

## 2MB-RAM-Erweiterung für Prof 180x

Helmut Bernhardt

Zur Erweiterung des Speichers über 512K auf dem CPU-Board des Prof 180x hinaus hat Conitec die Turbo-RAM vorgesehen. Das 'Turbo' in dieser Bezeichnung ist aber mit Vorsicht zu genießen. Da diese Speichererweiterung auf dem ECB-Bus steckt, liegen zwischen CPU und diesem Speicher 2 Datentreiber. Bei 9,2 MHz ist die Turbo-RAM nur mit einem Wait zu betreiben, was dann auch beim Zugriff auf die ursprünglichen 512K enorm bremst. Hier sei eine Alternative vorgeführt, die ohne Waits eine Erweiterung um 2MB in Form eines Huckepack-Boards auf dem CPU-Board bringt.

Um zu erklären, wie das machbar ist, soll zunächst auf die Verhältnisse eingegangen werden, wie im Prof der Zugriff auf verschiedene Speicher geregelt ist. Da dies alles hauptsächlich im PAL 14L8 gesteuert wird, sei das disassemblierte PAL-Listing hier wiedergegeben:

PAL14L8

ausgelesen mit c't PAL-Brenner, 1988 Be  
Grundversion nach Conitec

```
me ioe rd lir busak a18 e mm0 mm1 adr a5 gnd
a1 a7 a19 rom cas0 cas1 fdc gate ioswl dir a19 vcc
```

```
/dir      = /ioe * /rd * /adr * az * /a5           ;Richtungssteuerung
          + /me /rd /a19                          ;ECB-Bus-Daten
          + rd * lir * busak                       ;treiber
/iosel    = /ioe * lir * /a18 * /adr * a7 * /a5 * /a2
/fdc      = /ioe * lir * /a18 * /adr * a7 * /a5 * a2
gate      = mm0 * /mm1                            ;Freigabe ECB-Bus-
          + mm1                                    ;-treiber
/a19      = /mm1                                  ;Freigabe für
          + a18 * /mm0                             ;externen Speicher
          + /a18 * mm0
/rom      = /me * /a18 * /rd * /mm0 * /mm1
/cas0     = /me * /a18 * e * rd * /mm0 * /mm1
          + /me * /a18 * e * mm0
/cas1     = /me * a18 * e * /mm0
          + /me * a18 * e * mm0 * /mm1
```

'**adr**' ist ein Eingangssignal für das PAL, in dem A3, A4, und A6 entsprechend den zu erzeugenden I/O-Freigabesignalen im Bereich D8H-DFH decodiert sind. '**e**' ist ein spezielles Taktsignal des HD64180, das hier zum Timing der /cas-Signale verwendet wird. **mm0** und **mm1** sind Ausgänge eines 74LS259-Latches mit einzeln setzbaren Bits. Die übrigen Eingangssignale des Pals sind die normalen Signale der CPU.

Die I/O-mapped Freigabesignale **/iosel** und **/fdc** sind hier uninteressant. Für die Steuerung des ECB-Businterface und der memory-mapped Ressourcen werden im PAL die Signale **a19**, **/gate**, **/rom**, **/cas0** und **/cas1** erzeugt. /rom, /cas0 und /cas1 sind low aktive Freigabesignale für das ROM und für je einen 256K-Block RAM auf dem CPU-Board. /gate, das an die Enable-Pins der 74LS245-Bustreiber gelegt ist, steuert die grundsätzliche Möglichkeit der CPU, auf den ECB-Bus zugreifen zu können. Dieses Signal muß high sein, wenn die Prof 180x-Karte im Slave-Modus für sich allein werkeln soll und eine andere CPU-Karte den Bus kontrollieren soll.

**/dir** steuert die Richtung, in der der 74ALS245-Datentreiber die Daten vom/zum ECB-Bus treibt; /dir = 0 bedeutet, daß die Daten vom Prof 180 zum ECB-Bus getrieben werden. In die

Richtungsteuerung geht nicht nur ein, ob gelesen oder geschrieben wird (/rd) sondern auch, ob die adressierte Baugruppe auf dem CPU-Board oder 'draußen' auf dem Bus liegt. Wenn eine Baugruppe auf dem CPU-Board gelesen wird, muß der Bustreiber zum Bus hin treiben, um auf dem internen Datenbus keinen Datenkonflikt zu erzeugen. Außerdem wird auch das Signal **/busak** ausgewertet, da bei Zugriff eines externen DMA-Controllers auf Baugruppen des CPU-Bords die Treiberrichtung in Bezug auf /rd umgekehrt sein muß.

**a19** ist ein high aktives Freigabesignal, das auf den ECB-Bus gelegt ist und bei Conitec u.a. für die Steuerung der Turbo-RAM benutzt wird.

