

KAPITEL 4

Der Speicher-Aufbau

- 73 Schreib-Lese-Speicher (RAM)
- 74 RAM-Aufteilungsstecker
- 76 Festwert-Speicher (ROM)
- 77 Ein-/Ausgabe-Adressen
- 78 Verwendung der Seite Null

Der 6502-Mikroprozessor des Apple kann auf maximal 65.536 verschiedene Speicherstellen direkt zugreifen. Sie können sich den Speicher des Apple als ein Buch mit 256 „Seiten“ vorstellen mit jeweils 256 Speicherstellen auf jeder Seite. So ist beispielsweise „Seite \$30“ ein 256 Byte-Speicherbereich, der die Speicherstellen von Adresse \$3000 bis zur Adresse \$30FF enthält. Da der 6502-Mikroprozessor zwei Bytes zu je 8 Bit braucht, um die Adresse jeder Speicherstelle zu bilden, können Sie sich eines dieser Bytes als die Seitennummer und das andere Byte als die Adresse innerhalb der Seite vorstellen.

Die 256 Seiten des Apple-Speicherbereichs gliedern sich in drei Kategorien: Schreib-Lese-Speicher (RAM), Festwertspeicher (ROM) und alle Ein-/Ausgabe-Adressen (I/O). Unterschiedliche Speicherbereiche haben verschiedene Funktionen. Der prinzipielle Speicherplan des Apple sieht folgendermaßen aus:

| SPEICHERPLAN | | | |
|---------------------|------|--------------|-----------|
| Seiten-Nummer: | | | |
| Dezimal | Hexa | | |
| 0 | \$00 | RAM (48K) | |
| 1 | \$01 | | |
| 2 | \$02 | | |
| . | | | |
| . | | | |
| 190 | \$BE | | |
| 191 | \$BF | | |
| 192 | \$C0 | | I/O (2K) |
| 193 | \$C1 | | |
| . | | | |
| . | | | |
| 198 | \$C6 | | |
| 199 | \$C7 | | |
| 200 | \$C8 | I/O ROM (2K) | |
| 201 | \$C9 | | |
| . | | | |
| . | | | |
| 206 | \$CE | | |
| 207 | \$CF | | |
| 208 | \$D0 | | ROM (12K) |
| 209 | \$D1 | | |
| . | | | |
| . | | | |
| . | | | |
| 254 | \$FE | | |
| 255 | \$FF | | |

Schaubild 5. Aufteilung des Speichers in Funktionsbereiche

Schreib-Lese-Speicher (RAM)

Der RAM-Speicherbereich des Apple beginnt unten auf der Seite 0 und erstreckt sich bis zum Ende der Seite 191. Der Apple kann zwischen 4K (4096 Bytes) und maximal 48K (49.152 Bytes) RAM auf seiner Hauptplatine aufnehmen. Mit einer Apple Language Card (Teile-Nummer A2B0006) können Sie den RAM-Speicher des Apple auf insgesamt 64K (65.536 Bytes) erweitern. Diese zusätzlichen 16K RAM belegen den ROM-Speicherbereich des Apple, wobei zwei 4K-Teile des RAM-Speichers einen 4K-Bereich von \$D000 bis \$DFFF wechselweise benutzen.

Der größte Teil des Apple RAM-Bereichs steht Ihnen zur Speicherung von Programmen und Daten zur Verfügung. Jedoch stellt der Apple einige Adressen vom RAM dem System-Monitor, verschiedenen Sprachen und anderen Systemfunktionen zur Verfügung. Hier sehen Sie die Aufteilung der verfügbaren Bereiche im RAM-Speicher :

| Tabelle 16: RAM-Aufteilung | | | |
|-----------------------------------|------|---|---------------------------|
| Seiten-Nummer | | reserviert für: | |
| Dezimal | Hexa | | |
| 0 | \$00 | System-Programme | |
| 1 | \$01 | Sytem-Keller | |
| 2 | \$02 | GETLN Eingabe-Puffer | |
| 3 | \$03 | Sprungadressen für den Monitor | |
| 4 | \$04 | Text und Lo-Res Graphik Erste Seite | |
| 5 | \$05 | | |
| 6 | \$06 | | |
| 7 | \$07 | | |
| 8 | \$08 | Text und Lo-Res Graphik Zweite Seite | freier RAM-Bereich |
| 9 | \$09 | | |
| 10 | \$0A | | |
| 11 | \$0B | | |
| 12 | \$0C | | |
| bis 31 | \$1F | | |
| 32 | \$20 | Hi-Res Graphik Erste Seite | |
| bis 63 | \$3F | | |
| 64 | \$40 | Hi-Res Graphik Zweite Seite | |
| bis 95 | \$5F | | |
| 96 | \$60 | | |
| bis 191 | \$BF | | |

Weitere Angaben zu den ersten sieben Seiten :

Seite Null. Aufgrund des Aufbaus des 6502-Mikroprozessors kann die Seite Null von allen Maschinenprogrammen besonders vielfältig und vorteilhaft adressiert werden. Der System-Monitor verwendet ungefähr 20 Adressen der Seite Null. Das Apple Integer BASIC braucht einige mehr und Applesoft II BASIC und das Apple Disk Operating System benötigen den Rest. Die Tabellen 18, 19, 20 und 21 zeigen die Adressen der Seite Null, die von diesen Systemprogrammen gebraucht werden.

Seite Eins. Der 6502-Mikroprozessor des Apple reserviert alle 256 Bytes der Seite Eins zur Benutzung als „Keller“. Obwohl der Apple meist weniger als die Hälfte dieser Seite verwendet, sollten Sie nicht versuchen, irgendwelche Daten auf Seite Eins zu speichern, denn es ist sehr schwer festzustellen, welcher Bereich gerade unbenutzt ist.

Seite Zwei. Das GETLN-Unterprogramm, das den meisten Programmen und Sprachen zur Eingabe dient, benutzt die Seite Zwei als Eingabepuffer. Falls Sie sicher sind, daß Sie keine langen Eingabezeilen tippen, können Sie vorübergehend Daten im oberen Bereich der Seite Zwei speichern.

Seite Drei. Der Monitor des Apple (der Autostart-ROM ebenso wie der Monitor-ROM) nutzt die oberen 16 Adressen der Seite Drei von Adresse \$3F0 bis \$3FF (dezimal 1008 bis 1023). Die Verwendung dieser Adressen ist in Tabelle 14 (Seite 68) beschrieben.

Seite Vier bis Seite Sieben. Dieser Speicherbereich von 1024 Bytes wird für die Anzeige der ersten Bildschirmseite Text und Lo-Res Graphik gebraucht und steht deshalb nicht für Speicherzwecke zur Verfügung. Es werden 64 Speicherzellen dieses Bereichs nicht auf dem Bildschirm angezeigt. Diese 64 Adressen sind für die Peripherie-Karten reserviert (siehe Seite 87).

RAM-Aufteilungsstecker

Der RAM-Speicher des Apple setzt sich aus acht bis 24 integrierten Schaltkreisen zusammen. Diese ICs stecken in drei Socketreihen auf der Apple-Platine. Jede Reihe kann acht ICs der 4096-Bit-Ausführung (4K) oder der 16.384-Bit-Ausführung (16K) aufnehmen. Als 4K RAM ICs können Sie Bausteine der Mostek 4096-Serie mit der Bezeichnung „MK4096“ oder „MCM6604“ verwenden. Die 16K RAM ICs sind vom Typ 4116 und können die Bezeichnung „MK4116“ oder „UPD4160“ tragen. Jede Reihe muß aus acht ICs desselben Typs bestehen, obwohl jede Reihe einen anderen Typ enthalten kann.

Eine Reihe von acht 16K ICs ergibt 16.384 acht-Bit-Bytes RAM-Speicher. Der IC ganz links in einer Reihe ist das Bit 0 jedes Bytes in diesem Bereich, und der IC ganz rechts in einer Reihe ist das höchste Bit (Bit 7) jedes Bytes. Die vorderste Reihe enthält den RAM-Speicherbereich, der ab Adresse 0 beginnt. Die jeweils nächste Reihe schließt mit ihrem Speicherbereich an den vorhergehenden an.

Durch das Einstecken der sog. RAM-Aufteilungsstecker in drei IC-Socket links auf der Apple-Platine erfährt der Apple, wieviel Speicher er zur Verfügung hat und aus welchem Typ er sich zusammensetzt. Diese drei Aufteilungsstecker haben je 14 Füße und sehen wie große hohe ICs aus. Aber es sind keine Schaltkreise in ihnen, nur drei Verbindungsdrähte. Die Verbindungsdrähte schließen jede IC-Reihe an einen Bestimmten Speicherbereich des Apple an.

Apple liefert verschiedene Standardtypen von Aufteilungssteckern für die gebräuchlichsten Systemgrößen. Ein Satz davon befindet sich beim Kauf in Ihrem Apple und Sie erhalten jedesmal einen neuen Satz, wenn Sie eine Speichererweiterung für Ihren Apple kaufen. Falls Sie jedoch selbst Ihren Apple mit irgendwelchen RAM-ICs erweitern wollen, die Sie nicht von Apple gekauft haben, müssen Sie evtl. selbst eigene Stecker bauen (oder die im Apple ursprünglich vorhandenen verändern).

Es sind neun verschiedene RAM-Speicheraufteilungen in Ihrem Apple möglich. Diese neun Speichergrößen ergeben sich aus verschiedenen Kombinationen von 4K und 16K RAM ICs in den drei Sockelreihen Ihres Apples. Hier sind die neun Speicheraufteilungen :

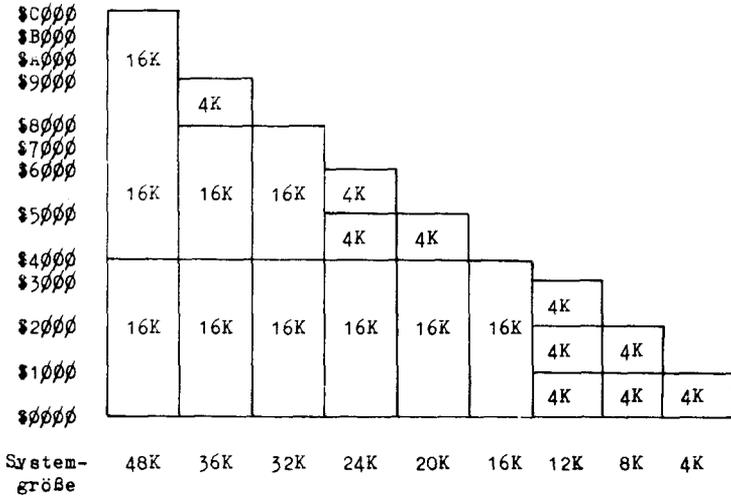


Schaubild 6. Mögliche Speicheraufteilungen

Die drei Füße oben rechts (von oben betrachtet) der 14 Füße eines Aufteilungssteckers symbolisieren die drei Sockelreihen auf der Hauptplatine des Apple. Eine Drahtverbindung sollte von jedem dieser drei Anschlüsse zu dem Anschluß führen, der dem gewünschten Speicherbereich entspricht. Die Anschlüsse der Speicheraufteilungsstecker werden wie folgt dargestellt :

| | | | | |
|------------|---------------|---|----|---------------------------|
| 4K Bereich | \$0000-\$0FFF | 1 | 14 | vordere Reihe ("C") |
| 4K Bereich | \$1000-\$1FFF | 2 | 13 | mittlere Reihe ("D") |
| 4K Bereich | \$2000-\$2FFF | 3 | 12 | hintere Reihe ("E") |
| 4K Bereich | \$3000-\$3FFF | 4 | 11 | kein Anschluß |
| 4K Bereich | \$4000-\$4FFF | 5 | 10 | 16K Bereich \$0000-\$3FFF |
| 4K Bereich | \$5000-\$5FFF | 6 | 9 | 16K Bereich \$4000-\$7FFF |
| 4K Bereich | \$8000-\$8FFF | 7 | 8 | 16K Bereich \$8000-\$BFFF |

Schaubild 7. Anschlußdiagramm der Speicheraufteilungsstecker

Enthält eine Reihe acht 16K-ICs, sollte der Anschluß der Reihe zu einem 16K-Bereichsanschluß führen. Entsprechend sollten Sie den Anschluß für diese Reihe mit einem 4K-Bereichsanschluß verbinden. Sie sollten keine 4K ICs einer Reihe in einen 16K-Bereich legen und umgekehrt. Es ist auch nicht ratsam, eine Reihe nicht anzuschließen oder zwei Reihen in denselben Speicherbereich zu legen.

Sie sollten unbedingt einen der beiden Speicherbereiche \$0000-\$0FFF oder \$0000-\$3FFF mit einem Speicherbaustein belegen. Es empfiehlt sich, den Speicherbereich aus einem zusammenhängenden Block zu bilden, es ist aber nicht unbedingt erforderlich.

So können Sie beispielsweise die erste Seite der Hi-Res-Graphik benutzen, wenn Sie nur drei Sätze 4K-ICs haben und dazu zwei Reihen 4K-ICs an den Speicherbereich für die Hi-Res-Graphik (4K-Bereiche \$2000-\$2FFF und \$3000-\$3FFF) anschließen. Die übrige 4K-Reihe muß dann an den 4K-Bereich \$0000-\$0FFF angeschlossen werden. So haben Sie 4K Byte RAM-Speicher zum Arbeiten und 8K Byte RAM-Speicher als Bildpuffer zur Verfügung.

Beachten Sie, daß die weißen Punkte auf den Speicheraufteilungssteckern in Richtung zum Tastenfeld liegen (Anschluß 1 wird durch den weißen Punkt markiert) müssen.

Es gibt ein Problem bei Apples mit der Platine der Version 0 und 20K oder 24K RAM-Speicher. In diesen Systemen tritt der 8K-Speicherbereich von \$4000 bis \$5FFF nochmals als Bereich von \$6000 bis \$7FFF auf, auch wenn der Bereich \$6000 bis \$7FFF selbst mit RAM-Bausteinen bestückt ist. So scheinen Systeme mit nur 20K oder 24K RAM-Speicher 24K oder 36K Speicher zu haben, aber dieser zusätzliche RAM-Speicher ist tatsächlich nicht vorhanden. Dieses Problem besteht nicht mehr beim Apple mit der Platinen-Version 1, die fast alle von Ihnen besitzen werden.

Festwert-Speicher (ROM)

Der Apple kann 2K (2.048 Bytes) bis 12K (12.228 Bytes) Festwert-Speicher auf seiner Hauptplatine aufnehmen. Dieser ROM-Speicherbereich kann den System-Monitor, verschiedene Dialekte der BASIC-Programmiersprache, einige System- und Hilfsprogramme oder Unterprogramme, wie es sie im Apple Programmer's Aid #1 Rom gibt, enthalten.

Der Apple ROM-Speicherbereich liegt ab Adresse \$D000 in den oberen 12K (48 Seiten) des Hauptspeichers. In den Obersten Speicherstellen benötigt der Apple einige Festwert-Speicherzellen. Beim Einschalten des Apple muß der Mikroprozessor ein Programm zum Ausführen haben. Die Anfangsadresse dieses Programms findet er in den beiden obersten Speicherstellen. Im Apple liegt diese Adresse und das Programm im selben ROM-IC. Dieses Programm initialisiert den Apple und macht es ihm möglich, damit zu arbeiten. (Die Beschreibung des Start-Vorgangs finden Sie auf Seite 39 unter „RESET-Vorgang“).

Die Tabelle 17 zeigt den Apple-Rom-Speicherbereich mit den darin enthaltenen Programmen und Programmpaketen.

| Tabelle 17: ROM-Aufteilung und Programme | | | |
|--|------|----------------------|--------------------------|
| Seiten-Nummer | | benutzt von: | |
| Dezimal | Hexa | | |
| 208 | \$D0 | Programmer's Aid 1 | Applesoft II BASIC |
| 212 | \$D4 | | |
| 216 | \$D8 | | |
| 220 | \$DC | | |
| 224 | \$E0 | Integer BASIC | |
| 228 | \$E4 | | |
| 232 | \$E8 | | |
| 236 | \$EC | | |
| 240 | \$F0 | | |
| 244 | \$F4 | Hilfs-Unterprogramme | |
| 248 | \$F8 | Monitor-ROM | Autostart-ROM |
| 252 | \$FC | | |

Es gibt sechs 24-füßige IC-Sockel auf der Apple-Hauptplatine, die die ROMs aufnehmen. Jeder Sockel kann einen 2048 x 8 Bit ROM des Typs 9316B enthalten. Der Rom ganz links auf der Apple-Hauptplatine enthält die Speicherzellen für den obersten 2K-Speicherbereich des Apple. Der ROM-IC ganz rechts enthält den ROM-Speicherbereich ab Seite \$D0. Wenn ein Sockel nicht von einem ROM besetzt ist, sind die Werte des zugehörigen Speicherbereichs nicht berechenbar.

Die Apple Firmware-Karte kann alle oder einige der ROMs auf der Apple-Platine abschalten und sie durch eigene ROMs ersetzen. Haben Sie eine Apple Firmware-Karte in irgendeinen Einsteckschlitze (slot) eingesetzt, können Sie die ROMs auf der Apple-Platine unwirksam machen, indem Sie den Kontrollschalter auf der Karte auf „UP“ stellen und **RESET** geben oder auf Adresse \$C080 (dezimal 49.280 oder - 16.256) zugreifen. Legen Sie den Kontrollschalter in die DOWN-position und geben **RESET**, oder greifen Sie auf Adresse \$C081 (dezimal 49.281 oder - 16.255) zu, so werden die ROMs auf der Apple-Platine wieder angeschaltet. Weitere Informationen entnehmen Sie dem Anhang A des Applesoft II BASIC Programmierhandbuchs.

Ein- / Ausgabe-Adressen

4096 Speicherstellen (16 Seiten) sind im Apple für Ein-/Ausgabe-Funktionen bestimmt. Dieser 4K-Bereich beginnt ab Adresse \$C000 (dezimal 49.152 oder - 16.384) und erstreckt sich bis Adresse \$CFFF (dezimal 53.247 oder -12.289). Da diese Funktionen etwas kompliziert sind, werden sie in einem eigenen Kapitel behandelt. Siehe Kapitel 5 : weitere Informationen über die Aufteilung der Ein-/Ausgabe-Adressen.

