4.4 PC-Hardware/Hauptspeicher

4.4.1 Räumliche Organisation von Speicher

Für den Anwender erscheinen RAM und ROM wie ein sehr langes Band an Information, auf dem zu jeder Position ein Inhalt gespeichert ist. Aber wie ist RAM und ROM in Wirklichkeit räumlich organisiert?

Dazu kann man sich ein 1-MiBit-RAM vorstellen, bei dem jede Speicherzelle eine Fläche von 1 µm mal 1 µm belegt. Baut man den Speicher jetzt linear auf (Abbildung 1 links), so ist der Speicher nur 1 µm breit, dafür aber 1 048 576 µm lang, also etwa 1,05 m. Das passt leider in kein tragbares Gerät.

Aus diesem Grund sind nahezu alle RAMs und ROMs anders aufgebaut. Man bricht die Stange in kleinere Stücke, und zwar so lange, bis man ein Quadrat erhält. Dieses Quadrat (auch Matrix genannt, siehe Abbildung 1 rechts) hat jetzt $1024 = \sqrt{1048576}$ Zeilen und Spalten. Damit ist seine Größe nur noch 1,024 mm x 1,024 mm. Durch Anordnung der Speicherzellen in drei Dimensionen

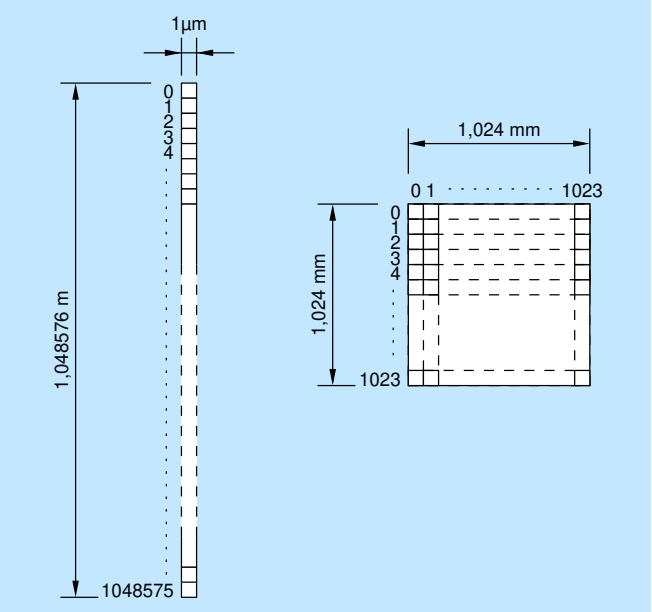

Abbildung 1: Anordnung der Speicherzellen

könnte man die Datenmenge in einem RAM oder ROM noch einmal gewaltig erhöhen. Man stößt dabei aber auf das Problem der Wärmeabfuhr. Die Zellen in der Mitte erzeugen Wärme, die nicht nach außen geleitet werden kann.

Nun stellt sich noch die Frage, wie man die Speicherzellen ansteuern (adressieren) kann, wenn sie in einem quadratischen Feld angeordnet sind. Die Antwort liefert Abbildung 2 anhand eines 64-Bit-ROMs. Drei Leitungen aus dem Adressbus führen zum Zeilendecoder (RAS=row adress select). Er wählt je nach dem Pegel dieser drei Leitungen eine von acht Zeilen aus. Nur diese eine Zeile ist aktiv. Drei andere Leitungen aus dem Adressbus führen zum Spaltendecoder, der eine von acht Spalten auswählt. Nur diese Spalte ist aktiv. Am Schnittpunkt der aktiven Zeile mit der aktiven Spalte liegt nun die eine aktive Speicherzelle der ganzen Matrix. Nur sie kann zu diesem Zeitpunkt gelesen (oder geschrieben) werden.

Der gelesenen Wert wird durch den Konzentrator entnommen und ausgegeben. Bei einem Lesevorgang wird aus der ganzen Speichermatrix also nur ein einziges Bit gelesen, bei einem Schreibvorgang nur ein einziges Bit geschrieben.

