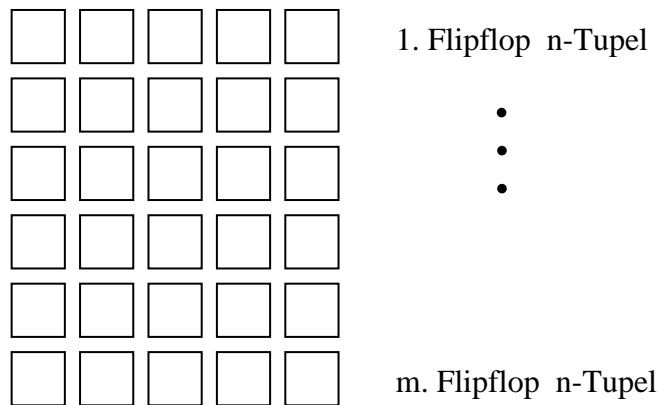


# Kapitel 7

## Versuch 700

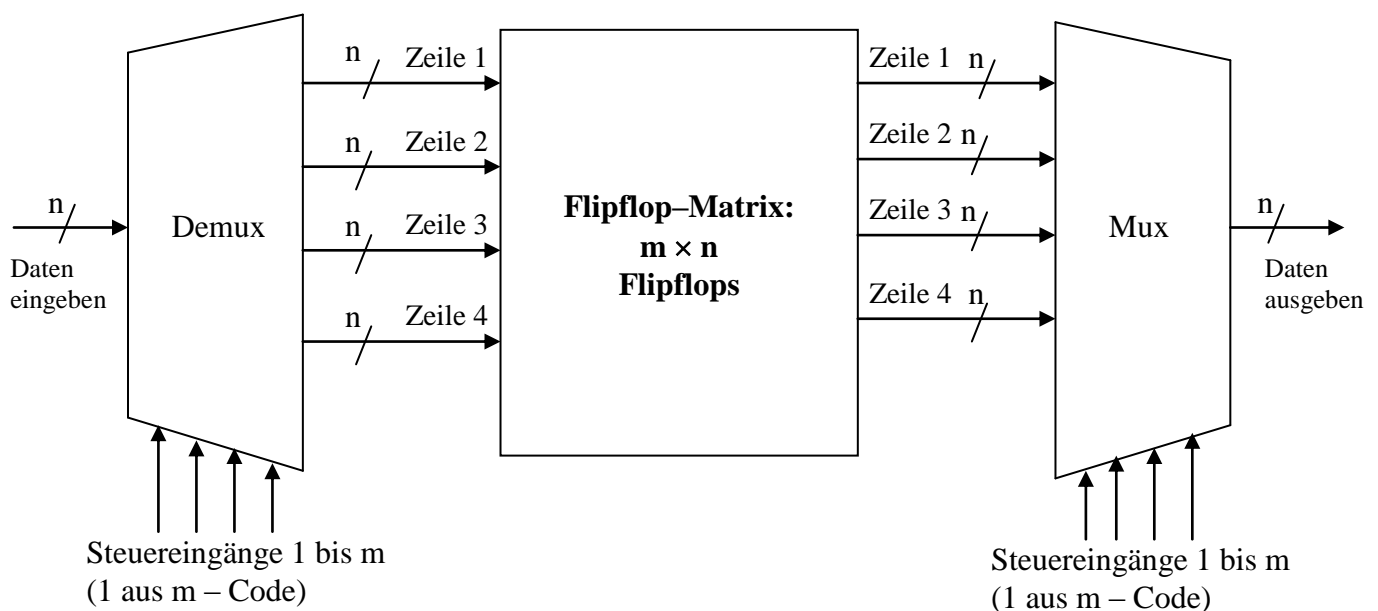
### Matrixbildung mit Speicherzellen–Zugriff über Multiplexer und Demultiplexer

Der Übergang vom einzelnen Flipflop zu einem Register entspricht dem Übergang von Buchstaben zu einer Zeile in handschriftlichen oder gedruckten Texten, oder dem Übergang von Skalarwerten zu einem Vektor in der Mathematik. Ein  $n$ -stelliges Register ist ein  $n$ -Tupel, dessen Elemente Flipflops sind. Man kann mehrere Fliflop–Tupel zu einem  $m$ -Tupel zusammenfassen und erhält dann im einfachsten Fall eine  $m \times n$  Matrix, deren Elemente Flipflops sind:

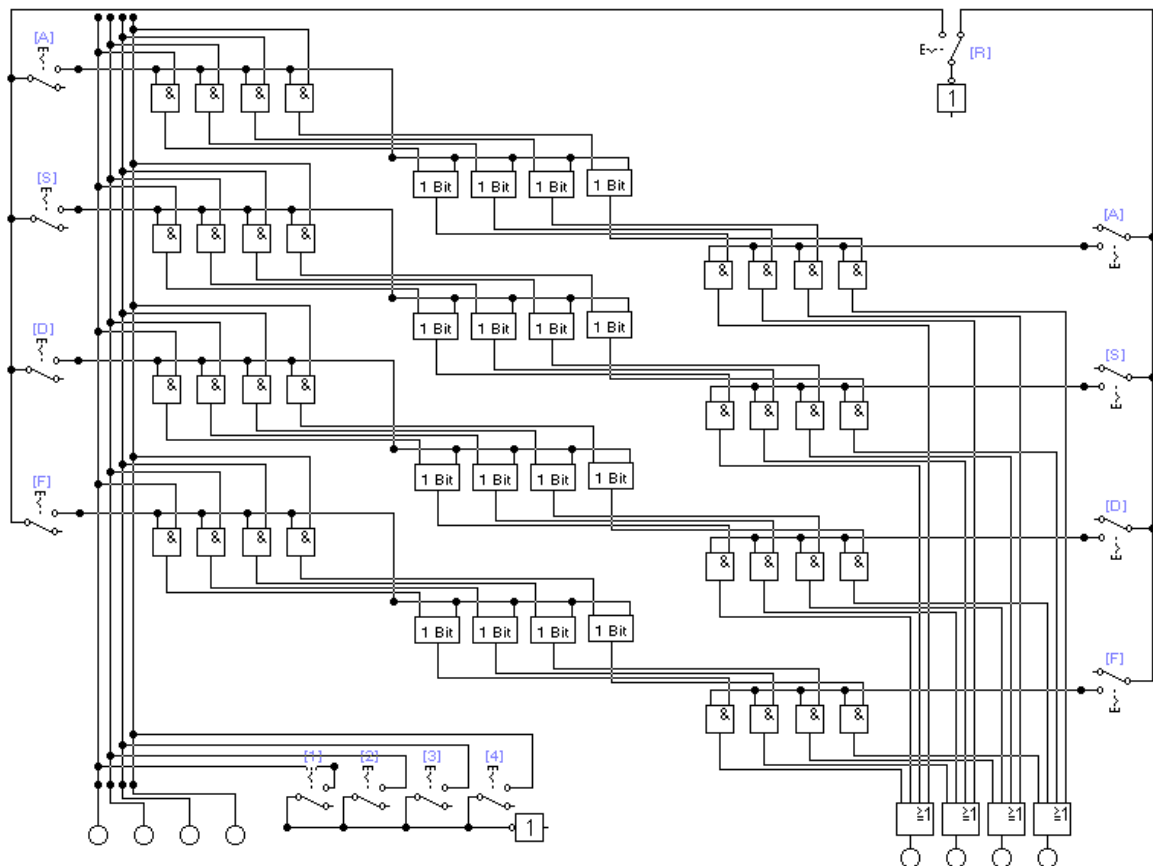


Jedes der  $m$  Flipflop  $n$ -Tupel kann ein  $n$ -stelliges Binärwort speichern.

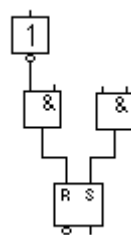
Die Flipflop–Matrix bildet die Grundstruktur adressierbarer Speicher mit wahlfreiem Zugriff (**R**andom **A**ccess **M**emory, **RAM**):



Jede Zeile der Flipflop-Matrix ist eine Speicherzelle. An den Steuereingängen des Demultiplexers kann mit einem Binärwort im 1-aus-n – Code eine dieser Speicherzellen ausgewählt und mit dem in den Demultiplexer eingegebenen Binärwort beschrieben werden. Entsprechend läuft der Lesevorgang über den Multiplexer ab. Die Zugriffszeit ist für alle Schreib- und Lesevorgänge gleich. In der Datei v700 finden Sie eine solche Speicherstruktur.



