

Raul Rojas, Frank Darius, Jürgen Drepper

68008-Karte für den Apple-II

Der folgende Bauvorschlag beweist erneut die Flexibilität des Apple-II, die vor allem seinem Slot-Konzept zu verdanken ist: Eine Karte mit dem zum 68000 voll kompatiblen Prozessor 68008. Sie funktioniert ähnlich wie die bekannten Z80-Karten, bietet aber alle Vorteile der internen 32-Bit-Struktur des 68008.

Der Mikroprozessor 68000 ist einer der populärsten unter den 16/32-Bit-Prozessoren. Ausschlaggebend für den Erfolg dieses Produkts der amerikanischen Firma Motorola ist seine innere Architektur, die einen sehr komfortablen Befehlssatz sowie zwölf Adressierungsmodi bietet. Das verbesserte Modell 68010 kann bis 16 Megabyte Speicherplatz durch Virtual-Memory-Technik adressieren und verfügt über eine intelligente Prefetch-Queue, die die Ausführung kleiner Programmschleifen deutlich beschleunigt. Der 68000 bietet intern 8 Daten- und 8 Adreßregister von je 32 Bit,

obwohl der Datenbus nur 16 Leitungen umfaßt [1]. So war es nur eine Frage der Zeit, bis es eine echte 32-Bit-Version gab: Ende Juni 1984 erschien in den USA das Modell 68020, kompatibel zum 68000 und 68010. Motorola liefert eine Version, die bis zu 2,5 MIPS (Millionen Befehle pro Sekunde) ausführen und bis zu 4 Gigabyte an logischem Speicher adressieren kann. Vom 68000-Prozessor existiert eine kleine Ausführung mit einem achtpoligen Datenbus, der 68008. Dadurch ist dieser Prozessor hardware-kompatibel zu anderen 8-Bit-Mikroprozessoren und kann

auf deren Datenspeicher und Peripheriebausteine zugreifen. So ergibt sich die Möglichkeit, den 68008 innerhalb eines älteren Mikrocomputers einzusetzen. Ohne sich einen neuen Datenspeicher oder Peripherie anschaffen zu müssen, hat man dann einen Prozessor der 68000-Familie zur Verfügung. Genau das leistet die Schaltung, die wir in diesem Artikel als Apple-Steckkarte vorstellen. Der 6502-Prozessor des Apple-II wird ausgeschaltet und die Kontrolle über den Rest des Computers vom 68008 übernommen, ähnlich wie bei den gängigen Z80-Karten. Dem 6502 kann zwischendurch per Software die Kontrolle übergeben werden, um dann Monitor-Routinen, wie das Darstellen von Ergebnissen auf dem Bildschirm, auszuführen. Beide Prozessoren können den gleichen Speicher benutzen und sich abwechseln.

Die Hardware-Problematik

Der Apple verfügt über acht Slots auf der Grundplatine, die den Anschluß von Zusatzprozessoren ermöglichen. Neben Daten- und Adreßbus bieten diese Slots weitere Leitungen mit Takt- und Kontrollsignalen, die jede zusätzliche Platine benutzen kann. Die Leitung 22 in dem 50poligen Bus wird DMA (Direct Memory Access) genannt. Wird diese Leitung auf Low (0 Volt) gezogen, werden der Adressen- und Datenbus des Apple in den Tristate-Zustand versetzt und der 6502 gestoppt. Andere Prozessoren oder Geräte können dann die Adressen- und Datenbusse treiben und Lese- oder Schreibzyklen auf dem logischen Speicher (RAM-, ROM- und I/O-Adressen) ausführen. Für eine reibungslose Übergabe an einen anderen Prozessor muß der 6502-Bus-Zugriff jeweils vollständig abgeschlossen sein. Für die Übergabe der Kontrolle an den 68008 muß folgendes geschehen:

- 1) DMA muß während $\Phi 1$ (Video-Phase) Low gesetzt werden.
- 2) 30 ns danach kann der Adressenbus vom 68008 getrieben werden.

Für die Rückgabe der Kontrolle an den 6502 gilt:

- 1) Die Adressen- und Datenpuffer der 68008-Karte müssen in den Tristate-Zustand versetzt werden.
- 2) DMA muß spätestens 178 ns nach einer positiven $\Phi 1$ -Flanke auf High gezogen werden, damit der 6502 noch seinen eigenen Adressenbus setzen kann (Bild 1).

