

# **TURBO-FREEZER XL/XE 2005**

Version 1.20 vom 23.09.2008  
für Freezer Software Version 3.02



## Inhalt

Inhalt .....	3
Vorwort des Neukonstrukteurs .....	5
Vorwort des Entwicklers (2005) .....	6
Vorwort des Entwicklers (1987) .....	7
1 Installation .....	8
1.1 Anstecken des Freezer am Parallelen Bus .....	8
1.1.1 ATARI 600 XL (mind. 64kRAM) und ATARI 800 XL .....	8
1.1.1 ATARI 130 XE .....	8
1.1 Erste Inbetriebnahme .....	8
1.1.1 Schritt 1: Grundkonfiguration setzen .....	8
1.1.1 Freezer starten .....	9
1.1.1 Wird das Kabel benötigt (nur Freezer XL)? .....	9
1.1.1 Im Falle eines Fehlerhaften Freezers .....	9
1.1.1 Erstellung der TURBO-FREEZER System-Disk .....	10
1.1 Für Lötexperten: Interne Stromversorgung .....	11
1.1 Für Lötexperten: Einbau einer SYSTEM RESET Taste .....	11
1.1 Kompatibilität zu 1MB Ramdisks .....	11
1 Einsatz des Freezers .....	13
1.1 Einfrieren und Auftauen von Programmen .....	14
1.1 Abspeichern von eingefrorenen Programmen .....	15
1.1 Wechseln zwischen eingefrorenem Programm und gespeichertem Programm in RAM-Disk oder Freezer-RAM .....	15
1.1.1 Hinweise zum Ladeprogramm von Bootdateien .....	16
1.1 Löschen des OS-Bereichs .....	16
1.1 Starten des DOS und des Debuggers .....	17
1.1 Aufrufen der Cartridge-Emulation .....	17
1.1 Backups nachladender Programme .....	17
1 Das eingebaute DOS .....	18
1.1 Eingabe von Befehlen .....	18
1.1 Grundregeln für die Befehlsausführung .....	18
1.1 Kommandoübersicht .....	19
1.1 Besonderheiten mancher Kommandos .....	19
1.5 Fehlermeldungen des DOS .....	20
1 Der Debugger .....	21
1.1 Befehlssatz des Debuggers .....	22
1.1.1 Eingabe Hexadezimal, Dezimal, Interner Code .....	23
1.1.1 Eingabe von Vektoren .....	23
1.1.1 Display List anzeigen .....	23
1.1.1 Suche .....	24
1.1.1 SIO-Kommando .....	24
1.1.1 Interrupt und VCOUNT anzeigen .....	24
1.1.7 Direct IO Modus .....	25
1.1.8 Zugriff auf erweiterten Speicher / RAM-Disks .....	25
1.1.9 Versionsnummer .....	25
1.1.10 Vektoren .....	26
1.1.11 Handler-Tabelle .....	26
1.1.12 Hilfe .....	26

1.1.13	Speicherkartierung / Memory Map.....	26
1.1.14	ATARI-SIO Direktkomando.....	26
1.2	Kommandos für die Cartridge-Emulation .....	26
1	Der Oldrunner .....	28
1	Die Cartridge-Emulation .....	29
2	Programmieren des Flash-ROMs.....	32
2.1	Hinweise zur angepassten Maxflash Software.....	34
3	Für Experten: Technische Details .....	36
3.1	Details der Cartridge-Emulation .....	37
3.2	Software-Konfiguration der Cartridge-Emulation .....	37
3.3	Der Oldrunner Modus.....	39
3.4	Der Freezer-Modus.....	39
3.5	Aktivierung des Freezers .....	40
4	Speicheraufteilung im Freezer .....	43
5	Weiterführende Informationen und Links .....	45
6	Garantie- und Verwendungsbestimmungen .....	46

## **Vorwort des Neukonstrukteurs**

Der Startschuss für die Entwicklung des neuen TURBO-FREEZERs fiel Anfang 2004 als Bernhard Engl dem ABBUC die Erlaubnis gab seine Produkte (die 1050 Turbo und den TURBO-FREEZER) neu aufzulegen. Für mich persönlich war der TURBO-FREEZER schon immer die interessanteste Erweiterung die je für die ATARIs existierte und ich hatte mich auch schon zuvor sehr intensiv mit dem TURBO-FREEZER beschäftigt und seine Funktionsweise vollständig analysiert. Demnach war es ziemlich klar, dass ich mich um den Nachbau des TURBO-FREEZERs kümmern würde. Mit beim Team waren noch Bernhard Pahl, Florian Dinger, Torsten Schall, Guus Assmann, Frank Schröder und gegen Ende auch noch Wolfram Fischer. An dieser Stelle möchte ich ein großes Dankeschön an alle Team-Mitglieder aussprechen, ohne euch wäre ich wohl nie so weit gekommen! Besonders bedanken möchte ich mich bei Guus, für die vielen Erweiterungs-Ideen, das Erstellen der Platinen-Layouts und den Aufbau der Prototypen, Bernhard Engl für alle Tips und die detaillierten Infos zum original TURBO-FREEZER, sowie bei allen anderen für die Anregungen und für's Testen!

Mein ursprünglicher Plan war, den original TURBO-FREEZER einfach mit aktuell erhältlichen Bauteilen neu aufzulegen. Zu dieser Zeit hätte ich mir nie gedacht, dass dabei zum Schluss eine so umfangreiche und mächtige Erweiterung herauskommen würde! Im Laufe des Projektes habe ich sowohl die Hardware als auch die Software des TURBO-FREEZERs nach und nach erweitert, so lange bis von der original TURBO-FREEZER Logik nur mehr das grundlegende (und meiner Meinung nach geniale) Funktionsprinzip übrig geblieben war. Einige der Erweiterungen am TURBO-FREEZER sind übrigens recht zufällig entstanden. Zum Beispiel die Cartridge Emulation: Da 16k EPROMs deutlich teurer sind als aktuelle Flash-ROMs, hat Guus vorgeschlagen ein Flash-ROM zu verwenden. Um das Flash-ROM direkt vom ATARI aus programmieren zu können, war es notwendig dass man das Flash-ROM in den Speicher des ATARIs einblenden kann. Das war die Basis der Cartridge Emulation :-). Ähnlich verhielt es sich beim verwendeten RAM: Das originale 2k RAM war schwer zu bekommen und auch recht teuer. Also haben wir uns in einem ersten Schritt dazu entschlossen, 32k RAM und 128k Flash-ROM zu verwenden. In einem nächsten Schritt haben wir das RAM von 32k auf 128k aufgestockt, damit war es nun auch möglich einen kompletten Snapshot im TURBO-FREEZER RAM abzulegen, und das Flash-ROM durch einen 512k Typ ersetzt. Dies erforderte aber wiederum einen leistungsfähigeren Logik-IC. Durch das neue Logik-IC war es dann wiederum möglich auch 16k OSS-Module und das SpartaDosX-Modul zu emulieren.

Mit dem aktuell erreichten Stand ist nun auch das Logik-IC komplett voll und wir alle sind zufrieden das maximal Mögliche aus der Hardware herausgeholt zu haben. Für die Software existieren schon einige Erweiterungsideen, doch mehr wird dazu noch nicht verraten.

Ein wichtiger Punkt während der Entwicklung war für mich immer, auch eine offene Basis für zukünftige Erweiterungen zu erstellen. So sind alle Informationen frei erhältlich und auf der TURBO-FREEZER-Platine ist sogar ein JTAG Interface integriert mit dem der Logik-IC neu programmiert werden kann. Somit kann jeder selber Erweiterungen am TURBO-FREEZER erstellen oder sogar komplett neue Funktionen realisieren!

Aktuelle Informationen und Updates zum TURBO-FREEZER XL/XE 2005 sind im Internet unter <http://turbofreezer.horus.com/> verfügbar. Dort gibt es auch eine Anleitung wie man die Logik des TURBO-FREEZERs selbst verändern kann. Bei technischen Fragen schickt mir am besten eine Email an [hias@horus.com](mailto:hias@horus.com).

Die Funktionsvielfalt des TURBO-FREEZER hat bisher nicht nur alle Team-Mitglieder überzeugt sondern auch die Mitglieder des ABBUC: beim Hardware Wettbewerb 2005 hat der TURBO-FREEZER mit deutlichem Abstand den ersten Platz belegt!

Ich wünsche euch viel Spass mit dem TURBO-FREEZER und bleibt den ATARI 8-bit Computern noch lange treu!

Salzburg, im Dezember 2005  
Matthias Reichl

### ***Vorwort des Entwicklers (2005)***

Die größte Anerkennung, die man als Entwickler eines einst erfolgreichen Produkts bekommen kann, besteht darin, dass ein dazu würdiger Nachfolger die Ideen aufgreift und weiterentwickelt. Wenn dies jedoch nach sechzehn Jahren und noch dazu im Bereich der Homecomputer geschieht, ist das sogar sensationell und zeigt, dass das damalige Produkt heute Kultstatus genießt. Mit dem TURBO-FREEZER XL/XE 2005 hat Matthias Reichl aufbauend auf meinem alten Grundkonzept aus den 1980er Jahren ein echtes Meisterstück abgeliefert, das unter Verwendung zeitgemäßer Mikrochips einen enorm erweiterten Funktionsumfang aufweist und nebenbei auch noch einige kleinere Schwächen beseitigt, die mir damals zwar bekannt waren, aber wegen der beschränkten Möglichkeiten der damaligen Halbleitertechnik leider nicht vermieden werden konnten, ohne dass die Chipanzahl und damit der Preis des Produktes unwirtschaftlich hoch geworden wäre.

Durch diese Verbesserungen und Erweiterungen, insbesondere auch der neuen genialen Cartridge-Emulationen, die das Vorbild nicht hatte, ist der neue TURBO-FREEZER XL/XE nicht nur der beste und perfektteste Freezer für den 8-Bit ATARI, der jemals existiert hat, sondern ein wahres Multitalent, das ebenfalls das Zeug dazu hat, einmal legendär und zum Kult zu werden. Er stellt den vorläufigen Höhepunkt einer technischen Entwicklung dar, die vor mehr als 20 Jahren begann: in meinen inzwischen vergilbten Notizbüchern tauchen erste Skizzen des Konzepts etwa im Herbst 1984 auf (das herauszufinden hat mich selber überrascht).

Nach einer so langen Zeit eine stark erweiterte, verbesserte und zeitgemäße Neuauflage eines Computer-Hardwarezusatzes erleben zu dürfen, ist nicht nur ein Wunder, sondern für mich eine besondere Freude und Ehrung, und ich wünsche allen stolzen Besitzern des neuen TURBO-FREEZER XL/XE ebenfalls viel Freude damit!

Zürich, im November 2005  
Bernhard Engl

## ***Vorwort des Entwicklers (1987)***

Nach über einem Jahr Entwicklungszeit, hunderten von verbrauchten ICs, 5300 Assemblerzeilen und fünfstelligen Kosten ist es soweit: Mit dem TURBO-FREEZER XL ist nicht nur der erste echte Freezer für den ATARI fertig, sondern zugleich auch die einzige Erweiterung, die der Freak braucht, um seinen ATARI optimal und maximal zu nutzen. Denn zusätzlich zum Freezer befindet sich auf derselben Platine noch Platz für einen Oldrunner sowie für 256 KBytes RAM, so dass der 800 XL damit auf bis zu 320K erweitert werden kann.

Dank der Verwendung dreier ASICs, die zusammen über 40 normale TTL-ICs ersetzen und dank der gerade noch geglückten Vermeidung einer teuren Multilayerplatine ist es gelungen, dies alles zu einem konkurrenzlos günstigen Preis zu realisieren. Zudem kommt der TURBO-FREEZER XL ohne Eingriffe in den Rechner aus, was durch die für den Freezer ohnehin nötige hochentwickelte Memory-Management Logik ermöglicht wird.

Dieselbe sorgfältige Abstimmung aller Komponenten findet sich auch bei der eingebauten Software, die aus dem Freezer, einem Mini-DOS und einem Debugger besteht. Zusammen mit den Möglichkeiten der Hardware ergibt sich ein Power-Tool, das dem ATARI-User ungeahnte Möglichkeiten durch die totale Beherrschung des Programmablaufs eröffnet. Kein Wunder, dass bisher jeder, der den TURBO-FREEZER XL in Aktion gesehen hat, selber ein Exemplar haben möchte. Der gesteigerte Spaß am Computern und das Gefühl, mit jedem Programm das tun zu können, was man selber will (im Gegensatz zu dem, was der Programmierer festgelegt hat) ist schon allein den Kaufpreis wert. Ganz abgesehen von dem Spareffekt bei Ausnutzung der vorgesehenen Erweiterungen, die zusammen ohne Freezer mehr kosten würden als ein voll ausgebauter TURBO-FREEZER XL.

Doch mehr dazu unter den Beschreibungen der einzelnen Funktionen! Ich wünsche jedem stolzen Besitzer noch viel Spaß mit dem faszinierendsten Produkt, das für den ATARI je auf den Markt gekommen ist.

München, im Mai 1987  
Bernhard Engl

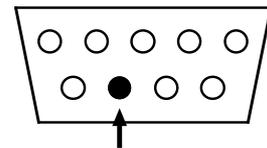
## 1 Installation

Der TURBO-FREEZER XL/XE wird einfach am Parallelen Bus angesteckt. Die Stromversorgung erfolgt durch den Parallelen Bus (beim ATARI 600 XL und bei älteren ATARI 800 XL sowie beim ATARI XE) oder durch das angelötete Kabel, das an einen Joystickport angesteckt wird (nur beim TURBO-FREEZER XL). Ein Öffnen des ATARIs oder Lötarbeiten gleich welcher Art sind daher nicht notwendig. In solchen Dingen erfahrene User mit dem entsprechenden Werkzeug können sich aber zusätzliche Möglichkeiten einbauen.

### 1.1 Anstecken des Freezer am Parallelen Bus

#### 1.1.1 ATARI 600 XL (mind. 64kRAM) und ATARI 800 XL

Einfach die eingeschnappte Abdeckung hinten am ATARI entfernen, die Blechungen mit den Fingern vorsichtig ein wenig aufbiegen und den Freezer ohne Gewaltanwendung einstecken. Das Kabel an Pin 7 des Joystickport 2 anstecken. Pin 7 ist in der nebenstehenden Zeichnung markiert.



Hinweis: Das Kabel darf auf keinen Fall angesteckt werden wenn der ATARI eingeschaltet ist, ansonsten können der ATARI und/oder der Freezer Schaden nehmen! Wenn man (versehentlich) während des Betriebes das Kabel aussteckt, so sollte man zuerst den ATARI ausschalten, danach das Kabel wieder anstecken und dann den ATARI wieder einschalten.

(Anmerkung 2005: obwohl die Firma Lattice (Hersteller der GALs) in den damaligen Datenbüchern damit angegeben hat, dass "GALs will not latch up under any condition", hat uns ein ganzer Haufen abgerauchter Freezer davon überzeugt, dass das bloße Angabe war. Heutige CMOS-Chips sind robuster und verwenden keine olle Ladungspumpe mehr für den Substrat-Backbias. Heutige CMOS Chips sind kaum mehr kaputtzukriegen. Aber dennoch: Vorsicht ist die Mutter der Porzellankiste! Daher die Warnung. B.E.)

#### 1.1.2 ATARI 130 XE

Den Freezer hinten am ATARI einstecken. Die Schalter und die Modul-Steckleiste müssen oben liegen.

