

## Baugruppen

### Speicherarten (Aufbau und Wirkungsweise):

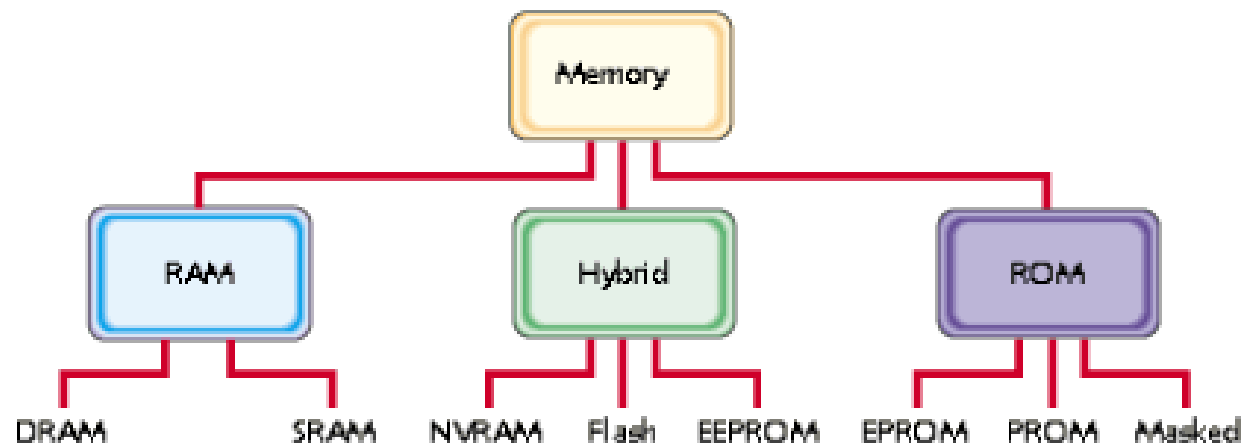


Abb.32: Gängige Speichertypen

- Speicher dient zur Aufbewahrung von Daten
- **Festwertspeicher** halten Daten auf Dauer (auch nach abschalten Versorgungsspannung)
- **Schreib-Lesespeicher** verlieren Daten nach Verlust von Versorgungsspannung

- Speicherfeld besteht aus vielen 1-Bit-Speicherelementen, die z.B. byteweise angeordnet sind

Beispiel: Bei einem Prozessor mit 32 Bit-Architektur kann ein max. Speicher von  $z_s = 2^n = 2^{32} = 4.294.967.296$  Byte = 4.194.304 KiB = 4 GiB adressiert werden.  
 (-> bzw. Speicherzellen angewählt werden)

- Größe von Speichern bei bitweiser Speicherung mit z.B. 1 Kibit =  $2^{10}$  bit = 1024 bit angegeben (oder bei wortweiser Speicherung z.B. 1 MiB =  $1.048.576 \cdot 8$  bit)

-> Schreib-Lese-Speicher hat (hier) 4 Adressleiter ->  $2^4 = 16$  Zeilen anwählbar  
 -> Daten werden byteweise gespeichert

