

C P U 2 8 0

HARDWARE-HANDBUCH

Stand 210690 TR

Inhaltsverzeichnis

1. Überblick	3
2. Schaltungsbeschreibung	3
2.1 Reset, CPU und Adreßräume	3
2.2 EPROM	4
2.3 DRAM, Burst-Mode	4
2.4 ECB-Interface und I/O-Timing	5
2.5 Echtzeituhr und NV-RAM	6
2.6 Floppycontroller	6
2.7 Serielle Schnittstellen	7
2.8 General-Purpose-I/O (GPIO)	7
3. Bestückung, Inbetriebnahme	7
4. Verwendung der Steckbrücken (Jumper) und Lötbrücken	8
A1. Stückliste	9
A2. Steckerbelegungen	10
A3. GAL-Programmierung	12
A4. Organisation Adreßräume Speicher/IO	14
A5. Schaltbild	15
A6. Bestückungsplan	16
A7. Timing-Diagramme Speicherzugriffe, ECB-Bus	17

1. Überblick

Die CPU280 stellt einen vollständigen Computer im Einfach-Europaformat (100x 160 mm) dar. Die CPU280 ist dafür konzipiert, alle Leistungen des Prozessors Z280 bei minimalem technischem Aufwand möglichst optimal auszunutzen. Funktionen, die zwar möglich wären, aber in der Praxis kaum genutzt werden, wurden daher zugunsten einer einfachen und klaren Struktur bewußt nicht realisiert. Aufgrund der vollständigen Kompatibilität des Z280 zum Z80 kann bestehende Software (z.B. unter CP/M) weiterverwendet werden. Zukünftige Software kann von der größeren Leistung und dem umfangreicheren Befehlssatz des Z280 profitieren.

Auf der Karte befinden sich zwei EPROM's mit insgesamt 64 oder 128 KB Kapazität, 512 KB bis 4 MB dynamisches RAM, eine Echtzeituhr mit integriertem batteriegepuffertem RAM, ein Floppycontroller für bis zu vier Laufwerke, sowie zwei serielle Schnittstellen nach V.24. Die Verbindung nach außen erfolgt über ein ECB-Bus-Interface, welches Zugriffe auf externe I/O-Baugruppen ermöglicht. Darüberhinaus stehen noch drei durch Software einlesbare Steckbrücken und drei LED's zur Verfügung (z.B. zur Systemkonfiguration und für 'harte' Fehlermeldungen).

In der CPU280 wird der Z280 bei größtmöglichem Durchsatz betrieben, d.h. mit 16-Bit-Bus (Z-Bus), Taktverhältnis 1:1 (externer Takt gleich interner Takt), keine Wait-States, Taktfrequenz bis zu 12,5 MHz. Da der ECB-Bus bei dieser hohen Frequenz nicht mehr arbeitet und auch keine Peripherie-IC's für 12 MHz erhältlich sind, werden I/O-Zugriffe mit 4 Wait-States gebremst und das Bus-Timing entsprechend gestreckt. Der Bustakt beträgt dabei nur noch die Hälfte des CPU-Taktes (6 MHz).

Das Schaltungskonzept der CPU280 ist extrem geradlinig und die Timing-Steuerung arbeitet vollständig synchron. Dadurch wird eine hohe Zuverlässigkeit, Unabhängigkeit gegenüber Chargenstreuungen und Temperatureinflüssen, aber auch eine einfache Fehlersuche garantiert. Die Leiterplatte ist in einfacher Zwei-Lagen-Technik realisiert, wodurch unnötige Kosten vermieden werden.

Die CPU280 wird mit einer Spannung von 5 V betrieben, weitere Spannungen sind nicht erforderlich. Der Stromverbrauch der Karte liegt bei etwa 350 mA.

2. Schaltungsbeschreibung

2.1 Reset, CPU und Adreßräume

Auf der CPU280 sorgt ein Spannungswächter (IC5) für ein sicheres Reset-Signal, solange (oder sobald) die Betriebsspannung unter 4,6 V absinkt. Über den ECB-Bus kann die Karte durch Aktivieren des Reset-Einganges ebenfalls zurückgesetzt werden. Das Reset-Signal steht in gepufferter Form am ECB-Pin 'Reset-Out' für weitere Karten zur Verfügung.

Die Konfiguration der kritischen CPU-Timingparameter kann beim Z280 nur während des Reset erfolgen. Dazu muß bei der steigenden Flanke von /RESET das /WAIT-Signal anstehen, gleichzeitig die gewünschte Konfiguration (für das Bus Init Register) am Datenbus D0..D7. Bei der CPU280 wird dies durch die Schaltung um IC32A und T1 erreicht, wobei die Steckbrücken J1..J4 und die Widerstandsnetzwerke RN2, RN6 und R4..R7 die entsprechenden Daten bereitstellen. Mit Hilfe der Brücken J1..J4 kann die Anzahl der Wait-States für das EPROM sowie das CPU-Taktverhältnis (intern/extern) eingestellt werden. Die übrigen auf diese Weise einstellbaren Parameter sind für die CPU280 unerheblich und deswegen nicht durch Steckbrücken zugänglich gemacht.

Der Systemtakt wird von der CPU durch den internen Quarzoszillator erzeugt. Die Frequenz dieses Taktes richtet sich nach der verwendeten CPU, sollte aber immer ein Vielfaches von 2,4576 MHz betragen, damit die CPU-interne serielle Schnittstelle mit einer Standard-Baudrate betrieben werden kann. Damit ergeben sich Taktfrequenzen von 9,8304 MHz oder 12,288 MHz (für die 10-MHz- bzw. 12,5-MHz-CPU). Sollten schnellere Versionen des Z280 erscheinen, so sind J1 und J2 auf eine Taktumsetzung von 1:2 zu stecken. Dadurch kann ein CPU-Takt von maximal 25 MHz erreicht werden, wobei die restliche Karte weiter mit 12,5 MHz arbeitet.

Die nicht kartenintern verwendeten CPU-Signale (DMA- und Timer-Steuersignale) sind auf kleine Lötbrücken geführt, welche für die Eingänge bereits mit Masse vorverbunden sind. Auf diese Weise bleiben keine CMOS-Eingänge unbeschaltet, und die Signale können an den Lötbrücken leicht identifiziert und abgegriffen werden, um sie z.B. über spezielle Leitungen an externe Peripherie zu führen.

Durch die interne MMU (Speicherverwaltung) verfügt der Z280 über einen Speicher-Adreßraum von insgesamt 16 MB, wobei für die beiden Hälften zu 8 MB jeweils einige Timing-Parameter unabhängig voneinander eingestellt werden können. Da nach einem Reset der Prozessor bei Adresse 0 mit der Programmausführung beginnt, wird die untere 8-MB-Hälfte vom EPROM belegt. Die obere Hälfte (ab 800000h) adressiert das auf der Karte befindliche dynamische RAM.

Der I/O-Adreßraum des Z280 ist ebenfalls 16 MB groß, wobei die oberen 8 Adreßbits vom 'I/O Page Register' gestellt werden. Die unteren 16 Adreßbits werden wie beim Z80 aus den CPU-Registern gebildet. Da alle I/O-Befehle offiziell nur 8 Adreßbits benutzen, werden auch auf der CPU280 bei allen I/O-Zugriffen die mittleren 8 Adreßbits (A8 bis A15) nicht dekodiert, d.h. ignoriert. Über die I/O-Page kann dann zwischen verschiedenen I/O-Gruppen gewählt werden, in denen jeweils 256 Adressen angesprochen werden können.

Die Busbreite des Z-Bus beträgt bei Speicherzugriffen 16 Bit und bei I/O-Zugriffen 8 Bit. Deswegen kann 8-Bit-Peripherie uneingeschränkt verwendet werden, obwohl der karteninterne Bus (zumindest für Speicher) 16 Bit breit ist.

Weitere Einzelheiten zum Z280 selbst, insbesondere Möglichkeiten und Programmierung der internen Bausteine, sind dem Datenblatt sowie dem 'Technical Manual' zu entnehmen.

2.2 EPROM

Das auf der Karte befindliche EPROM ist durch IC9 und IC10 realisiert. Es ergibt sich damit eine Kapazität von 32K x 16 Bit (d.h. 64 KB, bei Verwendung von 27C256) oder 64K x 16 Bit (128 KB, bei Verwendung von 27C512). Der verwendete Typ kann durch die Steckbrücke J5 gewählt werden. Mischbestückung und andere EPROM-Typen sind nicht vorgesehen.

Das EPROM belegt die physikalischen Adressen 000000-00FFFFh bzw. 000000-01FFFFh. Da keine weitere Dekodierung der höherwertigen Adreßleitungen erfolgt, tritt eine 'Adreßspiegelung' auf: das EPROM ist innerhalb der unteren 8 MB insgesamt 128 bzw. 64 mal erreichbar.

Da auf das EPROM üblicherweise nur beim Booten einmal kurz zugegriffen wird (um den Inhalt ins RAM zu übertragen), können hier problemlos Wait-States eingefügt werden, so daß die Zugriffszeit der EPROMs unkritisch ist. Die preiswerten 200- oder gar 250-ns-Typen sind hier völlig ausreichend. Bei maximaler Taktfrequenz (12.5 MHz) können 150-ns-Speicher ohne Wait-States, 200-ns-Typen mit einem und 250-ns-Typen mit zwei Wait-States betrieben werden. Die Anzahl der Wait-States für den EPROM-Bereich kann mit den Steckbrücken J3 und J4 eingestellt werden.

2.3 DRAM, Burst-Mode

Das dynamische RAM (IC11 bis IC18) bildet den Arbeitsspeicher der CPU280. Es können verschiedene Konfigurationen bestückt werden. Die verwendbaren RAM-Typen sind 1-MB-Chips und 4-MB-Chips der Organisation 256K x 4 bzw. 1M x 4 (Typen 514256 und 514400). Aus Platzgründen sowie wegen der Verfügbarkeit pinkompatibler Typen kommen RAMs im ZIP-Gehäuse zum Einsatz.

Die Mindestbestückung besteht aus vier 514256 (IC11..IC14), wodurch sich eine Kapazität von 512 KB ergibt. Standardmäßig werden acht dieser Speicher bestückt (1 MB). Darüberhinaus ist der Ausbau auf 2 MB und 4 MB möglich (4 bzw. 8 514400). Beim Wechsel der RAM-Typen ist die Programmierung von IC22 zu ändern. Das RAM belegt die physikalischen Adressen ab 800000h, bei Maximalausbau bis BFFFFFFh.

Die RAMs sollten Zugriffszeiten von 80 ns haben, um einen sicheren Betrieb bei maximalem Takt zu gewährleisten. Das RAM wird übrigens grundsätzlich ohne Wait-States betrieben. Es ist wegen der Timing-Steuerung für den Burst-Mode auch nicht erlaubt, Wait-States für die RAM-Zugriffe zu deklarieren.

Der Z280 verfügt über die Möglichkeit, beim Laden von Befehlen aus dem externen Speicher in den internen Cache (Opcode Fetch) mit einer speziellen Zugriffsart den Datentransfer zu beschleunigen. Hierzu wird zunächst ein normaler Speicherzugriff begonnen, bei dem aber der Inhalt von vier Speicherworten in vier aufeinanderfolgenden Taktzyklen (als 'Burst') übertragen werden, ohne weitere Adreßinformation von der CPU. Diese Zugriffsart (der Burst-Mode) ist etwa doppelt so schnell wie das Lesen einzelner Speicherworte. Für die Ausnutzung derartiger Mechanismen existieren spezielle RAM-Typen (Nibble-Mode-RAMs), welche bei aufeinanderfolgenden Zugriffen innerhalb einer Speicherzeile selbständig auf vier aufeinander folgende Adressen zugreifen. Da diese RAMs aber nicht in 4 Bit breiter Organisation verfügbar sind (und ansonsten immer mindestens 16 RAM-Chips nötig wären), erfolgt die Adreßberechnung bei der CPU280 außerhalb der RAMs durch eine entsprechende Hardware. Die RAMs werden dabei im 'Fast Page Mode' betrieben (Standard-Betriebsart) und erhalten bei jedem weiteren Zugriff die neue Adresse und einen CAS-Impuls. Dadurch ist es möglich, mit wenigen preiswerten Standard-RAMs die gleiche Leistung (bei größerer Flexibilität) zu erreichen wie mit vielen teuren Spezial-RAMs.

Das gesamte Timing für die dynamischen RAMs wird synchron erzeugt, d.h. einzig vom Prozessortakt gesteuert. Hieraus resultiert eine sehr große Betriebssicherheit auch bei stark schwankenden Umgebungsbedingungen (z.B. Temperatur, Betriebsspannung), sowie ein völlig abgleichfreier Betrieb. Lediglich die Dauer des CAS-Pulses im Burst-Mode wird durch eine Laufzeit erzeugt und kann durch Ändern von C6 optimiert werden (die Pulsdauer ist einfach zu kurz für eine synchrone Erzeugung).

2.4 ECB-Interface und I/O-Timing

Das ECB-Interface besteht im wesentlichen aus den Treibern für Daten-, Adreß- und Steuerbus (IC6 bis IC8), sowie der Takterzeugung mit IC4A. Grundsätzlich unterstützt dieser ECB-Anschluß nur I/O-Zugriffe, da Speicherzugriffe mit 16 Bit Breite bei 12.5 MHz dort schlicht unmöglich sind. Der Aufbau eines Interface zur Anpassung der Speicherzugriffe auf den ECB-Bus ist zwar möglich, aber recht aufwendig. Außerdem ist die Speicherkapazität der Karte bei vollem Ausbau wohl für alle Anwendungen ausreichend.

Es werden dementsprechend nur die Signale auf den Bus geführt, die dort für I/O-Zugriffe benötigt werden. Die Interrupt-Eingänge der CPU werden ungepuffert (aber mit Pull-Up) angeschlossen. Die ständig inaktiven Signale (/MRQ und /BUSAK) sind über normalerweise offene Lötbrücken auf VCC gelegt, so daß bei Fehlen einer geeigneten Bustimerierung diese Signale einen definierten Pegel erhalten können. Für externe Karten, welche dynamische Speicher enthalten (z.B. RAM-Disks) steht weiter ein Refresh-Signal zur Verfügung. Dieses wird alle 15.6 µs für die Dauer eines Speicherzyklus aktiv (240 ns bei 12.5 MHz), entsprechende Programmierung des CPU-internen Refresh-Controllers vorausgesetzt.

Die Ansteuerung externer I/O-Karten über den ECB-Bus erfolgt durch eine eigene I/O-Page des Z280, so daß der volle I/O-Adreßraum des Prozessors zur Verfügung steht. Aufgrund Platzmangels sowie des offiziellen I/O-Adreßraumes der Z80-Familie sind aber nur die unteren 8 Adreßbits (A0 bis A7) auf den Bus geführt, was zu 256 nutzbaren Adressen auf dem ECB-Bus führt. Dies dürfte wohl in den allermeisten Fällen mehr als genug sein.

Bei I/O-Zugriffen müssen generell 4 Wait-States eingefügt werden, um dem Bustiming gerecht zu werden. Dabei wird der CPU-Takt durch IC4A auf die Hälfte geteilt, um den Bus-Takt zu erzeugen. Dieser wird zu Beginn eines Zugriffs (durch /AS) so synchronisiert, daß die zeitliche Abfolge von Bustakt und Steuersignalen der des Z80 bei 6 MHz entspricht. Diese Synchronisation ist nötig, da manche Z80-Peripheriebausteine sonst nicht korrekt arbeiten (z.B. Z80-PIO).

