

# **TURBO FREEZER XL/XE 2011**

Anleitung vom 17. 3. 2013  
für Freezer-Softwareversion 3.10

Copyright ©1987-2013 Peter Dell, Florian Dingler, Bernhard Engl, Matthias Reichl

Diese Anleitung ist urheberrechtlich geschützt und steht unter der Creative Commons Lizenz BY-NC-SA 3.0.

<http://creativecommons.org/licenses/by-nc-sa/3.0/de>

Dies bedeutet, Sie müssen den Namen der Autoren / Rechteinhaber in der von ihnen festgelegten Weise nennen. Dieses Werk bzw. dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden. Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.

# Inhaltsverzeichnis

<b>1</b>	<b>Installation</b>	<b>11</b>
1.1	Überprüfen der Stromversorgung . . . . .	12
1.2	Zusammenbau des Freezers . . . . .	12
1.2.1	Schutz vor elektrostatischer Entladung sicherstellen . . . . .	12
1.2.2	Freezer-Elektronik mit der Adapterplatine verbinden . . . . .	12
1.2.3	Freezer-Elektronik von der Adapterplatine trennen . . . . .	13
1.3	Anstecken des Freezers . . . . .	14
1.3.1	Atari 600 XL (mind. 64k RAM) und Atari 800 XL . . . . .	14
1.3.2	Atari 800 XE, 65 XE und 130 XE . . . . .	14
1.4	Inbetriebnahme des Freezers . . . . .	14
1.4.1	Grundkonfiguration setzen . . . . .	14
1.4.2	Freezer starten . . . . .	14
1.4.3	Systemdiskette erstellen . . . . .	15
1.4.4	Stereo-POKEY-Modus aktivieren (optional) . . . . .	16
1.4.5	Integrierte 512k RAM-Erweiterung aktivieren (optional) . . . . .	16
1.4.6	Steckmodule am Atari XE verwenden . . . . .	17
1.5	Erweiterte Konfigurationen für Lötexperten . . . . .	17
1.5.1	Interne Stromversorgung bereitstellen . . . . .	17
1.5.2	SYSTEM RESET-Taste einbauen . . . . .	18
1.5.3	Kompatibilität zu 1MB RAM-Erweiterungen herstellen . . . . .	18
<b>2</b>	<b>Einsatz</b>	<b>19</b>
2.1	Freezer-Hauptmenü aufrufen . . . . .	20
2.2	Programm einfrieren und auftauen . . . . .	20
2.3	Eingefrorenes Programm abspeichern . . . . .	21
2.3.1	Ladeprogramm für Bootdisketten anpassen . . . . .	22
2.3.2	Nachladendes Programm einfrieren . . . . .	22
2.4	Zwischen eingefrorenem und gespeichertem Programm wechseln . . . . .	23
2.5	RAM unter dem OS-ROM löschen . . . . .	23
2.6	Debugger und DOS starten . . . . .	23
2.7	Menü der Cartridge-Emulation aufrufen . . . . .	24
<b>3</b>	<b>Debugger und DOS</b>	<b>25</b>
3.1	Disk Operating System (DOS) . . . . .	25
3.1.1	Ausführen von Kommandos . . . . .	26
3.1.2	Kommandoübersicht . . . . .	26
3.1.3	Besonderheiten mancher Kommandos . . . . .	27

3.1.4	Fehlermeldungen . . . . .	27
3.2	Debugger . . . . .	28
3.2.1	Zugriff auf Speicher und Hardwareregister . . . . .	28
3.2.2	Kommandoübersicht . . . . .	29
3.2.3	Werte eingeben (hexadezimal, dezimal, interner Code) . . . . .	31
3.2.4	Vektoren eingeben . . . . .	31
3.2.5	Auf erweiterten Speicher, RAM, ROM zugreifen . . . . .	32
3.2.6	Display List anzeigen . . . . .	32
3.2.7	Speicher durchsuchen . . . . .	33
3.2.8	SIO-Kommandos ausführen . . . . .	33
3.2.9	Interrupt und VCOUNT anzeigen . . . . .	34
3.2.10	Direct-I/O-Modus aktivieren . . . . .	35
3.2.11	Versionsnummer anzeigen . . . . .	35
3.2.12	OS-Vektoren anzeigen . . . . .	35
3.2.13	Handler-Adressen-Tabelle anzeigen . . . . .	35
3.2.14	Speicherbelegungsübersicht anzeigen . . . . .	36
3.2.15	Druckerausgabe aktivieren . . . . .	36
3.2.16	AtariSIO fernsteuern . . . . .	36
<b>4</b>	<b>512k RAM-Erweiterung</b>	<b>37</b>
<b>5</b>	<b>Oldrunner – Atari OS Rev. B</b>	<b>39</b>
<b>6</b>	<b>Cartridge-Emulation</b>	<b>41</b>
6.1	Grundlagen . . . . .	41
6.1.1	Banknummern . . . . .	41
6.1.2	Modultypen . . . . .	42
6.2	Aktivieren der Cartridge-Emulation . . . . .	42
6.3	Menü der Cartridge-Emulation . . . . .	43
6.4	Kommandoübersicht . . . . .	44
6.5	Weiterführende Informationen . . . . .	45
6.5.1	Verwendung „richtiger“ Module . . . . .	45
6.5.2	TRIG3 – \$D013 . . . . .	45
<b>7</b>	<b>Programmieren des Flash-ROMs</b>	<b>47</b>
7.1	Grundlagen . . . . .	47
7.2	Bänke und Blöcke im Flash-ROM . . . . .	47
7.3	Flash-Programm starten . . . . .	48
7.4	Flash-Programm verwenden . . . . .	48
7.5	Flash-ROM und Freezer-RAM programmieren . . . . .	49
7.6	Freezer-Software aktualisieren . . . . .	50
<b>8</b>	<b>Für Experten: Technische Details</b>	<b>51</b>
8.1	Hardware . . . . .	51

8.2	Speicheraufteilung . . . . .	51
8.2.1	Freezer-RAM . . . . .	51
8.2.2	Flash-ROM . . . . .	51
8.3	Freezer . . . . .	52
8.3.1	Shadowing der Hardwareregister . . . . .	52
8.3.2	Einblenden des Freezer-Speichers . . . . .	53
8.3.3	Aktivierung des Freezers . . . . .	53
8.3.4	Konfigurationsregister . . . . .	54
8.3.5	Freezer-Software und OS-ROMs . . . . .	55
8.4	512k RAM-Erweiterung . . . . .	56
8.5	Oldrunner . . . . .	57
8.6	Cartridge-Emulation . . . . .	57
8.6.1	Konfigurationsschalter . . . . .	57
8.6.2	Konfigurationsregister . . . . .	57
<b>9</b>	<b>Weiterführende Informationen und Downloads</b>	<b>63</b>



# Vorwort

## Vorwort des Neukonstrukteurs (2012)

Der TURBO FREEZER 2005 war ein großer Erfolg. Er kam in der Atari Community sehr gut an und wir bekamen viel positives Feedback und Ideen für neue Funktionen. Dabei erwies sich der Einsatz von Flash-ROM anstelle eines EPROMS als sehr hilfreich. Es ermöglichte den Benutzern sehr einfach die Software vom Atari aus zu aktualisieren und so in den Genuss der neuen Funktionen zu kommen.

Insbesondere der Debugger wurde seit der Erstauflage 2005 erheblich erweitert. Es kam eine Online-Hilfe dazu, Kommandos um Speicherblöcke zu verschieben oder zu vergleichen, Übersichten für die Speicherbelegung und die OS-Vektoren, die Möglichkeit Debugging-Sitzungen am Drucker mit zu protokollieren und noch vieles mehr. An dieser Stelle ein großes Dankeschön vor allem an Erhard Pütz und Carsten Strotmann für die vielen guten Ideen!

Nachdem die Erstauflage ausverkauft war erreichten uns mehr und mehr Anfragen doch eine weitere Serie aufzulegen. 2010 wurde es dann konkreter. Zusammen mit Wolfram Fischer, der auch schon an der TURBO FREEZER 2005 Version mitgearbeitet hatte, begann ich mit der Planung einer Neuauflage. Mittlerweile war der Lattice M4A5 CPLD nur mehr relativ schwierig zu bekommen und er war auch deutlich teurer als vergleichbare Chips. Auch hätten wir gerne einen etwas „größeren“ Chip verwendet um einige neue Ideen und Funktionen umsetzen zu können. So entschieden wir uns einen Xilinx XC95144XL als Basis für den neuen Freezer zu verwenden. Bei etwa gleichem Preis bot er im Vergleich zum Lattice CPLD mehr als doppelt so viel Leistung (I/O Pins und Platz für Logik). So hatten wir die Möglichkeit lang ersehnte Features – z. B. eine RAM-Erweiterung, wie sie auch schon der allererste Freezer von 1987 hatte – umzusetzen.

Wolfram baute Ende 2010 einen ersten Prototypen der neuen Hardware, dadurch hatte ich die Basis um Anfang 2011 mit der Entwicklung der neuen Logik und Software zu beginnen. Dabei wurde mir schnell klar, dass es am besten ist die Logik von Grund auf neu zu entwickeln und die einzelnen Funktionsblöcke (Freezer, Cartridge-Emulation etc.) voneinander zu trennen. Dadurch war es recht einfach möglich Erweiterungen in den einzelnen Blöcken vorzunehmen.

Die Logik der Cartridge-Emulation wurde komplett umgekrempelt, die Freezer Logik hielt ich aber bewusst kompatibel zur alten TURBO FREEZER 2005 Logik: Ich wollte, dass die Freezer-Software mit nur minimalen Änderungen sowohl am „alten“ als auch am neuen Freezer läuft und damit auch Besitzer des alten Freezers in den Genuss weiterer Updates kommen können.

Im Sommer 2011 war es dann soweit, nach einigen neuen Hardware Revisionen und etlichen Änderungen an Software und Logik war der neue Freezer fertig! Leider verzögerte

---

sich der eigentlich für Ende 2011 geplante Beginn der Serienproduktion um fast ein ganzes Jahr. Wolfram musste in eine neue Wohnung übersiedeln und ich hatte viel in der Arbeit zu tun – damit hatten wir beide so gut wie keine Zeit für unser Hobby. Dank der Unterstützung durch den ABBUC konnte die Produktion aber schließlich im Herbst 2012 endlich beginnen! Vielen Dank an Wolfgang Burger für die Unterstützung und an alle Vorbesteller die geduldig auf den neuen Freezer gewartet haben.

Salzburg, im November 2012

Matthias Reichl

## **Vorwort des Neukonstrukteurs (2005)**

Der Startschuss für die Entwicklung des neuen TURBO FREEZER 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 im Team waren noch Bernhard Pahl, Florian Dingler, 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 Tipps und die detaillierten Infos zum original TURBO FREEZER, sowie bei allen anderen für die Anregungen und fürs 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 ist.

Einige der Erweiterungen am TURBO FREEZER sind übrigens recht zufällig entstanden, z. B. 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 Freezer-RAM abzulegen und das Flash-ROM wurde durch einen 512k Typ ersetzt. Dies erforderte aber wiederum einen leistungsfähigeren Logik-IC.

Durch den neuen Logik-IC war es dann wiederum möglich, auch 16k OSS-Module und das SpartaDOS X-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 E-Mail an [hias@horus.com](mailto:hias@horus.com).

Die Funktionsvielfalt des TURBO FREEZERS 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-Emulation, 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 256k 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 wird einfach am parallelen Bus angesteckt. Dieser unterscheidet sich bei Atari XL und XE Computern. Bei Atari XL Computern besteht der parallele Bus aus dem Parallel Bus Interface (PBI), welches mit „Parallel Bus“ beschriftet ist. Bei Atari XE Computern besteht der parallele Bus aus dem Modulschacht und dem Enhanced Cartridge Interface (ECI), die mit „Cartridge“ bzw. „Expansion“ beschriftet sind. Die Stromversorgung erfolgt durch den parallelen Bus. Beim Atari 600 XL und bei älteren Atari 800 XL sowie Atari XE Computern liegt die notwendige Stromversorgung direkt am parallelen Bus an. Bei neueren Atari 800 XL ist sie dort nicht mehr vorhanden und muss über das am Freezer angelötete Kabel über den Joystickport zugeführt werden. 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.

Der TURBO FREEZER besteht aus zwei Teilen, der Freezer-Elektronik im schwarzen Gehäuse und einer Adapterplatine, welche die Verbindung zum PBI bzw. ECI herstellt. Die Adapterplatinen sind auch separat erhältlich, dadurch lässt sich ein TURBO FREEZER einfach und kostengünstig sowohl an Atari XL als auch an Atari XE Computern verwenden.

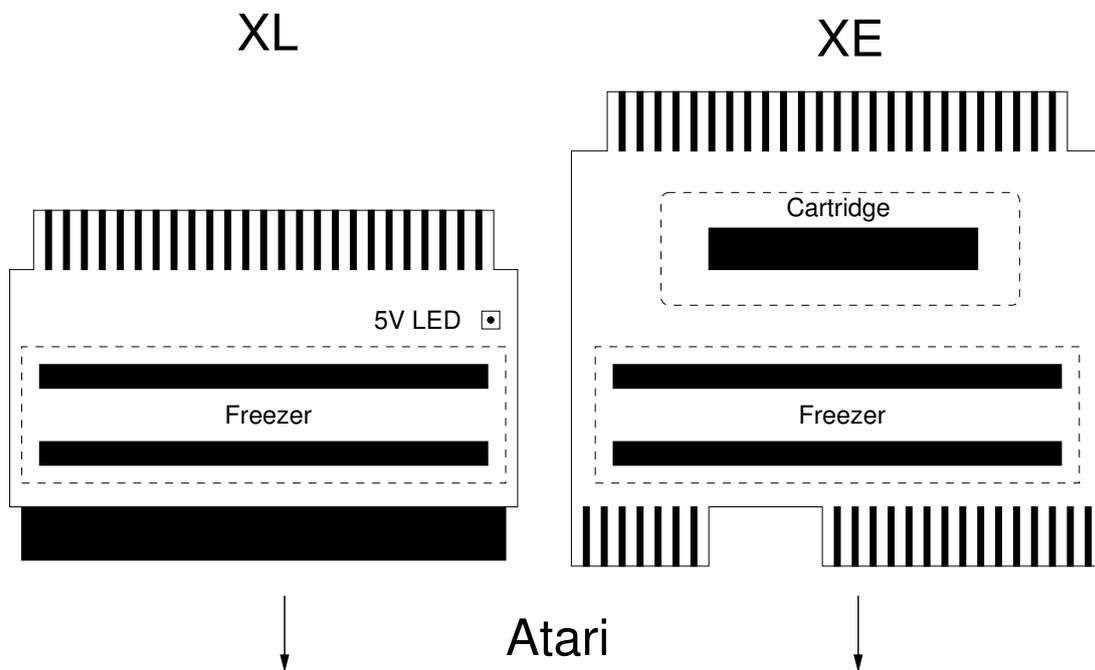


Abbildung 1.1: Atari XL und XE Adapterplatinen

Es gibt einige wenige Modelle der XE-Serie (vornehmlich 65 XE), die keinen ECI-Anschluss besitzen. Diese Rechnermodelle sind daran erkennbar, dass sie hinten nur einen Steckplatz für ein Modul „Cartridge“, aber der zweite Anschluss „Expansion“ fehlt. Bei diesen XE Modellen kann der Freezer leider nicht verwendet werden.

