

Einen Speicher kann man sich ähnlich wie eine Bibliothek vorstellen. Jede Menge Regale mit Büchern, in denen Informationen festgehalten sind. Was uns an dieser Bibliothek besonders interessiert, ist ihre Kartei und vor allem wie sie gegliedert ist, denn nur über eine gut geführte Kartei kommt man halbwegs schnell an das Buch, das man sucht.

# Adreßdekodierung

Mit Z 80  
und 6502

Da hat man nun einen Katalog mit zigtausenden von Büchern der verschiedensten Gebiete vor sich. Wenn eine Bibliothek etwas wert ist, hat sie natürlich auch Elektronik-Bücher. Die findet man beispielsweise unter "E". Elektronik-Bücher über Rechner haben vielleicht die Kennbuchstaben "ER", Bücher über Analogtechnik finden sich unter "EA". In der Datenverarbeitung ist der Buchstabe E die Information mit der höheren Wertigkeit, weil sie einen Oberbegriff zu den untergeordneten Buchstaben R und A darstellt. Diese Reihe von Bezeichnungen läßt sich beliebig erweitern, wobei jeder zusätzliche Buchstabe von geringerer Wertigkeit als der vorhergehende ist, schon allein deshalb, weil jeder Zusatzbuchstabe die in Frage kommende Büchermenge weiter verringert. Zum Abschluß dieser Betrachtungen noch ein simples Beispiel: Viele Händler zeichnen ihre Waren viel lieber mit 9,98 DM als mit 10 DM aus. Was soll das? Die Information mit der höchsten Wertigkeit, also die vollen DM, scheint sehr unterschiedlich zu sein, in Wirklichkeit beträgt der Unterschied zwischen den beiden Preisen lächerliche 2 Pfennige. Trotzdem kommt einem das Produkt für 9,98 billiger vor als das für 10 DM – und das nur, weil man die Information mit der geringeren Wertigkeit (die "Pfennige") weniger beachtet.

## Verzweigung und doppelte Adressierung

Den Speicher eines Rechners kann man sich etwa wie einen Stapel aus vielen kleinen Kästchen vorstellen. Diese Kästchen enthalten 8 der kleinsten Informationseinheiten (bit), auf die man aber nicht einzeln zugreifen kann. Solche zusammenhängenden "Bits" werden als Wort bezeichnet. Zu einem 8-bit-Wort sagt man auch: 1 Byte. Das Datenwort gelangt vom Speicher über den Datenbus ins Innere des Computersystems. Der Datenbus muß deshalb 8 parallele Leitungen (D0...D7) haben, für jedes Bit eine. Damit der Prozessor auf diese Worte im Speicher zugreifen kann, "verständigt" er sich mit dem Speicher über einen Adreßbus mit 16 Leitungen (A15...A0).

In Bild 1 sieht man die sechs niederwertigsten Bits (A5...A0), die sich wie die Äste eines Baumes immer stärker verzweigen. Ob man sich an einer Verzweigung nach links oder rechts wendet ist egal, denn mit jedem Schritt nähert man sich einem Speicherplatz. Die Entscheidung links oder rechts wird durch einen logischen Pegel dargestellt; nur