Vervollständigen Sie die Schaltung, in dem Sie das Schaltungsfragment



vervollständigen und ein 1-Bit-Speicherzellen Makro  erzeugen.

Testen Sie die Schaltung, in dem Sie die folgenden vier Binärworte unter den Adressen A, S, D und F in der funktionsfähigen Schaltung speichern und wieder auslesen.

Adresse	Binärwort
A	1000
S	0100
D	0010
F	0001

## Versuch 705

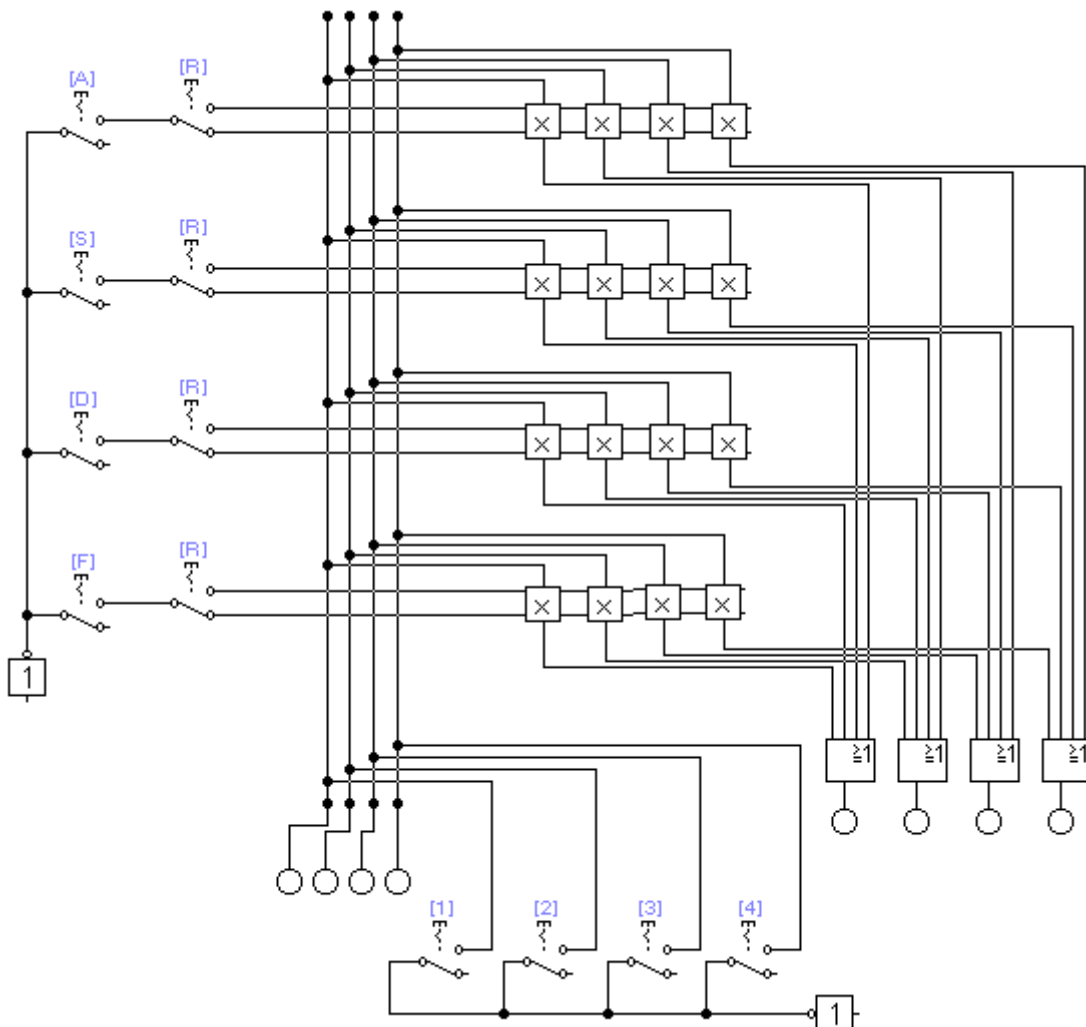
### Speicherstruktur mit verteiltem Multiplexer und Demultiplexer

Die obige Schaltung besteht aus drei klar getrennten Funktionseinheiten: Demultiplexer, Speichermatrix und Multiplexer. Sehen Sie sich ihre Speicherzelle „1 Bit“ noch einmal genauer an. Die beiden UND-Gatter in dieser Zelle könnten die Funktion des Demultiplexers übernehmen, in dem man in jedem „1 Bit“ – Makro den Teil des Demultiplexers unterbringt, der auf das speichernde Flipflop wirkt. Den Multiplexer könnte man in der gleichen Weise „dezentralisieren“.

Bauen Sie in der Versuchsdatei v705 in das Makro



eine Variante ihres im vorherigen Versuch entwickelten Makros „1 Bit“ so ein, dass die obige Speicherschaltung sich in der folgenden kompakteren Form aufbauen lässt.



Testen Sie ihre Schaltung wieder durch Ein- und Auslesen verschiedener Werte an verschiedenen Adressen.

## Versuch 710

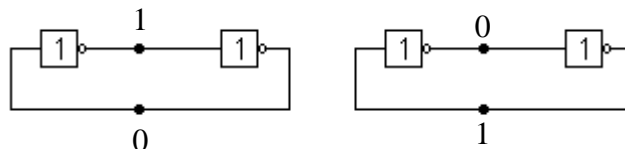
### Eine statische 1-Bit-MOS-RAM-Zelle

Speichernde Bauelemente der Speicherschaltungen des vorigen Abschnittes sind SR-Flipflops. Auf jede Speicherzelle dieser Schaltungen kann nach Eingaben einer Adresse direkt zugegriffen werden. Sie werden deshalb als Speicher mit wahlfreiem Zugriff (random access memories: RAM) bezeichnet und von Speichern mit sequentiellen Zugriff unterschieden. Standardbeispiele für die Speicher mit sequentiellen Zugriff sind Floppydisk und Harddisk, bei denen der Schreib-Lesekopf im Extremfall erst dann auf eine Speicherzelle zugreifen kann, wenn sich alle Zellen einer Spur an ihm vorbeibewegt haben.

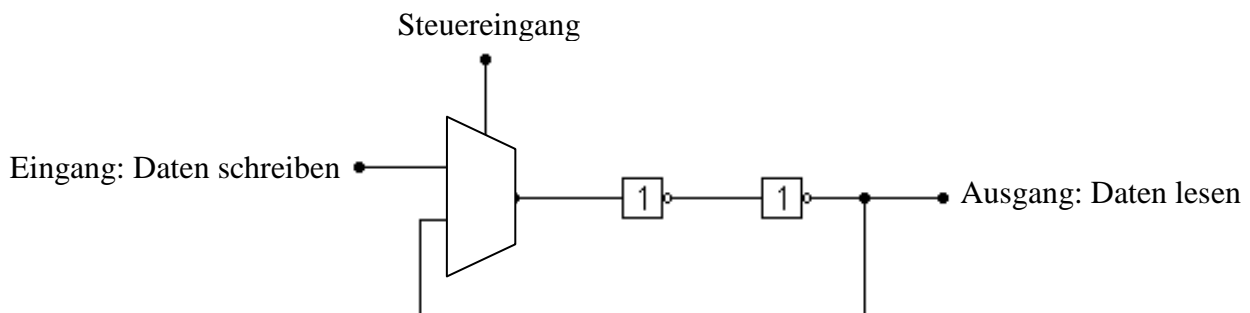
Das aus zwei NAND- oder NOR-Gattern bestehende SR-Flipflop hält einen gespeicherten Zustand konstant, bis die Versorgungsspannung abgeschaltet wird. Es wird deshalb als statisches Speicherelement bezeichnet. Unsere bisher untersuchten Speicherschaltungen gehören also zu den statischen Speichern mit wahlfreiem Zugriff (static random access memory: SRAM).