**Achtung:** Um Beschädigungen beim Transport zu vermeiden wurde die Freezer-Elektronik vor dem Versand von der Adapterplatine getrennt. Um Beschädigungen beim Ein- bzw. Ausstecken der Adapterplatine in den Atari oder beim Ein- bzw. Ausstecken der Freezer-Platine in die Adapterplatine zu vermeiden, sollte man immer den Atari ausschalten.

### 1.1 Überprüfen der Stromversorgung

Vor dem Zusammenbau sollten Besitzer eines Atari XL Computers überprüfen, ob am PBI die 5V Stromversorgung vorhanden ist. Besitzer eines Atari XE Computers können diesen Schritt überspringen.

Zuerst den Atari ausschalten. Dann die leere die Atari XL Adapterplatine ohne das Freezer-Elektronik Modul am PBI anstecken. Anschließend den Atari einschalten. Leuchtet die 5V LED auf der Adapterplatine, so liegen 5V am PBI an und das Stromversorgungskabel zum Joystickport wird nicht benötigt. Man sollte es aber trotzdem nicht abschneiden, sondern ordentlich zusammenrollen denn es kann ja später einmal notwendig sein, den Freezer auch an einem Atari ohne Stromversorgung am PBI einzusetzen. Danach den Atari ausschalten und die Adapterplatine wieder vom PBI abziehen.

### 1.2 Zusammenbau des Freezers

#### 1.2.1 Schutz vor elektrostatischer Entladung sicherstellen

Vor dem Zusammenbau des TURBO FREEZERS sollte man sich **unbedingt erden** um statische Aufladungen zu entladen, die z.B. durch den Teppich, die Schuhe, die Kleidung usw. entstehen. Diese Ladungen können ohne weiteres einige 1000 Volt betragen und könnten den XILINX-Chip und weitere ICs des Freezers zerstören. **Auch die Stiftleisten des Freezers sollte man niemals berühren, ohne geerdet zu sein.** Dies gilt im Allgemeinen für alles andere an Elektronik, wobei der Atari selber nicht ganz so empfindlich ist, aber dennoch Schaden nehmen kann.

Am besten berührt man unmittelbar vor dem Zusammenbau die Heizung oder die Schutzleiter-Lasche der Steckdose. Gut geeignet, und im professionellen Bereich sehr verbreitet, sind auch ESD-Armbänder (electro-static discharge, ESD) zum dauerhaften Erden.

#### 1.2.2 Freezer-Elektronik mit der Adapterplatine verbinden

An der Unterseite des Moduls mit der Freezer-Elektronik befinden sich zwei Stiftleisten mit je 2x25 Kontakten. Auf der Adapterplatine befinden sich zwei Buchsenleisten mit je 2x25

Kontakten. Das Freezer-Modul mit der Freezer-Elektronik muss so auf die Adapterplatine gesetzt werden, dass die Vorderseite mit den Schaltern in Richtung des PBI- bzw. ECI-Anschlusses des Ataris zeigt. **Auf keinen Fall darf man das Modul umgekehrt einsetzen, ansonsten kann der Freezer zerstört werden.**

Als erstes muss das Modul mit der Freezer-Elektronik mit den Stiften so über den Buchsen platziert werden, dass alle Stifte über den jeweiligen Buchsen zu liegen kommen. **Es dürfen nirgends Stifte überstehen.** Am besten nimmt man die Adapterplatine in eine und das Modul mit der Elektronik in die andere Hand. Dann das Modul vorsichtig aufsetzen aber noch nicht hineindrücken und von allen Seiten überprüfen ob die Stifte richtig anliegen. Danach das Modul mit einem festen Druck mit der Adapterplatine verbinden. Das geht am einfachsten wenn man dazu den Freezer auf den Kopf dreht, mit den Fingern beider Hände das Modul von links bzw. rechts hält und dann mit beiden Daumen von unten auf die Adapterplatine drückt.

Nach dem Zusammenstecken sollte man noch überprüfen, ob alle Stifte komplett in den Buchsen stecken. Stehen auf einer Seite Stifte über, muss das Module mit der Freezer-Elektronik wieder abgesteckt und erneut, dieses Mal mit korrekter Ausrichtung, auf die Platine gesteckt werden. Sind die Stifte noch nicht komplett in die Buchsen eingedrungen, sollte man noch mal etwas kräftiger nachdrücken bis man von der Seite nichts mehr von den Stiften sehen kann.

### 1.2.3 Freezer-Elektronik von der Adapterplatine trennen

Als erstes muss der Atari ausgeschaltet und der Freezer vom PBI bzw. ECI-Anschluss abgesteckt werden. Durch die beiden 50-poligen Anschlüsse ist die Freezer-Elektronik recht fest mit der Adapterplatine verbunden. **Man darf auf keinen Fall zu große Kraft aufwenden oder ruckartig vorgehen**, ansonsten können die Anschlüsse abbrechen und der Freezer dadurch zerstört werden.

Am besten geht man wie folgt vor: Abwechselnd am linken und rechten Rand des Freezer-Gehäuses, in der Mitte zwischen beiden Anschlussreihen, mit dem Daumen das Freezer-Gehäuse um jeweils 0.5 bis maximal 1 mm vorsichtig anheben. Dabei aufpassen, dass sowohl am linken und am rechten Rand als auch an der vorderen und hinteren Anschlussleiste die Kontakte etwa gleich weit herausgezogen werden.

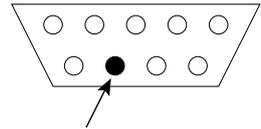
Nach ca. 2 mm lässt die Kraft der Buchsenleisten deutlich nach. Nun muss man besonders vorsichtig vorgehen, da sonst leicht die Anschlüsse verbogen werden können. Nach rund 5 mm ist es geschafft, die Freezer-Elektronik ist von der Adapterplatine getrennt.

## 1.3 Anstecken des Freezers

### 1.3.1 Atari 600 XL (mind. 64k RAM) und Atari 800 XL

Einfach die eingeschnappte PBI Abdeckung hinten am Atari entfernen, die Blechzungen mit den Fingern vorsichtig ein wenig aufbiegen und den Freezer ohne Gewaltanwendung einstecken.

Falls der PBI über eine 5V Stromversorgung verfügt (siehe Abschnitt 1.1), kann die Verbindung zum Joystickport entfallen. Ist die Stromversorgung am PBI nicht vorhanden, muss Stromversorgungskabel an Pin 7 des Joystickports 2 angesteckt werden. Pin 7 ist in der nebenstehenden Zeichnung markiert.



**Achtung:** Das Stromversorgungskabel darf im laufenden Betrieb auf keinen Fall abgesteckt werden, ansonsten können der ATARI und/oder der Freezer Schaden nehmen. Sollte das Stromversorgungskabel versehentlich abgesteckt worden sein so muss der Atari sofort ausgeschaltet werden.

### 1.3.2 Atari 800 XE, 65 XE und 130 XE

Einfach den Freezer hinten am Atari in Modulschacht und ECI-Anschluss einstecken.

## 1.4 Inbetriebnahme des Freezers

### 1.4.1 Grundkonfiguration setzen

Alle Schalter am Freezer nach links schieben damit wird folgende Grundkonfiguration aktiviert:

- „**CartEmu**“: OFF, Cartridge-Emulation aus
- „**FlashWrite**“: OFF, Schreibfreigabe für Flash-ROM aus
- „**OldOS**“: OFF, Oldrunner aus
- „**Stereo**“: OFF, Stereo-POKEY-Modus aus (nicht vorhanden)
- „**Ramdisk**“: OFF, 512k RAM-Erweiterung aus

### 1.4.2 Freezer starten

Den Atari Computer einschalten und sonstige Hardware ausgeschaltet lassen. 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 anstatt der **READY** Meldung die Meldung **ATARI COMPUTER - MEMO PAD**. Falls die Cartridge-Emulation eingeschaltet sein sollte, erscheint der grüne Bildschirm der mit dem **TURBO CARTRIDGE**

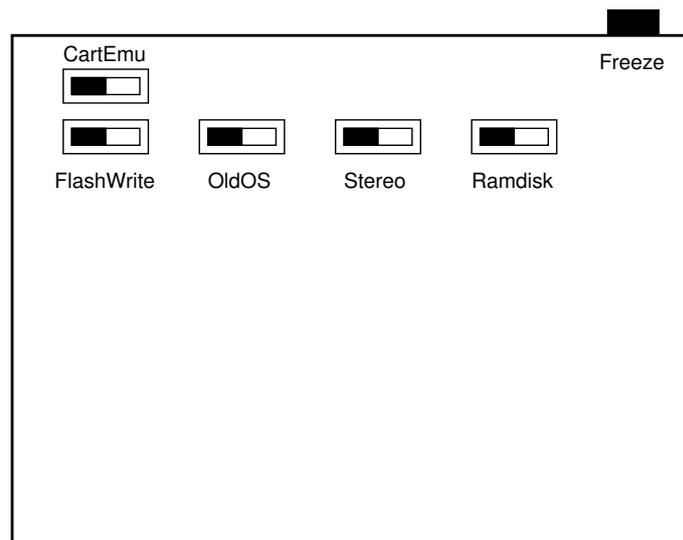


Abbildung 1.2: Schalter am TURBO FREEZER

Menü der Cartridge-Emulation. In diesen Fällen die Grundkonfiguration gemäß Abschnitt 1.4.1 überprüfen.

Ist die **READY** Meldung erschienen, einfach den Knopf „**Freeze**“ oben rechts drücken, so dass das Freezer-Hauptmenü erscheint. Tut sich nichts, ist entweder der Freezer schlecht eingesteckt, das Stromversorgungskabel (wenn vorhanden) nicht richtig angeschlossen oder der Freezer ist kaputtgegangen. Anschließend im Freezer-Hauptmenü 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 einsatzbereit.

Sollte der Freezer aus irgendeinem Grund nicht richtig funktionieren, diesen nicht gleich zurücksenden, sondern erst einmal einen Brief oder eine E-Mail mit einer Beschreibung der Symptome senden. Es ist aufgrund des 100%-igen Endabnahmetests jedes Freezers praktisch ausgeschlossen, dass es am Freezer liegt. Es liegt wahrscheinlich an einem Einbaufehler oder am Atari selbst. Es ist besser, diese Möglichkeiten durch eine Rückfrage auszuschließen, bevor der Freezer selbst zur Überprüfung eingeschickt wird. Dies vermeidet unnötige Wartezeiten und Kosten.

### 1.4.3 Systemdiskette erstellen

Gleich nach der erfolgreichen Installation des TURBO FREEZERS sollte die Systemdiskette erstellt werden. Auf der Systemdiskette sind das Flash-Programm das Beschreiben des Flash-ROMs durch den Atari sowie die Freezer-Software enthalten. Damit kann der TURBO FREEZER jederzeit wieder in den Auslieferungszustand zurückgesetzt werden. Im Auslieferungszustand enthält das Flash-ROM die dazu notwendige Software. Die Systemdiskette wird wie folgt erstellt:

1. Den Atari ausschalten und eine Diskettenstation anschließen.
2. Den Schalter „**CartEmu**“ nach rechts schieben, um Cartridge-Emulation zu aktivieren.
3. Nach dem Einschalten des Ataris sollte das Menü der Cartridge-Emulation erscheinen. Die Standardeinstellungen sind schon richtig gesetzt, sie müssen nur noch mit [RETURN] bestätigt werden.
4. Jetzt befindet man sich in der „TURBO FREEZER System Disk Writer“ Software. Eine leere Diskette ins Diskettenlaufwerk „D1:“ einlegen und [RETURN] drücken. Die Software formatiert nun die Diskette in mittlerer Dichte und schreibt die Freezer-Software darauf. Sollte dabei ein Fehler auftreten (z. B. auf Grund einer defekten Diskette), so wird die Meldung **ERROR** ausgegeben und man muss eine neue Diskette einlegen und das Programm neu starten. Hierzu die Rückfrage **restart program?** einfach mit [Y] bestätigen. Zur Sicherheit sollte man eine zweite Kopie der Systemdiskette erstellen, schreibschützen an einem sicheren Ort verwahren.
5. Nach dem erfolgreichen Erstellen der Systemdiskette den Atari abschalten und den Schalter „**CartEmu**“ wieder in die Ausgangsposition (nach links) zurück schieben.

Wenn alle Stricke reißen und man die Systemdiskette verliert oder versehentlich überschreibt (ja, so etwas kann schon mal passieren), gibt es im Internet immer noch unter <http://turbofreezer.horus.com> ein ATR-Diskettenimage der Systemdiskette zum Download.

### 1.4.4 Stereo-POKEY-Modus aktivieren (optional)

Falls der Atari eine Stereo-POKEY-Erweiterung enthält und diese im Stereo-POKEY-Modus ist, muss der Schalter „**Stereo**“ nach rechts geschoben werden. Dadurch speichert der Freezer auch die Hardwareregister des zweiten POKEYs im Shadow-RAM. Beim Aktivieren des Freezers wird der zweite POKEY deaktiviert. Bei Auftauen wird sein Zustand dann wieder korrekt wiederhergestellt.

Grundsätzlich gilt, dass die Position dieses Schalters genau mit der Konfiguration des Ataris übereinstimmen muss. Eine Änderung der Schalterstellung im laufenden Betrieb ist zwar erlaubt, das Shadow-RAM im Freezer enthält dann aber falsche Daten. Um das zu korrigieren, muss man [RESET] drücken, damit das Betriebssystem den Custom-Chip Bereich neu initialisiert. Alternativ ändert man die Schalterstellung nur bei ausgeschaltetem Atari.

### 1.4.5 Integrierte 512k RAM-Erweiterung aktivieren (optional)

Als besonderes Feature enthält der Freezer eine integrierte, batteriegepufferte 512k RAM-Erweiterung. Damit lässt sich jeder Atari so einfach wie bisher noch nie von 64k auf insgesamt 576k erweitern. Und die Daten in der RAM-Erweiterung bleiben durch die Batteriepufferung sogar nach dem Ausschalten erhalten.

Schiebt man den Schalter „**Ramdisk**“ nach rechts, wird die RAM-Erweiterung aktiviert. Eine eventuell vorhandene interne RAM-Erweiterung im Atari wird dabei automatisch deaktiviert.

### 1.4.6 Steckmodule am Atari XE verwenden

Da der Freezer an Atari XE Computern den Modulport blockiert, ist auf der XE Adapterplatine ein Modulanschluss vorhanden. Damit kann man trotz angestecktem Freezer weiterhin auch Steckmodule verwenden. Die Module sind dabei so einzustecken, dass die Beschriftung des Moduls in Richtung des Ataris zeigt. Man darf das Module keinesfalls verkehrt herum einstecken, ansonsten können Modul, Freezer und Atari Schaden nehmen.

Rund um den Modulanschluss sind Ausfräsungen vorhanden in die eine evtl. vorhandene „Plastikwanne“, wie sie in den Atari XL Computern am Modulschacht zu finden ist, eingesetzt werden kann. Dadurch entfällt dann das lästige manuelle Entriegeln mancher Module. Diese „Plastikwannen“ sind leider nicht einzeln erhältlich. Die einzige Möglichkeit eine zu bekommen ist das Schlachten eines defekten Atari XL Computers.

## 1.5 Erweiterte Konfigurationen für Lötexperten

**Warnung:** Alle folgenden Erweiterungen sind wirklich nur für Lötexperten geeignet. Wer kein solcher ist, sollte sich lieber einen solchen Experten suchen. Es lohnt sich weder, zu diesem Zweck eine extrem teurere Servicestelle aufzusuchen, noch lohnt es sich, durch Ungeschick den armen Atari zu ruinieren.