Interrupts von ECB-Karten werden durch die CPU280 voll unterstützt. Die /INT-Leitung

des Bus wird auf den INTA-Eingang des Prozessors geführt und ist dort vektorisierbar. Die Interruptquellen auf der Karte belegen eigene INT-Eingänge, so daß der ECB-Bus-Interrupt auch unabhängig von den anderen Quellen geschaltet werden kann. Der Interrupt-Acknowledge-Zyklus weist durch Wait-States ebenfalls ein 6-MHz-Z80-Timing auf. Da der RETI-Zyklus als Speicherzyklus und somit für den Bus zu schnell abläuft, der Z280 im bevorzugten Interrupt-Mode 3 einen speziellen RETI-Befehl verwendet und darüberhinaus auch kein M1-Signal auf dem Bus dabei erzeugt wird, muß für den Bus ein langsamer Z80-RETI-Zyklus extra erzeugt werden. Dieser besteht aus zwei aufeinander folgenden Zugriffen jeweils mit /M1 und /RD aktiv sowie /IORQ inaktiv, bei denen auf dem Datenbus die Bytes EDh und 4Dh anliegen. Diese Signale werden auf der CPU280 durch Zugriffe auf eine spezielle I/O-Page erzeugt, bei denen die M1-Leitung des Bus aktiviert wird. Das Auslesen zweier Speicherzellen im NVRAM der Echtzeituhr (die natürlich die gewünschte Information beinhalten müssen) führt dann zu den richtigen und langsamen (I/O-Zugriff!) Signalen auf dem Bus. Auf diese Weise wird auch im Interruptbetrieb echte Bus-Kompatibilität zum 6-MHz-Z80 erreicht.

2.5 Echtzeituhr und NV-RAM

Als Echtzeituhr wird das IC DS 1287 A von Dallas verwendet (IC27). Dieses beinhaltet außer der eigentlichen Echtzeituhr mit Alarm und Kalender noch einen programmierbaren Rechteck/Interruptgenerator sowie 50 Byte nichtflüchtigen Speicher ('NVRAM'). Darüberhinaus enthält es gleich die zum Betrieb notwendige Lithiumbatterie, so daß keine Versorgung über externe Batterie oder gar über den ECB-Bus notwendig ist (bei der der Inhalt verlorengehen würde, sobald die Karte vom Bus abgezogen wird!).

Durch die Steckbrücke J10 kann der Baustein (nur bei abgeschalteter Versorgungsspannung) einen internen Reset erhalten. Der Interrupt-Ausgang ist auf den Interrupt B der CPU geführt (zusammen mit der seriellen Schnittstelle IC31) und erhält innerhalb der CPU einen festen Vektor. Der Rechteckausgang des RTC wird auf ein Bit des GPI (siehe 2.8) geführt und kann dort abgefragt werden.

Für weitere Details zu den RTC-Registern und deren Programmierung sei auf dessen Datenblatt verwiesen.

2.6 Floppycontroller

Als Floppycontroller kommt der FDC 37 C 65 B (IC28) zum Einsatz, welcher ohne jede weitere Hardware die Schnittstelle zwischen CPU und Laufwerken bildet. Dieser Baustein enthält den eigentlichen FDC (765 Kern), zwei Taktoszillatoren, eine PLL für die Lesedaten sowie alle nötigen Leistungstreiber (48 mA) für das FDD-Interface. Es können bis zu vier Laufwerke angeschlossen werden, wobei Größe und Format beliebig sind. Alle Formate nach dem IBM-3740-Verfahren können bearbeitet werden. Sogar die Verwendung von nicht geschwindigkeitsumschaltbaren HD-Laufwerken (5¼ Zoll) für normale DD-Formate wird unterstützt (durch den zweiten Quarzoszillator mit 9.6 MHz). Der Floppycontroller wird durch die CPU-interne DMA0 bedient, wodurch die Realisierung eines Hintergrundprozesses für Floppyzugriffe möglich wird. Der FDC erzeugt Interrupts auf der INTC-Leitung der CPU, welche ausschließlich für den FDC verwendet wird. Die Belegung der Steckerleiste für die Floppylaufwerke (CN2) entspricht dem Standard für 3½- bzw. 5¼-Zoll-Laufwerke. Zur Verwendung von 8-Zoll-Laufwerken sind die entsprechenden Leitungen im Anschlußkabel zu vertauschen.

Da üblicherweise recht kurze Anschlußkabel verwendet werden, sind die FDD-Signale mit 330Ω statt mit 150Ω abgeschlossen. Dadurch wird weniger Strom für die 'Heizung' verwendet. Sollen längere Kabel verwendet werden, kann RN3 durch ein entsprechend niederohmigeres Exemplar ersetzt werden.

Zum Betrieb von 5¼-Zoll-HD-Laufwerken ist auf der CPU-Karte ein Transistor nachzurüsten, welcher bei Betätigung durch das GPO-Signal 'DRV' Pin 2 des Laufwerks (LOW DENSITY) auf Low-Pegel legt (einfacher invertierender Schalter mit NPN-Transistor). In diesem Fall ist die Verbindung vom FDC zu diesem Pin natürlich zu trennen.

2.7 Serielle Schnittstellen

Die CPU280 beinhaltet zwei serielle Schnittstellen nach V.24. Eine wird durch den CPU-internen UART gebildet, während die zweite durch einen 'Twenty-Pin-UART' (TPUART, IC31) auf der Karte realisiert ist. Letzterer stellt alle Signale für eine serielle 4-Draht-Schnittstelle zur Verfügung (zwei Daten- und zwei Handshakeleitungen), wogegen die CPU nur die Daten liefert. Daher müssen die Handshake-Signale für die CPU-interne Schnittstelle durch Einzelbit-I/O innerhalb des GPIO erzeugt werden.

Beide Schnittstellen sind interruptfähig. Der CPU-interne UART belegt innerhalb der CPU-Bausteine einen eigenen Interruptvektor, der TPUART kann über die INTB-Leitung (welche er mit dem RTC teilt) Interrupts auslösen.

Beide Schnittstellen unterstützen Baudraten von 50 bis 38400 Baud (bei der CPU entsprechender Takt vorausgesetzt), als Datenformate sind jeweils 7 oder 8 Datenbits, 1 oder 2 Stopbits, optional Parity even oder odd möglich.

Die Pufferung und Pegelumsetzung der Schnittstellensignale erfolgt im Treiberbaustein IC29 (LT 1134). Dieses IC beinhaltet jeweils vier V.24-Sender und -Empfänger sowie die zum 5-V-Betrieb benötigten Spannungswandler.

2.8 General-Purpose-I/O (GPIO)

Zur Realisierung verschiedener Einzelbit-I/O-Aufgaben dient das mit IC25 und IC26 aufgebaute 'GPIO'. Die Bit-Ausgabe (GPO) erfolgt durch ein adressierbares 8-Bit-Latch (IC25). Hierbei werden sowohl die Adreßeingänge als auch der Datenbit-Eingang an den Adreßbus angeschlossen. Dadurch kann durch Schreibzugriffe auf verschiedene Adressen (unabhängig von den Daten am Datenbus) jeweils ein Bit gesetzt oder zurückgesetzt werden. Durch einen Reset werden alle Ausgänge auf 0 zurückgesetzt.

Vier der acht Ausgänge dienen verschiedenen Steuerungsaufgaben (Handshake für CPU-UART, Motorsteuerung für FDD, Signale für FDC), drei weitere Ausgänge steuern Leuchtdioden, welche z.B. zum Signalisieren von harten Fehlern dienen können. Ein Ausgang ist auf den Eingabeport rückgekoppelt. An dessen Signal kann nach einem 'Reset' festgestellt werden, ob dieser durch das Hardware-Signal RESET ausgelöst wurde oder nur durch einen Software-Einsprung in die Kaltstart-Routine.

Die Bit-Eingabe (GPI) erfolgt durch einen einfachen Bustreiber (IC26). Es werden außer dem Reset-Flag noch das CTS-Signal der seriellen Schnittstelle um den CPU-UART, das Rechtecksignal des RTC sowie drei durch den Anwender benutzbare Steckbrücken (J7 bis J9) eingelesen.

3. Bestückung, Inbetriebnahme

Die Bestückung der CPU280 erfolgt in der üblichen Reihenfolge: zunächst sollten die flachen Bauteile eingelötet werden, dann ist mit Bauteilen steigender Bauhöhe fortzusetzen. Beim Einbau der Quarze ist unbedingt eine isolierende Zwischenlage zu verwenden! Es eignen sich spezielle Kunststoff- oder Glimmerscheiben, zur Not läßt sich auch Gewebe-Tape verwenden. Da es für die ZIP-RAMs keine fertigen Sockel gibt, müssen zum Vermeiden direkten Einlötens SIL-Streifen verwendet werden. Die Bauhöhe der RAMs auf Sockelstreifen ist gerade klein genug, um die Platine in einen Standard-Einschub (4 TE) stecken zu können.

Der Stecker CN2 zum Anschluß der Floppylaufwerke kann wahlweise auf der Bestückungsseite (Bohrungen näher am Karteninneren) oder auf der Lötseite (näher am Kartenrand) angebracht werden. Dadurch ist es möglich, das Anschlußkabel entweder an der Lötseite oder an der Bestückungsseite vorbeizuführen (je nach Lage der CPU280 im 19-Zoll-Rahmen).

Für die Inbetriebnahme wird die CPU280 mit einem Test-Programm versehen (EPROMs), welches die vorhandene Hardware Stück für Stück testet und dabei Testinformationen über eine der seriellen Schnittstellen oder die LEDs auf der Karte abgibt. Nach Anlegen der Versorgungsspannung darf die vollbestückte Karte maximal 400 mA benötigen (am besten wird zunächst an einem Netzteil mit Strombegrenzung getestet). Einzelheiten zum Test-

programm sind dessen Beschreibung zu entnehmen.

Wie bereits erwähnt, ist ein Ausmessen der Dauer des CAS-Pulses im Burst-Mode empfehlenswert. Diese Zeit wird durch die Laufzeiten der GALs sowie das RC-Glied aus R8 und C6 bestimmt und sollte etwa 20 ns betragen. Gegebenenfalls ist C6 entsprechend zu verändern.

4. Verwendung der Steckbrücken (Jumper) und Lötbrücken

Auf der CPU280 befinden sich insgesamt 10 Steckbrücken, mit denen verschiedene Parameter (zur Systemkonfiguration) eingestellt werden können. Darüberhinaus existieren noch 12 Lötbrücken, mit denen optional bestimmte Signale mit festen Potentialen verbunden werden können. Die Brücken sind im Folgenden aufgeführt:

J1,J2 Diese beiden Brücken bestimmen das Taktverhältnis der CPU (Verhältnis interner zu externer Takt). Hierbei gilt:

J1 gesteckt	J2 gesteckt	Verhältnis 2:1
J1 offen	J2 gesteckt	Verhältnis 1:1
J1 gesteckt	J2 offen	Verhältnis 4:1
J1 offen	J2 offen	verboten

J3,J4 Mit diesen beiden Brücken wird die Anzahl der Wait-States bei EPROM-Zugriffen wie folgt eingestellt:

J3 gesteckt	J4 gesteckt	0 Wait-States
J3 offen	J4 gesteckt	1 Wait-State
J3 gesteckt	J4 offen	2 Wait-States
J3 offen	J4 offen	3 Wait-States

J5 Auswahl des EPROM-Typs:
 J5 in Richtung CPU: 27 C 256 (vorverbunden)
 J5 in Richtung RAM: 27 C 512

J6 Prekompensation des FDC auf inneren Spuren:
 J6 gesteckt: 187 ns
 J6 offen: 125 ns

J7,J8,J9 Durch GPI einlesbare Anwender-Konfigurationsbrücken

J10 Interner Reset des RTC (nur bei abgeschalteter VCC)

LJ1 Verbindung des ECB-Signals /MRQ mit VCC
 LJ2 Verbindung des ECB-Signals /BUSAK mit VCC
 LJ3..5 Verbindung CPU-Pin CTIO0..2 mit GND
 LJ6..8 Verbindung CPU-Pin CTIN0..2 mit GND (vorverbunden)
 LJ9..10 Verbindung CPU-Pin /DMASTB1, /DMASTB0 mit GND
 LJ11..12 Verbindung CPU-Pin /RDY1..2 mit GND (vorverbunden)

A1. Stückliste

IC1		Z 280 MPU, 12.5 MHz (oder 10 MHz)
IC2,IC3		74 HCT 373
IC4		74 HCT 74
IC5		TL 7705
IC6,IC8		74 ACT 244 (oder ALS)
IC7		74 ACT 245 (oder ALS)
IC9,IC10		27 C 256 oder 27 C 512
IC11...IC18		514256-80 oder 514400-80 (ZIP)
IC19,IC20		74 ACT 158 (oder ALS)
IC21...IC24		GAL 16 V 8 Q-25
IC25		74 HCT 259
IC26		74 HCT 367
IC27		DS 1287 A (oder MK 48 T 87 A)
IC28		FDC 37 C 65 B (PLCC)
IC29		LT 1134
IC30		74 HCT 175
IC31		COM 81 C 17
IC32		74 HCT 14
T1		BC 547 B o.ä.
D1,D2,D3		LED 3 mm rot
Q1		24.576 MHz HC-18U (oder 19.6608 MHz)
Q2		16.000 MHz HC-18U
Q3		9.600 MHz HC-18U
Q4		5.0688 MHz HC-18U
C1,C2		10p ker. RM 5 (18p bei 19.6608 MHz Q1)
C3		10µ Ta. RM 2.5
C4		100n ker. RM 5
C5		1n ker. RM 5
C6		120p ker. RM 5 (am besten ausmessen)
C7		15p ker. RM 5
C8		47p ker. RM 5
C9		56p ker. RM 5
C10		68p ker. RM 5
C11...C14		1µ Ta. RM 2.5
CK1...CK27		25x 100n ker. RM 5, 2x 10µ Ta. RM 2.5
R1,R2		4 k 7
R3,R9,R10		1 k
R4...R7		3 k 3
R8		100 R
RN1		RSIL 5x 2 k 2
RN2,RN6		RSIL 4x 3 k 3
RN3		RSIL 5x 330 R
RN4		RSIL 5x 4 k 7
RN5		RSIL 3x 330 R (aus Einzelwiderständen)
CN1	ECB	VG-Leiste 64-pol. ac-bestückt (DIN 41612 C)
CN2	FDD	Pfosten 50-pol. (2x25), wahlweise auf Löt- oder Bauteilseite
CN3	V.24	Pfosten 20-pol. (2x10)
J1...J4		Pfosten 2x4, CPU-Init
J5		Pfosten 1x3, EPROM-Typ
J6...J9		Pfosten 2x4, FDC-Precomp und User-Jumper
J10		Pfosten 1x2, RTC-Reset

A2. Steckerbelegungen

Steckerbelegung CN1 (ECB-BUS):

a	Nr	c
+5V	1	+5V
D5	2	D0
D6	3	D7
D3	4	D2
D4	5	A0
A2	6	A3
A4	7	A1
A5	8	
A6	9	A7
/WAIT	10	
	11	IEI
	12	
	13	
	14	D1
	15	
2xCLK	16	IEO
	17	
	18	
	19	
/M1	20	/NMI
	21	/INT
	22	/WR
	23	
	24	/RD
	25	
	26	/RESOUT
/IORQ	27	
/RFSH	28	
	29	CLK
	30	(/MRQ)
(/BUSAK)	31	/RESIN
GND	32	GND

Steckerbelegung CN2 (Floppy-Laufwerke):

Maxi Mini	Signal	Umbelegung für 8"
(1..49 ungerade: GND)		
2	RWC/RPM	
4	MOTOFF	
6	MOTOFF	
8	MOTOFF	
10	nc	
12	DCHG	
14	SIDE SEL	
16	HDLD	
18	2 RWC/RPM	HDLD (16,20)
20	4 HDLD	INDEX (24)
22	6 DS3	nc
24	8 INDEX	MOTOFF (4,6,8)
26	10 DS0	
28	12 DS1	
30	14 DS2	
32	16 MOTON	DS3 (22)
34	18 DIRECTION	
36	20 STEP	
38	22 WRITE DATA	
40	24 WRGATE	
42	26 TRK0	
44	28 WRPROT	
46	30 READ DATA	
48	32 SIDE SEL	(nc)
50	34 DCHG	(nc)

Steckerbelegung CN3 (V.24-Schnittstellen):

Belegung ermöglicht direktes Aufquetschen zweier DSUB-9, deren Pin-Nummern mit der DSUB-25 Belegung übereinstimmen.