In einem der nächsten Abschnitte werden wir auch noch dynamische Speicher (dynamic random access memory: DRAM) untersuchen, deren Zellen auch bei konstanter Versorgungsspannung ihren Speicherinhalt verlieren, weil in ihnen das Potential, das den gespeicherten Binärzustand 1 repräsentieren soll, allmählich sinkt. Wenn in dynamischen Speicherzellen der Binärzustand 1 (bei konstanter Versorgungsspannung) dauerhaft gespeichert werden soll, muss deshalb von Zeit zu Zeit mit einer Spezialschaltung „aufgefrischt“ werden, damit das Potential niemals unter den Schwellwert für den Binärzustand 1 sinkt (Refreshing).

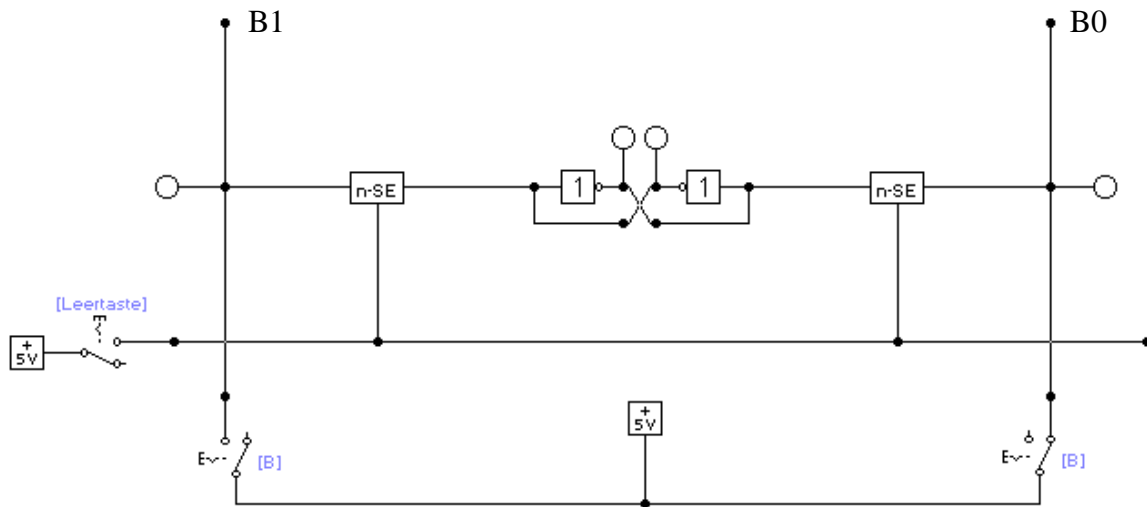
Für den Aufbau von grossen statischen Speichern eignen sich NAND- oder NOR-Basisflipflops nicht, weil sie zu viele Transistoren enthalten. Einer weniger aufwendigen 1-Bit-MOS-RAM-Zelle liegt das folgende Schaltungsprinzip zugrunde.



Sowohl bei einer 1 als auch bei einer 0 am Ausgang des ersten Inverters bewirkt die Rückkopplung eine Stabilisierung des jeweils „gespeicherten“ Wertes. Um aus dieser Rückkopplungsschaltung eine Speicherzelle zu konstruieren, müssen wir sie um einen Multiplexer ergänzen.



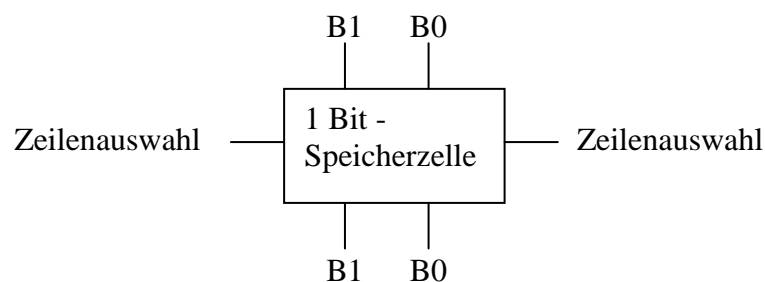
In der Datei v710 finden Sie diese Schaltungs-idee in einer modifizierten Form, wie man sie in statischen Halbleitern finden kann.



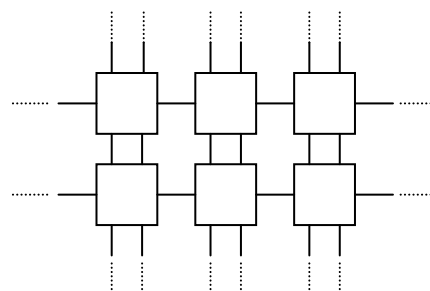
Bei einem Schreibvorgang werden die beiden Leitungen B1 und B0 immer „komplementär“ benutzt, d.h. sie haben entweder den Zustand  $(B1, B0) = (0, 1)$  oder  $(B1, B0) = (1, 0)$ . Diese Zustände können in die Speicherzelle geschrieben werden, wenn die beiden Transistoren „leitend“ („durchlässig“) sind, d.h. wenn die Steuerleitung (Leertaste) den Binärzustand 1 hat. Die Zelle speichert den geschriebenen Zustand, wenn die beiden n-SE nicht leiten.

Machen Sie sich nun die Funktionsweise der Schaltung klar, in dem Sie die verschiedenen Speicherzustände in die Speicherzelle schreiben.

Diese 1 Bit – Speicherzelle sei nun zu einer Blackbox zusammengefasst:



Man kann mit diesem Zellentyp eine Speicherschaltung mit einer  $m \times n$  Flipflop-Matrix bilden:

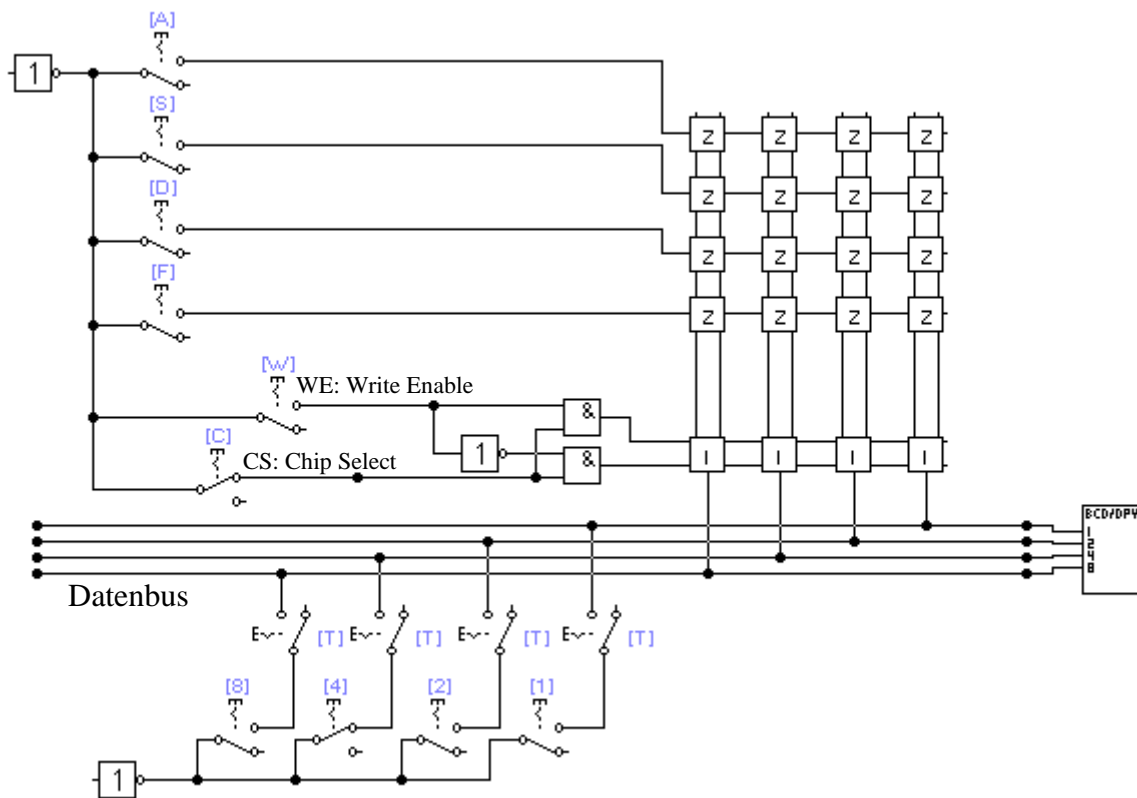


Eine solche Speicherstruktur sollen Sie in dem nächsten Versuch aufbauen.

