

# Musterlösungen zur Klausur

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 28. März 2022, 10:30 – 12:30 Uhr

Name:	Vorname:	Matrikelnummer:
<b>Bond</b>	<b>James</b>	<b>007</b>

<b>Digitaltechnik und Entwurfsverfahren (TI-1)</b>	
Aufgabe 1	10 von 10 Punkten
Aufgabe 2	8 von 8 Punkten
Aufgabe 3	11 von 11 Punkten
Aufgabe 4	6 von 6 Punkten
Aufgabe 5	10 von 10 Punkten

<b>Rechnerorganisation (TI-2)</b>	
Aufgabe 6	8 von 8 Punkten
Aufgabe 7	10 von 10 Punkten
Aufgabe 8	10 von 10 Punkten
Aufgabe 9	9 von 9 Punkten
Aufgabe 10	8 von 8 Punkten

<b>Gesamtpunktzahl:</b>	90 von 90 Punkten
-------------------------	-------------------

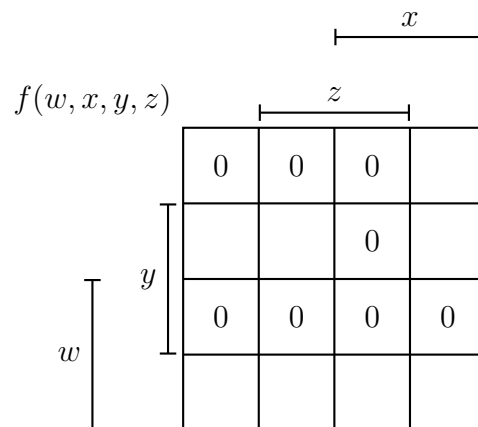
	<b>Note: 1,0</b>
--	------------------

**Aufgabe 1** *Schaltfunktionen*

(10 Punkte)

1. KV-Diagramm:

3 P.

2. Primimplikanten und Kernprimimplikanten  $f(w, x, y, z)$ :

2 P.

$$\underline{A} : w \bar{y}$$

$$\underline{B} : \bar{w} \bar{x} y$$

$$C : \bar{w} y \bar{z}$$

$$D : x \bar{y} \bar{z}$$

$$E : \bar{w} x \bar{z}$$

3. Disjunktive Minimalform von  $f(w, x, y, z)$ :

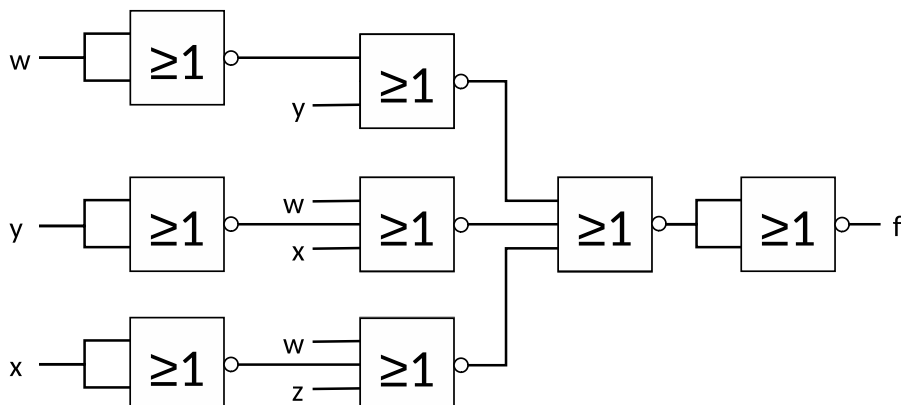
1 P.

$$f(w, x, y, z) = A \vee B \vee E \quad (= w \bar{y} \vee \bar{w} \bar{x} y \vee \bar{w} x \bar{z})$$

