

386EX-CARD III

Technisches Handbuch

taskit GmbH

**Seelenbinderstr. 33
12555 Berlin (Germany)**

**Telefon +49(0)30 / 611295-0
Fax +49(0)30 / 611295-10
<http://www.taskit.de>**

Alle Rechte an dieser Dokumentation und dem hierin beschriebenen Produkt verbleiben bei

taskit GmbH.

Bei der Erstellung der Dokumentation wurde mit Sorgfalt vorgegangen. Selbstverständlich können Fehler trotzdem nicht vollständig ausgeschlossen werden, so daß weder die o.a. Firma noch der Vertreiber für fehlerhafte Angaben, daraus resultierende Fehlfunktion oder deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen. Waren-, Marken- und Firmennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Kein Teil davon darf ohne ihre schriftliche Genehmigung in irgendeiner Form reproduziert, verarbeitet, vervielfältigt oder verbreitet werden.

Copyright (C) *taskit GmbH*, Berlin.

V1.01 (01.07.2005)

386EX-CARD III

INHALTSVERZEICHNIS

| | |
|---|-----------|
| 1. ALLGEMEINE TECHNISCHE DATEN..... | 7 |
| 1.1. CPU..... | 7 |
| 1.2. SPEICHER..... | 7 |
| 1.3. FIRMWARE..... | 7 |
| 1.4. SCHNITTSTELLEN..... | 7 |
| 1.5. POWER-MANAGEMENT..... | 7 |
| 1.6. SONSTIGES..... | 7 |
| 2. EINFÜHRUNG..... | 8 |
| 3. INBETRIEBNAHME DES 386EX-CARD STARTERKITS..... | 9 |
| 3.1. INHALT DES STARTERKITS..... | 9 |
| 3.2. SOFTWARE INSTALLATION..... | 9 |
| 3.3. PIF-EURO-BASE..... | 9 |
| 3.4. 386EX-CARD STARTEN..... | 10 |
| 3.5. 386EX-CARD DOS PROMPT..... | 10 |
| 3.6. BESCHLEUNIGEN DES BOOT-VORGANGS..... | 11 |
| 3.7. PC-LAUFWERKE ÜBER REMOTE-DRIVE EINBINDEN..... | 11 |
| 3.8. BIOS KONFIGURIEREN..... | 11 |
| 4. PROGRAMMIERUNG DER 386EX-CARD..... | 12 |
| 4.1. VERWENDEN VON PC-COMPILERN..... | 12 |
| 4.2. VERWENDUNG VON DPMI TREIBERN..... | 12 |
| 4.3. ZUGRIFF AUF DIE PIF-PERIPHERIE..... | 12 |
| 4.4. TESTEN VON PROGRAMMEN..... | 12 |
| 4.4.1. <i>Debugging auf dem PC</i> | 12 |
| 4.4.2. <i>Remote Debugging</i> | 12 |
| 5. HARDWARE..... | 13 |
| 5.1. 386EX-CORE..... | 13 |
| 5.2. SPEICHER..... | 13 |
| 5.2.1. <i>RAM</i> | 13 |
| 5.2.1.1. RAM Layout..... | 13 |
| 5.2.1.2. Batterie-Pufferung..... | 13 |
| 5.2.1.3. Goldcap..... | 13 |
| 5.2.2. <i>Flash-Speicher</i> | 13 |
| 5.2.2.1. Flash Speicher Layout..... | 13 |
| 5.2.2.2. Beschränkte Zahl von Lösch-Zyklen..... | 14 |
| 5.2.2.3. Flashdisk..... | 14 |
| 5.2.2.4. Erstellung einer neuen Flashdisk..... | 14 |
| 5.3. IDE-SCHNITTSTELLE..... | 15 |
| 5.4. COMPACT FLASH..... | 15 |
| 5.5. PCF8593 - REAL TIME CLOCK..... | 16 |
| 5.6. TCU (TIMER/COUNTER-UNIT)..... | 17 |
| 5.7. INTERRUPT CONTROLLER..... | 18 |
| 5.7.1. <i>Allgemeines</i> | 18 |
| 5.7.2. <i>Flanken- und Pegeltriggerung</i> | 18 |
| 5.7.3. <i>Zugeordnete Interrupt-Vektoren</i> | 18 |
| 5.7.4. <i>Maskieren von Interrupts</i> | 18 |
| 5.7.5. <i>Zurücksetzen der Interrupt-Controller</i> | 18 |
| 5.7.6. <i>Nicht-Speicherung von IRQs</i> | 19 |
| 5.7.7. <i>Spurious Interrupt</i> | 19 |
| 5.7.8. <i>IRQ-Priorität</i> | 19 |
| 5.8. SERIELLE SCHNITTSTELLEN..... | 20 |
| 5.8.1. <i>Signale der seriellen Schnittstellen</i> | 20 |
| 5.8.2. <i>BIOS Funktionen</i> | 20 |
| 5.8.3. <i>Hardware-Interrupt der seriellen Schnittstellen</i> | 20 |
| 5.8.4. <i>Register der UARTs</i> | 21 |
| 5.9. SYNCHRONE SERIELLE SCHNITTSTELLE..... | 22 |
| 5.10. I/O-PORTS..... | 22 |

386EX-CARD III

| | |
|--|-----------|
| 5.11. PIF-BUS..... | 23 |
| 5.11.1. Überblick..... | 23 |
| 5.11.2. Hardware-Design für den PIF-Bus..... | 23 |
| 5.11.3. PIF-Bus: Mechanik..... | 24 |
| 5.11.4. PIF-Bus Signale..... | 24 |
| 5.11.5. PIF-Bus-Timing (Write)..... | 25 |
| 5.11.6. PIF-Bus-Timing (Read)..... | 26 |
| 5.12. SUPERVISOR-IC..... | 27 |
| 5.12.1. System Reset..... | 27 |
| 5.12.2. Batterie-Umschaltung..... | 27 |
| 5.12.3. Powerfail-Komparator..... | 27 |
| 5.12.4. Watchdog..... | 27 |
| 6. PC-PROGRAMME | 28 |
| 6.1. VTERM | 28 |
| 6.1.1. Kommandozeilen-Parameter | 28 |
| 6.1.2. VTERM-Kommandos | 28 |
| 6.1.3. Datei-Transfer mit VTERM..... | 28 |
| 6.2. FLASHHDD | 29 |
| 6.3. ROMDRV..... | 29 |
| 6.4. BIN2HEX..... | 29 |
| 6.5. HEX2BIN..... | 29 |
| 7. 386EX-CARD PROGRAMME..... | 30 |
| 7.1. EINBINDEN VON REMOTE-LAUFWERKEN MIT RDRIVE, RMAP UND RMCWD..... | 30 |
| 7.2. XLOAD | 30 |
| 7.3. XSEND | 30 |
| 7.4. ZTRANS | 31 |
| 8. BIOS-SETUP..... | 32 |
| 8.1. SETUP HAUPTMENU | 32 |
| 8.1.1. Date und Time..... | 32 |
| 8.1.2. Console Port | 32 |
| 8.1.3. RAM-Disk Size | 32 |
| 8.1.4. ROM-Disk Size..... | 33 |
| 8.1.5. Master HDD / Slave HDD | 33 |
| 8.1.6. Real Mode Mem | 33 |
| 8.2. ADVANCED SETUP | 34 |
| 8.2.1. Power on messages | 34 |
| 8.2.2. Emergency Boot Jumper J1 | 34 |
| 8.2.3. Wait For Key on Error..... | 34 |
| 8.2.4. Fast Boot..... | 34 |
| 8.2.5. IDE-Einstellungen..... | 34 |
| 8.2.6. ROMDOS Einstellungen | 35 |
| 8.2.7. Flash File System Einstellungen..... | 35 |
| 8.2.8. SRAM und Flash Waitstates..... | 35 |
| 8.2.9. IO Cycle length | 35 |
| 8.2.10. HALT switches CPU clock..... | 35 |
| 8.3. I/O CONFIGURATION SETUP..... | 36 |
| 8.3.1. Einstellung der seriellen Schnittstellen:..... | 36 |
| 8.3.2. PRINTER PORTS..... | 36 |
| 8.3.3. X2 Connector Configuration..... | 36 |
| 8.3.4. PORT Initialisierung..... | 37 |
| 8.4. FLASH SETUP | 37 |
| 8.4.1. Flash Update Setup..... | 37 |
| 8.4.2. Flash Backup Setup..... | 37 |
| 8.4.3. Flash Erase Setup | 37 |
| 8.5. EXIT SETUP | 38 |

386EX-CARD III

| | |
|---|-----------|
| 9. BIOS - REFERENZ..... | 39 |
| 9.1. INT 10H - VIDEO SERVICE | 39 |
| 9.1.1. INT 10h Funktion 00h - Set Video Mode | 39 |
| 9.1.2. INT 10h Funktion 02h - Set cursor Position | 39 |
| 9.1.3. INT 10h Funktion 03h - Get current cursor position | 39 |
| 9.1.4. INT 10h Funktion 06h/07h - Scroll current page up/down | 39 |
| 9.1.5. INT 10h Funktion 09h - Write Char/Attribute to Screen | 39 |
| 9.1.6. INT 10h Funktion 0Ah - Write character to screen | 40 |
| 9.1.7. INT 10h Funktion 0Eh - Write Teletype to screen | 40 |
| 9.2. INT 11H - EQUIPMENT CHECK SERVICE | 40 |
| 9.3. INT 12H - MEMORY SIZE | 40 |
| 9.4. INT 13H - DISKETTE SERVICES | 41 |
| 9.4.1. INT 13h Funktion 01h - Read Disk Status | 41 |
| 9.4.2. INT 13h Funktion 02h - Read Disk Sectors | 41 |
| 9.4.3. INT 13h Funktion 03h - Write Disk Sectors | 41 |
| 9.4.4. INT 13h Funktion 08h - Read Drive Parameter | 42 |
| 9.5. INT 14H – FUNKTIONEN DER ASYNCHRONEN SERIELLEN SCHNITTSTELLEN | 42 |
| 9.5.1. INT 14h Funktion 00h – Serielle Schnittstelle initialisieren | 42 |
| 9.5.2. INT 14h Funktion 01h – Zeichen senden | 43 |
| 9.5.3. INT 14h Funktion 02h – Zeichen empfangen | 43 |
| 9.5.4. INT 14h Funktion 03h – Status einer seriellen Schnittstelle abfragen | 43 |
| 9.5.5. INT 14h Funktion 04h - Extended Init | 44 |
| 9.6. INT 15H - SYSTEM SERVICES | 45 |
| 9.6.1. INT 15h Funktion 24h - A20 Gate-Control | 45 |
| 9.6.2. INT 15h Funktion 87h - Move Memory Block | 45 |
| 9.6.3. INT 15h Funktion C0h - Get System Config Table | 45 |
| 9.6.4. INT 15h Funktion A1h - Int 10h / Int 16h I/O umleiten | 45 |
| 9.7. INT 15H FUNKTION C3H - 386EX-CARD SPEZIFISCHE FUNKTIONEN | 45 |
| 9.7.1. INT 15h Funktion C300h - Watchdogfunktion deaktivieren | 45 |
| 9.7.2. INT 15h Funktion C301h - Watchdog freigeben | 45 |
| 9.7.3. INT 15h Funktion C302h - Watchdog zurücksetzen | 45 |
| 9.7.4. INT 15h Funktion C303h - NMI sperren | 45 |
| 9.7.5. INT 15h Funktion C304h – NMI freigeben | 46 |
| 9.7.6. INT 15h Funktion C310h – Prozessortakt abfragen | 46 |
| 9.7.7. INT 15h Funktion C311h – Prozessortakt setzen | 46 |
| 9.7.8. INT 15h Funktion C312h - CPU in den IDLE Mode versetzen | 46 |
| 9.7.9. INT 15h Funktion C313h - CPU in den Stop Mode versetzen | 46 |
| 9.7.10. INT 15h Funktion C314h – synchrone serielle Schnittstelle: Senden | 46 |
| 9.7.11. INT 15h Funktion C315h – synchrone serielle Schnittstelle: Empfangen | 46 |
| 9.7.12. INT 15h Funktion C320h - EEPROM auslesen | 46 |
| 9.7.13. INT 15h Funktion C321h - EEPROM beschreiben | 47 |
| 9.7.14. INT 15h Funktion C322h - I2C-Bus Datenbyte lesen | 47 |
| 9.7.15. INT 15h Funktion C323h - I2C-Bus Datenbyte schreiben | 47 |
| 9.7.16. INT 15h Funktion C324h - I2C-Bus Datenblock lesen | 47 |
| 9.7.17. INT 15h Funktion C325h - I2C-Bus Datenblock schreiben | 47 |
| 9.7.18. INT 15h Funktion C326h - I2C-Bus anfordern | 48 |
| 9.7.19. INT 15h Funktion C327h - I2C-Bus freigeben | 48 |
| 9.7.20. INT 15h Funktion C328h - I2C-Bus Datenbyte lesen mit 2 Adreßbytes | 48 |
| 9.7.21. INT 15h Funktion C329h - I2C-Bus Datenbyte schreiben mit 2 Adreßbytes | 48 |
| 9.7.22. INT 15h Funktion C330h – Hardware Serien-Nummer abfragen | 48 |
| 9.8. INT 16H - KEYBOARD SERVICE | 49 |
| 9.8.1. INT 16h Funktion 00h - Read Keyboard Input | 49 |
| 9.8.2. INT 16h Funktion 01h - Read Keyboard Status | 49 |
| 9.8.3. INT 16h Funktion 05h – Tastendruck simulieren | 49 |
| 9.9. INT 17H - PARALLEL SERVICE | 50 |
| 9.9.1. INT 17h Funktion 00h - Print Character | 50 |
| 9.9.2. INT 17h Funktion 01h - Initialize Printer | 50 |
| 9.9.3. INT 17h Funktion 02h - Get Printer Status | 50 |
| 9.10. INT 18H - BOOT FAILURE | 50 |
| 9.11. INT 19H - BOOT SYSTEM | 50 |

386EX-CARD III

| | |
|---|-----------|
| 9.12. INT 1Ah - UHREN- UND TIMER-FUNKTIONEN | 51 |
| 9.12.1. INT 1Ah Funktion 00h - Read System timer | 51 |
| 9.12.2. INT 1Ah Funktion 01h - Set System timer..... | 51 |
| 9.12.3. INT 1Ah Funktion 02h - Read Real Time Clock | 51 |
| 9.12.4. INT 1Ah Funktion 03h - Set Real Time Clock..... | 51 |
| 9.12.5. INT 1Ah Funktion 04h - Read RTC Date | 51 |
| 9.12.6. INT 1Ah Funktion 05h - Set RTC Date..... | 51 |
| 9.12.7. INT 1Ah Funktion 06h - Set / Enable RTC Interrupt..... | 52 |
| 9.12.8. INT 1Ah Funktion 07h - Disable RTC Interrupt | 52 |
| 9.12.9. INT 1Ah Funktion 08h : Synchronize System Timer | 52 |
| 9.12.10. INT 1Ah Funktion 09h : Setze zyklischen RTC-Interrupt..... | 52 |
| 9.13. INT 1Bh BIS 1Fh..... | 52 |
| 9.14. INT 5Fh - FLASH SERVICES..... | 53 |
| 9.14.1. INT 5Fh Funktion 00h - Flash Erase Block..... | 53 |
| 9.14.2. INT 5Fh Funktion 01h - Flash Read Block..... | 53 |
| 9.14.3. INT 5Fh Funktion 02h - Flash Write Block..... | 53 |
| 9.14.4. INT 5Fh Funktion 03h - Flash Erase and Write Block..... | 53 |
| 9.14.5. INT 5Fh Funktion 04h - Read Flash Chip and Manufacturer ID..... | 54 |
| 10. TABELLEN..... | 55 |
| 10.1. I/O-ADRESSEN | 55 |
| 10.2. INTERRUPT-TABELLE..... | 56 |
| 10.3. STECKERBELEGUNG..... | 58 |
| 10.3.1. PIF (X1) | 58 |
| 10.3.2. I/O (X2) | 58 |
| 10.3.3. COM2 und synchrone serielle Schnittstelle (X3) | 58 |
| 10.3.4. JTAG / I ² C-Bus (X6)..... | 59 |
| 10.4. VOM BIOS VERWENDETE ANSI ESCAPE SEQUENZEN..... | 60 |
| 10.5. DEFAULT-I/O-ADRESSEN DER PIF-CARDS UND PIF-MODULE..... | 61 |
| 10.6. ELEKTRISCHE DATEN..... | 63 |
| 11. MAßZEICHNUNG 386EX-CARD..... | 64 |

1. Allgemeine Technische Daten

1.1. CPU

- Intel 386EX "Embedded" Prozessor (Erweiterung des 386SX) mit 33 MHz

1.2. Speicher

- 2 MB, 4 MB oder 8 MB Flash-Speicher mit Flashfile-System
- Batteriegestütztes SRAM 1 MB, 2 MB, 3MB oder 4MB
- max. 896 kB DOS-Arbeitsspeicher
- Serielles EEPROM mit 256 Byte

1.3. Firmware

- PC-kompatibles BIOS, Konfigurationsmenü im BIOS-Setup
- FreeDOS 7.1
- Datalight ROMDOS (optional, lizenzpflichtig)

1.4. Schnittstellen

- zwei PC-kompatible serielle Schnittstellen (8250-kompatibel), maximal 1,042 MBaud, oder 115,2 kBaud als maximale PC-kompatible Baudrate, TTL-Pegel, durch Anschluß von IF-Modulen (IF232-9DIR, IF485) als RS-232 oder RS-485 konfigurierbar
- PIF-Bus
- IDE-Schnittstelle für den Anschluß von Festplatten und CD-Laufwerken (optional)
- CompactFlash Sockel für Module mit bis zu 256 MB als IDE-kompatible Flash-Disk (optional)
- Ethernet (Twisted Pair, 10 MBit)
- synchrone serielle Schnittstelle bis maximal 8,33 MBaud
- I²C-Bus
- JTAG-Schnittstelle für Testzwecke und zum Nachladen des BIOS
- maximal 13 digitale I/O-Ports, einzeln als Ein- oder Ausgang programmierbar
- 6 freie IRQs
- Echtzeituhr-Alarm Ausgang
- Timer-Clock, Timer-Gate und Timer-Out Signale

1.5. Power-Management

Durch die Powermanagement-Funktionen des BIOS kann eine Anwendung die CPU immer dann, wenn keine Aktivität stattfindet, in den Idle- oder Powerdown-Modus versetzen. Der Stromverbrauch wird dadurch drastisch reduziert.

Im Idle-Mode kann durch einen beliebigen Hardware-Interrupt der aktive Zustand wiederhergestellt werden. Im Powerdown-Mode funktionieren die Interrupts der beiden seriellen Schnittstellen nicht mehr und die Timer-Interrupts nur dann, wenn die Timer mit externem Takt betrieben werden.

Stromverbrauch in verschiedenen Zuständen:

- 350 mA bei 33 MHz Takt, Ethernet Controller aktiviert
- 280 mA bei 33 MHz Takt, Ethernet Controller deaktiviert
- 9,5 mA im Idle Mode bei minimalem Takt (1,5MHz), Ethernet Controller deaktiviert
- 6,2 mA im Powerdown Mode (Oszillator abgeschaltet)

1.6. Sonstiges

- drei 8254-kompatible Timer, davon zwei auch extern zu betreiben
- zwei 8259-kompatible Interrupt-Controller
- Echtzeituhr (Real Time Clock, RTC) Philips PCF 8593, optional mit Batterie-Stützung
- Watchdog, per Software ausschaltbar
- Eindeutige, unveränderbare Hardware-Seriennummer, per Software auslesbar (kann für den Kopierschutz von Anwenderprogrammen verwendet werden). Jeder Serien-Nr.-IC ist nicht nur für die 386EX-Card, sondern weltweit eindeutig.
- lieferbar mit erweitertem Temperaturbereich -40°C ... +85°C
- Abmessungen: 54 x 96 mm

2. Einführung

Die 386EX-Card kann überall dort eingesetzt werden, wo geringer Raum- und Strombedarf eine Rolle spielen und man sich dennoch wegen der leichten Programmierbarkeit eine PC-kompatible Lösung wünscht. Beispiele für Anwendungen sind mobile oder nichtmobile Datenerfassungsgeräte, LCD-Terminals, Meß- und Prüfgeräte, Alarmanlagen oder jede einfache oder weniger einfache Automatisierungsaufgabe.

Die 386EX-Card läßt sich wie ein DOS-PC mit allen üblichen DOS-Compilern programmieren, wie Borland- und Microsoft-C, Pascal, Basic u. a. Der Ablauf beim Erstellen eines Programms ist dabei denkbar einfach: da die Laufwerke des PCs unter dem Terminalprogramm als Laufwerke der 386EX-Card abgebildet werden, kann man die erzeugte .exe-Datei per DOS-Befehl direkt starten - oder auch per DOS-copy erst auf die Flash-Disk kopieren und von dort starten. Zur Erleichterung der Inbetriebnahme gibt es ein Starterkit, das eine Entwicklerplatine mit Stromversorgung und die notwendige Software enthält.

Das BIOS-Setup enthält umfangreiche Konfigurationsmöglichkeiten für Speicheraufteilung, serielle Schnittstellen und I/O-Ports. So werden vom BIOS auch zusätzliche serielle Schnittstellen unterstützt. Die 5 universellen I/O-Pins können einzeln auf Ein- oder Ausgang, mit oder ohne Interrupt-Funktion, eingestellt werden. Beliebige Hardware-Initialisierungen für die Peripherie können im BIOS-Setup vorgegeben werden. Daneben enthält das BIOS-Setup-Menü die Funktionen zum Laden von BIOS-Updates oder des optionalen ROMDOS sowie zum Laden oder Löschen der Flash-Disk.

3. Inbetriebnahme des 386EX-Card Starterkits

3.1. Inhalt des Starterkits

Das 386EX-Card Starterkit besteht aus folgenden Bestandteilen :

- 386EX-Card
- PIF-Euro-Base (PIF-Basis- und Prototyp-Board)
- Steckernetzteil, Eingang AC 230V, Ausgang DC 9 bis 12V, mindestens 400 mA
- Verbindungskabel (serielles Nullmodem-Kabel mit zwei 9-poligen DSUB-Buchsen)
- CD mit Firmware-Dateien zum Produkt (BIOS, Utilities, Libraries) sowie FreeDOS und ROMDOS (Extra-Lizensierung erforderlich)
- Handbuch

3.2. Software Installation

Zur Installation der Software legen sie die beigelegte Diskette in Laufwerk A: oder B: und geben im DOS Prompt **A:INSTALL bzw. B:INSTALL** ein. Das Installationsprogramm erfragt alle notwendigen Daten und kopiert die Software in das angegebene Verzeichnis.