## Versuch 715


### Speicherschaltungen mit Schreib-Lesebus und Interface zum Datenbus

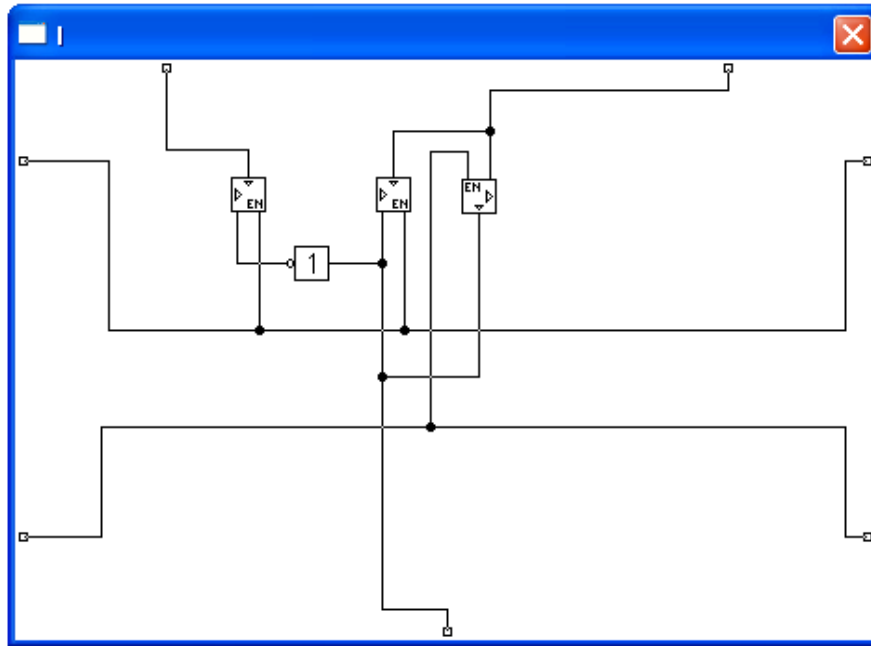
In der Datei v715 finden sie die folgende Speicherschaltung:



Die Schaltung enthält  $m$  Wortzellen mit jeweils  $n$  Bit. Die horizontal verlaufenden Leitungen (A, S, D und F) sind die Adress-Leitungen, sie adressieren jeweils ein  $n$ -Bit-Wort. Jedes Flipflop (diese sollen in den Makros Z verwirklicht werden) ist über die vertikal verlaufenden Leitungen mit den beiden Busleitungen des Schreib-Lese-Busses verbunden. Die Durchschaltelemente bilden in dieser Schaltung sowohl den Multiplexer als auch den Demultiplexer. Mit W werden die Betriebszustände „Schreiben“ und „Lesen“ eingestellt oder abgeschaltet.

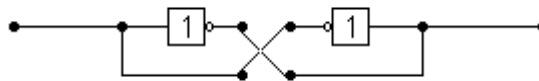
Das Lesen erfolgt über die Leitung B0, nur in umgekehrter Richtung.


Das Makro  enthält die folgende Interfaceschaltung zum Anschluss an den Datenbus:

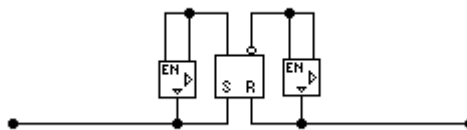


Machen Sie sich anhand der Einbettung in die Gesamtschaltung die Funktionsweise dieses Makros klar.

Bauen Sie in das leere Makro  die 1-Bit-RAM-Speicherzelle des vorigen Versuchs ein. Bauen Sie dabei **an Stelle der Schaltung**



das in der Datei v715 **vorhandene** Makro  ein. Es enthält die folgende Kunstschaltung, die dasselbe leistet wie die obige Schaltung mit den beiden Invertern:



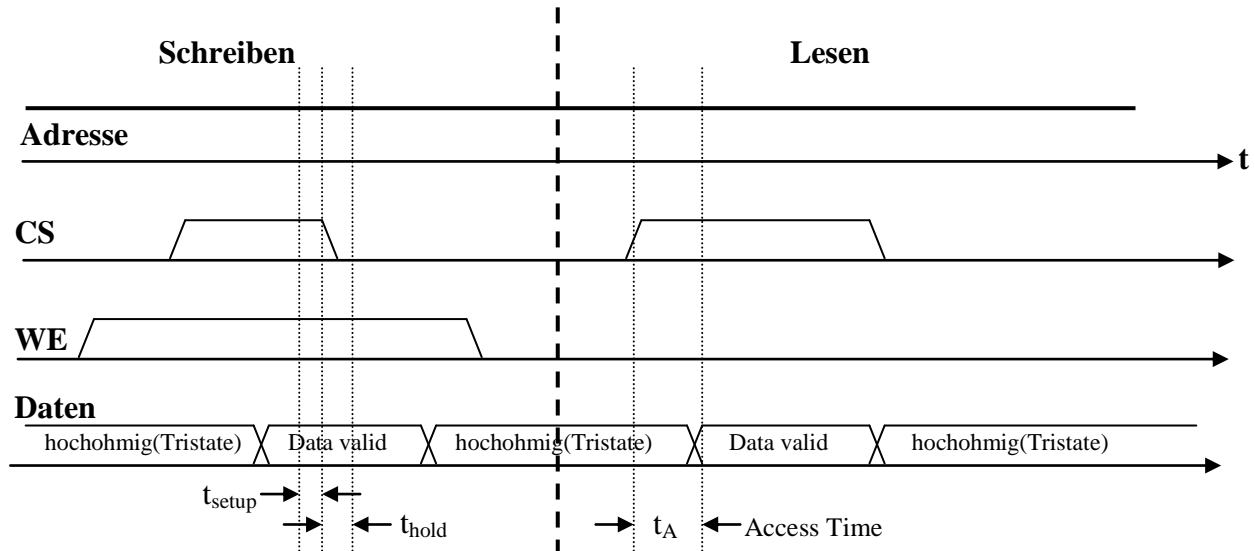
Dieser Umbau ist leider erforderlich, weil die Simulation sonst nicht stabil läuft (Schwäche des EWB-Simulationsalgorithmus).

Demonstrieren Sie die Arbeitsweise ihrer Speicherschaltung in folgenden Schritten:

- 1) Schliessen Sie die Datensender durch Schalten der Tasten T an den Datenbus an. (die Tasten T machen die Datensender zu Tristate-Modulen.)
- 2) Schreiben Sie in die vier 4-Bit-Zellen nacheinander die Zahlen 4, 3, 2 und 1.
- 3) Trennen Sie den Datensender mit der Taste T vom Datenbus.
- 4) Geben Sie mit vier Lesevorgängen die Inhalte der Speicherzellen nacheinander auf den Datenbus aus.

## Theorie 716

Die Wirkungsweise der Schaltung lässt sich anhand des nachfolgenden Zeitdiagramms erklären.



Die Angaben „hochohmig (Tristate)“ in der vierten Zeile des obigen Timings kennzeichnen die Zeiträume, in denen der Datenbus den Zustand Z hat. Der Datenbus hat genau dann den Zustand Z, wenn die Schalter T des Datensenders nicht leiten und die mit dem Bus verbundenen Tristategatter-Ausgänge in den Interfaces I hochohmig sind.

Die Zugriffszeit  $t_A$  (Access Time) des Speichers ist die Zeit, die zum Lesen des Speichers benötigt wird. Jeder Lesevorgang wird mit der steigenden Flanke des CS-Signals gestartet. Nach Ablauf der Zugriffszeit steht der ausgelesene Speicherinhalt auf dem Datenbus zur Verfügung. Der Schreibvorgang wird durch die fallende Flanke des CS-Signals ausgelöst. Je „schneller“ der Speicher, d.h. je kürzer die Zugriffszeit  $t_A$  ist, desto teurer ist der Speicher.