Verwendung der Seite Null

Tabelle 18: Belegung der Seite Null durch den Monitor

| Dezimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Hexa | \$0 | \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 | \$8 | \$9 | \$A | \$B | \$C | \$D | \$E | \$F |
| 0 | \$00 | | | | | | | | | | | | | | | |
| 16 | \$10 | | | | | | | | | | | | | | | |
| 32 | \$20 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 48 | \$30 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 64 | \$40 | * | * | * | * | * | * | * | * | * | | | | | * | * |
| 80 | \$50 | * | * | * | * | * | | | | | | | | | | |
| 96 | \$60 | | | | | | | | | | | | | | | |
| 112 | \$70 | | | | | | | | | | | | | | | |
| 128 | \$80 | | | | | | | | | | | | | | | |
| 144 | \$90 | | | | | | | | | | | | | | | |
| 160 | \$A0 | | | | | | | | | | | | | | | |
| 176 | \$B0 | | | | | | | | | | | | | | | |
| 192 | \$C0 | | | | | | | | | | | | | | | |
| 208 | \$D0 | | | | | | | | | | | | | | | |
| 224 | \$E0 | | | | | | | | | | | | | | | |
| 240 | \$F0 | | | | | | | | | | | | | | | |

Tabelle 19: Belegung der Seite Null durch Applesoft II BASIC

| Dezimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Hexa | \$0 | \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 | \$8 | \$9 | \$A | \$B | \$C | \$D | \$E | \$F |
| 0 | \$00 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 16 | \$10 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 32 | \$20 | | | | | | | | | | | | | | | |
| 48 | \$30 | | | | | | | | | | | | | | | |
| 64 | \$40 | | | | | | | | | | | | | | | |
| 80 | \$50 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 96 | \$60 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 112 | \$70 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 128 | \$80 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 144 | \$90 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 160 | \$A0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 176 | \$B0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 192 | \$C0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 208 | \$D0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 224 | \$E0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 240 | \$F0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |

Tabelle 20: Belegung der Seite Null durch Apple DOS 3.2

| Dezimal | Hexa | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | \$0 | \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 | \$8 | \$9 | \$A | \$B | \$C | \$D | \$E | \$F |
| 0 | \$00 | | | | | | | | | | | | | | | | |
| 16 | \$10 | | | | | | | | | | | | | | | | |
| 32 | \$20 | | | | | | | * | * | | | * | * | * | * | * | * |
| 48 | \$30 | | | | | | * | * | * | * | * | | | | | * | * |
| 64 | \$40 | * | * | * | * | * | * | * | * | * | | * | * | * | * | | |
| 80 | \$50 | | | | | | | | | | | | | | | | |
| 96 | \$60 | | | | | | | | * | * | * | * | | | | | * |
| 112 | \$70 | * | | | | | | | | | | | | | | | |
| 128 | \$80 | | | | | | | | | | | | | | | | |
| 144 | \$90 | | | | | | | | | | | | | | | | |
| 160 | \$A0 | | | | | | | | | | | | | | | | * |
| 176 | \$B0 | * | | | | | | | | | | | | | | | |
| 192 | \$C0 | | | | | | | | | | | * | * | * | * | | |
| 208 | \$D0 | | | | | | | | | * | | | | | | | |
| 224 | \$E0 | | | | | | | | | | | | | | | | |
| 240 | \$F0 | | | | | | | | | | | | | | | | |

Tabelle 21: Belegung der Seite Null durch Integer BASIC

| Dezimal | Hexa | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | \$0 | \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 | \$8 | \$9 | \$A | \$B | \$C | \$D | \$E | \$F |
| 0 | \$00 | | | | | | | | | | | | | | | | |
| 16 | \$10 | | | | | | | | | | | | | | | | |
| 32 | \$20 | | | | | | | | | | | | | | | | |
| 48 | \$30 | | | | | | | | | | | * | * | * | * | | |
| 64 | \$40 | | | | | | | | | | | * | * | * | * | | |
| 80 | \$50 | | | | | | * | * | * | * | * | * | * | * | * | * | * |
| 96 | \$60 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 112 | \$70 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 128 | \$80 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 144 | \$90 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 160 | \$A0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 176 | \$B0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 192 | \$C0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 208 | \$D0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 224 | \$E0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 240 | \$F0 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |

KAPITEL 5

Die Organisation der Ein- und Ausgabe

- 82 Eingebaute Ein-/Ausgabe-Funktionen
- 84 Peripherie-Ein-/Ausgabe
- 84 Ein-/Ausgabebereich für Geräte-Anschlußkarten (Peripherie-Karten)
- 84 ROM-Speicherplatz für Ein-/Ausgabe-Geräte
- 85 Ein-/Ausgabe Programmierungshinweise
- 87 Peripherie-Zwischenspeicher
- 87 Die CSW-/KSW-Verzweigungsadressen
- 88 Erweiterungs-ROM

Die Ein-/Ausgabe-Funktionen des Apple gliedern sich in zwei Kategorien : zum einen Funktionen, die auf der Apple-Hauptplatine realisiert sind und zum anderen die Funktionen, die durch Interface-Karten ausgeführt werden, die in den Geräteschlitzen des Apple stecken. Beide Funktionstypen kommunizieren mit dem Mikroprozessor und Ihren Programmen über 4096 Adressen im Speicherbereich des Apple. Dieses Kapitel beschreibt die Speicheraufteilung und die Funktionsweise der verschiedenen Ein- und Ausgabegeräte. Die Hardware, die dies leistet, wird im folgenden Kapitel beschrieben.

Eingebaute Ein-/Ausgabe-Funktionen

Die meisten der eingebauten Ein-/Ausgabemöglichkeiten sind kurz in Kapitel 1 „Erste Annäherung an den Apple“ beschrieben.

Die Ein-/Ausgabefunktionen auf der Platine des Apple werden von 128 Speicheradressen im Bereich von Adresse \$C000 bis \$C07F (dezimal 49152 bis 49279 oder von -16384 bis -16257) kontrolliert. 27 verschiedene Funktionen teilen sich diesen Bereich. Offensichtlich werden einige Funktionen von mehr als einer Adresse angesprochen : in einigen Fällen können 16 verschiedene Adressen exakt dieselbe Funktion ausführen. Diese 128 Adressen können in fünf Typen eingeteilt werden : Dateneingänge, Signalimpulse (strobes), Programmschalter, Wechselschalter und duale Eingänge (flag inputs).

Dateneingänge. Der einzige Dateneingang auf der Apple-Platine ist die Adresse, deren Wert den aktuellen Zustand der eingebauten Tastatur des Apple anzeigt. Das höchste Bit (Bit 7) stellt einen dualen Eingang (siehe unten) dar. Die unteren sieben Bits geben den ASCII-Code der Taste wieder, die als letzte betätigt wurde.

Duale Eingänge. Der Zustand der meisten internen Eingangsspeicherzellen des Apple wird durch ein einziges Bit dargestellt. Dieses Bit erscheint immer in der höchsten Position dieses Bytes (Bit 7). Diese dualen Eingänge können nur zwei Zustände annehmen : „an“ und „aus“. Der jeweilige Zustand kann leicht von jeder Sprache ermittelt werden. Eine höhere Programmiersprache kann einen „PEEK“-Befehl oder ein entsprechendes Kommando verwenden, um den Wert eines dualen Eingangs zu lesen. Ist dieser Wert größer oder gleich 128, so ist der Zustand dieses Eingangs „an“. Ist der Wert kleiner als 128, ist der Zustand „aus“.

Maschinenprogramme können die Inhalte eines dualen Eingangs in ein internes 6502-Register laden (oder den BIT-Befehl verwenden) und unabhängig vom Zustand des Negativ-Bits (Zustands-Bit N des Prozessors) verzweigen. Ein BMI-Befehl (Sprung bei „Minus“) führt zu einer Verzweigung, wenn das Bit N gesetzt ist, und ein BPL-Befehl (Sprung bei „Plus“) führt zu einer Verzweigung, wenn das Bit N gelöscht ist.

Die Ein-Bit-Eingänge (Drucktasten des Spielanschlusses), der Kassetteneingang, der Tastenimpulseingang und die Spielsteuereingänge sind alle von diesem Typ.

Impulsausgänge. Der Impulsausgang des Spielanschlusses, der Tastenlöschimpuls und der Spielsteuer-rücksetzimpuls werden alle von Adressen gesteuert. Liest Ihr Programm den Inhalt einer dieser Adressen, wird die Funktion ausgelöst, die an diese Adresse angeschlossen ist. Im Falle des Impulsausgangs am Anschluß 5 des Spielanschlusses fällt die Spannung von +5 V auf 0 V für die Dauer von 0,98 Mikrosekunden und geht dann auf +5 V zurück. Beim Tastenlöschimpuls wird der duale Eingang der Tastatur (siehe oben) auf „aus“ gestellt, und im Falle des Spielsteuerrücksetzimpulses werden alle dualen Eingänge der Spielsteuerung auf „aus“ gestellt und die Zeitmess-Schleifen zurückgesetzt.

Ihr Programm kann auch die Tastenfeld- und Spielsteuerrücksetzimpulse auslösen, indem zu den entsprechenden Adressen geschrieben wird, aber Sie sollten nicht durch einen Schreibvorgang auf den Impulsausgang des Spielanschlusses zugreifen, andernfalls werden zwei Impulse von 0,98 Mikrosekunden Dauer im Abstand von 24,43 Nanosekunden abgegeben. Dies liegt an der Methode des 6502-Mikroprozessors, bei einem Schreibvorgang erst den Inhalt dieser Speicherstelle zu lesen und ihn danach zu überschreiben. Diese doppelte Auslösung bleibt bei den Tastenfeld- und Spielsteuerrücksetzimpulsen unbemerkt, kann aber bei dem Impulsausgang des Spielanschlusses Probleme verursachen.

Wechselschalter. Zwei andere Impulsausgänge sind intern an zwei Flip-Flops angeschlossen (Jeder Flip-Flop hat zwei Zustände, die z.B. als „an“ und „aus“ interpretiert werden können). Immer wenn Sie auf die Adresse zugreifen, die diesem Impulsausgang zugeordnet ist, wechselt der Flip-Flop in den anderen Zustand. Diese Wechselschalter versorgen den Kassettenausgang und den eingebauten Lautsprecher.

Es gibt praktisch keinen Weg, den Zustand eines internen Wechselschalters festzustellen. Die Arbeitsweise der Wechselschalter erfordert, daß Sie nur durch Lesevorgänge auf ihre Steueradressen zugreifen und nicht durch Schreibvorgänge (siehe oben : Impulsausgänge).

Programmschalter. Programmschalter sind Schalter mit zwei Positionen. Jede der beiden Zustände wird durch eine eigene Speicheradresse ausgelöst. Greifen Sie auf die eine Adresse zu, die dem Schalter zugeordnet ist, so wird der Schalter auf den einen Zustand gesetzt, greifen Sie auf die andere Adresse zu, so wird der andere Zustand ausgelöst. Bei jedem Zugriff wird der Schalter ohne Rücksicht auf seinen vorherigen Zustand in die gewünschte Position gesetzt. Es gibt keine Möglichkeit festzustellen, in welchem Zustand sich ein Programmschalter befindet. Sie können ohne Bedenken Programmschalter durch Schreibvorgänge betätigen. Die zwei Impulse haben den gleichen Effekt wie ein einzelner. Die Signalausgänge des Spielanschlusses und die Bildmodusschalter (für die verschiedenen Text- und Graphik-Zustände) werden alle durch Programmschalter betätigt.

Anordnung der Adressen, die die internen Ein-/Ausgabefunktionen kontrollieren :

| Tabelle 22: Eingebaute Ein-/Ausgabe-Adressen | | | | | | | | | | | | | | | | |
|--|------------------------------------|-----|-----|-----|-----|-----|-----|-----|-------------------|------|------|------|-----|-----|-----|-----|
| | \$0 | \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 | \$8 | \$9 | \$A | \$B | \$C | \$D | \$E | \$F |
| \$C000 | Tastendruckspeicher | | | | | | | | | | | | | | | |
| \$C010 | Tastenlöschimpuls | | | | | | | | | | | | | | | |
| \$C020 | Kassettenausgang | | | | | | | | | | | | | | | |
| \$C030 | Lautsprecher | | | | | | | | | | | | | | | |
| \$C040 | Impulsausgang des Spielanschlusses | | | | | | | | | | | | | | | |
| \$C050 | gr | tx | un | mi | er | zw | lo | hi | sig0 | sig1 | sig2 | sig3 | | | | |
| \$C060 | ke | d1 | d2 | d3 | a0 | a1 | a2 | a3 | wie \$C060-\$C067 | | | | | | | |
| \$C070 | Spielsteuerrücksetzimpuls | | | | | | | | | | | | | | | |

Erläuterung der Abkürzungen :

| | | | |
|-----|----------------------------|----|---------------------------------|
| gr | setze Graphik-Zustand | tx | setze Text-Zustand |
| un | nur Text oder nur Graphik | mi | setze Text und Graphik gemischt |
| er | Anzeige der ersten Seite | zw | Anzeige der zweiten Seite |
| lo | Anzeige auf Lo-Res Graphik | hi | Anzeige auf Hi-Res Graphik |
| sig | Signalausgänge | d | Drucktasteneingänge |
| a | Analogeingänge | ke | Kassetteneingang |

Peripherie-Ein-/Ausgabe

Hinten auf der Apple-Platine liegt eine Reihe von acht Eingabeschlitzen oder auch Peripherie-Anschlüssen. In sieben dieser Anschlüsse können Sie beliebige Peripherie-Karten stecken, die speziell für den Apple konstruiert sind. Um diese Peripheriekarten einfacher und vielseitiger zu gestalten, ist ein 280 Byte-Speicherbereich für jeden dieser sieben Peripherie-Anschlüsse vorgesehen. Es gibt auch einen 2K Byte großen gemeinsamen Bereich, auf den alle Peripheriekarten zugreifen können.

Jeder Peripherie-Anschluß hat seine eigene Nummer. Der Einsteckschlitz ganz links hat die Nummer 0 und der ganz rechts die Nummer 7. Einsteckschlitz 0 ist für besondere Aufgaben reserviert : für RAM, ROM oder Interface-Erweiterungen. Alle anderen Einsteckschlitze (1 bis 7) haben spezielle Steueranschlüsse, die jeweils zu bestimmten Zeiten aktiviert werden.

Ein-/Ausgabebereich für Geräte-Anschlusskarten (Peripherie-Karten)

Jedem Einsteck-Anschluß sind 16 Speicherplätze für Ein- und Ausgabezwecke zugeordnet. Der Speicherbereich für Anschluß 0 reicht von \$C080 bis \$C08F, für Anschluß 1 von \$C090 bis \$C09F usw. Jede Peripheriekarte kann über diese Bereiche frei verfügen und über Anschluß 41 (sog. DEVICE SELECT = Geräte-Auswahl) erfahren, ob sie angesprochen ist. Wenn die Spannung an diesem Anschluß auf 0 Volt absinkt, befindet sich die Adresse, die der Mikroprozessor aufruft, irgendwo in dem Bereich der zugeordneten 16 Bytes. Die Peripheriekarte kann dann an den unteren vier Adressanschlüssen feststellen, welche der 16 Adressen aufgerufen wurde.

| | \$0 | \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 | \$8 | \$9 | \$A | \$B | \$C | \$D | \$E | \$F | | |
|--------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|
| \$C080 | Ein-/Ausgabebereiche für Karten-Nummer | | | | | | | | | | | | 0 | | | | | |
| \$C090 | | | | | | | | | | | | | 1 | | | | | |
| \$C0A0 | | | | | | | | | | | | | 2 | | | | | |
| \$C0B0 | | | | | | | | | | | | | 3 | | | | | |
| \$C0C0 | | | | | | | | | | | | | 4 | | | | | |
| \$C0D0 | | | | | | | | | | | | | 5 | | | | | |
| \$C0E0 | | | | | | | | | | | | | 6 | | | | | |
| \$C0F0 | | | | | | | | | | | | | 7 | | | | | |

ROM-Speicherplatz für Ein-/Ausgabe-Geräte

Für jeden Geräteanschluß ist eine 256 Byte-Seite (page) reserviert. Diese Seite wird normalerweise gebraucht, um dort 256 Bytes in ROM oder programmierbaren ROM (PROM) zu beherbergen, um diese

Zusatzgeräte zu steuern. So können die Peripherie-Karten als „intelligent“ bezeichnet werden; sie enthalten ihre eigenen Steuerprogramme. Man braucht dann keine Programme zu laden, um diese Geräte zu steuern.

Die Seite im Speicher, die für jeden Geräteanschluß reserviert ist, hat die Anfangsadresse \$Cn, wobei n die Geräteanschlußnummer ist. Geräteanschluß 0 hat keinen derart reservierten Speicherplatz, daher können die meisten Apple-Anschlußkarten dort nicht gebraucht werden. Das Signal am Steckkontakt 1 (sog. I/O-SELECT = Ein-/Ausgabe-Anwahl) der Geräteanschlüsse wird aktiv (Spannung fällt von +5 V auf 0 V), wenn der Mikroprozessor sich auf eine Adresse bezieht, die in der reservierten Seite des angeschlossenen Geräts liegt. Die Gerätekarten können dieses Signal dazu benutzen, die PROM's freizugeben und über die unteren acht Adreßanschlüsse jedes Byte im PROM anzusprechen.

Tabelle 24: PROM-Bereiche für Ein-/Ausgabe-Gerätekarten

| | \$0 | \$1 | \$2 | \$3 | \$4 | \$5 | \$6 | \$7 | \$8 | \$9 | \$A | \$B | \$C | \$E | \$F | |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------------|
| \$C1 | | | | | | | | | | | | | | | | 1 |
| \$C2 | | | | | | | | | | | | | | | | 2 |
| \$C3 | | | | | | | | | | | | | | | | 3 |
| \$C4 | | | | | | | | | | | | | | | | } 4 5 6 7 |
| \$C5 | | | | | | | | | | | | | | | | |
| \$C6 | | | | | | | | | | | | | | | | |
| \$C7 | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

PROM-Speicherbereich für Karten-Nummer

Ein-/Ausgabe Programmierungshinweise

Die Programme in den PROM's der Peripheriekarten sollten verschiebbar sein, d.h. lageunabhängig davon funktionieren, wo sie sich im Speicherbereich des Apple befinden. Absolute Bezüge auf Adressen innerhalb der eigenen Programme sind möglichst zu vermeiden. Dies erreichen Sie, wenn Sie alle Sprünge mit absoluter Adressangabe (jump) durch bedingte oder erzwungene Relativsprünge (branch) ersetzen.

Natürlich können Sie aber auch PROM's mit Unterprogrammen benutzen, die nicht lageunabhängig sind. Der einzige Nachteil, der Ihnen daraus erwächst, ist, daß Ihre Gerätekarte auf einen bestimmten Einsteck-Anschluß festgelegt ist. Dadurch lassen sich allerdings viele Bytes sparen, wenn der Speicherplatz im PROM einer Gerätekarte knapp werden sollte.

Achten Sie vor allem darauf, daß alle Werte der internen Register der 6502-Zentraleinheit gerettet werden. Dazu gibt es ein Unterprogramm im Apple-Monitor-ROM mit dem Namen „IOSAVE“. Es rettet die Inhalte aller Register und schreibt sie in der Reihenfolge A-X-Y-P-S in die Speicherplätze \$45 bis \$49. Die Aufrufadresse ist \$FF4A. Für den umgekehrten Fall gibt es ab Adresse \$FF3F ein Unterprogramm „IORESTORE“, das alle Inhalte der internen Register von diesen Speicherplätzen wieder zurückholt. Rufen Sie am Ende Ihrer PROM-Unterprogramme dieses Unterprogramm auf.

Die Ein-/Ausgabe einzelner Zeichen geschieht meist durch den Akkumulator des 6502-Mikroprozessors. Während der Ausgabe ist das höchste Bit des auszugebenden Zeichens im Akkumulator gesetzt. Das Unterprogramm, das das Zeichen von Eingabegeräten in den Akkumulator lädt, hat bei der Eingabe ebenfalls das höchste Bit gesetzt.

Ein Programm in einem PROM einer Gerätekarte kann feststellen, in welchen Anschluß die Karte gesteckt worden ist. Dazu dient die Befehlsfolge :

```

0300-    20 4A FF    JSR    $FF4A
0303-    78          SEI
0304-    20 58 FF    JSR    $FF58
0307-    BA          TSX
0308-    BD 00 01    LDA    $0100
030B-    29 0F          AND    #0F
0310-    A8          TAY

```

Nach Ausführung dieser Schritte steht die Anschluß-Nummer n im Index-Register Y im Format \$0n. Das Programm kann diesen Wert nutzen, indem es ihn viermal nach links schiebt, um \$n0 zu erhalten.

```

0311-    98          TYA
0312-    0A          ASL
0313-    0A          ASL
0314-    0A          ASL
0315-    0A          ASL
0316-    AA          TAX

```

Ein Programm kann diese Zahl im Indexregister X nutzen, um mit einem Befehl im indizierten Adreßmodus auf die 16 Ein-/Ausgabeadressen zuzugreifen, die für jede Karte reserviert sind.

