



UNIVERSITÄT KARLSRUHE (TH)  
Fakultät für Informatik  
Institut für Prozessrechentchnik, Automation und Robotik (IPR)  
Prof. Dr. U. Brinkschulte, Dr. T. Asfour

**Musterlösungen**  
zur Klausur „Technische Informatik I/II“  
am 26. Januar 2006, 18.00 – 20.00 Uhr

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

<b>Technische Informatik I</b>	
Aufgabe 1	11 von 11 Punkten
Aufgabe 2	8 von 8 Punkten
Aufgabe 3	13 von 13 Punkten
Aufgabe 4	9 von 9 Punkten
Aufgabe 5	4 von 4 Punkten

<b>Technische Informatik II</b>	
Aufgabe 6	7 von 7 Punkten
Aufgabe 7	12 von 12 Punkten
Aufgabe 8	6 von 6 Punkten
Aufgabe 9	12 von 12 Punkten
Aufgabe 10	8 von 8 Punkten

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

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

# Aufgabe 1

1. DMF:

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

	$a$				
	1	1	0	-	
$b$	0	0	1	0	$d$
	0	0	1	0	
	1	-	0	0	
	$c$				

2. KMF:

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

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

	$a$				
	1	1	0	-	
$b$	0	0	1	0	$d$
	0	0	1	0	
	1	-	0	0	
	$c$				

3.

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 Kernpimimplikant mehr
$d \bar{b} a$		Nicht angrenzend an den beiden Kernpimimplikanten.
$c b \bar{a}$	X	$\bar{c} b \bar{a}$ wird kein Kernpimimplikant mehr.

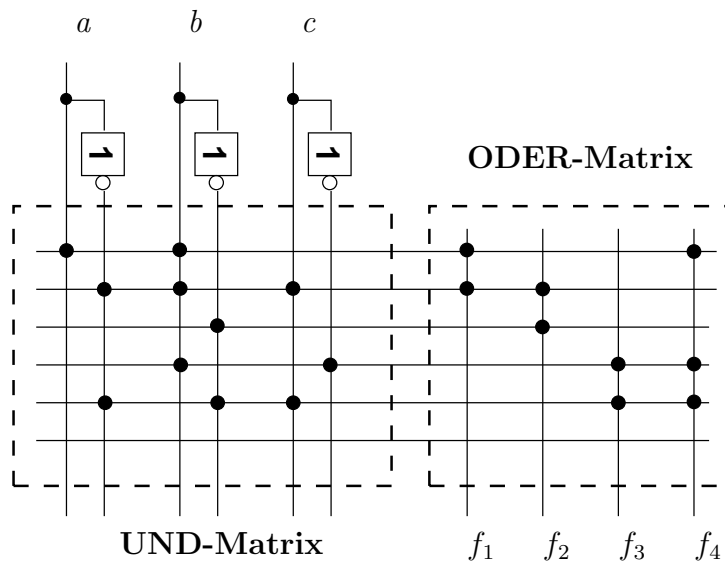
4. PLA: Bündelminimierung der Funktionen:

$$f_1 = b a \vee c b \bar{a}$$

$$f_2 = \bar{b} \vee c b \bar{a}$$

$$f_3 = \bar{c} b \vee c \bar{b} \bar{a}$$

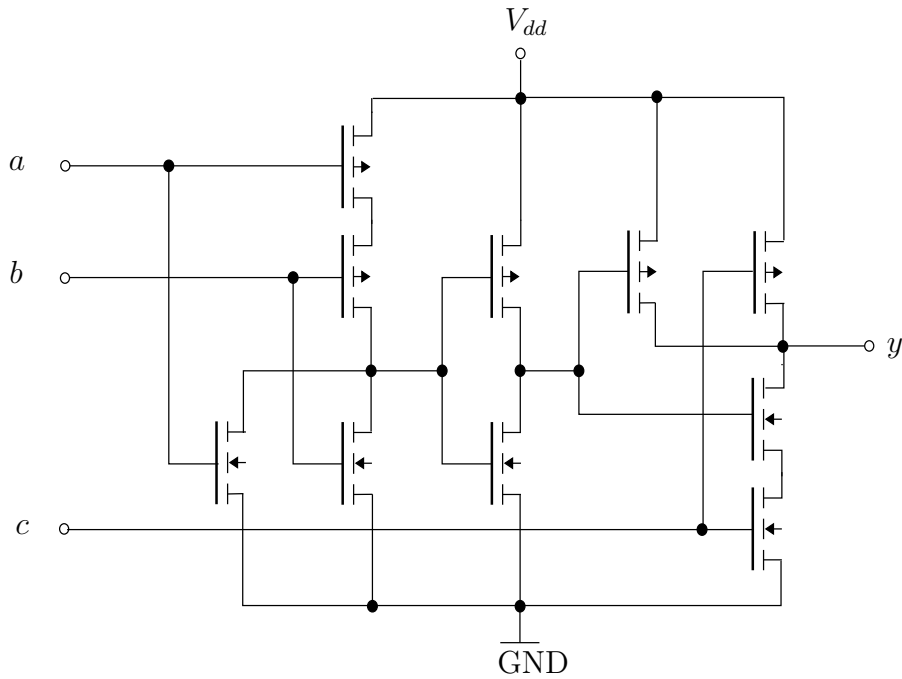
$$f_4 = \bar{c} b \vee b a \vee c \bar{b} \bar{a} = f_3 \vee b a$$



