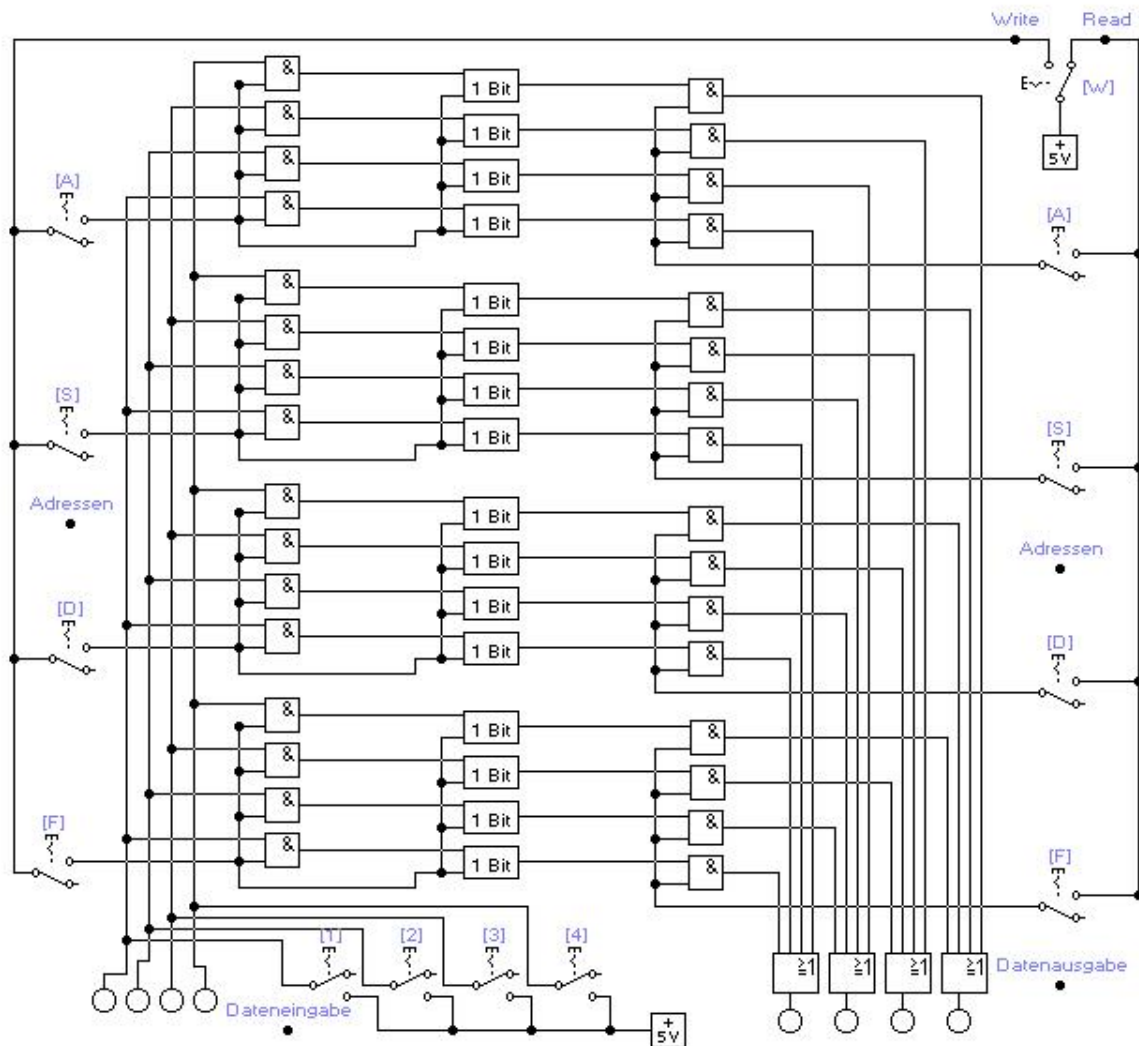
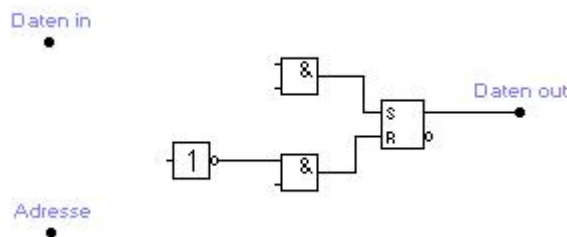




