Übung 6

Karlsruher Institut für Technologie

Cache Speicher - Fortsetzung

- Aufgaben

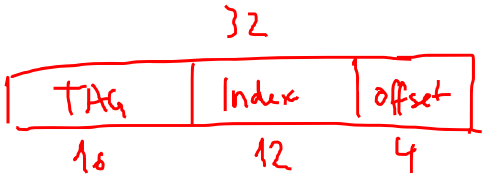
Virtuelle Speicherverwaltung

Aufgabe 5

Bei einem Cache-Speicher mit einer Speicherkapazität von **128 Kbyte** ist die Hauptspeicheradresse in ein **16 Bit** Tag-Feld, ein **12 Bit** Index-Feld und ein **4 Bit** Byte-Offset unterteilt.

1. Bestimmen Sie die Blockgröße in Bytes.
2. Wie viele Einträge besitzt der Cache-Speicher?
3. Wie ist der Cache-Speicher organisiert?

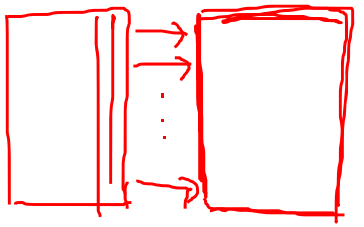
Aufgabe 5.1



Cache-Kapazität = **128 Kbyte**
16 Bit Tag-Feld, **12 Bit** Index-Feld, **4 Bit** Byte-Offset

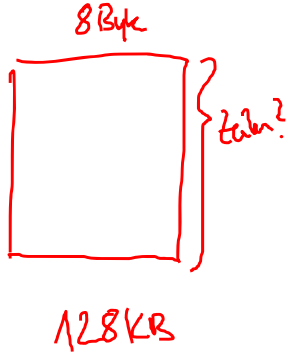
Blockgröße:

$2^4 \Rightarrow 16$ Byte



Aufgabe 5.2

Cache-Kapazität = **128 Kbyte**
16 Bit Tag-Feld, **12 Bit** Index-Feld, **4 Bit** Byte-Offset



Anzahl der Einträge im Cache:

$$\frac{128\ \text{KByte}}{16\ \text{Byte}} = 8\text{K} = 8192\ \text{Zeilen}$$

Aufgabe 5.3

Cache-Kapazität = **128 Kbyte**
16 Bit Tag-Feld, **12 Bit** Index-Feld, **4 Bit** Byte-Offset

Organisation:

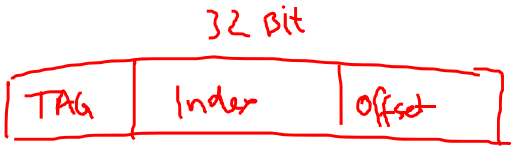
PM
AV
An

$2^{12} = 4k = 4096$ Sätze

$\frac{\text{Zeilen}}{\text{Sätze}} = \frac{8k}{4k} = 2$

2-fach satzassoziativen Cache

Aufgabe 6

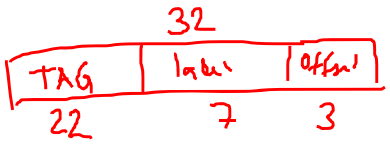


Es soll ein 3-fach satzassoziativer *(3-way set associative cache)* Cache-Speicher mit **128 Sätzen** und einer Blockgröße von **8 Byte** realisiert werden.

Nehmen Sie an, dass die Hauptspeicheradresse 32 Bit breit ist. Zur Verwaltung eines Cacheblocks wird zwei Statusbits (Valid-Bit: V und Dirty Bit: D) verwendet.

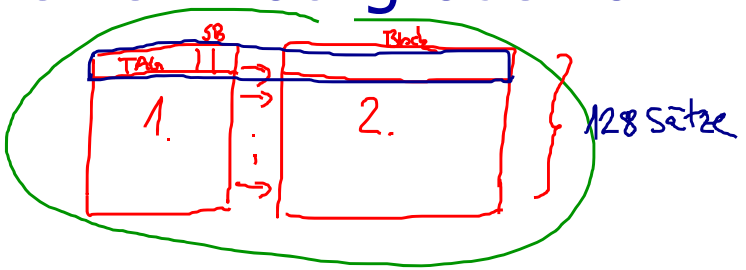
Bestimmen Sie den insgesamt erforderlichen Speicherbedarf zur Realisierung dieses Cache-Speichers.

Lösung 6



3-fach-assoziativer *(3-way set associative cache)*
Cache-Speicher mit **128 Sätzen** und einer Blockgröße von **8 Byte**

$ld(8) = 3 \quad 2^3 = 8$
$ld(128) = 7 \quad 2^7 = 128$



Speicherbedarf für eine Zeile:

Tag-Länge + Anzahl der Statusbits + Blockgröße

22 Bit + 2 Bit + 8 Byte

= (22 + 2) Bit + 8 Byte = 3 + 8 Byte = 11 Byte

384
384
4224

Speicherbedarf insgesamt:

11 Byte/Zeile · 3 ~~Zeilen~~/Satz · 128 ~~Sätze~~ = 11 · 3 · 128 Byte = 11 · 384 = 4224 Byte

Aktualisierungsstrategien

- Write-through
  - No-Write Allocation: Bei einem Write-Miss wird nur der Hauptspeicher und nicht der Cache aktualisiert
  - Write-Allocation: Bei einem Write-Miss wird der Hauptspeicher und der Cache aktualisiert
- Write-back: Bei Schreibzugriffen wird nur der Cache und nicht der Hauptspeicher aktualisiert.

Aktualisierungsstrategien

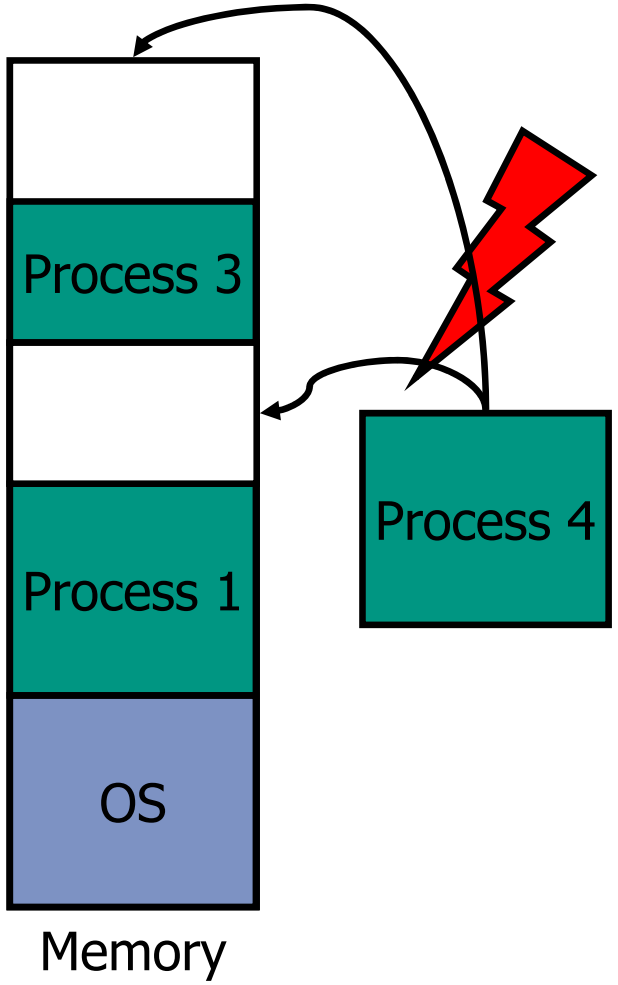
Cache-Operation	write-through no write-allocation	write-through write allocation	write-back
Read-Hit	Cache-Datum → CPU	Cache-Datum → CPU	Cache-Datum → CPU
Read-Miss	Sp.-Block, Tag → Cache Sp.-Datum → CPU V = 1	Sp.-Block, Tag → Cache Sp.-Datum → CPU V = 1	Cache-Zeile → Speicher Sp.-Block, Tag → Cache Sp.-Datum → CPU V = 1, D = 0
Write-Hit	CPU-Datum → Cache, Speicher	CPU-Datum → Cache, Speicher	CPU-Datum → Cache D = 1
Write-Miss	CPU-Datum → Speicher	Sp.-Block, Tag → Cache V =1 CPU-Datum → Cache, Speicher	Cache-Zeile → Speicher Sp.-Block, Tag → Cache V = 1 CPU-Datum → Cache D = 1

Nur dann erforderlich, wenn der Block gültig und Dirty ist (V=1, D=1)

Virtuelle Speicherverwaltung

Speicherverwaltung

- Hauptspeicherkapazität ist begrenzt
  - Prozess benötigt mehr Speicher als der Hauptspeicher
  - Mehr aktive Prozesse als der Hauptspeicher “vertragen” kann.
- Effiziente Schutzmechanismen



