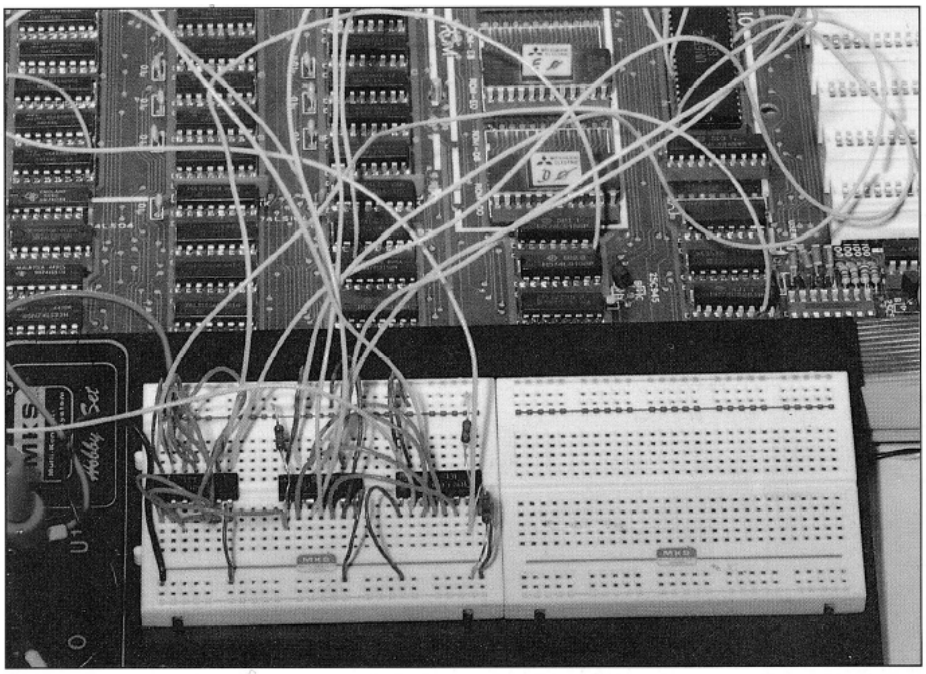


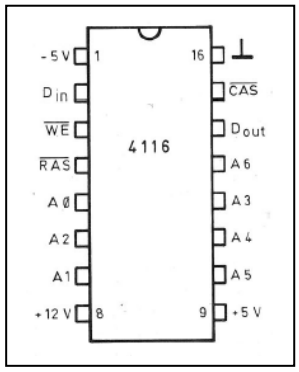
Multiplying Memory fourtimes

192 KByte RAM in the Apple II +
from Bernd Montag

In the age of computers with advanced memory the good old Apple does not impress anybody with 48 Kbyte.
Esspecially when the graphicpeges are used the available memory to program itself reduces dramatically.
But with a little handcraft the mainboard can be populated with 192 Kbyte of RAM. With that memory also under CP/M six HiRespages are available that don't get in conflict with the TPA. It´s just also possible to implement CP/M 3.0. Of course the Apple remains still the former one and all Apple-program will run at the modified Computer.
The Picture above shows the first experimental assembly.
Several Apple owners look with some regret to the new computers that are supplied with large amounts of memory.

At the Apple one of the annoying things is the fact that the two graphicpages are located in the middle of the TPA due to the fixed addressing and use each 12 Kbyte. It is possible to use graphic-routines in Turbopacal and CP/M (issue c´t 2/85) but they can´t be compiled due to emerging space of memory.
The Apple II+ with 48 Kbyte of RAM on the Mainboard is normally populated with three rows of each 8 time 4116 chips. Thay can be replaced rather easy with 4164 chips with fourtimes capacity. The pinouts are nearly same as the 4116 chips except that the 5 Volt must be fixed to the former 12 Volt supply and the adressport **A7** must receive the proper Data. Due to the fact that the 4164 is also available with the ability to gain with 128 refresh-adresses (NEC, Hitachi, Mitsubishi, Fujitsu, …) the refresh must not be altered. In case that at some chips the addresslines **A0** to **A6** are not in the same order this is without importance to the modification. In case that your Apple is a clone and is populated with4164 instead –

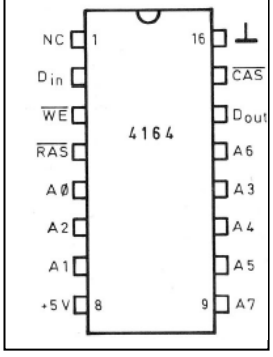
it is possible to apply the modification with 41256 chips. This chips must be supplied with a kind of reproduced 8 Bit refresh if the computer or the videologic does not provide it.
Additional Logic-chips
As result of the modification the user recieves instead of one bank now four banks of 48 KByte. The selection of the bank is executed by 3 simple TTL-Logic-chips that enable the use the different banks for programs and graphic bank now four banks of 48 KByte. The selection of the bank is executed by 3 simple TTL-Logic- bank now four banks of 48 KByte.

