Paul Rojas, Frank Darius, Jürgen Drepper

68008 CPU card for the APPLE - II

The following project is again evidence for the flexibility of the Apple-II, that is resulting from its concept of providing slots. A card with the 68008 CPU that is internal compatible with the 68000 CPU. The card operates similar to the common Z80 CPU cards but provides internal full 32-bit structure of the 68008.

One of the most popular processor-families is the 68000 among the 16/32 bit CPUs. The success of the product from Motorola is the internal architecture that provides a very comfortable set of commands and twelve kinds of addressing the memory. The advanced 68010 can address up to 16 megabyte by using a virtualisation technic and supports this with a very intelligent prefetch-queue, that itself speeds up small loops within the programs. The 68000 provides internal 8 data- and 8 adressregisters with 32 bit size allthough the databus itself only consists with 16 lines.
So it just was a question of time until a true 32bit version entered the market when in June 1984 the 68020 was introduced in the USA and it was compatible to the 68000 and the 68010. Motorola offers a chip the is able to execute 2,5 MIPS(million instructions per second) and that can address up to 4 Gigabyte of logic memory.

From the 68000 CPU there is also a "small brother" available with an external Databus with 8 lines - the 68008. By that the cpu is compatible to other 8-bit CPUs and can thereby access their memory and peripherials. So there is a good path to integrate the 68008 in old computers.
By this attempt the user gets the chance to use a cpu from the 68000-family without the need to buy separate data-memory and separate peripherials. This is exactly the target of this project and the circuitary fullfills the goals by making a card for the Apple slot.
The 6502 cpu of the Apple-II is switched off and the rest of the computer is ruled by the 68008 cpu similar to the handling by the Z80 cpu-cards.
The control can be given back to the 6502 by software to control the monitor - routines to display for example the results of the program to the user at the display. Both cpu´s can share the same **RAM** and interchange with operation.

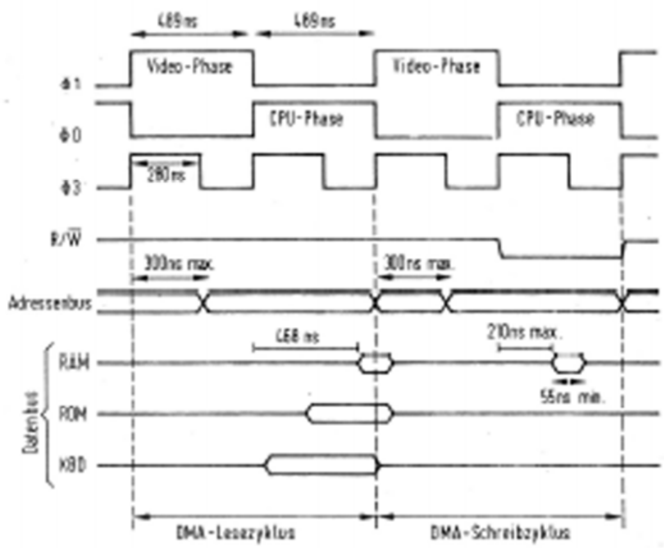

Bild 1. Das Timing der Schreib- und Lesezugriffe beim Apple-II-Plus. Vorsicht: Beim Apple-IIe sieht es etwas anders aus!

the picture displays the timing of the read- and write-cycles at the Apple II Plus - **but carefull**: at the Apple IIe the timing slightly shifted !

The Hardware Problems

The Apple provides the user with eight slots on the mainboard that permit the attachment of additional CPU-cards. Besides the databus and the adressbus also several timing and control-lines are provided that can be used by the inserted cards. For example the **line 22** in the slot is the **DMA**-line that controls the Direct Memory Access. If this line is pulled down to low (i.e. **0** Volt) the address- and datebuslines are triggered to the tristate-status and the 6502 is stopped. Alternating CPUs or peripherials are then enabled to access the address- and datalines and are permitted to perform read - or write-cycles to the logic memory (**RAM**, **ROM** or **IO** addresses).
For the clueless handover to the alternating cpu the accesses of the 6502 must be completed before and for the handover of the control to the 68008 the following tasks must be executed:

1) **DMA** must be set during **Φ1** (videophase) to low.
2) 30 ns thereafter the adressbus of the 68008 is permitted to become active.