Speicherverwaltung

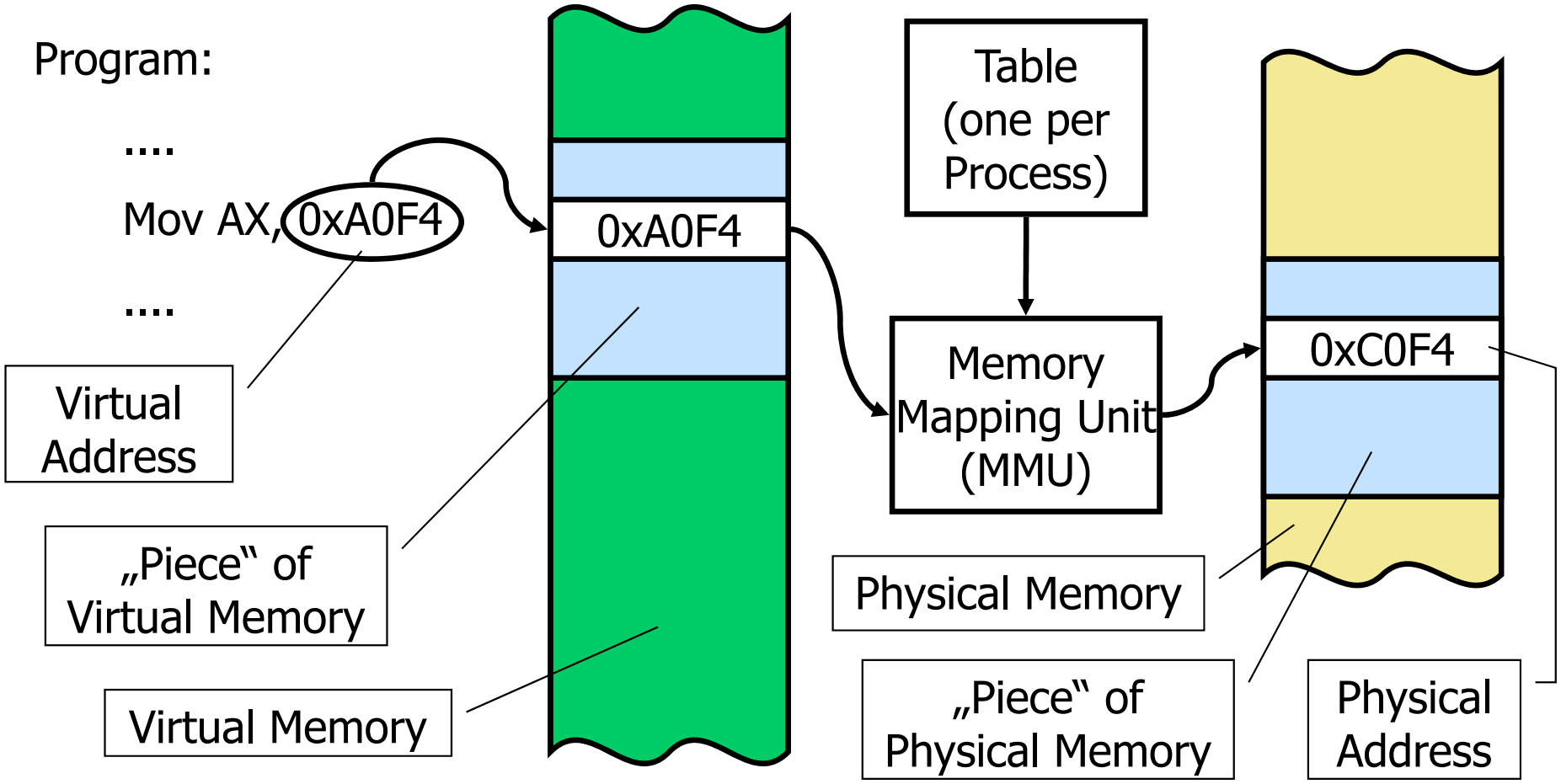
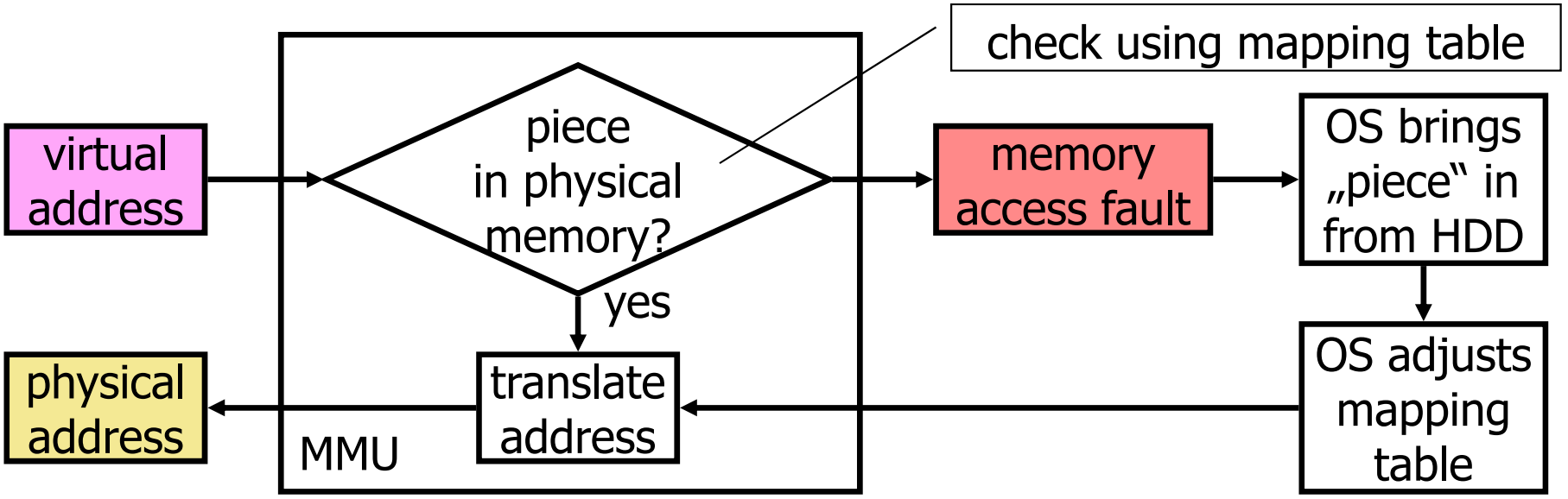
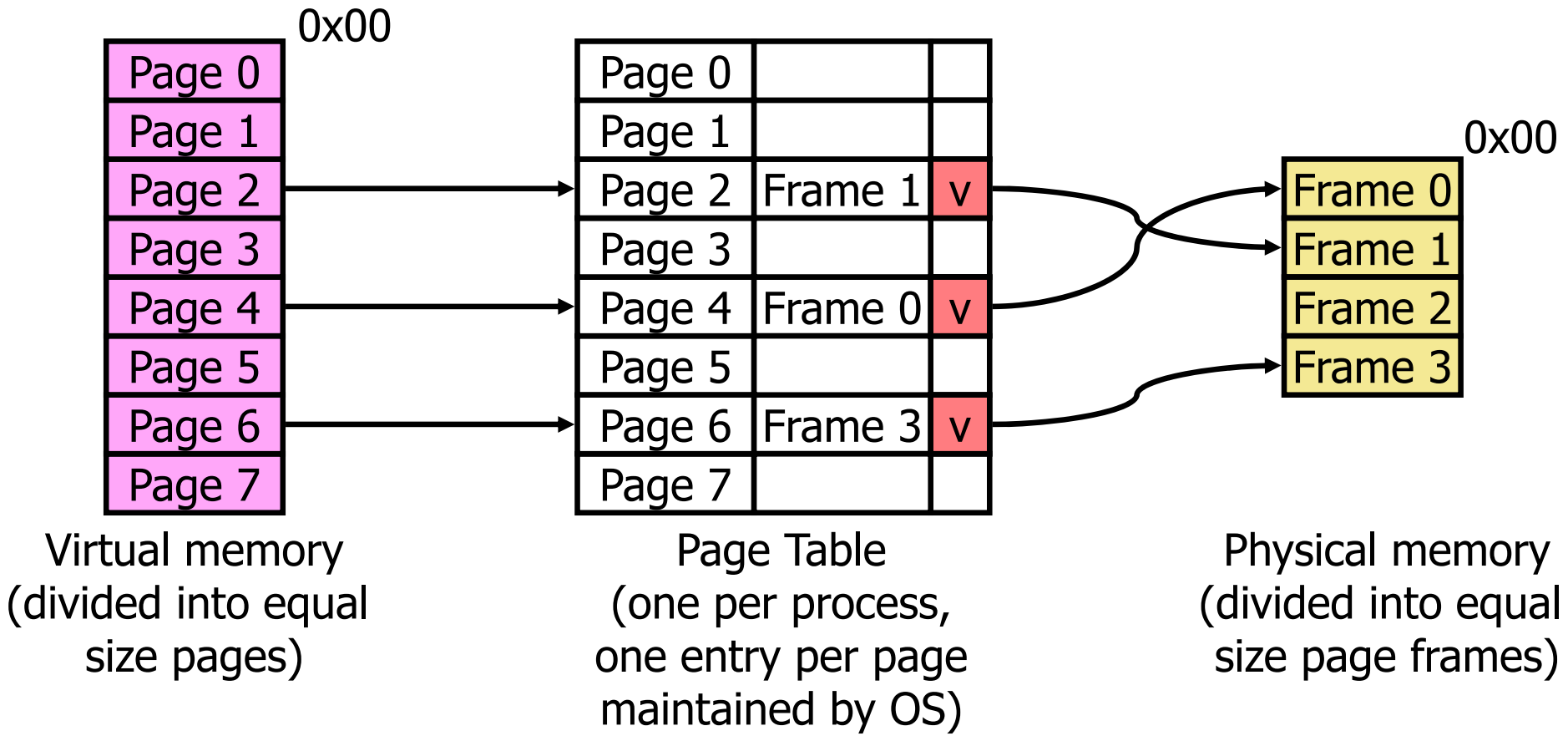