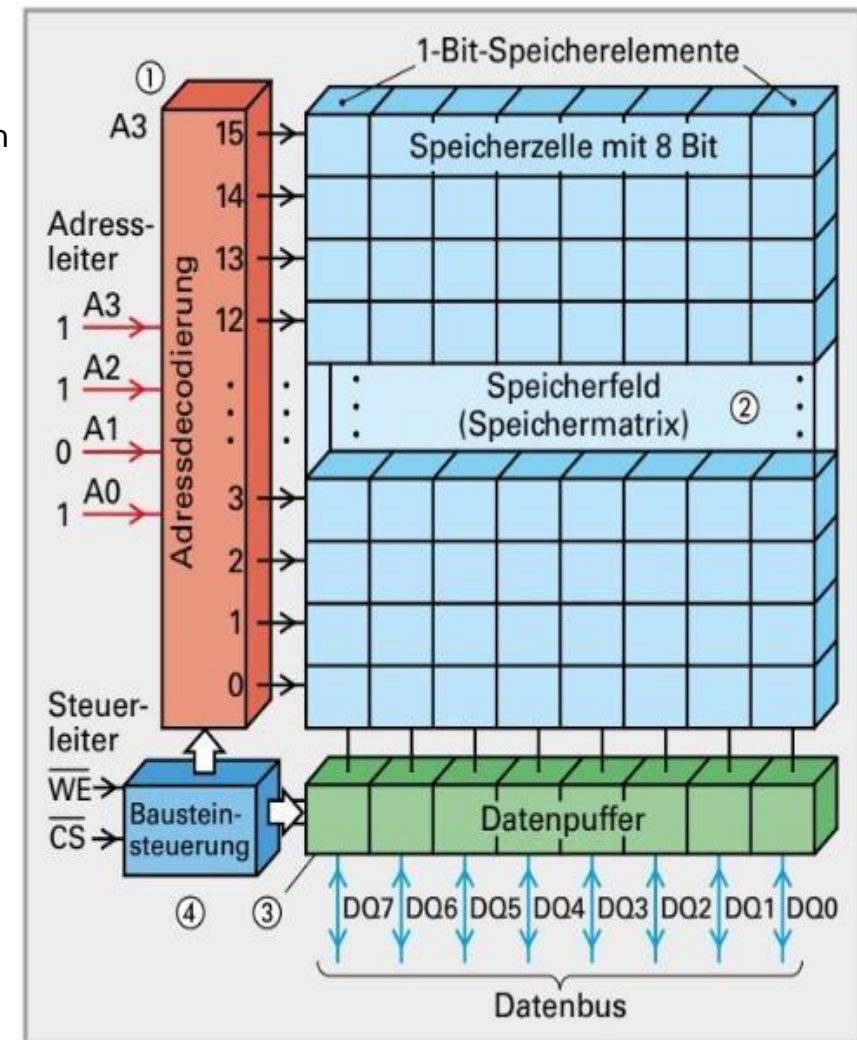


Abb.33: Aufbau Speicher

RAM (Arbeitsspeicher):

# Arbeitsspeicher



- auch 'Hauptspeicher'
- Zwischenspeichern von
  - Daten für CPU
  - Ergebnisse
- schnell

	Festplatte	Arbeitsspeicher
Datenrate	100 MB/s	25 GB/s
Zugriffszeit	7 ms	60 ns

**temporär**

→ 250 : 1

→ 120,000 : 1

# RAM



→ random access memory

= wahlfreier Speicher



## VRAM

- volatil/flüchtig
- Speichern nur mit Stromzufuhr

## NVRAM

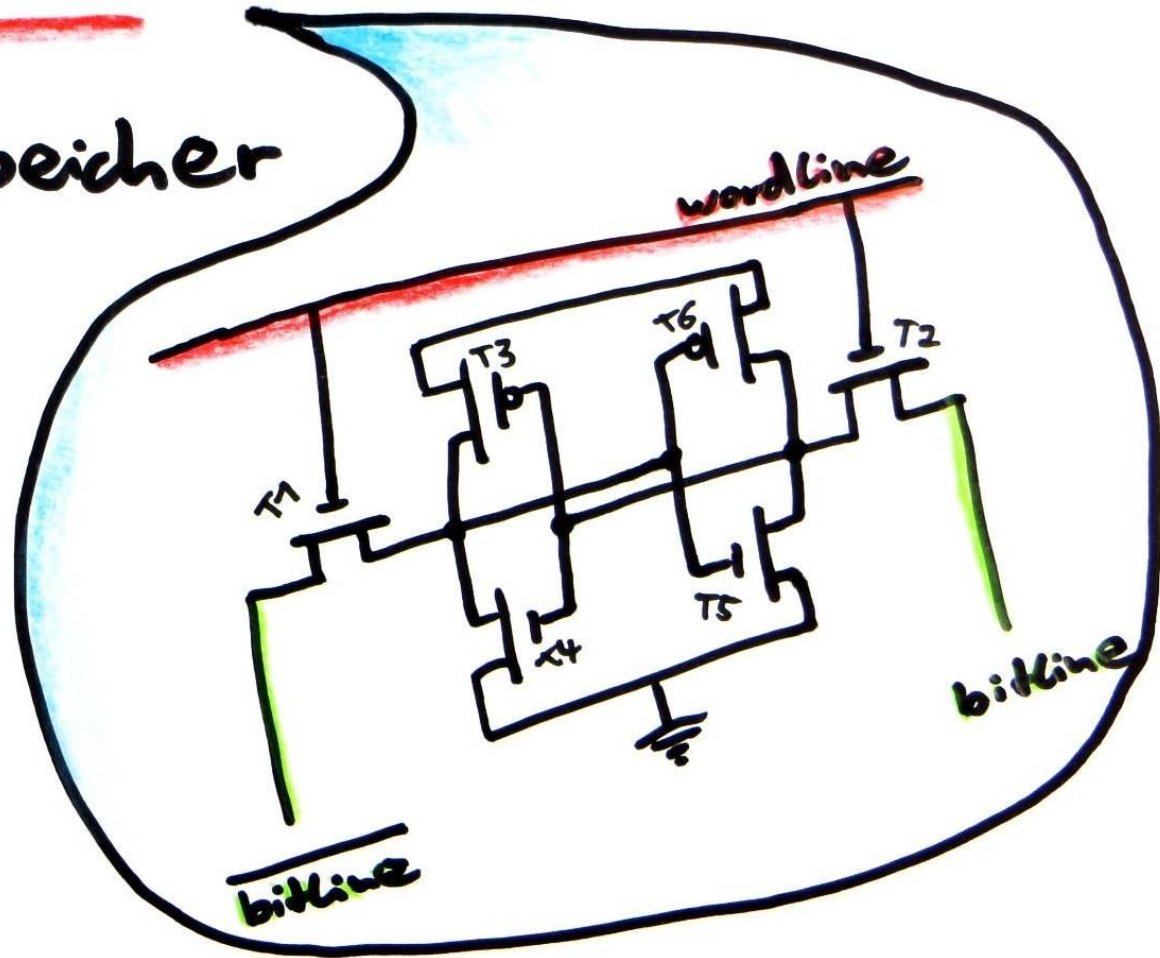
- nicht volatil/flüchtig
- Speichern ohne Stromzufuhr