### 1.2 Erste Inbetriebnahme

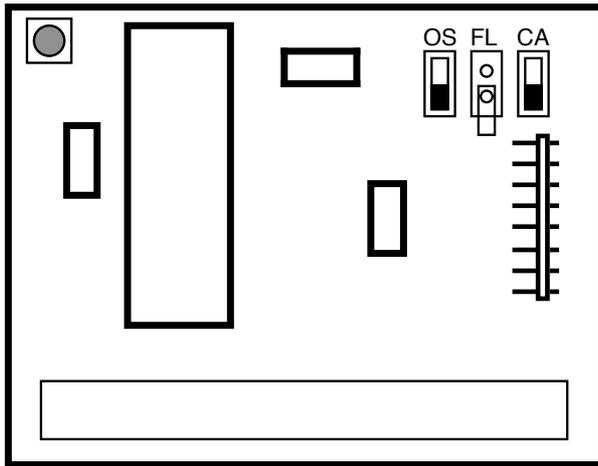
#### 1.2.1 Schritt 1: Grundkonfiguration setzen

Den Freezer in die Grundkonfiguration bringen:

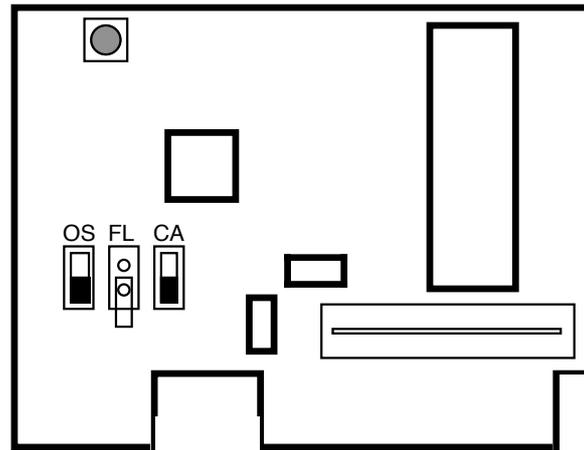
- Oldrunner „OS“ OFF,
- Flash „FL“ OFF
- Cartridge Emulation „CA“ OFF

Beim Freezer XL heißt das, alle Schalter zur Platinenmitte schieben, den Jumper in der Mitte auf „offen“ setzen.

Für den Freezer XE heißt das, alle Schalter zum ATARI hin schieben, den Jumper auf „offen“ setzen.



Freezer XL



Freezer XE

### 1.2.2 Freezer starten

Den ATARI alleine einschalten. Wenn nicht nach der üblichen kurzen Wartezeit die „READY“ Meldung des BASIC erscheint, sondern verdächtige Symptome auftreten (Systemabsturz, dunkles Bild, Rauchentwicklung) sofort ausschalten und den Fehler suchen. Ist der Oldrunner eingeschaltet, erscheint „ATARI COMPUTER – MEMO PAD“ statt „READY“. Falls die Cartridge-Emulation eingeschaltet sein sollte, erscheint der grüne Cartridge Bildschirm „TURBO CARTRIDGE“. In diesen Fällen die Grundkonfiguration gemäß Kapitel 1.2.1 überprüfen.

Ist die „READY“ erschienen, einfach den (roten) **Freezerknopf (links oben auf der Platine)** drücken, so dass das Freezer-Menü erscheint. Tut sich nichts, ist entweder der Freezer schlecht eingesteckt, das Kabel (wenn vorhanden) nicht richtig angeschlossen oder der Freezer ist kaputtgegangen.

Anschließend im Freezer-Menü die **[Leertaste]** drücken. Dadurch wird zum BASIC zurückgegangen. Ausprobieren, ob das BASIC Tasteneingaben annimmt. Wenn es das nicht tut, liegt ein Fehler vor.

Wenn bis zu diesem Punkt alles geklappt hat, ist der TURBO-FREEZER XL/XE einsatzbereit!

### 1.2.3 Wird das Kabel benötigt (nur Freezer XL)?

Um auszuprobieren, ob das Kabel überflüssig ist, wieder ins Freezer-Menü gehen und das Kabel abziehen. Wenn der ATARI abstürzt ist das Kabel notwendig. Funktioniert der Freezer auch so, dann ist das Kabel mit diesem ATARI nicht notwendig. Man sollte es aber trotzdem nicht abschneiden, sondern schön zusammenrollen; immerhin könnte es einmal nötig sein, den Freezer auch an einem ATARI ohne Stromversorgung am Parallelen Bus einzusetzen.

### 1.2.4 Im Falle eines Fehlerhaften Freezers

Sollte der Freezer aus irgendeinem Grund nicht richtig funktionieren, diesen nicht gleich zurücksenden, sondern erst einmal einen Brief oder eine Email mit einer Beschreibung der Symptome senden. Es ist aufgrund des 100%-igen Endtests eines jeden TURBO-FREEZER XL/XE praktisch ausgeschlossen, dass es am Freezer liegt. Es liegt wahrscheinlich an einem Einbaufehler oder am ATARI selbst, und es ist besser, diese Möglichkeiten durch eine Rückfrage auszuschließen, bevor der Freezer selbst zur Überprüfung eingesandt wird, um unnötige Wartezeiten und Kosten zu vermeiden.

(Hinweis 2005: das war das schon damals bekannte Timingproblem mit manchen ATARIs, bei dem die ATARI-eigene Spezifikation des parallelen Bus-Timings krass verletzt wurde. Wir haben es behoben, indem wir uns den ATARI einsenden ließen und die CPU oder den ANTIC kostenlos gewechselt haben. Die Chips hat uns ATARI Deutschland ebenso kostenlos zur Verfügung gestellt. Die neuen Freezer haben Latches an den kritischen Adressleitungen, so dass dies kein Problem mehr sein sollte, auch wenn die damaligen schlechten Chips im Spiel wären. B.E.)

### 1.2.5 Erstellung der TURBO-FREEZER System-Disk

Gleich nach der erfolgreichen Installation des TURBO-FREEZERS sollte die System-Disk erstellt werden. Im Auslieferungszustand enthält das Flash-ROM die dazu notwendige Software. Auf der System Disk sind die Flasher-Software für den Atari sowie die TURBO-FREEZER-Software enthalten. Damit kann der TURBO-FREEZER jederzeit wieder in den Auslieferungszustand zurückgesetzt werden.

Die System Disk wird wie folgt erzeugt:

1. Den ATARI ausschalten, eine Diskettenstation anschliessen, und die Cartridge-Emulation am TURBO-FREEZER aktivieren (beim XL: CA Schalter nach oben, beim XE: CA Schalter weg vom Atari schieben). Diskettenstation einschalten. ATARI einschalten.
2. Nach dem Einschalten des ATARIs sollte das Auswahlmenü der Cartridge-Emulation erscheinen. Die Default Einstellungen sind schon richtig gesetzt, sie müssen nur noch mit [RETURN] bestätigt werden.
3. Jetzt ist man in der „TURBO-FREEZER System Disk Writer“ Software. Eine leere Diskette ins Diskettenlaufwerk (D1 :) einlegen und [RETURN] drücken. Die Software formatiert nun die Disk (in Enhanced Density) und schreibt die TURBO-FREEZER-Software darauf. Sollte dabei ein Fehler auftreten (z. B. auf Grund einer defekten Diskette), so wird „ERROR“ ausgegeben und man muss eine neue Diskette einlegen und das Programm neu starten (bei „restart program“ einfach „[y]“ eingeben). Zur Sicherheit sollte man eine zweite Kopie der System-Disk erstellen und diese (schreibgeschützt) an einem sicheren Ort verstauen.
4. Nach dem erfolgreichen Erstellen der System Disk den Atari einfach abschalten und den Schalter „CA“ wieder in die Ausgangsposition zurück schieben.

Wenn alle Stricke reissen und man die System Disk verliert oder versehentlich überschreibt (jaja, sowas kann schon mal passieren!), gibt es im Internet immer noch unter <http://turbofreezer.horus.com/> ein ATR-Image der System Disk zum Download.

### **1.3 Für Lötexperten: Interne Stromversorgung**

Wer einen ATARI 800 XL ohne Stromversorgung am Parallelen Bus hat, kann diese nachträglich einbauen, was aber sehr schwierig ist und großes Geschick erfordert. Wer sich das nicht zutraut, kann auch nur die Leitung des Freezers verlängern und im Inneren des ATARIs an die Stromversorgung anschließen. In diesen Fällen bleibt der Joystickport dann für seinen eigentlichen Zweck frei!

**Warnung: dies ist wirklich nur für Lötexperten. Wer kein solcher ist, sollte sich lieber einen solchen Experten suchen. Es lohnt sich weder, zu diesem Zweck eine Servicestelle aufzusuchen (extrem teuer!) noch lohnt es sich, durch Ungeschick den armen ATARI zu ruinieren!**

Um Anfänger abzuschrecken ist die folgende Beschreibung des Umbaus auf das absolute Minimum beschränkt worden. Experten finden sich damit problemlos zurecht! Wer sich einen „Experten“ gesucht hat und bemerkt, dass dieser im Dunkeln tappt, hat damit noch die letzte Chance, diesen offensichtlichen Amateur zu stoppen!

Hier die Beschreibung: der Abgriff der Stromversorgung erfolgt an dem Ende von L1, von dem eine Leiterbahn in den abgeschirmten Teil führt. Die Ferritperle L1 befindet sich unmittelbar neben dem Einschalter. An diesen Punkt wird entweder das verlängerte Versorgungskabel des Freezers angelötet, oder es wird von hier je eine Leitung zu Pin 47 und Pin 48 des Parallelen Bus geführt (die vorletzten Kontaktflächen vor Pin 49 und 50, je eine auf der Oberseite und eine auf der Unterseite der Platine). Das ist schwieriger, weil dazu der ATARI ganz zerlegt werden muss, und weil es knifflig ist, an den Enden der Kontaktflächen zu löten, ohne dass das Zinn auf den aktiven Teil der Kontaktfläche kriecht, was in jedem Fall vermieden werden muss.

### **1.4 Für Lötexperten: Einbau einer SYSTEM RESET Taste**

Für den Oldrunnerbetrieb und für das Anhalten von Programmen ohne Interrupts ist es günstig, eine SYSTEM RESET-Taste einzubauen, die einen nicht abschaltbaren NMI Interrupt auslösen kann. Der Taster wird so eingebaut, dass Pin 6 des ANTIC auf Ground geschaltet werden kann. Der Abgriff erfolgt am besten am Pull-Up Widerstand R31.

**Warnung: siehe Punkt 1.3. Dazu kommen noch die mechanischen Arbeiten für den Einbau des Tasters, die zusätzliches Werkzeug und handwerkliches Geschick erfordern, um ein ansehnliches Resultat zu ergeben!**

### **1.5 Kompatibilität zu 1MB Ramdisks**

Da der TURBO-FREEZER die Refresh-Leitung verwendet um den Atari anzuhalten, kommt es leider zu Kompatibilitätsproblemen mit allen RAM Erweiterungen, die eine eigene Refresh-Logik auf Basis des Refresh-Signals vom ANTIC einsetzen. Dies betrifft hauptsächlich 1MB Erweiterungen wie z. B. die Newell Ramdisk. Die meisten 256k-RAM-Erweiterungen sowie der 130XE sind davon nicht betroffen.

Mit zwei zusätzlichen Bauteilen (einer Kleinsignal Schottky Diode, z. B. vom Typ BAT 85 und einem 4.7kOhm-Widerstand) lässt sich dieses Problem jedoch sehr einfach beheben.

**Warnung: siehe Punkt 1.3!**

Nach dem Öffnen des Ataris muss der Pin 8 am ANTIC hochgebogen werden. Das Kabel, das von Pin 8 zur Ramdisk führt, vorher ablöten.

Ist der ANTIC gesockelt, den ANTIC aus der Fassung ziehen und sehr vorsichtig den Pin hochbiegen. Achtung: der Pin bricht sehr leicht ab wenn man ihn zu stark biegt! Es reicht aus den Pin soweit wegzubiegen, dass er beim Einstecken des ANTIC in den Sockel nicht mehr im Sockel sitzt.

Ist der ANTIC direkt auf der Platine eingelötet, den Pin mit einem Miniatur-Seitenschneider knapp über der Platine abknipsen und ebenfalls vorsichtig hochbiegen.

Als erstes muss die Diode zwischen dem hochgebogenen Pin 8 und Pin 8 des Sockels (bzw. der Platine) gelötet werden. Die Kathode (am Gehäuse mit einem dunklen Ring markiert) muss zum ANTIC, die Anode zum Sockel.

Die einfachste Lösung dafür ist es, die Diode direkt an Pin 8 des ANTIC anzulöten und von der Anode ein dünnes Kabel zur Platinen-Unterseite zu führen, das dort mit Pin 8 verbunden wird.

Ein Ende des Widerstandes muss mit der Anode der Diode verbunden werden, das andere Ende des Widerstandes mit Pin 21 des ANTIC (+5V).

Nun kann das Kabel, das von Pin 8 am ANTIC zur Ramerweiterung führt wieder an Pin 8 am ANTIC angelötet werden.

## 2 Einsatz des Freezers

Wer hat sich nicht schon über Spielprogramme geärgert, die zwar an sich ganz und gar hervorragend sind, aber über keine jederzeit aktivierbare Pausefunktion verfügen? Oder die von jener frustrierenden Sorte, bei der man nach Verlust aller Leben wieder von vorne anfangen und ewig spielen muss, bis man wieder dort ist, wo man mal war? Oder solche, bei denen erst in höheren Stufen so richtig Action aufkommt, es aber trotzdem notwendig ist, vorher viel Zeit mit dem Erreichen dieser Stufen zu vertun? Hier ist ein Freezer genau das richtige: ein Programm kann jederzeit, an jedem Punkt, eingefroren werden und in dieser Form auf einem beliebigen Massenspeicher gespeichert werden. Von dort kann es beliebig später wieder geladen und an genau derselben Stelle fortgesetzt werden, an der es eingefroren wurde. Und das beliebig oft, so dass es kein Problem ist, hundert Leben zu verbraten, um eine neue Prüfung zu meistern, obwohl man nur noch ein oder zwei Leben hat...

Damit der Freezer richtig Spaß machen kann, muss er jederzeit verfügbar sein, residente Software haben (also ohne umständliches Laden von Diskette oder Cassette auskommen) und vollautomatisch und in Sekundenschnelle arbeiten. Daher wurde beim TURBO-FREEZER XL/XE darauf verzichtet, Sparmaßnahmen zu ergreifen, die durchaus möglich gewesen wären, so z. B. die teilweise Rekonstruktion der Hardwareregister durch Eingaben des Users oder nichtresidente Software. Statt solcher dubiosen Sparmaßnahmen, wie sie teils bei den Freezern für andere Computersysteme zu beobachten sind, wurde beim TURBO-FREEZER XL/XE ohne Rücksicht auf den Aufwand das Beste geschaffen, was mit dem Stand der Technik möglich ist. Der Aufwand ist aber, wie das Resultat beweist, nicht ohne Wirkung geblieben! Wer die umständlichen Primitiv-Freezer anderer Computer kennt, kann über den mühelosen und sekundenschnellen Einsatz des TURBO-FREEZER XL nur begeistert sein. Beispiel: um ein Spiel in die RAM-Disk wegzufrieren und trotzdem drei Sekunden später weiterzuspielen, sind nur drei Tastendrucke nötig! Dasselbe gilt für das spätere Laden und Auftauen aus der RAM-Disk. So manchem C64-User, der bisher mit dem Mangel an Freezern beim ATARI angeben könnte, dürfte dabei der Kinnladen auf den Fußboden krachen!

Doch der Freezer kann noch viel mehr. Es sind beliebige Konvertierungen zwischen Cassette und Diskette möglich. Cassettenuser mit neuerworbener Floppy können ihre geliebte Softwaresammlung auf Diskette „mitnehmen“. Diskettenuser kommen auch ohne wiederholte Ladequalen an die Programme, die es nur auf Cassette gibt, oder können durch Kauf der billigeren Cassettenversionen Geld sparen!

Und es gibt noch einen Punkt: der TURBO-FREEZER XL/XE ist der erste Freezer, bei dem ein DOS und ein Debugger eingebaut sind, die jederzeit zur Verfügung stehen, und die jederzeit verwendet werden können, ohne dass das eingefrorene Programm dabei beschädigt wird! Volle Disketten oder sonstige fatale Vorkommnisse verlieren damit auch bei der Verwendung von Anwenderprogrammen ohne DOS-Funktionen ihre Schrecken, ebenso die üblen Bugs, die man ohne Einblick in die Hardwareregister und den unversehrten Systemzustand nie findet.

## 2.1 Einfrieren und Auftauen von Programmen

Nach dem Drücken des roten Knopfs wird das Programm beim nächsten Interrupt eingefroren, und der Freezer übernimmt die Kontrolle über den ATARI. Über das Freezer-Hauptmenü können dann alle weiteren Funktionen aufgerufen werden. Ein Auftauen des eingefrorenen Programms ist durch Betätigung der [Leertaste] möglich, das Programm läuft dann an genau derselben Stelle weiter, an dem es eingefroren wurde.

Nach dem Laden eines eingefrorenen Programms von Diskette, Cassette, Freezer-RAM oder RAM-Disk wird dieses automatisch aufgetaut.

