

Musterlösungen zur Klausur

Digitaltechnik und Entwurfsverfahren & Rechnerorganisation
und

Technische Informatik I/II

am 23. Februar 2018, 11:00 – 13:00 Uhr

Name:	Vorname:	Matrikelnummer:
Bond	James	007

Digitaltechnik und Entwurfsverfahren/TI-1	
Aufgabe 1	11 von 11 Punkten
Aufgabe 2	8 von 8 Punkten
Aufgabe 3	7 von 7 Punkten
Aufgabe 4	10 von 10 Punkten
Aufgabe 5	9 von 9 Punkten

Rechnerorganisation/TI-2	
Aufgabe 6	10 von 10 Punkten
Aufgabe 7	10 von 10 Punkten
Aufgabe 8	10 von 10 Punkten
Aufgabe 9	8 von 8 Punkten
Aufgabe 10	7 von 7 Punkten

Gesamtpunktzahl:	90 von 90 Punkten
------------------	-------------------

Note:	1,0
-------	-----

Aufgabe 1

1. DMF:

2 P.

$$y_{DMF} = c \, b \, a \vee \bar{c} \, \bar{b}$$

a			
1	1	0	–
0	0	1	0
0	0	1	0
1	–	0	0
c			

b d

2. KMF:

3 P.

$$y_{KMF} = (c \vee \bar{b}) \cdot (\bar{c} \vee b) \cdot (\bar{b} \vee a) \quad \text{oder}$$

$$y_{KMF} = (c \vee \bar{b}) \cdot (\bar{c} \vee b) \cdot (\bar{c} \vee a)$$

a			
1	1	0	–
0	0	1	0
0	0	1	0
1	–	0	0
c			

b d

3.

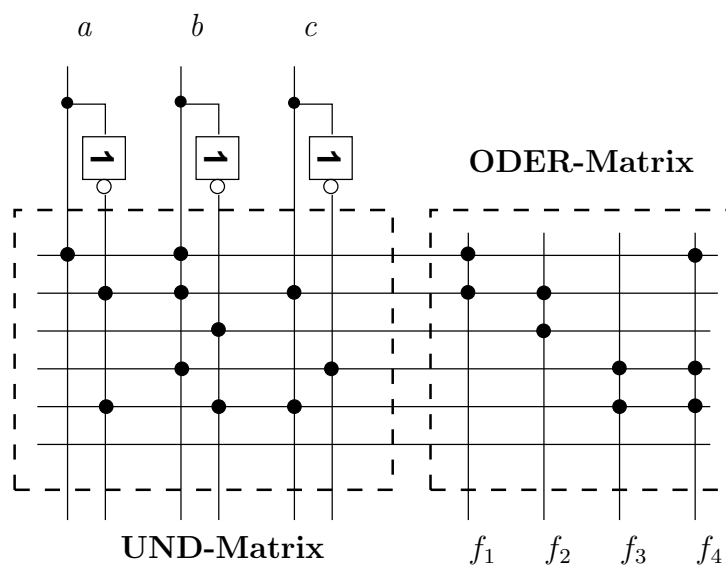
2 P.

Produktterm	X	Erklärung
$\bar{d} \bar{c} b$		Überdeckt weder m_0 noch m_{10}
$c \bar{b}$	X	$\bar{d} \bar{c} \bar{a}$ wird kein Kernprimimplikant mehr
$d \bar{b} a$		Nicht angrenzend an den beiden Kernprimimplikanten.
$c b \bar{a}$	X	$\bar{c} b \bar{a}$ wird kein Kernprimimplikant mehr.

4. PLA: Bündelminimierung der Funktionen:

4 P.

$$\begin{aligned}
 f_1 &= b a \vee c b \bar{a} & f_2 &= \bar{b} \vee c b \bar{a} \\
 f_3 &= \bar{c} b \vee c \bar{b} \bar{a} & f_4 &= \bar{c} b \vee b a \vee c \bar{b} \bar{a} = f_3 \vee b a
 \end{aligned}$$