In der Datei v700 finden Sie eine solche Speicherstruktur mit vier Binärworten und vier Adressen A, S, D, F dargestellt:



Bauen Sie folgendes Schaltungsfragment in die 1 Bit Speicherzelle ein und ergänzen Sie die Schaltung:

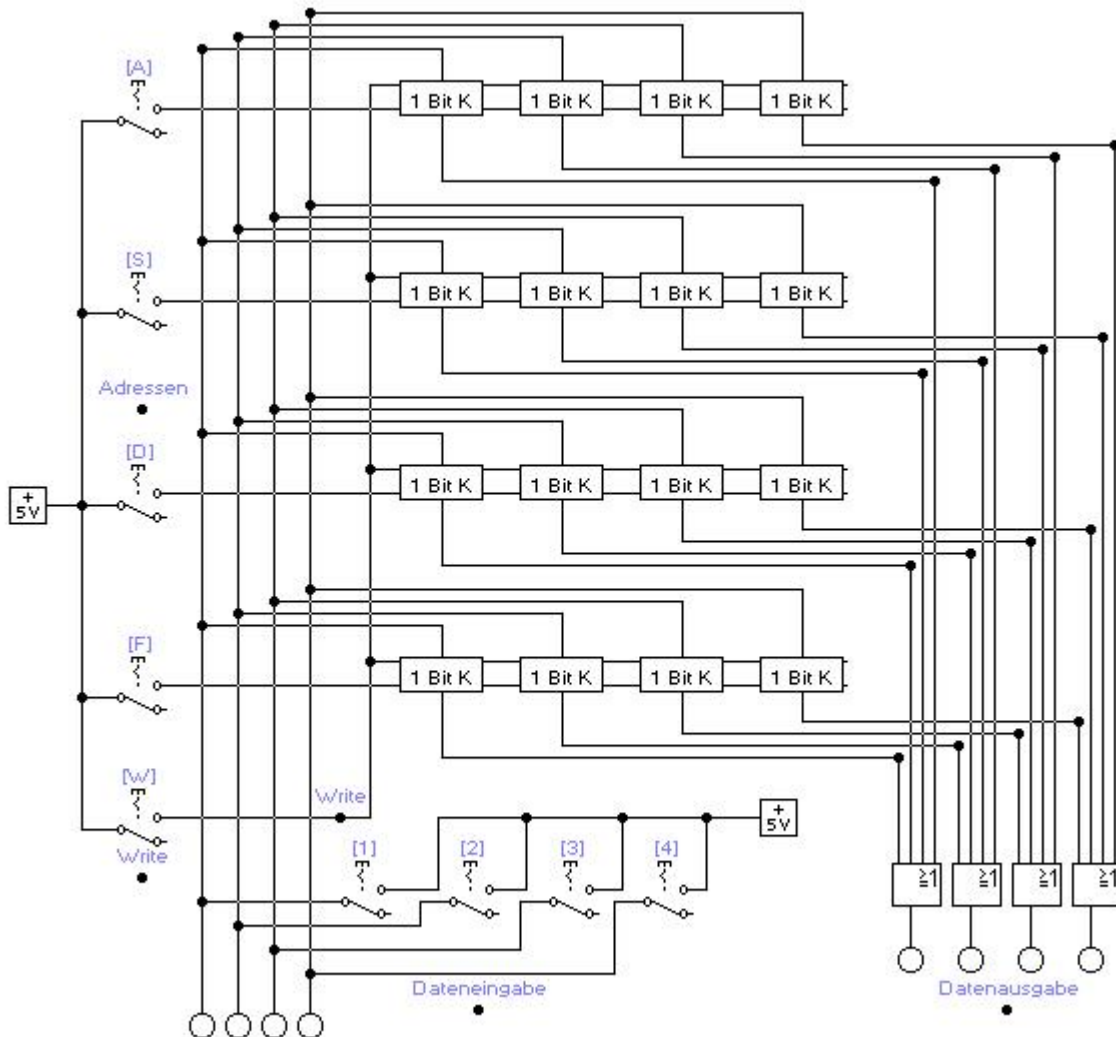


Testen Sie die Schaltung, indem Sie die folgenden vier Binärworte unter den Adressen A, S, D und F speichern und wieder auslesen.

