Rechnerstrukturen

5. Speicher

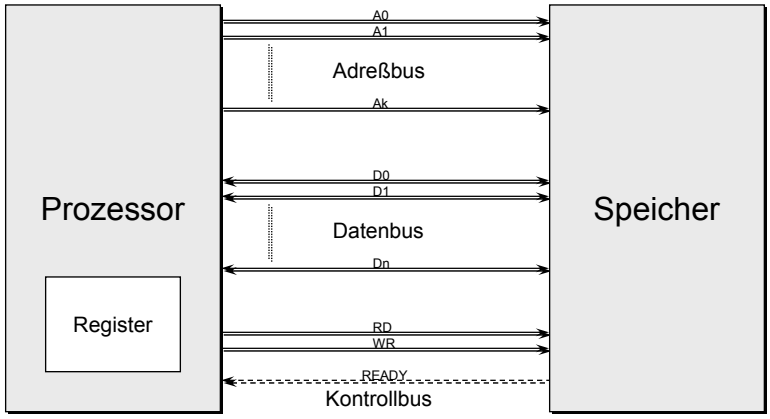
5.1

Inhalt

- Motivation
- Speichertypen
  - RAM / ROM
  - Dynamisches RAM
- Cache-Speicher
  - Voll Assoziativ
  - n-Wege Assoziativ
  - Direct Mapping

5.2

Der Hauptspeicher



5.3

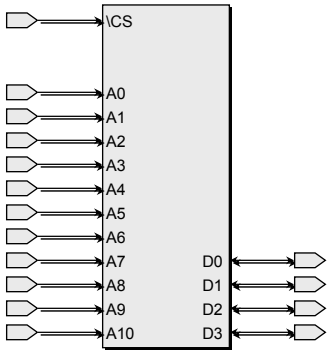
Speichertypen

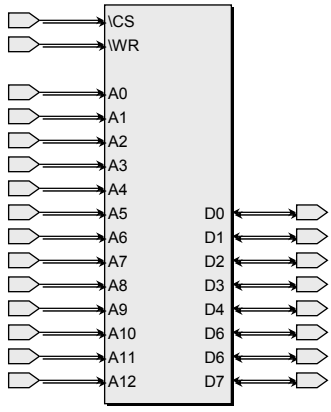
- Lese- und Schreibspeicher (RAM)
  - Flüchtig
  - Statische RAM (Flip-Flop)
  - Dynamische RAM (Kondensatoren)
  - Statisch-dynamische RAM
    - Kondensator im Flip-Flop-Pelz
- Nurlesespeicher (ROM)
  - Nicht-flüchtig
  - PROM = Einmal programmierbar
  - EPROM = Erasable ROM (UV-Licht)
  - EEPROM = Electrically erasable ROM
  - Flash ROM = “Fast ein nicht-flüchtiges RAM”