Die Analyse des disassemblierten PAL-Listings ergibt für die Steuerung der Freigaben von ROM und internem und externem RAM die Beziehungen:

Steuersignale			freigegebene Baugruppe	Pegel v. /gate	Pegel v. A19
mm1	mm0	A18			
0	0	0	ROM bei /rd; RAM bei /wr	1	0
0	0	1	RAM Bank 1	1	0
0	1	0	RAM Bank 0	0	0
0	1	1	RAM Bank 1	0	0
1	0	0	externes RAM	0	1
1	0	1	RAM Bank 1	0	0
1	1	0	RAM Bank 0	0	0
1	1	1	externes RAM	0	1

Wenn mm0 und mm1 beide low sind, ist /gate high und es kann nicht auf Speicher auf dem ECB-Bus zugegriffen werden. Durch Umschalten auf mm0=1 wird das beim Lesen in die unteren 256K eingeblendete ROM abgeschaltet. Mit mm1=1 wird der grundsätzliche Zugriff auf externe Speicher ermöglicht. Dann wird mit mm0 gesteuert, ob der externe Speicher in die unteren oder die oberen 256K des HD64180-Adreßraums eingeblendet wird.

Für das Bereitstellen des bei Conitec 'externen' Erweiterungsspeichers nun innerhalb des CPU-Boards muß die Bussteuerung nun etwas anders gestaltet werden. Außerdem müssen Freigabesignale für zwei weitere RAM-Banks mit je 1 MB geschaffen werden. Dafür muß das PAL zunächst entrümpelt werden. Auf das Signal /gate kann man verzichten, wenn man die CPU-Karte nur als Master fahren will. Die Pins 19 der 74ALS245- Bustreiber werden einfach an Masse gelegt.

Da kein externer Speicher zu steuern ist, wird auch das Signal a19 nicht mehr benötigt. Es sind dann zwei Ausgänge und ein Eingang am PAL freigeworden. Die beiden Ausgänge sollen die Freigabesignale für die beiden zusätzlichen 1MB-RAM-Blocks ergeben und der Eingang wird für die Selektion eines dieser beiden Blocks mit einem Signal Q2 beaufschlagt. Dann bekäme ein neues PAL den Inhalt:

PAL20L8  
 08.08.90 H. Bernhardt  
 Erweiterung des Prof 180x auf 2,5 MB RAM

```
me ioe rd lir busak a18 e mm0 mm1 adr a5 gnd
a2 a7 cas2 rom cas0 cas1 fdc cas3 iosel dir q2 vcc

/dir      = /ioe * /rd * /adr * a7 * /a5
          + /me * /rd
          + rd * lir * busak
/iosel    = /ioe * lir */a18 * /adr * a7 * /a5 * /a2
/fdc      = /ioe * lir */a18 * /adr * a7 * /a5 * a2
/rom      = /me * /a18 * /mm0 * /mm1 * /rd
/cas0     = /me * /a18 * e * /mm0 * /mm1 * rd
```

```

      * /me * /a18 * e * mm0
/cas1 = /me * a18 * e * /mm0
      + /me * a18 * e * mm0 * /mm1
/cas2 = /me * /a18 * e * /mm0 * mm1 * /q2
      + /me * a18 * e * mm0 * mm1 * /q2
/cas3 = /me * /a18 * e * /mm0 * mm1 * q2
      + /me * a18 * e * mm0 * mm1 * q2
    
```

Das Steuersignal q2 und auch noch zwei Pseudoadressen zur Selektion eines von 4 Blocks mit 256K innerhalb eines 1MB-Blocks im Erweiterungsspeicher werden mit einem zusätzlich bereitgestellten Latch 74LS174 erzeugt. Dieses Latch wird durch einen Out-Befehl an Port D7H beschrieben. D0 und D1 geben darin die Pegel der Pseudoadressen a18 und a19 und D2 den Pegel von q2 vor.

Die gesamte Erweiterungsschaltung incl. RAMs kommt auf einem Huckepackboard unter, das in die Sockel der Adreßmultiplexer Z17 und Z18 (74HC158) sowie Z11 (74HC51) und des Datenlatch Z15 (74LS374) der Druckerschnittstelle gesteckt wird und diese ICs mit aufnimmt. Damit stehen die gemultiplexten Adressen, die Daten, die Steuersignale /RAS, MUX und /MUX auf dem Huckepackboard zur Verfügung. Die Pseudo- Adressen a18' und a19' werden mit einem zusätzlichen 74HC51 Gatter gemultiplext.

