



SRAMDISK

Silicon-Disk- Baugruppe mit
2 MB Speicherkapazität und Batteriepufferung
für den NDR-Klein-Computer.

Stand: April 2007

Copyright © by Gerald Ebert

Wichtiger Hinweis:

Die in dieser Anleitung wiedergegebenen Schaltungen und Verfahren werden ohne Rücksicht auf die Patentlage oder Lizenzrechte Dritter mitgeteilt. Sie sind ausschließlich für private Zwecke und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden. *)

Alle Schaltungen und technische Angaben in dieser Anleitung wurden vom Autor sorgfältig erarbeitet bzw. zusammengestellt. Trotzdem sind Fehler nicht auszuschließen. Daher kann der Autor weder eine Garantie noch die juristische Verantwortung oder irgend eine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

Die Rechte an Firmennamen, Logos und Warenzeichen, die in dieser Anleitung genannt werden, liegen bei den jeweiligen Inhabern.

*) Bei gewerblicher Nutzung ist vorher die Genehmigung des möglichen Lizenz- oder Rechteinhabers einzuholen.

Inhalt:

Vorwort	4
1 Kurzbeschreibung der Funktion	4
2 Technische Daten	4
3 Prinzipbeschreibung	5
3.1 Blockschaltbild SRAMDISK	5
3.2 Beschreibung des Blockschaltbildes	6
3.3 Die Batteriepufferung der SRAMs	6
3.4 Die Aktivitätsanzeige	7
4 Aufbauanleitung	7
4.1 Umgang mit IC's	7
4.2 Stückliste	8
4.3 Aufbau Schritt für Schritt	8
4.4 Belegung von Steckleisten, Buchsen und Jumpers	10
5 Literaturhinweise und –nachweise	10
Schaltplan	12
Bestückungsplan	13
Layout Bestückungsseite mit Aufdruck	14
Layout Bestückungsseite	15
Layout Lötseite	16
Programmbeispiel zur Steuerung der Baugruppe	17

Vorwort

Schon zu Beginn meiner Arbeit mit dem NDR-Klein-Computer (NKC) habe ich nach einem schnellen Speichermedium gesucht, das wie eine Festplatte nutzbar ist, aber nicht so teuer sein soll. Zu dieser Zeit nutzte ich die Möglichkeiten der ROA256/1M als schnelles Speichermedium zum Entwickeln von Programmen.

Jetzt, ein paar Jahre (genauer 18) kann ich mir diesen „Traum“ von damals erfüllen. Das Ergebnis ist die Baugruppe SRAMDISK. Eine I/O-Baugruppe die sich fast wie ein Massenspeicher (FD/HD) verhält.

Eine Platine werde ich aus Kostengründen erst mal nicht produzieren lassen. Meine gefädelte Variante tut's auch.

Da nichts auf dieser Welt perfekt sein kann, werden sich auch hier Fehler eingeschlichen haben. Wenn Sie einen entdecken, so teilen Sie mir dies mit einer möglichst genauen Beschreibung im NKC-Forum (siehe Literatur) mit. Danke.

1 Kurzbeschreibung der Funktion

Der zentrale Teil der Baugruppe sind die SRAMs mit einer maximalen Speicherkapazität von 2 MB. Eine Adressierungslogik, die den Aufbau einer Festplatte simuliert, steuert den Zugriff auf den Speicher. Auf den NKC-Bus sind nur noch ein Daten- und zwei Sektorregister sichtbar. Die bewährte Batteriepufferung von der ROA256/1M (leicht modifiziert) sorgt dafür, daß auch nach dem Ausschalten Ihres NKC die gespeicherten Informationen in den Speicherbausteinen der Baugruppe nicht verloren gehen.