Adresse	Binärwort
A	1000
S	1100
D	1010
F	0011

## Versuch 710 Eine kompaktere Speicherstruktur

Die vorherige Schaltung besteht aus drei getrennten Funktionseinheiten: Demultiplexer, Speichermatrix und Multiplexer. Leider benötigt die Schaltung unnötig viele Bauteile. Man kann den Multiplexer samt der doppelten Adressierung (rechts) einsparen und die Ansteuerung in die „1 Bit“ Speicherzelle integrieren. Ebenso soll es eine Trennung von Adressleitung und Read/Write-Leitung ( $R=0$ ,  $W=1$ ) als Ansteuerung geben. Die Speicherwerte sind nun optisch in einer Matrix (und nicht mehr untereinander) dargestellt.



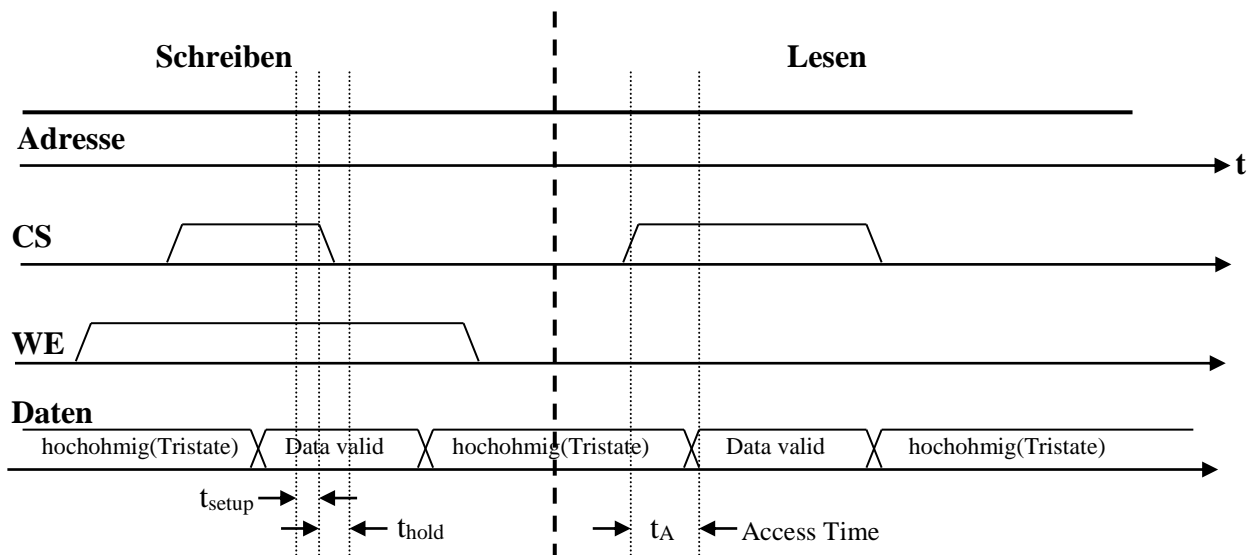
Ergänzen Sie in v710 im Makro „1 Bit K“ eine Variante des entwickelten Makros „1 Bit“, so dass sich eine Speicherschaltung in einer etwas kompakteren Form aufbauen lässt.

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

## Versuch 720 Eine statische 1-Bit-MOS-RAM-Zelle

In der Realität ist es jedoch so, dass die Speicherbausteine in Computern noch etwas kompakter aufgebaut sind. Es gibt einen Adressbus, einen Datenbus sowie eine **WE** (Write-Enable)-Leitung und eine **CS** (Chip-Select)-Leitung die den Baustein ein- und ausschalten kann. Es wird also ein Baustein mit 4 Anschlüssen benötigt (der Stromanschluss ist nicht mitgezählt).

In den Datenblättern von Speicherherstellern findet man folgendes Timingdiagramm:



Die Angaben „hochohmig (Tristate)“ in der vierten Zeile des obigen Timingdiagramms kennzeichnen die Zeiträume, in denen der Datenbus den Zustand Z hat.

