Rechnerarchitekturen 1*

Speicher & Speicher-Hierarchie

Prof. Dr. Alexander Auch

*Teilweise entnommen aus “Mikrocomputercomputertechnik 1” von Prof.Dr-Ing. Ralf Stiehler, sowie Patterson & Hennessy

Ziele der Veranstaltung

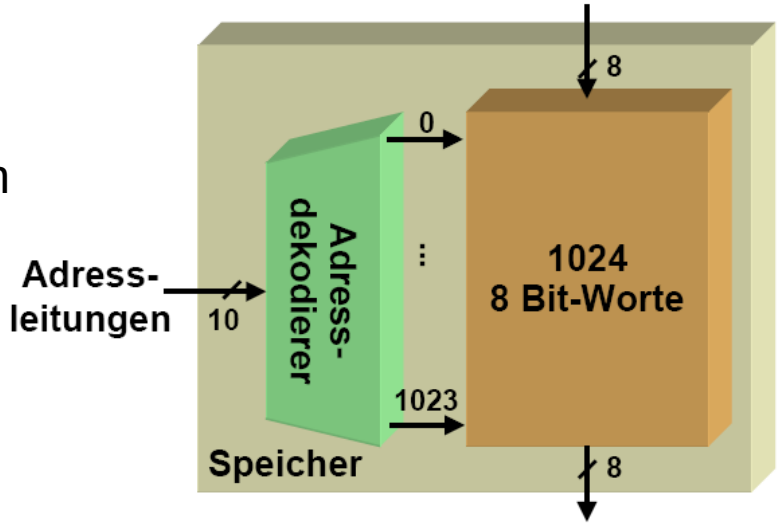
- **Rechnerentwurf**:
  - Prozessor, Speicher, Ein-/Ausgabe
  - Entwurfs- und Optimierungsmöglichkeiten
- **Prozessorentwurf**:
  - Befehlsverarbeitung
  - Entwurfs- und Optimierungsmöglichkeiten
- **Assemblerprogrammierung**:
  - im MIPS-Simulator MARS

Halbleiterspeicher

Grundlegende Begriffe

Speicherkapazität

ist das Produkt aus der Anzahl der adressierbaren Speichereinheiten (z.B. Bytes = 8-Bit-Worte) und der Wortbreite



Beispiel : Speicher mit 1kByte Speicher :

Die Speicherkapazität wird oft explizit als Produkt der Anzahl der Worte und der Wortbreite angegeben, um die Organisationsform zu kennzeichnen

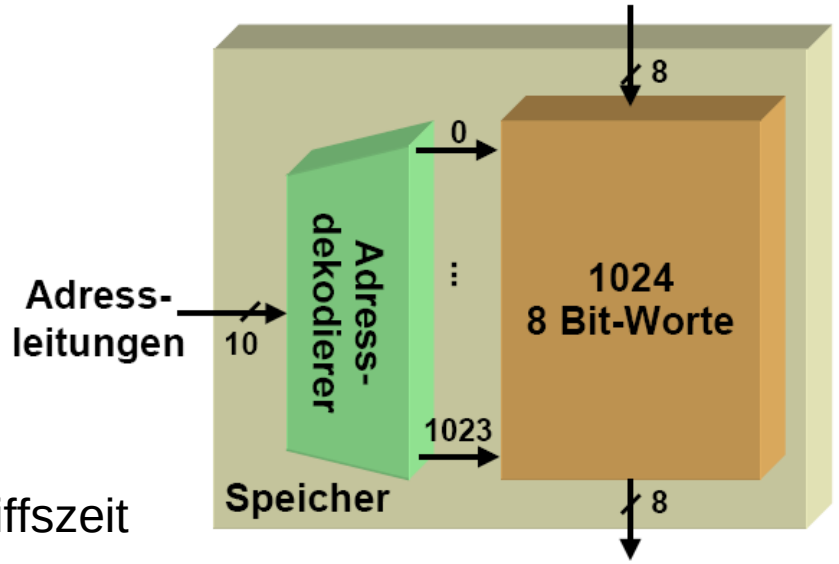
Beispiel:
256K * 4 ist ein Speicher mit 2^{18} 4-Bit Worten und einer Kapazität von 128 KByte

Grundlegende Begriffe

Datenrate oder Bandbreite (transfer rate, bandwidth)

=

Bitbreite des ausgelesenen Datenwortes/ Zugriffszeit



Beispiele :

SRAM-Speicher mit 256K*4, Zugriffszeit 12ns:
Datenrate = 4 Bit/12ns ≈ 4Bit * 83MHz = 41,5 MB/sec.

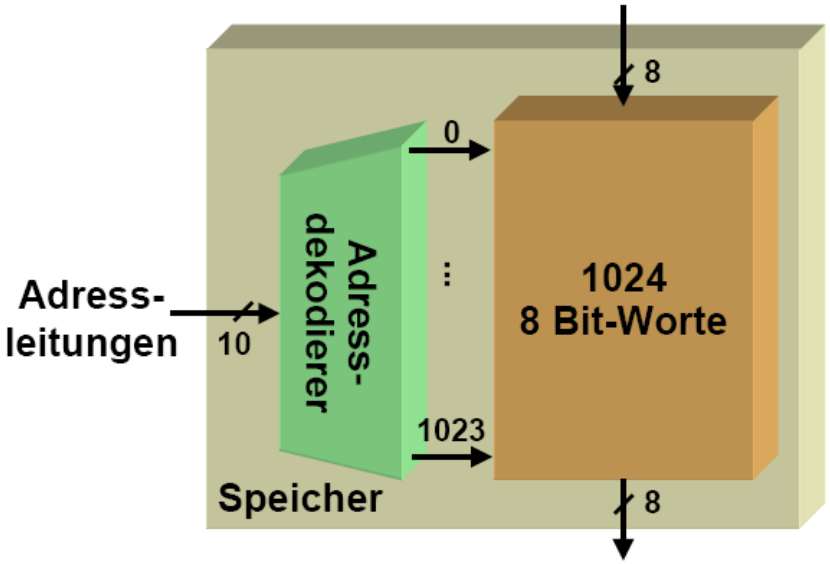
SRAM-Speicher mit 64K*16, Zugriffszeit 15ns:
Datenrate = 16 Bit/15ns ≈ 16Bit * 67MHz ≈ 134 MB/sec.

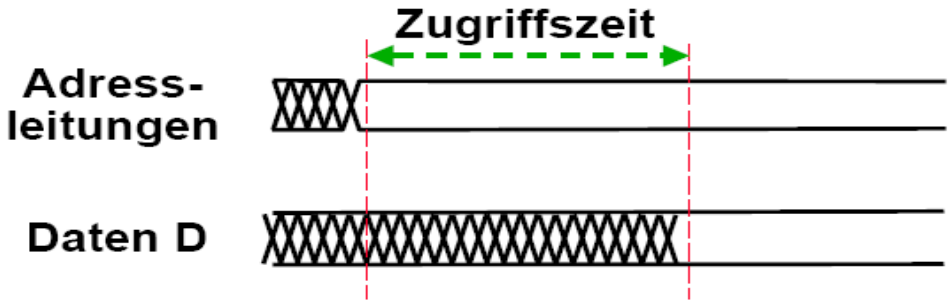
=> Sowohl Speicherorganisation als auch Zugriffszeit bestimmen die Bandbreite !

Grundlegende Begriffe

Speicherzugriffszeit (Access time)

ist die Zeit, die vom Anlegen der Adresse an den Adressleitungen bis zum Erscheinen der Daten an den Datenleitungen des Speichers vergeht.

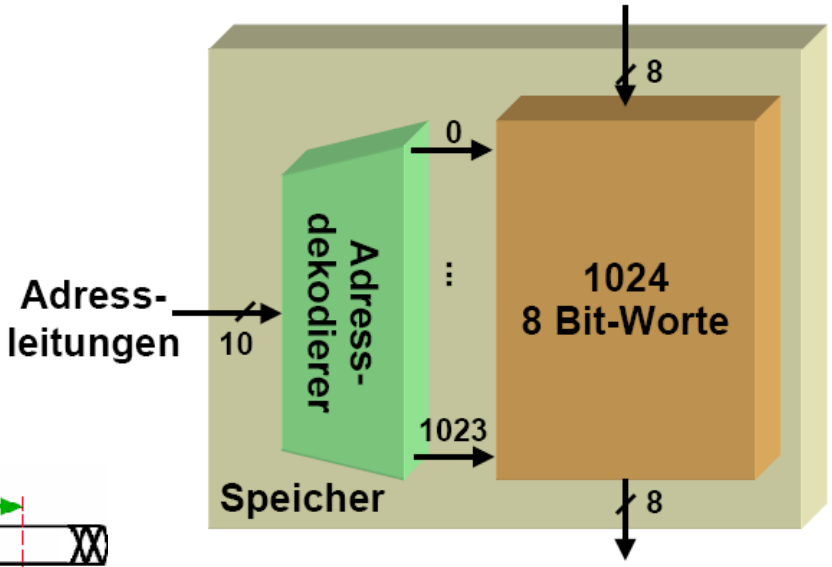


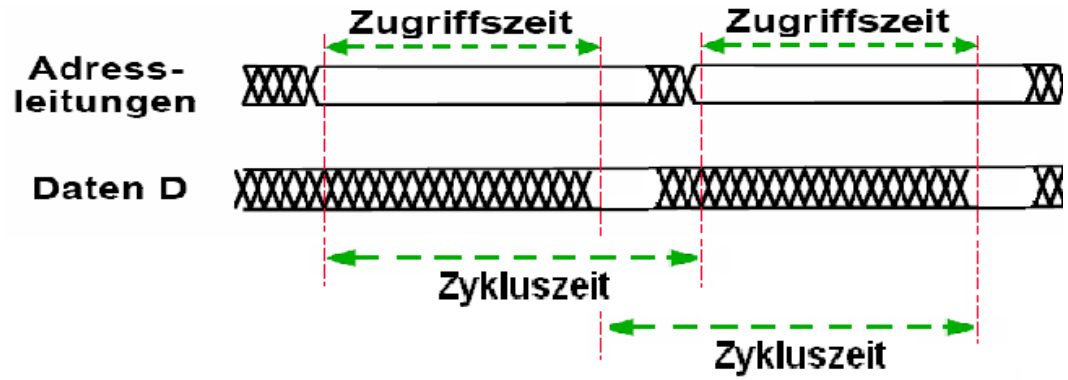


Grundlegende Begriffe

Speicherzykluszeit (cycle time)

ist die minimale Zeit, die zwischen 2 aufeinanderfolgenden Anlegen von Adressen an den Speicher vergeht.

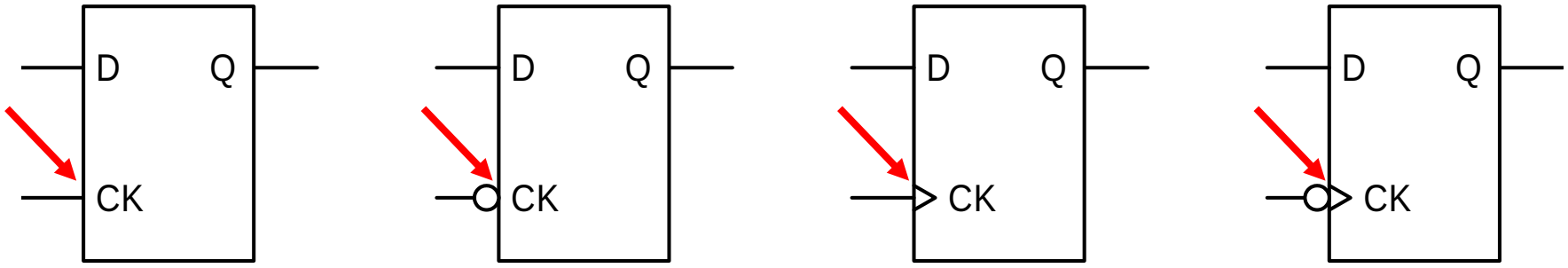




Idealerweise ist Zugriffszeit = Zykluszeit, in der Realität ist die Zykluszeit größer

Zykluszeiten für Lesen und Schreiben sind in der Regel unterschiedlich.

Eine einfache Speicherzelle : Flipflop



Man unterscheidet :

a) pegelgesteuert, schaltet bei CK=1

b) pegelgesteuert, schaltet bei CK=0

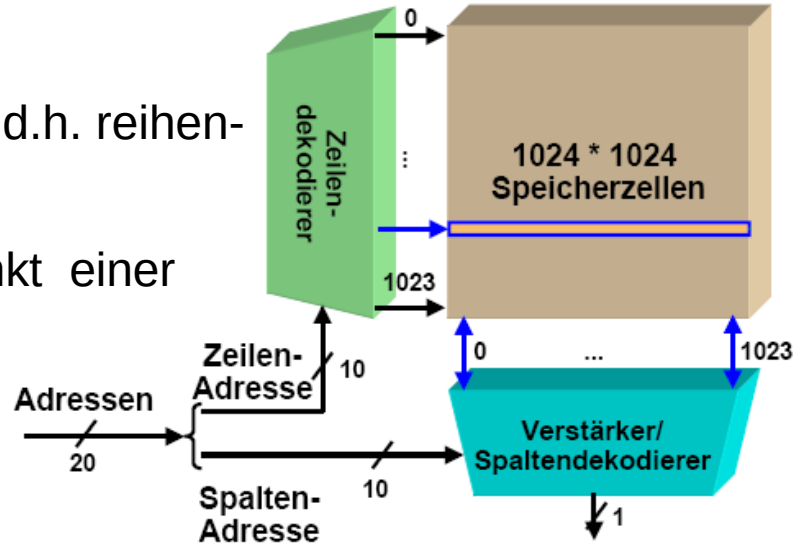
c) flankengesteuert, schaltet bei Flanke CK von 0 auf 1

d) flankengesteuert, schaltet bei Flanke CK von 1 auf 0

Speichermatrix

Die Speicherelemente sind matrizenartig , d.h. reihen- und spaltenweise angebracht.

Jedes Speicherelement liegt im Schnittpunkt einer Zeilen-Auswahlleitung und einer Spalten-Auswahlleitung.



