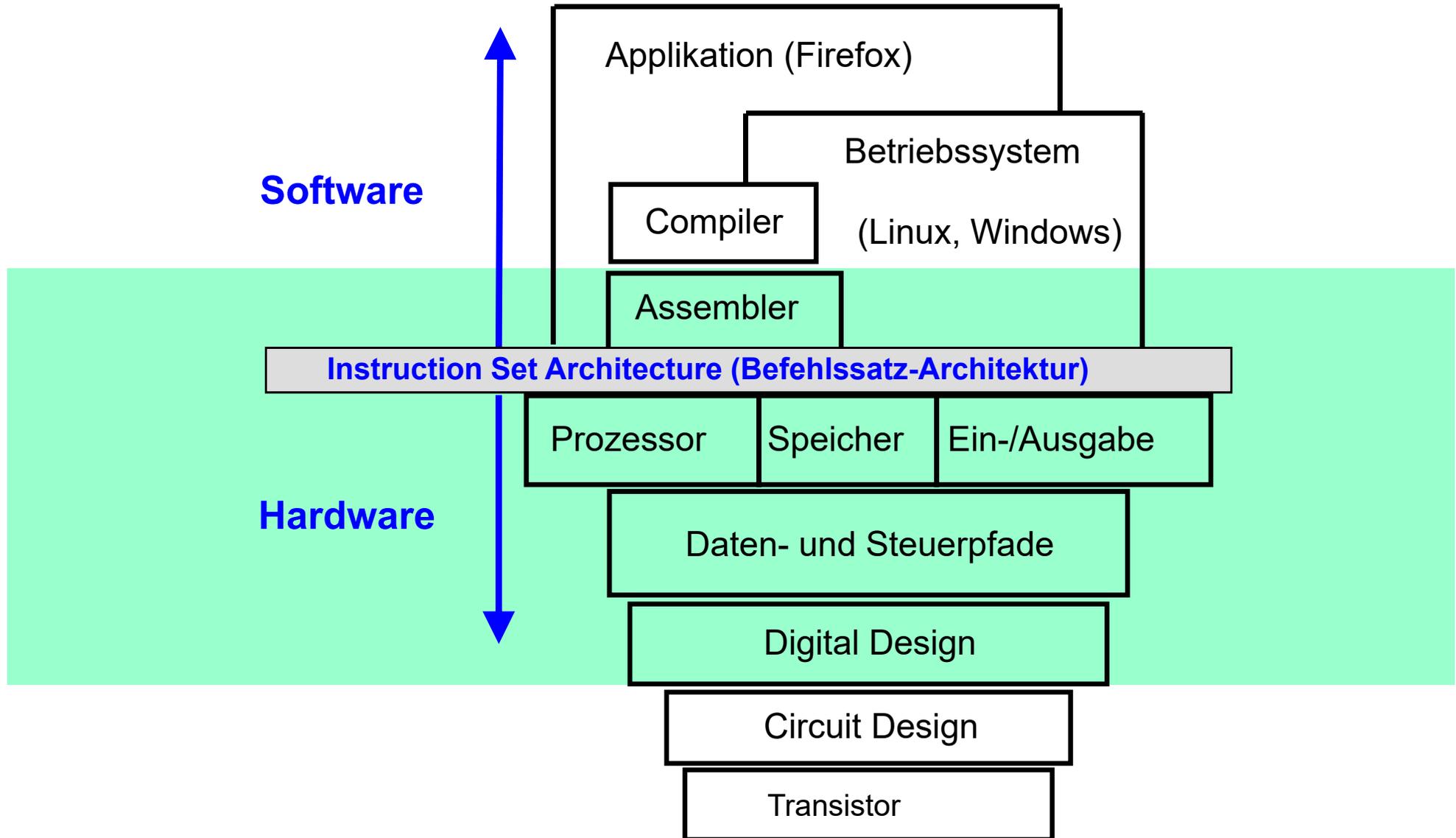


Kapitel 10

Betriebssystemunterstützung

Virtuelle Speicherverwaltung
Schutzmechanismen
Unterbrechungsbehandlung

10.1 Aufgaben des Betriebssystems



10.2 Virtuelle Speicherverwaltung

■ Fragen der Speicherverwaltung

- Beim Austausch von Daten zwischen Haupt- und Hintergrundspeicher ergeben sich drei Probleme:
 - **Einlagerungszeitpunkt:**
 - Wann werden Segmente oder Seiten in den Hauptspeicher eingelagert?
 - **Zuweisungsproblem:**
 - An welche Stelle des Hauptspeichers werden die Seiten oder Segmente eingelagert?
 - **Ersetzungsproblem:**
 - Welche Segmente oder Seiten müssen ausgelagert werden, um Platz für neu benötigte Daten zu schaffen?

10.2 Virtuelle Speicherverwaltung

- **Fragen der Speicherverwaltung**
- **Der Einlagerungszeitpunkt**
 - Gängiges Verfahren: Einlagerung auf Anforderung (Demand Paging bei Seitenverfahren)
 - Hierbei werden Daten eingelagert, sobald auf sie zugegriffen wird, sie sich aber nicht im Hauptspeicher befinden
 - Der Zugriff auf ein nicht im Hauptspeicher vorhandenes Segment oder Seite heißt Segment- oder Seiten-Fehler (segment fault, page fault)

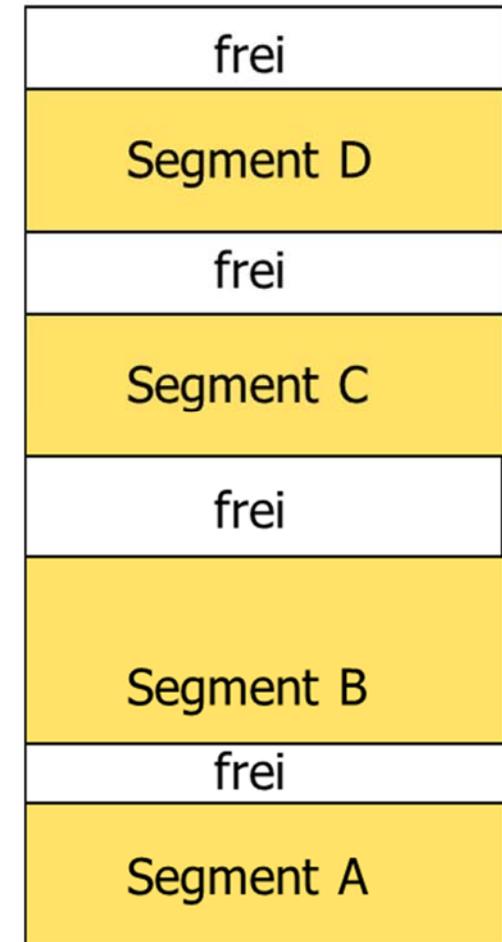
10.2 Virtuelle Speicherverwaltung

- Fragen der Speicherverwaltung
- Das Zuweisungsproblem
 - Segmentierungsverfahren:
 - Hier muss eine ausreichend große Lücke im Hauptspeicher gefunden werden (bei Seitenverfahren sind hingegen alle Seiten gleich groß)
 - Drei Strategien (Betriebssystem):
 - **first-fit**: erste passende Lücke wird genommen
 - **best-fit**: kleinste passende Lücke wird genommen
 - **worst-fit**: größte passende Lücke wird genommen

10.2 Virtuelle Speicherverwaltung

- Fragen der Speicherverwaltung
- Das Zuweisungsproblem
 - Segmentierungsverfahren
 - Problem bei allen drei Strategien
 - Der Speicher zerfällt nach einiger Zeit in belegte und unbelegte Speicherbereiche → externe Fragmentierung (Verschnitt außerhalb der Segmente)
 - Interne Fragmentierung bedeutet: innerhalb der Segmente
 - Die unbelegten Speicherbereiche sind oft zu klein, um weitere Segmente aufnehmen zu können