# SRAM

- statischer Speicher

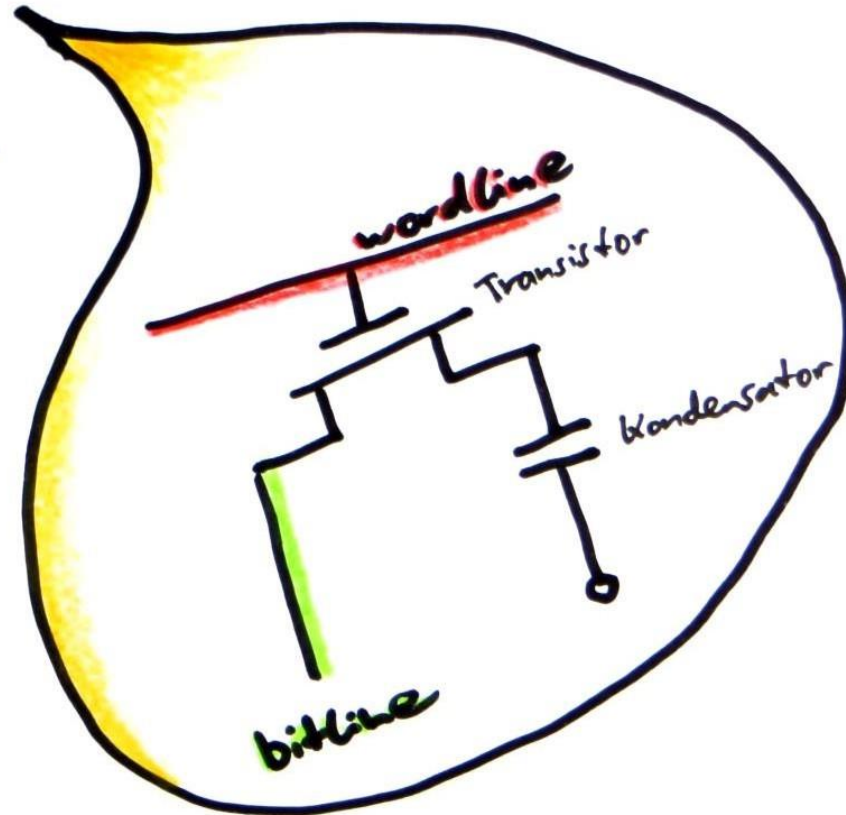
- + schnell
- + keine Auffrischung
- komplex  
⇒ groß