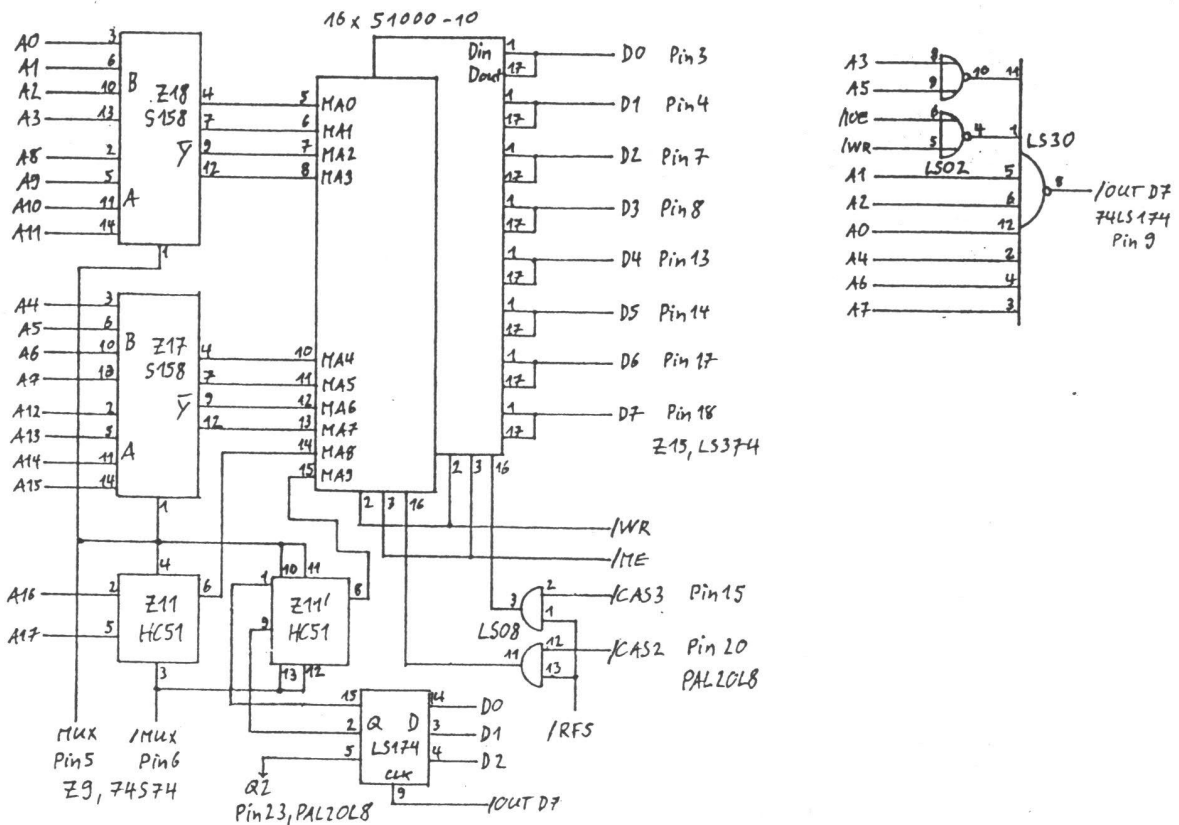
Es müssen dann nur noch folgende Signale durch freie Verdrahtung zwischen Huckepackbord und CPU-Bord verbunden werden:

/ioe (74LS02, Pin6)	von Pin5 des Z7, 74ALS245
/wr (RAMs, Pin2)	von Pin6 des Z7, 74ALS245
/me (RAMs, Pin3)	von Pin2 des Z7, 74ALS245
/rfs (74LS08, Pins 1,13)	von HD64180, Pin57
q2 (74LS174, Pin5)	an Pin23 des PAL
/cas3 (74LS08, Pin12)	von Pin8 des PAL
/cas2 (74LS08, Pin1)	von Pin13 des PAL

Und wofür ist so viel Speicher gut? Natürlich wird der erstmal als RAM-Floppy genutzt. Anstatt diesen Speicher zusammen mit den restlichen 320K, die im Grundspeicher schon als RAM-Floppy genutzt werden, als eine RAM-Floppy zu betreiben, soll hier eine zweite RAM-Floppy von 2MB Kapazität eingerichtet werden. Das vereinfacht die Software etwas, wenn nicht ständig geprüft werden muss, ob ein zu übertragender Sektor nun auf der Erweiterung oder im Grundspeicher zu holen ist. Der benötigte Treiber läßt sich aus dem für die 'alten' RAM-Floppy bestehenden Treiber ohne große Änderungen umformen. Es kommt beim Login nur hinzu, daß mm1 auf high gesetzt werden muß.

Bei den Lese- und Schreib-Routinen ist noch nachzurüsten, daß der in (@trk) übergebene Wert aufzuteilen ist in D0 und D1, die zusammen mit einem gesetzten D2 das Adreßzusatzbyte des DMAC ergeben, und daß D2-D4 nach zweimaligem Rechtsshiften als D0-D2 an den Port D7H ausgegeben werden. In der Login-Routine der 'alten' RAM-Floppy sollte vorsichtshalber grundsätzlich mm1 low gesetzt werden.