In einem praktischen System wird bei einem Speicherzugriff nicht nur ein einziges Bit gelesen oder geschrieben, sondern ein ganzes Datenwort (Speicherwort), bestehen aus so vielen Bits, wie

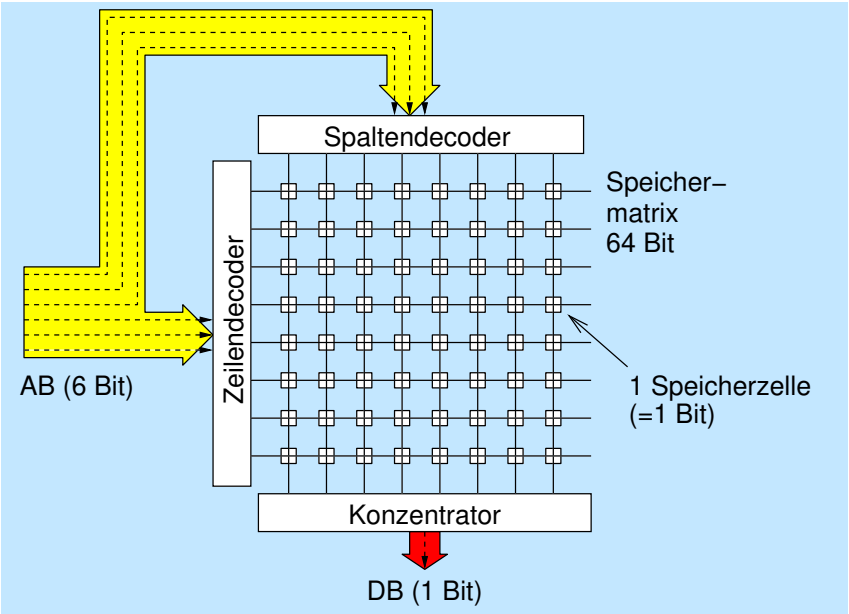

Abbildung 2: Adressierung der Speicherzellen in RAM und ROM

der Datenbus breit ist (Datenbusbreite DBB). Zu diesem Zweck schaltet man genau so viele Quadrate (Matrizen) parallel an den Adressbus. Die ausgegebenen Bits werden dann nebeneinander auf den Datenbus geleitet (siehe Abbildung 3).

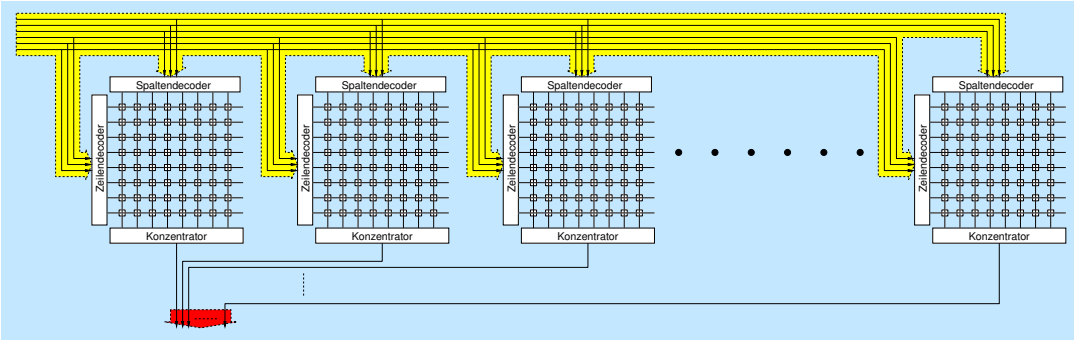

Abbildung 3: Zusammenschaltung der Speichermatrizen in einem n-Bit-System

4.4.2 Realisierung der Speicherzellen (ROM)

Die Realisierung der einzelnen Speicherzellen ist zumindest beim ROM sehr einfach. Abbildung 4 zeigt noch einmal die Anordnung, links in der Aufsicht und rechts in räumlicher Darstellung.

Ein einzelnes Bit könnte nun so verwirklicht werden, dass an der Kreuzung einer Zeilen- und einer Spaltenleitung eine optionale Verbindung hergestellt wird (Abbildung 5). Eine vorhandene Verbindung entspräche einer 1, eine fehlende Verbindung einer 0. Bei einer 1 würde die eingezeichnete Lampe leuchten, bei einer 0 nicht.

Leider hat diese einfache Technik ein Problem: Wie in Abbildung 6 angedeutet, können benachbarte Bits ungewollt dort eine Verbindung (also ein 1-Bit) vortäuschen, wo keine Verbindung vorhanden ist. In der räumlichen Darstellung erkennt man, dass der Stromfluss von der oberen zur unteren Ebene, zurück zur oberen Ebene und schließlich wieder nach unten verläuft.