Um die Anzahl der Auswahlleitungen (und der zugehörigen Ansteuerschaltungen) zu minimieren, versucht man, die Speichermatrix möglichst quadratisch aufzubauen.

Spalten-Auswahlleitungen (MSBs) nennt man Daten- oder Bitleitungen. Zeilen-Auswahlleitungen (LSBs) werden als Wortleitungen bezeichnet.

Addressierung der Speichermatrix

⇒ die Adressierung erfolgt über die Adresspins des Bausteins, die mit den entsprechenden Signalen des Adressbusses beschaltet werden.

Es sind oft zusätzliche speicherinterne Schnittstellen nötig,

⇒ damit man Treiberleistung zur Verfügung stellen kann, um die Belastung der Adressleitungen des Mikroprozessors klein zu halten.

⇒ um Pegelanpassungen (z.B. TTL-Kompatibilität) sicherzustellen

Bei DRAMs wird dem Baustein die Speicheradresse oft in mehreren Teilen sequentiell zugeführt, um die Anzahl der Anschlusspins klein zu halten. Dann müssen die Teiladressen in den Interfaces zwischengespeichert werden.

Synchrone Speicherbausteine

⇒ Speicherzugriffe werden per Taktsignal mit Vorgängen auf dem Bus synchronisiert

⇒ Adressen, Daten und Steuersignale werden in Registern zwischengespeichert und durch Triggerung des Bustakts (CLK) speicherintern zur Verfügung gestellt.

⇒ oft ist (De-)Aktivierung des CLK-Eingangs mit Clock Enable Signal möglich

Steuerlogik von Speicherbausteinen (1)

Die Steuerlogik ist die Schnittstelle zum Steuerbus des Prozessors

⇒ Die wichtigsten Signale sind ChipSelect (CS, CE), Read/Write (RW) und OutputEnable (OE)

Chip Select (CS) oder Chip Enable (CE)

⇒ CS wählt einen Baustein aus mehreren Speicherbausteinen im System aus
⇒ CS wird häufig durch einen Decoder aus den MSBs der Adresssignale erzeugt !
⇒ CS kann man auch zum Energiesparen durch Deaktivierung des Chips nutzen

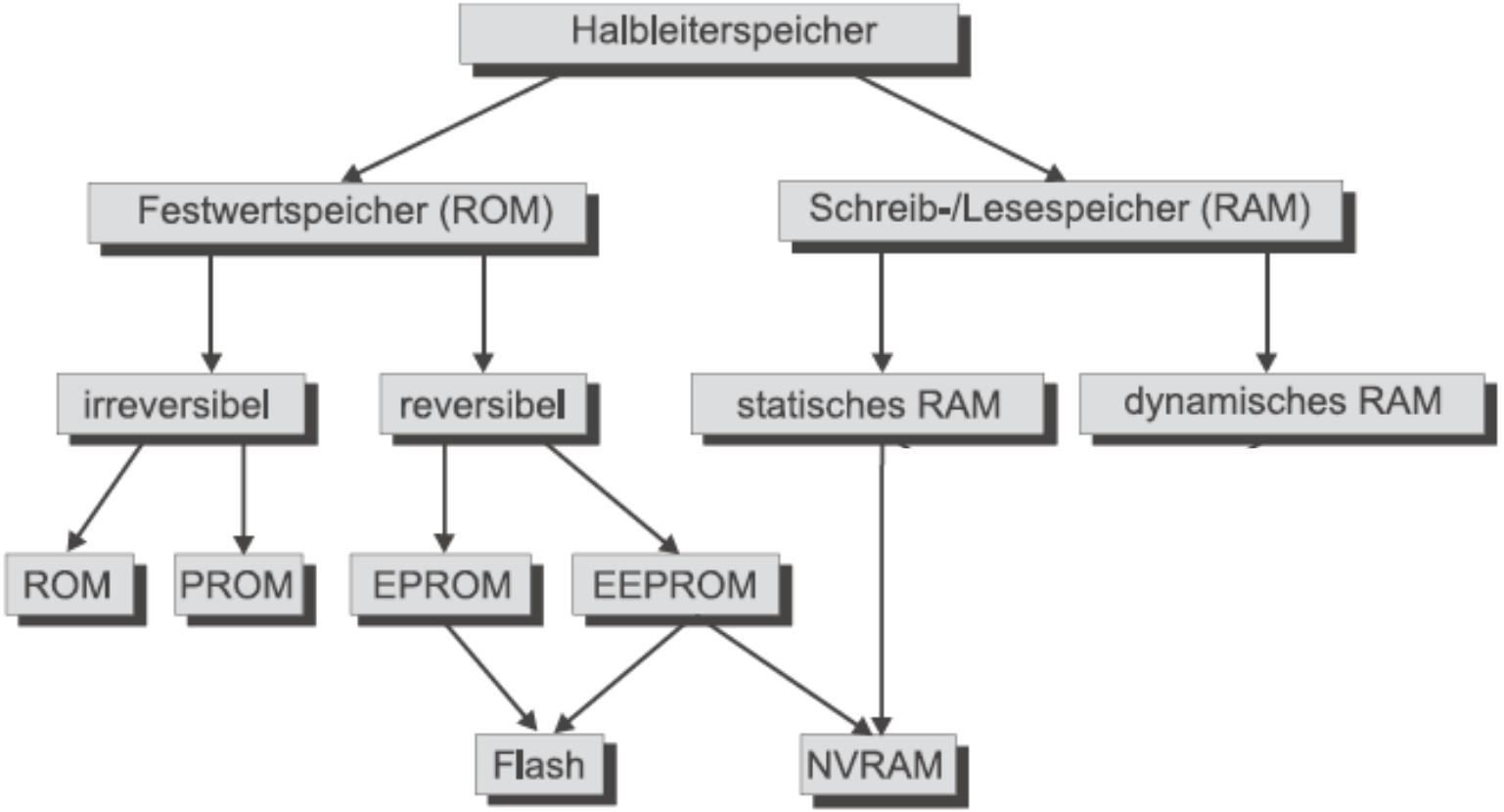
Read/Write (RW)

⇒ RW wird nur bei Speichertypen benötigt, deren Inhalte von System änderbar sind (RAM, EEPROM,…)
⇒ RW aktiviert Lese/ bzw. Schreibverstärker im Speicherbaustein
⇒ RW selektiert die Richtung der Treiber für die Datenbits
⇒ Flanke des RW-Signals kann als Clock-Signal dienen

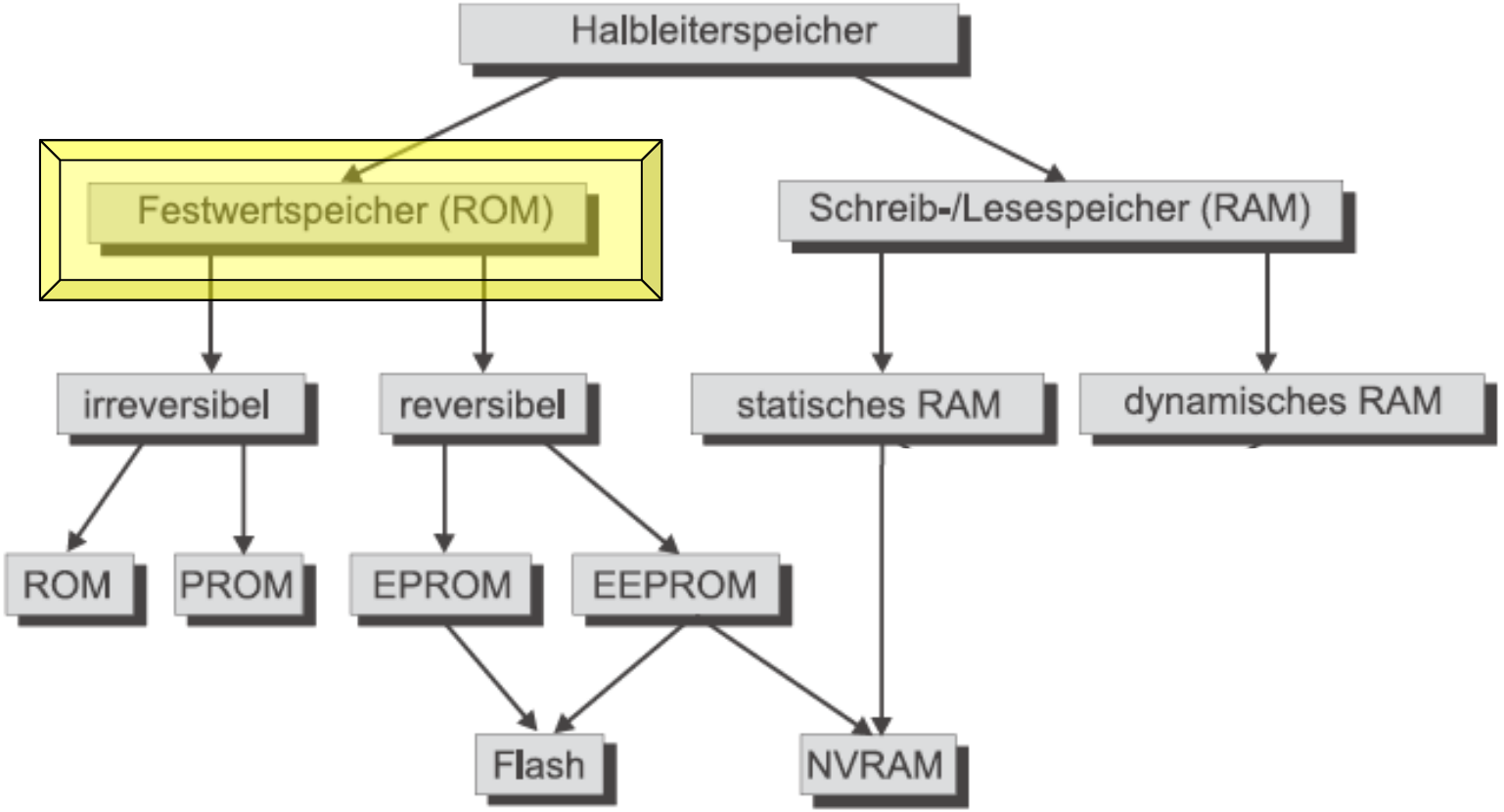
Output Enable (OE)

⇒ OE legt die ausgelesenen Speicherinhalte über 3-State-Buffer auf den Datenbus

Klassifizierung von Halbleiterspeichern



Mikrocomputertechnik – Halbleiterspeicher



Festwertspeicher

Inhalt kann vom Mikroprozessor während des normalen Betriebs nur gelesen, nicht aber verändert werden => **ROM**: *Read Only Memory*

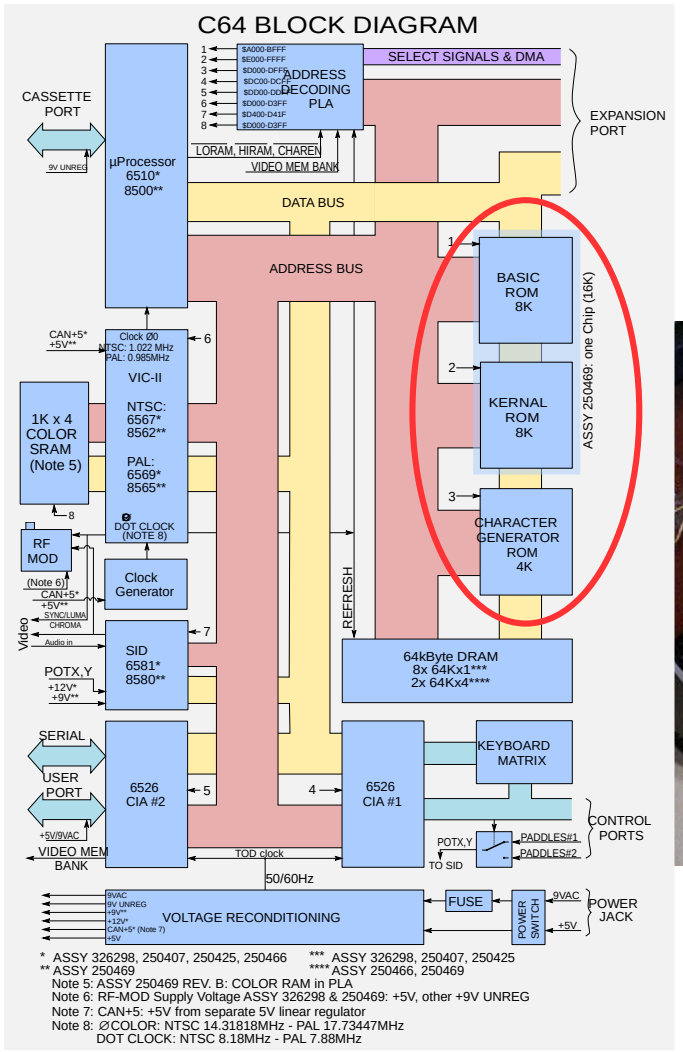
Festwertspeicherinhalte sind nicht flüchtig (*non volatile*), d.h. sie bleiben auch nach dem Ausschalten der Betriebsspannung erhalten.