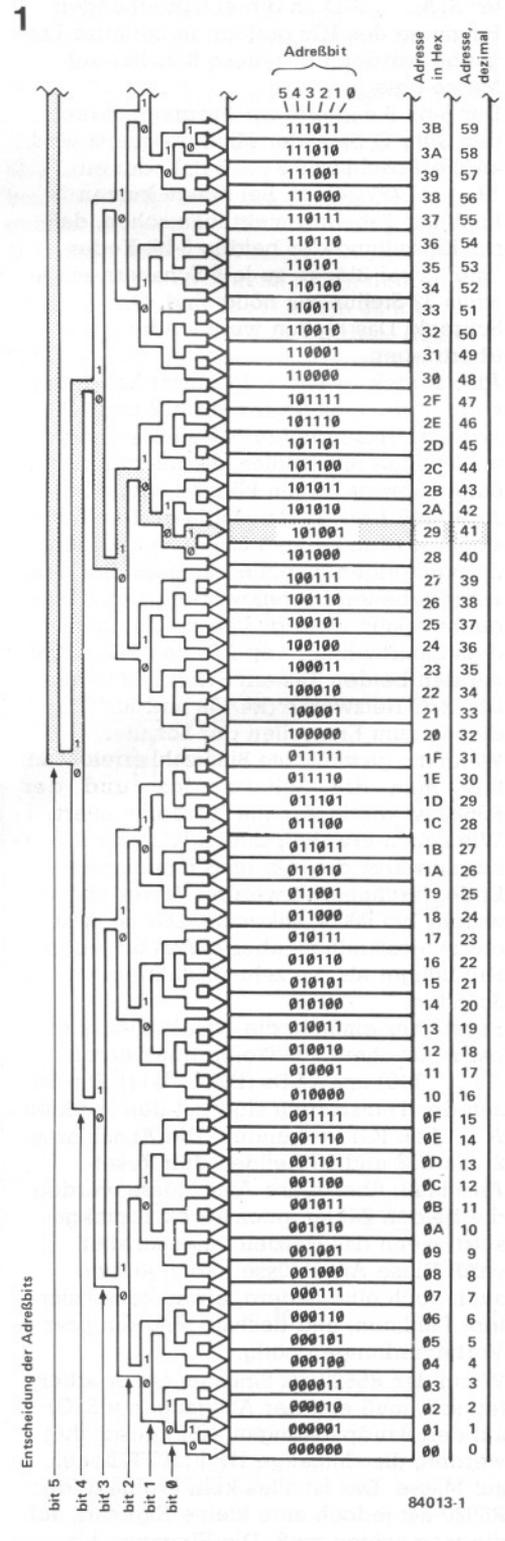


Bild 1. Dieses "Baumdiagramm" mit 6 niederwertigen Adreßbits zeigt, wie man mit der Festlegung des logischen Pegels eines Bits je nach Wertigkeit des Bits einen mehr oder weniger großen Bereich dekodieren kann.

die Pegel 1 oder 0 sind zulässig. Je höherwertiger ein Adreßbit ist, desto größer ist auch der Speicherbereich, den es adressiert. Wenn also die Bits 5 und 4 in Bild 1 logisch 0 sind, belegt Bit 3 (ebenfalls logisch 0) den Bereich 00 ... 07. Ist dieses Bit logisch 1, gibt es nur den Weg auf Bereich 08 ... 0F frei. Wenn Bit 4 jetzt logisch 1 wird und Bit 5 so bleibt wie es ist, kommt man mit Bit 3 zu den Bereichen 10 ... 17 und 18 ... 1F.

Angenommen der logische Pegel von Bit 3 ist einmal nicht definiert und die Bits 5 und 4 sind logisch 0, dann sind die Bereiche, die vorher durch den Pegel von Bit 3 eingegrenzt wurden, nicht mehr eindeutig auszumachen. Dadurch werden die Speicherbereiche 00 ... 07 und 08 ... 0F verwechselt, die Adreßworte 00 und 08 haben plötzlich das gleiche Ziel, genauso die Worte 01 und 09, 02 und 0A ... 07 und 0F. So etwas nennt man auch *doppelte Adressierung*. Je nach Wertigkeit des nicht definierten Bits, ist der doppelt adressierte Speicherbereich mehr oder weniger groß.

$$2^{16} = 65536$$

In Bild 2 sieht man, welchen Einfluß die höherwertigen Adreßleitungen (A15 ... A10) auf die Verteilung des adressierbaren Speicherbereichs haben. Die Mengen, die mit dem Buchstaben "K" bezeichnet werden, sind immer *Vielfache von 1024* (und nicht von 1000). Damit wird die Anzahl der Speicherzellen angegeben, die mit den ersten 10 Adreßleitungen (A9 ... A0;  $2^{10} = 1024$ ) adressiert werden. Wenn es sich also um Speicherplätze handelt, bezeichnet "K" immer 1024 Byte und nicht bit. Und jetzt zurück zu Bild 2.