Hauptspeicher



10.2 Virtuelle Speicherverwaltung

- Fragen der Speicherverwaltung
- Das Zuweisungsproblem
 - Seitenverfahren
 - Hier taucht dieses Problem nicht auf, da alle Seiten gleich groß sind und somit immer “passende Lücken” entstehen → keine externe Fragmentierung
 - Jedoch: Problem der internen Fragmentierung
 - Einheitliche Seitengröße → auf der letzten Seite jedes Programm-Moduls entsteht mit hoher Wahrscheinlichkeit ein ungenutzter Leerraum

10.2 Virtuelle Speicherverwaltung

- Fragen der Speicherverwaltung
- Das Ersetzungsproblem
 - Segmentierungsverfahren:
 - Meist wird die Anzahl der gleichzeitig von einem Prozess benutzbaren Segmente limitiert:
 - → bei Einlagerung eines neuen Segments wird ein zuvor für diesen Prozess benutztes Segment ausgelagert
 - Es ist jedoch auch eine der im folgenden für Seitenwechsel-Verfahren beschriebenen Methoden möglich

10.2 Virtuelle Speicherverwaltung

- Fragen der Speicherverwaltung
- Das Ersetzungsproblem
 - Bei Seitenwechselfverfahren
 - 5 gängige Strategien zum Ersetzen einer Seite:
 - FIFO (first-in-first-out): die sich am längsten im Hauptspeicher befindende Seite wird ersetzt
 - LIFO (last-in-first-out): die zuletzt eingelagerte Seite wird ersetzt
 - LRU (least recently used): die Seite, auf die am längsten nicht zugegriffen wurde, wird ersetzt
 - LFU (least frequently used): die seit ihrer Einlagerung am seltensten benutzte Seite wird ersetzt
 - LRD (least reference density): Mischung aus LRU und LFU; die Seite mit der geringsten Zugriffsdichte (Anzahl Zugriffe / Einlagerungszeitraum) wird ersetzt
 - Daneben werden bevorzugt solche Seiten ersetzt, die nicht verändert wurden → kein Rückschreiben der geänderten Seite erforderlich

10.2 Virtuelle Speicherverwaltung

■ Zusammenfassung

■ Segmentierung (variable Größe)

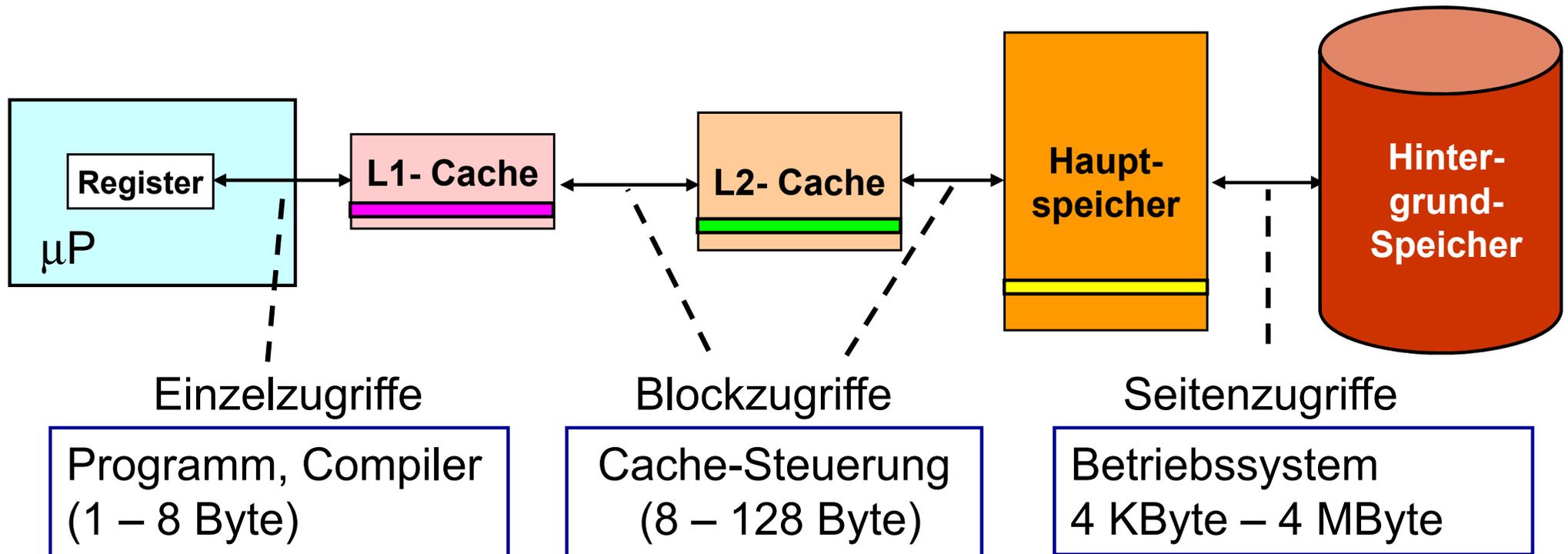
- logische Abbildung einer Programmstruktur
- geringer Datentransfer
- umfangreicher Datentransfer beim Ein-/Auslagern
- externe Fragmentierung

■ Seitenwechsel-Verfahren (feste Größe)

- geringerer Verwaltungsaufwand
- bessere Hauptspeicherauslastung
- häufiger Datentransfer
- interne Fragmentierung