5.4

Beispiel ROM und SRAM

- 2048 x 4 Bit ROM
- 8128 x 8 Bit statisches RAM

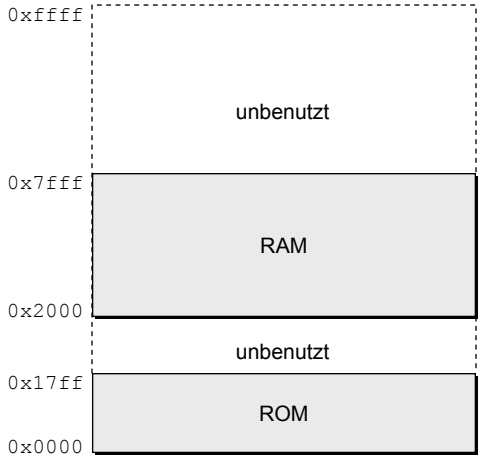




5.5

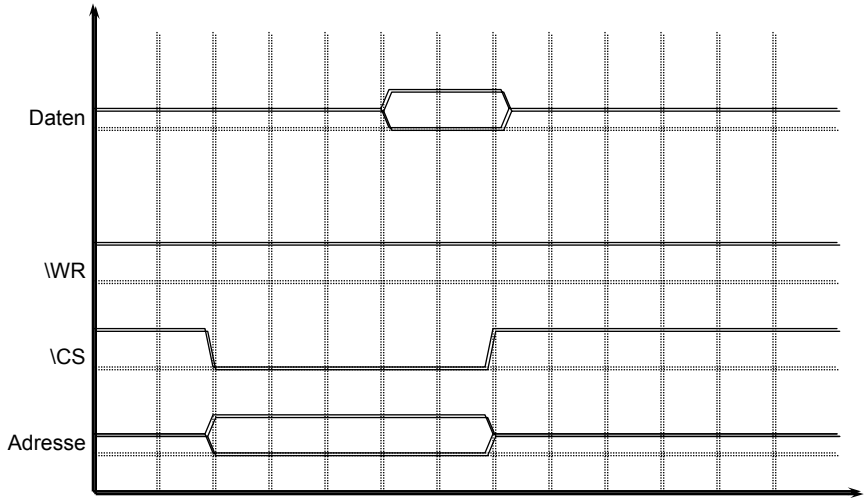
Aufbau eines 64 K x 16 Bit-Speichers?