Die Adreßleitung A15, egal ob sie logisch 0 oder 1 ist, teilt den gesamten, adressierbaren Speicherbereich mit Hilfe von 16 Leitungen ( $2^{16} = 65536$ ) in zwei gleiche Blöcke von je 32768 Worten auf. Innerhalb jedes Blocks findet mit Leitung A14 die nächste Aufteilung statt; es entstehen insgesamt 4 Blöcke zu je 16384 Worten. So geht das

ADRESSEN		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEZ.	HEX																
0	0000																0000
↓	↓																↓↓↓↓
15	000F	0000	0000	0000	0000												1111
↓	↓																↓↓↓↓
16	0010	0000	0000	0000	0001												0000
↓	↓																↓↓↓↓
31	001F																1111
↓	↓																↓↓↓↓
32	0020	0000	0000	0000	0												0000
↓	↓																↓↓↓↓
63	003F																1111
↓	↓																↓↓↓↓
64	0040	0000	0000	0000	00												0000
↓	↓																↓↓↓↓
127	007F																1111
↓	↓																↓↓↓↓
128	0080	0000	0000	0000	000												0000
↓	↓																↓↓↓↓
255	00FF																1111
↓	↓																↓↓↓↓
256	0100	0000	0001	0000	0000												0000
↓	↓																↓↓↓↓
511	01FF																1111
↓	↓																↓↓↓↓
512	0200	0000	0	0000	0000												0000
↓	↓																↓↓↓↓
1023	03FF																1111
↓	↓																↓↓↓↓
1024	0400	0000	00	0000	0000												0000
↓	↓																↓↓↓↓
2047	07FF																1111
↓	↓																↓↓↓↓
2048	0800	0000	000	0000	0000												0000
↓	↓																↓↓↓↓
4095	0FFF																1111
↓	↓																↓↓↓↓
4096	1000	0001	0000	0000	0000												0000
↓	↓																↓↓↓↓
8191	1FFF																1111
↓	↓																↓↓↓↓
8192	2000	001	0	0000	0000	0000											0000
↓	↓																↓↓↓↓
16383	3FFF																1111
↓	↓																↓↓↓↓
16384	4000	00	00	0000	0000	0000											0000
↓	↓																↓↓↓↓
32767	7FFF																1111
↓	↓																↓↓↓↓
32768	8000	000	000	0000	0000	0000											0000
↓	↓																↓↓↓↓
65535	FFFF																1111

Tabelle 1. Mit 16 Adreßleitungen erhält man einen adressierbaren Bereich von 65536 Worten. In dieser Tabelle wird deutlich, wie der Speicherbereich in mehr oder weniger große Blöcke aufgeteilt wird, je nach Wertigkeit des entsprechenden Bits.

weiter bis Leitung A10, die Blöcke sind jetzt nur noch 1024 Worte groß. Wenn der logische Pegel einer Adreßleitung einmal nicht definiert ist, werden zwei sonst getrennte Blöcke durcheinander geworfen. Beispiel: Der logische Pegel von A15 ist nicht definiert, also werden die Adressen 0 und 32768, die Adressen 1 und 32769 usw. verwechselt. Beim Adressieren nie vergessen, daß man immer von 0 aus hochzählt, gleichgültig welche Basis (dezimal, hexadezimal, binär) zugrunde liegt. In Tabelle 1 sind noch mal alle 16 Adreßleitungen und ihre 65536 Kombinations-