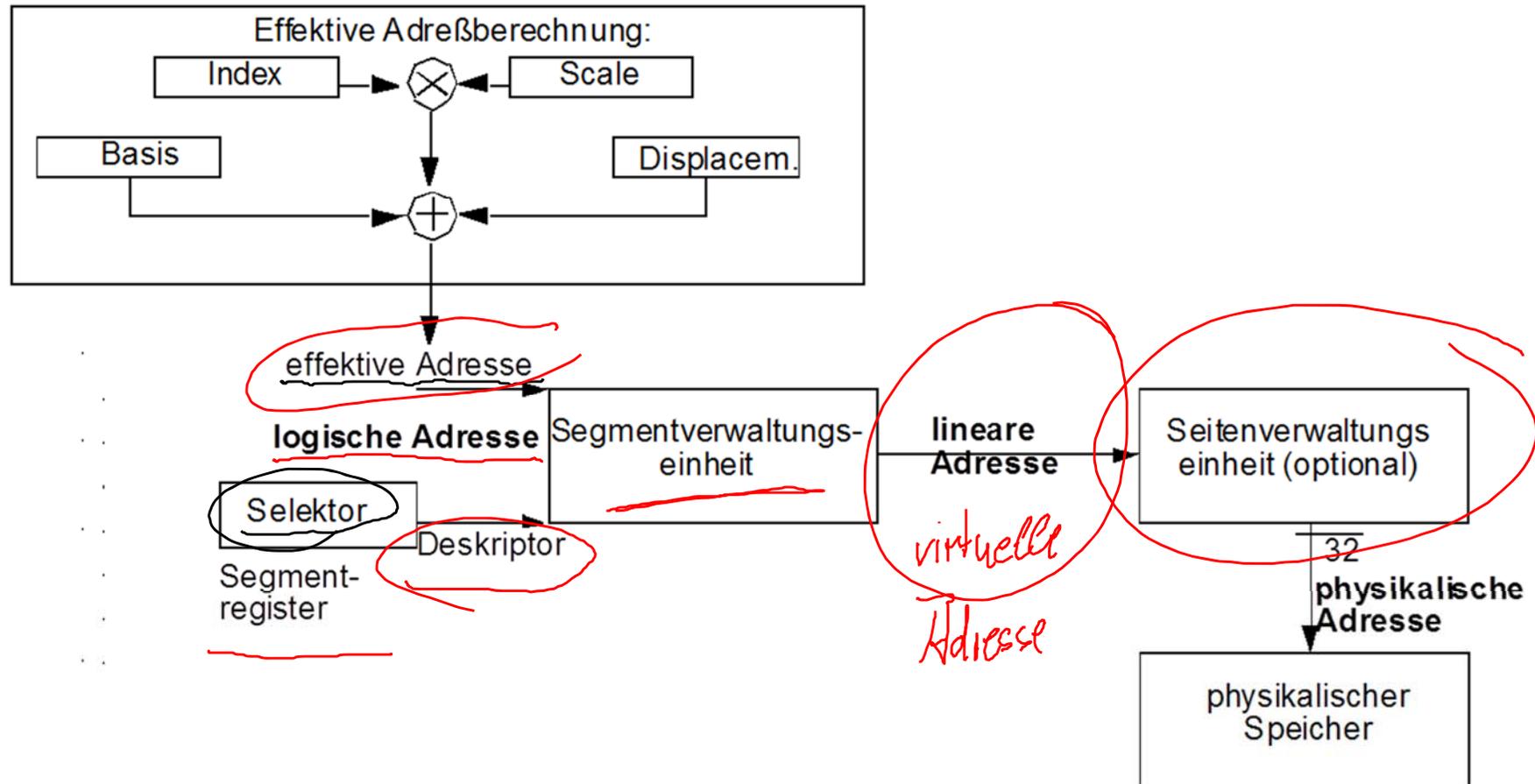
10.2 Virtuelle Speicherverwaltung

■ Speicherhierarchie



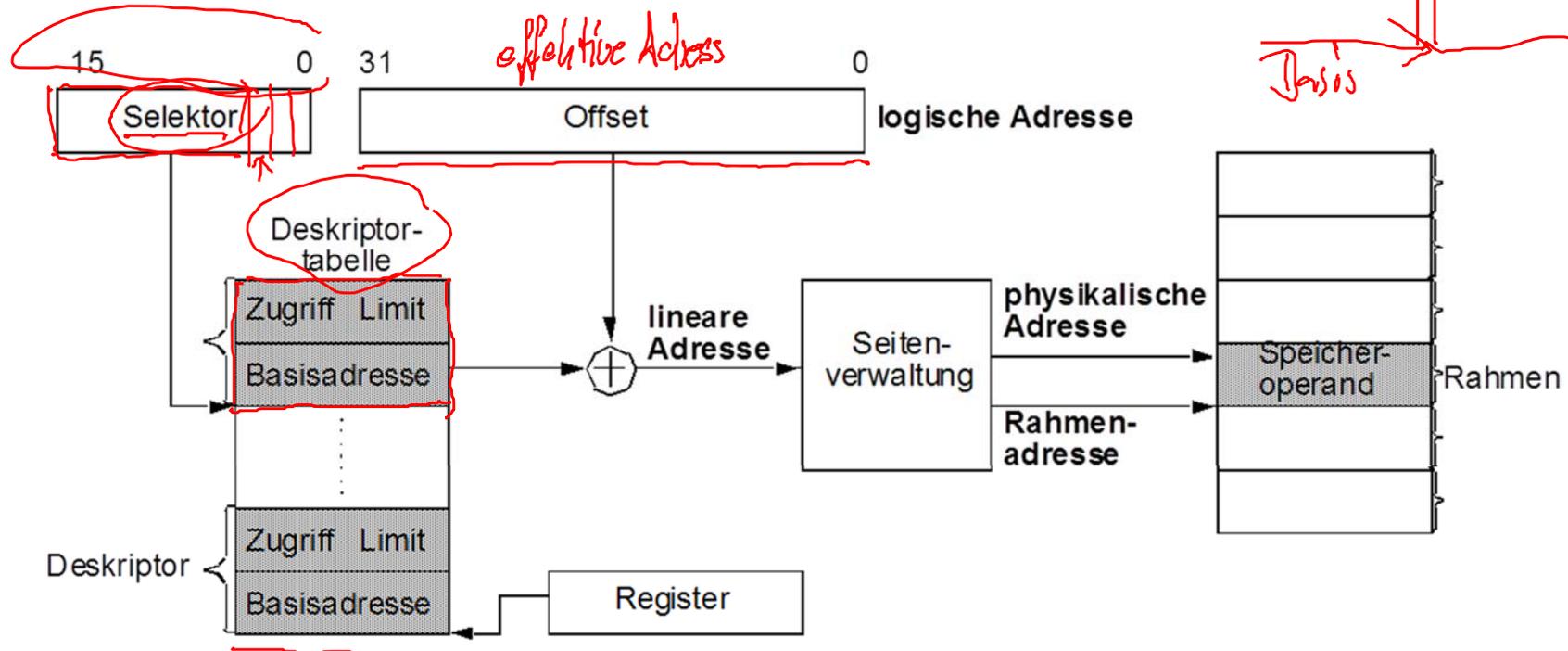
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)



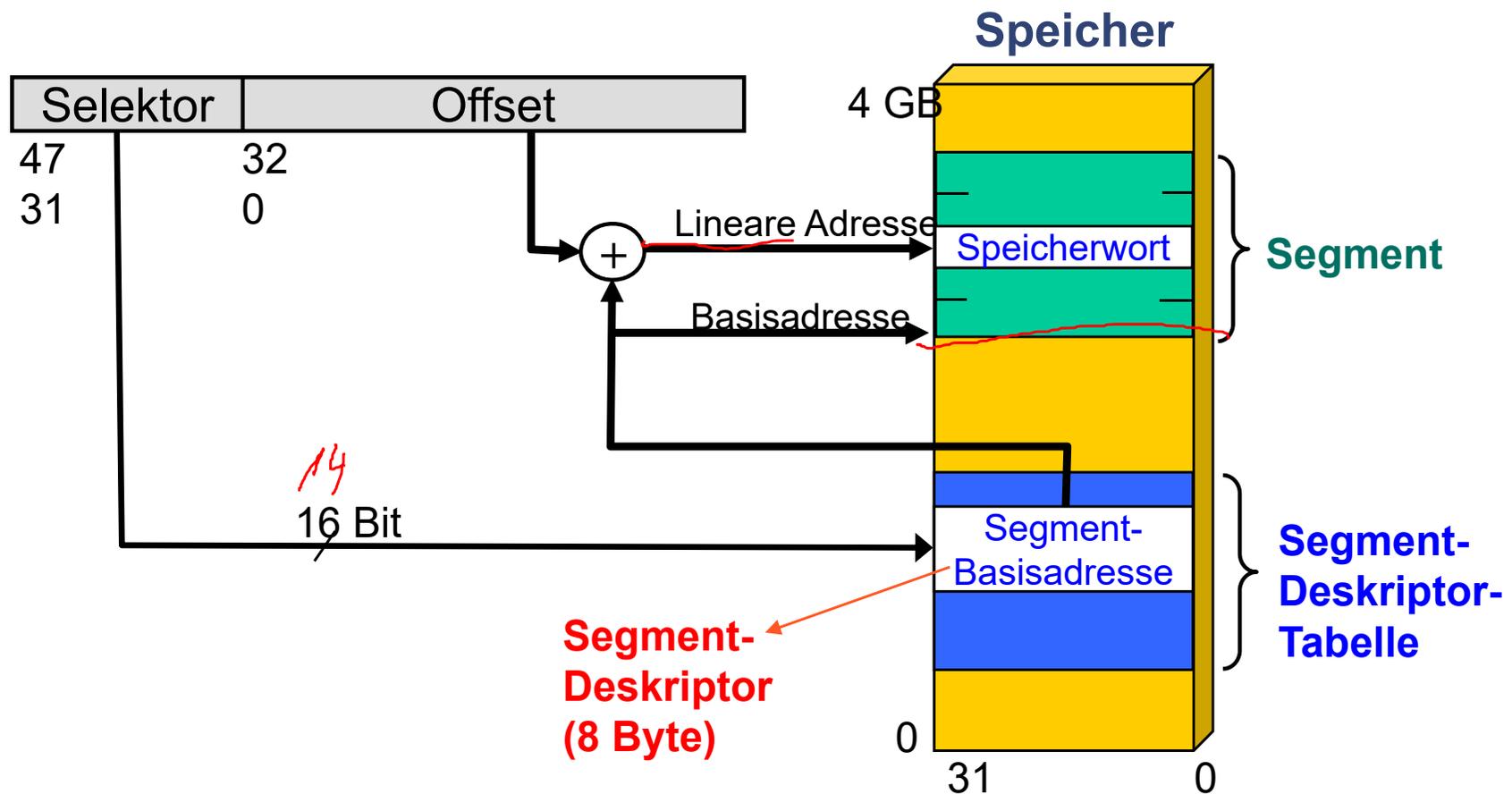
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)
 - Segmentverwaltung
 - Segmentierung mittels Erweiterung der virtuellen Adresse
 - 64 TBytes (2^{46}) logischer Adressraum
 - Pro Task (Prozess) maximal 16K (2^{14}) Selektoren
 - Pro Segment max. 4GBytes