3.3. PIF-Euro-Base

Die PIF-Euro-Base besitzt zwei Steckplätze für CPU- und PIF-Cards. Die CPU-Karte gehört auf Steckplatz A ("Modul-A", nur hier ist der Treiber-IC für die erste serielle Schnittstelle angeschlossen). Der Steckplatz B ("Modul-B") ist zusätzlichen PIF-Cards vorbehalten. Die CPU-Karte wird so aufgesteckt, daß die mit "PIF" bezeichnete Stiftleiste (X1 auf der 386EX-Card, weißes Quadrat als Markierung) auf die mit "PIF" bezeichnete Buchsenleiste der PIF-Euro-Base kommt. **Bitte achten Sie darauf, daß die 386EX-Card nicht seitlich versetzt gesteckt ist.**

Die 25poligen DSUB-Buchsen des PIF-Euro-Base sind mit den I/O-Buchsenleisten von Modul-A und Modul-B (X6 bzw. X7) verbunden. Auf diese Weise sind die I/O-Signale der CPU und anderer PIF-Cards leicht von den Frontseiten eines Europakarten-Gehäuses zugänglich. Optional können statt der DSUB-Buchsen auch gerade Stift- oder Buchsenleisten im 2,54mm-Raster bestückt werden (z. B. für Flachbandkabel-Anschluß).

Die PIF-Anschlüsse (X2 bzw. X3) der beiden Steckplätze sind miteinander verbunden, da sie die Bus-Verbindung zwischen CPU und Zusatzkarten (PIF-Cards) herstellen. Dagegen sind I/O-A und I/O-B nicht verbunden, da die verschiedenen PIF-Cards die unterschiedlichsten I/O-Signale haben. Allerdings werden einige I/O-Signale der 386EX-Card, insbesondere IRQ-Leitungen (Interrupt Request) von einigen PIF-Cards verwendet (z.B. PIF-SIO/LPT). In derartigen Fällen sollte man die PIF-Card besser auf Steckplatz A zwischen PIF-Euro-Base und CPU-Karte stecken. Zu diesem Zweck besitzen derartige PIF-Cards auf der Unterseite zusätzlich Buchsenleisten für PIF-Bus und I/O.

Eine zusätzliche Wannestiftleiste für den PIF-Bus (X1) erlaubt es, weitere PIF-Cards und PIF-Module per Flachbandkabel an den PIF-Bus anzuschließen.

Das PIF-Euro-Base besitzt RS232-Treiber/Empfänger für die Signale RxD, TxD, RTS und CTS der ersten seriellen Schnittstelle der CPU-Karte. Das serielle Verbindungskabel wird am PIF-Euro-Base an den 9 poligen DSUB-Stecker angeschlossen sowie auf der PC-Seite an eine freie serielle Schnittstelle (COM-Port). Falls ein anderes Kabel als das mitgelieferte verwendet wird, muß dieses ein sogenanntes Nullmodem-Kabel sein ("gekreuztes Kabel"). Das mitgelieferte Kabel unterstützt nur TxD und RxD. Die Signale CTS und DCD sind auf der jeweils gleichen Seite mit RTS verbunden.

Das Steckernetzteil wird an der DC-Buchse neben dem Schalter angeschlossen.

3.4. 386EX-Card starten

Im Unterverzeichnis UTIL befindet sich das Terminal-Programm VTERM.EXE. Dieses wird ohne weitere Parameter gestartet. Falls ein anderer COM-Port als COM1 verwendet wird, können Sie durch Drücken von ALT-C einen neuen COM-Port anwählen. Mit ALT-H bekommen Sie eine Übersicht der Kommandos des Terminalprogramms. Schalten Sie die 386EX-Card ein. Bei korrekter Einstellung können Sie nun das Hochfahren des BIOS beobachten. Nach dem Speichertest wird DOS gebootet, und sie gelangen zum DOS-Prompt der 386EX-Card.

3.5. 386EX-Card DOS Prompt

Unter dem DOS Prompt der 386EX-Card kann man wie beim PC gewohnt arbeiten. Die 386EX-Card stellt mehrere "Laufwerke" zur Verfügung. Laufwerk A: ist eine ROM-Disk, Laufwerk B: eine RAM-Disk, Laufwerk C: eine Flashdisk, von der in der Standardeinstellung auch gebootet wird, und Laufwerk D: ein (optionales) CompactFlash-Modul. Auf diese Laufwerke kann wie auf Disketten- bzw. Festplattenlaufwerke im PC zugegriffen werden. Die Laufwerke A: , B: und D: sind standardmäßig deaktiviert, können aber im BIOS-Setup aktiviert werden. Die RAM-Disk (Laufwerk B:) kann dann sofort verwendet werden. Um Laufwerk A: verwenden zu können, muß über das BIOS-Setup erst ein ROMDISK-Image geladen werden.

3.6. Beschleunigen des Boot-Vorgangs

Beim Booten von ROMDOS wartet das System einige Sekunden, um es dem Anwender zu ermöglichen, mit der Taste F5 die Ausführung von config.sys und autoexec.bat zu umgehen bzw. mit F8 die Ausführung im einzelnen zu kontrollieren. Durch den Befehl

```
switches=/f
```

am Anfang der Datei config.sys wird diese Wartezeit vermieden, es ist dann aber auch kein Eingriff in den Boot-Vorgang möglich.

Werden im BIOS-Setup auch noch die Optionen "Boot Delay", "Boot Messages" und "Memory Test" abgeschaltet, so dauert das Starten der 386EX-Card nach dem Einschalten oder Reset bis zum Erscheinen des DOS-Prompts weniger als zwei Sekunden.

3.7. PC-Laufwerke über Remote-Drive einbinden

Neben der Möglichkeit Programme und Daten mit dem PC über X-Modem oder Z-Modem auszutauschen, wozu jedes Terminal-Programm, das über diese Funktionen verfügt, geeignet ist, steht bei Verwendung des mitgelieferten Programms VTERM (im Verzeichnis UTIL) ein Verfahren zur Einbindung von PC-Laufwerken zur Verfügung. Hierzu dienen die Programme rdrive.exe, rmap.exe und rmcwd.exe. In der auf der Flash-Disk befindlichen Datei MAP.BAT finden sich Beispiele, wie (einfach) die Initialisierung der Treiber bewerkstelligt werden kann. Einzelheiten zur Verwendung von des Remote-Drive Systems sind im Kapitel über RDRIVE zu finden.

3.8. BIOS konfigurieren

Das mit der 386EX-Card gelieferte BIOS ist so konfiguriert, daß die Kommunikation über die erste interne serielle Schnittstelle erfolgt. Die Baudrate beträgt normalerweise 57600 Baud. Die Konfiguration der Karte erfolgt im BIOS Setup. Um in das Setup der Karte zu gelangen, muß während des Bootens die Taste S gedrückt werden.

Achtung: Verschiedene Einstellungen im Setup, z.B. das Abschalten der Standard-I/O Schnittstelle, können dazu führen, daß man nicht mehr auf die 386EX-Card zugreifen kann. Im Falle von falschen Einstellungen im Setup, die den weiteren Zugriff verhindern, ist es möglich, durch Kurzschließen des "Config-Jumpers" oder "Emergency Jumpers" J1 wieder Standard-Einstellungen für die COM1 zu bekommen. Das BIOS springt anschließend ins Setup.

Diese Möglichkeit kann durch die Setup-Einstellung "Emergency Jumper: disabled" allerdings auch abgeschaltet werden, um unberechtigte Zugriffe auf die Software zu verhindern.

Dem "Jumper" J1 entspricht auf der Platine ein nicht bestückter SMD-Widerstand (siehe Bestückungsaufdruck).

In keinem Fall sollte während des Abspeicherns der Setup-Einstellungen die Versorgungsspannung abgeschaltet werden, da sonst das BIOS gelöscht werden könnte und kein weiterer Zugriff auf die Karte mehr möglich ist. Das Abspeichern selbst dauert nur wenige Sekunden, anschließend bootet die Karte neu.

4. Programmierung der 386EX-Card

4.1. Verwenden von PC-Compilern

Als DOS-kompatible und weitgehend PC-kompatible CPU-Karte läßt sich die 386EX-Card weitgehend wie ein normaler DOS-PC programmieren. Das heißt, daß die üblichen Programmierwerkzeuge für den PC, so weit sie für DOS-Programme geeignet sind, auch für die 386EX-Card verwendet werden können. Insbesondere ist dabei an Basic, Pascal und "C" zu denken. Gewisse Einschränkungen müssen beachtet werden, was die Ausgabe anbetrifft: Der jeweilige Compiler muß so eingestellt werden, daß die Ausgabe über die Funktionen des BIOS (Int 10h) oder des DOS (diese rufen ihrerseits den Int 10h auf) erfolgt. Nur dann kann die Ausgabe auf die serielle Schnittstelle - die ja den Consolen-Port der 386EX-Card darstellt - umgeleitet werden. Ausgaben, die den Bildschirmspeicher direkt programmieren - eine beliebte, weil schnelle, Methode beim PC - sind auf der 386EX-Card normalerweise nicht möglich (jedoch gibt es für einige LCDs BIOS-Erweiterungen, die den virtuellen 8086-Mode des 386-Prozessors verwenden und Zugriffe auf VGA-Chip und Bildschirmspeicher abfangen und kompatibel umsetzen).

4.2. Verwendung von DPMI Treibern

Die 386EX-Card III besitzt maximal 4 MB RAM. Der RAM-Bereich oberhalb von 1MB kann von Anwendungsprogrammen unter DOS nur für eine RAM-Disk oder aber durch Verwendung des "DOS Protected Mode Interface" auch für Code und Daten voll genutzt werden. Der Einsatz von DPMI-Treibern ist aber in gewisser Weise eingeschränkt: auf die on-board Flashdisk kann nicht mehr geschrieben werden, solange ein DPMI-Treiber aktiv ist. Dies liegt daran, daß das BIOS für die Flashdisk einen eigenen Protected-Mode-Treiber besitzt, der nicht mit DPMI kompatibel ist. Lesezugriffe auf die Flashdisk funktionieren dagegen. Voll funktionsfähig sind im übrigen die optional vorhandene RAM-Disk und die CompactFlash Card.

4.3. Zugriff auf die PIF-Peripherie

Der PIF-Bus wird "I/O-mapped" angesprochen, also durch IN- und OUT-Befehle des 386EX-Prozessors. Dies geschieht auf den 64 I/O-Adressen von 300h bis 33Fh. Der Datenbus des PIF-Bus ist 8 Bit breit.

4.4. Testen von Programmen

4.4.1. Debugging auf dem PC

Programme für die 386EX-Card sollten so weit wie möglich auf dem Entwicklungs-PC selbst getestet werden, da hier die PC-Entwicklungsumgebungen und Debugger uneingeschränkt zur Verfügung stehen. Allerdings sind hier zunächst keine Programm-Zugriffe auf Hardware-Erweiterungen über den PIF-Bus möglich. Es ist aber möglich, über die Interface-Karte PIF-ISA-Base den PIF-Bus direkt an den PC anzuschließen. Diese Karte belegt einen AT-Slot, ihre Hardware-Adressen und IRQ lassen sich flexibel konfigurieren. Diese Methode funktioniert für alle PIF-I/O-Karten (nicht jedoch für LCDs, die BIOS-Erweiterungen verwenden).

4.4.2. Remote Debugging

Eine effektive Methode, Programme im Quelltext direkt auf der 386EX-Card zu debuggen, gibt es in Form des Remote-Kernels des Turbo-Debuggers von Borland. Hierzu muß eine der beiden seriellen Schnittstellen der 386EX-Card nur für den Debugger frei zur Verfügung stehen, irgendwelche anderen Aktivitäten auf der gleichen Schnittstelle stören die Kommunikation zwischen dem Remote-Kernel und dem Host-Programm. Der Remote-Kernel (tdremote.exe) wird auf die Flashdisk der 386EX-Card kopiert und dort gestartet.

5. Hardware

5.1. 386EX-Core

Den CPU Kern des Intel 386EX bildet ein voll statischer 386SX. Dieser hat einen 16 Bit breiten Daten- und einen 26 Bit (386SX: 24 Bit) breiten Adreßbus. Es wird ein Adreßraum von 64 MByte Speicher und 64 kByte I/O bereitgestellt. Der Prozessorkern unterstützt Protected-Mode-Anwendungen.

5.2. Speicher

5.2.1. RAM

5.2.1.1. RAM Layout

Die 386EX Card kann mit bis zu zwei Megabyte SRAM bestückt werden. Dieses ist in zwei Bereiche unterteilt, Lower Memory und Extended Memory. Die Größe des Lower Memory Bereichs kann im Setup mit bis zu 896 kByte eingestellt werden. Dieser Bereich ist im Real Mode des Prozessors adressierbar. Der restliche Speicher wird ab der 1 Megabyte Grenze eingeblendet und steht als Extended Memory zur Verfügung. Durch Freischalten des A20-Gates (s. 386EX-Manual) können ab Segmentadresse FFFFh weitere 64kByte RAM (minus 16 Bytes) im Real Mode angesprochen werden. Allerdings sollte dann keine RAM-Disk eingerichtet sein.

5.2.1.2. Batterie-Pufferung

Das statische RAM und der Uhren-IC (RTC) können über eine Batterie mit Spannung versorgt werden, so daß das RAM beim Abschalten der Betriebsspannung seinen Inhalt nicht verliert und die Uhr weiterläuft. (ACHTUNG : Der RAM-Bereich von DOS und BIOS wird beim Booten gelöscht, nicht jedoch der vom Anwender verwaltete Speicher und das Extended Memory).

Die Pufferung kann durch eine direkt auf der Platine montierte Lithiumbatterie, einen Goldcap oder extern über den I/O-Stecker erfolgen. Die minimale Spannung für SRAM und RTC beträgt 2V. Bei einer Spannung der Lithium-Zelle von 3,2 V verbrauchen SRAM und RTC zusammen etwa 2 µA (typischer Wert).

Die auf der 386EX-Card zur On-Board Bestückung vorgesehene Lithium-Batterie Renata CR2032 hat laut Datenblatt des Herstellers eine Kapazität von 200 mAh und hält damit bei 2µA Belastung theoretisch 11 Jahre (jedoch beträgt die garantierte Lebensdauer nur 10 Jahre, unabhängig von den Einsatzbedingungen). Die Batterie wird nicht belastet, wenn die Betriebsspannung anliegt.

5.2.1.3. Goldcap

Über den Widerstand R18 kann ein Ladestrom zugeführt werden, falls ein "Goldcap", d.h. ein Kondensator mit besonders großer Kapazität (z.B. 1F) statt Batteriepufferung verwendet werden soll. Hiermit ist ein Datenerhalt über einige Stunden möglich. Der Vorteil gegenüber einer Batterie besteht in der Wartungsfreiheit. **Achtung: Bei Verwendung einer Lithium-Batterie muß R18 auf jeden Fall unbestückt bleiben !**

5.2.2. Flash-Speicher

5.2.2.1. Flash Speicher Layout

Die 386EX-Card kann 2 MB, 4 MB oder 8 MB Flashspeicher bestückt werden. Dieser ist in Blöcken zu 128 kB organisiert, die einzeln gelöscht werden können. BIOS und ROMDOS-Kernel belegen 96 kB des Flash-Speichers, 32kB sind für eine BIOS-Erweiterung reserviert. Der Rest steht als Flash-Disk, ROM-Disk oder als frei vom Applikationsprogramm verwendbarer Speicher zur Verfügung. Hierfür gibt es die BIOS-Funktionen des Int 5Fh.

Ein Teil des Flash-Speichers, nämlich 128kB, 256kB oder 512kB (einschließlich BIOS und ROMDOS), kann im untersten MB des Adreßraums, d.h. im Real Mode Adreßraum, eingeblendet werden. Hier kann auch Real-Mode Programm-Code untergebracht werden, der direkt im Flash ausgeführt wird. Dieser Teil des Adreßraums geht dann für das RAM verloren.

5.2.2.2. Beschränkte Zahl von Lösch-Zyklen

Der Flash-Speicher wird aus "Large Sector Flash-ICs" (z.B. 29F080 von AMD oder kompatibel) gebildet. Diese sind für eine beschränkte Zahl von Löschkzyklen pro Block ausgelegt (überlicherweise sind eine Million Löschkzyklen vom Hersteller garantiert). Dies bedeutet, daß der Flash-Speicher, insbesondere die Flashdisk, nicht für permanente Schreiboperationen eines Programms geeignet ist, da man mit einem entsprechenden Programm die zulässige Zahl von Löschkzyklen pro Block in relativ kurzer Zeit überschreiten kann. Für derartige Zwecke muß eine RAM-Disk oder eine Harddisk verwendet werden. Ähnliche Beschränkungen gelten für CompactFlash Cards, die bei der 386EX-Card als optional als zusätzliche Flashdisk eingesetzt werden können.

5.2.2.3. Flashdisk

Der größte Teil des Flash-Speichers wird normalerweise von der Flash-Disk eingenommen die wie eine Festplatte organisiert ist und vom jeweiligen Betriebssystem mittels der BIOS-Funktionen des Int 13h angesprochen wird. Bei Verwendung von FreeDOS befindet sich das komplette Betriebssystem auf der Flashdisk, genauso wie beim PC auf der Festplatte. Wird Datalight ROMDOS eingesetzt befindet sich ein Teil des Betriebssystems quasi als BIOS-Extension in einem schreibgeschützten Speicherbereich und auf der Flashdisk befindet sich vom Betriebssystem nur die Datei COMMAND.COM des ROMDOS, außerdem einige Dienstprogramme für die serielle Datenübertragung (rdrive, xload, xsend) sowie typischerweise das Applikationsprogramm und die Applikationsdaten des Anwenders.

Das Flash-File-System der 386EX-Card ist äußerst stabil gegen plötzliche Stromausfälle, die vor allem bei batteriebetriebenen Geräten leicht auftreten können. Es ist praktisch unmöglich, daß nach einem Stromausfall das Flash-File-System der 386EX-Card nicht mehr funktioniert. Es kann allenfalls vorkommen, daß "verlorene Cluster" entstehen, wenn unter DOS noch Dateien geöffnet waren. Diese lassen sich mit dem DOS-Dienstprogramm CHKDSK wieder löschen.

5.2.2.4. Erstellung einer neuen Flashdisk

- Auf dem PC wird ein Verzeichnis erstellt, daß alle Dateien aufnehmen soll, die später in der Flash-Disk der 386-Card erscheinen sollen.
- Alle benötigten Dateien werden in das zuvor erstellte Verzeichnis kopiert. Beim ROMDOS sollten hierunter zumindest die **command.com** und ein Programm zur Datenübertragung (rdrive.exe und rmap.exe, oder xload.com) fallen. Ebenfalls kann man hier eine passende **config.sys** und **autoexec.bat** erstellen.

Wenn FreeDOS zum Einsatz kommt, muss in diesem Verzeichnis neben der **command.com** und ggf. den Datenübertragungsprogrammen auch **kernel.sys** und **fdconfig.sys** vorhanden sein. Name und Ort der Start-Batchdatei (wie die **autoexec.bat** bei MS-DOS oder ROMDOS) können in der fdconfig.sys frei definiert werden (Standardname: **fdauto.bat**).

- Mit dem Programm **flashhdd.exe** wird eine Flash-Image Datei, entsprechend der Beschreibung in Kapitel 8, erzeugt.
- Nach erneutem Starten der 386-Card verzweigt man durch Betätigen der Taste <S> im Terminalprogramm während des Speichertests in das Setup der CPU Karte.
- Durch <L> oder mit den Cursor-Tasten auf der Tastatur gelangen Sie zum **FLASH**-Menü. Hier wählen Sie die Funktion **Update Flashdisk** aus und bestätigen Sie mit <Enter>. Die Sicherheitsanfrage muß mit 'Y' beantwortet werden. Das BIOS löscht darauf den Flashdisk-Bereich und startet anschließend den Datei-Übertragungsmodus (dies erkennt man am Erscheinen der Protokoll-Zeichen "\$"). Schicken Sie hierauf die Datei durch das Terminalprogramm ab (ALT+<S> bei Vterm, Datei-Sende-Befehl). Wählen Sie hierzu das Übertragungsprotokoll „X-Modem“ und geben Sie den Pfad der zuvor angelegten Flash-Image-Datei ein. Einige Sekunden nach Beendigung der Übertragung bootet die Karte neu.

5.3. IDE-Schnittstelle

An der On-Board IDE-Schnittstelle können maximal zwei Festplatten, CD-Laufwerke, LS100- oder andere IDE-kompatible Geräte angeschlossen werden. Der Steckverbinder im 2mm-Raster ist für 2,5-Zoll Notebook-Festplatten vorgesehen, Geräte mit 2,54mm Steckverbindern benötigen einen Adapter.

5.4. Compact Flash

Zusätzlich zu der immer vorhandenen on-board Flash-Disk mit einer Kapazität von bis zu 8MB ist auf der 386EX-Card ein Sockel für eine CompactFlash Card („CF-Card“) vorgesehen. Diese u.a. durch Digital-Kameras bekannten Flashspeicher-Module sind IDE-kompatibel und lassen sich vom PC aus über USB-Adapter (oder auch PCMCIA-Adapter, falls der PC einen PCMCIA-Slot gibt) auslesen und beschreiben. Die wechselbaren Medien fassen gegenwärtig maximal 4GByte. Der Platzbedarf der 386EX-Card steigt mit eingesetzter Compact-Flash Card nur geringfügig (ca. 6mm in der Dicke).

Der CompactFlash Sockel ist alternativ zur IDE-Schnittstelle, da nicht beide Steckverbinder gleichzeitig bestückt werden können. Eine CompactFlash Card kann allerdings auch an der IDE-Schnittstelle betrieben werden, da beide Schnittstellen im wesentlichen die gleichen Signale besitzen. Hierzu ist eine einfache Adapterplatine mit Flachkabelverbindung notwendig. Ein CompactFlash Sockel kann so auch mechanisch von der CPU-Karte abgesetzt werden, etwa um die CompactFlash Card von einer Gehäuseöffnung aus zugänglich zu machen.

Eine CompactFlash Card wird im BIOS-Setup wie eine Festplatte eingetragen. Sie kann sowohl im CHS-Mode als auch im LBA-Mode betrieben werden. Eine Änderung dieser Einstellung erfordert aber eine erneute Partitionierung und Formatierung. Gegenwärtig ist bei der Formatierung von CompactFlash Cards eher der CHS-Mode verbreitet. Eine fabrikneue, vorformatierte CF-Card ist also wahrscheinlich mit CHS formatiert (dagegen werden Festplatten praktisch immer im LBA-Mode betrieben). CF-Cards, die auf einem PC oder in einer Digitalkamera eine Standard-Formatierung erhalten haben, sind normalerweise mit CHS formatiert, so daß man im Interesse der Kompatibilität dieses Format bevorzugen sollte. Manche fabrikneuen CF-Cards müssen vor dem Gebrauch neu partitioniert und formatiert werden, obwohl sie bereits eine Formatierung zu besitzen scheinen.