4. Schaltnetz mit NOR-Gatter:

4 P.

$$\begin{aligned}
 \bar{f} &= \overline{w \bar{y} \vee \bar{w} \bar{x} y \vee \bar{w} x \bar{z}} \\
 &= \overline{\bar{w} \vee \bar{y} \vee \bar{w} \vee x \vee \bar{y} \vee \bar{w} \vee \bar{x} \vee z} \\
 &= \text{NOR}_3(\text{NOR}_2(\bar{w}, y), \text{NOR}_3(w, x, \bar{y}), \text{NOR}_3(w, \bar{x}, z)) \\
 f &= \text{NOR}_2(\bar{f}, \bar{f})
 \end{aligned}$$

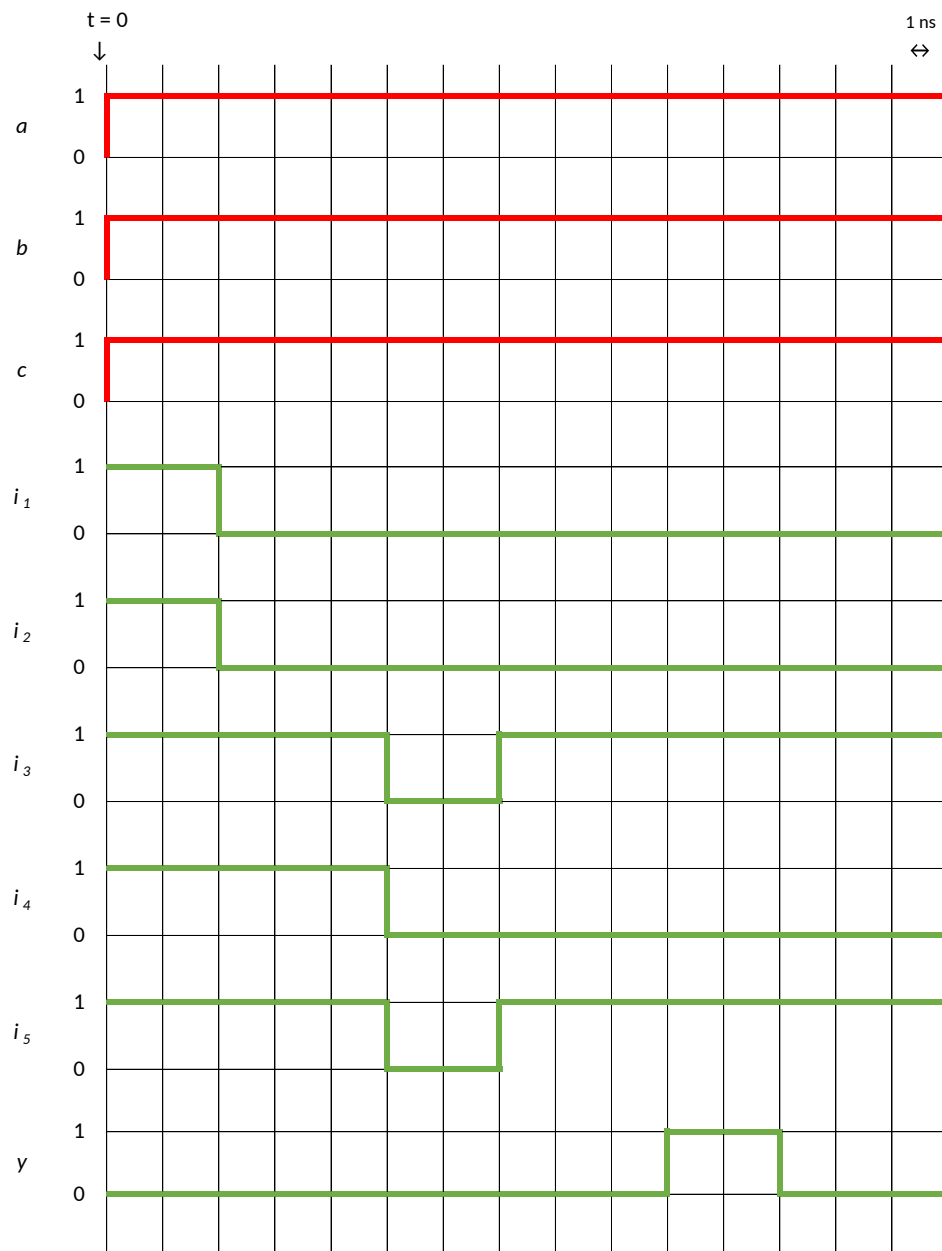


**Aufgabe 2** Laufzeiteffekte

(8 Punkte)