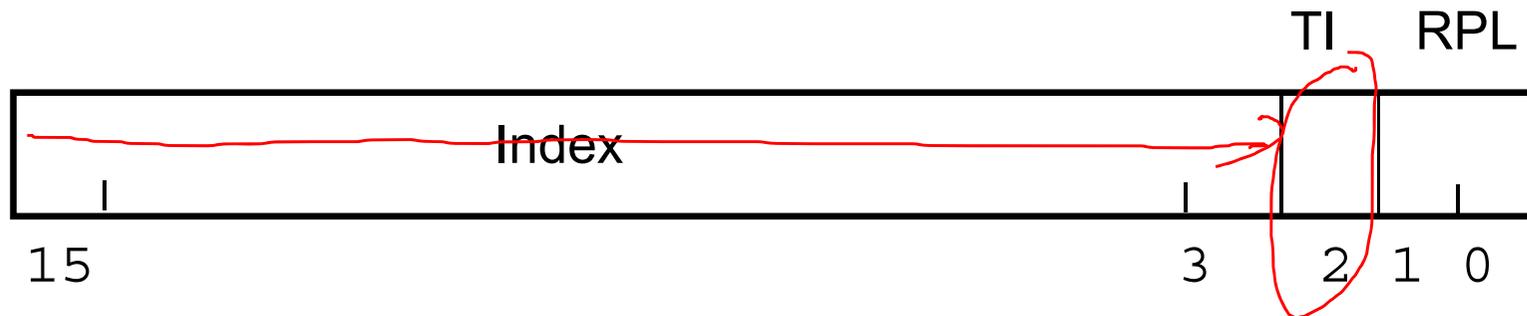
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)
 - Segmentverwaltung
 - Segmentierung mittels Erweiterung der virtuellen Adresse



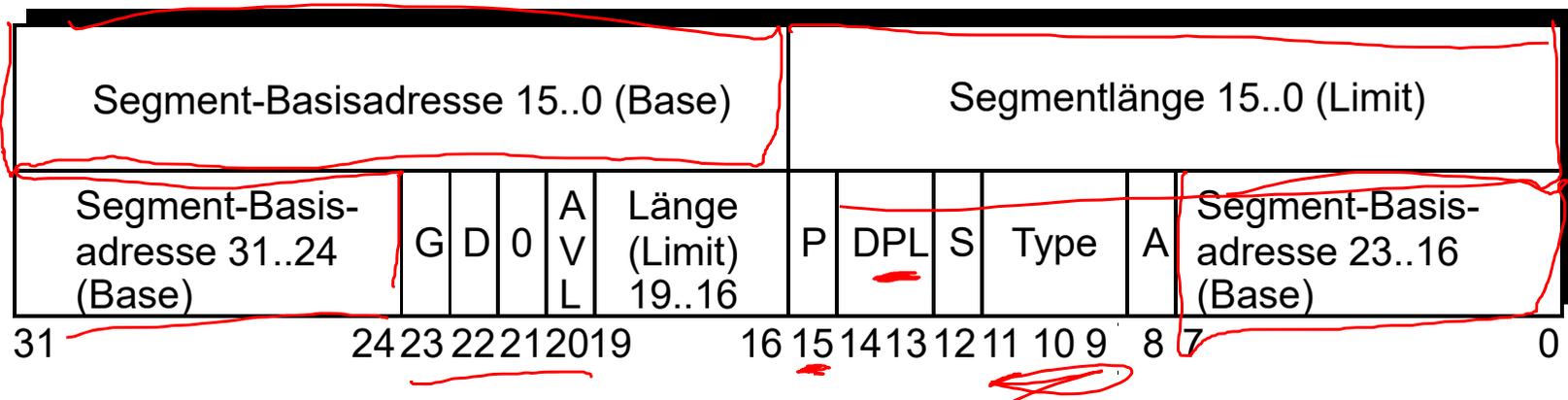
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)
 - Segmentverwaltung
 - Segmentselektor
 - Indexfeld und TI (Tabellenindikator) selektieren einen Eintrag in der Segment-Deskriptor-Tabelle
 - RPL-Bits (Requested Privilege Level) geben eine Privileg-Ebene an, die ein Befehl besitzen muss, um auf das gewünschte Segment zugreifen zu dürfen



