Rechnerarchitektur (AIN 2)

SoSe 2021

Kapitel 5

Die Speicherhierarchie

Prof. Dr.-Ing. Michael Blaich
mblaich@htwg-konstanz.de

Kapitel 5 - Gliederung

Kapitel 5: Die Speicherhierarchie

5.1 Einführung

5.2 Caching

5.3 Virtueller Speicher

5.4 DRAM – Dynamic Random Access Memory

5.5 Zusammenfassung

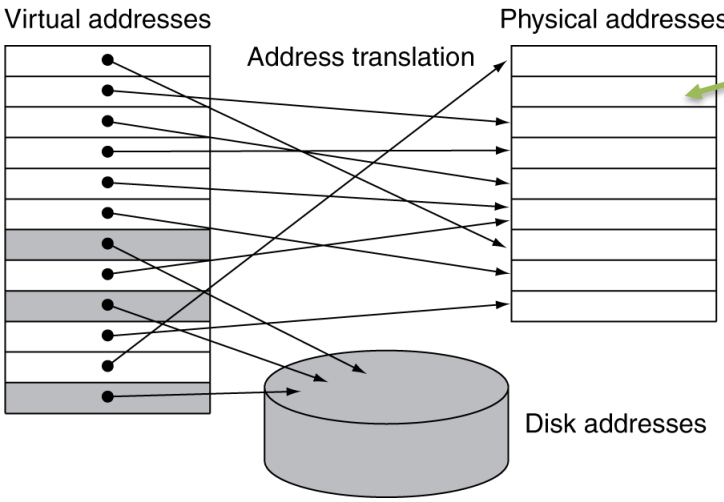
Virtueller Speicher (Virtual Memory, VM)

- Alle Programme teilen sich einen Hauptspeicher
  - jedes Programm erhält einen virtuellen Adressraum, um häufig verwendeten Code und Daten zu halten
    - virtueller Adressraum kann größer sein als physikalischer Speicher
    - nur Teile des virtuellen Speicherbereichs im Hauptspeicher gehalten, der Rest liegt im Swap-Space auf der Festplatte und muss bei einem Zugriff nachgeladen werden
  - virtueller Speicher ist vom Zugriff anderer Programme geschützt

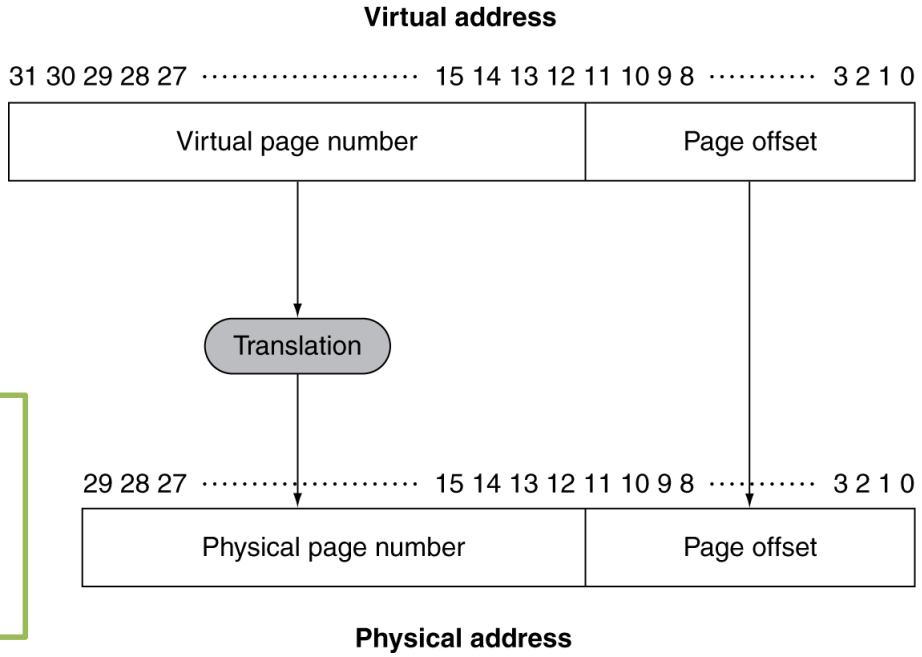
- Nutzung von Hauptspeicher als Cache für sekundären Festplattenspeicher
  - gemeinsam von CPU Hardware und Betriebssystem verwaltet

- CPU und Betriebssystem übersetzen virtuelle Adressen in physikalische Adressen
  - Page (Seite): virtueller Speicher-Block (VM-Block)
  - Page-Fault (Seitenzugriffsfehler): Zugriff auf Seite, die nicht im Hauptspeicher vorliegt, so dass die virtuelle Adresse nicht in eine physikalische Adresse aufgelöst werden kann

Übersetzung von virtuellen in physikalische Adressen



Virtueller Speicher ist in Pages mit einer festen Größe unterteilt: 4-16 kBytes typisch, Desktop/Server bis 64 kBytes, Embedded 1kByte



virtuelle Adresse: Adresse im privaten Adressraum eines Programms

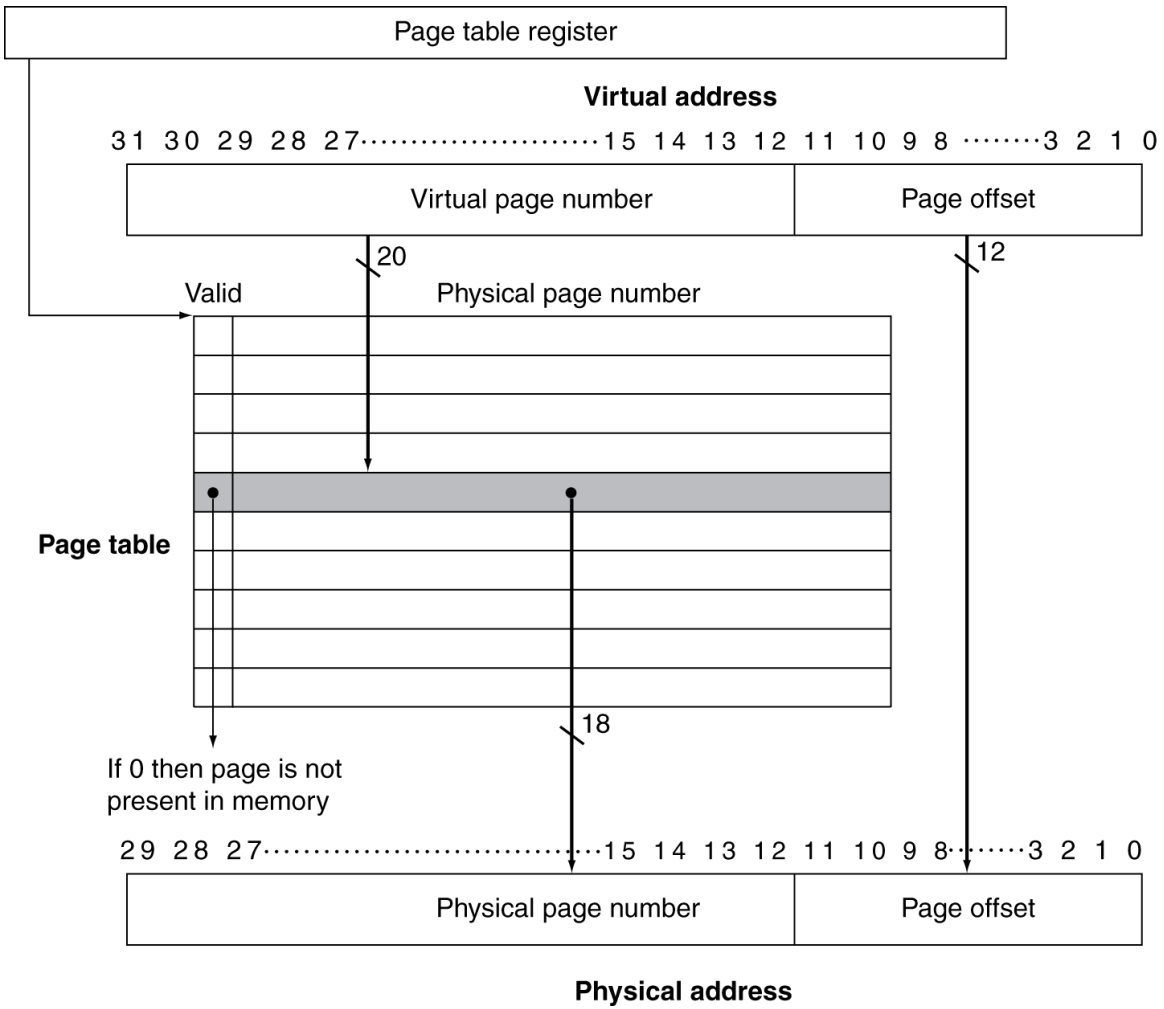
physikalische Adresse: Adresse im Hauptspeicher

Virtueller Speicher kann größer sein als physikalischer Speicher:

- Programme und Daten müssen nicht vollständig im Hauptspeicher vorliegen
- Das Programm sieht einen größeren Hauptspeicher als eigentlich verfügbar

Bei einem Page-Fault lädt das Betriebssystem Seiten von der Festplatte nach, was Millionen von Takten dauern kann. Zur Vermeidung von Page-Faults wird eine voll-assoziative Organisation mit intelligenten Replacement-Strategien verwendet.

