

# Benutzerhandbuch

PROFESSIONAL-030PLUS500-BOARD

HARMS Computertechnik

Bremen November 1992

© by Harms Computertechnik

Autoren: Oliver Pophanken, Oliver Harms

Text, Abbildungen und Programme wurden mit größter Sorgfalt erarbeitet. Der Hersteller und die Autoren können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Kein Teil dieser Anleitung darf ohne schriftliche Genehmigung des Autors in irgendeiner Form durch Fotokopie, Mikrofilm oder andere Verfahren reproduziert oder in eine für Maschinen, insbesondere Datenverarbeitungsanlagen, verwendbare Sprache übertragen werden. Auch die Rechte der Wiedergabe durch Vortrag, Funk und Fernsehen sind vorbehalten.

## **Vorwort**

Wann immer Sie Ihren Amiga ausschalten, warten Sie auf jeden Fall mindestens 30 Sekunden, ehe Sie ihn wieder einschalten. Diese Zeit ist nötig, damit die RAM-Bausteine alle Informationen verlieren.

Wenn Sie Ihren Amiga nach kürzerer Zeit wieder einschalten, kann das zur Folge haben, daß eine "Guru Message" auf Ihrem Bildschirm ausgegeben wird. Diese kann durch drücken der linken Maus-Taste zurückgesetzt werden.

Sämtliche Bemerkungen, die sich auf den Amiga 2000 beziehen sind anhand der Benutzerhandbücher nachzuvollziehen, die mit zum Lieferumfang von Commodore gehörten. Selbiges gilt auch für die Anwendung von CLI -Befehlen oder die Benutzung der Workbench.

Commodore ist ein eingetragenes Warenzeichen von Commodore Electronics Limited.

Amiga, Amiga 2000, AmigaDOS, Kickstart, Workbench und CLI sind eingetragene Warenzeichen von Commodore-Amiga, Inc.

Der Inhalt dieses Handbuches kann ohne Ankündigung geändert werden und ist nicht als eine Garantieerklärung anzusehen.

## **Inhaltsverzeichnis**

	<b>Vorwort</b>	<b>3</b>
	<b>Inhaltsverzeichnis</b>	<b>4</b>
<b>1.</b>	<b>Das Professional-030PLUS500-Board</b>	<b>5</b>
1.1	Allgemeine Beschreibung	5
1.2	Jumperstellungen	9
1.3	Cache Control register	10
1.4	RAM-Bestückung	11
1.5	Der Coprozessor	12
1.6	Lieferumfang	13
1.7	Rücksendung	13
<b>2.</b>	<b>Einbau in den Amiga 500</b>	<b>14</b>
2.1	Der erste Einsatz	17
2.2	Installation unter Kickstart 1.3	18
2.3	Installation unter Kickstart 2.0	19
<b>3.</b>	<b>Software</b>	<b>20</b>
3.1	Testprogramme	22
3.1.1	Flaot	22
3.1.2	Savage	22
3.1.3	Whetstone	23
3.1.4	AmigaBench	24
3.2	Utility-Programme	26
3.2.1	SetCPU V1.6	26
3.2.2	Memroutines	38
3.2.3	Joylib	40
3.2.4	FPU Mathtrans.Library V1.1	41
<b>4.</b>	<b>Lizenzvertrag</b>	<b>43</b>
4.1	Hardware Haftung und Garantie	45
<b>5.</b>	<b>Fehlerbehandlung</b>	<b>46</b>
5.1	Allgemeine Fehler	46
5.2	Festplattencontroller	46
<b>6</b>	<b>Technische Daten</b>	<b>47</b>

# **1. Das Professional-030PLUS500-Board**

## **1.1 Allgemeine Beschreibung**

Das PROFESSIONAL-030PLUS500-BOARD ist eine Erweiterungskarte für den Amiga 500, um die Rechenleistung des Computers zu erhöhen. Das Board ist auch in dem Amiga 2000 einsetzbar. Das PROFESSIONAL-030PLUS500-BOARD besteht aus einer vierlagigen Platine, auf der sich drei große Sockel befinden. Die quadratischen Sockel sind für die neuen Prozessoren, wobei der kleinere den Coprozessor und der große die 68030 CPU aufnimmt. Der längliche Sockel ist für ihre alte 68000er CPU gedacht. Weiterhin sind auf dem Board zwei RAM-Bänke untergebracht. Die Bestückung ist unterschiedlich, je nach dem, wieviel Speicher bestellt wurde. In dem SMD-Chips (Gate-Arrays) ist die Logik vom PROFESSIONAL-030PLUS500-BOARD untergebracht. Mit dem Oszillator wird die Taktfrequenz für den Coprozessor generiert. Dann sind da noch ein paar diskrete Bauteile, die die Analogseite des Boards abdecken (Widerstandsnetzwerke, Kondensatoren, etc.).

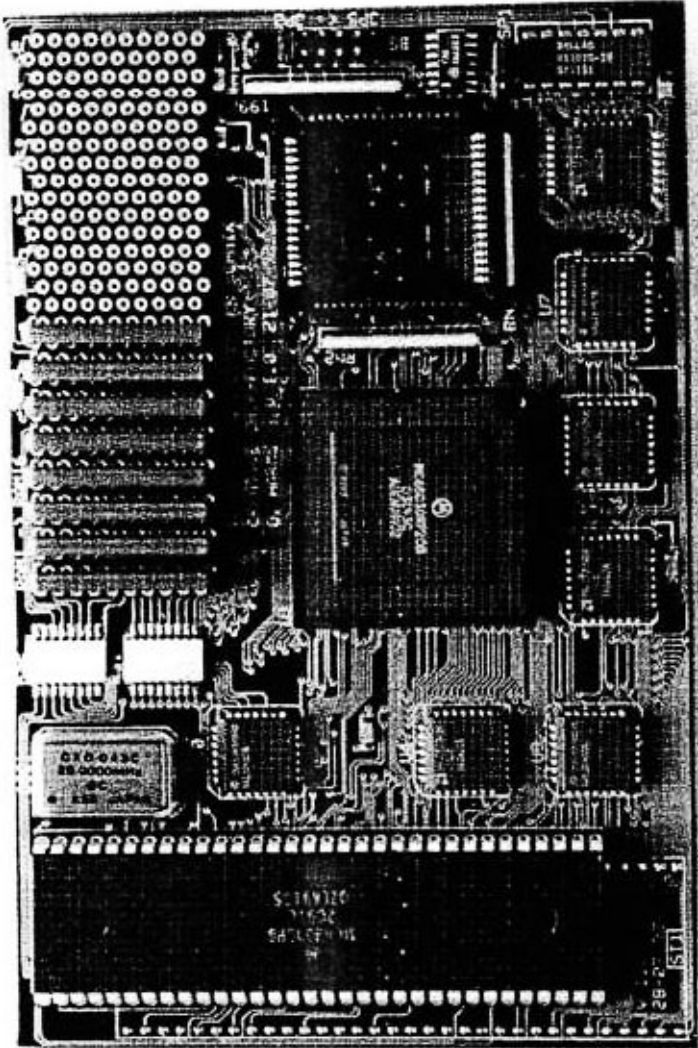
Der Amiga 2000 und Amiga 500, sowie der Amiga 500+ werden von Commodore mit dem Mikroprozessor 68000 von Motorola ausgeliefert. Dieser MC 68000 ist ein Prozessor der zu seiner Umgebung 16 Bit breite Daten- und Adressleitungen unterhält. Im Laufe der Zeit gesellten sich noch weitere Prozessoren der gleichen Serie dazu. Das sind MC 68010, MC 68020 und der MC 68030. Seit einiger Zeit ist auch der MC 68040 erhältlich. Durch die Verwendung des PROFESSIONAL-030PLUS500-BOARD wird Ihr Amiga mit dem MC68030 betrieben (umschaltbar auf MC68000). Der Vorteil eines 68030- gegenüber einem 68020-Prozessors besteht hauptsächlich darin, daß er asynchron getaktet werden kann. Das bedeutet, daß der Prozessor nicht den selben Takt besitzen muß wie beispielsweise die Customchips (Agnus, Blitter, Denise, ...) des Amiga. Eine Eigenschaft, die der FPU (68881 und 68882) auch gegeben ist. Neben dem asynchronen Takten besitzt der 68030 ein Instruktion-, Daten-Cache und einen sogenannten Burst-Mode, bei dem sich die Speicherzugriffe pro Langwort enorm verkürzen. Der 68020 hat hingegen nur ein Instruktion-Cache. Neben diesen Änderungen gesellt sich noch eine weitere, die Memory Management Unit 68851 (MMU). Sie ist bereits im 68030 integriert.

Damit man auch Software betreiben kann, die nur auf dem 68000 läuft ist das PROFESSIONAL-030PLUS500-BOARD so konzipiert worden, daß eine Hardwareumschaltung möglich ist. Das PROFESSIONAL-030PLUS500-BOARD wird standardmäßig im 68030 Modus gejumptert ausgeliefert. Die

68000er CPU muß zuerst auf das PROFESSIONAL-030PLUS500-BOARD gesteckt werden, bevor es in den 68000er Sockel des Amiga gesteckt wird. Näheres in Kapitel 2.