Aufgabe 2

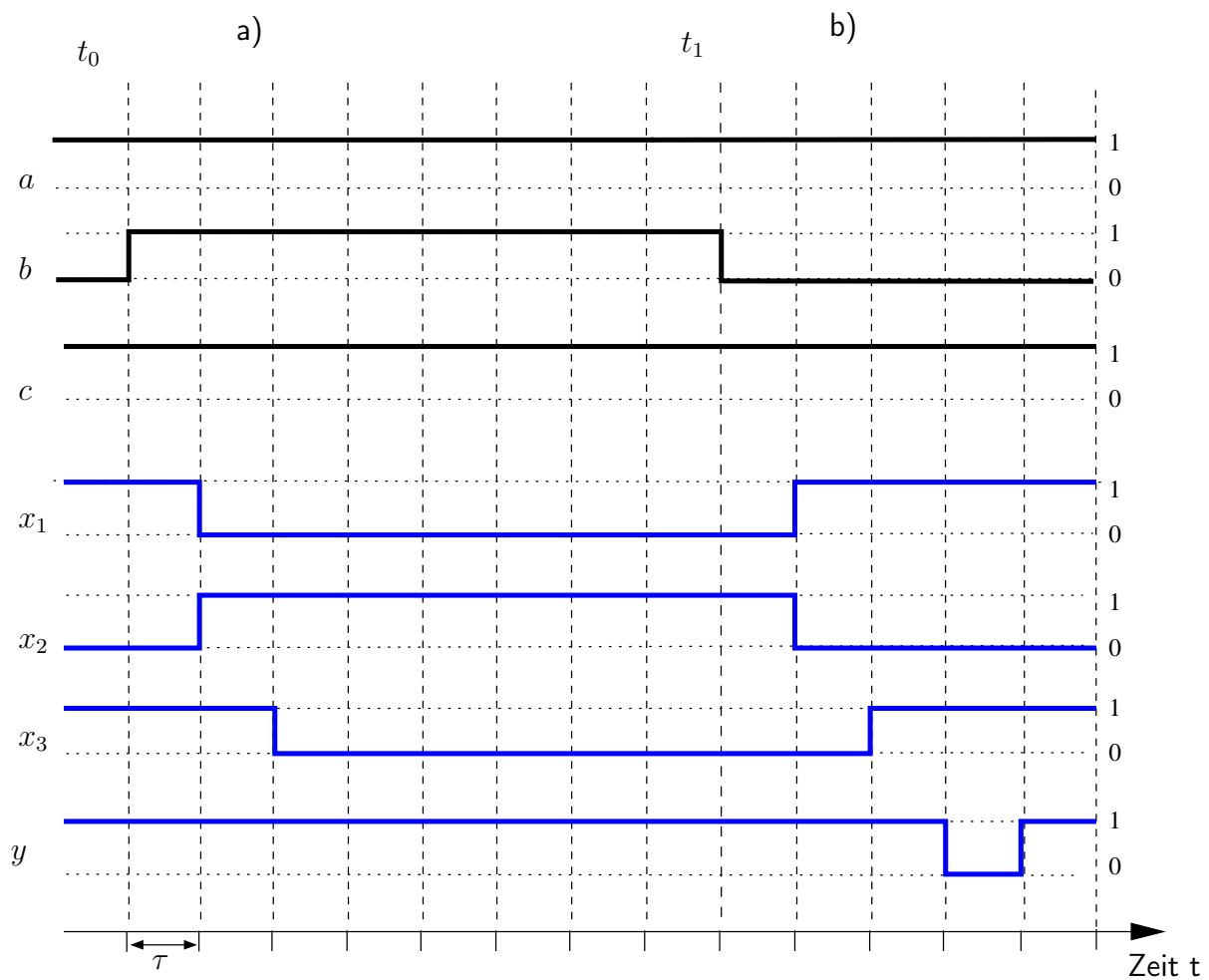
8 P.

Schaltwerk	1	2	3	4
zählt vorwärts			×	×
zählt rückwärts		×		
ist synchron	×	×	×	
kann bei <i>jedem</i> Zählerstand mit Hilfe von x angehalten werden.		×		×

Aufgabe 3

3 P.

1.



2. Typ des Fehlers und Behebungsmöglichkeit:

2 P.

Es tritt ein Hasardfehler beim Übergang $B_7 \rightarrow B_5$ zum Zeitpunkt t_1 auf.