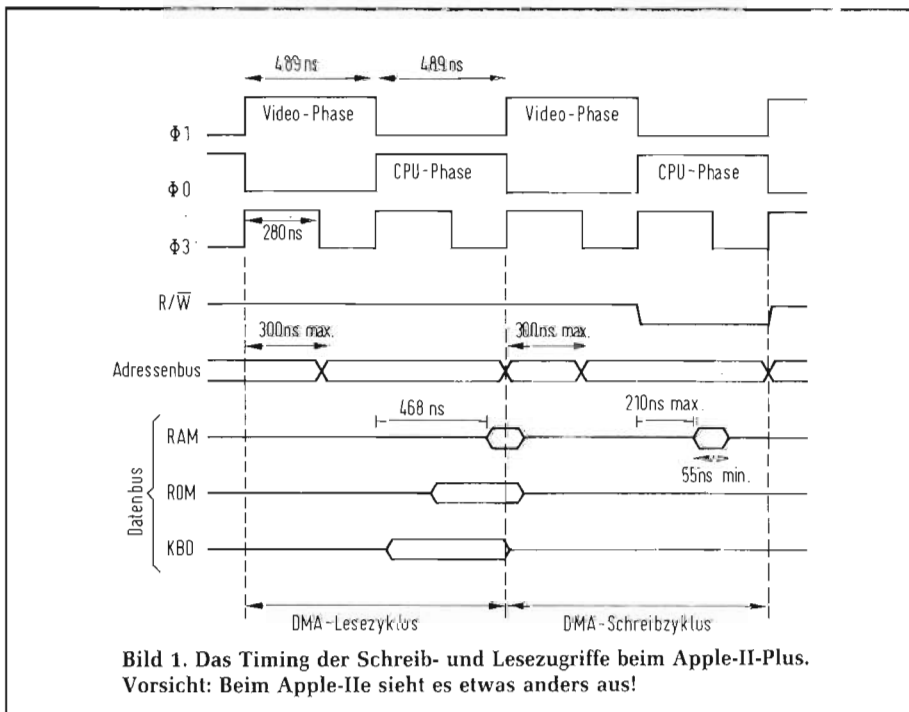


Bild 1. Das Timing der Schreib- und Lesezugriffe beim Apple-II-Plus. Vorsicht: Beim Apple-IIe sieht es etwas anders aus!

Es muß also gewährleistet sein, daß entweder der 6502 oder der 68008 die Kontrolle über Speicher und Peripherie des Apples hat und die Übergabe der Kontrolle das DMA-Protokoll erfüllt.

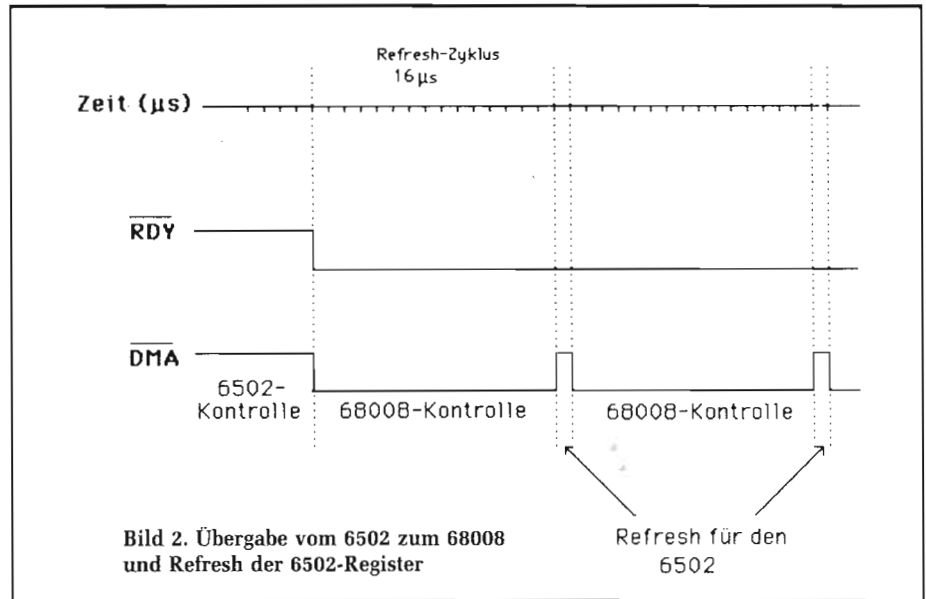
Eine Schwierigkeit ergibt sich aus der Dauer der DMA-Zugriffe. Der NMOS-6502-Prozessor ist ein dynamischer Prozessor und kann nicht länger als 40 Mikrosekunden angehalten werden (Rockwell: 17 Mikrosekunden). Andernfalls verliert er den Inhalt seiner Register und springt beim Ende des DMA-Zugriffs auf eine falsche Programmstelle. Deshalb müssen die DMA-Zugriffe weniger als 40 Mikrosekunden dauern. Um die Kontrolle eines Zusatz-Prozessors über den Computer für längere Zeit zu ermöglichen (ohne die Register des 6502 zu verlieren), müssen DMA-Zugriffe zwischendurch regelmäßig unterbrochen werden. In unserer Schaltung haben wir dies hardwaremäßig realisiert: Alle 16 Mikrosekunden unterbricht der 68008 seine Adressen-Zugriffe für eine Mikrosekunde und läßt den 6502 einen Wait-Zyklus ausführen.