For the return of the control to the 6502 the following must be performed:

1) The address and databuffers must be set back to tristate-status.
2) the DMA-line must be dragged up to high within 178ns after the positive rise of the **Φ1** in oder to give the 6502 the possibility to set his own adressbus within proper time (see picture 1).

It must be kept sure that only one of both cpu´s has access to the memory or the peripherials and that the correct protocol of the **DMA** is respected.

One of the difficulties results from the length of the period of **DMA** access. The NMOS 6502 cpu is a dynamic processor and it must be avoided to shut off the processor for more than 40 microseconds (Rockwell only 17 microseconds).

Otherwise the cpu will loose the contents of its **registers** and then will jump to an unpredicted address after the end of the **DMA**-accesses. Therefor **DMA** accesses must be limited to a maximum of less than 40 microseconds. To keep the **registers** of the 6502 preserved the **DMA** must be limited within this limits periodically by short intermediate shutdowns..
This card realizes the task by using a hardware counter driving the timing: each 16 microseconds the 6502 performs a **wait-cycle**. While the 68008 controls the system every 16 microseconds a **wait-cycle** for the 6502 occurs (picture 2).

The circuitplan of the 68008

The introduced circuit (picture 3) contains from single blocks that consist as individual function-blocks. In the part 5 of the circuit the adresslines are buffered from the Apple and the adresslines above **A12** the **74LS283** adds a **1** to them. By this trick the 68008 adresses as 0 the Apple-adress $**1000** and all other addresses are shifted the same way 4 kByte upwards and that mapping protects the zeropage of the 6502 and the trap-vectors of the 68008 are relocated to another place. By this trick both cpu´s can be switched without conflict to each other. The flipflop in part **1** takes control of the switching of the card. By a write-access to the address $**Cx00** (**x** represents the slot no.) the **IOSEL** and **R/W** lines are set to low and the **Q**-line of the flipflop switches to low too. By an **OR**-gates and 4 Inverters the address-buffers are activated and **DMA** is set to low.
The arrangement of the inverters causes a shifting in the timing needed for the switching of the buffers and **Qinvert** of the flipflop drives the activation of the 68008.

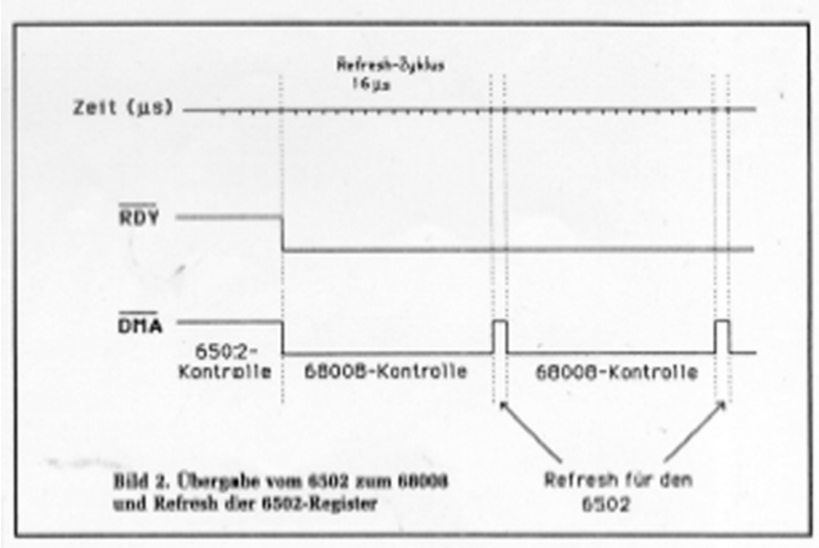

Bild 2. Übergabe vom 6502 zum 68008 und Refresh der 6502-Register

Another access to the same address - but from the 68008 switches the card off and passes the control back to the 6502. The buffers of the card are switched to **tristate-mode** and the **HALT**-line is drawn down to low. The **HALT** is also drawn down to low by **reset** as needed for the 68008. The architecture of the Apple request the **DMA** and **RDY**-line to be driven with open-collector and same is requested for the **HALT** and the **RES** of the 68008 cpu.
Both cpu´s can only use half of the time the memory and for the rest of the cycles the memory receives refresh by the video-access - but already during the video-phase the cpu must set the address for the next access. If you want to keep within the permitted timeframe the preparation of the access must start preparation 300ns after the starting of the video-phase.
For a RAM-read-access the data is available 468 ns after the positive rise at the **Φ0** on the databus to the cpu. The access to the ROM and Keyboard the delay is smaller till the data is available at the databus.
For write-access the data must be latest 210 ns after the rise at the **Φ0** availianle at the databus and must be for the period of at least minimum of 55 ns accessable.

