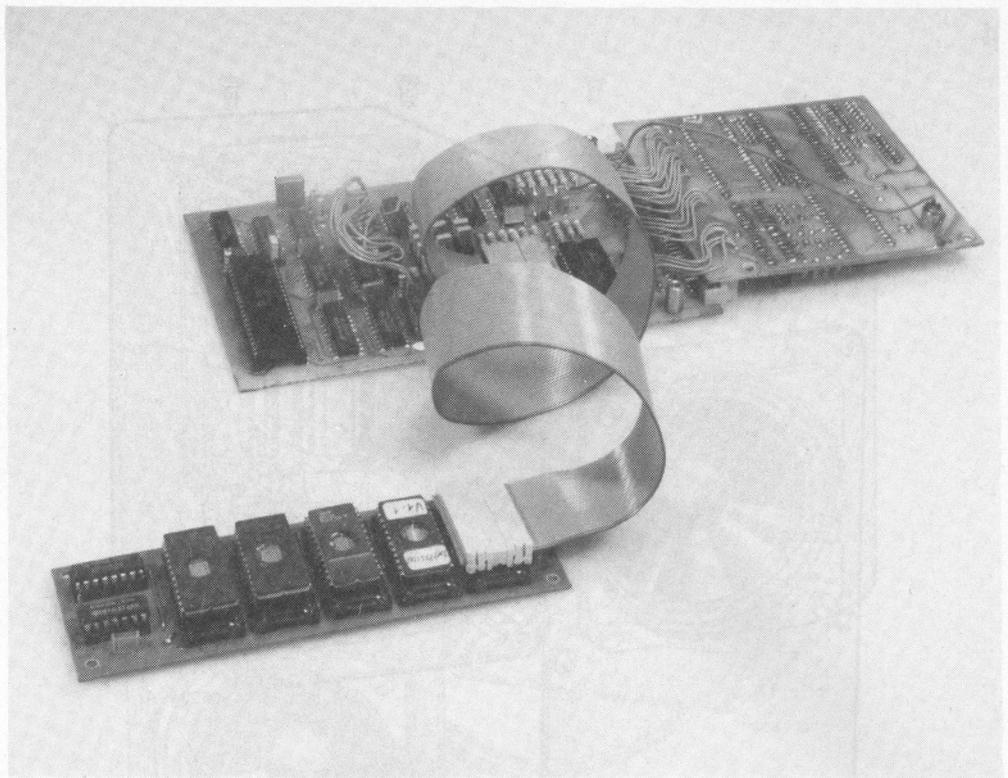


Foto. Hier haben wir die EPROM-Umschaltung mit der 6502-CPU-Platine (Elektor, November 1983) verbunden.



Vier EPROMs in einem Adreßbereich

EPROM-Umschaltung

Der Speicherbereich eines Mikrocomputers hat eine merkwürdige Eigenschaft: Zu Beginn — mit dem noch neuen Computer — scheint er sehr groß zu sein. Und dann fängt er auf einmal an zu schrumpfen. Natürlich nicht wirklich, in Wirklichkeit werden die eigenen Programme länger und länger... Mit dieser Schaltung wird der Speicher auf einen Schlag wieder größer — aber leider nur für EPROMs: Es handelt sich um einen besonders einfachen und eleganten "Soft-switch", der für alle EPROMs der 25XX- und der 27XX-Reihen geeignet ist.

Tabelle

Byte				
Hex	Dez.	D1	D0	EPROM
0	0	0	0	1
1	1	0	1	2
2	2	1	0	3
3	3	1	1	4
4	4	0	0	1
5	5	0	1	2
6	6	1	0	3
7	7	1	1	4
8	8	0	0	1
9	9	0	1	2
A	10	1	0	3
B	11	1	1	4
C	12	0	0	1
D	13	0	1	2
E	14	1	0	3
F	15	1	1	4

Tabelle. Die Tabelle erleichtert die Umschaltung zwischen den EPROMs per Software-Befehl (siehe Text).