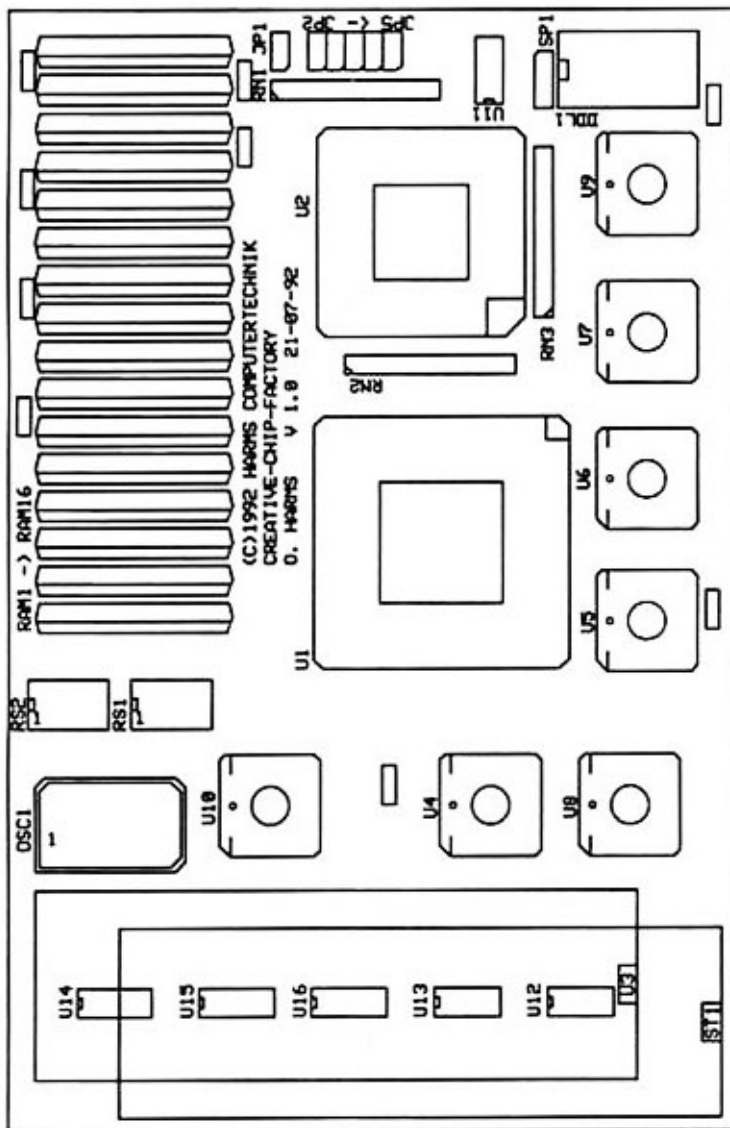
Weitere Änderungen an der Karte sind durch Jumper auf dem PROFESSIONAL-030PLUS500-BOARD einstellbar. Die Jumper sind auf Seite 9 beschrieben.

Bevor Sie zum Einbau schreiten packen Sie das PROFESSIONAL-030PLUS500-BOARD vorsichtig aus und inspizieren Sie es. Verwahren Sie die Verpackung und den Karton sorgfältig, um im Fehlerfalle das Board in der Originalverpackung zurücksenden zu können. Weist die Verpackung eine starke Beschädigung auf, verwenden Sie das Board auf keinen Fall. Informieren Sie stattdessen die Post oder Transportfirma und uns.



Das PROFESSIONAL-030PLUS500-BOARD

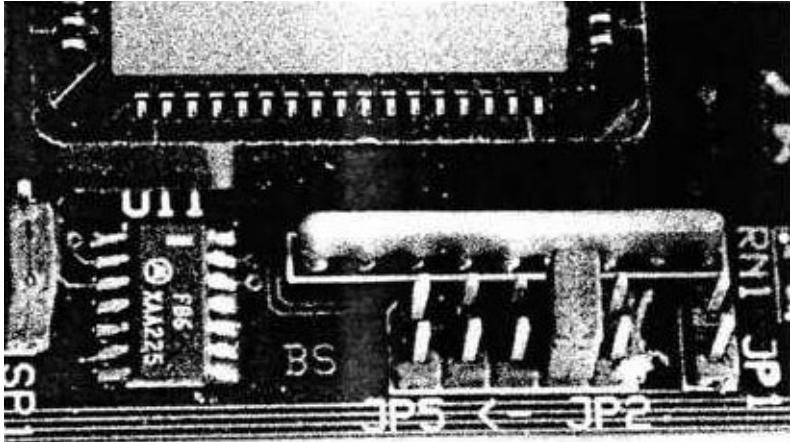
BESTÜCKUNGSDRUCK - BRÄUTEILSEITE



Der PROFESSIONAL-030PLUS500-BOARD Bestückungsplan



## 1.2 Jumperstellungen



- JP1 68030/68000 Umschaltung. Ist der Jumper gesteckt ist der 68000 Prozessor in Betrieb. Der 32 Bit Speicher kann nicht eingebunden werden. Ist kein Jumper vorhanden, dann bootet der Rechner im 68030 Modus und der 32 Bit Speicher kann eingebunden werden. Auch der mathematische Coprozessor kann jetzt genutzt werden. Das Turboboard ist aktiviert.
- JP5 Ist der Jumper gesteckt sind Daten- und Instructioncache hardwareseitig abgeschaltet. Es besteht keine Möglichkeit die Cache per Software zu einem späteren Zeitpunkt wieder zu aktivieren.
- JP6 Ist der Jumper gesteckt so ist die MMU (Memory Management Unit) im 68030 hardwareseitig abgeschaltet. Es besteht keine Möglichkeit die MMU zu einem späteren Zeitpunkt per Software zu aktivieren.
- SP1 Dient der Anpassung an Ihren Amiga. Sollte das Turboboard widererwartend in Ihrem Amiga nicht funktionieren, dann stecken Sie den Jumper um.

JP2-JP4 Hiermit wird die Speichergröße eingestellt, die auf Ihrem Board installiert ist.

JP2	JP3	JP4	Speicher	Adresse
○	○	○	1MB, RAM 1-8	090000-09FFFF hex
○	●	○	2MB, RAM 1-16	080000-09FFFF hex
●	○	○	4MB, RAM 1-8	060000-09FFFF hex
●	●	○	8MB, RAM 1-16	020000-09FFFF hex
○	○	●	1MB, RAM 1-8	4070000-407FFFF hex
○	●	●	2MB, RAM 1-16	4060000-407FFFF hex
●	○	●	4MB, RAM 1-8	4000000-403FFFF hex
●	●	●	8MB, RAM 1-16	4000000-407FFFF hex

● = Jumper geschlossen, ○ = Jumper offen

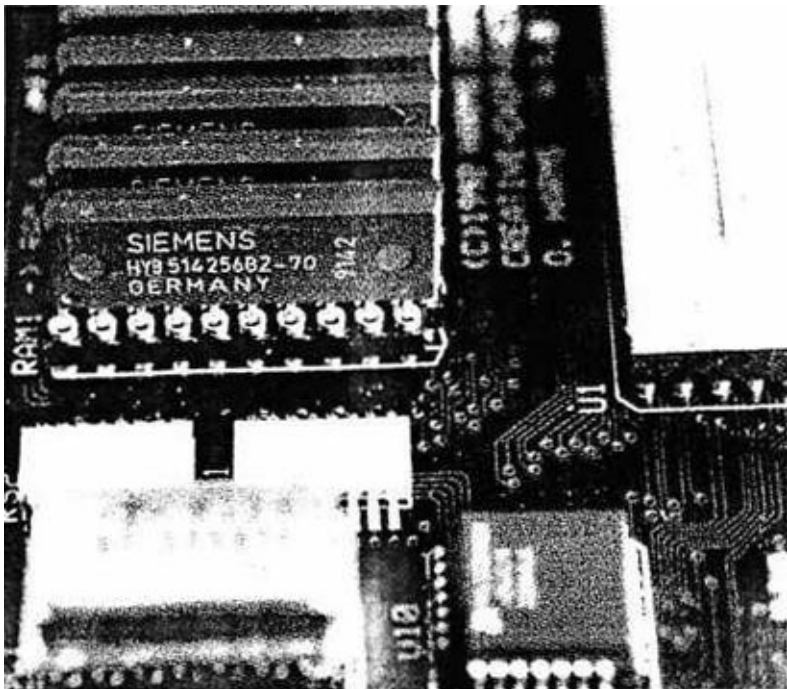
### 1.3 Cache Control Register

31	14	13				8	7		5	4			0	
00000000000000000000	WA	DBE	CD	CED	FD	ED	0	0	0	IBE	CI	CEI	FI	EI

- WA = Write Allocation
- DBE = Data Burst Enable
- CD = Clear Data Cache
- CED = Clear Entry in Data Cache
- FD = Freeze Data Cache
- ED = Enable Data Cache
- IBE = Instruction Burst Enable
- CI = Clear Instruction Cache
- CEI = Clear Entry in Instruction Cache
- FI = Freeze Instruction Cache
- EI = Enable Instruction Cache

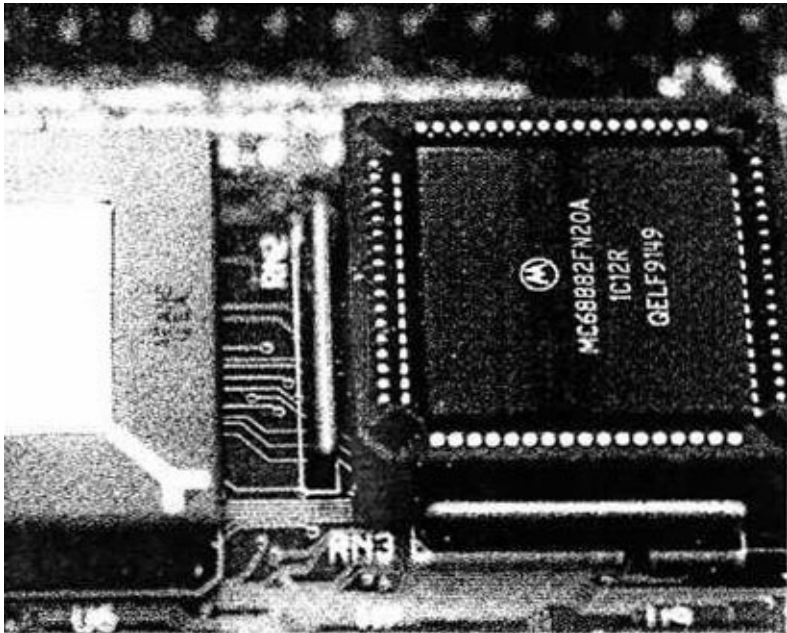
## 1.4 RAM-Bestückung

Die Speichergrößen ein und zwei MByte werden mit 1 MBit RAMs realisiert. Die Ausbaustufen vier und acht MByte werden mit 4 MBit RAMs realisiert. Die RAMs müssen eine 4 Bit Organisation aufweisen. Die RAMs haben dann z.B die Bezeichnung KM44C256AZ-7 bzw. KM44C1000AZ-7. Die Bauform der RAMs ist ZIP und sollten 80 ns Zugriffszeit haben oder schneller sein. Die RAM-Bausteine müssen mit der schrägen Seite nach außen in die Fassungen gesteckt werden. Es ist darauf zu achten, daß alle Beinchen in der Fassung stecken. Es dürfen keine umgebogen sein oder neben der Fassung stecken. Die Fassungen sind durchnummeriert von RAM 1 bis RAM 16. Die Jumper JP2 und JP3 sind entsprechend der RAM-Bestückung zu konfigurieren. Der Jumper JP4 hat hierbei eine besondere Stellung. Ist der Jumper gesetzt, dann wird der 32-Bit Speicher außerhalb des Amiga Autoconfig-Bereiches eingebunden. Das bedeutet, daß mehr als acht MByte in das System eingebunden werden können.