Beispielsweise lädt der Befehl

```

0317-    BD 80 C0    LDA    $C080,X

```

den Inhalt der ersten Ein-/Ausgabeadresse in den Akkumulator, die von der Peripheriekarte benutzt wird. Die Adresse \$C080 ist die Basisadresse für die erste Adresse, die von allen acht Peripherieanschlüssen gebraucht wird. Die Adresse \$C081 ist Basisadresse für die zweite Ein-/Ausgabeadresse usw. Die Basisadressen auf jeder Karte sind :

| Tabelle 25: Ein-/Ausgabe-Basisadressen | | | | | | | | |
|--|--------|------------------|--------|--------|--------|--------|--------|--------|
| Basis-Adresse | Ø | Einsteckschlitze | | | | | | |
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| \$C080 | \$C080 | \$C090 | \$C0A0 | \$C0B0 | \$C0C0 | \$C0D0 | \$C0E0 | \$C0F0 |
| \$C081 | \$C081 | \$C091 | \$C0A1 | \$C0B1 | \$C0C1 | \$C0D1 | \$C0E1 | \$C0F1 |
| \$C082 | \$C082 | \$C092 | \$C0A2 | \$C0B2 | \$C0C2 | \$C0D2 | \$C0E2 | \$C0F2 |
| \$C083 | \$C083 | \$C093 | \$C0A3 | \$C0B3 | \$C0C3 | \$C0D3 | \$C0E3 | \$C0F3 |
| \$C084 | \$C084 | \$C094 | \$C0A4 | \$C0B4 | \$C0C4 | \$C0D4 | \$C0E4 | \$C0F4 |
| \$C085 | \$C085 | \$C095 | \$C0A5 | \$C0B5 | \$C0C5 | \$C0D5 | \$C0E5 | \$C0F5 |
| \$C086 | \$C086 | \$C096 | \$C0A6 | \$C0B6 | \$C0C6 | \$C0D6 | \$C0E6 | \$C0F6 |
| \$C087 | \$C087 | \$C097 | \$C0A7 | \$C0B7 | \$C0C7 | \$C0D7 | \$C0E7 | \$C0F7 |
| \$C088 | \$C088 | \$C098 | \$C0A8 | \$C0B8 | \$C0C8 | \$C0D8 | \$C0E8 | \$C0F8 |
| \$C089 | \$C089 | \$C099 | \$C0A9 | \$C0B9 | \$C0C9 | \$C0D9 | \$C0E9 | \$C0F9 |
| \$C08A | \$C08A | \$C09A | \$C0AA | \$C0BA | \$C0CA | \$C0DA | \$C0EA | \$C0FA |
| \$C08B | \$C08B | \$C09B | \$C0AB | \$C0BB | \$C0CB | \$C0DB | \$C0EB | \$C0FB |
| \$C08C | \$C08C | \$C09C | \$C0AC | \$C0BC | \$C0CC | \$C0DC | \$C0EC | \$C0FC |
| \$C08D | \$C08D | \$C09D | \$C0AD | \$C0BD | \$C0CD | \$C0DD | \$C0ED | \$C0FD |
| \$C08E | \$C08E | \$C09E | \$C0AE | \$C0BE | \$C0CE | \$C0DE | \$C0EE | \$C0FE |
| \$C08F | \$C08F | \$C09F | \$C0AF | \$C0BF | \$C0CF | \$C0DF | \$C0EF | \$C0FF |

Ein-/Ausgabe-Adressen

Peripherie-Zwischenspeicher

Für jeden der acht Peripherie-Anschlüsse sind 8 Speicherzellen im RAM-Bereich reserviert. Diese 64 Adressen befinden sich in den Speicherseiten \$04 bis \$07 in dem Bereich, der für die Anzeige von Text und Blockgraphik reserviert ist. Die Inhalte dieser Speicherstellen werden jedoch nicht auf dem Bildschirm angezeigt und nicht von normalen Bildschirmoperationen* verändert. Die Peripheriekarten können diese Speicherstellen als Zwischenspeicher benutzen. Diese Zwischenspeicheradressen haben folgende Aufteilung :

| Basis-Adresse | Anschlußnummer | | | | | | |
|---------------|----------------|--------|--------|--------|--------|--------|--------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| \$0478 | \$0479 | \$047A | \$047B | \$047C | \$047D | \$047E | \$047F |
| \$04F8 | \$04F9 | \$04FA | \$04FB | \$04FC | \$04FD | \$04FE | \$04FF |
| \$0578 | \$0579 | \$057A | \$057B | \$057C | \$057D | \$057E | \$057F |
| \$05F8 | \$05F9 | \$05FA | \$05FB | \$05FC | \$05FD | \$05FE | \$05FF |
| \$0678 | \$0679 | \$067A | \$067B | \$067C | \$067D | \$067E | \$067F |
| \$06F8 | \$06F9 | \$06FA | \$06FB | \$06FC | \$06FD | \$06FE | \$06FF |
| \$0778 | \$0779 | \$077A | \$077B | \$077C | \$077D | \$077E | \$077F |
| \$07F8 | \$07F9 | \$07FA | \$07FB | \$07FC | \$07FD | \$07FE | \$07FF |

Anschluß 0 hat keinen solchen RAM-Zwischenspeicher zur Verfügung. Die Basisadressen werden von dem Apple DOS 3.2 und auch von allen Peripheriekarten benutzt. Einige dieser Adressen haben besondere Funktionen. Adresse \$7F8 enthält die Anschlußnummer (im Format \$Cn) der Peripheriekarte, die gerade aktiv ist, und Adresse \$5F8 enthält die Anschlußnummer der Disk-Steuerkarte, von der das aktive DOS jeweils geladen wurde.

Unter Verwendung der Anschlußnummer \$0n, die sich aus dem obigen Programmbeispiel ergibt, kann ein Unterprogramm direkt auf jede seiner acht Zwischenspeicherstellen zugreifen :

| | | | | | |
|-------|----|----|----|-----|----------|
| 031A- | B9 | 78 | 04 | LDA | \$0478,Y |
| 031D- | 99 | F8 | 04 | STA | \$04F8,Y |
| 0320- | B9 | 78 | 05 | LDA | \$0578,Y |
| 0323- | 99 | F8 | 05 | STA | \$05F8,Y |
| 0326- | B9 | 78 | 06 | LDA | \$0678,Y |
| 0329- | 99 | F8 | 06 | STA | \$06F8,Y |
| 032C- | B9 | 78 | 07 | LDA | \$0778,Y |
| 032F- | 99 | F8 | 07 | STA | \$07F8,Y |

Die CSW-/KSW-Verzweigungsadressen

Das Adressenpaar \$36 und \$37 (dezimal 54 und 55) wird CSW genannt („Character output Switch“). Im einzelnen heißt die Adresse \$36 CSWL (CSW Low) und die Adresse \$37 CSWH (CSW High). Dieses Adressenpaar enthält die Adresse des Unterprogramms, das der Apple gerade verwendet, um einzelne Zeichen auszugeben. Diese Adresse ist normalerweise \$DFD0, die Adresse des COUT-Unterprogramms (siehe Seite 32). Das PRINTER-Kommando **CTRL-P** und der BASIC-Befehl PR# können diese Adresse in die Startadresse eines Unterprogramms in einem PROM einer Peripheriekarte ändern. Beide Befehle setzen Adresse \$Cn00 in dieses Speicherstellenpaar, wobei n die Anschlußnummer des Befehls ist. Dies ist die Adresse der ersten Speicherstelle des PROM's auf der Peripheriekarte in dem betreffenden Anschluß.

* Siehe „Textfenster“, Seite 33.

Der Apple ruft dann dieses Unterprogramm jedesmal auf, wenn ein Zeichen ausgegeben werden soll. Das Unterprogramm kann die oben gegebenen Befehlsfolgen benutzen, um seine Anschlußnummer zu finden und die Ein-/Ausgabe- und Zwischenspeicheradressen dieses Anschlusses zu verwenden. Das Unterprogramm kann entweder mit einem RTS-Befehl (Rücksprung aus dem Unterprogramm) zu dem Programm oder zu der Sprache zurückkehren, das diese Ausgabe sendet oder es kann in das COUT-Unterprogramm (Startadresse \$FDF0) springen, um das Zeichen auf dem Bildschirm ausgeben zu lassen und dann erst zu dem aufrufenden Programm zurückzukehren.

Entsprechend enthalten die Adressen \$38 und \$39 (dezimal 56 und 57), KSWL und KSWH genannt (oder beide zusammen KSW genannt), die Startadresse des Unterprogramms, das der Apple gerade benutzt, um einzelne Zeichen einzulesen. Diese Adresse ist normalerweise \$FD1B, die Adresse des KEYIN-Unterprogramms. Das KEYBOARD-Kommando (**CTRL K**) des Monitors und der BASIC-Befehl IN ändern beide diese Adresse zu \$Cn00, wobei n wieder die Anschlußnummer aus dem Befehl ist. Der Apple wird das Unterprogramm am Anfang des PROM's auf der Peripheriekarte in diesem Anschluß aufrufen, wenn er ein einzelnes Zeichen von dem Eingabegerät erwartet. Das Unterprogramm sollte das eingegebene Zeichen in den Akkumulator des 6502-Mikroprozessors laden und mit RTS (Rücksprung aus dem Unterprogramm) das Unterprogramm abschließen. Bei ihm sollte das höchste Bit (Bit 7) des Zeichens gesetzt sein, bevor es beendet wird.

Die Unterprogramme im PROM einer Peripheriekarte können die Adressen der CSW- und KSW-Verzweigungsadressen ändern, so daß diese auf Adressen im PROM zeigen, die nicht die Anfangsadresse sind. Beispielsweise kann ein bestimmter PROM mit einer Befehlsfolge feststellen, in welchem Anschluß er sich befindet und nach einer Initialisierung mit dem aktuellen Zeichenbehandlungsunterprogramm beginnen. Ein Teil dieser Initialisierung könnte KSW oder CSW (je nachdem, ob es Zeichen, ein- oder ausgeben soll) so ändern, daß es auf den Beginn des Zeichenbehandlungs-Unterprogramms weist. Wenn dann der Apple das nächste mal eine Ein- oder Ausgabe von dieser Karte erwartet, überspringt das Unterprogramm die schon ausgeführte Initialisierungsbefehlsfolge und geht direkt zu der anstehenden Aufgabe über. Dies kann in zeitkritischen Situationen die Ausführung erheblich beschleunigen.

Eine Peripheriekarte kann sowohl Eingabe- wie auch Ausgabezwecken dienen, wenn ihr PROM verschiedene Unterprogramme für die verschiedenen Funktionen hat und CSW und KSW entsprechend verändert.

Der Initialisierungsvorgang im PROM einer Peripheriekarte kann feststellen, ob sie für Eingabe oder Ausgabe aufgerufen wurde, indem CSWH und KSWH überprüft werden. Welcher von beiden \$Cn enthält, ruft auch gerade diese Karte auf. Enthalten beide Verzweigungsadressen \$Cn, sollte Ihr Unterprogramm annehmen, daß es zur Ausgabe aufgerufen wurde.

Erweiterungs-ROM

Der 2K Speicherbereich von Adresse \$C800 bis \$CFFF ist reserviert für 2K ROM oder PROM auf einer Peripheriekarte, um große Programme aufzunehmen oder Unterprogramme zu steuern. Der ROM-Erweiterungsbereich hat auch den Vorteil der absoluten Platzierung im Apple Gesamtspeicherbereich, was Ihnen mehr Möglichkeiten beim Schreiben Ihrer eigenen Interface-Programme bietet.

Dieser PROM-Speicherbereich ist für alle Peripherieanschlüsse verfügbar und es können auch mehrere Karten über einen Erweiterungs-ROM für diesen Bereich verfügen, jedoch kann immer nur ein Erweiterungs-ROM zur Zeit aktiv sein.

Jede Gerätekarte mit Erweiterungs-ROM sollte einen Flip-Flop zum Anschalten des ROM's haben. Dieser Flip-Flop sollte von dem DEVICE SELECT-Signal (das den 256 Byte-PROM aktiviert) auf „an“ gestellt werden. Das bedeutet, daß der Erweiterungs-ROM jeder Karte teilweise freigegeben wird, nachdem Sie zum ersten Mal auf die Karte zugegriffen haben. Die andere Freigabe des Erweiterungs-ROM's sollte die I/O STROBE-Leitung (Anschluß 20 jedes Einsteckschlitzes) sein. Die Leitung wird aktiv, wenn der Mikroprozessor des Apple auf eine Adresse im Bereich des Erweiterungs-ROM's zugreift. Wenn diese Leitung aktiv wird und der eben erwähnte Flip-Flop „an“-geschaltet war, dann bezieht sich der Apple auf den Erweiterungs-ROM dieser bestimmten Karte (siehe Schaubild 8).

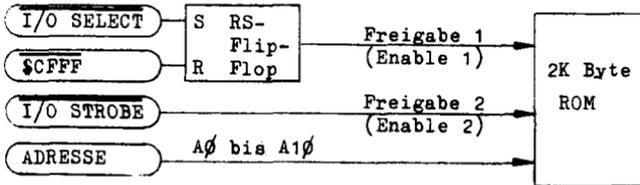


Schaubild 8: Freigabe-Schaltkreis für den Erweiterungs-ROM

Ein 256 Byte-PROM kann den alleinigen Zugriff auf den Erweiterungs-ROM-Bereich erhalten, wenn er auf die Adresse \$CFFF in seinem Initialisierungsvorgang zugreift. Dies ist eine spezielle Adresse und alle Peripheriekarten sollten sie als Signal werten, ihre Flip-Flops auf „aus“ zu stellen und damit ihre Erweiterungs-ROM's zu sperren. Natürlich sperrt das auch den Erweiterungs-ROM der Karte, die diesen ROM beanspruchen möchte, aber der ROM kann wieder freigegeben werden, wenn der Mikroprozessor einen anderen Befehl des 256 Byte-Steuer-PROM's erhält. Nun ist der Erweiterungs-ROM freigegeben. Die Steuerunterprogramme können dann direkt in die Programme des ROM's springen, wo sie sich 2K ungestörten absolut adressierten Speicherbereichs bedienen können :

| | | | | | |
|-------|----|----|----|-----|--------|
| 0332- | 2C | FF | CF | BIT | \$CFFF |
| 0335- | 4C | 00 | C8 | JMP | \$C800 |

Es ist möglich, Schaltkreise auf der Peripheriekarte einzusparen (auf Kosten des ROM-Bereichs), indem man die Spezialadresse \$CFFF nicht vollständig dekodiert. Falls Sie es sich tatsächlich leisten können, auf die letzten 256 Bytes Ihres ROM-Bereichs zu verzichten, kann der folgende einfache Schaltkreis ausreichen :

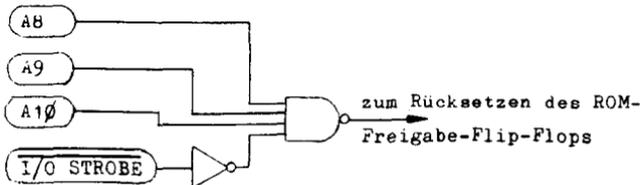


Schaubild 9: \$CFXX Dekodierung

KAPITEL 6

Die Hardware des Apple

- 92 Der Mikroprozessor
- 94 Die Taktsteuerung des Mikroprozessors
- 96 Stromversorgung
- 98 ROM-Speicherbereich
- 99 RAM-Speicherbereich
- 100 Der Video-Generator
- 101 Die Video-Ausgangsanschlüsse
- 102 Eingebaute Ein-/Ausgabefunktionen
- 103 Die „USER 1“-Lötbrücke
- 103 Der Ein/Ausgabeanschluß für Spiele
- 104 Die Tastatur
- 106 Der Tastatur-Anschluß
- 107 Kassettenanschlußbuchsen
- 107 Anschluß der Betriebsspannungen
- 108 Der Lautsprecher
- 108 Peripherie-Anschlüsse

Der Mikroprozessor

| | |
|---------------------|--|
| Modell | : MCS6502/SY6502 |
| Hersteller | : MOS Technology, Inc. Synertek Rockwell |
| Anzahl der Befehle | : 56 |
| Adreßarten | : 13 |
| Akkumulatoren | : 1 (A) |
| Index-Register | : 2 (X,Y) |
| Weitere Register | : Kellerzeiger (S) Prozessor-Status (P) |
| Keller | : 256 Bytes, auch direkt adressierbar |
| Status-Bits | : N (Vorzeichen) C (Übertrag) Z (Null) V (Überlauf bei Vorzeichenarithmetik) |
| Steuer-Bits | : I (Interrupt-Sperre) D (Dezimale Arithmetik) B (Interrupt durch BRK ausgelöst) |
| Interrupts | : 2 (IRQ, NMI) |
| Resets | : 1 (RES) |
| Adressierbereich | : 2^{16} (64K) Adressen |
| Adreß-Bus | : 16 Bit parallel |
| Daten-Bus | : 8 Bit parallel (für beide Datenrichtungen) |
| Spannungsversorgung | : +5 Volt |
| Leistungsaufnahme | : 0,25 Watt |
| Taktfrequenz | : 1,023 MHz |

Der Mikroprozessor bekommt seine Taktsignale, $\Phi 0$ und $\Phi 1$, von den Zeitkreisen, die unten beschrieben werden. $\Phi 0$ und $\Phi 1$ sind komplementäre Taktsignale. Verschiedene Handbücher, auch das MOS Technology Hardware Manual verwenden die Bezeichnung $\Phi 2$ für den $\Phi 0$ -Takt des Apple.

Der Mikroprozessor nutzt seine Adreß- und Datenbusse nur in der Zeitspanne, in der $\Phi 0$ aktiv ist. Während der Phase 1 führt dann der Mikroprozessor interne Operationen aus und braucht die Adress- und Datenbusse nicht.

Der Mikroprozessor hat einen 8-Bit-Datenbus für beide Richtungen und einen 16-Bit-Adreßbus. Die Adreßbusleitungen sind durch drei 8T97 three-state-Puffer (auf den Positionen H3, H4 und H5 auf der Platine) an den Mikroprozessor angeschlossen. Die Adreßleitungen werden nur während eines DMA-Zyklus abgetrennt und sind sonst immer aktiv. Die Adressen auf dem Adreßbus werden nach 300ns gültig, wenn Phase 1 aktiv wird und bleibt noch die ganze Phase 0 hindurch gültig.