Abbildung: virtuell → physikalisch

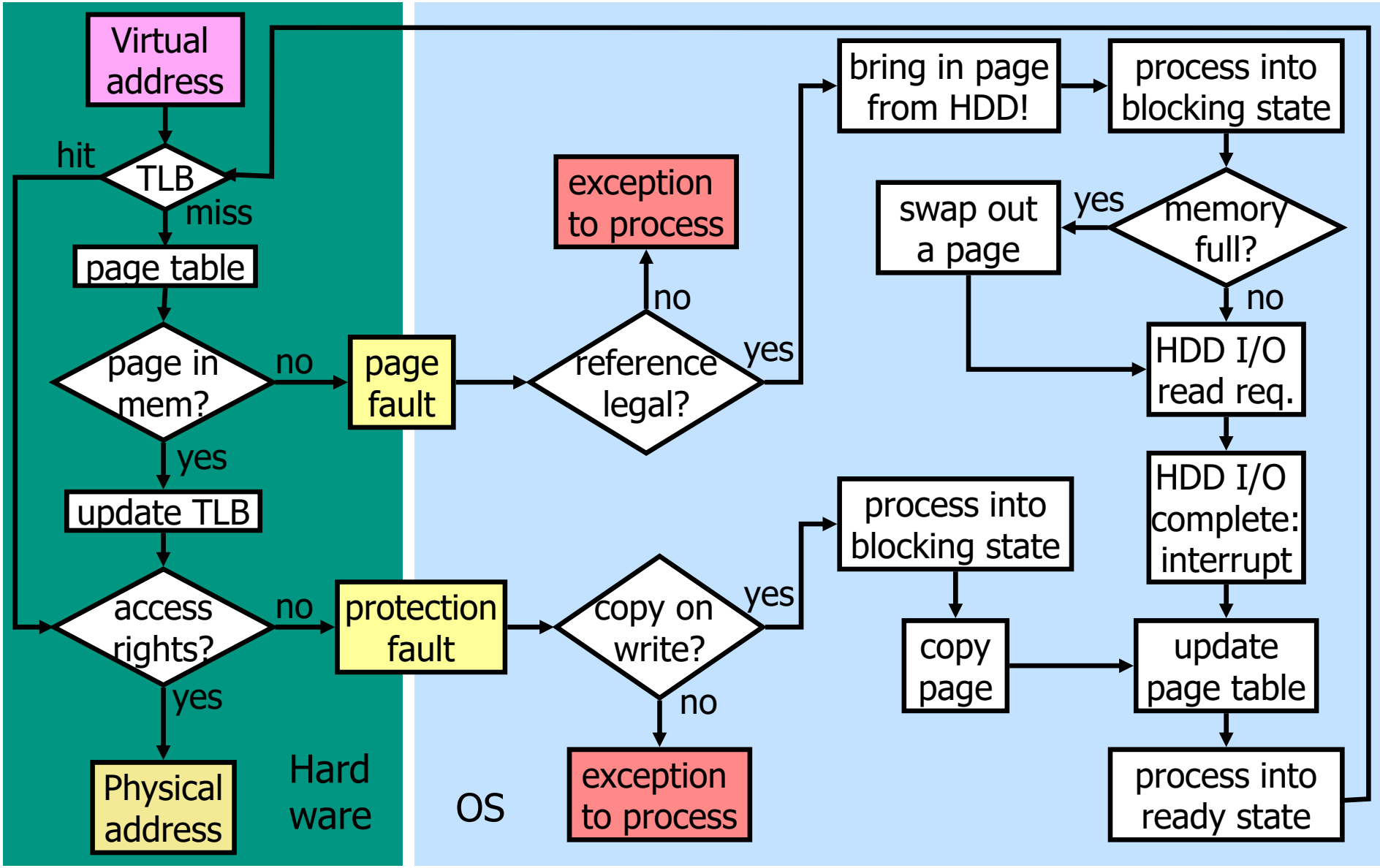
- Jeder Prozess hat seinen virtuellen Adressraum und seine Abbildungstabelle