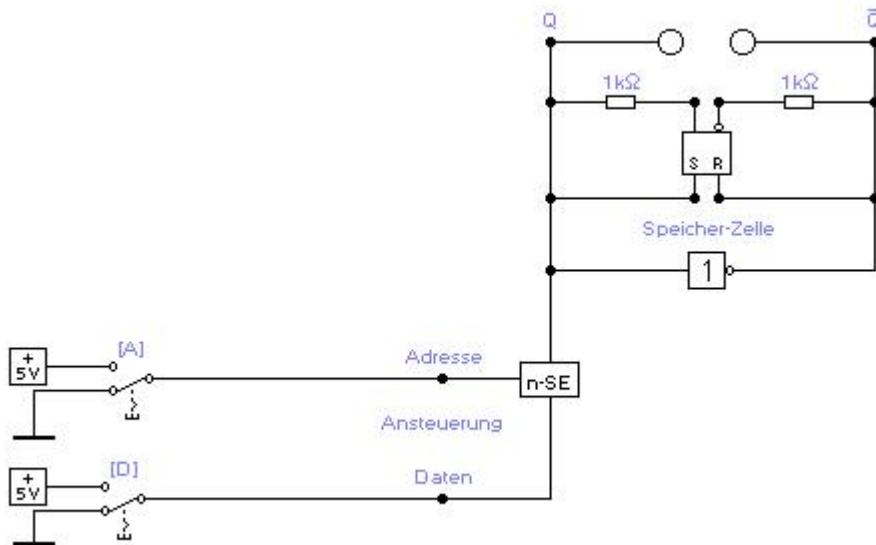
Die Zugriffszeit  $t_A$  (Access Time) des Speichers ist die Zeit, die zum Lesen des Speichers benötigt wird. Jeder Lesevorgang wird mit der steigenden Flanke des **CS**-Signals gestartet. Nach Ablauf der Zugriffszeit steht der ausgelesene Speicherinhalt auf dem Datenbus zur Verfügung. Der Schreibvorgang 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.

Ein RS-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).

Bei unserem Speicheraufbau benötigen wir also eine statische 1-Bit-Speicherzelle. Sie soll lesbar und beschreibbar sein. Die Ansteuerung erfolgt mit einem n-se-Schaltelement, wie es in Kapitel 1 eingeführt wurde. Das Speicherelement wird über die Adressleitung „A“ angesprochen ( $A = 1$ ) und kann mit Daten über „D“ beschrieben werden.

Dazu dient die Schaltung in v720:

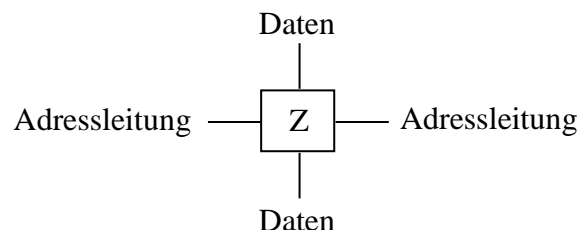


### Aufgabe:

Ergünden Sie die Funktionsweise der Speicher-Zelle. Das gespeicherte Bit kann an Q ausgelesen werden.

Schaltet man 8 dieser Speicherzellen parallel an einen Datenbus, so kann man ein 8-Bit-Wort ansprechen. Um verschiedene Speicherwörter ansprechen zu können, muss für jedes Wort eine Adressierung (eine Adressleitung) erfolgen.

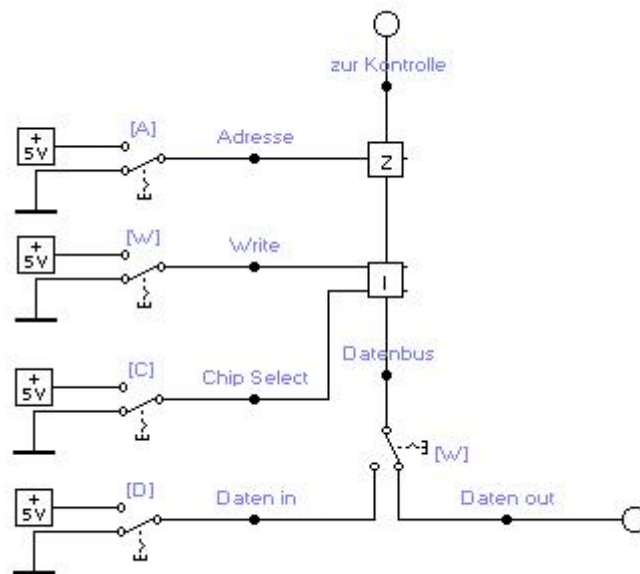
Die 1 Bit – Speicher-Zelle (incl. n-SE Element) wird in einer Box „Z“ (Speicher-Zelle) zusammengefasst:



## Versuch 730 Interface zur Ansteuerung von 1-Bit-Speicherzellen

Um die Speicherzelle „Z“ zu einem Speicherwort zu verbinden, das dann in eine Speichermatrix eingebunden werden soll, wird ein Interface „I“ benötigt, das die Anschlüsse ChipSelect (CS=1) und Write (W=1) enthält. Wenn W=0 ist, werden die Daten von der Speicherzelle „Z“ zum Datenbus gelesen. Bei W=1 werden die Daten vom Datenbus zur Speicherzelle „Z“ geschrieben. Um lesen (W=0) und schreiben (W=1) zu können, muss CS=1 sein. Bei CS= 0 ist die Speicherzelle vom Datenbus abgetrennt (und speichert ihren Inhalt).