Ein Drücken der [RESET]-Taste im Freezer-Hauptmenü bewirkt einen Kaltstart und schaltet den Freezer in geregelter Weise wieder ab.

Einfrieren/Freezen	[Freezerknopf]
Auftauen	[Leertaste]
Auftauen mit PM-Register löschen	[Shift][Leertaste]
Kaltstart	[RESET] wenn im Freezermenü

Taut man ein eingefrorenes Programm mit [Shift]+[Leertaste] auf, so werden die Player/Missile-Kollisionsregister gelöscht, bevor die Kontrolle an das eingefrorene Programm übergeben wird. Für gewöhnlich benötigt man diese Funktion nicht, bei einigen wenigen Spielen verhindert dies allerdings, dass man direkt nach dem Auftauen ein Leben verliert. Beim Laden von eingefrorenen Programmen steht diese Funktion übrigens auch zur Verfügung, einfach [Shift]+[E], [Shift]+[X] oder [Shift]+[C] drücken.

Lädt man ein eingefrorenes Programm mit [Control]+[E], [Control]+[X] oder [Control]+[C], so wird es nicht automatisch nach dem Laden gestartet. Man bleibt damit im Freezer und kann z. B. vor dem Auftauen noch Speicherzellen ändern. Diese Funktion ist besonders dann praktisch, wenn man bei einem Spiel nach der Speicherzelle für Leben/Energie/... sucht: Direkt nach dem Einfrieren das Programm abspeichern, dann eine Speicherzelle ändern und das Programm wieder auftauen. Hat man die falsche Speicherzelle erwischt, einfach wieder den Freezer aktivieren, das eingefrorene Programm mit [Control]+[E], [Control]+[X], [Control]+[C] laden und die nächste Speicherzelle ausprobieren. Somit hat man bei jedem Versuch die gleichen Ausgangsbedingungen und muss sich keine Gedanken darüber machen wie man die vorherigen Änderungen wieder rückgängig machen kann!

[E]	Programm von <b>externem Gerät (Cassette/Diskette)</b> starten
[Shift][E]	Programm externem Gerät (Cassette/Diskette) laden und vor dem Starten Player-Missile-Kollisionsregister löschen
[Control][E]	Programm externem Gerät (Cassette/Diskette) laden und NICHT starten
[X]	Programm aus der <b>RAM-Disk</b> starten
[Shift][X]	Programm aus der RAM-Disk laden und vor dem Starten Player-Missile-Kollisionsregister löschen
[Control][X]	Programm aus der RAM-Disk laden und NICHT starten
[C]	Programm aus dem <b>Freezer-RAM</b> starten
[Shift][C]	Programm aus dem Freezer-RAM laden und vor dem Starten Player-Missile-Kollisionsregister löschen
[Control][C]	Programm aus dem Freezer-RAM laden und NICHT starten

Im Prinzip kann jedes Programm an jeder beliebigen Stelle eingefroren werden, also auch während einer Disketten- oder Cassettenoperation. Zu empfehlen ist das nicht, da diese Operation dann unvollständig bleibt. Nach dem Auftauen des Programms besteht zwar für das OS die Möglichkeit, die Diskettenoperation zu wiederholen, dies ist aber nicht sicher! Die Wiederholung einer Cassettenoperation ist dem OS überhaupt nicht möglich, da das Cassettenlaufwerk dazu zu „dumm“ ist. Demnach ist es besser, Programme nur einzufrieren, wenn gerade keine Operationen mit der Peripherie erfolgen.

Es kann in äußerst seltenen Einzelfällen vorkommen, dass nach dem Drücken des Freezerknopfs nichts passiert, das Programm also ungerührt weiterläuft. Das Geheimnis dahinter: dieses Programm verwendet gar keine Interrupts! Möglich ist das natürlich nur bei ganz einfachen Programmen wie z. B. konvertierten Apple-Programmen, die die Fähigkeiten des ATARIs praktisch nicht nutzen. Und einfrieren kann man sie doch, indem man nämlich einen SYSTEM RESET auslöst (dazu braucht man aber einen ATARI mit eingebauter [SYSTEM RESET]-Taste nach Kapitel 1.4. Der hiermit verbundene Interrupt kann nämlich auf keinen Fall unterdrückt werden, so dass es gegen den TURBO-FREEZER XL/XE keine Gegenmaßnahme gibt.

## **2.2 Abspeichern von eingefrorenen Programmen**

Mit den Funktionen [ S ], [ R ] und [ F ] können eingefrorene Programme auf externe Massenspeicher wie Cassette oder Diskette (mit [ S ]), in die Ram-Disk (mit [ R ]) oder in das RAM im Freezer (mit [ F ]) weggespeichert werden.

Bei [ S ] wird mittels eines Untermenüs gefragt, ob die Speicherung auf Cassette als Boot oder als File (Datei) erfolgen soll. Vor dieser Eingabe sollten die nötigen Betriebsmittel bereitgestellt werden (Cassette bzw. Diskette eingelegt). Im Falle eines „Files“ auf Diskette wird mit einem weiteren Menü der Dateiname erfragt. Die Eingabe kann wie üblich editiert werden.

**Achtung:** beim Abspeichern als Boot wird direkt auf die Diskette in D1 : geschrieben, eventuell vorhandene Daten gehen dabei verloren!

Wenn statt eines Dateinamens nur [ RETURN ] gedrückt wird, verwendet der Freezer den Dateinamen „CORE“ (steht für „Speicherabzug“). Besonderheit: wenn eine bereits existierende Datei überschrieben werden soll, können auch Wildcards eingegeben werden. Stellt man kein „D :“, „D1 :“, „D2 :“ etc. vor den Dateinamen, so wird automatisch „D1 :“ verwendet. Der Freezer unterstützt übrigens maximal 8 Diskettenlaufwerke („D1 :“ bis „D8 :“).

[ S ]	eingefrorenes Programm speichern auf Diskette oder Cassette
[ R ]	eingefrorenes Programm in die RAM-Disk speichern
[ F ]	eingefrorenes Programm in das Freezer-RAM speichern

Es ist normal, wenn das Menü während der Ausgabeoperationen hin und wieder ausgeschaltet wird. Um die Operationen auf größtmögliche Geschwindigkeit zu züchten wurde in Kauf genommen, dass das Bild beim Wiedereinschalten „hüpfen“ kann. Dadurch dauert z. B. das Speichern in die RAM-Disk nur halb so lang, was die unsaubere Methode rechtfertigt.

## **2.3 Wechseln zwischen eingefrorenem Programm und gespeichertem Programm in RAM-Disk oder Freezer-RAM**

Mit der SWAP-Funktion im Freezer-Hauptmenü kann zwischen dem gerade eingefrorenen Programm und einem in der RAM-Disk oder dem Freezer-RAM bereits abgelegten Programm

gewechselt werden. Damit kann man sehr schnell zwischen zwei verschiedenen Programmen wechseln. Achtung: wenn kein lauffähiges Programm in der RAM-Disk oder dem Freezer-RAM vorhanden ist, stürzt der ATARI ab!

- [ W ]            Gefrorenes Programm tauschen mit Programm im **Freezer-RAM** und starten
- [ A ]            Gefrorenes Programm tauschen mit Programm in der **RAM-Disk** und starten

Die Zusatz Tasten [Shift] für Tauschen, PM-Register löschen und starten sowie [Control] für Tauschen und nicht Starten stehen hier ebenso wie bei den normalen Auftau-Funktionen zur Verfügung.

### 2.3.1 Hinweise zum Ladeprogramm von Bootdateien

Wird ein eingefrorenes Programm als Boot-Datei gespeichert, so schreibt der TurboFreezer ein Ladeprogramm mit auf die Diskette.

Dies funktioniert naturgemäß nicht mit allen Programmen, denn das Ladeprogramm braucht etwa 2kByte Speicherplatz. Benötigt das eingefrorene Programm den vollständigen Speicher des ATARI, funktioniert das nicht.

Standardmäßig benutzt das Ladeprogramm den Bereich zwischen \$C000 und \$C6FF. Das garantiert normalerweise die Funktion mit alten Programmen, die nur 48kByte Speicher benötigen.

Falls es zu Problemen kommt, kann man die Startadresse des Ladeprogramms leicht ändern:

Das erste Byte des Bootsektors enthält die Start-Page des Ladeprogramms (standardmäßig \$C0). Dieses Byte zu ändern ist mit dem TurboFreezer ganz leicht: man startet den Debugger, lädt den ersten Sektor, ändert das erste Byte auf den gewünschten Wert und schreibt den Sektor auf die Diskette zurück. Bitte beachten: nur Werte von \$05 bis \$C9 und von \$D0 bis \$F9 sind erlaubt. Wenn andere Werte benutzt werden, gibt es beim Booten von der Diskette einen "BOOT ERROR".

Beispiel: wenn die Startadresse des Ladeprogramms auf \$FO gesetzt werden soll, ist folgendes im Debugger einzugeben:

```
SR 1
C D700<F0
SW 1
```

### 2.4 Löschen des OS-Bereichs

Da es Programme gibt, die 64k RAM benötigen (und auch belegen) muss der Freezer auch den RAM-Bereich unter den OS-ROMs berücksichtigen. Da in diesem Bereich beim Einschalten nutzlose Zufallsdaten stehen, die sich nicht besonders zur Komprimierung eignen, handelt man sich dann bei 48k Programmen eine gewisse Ineffizienz und Speicherplatzverschwendung auf externen Datenträgern ein. Um das zu Vermeiden, kann das RAM von \$C000-\$FFFF mit der Funktion [ Z ] gelöscht werden, wenn das Programm nur 48k RAM belegt. Es ergibt sich eine Schrumpfung der Dateien von 25% bis 50% und dementsprechend geringere Ladezeiten.

- [ Z ]            RAM unter dem OS löschen

Die Funktion kann vor dem Booten des Programms oder vor dem Abspeichern eingesetzt werden. Die erste Alternative empfiehlt sich, wenn nicht ganz sicher feststeht, ob das Programm wirklich nichts im OS-Bereich ablegt.

## 2.5 Starten des DOS und des Debuggers

Mit der Funktion [D] wird das eingebaute DOS und der Debugger aktiviert, da diese eine Kommandozeile und einen vollständigen Bildschirm verwenden und nicht mit Menüs auskommen können. Trotzdem wird auch für diese Funktion kein RAM im ATARI belegt oder verändert. Die Zustände der eingefrorenen Hardwareregister bleiben selbstverständlich ebenfalls erhalten.

[D]            Debugger und DOS starten

## 2.6 Aufrufen der Cartridge-Emulation

Mit der Funktion [K] wird die Cartridge-Emulation (Steckmodul-Emulator) aufgerufen. Weitere Einzelheiten dazu sind im Kapitel 6 beschrieben. Falls der ATARI sich dabei aufhängt, bitte die [RESET]-Taste drücken.

[K]            Cartridge-Emulation aufrufen

Der Freezer macht dabei folgende Schritte:

- das TurboFreezer Menü sofort verlassen, die
- Cartridge-Emulation mit der Standard Menü Bank (62) aufrufen
- IRQs und NMIs abschalten
- OS ROM einschalten
- Returnadresse auf \$E474 setzen (WARMSTART)
- die normale Weiter-Funktion starten (entspricht [Space]-Taste drücken)

Mit der Funktion [Shift][K] wird ein Kaltstart (Sprung nach Adresse \$E477) ausgelöst. Dies ist zum Beispiel sehr nützlich, wenn man mit MyPicoDOS aus der Cartridge-Emulation starten möchte.

[Shift][K]    Kaltstart

## 2.7 Backups nachladender Programme

Das Einfrieren nachladender Programme selbst ist natürlich immer möglich. Nur muss dann nach dem Auftauen, vor jeder weiteren Aktion des laufenden Programms, die Originaldiskette wieder ins Laufwerk eingelegt werden. Dabei wird diese Disk aber freilich nicht geschont. Wünschenswert ist es, eine Backupdiskette zur Hand zu haben, um das Original an einem sicheren Ort wegschließen zu können. Mit einem kopierfähigen Floppy-Speeder tritt dabei aber heutzutage das Problem auf, dass bei weitem nicht alle neuen Programme kopiert werden können.

Weil der TURBO-FREEZER XL/XE einen auf der Orginaldiskette befindlichen Kopierschutz nicht entfernt, sondern lediglich den Spielstand abspeichert, um diesen später wieder beliebig oft auftauen zu können, muss eine kopiergeschützte Orginaldiskette wieder in das Diskettenlaufwerk eingelegt werden, bevor das Spiel aufgetaut wird. Bei Verwendung einer Diskette zur Speicherung des Spielstandes bitte die Orginaldiskette mit Schreibschutz versehen, und aufpassen, dass diese nicht versehentlich durch den Spielstand überschrieben wird.

Handelt es sich um ein Spiel, das von mehreren nicht kopiergeschützten Datendisketten nachlädt (z. B. Adventures), dann kann statt diesen eine Backup-Kopie benutzt werden, um die Orginaldisketten zu schonen.

### 3 Das eingebaute DOS

Wer kennt nicht das folgende Malheur: die letzten drei Stunden hat man ein Programm editiert und will es gerade abspeichern. Doch, wie entsetzlich, statt der erhofften Vollzugsmeldung kommt ein „File locked“ oder „Disk full“ Error. Da ist guter Rat teuer, denn ein DUP-Aufruf führt in der Regel zum Verlust des Programms (MEM.SAV ist viel zu zeitraubend und wird daher oft erst gar nicht aktiviert).

Wenn man schon mittels des Freezers ein Programm beliebig unterbrechen kann, bietet sich natürlich an, gleich ein DOS einzubauen, mit dem die wichtigsten Befehle zur Verfügung stehen. Es ist dann kein Problem mehr, solche kritischen Situationen zu bereinigen, um anschließend das eingefrorene Programm einfach fortzusetzen.

Das im TURBO-FREEZER XL/XE eingebaute DOS enthält alle für diesen Zweck notwendigen Dienste. Es ist voll Single-, Enhanced- und Double-Density tauglich sowie DOS 2.0 und DOS 2.5 kompatibel. Die Funktionen des 1050 TURBO Floppyspeeders werden voll unterstützt, außerdem können auch Happy/Speedy kompatible Floppys sowie die XF551 mit der hohen SIO Übertragungsrates angesprochen werden. Es kann aber auch eine beliebig anders getunte oder serienmäßige Floppy verwendet werden.

#### 3.1 Eingabe von Befehlen

Die Befehlseingabe erfolgt immer in der Kommandozeile unten am Bildschirm. Innerhalb der Kommandozeile ist ein Editieren in der üblichen Weise möglich. Der Cursor kann die Kommandozeile aber nicht verlassen. Mit den Cursor-Up und Cursor-Down Tasten kann man die zuletzt eingegebenen Kommandos wieder aufrufen. Der Freezer merkt sich hierbei die 4 letzten Kommandos.

Ein Diskettenbefehl besteht entweder aus einem drei Zeichen langen Kommando allein, oder aus dem Kommando, einem oder mehreren Leerzeichen und einem Filenamen/Dateinamen. Manche Kommandos erlauben die Angabe einer Option, die unmittelbar am Ende des Befehls, mit einem Schrägstrich getrennt, angehängt werden kann. Das Kommando muss unmittelbar nach dem Promptzeichen eingegeben werden. Außer dem oben erwähnten, zwingend vorgeschriebenen Leerzeichen sind keine Leerzeichen erlaubt. Filenamen/Dateinamen dürfen die Wildcards „\*“ für eine beliebige Zeichenfolge sowie „?“ für ein beliebiges Zeichen enthalten. Fehlerhafte Befehle werden einfach ignoriert und führen nicht zu einer Fehlermeldung. Eine solche kann nur auftreten, wenn bei der Befehlsausführung selbst ein Fehler auftritt.

#### 3.2 Grundregeln für die Befehlsausführung

Lässt man das „D:“ im Filenamen weg, so wird automatisch „D1:“ verwendet. Die RAM-Disk wird nicht unterstützt, da die Implementierung immer vom jeweils eingesetzten DOS oder RAM-Disk-Treiber abhängig ist.

Wildcards dürfen auch in Zieldateinamen angegeben werden. Es wird dann einfach die erste übereinstimmende Datei/File angesprochen. Die einzige Ausnahme bildet das RENAME Kommando REN, das weiter unten beschrieben wird.