Es handelt sich hierbei um einen Übergang, bei dem nur eine Variablen b ihren Wert wechselt \Rightarrow Der Übergang ist frei von Funktionshasards; der Hasardfehler tritt nicht aufgrund eines Funktionshasards auf und kann nur durch einen Strukturhasard bedingt sein \Rightarrow **1-statischer Strukturhasard**.

Behebung:

- Satz von Eichelberger: Realisierung der Schaltfunktion als die Disjunktion aller Primimplikanten (Fehlender Primimplikant $c a$ in die Realisierung aufnehmen, d. h. $y = b a \vee c \bar{b} \vee c a$)
- Die beim Übergang konstant bleibenden Eingangsvariablen (a und c) über ein zusätzliches UND-Gatter verknüpfen und das Ergebnis mit dem Ausgang des Schaltnetzes ODER-verknüpfen.

3. Übergang mit einem statischen 1-Funktionshasard:

2 P.

	a			
	0 ₀	0 ₁	1 ₅	1 ₄
b	0 ₂	1 ₃	1 ₇	0 ₆
	c			

Beispiele für Übergänge mit Funktionshasard: $B_4 \leftrightarrow B_7$, $B_4 \leftrightarrow B_3$, $B_5 \leftrightarrow B_3$.

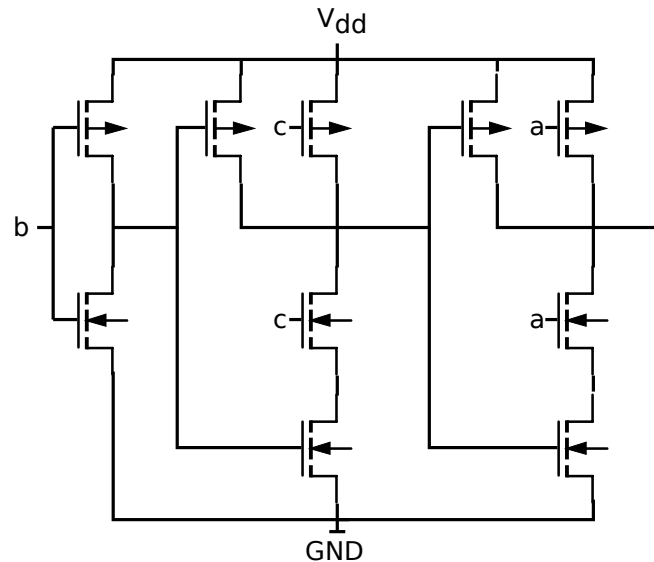
Begründung: Jeder Übergang, bei dem die zugehörige Folge der Funktionswerte nicht monoton ist, ist mit einem Funktionshasard behaftet.

Aufgabe 4

1. Umgeformte Schaltfunktion und Transistor-Schaltbild:

5 P.

$$\begin{aligned} y &= \bar{a} \vee \bar{b}c = \overline{\overline{\bar{a} \vee \bar{b}c}} = \overline{a \wedge \bar{b}c} \\ &= \text{NAND}_2(a, \text{NAND}_2(\bar{b}, c)) \end{aligned}$$



2. Unterschied zwischen n-Kanal- und einem p-Kanal-MOSFET:

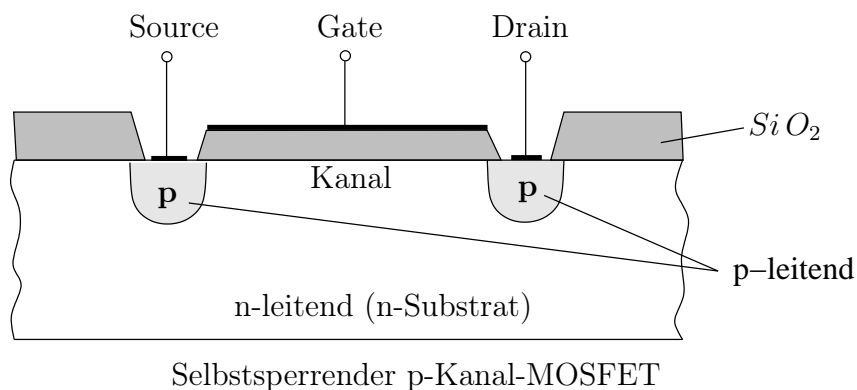
2 P.