Adress-Auflösung mit Page Tables



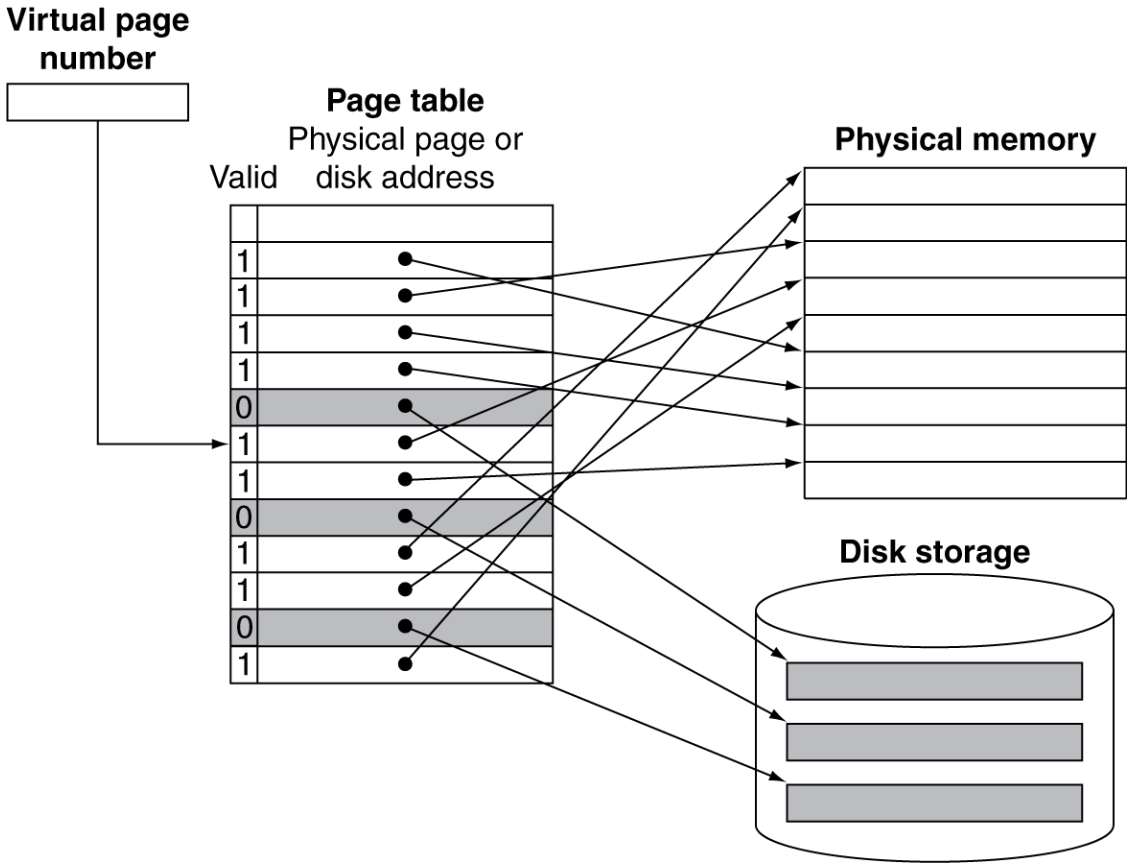
Page Table Register: Register in der CPU, das die Adresse der Page Table im Hauptspeicher enthält

Virtual Page Number: Teil der virtuellen Adresse, über den ein Eintrag in der Page Table indiziert wird

Page Table: Tabelle beinhaltet Eintrag (PTE) für jede Virtual Page Number

Page Table Entry (PTE): umfasst Valid-Flag (Seite im Hauptspeicher vorhanden) und im positiven Fall die physikalische Adresse der Seite (Physical Page Number)

Auswahl im Hauptspeicher gehaltener Seiten



Replacement-Strategie: einfache LRU Implementierung über Reference-Bit in PTE

- bei Nutzung auf "1"
- regelmäßig auf "0" zurückgesetzt
- Reference-Bit="0" kann ersetzt werden

Page Table Entry (PTE): umfasst Valid-Flag (Seite im Hauptspeicher vorhanden) und im positiven Fall die physikalische Adresse. Ansonsten optional die Adresse der Seite im Swap-Space auf der Festplatte

Daten auf Festplatte schreiben: Schreiben kostet Millionen von Takten. Schreiben einzelner Daten ist zu teuer, daher wird das Write-Back-Verfahren angewandt, um selten ganze Blöcke zu speichern. Dirty-Bit kennzeichnet veränderte Pages.

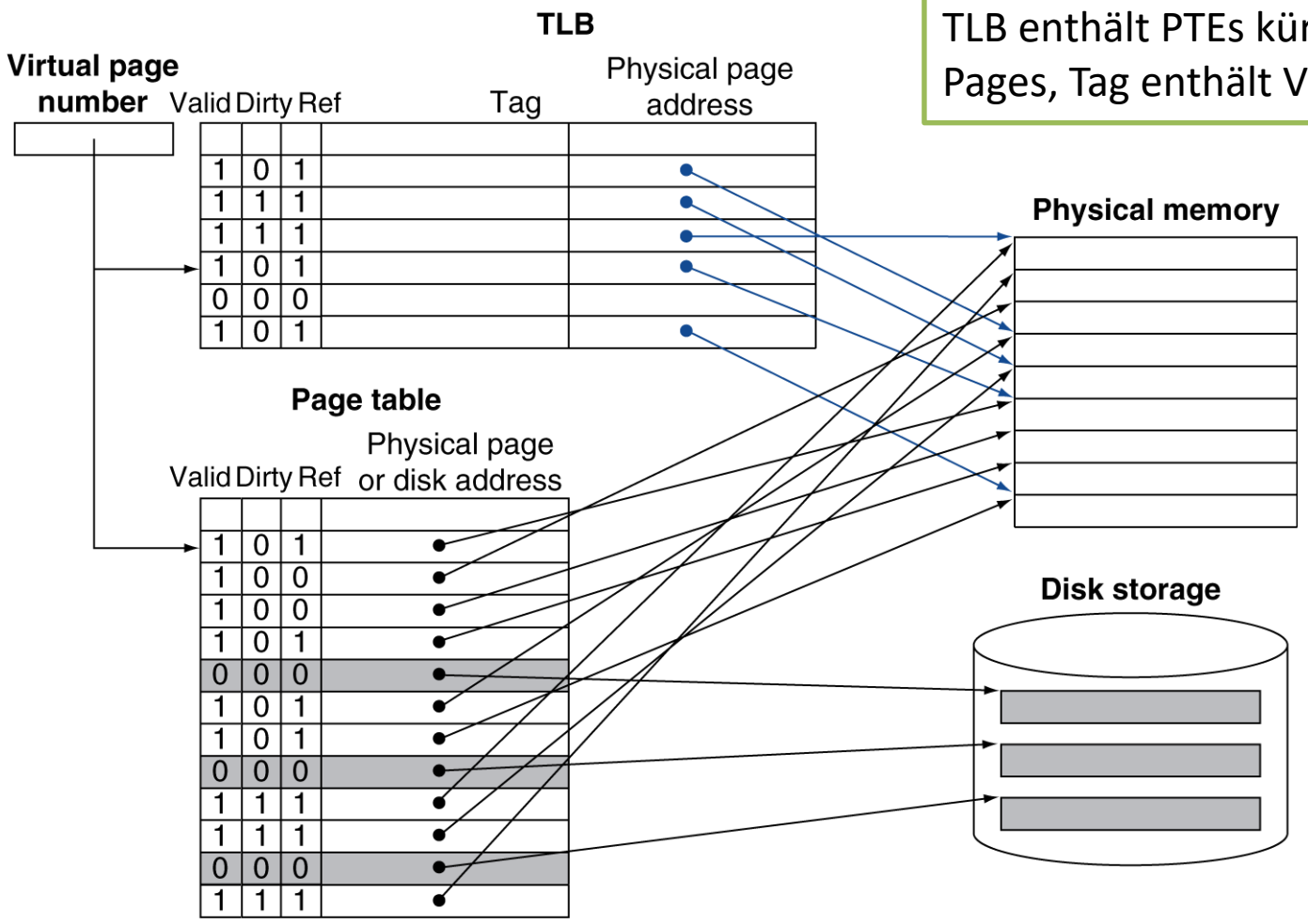
Page Table

- Page Table enthält einen Eintrag für jede virtuelle Adresse
  - Anzahl Einträge und damit auch Speicherbedarf abhängig von virtuellem Speicher und Seitengröße
    - virtueller Speicher 4GB, physikalischer Speicher 1GB, Seitengröße 4kB
      - eine Million PTEs
      - 19 Bit pro PTE, 18 Bit für physikalische Seite, 1 Bit für Valid-Bit
      - Page Table benötigt bis zu 3 MB
  - jeder Prozess hält eine eigene Page Table, deren Größe vom virtuellen Speicher des Prozesses abhängt
- Verfahren zur Reduktion der Größe der im Speicher gehaltenen Page Tables
  - dynamische Allokation von virtuellem Speicher d.h. von PTEs
  - mehrstufige Page Tables
    - Aufteilung einer 20-Bit-Adresse in zwei 10-Bit-Adressen
    - zur Auflösung der Adresse werden zwei Page Tables mit je 1024 Einträgen benötigt
      - erste Tabelle indiziert die „richtige“ zweite Tabelle
      - zweite Tabelle löst dann die virtuelle Adresse auf
    - nicht alle zweiten Tabellen müssen im Hauptspeicher gehalten werden