Während der 68008 das System kontrolliert, findet also zyklisch ein Refresh für den 6502 statt (Bild 2). Das 68000-Busverhalten ist übrigens in mc 9/1984 auf Seite 83 beschrieben.

Die Schaltung der 68008-Karte

Die vorgestellte Schaltung (Bild 3) besteht aus einzelnen Abschnitten, die jeweils logische Einheiten bilden. In Abschnitt 5 der Schaltung werden die Adressen-Leitungen gegenüber dem Apple gepuffert; ab A12 wird eine 1 mit dem 74LS283 addiert. Dadurch erkennt der 68008 als Adresse Null die Apple-Adresse \$1000, und auch alle anderen Adressen werden um diese 4 KByte verschoben. Mit diesem „Mapping“ wird die Zero-Page des 6502 geschützt und die Trap-Vektoren des 68008 auf eine andere Stelle im Speicher verlegt. So können beide Prozessoren abwechselnd die Kontrolle übernehmen, ohne sich dabei ins Gehege zu kommen.

Das Flipflop in Abschnitt 1 der Schaltung kontrolliert das An- und Ausschalten der Platine. Durch einen Schreibzugriff des 6502 auf die Adresse \$Cx00 mit x als Slotnummer werden die Leitungen IOSEL und R/W vorübergehend auf Low gesetzt, und die Leitung Q des Flipflops geht in den Low-Zustand. Durch ein OR und vier Inverter werden die Adressenpuffer aktiviert und DMA auf Low gesetzt. Die Anordnung der Inverter sorgt



für die nötige Zeitverschiebung beim An- und Ausschalten des Puffers. Q des Flipflops steuert die Aktivierung des 68008.

Ein weiterer Zugriff auf dieselbe Adresse \$Cx00, diesmal aber vom 68008, schaltet die Karte aus und reaktiviert den 6502. Die Puffer der Platine werden in den Tristate-Zustand versetzt und HALT auf Low gezogen. Auch bei einem Reset wird HALT auf Low gezogen, wie es für den 68008 nötig ist. Außerdem verlangt die Architektur des Apple, daß die Leitungen DMA und RDY mit Open-Collector-Ausgängen getrieben werden. Das ist auch der Fall bei HALT und RES des 68008.

Die CPU (6502 oder 68008) kann den Speicher nur die Hälfte der möglichen Zeit benutzen. Während der verbleibenden Zeit bekommt er durch Video-Zugriffe die nötigen Refresh-Signale. Doch schon während der Video-Phase muß die CPU den Adressenbus für den nächsten Zugriff setzen. Will man auf der sicheren Seite liegen, so soll der Zugriff spätestens 300 ns nach Beginn der Video-Phase vorbereitet werden.

Bei einem RAM-Lesezugriff liegen die Daten 468 ns nach der positiven Flanke von $\Phi 0$ auf dem Datenbus und können vom Prozessor gelesen werden. Bei ROM- und Tastatur-Zugriffen dauert es noch kürzer, bis die Angaben auf den Datenbus gelangen.