2

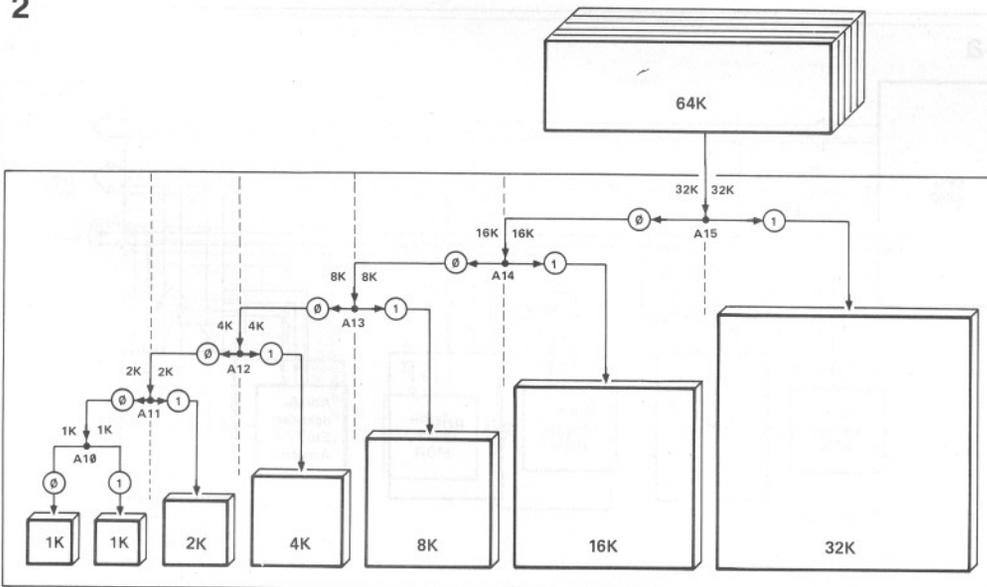


Bild 2. Hier ist zu sehen, wie der adressierbare Bereich mit den höherwertigen Bits in Blöcke aufgeteilt werden kann. Mit der Leitung A15 werden zwei Blöcke zu je 32 K "erzeugt", die selbst mit Leitung A14 wieder in je zwei Blöcke zu 16 K aufgeteilt werden usw.

möglichkeiten mit den zugehörigen Adressen zusammengefaßt. Zwar sieht es so aus, als ob diese Tabelle linear ist, das stimmt aber nicht, weil die Wertigkeit der Leitungen von rechts nach links zunimmt. Die höherwertigen Adreßbits nehmen mehr Speicher-raum ein, was man schon an der Größe der dekodierten Zone (in den Spalten unter 'Adressen' angegeben) sehen kann.

### Die Freigabesignale

Bis jetzt haben wir die Adressierung nur unter dem Aspekt der räumlichen Aufteilung betrachtet. Die meisten der ICs, mit denen man so zu tun hat, sind mit weniger als 16 Adreßleitungen ausgestattet, je nach Kapazität. Aus Bild 2 kann man zum Beispiel entnehmen, daß ein Block mit 4 K (ein 2732-EPROM beispielsweise) 12 Adreßleitungen haben muß. Jedes der 4096 Worte wird durch einen internen Adreßdekoder adressiert. Das gleiche gilt für einen Block von 2 K (das RAM 6116), dessen 11 Adreßleitungen (A10...A0) die 2048 Speicherzellen adressieren. Die eigentliche Adreßdekodierung ist genau genommen nicht

diese interne Dekodierung des Speicherbereichs, sondern die Einordnung eines Blocks in den Speicherbereich (durch den Prozessor). Bei den weiteren Erklärungen wollen wir uns auf den 6502 und den Z 80 beschränken, die beide 16 Adreßleitungen haben und damit bis zu 64 K Speicherplätze dekodieren können.

Jedes IC hat außer den Adreßleitungen noch eine oder mehrere Freigabe-Leitungen, die im allgemeinen logisch 0 sind – das sieht man auch an dem Negierungs-Strich über der Bezeichnung des jeweiligen Anschlusses. Die interne Adressierung funktioniert nur, wenn ein Freigabesignal am IC liegt und genau in diesem Moment Daten am Datenbus vorhanden sind. Um ein Freigabesignal zu erhalten, verknüpft man die hochwertigen Adreßleitungen zu Steuersignalen, die für den zeitlichen Ablauf der Befehlsverarbeitung unerlässlich sind (siehe Bild 3). Diese Steuersignale sind bei jedem Prozessor anders. Beim 6502 gibt es:

- das Taktsignal  $\Phi 2$ , das eine Freigabe des Schreib- und Lesezyklus nur während der zweiten Hälfte jedes Taktzyklus des Prozessors zuläßt;
- das  $R/\bar{W}$ -Signal, mit dem Lese- und Schreiboperationen unterschieden werden können.