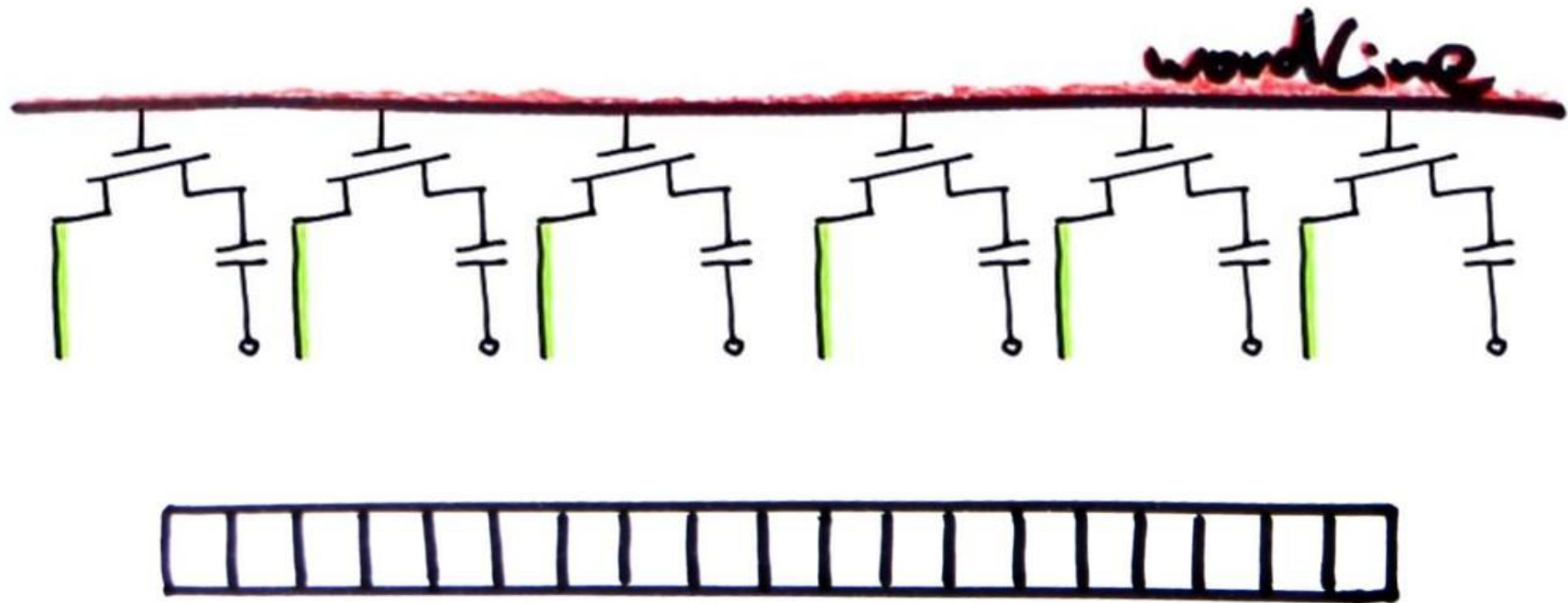
# DRAM

- dynamischer Speicher
- + einfach  
⇒ klein
- Auffrischung  
↓
- langsam

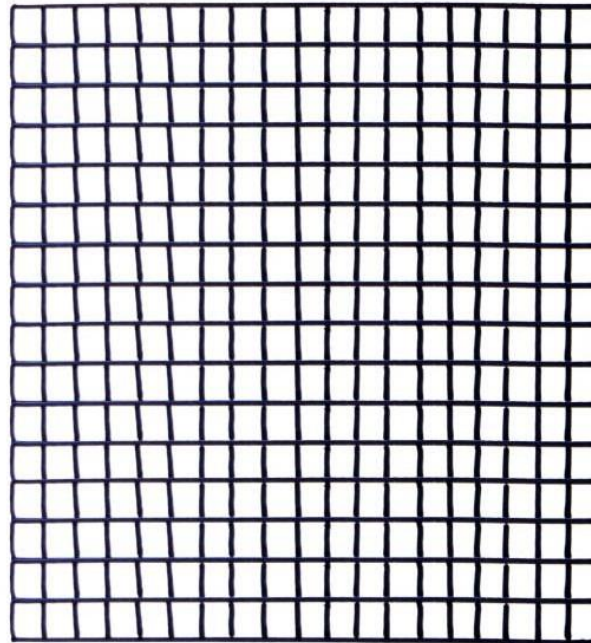


**Speichgröße > Schnelligkeit** ⇒ DRAM-Arbeitspeicher

# Speicherzeile / Page



# Speicherfeld



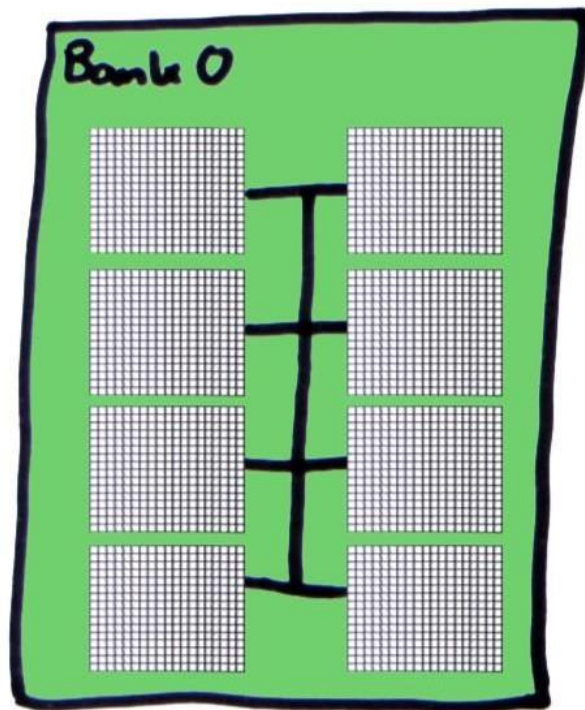
← Zeile 4096

↑ Spalte 1024

$$4096 \text{ Bit} \times 1024 \text{ Bit} = 4 \text{ MBit}$$

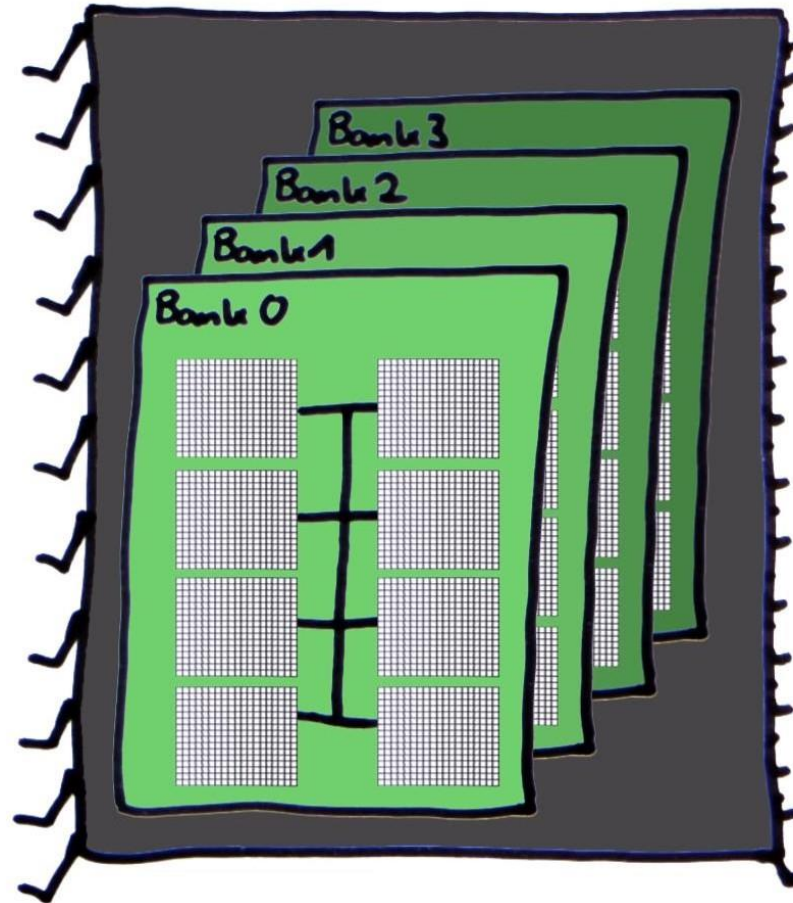


# Speicherbank



$$8 \times 4 \text{ MBit} = 32 \text{ MBit}$$