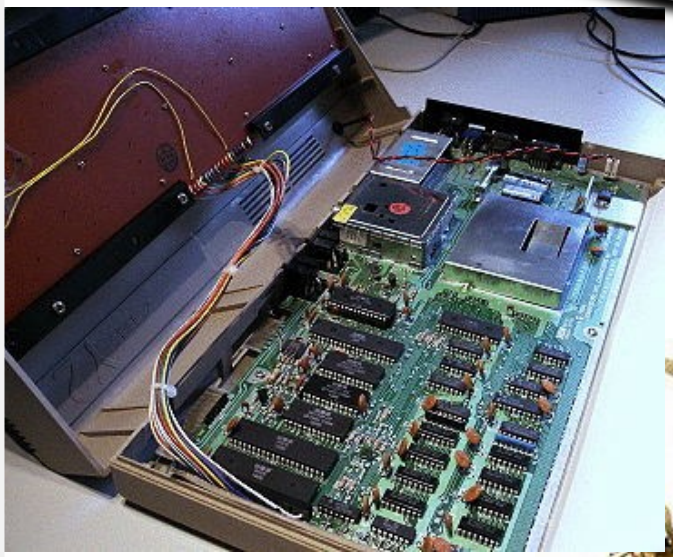
Festwertspeicher dienen hauptsächlich zur Aufnahme von Programmen und Daten, die dauernd und unverändert zur Verfügung stehen müssen.
⇒ Teile des Betriebssystems
⇒ Systemtabellen, Systemkonstanten

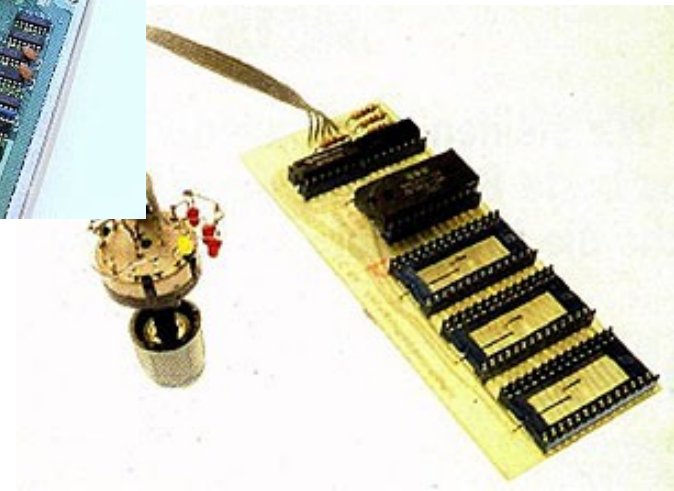
Die in ROMs abgelegte Software wird häufig auch als *Firmware* bezeichnet.

> Das Einschreiben der Information in ein ROM wird als Programmierung des Speicherbausteins bezeichnet.

Mikrocomputertechnik – Halbleiterspeicher







Festwertspeicher - Irreversible ROMs

Einprogrammierte Information kann auch außerhalb des normalen Einsatzes nicht mehr überschrieben werden, d.h. die Programmierung erfolgt stets außerhalb des Systems, in dem der Speicher eingesetzt wird.

Man unterscheidet 2 Typen:

MROM: *Maskenprogrammierte ROMs*,
⇒ Programmierung erfolgt bei der Herstellung des Bausteins
⇒ Hersteller muss Speicherinhalt vor der Produktion als Datei erhalten
⇒ erst ab großen Stückzahlen wirtschaftlich

PROM: *Programmable ROMs*
⇒ Programmierung mit Hilfe von speziellen Programmiergeräten durch Lieferanten oder vom Anwender selbst nach der Herstellung der Bausteine
⇒ für kleinere Stückzahlen
⇒ bei der Programmierung („Brennen“) werden bestimmte Bereiche in Speicherelementen physikalisch zerstört und damit irreversibel verändert

Mikrocomputertechnik – Halbleiterspeicher

Festwertspeicher - Reversible ROMs

Eingeschriebene Information kann geändert werden, dies ist in der Regel aber nicht während des normalen Betriebs und auch nicht durch den µP/µC des Systems möglich. Einsatz bei Prototypen und kleineren Stückzahlen

EPROM: *Erasable and Programmable ROM*

⇒ UV-löschbar

⇒ Baustein muss zum Löschen dem System entnommen, und für längere Zeit einer ultravioletten Strahlung ausgesetzt werden

⇒ Neue Programmierung vor dem Einsetzen ins System

EEPROM: *Electrically Erasable and Programmable ROM*

⇒ Löschen und Neuprogrammierung kann durch den µC selbst veranlasst werden

Unterschied zu RAM-Bausteinen

=> Löschen/Wiederbeschreiben dauert um viele Größenordngn. länger als Lesen

=> Anzahl der Schreibzyklen aus physikalischen Gründen begrenzt

=> Inhalt bleibt nach Abschalten der Betriebsspannung erhalten

Flash-Speicher

Größere Speicherbereiche können auf einmal (Blitzartig/Flash) gelöscht werden.

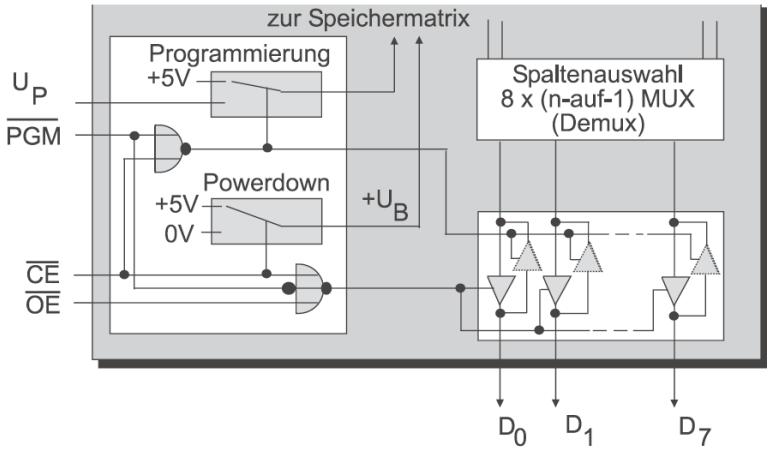
PROM - Programmierbares ROM

⇒ neben dem MROMs die beim Chiphhersteller programmiert werden, gibt es für kleinere Serien PROMs, die vom Benutzer programmiert werden können

⇒ Hierzu muss der PROM in ein spezielles Programmiergerät (Brenner) zum „Einbrennen“ der Daten eingesetzt werden.

⇒ Bausteine haben ein Zusatzsignal PGM, das zur Programmierung gesetzt wird

⇒ zusätzlicher Pin für höhere Spannung (typischerweise 12,5V anstelle 5V) während des Programmiervorgangs

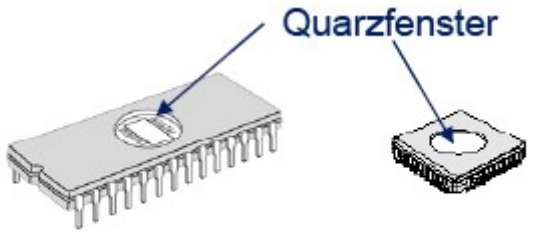


⇒ im Normalbetrieb ist der Ausgangstreiber aktiv, zur Programmierung werden die Ausgangstreiber gesperrt und die Eingangstreiber aktiviert.

EPROM – Löschbares (Erasable) Programmierbares ROM

⇒ insbesondere in der Entwicklungsphase einer Schaltung werden oft ROMs benötigt, die man löschen und neu programmieren kann

⇒ Dies ist mit einem EPROM möglich, bei dem die Programmierung des Bausteins gelöscht werden kann, wenn man das EPROM UV-Licht aussetzt.

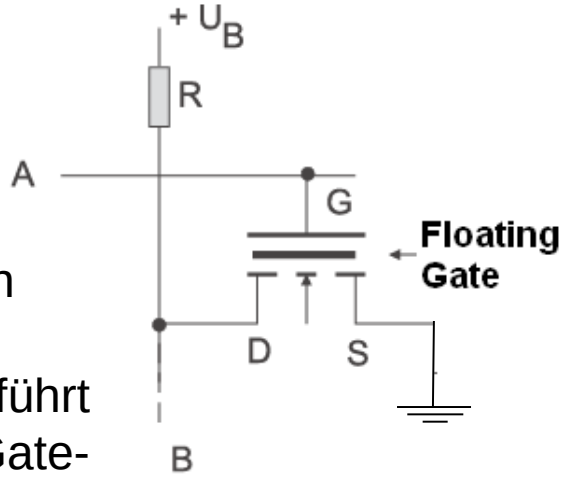


⇒ Nach Programmierung wird das Glasfenster zugeklebt

⇒ interner Aufbau von EPROMs ist anders wie die bisher vorgestellten PROMs, es kommen sog. FAMOS-Transistoren zum Einsatz

FAMOS = Floating Avalanche MOS

⇒ FAMOS hat zusätzlich zu einem MOS Transistor eine zweite Steuerelektrode (Floating Gate), die ganz vom Isolator umschlossen ohne äußeren Anschluss zwischen Gate und dem Substrat liegt.



⇒ Anlegen einer hohen Programmierspannung an A und B führt zum Stromfluss zwischen Drain und Source. Die hohe Gate-Spannung zieht die Elektronen an es kommt zum Durchbruch (Avalanche) von Elektronen durch die dünne Isolierschicht auf das Floating Gate.

EEPROM – electrically erasable PROM

⇒ EPROMs sind aufgrund des umständlichen Löschvorgangs für viele Anwendungen ungeeignet

⇒ bei EEPROM und Flash-Speicher kann der Prozessor den Speicherinhalt lesen, aber im Gegensatz zum PROM auch löschen und neu beschreiben

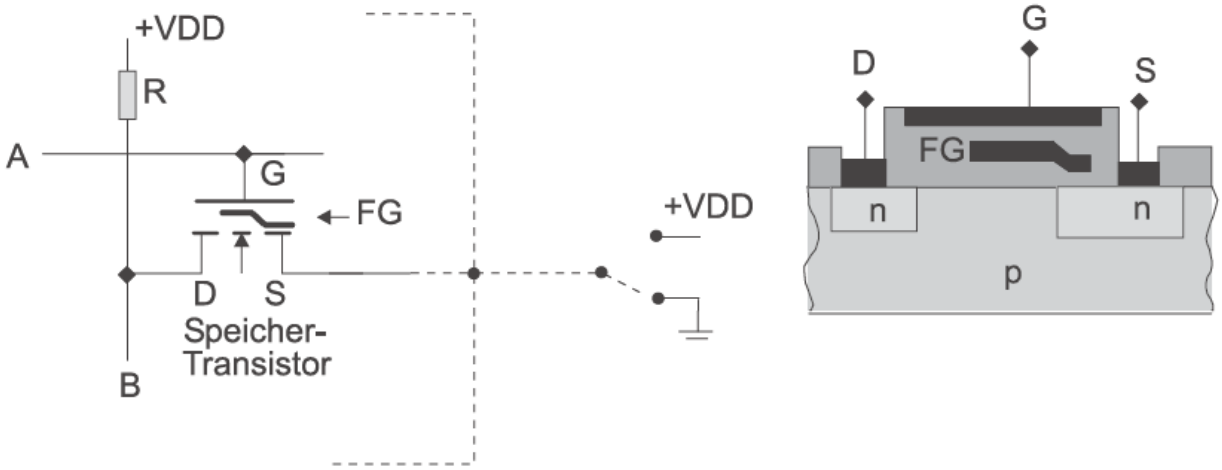
⇒ die Zeit zum Löschen und Beschreiben einer EEPROM- Speicherzelle liegt im Millisekundenbereich, was zu langsam ist für typische CPU-Taktzeiten, deswegen ist FLASH/EEPROM kein Ersatz für SRAM/DRAM !

⇒ Schreibzyklen einer Speicherzelle beträgt je nach Typ ca 10^4-10^6 Zyklen d.h. wenn z.B. in einer Programmschleife ein Speicherplatz eines EEPROMs zu oft angesprochen wird, kann der Baustein schon nach wenigen Sekunden das Ende seiner Lebensdauer erreichen !

⇒ technologische Grundlage für EEPROM und Flash ist der FLOTOX (Floating Gate Tunnel Oxid) Transistor

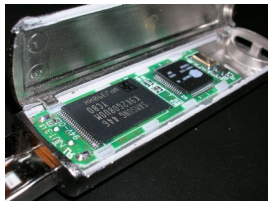
FLASH

=> Unterschied zwischen Flash und EEPROM ist im wesentlichen der, dass beim EEPROM einzelne Zellen beschrieben werden können, beim Flash werden große Speicherbereiche (manchmal der ganze Chip) komplett gelöscht, bevor ein Datum neu eingeschrieben werden kann.



⇒ Source-Anschlüsse aller Transistoren können im gesamten Speicherbaustein (bzw. in Teilbereichen) über einen Schalter (Schalttransistor) wahlweise mit Masse (normaler Betrieb) oder *VDD* verbunden werden (Löschvorgang)

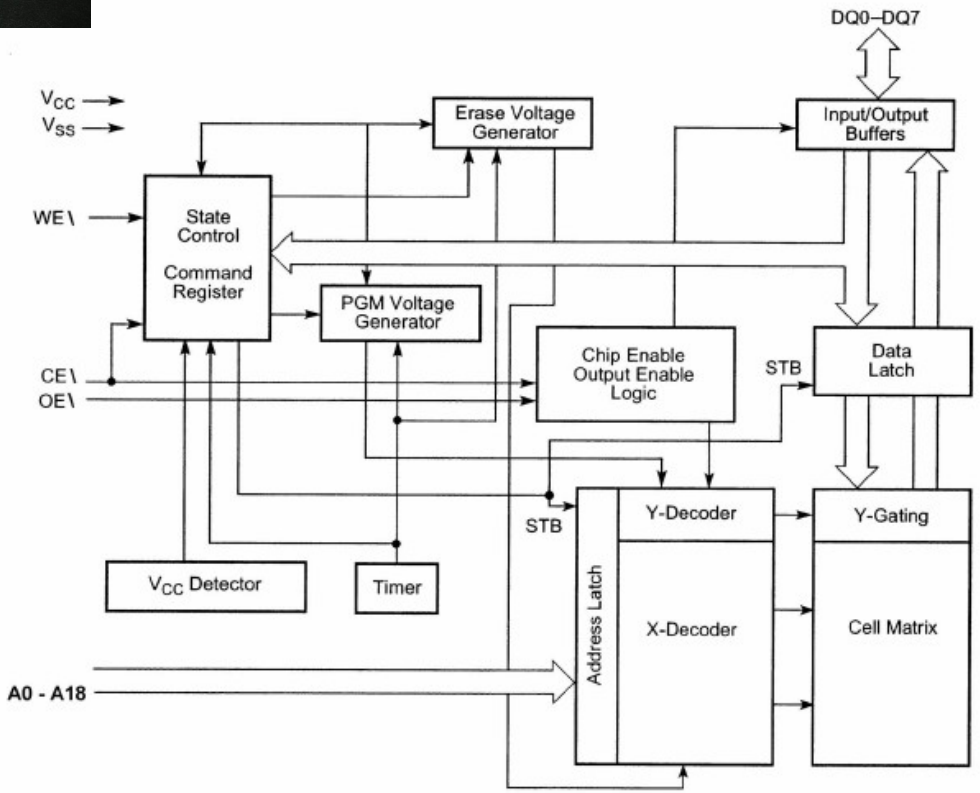
FLASH



AUSTIN SEMICONDUCTOR
Austin Semiconductor, Inc.