Der Unterschied zwischen den beiden Transistortypen besteht in der gegensätzlichen Dotierung der jeweiligen Zonen der Transistoren. Beim p-Kanal-MOSFET sind Source und Drain p-dotiert (siehe Aufgabenteil 3).

n-Kanal-MOSFETs können eine logische Null gut und eine logische Eins schlecht durchschalten, bei p-Kanal-MOSFETs ist es umgekehrt. Daher werden n-Kanal-MOSFETs im n-Netz von CMOS-Schaltungen verwendet, um den Funktionswert Null durchzuschalten, und p-Kanal-MOSFETs im p-Netz, um den Funktionswert Eins durchzuschalten.

3. Aufbau eines pMOS-Transistors:

3 P.



Aufgabe 5

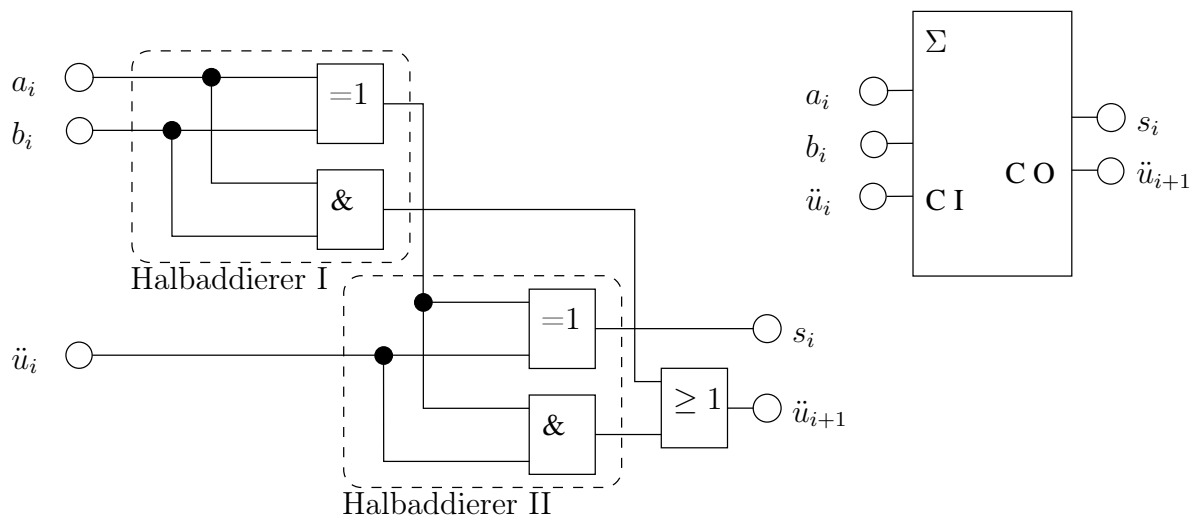
1. Unterschied zwischen Halbaddierer und Volladdierer:

1 P.

Ein Volladdierer berücksichtigt den Übertrag der vorhergehenden Stellen, deshalb besitzt er, zusätzlich zu den zwei Eingängen für die zu addierenden Dualziffern, einen Eingang für den Übertrag.

2. Schaltbild eines 1-Bit-Volladdierers:

3 P.



3. Anzahl der Prüfbits:

1 P.

Aufwand: $2^k \geq m + k + 1$. Hier: $m = 20 \Rightarrow k = 5$

4. Physikalische Ursache für Hasardfehler:

1 P.

Unterschiedliche Laufzeiten der Signale bei Eingabeänderungen aufgrund unterschiedlicher Schaltzeiten der Gatter und Leitungsverzögerungen (unterschiedliche Totzeiten der Signalpfade durch das Schaltnetz).

5. Unterschied zwischen einem PAL-Baustein und PLA-Baustein:

1 P.

Bei PAL-Bausteinen ist die ODER-Matrix bereits bei der Herstellung personalisiert, während die UND-Matrix programmierbar ist. Bei PLA-Bausteinen sind sowohl die UND- als auch die ODER-Matrix programmierbar.

6. Schieberegister als:

2 P.

- Serien-Parallel-Wandlung
- Parallel-Serien-Wandlung
- Warteschlange (FIFO-Speicher) oder Stapelspeicher (LIFO-Speicher)
- Umlaufspeicher
- Multiplikation und Division
- ...