# Speicherchip



$$4 \times 32 \text{ MBit} = 128 \text{ MBit}$$

$$128 \text{ MBit} / 8 = 16 \text{ MByte}$$

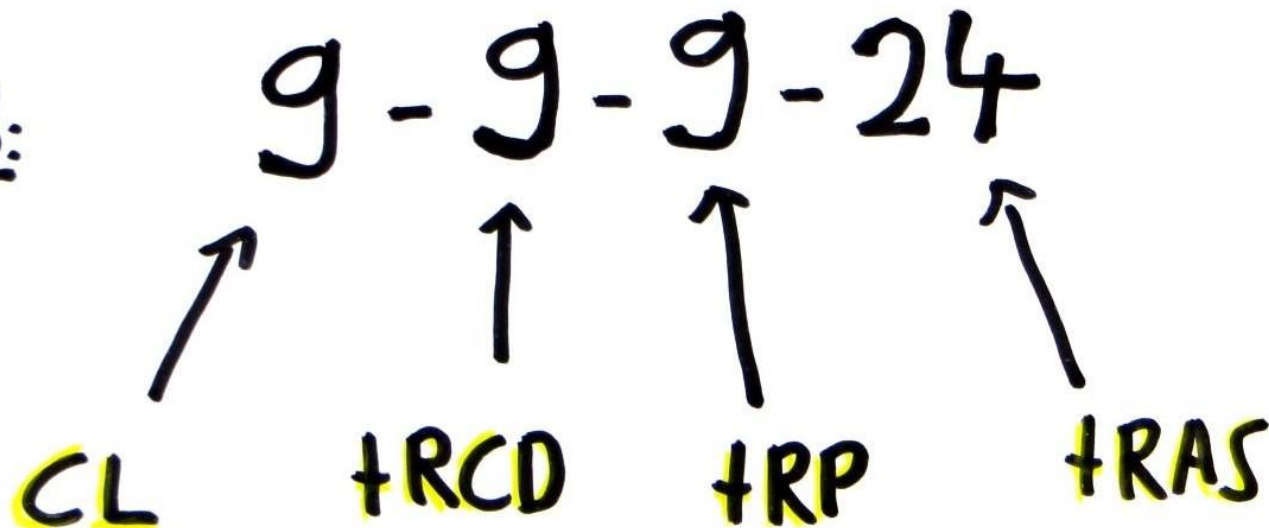
## Lesen / Schreiben

1. Ansprechen von Ziel-Chip/Bank
2. Adresse + Zeilendekoder → Zeile
3. Aktivieren der Speicherzeile
4. Adresse + Spaltendekoder → Spalte
5. Datenleitung an Bitline
  - ↳ Lesen
  - ↳ Schreiben
6. Deaktivieren der Speicherzeile

# Latenzen



z.B.:



- **CL oder CAS (Column Address Strobe Latency):** Die Anzahl der Zyklen, die zwischen der Anforderung von Daten aus dem Speicher durch den Prozessor und der Rückgabe der Daten vergehen.
- **tRCD (Row Column Delay):** Die Anzahl der Zyklen, die zwischen der Aktivierung der Zeile (RAS) und der Spalte (CAS), in der die Daten in der Matrix gespeichert werden, vergehen.
- **tRP (Row Precharge Time):** Die Anzahl der Zyklen, die zwischen der Deaktivierung des Zugriffs auf eine Datenzeile und dem Beginn des Zugriffs auf eine andere Datenzeile vergehen.
- **tRAS (Row Active Time):** Wie lange der Speicher warten muss, bis die nächste Speicherzugriffsanforderung ausgelöst werden kann.