- Gewünschtes Speicherlayout



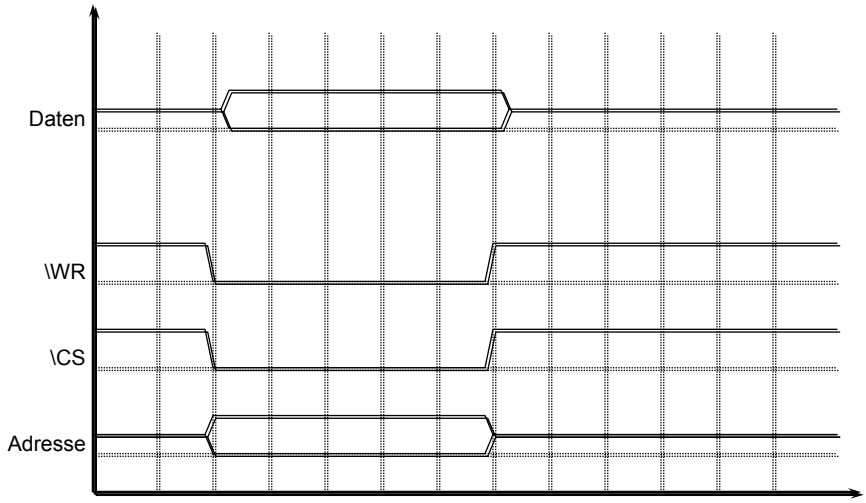
5.6

Lesezyklus



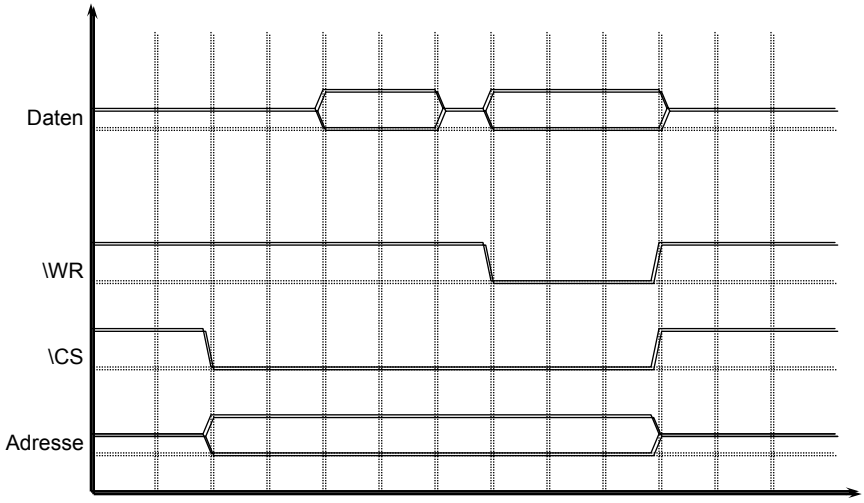
5.7

Schreibzyklus



5.8

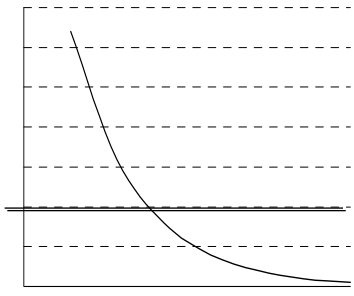
Read-Modify-Write-Zyklus



5.9

Dynamisches RAM (DRAM)

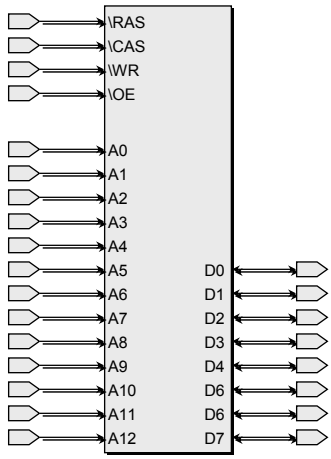
- 1 Kondensator pro Bit
  - Erheblich kleiner als statisches RAM
    - hohe Integrationsdichte
    - geringer Verbrauch
  - Kapazitiv = verhältnismäßig langsam
- Integrationsdichte
  - 64 Mbit-Chips Stand der Technik
  - 256 Mbit-Chips demnächst
  - 1 Gbit-Chips im Labor
- Nachteil
  - Kondensator verliert langsam Ladung
  - Schwellwert für sicheres Erkennen eines 1-Bits
  - Refresh ca. alle 4 bis 64 msec



5.10

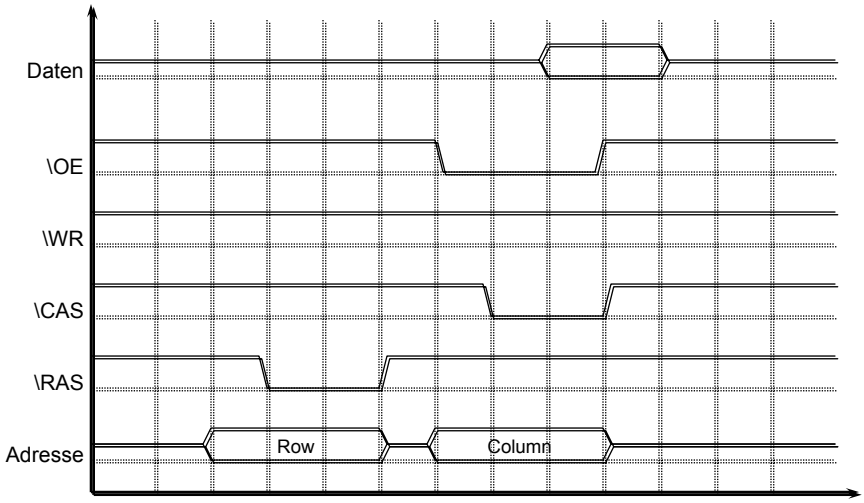
Typische Pin-Belegung

- Viele Adreßleitungen
  - 64 Mbit = 26 Leitungen
  - 256 Mbit = 28 Leitungen
  - 1 Gbit = 30 Leitungen
- Minimierung der Anschlüsse
  - Adresse in zwei Schritten
    - Row
    - Column
  - Adreßleitungen halbiert
  - RAS = Row Address Strobe
  - CAS = Column Address Strobe
- Spezielle Freischaltung der Ausgänge
  - OE = Output Enable