### 1.5.1 Interne Stromversorgung bereitstellen

Wer einen Atari 800 XL ohne Stromversorgung am PBI 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 beiden Fällen bleibt der Joystickport dann für seinen eigentlichen Zweck frei.

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.

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 Stromversorgungskabel des Freezers angelötet oder es wird von hier je eine Leitung zu Pin 47 und Pin 48 des PBI geführt. Dies sind die vorletzten Kontaktflächen vor Pin 49 und 50, je eine auf der Oberseite und eine auf der Unterseite der Platine. Das Anbringen an der Unterseite ist schwieriger, weil dazu der Atari vollständig zerlegt werden muss. Zudem ist es knifflig, an den Enden der Kontaktflächen zu löten, ohne dass das Zinn auf den aktiven Teil der Kontaktfläche kriecht. Letzte muss in jedem Fall vermieden werden.

### 1.5.2 SYSTEM RESET-Taste einbauen

Für die Verwendung des Oldrunners und für das Einfrieren von Programmen die keine Interrupts verwenden 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. Beim Atari 800 XL sowie allen Atari XE Modellen, beim Atari 600 XL ist es R34, beim Atari 1200 XL ist es R7.

**Warnung: Zusätzlich der allgemeinen Warnung aus dem ersten Abschnitt gilt hier, dass auch mechanischen Arbeiten für den Einbau des Tasters notwendig sind. Diese erfordern zusätzliches Werkzeug und handwerkliches Geschick, um ein ansehnliches Resultat zu erreichen.**

### 1.5.3 Kompatibilität zu 1MB RAM-Erweiterungen herstellen

Der TURBO FREEZER verwendet die Refresh-Leitung, um den Atari anzuhalten. Daher kommt es leider zu Kompatibilitätsproblemen mit allen internen RAM-Erweiterungen, die eine eigene Refresh-Logik auf Basis des Refresh-Signals des ANTIC einsetzen. Dies betrifft hauptsächlich 1MB RAM-Erweiterungen wie z. B. die Newell RAM-Erweiterung. Die meisten 256k RAM-Erweiterungen sowie der erweiterte Speicher des Atari 130 XE sind davon nicht betroffen. Mit zwei zusätzlichen Bauteilen lässt sich dieses Problem jedoch sehr einfach beheben. Benötigt werden eine Kleinsignal Schottky Diode, z. B. vom Typ „BAT 85“ und ein 4.7kOhm Widerstand.

**Warnung: Zusätzlich der allgemeinen Warnung aus dem ersten Abschnitt gilt hier, dass der Pin des ANTIC mit besondere Vorsicht behandelt werden muss. Der Pin bricht sehr leicht, wenn man ihn zu stark biegt.**

Nach dem Öffnen des Ataris muss der Pin 8 am ANTIC hoch gebogen werden. Das Kabel, das von Pin 8 zur RAM-Erweiterung führt, vorher ablöten. Ist der ANTIC gesockelt, den ANTIC aus der Fassung ziehen und sehr vorsichtig den Pin hoch biegen. Es reicht aus den Pin so weit weg zu biegen, 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 hoch biegen.

Als erstes muss die Diode zwischen dem hoch gebogenen Pin 8 und Pin 8 des Sockels (bzw. der Platine) gelötet werden. Die Kathode (am Gehäuse der Diode mit einem dunklen Ring markiert) muss mit dem ANTIC, die Anode mit Sockel verbunden werden. 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 RAM-Erweiterung führt wieder an Pin 8 am ANTIC angelötet werden.

## 2 Einsatz

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 Kassette auskommen) und vollautomatisch sowie in Sekundenschnelle arbeiten. Daher wurde beim TURBO FREEZER darauf verzichtet, Sparmaßnahmen zu ergreifen, die durchaus möglich gewesen wären, wie etwa die teilweise Rekonstruktion der Hardwareregister durch Eingaben des Users oder das nachladen der Freezer-Software. Statt solche dubiosen Sparmaßnahmen zu ergreifen, wie sie teils bei den Freezern für andere Computersysteme zu beobachten sind, wurde beim TURBO FREEZER ohne Rücksicht auf den Aufwand das Beste geschaffen, was mit dem Stand der Technik möglich ist. Und der Aufwand ist, 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 FREEZERS nur begeistert sein. Um zum Beispiel ein Spiel einzufrieren, in der RAM-Erweiterung zu speichern und trotzdem drei Sekunden später weiterzuspielen, sind nur drei Tastendrucke nötig. Dasselbe gilt für das spätere Laden und Auftauen aus der RAM-Erweiterung.

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

Und es gibt noch einen wichtigen Punkt. Der TURBO FREEZER ist ein Freezer, bei dem ein DOS und ein Debugger eingebaut sind, Diese stehen jederzeit zur Verfügung können und verwendet werden, ohne dass das eingefrorene Programm dabei beschädigt wird. Volle Disketten oder sonstige fatale Vorkommnisse verlieren damit auch bei der Verwendung von Anwendungsprogrammen ohne DOS-Funktionen ihre Schrecken. Dies gilt ebenso für die üblen Bugs, die man ohne Einblick in die Hardwareregister und den

unversehrten Systemzustand nie findet.

## 2.1 Freezer-Hauptmenü aufrufen

Nach dem Drücken des Knopfes „Freeze“ 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 der es eingefroren wurde. Im Freezer-Hauptmenü stehen folgende Funktionen zur Verfügung.

[Leertaste]

Eingefrorenes Programm auftauen

[RESET] Kaltstart durchführen und Freezer deaktivieren

[S] Eingefrorenes Programm auf Diskette oder Kassette speichern

[F] Eingefrorenes Programm in das Freezer-RAM speichern

[R] Eingefrorenes Programm in die RAM-Erweiterung speichern

[E] Eingefrorenes Programm von Diskette oder Kassette laden und starten

[C] Eingefrorenes Programm aus dem Freezer-RAM laden und starten

[X] Eingefrorenes Programm aus der RAM-Erweiterung laden und starten

[W] Eingefrorenes Programm mit Programm im Freezer-RAM tauschen und starten

[A] Eingefrorenes Programm mit Programm in der RAM-Erweiterung tauschen und starten

[Z] RAM unter dem OS-ROM löschen

[D] Debugger starten

[K] Menü der Cartridge-Emulation starten

[Shift]+[K]

Menü der Cartridge-Emulation starten, dabei RAM nicht löschen (Warmstart)

[Shift]+[Leertaste], [E], [C], [X], [W], [A]

Wie oben, dabei vor dem Auftauen die Player/Missile-Kollisionsregister löschen

[Control]+[E], [C], [X], [W], [A]

Wie oben, aber eingefrorenes Programm nur Laden und nicht starten

## 2.2 Programm einfrieren und auftauen

Im Prinzip kann jedes Programm an jeder beliebigen Stelle eingefroren werden, also auch während einer Disketten- oder Kassettenoperation. Zu empfehlen ist das jedoch nicht, da diese Operation dann unvollständig bleibt. Nach dem Auftauen des Programms besteht zwar für das Betriebssystem die Möglichkeit, die Diskettenoperation zu wiederholen, dies

ist aber nicht sicher. Die Wiederholung einer Kassettenoperation ist dem Betriebssystem überhaupt nicht möglich, da das Kassettenlaufwerk dazu zu „dumm“ ist. Demnach ist es besser, Programme nur einzufrieren, wenn gerade keine Operationen mit der Peripherie erfolgen.

Es kann in äußerst seltenen Fällen vorkommen, dass nach dem Drücken des Knopfes „Freeze“ nichts passiert, das Programm also ungerührt weiterläuft. Dies ist dann, der Fall, wenn das gar keine Interrupts verwendet. Möglich ist das natürlich nur bei ganz einfachen Programmen wie z. B. konvertierten Apple-Programmen, die die Fähigkeiten des Ataris praktisch nicht nutzen. Um solche Programme dennoch einzufrieren kann man den Atari, wie im Abschnitt 1.5.2 beschrieben, mit einer [SYSTEM RESET]-Taste erweitern. Der hiermit verbundene Interrupt kann nämlich auf keinen Fall unterdrückt werden, so dass es gegen den TURBO FREEZER keine Gegenmaßnahme mehr gibt.

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. Hier einfach [Shift]+[E] / [C] / [X] / [W] / [A] drücken.

Nach dem normalen Laden eines eingefrorenen Programms von Diskette, Kassette, Freezer-RAM oder RAM-Erweiterung wird dieses automatisch aufgetaut. Lädt man ein eingefrorenes Programm mit [Control]+[E] / [C] / [X] / [W] / [A], 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 usw. sucht. Hierzu kann man das Programm abspeichern direkt nach dem Einfrieren, 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]/[C]/[X] 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.

## 2.3 Eingefrorenes Programm abspeichern

Mit den Funktionen [S], [F], [R] können eingefrorene Programme auf externe Massenspeicher wie Kassette oder Diskette (Taste [S]), in die RAM-Erweiterung (Taste [R]) oder in das Freezer-RAM (Taste [F]) weggespeichert werden. Bei [S] wird mittels eines Untermenüs gefragt, ob die Speicherung, als Bootkassette bzw. Bootdiskette oder als einzelne Datei erfolgen soll. Vor dieser Eingabe sollten die Kassette bzw. Diskette eingelegt werden. Beim Speichern als einzelne Datei auf einer Diskette wird mit einem weiteren Menü der Dateiname erfragt.

**Achtung:** Beim Abspeichern als Bootdiskette wird direkt auf die Diskette in „D1:“ geschrieben. All eventuell auf der Diskette vorhandenen Daten gehen dabei verloren.

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

Um bei den Operationen die größtmögliche Geschwindigkeit zu erreichen wird das Freezer-Hauptmenü während der Ausgabeoperationen hin und wieder ausgeschaltet. Dadurch, dass in Kauf genommen wurde, dass das Bild beim Wiedereinschalten „hüpfen“ kann, dauert z.B. das Speichern in die RAM-Erweiterung nur halb so lang, was die unsaubere Methode rechtfertigt.

### 2.3.1 Ladeprogramm für Bootdisketten anpassen

Wird ein eingefrorenes Programm als Bootdiskette gespeichert, so schreibt der Freezer ein Ladeprogramm mit auf die Diskette. Dies funktioniert naturgemäß nicht mit allen Programmen, denn das Ladeprogramm braucht etwa 2k Speicherplatz. Benötigt das eingefrorene Programm den vollständigen Speicher des Ataris, 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 dennoch 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 Turbo-Freezer 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. Hierbei ist zu beachten, dass nur Werte von \$05 bis \$C9 und von \$D8 bis \$F9 erlaubt sind. Wenn andere Werte benutzt werden, erhält man beim Booten der Diskette einen `BOOT ERROR`. Um zum Beispiel die Startadresse des Ladeprogramms auf \$F0 gesetzt werden soll, sind folgende Kommandos Debugger einzugeben:

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

### 2.3.2 Nachladendes Programm einfrieren

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. Selbst mit einem kopierfähigen Floppy-Speeder tritt dabei aber heutzutage das Problem auf, dass bei weitem nicht alle neuen Programme kopiert werden können.

Der TURBO FREEZER entfernt einen auf der Originaldiskette befindlichen Kopierschutz nicht, sondern speichert lediglich den Programmzustand ab. Um diesen später wieder

beliebig oft auftauen zu können, muss eine kopiergeschützte Originaldiskette wieder in das Diskettenlaufwerk eingelegt werden, bevor das Programm aufgetaut wird. Bei Verwendung einer Diskette zur Speicherung des Spielstandes sollte man die Originaldiskette mit einem 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. ein Adventure), empfiehlt es sich, diese zu kopieren und die Kopien zu verwenden, um die Originaldisketten zu schonen.

## 2.4 Zwischen eingefrorenem und gespeichertem Programm wechseln

Mit der Wechselfunktion im Freezer-Hauptmenü kann zwischen dem gerade eingefrorenen Programm und einem im Freezer-RAM (Taste [W]) oder in der RAM-Erweiterung (Taste [A]) bereits abgelegten Programm gewechselt werden. Damit kann man sehr schnell zwischen zwei verschiedenen Programmen wechseln. Hierbei sollte man darauf achten, dass sich in den RAM-Erweiterung bzw. dem Freezer-RAM auch wirklich ein eingefrorenes Programm befindet, da der Atari ansonsten beim Auftauen abstürzt.

Die Zusatz Tasten [Shift] für das Löschen der Player/Missile-Kollisionsregister sowie [Control] um das automatische Starten zu verhindern stehen hier ebenso wie bei den normalen Funktionen zum Auftauen zur Verfügung.

## 2.5 RAM unter dem OS-ROM löschen

Da es Programme gibt, die 64k RAM benötigen und auch belegen, muss der Freezer auch den RAM-Bereich unter dem OS-ROM berücksichtigen. Nach dem Einschalten des Computer befinden sich in diesem Bereich nutzlose Zufallsdaten. Da sich diese 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 Verkleinerung der Dateien von 25% bis 50% und dementsprechend geringere Ladezeiten.

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

## 2.6 Debugger und DOS starten

Mit der Funktion [D] wird das eingebaute Debugger mit DOS-Funktionen aktiviert. Dieser verwendet eine Kommandozeile und den vollständigen Bildschirm, sodass er nicht mit Menüs auskommen würde. Trotzdem wird auch für diese Funktion kein RAM im Atari belegt oder verändert. Die Zustände der eingefrorenen Hardwareregister bleiben

selbstverständlich ebenfalls erhalten. Die vollständige Beschreibung des Debuggers und der DOS-Funktionen findet sich im Kapitel 3.

### 2.7 Menü der Cartridge-Emulation aufrufen

Mit der Funktion [K] wird das Menü der Cartridge-Emulation aufgerufen. Diese Funktion ist nur dann verfügbar, wenn der Schalter „**CartEmu**“ oder der Schalter „**FlashWrite**“ (oder beide) eingeschaltet sind, d.h. rechts stehen. Ansonsten ist die Cartridge-Emulation komplett deaktiviert und das Menü der Cartridge-Emulation kann natürlich nicht eingeblendet werden. Die vollständige Beschreibung der Cartridge-Emulation findet sich im Kapitel 6. Beim Aufrufen der Cartridge-Emulation führt der Freezer folgende Schritte durch:

- Das Freezer-Hauptmenü wird verlassen
- Das Menü der Cartridge-Emulation wird aktiviert
- Die IRQs und NMIs werden abgeschaltet
- Das OS-ROM wird eingeschaltet
- Die Rücksprungadresse wird auf \$E477 gesetzt (Kaltstart)
- Das normale Auftauen wird durchgeführt (entspricht der [Leertaste])

Mit der Funktion [Shift]+[K] wird ein anstatt des Kaltstart ein Warmstart (Rücksprung nach Adresse \$E474) ausgelöst. Dies kann zum Beispiel dann nützlich sein, wenn man den Inhalt des internen RAM im Atari erhalten möchte wenn man ein Modul startet.

## 3 Debugger und DOS

Vom Freezer-Hauptmenü gelangt man mit der [D] Taste in den eingebauten Debugger mit DOS Funktionen.

Die Eingabe von Kommandos 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 Pfeiltasten [CURSOR UP] und [CURSOR DOWN] kann man die zuletzt eingegebenen Kommandos wieder aufrufen. Der Freezer merkt sich hierbei die 4 letzten Kommandos.