↳ SDRAM

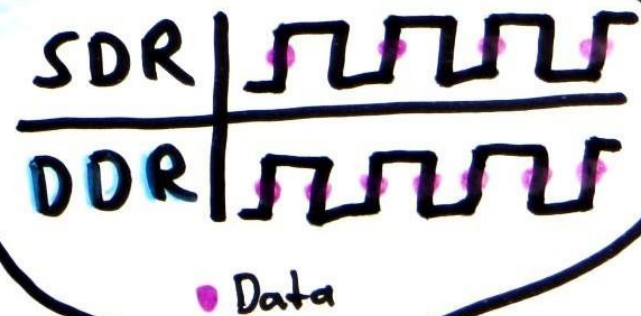
„Synchronous DRAM“

- DRAM Taktrate synchron zu Systembus

→ DDR

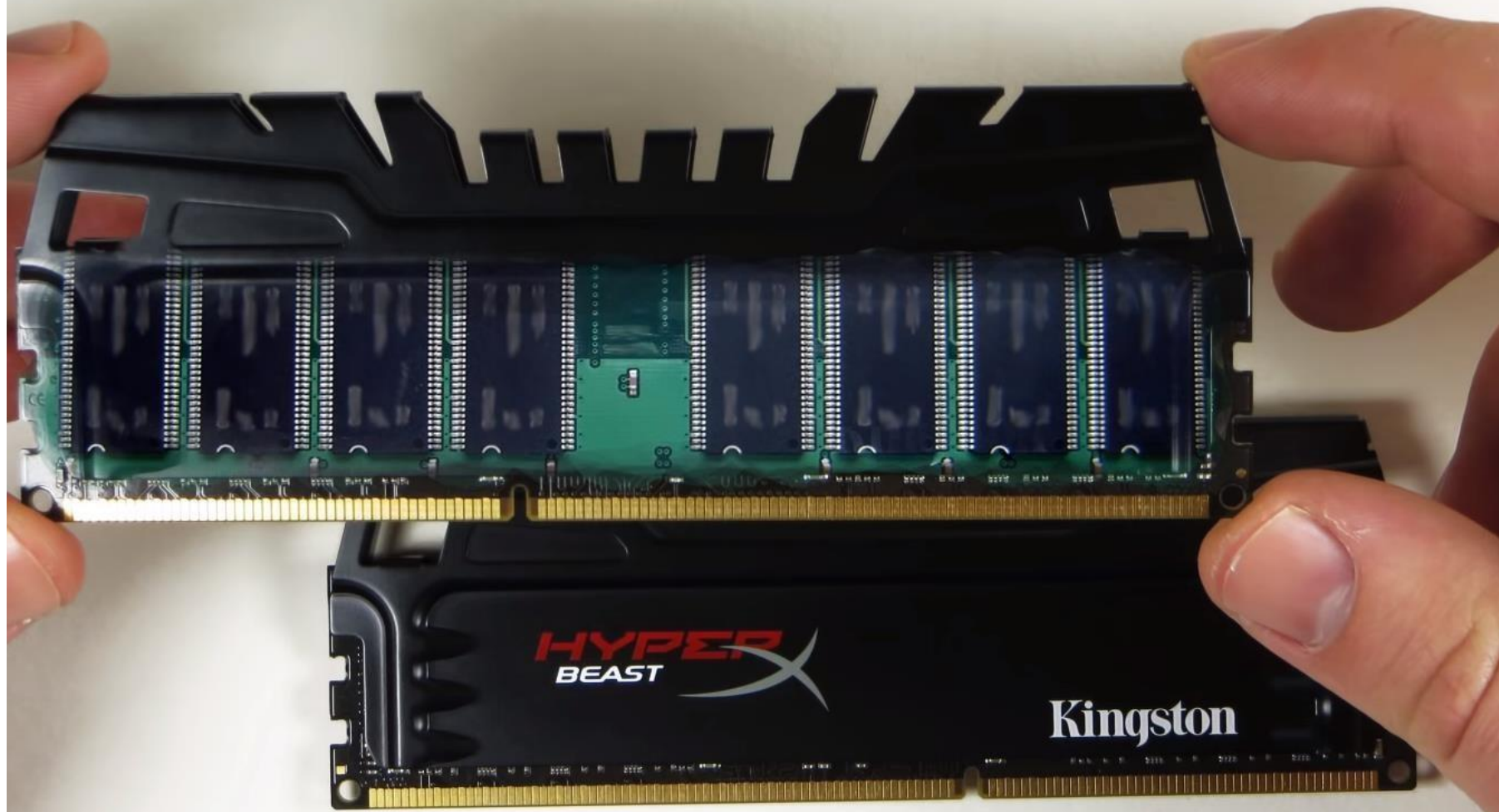
„Double Data Rate“

	Taktrate	
DDR-400	200MHz	→ 400MHz
DDR2-800	400MHz	→ 800MHz
DDR3-1600	800MHz	→ 1600MHz





$$8GB / 16 = 500MB$$





## **ROM & Co:**

Nichtflüchtiger Speicher wird für die Aspekte des Computers verwendet, die sich nicht ändern, wie beispielsweise für das anfängliche Hochfahren des Computers (BIOS) oder die Firmware-Anweisungen, die einen Drucker zum Laufen bringen.

- ROM = **Read Only Memory**
- Zustand einer Speicherzelle definiert durch feste Verdrahtung von Bauelementen zwischen Wort- und Bitleitungen
- Daten müssen bereits beim Herstellungsprozess berücksichtigt werden
- Vorteil: sehr kompakter Schaltungsaufbau, da optimiert auf zu speichernde Daten
- Nachteil: sehr unflexibel (für die Änderung eines Bits muss die Schaltung geändert werden) -> reine ROMs sind sehr selten

## **PROM:**

- PROM = **P**rogrammable **ROM**
- Information wird „eingebrannt“
- Hierfür wird eine Maske verwendet, die für die Erzeugung der einzelnen Schichten des Chips verwendet wird
- Vorteil: flexibler wie reine ROMs, gleicher Herstellungsprozess für verschied. Chips (nur unterschiedl. Maske)
- Nachteil: Programmierung nur durch Hersteller möglich