FLASH
AS29F040

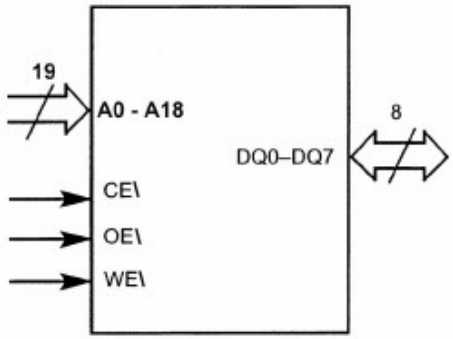
FUNCTIONAL BLOCK DIAGRAM



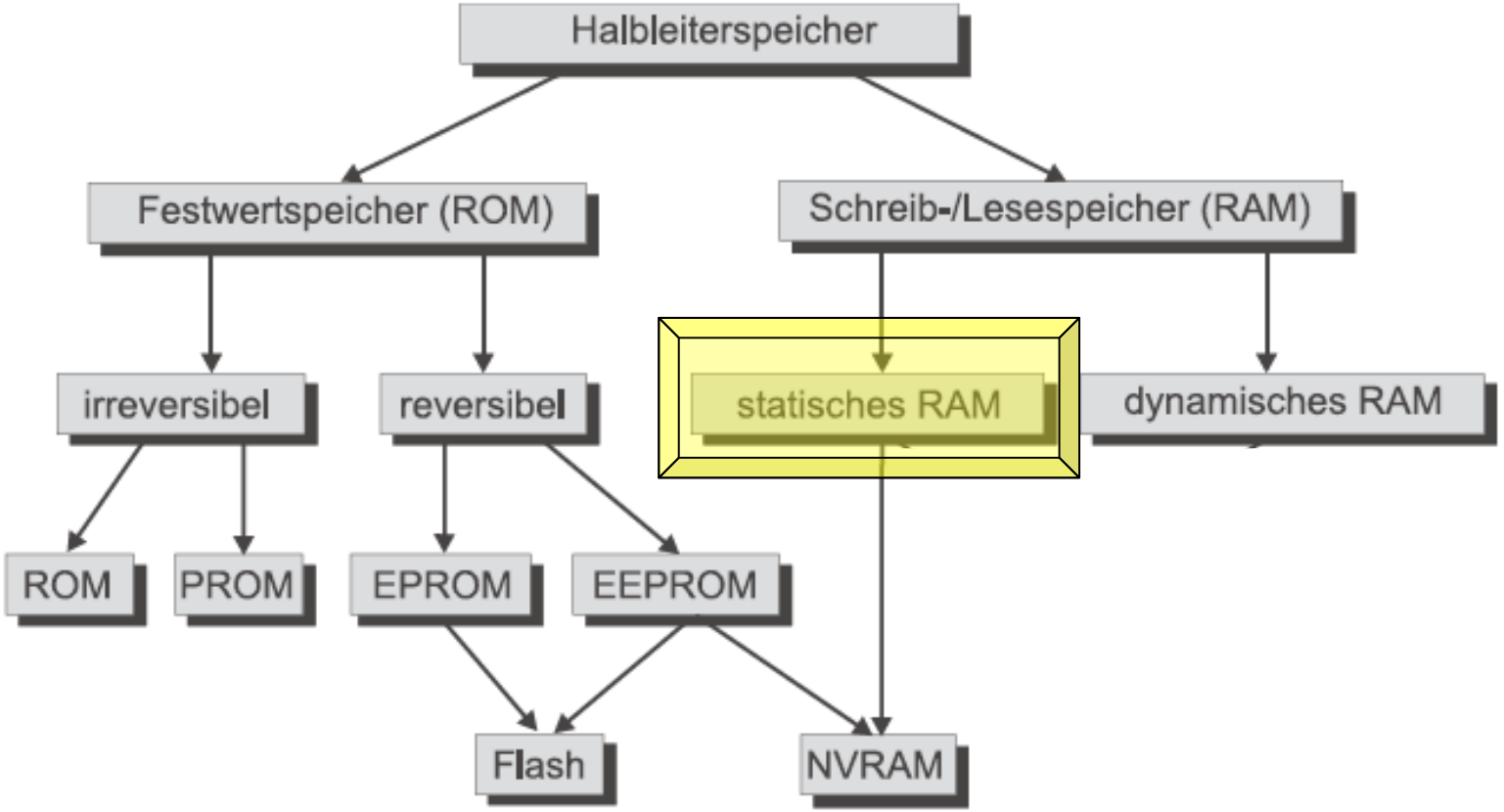
PIN CONFIGURATION

PIN	DESCRIPTION
A0 - A18	Address Inputs
DQ0 - DQ7	Data Inputs/Outputs
CE\	Chip Enable
OE\	Output Enable
WE\	Write Enable
V_{CC}	+5V Single Power Supply
V_{SS}	Device Ground

LOGIC SYMBOL



Mikrocomputertechnik – Halbleiterspeicher



Mikrocomputertechnik – Halbleiterspeicher

Schreib-/ Lesespeicher (RAM: *Random Access Memory)*

⇒ Inhalt ist flüchtig, d.h. geht mit dem Abschalten der Betriebsspannung verloren
Man unterscheidet 2 grundsätzliche Typen, SRAM und DRAM.

SRAMs: Statische RAMs
⇒ Information wird in Zellen gespeichert, die aus Flip-Flops bestehen, also einer Schaltung mit 2 stabilen Zuständen
⇒ SRAMs halten eine einmal eingeschriebene Information so lange, bis sie durch einen erneuten Speichervorgang verändert wird
⇒ Bipolar oder MOS-Technologie
⇒ schnell und teuer, daher in Cachehierarchie eher prozessornah

DRAMs: Dynamische RAMs
⇒ Information wird als elektrische Ladung in einem Kondensator gespeichert
⇒ Lesen bedingt in der Regel das Entladen des Kondensators, so dass danach der gelesene Wert wieder eingeschrieben werden muss.
⇒ Ladung geht durch unvermeidbare Leckströme kontinuierlich verloren, so dass in regelmäßigen Abständen aufgefrischt werden muss (Refresh Steuerlogik)
⇒ MOS Transistoren
⇒ langsam aber billig, Verwendung eher im prozessorfernen Hauptspeicher

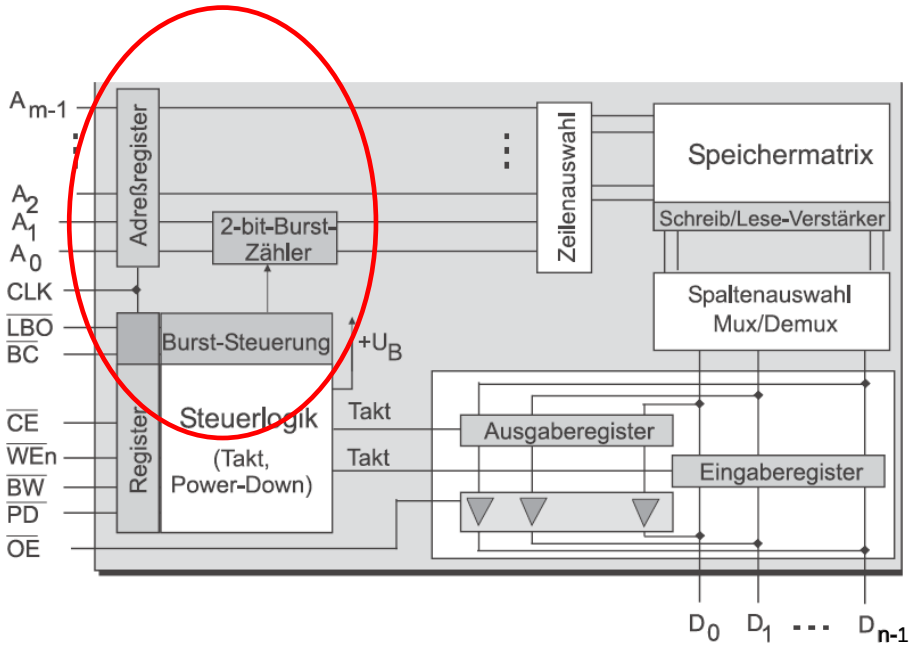
Asynchrone SRAM-Bausteine

Asynchrone SRAMs bestehen aus einer Speichermatrix, bidirektionalen Treibern und einer einfachen Steuerlogik mit /CE (/CS), /RW (/WE) und /OE.

Die Funktionsweise entspricht im Prinzip dem einführenden Beispiel.

Synchrone SRAM Bausteine

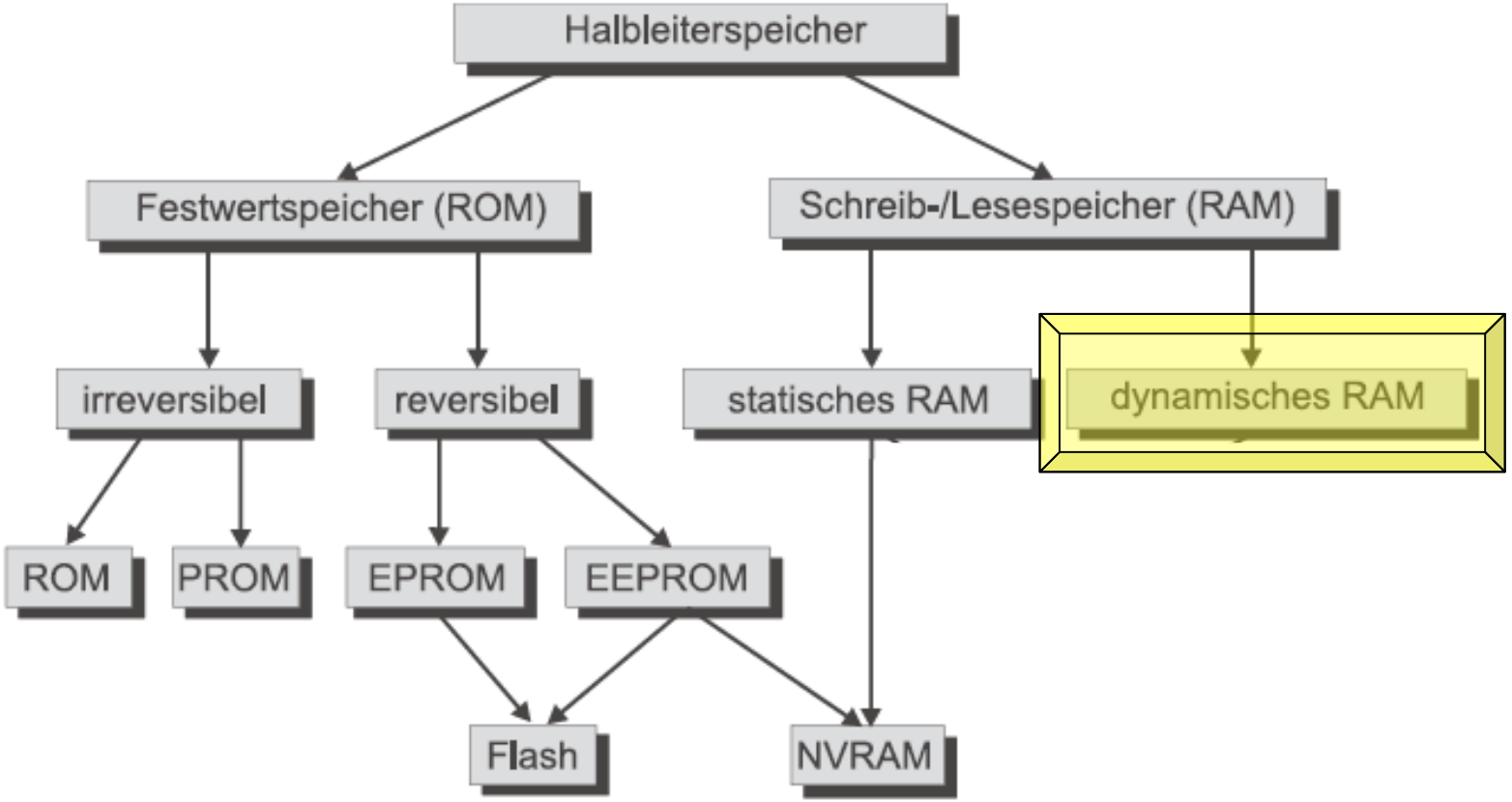
⇒ Steuersignale und Daten werden in zusätzlichen Registern aufgefangen
⇒ Register für Adress- und Steuersignale werden von externer Clock getriggert
⇒ Register für Daten werden von interner Logik gesteuert, meist braucht es eine Latenzzeit von 1 oder 2 Takten, bevor die Daten am Ausgang stehen



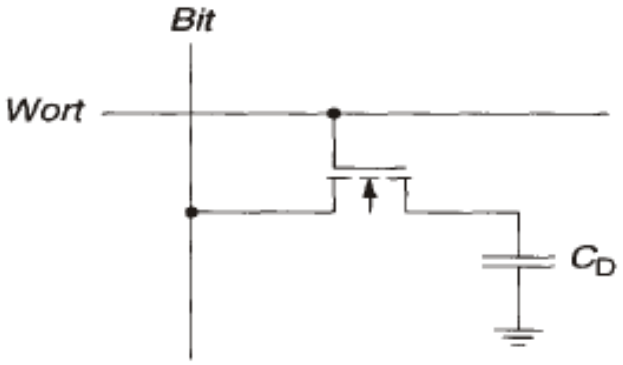
⇒ weitere Signalleitungen, z.B. für Burstübertragung, PowerDown, ByteWrite

⇒ Burst: liest mehrere benachbarte Speicherzellen nacheinander aus

Mikrocomputertechnik – Halbleiterspeicher



DRAM - Dynamische RAMs



Einfache Zellstruktur : Strom fließt wie bei der statischen CMOS-Zelle nur zu den Zeitpunkten des Schreibens bzw. Lesens.