Aufgabe 6

1. MIPS-Assembler:

3 P.

```
(a)          bne $s4, $s3, label
              add $s5, $s4, $s3
label:      ...
```

```
(b)          beq $s4, $s3, label1
              add $s5, $s4, $s3
              j label2
label1:      sub $s5, $s4, $s3
label2:      ....
```

```
(c)          slt $s5, $s3, $s4
```

2. Laden von 1111 0000 0011 1101 0000 1001 0000 1001 ins Register \$s0:

2 P.

```
lui $s0, 1111 0000 0011 1101    # load upper immediate
ori $s0, 0000 1001 0000 1001
```

oder auch

```
lui $s0, 1111 0000 0011 1101
addi $s0, $s0, 0000 1001 0000 1001
```

3. Die 2 niedrigstwertigen Bits einer Wortadresse haben den Wert 0

1 P.

4. Register- und Speicherinhalte nach der Ausführung:

4 P.

Registersatz		Hauptspeicher	
Register	Inhalt	Adresse	Inhalt
\$t0	0x10	\$0x20	0x22
\$t1	0x30	\$0x24	0x30
\$t2	0x16	\$0x28	0x30
\$t3	0x20	\$0x2C	0x50
\$t4	0x30	\$0x30	0x60

Aufgabe 7

1. Datenabhängigkeiten:

5 P.

- Echte Abhängigkeiten (*True Dependence*)

$$\begin{array}{llll}
 S_1 \rightarrow S_3 (\$t1) & S_1 \rightarrow S_9 (\$t1) & & \\
 S_2 \rightarrow S_3 (\$t2) & S_2 \rightarrow S_4 (\$t2) & S_2 \rightarrow S_6 (\$t2) & \\
 S_3 \rightarrow S_6 (\$t3) & & & \\
 S_4 \rightarrow S_9 (\$t1) & & & \\
 S_5 \rightarrow S_7 (\$t4) & & & \\
 S_6 \rightarrow S_8 (\$t5) & & &
 \end{array}$$

- Gegenabhängigkeiten (*Anti-Dependence*):

$$\begin{array}{l}
 S_1 \rightarrow S_7 (1000(\$t0)) \\
 S_2 \rightarrow S_8 (1000(\$t0)) \\
 S_3 \rightarrow S_4 (\$t1)
 \end{array}$$

- Ausgabe-Abhängigkeiten (*Output Dependence*): $S_1 \rightarrow S_4 (\$t1)$

2. Behebung der Konflikte:

3 P.

```

S1:  lw    $t1, 1000($t0)
S2:  lw    $t2, 1004($t0)
     NOP
     NOP
S3:  add   $t3, $t2, $t1
S4:  addi  $t1, $t2, 8
S5:  subi  $t4, $t0, 2
S6:  and   $t5, $t3, $t2
     NOP
S7:  sw    $t4, 1000($t0)
S8:  sw    $t5, 1004($t0)
S9:  sw    $t1, 1008($t0)

```

3. Anzahl zur Ausführung notwendigen Taktzyklen

1 P.

- Sequenzielle Ausführung: $9 * 5 \text{ Takte} = 45 \text{ Takte}$
- DLX-Pipeline ohne Forwarding: $12 + (5 - 1) \text{ Takte} = 16 \text{ Takte}$