## 1.5 Der Coprozessor

Das PROFESSIONAL-030PLUS500-BOARD kann mit einem mathematischem Coprozessor ausgestattet werden. Die verwendbaren Typen sind MC68881 oder MC68882. Die Gehäuseform ist PLCC. Wenn Sie einen Coprozessor bestellen sollten, dann geben Sie bitte die Gehäuseform mit an. Ein Vorhandensein eines Coprozessors wird vom Turboboard selbstständig erkannt. Ein Quarzoszillator für den Coprozessor ist allerdings Voraussetzung. Zur Zeit ist eine maximale Taktfrequenz von 33 MHz für einen MC68882 machbar. Schnellere Typen (bis 50 MHz) können später (wenn verfügbar) eingesetzt werden. Das Board ist dafür schon ausgelegt. Der Coprozessor hat eine angeschrägte Kante. Diese Kante liegt der CPU am nächsten. Für das Entfernen eines Coprozessors empfehlen wir einen sogenannten "Chippuller". Das ist ein Werkzeug, mit dem man den Coprozessor aus dem PLCC-Sockel heben kann. Beim Einsetzen des Quarzoszillator ist auf die Markierung am Gehäuse und Sockel zu achten.



Coprozessor im PLCC -Gehäuse

## **1.6 Lieferumfang**

Das PROFESSIONAL-030PLUS500-BOARD umfaßt folgende Komponenten:

Das PROFESSIONAL-030PLUS500-BOARD

- Eine Utilitydiskette
- Eine Anleitung
- Eine Registrierungskarte

Wenn eines dieser Teile fehlt, setzen Sie sich umgehend mit uns in Verbindung. Wenn das PROFESSIONAL-030PLUS500-BOARD mit einem mathematischen Coprozessor und/oder zusätzlichem 32-Bit Speicher bestellt haben, sind diese Chips bereits installiert und getestet.

## **1.7 Rücksendung**

Es wird vorausgesetzt, daß Sie für eine eventuelle Rücksendung des Boards die vollständige Originalverpackung sorgfältig aufbewahren. Sollten die Originalverpackungsmaterialien nicht mehr vorhanden sein, so raten wir, die Platine in eine antistatische Ummantelung in einem gut geschützten Karton zu verpacken.

## 2.0 Einbau in den Amiga 500

Dieser Abschnitt beschreibt den Einbau des PROFESSIONAL-030PLUS500-Boards in den Amiga 500. Die Installation ist eine relativ einfache Sache, die jeder mit geringen mechanischen Kenntnissen durchführen kann. Sollten Sie jedoch die leisesten Zweifel haben, setzen Sie sich einfach mit uns in Verbindung. Wir werden Ihnen weiterhelfen. Für die Installation benötigen Sie folgendes Werkzeug:

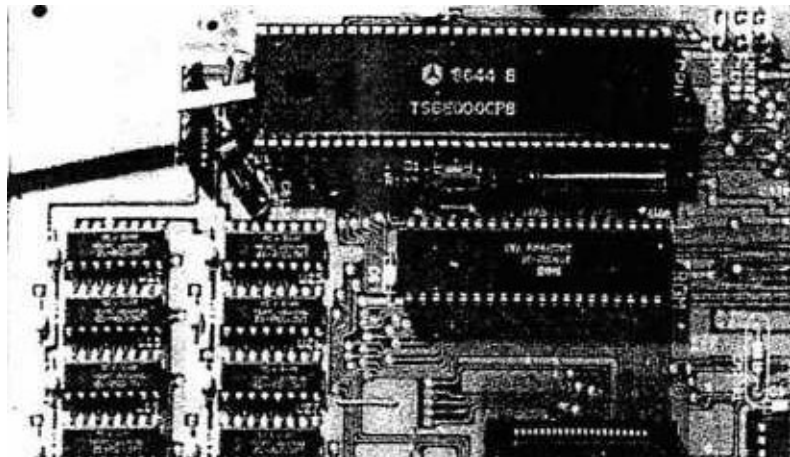
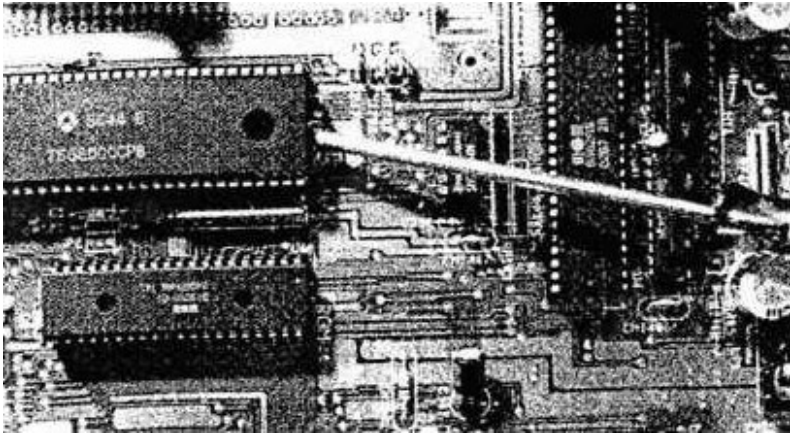
- Einen mittleren Kreuzschlitzschraubendreher zum Aufschrauben des Amiga Gehäuses.
- Einen flachen, möglichst breiten, Klingenschraubendreher, um die 68000 CPU aus ihrer Fassung zu hebeln.

Vor dem Öffnen des Amigas sind die Garantiebedingungen von Commodore zu beachten.

Weiterhin trennen Sie den Amiga vom Netz. Es sollte auch ausreichend Platz, beim Aufschrauben des Rechners zur Verfügung stehen. Hierzu ist es sinnvoll den Amiga von den Peripheriegeräten zu lösen.

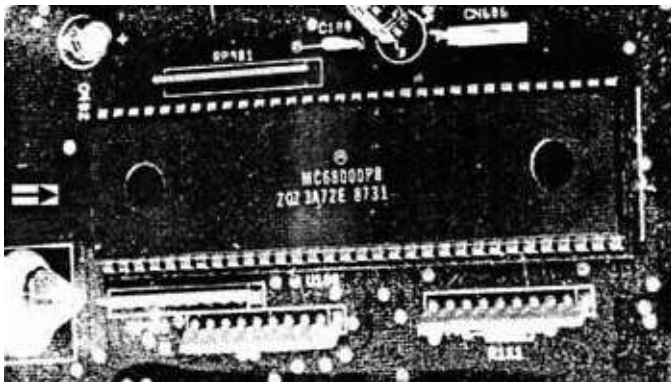
Der Amiga 500 wird umgedreht, so daß die Unterseite des Rechners frei zugänglich ist. An den langen Seite befinden sich jeweils drei kleine Kreuzschrauben, die gelöst werden müssen. Diese Schrauben sollte man gesondert aufbewahren, damit sie mit den noch zu lösenden Schrauben nicht verwechselt werden. Sind alle sechs Schrauben entfernt, kann man den Amiga wieder umdrehen. Jetzt kann man die obere Gehäusehälfte abnehmen. Als nächstes muß die Tastatur entfernt werden. Hierzu zieht man das Tastaturkabel vom Motherboard. Für den späteren Zusammenbau ist es hilfreich, sich die Orientierung des Steckers zu merken (makieren). Die Tastatur läßt sich jetzt schräg nach oben aus dem Amiga entfernen. Für den nächsten Schritt benötigt man wieder den Kreuzschraubendreher. Mit diesem werden die vier Schrauben der Abschirmung gelöst. Nachdem die Schrauben entfernt wurden, müssen noch ein paar Blechlaschen gerade gebogen werden, um die Abschirmung entfernen zu können. Unter der Abschirmung wird das Motherboard sichtbar. Auf der linken Seite beim Expansionsport befindet sich die CPU.

Der Mikroprozessor wird dann durch einen geeigneten Schraubendreher vorsichtig aus seiner Fassung gehiebt. Durch wechselseitiges Hebeln ist die Gefahr von verbogenen Beinchen am geringsten. Wer einen geeigneten Chippuller sein eigen nennt, kann auch diesen zum Einsatz bringen.



Die CPU wird dann in den Sockel auf dem PROFESSIONAL-030PLUS500-BOARD gesteckt. Die CPU und der Sockel haben Markierungen (Einkerbung), die beim Einstecken in der gleichen Stelle liegen müssen. Auch hierbei ist zu beachten, daß keines der Beinchen umbiegt. Nun kann das PROFESSIONAL-030PLUS500-BOARD in den freien 68000er Sockel des Amiga gesteckt werden. Wie schon beim Einbau des 68000er in den Sockel der Turboboards muß auch hier darauf geachtet werden, daß alle Pins in den Sockel kommen und keines dabei umknickt. Sollte es doch einmal vorkommen, daß ein Pin verbogen ist, dann biegen Sie es bitte ganz vorsichtig in die Ausgangsposition. Falls einer der Pins abgebrochen ist, wird das Turboboard nicht einwandfrei funktionieren.

Steckt das Board richtig fest im Sockel ist die Installation abgeschlossen. Es kann wieder mit dem Zusammenbau des Rechners begonnen werden. Da der Board, von den Abmessungen her, sehr dicht an die Abschirmung kommt, empfehlen wir das Abschirmblech nicht mit einzubauen. Die Tastatur wird von schräg oben unter die Plastiklaschen des unteren Gehäuseteils geschoben. Der Tastaturstecker wird wieder auf das Motherboard gesteckt (richtige Polung beachten). Abschließend wird das obere Gehäuseteil aufgesetzt. Der ganze Amiga wird dann wieder umgedreht und die sechs Schrauben eingesteckt und festgezogen. Nachdem die Stromversorgung und die Peripheriegeräte wieder angeschlossen wurden ist der Amiga wieder einsatzbereit.