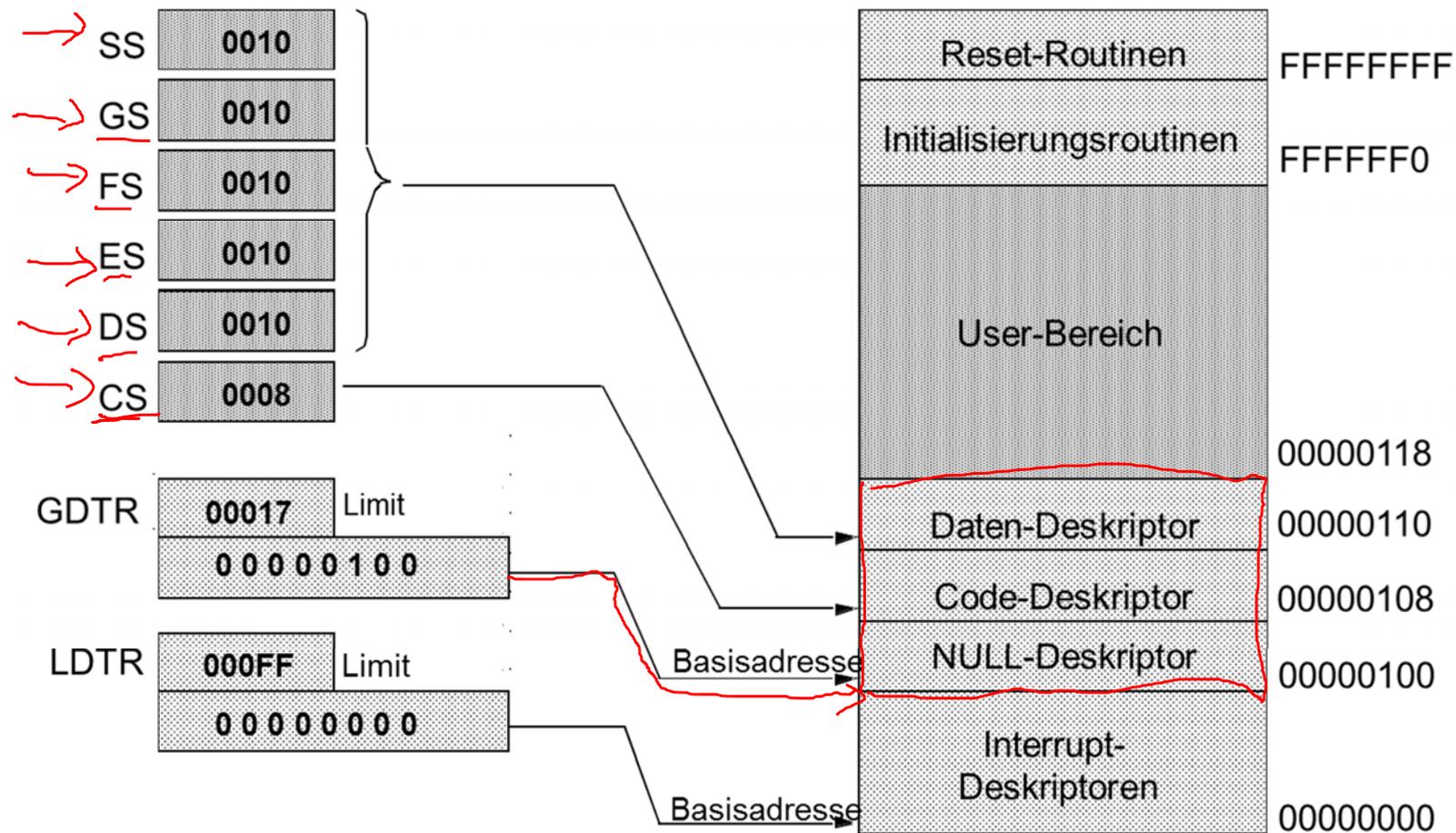
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)
 - Segmentverwaltung
 - Segmentdeskriptoren
 - Jeder Segmentdeskriptor beschreibt das zugehörige Segment durch folgende Attribute:
 - Segment-Basisadresse (base address)
 - Segment-Größe in Bytes (limit)
 - Zugriffsrechte auf das Segment (access rights) zur Realisierung von Schutzmechanismen



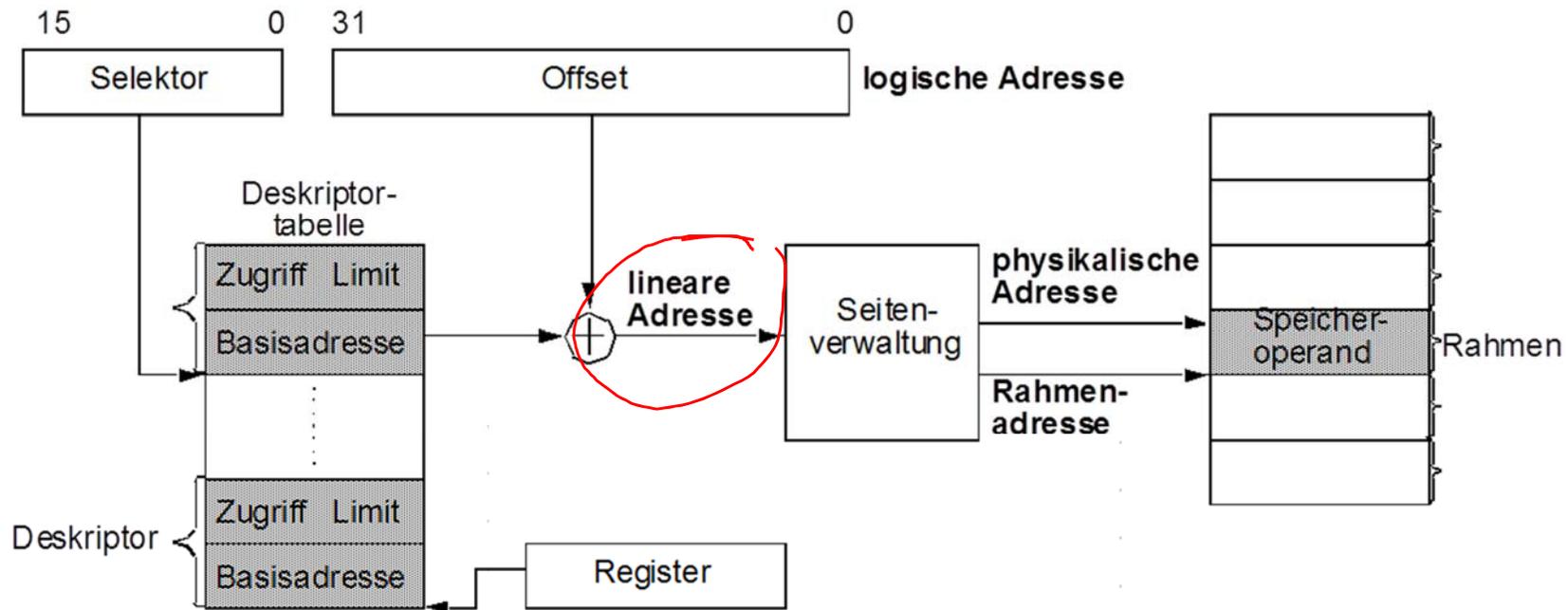
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)
 - Segmentverwaltung: Initialisierung eines einfachen Systems