Bei Schreib-Zugriffen müssen die Daten maximal 210 ns nach $\Phi 0$ -High am Datenbus anliegen und dort mindestens 55 ns lang verbleiben.

Die zwei Chips 74LS373 und 74LS244 in Abschnitt 3 der Schaltung treiben den Datenbus in zwei Richtungen. Es wäre vielleicht möglich, sie durch einen Transceiver zu ersetzen, aber weil die Daten aus der Tastatur eine sehr kleine Hold-Zeit nach der negativen Flanke von $\Phi 0$ haben, ist es einfacher, ein Latch für Lesezugriffe zu verwenden, der mit $\Phi 0$ geöffnet und geschlossen wird (Pin 11). Es ist dies die gleiche Technik wie bei Z80-Karten.

Die drei Flipflops in Reihe (Abschnitt 2 der Schaltung) haben die Funktion, das Timing der Schreib- und Lese-Zugriffe zu überwachen. Am Ende einer CPU-Phase wird der Eingang DTACK vom 68008 nur dann auf Low gesetzt (was das erfolgreiche Ende eines Lese- oder Schreibzyklus bedeutet), wenn am Adressenbus rechtzeitig eine Adresse anlag; also spätestens 300 ns nach Beginn der vorhergehenden Video-Phase. Gleichzeitig wird durch die Flipflops die Bedingung realisiert, daß während der Video-Phase kein Schreib-Zugriff über den 74LS244 auf den Datenbus stattfinden kann.

Der Zähler (74LS393) ist für den Refresh des 6502 zuständig. Alle 16 µs erzeugt er ein Signal, das den Adressenpuffer der Platine am Anfang einer Video-Phase deaktiviert und die DMA-Leitung (Leitung 22), für eine Mikrosekunde auf High zieht. Die Leitung RDY bleibt Low, so daß ein Leerlaufzyklus des 6502 erzeugt wird. Nachdem der 6502 so seine Register aufgefrischt hat, wird bei der nächsten Video-Phase die Kontrolle an den 68008 zurückgegeben. Eine Leitung vom Refresh-Signal zur Flipflop-Reihe