Der zum Zeitpunkt der fallenden Flanke des CS-Signals auf dem Datenbus herrschende Binärzustand wird „geschrieben“. Der Binärzustand des Datenbus muss während der Wirkzeit  $t_{\text{setup}} + t_{\text{hold}}$  konstant sein, um fehlerfrei in den Speicher geschrieben werden zu können.

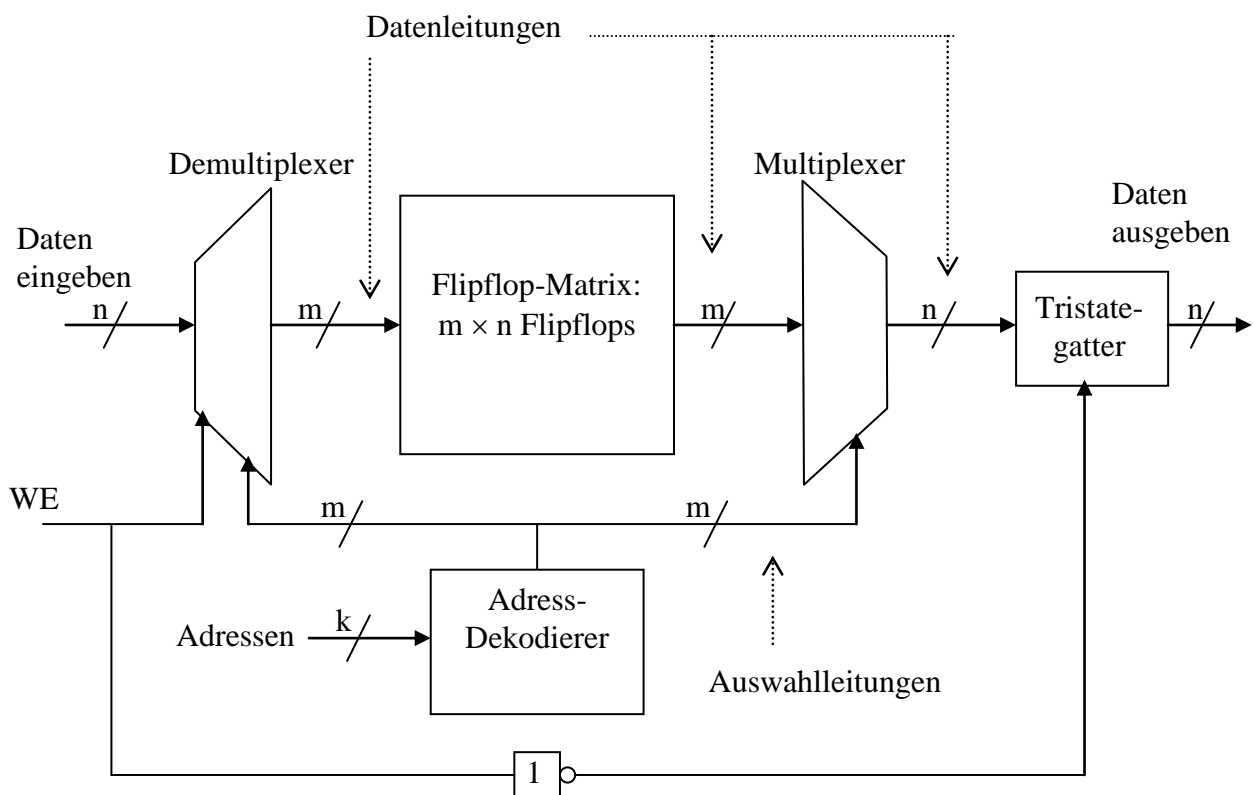


## Versuch 720

### Speicherschaltungen mit Adress-Decodierer

In den bisher vorgestellten Speicherschaltungen wählten wir die Wortzellen mit Tastenschaltern aus. Wir drückten einen von  $m$  Tastenschaltern, wenn die Speicherschaltung  $m$  Wortzellen hatte. Jede Wortzelle hatte also eine  $m$ -stellige Adresse im 1-aus- $m$ -Code. Bei grossen Speichern mit vielen Millionen Speicherzellen hätte die 1-aus- $m$ -kodierte Adresse also viele Millionen Stellen!

Deshalb muss man die Adressen der Speicherzellen als  $k$ -stellige Binärzahl codieren. Die Codewandlung vom 1-aus- $m$ -Code zum Binärcode besorgt ein Adress-Decodierer:

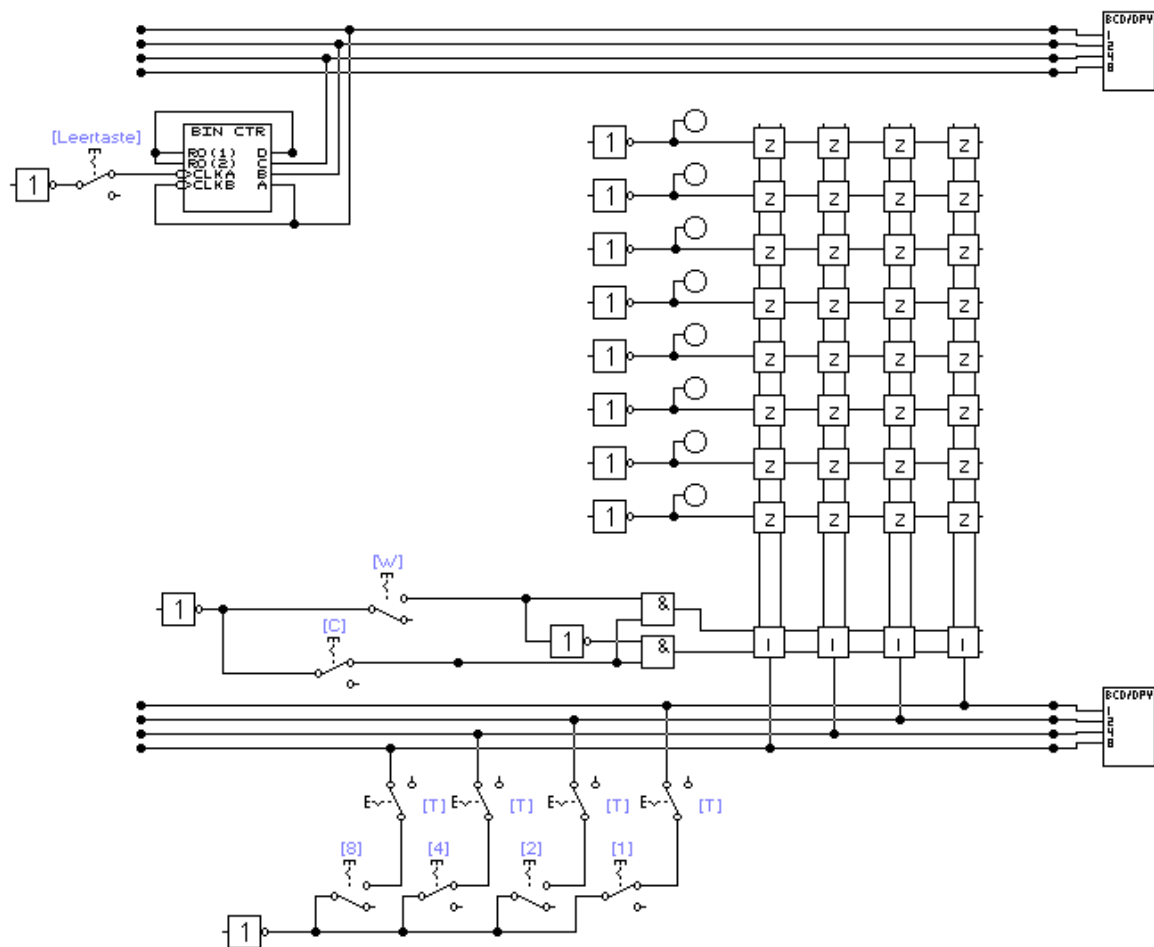


Der Speicher enthält  $m$   $n$ -Bit-Worte und hat  $k$ -stellige Adressen ( $m = 2^k$ ).

Das Steuersignal WE schaltet zwischen den Betriebsarten „Schreiben“ und „Lesen“ um:

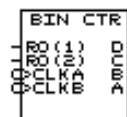
- WE = 1 schaltet den Tristateausgang in den hochohmigen Zustand
- WE = 0 schaltet alle  $m$  Ausgänge des Demultiplexers in den Binärzustand 0.