Die im Setup eingetragene Festplatte oder CF-Card bekommt unter DOS den Laufwerks-Buchstaben D:. Falls jedoch im Setup der Eintrag "Swap Hdd0 and Hdd1" auf "disable" gestellt wurde, wird C: zugeordnet. Für die Partitionierung und Formatierung können die DOS-Programme FDISK und FORMAT verwendet werden.

5.4.1. LBA und CHS

Der Vollständigkeit halber folgen hier einige Erläuterungen der Begriffe „LBA“ (logical block addressing) und „CHS“ (cylinder/head/sector).

Bei LBA wird der Festplatte oder CF-Card zur Adressierung eines Sektors eine logische Nummer übergeben, die Sektoren sind also fortlaufend nummeriert.

Eine IDE-Festplatte oder CF-Card kann aber auch so konfiguriert werden, daß zur Adressierung drei Werte für Zylinder, Schreib-/Lesekopf und Sektor (d.h. die Sektornummer innerhalb der jeweiligen Spur) verwendet werden (CHS). Die zu verwendenden CHS-Werte können vom BIOS aus dem IDE-Controllers ausgelesen werden („Auto-CHS), sie können vom BIOS gegebenenfalls aber auch geändert werden (was nicht empfohlen wird). Diese Werte haben üblicherweise nichts mit der realen internen Festplattegeometrie zu tun (außer bei sehr alten Exemplaren) und werden vom IDE-Controller nach Bedarf umgerechnet. Die eigentliche Geometrie ist normalerweise komplizierter und von außen nicht erkennbar.

Die BIOS-API (Int 13h), d.h. die Programmierschnittstelle des BIOS, auf die auch das DOS zugreift, verwendet ihrerseits eine eigene CHS-Geometrie, die nicht notwendig kompatibel zu den CHS-Werten der IDE-Schnittstelle ist. Eine eins-zu-eins Umsetzung der Werte führt dazu, daß für jeden der drei Werte (C, H und S) der sowohl für IDE als auch für BIOS/DOS zulässige Minimalwert verwendet werden muß. Daraus würde eine Beschränkung der zulässigen Kapazität von ca. 506MB resultieren. Das BIOS führt deshalb im CHS-Mode eine Umrechnung nach einem sehr einfachen Algorithmus durch und kann dadurch die maximal adressierbare Kapazität auf ca. 7,5GB steigern, was etwa der maximal adressierbaren Kapazität durch die 24-Bit-Parameter des BIOS Int 13h entspricht.

Auch im LBA-Mode muß das BIOS eine geeignete CHS-Umrechnung für die Int 13h Schnittstelle vornehmen. Hierbei werden 63 Sektoren verwendet, für die Köpfe das Minimum der Zahlen 16, 32, 64, 128 und 255 und für die Zylinder das Maximum von 1023, 512, 256 usw., so daß das Produkt der drei Werte kleiner oder gleich der Gesamtanzahl der Sektoren wird. Auch hier entsteht durch die BIOS-Schnittstelle eine Beschränkung der Kapazität auf etwa 7,5GB. Der Zugriff auf größere

Festplatten durch erweiterte BIOS-Funktionen ist bei der 386EX-Card gegenwärtig nicht möglich (zumal aktuelle CF-Cards die erhöhte Kapazität gar nicht nutzen könnten).

5.5. PCF8593 - Real Time Clock

Als Echtzeit-Uhrenbaustein (RTC) wird auf der 386EX-Card III der PCF8593 von Philips verwendet. Dieser kann unabhängig von der Hauptversorgungsspannung mittels einer externen oder auf der Platine bestückten Lithium-Batterie betrieben werden.

Die RTC kann einen Interrupt (IRQ 8) auslösen, und zwar entweder zu einer vorher programmierten Zeit, oder zyklisch alle 1/100s, 1s, 1min, 1h, 1d oder Vielfache dieser Grundintervalle.

Für das Auslesen und Programmieren stehen die Funktionen des BIOS-Interrupts 1Ah zur Verfügung (wie beim PC).

Das low-aktive RTC-Interrupt-Signal (Open-Drain-Ausgang) ist auf den I/O-Stecker der 386EX-Card gelegt (/RTCON) und kann vom Anwender benutzt werden, um Aktionen der Peripherie auszulösen. Dies funktioniert auch dann, wenn die Versorgungsspannung der 386EX-Card abgeschaltet ist und die RTC nur über eine Batterie versorgt wird. Auf Seiten der Peripherie ist dann ein Pull-Up-Widerstand von etwa 2 kOhm an /RTCON zur Batteriespannung erforderlich. Beispielsweise könnte mit Hilfe der RTC die Versorgungsspannung zu einer bestimmten Zeit eingeschaltet werden.

5.6. TCU (Timer/Counter-Unit)

Die TCU ist weitgehend kompatibel zum 8254 von Intel (und damit zum PC). Daneben gibt es noch den (nicht PC-kompatiblen) 32-Bit Zähler des 386EX-Watchdog-Timers.

Eigenschaften des 8254:

- Drei 16-Bit Zähler
- Sechs programmierbare Zählmodi
- BCD oder Binäres Zählen
- Taktquelle intern oder extern wählbar

Der **Timer 0** wird vom BIOS nach PC-Standard im Mode 3 auf eine Ausgangsfrequenz von 18,206 Hz programmiert. Sein Ausgang ist mit der Interruptleitung IRQ 0 verbunden. Die BIOS-Interrupt-Routine inkrementiert bei jedem Aufruf die 32-Bit Timer-Variable bei Adresse 40h:6Ch, welche die Grundlage für die DOS-Systemzeit ist (periodischer Timer-Interrupt, System-Timer).

Der **Timer 1** - beim normalen PC für den DRAM-Refresh zuständig - wie auch **Timer 2** (beim PC für die Ansteuerung des Lautsprechers) stehen zur freien Verfügung. Bei der 386EX-Card können auch Timer 1 und Timer 2 einen Interrupt auslösen (IRQ 10 und 11).

Alle drei Timer können mit internem oder externem Takt betrieben werden. Der interne Takt leitet sich aus dem CPU-Oszillator her.

Der interne Takt wird durch einen Vorteiler aus dem CPU-Takt von 33,33 MHz erzeugt. Der Wert des Vorteilers ist so gewählt, daß der Takt etwas langsamer ist als der PC-typische Eingangstakt von 1,193182 MHz, d.h. als nächsthöhere ganze Zahl zum Quotienten aus 33,33MHz und 1,193182 MHz. Man erhält so einen Vorteiler von 28 und einen Timer-Eingangstakt von 1,1905 MHz. Beim Heruntertakten des CPU-Oszillators durch den BIOS-Int 15h Funktion C311h wird der Vorteiler entsprechend angepaßt.

Folgende Timer-Signale sind nach außen auf den I/O-Stecker (X2) geführt:

| Timer | Output | Clock | Gate |
|---------|--------|--------|--------|
| Timer 0 | Pin 17 | Pin 18 | Pin 15 |
| Timer 1 | Pin 14 | Pin 16 | --- |
| Timer 2 | Pin 12 | Pin 13 | Pin 11 |

Die Signale von Timer 0 und Timer 1 sind mit anderen Funktionen zusammengelegt. Die jeweils richtige Einstellung kann im BIOS-I/O-Setup (siehe dort) vorgenommen werden.

Der **386EX-Watchdog-Timer** ist ein 32-Bit Zähler, der bei der 386EX-Card aber nicht als Watchdog Verwendung findet. Er dient hier normalerweise nur dazu, fehlerhafte PIF-Bus-Zyklen zu beenden (Bus-Monitor-Mode, zur Beendigung von "Ready-Hang-Conditions"). Diese Funktion ist aber nicht unbedingt notwendig, so daß man ihn auch als universellen Zähler/Timer verwenden kann. Der Eingangstakt entspricht dem CPU-Takt (33,33 MHz bei vollem Takt). Der Timer kann einen IRQ15 auslösen.

5.7. Interrupt Controller

5.7.1. Allgemeines

Der 386EX Prozessor besitzt On-Chip zwei 8259-kompatible Interrupt-Controller ("PIC", Programmable Interrupt Controller) und entspricht damit dem PC. Hierdurch stehen wie beim PC 15 Interrupt-Requests (IRQs) zur Verfügung, die teilweise bereits durch Einheiten der 386EX-Card belegt sind. Für Anwendungen frei sind die IRQs 1, 5, 6, 7, 9 und 13. Wird kein RTC-Interrupt benötigt, kann auch der IRQ8 anderweitig verwendet werden (RTC-On Signal, Open-Drain Ausgang der RTC). Wenn die erste oder zweite serielle Schnittstelle ohne Interrupt betrieben werden kann oder nicht benötigt wird, so steht auch IRQ4 bzw. IRQ3 extern zur Verfügung. Zu beachten ist, daß die IRQs 1 und 8 auf dem Stecker invertiert, also low-aktiv sind.

Die IRQ-Signale befinden sich auf dem PIF-Bus-Stecker (X1) oder dem I/O-Stecker (X2) (siehe Tabellen im Anhang).

Der PIC2 ist mit seinem Ausgang an den IRQ2-Eingang des PIC1 angeschlossen. Der IRQ2-Eingang ist deshalb als Slave-Eingang konfiguriert, so daß es den IRQ2 im eigentlichen Sinn nicht gibt. Dem Vektor Int 0Ah ist also auch kein IRQ zugeordnet.

Die Einzelheiten der Programmierung der Interrupt-Controller kann man dem Manual des 386EX-Prozessors von Intel oder der gängigen PC-Literatur entnehmen. Hier sind deshalb nur einige Hinweise aufgeführt.

5.7.2. Flanken- und Pegeltriggerung

Die Interrupt-Controller arbeiten traditionell beim PC mit Flankentriggerung. Eine Umprogrammierung auf Pegel-Triggerung ist nicht unbedingt zu empfehlen, da bei den BIOS Interrupt-Funktionen, speziell denen der RTC und des Timer 0, Probleme auftreten werden. Falls der RTC-Interrupt nicht benötigt wird, kann der PIC2 auf Pegel-Triggerung eingestellt werden. Der Vorteil der Pegel-Triggerung: mehrere Einheiten können sich – durch Wired-OR bzw. Wired-AND Verknüpfung - einen IRQ teilen (das geht bei Flankentriggerung nicht gut, da Flanken verlorengehen, wenn zwei Interrupt-Signale auf einer Leitung gleichzeitig auftreten). Der Nachteil der Pegel-Triggerung besteht darin, daß die Interrupt-Quelle innerhalb der Service-Routine sofort zurückgesetzt werden muß, da sonst weitere Interrupts auftreten.

5.7.3. Zugeordnete Interrupt-Vektoren

Jedem PIC kann per Software-Initialisierung ein Bereich von acht aufeinander folgenden Interrupt-Vektoren zugeordnet werden. Beim PC werden traditionell vom BIOS dem PIC1 die Vektoren Int08h bis Int0Fh und dem PIC2 die Vektoren Int70h bis Int77h zugeordnet. Die Vektoren des PIC1 liegen daher entsprechend der PC-Tradition - wie auch die meisten BIOS-Funktionen - in dem von Intel als "reserved" vorgesehenen Bereich von Int0 bis Int1Fh, d.h. sie teilen sich den Vektor mit gewissen Prozessor-Exceptions. In der Praxis führt dies aber kaum zu Problemen.

5.7.4. Maskieren von Interrupts

Um einen IRQ zu aktivieren, muß das zugehörige Bit im Maskenregister des PIC auf 0 gesetzt werden. Für die IRQs des PIC2 muß zusätzlich das Maskenbit 2 des PIC1 (entsprechend IRQ2) auf 0 gesetzt werden. Die Maskenregister liegen bei den I/O-Adressen 21h bzw. A1h für PIC1 bzw. PIC2.

5.7.5. Zurücksetzen der Interrupt-Controller

Jeder IRQ-Eingang besitzt im Interrupt-Controller ein In-Service-Bit. Die Interrupt-Service Routinen müssen grundsätzlich das In-Service-Bit des betreffenden IRQ zurücksetzen, da andernfalls kein weiterer IRQ mit gleicher oder niedrigerer Priorität mehr erzeugt werden kann. Dies geschieht normalerweise durch den "unspezifischen Rücksetzbefehl"

out [20h], 20h

bzw. für den PIC2:

out [0A0h], 20h

oder in "C":

```
_outp(0x20, 0x20); _outp(0xA0, 0x20);
```

also durch Ausgabe des Byte 20h auf die I/O-Adresse 20h bzw. A0h. Bei den IRQs des PIC2 muß immer auch das In-Service-Bit des IRQ2 zurückgesetzt werden, man muß also beide der angegebenen Out-Befehle ausführen.

Grundsätzlich ist zu beachten, daß die CPU beim Einsprung in eine Interrupt-Routine zunächst alle Interrupts sperrt durch Rücksetzen des Interrupt-Enable Flags. Dies gilt für IRQ-Service-Routinen wie auch für Software-Interrupts (nur im Protected Mode kann hier auch für jeden Interrupt eine andere Einstellung gewählt werden). Man sollte innerhalb der Interrupt-Routine daher möglichst bald (je nach Notwendigkeit der Applikation) die Interrupts wieder freigeben.

5.7.6. Nicht-Speicherung von IRQs

IRQs werden beim 8259 nicht zwischengepeichert (auch wenn das Intel-Manual diesen Eindruck zu erwecken scheint). Nur direkt während eines Interrupt-Acknowledge-Zyklus der CPU werden die Zustände der IRQ-Eingänge eingefroren, um eine eindeutige Ausgabe des Interrupt-Vektors zu ermöglichen. Das Interrupt Request Register (IRR) gibt ansonsten nur den Zustand der IRQ-Eingänge wieder. Dies gilt unabhängig davon, ob ein IRQ ausmaskiert ist. Wenn im flankengetriggerten Modus ein IRQ gerade bearbeitet wird (In-Service-Bit gesetzt), wird das betreffende Bit des IRR als 0 gelesen, da das Edge-Sense-Latch den Eingang sperrt. Das Edge-Sense-Latch wird durch einen Low-Impuls an dem IRQ-Eingang zurückgesetzt, auch wenn das In-Service-Bit noch gesetzt ist, so daß man dann den IRQ-Eingang via IRR wieder einlesen kann.

5.7.7. Spurious Interrupt

Die Nicht-Speicherung von IRQ-Impulsen macht auch den Default- oder Spurious-Interrupt notwendig. Dieser tritt bei "unsauberen" IRQ-Signalen auf, z.B. bei IRQs von prellenden Tastaturen. Wenn die CPU als Reaktion auf einen IRQ einen INTA-Zyklus durchführt, der PIC den zugehörigen IRQ-Eingang aber bereits vergessen hat (da das Signal zurückgenommen wurde), muß der PIC dennoch einen Interrupt-Vektor auf den Datenbus ausgeben (ein Zufallswert auf dem Datenbus könnte sonst zum Absturz des Rechners führen). Dieses ist dann der "Spurious IRQ7". Er wird demgemäß auch dann aufgerufen, wenn der IRQ7 ausmaskiert ist. Das In-Service-Bit des IRQ7 wird beim Spurious IRQ nicht gesetzt. Beim PIC2 kann auch ein Spurious IRQ15 auftreten. Ob ein IRQ7 oder ein IRQ15 auftritt, hängt dann vom Timing des Eingangssignals ab. Wegen der Durchlaufverzögerung des PIC2 liegt das IRQ-Signal am PIC1 etwas länger an, so daß eventuell der PIC1 den IRQ nicht als "Spurious" betrachtet und den PIC2 aktiviert, der dann einen Spurious IRQ15 erzeugt.

5.7.8. IRQ-Priorität

Die Priorität der IRQs ist beim PC üblicherweise so festgelegt, daß der IRQ0 die höchste Priorität besitzt, die anderen IRQs folgen gemäß ihrer Nummer. Da der PIC2 an IRQ2 angeschlossen ist, liegen die IRQs des PIC2 prioritätsmäßig vor dem IRQ3. Die Service-Routinen des PIC2 können allerdings nicht von höher priorisierten IRQs des PIC2 unterbrochen werden, solange das In-Service-Bit des IRQ2 nicht zurückgesetzt wurde. Setzt man aber dieses Bit zurück, können sie auch durch IRQs des PIC1 mit niedrigerer Priorität unterbrochen werden.

Die Prioritäten können auch geändert werden, allerdings nur zyklisch innerhalb der einzelnen PICs. D.h. man legt per Befehl den IRQ mit der niedrigsten Priorität fest, woraus sich automatisch die anderen Prioritäten innerhalb dieses PICs ergeben. Die Festlegung der niedrigsten Priorität erfolgt für den PIC1 bzw. PIC2 durch den Befehl

```
out [20h], 0C0h+ IRQ-Nr  
bzw.  
out [0A0h], 0C0h+ IRQ-Nr
```

Z.B. legt

```
out [20h], 0C3h
```

den IRQ3 auf die niedrigste Priorität, wodurch der IRQ4 (von COM1) die höchste Priorität bekommt.

5.8. Serielle Schnittstellen

Zwei zu dem bekannten 16C450 kompatible UARTs sind auf dem 386EX-Prozessor integriert. Im Gegensatz zu den heute auf PC-Mainboards integrierten UARTs (welche zum 16C550 kompatibel sind) besitzen sie daher auch keine FIFOs.

Die erste serielle Schnittstelle (COM1) wird standardmäßig als Ein-/Ausgabeeinheit der 386EX-Card verwendet (Gerät "CON" von DOS). Man kann für CON aber auch die COM2 oder eine externe serielle Schnittstelle einstellen (möglich sind COM1 bis COM4).

5.8.1. Signale der seriellen Schnittstellen

Beide serielle Schnittstellen besitzen die beim PC üblichen acht Signale: Datenleitungen (RXD, TXD), Modem-Status-Eingänge (DSR, CTS, DCD und RI) und Modem-Control-Ausgänge (RTS, DTR). Häufig wird eines der Paare DTR/DSR oder RTS/CTS für "Handshake-Betrieb" verwendet.

Die acht Signale der COM1 können einzeln als Ein- oder Ausgabeports umkonfiguriert werden (s. Kapitel "I/O-Ports").

Die Signale der RTS, DTR, DSR und RI der COM2 können wahlweise auch als Signale der synchronen seriellen Schnittstelle verwendet werden (s. dort).

5.8.2. BIOS Funktionen

Das BIOS stellt die PC-üblichen INT 14h-Funktionen zur Bedienung der seriellen Schnittstelle zur Verfügung. Abweichend von der PC-Tradition werden diese bei der 386EX-Card mit Empfangs-Interrupt betrieben. Diese Einstellung kann im BIOS-Setup deaktiviert werden. Die im Setup einstellbare Buffer-Größe ist nur bei Interrupt-Betrieb wirksam. Um einen geringeren Rechenzeitaufwand zu erreichen, werden die seriellen Schnittstellen in vielen Anwendungen auch direkt, d.h. durch Zugriff auf die UART-Register, programmiert. Hierbei wird üblicherweise zumindest der Receive-Interrupt verwendet.

5.8.3. Hardware-Interrupt der seriellen Schnittstellen

COM1 und COM2 belegen wie beim PC die IRQ 4 und 3. Falls man für die seriellen Schnittstellen die Interrupts nicht benötigt, können IRQ4 und 3 auch extern für andere Zwecke verwendet werden.

Es gibt pro UART vier verschiedene Interrupt-Quellen, die aber alle die gleiche IRQ-Leitung verwenden und mittels des Interrupt Enable Registers (IER) einzeln freigegeben werden:

- Line-Status-Interrupt – bei Overrun-, Parity- oder Framing-Error oder Break ;
- Receive-Interrupt – ein Zeichen wurde komplett empfangen (Receive Buffer full);
- Transmit-Interrupt – ein Zeichen wurde komplett gesendet (Transmit Buffer empty);
- Modem-Status-Interrupt – der Zustand von DCD, RI, CTS oder DSR hat sich geändert ¹.

Dabei bedeutet

Overrun-Error: Das Receive-Buffer-Register wurde durch ein weiteres Zeichen überschrieben, d.h. das vorige Zeichen wurde vom Prozessor nicht rechtzeitig abgeholt;

Parity-Error: Die Parität oder auch das Paritätsbit (bei forced parity) des empfangenen Zeichens war ungleich dem Komplement des Even Parity Select (EPS) Bit im Line Control Register. Dies geschieht nur bei eingeschaltetem Parity-Bit. Bei "forced parity" kann man das Parity-Bit ähnlich wie ein neuntes Datenbit verwenden, da man es beim Senden mittels des EPS Bits setzen kann und beim Empfangen mittels des Line Status Interrupts auswerten kann.

Framing-Error: Zeichen hatte kein Stop-Bit (oder ein Stop-Bit zuwenig, wenn 1,5 oder 2 Stopbits eingestellt sind). Ein gültiges Stop-Bit bedeutet high-Zustand der RXD-Leitung.

Break-Bedingung: Die RXD-Leitung ging für die Dauer von mehr als einem Zeichen auf low. Das hat immer auch einen Framing-Error zur Folge. Eine Break-Bedingung kann beim Senden durch Setzen des Break Bits des Line-Control-Registers erzeugt werden. Die Mindestzeitdauer von einem Zeichen erreicht man, indem man anschließend ein Zeichen sendet und wartet, bis das Transmit Shift Register leer ist. Danach wird das Break Bit wieder zurückgesetzt. Man kann auf diese Weise dem Empfänger einen Sonderzustand signalisieren, ohne auf bestimmte Byte-Werte oder Byte-Sequenzen zurückgreifen zu müssen.

¹ bei RI: nur steigende Flanken erzeugen einen IRQ.

