

Speicher vierfach

192 KByte RAM im Apple II +

Bernd Montag

Im Zeitalter der Speicherriesen beeindruckt der gute alte Apple mit seinen 48 KByte natürlich niemanden. Vor allem beim Einsatz der Grafik schmilzt der verfügbare Arbeitsspeicher schnell dahin. Dabei lassen sich auf der Hauptplatine mit minimalem Aufwand und etwas handwerklichem Geschick 192 KByte RAM installieren. Dadurch stehen selbst unter CP/M sechs HiRes-Bildschirme zur Verfügung, die sich nicht mit der TPA überlagern. Sogar die Implementation von CP/M 3.0 ist möglich. Selbstverständlich bleibt der Apple sonst ganz der alte, und alle Programme laufen auch auf dem modifizierten Rechner. Das Foto zeigt den Versuchsaufbau.

Mancher Apple-Besitzer wird schon wehmütig auf modernere Rechner geblickt haben, die geradezu verschwenderisch mit Speicherplatz ausgestattet sind. Beim Apple stört besonders die ungünstige Adreßlage der beiden Grafikseiten, die bei CP/M mitten in der TPA liegen und jeweils 12 KByte beanspruchen. Unter CP/M sind Grafik-Routinen in Turbo-Pascal zwar machbar, wie das Turbo-Graf-Paket zeigt (c't 2/85), können aber aus Platzmangel nicht im Speicher kompiliert werden.

Der Apple II+ mit 48 KByte RAM auf der Hauptplatine ist in den meisten Fällen mit 3 mal 8 Speicher-Chips vom Typ 4116 ausgestattet. Diese lassen sich relativ leicht durch Chips vom Typ 4164 mit der vierfachen Speicherkapazität ersetzen (Stückpreis 3 DM). Ihre Pinbelegung ist weitgehend identisch mit den 4116er-Bausteinen. Nur die 5-Volt-Versorgung muß auf die ehemalige 12-Volt-Versorgung gelegt werden, und die Adreßleitung A7 ist mit geeigneten Informationen zu versorgen. Da der 64-KBit-Chip 4164 auch in Versionen erhältlich ist, die mit 128 Refresh-Adressen

auskommen (NEC, Hitachi, Mitsubishi, Fujitsu, ...), ist auch der Datenerhalt weiterhin gewährleistet. Sollten bei einigen Speicher-Chips die Adreßanschlüsse A0 bis A6 anders angeordnet sein, so ist das ohne Bedeutung.

Falls Ihr Apple mit acht RAM-Bausteinen vom Typ 4164 bestückt ist, wäre der Speicherausbau durch den Einsatz der 41256er-Chips möglich. Diese Bausteine benötigen allerdings einen 8-Bit-Refresh, der nachgebildet werden müßte, falls der Rechner, beziehungsweise die Video-Logik, diesen nicht zur Verfügung stellt.

Zusatzlogik

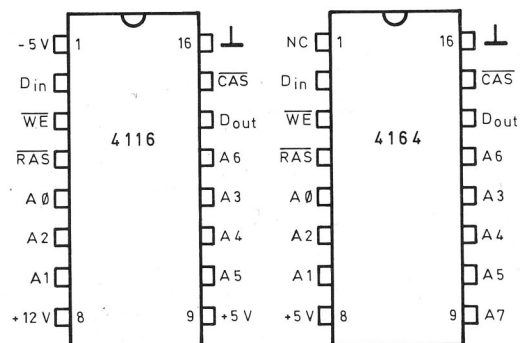
Als Ergebnis erhält man jedenfalls statt einer nun vier Speicherbänke zu je 48 KByte. Die Auswahl einer Bank geschieht mit einer kleinen Zusatzlogik bestehend aus drei TTL-ICs, die es auch ermöglicht, für Programme und für Grafik verschiedene Bänke zu benutzen. So stehen zum Beispiel unter CP/M eine durchgehende TPA von 56 KByte und gleichzeitig sechs HiRes-Schirme zur Verfügung.