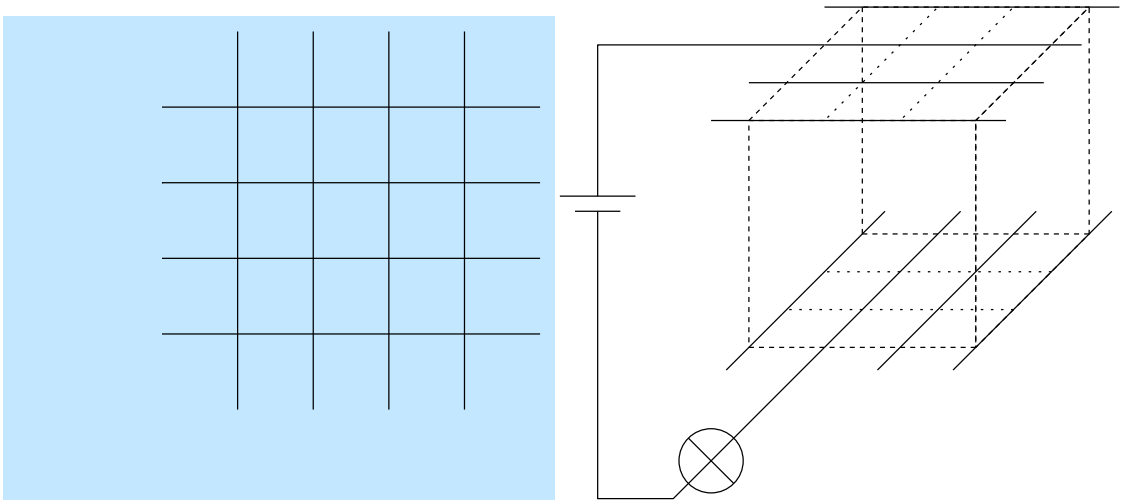

Abbildung 4: Realisierung I

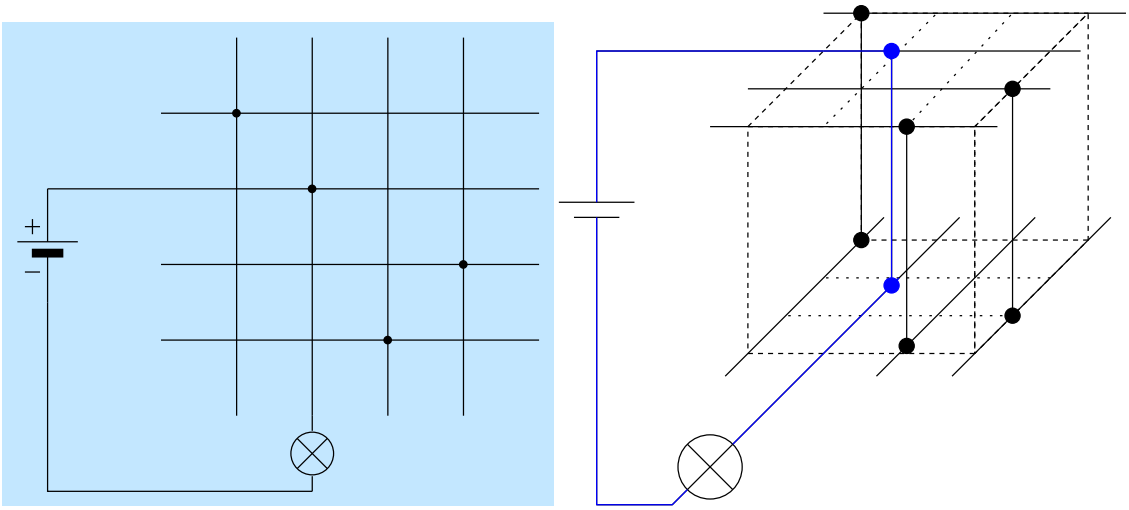

Abbildung 5: Realisierung II

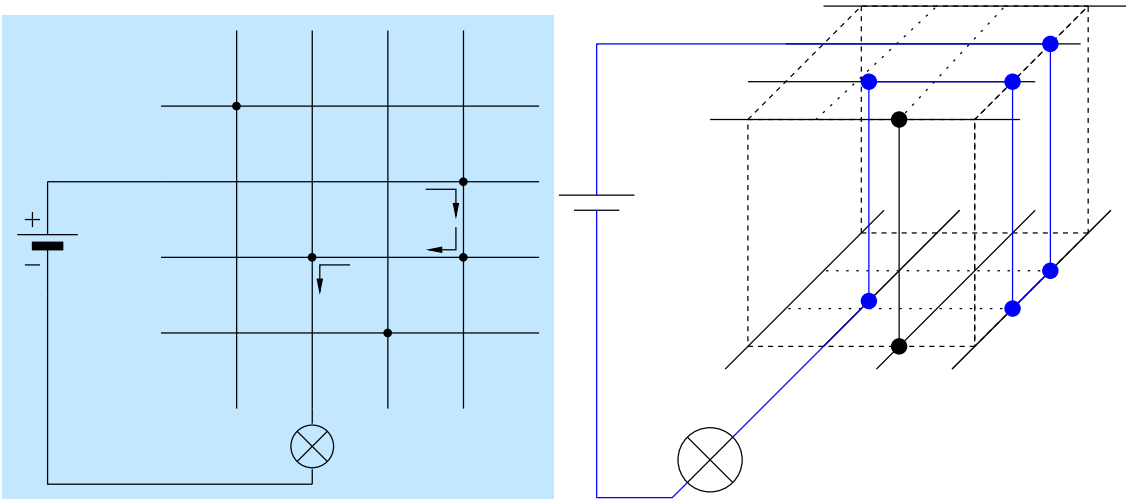

Abbildung 6: Realisierung III

Man löst dieses Problem durch Stromventile, so genannte Dioden. Diese Bauteile lassen den Strom nur in einer Richtung (in Richtung des Pfeils) durch. Ein 1-Bit wird nun anstatt durch eine Verbindung durch