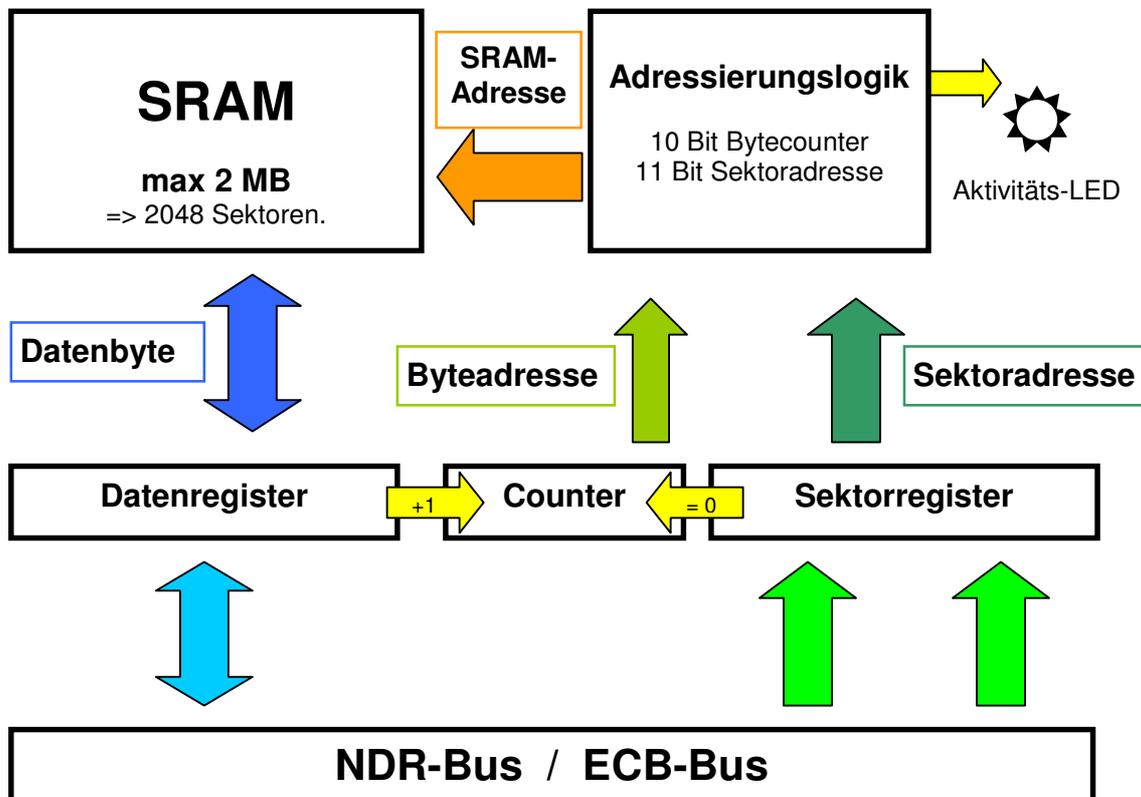
2 Technische Daten

- Europakarte 160 x 100 mm doppelseitig mit Bestückungsaufdruck
- NDR-Bus und ECB-Bus
- Stromaufnahme ca. 350 mA (Bestückung mit LS-Typen)
ca. 2µA im Power-Down-Mode
- Batteriepufferung der Speicherbausteine; SRAMs
- Aktivitätsanzeige durch LED

3 Prinzipbeschreibung

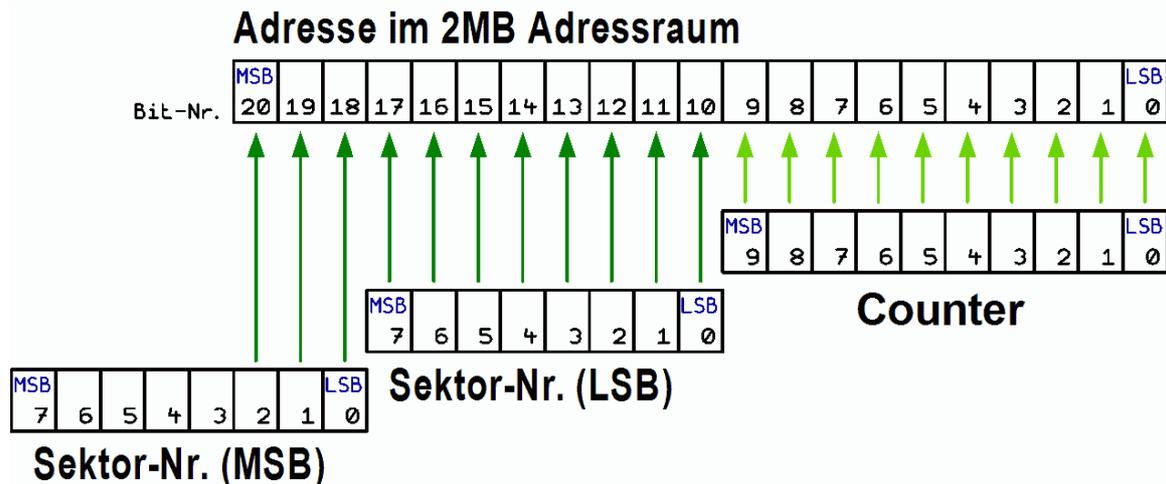
Nach dem Schreiben der Sektornummer in die Sektorregister werden die Daten in bzw. von SRAMDISK Sektorweise (1024 Bytes beim NKC) gelesen bzw. geschrieben. Nach jedem Zugriff auf das Datenregister zählt ein Counter die Byteadresse innerhalb eines Sektors um 1 hoch, sodaß beim nächsten Zugriff auf das nachfolgende Byte zugegriffen werden kann. Ein Schreibzugriff auf eines der Sektorregister setzt diesen Zähler wieder auf 0 zurück. Durch diesen Mechanismus erreicht die Baugruppe eine sehr hohe Datenübertragungsrate, die nur noch von Speicher zu Speicher Transfer übertroffen wird. Auf meinem NKC (CPU68000/12MHz) erreiche ich Übertragungsraten bis 950 kB/s. Das ist mehr als 30 mal schneller als ein Diskettenlaufwerk.

3.1 Blockschaltbild SRAMDISK



3.2 Beschreibung des Blockschaltbildes

Die maximale Speicherkapazität der Baugruppe ist 2 MB, wo bei der Speicher nicht komplett bestückt werden muß. Die Adressierung der 2 MB wird in der Adreßlogik in zwei Stufen berechnet. Diese simuliert die Aufteilung des Speichers von Disketten und Festplatten in Sektoren und Bytes. Der Counter erledigt dabei die Byteadressierung innerhalb eines Sektors und die Sektorregister bilden die Sektornummer. Folgende Graphik zeigt, wie sich die Adresse im 2MB Adreßraum der SRAMs zusammensetzt.



Im Gegensatz zur Sektornummer kann der Counter nicht direkt gesetzt werden. Er läßt sich nur durch Schreiben auf eines der Sektorregister auf 0 zurücksetzen und durch Lesen/Schreiben auf das Datenregister um 1 hochzählen. Dadurch kann ein kontinuierlicher Datenstrom der Länge eines Sektors (1024 Bytes) vom Datenregister gelesen werden. Dieser Mechanismus ermöglicht erst die hohe Übertragungsgeschwindigkeit der Baugruppe.