Zur Auswahl der Bänke wird ein 4-Bit-Port benötigt, der sich noch im Bereich von C060 bis C06F unterbringen läßt. Dort liegen zwar schon der Kassetteneingang sowie die Push-Bottom- und Analog-Eingänge, aber zweimal hintereinander. Ab C068 wiederholen sich alle Eingänge, so daß Push-Bottom 2 sowohl unter C062 als auch unter C06A eingelesen werden kann. Eine Logik, bestehend aus vier NOR-Gattern, erzeugt beim Zugriff auf den Bereich von C068 bis C06F einen Load-Impuls für einen 4-Bit-Zähler, der daraufhin die an den Datenleitungen D0 bis D3 liegende Byte-Hälfte übernimmt. Der Zähler hat hier nur die Funktion eines Speichers für die Nummer der CPU-Bank und der Grafik-Bank.

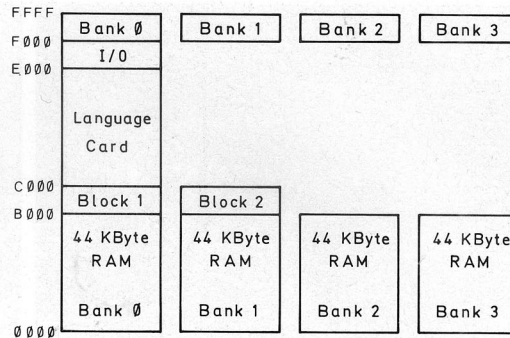
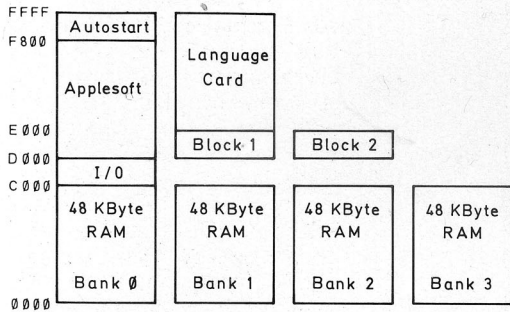
Die erwähnten Eingänge des Apple sind jetzt nur noch über die Adressen C060 bis C067 erreichbar, was aber nicht störend ist. Im Zweifelsfall kann die Banking-Logik auch ausgeschaltet werden, womit die alte Adreßaufteilung wiederhergestellt ist.

Zwei Zugriffe

Die Zählerausgänge liegen an einem 4-zu-1-Multiplexer, der letztlich das Signal für die achte Adreßleitung der Speicher-Chips liefert. Er wird mit zwei Selektsignalen versorgt.



Die Pin-Belegung des 4164 ist mit der des im Apple verwendeten 4116 fast identisch.



auf der Language-Karte ebenfalls durch einen 4164-Chip ersetzen. Sicherstellen, daß auch auf der Language-Karte keine Verbindung mehr zu den ursprünglichen Versorgungsspannungen besteht.

Turbo-Banking

Ist der Umbau fehlerfrei ausgeführt, verhält sich der Apple wie ein normaler 48-KByte-Rechner. Durch Schreibzugriffe auf die Adresse C068 können die Bänke umgeschaltet werden. Unter CP/M wäre es möglich bei einer TPA von 56 KByte (0000-DFFF) mit zwei parallel liegenden 48-KByte-Bänken (0000-BFFF) zu arbeiten. Es könnte also auch CP/M 3.0 implementiert werden.

Die Turbo-Pascal-Programme zeigen eine erste Nutzung der Speicherbänke unter CP/M mit einer 56 KByte großen TPA. Die Zugriffe auf die parallel liegenden Grafikbereiche sind zwar nicht zeitoptimiert, aber es geht ja auch nur ums Prinzip.

Die Routinen SByte und RByte sind in Z80-Code geschrieben. Sie schreiben in eine beziehungsweise lesen aus einer spezifizierten Bank an eine vorgegebene Adresse ein bestimmtes

Die Adreßaufteilung, wie sie die 6502-CPU unter DOS sieht (links). Unter CP/M stellt sich die Speicherstruktur des Apple etwas anders dar (rechts).

Ax schaltet von der Row- auf die Column-Adresse um, und P0 ermöglicht die Unterscheidung zwischen Prozessor- und Video-Zugriff. Die CPU greift grundsätzlich nur während der zweiten Hälfte des Buszyklus auf den Speicher zu, der Video-Zugriff erfolgt in der ersten Hälfte. Da $\Phi 0$ während der ersten Zyklushälfte 'low' und während der zweiten 'high' ist, kann man mit diesem Signal zwischen Video- und Prozessor-Zugriffen unterscheiden und für Programm und Grafik unterschiedliche Speicherbänke benutzen.