Der Z 80 hat folgende Signale:

- $\bar{WR}$  und  $\bar{RD}$  zum Unterscheiden von Schreiben und Lesen;
- $\bar{MREQ}$  und  $\bar{IOREQ}$  zur Unterscheidung von Operationen im Speicher und Operationen an den Ein-/Ausgabemodul, für die der Z 80 spezielle Befehle hat.

Diese Unterschiede sind auch in den Bildern 4a und 4b dargestellt. Die Freigabesignale aus den hochwertigen Adreßsignalen und den Steuersignalen werden hier alle  $\bar{CS}$  (Chip Select) genannt. Aus Bequemlichkeit nimmt man an, daß sie immer bei logisch 0 aktiv sind. Bei manchen Systemen und Herstellern findet man allerdings auch andere Kürzel als  $\bar{CS}$ ; manchmal sind die Freigabesignale bei logisch 1 aktiv.

Bevor wir zu den logischen Verknüpfungen kommen, aus denen die Freigabesignale erzeugt werden, ist es vielleicht ganz gut,

Bild 3. Um einen Speicher adressieren zu können, genügt es nicht, einen Adreß- und einen Datenbus zu haben; unbedingt notwendig sind noch die Steuersignale, damit Schreib- und Lesevorgänge richtig ablaufen.

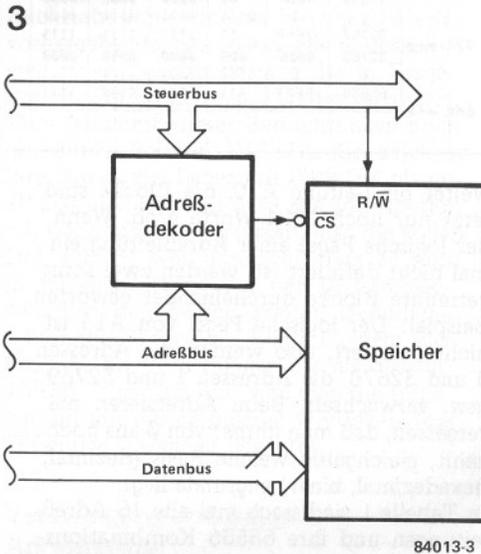
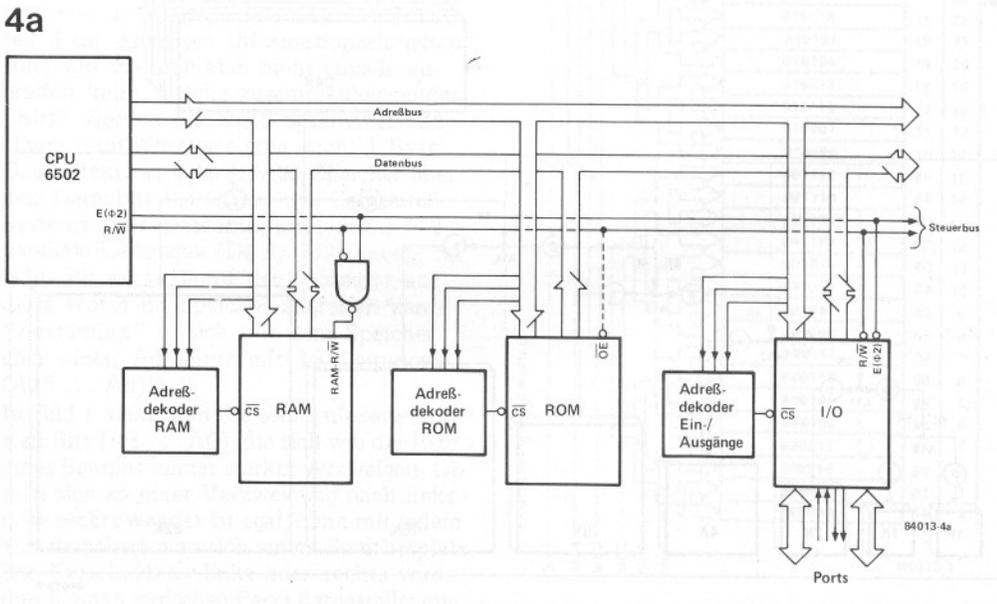
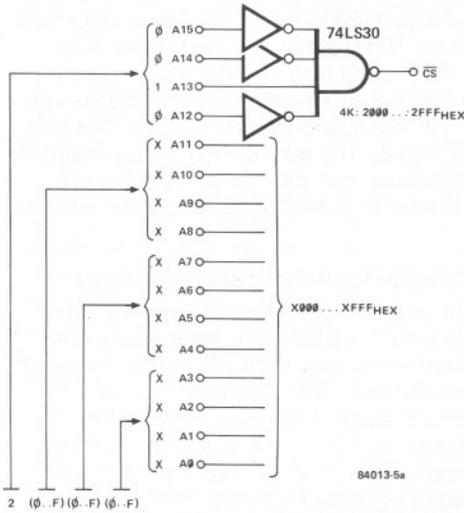