Peripherie-Anschlüsse

Anschluß
der
Strom-
versorgung

USER 1
Lötbrücke

Kassetten-
Anschluß-
buchsen

Video-
Anschlüsse

Spiel
Ein-/Ausg.
Anschluß

Lautsprecher-
Anschluß

Tastenfeld-
Anschluß

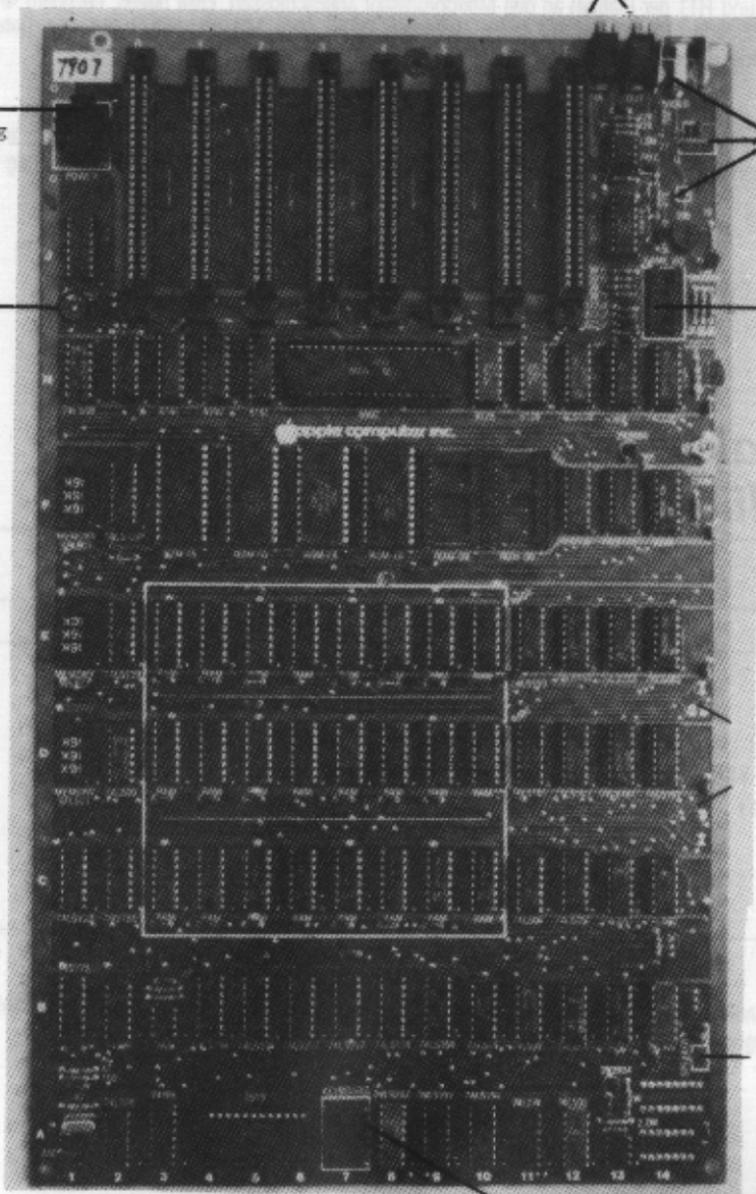


Schaubild 10: Die Hauptplatine des Apple

Der Datenbus ist durch zwei bilaterale (nach beiden Seiten offene) 8T28-three-state-Puffer (auf den Positionen H10 und H11 der Platine) an den Mikroprozessor angeschlossen. Etwa 300ns, nachdem $\Phi 1$ und das READ/WRITE-Signal (R/W) auf 0V fallen, kommen Daten vom Mikroprozessor auf den Datenbus. Ansonsten liest oder ignoriert der Mikroprozessor den Datenbus.

DIE RDY, RES, IRQ und NMI-Leitungen zum Mikroprozessor werden alle durch 3,3 kOhm-Widerstände auf +5V „hoch“ gehalten. Diese Leitungen stehen auch an den Peripherie-Anschlüssen zur Verfügung (siehe Seite 108).

Die SET OVERFLOW (SO)-Leitung ist fest mit Masse verbunden.

Die Taktsteuerung des Mikroprozessors

Tabelle 27: Beschreibung der Taktsignale

| | |
|----------------|---|
| 14M | 14,318MHz-Ausgang des Quartz-Oszillators. Alle Zeitsignale werden aus diesem Signal abgeleitet. |
| 7M | 7,159MHz-Signal (Zeitwischensignal) |
| COLOR REF | 3,580MHz Farbsynchron-Frequenz |
| $\phi 0$ (2) | Phase 0 System-Takt, 1,023MHz, komplementär zu $\phi 1$ |
| $\phi 1$ | Phase 1 System-Takt, 1,023MHz, komplementär zu $\phi 0$ |
| Q3 | Ein asynchrones Universal-Signal der doppelten Taktfrequenz (2,045MHz) |

An allen Peripherie-Anschlüssen stehen die Signale 7M, $\phi 0$, $\phi 1$ und Q3 zur Verfügung. Das 14M- und COLOR REF-Signal ist dort nicht verfügbar.

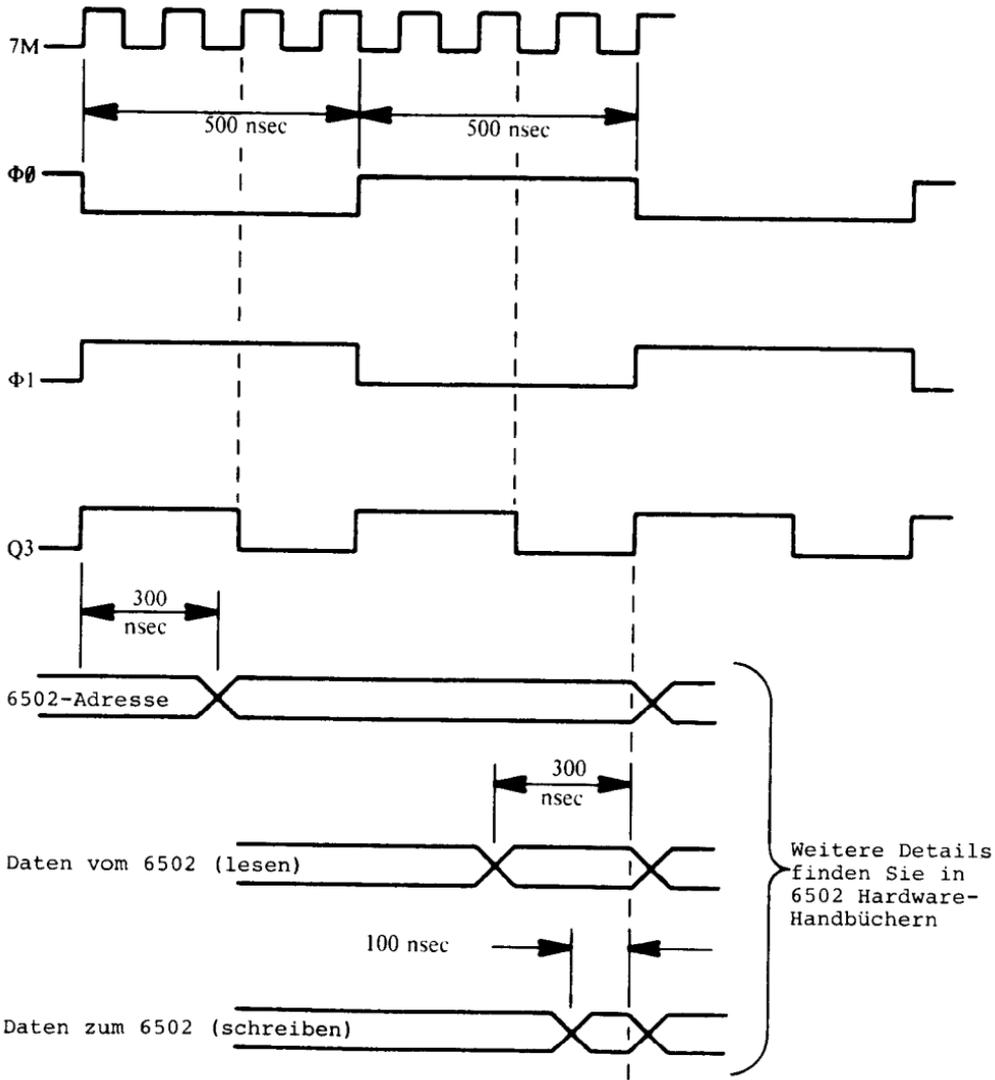


Schaubild 11. Die Taktsignale und ihre Beziehung zueinander

Stromversorgung

Das Apple Netzteil (U.S. Patent 4130862)

| | |
|------------------------|--|
| Eingangsspannung | 107V bis 132V oder 214V bis 264V (Spannungsbereich durch Spannungswahl- schalter einstellbar*) |
| Ausgangsspannungen | +5V +11,8V -12,0V -5,2V |
| Leistungsaufnahme | 60 Watt maximal (volle Leistung) 79 Watt maximal (stoßweise**) |
| Maximale Belastbarkeit | +5V mit 2,5A -5V mit 250mA +12V mit 1,5A (ca. 2,5A stoßweise**) -12V mit 250mA |
| Betriebstemperatur | 55°C |

Das Apple Netzteil ist ein hochfrequent schaltendes Netzteil. Während die meisten anderen Netzteile einen großen Transformator mit vielen Windungen brauchen, um die Eingangsspannung in kleinere Spannungen umzuwandeln und diese dann gleichzurichten und zu regulieren, formt das Netzteil des Apple die anliegende Wechselfrequenz in Gleichspannung um und versorgt damit einen Hochfrequenz-Schwingkreis. Der Ausgang dieses Oszillators liegt an einem kleinen Transformator mit vielen Windungen. Die Spannungen der Sekundärwicklungen werden dann zu den Ausgangsspannungen reguliert.

Der + 5V Spannungsausgang wird mit einer Referenzspannung verglichen und die Spannungsdifferenz reguliert den Schwingkreis. Verläßt die Ausgangsspannung den Toleranzbereich, so wird die Frequenz des Oszillators geändert, damit die Spannung wieder ihr normales Niveau erreicht.

Falls zufällig eine der Ausgangsspannungen des Netzgeräts kurzgeschlossen werden sollte, stoppt ein Rückkopplungsschaltkreis im Netzgerät den Oszillator und schaltet alle Ausgangsschaltkreise ab. Dann wartet das Netzteil ca. 1/2 Sekunde und versucht den Oszillator wieder zu starten. Ist der Ausgang noch immer kurzgeschlossen, wird wieder abgeschaltet und gewartet. Dieser Ablauf wird fortgesetzt, bis der Kurzschluß beseitigt oder der Strom abgeschaltet wird.

Wird der Anschlußstecker des Netzgeräts aus dem Anschluß auf der Apple-Platine gezogen, so merkt das Netzgerät diesen „Leerlauf“-Zustand und schließt sich selbst kurz. Das löst den oben beschriebenen Schutzvorgang aus, der die Ausgangsspannungen abschaltet. So wird ein Schaden in der internen Schaltung des Netzgeräts vermieden.

* Der Spannungswahlschalter ist auf einigen Modellen des Apple nicht vorhanden.

** Das Netzteil kann 20 Minuten ohne Schaden bei stoßweiser Belastung arbeiten, wenn es dann 10 Minuten normal belastet wird.

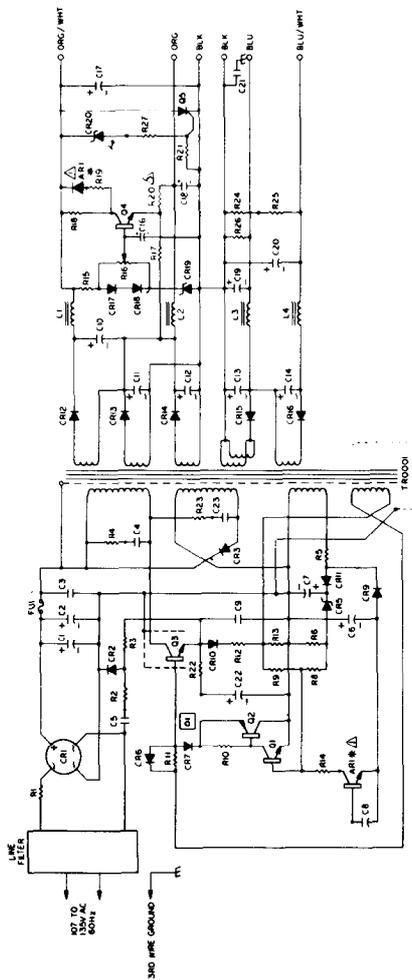


Schaubild 12. Schaltbild des Netzgerätes.

Falls eine der Ausgangsspannungen den erlaubten Bereich verläßt, was auf ein Problem innerhalb oder außerhalb des Netzgeräts zurückzuführen ist, wird das Netzgerät sich abschalten, um die Bauteile auf der Apple-Platine zu schützen. So ist gesichert, daß entweder alle Spannungen korrekt sind oder alle Spannungen abgeschaltet werden.

Liegt eine der oben genannten Fehlerarten vor, so wird der eingebaute Schutzschaltkreis die Schwingungen am Transformator abschalten. Nach einer kurzen Zeit wird das Netzgerät mit den Schwingungen beginnen. Ist die Fehlerbedingung noch nicht beseitigt, so wird das Gerät erneut abgestellt. Dieser Ablauf kann beliebig lange durchgeführt werden, ohne daß das Netzgerät Schaden nimmt. Jedesmal, wenn der Oszillator abgestellt wird und wieder neu startet, durchläuft die Frequenz den hörbaren Bereich. So werden Sie ein gleichmäßiges „Klick klick klick“ aus dem Netzgerät vernehmen und sind gewarnt, daß mit einer Ausgangsspannung etwas nicht in Ordnung ist.

Unter keinen Umständen sollten Sie mehr als 280V an das Netzgerät legen, wenn der Spannungswahlschalter auf 220V steht (oder mehr als 140V, wenn der Schalter auf 110V steht bzw. das Gerät nur für 110V ausgelegt ist). Das hätte dauerhaften Schaden am Netzgerät zur Folge.

Sie sollten Ihren Apple an eine gut geerdete Steckdose anschließen. Es ist für den Apple sehr wichtig, daß er eine gute Erdverbindung hat.

Vorsicht : Es gibt gefährliche Hochspannungen im Gehäuse des Netzteils. Ein großer Teil der internen Schaltung ist nicht von der Versorgungsspannung isoliert und es wird Spezialwerkzeug für jede Arbeit am Netzgerät benötigt. **VERSUCHEN SIE NICHT, IHR NETZGERÄT SELBST ZU REPARIEREN!** Senden Sie es Ihrem Apple-Händler zur Reparatur.

Rom-Speicherbereich

Der Apple kann bis zu sechs 2K x 8 Bit maskenprogrammierte Festwertspeicher-ICs (ROMs) aufnehmen. Einer dieser sechs ROMs wird von einem 74LS138 (in Position F12 auf der Platine des Apple) freigegeben, wenn der Adressbus des Mikroprozessors eine Adresse zwischen \$D000 und \$FFFF hält. Die acht Datenausgänge aller ROMs sind mit den Datenleitungspuffern des Mikroprozessors verbunden, und die Adreßeingänge der ROMs sind an die Puffer der Adreßleitungen A0 bis A10 angeschlossen.

| | | | | |
|-----|----|---|----|------------|
| A7 | 1 | 0 | 24 | +5V |
| A6 | 2 | | 23 | A8 |
| A5 | 3 | | 22 | A9 |
| A4 | 4 | | 21 | <u>CS3</u> |
| A3 | 5 | | 20 | <u>CS1</u> |
| A2 | 6 | | 19 | A10 |
| A1 | 7 | | 18 | CS2 |
| A0 | 8 | | 17 | D7 |
| D0 | 9 | | 16 | D6 |
| D1 | 10 | | 15 | D5 |
| D2 | 11 | | 14 | D4 |
| Gnd | 12 | | 13 | D3 |

Schaubild 13. Anschlüsse des 9316B ROM.

Die ROMs sind zur Freigabe an drei „chip-select“-Leitungen angeschlossen. CS1 und CS3, die beide bei 0V aktiv sind, sind zusammen an den 74LS138 auf Position F12 angeschlossen, der die einzelnen ROMs auswählt. CS2, an das alle ROMs angeschlossen sind, ist mit positiver flanke aktiv und steht als INH an allen Peripherie-Anschlüssen zur Verfügung. Zieht eine der eingesteckten Gerätekarten diese Leitung auf 0V, so werden alle ROMs auf der Apple-Platine gesperrt.

Die ROMs sind den „2316“ und „2716“ PROMs ähnlich. Jedoch sind die Freigabeeingänge der PROMs oft von anderer Polarität, so daß sie nicht direkt in die Platine gesteckt werden können.

RAM-Speicherbereich

Der Apple verwendet 4K und 16K dynamische RAMs für seinen Hauptspeicher. Dieser RAM-Speicherbereich wird sowohl vom Mikroprozessor als auch von der Video-Ausgabe-Schaltung genutzt. Der Mikroprozessor und die Video-Anzeige wechseln sich im Gebrauch des RAM-Bereichs ab: Der Mikroprozessor greift nur in Phase 0 auf die Adressen des RAM-Bereichs zu und der Video-Teil erneuert sein Bild während Phase 1 (Φ1).

Die drei 74LS153 ICs auf E11, E12 und E13, der 74LS283 auf E14 und eine Hälfte des 74LS257 auf C12 bilden den Adressen-Multiplexer für den RAM-Speicherbereich. Sie nehmen die Adressen des Mikroprozessors und des Video-Generators und verteilen sie auf sechs RAM-Adreßleitungen. Die anderen RAM-Adreßsignale, RAS und CAS und das Signal, das Adreßleitung 6 für 16K RAMs und CS für 4K RAMs ist, werden von dem RAM-Auswahl-Schaltkreis erzeugt. Dieser Schaltkreis besteht aus zwei 74LS139 auf E2 und F2, einem halben 74LS153 auf C1, einem und einem halben 74LS257 auf C12 und J1 und den drei RAM-Aufteilungssteckern auf D1, E1 und F1. Dieser Schaltkreis verzweigt die Signale auf jede Reihe von RAM-ICs, abhängig davon, welcher RAM-Typ in dieser Reihe steckt.

In Phase 1 frischt der Video-Schaltkreis die dynamischen RAMs automatisch auf. Da der Bildschirm wenigstens einen 1K-Speicherbereich anzeigt, muß jede Speicherstelle in diesem Bereich fünfzig Mal in der Sekunde angesprochen werden.

Nun wird sogar jedes Bit des gesamten 48K RAM-Bereichs aufgefrischt. Durch die abwechselnden Video- und Mikroprozessor-Zugriffsphasen laufen die Bildausgabe, der Mikroprozessor und die RAM-Auffrischung alle bei „vollem Tempo“ und ohne sich gegenseitig zu stören.

Die Daten-Eingänge der RAM's bekommen ihre Information direkt vom Datenbus. Den Datenausgängen der RAM's werden zwei 74LS174 (auf B5 und B8) nachgeschaltet. Diese eingeklinkten Datenausgänge werden einerseits über die Multiplexer 74LS257 (auf Position B6 und B7) auf den Datenbus gegeben und andererseits direkt an die Zeichen-, Farben- und Punkt-Generatoren gekoppelt.

| | | | | |
|-----------|---|---|----|-----------|
| -5V | 1 | o | 16 | Gnd |
| Daten Ein | 2 | | 15 | CAS |
| R/W | 3 | | 14 | Daten Aus |
| RAS | 4 | | 13 | CS |
| A5 | 5 | | 12 | A2 |
| A4 | 6 | | 11 | A1 |
| A3 | 7 | | 10 | A0 |
| +12V | 8 | | 9 | +5V |

4096 4K-RAM

| | | | | |
|-----------|---|---|----|-----------|
| -5V | 1 | o | 16 | Gnd |
| Daten Ein | 2 | | 15 | CAS |
| R/W | 3 | | 14 | Daten Aus |
| RAS | 4 | | 13 | A6 |
| A5 | 5 | | 12 | A2 |
| A4 | 6 | | 11 | A1 |
| A3 | 7 | | 10 | A0 |
| +12V | 8 | | 9 | +5V |

4116 16K-RAM

Schaubild 14. Anschlüsse der RAM-IC's

Der Video-Generator

Es gibt 192 Abtastzeilen auf dem Bildschirm, die zu 24 Zeilen von je 8 Abtastzeilen gruppiert werden. Jede Abtastzeile zeigt den ganzen oder Teil-Inhalt von 40 Speicherbytes.

Die Schaltung zur Erzeugung des Video-Signals erhält ihre Synchronisations- und Taktsignale von einer 74LS161 Zählerkette auf den Positionen D11 bis D14. Diese Zähler generieren fünfzehn Synchronisationssignale :

H0 H1 H2 H3 H4 H5
V0 V1 V2 V3 V4
VA VB VC

Die „H“-Signal-Familie ist die horizontale Position auf dem Bildschirm, von binär 000000 bis 100111 (dezimal 0 bis 39).