Die Kommandos zur Manipulation von Directoryeinträgen DEL, LOC, UNL, und REN können auch mehrere Dateien hintereinander bearbeiten. Um die Gefahr ungewollter Aktionen zu senken, wird ohne Angabe einer Option am Ende des Befehls nur die erste Datei mit übereinstimmenden Dateinamen bearbeitet. Mit der Option „/Q“ erfolgt für jeden übereinstimmenden Dateinamen eine

Rückfrage, die mit „[ Y ]“ oder einer beliebigen anderen Taste beantwortet werden kann, eine Bearbeitung der Datei erfolgt nur bei „[ Y ]“. Ein Abbruch ist mit der [ BREAK ]-Taste möglich.

Wer sich sicher ist, kann auch die Option „/A“ verwenden, die alle Dateien mit übereinstimmenden Dateinamen ohne Rückfrage bearbeitet.

Fehler, die bei der Bearbeitung auftreten können, werden im Klartext ausgegeben, allerdings erfolgt dadurch eine Rückkehr zum Freezer-Hauptmenü.

### 3.3 Kommandoübersicht

Die Kommandos werden hier nur tabellarisch aufgelistet, da sie an sich nicht neu und unbekannt sind. Anfänger können die Bedeutung der einzelnen Kommandos in jedem DOS-Manual nachschlagen, z. B. in dem, das jeder ATARI 1050 beiliegt.

DIR	Directory aller Dateien anzeigen
DIR Dateiname	Directory bestimmter Dateien anzeigen
DEL Dateiname	Datei löschen (Delete)
FMS	Formatieren Single Density
FME	Formatieren Enhanced Density
FMD	Formatieren Double Density <sup>1</sup>
LOC Dateiname	Datei sperren (Lock)
UNL Dateiname	Datei entsperren (Unlock)
REN Dateiname, Ersatzname	Datei umbenennen (Rename)
LOA Dateiname	Objektdatei laden (Load)
LOA Dateiname/N	nur Ladeadresse(n) anzeigen
LOA Dateiname, HexAdr	Objektdatei laden nach Adresse (Load), COM-header ignorieren
SAV Dateiname, HexAdr1, HexAdr2	Objektdatei erzeugen (Save von HexAdr1 bis einschließlich HexAdr2)
SAV Dateiname/N, HexAdr1, HexAdr2	Objektdatei ohne COM Header erzeugen (Raw Data)

<sup>1</sup> ein Formatieren in Double Density (180 KBytes per Seite) ist mit einer serienmäßigen ATARI 1050 Diskettenstation nicht möglich. Abhilfe bringt z. B. der Einbau eines 1050 TURBO, was die preiswerteste Möglichkeit ist, eine so hohe Speicherkapazität zu erhalten.

### 3.4 Besonderheiten mancher Kommandos

Einige der Kommandos haben Besonderheiten gegenüber herkömmlichen DOSs, die eine Verbesserung darstellen, oder die sich aus den besonderen Gegebenheiten der Funktion in einem Freezer zwangsläufig ergeben.

Bei Umbenennen von Dateien (REN) ist es möglich, auch im Ersatznamen beliebige Wildcards zu verwenden. Anstelle der Wildcards traten dann die entsprechenden Buchstabenfolgen aus dem vorherigen Namen der Datei.

Mithilfe dieser Eigenschaft ist es möglich, eine zusätzliche Gruppenbildung effizient zu bearbeiten, man ist also nicht auf den Primär- bzw. Sekundärnamen als Gruppenkriterium beschränkt.

Objektdateien können mit LOA zwar geladen werden. Der Ladebereich wird dabei angezeigt (bei Compoundfiles mehrere). Dabei steht der gesamte 64k-Adressraum des Rechners zur Verfügung,

d. h. es können sogar die Hardwareregister geladen werden (die zu diesem Zeitpunkt ja eingefroren sind). Gestartet wird das Programm aber NICHT, um eine Fehlbedienung der Memory-Management Logik des Freezers durch mehrstufig ladende Programme auszuschließen. Hängt man an einen Ladebefehl „/N“ an, wird die Datei nicht geladen, aber die Speicherbereiche werden angezeigt, in die die Datei geladen werden würde. Das ist nützlich, wenn man wissen möchte, wohin eine Datei geladen wird. Beispiel „LOA FONT.DAT/N“

Mit LOA kann man auch Rohdaten (ohne COM-Header) einlesen. Dazu muss man natürlich angeben, wohin die Daten geladen werden sollen. Beispiel: „LOA FONT.DAT,8000“

### **3.5 Fehlermeldungen des DOS**

Bei Verwendung des DOS (auch die Dateilade- und Speicherfunktionen des Freezers für eingefrorene Programme rufen das DOS auf) können bestimmte Fehler auftreten. Die Fehlermeldungen erfolgen im Klartext, aber in englischer Sprache (um auch die Märkte im benachbarten Europa und in den USA offen zu halten ist das leider auch bei deutschen Produkten notwendig). Wer kein Englisch kann findet hier die Bedeutungen:

FILE NOT FOUND	die Datei wurde nicht gefunden
FILE# MISMATCH	die (interne) Dateinummer ist fehlerhaft, die Dateistruktur ist wahrscheinlich beschädigt. Wenn möglich die anderen Dateien retten und Disk formatieren
BAD DISK I/O	das Kommando kann durch einen Busfehler oder Diskettenfehler oder durch eingeschalteten Schreibschutz nicht ausgeführt werden.
NO DRIVE	keine Diskettenstation ansprechbar
DISK FULL	die Diskette ist voll
FILE LOCKED	die Datei ist gesperrt
DIRECTORY FULL	das Directory ist voll (maximal 64 Dateien/Diskette)

## 4 Der Debugger

Ein Debugger wird von Maschinencode-Kennern benutzt, um Objektcode direkt im Speicher zwecks Fehlersuche, Veränderungen und Verbesserungen zu bearbeiten. In der Umgebung eines Freezers hat das besondere Vorzüge, allerdings auch ein paar prinzipbedingte Nachteile, die sich durch die richtige Arbeitsweise umgehen lassen.

Der größte Vorzug ist zweifellos der, dass mit dem eingefrorenen Systemzustand gearbeitet werden kann. Es entsteht dabei der Eindruck, quasi mit einem zweiten ATARI in den ersten, gestoppten ATARI „hineinsehen“ und ändern zu können, wobei der Stop jederzeit wieder aufgehoben werden kann. Debugger ohne Freezer haben hier mit schwierigen Problemen zu kämpfen, da sie allein durch ihren Eigenbedarf an RAM und Ein-/Ausgabeaktionen den Systemzustand laufend verändern. Der Fachausdruck dafür ist „trashen“, was für „zu Müll verwandeln“ steht.

Ein normaler Debugger braucht für seine eigene Funktion RAM und trasht die Informationen, die das zu bearbeitende Programm dort abgelegt haben kann. Selbst wenn der Debugger aufwendiger ausgestattet ist (durch eigenen RAM), trasht er immer noch die Hardwareregister von ANTIC, POKEY und GTIA. Die OS-Schattenregister ignorieren die Player-Missile Graphik und die Sounderzeugung und werden von vielen Programmen sowieso deaktiviert. Durch die getrashten Hardwareregister ist ein Wiederstart eines Programms, wenn überhaupt, in der Regel nur nach sehr mühseliger Kleinarbeit möglich. Abgesehen davon bieten Debugger ohne Freezer keine Möglichkeit, die (nicht lesbaren!) Hardwareregister auszulesen, man kommt also an die hineingeschriebenen Werte auch dann nicht heran, wenn das Register vom Debugger nicht getrasht worden ist.

Mit dem im TURBO-FREEZER XL/XE eingebauten Debugger kommt man an alle Informationen über den Systemzustand heran. Die Hardwareregisterinhalte (der echte Inhalt, nicht die zurückgegebenen Statuswerte) können im I/O Bereich gelesen und auf Wunsch geändert werden, ohne Abstürze befürchten zu müssen, da sich die Änderungen ja nur auf das eingefrorene Programm beziehen! Ebenso kann beliebig im gesamten RAM-Bereich gearbeitet werden (auch unter den OS-ROMs), ohne dass irgend etwas getrasht wird oder abstürzt.

Hinzu kommen verschiedene Vorzüge, die sich aus der Arbeitsumgebung des TURBO-FREEZER XL/XE ergeben. Mit den Lade- und Speichermöglichkeiten für eingefrorene Programme und den DOS-Funktionen können Änderungen ohne langes Speichern und Laden sofort ausprobiert und bei Bedarf auch mühelos wieder rückgängig gemacht werden. Widerspenstige Codemonster, die beim Laden Teile des DOS überschreiben und sich auf herkömmliche Weise kaum bearbeiten lassen verlieren ihre Schrecken.

Ein paar Nachteile gibt es allerdings auch. Dadurch, dass mit dem eingefrorenen Systemzustand gearbeitet wird, Änderungen also erst nach dem Auftauen Wirkung zeigen, wird es umständlich, z. B. in der Ram-Disk zu arbeiten. Nach dem Auftauen befindet sich das Programm wieder in voller Aktion, dadurch ist ein Start an einer anderen Stelle als dem Einfrierpunkt in einigen Fällen erst nach sorgfältigem Abschalten über die eingefrorenen Hardwareregister möglich.

Einige Einschränkungen wurden aus Platzgründen notwendig, so z. B. der Verzicht auf einen Blick in den OS-ROM und der Verzicht auf eine Möglichkeit, Compoundfiles unmittelbar zu erzeugen. Ob das als Nachteil empfunden wird, hängt vom Arbeitsstil ab. Der Autor selbst verfolgt OS-Aufrufe nie per Debugger, sondern anhand des OS-Listings und arbeitet lieber mit handlichen Einzelteilen als mit einem wüsten Codemonster, was ein Compoundfile eben ist. Schließlich ist es

keine Mühe, nach Abschluss aller Arbeiten und Tests die Teile mit einem DOS zu einem Compoundfile zu binden.

#### 4.1 Befehlssatz des Debuggers

Der auf das Nötigste beschränkte Befehlssatz orientiert sich stark an der Editor Assembler Cartridge von ATARI und entspricht daher vielen anderen Debuggern, so dass eine tabellarische Auflistung der Befehle genügt. Bei der Befehlseingabe dürfen keine Leerzeichen eingefügt werden, die Befehle müssen unmittelbar nach dem Prompt beginnen. Zur Eingabe steht nur die Eingabezeile zur Verfügung, dafür werden im Ausgabebereich ALLE Änderungen mitprotokolliert. Alle Eingaben verstehen sich in Hexadezimal. Durch Auslassen eines Werts können Speicherstellen oder Register ohne Änderung des Inhalts übersprungen werden.

Beispiel: C100<0A, , 4D ändert \$100 auf \$0A und \$102 auf \$4D. Danach steht der interne Adresszeiger (PC) auf \$103. Der Inhalt von \$101 bleibt unverändert.

Bei allen Kommandos können Adressangaben weggelassen werden, der Debugger verwendet dann den internen Adresszeiger bzw. einen der übersichtlichem Darstellung förderlichen Endwert. Die Ausgabe kann jederzeit mit den Tasten [ S ] (Stop) und [ Q ] (Weiter) angehalten bzw. fortgesetzt werden.

Q	zum TurboFreezer-Hauptmenü gehen (QUIT)
D Adr1,Adr2	Speicherinhalt anzeigen von Adr1 bis Adr2 (DISPLAY)
D Adr	8 Bytes anzeigen
D Adr,	128 Bytes anzeigen
I Adr1,Adr2	Speicherinhalt plus Zeichen im Screen Code anzeigen von Adr1 bis Adr2 (DISPLAY INTERNAL)
I Adr	8 Bytes anzeigen
I Adr,	128 bytes anzeigen
R	Register anzeigen
R<Byte,Byte	Register ändern
L Adr1,Adr2	Disassemblieren (LIST)
DL Adr1,Adr2	Display List Disassembler
G Adr	Returnadresse ändern (GO)
C Adr<Byte,Byte ...	Speicherinhalt ändern (CHANGE)
BM start,ende,ziel	Block verschieben (BLOCK MOVE) von start...ende nach ziel...ziel+(ende-start) Blöcke dürfen überlappen, z. B. ziel=start+1
BS start,ende,wert	Block mit Wert setzen (BLOCK SET)
BC start,ende,start2	Blöcke vergleichen (BLOCK COMPARE)
SR Sektornummer	Sektor lesen (SECTOR READ)
SW Sektornummer	Sektor schreiben (SECTOR WRITE)

SIO ...	SIO-Kommando ausführen
SIOR	Highspeed SIO Routine zurücksetzen (SIO RESET)
/Adr1,Adr2/Byte ...	Suchfunktion
V	Interrupt und VCOUNT zum Zeitpunkt des Einfrierens anzeigen
V<Byte	VCOUNT ändern
VEC	OS Vektoren anzeigen
HAT	Handler Tabelle (HATABS) anzeigen
DIO<Byte	Direct-IO-Modus aktivieren/deaktivieren (Achtung, gefährlich!)
a Kommando...	AtariSIO Remote Control
VER	Freezer Software Version anzeigen

Anmerkungen: Mit Ausnahme der Leerzeichen direkt nach dem Befehl müssen die hier aus Gründen der Lesbarkeit eingefügten Leerzeichen bei der Eingabe weggelassen werden. Der PC kann aufgrund der besonderen internen Behandlung nicht mit R< geändert werden, sondern nur mit G. Der Sektorpuffer für SW und SR ist mit dem internen Sektorpuffer für das DOS identisch und wird vom Memory-Manager nach \$D700 im eingefrorenen Adressraum gespiegelt, wo eine Bearbeitung mit dem Debugger erfolgen kann. Physikalisch ist er dort nicht vorhanden! Um einen Sektor von einem anderen Laufwerk als D1: zu lesen muss man einfach „Dx:“ direkt vor die Sektornummer stellen.

#### 4.1.1 Eingabe Hexadezimal, Dezimal, Interner Code

Adressen und Bytes werden üblicherweise in Hexadezimal eingegeben. Um Dezimalzahlen zu verwenden, muss die Adresse bzw. das Byte mit einem Prozent-Zeichen („%“) begonnen werden, z. B.: „C%710<%10“ ist das gleiche wie „POKE 710,10“ in Basic.

Bytes können auch in ATASCII oder als interner ATARI Screen Code eingegeben werden. Für ATASCII muss ein einfaches Anführungszeichen („'“), für Screen Code ein Klammeraffe („@“) vorangestellt werden.

Z. B.: „C0600<'H, 'a, '1, '1, 'o“ oder „C9C40<@S, @c, @r, @e, @e, @n“.

#### 4.1.2 Eingabe von Vektoren

Speziell für die Arbeit mit Vektoren (z. B. Display-List oder Interrupt-Vektoren) existiert eine weitere Variante um Adressen anzugeben: Beginnt man eine Adresse mit einem Stern ("\*"), so wird der (2 Byte) Vektor an der angegebenen Adresse ausgelesen und der Inhalt des Vektors als Adresse verwendet. Damit kann man z. B. sehr schnell die aktuell verwendete Display-List ausgeben lassen: „DL\*230“ oder „DL\*%560“.

#### 4.1.3 Display List anzeigen

Der Display List Disassembler gibt die ANTIC Befehle als Mnemonics aus (so wie im Datenblatt zum ANTIC beschrieben). „BLK x“ bedeutet x Leerzeilen, „CHR x“ Text Modus x, „MAP x“ Grafik Modus x, „JMP Adr“ ist ein Sprung zur angegebenen Adresse, „JVB Adr“ ein Sprung bei dem zuvor auf den Vertical Blank Interrupt gewartet wird. Neben dem Mnemonic werden, falls vorhanden, noch folgende Optionen ausgegeben: „LMS Adr“ heisst Memory Scan Counter laden,

„H“ und „V“ bedeuten horizontales bzw. vertikales Scrolling aktiviert und „I“ bedeutet Display List Interrupt auslösen.

#### 4.1.4 Suche

Die Suchfunktion kann auf verschiedene Arten verwendet werden: „/Adr/Byte1,Byte2 . . .“ sucht ab der angegebenen Adresse nach der Bytefolge. Sobald die Bytefolge gefunden wurde bricht die Suche ab und die Adresse wird ausgegeben. Dabei wird auch der interne Adresszähler auf die gefundene Adresse gesetzt, man kann also z. B. direkt nach der Suche mit „L“ die gefundene Speicheradressen disassemblieren.