1	1	GND	Masse (Schutzmasse 1)
2	6	nc	
3	2	TXD 1	Sendedaten 1
4	7	GND	Masse (Signalmasse 1)
5	3	RXD 1	Empfangene Daten 1
6	8	nc	
7	4	RTS 1	Request-to-Send 1
8	9	nc	
9	5	CTS 1	Clear-to-Send 1
10	-	nc	
11	1	GND	Masse (Schutzmasse 2)
12	6	nc	
13	2	TXD 2	Sendedaten 2
14	7	GND	Masse (Signalmasse 2)
15	3	RXD 2	Empfangene Daten 2
16	8	nc	
17	4	RTS 2	Request-to-Send 2
18	9	nc	
19	5	CTS 2	Clear-to-Send 2
20	-	nc	

A3. GAL-Programmierung

TITLE CPU280 RAM-TIMING AND NIBBLEMODE IC21
 AUTHOR TILMANN REH
 COMPANY REHDESIGN
 DATE 12.06.1990

CHIP Z280RAM PALCE16V8

NCLK A3 A1 A2 A4 IE DS OE MQD GND
 QOE MUX WR MA0 MA1 NC1 NC2 MQA FFR VCC

EQUATIONS

$/WR = /OE * /DS$
 $MA0 := /MQA * A3 + MQA * /MUX * A1 + MUX * /MA0$
 $MA1 := /MQA * A4 + MQA * /MUX * A2 + MUX * MA0 + MUX * MA1$
 $/FFR = IE * DS * MQD$
 $NC1 = GND$
 $NC2 = GND$

TITLE CPU280 CAS-DECODER IC22
 AUTHOR TILMANN REH
 COMPANY REHDESIGN
 DATE 12.06.1990

CHIP Z280CAS PALCE16V8

CLK RW BW RFSH A0 A19 A20 A21 A22 GND
 MUX BR F NCK CAS1L NCLK CAS0H CAS0L MQA CAS1H VCC

EQUATIONS

$BRF = /RFSH$
 $NCLK = /CLK$

$/CAS0L = MUX * /RFSH * /A20 * /A19 * BW * A0$
 $+ MUX * /RFSH * /A20 * /A19 * /BW * CLK$
 $+ MUX * /RFSH * /A20 * /A19 * /BW * NCK$
 $+ MUX * /RFSH * /A20 * /A19 * /BW * /RW$
 $+ RFSH * /MUX * MQA$

$/CAS0H = MUX * /RFSH * /A20 * /A19 * BW * /A0$
 $+ MUX * /RFSH * /A20 * /A19 * /BW * CLK$
 $+ MUX * /RFSH * /A20 * /A19 * /BW * NCK$
 $+ MUX * /RFSH * /A20 * /A19 * /BW * /RW$
 $+ RFSH * /MUX * MQA$

$/CAS1L = MUX * /RFSH * /A20 * A19 * BW * A0$
 $+ MUX * /RFSH * /A20 * A19 * /BW * CLK$
 $+ MUX * /RFSH * /A20 * A19 * /BW * NCK$
 $+ MUX * /RFSH * /A20 * A19 * /BW * /RW$
 $+ RFSH * /MUX * MQA$

$/CAS1H = MUX * /RFSH * /A20 * A19 * BW * /A0$
 $+ MUX * /RFSH * /A20 * A19 * /BW * CLK$
 $+ MUX * /RFSH * /A20 * A19 * /BW * NCK$
 $+ MUX * /RFSH * /A20 * A19 * /BW * /RW$
 $+ RFSH * /MUX * MQA$

TITLE CPU280 SYSTEM-SIGNALS IC23
 AUTHOR TILMANN REH
 COMPANY REHDESIGN
 DATE 12.06.1990

CHIP Z280SYS PALCE16V8

ST3 ST2 ST1 ST0 AS DS A21 A22 A23 GND
 MQA MDA NAS ROM BIO M1 RFSH NNAS XIO VCC

EQUATIONS

NAS = /AS
 NNAS = NAS
 /ROM = ST3 * /A23
 MDA = ST3 * NNAS * A23 * /A22 * /A21
 + /ST3 * /ST2 * /ST1 * ST0 * NNAS
 + MQA
 RFSH = /ST3 * /ST2 * /ST1 * ST0
 /BIO = /ST3 * /ST2 * ST1 * /ST0 * /A23 * /A22 * /A21 * /DS
 + /ST3 * ST2 * /ST1 * /ST0
 /XIO = /ST3 * /ST2 * ST1 * /ST0 * /A23 * A22
 + /ST3 * /ST2 * ST1 * /ST0 * A23 * /A22
 /M1 = /ST3 * ST2 * /ST1 * /ST0
 + /ST3 * /ST2 * ST1 * /ST0 * A23 * /A22

TITLE CPU280 IO-ADRESS-DECODER IC24
 AUTHOR TILMANN REH
 COMPANY REHDESIGN
 DATE 12.06.1990

CHIP Z280IO PALCE16V8

NC NC XIO DS A6 A5 A7 OE IE GND
 NC UART GPO GPI RTC DACK LDOR LDRSR FDC VCC

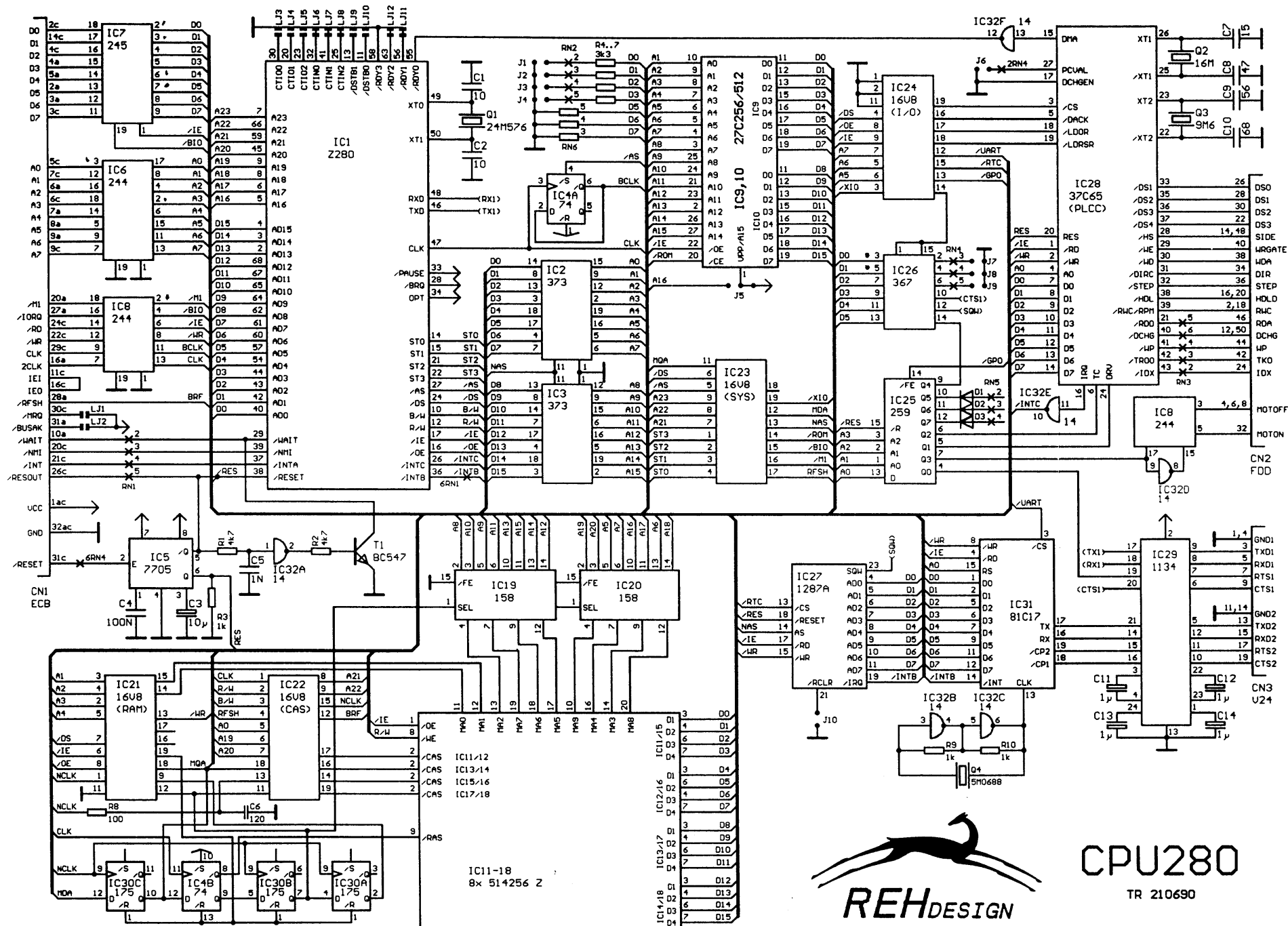
EQUATIONS

/RTC = /XIO * /A7 * /A6
 /FDC = /XIO * /A7 * A6 * /A5
 /DACK = /XIO * /A7 * A6 * A5
 /LDOR = /XIO * A7 * /A6 * /A5
 /LDRSR = /XIO * A7 * /A6 * A5
 /UART = /XIO * A7 * A6 * /A5
 /GPI = /XIO * A7 * A6 * A5 * /IE * /DS
 /GPO = /XIO * A7 * A6 * A5 * /OE * /DS

A4. Organisation Adreßräume Speicher/IO

Speicher:	000000-01FFFF 800000-BFFFFFFF	Boot/System-EPROM max. 4 MB DRAM Onboard
I/O:	00xx00-00xxFF 40xx00-40xxFF 80xx00-80xxFF FExx00-FFxxFF	ECB-Bus (256 Adressen) Onboard-I/O Onboard-I/O mit /M1 auf ECB-Bus Onchip-I/O Z280
Onboard-I/O:	00-3F 40,41 60 80 A0 C0 E0-EF	RTC / NVRAM FDC FDC-DACK FDC-LDOR FDC-LDRSR UART GP-I/O
GP-Output:	E0 / E1 E2 / E3 E4 / E5 E6 / E7 E8 / E9 EA / EB EC / ED EE / EF	RTS1 off / on DRV off / on TC off / on MOTOR off / on WARM off / on LED1 on / off LED2 on / off LED3 on / off
GP-Input:	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5	User-Jumper 1 (J7) User-Jumper 2 (J8) User-Jumper 3 (J9) CTS1 SQW WARM
Interrupts:	NMI INTA INTB INTC	über ECB-Bus (intern vektorisiert) über ECB-Bus (interne oder externe Vektoren) RTC, TPUART (intern) FDC (intern)

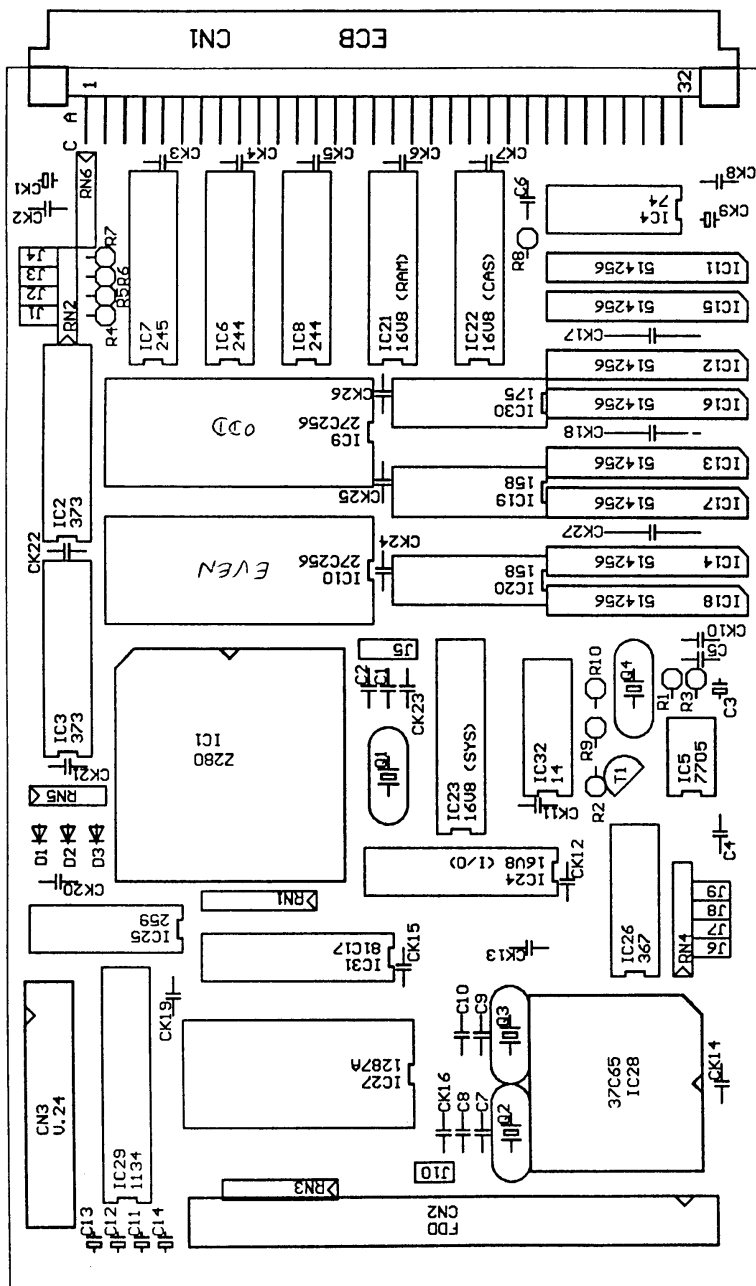
A5. Schaltbild



CPU280

TR 210690

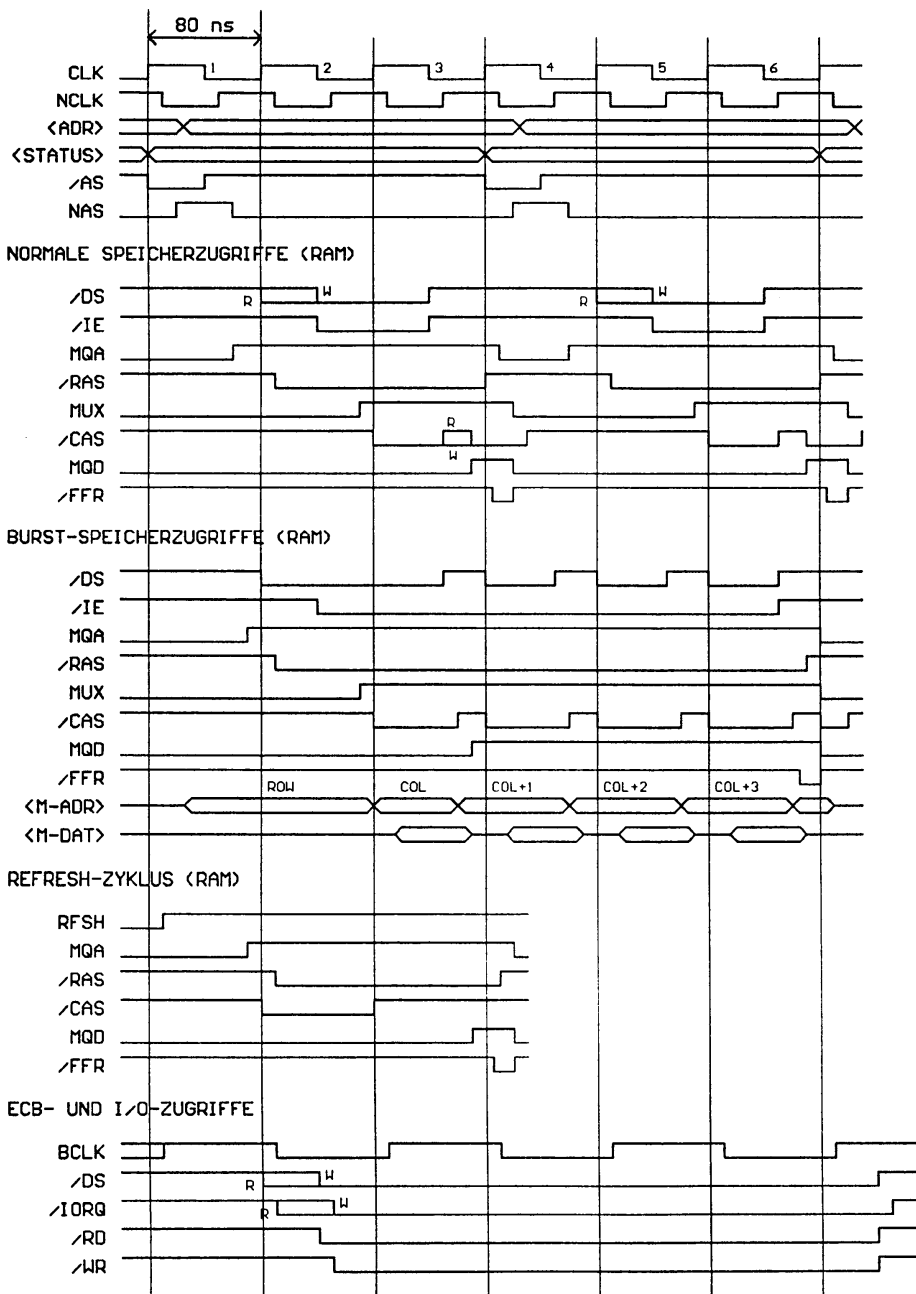
A6. Bestückungsplan



A7. Timing-Diagramme Speicherzugriffe, ECB-Bus

Timing CPU280, 12.5 MHz

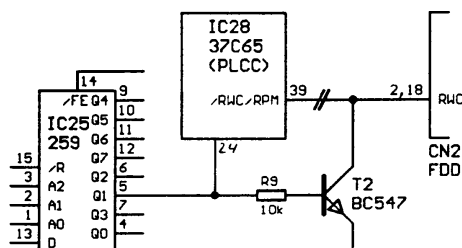
TR 260690



Addendum

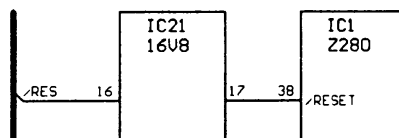
1. Schreibstrom-Umschaltung

Wie bereits in 2.6 erwähnt, ist zum Betrieb von 5¼-Zoll-HD-Laufwerken ein Transistor nachzurüsten, der den Schreibstrom am Laufwerk entsprechend der Dichte umschaltet. Der entsprechende Ausgang des FDC (/RWC) erfüllt diese Funktion leider nur im PC-AT-Modus, in dem nur zwei Laufwerke adressierbar sind.