Die Signale V0 bis V4 sind die vertikale Position auf dem Bild, von binär 00000 bis 10111 (dezimal 0 bis 23). Die VA, VB und VC Signale sind die vertikale Abtastzeilenposition innerhalb der vertikalen Bildschirmzeile, von binär 000 bis 111 (dezimal 0 bis 7).

Diese Signale werden zu RAM-Adreß-Multiplexern geschickt, die sie abhängig von der Stellung der Programmschalter für den Bildmodus (siehe unten) in die Adresse einer einzigen RAM-Speicherzelle umformen. Die RAM-Multiplexer senden dann in Phase 1 die Adresse dem RAM-Speicherbereich. Die Daten der in Phase 1 angesprochenen RAM-Speicherzelle werden über die beiden 74LS174 auf B5 und B8 in die Video-Generatoren gegeben. Der 74LS283-Schaltkreis auf E14 ordnet die Speicheradressen so um, daß die Speicheraufteilung auf dem Bildschirm durcheinander gebracht wird.

Soll auf dem aktuellen Bereich des Bildschirms ein Textzeichen stehen, dann gibt der Video-Generator die unteren sechs Bits dieses Zeichens dem 2513-Zeichengenerator auf A5. Die sieben Reihen jedes Zeichens werden von den VA, VB und VC-Signalen abgetastet und die Ausgabe des Zeichengenerators wird durch einen 74166 auf A3 in einen seriellen Punktstrom verwandelt. Dieser Bit-Strom wird durch eine Exklusiv-ODER-Schaltung (1/4 74LS86 auf B2) geführt, wo er invertiert wird, falls Bit 7 des Datenbytes 0 ist und entweder das sechste Bit des Zeichencodes 0 ist oder der 555-Zeitgeber (Position B3) auf + 5V steht. So werden inverse und blinkende Zeichen erzeugt. Der Bit-Strom wird dann dem Video-Datenselektor (siehe unten) übergeben.

Ist der Bildschirm des Apple im Graphik-Zustand, wird das Datenbyte aus dem RAM in zwei 74LS194-Schieberegister gegeben (Position B4 und B9). Hier werden jeweils vier Bits in einen seriellen Datenstrom verwandelt. Diese zwei Datenströme werden ebenfalls dem Video-Datenselektor weitergegeben.

Der 74LS257 Datenselektor auf A8 schaltet zwischen Blockgraphik und hochauflösender Graphik um. Der serielle Hi-Res-Punktstrom wird für einen halben Takt von dem 74LS74 auf A11 verzögert, wenn das höchste Bit des Bytes gesetzt ist. So werden die zwei Farbkombinationen der Hi-Res Graphik realisiert.

Dieser Datenselektor mischt die Datenströme aus den oben genannten Quellen entsprechend der Stellung der Video-Programmschalter.

Der 74LS194 auf der Position A10 und der 74LS151 auf A9 wählen einen der seriellen Datenströme für Text und Blockgraphik oder für Hi-Res Graphik aus. Schließlich wird die endgültige serielle datenausgabe mit dem zusammengesetzten Synchrosignal und dem Farbsynchrosignal gemischt und an die Video-Ausgangsbuchsen gelegt.

Die Programmschalter, die den Bildmodus bestimmen, sind Bestandteil der Ein-/Ausgabefunktionen des Apple. Die logischen Gatter auf den Positionen A11, A12, B11, B12 und B13 stellen diese Programmschalter dar.

Das Farbsynchronsignal wird von den Gattern auf B12, B13 und C13 erzeugt und vom Widerstand R5, der Spule L1, dem Kondensator C2 und dem Trimmkondensator C3 kontrolliert. Mit diesem Trimmkondensator können Sie die Farbtöne auf dem Bildschirm justieren. Transistor Q6 und sein Basiswiderstand R27 schalten das Farbsynchronsignal ab, sobald der Apple Text ausgibt.

Die Video-Ausgangsanschlüsse

Das Videosignal, das von der obengenannten Schaltung erzeugt wird, ist ein positives Bildaustastsynchronsignal, das an jeden Monitor mit direktem Videoeingang oder Studiomonitor angeschlossen werden kann. Diese Signale sind an drei Stellen auf der Platine des Apple verfügbar :

RCA-Buchse. Hinten auf der Hauptplatine, nahe am rechten Ende, ist eine RCA-Standardphonobuchse. Die Hülse dieser Buchse ist an die Masse des Apple angeschlossen und der Kontakt ist über ein 200 Ohm-Potentiometer mit dem Videoausgangssignal verbunden. Die Spannung an diesem Anschluß kann mit dem Potentiometer von 0V bis 1V Spitzenspannung eingestellt werden.

Video-Hilfsanschluß. Rechts auf der Hauptplatine, in der Nähe der Rückwand ist ein Molex KK 100-Anschluß mit vier quadratischen Stiften. Dieser Anschluß ist mit dem Videoausgang und zwei Versorgungsspannungen ausgestattet. Schaubild 15 zeigt diesen Anschluß.

Tabelle 28: Anschlüsse der Video-Vierstiftbuchse

| Anschluß | Name | Beschreibung |
|----------|-------|---|
| 1 | Masse | Gemeinsame Masse des Systems: 0V |
| 2 | Video | Positives Signalgemisch. Der Pegel für schwarz liegt bei etwa 0,75V, der Pegel für weiß bei 2,0V und der Synchronboden (sync tip) ist 0V. Der Ausgangspegel ist nicht einstellbar und nicht gegen Kurzschluß gesichert. |
| 3 | +12V | +12V Versorgungsspannung |
| 4 | -5V | -5V Versorgungsspannung |

Video-Anschlußstift. Dieser einzelne Metallstift für „wire-wrap“-Anschluß unterhalb der Video-Vierstiftbuchse ist an dasselbe Videosignal angeschlossen. Er ist als Verbindungspunkt für die Euroapple PAL/SECAM Farbkarte vorgesehen.

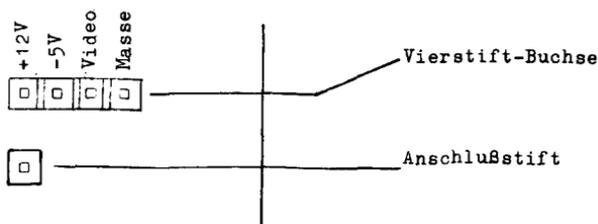


Schaubild 15. Video-Hilfsanschlüsse

Eingebaute Ein-/Ausgabefunktionen

Die eingebauten Ein-/Ausgabefunktionen des Apple sind in 128 Adressen ab Adresse \$C000 abgelegt. Als Ein-/Ausgabewähler (I/O selector) arbeitet ein 74LS138 auf der Position F13 der Hauptplatine. Er entschlüsselt diese speziellen Adressen und ermöglicht die verschiedenen Funktionen.

Der 74LS138 wird durch einen weiteren Baustein dieses Typs (auf H12) geschaltet, wenn der Adreßbus des Apple eine Adresse zwischen \$C000 und \$C0FF enthält. Der Ein-/Ausgabewähler teilt diesen 256 Byte-Bereich in acht 16 Byte-Bereiche ein und ignoriert den Bereich von \$C080 bis \$C0FF. Jeder Ausgang des 74LS138 wird aktiv (OV), wenn der zugehörige Bereich angesprochen wird.

Die Leitung „0“ des E/A-Wählers lenkt die Daten des Tastenfeldes in die RAM-Multiplexer (auf B6 und B7).

Die Leitung „1“ des E/A-Wählers setzt den 74LS74-Flip-Flop auf B10 zurück und löscht damit den Tastaturimpuls.

Die Leitung „2“ triggert einen halben 74LS74 auf K13. Der Ausgang dieses Flip-Flop's ist über einen Spannungsteiler an die Kassettenausgangsbuchse angeschlossen.

Die Leitung „3“ triggert die andere Hälfte des 74LS74 auf K13. Dieser Ausgang ist über einen Kondensator und einem Darlington-Verstärker mit dem Lautsprecheranschluß auf der Platine rechts unter dem Tastenfeld verbunden.

Die Leitung „4“ liegt direkt an Anschluß 5 des Spiel-Ein/Ausgangs. Dieser Anschlußstift ist der Impulsausgang (C040 ST0BE).

Die Leitung „5“ gibt den 74LS259 auf der Position F14 frei. Dieser IC enthält die Programmschalter für die Bildanzeige und die Signalausgänge des Spielanschlusses. Die Programmschalter werden von den Adreßleitungen A1 bis A3 ausgewählt und erhalten ihren Zustand von der Adressleitung A0.

Die Leitung „6“ gibt den 74LS251 acht-Bit-Multiplexer auf H14 frei. Dieser Multiplexer verbindet eine seiner acht Eingangsleitungen mit Bit 7 des Datenbusses (direkt an Anschluß 11 des Puffers 8T28 auf H11 vor der Datenleitung 7 des Mikroprozessors). Die unteren drei Adreßleitungen (Leitung 0 bis 2) bestimmen, welcher der acht Eingänge ausgewählt wird. Vier der Eingänge kommen von dem 558 Vierfach-Zeitgeber auf Position H13. Die Zeitgeber sind an die vier Analogeingänge der Spielsteuerung angeschlossen. Drei weitere Eingänge des Multiplexers sind die drei Ein-Bit (Drucktasten)-Eingänge am Spieleanschluß. Schließlich ist der letzte Eingang des Multiplexers über einen 741-Operationsverstärker auf Position K13 an die Kassetteneingangsbuchse angeschlossen.

Die Leitung „7“ des E/A-Wählers setzt alle vier 558-Zeitgeber auf Position H13 zurück. Die vier Eingänge der Zeitgeber sind über ein RC-Netzwerk aus 0,022µF Kondensatoren und 100 Ohm Widerständen mit den Analogeingängen der Spielsteuerungen verbunden. Die angeschlossenen 150 kOhm-Potentiometer bestimmen mit ihrem veränderbaren Widerstand die Zeitcharakteristik der vier Zeitkreise.

Die „USER 1“-Lötbrücke

Auf der Platine des Apple gibt es links vom Einsteckschlitz 0 ein nicht gekennzeichnetes Paar von Lötstellen, „User 1“-Lötbrücke („Benutzer 1“-Lötbrücke) genannt. Diese Brücke wird auf Photo 8 gezeigt. Wenn Sie diese beiden Lötstellen mit einem Draht verbinden, wird die USER 1-Leitung auf jedem Peripherieanschluß aktiv. Zieht dann eine der Peripheriekarten diese Leitung auf Null, so wird die gesamte interne Dekodierung gesperrt. Die I/O SELECT- und die DEVICE SELECT-Leitungen gehen alle auf + 5V und bleiben es, solange USER 1 auf Null liegt, unabhängig von der Adresse des Adreßbusses.

Die USER 1
Lötbrücke

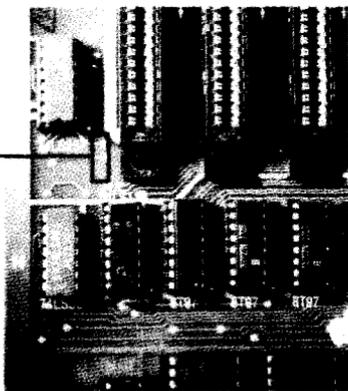


Photo 7. Die USER 1 Lötbrücke.

Der Ein-/Ausgabeanschluss für Spiele

| | | | |
|-------------------------------|---|----|----------------|
| +5V | 1 | 16 | NC |
| SW \emptyset | 2 | 15 | AN \emptyset |
| SW1 | 3 | 14 | AN1 |
| SW2 | 4 | 13 | AN2 |
| \emptyset 4 \emptyset STB | 5 | 12 | AN3 |
| PDL \emptyset | 6 | 11 | PDL3 |
| PDL2 | 7 | 10 | PDL1 |
| GND | 8 | 9 | NC |

Schaubild 16. Spieleanschlußbelegung

Tabelle 29: Beschreibung der Spieleanschlußsignale

| Anschluß | Name | Beschreibung |
|-----------|-----------------------------|--|
| 1 | +5V | +5V Stromversorgung. Der maximale Strom sollte weniger als 100mA betragen. |
| 2-4 | SW ϕ -SW2 | Ein-Bit-Eingänge (Drucktasten). Es sind Standard-TTL-Eingänge der 74LS-Serie. |
| 5 | $\overline{C\phi4\phi}$ STB | Der Impulsausgang ist ein Standard-TTL 74LS-Ausgang. Dieser Anschluß liegt normalerweise an +5V und geht beim Zugriff auf eine Adresse von $\$C\phi4\phi$ bis $\$C\phi4F$ für die Dauer von Phase ϕ auf 0V. |
| 6,7,10,11 | PDL ϕ -PDL2 | Spielsteuereingänge. Diese analogeingänge sollten mit 150 kOhm-Potentiometer an +5V angeschlossen werden. |
| 8 | GND | Elektrische Masse des Systems: 0V. |
| 12-15 | AN ϕ -AN3 | Signal-Ausgänge (Annunciator). Diese Standard-Ausgänge der TTL 74LS-Serie sollten gepuffert werden, falls sie andere als TTL-Eingänge treiben sollen. |
| 9,16 | NC | Kein Anschluß. |

Die Tastatur

Die Tastatur des Apple wird von einem monolithischen Tastenfelddekode-ROM MM5740 ausgewertet. Die Eingänge dieses ROM's, die Anschlüsse 4 bis 12 und 22 bis 31, sind mit der Tastenmatrix verbunden. Die Ausgänge werden mit einem 7404 gepuffert und sind mit dem Tastatur-Anschluß auf der Platine verbunden (siehe unten).

Der Tastaturdekode fragt sehr schnell die Tastenmatrix ab, um gedrückte Tasten zu finden. Dieser Abtastvorgang wird von einem freilaufenden Oszillator aus drei Teilen eines 7400 (Position U4 auf der Tastenfeldplatine) gesteuert. Die Frequenz des Oszillators wird von C6, R6 und R7 auf der Tastenfeldplatine bestimmt.

Die **REPT**-Taste der Tastatur ist an einen 555-Zeitgeber (Position U3 auf der Tastenfeldplatine) angeschlossen. Dieser IC generiert mit einem Kondensator und drei Widerständen ein 10 Hz-„REPeAT“-Signal.

Wird der 220 kOhm-Widerstand R3 durch einen Widerstand geringeren Wertes ersetzt, wird die **REPT**-Taste die Zeichen schneller wiederholen. Sehen Sie dazu das Schaltbild und die Entschlüsselung der Tasten auf dem Schaubild 17.

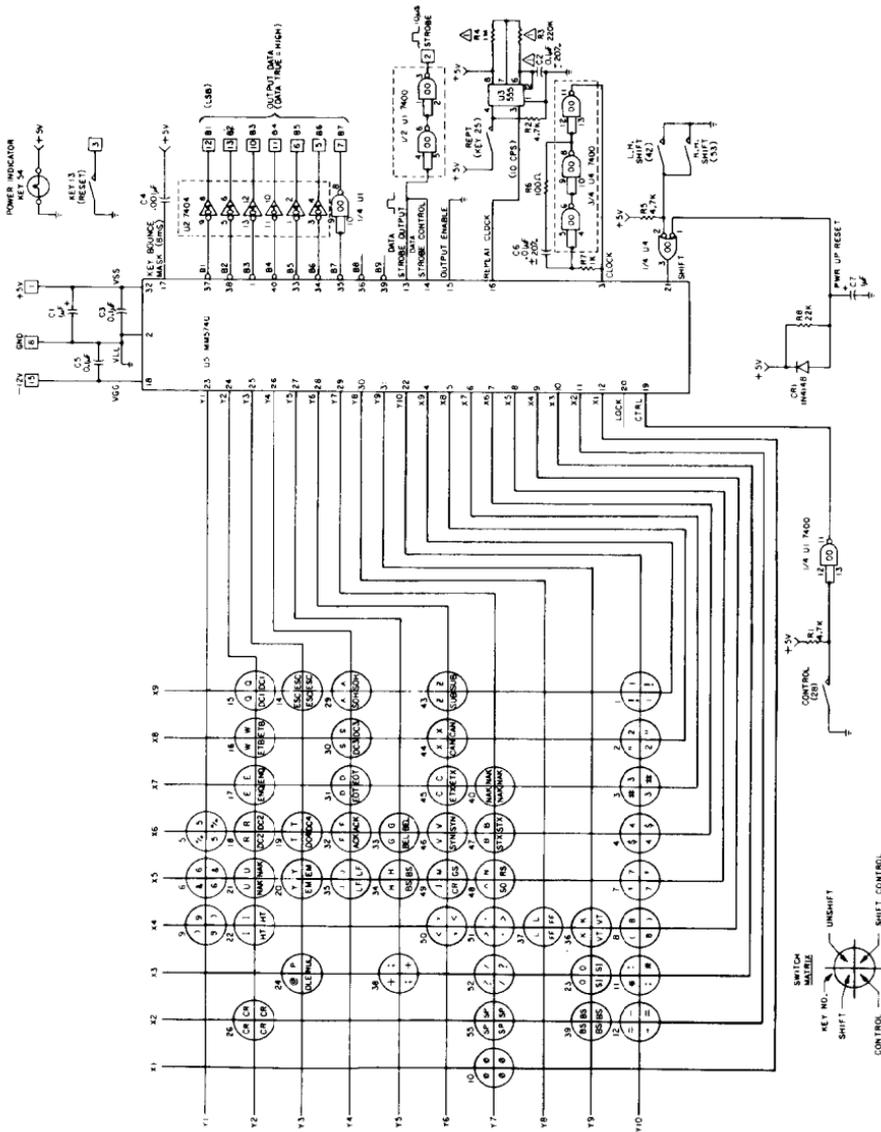


Schaubild 17. Schaltbild der Apple-Tastatur

Der Tastatur-Anschluss

Die Daten vom Tastenfeld des Apple gehen direkt in die RAM-Datenmultiplexer und Zwischenspeicher (zwei 74LS257 auf Position B6 und B7 auf der Apple-Hauptplatine). Die STROBE-Leitung des Tastaturanschlusses setzt einen 74LS74-Flip-Flop (auf Position B10), der das Bit 7, den Abtastimpuls der Speicherstelle \$C000, darstellt. Wird die „0“-Leitung des Ein-/Ausgabewählers (Position F13) aktiv, werden die Daten der sieben Eingänge des Tastaturanschlusses und der Zustand dieses Flip-Flop's über die Multiplexer in den Datenbus gegeben.

Tabelle 30: Beschreibung des Tastaturanschlusses

| Anschluß | Name | Beschreibung |
|-----------|--------|---|
| 1 | +5V | +5V Stromversorgung. Der maximale Strom sollte weniger als 120mA betragen. |
| 2 | STROBE | Abtastausgangsimpuls vom Tastenfeld. Dieser Leitung sollte wenigstens ein 10us dauerndes Signal gegeben werden, wenn eine Taste des Tastenfelds gedrückt wird. Es darf ein negativer oder auch ein positiver Impuls sein. |
| 3 | RESET | RESET-Leitung des Mikroprozessors. Dieser Anschluß liegt normalerweise auf +5V und sollte durch die RESET-Taste auf 0V gelegt werden. |
| 4,9,16 | NC | Kein Anschluß. |
| 5-7,10-13 | B1-B7 | 7 Bit ASCII-Dateneingänge für die Datenbusleitungen D ϕ -D6. |
| 8 | GND | Elektrische Masse des Systems: OV. |
| 15 | -12V | -12V Stromversorgung. Das Tastenfeld sollte diesem Anschluß weniger als 50mA entnehmen. |

| | | | |
|--------|---|----|------|
| +5V | 1 | 16 | NC |
| STROBE | 2 | 15 | -12V |
| RESET | 3 | 14 | NC |
| NC | 4 | 13 | B2 |
| B6 | 5 | 12 | B1 |
| B5 | 6 | 11 | B4 |
| B7 | 7 | 10 | B3 |
| GND | 8 | 9 | NC |

Schaubild 18. Tastenfeldanschlußbelegung

Kassettenanschlussbuchsen

An den zwei Miniatur-Phonobuchsen hinten auf der Apple-Platine können Sie Ihren Apple mit einem normalen Kassettenrekorder verbinden.