1. Verlauf der Signale:

4 P.





**Aufgabe 3**    *Boolesche Algebra*

(11 Punkte)

1.

$$\begin{aligned}
(a \leftrightarrow b) \leftrightarrow c &= (a \leftrightarrow b)c \vee (a \nleftrightarrow b)\bar{c} \\
&= (ab \vee \bar{a}\bar{b})c \vee (a\bar{b} \vee \bar{a}b)\bar{c} \\
&= abc \vee \bar{a}\bar{b}c \vee a\bar{b}\bar{c} \vee \bar{a}b\bar{c} \\
&= a(bc \vee \bar{b}\bar{c}) \vee \bar{a}(b\bar{c} \vee \bar{b}c) \\
&= a(b \leftrightarrow c) \vee \bar{a}(b \nleftrightarrow c) \\
&= a \leftrightarrow (b \leftrightarrow c)
\end{aligned}$$

3 P.

2.

$$\begin{aligned}
a \leftrightarrow b \leftrightarrow c &= a \leftrightarrow (b \leftrightarrow c) && \text{Assoziativgesetz} \\
&= a \nleftrightarrow \overline{(b \leftrightarrow c)} && (x \leftrightarrow y = xy \vee \bar{x}\bar{y} = x \nleftrightarrow \bar{y}) \\
&= a \nleftrightarrow (b \nleftrightarrow c) && \text{Inverses Element} \\
&= a \nleftrightarrow b \nleftrightarrow c && \text{Assoziativgesetz}
\end{aligned}$$

3 P.

3.

- Negation:

$$\bar{a} = a \rightarrow 0$$

5 P.

- Konjunktion:

$$a \wedge b = (a \rightarrow (b \rightarrow 0)) \rightarrow 0$$

- Disjunktion:

$$a \vee b = (a \rightarrow 0) \rightarrow b$$

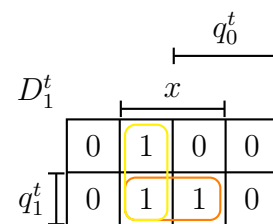
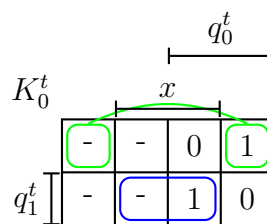
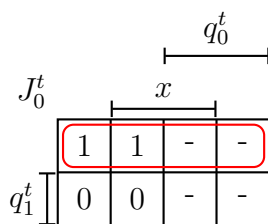
**Aufgabe 4** *Schaltwerke*

(6 Punkte)

1. DMF der Ansteuerfunktionen:

3 P.

Zustand		Eingabe	Folgezustand		Ansteuerfunktionen der Flipflops		
$q_0^t$	$q_1^t$	$x^t$	$q_0^{t+1}$	$q_1^{t+1}$	$J_0^t$	$K_0^t$	$D_1^t$
0	0	0	1	0	1	-	0
0	0	1	1	1	1	-	1
0	1	0	0	0	0	-	0
0	1	1	0	1	0	-	1
1	0	0	0	0	-	1	0
1	0	1	1	0	-	0	0
1	1	0	1	0	-	0	0
1	1	1	0	1	-	1	1