Man beachte, daß ein Interrupt eines UARTs durch eine geeignete Aktion zurückgesetzt werden muß, bevor ein neuer Interrupt bei der CPU ausgelöst werden kann (Flankentriggerung bei PC-Interrupts!). Dies gilt vor allem auch für zwischenzeitlich hinzugekommene Interrupts anderer Interrupt-Quellen des UARTs. Das Rücksetzen geschieht je nach Interrupt-Quelle durch

| | |
|-------------------------|---|
| Line Status Interrupt: | Auslesen des LSR |
| Receive Interrupt: | Auslesen des Receive-Buffers |
| Modem Status Interrupt: | Auslesen des MSR |
| Transmit Interrupt: | Schreiben des Transmit-Buffers oder Auslesen des IIR, falls der Transmit-Interrupt aktuell die höchste Priorität hat. |

5.8.4. Register der UARTs

Divisor-Latch low (DLL, Adresse 0), Divisor-Latch high (DLH, Adresse 1)

Interrupt Enable Register (IER, Adresse 1):

- Bit 0: Receive Interrupt
- Bit 1: Transmit Interrupt
- Bit 2: Line Status Interrupt
- Bit 3: Modem Status Interrupt
- Bit 4..7: 0

Line Control Register (LCR, Adresse 3):

- Bit 0: Word Length Bit 0
- Bit 1: Word Length Bit 1
- Bit 2: No. of Stop Bits (1 or 2)
- Bit 3: Enable Parity Bit
- Bit 4: Select Even Parity
- Bit 5: Select Forced Parity
- Bit 6: Set Break
- Bit 7: Divisor Latch Enable

Interrupt Identification (Status) Register (IIR oder ISR, Adresse 2):

- Bit 0: 0 = Interrupt Pending
- Bit 1..2: 0 = Modem Status Interrupt
1 = Transmit Interrupt
2 = Receive Interrupt
3 = Line Status Interrupt
- Bit 3..7: 0

Modem Control Register (MCR, Adresse 4):

- Bit 0: /DTR
- Bit 1: /RTS
- Bit 2..3: 0 (Test Bits)
- Bit 4: Set Loop-Back Mode
- Bit 5..7: 0

Line Status Register (LSR, Adresse 5):

- Bit 0: Received Data Ready
- Bit 1: Overrun Error
- Bit 2: Parity Error
- Bit 3: Framing Error
- Bit 4: Break Condition
- Bit 5: Transmitter Hold Register Empty
- Bit 6: Transmitter Shift Register Empty
- Bit 7: 0

Modem Status Register (MSR, Adresse 6):

- Bit 0: Delta CTS
- Bit 1: Delta DSR
- Bit 2: Delta RI
- Bit 3: Delta DCD
- Bit 4: /CTS
- Bit 5: /DSR
- Bit 6: /RI
- Bit 7: /DCD

Scratch Register (SCR, Adresse 7)

5.9. Synchrone serielle Schnittstelle

Der 386EX-Prozessor besitzt neben den beiden asynchronen auch eine synchrone serielle Schnittstelle (SSIO). Für Senden und Empfangen gibt es jeweils eine Daten- und eine Taktleitung. Diese vier Signale liegen auf dem X3-Stecker auf den gleichen Pins wie die Signale DTR, DSR, RI und RTS der COM2.

Die SSIO kann mit maximal dem halben Prozessortakt (CLK2 / 4) betrieben werden, bei der 386EX-Card III mit 33,33 MHz CPU-Takt also mit maximal 16,67 MBaud. Übertragen werden 16-Bit Worte.

Sowohl Empfänger als auch Sender können im Master- oder im Slave-Mode, d.h. mit aktivem oder passivem Takt arbeiten. Der Interrupt IRQ9 wird optional für die Zustände "Transmit Buffer empty" und "Receive Buffer Full" erzeugt.

Bei der Verwendung der synchronen seriellen Schnittstelle sollte man beachten, daß diese herstellerseitig zwei Bugs enthält, die von Intel zwar dokumentiert, jedoch nie behoben wurden:

- Autotransmit Mode. Dieser wäre an sich der normale Master-Transmit Mode. Er ist jedoch nur bei maximaler Baudrate richtig verwendbar, da das erste Bit eines Datenwortes unabhängig von der eingestellten Baudrate mit maximalem Takt ausgegeben wird.
- Das Transmit Buffer Empty Bit des Status-Registers gibt keinen korrekten Wert zurück. Man muß vielmehr das Baudrate-Counter Register "pollen" und abwarten, bis das erste Bit vollständig ausgegeben wurde, bevor man den Transmitter abschalten darf.

5.10. I/O-Ports

Die 386EX-Card besitzt auf dem I/O-Stecker maximal 13 frei programmierbare digitale I/O-Ports. Diese können unabhängig voneinander als Eingang oder Ausgang konfiguriert werden. Diese Pins werden wahlweise auch vom Timer, Interrupt-Controller und der ersten seriellen Schnittstelle verwendet (s. BIOS-I/O-Setup).

Angesprochen werden die Ports als Teil der Port-Register P1, P2 und P3 des 386EX-Prozessors. Die Richtung (Eingang oder Ausgang) wird für einige Pins im BIOS-I/O-Setup, ansonsten mittels dem Richtungs-Registes des Ports festgelegt (0 = Ausgang, 1 = Eingang oder Open-Drain Ausgang). Wenn die Eingänge verwendet werden sollen, müssen die betreffenden Bits des Ausgangsports auf 1 stehen (wegen der Open-Drain-Ausgänge).

| | Konfiguration | Pin-Zustand | Ausgang | Richtung |
|----|---------------|-------------|---------|----------|
| P1 | F820h | F860h | F862h | F864h |
| P2 | F822h | F868h | F86Ah | F86Ch |
| P3 | F824h | F870h | F872h | F874h |

Die einzelnen Bits dieser Register sind den Pins auf dem I/O-Stecker folgendermaßen zugeordnet:

| Port-Bit | X2 | Alternative Funktionen |
|----------|--------|---------------------------------------|
| P1.0 | Pin 9 | DCD von COM1 |
| P1.1 | Pin 7 | RTS von COM1 |
| P1.2 | Pin 6 | DTR von COM1 |
| P1.3 | Pin 2 | DSR von COM1 |
| P1.4 | Pin 3 | RI von COM1 |
| P2.5 | Pin 4 | RXD von COM1 |
| P2.6 | Pin 5 | TXD von COM1 |
| P2.7 | Pin 8 | CTS von COM1 |
| P3.0 | Pin 17 | IRQ4 oder Ausgang von Timer 0 (TOUT0) |
| P3.1 | Pin 14 | IRQ3 oder Ausgang von Timer 1 (TOUT1) |
| P3.3 | Pin 20 | IRQ5 |
| P3.4 | Pin 21 | IRQ6 |
| P3.5 | Pin 22 | IRQ7 |

Die übrigen Bits der Register der Ports P1, P2 und P3 sollten durch Anwendungsprogramme nicht geändert werden, da sie innerhalb der 386EX-Card verwendet werden.

5.11. PIF-Bus

5.11.1. Überblick

Der PIF-Bus ist ein einfacher 8-Bit Erweiterungsbus zum Anschluß von Peripherie-Karten an die 386EX-Card. Die Bus-Architektur ist an die Schnittstellen diverser LCDs angelehnt (deren Stecker-Belegung jedoch nicht einheitlich ist). So lassen sich LCDs mit dem Controller Toshiba T6963C sogar direkt am PIF-Bus betreiben.

Der Adreßraum besteht aus 64 I/O-Adressen. Es werden jedoch nicht 6 Adreßleitungen verwendet, sondern 4 Chip-Select-Leitungen und 4 Adreßleitungen. Von den Chip-Select-Leitungen ist stets nur eine einzige aktiv (1 aus 4 Code). Jedem Chip-Select sind somit 16 I/O-Adressen zugeordnet. Durch dieses Prinzip vereinfacht sich die Adreßdekodierung.

In vielen Fällen wird man sogar ganz ohne Adreßdekodierung auskommen. So kann man etwa den bekannten PIO-Baustein 82C55 direkt am PIF-Bus betreiben, indem man die Signale /CS0, /RD, /WR, A0, A1, die Datenleitungen sowie die Betriebsspannung verwendet. Hierbei würden von den 16 Adressen, die zu Chip-Select 0 (/CS0) gehören, effektiv nur vier verwendet werden, obwohl alle 16 belegt sind. Diese "Verschwendung" von Adressen ist in vielen Systemen, die nur wenig Peripherie benötigen, kein Problem und vereinfacht das Design.

Wesentlich sind die low-aktiven Read- (/RD) und die Write-Leitungen (/WR), von denen bei jedem PIF-Bus-Zugriff genau eine aktiv ist, je nachdem, ob es sich dabei um einen Lese- oder einen Schreibzyklus handelt. Die Daten werden jeweils auf der **steigenden** Flanke, also gegen Ende des Bus-Zyklus, übernommen.

5.11.2. Hardware-Design für den PIF-Bus

Die folgenden Punkte müssen beachtet werden, wenn man Hardware für den Anschluß an den PIF-Bus entwirft.

1. Der Zugriff auf PIF-Peripherie erfolgt durch I/O-Befehle. "Memory-Mapped" Zugriffe sind nicht möglich.
2. Die vier Adreßleitungen des PIF-Bus entsprechen den untersten vier Adreßleitungen des CPU-Busses. Sie können daher jeden Offsetwert von 0 bis 0Fh annehmen.
3. Genau eine Chip-Select Leitung ist bei einem gültigen PIF-Bus Zugriff aktiv (low).
4. Die vier Chip-Select Leitungen werden aus den Adreßleitungen A4 und A5 des CPU-Busses dekodiert. Sie entsprechen daher Offset-Werten von 0h, 10h, 20h und 30h.
5. Die Basis-Adresse des PIF-Bus wird zu den genannten Offset-Werten addiert. Bei der 386EX-Card ist sie 300h. Sie kann bei anderen CPU-Karten anders liegen.
6. Genau eines der Signale /RD und /WR ist während eines gültigen PIF-Bus Zugriffs aktiv (low). Die Peripherie muß diese Signale wie auch die Chip-Select-Signale auswerten, andernfalls können fehlerhafte Bus-Zyklen stattfinden.
7. Die Daten werden sowohl beim Lesen als auch beim Schreiben mit der steigenden Flanke des /RD bzw. /WR Signals übernommen.
8. Die Dauer eines PIF-Bus Zyklus kann im BIOS-Setup auf Werte von 170ns, 320ns, 560ns und 980 ns eingestellt werden. Eine Veränderung des CPU-Taktes bewirkt eine entsprechende Veränderung dieser Werte.
9. Ready-Signal: Dieses Signal wird von der Peripherie-Hardware erzeugt, um PIF-Bus Zyklen zu verlängern. Adressen, Chip-Select und /RD oder /WR bleiben so lange gültig, bis die Peripherie das Ready-Signal wieder freigibt (auf high schaltet). Das Signal besitzt auf der CPU-Karte einen Pull-Up-Widerstand. Die Peripherie muß Open-Collector (Open-Drain) Ausgänge verwenden, wenn mehr als eine Peripherie-Einheit das Ready-Signal verwendet.

5.11.3. PIF-Bus: Mechanik

Die PIF-Bus Signale liegen bei der CPU-Karte auf einer 26poligen, zweireihigen Wannens-Stiftleiste im 2,54mm Raster. Dadurch können übliche Flachbandkabel mit Pfostenverbindern in Schneidklemmtechnik eingesetzt werden. Bei der Verwendung von Flachbandkabeln sollte die Kabellänge 30 cm nicht überschreiten, um Störungen durch Übersprechen und Leitungsreflexionen gering zu halten.

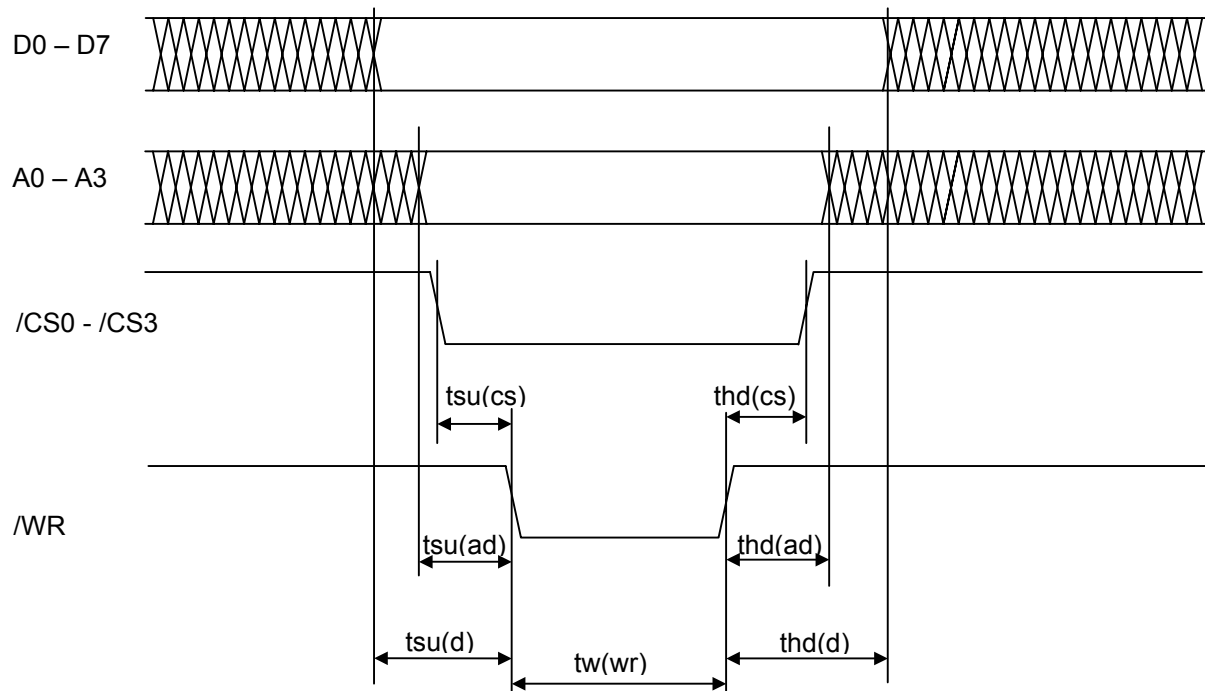
Entfernungen bis ca. 1,5m sind möglich, wenn zusätzliche GND-Leitungen verwendet werden. Insbesondere sollten dann die /RD und /WR Signale durch GND-Leitungen voneinander und von den anderen Signalen abgeschirmt werden. Diese GND-Leitungen sollten an beiden Kabelenden miteinander verbunden sein. Die CPU-Karte muß sich an einem Ende des Kabels befinden. Ausgangsseitige Abschlußwiderstände von 39 Ohm in Reihe zu /RD, /WR, /CSn und A(n) sind zu empfehlen, wenn sich die angeschlossenen Module nur im Bereich der Kabelenden befinden.

In der Mehrzahl der Anwendungsfälle werden CPU-Platine und Peripherie-Karte(n) allerdings direkt zusammengesteckt. Verschiedene lieferbare PIF-Cards sind mit einer Buchsen-/Stecker-Kombination ausgestattet, die auch das Stapeln mehrerer Karten ermöglicht.

5.11.4. PIF-Bus Signale

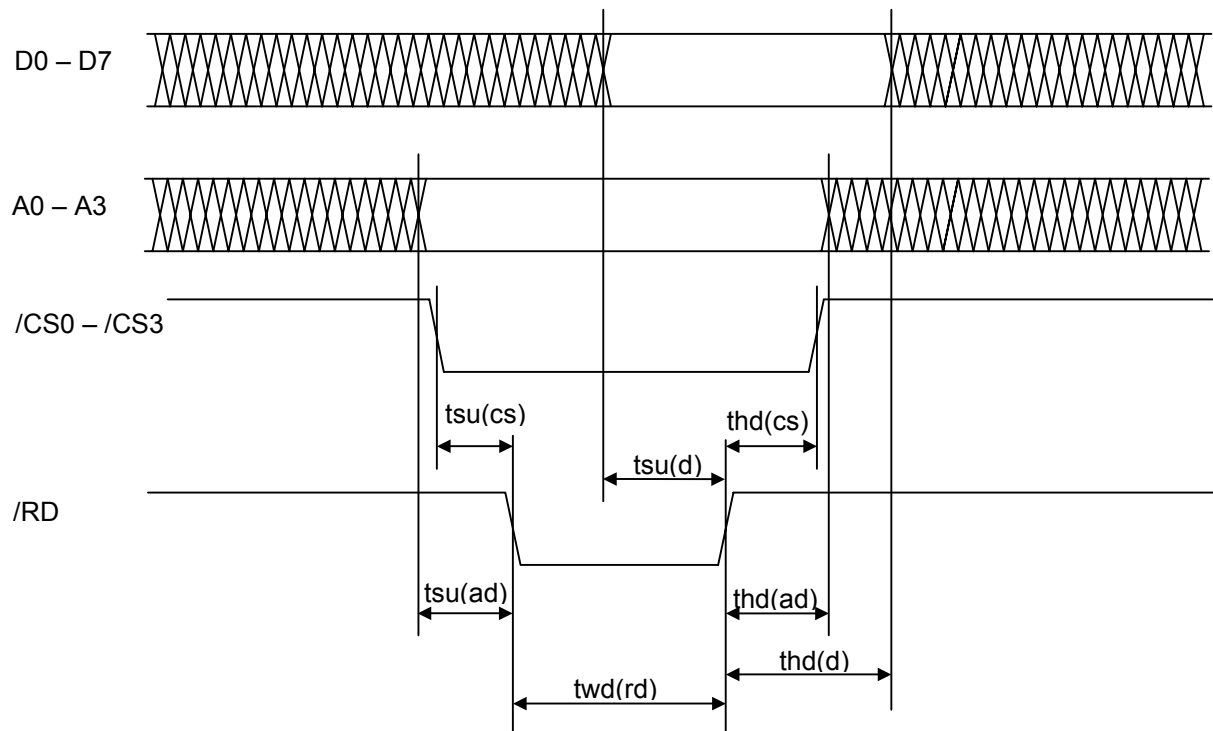
| Signal | Pin-Nr. | I/O | aktiv | Beschreibung |
|---------------|---------------|-----|-------|--|
| D0 ... D7 | 11 ... 18 | I/O | high | Datenleitungen |
| /CS0 ... /CS3 | 7, 22, 23, 24 | O | low | Chip-Select. Bei jedem Bus-Zyklus ist jeweils genau ein Chip-Select aktiv |
| A0 ... A3 | 8, 9, 20, 21 | O | high | Adreßleitungen |
| /RD | 6 | O | low | Read-Signal. Ist bei jedem Lese-Zugriff aktiv. |
| /WR | 5 | O | low | Write-Signal. Ist bei jedem Schreib-Zugriff aktiv. |
| /RESET | 10 | O | low | Reset-Signal. Dies ist das gepufferte Ausgangssignal des Reset-Generators MAX690 der 386EX-Card (siehe dort) |
| /INT | 25 | I | low | Interrupt-Request. Auf der 386EX-Card wird dieses Signal invertiert an den IRQ1 geführt. Es besitzt einen 10kOhm Pull-Up Widerstand. |
| Ready | 19 | I | high | Ready: dient zur Verlängerung von PIF-Bus-Zyklen durch Peripherie-Einheiten (low = not ready). Der Bus-Zyklus wird von der CPU erst beendet, wenn das Ready-Signal wieder high ist. |
| VCC | 3 | | | 5V-Versorgungsspannung |
| VEE | 4 | | | Kann für die Kontrastspannung von LCDs verwendet werden, auf der 386EX-Card nicht kontaktiert. |
| /CS4 | | I | low | Wenn der (normalerweise nicht bestückte) Widerstand R9 mit 0 Ohm bestückt wird, erhält man ein zusätzliches Chip-Select-Signal. Es wird mit den I/O-Adressen 340h bis 34Fh angesprochen. |
| GND | 1, 2, 26 | | | Masse (negativer Anschluß der Versorgungsspannung) |

5.11.5. PIF-Bus-Timing (Write)



| | | | |
|-----------|------------------------|---|--------------|
| $tsu(cs)$ | Chip Select setup time | /CS low to /RD or /WR low (Chip Select valid to I/O valid) | ≥ 70 ns |
| $thd(cs)$ | Chip Select hold time | /RD or /WR high to /CS high (Chip Select hold after I/O invalid) | ≥ 20 ns |
| $tsu(ad)$ | Address setup time | A0..A3 valid to /RD or /WR low | ≥ 70 ns |
| $thd(ad)$ | Address hold time | A0..A3 valid after /RD or /WR high (address hold after I/O invalid) | ≥ 20 ns |
| $tsu(d)$ | Data setup time | D0..D7 valid to /RD or /WR low | ≥ 70 ns |
| $thd(d)$ | Data hold time | D0..D7 valid after /RD or /WR high (data hold after I/O invalid) | ≥ 20 ns |
| $tw(wr)$ | Write pulse width | D0..D7 valid to /WR low | ≥ 70 ns |

5.11.6. PIF-Bus-Timing (Read)



| | | | |
|---------|------------------------|--|---------|
| tsu(cs) | Chip Select setup time | /CS low to /RD low (Chip Select valid to Read valid) | ≥ 70 ns |
| thd(cs) | Chip Select hold time | /RD high to /CS high (Chip Select hold after Read invalid) | ≥ 20 ns |
| tsu(ad) | Address setup time | A0..A3 valid to /RD low | ≥ 70 ns |
| thd(ad) | Address hold time | A0..A3 valid after /RD high (address hold after Read invalid) | ≥ 20 ns |
| tsu(d) | Data setup time | D0..D7 valid to /RD high (Data valid to Read invalid) | ≥ 70 ns |
| thd(d) | Data hold time | D0..D7 valid after /RD high (data hold after I/O invalid) | ≥ 0 ns |
| tw(rd) | Read pulse width | | ≥ 70 ns |

5.12. Supervisor-IC

Der auf der 386EX-CARD eingesetzte IC MAX690 (oder kompatibel) erfüllt eine Reihe von zentralen Aufgaben, weshalb er auch als "Supervisor-Chip" bezeichnet wird.

5.12.1. System Reset

Beim Einschalten erzeugt MAX690 einen definierten Reset-Impuls von 50ms Länge. Dieser Impuls wird unabhängig von der Anstiegsgeschwindigkeit der Versorgungsspannung erzeugt. Dieses Signal erscheint auch als PIF-Reset auf den Steckerverbindern X1 und X2.

Sinkt die Versorgungsspannung unter 4,65 Volt, geht der Reset-Ausgang auf Low. Daraufhin ist jede Aktivität der Karte gesperrt.

5.12.2. Batterie-Umschaltung

Bei Ausfall der Versorgungsspannung werden das statische RAM und die Echtzeituhr von Netz- auf Batteriebetrieb umgeschaltet (sofern eine Batterie bestückt ist bzw. eine entsprechende Spannung am V_{batt} -Eingang des X2-Steckers anliegt).

5.12.3. Powerfail-Komparator

Der MAX690 besitzt einen Spannungskomparator, dessen Ausgang zur Erzeugung eines NMI (nicht-maskierbaren Interrupts) verwendet wird. Die Komparatorschwelle ist mit Widerständen einzustellen. Überwacht wird der Vunst-Eingang des Steckers X2, an den typischerweise die unstabilierte Versorgungsspannung angeschlossen wird. Diese "Power-Fail"-Erkennung dient z.B. zur Sicherung wichtiger Daten oder des aktuellen Programmstatus bei einem plötzlichen Ausfall der Versorgungsspannung. Der Power Fail Input des MAX690 löst den NMI bei einer Spannung von 1,3 V aus. Der Spannungsteiler R24 / R23 muß V_{unst} derart herunterteilen, daß im Normalzustand der MAX690 noch nicht anspricht.