eine Diode realisiert. Bei einem 0-Bit fehlt die Diode (Abbildung 7). In der räumlichen Darstellung sieht man, dass der Strom nur noch von der oberen Ebene zur unteren fließen kann, aber von dort nicht mehr zurück nach oben.

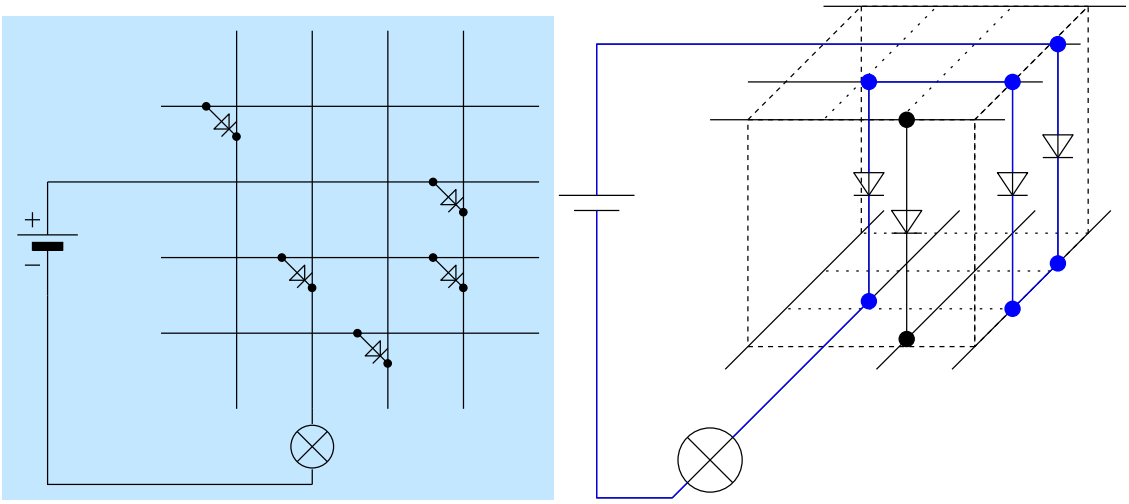

Abbildung 7: Realisierung IV

4.4.3 ROM-Typen

Die einzelnen ROM-Typen unterscheiden sich dadurch, wie nun die 0- und 1-Bits in das ROM hineinkommen.

