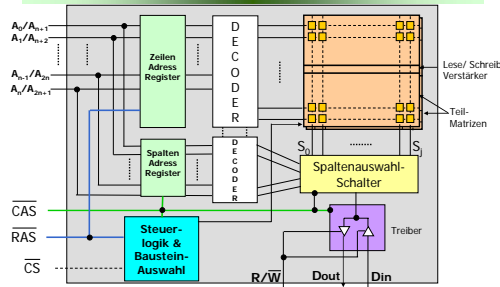


6.4. Dynamische RAM Bausteine

- Größte Integrationsdichte aller Halbleiterspeicher
- Bitweise organisiert, getrennter Dateneingang und Datenausgang
- Speicheradressen gemultiplext (spart Anschlüsse) und im Speicherchip in Registern zwischengespeichert:
 - Auswahl der Zeilenadresse über das **RAS-Signal** (Row Address Select)
 - Auswahl der Spaltenadresse über das **CAS-Signal** (Column Address Select)

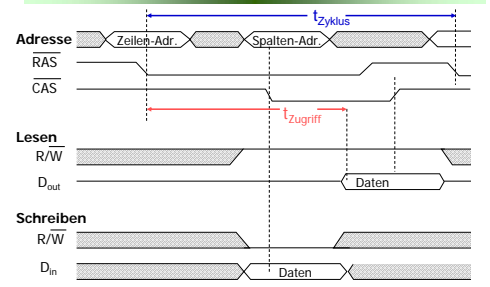
6.4. Dynamische RAM-Bausteine



Pin-Belegung DRAM-ICs mit 2Mx8 Organisation

Pin Name	Pin Function
Vcc	Power (+5V)
DQ0	Data In/Out
DQ1	Data In/Out
DQ2	Data In/Out
DQ3	Data In/Out
W	Read/Write Input
RAS	Row Address Strobe
A11	Address Inputs
A10	Address Inputs
A0	Address Inputs
A1	Address Inputs
A2	Address Inputs
A3	Address Inputs
Vcc	Power (+5V)
Vss	Ground
DQ0-7	Data In/Out
CAS	Column Address Strobe
OE	Data Output Enable

Adressierung eines dynamischen RAM-Bausteins



Adressierung eines dynamischen RAM-Bausteins

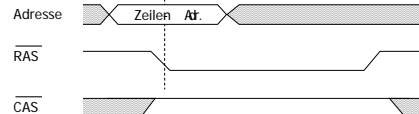
Adressieren:
negative RAS-Flanke übernimmt die Zeilenadresse ins Adressregister, negative CAS-Flanke die Spaltenadresse

Lesen:
Daten erscheinen eine gewisse Zeit nach der negativen CAS-Flanke am Ausgang, Zyklus wird durch CAS = 1 wieder beendet

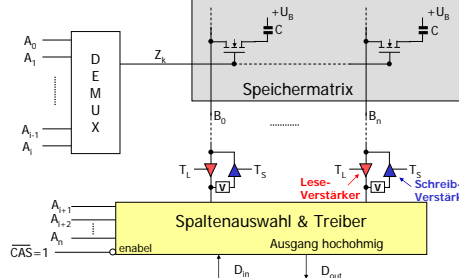
Schreiben:
Das zu schreibende Datum muss gleichzeitig mit Spaltenadresse am Dateneingang anliegen. Datenübernahme geschieht mit negativer CAS-Flanke

Auffrischen dynamischer RAMs

- Geschieht zeilenweise, jede Zeile muss aufgefrischt werden (alle 2 msec)
- Nur die Zeilenadresse wird an den Baustein angelegt, RAS = 0 und CAS konstant auf 1



Aufbau der Auffrischlogik

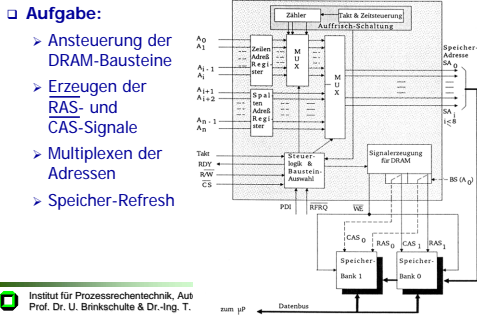


Auffrischen dynamischer RAMs

- Der Zeileninhalt wird zunächst, gesteuert vom Takt T_L über die Leseverstärker ausgelesen
- Über eine kleine Verzögerung V wird dieser Inhalt dann mittels der Schreibverstärker gesteuert vom Takt T_S zurückgeschrieben