Speicherzelle besteht nur aus einem Transistor und einem Kondensator (C ≈ 50…100fF)

Schreiben

Aktivieren der Wortleitung → T leitet → Datum wird auf Bitleitung gelegt

1 → C wird auf U_{DD} aufgeladen

0 → C wird auf 0V entladen

Deaktivieren der Wortleitung → T hochohmig → C speichert Info in Form el. Ladung

Lesen

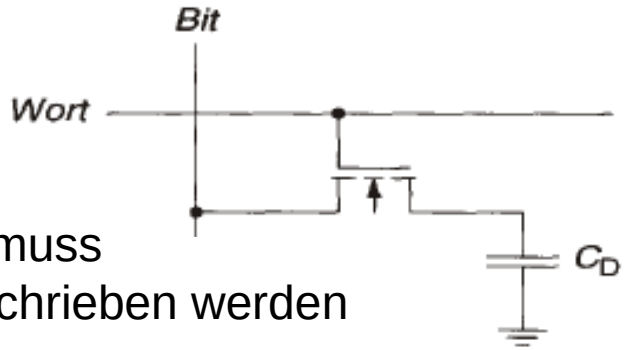
Aktivieren der Wortleitung

C (un-)geladen: (keine) Entladung über Bitleitung

(keine) Änderung der Spannung auf Bitleitung

=> Auswertung durch einen Leseverstärker, der ein digitales Signal erzeugt

DRAM - Dynamische RAMs



Refresh

⇒ beim Lesen wird C entladen, d. h. nach dem Lesen muss das ursprüngliche Datum in die Zelle zurückgeschrieben werden

⇒ Auch bei Nichtaktivierung der Zelle entladen sich die Kondensatoren durch Leckströme

⇒ Ein regelmäßiger Refresh (im ms-Bereich) des gesamten Speichers ist notwendig (Refresh bedeutet auslesen und neu einschreiben)

Precharge-Technik und Leseverstärker

⇒ Mit zunehmender Miniaturisierung wird das Verhältnis der reinen Zellkapazität zur parasitären Kapazität der Bit-Auswahlleitungen immer kleiner.

⇒ Es wird zunehmend schwieriger, den Ladezustand des Kondensators sicher zu erkennen und auszuwerten.

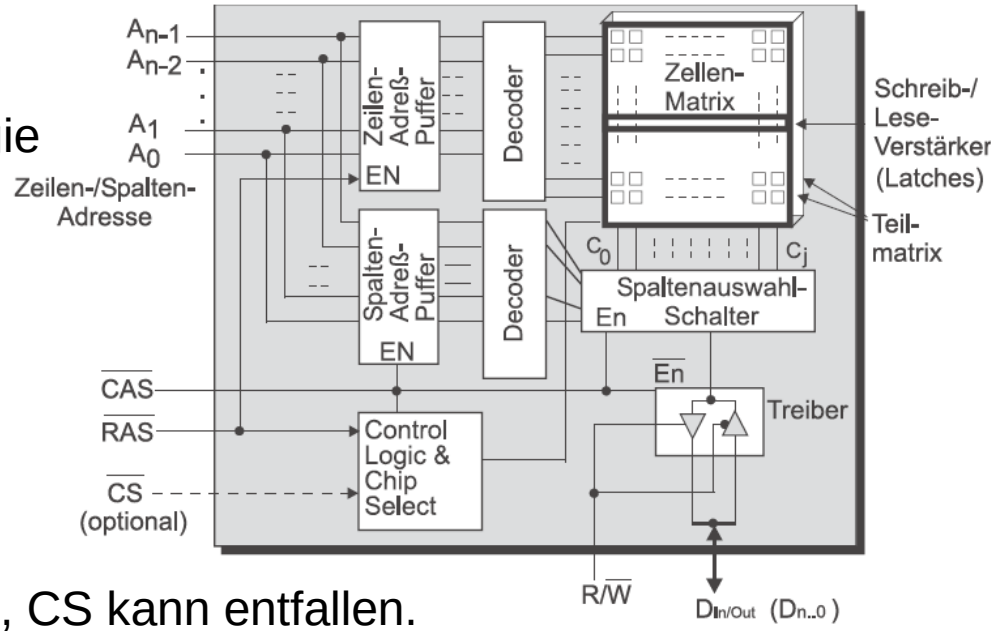
⇒ Einsatz von differentiellen Leseverfahren mit Precharge-Vergleichstechnik

DRAM Bausteine

Integrationsdichte SRAM vs DRAM am Beispiel der Infineon 0,20 µm Technologie

1 SRAM-Zelle : ca. 10 - 11 µm²
1 DRAM-Zelle : ca. 0,7 - 1 µm²

⇒ Die Adresse wird zur Pinreduktion gemultiplext und mit den Signalen CAS (Spalte) und RAS (Zeile) eingelesen.



RAS dient oft auch zur Bausteinauswahl, CS kann entfallen.

Reine DRAMs sind quasi vom Markt verschwunden, es gibt nun eine Vielzahl von Untertypen mit verschiedenen Ansätzen für einen beschleunigten Zugriff :

FPM (*Fast Page Mode*), EDO Mode (*Extended Data Output* DRAMs, SDRAM (*Synchrone* DRAMs), DDR (*Double Data Rate* DRAMs), Rambus, und weitere mehr…

DRAM Controler

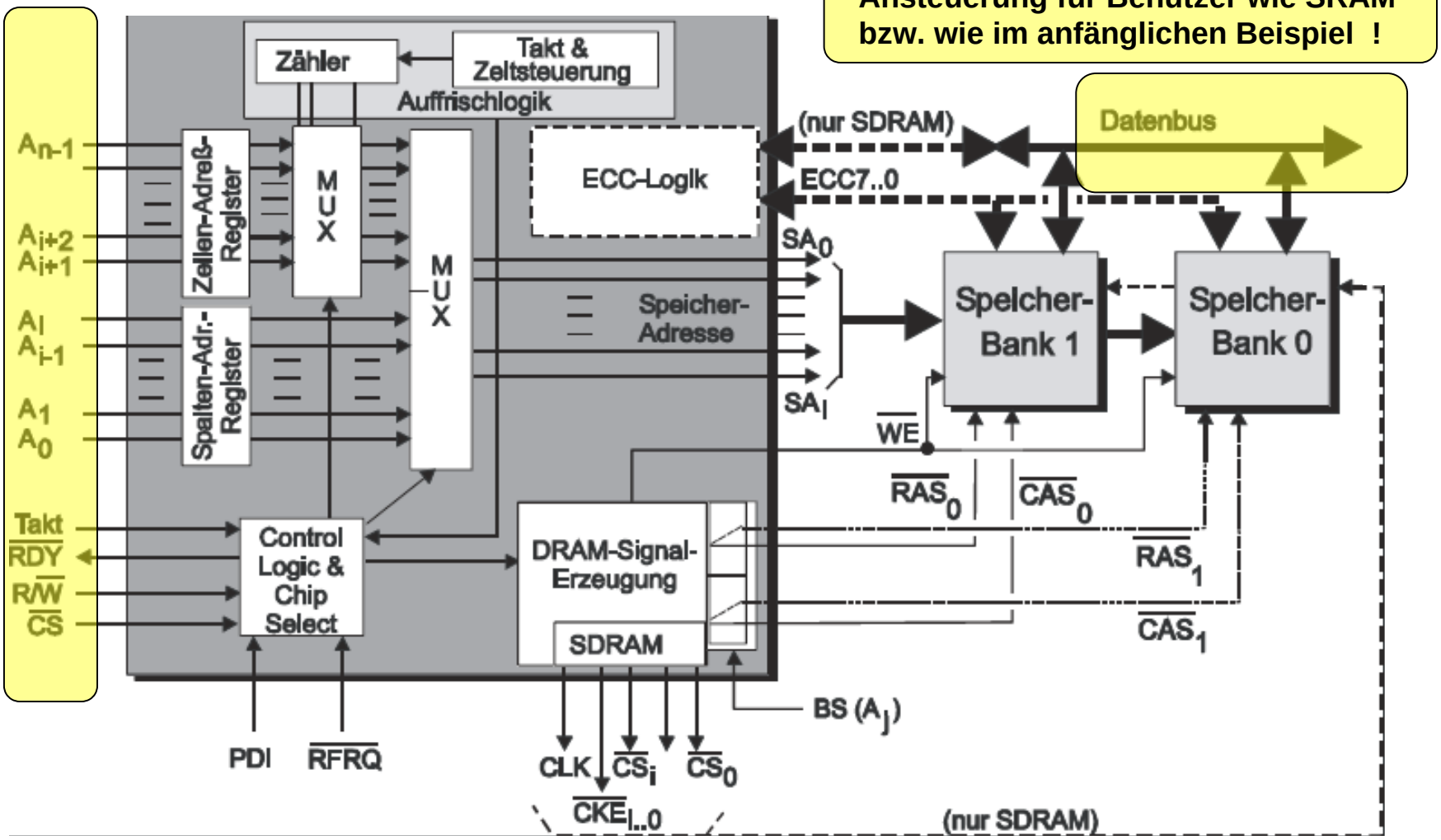
⇒ DRAM Bausteine benötigen eine relativ aufwendige Ansteuerung
⇒ daher gibt es DRAM-Controler als integrierte Schaltungen fertig zu kaufen
⇒ in manchen sog. Brückenbausteinen (Bridges) ist ein solcher Controler bereits vorhanden, z.B. in der PCI Host-Bridge
⇒ viele moderne Mikroprozessoren und Mikrocontroller haben bereits einen DRAM Controler-on-Chip
⇒ Fehlerkorrektur mit redunanten Zusatzprüfbits ist oft On-Chip integriert. ECC (Error Correction Code)
⇒ DRAM Controler enhalten selbst Speicher als Pufferregister für Daten sowie Steuer- und Statusregister zur Kontrolle des Arbeitsspeichers

⇒ **DRAM Controler sind meist programmierbar und können unterschiedliche Typen von DRAMs unterstützen und diverse Parameter setzen wie z.B.**

DRAM-Typ (EDO, SDRAM,..) Refresh-Typ (RAS-only, CAS before RAS,…), Refresh Rate, Wartezyklen vor RAS/CAS-Aktivierung, Größe von Pages, Taktfrequenz für SDRAMs, Anzahl der Bänke in SDRAM-Speicher, Verzögerungszeiten für Steuersignale usw.

DRAM Controler

Ansteuerung für Benutzer wie SRAM bzw. wie im anfänglichen Beispiel !



Organisation des Arbeitsspeichers

⇒ dem Prozessor erscheint der Arbeitsspeicher wie eine lineare Liste von Einträgen, von denen jeder durch Angabe einer Adresse wahlfrei (Random Access) selektiert werden kann.

⇒ in einfachen Systemen ist der Arbeitsspeicher direkt an den Prozessorbus angeschlossen, in komplexeren Systemen an einen Speicherbus oder Peripheriebus.

⇒ ein Speicherwort entspricht der maximalen Informationsmenge, diedurch einen einzigen Speicherzugriff über Speicherbus übertragen werden kann.

⇒ viele Prozessoren mit einer großen Datenbusbreite können auch einzelne Bytes oder 16-bzw. 32-bit-Wörter schreiben bzw. lesen.

⇒ die maximale Kapazität des Arbeitsspeichers wird durch die Breite der Adresse, die der µP bzw. Speichercontroller ausgibt vorgegeben

⇒ Die Kapazität des tatsächlichen Arbeitsspeichers ist meist erheblich kleiner und stellt den physikalischen Adressraum des Mikroprozessors dar.

Memory Map

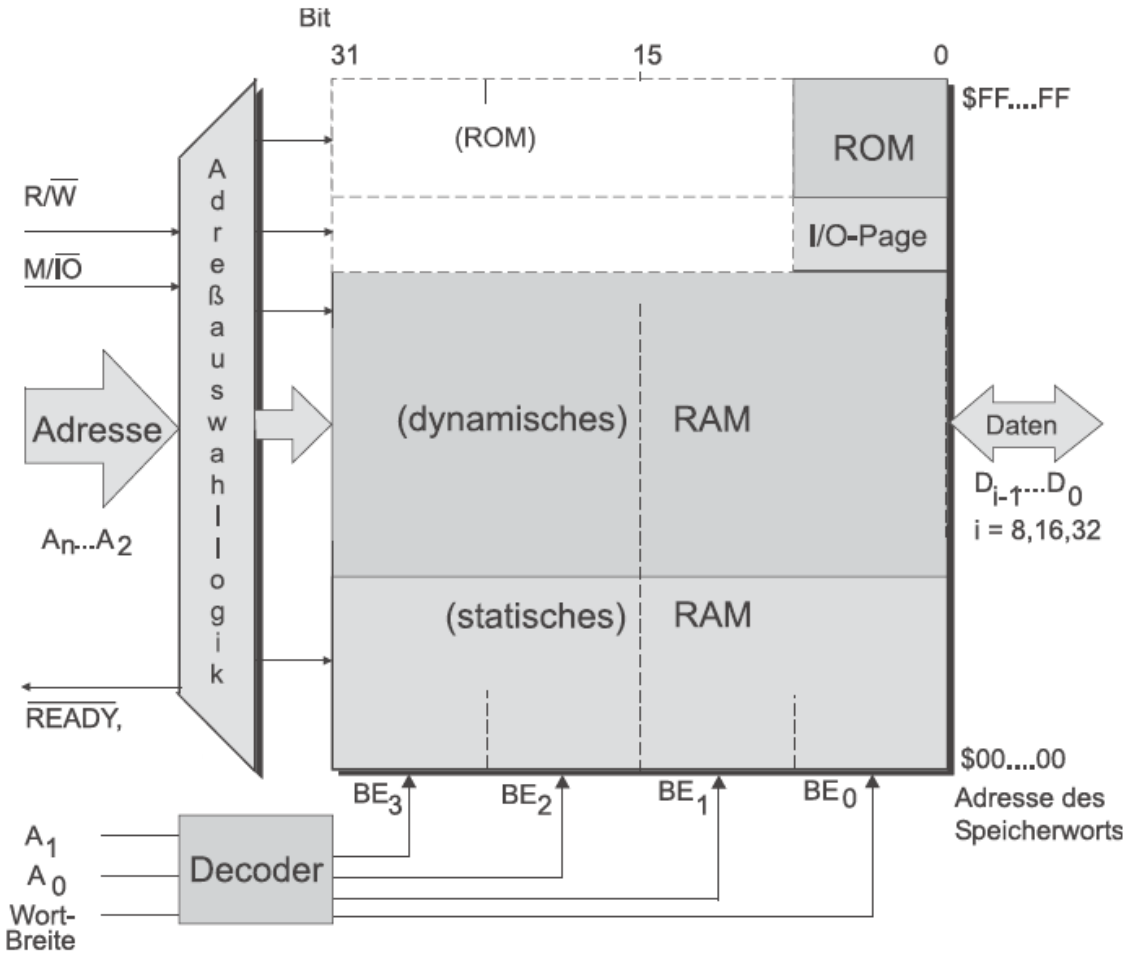
ROM (8 Bit)
z.B. Programme zum Booten des Betriebssystems