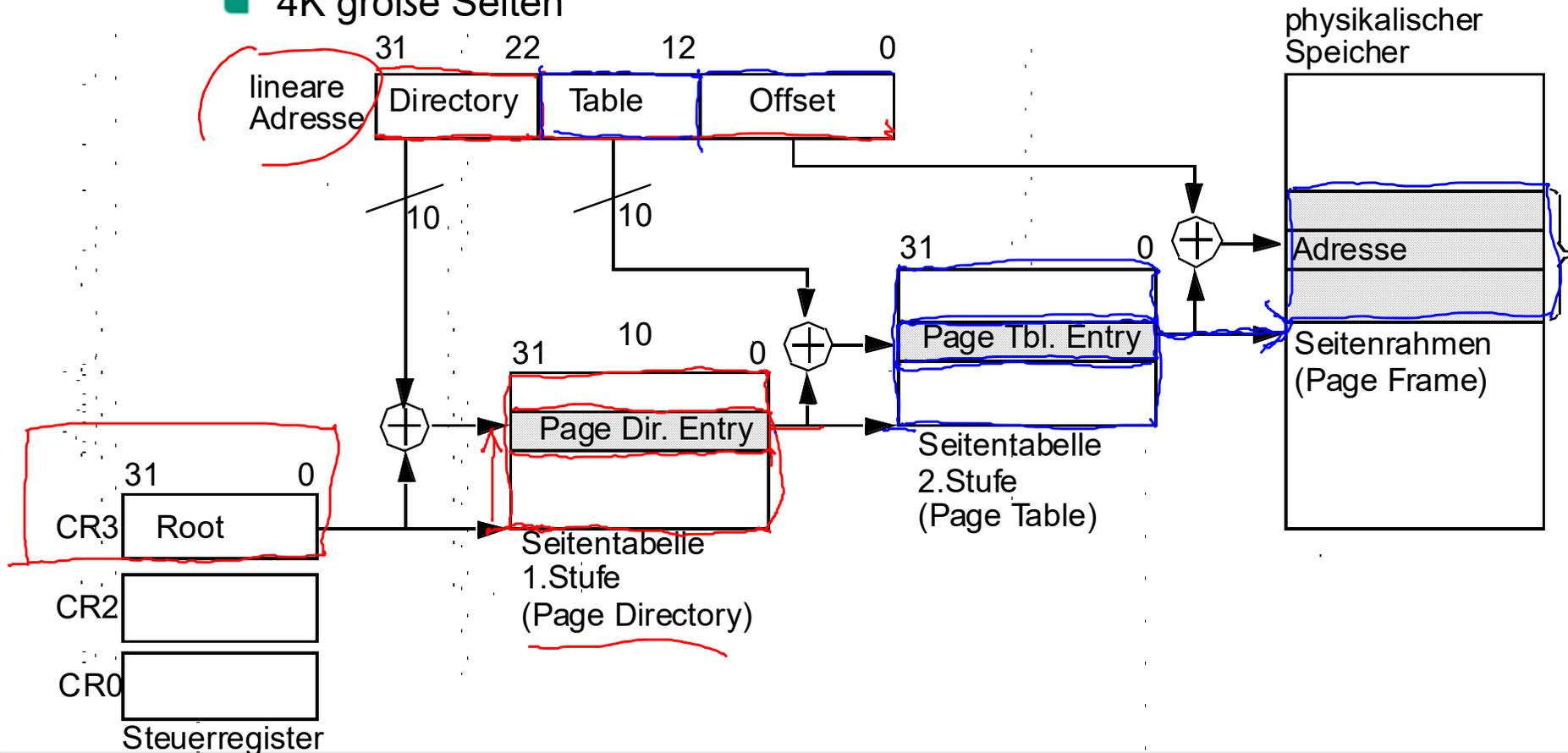
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)
 - Seitenverwaltung



10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)
 - Seitenverwaltung
 - Adressumsetzung mit 32-Bit physikalischer Adressierung
 - 4K große Seiten

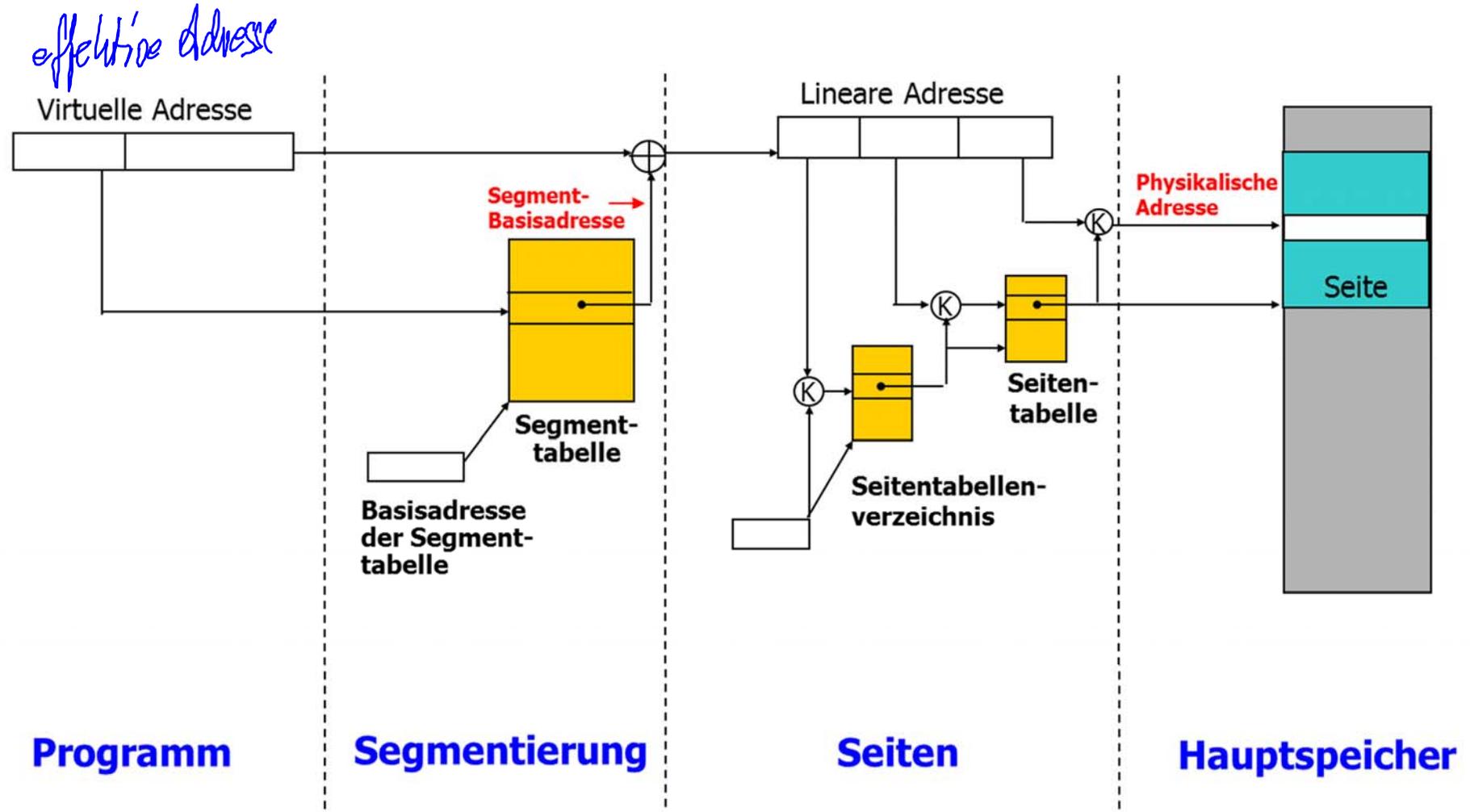


10.2 Virtuelle Speicherverwaltung

- **Umsetzung von virtuellen in physikalische Adressen**
- **Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)**
 - **Seitenverwaltung**
 - Seitentabellenverzeichnis → Seitentabellen → Seiten
 - Seitentabellenverzeichnis, Seitentabellen und Seiten sind jeweils 4 KBytes groß und jeder Tabellen-Eintrag umfasst 4 Byte
 - In den Seitentabellen und -verzeichnis sind jeweils 1024 Einträge enthalten
 - Lineare Adresse wird in drei Teilen zerlegt
 - Die höchstwertigen 10 Bits selektieren einen Eintrag im Seitentabellen-Verzeichnis, dessen Basisadresse in einem speziellen Systemregister (CR3) abgelegt ist
 - Die nächsten 10 Bits der linearen Adresse selektieren einen der 1024 Einträge aus der Seitentabelle; in diesem Eintrag steht die Basisadresse einer Seite
 - Die niedrigstwertigen 12 Bits werden als Offset zur Seitenadresse addiert, um die endgültige, physikalische Adresse zu erhalten

10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
- Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)