Markierung am MC68000



## 2.1 Der erste Einsatz

Es ist nicht notwendig spezielle Software für den Betrieb des PROFESSIONAL-030PLUS500-BOARD vor dessen ersten Gebrauch zu laden. Ohne Änderungen in der Startup-sequence ihres Bootmedium ist das PROFESSIONAL-030PLUS500-BOARD sofort einsatzfähig. Um die Leistungsfähigkeit des PROFESSIONAL-030PLUS500-BOARD aber voll auszunutzen stehen mehrere Möglichkeiten zur Verfügung. Wichtigstes Programm für die Operationen ist das Programm SetCPU V1.6 und das CPU Programm von der Workbench 2.0. Mit diesem Programm ist es möglich das PROFESSIONAL-030PLUS500-BOARD noch effizienter einzustellen. Da der Amiga von Haus aus nur den Instruction-Cache aktiviert, liegt der Data-Cache brach. Mit dem Befehl `setcpu cache` wird auch noch der Data-Cache des MC68030 aktiviert. Eine weitere Möglichkeit die Geschwindigkeit zu steigern ist das Betriebssystem in den 32-Bit Speicher zu verlegen. Das hat zur Folge, daß die Betriebssystemaufrufe schneller abgearbeitet werden. Als kleinen Nebeneffekt muß man allerdings auf 256KB bzw. 512 KB 32-Bit Fastmem verzichten, der dann durch das Betriebssystem belegt ist. Der Befehl hierfür lautet `setcpu fastrom`. Diese beiden Befehle lassen sich auch zu einer Zeile zusammenfassen. Eingebunden in die Startup-sequence könnte die Zeile dann wie folgt aussehen:

```
setcpu cache fastrom
```

Voraussetzung hierfür ist allerdings, daß das Programm SetCPU V1.6 zuvor in das c: — Verzeichnis des Bootmedium von der Utility - Diskette kopiert wurde. In der Anleitung zu SetCPU finden Sie vielleicht noch die eine oder andere Idee Ihre Hardware noch besser auszunutzen. Allerdings ist Vorsicht geboten, denn eine falsche Einstellung bringt das PROFESSIONAL-030PLUS500-BOARD unweigerlich zum Absturz. Um allen Schwierigkeiten aus dem Weg zu gehen kann man auch die automatische Installation von der Utility - Diskette wählen.

## 2.2 Installation unter Kickstart 1.3

Für die Installation unter Kickstart 1.3 wird folgendes benötigt:

1. Kickstart 1.3
2. Workbench 1.3
3. Ein c: Verzeichnis auf Ihrem System
4. folgende CLI Befehle:
  - copy
  - if
  - else
  - echo
  - wait
  - join
  - delete
  - cd
5. Die verwendete Shell muß die Befehle skip und lab verstehen.

Ist einer dieser Punkte nicht erfüllt, kann es zu Komplikationen während der Installation kommen. Sorgen Sie also bitte dafür, daß alle Anforderungen erfüllt sind.

Die Installation der benötigten Software wird mittels eines Scriptprogrammes durchgeführt. Dieses Scriptprogramm befindet sich auf der mitgelieferten Utilitydiskette. Von der Workbench aus lassen sich die Verzeichnisse durch Doppelklicken öffnen. Für die Installation unter Kickstart 1.3 gibt es auf der Diskette ein Verzeichnis namens Install-1.3. In diesem Verzeichnis stehen die Scriptprogramme die für die unterschiedlichen Konfigurationen benötigt werden. Entsprechend der Ausstattung des PROFESSIONAL-030PLUS500-Boards kann nun eine Installation begonnen werden. Vom Professional-030PLUS500-BOARD wird der 32 — Bit Speicher durch einen Befehl in das Amiga System einbindet. Die Ausbaustufen 1, 2, 4, 8 MB lassen sich so softwareseitig in das System einbinden. Ist Ihnen beim Installieren ein Fehler unterlaufen ist das nicht weiter tragisch. Sie können eine andere Konfiguration einfach überinstallieren. Selbst wenn sich das PROFESSIONAL-030PLUS-BOARD nicht mehr im Amiga befindet, ist eine Änderung der Startup-Sequence nicht notwendig.

Soll der 32-Bit Speicher außerhalb des Autoconfigbereiches des Amigas eingebunden werden, so installieren Sie den Speicher als Highmem. Die Installationsscripts befinden sich in der Schublade Install-1.3-High.

## 2.3 Installation unter Kickstart 2.0

Für die Installation unter Kickstart 2.0 wird folgendes benötigt:

1. Kickstart 2.0
2. Workbench 2.0
3. Ein c: Verzeichnis auf Ihrem System
4. folgende CLI Befehle
  - copy
  - echo
  - wait
  - join
  - delete
5. Die verwendete Shell muß die Befehle if, else, cd skip und lab verstehen.

Ist einer dieser Punkte nicht erfüllt, kann es zu Komplikationen während der Installation kommen. Sorgen Sie also bitte dafür, daß alle Anforderungen erfüllt sind.

Die Installation der benötigten Software wird mittels eines Scriptprogrammes durchgeführt. Dieses Scriptprogramm befindet sich auf der mitgelieferten Utilitydiskette. Von der Workbench aus lassen sich die Verzeichnisse durch Doppelklicken öffnen. Für die Installation unter Kickstart 2.0 gibt es auf der Diskette ein Verzeichnis namens Install-2.0. In diesem Verzeichnis stehen die Scriptprogramme die für die unterschiedlichen Konfigurationen benötigt werden. Entsprechend der Ausstattung des PROFESSIONAL-030PLUS500-Boards kann nun eine Installation begonnen werden. Beim booten bindet ein Programm den 32 - Bit Speicher vom PROFESSIONAL-030PLUS500-BOARD in das Amiga System ein. Die Ausbaustufen 1, 2, 4, 8 MB lassen sich so softwareeseitig in das System einbinden. Ist Ihnen beim Installieren ein Fehler unterlaufen ist das nicht weiter tragisch. Sie können eine andere Konfiguration einfach überinstallieren. Selbst wenn sich das PROFESSIONAL-030PLUS500-BOARD nicht mehr im Amiga befindet, ist eine Änderung der Startup-Sequence nicht notwendig.

Soll der 32-Bit Speicher außerhalb des Autoconfigbereiches des Amigas eingebunden werden, so installieren Sie den Speicher als Highmem. Die Installationsscripts befinden sich in der Schublade Install-2.0-High.

### 3. Software

Auf der mitgelieferten Diskette befinden sich folgende Programme:

```
MemRoutines (dir)
    .info
    memcpy.asm
    memory1.lib
    readme
    Readme.info
JoyLib (dir)
    .info
    joy0.o
    joydemo.c
    JoyLib.doc
    joystick.h
    joystick_lib.fd
    linklib.with
BenchMarks (dir)
    AmigaBench (dir)
        AmigaBench
        AmigaBench.asm.info
        AmigaBench.doc.info
        AMIGA-Performance-68000
        AMIGA-Performance-68020
        AmigaBench.info
    AmigaBench.asm
    AmigaBench .doc
    AmigaBench.info
    AMIGA-Performance-68000.info
    AMIGA-Performance-68020.info
SetCPU (dir)
    CardROMList
    control.i
    diskio.c
    idents.a
    memory.c
    mmu.c
    POSTER
    setcpu
    SetCPU.h
    SetCPU.txt
    control.a
    coolhand.c
    expdev.c
    makefile
    misc.c
    other.a
    reboot.a
    SetCPU.c
    setcpu.i
    SetCPU.txt.info
Mathtrans (dir)
    .info
    BenchMark.c
    Mathtrans.DOC
    mathtrans.library
    ReadMe.fnf
    StartBenchMark
    BenchMark
    BenchMark.info
    Mathtrans.DOC.info
    MatLib.asm
    ReadMe.fnf.info
    StartBenchMark.info
Install-1.3 (dir)
1MB-Install-SoftConfig1.3
2MB-Install-HardConfig1.3
2MB-Install-SoftConfig1.3
3MB-Install-SoftConfig1.3
4MB-Install-HardConfig1.3
4MB-Install-SoftConfig1.3
8MB-Install-HardConfig1.3
p030-boardconfig-hard-1.3-2mb
p030-boardconfig-hard-1.3-8mb
p030-boardconfig-soft-1.3-2mb
1MB-Install-SoftConfig1.3.info
2MB-Install-HardConfig1.3.info
2MB-Install-SoftConfig1.3.info
3MB-Install-SoftConfig1.3.info
4MB-Install-HardConfig1.3.info
4MB-Install-SoftConfig1.3.info
8MB-Install-HardConfig1.3.info
p030-boardconfig-hard-1.3-4mb
p030-boardconfig-soft-1.3-1mb
p030-boardconfig-soft-1.3-3mb
```

```

p030-boardconfig-soft-1.3-2mb Startup-P030
Install-2.0 (dir)
1MB-Install-SoftConfig2.0 1MB-Install-SoftConfig2.0.info
2MB-Install-HardConfig2.0 2MB-Install-HardConfig2.0.info
2MB-Install-SoftConfig2.0 2MB-Install-SoftConfig2.0.info
3MB-Install-SoftConfig2.0 3MB-Install-SoftConfig2.0.info
4MB-Install-HardConfig2.0 4MB-Install-HardConfig2.0.info
4MB-Install-SoftConfig2.0 4MB-Install-SoftConfig2.0.info
8MB-Install-HardConfig2.0.info p030-config-hard-2.0-2mb
p030-config-hard-2.0-2mb.info p030-config-hard-2.0-4mb
p030-config-hard-2.0-4mb.info p030-config-hard-2.0-8mb
p030-config-hard-2.0-8mb.info p030-config-soft-2.0-1mb
p030-config-soft-2.0-1mb.info p030-config-soft-2.0-2mb
p030-config-soft-2.0-2mb.info p030-config-soft-2.0-3mb
p030-config-soft-2.0-3mb.info p030-config-soft-2.0-4mb
p030-config-soft-2.0-4mb.info
Shell-Tools (dir)
Sources (dir)
float.c savage.c
sieve.c TurboRamTest.c
whetstone.c
float.020.881 float.020.A.IEEE
float.020.FFP float.68000
float.A.IEEE float.FFP
float.m.IEEE savage.020.881
savage.020.FFP savage.020.IEEE
savage.68000.ffp savage.68000.ieee
sieve turboramtest
whetstone.020.881 whetstone.68000
addmem BenchMarks.info
Disk.info icony
Install-1.3.info Install-2.0.info
joyleft.info Mathtrans.info
memroutines.info Muchmore
Muchmore.info readme_first
readme_first.info SetCPU.info
Shell-Tools.info

```

Im readme\_first stehen Informationen, die zum Zeitpunkt der Drucklegung noch nicht berücksichtigt worden sind.