In der Datei v720 finden sie die folgende Speicherschaltung.



Diese Schaltung ist nach dem gleichen Prinzip wie der vorige Versuch aufgebaut. Sie wurde um die Leitungen eines Adressbusses und eine Zäblerschaltung zum Erzeugen der Adressen ergänzt.

Es befindet sich ausserdem ein EWB-Zähler in der Datei: der „BIN CTR“.

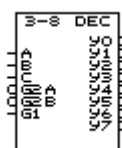


Dieser zählt asynchron und binär modulo 16, wenn der Ausgang A (LSB, least significant bit) des Zählers mit dem Eingang CLKB verbunden wird. In diesem Versuch zählt der Zähler modulo 8.

Machen Sie sich die Funktionsweise dieses Bauelements klar und schauen Sie dafür gegebenenfalls in der EWB-Hilfe nach.

Bauen Sie in die Makros  ihre 1-Bit-Speicherzelle ein und bauen Sie zum Schluss

noch den EWB-Decodierer (Generic 3-to-8 Decoder, G1 auf Vcc legen)



als Adress-Decodierer in die Schaltung ein.

## Theorie 725

### Speicherorganisation

Die adressierbare Einheit der bisher behandelten Speicherschaltungen war die Speicherzelle. Wir ordneten die speichernden Fliflops als Matrizen an, d.h. in rechteckigen Strukturen mit m Zeilen und n Spalten. Jede Zeile bildete mit ihren n Flipflops eine n-Bit-Speicherzelle, die mit einer bestimmten Adresse erreichbar ist. Unsere Versuchsschaltungen hatten 4-Bit-Worte. Die Speicherschaltungen zum letzten Versuch hatte die Speicherorganisation

$$4 \times 4 \text{ Bit}$$

Wir werden die Organisation eines Speichers auch künftig in der Form

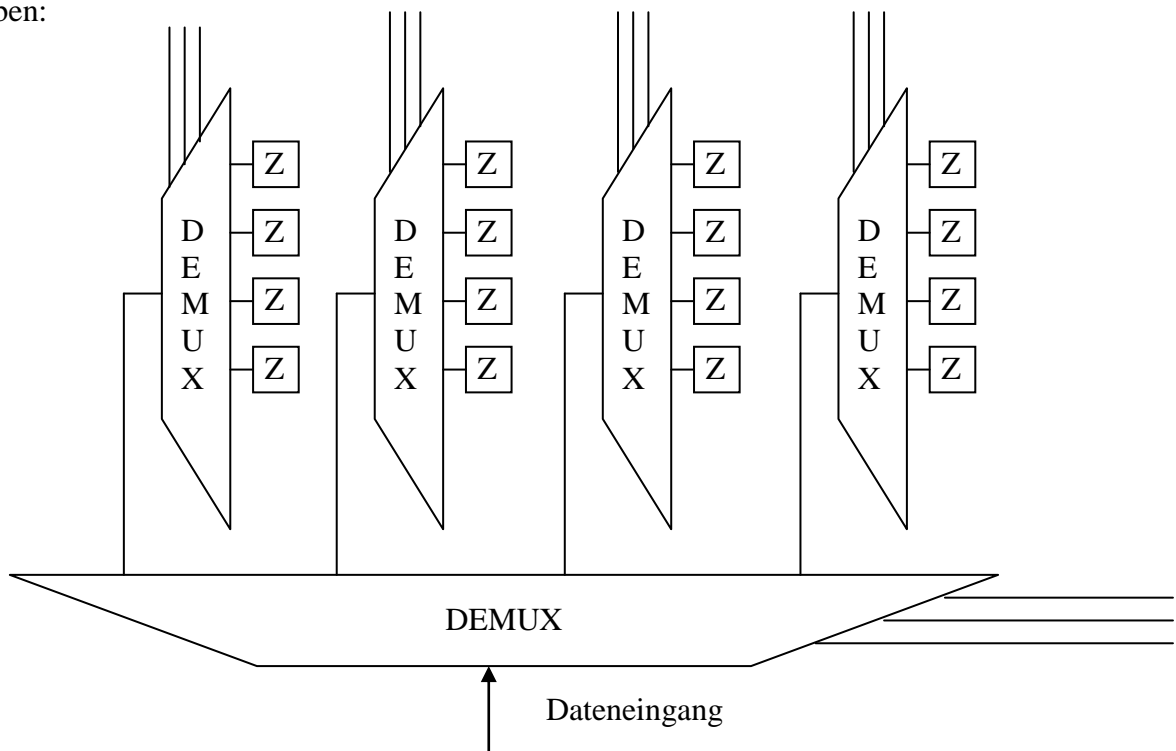
$$\text{Anzahl der adressierbaren Speichertzellen} \times \text{Breite der Speicherzellen}$$

angeben. Dabei geben wir die „Breite der Wortzellen“ entweder in Bit (d.h. als Stellenzahl, Anzahl der Flipflops) oder in Byte an. Wollte man jeden Speicher mit der Organisation

$$m \times n \text{ Bit}$$

wie in unseren bisherigen Schaltungen als  $m \times n$  – Matrix aufbauen, käme man z.B. bei einem  $1024 \times 4$  Bit – Speicher zu einer sehr „länglichen“ Matrix. Technologische Probleme der Miniaturisierung der Schaltung auf Siliziumsubstraten lassen sich eher mit quadratischen Strukturen bewältigen. Speicherchips mit sehr großer Speicherkapazität haben meistens 1-Bit-Speicherworte, d.h. jede 1-Bit-Speicherzelle ist adressierbar. Das kann man mit kaskadierten Multiplexern oder Demultiplexern erreichen:

Schreiben:



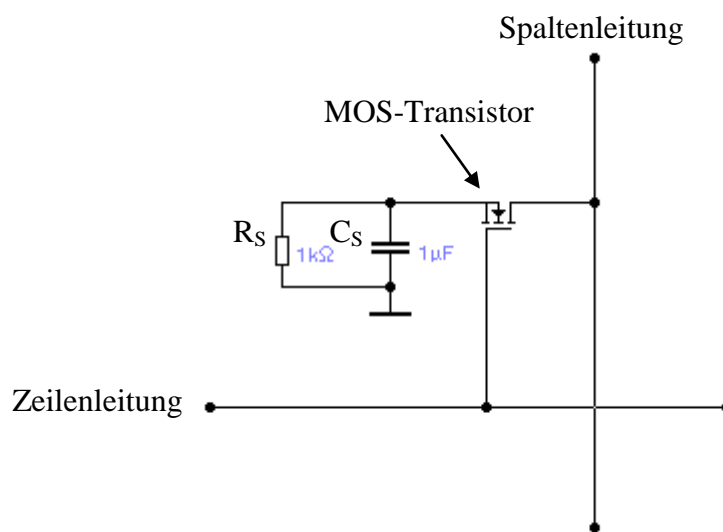
Für das Lesen wird die Speicherstruktur analog mit Multiplexern aufgebaut.

## Versuch 730 & 731

### Aufbau dynamischer Speicher mit Ein-Transistor-Speicherzellen

Eine statische Speicherzelle enthält sechs bis acht Transistoren, eine dynamische Speicherzelle nur einen. Dynamische Speicherzellen (DRAM-Zelle) beanspruchen also weniger Fläche als statische. Dynamische und statische Speicher unterscheiden sich durch ihr Speicherprinzip: Jede statische 1-Bit-Speicherzelle speichert einen von zwei möglichen Schaltzuständen so, wie z.B. ein mechanischer Kippschalter den Schaltzustand „ein“ oder den Schaltzustand „aus“ speichert. Dynamische Speicherzellen speichern elektrische Ladung.