## Aufgabe 2

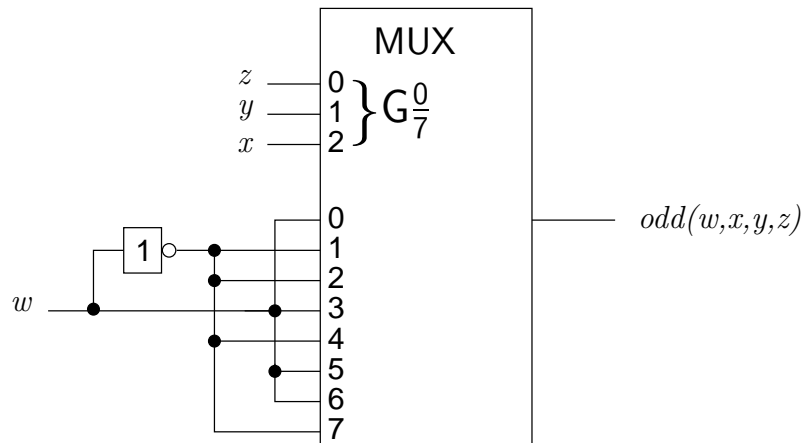
1. CMOS-Transistor-Schaltbild von  $f(c, b, a)$ :

$$f(c, b, a) = \overline{c(b \vee a)} = c \overline{(b \vee a)} = c \overline{(b \vee a)}$$



2. Schaltnetz von  $p = \text{odd}(w, x, y, z)$ :

$$p = \text{odd}(w, x, y, z) = \text{MINT}(1, 2, 4, 7, 8, 11, 13, 14)$$



### Aufgabe 3

1. Anzahl der Zustände: 2 Flipflops  $\Rightarrow 2^2$  Zustände = 4 Zustände
2. Kodierte Ablauftabelle:

- Ansteuergleichungen und Ausgabefunktion: Ablesen aus dem Schaltbild

$$\begin{aligned}
 j_B^t &= \overline{A}^t \leftrightarrow x^t & k_B^t &= \overline{A}^t \leftrightarrow x^t \\
 j_A^t &= B^t & k_A^t &= x^t \\
 y^t &= \overline{B}^t (\overline{A}^t \leftrightarrow x^t)
 \end{aligned}$$

- Übergangsgleichungen: Die charakteristische Gleichung des JK-Flipflops lautet:

$$q^{t+1} = (j \overline{q} \vee \overline{k} q)^t$$

Mit den Ansteuergleichungen der Flipflops erhält man die Zustandsübergangsgleichungen für  $A$ , und  $B$ :

$$\begin{aligned}
 B^{t+1} &= (\overline{A}^t \leftrightarrow x^t) \overline{B}^t \vee \overline{(\overline{A}^t \leftrightarrow x^t)} B^t \\
 A^{t+1} &= B^t \overline{A}^t \vee \overline{x} A^t
 \end{aligned}$$