Um die Leitungskapazitäten besser zu verteilen, werden bei hochintegrierten Bausteinen die Schreib-/Leseverstärker auch oft in die Mitte der Spaltenleitungen verlegt (statt am Ende wie im Bild dargestellt)

DRAM-Controller



DRAM-Controller

Bis zu 4 getrennte RAS- / CAS-Ausgänge bei DRAM-Controllern → Unterstützung von bis zu 4 Speicherbanken

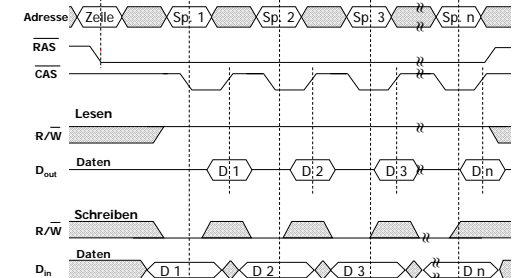
Vorteil:
durch Zyklusüberlappung (*interleaving*) kann der Zugriff verkürzt werden.

Die Auswahl der Speicherbank (BS) wird z. B. mit A_0 beschaltet → fortlaufende Speicheradressen liegen in unterschiedlichen Speicherbanken

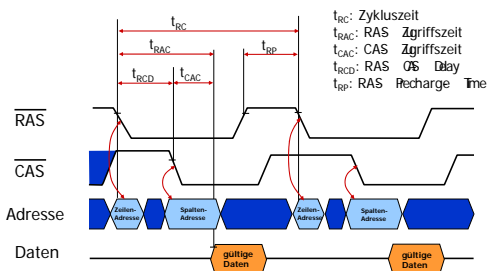
6.5 Techniken zur Zugriffsbeschleunigung

- Ausgangspunkt:**
Zwischen Prozessor und Speicher oder Speicher und Speicher werden keine einzelnen Bytes übertragen, sondern benachbarte Gruppen von Bytes (Blöcke)
- Beschleunigter Zugriff auf den Speicher-Baustein, wenn alle zu lesenden oder schreibenden Speicherzellen in einer Zeile (Seite) liegen.
- Die Zeilenadresse wird bei einem wiederholten Zugriff auf diese Zeile (auch *page* genannt) nur einmal angelegt (wird im Register gespeichert). Dann werden in schneller Folge die Spaltenadressen angelegt (*fast page mode: FPM-DRAM*) → erheblich beschleunigter Zugriff

Seitenzugriff bei DRAMs



DRAM Timing-Parameter



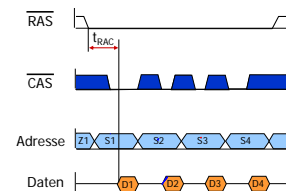
DRAM Timing-Parameter

- t_{RAC} : minimale Zeitdauer, die zwischen der fallenden Flanke von RAS bis zur Ausgabe der gewünschten Daten vergeht.
 - $t_{RAC} = 60$ ns (bei einem 4 MB DRAM)
- t_{RC} : Minimale Zeitdauer von Beginn eines Zeilenzugriffs bis zum nächsten (Zykluszeit).
 - $t_{RC} = 110$ ns bei einem 4Mbit DRAM mit t_{RAC} von 60 ns
- t_{CAC} : minimale Zeitdauer, die zwischen der fallenden Flanke von CAS bis zur Ausgabe der gewünschten Daten vergeht.
 - $t_{CAC} = 15$ ns bei einem 4Mbit DRAM mit t_{RAC} von 60 ns.
- t_{RC} : Minimale Zeitdauer vom Beginn eines Spaltenzugriffs bis zum nächsten.
 - $t_{RC} = 35$ ns bei einem 4Mbit DRAM mit t_{RAC} von 60 ns

Fast Page Mode DRAM (FPM-DRAM)

- Oft liegen aufeinander folgende Speicherzugriffe in der gleichen Zeile des DRAMs. Das wird bei FPM-DRAMs ausgenutzt:
- Initialisieren des ersten Lesezugriffs läuft wie bei normalem DRAM.
- Nach dem ersten Lesezyklus lässt die Speichersteuerung das RAS-Signal einfach auf aktiv → Zeile (*page*) bleibt aktive.
- Bei den folgenden Lesezugriffen übergibt die Speichersteuerung nur noch jeweils eine neue Spaltenadresse an das DRAM.
- RAS-Precharge-Zeit (t_{RP}) und die RAS-CAS-Delay (t_{RCD}) fallen bei den Folgezugriffen weg.