Beispiel: Minimum der unstabilierten Spannung im Normalfall = 9V. Nach Ausfall der Versorgung wird die Schaltung noch eine Zeitlang über die Ladeelkos versorgt. Sinkt die Spannung unter 8V, wird ein NMI ausgelöst. Für den Spannungsteiler gilt

$$8V / 1,3V = (R24+R23) / R23. \text{ Setzt man } R23 = 1k, \text{ so ist } R24 = 1k * (8V / 1,3V - 1) = 5,1k.$$

Der Widerstand R22 (Pull-Up nach V_{cc}) sollte bei Bestückung des Spannungsteilers entfernt werden.

Durch die Bestückung des Widerstands R21 kann eine Hysterese eingestellt werden. Bezeichnen U_{max} und U_{min} die Grenzen des Hysteres-Fensters und U_{nmi} die MAX690-Resetschwelle von 1,3V, so gilt

$$R21 = R24 * V_{\text{CC}} / (U_{\text{max}} - U_{\text{min}})$$

$$R23 = R21 / (U_{\text{max}} / U_{\text{nmi}} - (R24 + R21) / R21)$$

5.12.4. Watchdog

Um die Betriebssicherheit der Baugruppe zu erhöhen, kann der "Watchdog"-Mechanismus des MAX690 benutzt werden. Wenn bei aktiviertem Watchdog nicht mindestens alle 1,6 Sekunden eine Flanke am "Watchdog-Input" (WDI) den internen Zähler des MAX690 zurücksetzt, wird ein Reset ausgelöst. Der Watchdog kann per Software gesperrt sowie freigegeben und zurückgesetzt werden. Hierzu dienen die Unterfunktionen 0, 1 und 2 der Funktion C3h des INT 15h (siehe BIOS-Referenz).

6. PC-Programme

6.1. VTERM

VTERM.EXE ist das Standard-Terminalprogramm für die 386EX-Card und damit die unentbehrliche Verbindung zur 386EX-Card während der Software-Entwicklung.

6.1.1. Kommandozeilen-Parameter

VTERM läßt sich mit folgenden Kommandozeilen-Parametern aufrufen:

-? : Kommandozeilen-Parameter Übersicht
-b(baud) : Übertragungsrate einstellen
-c(1-4) : Seriellen Port auswählen
-m : Schwarz/Weiß Darstellung wählen
-o : Logdatei öffnen
-t(AHT) : Terminalemulation wählen

6.1.2. VTERM-Kommandos

Folgende Tasten sind mit Funktionen belegt :

ALT-B : Übertragungsrate einstellen
ALT-C : Seriellen Port wählen
ALT-D : Remote-Laufwerke zuweisen
ALT-E : Lokales ECHO ein/ausschalten
ALT-F : Handshake einstellen
ALT-H : Hilfe aufrufen
ALT-O : Logdatei öffnen/schließen
ALT-P : Übertragungsparameter einstellen
ALT-R : Datei empfangen
ALT-S : Datei senden
ALT-T : Terminal Emulation einstellen
ALT-W : Einstellungen sichern
ALT-X : VTERM verlassen
ALT-Y : Bildschirm löschen
ALT-Z : DOS Kommando

Zusätzlich zu den üblichen Terminalfunktionen (Ausgabe auf dem Bildschirm, Eingabe über die Tastatur des Host-PC, sowie Datei-Transfer) erlaubt VTERM den direkten Zugriff von der 386EX-Card auf die Laufwerke des PCs mit Hilfe des TSR-Programms RDRIVE sowie der Alternativmethode unter Verwendung des Treibers RMTDRV.SYS.

6.1.3. Datei-Transfer mit VTERM

Abgesehen von der Dateiübertragung via RDRIVE (oder ggf. RMTDRV.SYS), die automatisiert abläuft, muß VTERM zur Unterscheidung von der normalen Bildschirm-Tastatur-Ein/Ausgabe mit besonderen Kommandos auf Datei-Empfang bzw. Datei-Senden eingestellt werden.

Dies betrifft insbesondere die Datei-Transfers des BIOS-Setup (Flash-Update/Backup) und die Kommunikation mit XSEND und XLOAD. Man startet zunächst den Transfer auf der 386EX-Card und stellt anschließend mit ALT-R bzw. ALT-S VTERM auf Empfang oder Senden.

Die Übertragung auf der Seite der 386EX-Card findet im allgemeinen per Xmodem-Protokoll statt, daher muß VTERM ebenfalls auf Xmodem eingestellt werden. Danach wird der Name der Datei eingegeben, die gesendet werden bzw. der Name, unter welchem die zu empfangende Datei abgespeichert werden soll (Xmodem überträgt den Dateinamen nicht).

6.2. FLASHHDD

Flashhdd.exe erlaubt die Erstellung einer Flash-Image-Datei basierend auf dem Inhalt eines beliebigen Verzeichnisses. Diese wird per BIOS-Setup als Laufwerk C: in den Flash-Speicher der 386EX-Card übertragen. Um DOS booten zu können, müssen sich bei FreeDOS alle benötigten Startdateien (mindestens KERNEL.SYS, COMMAND.COM und FDCONFIG.SYS) in dem betreffenden Verzeichnis befinden. Bei Datalight ROMDOS reicht im Prinzip die COMMAND.COM allein aus.

Aufruf:

```
FLASHDD [/B<n>] [/F<n>] [/S<n>] [/V][/?] <Quellverzeichnis> [<Zieldatei>]
```

Optionen:

```
/B<n>  n = Zahl der Blöcke (default 14).
/S<m>  m = Blockgröße in kB (128 bei der 386EX-Card III, default 128).
/V     Verbose (Dateien und Verzeichnisse anzeigen)
/?     Hilfe anzeigen
/??   Hilfe in Englisch (ab Version 1.32)
/M     MS-DOS kompatibles Format der Flashdisk (MS-DOS Systemdateien hinzufügen)
/F     FreeDOS kompatibles Format der Flashdisk (ab Version 1.32)
```

Der maximale Wert für n hängt von der Bestückung gemäß der folgenden Tabelle ab:

| Gesamtkapazität | Flash-ICs bestückt | n |
|-----------------|--------------------|----|
| 2 MB | 2 mal 29F080 | 15 |
| 4 MB | 2 mal 29F016 | 31 |
| 8 MB | 2 mal 29F032 | 63 |

Man achte darauf, daß die Zieldatei nicht versehentlich im Quellverzeichnis erstellt wird (sonst Fehler durch Rekursion).

Falls MS-DOS oder FreeDOS zum Einsatz kommen, ist es notwendig, ein ggf. im Speicher befindliches ROMDOS im BIOS-Setup zu deaktivieren oder zu löschen.

6.3. ROMDRV

Romdrv.exe erlaubt die Erstellung einer ROMDISK-Image Datei basierend auf den Inhalt eines beliebigen Verzeichnisses. Diese wird per BIOS-Setup als Laufwerk A: in den Flash-Speicher der 386EX-Card übertragen.

Aufruf:

```
romdrv <Quellverzeichnis> [<Zieldatei>]
```

Man achte darauf, daß die Zieldatei nicht versehentlich im Quellverzeichnis erstellt wird (sonst Fehler durch Rekursion).

6.4. Bin2hex

BIN2HEX ist ein Programm für das Erstellen einer Intel-Hex86-Datei aus einer Binärdatei (z.B. ".COM"-File).

6.5. Hex2bin

HEX2BIN erzeugt eine Binär-Datei aus einer Intel-Hex86-Datei.

7. 386EX-Card Programme

7.1. Einbinden von Remote-Laufwerken mit RDRIVE, RMAP und RMCWD

Dieses Programm ermöglicht das Einbinden der PC-Laufwerke in als Laufwerke der 386EX-Card. Das Programm wird beim Aufruf resident geladen. Danach können Dateien vom und zum Host-PC wie in einem Netzwerk z.B. per copy-Befehl übertragen werden. Dies ist für die 386EX-Card das Standardverfahren für die Übertragung von Dateien. Außerdem können Programme direkt vom Host-Laufwerk aus auf der 386EX-Card gestartet werden, ohne sie erst auf ein lokales Laufwerk zu kopieren.

Aufruf:

```
rdrive [-?] [-c<n>] [-u]
```

```
-?           : Hilfe
-c<1..4>    : wähle COM-Port 1 bis 4 (default: COM1)
-u          : entfernt RDRIVE aus dem RAM-Speicher
```

Zum Ändern der seriellen Schnittstelle mit -c muß RDRIVE zuvor mit -u entladen werden.

Mit dem Programm RMAP können nun Laufwerke und Verzeichnisse des Host-PC auf Laufwerke bzw. Laufwerksbuchstaben der 386ex-Card abgebildet werden. Dabei bezeichnet LOCAL den auf der 386EX-Card zu verwendenden Laufwerksbuchstaben und REMOTE das Laufwerk oder Verzeichnis des Host-PC.

```
RMAP /LOCAL=D /REMOTE=C
```

stellt beispielsweise das Laufwerk C: des Host-PC als Laufwerk D: der 386ex-Card zur Verfügung. Es gibt keine feste Regel, welche Buchstaben in welcher Reihenfolge benutzt werden müssen – unabhängig davon, ob es sich bei den Remote-Laufwerken um lokale Laufwerke des Host-PC oder um Netzlaufwerke handelt.

Das Abbilden von Verzeichnissen ist genauso einfach:

```
RMAP /LOCAL=E /REMOTE=C:\Programs\Files386
```

Die Zuweisungen können jederzeit überschrieben oder mit Angabe des lokalen Laufwerks-Buchstaben gelöscht werden:

```
RMAP /LOCAL=D
```

Der Befehl RMAP allein gibt ein Liste der aktuellen Zuweisungen aus.

Das Programm RMCWD.EXE weist automatisch das Verzeichnis, aus dem heraus Vterm gestartet wurde, dem angegebenen LW-Buchstaben zu.

7.2. XLOAD

Xload ist ein einfaches Programm, um Dateien vom Host in die RAM- oder Flash-Disk zu übertragen. Hierzu wird das Übertragungsprotokoll XMODEM eingesetzt. Nach dem Aufruf mit

```
xload [com port ] file
```

wartet Xload, bis auf der Hostseite die Übertragung gestartet wird.

7.3. XSEND

Xsend ist ein einfaches Programm, um Dateien von der 386EX-Card zum Host zu übertragen. Hierzu wird das Übertragungsprotokoll XMODEM eingesetzt. Nach dem Aufruf mit

```
xsend [com port ] file
```

muß auf der Hostseite der Empfang gestartet werden.

7.4. ZTRANS

Ztrans bietet gegenüber Xsend und Xload erweiterte Funktionalität, insbesondere die Übertragung des Dateinamens sowie die Übertragung mehrerer Dateien mit einem Befehl. Das zugrundeliegende Protokoll ist ZMODEM. Dieses wird von VTERM nicht unterstützt. Stattdessen muß ein ZMODEM-fähiges Terminalprogramm eingesetzt werden (z.B. Windows Hyperterminal), oder auf dem Host muß ebenfalls Ztrans gestartet werden.

Aufruf:

```
ztrans [/R] [/Bn] [/Cn] [/?] <Datei(en)>
```

Optionen

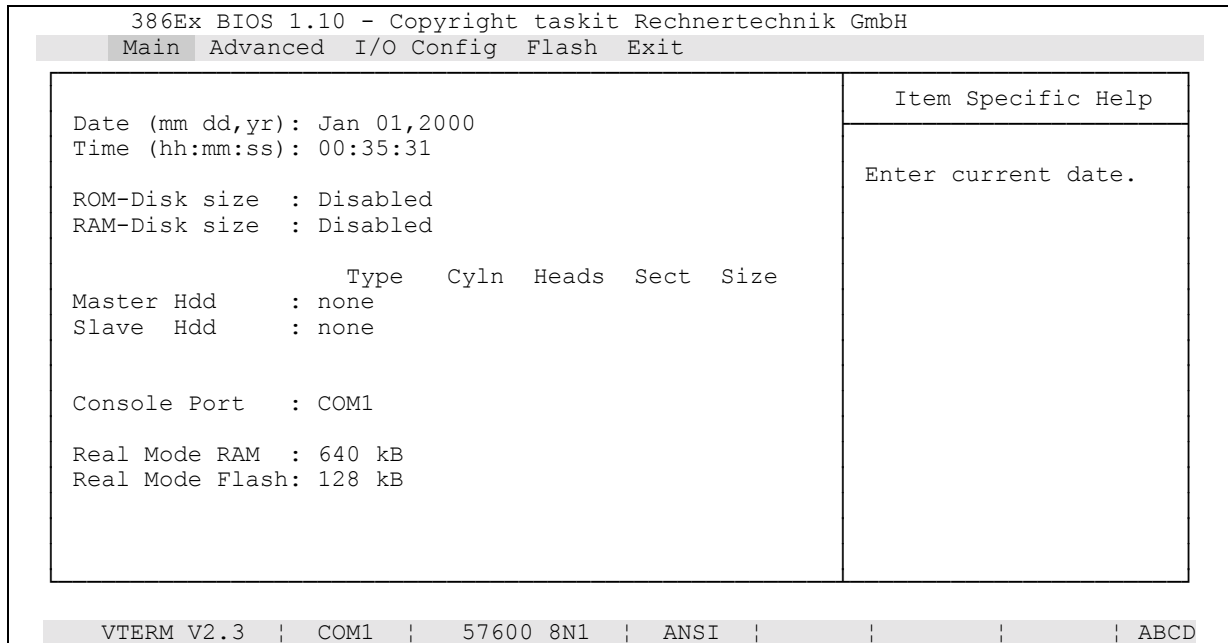
| | |
|-----|------------------------|
| /R | Empfangen statt Senden |
| /Bn | Baudrate einstellen |
| /Cn | Schnittstelle wählen |
| /? | Hilfe anzeigen |

Für <Datei(en)> sind Wildcards möglich.

8. BIOS-Setup

Das Setup der 386-Card bietet diverse Einstellungsmöglichkeiten, um die Baugruppe an die Bedürfnisse der Anwendung anzupassen. Um in das Setup zu gelangen, muß nach einem Neustart der Baugruppe am angeschlossenen Terminalprogramm während des Speichertests die Taste <S> gedrückt werden. Daraufhin wird der Speichertest abgebrochen und es erscheint das Setup-Hauptmenu. Hier kann mit den Zifferntasten oder den Cursortasten ein Menüpunkt ausgewählt werden.

8.1. Setup Hauptmenu



8.1.1. Date und Time

Einstellen der Echtzeituhr (RTC). Diese Werte bleiben nach dem Abschalten nur dann erhalten, wenn eine Lithium-Batterie bestückt ist oder eine andere Spannungsquelle an den Vbatt-Anschluß des I/O-Steckers angeschlossen ist.

8.1.2. Console Port

Hier wird diejenige serielle Schnittstelle eingetragen, über die das BIOS-Setup sowie das ROMDOS (Gerät "CON") angesprochen werden sollen. **Achtung:** wird hier "none" oder eine nicht vorhandene Schnittstelle eingestellt, so kann beim nächsten Start nicht mehr auf das Setup und den DOS-Prompt zugegriffen werden. Dies ist zuweilen sinnvoll, um unberechtigte Zugriffe auf die 386EX-Card zu verhindern. Ein Rücksetzen des Console-Ports ist aber immer noch durch Kurzschließen des Jumpers J1 während des Starts möglich. Das BIOS stellt dann automatisch Default-Werte für die erste serielle Schnittstelle ein und springt ins Setup.

8.1.3. RAM-Disk Size

Hier wird die Größe der RAM-Disk im Extended Memory eingestellt. Die RAM-Disk wird automatisch vom BIOS nach dem Reset initialisiert und kann als Laufwerk A: (wenn keine ROM-Disk vorhanden ist, sonst Laufwerk B:) angesprochen werden. Falls das SRAM mit Batterie-Stützung betrieben wird, bleibt der Inhalt der RAM-Disk nach dem Abschalten erhalten. In diesem Fall unterbleibt die Neu-Initialisierung durch das BIOS. Änderungen in der RAM-Konfiguration (Menüpunkt "Real Mode Mem") führen beim Neustart zur Neu-Initialisierung der RAM-Disk (und damit zum Löschen der darin enthaltenen Daten).

8.1.4. ROM-Disk Size

ermöglicht das Einrichten einer schreibgeschützten ROM-Disk. Die Kapazität beträgt 128 kB, 256 kB oder 384 kB. Die ROM-Disk ist - falls vorhanden - immer Laufwerk A:

8.1.5. Master HDD / Slave HDD

Falls eine Festplatte oder ein CompactFlash Modul an die 386EX-Card angeschlossen ist, muß diese hier eingetragen werden. Bei aktuellen Harddisks und CompactFlash wird normalerweise "LBA" ("logical block addressing") oder "CHS" ("cylinder/head/sector") gewählt, ansonsten kann eine frei definierbare Geometrie ("User") oder eine von 46 vorgegebenen Geometrien eingetragen werden. Die Geometrie ist dabei durch die Zahl der Zylinder, die Zahl der Schreib-/Lese-Köpfe und die Zahl der Sektoren pro Spur bestimmt. Aktuelle Festplatten und CompactFlash-Module verwenden dabei "virtuelle" Werte, die nicht notwendig der physikalischen Geometrie entsprechen.

Die Software-Schnittstelle von BIOS und DOS – der Int 13h – unterstützt nur Kapazitäten von maximal 8,2 GB (genau: 1024 Zylinder x 255 Köpfe x 63 Sektoren). Falls eine Festplatte eine höhere Kapazität besitzt, ist der über 8,2 GB liegende Speicherplatz nur mit Hilfe zusätzlicher Software-Treiber zugänglich.

Bei "LBA"-Einstellung verwendet das BIOS folgende Geometrien für den Int13h:

63 Sektoren;
16, 32, 64, 128 oder 255 Köpfe (das Minimum je nach verfügbarer Kapazität);
maximal 1024 Zylinder.

Bei "CHS" Einstellung liest das BIOS die Geometrie aus den internen Parametern der Festplatte aus. Da diese sich an den Beschränkungen der IDE-Schnittstelle (65535/16/63) und nicht denen des Int13h orientieren, wird – falls notwendig – vom BIOS eine Übersetzung in Int13h-kompatible Werte durchgeführt.

Hierzu wird solange die Zahl der Köpfe verdoppelt und die Zahl der Zylinder durch 2 geteilt, bis die Zahl der Zylinder kleiner oder gleich 1024 ist. Die Zahl der Köpfe muß dabei unterhalb 256 bleiben. Die Zahl der Zylinder wird anschließend auf maximal 1024 eingeschränkt.

Da die gewählte Geometrie sich auch auf die Formatierung durch das Betriebssystem auswirkt (insbesondere auf die Lage des Bootsektors am Anfang der Spur 1), muß die vor der Formatierung gewählte Geometrie für das betreffende Medium immer beibehalten werden.

Die vom BIOS tatsächlich verwendete Int13h-Geometrie wird beim Booten in der BIOS Configuration Box angezeigt.

8.1.6. Real Mode Mem

Hier wird eingestellt, wieviel des Flash- und RAM-Speichers im untersten Megabyte des CPU-Adreßraum eingeblendet werden soll. Diese Speicherbereiche sind dann im Real Mode der 386EX-CPU zugänglich. Möglich sind

für das Flash: 128 kB,
256 kB,
512 kB.

für das RAM: 256 kB,
512 kB,
640 kB,
768 kB,
896 kB.

Die Summe beider Werte kann 1 MB nicht überschreiten. Das RAM ist immer ab Adresse 0 zugreifbar, der Flashspeicher unmittelbar unterhalb von 100000h (1 MB). Default-Werte sind 128 kB Flash und 640 kB RAM. Größere Werte für das RAM sind unter anderem dann sinnvoll, wenn mehr DOS-Arbeitsspeicher benötigt wird. Wird dagegen eine größere RAM-Disk benötigt, so muß ein entsprechend kleinerer Wert eingestellt werden. Der Speicherbereich der RAM-Disk darf sich nicht mit dem DOS-Speicher überschneiden.

8.2. Advanced Setup

| 386Ex BIOS 1.10 - Copyright taskit Rechner-technik GmbH | |
|---|--------------------|
| Main Advanced I/O Config Flash Exit | |
| Power on messages : Enabled | Item Specific Help |
| System Configuration Box : Enabled | |
| Display "Hit <S>..." : Enabled | |
| Emergency Boot Jumper J1 : Enabled | |
| Wait For Key on Error : Enabled | |
| Fast Boot : Disabled | |
| Ide Block Transfer : Disabled | |
| Swap Hdd 0 and 1 : Enabled | |
| Write Protect Hdd : none | |
| Boot Sequence : C: A: | |
| Enable ROM-DOS : Enabled | |
| ROM-DOS Bootdrive : C: | |
| Flash File System : Enabled | |
| DOS/Non-DOS Flashdisk : DOS | |
| SRAM Waitstates : 2 | |
| FLASH Waitstates : 3 | |
| IO Cycle length : 980 ns | |
| HALT switches CPU clock : Enabled | |
| CPU Clock (CLK2) in MHz : 66,7 | |
| VTERM V2.3 COM1 57600 8N1 ANSI ABCD | |

8.2.1. Power on messages

Bei "disabled" werden beim Booten die Copyright-Meldungen, die Meldungen des RAM-Tests und die BIOS-Config-Box unterdrückt. Statt dessen erscheint vom BIOS nur die Meldung "Booting..." sowie ggf. vom ROMDOS "Starting ROMDOS..."

System Configuration Box: Unterdrückt bzw. aktiviert die Ausgabe der BIOS Konfigurations-Box
Display "Hit <S>..."

8.2.2. Emergency Boot Jumper J1

Die Möglichkeit, durch Kurzschließen des "Jumpers" J1 BIOS-Default-Einstellungen zu erzwingen, kann hier deaktiviert werden. Diese Option kann für Anwendungen als Schutz vor unberechtigten Manipulationen mittels des BIOS-Setups verwendet werden.

8.2.3. Wait For Key on Error

Falls eine Harddisk im Setup eingetragen ist, die nicht funktioniert oder nicht vorhanden ist, hält das BIOS beim Booten an und fordert zum Drücken einer Taste auf. Man kann dann mit "S" das Setup starten. Das Abschalten dieser Option ist nicht sinnvoll, wenn ROMDOS gebootet wird, da dieses dann mit einer Fehlermeldung den Rechner anhält.

8.2.4. Fast Boot

Das BIOS führt nur einen abgekürzten RAM-Test durch (spart Zeit beim Booten).