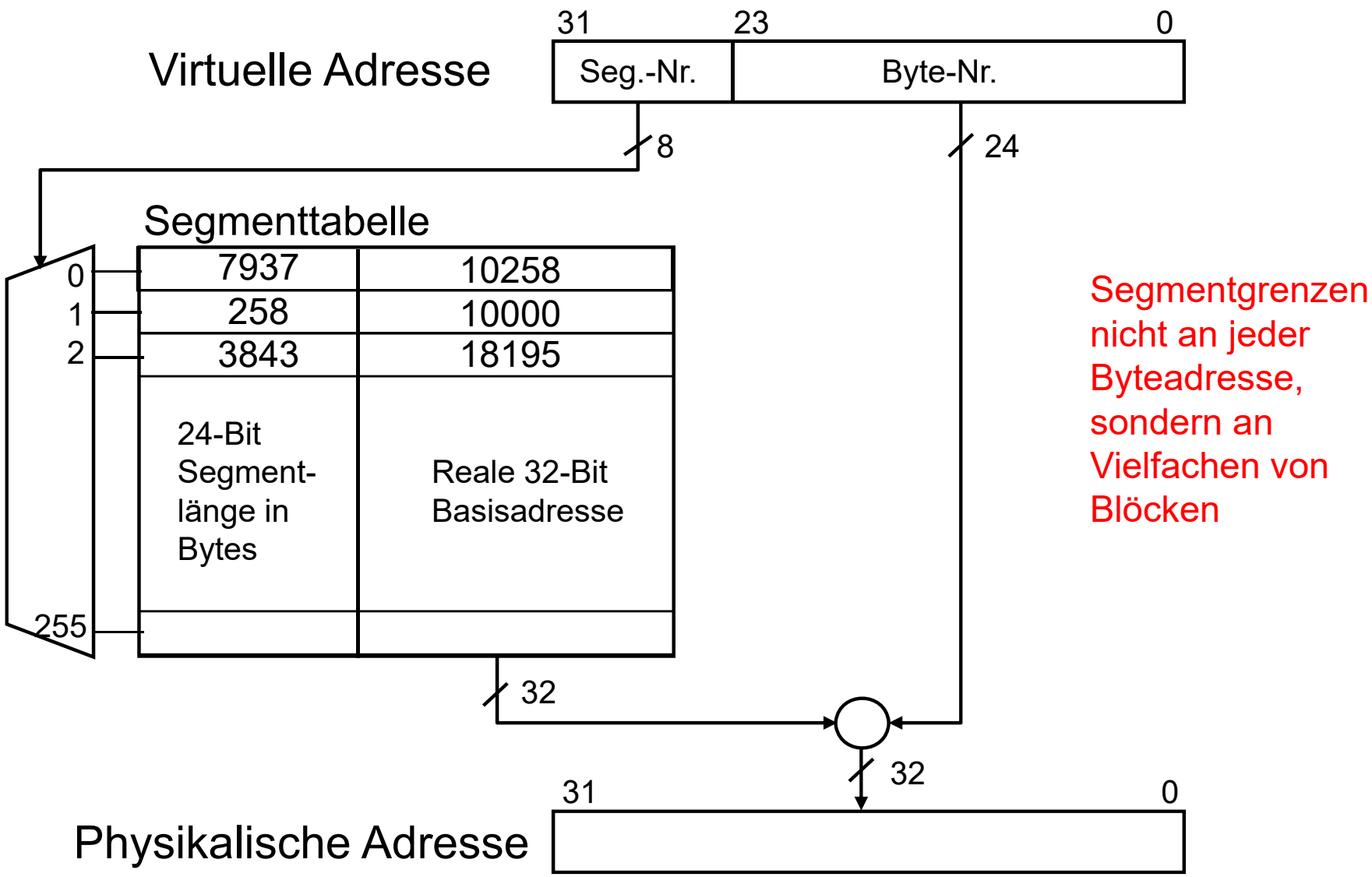
Seitenwechsel



Zusammenfassung

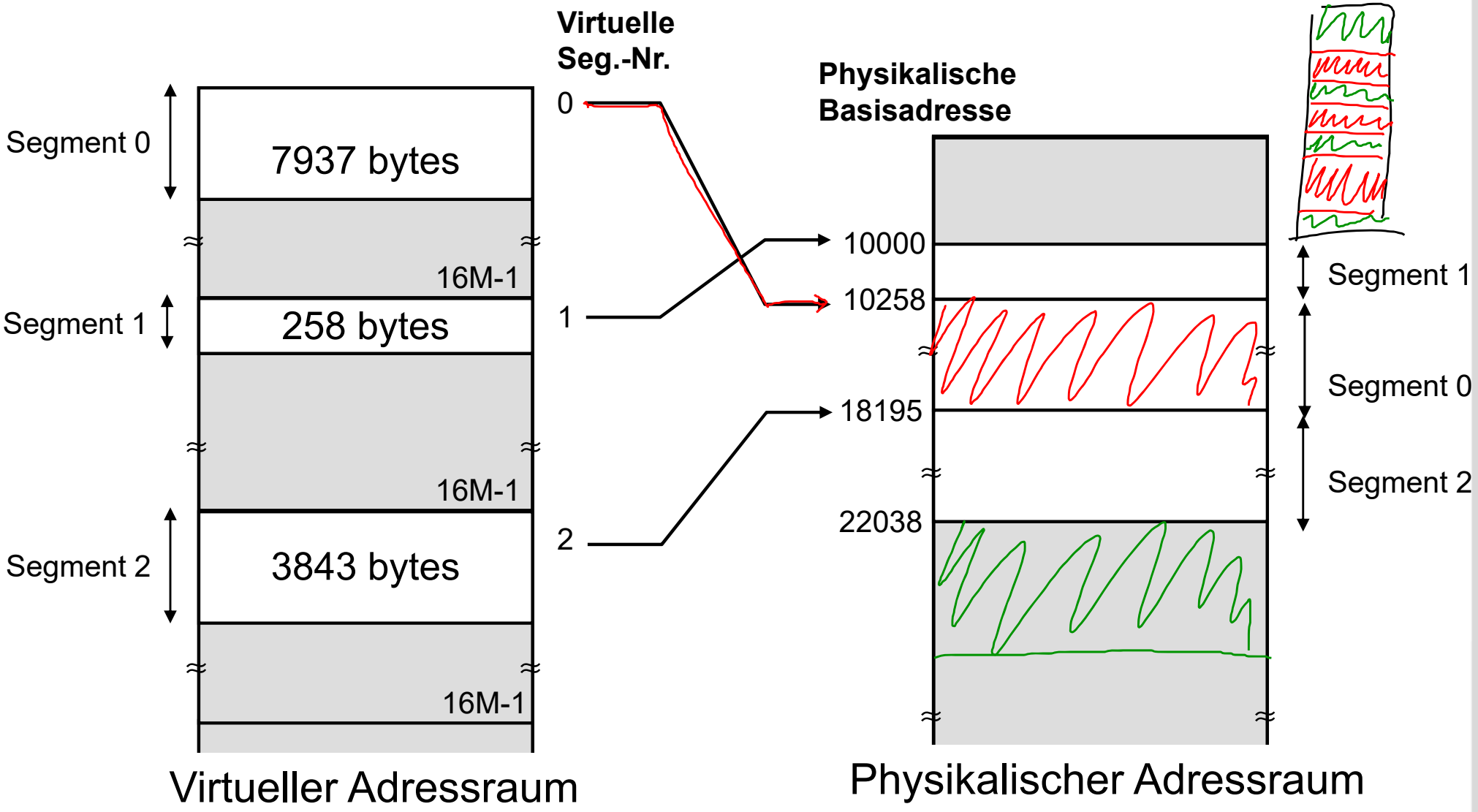


Segmentbasierte Speicherverwaltung

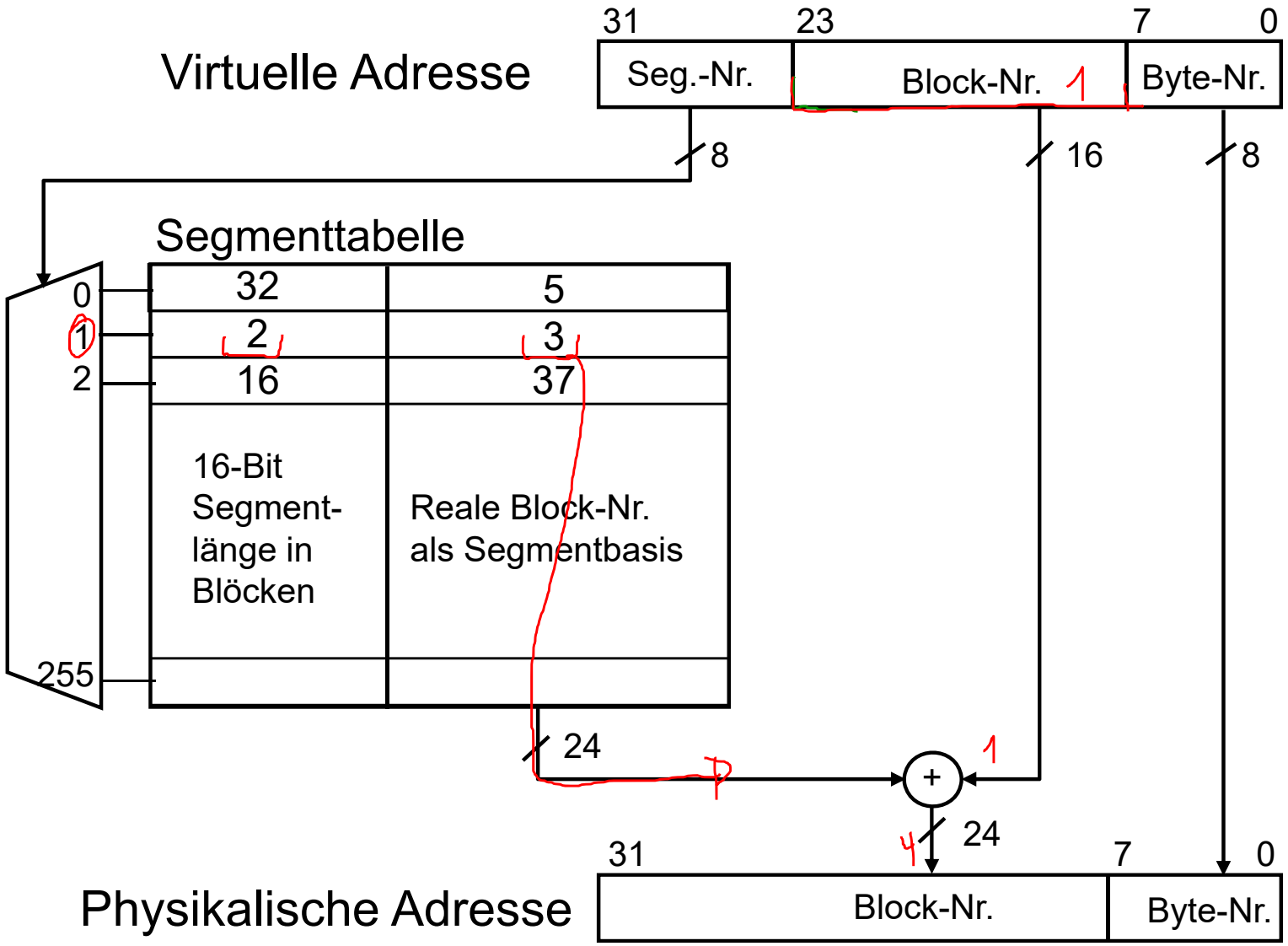


Segmentgrenzen nicht an jeder Byteadresse, sondern an Vielfachen von Blöcken

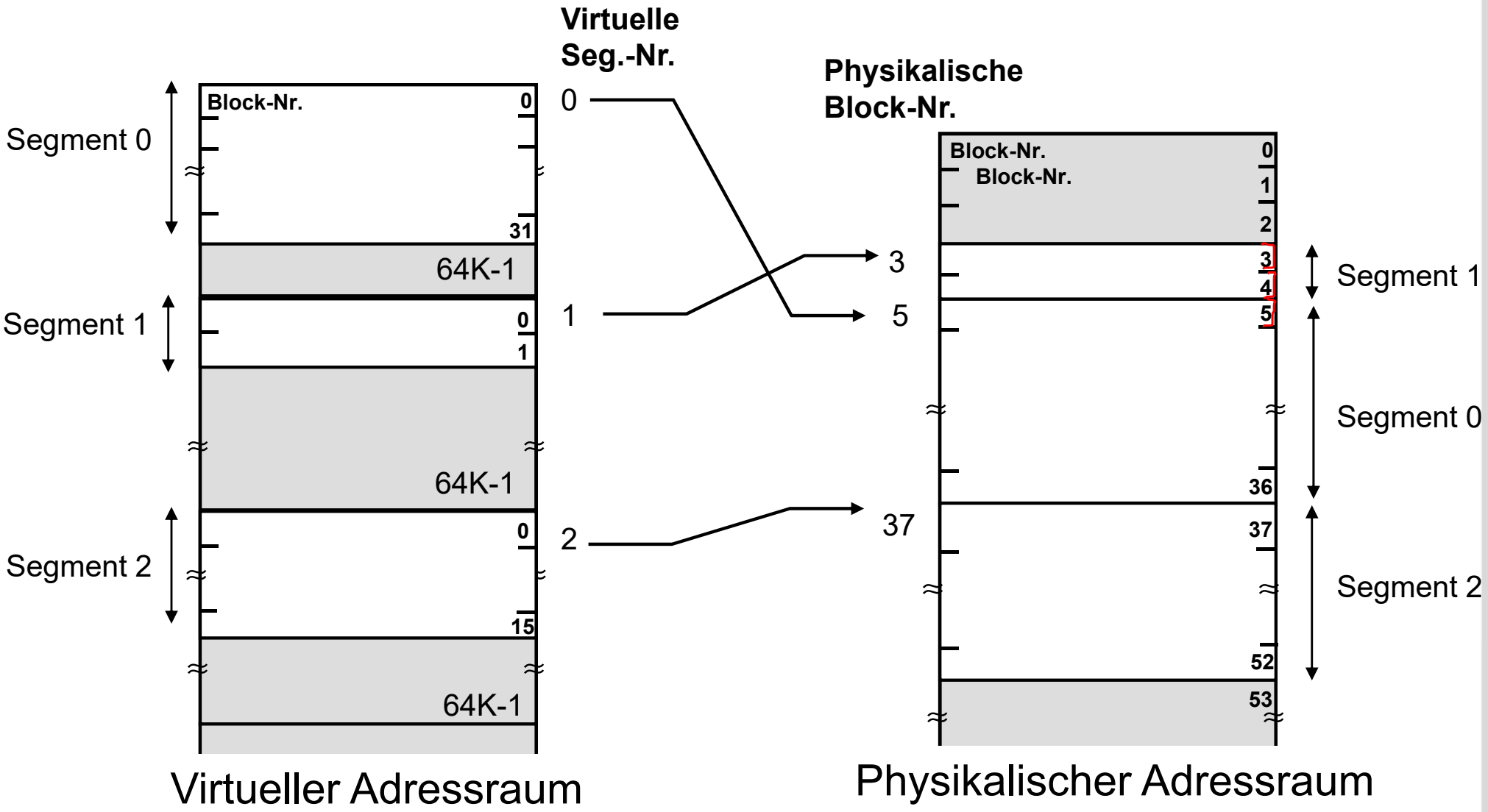
Virtueller und physikalischer Adressraum