Der Debugger unterstützt auch die sonst leider viel zu selten verwendete [HELP] Taste. Durch einen Druck darauf wird eine kurze Hilfe mit einer Übersicht aller Kommandos angezeigt. Mit den Tasten [1], [2]... springt man zu den einzelnen Hilfeseiten, mit [DEL] geht man eine Seite zurück, mit [Leertaste] oder [HELP] eine Seite vor. Jede andere Taste beendet die Hilfe. Der Inhalt der Kommandozeile bleibt erhalten, man kann die Hilfe aufrufen. Das heißt, man kann sie auch während der Eingabe eines Kommandos aufrufen wenn man die genaue Syntax nicht mehr im Kopf hat.

### 3.1 Disk Operating System (DOS)

Wer kennt nicht die folgende Situation: 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` Fehler. Da ist guter Rat teuer. Ein DUP-Aufruf via DOS führt in der Regel zum Verlust des Programms, denn „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 Kommandos zur Verfügung stehen. Es ist dann kein Problem mehr, solche kritischen Situationen zu meistern, um anschließend das eingefrorene Programm einfach fortzusetzen. Das im TURBO FREEZER eingebaute DOS enthält alle für diesen Zweck notwendigen Funktionen. 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 Floppies sowie die XF-551 mit der hohen SIO-Übertragungsrate angesprochen werden. Es kann aber auch eine beliebig anders getunte oder serienmäßige Floppy verwendet werden.

Ein Diskettenkommando besteht entweder aus einem drei Zeichen langen Kommando allein oder aus dem Kommando, einem oder mehreren Leerzeichen und einem Dateinamen. Manche Kommandos erlauben die Angabe einer Option, die unmittelbar am Ende des Kommandos, 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. Dateinamen dürfen die Wildcards „\*“ für eine beliebige Zeichenfolge sowie „?“ für ein beliebiges Zeichen enthalten. Fehlerhafte Kommandos werden einfach ignoriert und führen nicht zu einer Fehlermeldung. Eine solche kann nur auftreten, wenn bei der Ausführung selbst ein Fehler auftritt.

#### 3.1.1 Ausführen von Kommandos

Lässt man den Präfix **D:** im Dateinamen weg, so wird automatisch **D1:** verwendet. Eine RAM-Disk wird nicht unterstützt, da die Implementierung immer vom jeweils eingesetzten DOS oder RAM-Disk-Treiber abhängig ist. Es werden nur die SIO-Geräte **D1:** bis **D8:** unterstützt. PBI Geräte können aus technischen Gründen leider auch nicht angesprochen werden. Wildcards dürfen auch in Zieldateinamen angegeben werden. Es wird dann einfach die erste übereinstimmende Datei angesprochen. Die einzige Ausnahme bildet das Kommando **REN** zum Umbenennen von Dateien, das weiter unten beschrieben wird.

Die Kommandos **DEL**, **LOC**, **UNL** und **REN** zur Manipulation von Verzeichniseinträgen können auch mehrere Dateien hintereinander bearbeiten. Um die Gefahr ungewollter Aktionen zu senken, wird ohne Angabe einer Option am Ende des Kommandos 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.

#### 3.1.2 Kommandoübersicht

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

<b>DIR</b>	Verzeichnis aller Dateien anzeigen
<b>DIR dateiname</b>	Verzeichnis bestimmter Dateien anzeigen
<b>DEL dateiname</b>	Datei löschen
<b>FMS</b>	Formatieren in einfacher Dichte
<b>FME</b>	Formatieren in mittlerer Dichte
<b>FMD</b>	Formatieren in doppelter Dichte, erfordert spezielle oder erweiterte Floppy
<b>LOC dateiname</b>	Datei sperren
<b>UNL dateiname</b>	Datei entsperren

---

```

REN dateiname,neuename
    Datei umbenennen
LOA dateiname
    Objektdatei laden
LOA dateiname/N
    Objektdatei Ladeadresse(n) anzeigen aber nicht laden
LOA dateiname,start
    Objektdatei laden nach Adresse start, COM-Header ignorieren
SAV dateiname,start,ende
    Objektdatei erzeugen mit Speicherinhalt von Adresse start bis einschließlich
    ende
SAV dateiname/N,start,ende
    Datei ohne COM-Header (RAW-Datei) erzeugen mit Speicherinhalt von
    Adresse start bis einschließlich ende

```

### 3.1.3 Besonderheiten mancher Kommandos

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

Bei Umbenennen von Dateien mit dem Kommando **REN** ist es möglich, auch der Angabe des neuen Namens beliebige Wildcards zu verwenden. Anstelle der Wildcards treten dann die entsprechenden Buchstabenfolgen aus dem vorherigen Namen der Datei. Dies ermöglicht Gruppen von Dateien effizient zu bearbeiten, ohne auf den Primär- bzw. Sekundärnamen (also die Dateierweiterung) als Gruppenkriterium beschränkt zu sein.

Objektdateien können mit **LOA** geladen werden. Der Speicherbereich, in den die Datei geladen wird, wird dabei angezeigt. Bei zusammengesetzten (Compound) Dateien werden alle Speicherbereiche angezeigt. Dabei steht der gesamte 64k-Adressraum des Rechners zur Verfügung. Insbesondere können sogar die Hardwareregister geladen werden, die zu diesem Zeitpunkt ja eingefroren sind. Gestartet wird das geladene Programm aber **nicht**, um eine Fehlbedienung der Memory-Management Logik des Freezers durch mehrstufig ladende Programme auszuschließen.

Hängt man das Ladekommando die Option **/N** an, wird die Datei nicht geladen, aber die Speicherbereiche werden angezeigt, in die die Datei geladen werden würde. Das ist nützlich, wenn man nur wissen möchte, wohin eine Objektdatei geladen würde, z. B. **LOA FONT.DAT/N**. Mit dem Kommando **LOA** kann man auch Rohdaten (ohne COM-Header) einlesen. Dabei muss man angeben, wohin die Daten geladen werden sollen, z. B. **LOA FONT.DAT,8000**.

### 3.1.4 Fehlermeldungen

Bei Verwendung des DOS und der Lade- und Speicherfunktionen des Freezers für eingefrorene Programme können Fehler auftreten. Die zugehörigen Fehlermeldungen erfolgen im Klartext in englischer Sprache.

**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 Diskette formatieren

**BAD DISK I/O**

Das Kommando kann durch einen Busfehler oder Diskettenfehler oder durch eingeschalteten Schreibschutz nicht ausgeführt werden

**NO DRIVE** Die Diskettenstation ist nicht ansprechbar

**DISK FULL**

Die Diskette ist voll

**FILE LOCKED**

Die Datei ist gesperrt

**DIRECTORY FULL**

Das Verzeichnis ist voll (maximal 64 Dateien pro Diskette)

## 3.2 Debugger

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

### 3.2.1 Zugriff auf Speicher und Hardwareregister

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 Stopp 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 hat. Selbst wenn der Debugger aufwendiger ausgestattet ist (durch eigenes RAM), trasht er immer noch die Hardwareregister von ANTIC, POKEY und GTIA. Es gibt keine OS-Schattenregister für die Player-Missile Graphik und die Sounderzeugung die existierenden OS-Schattenregister 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 (write-only) 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 eingebauten Debugger kommt man an alle Informationen über den Systemzustand heran. Die Hardwareregisterinhalte (d.h. der echte Inhalt, nicht die zurückgegebenen Statuswerte) können im I/O Bereich gelesen und auf Wunsch geändert werden. Dies kann erfolgen 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 dem RAM unter dem OS-ROM), ohne dass irgendetwas getrasht wird oder abstürzt.

Hinzu kommen verschiedene Vorzüge, die sich aus der Arbeitsumgebung des TURBO FREEZER 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.

Die Software des originalen TURBO FREEZER von 1987 brachte – unter anderem auch platzbedingt – einige Nachteile mit sich. Man konnte nur mit dem eingefrorenen Systemzustand arbeiten, Änderungen an den Hardwareregistern zeigten erst nach dem Auftauen Wirkung. Dadurch war es z. B. schwierig mit der RAM-Erweiterung zu arbeiten. Durch die fortlaufenden Erweiterungen an der Software wurde der Großteil der ursprünglichen Einschränkungen mittlerweile beseitigt. Im Direct-I/O-Modus können Erweiterungen im Custom-Chip Bereich und Bankswitching Module direkt vom Debugger aus angesprochen werden und das PB Kommando ermöglicht die Steuerung des Zugriffs auf die RAM-Erweiterung sowie das OS-ROM bzw. das RAM unter dem OS-ROM.

### 3.2.2 Kommandoübersicht

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 Kommandos genügt. Bei der Eingabe von Kommandos dürfen keine Leerzeichen eingefügt werden, die Kommandos müssen unmittelbar nach dem Prompt beginnen. Zur Eingabe steht nur die Eingabezeile zur Verfügung, dafür werden im Ausgabebereich alle Änderungen mitprotokolliert. Alle Eingaben erfolgen in hexadezimaler Schreibweise. Durch Auslassen eines Wertes können Speicherstellen oder Register ohne Änderung des Inhalts übersprungen werden.

Beispiel: `C100<0A, ,4D` ändert den Inhalt in Speicherstelle \$100 auf \$0A und in Speicherstelle \$102 auf \$4D. Danach steht der interne Adresszeiger auf \$103. Der Inhalt der Speicherstelle \$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] angehalten und [Q] fortgesetzt werden.

- Q            Zum Freezer-Hauptmenü springen
- D **start**    8 Bytes plus ATASCII Zeichen ab Adresse **start** anzeigen
- D **start**,    128 Bytes ab Adresse **start** anzeigen

D *start,ende* Speicherinhalt anzeigen von Adresse *start* bis *ende*

I *start* 8 Bytes plus Zeichen im internen Atari Bildschirmcode ab Adresse *start* anzeigen

I *start,* 128 Bytes ab Adresse *start* anzeigen

I *start,ende* Speicherinhalt anzeigen von Adresse *start* bis *ende*

L *start* Disassemblieren ab Adresse *start*, eine Bildschirmseite anzeigen

L *start,ende* Disassemblieren von Adresse *start* bis *ende*

DL *start* Display List anzeigen ab Adresse *start*

DL *start,ende* Display List anzeigen von Adresse *start* bis *ende*

C *start<byte1,byte2...* Speicherinhalt ändern ab Adresse *start*

VEC OS-Vektoren anzeigen

HAT Handler Tabelle anzeigen

M Speicherbelegungsübersicht anzeigen

PB Wert des Registers PORTB für den Speicherzugriff (steuert Zugriff auf RAM-Erweiterung, OS, BASIC) anzeigen

PB<wert Wert des Registers PORTB für den Speicherzugriff auf *wert* ändern

DIO Direct-I/O-Modus anzeigen

DIO<wert Direct-I/O-Modus aktivieren/deaktivieren, mit Vorsicht anwenden

R Register anzeigen

R<wert1,wert2... Register ändern

G *start* Rücksprungadresse für das Auftauen auf *start* ändern  
/*start,ende/wert...* Suche *wert...* im Speicherbereich von Adresse *start...ende*

BM *start,ende,ziel* Verschiebe Speicherbereich von *start...ende* nach *ziel...ziel+(ende-start)*

BS *start,ende,wert* Setze Speicherbereich von *start...ende* auf *wert*

BC *start,ende,start2* Vergleiche Speicherbereich von *start...ende* mit Speicherbereich ab *start2*

PR<wert Druckerausgabe aktivieren/deaktivieren

; TEXT Text oder Kommentar ausgeben

---

SR nummer	Sektor nummer lesen
SW nummer	Sektor nummer schreiben
SIO ...	SIO-Kommando ausführen
SIOR	Highspeed SIO-Routine zurücksetzen
V	Interrupt und Wert des Registers VCOUNT zum Zeitpunkt des Einfrierens anzeigen
V<wert	Wert des Registers VCOUNT zum Zeitpunkt des Auftauens ändern
a COMMAND...	AtariSIO Remote Control
VER	Freezer-Softwareversion anzeigen

Anmerkung: Mit Ausnahme der Leerzeichen direkt nach dem Kommando müssen die hier aus Gründen der Lesbarkeit eingefügten Leerzeichen bei der Eingabe weggelassen werden. Der Program Counter (PC) kann aufgrund der besonderen internen Behandlung nicht mit dem Kommando R< geändert werden, sondern nur mit G.

### 3.2.3 Werte eingeben (hexadezimal, dezimal, interner Code)

Adressen und Bytewerte werden üblicherweise in hexadezimaler Schreibweise eingegeben. Um Dezimalzahlen zu verwenden, muss die Adresse bzw. der Bytewerte mit einem Prozent-Zeichen % begonnen werden. Folgendes Kommando entspricht der Eingabe „POKE 710,10“ in BASIC.

```
C%710<%10
```

Bytewerte können auch in ATASCII oder als internen Atari Bildschirmcode eingegeben werden. Für ATASCII muss ein einfaches Anführungszeichen ('), für Screen Code ein Klammeraffe (@) vorangestellt werden.

```
C0600<'H,'a,'l,'l,'o
C9C40<@S,@c,@r,@e,@e,@n
```

### 3.2.4 Vektoren eingeben

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 effektive Adresse verwendet. Damit kann man z. B. sehr schnell die aktuell verwendete Display-List ausgeben lassen.

```
DL*230
DL*\%560
```

### 3.2.5 Auf erweiterten Speicher, RAM, ROM zugreifen

Der Debugger bietet beim Speicherzugriff volle Kontrolle über die Speicherverwaltungsfunktionen der Atari XL/XE Computer. Die Speicherverwaltungseinheit kann über das Register PORTB (\$D301) der PIA gesteuert werden. Somit kann man problemlos in die Speichererweiterung (\$4000...\$7FFF) oder das RAM unter dem OS-ROM (\$C000...\$CFFF,\$D800...\$FFFF) auslesen und dort auch Daten ändern.

Die PORTB Steuerung bezieht sich dabei ausschließlich auf den Debugger. Änderungen an der Steuerung wirken sich nicht auf das angehaltene Programm aus. Möchte man den Zustand nach dem Auftauen ändern, so muss der (eingefrorene) Wert von PORTB mit `C D301<byte` gesetzt werden.

Beim Aktivieren des Freezers wird die PORTB Steuerung auf den aktuellen PORTB Wert gesetzt. Das heißt, man sieht im Debugger den exakt gleichen Zustand wie das angehaltene Programm. Die Steuerung kann mit dem Kommando PB abgefragt bzw. geändert werden:

PB           Aktuellen PORTB Wert für Speicherzugriff anzeigen

PB<wert    Neuen PORTB Wert für Speicherzugriff setzen

Möchte man z. B. auf das RAM unter dem OS-ROM zugreifen, gibt man `PB<FE` ein um Bit 0 von PORTB auf 0 zu setzen. Mit `PB<E3` erhält man Zugriff auf die erste Bank der Speichererweiterung im Atari 130 XE. Der Wert von PORTB zum Zeitpunkt des Einfrierens kann jederzeit mit `D D301` abgefragt werden – vorausgesetzt man hat ihn nicht schon per `C D301<wert` geändert.

Beim Zugriff auf die eingefrorenen PIA-Register (\$D3xx) ist eine Besonderheit des Freezers zu beachten. Normalerweise wiederholen sich im Atari die Werte der PIA-Register alle 4 Bytes. Im Debugger werden jedoch 8 Bytes angezeigt.

\$D300, \$D301

sind die gewohnten Register PORTA und PORTB

\$D302, \$D303

enthalten die Datenrichtungsregister für PORTA und PORTB. Diese sind normalerweise zugreifbar über PORTA bzw. PORTB wenn Bit 2 von PACTL / PBCTL auf 0 steht.