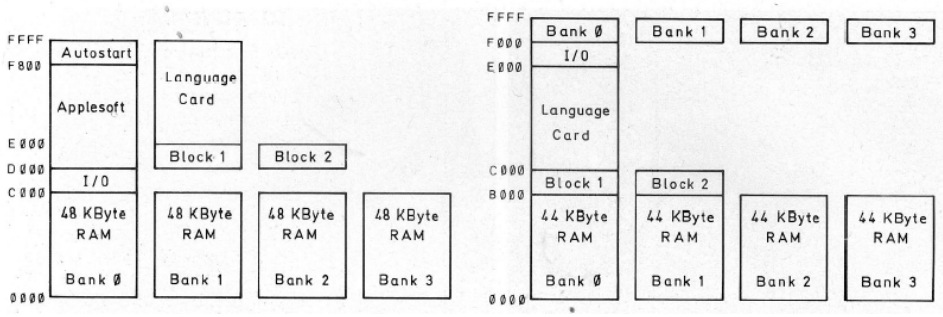


The selection of the bank is executed by 3 simple TTL-Logic-chips that enable the use the different banks for programs and graphic. For example within CP/M there is a complete block of 56 Kbyte available to the TPA and additional six HiRes-pages.
For the addressing of the four banks a 4-Bit port is required that might be within the area of **C060** and **C06F**.
Within that requires area there is the input of the casstteport and the push-buttons and analog-input-ports but they are both two-times behind each other From **C068** upwards all inputs are repeated so that for example the push-button 2 is available at **C062** as well as at **C06A**.

A logic with 4 **NOR**-gates generates from an access to the area from **C068** to **C06F** a loadpulse for a 4-bit-counter that picks up theinformation from the data-lines **D0** to **D3**.The counter is only used as a memory for the number of the CPU-bank or the graphic-bank. The mentioned inputs are now only available at the address-area from **C060** to **C067** – but this causes no conflict. In case of doubt the logic moight be switched off for compatibility purposes.
two accesses
the counter-outputs are

The pinout of the 4164 is nearly identical with the 4116 used in the Apple





Upside left picture the address-space as the CPU sees it from DOS and at the right picture the address-space varies from that with Dos under the control of CPM.

connected to the input of a 4 to 1 multiplexingchip that in fact just generates the signal for the 8th adresslines of the memorychips and its controlled by two select-signals. **A_x** switches from row-adresses to column-adresses and **P0** enables to makes difference between access of the CPU or the Video-access. The CPU only accesses the memory at the second half of the cycle the access of the Video is performed in the first half of the cycle. The **Φ0** is during the first half of the cycle low and in the second half of the cycle is high and you can determine between both kinds of access and use the differ-rent banks of memory. The data-bits D0 and D1 select the bank for the program and the data-bits D2 and D3 determine the bank from which the display-data is taken from. But which banks are selected at the startup of the computer? This problem is solved by the counter. That counter is also reseted by the same line as the 6502 CPU at the clear-input the counter is thereforset to zero. So at startup automatically the bank 0 is selected for CPU- and Video-access.