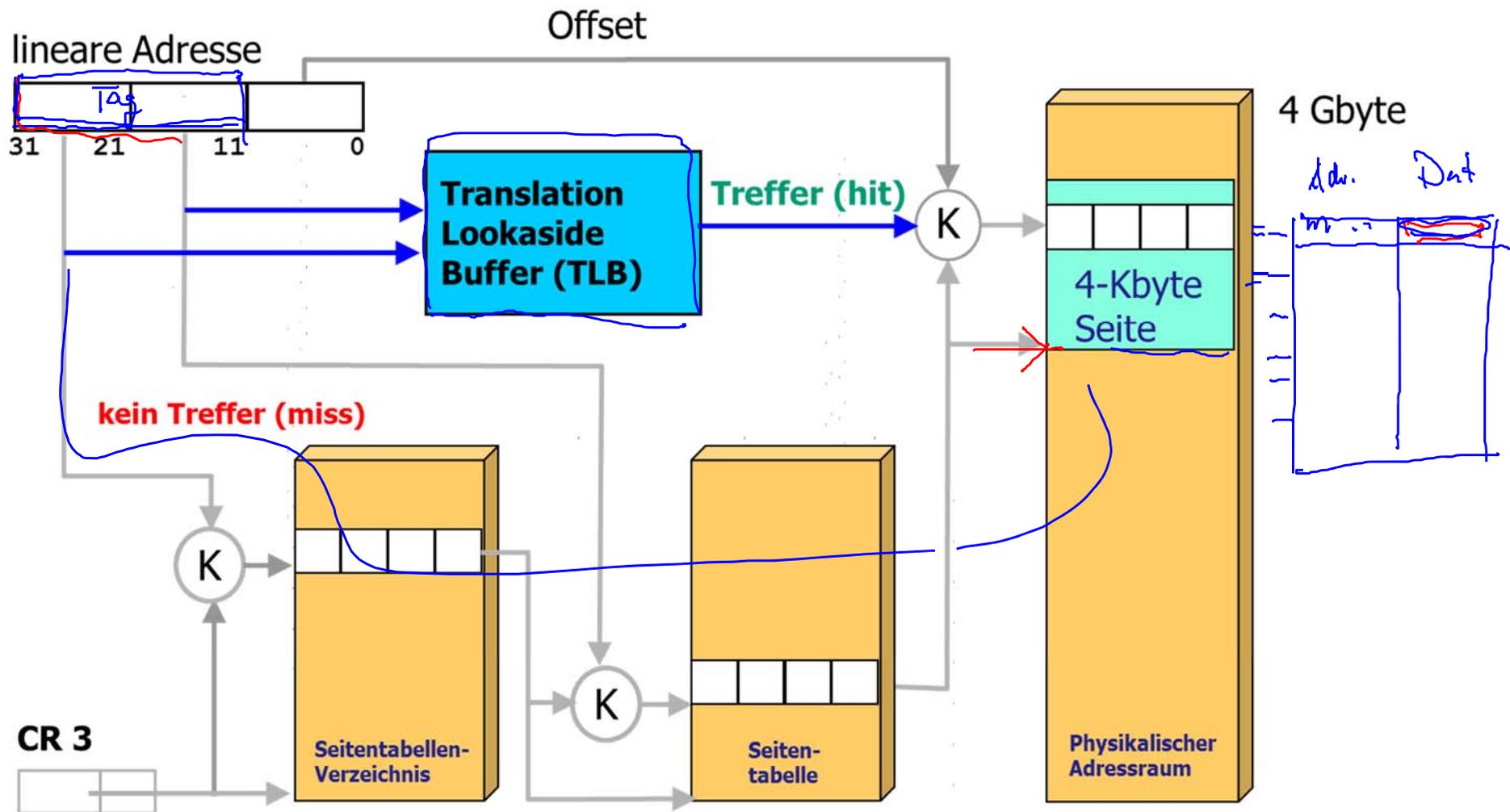


10.2 Virtuelle Speicherverwaltung

- **Umsetzung von virtuellen in physikalische Adressen**
- **Fallstudie IA-32 (Intel 32-Bit Befehlssatzarchitektur)**
 - Sowohl bei segmentorientierter wie bei seitenorientierter Speicherverwaltung gilt:
 - Befindet sich eine Seite oder ein Segment nicht im Hauptspeicher, so löst der Prozessor eine Unterbrechung aus, um die Seite oder das Segment durch das Betriebssystem zu laden (Seiten- oder Segmentfehltalarm)
 - **Erkennung eines Segmentfehlers:**
 - Bit im Segment-Deskriptor zeigt an, ob das Segment im Hauptspeicher ist oder nicht
 - **Erkennung eines Seitenfehlers:**
 - Seite oder Seitentabelle befindet sich nicht im Hauptspeicher (Seitenfehler)
 - Spezielles Kennungsbit im Seitentabellen-Verzeichnis (Seitentabellenfehler)

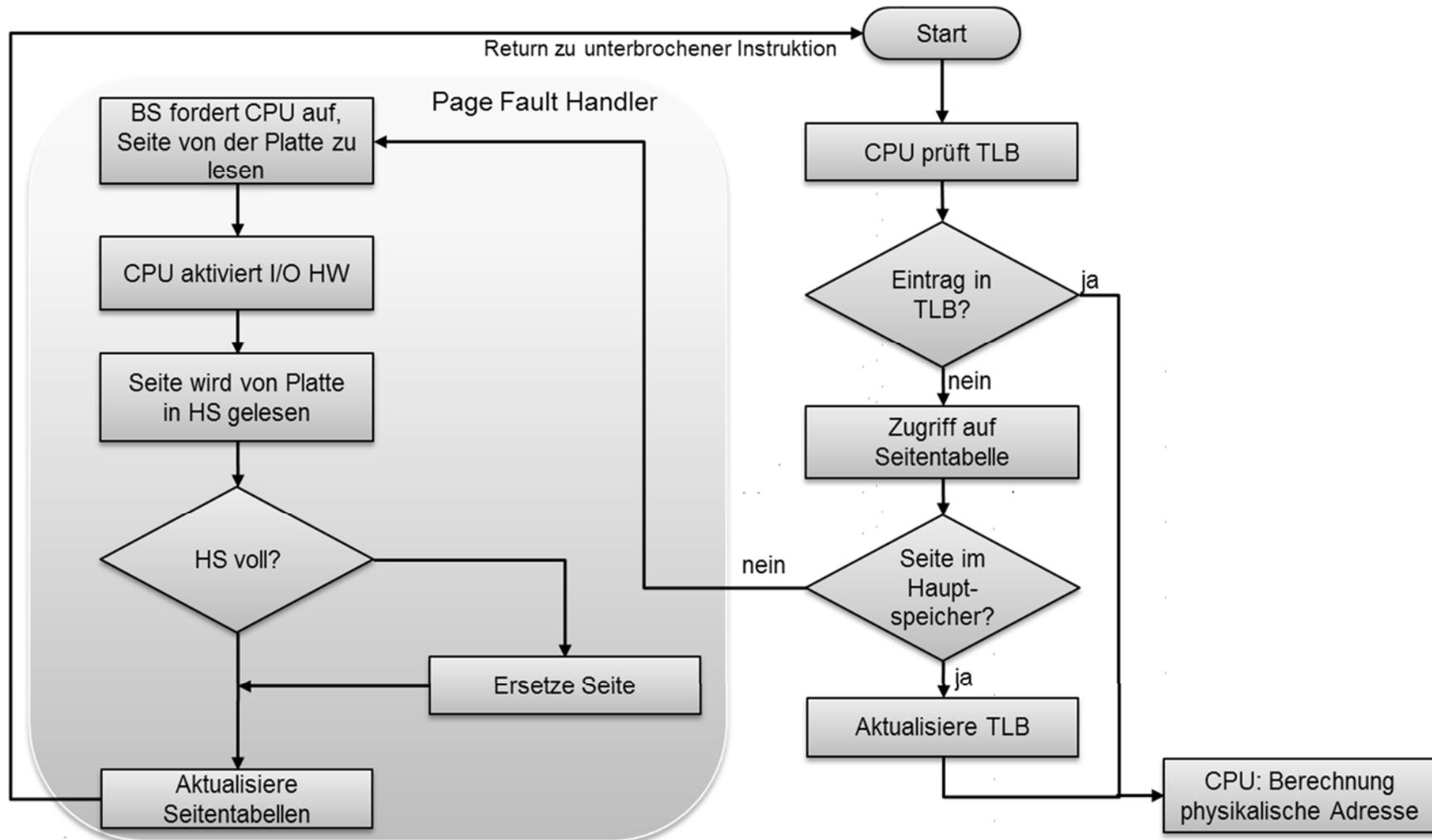
10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
 - Beschleunigung durch Translation Lookaside Buffer (TLB)