The two chips **74LS373** and **74LS244** in part 3 of the circuitplan drive the databus in both directions. It also probably might be possible to use a transceiver but due to the fact that the data from the keyboard only has a short hold-time after the positive rise at the **Φ0** on the databus it is more secure to use a latch that is opened and closed with **Φ0** (Pin 11). It is the same technic as used by the Z80 cards.

the three flipflops in a row (part 2 of the circuitplan) are used to control the timing of the read- and write-cycles. At the end of a cpu-cycle the **DTACK** pin of the 68008 (which is a indicator of a successful read- or write-cycle) if at the address-bus a address was available - this must be true not later than 300 ns after the beginning of the preleading video-phase.
At the same time the flipflops secure that no access to the databus can be performed within the video-phase at the **74LS244**.
The counter (**74LS393**) is responsible for the refresh of the 6502. Every 16 µs it generates a signal that deactivates the address-buffer of the card at the beginning of a videophase and pulls up the **DMA**-line to high for the period of one microsecond.
The **RDY**-line remains low and therefore a **wait-cycle** is performed with the 6502. After the 6502 has performed the **wait-cycle** and the **registers** of the 6502 have been refreshed by this action, the control is given back to the 68008 cpu at the next video-phase. A line from the refresh-signal to the row of flipflops keeps sure that at the card the **DTACK** pin of the 68008 cant be pulled down to low.