Mit „/“ kann die Suche fortgesetzt werden.

Die Bytefolge kann aus bis zu 8 Bytes bestehen, was wohl in den meisten Fällen mehr als ausreichend ist. In der Bytefolge können auch einzelne Bytes ausgelassen werden, diese Bytes werden dann bei der Suche einfach ignoriert. „/1000/8D, ,D4“ liefert z. B. die erste Adresse ab \$1000 die ein Byte in den ANTIC schreibt (STA \$D4xx). Die Suchbytes können auch zusätzlich mit einer Bit-Maske versehen werden. Dazu muss direkt hinter dem Suchbyte ein „&“ gefolgt von der Bit-Maske angehängt werden, z. B.: „03&0F“. Diese Bit-Maske wirkt sich nun folgendermassen auf die Suche aus: Der Inhalt des Speichers wird mit der Bit-Maske AND verknüpft und danach mit dem Suchbyte verglichen.

Gibt man zusätzlich zur Startadresse eine Endadresse mit an, also „/Adr1,Adr2/Byte1,Byte2 . . .“, so werden alle Adressen im angegebenen Bereich ausgegeben, welche die Bytefolge enthalten. Die Suche stoppt also nicht bei der ersten gefundenen Adresse. Oft möchte man den gesamten Speicher durchsuchen, dafür gibt es eine besonders kurze Variante: „//Byte1,Byte2 . . .“ ist identisch zu „/0000,FFFF/Byte1,Byte2 . . .“.

Bei längeren Bytefolgen und einem großen Speicherbereich kann die Suche mehrere Sekunden dauern. Mit der [Break]-Taste kann die Suche jederzeit unterbrochen werden, mit [/] kann man sie wieder fortsetzen.

#### 4.1.5 SIO-Kommando

Mit dem „SIO“ Befehl können beliebige SIO-Kommandos (genauso wie über den SIO-Vektor \$E459) ausgeführt werden. Die Speicheradresse (DBUF) ist dabei fix auf den Sektorpuffer (\$D700) gesetzt, die Länge (DBYT) darf maximal \$0100 sein. Nach „SIO“ müssen die Werte für Device, Unit, Command, Direction, Timeout, Length, Daux folgen, mit der gleichen Bedeutung wie die Speicherzellen \$0300-\$030B im ATARI. Der SIO-Befehl merkt sich die zuletzt eingegebenen Parameter, so dass bei darauffolgenden SIO-Befehlen Teile der Parameter weggelassen werden können. Lässt man alle Parameter weg, so wird der letzte SIO-Befehl wiederholt. Zu Beginn sind die Parameter auf „Get Status“ von D1: gesetzt, sodass nichts schlimmes passieren kann wenn man einfach nur „SIO“ eingibt. Man sollte mit dem SIO-Befehl allerdings sehr vorsichtig umgehen und sicherheitshalber die Parameter mehrfach überprüfen, ansonsten hat man sehr schnell die falsche Diskette formatiert!

#### 4.1.6 Interrupt und VCOUNT anzeigen

Der „V“ Befehl gibt den Typ des Interrupts (NMI oder IRQ) an, mit dem der Freezer aktiviert wurde, zusammen mit der originalen Adresse des Interrupt-Vektors (den der Freezer dann auf die Freezer-Software „verbogen“ hat) und zusätzlich noch den Wert von VCOUNT beim Start der Freezer-Software sowie der errechnete Wert von VCOUNT zum Zeitpunkt des Interrupts. Dieser

Befehl wurde hauptsächlich zum Testen der Freezer-Software implementiert und dürfte für die meisten User wenig nützlich sein. Hier aber trotzdem eine kurze Erklärung was die Werte bedeuten und wo sie herkommen: Wenn der Freezer aktiviert wird, so blendet er das ROM mit der Freezer-Software ein und „verbiegt“ das High-Byte des Interrupt-Vektors sodass die Freezer-Software gestartet wird. Da das Low-Byte des Vektors nicht verändert wird, enthält die Freezer-Software eine Page voller „NOPs“. Es kann nun also passieren, dass der ATARI etliche NOPs abarbeiten muss bevor die eigentliche Freezer-Software gestartet wird und VCOUNT auslesen kann. Anhand des originalen Interrupt-Vektors versucht die Freezer-Software nun zu berechnen wie viel Zeit mit NOPs verbracht wurde und passt VCOUNT dementsprechend an, damit das Programm an exakt der Stelle fortgesetzt werden kann wo es unterbrochen wurde. Die Berechnung ist nicht 100% exakt und kann je nach Grafik-Modus etwas abweichen. Für den (sehr seltenen) Fall, dass der errechnete Wert nicht stimmen sollte und Probleme nach dem Auftauen auftreten, kann man nun per „V<Byte“ VCOUNT von Hand setzen.

Änderungen in den Hardwareregistern äußern sich normalerweise erst nach dem Auftauen, schließlich wird der eingefrorene Adressraum manipuliert, und nicht, wie bei konventionellen Debuggern, der reale Adressraum!

#### 4.1.7 Direct IO Modus

Für Profis gibt es beim neuen 2005er Freezer aber die Möglichkeit, dieses Verhalten abzuschalten, den „Direct IO Modus“. Mit „DIO<1“ wird er aktiviert, mit „DIO<0“ deaktiviert und mit „DIO“ wird der aktuelle Zustand ausgegeben.

**Achtung:** mit aktiviertem Direct-IO-Modus wird komplett am Freezer „vorbei“ gearbeitet und man hat gute Chancen sich dabei ordentlich in den Fuss zu schießen! Da der Freezer im Direct-IO nichts von den Änderungen an den Hardware-Registern mitbekommt, sollte man die selben Änderungen auch noch mit deaktiviertem Direct-IO-Modus vornehmen - ansonsten werden die Register beim Auftauen wieder überschrieben.

#### 4.1.8 Zugriff auf erweiterten Speicher / RAM-Disks

Die Debugger-Funktionen des TURBO-Freezers unterstützen alle Speichererweiterungen, die Speicherbänke via \$D301 (PORTB) umschalten. Standardmäßig wird der eingebaute Speicher des ATARI von \$4000-\$7FFF benutzt. Will man eine andere Speicherbank auswählen, benutzt man den PB-Befehl.

Benutzt man PB ohne Parameter, wird die derzeit benutzte Speicherbank angezeigt. Standard ist FF, an den normalen ATARI-Speicher wird zugegriffen.

Mit PB<Byte setzt man den Wert von PORTB für den Speicherzugriff. Beispiel: PB<E3 um die erste Speicherbank eines ATARI 130XE anzuwählen. PB<FF um wieder auf den normalen Speicher umzuschalten.

#### 4.1.9 Versionsnummer

Die Versionsnummer und das Datum der Freezer-Software kann auch im Debugger mit dem Kommando „VER“ ermittelt werden. Für den Fall, dass das Flash-ROM keine Freezer-Software enthält, wird anstatt der Versionsnummer „n/a“ ausgegeben.

Beispiel:

3.02 2008-05-11

#### 4.1.10 Vektoren

Mit dem „VEC“ Kommando kann man die wichtigsten OS-Vektoren aus den Pages 0 und 2 übersichtlich anzeigen lassen: zuerst den symbolischen Namen des Vektors, dann seine Adresse und dann seinen Wert.

Beispiel:

```
DOSINI      000C:      0000
DOSVEC      000A:      F223
VPRCED      0202:      C0CD
usw...
```

#### 4.1.11 Handler-Tabelle

Mit dem „HAT“ Kommando kann man die Einträge der Handlertabelle (\$031A-\$033A) anzeigen lassen. Zuerst wird die Adresse gezeigt, dann 3 Bytes, dann der Gerätebuchstabe (bzw. „---“ wenn der Eintrag leer ist oder Null) und dann die Adresse der Gerätehandlertabelle.

Beispiel:

```
031A      50 30 E4      P:      E430
031D      43 40 E4      C:      E440
0320      45 00 E4      E:      E400
usw...
```

#### 4.1.12 Hilfe

Wenn man die [HELP]-Taste drückt, wird eine kurze Hilfe angezeigt. Durch Drücken der Tasten [1], [2], [3], ... springt man zu den einzelnen Hilfeseiten, [DEL] springt eine Seite zurück, [Leertaste] oder [HELP] eine Seite weiter. Jede andere Taste beendet die Hilfe.

#### 4.1.13 Speicherkartierung / Memory Map

Drückt man die Taste [M] zeigt der Debugger eine Kartierung des Speichers nach benutzten und unbenutzten Speicherseiten (Pages). Enthält eine Seite/Page nur Nullen wird ein Punkt ‚.‘ angezeigt, enthält sie mindestens ein anderes Zeichen, wird ein Stern ‚\*‘ angezeigt. Dies ist zum Beispiel nützlich, wenn man eine leere Seite/Page für den Boot-Loader sucht (siehe Kapitel 2.3.1).

#### 4.1.14 ATARI-SIO Direktkommando

Mit dem „a“ Kommando können Remote Control Kommandos an ATARISIO gesendet werden. Damit lässt sich ATARISIO komplett fernsteuern ohne dass ein eigenes Programm geladen werden muss. Nähere Informationen dazu finden sich in der ATARISIO Anleitung.

### 4.2 Kommandos für die Cartridge-Emulation

Ein Anzahl von Kommandos, die alle mit „K“ beginnen, ist vorhanden um die eingebaute Cartridge-Emulation zu kontrollieren (Diese Kommandos sind identisch mit dem händischen Ändern der Konfigurations-Speicherzellen ab \$D500, aber viele Leute haben lieber leicht zu merkende Menmonics als Hexadressen).

Hieri ist eine Liste der Kommandos und ihre zugehörigen Adressen (\$D5xx).

- KON: CartEmu aktivieren (\$D581)
- KOF: CartEmu deaktivieren (\$D580)

- KM <Modus>: setze den CartEmu Modus (\$D588...\$D58A). <Modus> muss eine der folgenden Optionen sein: '8' (8k), '1' (16k), 'O' (OSS) oder 'S' (SDX)  
Wenn der Modus auf OSS gesetzt wird, wird die erste OSS Speicherbank (\$D500) automatisch selektiert. Wenn als Modus SDX gesetzt wird, wird die erste SDX Speicherbank (\$D5E0) selektiert.
- KB <bank>: setze CartEmu Speicherbank (\$D540...\$D57F)
- KOS <bank>: setze OSS Bank. <bank> muss '1', '2', '3' oder 'O' (off/aus) sein. Der Befehl resultiert in einem Schreibzugriff auf \$D500, \$D509, \$D501 und \$D508.
- KSD <bank>: setze SDX bank ('0'...'7' oder 'O' für off/aus). Schreibt nach \$D5E0...\$D5E7 (0-7) oder \$D5E8 (off/aus)
- KRA: setze Speicherquelle auf Freezer-RAM (\$D587)
- KRO: setze Speicherquelle auf Freezer Flash ROM (\$D586)
- KKE: setze „keep settings“ (behalte Einstellungen bei Reset) auf an/on (\$D583)
- KRE: setze „keep settings“ (behalte Einstellungen bei Reset) auf aus/off (\$D582)
- KWE: ermöglicht Schreiben (\$D585) / Schreibschutz aus. Dies ist nur nützlich, um den Inhalt des Freezer-RAMs zu ändern. Das Flash-ROM kann nicht vom Debugger aus geändert werden!  
Hinweis: nicht vergessen, nach dem Ändern des Freezer-RAMs einen Schreibschutzbefehl zu geben, damit nicht versehentlich etwas überschrieben/gelöscht wird!
- KWP: schaltet „Schreiben“ aus / Schreibschutz ein (\$D584)
- KDE: setze die CartEmu auf Standardwerte („default settings“), als wenn der Computer mit eingeschaltetem CartEmu Schalter hochgefahren wird. Folgende Einstellungen werden gesetzt:
  - Schreiben ausgeschaltet („writes disabled“)
  - behalte Einstellungen bei Reset eingeschaltet („keep settings on“)
  - Cartridge Emulation eingeschaltet
  - Speicherquelle Flash ROM
  - Modus 8k
  - Bank 62 (\$3E) – die Speicherbank mit der CartridgeEmulation

## 5 Der Oldrunner

Der ATARI ist der einzige Homecomputer der 8-Bit Klasse, der ein sauber gegliedertes und durchdachtes OS enthält. Es ermöglicht jederzeit Änderungen, Erweiterungen und Verbesserungen, ohne dass vorhandene Programme angepasst werden müssten. Daher konnte ATARI bei der XL-Serie ein neues, leistungsfähigeres OS verwenden.

Ärgerlicherweise tauchten dann doch Programme auf (Jahrgang 1980-1983), die auf der XL-Serie nicht funktionierten. Die Schuld liegt bei den Programmierern der betroffenen Programme, die die offiziellen Programmierrichtlinien nicht eingehalten hatten.

Um die unverträglichen Programme trotzdem auch auf den XL-ATARIs laufen lassen zu können, wurden sogenannte Translator-Disks herausgebracht, die das OS-ROM abschalten und das alte OS kurzerhand in dem darunter befindlichen RAM ablegen. Diese Softwarelösung funktioniert zwar in den meisten Fällen, ist aber wegen des dauernden Laderei unbequem und zeitraubend. Ideal ist nur eine Hardwarelösung, bei der das alte OS auf einem EPROM abgelegt ist und damit jederzeit unveränderlich zur Verfügung steht. Nur diese Lösung funktioniert mit allen unverträglichen Programmen. Leider erforderten solche „Oldrunner“ bisher immer Eingriffe in den ATARI und einen dicken Geldbeutel (Preise bis ca. 100 DM!).

Mit dem TURBO-FREEZER XL/XE ist es nun erstmals möglich, einen Oldrunner ohne Eingriffe in den ATARI zu realisieren. Die Memory-Management-Logik macht's möglich!

Mit dem Schalter am TURBO-FREEZER XL/XE, der mit „OS“ beschriftet ist, kann der Oldrunner ein- und ausgeschaltet werden. Bei der XL-Version ist es der Schalter Richtung Platinen-Mitte, bei der XE-Version der Schalter Richtung Platinen-Rand. Eingeschaltet ist er, wenn der Schieber näher am Rand der Platine steht (XL-Version) bzw. der Schieber vom ATARI weg zeigt (XE-Version). Es ist ratsam, diesen Schalter nicht bei eingeschaltetem ATARI zu betätigen (da die Initialisierung im RAM nicht zum jeweils „anderen“ OS passt, erfolgt ein Systemabsturz).

XL: Schalter „OS“ (in der Platinenmitte) Richtung Rand = „Oldrunner AN“  
XE: Schalter „OS“ (am Platinenrand) Richtung hinten = „Oldrunner AN“

Aufgrund einiger Umstände ist es ratsam, nur dann im Oldrunner-Modus zu arbeiten, wenn das Programm anders nicht läuft. Im Oldrunner-Modus gibt es kein eingebautes BASIC, keine Ram-Disk und auch keinen nichtdestruktiven RESET! Die [RESET]-Taste beim XL/XE bewirkt im Oldrunner-Modus immer(!) einen Kaltstart. Die entsprechende Taste beim „alten“ ATARI hieß [SYSTEM RESET] und bewirkte einen NMI-Interrupt. Sie kann zwar ohne großen Aufwand nachgerüstet werden (siehe unter 1.4), das erfordert aber einen Eingriff in den ATARI und ist nur Experten vorbehalten. Unbedingt nötig ist diese Taste ohnehin nicht, da die meisten unverträglichen Programme Spiele sind, die den SYSTEM RESET in vielen Fällen auf Kaltstart oder Absturz programmiert haben. Es handelt sich dabei um einen lächerlichen Versuch, die „Cracker“ zu ärgern, der aber seinen Zweck verfehlt hat und nur den Anwender ärgert.

## 6 Die Cartridge-Emulation

Mit der Cartridge-Emulation bietet der TURBO-FREEZER XL/XE ein weiteres, sehr mächtiges Feature: der TURBO-FREEZER XL/XE kann Standard 8k, Standard 16k, 16k OSS Bankswitching Module (16k Module, die nur 8k Speicher belegen, wie z. B. Mac/65, Action und Basic XL/XE) und sogar das 64k SpartaDosX Modul (64k Modul, das nur 8k Speicher belegt) emulieren! Für die Cartridge-Emulation steht der gesamte freie Speicher im Flash-ROM (472k) sowie im Freezer-RAM (112k) zur Verfügung. Das heisst, mit dem TURBO-FREEZER XL/XE hat man ständig Zugriff auf bis zu 59 verschiedene Module!