3.3 Die Batteriepufferung der SRAMs

Die Simulation einer FD/HD erfordert eine Batteriepufferung der SRAMs. Die Schaltung habe ich weitgehend von der Baugruppe ROA256/1M übernommen. Daß diese Schaltung wirklich gut ist, zeigte sich als ich meinen NKC nach fast 18 Jahren vom Dachboden geholt habe. Die Information in den gepufferten SRAMs war immer noch komplett erhalten. Ich habe nur noch ein paar kleine Anpassungen auf meine Baugruppe gemacht, ansonsten ist meine Pufferschaltung identisch zur ROA256/1M.

Die Schaltung selbst ist etwas aufwendig, da das CS-Signal der SRAMs im Power-Down-Mode auf Highpegel gehalten werden und daher im Pufferzustand aus der Batterie gespeist werden muß. Deshalb ist es notwendig die Steuerung dieses Signals durch die Adressierungslogik über sogenannte Open-Collector-Treiber (IC11 - LS07) laufen zulassen. Würde man das Signal an den Dekoder (IC10 – LS139) anschließen, würde dieser die Batterie in ein paar Stunden leersaugen. Die Dioden sorgen einerseits dafür, daß die SRAMs stets mit der Versorgung höchster Spannung gespeist werden, als auch für die Entkoppelung der Pufferbatterie von der Stromversorgung. Die Transistoren dienen zusammen mit dem Spannungswächter (IC15 TL7705) zum Entkoppeln der Stromversorgung des Open-Collector-Treiber und zum Überbrücken der Diode, über die die Stromversorgung an die SRAMs geleitet wird, um den durch sie verursachten Spannungsabfall (0,7V) nicht so groß werden zulassen. Diese Überbrückung ist bei der Diode über die die Pufferspannung geleitet wird nicht notwendig, da die verbrauchsbestimmende Größe im Pufferbetrieb der Strom ist, den dann die SRAMs „ziehen“. Die Batteriespannung von 3,6V liegt immer an und die SRAMs geben sich auch mit 3 V zufrieden ohne mehr Strom zu brauchen. Den Strom im Pufferbetrieb bei Vollbestückung mit den von mir eingesetzten Bausteinen konnte ich nicht messen. Er muß unter 5µA liegen. Laut Datenblatt sind es 0,45µA pro SRAM.

3.4 Die Aktivitätsanzeige

Die Aktivitätsanzeige zeigt den Zugriff auf das Datenregister durch eine LED an. Sie ist reine Spielerei und hat keinen Einfluß auf die Funktion der Baugruppe. Einfacher gesagt: man kann sie einfach weglassen.

Die Funktionsweise ist sehr einfach. Das ENABLE-Signal des Datenregisters ist mit dem INT0-Pin des ATtiny13 verbunden. Wechselt dieses Signal von High- auf Lowpegel (Falling Edge) wird im ATtiny13 ein Interrupt ausgelöst, der einen Timer initialisiert und aktiviert sowie die LED einschaltet. Ist der Timer abgelaufen, dann schaltet er die LED wieder aus. Schaltungstechnisch ist das einfacher als ein retrIGGERbares Monoflop und braucht auch weniger Platz auf der Platine.

Die Fusebits des ATtiny13 müssen nicht programmiert werden. Das Programm ist für den Auslieferungszustand parametrisiert.

4 Aufbauanleitung

4.1 Umgang mit IC's

CMOS-Bausteine sind hochempfindlich gegen elektrostatische Aufladung! Bewahren oder Transportieren Sie CMOS-Bausteine nur auf leitenden Schaumstoff! Alle Pins müssen kurzgeschlossen sein.

Achten Sie darauf, daß Sie Verbindung mit einer Erdungsmöglichkeit haben, bevor Sie mit diesen Bausteinen arbeiten. Geeignete ESD-Artikel gibt es im Fachhandel.