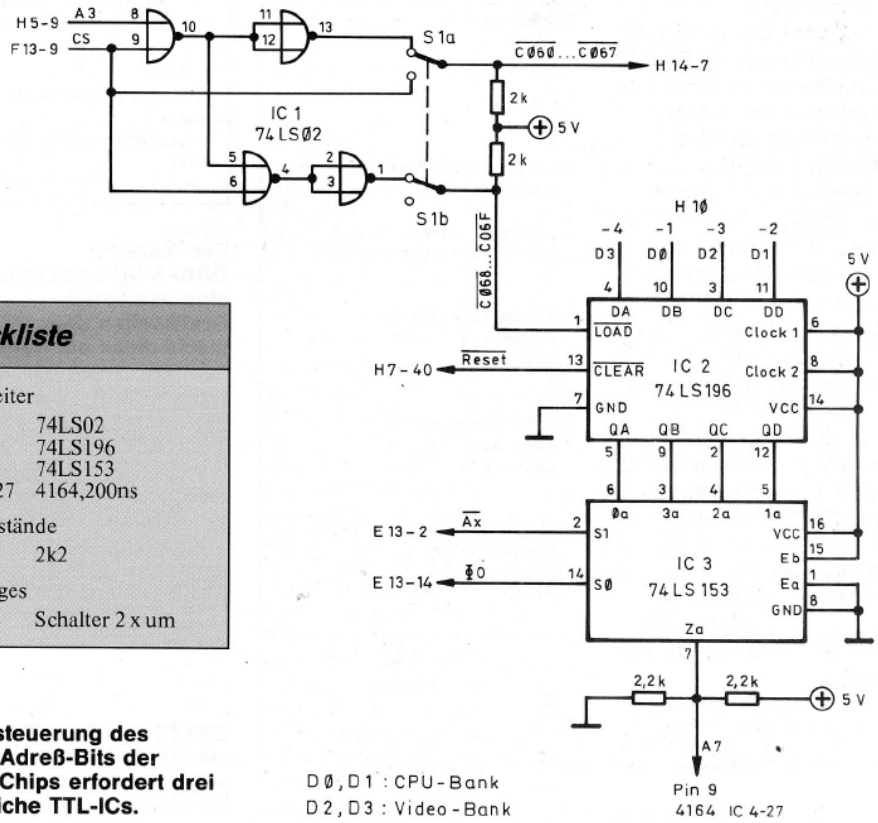
The Mod

Most of the cloneboards have spare-area where the three additional chips for the banking-logic can be soldered at. Otherwise the Logic might also be setup at a experimental-PCB. In single steps splitted the modification is carried out as follows:

1. mounting of the logic circuitary.
2. removal of all 4116 chips from the motherboard.
3. removal of all capacitors related to the 5 Volt supply in the RAM area.
4. Cut apart of the -5 Volt, +12 Volt and +5 Volt supply-lines for the RAM-chips / sockets.
5. Pin 1 and Pin 8 of the RAM-sockets are connected to + 5 Volt. In case that the former -5 Volt lines had been filte-red with tantal-capaci-tors, there should be replaced by ceramic-capacitors of 100 nF.
6. Pin 9 of the sockets is to be soldered to the additional logic.
7. The logic should be inserted between **pin 9** of the 74LS138 at position **F13** and the **Pin 7** of the 74LS251 an the position **H14**.
8. At the additional logic the data-lines **D0** to **D3** and the address-line **A3** and the supplyvoltage shall be connected.
9. the voltages at the RAM-sockets are to be control-led with up-powered computer.
10. After the computer is powered down again the 4164 chips are to be inserted into the sockets.
11. the 4116 chip which is located at the language-card and connected to the Mainboard by the flat-ribboncable is to be re-placed with a 4164 chip. Make sure, that none of the supply-voltages of the language-card is connected to the mainboard anymore !

Turbo-Banking

If the mofification has been performed without mistakes the computer behaves after next power-up like a normal 48 kByte Apple II. With write-accesses to the address C068 the banks may be switched. Under CP/Mit is possible with the TPA of 56 KByte (0000 to DFFF) to work with two parallel 48 Kbyte-banks (0000 to BFFF). So it is also possible to implement CP/M 3.0. The Turbo-Pascal-programs show up with the first use of the memory-banks with a 56 Kbyte large TPA. The access to the switched graphic-pages is not optimized at the moment far fast access but at the moment at least the usage of the banks can be demonstrated. The Routines SByte and RByte are written in Z80-coding. They write to a spacified bank or read from a specified bank a specific Byte.



Stückliste

Halbleiter

IC1	74LS02
IC2	74LS196
IC3	74LS153
IC4 - 27	4164,200ns

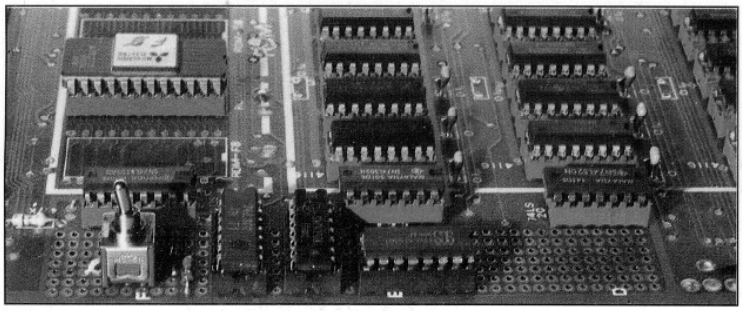
Widerstände

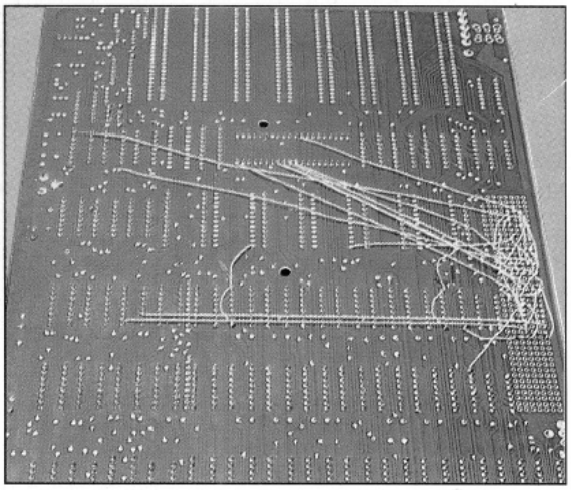
R1-4	2k2

Sonstiges

S1	Schalter 2 x um

Die Ansteuerung des achten Adreß-Bits der 4164er-Chips erfordert drei zusätzliche TTL-ICs.