8.2.5. IDE-Einstellungen

Ide Block Transfer :

Swap Hdd 0 and 1 : Vertauscht die Laufwerksbuchstaben C und D (DOS) und die Laufwerksnummern 80h und 81h (für den BIOS Int 13h). Falls nur eine Festplatte (oder CompactFlash) angeschlossen ist, wird hierdurch die On-Board Flash-Disk zu Laufwerk C: (Nr. 80h) und die Festplatte zu Laufwerk D: (Nr. 81h). Falls zwei Festplatten angeschlossen sind, wird die als Slave eingestellte Festplatte zu Laufwerk C: und die als Master eingestellte zu Laufwerk D:. Die On-Board Flash-Disk ist dann Laufwerk E: (Nr. 82h). Da grundsätzlich von Laufwerk C: oder A: gebootet wird, kann daher bei zwei Festplatten nicht mehr von der On-Board Flash-Disk gebootet werden.

Write Protect Hdd : für Laufwerk C: und Laufwerk D: (oder alle beide) kann ein Schreibschutz gesetzt werden. Es sind dann keine Schreibzugriffe durch das BIOS mehr möglich (jedoch noch durch direkte Speicher-Adressierung).

Boot Sequence: mögliche Werte sind: A: ; C: ; A:, C: ; C:, A:

Dies funktioniert nur, wenn kein ROMDOS aktiv ist. Außerdem muß A: eine ROM-Disk sein, die RAM-Disk ist grundsätzlich nicht bootfähig.

8.2.6. ROMDOS Einstellungen

ROM-DOS : aktiviert bzw. deaktiviert das ROMDOS.

ROM-DOS Bootdrive : Legt das Laufwerk fest, von dem ROMDOS die Dateien command.com, config.sys und autoexec.bat liest.

8.2.7. Flash File System Einstellungen

Flash File System: schaltet das Flash File System des BIOS für die On-Board Flash-Disk ab.

Flash FS FAT monitoring : durch "FAT Monitoring" durch das BIOS wird eine erhebliche Geschwindigkeitssteigerung des Flash File Systems der On-Board Flash-Disk erzielt. Dies funktioniert jedoch nur unter DOS. Bei Einsatz eines anderen Betriebssystems muß das FAT Monitoring abgeschaltet werden.

8.2.8. SRAM und Flash Waitstates

die Default-Einstellung sollte normalerweise nicht geändert werden. Falls jedoch ein CPU-Takt unter 33 MHz verwendet wird (durch umprogrammierten Oszillator), so kann auch ein anderer Wert eingestellt werden. Dieser errechnet sich wie folgt:

$$n > t_R * f_{CPU} - 1$$

Dabei bedeutet:

t_R = Zugriffszeit des RAMs / Flashes (nach Datenblatt), n = Zahl der Waitstates

$$f_{CPU} = \text{Oszillator-Frequenz} / 2 (\leq 33 \text{ MHz})$$

Für $t_R = 55 \text{ ns}$ und $f_{CPU} = 33 \text{ MHz}$ gilt

$$t_R * f_{CPU} - 1 = 0,815$$

Der nächstgrößere ganzzahlige Wert für n ist 1. Das 55ns RAM funktioniert also bei 1 Waitstate.

Für $t_R = 70 \text{ ns}$ und $f_{CPU} = 33 \text{ MHz}$ gilt

$$t_R * f_{CPU} - 1 = 1,31$$

Für 70 ns RAM müssen daher zwei Waitstates eingestellt werden.

Für $t_R = 120 \text{ ns}$ und $f_{CPU} = 33 \text{ MHz}$ gilt

$$t_R * f_{CPU} - 1 = 2,93$$

Für 120 ns Flash müssen daher drei Waitstates eingestellt werden.

Für $t_R = 90 \text{ ns}$ und $f_{CPU} = 33 \text{ MHz}$ gilt

$$t_R * f_{CPU} - 1 = 1,97$$

Für 90 ns Flash müssen daher zwei Waitstates eingestellt werden.

8.2.9. IO Cycle length

möglich sind 170 ns, 320 ns, 560 ns und 980 ns. Der hier eingestellte Wert bestimmt die Zugriffszeiten der Lese- und Schreibzugriffe auf dem PIF-Bus.

8.2.10. HALT switches CPU clock

Bei der 386EX-Card III führt ein Halt-Befehl des 386-Prozessors normalerweise dazu, daß der Hauptoszillator abgeschaltet und der CPU-Takt auf ca. 300 kHz reduziert wird. Hierdurch sinkt der Stromverbrauch der Platine im Idle-Mode der CPU erheblich. In manchen Fällen ist diese Heruntertakten jedoch nicht erwünscht, da das Zeitverhalten der Timer sich ändert. Durch die Einstellung "Disable" läuft der Oszillator mit vollem Takt weiter.

8.3. I/O Configuration Setup

| 386Ex BIOS 1.10 - Copyright taskit Rechnertechnik GmbH | | | | | Item Specific Help | |
|--|-------|-------|----------------|-------|--------------------------|--|
| Main Advanced I/O Config Flash Exit | | | | | | |
| | COM1 | COM2 | COM3 | COM4 | | |
| Base : | 3F8 | 2F8 | none | none | Base address of COM port | |
| Baudrate : | 57600 | 57600 | 57600 | 57600 | | |
| Setting : | 8 N 1 | 8 N 1 | 8 N 1 | 8 N 1 | | |
| Interrupt : | IRQ04 | IRQ03 | none | none | | |
| Buffer : | 256 | 256 | 256 | 256 | | |
| | LPT1 | LPT2 | LPT3 | | | |
| Base : | none | none | none | | | |
| X2 Connector Configuration: | | | | | | |
| Pin 14 : | INPUT | | Timer0 Gate : | VCC | | |
| Pin 17 : | INPUT | | Timer1 Clock : | INT | | |
| Pin 20 : | IRQ5 | | Timer2 Clock : | INT | | |
| Pin 21 : | IRQ6 | | Timer2 Gate : | VCC | | |
| Pin 22 : | IRQ7 | | | | | |
| | Init1 | Init2 | Init3 | Init4 | | |
| Port : | 0000 | 0000 | 0000 | 0000 | | |
| Value : | 00 | 00 | 00 | 00 | | |
| Mask : | 00 | 00 | 00 | 00 | | |
| VTERM V2.3 COM1 57600 8N1 ANSI ABCD | | | | | | |

8.3.1. Einstellung der seriellen Schnittstellen:

Achtung: Durch Veränderung der Einstellungen für die seriellen Schnittstellen kann der Zugriff auf die 386EX-Card vollständig blockiert werden (auch versehentlich). Man sollte also genau wissen, welche Einstellung man zu welchem Zweck verändert.

Das BIOS und das DOS unterstützen maximal vier serielle Schnittstellen. Diese können vom Anwenderprogramm mit Hilfe des BIOS über den Int 14h angesprochen werden oder mit Hilfe des DOS als Geräte (Device) COM1 bis COM4. Weitere serielle Schnittstellen werden von BIOS und DOS nicht unterstützt, diese müssen durch direkten Zugriff auf die Hardware programmiert werden.

COM Ports : Mit TYPE wird der angeschlossene UART-Typ für die 4 möglichen COM Ports eingestellt. BASE gibt die Basis-Adresse des UARTs an. Mit BAUDRATE wird die Baudrate der Schnittstelle eingestellt. Mit SETTING werden Zahl der Daten-Bits, Parität und Zahl der Stop-Bits eingestellt. Unter INTERRUPT kann angegeben werden, ob die Schnittstelle im Polling-Betrieb oder per Empfangs-Interrupt arbeiten soll. In diesem Fall muß die richtige Interrupt-Leitung eingestellt werden. Verwendet man den Interrupt-Betrieb, kann unter BUFFER die Größe des vom BIOS-Interrupt-Handler verwendeten Empfangspuffers in Byte angegeben werden.

8.3.2. PRINTER PORTS

Hiermit wird die Basis-Adresse der optionalen Drucker-Schnittstellen eingestellt.

8.3.3. X2 Connector Configuration

Festlegung der Funktion einzelner Pins des I/O-Steckers (X2). Diese können als Digital I/O (Input, Open-Drain Output oder Output), als Timerfunktionen oder als Interrupts konfiguriert werden. Folgende Einstellungen sind möglich:

| Pin | Digital I/O | IRQ | Timer |
|-----|-------------|-----|----------------|
| 14: | P3.1 | - | Timer 1 Output |
| 15: | - | 9 | Timer 0 Gate |
| 16: | - | 13 | Timer 1 Clock |
| 17: | P3.0 | - | Timer 0 Output |
| 20: | P3.3 | 5 | - |
| 21: | P3.4 | 6 | - |
| 22: | P3.5 | 7 | - |

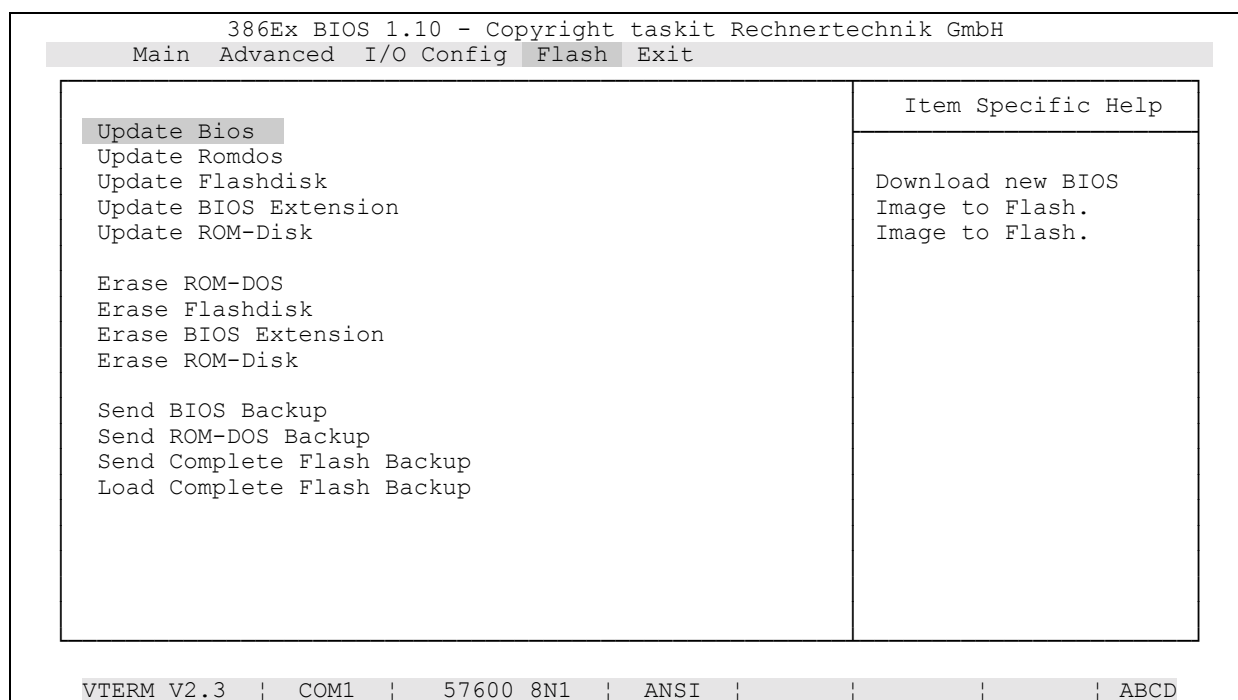
Timer Konfiguration : GATE INPUT SOURCE gibt an, wie der Gate- bzw. Clock-Anschluß eines Timers angeschlossen ist: VCC = ständig aktiv (default) oder PIN (des I/O-Steckers).

CLOCK SOURCE gibt die Quelle des Taktes für Timer an: INT = intern (default) oder PIN (extern über einen Pin des I/O-Steckers).

8.3.4. PORT Initialisierung

Hier besteht die Möglichkeit, schaltungsabhängige Initialisierungen an I/O-Ports (also auch am PIF-Bus angeschlossener Peripherie-Einheiten) sehr früh im Bootvorgang durchzuführen. Man kann maximal vier I/O-Adressen mit ihren Initialisierungswerten angeben. Die I/O Zugriffe werden wenige Mikrosekunden nach dem Reset durchgeführt.

8.4. Flash Setup



8.4.1. Flash Update Setup

Im Flash Update Setup können verschiedene Bereiche des Flash auf der 386EX-Card neu programmiert werden. Dazu wird der aktuelle Inhalt gelöscht und mit den über die serielle Schnittstelle geladenen Daten neu programmiert. Das Laden neuer Daten erfolgt mit dem Übertragungsprotokoll XMODEM. Nach Anwählen eines Menüpunktes beginnt die Karte ein Protokollzeichen (§) zu senden. Hierauf muß auf dem Terminal Programm die Übertragung mit XMODEM gestartet werden (bei VTERM: ALT-S, XMODEM, Dateiname). Nach erfolgreichem Download wird die Karte je nach gewähltem Bereich neu gebootet, oder man gelangt zurück in das Setup-Menü.

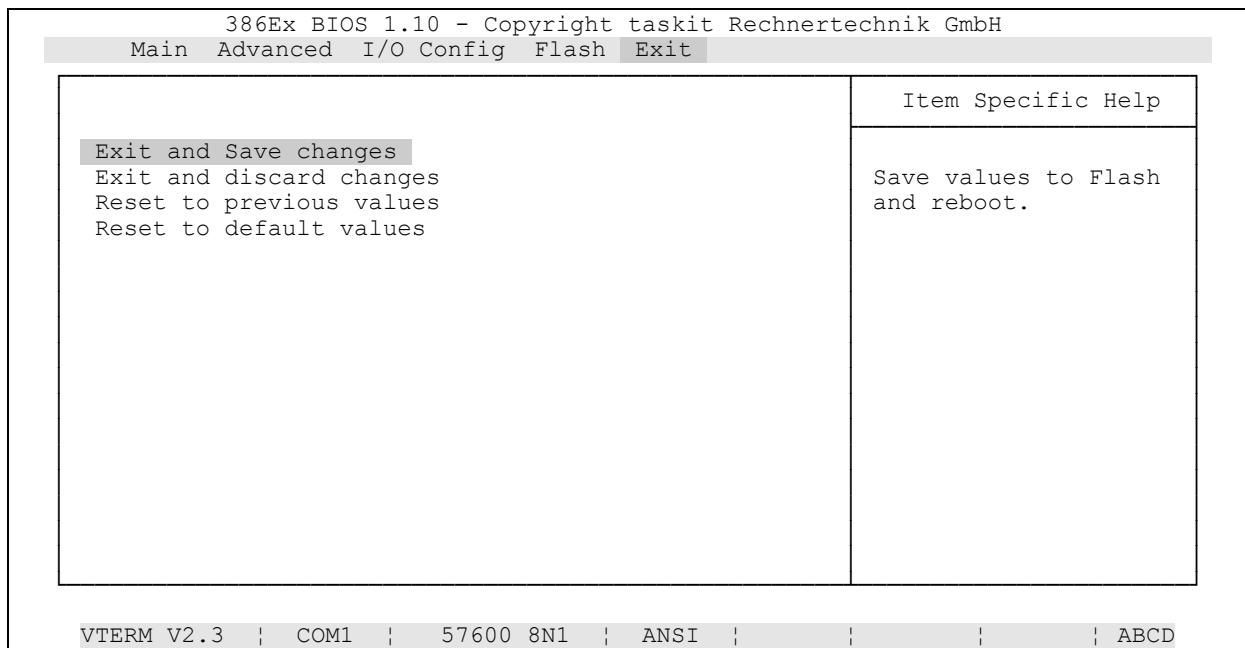
8.4.2. Flash Backup Setup

Hier können einzelne Bereiche des Flash zum PC gesendet werden. Nach Anwählen eines Menüpunktes muß auf dem Terminal-Programm der Empfang mit XMODEM gestartet werden (bei VTERM: ALT-R, XMODEM, Dateiname).

8.4.3. Flash Erase Setup

Hier können einzelne Bereiche des Flash auf der 386EX-Card selektiv gelöscht werden. Das Löschen erfolgt sektorweise (1 Flash-Sektor enthält 64kB oder 128 kB). Das Löschen eines Sektors dauert normalerweise weniger als eine Sekunde.

8.5. Exit Setup



Exit and Save changes: Setup beenden und Änderungen speichern

Exit and discard changes: Setup beenden, ohne Änderungen zu speichern

Reset to previous values: Alle seit dem Start des BIOS-Setups durchgeführten Änderungen werden verworfen.

Reset to default values: Alle Einstellungen des BIOS-Setups werden auf Standard-Werte gesetzt (Default-Einstellungen).

9. BIOS - Referenz

9.1. INT 10h - Video Service

9.1.1. INT 10h Funktion 00h - Set Video Mode

Aufruf: AH = 00h
AL = Video Modus

Rückgabe: keine

Beschreibung: Da die 386EX-Card keine Video-Hardware besitzt, dient diese Funktion nur zum Löschen des Bildschirms.

9.1.2. INT 10h Funktion 02h - Set cursor Position

Aufruf: AH = 02h
DH = Zeile
DL = Spalte

Rückgabe: keine

Beschreibung: Hiermit kann die Cursor Position verändert werden.

9.1.3. INT 10h Funktion 03h - Get current cursor position

Aufruf: AH = 03h

Rückgabe: AX = 00h
DH = Zeile
DL = Spalte

Beschreibung: Hiermit kann die aktuelle Cursor Position erfragt werden.

9.1.4. INT 10h Funktion 06h/07h - Scroll current page up/down

Aufruf: AH = 05h/06h
BH = Neues Farbattribut

Rückgabe: keine

Beschreibung: Da die 386EX-Card keine Video-Hardware besitzt, dienen diese Funktionen nur zum Löschen des Bildschirms mit dem angegebenen Farbattribut.

9.1.5. INT 10h Funktion 09h - Write Char/Attribute to Screen

Aufruf: AH = 09h
AL = Zeichen
BL = Farbattribut
CX = Anzahl der Zeichen

Rückgabe: keine

Beschreibung: Das Zeichen wird CX mal mit dem angegebenen Farbattribut ausgegeben. Die Cursor Position wird nicht verändert. Anders als auf einem PC gilt das Farbattribut für alle folgenden Ausgaben mit den Funktionen 00h, 0Ah und 0Eh.

9.1.6. INT 10h Funktion 0Ah - Write character to screen

Aufruf: AH = 0Ah
AL = Zeichen
CX = Anzahl der Zeichen

Rückgabe: keine

Beschreibung: Das Zeichen wird CX mal ausgegeben. Die Cursor Position wird nicht verändert.

9.1.7. INT 10h Funktion 0Eh - Write Teletype to screen

Aufruf: AH = 0Eh
AL = Zeichen

Rückgabe: keine

Beschreibung: Das Zeichen in AL wird ausgegeben, wobei die Steuerzeichen 07h (Beep), 08h (Backspace), 0Ah (Linefeed) und 0Dh (Carriage Return) interpretiert werden. Dies ist die schnellste Ausgabemöglichkeit, da keine Escape Sequenzen gesendet werden müssen.

9.2. INT 11h - Equipment Check Service

Aufruf: keine

Rückgabe: AX = Inhalt von 40:10

| | |
|--------------|-----------------------|
| Bits 15 - 14 | = Anzahl der Drucker |
| Bits 13 - 12 | = Reserviert |
| Bits 11 - 9 | = Anzahl Disketten |
| Bit 8 | = Reserviert |
| Bits 5 - 4 | = Video Modus |
| Bit 3 | = Reserviert |
| Bit 2 | = Maus installiert |
| Bit 1 | = Coprozessor |
| Bit 0 | = Boot Disk vorhanden |

Beschreibung: Diese Funktion gibt den Inhalt der Speicherzelle 40:10h zurück.

9.3. INT 12h - Memory size

Aufruf: keine

Rückgabe: AX = Inhalt von 40:13h

Beschreibung: Diese Funktion gibt den Inhalt der Speicherzelle 40:13h zurück. Dies gibt den freien Speicher in Kilobytes an.

9.4. INT 13h - Diskette Services

Da die Flash-Disk der 386EX-Card wie ein Disketten-Laufwerk organisiert ist, gilt das folgende auch für die Flash-Disk.

9.4.1. INT 13h Funktion 01h - Read Disk Status

Aufruf: AH = 01h
DL = Laufwerk (0 oder 1)

Rückgabe: AH = 0 Kein Fehler
= sonst Fehler Code
CF = 0 Kein Fehler
= 1 Fehler

Beschreibung: Dies liest den letzten Fehlercode aus und setzt ihn wieder zurück.

9.4.2. INT 13h Funktion 02h - Read Disk Sectors

Aufruf: AH = 02h
AL = Anzahl der Sektoren
CH = Track
CL = Sector
DH = Kopf
DL = Laufwerk (0 oder 1)
ES:BX = Zeiger zum Sektorbuffer

Rückgabe: AH = 0 Kein Fehler
= sonst Fehler Code
AL = Anzahl der gelesenen Sektoren
CF = 0 Kein Fehler
= 1 Fehler

Beschreibung: Diese Funktion liest die angegebene Anzahl der Sektoren in einen Puffer ein.

9.4.3. INT 13h Funktion 03h - Write Disk Sectors

Aufruf: AH = 03h
AL = Anzahl der Sektoren
CH = Track
CL = Sector
DH = Kopf
DL = Laufwerk (0 oder 1)
ES:BX = Zeiger zum Sektorbuffer

Rückgabe: AH = 0 Kein Fehler
= sonst Fehler Code
AL = Anzahl der gelesenen Sektoren
CF = 0 Kein Fehler
= 1 Fehler

Beschreibung: Diese Funktion schreibt die angegebene Anzahl der Sektoren auf das Laufwerk. Diese Funktion ist nur für die RAM-Disk gültig.

9.4.4. INT 13h Funktion 08h - Read Drive Parameter

| | | |
|------------------|-------|--|
| Aufruf: | AH | = 08h |
| | DL | = Laufwerk (0 oder 1) |
| Rückgabe: | AX | = 0 |
| | CH | = Letzter Track |
| | CL | = Letzter Sektor |
| | DH | = Anzahl der Köpfe |
| | DL | = Anzahl der installierten Laufwerke |
| | ES:DI | = Pointer auf Diskette Parameter Table |
| | CF | = 0 Kein Fehler = 1 Fehler |

Beschreibung: Diese Funktion liefert die Parameter eines Laufwerkes.