4.2 Stückliste

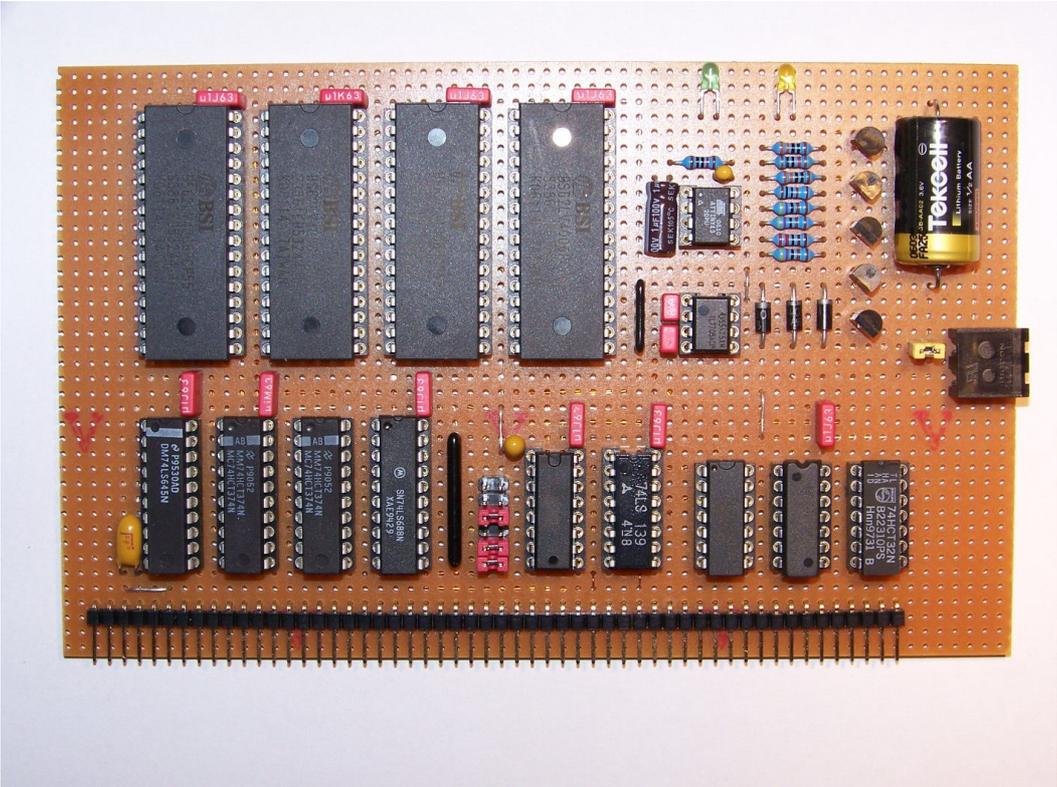
Anzahl	Kennung	Bauteil (klassisch)	Bauteil (alternativ)
1			Platine
4	IC 1 - 4		SRAM 628512
1	IC 5	74LS245	74HCT245
2	IC 6, 7	74LS374	74HCT374
1	IC 8	74LS688	74HCT688
1	IC 9		74HCT4040
1	IC 10	74LS139	74HCT139
1	IC 11		74LS07
1	IC 12	74LS00	74HCT00
1	IC 13	74LS32	74HCT32
1	IC 14		TL7705A
1	IC 15		ATtiny13
2	T 1, 4		Transistor BC327
3	T 2, 3, 5		Transistor BC337
1	C 1		Kondensator 10µF Tantal
1	C 18		Kondensator 4,7 µF Tantal
1	C 14		Kondensator 1 µF Elko
1	C 17		Kondensator 1 µF Tantal
12	C 2 – 13		Kondensator 100nF MKS-2
2	C 15, 16		Kondensator 100 nF MKS-02
2	R 9, 10		Widerstand 330 Ω
1	R 8		Widerstand 1 kΩ
1	R 4		Widerstand 2,2 kΩ
3	R 5 – 7		Widerstand 4,7 kΩ
2	R 1, 3		Widerstand 10 kΩ
1	RN 1		Widerstandsnetzwerk 8 x 4,7 kΩ
1	RN 2		Widerstandsnetzwerk 4 x 3,3 kΩ
3	D 1 - 3		Siliziumdiode 1N4001
2	D 4, 5		LED, standard
1	BAT 1		Lithium-Batterie 3,6 V, ½ AA
1	ST 1	Messerleiste DIN 41612 Bauform C 64 pol	für ECB-Bus
1	ST 2	Stiftleiste 1 x 36 pol & 1 x 18 pol	gew. für NDR-Bus
1	ST 3	Stiftleiste 2 x 6 pol	
1	ST 4	Stiftleiste 1 x 2 pol	

4.3 Aufbau Schritt für Schritt

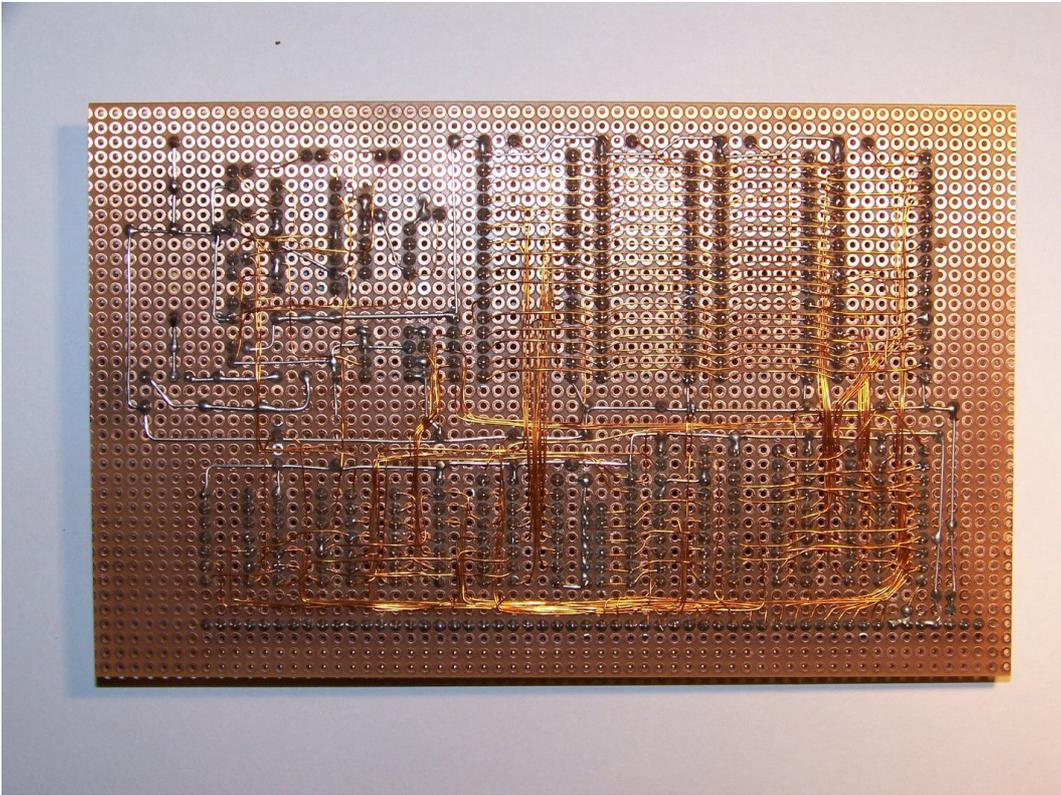
Entfällt

Eine Platine werde ich aus Kostengründen vorerst nicht produzieren lassen. Deshalb spare ich mir eine detaillierte Anleitung.