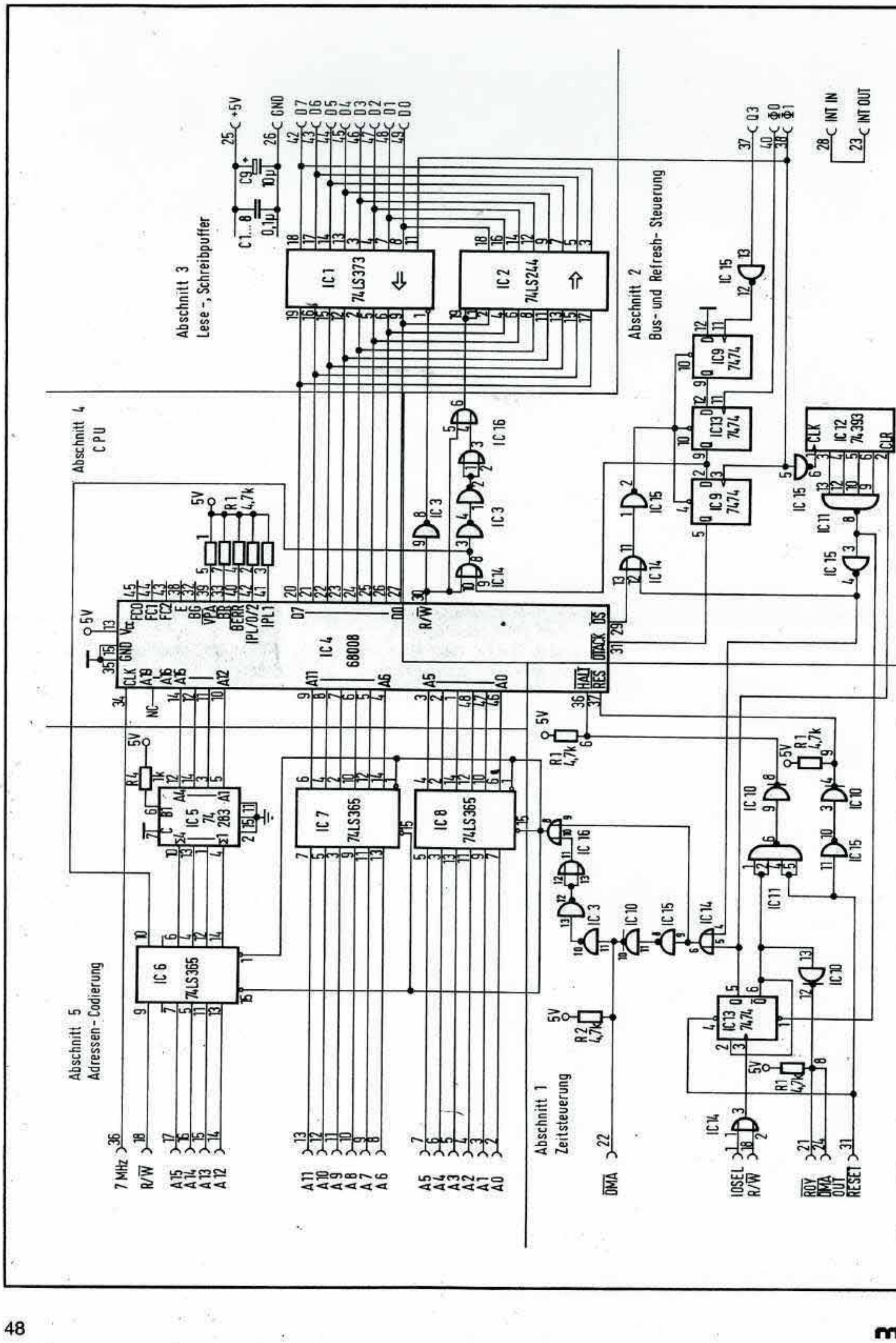

Bild 3. Schaltung der 68008-Karte. R1 ist ein Netzwerk aus acht Widerständen

Stückliste 68008-Karte

Halbleiter

2	74 LS 04	IC 3/15	HCT
1	74 LS 05	IC 10	
1	74 LS 20	IC 11	
2	74 LS 32	IC 14/16	ALS
2	74 LS 74	IC 9/13	
1	74 LS 244	IC 2	ALS
1	74 LS 283	IC 5	
3	74 LS 365	IC 6	
1	74 LS 373	IC 1	ALS
1	74 LS 393	IC 12	
1	MC 68008 P	IC 4	

Widerstände

1	4,7 kOhm Netzwerk	R1
1	4,7 kOhm	R2

Kondensatoren

8	100 nF	C1...C8
1	10uF/10V/Tantal	C9

Sonstiges

9	14-pol. IC-Sockel
4	16-pol. IC-Sockel
2	20-pol. IC-Sockel
1	48-pol. IC-Sockel
1	Platine

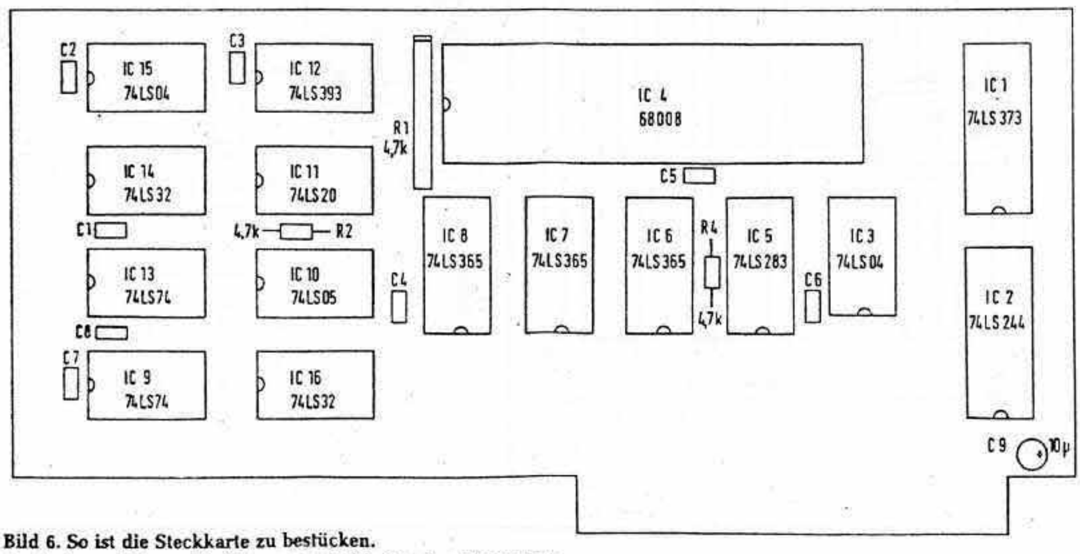

Bild 6. So ist die Steckkarte zu bestücken.
Am besten setzt man die ICs erst am Schluß in ihre Fassungen

The **CLK**-line of the 68008 is connected to the **7MHz**-line of the Apple. Allthough the cpu can operate up to 8 MHz it is fixed by this line to operate asyncronous to the **Φ0** and the **Φ1** signals.
This opes a "time-window" of 15 microseconds to the data-bus and closes that window for 1 microsecond for the refresh of the 6502 cpu.
This is a difference to the usual Z80 cards that are syncronized with the video- and cpu-phases of the Apple.