I/O Bereich (meist 8 Bit)
Memory-Mapping ordnet diesem Bereich Register der Peripherie-Bausteine zu, Zugriff auf Peripherie benötigt keine eigenen Befehle

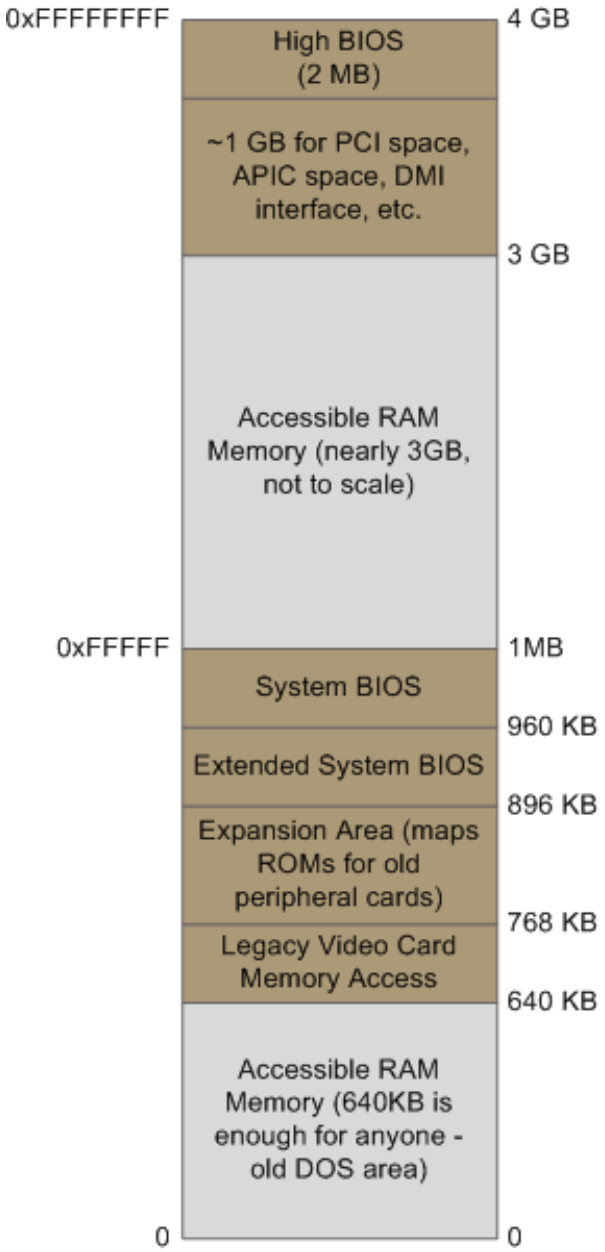
DRAM : Größter Teil (Kosten)

SRAM : kleiner Teil, sehr schnell

Adressauswahl pro Byte
Daher werden LSBs zur Byteselektion einem Decoder zugeführt



Memory Mapping bei Pentium PCs



Besondere Speichertypen : Dual Ported RAM (meist SRAM)

⇒ jede Speicherzelle kann von 2 voneinander unabhängigen Ports ausgelesen oder beschrieben werden
⇒ 2 unabhängige Daten bzw. Adressports
⇒ auch die Ansteuerelektronik ist in symmetrischer Form zweifach vorhanden
⇒ Probleme gibt es, wenn beide Ports gleichzeitig auf dieselbe Adresse schreiben wollen oder ein Port gerade schreibt und zweite gleichzeitig von derselben Adresse lesen will
⇒ Problemkonstellationen müssen erkannt und durch Blockierung eines Ports gelöst werden. Die dazu erforderliche Arbitrationslogik hierfür ist üblicherweise auf dem Chip enthalten. Signale : Busy , Semaphore

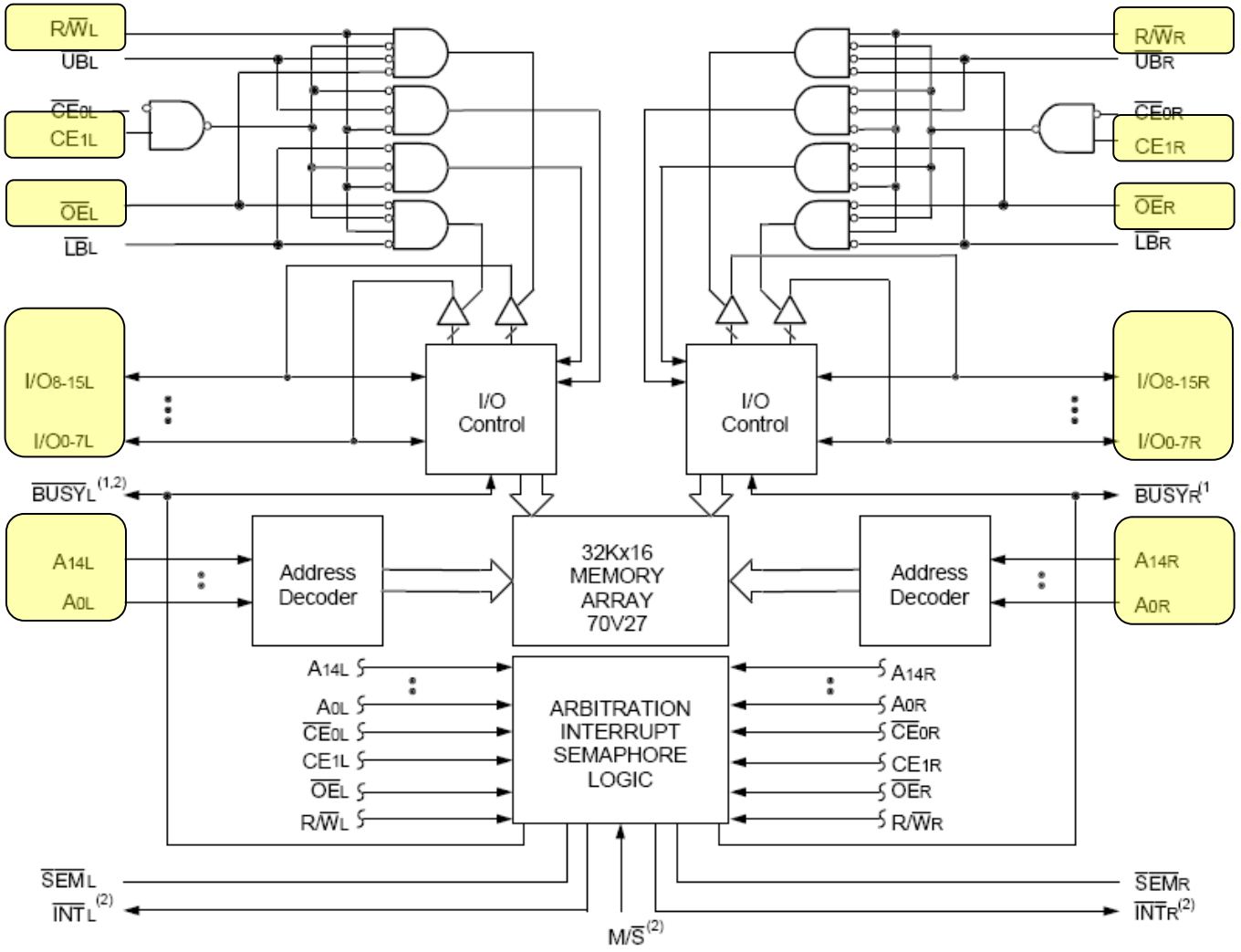
Dual Ported RAM - Anwendungen

Multiprozessorsysteme, Datenaustausch zwischen 2 Prozessoren, Kopplung von Mikroprozessorsystemen oder Bussystemen, Video-RAM (VRAM, sehr schnell !)

Beispielrechnung VRAM : Bildspeicher mit 1024x1024 Pixel muss 50 mal in der Sekunde zyklisch auf den Bildschirm gegeben werden => ca. 20ns pro Pixel

Andererseits wird Inhalt vom Prozessor laufend aktualisiert → Dual-Port-RAM

Beispiel: Dual Ported RAM - IDT 70V27



Steuerbus Datenbus Adressbus

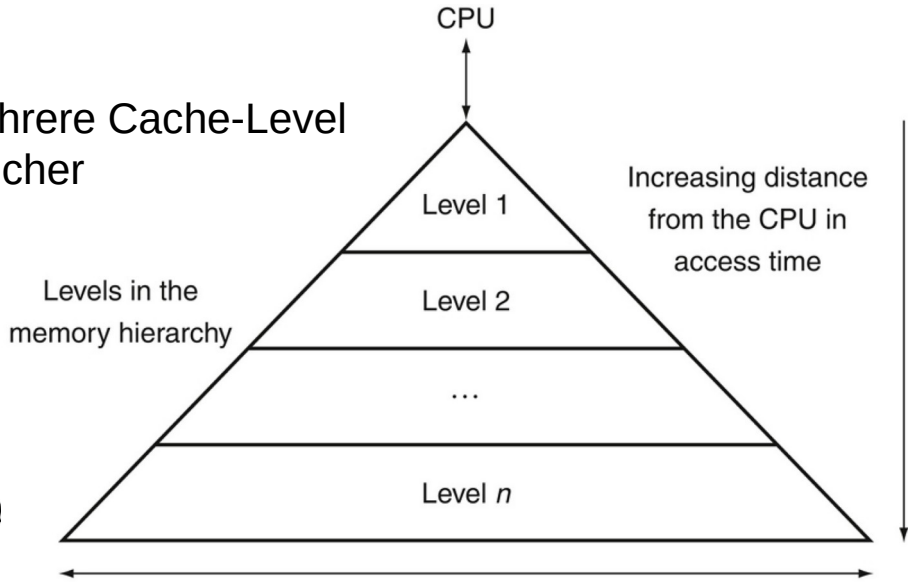
Literatur zu diesem Kapitel

[1] H. Bähring, *Mikrorechnertechnik*, Band 2
[2] U. Tietze, Ch. Schenk, Halbleiter-Schaltungstechnik
[3] K. Beuth, *Digitaltechnik*
[4] C. Siemers, A. Sikora, Taschenbuch Digitaltechnik,
[5] K. Wüst, *Mikroprozessortechnik*

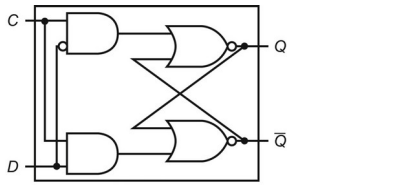
Die Speicher-Hierarchie

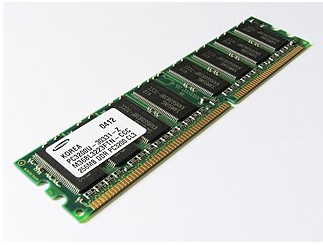
Mikrocomputertechnik – Halbleiterspeicher

- Moderne Rechnerarchitekturen besitzen mehrere Cache-Level zwischen CPU und dem (großen) Hauptspeicher
- Je schneller der Zugriff, desto teurer
- Idee: Speicherzugriffe sind meist lokal – sowohl zeitlich, als auch räumlich (Code)



Speed		Size	Cost ($/bit)	Current technology
	Processor			
Fastest	Memory	Smallest	Highest	SRAM
	Memory			DRAM
Slowest	Memory	Biggest	Lowest	Magnetic disk or flash memory

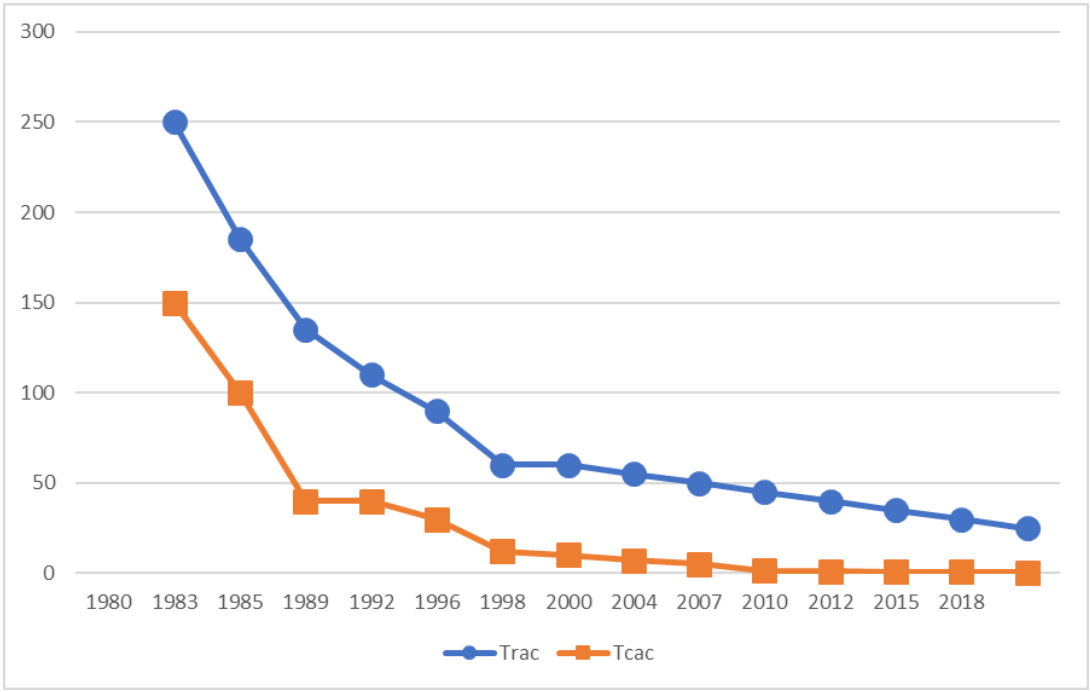




Quelle: Wikipedia, Patterson&Hen.