\$D304, \$D305

enthalten die Werte von PACTL bzw. PBCTL die normalerweise über die Adresse \$D302 bzw. \$D303 zugreifbar sind.

\$D306, \$D307

sind unbenutzt

### 3.2.6 Display List anzeigen

Das Kommando DL die ANTIC-Befehle ab der angegebenen Speicheradresse als die im Datenblatt zum ANTIC beschriebenen Mnemonics aus.

BLK x      x Leerzeilen

CHR x      Text Modus x

MAP x	Grafik Modus x
JMP adr	Sprung zur angegebenen Adresse
JVB adr	Sprung bei dem zuvor auf den Vertical Blank gewartet wird

Neben dem Mnemonic werden, falls vorhanden, noch folgende Optionen ausgegeben:

LMS adr	Memory Scan Counter (Zeiger für die aktuelle Bildschirmspeicheradresse) laden
H	Horizontales Scrolling aktiviert
V	Vertikales Scrolling aktiviert
I	Display List Interrupt (DLI) auslösen

### 3.2.7 Speicher durchsuchen

Die Suchfunktion kann auf verschiedene Arten verwendet werden. Mit der Variante `/start/wert1,wert2...` wird ab der angegebenen Adresse nach der angegebenen Bytefolge gesucht. 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 dem L Kommando ab der gefundenen Speicheradresse disassemblieren. Mit dem `/` Kommando 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 mit dem Befehl „STA \$D4xx“ ein Byte in den ANTIC schreibt.

Die Bytefolge 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 bewirkt, dass der Inhalt des Speichers zunächst mit Bit-Maske AND verknüpft und danach mit dem Werte aus der Bytefolge verglichen.

Gibt man zusätzlich zur Startadresse eine Endadresse mit an, so werden alle Adressen im angegebenen Bereich ausgegeben, welche die Bytefolge enthalten. Die Suche stoppt also nicht bei der ersten gefundenen Adresse, z. B. `/start,ende/wert1,wert2...`. Oft möchte man den gesamten Speicher durchsuchen. Deshalb gibt es dafür es eine besonders kurze Variante `//wert1,wert2...`, die identisch zu `/0000,FFFF/wert1,wert2...` ist.

Bei längeren Bytefolgen und einem großen Speicherbereich kann die Suche mehrere Sekunden dauern. Mit der [BREAK] Taste kann die Suche jederzeit unterbrochen werden. Mit dem `/` Kommando kann sie dann wieder fortgesetzt werden.

### 3.2.8 SIO-Kommandos ausführen

Die Kommandos `SW`, `SR`, `SIO` sowie das eingebaute DOS verwenden einen internen Sektorpuffer. Dieser wird vom Memory-Manager an die Adresse \$D700 im eingefrorenen Adressraum gespiegelt, wo eine Bearbeitung mit dem Debugger erfolgen kann. Physikalisch ist er an der Adresse nicht vorhanden. Um einen Sektor von einem anderen Laufwerk als D1: zu lesen bzw. ihn dorthin zu schreiben muss man einfach `Dx:` direkt vor die Sektornummer stellen.

SR 100  
SR D2:200  
SW D3:300

Mit dem Kommando `SIO` können beliebige SIO-Kommandos genauso wie über den SIO-Vektor `$E459` ausgeführt werden. Die Werte haben die gleiche Bedeutung wie die Speicherstellen `$0300...$030B` im Atari. Die Werte für die Länge `length` darf maximal `$0100` sein.

`SIO device, unit, command, direction, timeout, length, daux`

Das Kommando `SIO` merkt sich die zuletzt eingegebenen Parameter, so dass bei darauffolgenden `SIO` Kommandos Teile der Parameter weggelassen werden können. Lässt man alle Parameter weg, so wird der letzte `SIO` Kommando wiederholt. Zu Beginn sind die Parameter auf „Get Status“ von `D1`: gesetzt, sodass nichts Schlimmes passieren kann wenn man einfach nur `SIO` ohne Parameter eingibt. Man sollte mit dem Kommando `SIO` allerdings stets sehr vorsichtig umgehen und sicherheitshalber die Parameter mehrfach überprüfen, ansonsten hat man sehr schnell die falsche Diskette formatiert.

### 3.2.9 Interrupt und VCOUNT anzeigen

Der Freezer kann beim Auftreten eines Interrupts das laufende Programm einfrieren. Hierzu blendet der Freezer kurzzeitig einen eigenen Wert an der Stelle des Interrupt-Vektors ein und zwingt die CPU so den Freezer aufzurufen. Das Kommando `V` gibt den Typ des Interrupts (NMI oder IRQ) an, mit dem der Freezer aktiviert wurde sowie die originale Adresse des zugehörigen Interrupt-Vektors. Zusätzlich noch den Wert von `VCOUNT` beim Start der Freezer-Software sowie der errechnete Wert von `VCOUNT` zum Zeitpunkt des Interrupts angezeigt. Dieser Befehl wurde hauptsächlich zum Testen der Freezer-Software implementiert und dürfte für die meisten Benutzer 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 in den Adressbereich der CPU ein und „verbiegt“ das High-Byte des Interrupt-Vektors zu Adresse des ROMs, so dass die Freezer-Software gestartet wird. Da das Low-Byte des Vektors nicht verändert wird, enthält die Freezer-Software zu Beginn eine Page voller „NOP“ Befehle. Es kann nun also passieren, dass der Atari etliche dieser „NOP“ Befehle abarbeiten muss, bevor die eigentliche Freezer-Software gestartet wird und `VCOUNT` ausgelesen werden kann.

Anhand des originalen Interrupt-Vektors versucht die Freezer-Software nun zu berechnen, wie viel Zeit mit „NOP“ Befehlen verbracht wurde und passt den `VCOUNT` Wert dementsprechend an. Hierdurch kann 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 mit `V<wert` den `VCOUNT` Wert von Hand setzen.

### 3.2.10 Direct-I/O-Modus aktivieren

Änderungen in den Hardwareregistern äußern sich normalerweise erst nach dem Auftauen, denn beim TURBO FREEZER wird der eingefrorene Adressraum manipuliert, und nicht, wie bei konventionellen Debuggern, der reale Adressraum. Für Profis gibt es beim TURBO FREEZER aber die Möglichkeit, dieses Verhalten abzuschalten und den „Direct-I/O-Modus“ zu aktivieren.

DI0       Aktuellen Zustand anzeigen: 0 = deaktiviert (Standard), 1 = aktiviert  
 DI0<0     Direct-I/O-Modus deaktivieren  
 DI0<1     Direct-I/O-Modus aktivieren

**Achtung:** Bei aktiviertem Direct-I/O-Modus wird komplett am Freezer gearbeitet und man hat gute Chancen ungewollt Fehler zu machen. Da der Freezer im Direct-I/O-Modus die Änderungen an den Hardwareregistern nicht mitbekommt, sollte man dieselben Änderungen auch noch mit deaktiviertem Direct-I/O-Modus vornehmen. Ansonsten werden die Register beim Auftauen wieder überschrieben.

### 3.2.11 Versionsnummer anzeigen

Mit dem Kommando VER werden die Versionsnummer und das Datum der Freezer-Software im Format 3.10 2012-11-09 ausgegeben. Für den Fall, dass das Flash-ROM keine Freezer-Software enthält, wird anstatt der Versionsnummer n/a ausgegeben.

### 3.2.12 OS-Vektoren anzeigen

Mit dem VEC Kommando kann man die wichtigsten OS-Vektoren aus den Pages 0 und 2 übersichtlich anzeigen lassen. Es wird jeweils der symbolische Name des Vektors, seine Adresse und seinen Wert ausgegeben.

```
DOSINI 000C: 0000   CASINI 0002: FFFF
DOSVEC 000A: F223
VPRCED 0202: COCD   VINTER 0204: COCD
...
```

### 3.2.13 Handler-Adressen-Tabelle anzeigen

Der HAT Kommando listet die Einträge der Handler-Adressen-Tabelle (\$031A...\$033A) auf. Zuerst wird die Adresse gezeigt, dann die 3 Bytes des Eintrags, gefolgt vom Gerätebuchstaben und der Adresse der Handler-Tabelle (bzw. -- 0000 falls der Eintrag leer ist).

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

### 3.2.14 Speicherbelegungsübersicht anzeigen

Das Kommando `M` zeigt eine Kartierung des Speichers nach benutzten und unbenutzten Speicherseiten (Pages) an. Enthält eine Page nur 0 Werte, wird ein Punkt „.“ angezeigt. Enthält sie mindestens ein Byte mit einem Wert ungleich 0, wird ein Stern „\*“ angezeigt. Dies ist zum Beispiel nützlich, wenn man eine leere Page für den Boot-Loader sucht (siehe Abschnitt 2.3.1).

### 3.2.15 Druckerausgabe aktivieren

Mit dem Kommando `PR<wert` wird die Druckerausgabe gesteuert. Werte von 1 bis 8 aktivieren die Ausgabe auf `P1:` (Standard Drucker) bis `P8:`. Der Wert 0 deaktiviert die Druckerausgabe. Mit `PR` wird der aktuelle Zustand der Druckerausgabe am Bildschirm ausgegeben. Bei aktivierter Druckerausgabe wird alles, was im Debugger Bildschirm erscheint, parallel auf dem angegebenen Drucker ausgegeben. Dies ist eine praktische Funktion, um Debugger Sitzungen zu protokollieren.

Gerade bei längeren Debugger Sitzungen kann es sehr hilfreich sein, in diese Protokolle auch Kommentare einzufügen. Auch das kann man recht einfach mit dem Kommando `;` des Debuggers erledigen. Alles, was nach dem Strichpunkt folgt, wird im Debugger Fenster (und damit bei aktivierter Druckerausgabe auch am Drucker) ausgegeben. So spart man sich den regelmäßigen Griff zum Bleistift und die Protokolle sind auch Monate nach der Debugger Sitzung noch verständlich.

### 3.2.16 AtariSIO fernsteuern

Mit dem Kommando `a` können Steuerbefehle 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 512k RAM-Erweiterung

Die eingebaute RAM-Erweiterung des TURBO FREEZERS ist kompatibel zu den verbreiteten Rambo/AtariMagazin Erweiterungen. Bit 4 von PORTB aktiviert den Zugriff auf die RAM-Erweiterung (0 = RAM-Erweiterung bei \$4000...\$7FFF einblenden, 1 = internen Atari Speicher verwenden). Die Bits 2,3,5,6 und 7 wählen eine der 32 16k Bänke aus. Ein separater ANTIC Zugriff, wie beim Atari 130 XE und einigen anderen RAM-Erweiterungen vorhanden, wird nicht unterstützt. Zum einen liegt das dazu benötigte HALT Signal nicht am PBI an. Zum anderen gibt es nur ganz wenige Demos, welche diesen Modus überhaupt verwenden und die meisten davon sind mittlerweile auch für RAM-Erweiterungen ohne separaten ANTIC Zugriff angepasst worden.

Beim Einsatz von Programmen, die eine RAM-Erweiterung mit separatem ANTIC Zugriff erfordern, ist folgende Einschränkung zu beachten. Befindet sich im Atari eine interne RAM-Erweiterung, die den separaten ANTIC Zugriff unterstützt, kann es bei aktivierter 512k RAM-Erweiterung am Freezer zu Fehlfunktionen kommen. Ist nur der ANTIC Zugriff auf die RAM-Erweiterung aktiviert (über Bit 5 von PORTB), nicht aber der CPU (bzw. kombinierter ANTIC/CPU) Zugriff (über Bit 4), so greift der ANTIC auf die interne RAM-Erweiterung zu statt auf die RAM-Erweiterung am Freezer. Die beste Lösung ist diese Situation erst gar nicht auftreten zu lassen und eine der beiden RAM-Erweiterungen zu deaktivieren.



## 5 Oldrunner – Atari OS Rev. B

Der Atari 400/800 ist der einzige 8-Bit Homecomputer, der ein sauber gegliedertes und durchdachtes Betriebssystem (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 existieren doch Programme (Jahrgang 1980-1983), die auf der Atari XL/XE Serie nicht funktionieren. Die Schuld liegt bei den Programmierern der betroffenen Programme, die die offiziellen Programmierrichtlinien nicht eingehalten hatten.

Um die unverträglichen Programme trotzdem auch auf den Atari XL/XE laufen lassen zu können, wurden sogenannte „Translator“ Disketten herausgebracht. Diese schalten das OS-ROM ab und kopieren die alte OS-Version „Rev. B“ in das unter dem OS-ROM befindliche RAM. Diese Softwarelösung funktioniert zwar in den meisten Fällen, ist aber wegen des dauernden Ladens unbequem und zeitraubend. Ideal ist nur eine Hardwarelösung, bei der das alte OS auf einem ROM abgelegt ist und damit jederzeit unveränderlich zur Verfügung steht. Nur diese Lösung funktioniert mit allen unverträglichen Programmen. Leider erforderten solche sogenannten „Oldrunner“ normalerweise Eingriffe in den Atari.

Mit dem TURBO FREEZER ist es nun möglich, einen Oldrunner ohne Eingriffe in den Atari zu realisieren. Die Memory-Management-Logik macht dies möglich. Mit dem Schalter am TURBO FREEZER, der mit „**OldOS**“ beschriftet ist, kann der Oldrunner ein- und ausgeschaltet werden. Eingeschaltet ist er, wenn der Schalter in der rechten Position ON steht. Es ist ratsam, diesen Schalter nicht bei eingeschaltetem Atari zu betätigen. Ansonsten passt der Inhalt des RAMs nicht zur neu ausgewählten OS-Version und das System stürzt ab.

Aufgrund einiger Umstände ist es ratsam, nur dann den Oldrunner zu aktivieren, wenn das Programm anders nicht läuft. Beim aktivem Oldrunner gibt es kein eingebautes BASIC, keine RAM-Erweiterung und keinen Warmstart. Die [RESET] Taste beim Atari XL/XE bewirkt bei aktivem Oldrunner immer (!) einen Kaltstart und der Inhalt des RAMs geht verloren. Die dem [RESET] beim XL/XE entsprechende Taste beim Atari 400/800 hieß [SYSTEM RESET] und bewirkte einen nicht-maskierbaren Interrupt (NMI). Sie kann zwar ohne großen Aufwand nachgerüstet werden (siehe Abschnitt 1.5.2), das erfordert aber einen Eingriff in den Atari und ist damit Experten vorbehalten. Abgesehen davon ist diese Taste ohnehin nicht unbedingt notwendig, 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 Benutzer ärgert.



## 6 Cartridge-Emulation

Mit der Cartridge-Emulation bietet der TURBO FREEZER eine weitere, sehr mächtige Funktion. Diese ermöglicht, Standard 8k und 16k Module sowie Bankswitching Module nach den AtariMax/MegaMax und OSS Standards zu emulieren. Zusätzlich kann auch das neue SpartaDOS X (Ultimate1MB Bankswitching, bis zu 512k) emuliert werden. Auch die Kombination von SpartaDOS X und z. B. einem OSS Modul ist dabei erlaubt.