Schnelle Adress-Auflösung via Translation Lookaside Buffer (TLB)

- Auflösen der virtuellen Adresse erfordert zusätzlichen Speicherzugriff auf PTE
  - schneller Speicherzugriff durch Cachen der PTE im TLB (Translation Lookaside Buffer)
  - effektiv aufgrund guter Lokalitätseigenschaften der Page Table

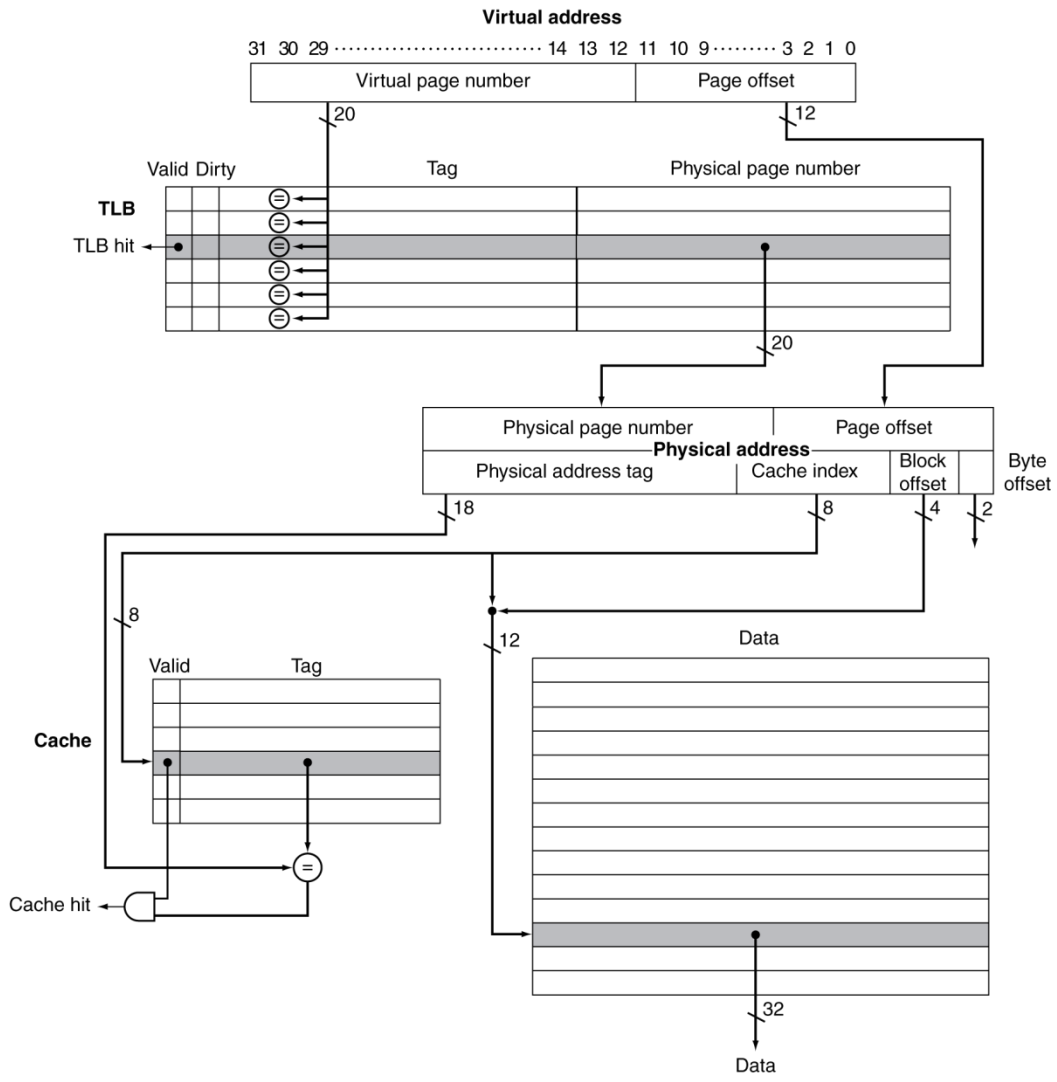
TLB enthält PTEs kürzlich aufgelöster Pages, Tag enthält Virtual Page Nummer



Kennzahlen TLB:

- 16–512 PTEs
- 0.5–1 Takte bei Hit
- 10–100 Takte bei Miss
- 0.01%–1% Miss-Rate
- Alternativen:
  - klein und voll-assoziativ
  - groß und wenig assoziativ

Zusammenspiel: Virtueller Speicher und Cache



Zwei Möglichkeiten für Zugriff auf Cache:

- Tag der virtuellen Adresse
  - verschiedene virtuelle Adressen für geteilten physikalischen Speicher
- Tag der physikalischen Adresse (abgebildet)
  - Auflösung der virtuellen Adresse notwendig für Zugriff auf Cache

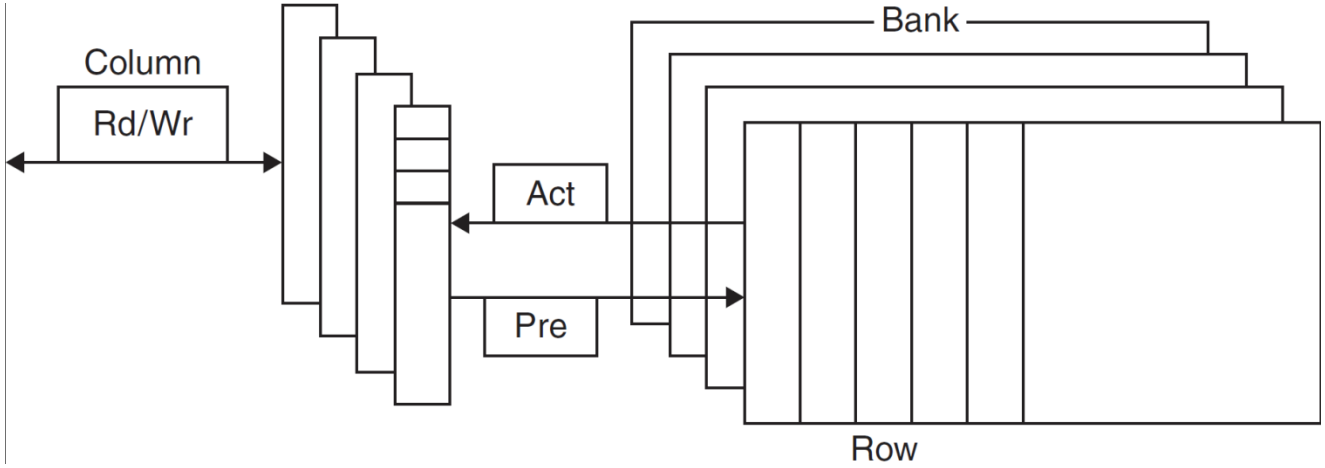
Kapitel 5 - Gliederung

Arten von Speicher

- Static RAM (SRAM)
  - direkte Zugriff auf Speicher in einem Takt,
  - realisiert über ICs (Integrated Circuits) in der CPU
  - 0.5ns – 2.5ns, $2000 – $5000 pro GB
- Dynamic RAM (DRAM)
  - Zugriff über Bus
  - 50ns – 70ns, $20 – $75 pro GB
- Festplatte
  - 5ms – 20ms, $0.20 – $2 pro GB
- Idealer Speicher
  - Zugriffszeit von SRAM
  - Kapazität und Kosten einer Festplatte