## 3.1 Testprogramme

Im Unterverzeichnis Shrltools befinden sich einige Programme, mit deren Hilfe man die Funktionen des MC68030 und der FPU 68881/2 testen kann. Dadurch sind Leistungsvergleiche zwischen den verschiedenen Systemkonfigurationen möglich. Durch Systemänderungen hat man die Möglichkeit auszutesten, bei welcher Konfiguration das System die größte Leistung bringt. Anhand der Dateieindungen, kann man die Art der Compilation erkennen. So ist z.B die Datei float.68000 für den MC68000 compiliert und die Datei float.030.881.IEEE für den MC68000 und FPU68881 unter Verwendung der IEEE Mathe-Library. Das .a. steht für die Amiga-Library und das .m. für die Manx-Library.

### 3.1.1 Float

Mit diesem Test werden Leistungsdaten von Fließkommaoperationen der unterschiedlichen Prozessoren ermittelt. Der Code liegt in mehreren Variationen vor:

- Für den MC68030 mit FPU68881/2 Unterstützung
- Für den MC68030 mit IEEE Mathe-Library
- Für den MC68030 mit MANX Mathe-Library
- Für den MC68030 mit Motorola Fast Floating Point-Emulation
- Für den MC68000

Das Source - Listing des Programms Float befindet sich in dem Unterverzeichnis Shell-Tools/Sources. Durch unterschiedliche Optionen am Compiler und Linker erhält man den Code für die verschiedenen Prozessoren.

### 3.1.2 Savage-Test

Dieses Programm gibt es in fünf verschiedenen Ausführungen:

- Savage.030.881 Ist ein reiner Floatingpointtest des mathematischen Coprozessors.
- Savage.68000.ieee Ist ein Floatingpointtest nach ieee Standard. Hierzu werden mindestens die neuen Libraries der Workbench 1.3 benötigt. Diese Libraries unterstützen auch den mathematischen Coprozessor. Den Vorteil, den man durch die Verwendung dieser Library hat, liegt darin begründet, daß der Code mit und ohne Coprozessor läuft.

- Savage.68000.ffp Ist ein Floatingpointtest nach Motorola Fast-Floating-Point Standard.
- Savage.030.ieee Hier wird die Architektur des 68030 mit zur Optimierung ausgenutzt
- Savage.68030.ffp Auch hier wird die Architektur des 68030 mit ausgenutzt.

Das Source - Listing des Programms Savage befindet sich im Unterverzeichnis Shell-Tools/Sources. Durch unterschiedliche Optionen am Compiler und Linker erhält man den Code für die verschiedenen Prozessoren.

### **3.1.3 Whetstone-Test**

Der Whetstone Test ist ein kombinierter Test aus CPU- und Coprozessorleistung. Es ist hiermit ein Vergleich mit einem PC möglich, auf dem es diesen Test auch gibt. Die einzelnen Operationen werden beim Programmablauf ausgewiesen.

Das Source - Listing des Programms Whetstone befindet sich im Unterverzeichnis Shell-Tools/Sources. Durch unterschiedliche Optionen am Compiler und Linker erhält man den Code für die verschiedenen Prozessoren.

### 3.1.4 AmigaBench

AmigaBench ist ein für den Amiga angepaßte Version des allgemein bekannten Benchmark-Tests Dhrystone. Es beinhaltet mehrere Optionen. Diese Optionen bekommt man durch Anhängen des Fragezeichens an den Befehl aufgelistet. Das Programm wurde geschrieben von Al Aburto, von dem auch folgende Anleitung stammt:

AmigaBench  
Al Aburto

'ala' on BIX

05 Aug 1989

Hello Fellow Amigeans!

AmigaBench is a program in which I intended to include 680X0 Amiga assembly optimized versions of the more popular so-called "benchmark" programs.

I've only made it to the Dhrystone so far, but I thought the results are interesting enough to upload what I got now. Besides I think someone like 'Jez' could find even more optimizations to do :-).

AmigaBench will run from Workbench or the CLI/AmigaShell. I included a dumb ICON, executable file, and the assembly source code. Its all publicdomain of course.

Just type '?' and hit the RETURN key and a list of the program commands will be printed to the screen.

The 'ca' command just shows you where in memory the program was loaded (\_main), the addresses of some of the routines, and the location of the global data (\_A4Ref).

The 'cs' command prints out the Cache Status if you have an 020 or 030 CPU.



The 'wc' command allows you to Write to the Cache (if you have an 020 or 030 CPU). For example '51' will turn the 'Write Allocate' feature ON, and '50' will turn it OFF. The various options are shown.

- The '1' command runs what I call the 'Standard 68000 Assembly Version' of the Dhrystone. The run time in TICKS and the Dhrystones/sec are printed out. Also your CPU/FPU type is printed out. I don't check for the 68030 and 68882 in the program (I have another independent program in my startup-sequence that I use). So if the ExecBase flags are not set right AmigaBench will not report the 68030 or 68882 even if they are really installed. I'll fix this later.
- The '2' command runs an 'Optimized 68000 Assembly Version' of the Dhrystone.
- The '3' command runs an 'Optimized 68020 Assembly Version' of the Dhrystone. This should be the fastest version on those 020 and 030 systems.

That's it for now.

I'm getting about 5400 Dhrystones/sec from my CSA 14.32 MHz 68020 board with 32-bit fast static RAM. I'm sure Ronin will be faster and the GVP 030 really fast at 25 MHz or more. Post or send me results please.

Al Aburto

05 Aug 1989

## 3.2 Utility - Programme

Auf der Diskette befinden sich noch weitere Programme. Diese sind thematisch in Unterverzeichnisse gegliedert.

### 3.2.1 SetCPU V 1.6 Originalanleitung

SetCPU V1.60

by Dave Haynie

June 15. 1990

SetCPU V1.60 is a program designed for identification and modification of system parameters roughly related to different versions of the Motorola 68000 family processors. The program will identify the various types of processors and coprocessors in any 680x0 system. It also makes an attempt to correctly identify an incorrectly designed but still possibly functional 68020 system, several of which are known to exist as Amiga coprocessor boards. It contains MMU code to locate kernel ROM in write protected 32 bit ROM.

In any case, the syntax of the program is given as follows:

```
SetCPU [INST|DATA] [[NO]CACHE|[NO]BURST] [CONFIG n]
[BITS n] [TRAP n] [KICKROM path|dfN:] [DELAY n]
[KEEPEXEC] [CARDROM path] [VERBOSE] [[NO]FASTROM
[path] [KEYPATCH n] [HEAD] [NOSTACK]] [ROMBOOT] [CHECK
680x0|68851|6888x|MMU|FPU|MMUON|MMUROM|MMUALIEN]
```

where '[' ]' indicates an optional parameter, '|' indicates a choice of parameters. Typing 'SetCPU ?' will retrieve this same syntax diagram. Typing SetCPU alone will result in the SYSTEM configuration being send to the console, my current system returns this:

```
SYSTEM: 68030 68882 FASTROM (INST: CACHE NOBURST)
(DATA: CACHE NOBURST)
```

This indicates I have a 68030/68882 system, I've previously installed the FASTROM translation, and both caches are turned on. Issuing the command 'SetCPU fastom cache' would recreate such a setup. Note that any parameters that don't make sense to the real system configuration,

such as asking to modify the data cache on a 68020 system or install the FASTROM translation on a 68000 system are just ignored.

## **1. CHANGES**

- Since the V1.50 release of SetCPU, the following changes have been made to the SetCPU program:
- FASTROM now supports loading of a ROM image file.
- The patch manager has been removed, except for the KEYPATCH option.
- ROMs beyond 256K are now supported, from both file and special KickStart disk.
- ROM images are supported assembled for various memory locations, so that Commodore's developer files assembled at \$00F00000, for example, will work. SetCPU determines the size, base, and jump address of a ROM independently.
- ExecBase can be erased on KICKROM resets to prevent Chip RAM sizing problems and other OS switching errors.
- A programmable KICKROM DELAY option allows KICKROM to work on better on some machines with extremely slow 8520 startup times.
- Caching for any Bridge Card memory will always be disabled.
- The system stack, if found in Chip memory, is relocated to Fast memory for FASTROM translations.
- Some 2.0 functions are activated when run under the 2.0 OS, so that most if not all of the basic SetCPU functions operate properly.
- The MMU code is intelligent enough to avoid trashing an MMU setup not generated by SetCPU.

## **2. DISTRIBUTION**

This program is placed in the public domain, and may be used or distributed as you like.

## **3. CPU IDENTIFICATION**

There are two basic types of functions performed by SetCPU. The first of these is CPU system identification and cache control. SetCPU will tell about the type of CPU setup in your machine, which consists of the CPU itself and sometimes FPU or MMU coprocessors. If the CPU supports caches, SetCPU will let you switch these caches, and associated cache line burst mode, on and off. Finally, SetCPU can be used in a Startup-Sequence or other script to make decisions based on the system that's running. This is quite useful with accelerator cards like the Commodore A2620 that let you boot the machine with either 68020 or 68000 in charge. The individual CPU group commands are given below in detail:

### **[NO]CACHE**

This command will switch on or off 68020 and 68030 caches. If not qualified, it'll act on both instruction and data caches of the 68030.

### **[NO]BURST**

This command will switch on or off the burst cache line fill request of the 68030. If not qualified, it'll act on both instruction and data caches.

### **INST**

This qualifies a CACHE or BURST operation to restrict its application to the instruction cache only.

### **DATA**

This qualifies a CACHE or BURST operation to restrict its application to the data cache only.