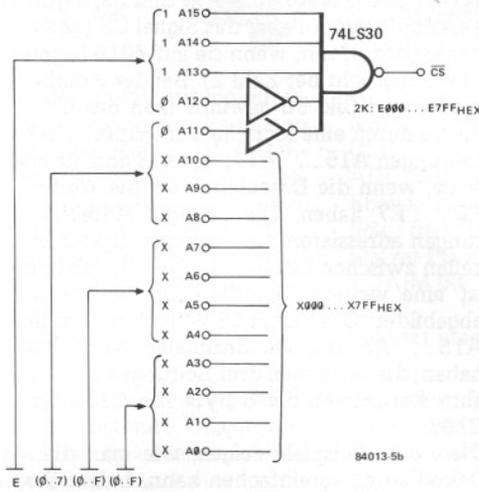
Bild 4a. Der 6502 hat weder Befehle noch spezielle Signale, um die Speicher der Ein-/Ausgabemodul auseinanderhalten zu können. Daher braucht man unbedingt ein Taktsignal  $\Phi 2$  und eine Umschaltung für Lese- und Schreibbetrieb ( $R/\bar{W}$ ).



5a



5b



Adreßdekodierung  
Elektor Januar 1984

auf die Bedeutung der Hex-Zahlen einzu-  
gehen.

Wir haben also 16 Adreßleitungen, die in  
4 Gruppen zu je 4 Leitungen aufgeteilt  
werden. Jeder Gruppe mit vier Leitungen  
entspricht eine Hexadezimalzahl (auch Hex-  
zahl) von 0...F, das entspricht 0...15  
im Dezimalkode. Bei der Adresse 4A2F  
entspricht die 4 dem Binärwort auf den  
Leitungen A15, A14, A13 und A12 (0100),  
das A dem Binärwort auf den Leitungen  
A11, A10, A9 und A8 (1010), die 2 den  
Informationen auf den Leitungen A7, A6,  
A5 und A4 (0010) und das F dem Binärwort  
auf den Leitungen A3, A2, A1 und A0  
(1111). Diese "Dekodierung" ist ziemlich  
einfach und ermöglicht es, sofort den  
Zustand der 16 Adreßleitungen herauszu-  
finden, der einer hexadezimalen Adresse  
entspricht.

5c

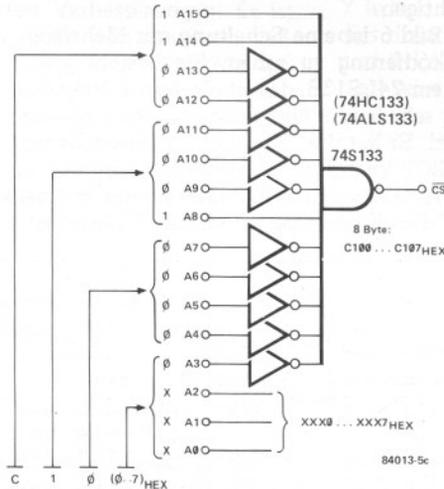


Bild 5. Beispiele für eine  
Dekodierung mit festen  
logischen Verknüpfungen  
für 4 K, 2 K und 8 Byte:  
Je kleiner der adressierte  
Bereich, desto höher die  
Anzahl der verknüpften  
Adressierungs-Signale.