DRAM

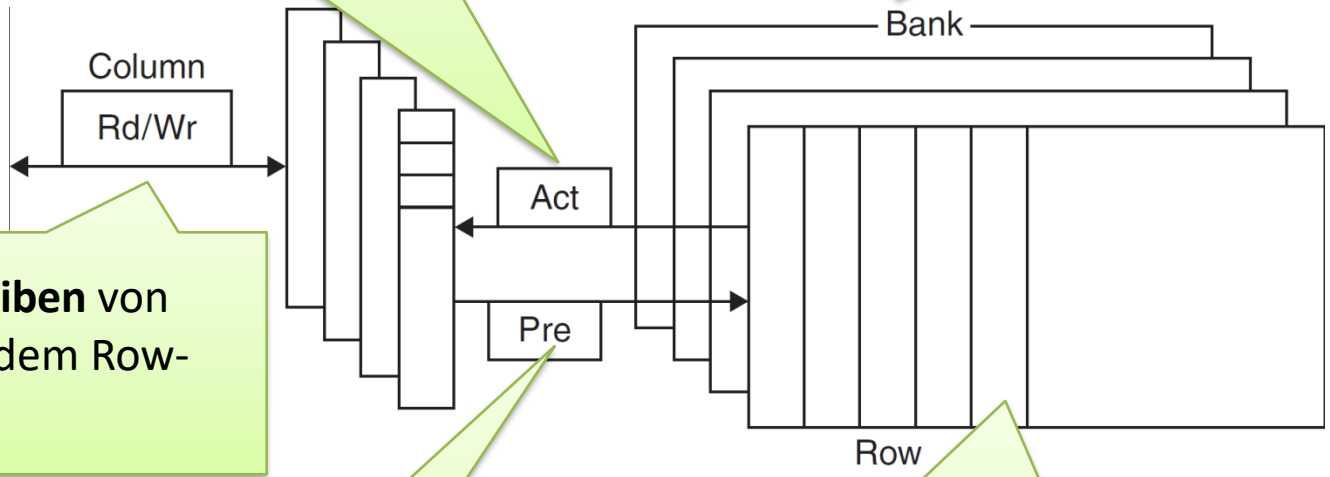
- Bit wird als Ladung in einem Kondensator gespeichert
  - Zustand des Kondensators: geladen/entladen
  - Zugriff auf Bit (Ladung) über eigenen Transistor
  - Daten müssen regelmäßig aufbereitet werden
    - Inhalt lesen und zurückschreiben
    - wird “reihen-weise” durchgeführt (row)
- Aufbau und Operationsprinzip von DRAM



Aufbau und Operationsprinzip von DRAM

- DRAM in **Speicherbänken** angeordnet
- mehrere Speicherbänke erlauben Daten „parallel“ zu lesen

Act: Lesen des Inhalts einer Row der aktiven Bank in den Row-Buffer

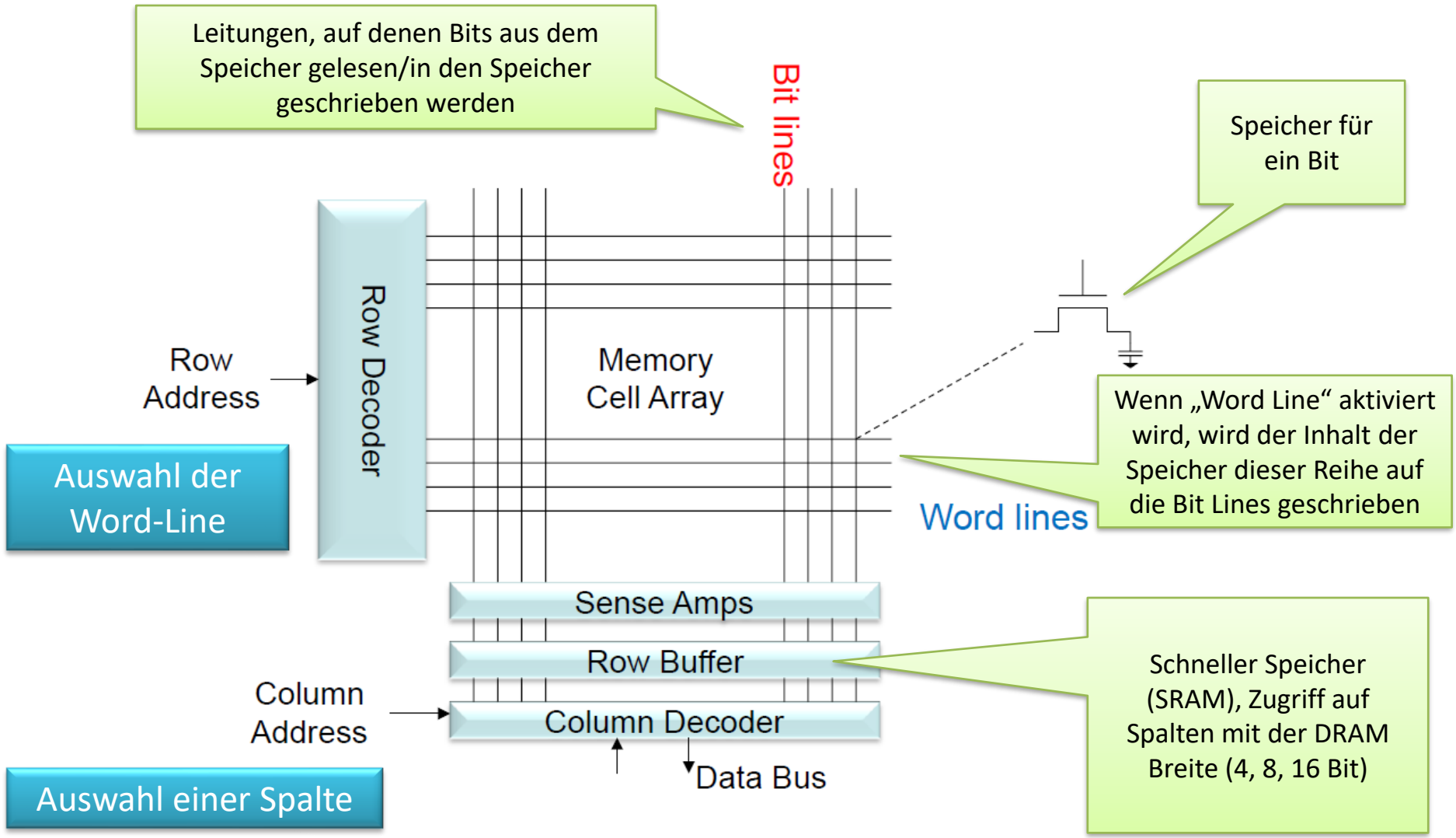


Lesen/Schreiben von Spalten aus dem Row-Buffer

Precharge: Öffnen/Schliessen einer Bank

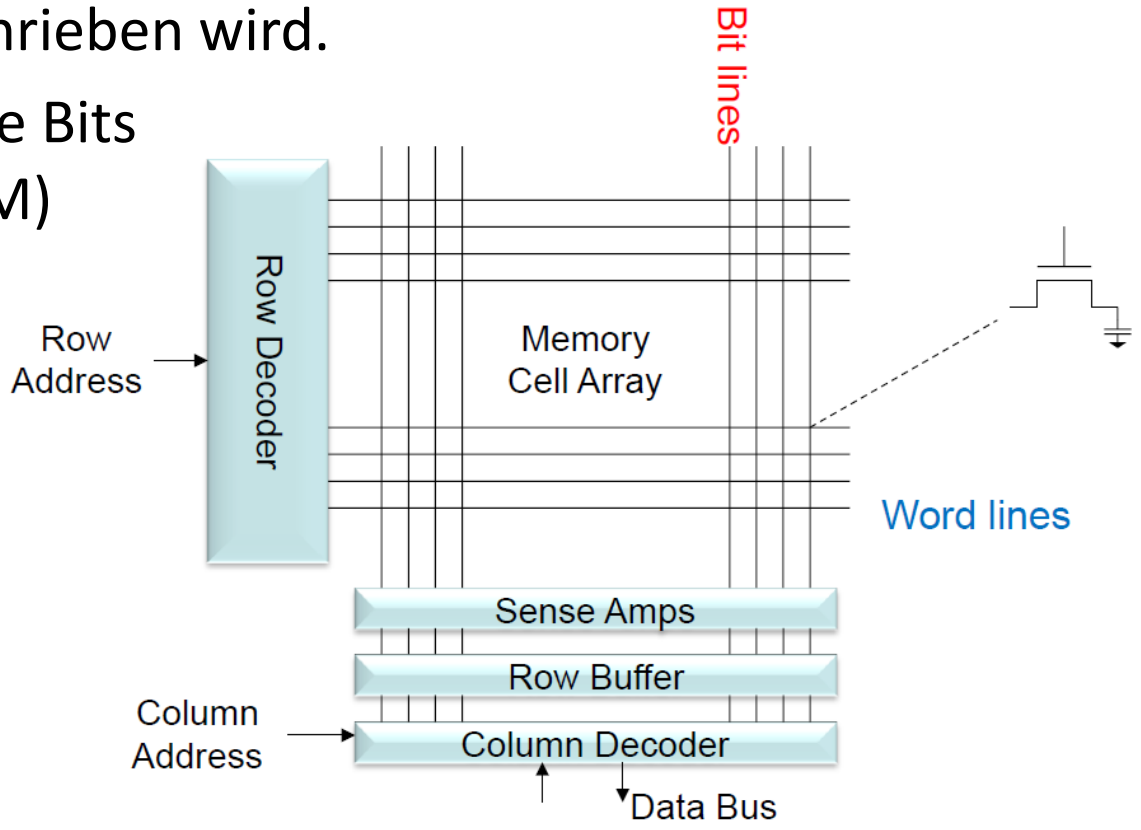
- Speicherbänke bestehen aus **Rows** (Reihen)
- Rows umfassen mehreren **Pages** (Seiten), die in **Spalten** angeordnet sind