## CHECK

This option lets you check for the existence of a particular CPU system component in a script. It works like this:

```
SetCPU CHECK 68020

If WARN
    echo ``No 68020 here!``
Else
    echo ``Sho nuff got a 68020 here!``
Endif
```

The arguments to CHECK can be any of:

68000	Matches the obvious
68010	“
68020	“
68030	“
68040	“
68851	“
68881	“
68882	“
FPU	Matches 68881, 68882, or 68040
MMU	Matches 68851, 68030, or 68040
MMUON	Matches any case in which the MMU is enabled
MMUROM	Matches an active SetCPU ROM translation
MMUALIEN	Matches any MMU setup no created by SetCPU

If any cache parameter doesn't apply to the system in use, it'll just be ignored. Use the data cache and all burst modes with caution. Some 68030 systems aren't designed to correctly support the data cache, so switching it on may cause an instant system crash. Even on systems that correctly support the 68030 data cache, some device drivers, especially those for DMA devices, may not work properly with the data cache enabled. You may wish to check with your system vendors to make sure before using the data cache in your standard system setup. The Commodore A2091's device driver does correctly support data caching. However, the use of the data cache is not recommended without an MMU setup, such as FASTROM or one of the KICK setups, invoked. The Amiga OS uses memory that's the same in both Supervisor and User modes of the 680x0. This requires the

setting of the 68030's Write Allocate bit for safe operation, and SetCPU will always insure that Write Allocate is set. However, this mode causes the data cache to be updated on longword writes even for locations that are driven noncacheable in hardware. So data caching without a proper MMU setup can cause problems with some I/O devices. With the MMU setup, SetCPU will map the standard Amiga I/O regions as noncacheable.

SetCPU may report a "FPU Logic Error" on certain 68020 systems. This is indicating a hardware problem with that board's floating point coprocessor decoding, which results in the FPU responding to the MMU addresses as well as its own. SetCPU knows how to handle such a board, but future software using the MMU may not, so it's a good idea to report this problem to the board vendor for repair.

#### **4, ROM TRANSLATIONS**

The second thing that SetCPU V1.60 manages are ROM translations. Using the MMU on systems so equipped, it can locate the Kernel ROM in the much faster 32 bit wide memory provided on many 32 bit systems. It can also boot a ROM based system with an alternate version of KickStart. Most of the options here relate to MMU translation setup and various modifications of the basic translation premise.

As of this release, SetCPU's MMU configurations will support memory outside of the 68000's 24 bit address space, when it is present. Such memory will be automatically recognized and supported by SetCPU if it is linked into the free memory lists when SetCPU builds its FASTROM or KICKROM. Alternatively, the number of significant bits of address in the system can be specified by the BITS command, and SetCPU will build the appropriate MMU table for such a system.

Another feature of this release is support for ROM images of either 256K or 512K in size. KickROMs may be assembled for locations other than the \$00FC0000 or \$00F80000 base used by physical systems. SetCPU will compute the size, base address, and start address for any KickROM image. ROM images can now be loaded from disk for FASTROM translations as well as KICKROM translations, though a FASTROM image must be the same KickStart release as the current ROM in the system. This facility's main purpose is to support loading of patched ROMs with the necessity of rebooting via KICKROM. As a result of this, the only patch now done by SetCPU itself is the optional KEYPATCH, which may be required for proper operation of the accelerated ROM code on some systems.

## **[NO]FASTROM**

This activates the FASTROM translation on or off an MMU equipped system. When switching on, it first allocates at least 256K of memory for the ROM image, then at least 512 bytes of memory for the MMU table. It copies the ROM into the image area, then applies the translation by-pointing the MMU at the table and activating it. The NOFASTROM option will switch off the MMU and reclaim the memory used for the ROM image and MMU table. If any other program set up the MMU for something, invoking this option could be a very bad thing to do. In general, until there's some level of OS support for the MMU in Amiga systems, you're really safe using only one MMU tool at a time. If you have an A2620 or A2630 system, this option will always get 32 bit memory for you; if not, you'll have to make sure that your 32 bit memory is the first MEMF\_FAST memory in the memory list for it to be used for the ROM image. Also, that ROM image will be allocated as far back on that memory list as possible unless the "HEAD" option is specified. The SetCPU "SYSTEM" line will report this setup as a "FASTROM" setup.

Suboptions are:

### **[file]**

Specifying a file with a valid ROM image will load that ROM image instead of the system's physical ROM image. The one restriction is that the disk-loaded ROM image must be the same ROM revision as the current system ROM. If they aren't the same revision, the KICKROM option can be used instead, but that'll require a reboot. This option can be used to load a patched version of the current ROM without reboot in most cases.

### **[KEYPATCH n]**

This will patch the keyboard scanning routine for machines that have Cherry keyboards (small function keys). The "n" parameter allows a variable delay between 1 and 100 to be specified; the delay depends on the keyboard, but should be pretty independent of CPU speed.

### **[NOSTACK]**

This will prevent the attempted translation of supervisor stack into 32 bit memory. By default, such translation will be done if the supervisor stack is found in Chip memory.

## **CARDROM path**

When used in conjunction with the FASTROM option, this allows ROMs from expansion cards to be located in fast memory as well. The path should reference a file containing lists of expansion cards that should be translated if found. It's necessary to read this from a user-defined file, rather than from the expansion environment itself, since an expansion device's ROM could be located close to that device's registers; there's no way for SetCPU to know it's safe to translate a card ROM image unless you tell it. On my system I read a file called CardROMList, which currently contains the single line:

```
0x202 0x01 0x10000 0x8000 0x4000 CBM_2090A_Disk_Controller
```

All the numbers given are in C language hex format. The parameters are, in order, the device's manufacturer code, product code, the device's size (in bytes), the ROM's offset from the configured board's base address (in bytes), and the size of the ROM area to be translated (in bytes). The final item is text string to identify the device; this'll be displayed by the VERBOSE option if the ROM translation does in fact take place. The '\_' characters in the name will be translated to ' ' characters. Note that the CardROM translations are currently based on 16K chunks, and SetCPU will ignore requests for translations of less than 16K, and round down to 16K boundaries for larger translation requests.

## **HEAD**

This option causes the SetCPU memory allocator to attempt memory allocation for its translated objects from the start of 32 bit memory instead from the end, as it usually docs. Allocation from the end usually results in less fragmentation than from the start (due to the alignment restrictions of MMU objects), though this option is useful when dealing with merged memory' lists. It is ignored when the ROM image and tables are in chip \$00C00000 memory.

## **KICKROM path|dfN:**

The KickROM option allows the system to be restarted with an alternate ROM image. This can be from a KickStart disk in a specified floppy drive, or from a given file name. If the ROM image is accessible, this command will cause the system to be immediately rebooted into the new OS. Note that pre-1.3 versions of the Amiga operating system will probably have



some trouble with expansion cards, especially autoboot cards. For that reason there's the CONFIG 0 option, which is explained later.

The KICKROM command will reboot the machine with the new OS, but that ROM image will be physically located in either memory at \$00C00000, if it's available, otherwise it'll use Chip memory. Once the new OS has started up, issuing either 'SetCPU FASTROM ...' or 'SetCPU KICKROM...' will cause that image to be moved into fast memory, and the slow memory will be given back to the system. The SetCPU 'SYSTEM' line will report a 16 bit KICKROM image as a 'SLOWKICK', and a 32 bit KICKROM image as a 'FASTKICK'. A machine running from a SLOWKICK kernel can't be re-KICKROMed, but can be from a FASTKICK kernel.

Suboptions are:

**[DELAY n]**

This option sets the delay after reset before any code is run. The parameter may be set from 0 to 100, where 0 sets no delay. The default value is 10, which sets the delay value used in SetCPU V1.50. The need for a delay depends on the machine you're on. If your machine hangs after SetCPU loads KickStart, there's a real good chance that you need a longer delay.

**[KEEPEXEC]**

Normally, KICKROM will clear the ExecBase pointer, to cause a the new version of the ROM to be rebuild from a cold boot condition. This option prevents that clearing action.

**CONFIG n**

This option controls if and how expansion devices are recognized on a KICKROM boot. At the default configuration level, level 2, the expansion cards are left alone, allowing the new Kernel to try and configure them. Since some older operating system will choke on autoboot devices, this option will allow suppression of them for the rebooting process. When requesting a KICKROM boot, a CONFIG level of 0 or 1 will prevent the devices from being recognized.

Once rebooted in the new OS, moving from a SLOW to a FAST Kick-image, as described above, the CONFIG status will be honored. If the devices weren't suppressed, nothing special happens. If they were, they'll stay suppressed, and you very likely won't have the memory to support a FAST Kick image. Specifying a CONFIG level of 2 at this point will attempt to configure the devices without autobooting. At level 1, the devices will be made visible to the system again, but nothing will be done with them.

As of the latest release, CONFIG 0 appears to be required with the 1.2 operating system, at least if there's any autoboot device, even if you're attempting to move from a slow to fast kick image. The next release will attempt to allow 1.2 to configure non-autobooting devices at this point.

## **TRAP n**

This option controls the level of error trapping handled for you by the SetCPU system. The numeric parameter is actually optional for compatibility with SetCPU V1.4. If no TRAP is specified, the default level 2 is enacted. If the TRAP command is given without a parameter, trap level 0 will be setup.

Trap level 0 causes the MMU to look at all 32 bits of address; access to any memory outside of the 24 bit space will result in an exception, which if unhandled, results in a GURU 2. Trap level 1 will set up the MMU to only look at 24 bits of address space. Trap level 2 works like level 1, but additionally sets up a trap handler for the Bus Error exception (which usually surfaces as a GURU #2), For normal operation (eg, running other people's code), Trap level 2 is probably what you want. For final testing of your own code, levels 0 or 1 can catch things which would go unnoticed on a 68000 machine, such as writing to ROM space or out of the 24 bit address space.

The exception handler used for level 2 trapping catches things like writes to protected areas of memory. It just tells the bus machine not to complete the write, and signals no error. There's a slight chance that this won't be enough repair for a program doing something really outlandish - at that point, running at level 1 will let the GURU happen, which might help if you're debugging your own code. Other than that, there's probably nothing you can do to get such a program working with the MMU turned on, other than having it fixed. The other thing to consider is that this exception

handler could conflict with another system-level handler installed by a GOMF-like program. That shouldn't cause a big problem, since you'll the one that was installed later, both of which presumably trap the error, but it's something to be aware of. Under V1.3 and earlier releases, a DOS bug can cause invalid accesses, which cause the exception, when running the EndCLI or NewCLI/NewShell programs; running at level 1 or 2 will avoid gurus with these commands.

## **BITS n**

This option forces the MMU table for KICKROM or FASTROM to be built to support a specific number of bits, regardless of the actual bits apparently used by the system. Valid significant bits range from 24 through 32.