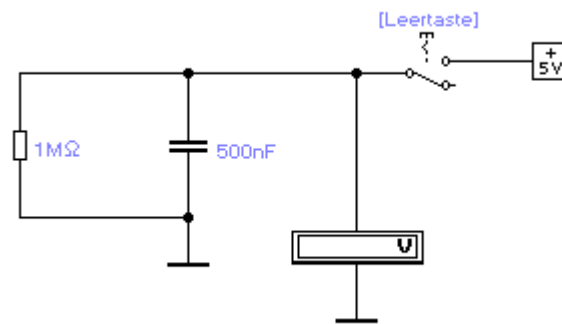
Dynamische Ein-Transistor-Speicherzellen für ein Bit Speicherkapazität können nach folgender Schaltungs-idee mit einem MOS-Transistor aufgebaut werden:



Tatsächlich besteht eine Speicherzelle dieser Art nur aus einem einzigen MOS-Transistor. Der Kondensator  $C_S$  und der Widerstand  $R_S$  verbergen sich in diesem Transistor. Wir stellen den Transistor im Folgenden als gesteuerten Schalter dar:

Wenn der Transistor leitet, wird der Kondensator aufgeladen, weil die Gleichspannungsquelle elektrische Ladung in den Kondensator drückt. Dabei steigt das Potential am Kondensator. Wenn der Transistor sperrt, bleibt der Kondensator aufgeladen. Er „speichert“ das beim Aufladen entstandene Potential. Leider verhalten sich realisierbare Kondensatoren nicht so ideal. Jeder Kondensator verliert mehr oder weniger schnell auch dann seine Ladung, wenn man ihn nach dem Aufladevorgang mit grossen Aufwand von seiner Umgebung isoliert, weil er einen internen „Leckwiderstand“ ( $R_S$ ) besitzt.

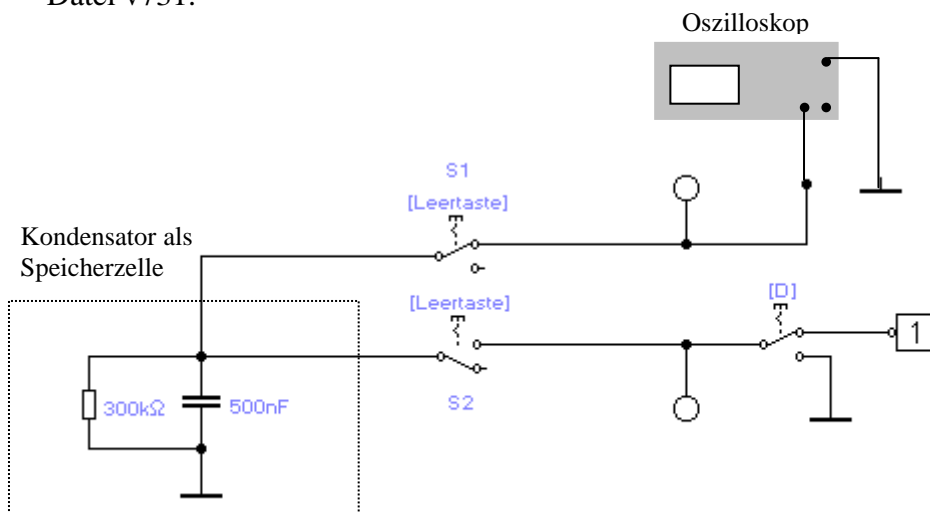
In der Datei v730 finden Sie die folgende Schaltung:



Laden Sie den Kondensator C in der folgenden Schaltung auf, indem Sie den Tastenschalter in den Durchlasszustand schalten.

Schalten Sie den Transistor dann in den Sperrzustand. Was beobachten Sie?

Beobachten Sie nun den zeitlichen Verlauf der Kondensatorentladung in der Schaltung in Datei v731.



Der Kondensator  $C_s$  mit seinem Leckwiderstand  $R_s$  wird in dieser Schaltung als Speicherzelle benutzt, in der ein Bit gespeichert werden kann. Abhängig von der Stellung des Schalters D kann über den Schalter  $S_2$  der Binärzustand 0 oder 1 in die Speicherzelle geschrieben werden. Über den Schalter  $S_1$  kann der in der Zelle gespeicherte Binärwert gelesen werden.

Schalten Sie die Simulation ein. Stellen Sie die Schalter  $S_1$ ,  $S_2$  und D so ein, dass der Binärzustand 1 am Lämpchen „data out“ und am Oszilloskop ablesbar ist. Was beobachten Sie?

Skizzieren Sie das beobachtete Oszilloskop:



Schreiben Sie den Binärzustand 0 in die Speicherzelle. Lesen Sie den in der Zelle gespeicherten Binärzustand 0 am Lämpchen „data out“ ab.

Welcher der beiden Binärzustände ist „flüchtig“ (volatile)?

## Refreshing

Dieser Entladevorgang findet immer statt, d.h. auch dann, wenn die Zelle sich im Ruhezustand befindet. In den vergangenen Versuchen benutzten wir Kondensatoren mit Kapazitäten in der Größenordnung  $\mu\text{F}$  ( $1\mu\text{F} = 10^{-6}$  Farad). Leitungskapazitäten und Speicherzellenkapazitäten auf Speicherchips haben die Größenordnung fF ( $1\text{ fF} = 10^{-15}$  Farad). Die winzige Kapazität einer dynamischen Speicherzelle auf einem Speicherchip entlädt sich in etwa 2 Millisekunden. Deshalb muss jede Zelle, in der eine Ladung dauerhaft gespeichert werden soll, nach Ablauf einer Refreshzeit  $t_R$ , die so kurz ist, dass das Potential nicht unter den für den Binärzustand 1 vorgeschriebenen Pegel sinkt, nachgeladen werden. Eine Refreshschaltung löst in dabei in hinreichend kurzen Zeitabständen für jede Zelle des Speichers einen Lesevorgang aus. Die Refresh-Schaltung adressiert dabei alle Speicherzellen z.B. in der Reihenfolge aufsteigender Adressen. Wenn sie die letzte Zelle erreicht hat, beginnt sie wieder mit der ersten (zyklische Arbeitsweise). Damit ist sichergestellt, dass jede Zelle, in der der Binärzustand 1 gespeichert ist, durch einen Lesevorgang wieder voll aufgeladen wird, bevor das Potential der Zelle noch nicht unter den niedrigsten für den Binärzustand 1 zulässigen Wert gesunken ist.

## Versuch 735 Festwertspeicher

**ROM (Read Only Memory)** ist der Oberbegriff für alle Festwertspeicher. Wir unterscheiden hier die folgenden Festwertspeicherarten:

- Masken-ROM
- Sicherungs-PROM
- EPROM (Erasable PROM)
- EEPROM (Electrically Erasable PROM)

**Masken-ROMs** erhalten ihren Speicherinhalt bei der Herstellung des Halbleiterchips durch die Form der Metallisierungsmaske.