DRAM – Prinzip der Rows



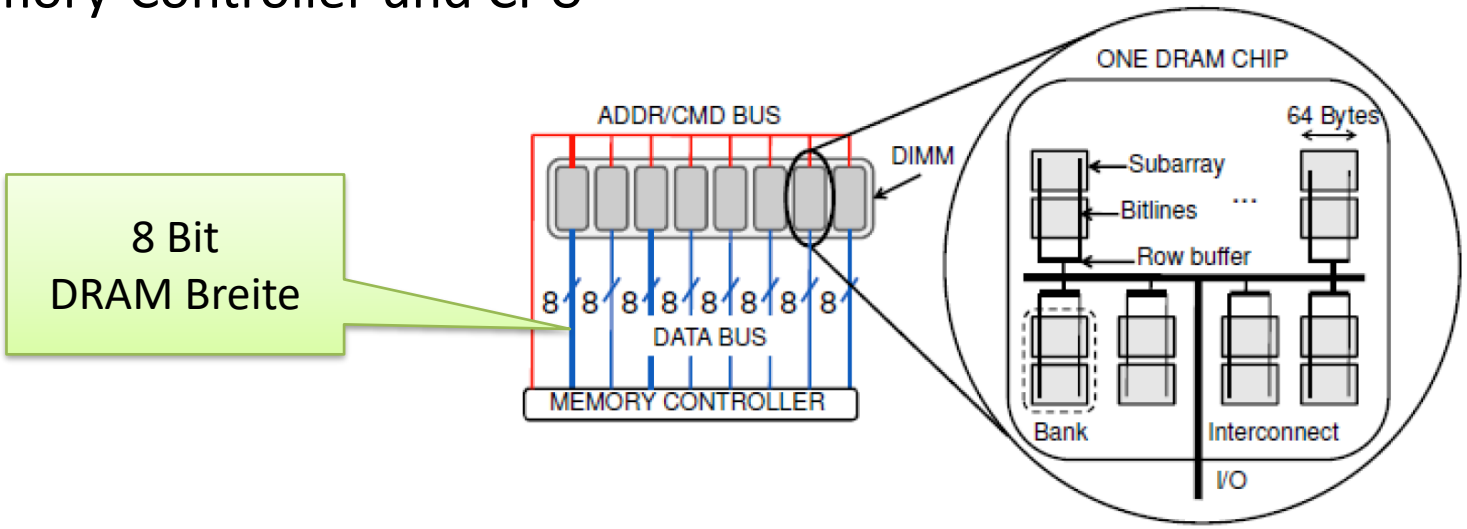
DRAM – Prinzip der Rows

- Speicher liegt als Gitter vor, an jedem Kreuzungsbit liegt ein Bit.
- Der Row Decoder funktioniert wie ein MUX und spezifiziert die Reihe, die gelesen wird, so dass auf der Word Line Spannung liegt.
- Spannung auf der Word Line verursacht, dass das entsprechende Bit auf die Bit Line beschrieben wird.
- Am Taktende werden die Bits in den Row Buffer (SRAM) geschrieben.
- Aus dem Row Buffer werden dann über den Column Decoder die entsprechenden Bits gelesen.

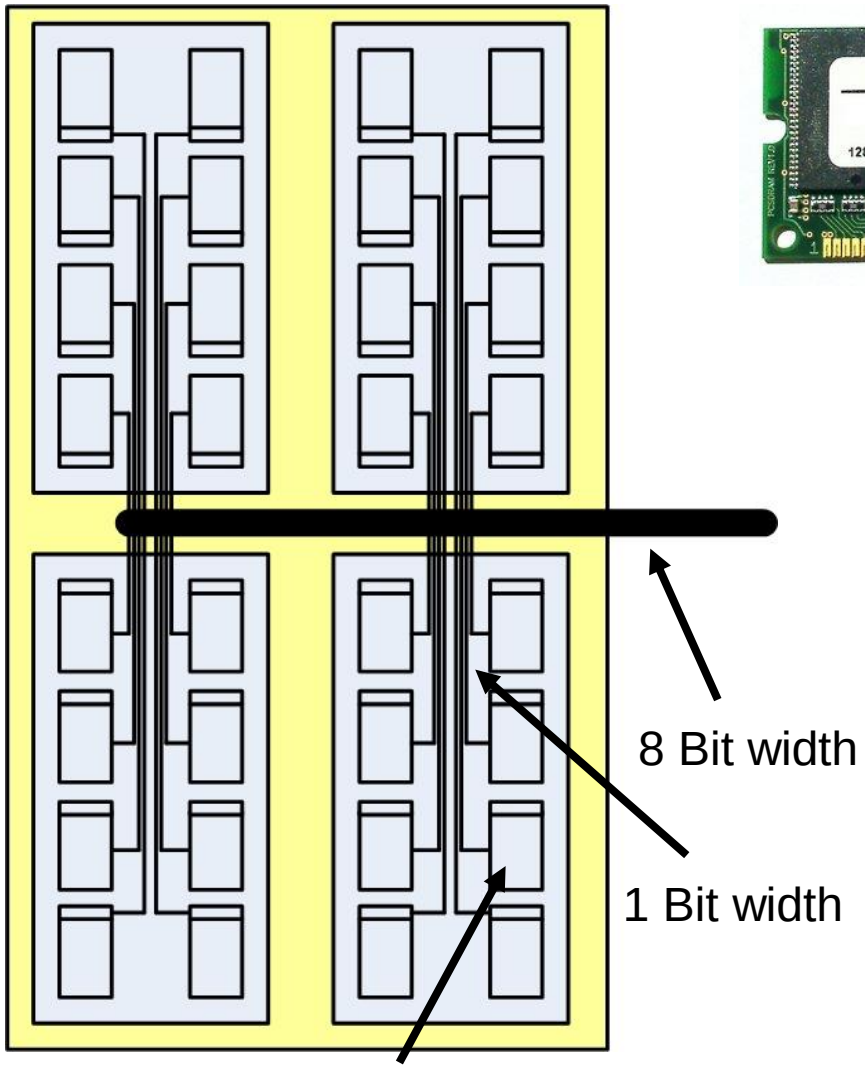


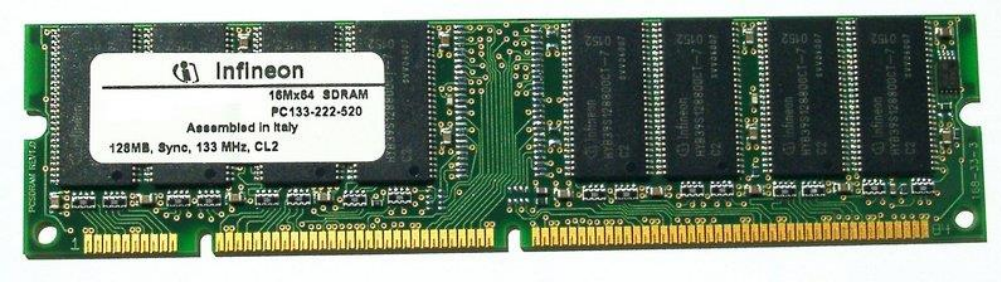
DIMM (Dual Inline Memory Module)

- DIMMs sind die Speichermodule, die in PCs verbaut sind.
- DIMMs bestehen aus mehreren DRAM Chips, die wiederum aus Sub Arrays bestehen.
- Über den Datenbus können 64 Bit gelesen werden, 8 Bits pro DRAM.
- Diese 8 Bits pro DRAM werden durch die Anordnung in Sub-Arrays parallel gelesen.
- Gruppierung mehrere DRAMs zu einem DIMM mit Zugriff über einen Speicher-Controller ermöglicht hohe Datentransferraten zwischen Memory-Controller und CPU



SDRAM (Synchronous DRAM) Struktur





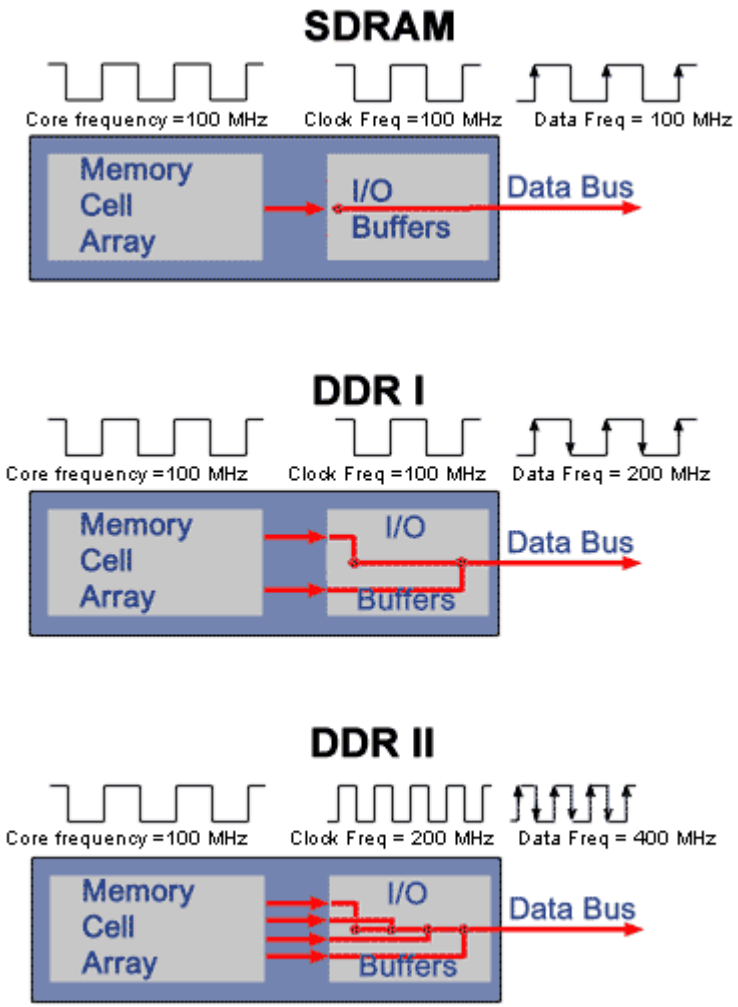
- 1GB DIMM besteht aus 8 SDRAM Chips zu je 1 Gbit
- 1Gbit SDRAM Chip besteht aus 4 256Mbit-Bänken mit je 8 32Mbit-Arrays
- Array besteht aus 16K Reihen und 2K Spalten mit eigenem Row-Buffer
- Spaltenbreite ist 1 Bit
- Breite des SDRAM-Chips sind 8 Bit
- Breite des DIMMs sind 64 Bit
- Bei einer Taktung von 100MHz ergibt sich eine SDRAM Bandbreite von 800Mbps und eine DIMM Bandbreite von 800MByte/s