### Feste logische Verknüpfungen

Wir kommen jetzt zu dem Teil der Adreß-  
dekodierung, der mit Hilfe von mehr oder  
weniger komplexen logischen Verknüp-  
fungen gemacht wird. Angenommen ein

4b

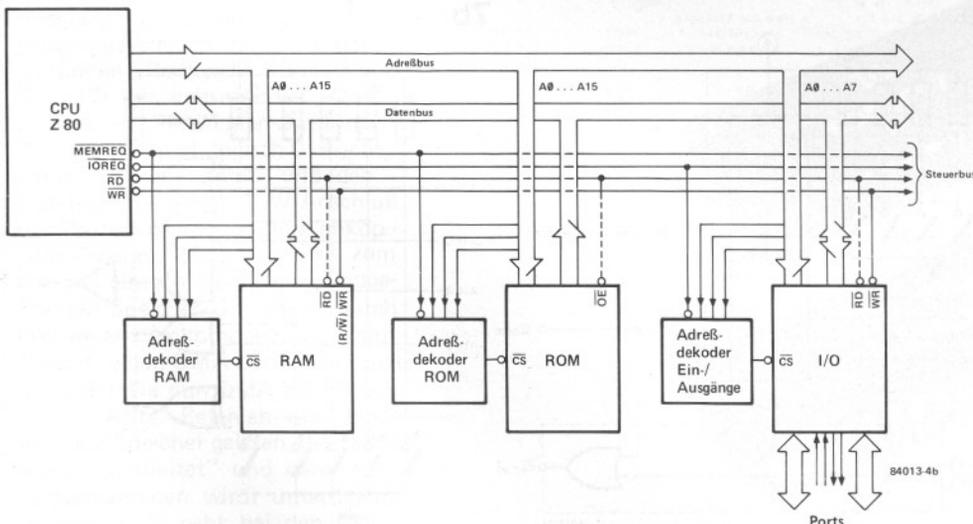


Bild 4b. Die interne Struk-  
tur eines Z80 ist mit der  
eines 6502 durchaus  
vergleichbar, nur hat der  
Z80 wesentlich mehr und  
speziellere Steuerbefehle.  
Der zeitliche Ablauf dieser  
Signale soll hier aber nicht  
weiter beschrieben werden.

Speicher soll zwischen den Adressen 2000 und 2FFF adressiert werden. Die Leitungen A15...A12 werden, wie in Bild 5a, logisch verknüpft, liefern aber das Signal  $\overline{CS}$  (aktiv bei logisch 0) nur, wenn sie auf 0010 liegen (das entspricht der Zahl 2). Bei der Kombination aus Bild 5b gewinnt man das  $\overline{CS}$ -Signal durch eine logische Verknüpfung der Leitungen A15...A11; dieses Signal ist erst aktiv, wenn die Datenleitungen die Werte E0...E7 haben. Die anderen Adreßleitungen adressieren die restlichen Speicherzellen zwischen E000 und E7FF. In Bild 5c ist eine weitere Dekodierungsmöglichkeit abgebildet: Das Signal  $\overline{CS}$  ist logisch 0, wenn A15...A3 den hexadezimalen Wert C10 haben; die restlichen drei Leitungen dienen zum Adressieren der 8 Byte von C100 bis C107.

Diese drei Beispiele zeigen, wie man die Dekodierung vereinfachen kann, indem die Anzahl der hochwertigen Adreßleitungen zur Erzeugung des Freigabe-Signals erhöht und gleichzeitig die Größe des adressierten Bereichs verringert wird.

Vor lauter Vereinfachung haben wir jedoch vergessen, die in der Praxis unbedingt notwendigen Steuersignale mit zu berücksichtigen.