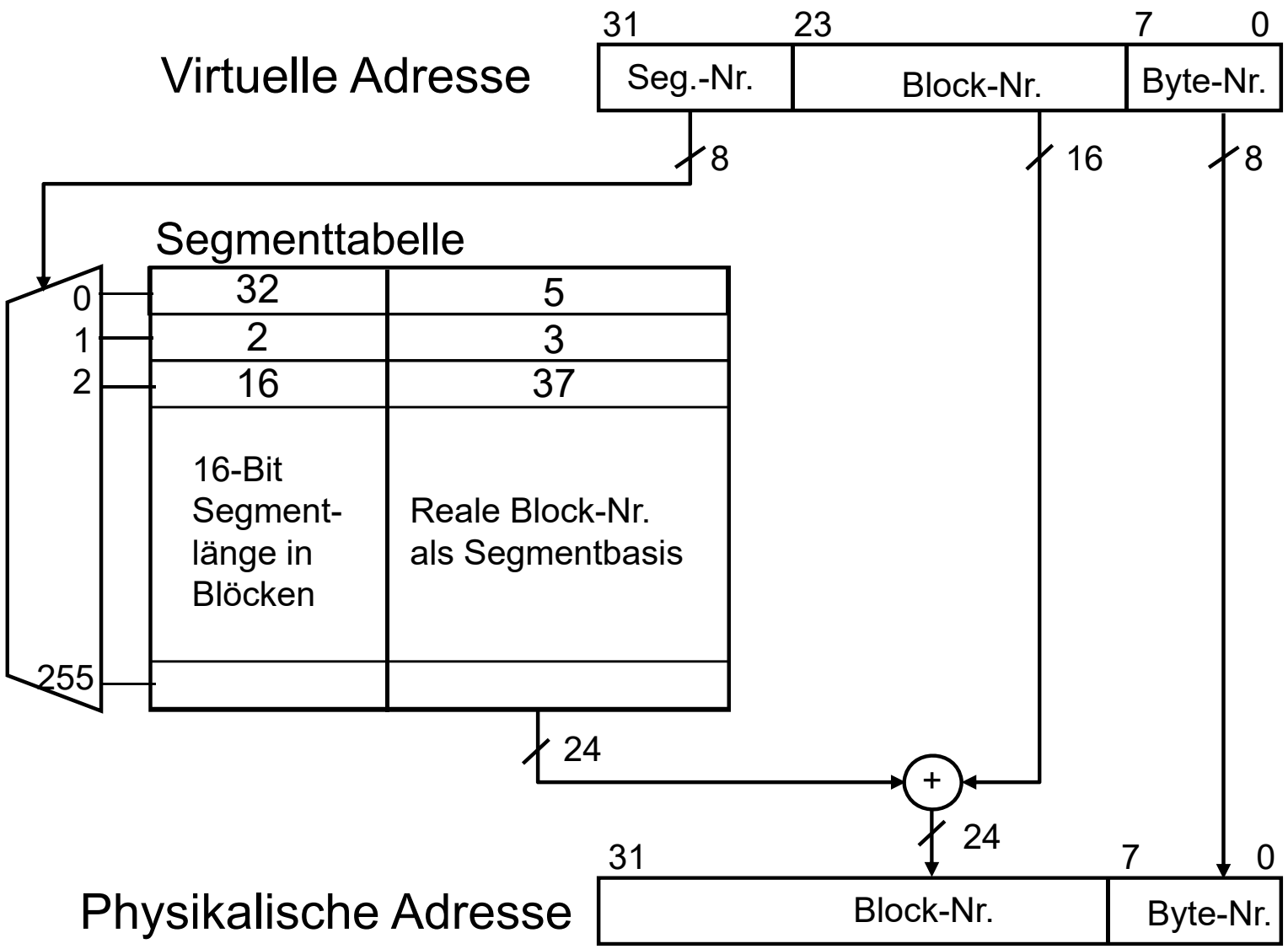
Segmentbasierte Speicherverwaltung



Virtueller und physikalischer Adressraum

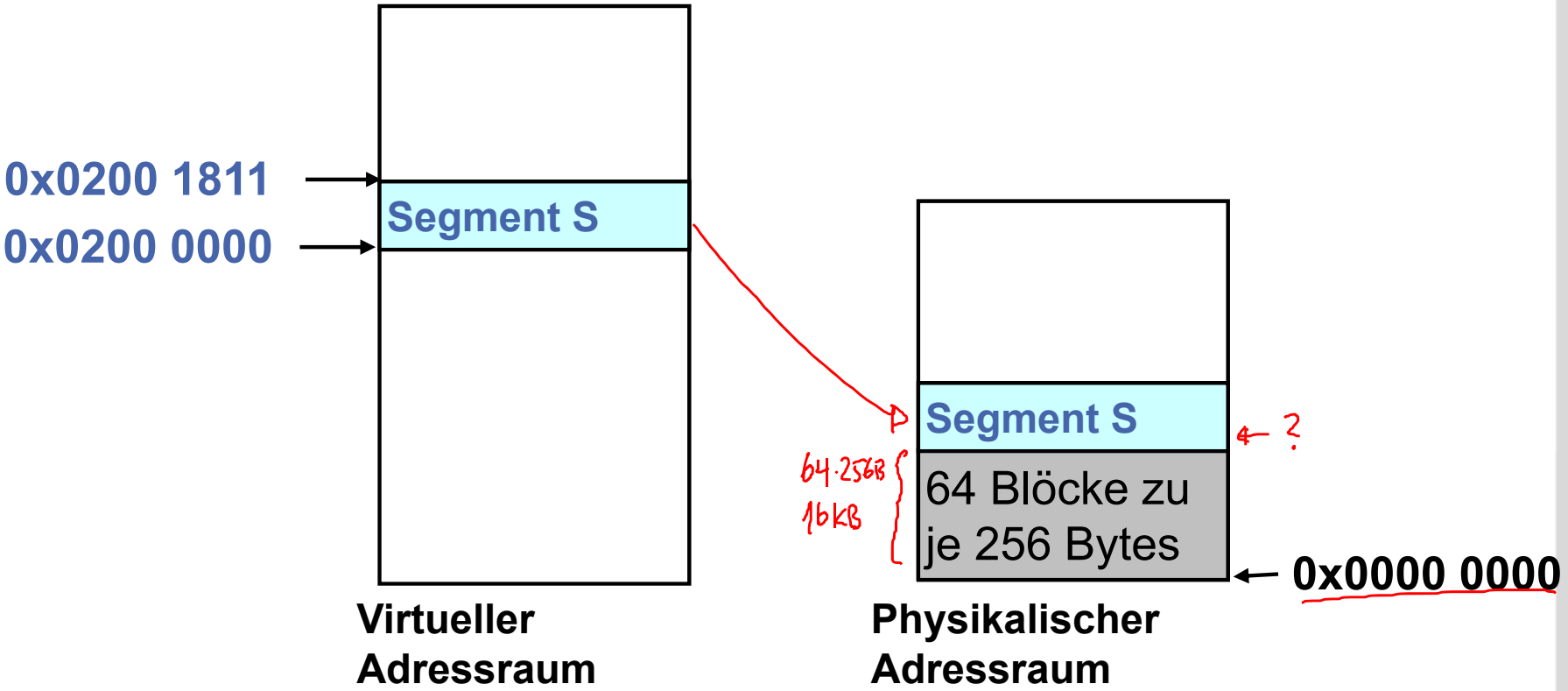


Segmentbasierte Speicherverwaltung



Aufgabe 1

Der Hauptspeicher sei bereits beginnend ab Adresse 0 mit **64 Blöcken** zu je **256 Bytes** belegt. Im Anschluss an diesen Bereich soll ein Segment **S** mit der virtuellen Anfangsadresse **0x0200 0000$_{16}$** und der virtuellen Endadresse **0x0200 1811$_{16}$** geladen werden.



Lösung 1.1

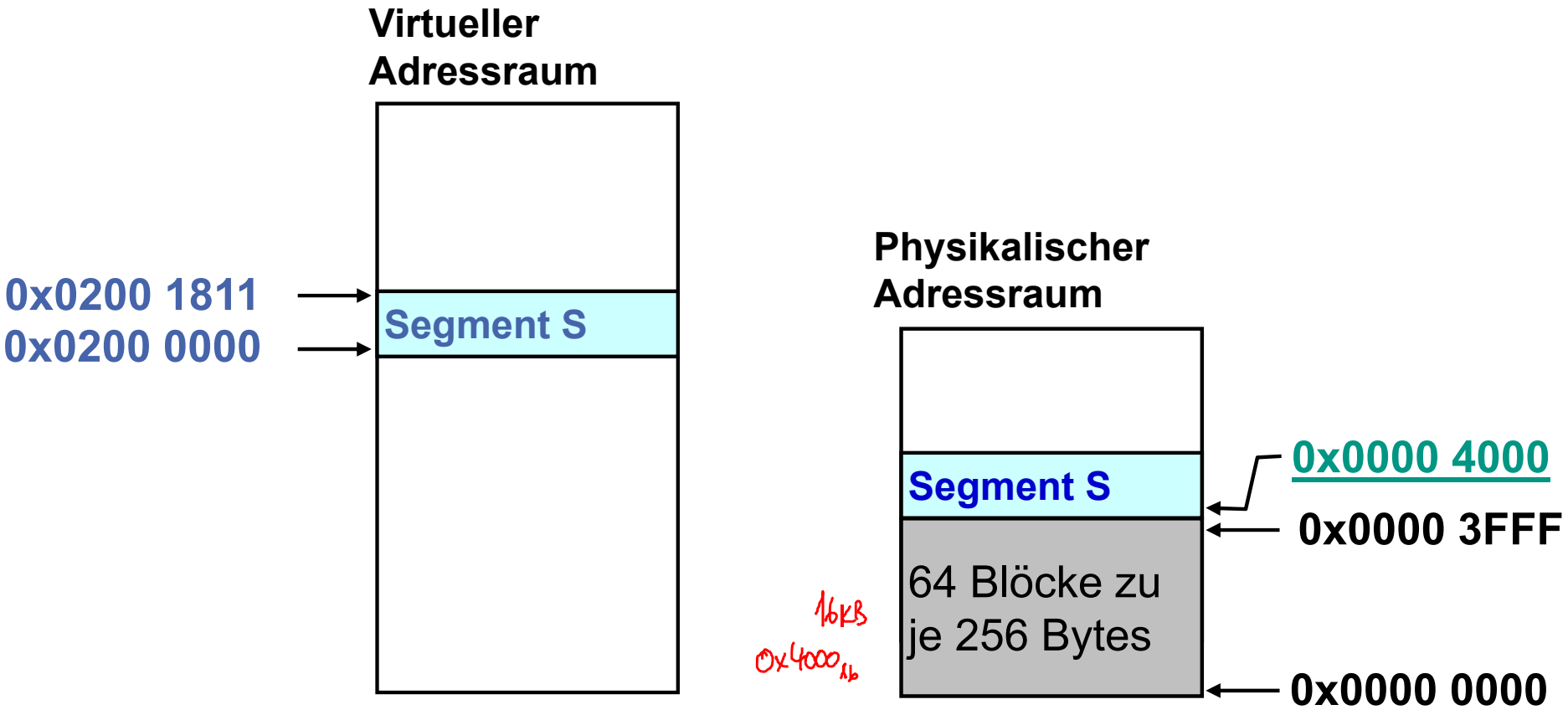