DRAM-Kosten und Zugriffszeiten

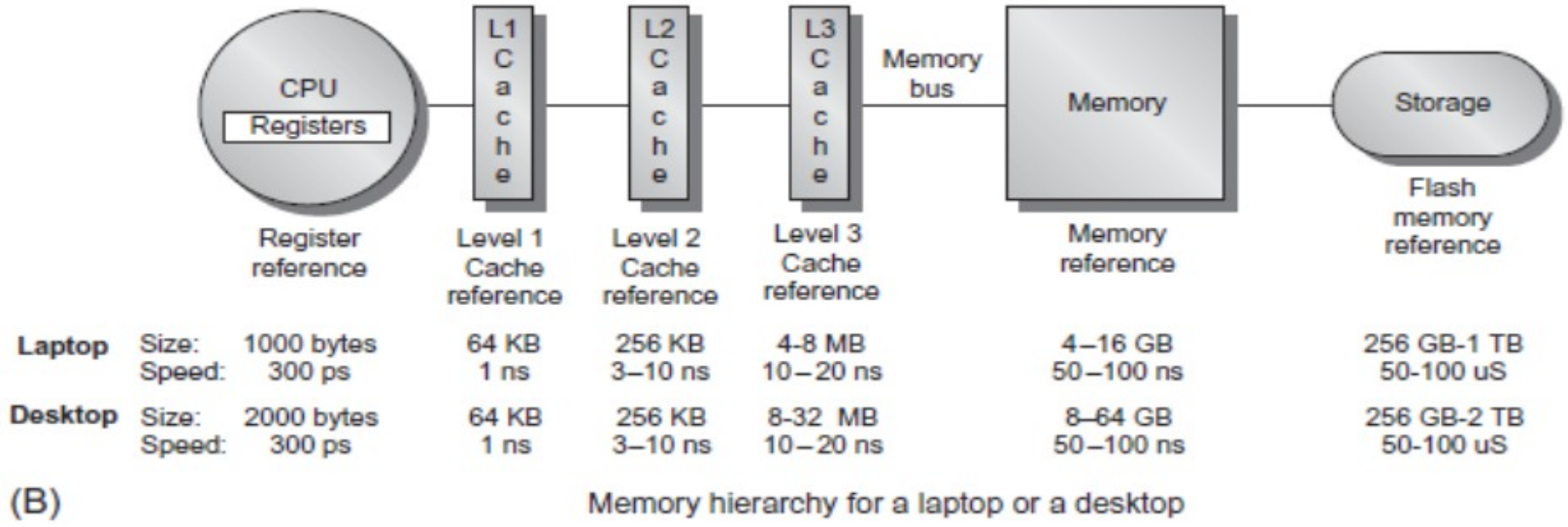
Jahr	Chip-Größe	$/GB
1980	64 Kibibit	$6,480,000
1983	256 Kibibit	$1,980,000
1985	1 Mebibit	$720,000
1989	4 Mebibit	$128,000
1992	16 Mebibit	$30,000
1996	64 Mebibit	$9,000
1998	128 Mebibit	$900
2000	256 Mebibit	$840
2004	512 Mebibit	$150
2007	1 Gibibit	$40
2010	2 Gibibit	$13
2012	4 Gibibit	$5
2015	8 Gibibit	$7
2018	16 Gibibit	$6



Zeilenzugriffszeiten t_{RAC} und
Spaltenzugriffszeiten t_{CAC} in ns

Caches

Typische heutige Systeme besitzen mehrere Cache-Ebenen (SRAM)



(B) Memory hierarchy for a laptop or a desktop

Core i7:

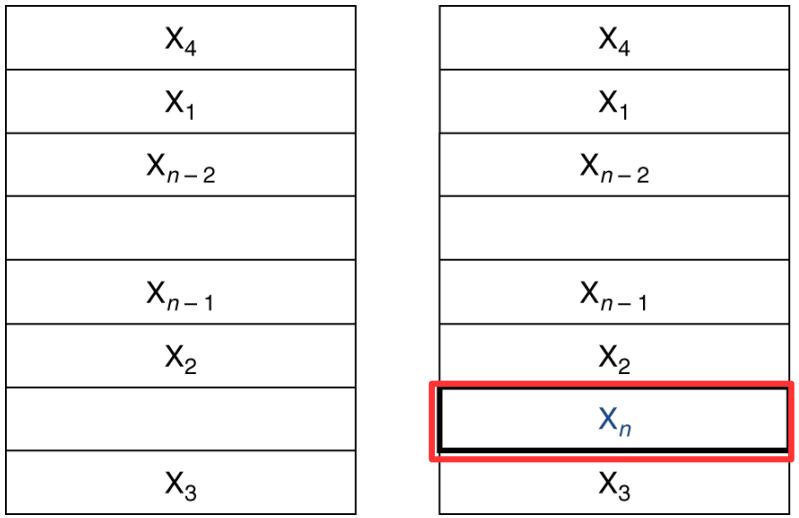
Characteristic	L1	L2	L3
Size	32 KiB I/32 KiB D	256 KiB	2 MiB per core
Associativity	both 8-way	4-way	16-way
Access latency	4 cycles, pipelined	12 cycles	44 cycles
Replacement scheme	Pseudo-LRU	Pseudo-LRU	Pseudo-LRU but with an ordered selection algorithm

Cache-Zugriff

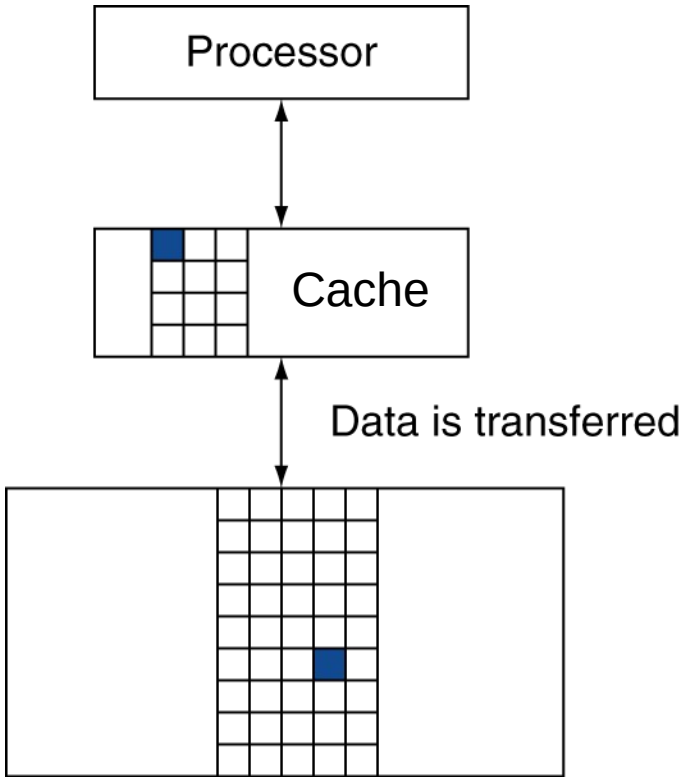
Wenn der SRAM-Cache die Daten nicht vorhält, muss die CPU sie aus dem DRAM-basierten Hauptspeicher laden

Aber wie speichern / ablegen?
Wie weiß die CPU, welche Werte im Cache sind?

Beispiel: Zugriff auf Zellen X_1, …, X_{n-1}, X_n



a. Before the reference to X_n b. After the reference to X_n

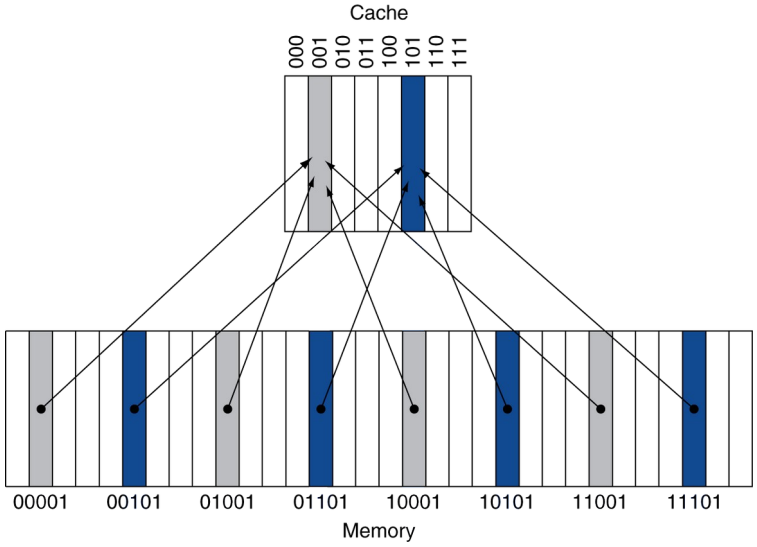


Direct Mapped Caches

Für jede Speicher-Adresse gibt es genau einen Platz im Cache, abhängig von der Adresse.

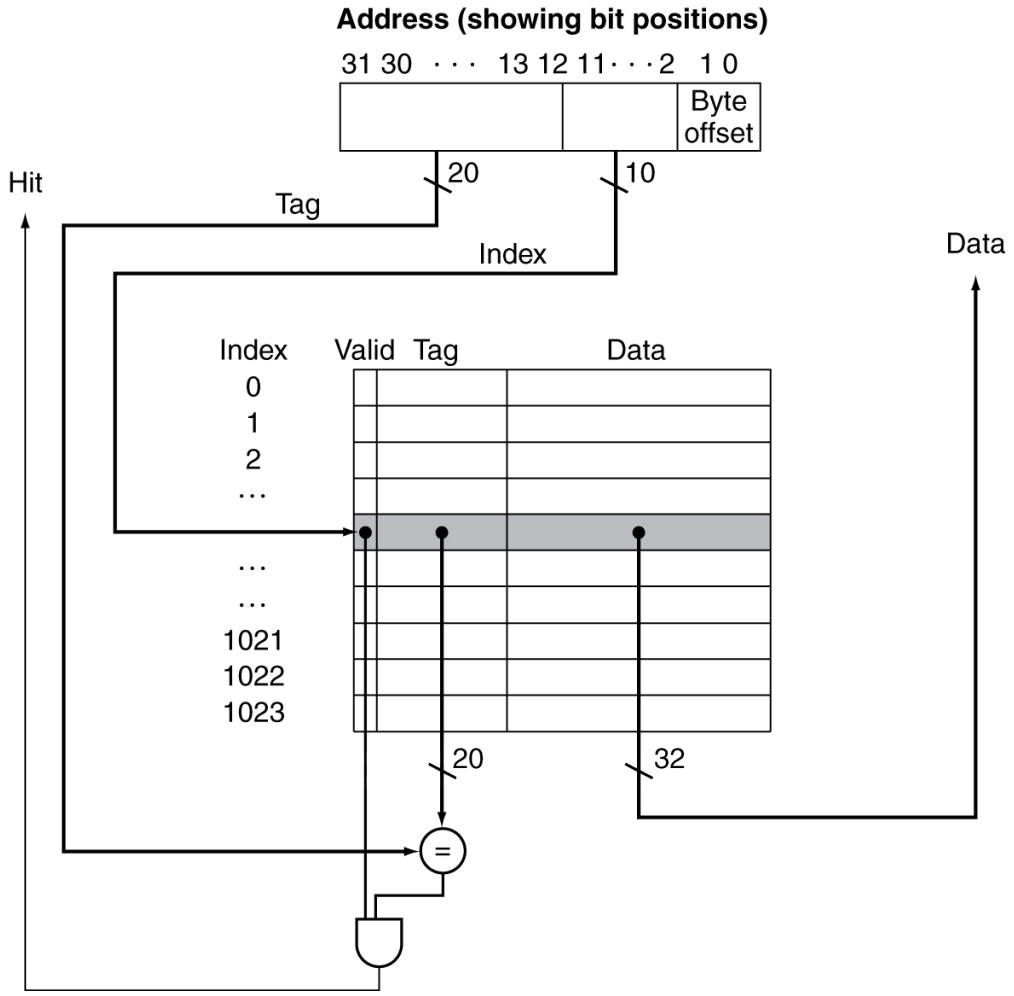
Aber: Cache ist kleiner als Hauptspeicher!