Timing-Diagramm eines FPM-DRAM



Zugriffszeit reduziert sich erheblich. Erst wenn der Speicherzugriff des Prozessors auf eine andere Zeile erfolgt, muss der Chipsatz einen normalen Lesevorgang initialisieren.

FPM-DRAM

Typische FPM-DRAM-Zugriffszeiten: 60 oder 70 ns

CAS Zugriffszeit (t_{CAS}) bei einem 60 ns Baustein im FPM Mode beträgt ca. 40 ns → Daten können im Abstand von 40 ns gelesen werden.

PC mit 66 MHz Bustakt (Taktperiode = 15 ns)

→ Prozessor kann im FPM Mode nur bei jedem dritten Takt auf eine Zeile im Speicher zugreifen.

Initialisierung eines Lesevorgangs mit dem Anlegen von Zeilen- und Spaltenadresse dauert dagegen ganze fünf Takte 75 ns.

Maximale Datentransferrate: 200 Mbyte/sec

Prozessoren ab Intel Pentium führen durch ihren 64 Bit breiten Datenbus Speicherzugriffe mit 8 Byte durch. Bei einer CAS Zugriffszeit von 40 ns lassen sich 64 Bit in diesem Zeitraum übertragen.

EDO-DRAM

Typische EDO-DRAM-Zugriffszeiten: 50 bis 70 ns CAS Zugriffszeit (t_{CAS}) bei einem 60 ns EDO DRAM Baustein auf 25 ns (gegenüber 40 ns bei FPM DRAM)

PC mit 66 MHz Bustakt (Taktperiode = 15 ns)

→ Prozessor kann im EDO Mode bei jedem zweiten Takt auf eine Zeile im Speicher zugreifen.

Initialisierung eines Lesevorgangs mit dem Anlegen von Zeilen- und Spaltenadresse dauert, wie bei FPM DRAM fünf Takte.

Maximale Datentransferrate: 300 MByte/sec (bei 64 Bit Datenbus)

Performance-Steigerung von 50 %. In der Praxis fallen die Geschwindigkeitsgewinne wesentlich geringer aus (wenige Prozente).

EDO-DRAM

EDO-DRAM (Extended Data Output):

Die Datenausgabe wird beim Lesen vom CAS-Signal mittels interner Pufferung entkoppelt.

→ die Daten stehen länger am Ausgang zur Verfügung
→ bessere Verschachtelungsmöglichkeiten bei Lesezugriffen

Der Prozessor kann Daten auslesen, während die Speichersteuerung eine neue Spaltenadresse an das DRAM übergibt.

Zusammenfassung: FPM, EDO

- Sie arbeiten asynchrone zum Systembus
- Für eine Datenübertragung ist ein Handshaking-Verfahren notwendig. Ein Lesevorgang läuft wie folgt:
 - Prozessor signalisiert der Speichersteuerung, dass eine Adresse anliegt.
 - Wenn die Daten am Ausgang des DRAMs bereitliegen, teilt die Speichersteuerung dem Prozessor dies mit (BRDY-Signal). Erst dann liest der Prozessor die Daten ein.
 - Dazwischen ist die CPU im Leerlauf und führt Wartezyklen aus.
- Varianten von EDO-DRAM (BEDO-DRAM) können die Daten ohne Wartezyklen liefern, aber nur bis zu einem Bustakt von 66 MHz.

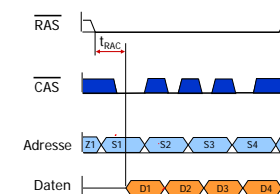
EDO-DRAM

Weiterentwicklung der FPM DRAMs. Durch eine einfache Modifikation der Steuerung der eine Geschwindigkeitssteigerung erreicht.

- EDO DRAMs sind um einen Latch-Speicher am Ausgang erweitert.
- Die gelesenen Daten bleiben bis zum nächsten Aktivieren des CAS-Signals gültig.
- Der Prozessor kann Daten auslesen, während die Speichersteuerung eine neue Spaltenadresse an das DRAM übergibt.
- Durch dieses „Pipelining“ verkürzt sich die Wartezeit zwischen zwei aufeinander folgenden CAS-Impulsen → höherer Datendurchsatz

Schreibzugriffe bleiben wie bei FPM-Speicher unbeschleunigt und entsprechen von der Performance her den normalen Standard-DRAM.