Kapitel 5 - Gliederung

DRAM Entwicklung

- EDO (Extended Data Output) RAM, 1995:
  - asynchron, explizite Control-Signale zwischen CPU und Speicher
- SDRAM (Synchronous DRAM), 1997:
  - synchrone Speicherzugriffe erfolgen durch CPU-Clock als Taktgeber
  - aufeinanderfolgendes Lesen mehrere Blöcke (Spalten) aus einer Row ohne Änderung der Spaltenadresse
    - Änderungen von Zeilen- und Spaltenadresse dauern mehrere Takte und sind daher zu vermeiden
- DDR (Double Data Rate) SDRAM, 2000:
  - Daten werden bei steigender und fallender Flanke also mit doppelter Frequenz der Clock übertragen
  - Prefetching: Übertragung mehrerer (zweier) Bits pro Takt vom Speicher in den I/O Buffer
- DDR2/DDR3/DDR4 SDRAM, 2004/2007/2012:
  - weitere Steigerung der externen Taktung des Speicherbuses, Steigerung der internen Taktung durch Prefetching und Scheduling

Taktung von DRAM Core, Speicher-Controller und Speicherbus

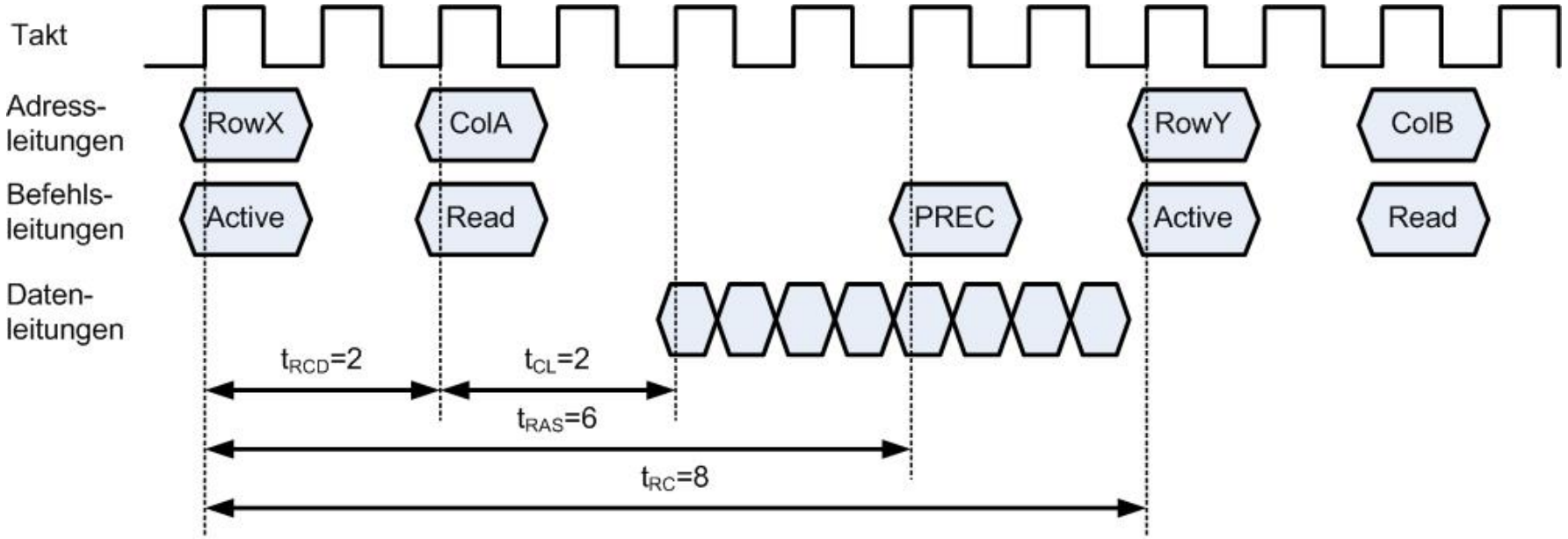


SDRAM:
DRAM Core, I/O Buffer und Datenbus arbeiten mit einer Geschwindigkeit

DDR 1:
Verdopplung der Geschwindigkeit des Datenbuses durch Senden von Daten bei steigender und fallender Flanke (zweimal pro Takt)

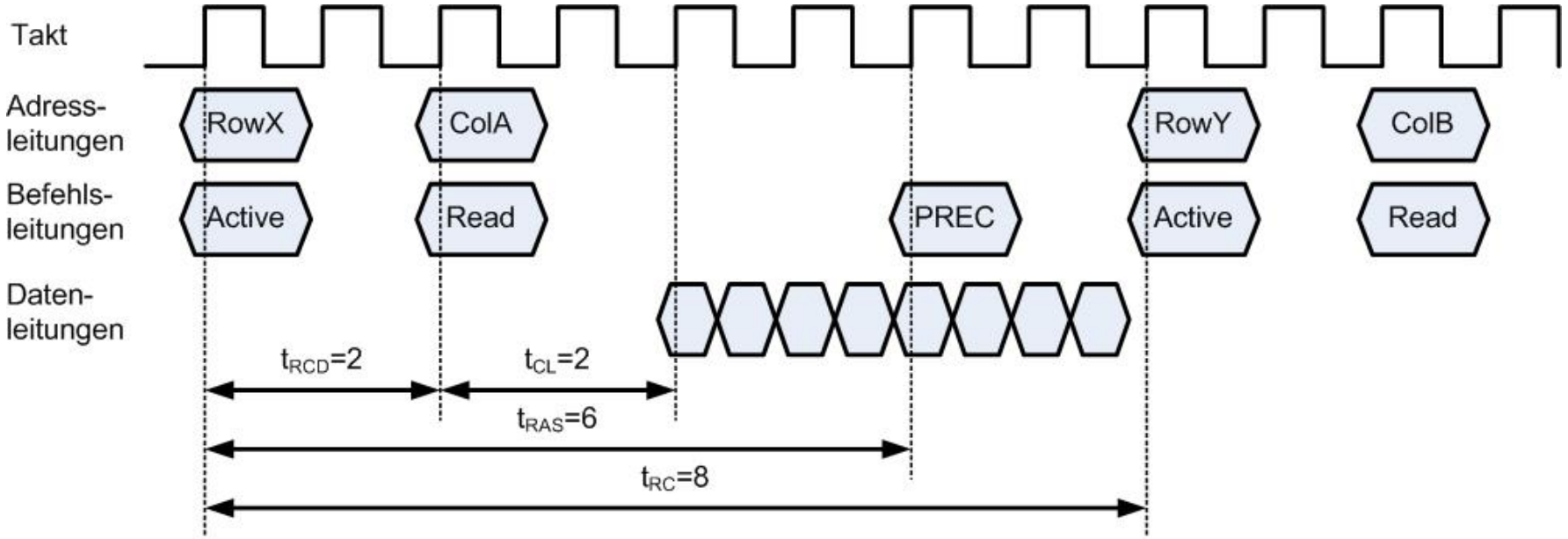
DDR 2:
Erhöhung der Taktung von Bus und Speicher-Controller, paralleler Zugriff (Pre-Fetching) auf Daten im Speicher