9.5. INT 14h – Funktionen der asynchronen seriellen Schnittstellen**9.5.1. INT 14h Funktion 00h – Serielle Schnittstelle initialisieren**

| | | |
|------------------|------------|---|
| Aufruf: | AH | = 00h |
| | AL | = Parameter |
| | Bits 7 - 5 | = Baudrate |
| | | 000 - 110 Baud |
| | | 001 - 150 Baud |
| | | 010 - 300 Baud |
| | | 011 - 600 Baud |
| | | 100 - 1200 Baud |
| | | 101 - 2400 Baud |
| | | 110 - 4800 Baud |
| | | 111 - 9600 Baud |
| | Bits 4 - 3 | = Parity |
| | | x0 - keine |
| | | 01 - ungerade |
| | | 11 - gerade |
| | Bit 2 | = Stop Bits |
| | | 0 - 1 Stop Bit |
| | | 1 - 2 Stop Bits |
| | Bit 1 - 0 | = Datenwortlänge |
| | | 00 - 5 Bits |
| | | 01 - 6 Bits |
| | | 10 - 7 Bits |
| | | 11 - 8 Bits |
| | DX | = Nr. der seriellen Schnittstelle (0 - 3) |
| Rückgabe: | AH | = Line Status wie bei Funktion 1 |
| | AL | = Modem Status |

9.5.2. INT 14h Funktion 01h – Zeichen senden

Aufruf: AH = 01h
AL = Zu sendendes Zeichen
DX = Nr. der seriellen Schnittstelle (0 - 3)

Rückgabe: AL = gesendetes Zeichen
AH = Line Status:
Bit 7 = 1 Timeout error
Bit 6 = 1 Transmitter shift register empty
Bit 5 = 1 Transmitter buffer register empty
Bit 4 = 1 Break condition = RXD low für die Dauer von mehr
als einer Wortlänge
Bit 3 = 1 Framing error = ungültiges Stop-Bit
Bit 2 = 1 Parity error
Bit 1 = 1 Overrun error = Receive Buffer wurde überschrieben
Bit 0 = 1 Receive data ready

9.5.3. INT 14h Funktion 02h – Zeichen empfangen

Aufruf: AH = 02h
DX = Nr. der seriellen Schnittstelle (0 - 3)

Rückgabe: AL = Empfangenes Zeichen
AH = Line Status wie bei Funktion 1

Bemerkung: Ein Timeout entsteht nach ca. 1 Sekunde. Die seriellen Schnittstellen werden je nach Einstellung im BIOS-Setup (IRQn oder Polled) mit Interrupt oder im Polling-Modus betrieben.

9.5.4. INT 14h Funktion 03h – Status einer seriellen Schnittstelle abfragen

Aufruf: AH = 03h
DX = Nr. der seriellen Schnittstelle (0 - 3)

Rückgabe: AH = Line Status wie bei Funktion 1

9.5.5. INT 14h Funktion 04h - Extended Init

| | | | |
|------------------|----|---|--|
| Aufruf: | AH | = 04h | |
| | AL | = 1 | = Break (TXD low setzen) |
| | | = 0 | = kein Break |
| | BH | = Parity | |
| | | 00h | = no parity. |
| | | 01h | = odd parity. |
| | | 02h | = even parity. |
| | BL | - Stop bits | |
| | | 00h | = 1 Stop Bit |
| | | 01h | = 2 Stop Bits oder 1,5 bei 5 Datenbits |
| | CH | - Data length | |
| | | 00h | = 5 Bits. |
| | | 01h | = 6 Bits. |
| | | 02h | = 7 Bits. |
| | | 03h | = 8 Bits. |
| | CL | - Baudrate | |
| | | 00h | = 110 Baud |
| | | 01h | = 150 Baud |
| | | 02h | = 300 Baud |
| | | 03h | = 600 Baud |
| | | 04h | = 1200 Baud |
| | | 05h | = 2400 Baud |
| | | 06h | = 4800 Baud |
| | | 07h | = 9600 Baud |
| | | 08h | = 19200 Baud |
| | | 09h | = 38400 Baud |
| | | 0Ah | = 57600 Baud |
| | | 0Bh | = 115200 Baud |
| | DX | = Nr. der seriellen Schnittstelle (0 - 3) | |
| Rückgabe: | AH | = Line Status wie bei Funktion 1 | |
| | AL | = Modem Status | |

Bemerkung: Die Funktion erlaubt höhere Baudraten als Funktion 00h.

9.6. INT 15h - System Services

9.6.1. INT 15h Funktion 24h - A20 Gate-Control

9.6.2. INT 15h Funktion 87h - Move Memory Block

9.6.3. INT 15h Funktion C0h - Get System Config Table

Aufruf: AH = C0h
Rückgabe: AH = 00h
ES:BX = Adresse System Config Table

Beschreibung: Diese Funktion liefert die Adresse der System Configuration Table.

9.6.4. INT 15h Funktion A1h - Int 10h / Int 16h I/O umleiten

Die Funktion leitet die Video-Ausgabe des BIOS Int 10h und die Tastatur-Eingabe-Funktionen des Int 16h auf eine serielle Schnittstelle um. Der Parameter 0 (none) schaltet sämtliche Ein- und Ausgaben des Int 10h und Int 16h ab.

Aufruf: BX = COM-Port (1 = COM1, ..., 4 = COM4, 0 = none)

9.7. INT 15h Funktion C3h - 386EX-Card spezifische Funktionen

9.7.1. INT 15h Funktion C300h - Watchdogfunktion deaktivieren

Aufruf: AH = C3h
AL = 00h

Rückgabe: --

Beschreibung: Nach Aufruf dieser Funktion ist die Watchdogfunktion inaktiv.

9.7.2. INT 15h Funktion C301h - Watchdog freigeben

Aufruf: AH = C3h
AL = 01h

Rückgabe: --

Beschreibung: Nach Aufruf dieser Funktion muß der Watchdog mindestens einmal in 1,6 Sekunden zurückgesetzt werden, andernfalls wird ein Reset ausgelöst.

9.7.3. INT 15h Funktion C302h - Watchdog zurücksetzen

Aufruf: AH = C3h
AL = 02h

Rückgabe: --

Beschreibung: Der Aufruf dieser Funktion mindestens alle 1,6 Sekunden verhindert den Reset der 386EX-Card durch den Watchdog.

9.7.4. INT 15h Funktion C303h - NMI sperren

Aufruf: AH = C3h
AL = 03h

Rückgabe: --

9.7.5. INT 15h Funktion C304h – NMI freigeben

Aufruf: AH = C3h
AL = 04h

Rückgabe: --

Bemerkung: Ein NMI wird vom Power-Fail-Komparator des Supervisor-Chips erzeugt, wenn die Spannung Vunst am X2-Stecker unter einen bestimmten Wert sinkt. Siehe das Kapitel über den Supervisor-IC.

9.7.6. INT 15h Funktion C310h – Prozessortakt abfragen

Aufruf: AH = C3h
AL = 10h

Rückgabe: AL = 3 : CPU-Clock = 33,33 MHz (CLK2 = 66,67 MHz)
= 2 : CPU-Clock = 16,67 MHz (CLK2 = 33,33 MHz)
= 1 : CPU-Clock = 8,33 MHz (CLK2 = 16,67 MHz)
= 0 : CPU-Clock = 922 kHz (CLK2 = 1,8432 MHz)

9.7.7. INT 15h Funktion C311h – Prozessortakt setzen

Aufruf: AH = C3h
AL = 11h
BL = Takt wie bei Funktion C310h

Rückgabe: --

Bemerkung : Der Vorteiler (CLKPRS) für den Eingangstakt von Timer 0 wird bei der Taktumschaltung angepaßt.

9.7.8. INT 15h Funktion C312h - CPU in den IDLE Mode versetzen

Aufruf: AH = C3h
AL = 12h
BL = *Takt*

Bemerkung : Interne Peripherie des 386EX (insbesondere Timer und UARTs) läuft weiter. Rückkehr aus dem Idle-Modus nur durch Hardware Interrupt.

9.7.9. INT 15h Funktion C313h - CPU in den Stop Mode versetzen

Aufruf: AH = C3h
AL = 13h

Bemerkung: Interne Peripherie des 386EX wird gestoppt. Rückkehr aus Stop-Modus nur durch externen Hardware Interrupt, also entweder durch die RTC oder durch über PIF-Bus oder I/O-Stecker an die 386EX-Card angeschlossene zusätzliche Peripherie.

9.7.10. INT 15h Funktion C314h – synchrone serielle Schnittstelle: Senden

Aufruf: AH = C3h
AL = 14h
BX = Datenwort

9.7.11. INT 15h Funktion C315h – synchrone serielle Schnittstelle: Empfangen

Aufruf: AH = C3h
AL = 15h

Rückgabe: AX = Datenwort

9.7.12. INT 15h Funktion C320h - EEPROM auslesen

Aufruf: AH = C3h
AL = 20h
BH = Adresse im EEPROM

Rückgabe: AL = Datenbyte
Beschreibung: Diese Funktion liest das Datenbyte im EEPROM an der übergebenen Adresse

9.7.13. INT 15h Funktion C321h - EEPROM beschreiben

Aufruf: AH = C3h
AL = 21h
BH = Adresse im EEPROM
BL = Datenbyte
Beschreibung: Diese Funktion schreibt das Datenbyte im EEPROM an der übergebenen Adresse

9.7.14. INT 15h Funktion C322h - I2C-Bus Datenbyte lesen

Aufruf: AH = C3h
AL = 22h
BH = I2C-Bus Adresse
CH = I2C-Chip Adresse
Rückgabe : AL = gelesenes I2C-Bus Datenbyte
Carry-Flag = 0: kein Fehler
Carry-Flag = 1: Fehler

9.7.15. INT 15h Funktion C323h - I2C-Bus Datenbyte schreiben

Aufruf: AH = C3h
AL = 23h
BH = I2C-Bus Adresse
BL = I2C-Bus Datenbyte
CH = I2C-Chip Adresse
Rückgabe : Carry-Flag = 0: kein Fehler
Carry-Flag = 1: Fehler

9.7.16. INT 15h Funktion C324h - I2C-Bus Datenblock lesen

Aufruf: AH = C3h
AL = 24h
BH = I2C-Bus Adresse
CH = I2C-Chip Adresse
CL = Anzahl zu lesender Bytes (1...256)
ES:DI = Far-Zeiger auf Lese-Puffer im RAM
Rückgabe : ES:[DI] = gelesene Datenbytes
Carry-Flag = 0: kein Fehler
Carry-Flag = 1: Fehler

Bemerkung: Der Wert CL = 0 wird als 256 interpretiert.

9.7.17. INT 15h Funktion C325h - I2C-Bus Datenblock schreiben

Aufruf: AH = C3h
AL = 25h
BH = I2C-Bus Adresse
CH = I2C-Chip Adresse
CL = Anzahl zu schreibender Bytes (0...255)
ES:SI = Far-Zeiger auf zu schreibende Daten
Rückgabe : Carry-Flag = 0: kein Fehler
Carry-Flag = 1: Fehler

Bemerkung: Falls CL = 0, so wird nur die Chip-Adresse und die Bus-Adresse ausgegeben. Die Bus-Adresse kann auch universell als Parameter-Byte interpretiert werden.

9.7.18. INT 15h Funktion C326h - I2C-Bus anfordern

Aufruf: AH = C3h
AL = 26h

Rückgabe : Carry-Flag = 0: I2C-Bus erfolgreich angefordert
Carry-Flag = 1: I2C-Bus war besetzt

Beschreibung: Da I2C-Bus Zyklen sich bei der 386EX-Card nicht überlagern dürfen, sollte der Bus vor jeder Übertragung angefordert werden. Das zugehörige I2C-Bus Flag wird dadurch gesetzt und verhindert, daß weitere I2C-Bus Funktionen die aktuelle Funktion unterbrechen. Dies ist nur von Bedeutung, wenn I2C-Funktionen innerhalb von Interrupt-Routinen ausgeführt werden sollen (z.B. durch den BIOS-Interrupthandler für die RTC).

9.7.19. INT 15h Funktion C327h - I2C-Bus freigeben

Aufruf: AH = C3h
AL = 27h

Beschreibung: Das I2C-Bus Flag wird wieder zurückgesetzt. Falls der I2C-Bus zuvor mit der Funktion C326h angefordert wurde, muß er mit dieser Funktion zurückgesetzt werden. Andernfalls sind keine weiteren I2C-Bus Zugriffe mehr möglich.

9.7.20. INT 15h Funktion C328h - I2C-Bus Datenbyte lesen mit 2 Adreßbytes

Aufruf: AH = C3h
AL = 28h
BH = I2C-Bus Adresse 0
CH = I2C-Chip Adresse
CL = I2C-Bus Adresse 1

Rückgabe : AL = gelesenes I2C-Bus Datenbyte
Carry-Flag = 0: kein Fehler
Carry-Flag = 1: Fehler

9.7.21. INT 15h Funktion C329h - I2C-Bus Datenbyte schreiben mit 2 Adreßbytes

Aufruf: AH = C3h
AL = 29h
BH = I2C-Bus Adresse 0
BL = I2C-Bus Datenbyte
CH = I2C-Chip Adresse
CL = I2C-Bus Adresse 1

Rückgabe : Carry-Flag = 0: kein Fehler
Carry-Flag = 1: Fehler

9.7.22. INT 15h Funktion C330h – Hardware Serien-Nummer abfragen

Aufruf: AH = C3h
AL = 30h

Rückgabe: AX = unteres Datenwort der Serien-Nummer
BX = mittleres Datenwort der Serien-Nummer
CX = oberes Datenwort der Serien-Nummer

Beschreibung: Diese Funktion liest die sechs Bytes des Serien-Nummer-ICs der 386EX-Card aus. Jedes Exemplar des Serien-Nummer-ICs ist vom Hersteller mit einer eigenen, von allen anderen Exemplaren verschiedenen Serien-Nummer fest programmiert.

9.8. INT 16h - Keyboard Service

9.8.1. INT 16h Funktion 00h - Read Keyboard Input

Aufruf: AH = 00h

Rückgabe: AH = Scancode erweiterte Tasten
AL = Tastenwert

Beschreibung: Diese Funktion liest eine Taste ein. Nur einige erweiterte Tasten werden unterstützt, da hierzu ANSI Escape Sequenzen verwendet werden (Siehe Anhang).

9.8.2. INT 16h Funktion 01h - Read Keyboard Status

Aufruf: AH = 01h

Rückgabe: ZF = 1 - Kein Zeichen vorhanden
= 0 - Zeichen vorhanden

Beschreibung: Hiermit wird ermittelt, ob ein Zeichen im Tastaturpuffer vorliegt. Anders als beim PC wird das Zeichen nicht mit zurückgeliefert.

9.8.3. INT 16h Funktion 05h – Tastendruck simulieren

Aufruf: AH = 05h
CH = Scan-Code der Taste
CL = ASCII-Code der Taste

Rückgabe: AL = 0 – Funktion erfolgreich
= 1 – Tastaturpuffer war voll

Beschreibung: Mit dieser Funktion kann ein Anwendungsprogramm Werte in den BIOS-Tastatur-Puffer schreiben.

9.9. INT 17h - Parallel Service

9.9.1. INT 17h Funktion 00h - Print Character

| | | |
|------------------|------------------|---------------------------|
| Aufruf: | AH | = 00h |
| | AL | = Zeichen |
| | DX | = LPT Port (0 - 2) |
| Rückgabe: | AH | = Drucker Status |
| | Bit 7 | = 1 Drucker nicht besetzt |
| | Bit 6 | = 1 Acknowledgment |
| | Bit 5 | = 1 Out of Paper |
| | Bit 4 | = 1 Drucker selektiert |
| | Bit 3 | = 1 Drucker Fehler |
| | Bit 2 - 1 | = reserviert |
| Bit 0 | = Timeout Fehler | |

Beschreibung: Mit dieser Funktion wird ein Zeichen ausgegeben.

9.9.2. INT 17h Funktion 01h - Initialize Printer

| | | |
|------------------|------------------|---------------------------|
| Aufruf: | AH | = 01h |
| | DX | = LPT Port (0 - 2) |
| Rückgabe: | AH | = Drucker Status |
| | Bit 7 | = 1 Drucker nicht besetzt |
| | Bit 6 | = 1 Acknowledgment |
| | Bit 5 | = 1 Out of Paper |
| | Bit 4 | = 1 Drucker selektiert |
| | Bit 3 | = 1 Drucker Fehler |
| | Bit 2 - 1 | = reserviert |
| Bit 0 | = Timeout Fehler | |

Beschreibung: Hiermit wird der Drucker zurückgesetzt.

9.9.3. INT 17h Funktion 02h - Get Printer Status

| | | |
|------------------|------------------|---------------------------|
| Aufruf: | AH | = 02h |
| | DX | = LPT Port (0 - 2) |
| Rückgabe: | AH | = Drucker Status |
| | Bit 7 | = 1 Drucker nicht besetzt |
| | Bit 6 | = 1 Acknowledgment |
| | Bit 5 | = 1 Out of Paper |
| | Bit 4 | = 1 Drucker selektiert |
| | Bit 3 | = 1 Drucker Fehler |
| | Bit 2 - 1 | = reserviert |
| Bit 0 | = Timeout Fehler | |

Beschreibung: Hiermit wird der Status des Druckers ausgelesen.

9.10. INT 18h - Boot Failure

Beschreibung: Diese Funktion wird nach erfolglosen Bootversuchen angesprungen.

9.11. INT 19h - Boot System

Beschreibung: Diese Funktion wird nach vollständiger Initialisierung des BIOS angesprungen. Sie versucht von Diskette zu booten und das Betriebssystem zu starten. Mißlingt dies, wird ein INT 18h ausgeführt.

9.12. INT 1Ah - Uhren- und Timer-Funktionen

9.12.1. INT 1Ah Funktion 00h - Read System timer

Aufruf: AH = 00h

Rückgabe: AH = 00h
AL = 24h Überlauf Flag
CX:DX = System Ticks seit Mitternacht

Beschreibung: Diese Funktion liest den "System Timer" aus. Dies ist eine 4-Byte BIOS-Variable bei Adresse 40h:6Ch. Diese wird vom Timer0-Interrupt 18,2 mal in der Sekunde erhöht. Um Mitternacht wird sie auf 0 gesetzt und das 24-Stunden-Flag (1Byte Variable bei 40h:70h) um 1 erhöht. Das 24-Stunden-Flag wird bei jedem Auslesen des System-Timers mit Int 1Ah Funktion 0 wieder auf 0 gesetzt.

9.12.2. INT 1Ah Funktion 01h - Set System timer

Aufruf: AH = 01h
CX:DX = System Ticks seit Mitternacht

Rückgabe: AH = 00h

Beschreibung: Diese Funktion setzt den System Timer. Dieser wird vom Timer 0-Interrupt 18,2 mal in der Sekunde erhöht.

9.12.3. INT 1Ah Funktion 02h - Read Real Time Clock

Aufruf: AH = 02h

Rückgabe: AH = 00h
AL = Stunden BCD
CH = Stunden in BCD
CL = Minuten in BCD
DH = Sekunden in BCD

Beschreibung: Diese Funktion liest die Uhrzeit der RTC aus.

9.12.4. INT 1Ah Funktion 03h - Set Real Time Clock

Aufruf: AH = 03h
AL = Stunden in BCD
CH = Stunden in BCD
CL = Minuten in BCD
DH = Sekunden in BCD

Rückgabe: AH = 00h

Beschreibung: Diese Funktion setzt die Uhrzeit der RTC.

9.12.5. INT 1Ah Funktion 04h – Read RTC Date

Aufruf: AH=04h

Rückgabe: CH =Jahrhundert (19 oder 20)
CL =Jahr
DH =Monat
DL =Tag

Beschreibung: Diese Funktion liest das Datum der RTC aus.

9.12.6. INT 1Ah Funktion 05h – Set RTC Date

Aufruf: AH =05h
CH =Jahrhundert (19 oder 20)
CL =Jahr
DH =Monat
DL =Tag

Rückgabe: keine

Beschreibung: Diese Funktion setzt das Datum der RTC.

9.12.7. INT 1Ah Funktion 06h – Set / Enable RTC Interrupt

Aufruf: AH =06h
 CH =Stunde
 CL =Minute
 DH =Sekunde

Rückgabe: Carry-Flag = O.K.
 Carry-Flag = 1: es ist schon ein Interrupt programmiert

Beschreibung: Die Echtzeituhr erzeugt zur programmierten Zeit desselben Tages einen Interrupt. Das Anwenderprogramm kann eine Funktion in den Interrupt 4Ah einklinken, die darauf bei jedem RTC-Interrupt aufgerufen wird. Wird dieser Interrupt verwendet, um aus dem STOP-Modus herauszukommen, ist ein Einklinken in den Interrupt 4Ah nicht nötig. Alle Werte, die der Funktion übergeben werden, müssen BCD kodiert sein. Es ist immer nur eine einzige Interrupt-Zeit aktiv. Ist bereits ein Interrupt programmiert, muß dieser zuerst mit Hilfe der Funktion 07h gelöscht werden.

9.12.8. INT 1Ah Funktion 07h – Disable RTC Interrupt

Aufruf: AH =07h

Rückgabe: keine

Beschreibung: Mit Hilfe dieser Funktion kann eine einprogrammierte Interrupt-Zeit wieder gelöscht werden. Der RTC-Interrupt wird dann nicht mehr erzeugt. Diese Funktion muß auch immer dann aufgerufen werden, wenn die Interrupt-Zeit geändert werden soll. Erst nach Aufruf dieser Funktion kann mit der Funktion 06h eine neue Zeit programmiert werden.

9.12.9. INT 1Ah Funktion 08h : Synchronize System Timer

Aufruf: AH - 08h

Rückgabe: keine

Beschreibung: Der System-Timer wird mit dem Inhalt der Echtzeituhr synchronisiert.

9.12.10. INT 1Ah Funktion 09h : Setze zyklischen RTC-Interrupt

Aufruf: AH = 09h
 AL = 0: Interrupt jede 1/100 Sekunde
 = 1: Interrupt jede Sekunde
 = 2: Interrupt jede Minute
 = 3: Interrupt jede Stunde
 = 4: Interrupt jeden Tag
 CL = 1..99 oder 1..59 oder 1..23: Vielfaches des Grundintervalls in AL, falls Bit4 von AL = 1

Rückgabe: keine

Bemerkung : Für das Aktivieren und Deaktivieren des zyklischen RTC-Interrupts gelten die gleichen Bemerkungen wie bei Funktion 7. Der Interrupt alle 1/100 Sekunde ist bei der RTC PCF8593, die bei der 386EX-Card III zum Einsatz kommt, nicht sehr genau, da die RTC 1/64s und 1/128s Zyklen verwendet, um im Mittel auf 1/100s zu kommen. Erst der 1/4s-Interrupt ist genau.

9.13. INT 1Bh bis 1Fh

Diese Interruptvektoren zeigen nicht auf eine ausführbare Funktion, sondern auf verschiedene BIOS-Tabellen.

9.14. INT 5Fh - Flash Services

9.14.1. INT 5Fh Funktion 00h - Flash Erase Block

Aufruf : AH = 00h
 DX:DI = 32-Bit Block-Startadresse
 = 32-Bit Flash-Startadresse + 20000h * Block-Nr.
 oder = 32-Bit Flash-Startadresse + 10000h * Block-Nr.

abhängig davon, ob ein oder zwei Flash-ICs bestückt sind.

Rückgabe : Carry-Flag = 0: kein Fehler
 Carry-Flag = 1: Fehler

Beschreibung: (Siehe auch Funktion 02h). Löschen eines Flash-Blocks von 128 kB oder 64 kB.