Kassetteneingangsbuchse: Diese Buchse ist vorgesehen, um mit der „Kopfhörer“- (Earphone) oder „Monitor“-Ausgangsbuchse des Kassettenrekorders verbunden zu werden. Die Eingangsspannung sollte 1V Spitzenspannung sein. Der Eingangswiderstand beträgt 12 kOhm.

Kassettenausgangsbuchse: Diese Buchse ist für den „Mikrophon“-Eingang des Kassettenrekorders vorgesehen. Die Ausgangsspannung ist 25mV bei einer 100 Ohm-Belastung.

Anschluss der Betriebsspannungen

Der Anschluß der Betriebsspannungen an die Platine des Apple erfolgt an einem AMP#9-35028-1 Anschlußsockel mit sechs Stiften.

Tabelle 31: Beschreibung des Betriebsspannungsanschlusses

| Anschluß | Name | Beschreibung |
|----------|------|--|
| 1,2 | GND | Elektrische Masse für die Platine des apple. |
| 3 | +5V | +5,0V Spannungsversorgung vom Netzgerät. Ein Apple mit 48K RAM-Speicher ohne Peripherie braucht ca. 1,5A von diesem Anschluß. |
| 4 | +12V | +12,0V Spannungsversorgung vom Netzgerät. Ein Apple mit 48K RAM Speicher ohne Peripherie braucht ca. 400mA von diesem Anschluß. |
| 5 | -12V | -12,0V Spannungsversorgung vom Netzgerät. Ein Apple mit 48K RAM-Speicher ohne Peripherie braucht ca. 12,5mA von diesem Anschluß. |
| 6 | -5V | -5,0V Spannungsversorgung vom Netzgerät. Ein Apple mit 48K RAM-Speicher ohne Peripherie braucht 0mA von diesem Anschluß. |

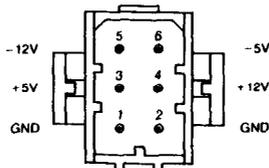


Schaubild 19. Stecker zum Anschluß der Betriebsspannungen

Der Lautsprecher

Der interne Lautsprecher des Apple wird über einen Darlington-Verstärker von einem halben 74LS74 Flip-Flop gesteuert. Der Anschluß des Lautsprechers erfolgt über eine Verbindungsbuchse der Serie Molex KK 100.

Tabelle 32: Beschreibung des Lautsprecheranschlusses

| anschluß | Name | Beschreibung |
|----------|------|---|
| 1 | SPKR | Lautsprechersignal. Dieser Anschluß liefert ca. 0,5W an einer 8 Ohm-Last. |
| 2 | +5V | +5V Spannungsversorgung. |

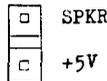


Schaubild 20. Anschluß des Lautsprechers

Peripherie-Anschlüsse

Die acht Peripherie-Steckleisten an der hinteren Kante der Hauptplatine sind vom Typ Winchester #2HW25CO-111 mit 50 Kontakten. Die Anschlußbelegungen werden im Schaubild 21 gezeigt, und die Beschreibung der Signale finden Sie auf den folgenden Seiten.

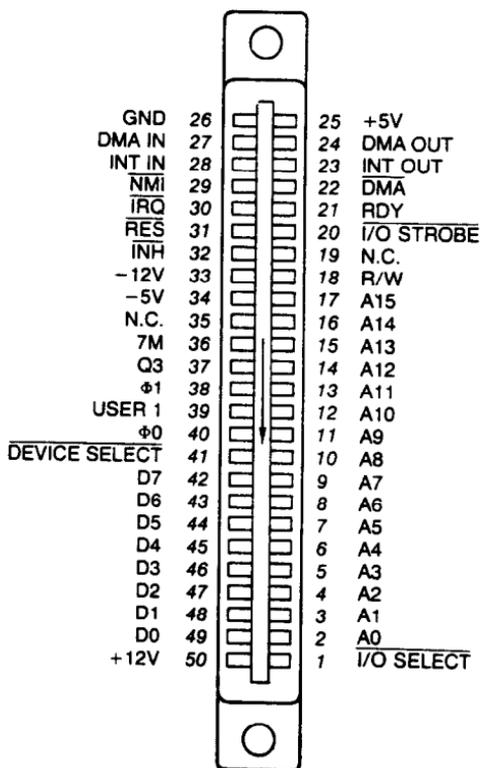


Schaubild 21. Anschlußbelegung der Peripherie-Steckleisten

Tabelle 33 : Signale an den Peripherie-Anschlüssen

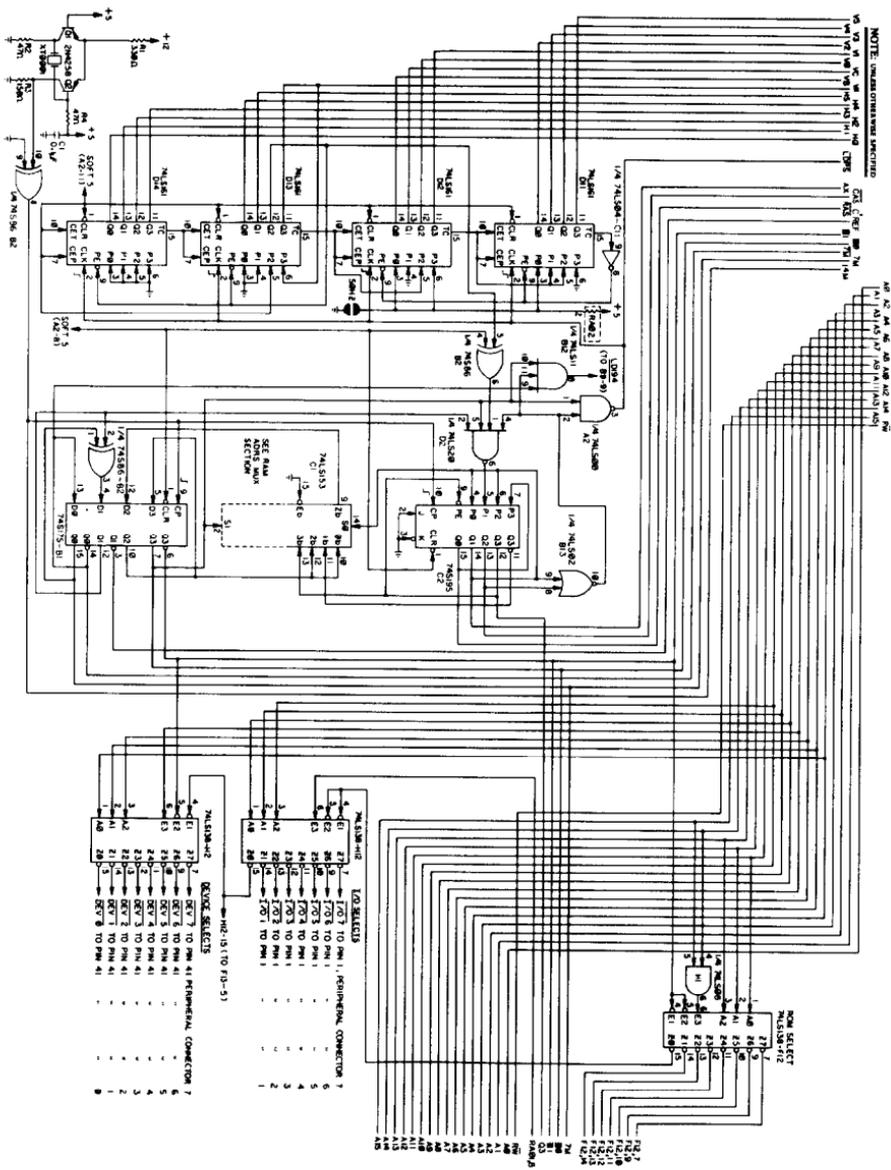
| Anschluß | Name | Beschreibung |
|----------|------------|--|
| 1 | I/O SELECT | Diese Leitung liegt normalerweise an + 5 V. Wenn der Mikroprozessor auf Seite \$Cn zugreift (wobei n die Einsteckschlitzznummer ist,) sinkt die Spannung auf 0V ab. Dieses Signal wird während $\Phi 1$ aktiv und treibt 10 LSTTL-Lasten*. Dieses Signal steht nicht am Einsteckschlitzz 0 zur Verfügung. |
| 2-17 | A0-A15 | Der gepufferte Adreßbus. Die Adressen dieser Anschlüsse werden in $\Phi 1$ gültig und bleiben es in $\Phi 0$. Jede dieser Leitungen treibt 5 LSTTL-Lasten*. |
| 18 | R/W | Gepuffertes READ/WRITE-Signal (Lesen/Schreiben). Dieses Signal ist zur selben Zeit gültig wie der Adreßbus und geht auf + 5 V in einem Lesevorgang und auf 0V in einem Schreibvorgang. Diese Leitung kann 2 LSTTL-Lasten versorgen*. |
| 19 | SYNC | Nur am Einsteckschlitzz 7 ist dieser Anschluß mit dem SYNC-Taktsignal des Videogenerators verbunden. |
| 20 | I/O STROBE | Diese Leitung treibt 4 LSTTL-Lasten und geht während $\Phi 0$ auf Null, wenn der Adreßbus eine Adresse zwischen \$C800 und \$CFFF enthält. |
| 21 | RDY | Der RDY-Eingang des 6502-Mikroprozessors. Wird diese Leitung in $\Phi 1$ auf Null gezogen, so stoppt der Mikroprozessor und hält die aktuelle Adresse im Adreßbus fest. |
| 22 | DMA | Wird dieser Anschluß auf 0V gelegt, so wird der Adreßbus gesperrt und der Mikroprozessor gestoppt. Diese Leitung wird durch einen 3 K Ω - Widerstand auf +5 V gehalten. |
| 23 | INT OUT | „Daisy-Chain“ Interrupt-Ausgang zu Geräten niedrigerer Priorität. Dieser Anschluß wird normalerweise mit Anschluß 28 (INT IN) verbunden. |
| 24 | DMA OUT | „Daisy-Chain“ DMA-Ausgang zu Geräten niedrigerer Priorität. Dieser Anschluß wird normalerweise mit Anschluß 22 (DMA IN) verbunden. |

* Die Belastungsangaben beziehen sich auf jede einzelne Peripheriekarte.

| Anschluß | Name | Beschreibung |
|----------|--------|--|
| 25 | +5V | +5 V Spannungsversorgung. Für alle Peripheriekarten stehen insgesamt 500 mA zur Verfügung. |
| 26 | GND | Elektrische Masse des Systems. |
| 27 | DMA IN | „Daisy-Chain“ DMA-Eingang von Geräten höherer Priorität. Gewöhnlich mit Anschluß 24 (DMA OUT) verbunden. |
| 28 | INT IN | „Daisy-Chain“ Interrupt-Eingang von Geräten höherer Priorität. Gewöhnlich mit Anschluß 23 (INT OUT) verbunden. |
| 29 | NMI | Nicht maskierbarer Interrupt (hardwaremäßiges Einschleichen eines speziellen Unterprogramms). Wenn diese Leitung auf 0 gezogen wird, beginnt der Apple einen Interrupt-Ablauf und springt dann zu einem Interrupt-Behandlungsprogramm über Adresse \$3FB. |
| 30 | IRQ | Interrupt mit Nachfrage (Interrupt ReQuest). Wenn diese Leitung auf 0V liegt und das „I“-Bit des 6502-Mikroprozessors (Interrupt-Sperre) nicht gesetzt ist, beginnt der Apple einen Interrupt-Ablauf und springt zu dem Interrupt-Behandlungsprogramm, dessen Adresse in den Speicherzellen \$3FE und \$3FF zu finden ist. |
| 31 | RES | Wird dieser Anschluß an 0V gelegt, so beginnt der Mikroprozessor einen RESET-Ablauf (siehe Seite 45). |
| 32 | INH | Wenn diese Leitung auf 0 gezogen wird, werden alle ROMs auf der Platine des Apple abgeschaltet. Diese Leitung wird durch einen 3 K Ω -Widerstand auf +5V gehalten. |
| 33 | -12V | - 12V Spannungsversorgung. Der Maximalstrom beträgt 200 mA für alle Peripheriekarten zusammen. |

| | | |
|-------|---------------|--|
| 34 | -5V | - 5 V Spannungsversorgung. Der maximal zulässige Strom beträgt 200 mA für alle Peripheriekarten zusammen. |
| 35 | COLOR REF | Nur bei Einsteckschlitz 7 ist dieser Anschluß mit dem Farbsynchronsignal des Video-Generators verbunden. Es ist darauf zu achten, daß das Farbsynchronsignal die richtige Frequenz hat. |
| 36 | 7M | 7 MHz-Takt. Diese Leitung treibt zwei LSTTL-Lasten*. |
| 37 | Q3 | Asymmetrischer 2 MHz-Takt. Dieser Anschluß treibt zwei LSTTL-Lasten*. |
| 38 | 1 | Phase 1-Takt des Mikroprozessors. Dieser Anschluß kann zwei LSTTL-Lasten versorgen*. |
| 39 | USER 1 | Wenn diese Leitung auf « 0 » gezogen wird, ist jegliche Adressierung der internen Ein-/Ausgabe unterbrochen (siehe Seite 103). |
| 40 | $\Phi 0$ | Phase 0-Takt des Mikroprozessors. Dieser Anschluß kann zwei LSTTL-Lasten versorgen*. |
| 41 | DEVICE SELECT | Leitung wird auf jedem Peripherie-Anschluß aktiv (0/ Volt), wenn der Adreßbus eine Adresse zwischen $\$C0n0$ und $\$C0nF$ gespeichert hat, wobei n die um \$8 erhöhte Einsteckschlitznummer angibt. Die Leitung treibt 10 LSTTL-Lasten.* |
| 42-49 | D0-D7 | In zwei Richtungen gepufferter Datenbus. Die Dateninformation auf dieser Leitung liegt 300ns in Phase 0 beim Schreibvorgang vor und sollte beim Lesevorgang nicht weniger als 100ns vor dem Ende von $\Phi 0$ erhalten bleiben. |
| 50 | +12V | Positive 12 Volt Stromversorgung. Bis zu 250mA können insgesamt an alle Peripherie-Karten abgegeben werden. |

* Die Belastungsangaben beziehen sich auf jede einzelne Peripheriekarte.