Ich habe ich mir eine Prototypenplatine möglichst layoutnah gefädelt. Als Anregung zum Nachbau die Bilder meiner Baugruppe:



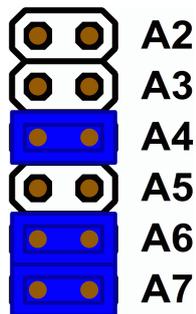
Prototypenplatine gefädelt, Bestückungsseite



Prototypenplatine gefädelt, Lötseite

4.4 Belegung von Steckleisten, Buchsen und Jumpfern

ADDR:



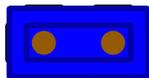
Hier ist die Jumperstellung der Standard-I/O-Adresse für die SRAMDISK-Baugruppe (\$2C-\$2F) dargestellt. Grundsätzlich kann der I/O-Adressbereich nach den eigenen Wünschen eingestellt werden. Denken Sie aber daran, daß das Grundprogramm dann die Baugruppe nicht mehr unterstützen kann.

Ein gesetzter Jumper bedeutet ein 0-Bit in der Adreßangabe.

ST4:



Die Pufferspannung der SRAMs ist unterbrochen.
Die SRAMs verlieren ihre Dateninformation.



Die SRAMs werden von der Batterie auf der Platine gepuffert.
Die Dateninformation in den SRAMs bleibt auch nach dem Ausschalten des NKC erhalten.

5 Literaturhinweise und –nachweise

Bücher

- ❖ Rolf-Dieter Klein
„Rechner modular“
Der NDR-Klein-Computer – selbstgebaut und programmiert
Franzis-Verlag, München. ISBN 3-7723-8721-7
- ❖ Rolf-Dieter Klein
“Die Prozessoren 68000 und 68008“
Rechnerarchitektur und Sprache im NDR-Klein-Computer
Franzis-Verlag, München. ISBN 3-7723-7651-7

Datenblätter

- * SRAM: BS62LV4006, Rev. 1.1 vom Jan/2004
- * Philips 74HCT4040, vom Dez/1990
- * TL7705A, vom März/1997
- * Atmel ATtiny13, Rev. 2535g-AVR-01/07
- * ROA256/1M Aufbauanleitung

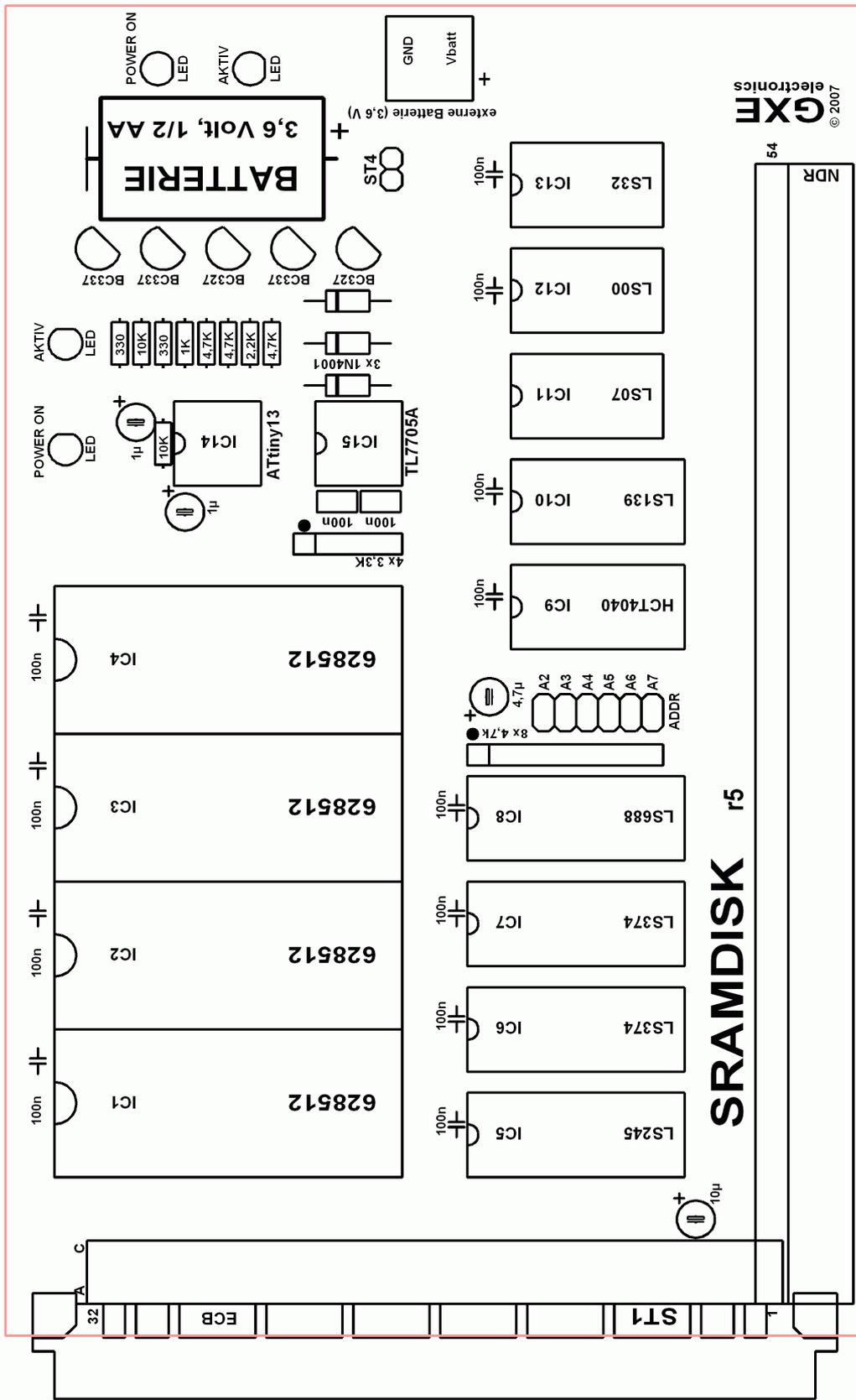
Quellen im Internet

<http://www.atmel.com/avr/>
<http://sourceforge.net/projects/winavr/>
<http://www.nongnu.org/avr-libc/>