10.2 Virtuelle Speicherverwaltung

- Umsetzung von virtuellen in physikalische Adressen
 - Beschleunigung durch Translation Lookaside Buffer (TLB)



10.2 Virtuelle Speicherverwaltung

■ Virtuelle und reale Cache-Adressierung

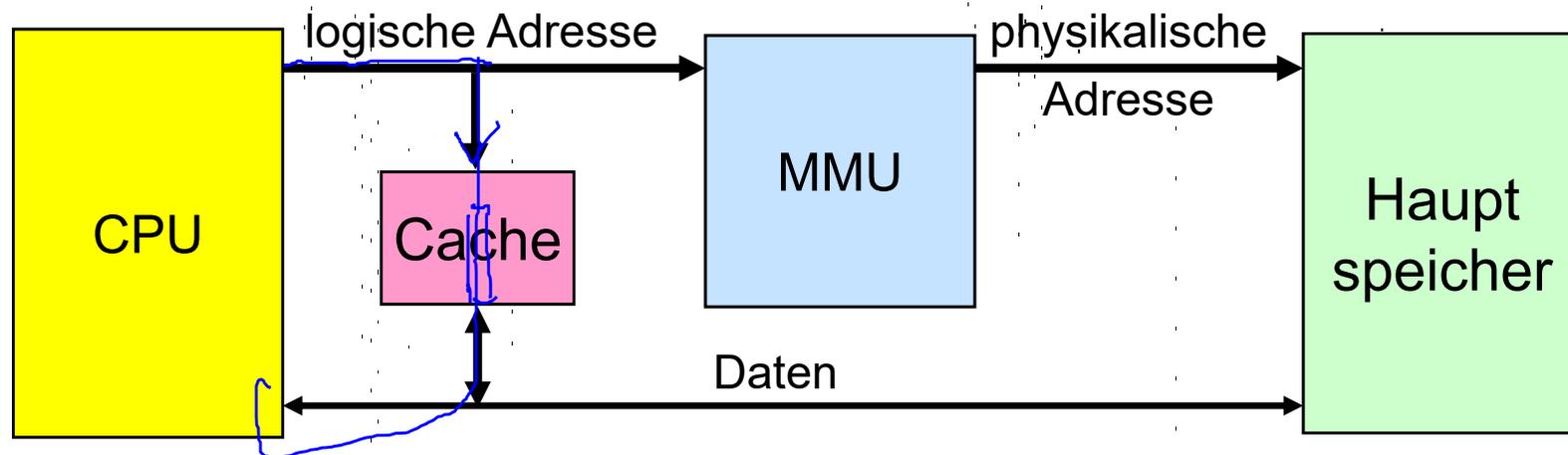
- Bei Mikroprozessoren mit Speicherverwaltungseinheit gibt es drei Möglichkeiten, den Cache zu adressieren:
 1. Mit der virtuellen, vom Prozessor erzeugten Adresse
 2. Mit der realen, von der Speicherverwaltungseinheit erzeugten
 3. Mit je einem Teil der virtuellen und der realen Adresse

10.2 Virtuelle Speicherverwaltung

■ Virtuelle und reale Cache-Adressierung

■ Virtuelle Cache-Adressierung:

- Cache wird zwischen CPU und MMU gelegt
- Die höherwertigen Bits der logischen Adressen als Tags abgelegt



■ Vorteil:

- bei Cache-Hit wird die MMU nicht benötigt, d. h. keine Verzögerung durch die Adressberechnung der MMU

10.2 Virtuelle Speicherverwaltung

■ Virtuelle und reale Cache-Adressierung

■ Virtuelle Cache-Adressierung:

■ Taskwechsel:

- Mehrere voneinander unabhängige Tasks überdecken sich teilweise oder ganz mit ihren virtuellen Adressen und teilen sich den Hauptspeicher
- Ein und dieselbe virtuelle Adresse hat bezogen auf die unterschiedlichen Tasks unterschiedliche reale Entsprechungen

■ Problem:

- führt zu Inkohärenzen, da der Cache virtuell adressiert wird und nicht berücksichtigen kann, dass die MMU taskabhängig unterschiedliche reale Adresse erzeugt

■ Lösung:

- Bei Taskwechsel „cache-clear“ oder „cache-flush“

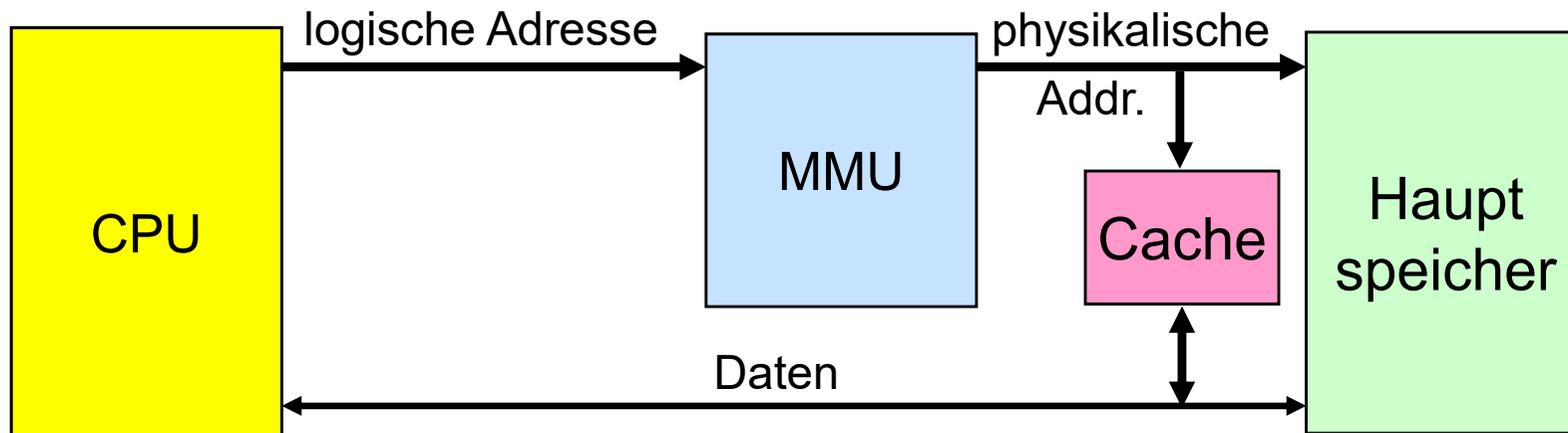


10.2 Virtuelle Speicherverwaltung

■ Virtuelle und reale Cache-Adressierung

■ Reale Cache-Adressierung

- wird zwischen MMU und Speicher gelegt
- Die höherwertigen Bits der physikalischen Adressen werden als Tags abgelegt



■ Vorteil:

- physikalische Adresse ist i.A. viel kleiner als die logische Adresse □
weniger Bits müssen als Tag gespeichert werden

■ Nachteil :

- Cache-Hit ist langsamer, da die Adresse vor dem Cache Zugriff erst übersetzt werden muss