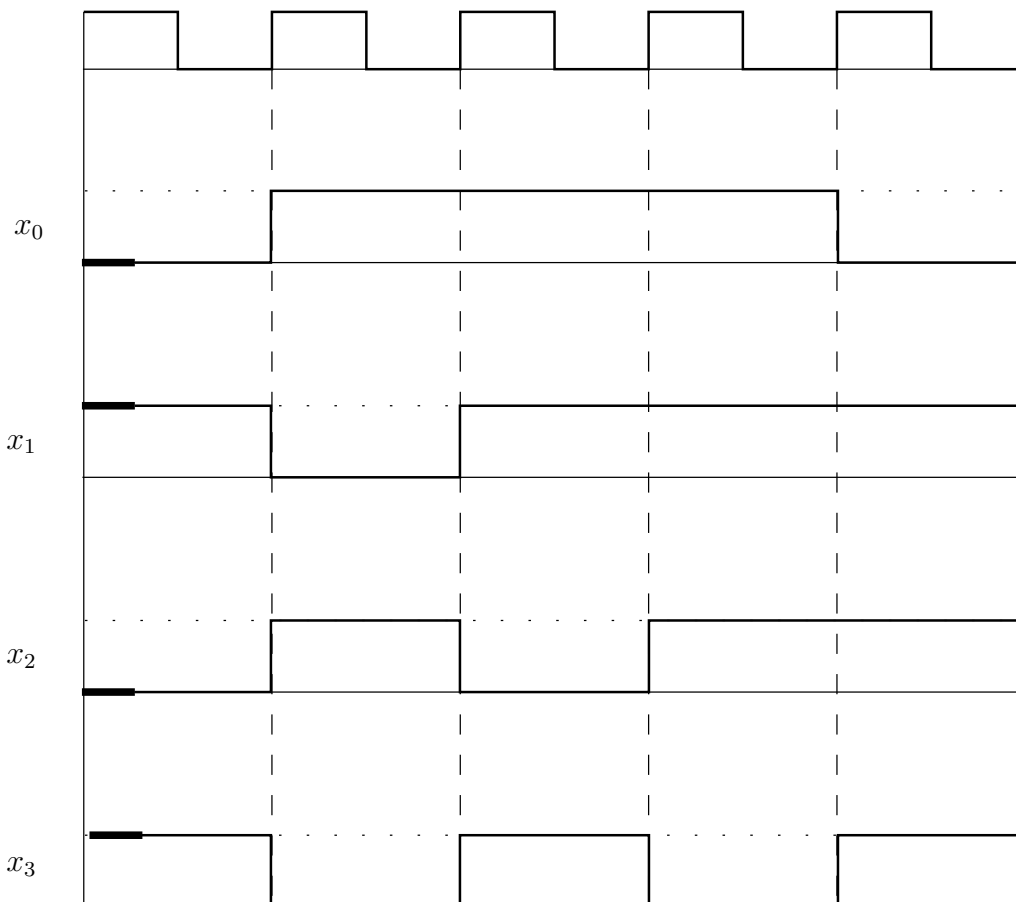
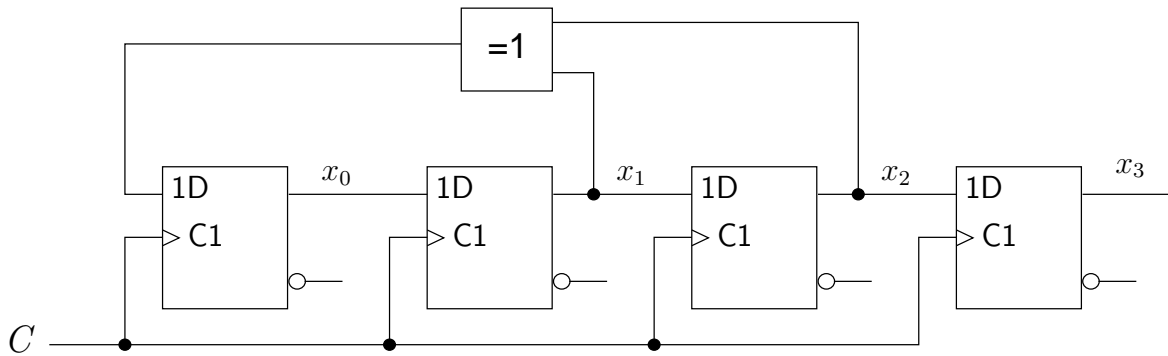
- Kodierte Ablauftabelle:

Zustand		Eingabe	Folgezustand		Ausgabe
$A^t$	$B^t$	$x^t$	$A^{t+1}$	$B^{t+1}$	$y^t$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	0	0

3. Eingabe-, Ausgabe-, und Zustandsfolgen:

$t$	1	2	3	4	5	6	7	8	9	10
$e$	1	0	1	1	0	0	1	0	1	1
$a$	0	0	0	1	1	0	1	1	0	1
$Z_i$	$Z_0$	$Z_1$	$Z_2$	$Z_3$	$Z_2$	$Z_0$	$Z_3$	$Z_2$	$Z_0$	$Z_1$