Die Zusatzlogik läßt sich bequem auf dem freien Rasterfeld der Hauptplatine unterbringen. Die Verdrahtung versteckt sich unter dem Motherboard.

```
const endadr=$C900;
{ im Option-Menue : COM-File, END-Adr:=$C8FF }
{ endadr = default-Adresse - Codelaenge - 1 }

{ nur fuer APPLE II/II+ mit 208 KByte. }
{  (C) Bernd Montag 3/1986             }


procedure SBYTE (seite,adresse,wert:integer);
    external $C900;

function RBYTE (seite,adresse:integer):integer;
    external $C912;

procedure INSTALLIERE;
type code = array [1..36] of byte;
const
  anzahl=36;
  prg : code =
  ($C1,          { pop  b      = PROCEDURE SByte }
   $D1,          { pop  d      wert }
   $E1,          { pop  h      adresse }
   $7B,          { mov  a,e    wert->a }
   $D1,          { pop  d      seite }
   $C5,          { push b      }
   $44,          { mov  b,h    h->b }
   $4D,          { mov  c,l    l->c }
   $21,$68,$E0,  { lxi  h,e068 }
   $73,          { mov  m,e    selektiere seite }
   $02,          { stax b      speicher wert }
   $7B,          { mov  a,e    }
   $E6,$0C,      { ani  0c     loesche bit0 & bit1 }
   $77,          { mov  m,a    selektiere arbeitsseite }
   $C9,          { ret         }

   $E1,          { pop  h      = FUNCTION RByte }
   $D1,          { pop  d      adresse }
   $C1,          { pop  b      seite }
   $E5,          { push h      }
   $21,$68,$E0,  { lxi  h,e068 }
   $71,          { mov  m,c    selektiere seite }
   $1A,          { ldax d      lade wert }
   $5F,          { mov  e,a    wert->e }
   $79,          { mov  a,c    }
   $E6,$0C,      { ani  0c     }
   $77,          { mov  m,a    selektiere arbeitsseite }
   $6B,          { mov  l,e    ergebnis lsb }
   $26,$00,      { mov  h,00   ergebnis msb }
   $C9           { ret         } );

var i:integer;

begin
  for i:=1 to anzahl do
    mem [endadr+I-1] := prg [I];
    { Schiebt Routinen in den      }
    { Common-Bereich $B000-$DFFF = Language-Card }
end;
```

```
var _page, _wpage, : integer;

procedure GRAFMODE (Seite, code:integer);{ Seite = 1..6 }
var offset, basis, i:integer;
begin
   Mem[$E052] := 0;  { nur Grafik }
   Mem[$E057] := 0;  { Hi-Res-"   }
   _wpage :=((Seite+1) div 2); _page:=_wpage*4;
   basis := _page+_wpage;
   Mem[$E068] := _page;
   offset:=(Seite+1) mod 2;
   if offset=0 then Mem[$E054] := 0   { 1. Seite }
               else Mem[$E055] := 0;  { 2. Seite }
   Mem[$E050] := 0;
   if code>0 then
      for i:=4096+offset*8192 to 12287+offset*8192 do
          sbyte( basis, i, 0 );  { loescht Grafikseite }
end;
procedure TEXTMODE;
begin
    Mem[$E054] := 0;  { 1. Seite   }
    Mem[$E051] := 0;  { Textmode   }
    Mem[$E068] := 0;  { aus Bank 0 }
end;
procedure GRAFINIT;
begin
  grafmode(1,0); textmode;
end;
```

The procedure Grafmode activates one of the six graphicpages and deletes it on demand

They are relocatable within the common-area (C000 to DFFF).
all Parameters are passed over in the common-area. After the access the bank **0** is to be selected again, because there the TPA is located.
Because the routines are located in the common-area the should be located between the turbo-loader and the memory reserved for variables. The final-limit-adress of the memory allocated to the variables must therefore be reduced by the size of the routines (some average 40 Bytes) .

SByte and RByte pickup first the return-adress for jumping back and then its parameters from stack

Within the Pascal-program the Procedure “Installiere” must be called before the first access to the pascal-program, which relocates the both machine-code-procedures to their final storage place.

c´t