Die Cartridge-Emulation ist nicht nur für alle diejenigen interessant, die häufig mit verschiedenen Steckmodulen hantieren und das ewige Modul-Wechseln satt haben (was nebenbei bemerkt auch den Modulschacht im ATARI stark beansprucht), sondern auch für diejenigen, die selber Module entwickeln. Da die Daten für die Steckmodule nicht nur im Flash-ROM sondern auch im Freezer-RAM abgelegt werden können, sind Änderungen an den Modul-Daten in Sekundenschnelle erledigt!

Unter <http://turbofreezer.horus.com/> steht eine angepasste Version der AtariMax MaxFlash Software bereit. Damit können beliebige Disk-Images sowie COM/EXE Files in Modul-Images konvertiert werden. Als spezielle Erweiterung unterstützt der TURBO-FREEZER XL/XE sogar mehrere Disk Images gleichzeitig und der restliche freie Speicherbereich des Flash-ROMs kann nach wie vor für die Emulation von Modulen verwendet werden. Damit ist der TURBO-FREEZER XL/XE bedeutend flexibler als das AtariMax MaxFlash-Modul!

Um die Cartridge-Emulation zu aktivieren, muss nur der mit „CA“ beschriftete Jumper gesetzt werden bzw. der Schalter umgelegt werden. Bei der XL-Version befindet sich der Schalter (bzw) Jumper am Rand der Platine, bei der XE-Version liegt er Richtung Platinenmitte. Die Cartridge-Emulation ist aktiv, wenn der Jumper gesetzt ist oder der Schieber Richtung Platinenrand zeigt (XL-Version) bzw. vom ATARI weg zeigt (XE-Version).

XL: Schalter „CA“ (am Platinenrand) Richtung Rand = „Cartridge-Emulation AN“

XE: Schalter „CA“ (in der Platinenmitte) Richtung hinten = „Cartridge-Emulation AN“

Wie beim Oldrunner empfiehlt es sich den Schalter nur bei ausgeschaltetem ATARI zu ändern, da ansonsten der ATARI abstürzt.

Nach dem Einschalten erscheint bei aktivierter Cartridge-Emulation ein kleines Auswahlmenü, mit dem sich die Cartridge-Emulation vollständig konfigurieren lässt. Im Auslieferungszustand des TURBO-FREEZER XL/XE ist das Flash-ROM übrigens (bis auf die Freezer-Software und den TURBO-FREEZER-System-Disk-Writer) leer. Im nächsten Kapitel wird beschrieben, wie das Flash-ROM sowie das Freezer-RAM mit Modul-Daten befüllt werden können.

Im Auswahlmenü stehen folgende Konfigurationsmöglichkeiten bereit:

Mit der [M]-Taste kann der Cartridgetyp zwischen 8k, 16k, OSS, SDX, PDOS und OFF gewechselt werden. „PDOS“ ist auch über die Taste [P] erreichbar, „OFF“ über die Taste [O]. PDOS stellt ein vollwertiges MyPicoDos zur Verfügung, nach dem Start von MyPicoDos wird die Cartridge-Emulation automatisch deaktiviert. Das heisst, man kann jederzeit ein sehr mächtiges GameDOS starten ohne extra die Diskette damit initialisieren zu müssen oder eine Bootdisk zu suchen! „OFF“ bedeutet, dass die Cartridge-Emulation deaktiviert ist.

Unter „BANK“ ist die aktuell ausgewählte Banknummer zu sehen. Mit den Tasten [ 0 ]–[ 9 ] sowie [ Delete ] kann man hier eine (dezimale) Banknummer von 0-59 eingeben.

Mit der [ S ] Taste kann zwischen RAM und ROM als Quelle für die Cartridge-Emulation gewählt werden.

Die nächste Option, „KEEP (settings)“, die mit der [ K ] Taste zwischen „YES“ und „NO“ umgeschaltet werden kann, legt fest was bei einem Druck auf [ RESET ] passiert. Steht „Keep“ auf „YES“, so bleiben alle Einstellungen auch nach einem Reset erhalten. Steht „Keep“ auf „NO“, so werden bei einem Reset alle Einstellungen zurückgesetzt und man landet wieder im Cartridge-Auswahl Menü.

Mit der [ B ] Taste schließlich kann man festlegen, ob nach dem Aktivieren der Cartridge-Emulation ein Kaltstart (COLD) oder ein Warmstart (WARM) durchgeführt werden soll. Für gewöhnlich sollte man diese Einstellung auf „COLD“ belassen, da ansonsten das Modul nicht korrekt initialisiert werden kann. Die Einstellung „WARM“ ist dann sinnvoll, wenn man versehentlich „KEEP“ auf „NO“ gesetzt hat und nun im Auswahlmenü gelandet ist. Stellt man die Bank-Nummer und den Cartridgetyp auf die vorherigen Werte und setzt „BOOT“ auf „WARM“, so bleiben alle Daten im Speicher erhalten und man kann die Arbeit wie nach einem normalen Reset fortsetzen.

Hat man alle Optionen wie gewünscht eingestellt, so kann die aktuelle Konfiguration mit der [ Return ] Taste aktiviert werden!

[ M ]	Modus wechseln 8k, 16k, SDX, PDOS, OFF
[ O ]	Modus OFF direkt wählen
[ P ]	Modus PDOS direkt wählen
[ 0 ]–[ 9 ], [ Delete ]	Bank-Nummer eingeben
[ S ]	RAM/ROM als Quelle für Cartridge-Emulation auswählen
[ K ]	Keep-Settings YES/NO auswählen
[ B ]	Bootverhalten COLD- oder WARM-Start wählen
[ Return ]	Cartridge-Emulation starten

Um mit der Cartridge-Emulation umgehen zu können muss man ein paar wenige Fakten über ihre Funktionsweise verstehen: Das Flash-ROM ist intern in 8k große „Bänke“ unterteilt. Da das Flash-ROM 512k groß ist gibt es insgesamt 64 8k Bänke (Banknummern 0-63). Die obersten 40k des Flash-ROMs sind von der Freezer-Software belegt und können somit nicht für die Cartridge-Emulation verwendet werden. Das heisst, für die Cartridge-Emulation stehen die Bank Nummern 0 bis 58 zur Verfügung. Beim Freezer-RAM sind die obersten 16k für die Freezer-Software reserviert, es stehen hier also die Bänke 0 bis 13 zur Verfügung. Wenn man einen Snapshot in das Freezer -RAM abspeichert, so werden die untersten 64k des RAMs dafür verwendet, das heisst es sind nur mehr die Bänke 8-13 verfügbar!

Beim Befüllen des Flash-ROMs (bzw. Freezer-RAMs) muss weiters noch folgendes beachtet werden: Standard 16k Module und 16k OSS Module müssen jeweils an einer geraden Bank Nummer beginnen (mit anderen Worten: an einer 16k Grenze), SpartaDosX Module müssen an einer durch 8 teilbaren Banknummer beginnen (also an einer 64k Grenze).

Die Bezeichnung „Cartridge-Emulation“ hat übrigens auch einen tieferen Hintergrund: „Richtige“ Module können über eine spezielle Leitung am Modulschacht dem ATARI mitteilen, ob ein Modul

eingelegt ist oder nicht. Der TURBO-FREEZER XL/XE wird aber nun am PBI angeschlossen, und dort steht diese Leitung leider nicht zur Verfügung. In den meisten Fällen stellt dies aber kein Problem dar. Die einzige Ausnahme sind Bankswitching Module, die auch komplett abgeschaltet werden können. Der ATARI vergleicht im VBI regelmäßig den Wert von TRIG3 (\$D013) mit dem in GINTLK (\$03FA) gespeicherten Wert. Sind die Werte unterschiedlich, so bedeutet dies, dass ein Modul eingelegt oder herausgenommen wurde und der ATARI bleibt in einer Endlosschleife hängen. Beim nächsten Reset wird dann ein Kaltstart ausgeführt.

Ein Bankswitching-Modul muss also beim An- bzw Abschalten des kompletten Moduls auch den Wert in GINTLK ändern. Am einfachsten ist es, den Wert von TRIG3 auszulesen und in GINTLK abzulegen. Leider machen das nicht alle Module so, einige setzen den Wert von GINTLK direkt auf 0 oder 1. Da der Wert von TRIG3 aber bei der Cartridge-Emulation konstant bleibt, hängt sich der ATARI auf.

Damit auch solche Module mit der Cartridge-Emulation funktionieren, muss der Programmcode im Modul gepatcht werden. Auf der Diskette gibt es dazu das Programm „PATCH.COM“. Es kann ROM-Images von Basic XE sowie SpartaDosX in den Versionen 4.19, 4.21 und 4.22 so ändern, dass sie mit der Cartridge -Emulation funktionieren. Basic XL, MAC/65 und ‚Action!‘ funktionieren übrigens ohne Modifikation.

## 7 Programmieren des Flash-ROMs

Die Software des TURBO-FREEZER XL/XE ist nicht, wie sonst üblich, in einem EPROM sondern in einem Flash-ROM abgelegt. Der große Vorteil des Flash-ROMs besteht darin, dass es direkt vom ATARI aus neu programmiert werden kann! Damit können jederzeit Software-Updates eingespielt werden und der unbelegte Teil des Flash-ROMs mit Modul-Daten befüllt werden, ohne dass ein spezielles Programmiergerät notwendig ist! Das Flash-ROM am TURBO-FREEZER XL/XE kann übrigens bis zu 1.000.000 Mal neu programmiert werden, was für die meisten Experimente wohl ausreichend sein sollte. Für den unwahrscheinlichen Fall, dass jemand die 1 Million Zyklen erreicht und das Flash-ROM kaputt geht braucht der Freezer nicht eingeschickt zu werden. Das Flash-ROM ist gesockelt und man kann einfach ein neues AMD 29F040 einsetzen. Wer mit 128k Flash-ROM auskommt, kann auch ein EEPROM vom Typ SST 29EE010 oder Winbond 29EE011 einsetzen. Diese ICs wurden häufig als BIOS Chips auf PC Mainboards verwendet und können leicht von defekten Mainboards „ausgeschlachtet“ werden. Andersherum kann man den TURBO-FREEZER sogar als Programmiergerät für diese BIOS Chips verwenden, wenn beim Flashen des PC BIOSes etwas schiefgegangen ist! Zusätzlich zu den drei genannten ICs wird auch noch der AMD 29F010 unterstützt, durch Erweiterungen an der Flasher Software sind sogar noch mehr Typen möglich!

Das Flash-ROM am TURBO-FREEZER XL/XE ist normalerweise gegen das Überschreiben geschützt. Um die Schreibfunktion auf das Flash-ROM freizuschalten, muss der mit „FL“ beschriftete „Flash Write“ Jumper gesetzt werden. Der Jumper befindet sich in der Mitte zwischen den Schaltern für den Oldrunner und die Cartridge-Emulation. Ist der Jumper offen, so sind das Flash-ROM und das Freezer-RAM schreibgeschützt. Ist der Jumper geschlossen, können Flash-ROM und Freezer-RAM beschrieben werden.

XL/XE: Jumper „FL“ (zwischen den Schaltern) geschlossen = „Flash schreiben möglich“

Es ist sehr unwahrscheinlich, dass irgendeine Software das Flash-ROM am Freezer versehentlich überschreibt. Um das Flash-ROM zu programmieren muss eine spezielle Sequenz zum Flash-ROM geschickt werden, ansonsten werden Schreibzugriffe einfach ignoriert. Deshalb ist es recht ungefährlich den Flash Write Jumper geschlossen zu lassen. Wer auf Nummer sicher gehen will, sollte den Jumper aber gleich nach dem Programmieren abziehen.

Um das Flash-ROM bzw das Freezer-RAM zu programmieren, muss zuerst der Flash Write Jumper gesteckt werden und dann muss das Programm „FLASH.COM“ von der TURBO-FREEZER-Diskette gestartet werden. Wenn der Jumper nicht gesteckt ist, kann das Programm das Flash-ROM nicht finden und es wird eine Fehlermeldung ausgegeben. In diesem Fall kann man den Jumper auch nachträglich stecken und dann bei „restart program?“ mit [y] antworten.

Das eingesetzte Flash-ROM vom Typ AMD 29F040 ist intern in 8 Blöcke zu je 64k aufgeteilt. Diese Blöcke können unabhängig voneinander neu programmiert werden. Das bedeutet, eine Änderungen z. B. an den Daten für die Cartridge-Emulation beeinflusst nicht die Freezer-Software (und umgekehrt). Die Aufteilung in 64k Blöcke hat leider auch einen kleinen Nachteil: es ist nicht möglich, nur einen Teil eines 64k Blocks neu zu programmieren. Um den gesamten Platz im Flash-ROM ausnützen zu können, muss man also z. B. vorher mehrere 8k ROM Images zu einem großen Image zusammenfassen. Für das Freezer-RAM gilt die Einschränkung natürlich nicht, hier kann jede 8k Bank separat programmiert werden!

Grundsätzlich darf das Image-File bis zu 512k groß sein. Die einzige Einschränkung ist, dass die Größe ein Vielfaches von 8k ist. Ist das Image-File z. B. nur 16k groß, so werden die restlichen 48k des 64k Blockes einfach gelöscht.

Nach dem Start gibt das Flashprogramm folgende Informationen aus:

- Typ des Flash-ROMs
- Größe der Bänke (8k oder 64k)
- die aktuelle Versionsnummer inklusive Erstelldatum der Freezer-Software.

Für den Fall, dass das Flash-ROM keine Freezer-Software enthält, wird anstatt der Versionsnummer „n/a“ ausgegeben. Der Versionsnummer kann auch im Debugger mit dem Kommando „VER“ ermittelt werden.

Im Flashprogramm hat man nun folgende Optionen zur Auswahl:

- |                                   |  |
|-----------------------------------|--|
| 1) Program flash ROM              | damit werden ein oder mehrere Blöcke des Flash-ROMs neu programmiert.        |
| 2) Write flash ROM to file        | der Inhalt des ROM wird in ein File auf Disk geschrieben.                    |
| 3) Erase flash ROM                | damit wird das gesamte Flash-ROM (inklusive der Freezer Software!) gelöscht. |
| 4) Program Freezer RAM            | das Freezer-RAM für die Cartridge-Emulation beschreiben.                     |
| 5) Write Freezer RAM to file      | der Inhalt des RAM wird in ein File auf Disk geschrieben.                    |
| 6) Erase Freezer RAM              | der Inhalt des Freezer RAMs wird gelöscht                                    |
| 7) Start Cartridge emulation menu | damit wird das Cartridge Auswahl Menü aus dem Freezer Flash-ROM gestartet.   |
| 0) Exit Program                   | beendet das Programm   |

Beim Programmieren von Flash-ROM bzw. Freezer-RAM geht man im allgemeinen wie folgt vor: Als erstes wird man nach der ersten zu programmierenden Banknummer gefragt. Diese Banknummer muss unbedingt an einer Blockgrenze beginnen, die aktuelle Blockgröße wird unterhalb des erkannten Flash Typs angezeigt.

Danach muss man den Filenamen des ROM-Images eingeben. Dieses ROM-Image darf nur die zu programmierenden Daten enthalten (keinen COM-Header oder ähnliches) und die File-Länge muss durch 8k teilbar sein.

Es werden dann die einzelnen 8k Bänke des Flash-ROMs (oder des RAMs) der Reihe nach programmiert, so lange bis entweder das Ende des Imagefiles oder die letzte Banknummer erreicht ist.

Nach dem Programmieren des Freezer-ROMs oder des Freezer-RAMs kann entweder durch Drücken der Taste [R] das Cartridge-menü gestartet werden oder ins Menü von FLASH.Com zurückgekehrt werden (jede andere Taste).

Die Speicherbanknummer wird auf die Start-Bank gesetzt und das Programm fragt nach dem Modus der Cartridge-Emulation. Wird dabei eine ungültige Taste gedrückt, wird die Funktion abgebrochen.