Zugriff auf DDR SDRAM



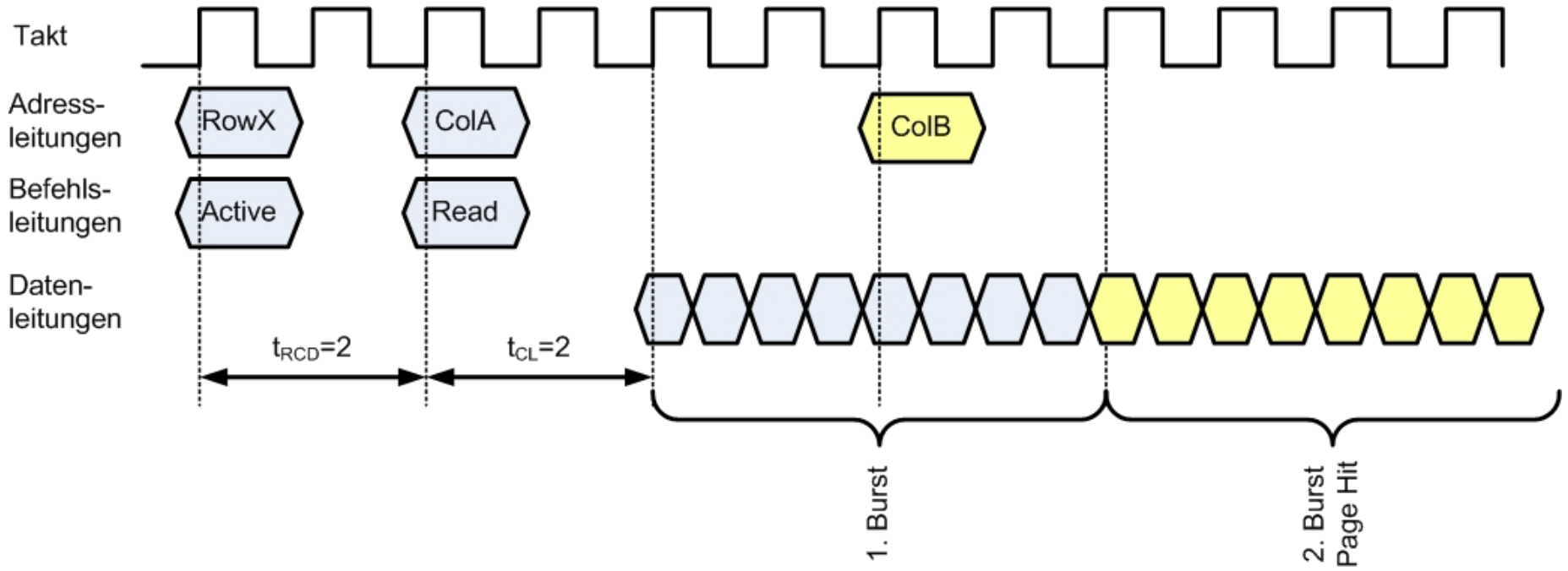
- Steuerung des Speichers über Adress- und Befehlsleitungen
- Lesen einer Zeile in den Row-Buffer
  - RowX: Adresse der Zeile
  - Befehl: Active
- Lesen einer Spalte aus dem Row-Buffer
  - ColA: Adresse der Spalte
  - Befehl: Read

Zugriff auf DDR SDRAM



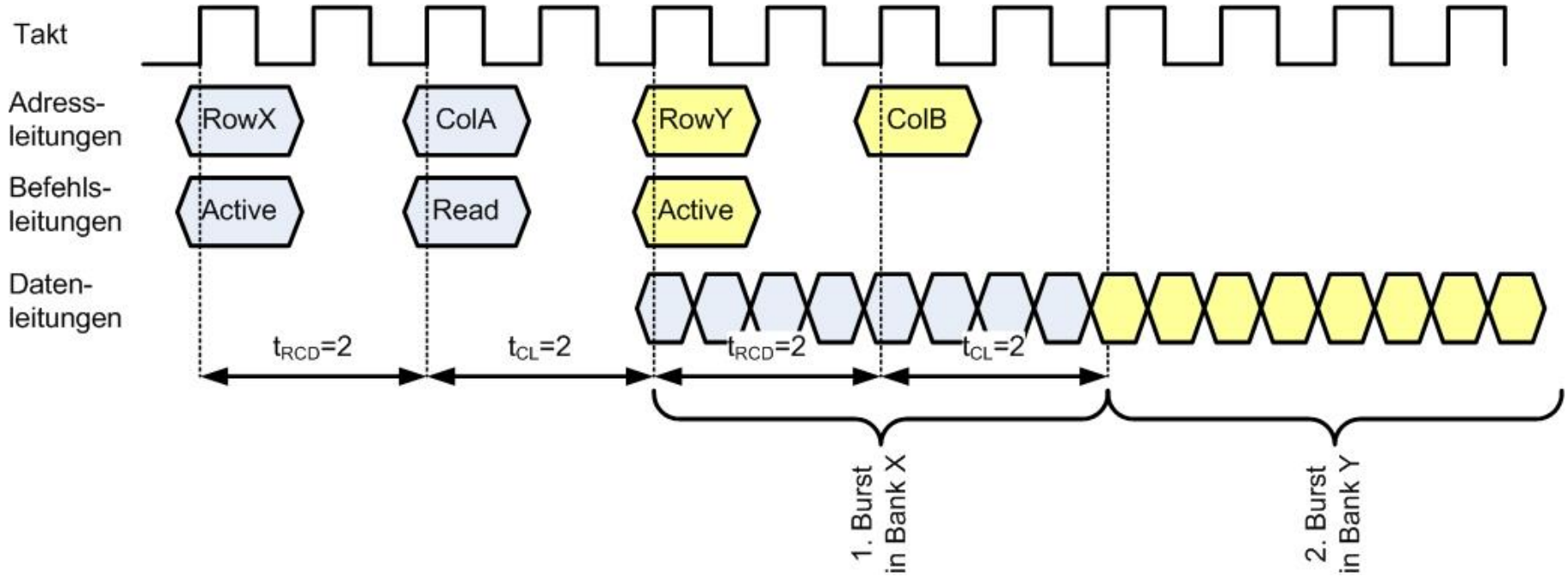
- Spalte mit 8 Bit wird gelesen (Burst Mode)
- Zeiten zum Umschalten:
  - RAS-to-CAS-Delay (t_{RCD}): Zeit zwischen Auswahl der Reihe und Spalte
  - CAS latency (t_{CL}): Zeit vom Auswählen der Spalte bis Lesebeginn
  - RAS-to-Precharge-Latency (t_{RAS}): Zeit bis zum Schließen einer Reihe
  - Read-Cycle-time (t_{RC}): Dauer eines Lesezyklus
- 100% Overhead bei Burst von 8 Bit
- PREC: Precharge, Schließen einer Reihe

Zugriff auf DDR SDRAM: Page Hit



- Page:
  - verfügbare Daten im Row-Buffer
  - Page-Hit: weitere Spalte der gleichen Zeile wird gelesen
  - nur Änderung der Spalte, nicht der Zeile
- Overhead bei 2 Spalten: 50%
- Overhead bei k Spalten: 100%/k

Zugriff auf DDR SDRAM: mehrere Bänke



- jede Bank verfügt über einen eigenen Row-Buffer und eigene Steuerleitungen
  - Neue Reihe einer Bank kann in den Row-Buffer geladen werden, während Daten aus dem Row-Buffer einer anderen Bank gelesen werden
  - bei geschicktem Scheduling kein Overhead
  - Interleaving: aufeinanderfolgende Datenblöcke in verschiedenen Bänken
- Overhead: auch bei Wechsel der Zeile praktisch eliminiert, falls die Burst-Länge groß genug ist

Kapitel 5 - Gliederung

Entwicklung der Speicherzugriffszeiten

Production year	Chip size	DRAM Type	Row access strobe (RAS)		Column access strobe (CAS)/ data transfer time (ns)	Cycle time (ns)
			Slowest DRAM (ns)	Fastest DRAM (ns)		
1980	64K bit	DRAM	180	150	75	250
1983	256K bit	DRAM	150	120	50	220
1986	1M bit	DRAM	120	100	25	190
1989	4M bit	DRAM	100	80	20	165
1992	16M bit	DRAM	80	60	15	120
1996	64M bit	SDRAM	70	50	12	110
1998	128M bit	SDRAM	70	50	10	100
2000	256M bit	DDR1	65	45	7	90
2002	512M bit	DDR1	60	40	5	80
2004	1G bit	DDR2	55	35	5	70
2006	2G bit	DDR2	50	30	2.5	60
2010	4G bit	DDR3	36	28	1	37
2012	8G bit	DDR3	30	24	0.5	31

Entwicklung der Speichermodule

Standard	Clock rate (MHz)	M transfers per second	DRAM name	MB/sec /DIMM	DIMM name
DDR	133	266	DDR266	2128	PC2100
DDR	150	300	DDR300	2400	PC2400
DDR	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10,664	PC10700
DDR3	800	1600	DDR3-1600	12,800	PC12800
DDR4	1066–1600	2133–3200	DDR4-3200	17,056–25,600	PC25600

Faktor 2
(DDR)

Faktor 64
(Busbreite)
pro Takt müssen also
mindestens 8 Bits pro
DRAM geladen werden