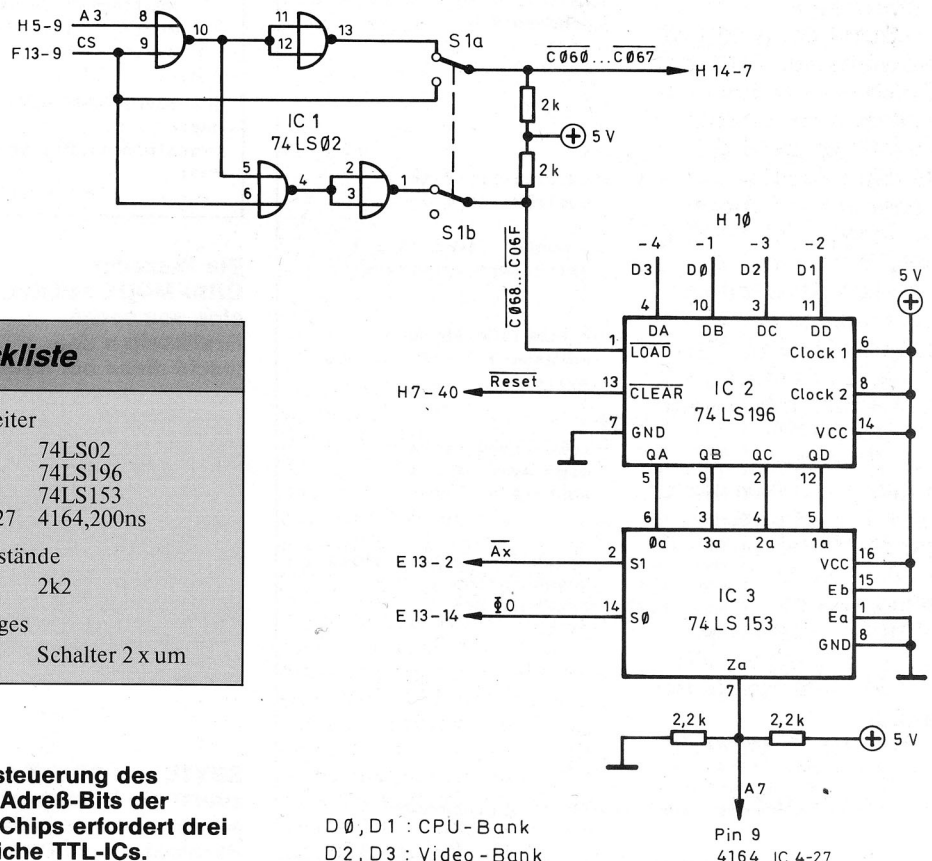
Die Datenbits D0 und D1 selektieren die Bank für das Programm, und die Datenbits D2 und D3 bestimmen die Bank, aus der die Daten für den Bildaufbau entnommen werden. Aber welche Bänke sind nach dem Einschalten des Rechners selektiert? Dieses Problem löst ebenfalls der Zähler. Er wird von derselben Reset-Leitung, die auch die 6502-CPU mit einem Reset-Impuls versorgt, über den Clear-Eingang auf Null zurückgesetzt. Nach dem Einschalten ist damit automatisch Bank 0 für Prozessor- und Video-Zugriff angewählt.

Ein Eingriff

Bei vielen Nachbauten gibt es freie Lochrasterfelder auf der Hauptplatine, auf der die drei ICs der Banking-Logik untergebracht werden können. Genau so gut lassen sich die ICs auch auf einer eigenen Lochrasterplatine unterbringen. Im einzelnen erfolgt der Umbau in folgenden

Arbeitsschritten:

1. Aufbau der Zusatzschaltung.
2. Entfernen aller 4116-Chips von der Hauptplatine.
3. Entfernen aller Glättungskondensatoren für die +5 Volt innerhalb des auf der Platine markierten RAM-Feldes.
4. Unterbrechen der -5/+12/+5-Volt-Versorgungsleitungen zu den RAMs.
5. Pin 1 und Pin 8 der RAM-Sockel mit +5 Volt verbinden. Werden Tantalkondensatoren für die Glättung der -5-Volt-Versorgung verwendet, so sollten diese durch keramische (100 nF) ersetzt werden.
6. Pin 9 der RAM-Sockel mit der Zusatzlogik verbinden.
7. In die Verbindung zwischen dem 74LS138, Pin 9 auf Position F13 und dem 74LS251, Pin 7 auf Position H14 die Zusatzlogik einschleifen.
8. Zusatzlogik mit den Datenleitungen D0 bis D3, der Adreßleitung A3 und der Stromversorgung verbinden.
9. Bei eingeschaltetem Rechner die Spannungen an den RAM-Sockeln kontrollieren.
10. Bei ausgeschaltetem Rechner die 4164er einsetzen.
11. Den über ein Flachbandkabel angeschlossenen 4116er

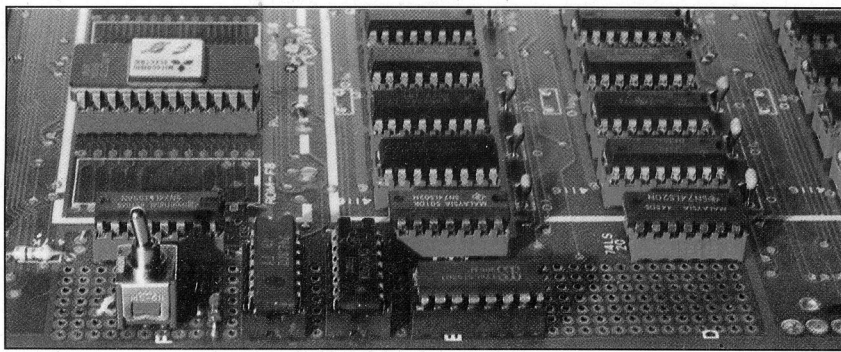


Stückliste

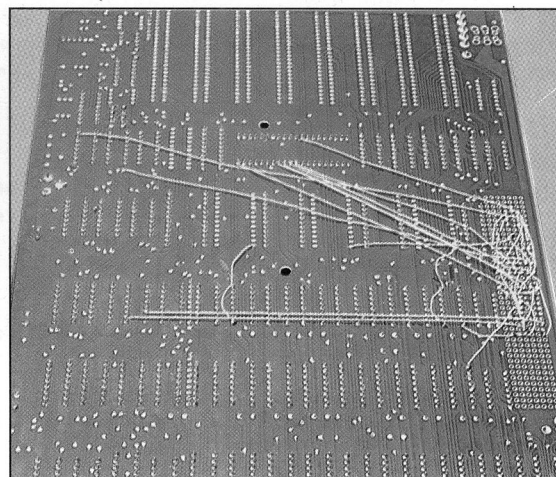
| Halbleiter | |
|-------------|-----------------|
| IC1 | 74LS02 |
| IC2 | 74LS196 |
| IC3 | 74LS153 |
| IC4-27 | 4164, 200ns |
| Widerstände | |
| R1-4 | 2k2 |
| Sonstiges | |
| S1 | Schalter 2 x um |

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

D0, D1 : CPU-Bank
D2, D3 : Video-Bank



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;

```

Die Prozedur **GRAFMODE** aktiviert eine von sechs Grafikseiten und löscht diese bei Bedarf.

SBYTE und **RBYTE** holen zuerst die Rücksprungadresse und dann ihre Parameter vom Stack.

Byte. Sie sind innerhalb des Common-Bereichs (C000-DFFF) beliebig verschiebbar. Alle Parameter werden über den Stack übergeben. Nach dem Zugriff wird wieder Bank 0 selektiert, da diese die TPA enthält.

Da die Routinen im Common-Bereich stehen müssen, werden sie zwischen den Turbo-Lader und den Variablenpeicher kopiert. Die Endadresse des Variablenbereichs ist deshalb im Optionen-Menü um die Länge der Routinen zu reduzieren (etwa 40 Byte). Im Pascal-Programm muß vor dem ersten Aufruf unbedingt die Prozedur 'Installiere' aufgerufen werden, die die beiden Maschinenprogramme an ihren endgültigen Platz kopiert.