NKC-Forum

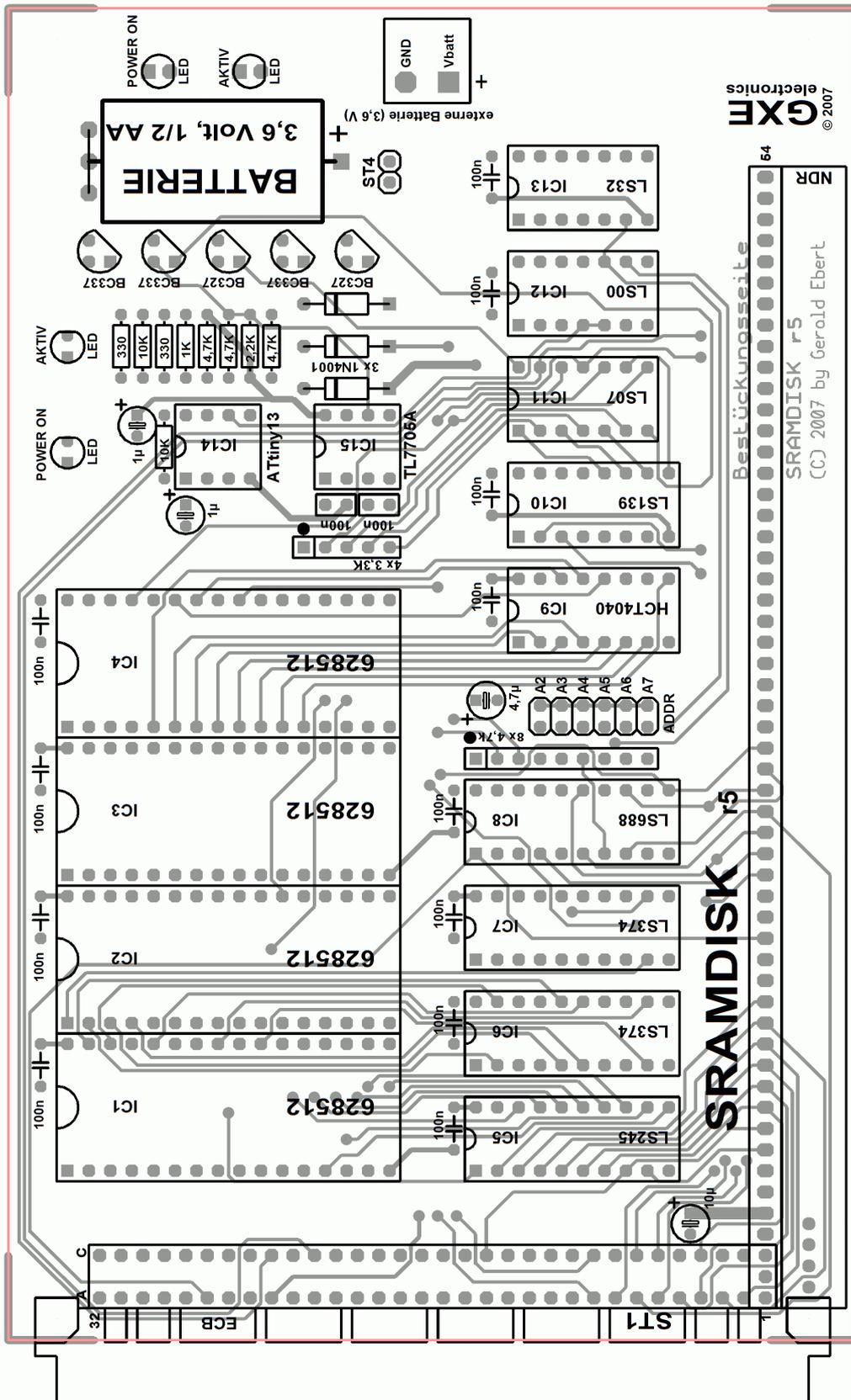
<http://www.dr crazy.de/forum/>

Bestückungsplan

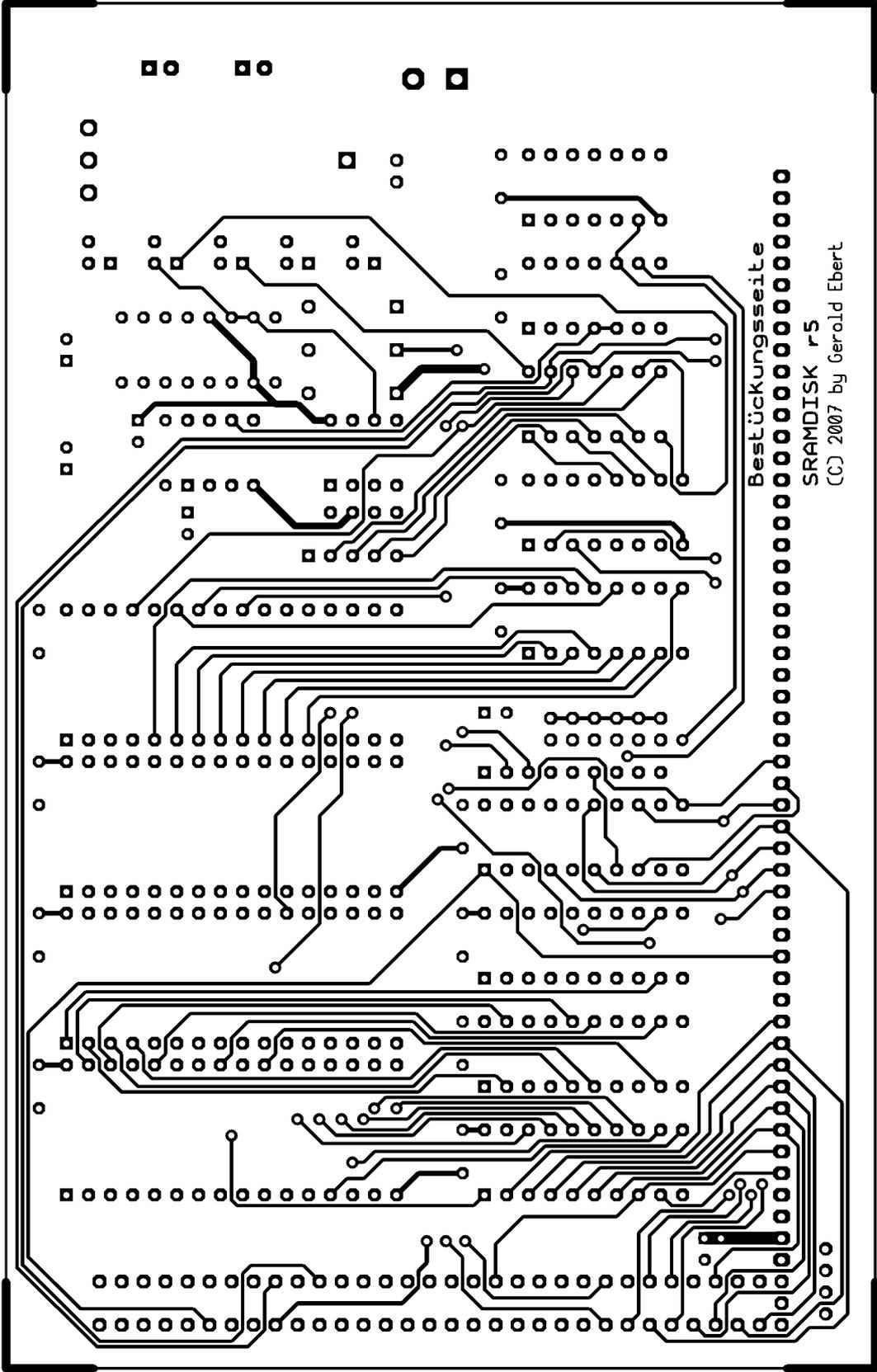


GXE electronics © 2007

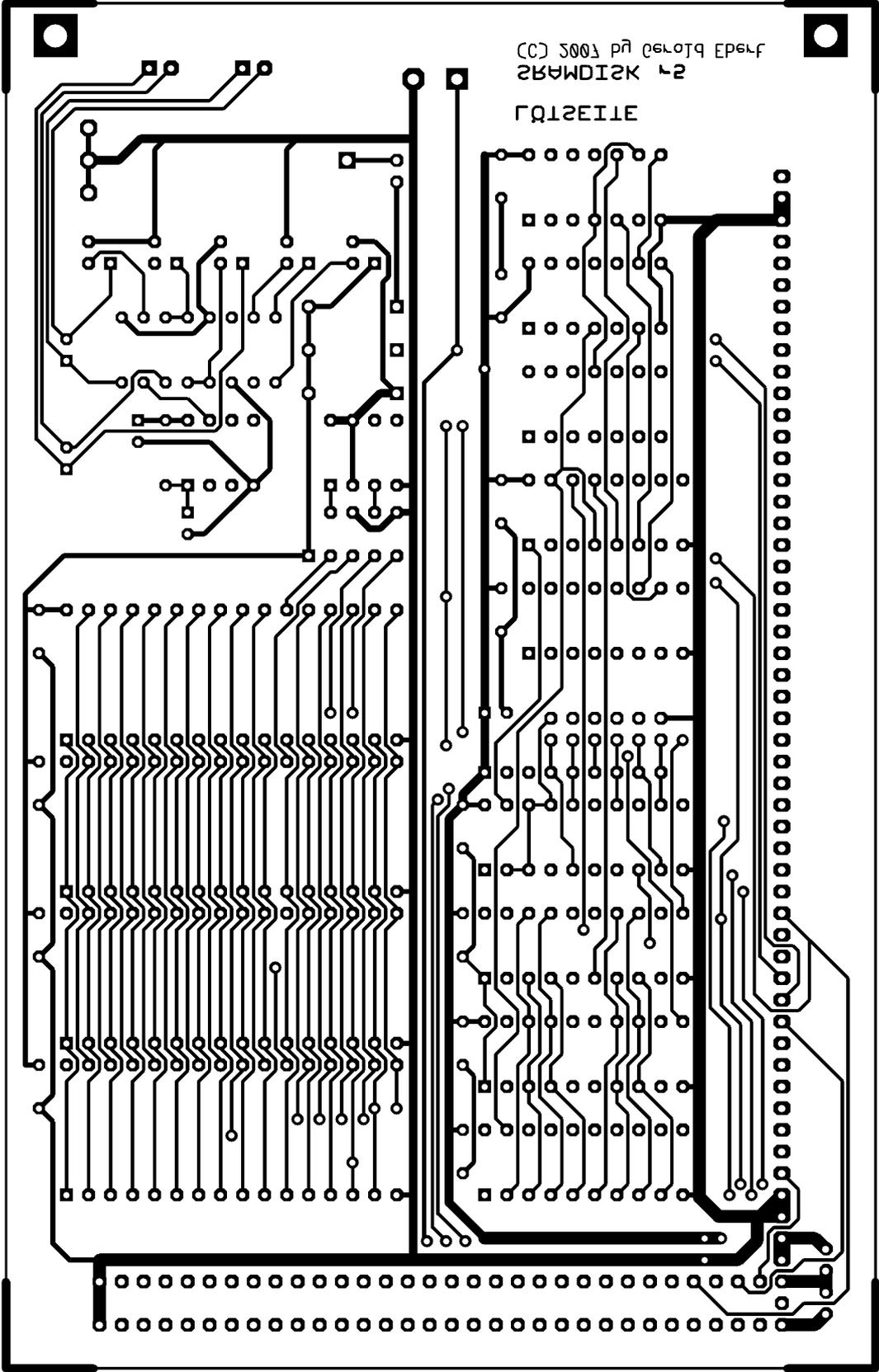
Layout Bestückungsseite mit Aufdruck



Layout Bestückungsseite



Layout Lötseite



Programmbeispiel zur Steuerung der Baugruppe

```
*****
***!                                     !***
***! .....SRAMDISK Testprogramm.....!***
***!                                     !***
***! (P)+(C) 2007  Gerald Ebert         !***
***!                                     !***
*****
```

*>>>>>> LABELS

```
cpu      equ      2          * M68000/10
sdbase   equ      $ffffff2c  * Basis I/O-Adresse ($2C-$2F)
sddata   equ      (sdbase+0)*cpu * Daten Register
sdsectl  equ      (s2base+1)*cpu * Sektor Register Low Byte
sdsecth  equ      (s2base+2)*cpu * Sektor Register High Byte
sdsectsz equ      1024      * Sektorgroesse in Bytes
```

*>>>>>> START DES TESTPROGRAMMS

```
      clr      d0
      lea     $20000, a0
loop1: movea.l  a0, a1
      move.b  d0, (a1)+
      addq   #1, d0
      cmp    #sdsectsz, d0
      blo.s  loop1
      clr    d0