In Versuch v730 ist eine Testschaltung zum Ansteuern einer 1-Bit-Speicherzelle gegeben. Sie finden im Interface „I“ n-se-Schaltelemente und Tristategatter, die in Kapitel 1 vorgestellt wurden.



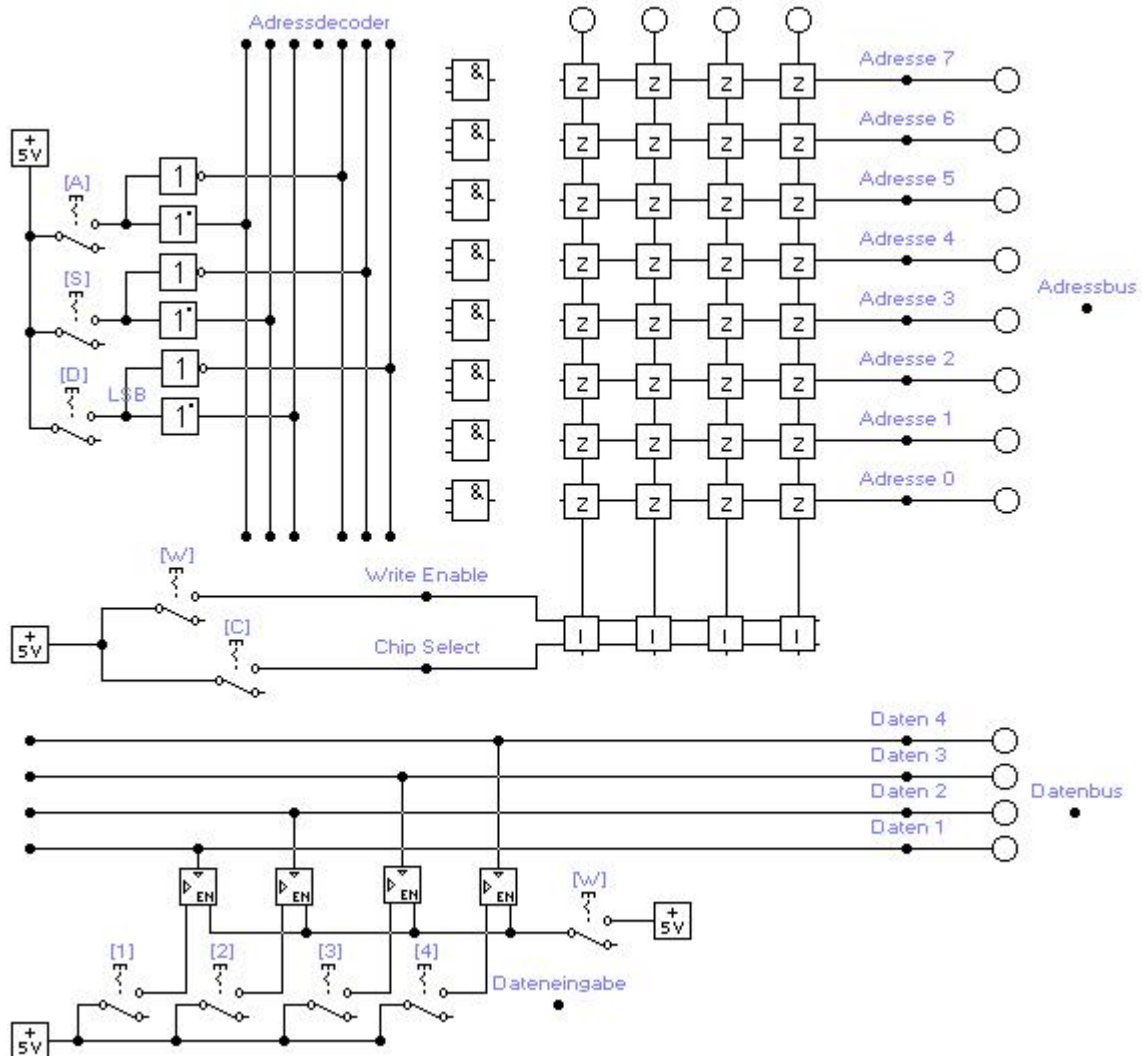
### Aufgabe:

Untersuchen Sie in v730 die Schaltung und erklären Sie die Wirkungsweise von „I“.