2. Reset-Signal

Um eine der Z280-Spezifikation entsprechende Anstiegszeit des Reset-Signals zu gewährleisten, sollte das Reset-Signal für die MPU durch Zwischenschalten eines Puffers (innerhalb von IC21) aufbereitet werden. Dazu sind die beiden Leiterbahnen am Reset-Pin der MPU aufzutrennen und miteinander zu verbinden. Den Reset-Eingang an IC21 verbindet man am einfachsten mit dem entsprechenden Pin des Bussteckers, den Ausgang von IC21 direkt mit der MPU.



Die Programmierung des RAM-GAL's IC21 wird wie folgt erweitert:

2. Zeile Pinliste: QOE MUX WR MA0 MA1 RES CPURES MQA FFR VCC
Gleichungen: CPURES = RES

3. GAL-Programmierung System-GAL

Die Programmierung des System-GAL's IC23 wurde wie folgt geändert:

/ROM = ST3 * /A23 * /DS
/XIO = /ST3 * /ST2 * ST1 * /ST0 * /A23 * /A22 * A21
+ /ST3 * /ST2 * ST1 * /ST0 * /A23 * A22 * /A21
/M1 = /ST3 * ST2 * /ST1 * /ST0
+ /ST3 * /ST2 * ST1 * /ST0 * /A23 * /A22 * A21

Dadurch ändert sich auch die Belegung der I/O-Pages:

00xx00-00xxFF	ECB-Bus (256 Adressen)
20xx00-20xxFF	Onboard-I/O mit /M1 auf ECB-Bus
40xx00-40xxFF	Onboard-I/O
FExx00-FFxxFF	Onchip-I/O Z280

1. Allgemeines

Die vorliegende Version des Betriebssystems ist eine zwar voll funktionsfähige, aber noch nicht voll ausgebaute Beta-Version. 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 (Schnittstellen und Laufwerke). Mit Rücksicht auf spätere Updates meinerseits empfehle ich jedem, 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 Treiberroutrinen kann deren Ansteuerung ersehen werden; u.U. lassen sich durch wenige Änderungen andere Karten verwenden.

2. Bootlader

Nach einem Reset wird zunächst anhand der Echtzeituhr die Taktfrequenz des Prozessors ermittelt. Ist die Echtzeituhr nicht ansprechbar, wird eine Frequenz von 12.288 MHz angenommen. Danach wird die Konsolenschnittstelle initialisiert (entsprechend den Werten im Setup; bei Inbetriebnahme $8 + n + 1$ mit 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. Verläuft der RAM-Test fehlerfrei, so wird der Bootlader ins RAM kopiert und dort gestartet. Die auf der Karte befindlichen LED's zeigen den Zustand des Rechners während dieser Operationen an. Nach Reset leuchten zunächst (hardwaredmäßig) alle LED's. Unmittelbar darauf wird eine LED gelöscht, die zweite nach Erkennung der Taktrate und die dritte nach erfolgreichem Ablauf des RAM-Test. Weitere Hardware-Tests sind vorgesehen, aber derzeit noch nicht implementiert.

Im Allgemeinen (d.h. greift man nicht in den Bootvorgang ein) 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, 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'.

Drücken der Leertaste während des RAM-Test verhindert das Löschen der RAM-Disk(s) beim Booten, durch Betätigen von ESC wird statt aus dem EPROM von Diskette gebootet (Bootlaufwerk wurde im Setup angegeben, Default ist A:).

Da verschiedene Konsolenschnittstellen zur Verwendung kommen können und während der ersten Meldungen des Bootladers (und zur Initialisierung) noch kein RAM verwendet werden darf (noch nicht getestet!), sind die benötigten Funktionen Initialisieren, Zeicheneingabe 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. AutoFormat-System

Sowohl Bootlader als auch Betriebssystem verfügen über eine AutoFormat-Funktion, d.h. passen sich an das aktuelle Diskettenformat an. Hierzu befindet sich auf der Diskette in Spur 0, Sektor 1 ein 128 Byte großer Parameterblock, der das physikalische und logische Diskettenformat vollständig beschreibt.

Wird ein Diskettenwechsel festgestellt, 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. Die Default-Einstellung wird im Lader-BIOS bzw. im Tabellen-Modul 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 das Format 'Reh-CP/M 3 V2.1' mit 400 Blocks, 256 Directory-Einträgen und 1024 Byte großen Sektoren eingetragen. 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 auf Diskette hat den folgenden Aufbau:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	21	AE	BF	- D P B -												
10	- D P B -			psec	ptrk	tk0a	skwa	offa	t1a	tk0b	skwb	offb	t1b	Flag	Flg2	
20	21	- N A M E -														
30	- N A M E -				frei (00)										S1	S33
40	21	Sektortabelle (64 Sektoren)														
50																
60	21															
70																

Aufbau des Bit-Flags (Position 1Eh):

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

Die Angaben in den 4-Byte-Feldern 'tk0 skw off t1' beschreiben die Spurabfolge bzw. Spurübersetzung. Da allerdings nicht alle Spurübersetzungen auf diese Weise parametrisierbar sind, wird noch nach anderen Methoden gesucht, so daß an dieser Stelle Änderungen der Parameterblockdefinition wahrscheinlich sind. Für sachdienliche Hinweise bin ich jederzeit dankbar.

4. System-Generierung

Zunächst müssen die BIOS-Quelltexte assembliert werden. Da es sich um echte Z280-Quelltexte handelt, muß zunächst eine Vorbehandlung durch PRE280 erfolgen. Dieses Programm erzeugt einen Zwischen-Quelltext (.MAC) und setzt die Z280-Anweisungen darin in Z80-Befehle und DB/DW-Pseudobefehle um. Diese Zwischendatei kann dann von einem normalen Z80-Assembler bearbeitet (und danach gelöscht) werden. Die Submit-Datei

ASM verbindet diese beiden Aufrufe.

PRE280 ist ein freies Programm von Axel 'Fifi' Zinser und liegt bei. Allerdings dürfen ohne seine ausdrückliche Zustimmung keine Änderungen daran gemacht und weitergegeben werden.

Zum Erzeugen einer neuen System-Datei (CPM3.SYS) sind zunächst die Module des BIOS zu einer 'SPR'-Datei (System Page Relocatable) zusammenzubinden. Dies muß mit dem DRI-Linker LINK erfolgen, da nur dieser korrektes SPR-Format erzeugt. Danach ist mit GENCPM unter Verwendung der BDOS-SPR-Dateien die Systemdatei zu erstellen. GENCPM verwendet hierbei zunächst die Einstellungen aus GENCPM.DAT, wobei man die gewählten Puffergrößen frei nach eigenem Geschmack verändern kann.

Aufgrund einer ungünstigen Programmierung des residenten BDOS beim Zugriff auf die Hash-Tabellen darf aber unter keinen Umständen das Hashing für irgendein Laufwerk aktiviert werden! Die Bank-Umschaltung meines BIOS unterstützt diese Art Zugriffe nicht und es kommt zwangsläufig zu Abstürzen. Später wird es von mir ein geändertes residenten BDOS geben, welches Hashing erlaubt und darüberhinaus eine etwa 1k größere TPA bietet.

Das Linken des BIOS und der Aufruf von GENCPM sind in der Submit-Datei GEN zusammengefaßt. Hier kann auch die Reihenfolge der Module beim Linken ersehen werden.

Der Bootlader wird nur aus den Teilen Lader-BIOS, Lader-DOS und Setup gelinkt. Als Ergebnis entsteht eine normale Objektcode-Datei (COM oder CIM). Dieser Linker-Aufruf (ich verwende SLRNK) ist in der Submit-Datei GENL untergebracht.

Soll das erzeugte System aus dem EPROM gebootet werden, so kann man aus den Dateien LOADER.CIM, CCP.COM und CPM3.SYS mit Hilfe des Programms GENEPR eine Datei SYSTEM.EPR erzeugen. Diese enthält die genannten drei Dateien des Betriebssystems an bestimmten Adressen, passend für den Bootlader. Zum Programmieren der EPROM's muß sie allerdings noch durch SPLIT16 in gerade (EVN) und ungerade (ODD) Hälfte aufgeteilt werden. Die Pascal-Quelltexte von GENEPR und SPLIT16 liegen bei.

5. 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 abgewartet, bei der man zwischen Abbruch, Ignorieren und Wiederholen wählen kann. Die Wiederholen-Funktion arbeitet leider noch nicht unter allen Umständen fehlerfrei, es kann daher zu einem 'harten' Fehler bei der FDC-Ansteuerung kommen. In diesem Fall (Meldung DIO-Error 2) hilft nur noch Reset. An der Beseitigung wird gearbeitet.

Treten Fehler innerhalb der CPU auf (Traps, z.B. bei Division durch Null oder beim Versuch, einen privilegierten Befehl auszuführen), so wird eine entsprechende Fehlermeldung ausgegeben und der Kommandoprozessor gestartet. Man befindet sich also wieder auf der Betriebssystem-Ebene.

Auch lassen sich Programme z.B. aus Endlosschleifen recht einfach mit NMI abbrechen, hier wird (ohne weitere Meldung) ebenfalls der Kommandoprozessor gestartet.

6. 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, welche bei Aufruf des Programms zunächst auf dem Default-Laufwerk, dann auf A: gesucht wird. Ändert man Formatdefinitionen oder fügt neue hinzu, so wird die Datei erst dann geschlossen, wenn der Format-Manager verlassen wird. Die Diskette mit der Formatdaten-Datei darf also zwischendurch auf keinen Fall gewechselt werden.

Die Verwaltung der Formatdaten beschränkt sich auf ein sinnvolles Minimum. Daher können Formate nicht beliebig wieder gelöscht oder umsortiert werden, sondern es können nur vorhandene Formatdefinitionen geändert oder neue hinzugefügt werden. Deswegen ist es sinnvoll, zum Ausprobieren fremder Formate immer dieselbe Formatnummer zu verwenden, damit die Datei nicht unnötig groß wird.

Eine formatierte Tabelle aller definierten Formate läßt sich vom Format-Manager ausgeben. Der Drucker sollte hierzu vorher auf Schmalschrift eingestellt werden, da etwa 120 Zeichen pro Zeile gedruckt werden. Aufgrund verschiedener Ansteuerung der diversen Drucker erfolgt diese Einstellung nicht durch den Format-Manager.

7. Bevorzugte Formate

Zu bevorzugen sind alle Formate, welche auf legalem und möglichst einfachem Wege eine optimale Disketten-'Performance' erzielen. Dies bedeutet für die einzelnen Parameter:

- Sektorgröße möglichst 1k, Sektoranzahl maximal,
- Physikalischer Skew 1,
- logischer Skew 1 bei Multi-Sector-I/O, ansonsten 2,
- Null Offsetspuren
- bei 2 Seiten: Sektor- oder Halbspurüberlauf
- Blockgröße möglichst 2k, evtl. 4k (bei großen Kapazitäten)

Bei Unklarheiten bitte ich, den Uwe Herczeg oder mich anzusprechen, damit nicht auch unter uns die Formatvielfalt unsinnig groß wird.

Vorschläge zur Gestaltung des noch zu erstellenden Software-Handbuch (Sowie evtl. Änderungswünsche zum Hardware-Handbuch) sind zwang- und formlos an mich zu richten.

Historie

Ab hier zeigen die Texte einen kleinen Überblick der Entwicklung 'CPU280' auf, wie sie sich mir darstellte.

Die Kommunikation war nicht nur in Briefform sondern auch natürlich telefonisch, auf Treffen und ganz modern über ein Forum bei compuserve.

Beispiel:

** Nr. 18 (Posteingang) **

Datum: Fri, 18 May 90 23:24:24 +0200

Von:

"Tilman Reh" <tilmann@cosmo.uucp>

An:

uweduc

Betreff: CPU280

EB an: tilmann@cosmo.uucp

Hallo Uwe,

hier sind alle bisher Beteiligten, aber auch die nur Interessierten am Projekt CPU280. Ich denke, zu dem Treffen sollten wir ruhig alle einladen.

....

Die weiteren Scans sind nicht mehr so gut lesbar da 4 Blätter auf eine A4 Seite gedruckt wurde und so der Text sehr klein ist.

Hallo, Z280-Interessierte (bzw. Begeisterte)!

hier also endlich mein Rundschreiben bezüglich der von mir entwickelten Z280-ECB-Karte (kurz: CPU280). Wie einige wohl schon wissen, ist das Layout der Platine inzwischen abgeschlossen und die Prototypen sind in der Herstellung. Ich rechne etwa Mitte Januar mit den Platinen und werde dann den 'Nulltyp' aufbauen und testen sowie mit der BIOS-Programmierung beginnen.

Wie Ihr an den Unterlagen sehen könnt, habe ich an der Schaltung einen kleinen Abstrich machen müssen: Die Steckerleisten für eine Huckepack-Karte sind ersatzlos gestrichen. Nur so war ein zweistufiges Layout (in angemessener Zeit) möglich. Multilayer wäre bestimmt wegen der Kosten auch bei Euch abgelehnt worden. Also lieber die Leisten streichen, die ich bei meinem jetzigen 180er System sowieso niemals vermisst habe.

Die restlichen Leistungsmerkmale sind aber voll beibehalten worden. Einige kleinere Schaltungsdetails (RAM-Timing, Echtzeituhr) sind gegenüber dem Erst-Entwurf geringfügig geändert. Alles in allem dürfte diese Karte vom Preis-Leitungs-Verhältnis nicht gerade schlecht abschneiden. Die Platine müßte sich nach meiner Schätzung für etwa 50 DM herstellen lassen, die Bauteile kosten etwa 350 DM zuzüglich RAM (zur Zeit etwa 100 DM pro 512 KB).

Wo ich gerade bei den Kosten bin: Es wurde angeregt, alle Interessierten sollten vorab einen gewissen Betrag zur Verfügung stellen, damit eine gewisse Sicherheit in die Finanzen kommt. Normalerweise springt nach einem gewissen Geldeinsatz niemand ohne triftigen Grund wieder ab und läßt die anderen auf den Kosten sitzen. Ich halte hier einen Betrag von 250 DM für sinnvoll. Bitte schickt mir umgehend einen Scheck über diesen Betrag zu. Der wird erst eingelöst, wenn Ihr die Sachen bekommt (wie bei den Software-Bestellungen über Helmut). Diese Lösung müßte eigentlich von allen ernsthaft Interessierten bejaht werden.