NOTE: UNDESIGNATED SIGNALS ARE ACTIVE LOW UNLESS OTHERWISE INDICATED

Schaubild 22-1. Schaltplan des Apple II

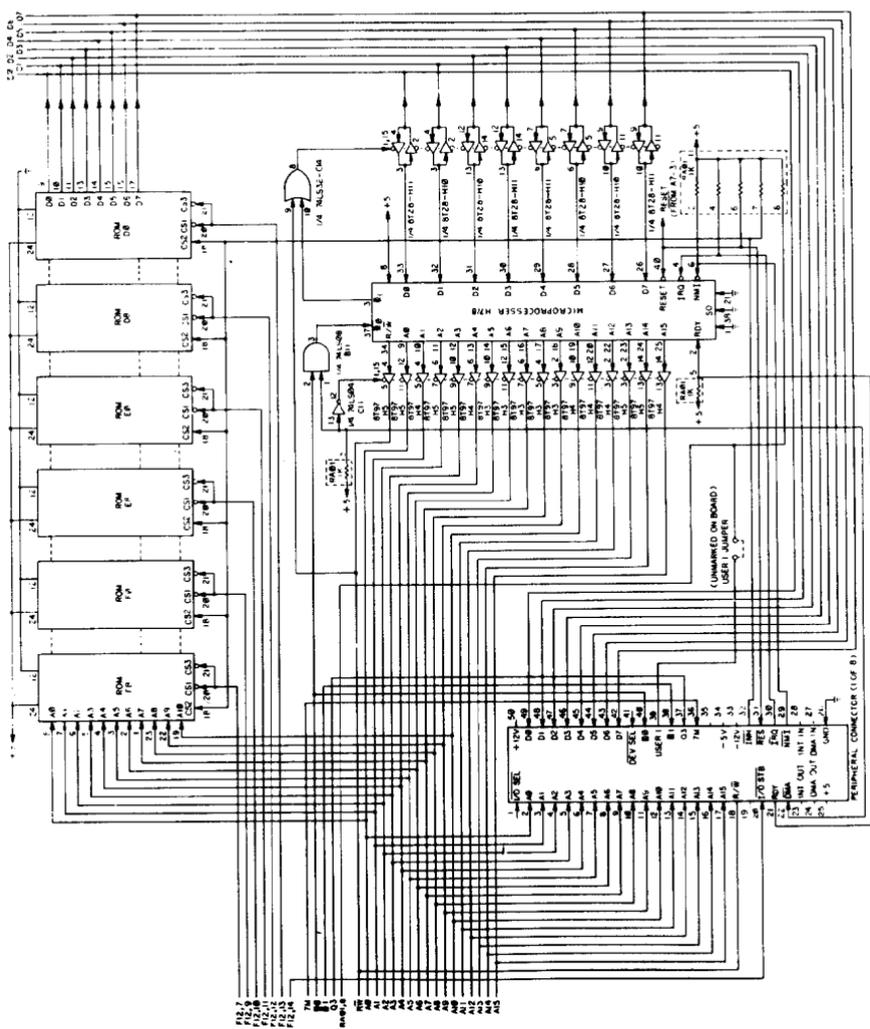


Schaubild 22-2. Schaltplan des Apple II

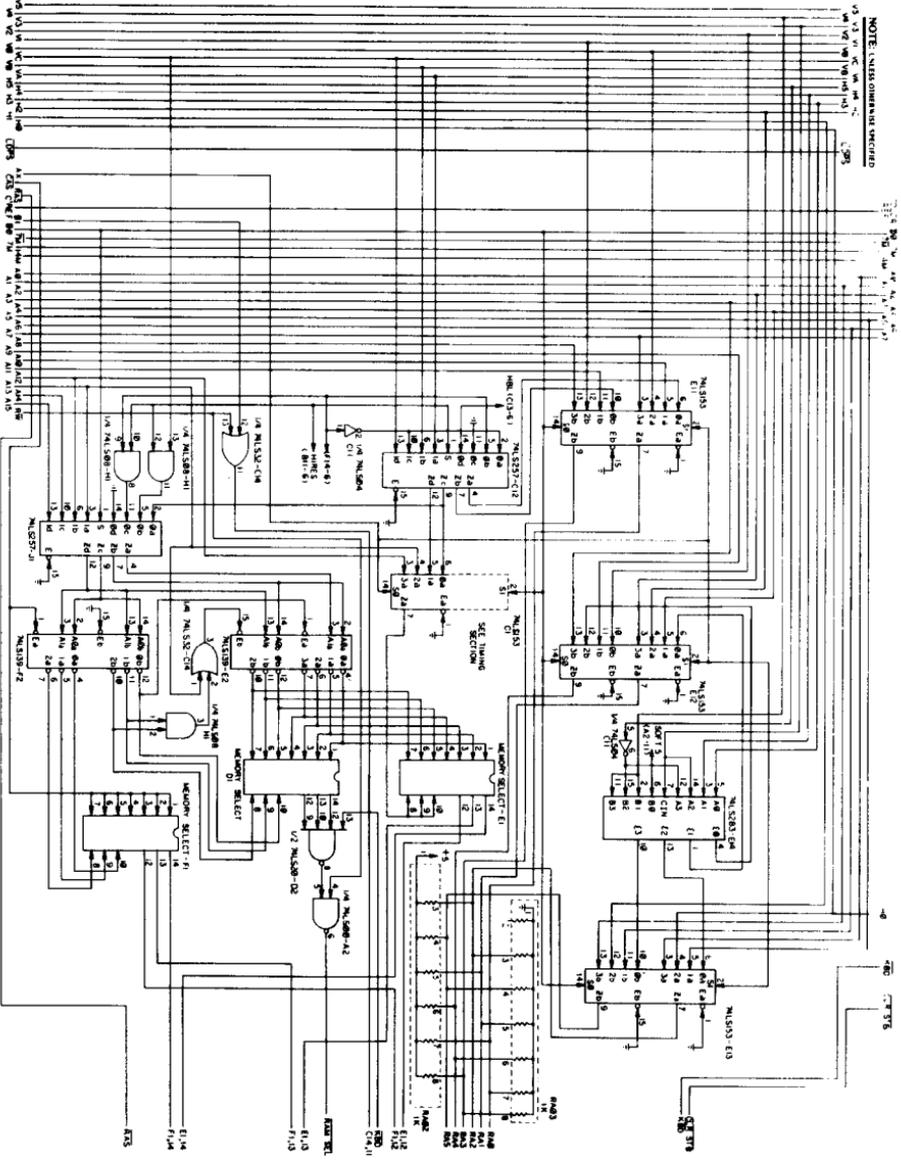


Schaubild 22-3. Schaltplan des Apple II

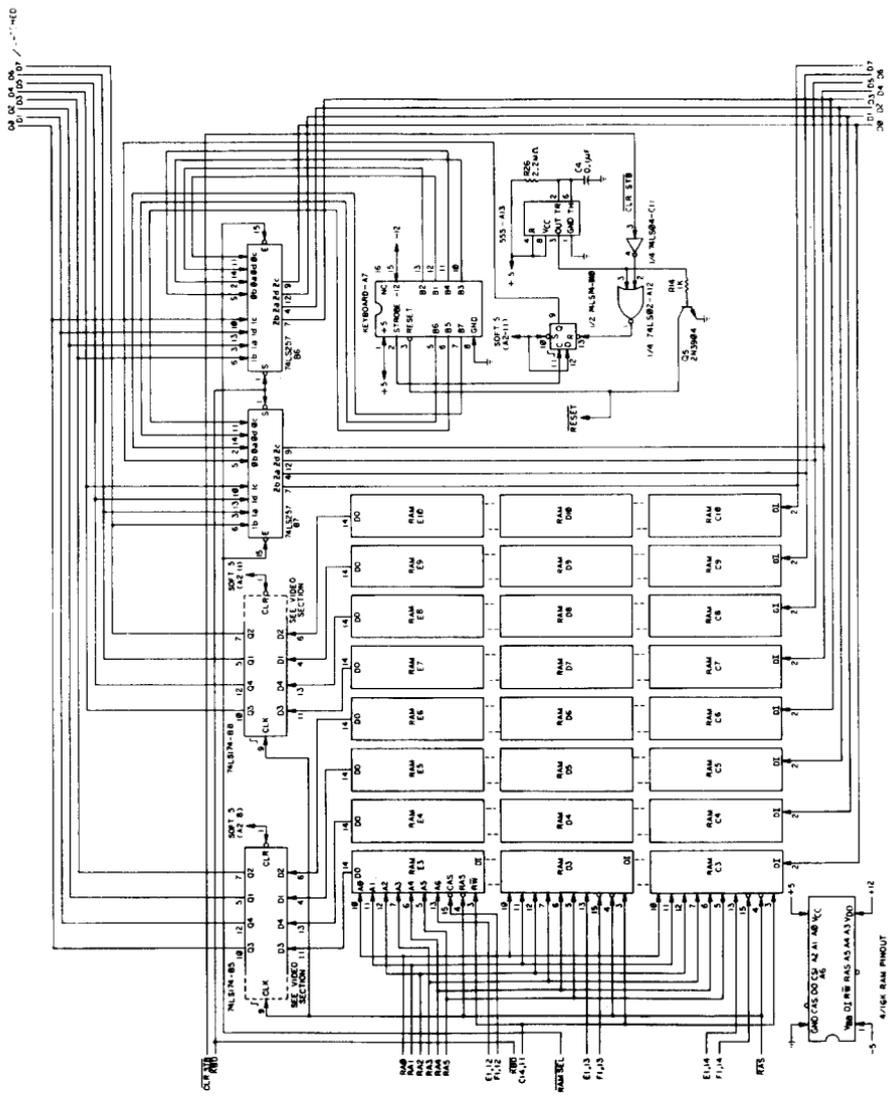


Schaubild 22-4. Schaltplan des Apple II

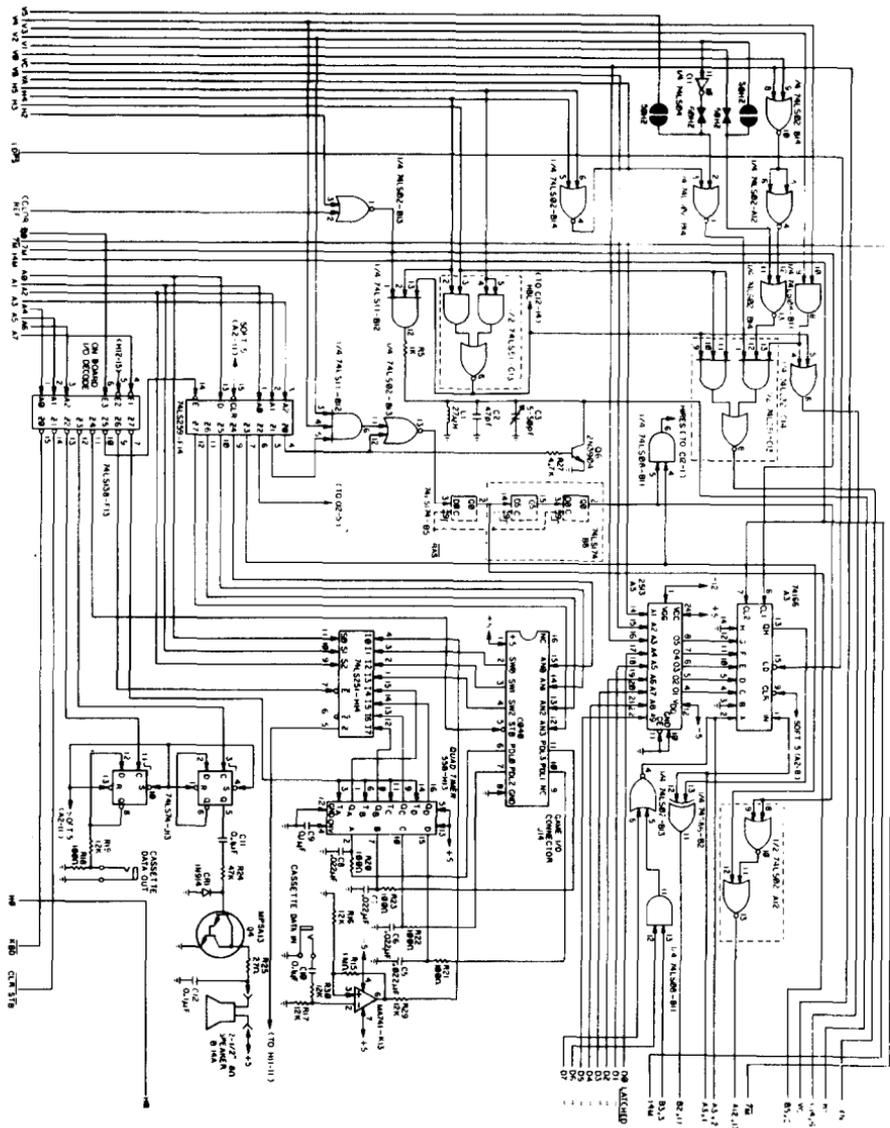


Schaubild 22-5. Schaltplan des Apple II

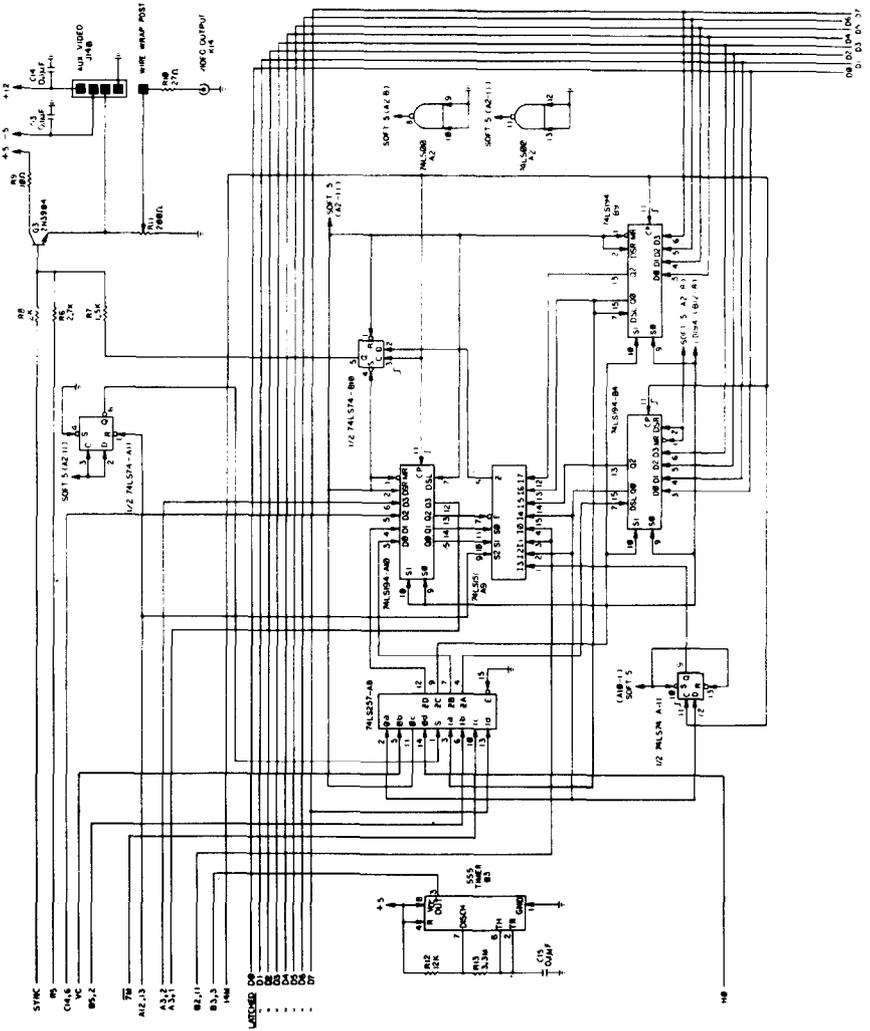


Schaubild 22-6. Schaltplan des Apple II

ANHANG A

Der Befehlssatz des 6502-Mikroprozessors

Befehlssatz des 6502-Mikroprozessors

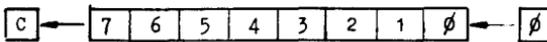
| | |
|------------|---|
| ADC | Addiere Speicher zum Akkumulator mit Übertrag (Carry) |
| AND | Logisches „UND“ : Speicher mit Akkumulator |
| ASL | (Akkumulator oder Speicher) ein Bit nach links schieben |
| BCC | Verzweigung bei gelöschtem Übertrag (Carry) |
| BCS | Verzweigung bei gesetztem Übertrag (Carry) |
| BEQ | Verzweigung bei Null |
| BIT | Vergleiche Bits aus Speicher mit Akkumulator |
| BMI | Verzweigung bei Minus |
| BNE | Verzweigung bei ungleich Null |
| BPL | Verzweigung bei Plus |
| BRK | Interrupt erzwingen |
| BVC | Verzweigung bei gelöschtem Überlauf (Overflow) |
| BVS | Verzweigung bei gesetztem Überlauf (Overflow) |
| CLC | Lösche Übertrag (Carry) |
| CLD | Lösche Dezimalmodus |
| CLI | Lösche Interruptsperre |
| CLV | Lösche Überlauf (Overflow) |
| CMP | Vergleiche Akkumulator mit Speicher |
| CPX | Vergleiche Indexregister X mit Speicher |
| CPY | Vergleiche Indexregister Y mit Speicher |
| DEC | Vermindere Speicher um 1 |
| DEX | Vermindere Indexregister X um 1 |
| DEY | Vermindere Indexregister Y um 1 |
| EOR | Logisches „Exklusives ODER“ : Speicher mit Akkumulator |
| INC | Erhöhe Speicher um 1 |
| INX | Erhöhe Indexregister X um 1 |
| INY | Erhöhe Indexregister Y um 1 |
| JMP | Unbedingter Sprung |
| JSR | Springe ins Unterprogramm |
| LDA | Lade Speicher in den Akkumulator |
| LDX | Lade Speicher in das Indexregister X |
| LDY | Lade Speicher in das Indexregister Y |
| LSR | (Akkumulator oder Speicher) ein Bit nach rechts schieben |
| NOP | Keine Operation |
| ORA | Logisches „ODER“ : Speicher mit Akkumulator |
| PHA | Bringe Akkumulator in den Keller |
| PHP | Bringe den Prozessor-Status in den Keller |
| PLA | Hole Akkumulator aus dem Keller |
| PLP | Hole den Prozessorstatus aus dem Keller |
| ROL | (Akkumulator oder Speicher) ein Bit im Kreis nach links schieben |
| ROR | (Akkumulator oder Speicher) ein Bit im Kreis nach rechts schieben |
| RTI | Rücksprung aus dem Interrupt |
| RTS | Rücksprung aus dem Unterprogramm |
| SBC | Subtrahiere Speicher und Übertrag vom Akkumulator |
| SEC | Setze Übertrag (Carry) |
| SED | Setze Dezimalmodus |
| SEI | Setze Interruptsperre |
| STA | Speichere aus dem Akkumulator |
| STX | Speichere aus dem Indexregister X |
| STY | Speichere aus dem Indexregister Y |
| TAX | Transportiere Akkumulator zum Indexregister X |
| TAY | Transportiere Akkumulator zum Indexregister Y |
| TSX | Transportiere Kellerzeiger zum Indexregister X |

| | |
|------------|--|
| TXA | Transportiere Indexregister X zum Akkumulator |
| TXS | Transportiere Indexregister X zum Kellerzeiger |
| TYA | Transportiere Indexregister Y zum Akkumulator |

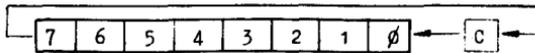
Erklärung zu den folgenden Tabellen

| | |
|---------------|--|
| A | Akkumulator |
| X, Y | Index-Register |
| M | Speicher |
| C | Übertrag (Carry) |
| \bar{C} | Negativer Übertrag (Borrow) |
| P | Prozessor-Status-Register |
| S | Kellerzeiger |
| ✓ | Ändert sich |
| - | Keine Änderung |
| + | Addition |
| \wedge | Logisches UND |
| - | Subtraktion |
| ∇ | Logisches exklusives ODER (ausschließendes ODER) |
| \uparrow | Transport aus dem Keller |
| \downarrow | Transport in den Keller |
| \rightarrow | Transport in Richtung |
| \leftarrow | Transport in Richtung |
| \vee | Logisches ODER |
| PC | Befehlszähler |
| PCH | Befehlszähler hohes Byte |
| PCL | Befehlszähler niedriges Byte |
| OPER | Operand |
| # | Unmittelbare Adressierart |

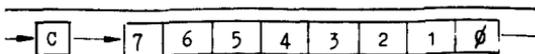
Schema 1 : ASL = (Akkumulator oder Speicher) um ein Bit nach links schieben



Schema 2 : ROL = (Akkumulator oder Speicher) ein Bit im Kreis nach links schieben



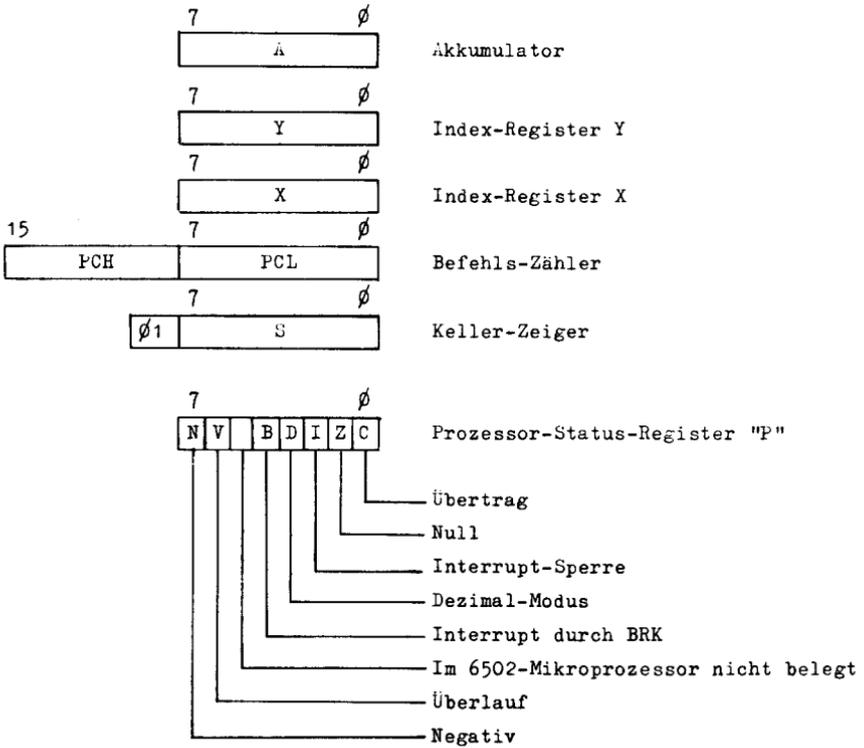
Schema 3 : ROR = (Akkumulator oder Speicher) ein Bit im Kreis nach rechts schieben



föBemerkung zu BIT : Bit 6 und 7 werden ins Status-Register übertragen :

N = Bit 7; V = Bit 6; Z = 1, falls A „UND“ M = 0, sonst Z = 0.

Register des 6502-Mikroprozessors



Die folgende Aufstellung der Befehle mit ihren Adreßarten ist dem Englischen entnommen. Die deutschen Bezeichnungen finden Sie ab Seite 120 und weitere Erklärungen für ASL, BIT, LSR, ROL und ROR auf Seite 121.