## Versuch 740 Speicherschaltung mit Adressdecoder und Datenbus

Nachdem nun alle Steuerelemente erarbeitet sind, lässt sich eine optimale Speicherschaltung in einer  $m \times n$  Flipflop-Matrix aufbauen.

In der Datei v740 finden sie die folgende Speicherschaltung.



Vervollständigen Sie in v740 das Interface „I“ mit den Schaltungen, die Sie in den vorigen Versuchen erarbeitet haben.

Vervollständigen Sie den Adressdecoder zu einem 3 zu 8 Adressdecoder. „D“ sei das LSB. Schließen Sie noch die Datenleitungen an den Datenbus an. „4“ sei das LSB.

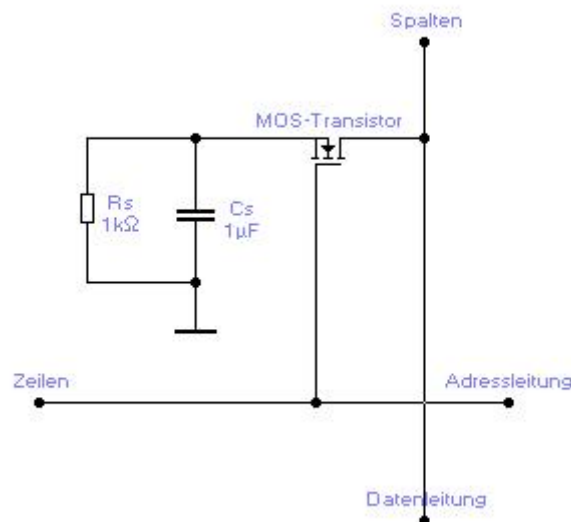
Demonstrieren Sie die Arbeitsweise ihrer Speicherschaltung, indem Sie verschiedene Adressen (Schalter A,S,D) anwählen und dort verschiedene Daten (Schalter 1,2,3,4) einschreiben und anschließend wieder auslesen.

## Versuch 750 Aufbau dynamischer Speicherzellen

In diesem Abschnitt wird ein anderer Speicheraufbau untersucht. Es ist ein dynamischer Speicher DRAM (dynamic random access memory). Seine Zellen verlieren auch bei konstanter Versorgungsspannung ihren Speicherinhalt, weil in ihnen das Spannungspotential, das den gespeicherten Binärzustand 1 repräsentieren soll, allmählich sinkt. Wenn in dynamischen Speicherzellen der Binärzustand 1 (bei konstanter Versorgungsspannung) dauerhaft gespeichert werden soll, muss daher das Potential regelmäßig „aufgefrischt“ werden, damit es nicht unter den Schwellwert für den Binärzustand 1 sinkt (Refresh).

Eine statische Speicherzelle enthält ein Flip-Flop, eine dynamische Speicherzelle genau einen Transistor. 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 Schaltungsidee mit einem MOS-Transistor aufgebaut werden:



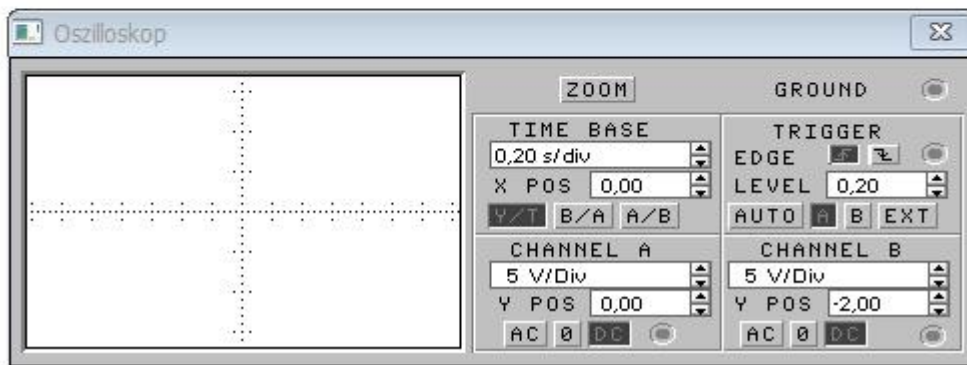
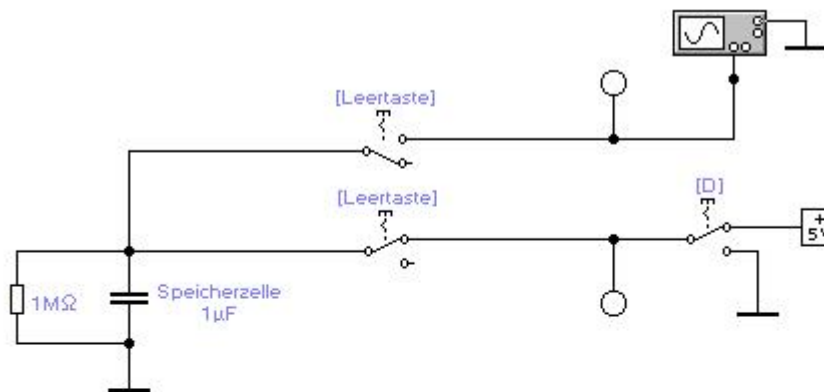
Der Aufbau einer einzelnen DRAM-Speicherzelle ist sehr einfach: Sie besteht nur aus einem MOS-Transistor und einem Kondensator  $C_S$ . Die Information (0 oder 1) wird als elektrische Ladung im Kondensator gespeichert. Jede Speicherzelle speichert ein Bit. Der Widerstand  $R_S$  ist kein echter Widerstand sondern ein Leckwiderstand, der konstruktionsbedingt vorliegt.

Wir stellen den Transistor im Folgenden als gesteuerten Schalter dar:

Wenn der Transistor leitet (Adressleitung = 1), wird der Kondensator aufgeladen, weil die Gleichspannungsquelle (Datenleitung = 1) elektrische Ladung in den Kondensator fließen lässt. Dabei steigt das Potential am Kondensator. Wenn der Transistor sperrt (Adressleitung = 0), bleibt der Kondensator aufgeladen. Er „speichert“ das beim Aufladen entstandene Potential. Leider verhalten sich realisierbare Kondensatoren nicht 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 v750 finden Sie die folgende Schaltung:



Der Kondensator mit seinem Leckwiderstand 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 (Leertaste) der Binärzustand 0 oder 1 in die Speicherzelle geschrieben werden. Am Oszilloskop kann der jeweilige Ladezustand nach dem Aufladen des Kondensators beobachtet werden.

Schalten Sie die Simulation ein. Stellen Sie die Leertaste und D so ein, dass der Binärzustand 1 am Oszilloskop ablesbar ist. Es kann sein, dass man die Leertaste mehrfach betätigen muss, damit das Oszilloskop reagiert.

Beobachten Sie den zeitlichen Verlauf der Kondensatorentladung und skizzieren Sie das beobachtete Oszillogramm:



Welcher der beiden Binärzustände (0 oder 1) ist instabil und muss refresht werden?

## 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  $0,1\text{ pF}$  ( $1\text{ pF} = 10^{-12}$  Farad). Die winzige Kapazität einer dynamischen Speicherzelle auf einem Speicherchip entlädt sich durch den Leckwiderstand. Deshalb muss jede Zelle, in der eine Ladung dauerhaft gespeichert werden soll, nach Ablauf einer Refreshzeit von ca. 32-64 Millisekunden nachgeladen werden. Eine Refreshschaltung löst 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 760 Programmierbare Speicherbausteine