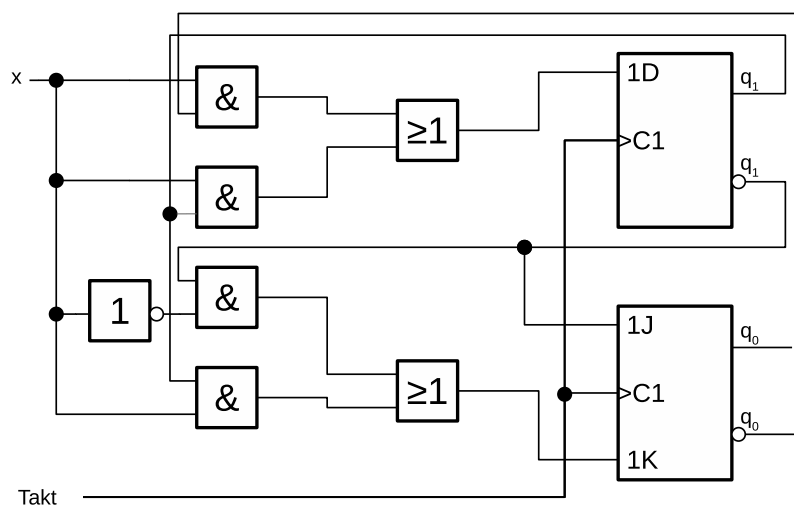


Man erhält:

$$\begin{aligned}
 J_0^t &= \bar{q}_1^t \\
 K_0^t &= \bar{x} \bar{q}_1^t \vee x q_1^t \\
 D_1^t &= x \bar{q}_0^t \vee x q_1^t
 \end{aligned}$$

2. Schaubild:

3 P.



**Aufgabe 5** *Rechnerarithmetik*

(10 Punkte)

- 1.
- $-\frac{1}{4}$
- als 32-Bit IEEE-754 Gleitkomma:

2 P.

$$\begin{aligned}
-\frac{1}{4} &\Rightarrow \text{Sign} = 1 \\
\frac{1}{4} &= 1,0 \cdot 2^{-2} \Rightarrow \text{Man} = 0 \\
\text{Exp} = -2 &\Rightarrow \text{Char} = \text{Exp} + 127_{10} = 125_{10} = 0111\ 1101_2
\end{aligned}$$

31	30	23	22	0
1	01111101	0000	...	0000

Die Zahl kann exakt repräsentiert werden.

2. unsigned
- $u$
- :

1 P.

$$\begin{array}{r}
0x\ B\ F\ D\ 3 \\
- \ 0x\ A\ B\ 1\ 2 \\
\hline
0x\ 1\ 4\ C\ 1
\end{array} \Rightarrow u = 0x14C1$$

3. signed
- $u$
- :

1 P.

 $s$  und  $t$  liegen als Zweierkomplement vor. Es ist keine Umformung notwendig:

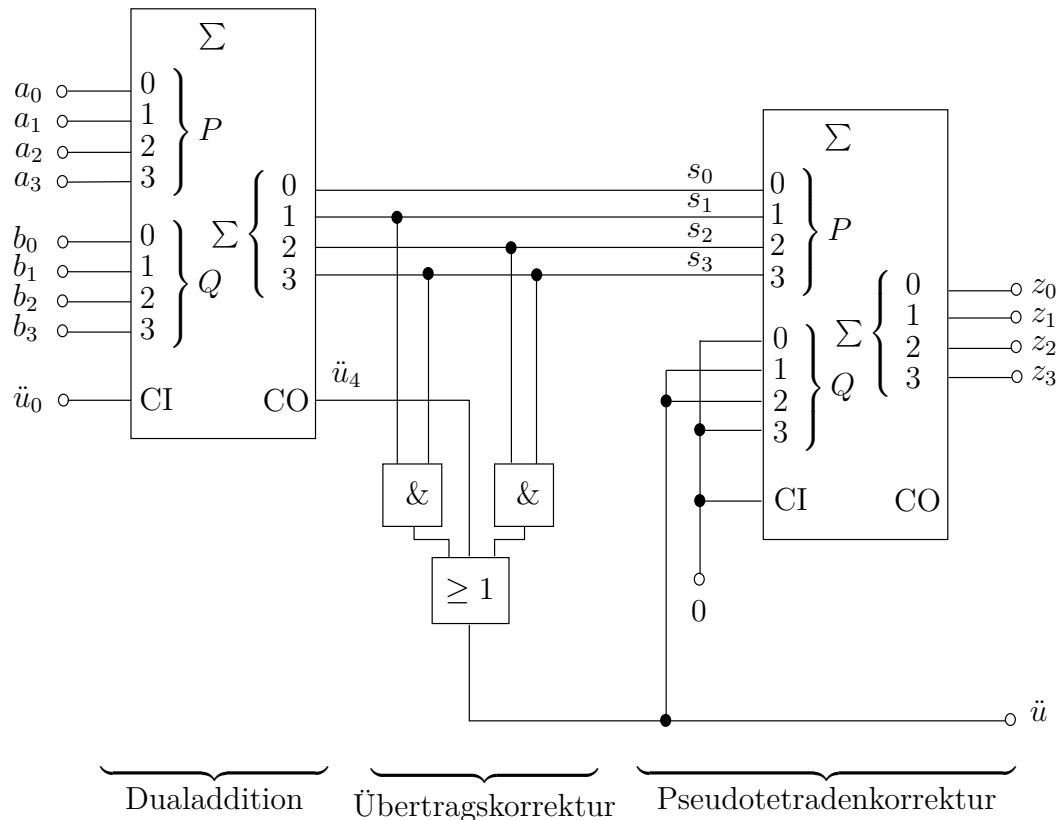
$$\begin{array}{r}
0x\ B\ F\ D\ 3 \\
- \ 0x\ A\ B\ 1\ 2 \\
\hline
0x\ 1\ 4\ C\ 1
\end{array} \Rightarrow u = 0x14C1$$

4. BCD-Addierer für eine Tetrade: Eine Korrektur ist durch die Addition von 0110 zu  $(s_3 s_2 s_1 s_0)$  notwendig, wenn

3 P.

- ein Übertrag bei der Addition der beiden Tetraden  $(a_3 a_2 a_1 a_0)$  und  $(b_3 b_2 b_1 b_0)$  auftritt, d. h.  $\ddot{u}_4 = 1$  oder
- das Ergebnis  $(s_3 s_2 s_1 s_0)$  eine Pseudotetrade ist, d. h.  $s_3 = s_2 = 1$  oder  $s_3 = s_1 = 1$

Damit ergibt sich:  $\ddot{u} = \ddot{u}_4 \vee s_1 s_3 \vee s_2 s_3$



5. Datenwörter: (Man beachte:  $k_i = Q S_{i-1}$ )

3 P.

Position	12	11	10	9	8	7	6	5	4	3	2	1
	$m_8$	$m_7$	$m_6$	$m_5$	$k_4$	$m_4$	$m_3$	$m_2$	$k_3$	$m_1$	$k_2$	$k_1$
Codewort 1:	0	1	1	0	0	0	0	1	1	0	0	0
Codewort 2:	1	0	0	0	0	0	0	1	1	0	0	1

Die Prüfbits lassen sich nach den folgenden Regeln berechnen:

$$\begin{aligned}
 k_1 &= k_1 \oplus m_1 \oplus m_2 \oplus m_4 \oplus m_5 \oplus m_7 \\
 k_2 &= k_2 \oplus m_1 \oplus m_3 \oplus m_4 \oplus m_6 \oplus m_7 \\
 k_3 &= k_3 \oplus m_2 \oplus m_3 \oplus m_4 \oplus m_8 \\
 k_4 &= k_4 \oplus m_5 \oplus m_6 \oplus m_7 \oplus m_8
 \end{aligned}$$



- Codewort 1: **0 1 1 0 0 0 0 1 1 0 0 0**  $\Rightarrow k_4 k_3 k_2 k_1 = 0 0 0 0 \Rightarrow$  Es liegt kein Fehler vor  $\Rightarrow$  Datenwort 1: **0 1 1 0 0 0 1 0**
- Codewort 2: **1 0 0 0 0 0 0 1 1 0 0 1**  $\Rightarrow k_4 k_3 k_2 k_1 = 1 1 0 0 \Rightarrow$  Es liegt ein Fehler an der 12. Position vor  $\Rightarrow$  Datenwort 2: **0 0 0 0 0 0 1 0**