Für die Cartridge-Emulation steht der gesamte freie Speicher im Flash-ROM (960k) sowie im Freezer-RAM (384k) zur Verfügung. Das heißt, mit dem TURBO FREEZER hat man direkten Zugriff auf bis zu 168 verschiedene Module. Die Cartridge-Emulation ist nicht nur für alle diejenigen interessant, die häufig mit verschiedenen Steckmodulen hantieren und das ewige Wechseln des Modules (was nebenbei bemerkt auch den Modulschacht im ATARI stark beansprucht) satt haben, 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 Moduldaten in Sekundenschnelle erledigt und testbar. Das Freezer-RAM ist batteriegepuffert, d.h. der Inhalt geht nach dem Ausschalten des Ataris nicht verloren. Daher kann das Freezer-RAM genauso wie das Flash-ROM zum Speichern permanenter Moduldaten verwendet werden.

### 6.1 Grundlagen

Um das volle Potential der Cartridge-Emulation ausschöpfen zu können muss man folgende Details über die interne Funktionsweise kennen.

#### 6.1.1 Banknummern

Das Flash-ROM und das Freezer-RAM sind intern in 8k große „Bänke“ unterteilt. Da 1MB Flash-ROM zur Verfügung stehen, gibt es insgesamt 128 8k Bänke mit den Banknummern 0...127. Die obersten 64k 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 Banknummern 0 bis 119 zur Verfügung.

Beim 512k Freezer-RAM sind die obersten 128k für die Freezer-Software reserviert, es stehen hier 384k, also die Bänke 0...47 zur Verfügung. Snapshots belegen die Bänke 48...55 im Freezer-RAM. Verzichtet man auf Snapshots können auch diese Bänke für die Cartridge-Emulation verwendet werden.

16k Module belegen 2 aufeinanderfolgende 8k Bänke. Die Daten müssen dabei an einer 16k Grenze beginnen, also an einer geraden Banknummer. Legt man z. B. ein 16k Modul in den Bänken 2 und 3 ab, so wird die Bank 2 bei \$8000...\$9FFF eingeblendet und die Bank 3 bei \$A000...\$BFFF.

512k Module, SpartaDOS X und Module mit TURBO FREEZER 2005 Bankswitching (8k old), müssen an einer 512k Grenze beginnen. Das heißt, sie können im Flash-ROM nur ab Bank 0 oder 64 beginnen, oder im Freezer-RAM ab Bank 0.

Die SpartaDOS X Emulation verwendet das gleiche Bankswitching Schema wie die „Ultimate 1MB“ Erweiterung. Die Images dafür können bis zu maximal 512k groß sein. Aktuell ist das SpartaDOS X 4.45 Image für die Ultimate 1MB Erweiterung 256k groß und kann deshalb problemlos im Flash-ROM ab Bank 0 oder 64 oder im Freezer-RAM ab Bank 0 abgelegt werden.

### 6.1.2 Modultypen

Der Modultyp legt fest, an welcher Adresse im Atari die Daten aus der Cartridge-Emulation eingeblendet werden sollen. Des weiteren werden je nach Modultyp zusätzliche Bankswitchingregister (z. B. für OSS Module) im Bereich \$D500...\$D5FF eingeblendet.

Folgende Modultypen werden unterstützt:

- 8k            8k Standard Modul, \$A000...\$BFFF.  
Über das Bankswitchingregister der Cartridge-Emulation kann die Software auf den gesamten Speicher, d.h. Freezer-ROM und Freezer-RAM zugreifen.
- 8k+RAM     8k Modul, \$A000...\$BFFF  
mit optionaler 8k RAM-Bank bei \$9000...\$BFFF. Beide Bereiche können über getrennte Bankregister unabhängig voneinander ausgewählt werden.
- 8k old      512k Bankswitching Modul,\$A000...\$BFFF.  
TURBO FREEZER 2005 kompatibles Bankswitching.  
Als Startbank muss 0 oder 64 ausgewählt werden.
- 8k AtariMax  
1MB (8MBit) Bankswitching Modul, \$A000...\$BFFF.  
Bankswitching nach dem AtariMax/MegaMax Standard.  
  
**Achtung:** In der Cartridge-Emulation stehen 64k weniger als im AtariMax 8MBit Modul zur Verfügung. Daher laufen Module, die den gesamten 1MB Speicher belegen nicht, z. B. Space Harrier. Beim Erstellen eigener Images mit der MaxFlash Software sollte man darauf achten, dass man maximal 960k belegt. Zudem muss als Start-Bank die Bank 0 ausgewählt werden.
- 16k         16k Standard Modul, \$8000...\$BFFF.  
Bankumschaltung ebenso wie im 8k Modus möglich.
- OSS         16k OSS Bankswitching Modul, \$A000...\$BFFF.  
Zum Beispiel MAC/65 oder ACTION!.

## 6.2 Aktivieren der Cartridge-Emulation

Um die Cartridge-Emulation zu aktivieren, muss der „**CartEmu**“ Schalter bei ausgeschaltetem Atari nach rechts (ON) geschoben werden. Nach dem Einschalten des Ataris erscheint

ein kleines Menü, mit dem sich die Cartridge-Emulation konfigurieren lässt. Zudem gelangt man jederzeit vom TURBO FREEZER Menü mit der Taste [K] oder [Shift]+[K] in das Menü der Cartridge-Emulation. Dies funktioniert nur, wenn die Cartridge-Emulation auch aktiviert ist, d.h. wenn mindestens einer der Schalter „**CartEmu**“ oder „**FlashWrite**“ in der rechts (ON) steht. Stehen beide Schalter links (OFF), ist die Cartridge-Emulation komplett deaktiviert.

## 6.3 Menü der Cartridge-Emulation

In der Bildschirmmitte wird die aktuelle Konfiguration angezeigt, am unteren Rand wird eine Online-Hilfe mit den verfügbaren Optionen eingeblendet.

MODE	Wählt den emulierten Modultyp. OFF deaktiviert die Cartridge-Emulation. PicoDos startet anstatt eines Moduls das integrierte MyPicoDos.
SRC	Wählt Flash-ROM oder Freezer-RAM als Quelle für die Moduldaten.
BANK	Wählt die Flash-ROM oder Freezer-ROM Startbank.
SDX	Wählt die SpartaDOS X Emulation. Hier kann zwischen OFF, ROM Bank 0, ROM Bank 64 und RAM Bank 0 gewählt werden.
BOOT	Wählt ob das Modul durch einen Kaltstart (COLD) oder Warmstart (WARM) gestartet werden soll. Diese Option sollte für gewöhnlich auf COLD gesetzt sein, da ansonsten die Variablen des Betriebssystems und das Modul nicht korrekt initialisiert werden. Die Option WARM ist für Entwickler gedacht und macht nur dann Sinn, wenn die Cartridge-Emulation vom Freezer-Hauptmenü aus mit [Shift]+[K] aktiviert wurde. Damit kann die Cartridge-Emulation umkonfiguriert werden ohne dass das interne RAM des Ataris gelöscht wird. Dies kann beim Entwickeln hilfreich sein.

Einige häufig benötigte Konfigurationen können mit einem einzigen Tastendruck ausgewählt werden:

[D]	Wählt die Standardkonfiguration: Mode 8k, Bank 0, Source ROM, SDX OFF und Boot Cold.
[P]	Wählt den Mode PicoDos.
[O]	Deaktiviert die Cartridge-Emulation: Mode OFF, SDX OFF

Ein Druck auf die [RETURN] Taste aktiviert die ausgewählte Konfiguration. Mit [ESC] wird die Cartridge-Emulation deaktiviert und der Atari neu gestartet.

## 6.4 Kommandoübersicht

Im Debugger (siehe Kapitel 3) gibt es folgende Kommandos, die alle mit K beginnen, um die Cartridge-Emulation zu kontrollieren. Diese Kommandos bewirken das gleiche wie das händische Ändern der Konfigurationsregister ab \$D500, aber angeblich soll es Leute geben, die Mnemonics der Eingabe von hexadezimalen Werten vorziehen.

K	Cartridge-Emulation Konfiguration anzeigen
K<0	Cartridge-Emulation komplett deaktivieren: Setzt Modultyp und SpartaDOS X Emulation auf OFF, alle Bänke auf 0 und Quelle auf ROM
KM<typ	Modultyp der Cartridge-Emulation setzen. Folgende Werte sind möglich: 0 ausgeschaltet 8 8k Modul 8R 8k Modul mit optionaler 8k RAM-Bank bei \$8000 80 8k Modul mit TURBO FREEZER 2005 Bankswitching A 8k AtariMax/MegaMax Modul 16 16k Modul 0 16k OSS Modul
KB<bank	8k Hauptbank wählen, (\$00...\$7F).
KBE<modus	8k Hauptbank ein/ausschalten: 0 aus 1 ein
KR<bank	Optionale 8k RAM-Bank wählen (\$00...\$3D)
KRE<modus	Optionale 8k RAM-Bank ein/ausschalten: 0 aus 1 ein
KX<modus	SpartaDOS X Emulation wählen: 0 aus 1 ein und im Flash-ROM ab Bank 0 2 ein und im Flash-ROM ab Bank 64 3 ein und im Freezer-RAM ab Bank 0
KXB<bank	Bank der SpartaDOS X Emulation setzen (\$00...\$3F). Nur möglich, wenn die Emulation über KX aktiviert wurde

**KXM<modus**

SpartaDOS X Modus setzen. Nur möglich, wenn die Emulation über KX aktiviert wurde

- 0 SpartaDOS X aus, zusätzliches Modul aus
- 1 SpartaDOS X ein, zusätzliches Modul aus
- C SpartaDOS X aus, zusätzliches Modul ein

**KO<bank** OSS-Bank setzen. Nur möglich, wenn die Emulation über KO aktiviert wurde

- 0 OSS-Modul aus
- 1...3  
OSS-Bank 1...3 wählen

**KW<modus** Schreibzugriff auf die Cartridge-Emulation ein/ausschalten:

- 0 aus
- 1 ein

**KS<source**

Quelle für die Cartridge-Emulation setzen:

- 0 Flash-ROM
- 1 Freezer-RAM

**KN<modus** Menü der Cartridge-Emulation ein/ausschalten:

- 0 Cartridge-Emulation normal, Menü deaktiviert
- 1 Cartridge-Emulation aus, Menü aktiviert

## 6.5 Weiterführende Informationen

### 6.5.1 Verwendung „richtiger“ Module

Möchte man ein „richtiges“ Modul im Modulschacht des Atari 600 XL / 800 XL oder im Modulsteckplatz der Atari XE Adapter-Platine des TURBO FREEZERS verwenden, so sollte die Cartridge-Emulation komplett deaktiviert werden. Hierzu müssen also sowohl der „**CartEmu**“ als auch der „**FlashWrite**“ Schalter nach links (OFF) geschoben werden. Ansonsten kann es zu Fehlfunktionen kommen, insbesondere dann, wenn das richtige Module ein Bankswitching Modul ist.

### 6.5.2 TRIG3 – \$D013

Die Funktionsweise der Cartridge-Emulation wurde beim TURBO FREEZER 2011 im Vergleich zum TURBO FREEZER 2005 in einem kleinen, aber sehr wichtigen Detail verbessert. Bei aktivierter Cartridge-Emulation wird beim Lesen des TRIG3 (\$D013) Registers korrekt gemeldet, ob ein Modul vorhanden ist. Damit verhält sich die neue Cartridge-Emulation 100% identisch zu einem „richtigen“ Modul und das lästige Patchen von Modul-Images, das beim TURBO FREEZER 2005 notwendig war, entfällt.



# 7 Programmieren des Flash-ROMs

## 7.1 Grundlagen

Die Software des TURBO FREEZERS 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 kann mit Moduldaten beschrieben werden, ohne dass ein spezielles Programmiergerät notwendig ist. Das Flash-ROM am TURBO FREEZER kann übrigens bis zu 1.000.000 Mal neu programmiert werden, was für die meisten Experimente wohl ausreichend sein sollte.

Das Flash-ROM und das Freezer-RAM im TURBO FREEZER ist normalerweise gegen ein Überschreiben geschützt. Um die Schreibfunktion auf beides freizuschalten, muss der Schalter „**FlashWrite**“ nach rechts (ON) geschoben werden. Steht der Schalter in der linken Position (OFF), sind das Flash-ROM und das Freezer-RAM schreibgeschützt. Es ist sehr unwahrscheinlich, dass irgendeine Software das Flash-ROM im Freezer versehentlich überschreibt. Um das Flash-ROM zu programmieren muss eine spezielle Schreibsequenz zum Flash-ROM geschickt werden, ansonsten werden Schreibzugriffe einfach ignoriert. Deshalb ist es recht ungefährlich den „**FlashWrite**“ Schalter permanent in der rechten Position (ON) zu lassen.

Beim Einsatz von „richtigen“ Bankswitching Modulen im Modulschacht des Ataris sollte aber sowohl der Schalter „**CartEmu**“ als auch der Schalter „**FlashWrite**“ auf OFF geschoben werden. Hierdurch wird ein Konflikt zwischen der Cartridge-Emulation und dem Bankswitching Modul vermieden.

**Achtung:** Es ist wichtig, dass beim Flashen der Schalter „**OldOS**“ unbedingt auf OFF steht um sicher zu gehen, dass das Flash-ROM exklusiv dem Flash-Programm zur Verfügung steht. Wenn gleichzeitig der Oldrunner aktiv ist, kommt es zu Konflikten beim Zugriff – z. B. wenn in Interrupt auftritt – und der Atari kann abstürzen.

## 7.2 Bänke und Blöcke im Flash-ROM

Um mit dem Flash-ROM perfekt umgehen zu können, ist es hilfreich, ein paar Details über seinen internen Aufbau zu kennen. Das eingesetzte Flash-ROM ist intern in Blöcke zu je 64k (d.h. acht 8k-Bänke) aufgeteilt. Diese Blöcke können unabhängig voneinander neu programmiert werden. Das bedeutet, dass Änderungen z. B. an den Daten für die Cartridge-Emulation nicht die Freezer-Software beeinflussen 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. Freie, noch nicht programmierte

Teile eines 64k Blocks können aber auch später beschrieben werden. So kann man z. B. zuerst ein 8k Modul in Bank 0 schreiben und später ein weiteres 8k Modul in Bank 1 des gleichen Blocks. Es empfiehlt sich aber, vorher mehrere z. B. 8k ROM Module zu eine großen Datei zusammenfassen und diese dann in einem Rutsch zu programmieren. Für das Freezer-RAM gilt die Einschränkung auf 64k Blöcke natürlich nicht. Dort kann jede 8k Bank separat programmiert werden.

Liegt die Startbank nicht an einer 64k Blockgrenze, so fragt das Flash-Programm nach, ob der gesamte Block zuvor gelöscht werden soll oder nicht. Löscht man den 64k Block nicht, kann man noch unbenutzte (d.h. zuvor gelöschte) Teile eines 64k Blocks mit Daten füllen. Wenn man versehentlich einen bereits benutzten Teil überschreiben will, so kommt es zu einer Fehlermeldung und man muss den gesamten 64k Block neu programmieren.

### 7.3 Flash-Programm starten

Um das Flash-ROM bzw. das Freezer-RAM zu programmieren, muss zuerst der Schalter „**FlashWrite**“ nach rechts auf **ON** geschoben werden. Dann muss das Programm **FLASH.COM** von der Systemdiskette gestartet werden. Steht der Schalter auf **OFF**, kann das Programm das Flash-ROM nicht finden und es wird eine Fehlermeldung ausgegeben. In diesem Fall kann man den Schalter auch nachträglich nach rechts schieben und dann die Rückfrage **restart program?** mit **[Y]** antworten. Nach dem Start gibt das Flash-Programm folgende Informationen aus:

- Typ des Flash-ROMs
- Versionsnummer und Datum der Freezer-Software im Flash-ROM

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** angezeigt werden.