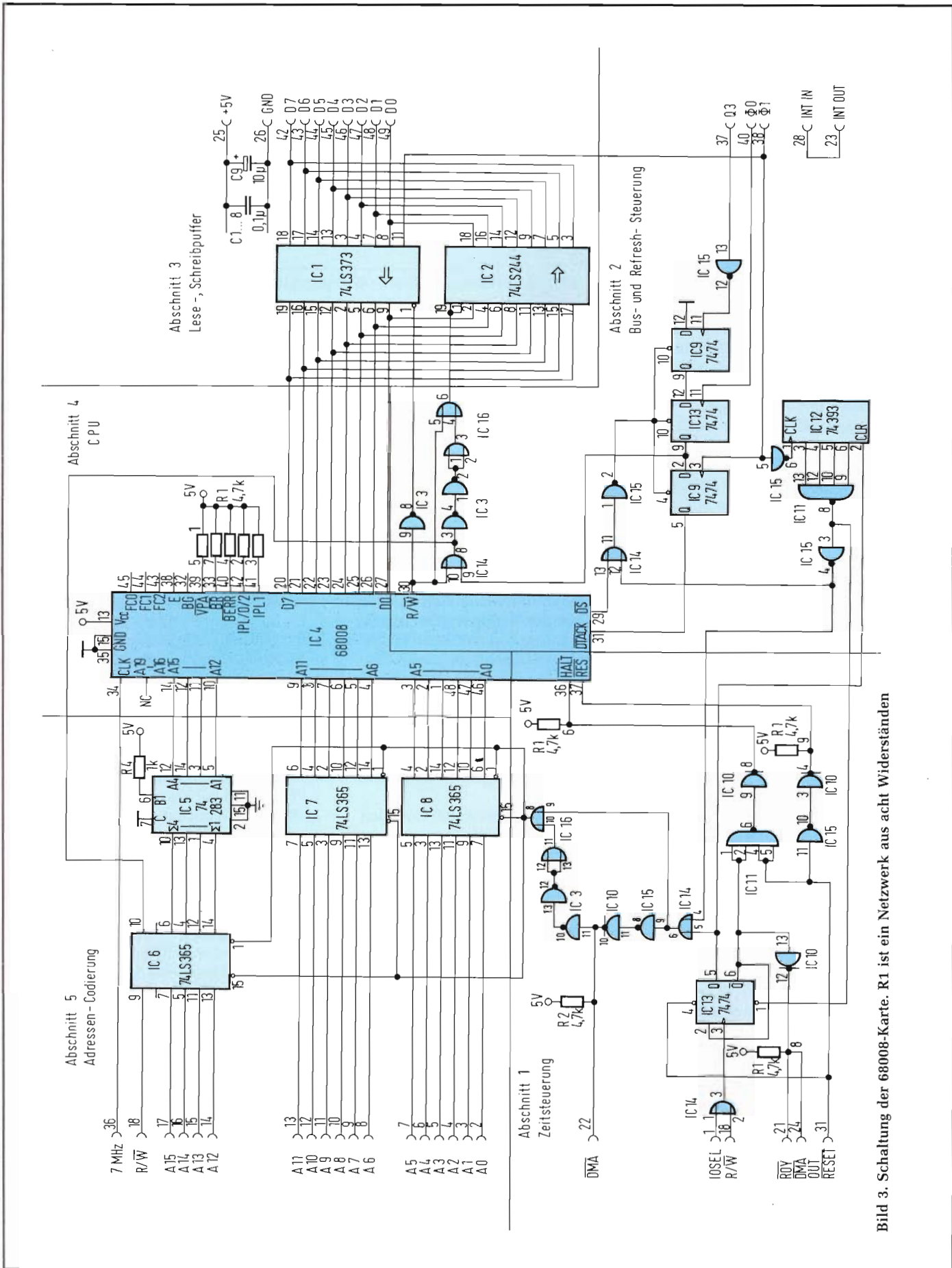


Bild 3. Schaltung der 68008-Karte. R1 ist ein Netzwerk aus acht Widerständen

sorgt innerhalb der Platine dafür, daß während des Refresh der DTACK-Eingang des 68008 nicht Low gesetzt wird. Der CLK-Eingang des 68008 ist an die 7-MHz-Leitung des Apple angeschlossen. Der Prozessor, der auch mit 8 MHz betrieben werden könnte, läuft also asynchron zu den $\Phi 0$ - und $\Phi 1$ -Signalen. Der Datenbus ist durch ein „Zeitfenster“ 15 Mikrosekunden lang geöffnet und

1 Mikrosekunde geschlossen. Dies beides sind Unterschiede zu den gängigen Z80-Karten, wo der Prozessor mit den Video- und CPU-Phasen des Apples synchronisiert ist.

Der 68008 hat während 94 % der Zeit der CPU-Phase die Möglichkeit, Adressen- und Datenbusse zu benutzen. Doch weniger die verbleibenden 6 %, ohne

Zugriff auf die Busse, beschränken die Verarbeitungsgeschwindigkeit des 68008, als vielmehr der langsame Zugriff auf das Apple-RAM. Dies kann etwas kompensiert werden durch die Prefetch-Queue des 68008: Immer zwischendurch, wenn der Prozessor gerade am Rechnen ist, aber sonst am Bus nichts zu tun hat, wird schon der folgende Befehl eingelesen.