The 68008 has at 94% of the CPU-phase the possibility to access the addressing and databusses.

But the speed of the cpu is not really limited by the remaining 6% - its more limited by the slow access and speed of the Apple-RAM chips.

This is compensated a little by the prefetch-queue of the 68008: Each time the cpu is executing internal operation and no instructions to the bus are done the next instruction is read by it.

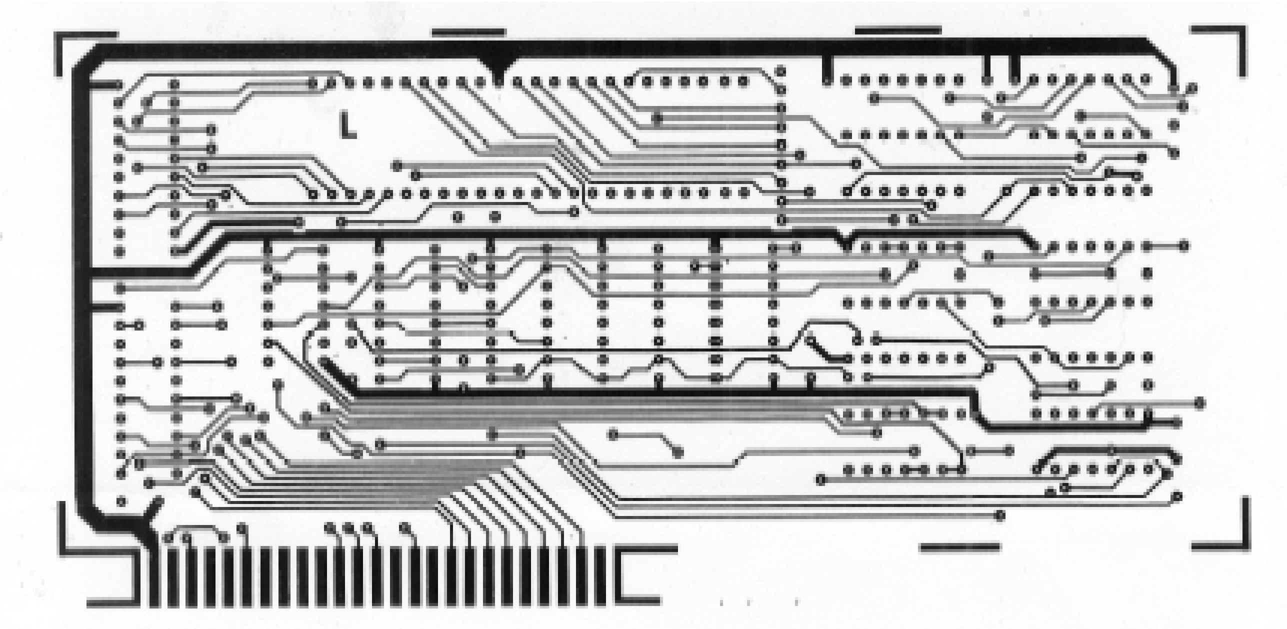

Bild 4. Lötseite der Platine

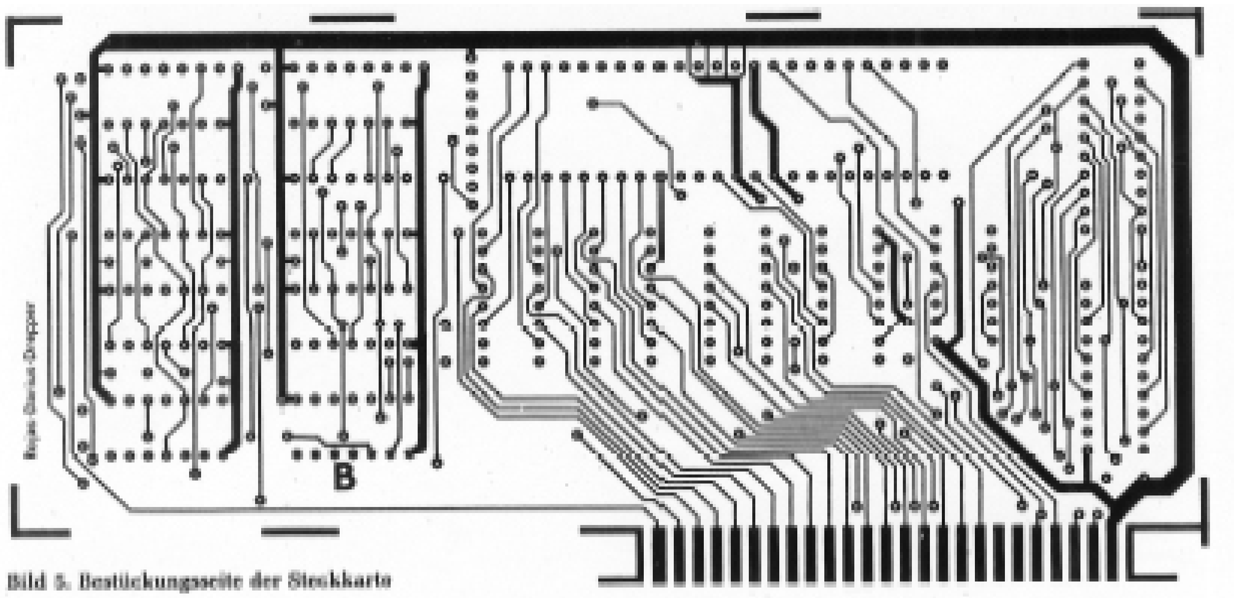

Bild 5. Bestückungsseite der Steckkarte