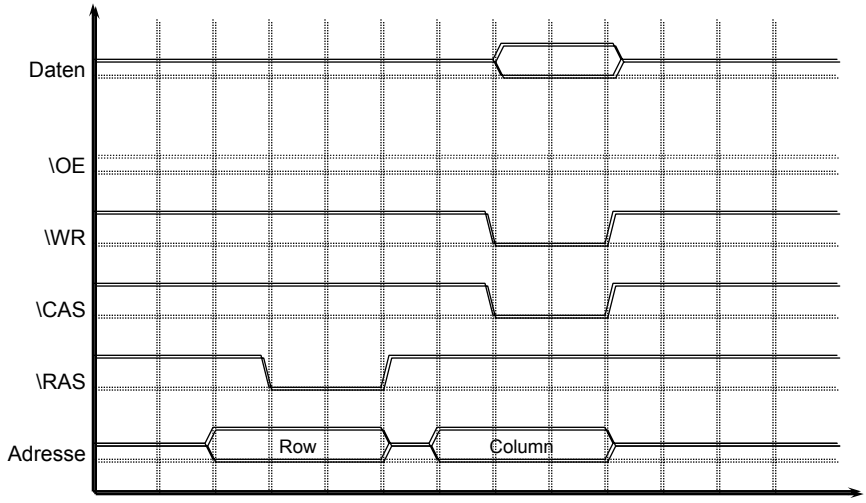
5.11

Lesezyklus



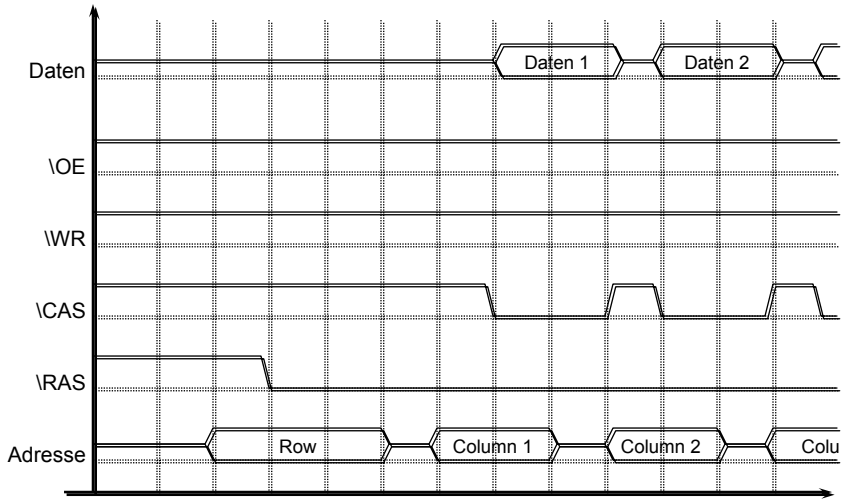
5.12

Schreibzyklus



5.13

Fast-Page-Mode



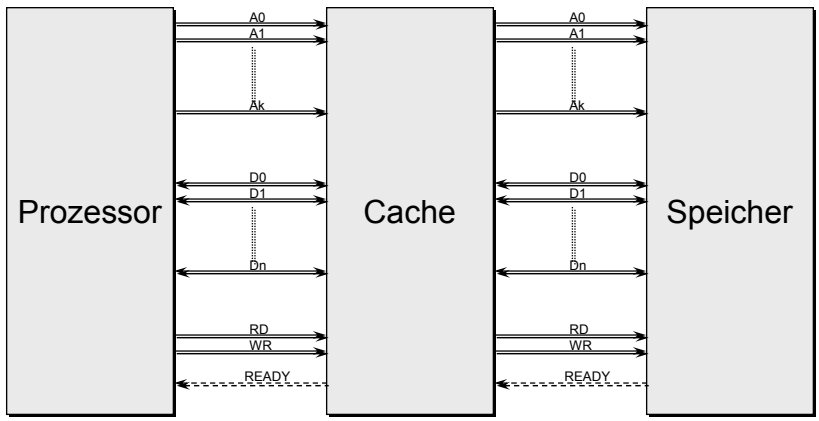
5.14

Refresh

- Bei jedem Zugriff auf eine Reihe mittels \RAS (lesend oder schreibend), wird diese Reihe aufgefrischt
  - Jede Reihe mindestens alle 8 bis 64 msec ansprechen
- Verschiedene Techniken
  - RAS-Only-Refresh
    - Nur RAS-Signal für Refresh
  - CAS-Before-RAS-Refresh
    - Aktiviert internen Row-Zähler
- Modi
  - Burst-Modus: Alle x msec alle Reihen auffrischen
  - Distributed-Modus: Auffrischen einzelner Reihen zeitlich verteilen
  - Hidden-Refresh: Refresh während zugriffsfreier Zeiten
- DRAM-Controller
- SDRAM: DRAM mit internem Controller

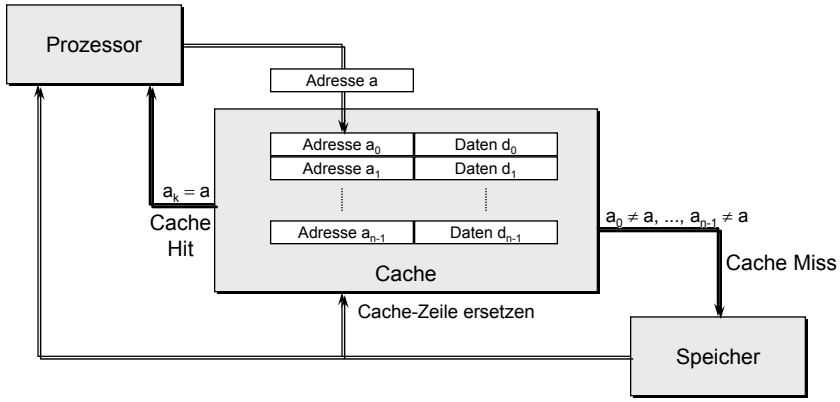
5.15

Der Cache



5.16

Funktionsweise



- Idealerweise gleichzeitiger Vergleich aller a_i mit a
  - Assoziative Suche
- Ersetzungsstrategie
  - Welche Zeile im Cache wird bei Cache Miss ersetzt