Zur Materialbeschaffung wäre zu sagen, daß es wohl das günstigste ist, wenn ich selbst das ganze Zeug einkaufe und dann zusammen mit den Platinen und den programmierten GAL's an Euch weiterschicke. Ich schlage vor, für alle (Scheck-) Beteiligten jeweils einen vollständigen Satz aller Halbleiter, IC-Fassungen und Steckverbinder sowie der Widerstandsnetzwerke und Quarze einzukaufen bzw. zusammenzustellen. Der ganze Kleinkram macht preislich recht wenig aus, so daß auch diejenigen mit einem gewissen Bauteilvorrat eigentlich damit zufrieden sein müßten. Widerstände und Kondensatoren muß ich wohl nicht einzeln um-packen. Ich bitte Euch, hierzu Eure Meinung abzugeben.

Und nun, wie angedroht, die Beschreibung meines bisherigen AutoFormat-Systems unter CP/M-Plus. Verwendeter Formalismus: Seitenzählung 0..1, Spurzahl 0..max, Sektorzählung 1..max.

Wesentliches Merkmal ist dabei das Speichern aller zum Bearbeiten der Diskette notwendigen Daten auf der Diskette selbst. Dazu werden die ersten 128 Byte des Sektors 1 auf Spur 0 (Seite 0) verwendet. Beim 'kalten' Einloggen des Laufwerks wird diese Information gelesen und die CP/M-Parametertabellen entsprechend angepaßt. Beim Lesen des Parameterblocks paßt sich der FDC selbständig an Schreibdichte und Sektorgröße an, bei der anschließenden Bearbeitung der Diskette wird nur noch der Parameterblock als Grundlage benutzt. Dadurch ist auch das Bearbeiten von Formaten möglich, die auf Spur 0 eine andere Dichte oder Sektorgröße aufweisen (z.B. ECMA).

Aufbau des Disketten-Parameterblocks:

Adresse	Inhalt
00	Identifikationsbyte 5Eh
01..11	CP/M-DPB (17 Byte)
12	Anzahl Spuren (Bits 0..6), TPI-Flag (Bit7: 0=48, 1=96)
13	Flag Invers/Normal
14	Density Flag (00h=FM, 40h=MFM)
15..1D	00h
1E	Skewtable(1)
1F	Skewtable(33)
20	5Eh
21..3F	Skewtable(2..32)
40	5Eh
41..5F	Skewtable(34..64)
60	21h
61..75	Name des Formates (string20)
76..7F	reserved

Wird kein Parameterblock gefunden (oder kein Sektor 1), so werden die aktuell eingestellten Parameter nicht verändert. Mit Hilfe des Format-Managers (ein zum System gehörendes Programm) können diese auch unmittelbar geladen werden, so daß auch Fremdformate ohne Parameterblock bearbeitet werden können (wichtig, wenn Sektor 1 schon beschrieben ist!). Auch nachträgliches Schreiben von Parameterblöcken ist möglich.

Die Identifikationsbytes 5Eh bzw. 21h werden vom Betriebssystem innerhalb der Directory ignoriert, so daß auch Formate mit 0 Offsetspuren kein Problem darstellen. Allerdings werden dann bei 'SHOW USER' unter User 14 '0 Files' angezeigt, weshalb in Zukunft dieses Byte auf 60h gelegt werden wird (wird vom CP/M total ignoriert).

Übrigens wird bisher beim ersten Laufwerkszugriff nach dem Booten auch die Größe des Laufwerks (5"/8") anhand des zeitlichen Abstandes der Indexpulse ermittelt. In Zukunft wird dies wohl über ein Setup eingestellt und im NVRAM gespeichert werden.

Die speziellen Funktionen, die der Format-Manager zum Bearbeiten der Disketten benötigt (formatieren, testen, ID lesen, Parameter lesen/schreiben usw.) werden über die USER-Funktion des BIOS realisiert. Hierbei dient das Register DE als Funktionsnummer und BC als Argument (bzw. Zeiger auf mehrere Argumente). Ergebnisse werden in Speicherbereichen oder Registern übergeben (ähnlich wie bei BDOS-Aufrufen). Nur diese enge Einbindung in das BIOS erlaubt auch dem Format-Manager die volle Ausnutzung der vorhandenen BIOS-Unterprogramme, z.B. zur Interruptbehandlung, sowie die Portierbarkeit auf verschiedene Systeme.

Bisherige Funktionen der BIOS-USER-Funktion:

0	Return BIOS Version Number
1	Set Motor-Off Delay
2	Wait for Ready / Get Size
3	Set Physical Track and Seek
4	Read ID Information
5	Read Physical Sector
6	Write Physical Sector
7	Format a Track
8	Motor Off
9	Set BIOS Parameter Blocks
10	ERRCHK (Check FDC Results)
11	Get Selected Drive

Nun zu weiteren Details der Parametertabellen: Grundsätzlich wird als Spüranzahl die tatsächliche (physikalische) Spüranzahl angegeben. Zugriff auf Seite 1 erfolgt über große Sektornummern (Sektorüberlauf) oder über (zu) große Spürnummern (Spürüberlauf). Daher benötigt das BIOS keine Information über die aktuelle Zugriffsart. Die in der Skewtable abgelegten Werte enthalten in Bit 7 (MSB) die Information, auf welcher Seite der Diskette der entsprechende physikalische Sektor zu suchen ist (dadurch wird die maximale Sektornummer auf 127 begrenzt). Dies hat den Vorteil, daß Skewfaktoren realisiert werden können, die sich über beide Seiten einer Spur erstrecken. Auch Formate, die auf Seite 1 nicht wieder bei 1 beginnend nummerieren, sind möglich (z.B. Kaypro). Nachteilig ist natürlich, daß Sektornummern >127 nicht zu bearbeiten sind (werden als entsprechend kleinere Nummern auf Seite 1 gesucht).

Leider gibt es einige Systeme, welche die physikalischen Spuren in zwei logische Spuren unterteilen (für die beiden Seiten). Dies ist im BIOS sehr leicht zu handhaben (solange keine Formatflexibilität gefordert ist) und erlaubt auch Sektornummern bis zu 255, da die Seiteninformation in der Spürnummer übergeben wird. Diese Formate sind relativ harmlos und lassen sich auch mit meinem bisherigen System bearbeiten, solange die Anzahl der 'Offsetspuren' gerade ist und die Sektornummern unter 128 bleiben. Ansonsten gibt's halbe physikalische Offsetspuren (kann CP/M nicht) oder das oben beschriebene Problem. Bisher ist mir noch nicht eingefallen, wie man alle drei Exoten (nicht durchgängige Sektornummern, halbe Spuren, zu große Sektornummern) unter einen Hut kriegt, ohne gleich die Konvertierungsroutinen auf Diskette mitzuspeichern. Zu bedenken ist weiter, daß es nicht nur um die Konvertierung innerhalb des BIOS geht, sondern auch um eine einheitliche Erscheinung zur Verwaltung mit dem Format-Manager. In diesem Sinne ist auch die Einbindung von HD-Formaten zu sehen (evtl. durch ein anderes Density-Byte), die wesentlich wichtiger als die Unterstützung von 'Chaos-Formaten' ist. Ich bitte Euch daher, Euch hierzu ein paar Gedanken zu machen und mir mitzuteilen.

Der Format-Manager ist ein menügesteuertes Turbo-Pascal-Programm, über das alle formatspezifischen Funktionen ablaufen. Außer den Funktionen zum Bearbeiten der bereits definierten Formate kann innerhalb dieses Programms eine Diskette untersucht, formatiert oder mit Parameterblock beschrieben werden. Auch das Setzen der Parameter im BIOS-RAM ist hiermit möglich (zur Bearbeitung von Fremdformaten). Als Anlage lege ich eine Liste der von mir bereits eingegebenen Formate bei. Daran könnt Ihr sehen, welche Parameter zur vollständigen Beschreibung eines Formats benötigt werden. Diese Daten werden in einer Datei gespeichert. Der DPB wird erst bei Bedarf errechnet. Die Skewtable wird zwar mit in der Datei gespeichert, aber normalerweise aus dem Skewfaktor berechnet. Nur bei Sektortabellen, die nicht den normalen Regeln folgen, wird die Sektorabfolge von Hand eingegeben (in der Tabelle erscheint dann ein logischer Skew von 0).

Das nächste, was mir zum Thema Disk-I/O einfällt, ist die automatische Diskwechselerkennung. Uwe hat schon ganz interessiert gefragt, wie ich das denn mache, ohne Schalter am Laufwerk. Ganz einfach: Über das Write-Protect-Signal. In meinem derzeitigen System wird alle 200 ms der Status der Laufwerke geprüft. Ändert sich das WP-Signal, wird die Diskette 'ausgeloggt'. Vorteil der Methode: nicht jedes Laufwerk hat einen Schalter für DCHG, aber jedes Laufwerk hat eine WP-Lichtschanke. Die 200 ms Abtastrate sind übrigens experimentell ermittelt. Es treten hierbei noch keine unerkannten Diskwechsel auf (beim Toaster-Effekt), und die für diese Funktion verbratene Rechenleistung beträgt etwa 0.25 %, was meiner Ansicht nach die Sache wohl wert ist. Wenn nur die Laufwerke untersucht werden, die bisher

eingelogg't waren, wird dieser Wert nochmal erheblich kleiner (bisher sind bei mir 2 Laufwerke angeschlossen, aber es werden 4 geprüft).

Wer schon einmal versucht hat, unter CP/M-Plus die Media Flags zu benutzen, weiß, daß das DOS an dieser Stelle einen (den einzigen mir bekannten) Fehler hat: Es schert sich überhaupt nicht um die Flags, sondern macht munter weiter. Deshalb habe ich bisher die Login-Vektoren des DOS (2x 16 Bit) entsprechend manipuliert (direkte Nachbildung der entsprechenden DOS-Funktion). Verwendung des BDOS ist innerhalb der Interruptroutine leider nicht möglich, da das DOS nicht reentrant ist. Wie ich kürzlich (indirekt) von Wolfgang Mues erfahren habe, funktioniert aber alles, wenn die Sektorpuffer ebenfalls gelöscht (bzw. verändert) werden. Das wäre natürlich dann die saubere Methode (ohne absolute DOS-Zugriffe). Falls jemand das BDOS auf 2280 umschreibt, könnten dabei vielleicht die Login-Vektoren global deklariert werden (SCB?), so daß der direkte Weg wieder frei ist (ist einfacher und schneller).

Ich möchte noch kurz auf eine weitere CP/M-Erweiterung aufmerksam machen, die in meinem bisherigen System realisiert ist. Wie inzwischen bekannt, verwaltet CP/M-3 die Schnittstellen direkt vorbildlich. Auch Ändern von Einstellungen ist über DEVICE möglich. Allerdings fehlt die Möglichkeit, das Hardware-Handshake abzuschalten oder das Protokoll (Wortbreite, Parity) zu ändern. Deshalb habe ich ab dem Beginn der Device-Tabelle rückwärts eine weitere Tabelle angelegt, welche in jeweils einem Byte pro Device diese physikalischen Parameter enthält. Das Programm DEV, welches diese Tabelle manipuliert, existiert bereits (wurde vor Jahren von einem CP/M-Kollegen geschrieben).

Aufbau der Tabelle für zusätzliche Kommunikationsparameter:

Bit 7,6 :	00 = 5 Bit	01 = 7 Bit
	10 = 6 Bit	11 = 8 Bit
Bit 5 :	0 = RX/TX Enable	1 = Auto Enables (Hard-Handshake)
Bit 3,2 :	00 = verboten	01 = 1 Stopbit
	10 = 1.5 Stopbits	11 = 2 Stopbits
Bit 1 :	0 = Parity odd	1 = Parity even
Bit 0 :	0 = Parity disable	1 = Parity enable

Nochmal zur Erinnerung: Bitte meldet Euch wegen

- Scheck & Umfang des Teilesatzes
- Verbesserungen am Parameterblock, Spur/Sektorkonvertierung
- sonstigen Ideen, was noch sinnvoll wäre

... und natürlich bei allen Fragen, die auftauchen!

Herzliche Grüße, Euer Tilmann (201289)

long live CP/M-Plus!

Stand 16.12.89 T.Reh

Stückliste:

IC1	Z280
IC2,IC3	74 HC(T) 373
IC4	74 HC(T) 74
IC5	TL 7705
IC6,IC8	74 ALS 244
IC7	74 ALS 245
IC9,IC10	27 C 256, 27 C 512
IC11..IC18	TC 514256 Z
IC19,IC20	74 HC(T) 158
IC21..IC24	GAL 16 V 8
IC25	74 HC(T) 259
IC26	74 HC(T) 244
IC27	DS 1287
IC28	FDC 37 C 65 A/B
IC29	1488, 75188
IC30	1489, 75189
IC31	COM 81 C 17
IC32	74 HC(T) 14
IC33	74 HC(T) 175
T1	BC 547 B
D1..D4	Zeilen-LED (z.B. LD 464, LD 474)
Q1	20 MHz HC-18U (vorläufig; 18.432 MHz?)
Q2	16 MHz HC-18U
Q3	9.6 MHz HC-18U
Q4	5.0688 MHz HC-18U
C1,C2	22p ker. RM 5
C3	10µ Ta. RM 2.5
C4	100n ker. RM 5
C5	(WAIT-Delay)
C6	(CAS-Burstpulse-Delay)
C7	(FDC-DRQ-Delay)
C8	15p ker. RM 5
C9	47p ker. RM 5
C10	56p ker. RM 5
C11	68p ker. RM 5
CK1..CK26	22x 100n ker. RM 5, 4x 10µ 16V Ta. RM 2.5
R1,R2	4 k 7
R3,R9,R10,R11	1 k
R4,R5,R6,R7	3 k 3
R8	100 R
RN1	RSIL 5x 2 k 2
RN2	RSIL 4x 2 k 2
RN3	RSIL 5x 330 R
RN4	RSIL 7x 10 k
RN5	RSIL 4x 330 R
RN6	RSIL 3x 3 k 3
CN1	ECB
CN2	FDD
CN3	V24
J1..J4	
J5	
J6	
J7..J11	
	VG-Leiste 64-pol. ac-bestückt (DIN 41612 C)
	Pfosten 50-pol. (2x25)
	Bestückung wahlweise auf Löt- oder Bauteilseite
	Pfosten 20-pol. (2x10)
	Pfosten 2x4, CPU-Init
	Pfosten 1x3, EPROM-Typ
	Pfosten 1x2, FDC Precompensation
	Pfosten 2x5, User-Jumper

Steckerbelegungen:CN1 (ECB-BUS):

(Signale MRQ und BUSAK fest auf +5V gelegt)

a	Nr	c
+5V	1	+5V
D5	2	D0
D6	3	D7
D3	4	D2
D4	5	A0
A2	6	A3
A4	7	A1
A5	8	
A6	9	A7
/WAIT	10	
	11	
	12	
	13	
	14	D1
	15	-12V
2xCLK	16	
	17	
	18	
+12V	19	
/M1	20	/NMI
	21	/INT
	22	/WR
	23	
	24	/RD
	25	
	26	/RESOUT
/IORQ	27	
	28	
	29	CLK
	30	(/MRQ)
(/BUSAK)	31	/RESIN
GND	32	GND

CN3 (V24-Schnittstellen):

Belegung ermöglicht direktes Aufquetschen zweier DSUB-9, deren Pin-Nummern mit der DSUB-25 Belegung übereinstimmen.