4.4.3.1 MROM Beim MROM (maskenprogrammiertes ROM) wird das schon bei der Produktion festgelegt. Der Auftraggeber schickt sein Programm (oder sonstigen gewünschten Inhalt) per Netz oder Datenträger zum Halbleiterhersteller. Dort wird eine dem Inhalt entsprechende fotografische Maske erstellt. Sie ist an den Stellen, an denen eine Eins stehen soll, geschwärzt und an den

anderen nicht (oder umgekehrt). An diesen Stellen (mit der Eins) entsteht im Produktionsprozess eine Diode, an den anderen nicht (Abbildung 8). Mit Hilfe der Maske wird nun die gewünschte

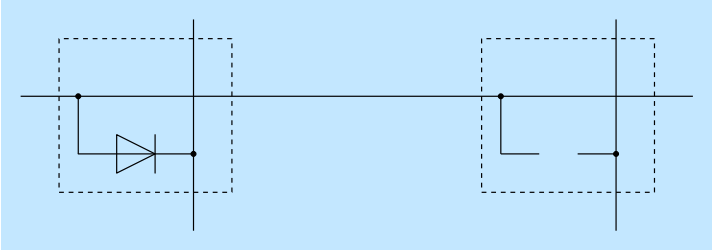

Abbildung 8: Speicherzellen im MROM, links: 1, rechts: 0

Anzahl von ROMs (z.B. 1000) gefertigt und schließlich an den Kunden zurückgeschickt. Dieser ROM-Typ wird bis heute verwendet, wenn hohe Stückzahlen gebraucht werden.

4.4.3.2 PROM Für kleinere Stückzahlen und Einzelfertigung wurde schon früh der Wunsch laut nach einem ROM, das man sich selbst programmieren kann. Dazu wurde das PROM (*programmable ROM*) erfunden, auch OTP-ROM (*one time programmable ROM*) genannt. Beim PROM sind werkseitig nur Einsen einprogrammiert, d.h., an jedem Kreuzungspunkt befindet sich eine Diode. In Reihe zu dieser Diode ist allerdings eine Art Sicherung eingebaut (Abbildung 9). Mit

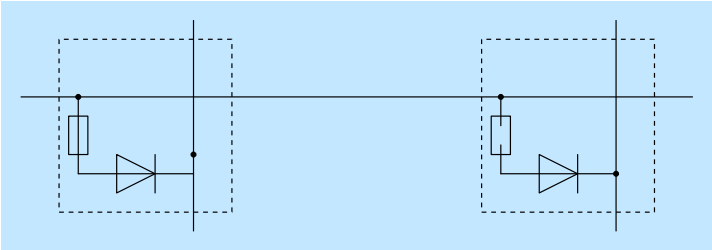

Abbildung 9: Speicherzellen im PROM, links: 1, rechts: 0

einem speziellen Programmiergerät kann man die Sicherung gezielt zerstören und damit an dieser Stelle eine Null einprogrammieren (Programmierung durch Zerstörung). Diesen Vorgang kann man nicht mehr rückgängig machen. Insofern kann man ein PROM also nur einmal benutzen. Dieser ROM-Typ wird gerne für Kleinserien verwendet. Für ein Firmware-Update benötigt man ein neues PROM.

4.4.3.3 EPROM Im Entwicklungslabor braucht man ein ROM, das man mehrfach verwenden kann. Dazu wurde das EPROM entwickelt. Man kann es mit Hilfe harter UV-Strahlung löschen: Das EPROM wird in ein spezielles Löschgerät gelegt und mehrere Minuten bestrahlt. Das EPROM hat auf der Gehäuseoberseite ein Fenster, durch das die Strahlung direkt auf den Chip treffen kann (Abbildung 10). Beim Umgang mit dem Löschgerät muss man vorsichtig sein, da die verwendete Strahlung Haut und Augen dauerhaft schädigen kann. EPROMs benutzen eine andere Technik als PROMs; bei EPROMs ist jede Speicherzelle durch einen Floating-Gate-MOSFET realisiert [1] (Abbildung 11).