Timing-Diagramm eines EDO-DRAM



Eine neuer Lesevorgang kann beginnen, bevor der alte abgeschlossen ist.

SDRAM

SDRAM-Technologie hat sich (durch intensive Unterstützung von Intel) schnell durchgesetzt und beherrscht heute den Speichermarkt.

- Alle Ein- und Ausgangssignale sind synchron zum Systemtakt.
- Prozessor, Chipsatz und Speicher kommunizieren über ein Bussystem, das synchron mit der gleichen Frequenz getaktet ist.

Intern sind SDRAMs aus zwei unabhängigen Speicherbanken aufgebaut (auch bis zu 4 Speicherbanken).

Nach dem Anlegen der Zeilen- und Spaltenadresse, generiert die Speichersteuerung die nachfolgenden Adressen und führt einen alternierenden und überlappenden Zugriff auf die beiden Speicherbanken selbstständig aus

SDRAM

SDRAMs (Synchrone Dynamische RAMs):

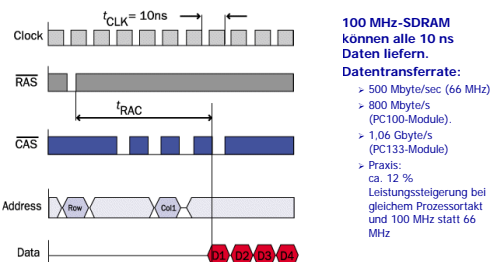
SDRAMs arbeiten synchron mit dem Systemtakt

Intern bestehen SDRAM-Bausteine aus zwei unabhängigen Speicherbanken. Eine Bank kann vorgeladen werden (Precharge), während die andere Bank einen Lese- oder Schreibzugriff durchführt.

Aktuelle SDRAMs besitzen je nach Kapazität sogar vier interne Speicherbanken.

SDRAMs können Busgeschwindigkeiten von bis zu 100 MHz bearbeiten.

Timing-Diagramm eines SDRAM



100 MHz-SDRAM können alle 10 ns Daten liefern.
Datentransferrate:

- 500 Mbyte/sec (66 MHz)
- 800 Mbyte/s (PC100-Module)
- 1,06 Gbyte/s (PC133-Module)
- Praxis: ca. 12 % Leistungssteigerung bei gleichem Prozessortakt und 100 MHz statt 66 MHz

DDR-DRAM

Nächste Stufe der SDRAM-Entwicklung (SDRAM II). Bestehen intern aus vier unabhängigen Speicherbanken, die parallel „Instruktionen“ bearbeiten können.

Prinzip der DDR-DRAMs:

Erweiterung der Bandbreite durch Nutzung beider Taktflanken. Daten werden bei steigender und fallender Taktflanke übertragen → doppelter Datendurchsatz

Laufzeitverzögerungen sind sehr kritisch, deshalb wird zur Synchronisation nicht nur der Systemtakt, sondern auch ein bidirektionales Strobe-Signal (DQS) benutzt.

DDR-SDRAM

□ **DDR-SDRAM (Double-Data-Rate-SDRAM)**

Die DDR-SDRAMs entsprechen in Bauform und Funktionsweise den "normalen" SDRAM-Modulen, jedoch werden im Gegensatz zu diesen die Speicherzellen zweimal pro Takt ausgelesen bzw. geschrieben. Dadurch erreichen die DDR-SDRAM Module den doppelten Datendurchsatz.

□ **SLDRAM (Sync Link RDRAM)**

Weiterentwicklung der SDRAM-Technologie, die höhere Busfrequenzen erlaubt und damit eine höhere Leistung ermöglicht.

RDRAM/ Concurrent - RDRAM / Direct RDRAM

□ **RDRAM/ Concurrent - RDRAM / Direct RDRAM**

Die RDRAM-Technologie gibt es seit 1995 und hat sich in vielen Workstations und Spielekonsolen bewährt. Der spezielle Bus (Rambus Channel) setzt jedoch ein entsprechendes Design der Hauptplatine voraus. Die Varianten **Concurrent RDRAM** und **Direct RDRAM** unterscheiden sich hinsichtlich der maximalen Taktrate und der eingesetzten Protokolle.