Daß man in ein EPROM nicht einfach Daten einschreiben kann, sondern sie programmieren muß, weiß jeder Computeranwender (hoffentlich). Aber wer oder was hindert einen eigentlich, ein Datenwort in den Adreßbereich zu schreiben, in dem das (die) EPROM(s) liegen? Schließlich kann man damit nichts "kaputt machen" — höchstens das Programm stürzt ab. Bei dieser Schaltung wird bewußt ein Datenwort in den EPROM-Adreßbereich geschrieben: Das EPROM stört sich nicht dran, aber die Dekoderlogik der Schaltung wertet die Information aus und selektiert eines von vier EPROMs. Dieses EPROM bleibt selektiert, bis durch einen erneuten Schreibzugriff auf den EPROM-Adreßbereich ein anderes EPROM gewählt wird. Also eine sehr elegante und auch leicht zu programmierende Lösung. Kommen wir deshalb gleich zu den Details.

Schaltung

Ein Blockschaltbild sparen wir uns diesmal: Das Schaltbild selbst besteht aus lauter "Blöcken". Zunächst erkennt man 5 Sockel. Davon sind die Sockel EPROM1...EPROM4 für die maximal 4 EPROMs vorgesehen, die das "switch-board" aufnehmen kann. In den fünften Sockel ("Master") wird ein DIL-Stecker, in den man das eine Ende eines Stückes Flachbandkabel preßt, gesteckt. Über das Flachkabel und einen weiteren DIL-Stecker an dessen anderem Ende wird die Schaltung mit dem Sockel des ursprünglichen EPROMs verbunden, das EPROM selbst wird in einen der Sockel auf der Zusatzplatine gesteckt. Auf dieser Platine sind mit wenigen Ausnahmen die gleichnamigen EPROM-Anschlüsse miteinander verbunden. Die Anschlußbelegung in () bezieht sich auf

24-polige EPROMs, die ohne Klammern auf 28-polige EPROMs. Die Ausnahmen sind:

- Die Anschlüsse \overline{OE} (Output Enable) der EPROM-Sockel 1-4 sind mit der Selektionslogik verbunden. Diese sorgt dafür, daß höchstens eine der Leitungen auf logisch 0 liegt, dadurch wird das gewünschte EPROM und nur dieses selektiert.

- Die Anschlüsse Pin 20 und Pin 22 des "Master"-Sockels führen auf eine Brücke (Pin 10 ist auch mit den gleichnamigen Anschlüssen der anderen Sockel verbunden). Bei 27XX-EPROMs muß die Brücke A-B, bei 25XX-EPROMs dagegen die Brücke A-C gelegt werden.

- Verwendet man 24-polige EPROMs, muß die Brücke VCC24 gelegt werden, bei 28-poligen EPROMs die Brücke VCC28. Im ersten Fall kann C2 entfallen, im zweiten C3.

Damit ist der rein optisch größte Teil der Schaltung schon beschrieben. Kommen wir zur Selektionslogik. Diese besteht aus zwei Flip-Flops (IC1, 74LS74) und zwei 2-bit-Binärdkodern (IC2, 74LS139). Ihren Funktionszusammenhang zeigt Bild 2. Im

Zeitraum T1 schreibt der Computer irgendwelche Daten in den RAM-Bereich: Die Schreibimpulse (NWDS, Negative Write Data Strobe) beeinflussen die Selektionslogik nicht. Diese Schreibimpulse haben bei unterschiedlichen Computern eine unterschiedliche Bezeichnung: R/\overline{W} , \overline{WR} und so fort. Man muß sich also selbst den richtigen "heraussuchen". Im Zeitraum T2 will der Computer auf das nun "vervielfachte" EPROM zugreifen — lesend, versteht sich. Dadurch wird der Dekoder 2 über den Dekoder 1 (beide in IC2) freigegeben. Einer der Ausgänge $2Y0 \dots 2Y3$ geht auf logisch 0 und selektiert dadurch eines der EPROMs. Welches, hängt von den Ausgangspegeln der beiden Flip-Flops ab.