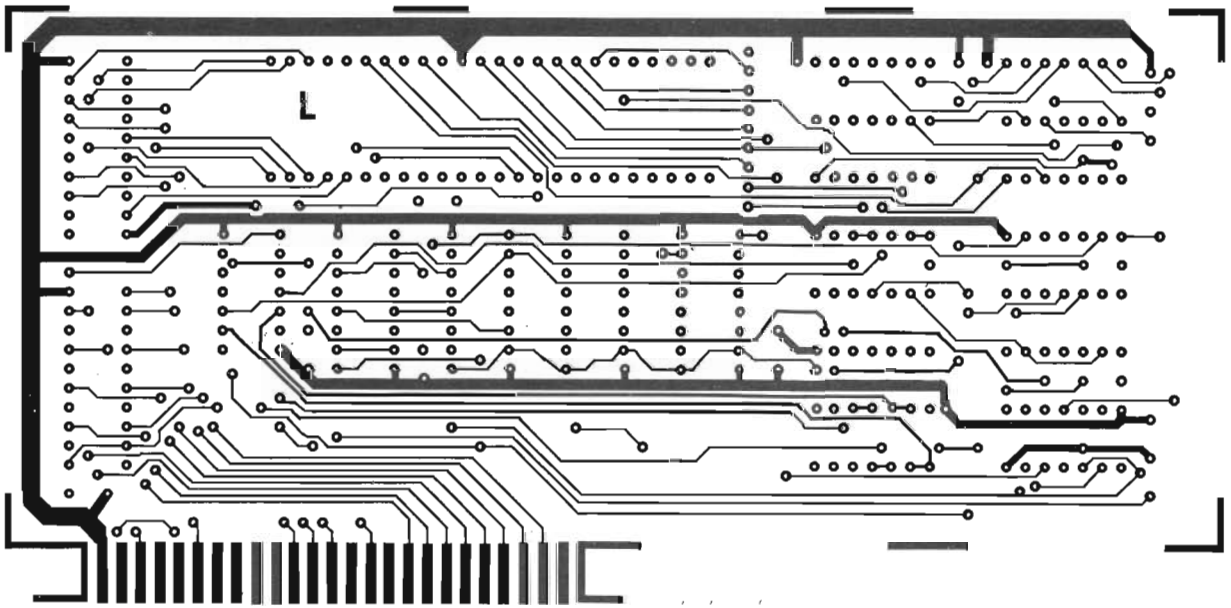


Bild 4. Lötseite der Platine

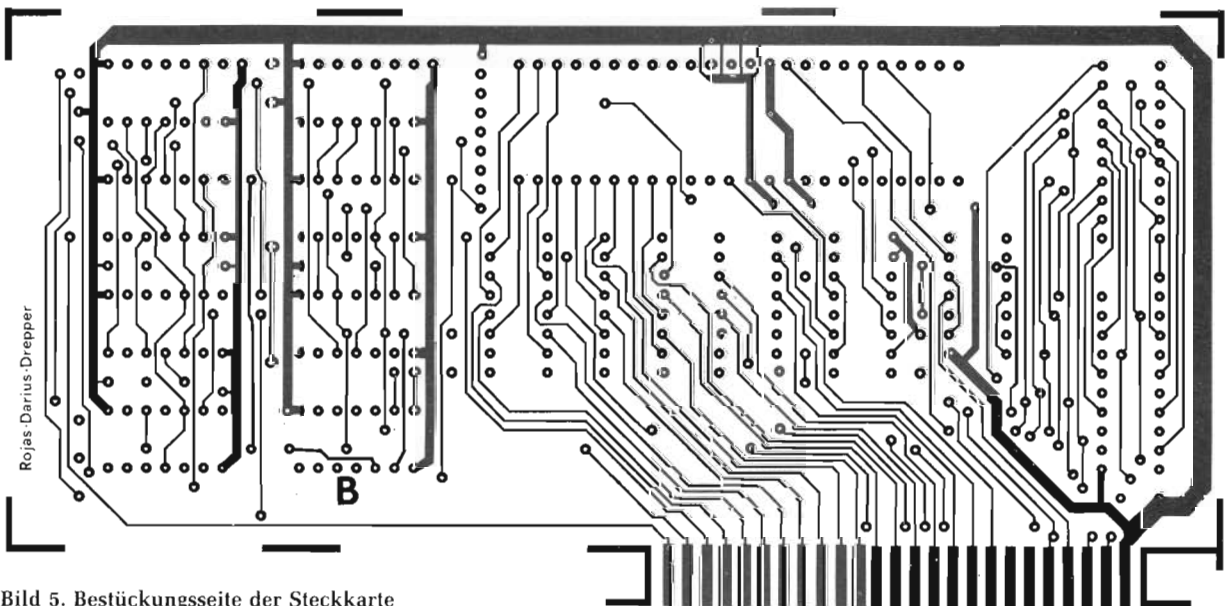


Bild 5. Bestückungsseite der Steckkarte

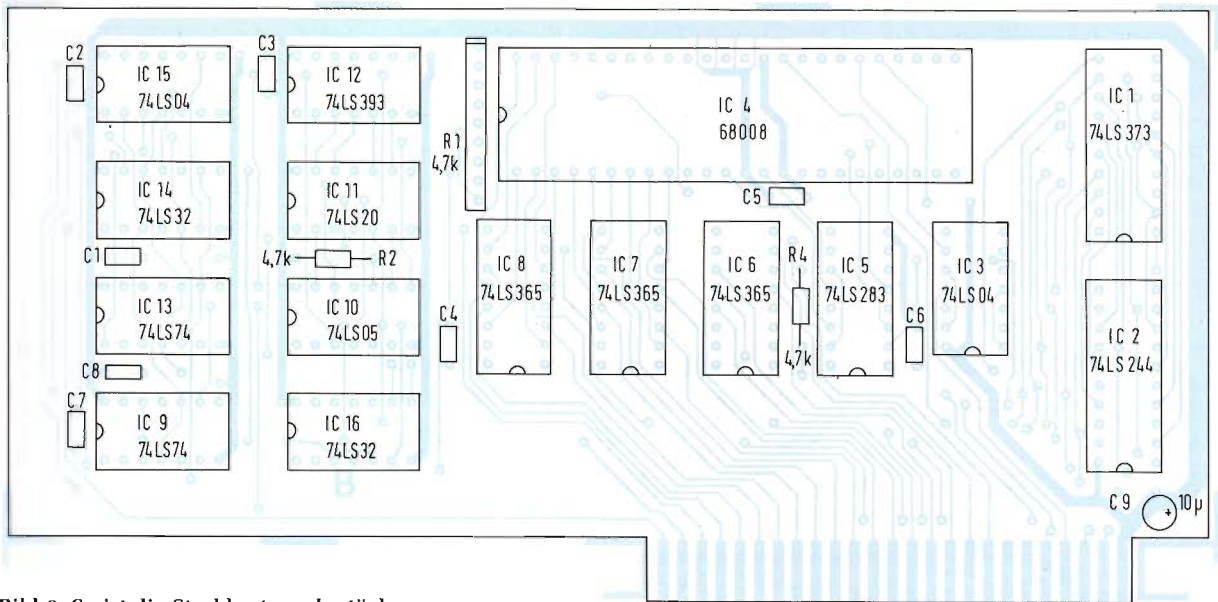


Bild 6. So ist die Steckkarte zu bestücken.
Am besten setzt man die ICs erst am Schluß in ihre Fassungen

Noch zwei letzte Details: Die vorgestellte Karte muß an den Anfang der DMA-Daisy-Chain gesetzt werden, falls es eine gibt (höchste Priorität vor allen anderen Steck-Karten für den Direct Memory Access). Sie ist ferner auf das Timing des Apple-II+ und kompatible Geräte ausgelegt. Für den Apple-IIe muß das Timing der Schreib- und Lese-Zyklen wahrscheinlich geändert werden. Der Chip 74LS393 verfügt über einen weiteren Zähler, der dafür eingesetzt werden könnte.

Bild 4 zeigt die Lötseite, Bild 5 die Bestückungsseite und Bild 6 den Bestückungsplan der Steckkarte. (Die Platine ist von unserem Software-Service erhältlich.)

Ein Testprogramm

Das Programm in Bild 7 ist ein kleines Test-Programm, das nur ein A und ein B auf dem Bildschirm abwechselnd schreibt. Das Programm wird ab Adresse \$1000 gespeichert (Adresse Null für den

68008). Aus der Schleife kommt man mit einem Reset zum normalen Apple-Betriebssystem zurück. Das Programm kann mit dem Monitor des Apple geladen und (für Slot 3) mit C300 (Return) gestartet werden.

Will man, daß das Programm selbst die Kontrolle an den 6502 zurückgibt, dann kann man einfach eine Null in die Adresse \$Cx00 (\$Bx00 für den 68008) schreiben, wobei x die Slotnummer ist (Bild 8).

Der Leser wird später sicher kompliziertere Programme für die Karte schreiben wollen. Dafür kann er Cross-Assembler unter DOS benutzen, von denen es einige schon auf dem Markt gibt.

```

1000 00 00 00 00 00      0      ;Stack-Adresse
1004 00 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 00      0      ;Stack-Adresse
1004 00 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 00 B1 00  MOVE.B #0,*B100      ;Slot 1
    
```

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

Literatur

- [1] Starnes, T.W.: Design Philosophy Behind Motorola's M68000. Byte, April, Mai und Juni 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-Familie: Teil 1, Grundlagen und Architektur. Te-Wi-Verlag, München 1984.
- [5] Apple II Benutzer Handbook.
- [6] Microsoft Z80 Card Manual.