      bsr   sdwrite
      lea  $21000, a0
      bsr  sdread
      movea.l a0, a1
      lea  $20000, a0
loop2: move    #(sdsectsz/4)-1, d0 * Sektorlaenge (eigentlich 4-Bytes)
      move.l (a0)+, d1
      cmp.l  (a1)+, d1
      bne.s  error
      dbra  d0, loop2
      lea  oktext(pc), a0
      bra.s  ausgabe
error: lea  failtext(pc), a0
ausgabe: move.b (a0)+, d0
      beq.s  ende
      moveq  #!CO, d7
      trap  #1
      bra.s  ausgabe
ende:  rts
      * Erzeuge 4 Zahlenreihen 0...255
      * Sektornummer 0 laden
      * Sektor in SRAMDISK schreiben
      * Gleichen Sektor an andere Stelle lesen
      * vergleiche beide Puffer
      * Fehler: Puffer nicht gleich
      * bis alle Bytes verglichen sind
      * OK: Puffer sind gleich
      * Gib entsprechenden Text aus
      * Zeichen in d0 laden
      * Letztes Zeichen -> Fertig.
      * Zeichen in d0 ausgeben
      * nächstes Zeichen
      * ENDE
```

```
oktext: dc.b      'Sektor in SRAMDISK korrekt geschrieben.', 13, 10, 0
failtext: dc.b     'Fehler in SRAMDISK Daten.', 13, 10, 0
```

```
*>>>>>> LESE SEKTOR VON SRAMDISK
; '
; '
; '
d0 => Sektornummer
a0 => Pufferadresse (Ziel)
```

```