## **EPROM:**

- EPROM = **E**rasable **P**ROM
- Spezielle Speichertransistoren (Floating-Gate-MOSFETs) repräsentieren Zustand eines Bits
- Programmierung durch spezielles Gerät
- Löschar durch Bestrahlung mit UV-Licht  
(Chips ist mit lichtdurchlässigem Fenster versehen)
- Programmierzeit: 50ms
- Löszeit: 20 Minuten
- Vorteil: Chip mehrfach nutzbar
- Nachteil: Löschen umständlich und langsam

## **EEPROM:**

- EPROM = **E**lectrical **P**ROM
- z.B. Speicherung kleiner Datenmengen in elektronischen Geräten
- Bits werden wie beim EPROM in speziellen Speichertransistoren (Floating-Gate-MOSFETs) gesichert
- Elektronen werden elektrisch aus Speichertransistor abgesaugt
- Schreiben und Löschen erfolgt byteweise -> Chip muss nicht ausgebaut werden
- Programmierzeit: 50ms
- Löszeit: 1ms
- Vorteil: Löschen einfacher und schneller

## **Flash-Speicher:**

- Verwendet in Memory-Cards, Handys, Modems, BIOS, SSD, ...
- Ersatz für Festplatten und Disketten -> weniger empfindlich geg. Stöße etc.
- Kapazität bis zu 512 MB
- Speicherfähigkeit: 10-100 Jahre
- Ca. 100.000 Programmier- und Löschzyklen möglich
- Ähnlich wie ein EEPROM aufgebaut (Floating-Gate-MOSFETs)
- Blockweiser Zugriff z.B. 512 Byte (im Gegensatz zu EEPROM) bei NAND-Technik
- Lesezugriff: ca. 100-200ns
- Block-Schreibzeit: 10-200ms
- Block-Löschzeit: 2-100ms
- Vorteile:
  - > Sehr flexibel
  - > Chip muss zum programmieren nicht ausgebaut werden

**(-> Floating-Gate-MOSFET später bei SSD)**



- einzelnen Speicherzellen bilden bei ROM-Baustein Schnittpunkte zwischen Datenleitungen und den Zeilenleitungen  
-> Diode stellt als Verbindung das High-Bit dar -> ohne Verbindung ist es Low-Bit (Abb.35)
- jede Speicherzelle (1 Bit) besteht aus Diode und einer **Schwachstelle** -> Schwachstellen werden durch Programmier-Gerät zerstört und damit 1 Bit gesetzt -> Zustand bleibt dann für immer erhalten (Abb.36)

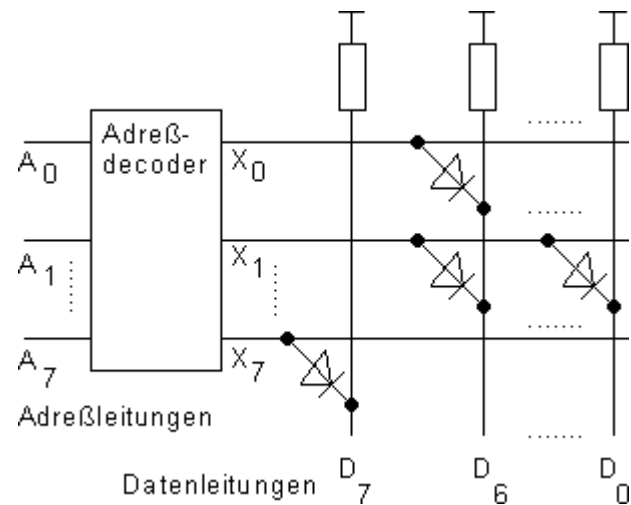


Abb.35: Aufbau ROM-Baustein

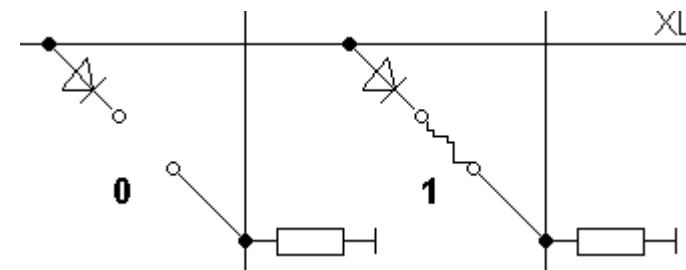


Abb.36: Aufbau PROM-Baustein