## **ROMBOOT**

This option forces a reset to physical ROM without hanging the system, even when the MMU is active.

## **VERBOSE**

This option more fully describes the system translations.

## **5. ROM FILE FORMATS**

The 256K KickStart disk the KICKROM option will look for is the standard Commodore KickStart format, which is a standard format floppy with the work 'KICK' at the start of the disk, followed by 512 blocks of 512 bytes each, a plain dump of the ROM image. 512K KickStart disks look just the same, only, of course, with 512K of ROM. The KickStart loader will actually check the first 32 blocks of disk for the start of ROM.

The size of the KickStart image is determined by the first longword of the file. Optionally, disk files can contain two extra longwords at the beginning of the file, the first being a \$00000000, second the expected size of the ROM image. That expected size will be compared with the expected based on the first longword in the ROM image and the actual length of the loaded file, in the case of a disk file rather than a KickStart. The base address and starting address are determined from the ROM image, and

SetCPU will attempt to use them. Some ROM images can cause a conflict with other system resources.

## **6. CREDITS**

While this program is an entirely original work, nothing happens in vacuum, this one included I'd like to mention folks who, directly or indirectly, helped make this thing happen, by providing example MMU code, suggestions, and incentives. These folks include Neil Katin, Jez San, GVP Inc., Dale Luck, Bryce Nesbitt, Andy Finkel, and the other Commodore-Amiga software people, and the Commodore-Amiga Technical Support folks.

## **7. POTENTIAL BUGS AND OTHER NOTES**

I should point out here that much of what SetCPU does is of a rather dubious nature. Everything that's possible to do correctly under the 2.0 release of the OS, including CPU/FPU identifications and cache control, is done via the approved 2.0 methods when running in 2.0, and via my own tricks when run under 1.3 or earlier releases.

While it's impossible for an application to correctly use the MMU under 1.3 or 2.0, SetCPU attempts to be intelligent about its use of the MMU. It will check for the use of the MMU by an agent other than SetCPU, and refuse to modify the current MMU setup if such an alien MMU setup is found. For systems with an unused MMU, SetCPU will do it's best to be safe about the modifications it makes to the memory map. There may be problems with this program's MMU code on the Amiga 3000, but based on the SuperKickStart and CPU programs that are shipped with the first A3000s, this should not be of immediate concern, since the A3000 comes with equivalent functionality. Should an A3000-safe version of SetCPU eventually become needed, I'll make the attempt to track down any A3000 bugs I can find. Just because you help design a machine doesn't necessarily mean you have one to code and test on at home.

Finally, if you wish to contact me regarding bug reports, new releases, contributions of cash or macadamia nuts, or pretty much anything else, I can be reached at the below addresses.

-Dave Haynie

Logical Address:

PLINK:	D-Dave H
bix:	hazy
usenet:	{uunet.rutgers}!cbmvax!daveh

Physical Address:

284 Memorial Drive  
Gibbstown, NJ  
08027

### 3.2.2 Memroutines

A few months ago, when I was doing some serious debugging, I accidentally discovered that the Lattice C functions memcpy, memcmp, and memset, as well as the Aztec C functions movmem and setmem, all deal with data one byte at a time. With a 68000 processor, this doesn't matter much, especially if you take advantage of Lattice C's capability to deal with these functions as "built-in". However, Amigas with 68020 and 68030 processors, and 32-bit wide memory, are starting to proliferate. With this sort of hardware, you can definitely improve performance by processing data a long word at a time, instead of a byte at a time.

The three functions in this package, memcopy(), memcmp(), and memset(), are 'plug-compatible' replacements for the Lattice functions of the same name.

```
char *memcpy(),*memset(),*to,*from,*s1,*s2,*toaddr,ch;
```

```
long int memcmp(),count;
```

```
toaddr = memcpy(to, from, count);
```

```
from = pointer to data to be copied
to = pointer to area that data is to be copied to
count = number of bytes to be copied
toaddr = pointer to area that data was copied to
```

```
x = memcmp(s1, s2, count);
```

```
s1, s2 = pointers to data to be compared
count = number of bytes to be compared
x = 0 if s1 and s2 are identical; Contains a negative value
if s1 < s2, or a positive value if s1 > s2
```

```
toaddr = memset(to, ch, count);
```

```
to = pointer to area that is to be set
ch = value to set area to
count = number of bytes to set
toaddr = pointer to area that was set
```

Note that with Aztec C, the 'count' parameter must be 'long'. Two libraries are provided, `memorya.lib`, the Aztec version (sorry, 3.6a), and `memoryl.lib`, the Lattice version. When linking, this file must be included before the normal library (`c.lib` for Aztec, `lc.lib` for Lattice). If you are using Lattice C, `#include string.h` statements should be removed or commented out. Using these routines with Aztec C should never cause a performance penalty, even with a 68000 processor, unless you call them a disproportionately large number of times with parameters that are not word-aligned. With Lattice C, you may find a degradation in performance if most of your calls to the functions have a value of less than 8 for 'count'. This is because using the built-in versions of the functions means less overhead entering and exiting the functions.

This software is public-domain.

Robert Broughton  
328-1027 Davie St.  
Vancouver, BC V6E 4L2  
Canada  
USENet: a1040@mindlink.UUCP

### 3.2.3 JoyLib

Bei dieser Software handelt es sich um eine sehr schnelle Routine für die Joystickabfrage. Sie ist so geschrieben, daß sie auf jedem 680x0 Prozessor läuft. Die Software liegt in zwei Versionen vor. Die erste ist eine Standard 'o'-Routine für alle Sprachen, die in der Lage sind '\_joy0/1' zu verwenden. Diese Routine ist kompatibel zum BLINK-Format. Der andere Teil ist eine Version, der Modula und anderen Sprachen zugeordnet ist, die '.fd' zur Bildung eines Interfaces zu einer anderen Sprache benötigen.

```
JoyLib
#####
```

Hello World! This is a new joystick routine with a special algorithm I'd call "Advanced Hufmann Word Shift Comparison" (just kidding).

There are two versions of the routines:

- 1) standard ".o" linkage for any language able to call "joy0/1" and compatible to BLINK format.
- 2) a shared library (thanks to BLINK) for such nifty things like Modula and all... A .fd for building interfaces to \*any\* language is provided.

The routines will return a single argument in D0, which bits represent joystick directions:

UP = 4

LEFT = 1   RIGHT = 2

DOWN = 8

FIRE = 16

Simple enough, isn't it?

This stuff is Public Domain; use it, inflame it, molest it...

Olli

\*\*\*->

Snail to: Oliver Wagner   Lansberge 5   4322 Sprockhoevel   West Germany



### 3.2.4 FPU-Mathtrans.Library V1.1

FPU-Mathtrans.Library V1.1  
28.1.1990

(c) by Heiner Hückstädt

This library was written for all people, who own the Motorola floating point unit MC 68881, MC 68882 or MC 68040 (haha) and never use it, because most of the software, written on the Amiga make use of the Fast Floating Point format, which is absolutely undigestible for the numeric coprocessor. (It's a shame !!) Only a few software developers spread a recompiled FPU version, too.

This library is to increase the performance of that software products, who take much use of the FFP-transcendental functions (Sin, Cos, Pow, etc.) like Distant Suns (very very good program, but where is the 68020/881 version) or others.

#### How to use this library

You must have a 68020/68030 - 68881/68882 Turboboard ! If so, copy this library to your LIBS: directory. The old mathtrans.library will be deleted.

#### Realisation

The FPU MC68881/82 knows for every function (except SPTieee, SPFieee and SPPow) a hardware instruction like fsin.x, fcox.x, flogn.x, etc. But all inputs for the mathtrans.library and the mathffp.library come in this #\$\$-FFP-format, so every input (and output, too) has to be converted from FFP to IEEE single precision and back using SPTieee and SPFieee. (Excuse me Commodore, but these functions were stolen from your original mathtrans.library.)

#### Profit

On my configuration (68020 7Mhz, 68881 25Mhz, no 32 bit-ram) my genius BenchMark shows a speed increase of 7.1 overall. This may differ due to your money, megahertz and megabytes.

## THIS PROGRAM IS IN THE PUBLIC DOMAIN

Permission is hereby granted to distribute this program, source, and documentation for non-commercial purposes, so long as the copyright notices are not removed. This program may not be distributed for profit without the permission of the author.

Comments, questions, donations, bugs due to this library may come to

Heiner Hückstädt  
Gellertstraße 12  
D-5090 Leverkusen 1

That's all, hope you can use it.

## 4. Lizenzvertrag

### 1. Urheberrecht:

Das Programm und die damit verbundenen Unterlagen unterliegen dem Urheberrecht. Das Programm bzw. die Unterlagen oder jegliche Kopien dürfen nicht verwendet, kopiert, abgeändert oder übertragen werden, sofern nicht ausdrücklich in diesem Vertrag vorgesehen.

### 2. Lizenz

Sie haben das einfache Recht, das beiliegende Programm lediglich auf einem einzelnen Computer zu verwenden. Das Programm darf materiell von einem Computer auf einen anderen übertragen werden, vorausgesetzt, daß das Programm jeweils nur auf einem Computer verwendet wird. Das Programm darf nicht elektronisch über ein Netz von einem Computer auf einen anderen übertragen werden. Es dürfen keine Kopien des Programms oder der Begleitunterlagen an andere verteilt werden. Das Programm bzw. die Unterlagen dürfen nicht geändert oder übersetzt werden.

### 3. Sicherungsunterlagen und Übertragung

Sie dürfen lediglich eine (1) Kopie des Programms für Sicherungszwecke anfertigen. Die Urheberrechtanzeigee muß reproduziert und der Sicherungskopie beigelegt werden. Das Produkt darf nur an eine andere Partei übertragen und konzessioniert werden, wenn die andere Partei den Bedingungen und Konzessionen dieses Vertrages zustimmt und einen Meldeschein ausfüllt und diesen an Harms Computer — Systeme zurücksendet. Wenn sie das Programm übertragen, müssen sie gleichzeitig die Unterlagen und die Sicherungskopie übertragen oder die Unterlagen übertragen und die Sicherungskopie vernichten (löschen).

### 4. Bedingungen

Diese Lizenz ist bis zu ihrer Beendigung wirksam. Sie können sie beenden, indem Sie das Programm und die Unterlagen und sämtliche Kopien davon vernichten. Diese Lizenz läuft ebenfalls ab, wenn Sie irgendeiner Bedingung oder Kondition dieses Vertrages nicht nachkommen. Sie erklären sich einverstanden, bei einer solchen Beendigung sämtliche Kopien des Programms und der Unterlagen zu vernichten.