9.14.2. INT 5Fh Funktion 01h - Flash Read Block

Aufruf: AH = 01h
 DX:DI = 32-Bit Quell-Adresse (erstes zu lesendes Byte)
 = Flash-Startadresse + 20000h*Block-Nr. + Offset
 ES:BX = Ziel-Adresse
 CX = Zahl der zu lesenden Bytes

Rückgabe : Carry-Flag = 0: kein Fehler
 Carry-Flag = 1: Fehler

Beschreibung: "Offset" gibt die Startadresse relativ zum Blockanfang an. Die Quell-Adresse ist eine 32-Bit-Adresse, da der Flash-Speicher im Protected-Mode angesprochen wird. Die Ziel-Adresse (im RAM) ist eine Real-Mode Adresse, also in der Form *Segment:Offset*. Flash-Start hängt von der Bestückung ab: 1 MB: 3F00000h, 2 MB: 3E00000h, 4 MB: 3C00000h, 8 MB: 3800000h. Wenn nur ein Flash-Baustein bestückt ist, muß bei der Quell-Adresse 10000h statt 20000h angegeben werden, da die Blöcke dann nur 64 kB groß sind.

9.14.3. INT 5Fh Funktion 02h - Flash Write Block

Aufruf: AH = 02h
 DX:DI = 32-Bit Ziel-Adresse
 ES:BX = Quell-Adresse
 CX = Zahl der zu schreibenden Bytes

Rückgabe: Carry-Flag = 0: kein Fehler
 Carry-Flag = 1: Fehler

Beschreibung: (Siehe auch Funktion 02h). Die Funktion führt keine Lösch-Operation auf den Block aus. Falls der zu schreibende Bereich nicht gelöscht ist, kehrt die Funktion mit Fehler zurück.

9.14.4. INT 5Fh Funktion 03h - Flash Erase and Write Block

Aufruf: AH = 03h
 DX:DI = 32-Bit Ziel-Adresse
 ES:BX = Quell-Adresse
 CX = Zahl der zu schreibenden Bytes

Rückgabe: Carry-Flag = 0: kein Fehler
 Carry-Flag = 1: Fehler

Beschreibung: (Siehe auch Funktion 02h). Der betreffende Flash-Block wird vor dem Schreiben gelöscht.

9.14.5. INT 5Fh Funktion 04h - Read Flash Chip and Manufacturer ID

Aufruf: AH = 04h
DX:DI = 32-Bit Quell-Adresse = 03F0:0000
ES:BX = Ziel-Adresse

Rückgabe : Carry-Flag = 0: kein Fehler
Carry-Flag = 1: Fehler

Beschreibung: Liest den ID-Code der Flash-ICs aus.
29F032: 41
29F016: AD
29F080: D5
29F800T: 22D6
29F160T: 22D2
Manufacturer: AMD: 01, Fujitsu: 04

10. Tabellen

10.1. I/O-Adressen

| Port Adresse | Funktion |
|----------------------|---|
| 000h – 00Fh | DMA Controller |
| 020h – 021h | 1. Interrupt Controller (Master) |
| 022h – 023h | 386EX Control Register |
| 040h – 043h | Timer 0, 1 und 2 |
| 080h – 083h | DMA Pageregister |
| 092h | A20-Gate, CPU Reset |
| 0A0 – 0A1 | 2. Interrupt Controller (Slave) |
| 1F0h - 1FFh | IDE, CompactFlash |
| 2F8h – 2FFh | zweite serielle Schnittstelle (COM2) |
| 300h – 30Fh | PIF-Bus CS0 |
| 310h – 31Fh | PIF-Bus CS1 |
| 320h – 32Fh | PIF-Bus CS2 |
| 330h – 33Fh | PIF-Bus CS3 |
| 340h – 34Fh | Ethernet Controller (Crystal CS8900A) |
| 3B0h | Konfigurationsregister (Takt, NMI, Ethernet Sleep; wird vom BIOS verwaltet) |
| 3F6h – 3F7h | IDE, CompactFlash |
| 3F8h – 3FFh | erste serielle Schnittstelle (COM1) |
| F000h – FFFFh | 386EX Peripherie: |
| F480h – F48Bh | synchrone serielle Schnittstelle |
| F860 | I/O-Port P1 Input |
| F862 | I/O-Port P1 Output |
| F864 | I/O-Port P1 Direction |
| F868 | I/O-Port P2 Input |
| F86A | I/O-Port P2 Output |
| F86C | I/O-Port P2 Direction |
| F870 | I/O-Port P3 Input |
| F872 | I/O-Port P3 Output |
| F874 | I/O-Port P3 Direction |

10.2. Interrupt-Tabelle

| Vektor | Adresse | Verwendung | Service-Routine durch | Typ |
|---------|-----------|---|-----------------------|---------------|
| 00 | 000 | Divide by Zero | BIOS oder Anwendung | CPU-Exception |
| 01 | 004 | Single Step Debug Interrupt | Debugger | CPU-Exception |
| 02 | 008 | Non-Maskable Interrupt (NMI) | Anwendung | Hardware-NMI |
| 03 | 00C | One-Byte Debug Breakpoint | Debugger | CPU-Befehl |
| 04 | 010 | Overflow (aufgerufen durch INTO Befehl) | Anwendung | CPU-Exception |
| 05 | 014 | Array Bounds Check (aufgerufen durch BOUNDS Befehl) | Anwendung | CPU-Exception |
| 06 | 018 | Ungültiger Befehls-Code | Debugger | CPU-Exception |
| 07 | 00C | Coprozessor nicht vorhanden | Anwendung | CPU-Exception |
| 08 | 020 | Timer 0 (System-Timer) | BIOS | Hardware-IRQ0 |
| 09 | 024 | PIF-Bus IRQ (PIF-INT) | Anwendung | Hardware-IRQ1 |
| 0A | 028 | reserviert | BIOS | Software |
| 0B | 02C | serielle Schnittstelle (COM2) | BIOS oder Anwendung | Hardware-IRQ3 |
| 0C | 030 | serielle Schnittstelle (COM1) | BIOS oder Anwendung | Hardware-IRQ4 |
| 0D | 034 | frei für Anwendung | Anwendung | Hardware-IRQ5 |
| 0E | 038 | frei für Anwendung | Anwendung | Hardware-IRQ6 |
| 0F | 03C | frei für Anwendung | Anwendung | Hardware-IRQ7 |
| 10 | 040 | Video Funktionen | BIOS | Software |
| 11 | 044 | System Konfiguration (Equipment Check) | BIOS | Software |
| 12 | 048 | RAM Größe | BIOS | Software |
| 13 | 04C | Disk Funktionen | BIOS | Software |
| 14 | 050 | Serial Port Funktionen (COM1-4) | BIOS | Software |
| 15 | 054 | diverse, u.a. 386EX-Card-spezifische Funktionen | BIOS | Software |
| 16 | 058 | Keyboard Funktionen | BIOS | Software |
| 17 | 05C | Parallel Port Funktionen (LPT) | BIOS | Software |
| 18 | 060 | Boot Fehler | BIOS | Software |
| 19 | 064 | Boot Loader | BIOS oder DOS | Software |
| 1A | 068 | System-Timer- und RTC-Funktionen | BIOS | Software |
| 1B | 06C | reserviert | BIOS | Software |
| 1C | 070 | Timer User Funktion (aufgerufen von Int 08h) | Anwendung | Software |
| 1D | 074 | reserviert | BIOS | Software |
| 1E | 078 | Disk Parameter Tabelle | -- | BIOS-Tabelle |
| 1F | 07C | reserviert | BIOS | Software |
| 20 – 3F | 080 – 0FC | reserviert für DOS | DOS | Software |
| 40 – 49 | 100 – 124 | reserviert für BIOS | BIOS | Software |
| 4A | 128 | RTC User Funktion (aufgerufen von Int 70h) | Anwendung | Software |
| 4B – 5E | 12C – 178 | reserviert für BIOS | BIOS | Software |
| 5F | 17C | Flash Funktionen | Anwendung | Software |
| 60 – 6F | 180 – 1BC | frei für Anwendung | Anwendung | Software |

Fortsetzung Interrupt-Tabelle

| Vektor | Adresse | Verwendung | Service-Routine durch | Typ |
|---------|-----------|---|--------------------------------------|----------------|
| 70 | 1C0 | Real time Clock (RTC) Alarm | BIOS | Hardware-IRQ8 |
| 71 | 1C4 | synchrone serielle Schnittstelle oder frei für Anwendung | BIOS oder Anwendung | Hardware-IRQ9 |
| 72 | 1C8 | Timer 1 | LCD-BIOS-Extension oder Anwendung | Hardware-IRQ10 |
| 73 | 1CC | Timer 2 | Anwendung | Hardware-IRQ11 |
| 74 | 1D0 | reserviert (DMA) | -- | Hardware-IRQ12 |
| 75 | 1D4 | frei für Anwendung | Anwendung | Hardware-IRQ13 |
| 76 | 1D8 | IDE (Compact-Flash) | BIOS | Hardware-IRQ14 |
| 77 | 1DC | 386EX-Watchdog Timer | Anwendung | Hardware-IRQ15 |
| 78 – FF | 1E0 – 3FC | frei für Anwendung | Anwendung | Software |

10.3. Steckerbelegung

10.3.1. PIF (X1)

| Pin | Signal | Pin | Signal |
|-----|--------|-----|----------------|
| 1 | GND | 2 | GND |
| 3 | VCC | 4 | VEE oder /CS4* |
| 5 | /WR | 6 | /RD |
| 7 | /CS0 | 8 | A0 |
| 9 | A1 | 10 | /Reset |
| 11 | D0 | 12 | D1 |
| 13 | D2 | 14 | D3 |
| 15 | D4 | 16 | D5 |
| 17 | D6 | 18 | D7 |
| 19 | Ready | 20 | A2 |
| 21 | A3 | 22 | /CS1 |
| 23 | /CS2 | 24 | /CS3 |
| 25 | /INT | 26 | GND |

* Die VEE-Leitung kann als fünftes Chip-Select (/CS4) verwendet werden, falls der Jumper R9 (0 Ohm Widerstand Bauform 805) bestückt ist.

10.3.2. I/O (X2)

| Pin | Signal | | Pin | Signal | |
|-----|---------------|--------------|-----|------------------------|---------------|
| 1 | VCC | | 2 | COM1 /DSR | I/O P1.3 |
| 3 | COM1 /RI | I/O P1.4 | 4 | COM1 RXD | I/O P2.5 |
| 5 | COM1 TXD | I/O P2.6 | 6 | COM1 /DTR | I/O P1.2 |
| 7 | COM1 /RTS | I/O P1.1 | 8 | COM1 /CTS | I/O P2.7 |
| 9 | COM1 /DCD | I/O P1.0 | 10 | GND | |
| 11 | Timer 2 Gate | | 12 | Timer 2 Out | |
| 13 | Timer 2 Clock | | 14 | Timer 1 Out | IRQ3 I/O P3.1 |
| 15 | IRQ9 | Timer 0 Gate | 16 | Timer 1 Clock | IRQ13 |
| 17 | I/O P3.0 | IRQ4 | 18 | COM-Clock (1,8432 MHz) | |
| 19 | /RTC-On | IRQ8 | 20 | I/O P3.3 | IRQ5 |
| 21 | IRQ6 | I/O P3.4 | 22 | I/O P3.5 | IRQ7 |
| 23 | VUNST | | 24 | VBATT | |
| 25 | /RESET PIF | | 26 | GND | |

10.3.3. COM2 und synchrone serielle Schnittstelle (X3)

| Pin | Signal | | Pin | Signal | |
|-----|-----------|--------|-----|-----------|--------|
| 1 | VCC | | 2 | COM2 /DSR | STXCLK |
| 3 | COM2 /RI | SSIORX | 4 | COM2 RXD | |
| 5 | COM2 TXD | | 6 | COM2 /DTR | SRXCLK |
| 7 | COM2 /RTS | SSIoTX | 8 | COM2 /CTS | |
| 9 | COM2 /DCD | | 10 | GND | |

10.3.4. JTAG / I²C-Bus (X6)

| PIN | Signal |
|-----|----------------------------|
| 1 | VCC |
| 2 | SDA (I ² C-Bus) |
| 3 | SCL (I ² C-Bus) |
| 4 | GND |
| 5 | TMS (JTAG) |
| 6 | TCK (JTAG) |
| 7 | TDO (JTAG) |
| 8 | TDI (JTAG) |

10.3.5. Ethernet Twisted Pair (X8)

| PIN | Signal |
|-----|--------|
| 1 | TXD+ |
| 2 | TXD- |
| 3 | RXD+ |
| 4 | RXD- |

10.4. Vom BIOS verwendete ANSI Escape Sequenzen

Folgende Escape Sequenzen werden vom Video BIOS (INT 10h) verwendet :

| Escape Sequenz | Bedeutung | INT 10h |
|--------------------|-------------------------|---------------|
| ESC[# ; # H | Set cursor position | 02h |
| ESC[6 n | Status request | 03h |
| ESC[s | Save cursor position | 09h, 0Ah |
| ESC[u | Restore cursor position | 09h, 0Ah |
| ESC[2J | Erase screen | 00h, 0Eh |
| ESC[# ; ... ; # m | Set colors | 06h, 07h, 09h |

Folgende Escape Sequenzen werden vom Tastatur-BIOS (INT 16h) erkannt :

| Taste | Normal | + Shift | + Ctrl | + Alt |
|--------|--------|---------|--------|-------|
| F1 | ESC[M | ESC[Y | ESC[k | ESC[w |
| F2 | ESC[N | ESC[Z | ESC[l | ESC[x |
| F3 | ESC[O | ESC[a | ESC[m | ESC[y |
| F4 | ESC[P | ESC[b | ESC[n | ESC[z |
| F5 | ESC[Q | ESC[c | ESC[o | ESC[@ |
| F6 | ESC[R | ESC[d | ESC[p | ESC[[|
| F7 | ESC[S | ESC[e | ESC[q | ESC[\ |
| F8 | ESC[T | ESC[f | ESC[r | ESC[] |
| F9 | ESC[U | ESC[g | ESC[s | ESC[^ |
| F10 | ESC[V | ESC[h | ESC[t | ESC[_ |
| Left | ESC[D | Home | ESC[H | |
| Right | ESC[C | End | ESC[F | |
| Up | ESC[A | PgUp | ESC[I | |
| Down | ESC[B | PgDown | ESC[G | |
| Insert | ESC[L | | | |

10.5. Default-I/O-Adressen der PIF-Cards und PIF-Module

| Chip-Select | I/O-Adresse | PIF-Modul oder PIF-CARD | | |
|-------------|-------------|-------------------------|-----------|---------|
| CS0 | 300 | PIF-IDE | PIF-SDISK | PIF-CAN |
| | 301 | | | |
| | 302 | | | |
| | 303 | | | |
| | 304 | | | |
| | 305 | | PIF-SCC | |
| | 306 | | | |
| | 307 | | | |
| | 308 | | | |
| | 309 | | | |
| | 30A | PIF-SIO/LPT LPT | | |
| | 30B | | | |
| | 30C | | | |
| | 30D | | | |
| | 30E | | | |
| 30F | | | | |
| CS1 | 310 | PIF-PIO | | |
| | 311 | | | |
| | 312 | | | |
| | 313 | PIF-ADC12/6CH | | |
| | 314 | | | |
| | 315 | | | |
| | 316 | PIF-LCD-BASE LCD-PIO | | |
| | 317 | | | |
| | 318 | | | |
| | 319 | PIF-LCD-BASE KEY-PIO | | |
| | 31A | | | |
| | 31B | | | |
| | 31C | | | |
| | 31D | | | |
| | 31E | | | |
| 31F | | | | |

(Fortsetzung: Default-I/O-Adressen der PIF-Cards und PIF-Module)

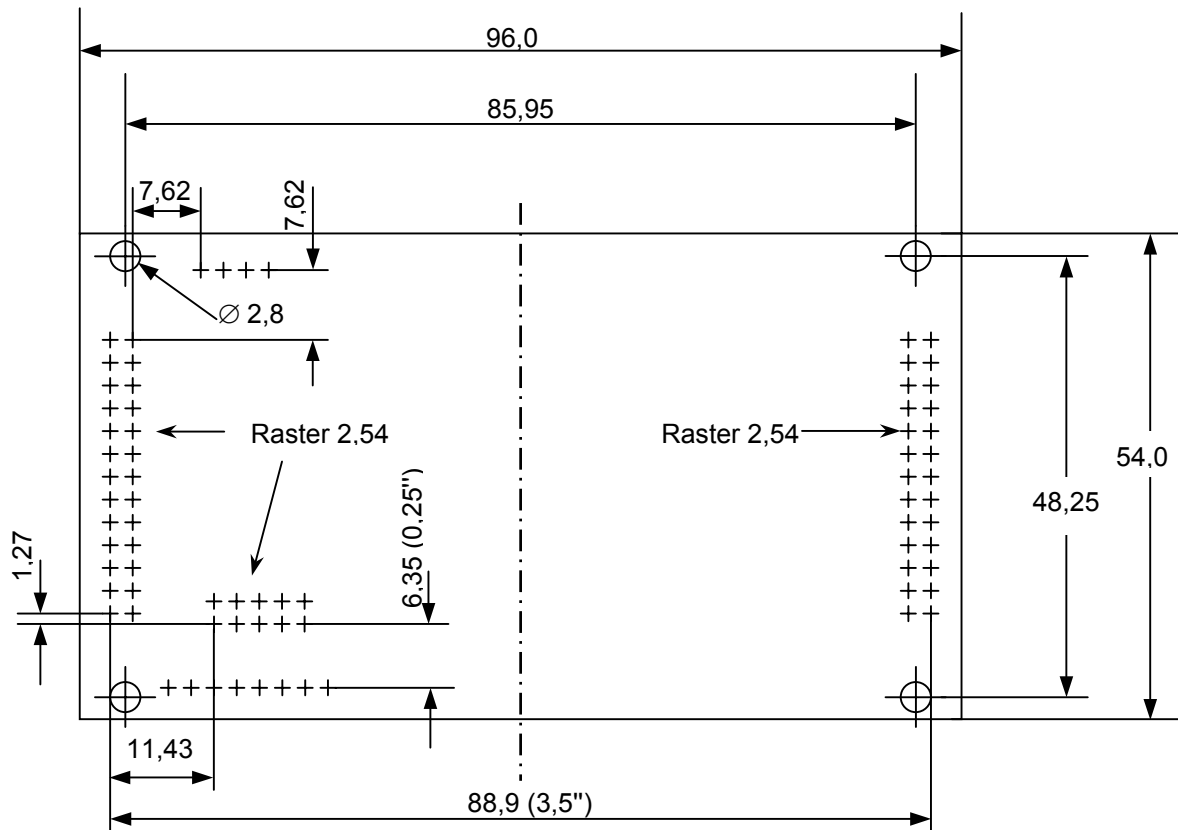
| | | | | |
|-----|-------------------|-----------|--|--------------|
| CS2 | 320 | PIF-SIO | | PIF-Ethernet |
| | 321 | | | |
| | 322 | | | |
| | 323 | | | |
| | 324 | PIF-TIMER | | |
| | 325 | | | |
| | 326 | | | |
| | 327 | | | |
| | 328 | PIF-LPT | | |
| | 329 | | | |
| | 32A | | | |
| | 32B | | | |
| | 32C | | | |
| 32D | PIF-I/O-24V | | | |
| 32E | PIF-RELAIS/230VAC | | | |
| 32F | PIF-RELAIS/24VDC | | | |

| | | | | | |
|-----|-----|-----------------------------|--|--------------|---------|
| CS3 | 330 | PIF-SIO/LPT SIO-1 = COM4 | | PIF-Ethernet | |
| | 331 | | | | |
| | 332 | | | | |
| | 333 | | | | |
| | 334 | | | | |
| | 335 | | | | |
| | 336 | | | | |
| | 337 | | | | |
| | 338 | | | | PIF-LPT |
| | 339 | | | | |
| | 33A | PIF-SIO/LPT SIO-0 = COM3 | | | |
| | 33B | | | | |
| | 33C | | | | |
| | 33D | | | | |
| 33E | | | | | |
| 33F | | | | | |

10.6. Elektrische Daten

| Symbol | Beschreibung | Parameter | min. | typ. | max. | Einheit | |
|--------|---|--|------|------|------|---------|--|
| Vcc | Betriebsspannung | | 4,75 | 5 | 5,25 | V | |
| Vres | Reset-Schwelle | | 4,5 | 4,65 | 4,75 | V | |
| Vnmi | Power-Fail Komparator-Schwelle (für NMI) | | 1,20 | 1,25 | 1,30 | V | |
| Icc | Betriebsstrom bei max. Takt | CLK2 = 66,6 MHz | | 320 | 400 | mA | |
| | | CLK2 = 33,3 MHz | | 180 | | mA | |
| | | CLK2 = 16,7 MHz | | 105 | | mA | |
| | | CLK2 = 1,5 MHz | | 38 | | mA | |
| | Idle-Mode: | | | | | | |
| | | CLK2 = 66,6 MHz | | 100 | | mA | |
| | | CLK2 = 33,3 MHz | | 66 | | mA | |
| | | CLK2 = 16,7 MHz | | 50 | | mA | |
| | Deep Powerdown | CLK2 = 1,5 MHz | | 37 | | mA | |
| | | CLK2 = 0 | | 6,5 | | mA | |
| Vbatt | Batteriespannung (für SRAM und RTC) | | 2,0 | 3 | 5,25 | V | |
| | Batteriedaten nach Herstellerangaben | Lithium-Batterie Renata CR2032RH | | | | | |
| Cbatt | Batteriekapazität | | | 200 | | mAh | |
| Tbatt | Betriebstemperatur | | -20 | | 70 | °C | |
| | Selbstentladung | Umgebungstemperatur = 25°C | | | 1% | 1/Jahr | |
| Ibatt | Batteriestrom bei abgeschalteter Betriebsspannung | 1MB SRAM Bestückung Batteriespannung = 3V Umgebungstemperatur = 25°C | | 5 | | µA | |
| | | Umgebungstemperatur = 70°C | | | | µA | |
| | | Umgebungstemperatur = 85°C | | | | µA | |

11. Maßzeichnung 386EX-Card III



Die beiden 26-poligen Wannestifteleisten (PIF und I/O) liegen zueinander im Raster 2,54 mm.

Die 10-polige Wannestifteleiste (SIO1) liegt zu PIF und I/O im Halbraster 1,27 mm. Dies ist wichtig, wenn eine Grundplatine direkt mit SIO1 verbunden werden soll.

| | |
|---|--------|
| Bauhöhe der Bauteile auf der Vorderseite: | 4,2 mm |
| Bauhöhe der Bauteile auf der Rückseite: | 2,5 mm |
| Stärke der Leiterplatte: | 1,6 mm |
| Höhe der Steckverbinder (Wannestifteleisten): | 9 mm |