### 7.4 Flash-Programm verwenden

Im Flash-Programm stehen folgende Funktionen zur Auswahl:

#### Program **CartEmu flash**

Programmieren des Flash-ROMs Bereiches für die Cartridge-Emulation (Bänke 0...119). Der Bereich für die Freezer-Software (Bänke 120...127) bleibt hierbei geschützt und kann nicht versehentlich überschrieben werden. Arbeitet man nur mit der Cartridge-Emulation, so sollte immer ausschließlich diese Funktion verwenden.

#### Program **CartEmu+Freezer flash**

Programmieren des gesamten Flash-ROMs (Bänke 0...127). Damit lässt sich auch die Freezer-Software (Bänke 120...127) aktualisieren.

#### Write **flash to file**

Sichern des Flash-ROM Inhaltes in eine Datei.

**Program RAM**

Programmieren des Freezer-RAMs (Bänke 0...47).

**Write RAM to file**

Sichern des Freezer-RAM Inhaltes in eine Datei.

**Erase flash**

Löschen des gesamten Flash-ROMs.

**Achtung:** Damit wird das gesamte Flash-ROM, inklusive der Freezer-Software gelöscht. Damit der TURBO FREEZER danach wieder wie gewohnt verwendet werden kann, muss unbedingt die Freezer-Software erneut ab Bank 120 programmiert werden. Ansonsten stürzt der Atari beim Drücken des Knopfes „Freeze“ oder bei aktivierter Cartridge-Emulation einfach ab.

**Start CartEmu**

Springt in das Menü der Cartridge-Emulation. So kann man ein gerade programmiertes Module direkt ausprobieren.

## 7.5 Flash-ROM und Freezer-RAM programmieren

Beim Programmieren von Flash-ROM bzw. Freezer-RAM geht man im wie folgt vor. Als erstes erfragt das Flash-Programm vom Benutzer die Nummer der ersten zu programmierende Bank. Zulässige Werte sind 0...127 beim Flash-ROM und 0...47 beim Freezer-RAM. Dann fragt das Flash-Programm nach der Anzahl der zu programmierenden 8k Bänke. Bestätigt man den Standardwert 0, so wird bis automatisch zum Ende der Datei programmiert.

Danach muss man den Namen der Datei mit den zu programmierenden Daten eingeben. Die Datei darf nur die zu programmierenden Daten enthalten, d.h. sie darf keinen COM-Header oder ähnliches besitzen und die Dateigröße muss ein Vielfaches von 8k. Es werden dann die einzelnen 8k Bänke des Flash-ROMs (oder des Freezer-RAMs) der Reihe nach programmiert. Beim Programmieren des Flash-ROMs wird am Beginn jeder 64k Block-Grenze der Block vorher automatisch gelöscht. Falls vor dem Ende der Datei das Flash-ROM bzw. Freezer-RAM-Bereiches erreicht wird, wird die Programmierung an dieser Stelle beendet.

Während der Tastatureingaben und während des Programmierens kann die aktuelle Funktion jederzeit mit der [BREAK] Taste abgebrochen werden.

## 7.6 Freezer-Software aktualisieren

Hat man versehentlich die Freezer-Software im Flash-ROM überschrieben oder gelöscht, oder möchte man die Software durch eine neuere Version ersetzen, so muss die Datei `FREEZER.ROM` von der Systemdiskette in die Bänke 120...127 des Flash-ROMs geschrieben werden. Hierzu sind folgende Schritte erforderlich:

1. Den „**FlashWrite**“ Schalter nach rechts (**ON**) schieben.
2. Das Flash-Programm `FLASH.COM` von der Systemdiskette laden.
3. Den 2. Menüpunkt (`Program CartEmu+Freezer flash`) auswählen.
4. Als Startbank 120 eingeben.
5. Den Standardwert 0 bei der Bank-Anzahl mit `[RETURN]` bestätigen.
6. Als Dateinamen `FREEZER.ROM` eingeben.

# 8 Für Experten: Technische Details

## 8.1 Hardware

Herzstück des Freezers ist der CPLD Xilinx XC95144XL. Er enthält die gesamte Logik des Freezers und steuert das Freezer-RAM und das Flash-ROM an. Insgesamt sind 1MB Flash-ROM und 1MB batteriegepuffertes Freezer-RAM vorhanden, die sich einzelnen Funktionsblöcke Freezer, Cartridge-Emulation und 512k RAM-Erweiterung untereinander teilen.

Wenn das Freezer-RAM oder das Flash-ROM des Freezers in den Atari eingeblendet werden soll, muss das interne RAM und ROM im Atari deaktiviert werden. Das geschieht durch einen kleinen Trick. Der Freezer zieht hierbei den eigentlich als Ausgang gedachten Refresh-Pin auf „Low“. Die veranlasst die MMU im Atari zu glauben, dass der ANTIC einen Speicher-Refresh durchführt, woraufhin sie den eingebauten Speicher deaktiviert. Laut Bernhard Engl, der heute hauptberuflicher Chipdesigner ist, ist es bei der damaligen NMOS-Depletion-Load-Technologie ungefährlich, einen „High“-Pegel extern auf „Low“ zu ziehen, weil das bei NMOS ohnehin intern so geschieht. Wäre der ANTIC in CMOS-Technologie gefertigt gewesen, dann wäre das möglicherweise gefährlich gewesen, hätte den ANTIC beschädigen können – und es hätte niemals einen TURBO FREEZER gegeben.

## 8.2 Speicheraufteilung

### 8.2.1 Freezer-RAM

512k des Freezer-RAMs sind fest der 512k RAM-Erweiterung zugeordnet. Die restlichen 512k RAM teilen sich der Freezer und die Cartridge-Emulation. Der Freezer kann auf die oberen 128k (Bänke 48...63) davon zugreifen. Die obersten 16k (Bänke 62 und 63) sind rein für den Freezer (z. B. für das Hardware Register Shadowing) reserviert. Das Bankswitching im Freezer adressiert das Freezer-RAM aber nicht in 8k Bänken, wie die Cartridge-Emulation, sondern in 4k großen Bänken. Die Cartridge-Emulation hat Zugriff auf das gesamte RAM, mit Ausnahme der letzten 16k. Insgesamt also maximal 496k (Bänke 0...61).

### 8.2.2 Flash-ROM

Die Cartridge-Emulation hat Zugriff auf das gesamte 1MB Flash-ROM, der Freezer auf die obersten 512k davon. Die obersten 64k sind für die Freezer-Software, das Oldrunner-OS und das Menü der Cartridge-Emulation reserviert. Die obersten 64k (Bänke 120...127) sind aktuell wie folgt belegt.

- Bank 120: Frei
- Bank 121: Frei
- Bank 122: Frei
- Bank 123: Freezer-Software Bank 3
- Bank 124: Freezer-Software Bank 1 (Startbank)
- Bank 125: Freezer-Software Bank 2
- Bank 126: Menü der Cartridge-Emulation
- Bank 127: Oldrunner-OS

Die Bänke 127 (Oldrunner-OS), 126 (Cartridge-Emulation) und 124 (Freezer-Software) sind in der CPLD Logik „fest verdrahtet“. Die Belegung der restlichen Bänke 120...123 und 125 können sich in späteren Versionen noch ändern.

## 8.3 Freezer

Solange man den Knopf „Freeze“ nicht drückt, ist der Freezer vom Atari aus absolut unsichtbar. Das heißt, keine Software kann feststellen ob, ein Freezer am Atari angesteckt ist. Erst wenn der Knopf gedrückt wird, tritt der Freezer sichtbar in Aktion.

### 8.3.1 Shadowing der Hardwareregister

Um den Inhalt der nicht auslesbaren Hardwareregister von ANTIC, GTIA und POKEY auslesen zu können, verwendet der Freezer folgenden „Trick“. Bei Schreibzugriffen in den Bereich \$D000...\$D7FF wird auch das Freezer-RAM aktiviert. Die Freezer-Software kann so später die Daten der Custom Chips von dort problemlos auslesen. Das Fortschreiben geschieht aber nur, solange der Knopf „Freeze“ noch nicht gedrückt wurde. Ansonsten würde die Freezer-Software selbst die gespeicherten Daten überschreiben.

Im Atari sind die Adressen der Custom Chips nicht vollständig ausdekodiert. Der ANTIC ist z. B. über die Adressen \$D40x, \$D41x, ... erreichbar. Schreibt man z. B. zuerst \$22 nach \$D400 und danach \$00 nach \$D410, so landen beide Werte im ANTIC D<sub>MACTL</sub> Register. D<sub>MACTL</sub> ist steht nach dem 2. Schreibzugriff auf \$00.

Einige Programme verwenden diese Tricks. Damit der Freezer nach dem Auftauen den Zustand wieder exakt herstellen kann, muss die Freezer-Logik darauf Rücksicht nehmen. Beim Shadowing der Hardwareregister werden deshalb die nicht ausdekodierten Adressleitungen auf „Low“ gezogen. Bei ANTIC (\$D4xx) und POKEY (\$D2xx) sind das A4–A7, bei GTIA (\$D0xx) sind es A5–A7.

Befindet sich im Atari eine Stereo-POKEY-Erweiterung, so ist der erste POKEY über \$D20x, \$D22x, ... erreichbar, der zweite POKEY über \$D21x, \$D23x, ... . Schiebt man den Schalter „Stereo“ am Freezer nach rechts (0N), ändert sich die Shadowing-Logik. Bei Zugriffen auf \$D2xx werden dann die Adressleitungen A5–A7 auf „Low“ gezogen und die Daten beider POKEYs werden korrekt gespeichert.

### 8.3.2 Einblenden des Freezer-Speichers

Wenn der Freezer aktiviert ist, wird im Bereich \$0000...\$3FFF das Freezer-RAM bzw. des Flash-ROM eingeblendet. Mittels Bankswitching kann zwischen 32 jeweils 4k großen RAM-Bänken (insgesamt 128k) und 64 jeweils 8k großen ROM Bänken (insgesamt 512k) umgeschaltet werden. Die Speicherbelegung im Atari sieht wie folgt aus.

- \$2000...\$3FFF: Umschaltbare 8k ROM-Bank (Bank 64...127)
- \$1000...\$1FFF: Umschaltbare 4k RAM-Bank (Bank 0...31)
- \$0000...\$0FFF: Feste 4k RAM-Bank (Bank 31)

Die 64 jeweils 8k großen Freezer-ROM-Bänke befinden sich am Ende des Flash-ROMs und entsprechen damit den Bänken 64...127 der Cartridge-Emulation. Die 32 jeweils 4k großen RAM-Bänke befinden sich am Ende des Freezer-RAMs und entsprechen den Bänken 48...63 der Cartridge-Emulation. Die letzte 4k Bank (Bank 31) wird für das Shadowing der Hardware Register verwendet und wird bei aktiviertem Freezer ab Adresse \$0000 im Atari eingeblendet.

### 8.3.3 Aktivierung des Freezers

Intern kennt die Freezer-Logik 5 verschiedene Zustände.

- Freezer deaktiviert
- Freezer halb aktiviert
- Freezer Aktivierung ausgelöst
- Freezer aktiviert
- Freezer temporär deaktiviert

Nach dem Einschalten des Ataris befindet sich der Freezer im Zustand „deaktiviert“. In diesem Zustand und im Zustand „halb aktiviert“ werden Schreibzugriffe in den Custom Chips Bereich im Shadow-RAM gespeichert.

Drückt man den Knopf „Freeze“, so passiert erst einmal noch nichts. Erst, wenn der erste Zugriff auf den Bereich \$FFF8...\$FFFF (hier liegen unter anderem die IRQ und NMI-Vektoren) erfolgt und der Knopf „Freeze“ noch gedrückt ist, regiert der Freezer. Er merkt sich den Zustand der A2 Leitung (das wird später benötigt um zwischen IRQ und NMI unterscheiden zu können) und geht in den Zustand „halb aktiviert“ über. Im nächsten Taktzyklus gibt es nun folgende beiden Möglichkeiten:

- Wenn der Knopf „Freeze“ 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.
- Wenn der Knopf „Freeze“ immer noch gedrückt ist, und wieder ein Zugriff auf den Speicherbereich \$FFF8...\$FFFF erfolgt, beginnt der nachfolgend beschriebene Übergang in den Zustand „Aktivierung ausgelöst“.

Der Freezer deaktiviert den Speicher im Atari und legt den Wert \$21 an den Datenbus. Damit werden die Interrupt-Vektoren an die Adresse \$21xx verbogen, da die CPU ja gerade versucht das High-Byte des Vektors zu laden. Das Low-Byte stammt vom vorherigen Zugriff auf den \$FFF8...\$FFFF Bereich und hat den originalen Wert.

Danach geht der Freezer in den Zustand „Aktivierung ausgelöst“ über. Genauso wie im Zustand „aktiviert“ blendet der TURBO FREEZER nun seinen eigenen Speicher ein. In den Speicherbereich \$2000...\$3FFF wird die ROM-Bank 124 mit dem Start-Code der Freezer-Software eingeblendet. Ab \$2000 liegt eine Page voller „RTI“ Befehle. Danach, also ab \$2100, liegt eine sogenannte „NOP Slide“, d.h. eine Page voller „NOP“ Befehle für die „umgebogene“ Interrupt-Routine. Darauf folgt die eigentliche Freezer-Software, beginnend mit der Initialisierungs-Routine.

In der Aktivierungs-Phase auftretende IRQs sind kein Problem, denn während der Ausführung einer Interrupt-Routine werden die IRQs von der CPU automatisch gesperrt. Tritt während der Aktivierungs-Phase ein NMI auf, so biegt der Freezer den Interrupt-Vektor auf die Adresse \$20xx um. Dort liegen aber nur „RTI“ Befehle und der NMI wird sofort wieder verlassen. Der Freezer „unterdrückt“ also alle weiteren Interrupts während der Aktivierungs-Phase.

Nach dem Durchlaufen der NOP-Slide und dem Sperren der Interrupts versetzt die Software per Schreibzugriff auf die Adresse \$D720 den Freezer schließlich in den Zustand „aktiviert“. Möchte die nun aktivierte Freezer-Software auf den Atari-Speicher im Bereich \$0000...\$3FFF zugreifen so muss zuerst das Freezer-RAM bzw. das Flash-ROM vorübergehend ausgeblendet werden. Dies geschieht durch einen Schreibzugriff auf die Adresse \$D700, wodurch der Freezer in den Zustand „temporär deaktiviert“ übergeht. Ein weiterer Schreibzugriff auf die Adresse \$D700 versetzt den Freezer wieder in den Zustand „aktiviert“ und das Freezer-RAM bzw. das Flash-ROM wird wieder eingeblendet.

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

### 8.3.4 Konfigurationsregister

Wenn der Freezer aktiviert ist, werden im Bereich \$D700...\$D7FF eine Reihe von Konfigurationsregistern eingeblendet. Bei Zugriffen auf diese Register wird der Speicher im Atari deaktiviert, um Probleme mit evtl. vorhandenen internen Erweiterungen zu vermeiden.

#### \$D700...\$D70F: Freezer State Control (w)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Schreibzugriffe auf dieses Register schalten den Freezer zwischen den Zuständen „aktiviert“ und „temporär deaktiviert“ um.

**\$D700...\$D70F: Freezer Disable (r)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lesezugriff auf dieses Register versetzt den Freezer in den Zustand „deaktiviert“.