**Aufgabe 6** *MIMA-Architektur*

(8 Punkte)

1. Kodierung des Mikroprogramms für die Lese-Phase:

4 P.

Takt	Adresse	Befehl in hexadezimaler Schreibweise
1. Takt	0x00	2 1 0 8 8 0 1 (X = P <sub>w</sub> = S = 1; R = 1)
2. Takt	0x01	1 4 0 0 8 0 2 (Y = E = 1; R = 1)
3. Takt	0x02	0 0 0 1 8 0 3 (C <sub>2</sub> -C <sub>0</sub> = 001; R = 1)
4. Takt	0x03	0 A 0 0 0 0 4 (Z = P <sub>r</sub> = 1)
5. Takt	0x04	0 0 9 0 0 0 5 (I <sub>r</sub> = D <sub>w</sub> = 1)

2. Mikroprogramme:

4 P.

LDV	STV
7. Takt: IR → SAR; R = 1	7. Takt: Akku → SDR
8. Takt: R = 1	8. Takt: IR → SAR; W = 1
9. Takt: R = 1	9. Takt: W = 1
10. Takt: SDR → Akku	10. Takt: W = 1
EQL	JMP
7. Takt: IR → SAR; R = 1	7. Takt: IR → IAR
8. Takt: Akku → X; R = 1	
9. Takt: R = 1	
10. Takt: SDR → Y	
11. Takt: ALU auf Vergleich	
12. Takt: Z → Akku	

**Aufgabe 7** *C und RISC-V Assembler*

(10 Punkte)

1.            `addi t2, t2, -5`  
               `add t0, t1, t2`

1 P.

2.            `f = g + h + i`

1 P.

3.            `slli t0, t0, 2`        `# t0 = i*4`  
               `add t0, a0, t0`    `# t0 = &A[i]`  
               `lw t0, 0(t0)`      `# t0 = A[i]`  
               `slli t1, t1, 2`        `# t1 = j*4`  
               `add t1, a0, t1`    `# t1 = &A[j]`  
               `lw t1, 0(t1)`      `# t1 = A[j]`  
               `add t1, t0, t1`    `# t1 = A[i] + A[j]`  
               `sw t1, 32(a1)`    `# B[8] = A[i] + A[j]`

4 P.

4. Inhalte der Zielregister:

4 P.

Befehl	Zielregister =        (z. B. <code>\$s7 = 0x0000 F00A</code> )
<code>addi s1, zero, 0x28</code>	<code>\$s1 = 0x0000 0028</code>
<code>srai s2, s1, 1</code>	<code>\$s2 = 0x0000 0014</code>
<code>slti s3, s2, 10</code>	<code>\$s3 = 0x0000 0000</code>
<code>lui s4, 0x21</code>	<code>\$s4 = 0x0002 1000</code>
<code>xor s5, s4, s1</code>	<code>\$s5 = 0x0002 1028</code>
<code>andi s6, s5, -1</code>	<code>\$s6 = 0x0002 1028</code>

**Aufgabe 8** *Pipelining*

(10 Punkte)

1.
  - ohne Pipelining:  
 $250 + 350 + 250 + 400 + 200 = 1450 \text{ [ps]}$ 

0,5 P.
  - mit Pipelining:  
 Pipelinestufe mit der größten Latenz  $\Rightarrow 400 \text{ ps}$ 

0,5 P.
2.
  - ohne Pipelining:  
 $1450 \text{ ps}$ 

0,5 P.
  - mit Pipelining:  
 5 Pipelinestufen  
 $5 \cdot 400 \text{ ps} = 2000 \text{ ps}$ 