Übersicht einiger Speichermodule

Speichermodul	Speicherchip	Timing	Bustakt	Zykluszeit	CL	t_{RCD}	t_{RP}	t_{RAS}	t_{RC}
PC100-333	PC100-333	3,0-3-3	100 MHz	10 ns	30 ns	30 ns	30 ns	50 ns	80 ns
PC133-222	PC133-222	2,0-2-2	133 MHz	7,5 ns	15 ns	15 ns	15 ns	45 ns	60 ns
PC2100-2022	DDR266 (Intel)	2,0-2-2	133 MHz	7,5 ns	15 ns	15 ns	15 ns	45 ns	60 ns
(PC3200-2022)	DDR400	2,0-2-2	200 MHz	5 ns	10 ns	10 ns	10 ns	40 ns	50 ns
PC2-3200-3-3-3	DDR2-400B	3-3-3 (-8)	200 MHz	5 ns	15 ns	15 ns	15 ns	40 ns	55 ns
PC2-6400-5-5-5	DDR2-800D	5-5-5 (-18)	400 MHz	2,5 ns	12,5 ns	12,5 ns	12,5 ns	45 ns	57,5 ns
PC2-8500-5-5-5	DDR2-1066D	5-5-5 (-24)	533 MHz	1,875 ns	9,375 ns	9,375 ns	9,375 ns	45 ns	54,375 ns
PC3-6400-6-6-6	DDR3-800E	6-6-6	400 MHz	2,5 ns	15 ns	15 ns	15 ns		
PC3-6400-5-5-5	DDR3-800D	5-5-5	400 MHz	2,5 ns	12,5 ns	12,5 ns	12,5 ns		
PC3-8500-6-6-6	DDR3-1066E	6-6-6 (-20)	533 MHz	1,875 ns	11,25 ns	11,25 ns	11,25 ns	37,5 ns	48,75 ns
PC3-12800-10-10-10	DDR3-1600J	10-10-10 (-28)	800 MHz	1,25 ns	12,5 ns	12,5 ns	12,5 ns	35 ns	47,5 ns
PC3-14900-10-10-10	DDR3-1866J	10-10-10 (-32)	933 MHz	1,071 ns	10,7 ns	10,7 ns	10,7 ns		
PC3-17000-13-13-13	DDR3-2133	13-13-13 (-36)	1.066 MHz	0,93 ns	12,16 ns	12,16 ns	12,16 ns		
PC3-17000-11-11-11	DDR3-2133	11-11-11 (-36)	1.066 MHz	0,93 ns	10,29 ns	10,29 ns	10,29 ns		
PC4-1866-13-13-13	DDR4-1866	13-13-13	933 MHz	1,07 ns	13,92 ns				
PC4-2133-15-15-15	DDR4-2133	15-15-15	1.066 MHz	0,93 ns	14,06 ns				
PC4-2400-?	DDR4-2400	?	1.200 MHz	0,83 ns	?				
PC4-3200-?	DDR4-3200	?	1.600 MHz	0,63 ns	?				

- Speichermodulname: PCxxx-yyy
  - xxx (bis DDR3): Busbandbreite in MBps=Bustakt*8B*2 (bei DDR)
    - PC2-6400: 400MHz*8B*2=6400MBps
  - yyy: Latenzen in Taktzyklen
    - PC2-6400-5-5-5:
      - Zykluszeit=1/Bustakt=2,5ns
      - CL, t_{RCD}, t_{RP} jeweils 5 Taktzyklen

Kapitel 5 - Gliederung

Kapitel 5: Die Speicherhierarchie

5.1 Einführung

5.2 Caching

5.3 Virtueller Speicher

5.4 DRAM – Dynamic Random Access Memory

5.5 Zusammenfassung

Zusammenfassung Speicherhierarchie

- Grundlegende Prinzipien treffen auf allen Ebenen der Speicherhierarchie zu
  - basierend auf der Grundidee des Cachens
- Grundlegende Verfahren auf jeder Ebene der Speicher-Hierarchie
  - Platzieren von Blöcken im Cache / Cache-Organisation
  - Finden eines Blocks im Cache
  - Ersetzen eines Blocks bei einem Miss
  - Strategie zum Schreiben eines Blocks

Platzieren von Blöcken im Cache

- Bestimmt durch den Grad an Assoziativität
  - Direct mapped (1-way-associative)
    - keine Auswahl bei der Platzierung
  - N-way-set-associative
    - N Möglichkeiten zur Auswahl innerhalb eines Sets
  - Fully-associative
    - freie Auswahl innerhalb des Caches
- Je größer die Assoziativität desto geringer die Miss-Rate
  - aber Assoziativität erhöht auch die Komplexität, die Kosten und nicht zuletzt die Zugriffszeit

Blöcke im Cache finden

- Feststellen, ob und wo sich die Daten zu einer Speicheradresse im Cache befinden
  - hängt von der Cache-Organisation ab

Assoziativität	Methode zur Lokalisierung des Cache-Eintrag	Anzahl Tag-Vergleiche
Direct mapped	Index	1
N-way-set-associative	Set-Index, dann Suche in allen Einträgen im Set	n
Fully-associative	Suche in allen Einträgen	#Einträge
	Alternative: Full Lookup Table	0

- Hardware Caches
  - Anzahl der Vergleiche reduzieren, um Kosten zu sparen
- Virtueller Speicher
  - Full Lookup Table (Page Table) ermöglicht vollständige Assoziativität
  - Vorteil durch geringe Miss-Rate

Replacement Strategie

- Möglichkeit der Auswahl des zu ersetzenden Blocks bei Assoziativität
  - Least recently used (LRU)
    - komplexe und teure Hardware bei hoher Assoziativität
  - Random
    - erreicht fast die Performance von LRU, aber wesentlich günstiger zu implementieren
- Virtueller Speicher
  - vereinfachtes LRU Verfahren mit Hardwareunterstützung
    - Markierung von referenzierten Seiten mit Bit
    - regelmäßiges Löschen dieses Bits
    - Seiten ohne gesetztes Bit werden zuerst ersetzt, die Auswahl erfolgt zufällig

Schreiben von Blöcken

- Write-through
  - Daten werden sowohl auf oberer als auch auf unterer Ebene geschrieben
  - einfaches Ersetzen von Blöcken
  - benötigt Schreib-Puffer um CPU-Stalling bei Schreibzugriffen zu vermeiden
- Write-back
  - Daten werden zunächst nur auf der oberen Ebene geschrieben
  - Schreiben der Daten auf unterer Ebene beim Ersetzen des Blocks
  - Zustand der Daten im Cache (Konsistenz mit Daten im Hauptspeicher) muss gehalten werden
  - Schreib-Puffer verringert Miss-Penalty, da nicht zusätzlich auf Rückschreiben des Blocks gewartet werden muss
- Virtueller Speicher
  - aufgrund der großen Schreibverzögerung auf Festplatten ist nur Write-back möglich

Cache Optimierung

Veränderung	Auswirkung auf Miss Rate	Negative Auswirkung
Größerer Cache	reduziert Capacity-Misses	kann Zugriffszeit vergrößern
Höherer Grad an Assoziativität	reduziert Conflict-Misses	kann Zugriffszeit vergrößern
Größere Blöcke	reduziert Compulsory-Misses	größere Miss-Penalty

- Compulsory-Miss: Miss bei erstem Zugriff auf einen Block
- Capacity-Miss: Miss durch beschränkte Cache-Kapazität, auf ersetzten Block wird noch einmal zugegriffen
- Conflict-Miss: Miss durch beschränkte Anzahl Plätze in einem Set, würde in einem fully-associtative Cache nicht auftreten

Multi-Level On-Chip Caches

Characteristic	ARM Cortex-A8	Intel Nehalem
L1 cache organization	Split instruction and data caches	Split instruction and data caches
L1 cache size	32 KiB each for instructions/data	32 KiB each for instructions/data per core
L1 cache associativity	4-way (I), 4-way (D) set associative	4-way (I), 8-way (D) set associative
L1 replacement	Random	Approximated LRU
L1 block size	64 bytes	64 bytes
L1 write policy	Write-back, Write-allocate(?)	Write-back, No-write-allocate
L1 hit time (load-use)	1 clock cycle	4 clock cycles, pipelined
L2 cache organization	Unified (instruction and data)	Unified (instruction and data) per core
L2 cache size	128 KiB to 1 MiB	256 KiB (0.25 MiB)
L2 cache associativity	8-way set associative	8-way set associative
L2 replacement	Random(?)	Approximated LRU
L2 block size	64 bytes	64 bytes
L2 write policy	Write-back, Write-allocate (?)	Write-back, Write-allocate
L2 hit time	11 clock cycles	10 clock cycles
L3 cache organization	–	Unified (instruction and data)
L3 cache size	–	8 MiB, shared
L3 cache associativity	–	16-way set associative
L3 replacement	–	Approximated LRU
L3 block size	–	64 bytes
L3 write policy	–	Write-back, Write-allocate
L3 hit time	–	35 clock cycles

2-Level TLB Organisation

Characteristic	ARM Cortex-A8	Intel Core i7
Virtual address	32 bits	48 bits
Physical address	32 bits	44 bits
Page size	Variable: 4, 16, 64 KiB, 1, 16 MiB	Variable: 4 KiB, 2/4 MiB
TLB organization	1 TLB for instructions and 1 TLB for data Both TLBs are fully associative, with 32 entries, round robin replacement TLB misses handled in hardware	1 TLB for instructions and 1 TLB for data per core Both L1 TLBs are four-way set associative, LRU replacement L1 I-TLB has 128 entries for small pages, 7 per thread for large pages L1 D-TLB has 64 entries for small pages, 32 for large pages The L2 TLB is four-way set associative, LRU replacement The L2 TLB has 512 entries TLB misses handled in hardware