Die Tabelle dient als Nachschlagewerk für mögliche Adreßarten, hexadezimale Befehls-codes, Anzahl der Zyklen und Veränderung der Status-Bits. Ausführliche Erklärungen der Befehle und der Arbeitsweise des Mikroprozessors finden Sie in 6500 Programmier-Handbüchern. Noch eine Hilfe: „Immediate“ heißt „unmittelbar“, „Zero Page“ steht für „Seite Null“ und „Implied“ bedeutet „impliziert“.

| Name Description | Operation | Addressing Mode | Assembly Language Form | HEX OP Code | No. Bytes | "F" Status Reg. N Z C I O V |
|---|--|---|---|--|--------------------------------------|-------------------------------------|
| ADC Add memory to accumulator with carry | A-M-C → A C | Immediate Zero Page Zero Page.X Absolute Absolute.X Absolute.Y Absolute.Y (Indirect.X) (Indirect.Y) | ADC #Oper ADC Oper ADC Oper.X ADC Oper ADC Oper.X ADC Oper.Y ADC (Oper.X) ADC (Oper).Y | 69 65 75 6D 7D 79 61 71 | 2 2 2 3 3 3 2 2 | √√√--√ |
| AND "AND" memory with accumulator | A A M → A | Immediate Zero Page Zero Page.X Absolute Absolute.X Absolute.Y Absolute.Y (Indirect.X) (Indirect.Y) | AND #Oper AND Oper AND Oper.X AND Oper AND Oper.X AND Oper.Y AND (Oper.X) AND (Oper).Y | 29 25 35 2D 3D 39 21 31 | 2 2 2 3 3 3 2 2 | √√----- |
| ASL Shift left one bit (Memory or Accumulator) | siehe Schaubild 1 | Accumulator Zero Page Zero Page.X Absolute Absolute.X | ASL A ASL Oper ASL Oper.X ASL Oper ASL Oper.X | 0A 06 16 0E 1E | 1 2 2 3 3 | √√√--- |
| BCC Branch on carry clear | Branch on C=0 | Relative | BCC Oper | 90 | 2 | ----- |
| BCS Branch on carry set | Branch on C=1 | Relative | BCS Oper | 80 | 2 | ----- |
| BEQ Branch on result zero | Branch on Z=1 | Relative | BEQ Oper | F0 | 2 | ----- |
| BIT Test bits in memory with accumulator | A A M, M ₇ → N, M ₆ → V | Zero Page Absolute | BIT* Oper BIT* Oper | 24 2C | 2 3 | M ₇ √-----M ₆ |
| BMI Branch on result minus | Branch on N=1 | Relative | BMI Oper | 30 | 2 | ----- |
| BNE Branch on result not zero | Branch on Z=0 | Relative | BNE Oper | D0 | 2 | ----- |
| BPL Branch on result plus | Branch on N=0 | Relative | BPL oper | 10 | 2 | ----- |
| BRK Force Break | Forced Interrupt PC-2 † P † | Implied | BRK* | 00 | 1 | ---1-- |
| BVC Branch on overflow clear | Branch on V=0 | Relative | BVC Oper | 50 | 2 | ----- |

* BIT : Bit 6 und 7 werden ins Status-Register übertragen : N = Bit 7; V = Bit 6; Z = 1, falls $A \wedge M = 0$, sonst $Z = 0$.
BRK : Ein BRK-Befehl führt unabhängig vom I-Bit (Interrupt-Sperre) zum Interrupt.

| Name Description | Operation | Addressing Mode | Assembly Language Form | HEX OP Code | No. Bytes | "P" Status Reg N Z C I O V |
|---|---------------|--|---|--|--------------------------------------|-------------------------------|
| BVS Branch on overflow set | Branch on V=1 | Relative | BVS Oper | 70 | 2 | ----- |
| CLC Clear carry flag | 0 → C | Implied | CLC | 1B | 1 | ---0-- |
| CLD Clear decimal mode | 0 → D | Implied | CLD | D8 | 1 | -0---- |
| CLI | 0 → I | Implied | CLI | 5B | 1 | ---0-- |
| CLV Clear overflow flag | 0 → V | Implied | CLV | B8 | 1 | 0----- |
| CMP Compare memory and accumulator | A — M | Immediate Zero Page Zero Page, X Absolute Absolute, X Absolute, Y (Indirect, X) (Indirect, Y) | CMP #Oper CMP Oper Zero Page CMP Oper, X Absolute CMP Oper, X Absolute, X CMP Oper, Y Absolute, Y CMP (Oper, X) CMP (Oper, Y) | C9 C5 D5 CD DD D9 C1 D1 | 2 2 2 3 3 3 2 2 | √√---- |
| CPX Compare memory and index X | X — M | Immediate Zero Page Absolute | CPX #Oper CPX Oper Zero Page CPX Oper Absolute | E0 E4 EC | 2 2 3 | √√---- |
| CPY Compare memory and index Y | Y — M | Immediate Zero Page Absolute | CPY #Oper CPY Oper Zero Page CPY Oper Absolute | C0 C4 CC | 2 2 3 | √√---- |
| DEC Decrement memory by one | M — 1 → M | Zero Page Zero Page, X Absolute Absolute, X | DEC Oper DEC Oper, X Zero Page DEC Oper Absolute DEC Oper, X Absolute, X | C6 D6 CE DE | 2 2 3 3 | √----- |
| DEX Decrement index X by one | X — 1 → X | Implied | DEX | CA | 1 | √----- |
| DEY Decrement index Y by one | Y — 1 → Y | Implied | DEY | B8 | 1 | √----- |

| Name Description | Operation | Addressing Mode | Assembly Language Form | HEX OP Code | No. Bytes | "P" Status Reg. N Z C I O V |
|---|--|---|---|--|--------------------------------------|--------------------------------|
| EOR "Exclusive-Or" memory with accumulator | A V M → A | Immediate Zero Page Zero Page.X Absolute Absolute.X Absolute.Y (Indirect.X) (Indirect.Y) | EOR #Oper EOR Oper EOR Oper.X EOR Oper EOR Oper.X EOR Oper.Y EOR (Oper.X) EOR (Oper).Y | 49 45 55 4D 5D 59 41 51 | 2 2 2 3 3 3 2 2 | √ --- |
| INC Increment memory by one | M - 1 → M | Zero Page Zero Page.X Absolute Absolute.X | INC Oper INC Oper.X INC Oper INC Oper.X | E6 F6 EE FE | 2 2 3 3 | √ --- |
| INX Increment index X by one | X + 1 → X | Implied | INX | E8 | 1 | √ --- |
| INY Increment index Y by one | Y + 1 → Y | Implied | INY | C8 | 1 | √ --- |
| JMP Jump to new location | (PC+1) → PCL (PC+2) → PCH | Absolute Indirect | JMP Oper JMP (Oper) | 4C 6C | 3 3 | ----- |
| JSR Jump to new location saving return address | PC+2 ↓ (PC+1) → PCL (PC+2) → PCH | Absolute | JSR Oper | 20 | 3 | ----- |
| LDA Load accumulator with memory | M → A | Immediate Zero Page Zero Page.X Absolute Absolute.X Absolute.Y (Indirect.X) (Indirect.Y) | LDA #Oper LDA Oper LDA Oper.X LDA Oper LDA Oper.X LDA Oper.Y LDA (Oper.X) LDA (Oper).Y | A9 A5 B5 AD BD B9 A1 B1 | 2 2 2 3 3 3 2 2 | √ --- |
| LDX Load index X with memory | M → X | Immediate Zero Page Zero Page.Y Absolute Absolute.Y | LDX #Oper LDX Oper LDX Oper.Y LDX Oper LDX Oper.Y | A2 A6 B6 AE BE | 2 2 2 3 3 | √ --- |
| LDY Load index Y with memory | M → Y | Immediate Zero Page Zero Page.X Absolute Absolute.X | LDY #Oper LDY Oper LDY Oper.X LDY Oper LDY Oper.X | A0 A4 B4 AC BC | 2 2 2 3 3 | √ --- |

| Name Description | Operation | Addressing Mode | Assembly Language Form | HEX OP Code | No Bytes | "P" Status Reg. N Z C B D V |
|---|----------------------|---|---|--|--------------------------------------|--------------------------------|
| LSR Shift right one bit (memory or accumulator) | siehe Schaubild 1 | Accumulator Zero Page Zero Page.X Absolute Absolute X | LSR A LSR Oper LSR Oper.X LSR Oper LSR Oper X | 4A 46 56 4E 5E | 1 2 2 3 3 | 0√---- |
| NOP No operation | No Operation | Implied | NOP | EA | 1 | ----- |
| ORA "OR" memory with accumulator | A V M → A | Immediate Zero Page Zero Page.X Absolute Absolute X Absolute Y (Indirect.X) (Indirect.Y) | ORA #Oper ORA Oper ORA Oper.X ORA Oper ORA Oper.X ORA Oper.Y ORA (Oper.X) ORA (Oper).Y | 09 05 15 00 1D 19 01 11 | 2 2 2 3 3 3 2 2 | √----- |
| PHA Push accumulator on stack | A ↓ | Implied | PHA | 48 | 1 | ----- |
| PHP Push processor status on stack | P ↓ | Implied | PHP | 08 | 1 | ----- |
| PLA Pull accumulator from stack | A ↑ | Implied | PLA | 68 | 1 | √~----- |
| PLP Pull processor status from stack | P ↑ | Implied | PLP | 28 | 1 | From Stack |
| ROL Rotate one bit left (memory or accumulator) | siehe Schaubild 2 | Accumulator Zero Page Zero Page.X Absolute Absolute.X | ROL A ROL Oper ROL Oper.X ROL Oper ROL Oper.X | 2A 26 36 2E 3E | 1 2 2 3 3 | √√√---- |
| ROR Rotate one bit right (memory or accumulator) | siehe Schaubild 3 | Accumulator Zero Page Zero Page.X Absolute Absolute.X | ROR A ROR Oper ROR Oper.X ROR Oper ROR Oper.X | 6A 66 76 6E 7E | 1 2 2 3 3 | √√√---- |

| Name Description | Operation | Addressing Mode | Assembly Language Form | HEX OP Code | No. Bytes | "P" Status Reg N Z C I D V |
|---|-----------------------|---|---|--|--------------------------------------|-------------------------------|
| RTI Return from interrupt | P ← PC † | Implied | RTI | 40 | 1 | From Stack |
| RTS Return from subroutine | PC †, PC-1 → PC | Implied | RTS | 60 | 1 | ----- |
| SBC Subtract memory from accumulator with borrow | A - M - \bar{C} → A | Immediate Zero Page Zero Page.X Absolute Absolute.X Absolute.Y (Indirect.X) (Indirect.Y) | SBC #Oper SBC Oper SBC Oper.X SBC Oper SBC Oper.X SBC Oper.Y SBC (Oper.X) SBC (Oper).Y | E9 E5 F5 ED FD F9 F1 E1 | 2 2 2 3 3 3 2 2 | √√√--- |
| SEC Set carry flag | 1 → C | Implied | SEC | 38 | 1 | --1---- |
| SED Set decimal mode | 1 → D | Implied | SED | F8 | 1 | -----1- |
| SEI Set interrupt disable status | 1 → I | Implied | SEI | 78 | 1 | ----1--- |
| STA Store accumulator in memory | A → M | Zero Page Zero Page.X Absolute Absolute.X Absolute.Y (Indirect.X) (Indirect.Y) | STA Oper STA Oper.X STA Oper STA Oper.X STA Oper.Y STA (Oper.X) STA (Oper).Y | 85 95 80 90 99 81 91 | 2 2 3 3 3 2 2 | ----- |
| STX Store index X in memory | X → M | Zero Page Zero Page.Y Absolute | STX Oper STX Oper.Y STX Oper | 86 96 8E | 2 2 3 | ----- |
| STY Store index Y in memory | Y → M | Zero Page Zero Page.X Absolute | STY Oper STY Oper.X STY Oper | 84 94 8C | 2 2 3 | ----- |
| TAX Transfer accumulator to index X | A → X | Implied | TAX | AA | 1 | √√----- |
| TAY Transfer accumulator to index Y | A → Y | Implied | TAY | A8 | 1 | √√----- |
| TSX Transfer stack pointer to index X | S → X | Implied | TSX | BA | 1 | √√----- |

| Name Description | Operation | Addressing Mode | Assembly Language Form | HEX OP Code | No. Bytes | "P" Status Reg. N Z C I B V |
|--|-----------|--------------------|------------------------------|-------------------|--------------|--------------------------------|
| TXA Transfer index X to accumulator | X → A | Implied | TXA | 8A | 1 | √√---- |
| TXS Transfer index X to stack pointer | X → S | Implied | TXS | 9A | 1 | ----- |
| TYA Transfer index Y to accumulator | Y → A | Implied | TYA | 00 | 1 | √√---- |

Hexadezimale Befehls-Codes

| | | | | |
|-----------|----------------------------|--|-----------|----------------------------|
| 00 | BRK | | 26 | ROL - Seite Null |
| 01 | ORA - (Indirekt, X) | | 27 | NOP |
| 02 | NOP | | 28 | PLP |
| 03 | NOP | | 29 | AND - Unmittelbar |
| 04 | NOP | | 2A | ROL - Akkumulator |
| 05 | ORA - Seite Null | | 2B | NOP |
| 06 | ASL - Seite Null | | 2C | BIT - Absolut |
| 07 | NOP | | 2D | AND - Absolut |
| 08 | PHP | | 2E | ROL - Absolut |
| 09 | ORA - Unmittelbar | | 2F | NOP |
| 0A | ASL - Akkumulator | | 30 | BMI |
| 0B | NOP | | 31 | AND - (Indirekt),Y |
| 0C | NOP | | 32 | NOP |
| 0D | ORA - Absolut | | 33 | NOP |
| 0E | ASL - Absolut | | 34 | NOP |
| 0F | NOP | | 35 | AND - Seite Null, X |
| 10 | BPL | | 36 | ROL - Seite Null, X |
| 11 | ORA - (Indirekt),Y | | 37 | NOP |
| 12 | NOP | | 38 | SEC |
| 13 | NOP | | 39 | AND - Absolut,Y |
| 14 | NOP | | 3A | NOP |
| 15 | ORA - Seite Null, X | | 3B | NOP |
| 16 | ASL - Seite Null, X | | 3C | NOP |
| 17 | NOP | | 3D | AND - Absolut, X |
| 18 | CLC | | 3E | ROL - Absolut, X |
| 19 | ORA - Absolut,Y | | 3F | NOP |
| 1A | NOP | | 40 | RTI |
| 1B | NOP | | 41 | EOR - (Indirekt, X) |
| 1C | NOP | | 42 | NOP |
| 1D | ORA - Absolut, X | | 43 | NOP |
| 1E | ASL - Absolut, X | | 44 | NOP |
| 1F | NOP | | 45 | EOR - Seite Null |
| 20 | JSR | | 46 | LSR - Seite Null |
| 21 | AND - (Indirekt, X) | | 47 | NOP |
| 22 | NOP | | 48 | PHA |
| 23 | NOP | | 49 | EOR - Unmittelbar |
| 24 | BIT - Seite Null | | 4A | LSR - Akkumulator |
| 25 | AND - Seite Null | | 4B | NOP |

| | | | |
|-----------|----------------------------|-----------|-----------------------------|
| 4C | JMP - Absolut | 74 | NOP |
| 4D | EOR - Absolut | 75 | ADC - Seite Null, X |
| 4E | LSR - Absolut | 76 | ROR - Seite Null, X |
| 4F | NOP | 77 | NOP |
| 50 | BVC | 78 | SEI |
| 51 | NOP | 79 | ADC - Absolut, Y |
| 52 | NOP | 7A | NOP |
| 53 | NOP | 7B | NOP |
| 54 | NOP | 7C | NOP |
| 55 | EOR - Seite Null, X | 7D | ADC - Absolut, X NOP |
| 56 | LSR - Seite Null, X | 7E | ROR - Absolut, X NOP |
| 57 | NOP | 7F | NOP |
| 58 | CLI | 80 | NOP |
| 59 | EOR - Absolut, Y | 81 | STA - (Indirekt, X) |
| 5A | NOP | 82 | NOP |
| 5B | NOP | 83 | NOP |
| 5C | NOP | 84 | STY - Seite Null |
| 5D | EOR - Absolut, X | 85 | STA - Seite Null |
| 5E | LSR - Absolut, X | 86 | STX - Seite Null |
| 5F | NOP | 87 | NOP |
| 60 | RTS | 88 | DEY |
| 61 | ADC - (Indirekt, X) | 89 | NOP |
| 62 | NOP | 8A | TXA |
| 63 | NOP | 8B | NOP |
| 64 | NOP | 8C | STY - Absolut |
| 65 | ADC - Seite Null | 8D | STA - Absolut |
| 66 | ROR - Seite Null | 8E | STX - Absolut |
| 67 | NOP | 8F | NOP |
| 68 | PLA | 90 | BCC |
| 69 | ADC - Unmittelbar | 91 | STA - (Indirekt), Y |
| 6A | ROR - Akkumulator | 92 | NOP |
| 6B | NOP | 93 | NOP |
| 6C | JMP - Indirekt | 94 | STY - Seite Null, X |
| 6D | ADC - Absolut | 95 | STA - Seite Null, X |
| 6E | ROR - Absolut | 96 | STX - Seite Null, Y |
| 6F | NOP | 97 | NOP |
| 70 | BVS | 98 | TYA |
| 71 | ADC - (Indirekt), Y | 99 | STA - Absolut, Y |
| 72 | NOP | 9A | TXS |
| 73 | NOP | 9B | NOP |

| | | | | |
|-----------|----------------------------|--|-----------|----------------------------|
| 9C | NOP | | C4 | CPY - Seite Null |
| 9D | STA - Absolut, X | | C5 | CMP - Seite Null |
| 9E | NOP | | C6 | DEC - Seite Null |
| 9F | NOP | | C7 | NOP |
| A0 | LDY - Unmittelbar | | C8 | INY |
| A1 | LDA - (Indirekt, X) | | C9 | CMP - Unmittelbar |
| A2 | LDX - Unmittelbar | | CA | DEX |
| A3 | NOP | | CB | NOP |
| A4 | LDY - Seite Null | | CC | CPY - Absolut |
| A5 | LDA - Seite Null | | CD | CMP - Absolut |
| A6 | LDX - Seite Null | | CE | DEC - Absolut |
| A7 | NOP | | CF | NOP |
| A8 | TAY | | D0 | BNE |
| A9 | LDA - Unmittelbar | | D1 | CMP - (Indirekt),Y |
| AA | TAX | | D2 | NOP |
| AB | NOP | | D3 | NOP |
| AC | LDY - Absolut | | D4 | NOP |
| AD | LDA - Absolut | | D5 | CMP - Seite Null, X |
| AE | LDX - Absolut | | D6 | DEC - Seite Null, X |
| AF | NOP | | D7 | NOP |
| B0 | BCS | | D8 | CLD |
| B1 | LDA - (Indirekt),Y | | D9 | CMP - Absolut,Y |
| B2 | NOP | | DA | NOP |
| B3 | NOP | | DB | NOP |
| B4 | LDY - Seite Null, X | | DC | NOP |
| B5 | LDA - Seite Null, X | | DD | CMP - Absolut, X |
| B6 | LDX - Seite Null,Y | | DE | DEC - Absolut, X |
| B7 | NOP | | DF | NOP |
| B8 | CLV | | E0 | CPX - Unmittelbar |
| B9 | LDA - Absolut,Y | | E1 | SBC - (Indirekt, X) |
| BA | TSX | | E2 | NOP |
| BB | NOP | | E3 | NOP |
| BC | LDY - Absolut, X | | E4 | CPX - Seite Null |
| BD | LDA - Absolut, X | | E5 | SBC - Seite Null |
| BE | LDX - Absolut,Y | | E6 | INC - Seite Null |
| BF | NOP | | E7 | NOP |
| C0 | CPY - Unmittelbar | | E8 | INX |
| C1 | CMP - (Indirekt, X) | | E9 | SBC - Unmittelbar |
| C2 | NOP | | EA | NOP * |
| C3 | NOP | | EB | NOP |

EC CPX - Absolut
ED SBC - Absolut
EE INC - Absolut
EF NOP
F0 BEQ
F1 SBC - (Indirekt),Y
F2 NOP
F3 NOP
F4 NOP
F5 SBC - Seite Null, X
F6 INC - Seite Null, X
F7 NOP
F8 SED
F9 SBC - Absolut,Y
FA NOP
FB NOP
FC NOP
FD SBC - Absolut, X
FE INC - Absolut, X
FF NOP

* Es gibt nur diesen einen NOP-Befehl. Alle anderen NOP-Befehle sind für zukünftige Erweiterungen des 6502-Mikroprozessors vorgesehen und sollten deshalb nicht verwendet werden.