6.6 Organisation des Hauptspeichers

Hauptspeicher:

- lineare Liste von Speicherworten
- Aufgebaut aus Speicherbausteinen
- Zugriffszeit hängt allein von der Art der verwendeten Speicherbausteine ab
- Die Breite des Arbeitsspeichers entspricht i. A. der Breite des Datenbus (8, 16, 32, 64 Bit). Dies entspricht der maximalen Informationsmenge, auf die in einem Buszyklus zugegriffen werden kann.

6.6 Organisation des Arbeitsspeichers

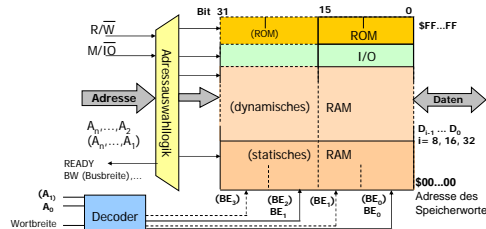
Bei Prozessoren mit einer Datenbusbreite > 8 Bit kann meist immer noch auf einzelne Bytes zugegriffen werden
→ Speicherkapazität wird auch bei breiteren Organisationsformen in Bytes angegeben

Die maximale Kapazität des Hauptspeichers ist durch die Breite des Adressbusses gegeben

- 8-Bit Prozessoren mit 16-Bit Adressbus: 64 kByte
- 16-Bit Prozessoren mit 24-Bit Adressbus: 16 Mbyte
- 32-Bit Prozessoren mit 32-Bit Adressbus: 4 GByte

Speicher-Belegungsplan (memory map)

Gibt an, welche Speicherbausteine für welche Bereiche des Hauptspeichers verwendet wurden



Speicher-Belegungsplan (memory map)

Im Beispiel:

- obere Adressen: ROM, z.B. für nicht flüchtige Teile des Betriebssystems (Bootstrap, BIOS)
- dann: IO-Bereich (Prozessor mit memory mapped IO)
- Rest: RAM
meist dynamische RAM-Bausteine, da diese große Kapazität besitzen und billig sind
Nachteil: Sie sind auch langsam
Aus diesem Grund werden manchmal in kleinen Speicherbereichen auch statische RAMs eingesetzt, auf die ohne Wartezyklen zugegriffen werden kann

Speicher-Belegungsplan (*memory map*)

Die Speicherbreite kann über den Speicherbereich variieren

Die Byte- und Breiten-Auswahl eines Speicherzugriffs erfolgt in der Regel über die niederwertigsten Adressbits (z.B. A_0 , A_1 bei 32 Bit Speicherbreite) sowie spezieller Wortbreite-Signale vom Mikroprozessor

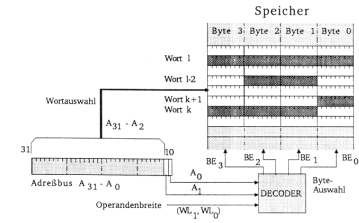
Speicherbelegungspläne werden häufiger noch feiner untergliedert, indem z. B. die genaue Lage von Betriebssystemtabellen im ROM-Bereich oder von Geräten im IO-Bereich angegeben wird.

Adressauswahl

- Der höherwertige Teil der Adresse dient über eine Adressauswahllogik zur Speicherbaustein-Auswahl
→ CS-Signale der Speicherbausteine
Hier wird i. A. auch die Zugriffsrichtung, Speicher- bzw. I/O-Zugriffe sowie eventuelle Wartezyklen ermittelt
- Der mittlere Teil der Adresse geht direkt an die Adresseingänge der Bausteine
- Der niederwertige Teil der Adresse dient zusammen mit Wortbreiten-Signalen zur Wortauswahl innerhalb der Speicherbreite

Adressauswahl

Auswahl eines Bytes, Worts oder Doppelworts in einem Speicherwort



Adressauswahl

zum Beispiel:

oberes Wort: 32 **Wort**, aligned

$A_1 A_0 = 00$ [Startbyte 0], $WL_1 WL_0 = 00$ [Wortbreite 32 Bit]

mittleres Wort: 16 **Wort**, unaligned

$A_1 A_0 = 01$ [Startbyte 1], $WL_1 WL_0 = 01$ [Wortbreite 16 Bit]

unteres Wort: 32 **Wort**, unaligned (2 Speicherzugriffe nötig)

$A_1 A_0 = 01$ [Startbyte 1], $WL_1 WL_0 = 00$ [Wortbreite 32 Bit]