5.17

Ziele

- Eigenschaften von Cache und Hauptspeicher:
  - Zugriffsgeschwindigkeit t_C und t_M
  - Kapazität C_C und C_M
  - Preis pro Byte P_C und P_M

$$t_C << t_M, P_C >> P_M, C_C << C_M$$

- Eigenschaften des Cache-Hauptspeicher-Verbund:

$$t_{eff} = p \cdot t_C + (1-p) \cdot t_M, P_{eff} = \frac{P_C \cdot C_C + P_M \cdot C_M}{C_C + C_M}, C_{eff} = C_M$$

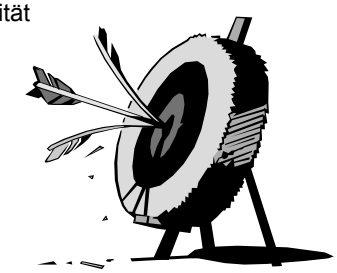
- Ziel

$$t_{eff} \approx t_C \Rightarrow p \approx 1, P_{eff} \approx P_M \Rightarrow C_C << C_M$$

5.18

Trefferrate

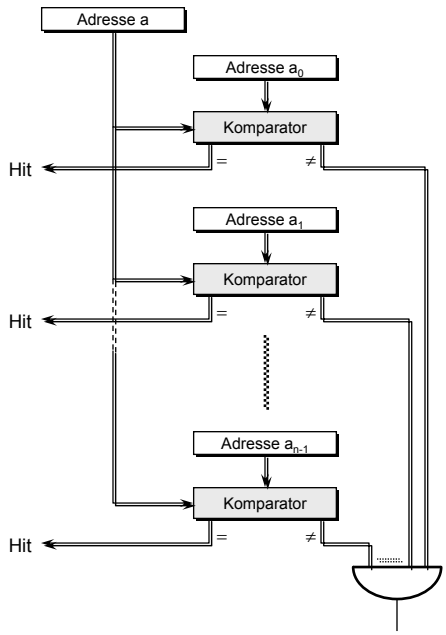
- p = 0.85 - 0.95 wird meist erreicht
- Referenzlokalität
  - Zugriff auf Instruktionen in einer Schleife
  - Sequentieller Zugriff auf Instruktionen und Daten
  - Häufig referenzierte und ev. veränderte Daten
  - ...
- Gilt primär nur für prozedurale und imperative Sprachen
  - Keine so ausgeprägte Referenzlokalität bei funktionalen und logischen Programmiersprachen