0,5 P.
3.
  - Geeignete Stufe:  
 MEM, da größte Latenz
 

1 P.
  - Zykluszeit:  
 $350 \text{ ps}$ 

1 P.
4. Registerinhalte:
 

2 P.

Register	t1	t2	t3	t4
Inhalt	12	7	15	12

5. • Programmstück:

2 P.

```
S0:    addi t1, t0, 5
        NOP
        NOP
S1:    add  t2, t1, t0
S2:    addi t3, t1, 15
        NOP
        NOP
S3:    add  t4, t3, t1
```

- Registerinhalte:

2 P.

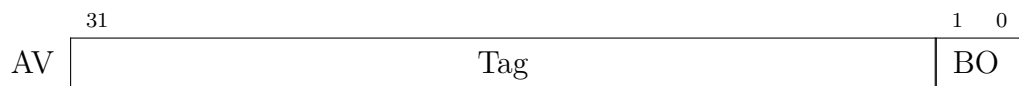
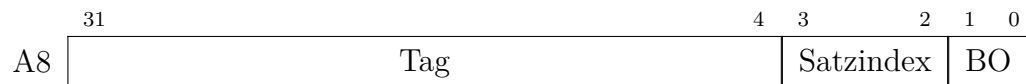
Register	t1	t2	t3	t4
Inhalt	12	19	27	39

**Aufgabe 9** *Cache-Speicher*

(9 Punkte)

1. Unterteilung der Hauptspeicheradresse:

3 P.



2. Anzahl der Vergleiche:

1 P.

Cache	Anzahl der Vergleiche
DM	1
A8	8
AV	32

3. Lesezugriffe von links nach rechts; »×« für Cache-Hit und »-« für Cache-Miss:

5 P.

Adresse:	0x0B	0x2B	0x07	0x0C	0x1E	0x0A	0x1A	0x05	0x04	0x29
DM	-	-	-	-	-	×	-	×	×	×
A8	-	-	-	-	-	×	-	×	×	×
AV	-	-	-	-	-	×	-	×	×	×

## Aufgabe 10 Virtuelle Speicherverwaltung (8 Punkte)

1. Anzahl der Bits der virtuellen Adresse: 34 Bits
2. Anzahl der Bits der physikalischen Adresse: 27 Bits
3. Anzahl der Bits der virtuellen Seitennummer: 22 Bits
4. Anzahl der Bits der physikalische Seitennummer: 15 Bits
5. *first-fit*-Strategie:

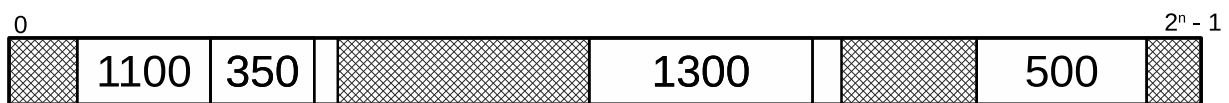
1 P.

1 P.

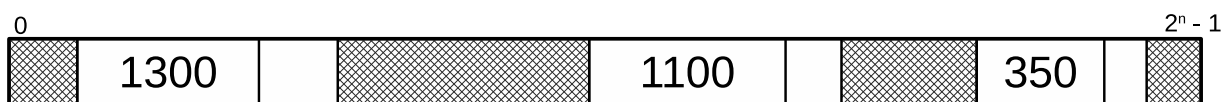
1 P.

1 P.

2 P.



*best-fit*-Strategie:



Für die Speicheranforderung mit 500 Byte ist bei der *best-fit*-Strategie kein Platz mehr vorhanden.

6. *best-fit*-Strategie:

2 P.

Vorteil: Geringere Fragmentierung und damit bessere Speicherausnutzung

Nachteil: Höherer Suchaufwand gegenüber der *first-fit*-Strategie zum Finden einer freien Lücke.

*first-fit*-Strategie:

Kleine Lücken im Hauptspeicher sammeln sich an den unteren Adressen, während am Speicherende die größeren Lücken zu finden sind. Damit erhöht sich die mittlere Anzahl der erforderlichen Suchschritte zum Auffinden der ersten passenden Lücke.