→ Speicherblock-Adresse modulo (Block-Einträge im Cache)
Modulo: in diesem Fall „höhere Bits abschneiden“
→ Restliche Bits „mappen“ auf den Block
→ wir müssen die oberen Bits aber zusätzlich ablegen (Tags)



Direct Mapped Caches

Speicher-Adressen werden aufgeteilt in Index in den Cache, sowie Tag



Direct Mapped Cache: Beispiel

- 8-blocks, 1 word/block, direct mapped
- Zu Anfang ist der Cache leer

Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

Direct Mapped Cache: Beispiel

Word addr	Binary addr	Hit/miss	Cache block
22	10 110	Miss	110

Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	**Y**	**10**	**Mem[10110]**
111	N		

Direct Mapped Cache: Beispiel

Word addr	Binary addr	Hit/miss	Cache block
26	11 010	Miss	010

Index	V	Tag	Data
000	N		
001	N		
010	**Y**	**11**	**Mem[11010]**
011	N		
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Direct Mapped Cache: Beispiel

Word addr	Binary addr	Hit/miss	Cache block
22	10 110	Hit	110
26	11 010	Hit	010

Index	V	Tag	Data
000	N		
001	N		
010	Y	11	Mem[11010]
011	N		
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Direct Mapped Cache: Beispiel

Word addr	Binary addr	Hit/miss	Cache block
16	10 000	Miss	000
3	00 011	Miss	011
16	10 000	Hit	000

Index	V	Tag	Data
000	**Y**	**10**	**Mem[10000]**
001	N		
010	Y	11	Mem[11010]
011	**Y**	**00**	**Mem[00011]**
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Direct Mapped Cache: Beispiel

Word addr	Binary addr	Hit/miss	Cache block
18	10 010	Miss	010

Index	V	Tag	Data
000	Y	10	Mem[10000]
001	N		
010	**Y**	**10**	**Mem[10010]** Wert überschrieben
011	Y	00	Mem[00011]
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Assoziative Caches

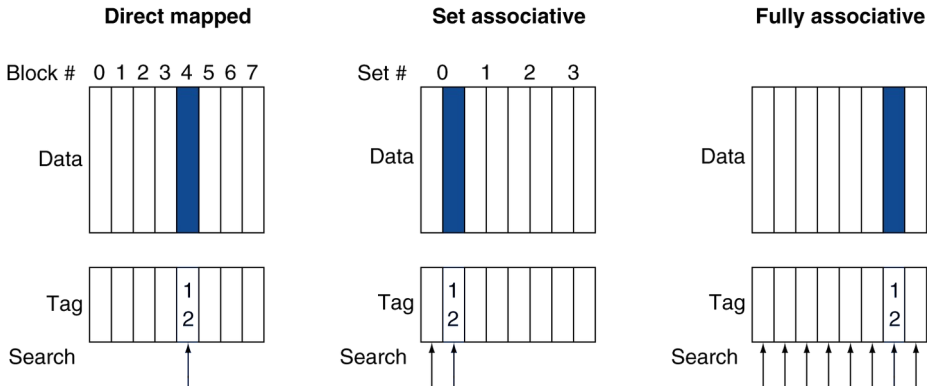
Caches sind meistens „assoziativ“, d.h., sie haben pro Index mehrere Einträge

n-way bedeutet:
n Einträge pro Index

Characteristic	L1	L2	L3
Size	32 KiB I/32 KiB D	256 KiB	2 MiB per core
Associativity	both 8-way	4-way	16-way
Access latency	4 cycles, pipelined	12 cycles	44 cycles
Replacement scheme	Pseudo-LRU	Pseudo-LRU	Pseudo-LRU but with an ordered selection algorithm

Bei neuem Cache-Eintrag:
- ➔ Speicheradresse modulo (Zeilen im Cache)
- ➔ Freien Eintrag finden (z.B. 8-way: Zeile hat 8 Plätze)

Cache-Lookup:
- ➔ Adresse bestimmen (mod)
- ➔ Alle belegten Einträge suchen, Tag vergleichen



n-way Assoziative Caches

One-way set associative
(direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

Assoziative Caches: Beispiel

4-Block-Cache, direct Mapped
Zugriff auf Adressen: 0, 8, 0, 6, 8

Block address	Cache index	Hit/miss	Cache content after access			
			0	1	2	3
0	0	miss	**Mem[0]**			
8	0	miss	**Mem[8]**			
0	0	miss	**Mem[0]**			
6	2	miss	Mem[0]		**Mem[6]**	
8	0	miss	**Mem[8]**		Mem[6]	

Assoziative Caches: Beispiel

Zugriff auf Adressen: 0, 8, 0, 6, 8

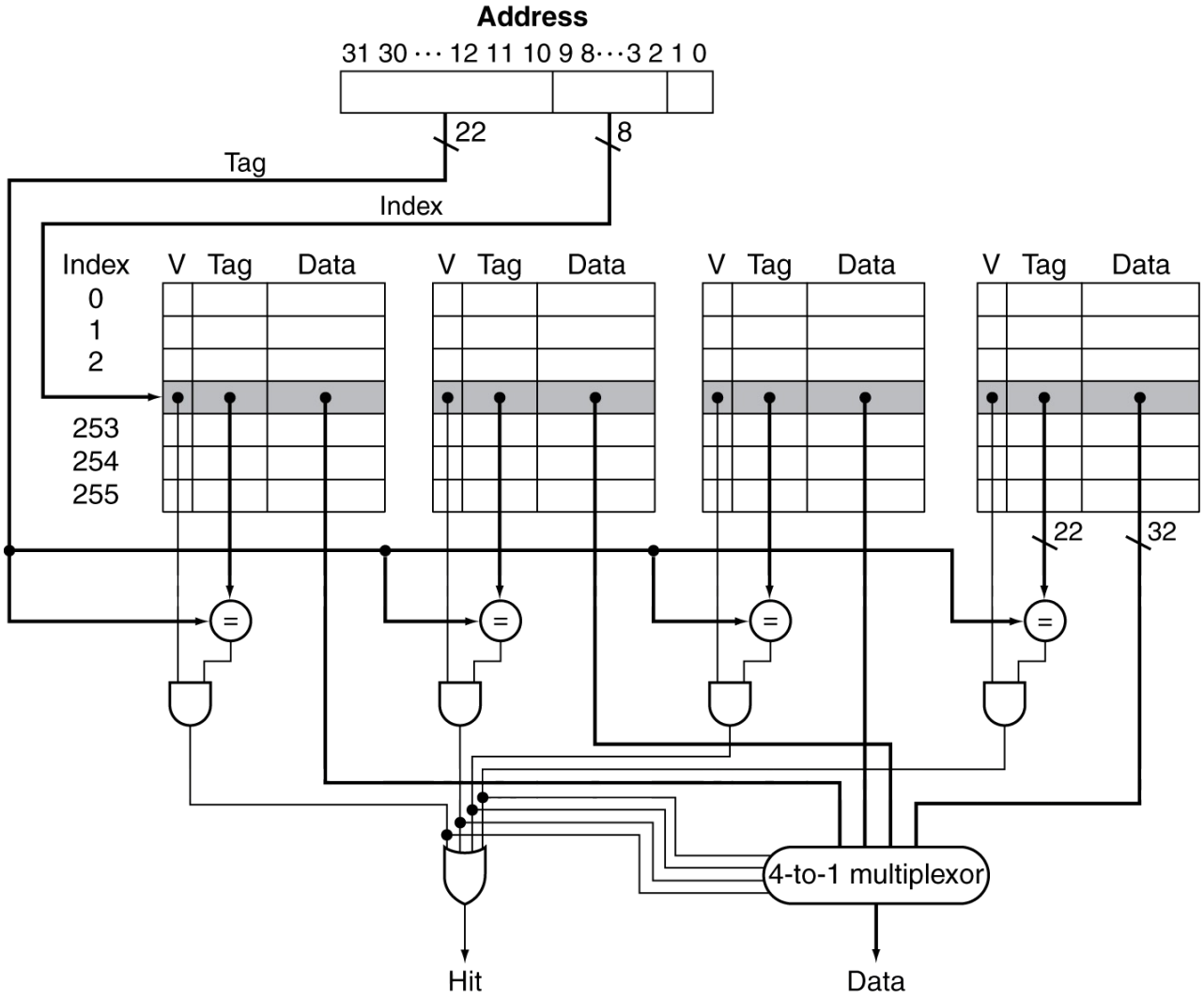
2-way set associative:

Block address	Cache index	Hit/miss	Cache content after access			
			Set 0		Set 1	
0	0	miss	**Mem[0]**			
8	0	miss	Mem[0]	**Mem[8]**		
0	0	hit	**Mem[0]**	Mem[8]		
6	0	miss	Mem[0]	**Mem[6]**		
8	0	miss	**Mem[8]**	Mem[6]		

Fully associative (**alle** Einträge müssen durchsucht werden):

Block address		Hit/miss	Cache content after access			
0		miss	**Mem[0]**			
8		miss	Mem[0]	**Mem[8]**		
0		hit	**Mem[0]**	Mem[8]		
6		miss	Mem[0]	Mem[8]	**Mem[6]**	
8		hit	Mem[0]	**Mem[8]**	Mem[6]	

Assoziative Caches: techn. Umsetzung



Virtueller Speicher

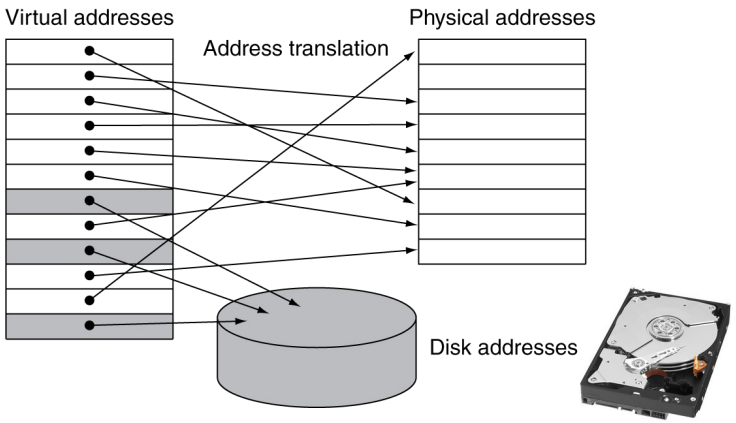
Virtueller Speicher

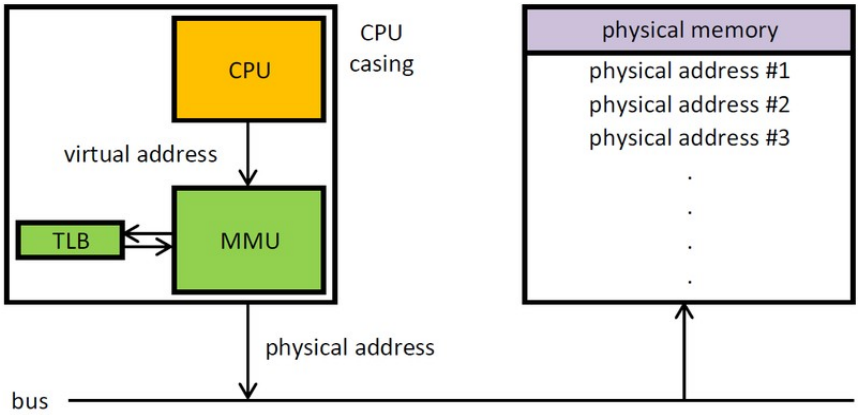
Moderne Rechner besitzen eine Memory Management Unit

Die MMU
- rechnet virtuelle Speicheradressen in physikalische um
- bietet Speicherschutzfunktion: Lesen / Schreiben erlaubt/verboten

Speicher-Seiten, die auf Festplatte ausgelagert sind, dürfen weder beschrieben noch gelesen werden

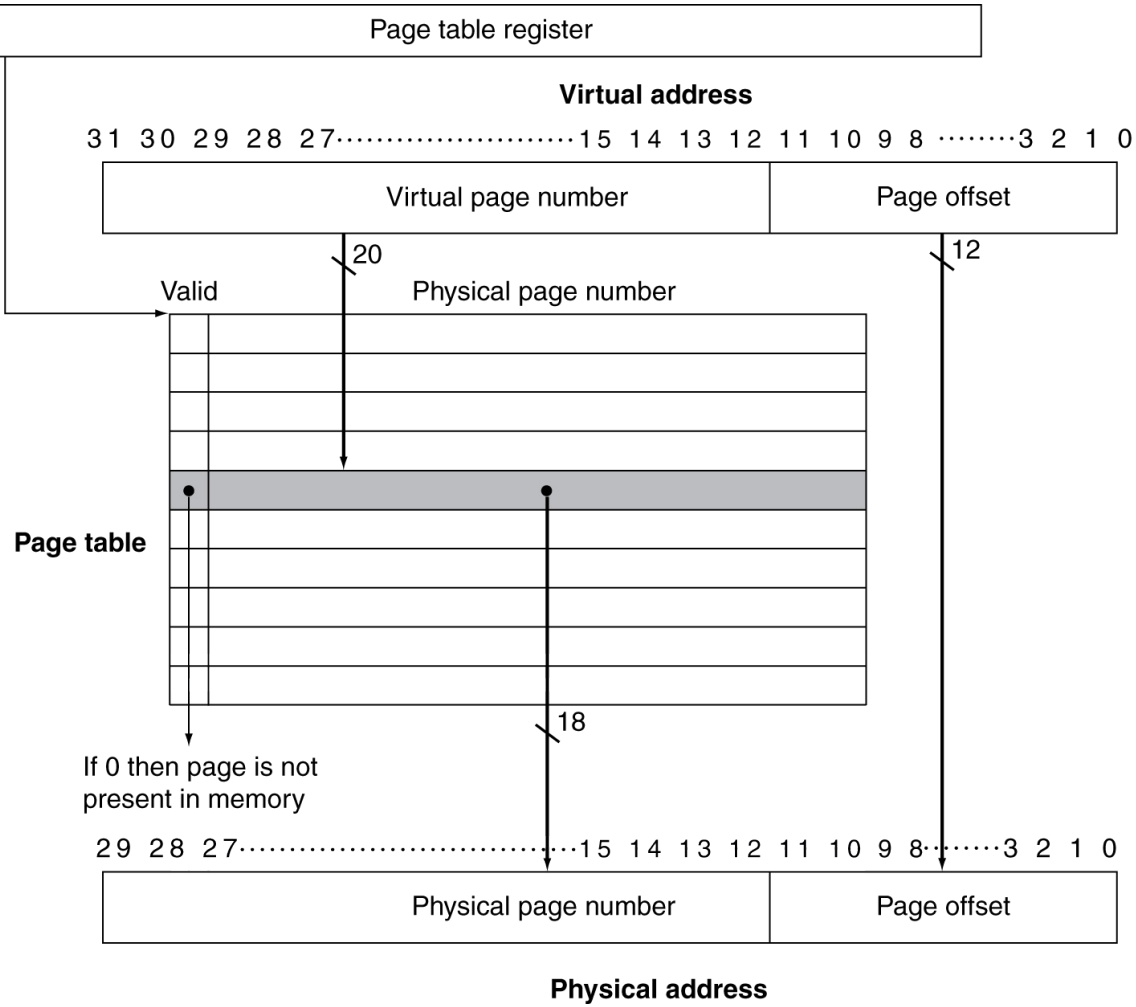
➔ bei Zugriff Trap ins Betriebssystem, das die Seiten nachlädt (Page Fault)





Quelle: Wikipedia

Abbildung virtueller zu physikalischen Adressen



TLB – Translation Lookaside Buffer

Ein spezieller Hardware-Puffer agiert als Cache für die Adressumsetzung: der Translation Lookaside Buffer (erspart Zugriff auf Page table)