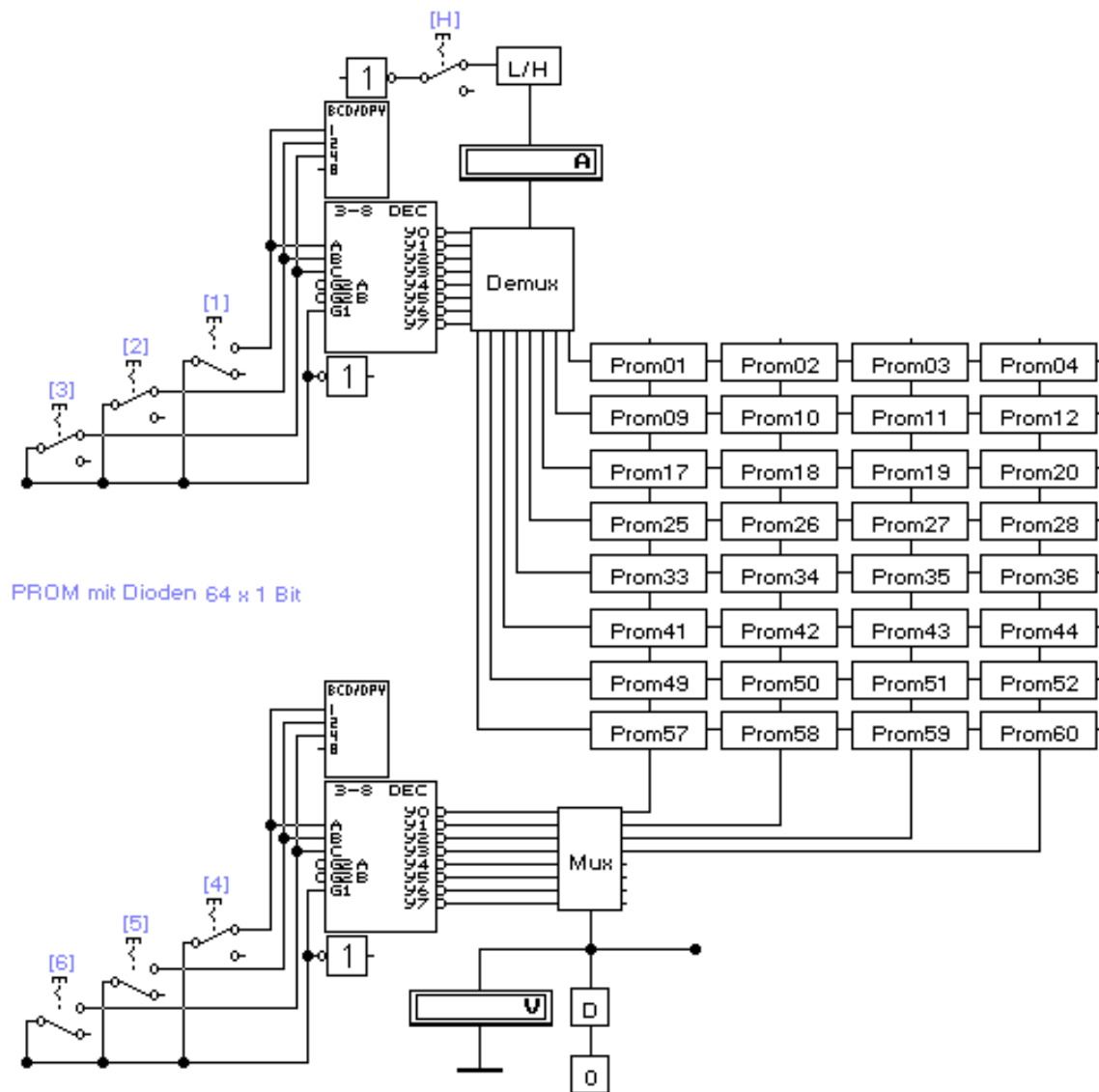
**Sicherungs-PROMs** erhalten ihren Speicherinhalt durch Programmieren beim Anwender. Jede Speicherzelle eines Sicherungs-PROMs enthält eine Sicherung (fuse), die beim Programmieren durch einen hinreichend hohen Strom zum Schmelzen gebracht werden kann. Wir werden im Folgenden ein Sicherungs-PROM untersuchen, bei dem eine Speicherzelle den Binärzustand 0 speichert, wenn die Sicherung durchgebrannt ist. Weil das Durchbrennen der Sicherungen irreversibel ist, können Sicherungs-PROMs nur einmal programmiert werden.

**EPROMs** können mehrfach programmiert werden. Die Programmierung erfolgt beim Anwender. Jede Speicherzelle besteht aus einem speziellen MOS-Feldeffekt-Transistor, der zusätzlich zwischen Gate und Kanal ein sogenanntes Schwebegate besitzt. Mit einer Programmierspannung von 10 bis 25 Volt kann man auf dieses Gate eine Ladung injizieren und damit einen Binärzustand speichern. Der Vorgang ist reversibel: Durch ein Quarzglasfenster im Gehäuse kann man den Speicherchip mit UV-Licht bestrahlen und dabei den gesamten Speicher in etwa 20 Minuten löschen. Je nach Typ enthält der Speicher nach dem Löschen nur noch Nullen oder nur noch Einsen.

**EEPROMs** kann der Anwender wie EPROMs elektrisch programmieren. Darüber hinaus kann sie der Anwender aber auch elektrisch löschen. Dabei ist es in der Regel möglich, sowohl den ganzen Inhalt auf einmal als auch einzelne Worte zu löschen. Allerdings dauert der Lesevorgang etwa 10 bis 15 ms, also sehr viel länger als beim RAM.

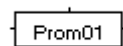
**Flash-EEPROMs** sind elektrisch löschbare Speicher, die im Prinzip wie EEPROMs arbeiten aber in etwa 10 ms, d.h. sehr viel schneller als EEPROMs löscher sind. Es können entweder nur ganze Chips oder Sektoren gelöscht werden, nicht aber einzelne Zellen.

In der Versuchsdatei v735 finden Sie die folgende PROM-Schaltung:

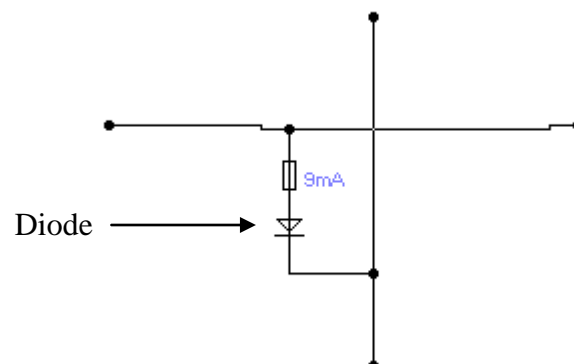


PROM mit Dioden 64 x 1 Bit

Die Bit-Zellen der PROM-Speicherschaltung



sehen wie folgt aus:



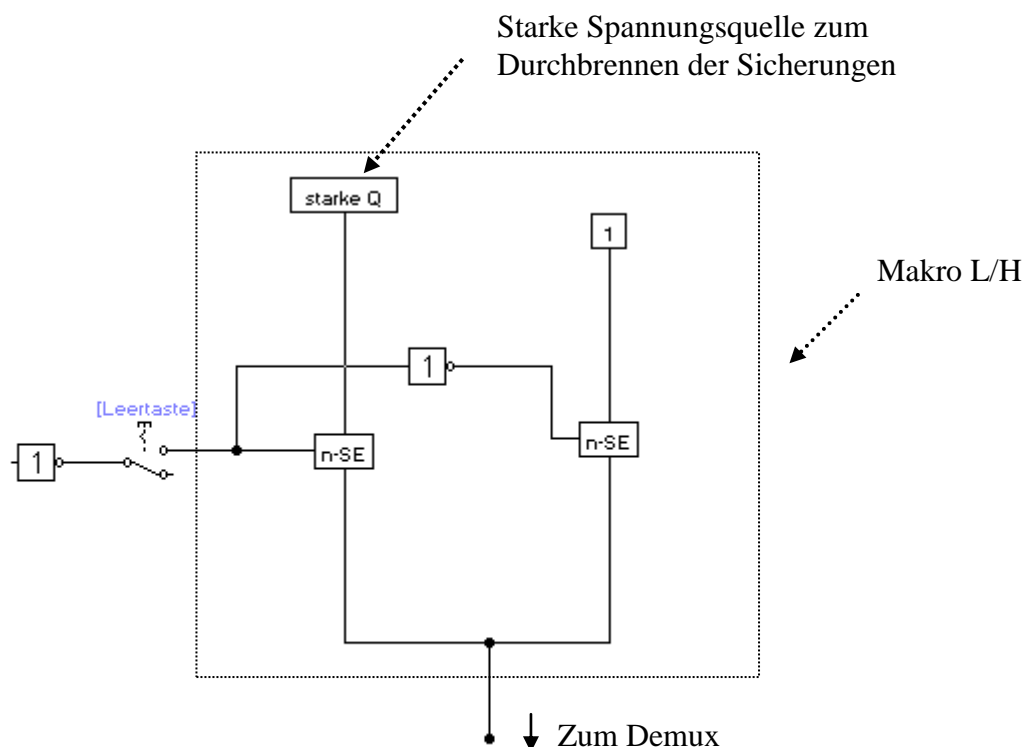


Die Diode lässt das Potential nur in der angegebenen Durchlassrichtung wirken. In der entgegengesetzten Richtung wirkt sie praktisch wie ein nicht leitender (d.h. geöffneter) Schalter. Der Speicher gehört also zur Gattung „Sicherungs-PROM“.

Das Makro



enthält die Stromversorgung. Diese Schaltung hat die folgende abstrahierte Struktur:



Experimentieren Sie mit der Schaltung, bis Sie in der Lage sind, die Arbeitsweise der gesamten Schaltung zu erklären.

Brennen Sie beispielhaft eine Zelle durch. Nehmen Sie nicht die Zelle oben links.