4.4.3.4 Flash-EPROM Für Anwender ist die Benutzung von EPROMs zu teuer (Programmier- und Löschgerät nötig) und zu aufwendig (ausstecken, löschen, ins Programmiergerät stecken, Programm einbringen, ausstecken, zurück ins System stecken). Anwender brauchen ein EPROM, das

[1] Heutzutage werden auch PROMs mit dieser Technik verwirklicht, allerdings ohne das Fenster

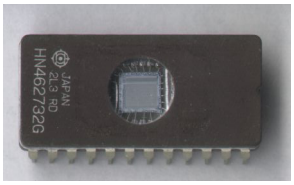

Abbildung 10: EPROM

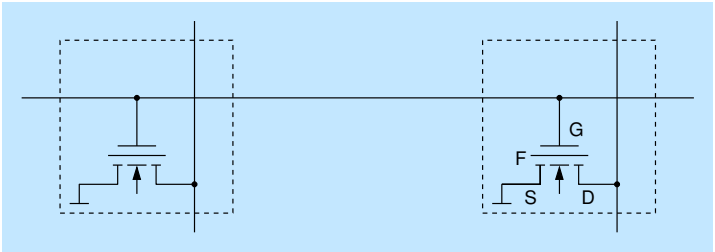

Abbildung 11: Speicherzellen im EPROM, falls Floating Gate (F) negativ aufgeladen: 1; sonst: 0

man im Ursprungssystem löschen kann. Ein solches Bauteil ist das Flash-EPROM (auch Flash-Speicher oder Flash-EEPROM genannt). Man kann es elektrisch löschen (und programmieren), allerdings nicht Wort für Wort, sondern nur komplett oder blockweise (z.B. ein Achtel des Chips auf einmal). Flash-EPROMs eignen sich deshalb für die Realisierung des BIOS-ROMs wie auch für SSDs (*solid state disks*, das sind Laufwerke, die nur aus Chips bestehen und keine mechanischen Teile mehr besitzen).

4.4.3.5 EEPROM Oft braucht man Speicher, der sich so einfach wie ein RAM Byte für Byte löschen und wiederbeschreiben lässt. Dazu gibt es das EEPROM (*electrically erasable PROM*), auch E^2PROM oder EAROM (*electrically alterable ROM*) genannt. Die Technik ist ähnlich der des Flash-EPROMs, jedoch erlauben zusätzliche Verbindungen das separate Löschen jeder Speicherzelle.

4.4.3.6 NVRAM Ebenso, wie ROMs entwickelt wurden, die schon fast wie ein RAM geschrieben und gelesen werden können, gibt es mittlerweile auch RAMs, die ihre Information zumindest eine gewisse Zeit nach dem Abschalten der Betriebsspannung behalten. Sie nennen sich NVRAM oder kurz NVR (*non-volatile RAM*). Eine Technik für NVRAMs funktioniert so, dass ein energiesparendes statisches CMOS-RAM durch einen Akku oder eine Primärzelle gepuffert wird.

4.4.4 RAM-Typen: SRAM und DRAM

Beim RAM gibt es eine etwas größere Vielfalt an Typen. Prinzipiell unterteilt man RAM am besten nach Technik, Speicherdauer und Art der Kommunikation mit dem Bussystem. Bei den Techniken ist zur Zeit nur die Speicherung in MOSFET-Halbleiter-Chips gebräuchlich. Vor langer Zeit (bis Ende der 60-er Jahre) gab es noch den Ringkernspeicher, bei dem jedes Bit in einem einzelnen magnetisierbaren Ring gespeichert war. Bei ihm kann man jedes Bit noch mit dem bloßen Auge zu erkennen. Für die Zukunft ist schon seit Jahren der Magnetblasenspeicher im Gespräch, bei dem die Information in einem magnetisierbaren dreidimensionalen Körper gespeichert wird.

Die RAMs in Halbleitertechnik kann man einteilen in Statische RAMs (*static RAM*, SRAM) und dynamische RAMs (*dynamic RAM*, DRAM).

4.4.4.1 SRAM Statisches RAM funktioniert nach dem Prinzip der Selbsthaltung. Jede Speicherzelle bildet ein sogenanntes Flip-Flop. Man kann sich das so vorstellen wie in Abbildung 12.