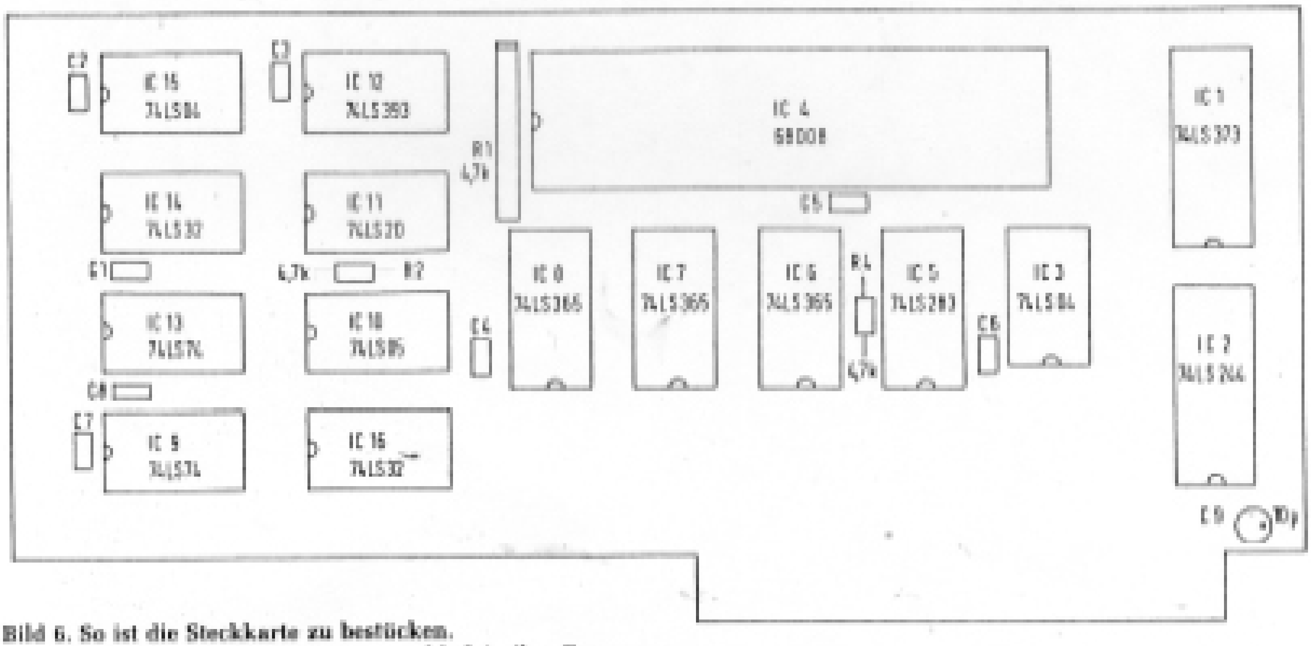

Bild 6. So ist die Steckkarte zu bestücken.
Am besten setzt man die ICs erst am Schluß in ihre Fassungen

Two important details at the end: This card must be placed at the beginning of the **DMA-chain** in the computer (i.e. slots) with the highest priority ! The Card is designed for the use in an **Apple II** or **II+** and / **or compatibles**. For use in the **IIe** the timing of the card must probably be changed slightly. The **74LS393** chip has an additional counter that might be used for this purpose.