Für den Fall dass man (versehentlich) die Freezer-Software im Flash-ROM überschrieben hat und der Freezer somit nicht mehr korrekt funktioniert, oder für den Fall dass man eine neue Version der Freezer-Software einspielen möchte, geht man wie folgt vor: Beim original Flash-ROM muss die Datei FREEZER.ROM von der Freezer-Software-Disk in die Bänke 56-63 programmiert werden. Man muss als Start Bank also „56“ eingeben und als Dateinamen „FREEZER.ROM“. Das war's schon! Hat man das Flash-ROM durch einen der angegebenen 128k Typen ersetzt, so muss man die Datei „FRZ128.ROM“ in die Bänke 12-15 programmieren.

Da die Freezer-Software zur Zeit 40k groß ist, ist knapp die erste Hälfte des FREEZER.ROM Files (also die Bänke 56-58) unbenutzt. Wer will, kann mit ein paar Tricks auch noch diesen freien Teil für die Cartridge-Emulation benutzen: Dazu muss man ein Image erzeugen, das in den ersten 24k die gewünschten Daten für die Cartridge-Emulation enthält. Die zweiten 40k müssen nun die Freezer-Software enthalten (entweder aus FRZ128.ROM oder aus der zweiten Hälfte von FREEZER.ROM).

## 7.1 Hinweise zur angepassten Maxflash Software

Die gesamte Anleitung zur AtariMax Maxflash-Software gibt es im Internet unter <http://www.atarimax.com/>. Im folgenden werden nur die Änderungen für den TURBO-FREEZER XL/XE beschrieben. Die original Software kann übrigens nicht mit dem TURBO-FREEZER XL/XE verwendet werden, allerdings unterstützt die angepasste Software sowohl die Maxflash Cartridges als auch den Freezer!

Installation der Software: anstatt des TASM Assemblers, den es nur für DOS/Windows gibt, verwendet die angepasste Software den ATasm als Assembler. Da ATasm frei im Sourcecode erhältlich ist, und sowohl unter Windows als auch unter Linux funktioniert, kann nun endlich auch die Maxflash Software unter Linux eingesetzt werden! ATasm sollte in Version 1.05b (oder neuer) installiert sein, mit älteren Versionen ist sie nicht getestet.

Speziell für den TURBO-FREEZER XL/XE gibt es einige neue Optionen:

-FRZ32KB -FRZ48KB -FRZ64KB -FRZ80KB -FRZ96KB -FRZ128KB  
-FRZ192KB -FRZ256KB -FRZ320KB -FRZ384KB -FRZ448KB

legen fest, wie groß das Cartridge Image werden soll. Man muss unbedingt eine dieser Optionen angeben! Die Grösse (32/64/...) KB ist übrigens in kBytes, die Werte für die Maxflash Cartridge (1MB, 4MB, 8MB) waren in kBit!

Möchte man mehrere Images im Freezer unterbringen, so muss man explizit die Banknummer angeben, ab der das Image im Freezer abgelegt wird. Standardmässig wird die erste Bank (Bank Nummer 0) verwendet, mit der Option -FRZBANK=x kann die Banknummer auf einen anderen Wert gesetzt werden.

Hier zwei Beispiele:

```
perl maxflash.pl -EXEPACKER -FRZ64KB test.bin ../test/
```

Erzeugt ein 64KB Image für die Freezer Bänke 0-7. Das Image „test.bin“ kann nun mit der Flasher Software ab Banknummer 0 in das Flash (oder auch das RAM) programmiert werden.

```
perl maxflash.pl -EXEPACKER -FRZ64KB -FRZBANK=8 test.bin ../test/
```

Erzeugt ein 64KB Image für die Bänke 8-15. Das Image muss ab Banknummer 8 ins Flash programmiert werden.

Folgende Optionen sind in der TURBO-FREEZER XL/XE -Maxflash-Software sinnlos und werden nicht unterstützt:

-BIN2ATR	(U)	Make existing BIN file into flash programming image.
-BIN2ALL	(U)	Make existing BIN file into CART/ATR images.
-FLASHER	(*)	Make bootable ATR image for programming MAXFLASH cart
-8MB	(*)	Create image for 8Mb cartridge. (default is 1MB)
-NOBIN	(*)	Delete raw binary after processing.

Im Cartridge-Auswahlmenü muss für Maxflash Images als Typ „8K“ angegeben werden und die Bank auf die entsprechende Startbank eingestellt werden.

## 8 Für Experten: Technische Details

Herzstück des Freezers ist der CPLD, der die gesamte Logik des Freezers (und der Cartridge-Emulation) enthält. Einerseits ist er mit dem ATARI verbunden, andererseits steuert er auch das RAM und Flash-ROM am Freezer an. Auch die Jumper bzw. Schalter und der Freezer-Knopf auf der Platine sind direkt mit dem CPLD verbunden.

Von der Hardwareseite her sind RAM und Flash-ROM in 4k große Bänke unterteilt. Üblicherweise werden RAM und Flash allerdings in 8k großen Bänken angesprochen, es gibt nur zwei Spezialfälle bei denen wirklich 4k Bänke verwendet werden: zum einen bei der Emulation von OSS-Bank-switching Cartridges, zum anderen bei aktiviertem Freezer, wenn von der Freezer-Software aus auf das RAM am Freezer zugegriffen wird. Ansonsten erlaubt die Logik im CPLD immer nur die Auswahl von 8k Bänken.

Für das Shadowing der Hardware Adressen im ATARI (wie z. B. ANTIC, GTIA etc, doch dazu später mehr) gibt es einen weiteren Spezialfall. Die Adressen der ICs sind nicht vollständig adressiert, was der Freezer natürlich auch simulieren muss. Deshalb enthält der CPLD eine spezielle Logik, die beim Hardware Shadowing gegebenenfalls die Adressleitungen A4 und A5-A7 zum RAM hin ausblenden kann. Im Normalfall werden diese Adressleitungen direkt zum RAM durchgeschaltet, nur beim Hardware Shadowing werden sie deaktiviert.

Wenn das RAM oder das Flash-ROM am Freezer in den ATARI eingeblendet werden soll, muss natürlich das interne RAM/ROM im ATARI deaktiviert werden. Das geschieht durch einen kleinen Trick: Der Freezer zieht hierbei den (eigentlich als Ausgang gedachten) Refresh-Pin auf Low, worauf der ATARI glaubt, dass der Antic einen Speicher-Refresh durchführt und deaktiviert daraufhin den eingebauten Speicher.

Laut Bernhard Engl (heute hauptberuflicher Chipdesigner) ist es bei der damaligen NMOS-Depletion-Load-Technologie ungefährlich, einen „High“-Pegel extern auf „Low“ zu ziehen, weil das bei NMOS ohnehin so geschieht. Wäre der ANTIC in CMOS-Technologie gefertigt gewesen, dann wäre das möglicherweise gefährlich gewesen und hätte den ANTIC beschädigen können, und es hätte niemals einen TURBO-FREEZER XL/XE gegeben. Seine frühen Freezer-Konzepte ab 1984 (die tatsächlich realisiert worden waren) kannten diesen Trick mit dem Refresh noch nicht und sie wurden dadurch realisiert, einen kompletten ATARI mit einer speziellen Speicherselektionslogik komplett neu aufzubauen. Das war 1985. Erst als er den Trick mit dem Refresh entdeckte, der ATARI 800XL mit seinem parallelen Bus genügend verbreitet war, und die brandneuen GALs es erlaubten, die aufwendige Freezer-Logik auf wenige Chips zu komprimieren, erst dann war es möglich, den Freezer als Produkt auf den Markt zu bringen. Das war 1987. Drei Jahre nach den ersten Versuchen. Das Produkt zu entwickeln hat dennoch mehr als ein Jahr gebraucht (jedoch kein Mannjahr, denn nebenbei musste ja noch Elektrotechnik studiert werden). Zum Vergleich: der 6502 Mikroprozessor selbst hat bloß drei Mannjahre an Entwicklung verschlungen (persönliche Korrespondenz von Bernhard Engl mit Bill Mensch). Heutige PC-Prozessoren enthalten Tausende von Mannjahren an Entwicklung.

Nun zur Beschreibung der einzelnen Grundfunktionen im Freezer: Die eigentliche Freezer-Logik und die Cartridge-Emulation sind unabhängig voneinander implementiert. Nur quasi ganz zum Schluss, wo es darum geht ob RAM oder das Flash-ROM angesprochen werden und ob das interne ATARI RAM/ROM deaktiviert werden sollen sind diese beiden Teile zusammengeführt. Das bedeutet, dass auch von der Freezer-Software aus die Cartridge-Emulation gesteuert werden kann und man z. B. mit dem Freezer-Debugger eine emulierte Cartridge disassemblieren kann.

## 8.1 Details der Cartridge-Emulation

Die Grundkonfiguration der Cartridge-Emulation wird durch den Jumper „Flash Write“ und den Schalter „CartEmu“ gesteuert. Sind sowohl der Schalter als auch der Jumper offen, so ist die Cartridge-Emulation komplett deaktiviert. Somit ist es möglich, normale Cartridges im Modulschacht des ATARIs zu verwenden ohne dass sich diese mit der Cartridge-Emulation in die Quere kommen.

Ist nur der „Flash Write“ Jumper gesetzt, so wird beim Start des ATARIs die Cartridge-Emulation noch nicht aktiviert, aber man kann per Software das Flash oder RAM in den ATARI einblenden lassen und auch darauf schreibend zugreifen. Dieser Modus wird zum Beispiel von der Flasher Software benötigt.

Damit auch eine Cartridge-Emulation aus dem RAM möglich ist, und damit das Flash-ROM besser geschützt wird, muss der Schreibzugriff zuvor aber noch per Software explizit freigeschaltet werden. Beim Einschalten (und nach jedem Druck auf die [Reset]-Taste) wird die Schreiberlaubnis übrigens wieder automatisch zurückgesetzt.

Wenn der „CartEmu“ Schalter geschlossen ist, so wird beim Einschalten des ATARIs die Cartridge-Emulation aktiviert. Dies ist unabhängig davon ob auch der „Flash Write“ Jumper aktiviert ist. Sind beide geschlossen, so ist die Cartridge-Emulation beim Start aktiv und man kann auch schreibend auf das Flash zugreifen. Ist nur der „CartEmu“ Schalter geschlossen, so werden Schreibzugriffe auf das Flash-ROM oder RAM unterbunden.

Auf der Freezer Platine befindet sich eine eigene kleine Schaltung, die beim Einschalten des ATARIs einen Impuls erzeugt. Zusätzlich ist der CPLD mit der Reset-Leitung des ATARIs verbunden. Somit kann die Freezer Logik unterscheiden, ob der ATARI gerade neu eingeschaltet worden ist, oder ob die [Reset]-Taste am ATARI gedrückt wurde.

Wird der ATARI eingeschaltet und der CartEmu Schalter ist aktiviert passiert folgendes: Die Cartridge-Emulation wird aktiviert, der Cartridge Typ wird auf „8k“ gesetzt, als Quelle für die Cartridge-Emulation wird das Flash-ROM gewählt, der Schreibzugriff auf Flash und RAM wird deaktiviert, die Logik wird so konfiguriert dass die Einstellungen bei einem Reset erhalten bleiben, und die Flash Bank Nummer wird auf 62 gesetzt. In dieser Bank befindet sich das Cartridge Auswahl Menü, das daraufhin gestartet wird.

Normalerweise bleiben die Einstellungen, die man im Cartridge-Auswahlmenü getroffen hat, auch bei einem Druck auf Reset erhalten. Setzt man aber „KEEP (settings)“ auf „OFF“, so wird bei einem Reset, genauso wie beim Einschalten des ATARIs, die gesamte Cartridge-Emulation zurückgesetzt und man landet wieder im Auswahl Menü. Somit kann man relativ einfach, ohne Ein- und Ausschalten des ATARIs, mehrere Cartridgekonfigurationen testen.

## 8.2 Software-Konfiguration der Cartridge-Emulation

Ist einer der beiden CarEmu/FlashWrite Schalter/Jumper geschlossen, so kann die Cartridge-Emulation durch Zugriff auf den Adressbereich \$D500-\$D5FF konfiguriert werden. Folgende Adressen werden dafür verwendet:

\$D580	Cartridge-Emulation deaktivieren
\$D581	Cartridge-Emulation aktivieren

\$D582	Bei Druck auf [ Reset ] die Konfiguration zurücksetzen (keep=off)
\$D583	Bei Druck auf [ Reset ] die Konfiguration beibehalten (keep=on)
\$D584	Schreibzugriff auf Flash/RAM verbieten
\$D585	Schreibzugriff auf Flash/RAM erlauben
\$D586	Quelle für Cartridge-Emulation ist das Flash-ROM
\$D587	Quelle für Cartridge-Emulation ist das Freezer-RAM
\$D588	Modus 8k Cartridge (\$A000-\$BFFF)
\$D589	Modus 16k Cartridge (\$8000-\$BFFF)
\$D58A	Modus OSS Bankswitching Cartridge (16k in \$A000-\$BFFF)
\$D58B	Modus SpartaDosX Cartridge (64k von \$A000-\$BFFF)
\$D540-\$D57F	Banknummer 0-63 (\$D540=Bank 0, \$D541=Bank 1, ...)

Bei der Emulation von 16k Cartridges und von OSS Cartridges können nur gerade Banknummern verwendet werden! Genauer gesagt wird das unterste Bit der Banknummer ignoriert, bei 16k Cartridges wird stattdessen die Adressleitung A13 verwendet, bei OSS Cartridges werden A12 und A13 durch eine eigene Bankswitchinglogik gesteuert:

Im Bereich \$B000-\$BFFF ist bei OSS Cartridges immer fix der erste 4k Block (Nummer 0) eingeblendet. Der Bereich \$A000-\$AFFF wird durch folgende Adressen gesteuert:

\$D500	4k Block Nummer 1 einblenden
\$D509	4k Block Nummer 2 einblenden
\$D501	4k Block Nummer 3 einblenden

Durch einen Zugriff auf \$D508 können OSS Cartridges (vorübergehend) komplett abgeschaltet werden.

Bei der Emulation von SpartaDosX Cartridges werden die untersten 3 Bits der Banknummer ignoriert, stattdessen werden folgende Adressen zur Bankumschaltung verwendet:

\$D5E0	8k Bank Nummer 7 einblenden
\$D5E1	8k Bank Nummer 6 einblenden
\$D5E2	8k Bank Nummer 5 einblenden
\$D5E3	8k Bank Nummer 4 einblenden
\$D5E4	8k Bank Nummer 3 einblenden
\$D5E5	8k Bank Nummer 2 einblenden
\$D5E6	8k Bank Nummer 1 einblenden
\$D5E7	8k Bank Nummer 0 einblenden
\$D5E8-\$D5EF	Cartridge (vorübergehend) abschalten

Um die Cartridge-Emulation zu konfigurieren sollten die einzelnen Einstellungen in folgender Reihenfolge vorgenommen werden:

1. Keep Settings auf on/off (\$D582/\$D583)
2. Cartridge-Emulation aktiv/deaktiv

Wenn Cartridge-Emulation aktiviert, dann:

3. Quelle für Cartridge-Emulation (Flash/RAM)
4. Cartridgetyp (8k/16k/OSS/SDX)
5. Banknummer setzen

6a. Wenn OSS Cartridge, dann OSS Bank Nummer 1 einblenden

6b. Wenn SDX Cartridge, dann SDX Bank Nummer 0 einblenden

Möchte man das Flash bzw. RAM beschreiben, so kann man nun per Zugriff auf \$D585 den Schreibzugriff erlauben. Wenn man mit dem Beschreiben fertig ist, sollte man sofort wieder den Schreibzugriff per \$D584 deaktivieren, damit nicht versehentlich Daten überschrieben werden können!

### **8.3 Der Oldrunner Modus**

Ist der Oldrunner-Modus aktiv, so passiert folgendes: Bei einem Lese-Zugriff auf \$E000-\$FFFF wird die 8k Bank Nummer 63 (also die letzte Bank) des Flash-ROMs in den ATARI eingeblendet.

Bei einem Zugriff auf \$C000-\$CFFF wird das interne ATARI RAM/ROM ausgeblendet, aber kein RAM oder Flash-ROM vom Freezer eingeblendet. Genau wie beim Original ATARI 400/800 wird also in diesem Bereich eine „Lücke“ simuliert.

### **8.4 Der Freezer-Modus**

So lange man den Freezerknopf nicht drückt, ist der Freezer vom ATARI aus absolut unsichtbar. Das heisst, keine Software kann feststellen ob ein Freezer am ATARI angesteckt ist!