Zunächst kann die gestrichelt eingezeichnete Linie weggelassen werden. Es wird ein Stromkreis

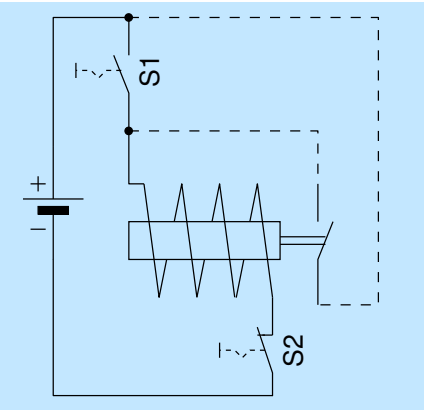

Abbildung 12: Statisches RAM (SRAM), Prinzip

eingeschaltet durch Schalter S1. Im Stromkreis befindet sich ein Elektromagnet, der dadurch mit eingeschaltet wird. Der Elektromagnet wirkt auf einen weiteren Schalterkontakt K1, der ebenfalls geschlossen wird. Diese Anordnung aus Elektromagnet und Schalterkontakt nennt man in der Elektrotechnik Relais oder auch Schütz. Man kann also mit S1 den Kontakt K1 schließen. Lässt man S1 los, dann öffnet K1 auch. Nun soll die gestrichelte Linie mit einbezogen werden. Wenn man jetzt S1 schließt, dann sorgt K1 kurz darauf dafür, dass S1 überbrückt wird. Lässt man ab diesem Zeitpunkt S1 los, bleibt der Stromkreis trotzdem geschlossen. Der Stromkreis hat sich gemerkt (gespeichert), dass S1 gedrückt worden war. Will man dies beenden, hat man zwei Möglichkeiten: Entweder man schaltet die Stromversorgung aus, dann lässt der Elektromagnet los und K1 öffnet (man sagt, das Relais fällt ab). Oder aber man betätigt S2. S2 ist ein so genannter Öffner, d.h., bei Betätigung öffnet er den Stromkreis, und auch hier fällt das Relais ab. Im statischen RAM ist für jedes Bit eine solche Selbsthalteschaltung vorhanden, die allerdings nicht aus Relais besteht (das war bei Zuses Z3 noch so), sondern aus zwei oder mehr Transistoren (oft so genannte MOSFETs) in einem Chip.

4.4.4.2 DRAM Dynamisches RAM dagegen speichert in jeder Speicherzelle eine winzige elektrische Ladung in einer Art Kondensator (Abbildung 13). Ein Kondensator ist in seiner Wirkung

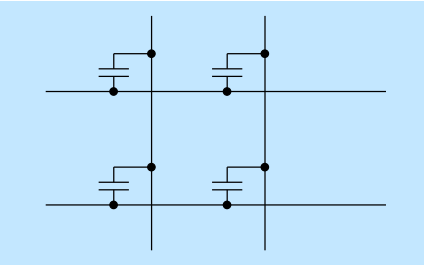

Abbildung 13: Dynamisches RAM (DRAM), sehr stark vereinfacht

ähnlich einem Akkumulator, daher auch das ähnliche Schaltsymbol. Wenn man an ihm eine Spannung anlegt, dann kann er eine gewisse Ladungsmenge speichern. Dann kann man in einem leeren Kondensator eine 0 und im aufgeladenen Kondensator eine 1 speichern. Damit das Aufladen und Entladen des Kondensators schnell geht, muss er eine *sehr* kleine Kapazität haben. Das bedeutet aber auch, dass er sich durch sogenannte Leckströme auch schnell wieder von selbst entladen kann. Die DRAM-Speicherzelle ist also vergesslich. Ihre Speicherzeit liegt im Bereich zwischen

einigen Millisekunden und einigen Sekunden. Nun sind Computer extrem abhängig davon, dass nie (*wirklich nie*) ein Bit umkippt. Anders als ein Lebewesen, das auch mit mehreren zerstörten Nervenzellen noch gut weiterleben kann, stürzt ein Computerprogramm in der Regel ab, sobald ein Bit z.B. in einem Befehlswort falsch ist. Daher braucht man bei einem RAM stets einen Refresh: In regelmäßigem Abstand (z.B. alle 20 ms) wird jedes Speicherwort nacheinander ausgelesen und sofort wieder geschrieben (genauso, wie man als Mensch der Vergesslichkeit abhilft).

Da DRAMs weniger Technik pro Speicherzelle brauchen als SRAMs, sind sie bei gleicher Datenmenge preisgünstiger als diese. Daher wird der Hauptspeicher in PCs durchgehend als DRAM ausgeführt. Im Cache dagegen kommt es auf Geschwindigkeit an; dort wird SRAM verwendet. Ebenso benutzt man SRAM bei Mikrocontroller-Systemen, um den Mikrocontroller von Refresh-Aufgaben zu entlasten und um die Möglichkeit zu haben, das System im Sparmodus auf Taktfrequenz null herunterzutakten.