5.19

Voll-assoziativer Cache

- Hoher Aufwand
  - Komparator für jedes a_k
  - Großer Flächenbedarf
- Keine Duplikate
  - Jede Adresse wird maximal einmal gespeichert
- Ersetzungsstrategie
  - Welche Zeile wird bei Miss ersetzt
  - LRU gängig: Verhalten in der näheren Vergangenheit dem Verhalten in der näheren Zukunft sehr ähnlich

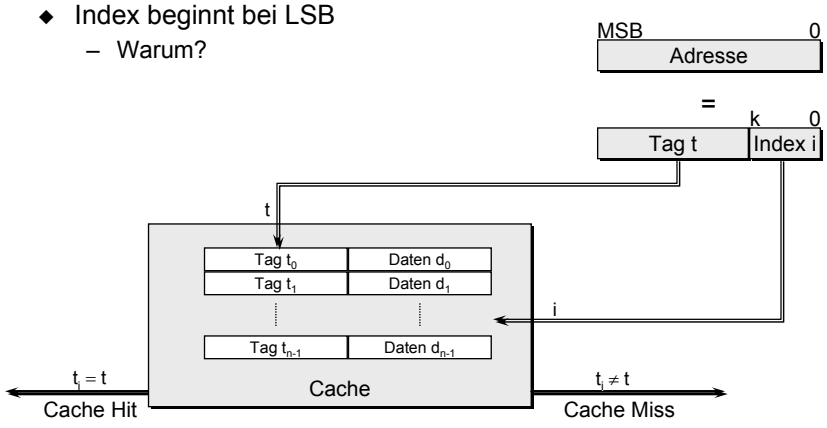


Miss = Nachladen und Ersetzen

5.20

Direct-Mapped Cache

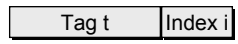
- Cache-Zeile wird direkt über die Adresse bestimmt
  - Tag wird als Adreßanteil in der Cache-Zeile gespeichert
  - Index bestimmt den Zeileneintrag
- Index beginnt bei LSB
  - Warum?

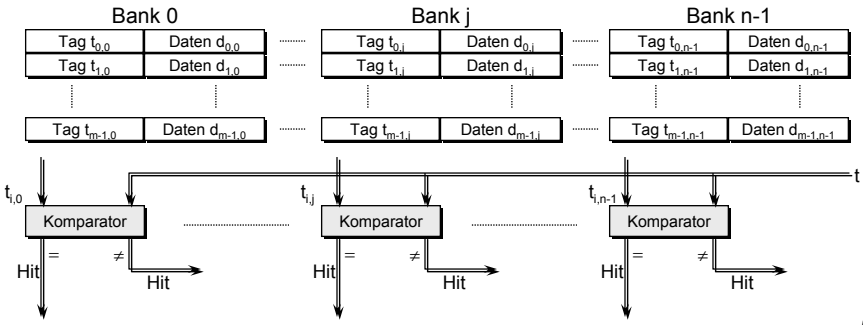


5.21

n-Wege-assoziativer Cache

- "Halbassoziativer" Cache
  - Einsteig in den Cache analog zu Direct Mapped
  - n parallele Tabellen werden durchsucht
  - Zeilenersetzung auf n Cache-Zeilen
- n = 2 oder n = 4 gängig





5.22