4. Verläufe der Signale  $x_0, x_1, x_2$  und  $x_3$ :



## Aufgabe 4

1.  $86, 22_{10}$  zur Basis 5:  $86, 22_{10} = 321, 10\bar{2}$

$$86 = 3 \times 5^2 + 2 \times 5^1 + 1 \times 5^0$$

$$0, 22 \times 5 = 1, 1 \quad 0, 1 \times 5 = 0, 5 \quad 0, 5 \times 5 = 2, 5 \dots$$

2.  $435, 317_8$  zur Basis 16:  $435, 317_8 = (100011101, 011001111)_2 = 11D, 678_{16}$

3.  $-65_{10}$  in eine 16-Bit Zweierkomplement Zahl:  $-65_{10} = 1111\ 1111\ 1011\ 1111_{ZK}$

$$65 = 0000\ 0000\ 0100\ 0001$$

$$\text{Einerkomplement: } 1111\ 1111\ 1011\ 1110$$

$$\text{Zweierkomplement: } 1111\ 1111\ 1011\ 1111$$

4.  $(1111111100111100)_{ZK} = -196_{10}$

5.  $(11000001111001010000000000000000)_{IEEE} = -28, 625$

$$VZ = 1, \text{ Char} = 10000011_2 = 131_{10} \Rightarrow \text{Exp} = (131 - 127)_{10} = 4_{10}$$

$$\text{Mantisse} = 110010100000000000000000$$

$$\text{Zahlenwert} = (-1)^1 \cdot (1, 110010100 \dots 0)_2 \cdot 2^4 = -(11100, 10100 \dots 0)_2 = -28,625_{10}$$

6.  $\ddot{u}_4$ :

$$\begin{aligned} \ddot{u}_4 &= g_3 \vee p_3(g_2 \vee p_2(g_1 \vee p_1(g_0 \vee p_0 \ddot{u}_0))) \\ &= b_3 \vee \bar{b}_3 b_2 \vee \bar{b}_3 \bar{b}_2 b_1 \vee \bar{b}_3 \bar{b}_2 \bar{b}_1 b_0 \\ &= b_3 \vee b_2 \vee b_1 \vee b_0 \end{aligned}$$

Dabei ist:

$$g_3 = b_3, p_3 = \bar{b}_3, g_2 = b_2, p_2 = \bar{b}_2, g_1 = b_1, p_1 = \bar{b}_1, g_0 = 0, p_0 = b_0, \ddot{u}_0 = 1$$

### 2. Lösungsweg:

Die einzige Eingabe, welche bei  $A = 1110$  und  $\ddot{u}_0 = 1$  keinen Übertrag generiert, ist  $B = 0000$ . Somit gilt  $\bar{\bar{u}}_4 = \bar{b}_3 \bar{b}_2 \bar{b}_1 \bar{b}_0 \Rightarrow \ddot{u}_4 = b_3 \vee b_2 \vee b_1 \vee b_0$

## Aufgabe 5

	richtig	falsch
Die Normalformen einer Booleschen Funktion sind eindeutig.	×	
Implikanten $n$ -ter Ordnung überdecken, bei vollständig definierten Schaltfunktionen, $2^n$ Minterme.	×	
Funktionshasards in einem Schaltnetz können behoben werden.		×
Laufzeiteffekte von Gattern sind bei asynchronen Schaltwerken unkritisch.		×
Bei Mealy-Schaltwerken erfolgen Änderungen der Ausgabe nur synchron zum Takt.		×
Die Zustandskodierung bei synchronen Schaltwerken beeinflusst die Größe der Ausgabe- und Übergangs-Schaltnetze.	×	
Die Additionszeit bei <i>Carry-Ripple</i> -Addierern ist proportional zur Anzahl der zu addierenden Stellen.	×	
<i>minreal</i> ist die kleinste darstellbare positive Zahl.		×

## Aufgabe 6

1. Mikroprogramme:

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

2. Anzahl der ALU-Operationen: 8

3. Maximale Anzahl der MIMA-Befehle: 31

4. Adressen bei MIMA: Die höchstwertigen 4 Bits werden abgeschnitten.

## Aufgabe 7

1. (a) do-while-Schleife in MIPS:

```

Loop:   add $s3, $s3, $s4
        add $t1, $s3, $s3
        add $t1, $t1, $t1
        add $t1, $t1, $s6
        lw  $t0, 0($t1)
        beq $t0, $s5, Loop

```

- (b) else-if-Anweisungen in MIPS:

```

        bne $s3, $s4, Else1
        add $t1, $s5, $s5
        add $t1, $t1, $t1
        add $t1, $t1, $s6
        lw  $t0, 0($t1)
        add $s3, $s3, $t0
        j   ferig
Else1:  bne $s3, $s5, Else2
        add $t1, $s4, $s4
        add $t1, $t1, $t1
        add $t1, $t1, $s6
        lw  $t0, 0($t1)
        add $s3, $s3, $t0
        j   ferig
Else1:  add $s3, $s4, $s5

```

- 2.