sdread: movem.l  d0/a0/a1, -(a7)
      lea  sddata.w, a1
      move.b d0, sdsectl.w
      lsr  #8, d0
      move.b d0, sdsecth.w
      moveq #(sdsectsz/8)-1, d0
sdrdloop: move.b (a1), (a0)+
      * Lade Datenregisteradresse
      * Setze Sektornummer Low Byte
      * Setze Sektornummer High Byte
      * Anzahl Bloেকে fuer Zaehler
      * Übertrage ein Byte in den Puffer
      * Block gelesen
```



```

        move.b    d0, sdsecth.w
        lea     sdbuf1(pc), a0
        move.b    (a0)+, (a1)
        move.b    (a0)+, (a1)
        move.b    (a0)+, (a1)
        move.b    (a0)+, (a1)
        move.l    sdbuf2(pc), d0
        cmp.l    sdkenn(pc), d0
        beq.s    sdictst1
        andi    #$fe, ccr
        rts
sdictst1: ori    #$01, ccr
        rts

sdkenn:  dc.l    $55AA33CC
sdbuf1:  ds.l    1
sdbuf2:  ds.l    1

        end

```

* Setze Bytezaehler zurück
 * Stelle Inhalt des Sektor wieder her

 * Vergleiche Kennung nach Schreiben
 * und Lesen aus Sektor
 * -> Sektor vorhanden
 * Sektor nicht vorhanden, C = 1

 * Sektor vorhanden, C = 0

 * Kennung fuer Pruefung
 * Puffer zur Sicherung des Sektorinhalts
 * Puffer zum Lesen der Kennung aus Sektor