Picture 4 shows the card at the soldering side, picture 5 shows the card at the side where it is populated and picture 6 shows where the IC´s and other components are to be placed.

The Testprogram

The picture 7 displays a small Testprogram that writes alternating a **A** or **B** to the display.

The Programm is loaded to the address $**1000** of the Apple (the address **0** for the 68008). To end the cycles a **reset** puts back the computer to the regular Apple-mode. The program can be written with the monitor in the Apple-memory (for slot 3) and the command **C300** (**return**) starts it.

If you want the program to terminate by itself and hand back control to the Apple you just must write a zero in the address $C**x**00 ($**Bx00** for the 68008), where **x** represents the slotnumber (picture 8).

If the user wants to write larger programs for the card it is usefull to use one of the available crossassemblers for DOS and the 68000.

```
1000 00 00 00 00              0                ;Stack-Adresse
1004 00 00 00 08              8                ;PC
1008 13 FC 00 41 00 00 F4 28  L: MOVE.B #41,#F428   ;Schreib A
1010 13 FC 00 42 00 00 F4 28  MOVE.B #42,#F428 ;Schreib B
1018 4E F9 00 00 00 08             JMP L
```

Bild 7. Mini-Testprogramm für die 68008-Karte. Es schreibt abwechselnd ein A und ein B auf dieselbe Bildschirm-Position

```
1000 00 00 00 00              0                ;Stack-Adresse
1004 00 00 00 08              8                ;PC
1008 13 FC 00 41 00 00 F4 28  MOVE.B #41,#F428  ;Schreib A
1010 13 FC 00 42 00 00 F4 28  MOVE.B #42,#F428  ;Schreib B
1018 13 FC 00 00 00 00 B1 00  MOVE.B #0,#B100   ;Slot 1
```

Bild 8. So wird der Rücksprung zum 6502-Modus durchgeführt

Literature:

[1] Starnes T.W.: Design Philosophy behind Motorola´s M6800, Byte April, May and June 1983
[2] Baum P.: DMA Protocol - Call A.P.P.L.E , April 1984
[3] Gayler W.D.: the Apple II Circuit description, Howard W. Sams Co. Indianapolis 1983
[4] Hilf W.; Nausch A.: M68000 Family Part 1 Te-Wi publishing, Munich 1984
[5] Apple II User Manual
[6] Microsoft Z80 Card User Manual

Addendum / Corrections

68008 card for the APPLE-II

mc 1985, Issue 9, Side 46

In the circuitplan (picture 3) is the pin 11 of the IC 1 (74LS373) connected with **Φ1** of the Slotconnector (connection 38) - in facr the pin of the 74LS373 must be connected (as displayed in the layout of the PCB) to the **Φ0** (the connection 40 at the slot).

68008 card for the APPLE-II

mc 1985, Issue 9, Side 46

It turned out that with some compatible computers the 68008 card faced timing-problems. In the most cases this issue was just simply solved by swapping the 74LS74 to standard 7474 chips.

In very rare cases it happened that the contents of the memory-cells changed when switching the control from 6502 to 68008 or backwards.This problem was related in that specific cases to problems with temperature and was solved by adding a capacitor with a value between 100 pF and 300 pF between pin 36 (HALT) of the 68008 and ground - that delays the HALT-signal a very little bit and avoids conflict with the DMA-access. Another advice would be to swap the ICs 10 and 11 (74LS05 and 74LS20) against faster versions (i.e. 74ALS05 and 74ALS29). In case that all these steps did not solve an existing problem the LS-typ chips should be swapped again back to the LS-versions and remove the capacitor. But instead solder a resistor with 1 kilo Ohm between pin 8 and pin 10 at IC10 (the 74LS05). With this solution the trouble will be solved surely but at the trade that the 68008 cpu will operate at less speed.

The card should not be used at the same time in a system as the Z80 card. It should be avoided - unless the card has been altered like in the correction description of the issue 2 of this year.

Remark from me:
Unfortunately I don´t have that issue….
So the only sure solution is to pull the Z80 card out of the computer, when working with the 68008 card !