### 5. Haftungsausschluß für das Programm

Das Programm wird "in gegenwärtigen Zustand" bereitgestellt ohne jegliche Garantie irgendeiner Art, sei es ausdrücklich oder stillschweigend inbegriffen, einschließlich, jedoch ohne Beschränkung darauf, der stillschweigend mit eingeschlossenen Garantien der Marktfähigkeit und Eignung für einen bestimmten Zweck. Das gesamte Risiko bezüglich der Ergebnisse und der Leistung des Programms wird von Ihnen getragen, sollte sich das Programm als schadhaft erweisen, übernehmen Sie (und nicht Harms Computertechnik) die gesamten Kosten für sämtliche erforderlichen Wartungs-, Reparatur- oder Korrekturarbeiten. Weiterhin übernimmt Harms Computertechnik keinerlei Garantie, Gewährleistung

bzw. gibt keinerlei Garantieerklärungen bezüglich der Anwendungsergebnisse des Programms hinsichtlich Richtigkeit, Genauigkeit, Zuverlässigkeit, Gültigkeit oder sonstigem; und Sie vertrauen auf das Programm und die Ergebnisse ausschließlich auf eigenes Risiko.

#### 6. Beschränkte Gewährleistung für die Diskette

Harms Computertechnik garantiert dem ursprünglichen Lizenznehmer lediglich für einen Zeitraum von neunzig (90) Tagen ab dem Datum des ursprünglichen Kaufes, daß die Diskette(n), auf der (denen) das Programm aufgenommen ist, keine Material- oder Verarbeitungsfehler aufweist (aufweisen). Falls ein durch diese Gewährleistung gedeckter Fehler während dieser Garantiezeit (90 Tagen) auftritt und Sie spätestens fünf (5) Tage nach Ablauf dieses Zeitraumes von 90 Tagen an Harms Computertechnik zurückgegeben wird, so wird Harms Computertechnik die Diskette entweder reparieren oder ersetzen. Diese Garantie steht an Stelle von sämtlichen sonstigen ausdrücklichen oder gesetzlichen Garantien, und einer jeglichen stillschweigend inbegriffenen Garantie, einschließlich, jedoch ohne Beschränkung darauf, der stillschweigend miteingeschlossenen Garantien der Marktfähigkeit und Eignung für einen bestimmten Zweck, wird hiermit auf den besagten Zeitraum von 90 Tagen eingeschränkt.

Die Haftung von Harms Computertechnik beschränkt sich ausschließlich auf die Reparatur oder auf den Ersatz des schadhafte(n) Produktes in ihrem allgemeinen Ermessen und umfaßt in keinem Falle Schadenersatz für Verwendungsverlust oder Verlust von erwarteten Gewinnen oder Vorteilen oder sonstigen Neben- oder Folgekosten, Ausgaben oder Schäden, einschließlich, ohne Beschränkung darauf, jeglicher Daten oder Informationen, eventuell verloren gehen oder falsch wiedergegeben werden, selbst wenn Harms Computertechnik von der Möglichkeit solcher Schäden in Kenntnis gesetzt worden ist.

Einige Länder lassen keine Beschränkung der Dauer einer stillschweigend miteingeschlossenen Garantie zu, so daß die obenstehende Beschränkung möglicherweise auf Sie nicht zutrifft. Einige Länder lassen keinen Ausschluß bzw. keine Beschränkung von Neben- oder Folgeschäden zu, so daß die obige Beschränkung bzw. der obige Ausschluß möglicherweise auf Sie nicht zutrifft. Diese Garantie gibt Ihnen bestimmte gesetzliche Rechte, und Sie haben möglicherweise sonstige Rechte, die von Land zu Land verschieden sind.

#### 7. Dieser Lizenzvertrag unterliegt den Gesetzen der Bundesrepublik Deutschland.

## **4.1 Hardware Haftung und Garantie**

Hanns Computertechnik garantiert dem ursprünglichen Käufer für einen Zeitraum von sechs Monaten ab dem Datum des ursprünglichen Kaufes, daß das PROFESSIONAL-030PLUS500-BOARD keine Material- oder Verarbeitungsfehler aufweist. Falls ein durch diese Gewährleistung gedeckter Fehler während dieser Garantiezeit (sechs Monate) auftritt und Sie spätestens fünf (5) Tage nach Ablauf dieses Zeitraumes von 6 Monaten an Harms Computertechnik zurückgegeben wird, so wird Harms Computertechnik das PROFESSIONAL-030PLUS500-BOARD entweder reparieren oder ersetzen.

Diese Garantie steht an Stelle von sämtlichen sonstigen ausdrücklichen oder gesetzlichen Garantien und einer jeglichen stillschweigend inbegriffenen Garantie, einschließlich, jedoch ohne Beschränkung darauf, der stillschweigend miteingeschlossenen Garantien der Marktfähigkeit und Eignung für einen bestimmten Zweck, wird hiermit auf den besagten Zeitraum von sechs Monaten eingeschränkt.

Die Haftung von Harms Computertechnik beschränkt sich ausschließlich auf die Reparatur oder auf den Ersatz des schadhafte Produktes in ihrem allgemeinen Ermessen und umfaßt in keinem Falle Schadenersatz für Verwendungsverlust oder Verlust von erwarteten Gewinnen oder Vorteilen oder sonstigen Neben- oder Folgekosten, Ausgaben oder Schäden, einschließlich, ohne Beschränkung darauf, jeglicher Daten oder Informationen, eventuell verloren gehen oder falsch wiedergegeben werden, selbst wenn Harms Computertechnik von der Möglichkeit solcher Schäden in Kenntnis gesetzt worden ist.

Einige Länder lassen keine Beschränkung der Dauer einer stillschweigend miteingeschlossenen Garantie zu, so daß die obenstehende Beschränkung möglicherweise auf Sie nicht zutrifft. Einige Länder lassen keinen Ausschluß bzw. keine Beschränkung von Neben- oder Folgeschäden zu, so daß die obige Beschränkung bzw. der obige Ausschluß möglicherweise auf Sie nicht zutrifft. Diese Garantie gibt Ihnen bestimmte gesetzliche Rechte, und Sie haben möglicherweise sonstige Rechte, die von Land zu Land verschieden sind.

Fehler, die durch unsachgemäßen Gebrauch oder falschen Einbau der Boards verursacht wurden, unterliegen nicht der Garantie.

Falls trotz unserer regelmäßigen Qualitätskontrollen Mangel an dem Produkt zu verzeichnen sind, sind diese schriftlich innerhalb 2 (zwei) Wochen bei uns anzuzeigen.

## **5 Fehlerbehandlung**

### **5.1 Allgemeine Fehler**

Fehler im direkten Zusammenhang mit dem PROFESSIONAL-030PLUS500-BOARD sind nicht bekannt.

Sollten trotzdem Fehler auftauchen sind sie meistens darauf zurückzuführen, daß das Programm spezielle Hardwareeigenschaften des MC68000 nutzt, die nicht mit dem MC68030 konform laufen. Diese Art der Programmierung wird gelegentlich bei Spielen angewandt und kann dort zu Fehlern führen. Bei Fehlern im Zusammenhang mit älterer Software kann auch die Verwendung vom neuen Betriebssystem Kickstart 2.0 zur Ursache haben. Weitere Ursachen können falsch angeforderter Speicher sein. Zum Beispiel wenn ein Programm 512 kB Chip-Memory erwartet und 1 MB zur Verfügung stehen. Diese Fehler haben nichts mit dem PROFESSIONAL-030PLUS500-BOARD zu tun, sondern sind genereller Art. Fehler können in Verbindung mit Hardware von Fremdherstellern auftreten, die sich bei der Entwicklung nicht an die Richtlinien von Commodore gehalten haben. Falls Probleme auftauchen sollten, dann entfernen sie diese Hardware und probieren nur den Amiga mit dem PROFESSIONAL-030PLUS500-BOARD. Durch einzelnes Hinzufügen der Fremdprodukte läßt sich meist das Teil identifizieren, welches den Fehler verursacht. Das Wechseln der Hardware wird im ausgeschalteten Zustand vorgenommen. Die Garantiebedingungen der einzelnen Hersteller sind dabei zu beachten.

Sogenannte Benchmarktest können unterschiedliche Ergebnisse anzeigen. Das verwendete Betriebssystem und vorher geladene Programme haben einen Einfluß auf das Ergebnis. Um Systeme miteinander vergleichen zu können müssen genau die gleichen Randbedingungen geschaffen werden.

### **5.2 Festplattencontroller**

Bei Festplattencontrollern mit DMA muß das DMA unterbunden werden. Hierzu muß in der Installationssoftware vom Controller eine MASK-Adresse eingetragen werden. Der Eintrage lautet 01FFFFFF hex. Bei einigen Controllern muß eine weitere Null vor den Eintrag eingefügt werden. Daß der Datentransfer nach diesem Eintrag langsamer sein kann liegt in der Natur der Sache. Es kann eben kein DMA mehr stattfinden.

## 6. Technische Daten

Das PROFESSIONAL-030PLUS500-BOARD wird mit folgenden Daten, je nach Ausführung, ausgeliefert:

- Prozessorboard für den 68000 Sockel
- MC68030 mit integrierter MMU
- MC68030 14.3 MHz Taktfrequenz, gesockelt
- MC68000 mit 7,19 MHz getaktet
- Prozessoren umschaltbar von MC68030 nach MC68000 (Hardwareumschaltung)
- Coprozessor MC68882 oder MC68881 (optional bis 50 MHz) einsetzbar im PLCC Gehäuse
- Automatische Coprozessorerkennung
- 1, 2, 4, 8MB 32-Bit RAM
- schnelles Memory mit 100% I — D Caching
- Separater Takt für Prozessor und Co-Prozessor möglich
- Sockel für FPU-Oszillator vorhanden
- 100% MMU Unterstützung, verschiedene Kickstartversionen ladbar
- I — D DMA-Caching im Amiga Adressraum
- Kickstart 2.0 kompatibel

Technische Änderungen vorbehalten.

# *Amiga Hardware World*

*Everything about Amiga hardware...*

~

*<http://amiga.resource.cx>*