**\$D710...\$D71F: Freezer-RAM A4/A5 (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lesezugriff auf dieses Register verändert die Steuerung der Adressleitungen A4 und A5 beim Zugriff auf das Freezer-RAM (\$0000...\$1FFF). A4 wird auf den Zustand von A2 bei der Aktivierung des Freezers gesetzt, A5 auf den aktuellen Zustand des „Stereo“-Schalters. Dadurch kann die Freezer-Software ermitteln, ob der Freezer bei einem NMI oder einen IRQ aktiviert wurde und ob der Stereo POKEY Modus aktiviert ist oder nicht. Ein Schreibzugriff auf das Register stellt den Normalzustand wieder her.

**\$D720...\$D72F: Freezer Startup Complete (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf dieses Register versetzt den Freezer vom Zustand „Aktivierung ausgelöst“ in den Zustand „aktiviert“.

**\$D740...\$D77F: Flash ROM Bank Select (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf diese Register wählt die 8k Flash-ROM-Bank aus, die bei \$2000...\$3FFF eingeblendet wird. Ein Zugriff auf \$D740 wählt die Bank 64, \$D741 die Bank 65, ... \$D77F die Bank 127.

**\$D780...\$D79F: Freezer RAM Bank Select (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf diese Register wählt die 4k Freezer-RAM-Bank aus, die bei \$1000...\$2FFF eingeblendet wird. Ein Zugriff auf \$D780 wählt die Bank 0, \$D781 die Bank 1, ... \$D79F die Bank 31.

**8.3.5 Freezer-Software und OS-ROMs**

Die Freezer-Software verwendet für die meisten Funktionen (wie z. B. Tastatureingabe) nicht die Routinen des Betriebssystems sondern enthält dafür eigenen Code. Dadurch ist die Software weitgehend unabhängig vom eingesetzten Betriebssystem und vor allem auch vom Zustand des Ataris während des Einfrierens. Um die korrekte Funktion der Freezer-Software zu gewährleisten muss das eingesetzte Betriebssystem dennoch die folgenden Funktionen zur Verfügung stellen und sich dabei so wie die originalen Betriebssystemversionen verhalten.

- Standard-Zeichensatz bei \$E000...\$E3FF

- NMI und IRQ Verarbeitung
- SIO-Funktion (\$E459), wird verwendet durch das Kommando SIO im Debugger und beim Laden oder Speichern auf Kassette

Alle bisher erhältlichen alternativen OS-Versionen wie z. B. „QMEG OS“ oder „Romdisk OS“ erfüllen diese Voraussetzungen. Probleme gibt es aber aktuell mit dem „Ultimate 1MB BIOS“. Es enthält weder eine standardmäßige Interrupt-Verarbeitung noch einen Zeichensatz ab \$E000, deshalb ist der Bildschirmaufbau komplett gestört falls man versucht im „Ultimate 1MB BIOS“ den Freezer zu aktivieren.

## 8.4 512k RAM-Erweiterung

Für die 512k RAM-Erweiterung steht ein separater 512k RAM Chip zur Verfügung. Der Freezer, die Cartridge-Emulation im Freezer-RAM und die RAM-Erweiterung sind dadurch in vollem Umfang parallel nutzbar.

Da die PIA-Signale nicht am PBI anliegen, muss im CPLD das Verhalten der PIA-Register nachgebildet werden. Die PIA im Atari läuft dabei „parallel“ weiter, damit BASIC, OS-ROM und Self Test wie gewohnt gesteuert werden können. Lesezugriffe auf die PIA werden im CPLD ignoriert, d.h. bei ihnen liest der Atari wie gewohnt die echten Register der PIA aus. Die Logik reagiert dabei nur auf Schreibzugriffe auf die Adressen \$D301 (PORTB) und \$D303 (PBCTL). Hierbei speichert der CPLD auch den Wert von Bit 2 von PBCTL, welches den Zugriff auf das PORTB Daten-Register (Bit 2 = 1) bzw. das Daten-Richtungs-Register (Bit 2 = 0) steuert.

Schreibzugriffe auf \$D301 speichert der CPLD entweder in dem internen Daten- oder dem internen Datenrichtungsregister. Wird ein PIA-Pin auf Eingang geschaltet, so ziehen Pull-Up-Widerstände im Atari die entsprechenden Pins auf „High“. Wird ein PIA-Pin auf Ausgang geschaltet, legen die Werte im Datenregister fest, ob ein „Low“ oder ein „High“ Signal ausgegeben werden soll.

Der CPLD emuliert nun genau dieses Verhalten und verhält sich damit exakt so wie jede andere RAM-Erweiterung im Atari. So schaltet beispielsweise das Oldrunner-OS PORTB auf Eingang, denn der Atari 400/800 hatte ja 4 Joystick-Eingänge – dafür aber keine PORTB RAM/ROM-Steuerung. Hierdurch wird automatisch jede RAM-Erweiterung deaktiviert denn Bit 4 von PORTB muss ja „Low“ sein, um die RAM-Erweiterung zu aktivieren. Der emulierte Pull-Up-Widerstand im CPLD zieht diesen Pin aber automatisch auf „High“, so wie es auch an der PIA der Fall wäre.

Der Rest der RAM-Erweiterungs-Logik ist relativ simpel. Steht der Schalter „**Ramdisk**“ auf ON und hat das Bit 4 des emulierten PORTB Registers den Wert 0, so wird bei Zugriffen auf \$4000...\$7FFF der interne Speicher des Ataris ausgeblendet und stattdessen eine 16k Bank des 512k RAM-Erweiterung eingeblendet. Die Bits 2,3,5,6,7 des emulierten PORTB-Register legen dabei fest, welche der insgesamt 32 jeweils 16k großen Bänke verwendet werden soll.

## 8.5 Oldrunner

Ist Schalter „**OldOS**“ auf **ON**), so ist der Oldrunner aktiv. Bei Zugriffen auf den Bereich \$E000...\$FFFF wird dann die Bank 127 eingeblendet und bei Zugriffen auf den Bereich \$C000...\$CFFF der interne Speicher ausgeblendet. Dadurch verhält sich der Atari genauso wie der originale Atari 400/800, der bei \$C000...\$CFFF über keinen Speicher verfügte. Bei Zugriffen auf das Math Pack (\$D800...\$DFFF) wird weiterhin das interne Atari OS-ROM eingeblendet, also das dort vorhandene Math Pack verwendet.

## 8.6 Cartridge-Emulation

Die Cartridge-Emulation bietet sehr umfangreiche Konfigurationsmöglichkeiten. Eine Konfiguration rein durch Schalter wäre sehr unübersichtlich geworden. Deshalb erfolgt der Großteil der Konfiguration per Software z. B. im Menü der Cartridge-Emulation oder auch per Debugger im Freezer.

### 8.6.1 Konfigurationsschalter

Eine Funktion der Schalter „**CartEmu**“ und „**FlashWrite**“ ist die Steuerung des Zugriffs auf die Konfigurationsregister. Stehen beide Schalter auf **OFF**, sind die Konfigurationsregister nicht erreichbar. Steht mindestens einer der beiden Schalter auf **ON**, so kann bei \$D590...\$D596 auf die Konfigurationsregister zugegriffen werden.

Steht während des Einschaltens des Ataris der Schalter „**CartEmu**“ in der Position **ON**, so wird Bit 2 von Register \$D595 mit 1 initialisiert. Damit wird die restliche Konfiguration der Cartridge-Emulation außer Kraft gesetzt und das Menü der Cartridge-Emulation aktiviert.

Steht der Schalter „**FlashWrite**“ in der Position **OFF**, so sind keine Schreibzugriffe auf das Flash-ROM bzw. Freezer-RAM möglich. The einzige Ausnahme ist der **8k+RAM** Modus, bei dem Schreibzugriffe auf die optionale 8k Freezer-RAM-Bank bei \$8000...\$9FFF immer möglich sind. Steht der Schalter „**FlashWrite**“ auf **ON**, sind Schreibzugriffe auf das Flash-ROM und Freezer-RAM nach vorheriger Freigabe durch Bit 0 des Registers \$D595 möglich.

### 8.6.2 Konfigurationsregister

Steht mindestens einer der beiden Schalter „**CartEmu**“ oder „**Flash Write**“ auf **ON**, so werden bei \$D590...\$D596 die Konfigurationsregister der Cartridge-Emulation eingeblendet. Diese Register sind sowohl schreib- als auch lesbar. Alle unbenutzten Bits dieser Register sollten mit 0 beschrieben werden. Beim Lesen sind die unbenutzten Bits ebenfalls auf 0 gesetzt. Nach dem Einschalten sind alle Register außer Register \$D595 auf \$00 gesetzt. Steht der Schalter „**CartEmu**“ auf **OFF**, wird es auf \$00 gesetzt, bei **ON** auf \$04. Je nach gesetztem Modus (z. B. im OSS oder AtariMax Modus) werden im Bereich \$D500...\$D5FF weitere Register eingeblendet.

**\$D590: Bank (r/w)**

7	6	5	4	3	2	1	0
-	bank						

Nummer der ausgewählten Bank für die Cartridge-Emulation. Im 16k Modus wird Bit 0 ignoriert.

**\$D591: Bank Enable (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	enable

Status der ROM-Bank des Moduls. Der Wert 0 blendet das Modul aus, 1 blendet es ein. Im 8k+RAM Modus kontrolliert dieses Bit nur den Bereich bei \$A000...\$BFFF. Der optionale RAM-Bereich bei \$8000...\$9FFF wird durch das Register \$D593 gesteuert.

**\$D592: RAM Bank (r/w)**

7	6	5	4	3	2	1	0
-	-	bank					

Nummer der ausgewählten die RAM-Bank (0...63) im 8k+RAM Modus.

**\$D593: RAM Bank Enable (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	enable

Status der RAM-Bank des Moduls. Im 8k+RAM Modus blendet der Wert 0 die RAM-Bank aus, 1 blendet sie ein.

**\$D594: Mode (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	mode		

Typ der emulierten Moduls. Folgende Modultypen werden unterstützt.

- 0: Cartridge-Emulation deaktiviert
- 1: 8k Modul
- 2: 8k+RAM Modul
- 3: 8k Modul mit TURBO FREEZER 2005 Bankswitching
- 4: 16k Modul
- 5: OSS Modul
- 6: AtariMax Modul
- 7: reserviert, nicht verwenden!

**\$D595: Misc Configuration (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	menu	source	write

- menu** Steuerung des Menüs der Cartridge-Emulation  
 0 Normale Cartridge-Emulation  
 1 Menü der Cartridge-Emulation einblenden (Bank 126)
- source** Quelle der Moduldaten  
 0 Flash-ROM  
 1 Freezer-RAM
- write** Steuerung des Schreibzugriffs  
 0 Schreibzugriffe unterbunden  
 1 Schreibzugriffe erlaubt, wenn zusätzlich der Schalter „FlashWrite“ in der Position ON steht

**\$D596: SDX Configuration (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	config	

- 00 SpartaDOS X deaktiviert  
 01 SpartaDOS X im Freezer-RAM ab Bank 0  
 10 SpartaDOS X im Flash-ROM ab Bank 0  
 11 SpartaDOS X im Flash-ROM ab Bank 64

**8.6.2.1 Zusätzliche Register im 8k Modus mit TURBO FREEZER 2005 Bankswitching****\$D540...\$D57F: Bank Select (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf diesen Bereich wählt die Bank aus. Ein Zugriff auf Adresse \$D540 wählt Bank 0, \$D541 wählt Bank 1, ..., \$D57F wählt Bank 63. Ist das Bankregister \$D590 auf eine Bank im Bereich 64...127 gesetzt, werden über \$D540...\$D57F statt der Bänke 0...63 die Bänke 64...127 angesprochen.

**\$D580: Cart Disable (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf dieses Register blendet das Modul aus.

**\$D581: Cart Enable (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf dieses Register blendet das Modul ein.

**\$D584: Write Disable (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf dieses Register deaktiviert Schreibzugriffe auf das Modul.

**\$D585: Write Enable (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf dieses Register erlaubt Schreibzugriffe auf das Modul.

**8.6.2.2 Zusätzliche Register im OSS Modus****\$D500...\$D50F: OSS Bank Select (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Diese Register realisieren die Bankauswahl bzw. Cart Disable wie bei den OSS Modulen vom Typ „M091“. Ein Lese- oder Schreibzugriff auf die jeweilige Adresse bewirkt folgendes.

- \$D500 Modul eingeblendet, \$A000...\$AFFF Bank 1, \$B000...\$BFFF Bank 0
- \$D501 Modul eingeblendet, \$A000...\$AFFF Bank 3, \$B000...\$BFFF Bank 0
- \$D508 Modul ausgeblendet
- \$D509 Modul eingeblendet, \$A000...\$AFFF Bank 2, \$B000...\$BFFF Bank 0

**8.6.2.3 Zusätzliche Register im AtariMax Modus****\$D500...\$D57F: Bank Select (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf diesen Bereich aktiviert das Modul und wählt die Bank aus. Ein Zugriff auf Adresse \$D500 wählt Bank 0, \$D501 wählt Bank 1, ..., \$D57F wählt Bank 127.

**\$D580: Cart Disable (r/w)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Ein Lese- oder Schreibzugriff auf dieses Register blendet das Modul aus.

**Achtung:** Beim original AtariMax Modul können zum Ausblenden auch die Adressen \$D581...\$D5FF verwendet werden, die Cartridge-Emulation stellt nur diese einzige Adresse zur Verfügung.

### 8.6.2.4 Zusätzliche Register bei aktiviertem SpartaDOS X

#### \$D5E0: Bank Select (r/w)

7	6	5	4	3	2	1	0
control		bank					

**control** SpartaDOS X Steuerung

00 SpartaDOS X eingeblendet, zusätzliches Modul ausgeblendet

01 SpartaDOS X eingeblendet, zusätzliches Modul ausgeblendet

10 SpartaDOS X ausgeblendet, zusätzliches Modul eingeblendet

11 SpartaDOS X ausgeblendet, zusätzliches Modul ausgeblendet

**bank** SpartaDOS X Bankauswahl (Bank 0...63)

Wenn die SpartaDOS X Emulation über das Register \$D596 aktiviert wurde, hat sie Vorrang gegenüber der restlichen Cartridge-Emulation. Die Cartridge-Emulation verhält sich dadurch so, als ob sie auf ein original SpartaDOS X Modul „aufgesteckt“ wäre.



## 9 Weiterführende Informationen und Downloads

- TURBO FREEZER  
<http://turbofreezer.horus.com>  
Aktuelle Infos, Updates etc..
- ABBUC  
<http://www.abbuc.de>  
Bezugsquelle für den TURBO FREEZER.
- Xilinx  
<http://www.xilinx.com>  
Datenblätter zum CPLD XC95144XL, Download der Entwicklungsumgebung ISE WebPACK.
- ST Microelectronics  
<http://www.st.com>  
Datenblätter zum Flash-ROM 29F040B.

# Garantie- und Verwendungsbestimmungen

Auf Material und Zusammenbau werden 6 Monate Garantie gewährt. Sollte der TURBO FREEZER defekt sein, wird er gegen einen funktionstüchtigen TURBO FREEZER umgetauscht. Sollte der TURBO FREEZER wegen Hardwareinkompatibilität nicht funktionieren, wird nach freigemachter Rücksendung der Kaufpreis zurück-erstattet. 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, Wandlung, 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 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 FREEZERS 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.

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.

Diese Anleitung ist urheberrechtlich geschützt und steht unter der Creative Commons Lizenz BY-NC-SA 3.0.

<http://creativecommons.org/licenses/by-nc-sa/3.0/de>