Die Flip-Flop-Eingänge sind mit den Datenbusleitungen D0 und D1 verbunden. Im Zeitraum T3 erhält der Computer vom Anwender den Befehl, in den EPROM-Bereich zu schreiben. Das kann zum Beispiel durch einen POKE-Befehl geschehen. \overline{OE} und die NWDS-Leitung gehen auf logisch 0 und dann wieder auf logisch 1. Dadurch wird der Ausgang $1Y0$ aktiv, auch er geht erst auf logisch 0 und dann

1

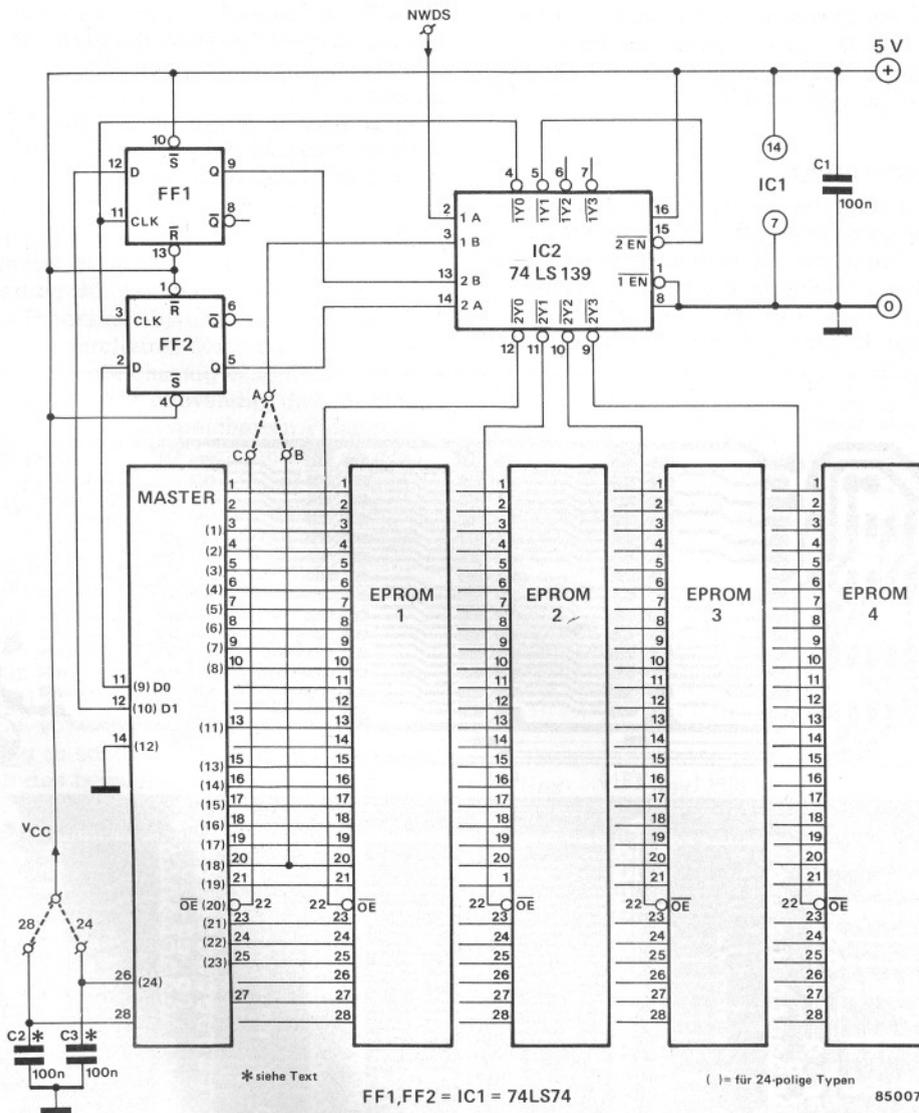


Bild 1. Das Schaltbild der EPROM-Umschaltung ist sehr übersichtlich: Mit Ausnahme weniger Leitungen können alle EPROM-Anschlüsse miteinander verbunden werden.