Intern kennt die Freezerlogik 4 verschiedene Zustände:

- Freezer deaktiviert
- Freezer halb aktiviert
- Freezer aktiviert
- Freezer temporär deaktiviert

Nach dem Einschalten befindet sich der Freezer im Zustand „deaktiviert“.

Wie schon erwähnt, ist im Freezer-Modus das RAM in 4k Bänke (anstatt der sonst üblichen 8k Bänke) unterteilt. Insgesamt stehen hier also also 32 4k Bänke zur Verfügung.

Um die nicht auslesbaren Hardware-Register im Bereich \$D000-\$D7FF doch auslesen zu können, verwendet der Freezer folgenden „Trick“: Schreibt man Daten in den Bereich \$D000-\$D7FF, so werden diese auch in die 4k Bank Nummer 31 (also die letzte Bank) des RAM-Chips am Freezer geschrieben, wenn sich der Freezer im Zustand "deaktiviert" befindet. Der ATARI bemerkt davon aber wiederum nichts.

Bei Schreibzugriffen auf \$D2xx, \$D3xx und \$D4xx werden die Adressleitungen A4-A7 zum RAM deaktiviert. Das ist notwendig, da im ATARI die Adressleitungen der Peripheriechips nicht vollständig ausdekodiert sind. Schreibt man z. B. zuerst \$22 in \$D400 und dann \$00 in \$D410, so landen beide Werte im DMACTL Register des ANTICS, gültig ist aber der letzte geschriebene Wert. Durch das Ausmaskieren der Adressleitungen landen beide Schreibzugriffe an der Adresse \$0400 in der 4k RAM Bank Nummer 31, der zweite Schreibzugriff überschreibt also den ersten.

Bei Schreibzugriffen auf \$D0xx werden übrigens nur A5-A7 ausmaskiert, da der GTIA insgesamt 32 Bytes (\$D000-\$D01F) belegt.

## **8.5 Aktivierung des Freezers**

Drückt man nun den Freezerknopf, so passiert erstmal noch nichts. Erst, wenn der erste Zugriff auf den Bereich \$FFF8-\$FFFF (hier liegen unter anderem die IRQ und NMI Vektoren) erfolgt und der Freezerknopf gedrückt ist, geht der Freezer in den Zustand „halb aktiviert“ über.

Im nächsten Taktzyklus gibt es nun folgende Möglichkeiten:

Wenn der Freezerknopf nicht mehr gedrückt ist, oder wenn ein Zugriff auf einen anderen Speicherbereich als \$FFF8-\$FFFF erfolgt, geht der Freezer wieder in den Zustand „deaktiviert“ zurück.

Nur wenn der Freezerknopf immer noch gedrückt ist, und wieder ein Zugriff auf \$FFF8-\$FFFF erfolgt, aktiviert sich der Freezer:

Das RAM/ROM im ATARI (bzw das Oldrunner OS im Freezer) wird ausgeblendet und stattdessen wird die Flash-Bank Nummer 60 eingeblendet. An den entsprechenden Stellen (die letzten 8 Bytes von Bank 61) stehen nun Werte, die die Interrupt-Vektoren auf die Freezer-Software „umbiegen“.

Nachdem der Freezer der CPU seine eigenen Interrupt-Vektoren „reingeschmuggelt“ hat, geht der Freezer in den Zustand „aktiviert“ über.

Im Zustand „aktiviert“ passiert nun folgendes:

Von \$0000-\$1FFF wird das RAM des Freezers eingeblendet (dazu gleich mehr).

Von \$2000-\$3FFF wird das ROM des Freezers eingeblendet. Ganz zu Beginn des ROMs liegt die „umgebogene“ Interrupt-Routine, danach folgt die eigentliche Freezer-Software.

Möchte man von der Freezer-Software aus auf den Bereich \$0000-\$3FFF zugreifen, so muss zuerst das RAM und ROM des Freezers vorübergehend ausgeblendet werden:

Ein Schreibzugriff auf die Adresse \$D700 schaltet den Freezer zwischen den Zuständen „aktiviert“ und „temporär deaktiviert“ hin und her.

Beim Beenden der Freezer-Software soll der Freezer natürlich wieder in den Zustand „deaktiviert“ versetzt werden. Dies geschieht mit einem Lesezugriff auf die Adresse \$D700.

Um die 128k RAM kompatibel mit der bestehenden Freezer-Software nutzen zu können und andererseits Erweiterungen der Software möglichst einfach zu halten, wurde das RAM folgendermassen unterteilt:

Im Bereich \$0000-\$0FFF ist immer der letzte 4k Bank des RAMs (also Bank 31) eingeblendet.

Im Bereich \$1000-\$1FFF kann eine beliebige andere 4k Bank des RAMs eingeblendet werden. Sinnvollerweise sollte man hier nur die Bänke 0-30 verwenden, da ja Bank 31 bereits von \$0000-\$0FFF verwendet wird.

Die original Freezer-Software verwendet nur den Bereich von \$0000-\$07FF des RAMs – der original Freezer hatte ja nur 2k RAM insgesamt – das heisst also, für eigene Erweiterungen steht

der Bereich von \$0800-\$0FFF (immer verfügbar) sowie 31 Bänke im Bereich \$1000-\$1FFF zur Verfügung.

Zwischen den RAM-Bänken kann durch Zugriff auf die Adressen \$D780-\$D79F umgeschaltet werden. \$D780 ist Bank 0, \$D781 ist Bank 1, ...

Das Flash-ROM ist wie in der Cartridge-Emulation auch in 8k Bänke unterteilt. Damit die Cartridge-Emulation und der Freezer-Modus unabhängig voneinander funktionieren, gibt es im Freezer-Modus aber eigene Bankselect Adressen, \$D740-\$D77F. \$D740 aktiviert die 8k Bank Nummer 0, \$D741 Bank Nummer 1 und so weiter.

In der aktuellen Version ist Bank Nummer 63 vom Oldrunner OS belegt, Bank Nummer 62 von der Cartridge Auswahl, und die Bänke 59 bis 61 werden von der Freezer-Software belegt. Bank Nummer 60 ist hierbei die „Hauptbank“ der Freezer-Software. Da das original Freezer-ROM absolut voll war (und somit kein Platz mehr für Erweiterungen war), verwendet die aktuelle Software zusätzlich auch noch die Bank Nummer 61.

Zukünftige Erweiterungen an der Freezer-Software sollten, falls notwendig, zuerst immer die oberen Bänke 58, 57, ... belegen. So belegt die Freezer-Software immer die „oberen“ Banknummern, während die „unteren“ Banknummern für die Cartridge-Emulation zur Verfügung stehen.

Wenn man innerhalb der Freezer-Software von einer Bank in eine andere wechseln will, muss man beachten, dass immer die ganze 8k Bank umgeschaltet wird. Das heisst, direkt nach dem Zugriff auf \$D740-\$D77F ist man schon in der neuen Bank und der nächste Befehl wird schon aus der neuen ROM Bank gelesen!

Mit folgendem einfachem Trick ist aber auch das Wechseln zwischen verschiedenen ROM-Bänken recht einfach:

Angenommen, die Freezer Software ist auf zwei Bänke verteilt und man möchte von Bank 60 aus die Routine "FOO" in Bank 61 aufrufen. Links ist der Assemblercode für Bank 60, rechts der für Bank 61.

Zuerst mal der Code der eigentlichen Routinen FOO und BAR:

Bank 60				Bank 61			
\$2A10	A9 10	BAR	LDA #\$10	...			
\$2A12	60		RTS	...			
\$2E82		...	A9 20	FOO	LDA #\$20		
\$2E84		...	60		RTS		

Die Routine „FOO“ in Bank 61 liegt also bei Adresse \$2E82, weiter gibt es ab Adresse \$2A10 in Bank 60 eine Routine „BAR“. Von Bank Nummer 61 kann die Routine FOO ganz normal per „JSR FOO“ aufgerufen werden, das wird dann zu „20 82 2E“ assembliert.

Um in eine andere Bank zu wechseln, kann man folgendes „Sprungbrett“ für einen JSR Aufruf verwenden:

Bank 60				Bank 61			
\$3FF0	8D 7D D7	SFOO STA \$D77D		8D 7C D7	SBAR STA \$D77C		
\$3FF3	20 10 2A	JSR BAR		20 82 2E	JSR FOO		
\$3FF6	8D 7D D7	STA \$D77D		8D 7C D7	STA \$D77C		
\$3FF9	60	RTS		60	RTS		

Ein JMP-Aufruf kann so aussehen:

\$3FFA	8D 7D D7	JFOO STA \$D77D		8D 7C D7	JBAR STA \$D77C		
\$3FFD	4C 10 2A	JMP BAR		4C 82 2E	JMP FOO		

Hier die Erklärung, was genau bei einem Aufruf von „JSR SFOO“ aus Bank Nummer 60 heraus passiert:

Zuerst wird aus Bank Nummer 60 ab Adresse \$3FF0 der Befehl „STA \$D77D“ gelesen und danach ausgeführt. Nach der Ausführung des Befehls ist Bank 61 aktiv.

Nun wird aus Bank 61 ab Adresse \$3FF3 der Befehl „JSR FOO“ gelesen und ausgeführt. Nach dem Ende des Unterprogrammes kehrt die CPU nach Adresse \$3FF6 (immer noch in Bank 61) zurück.

Danach wird aus Bank 61 ab Adresse \$3FF6 der Befehl „STA \$D77C“ gelesen, ausgeführt, und damit wieder auf Bank 60 umgeschaltet.

In Bank 60 steht an der Adresse \$3FF9 nun der RTS Befehl und der Unterprogramm-Aufruf (JSR SFOO) wird beendet.

Insgesamt werden für diesen Aufruf die Adressen \$3FF0–3FF2 sowie \$3FF9 aus Bank 60 und \$3FF3–\$3FF8 aus Bank 61 verwendet.

Die „ungenutzten“ Adressen kann man nun wie in dem Beispiel gezeigt dazu verwenden, ein „Sprungbrett“ zum Aufruf der Routine BAR in Bank 60 von der Bank 61 aus unterzubringen. Der Code ist mehr oder weniger derselbe wie für den Aufruf der Routine FOO in Bank 61, nur sind eben die Banknummern vertauscht.

Dieses System ist zwar recht simpel, aber man muss einige wichtige Details beachten:

- Im Gegensatz zu einem „normalen“ JSR, das sich ja die (gesamte) Rücksprungadresse merkt, muss man hier selber dafür sorgen, dass man wieder in die richtige Bank zurückspringt.
- Will man eine Unteroutine vorzeitig beenden (zum Beispiel im Fehlerfall), so muss man beachten, dass beim JSR-Aufruf entweder 2 Return-Adress-Bytes (wenn die Routine aus der selben Bank aufgerufen wurde) oder 4 Bytes (wenn die Routine aus einer anderen Bank aufgerufen wurde) am Stack liegen können. In der original Freezer Software bereitet dies keine Probleme, da in der Fehlerbehandlung der Stackpointer komplett zurückgesetzt wird.

## 9 Speicheraufteilung im Freezer

Bank	64K	Block	Bereich	Inhalt
63	E000-FFFF	8	512	Freezer / Oldrunner OS
62	C000-DFFF		504	Freezer / Cartridge Menu
61	A000-BFFF		496	Freezer
60	8000-9FFF		488	Freezer
59	6000-7FFF		480	Freezer
58	4000-5FFF		472	
57	2000-3FFF		464	
56	0000-1FFF		456	
55	E000-FFFF	7	448	
54	C000-DFFF		440	
53	A000-BFFF		432	
52	8000-9FFF		424	
51	6000-7FFF		416	
50	4000-5FFF		408	
49	2000-3FFF		400	
48	0000-1FFF		392	
47	E000-FFFF	6	384	
46	C000-DFFF		376	
45	A000-BFFF		368	
44	8000-9FFF		360	
43	6000-7FFF		352	
42	4000-5FFF		344	
41	2000-3FFF		336	
40	0000-1FFF		328	
39	E000-FFFF	5	320	
38	C000-DFFF		312	
37	A000-BFFF		304	
36	8000-9FFF		296	
35	6000-7FFF		288	
34	4000-5FFF		280	
33	2000-3FFF		272	
32	0000-1FFF		264	

Bank	64K	Block	Bereich	Inhalt
31	E000-FFFF	4	256	
30	C000-DFFF		248	
29	A000-BFFF		240	
28	8000-9FFF		232	
27	6000-7FFF		224	
26	4000-5FFF		216	
25	2000-3FFF		208	
24	0000-1FFF		200	
23	E000-FFFF	3	192	
22	C000-DFFF		184	
21	A000-BFFF		176	
20	8000-9FFF		168	
19	6000-7FFF		160	
18	4000-5FFF		152	
17	2000-3FFF		144	
16	0000-1FFF		136	
15	E000-FFFF	2	128	
14	C000-DFFF		120	
13	A000-BFFF		112	
12	8000-9FFF		104	
11	6000-7FFF		96	
10	4000-5FFF		88	
9	2000-3FFF		80	
8	0000-1FFF		72	
7	E000-FFFF	1	64	
6	C000-DFFF		56	
5	A000-BFFF		48	
4	8000-9FFF		40	
3	6000-7FFF		32	
2	4000-5FFF		24	
1	2000-3FFF		16	
0	0000-1FFF		8	

## 10 Weiterführende Informationen und Links

Website zum TURBO-FREEZER mit aktuellen Infos, Updates etc:  
<http://turbofreezer.horus.com/>

Website des ABBUC, Bezugsquelle für den TURBO-FREEZER:  
<http://www.abbuc.de/>

Lattice Semiconductors: Datenblätter zum iM4A5, Download der Entwicklungsumgebung  
ispLever Starter:  
<http://www.latticesemi.com/>

AMD: Datenblätter zum Flash ROM 29F040:  
<http://www.amd.com/>

## 11 Garantie- und Verwendungsbestimmungen

Auf Material und Zusammenbau werden 6 Monate Garantie gewährt. Sollte der Freezer defekt sein, wird er gegen einen funktionstüchtigen Freezer umgetauscht. Sollte der Freezer wegen Hardwareinkompatibilität nicht funktionieren, wird nach freigemachter Rücksendung der Kaufpreis zurückerstattet.

Diese Garantie erstreckt sich nur auf die Hardware. Das Risiko für Fehlerlosigkeit und Funktionstüchtigkeit der eingebauten Software liegt allein beim Käufer oder Anwender. Ansprüche gleich welcher Art bezüglich der Software, wie Austausch, Verbesserung, Minderung, Wandelung, Aufdatierung usw. bestehen nicht.

Nicht unter die Garantie fallen Schäden durch Transport, unsachgemäßen Einbau oder unsachgemäße Handhabung, statische Aufladung, Verschmutzung, versuchter Erweiterung und Eingriffe gleich welcher Art in die Elektronik außerhalb unseres Labors. Keine Garantie besteht für Teile, die nicht von uns eingebaut wurden.

Der TURBO-FREEZER XL/XE darf ausschließlich im Rahmen der gesetzlichen Bestimmungen, insbesondere den Urheberrechts angewandt werden. Eine Haftung oder Verantwortung für Schäden gleich welcher Art, die dem Käufer, Anwender oder Dritten, lebender oder toter Materie durch Verwendung oder Missbrauch des TURBO-FREEZER XL/XE entstehen könnten oder entstanden sind, wird von uns strikt abgelehnt. Es ist einzig und allein Sache des Käufers, Eigentümers oder Anwenders, missbräuchliche Verwendung oder sonstige Schäden auszuschließen. Eltern haften für ihre Kinder.

Es ist Sache des Anwenders, sich von den gesetzlichen Vorschriften Kenntnis zu verschaffen und im Zweifelsfall den Ratschlag eines Fachmanns einzuholen. Er kann sich nicht auf Unkenntnis berufen.

Diese Anleitung ist urheberrechtlich geschützt. Alle Rechte, insbesondere der Übersetzung in fremde Sprachen, vorbehalten. Kein Teil der Anleitung darf ohne schriftliche Genehmigung des Autors in irgendeiner Form – durch Photokopie, Mikrofilm oder andere Verfahren – reproduziert oder in eine von Maschinen, insbesondere von Datenverarbeitungsanlagen, verwendete Sprache übertragen werden.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in dieser Anleitung berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass diese Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürfen. Alle Angaben erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Jede Haftung von Seiten des Herstellers ist ausgeschlossen. Verwendung auf eigene Gefahr.