**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)
- Flash-EEPROM

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

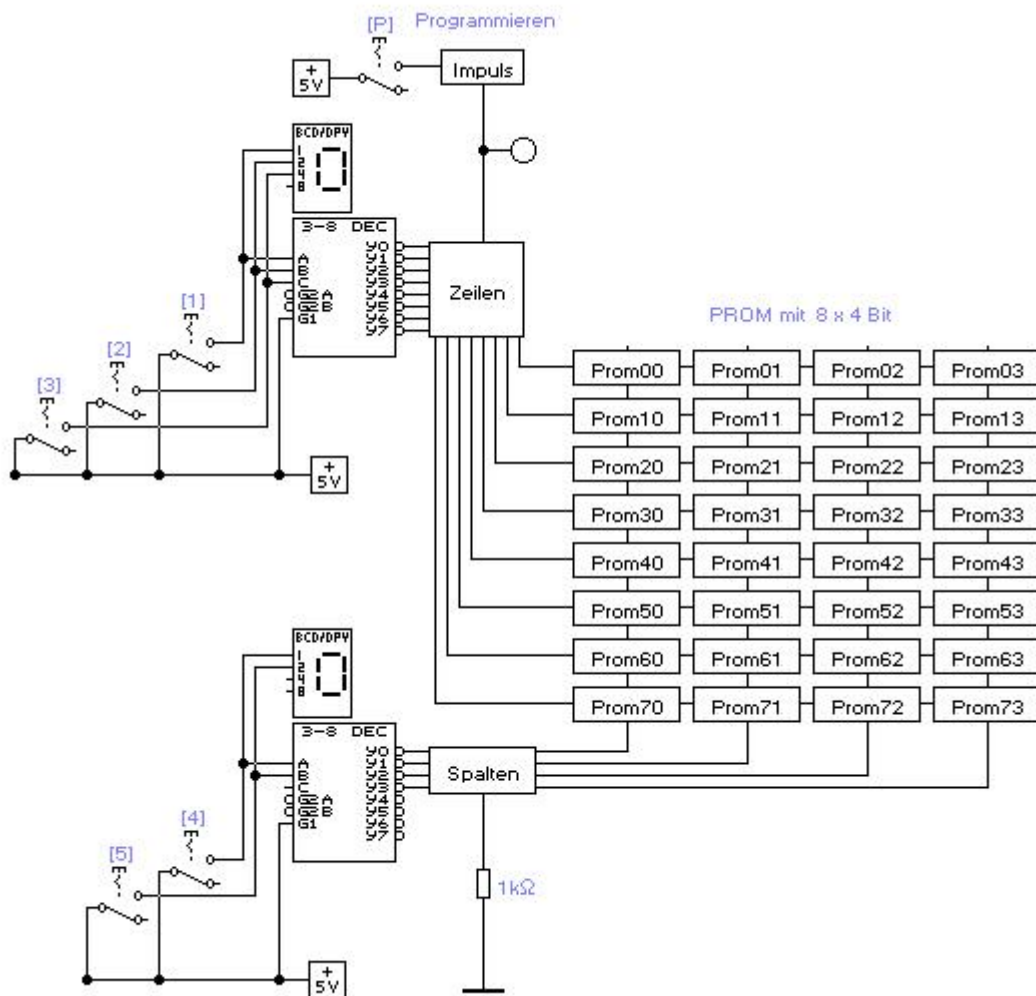
**Sicherungs-PROMs** erhalten ihren Speicherinhalt durch Programmieren beim Anwender. Jede Speicherzelle eines Sicherungs-PROMs enthält eine Sicherung, die beim Programmieren durch einen hinreichend hohen Strom durchgebrannt werden kann. Bei diesem Versuch wird ein Sicherungs-PROM untersucht, bei dem 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.

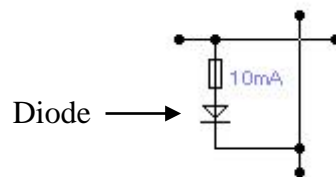
**EEPROMs** kann der Anwender wie EPROMs elektrisch programmieren. Darüber hinaus kann sie der Anwender aber auch elektrisch löschen. Dabei ist es in der Regel möglich, sowohl den ganzen Inhalt auf einmal als auch einzelne Worte zu löschen.

**Flash-EEPROMs** sind elektrisch löschbare Speicher, die im Prinzip wie EEPROMs arbeiten. Es können entweder nur ganze Chips oder Sektoren gelöscht werden, nicht aber einzelne Zellen.

In der Versuchsdatei v760 finden Sie die folgende PROM-Schaltung:



Die Bit-Zellen der PROM-Speicherschaltung z.B. Prom00 sehen wie folgt aus:



Die Diode lässt den Programmierimpuls nur in der angegebenen Durchlassrichtung wirken. In der entgegengesetzten Richtung sperrt sie den Stromfluss. Der Speicher gehört also zur Gattung „Sicherungs-PROM“.

Das Makro „Impuls“ enthält einen Impulsgenerator, der die Sicherung mittels eines Programmierimpulses dauerhaft durchbrennt. Dazu muss der Schalter P eingeschaltet werden.

**Aufgabe:**

Wählen Sie eine Zeile und eine Spalte über den 3-zu-8 Decoder aus. Öffnen Sie das zugehörige Makro PromXY und brennen Sie die Zelle (Sicherung) durch. Sie können das Durchbrennen im geöffneten Makro PromXY beobachten.