### **Abb.1: Schaltung**



Folgende Datenstrukturen sind für das BDOS vorzugeben:

```

cseg                ;Drive Table
@dtbl dw fdsd0,fdsd1,fdsd2,fdsd3 ; 4 Floppy Drives
      dw fdram                ; 'alte' RAM-Floppy
      dw f2dram              ; 2MB-RAM-Floppy
      dw hdsk0,hdsk1,hdsk2,hdsk3 ; 4 Harddisk-Partitionen

dseg                ;dph's koennen in Bank 0 liegen
      ; extended disk parameter header 2MB-RAM-Floppy
      dw f2write
      dw f2read
      dw f2login
      dw f2init
      dw 0,0
fd2parm dph 0,dpbr2

cseg                ;disk parameter block fuer 2MB-RAM-Floppy
dpbr2 dpb 1024,64,2048,4096,512,0,8000
      ;1024 Bytes pro Sektor, 64 Sektoren pro Track
      ;2048 Tracks, Blockgroesse 4k, 512 DIR-Entries
      ;keine Systemspuren, unremovable media
    
```

Da diese 2MB (wie das sonst meistens bei RAM-Floppies für 8Bit-Computer aussieht) nicht I/O-mapped sind, kann man darin auch Programme laufen lassen. Es liegt dann allerdings am Können des Programmierers, mit dem BDOS ein Abkommen zu treffen, daß dieses nach dem Abarbeiten einer BDOS-Funktion wieder ordnungsgemäß die Bank und auch noch die entsprechende 256K-Portion einschaltet, in der das Programm den BDOS-Call abgesetzt hat.

Dieses Problem sei an die Soft-Cracks weitergegeben.

**Abb.2: Bestückungs- und Anschlußplan: J1 bleibt bei 1MB-Cips offen.**

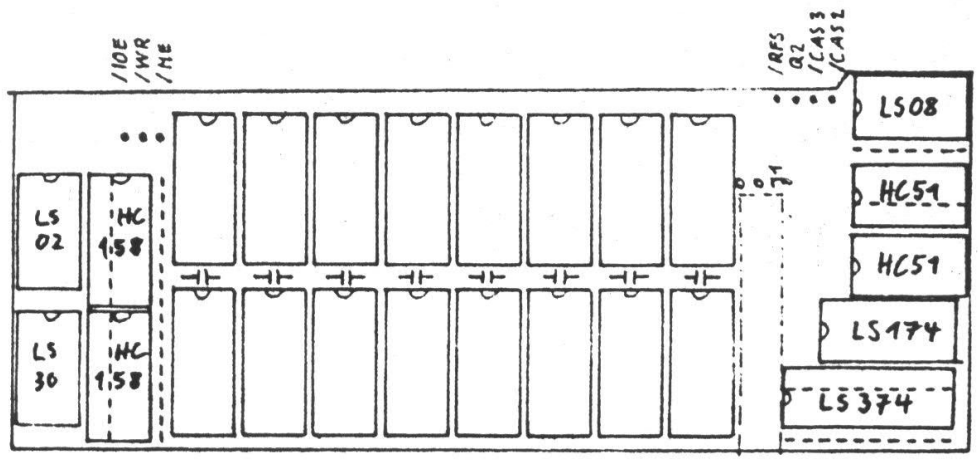


Abb.3: Layout der Bestückungsseite der 2MB-Erweiterung

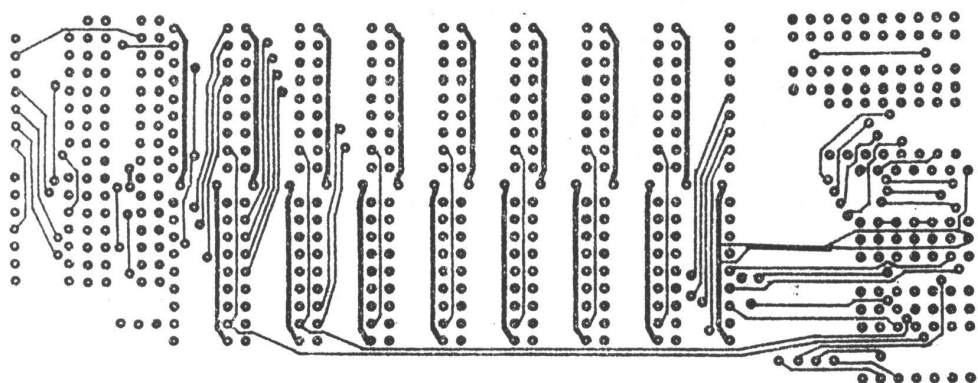


Abb.4: Layout der Lötseite der 2MB-Erweiterung