64 Blöcken zu je **256 Bytes** im Hauptspeicher ab Adresse 0

Im Anschluß daran: Segment **S** mit der virtuellen Anfangsadresse $\mathbf{02000000_{16}}$ und Endadresse $\mathbf{02001811_{16}}$

1. Geben Sie die Ladeadresse des Segmentes **S** unter der Bedingung, dass der Hauptspeicher lückenlos gefüllt werden soll an.

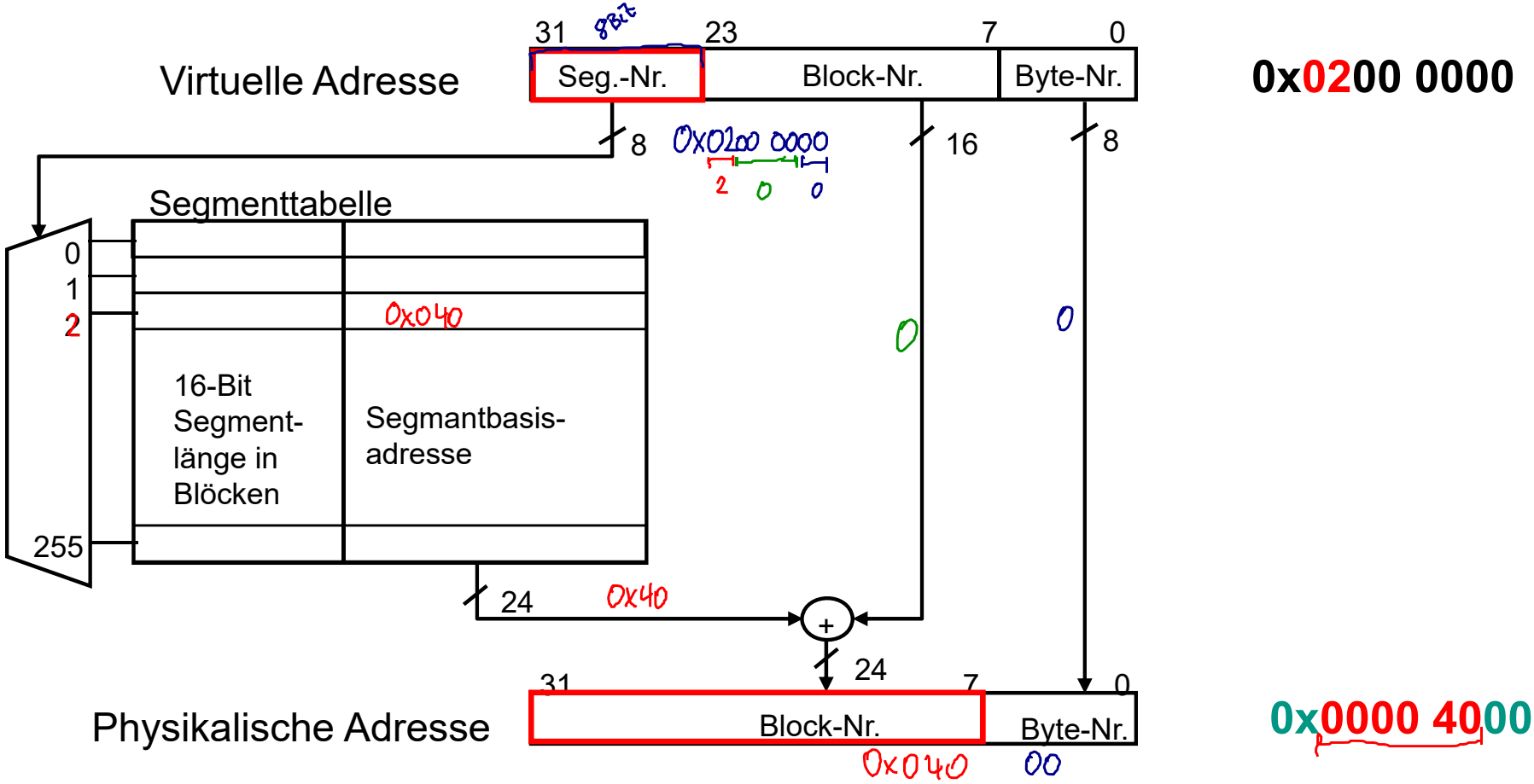
Lösung 1.1

1. Geben Sie die **Ladeadresse** des Segmentes **S** unter der Bedingung, dass der Hauptspeicher lückenlos gefüllt werden soll an.