- (a) `ptr = &i;` in MIPS-Assembler:

```

        la $t0, i      # Adresse bestimmen
        sw $t0, ptr

```

- (b) `i = *ptr;` in MIPS-Assembler:

```

        lw $t0, ptr
        lw $t0, 0($t0)
        sw $t0, i

```

- (c) `i = **ptr` in MIPS-Assembler:

```

        lw $t0, ptr
        lw $t0, 0($t0)
        lw $t0, 0($t0)
        sw $t0, i

```

(d) \*ptr = i in MIPS-Assembler:

```
lw $t0, ptr      # $t0 = Adresse von *ptr
lw $t1, i        # $t1 = i
sw $t1, 0($t0)  # i an der Adresse ptr schreiben
```

## Aufgabe 8

1. (a) Echte Abhängigkeiten (*True Dependence*)

$$\begin{array}{llll}
 S_1 \rightarrow S_3 (R1) & S_1 \rightarrow S_9 (R1) & & \\
 S_2 \rightarrow S_3 (R2) & S_2 \rightarrow S_4 (R2) & S_2 \rightarrow S_6 (R2) & \\
 S_3 \rightarrow S_6 (R3) & & & \\
 S_4 \rightarrow S_9 (R1) & & & \\
 S_5 \rightarrow S_7 (R4) & & & \\
 S_6 \rightarrow S_8 (R5) & & & 
 \end{array}$$

(b) Behebung der Konflikte:

```
S1:   lw    R1, 1000(R0)
S2:   lw    R2, 1004(R0)
      NOP
S3:   add   R3, R2, R1
S4:   addi  R1, R2, 8
S5:   subi  R4, R0, 2
S6:   and   R5, R3, R2
S7:   sw    R4, 1000(R0)
S8:   sw    R5, 1004(R0)
S9:   sw    R1, 1008(R0)
```

## Aufgabe 9

1. (a) Blockgröße in Bytes: 4 Bit Byte Offset  $\Rightarrow$  Blockgröße =  $2^4 = 16$  Byte

(b) Anzahl der Einträge:

$$\text{Anzahl der Einträge} = \frac{\text{Kapazität}}{\text{Blockgröße}} = \frac{512 \text{ KByte}}{16 \text{ Byte}} = 32 \text{ K Einträge}$$

(c) Cache-Organisation:

12 Bit Index-Feld  $\Rightarrow$  Es lassen sich  $2^{12} = 4 \text{ K}$  Sätze im Cache adressieren

$$\text{Assoziativität} = \frac{32 \text{ K}}{4 \text{ K}} = 8$$

Der Cache ist als 8-fach assoziativer Speicher (*8-way set associative*) organisiert

2. Speicherbedarf:

Für jede Zeile sind (Tag + 1 Statusbit + Daten pro Zeile) Bits erforderlich.

- Daten pro Zeile  $8 \text{ Byte} \times 8 = 64 \text{ Bit}$
- Tag =  $32 - 8 - 3 = 21 \text{ Bit}$  (8 Bit Satzindex und 3 Byte-Offset)

Speicherbedarf für eine Zeile:  $21 + 1 + 64 \text{ Bit} = 86 \text{ Bits}$

Speicherbedarf für den gesamten Cache:  $86 \cdot 256 \cdot 4 = 88064 \text{ Bits} = 11008 \text{ Byte}$

3.

<b>Adresse</b>	<b>0</b>	<b>16</b>	<b>48</b>	<b>8</b>	<b>56</b>	<b>16</b>	<b>8</b>	<b>56</b>	<b>32</b>	<b>0</b>
<b>read/write</b>	r	r	r	r	r	r	r	w	w	r
<b>Index</b>	0	2	2	1	3	2	1	3	0	0
<b>Tag</b>	0	0	1	0	1	0	0	1	1	0
<b>Hit/Miss</b>	Miss	Miss	Miss	Miss	Miss	Miss	Hit	Hit	Miss	Hit