1	1	GND	Masse (Schutzmasse 1)
2	6	nc	
3	2	TXD 1	Sendedaten 1
4	7	GND	Masse (Signalmasse 1)
5	3	RXD 1	Empfangene Daten 1
6	8	nc	
7	4	RTS 1	Request-to-Send 1
8	9	nc	
9	5	CTS 1	Clear-to-Send 1
10	-	nc	
11	1	GND	Masse (Schutzmasse 2)
12	6	nc	
13	2	TXD 2	Sendedaten 2
14	7	GND	Masse (Signalmasse 2)
15	3	RXD 2	Empfangene Daten 2
16	8	nc	
17	4	RTS 2	Request-to-Send 2
18	9	nc	
19	5	CTS 2	Clear-to-Send 2
20	-	nc	

CN2 (Floppy-Laufwerke) :

1..49 ungerade: GND

Maxi	Mini	Signal	Umbelegung für 8"
2		RWC/RPM	
4		MOTOFF	
6		MOTOFF	
8		MOTOFF	
10		nc	
12		DCHG	
14		SIDE SEL	
16		HOLD	
18	2	RWC/RPM	HOLD (16,20)
20	4	HOLD	INDEX (24)
22	6	DS3	nc
24	8	INDEX	MOTOFF (4,6,8)
26	10	DS0	
28	12	DS1	
30	14	DS2	
32	16	MOTON	DS3 (22)
34	18	DIRECTION	
36	20	STEP	
38	22	WRITE DATA	
40	24	WRGATE	
42	26	TRKO	
44	28	WRPROT	
46	30	READ DATA	
48	32	SIDE SEL	(nc)
50	34	DCHG	(nc)

PAL-Codierungen

IC21, GAL 16V8, RAM-Timing und Adressgenerator

IN: NCLK,OE,A1,A2,A3,A4,AS,IE,DS,MDQ,MUX
OUT: NAS,MA0,MA1,FFR

NAS = /AS
MA0 := /AS * A3 + AS * MUX * A1 + /MUX * /MA0
MA1 := /AS * A4 + AS * MUX * A2 + /MUX * MA0 + /MUX * MA1
FFR = IE * DS * MDQ

IC22, GAL 16V8, CAS-Timing und Decodierung

IN: RW,BW,RFSH,A0,A19,A20,MUX,CLK,NCK
OUT: CASOL,CASOH,CAS1L,CAS1H

/CASOL = /MUX * /RFSH * /A20 * /A19 * BW * /A0 * CLK
+ /MUX * /RFSH * /A20 * /A19 * BW * /A0 * NCK
+ /MUX * /RFSH * /A20 * /A19 * BW * /A0 * /RW
+ /MUX * /RFSH * /A20 * /A19 * /BW * CLK
+ /MUX * /RFSH * /A20 * /A19 * /BW * NCK
+ /MUX * /RFSH * /A20 * /A19 * /BW * /RW
+ MUX * RFSH * /CLK
+ MUX * /CASOL
/CASOH = /MUX * /RFSH * /A20 * /A19 * BW * A0 * CLK
+ /MUX * /RFSH * /A20 * /A19 * BW * A0 * NCK
+ /MUX * /RFSH * /A20 * /A19 * BW * A0 * /RW
+ /MUX * /RFSH * /A20 * /A19 * /BW * CLK
+ /MUX * /RFSH * /A20 * /A19 * /BW * NCK
+ /MUX * /RFSH * /A20 * /A19 * /BW * /RW
+ MUX * RFSH * /CLK
+ MUX * /CASOH

CAS1 identisch bis auf A19,A20

IC23, GAL 16V8, System-Signale

IN: CLK,ST0,ST1,ST2,ST3,A21,A22,A23,AS,MOA
OUT: BIO,XIO,ROM,RFSH,M1,MDA,NCLK

/BIO = /ST3 * /ST2 * ST1 * /ST0 * /A23 * /A22 * /A21
+ /ST3 * ST2 * /ST1 * /ST0
/XIO = /ST3 * /ST2 * ST1 * /ST0 * /A23 * A22
/ROM = ST3 * /A23
RFSH = /ST3 * /ST2 * /ST1 * ST0
M1 = /ST3 * ST2 * /ST1 * /ST0 + "RET1-STATUS"
MDA = ST3 * /AS * A23 * /A22 * /A21
+ /ST3 * /ST2 * /ST1 * ST0 * /AS + MOA
NCLK = /CLK

IC24, GAL 16V8, I/O-Decoder

IN: XIO,A5,A6,A7,IE,DS
OUT: RTC,FDC,LDOR,LDRSR,UART,GPI,GPO,WR

/RTC = /XIO * /A7 * /A6
/FDC = /XIO * /A7 * A6 * /A5
/LDOR = /XIO * /A7 * A6 * A5
/LDRSP = /XIO * A7 * /A6 * /A5
/UART = /XIO * A7 * /A6 * A5
/GPI = /XIO * A7 * A6 * /IE * /DS
/GPO = /XIO * A7 * A6 * IE * /DS
/WR = IE * /DS

Speicher/I/O-Organisation

Speicher: 000000-01FFFF Boot/System-EPROM
800000-8FFFFF 1 MB RAM Onboard

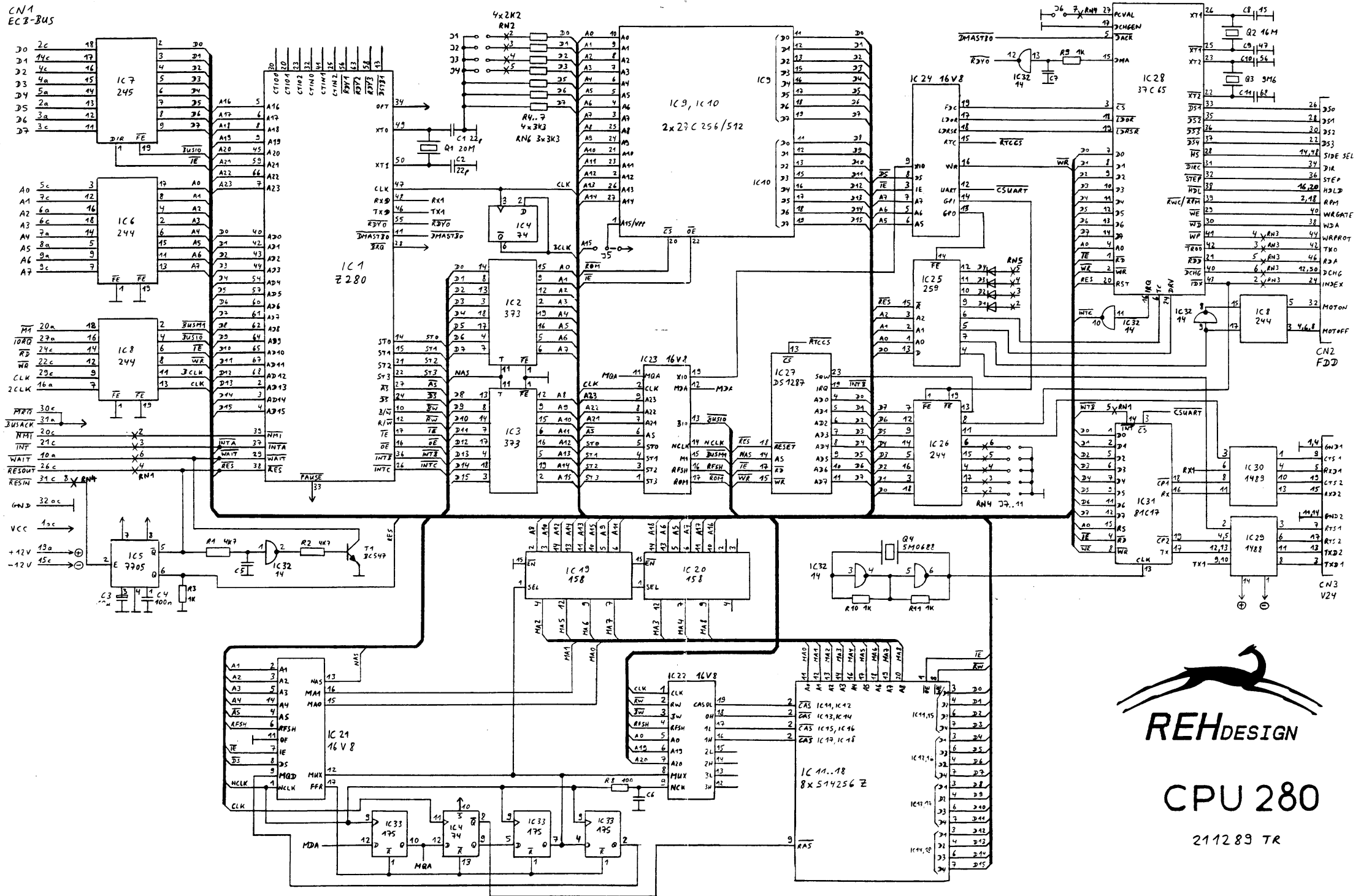
I/O: 000000-0000FF ECB-Bus
400000-4000FF Onboard-I/O
FE0000-FFFFF Onchip-I/O

Onboard: 00-3F RTC / NVRAM
40..41 FDC
60 FDC-LDOR
80 FDC-LDRSR
A0,A1 TPUART
C0 GPIO

Interrupt-Quellen

NMI: ECB-Bus
INTA: ECB-Bus
INTB: RTC, TPUART
INTC: FDC

CN1
ECB-BUS

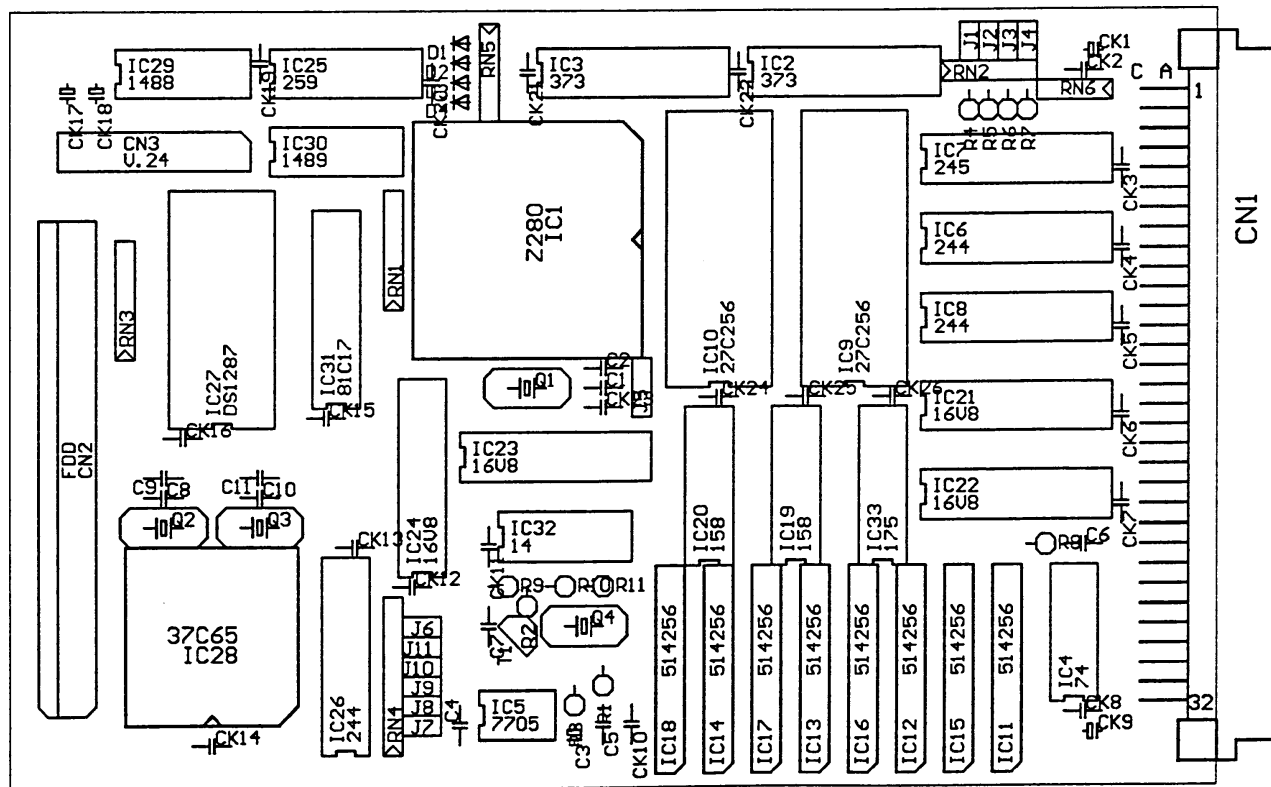


CPU 280

211289 TR

Gesamtliste Diskettenformate vom 3.1.90 19:55:26

Nr.	Name	Größe	Trk	tpi	Sei	Mod	Dense	SecGr	Secs	Blk	Dir	Off	Skw	Auf	PSk
1	Reh CP/M 3 V2.1	5"	80		2	Sek	MFM	1024	5	2	256	0	2	N	1
2	Reh/GHS CP/M 3	5"	80		2	Sek	MFM	1024	5	4	128	1	2	N	1
3	IBM Standard	8"	77		1		FM	128	26	1	64	2	6	N	1
4	Philips P2000 80T	5"	80		2	Sek	MFM	256	16	2	128	0	1	N	1
5	Osborne 1 DD	5"	40	48	1		MFM	1024	5	1	64	3	1	N	1
6	Osborne 2 DD	5"	80		1		MFM	1024	5	2	128	3	1	N	1
7	Schaefer CP/M 2.2	5"	80		1		MFM	256	16	2	128	4	1	N	1
8	Reh/GHS CP/M 2.2 DS	5"	80		2	Sek	MFM	256	16	4	128	4	1	N	1
9	DEC VT180	5"	40	48	1		MFM	512	9	1	64	2	2	N	1
10	DEC Rainbow 100	5"	80		1		MFM	512	10	2	128	2	2	N	1
11	Wavemate Bullet	5"	80		2	Sek	MFM	1024	5	2	128	1	1	N	1
12	Alphatronic P2	5"	40	48	1		MFM	256	16	1	64	2	1	N	1
13	Alphatronic PC	5"	40	48	2	Sek	MFM	256	16	2	128	2	1	N	1
14	EPSON QX10	5"	40	48	2	Sek	MFM	256	16	2	64	4	1	N	1
15	mc 80 track	5"	80		1		FM/MFM	256	16	2	64	4	1	N	1
16	Siemens PC16-10	5"	40	48	2	Sek	MFM	512	9	2	64	2	1	N	1
17	Siemens PC16-11	5"	80		2	Sek	MFM	512	9	2	256	2	1	N	1
18	ZDOS DS 80T	5"	80		2	Trk	MFM	512	10	2	128	2	2	I	1
19	Schneider Vortex	5"	80		2	Sek	MFM	512	9	4	128	1	1	N	5
20	IBM-PC DS 9 Spt	5"	40	48	2	Sek	MFM	512	9	1	32	0	1	N	1
21	ITT 3030 CP/M	5"	70		2	Trk	MFM	256	16	2	128	4	1	N	1
22	Philips P3500	5"	80		2	Sek	MFM	256	16	2	256	0	1	N	1
23	KWS CP/M 68k	5"	80		2	Sek	MFM	256	16	4	128	2	1	N	1
24	Oettle-Reichler	5"	80		2	Sek	MFM	1024	5	4	128	2	1	N	1
25	MS-DOS 800k ???	5"	80		2	Sek	MFM	1024	5	2	128	0	1	N	1
26	IBM-PC SS 8 Spt	5"	40	48	1		MFM	512	8	1	32	0	1	N	1
27	IBM-PC DS 8 Spt	5"	40	48	2	Sek	MFM	512	8	2	128	0	1	N	1
28	mc-CP/M FLO2	5"	80		2	Sek	MFM	1024	5	2	256	2	1	N	1
29	c't PROF #4	5"	80		2	Sek	MFM	512	10	2	128	3	1	N	1
30	IBM CP/M-86	5"	40	48	1		MFM	512	8	1	64	1	1	N	1
31	Kaypro IV C128	5"	40	48	2	Sek	MFM	512	10	2	128	0	0	N	5
32	prof #4 Skew 2	5"	80		2	Sek	MFM	512	10	2	128	3	2	N	1
33	Reh CP/M 3 V2.0	5"	80		2	Sek	MFM	1024	5	2	256	0	3	N	1
34	Marco-Spezial	5"	80		2	Sek	MFM	512	9	4	128	1	1	N	1
35	P2000C	5"	80		2	Sek	MFM	256	16	4	128	1	2	N	1



Drawing: Z280.BRD

Last modified: 21.12.1989 14:30:32

Plot date: 04.01.1990 09:56:46

Hallo, CP/M-Freunde!