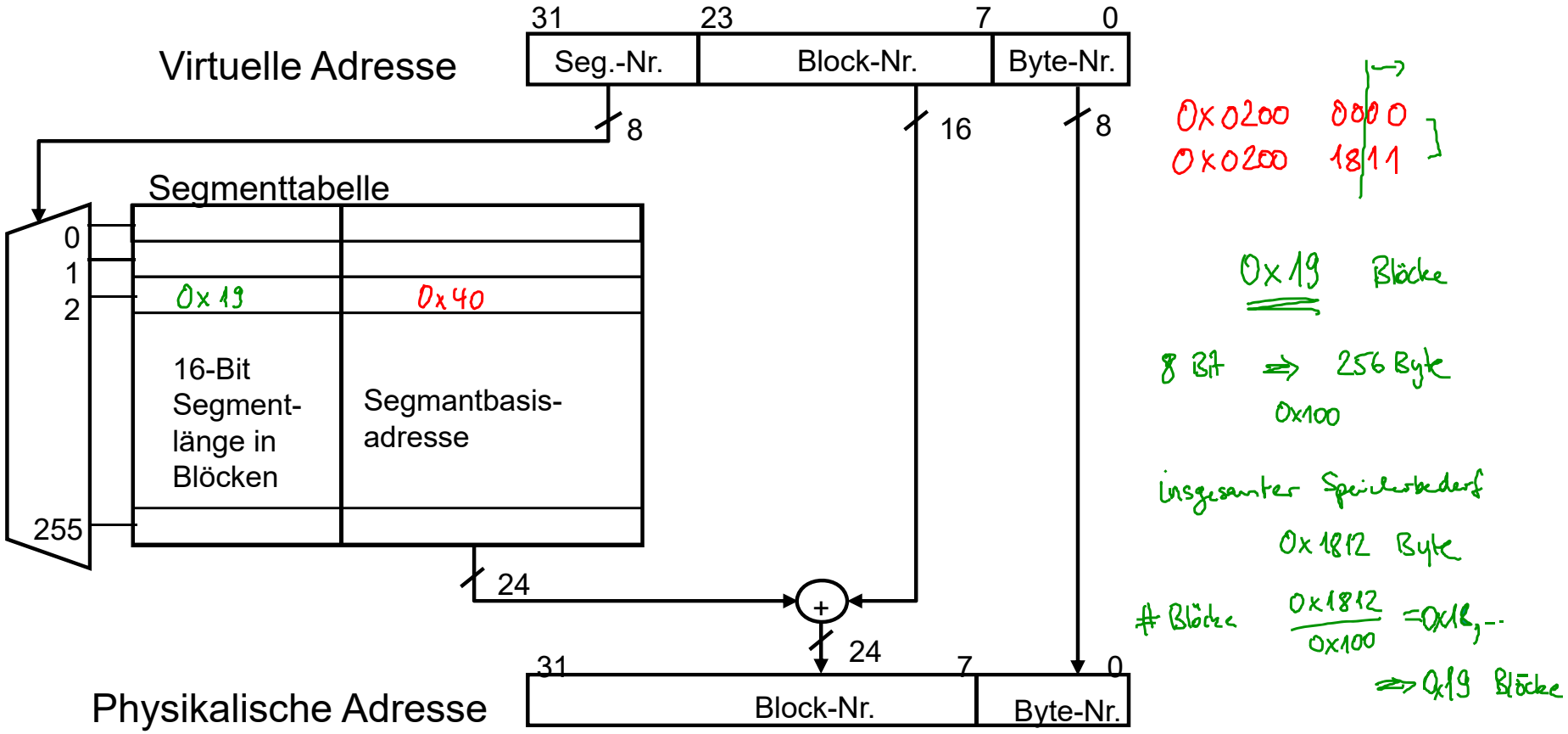
Lösung 1.2

2. An welcher Position im Registerspeicher der MMU muss die physikalische Blocknummer als Abbildungsinformation abgelegt werden und welchen Wert hat sie?