In Bild 6 ist eine Schaltung mit Mehrfach-Dekodierung zu sehen. Sie besteht aus einem 74LS138, das ist ein 1-zu-8-Dekoder, der sehr häufig für solche Zwecke eingesetzt wird. Er hat drei Eingänge für binäre Daten (ABC) und drei Freigabe-Eingänge (G1, G2A, G2B). Das Signal G2A ist nur in einem 8-K-Block von C000 und DFFF aktiv; es

wird mit den Leitungen A15...A13 erzeugt. Der Eingang G2B erhält vom Z80 das Signal MREQ, vom 6502 eine logische 0. Durch Verknüpfen der Leitungen A12...A10 erhält man ein Binärwort von 3 bit, mit dem 8 aufeinanderfolgende Blöcke von je 1 K dekodiert werden können. Die acht  $\overline{CS}$ -Signale, die man daraus erhält, können zusammen mit den Steuersignalen  $\overline{WE}$ ,  $\overline{RD}$  oder  $R/\overline{W}$  zum Speicher geführt werden.

### Variable logische Verknüpfungen

Die bisher besprochenen Beispiele sind alle nicht variabel. Man kann eine Adreßdekodierung aber auch wie in Bild 7 gezeigt durchführen. Die Schaltung in Bild 7a enthält einen 4-bit-Vergleicher, dem das Binärwort A0...A3 über die Adreßleitungen A15...A12 zugeführt wird. Der 74LS85 vergleicht dieses Wort mit dem Binärwort, das die vier mit Masse verbundenen Schalter zusammen mit den vier Pull-up-Widerständen liefern. Wenn das Binärwort A0...A3 gleich dem Binärwort B0...B3 ist, wird der Anschluß 3 (A=B) logisch 1. Dieses Signal wird invertiert, tritt an die Stelle des Signals  $\overline{CS}$  und dekodiert einen Block von 4 K (X000...XFFF, wobei X die Hexzahl ist, die dem Binärwort B0...B3 entspricht). Eine ähnliche, programmierbare Adreßdekodierung kann man mit EXNOR-Gattern aufbauen, etwa so wie in Bild 7b. Die Ausgänge mit den offenen Kollektoren (74LS266) werden nur dann logisch 1, wenn beide Eingänge jedes Gatters auf gleichem logischen Pegel liegen. Jedes Gatter vergleicht ein Bit des Adreßwortes (von den Leitungen A15...A12) mit dem entsprechenden Bit des mit den Schaltern programmierten Binärwortes. Diese Art der Adreßdekodierung ist sehr anpassungsfähig. Außerdem kann man auch hier die Anzahl der hochwertigen Adreßleitungen vergrößern (gestrichelte Linien in Bild 7b) und den Block verkleinern, der mit einem  $\overline{CS}$ -Signal freigegeben wird.

Mit diesem letzten Beispiel schließen wir das Kapitel über die Adreßdekodierung (vorläufig) ab. Bestimmt haben wir noch einiges vergessen; trotzdem hoffen wir, daß ein paar Dinge klarer geworden sind.

Bild 6. Mit dem 1-zu-8-Dekoder 74LS138 kann ein Block von 8 K einfach in 8 Blöcke zu je 1 K aufgeteilt werden, wobei jeder Block sein eigenes  $\overline{CS}$ -Signal hat. Der zweite Freigabe-Eingang des Dekoders wird bei Z80-Systemen anders belegt als bei 6502-Systemen.

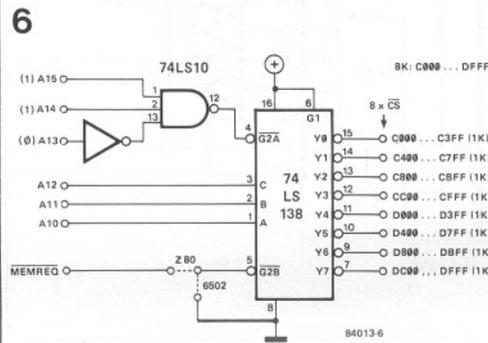


Bild 7. Bei einigen Anwendungen ist es sinnvoll, eine programmierbare Adressierung zu haben. Das wird hier mit einem Vergleichere erreicht, der dafür sorgt, daß das Binärwort der Leitungen A15...A12 und die Information von den Schaltern gleich sind.