Nachdem Ihr lange nichts mehr von mir gehört haben, möchte ich Euch nun über den Stand der CPU280-Dinge aufklären.

Zunächst: Die Hardware läuft. Nach anfänglichen Timing-Problemen (die CPU hat kürzere Signallaufzeiten als im Datenbuch angegeben) sieht nun die GAL-Programmierung etwas anders aus. Bei derzeit 9.216 MHz Bustakt klappt alles ganz prima, der 16-Bit-Bus mit Cache und Burst-Mode arbeiten wie geplant.

Aber jetzt: Von der Leistung der Maschine bin ich unheimlich enttäuscht. Ich habe ausführliche (Assembler-) Benchmarks durchgeführt, mit allen denkbaren Kombinationen von Wait-States, Busfaktor, Cache und Burst. Ergebnis: Bei vollem Bustakt (1:1) ist der Z280 mit Cache und Burst fast so schnell wie ein 64180 bei gleicher Taktfrequenz. Werden viele Sprünge, Unterprogrammaufrufe und dergleichen ausgeführt, ist er erheblich langsamer. In äquivalenter Z80-Frequenz: Größenordnung 7-12 MHz. Diese Tests verwendeten alle nur Z80-Befehle, da in der Praxis wohl kaum Z280-Programme verfügbar sind. Damit man sie als PD weitergeben kann, darf man noch nicht einmal Neuentwicklungen in Z280 schreiben. Die normalerweise verwendeten Programme sind sogar oft nur in 8080-Code geschrieben.

Auf meine Bitte hin hat Wilfried Schmitts Hochsprachen-Benchmarks auf seiner Z280-EURO laufen lassen (mein BIOS ist noch nicht). Hier die Ergebnisse im Vergleich zu meiner CPU180 (Alle Werte haben eine Genauigkeit von +/- 0,1):

Test	Z280EURO (10MHz)	Turbo gepatcht	CPU180 (9.2 MHz)
Intmath :	2,0	0,7	2,2
Realmath :	27,6	21,2	21,4
Triglog :	45,0	21,3	37,5
Store RamDisk :	1,0	1,0	0,8
Store Disk :	5,8	5,9	11,3

Die mittlere Spalte stammt von einem Benchmark, der mit einem für den Z280 gepatchten Turbo-Compiler erzeugt wurde. Ähnliches ist natürlich auch für den 180 möglich, allerdings habe ich dies bisher nicht gemacht, deswegen habe ich keine Vergleiche dazu.

Die verschiedenen Zeiten beim Disk-I/O rühren daher, daß auf der Z280EURO mit Multi-Sector-I/O ohne Skew gearbeitet wird, während auf der CPU180 jeder Sektor einzeln geschrieben (und verifiziert!) wird, bei einem Skew von 2. Das BIOS der CPU180 ist außerdem bisher in keiner Beziehung zeitoptimiert.

Aufgrund dieser Erfahrungen bin ich nun etwas demotiviert, was die weitere Arbeit an der CPU280 angeht. Streng genommen wäre der Einsatz dieser Karte gegenüber meinem derzeitigen System ein Rückschritt. Jedenfalls scheint mir der doch immense Schaltungsaufwand für 16-Bit-Bus und Burst-Mode nicht durch entsprechende Leistung gerechtfertigt zu werden. Ich möchte deshalb die Frage an Euch richten, ob man nicht besser eine überarbeitete CPU180-Neuaufgabe machen sollte. Die Ausstattung wäre genau die gleiche, mit nur einer Einschränkung (nur noch 184 statt 256 I/O-Adressen auf dem ECB-Bus, da 72 Adressen durch CPU-internes I/O und Onboard-I/O belegt werden). Dafür ist die CPU180 schaltungstechnisch wesentlich einfacher und damit auch preiswerter, vom niedrigeren Preis der CPU einmal abgesehen (31,- gegen 75,- DM). Zur 'Speed' wäre noch zu sagen, daß es den Z180 ebenfalls als 10-MHz-Version gibt, und eine 12.5-MHz-Version ist nach Herstellerangaben nicht auszuschließen. Ich persönlich halte dies jedenfalls für wahrscheinlicher als die 25-MHz-Version vom Z280.

Daher nun die Frage, wie es weitergehen soll. Ich selbst kann mich nicht so recht entscheiden (die neuen Adressierungsarten locken mich doch etwas), vor allem wegen der Schwierigkeiten beim PD-Austausch und der von der Art des Programms abhängigen (insgesamt zu niedrigen) Rechenleistung. Ich möchte Euch daher bitten, mir baldmöglichst Eure Meinung mitzuteilen, damit ich weiß, an welcher der beiden Karten ich weiterarbeiten soll.

Herzliche Grüße, Euer Tilmann (030390)

Hallo!

Nachdem ich das Rundschreiben bereits fertig hatte, erfuhr ich von Andreas Burmester, daß mit der Z280EURO von Wilfried wohl irgendetwas nicht stimmen könne. Jedenfalls seien die angegebenen Benchmark-Werte falsch. Er gab mir daraufhin die Werte seines Rechners, allerdings fehlt mir dazu jede Vergleichsmöglichkeit, da er nur die Z280-optimierten Werte angegeben hat.

Es scheint, daß der Z280 wohl doch eine etwas höhere Leistung hat als zunächst befürchtet. Richtig stark wird er allerdings erst, wenn man konsequent in Z280 programmiert. Dazu gehört z.B. auch, daß 16-Bit-Worte auf geraden Adressen liegen sollten. Leider existieren bisher keine derartigen Programme, und innerhalb der PD-Welt wird sich derartiges auch kaum durchsetzen.

Deswegen bleibt meine Frage bestehen: Welches der beiden CPU-Systeme sollen wir favorisieren? Der Leistungsunterschied ist nach Andreas' Aussagen wohl doch nicht vorhanden, es bleibt natürlich der erheblich größere Schaltungsaufwand. Blicke noch zu bemerken, daß ich für meine CPU180 bereits ein fertiges BIOS habe, welches man nun konsequent weiterentwickeln könnte (z.B. I/O als parallele Prozesse über Interrupts, Disk-Cache usw.). Bei der CPU280 fange ich BIOS-mäßig exakt bei Null an. Bitte meldet Euch bald, Eure Meinung ist wichtig!

nochmals Grüße, Tilmann (060390)

Hallo.

Ihr seid sicher gespannt, wie es denn nun weitergeht. Ich habe von vielen Antwort bekommen, daß die CPU180 durchaus eine attraktive Alternative sei, manchen war sie sogar lieber. Allerdings hatte ich ja schon immer einen Hang zu den schönen neuen Z280-Befehlen und Adressierungsarten... Ich war also richtig in der Klemme.

Dann kam eine Meldung von Zilog: die Z80-CPU ist als 20-MHz-CMOS-Version ab sofort erhältlich. Dies brachte mich auf den Gedanken: wenn die so etwas bauen können, gibt's vielleicht demnächst auch den (seit 1983 angekündigten) 25-MHz-Z280 (damals noch Z800). Oder auch einen schnelleren Z180. Rücksprache bei Zilog erbrachte folgendes: Z180 derzeit 10 MHz, evtl. später 12.5 MHz; Z280 derzeit mit 12.5 MHz in Bemusterung, schneller wohl nie.

Meine Überlegungen daraufhin: Die Performanz des 12.5-MHz-Z280 dürfte etwa dem 10-MHz-Z180 vergleichbar sein (die etwas aufwendigeren internen Zugriffe und die Pipeline-Verluste müßten durch die höhere Taktfrequenz ausgeglichen werden). Da aber dessen Software (vom BIOS abgesehen) interessanter ist, wollte ich schon gerne den Z280 verwenden, falls nicht technische Probleme dies verhindern. Als technisches Problem lag kongret die Ansteuerung des FDC vor: das Ding braucht ein Terminal-Count-Signal am Ende einer DMA-Übertragung, und dieses Signal kann von der Z280-internen DMA nicht erzeugt werden. Der Betrieb des FDC über die DMA ist aber zwingende Voraussetzung für das später angestrebte 'Parallel-BIOS' (Übrigens verwenden B&R auf ihrer Karte keine DMA für Disk-I/O, sondern die CPU macht den ganzen Transfer).

Als sich nach mehreren Rücksprachen mit den beteiligten Halbleiterherstellern eine saubere Softwarelösung abzeichnete, habe ich die BIOS-Programmierung der CPU280 in Angriff genommen. Da das Disk-I/O völlig CPU-unabhängig über die DMA läuft, ist somit die Entscheidung zugunsten der CPU280 gefallen.

Heute nun ist zum ersten Mal ein voll funktionsfähiges CP/M-3 auf der CPU280 gelaufen. Noch ohne I/O-Redirection und sonstige Gimmicks, aber gut genug, um die HL-Benchmarks einmal selber durchzuführen. Hier die Ergebnisse (alle Zeiten in Sekunden):

Prozessor	Takt	INTMATH	REALMATH	TRIGLOG
Z80A	4.000 MHz	6.3	54.5	95.5
64180	9.216 MHz	2.1	21.4	37.6
Z280	9.216 MHz	1.6	22.3	37.0
Z280 EVEN	9.216 MHz	1.5	19.2	32.3
Z280	12.288 MHz	1.3	16.8	27.9
Z280 EVEN	12.288 MHz	1.2	14.5	24.3

Die Tests mit 'EVEN' bezeichnen eine Programmversion, bei der alle Wortzugriffe auf gerade Adressen erfolgen. Die Werte des Z280 bei 12.288 MHz sind hochgerechnet, da ich derzeit nur die 10-MHz-Version habe. Letztlich wird die CPU280 aber mit 12.288 MHz laufen (die hochgerechneten Werte sind Maximalwerte und können bis zu 0.2 Sekunden zu hoch sein).

Wie Ihr Euch denken könnt, war ich direkt angenehm überrascht. Meine Assembler-Benchmarks waren wohl doch nicht repräsentativ genug. Also gibt es jetzt kein Halten mehr: die CPU280 wird gebaut!

Am Schaltungsdesign hat sich inzwischen hier und da etwas geändert (insbesondere GAL-Programmierung), aber vom Gesamtumfang und der Funktionsweise stimmt noch alles mit dem Erst-Entwurf überein. Deswegen schicke ich Euch die aktuellen Unterlagen erst mit Platinen und Bauteilen zu.

Wie es jetzt weitergeht? Ich werde jetzt parallel am neuen Layout (ein paar Bauteile müssen aus mechanischen Gründen etwas verschoben werden, GAL's teilweise völlig anders belegt) sowie am BIOS arbeiten. Nach meiner vorsichtigen Schätzung wird das den Mai kosten. Dann wird es ernst: die Platinen müssen hergestellt werden, was wohl dann bis Ende Juni dauern wird. Bis dahin dürfte auch mein BIOS soweit sein wie mein derzeitiges 180er, so daß Platinen, Bauteile und Betriebssystem auf Reise gehen können. Mit viel Glück habe ich sogar bis dahin eine Dokumentation fertig, die Euch bei Aufbau und Inbetriebnahme etwas unterstützen könnte.

Das BIOS wird alle On-Board-Funktionen voll unterstützen, darüberhinaus die c't-Solid-State-Floppy und die c't-RAM-Disk (auch wenn letztere nichts taugt), wobei diese automatisch erkannt und installiert werden. Weitere Gerätschaft (z.B. HDC oder weitere Schnittstellen) wird jeder selbst installieren müssen. Ach ja: als externe Schnittstellen sind in meinem BIOS noch eine Spooler-Karte und eine I/O-Karte mit SIO/PIO/CTC (beide selbstgestrickt) eingebunden.

Möglicher Speicherausbau: derzeit 512k und 1M. Mit den neuen 4-MB-RAMs sind auch die Ausbaustufen 2 MB und 4 MB möglich. Beim Wechsel von 1-MB-RAMs auf 4-MB-RAMs ist lediglich ein GAL umzuprogrammieren.

Ich möchte diejenigen, die noch keinen Scheck geschickt haben, noch einmal daran erinnern, daß die Anzahl bestellter Platinen und Teilesätze exakt identisch der Anzahl vorhandener Schecks sein wird. Wer später noch aufspringen will, verursacht großen Aufwand und damit erhöhte Kosten (später werde ich auch keine Teilesätze mehr liefern).

Grüße, Tilmann (240490)

- Der Z280 ist rehabilitiert! -

Liebe CP/M-Freunde,

endlich ist es soweit! Wir werden uns treffen, um uns und den neuen Rechner kennenzulernen. Doch dazu später mehr, vorher noch ein paar technische Informationen:

Die Platinen für die CPU280 sind derzeit in Herstellung (bereits die neue, überarbeitete Version) und ich rechne bald mit der Lieferung. Unterdessen programmiere ich am BIOS weiter, derzeit bin ich beim AutoFormat-System mit einem neuen Parameterblock und noch mehr Möglichkeiten als bisher. Inzwischen habe ich auch ein Musterexemplar der neuen CPU (12.5 MHz) erhalten, so daß mein CPU280-Prototyp jetzt mit 12.288 MHz läuft (einwandfrei übrigens). In diesem Zusammenhang auch noch einmal die aktuellen HL-Benchmarks:

Prozessor	Taktfrequenz	Integer	Real	TrigLog
Z80A	4 MHz	6.3	54.5	95.5
Z180	9.216 MHz	2.1	21.4	37.6
Z280	12.288 MHz	1.2	15.4	25.7 (alle Daten ODD)
Z280	12.288 MHz	1.1	13.8	23.2 (alle Daten EVEN)

Das Hardware-Handbuch ist auch schon fertig (technische Unterlagen und Schaltungsbeschreibung), so daß ich höchstwahrscheinlich Platine und Handbuch bei unserem Treffen 'ausliefern' kann. Bezüglich Bauteilbestellung bestehen noch zwei Probleme: Erstens kann Zilog den schnellen Prozessor noch nicht in Stückzahlen liefern (Serienanlauf 3Q90), zweitens machen inzwischen so viele Leute mit, daß ich beim Einkauf der Bauteile einen erheblichen Vorschub zahlen muß (die Schecks decken eben nur einen Teil der Materialkosten). Ich denke, auf die Prozessoren sollten wir warten (deswegen habe ich auch sonst noch keine Bauteile gekauft), und die Schecks werde ich nicht erst bei Auslieferung, sondern bereits beim Einkauf der Teile einlösen. Außerdem sollten wir den Vorauszahlungsanteil noch etwas erhöhen. Ihr könnt Euch selbst ausrechnen, wieviel das Material für gut 20 Rechner kostet...

Das Software-Handbuch, welches meine CP/M-3-Implementation beschreibt und auch detaillierte Angaben zu den verschiedenen Parametern und Mechanismen enthält, muß noch geschrieben werden. Damit werde ich aber erst nach weitgehender Fertigstellung des BIOS beginnen. Früh genug für Eure Inbetriebnahme unter CP/M, aber nicht früh genug für unser Treffen.

Nun aber zum Wesentlichen: Unser Treffen findet am **Samstag, dem 21.7.** bei mir in Siegen statt. Es beginnt mittags um **12 Uhr** und wird vermutlich bis in den frühen Abend dauern (Ihr sollt Gelegenheit haben, noch am Samstag nach Hause zu kommen). Um die Mittagsverpflegung möchte ich mich nicht kümmern, Ihr solltet also etwas gegessen haben, wenn Ihr hier ankommt. Kaffee und Kuchen nachmittags werde ich stellen.

Ein sehr wesentlicher Aspekt unseres Treffens ist das persönliche Kennenlernen. Außerdem wollt Ihr natürlich alle die CPU280 in Aktion sehen. Aus Zeitgründen können wir aber nicht auf jedes Detail stundenlang eingehen, deswegen werde ich ein paar der besonderen Funktionen (Bootlader, AutoFormat, Diskwechsel, FormatManager) vorführen und ein bißchen über das zukünftig Geplante berichten. Ihr solltet Euch ein bißchen darauf vorbereiten und auch Eure Fragen und Wünsche schon vorher überlegen, so daß die kurze Zeit hier auch gut genutzt wird. Übrigens wird Günther Schock sein LCD-Terminal mitbringen, das können wir uns dann (vielleicht an der CPU280?) auch einmal ansehen.