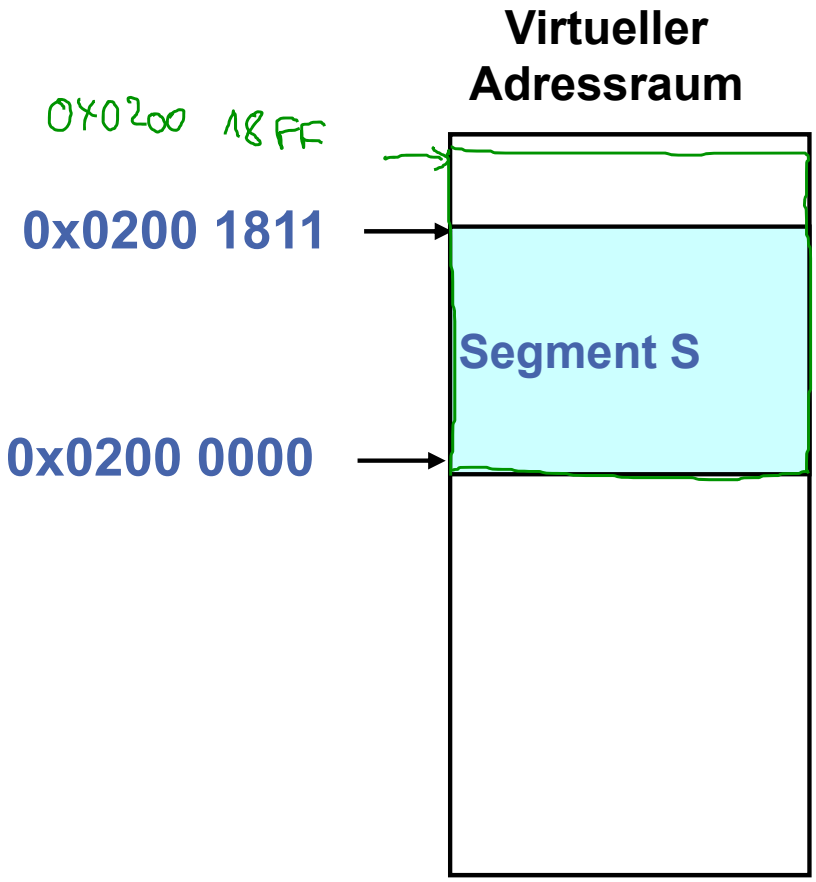
Lösung 1.3

3. Welchen Wert hat die für die Überprüfung von Segmentüberschreitungen einzutragende Blockanzahl?



Lösung 1.3

3. Welchen Wert hat die für die Überprüfung von Segmentüberschreitungen einzutragende Blockanzahl?



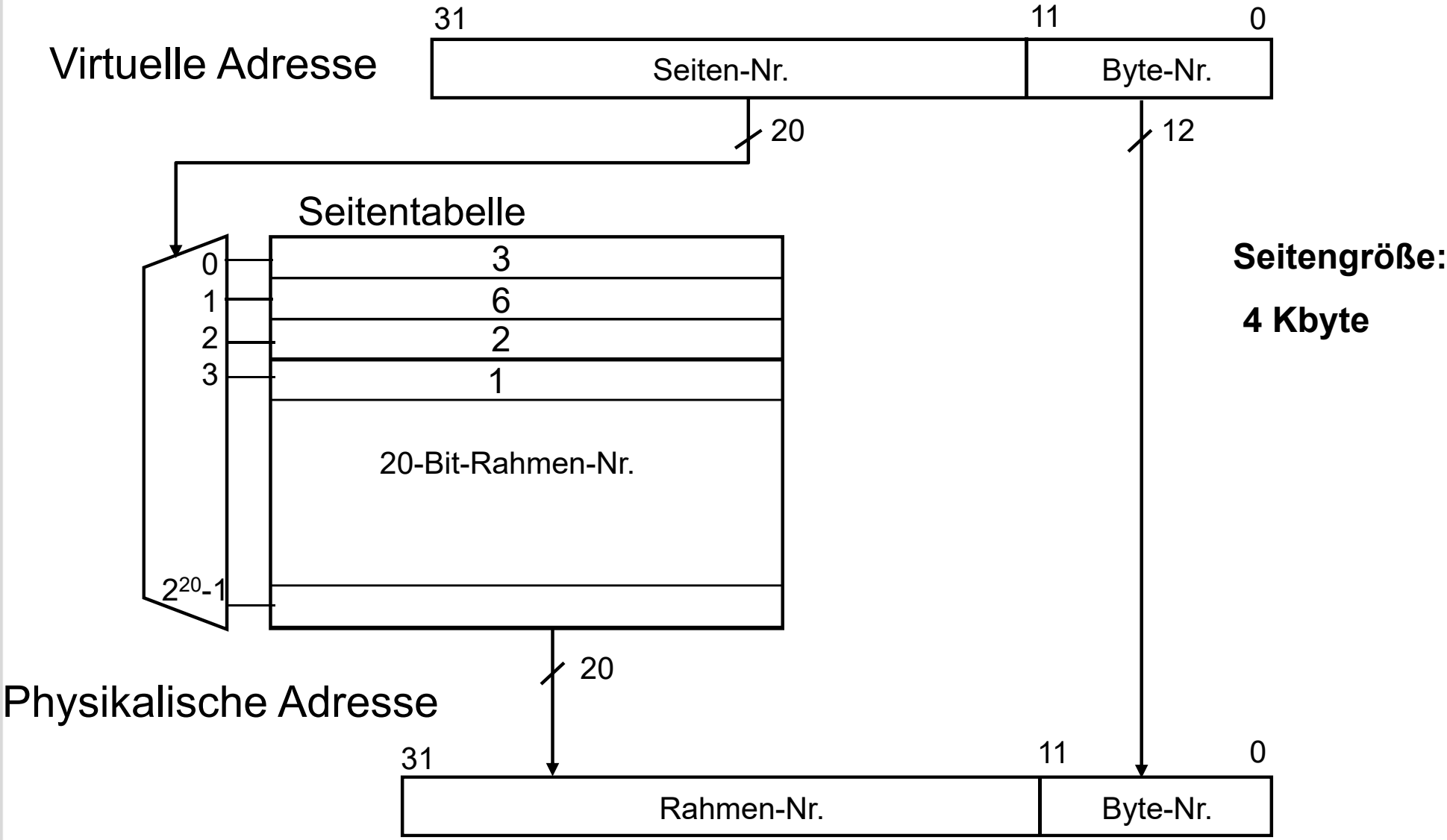
Lösung 1.3

Segment S benötigt 19_{16} Blöcke (25_{10} Blöcke):

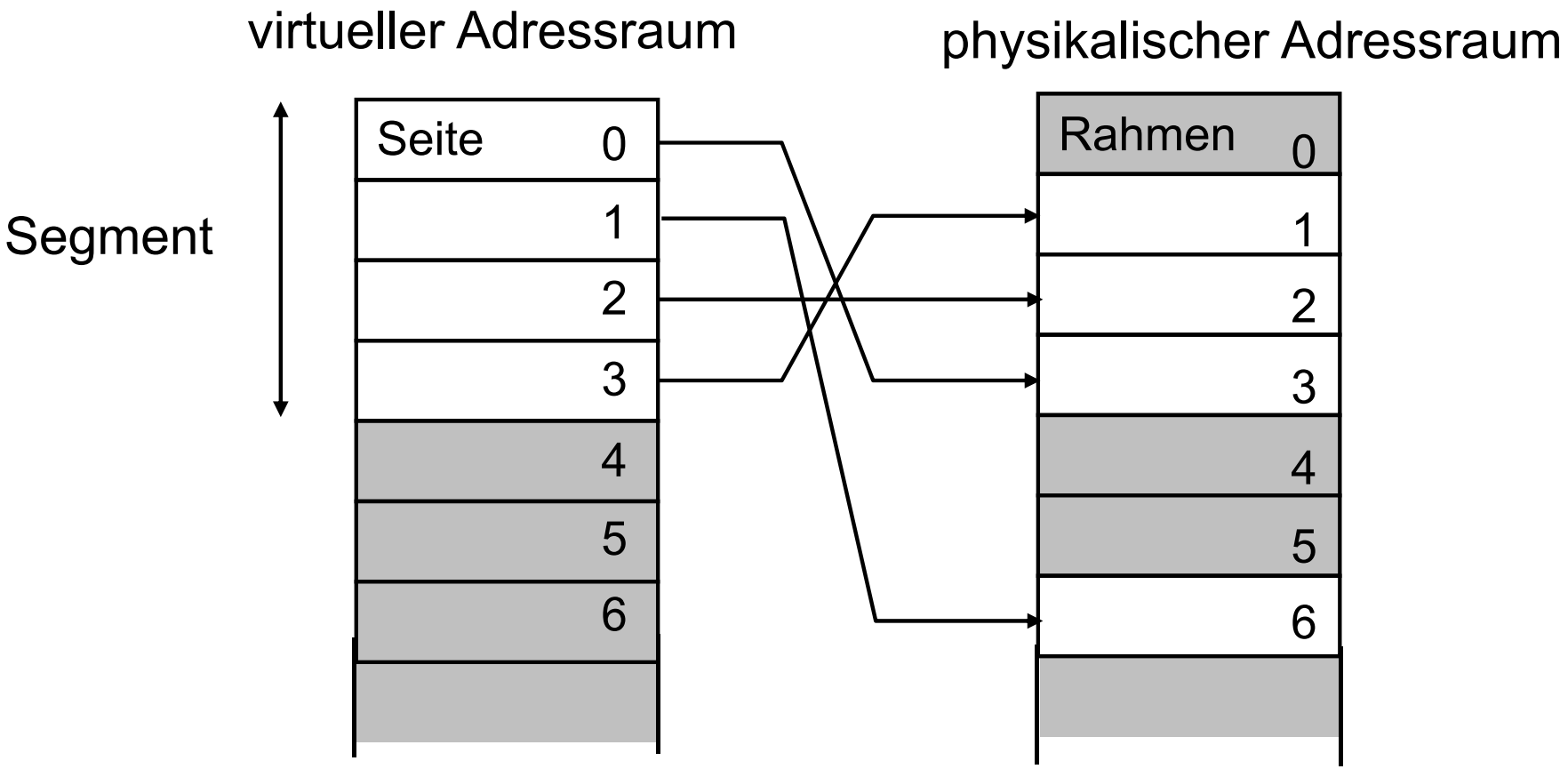
- Block 0 bis Block 18_{16} (24_{10}) sind voll
- Block 19_{16} (25_{10}) ist nur teilweise belegt (enthält nur $12_{16}=18_{10}$ Byte)

➔ Für eine Überprüfung von Segmentüberschreitung muss man für Segment S den Wert 25_{10} eintragen

Seitenwechsel (Paging)



Virtueller und physikalischer Adressraum



Aufgabe 2

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer Seitengröße von 4 KByte, 8 virtuellen Seiten und 4 physikalische Seiten. Die Seitentabelle ist wie folgt belegt:

Virtuelle Seiten-Nr.	Physikalische Seiten-Nr.
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

Lösung 2.1

1. Wie groß ist der physikalische und virtuelle Adressraum?

Physikalischer Adressraum:

4 Seiten x 4 KByte = 16 KByte

Virtueller Adressraum:

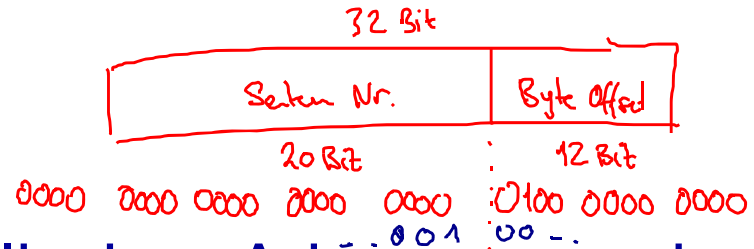
8 Seiten x 4 KByte = 32 KByte

Lösung 2.2

2. Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen:

   0, 1023, 1024, 4096, 24576

Lösung 2.2



2. Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen:

24 1024

0, 1023, 1024, 4096, 24576

Virtuelle Seiten-Nr.	Physikalische Seiten-Nr.
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

Virtuelle		Physikalische	
Adresse	Seiten-Nr.	Seiten-Nr.	Adresse
0	0	3	4096·3 + offset = 12K = 12228 (offset 0)
1023	0	3	4096·3 + 1023 = 13311
1024	0	3	4096·3 + 1024 = 13312
4096	1	1	4096
24576	6	0	4096·0 + 0 = ∅

TI-Klausur

- Findet statt am **23.02.2018**, ab 11:00 Uhr
  - Anmeldung noch möglich bis 16.02.2018
- Hörsaaleinteilung wird auf der Homepage bekanntgegeben
  - http://ti.ira.uka.de/Klausur
- Bei Problemen bezüglich der Anmeldung E-Mail Link auf der Homepage benutzen
- Keine Hilfsmittel erlaubt