2

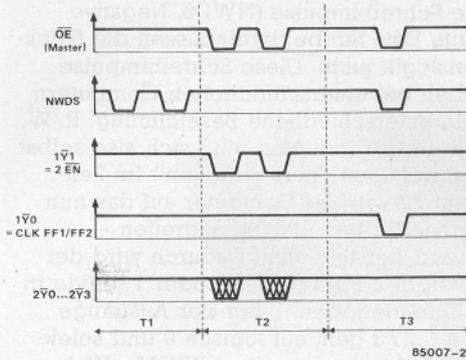


Bild 2. Das Zeitablaufdiagramm verdeutlicht die Funktionsweise der Selektionslogik.

wieder auf logisch 1. Da dieser Ausgang mit den Takteingängen (Clk) der beiden Flip-Flops verbunden ist, wird mit der ansteigenden Flanke an $1\bar{Y}0$ die Information auf der jeweiligen Datenbusleitung in das "zugehörige" Flip-Flop übernommen. Die Tabelle zeigt den Zusammenhang zwischen dem Byte auf dem Datenbus (hexadezimal und dezimal), den logischen Pegeln auf den Datenbusleitungen $D0$ und $D1$ und dem bei den folgenden Lese-Zugriffen angesprochenen EPROM. Durch die Flip-Flops bleibt dieses EPROM solange selektiert, bis ein erneuter Schreibversuch in den EPROM-Adressbereich erfolgt.

Anwendung

Bild 3 zeigt die Platine. Außer den beiden bereits genannten Brücken müssen auf der Platine noch 6 weitere gelegt werden: Dadurch reichte eine einseitige Platine. Alles weitere zum Bau ist oben schon gesagt. Es fehlt höchstens noch der Hin-

weis: Die Betriebsspannung für die Schaltung übernimmt der Computer über das Flachbandkabel.

Aus der Natur der Schaltung sollte deutlich sein: War das ursprüngliche EPROM zum Beispiel ein 2732, muß man auch auf der Platine den gleichen Typ verwenden — es sei denn, man "strickt" die Schaltung selbst entsprechend um. Typische Anwendungen für diese Schaltung sind:

- Laden eines umfangreichen Betriebssystemes aus den EPROMs ins RAM — statt von der Diskette. Das geht sehr schnell, und das Betriebssystem kann nicht mehr aus Versehen verloren gehen. Dafür muß man natürlich für den entsprechenden Computer ein passendes Umladeprogramm schreiben, diese Anwendung setzt also gute Programmierkenntnisse voraus.
- Vier "Bänke" mit Utility-Programmen statt nur einer fest im Computer speichern, oder (maximal) 4 Programmiersprachen.
- Umschaltzeichensätze auf einer VDU-Karte (oder für den Zeichengenerator eines Einplatinen-Computers).
- Umschaltung zwischen verschiedenen Tastaturbelegungen (Umschaltung durch Drücken eines Tasters, der OE auf Masse legt, und gleichzeitige Betätigung einer Zeichentaste. Der (negative) Strobe-Impuls wird mit NWDS verbunden, $D0$ und $D1$ mit entsprechenden Datenleitungen).
- Verschiedene Spiele werden durch einen kurzen Befehl aufgerufen, statt sie von der Kassette zu laden.

Stückliste

- C1, C2 oder C3 = 100 n*
- IC1 = 74LS74
- IC2 = 74LS139

* = insgesamt 2, siehe Text

außerdem:

- Platine 85007
- 5 Sockel, 24-pol. oder 28-pol. (nach Bedarf)
- 2 DIL-Stecker, für Preßverbindung, 24-pol. oder 28-pol.
- Flachbandkabel, 24-pol. oder 28-pol.

Bild 3. Platine und Bestückungsdruck für die EPROM-Umschaltung.