4. Struktur- oder Ressourcenkonflikte:

1 P.

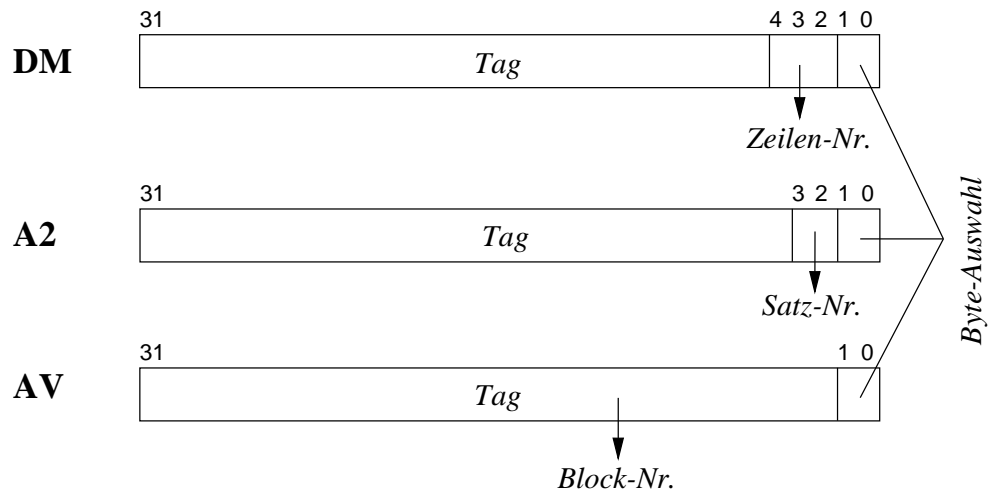
Treten auf, wenn zwei oder mehrere Pipeline-Stufen gleichzeitig dieselbe Ressource benötigen, auf diese aber nur einmal zugegriffen werden kann.

Sie können bei der DLX-Pipeline nicht auftreten, da diese entsprechend entworfen ist.

Aufgabe 8

1. Unterteilung der Hauptspeicheradresse:

3 P.



2. Anzahl der Vergleicher:

1 P.

Cache	Anzahl der Vergleicher
DM	1
A2	2
AV	8

3. »×« für Cache-Hit und »-« für Cache-Miss:

6 P.

Adresse:	0x0B	0x2B	0x07	0x0C	0x1E	0x0A	0x1A	0x05	0x04	0x29
DM	-	-	-	-	-	-	-	×	×	-
A2	-	-	-	-	-	×	-	×	×	-
AV	-	-	-	-	-	×	-	×	×	×

Aufgabe 9

1. Unterteilung der virtuellen Adresse:

1 P.



2. Physikalische Adressen:

4 P.

Virtuelle		Physikalische	
Adresse	Seitennummer	Seitennummer	Adresse
1024	0	2	$2 \cdot 2048 + 1024 = \mathbf{5120}$
2047	0	2	$2 \cdot 2048 + 2047 = \mathbf{6143}$
2048	1	0	$0 \cdot 2048 + 0 = \mathbf{0}$
2102	1	0	$0 \cdot 2048 + 54 = \mathbf{54}$
4095	1	0	$0 \cdot 2048 + 2047 = \mathbf{2047}$
4096	2	–	<i>page fault</i>
8192	4	1	$1 \cdot 2048 + 0 = \mathbf{2048}$
8202	4	1	$1 \cdot 2048 + 10 = \mathbf{2058}$

3. Eine Beschleunigung der Adressumsetzung durch den *TLB* wird erst beim zweiten Zugriff auf eine Seite und solange die entsprechenden Einträge aus dem Seitentabellen-Verzeichnis und der Seitentabelle aus dem TLB nicht verdrängt wurden erreicht.

1 P.

4. Breite des *Tags*:

2 P.

Seitengröße ist 4 KByte \Rightarrow Byte-Offset ist 12 Bit breit.

Der Tag ist dann $(n - 12)$ Bits breit

Aufgabe 10

1. Komponenten eines einfachen Rechnermodells: 1 P.
Steuerwerk, Rechenwerk, Speicher, Verbindungseinrichtung (Bus) und Eingabe-/Ausgabe-Einheiten
2.
 - (a) Befehlsregister 2 P.
 - (b) Statusregister
 - (c) Programmzähler
 - (d) Rechenwerk (ALU)
3.
 - (a) Einheitliche Befehlslänge (und einheitliches Befehlsformat) 2 P.
 - (b) Der Zugriff auf den Speicher erfolgt nur über *Load-Store-Befehle*
 - (c) festverdrahtet
 - (d) Getrennte Speicher und Busse für Befehle und Daten
4.
 - „in-order“ : Befehle werden entsprechend ihrer Programmordnung bearbeitet. 2 P.
 - „out-of-order“ : Die CPU bestimmt die Reihenfolge der abzuarbeitenden Befehle. Das Ergebnis entspricht der sequenziellen Ausführung der Befehle, ist jedoch auf die Prozessorstruktur optimiert.