieser Fehler ist der schwerwiegendste, da er (bisläng zumindest) nicht greifbar oder eingrenzbar ist.