Ich bitte **dringend** darum, daß jeder von Euch, der an diesem Treffen teilnehmen möchte, sich **so bald wie eben möglich verbindlich anmeldet**. Hierzu reicht ein kurzer Anruf oder eine Nachricht auf meinen elektronischen Telefonknecht (wenn ich nicht da bin).

Hier noch kurz eine Wegbeschreibung, schließlich sollt Ihr auch zu mir finden:

A45 Abfahrt Siegen/Netphen, noch auf dem Zubringer rechts Richtung Dreisbach-Siedlung. Am Stop-Schild rechts, ca. 1 km durch eine 'Zone 30', noch etwa 1 km bis diese Straße in eine andere einmündet (vorher geht's gut ausgebaut einen Berg hoch). An dieser Einmündung scharf links auf die Hauptstraße abbiegen, wieder ca. 1 km weiter zu Beginn einer großen Linkskurve geht gerade aus eine kleinere Straße ab. Dort hinein ('In der Großenbach'), die nächste Straße rechts, bei den beiden folgenden 'Gabeln' erst rechts, dann links, und dann einfach bis zum letzten Haus durchfahren. Das isses. Hier noch einmal die vollständige Adresse:

Tilman Reh, In der Großenbach 46, 5900 Siegen-Achenbach, 0271-312599

Alles weitere dann mündlich beim Treffen (Ihr dürft natürlich auch vorher anrufen, wenn noch etwas unklar ist).

Bis dahin viele Grüße, Tilman (020790)

Hallo, CP/M-Freunde!

Ich denke, jetzt habt Ihr lange genug gewartet. Zumindest auf mein Schreiben, in dem ich mich zunächst noch einmal herzlich für Euer Kommen zu unserem Treffen bedanken möchte. Ich lege unser Gruppenfoto bei, damit in Zukunft alle immer sehen können, wer am anderen Ende der Telefonleitung sitzt...

Wie Ihr alle wißt, war zu diesem Zeitpunkt noch ein Bug beim Power-Up vorhanden, und mit diesem Fehler wollte ich auf keinen Fall etwas ausliefern. Inzwischen ist der Fehler gefunden (es war der Quarz!) und behoben (Markenexemplar mit definierten Spezifikationen), so daß die CPU280 nun auch zuverlässig startet.

Seitdem habe ich noch etwas an System und Format-Manager herumgefeilt, und das Ganze ist inzwischen so weit gediehen, daß ich in dieser Woche die Herztransplantation (CPU180 raus - CPU280 rein) bei meinem Rechner durchgeführt habe. Keinerlei Komplikationen - der Patient ist wohl auf. Deswegen sollt Ihr nun auch endlich in den Genuß des Z280 kommen.

Natürlich kommt auch diesmal wieder etwas dazwischen! Vom 7.-23.9. fahre ich nämlich in Urlaub, und die Woche darauf ist das Wochenende in Mannheim (s.u.). Also werde ich erst danach anfangen können, Material zu bestellen. Ich vermute, es wird Ende Oktober, bevor die ersten 'Bausätze' an Euch rausgehen. Mit der bis dahin existierenden Software, bei der man mit einem umfangreichen späteren Update rechnen sollte (jetzt ist es zwar nicht 'Dirty', aber relativ 'Quick' programmiert, und noch ohne gewisse Features wie z.B. Interrupt-I/O und Multi-Sector). Wie ich soeben erfahren habe, wird auch der Z280-12.5 etwa dann lieferbar sein.

Den Bausatzumfang möchte ich hier noch einmal genau spezifizieren: Enthalten sein werden alle IC's (Speicher 1 MB) und alle IC-Fassungen, alle Quarze, die Widerstandsnetzwerke, die VG-Leiste und die benötigten Pfostenleisten, natürlich die Platine selbst, das Hardware-Handbuch, sowie eine Diskette (bevorzugt 5¼") mit der vollständigen Systemsoftware. **Nicht** enthalten sein werden die Kondensatoren und Widerstände, Transistor, LED's, Pfostenstecker, Jumper. Das Software-Handbuch muß erst noch geschrieben werden, und ich fürchte, daß ich das bis dahin nicht schaffe. Auf jeden Fall werdet Ihr aber genug Informationen bekommen, um das Ding ans Laufen zu bekommen und damit zu arbeiten.

Wie Ihr wißt, hat der Umfang der Bestellungen inzwischen ein Ausmaß angenommen, bei dem ich nicht mehr das Geld für den Teilekauf vorstrecken möchte. Deswegen hätte ich gerne, wie im Juli angesprochen, von den Bausatz-Bestellern einen weiteren Scheck über DM 250,-. Die Summe von damit DM 500,- reicht für den größten Teil des Materials, so daß nur noch ein kleiner Restbetrag vorzustrecken ist. Die meisten von Euch werden wohl nach Mannheim kommen, so daß sie mir dort das Ding übergeben können.

Nun zu dem Mannheim-Wochenende. Der CPC-Userclub Mannheim/Ludwigshafen organisiert dort am 29./30.9. das **1. Z80- und CP/M-Treffen Deutschlands** und lädt dazu **alle** CP/M-User und Clubs ein. Es wird Vorträge zu verschiedenen Themen geben, parallel dazu findet eine Computerbörse statt, und am Samstagabend ist Grillen angesagt. Genaue Info (und Anmeldung!) bei: Jürgens Linder, K2.23, 6800 Mannheim, Telefon 0621-103461.

Bis dahin, Tilmann (310890)

Hallo, CPU280-Besitzer!

Fast ein Jahr nach meinem ersten Rundschreiben möchte ich mich ein letztes Mal in dieser Form bei Euch melden.

Bei einigen läuft die Maschine schon, andere löten noch, wieder andere suchen Fehler... Aber ich denke, bald können sich alle am Z280 erfreuen. Wie Ihr Euch denken könnt, fängt jetzt die eigentliche Arbeit erst richtig an: echte Z280-Utilities warten auf die Programmierung, insbesondere Assembler und Compiler. Das BDOS müßte dringend mal umgeschrieben werden (seit dem Z80 überfällig!). Und an Hardware steht auch noch einiges an.

Zum Thema Hardware: Es wird schon an einem IDE-Interface gearbeitet (AT-Bus-Schnittstelle für Festplatten), wahrscheinlich ist auf der Karte auch noch etwas Platz für V.24 oder Centronics. Damit hätte man dann schonmal die wichtigste I/O-Karte.

Am Betriebssystem (Lader, BIOS und residentes DOS) mache ich derweil etwas weiter. Allerdings langsam, da ich mich auch mal um meine eigentliche Arbeit kümmern muß. Über den Stand der Dinge (und erkannte Fehler) werde ich in Zukunft wohl kaum noch in Rundschreiben berichten. Hier seid Ihr also gefordert, ab und zu mal nachzufragen, wie's aussieht. Dies gilt auch für Updates.

Nach dem Versenden des Materials habe ich noch folgende Fehler festgestellt:

1. Bei den EPROMs vom 14.11.90 mußte durch einen Patch sichergestellt werden, daß nach einem RTC-Reset tatsächlich das Default-Setup verwendet wird; die Version vom 27.11.90 ist korrigiert. Inzwischen (seit heute) kann die Defaulteinstellung durch J7 erzwungen werden (dadurch auch mehrmals möglich!).
2. Nach dem Formatieren einer Diskette holt der Format-Manager seine Daten grundsätzlich von Laufwerk A:. Deswegen sollte er auch von A: aufgerufen werden, sonst gibts Müll (es sei denn, man beendet sofort nach dem Formatieren).
3. Der Format-Manager berechnete die internen Daten von 5.25"-HD-Formaten falsch. Ab Versionsdatum 30.11. ist dieser Fehler behoben. Weiterhin wurde bei inversen Formaten die Directory falsch initialisiert (behoben ab 29.11.) und bei HD-Formaten ein fehlerhafter Parameterblock erzeugt (behoben ab 22.11.).
4. Es gibt manchmal Probleme beim Schreiben von HD-Disketten. Lesen ist dagegen problemlos möglich. Es handelt sich dabei um einen weiteren Chipfehler des Z280, der hoffentlich durch eine andere Programmierung des I/O-GALs kompensiert werden kann.
5. Um auch tatsächlich alle Formate bearbeiten zu können, habe ich sowohl in Lader als auch im System die Disk-Parameter entsprechend vergrößert (880 Blocks, 512 Directory als Grundlage für GENCPM).
6. Am Format-Manager habe ich einige kleinere Änderungen durchgeführt. Die nächste Version wird außerdem die neue Spurübersetzung beinhalten.
7. Inzwischen gibt es eine Definition für ein 5.25"-HD-Format.

Bezüglich der Chipfehler des Z280: Es gibt wohl inzwischen ein neues Errata Sheet zur aktuellen Maske. Allerdings habe ich noch keins davon, hoffe aber bald eins bekommen zu können. Die Systemprogrammierer unter Euch, die sich dafür interessieren, können mich ja demnächst mal danach fragen (am besten Mitte Januar - dieses Jahr läuft nicht mehr viel!).

Was jetzt noch fehlt (und ich beim Versenden der Päckchen vergessen habe), ist eine genaue Liste aller Teilesatz-Empfänger:

Peter Arbogast, Ahornweg
Bernd Bauer, Lüdemannstraße
Helmut Bernhardt, Hafenstraße
Fritz Chwolka, Saarstraße
Andreas Fischer, Danziger Straße
Lex de Groot, Ruyschstraat
Lucius Gruber, Hindenburgstraße
Morten Gundersen, Sagdalsveien
Jan Harbeck, Pfarramt,
Uwe Herczeg, Südstraße
Helmut Jungkunz, Zacherlstraße
Andreas Kisslinger, Haberstraße
Claus Littmann, Plockhorst,
Walter Lorenz jun., Mahräckerstraße
Wolfgang Mies, Geyssostraße
Stefan Nitschke, Germannstraße
Alexander Schmid, St. Cajetan Straße
Günther Schock, Stettiner Straße
Gerald Schröder, Am Schützenplatz
Rüdiger Sörensen, Thomas-Mann-Straße
Peter Winter, Tiergartenstraße
Axel Zinser, Podbielskistraße

Bleibt nur noch:

Viele Grüße, frohe Weihnachten und ein gutes neues Jahr!

Tilman Reh (171290)

Hallo, CPU280-Besitzer!

Fast ein Jahr nach meinem ersten Rundschreiben möchte ich mich ein letztes Mal in dieser Form bei Euch melden.

Bei einigen läuft die Maschine schon, andere löten noch, wieder andere suchen Fehler... Aber ich denke, bald können sich alle am Z280 erfreuen. Wie Ihr Euch denken könnt, fängt jetzt die eigentliche Arbeit erst richtig an: echte Z280-Utilities warten auf die Programmierung, insbesondere Assembler und Compiler. Das BDOS müßte dringend mal umgeschrieben werden (seit dem Z80 überfällig!). Und an Hardware steht auch noch einiges an.

Zum Thema Hardware: Es wird schon an einem IDE-Interface gearbeitet (AT-Bus-Schnittstelle für Festplatten), wahrscheinlich ist auf der Karte auch noch etwas Platz für V.24 oder Centronics. Damit hätte man dann schonmal die wichtigste I/O-Karte.

Am Betriebssystem (Lader, BIOS und residentes DOS) mache ich derweil etwas weiter. Allerdings langsam, da ich mich auch mal um meine eigentliche Arbeit kümmern muß. Über den Stand der Dinge (und erkannte Fehler) werde ich in Zukunft wohl kaum noch in Rundschreiben berichten. Hier seid Ihr also gefordert, ab und zu mal nachzufragen, wie's aussieht. Dies gilt auch für Updates.

Nach dem Versenden des Materials habe ich noch folgende Fehler festgestellt:

1. Bei den EPROMs vom 14.11.90 mußte durch einen Patch sichergestellt werden, daß nach einem RTC-Reset tatsächlich das Default-Setup verwendet wird; die Version vom 27.11.90 ist korrigiert. Inzwischen (seit heute) kann die Defaulteinstellung durch J7 erzwungen werden (dadurch auch mehrmals möglich!).
2. Nach dem Formatieren einer Diskette holt der Format-Manager seine Daten grundsätzlich von Laufwerk A:. Deswegen sollte er auch von A: aufgerufen werden, sonst gibts Müll (es sei denn, man beendet sofort nach dem Formatieren).
3. Der Format-Manager berechnete die internen Daten von 5.25"-HD-Formaten falsch. Ab Versionsdatum 30.11. ist dieser Fehler behoben. Weiterhin wurde bei inversen Formaten die Directory falsch initialisiert (behoben ab 29.11.) und bei HD-Formaten ein fehlerhafter Parameterblock erzeugt (behoben ab 22.11.).
4. Es gibt manchmal Probleme beim Schreiben von HD-Disketten. Lesen ist dagegen problemlos möglich. Es handelt sich dabei um einen weiteren Chipfehler des Z280, der hoffentlich durch eine andere Programmierung des I/O-GALs kompensiert werden kann.
5. Um auch tatsächlich alle Formate bearbeiten zu können, habe ich sowohl in Lader als auch im System die Disk-Parameter entsprechend vergrößert (880 Blocks, 512 Directory als Grundlage für GENCPM).
6. Am Format-Manager habe ich einige kleinere Änderungen durchgeführt. Die nächste Version wird außerdem die neue Spurübersetzung beinhalten.
7. Inzwischen gibt es eine Definition für ein 5.25"-HD-Format.

Bezüglich der Chipfehler des Z280: Es gibt wohl inzwischen ein neues Errata Sheet zur aktuellen Maske. Allerdings habe ich noch keins davon, hoffe aber bald eins bekommen zu können. Die Systemprogrammierer unter Euch, die sich dafür interessieren, können mich ja demnächst mal danach fragen (am besten Mitte Januar - dieses Jahr läuft nicht mehr viel!).

Was jetzt noch fehlt (und ich beim Versenden der Päckchen vergessen habe), ist eine genaue Liste aller Teilesatz-Empfänger:

Peter Arbogast, Ahornweg 13, 8067 Petershausen, 08137-7150
Bernd Bauer, Lüdemannstraße 49, 2300 Kiel, 0431-673819
Helmut Bernhardt, Hafenstraße 7, 2305 Heikendorf, 0431-241907
Fritz Chwotka, Saarstraße 34, 5173 Aldenhoven, 02464-8920
Andreas Fischer, Danziger Straße 36, 5902 Netphen, 02733-2286
Lex de Groot, Ruyschstraat 70, NL-1091 CE Amsterdam
Lucius Gruber, Hindenburgstraße 45, 6500 Mainz, 06131-672728
Morten Gundersen, Sagdalsveien 9c, N-2010 Strømmen
Jan Harbeck, Pfarramt, 2218 Wrist, 04822-5517

Uwe Herczeg, Südstraße 4, 7129 Botenheim, 07135-6331
Helmut Jungkumz, Zacherlstraße 14, 8045 Ismaning, 089-969374
Andreas Kisslinger, Haberstraße 10, 8000 München 50, 089-8122436

X Claus Littmann, Plockhorst, Zum Spring 15, 3155 Edemissen, 05372-7796
Walter Lorenz jun., Mahräckerstraße 9, 6000 Frankfurt/Main 50, 069-531656

X Wolfgang Mues, Geysstraße 3a, 3300 Braunschweig, 0531-330869
Stefan Nitschke, Hermannstraße 5, 7519 Walzbachtal 1, 07203-452
Alexander Schmid, St. Cajetan Straße 38/VII, 8000 München 80, 089-495326
Günther Schock, Stettiner Straße 9, 7157 Murrhardt, 07192-8447

Gerald Schröder, Am Schützenplatz 14, 2105 Seevetal 1, 04105-2602
Rüdiger Sörensen, Thomas-Mann-Straße 3a, 6500 Mainz, 06131-32860

Peter Winter, Tiergartenstraße 29-44, 7030 Böblingen, 07031-289563

Axel Zinser, Podbielskistraße 161, 3000 Hannover 1, 0511-699945 *fifi@hiss*

Bleibt nur noch:

Viele Grüße, frohe Weihnachten und ein gutes neues Jahr!

Tilman Reh (171290) *tilmann.reh@hrz.uni-siegen.dpp.de*

*uweduc@como.unip
jungkumz@doiter.deit.sub.ay*